

Universal Chiplet Interconnect Express™ (UCIe™)

Specification

July 10, 2023

Revision 1.1, Version 1.0

LEGAL NOTICE FOR THIS PUBLICLY AVAILABLE SPECIFICATION FROM UNIVERSAL CHIPLET INTERCONNECT EXPRESS, INC.

© 2022–2023 UNIVERSAL CHIPLET INTERCONNECT EXPRESS, INC. ALL RIGHTS RESERVED.

This Universal Chiplet Interconnect Express, Inc. Specification (this "**UCIe Specification**" or this "**document**") is owned by and is proprietary to Universal Chiplet Interconnect Express, Inc., a Delaware nonprofit corporation (sometimes referred to as "**UCIe**" or the "**UCIe Consortium**" or the "**Company**") and/or its successors and assigns.

NOTICE TO USERS WHO ARE MEMBERS OF UNIVERSAL CHIPLET INTERCONNECT EXPRESS, INC.:

If you are a Member of Universal Chiplet Interconnect Express, Inc. (herein referred to as a "**UCIe Member**"), and even if you have received this publicly available version of this UCIe Specification after agreeing to the UCIE Consortium's Evaluation Copy Agreement (a copy of which is available at www.uciexpress.org/specifications, each such UCIE Member must also be in compliance with all of the following UCIe Consortium documents, policies and/or procedures (collectively, the "**UCIe Governing Documents**") in order for such UCIe Member's use and/or implementation of this UCIe Specification to receive and enjoy all of the rights, benefits, privileges and protections of UCIe Consortium membership: (i) the UCIe Consortium's Intellectual Property Policy; (ii) the UCIe Consortium's Bylaws; (iii) any and all other UCIe Consortium policies and procedures; and (iv) the UCIe Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF UNIVERSAL CHIPLET INTERCONNECT EXPRESS, INC.:

If you are **not** a UCIe Member and have received this publicly available version of this UCIe Specification, your use of this document is subject to your compliance with, and is limited by, all of the terms and conditions of the UCIe Consortium's Evaluation Copy Agreement (a copy of which is available at www.uciexpress.org/specifications).

In addition to the restrictions set forth in the UCIe Consortium's Evaluation Copy Agreement, any references or citations to this document must acknowledge Universal Chiplet Interconnect Express, Inc.'s sole and exclusive copyright ownership of this UCIe Specification. The proper copyright citation or reference is as follows: **"© 2022–2023 UNIVERSAL CHIPLET INTERCONNECT EXPRESS, INC. ALL RIGHTS RESERVED."** When making any such citation or reference to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of Universal Chiplet Interconnect Express, Inc.

Except for the limited rights explicitly given to a non-UCIe Member pursuant to the explicit provisions of the UCIe Consortium's Evaluation Copy Agreement which governs the publicly available version of this UCIe Specification, nothing contained in this UCIe Specification shall be deemed as granting (either expressly or impliedly) to any party that is not a UCIe Member: (i) any kind of license to implement or use this UCIe Specification or any portion or content described or contained therein, or any kind of license in or to any other intellectual property owned or controlled by the UCIe Consortium, including without limitation any trademarks of the UCIe Consortium; or (ii) any benefits and/or rights as a UCIe Member under any UCIe Governing Documents. For clarity, and without limiting the foregoing notice in any way, if you are **not** a UCIe Member but still elect to implement this UCIe Specification or any portion described herein, you are hereby given notice that your election to do so does not give you any of the rights, benefits, and/or protections of the UCIe Members, including without limitation any of the rights, benefits, privileges or protections given to a UCIe Member under the UCIe Consortium's Intellectual Property Policy.

LEGAL DISCLAIMERS AND ADDITIONAL NOTICE TO ALL PARTIES:

THIS DOCUMENT AND ALL SPECIFICATIONS AND/OR OTHER CONTENT PROVIDED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, UNIVERSAL CHIPLET INTERCONNECT EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NON-INFRINGEMENT.

In the event this UCIe Specification makes any references (including without limitation any incorporation by reference) to another standard's setting organization's or any other party's ("**Third Party**") content or work, including without limitation any specifications or standards of any such Third Party ("**Third Party Specification**"), you are hereby notified that your use or implementation of any Third Party Specification: (i) is not governed by any of the UCIe Governing Documents; (ii) may require your use of a Third Party's patents, copyrights or other intellectual property rights, which in turn may require you to independently obtain a license or other consent from that Third Party in order to have full rights to implement or use that Third Party Specification; and/or (iii) may be governed by the intellectual property policy or other policies or procedures of the Third Party which owns the Third Party Specification. Any trademarks or service marks of any Third Party which may be referenced in this UCIe Specification are owned by the respective owner of such marks.

The **UCIe**™ and **UNIVERSAL CHIPLET INTERCONNECT EXPRESS**™ trademarks and logos (the "**UCIe Trademarks**") are trademarks owned by the UCIe Consortium. The UCIe Consortium reserves all rights in and to all of its UCIe Trademarks.

NOTICE TO ALL PARTIES REGARDING THE PCI-SIG UNIQUE VALUE PROVIDED IN THIS SPECIFICATION:
NOTICE TO USERS: THE UNIQUE VALUE THAT IS PROVIDED IN THIS SPECIFICATION FOR USE IN VENDOR DEFINED MESSAGE FIELDS, DESIGNATED VENDOR SPECIFIC EXTENDED CAPABILITIES, AND ALTERNATE PROTOCOL NEGOTIATION ONLY AND MAY NOT BE USED IN ANY OTHER MANNER, AND A USER OF THE UNIQUE VALUE MAY NOT USE THE UNIQUE VALUE IN A MANNER THAT (A) ALTERS, MODIFIES, HARMS OR DAMAGES THE TECHNICAL FUNCTIONING, SAFETY OR SECURITY OF THE PCI-SIG ECOSYSTEM OR ANY PORTION THEREOF, OR (B) COULD OR WOULD REASONABLY BE DETERMINED TO ALTER, MODIFY, HARM OR DAMAGE THE TECHNICAL FUNCTIONING, SAFETY OR SECURITY OF THE PCI-SIG ECOSYSTEM OR ANY PORTION THEREOF (FOR PURPOSES OF THIS NOTICE, "**PCI-SIG ECOSYSTEM**" MEANS THE PCI-SIG SPECIFICATIONS, MEMBERS OF PCI-SIG AND THEIR ASSOCIATED PRODUCTS AND SERVICES THAT INCORPORATE ALL OR A PORTION OF A PCI-SIG SPECIFICATION AND EXTENDS TO THOSE PRODUCTS AND SERVICES INTERFACING WITH PCI-SIG MEMBER PRODUCTS AND SERVICES).

Contents

Terminology 18
Reference Documents 23
Revision History 24

1.0 Introduction 25
1.1 UCIe Components 28
1.1.1 Protocol Layer 29
1.1.2 Die-to-Die (D2D) Adapter 29
1.1.3 Physical Layer 30
1.1.4 Interfaces 30
1.2 UCIe Configurations 31
1.2.1 Single Module configuration 31
1.2.2 Multi-module Configurations 32
1.3 UCIe Retimers 32
1.4 UCIe Key Performance Targets 34
1.5 Interoperability 35

2.0 Protocol Layer 36
2.1 PCIe 37
2.1.1 Raw Format 38
2.1.2 Standard 256B End Header Flit Format 38
2.1.3 68B Flit Format 38
2.1.4 Standard 256B Start Header Flit Format 38
2.1.5 Latency-Optimized 256B with Optional Bytes Flit Format 39
2.2 CXL 256B Flit Mode 39
2.2.1 Raw Format 39
2.2.2 Latency-Optimized 256B Flit Formats 39
2.2.3 Standard 256B Start Header Flit Format 40
2.3 CXL 68B Flit Mode 40
2.3.1 Raw Format 40
2.3.2 68B Flit Format 40
2.4 Streaming protocol 41
2.4.1 Raw Format 41
2.4.2 68B Flit Format 41
2.4.3 Standard 256B Flit Formats 41
2.4.4 Latency-Optimized 256B Flit Formats 42

3.0 Die-to-Die Adapter 43
3.1 Stack Multiplexing 44
3.2 Link Initialization 46
3.2.1 Stage 3 of Link Initialization: Adapter Initialization 46
3.2.1.1 Part 1: Determine Local Capabilities 46
3.2.1.2 Part 2: Parameter Exchange with remote Link partner 47
3.2.1.3 Part 3: FDI bring up 55
3.3 Operation Formats 55
3.3.1 Raw Format for all protocols 55
3.3.2 68B Flit Format 55
3.3.3 Standard 256B Flit Formats 59
3.3.4 Latency-Optimized 256B Flit Formats 62
3.3.5 Flit Format-related Implementation Requirements for Protocol Layer and Adapter 66
3.4 Decision table for Flit Format and Protocol 68

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.5 State Machine Hierarchy 70
3.6 Power Management Link States 71
3.7 CRC Computation 73
3.8 Retry Rules 74
3.9 Runtime Link Testing using Parity 75

4.0 Logical Physical Layer 78
4.1 Data and Sideband Transmission Flow 78
4.1.1 Byte to Lane Mapping 78
4.1.2 Valid Framing 80
4.1.2.1 Valid Framing for Retimers 81
4.1.3 Clock Gating 81
4.1.4 Free Running Clock Mode 81
4.1.5 Sideband transmission 81
4.2 Lane Reversal 82
4.2.1 Lane ID 83
4.3 Interconnect redundancy remapping 84
4.3.1 Data Lane repair 84
4.3.2 Data Lane repair with Lane reversal 87
4.3.3 Data Lane repair implementation 87
4.3.3.1 Single Lane repair 87
4.3.3.2 Two Lane repair 89
4.3.3.3 Single Lane repair with Lane reversal 91
4.3.3.3.1 x64 Advanced Package Pseudo Codes 91
4.3.3.3.2 x32 Advanced Package Pseudo Codes 91
4.3.3.4 Two Lane repair with Lane reversal 92
4.3.3.4.1 x64 Advanced Package Pseudo Codes 92
4.3.3.4.2 x32 Advanced Package Pseudo Codes 93
4.3.4 Clock and Track Lane remapping 94
4.3.5 Clock and Track Lane repair implementation 95
4.3.6 Valid Repair and implementation 96
4.3.7 Width Degrade in Standard Package Interfaces 97
4.4 Data to Clock Training and Test Modes 97
4.4.1 Scrambling and training pattern generation 99
4.5 Link Initialization and Training 102
4.5.1 Link Training basic operations 102
4.5.1.1 Transmitter initiated Data to Clock point test 102
4.5.1.2 Transmitter initiated Data to Clock eye width sweep 103
4.5.1.3 Receiver initiated Data to Clock point test 104
4.5.1.4 Receiver initiated Data to Clock Eye Width sweep 105
4.5.2 Link Training with Retimer 106
4.5.3 Link Training State Machine 107
4.5.3.1 RESET 108
4.5.3.2 Sideband Initialization (SBINIT) 109
4.5.3.3 MBINIT 112
4.5.3.3.1 MBINIT.PARAM 112
4.5.3.3.2 MBINIT.CAL 113
4.5.3.3.3 MBINIT.REPAIRCLK 113
4.5.3.3.4 MBINIT.REPAIRVAL 116
4.5.3.3.5 MBINIT.REVERSALMB 119
4.5.3.3.6 MBINIT.REPAIRMB 120
4.5.3.4 MBTRAIN 122
4.5.3.4.1 MBTRAIN.VALVREF 123
4.5.3.4.2 MBTRAIN.DATAVREF 124
4.5.3.4.3 MBTRAIN.SPEEDIDLE 125
4.5.3.4.4 MBTRAIN.TXSELFCAL 125

 Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5.3.4.5 MBTRAIN.RXCLKCAL 125
4.5.3.4.6 MBTRAIN.VALTRAINCENTER 126
4.5.3.4.7 MBTRAIN.VALTRAINVREF 126
4.5.3.4.8 MBTRAIN.DATATRAINCENTER1 127
4.5.3.4.9 MBTRAIN.DATATRAINVREF 127
4.5.3.4.10 MBTRAIN.RXDESKEW 128
4.5.3.4.11 MBTRAIN.DATATRAINCENTER2 129
4.5.3.4.12 MBTRAIN.LINKSPEED 129
4.5.3.4.13 MBTRAIN.REPAIR 132
4.5.3.5 LINKINIT 133
4.5.3.6 ACTIVE 133
4.5.3.7 PHYRETRAIN 133
4.5.3.7.1 Adapter initiated PHY retrain 134
4.5.3.7.2 PHY initiated PHY retrain 135
4.5.3.7.3 Remote Die requested PHY retrain 135
4.5.3.7.4 PHY retrain from LINKSPEED 136
4.5.3.8 TRAINERROR 136
4.5.3.9 L1/L2 137
4.6 Runtime Recalibration 137
4.7 Multi-module Link 137
4.7.1 Multi-module initialization 137
4.7.1.1 Sideband Assignment and Retimer Credits for Multi-module Configurations 141
4.7.2 Multi-module Interoperability between x64 and x32 Advanced Packages 141

5.0 Electrical Layer 144
5.1 Interoperability 144
5.1.1 Data rates 144
5.2 Overview 144
5.2.1 Interface Overview 144
5.2.2 Electrical summary 146
5.3 Transmitter Specification 147
5.3.1 Driver Topology 147
5.3.2 Transmitter Electrical parameters 148
5.3.3 24 and 32 GT/s Transmitter Equalization 149
5.4 Receiver Specification 151
5.4.1 Receiver Electrical Parameters 152
5.4.2 Rx Termination 152
5.4.3 24 and 32 GT/s Receiver Equalization 154
5.5 Clocking 155
5.5.1 Track 156
5.6 Supply noise and clock skew 158
5.7 Ball-out and Channel Specification 159
5.7.1 Voltage Transfer Function 161
5.7.2 Advanced Package 162
5.7.2.1 Loss and Crosstalk Mask 163
5.7.2.2 x64 Advanced Package Module Bump Map 163
5.7.2.3 x32 Advanced Package Module Bump Map 168
5.7.2.4 x64 and x32 Advanced Package Module Interoperability 171
5.7.2.5 Module Naming of Advanced Package Modules 174
5.7.3 Standard Package 178
5.7.3.1 Standard Package Module Bump Map 179
5.7.3.2 Module Naming of Standard Package Modules 182
5.7.3.2.1 Module Degrade Rules 188
5.8 Tightly Coupled mode 189
5.9 Interconnect redundancy remapping 189

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.9.1 Advanced Package Lane remapping 189
5.9.2 Standard Package Lane remapping 190
5.10 BER requirements, CRC and retry 190
5.11 Valid and Clock Gating 191
5.12 Electrical Idle 193
5.13 Sideband signaling 193
5.13.1 Sideband Electrical Parameters 194

6.0 Sideband 195
6.1 Protocol specification 195
6.1.1 Packet Types 196
6.1.2 Packet Formats 198
6.1.2.1 Register Access Packets 198
6.1.2.2 Messages without Data 201
6.1.2.3 Messages with data payloads 208
6.1.3 Flow Control and Data Integrity 216
6.1.3.1 Flow Control and Data Integrity over FDI and RDI 216
6.1.3.2 Flow Control and Data Integrity over UCIe sideband Link between dies 216
6.1.3.3 End-to-End flow control and forward progress for UCIe Link sideband 216
6.1.4 Operation on RDI and FDI 219

7.0 Configuration and Parameters 220
7.1 High level Software view of UCIe 220
7.2 SW Discovery of UCIe Links 221
7.3 Register Location Details and Access Mechanism 221
7.4 Software view Examples 223
7.5 UCIe Registers 226
7.5.1 UCIe Link DVSEC 226
7.5.1.1 PCI Express Extended Capability Header (Offset 0h) 228
7.5.1.2 Designated Vendor Specific Header 1, 2 (Offsets 4h and 8h) 228
7.5.1.3 Capability Descriptor (Offset Ah) 229
7.5.1.4 UCIe Link DVSEC - UCIe Link Capability (Offset Ch) 230
7.5.1.5 UCIe Link DVSEC - UCIe Link Control (Offset 10h) 231
7.5.1.6 UCIe Link DVSEC - UCIe Link Status (Offset 14h) 233
7.5.1.7 UCIe Link DVSEC - Link Event Notification Control (Offset 18h) 235
7.5.1.8 UCIe Link DVSEC - Error Notification Control (Offset 1Ah) 235
7.5.1.9 UCIe Link DVSEC - Register Locator 0, 1, 2, 3 Low (Offset 1Ch and when Register Locators 1, 2, 3 are present Offsets 24h, 2Ch, and 34h respectively) 239
7.5.1.10 UCIe Link DVSEC - Register Locator 0, 1, 2, 3 High (Offset 20h and when Register Locators 1, 2, 3 Are Present Offsets 28h, 30h, and 38h respectively) 240
7.5.1.11 UCIe Link DVSEC - Sideband Mailbox Index Low (Offset is design dependent) 240
7.5.1.12 UCIe Link DVSEC - Sideband Mailbox Index High (Offset is design dependent) 241
7.5.1.13 UCIe Link DVSEC - Sideband Mailbox Data Low (Offset is design dependent) 241
7.5.1.14 UCIe Link DVSEC - Sideband Mailbox Data High (Offset is design dependent) 241
7.5.1.15 UCIe Link DVSEC - Sideband Mailbox Control (Offset is design dependent) 242
7.5.1.16 UCIe Link DVSEC - Sideband Mailbox Status (Offset is design dependent) 242
7.5.1.17 UCIe Link DVSEC - Requester ID (Offset is design dependent) ... 242

 Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.18 UCIe Link DVSEC - Associated Port Numbers (Offset is design dependent) 243
7.5.1.19 Examples of setting the Length field in DVSEC for various Scenarios 243
7.5.2 UCIe Switch Register Block (UiSRB) DVSEC Capability 243
7.5.2.1 PCI Express Extended Capability Header (Offset 0h) 243
7.5.2.2 Designated Vendor Specific Header 1, 2 (Offsets 4h and 8h) 243
7.5.2.3 UCIe Switch Register Block (UiSRB) Base Address (Offset Ch) 244
7.5.3 D2D/PHY Register Block 245
7.5.3.1 UCIe Register Block Header 245
7.5.3.2 Uncorrectable Error Status Register (Offset 10h) 245
7.5.3.3 Uncorrectable Error Mask Register (Offset 14h) 246
7.5.3.4 Uncorrectable Error Severity Register (Offset 18h) 247
7.5.3.5 Correctable Error Status Register (Offset 1Ch) 247
7.5.3.6 Correctable Error Mask Register (Offset 20h) 248
7.5.3.7 Header Log 1 Register (Offset 24h) 248
7.5.3.8 Header Log 2 Register (Offset 2Ch) 249
7.5.3.9 Error and Link Testing Control Register (0ffset 30h) 250
7.5.3.10 Runtime Link Testing Parity Log 0 (Offset 34h) 251
7.5.3.11 Runtime Link Testing Parity Log 1 (Offset 3Ch) 251
7.5.3.12 Runtime Link Testing Parity Log 2 (Offset 44h) 252
7.5.3.13 Runtime Link Testing Parity Log 3 (Offset 4Ch) 252
7.5.3.14 Advertised Adapter Capability Log (Offset 54h) 252
7.5.3.15 Finalized Adapter Capability Log (Offset 5Ch) 252
7.5.3.16 Advertised CXL Capability Log (Offset 64h) 253
7.5.3.17 Finalized CXL Capability Log (Offset 6Ch) 253
7.5.3.18 Advertised Multi-Protocol Capability Log Register (Offset 78h) 253
7.5.3.19 Finalized Multi-Protocol Capability Log Register (Offset 80h) 253
7.5.3.20 Advertised CXL Capability Log Register for Stack 1 (Offset 88h) 254
7.5.3.21 Finalized CXL Capability Log Register for Stack 1 (Offset 90h) 254
7.5.3.22 PHY Capability (Offset 1000h) 255
7.5.3.23 PHY Control (Offset 1004h) 256
7.5.3.24 PHY Status (Offset 1008h) 257
7.5.3.25 PHY Initialization and Debug (Offset 100Ch) 258
7.5.3.26 Training Setup 1 (Offset 1010h) 259
7.5.3.27 Training Setup 2 (Offset 1020h) 259
7.5.3.28 Training Setup 3 (Offset 1030h) 260
7.5.3.29 Training Setup 4 (Offset 1050h) 260
7.5.3.30 Current Lane Map Module 0 (Offset 1060h) 261
7.5.3.31 Current Lane Map Module 1 (Offset 1068h) 261
7.5.3.32 Current Lane Map Module 2 (Offset 1070h) 261
7.5.3.33 Current Lane Map Module 3 (Offset 1078h) 261
7.5.3.34 Error Log 0 (Offset 1080h) 262
7.5.3.35 Error Log 1 (Offset 1090h) 263
7.5.3.36 Runtime Link Test Control (Offset 1100h) 264
7.5.3.37 Runtime Link Test Status (Offset 1108h) 266
7.5.3.38 Mainband Data Repair (Offset 110Ch) 266
7.5.3.39 Clock, Track, Valid and Sideband Repair (Offset 1134h) 268
7.5.3.40 UCIe Link Health Monitor (UHM) DVSEC 269
7.5.3.40.1 UHM Status (Offset Eh) 270
7.5.3.40.2 Eye Margin (Starting Offset 10h) 270
7.5.4 Test/Compliance Register Block 271
7.5.4.1 UCIe Register Block Header 272
7.5.4.2 D2D Adapter Test/Compliance Register Block Offset 272
7.5.4.3 PHY Test/Compliance Register Block Offset 272
7.5.4.4 D2D Adapter Test/Compliance Register Block 273
7.5.4.4.1 Adapter Compliance Control 273
7.5.4.4.2 Flit Tx Injection Control 274
7.5.4.4.3 Adapter Test Status (Offset 30h from D2DOFF) 275

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.4.4 Link State Injection Control Stack 0 (Offset 34h from D2DOFF) 276
7.5.4.4.5 Link State Injection Control Stack 1 (Offset 38h from D2DOFF) 277
7.5.4.4.6 Retry Injection Control (Offset 40h from D2DOFF) .. 277
7.5.4.5 PHY Test/Compliance Register Block 279
7.5.4.5.1 Physical Layer Compliance Control 1 (Offsets 000h, 400h, 800h, and C00h from PHYOFF) 279
7.5.4.5.2 Physical Layer Compliance Control 2 (Offsets 008h, 408h, 808h, and C08h from PHYOFF) 281
7.5.4.5.3 Physical Layer Compliance Status 1 (Offsets 010h, 410h, 810h, and C10h from PHYOFF) 282
7.5.4.5.4 Physical Layer Compliance Status 2 (Offsets 018h, 418h, 818h, and C18h from PHYOFF) 282
7.5.4.5.5 Physical Layer Compliance Status 3 (Offsets 020h, 420h, 820h, and C20h from PHYOFF) 283
7.5.5 Implementation Specific Register Blocks 283
7.6 UCIe Link Registers in Streaming Mode and System SW/FW Implications 284
7.7 MSI and MSI-X Capability in Hosts/Switches for UCIe interrupt 285
7.8 UCIe Early Discovery Table (UEDT) 285

8.0 Interface Definitions 287
8.1 Raw Die-to-Die Interface (RDI) 287
8.1.1 Interface reset requirements 291
8.1.2 Interface clocking requirements 291
8.1.3 Dynamic clock gating 291
8.1.3.1 Rules and description for lp_wake_req/pl_wake_ack handshake 291
8.1.3.2 Rules and description for pl_clk_req/lp_clk_ack handshake 292
8.1.4 Data Transfer 293
8.1.5 RDI State Machine 294
8.1.6 RDI bring up flow 295
8.1.7 RDI PM flow 296
8.2 Flit-Aware Die-to-Die Interface (FDI) 301
8.2.1 Interface reset requirements 307
8.2.2 Interface clocking requirements 308
8.2.3 Dynamic clock gating 308
8.2.3.1 Rules and description for lp_wake_req/pl_wake_ack handshake 308
8.2.3.2 Rules and description for pl_clk_req/lp_clk_ack handshake 309
8.2.4 Data Transfer 310
8.2.4.1 DLLP transfer rules for 256B Flit Mode 310
8.2.5 Examples of pl_flit_cancel Timing Relationships 311
8.2.6 FDI State Machine 313
8.2.7 Rx_active_req/Sts Handshake 313
8.2.8 FDI Bring up flow 314
8.2.9 FDI PM Flow 315
8.3 Common rules for FDI and RDI 320
8.3.1 Byte Mapping for FDI and RDI 320
8.3.2 Stallreq/Ack Mechanism 321
8.3.3 State Request and Status 322
8.3.3.1 Reset State rules 323
8.3.3.2 Active State rules 324
8.3.3.3 PM Entry/Exit rules 324

 Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3.3.4 Retrain State rules 324
8.3.3.5 LinkReset State rules 325
8.3.3.6 Disabled State rules 326
8.3.3.7 LinkError State rules 326
8.3.3.8 Common State rules 327
8.3.4 Example flow diagrams 328
8.3.4.1 LinkReset Entry and Exit 328
8.3.4.2 LinkError 329

9.0 Compliance 330
9.1 Protocol Layer Compliance 331
9.2 Adapter Compliance 331
9.3 PHY Compliance 332

A CXL/PCIe Register applicability to UCIe 333
A.1 CXL Registers applicability to UCIe 333
A.2 PCIe Register applicability to UCIe 333
A.3 PCIe/CXL registers that need to be part of D2D 335

B AIB Interoperability 336
B.1 AIB Signal Mapping 336
B.1.1 Data path 336
B.1.2 Always high Valid 336
B.1.3 Sideband 336
B.1.4 Raw Die-to-Die interface 337
B.2 Initialization 338
B.3 Bump Map 338

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figures

1-1 A Package Composed of CPU Dies, Accelerator Die(s), and I/O Tile Die Connected through UCIe ... 26
1-2 UCIe enabling long-reach connectivity at Rack/Pod Level ... 27
1-3 Standard Package interface ... 27
1-4 Advanced Package interface: Example 1 ... 28
1-5 Advanced Package interface: Example 2 ... 28
1-6 Advanced Package interface: Example 3 ... 28
1-7 UCIe Layers and functionalities ... 29
1-8 Physical Layer components ... 30
1-9 Single module configuration: Advanced Package ... 31
1-10 Single module configuration: Standard Package ... 31
1-11 Two-module configuration for Standard Package ... 32
1-12 Four-module configuration for Standard Package ... 32
1-13 Example of a Two-module Configuration for Advanced Package ... 32
1-14 Block Diagram for UCIe Retimer Connection ... 33

2-1 Color-coding Convention in Flit Format Byte Map Figures ... 37
2-2 68B Flit Format on FDI ... 41

3-1 Functionalities in the Die-to-Die Adapter ... 43
3-2 Example configurations ... 45
3-3 Stages of UCIe Link initialization ... 46
3-4 Parameter exchange for Adapter Capabilities for PCIe Flit Mode ... 50
3-5 Parameter Exchange if "68B Flit Mode" or "CXL 256B Flit Mode" is 1b in {FinCap.Adapter} ... 50
3-6 Both Stacks are PCIe ... 52
3-7 Both Stacks are CXL ... 52
3-8 Stack 0 is PCIe, Stack 1 is Streaming ... 53
3-9 Stack 0 is Streaming, Stack 1 is PCIe ... 53
3-10 Both Stacks are Streaming ... 54
3-11 Stack0 is Streaming, Stack 1 is CXL ... 54
3-12 Format 1: Raw Format ... 55
3-13 Format 2: 68B Flit Format ... 58
3-14 Format 2: 68B Flit Format PDS Example 1 ... 58
3-15 Format 2: 68B Flit Format PDS Example 2 — Extra 0s Padded to Make the Data Transfer a Multiple of 256B ... 58
3-16 Format 3: Standard 256B End Header Flit Format for PCIe ... 60
3-17 Format 3: Standard 256B End Header Flit Format for Streaming Protocol ... 60
3-18 Format 4: Standard 256B Start Header Flit Format for CXL.cachemem or Streaming Protocol ... 61
3-19 Format 4: Standard 256B Start Header Flit Format for CXL.io or PCIe ... 61
3-20 Format 5: Latency-Optimized 256B without Optional Bytes Flit Format for CXL.io ... 63
3-21 Format 5: Latency-Optimized 256B without Optional Bytes Flit Format for CXL.cachemem and Streaming Protocol ... 63
3-22 Format 6: Latency-Optimized 256B with Optional Bytes Flit Format for CXL.io or PCIe ... 64
3-23 Format 6: Latency-Optimized 256B with Optional Bytes Flit Format for CXL.cachemem ... 64
3-24 Format 6: Latency-Optimized 256B with Optional Bytes Flit Format for Streaming Protocol ... 65
3-25 State Machine Hierarchy examples ... 70
3-26 Example of hierarchical PM entry for CXL ... 72
3-27 Diagram of CRC calculation ... 73

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3-28 Successful Parity Feature negotiation between Die 1 Tx and Die 0 Rx 76
3-29 Unsuccessful Parity Feature negotiation between Die 1 Tx and Die 0 Rx 77

4-1 Bit arrangement within a byte transfer 78
4-2 Byte map for x64 interface 79
4-3 Byte map for x32 interface 79
4-4 Byte map for x16 interface 79
4-5 Byte to Lane mapping for Standard package x8 degraded 80
4-6 Valid framing example 81
4-7 Clock gating 81
4-8 A 64 bit Sideband serial packet transfer example 82
4-9 Sideband packet transmission: back to back 82
4-10 Data Lane remapping possibilities to fix potential defects 85
4-11 Data Lane remapping: Mux chain 86
4-12 Single Lane failure remapping 88
4-13 Single Lane remapping implementation 88
4-14 Two Lane failure remapping 90
4-15 Two Lane remapping implementation 90
4-16 Clock and Track repair 94
4-17 Clock and track repair: Differential Rx 94
4-18 Clock and track repair: Pseudo Differential Rx 94
4-19 Clock and Track Lane repair scheme 95
4-20 Clock and track repair: CKP repair 96
4-21 Clock and track repair: CKN repair 96
4-22 Clock and track repair: Track repair 96
4-23 Valid repair: Normal Path 97
4-24 Valid Repair: Repair Path 97
4-25 Test and training logic 98
4-26 Lane failure detection 98
4-27 All Lane error detection 99
4-28 LFSR implementation 100
4-29 LFSR alternate implementation 101
4-30 Example Retimer bring up when performing speed/width match 106
4-31 Link Training State Machine 108
4-32 MBINIT: Mainband Initialization and Repair Flow 112
4-33 Mainband Training 123
4-34 Example of Byte Mapping for Matching Module IDs 138
4-35 Example of Byte Mapping for Differing Module IDs 138
4-36 Example of Width Degradation with Byte Mapping for Differing Module IDs 138
4-37 Example of Byte Mapping with Module Disable 139
4-38 Decision Flow Chart for Multi-module Advanced Package 140
4-39 Decision Flow Chart for Multi-module Standard Package 140
4-40 Implementation Example Showing Two Different Operating Modes of the Same Hardware Implementation 142
4-41 RDI Byte-to-Module Assignment Example for x64 Interop with x32 142

5-1 x64 and x32 Advanced Package Module 145
5-2 Standard Package Module 145
5-3 Transmitter 147
5-4 Transmitter driver example circuit 148
5-5 Transmitter de-emphasis 150
5-6 Transmitter de-emphasis waveform 150
5-7 Receiver topology 151
5-8 Receiver Termination Map for Table 5-5 (Tx Swing = 0.4 V) 153
5-9 Receiver termination 153
5-10 Receiver termination map for Table 5-6 (TX Swing = 0.85 V) 154

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5-11 Example CTLE 155
5-12 Clocking architecture 156
5-13 Track Usage Example 157
5-14 Example Eye diagram 159
5-15 Example Eye Simulation Setup 160
5-16 Circuit for VTF calculation 161
5-17 Loss and Crosstalk Mask 163
5-18 Viewer Orientation Looking at the Defined UCIe Bump Matrix 163
5-19 10-column x64 Advanced Package Bump Map 165
5-20 16-column x64 Advanced Package Bump Map 166
5-21 8-column x64 Advanced Package Bump Map 167
5-22 10-column x64 Advanced Package Bump map: Signal exit order 168
5-23 10-column x32 Advanced Package Bump Map 169
5-24 16-column x32 Advanced Package Bump Map 169
5-25 8-column x32 Advanced Package Bump Map 170
5-26 10-column x32 Advanced Package Bump Map: Signal Exit Order 170
5-27 Example of Normal and Mirrored x64-to-x32 Advanced Package Module Connection 173
5-28 Example of Normal and Mirrored x32-to-x32 Advanced Package Module Connection 174
5-29 Naming Convention for One-, Two-, and Four-module Advanced Package Paired with "Standard Die Rotate" Configurations 175
5-30 Naming Convention for One-, Two-, and Four-module Advanced Package Paired with "Mirrored Die Rotate" Configurations 175
5-31 Examples for Advanced Package Configurations Paired with "Standard Die Rotate" Counterparts, with a Different Number of Modules 176
5-32 Examples for Advanced Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules 177
5-33 Standard Package Bump Map: x16 interface 180
5-34 Standard Package x16 interface: Signal exit order 180
5-35 Standard Package Bump Map: x32 interface 181
5-36 Standard Package x32 interface: Signal exit routing 181
5-37 Standard Package cross section for stacked module 181
5-38 Standard Package reference configuration 182
5-39 Naming Convention for One-, Two-, and Four-module Standard Package Paired with "Standard Die Rotate" Configurations 183
5-40 Naming Convention for One-, Two-, and Four-module Standard Package Paired with "Mirrored Die Rotate" Configurations 183
5-41 Examples for Standard Package Configurations Paired with "Standard Die Rotate" Counterparts, with a Different Number of Modules 185
5-42 Examples for Standard Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules 186
5-43 Additional Examples for Standard Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules 186
5-44 Example of a Configuration for Standard Package, with Some Modules Disabled 188
5-45 Data Lane repair resources 190
5-46 Data Lane repair 190
5-47 Valid Framing 191
5-48 Data, Clock, Valid Levels for Half-rate Clocking: Clock-gated Unterminated Link 191
5-49 Data, Clock, Valid Levels for Quarter-rate Clocking: Clock-gated Unterminated Link 192
5-50 Data, Clock, Valid Levels for Half-rate Clocking: Continuous Clock Unterminated Link 192
5-51 Data, Clock, Valid Gated Levels for Half-rate Clocking: Terminated Link 192

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5-52 Data, Clock, Valid Gated Levels for Quarter-rate Clocking: Terminated Link 193
5-53 Data, Clock, Valid Gated Levels for Half-rate Clocking: Continuous Clock Terminated Link 193
5-54 Sideband signaling 193

6-1 Format for Register Access request 199
6-2 Format for Register Access completions 199
6-3 Format for Messages without Data 201
6-4 Format for Messages with data payloads 208
6-5 Example Flow for Remote Register Access Request (Local FDI/RDI Credit Checks Are Not Explicitly Shown) 217

7-1 Software view Example with Root Ports and Endpoints 223
7-2 Software view Example with Switch and Endpoints 224
7-3 Software view Example of UCIe Endpoint 225
7-4 UCIe Link DVSEC 227
7-5 UCIe Link Health Monitor (UHM) DVSEC 269
7-6 UCIe Test/Compliance Register Block 271

8-1 Example configurations using RDI 287
8-2 Example Waveform Showing Handling of Level Transition 293
8-3 Data Transfer from Adapter to Physical Layer 294
8-4 RDI State Machine 294
8-5 Example flow of Link bring up on RDI 295
8-6 Successful PM entry flow 298
8-7 PM Abort flow 299
8-8 PM Exit flow 299
8-9 RDI PM Exit Example Showing Interactions with LTSM 300
8-10 Example configurations using FDI 301
8-11 Example Waveform Showing Handling of Level Transition 309
8-12 Data Transfer from Protocol Layer to Adapter 310
8-13 Example for pl_flit_cancel for Latency-Optimized Flits and CRC Error on First Flit Half 311
8-14 Example for pl_flit_cancel for Latency-Optimized Flits and CRC Error on Second Flit Half 312
8-15 Example for pl_flit_cancel for Latency-Optimized Flits and CRC Error on Second Flit Half, Alternate Implementation Example 312
8-16 Example for pl_flit_cancel for Standard 256B Flits 312
8-17 FDI State Machine 313
8-18 FDI Bring up flow 315
8-19 PM Entry example for CXL or PCIe protocols 318
8-20 PM Entry example for symmetric protocol 318
8-21 PM Abort Example 319
8-22 PM Exit Example 319
8-23 CXL.io Standard 256B Start Header Flit Format Example 320
8-24 FDI (or RDI) Byte Mapping for 64B Datapath 321
8-25 FDI (or RDI) Byte Mapping for 128B Datapath 321
8-26 LinkReset Example 328
8-27 LinkError example 329

9-1 Examples of Standard and Advanced Package setups for DUT and Golden Die Compliance Testing 331

B-1 AIB interoperability 336

 Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Tables

1 Terms and Definitions 18
2 Reference Documents 23
3 Revision History 24
1-1 Characteristics of UCIe on Standard Package 27
1-2 Characteristics of UCIe on Advanced Package 28
1-3 UCIe Key Performance Targets 34
1-4 Groups for different bump pitches 35
2-1 Specification to protocol mode requirements 37
3-1 Capabilities that Must Be Negotiated between Link Partners 47
3-2 Flit Header for Format 2 without Retry 56
3-3 Flit Header for Format 2 with Retry 56
3-4 Flit Header for Format 3 or Format 4 without Retry 61
3-5 Flit Header for Format 3 or Format 4 with Retry 62
3-6 Summary of Flit Formats 66
3-7 Protocol Mapping and Implementation Requirements 67
3-8 Truth Table for determining Protocol 68
3-9 Truth Table 1 69
3-10 Truth Table 2 69
4-1 Valid framing for Retimers 81
4-2 Lane ID: Advanced Package module 83
4-3 Lane ID: Standard Package Module 84
4-4 LFSR seed values 99
4-5 Data to Clock Training Parameters 103
4-6 State Definitions for Initialization 107
4-7 Per Lane ID pattern 119
4-8 Per Lane ID pattern examples 119
4-9 Standard Package Logical Lane map 122
4-10 Runtime Link Test Status Register based Retrain encoding 134
4-11 Retrain encoding 134
4-12 Retrain exit state resolution 134
4-13 Capability Register and Link Training Parameter Values for RDI Byte-to-Module Assignment Example for x64 Interop with x32 143
5-1 Electrical summary 146
5-2 Transmitter Electrical Parameters 148
5-3 Transmitter de-emphasis values 150
5-4 Receiver Electrical parameters 152
5-5 Maximum channel reach for unterminated Receiver (Tx Swing = 0.4 V) 153
5-6 Maximum Channel reach for unterminated Receiver (TX swing = 0.85V) 154
5-7 Forwarded clock frequency and phase 156
5-8 I/O Noise and Clock Skew 158
5-9 Eye requirements 159
5-10 Channel Characteristics 162
5-11 x64 Advanced Package Module Signal List 162
5-12 x64 and x32 Advanced Package Connectivity Matrix 172
5-13 Summary of Advanced Package Module Connection Combinations with Same Number of Modules on Both Sides 176
5-14 Summary of Advanced Package Module Connection Combinations with Different Number of Modules on Both Sides 177

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5-15 IL and Crosstalk for Standard Package: With Receiver Termination Enabled 178
5-16 IL and Crosstalk for Standard Package: No Rx Termination 178
5-17 Standard Package Module Signal List 178
5-18 Summary of Standard Package Module Connection Combinations with Same Number of Modules on Both Sides 184
5-19 Summary of Standard Package Module Connection Combinations with Different Number of Modules on Both Sides 187
5-20 Summary of Degraded Links when Standard Package Module-pairs Fail 188
5-21 Tightly coupled mode: Eye mask 189
5-22 Tightly coupled mode channel for Advanced Package 189
5-23 Raw BER requirements 191
5-24 Sideband Parameters summary 194

6-1 Opcode encodings mapped to Packet Types 196
6-2 FDI sideband: srcid and dstid encodings on FDI 196
6-3 RDI sideband: srcid and dstid encodings on RDI 197
6-4 UCIe Link sideband: srcid and dstid encodings for UCIe Link 197
6-5 Field descriptions for Register Access Requests 198
6-6 Mapping of Addr[23:0] for different requests 199
6-7 Field Descriptions for a Completion 200
6-8 Message Encodings for Messages without Data 201
6-9 Link Training State Machine related Message encodings for messages without data 204
6-10 Message encodings for Messages with Data 208
6-11 Link Training State Machine related encodings 210

7-1 Software view of Upstream and Downstream Device at UCIe interface 220
7-2 SW discovery of UCIe Links 221
7-3 Summary of location of various UCIe Link related registers 222
7-4 Register Attributes 226
7-5 UCIe Link DVSEC - PCI Express Extended Capability Header 228
7-6 UCIe Link DVSEC - Designated Vendor Specific Header 1, 2 228
7-7 UCIe Link DVSEC - Capability Descriptor 229
7-8 UCIe Link DVSEC - UCIe Link Capability 230
7-9 UCIe Link DVSEC - UCIe Link Control 231
7-10 UCIe Link DVSEC - UCIe Link Status 233
7-11 UCIe Link DVSEC - Link Event Notification Control 235
7-12 UCIe Link DVSEC - Error Notification Control 236
7-13 UCIe Link DVSEC - Register Locator 0, 1, 2, 3 Low 239
7-14 UCIe Link DVSEC - Register Locator 0, 1, 2, 3 High 240
7-15 UCIe Link DVSEC - Sideband Mailbox Index Low 240
7-16 UCIe Link DVSEC - Sideband Mailbox Index High 241
7-17 UCIe Link DVSEC - Sideband Mailbox Data Low 241
7-18 UCIe Link DVSEC - Sideband Mailbox Data High 241
7-19 UCIe Link DVSEC - Sideband Mailbox Control 242
7-20 UCIe Link DVSEC - Sideband Mailbox Status 242
7-21 UCIe Link DVSEC - Requester ID 242
7-22 UCIe Link DVSEC - Associated Port Numbers 243
7-23 UiSRB DVSEC - PCI Express Extended Capability Header 243
7-24 UiSRB DVSEC - Designated Vendor Specific Header 1, 2 244
7-25 UiSRB DVSEC - UiSRB Base Address 244
7-26 D2D/PHY Register Block - UCIe Register Block Header (Offset 0h) 245
7-27 Uncorrectable Error Status Register 245
7-28 Uncorrectable Error Mask Register 246
7-29 Uncorrectable Error Severity Register 247
7-30 Correctable Error Status Register 247
7-31 Correctable Error Mask Register 248

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7-32 Header Log 1 Register 248
7-33 Header Log 2 Register 249
7-34 Error and Link Testing Control Register 250
7-35 Runtime Link Testing Parity Log 0 Register 251
7-36 Runtime Link Testing Parity Log 1 Register 251
7-37 Runtime Link Testing Parity Log 2 Register 252
7-38 Runtime Link Testing Parity Log 3 Register 252
7-39 Advertised Adapter Capability Log Register 252
7-40 Finalized Adapter Capability Log Register 252
7-41 Advertised CXL Capability Log Register 253
7-42 Finalized CXL Capability Log Register 253
7-43 Advertised Multi-Protocol Capability Log Register 253
7-44 Finalized Multi-Protocol Capability Log Register 253
7-45 Advertised CXL Capability Log Register for Stack 1 254
7-46 Finalized CXL Capability Log Register for Stack 1 254
7-47 Physical Layer Capability Register 255
7-48 Physical Layer Control Register 256
7-49 Physical Layer Status Register 257
7-50 Phy Init and Debug Register 258
7-51 Training Setup 1 Register 259
7-52 Training Setup 2 Register 259
7-53 Training Setup 3 Register 260
7-54 Training Setup 4 Register 260
7-55 Current Lane Map Module 0 Register 261
7-56 Current Lane Map Module 1 Register 261
7-57 Current Lane Map Module 2 Register 261
7-58 Current Lane Map Module 3 Register 261
7-59 Error Log 0 Register 262
7-60 Error Log 1 Register 263
7-61 Runtime Link Test Control 264
7-62 Runtime Link Test Status Register 266
7-63 Mainband Data Repair Register 266
7-64 Clock, Track, Valid and Sideband Repair Register 268
7-65 UHM DVSEC - Designated Vendor Specific Header 1, 2 269
7-66 UHM Status 270
7-67 EML_Lnx_Mody 270
7-68 EMR_Lnx_Mody 270
7-69 UCIe Register Block Header (Offset 0h) 272
7-70 D2D Adapter Test/Compliance Register Block Offset (Offset 10h) 272
7-71 PHY Test/Compliance Register Block Offset (Offset 14h) 272
7-72 Adapter Compliance Control (Offset 20h from D2DOFF) 273
7-73 Flit Tx Injection Control (Offset 28h from D2DOFF) 274
7-74 Adapter Test Status 275
7-75 Link State Injection Control Stack 0 276
7-76 Link State Injection Control Stack 1 277
7-77 Retry Injection Control 277
7-78 Physical Layer Compliance Control 1 279
7-79 Physical Layer Compliance Control 2 281
7-80 Physical Layer Compliance Status 1 282
7-81 Physical Layer Compliance Status 2 282
7-82 Physical Layer Compliance Status 3 283
7-83 UEDT Header 285
7-84 UCIe Link Structure (UCLS) 285

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8-1 RDI signal list 288
8-2 FDI signal list 302
8-3 Requests Considered in Each State by Lower Layer 323
A-1 CXL Registers for UCIe devices 333
A-2 PCIe Registers for UCIe devices 334
B-1 AIB 2.0 Datapath mapping for Advanced Package 337
B-2 AIB 1.0 Datapath mapping for Advanced Package 337

 Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Terminology

Table 1. Terms and Definitions (Sheet 1 of 6)

Term	Definition
Ack	Acknowledge
ACPI	Advanced Configuration and Power Interface
Addr	Address
Advanced Package	This packaging technology is used for performance optimized applications and short reach interconnects
AFE	Analog Front End
ALMP	ARB/MUX Link Management Packet (as defined in *CXL Specification*)
APMW	Advanced Package Module Width
ARB/MUX	Arbiter/Multiplexer (as defined in *CXL Specification*)
b	Bit
B	Byte
BAR	Base Address Register
BDF	Bus Device Function
BEI	BAR Equivalent Indicator
BER	Bit Error Ratio
BFM	Bus Functional Model
bubble	Gap in data transfer and/or signal transitions. Measured in number of clock cycles.
CA	Completer Abort
CDM	Charged Device Model
clear cleared	If clear or reset is used and no value is provided for a bit, it is interpreted as 0b.
CLM	Current Lane Map
CMLS	Common Maximum Link Speed
CoWoS	Chip on Wafer on Substrate
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CXL	Compute eXpress Link
CXL 68B Flit Mode	This term is used to reference 68B Flit Mode related Protocol features defined in *CXL Specification*.
CXL 256B Flit Mode	This term is used to reference 256B Flit Mode related Protocol features defined in *CXL Specification*.
D2C	Data-to-Clock
D2D	Die-to-Die
DCC	Duty Cycle Correction
DDR	Double Data Rate Memory
DLLP	Data Link Layer Packet (as defined in *PCIe Base Specification*)
DP	Downstream Port
DSP	Downstream Switch Port (as defined in *CXL Specification*)
DVSEC	Designated Vendor-Specific Extended Capability (as defined in *PCIe Base Specification*)
E2E	End to end

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 1. Terms and Definitions (Sheet 2 of 6)

Term	Definition
EM	Eye Margin
EMIB	Embedded Multi-die Interconnect Bridge
EML	Eye Margin Left
EMR	Eye Margin Right
EMV	Eye Margin Valid
Endpoint EP	As defined in *PCIe Base Specification*.
eRCD	Exclusive Restricted CXL Device (as defined in *CXL Specification*)
eRCH	Exclusive Restricted CXL Host (as defined in *CXL Specification*)
FDI	Flit-Aware Die-to-Die Interface
FEC	Forward Error Correction
FEXT	Far-End CrossTalk
FH	Flit Header
FIR	Finite Impulse Response
Flit_Marker FM	Flit Marker (as defined in *PCIe Base Specification*)
FW	Firmware
FW-CLK	Forwarded Clock over the UCIe Link for mainband data Lanes
FIFO	First In, First Out
Flit	Link Layer unit of transfer (as defined in *CXL Specification*).
GB/s	GigaBytes per second
GHz	GigaHertz
GT/s	Giga Transfers/second
HMLS	Highest Maximum Link Speed of next-lower configuration.
HW	Hardware
I/O	Input/Output
IL	Insertion Loss
Lane	A pair of signals mapped to physical bumps, one for Transmission, and one for Reception. A xN UCIe Link is composed of N Lanes.
LCLK	Refers to the clock at which the Logical Physical Layer, Adapter and RDI/FDI interfaces are operating
LCRC	Link CRC
LFSR	Linear Feedback Shift Register
Link UCIe Link	A Link or UCIe Link refers to the set of two UCIe components and their interconnecting Lanes which forms a dual-simplex communications path between the two components.
LSM	Adapter Link State Machine
LTSM	Link Training State Machine
LTSSM	Link Training and Status State Machine (as defined in *PCIe Base Specification*)
LSB	Least Significant Bit
Mainband MB	Connection that constitutes the main data path of UCIe. Consists of a forwarded clock, a data valid pin, and N Lanes of data per module.
MCLS	Number of active Modules Current Link Speed
MHz	Mega Hertz

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 1. Terms and Definitions (Sheet 3 of 6)

Term	Definition
MMIO	Memory mapped Input/Output
MMPL	Multi-module PHY Logic
Module	UCIe main data path on the physical bumps is organized as a group of Lanes called a Module. For Standard Package, 16 Lanes constitute a single Module. For Advanced Package, 64 Lanes constitute a single Module.
MSB	Most Significant Bit
MT/s	Mega Transfers/second
Nak	Negatively acknowledge
NEXT	Near-End CrossTalk
NVMe	Non-Volatile Memory express
P2P	Peer to peer
PCIe PCI Express	Peripheral Component Interconnect Express (defined in *PCIe Base Specification*)
PCIe Flit Mode	This term is used to reference Flit Mode related Protocol features defined in *PCIe Base Specification*.
PCIe non-Flit Mode	This term is used to reference non-Flit Mode related Protocol features defined in *PCIe Base Specification*.
PDS	Pause of Data Stream
PHY	Physical Layer
PI	Phase Interpolator
PLL	Phase-Locked Loop
PM	Power Management states, used to refer to behavior and/or rules related to Power Management states (covers both L1 and L2).
RCD	Restricted CXL Device (as defined in *CXL Specification*)
RCH	Restricted CXL Host (as defined in *CXL Specification*)
RCiEP	Root Complex Integrated Endpoint
RCRB	Root Complex Register Block
RDI	Raw Die-to-Die Interface
RCKP_P RXCKP rxckp	Physical Lane for Clock Receiver Phase-1
RCKN_P RXCKN rxckn	Physical Lane for Clock Receiver Phase-2
RRDCK_P RXCKRD rxckRD	Physical Lane for redundant Clock/Track Receiver
RTRK_P RXTRK rxtrk	Physical Lane for Track Receiver
RD_P[N] RD_PN RXDATA[N] rxdataN	Nth Physical Lane for Data Receiver

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 1. Terms and Definitions (Sheet 4 of 6)

Term	Definition
RRD_P[N] RRD_PN RXDATARD[N] rxdataRD[N]	Nth Physical Lane for redundant Data Receiver
RVLD_P RXVLD rxvld	Physical Lane for Valid Receiver
RRDVLD_P RXVLDRD rxvldRD	Physical Lane for redundant Valid Receiver
RXDATASB rxdatasb	Physical Lane for sideband Data Receiver
RXCKSB rxcksb	Physical Lane for sideband Clock Receiver
RXDATASBRD rxdatasbRD	Physical Lane for redundant sideband Data Receiver
RXCKSBRD rxcksbRD	Physical Lane for redundant sideband Clock Receiver
remote Link partner	This term is used throughout this specification to denote the logic associated with the far side of the UCIe Link; to denote actions or messages sent or received by the Link partner of a UCIe die.
Replay Retry	Retry and Replay are used interchangeably to refer to the Link level reliability mechanisms.
reset	If reset or clear is used and no value is provided for a bit, it is interpreted as 0b.
RL	Register Locator
Root Complex	As defined in *PCIe Base Specification*.
Root Port RP	As defined in *PCIe Base Specification*.
{<SBMSG>}	Sideband message requests or responses are referred to by their names enclosed in curly brackets. See Chapter 6.0 for the mapping of sideband message names to relevant encodings. An asterisk in the <SBMSG> name is used to denote a group of messages with the same prefix or suffix in their name.
SC	Successful Completion
SERDES	Serializer/Deserializer
set	If set is used and no value is provided for a bit, it is interpreted as 1b.
Sideband SB	Connection used for parameter exchanges, register accesses for debug/compliance and coordination with remote partner for Link training and management. Consists of a forwarded clock pin and a data pin in each direction. The clock is fixed at 800 MHz regardless of the main data path speed. The sideband logic for the UCIe Physical Layer must be on auxiliary power and an "always on" domain. Each module has its own set of sideband pins.
SM	State Machine
SoC	System on a Chip
Standard Package	This packaging technology is used for low cost and long reach interconnects using traces on organic package/substrate
Strobe	Used interchangeably with clock for sideband clock
SW	Software
TCM	Tightly coupled mode

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 1. Terms and Definitions (Sheet 5 of 6)

Term	Definition
TCKP_P TXCKP txckp	Physical Lane for Clock Transmitter Phase-1
TCKN_P TXCKN txckn	Physical Lane for Clock Transmitter Phase-2
TRDCK_P TXCKRD txckRD	Physical Lane for redundant Clock/Track Transmitter
TTRK_P TXTRK txtrk	Physical Lane for Track Transmitter
TD_P[N] TD_PN TXDATA[N] txdataN	Nth Physical Lane for Data Transmitter
TRD_P[N] TRD_PN TXDATARD[N] txdataRD[N]	Nth Physical Lane for redundant Data Transmitter
TVLD_P TXVLD txvld	Physical Lane for Valid Transmitter
TRDVLD_P TXVLDRD txvldRD	Physical Lane for redundant Valid Transmitter
TXDATASB txdatasb	Physical Lane for sideband Data Transmitter
TXCKSB txcksb	Physical Lane for sideband Clock Transmitter
TXDATASBRD txdatasbRD	Physical Lane for redundant sideband Data Transmitter
TXCKSBRD txcksbRD	Physical Lane for redundant sideband Clock Transmitter
TXEQ	TX Equalization
UCIe	Universal Chiplet Interconnect express
UCIe-A	Used to denote x64 Advanced Package module
UCIe-A x32	Used to denote x32 Advanced Package module. See Chapter 5.0 for UCIe-A x32 Advanced Package bump matrices, and interoperability between x32 to x32 and x32 to x64 module configurations.
UCIe-S	Used to denote x16 Standard Package module.
UCIe die	This term is used throughout this specification to denote the logic associated with the UCIe Link on any given chiplet with a UCIe Link connection. It is used as a common noun to denote actions or messages sent or received by an implementation of UCIe.
UCIe Flit Mode	Operating Mode in which CRC bytes are inserted and checked by the D2D Adapter. If applicable, Retry is also performed by the D2D Adapter.
UCIe Link	A UCIe connection between two chiplets. These chiplets are Link partners in the context of UCIe since they communicate with each other using a common UCIe Link.
UCLS	UCIe Link Structure

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 1. Terms and Definitions (Sheet 6 of 6)

Term	Definition
UCIe Raw Format	Operating format in which all the bytes of a Flit are populated by the Protocol Layer.
UHM	UCIe Link Health Monitor
UIE	Uncorrectable Internal Error
UiRB	UCIe Register Block
UiSRB	UCIe Structure Register Block
Unit Interval UI	Given a data stream of a repeating pattern of alternating 1 and 0 values, the Unit Interval is the value measured by averaging the time interval between voltage transitions, over a time interval sufficiently long to make all intentional frequency modulation of the source clock negligible.
UP	Upstream Port
UR	Unsupported Request
USP	Upstream Switch Port
VH	Virtual Hierarchy (as defined in *CXL Specification*)
vLSM	Virtual Link State Machine
Vref	Reference voltage for receivers
VTF	Voltage Transfer Function
zero	Numerical value of 0 in a bit, field, or register, of appropriate width for that bit, field, or register.

Reference Documents

Table 2. Reference Documents[1]

Document	Document Location
PCI Express® Base Specification (PCIe Base Specification) *Revision 6.0.1*	www.pcisig.com
Compute Express Link Specification (CXL Specification) *Revision 3.1*	computeexpresslink.org
ACPI Specification (version 6.5 or later)	www.uefi.org

1. References to these documents throughout this specification relate to the versions/revisions listed here.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Revision History

Table 3 lists the significant changes in different revisions.

Table 3. Revision History

Revision	Date	Description
1.1	July 10, 2023	• Streaming Flit Format Capabilities (Allows Streaming protocols to use Adapter Retry/CRC) • Enhanced Multi-Protocol Multiplexing (Allows dynamic multiplexing of different protocols on the same Adapter) • Support for x32 Advanced Package Modules • Support for UCIe Link Health Monitoring • Definition of Hardware capabilities to enable Compliance • di/dt risk mitigation during clock gating • Incorporation of Errata and bug fixes over 1.0
1.0	February 17, 2022	Initial release.

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

1.0 Introduction

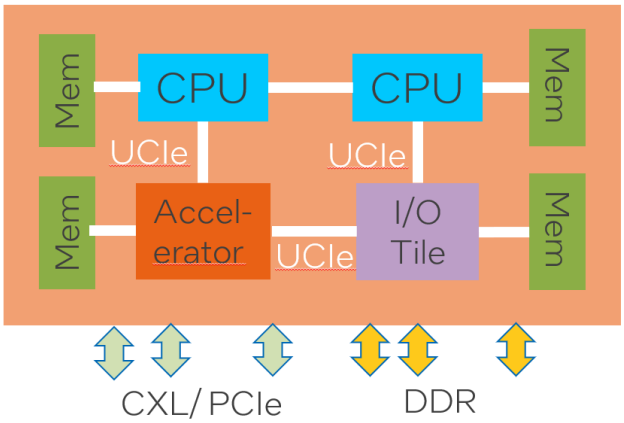
This chapter provides an overview of the Universal Chiplet Interconnect express (UCIe) architecture. UCIe is an open, multi-protocol capable, on-package interconnect standard for connecting multiple dies on the same package. The primary motivation is to enable a vibrant ecosystem supporting disaggregated die architectures which can be interconnected using UCIe. UCIe supports multiple protocols (PCIe, CXL, Streaming, and a raw format that can be used to map any protocol of choice as long as both ends support it) on top of a common physical and Link layer. It encompasses the elements needed for SoC construction such as the application layer, as well as the form-factors relevant to the package (e.g., bump location, power delivery, thermal solution, etc.). The specification is defined to ensure interoperability across a wide range of devices having different performance characteristics. A well-defined debug and compliance mechanism is provided to ensure interoperability. It is expected that the specification will evolve in a backward compatible manner.

While UCIe supports a wide range of usage models, some are provided here as an illustration of the type of capability and innovation it can unleash in the compute industry. The initial protocols being mapped to UCIe are PCIe, CXL, and Streaming. The mappings for all protocols are done using a Flit Format, including the Raw Format. Both PCIe and CXL are widely used and these protocol mappings will enable more on-package integration by replacing the PCIe SERDES PHY and the PCIe/CXL LogPHY along with the Link level Retry with a UCIe Adapter and PHY to improve the power and performance characteristics. UCIe provisions for Streaming protocols to also leverage the Link Level Retry of the UCIe Adapter, and this can be used to provide reliable transport for protocols other than PCIe or CXL. UCIe also supports a Raw Format that is protocol-agnostic to enable other protocols to be mapped; while allowing usages such as integrating a standalone SERDES/transceiver tile (e.g., ethernet) on-package. When using Raw Format, the Protocol Layer is responsible for reliable transport across the UCIe Link.

Figure 1-1 demonstrates an SoC package composed of CPU Dies, accelerator Die(s) and I/O Tile Die(s) connected through UCIe. The accelerator or I/O Tile can use CXL transactions over UCIe when connected to a CPU - leveraging the I/O, coherency, and memory protocols of CXL. The I/O tile can provide the external CXL, PCIe, and DDR pins of the package. The accelerator can also use PCIe transactions over UCIe when connected to a CPU. The CPU to CPU connectivity on-package can also use the UCIe interconnect, running coherency protocols.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 1-1. A Package Composed of CPU Dies, Accelerator Die(s), and I/O Tile Die Connected through UCIe

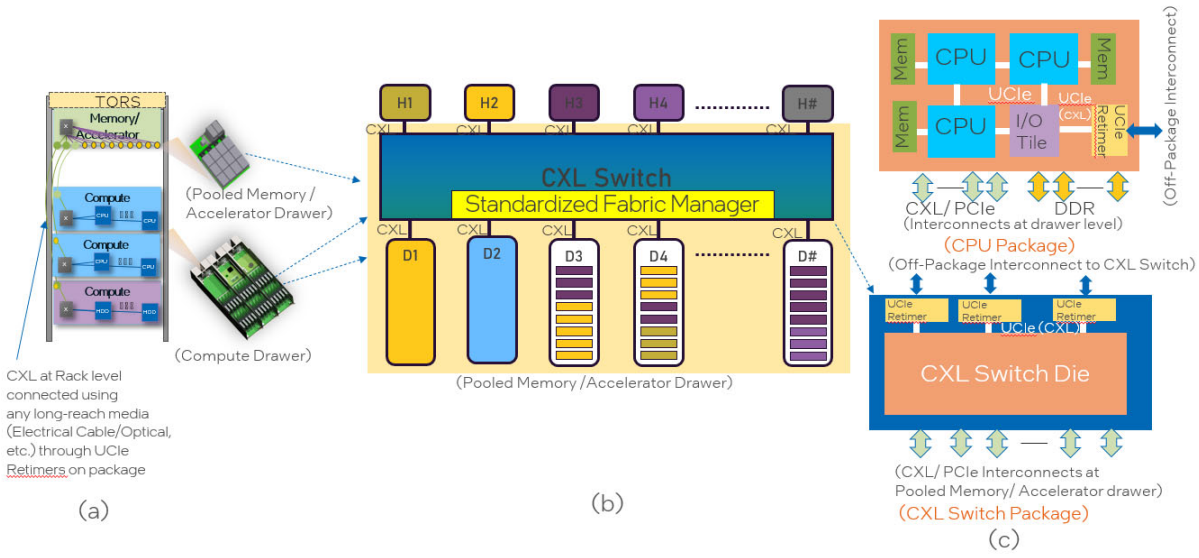


A UCIe Retimer may be used to extend the UCIe connectivity beyond the package using an Off-Package Interconnect. Examples of Off-Package Interconnect include electrical cable or optical cable or any other technology to connect packages at a Rack/Pod level as shown in Figure 1-2. The UCIe specification requires the UCIe Retimer to implement the UCIe interface to the Die that it connects on its local package and ensure that the Flits are delivered to the remote UCIe Die interface in the separate package following UCIe protocol using the channel extension technology of its choice.

Figure 1-2 demonstrates a rack/pod-level disaggregation using CXL protocol. Figure 1-2a shows the rack level view where multiple compute nodes (virtual hierarchy) from different compute chassis connect to a CXL switch which connects to multiple CXL accelerators/Type-3 memory devices which can be placed in one or more separate drawer. The logical view of this connectivity is shown in Figure 1-2b, where each “host” (H1, H2,…) is a compute drawer. Each compute drawer connects to the switch using an Off-Package Interconnect running CXL protocol through a UCIe Retimer, as shown in Figure 1-2c. The switch also has co-package Retimers where the Retimer tiles connect to the main switch die using UCIe and on the other side are the PCIe/CXL physical interconnects to connect to the accelerators/memory devices, as shown in Figure 1-2c.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 1-2. UCIe enabling long-reach connectivity at Rack/Pod Level



This version of UCIe Specification does not deal with 3D packaging. It allows two different packaging options: Standard Package (2D) and Advanced Package (2.5D). This covers the spectrum from lowest cost to best performance interconnects.

1. **Standard Package** - This packaging technology is used for low cost and long reach (10mm to 25mm, when measured from a bump on one Die to the connecting bump of the remote Die) interconnects using traces on organic package/substrate, while still providing significantly better BER characteristics compared to off-package SERDES. Figure 1-3 shows an example application using the Standard Package option. Table 1-1 shows a summary of the characteristics of the Standard Package option with UCIe.

Figure 1-3. Standard Package interface

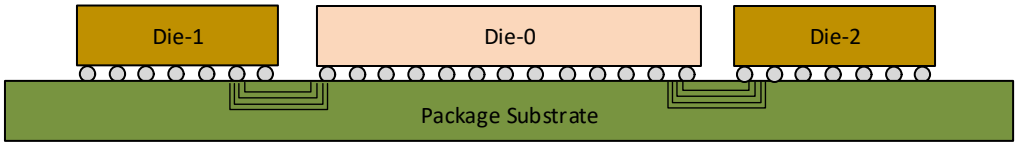


Table 1-1. Characteristics of UCIe on Standard Package

Index	Value
Supported speeds (per Lane)	4 GT/s, 8 GT/s, 12 GT/s, 16 GT/s, 24GT/s, 32 GT/s
Bump Pitch	100 um to 130 um
Channel reach (short reach)	10 mm
Channel reach (long reach)	25 mm
Raw Bit Error Rate (BER)[1]	1e-27 (<= 8 GT/s)
	1e-15 (>= 12 GT/s)

1. See Chapter 5.0 for details about BER characteristics.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2. **Advanced Package** - This packaging technology is used for performance optimized applications. Consequently, the channel reach is short (less than 2mm, when measured from a bump on one Die to the connecting bump of the remote Die) and the interconnect is expected to be optimized for high bandwidth and low latency with best performance and power efficiency characteristics. Figure 1-4, Figure 1-5, and Figure 1-6 show example applications using the Advanced Package option.

   Table 1-2 shows a summary of the main characteristics of the Advanced Package option.

Table 1-2. Characteristics of UCIe on Advanced Package

Index	Value
Supported speeds (per Lane)	4 GT/s, 8 GT/s, 12 GT/s, 16 GT/s, 24 GT/s, 32 GT/s
Bump pitch	25 um to 55 um
Channel reach	2 mm
Raw Bit Error Rate (BER)[1]	1e-27 (<=12GT/s)
	1e-15 (>=16GT/s)

1. See Chapter 5.0 for details about BER characteristics.

Figure 1-4. Advanced Package interface: Example 1

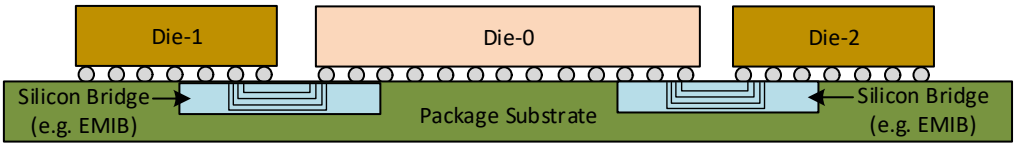


Figure 1-5. Advanced Package interface: Example 2

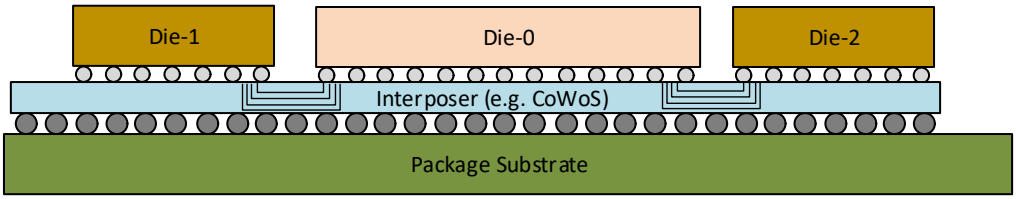
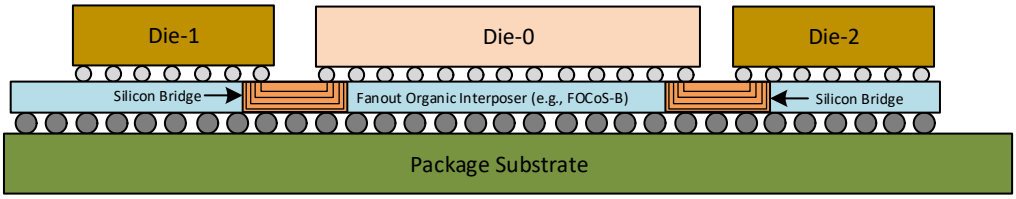


Figure 1-6. Advanced Package interface: Example 3

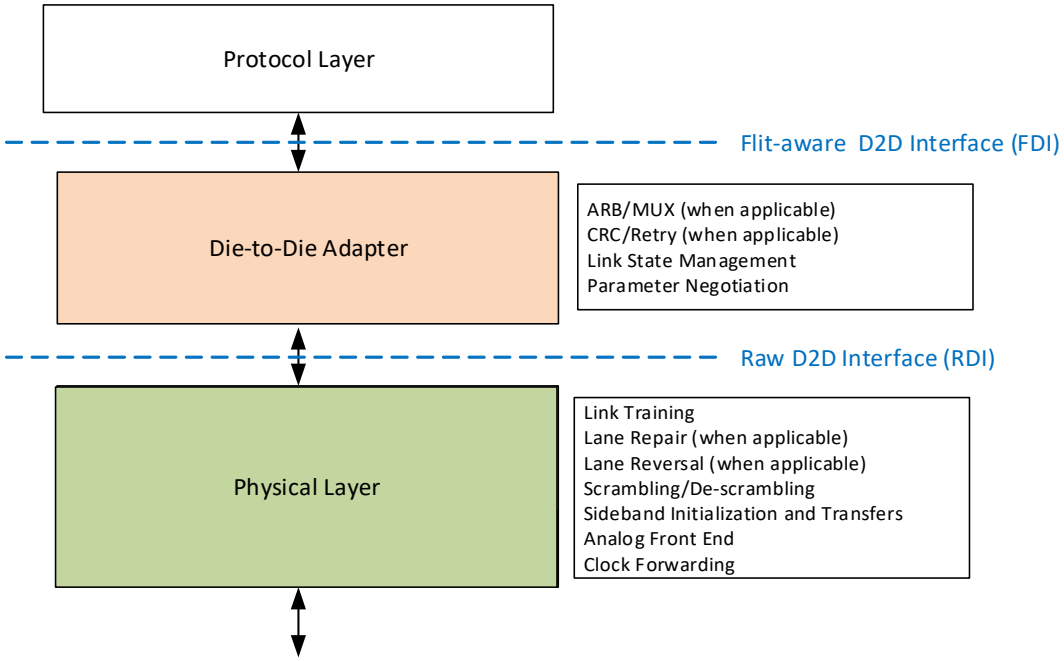


1.1 UCIe Components

UCIe is a layered protocol, with each layer performing a distinct set of functions. Figure 1-7 shows the three main components of the UCIe stack and the functionality partitioning between the layers. It is required for every component in the UCIe stack to be capable of supporting the advertised functionality and bandwidth. Several timeouts and related errors are defined for different handshakes and state transitions. All timeout values specified are minus 0% and plus 50% unless explicitly stated otherwise. All timeout values must be set to the specified values after Domain Reset. All counter values must be set to the specified values after Domain Reset.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 1-7. UCIe Layers and functionalities



1.1.1 Protocol Layer

While the Protocol Layer may be application specific, UCIe Specification provides examples of transferring CXL or PCIe protocols over UCIe Links. The following protocols are supported in UCIe for enabling different applications:

- PCIe from *PCIe Base Specification*.
- CXL from *CXL Specification*. Note that RCD/RCH/eRCD/eRCH are not supported.
- Streaming Protocol: This offers generic modes for a user defined protocol to be transmitted using UCIe.

For each protocol, different optimizations and associated Flit transfers are available for transfer over UCIe. Chapter 2.0 and Chapter 3.0 cover the relevant details of different modes and Flit Formats.

1.1.2 Die-to-Die (D2D) Adapter

The D2D Adapter coordinates with the Protocol Layer and the Physical Layer to ensure successful data transfer across the UCIe Link. It minimizes logic on the main data path as much as possible, thus providing a low-latency, optimized data path for protocol Flits. When transporting CXL protocol, the ARB/MUX functionality required for multiple simultaneous protocols is performed by the D2D Adapter. For options where the Raw BER is more than 1e-27, a CRC and Retry scheme is provided in the UCIe Specification for PCIe, CXL, or Streaming protocol, which is implemented in the D2D Adapter. See Section 3.8 for Retry rules.

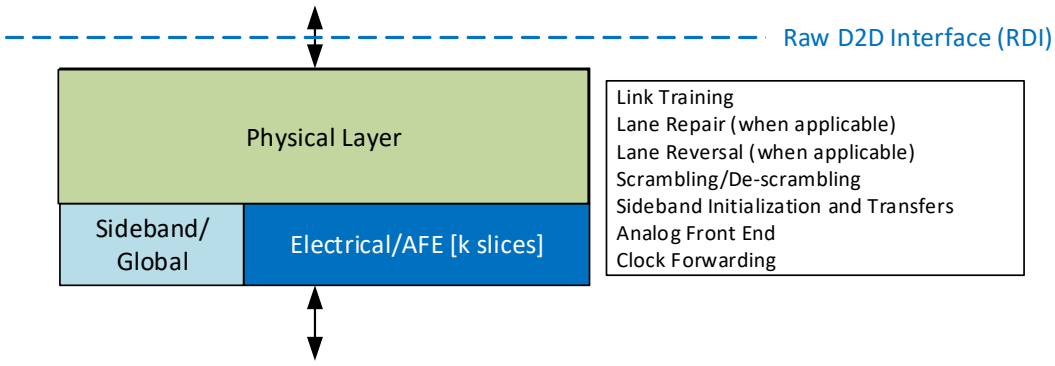
D2D Adapter is responsible for coordinating higher level Link state machine and bring up, protocol options related parameter exchanges with remote Link partner, and when supported, power management coordination with remote Link partner. Chapter 3.0 covers the relevant details for the D2D Adapter.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

1.1.3 Physical Layer

The Physical Layer has three sub-components as shown in Figure 1-8.

Figure 1-8. Physical Layer components



The UCIe main data path on the physical bumps is organized as a group of Lanes called a Module. A Module forms the atomic granularity for the structural design implementation of the UCIe AFE. The number of Lanes per Module for Standard and Advanced Packages is specified in Chapter 4.0. A given instance of Protocol Layer or D2D adapter can send data over multiple modules where bandwidth scaling is required.

The physical Link of UCIe is composed of two types of connections:

1. **Sideband**:
   This connection is used for parameter exchanges, register accesses for debug/compliance and coordination with remote partner for Link training and management. It consists of a forwarded clock pin and a data pin in each direction. The clock is fixed at 800MHz regardless of the mainband data rate. The sideband logic for the UCIe Physical Layer must be on auxiliary power and an "always on" domain. Each module has its own set of sideband pins.
   For the Advanced Package option, a redundant pair of clock and data pins in each direction is provided for repair.
2. **Mainband**:
   This connection constitutes the main data path of UCIe. It consists of a forwarded clock, a data valid pin, a track pin, and N Lanes of data per module.
   For the Advanced Package option, N=64 (also referred to as x64) or N=32 (also referred to as x32) and overall four extra pins for Lane repair are provided in the bump map.
   For the Standard Package option, N=16 (also referred to as x16) and no extra pins for repair are provided.

The Logical Physical Layer coordinates the different functions and their relative sequencing for proper Link bring up and management (for example, sideband transfers, mainband training and repair, etc.). Chapter 4.0 and Chapter 5.0 cover the details on Physical Layer operation.

1.1.4 Interfaces

UCIe defines the interfaces between the Physical Layer and the D2D Adapter (Raw D2D Interface), and the D2D Adapter and the Protocol Layer (Flit-aware D2D Interface) in Chapter 8.0.

The motivation for this is two-fold:

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Allow vendors and SoC builders to easily mix and match different layers from different IP providers at low integration cost and faster time to market. (For example, getting a Protocol Layer to work with the D2D Adapter and Physical Layer from any different vendor that conforms to the interface handshakes provided in the UCIe Specification.)
- Given that inter-op testing during post-silicon has greater overhead and cost associated with it, a consistent understanding and development of Bus Functional Models (BFMs) can allow easier IP development for this stack.

1.2 UCIe Configurations

This section describes the different configurations and permutations allowed for UCIe operation.

1.2.1 Single Module configuration

A single Module configuration is a x64 or x32 data interface in an Advanced Package, as shown in Figure 1-9. A single module configuration is a x16 data interface in a Standard Package, as shown in Figure 1-10. In multiple instantiations of a single module configuration where each module has its own dedicated Adapter, they operate independently (for example, they could be transferring data at different data rates and widths).

Figure 1-9. Single module configuration: Advanced Package

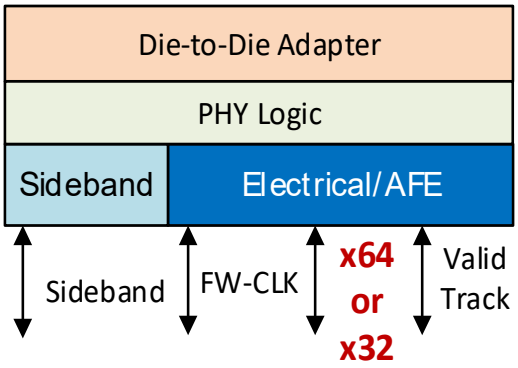
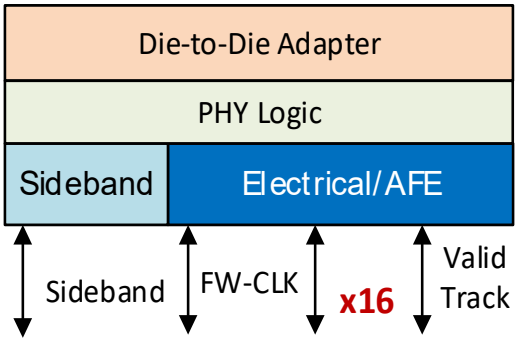


Figure 1-10. Single module configuration: Standard Package



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

1.2.2 Multi-module Configurations

This specification allows for two and four module configurations. When operating with a common Adapter, the modules in two-module and four-module configurations must operate at the same data rate and width. Examples of two-module and four-module configurations are shown in Figure 1-11 through Figure 1-13.

Figure 1-11. Two-module configuration for Standard Package

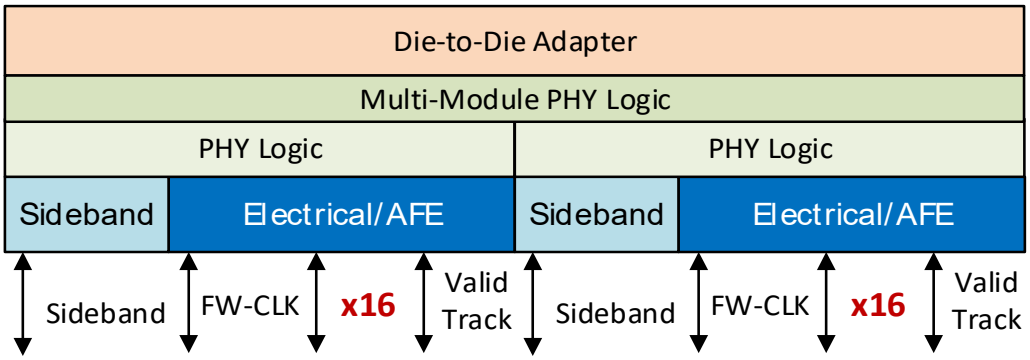


Figure 1-12. Four-module configuration for Standard Package

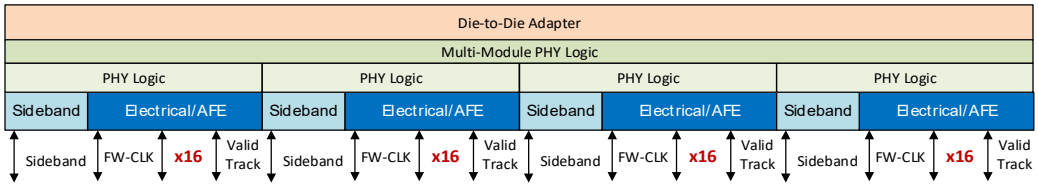
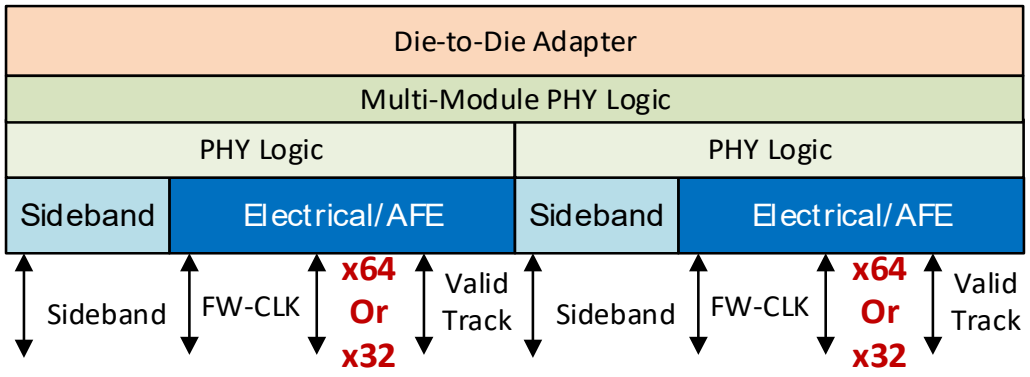


Figure 1-13. Example of a Two-module Configuration for Advanced Package



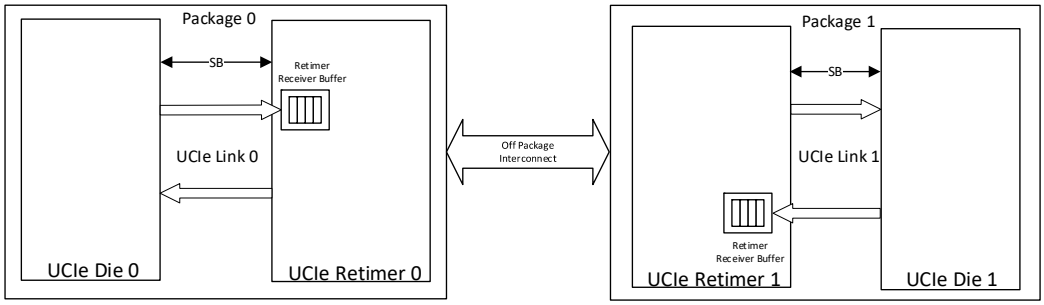
1.3 UCIe Retimers

As described previously, UCIe Retimers are used to enable different types of Off Package Interconnects to extend the channel reach between two UCIe Dies on different packages. Each UCIe Retimer has a local UCIe Link connection to a UCIe die on-package as well as an external connection for longer reach. Figure 1-14 shows a high level block diagram demonstrating a system utilizing UCIe Retimers to enable an Off Package Interconnect between UCIe Die 0 and UCIe Die 1. UCIe Retimer 0 and UCIe Die 0 are connected through UCIe Link 0 within Package 0. UCIe Retimer 1 and UCIe Die 1

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

are connected through UCIe Link 1 within Package 1. The terminology of "remote Retimer partner" is used to reference the UCIe Retimer die connected to the far side of the Off Package Interconnect.

Figure 1-14. Block Diagram for UCIe Retimer Connection



The responsibility of a UCIe Retimer include:

- Reliable Flit transfer across the Off Package Interconnect. Three options are available for achieving this as described below:
  - The Retimer is permitted to use the FEC and CRC natively defined by the underlying specification of the protocol it carries (e.g., PCIe or CXL) as long as the external interconnect conforms to the underlying error model (e.g., BER and error correlation) of the specification corresponding to the protocol it transports. The UCIe Links would be setup to utilize the Raw Format to tunnel native bits of the protocol it transports (e.g., PCIe or CXL Flits). In this scenario, the queue sizes (Protocol Layer buffers) must be adjusted on the UCIe Dies to meet the underlying round trip latency.
  - The Retimer is permitted to provide the necessary FEC, CRC and Retry capabilities to handle the BER of the Off Package Interconnect. In this case, the Flits undergo three independent Links; each UCIe Retimer performs an independent ACK/NAK for Retry with the UCIe die within its package and a separate independent ACK/NAK for Retry with the remote Retimer partner. For this scenario, protocols are permitted to use any of the applicable Flit Formats for transport over the UCIe Link.
  - The Retimer provides its own FEC by replacing the native PCIe or CXL defined FEC with its own, or adding its FEC in addition to the native PCIe or CXL defined FEC, but takes advantage of the built in CRC and Replay mechanisms of the underlying protocol. In this scenario, the queue sizes (Protocol Layer buffers, Retry buffers) must be adjusted on the UCIe Dies to meet the underlying round trip latency.
- Resolution of Link and Protocol Parameters with remote Retimer partner to ensure interoperability between UCIe Dies end-to-end (E2E). For example, Retimers are permitted to force the same Link width, speed, protocol (including any relevant protocol specific parameters) and Flit Formats on both Package 0 and Package 1 in Figure 1-14. The specific mechanism of resolution including message transfer for parameter exchanges across the Off Package Interconnect is implementation specific for the Retimers and they must ensure a consistent operational mode taking into account their own capabilities along with the UCIe Die capabilities on both Package 0 and Package 1. However, for robustness of the UCIe Links to avoid unnecessary timeouts in case the external interconnect requires a longer time to Link up or resolution of parameters with remote Retimer partner, UCIe Specification defines a "Stall" response to the relevant sideband messages that can potentially get delayed. The Retimers must respond with the "Stall" response within the rules of UCIe Specification to avoid such unnecessary timeouts while waiting for, or negotiating with remote Retimer partner. It is the responsibility of the Retimer to ensure the UCIe Link is not stalled indefinitely.
- Resolution of Link States for Adapter Link State Machine (LSM) or the RDI states with remote Retimer partner to ensure correct E2E operation. See Chapter 3.0 for more details.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Flow control and back-pressure:
  - Data transmitted from a UCIe Die to a UCIe Retimer is flow-controlled using credits. These credits are on top of any underlying protocol credit mechanism (such as PH, PD credits in PCIe). These UCIe D2D credits must be for flow control across the two UCIe Retimers and any data transmitted to the UCIe Retimer must eventually be consumed by the remote UCIe die without any other dependency. Every UCIe Retimer must implement a Receiver Buffer for Flits that it receives from the UCIe die within its package. The Receiver buffer credits are advertised to the UCIe die during initial parameter exchanges for the D2D Adapter, and the UCIe die must not send any data to the UCIe Retimer if it does not have a credit for it. One credit corresponds to 256B of data (including any FEC, CRC, etc.). Credit returns are overloaded on the Valid framing (see Section 4.1.2). Credit counters at the UCIe Die are re-assigned to their initial advertised value whenever RDI states transition away from Active. UCIe Retimer must drain or dump (as applicable) the data in its receiver buffer before re-entering Active state.
  - Data transmitted from a UCIe Retimer to a UCIe die is not flow-controlled at the D2D adapter level. The UCIe Retimer may have its independent flow-control with the other UCIe Retimer if needed, which is beyond the scope of this specification.

1.4 UCIe Key Performance Targets

Table 1-3 gives a summary of the performance targets for UCIe.

Table 1-3. UCIe Key Performance Targets

Metric	Link Speed/ Voltage	Advanced Package (x64)	Standard Package
Die Edge Bandwidth Density[1] (GB/s per mm)	4 GT/s	165	28
	8 GT/s	329	56
	12 GT/s	494	84
	16 GT/s	658	112
	24 GT/s	988	168
	32 GT/s	1317	224
Energy Efficiency[2] (pJ/bit)	0.7 V (Supply Voltage)	0.5 (<=12 GT/s)	0.5 (4 GT/s)
		0.6 (>=16 GT/s)	1.0 (<=16 GT/s)
		-	1.25 (32 GT/s)
	0.5 V (Supply Voltage)	0.25 (<=12 GT/s)	0.5 (<=16 GT/s)
		0.3 (>=16 GT/s)	0.75 (32 GT/s)
Latency Target[3]		<=2ns	

1. Die Edge Bandwidth Density (as defined in Chapter 5.0) is with 45-um (Advanced Package) and 110-um (Standard Package) bump pitch. For a x32 Advanced Package module, the Die Edge Bandwidth Density is 50% of the corresponding value for x64.
2. Energy Efficiency (energy consumed per bit to traverse from FDI to bump and back to FDI) includes all the Adapter and Physical Layer-related circuitry including, but not limited to, Tx, Rx, PLL, Clock Distribution, etc. Channel reach and termination are discussed in Chapter 5.0.
3. Latency includes the latency of the Adapter and the Physical Layer (FDI to bump delay) on Tx and Rx. See Chapter 5.0 for details of Physical Layer latency. Latency target is based on 16 GT/s. Latency at other data rates may differ due to data rate-dependent aspects such as data accumulation and transfer time. Note that the latency target does not include the accumulation of bits required for processing; either within or across Flits.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

1.5 Interoperability

Package designers need to ensure that Dies that are connected on a package can inter-operate. This includes compatible package interconnect (e.g., Advanced vs. Standard), protocols, voltage levels, etc. It is strongly recommended that a Die adopts Transmitter voltage of less than 0.85 V so that the Die can inter-operate with a wide range of process nodes in the foreseeable future.

This specification comprehends interoperability across a wide range of bump pitch for Advanced Packaging options. It is expected that over time, the smaller bump pitches will be predominantly used. With smaller bump pitch, we expect designs will reduce the maximum advertised frequency (even though they can go to 32G) to optimize for area and to address the power delivery and thermal constraints of high bandwidth with reduced area. Table 1-4 summarizes these bump pitches across four groups. Interoperability is guaranteed within each group as well as across groups, based on the PHY dimension specified in Chapter 5.0. The performance targets provided in Table 1-3 are with the 45 um bump pitch, based on the technology widely deployed at the time of publication of UCIe 1.0 and UCIe 1.1 Specifications (2022 – 2023).

Table 1-4. Groups for different bump pitches

Bump Pitch (um)	Minimum Frequency (GT/s)	Expected Maximum Frequency (GT/s)
Group 1: 25 - 30	4	12
Group 2: 31 - 37	4	16
Group 3: 38 - 44	4	24
Group 4: 45 - 55	4	32

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2.0 Protocol Layer

Universal Chiplet Interconnect express (UCIe) maps PCIe and CXL, as well as any Streaming Protocol. Throughout the UCIe Specification, Protocol-related features are kept separate from Flit Formats and packetization. This is because UCIe provides different transport mechanisms that are not necessarily tied to protocol features (e.g., PCIe non-Flit mode packets are transported using CXL.io 68B Flit Format). Protocol features include the definitions of Transaction Layer and higher layers, as well as Link Layer features not related to Flit packing/Retry (e.g., Flow Control negotiations etc.).

The following terminology is used throughout this specification to identify Protocol-level features:

- PCIe Flit Mode: To reference Flit Mode-related Protocol features defined in *PCIe Base Specification*
- PCIe non-Flit Mode: To reference non-Flit Mode-related Protocol features defined in *PCIe Base Specification*
- CXL 68B Flit Mode: To reference 68B Flit Mode-related Protocol features defined in *CXL Specification*
- CXL 256B Flit Mode: To reference 256B Flit Mode-related Protocol features defined in *CXL Specification*

The following protocol mappings are supported over UCIe:

- PCIe Flit Mode
- CXL 68B Flit Mode, CXL 256B Flit Mode: If CXL is negotiated, each of CXL.io, CXL.cache, and CXL.mem protocols are negotiated independently.
- Streaming Protocol: This offers generic modes for a user defined protocol to be transmitted using UCIe.

Note: RCD/RCH/eRCD/eRCH are not supported. PCIe non-Flit Mode is supported using CXL.io 68B Flit Format as the transport mechanism.

The Protocol Layer requirements for interoperability are as follows:

- A Protocol Layer must support PCIe non-Flit mode if it is advertising the 68B Flit Mode parameter from Table 3-1.
- If a Protocol Layer supports CXL 256B Flit Mode, it must support PCIe Flit Mode and 68B Flit Mode as defined in *CXL Specification* for CXL.io protocol.
- A Protocol Layer advertising CXL is permitted to only support CXL 68B Flit Mode without supporting CXL 256B Flit Mode or PCIe Flit Mode

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

IMPLEMENTATION NOTE

Table 2-1 summarizes the mapping of the above rules from a specification version to a protocol mode:

Table 2-1. Specification to protocol mode requirements

Native Specification Supported[1]	PCIe Non-Flit Mode	CXL 68B Flit Mode	CXL 256B Flit Mode	PCIe Flit Mode
PCIe	Mandatory	N/A	N/A	Optional
CXL 2.0	Mandatory (for CXL.io)	Mandatory	N/A	N/A
CXL 3.0	Mandatory (for CXL.io)	Mandatory	Mandatory	Mandatory (for CXL.io)

1. The same table applies to derivative version numbers for the specifications.

The Die-to-Die (D2D) Adapter negotiates the protocol with the remote Link partner and communicates it to the Protocol Layer(s). For each protocol, UCIe supports multiple modes of operation (that must be negotiated with the remote Link partner depending on the advertised capabilities, Physical Layer Status as well as usage models). These modes have different Flit Formats and are defined to enable different trade-offs around efficiency, bandwidth and interoperability. The spectrum of supported protocols, advertised modes and Flit Formats must be determined at SoC integration time or during the Die-specific reset bring up flow. The Die-to-Die Adapter uses this information to negotiate the operational mode as a part of Link Training and informs the Protocol Layer over the Flit-aware Die-to-Die Interface (FDI). See Section 3.2 for parameter exchange rules in the Adapter.

The subsequent sections provide an overview of the different modes from the Protocol Layer's perspective, hence they cover the supported formats of operation as subsections per protocol. The Protocol Layer is responsible for transmitting data over FDI in accordance with the negotiated mode and Flit Format. The illustrations of the Flit Formats in this chapter show an example configuration of a 64B data path in the Protocol Layer mapped to a 64 Lane module of Advanced Package configuration on the Physical Link of UCIe. Certain Flit Formats have dedicated bit positions filled in by the Adapter, and details associated with these are illustrated separately in Chapter 3.0. For other Link widths, refer to the Byte to Lane mappings defined in Section 4.1.1. Figure 2-1 shows the legend for color-coding convention used when showing bytes within a Flit in the Flit Format examples in the UCIe Specification.

Figure 2-1. Color-coding Convention in Flit Format Byte Map Figures

Color Shading	Description
	Some bits populated by the Protocol Layer, some bits populated by the Adapter.
	All bits populated by Adapter.
	All bits populated by the Protocol Layer.

2.1 PCIe

UCIe supports the Flit Mode defined in *PCIe Base Specification*. See *PCIe Base Specification* for the protocol definition. UCIe supports the non-Flit Mode using the CXL.io 68B Flit Formats as the transport mechanism. There are five UCIe operating formats supported for PCIe, and these are defined in the subsections that follow.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2.1.1 Raw Format

This format is optional. All bytes are populated by the Protocol Layer. The intended usage is for UCIe Retimers transporting PCIe protocol. An example usage of this format is where a CPU and an I/O Device are in different Rack/chassis and connected through a UCIe Retimer using Off-Package Interconnect as shown in Figure 1-2. Retry, CRC and FEC (if applicable) are taken care of by the Protocol Layer when using Raw Format. It is strongly recommended for the UCIe Retimers to check and count errors using either the parity bits of the 6B FEC or the Flit Mode 8B CRC defined in *PCIe Base Specification* for this mode to help characterize the Off Package Interconnect (to characterize or debug the Link that is the dominant source of errors). See Section 3.3.1 as well.

2.1.2 Standard 256B End Header Flit Format

This format is mandatory when PCIe Flit Mode protocol is supported. It is the standard Flit Format defined in *PCIe Base Specification* for Flit Mode and the main motivation of supporting this Flit Format is to enable interoperability with vendors that only support the standard PCIe Flit Formats. The Protocol Layer must follow the Flit Formats for Flit transfer on FDI, driving 0 on the fields reserved for Die-to-Die Adapter. The PM and Link Management DLLPs are not used over UCIe. The other DLLPs (that are applicable for PCIe Flit Mode) and Flit Status definitions follow the same rules including packing as defined in *PCIe Base Specification*. It is strongly recommended for implementations to optimize out any 8b/10b, 128b/130b, and non-Flit Mode related CRC/Retry or framing logic from the Protocol Layer in order to obtain area and power efficient designs for UCIe applications. Portions of the DLP bytes must be driven by the Protocol Layer for Flit_Marker assignment; see Section 3.3.3 for details of the Flit Format.

2.1.3 68B Flit Format

This mode is mandatory when PCIe protocol or CXL protocol is supported. The transport mechanism for this is the same as CXL.io 68B Flit Formats. See Section 2.3.2 for the CXL.io DLLP rules that apply for Non-Flit Mode for PCIe as well. It is strongly recommended for implementations to optimize out any 8b/10b, 128b/130b and non-Flit Mode related CRC/Retry logic from the Protocol Layer in order to obtain area and power efficient designs for UCIe applications. To keep the framing rules consistent, Protocol Layer for PCIe non-Flit mode must still drive the LCRC bytes with a fixed value of 0, and the Receiver must ignore these bytes and never send any Ack or Nak DLLPs. Framing tokens are applied as defined for CXL.io 68B Flit Mode operation in *CXL Specification*. It is recommended for the transmitter to drive the sequence number, DLLP CRC, Frame CRC and Frame parity in STP to 0; the receiver must ignore these fields. Given that UCIe Adapter provides reliable Flit transport, framing errors, if detected by the Protocol Layer, are likely due to uncorrectable internal errors and it is permitted to treat them as such.

2.1.4 Standard 256B Start Header Flit Format

This is an optional format for PCIe Flit Mode, supported if Standard Start Header for PCIe protocol Capability is supported. The Protocol Layer must follow the Flit Formats for Flit transfer on FDI, driving 0 on the fields reserved for Die-to-Die Adapter. The PM and Link Management DLLPs are not used over UCIe. The other DLLPs (that are applicable for PCIe Flit Mode) and Flit Status definitions follow the same rules including packing as defined in *PCIe Base Specification*. It is strongly recommended for implementations to optimize out any 8b/10b, 128b/130b and non-Flit Mode related CRC/Retry or framing logic from the Protocol Layer in order to obtain area and power efficient designs for UCIe applications. Portions of the DLP bytes must be driven by the Protocol Layer for Flit_Marker assignment; see Section 3.3.3 for details of the Flit Format.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2.1.5 Latency-Optimized 256B with Optional Bytes Flit Format

This is an optional format for PCIe Flit Mode, supported if Latency-Optimized Flit with Optional Bytes for PCIe protocol capability is supported. It is the Latency-Optimized Flit with Optional Bytes Flit Format for PCIe, as defined in Section 3.3.4. The Protocol Layer must follow the Flit Formats for Flit transfer on FDI, driving 0 on the fields reserved for Die-to-Die Adapter. The PM and Link Management DLLPs are not used over UCIe. The other DLLPs (that are applicable for PCIe Flit Mode) and Flit Status definitions follow the same rules including packing as defined in *PCIe Base Specification*. It is strongly recommended for implementations to optimize out any 8b/10b, 128b/130b and non-Flit Mode related CRC/Retry or framing logic from the Protocol Layer in order to obtain area and power efficient designs for UCIe applications. Portions of the DLP bytes must be driven by the Protocol Layer for Flit_Marker assignment; see Section 3.3.4 for details of the Flit Format.

2.2 CXL 256B Flit Mode

See *CXL Specification* for details on the protocol layer messages and slot formats for "CXL 256B Flit Mode". There are four possible operational formats for this protocol mode (there are two formats in Section 2.2.3), defined in the subsections that follow. The light orange bytes are inserted by the Adapter (see Figure 2-1). In cases where these are shown as part of the main data path (e.g., in the Standard 256B Flit Format), the Protocol Layer must drive 0 on them on the Transmitter, and ignore them on the Receiver.

2.2.1 Raw Format

This format is optional. All bytes are populated by the Protocol Layer. The intended usage is for UCIe Retimers transporting CXL 256B Flit Mode protocol. An example usage of this format is where a CPU and an I/O Device are in different Rack/chassis and connected through a UCIe Retimer using Off-Package Interconnect. Retry, CRC and FEC are taken care of by the Protocol Layer. It is strongly recommended for the UCIe Retimers to check and count errors using either the parity bits of the 6B FEC or the Flit Mode 8B CRC or 6B CRC; depending on which Flit Format was enabled. This helps to characterize and debug the Off-Package Interconnect which is the dominant source of errors. For CXL.cachemem, Viral or poison containment (if applicable) must be handled within the Protocol Layer for this format. See Section 3.3.1 as well.

2.2.2 Latency-Optimized 256B Flit Formats

The support for this format is strongly recommended for "CXL 256B Flit Mode" over UCIe. Two Flit Formats are defined, which provide two independent operating points. These formats are derived from the Latency-Optimized Flits defined in *CXL Specification*. The only difference for the second Flit Format is that it gives higher Flit packing efficiency by providing Protocol Layer with extra bytes. For CXL.io this results in extra 4B of TLP information, and for CXL.cachemem it results in an extra 14B H-slot that can be packed in the Flit. This slot is ordered between Slots 7 and 8. It is included in both Groups B and C, similar to Slot 7. See *CXL Specification* for reference of the packing rules. Support for the first or second format is negotiated at the time of Link bring up. See Section 3.3.4 for the details for the Flit Formats.

The Latency-Optimized formats enable the Protocol Layer to consume the Flit at 128B boundary, reducing the accumulation latency significantly. When this format is negotiated, the Protocol Layer must follow this Flit Format for Flit transfer on FDI, driving 0 on the fields reserved for Die-to-Die Adapter.

The Ack, Nak, PM, and Link Management DLLPs are not used over UCIe for CXL.io for any of the 256B Flit Modes. The other DLLPs and Flit_Marker definitions follow the same rules as defined in *CXL Specification*. Portions of the DLP bytes must be driven by the Protocol Layer for Flit_Marker assignment; see Section 3.3.3 for details on how DLP bytes are driven.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

For CXL.cachemem for this mode, FDI provides an **lp_corrupt_crc** signal to help optimize for latency while guaranteeing Viral containment. See Chapter 8.0 for details of interface rules for Viral containment.

2.2.3 Standard 256B Start Header Flit Format

This format is mandatory when "CXL 256B Flit Mode" protocol is supported. It is the Standard 256B Flit Format defined in *CXL Specification* for 256B Flit Mode and the main motivation of supporting this Flit Format is to enable interoperability with vendors that only support the Standard 256B Flit Formats. The Protocol Layer must follow the Flit Formats for Flit transfer on FDI, driving 0 on the fields reserved for Die-to-Die Adapter. The Ack, Nak, PM, and Link Management DLLPs are not used over UCIe for CXL.io. The other DLLPs and Flit Status definitions follow the same rules and packing as defined in *CXL Specification*. Portions of the DLP bytes must be driven by the Protocol Layer for Flit_Marker assignment; see Section 3.3.3 for details of the Flit Formats and on how DLP bytes are driven.

For CXL.cachemem in this format, FDI provides an **lp_corrupt_crc** signal to help optimize for latency while guaranteeing Viral containment. See Section 8.2 for details of interface rules for Viral containment.

See Section 3.3.3 for details about this Flit Format.

2.3 CXL 68B Flit Mode

The *CXL Specification* provides details on the protocol layer messages and slot formats for CXL 68B Flit Mode. There are two operational formats possible for this protocol, and these are defined in the subsections that follow. The light orange bytes are inserted by the Adapter (see Figure 2-1).

2.3.1 Raw Format

This format is optional. All bytes are populated by the Protocol Layer. The intended usage is for UCIe Retimers transporting "CXL 68B Flit Mode" protocol. An example usage of this format is where a CPU and an I/O Device are in different Rack/chassis and connected through a UCIe Retimer using an Off-Package Interconnect. Retry and CRC are taken care of by the Protocol Layer. See Section 3.3.1 as well.

2.3.2 68B Flit Format

This format is mandatory when CXL 68B Flit Mode protocol is negotiated. This follows the corresponding 68B Flit Format defined in *CXL Specification* and the main motivation of supporting this Flit Format is to enable interoperability with vendors that only support the baseline CXL formats. The Protocol Layer presents 64B of the Flit (excluding the Protocol ID and CRC) on FDI (shown in Figure 2-2), and the Die-to-Die Adapter inserts a 2B Flit Header and 2B CRC and performs the byte shifting required to arrange the Flits in the format shown in Figure 3-13.

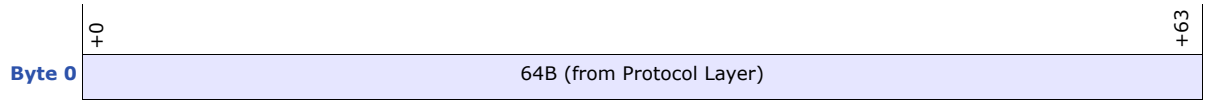
The Ack, Nak, and PM DLLPs are not used for CXL.io in this mode. Credit updates and other remaining DLLPs for CXL.io are transmitted in the Flits as defined in *CXL Specification*. For CXL.io, the Transmitter must not implement Retry in the Protocol Layer (because Retry is handled in the Adapter). To keep the framing rules consistent, Protocol Layer for CXL.io must still drive the LCRC bytes with a fixed value of 0, and the Receiver must ignore these bytes and never send any Ack or Nak DLLPs. Framing tokens are applied as defined for CXL.io 68B Flit Mode operation. It is recommended for the transmitter to drive the sequence number, DLLP CRC, Frame CRC and Frame parity in STP to 0. The receiver must ignore these fields. Given that UCIe Adapter provides reliable Flit

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

transport, framing errors, if detected by the Protocol Layer are likely due to uncorrectable internal errors and it is permitted to treat them as such.

For CXL.cachemem, the "Ak" field defined by *CXL Specification* in the Flit is reserved, and the Retry Flits are not used (because Retry is handled in the Adapter). Link Initialization begins with sending the INIT.Param Flit without waiting for any received Flits. Viral containment (if applicable) must be handled within the Protocol Layer for the 68B Flit Mode.

Figure 2-2. 68B Flit Format on FDI



2.4 Streaming protocol

This is the default protocol that must be advertised if none of the PCIe or CXL protocols are going to be advertised and negotiated with the remote Link partner. If Streaming Flit Format capability is not supported, then the operational formats that can be used are either Raw Format or vendor defined extensions. Streaming Flit Format capability is supported if any of 68B Flit Format for Streaming Protocol, Standard 256B End Header Flit Format for Streaming Protocol, Standard 256B Start Header Flit Format for Streaming Protocol, Latency-Optimized 256B Flit Format without Optional Bytes for Streaming Protocol or Latency-Optimized 256B Flit Format with Optional Bytes for Streaming Protocol bits are set in the UCIe Link Capability register.

2.4.1 Raw Format

This is mandatory for Streaming protocol support in Adapter implementations. Protocol Layer interoperability is vendor defined. All bytes are populated by the Protocol Layer. See Section 3.3.1 as well.

2.4.2 68B Flit Format

This format is only applicable if Streaming Flit Format capability is supported. It is an optional format that permits implementations to utilize the 68B Flit Format from the Adapter for Streaming protocols. See Section 3.3.2 for details of the Flit Format.

The Protocol Layer presents 64B per Flit on FDI, and the Die-to-Die Adapter inserts a 2B Flit Header and 2B CRC and performs the byte shifting required to arrange the Flits in the format shown in Figure 3-13. On the receive data path, the Adapter strips out the Flit Header and CRC bytes to only present the 64B per Flit to the Protocol Layer on FDI.

2.4.3 Standard 256B Flit Formats

This format is only applicable if Streaming Flit Format capability is supported. Implementations are permitted to utilize the Standard 256B Start Header Flit Format or Standard 256B End Header Flit Format from the Adapter for Streaming Protocols. See Section 3.3.3 for details of the Flit Format and to see which of the reserved fields in the Flit Header are driven by the Protocol Layer. The Protocol Layer presents 256B per Flit on FDI, driving 0b on the bits reserved for the Adapter. The Adapter fills in the applicable Flit Header and CRC bytes. On the Rx datapath, the Adapter forwards the Flit received from the Link as it is, and the Protocol Layer must ignore the bits reserved for the Adapter (for example the CRC bits).

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2.4.4 Latency-Optimized 256B Flit Formats

This format is applicable only when Streaming Flit Format capability is supported. Implementations are permitted to utilize the Latency-Optimized 256B with Optional Bytes Flit Format or Latency-Optimized 256B without Optional Bytes Flit Format for Streaming Protocols. See Section 3.3.4 for details of the Flit Format and to see which of the reserved fields in the Flit Header are driven by the Protocol Layer. The Protocol Layer presents 256B per Flit on FDI, driving 0b on the bits reserved for the Adapter. The Adapter fills in the applicable Flit Header and CRC bytes. On the Rx datapath, the Adapter forwards the Flit received from the Link as is, and the Protocol Layer must ignore the bits reserved for the Adapter (e.g., the CRC bits).

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

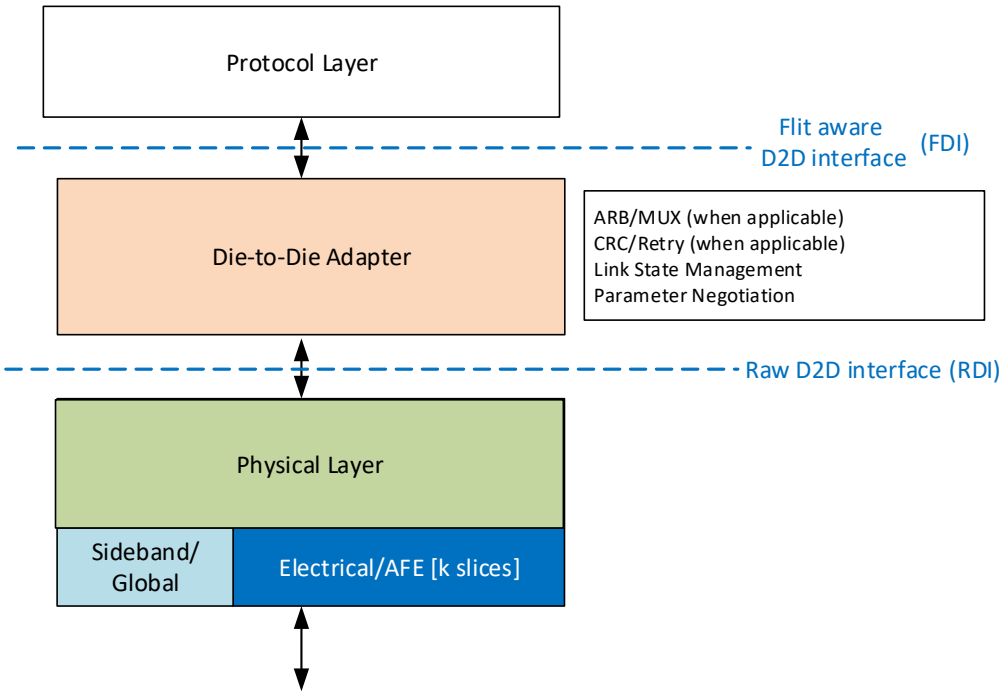
3.0 Die-to-Die Adapter

The Die-to-Die Adapter is responsible for:

- Reliable data transfer (performing CRC computation and Retry, or parity computation when applicable)
- Arbitration and Muxing (in case of multiple Protocol Layers)
- Link State Management
- Protocol and Parameter negotiation with the remote Link partner.

Figure 3-1 shows a high level description of the functionality of the Adapter.

Figure 3-1. Functionalities in the Die-to-Die Adapter



The Adapter interfaces to the Protocol Layer using one or more instances of the Flit-aware Die-to-Die interface (FDI), and it interfaces to the Physical Layer using the raw Die-to-Die interface (RDI). See Chapter 8.0 for interface details and operation.

The D2D Adapter must follow the same rules as the Protocol Layer for protocol interoperability requirements. Figure 3-2 shows example configurations for the Protocol Layer and the Adapter, where the Protocol identifiers (e.g., PCIe) only signify the protocol, and not the Flit Formats. To provide cost and efficiency trade-offs, UCIe allows configurations in which two protocol stacks are multiplexed onto the same physical Link.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.1 Stack Multiplexing

If Multi_Protocol_Enable parameter is negotiated, two stacks multiplexed on the same physical Link is supported when each protocol stack needs half the bandwidth that the Physical Layer provides. Both stacks must be of the same protocol with the same protocol capabilities. When Multi_Protocol_Enable is supported and negotiated, the Adapter must guarantee that it will not send consecutive flits from the same protocol stack on the Link. This applies in all cases including when Flits are sourced from FDI, from Retry Buffer, and when the data stream is paused and restarted. Adapter is permitted to insert NOP Flits to guarantee this (these Flits bypass the TX Retry buffer, and are not forwarded to the Protocol Layer on the receiver). When Flits are transmitted from the Retry Buffer, it is required to insert NOP Flits as needed to avoid sending consecutive Flits from the same Protocol stack. Note that there is no fixed pattern of Flits alternating from different Protocol Layers. For example, a Flit from Protocol Stack 0 followed by a NOP Flit, followed by a Flit from Protocol Stack 0 is a valid transmit pattern. A NOP Flit is defined as a Flit where the protocol identifier in the Flit Header corresponds to the D2D Adapter, and the body of the Flit is filled with all 0 data (The NOP Flit is defined for all Flit Formats supported by the Adapter, for all cases when it is operating in Raw Format). It is permitted for NOP flits to bypass the Retry buffer, as long as the Adapter guarantees that it is not sending consecutive Flits for any of the Protocol Layers. On the receiving side, the Adapter must not forward these NOP flits to the Protocol Layer. The receiving Protocol Layer must be capable of receiving consecutive chunks of the same Flit at the maximum Link speed, but it will not receive consecutive Flits. In addition to the transfer rate, both protocol stacks must operate with the same protocol and Flit Formats. Multi_Protocol_Enable and Raw Format are mutually exclusive. Each stack is given a single bit stack identifier that is carried along with the Flit header for de-multiplexing of Flits on the Receiver. The Stack Mux shown maintains independent Link state machines for each protocol stack. Link State transition-related sideband messages have unique message codes to identify which stack's Link State Management is affected by that message.

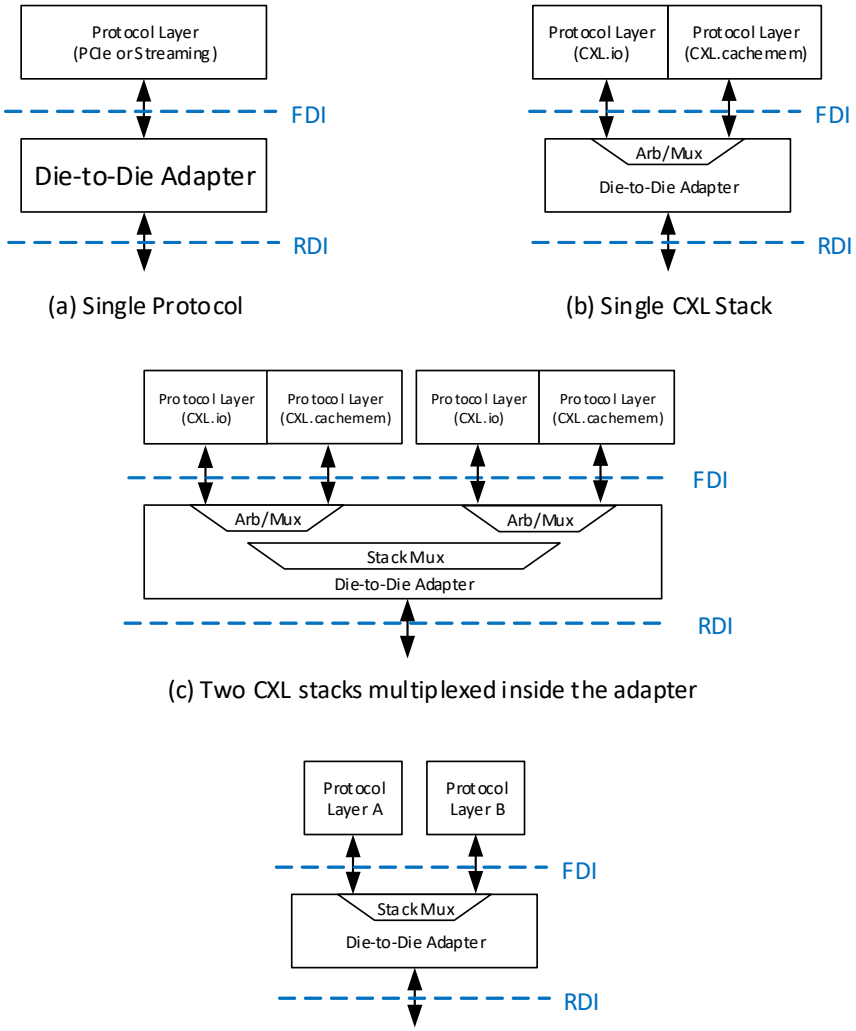
> **IMPLEMENTATION NOTE**
>
> The primary motivation for enabling the Multi_Protocol_Enable Parameter is to allow implementations to take advantage of the higher bandwidth provided by the UCIe Link for lower-bandwidth individual Protocol Layers, without the need to make a lot of changes to the UCIe Link. For example, two Protocol Layers that support the maximum bandwidth for CXL 68B Flit Mode (i.e., the equivalent of 32GT/s CXL SERDES bandwidth) can be multiplexed over a UCIe Link that supports their aggregate bandwidth.

If "Enhanced Multi_Protocol_Enable" is negotiated, dynamic multiplexing between two stacks of the same or different protocols on the same physical Link is supported. Both protocol stacks and the Adapter must support a common Flit Format for this feature to be enabled. The Adapter must advertise the maximum percentage of bandwidth that the receiver for each Protocol Layer can accept. The Adapter transmitter must support 100% (no throttling) and throttling one or both Protocol Layer(s) to 50% of maximum bandwidth. When 50% of the maximum bandwidth is advertised for a stack by an Adapter, the remote Link partner must guarantee that it will not send consecutive Flits for the same stack on the Link. This applies in all cases including when Flits are sourced from FDI, from Retry Buffer, and when the data stream is paused and restarted. Adapter is permitted to insert NOP Flits to guarantee this (these Flits bypass the TX Retry buffer, and are not forwarded to the Protocol Layer on the receiver). When Flits are transmitted from the Retry Buffer, it is required to insert NOP Flits as needed to avoid exceeding the negotiated maximum bandwidth. The receiving Protocol Layer must be capable of sinking Flits at the advertised maximum bandwidth percentage; in addition, Protocol Layer must be able to receive consecutive chunks of the same Flit at the maximum advertised Link speed. When this capability is supported, the Adapter must be capable of allowing each Protocol Layer to independently utilize 100% of the Link bandwidth. Furthermore, the arbitration is per Flit, and the Adapter must support round robin arbitration between the Protocol Layers if both

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

of them are permitted to use 100% of the Link bandwidth. Additional implementation specific arbitration schemes are permitted as long as they are per Flit and do not violate the maximum bandwidth percentage advertised by the remote Adapter for a given stack. The Flit header has a single bit stack identifier to identify the destination stack for the flit. The Stack Mux maintains independent Link state machines for each protocol stack. Link State transition-related sideband messages have unique message codes to identify which stack's Link State Management is affected by that message.

Figure 3-2. Example configurations

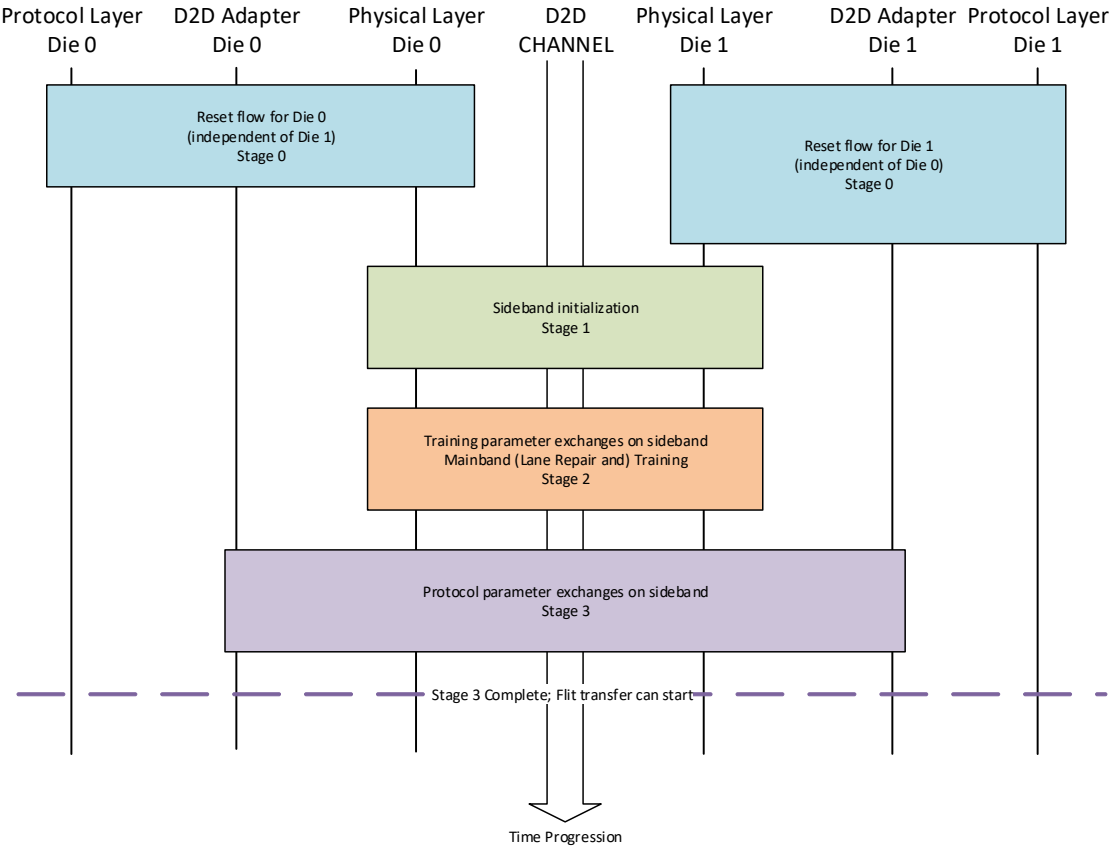


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.2 Link Initialization

Link Initialization consists of four stages before protocol Flit transfer can begin on mainband. Figure 3-3 shows the high-level steps involved in the Link initialization flow for UCIe. Stage 0 is die-specific and happens independently for each die; the corresponding boxes in Figure 3-3 are of different sizes to denote that different die can take different amount of time to finish Stage 0. Stage 1 involves sideband initialization. Stage 2 involves mainband training and repair. Details of Stage 1 and 2 are provided in Chapter 4.0. Stage 3 involves parameter exchanges between Adapters to negotiate the protocol and Flit Formats and is covered in Section 3.2.1.

Figure 3-3. Stages of UCIe Link initialization



3.2.1 Stage 3 of Link Initialization: Adapter Initialization

Stage 2 is complete when the RDI state machine moves to Active State. The initialization flow on RDI to transition the state from Reset to Active is described in Section 8.1.6. Once Stage 2 is complete, the Adapter must follow a sequence of steps in order to determine Local Capabilities, complete Parameter Exchanges, and bring FDI state machine to Active.

3.2.1.1 Part 1: Determine Local Capabilities

The Adapter must determine the results of Physical Layer training and if Retry is needed for the given Link speed and configuration. See Section 3.8 for the rules on when Retry must be enabled for Link operation. If the Adapter is capable of supporting Retry, it must advertise this capability to the remote Link partner during Parameter Exchanges. For UCIe Retimers, the Adapter must also determine the

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

credits to be advertised for the Retimer Receiver Buffer. Each credit corresponds to 256B of Mainband data storage.

3.2.1.2 Part 2: Parameter Exchange with remote Link partner

The following list of capabilities must be negotiated between Link partners. The capabilities (if enabled) are transmitted to the remote Link partner using a sideband message. In the section below, "advertised" means that the corresponding bit is 1b in the {AdvCap.Adapter} sideband message.

Table 3-1. Capabilities that Must Be Negotiated between Link Partners (Sheet 1 of 2)

Capability	Description and Requirements
"Raw Format"	This parameter is advertised if the corresponding bit in the UCIe Link Control register is 1b. Software/Firmware enables this based on system usage scenario. If the PCIe or CXL protocols are not supported, and Streaming Protocol is to be negotiated without any vendor-specific extensions and without Streaming Flit Format capability support, "Raw Format" must be 1b and advertised. If Streaming Flit Format capability or Enhanced Multi-Protocol capability is supported, then this must be advertised as 1b only if Raw Format is the intended format of operation. Software/firmware-based control on setting the corresponding UCIe Link Control is permitted to enable this.
"68B Flit Mode"	This is a protocol parameter. This must be advertised if the Adapter and Protocol Layer support CXL 68B Flit Mode or PCIe Non-Flit Mode. If PCIe Non-Flit Mode is the final negotiated protocol, it will use the CXL.io 68B Flit Mode formats as defined in *CXL Specification*. This is an advertised Protocol for Stack 0 if "Enhanced Multi_Protocol_Enable" is supported and enabled.
"CXL 256B Flit Mode"	This is a protocol parameter. This must be advertised if the Adapter and Protocol Layer support CXL 256B Flit Mode. This is an advertised Protocol for Stack 0 if Enhanced Multi-Protocol capability is supported and enabled.
"PCIe Flit Mode"	This is a protocol parameter. This must be advertised if the Adapter and Protocol Layer support PCIe Flit Mode. This is an advertised Protocol for Stack 0 if Enhanced Multi-Protocol capability is supported and enabled.
"Streaming"	This is a protocol parameter. This must be advertised if the Adapter and Protocol Layer support Streaming protocol in Raw Format or Streaming Flit Format capability is supported and the corresponding capabilities are enabled. This is an advertised Protocol for Stack 0 if Enhanced Multi-Protocol capability is supported and enabled. PCIe or CXL protocol must not be advertised if Streaming is advertised for a given Protocol Layer.
"Retry"	This must be advertised if the Adapter supports Retry. With the exception of the Link operating in Raw Format, the Link cannot be operational if the Adapter has determined Retry is needed, but "Retry" is not advertised or negotiated. See also Section 3.8.
"Multi_Protocol_Enable"	This must only be advertised if the Adapter is connected to multiple FDI instances corresponding to two sets of Protocol Layers. It must only be advertised if the Adapter (or SoC firmware in Stage 0 of Link Initialization) has determined that the UCIe Link must be operated in this mode. Both "Stack0_Enable" and "Stack1_Enable" must be 1b if this bit is advertised.
"Stack0_Enable"	This must be advertised if the Protocol Layer corresponding to Stack 0 exists and is enabled for operation with support for the advertised protocols.
"Stack1_Enable"	This must be advertised if the Protocol Layer corresponding to Stack 1 exists and is enabled for operation with support for the advertised protocols.
"CXL_LatOpt_Fmt5"	This must be advertised if the Adapter and Protocol Layer support *Format 5* defined in Section 3.3.4. The Protocol Layer does not take advantage of the spare bytes in this Flit Format. This must not be advertised if CXL protocol and CXL 256B Flit Mode are not supported or enabled.
"CXL_LatOpt_Fmt6"	This must be advertised if the Adapter and Protocol Layer support *Format 6* defined in Section 3.3.4. The Protocol Layer takes advantage of the spare bytes in this Flit Format. This must not be advertised if CXL protocol and CXL 256B Flit Mode are not supported or enabled.
"Retimer"	This must be advertised if the Adapter of a UCIe Retimer is performing Parameter Exchanges with a UCIe Die within its package.
"Retimer_Credits"	This is a 9-bit value advertising the total credits available for Retimer's Receiver Buffer. Each credit corresponds to 256B data. This parameter is applicable only when "Retimer" is 1b.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 3-1. Capabilities that Must Be Negotiated between Link Partners (Sheet 2 of 2)

Capability	Description and Requirements
"DP"	This is set by Downstream Ports to inform the remote Link partner that it is a Downstream Port. It is useful for Retimers to identify whether they are connected to a Downstream Port UCIe die. It is currently only applicable for PCIe and CXL protocols; however, Streaming Protocols are not precluded from utilizing this bit. If Enhanced Multi-Protocol capability is supported, this bit is applicable if either of the Protocol Layers is PCIe or CXL. This bit must be set to 0b if "Retimer" is set to 1b.
"UP"	This is set by Upstream Ports to inform the remote Link partner that it is an Upstream Port. It is useful for Retimers to identify whether they are connected to an Upstream Port UCIe die. It is currently only applicable for PCIe and CXL protocols; however, Streaming Protocols are not precluded from utilizing this bit. If Enhanced Multi-Protocol capability is supported, this bit is applicable if either of the Protocol Layers is PCIe or CXL. This bit must be set to 0b if "Retimer" is set to 1b.
"68B Flit Format"	This must be advertised if any of the following are true: • Enhanced Multi-Protocol capability is supported and enabled, AND the 68B Flit Format is supported and enabled • The 68B Flit Format for Streaming Protocol capability is supported and enabled Otherwise, it must be set to 0b.
"Standard 256B End Header Flit Format"	This must be advertised if any of the following are true: • Enhanced Multi-Protocol capability is supported and enabled, AND the Standard 256B End Header Flit Format is supported and enabled • The Standard 256B End Header Flit Format for Streaming Protocol capability is supported and enabled Otherwise, it must be set to 0b.
"Standard 256B Start Header Flit Format"	This must be advertised if any of the following are true: • PCIe Flit Mode is advertised and Standard Start Header for PCIe protocol capability is supported and enabled • Enhanced Multi-Protocol capability is supported and enabled, AND the Standard 256B Start Header Flit Format is supported and enabled • The Standard 256B Start Header Flit Format for Streaming Protocol capability is supported and enabled Otherwise, it must be set to 0b.
"Latency-Optimized 256B without Optional Bytes Flit Format"	This must be advertised if any of the following are true: • Enhanced Multi-Protocol capability is supported and enabled, AND the Latency-Optimized 256B without Optional Bytes Flit Format is supported and enabled • The Latency-Optimized 256B without Optional Bytes Flit Format for Streaming Protocol capability is supported and enabled Otherwise, it must be set to 0b.
"Latency-Optimized 256B with Optional Bytes Flit Format"	This must be advertised if any of the following are true: • PCIe Flit Mode is advertised and Latency-Optimized Flit with Optional Bytes for PCIe protocol capability is supported and enabled • Enhanced Multi-Protocol capability is supported and enabled, AND the Latency-Optimized 256B with Optional Bytes Flit Format is supported and enabled • The Latency-Optimized 256B with Optional Bytes Flit Format for Streaming Protocol capability is supported and enabled Otherwise, it must be set to 0b.
"Enhanced Multi_Protocol_Enable"	This must only be advertised if the Adapter is connected to multiple FDI instances corresponding to two sets of Protocol Layers. The two sets of Protocol Layers are permitted to be different protocols, but must support at least one common Flit Format. This must only be advertised if the Enhanced Multi-Protocol capability is supported and enabled; otherwise, it must be set to 0b. Both "Stack0_Enable" and "Stack1_Enable" must be 1b if this bit is advertised.
"Stack 0 Maximum Bandwidth_Limit"	This must be advertised if Enhanced Multi_Protocol_Enable is advertised and the Stack 0 protocol Receiver is limited to 50% of the maximum bandwidth; otherwise, it must be set to 0b.
"Stack 1 Maximum Bandwidth_Limit"	This must be advertised if Enhanced Multi_Protocol_Enable is advertised and the Stack 1 protocol Receiver is limited to 50% of the maximum bandwidth; otherwise, it must be set to 0b.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Once local capabilities are established, the Adapter sends the {AdvCap.Adapter} sideband message advertising its capabilities to the remote Link partner.

If PCIe or CXL protocol support is going to be advertised, the Upstream Port (UP) Adapter must wait for the first {AdvCap.Adapter} message from the Downstream Port (DP) Adapter, review the capabilities advertised by DP and then send its own sideband message of advertised capabilities. UP is permitted to change its advertised capabilities based on DP capabilities. Once the DP receives the capability advertisement message from the UP, the DP responds with the Finalized Configuration using {FinCap.Adapter} sideband message to the UP as shown in Figure 3-4. See Section 6.1.2.3 to see the message format for the relevant sideband messages.

Final determination for Protocol parameters:

- If "68B Flit Mode" is advertised by both Link partners, it is set to 1b in the {FinCap.Adapter} message
- If "CXL 256B Flit Mode" is advertised by both Link partners, it is set to 1b in the {FinCap.Adapter} message
- If "PCIe Flit Mode" is advertised by both Link partners, "PCIe Flit Mode" bit is set to 1b in the {FinCap.Adapter} message
- If Streaming Protocol is negotiated, no {FinCap.Adapter} messages are exchanged for that stack.

If "68B Flit Mode" or "CXL 256B Flit Mode" is set in the {FinCap.Adapter} message, there must be another handshake of Parameter Exchanges using the {AdvCap.CXL} and the {FinCap.CXL} messages to determine the details associated with this mode. This additional handshake is shown in Figure 3-5. The combination of {FinCap.CXL} and {FinCap.Adapter} determine the Protocol and Flit Format. See Section 6.1.2.3 for the message format of the relevant sideband messages. See Section 3.4 for how Protocol and Flit Formats are determined.

Final determination for other parameters if PCIe or CXL protocol is negotiated:

- If "Raw Format" is advertised by both Link partners, "Raw Format" is set to 1b in the {FinCap.Adapter} message.
- If both Link partners advertised "Retry" and "Raw Format" is not negotiated, Adapter Retry is enabled and "Retry" is set to 1b in the {FinCap.Adapter} message.
- If both Link partners advertised "Enhanced Multi_Protocol_Enable", both Stack0 and Stack1 are enabled by the adapter, and all three parameters ("Enhanced Multi_Protocol_Enable", "Stack0_Enable" and "Stack1_Enable") are set to 1b in the {FinCap.Adapter} message (if a {FinCap.Adapter} message is required to be sent).
- If both Link partners advertised "Multi_Protocol_Enable" and "Enhanced Multi_Protocol_Enable" is not negotiated, both Stack0 and Stack1 are enabled by the Adapter, and all three parameters ("Multi_Protocol_Enable", "Stack0_Enable", and "Stack1_Enable") are set to 1b in the {FinCap.Adapter} message.
- If "Enhanced Multi_Protocol_Enable" or "Multi_Protocol_Enable" is not negotiated, then the lowest common denominator is used to determine whether Stack0 or Stack1 is enabled, and the corresponding bit is set to 1b in the {FinCap.Adapter} message. If both Stack enables are advertised, then Stack0 is selected for operational mode and only Stack0_Enable is set to 1b in the {FinCap.Adapter} message.
- If CXL_LatOpt_Fmt5 is advertised by both Link partners, then it is set to 1b in the {FinCap.Adapter} message.
- If CXL_LatOpt_Fmt6 is advertised by both Link partners, then it is set to 1b in the {FinCap.Adapter} message.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-4. Parameter exchange for Adapter Capabilities for PCIe Flit Mode

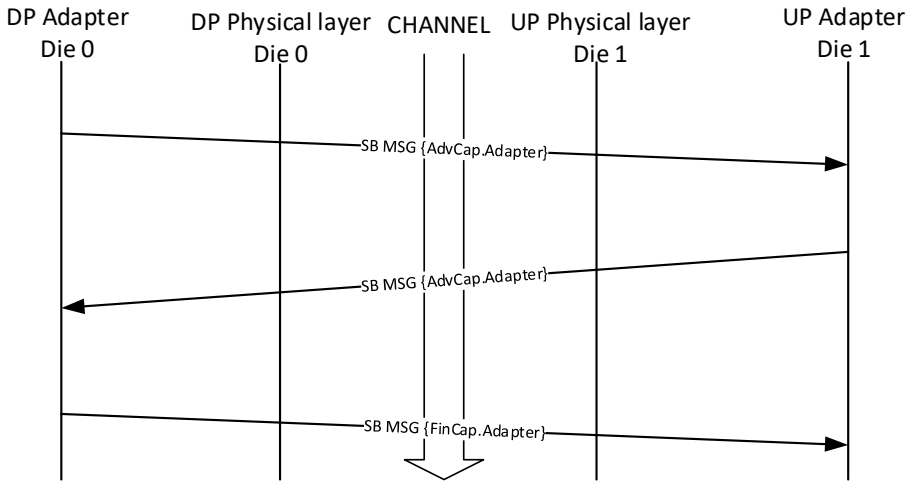
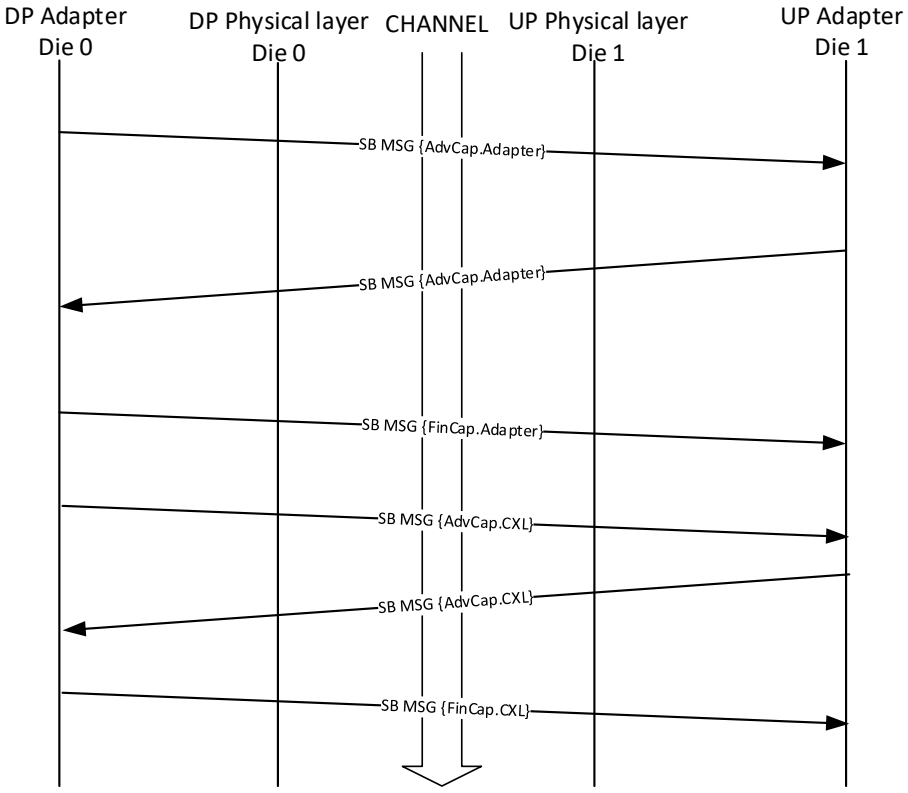


Figure 3-5. Parameter Exchange if "68B Flit Mode" or "CXL 256B Flit Mode" is 1b in {FinCap.Adapter}



If Streaming protocol is negotiated, there is no notion of DP and UP for parameter exchanges and each side independently advertises its capabilities. Additional Vendor Defined sideband messages are permitted to be exchanged to negotiate vendor-specific extensions. See Table 6-8 and Table 6-10 for additional descriptions of Vendor Defined sideband messages. The Finalized Configuration is implicitly determined based on the intersection of capabilities advertised by each side:

- Flit Formats are chosen based on the Truth Table resolution provided in Table 3-10

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- If both Link partners advertised Retry, and "Raw Format" is not negotiated, then Adapter Retry is enabled
- If "Multi_Protocol_Enable" is negotiated, both Stack 0 and Stack 1 are enabled by the adapter
- If "Multi_Protocol_Enable" or "Enhanced Multi_Protocol_Enable" is not advertised by at least one of the Link partners, then the lowest common denominator is used to determine whether Stack 0 or Stack 1 is enabled (i.e., if both Stack enables are advertised, then Stack 0 is selected for operational mode)

{FinCap.*} messages are not sent for Streaming Protocol. Adapter must determine vendor specific requirements in an implementation specific manner.

If "Enhanced Multi_Protocol_Enable" is negotiated, the {AdvCap.Adapter} and if applicable, the {FinCap.Adapter} messages determine the negotiated Flit Format of operation as well as the protocol for Stack 0. The Adapter uses {MultiProtAdvCap.Adapter} and if applicable, the {MultiProtFinCap.Adapter} sideband messages to negotiate the Stack 1 protocol. For Stack 1, if PCIe or CXL protocol support is going to be advertised, the UP Adapter must wait for the first message from the DP Adapter, review the capabilities advertised by DP and then send its own sideband message of advertised capabilities. UP is permitted to change its advertised capabilities based on DP capabilities. In the section below, "advertised" means that the corresponding bit is 1b in the {MultiProtAdvCap.Adapter} sideband message.

1. "68B Flit Mode": This must be advertised if the Adapter and Protocol Layer support CXL 68B Flit Mode or PCIe Non-Flit Mode on Stack 1.
2. "CXL 256B Flit Mode": This must be advertised if the Adapter and Protocol Layer support CXL 256B Flit Mode on Stack 1.
3. "PCIe Flit Mode": This must be advertised if the Adapter and Protocol Layer support PCIe Flit Mode on Stack 1.
4. "Streaming": This must be advertised if the Adapter and Protocol Layer support Streaming Flit Mode on Stack 1.

If "68B Flit Mode" or "CXL 256B Flit Mode" is set in the {MultiProtFinCap.Adapter} message, there must be another handshake of Parameter Exchanges using the {AdvCap.CXL} and the {FinCap.CXL} messages to determine the details associated with this mode. The non-Stall {*.CXL} messages are sent with a MsgInfo encoding of 0001h indicating that these messages are for Stack 1 negotiation.

Figure 3-6 to Figure 3-11 represent examples of different scenarios where Stack 0 and Stack 1 are of different protocols.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-6. Both Stacks are PCIe

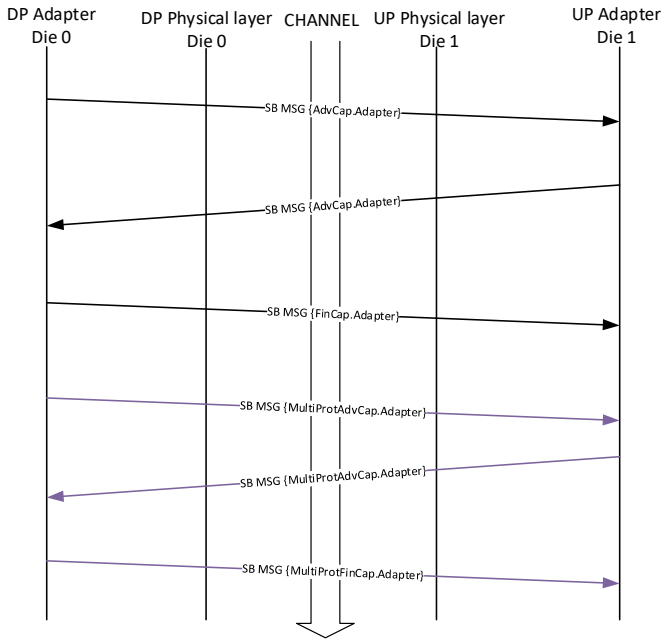
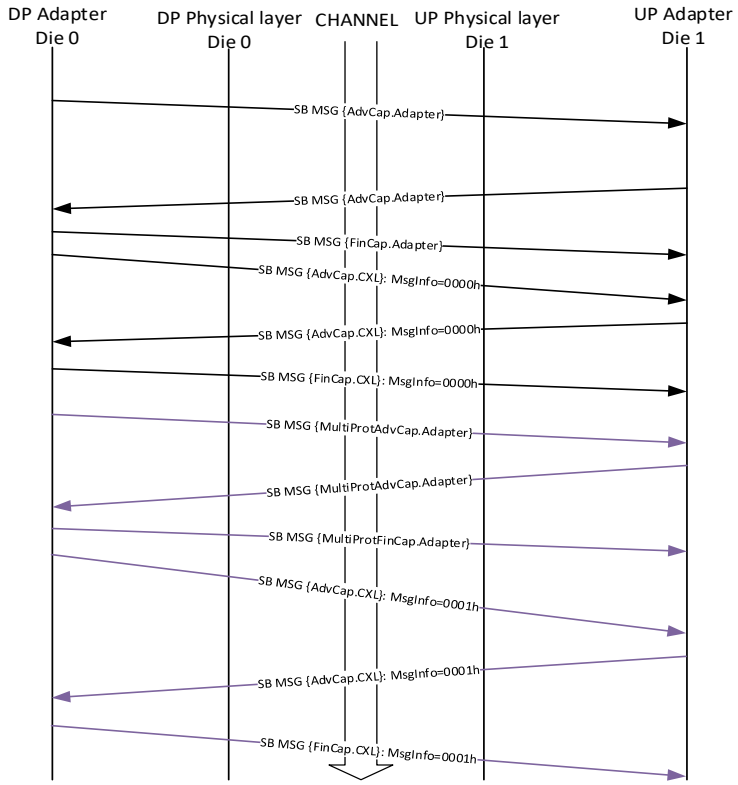


Figure 3-7. Both Stacks are CXL



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-8. Stack 0 is PCIe, Stack 1 is Streaming

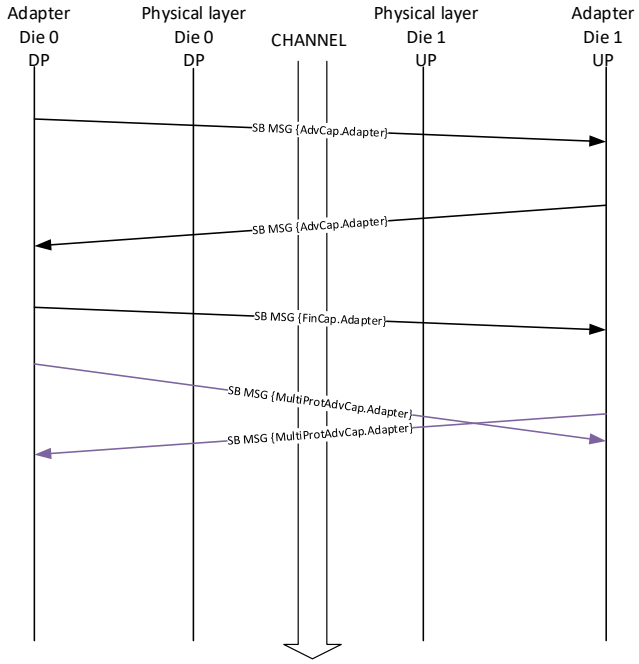
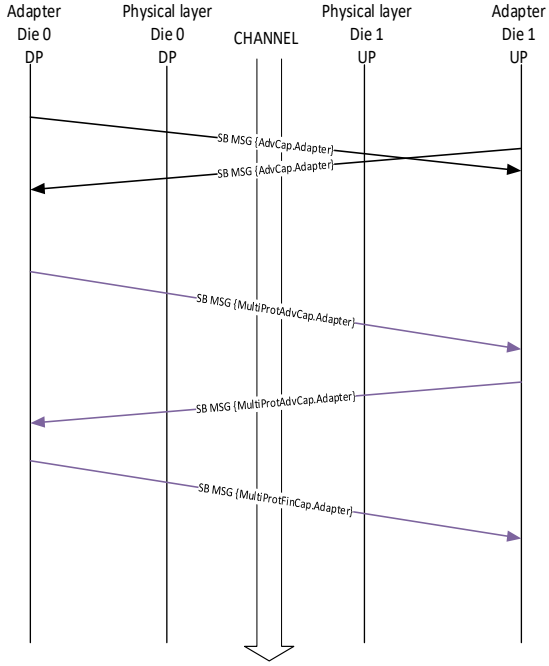


Figure 3-9. Stack 0 is Streaming, Stack 1 is PCIe



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-10. Both Stacks are Streaming

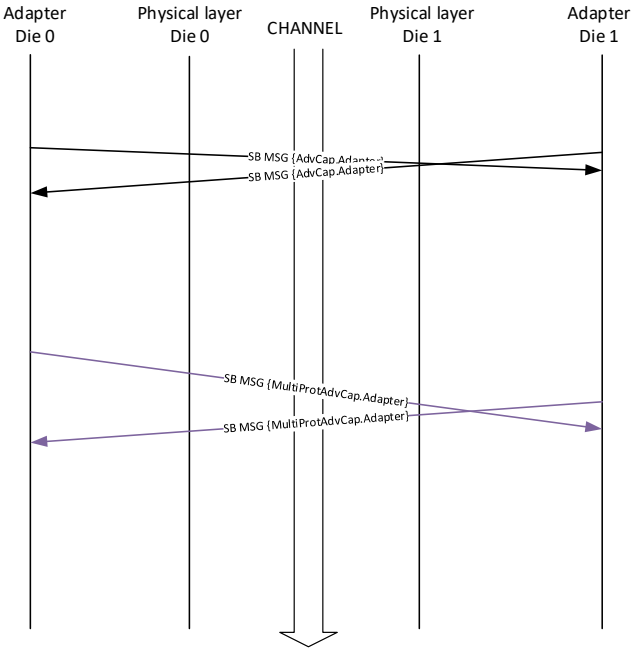
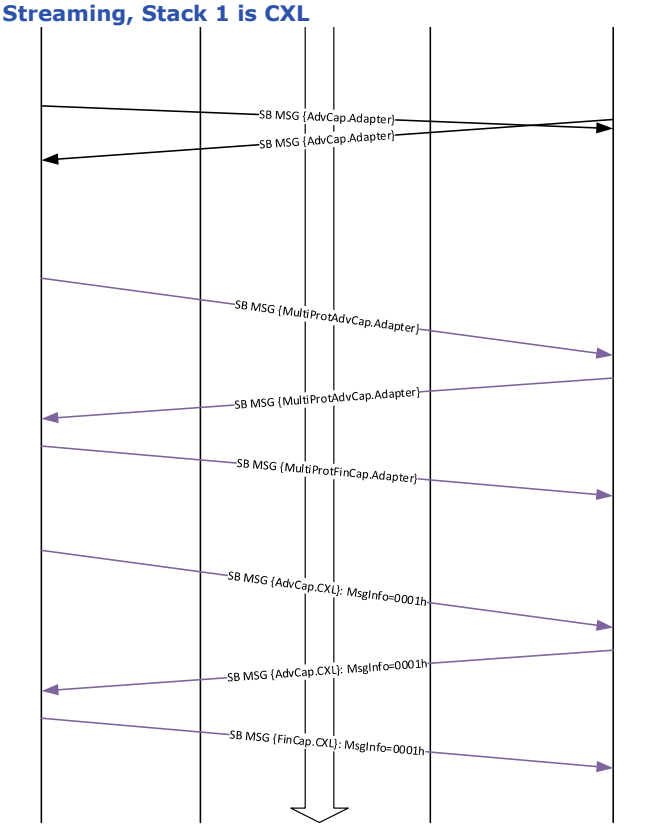


Figure 3-11. Stack0 is Streaming, Stack 1 is CXL



The Adapter must implement a timeout of 8ms (-0%/+50%) for successful Parameter Exchange

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

completion. For the purposes of measuring a timeout for Parameter Exchange completion, all steps in Part 1 and Part 2 of Stage 3 of Link Initialization are included. The timer only increments while RDI is in Active state. The timer must reset if the Adapter receives an {AdvCap.*.Stall},{FinCap.*.Stall},{MultiProtAdvCap.*.Stall}, or {MultiProtFinCap.*.Stall} message from remote Link partner. The 8ms timeouts for Parameter Exchanges or Link State Machine transitions are treated as UIE and the Adapter must take the RDI to LinkError state. UCIe Retimers must ensure they resolve the capability advertisement with remote Retimer partner (and merge with their own capabilities) before responding/initiating parameter exchanges with the UCIe die within its package. While resolution is in progress they must send the corresponding stall message once every 4ms to ensure there isn't a timeout on the UCIe die within its package.

3.2.1.3 Part 3: FDI bring up

Once Parameter Exchanges have successfully completed, the Adapter reflects the result to the Protocol Layers on FDI, and moves on to carry out the FDI bring up flow as defined in Section 8.2.8. Once FDI is in Active state, it concludes Stage 3 of Link Initialization and protocol Flit transfer can begin. When multiple stacks are enabled on the same Adapter, each stack may finish the FDI bring up flow (Section 8.2.8) at different times.

The data width on FDI is a function of the frequency of operation of the UCIe stack as well as the total bandwidth being transferred across the UCIe physical Link (which in turn depends on the number of Lanes and the speed at which the Lanes are operating). The data width on RDI is fixed to at least one byte per physical Lane per module that is controlled by the Adapter. The illustrations of the formats in this chapter are showing an example configuration of RDI mapped to a 64 Lane module of Advanced Package configuration on the Physical Layer of UCIe.

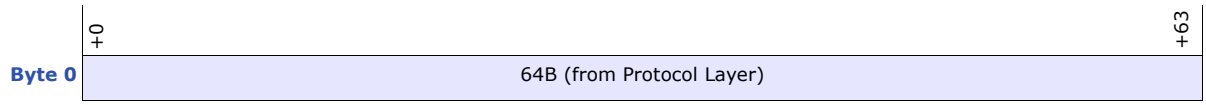
3.3 Operation Formats

In subsequent sections, when referring to CRC computation, a byte mapping of the Flit to CRC message (CRC message is the 128B input to CRC computation logic) is provided. See Section 3.7 for more details.

3.3.1 Raw Format for all protocols

Raw Format can only be used for scenarios in which Retry support from the Adapter is not required. If Raw Format is negotiated for CXL or PCIe protocols, the Adapter transfers data from Protocol Layer to Physical Layer without any modification. Figure 3-12 shows an example of this for a 64B data path on FDI and RDI. This is identified as *Format 1* during parameter negotiation.

Figure 3-12. Format 1: Raw Format



3.3.2 68B Flit Format

This Flit Format is identified as *Format 2* on UCIe. Support for this is mandatory when CXL 68B Flit Mode protocol or PCIe Non-Flit Mode protocol is supported. 68B Flit Format support is optional for Streaming protocols.

The Protocol Layer sends 64B of protocol information. The Adapter adds a two byte prefix of Flit Header and a two byte suffix of CRC. Table 3-3 gives the Flit Header format for *Format 2* when Retry from the Adapter is required. If Retry from the Adapter is not required, then the Flit Header format is as provided in Table 3-2.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Even if Retry is not required, the Adapter still computes and drives CRC bytes - the Receiver is strongly recommended to treat a CRC error as an Uncorrectable Internal Error in this situation. For CRC computation, Flit Byte 0 (i.e., Flit Header Byte 0) is assigned to CRC message Byte 0, Flit Byte 1 (i.e., Flit Header Byte 1) is assigned to CRC message Byte 1 and so on until Flit Byte 65 is assigned to CRC message Byte 65.

Retry is performed over this 68B Flit.

Table 3-2. Flit Header for Format 2 without Retry

Byte	Bit	Description	
		PCIe or CXL	Streaming Protocol
Byte 0	[7:6]	Protocol Identifier: 2'b00 : D2D Adapter NOP Flit or PDS Flit Header 2'b01 : CXL.io Flit 2'b10 : CXL.cachemem Flit 2'b11 : ARB/MUX Flit (Reserved encoding for PCIe)	Protocol Identifier: 2'b00 : D2D Adapter NOP Flit 2'b01 : Protocol Layer Flit Remaining encodings are Reserved.
	[5]	Stack Identifier: 1'b0 : Stack 0 1'b1 : Stack 1	
	[4]	1'b0 : Regular Flit Header 1'b1 : Pause of Data Stream (PDS) Flit Header	
	[3:0]	Reserved	
Byte 1[1]	[7]	1'b0 : Regular Flit Header 1'b1 : Pause of Data Stream (PDS) Flit Header	
	[6:0]	Reserved	

1. For a Test Flit, bits [7:6] of Byte 1 are 01b. See Section 9.2 for more details.

Table 3-3. Flit Header for Format 2 with Retry (Sheet 1 of 2)

Byte	Bit	Description	
		PCIe or CXL	Streaming Protocol
Byte 0	[7:6]	Protocol Identifier: 2'b00 : D2D Adapter NOP Flit or PDS Flit Header 2'b01 : CXL.io Flit 2'b10 : CXL.cachemem Flit 2'b11 : ARB/MUX Flit (Reserved encoding for PCIe)	Protocol Identifier: 2'b00 : D2D Adapter NOP Flit 2'b01 : Protocol Layer Flit Remaining encodings are Reserved.
	[5]	Stack Identifier: 1'b0 : Stack 0 1'b1 : Stack 1	
	[4]	1'b0 : Regular Flit Header 1'b1 : Pause of Data Stream (PDS) Flit Header	
	[3:0]	The upper four bits of Sequence number "S" (i.e., S[7:4])	

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 3-3. Flit Header for Format 2 with Retry (Sheet 2 of 2)

Byte	Bit	Description	
		PCIe or CXL	Streaming Protocol
Byte 1[1]	[7:6]	2'b00 : Regular Flit Header 2'b11 : Pause of Data Stream (PDS) Flit Header Other encodings are reserved	
	[5:4]	Ack or Nak information 2'b00 : Explicit Sequence number "S" of Flit if not PDS, otherwise the bitwise inverted value of "NEXT_TX_FLIT_SEQ_NUM - 1". (See *PCIe Base Specification* for the definition of NEXT_TX_FLIT_SEQ_NUM and the subtraction operation for sequence numbers) 2'b01 : Ack. The Sequence number "S" carries the Ack'ed sequence number. 2'b10 : Nak. The Sequence number "S" carries 255 if N=1, otherwise it carries N-1, where N is the Nak'ed sequence number. 2'b11 : Reserved	
	[3:0]	The lower four bits of Sequence number "S" (i.e., S[3:0]) Sequence number 0 is reserved and if present, it implies no Ack or Nak is sent.	

1. For a Test Flit, bits [7:6] of Byte 1 are 01b. See Section 9.2 for more details.

Because of the four bytes added by D2D Adapter, the alignment of the Flit does not exactly match the number of Lanes of the physical Link (which are always in multiples of 16). The bytes added by D2D Adapter require the Adapter to shift the Flits by four bytes for consecutive Flits. Data is always transferred in multiples of 256B (note that Retimer credits have a 256B data granularity). When the transmitting Adapter has no Flits to send, it terminates the data stream with a Pause of Data Stream (PDS) token followed by 0b padding to the next 64B count multiple boundary and at least two subsequent 64B chunks of all 0 value data. If the transfer is not at a 256B count multiple boundary, additional 64B transfers of all 0 value data are required to bring the transferred bytes to a 256B count multiple. The subsequent transfers of all 0 data mentioned above give the Receiver at least two 64B transfers to reset the receiving byte shifter. The PDS token and the 0 bytes following it must not be forwarded to the Protocol Layer. The PDS token is a variable-size Flit that carries a 2B special Flit Header, and 0 bytes padded on the remaining bytes of RDI. The Transmitter of PDS drives the following on the Flit header:

1. Bit [4] of Byte 0 as 1'b1
2. Bit [7] of Byte 1 as 1'b1
3. Bit [6] of Byte 1 as 1'b1
4. Bits [5:4] of Byte 1 as 2'b00 and the sequence number[7:0] is matching the expected value for a PDS Flit Header in this position

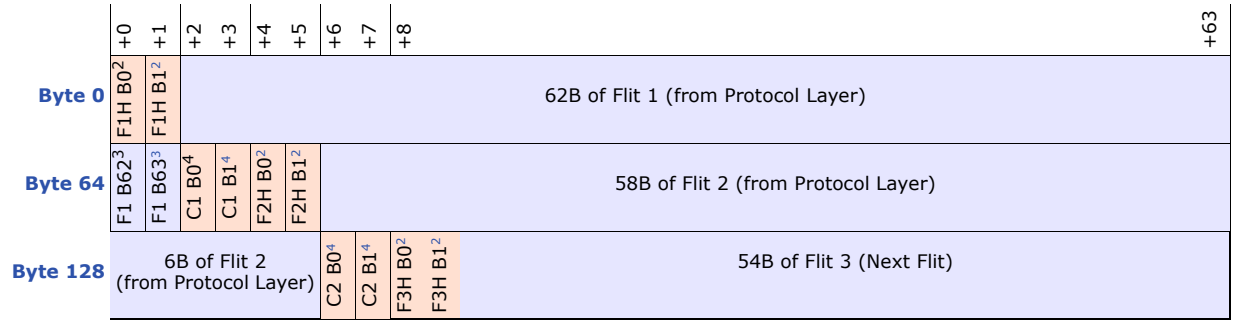
If Retry is enabled, the Receiver must interpret this Flit header as PDS if any two of the above four conditions are true. If Retry is disabled, the Receiver must interpret this Flit header as a PDS if conditions (1) and (2) are true. This guarantees that a PDS will be detected if there are fewer than three bit errors in the Flit Header. For three bit errors, it could result in a retry, but that will be handled seamlessly through the retry rules.

An implicit PDS occurs when Retry is triggered or RDI state goes through Retrain. The transmitter must insert 0s like it would for an actual PDS and start the replayed Flit from fresh alignment. For Retry and Retrain scenarios, the Receiver must also look for the expected sequence number in Byte 0 and Byte 1 of the received data bus.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

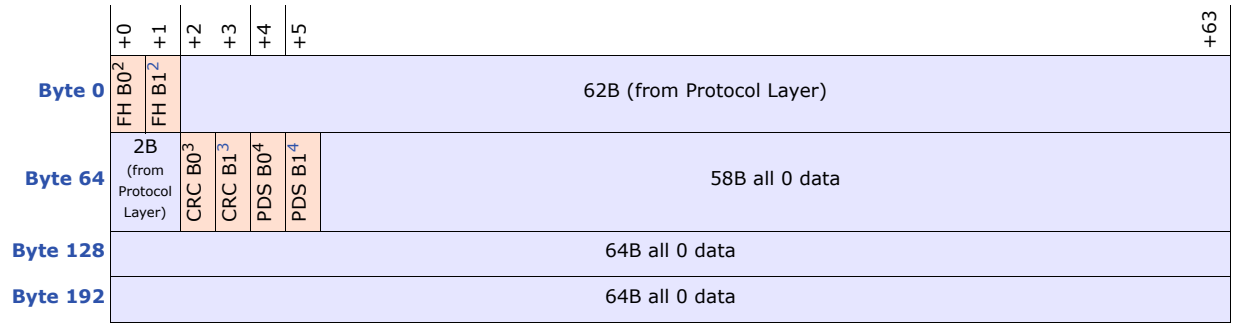
Figure 3-13 shows the 68B Flit Format. Figure 3-14 and Figure 3-15 provide examples of PDS insertion.

Figure 3-13. Format 2: 68B Flit Format[1]



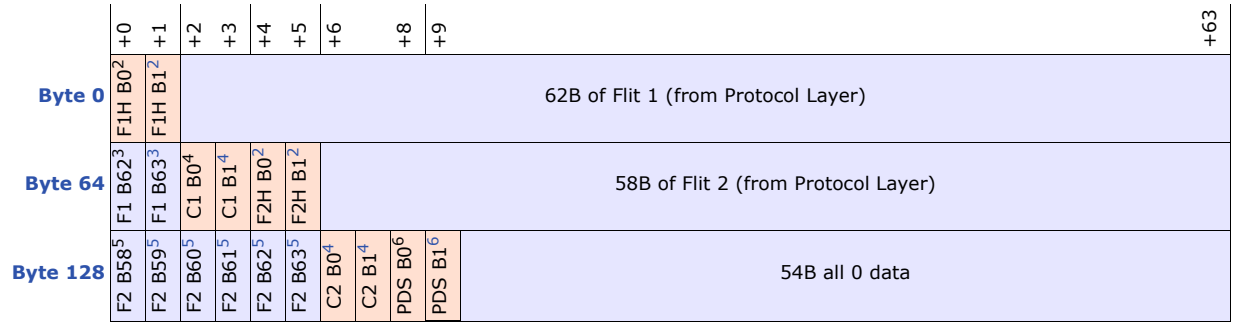
1. See Figure 2-1 for color mapping.
2. Flit 1 Header Byte 0, Flit 1 Header Byte 1, Flit 2 Header Byte 0, Flit 2 Header Byte 1, Flit 3 Header Byte 0, and Flit 3 Header Byte 1 respectively.
3. Flit 1 Byte 62 and Byte 63, respectively (from Protocol Layer).
4. Flit 1 CRC Byte 0, Flit 1 CRC Byte 1, Flit 2 CRC Byte 0, and Flit 2 CRC Byte 1, respectively.

Figure 3-14. Format 2: 68B Flit Format PDS Example 1[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. CRC Byte 0 and Byte 1, respectively.
4. PDS Flit Header Byte 0 and Byte 1, respectively.

Figure 3-15. Format 2: 68B Flit Format PDS Example 2 — Extra 0s Padded to Make the Data Transfer a Multiple of 256B[1]



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-15. Format 2: 68B Flit Format PDS Example 2 — Extra 0s Padded to Make the Data Transfer a Multiple of 256B[1] (Continued)

	+0 +1	+2 +3	+4 +5	+6 +8	+9 +63
Byte 192	64B all 0 data				
Byte 256	64B all 0 data				
Byte 320	64B all 0 data				
Byte 384	64B all 0 data				
Byte 448	64B all 0 data				

1. See Figure 2-1 for color mapping.
2. Flit 1 Header Byte 0, Flit 1 Header Byte 1, Flit 2 Header Byte 0, and Flit 2 Header Byte 1, respectively.
3. Flit 1 Byte 62 and Byte 63, respectively (from Protocol Layer).
4. Flit 1 CRC Byte 0, Flit 1 CRC Byte 1, Flit 2 CRC Byte 0, and Flit 2 CRC Byte 1, respectively.
5. Flit 2 Bytes 58 through Byte 63, respectively (from Protocol Layer).
6. PDS Flit Header Byte 0 and Byte, respectively.

3.3.3 Standard 256B Flit Formats

These are the Standard Flit Formats defined in *PCIe Base Specification* for PCIe Flit Mode and *CXL Specification* for CXL 256B Flit Mode. These are identified as "Standard 256B End Header Flit Format" (or *Format 3*) and "Standard 256B Start Header Flit Format" (or *Format 4*), respectively. Support for this is mandatory when PCIe Flit Mode or CXL 256B Flit Mode protocols are negotiated. Standard 256B Flit Formats (Start Header or End Header) support is optional with Streaming protocols.

The Protocol Layer sends data in 256B Flits, but it drives 0 on the bytes reserved for the Adapter (shown in light orange in Figure 3-16, Figure 3-17, Figure 3-18, and Figure 3-19). The 6B of DLP defined in *PCIe Base Specification* exist in *Format 3* and *Format 4* as well for PCIe and CXL.io protocols. However, since DLLPs are required to bypass the TX Retry buffer in PCIe and CXL.io protocols, the DLP bytes end up being unique since they are partially filled by the Protocol Layer and partially by the Adapter. DLP0 and DLP1 are replaced with the Flit Header for UCIe and are driven by UCIe Adapter. However, if the Flit carries a Flit Marker, the Protocol Layer must populate bit 4 of Flit Header byte 0 to 1b, as well as the relevant information in the Flit_Marker bits (these are driven as defined in *PCIe Base Specification*). Protocol Layer must also populate the Protocol Identifier bits in the Flit Header for the Flits it generates.

For Streaming protocols, Figure 3-18 shows the applicable Flit Format. Protocol Layer only populates bits [7:6] of Byte 0 of the Flit Header, and it must never set 2'b00 for bits [7:6].

Standard 256B Start Header Flit Format is optional for PCIe Flit Mode protocol. Figure 3-19 shows the Flit Format example.

FDI provides a separate interface for DLLP transfer from the Protocol Layer to the Adapter and vice-versa. The Adapter is responsible for inserting DLLP into DLP bytes 2:5 if a Flit Marker is not present. The credit update information is transferred as regular Update_FC DLLPs over FDI from the Protocol Layer to the Adapter. The Adapter is also responsible for formating these updates as Optimized_Update_FC format when possible and driving them on the relevant DLP bytes. The Adapter is also responsible for adhering to all the DLLP rules defined for Flit Mode in *PCIe Base Specification*. On the receive path, the Adapter is responsible for extracting the DLLPs or Optimized_Update_FC from the Flit and driving it on the dedicated DLLP interface provided on FDI.

Two sets of CRC are computed (CRC0 and CRC1). The same 2B over 128B CRC computation as previous formats is used.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

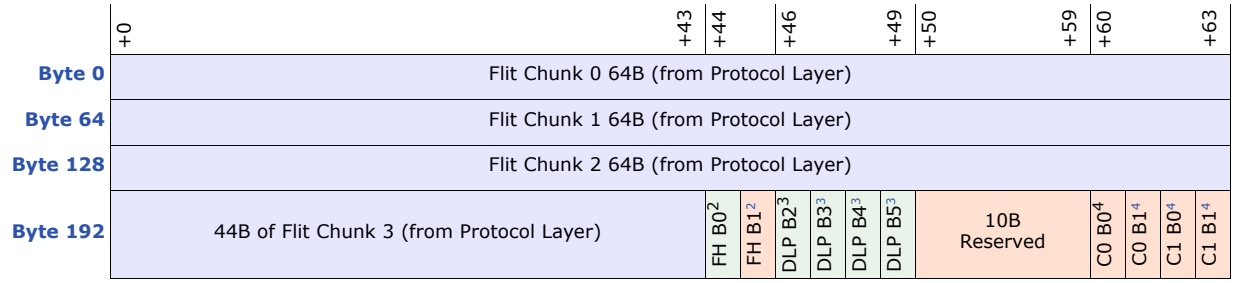
For PCIe, CXL and Streaming:

- For *Format 3*, CRC0 is computed using Flit Bytes 0 to 127 assigned to the corresponding bytes of the CRC message input. CRC1 is computed using Flit bytes 128 to 241 as the message input with Flit Byte 128 assigned to CRC message Byte 0, Flit Byte 129 assigned to CRC message Byte 1 and so on until Flit Byte 241 assigned to CRC message Byte 113 (including the Flit Header bits inserted by the Adapter. For PCIe and CXL.io, this includes the DLP bytes inserted by the Adapter).
- For *Format 4*, CRC0 is computed using Flit Bytes 0 to 127 assigned to the corresponding bytes of the CRC message input (including the Flit Header bits inserted by the Adapter). CRC1 is computed using Flit bytes 128 to 241 as the message input with Flit Byte 128 assigned to CRC message Byte 0, Flit Byte 129 assigned to CRC message Byte 1 and so on until Flit Byte 241 assigned to CRC message Byte 113 (for PCIe and CXL.io, this includes the DLP bytes inserted by the Adapter).

If Retry is not required, the Adapter still computes and drives CRC bytes - the Receiver is strongly recommended to treat a CRC error as an Uncorrectable Internal Error (UIE) in this situation.

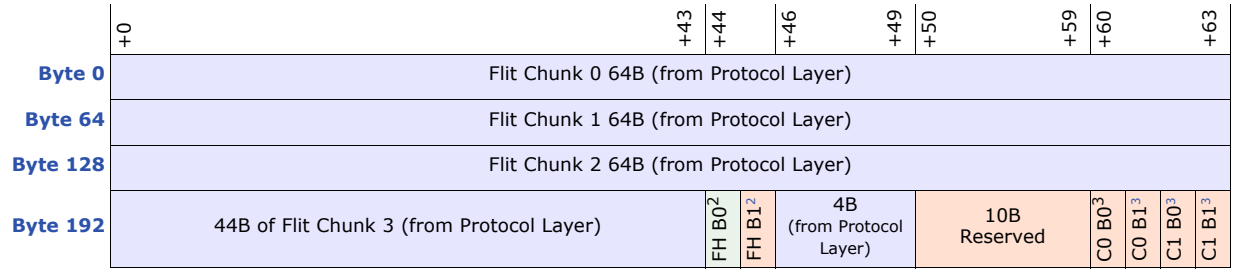
The Flit Header byte formats are shown in Table 3-5 when Retry is required; otherwise, it is as shown in Table 3-4.

Figure 3-16. Format 3: Standard 256B End Header Flit Format for PCIe[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. DLP Byte 2, Byte 3, Byte 4, and Byte 5, respectively.
4. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

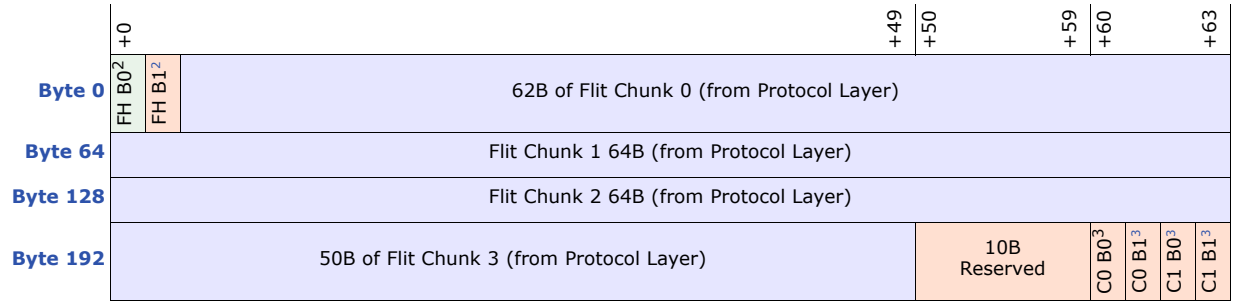
Figure 3-17. Format 3: Standard 256B End Header Flit Format for Streaming Protocol[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

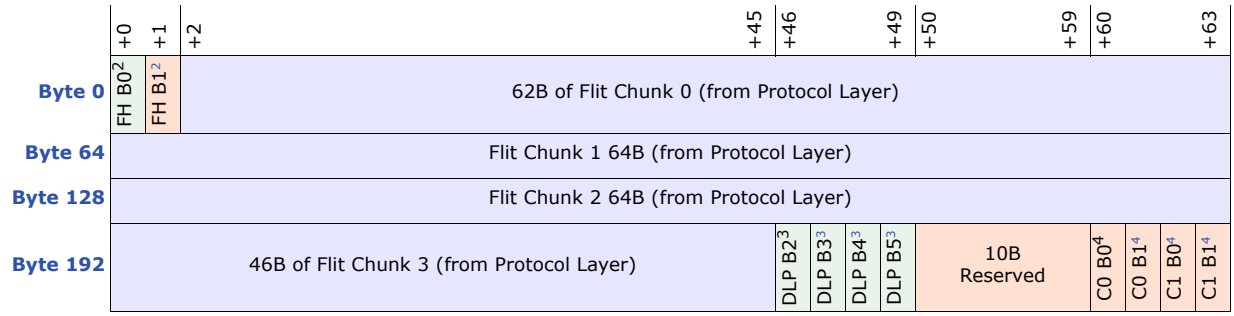
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-18. Format 4: Standard 256B Start Header Flit Format for CXL.cachemem or Streaming Protocol[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

Figure 3-19. Format 4: Standard 256B Start Header Flit Format for CXL.io or PCIe[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. DLP Byte 2, Byte 3, Byte 4, and Byte 5, respectively.
4. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

Table 3-4. Flit Header for Format 3 or Format 4 without Retry

Byte	Bit	Description		
		CXL 256B Flit Mode	PCIe Flit Mode	Streaming
Byte 0	[7:6]	Protocol Identifier: 2'b00 : D2D Adapter/CXL.io NOP Flit 2'b01 : CXL.io Flit 2'b10 : CXL.cachemem Flit 2'b11 : ARB/MUX Flit	Protocol Identifier: 2'b00 : D2D Adapter/PCIe NOP Flit 2'b01 : PCIe Flit All other encodings are reserved	Protocol Identifier: 2'b00 : D2D Adapter NOP Flit Remaining encodings are permitted to be used by Protocol Layer in a vendor defined manner. Protocol Layer must never set this to 2'b00 for Flits sent across FDI.
	[5]	Stack Identifier: 1'b0 : Stack 0 1'b1 : Stack 1		
	[4]	Reserved for CXL.cachemem For CXL.io or PCIe Flit Mode: 0b DLLP Payload in DLP 2..5 1b Optimized_Update_FC or Flit_Marker in DLP2..5		Reserved
	[3:0]	Reserved		
Byte 1[1]	[7:0]	Reserved		

1. For a Test Flit, bits [7:6] of Byte 1 are 01b. See Section 9.2 for more details.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 3-5. Flit Header for Format 3 or Format 4 with Retry

Byte	Bit	Description		
		CXL 256B Flit Mode	PCIe Flit Mode	Streaming
Byte 0	[7:6]	Protocol Identifier: 2'b00 : D2D Adapter/CXL.io NOP Flit 2'b01 : CXL.io Flit 2'b10 : CXL.cachemem Flit 2'b11 : ARB/MUX Flit	Protocol Identifier: 2'b00 : D2D Adapter/PCIe NOP Flit 2'b01 : PCIe Flit All other encodings are reserved	Protocol Identifier: 2'b00 : D2D Adapter NOP Flit Remaining encodings are permitted to be used by Protocol Layer in a vendor defined manner. Protocol Layer must never set this to 2'b00 for Flits sent across FDI.
	[5]	Stack Identifier: 1'b0 : Stack 0 1'b1 : Stack 1		
	[4]	Reserved for CXL.cachemem For CXL.io or PCIe Flit Mode: 0b DLLP Payload in DLP 2..5 1b Optimized_Update_FC or Flit_Marker in DLP2..5		Reserved
	[3:0]	The upper four bits of Sequence number "S" (i.e., S[7:4])		
Byte 1[1]	[7:6]	Reserved		
	[5:4]	Ack or Nak information 2'b00 : Explicit Sequence number "S" of the current Flit is present 2'b01 : Ack. The sequence number "S" carries the Ack'ed sequence number. 2'b10 : Nak. The sequence number "S" carries 255 if N=1, otherwise it carries N-1; where N is the Nak'ed sequence number. 2'b11 : Reserved		
	[3:0]	The lower four bits of Sequence number "S" (i.e., S[3:0]) Sequence number 0 is reserved and if present, it implies no Ack or Nak is sent.		

1. For a Test Flit, bits [7:6] of Byte 1 are 01b. See Section 9.2 for more details.

3.3.4 Latency-Optimized 256B Flit Formats

Two Latency-Optimized 256B Flit Formats are defined: *Format 5* and *Format 6*. It is strongly recommended that UCIe implementations support *Format 6* for CXL 256B Flit Mode protocol to get the best latency benefits.

Both formats look the same from the Adapter perspective, the only difference is whether the Protocol Layer is filling in the optional bytes of protocol information . The Latency-Optimized 256B without Optional bytes Flit Format (or *Format 5*) is when the Protocol Layer is not filling in the optional bytes, whereas the Latency-Optimized 256B with Optional bytes Flit Format (or *Format 6*) is when the Protocol Layer is filling in the optional bytes.

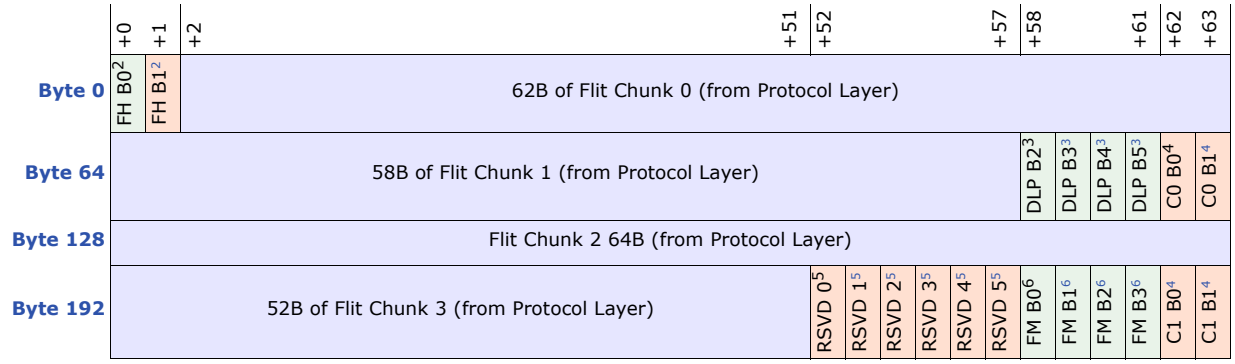
Latency-Optimized 256B Flit Formats (with Optional bytes or without Optional bytes) support is optional with Streaming protocols. Protocol Layer only populates bits [7:6] of the Flit Header, and it must never set 2'b00 for bits [7:6].

Latency-Optimized Flit with Optional Bytes Flit Format is optional for PCIe Flit Mode protocol. Figure 3-22 shows the Flit Format example.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

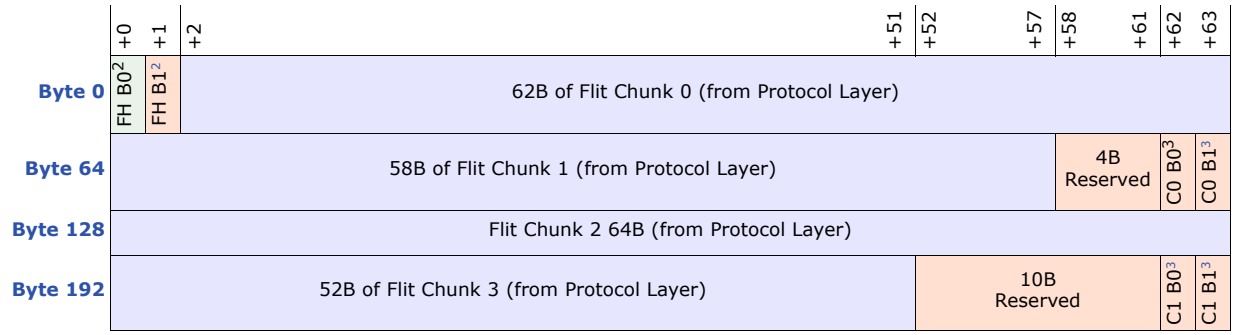
Two sets of CRC are computed. CRC0 is computed using Flit Bytes 0 to 125 assigned to the corresponding bytes of the CRC message input (including the Flit Header bits and if applicable, the DLP bits inserted by the Adapter). CRC1 is computed using Flit bytes 128 to 253 as the message input with Flit Byte 128 assigned to CRC message Byte 0, Flit Byte 129 assigned to CRC message Byte 1 and so on until Flit Byte 253 assigned to CRC message Byte 125. If Retry is not required, the Adapter still computes and drives CRC bytes - the Receiver is strongly recommended to treat a CRC error as UIE in this situation.

Figure 3-20. Format 5: Latency-Optimized 256B without Optional Bytes Flit Format for CXL.io[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. DLP Byte 2, Byte 3, Byte 4, and Byte 5, respectively.
4. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.
5. Reserved Byte 0 through Byte 5, respectively.
6. Flit_Marker or Optimized_Update_FC Byte 0, Byte 1, Byte 2, and Byte 3, respectively.

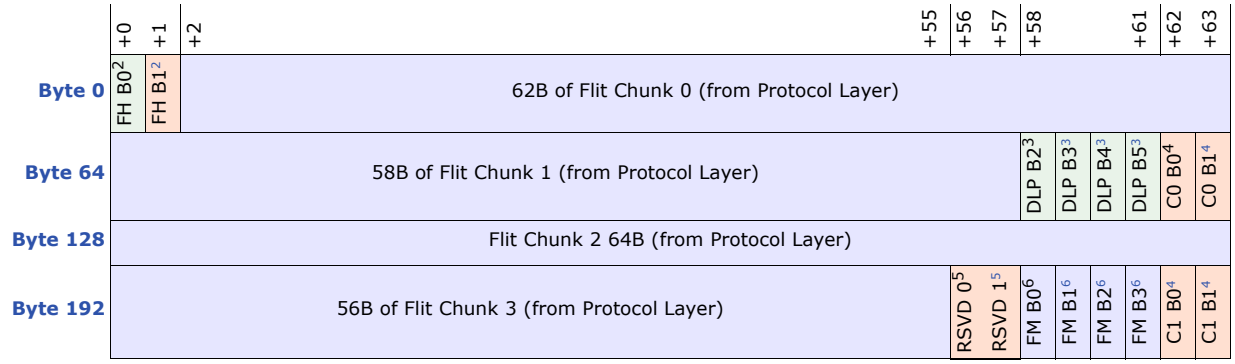
Figure 3-21. Format 5: Latency-Optimized 256B without Optional Bytes Flit Format for CXL.cachemem and Streaming Protocol[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

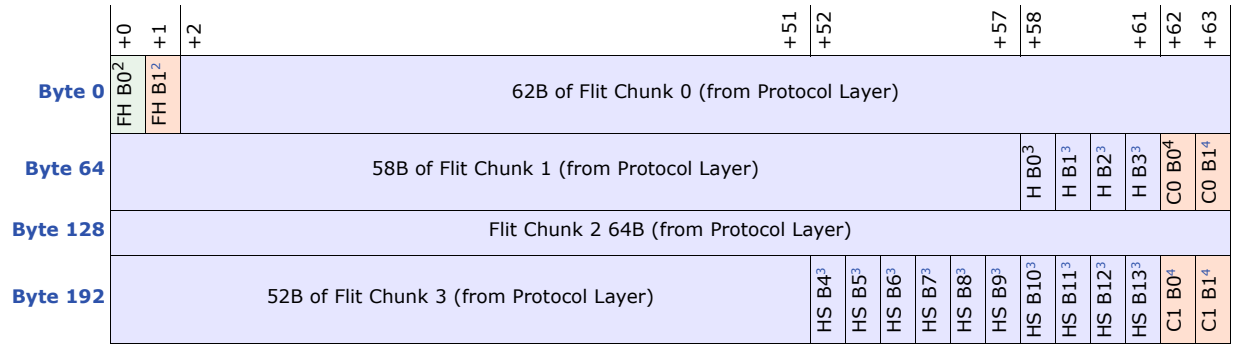
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-22. Format 6: Latency-Optimized 256B with Optional Bytes Flit Format for CXL.io or PCIe[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. DLP Byte 2, Byte 3, Byte 4, and Byte 5, respectively.
4. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.
5. Reserved Byte 0 and Byte 1, respectively.
6. Flit_Marker Byte 0, Byte 1, Byte 2, and Byte 3, respectively.

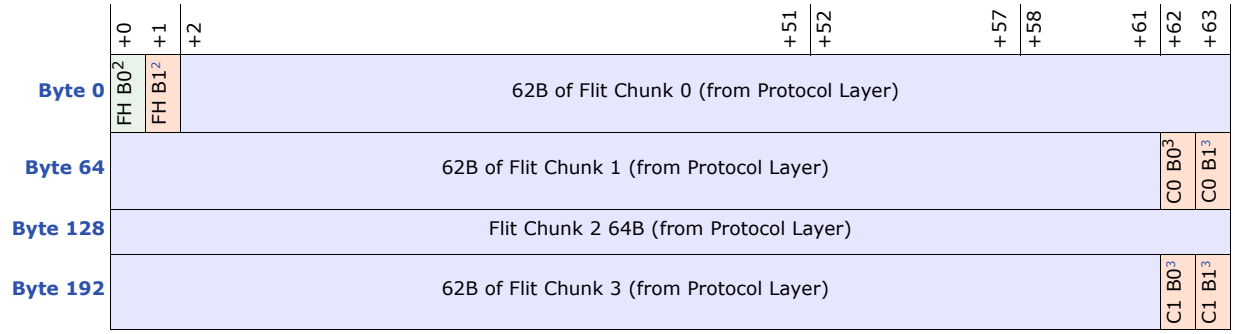
Figure 3-23. Format 6: Latency-Optimized 256B with Optional Bytes Flit Format for CXL.cachemem[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. H-slot Byte 0 through Byte 13, respectively (from Protocol Layer).
4. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-24. Format 6: Latency-Optimized 256B with Optional Bytes Flit Format for Streaming Protocol[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

The Flit Header byte formats are the same as Table 3-5 when Retry is required; otherwise, they are the same as Table 3-4. The DLP rules are also the same as defined in Section 3.3.3 for CXL protocol, except that Flit_Marker/Optimized_Update_FC has dedicated space in the Flit (i.e., bit[4] of Byte 0 corresponds to the Flit_Marker bytes, and not the DLP bytes). If Optimized_Update_FC is sent, the DLP bytes 2:5 shown in Figure 3-20 must be reserved. If bit[4] of Byte 0 in the Flit Header is 0b, then the Flit_Marker bytes are reserved.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.3.5 Flit Format-related Implementation Requirements for Protocol Layer and Adapter

Table 3-6 gives the list of the different Flit Formats supported in UCIe.

Table 3-6. Summary of Flit Formats

Format Number	Name	Notes	For Details, See Also
Format 1	Raw	Protocol Layer populates all the bytes on FDI. Adapter passes to RDI without modifications or additions.	• Section 3.3.1 • Figure 3-12
Format 2	68B Flit	Protocol Layer transmits 64B per Flit on FDI. Adapter inserts two bytes of Flit header and two bytes of CRC and performs the required barrel shifting of bytes before transmitting on RDI. On the RX, Adapter strips out the Flit header and CRC only sending the 64B per Flit to the Protocol Layer on FDI.	• Section 3.3.2 • Figure 3-13 • Figure 3-14
Format 3	Standard 256B End Header Flit	Protocol Layer transmits 256B of Flit on FDI, while driving 0b on the bits reserved for the Adapter. Adapter fills in the relevant Flit header and CRC information before transmitting on RDI. On the Rx, Adapter forwards the Flit received from the Link to the Protocol Layer without modifying any bits applicable to the Protocol Layer, and the Protocol Layer must ignore any bits not applicable for it. Flit Header is located on Byte 236 and Byte 237 of the Flit.	• Section 3.3.3 • Figure 3-16 • Figure 3-17
Format 4	Standard 256B Start Header Flit	Protocol Layer transmits 256B of Flit on FDI, while driving 0b on the bits reserved for the Adapter. Adapter fills in the relevant Flit header and CRC information before transmitting on RDI. On the Rx, Adapter forwards the Flit received from the Link to the Protocol Layer without modifying any bits applicable to the Protocol Layer, and the Protocol Layer must ignore any bits not applicable for it. Flit Header is located on Byte 0 and Byte 1 of the Flit.	• Section 3.3.3 • Figure 3-18 • Figure 3-19
Format 5	Latency-Optimized 256B without Optional Bytes Flit	Protocol Layer transmits 256B of Flit on FDI, while driving 0b on the bits reserved for the Adapter. Adapter fills in the relevant Flit header and CRC information before transmitting on RDI. On the Rx, Adapter forwards the Flit received from the Link to the Protocol Layer without modifying any bits applicable to the Protocol Layer, and the Protocol Layer must ignore any bits not applicable for it. CRC bytes sent with each 128B of the Flit. The optional Protocol Layer bytes are reserved in this format and not used by the Protocol Layer.	• Section 3.3.4 • Figure 3-20 • Figure 3-21
Format 6	Latency-Optimized 256B with Optional Bytes Flit	Protocol Layer transmits 256B of Flit on FDI, while driving 0b on the bits reserved for the Adapter. Adapter fills in the relevant Flit header and CRC information before transmitting on RDI. On the Rx, Adapter forwards the Flit received from the Link to the Protocol Layer without modifying any bits applicable to the Protocol Layer, and the Protocol Layer must ignore any bits not applicable for it. CRC bytes sent with each 128B of the Flit, and optional bytes are used by the Protocol Layer.	• Section 3.3.4 • Figure 3-22 • Figure 3-23 • Figure 3-24

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 3-7 gives the implementation requirements and Protocol Mapping for the different Flit Formats. For PCIe and CXL protocols, the implementation requirements must be followed by the Protocol Layer as well as the Adapter implementations. For Streaming protocols, the implementation requirements are for the Adapter only; Protocol Layer interoperability and implementation requirements are vendor specific.

Table 3-7. Protocol Mapping and Implementation Requirements

Format Number	Flit Format Name	PCIe Non-Flit Mode	PCIe Flit Mode	CXL 68B Flit Mode	CXL 256B Flit Mode	Streaming
1	Raw	Optional	Optional	Optional	Optional	Mandatory
2	68B	Mandatory	N/A	Mandatory	N/A	Optional[1]
3	Standard 256B End Header	N/A	Mandatory	N/A	N/A	Optional[1]
4	Standard 256B Start Header	N/A	Optional[2]	N/A	Mandatory	Optional[1]
5	Latency-Optimized 256B without Optional Bytes	N/A	N/A	N/A	Optional	Optional[1]
6	Latency-Optimized 256B with Optional Bytes	N/A	Strongly Recommended[3]	N/A	Strongly Recommended	Strongly Recommended[1]

1. If Streaming Flit Format capability is supported, else it is N/A.
2. If Standard Start Header for PCIe protocol capability is supported, else it is N/A.
3. If Latency-Optimized Flit with Optional Bytes for PCIe protocol capability is supported, else it is N/A.
   If Enhanced Multi-Protocol capability is supported where at least one of the stacks supports PCIe, this format and the corresponding capability are strongly recommended.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.4 Decision table for Flit Format and Protocol

Table 3-8 shows the Truth Table for determining Protocol. Once the protocol and Flit Format have been negotiated during initial Link bring up, they cannot be changed until the UCIe Physical Layer transitions to Reset state.

If a valid Protocol and Flit Format are not negotiated, then the Adapter takes the Link down and reports the error if applicable.

Table 3-8. Truth Table for determining Protocol[1]

{FinCap.Adapter} bits or {MultiProtFinCap.Adapter} bits[2]				{FinCap.CXL} bits		Protocol
68B Flit Mode	CXL 256B Flit Mode	PCIe Flit Mode	Streaming	PCIe	CXL.io	
1	1	1	x	0	1[3]	CXL[4]
1	0	1	x	1[5]	0	PCIe
1	0	0	x	0	1	CXL[4]
1	0	0	x	1	0	PCIe[6]
N/A	N/A	N/A	N/A	N/A	N/A	Streaming[7]

1. x indicates don't care in this Table.
2. If Enhanced_Multi-Protocol capability is negotiated then {MultiProt*.Adapter} messages are used to determine the protocol for Stack 1. Stack 0 protocol is determined using the {FinCap.*} messages.
3. CXL.io capable/enable must be 1b if CXL 256B Flit Mode is negotiated.
4. For CXL protocol, the specific combination of Single Protocol vs Type 1 vs. Type 2 vs. Type 3 is determined using the CXL.cache and CXL.mem capable/enable bits in addition to the CXL.io capable/enable bit in {FinCap.CXL}. The rules for that follow *CXL Specification*. When CXL is the protocol, if CXL 256B Flit Mode is 1, then the protocol follows CXL 256B Flit Mode rules; otherwise, the protocol follows CXL 68B Flit Mode rules.
5. PCIe capable/enable must be 1b if PCIe Flit Mode is 1b but CXL 256B Flit Mode is 0b.
6. For PCIe protocol, if PCIe Flit Mode is 1, then the protocol follows PCIe Flit Mode rules; otherwise, the protocol follows PCIe Non-Flit mode rules.
7. No {FinCap.*} message is sent for Streaming protocol negotiation, it is the negotiated protocol if PCIe or CXL are not negotiated, but Streaming protocol is advertised.

Table 3-9 (Truth Table 1) shows the truth table for deciding the Flit Format in which to operate if PCIe or CXL protocols are negotiated, and none of the following are negotiated:

- Enhanced Multi_Protocol_Enable
- Standard 256B Start Header for PCIe protocol capability
- Latency-Optimized Flit with Optional Bytes for PCIe protocol capability

Table 3-10 (Truth Table 2) provides the Truth Table for determining the Flit Format for Streaming Protocols if Streaming Flit Format capability is negotiated. Note that for Streaming Protocol negotiation, there are no {FinCap.*} messages exchanged. Each side of the UCIe Link advertises its own capabilities in the {AdvCap.Adapter} message it sends. The bits in Table 3-10 represent the logical AND of the corresponding bits in the sent and received {AdvCap.Adapter} messages. Truth Table 2 must be followed for determining the Flit Format if both sides of the Link have any of the following capabilities are supported and enabled for both sides of the Link:

- Enhanced Multi-Protocol Capability
- Standard Start Header Flit for PCIe protocol capability
- Latency-Optimized Flit with Optional Bytes for PCIe protocol capability

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

For situations where {FinCap.Adapter} messages are sent, the bits in the truth table represent the bits set in the {FinCap.Adapter} message.

It is permitted for the Adapter OR the Protocol Layer to take the Link down to LinkError if the desired Flit Format is not negotiated.

Table 3-9. Truth Table 1

{FinCap.Adapter} bits[1]						Flit Format
Raw Format	68B Flit Mode	CXL 256B Flit Mode	PCIe Flit Mode	CXL_LatOpt_ Fmt5	CXL_LatOpt_ Fmt6	
1	x	x	x	x	x	*Format 1*: Raw Format
0	x	1	x	0	0	*Format 4*: Standard 256B Start Header Flit Format for CXL
0	x	1	x	x	1	*Format 6*: Latency-Optimized 256B with Optional Bytes Flit Format for CXL
0	x	1	x	1	0	*Format 5*: Latency-Optimized 256B without Optional Bytes Flit Format for CXL
0	x	0	1	x	x	*Format 3*: Standard 256B End Header Flit Format for PCIe
0	1	0	0	x	x	*Format 2*: 68B Flit Format

1. x indicates don't care.

Table 3-10. Truth Table 2

Raw Format[1]	68B Flit Format[2]	Logical AND of Corresponding Bits in the Sent and Received {AdvCap.Adapter} Message OR the Bits Sent in the {FinCap.Adapter} Message[2]				Final Negotiated Flit Format[3]
		Standard 256B End Header Flit Format	Standard 256B Start Header Flit Format	Latency-Optimized 256B without Optional Bytes Flit Format	Latency-Optimized 256B with Optional Bytes Flit Format	
1	x	x	x	x	x	*Format 1*: Raw Format
0	1	0	0	x	0	*Format 2*: 68B Flit Format
0	x	1	0	x	0	*Format 3*: Standard 256B End Header Flit Format
0	x	x	1	x	0	*Format 4*: Standard 256B Start Header Flit Format
0	0	0	0	1	0	*Format 5*: Latency-Optimized 256B without Optional Bytes Flit Format
0	x	x	x	x	1	*Format 6*: Latency-Optimized 256B with Optional Bytes Flit Format

1. Raw Format is always explicitly enabled through UCIe Link Control register and advertised only when it is the required format of operation to ensure interoperability, and therefore appears as a higher priority in the decision table.
2. x indicates don't care.
3. Format 6 is the highest priority format when Raw Format is not advertised because it has the best performance characteristics. Between Format 4 and Format 3, Format 4 is higher priority because it enables lower latency through the D2D Adapter when multiplexing different protocols. Format 5 has the highest overhead and therefore has the lowest priority relative to other formats.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.5 State Machine Hierarchy

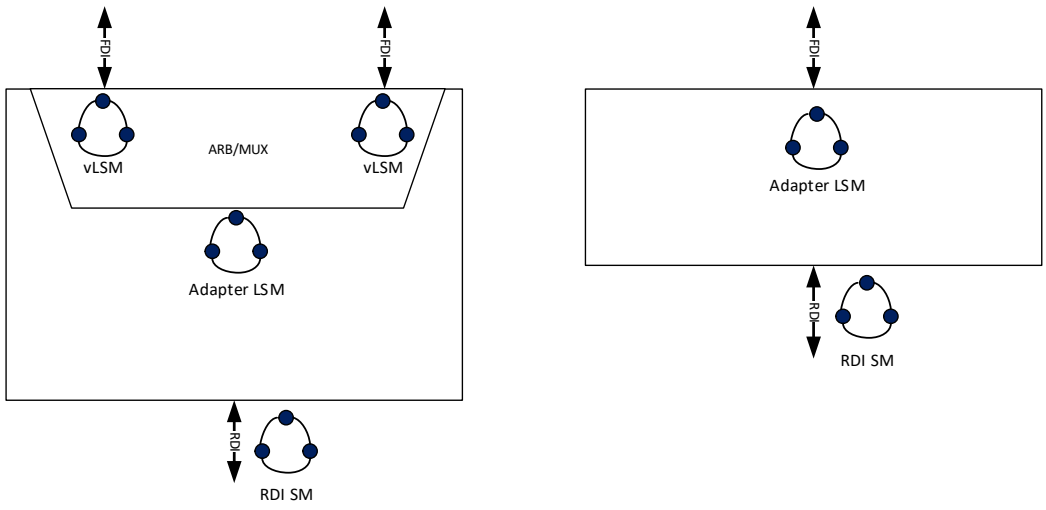
UCIe has a hierarchical approach to Link state management in order to have well-defined functionality partitioning between the different layers and also enabling common state transitions or sequencing at FDI and RDI.

Figure 3-25 shows examples of state machine hierarchy for different configurations. For CXL, the ARB/MUX vLSMs are exposed on FDI **`pl_state_sts`**. The Adapter LSM is used to coordinate Link states with remote Link Partner and is required for all configurations. Each protocol stack has its corresponding Adapter LSM. For PCIe or Streaming protocols, the Adapter LSM is exposed on FDI **`pl_state_sts`**.

The RDI state machine (SM) is used to abstract the Physical Layer states for the upper layers. The Adapter data path and RDI data width can be extended for multi-module configurations; however, there is a single RDI state machine for this configuration. The Multi-module PHY Logic creates the abstraction and coordinates between the RDI state and individual modules. The following rules apply:

- vLSM state transitions are coordinated with remote Link partner using ALMPs on mainband data path. The rules for state transitions follow the CXL 256B Flit Mode rules in the *CXL Specification*.
- Adapter LSM state transitions are coordinated with remote Link partner using {LinkMgmt.Adapter*} sideband messages. These messages are originated and received by the D2D Adapter.
- RDI SM state transitions are coordinated with the remote Link partner using {LinkMgmt.RDI*} sideband messages. These messages are originated and received by the Physical Layer.

Figure 3-25. State Machine Hierarchy examples



General rules for State transition hierarchy are captured below. For specific sequencing, see the rules outlined in Chapter 8.0.

- Active State transitions: RDI SM must be in Active before Adapter LSM can begin negotiation to transition to Active. Adapter LSM must be in Active before vLSMs can begin negotiations to transition to Active.
- Retrain State transitions: RDI SM must be in Retrain before propagating Retrain to Adapter LSMs. If RDI SM is in Retrain, Retrain must be propagated to all Adapter LSMs that are in Active state.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Adapter must not request Retrain exit on RDI before all the relevant Adapter LSMs have transitioned to Retrain.

- PM State transitions (both L1 and L2): Both CXL.io and CXL.cachemem vLSMs (if CXL), must transition to PM before the corresponding Adapter LSM can transition to PM. All Adapter LSMs (if multiple stacks are enabled on the same Adapter) must be in PM before RDI SM is transitioned to PM.
- LinkError State transitions: RDI SM must be in LinkError before Adapter LSM can transition to LinkError. RDI SMs coordinate LinkError transition with remote Link partner using sideband, and each RDI SM propagates LinkError to all enabled Adapter LSMs. Adapter LSM must be in LinkError before propagating LinkError to both vLSMs if CXL. LinkError transition takes priority over LinkReset or Disabled transitions. Adapter must not request LinkError exit on RDI before all the relevant Adapter LSMs and CXL vLSMs have transitioned to LinkError.
- LinkReset or Disabled State transitions: Adapter LSM negotiates LinkReset or Disabled transition with its remote Link partner using sideband messages. LinkReset or Disabled is propagated to RDI SM only if all the Adapter LSMs associated with it transition to LinkReset or Disabled. Disabled transition takes priority over LinkReset transition. If RDI SM moves to LinkReset or Disabled, it must be propagated to all Adapter LSMs. If Adapter LSM moves to LinkReset or Disabled, it must propagate it to both vLSMs for CXL protocol.

For UCIe Retimers, it is the responsibility of the Retimer die to negotiate state transitions with the remote Retimer partner and make sure the different UCIe Die are in sync and do not time out waiting for a response. As an example, referring to Figure 1-14, if UCIe Die 0 sends an Active Request message for the Adapter LSM to UCIe Retimer 0, UCIe Retimer 0 must resolve with UCIe Retimer 1 that an Active Request message has been forwarded to UCIe Die 1 and that UCIe Die 1 has responded with an Active Status message before responding to UCIe Die 0 with an Active Status message. The Off Package Interconnect cannot be taken to a low power state unless all the relevant states on UCIe Die 0 AND UCIe Die 1 have reached the low power state. UCIe Retimers must respond with "Stall" encoding every 4ms while completing resolution with the remote Retimer partner.

3.6 Power Management Link States

Power management states are mandatory for PCIe and CXL protocols. FDI supports L1 and L2 power states which follow the handshake rules and state transitions of CXL 256B Flit Mode. RDI supports L1 and L2 on the interfaces for Physical Layer to perform power management optimizations, however the Physical Layer is permitted to map both L1 and L2 to a common state internally. These together allow for global clock gating and enable system level flows like Package-Level Idle (C-states). Other Protocols are permitted to disable PM flows by always sending a PMNAK for a PM request from remote Link partner.

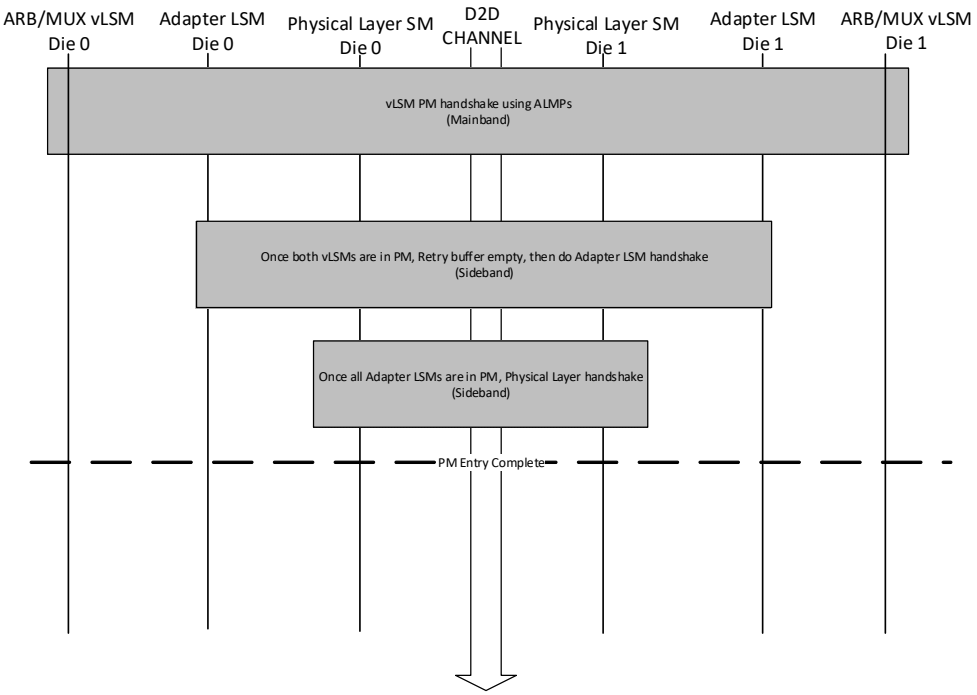
The Power management state entry sequence is as follows:

1. **Protocol Layer PM entry request**: FDI defines a common flow for PM entry request at the interface that is based on Link idle time. All protocols using UCIe must follow that flow when PM needs to be supported. For CXL protocol, D2D Adapter implements the ARB/MUX functionality and follows the handshakes defined in *CXL Specification* (corresponding to the "CXL 256B Flit Mode", since all ALMPs also go through the Retry buffer in UCIe). Even CXL 68B Flit Mode over UCIe uses the "CXL 256B Flit Mode" ALMP formats and flows (but the Flit is truncated to 64B and two bytes of Flit header and two bytes of CRC are added by the Adapter to make a 68B Flit). For PCIe protocol in UCIe Flit Mode, PM DLLP handshakes are NOT used. Protocol Layer requests PM entry on FDI based on Link idle time. The specific algorithm and hysteresis for determining Link idle time is implementation specific.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2. **Adapter Link State Machine PM entry**: The PM transition for this is coordinated over sideband with remote Link partner. In scenarios where the Adapter is multiplexing between two protocol stacks, each stack's Link State Machine must transition to PM independently.
3. **PM entry on RDI**: Once all the Adapter's LSMs are in a PM state, the Adapter initiates PM entry on the RDI as defined in Section 8.2.9.
4. Physical Layer moves to a deeper PM state and takes the necessary actions for power management. Note that the sideband Link must remain active because the sideband Link is used to initiate PM exit.

Figure 3-26. Example of hierarchical PM entry for CXL



PM exit follows the reverse sequence of wake up as mentioned below:

1. Active request from Protocol Layer is transmitted across the FDI and RDI to the local Physical Layer.
2. The Physical Layer uses sideband to coordinate wake up and retraining of the physical Link.
3. Once the physical Link is retrained, the RDI is in Active state on both sides, and the Adapter LSM PM exit is triggered from both sides (coordinated via sideband messages between Adapters as outlined in the FDI PM flow). For PCIe or Streaming protocol scenarios, this also transitions the Protocol Layer to Active state on FDI.
4. For CXL protocol, this step is followed by ALMP exchanges to bring the required protocol to Active state and then protocol Flit transfer can begin.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.7 CRC Computation

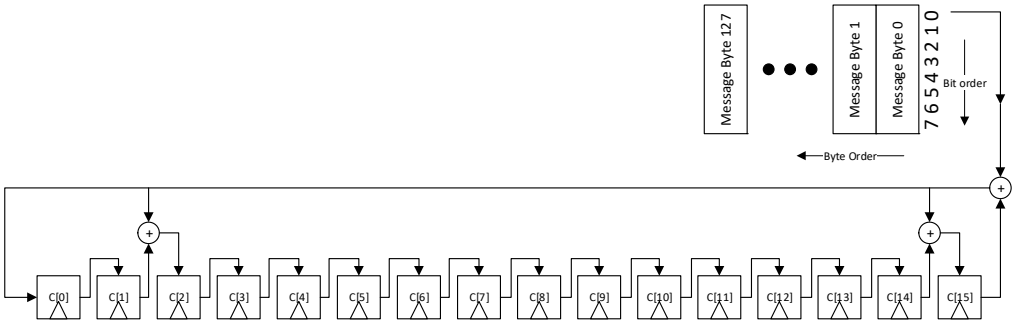
The CRC generator polynomial is $(x+1)*(x^{15} + x + 1) = x^{16} + x^{15} + x^2 + 1$. This gives a 3-bit detection guarantee for random bit errors: 2 bit detection guarantee is because of the primitive polynomial $(x^{15} + x + 1)$, and 1 additional bit error detection guarantee is provided by making it odd parity because of the (x+1) term in the polynomial.

The CRC is always computed over 128 Bytes of the message. For smaller messages, the message is zero extended in the MSB. Any bytes which are part of the 128B CRC message but are not transmitted over the Link are assigned to 0b. Whenever non-CRC bytes of the Flit populated by the Adapter are included for CRC computation (e.g., the Flit Header or DLP bytes), CRC is computed after the Adapter has assigned those bytes the values that will be sent over the UCIe Link. Any reserved bits which are part of the Flit are assigned 0b for the purpose of CRC computation.

The initial value of CRC bits for CRC LFSR computation is 0000h. The CRC calculation starts with bit 0 of byte 0 of the message, and proceeds from bit 0 to bit 7 of each byte as shown in Figure 3-27. In the figure, C[15] is bit 7 of CRC Byte 1, C[14] is bit 6 of CRC Byte 1 and so on; C[7] is bit 7 of CRC Byte 0, C[6] is bit 6 of CRC Byte 0 and so on.

The Verilog code for CRC code generation is provided in crc_gen.vs (attached to the PDF copy of this Specification). This Verilog code must be used as the golden reference for implementing the CRC during encode or decode. The code is provided for the Transmit side. It takes 1024 bits (bit 1023 is bit 7 of message Byte 127, 1022 is bit 6 of message byte 127 and so on; bit 1015 is bit 7 of message Byte 126 and so on until bit 0 is bit 0 of message Byte 0) as an input message and outputs 16 bits of CRC. On the Receiver, the CRC is computed using the received Flit bytes with appropriate zero padding in the MSB to form a 128B message. If the received CRC does not match the computed CRC, the flit is declared Invalid and a replay must be requested.

Figure 3-27. Diagram of CRC calculation



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.8 Retry Rules

For configurations where the raw BER is higher than 1e-27, Retry must be supported in the Adapter, unless the only format of operation is Raw Format. If Retry is not supported by the Adapter, Link speeds where the raw BER is higher than 1e-27 must NOT be advertised by the Physical Layer during Link Training, unless the format of operation is Raw Format. See Table 5-23 for the raw BER characteristics of different configurations. Once Retry has been negotiated during Part 2 of Stage 3 of Link Initialization described in Section 3.2.1.2, it cannot be disabled even if Link speed degrades during runtime. Retry can only be re-negotiated at the next Link Initialization (i.e., RDI moves to Reset). For multiple stacks with a common Adapter, the Tx Retry buffer is shared between the stacks.

The Retry scheme on UCIe is a simplified version of the Retry mechanism for Flit Mode defined in *PCIe Base Specification*. The rules that do not apply and the corresponding parameter changes are enumerated here:

- Selective Nak and associated rules are not applicable and must not be implemented. RX Retry Buffer-related rules are also not applicable and must not be implemented.
- Throughout the duration of Link operation, when not conflicting with PCIe rules of replay, Explicit Sequence number Flits and Ack/Nak Flits alternate. This allows for faster Ack turnaround and thus smaller Retry buffer sizes. It is permitted to send consecutive Explicit Sequence number Flits if there are no Ack/Nak Flits to send. To meet this requirement, all Explicit Sequence Number Flit transmissions described by the PCIe rules of replay that require the condition "CONSECUTIVE_TX_EXPLICIT_SEQ_NUM_FLIT < 3" to be met require "CONSECUTIVE_TX_EXPLICIT_SEQ_NUM_FLIT < 1" to be met instead, and it is not required to send three consecutive Flits with Explicit Sequence Number.
- All 10-bit retry related counters are replaced with 8-bit counters, and the maximum-allowed sequence number is 255 (hence 1023 in all calculations and initial value is replaced by 255).
- REPLAY_TIMEOUT_FLIT_COUNT is a 9-bit counter that saturates at 1FFh.
  - In addition to incrementing REPLAY_TIMEOUT_FLIT_COUNT as described in *PCIe Base Specification*, the count must also be incremented when in the Active state and a Flit Time (Number of Adapter clock cycles (**`lclk`**) required to transfer 256B of data at the current Link speed and width) has elapsed since the last Flit was sent and neither a Payload Flit nor a NOP Flit was transmitted. The counter must be incremented for every Flit Time in which a Flit was not sent (this could lead to it being incremented several times in-between Flits or prior to the limit being met). The added requirement compensates for the noncontinuous transfer of NOP Flits.
  - Replay Schedule Rule 0 of *PCIe Base Specification* must check for REPLAY_TIMEOUT_FLIT_COUNT ≥ 375.
- NAK_WITHDRAWAL_ALLOWED is always set to 0b. Note that this requires implementations to set the flag NAK_SCHEDULED=1b in the "Nak Schedule 0" set of rules.
- IDLE Flit Handshake Phase is not applicable. This is because the transition to Link Active (equivalent to LTSSM being in L0 for PCIe) is managed via handshakes on sideband, and there is no requirement for IDLE Flits to be exchanged. As per PCIe rules, any Flits received with all 0s in the Flit Header bytes are discarded by the Adapter.
- Sequence Number Handshake Phase timeout and exit to Link Retrain is 128 Flits transmitted without exiting Sequence Number Handshake Phase. The Adapter must generate NOP Flits to complete the Sequence Number Handshake Phase in case there are no Payload Flits to send.
- "Prior Flit was Payload" is always set to 1b. This bit does not exist in the Flit Header, and thus from the Retry perspective, implementations must assume that it is always set to 1.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3.9 Runtime Link Testing using Parity

UCIe defines a mechanism to detect Link health during runtime by periodically injecting parity bytes in the middle of the data stream when this mechanism is enabled. The receiver checks and logs parity errors for the inserted parity bytes.

When this mechanism is enabled, the Adapter inserts 64*N Bytes every 256*256*N Bytes of data, where N is obtained from the Error and Link Testing Control register (Field name: Number of 64 Byte Inserts). Software sets this based on the current Link width of operation, including the total number of Active modules interfaced to the Adapter. Only bit 0 of the inserted byte has the parity information which is computed as follows:

ParityByte X, bit 0 = ^((DataByte [X]) ^ (DataByte [X + 64*N]) ^(DataByte [X + 128*N])^....^(DataByte [X + (256*256*N - 64*N)]))

The remaining 7 bits of the inserted byte are Reserved.

The Transmitter and Receiver in the Adapter must keep track of the number of data bytes elapsed to compute or check the parity information. If the RDI state moves away from Active state, the data count and parity is reset, and both sides must renegotiate the enabling of the Parity insertion before next entry to Active from Retrain (if the mechanism is still enabled in the Error and Link Testing Control Register).

This mechanism is enabled by Software writing 1b to the enable bit in the register located in both Adapters across a UCIe Link (see Section 7.5.3.9 for register details). Software must trigger UCIe Link Retrain after writing to the enable bit on both the Adapters. Support for this feature in Raw Format is beyond the scope of this specification and is implementation-dependent. The Adapters exchange sideband messages while the Adapter LSMs are in Retrain to ensure the remote Link partner's receiver is prepared to receive the extra parity bytes in the data stream once the states transition to Active. The Adapter must not request Retrain exit to local RDI until the Parity Feature exchanges are completed. It is permitted to enable it during Initial Link bring up, by using sideband to access the remote Link partner's registers or other implementation specific means; however software must trigger Link Retrain for the feature to take effect.

Adapter sends a {ParityFeature.Req} sideband message to remote Link Partner if its Transmitter is enabled to send parity bytes ("Runtime Link Testing Tx Enable" bit in Section 7.5.3.9). Remote Adapter responds with a {ParityFeature.Ack} sideband message if its receiver is enabled and ready to accept parity bytes ("Runtime Link Testing Rx Enable" bit in Section 7.5.3.9). Figure 3-28 shows an example of a successful negotiation. If Die 0 Adapter Transmitter is enabled to insert parity bytes, it must send a {ParityFeature.Req} from Die 0 to Die 1.

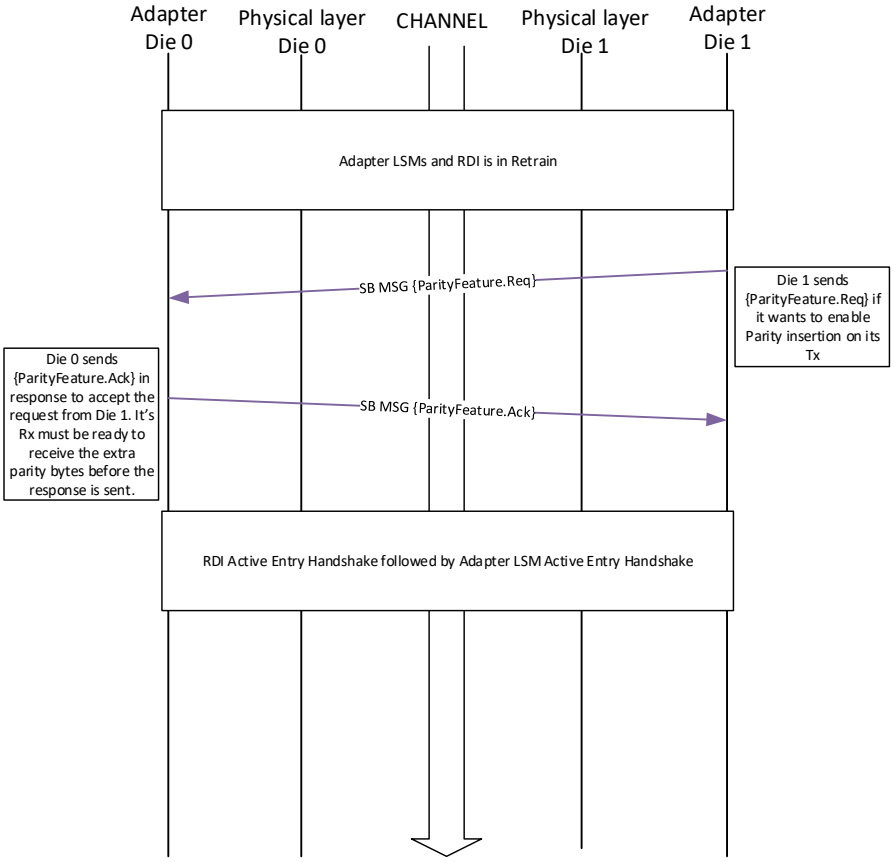
Adapter responds with a {ParityFeature.Nak} if it is not ready to accept parity bytes, or if the feature has not been enabled for it yet. The requesting Adapter must log the Nak in a status register so that Software can determine that a Nak had occurred. Figure 3-29 shows an example of an unsuccessful negotiation.

Note: The Adapters are permitted to transition to a higher latency data path if the Parity Feature is enabled. The explicit Ack/Nak handshake is provided to ensure both sides have sufficient time to transition to alternate data path for this mechanism.

The Parity bytes do not consume Retimer receiver buffer credits. The Retimer receiver must not write the Parity bytes into its receiver buffer or forward these to remote Retimer partner over the Off Package Interconnect. This mechanism is to help characterize local UCIe Links only.

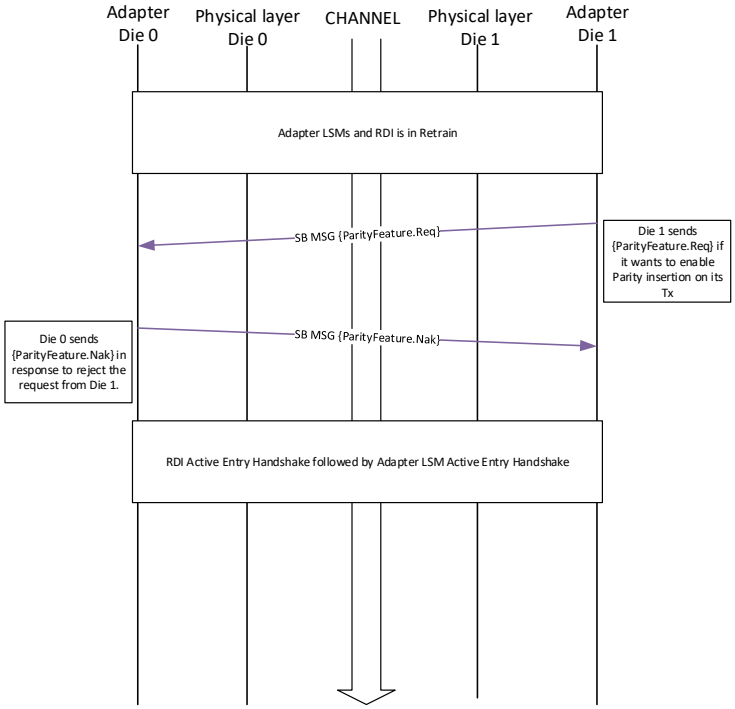
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-28. Successful Parity Feature negotiation between Die 1 Tx and Die 0 Rx



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 3-29. Unsuccessful Parity Feature negotiation between Die 1 Tx and Die 0 Rx



If a parity error is detected by a chiplet, the error is treated as a Correctable error and reported via the correctable error reporting mechanism. By enabling interrupt on correctable errors, SW can implement a BER counter in SW, if so desired.

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.0 Logical Physical Layer

The Logical PHY comprehends the following functions:

- Link initialization, training and power management states
- Byte to Lane mapping for data transmission over Lanes
- Interconnect redundancy remapping (when required)
- Transmitting and receiving sideband messages
- Scrambling and training pattern generation
- Lane reversal
- Width degradation (when applicable)

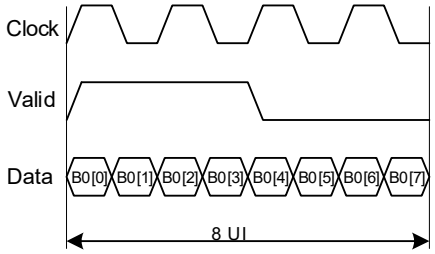
4.1 Data and Sideband Transmission Flow

This specification defines clock, valid and Data to send and receive data over the physical Lanes. The transmitted data is framed by the valid signal.

4.1.1 Byte to Lane Mapping

Data packets are transmitted in Bytes. Within each Byte, bit[0] is transmitted first. Figure 4-1 shows an example of bit arrangement within one byte transmission over Lane 0.

Figure 4-1. Bit arrangement within a byte transfer



Each Byte is transmitted on a separate Lane. Byte 0 (B0) is transmitted on Lane 0, Byte 1 is transmitted on Lane 1 and so on.

Figure 4-2 shows an example of a CXL Latency-Optimized Flit (Format 5) transmitted over a x64 interface (one x64 Advanced Package module or two x32 Advanced Package modules or four Standard Package modules). If the I/O width changes to x32 or x16 interface (Standard Package), transmission of one Byte per Lane is preserved as shown in Figure 4-3 and Figure 4-4 respectively.

Figure 4-5 shows an example for a width degraded Standard Package module.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-2. Byte map for x64 interface

LANE / UI	0	1	2	3	4	5	6	7	...	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0 - 7	B00	B01	B02	B03	B04	B05	B06	B07	...	B48	B49	B50	B51	B52	B53	B54	B55	B56	B57	B58	B59	B60	B61	B62	B63
8 - 15	B64	B65	B66	B67	B68	B69	B70	B71	...	B112	B113	B114	B115	B116	B117	B118	B119	B120	B121	B122	B123	B124	B125	B126	B127
16 - 23	B128	B129	B130	B131	B132	B133	B134	B135	...	B176	B177	B178	B179	B180	B181	B182	B183	B184	B185	B186	B187	B188	B189	B190	B191
24 - 31	B192	B193	B194	B195	B196	B197	B198	B199	...	B240	B241	B242	B243	B244	B245	B246	B247	B248	B249	B250	B251	B252	B253	B254	B255

Figure 4-3. Byte map for x32 interface

Lane / UI	0	1	2	3	4	5	6	7	...	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0 - 7	B0	B1	B2	B3	B4	B5	B6	B7	...	B16	B17	B18	B19	B20	B21	B22	B23	B24	B25	B26	B27	B28	B29	B30	B31
8 - 15	B32	B33	B34	B35	B36	B37	B38	B39	...	B48	B49	B50	B51	B52	B53	B54	B55	B56	B57	B58	B59	B60	B61	B62	B63
16 - 23	B64	B65	B66	B67	B68	B69	B70	B71	...	B80	B81	B82	B83	B84	B85	B86	B87	B88	B89	B90	B91	B92	B93	B94	B95
24 - 31	B96	B97	B98	B99	B100	B101	B102	B103	...	B112	B113	B114	B115	B116	B117	B118	B119	B120	B121	B122	B123	B124	B125	B126	B127
32 - 39	B128	B129	B130	B131	B132	B133	B134	B135	...	B144	B145	B146	B147	B148	B149	B150	B151	B152	B153	B154	B155	B156	B157	B158	B159
40 - 47	B160	B161	B162	B163	B164	B165	B166	B167	...	B176	B177	B178	B179	B180	B181	B182	B183	B184	B185	B186	B187	B188	B189	B190	B191
48 - 55	B192	B193	B194	B195	B196	B197	B198	B199	...	B208	B209	B210	B211	B212	B213	B214	B215	B216	B217	B218	B219	B220	B221	B222	B223
56 - 63	B224	B225	B226	B227	B228	B229	B230	B231	...	B240	B241	B242	B243	B244	B245	B246	B247	B248	B249	B250	B251	B252	B253	B254	B255

Figure 4-4. Byte map for x16 interface

Lane / UI	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 - 7	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
8 - 15	B16	B17	B18	B19	B20	B21	B22	B23	B24	B25	B26	B27	B28	B29	B30	B31
16 - 23	B32	B33	B34	B35	B36	B37	B38	B39	B40	B41	B42	B43	B44	B45	B46	B47
24 - 31	B48	B49	B50	B51	B52	B53	B54	B55	B56	B57	B58	B59	B60	B61	B62	B63
32 - 39	B64	B65	B66	B67	B68	B69	B70	B71	B72	B73	B74	B75	B76	B77	B78	B79
40 - 47	B80	B81	B82	B83	B84	B85	B86	B87	B88	B89	B90	B91	B92	B93	B94	B95
48 - 55	B96	B97	B98	B99	B100	B101	B102	B103	B104	B105	B106	B107	B108	B109	B110	B111
56 - 63	B112	B113	B114	B115	B116	B117	B118	B119	B120	B121	B122	B123	B124	B125	B126	B127
64 - 71	B128	B129	B130	B131	B132	B133	B134	B135	B136	B137	B138	B139	B140	B141	B142	B143
72 - 79	B144	B145	B146	B147	B148	B149	B150	B151	B152	B153	B154	B155	B156	B157	B158	B159
80 - 87	B160	B161	B162	B163	B164	B165	B166	B167	B168	B169	B170	B171	B172	B173	B174	B175
88 - 95	B176	B177	B178	B179	B180	B181	B182	B183	B184	B185	B186	B187	B188	B189	B190	B191
96 - 103	B192	B193	B194	B195	B196	B197	B198	B199	B200	B201	B202	B203	B204	B205	B206	B207
104 - 111	B208	B209	B210	B211	B212	B213	B214	B215	B216	B217	B218	B219	B220	B221	B222	B223
112 - 119	B224	B225	B226	B227	B228	B229	B230	B231	B232	B233	B234	B235	B236	B237	B238	B239
120 - 127	B240	B241	B242	B243	B244	B245	B246	B247	B248	B249	B250	B251	B252	B253	B254	B255

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-5. Byte to Lane mapping for Standard package x8 degraded

Lane	8	9	10	11	12	13	14	15
	or							
UI \ Lane	0	1	2	3	4	5	6	7
0 - 7	B0	B1	B2	B3	B4	B5	B6	B7
8 - 15	B8	B9	B10	B11	B12	B13	B14	B15
16 - 23	B16	B17	B18	B19	B20	B21	B22	B23
24 - 31	B24	B25	B26	B27	B28	B29	B30	B31
32 - 39	B32	B33	B34	B35	B36	B37	B38	B39
40 - 47	B40	B41	B42	B43	B44	B45	B46	B47
48 - 55	B48	B49	B50	B51	B52	B53	B54	B55
56 - 63	B56	B57	B58	B59	B60	B61	B62	B63
64 - 71	B64	B65	B66	B67	B68	B69	B70	B71
72 - 79	B72	B73	B74	B75	B76	B77	B78	B79
80 - 87	B80	B81	B82	B83	B84	B85	B86	B87
88 - 95	B88	B89	B90	B91	B92	B93	B94	B95
96 - 103	B96	B97	B98	B99	B100	B101	B102	B103
104 - 111	B104	B105	B106	B107	B108	B109	B110	B111
112 - 119	B112	B113	B114	B115	B116	B117	B118	B119
120 - 127	B120	B121	B122	B123	B124	B125	B126	B127
128-135	B128	B129	B130	B131	B132	B133	B134	B135
136-143	B136	B137	B138	B139	B140	B141	B142	B143
144-151	B144	B145	B146	B147	B148	B149	B150	B151
152-159	B152	B153	B154	B155	B156	B157	B158	B159
160-167	B160	B161	B162	B163	B164	B165	B166	B167
168-175	B168	B169	B170	B171	B172	B173	B174	B175
176-183	B176	B177	B178	B179	B180	B181	B182	B183
184-191	B184	B185	B186	B187	B188	B189	B190	B191
192-199	B192	B193	B194	B195	B196	B197	B198	B199
200-207	B200	B201	B202	B203	B204	B205	B206	B207
208-215	B208	B209	B210	B211	B212	B213	B214	B215
216-223	B216	B217	B218	B219	B220	B221	B222	B223
224-231	B224	B225	B226	B227	B228	B229	B230	B231
232-239	B232	B233	B234	B235	B236	B237	B238	B239
240-247	B240	B241	B242	B243	B244	B245	B246	B247
248-255	B248	B249	B250	B251	B252	B253	B254	B255

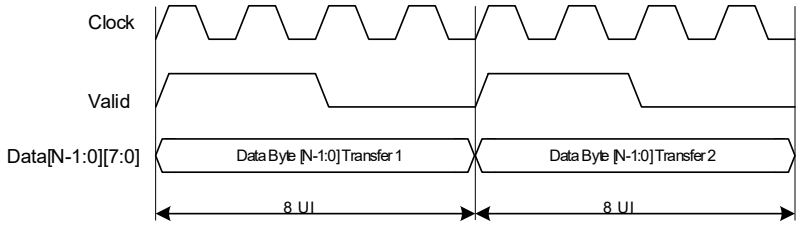
4.1.2 Valid Framing

Valid signal is used to frame the transmitted data. For each 8-bit data packet, valid is asserted for the first 4 UI and de-asserted for 4 UI. This will allow data transfer in Raw Format or various Flit Formats as described in Chapter 3.0 using one or multiple valid frames. An example is shown in Figure 4-6 where Transfer 1 and Transfer 2 can be from the same Flit or different Flits.

Note: The 8-UI block assertion is enforced by the Transmitter and tracked by the Receiver during ACTIVE state.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-6. Valid framing example



4.1.2.1 Valid Framing for Retimers

The UCIe Retimer releases credits to its local UCIe die using the Valid wire, as described in Table 4-1. Each credit tracks 256 Bytes of data (including any FEC, CRC, etc). The Valid Framing encodings ensure triple bit flip detection guarantee.

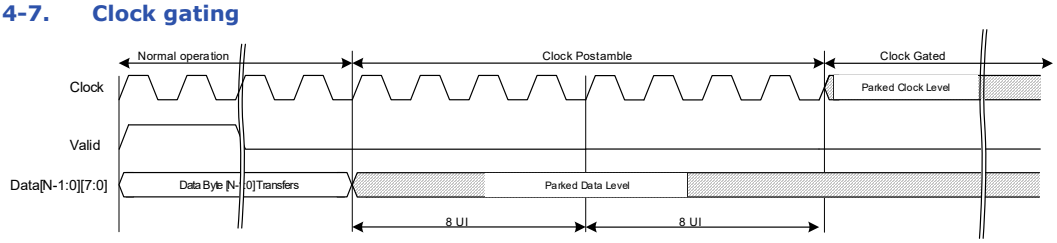
Table 4-1. Valid framing for Retimers

8-UI Valid (LSB first)	Encoding
1111_1111	Flit data transfer valid + 1 Credit release
1111_0000	Flit data transfer valid + no credit release
0000_1111	No Flit data transfer + 1 credit release
0000_0000	No Flit data transfer + no credit release

4.1.3 Clock Gating

Clocks must be gated when Valid signal is low after providing fixed 16 UI (8 cycles) of postamble clock for half-rate clocking and 32 UI (8 cycles) of postamble clock for quarter-rate clocking, unless free running clock mode is negotiated. Data and Clock signal parking levels are described in Section 5.11

Figure 4-7. Clock gating



4.1.4 Free Running Clock Mode

Free running clock mode is defined as the mode where the forwarded clock remains toggling even when the Transmitter for Valid is held low and there is no data transfer on the interface. This mode must be supported to allow disabling dynamic clock gating for normal operation or debug. This must be negotiated prior to mainband Link training through parameter exchange.

4.1.5 Sideband transmission

Each module supports a sideband interface with a serial data and clock pin pair. Sideband packet formats and encodings are shown in Chapter 6.0 and the electrical characteristics are shown in Section 5.13.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

As shown in Section 6.1.2, the sideband message formats are defined as a 64-bit header with 32 bits or 64 bits of data. A 64-bit serial packet is defined on the I/O interface to the remote die as shown in Figure 4-8. 32-bit data is sent using the 64-bit serial packet with MSBs padded with 0b. Two sideband serial packets on the I/O interface are separated by a minimum of 32 bits low as shown in Figure 4-9. A sideband message with data would be transferred as a 64-bit header followed by 32 bits of low followed by 64-bit data followed by 32 bits of low.

Figure 4-8. A 64 bit Sideband serial packet transfer example

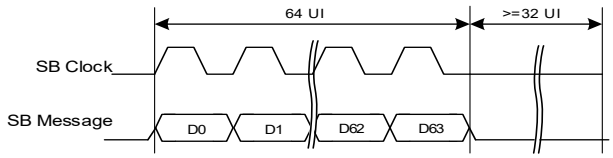
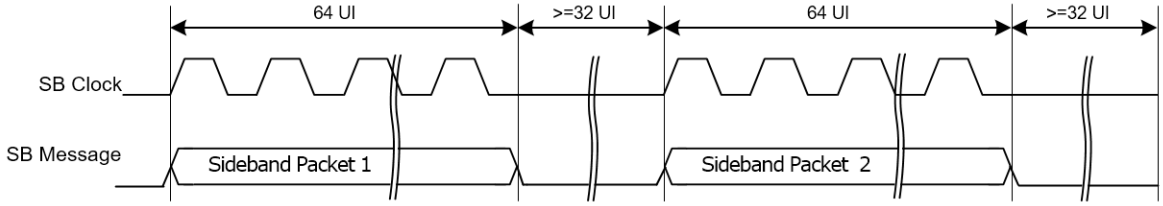


Figure 4-9. Sideband packet transmission: back to back



4.2 Lane Reversal

In Section 4.2, Section 4.3, and Section 4.5, the following nomenclature is used:

- **TD_P**: Physical Lane for Data Transmitter
- **RD_P**: Physical Lane for Data Receiver
- **TRD_P**: Physical Lane for Redundant Data Transmitter
- **RRD_P**: Physical Lane for Redundant Data Receiver
- **TD_L**: Logical Lane for Data Transmit
- **RD_L**: Logical Lane for Data Receive
- **TCKP_P**, **TCKN_P** and **TTRK_P**: Physical Lane for Clock and Track Transmitter
- **TCKP_L**, **TCKN_L** and **TTRK_L**: Logical Lane for Clock and Track Transmitter
- **RCKP_P**, **RCKN_P** and **RTRK_P**: Physical Lane for Clock and Track Receiver
- **RCKP_L**, **RCKN_L** and **RTRK_L**: Logical Lane for Clock and Track Receiver
- **TRDCK_P**: Physical Lane for Redundant Clock/Track Transmitter
- **RRDCK_P**: Physical Lane for Redundant Clock/Track Receiver
- **TRDCK_L**: Logical Lane for Redundant Clock/Track Transmitter
- **RRDCK_L**: Logical Lane for Redundant Clock/Track Receiver
- **TVLD_P**, **RVLD_P**: Physical Lane for Valid Transmitter and Receiver
- **TRDVLD_P**, **RRDVLD_P**: Physical Lane for Redundant Valid Transmitter and Receiver

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- **TVLD_L**, **RVLD_L**: Logical Lane for Valid Transmitter and Receiver
- **TRDVLD_L**, **RRDVLD_L**: Logical Lane for Redundant Valid Transmitter and Receiver

Devices must support Lane reversal of data Lanes within a Module. An example of Lane reversal is when physical Data Lane 0 on local die is connected to physical Data Lane (N-1) on the remote die (physical Data Lane 1 is connected to physical Data Lane N-2 and so on) where N = 16 for Standard Package, N = 64 for a x64 Advanced Package, and N=32 for a x32 Advanced Package.Redundant Lanes, in case of Advanced Package, are also reversed. Lane reversal must be implemented on the Transmitter only. The Transmitter reverses the logical Lane order on Data and Redundant Lanes.

Track, Valid, Clock, and sideband signals must not be reversed.

Lane reversal is discovered and applied during initialization and training (see Section 4.5.3.3.5).

4.2.1 Lane ID

To allow Lane reversal discovery, each logical Data and redundant Lane within a module is assigned a unique Lane ID. The assigned Lane IDs are shown in Table 4-2 and Table 4-3 for Advanced and Standard Package modules, respectively. Note that logical Lane numbers in Table 4-2 and Table 4-3 represent the logical Transmitter and Receiver Lanes. For example, Logical Lane Number = 0 represents **TD_L[0]/RD_L[0]** and so on.

In Table 4-2, for a x64 Advanced Package module, logical Lane numbers 64, 65, 66, and 67 represent Logical redundant Lanes **TRD_L[0]/RRD_L[0]**, **TRD_L[1]/RRD_L[1]**, **TRD_L[2]/RRD_L[2]**, **TRD_L[3]/RRD_L[3]**, respectively. For a x32 Advanced Package module, the Lane ID for TD_L[0:31] / RD_L[0:31], TRD_L[0]/RRD_L[0] and TRD_L[1]/RRD_L[1] will be represented by the set of Lane ID {0…31, 64, 65} respectively.

Table 4-2. Lane ID: Advanced Package module (Sheet 1 of 2)

Logical Lane Number	Lane ID	Logical Lane Number	Lane ID
0	8b'00000000	34	8b'00100010
1	8b'00000001	35	8b'00100011
2	8b'00000010	36	8b'00100100
3	8b'00000011	37	8b'00100101
4	8b'00000100	38	8b'00100110
5	8b'00000101	39	8b'00100111
6	8b'00000110	40	8b'00101000
7	8b'00000111	41	8b'00101001
8	8b'00001000	42	8b'00101010
9	8b'00001001	43	8b'00101011
10	8b'00001010	44	8b'00101100
11	8b'00001011	45	8b'00101101
12	8b'00001100	46	8b'00101110
13	8b'00001101	47	8b'00101111
14	8b'00001110	48	8b'00110000
15	8b'00001111	49	8b'00110001
16	8b'00010000	50	8b'00110010
17	8b'00010001	51	8b'00110011

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 4-2. Lane ID: Advanced Package module (Sheet 2 of 2)

Logical Lane Number	Lane ID	Logical Lane Number	Lane ID
18	8b'00010010	52	8b'00110100
19	8b'00010011	53	8b'00110101
20	8b'00010100	54	8b'00110110
21	8b'00010101	55	8b'00110111
22	8b'00010110	56	8b'00111000
23	8b'00010111	57	8b'00111001
24	8b'00011000	58	8b'00111010
25	8b'00011001	59	8b'00111011
26	8b'00011010	60	8b'00111100
27	8b'00011011	61	8b'00111101
28	8b'00011100	62	8b'00111110
29	8b'00011101	63	8b'00111111
30	8b'00011110	64	8b'01000000
31	8b'00011111	65	8b'01000001
32	8b'00100000	66	8b'01000010
33	8b'00100001	67	8b'01000011

Table 4-3. Lane ID: Standard Package Module

Logical Lane Number	Lane ID	Logical Lane Number	Lane ID
0	8b'00000000	8	8b'00001000
1	8b'00000001	9	8b'00001001
2	8b'00000010	10	8b'00001010
3	8b'00000011	11	8b'00001011
4	8b'00000100	12	8b'00001100
5	8b'00000101	13	8b'00001101
6	8b'00000110	14	8b'00001110
7	8b'00000111	15	8b'00001111

4.3 Interconnect redundancy remapping

As discussed in Section 5.9, Advanced Package modules require redundancy remapping to recover from faulty Lanes. This section provides the details of remapping. After a successful remapping/repair, any unused repair Lanes must have their Transmitters tri-stated and Receivers disabled.

4.3.1 Data Lane repair

The specification supports remapping (repair) of up to two data Lanes for each group of 32 data Lanes. **TD_P[31:0] (RD_P[31:0])** and **TD_P[63:32] (RD_P[63:32])** are treated as two separate groups of 32 Lanes that can be independently repaired using redundant Lanes, **TRD_P[1:0](RRD_P[1:0])** and **TRD_P[3:2](RRD_P[3:2])**, respectively. **TD_P[63:32]** (**RD_P[63:32]**) and hence **TRD_P[3:2]** (**RRD_P[3:2]**) do not apply to x32 Advanced Package Link.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

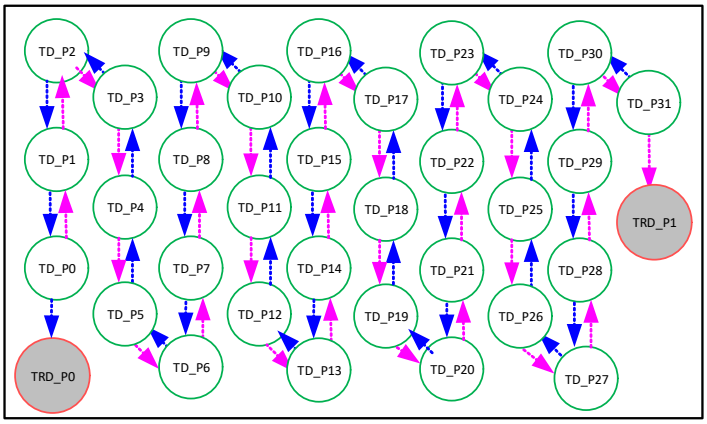
Lane remapping is accomplished by "shift left" or "shift right" operation. A "shift left" is when data traffic of logical Lane **TD_L[n]** on **TD_P[n]** is multiplexed onto **TD_P[n-1]**. A shift left puts TD_L[0] onto **TRD_P0** or **TD_L[32]** onto **TRD_P2**. A shift right operation is when data traffic **TD_L[n]** is multiplexed onto **TD_P[n+1]**. A shift right puts **TD_L[31]** onto **TRD_P1** or **TD_L[63]** onto **TRD_P3**. Refer to the pseudo codes in Section 4.3.3.1 and Section 4.3.3.2 that show the changes in mapping post repair.

After a data Lane is remapped, the Transmitter associated with the broken physical Lane is tri-stated and the Receiver is disabled. The Transmitter and the Receiver of the redundant Lane used for the repair are enabled.

Figure 4-10 shows transmit bump side of data Lane remapping for the first group of 32 Lanes. Both "shift left" and "shift right" remapping is needed to optimally repair up to any two Lanes within the group. Figure 4-11 shows details of the mux structure used for data Lane repair.

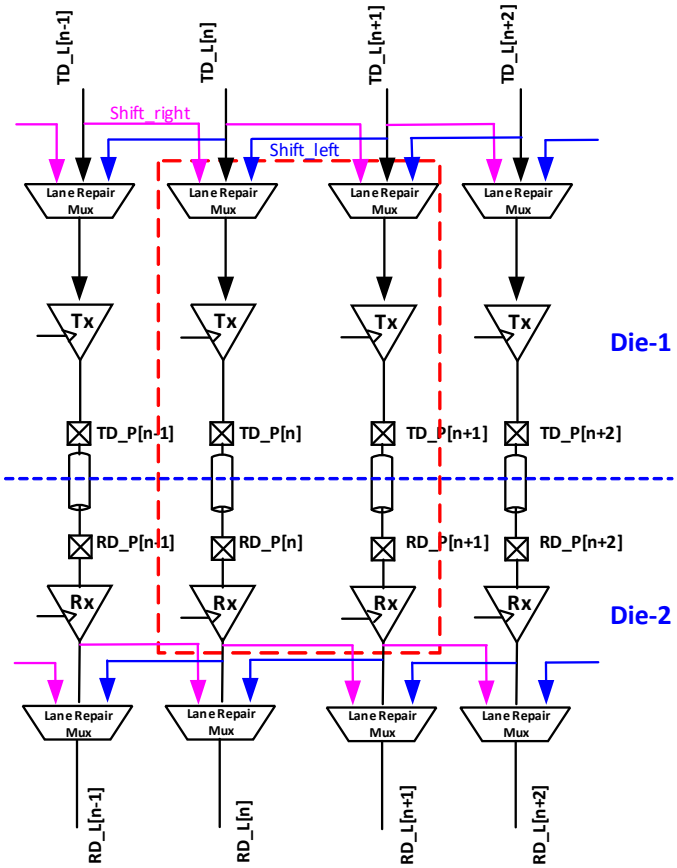
Note: Example repair implementations are shown for **TD_P[31:0]** for clarity. It should be noted that the same schemes are also applicable to **TD_P[63:32]**.

Figure 4-10. Data Lane remapping possibilities to fix potential defects



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-11. Data Lane remapping: Mux chain



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.2 Data Lane repair with Lane reversal

If Lanes are reversed, physical Lane 0 of Transmitter (**TD_P[0]**) is connected to physical Lane N-1 of the Receiver (**RD_P[N-1]**). N is 64 or 32 for x64 or x32 Advanced Package modules, respectively. N is 16 for Standard Package Modules. Refer to the pseudo codes in Section 4.3.3.3 and Section 4.3.3.4 that show the changes in mapping post repair.

4.3.3 Data Lane repair implementation

4.3.3.1 Single Lane repair

TRD_P[0](RRD_P[0]) must be used as the redundant Lane to remap any single physical Lane failure for **TD_P[31:0](RD_P[31:0])**. **TRD[2](RRD[2])** must be used as the redundant Lane to remap any single Lane failure for **TD_P[63:32] (RD_P[63:32])**.

Pseudo code for repair in **TD_P[31:0](RD_P[31:0])** (0<= x <=31):

```
IF failure occurs in TD_P[x]:
       IF x > 0:
              FOR 0 <= i < x:
                     TD_P[x-i-1] = TD_L[x-i]
                     RD_L[x-i] = RD_P[x-i-1]
       TRD_P[0] = TD_L[0]
       RD_L[0] = RRD_P[0]
```

Pseudo code for repair in **TD_P[63:32](RD_P[63:32])** (32<= x <=63) (this does not apply to x32 Advanced Package Link):

```
IF failure occurs in TD_P[x]:
       IF x > 32:
              FOR 0 <= i < x-32:
                     TD_P[x-i-1] = TD_L[x-i]
                     RD_L[x-i] = RD_P[x-i-1]
       TRD_P[2] = TD_L[32]
       RD_L[32] = RRD_P[2]
```


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

As shown in Figure 4-12 **TD_P[29]** is remapped in the direction to use **TRD_P[0]** as the repair resource. Figure 4-13 shows the circuit implementation.

Figure 4-12. Single Lane failure remapping

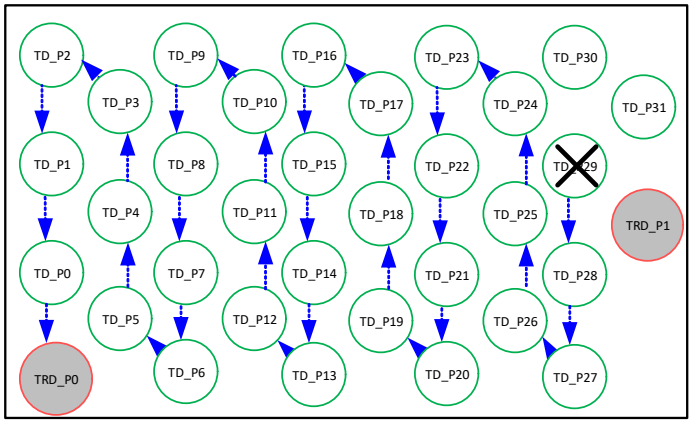
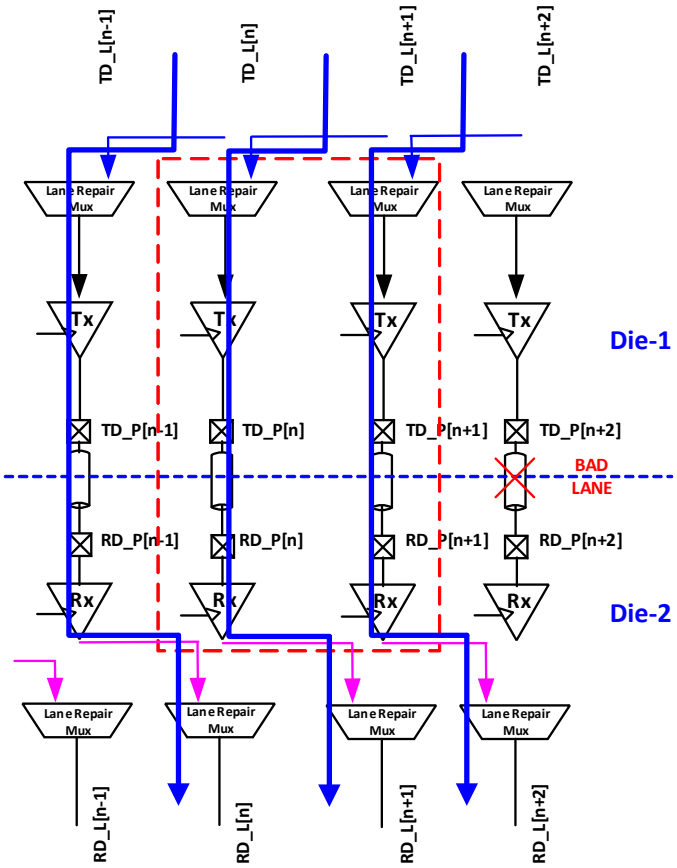


Figure 4-13. Single Lane remapping implementation



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.3.2 Two Lane repair

Any two Lanes within a group of 32 can be repaired using the two redundant bumps. For any two physical Lane failures in **TD_P[31:0] (RD_P[31:0])**, the lower Lane must be remapped to **TRD_P[0](RRD_P[0])** and the upper Lane is remapped to **TRD_P[1](RRD_P[1])**. For any two physical Lane failures in **TD_P[63:32] (RD_P[63:31])**, the lower Lane must be remapped to **TRD_P[2](RRD_P[2])** and the upper Lane is remapped to **TRD_P[3](RRD_P[3])**.

Pseudo code for two Lane repair in **TD_P[31:0](RD_P[31:0])** (0<= x,y <=31):

```
IF failure occurs in TD_P[x], TD_P[y] AND (x < y):
      IF x > 0:
            FOR 0 <= i < x:
                  TD_P[x-i-1] = TD_L[x-i]
                  RD_L[x-i] = RD_P[x-i-1]
      TRD_P[0] = TD_L[0]
      RD_L[0] = RRD_P[0]
      IF y < 31:
            FOR 0 <= j < (31-y):
                  TD_P[y+j+1] = TD_L[y+j]
                  RD_L[y+j]=RD_P[y+j+1]
      TRD_P[1] = TD_L[31]
      RD_L[31]=RRD_P[1]
```

Pseudo code for two Lane repair in **TD_P[63:32] (RD_P[63:32])** (32<= x,y <=63) (this does not apply to x32 Advanced Package Link):

```
IF failure occurs in TD_P[x], TD_P[y] AND (x < y):
      IF x > 32:
            FOR 0 <= i < x-32:
                  TD_P[x-i-1] = TD_L[x-i]
                  RD_L[x-i] = RD_P[x-i-1]
      TRD_P[2] = TD_L[32]
      RD_L[32] = RRD_P[2]
      IF y < 63:
            FOR 0 <= j < (63-y):
                  TD_P[y+j+1] = TD_L[y+j]
                  RD_L[y+j] = RD_P[y+j+1]
      TRD_P[3] = TD_L[63]
      RD_L[63] = RRD_P[3]
```


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Shown in Figure 4-14 is an example of two (physical Lanes 25 and 26) Lane remapping. Figure 4-15 shows the circuit implementation. Both Transmitter and Receiver must apply the required remapping.

Figure 4-14. Two Lane failure remapping

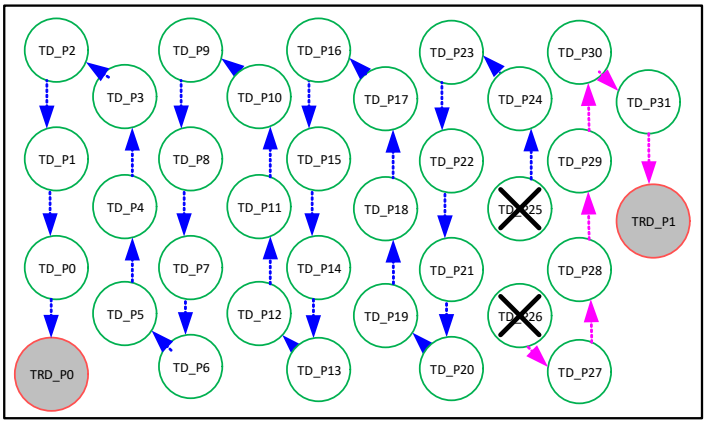
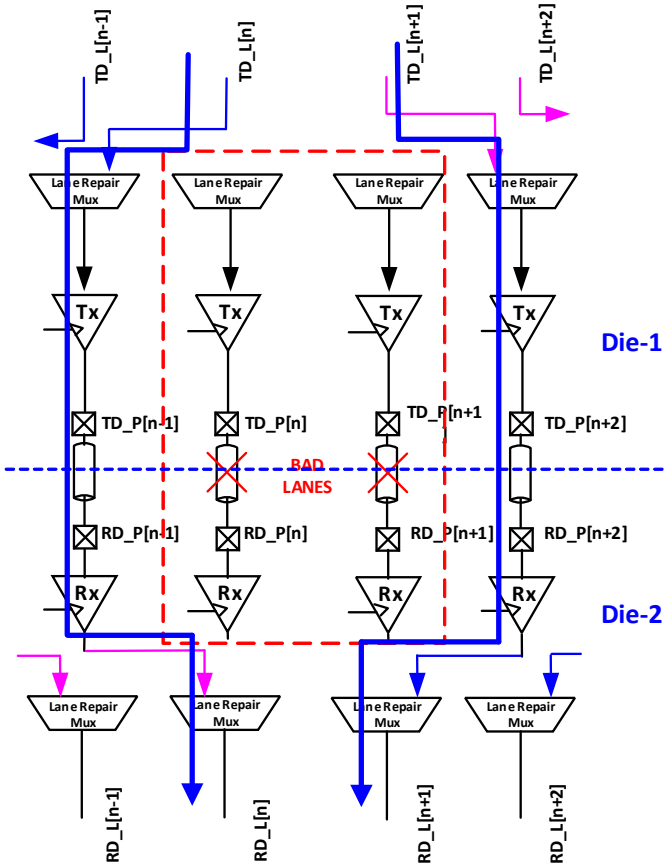


Figure 4-15. Two Lane remapping implementation



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.3.3 Single Lane repair with Lane reversal

For repair with Lane reversal (see Section 4.3.2) Transmitter side remapping is reversed to preserve shifting order for Receiver side remapping.

4.3.3.3.1 x64 Advanced Package Pseudo Codes

Pseudo code for one Lane failure in **TD_P[31:0](RD_P[32:63])** (0<= x <=31):

```
IF failure occurs in TD_P[x]:
        IF x > 0:
              FOR 0 <= i < x:
                    TD_P[x-i-1] = TD_L[63-x+i]
                    RD_L[63-x+i] = RD_P[63-x+i+1]
        TRD_P[0] = TD_L[63]
        RD_L[63] = RRD_P[3]
```

Pseudo code for one Lane failure in **TD_P[63:32](RD_P[0:31])** (32<= x <=63):

```
IF failure occurs in TD_P[x]:
      IF x > 32:
            FOR 0 <= i < x-32:
                  TD_P[x-i-1] = TD_L[63-x+i]
                  RD_L[63-x+i] = RD_P[63-x+i+1]
      TRD_P[2] = TD_L[31]
      RD_L[31] = RRD_P[1]
```

4.3.3.3.2 x32 Advanced Package Pseudo Codes

Pseudo code for one-Lane failure in **TD_P[31:0](RD_P[0:31])** (0<= x <=31):

```
IF failure occurs in TD_P[x]:
        IF x > 0:
              FOR 0 <= i < x:
                    TD_P[x-i-1] = TD_L[31-x+i]
                    RD_L[31-x+i] = RD_P[31-x+i+1]
        TRD_P[0] = TD_L[31]
        RD_L[31] = RRD_P[1]
```


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.3.4 Two Lane repair with Lane reversal

For repair with Lane reversal (see Section 4.3.2) Transmitter side remapping is reversed to preserve shifting order for Receiver side remapping.

4.3.3.4.1 x64 Advanced Package Pseudo Codes

Pseudo code for two Lane failure in **TD_P[31:0](RD_P[32:63])** (0<= x <=31):

```
IF failure occurs in TD_P[x], TD_P[y] AND (x < y):
       IF x > 0:
              FOR 0 <= i < x:
                     TD_P[x-i-1] = TD_L[63-x+i]
                     RD_L[63-x+i] = RD_P[63-x+i+1]
       TRD_P[0] = TD_L[63]
       RD_L[63] = RRD_P[3]
       IF y < 31:
              FOR 0 <= j < (31-y):
                     TD_P[y+j+1] = TD_L[63-y-j]
                     RD_L[63-y-j] = RD_P[63-y-(j+1)]
       TRD_P[1] = TD_L[32]
       RD_L[32] = RRD_P[2]
```

Pseudo code for two-Lane failure in **TD_P[63:32](RD_P[0:31])** (32<= x <=63):

```
IF failure occurs in TD_P[x], TD_P[y] AND (x < y):
       IF x > 32:
              FOR 0 <= i < x-32:
                     TD_P[x-i-1] = TD_L[63-x+i]
                     RD_L[63-x+i] = RD_P[63-x+(i+1)]
       TRD_P[2] = TD_L[31]
       RD_L[31] = RRD_P[1]
       IF y < 63:
              FOR 0 <= j < (63-y):
                     TD_P[y+j+1] = TD_L[63-y-j]
                     RD_L[63-y-j] = RD_P[63-y-(j+1)]
       TRD_P[3] = TD_L[0]
       RD_L[0] = RRD_P[0]
```


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.3.4.2 x32 Advanced Package Pseudo Codes

Pseudo code for two-Lane failure in **TD_P[31:0](RD_P[0:31])** (0<= x <=31):

```
IF failure occurs in TD_P[x], TD_P[y] AND (x < y):
        IF x > 0:
              FOR 0 <= i < x:
                    TD_P[x-i-1] = TD_L[31-x+i]
                    RD_L[31-x+i] = RD_P[31-x+i+1]
        TRD_P[0] = TD_L[31]
        RD_L[31] = RRD_P[1]
        IF y < 31:
              FOR 0 <= j < (31-y):
                    TD_P[y+j+1] = TD_L[31-y-j]
                    RD_L[31-y-j] = RD_P[31-y-(j+1)]
        TRD_P[1] = TD_L[0]
        RD_L[0] = RRD_P[0]
```


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.4 Clock and Track Lane remapping

The specification supports remapping of one broken Lane for **TCKP_P/RCKP_P**, **TCKN_P/RCKN_P** and **TTRK_P/RTRK_P** physical Lanes. The repair scheme is shown in Figure 4-16. Clock Lane remapping allows repair of single Lane failure for both differential and pseudo-differential implementation of the clock Receiver. The circuit details are shown in Figure 4-17 and Figure 4-18 for differential and pseudo-differential clock Receivers respectively.

After a Lane is remapped, the Transmitter is tri-stated. The Receiver of the physical redundant (**RRDCK_P**) Lane is disabled.

Figure 4-16. Clock and Track repair

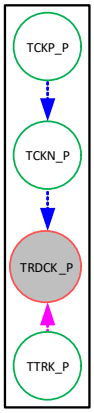


Figure 4-17. Clock and track repair: Differential Rx

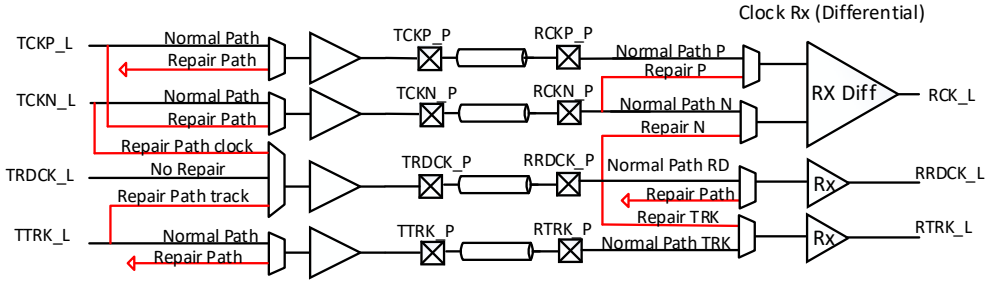
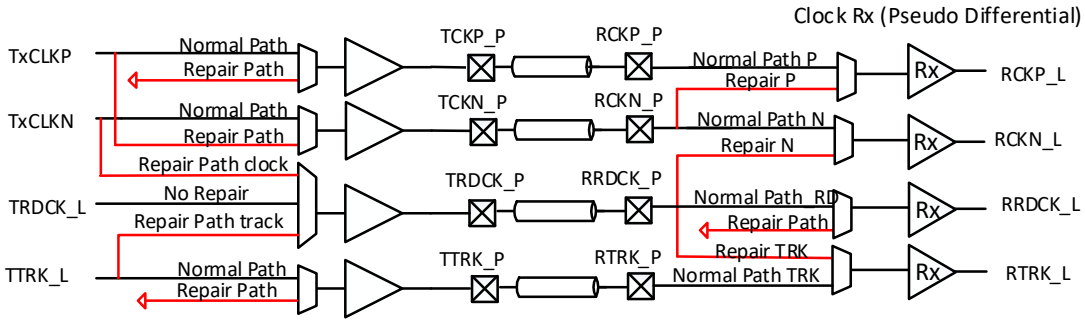


Figure 4-18. Clock and track repair: Pseudo Differential Rx



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.3.5 Clock and Track Lane repair implementation

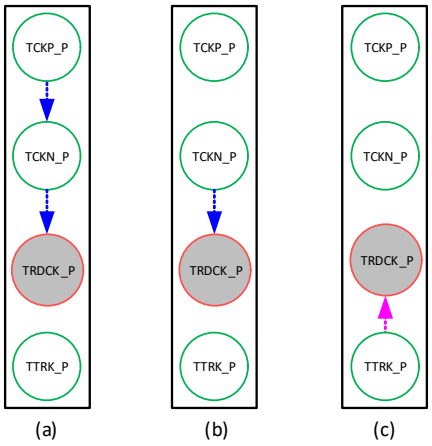
Pseudo code for Clock and Track Lane repair:

```
IF failure occurs in TCKP_P:
       TCKN_P = TCKP_L AND TRDCK_P = TCKN_L
       RCKP_L = RCKN_P AND RCKN_L = RRDCK_P
ELSE IF failure occurs on TCKN_P:
       TRDCK_P = TCKN_L
       RCKN_L = RRDCK_P
ELSE IF failure occurs in TTRK_P:
       TRDCK_P = TTRK_L
       RTRK_L = RRDCK_P
```

The implementation of Clock and Track Lane remapping is shown in Figure 4-19(a), Figure 4-19(b) and Figure 4-19(c) respectively. The corresponding circuit level details of remapping implementation are shown in Figure 4-20, Figure 4-21 and Figure 4-22.

Note that the both Transmitter and Receiver on **CKRD** Lane are required during detection phase and can be tri-stated and turned off if not used for repair.

Figure 4-19. Clock and Track Lane repair scheme



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-20. Clock and track repair: CKP repair

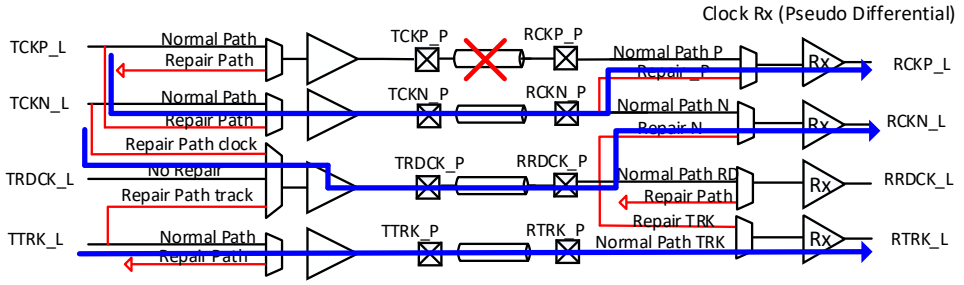


Figure 4-21. Clock and track repair: CKN repair

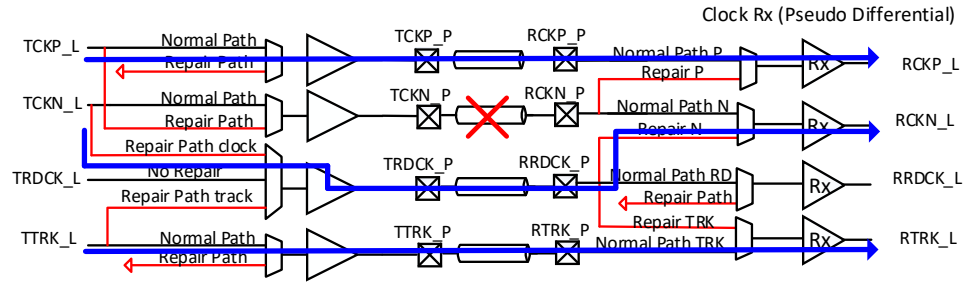
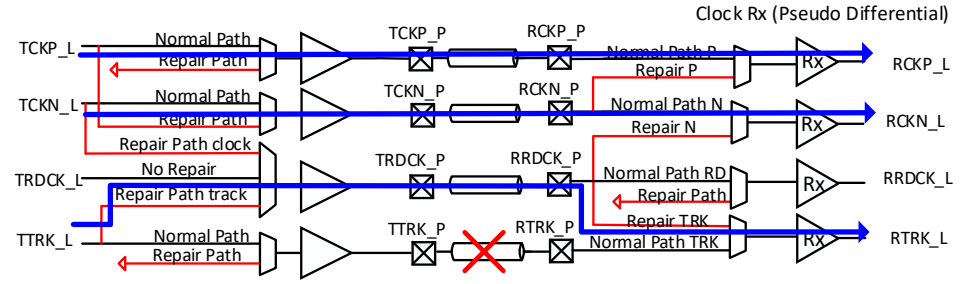


Figure 4-22. Clock and track repair: Track repair



4.3.6 Valid Repair and implementation

Valid Lane has a dedicated redundant Lane. If a failure is detected on the Valid physical Lane, redundant Valid physical Lane is used to send Valid. Valid failure detection and repair is performed during Link initialization and Training (Section 4.5.3.3.4).

Figure 4-23 shows the normal path for Valid and redundant valid Lanes. Figure 4-24 shows the repair path for Valid Lane failure.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-23. Valid repair: Normal Path

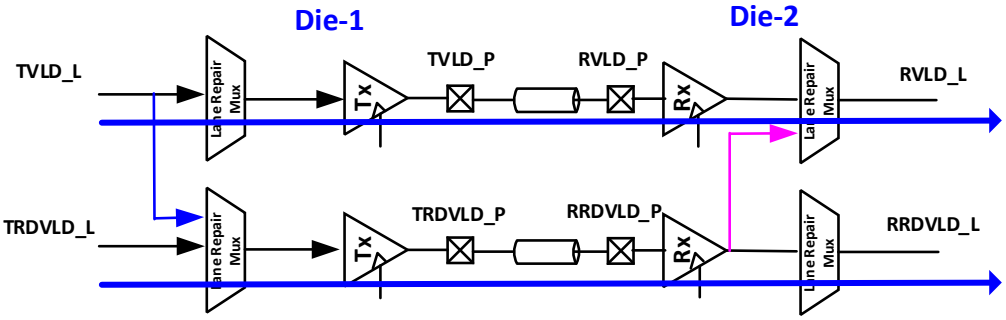
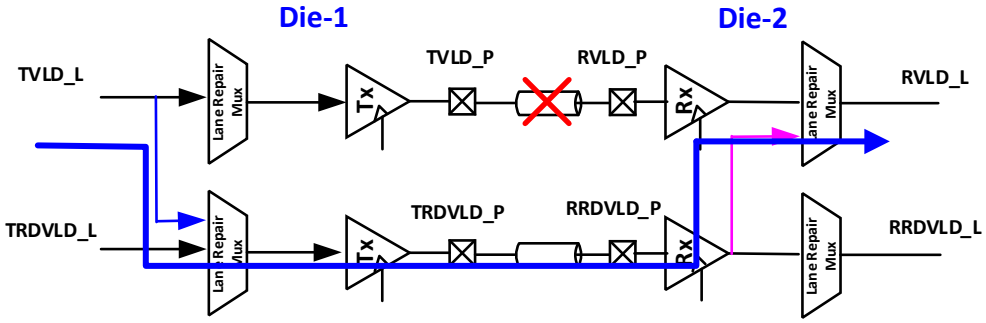


Figure 4-24. Valid Repair: Repair Path



4.3.7 Width Degrade in Standard Package Interfaces

In case of Standard Package x16 modules where Lane repair is not supported resilience against faulty Lanes is provided by configuring the Link to a x8 width (logical Lanes 0 to 7 or Logical Lanes 8 to 15 which exclude the faulty Lanes). For example, if one or more faulty Lanes are in logical Lane 0 to 7, the Link is configured to x8 width using logical Lanes 8 to 15. The configuration is done during Link initialization or retraining. Transmitters of the disabled Lanes go to Hi-Z and Receivers are disabled.

Figure 4-5 shows the byte to Lane mapping for a width degraded x8 interface

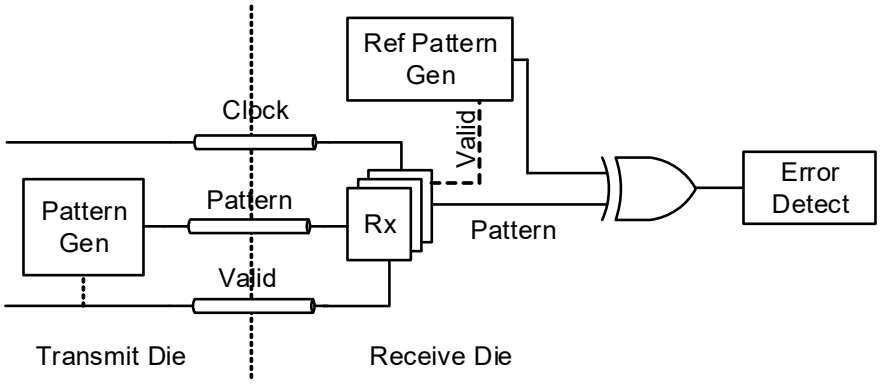
4.4 Data to Clock Training and Test Modes

Note: Sideband commands will be identified as {SB command}.

Figure 4-25 shows the infrastructure for interface training and testing. The Transmit Die and Receive Die implement the same Linear Feedback Shift Register (LFSR) described in Section 4.4.1. The pattern sent from the Transmitter along with forwarded clock and Valid is compared with locally generated reference pattern. Both transmit and receive pattern generators must start and advance in sync. The compare circuitry checks for matching data each UI. Any mismatch between the received pattern and pattern predicted by the local pattern generator is detected as an error.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

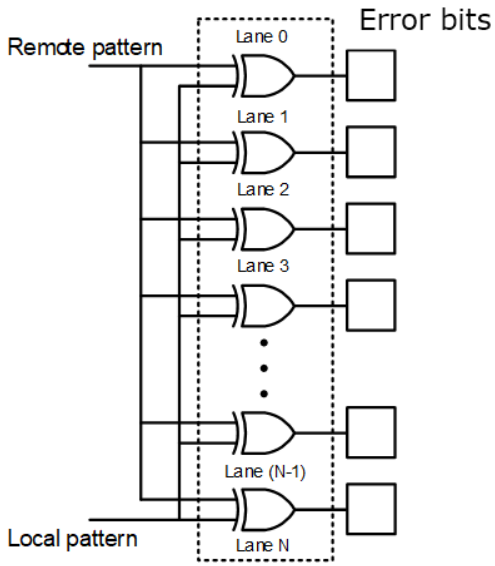
Figure 4-25. Test and training logic



The receive die must implement two types of comparison schemes

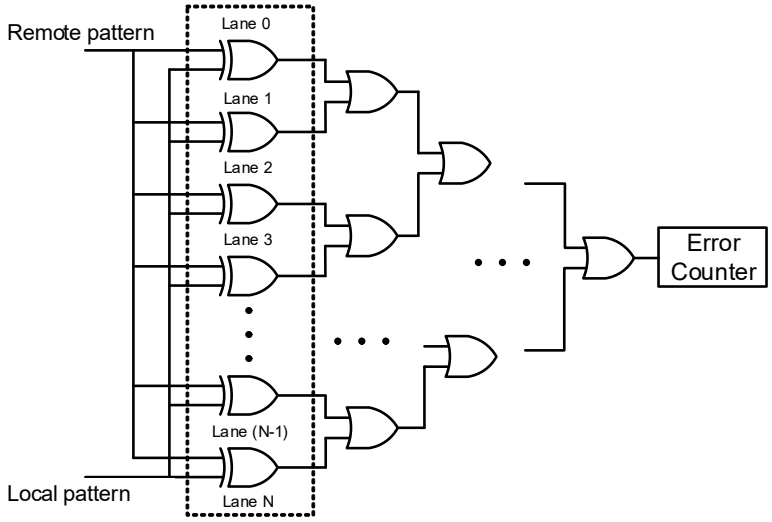
- **Per-Lane comparison**: Per-Lane comparison is used to identify the number of failing Lanes between the two dies. Any mismatch, above the set threshold between the received pattern and the expected pattern on each Lane, will set an internal error detect bit for each Lane. Once a pattern mismatch on a particular Lane is found, this error bit is set for the remainder of the test. Figure 4-26 illustrates Per-Lane comparison mode. This mode will indicate per-Lane errors within the module (N = 68 (64 + 4 RD) for a x64 Advanced Package Module, N=34 (32 + 2 RD) for a x32 Advanced Package Module and N=16 for a Standard Package Module, respectively). The per-Lane comparison results can be read via sideband.
- **Aggregate comparison**: In this mode, pattern mismatches each UI on any Lane within the module are accumulated into a 16-bit error counter. The Lane errors are ORed to generate a module-level error and counted as shown in Figure 4-27. This scheme can be used for module level margin and BER.

Figure 4-26. Lane failure detection



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-27. All Lane error detection



4.4.1 Scrambling and training pattern generation

A Linear feedback shift register (LFSR) is defined for scrambling and test pattern generation.

The LFSR uses the same polynomial as PCIe: $G(X)=X^{23} + X^{21} + X^{16} + X^{8} + X^{5} + X^{2} + 1$. Each Transmitter is permitted to implement a separate LFSR for scrambling and pattern generation. Each Receiver is permitted to implement a separate LFSR using the same polynomial for de-scrambling and pattern comparison. The implementation is shown in Figure 4-28. The seed of the LFSR is Lane dependent and the seed value for Lane number is modulo 8 as shown in Table 4-4.

Alternatively, implementations can choose to implement one LFSR with different tap points for multiple Lanes as shown in Figure 4-29. This is equivalent to individual LFSR per-Lane with different seeds.

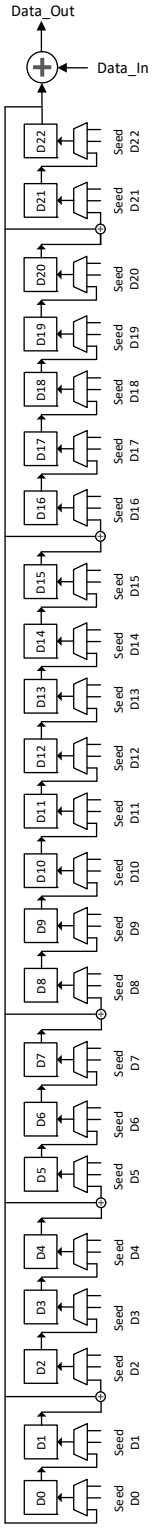
Table 4-4. LFSR seed values

Lane	Seed
0	23'h1DBFBC
1	23'h 0607BB
2	23'h1EC760
3	23'h18C0DB
4	23'h010F12
5	23'h19CFC9
6	23'h0277CE
7	23'h1BB807
RD0, RD2[1]	23'h18C0DB
RD1, RD3[2]	23'h010F12

1. Same as Lane 3.
2. Same as Lane 4.

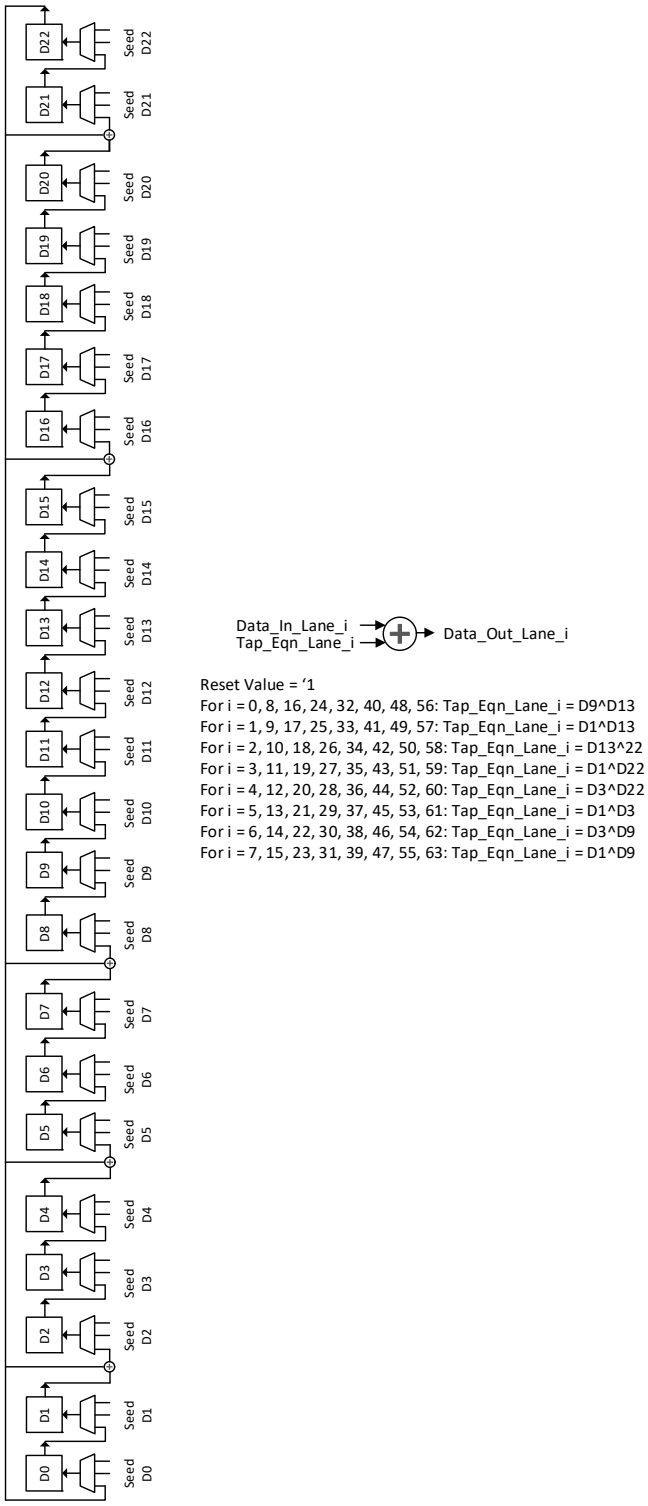
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-28. LFSR implementation



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-29. LFSR alternate implementation



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5 Link Initialization and Training

Link initialization and training are described at a Module level. The description of terms are used in this section are as follows:

- UCIe Module Partner: A UCIe Module Partner is the corresponding UCIe Module on the remote UCIe Die to which the UCIe Module connects. For example, if two UCIe Dies A and B are connected in the standard Die rotate configuration by a 2-module UCIe Link, Modules 0 and 1; UCIe Die A's Module 0 (UCIe Module A0) is connected to UCIe Die B's Module 1 (UCIe Module B1); then A0 is the UCIe Module Partner of B1 and vice-versa.
- Phase Interpolator (PI) in this specification is used to refer any method of generating and selecting sampling clock phase.
- Timeout: Every state except RESET and TRAINERROR in the Link Training State Machine has a residency timeout of 8ms. For states with sub-states, the timeout is per sub-state (i.e., if Physical Layer is in a state for greater than 8ms, then it transitions to TRAINERROR and eventually to RESET). Physical Layer sideband handshakes for RDI state transitions with remote Link partner also timeout after 8ms. The timeout counters must be reset if a sideband message with "Stall" encoding is received. All timeout values specified are -0% and +50% unless explicitly stated otherwise. All timeout values must be set to the specified values after Domain Reset exit. All counter values must be set to the specified values after Domain Reset.
- Electrical idle is described in Section 5.12.

When training during Link Initialization (i.e., Physical Layer transitions out of RESET state), hardware is permitted to attempt training multiple times. The triggers for initiating Link Training are:

- Software writes 1b to Start UCIe Link Training bit in UCIe Link Control
- Adapter triggers Link Training on RDI (RDI status is Reset and there is a NOP to Active transition on the state request)
- SBINIT pattern (two consecutive iterations of 64 UI clock pattern and 32 UI low) is observed on any sideband Receiver clock/data pair

Once hardware fails training after an implementation specific number of attempts, it must transition to RESET and wait for a subsequent Link Training trigger. Physical Layer must escalate a fatal error on RDI if Software triggered or RDI triggered Link training fails or there is a Link up to Link down transition due to a Physical Layer timeout.

4.5.1 Link Training basic operations

Some basic operations in Link training are defined in this section. These will be used in mainband initialization, training and margining.

4.5.1.1 Transmitter initiated Data to Clock point test

In this mode, the Transmitter initiates the data to clock training on all Lanes in the module at a single PI phase. When not performing the actions relevant to this state:

- Data, Valid, and Track Transmitters drive low
- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Data, Valid, and Clock Receivers are enabled
- Track Receiver is permitted to be disabled

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

The sequence of steps for this test are as follows for each UCIe Module of the UCIe Link:

1. The UCIe Module sets up the Transmitter parameters (shown in Table 4-5), sends a {Start Tx Init D to C point test req} sideband message to its UCIe Module Partner, and waits for a response. The data field of this message includes the required parameters, shown in Table 4-5. The Receiver on the UCIe Module Partner must enable the pattern comparison circuits to compare incoming mainband data to the locally generated expected pattern. Once the data to clock training parameters for its Receiver are setup, the UCIe Module Partner responds with a {Start Tx Init D to C point test resp} sideband message.
2. The UCIe Module resets the LFSR (scrambler) on its mainband Transmitters and sends sideband message {LFSR clear error req}. The UCIe Module Partner resets the LFSR and clears any prior compare results on its mainband Receivers and responds with {LFSR clear error resp} sideband message.
3. The UCIe Module sends the pattern (selected through "Tx Pattern Generator Setup") for the selected number of cycles ("Tx Pattern Count Setup") on its mainband Transmitter.
4. The UCIe Module Partner performs the comparison on its Receivers for each UI during the pattern transmission based on "Rx compare setup" and logs the results.
5. The UCIe Module requests its UCIe Module Partner for the logged results in Step 4 by sending {Tx Init D to C results req} sideband message. The UCIe Module Partner stops comparison on its mainband Receivers and responds with the logged results {Tx Init D to C results resp} sideband message.
6. The UCIe Module stops sending the pattern on its Transmitters and sends the {End Tx Init D to C point test req} sideband message and the UCIe Module Partner responds with {End Tx Init D to C point test resp}. When a UCIe Module has received the {End Tx Init D to C point test resp} sideband message, the corresponding sequence has completed.

Table 4-5. Data to Clock Training Parameters[1]

Parameter	Options
Clock sampling /PI phase control	Eye Center found by training
	Left Edge found by training
	Right Edge found by training
Tx Pattern Generator Setup	LFSR Pattern for Data Lanes
	Per-Lane ID pattern for Data Lanes as defined in Table 4-7
	VALTRAIN pattern four 1's followed by four 0's
Tx Pattern Count Setup	Burst Mode: Burst Count/Idle Count/Iteration Count
	Continuous Mode count
Rx Compare Setup	Maximum comparison error threshold
	Per-Lane or aggregate error compare

1. See Table 6-11 for the encodings associated with the Options column. See also the registers in Section 7.5.3.26 and Section 7.5.3.27.

4.5.1.2 Transmitter initiated Data to Clock eye width sweep

In this mode, the Transmitter initiates the data to clock training on all Lanes in the module and sweeps through the sampling PI phases. This mode can also be used to check system margin. When not performing the actions relevant to this state:

- Data, Valid, and Track Transmitters drive low
- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Data, Valid, and Clock Receivers are enabled
- Track Receiver is permitted to be disabled

The sequence of steps for this test are as follows:

1. The UCIe Module sets up the Transmitter parameters (shown in Table 4-5), sends a {Start Tx Init D to C eye sweep req} sideband message to its UCIe Module Partner, and then waits for a response. The data field of this message includes the required parameters, shown in Table 4-5. The Receiver on the UCIe Module Partner must enable the pattern comparison circuits to compare incoming mainband data to the locally generated expected pattern. Once the data to clock training parameters for its Receiver are setup, the UCIe Module Partner responds with a {Start Tx Init D to C eye sweep resp} sideband message.
2. The UCIe Module resets the LFSR (scrambler) on its mainband Transmitters and sends sideband message {LFSR clear error req}. The UCIe Module Partner resets the LFSR and clears any prior compare results on its mainband Receivers and responds with {LFSR clear error resp} sideband message.
3. The UCIe Module sends the pattern (selected through "Tx Pattern Generator Setup") for the selected number of cycles ("Pattern Count Setup") on its mainband Transmitter.
4. The UCIe Module Partner performs comparison on its Receivers for each UI during the pattern transmission based on "Rx compare setup" and logs the results.
5. The UCIe Module requests its UCIe Module Partner for the logged results in Step 4 by sending {Tx Init D to C results req} sideband message. The UCIe Module Partner stops comparison on its mainband Receivers and responds with the logged results {Tx Init D to C results resp} sideband message.
6. The UCIe Module sweeps through the PI phases on its forwarded clock Transmitter each time repeating Step 2 through Step 5 to find the passing PI phase range. The sweep steps and range are implementation specific.
7. The UCIe Module stops sending the pattern on its Transmitters and sends the {End Tx Init D to C eye sweep req} sideband message and the UCIe Module Partner responds with {End Tx Init D to C eye sweep resp}. When a UCIe Module has received the {End Tx Init D to C point eye sweep resp} sideband message, the corresponding sequence has completed.

4.5.1.3 Receiver initiated Data to Clock point test

In this mode, the Receiver initiates the data to clock training on all Lanes in the module at a single PI phase. When not performing the actions relevant to this state:

- Data, Valid, and Track Transmitters drive low
- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Data, Valid, and Clock Receivers are enabled
- Track Receiver is permitted to be disabled

The sequence of steps for this test are as follows:

1. The UCIe Module enables the pattern comparison circuits to compare incoming mainband data to the locally generated expected pattern, sets up the Receiver parameters (shown in Table 4-5), sends a {Start Rx Init D to C point test req} sideband message to its UCIe Module Partner, and then waits for a response. The data field of this message includes the required parameters, shown in Table 4-5. Once the data to clock training parameters for its Transmitter are setup, the UCIe Module Partner responds with a {Start Rx Init D to C point test resp} sideband message.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2. The UCIe Module Partner resets the LFSR (scrambler) on its mainband Transmitters and sends sideband message {LFSR clear error req}. The UCIe Module resets the LFSR and clears any prior compare results on its mainband Receivers and responds with {LFSR clear error resp} sideband message.
3. The UCIe Module Partner sends the pattern (selected through "Tx Pattern Generator Setup") for the selected number of cycles ("Pattern Count Setup") on its mainband Transmitter.
4. The UCIe Module performs the comparison on its mainband Receivers for each UI during the pattern transmission based on "Rx compare setup" and logs the results.
5. The UCIe Module Partner sends a sideband message {Rx Init D to C Tx count done req} sideband message once the pattern count is complete. The UCIe Module, stops comparison and responds with the sideband message {Rx Init D to C Tx count done resp}. The UCIe Module can now use the logged data for its Receiver Lanes.
6. The UCIe Module sends an {End Rx Init D to C point test req} sideband message and the UCIe Module Partner responds with an {End Rx Init D to C point test resp} sideband message. When a UCIe Module has received the {End Rx Init D to C point test resp} sideband message, the corresponding sequence has completed.

4.5.1.4 Receiver initiated Data to Clock Eye Width sweep

In this mode, the Receiver initiates the data to clock training on all Lanes in the module at a single PI phase. When not performing the actions relevant to this state:

- Data, Valid, and Track Transmitters drive low
- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Data, Valid, and Clock Receivers are enabled
- Track Receiver is permitted to be disabled

The sequence of steps for this test are as follows:

1. The UCIe Module enables the pattern comparison circuits to compare incoming mainband data to the locally generated expected pattern, sets up the Receiver parameters (shown in Table 4-5), sends a {Start Rx Init D to C eye sweep req} sideband message to its UCIe Module Partner, and then waits for a response. The data field of this message includes the required parameters, shown in Table 4-5. The Transmitter on the UCIe Module Partner must be idle. Once the data to clock training parameters for its Transmitter are setup, the UCIe Module Partner responds with a {Start Rx Init D to C eye sweep resp} sideband message.
2. The UCIe Module Partner resets the LFSR (scrambler) on its mainband Transmitters and sends sideband message {LFSR clear error req}. The UCIe Module resets the LFSR and clears any prior compare results on its mainband Receivers and responds with an {LFSR clear error resp} sideband message.
3. The UCIe Module Partner sends the pattern (selected through "Tx Pattern Generator Setup") for the selected number of cycles ("Pattern Count Setup") on its mainband Transmitter.
4. The UCIe Module performs the comparison on its mainband Receivers for each UI during the pattern transmission based on "Rx compare setup" and logs the results.
5. The UCIe Module Partner requests the UCIe Module for the logged data to clock test results through a {Rx Init D to C results req} sideband message. The UCIe Module responds with the logged results for its mainband Receiver using the {Rx Init D to C results resp} sideband message. The UCIe Module Partner can determine if the pattern comparison at the PI phase passed or failed based on the results log.

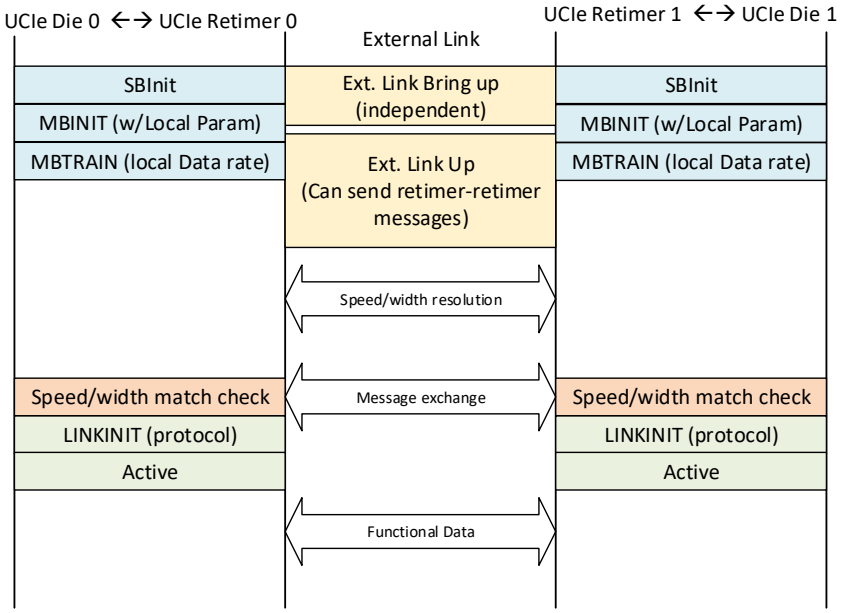
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6. The UCIe Module Partner sweeps through the PI phases on its forwarded clock each time it repeats Step 2 through Step 5 to find the passing PI phase range. The sweep pattern and range are implementation specific.
7. The UCIe Module Partner sends an {Rx Init D to C sweep done with results} sideband message with results for its mainband Transmitter. The UCIe Module can use the sweep results for its mainband Receivers.
8. The UCIe Module sends an {End Rx Init D to C eye sweep req} sideband message and the UCIe Module Partner responds with an {End Rx Init D to C eye sweep resp} sideband message. When a UCIe Module has received the {End Rx Init D to C eye sweep resp} sideband message, the corresponding sequence has completed.

4.5.2 Link Training with Retimer

The following diagram explains the initialization flow with UCIe Retimer. As shown in Figure 4-30, external and UCIe (UCIe Die to UCIe Retimer) Links are permitted to come up independently. When a UCIe Link trains up to local data rate and width, the remote UCIe information is requested over the external Link. If there is a data rate and width difference, each UCIe Link is permitted to be retrained to achieve a speed (data rate) and width match (if the UCIe Retimer requires this for operation, it must initiate Retraining from LinkSpeed state). This can happen multiple times during initial Link bring up or retraining. Once the UCIe Retimer has determined that the UCIe Link configuration is suitable for successful operation with remote Retimer partner, the UCIe Link proceeds to ACTIVE through protocol level Link initialization (LINKINIT).

Figure 4-30. Example Retimer bring up when performing speed/width match



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5.3 Link Training State Machine

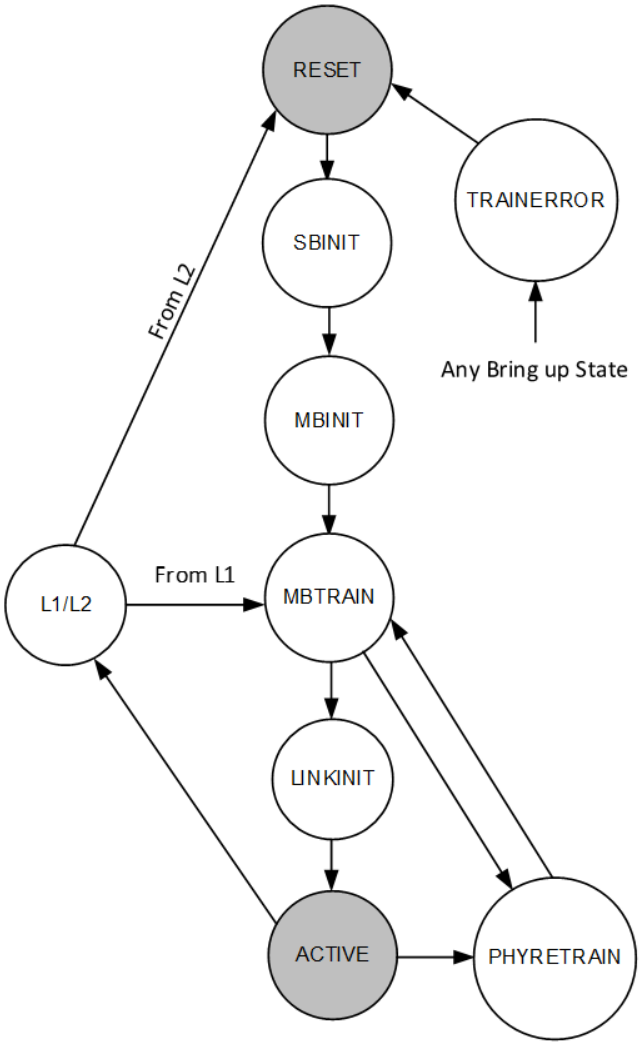
A high level initialization flow is shown in Figure 4-31. A high level description of each state is shown in Table 4-6. The details and actions performed in each state are described in the subsequent sections.

Table 4-6. State Definitions for Initialization

State	Description
RESET	This is the state following primary reset or exit from TRAINERROR
SBINIT	Sideband initialization state where the sideband is detected, repaired (when applicable) and out of reset message is transmitted
MBINIT	Following sideband initialization, Mainband (MB) is initialized at the lowest speed. Both dies perform on die calibration followed by interconnect repair (when applicable)
MBTRAIN	Mainband (Data, Clock and Valid signals) speed of operation is set to the highest negotiated data rate. Die-to-Die training of mainband is performed to center the clock with respect to Data.
LINKINIT	This state is used to exchange Adapter and Link management messages
ACTIVE	This is the state in which transactions are sent and received
L1/L2	Power Management state
PHYRETRAIN	This state is used to begin the retrain flow for the Link during runtime
TRAINERROR	State is entered when a fatal or non-fatal event occurs at any point during Link Training or operation.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-31. Link Training State Machine



4.5.3.1 RESET

Physical Layer must remain in RESET for a minimum of 4ms upon every entry to RESET, to allow PLLs to stabilize and any other Link Training initialization requirements to be met. The minimum conditions necessary to exit RESET are as follows:

- Power supplies are stable
- Sideband clock is available and running at 800 MHz
- Mainband and Die to Die Adapter clocks are stable and available
- Mainband clock is set to the slowest I/O data rate (2 GHz for 4 GT/s)
- Local SoC/Firmware not keeping the Physical Layer in RESET
- Link training trigger has occurred (triggers are defined in the beginning of Section 4.5)
- Mainband Transmitters are tri-stated

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Mainband Receivers are permitted to be disabled
- Sideband Transmitters are held low
- Sideband Receivers are enabled

4.5.3.2 Sideband Initialization (SBINIT)

In this state, the sideband (SB) interface is initialized and repaired (when applicable). When not performing the actions relevant to this state:

- UCIe Module mainband (MB) Transmitters remain tri-stated
- MB Receivers are permitted to be disabled
- SB Transmitters continue to be held Low
- SB Receivers continue to be enabled

The SB initialization procedure is performed at 800 MT/s with 800 MHz sideband clock.

Advanced Package has redundant SB clock and SB data Lanes (**DATASBRD**, **CKSBRD**) in addition to **DATASB** and **CKSB**. SBINIT sequence for **Advanced Package** where interconnect repair may be needed is as follows:

1. The UCIe Module must start and continue to send iterations of a 64 UI clock pattern (a clock pattern is defined as starting with 1b and toggling every UI of transmission, i.e., 1010...) and 32 UI low on both sideband data Transmitters (**TXDATASB** and **TXDATASBRD**). The UCIe Module must send strobes on both **TXCKSB** and **TXCKSBRD** during the 64-UI clock pattern transmission and be gated (held low) otherwise.
2. UCIe Module Partner must sample each incoming data patterns on its sideband Receivers with both incoming sideband clocks (this forms four Receiver/clock combinations).
3. A sideband data-clock Receiver combination detection is considered successful if 128 UI clock pattern is detected.
4. If a UCIe Module Partner detects the pattern successfully on at least one of its sideband data-clock Receiver combination, it must stop sending data and clock on its sideband Transmitters after four more iterations of 64 UI clock pattern and 32 UI low. This will allow for any time differences in both UCIe Module and UCIe Module Partner coming out of RESET state. The sideband Transmitter and Receiver on the UCIe Module must now be enabled to send and receive sideband messages
5. If pattern is not detected on its sideband Receiver, the UCIe Module must continue to alternate between sending the pattern on its sideband Transmitters for 1ms, and holding low for 1ms, for a total of 8 ms. The sideband Receiver of the UCIe Module must remain enabled during this time. Timeout occurs after 8ms. If a timeout occurs, the UCIe Module enters TRAINERROR state.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6. If detection is successful on more than one sideband data/clock combination, the device can pick a combination based on a priority order. Pseudocode for sideband assignment:

```
CKSB sampling DATASB = Result[0] # 1: Detected; 0: Not detected
CKSBRD sampling DATASB = Result[1] # 1: Detected; 0: Not detected
CKSB sampling DATASBRD = Result[2] # 1: Detected; 0: Not detected
CKSBRD sampling DATASBRD = Result[3] #  1: Detected; 0: Not detected

IF (Result[3:0] == XXX1):
       Sideband = (DATASB/CKSB)
ELSE IF (Result[3:0] == XX10):
       Sideband = (DATASB/CKSBRD)
ELSE IF (Result[3:0] == X100):
       Sideband = (DATASBRD/CKSB)
ELSE IF (Result[3:0] ==1000):
       Sideband = (DATASBRD/CKSBRD)
Else:
       Sideband is not functional
```

7. If the sideband on the UCIe Module is enabled to send and receive sideband messages (Step 4), the UCIe Module must start and continue to send {SBINIT Out of Reset} sideband message on both **TXDATASB** and **TXDATASBRD** while sending both **TXCKSB** and **TXCKSBRD** until it detects the same message in its sideband Receivers or a timeout occurs at 8 ms.
8. If {SBINIT Out of Reset} sideband message detection is successful on its sideband Receivers, the UCIe Module stops sending the sideband message. Before sending any further sideband messages, both UCIe Module and UCIe Module Partner must apply Sideband Data/Clock assignment (called the functional sideband) based on the information included in the {SBINIT Out of Reset} sideband message.
9. Any further sideband messages must be sent and received on the functional sideband. Any sideband message exchange can now be performed.
10. The UCIe Module sends the sideband message {SBINIT done req} and waits for a response. If this message is received successfully, UCIe Module Partner responds with {SBINIT done resp}. When a UCIe Module has sent and received {SBINIT done resp} it must exit to MBINIT

The next state is mainband initialization (MBINIT) if sideband message exchange is successful.

SBINIT sequence for **Standard Package** where interconnect Lane redundancy and repair are not supported is as follows:

1. The UCIe Module must start and continue to send iterations of 64 UI clock pattern (a clock pattern is defined as starting with 1b and toggling every UI of transmission, i.e., 1010...) and 32 UI low on its sideband Transmitter (**TXDATASB**). The UCIe Module must send strobe on its sideband clock (**TXCKSB**) during the 64-UI clock pattern duration and gated (held low) otherwise.
2. The UCIe Module Partner must sample incoming data pattern with incoming clock.
3. Sideband pattern detection is considered successful if 128 UI clock pattern is detected.
4. If the UCIe Module successfully detects the pattern, it stops sending data and clock on its sideband Transmitters after four more iterations of pattern in step 1. This will allow for any time differences in both UCIe Modules coming out of RESET. The UCIe Module sideband Transmitter and Receiver must now be enabled to send and receive sideband messages, respectively.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5. If pattern is not detected on its sideband Receiver, the UCIe Module continues to alternate between sending the pattern on its Transmitters for 1 ms, and holding low for 1 ms, for a total of 8 ms. The sideband Receiver must be enabled during this time. Timeout occurs after 8ms. If a timeout occurs, the UCIe Module must exit to TRAINERROR. If a pattern is detected successfully at any time, as described in Step 3, the UCIe Module enables sideband message transmission as described in Step 4 and continues to Step 6.
6. Once sideband detection is successful (Step 5), the UCIe Module must start and continue to send {SBINIT Out of Reset} sideband message on **TXDATASB** while sending **TXCKSB** until it detects the same message in its sideband Receivers or a timeout occurs.
7. If {SBINIT Out of Reset} sideband message detection is successful, the UCIe Module must stop sending the message. Any sideband message exchange can now be performed.
8. The UCIe Module must send the {SBINIT done req} sideband message. If this message is received successfully, each UCIe Module Partner responds with {SBINIT done resp}. When the UCIe Module has sent and received {SBINIT done resp}, it must exit to MBINIT.

The next state is mainband initialization (MBINIT) if sideband message exchange is successful. For the remainder of initialization and operations, when not transmitting sideband packets, sideband Transmitters are held Low and sideband Receivers are enabled.

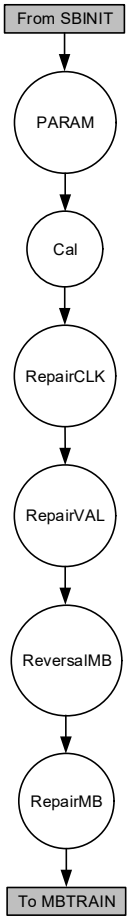
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5.3.3 MBINIT

In this state, the mainband (MB) interface is initialized and repaired or width degraded (when applicable). The data rate on the mainband is set to the lowest supported data rate (4 GT/s).

For **Advanced Package** interconnect repair may be needed. Sub-states in MBINIT allows detection and repair of data, clock, track and valid Lanes. For **Standard Package**, where no Lane repair is needed, sub-states are used to check functionality at lowest data rate and width degrade if needed.

Figure 4-32. MBINIT: Mainband Initialization and Repair Flow



4.5.3.3.1 MBINIT.PARAM

This state is used to perform exchange of parameters that are required to set up the maximum negotiated speed and other PHY settings. Mainband Transmitters remain tri-stated and mainband Receivers are permitted to be disabled. The following parameters are exchanged over sideband with UCIe Module Partner:

- "Voltage swing": The five bit value indicates the Transmitter voltage swing to the UCIe Module Partner. The UCIe Module Partner must use this value and its Receiver termination information to set the reference voltage (Vref) for its Receivers. The corresponding bits in {MBINIT.PARAM configuration resp} are reserved.
- "Maximum Data Rate": The four bit value indicates the Maximum supported Data rate to the UCIe Module Partner. This value must take into consideration all the required features at the data rate

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

(BER, CRC/Retry, quadrature clock phase support etc.). The UCIe Module Partner must compare this value with its supported maximum data rate and must respond with the maximum common data rate encoding in {MBINIT.PARAM configuration resp}. For example, a UCIe Module is 8 GT/s capable while the UCIe Module Partner advertises 16 GT/s, the UCIe Module must pick 8 GT/s and send it back in response.

- "Clock Mode": The one bit value indicates the UCIe Module's request to the UCIe Module Partner for a strobe or continuous clock. The UCIe Module Partner must use this information to set up the clock mode on its clock Transmitter. {MBINIT.PARAM configuration resp} must reflect the same value. Continuous clock mode requires the clock to be free running and is enforced after entry to MBTRAIN.RXCLKCAL. The clock remains free running through the remainder of MBTRAIN (unless MBTRAIN.LINKSPEED is exited due to errors) and in ACTIVE.
- "Clock Phase": The one bit value indicates the UCIe Module's request to UCIe Module Partner for the Clock Phase support on UCIe Module's forwarded clock. This should only be set 1b if the maximum data rate advertised is permitted to do so (see Table 5-7). The corresponding bit in {MBINIT.PARAM configuration resp} must be set to 1b if this was requested and the operational data rate allows it.
- "Module ID": The UCIe Module sends its "Module ID". This can be used by the UCIe Module Partner if in a multi-module configuration for Byte mapping, Module enable/disable information etc. The corresponding bits in {MBINIT.PARAM configuration resp} are reserved.
- "UCIe-A x32": This bit is set to 1b when APMW bit in DVSEC UCIe Link Capability register (see Section 7.5.1.4) is set to 1b (OR) 'Force x32 width mode in x64 Module' in the PHY Control register (see Section 7.5.3.23) is set; otherwise, the bit is set to 0b. If a x64 Advanced Package module supports width reduction to interoperate with a x32 Advanced Package Module, it uses this information from its link partner to condition the results during MBINIT.REVERSALMB. The corresponding bits in {MBINIT.PARAM configuration resp} are reserved.

Following is the sequence for parameter exchange:

1. The UCIe Module sends sideband message {MBINIT.PARAM configuration req} to exchange parameters with the UCIe Module Partner.
2. Once {MBINIT.PARAM configuration req} is received, the UCIe Module Partner resolves and responds with {MBINIT.PARAM configuration resp} sideband message.

4.5.3.3.2 MBINIT.CAL

This state is used to perform any calibration needed (e.g., Tx Duty Cycle correction, Receiver offset and Vref calibration). This state is common for both **Advanced Package** and **Standard Package**.

1. The UCIe Module maintains tri-state on all its mainband Transmitters, and mainband Receivers are permitted to be disabled in this state. The UCIe Module is permitted to perform implementation specific steps for Transmitter and Receiver calibration.
2. The UCIe Module must send the {MBINIT.CAL Done req} sideband message, and then wait for a response. If this message is received successfully, the UCIe Module Partner responds with the {MBINIT.CAL Done resp} sideband message. Once the UCIe Module has sent and received {MBINIT.CAL Done resp}, it must exit to MBINIT. REPAIRCLK.

4.5.3.3.3 MBINIT.REPAIRCLK

This sub-state is used to detect and apply repair (if needed) to clock and track Lanes for **Advanced Package** and for functional check of clock and track Lanes for **Standard Package**.

Following is the sequence for **Advanced Package**:

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Clock repair mapping is described in Section 4.3. Each clock, track and their redundant physical Lanes (**TCKP_P/RCKP_P**, **TCKN_P/RCKN_P**, **TTRK_P/RTRK_P**, and **TRDCK_P/RRDCK_P**) are independently checked to detect possible electrical opens or electrical shorts between the two clock pins. Single-ended clock Receivers or independent detection mechanism is required to ensure clock repair. The UCIe Module must enable Transmitters and Receivers on Clock, Track and their redundant Lanes. All other Transmitters are maintained in tri-state and Receivers are permitted to be disabled.

1. The UCIe Module sends the {MBINIT.REPAIRCLK init req} sideband message and waits for a response. The UCIe Module Partner when ready to receive pattern on **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L** responds with {MBINIT.REPAIRCLK init resp}.
2. The UCIe Module must now send 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on its **TCKP_L** only (**TCKN_L**, **TTRK_L**, and **TRDCK_L** are tri-stated). Clock repair pattern must not be scrambled.
3. The UCIe Module Partner detects this pattern on **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. Detection is considered successful if at least 16 consecutive iterations of clock repair pattern are detected. The UCIe Module Partner logs the detection result for **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**.
4. The UCIe Module after completing pattern transmission sends {MBINIT.REPAIRCLK result req} sideband message to get the logged result and waits for a response.
5. The UCIe Module Partner responds with {MBINIT.REPAIRCLK result resp} sideband message with log result of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. If detection is successful on **RCKP_L** only and not on any of **RCKN_L**, **RTRK**, and/or **RRDCK_L**, no repair is needed. Else if detection is successful on any two of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, an electrical short is implied. Else if detection is not successful on all of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, repair is needed on the physical Lane **TCKP_P/RCKP_P**.
6. After receiving the {MBINIT.REPAIRCLK result resp} sideband message, the UCIe Module sends the sideband message {MBINIT.REPAIRCLK init req} and waits for a response. The UCIe Module Partner when ready to receive pattern on **RCKP_L**, **RCKN_L**, **RTRK_L**, **RRDCK_L** responds with {MBINIT.REPAIRCLK init resp}.
7. After receiving the {MBINIT.REPAIRCLK init resp} sideband message, the UCIe Module must send 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on its **TCKN_L** only. (**TCKP_L**, **TTRK_L**, and **TRDCK_L** are tri-stated)
8. The UCIe Module Partner detects this pattern on all **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module Partner logs the detection result for **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**.
9. The UCIe Module after completing the pattern transmission, sends {MBINIT.REPAIRCLK result req} sideband message to get the logged result.
10. The UCIe Module Partner on receiving it responds with {MBINIT.REPAIRCLK result resp} sideband message with logged result of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. If detection is successful on **RCKN_L** and not on **RCKP_L**, **RTRK_L**, **RRDCK_L**, no repair is needed. Else if detection is successful on any two of **RCKN_L**, **RCKP_L**, **RTRK_L**, and **RRDCK_L**, an electrical short is implied. Else if detection is not successful on any of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, repair is needed on the physical Lane **TCKN_P/RCKN_P**.
11. After receiving the {MBINIT.REPAIRCLK result resp} sideband message, the UCIe Module sends the sideband message {MBINIT.REPAIRCLK init req}. The UCIe Module Partner when ready to receive pattern on **RCKP_L**, **RCKN_L**, **RTRK_L**, **RRDCK_L** responds with {MBINIT.REPAIRCLK init resp}.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

12. After receiving the {MBINIT.REPAIRCLK init resp} sideband message, the UCIe Module sends 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on **TRDCK_L** only. (**TCKP_L**, **TTRK_L**, and **TCKN_L** tri-stated)
13. The UCIe Module Partner detects this pattern on all **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module Partner logs the detection result for **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**.
14. The UCIe Module device after completing the pattern transmission sends {MBINIT.REPAIRCLK result req} sideband message to get the logged result.
15. The UCIe Module Partner responds with {MBINIT.REPAIRCLK result resp} sideband message with logged result of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. If detection is successful only on **RRDCK_L** and not on **RCKP_L**, **RTRK_L**, **RCKN_L**, **TRDCK_P/RRDCK_P** is available as a repair resource. Else if detection is successful on any two of **RCKN_L**, **RCKP_L**, **RTRK_L**, and **RRDCK_L**, an electrical short is implied. Else if detection is not successful on any of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, the physical Lane **TRDCK_P/RRDCK_P** is not available as a repair resource.
16. After receiving the {MBINIT.REPAIRCLK result resp} sideband message, the UCIe Module sends the sideband message {MBINIT.REPAIRCLK init req} and waits for a response. The UCIe Module Partner when ready to receive pattern on **RCKP_L**, **RCKN_L**, **RTRK_L**, **RRDCK_L** responds with {MBINIT.REPAIRCLK init resp}.
17. After receiving the {MBINIT.REPAIRCLK init resp} sideband message, the UCIe Module sends 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on TTRK_L only. (**TCKP_L**, **TCKN_L**, and **TRDCK_L** are tri-stated)
18. The UCIe Module Partner detects this pattern on all **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module Partner logs the detection result for **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**.
19. The UCIe Module after completing pattern transmission sends {MBINIT.REPAIRCLK result req} sideband message to get the logged result and waits for a response.
20. The UCIe Module Partner stops comparison and responds with {MBINIT.REPAIRCLK result resp} sideband message with logged result of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. If detection is successful only on **RTRK_L** and not on **RCKP_L**, **RCKN_L**, **RRDCK_L**, no repair is needed. Else if detection is successful on any two of **RCKN_L**, **RCKP_L**, **RTRK_L**, and **RRDCK_L**, an electrical short is implied. Else if detection is not successful on any of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, repair is needed on the physical Lane **TTRK_P/RTRK_P**.
21. If required, the UCIe Module applies repair on its clock/track Transmitter and sends the {MBINIT.REPAIRCLK apply repair req} sideband message with repair information. If a repair is needed for one of the clock or track pins (**CKP**, **CKN**, or **TRK**) and a repair resource is available, repair is applied as described in Section 4.3. The UCIe Module Partner applies repair and sends {MBINIT.REPAIRCLK apply repair resp} sideband message. Clock or Track is unrepairable if any of the following are true:
    - if repair is needed on any two of **RCKP_L**, **RCKN_L**, and **RTRK_L**
    - electrical short is detected
    - **RRDCK_L** is unavailable for repair when repair is needed

    If the clock or track is unrepairable, the UCIe Module and UCIe Module Partner must exit to TRAINERROR after performing TRAINERROR handshake (see Section 4.5.3.8).

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

22. If a repair is applied, UCIe Module must check the repair success by applying clock repair pattern and checking on the Receiver.
    a. The UCIe Module sends sideband message {MBINIT.REPAIRCLK check repair init req} to initiate check repair and waits for a response. The UCIe Module Partner responds with sideband message {MBINIT.REPAIRCLK check repair init resp} and is ready to receive and check clock repair pattern.
    b. After receiving the {MBINIT.REPAIRCLK check repair init resp} sideband message, the UCIe Module sends 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on **TCKP_L**. The UCIe Module Partner detects this pattern **RCKN_L**, **RCKP_L**, **RTRK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module requests for check result request using the sideband message {MBINIT.REPAIRCLK check results req} and the UCIe Module Partner responds with the sideband message {MBINIT.REPAIRCLK check results resp}. Repair is considered successful if pattern is detected only on **RCKP_L**. If repair is not successful, the UCIe Module and UCIe Module Partner must exit to TRAINERROR after performing TRAINERROR handshake (see Section 4.5.3.8).
    c. Step a and Step b are repeated for **TCKN_L** and **TTRK_L**.
23. If repair is successful or repair is not required, the UCIe Module sends {MBINIT.RepairCLK done req} sideband message and the UCIe Module Partner responds with {MBINIT.RepairCLK done resp} sideband message. When the UCIe Module has sent and received {MBINIT.RepairCLK done resp}, it must exit to REPAIRVAL.

For **Standard Package**, clock and track Lanes are checked for functional operation at the lowest data rate. The sequence is as follows:

1. The UCIe Module sends the sideband message {MBINIT.REPAIRCLK init req} and waits for a response. When ready to receive pattern on **RCKP_L**, **RCKN_L**, and **RTRK_L**, the UCIe Module Partner responds with {MBINIT.REPAIRCLK init resp}. On receiving the sideband message {MBINIT.REPAIRCLK init resp}, the UCIe Module sends 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on **TCKP_L**, **TCKN_L**, and **TTRK_L**. Clock repair pattern must not be scrambled.
2. The UCIe Module Partner detects this pattern on **RCKP_L**, **RCKN_L**, and **RTRK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module Partner logs the detection result for **RCKP_L**, **RCKN_L**, and **RTRK_L**.
3. After completing pattern transmission, the UCIe Module sends {MBINIT.REPAIRCLK result req} sideband message to get the logged result.
4. The UCIe Module Partner stops comparison and responds with {MBINIT.REPAIRCLK result resp} sideband message with logged result of **RCKP_L**, **RCKN_L**, and **RTRK_L**.
5. If detection is not successful on any one of **RCKP_L**, **RCKN_L**, and **RTRK_L**, the UCIe Module and UCIe Module Partner must exit to TRAINERROR after performing TRAINERROR handshake.
6. If detection is successful, the UCIe Module sends {MBINIT.RepairCLK done req} sideband message and the UCIe Module Partner responds with {MBINIT.RepairCLK done resp} sideband message. When the UCIe Module has sent and received the sideband message {MBINIT.RepairCLK done resp}, it must exit to MBINIT.REPAIRVAL.

4.5.3.3.4 MBINIT.REPAIRVAL

The UCIe Module sets the clock phase at the center of the data UI. The UCIe Module Partner must sample the received Valid with the received forwarded Clock. All Data Lanes must be held low during this state. Track and Data Receivers are permitted to be disabled. When not performing the actions relevant to this state:

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Clock Receivers are enabled

This state can be used to detect and apply repair (if needed) to Valid Lane for **Advanced Package** and for functional check of Valid for **Standard Package**.

Following is the sequence for **Advanced Package**:

1. The UCIe Module sends the sideband message {MBINIT.REPAIRVAL init req} and waits for a response. When ready to receive pattern on **RVLD_L** and **RRDVLD_L**, the UCIe Module Partner clears any previous comparison results and responds with {MBINIT.REPAIRVAL init resp}.
2. The UCIe Module on receiving {MBINIT.REPAIRVAL init resp} must send 128 iterations of VALTRAIN pattern (four 1's followed by four 0's) on **TVLD_L** along with the forwarded clock. VALTRAIN pattern must not be scrambled.
3. The UCIe Module Partner detects pattern on **RVLD_L** and **RRDVLD_L**. Detection is considered successful if at least 16 consecutive iterations of VALTRAIN pattern are detected. The Receiver logs the detection result for **RVLD_L** and **RRDVLD_L**.
4. After completing the pattern transmission, the UCIe Module must send {MBINIT.REPAIRVAL result req} sideband message and wait to get the logged result.
5. The UCIe Module Partner must respond with {MBINIT.REPAIRVAL result resp} sideband message with result in the previous step. If detection is successful on **RVLD_L** only and not on **RRDVLD_L**, no repair is needed. If detection is successful on both **RVLD_L** and **RRDVLD_L** or only on **RRDVLD_L**, the interconnect cannot be repaired. If detection is not successful on both **RVLD_L** and **RRDVLD_L**, Valid Lane (**TVLD_P/RVLD_P**) needs repair.
6. After receiving {MBINIT.REPAIRVAL result resp}, Step 1 must be repeated.
7. The UCIe Module sends 128 iterations of VALTRAIN repair pattern (four 1's followed by four 0's) on **TRDVLD_L** along with the forwarded clock. VALTRAIN pattern must not be scrambled.
8. The UCIe Module Partner detects pattern on **RVLD_L** and **RRDVLD_L**. Detection is considered successful if at least 16 consecutive iterations of VALTRAIN pattern are detected. The Receiver logs the detection result for **RVLD_L** and **RRDVLD_L**.
9. After completing the pattern transmission, the UCIe Module must send {MBINIT.REPAIRVAL result req} sideband message to get the logged result.
10. The UCIe Module Partner must stop comparison and respond with {MBINIT.REPAIRVAL result resp} sideband message with result from the previous step. If detection is successful on **RRDVLD_L** only and not on **RVLD_L**, **TRDVLD_P/RRDVLD_P** is available for repair. If detection is successful on both **RVLD_L** and **RRDVLD_L** or only on **RVLD_L**, the interconnect cannot be repaired. If detection is unsuccessful on both **RVLD_L** and **RRDVLD_L**, **TRDVLD_P/RRDVLD_P** is not available and cannot be used for Valid Lane repair.
11. If repair is required on (**TVLD_P/RVLD_P**) and repair resource is available, the UCIe Module applies repair on its Valid Transmitter (see Section 4.3.7) and sends sideband message {MBINIT.REPAIRVAL apply repair req} with repair information. The UCIe Module Partner, after receiving the message, must apply repair on its Valid Receiver and must respond with sideband message {MBINIT.REPAIRVAL apply repair resp}.
12. If a repair is applied, device must check the repair success by repeating Step 1 through Step 4.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

13. If repair is successful or repair is not required, the UCIe Module must send {MBINIT.RepairVAL done req} sideband message and the UCIe Module Partner must respond with {MBINIT.RepairVAL done resp}. When a UCIe Module has sent and received {MBINIT.RepairVAL done resp} sideband message, it must exit to REVERSALMB. Else if repair is unsuccessful or Valid Lane is unrepairable (in Step 11), the UCIe Module must exit to TRAINERROR after completing the TRAINERROR handshake.

For **Standard Package**, Valid Lane is checked for functional operation at the lowest data rate. Following is the flow:

1. The UCIe Module must send the sideband message {MBINIT.REPAIRVAL init req} and wait for a response. The UCIe Module Partner when ready to receive pattern on **RVLD_L**, must respond with {MBINIT.REPAIRVAL init resp}.
2. After receiving the sideband message {MBINIT.REPAIRVAL init resp}, the UCIe Module sends 128 iterations of VALTRAIN pattern (four 1's followed by four 0's) on **TVLD_L** along with the forwarded clock.
3. The UCIe Module Partner detects this pattern on **RVLD_L**. Detection is considered successful if at least 16 consecutive iterations of valid repair pattern are detected. The Receiver logs the detection result for **RVLD_L**.
4. After completing pattern transmission, the UCIe Module must send {MBINIT.REPAIRVAL result req} sideband message and wait to get the logged result.
5. The UCIe Module Partner must stop comparison and respond with {MBINIT.REPAIRVAL result resp} sideband message with result in the previous step.
6. If detection fails, the UCIe Module must exit to TRAINERROR after completing the TRAINERROR handshake.
7. If detection is successful, the UCIe Module must send {MBINIT.RepairVAL done req} sideband message and the UCIe Module Partner responds with {MBINIT.RepairVAL done resp}. When a UCIe Module has sent and received {MBINIT.RepairVAL done resp} sideband message, it must exit to REVERSALMB.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5.3.3.5 MBINIT.REVERSALMB

This state is entered only if Clock and Valid Lanes are functional. In this state, Data Lane reversal is detected. All the Transmitters and Receivers are enabled. The UCIe Module sets the forwarded clock phase at the center of the data UI. The UCIe Module Partner must sample the incoming Data with the incoming forwarded clock. The Track Transmitter is held low and the Track Receiver is permitted to be disabled. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

A 16-bit "Per Lane ID" pattern, shown in Table 4-7, is a Lane-specific pattern using the Lane ID described in Section 4.2.1. Example of "Per Lane ID" pattern for Lane 1 and Lane 31 are shown in Table 4-8. When "Per Lane ID" pattern is used, it must not be scrambled.

Table 4-7. Per Lane ID pattern

Pattern	0	1	0	1	Lane ID (LSB first)								0	1	0	1
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 4-8. Per Lane ID pattern examples

Lane 1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1
Lane 31	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Following is the Reversal MB sequence for **Advanced Package** and **Standard Package**:

1. The UCIe Module must send the {MBINIT.REVERSALMB init req} sideband message. When ready to receive "Per Lane ID" pattern and perform per-Lane pattern comparison, the UCIe Module Partner must respond with {MBINIT.REVERSALMB init resp}.
2. On Receiving the {MBINIT.REVERSALMB init resp} sideband message or entering from Step 8, the UCIe Module must send {MBINIT.REVERSALMB clear error req} sideband message. Upon receiving this message, the UCIe Module Partner clears any prior errors and responds with {MBINIT.REVERSALMB clear error resp}. After receiving {MBINIT.REVERSALMB clear error resp}, the UCIe Module sends 128 iterations of Per Lane ID pattern (LSB first) on all N data Lanes with correct Valid framing on the Valid Lane (see Section 5.11 and Section 4.1.2) along with the forwarded clock. Table 4-7 and Table 4-8 show examples of the Per Lane ID pattern. N is 68 (64 Data + 4 RD) for a x64 Advanced Package. N is 34 (32 Data + 2 RD) for a x32 Advanced Package. N is 16 for a Standard Package.
3. The UCIe Module Partner must perform a per-Lane compare on its Receivers on all N Lanes (see Section 4.4). Detection on a Lane is considered successful if at least 16 consecutive iterations of "Per Lane ID" pattern are detected. The UCIe Module Partner logs the detection result for its Receiver Lanes to be used for Lane reversal Detection.
4. After sending 128 iterations of "Per Lane ID" pattern, the UCIe Module stops sending the pattern and sends {MBINIT.REVERSALMB result req} sideband message to get the logged result.
5. The UCIe Module Partner stops comparison and must respond with {MBINIT.REVERSALMB result resp} sideband message with N-bit (68 for x64 Advanced Package, 34 for x32 Advanced Package, and 16 for Standard Package) per Lane result.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6. If majority of the Lanes show success (since some Lanes may need repair), Lane reversal is not needed. Skip to Step 11.
7. Else if results from Step 5 show majority of Lanes are unsuccessful, the UCIe Module applies Lane reversal on its Transmitters.
8. Following the Lane reversal application on its Transmitters, the UCIe Module repeats Step 1 through Step 5.
9. If majority of Lanes show success, the Lane reversal is needed. Lane reversal preserved for rest of the device operation. Skip to step 11.
10. Else if results from step 8 show majority Lanes are unsuccessful, the UCIe Module must exit to TRAINERROR after completing the TRAINERROR handshake.
11. The UCIe Module must send {MBINIT.ReversalMB done req} sideband message and the UCIe Module Partner responds with {MBINIT.RversalMB done resp}. When the UCIe Module has sent and received {MBINIT.ReversalMB done resp} sideband message, it must exit to REPAIRMB.

If a x64 Advanced Package module that supports interoperation with a x32 Advanced Package module had received "UCIe-A x32" as 1b during parameter exchanges, it must recognize that it is connected to a x32 Advanced Package module and appropriately interpret the received {MBINIT.REVERSALMB result resp} sideband message. In this scenario, the x64 applies steps 6 through 9 to lower 32 data lane set and 2 repair lane set. The x64 module applies Lane reversal (if required) within the lower 32 data lane set and 2 repair lane set.

If a x32 Advanced Package module had received "UCIe-A x32" as 0b during parameter exchanges, it must recognize that it is connected to a x64 Advanced Package module and appropriately interpret the received {MBINIT.REVERSALMB result resp} sideband message, looking for majority of success in the lower 32 data lane set and 2 repair lane set of the x64 module (the x64 module will always place the results of its receiver on the lower half of its data/repair lane set).

4.5.3.3.6 MBINIT.REPAIRMB

This state is entered only after Lane reversal detection and application is successful. All the Transmitters and Receivers on a UCIe Module are enabled. The UCIe Module sets the clock phase at the center of the data UI on its mainband Transmitter. The UCIe Module Partner must sample the incoming Data with the incoming forwarded clock on its mainband Receivers. The Track Transmitter is held low and the Track Receiver is permitted to be disabled. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

In this state, the mainband Lanes are detected and repaired (if needed) for **Advanced Package**. In this state, functional checks and width degrade (if needed) are performed for **Standard Package**.

Following is the sequence for **Advanced Package**:

1. The UCIe Module must send the {MBINIT.REPAIRMB start req} sideband message and waits for a response. The UCIe Module Partner responds with {MBINIT.REPAIRMB start resp}.
2. The UCIe Module device performs Transmitter initiated data to clock point training as described in Section 4.5.1.1 on its Transmitter Lanes.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

   a. The transmit pattern must be set up to send 128 iterations of continuous mode "Per Lane ID" Pattern. "Per Lane ID" pattern must not be scrambled. The Receiver must be set up to perform per Lane comparison.
   b. Detection on a Receiver Lane is considered successful if at least 16 consecutive iterations of "Per Lane ID" pattern are detected.
   c. LFSR Reset has no impact in MBINIT.REPAIRMB.
3. The UCIe Module receives the per-Lane pass/fail information over sideband message at the end of Transmitter initiated data to clock point test Step 2.
4. If Lane repair is required and the necessary repair resources are available, the UCIe Module applies repair on its mainband Transmitters as described in Section 4.3.1, and sends {MBINIT.REPAIRMB Apply repair req} sideband message. Upon receiving this sideband message, the UCIe Module Partner applies repair on its mainband Receivers as described in Section 4.3.1, and sends {MBINIT.REPAIRMB Apply repair resp} sideband message. Otherwise, if the number of Lane failures are more than repair capability (see Section 4.3), the mainband is unrepairable and the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.
5. If repair is not required, perform step 7.
6. If Lane repair is applied (step 4), the applied repair is checked by UCIe Module by repeating step 2 and step 3. If post repair Lane errors are logged in step 5, the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake. If repair is successful, perform step 7.
7. The UCIe Module sends {MBINIT.REPAIRMB end req} sideband message and the UCIe Module Partner responds with {MBINIT.REPAIRMB end resp}. When UCIe Module has sent and received {MBINIT.REPAIRMB end resp}, it must exit to MBTRAIN.

For **Standard Package**, mainband is checked for functional operation at the lowest data rate. Following is the sequence of steps:

1. The UCIe Module sends the {MBINIT.REPAIRMB start req} sideband message and waits for a response. The UCIe Module Partner responds with {MBINIT.REPAIRMB start resp} when ready to receive the pattern on its mainband Receivers.
2. The UCIe Module performs Transmitter initiated data to clock point training as described in Section 4.5.
   a. The transmit pattern must be set up to send 128 iterations of continuous mode "Per Lane ID" Pattern. The Receiver must be set up to perform per Lane comparison.
   b. Detection on a Receiver Lane is considered successful if at least 16 consecutive iterations of "Per Lane ID" pattern are detected.
   c. LFSR Reset has no impact in MBINIT.REPAIRMB.
3. The UCIe Module must send {MBINIT.REPAIRMB apply degrade req} indicating the functional Lanes on its Transmitter using one of the logical Lane map encodings from Table 4-9. If the remote Link partner indicated a width degrade in the functional Lanes, the UCIe Module must apply the corresponding width degrade to its Receiver. If the remote Link partner indicated all Lanes are functional, the UCIe Module sets its Transmitter and Receiver to the width corresponding to the functional Lane encoding determined on its Transmitter. The UCIe Module sends the {MBINIT.REPAIRMB apply degrade resp} after setting its Transmitter and Receiver lanes to the relevant width. If the width on the Transmitter or Receiver has changed, both Link partners must repeat Step 2. If the width on the Transmitter or Receiver has not changed, proceed to Step 4. If a "Degrade not possible" (00b) encoding is sent or received in the {MBINIT.REPAIRMB apply degrade req}, the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4. The UCIe Module sends {MBINIT.REPAIRMB end req} sideband message and the UCIe Module Partner responds with {MBINIT.REPAIRMB end resp}. When the UCIe Module has sent and received {MBINIT.REPAIRMB end resp}, it must exit to MBTRAIN.

Table 4-9. Standard Package Logical Lane map

Lane map code	Functional Lanes
00b	None (Degrade not possible)
01b	Logical Lanes 0 to 7
10b	Logical Lanes 8 to 15
11b	0 - 15

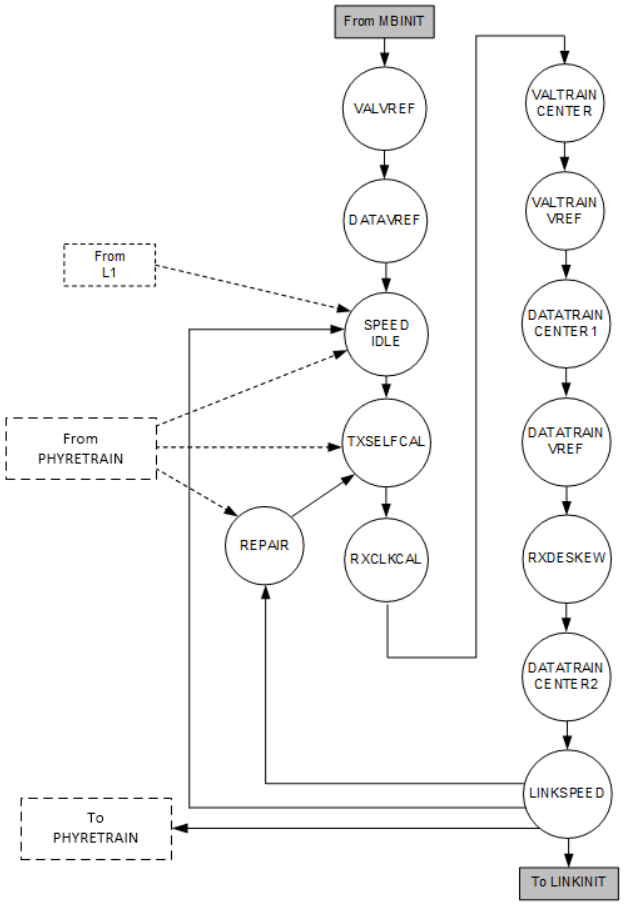
4.5.3.4 MBTRAIN

MBTRAIN state is used to setup operational speed and perform clock to data centering. At higher speeds, additional calibrations like Rx clock correction, Tx and Rx deskew may be needed to ensure Link performance. MBTRAIN uses sub-states to perform all the required calibration and training. UCIe Modules must enter each sub-state and the exit from each sub-state is coordinated between UCIe Module Partners through sideband handshakes. If a particular action within a sub-state is not needed, the UCIe Module is permitted to exit the sub-state through the relevant sideband handshake without performing the described operations in that sub-state.

Devices enter this state once the MBINIT is completed. This state is common for **Advanced** and **Standard Packages**.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-33. Mainband Training



4.5.3.4.1 MBTRAIN.VALVREF

Receiver reference voltage (Vref) to sample the incoming Valid is optimized in this state. The data rate on the mainband continues to be at the lowest supported data rate (4 GT/s). The UCIe Module Partner must set the forwarded clock phase at the center of the data UI on its mainband Transmitters. The UCIe Module must sample the pattern on Valid signal with the forwarded clock. All data Lanes and Track must be held low during Valid Lane reference voltage training. Track Receivers are permitted to be disabled. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Clock Receivers are enabled

The sequence for Valid Receiver reference voltage adjustment is as follows:

1. The UCIe Module must send the {MBTRAIN.VALVREF start req} sideband message and wait for a response. When {MBTRAIN.VALVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.VALVREF start resp}.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2. UCIe Module optimizes Vref on its Valid Receiver by adjusting Receiver reference voltage and performing one or more Receiver initiated point tests (see Section 4.5.1.3) or Receiver initiated eye width sweeps (see Section 4.5.1.4).
   a. The transmit pattern must be set to send 128 iterations of continuous mode "VALTRAIN" (four 1's and four zeros) pattern (see Table 4-5). This pattern must not be scrambled. The Receiver must be set up to perform comparison on the Valid Lane.
   b. Detection on a Receiver Lane is considered successful if "VALTRAIN" pattern detection errors are less than the set threshold (per Lane comparison threshold in Section 7.5.3.29).
   c. It should be noted that LFSR RESET has no impact in MBTRAIN.VALVREF.
3. The UCIe Module must send {MBTRAIN.VALVREF end req} sideband message after the Vref optimization (One way to perform Vref Optimization is to step through Vref and perform step 2 at each setting). When {MBTRAIN.VALVREF end req} is received, the UCIe Module Partner must respond with {MBTRAIN.VALVREF end resp}. When the UCIe Module has sent and received the sideband message {MBTRAIN.VALVREF end resp}, it must exit to MBTRAIN.DATAVREF.

4.5.3.4.2 MBTRAIN.DATAVREF

Receiver reference voltage (Vref) to sample the incoming data is optimized in this state. The data rate on the UCIe Module mainband continues to be at the lowest supported data rate (4 GT/s). The Transmitter sets the forwarded clock phase at the center of the data UI. The Track Transmitter is held low and the Track Receiver is permitted to be disabled. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

The sequence for data Receiver reference voltage adjustment is as follows:

1. The UCIe Module sends the sideband message {MBTRAIN.DATAVREF start req}. When {MBTRAIN.DATAVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATAVREF start resp}.
2. The UCIe Module optimizes data Vref by adjusting data Receiver reference voltage and performing one or more Receiver initiated point tests (see Section 4.5.1.3) or Receiver initiated eye width sweeps (see Section 4.5.1.4).
   a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in Section 4.4.1. The LFSR pattern on Data must be accompanied by correct Valid framing on the Valid Lane as described in Section 4.1.2. The Receiver must be set up to perform per Lane comparison.
   b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see Section 7.5.3.29).
3. The UCIe Module must send {MBTRAIN.DATAVREF end req} sideband message after the Vref optimization (One way to perform Vref Optimization is to step through Vref and perform step 2 at each setting). When {MBTRAIN.DATAVREF end req} is received, the UCIe Module Partner must respond with {MBTRAIN.DATAVREF end resp}. Once the UCIe Module has sent and received the sideband message {MBTRAIN.DATAVREF end resp}, it must exit to MBTRAIN.SPEEDIDLE.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5.3.4.3 MBTRAIN.SPEEDIDLE

This is an electrical idle state to allow frequency change. Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking). Clock Receivers are enabled. Data, Valid, and Track Transmitters are held low.

The following rules apply:

1. The data rate is determined as follows:
   - If this state is entered from MBTRAIN.DATAVREF, the UCIe Module transitions to a data rate based on the highest common speed between the two devices (see Section 4.5.3.3.1).
   - Else if this state is entered from L1, the operating speed in the last ACTIVE state before entering L1 must be restored.
   - Else if this state is entered from LINKSPEED or PHYRETRAIN (speed degrade), and the current speed is not 4 GT/s, the next-lower data rate must be picked.
   - Else the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.
2. The width of the Link is set to the width previously determined at the exit of MBINIT.REPAIRMB.
3. Upon completing the transition to the required data rate, the UCIe Module must send {MBTRAIN.SPEEDIDLE done req} sideband message. When {MBTRAIN.SPEEDIDLE done req} sideband message is received, UCIe Module Partner responds with {MBTRAIN.SPEEDIDLE done resp}. Once the UCIe Module has sent and received the {MBTRAIN.SPEEDIDLE done resp} sideband message, it must exit to MBTRAIN.TXSELFCAL.

4.5.3.4.4 MBTRAIN.TXSELFCAL

The UCIe Module calibrates its circuit parameters independent of the UCIe Module Partner. Mainband Transmitters are tri-stated. Mainband Receivers are permitted to be disabled.

1. UCIe Module is permitted to perform implementation specific Transmitter-related calibration.
2. Upon completion of calibration, the UCIe Module must send the {MBTRAIN.TXSELFCAL Done req} sideband message. When {MBTRAIN.TXSELFCAL Done req} sideband message is received, the UCIe Module Partner must respond with {MBTRAIN.TXSELFCAL Done resp}. When the UCIe Module has sent and received the {MBTRAIN.TXSELFCAL Done resp} sideband message, it must exit to MBTRAIN. RXCLKCAL.

4.5.3.4.5 MBTRAIN.RXCLKCAL

In this state, Data, Valid Transmitters are held low (Data and Valid Receivers are permitted to be disabled). When not performing the actions relevant to this state, Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if Strobe mode is negotiated.

1. The UCIe Module, when ready to perform calibration on its Clock receive path, sends the sideband message {MBTRAIN.RXCLKCAL start req}. When the sideband message {MBTRAIN.RXCLKCAL start req} is received, the UCIe Module Partner starts sending the forwarded clock and Track. Subsequently, the UCIe Module sends the sideband message {MBTRAIN.RXCLKCAL start resp}. The Transmitter clock must be free running and all Data Lanes and Valid must be held low. The UCIe Module is permitted to use the forwarded clock to perform any clock path-related and Clock-to-Track-related calibration. The UCIe Module Partner must not adjust any circuit or PI phase parameters on its Transmitters within this state.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2. When the required calibration (if any) is performed, the UCIe Module sends {MBTRAIN.RXCLKCAL done req} sideband message. When {MBTRAIN.RXCLKCAL done req} is received, the UCIe Module Partner stops sending forwarded clock and responds by sending {MBTRAIN.RXCLKCAL done resp} sideband message. When a UCIe Module has sent and received {MBTRAIN.RXCLKCAL done resp} sideband message it must exit to MBTRAIN.VALTRAINCENTER.

4.5.3.4.6 MBTRAIN.VALTRAINCENTER

To ensure the valid signal is functional, valid to clock training is performed before the data Lane training. The Receiver samples the pattern on Valid with the forwarded clock. Receiver reference voltage is set to the optimized value achieved through Vref training (see Section 4.5.3.4.1 and Section 4.5.3.4.2). All data and Track Transmitters are held low during valid to clock training. When not performing the actions relevant to this state, Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if Strobe mode is negotiated.

Following is the MBTRAIN.VALTRAINCENTER sequence:

1. The UCIe Module sends a sideband message {MBTRAIN.VALTRAINCENTER start req}. The UCIe Module Partner responds with {MBTRAIN.VALTRAINCENTER start resp}.
2. The UCIe Module must perform a Transmitter initiated full eye width sweep (see Section 4.5.1.2) or one or more Transmitter initiated point tests (see Section 4.5.1.1) to determine the correct Valid to clock centering.
   a. The transmit pattern must be set to send 128 iterations of continuous mode "VALTRAIN" (four 1's and four zeros) pattern (see Table 4-5). This pattern must not be scrambled. The Receiver must be set up to perform comparison on the Valid Lane.
   b. Detection on a Receiver Lane is considered successful if "VALTRAIN" pattern detection errors are less than set threshold (per Lane comparison threshold in Section 7.5.3.29).
   c. It should be noted that LFSR RESET has no impact in MBTRAIN.VALVREF.
3. The UCIe Module can use the received results log to assess valid functionality and margins. Following this, step 4 must be performed.
4. The UCIe Module must send {MBTRAIN.VALTRAINCENTER done req} sideband message. When {MBTRAIN.VALTRAINCENTER done req} is received the UCIe Module Partner responds with {MBTRAIN.VALTRAINCENTER done resp}. Once the UCIe Module has sent and received {MBTRAIN.VALTRAINCENTER done resp} sideband message, the UCIe Module must exit to MBTRAIN.VALTRAINVREF.

4.5.3.4.7 MBTRAIN.VALTRAINVREF

UCIe Module is permitted to optionally optimize the reference voltage (Vref) to sample the incoming Valid at the operating data rate. All Data and Track Transmitters are held low during Valid-to-Clock training. When not performing the actions relevant to this state, Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if Strobe mode is negotiated.

The sequence for Valid Receiver reference voltage adjustment is as follows:

1. The UCIe Module must send the sideband message {MBTRAIN.VALTRAINVREF start req}. When {MBTRAIN.VALTRAINVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.VALTRAINVREF start resp}.
2. UCIe Module optionally optimizes Vref by adjusting Receiver reference voltage on its Valid Receiver and performing Receiver initiated eye width sweep (see Section 4.5.1.4) or one or more Receiver initiated point test (see Section 4.5.1.3). Step 2 is optional and implementation specific.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

a. If Valid centering is performed, the transmit pattern must be set to send 128 iterations of continuous mode "VALTRAIN" (four 1's and four zeros) pattern (see Table 4-5). This pattern must not be scrambled. The Receiver must be set up to perform comparison on the Valid Lane.

b. Detection on a Receiver Lane is considered successful if "VALTRAIN" pattern detection errors are less than set threshold (per Lane comparison threshold in Section 7.5.3.29).

c. It should be noted that LFSR RESET has no impact in MBTRAIN.VALVREF.

3. The UCIe Module must send {MBTRAIN.VALTRAINVREF end req} sideband message after the Vref optimization is complete. When {MBTRAIN.VALTRAINVREF end req} is received, the UCIe Module Partner must respond with {MBTRAIN.VALTRAINVREF end resp}. Once the UCIe Module has sent and received the sideband message {MBTRAIN.VALTRAINVREF end resp}, it must exit to MBTRAIN.DATATRAINCENTER1.

4.5.3.4.8 MBTRAIN.DATATRAINCENTER1

In this state, the UCIe Module performs full data to clock training (including valid). LFSR patterns described in Section 4.4.1 must be used in this state. The Track Transmitter is held Low. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if strobe mode is negotiated
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

Following is the MBTRAIN.DATATRAINCENTER1 sequence:

1. The UCIe Module sends the {MBTRAIN.DATATRAINCENTER1 start req} sideband message. When {MBTRAIN.DATATRAINCENTER1 start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER1 start resp}.
2. The UCIe Module must perform a Transmitter initiated full eye width sweep (see Section 4.5.1.2) or one or more point tests (see Section 4.5.1.1) to determine the correct data to clock centering and adjust Transmitter per-bit deskew (if needed).

   a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in Section 4.4.1. The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in Section 4.1.2. The Receiver must be set up to perform per Lane comparison.

   b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see Section 7.5.3.29).

3. If the test is a success, the UCIe Module must set the clock phase to sample the data eye at the optimal point to maximize eye margins. The UCIe Module must send {MBTRAIN.DATATRAINCENTER1 end req} sideband message. When {MBTRAIN.DATATRAINCENTER1 end req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER1 end resp}. Once the UCIe Module has sent and received the {MBTRAIN.DATATRAINCENTER1 end resp} sideband message, it must exit to MBTRAIN.DATATRAINVREF.

4.5.3.4.9 MBTRAIN.DATATRAINVREF

UCIe Module is permitted to optionally optimize the reference voltage (Vref) on its data Receivers to optimize sampling of the incoming Data at the operating data rate. The Track Transmitter is held Low. When not performing the actions relevant to this state:

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if strobe mode is negotiated
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

The sequence for data Receiver reference voltage adjustment is as follows:

1. The UCIe Module must send the sideband message {MBTRAIN.DATATRAINVREF start req}. When {MBTRAIN.DATATRAINVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINVREF start resp}.
2. UCIe Module optionally optimizes Vref by adjusting Receiver reference voltage and performing either Receiver initiated eye width sweep (see Section 4.5.1.4) or one or more Receiver initiated point test (see Section 4.5.1.3). Step2is optional and implementation specific. If Data Vref optimization is performed:
   a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in Section 4.4.1. The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in Section 4.1.2. The Receiver must be set up to perform per Lane comparison.
   b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see Section 7.5.3.29).
3. The UCIe Module must send {MBTRAIN.DATATRAINVREF end req} sideband message after the Vref optimization is complete. When {MBTRAIN.DATATRAINVREF end req} is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINVREF end resp}. Once the UCIe Module has sent and received the sideband message {MBTRAIN.DATATRAINVREF end resp}, it must exit to MBTRAIN.RXDESKEW.

Note: It is possible that the eye opening in this step is insufficient (test fails) and a per-bit deskew may be needed on the Receiver. Thus, the UCIe Module must exit to MBTRAIN.RXDESKEW.

4.5.3.4.10 MBTRAIN.RXDESKEW

The UCIe Module is permitted to optionally perform per Lane deskew on its Receivers to improve timing margin in this state. The Track Transmitter is held Low. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if strobe mode is negotiated
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

Following is the MBTRAIN.RXDESKEW sequence:

1. The UCIe Module must send the sideband message {MBTRAIN.RXDESKEW start req}. When {MBTRAIN.RXDESKEW start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.RXDESKEW start resp}.
2. UCIe Module optionally performs per Lane deskew on its Receivers by a Receiver initiated full eyed width sweep (see Section 4.5.1.4) or one or more Receiver initiated point tests (see Section 4.5.1.3). Step2 is optional and implementation specific. If per Lane deskew is performed.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in Section 4.4.1. The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in Section 4.1.2. The Receiver must be set up to perform per Lane comparison.

b. Detection on a Receiver Lane is considered successful if total error count is less than the set threshold (see Section 7.5.3.29).

3. The UCIe Module must send {MBTRAIN.RXDESKEW end req} sideband message after the deskew is performed or skipped. When {MBTRAIN.RXDESKEW end req} is received, the UCIe Module Partner must respond with {MBTRAIN.RXDESKEW end resp}. Once UCIe Module has sent and received the sideband message {MBTRAIN.RXDESKEW end resp}, it must exit to MBTRAIN.DATATRAINCENTER2.

4.5.3.4.11 MBTRAIN.DATATRAINCENTER2

This state is needed for the UCIe Module to recenter clock to aggregate data in case the UCIe Module Partner's Receiver performed a per Lane deskew. The Track Transmitter is held Low. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if strobe mode is negotiated
- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

Following is the MBTRAIN.DATATRAINCENTER2 sequence:

1. The UCIe Module sends the sideband message {MBTRAIN.DATATRAINCENTER2 start req}. When {MBTRAIN.DATATRAINCENTER2 start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER2 start resp}.
2. The UCIe Module must perform a Transmitter initiated full eye width sweep (see Section 4.5.1.2) or one or more Transmitter initiated point tests (see Section 4.5.1.1) to determine the correct data to eye centering.

   a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in Section 4.4.1. The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in Section 4.1.2. The Receiver must be set up to perform per Lane comparison.

   b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see Section 7.5.3.29).

3. The UCIe Module uses the received training results to calculate the final eye center and set the clock phase to sample the data eye at the optimal point to maximize eye margins. The UCIe Module must send the {MBTRAIN.DATATRAINCENTER2 end req} sideband message. When {MBTRAIN.DATATRAINCENTER2 end req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER2 end resp}. Once UCIe Module has sent and received {MBTRAIN.DATATRAINCENTER2 end resp} sideband message, it must exit to MBTRAIN.LINKSPEED.

4.5.3.4.12 MBTRAIN.LINKSPEED

In this state, the UCIe Module checks Link stability at the operating date rate. The Track Transmitter is held Low. When not performing the actions relevant to this state:

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if strobe mode is negotiated

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled

The following steps must be executed sequentially, when applicable:

1. The UCIe Module sends the sideband message {MBTRAIN.LINKSPEED start req}. When {MBTRAIN.LINKSPEED start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.LINKSPEED start resp}.
2. The UCIe Module must perform Transmitter initiated Data to Clock point test (see Section 4.5.1.1) with the final clock sampling phase calculated in the previous MBTRAIN.DATACENTER2 state. LFSR pattern described in Section 4.4.1 must be used in this state.
   a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in Section 4.4.1. The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in Section 4.1.2. The Receiver must be set up to perform per Lane comparison.
   b. Detection on a Receiver Lane is considered successful if total error count is less than the set threshold (see Section 7.5.3.29).
3. For single-Module instantiations, if errors are encountered, the UCIe Module sets its Transmitters to an electrical idle state and sends {MBTRAIN.LINKSPEED error req} sideband message. If an {MBTRAIN.LINKSPEED error req} sideband message is received, the UCIe Module Partner must complete Step 1 and Step 2 and enters electrical idle on its Receiver and sends the sideband message {MBTRAIN.LINKSPEED error resp}. Once {MBTRAIN.LINKSPEED error resp} is received, the following rules apply:
   a. Based on the number of Lanes encountering errors, the UCIe Module checks if the failing Lanes can be repaired (for Advanced package) or Width degraded (for standard package). If Lanes can be can be repaired (for Advanced package) or Width degraded (for standard package), the UCIe Module must send {MBTRAIN.LINKSPEED exit to repair req} to the UCIe Module Partner. The UCIe Module Partner, if not initiating a speed degrade, enters MBTRAIN.REPAIR and sends the sideband message {MBTRAIN.LINKSPEED exit to repair resp}. If {MBTRAIN.LINKSPEED exit to repair resp} is received in response to a {MBTRAIN.LINKSPEED exit to repair req}, the UCIe Module must exit to MBTRAIN.REPAIR. If a UCIe Module is initiating a speed degrade, it must not respond to {MBTRAIN.LINKSPEED exit to repair req}.
   b. If the Lanes cannot be repaired (for Advanced package) or width degraded (for Standard package), the speed must be degraded. The UCIe Module sends {MBTRAIN.LINKSPEED exit to speed degrade req} sideband message and waits for a response from the remote Link partner. The UCIe Module Partner must respond with {MBTRAIN.LINKSPEED exit to speed degrade resp}. Following this handshake, the UCIe Module must exit to MBTRAIN.SPEEDIDLE to set data rate to next lower speed.
   c. If the UCIe Module receives an {MBTRAIN.LINKSPEED exit to speed degrade req} any outstanding {MBTRAIN.LINKSPEED exit to repair req} must be abandoned and the UCIe Module must respond to {MBTRAIN.LINKSPEED exit to speed degrade req}.
   d. Any outstanding {MBTRAIN.LINKSPEED done req} must be abandoned if a UCIe Module has received a {MBTRAIN.LINKSPEED error req}.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4. For single- or multi-module instantiations, if no errors are encountered, the UCIe Module must set the clock phase on its Transmitter to sample the data eye at the optimal point to maximize eye margins. If PHY_IN_RETRAIN is not set for single-module instantiations, proceed to Step 6. If PHY_IN_RETRAIN is not set, for multi-module instantiations, the UCIe Module must send the {MBTRAIN.LINKSPEED done req} (if not waiting for Link match criteria as a Retimer) to the remote UCIe Module Partner and wait for multi-module PHY Logic (MMPL) resolution in Step 5c. If the PHY_IN_RETRAIN variable is set, the following actions must be taken:
   a. If a change is detected in Runtime Link Testing Control register relative to the values at previous PHYRETRAIN entry, the UCIe Module must send {MBTRAIN.LINKSPEED exit to phy retrain req} and wait for a response. Upon receiving this message, the UCIe Module Partner must exit to PHY retrain and send {MBTRAIN.LINKSPEED exit to PHY retrain resp}. Once this sideband message is received, the UCIe Module must exit to PHY retrain.
   b. Else if no change is detected in the Runtime Link Testing Control register relative to the values at previous PHYRETRAIN entry, Busy bit in Runtime Link Testing Status and PHY_IN_RETRAIN variable must be cleared and the UCIe Module must proceed to Step 6.
5. For multi-module instantiations, if errors are encountered, the UCIe Module sets its Transmitters to an electrical idle state and sends {MBTRAIN.LINKSPEED error req} sideband message. If an {MBTRAIN.LINKSPEED error req} sideband message is received, the UCIe Module Partner must complete Step 1 and Step 2, enter electrical idle on its Receiver, and send the {MBTRAIN.LINKSPEED error resp} sideband message. Once {MBTRAIN.LINKSPEED error resp} is received, the following rules apply:
   a. Based on the number of Lanes encountering errors, the UCIe Module checks whether the failing Lanes can be repaired (for Advanced Package) or Width degraded (for Standard Package). If Lanes can be can be repaired (for Advanced Package) or Width degraded (for Standard Package), the UCIe Module must send {MBTRAIN.LINKSPEED exit to repair req} to the UCIe Module Partner.
   b. If the Lanes cannot be repaired (for Advanced Package) or width degraded (for Standard Package), the speed must be degraded. The UCIe Module sends {MBTRAIN.LINKSPEED exit to speed degrade req}.
   c. The UCIe Module informs MMPL of local and remote error or done requests and waits for resolution.
   d. Based on the resolution flow chart in Section 4.7, MMPL directs each Module to send either the {MBTRAIN.LINKSPEED exit to repair resp} (indicating next state is REPAIR), {MBTRAIN.LINKSPEED exit to speed degrade resp} (indicating next state is SPEEDIDLE with target speed to next-lower speed), {MBTRAIN.LINKSPEED multi-module disable module resp} (indicating next state is TRAINERROR and eventually RESET), or {MBTRAIN.LINKSPEED done resp} (indicating next state is LINKINIT). This is done regardless of the module's original error request or done request, and indicates the result of the resolution and next state to each module. The UCIe Module transitions to next state once it has sent and received the sideband response message that matches the expected resolution. Any mismatch on received message vs. expected resolution must take all modules to TRAINERROR. For Retimer dies, the resolution must take into account any Link match requirements, and while resolving the target configuration with remote Retimer partner, each UCIe Module from the Retimer die must send {MBTRAIN.LINKSPEED done resp} with stall encoding every 4 ms. The UCIe Retimer must ensure that this stall is not perpetual, and an implementation-specific timeout must be included in the Retimer. If {MBTRAIN.LINKSPEED done resp} with stall encoding is received, it must reset timers for state transition as well as any outstanding handshakes for multi-module resolution.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6. If the UCIe die is not a Retimer, proceed to Step 7. If the UCIe die is a Retimer, the following rules apply to achieve Link match (if required):
   a. Retimer must not send {MBTRAIN.LINKSPEED done req} unless the target Link speed and width of the remote Retimer partner resolves to current Link and width. Proceed to Step 7 if Link match is achieved or if it is not required.
   b. While resolving the target Link speed and width with the remote Retimer partner, if a Retimer has received an {MBTRAIN.LINKSPEED done req}, it must send {MBTRAIN.LINKSPEED done resp} with stall encoding every 4 ms. UCIe Retimer must ensure that this stall is not perpetual, and an implementation specific timeout must be included in the Retimer.
   c. If the local UCIe Link speed or width is greater than the remote Retimer UCIe Link, then it must treat this as an error condition, and perform Step 3 or Step 5 with repair or speed degrade (whichever is applicable).
7. The UCIe Module must send {MBTRAIN.LINKSPEED done req} sideband message. When {MBTRAIN.LINKSPEED done req} is received, the UCIe Module must respond with {MBTRAIN.LINKSPEED done resp} and when a UCIe Module has sent and received the {MBTRAIN.LINKSPEED done resp} sideband message, both Transmitters and Receivers are now enabled and idle and both devices exit to LINKINIT.

4.5.3.4.13 MBTRAIN.REPAIR

This state can be entered from PHYRETRAIN or from MBINIT.LINKSPEED. For Advanced package, this state will be used to apply repair and for Standard package, this state will be used for Link width degrade. Track, Data, and Valid Transmitters are held low. Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking).

For Advanced Package, if the number of repair resources currently available is greater than the number of Lanes encountering errors, repair must be applied:

1. The UCIe Module sends the sideband message {MBTRAIN.REPAIR init req} for its Transmitter and the UCIe Module Partner responds with {MBTRAIN.REPAIR init resp}.
2. If Lane repair is possible, the UCIe Module applies repair on its Transmitter Lanes as described in Section 4.3.1 and sends {MBTRAIN.REPAIR Apply repair req} sideband message. The UCIe Module Partner applies repair as described in Section 4.3.1 and responds with {MBTRAIN.REPAIR Apply repair resp} sideband message once the required repair is applied.
3. The UCIe Module must send {MBTRAIN.REPAIR end req} sideband message and waits for a response. The UCIe Module Partner must then respond with {MBTRAIN.REPAIR end resp}. When a UCIe Module has sent and received {MBTRAIN.REPAIR end resp}, it must exit to MBTRAIN.TXSELFCAL.

For a Standard package, if the number of Lanes encountering errors are all contained within Lane 0 through Lane 7 or Lane8 through Lane 15 width must be degraded to a x8 Link (Lane 0 ... Lane 7 or Lane 8 ... Lane 15):

1. The UCIe Module sends the sideband message {MBTRAIN.REPAIR init req} and the Receiver responds with {MBTRAIN.REPAIR init resp}.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

2. The UCIe Module must send {MBTRAIN.REPAIR apply degrade req} indicating the functional Lanes on its Transmitter using one of the logical Lane map encodings from Table 4-9. If the remote Link partner indicated a width degrade in the functional Lanes, the UCIe Module must apply the corresponding width degrade to its Receiver. If the remote Link partner indicated all Lanes are functional, the UCIe Module sets its Transmitter and Receiver to the logical lane map corresponding to the functional Lane encoding determined on its Transmitter. The UCIe Module sends the {MBTRAIN.REPAIR apply degrade resp} after setting its Transmitter and Receiver lanes to the relevant logical lane map and proceeds to Step 3 if a degrade is possible or if all Lanes are functional. If a "Degrade not possible" (00b) encoding is sent or received in the {MBINIT.REPAIRMB apply degrade req}, the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.
3. The UCIe Module must send {MBTRAIN.REPAIR end req} sideband message and wait for a response. The UCIe Module Partner must then respond with {MBTRAIN.REPAIR end resp}. When UCIe Module has sent and received {MBTRAIN.REPAIR end resp}, it must exit to MBTRAIN.TXSELFCAL.

4.5.3.5 LINKINIT

This state is used to allow die to die adapter to complete initial Link management before entering Active state on RDI. See Section 8.1.6 for more details on RDI bring up flow. Track, Data, and Valid Transmitters are held low. Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking) if Strobe mode is negotiated. Clock Receivers are enabled.

Once RDI is in Active state, the PHY will clear its copy of "Start UCIe Link training" bit from UCIe Link control register.

The LFSR must be RESET upon entering this state.

This state is common for **Advanced** and **Standard Package interfaces**.

4.5.3.6 ACTIVE

Physical layer initialization is complete, RDI is in Active state and packets from upper layers can be exchanged between the two dies.

All data in this state is scrambled using the scrambler LFSR described in Section 4.4.1. Clock gating rules as described in Section 5.11 apply.

This state is common for **Advanced** and **Standard Packages**.

4.5.3.7 PHYRETRAIN

A die can enter PHY retrain for a number of reasons. Track, Data, and Valid Transmitters are held low. Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking). The trigger for PHY to enter PHY retrain is one of the following scenarios:

- Adapter directed PHY retrain: Adapter can direct the PHY to retrain for any reason it deems necessary.
- PHY initiated PHY retrain: Local PHY must initiate retrain on detecting a Valid framing error.
- Remote die requested PHY retrain: Local PHY must enter PHY retrain on receiving a request from the remote die.
- If a change is detected in Runtime Link Testing Control register during MBTRAIN.LINKSPEED.

A variable PHY_IN_RETRAIN must be set when entering PHYRETRAIN.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 4-10. Runtime Link Test Status Register based Retrain encoding

Link Test Status Register		Retrain Encoding Resolution
Busy bit value	Repair Required	
0b	N/A	TXSELFCAL
1b	Repair needed	REPAIR (If repair resources are available)
		SPEEDIDLE (if unrepairable)
1b	No Repair	TXSELFCAL

Table 4-11. Retrain encoding

Retrain Encoding	State	Retain Condition
001b	TXSELFCAL	No Lane errors (Valid framing errors detected by PHY)
010b	SPEEDIDLE	Lane errors & faulty Lanes cannot be repaired
100b	REPAIR	Lane errors & faulty Lanes are repairable

Table 4-12. Retrain exit state resolution

Retrain reg Condition		Retrain Request Encoding		Resolved State Encoding		Exit
Die 0	Die 1	Die 0	Die 1	Die 0	Die 1	Both Dies
No Lane Errors	No Lane Errors	001b	001b	001b	001b	MBTRAIN.TXSELFCAL
No Lane Errors	Repair	001b	100b	100b	100b	MBTRAIN.REPAIR
No Lane Errors	Speed Degrade	001b	010b	010b	010b	MBTRAIN.SPEEDIDLE
Repair	No Lane Errors	100b	001b	100b	100b	MBTRAIN.REPAIR
Repair	Repair	100b	100b	100b	100b	MBTRAIN.REPAIR
Repair	Speed Degrade	100b	010b	010b	010b	MBTRAIN.SPEEDIDLE
Speed Degrade	No Lane Errors	010b	001b	010b	010b	MBTRAIN.SPEEDIDLE
Speed Degrade	Repair	010b	100b	010b	010b	MBTRAIN.SPEEDIDLE
Speed Degrade	Speed Degrade	010b	010b	010b	010b	MBTRAIN.SPEEDIDLE

4.5.3.7.1 Adapter initiated PHY retrain

Following is the sequence of steps for an Adapter initiated PHY retrain:

1. UCIe Module receives retrain request from the local Adapter (RDI state req moved to Retrain). Following this, the UCIe Module must complete the stall Req/Ack (**`pl_stallreq`**; **`lp_stallack`**) hand shake on RDI as described in Chapter 8.0.
2. After completion of stall Req/Ack handshake and transmitting any pending data over mainband, UCIe Module must send sideband message {LinkMgmt.RDI.Req.Retrain} to UCIe Module Partner.
3. The UCIe Module Partner on receiving the sideband message {LinkMgmt.RDI.Req.Retrain} must transition its RDI state to Retrain after completion of stall Req/Ack (**`pl_stallreq`**; **`lp_stallack`**) handshake on its RDI and there is no mainband data pending in the Receiver pipeline. After completion of stall Req/Ack handshake and transmitting any pending data over mainband, the UCIe Module Partner responds with {LinkMgmt.RDI.Rsp.Retrain}.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4. Once {LinkMgmt.RDI.Rsp.Retrain} is received and there is no mainband data pending in the receiver pipeline, the UCIe Module must transition its RDI to Retrain.
5. UCIe Module must send {PHYRETRAIN.retrain start req} with retrain encoding reflecting the contents of Runtime Link Test Control register except the Start bit (if the Busy bit in Runtime Link Test Status Register is set). Following this, the UCIe Module Partner compares the received retrain encoding with the local retrain encoding. If received retrain encoding is the same as the local retrain encoding, the UCIe Module Partner must respond with {PHYRETRAIN.retrain start resp}. If the retrain encodings do not match, the UCIe Module Partner must resolve according to Retrain encodings and resolutions shown in Table 4-10, Table 4-11, and Table 4-12 and then send {PHYRETRAIN.retrain start resp} with the resolved retrain encoding.
6. Once UCIe Module sends and receives the sideband message {PHYRETRAIN.retrain start resp}, it must exit to corresponding training state according to the resolved retrain register encoding.

4.5.3.7.2 PHY initiated PHY retrain

Following is the sequence of steps for PHY initiated PHY retrain:

1. On detecting a valid framing error, the UCIe Module must assert **pl_error** when transmitting that flit (or flit chunk) on RDI. Following this the UCIe Module (PHY) must complete the stall Req/Ack (**pl_stallreq**; **lp_stallack**) handshake on RDI.
2. The UCIe Module must send sideband message {LinkMgmt.RDI.Req.Retrain}.
3. The UCIe Module Partner on receiving the sideband message {LinkMgmt.RDI.Req.Retrain} must transition its RDI to retrain after completion of stall Req/Ack (**pl_stallreq**; **lp_stallack**) handshake on its RDI. Following that the UCIe Module Partner responds with {LinkMgmt.RDI.Rsp.Retrain}.
4. Once {LinkMgmt.RDI.Rsp.Retrain} is received, the UCIe Module must transition its RDI to Retrain.
5. UCIe Module must send {PHYRETRAIN.retrain start req} with retrain encoding reflecting the contents of Runtime Link Test Control register except the Start bit (if the Busy bit in Runtime Link Test Status Register is set). Following this, the UCIe Module Partner compares the received retrain encoding with the local retrain encoding. If received retrain encoding is the same as the local retrain encoding, the UCIe Module Partner must respond with {PHYRETRAIN.retrain start resp}. If the retrain encodings do not match, the UCIe Module Partner must resolve according to Retrain encodings and resolutions shown in Table 4-10, Table 4-11, and Table 4-12 and then send {PHYRETRAIN.retrain start resp} with the resolved retrain encoding.
6. Once a UCIe Module has sent and received the sideband message {PHYRETRAIN.retrain start resp}, it must exit to corresponding training state according to the resolved retrain encoding.

4.5.3.7.3 Remote Die requested PHY retrain

1. On receiving {LinkMgmt.RDI.Req.Retrain}, the UCIe Module must transition local RDI to retrain after completion of stall Req/Ack (**pl_stallreq**; **lp_stallack**) handshake on its RDI. Following that the UCIe Module responds with {LinkMgmt.RDI.Rsp.Retrain}.
2. Once {LinkMgmt.RDI.Rsp.Retrain} is received, the UCIe Module Partner must transition its RDI to retrain.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3. UCIe Module must send {PHYRETRAIN.retrain start req} with retrain encoding reflecting the contents of Runtime Link Test Control register except the Start bit (if the Busy bit in Runtime Link Test Status Register is set). Following this, the UCIe Module Partner compares the received retrain encoding with the local retrain encoding. If received retrain encoding is the same as the local retrain encoding, the UCIe Module Partner responds with {PHYRETRAIN.retrain start resp}. If the retrain encodings do not match, the UCIe Module Partner must resolve according to Retrain encodings and resolutions shown in Table 4-10, Table 4-11, and Table 4-12 and then send {PHYRETRAIN.retrain start resp} with the resolved retrain encoding.
4. Once a die has sent and received the sideband message {PHYRETRAIN.retrain start resp}, it must exit to corresponding training state according to the resolved retrain encoding.

4.5.3.7.4 PHY retrain from LINKSPEED

1. The UCIe Module must send {PHYRETRAIN.retrain start req} with retrain encoding reflecting the contents of Runtime Link Test Control register except the Start bit (if the Busy bit in Runtime Link Test Status Register is set). Following this, the UCIe Module Partner compares the received retrain encoding with the local retrain encoding. If received retrain encoding is the same as the local retrain encoding, the UCIe Module Partner must respond with {PHYRETRAIN.retrain start resp}. If the retrain encodings do not match, the UCIe Module Partner must resolve according to Retrain encodings and resolutions shown in Table 4-10, Table 4-11, and Table 4-12 and then send {PHYRETRAIN.retrain start resp} with the resolved retrain encoding.
2. Once a die has sent and received the sideband message {PHYRETRAIN.retrain start resp}, it must exit to corresponding training state according to the resolved retrain encoding.

4.5.3.8 TRAINERROR

This state used as a transitional state due to any fatal or non-fatal events that need to bring the state machine back to RESET state. This can happen during initialization and training or if "Start UCIe Link training" bit from UCIe Link control register is set when state machine is not in RESET. It is also used for any events that transition the Link from a Link Up to a Link Down condition. All mainband (Track, Data, and Valid) and Clock transmitters are tri-stated, and their receivers are permitted to be disabled.

The exit from TRAINERROR to RESET is implementation specific. For cases when there is no error escalation (i.e., RDI is not in LinkError), it is recommended to exit TRAINERROR as soon as possible. For cases when there is error escalation (i.e., RDI is in LinkError), it is required for Physical Layer to be in TRAINERROR as long as RDI is in LinkError.

See Chapter 8.0 for correctable, non-fatal, and fatal error escalation on RDI.

This state is common for **Advanced** and **Standard Packages**.

If sideband is Active, a sideband handshake must be performed for both UCIe Module and UCIe Module Partner to enter TRAINERROR state from any state other than SBINIT. The following is defined as the TRAINERROR handshake:

- The UCIe Module requesting exit to TRAINERROR must send {TRAINERROR Entry req} sideband message and wait for a response. The UCIe Module Partner must exit to TRAINERROR and respond with {TRAINERROR Entry resp}. Once {TRAINERROR Entry resp} sideband message is received, the UCIe Module must exit to TRAINERROR. If no response is received for 8ms, the LTSM transitions to TRAINERROR.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.5.3.9 L1/L2

PM state allows a lower power state than dynamic clock gating in ACTIVE. All mainband (Track, Data, and Valid) and Clock transmitters are tri-stated, and their receivers are permitted to be disabled.

- This state is entered when RDI has transitioned to PM state as described in Chapter 8.0. The PHY power saving features in this state are implementation specific.
- When local Adapter requests Active on RDI or remote Link partner requests L1 exit the PHY must exit to MBTRAIN.SPEEDIDLE. L1 exit is coordinated with the corresponding L1 state exit transitions on RDI.
- When local Adapter requests Active on RDI or remote Link partner requests L2 exit the PHY must exit to RESET. L2 exit is coordinated with the corresponding L2 state exit transitions on RDI.

4.6 Runtime Recalibration

Track signal can be used by the Receiver to perform periodic runtime calibration while in ACTIVE. Mainband data must continued to be sampled and processed (when accompanied by correct valid framing) during runtime recalibration. For unterminated Link, when not sending the required pattern the Track signal must alternate between being held low and held high (for anti aging) across consecutive Track recalibration iterations. For a terminated Link, when not sending the required pattern, the Track transmitter must go to Hi-Z.

The following sequence is used to request track pattern:

1. The UCIe Module enables the Track signal buffers on its Receiver and sends a {RECAL.track pattern init req} sideband message, and then waits for a response.
2. The UCIe Module Partner sends {RECAL.track pattern init resp} and enables its Track signal Transmitter (is preconditioned to drive low if needed). Following this, the UCIe Module Partner's Track Transmitter starts sending the pattern described in Section 5.5.1, along with the forwarded clock. If the link is in Clock-gated mode, the UCIe Module Partner should enable the clock and manage whether the Link should return to Clock-gated mode after the Track update is complete.
3. The UCIe Module on its Receiver performs the required recalibration and sends the {RECAL.track pattern done req} sideband message.
4. Upon receiving this message, the UCIe Module Partner's Track Transmitter stops sending the pattern and sends the {RECAL.track pattern done resp} sideband message.
5. The UCIe Module is permitted to disable the Track Receiver upon receiving the{RECAL.track pattern done resp} sideband message.

4.7 Multi-module Link

As described in Chapter 1.0, the permitted configurations for a multi-module Link are one-, two-, and four-module configurations. In a multi-module Link, each module is assigned a dedicated Module Identifier (Module ID), which is advertised to the remote Link partner during MBINIT.PARAM. Chapter 5.0 defines the permitted combinations of Module ID assignments for the different scenarios of Multi-module instantiations that must be supported by multi-module implementations.

4.7.1 Multi-module initialization

Each module in a multi-module configuration must initialize and train independently, using its sideband. If two or four modules are used, a separate multi-module PHY logic block coordinates across the modules, as described in Section 1.2.2. The MMPL is responsible for orchestrating data transfer and any associated byte swizzling for the Transmitters across the multiple modules such that the remote Link partner's Receivers observe the correct byte-to-Lane mapping (i.e., for any valid

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

transfer, bytes are laid out from LSB to MSB in ascending order of Module ID and Lane ID across all the active Lanes). Figure 4-34, Figure 4-35, Figure 4-36, and Figure 4-37 illustrate examples of the aforementioned byte swizzling for some of the Standard Package configurations. M0, M1, M2, and M3 in the figures correspond to Module ID 0, Module ID 1, Module ID 2, and Module ID 3, respectively. The figures provide the RDI byte-to-Module mapping. Figure 4-34 shows the scenario in which the remote Link partner's Module ID is the same. Figure 4-35 shows an example of a scenario in which the remote Link partner's Module ID is different. Figure 4-36 shows an example of width degradation for Standard package in which the remote Link partner's Module ID is different (note that the bytes are laid out in ascending order of Module ID and Lane ID across all the active Lanes at the Receiver in all cases). Figure 4-37 shows a scenario in which two modules are disabled. This corresponds to a case outlined in Chapter 5.0 in which a stacked configuration is connected with an unstacked configuration and the M0 and M2 modules are disabled; the remaining bytes of RDI are sent over subsequent 8-UI intervals such that M1 on the remote Link partner receives the least significant bytes.

Figure 4-34. Example of Byte Mapping for Matching Module IDs

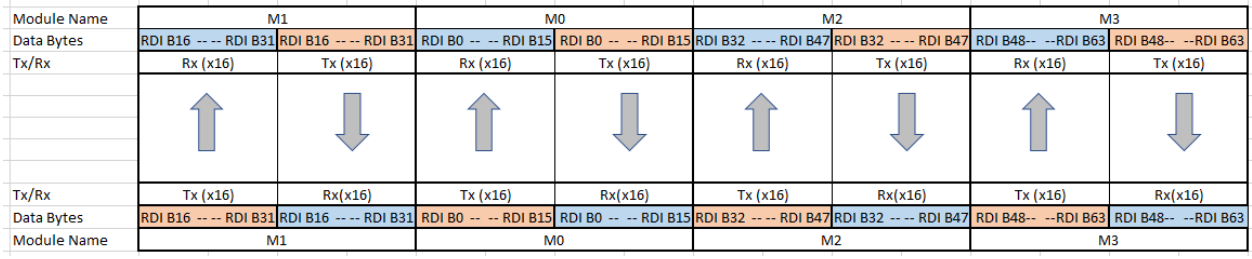


Figure 4-35. Example of Byte Mapping for Differing Module IDs

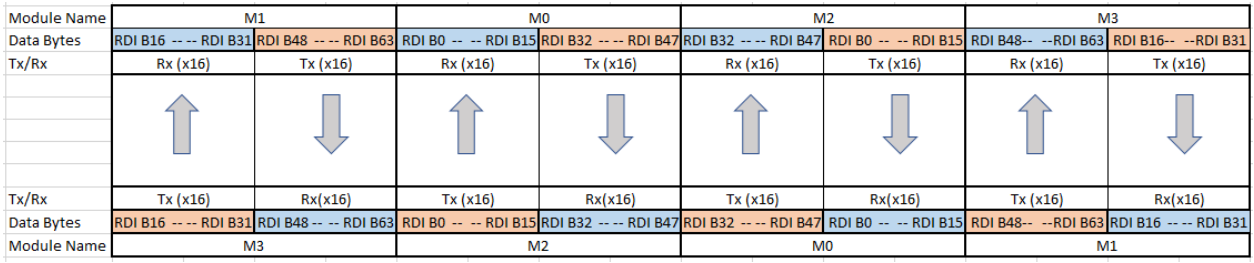
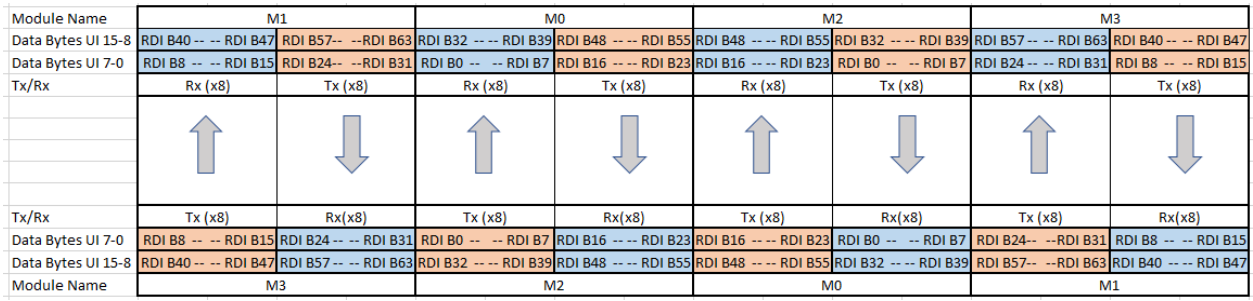
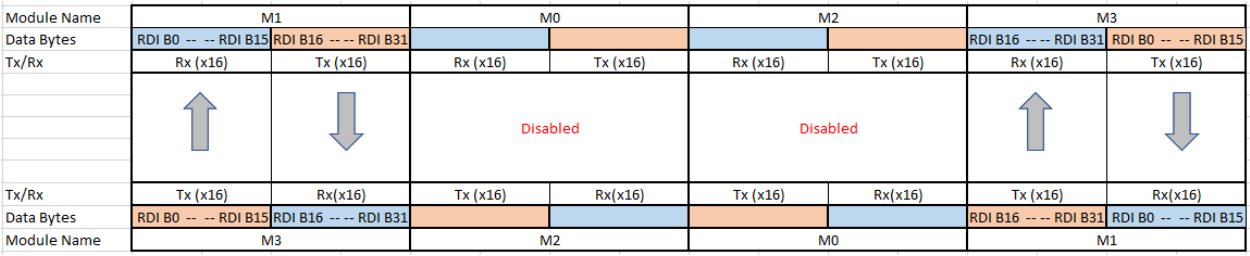


Figure 4-36. Example of Width Degradation with Byte Mapping for Differing Module IDs



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-37. Example of Byte Mapping with Module Disable



Each module in a multi-module Link must operate at the same width and speed. During initialization or retraining, if any module failed to train, the MMPL must ensure that the multi-module configuration degrades to the next permitted configuration for width or speed degrade (see Figure 4-38 and Figure 4-39). Subsequently, any differences in width and speed between the different modules must be resolved using the following rules:

1. For Standard package multi-module configuration, if width degrade is reported for any of the modules:
   a. If less than or equal to half the number of modules report width degrade at the current Link speed, the corresponding Modules must be disabled. The MMPL must ensure that the multi-module configuration degrades to the next permitted configuration for width or speed degrade (see Figure 4-38 and Figure 4-39). For example, if three out of four modules are active, the MMPL must degrade the Link to a two-module configuration.
   b. If the majority of modules report width degrade at the current Link speed, refer to the pseudo code below:

```
CLS: Current Link Speed
CLS-1: Next lower allowed Link Speed
M is number of active Modules
Aggregate Raw BW(M, CLS) = Common Minimum Link width * M * CLS
If modules report Width degrade:
       If CLS = 4 GT/s
              Apply Width degrade for all modules
       Else If Aggregate Raw BW(M, (CLS-1)) > Aggregate raw BW (M/2, CLS):
              Attempt Speed Degrade
       Else:
              Apply Width degrade for all modules
```

2. For Advanced or Standard package multi-module configuration, if any Module reports speed difference, refer to the pseudo code below:

```
IF modules report speed difference:
       CMLS: Common Maximum Link Speed
       HMLS: Highest Maximum Link Speed of next lower configuration
       IF HMLS/2 > CMLS:
              Modules degrade to next lower configuration
       Else:
              Speed for all modules degrades to CMLS
```

Figure 4-38 and Figure 4-39 provide a consolidated view of the above two rules as a flow chart that Advanced Package and Standard Package implementations, respectively, must follow. Note that the "Yes" condition for HMLS/2 > CMLS question is there to cover the base case of 4 GT/s. In other words, if some module(s) passed MBINIT but failed 4 GT/s in LinkSpeed, then the "Yes" arc will result in module disable instead of TrainError (because CMLS will be 0 for that) and provide the opportunity to remain operational at 4 GT/s for the modules that were still operational at 4 GT/s.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-38. Decision Flow Chart for Multi-module Advanced Package

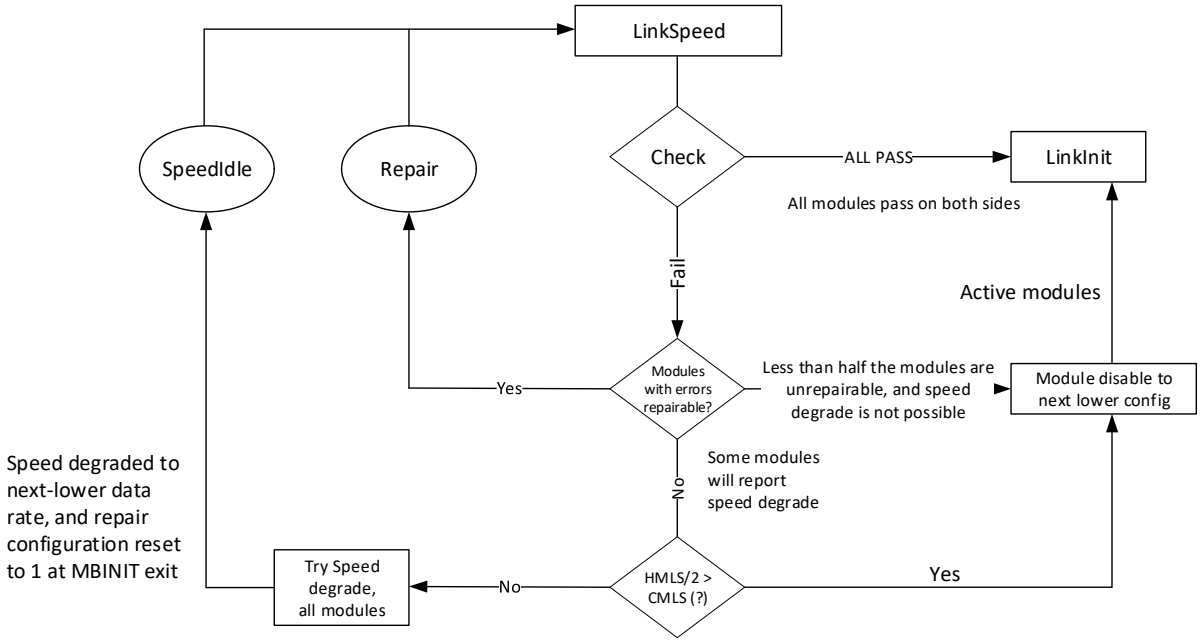
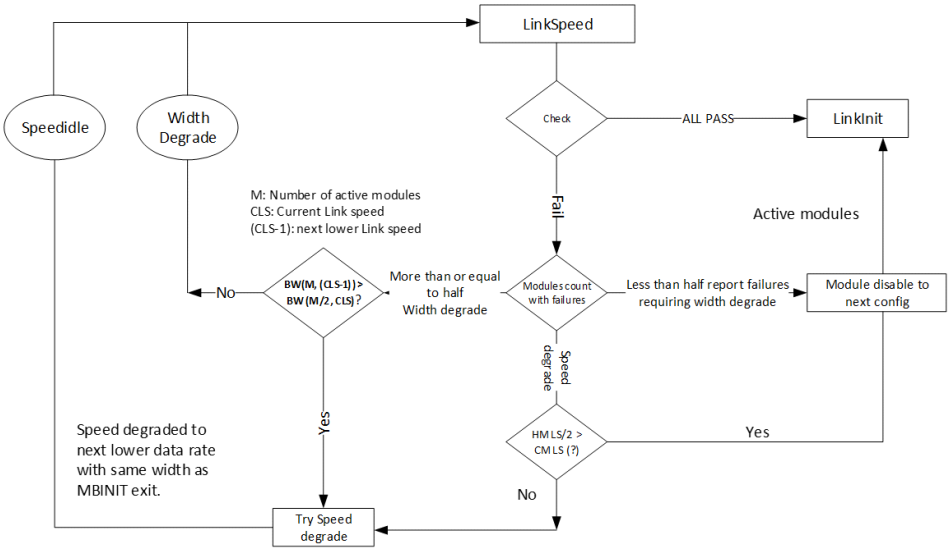


Figure 4-39. Decision Flow Chart for Multi-module Standard Package



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

4.7.1.1 Sideband Assignment and Retimer Credits for Multi-module Configurations

During Link initialization, training and retraining (see Section 4.5) all LTSM handshake-related sideband messages are sent on individual module sideband interfaces.

Note: For all other sideband messages from upper layers or related to RDI state transitions, a single sideband is used to send and receive sideband messages. A device must send sideband messages on the sideband interface of the numerically least Module ID whose LTSM is in Active state. A message sent on a given Module ID could be received on a different Module ID on the sideband Receiver.

Similarly, Retimer credits are returned on the Valid signal of the numerically least Module ID whose LTSM is in Active state. Credits sent on a given Module ID could be received on a different Module ID on the remote Link partner.

4.7.2 Multi-module Interoperability between x64 and x32 Advanced Packages

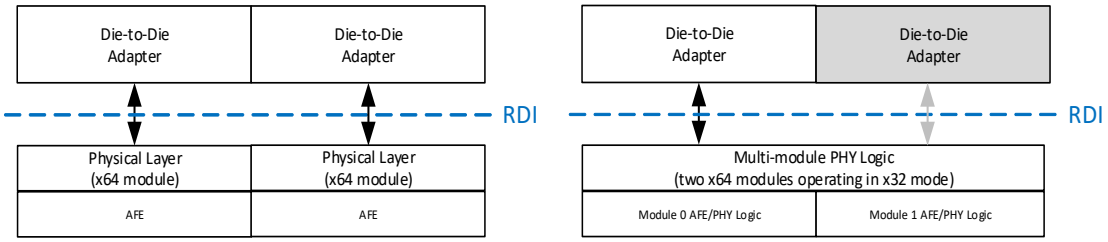
MMPL is responsible for the appropriate byte swizzling and width adjustment when a multi-module x64 Advanced Package module is connected to a corresponding multi-module x32 Advanced Package module. All the modules in a multi-module configuration must be of the same type (in this context, all the modules within a multi-module set must be x64 Advanced, or x32 Advanced). All the rules related to module naming conventions and disabled configurations apply to x32 Advanced Package Modules as well.

One example of interoperation between UCIe-A x64 and UCIe-A x32 is when the UCIe-A x64 Stack (including RDI and FDI maximum throughput) is bandwidth-matched (Full Width Mode) by the remote Link partner's maximum throughput for a given interface. Figure 4-40 shows an example of two x64 modules that are capable of operating as two independent UCIe stacks with independent Adapters and Protocol Layers (bypass MMPL logic in configuration (a) in Figure 4-40) and is also capable of operating as a multi-module configuration when connected to a corresponding multi-module configuration of x32 Advanced Package to achieve the equivalent bandwidth of a single x64 module (configuration (b) in Figure 4-40). In the latter configuration, one of the Adapters (shown in gray) is disabled.

Software and firmware is permitted to use UCIe DVSEC Link Capability and Control registers to determine within which configuration to train the link.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 4-40. Implementation Example Showing Two Different Operating Modes of the Same Hardware Implementation



Two D2D stacks enabled; MMPL bypassed and each stack uses one module and operates in Full Width mode
For each stack,

UCIe Link DVSEC Capability Register	Max Link Width	x64
	APMW	0
Link Training Parameter	UCIe-A x32	0

(a)

Only one D2D stack enabled; Uses MMPL with two modules and operates in Full Width mode

UCIe Link DVSEC Capability Register	Max Link Width	x64
	APMW	1
Link Training Parameter	UCIe-A x32	1

(b)

Another example of interoperation between UCIe-A x64 and UCIe-A x32 is when the UCIe-A x64 Stack degrades bandwidth (Degraded Width Mode) to match the remote Link partner's maximum throughput. Figure 4-41 shows an example of RDI byte-to-module assignments for a four-module set of x64 Advanced Package modules interoperating with a four-module set of x32 Advanced Package modules. The example is for a 256B RDI width on the x64 set, and a 128B RDI on the x32 set. On the Transmitter side of x64 modules, the MMPL throttles RDI, as required, because the MMPL can only send half the bytes over 8 UI; and on the Receiver side, the MMPL accumulates 16 UI worth of data before forwarding it over RDI (assumes data transfers are in chunks of 256B OR appropriate pause of data stream indications are applied and detected by MMPLs/Adapters within the data stream).

Figure 4-41. RDI Byte-to-Module Assignment Example for x64 Interop with x32

	M1		M0		M2		M3	
UI 8 - 15	RDI B160 -- -- RDI B191	RDI B224 -- -- RDI B255	RDI B128 -- -- RDI B159	RDI B192 -- -- RDI B223	RDI B192 -- -- RDI B223	RDI B128 -- -- RDI B159	RDI B224 -- -- RDI B255	RDI B160 -- -- RDI B191
UI 0 - 7	RDI B32 -- -- RDI B63	RDI B96 -- -- RDI B127	RDI B0 -- -- RDI B31	RDI B64 -- -- RDI B95	RDI B64 -- -- RDI B95	RDI B0 -- -- RDI B31	RDI B96 -- -- RDI B127	RDI B32 -- -- RDI B63
	Rx (x32)	Tx (x32)	Rx (x32)	Tx (x32)	Rx (x32)	Tx (x32)	Rx (x32)	Tx (x32)
	Tx (x64)	Rx(x64)	Tx (x64)	Rx(x64)	Tx (x64)	Rx(x64)	Tx (x64)	Rx(x64)
UI 0 - 7	RDI B32 -- -- RDI B63	RDI B96 -- -- RDI B127	RDI B0 -- -- RDI B31	RDI B64 -- -- RDI B95	RDI B64 -- -- RDI B95	RDI B0 -- -- RDI B31	RDI B96 -- -- RDI B127	RDI B32 -- -- RDI B63
UI 8 - 15	RDI B160 -- -- RDI B191	RDI B224 -- -- RDI B255	RDI B128 -- -- RDI B159	RDI B192 -- -- RDI B223	RDI B192 -- -- RDI B223	RDI B128 -- -- RDI B159	RDI B224 -- -- RDI B255	RDI B160 -- -- RDI B191
	M3		M2		M0		M1	

For the example shown in Figure 4-41, a single Adapter is operating with all four Modules. The D2D stack uses MMPL with 4 modules, with each of the x64 modules operating in Degraded Width Mode,

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

and only 32 lanes routed per module. The corresponding values in the capability register and Link Training parameter are as listed in Table 4-13.

Table 4-13. Capability Register and Link Training Parameter Values for RDI Byte-to-Module Assignment Example for x64 Interop with x32

UCIe Link DVSEC Capability Register	Max Link Width	x128
	APMW	1
Link Training Parameter	UCIe-A x32	1

See Section 5.7.2.4 for comprehensive rules of interoperation between x64 and x32 Advanced Package modules.

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.0 Electrical Layer

Key attributes of electrical specification include:

- Support for 4, 8, 12, 16, 24, and 32 GT/s data rates
- Support for Advanced and Standard package interconnects
- Support for clock and power gating mechanisms
- Single-ended unidirectional data signaling
- DC coupled point-to-point interconnect
- Forwarded clock for transmit jitter tracking
- Matched length interconnect design within a module
- Tx driver strength control and unterminated Rx for Advanced Package
- Tx termination and data rate and channel reach dependent Rx termination for Standard Package

5.1 Interoperability

5.1.1 Data rates

A device must support 4 GT/s and all the data rates data rates between 4 GT/s and the highest supported data rate. For example, a device supporting 16 GT/s must also support 4, 8, 12 GT/s Data rates. Spread-Spectrum Clocking (SSC) is allowed, but no ppm difference is allowed between the local Transmitter and the remote Receiver (see Figure 5-12).

5.2 Overview

5.2.1 Interface Overview

High-level block diagrams of UCIe PHY are shown in Figure 5-1 and Figure 5-2. The UCIe physical interface consists of building blocks called Modules. A Module that uses advanced packaging technology (e.g., EMIB, CoWoS) called "Advanced Package Module" consists of a pair of clocks, 64 or 32 single-ended data Lanes for x64 or x32 Advanced Package Module, respectively, a data valid Lane each direction (transmit and receive) and a Track Lane. There is a low-speed sideband bus for initialization, Link training, and configuration reads/writes. The sideband consists of a single-ended sideband data Lane and single-ended sideband clock Lane in both directions (transmit and receive).

The "Standard Package Module" uses a traditional Standard packaging with larger pitch. A Standard Package Module consists of a pair of clocks, 16 single-ended data Lanes, a data valid Lane and Track Lane in each direction (transmit and receive). There is a low-speed sideband bus for initialization, Link training, and configuration reads/writes. The sideband consists of a single-ended sideband data Lane and single-ended sideband clock Lane in both directions (transmit and receive).

For some applications, multiple modules (2 or 4) can be aggregated to deliver additional bandwidth.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

To avoid reliability issues, we recommend limiting the Transmitter output high (V_{OH}) to a maximum of 100mV above the receiving chiplet's Receiver front end circuit power supply rail. Over-stress protection circuit may be implemented in Receiver when Transmitter output high is more than 100mV above Receiver power supply rail.

Figure 5-1. x64 and x32 Advanced Package Module

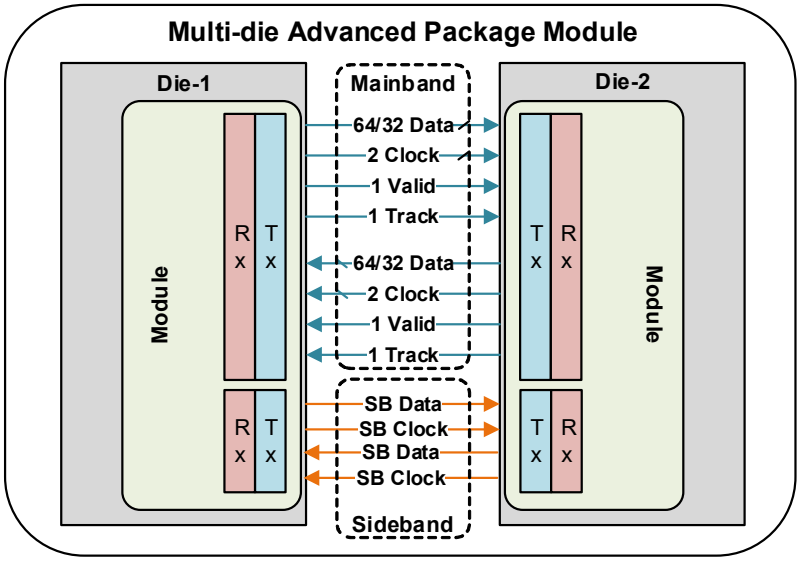
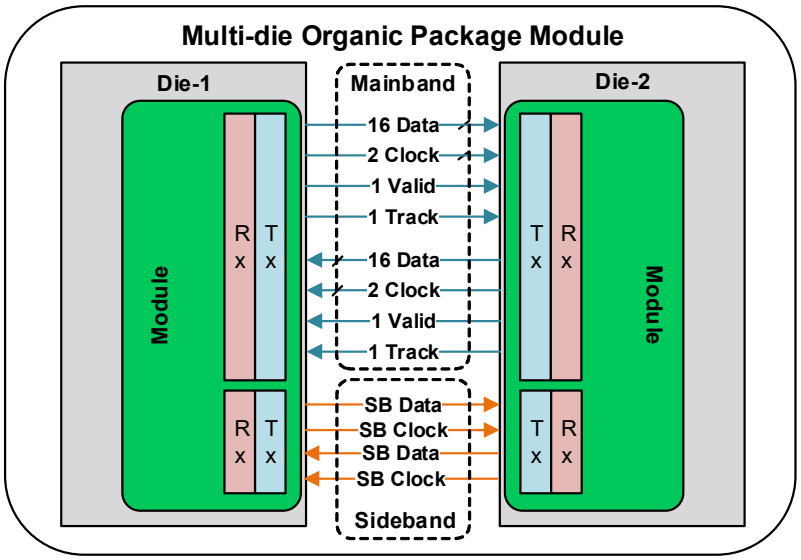


Figure 5-2. Standard Package Module



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.2.2 Electrical summary

Table 5-1 defines the PHY electrical characteristics of a UCIe device.

Table 5-1. Electrical summary

Parameter	Advanced Package (x64)			Standard Package			
Data Width (per module)	64	64	64	16	16	16	16
Data Rate (GT/s)	4/8/12	16	24/32	4-16	4/8/12	16	24/32
Power Efficiency Target (pJ/b)	See Table 1-3						
Latency Target (TX+RX) (UI)[1] (Target upper bound)	12	12	16	12	12	12	16
Idle Exit/Entry Latency (ns) (target upper bound)	0.5	1	1	0.5	0.5	1	1
Idle Power (% of peak power) (target upper bound)	15	15	15	15	15	15	15
Channel Reach (mm)	2	2	2	2-10	25	25	25
Die Edge Bandwidth Density (GB/s/mm)[2]	See Table 1-3						
Bandwidth area density (GB/s/mm^2)	158/316/473	631	710/947	21-85	21/42/64	85	109/145
PHY dimension width (um)[3]	388.8	388.8	388.8	571.5[4]	571.5[4]	571.5[4]	571.5[4]
PHY dimension Depth (um)[5]	1043	1043		1320	1320	1320	1540
ESD[6]	30V CDM (Anticipating going to 5-10V in Future.)						

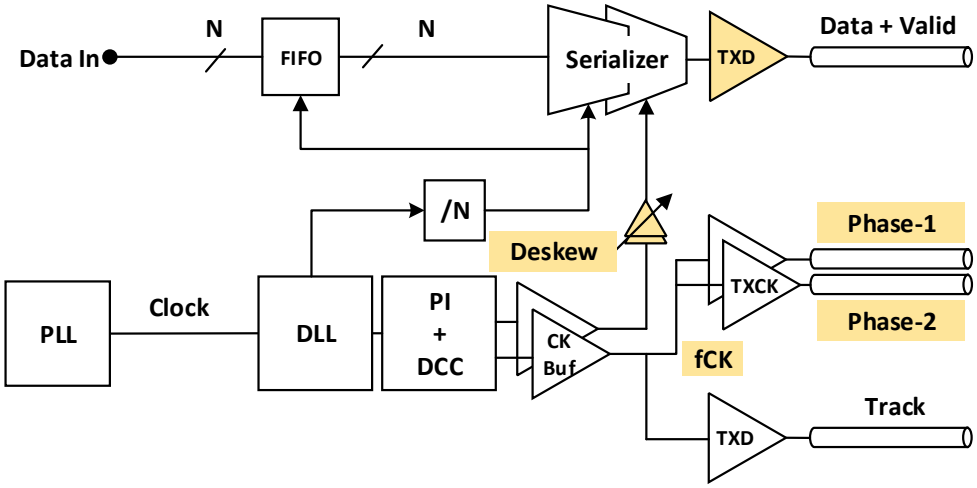
1. Electrical PHY latency target. For overall latency target, see Table 1-3.
2. Die edge bandwidth density defined as total I/O bandwidth in GB per sec per mm silicon die edge. Bandwidth density shown for x64 Advanced Package module. For x32 Advanced Package module, the Die Edge Bandwidth density is 50% of the corresponding value for x64.
3. For compatibility, PHY dimension width must match spec for Advanced Package. Tolerance of PHY dimension width for Standard Package can be higher because there is more routing flexibility. For best channel performance, it's recommended for width to be close to spec.
4. Standard Package PHY dimension width is the effective normalized width of one (x16) module based on x32 interface (see Figure 5-35 and Figure 5-36) with reference design 1 in Figure 5-38.
5. PHY dimension depth is an informative parameter and depends on bump pitch. Number in the table is based on 45-um bump pitch for 10-column x64 Advanced Package and 100-um bump pitch for Standard Package. See Section 5.7.2 for informative values of PHY dimension depth for combinations of the x64 and x32 Advanced Package modules in 10-column, 16-column, and 8-column bump matrix construction.
6. Reference (Industry Council on ESD Target Levels): White Paper 2: A Case for Lowering Component-level CDM ESD Specifications and Requirements.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.3 Transmitter Specification

The Transmitter topology is shown in Figure 5-3. Each data module consists of N single-ended data Transmitters plus a Valid signal. N is 68 (64 Data + 4 Redundant Data) for x64 Advanced Package. N is 34 (32 Data + 2 Redundant Data) for a x32 Advanced Package. N is 16 for Standard Package. There is a pair of Transmitters for clocking and a Track signal in each module. The clock rates and phases are discussed in detail in Section 5.5.

Figure 5-3. Transmitter



The Valid signal is used to gate the clock distribution to all data Lanes to enable fast idle exit and entry. The signal also serves the purpose of Valid framing, see Section 4.1.2 for details. The Transmitter implementation for Valid signal is expected to be the same as for regular Data.

The Track signal can be used for PHY to compensate for slow-changing variables such as voltage or temperature. Track is a unidirectional signal similar to a data bit. The UCIe Module sends a clock pattern (1010…) aligned with Phase-1 of the forwarded clock signal on its Track Transmitter when requested over the sideband by the UCIe Module Partner for its Track Receiver. See Section 4.6 for more details.

5.3.1 Driver Topology

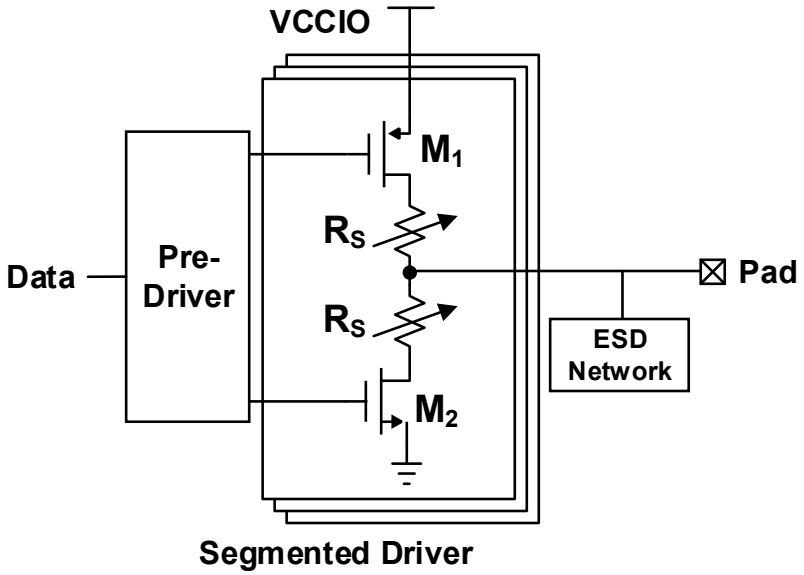
The Transmitter is optimized for simplicity and low power operation. An example of a low power Transmitter driver is shown in Figure 5-4. Separate pull-up and pull-down network strengths are permitted to achieve optimal performance across different channel configurations.

A control loop or training is recommended to adjust output impedance to compensate for the process, voltage and temperature variations. Control loop and training are implementation specific and beyond the scope of this specification. In low power states, the implementation must be capable of tri-stating the output.

It is recommended to optimize the ESD network to minimize pad capacitance. Inductive peaking technique such as T-coil may be needed at higher data rates.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-4. Transmitter driver example circuit



5.3.2 Transmitter Electrical parameters

Table 5-2 defines the Transmitter electrical parameters.

Table 5-2. Transmitter Electrical Parameters (Sheet 1 of 2)

Parameter	Min	Typ	Max	Unit
Data Lane TX Swing[1]	0.4			V
Fwd Clock Tx Swing (single ended)	0.4			V
Incoming Clock Rise/Fall time[2]	0.1	0.22	0.25	UI
Incoming Differential Clock Overlap[2]	-	-	30	mUI
Incoming Data Rise/Fall time[2]	-	0.35	-	UI
Driver Pull-up/Pull-down Impedance for Advanced Package[3]	22	25	28	Ohms
Impedance Step Size for Advanced Package[4]			0.5	Ohms
Driver Pull-up/Pull-down Impedance for Standard Package[5]	27	30	33	Ohms
Impedance Step Size for Standard Package[4]	-	-	0.5	Ohms
1-UI Total Jitter[6]	-	-	96/113	mUI pk-pk
1-UI Deterministic Jitter (Dual Dirac)[7]	-	-	48	mUI pk-pk
Tx Data/clock Differential Jitter (Divergent Path)[8]			60	mUI pk-pk
Duty Cycle Error[9]	-0.02	-	0.02	UI
Lane-to-Lane Skew Correction Range (up to 16 GT/s)[10]	-0.1	-	0.1	UI
Lane-to-Lane Skew Correction Range (up to 32 GT/s)[10]	-0.15	-	0.15	UI
Lane-to-Lane Skew Correction Range (up to 16 GT/s)[11]	-0.14	-	0.14	UI
Lane-to-Lane Skew Correction Range (up to 32 GT/s)[11]	-0.22	-	0.22	UI
Lane-to-Lane Skew[9]	-0.02	-	0.02	UI
Clock to Mean Data Training Accuracy[12]	-0.07	-	0.07	UI
Phase Adjustment Step[13]	-	-	16	mUI

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-2. Transmitter Electrical Parameters (Sheet 2 of 2)

Parameter	Min	Typ	Max	Unit
TX Pad Capacitance (for all speeds)[14]	-	-	250	fF
TX Pad Capacitance (8 GT/s capable design)[14]	-	-	300	fF
TX Pad Capacitance (16 GT/s capable design)[15]	-	-	200	fF
TX Pad Capacitance (32 GT/s capable design)[15]	-	-	125	fF

1. For recommended maximum Transmitter voltage, see Section 1.5.
2. Expected input (informative). Measured 20% to 80%. Differential clock overlap is deviation from the ideal differential phase (180 degrees apart).
3. Driver pull-up/down impedance is calibrated at midpoint of Transmitter signal swing.
4. Impedance step size is an informative parameter and can be implementation specific to meet Driver pull-up/pull-down impedance.
5. Driver pull-up/pull-down impedance is calibrated at midpoint of Transmitter signal swing (with nominal Rx termination when applicable).
6. At BER 1E-15/1E-27.
7. Data dependent jitter excluding Duty Cycle Error.
8. Includes absolute random jitter and untracked deterministic jitter of the divergent path due to delay mismatch (in the matched architecture).
9. Post correction.
10. Advanced Package.
11. Standard Package.
12. Includes static and tracking error.
13. Informative parameter. Phase adjustment step size must be chosen to meet other timing parameters, including Clock-to-Mean Data Training Accuracy, Lane-to-Lane skew, and Duty cycle error (if applicable).
14. Effective pad capacitance Advanced Package.
15. Effective pad capacitance Standard Package.

5.3.3 24 and 32 GT/s Transmitter Equalization

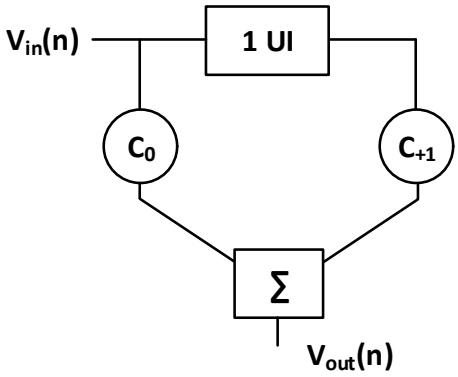
Transmitter equalization is recommended for 16 GT/s and must be supported at 24 GT/s and 32 GT/s data rates to mitigate the channel ISI impact. Tx equalization is de-emphasis only for all applicable Data rates.

Tx equalization coefficients for 24 and 32 GT/s are based on the FIR filter shown in Figure 5-5. Equalization coefficient is subject maximum unity swing constraint.

The Transmitter must support the equalization settings shown in Table 5-3. Determination of de-emphasis setting is based on initial configuration or training sequence, where the value with larger eye opening will be selected.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-5. Transmitter de-emphasis



$$V_{out}(n) = C_0 V_{in}(n) + C_{+1} V_{in}(n-1) + |C_{+1}|$$

$$|C_0| + |C_{+1}| = 1$$

$$V_{in}(n) = \{0, +1\}$$

Figure 5-6. Transmitter de-emphasis waveform

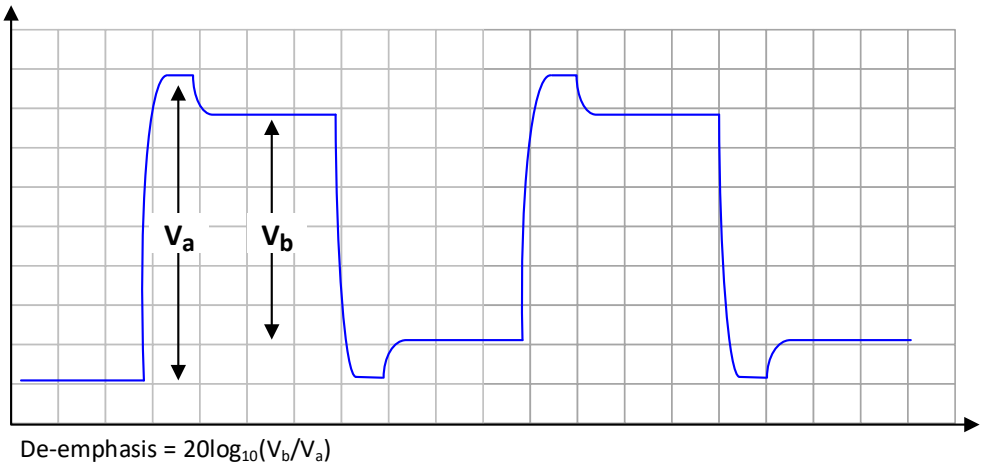


Table 5-3. Transmitter de-emphasis values

Setting	De-emphasis	Accuracy	C_{+1}	V_b/V_a
1	0.0 dB	-	0.000	1.000
2	-2.2 dB	+/- 0.5 dB	-0.112	0.776

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.4 Receiver Specification

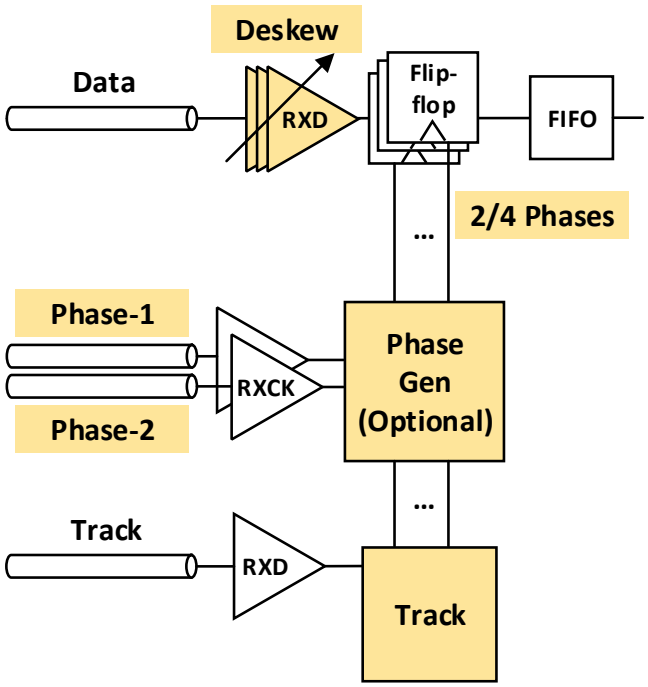
The Receiver topology is illustrated in Figure 5-7. Each Module (Advanced Package and Standard Package) consists of clocks Receivers, data Receivers, and Track Receiver.

The received clock is used to sample the incoming data. The Receiver must match the delays between the clock path and the data/valid path to the sampler. This is to minimize the impact of power supply noise induced jitter. The data Receivers may be implemented as 2-way or 4-way interleaved. For 4-way interleaved implementation the Receiver needs to generate required phases internally from the two phase of the forwarded clock. This may require duty cycle correction capability on the Receiver. The supported forwarded clock frequencies and phases are described in Section 5.5.

At higher data rates, deskew capability may be needed in the receiver to achieve the matching requirements between the data Lanes. Receiver Deskew, when applicable, can be performed during mainband training. More details are provided in Section 4.5.

The UCIe Module, upon requesting the Track signal, receives a clock pattern (1010…) aligned with Phase-1 of the forwarded clock signal on its Track Receiver from the UCIe Module Partner's Track Transmitter and may use the Track signal to track the impact of slow varying voltage and temperature changes on sampling phase.

Figure 5-7. Receiver topology



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.4.1 Receiver Electrical Parameters

The specified Receiver electrical parameters are shown in Table 5-4.

Table 5-4. Receiver Electrical parameters

Parameter	Min	Typ	Max	Unit
RX Input Impedance[1]	45	50	55	Ohms
Impedance Step Size[1]	-	-	1	Ohms
Data/Clock Total Differential Jitter[2] [3]	-	-	60	mUI pk-pk
Lane-to-Lane skew (up to 16 GT/s)[4]	-0.07	-	0.07	UI
Lane-to Lane skew (> 16 GT/s)[4]	-0.12	-	0.12	UI
Phase error[5] (Including Duty cycle error and in-phase quadrature mismatch)	-0.04	-	0.04	UI
Per-Lane deskew adjustment step[6]	-	-	16	mUI
Output Rise Time[7]	-	-	0.1	UI
Output Fall Time[7]	-	-	0.1	UI
RX Pad Capacitance[8]	-	-	200	fF
RX Pad Capacitance (up to 8 GT/s)[1]	-	-	300	fF
RX Pad Capacitance (up to 16 GT/s)[1] [9]	-	-	200	fF
RX Pad Capacitance (24 and 32 GT/s)[1] [9]	-	-	125	fF
Rx Voltage sensitivity	-	-	40	mV

1. Standard Package mode with termination. Impedance step size is an informative parameter and can be implementation specific to meet Rx Input Impedance.
2. Based on matched architecture.
3. Includes absolute random jitter and untracked deterministic jitter of the divergent path due to delay mismatch (in the matched architecture).
4. Require Rx per-Lane deskew if limit is exceeded.
5. Residual error post training and correction.
6. When applicable (informative).
7. Expected output (informative). Measured 20% to 80%.
8. Advanced Package.
9. Effective Pad capacitance.

5.4.2 Rx Termination

Rx termination is applicable only to Standard Package modules. All Receivers on Advanced Package modules must be unterminated.

Receiver termination on Standard Package is data rate and channel dependent. Table 5-5 shows the maximum data rate and channel reach combinations for which the Receivers in Standard Package Modules are recommended to remain unterminated for a minimally compliant Transmitter. Figure 5-8 shows an alternate representation of termination requirement. The area below the curve in Figure 5-8 shows the speed and channel-reach combinations for which the Receivers in Standard Package Modules are recommended to remain unterminated. Termination is required for all other combinations. Receivers must be ground-terminated when applicable, as shown in Figure 5-9.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-5. Maximum channel reach for unterminated Receiver (Tx Swing = 0.4 V)

Data Rate (GT/s)	Channel Reach (mm)
12	3
8	5
4	10

Figure 5-8. Receiver Termination Map for Table 5-5 (Tx Swing = 0.4 V)

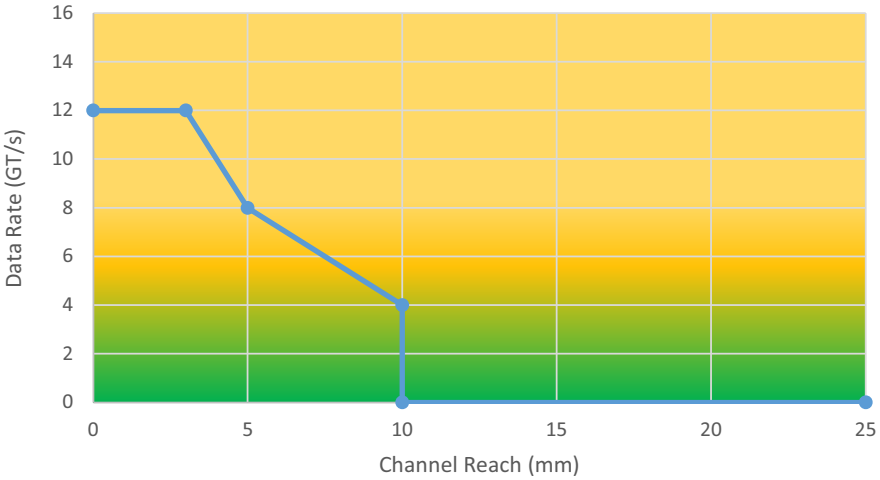
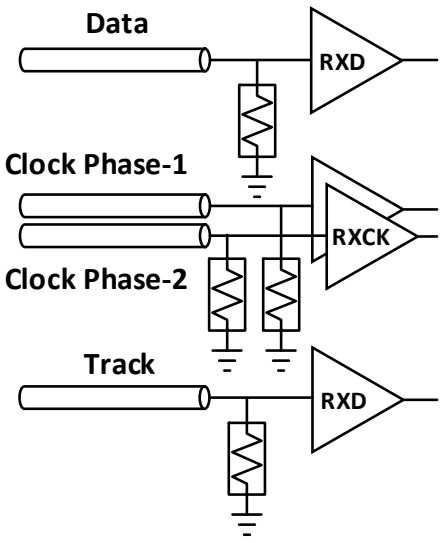


Figure 5-9. Receiver termination



For higher Transmitter swing, unterminated Receiver can be extended to longer channel and high data rate. Table 5-6 shows the maximum data rate and channel reach combinations for Transmitter swing and 0.85 V (maximum recommended swing). Figure 5-10 shows an alternate representation of termination requirement. The area below the curve in Figure 5-10 shows the speed and channel reach

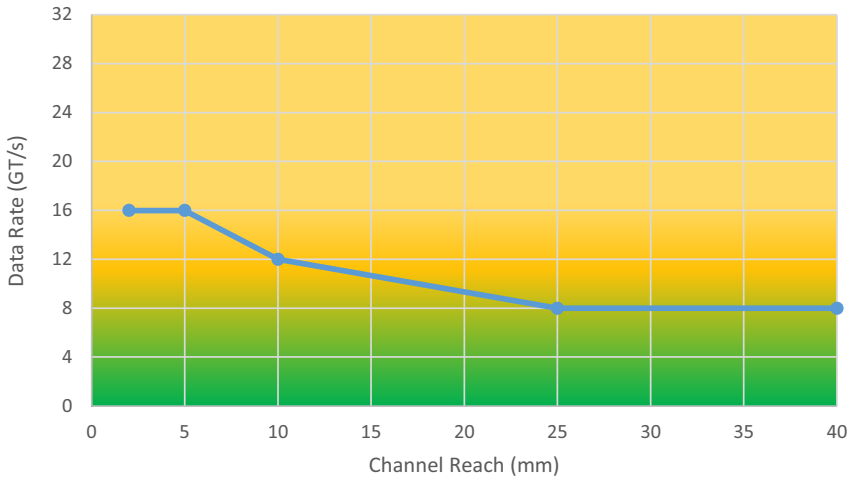
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

combinations for which the Receivers in Standard Package Modules are recommended to remain unterminated.

Table 5-6. Maximum Channel reach for unterminated Receiver (TX swing = 0.85V)

Data Rate (GT/s)	Channel Reach (mm)
16	5
12	10
8 and below	All supported Lengths

Figure 5-10. Receiver termination map for Table 5-6 (TX Swing = 0.85 V)



5.4.3 24 and 32 GT/s Receiver Equalization

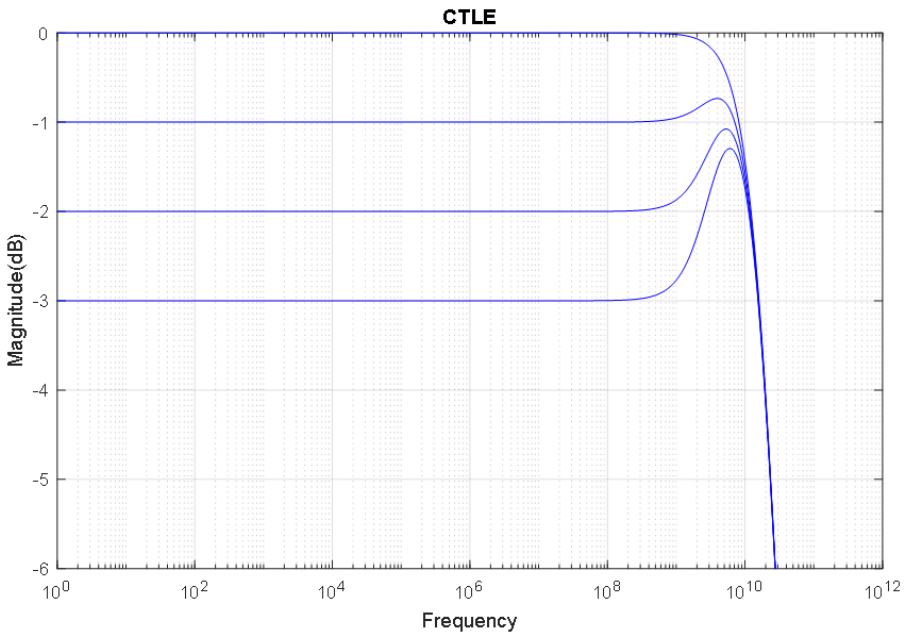
Receiver equalization may be implemented at 24 GT/s and 32 GT/s data rates. This enables Link operation even when TX equalization is not available. Implementation can be CTLE, inductive peaking, 1-tap DFE, or others. Expected RX equalization capability is equivalent of 1st order CTLE. Example transfer function curves of a first order CTLE are shown in Figure 5-11 and the corresponding equation is shown below:

$$H(s) = \omega_{p2}\left(\frac{s + A_{DC}\omega_{p1}}{(s+\omega_{p1})(s+\omega_{p2})}\right)$$

where, ω_{p2} = 2π*DataRate, ω_{p1} = 2π*DataRate /4, and A_{DC} is the DC gain.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-11. Example CTLE



5.5 Clocking

Figure 5-12 shows the forwarded clocking architecture. Each module supports a two-phase forwarded clock. It is critical to maintain matching between all data Lanes and valid signal within the module. The Receiver must provide matched delays between the Receiver clock distribution and Data/Valid Receiver path. This is to minimize the impact of power supply noise-induced jitter on Link performance. Phase adjustment is performed on the Transmitter as shown in Figure 5-12. Link training is required to set the position of phase adjustment to maximize the Link margin.

At higher data rates, Receiver eye margins may be small and any skew between the data Lanes (including Valid) may further degrade Link performance. Per-Lane deskew must be supported on the Transmitter at high data rates.

This specification supports quarter-rate clock frequencies at data rates (24 GT/s and 32 GT/s). The forwarded clock Transmitter must support quadrature phases in addition to differential clock at these data rates (to enable either quarter-rate or half-rate Receiver implementations). Table 5-7 shows the clock frequencies and phases that must be supported at different data rates. Forwarded Clock Phase is negotiated during Link Initialization and Training (see Section 4.5.3.3.1).

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.5.1 Track

Track signal can be used to perform runtime recalibration to adjust the Receiver clock path against slow varying voltage, temperature and transistor aging conditions.

When requested by the UCIe Module, the UCIe Module Partner sends a clock pattern (1010…) aligned with Phase-1 of the forwarded clock on its Track Transmitter, as shown in Figure 5-12.

Figure 5-12. Clocking architecture

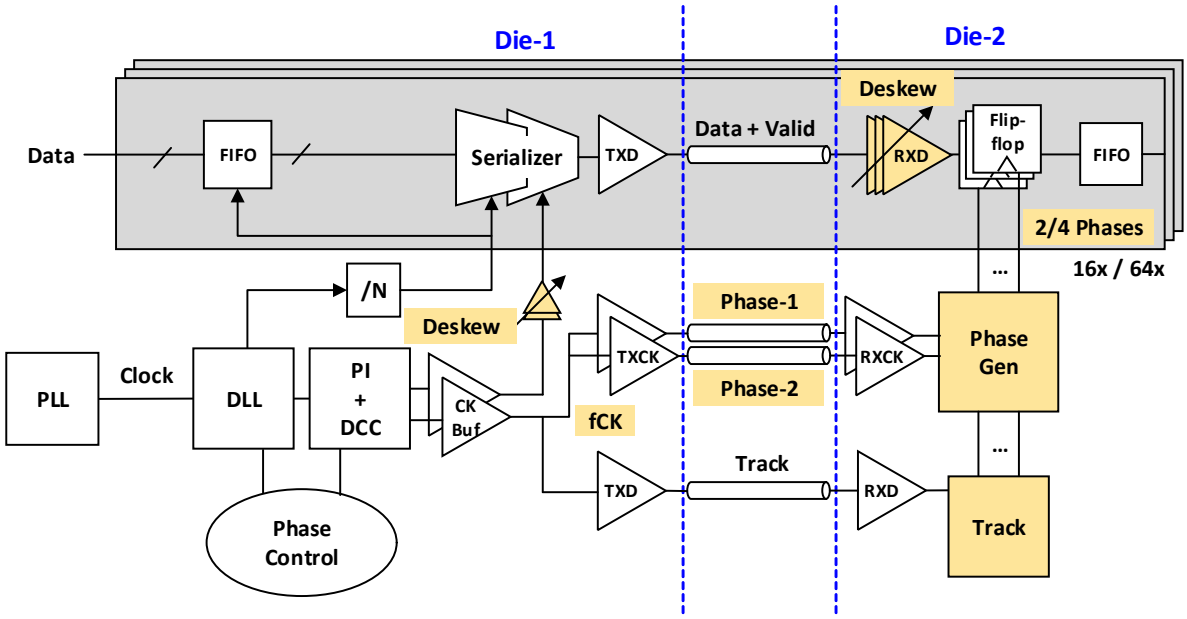


Table 5-7. Forwarded clock frequency and phase

Data rate (GT/s)	Clock freq. (fCK) (GHz)	Phase -1	Phase-2	Deskew (Req/Opt)
32	16	90	270	Required
	8	45	135	Required
24	12	90	270	Required
	6	45	135	Required
16	8	90	270	Required
12	6	90	270	Required
8	4	90	270	Optional
4	2	90	270	Optional

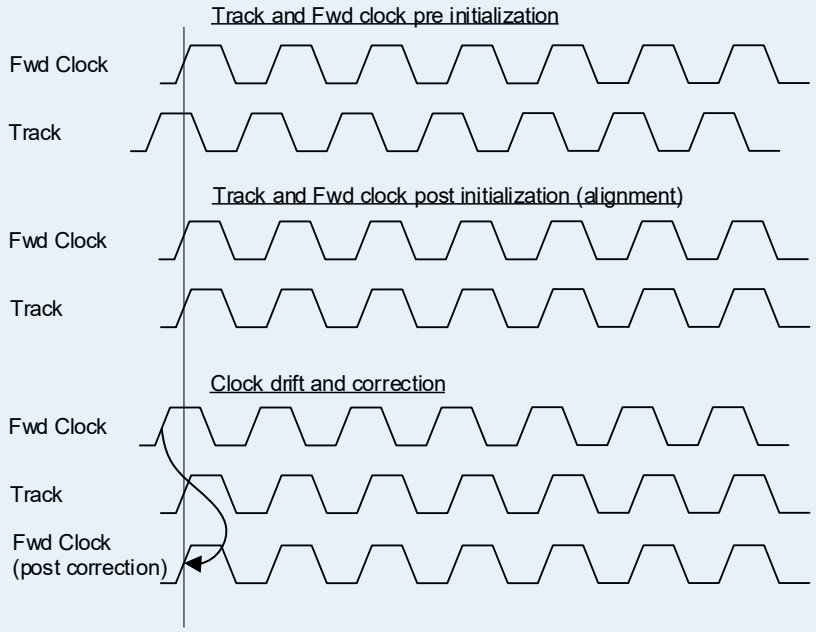
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

IMPLEMENTATION NOTE

This implementation note provides an example usage for Track signal to calibrate out slow varying temperature- and voltage-related delay drift between Data and Clock on the Receiver.

Track uses the same type of Tx driver and Rx receiver as Data (see Figure 5-12). A clock pattern aligned with Phase-1 of the forwarded clock is sent from Track Transmitter and received on the Track Receiver. Any initial skew can be calibrated out during initialization and training (MBTRAIN.RXCLKCAL) on the Receiver side. During run-time, any drift between Data and the forwarded clock can be detected. One method for detecting the drift is to sample Track with the forwarded clock. An implementation-specific number of samples can be collected, averaged if needed, and used for drift detection. This drift can then be corrected on the forwarded clock (if needed).

Figure 5-13. Track Usage Example



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.6 Supply noise and clock skew

I/O Vcc noise and the clock skew between data modules shall be within the range specified in Table 5-8.

Table 5-8. I/O Noise and Clock Skew

Parameter	Min	Nom	Max	Unit
I/O Vcc noise for 4 GT/s and 8 GT/s[1]	-	-	80	mVpp
I/O Vcc noise for 12 GT/s and 16 GT/s[1]	-	-	40-50	mVpp
I/O Vcc noise for 24 GT/s and 32 GT/s[1]	-	-	30	mVpp
Module to module clock skew[2]	-	-	60	ps

1. I/O VCC noise includes bandwidth above 20 MHz.
2. Applies only to multi-module instantiations.

IMPLEMENTATION NOTE

Due to different micro bump max current capacity and power delivery requirements, PHY in Advanced Package may have TX providing I/O power supply to RX circuits.

Due to low current draw, sideband supply voltage is strongly recommended to be on an always-on power domain.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.7 Ball-out and Channel Specification

UCIe interconnect channel needs to meet the requirement of minimum rectangular eye open as specified in Table 5-9 under channel compliance simulation conditions with noiseless and jitter-less behavioral TX and RX models.

Figure 5-14. Example Eye diagram

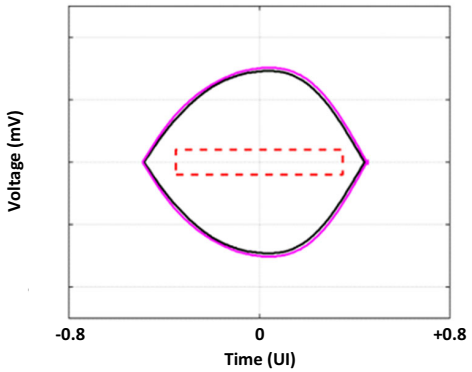


Table 5-9. Eye requirements

Data Rate (GT/s)	Eye Height (mV)	Eye width (UI)
4, 8, 12, 16[1] [3]	40	0.75
24, 32[1] [2] [3]	40	0.65

1. Rectangular mask.
2. With equalization enabled.
3. Based on minimum Tx swing specification.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

IMPLEMENTATION NOTE

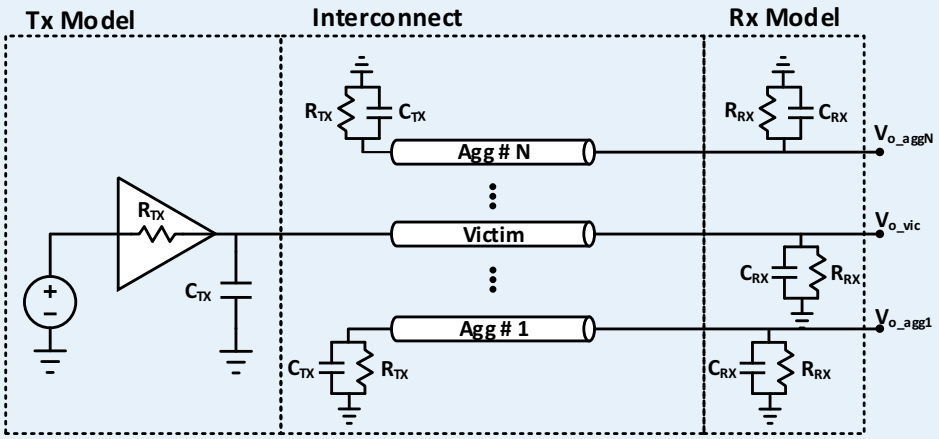
Figure 5-15 shows an example circuit setup that can be used to generate the statistical eye diagram shown in Figure 5-14. RTX is the Transmitter impedance and RRX represents the Receiver termination. CTX, CRX represent effective Transmitter and Receiver capacitance, respectively. For crosstalk, the 19-largest aggressors need to be included. Transmitter equalization (TXEQ) is enabled at 24 GT/s and 32 GT/s.

The eye diagram was generated using a two-step process.

1. Generate ISI and XTALK channel step response using circuit setup shown in Figure 5-15.
2. Use the generated channel response in a signal-integrity or channel-simulation tool to generate a statistical eye diagram (see Figure 5-14)

Other equivalent methods may be used, depending on the signal-integrity or channel-simulation tool.

Figure 5-15. Example Eye Simulation Setup

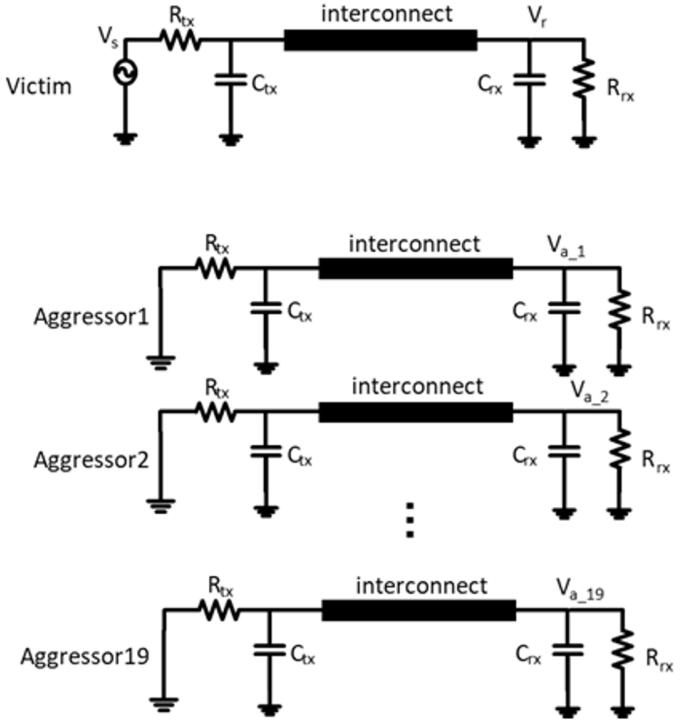


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.7.1 Voltage Transfer Function

Voltage Transfer Function (VTF) based metrics are used to define insertion loss and crosstalk. VTF metrics incorporate both resistive and capacitive components of TX and RX terminations. Figure 5-16 shows the circuit diagram for VTF calculations.

Figure 5-16. Circuit for VTF calculation



VTF loss is defined as the ratio of the Receiver voltage and the Source voltage, as shown in the equations below:

$$L(f) = 20\log 10\left|\frac{V_r(f)}{V_s(f)}\right|$$

$$L(0) = 20\log 10\left(\frac{R_{rx}}{R_{tx}+R_{channel}+R_{rx}}\right)$$

L(f) is the frequency dependent loss and L(0) is the DC loss. For unterminated channel, L(0) is effectively 0.

VTF crosstalk is defined as the power sum of the ratios of the aggressor Receiver voltage to the source voltage. 19 aggressors are included in the calculation. Based on crosstalk reciprocity, VTF crosstalk can be expressed as:

$$XT(f) = 10\log 10\left(\sum_{i=1}^{19}\left|\frac{V_{ai}(f)}{V_s(f)}\right|^2\right)$$

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.7.2 Advanced Package

Table 5-10. Channel Characteristics

Data Rate	4-16 GT/s	24, 32 GT/s
VTF Loss (dB)	$L(f_N) > -3$	$L(f_N) > -5$
VTF Crosstalk (dB)[1]	$XT(f_N) < 1.5\ L(f_N) - 21.5$ and $XT(f_N) < -23$	$XT(f_N) < 1.5\ L(f_N) - 19$ and $XT(f_N) < -24$

1. Based on Voltage Transfer Function Method (Tx: 25 ohm / 0.25 pF; Rx: 0.2 pF).

f_N is the Nyquist frequency. The equations in the table form a segmented line in 2-D map of loss and crosstalk, defining the pass/fail region.

Table 5-11. x64 Advanced Package Module Signal List (Sheet 1 of 2)[1]

Signal Name	Count	Description
Data		
TXDATA[63:0]	64	Transmit Data
TXVLD	1	Transmit Data Valid; Enables clocking in corresponding module
TXTRK	1	Transmit Track signal
TXCKP	1	Transmit Clock Phase-1
TXCKN	1	Transmit Clock Phase-2
TXCKRD	1	Redundant for Clock and Track Lane repair
TXDATARD[3:0]	4	Redundant for Data Lane repair
TXVLDRD	1	Redundant for Valid
RXDATA[63:0]	64	Receive Data
RXVLD	1	Receive Data Valid; Enables clocking in corresponding module
RXTRK	1	Receive Track.
RXCKP	1	Receive Clock Phase-1
RXCKN	1	Receive Clock Phase-2
RXDATARD[3:0]	4	Redundant for Data Lane repair
RXCKRD	1	Redundant for Clock Lane repair
RXVLDRD	1	Redundant for Valid
Sideband		
TXDATASB	1	Sideband Transmit Data
RXDATASB	1	Sideband Receiver Data
TXCKSB	1	Sideband Transmit Clock
RXCKSB	1	Sideband Receive Clock
TXDATASBRD	1	Redundant Sideband Transmit Data
RXDATASBRD	1	Redundant Sideband Receiver Data
TXCKSBRD	1	Redundant Sideband Transmit Clock
RXCKSBRD	1	Redundant Sideband Receive Clock

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-11. x64 Advanced Package Module Signal List (Sheet 2 of 2)[1]

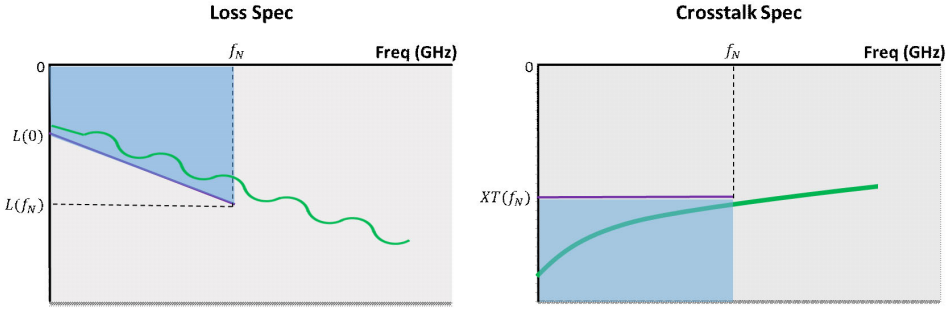
Signal Name	Count	Description
Power and Voltage		
VSS		Ground Reference
VCCIO		I/O supply
VCCFWDIO		Forwarded power supply from remote Transmitter supply to local Receiver AFE (see Tightly Coupled mode in Section 5.8)
VCCAON		Always on Aux supply (sideband)

1. For x32 Advanced Package module, the **TXDATA[63:32]**, **TXRD[3:2]**, **RXDATA[63:32]**, and **RXRD[3:2]** signals do not apply. All other signals are the same as the x64 Advanced Package Module signals.

5.7.2.1 Loss and Crosstalk Mask

Loss and crosstalk are specified by a mask defined by the $L(f_N)$ and $XT(f_N)$ at Nyquist frequency. It is a linear mask from DC to f_N for loss and flat mask for crosstalk, illustrated by Figure 5-17. Loss from DC to f_N needs to be above the spec line. Crosstalk from DC to f_N needs to be below the spec line.

Figure 5-17. Loss and Crosstalk Mask



5.7.2.2 x64 Advanced Package Module Bump Map

All bump matrices in this section and hereinafter are defined with "dead bug" view which means the viewer is looking directly at the UCIe micro bumps facing up, with the die flipped like a "dead bug" as illustrated in Figure 5-18.

Figure 5-18. Viewer Orientation Looking at the Defined UCIe Bump Matrix

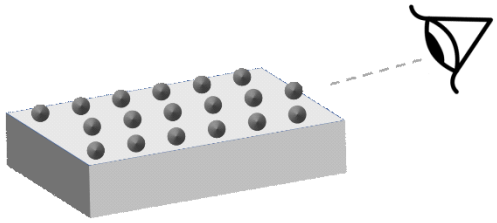


Figure 5-19, Figure 5-20, and Figure 5-21 show the reference bump matrix for the 10-column, 16-column, and 8-column x64 Advanced Package Modules, respectively. The lower left corner of the bump map will be considered "origin" of a bump matrix and the leftmost column is Column 0.

It is strongly recommended to follow the bump matrices provided in Figure 5-19, Figure 5-20, and Figure 5-21 for x64 Advanced Package interfaces.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

The 10-column bump matrix is optimal for bump pitch range of 38 to 50 um. To achieve optimal area scaling with different bump pitches, the optional 16-column and 8-column bump matrices are defined for bump ranges of 25 to 37 um and 51 to 55 um, respectively, which will result in optimal Module depth while maintaining Module width of 388.8 um, as shown in Figure 5-20 and Figure 5-21, respectively.

The following rule must be followed for the 10-column x64 Advanced Package bump matrix:

- The signal order within a column must be preserved. For example, Column 0 must contain the signals: **txdataRD0**, **txdata0**, **txdata1**, **txdata2**, **txdata3**, **txdata4**, …, **rxdata59**, **rxdata60**, **rxdata61**, **rxdata62**, **rxdata63**, **rxdataRD3**, and **txdatasbRD**. Similarly, 16-column and 8-column x64 Advanced Packages must preserve the signal order within a column of the respective bump matrices.

It is strongly recommended to follow the supply and **ground** pattern shown in the bump matrices. It must be ensured that sufficient supply and **ground** bumps are provided to meet channel characteristics (FEXT and NEXT) and power-delivery requirements.

The following rules must be followed when instantiating multiple modules of Advanced Package bump matrix:

- Modules must be stepped in the same orientation and abutted.
- Horizontal or vertical mirroring is not allowed.
- Module stacking is not allowed.

Additionally, in multi-module instantiations it is strongly recommended to add one column of **VSS** bumps on each outside edge of the multi-module instantiation.

Mirror die implementation may necessitate a jog or additional metal layers for proper connectivity.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

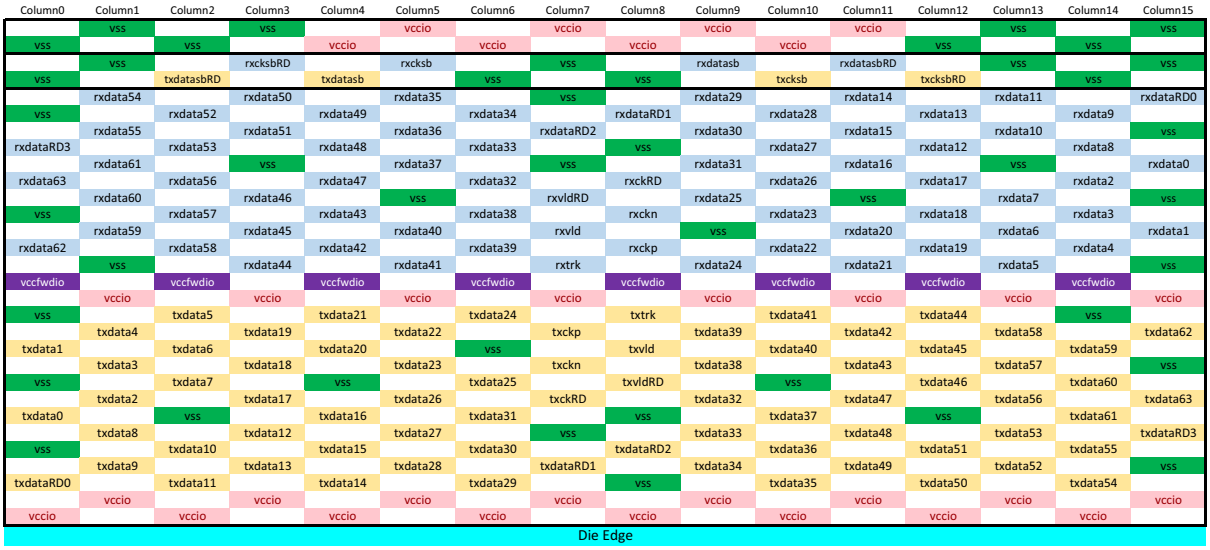
Figure 5-19. 10-column x64 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
vss		vss		vccio		vccio		vss	
	vss		vccio		vccio		vss		vss
vss		vss		vccio		vccio		vss	
	rxcksbRD		rxcksb		vccio		rxdatasb		rxdatasbRD
txdatasbRD		txdatasb		vccio		txcksb		txcksbRD	
	rxdata50		rxdata35		rxdata29		rxdata14		rxdataRD0
rxdataRD3		rxdata49		rxdata34		rxdata28		rxdata13	
	rxdata51		rxdata36		rxdata30		rxdata15		vss
rxdata63		vccio		rxdata33		vccio		rxdata12	
	rxdata52		vss		rxdata31		vss		rxdata0
vss		rxdata48		rxdata32		rxdata27		rxdata11	
	rxdata53		rxdata37		rxdataRD1		rxdata16		rxdata1
rxdata62		rxdata47		rxdataRD2		rxdata26		rxdata10	
	rxdata54		rxdata38		vss		rxdata17		vss
rxdata61		rxdata46		vccio		rxdata25		rxdata9	
	rxdata55		rxdata39		rxckRD		rxdata18		rxdata2
vss		rxdata45		rxvldRD		rxdata24		rxdata8	
	rxdata56		vss		rxckn		rxdata19		rxdata3
rxdata60		rxdata44		rxvld		vss		rxdata7	
	rxdata57		rxdata40		rxckp		rxdata20		vss
rxdata59		rxdata43		rxtrk		rxdata23		rxdata6	
	rxdata58		rxdata41		vss		rxdata21		rxdata4
vss		rxdata42		vccio		rxdata22		rxdata5	
	vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio
vccio		txdata21		vccio		txdata41		txdata58	
	txdata5		txdata22		vss		txdata42		vss
txdata4		txdata20		txckp		txdata40		txdata57	
	txdata6		txdata23		txtrk		txdata43		txdata59
vss		txdata19		txckn		vss		txdata56	
	txdata7		vss		txvld		txdata44		txdata60
txdata3		txdata18		txckRD		txdata39		txdata55	
	txdata8		txdata24		txvldRD		txdata45		vss
txdata2		txdata17		vccio		txdata38		txdata54	
	txdata9		txdata25		vss		txdata46		txdata61
vccio		vccio		vccio		vccio		vccio	
	txdata10		txdata26		txdataRD2		txdata47		txdata62
txdata1		txdata16		txdataRD1		txdata37		txdata53	
	txdata11		txdata27		txdata32		txdata48		vss
txdata0		vss		txdata31		vss		txdata52	
	txdata12		vss		txdata33		vss		txdata63
vss		txdata15		txdata30		txdata36		txdata51	
	txdata13		txdata28		txdata34		txdata49		txdataRD3
txdataRD0		txdata14		txdata29		txdata35		txdata50	
	vccio		vccio		vccio		vccio		vccio
vccio		vccio		vccio		vccio		vccio	
Die Edge									

Note: In Figure 5-19, at 45-um pitch, the module depth of the 10-column reference bump matrix as shown is approximately 1043 um.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

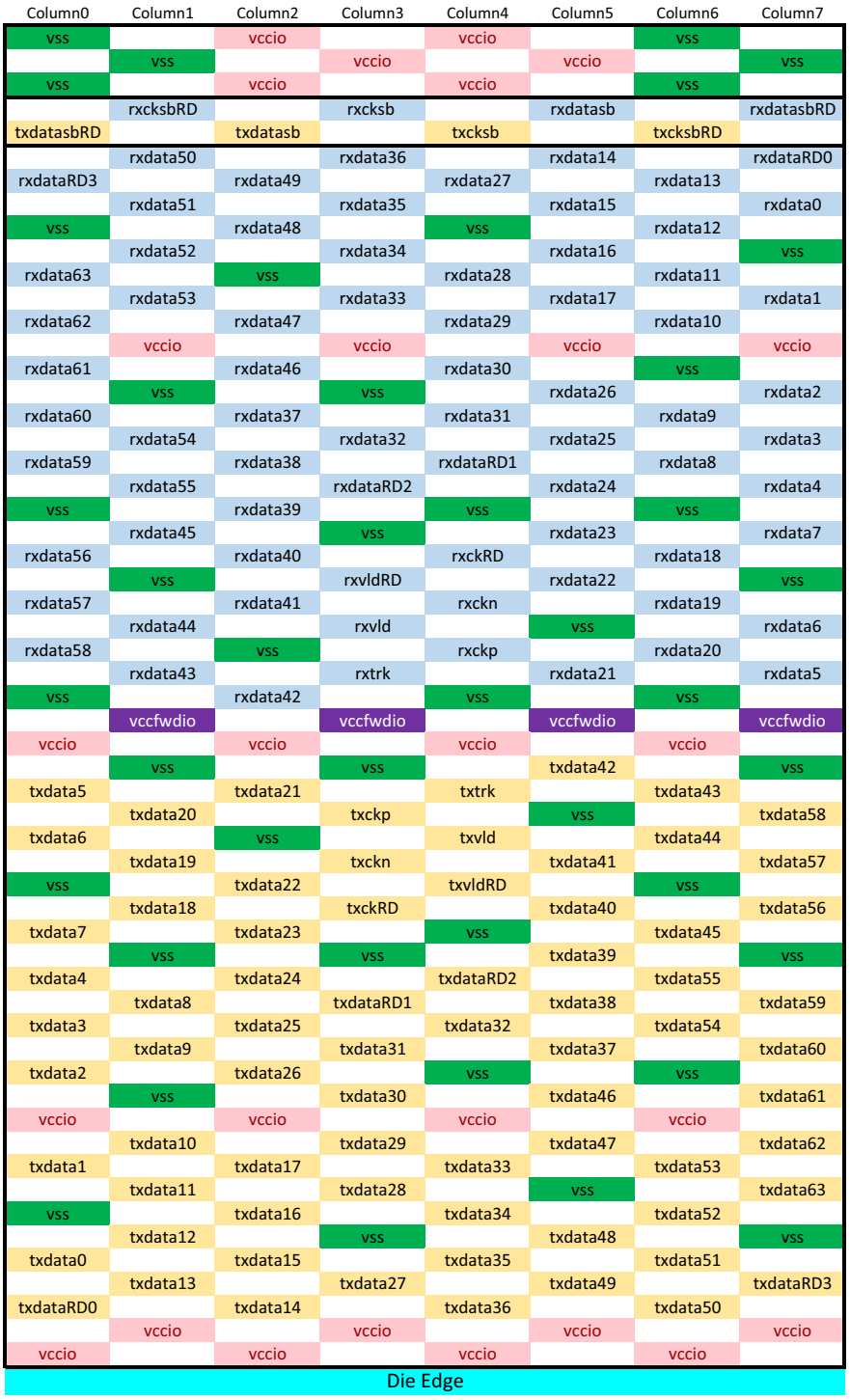
Figure 5-20. 16-column x64 Advanced Package Bump Map



Note: In Figure 5-20, at 25-um pitch, the module depth of the 16-column reference bump matrix as shown is approximately 388 um.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-21. 8-column x64 Advanced Package Bump Map



Note: In Figure 5-21, at 55-um pitch, the module depth of the 8-column reference bump matrix as shown is approximately 1,585 um.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-22 shows the signal exit order for the 10-column x64 Advanced Package bump map.

Figure 5-22. 10-column x64 Advanced Package Bump map: Signal exit order

Tx Breakout	Left to Right																
		txdataRD0	txdata0	txdata1	txdata2	txdata3	txdata4	txdata5	txdata6	txdata7	txdata8	txdata9	txdata10	txdata11	txdata12	txdata13	Cont…
	Cont…	txdata14	txdata15	txdata16	txdata17	txdata18	txdata19	txdata20	txdata21	txdata22	txdata23	txdata24	txdata25	txdata26	txdata27	txdata28	Cont1…
	Cont1…	txdata29	txdata30	txdata31	txdataRD1	txckRD	txckn	txckp	txtrk	txvld	txvldRD	txdataRD2	txdata32	txdata33	txdata34	txdata35	Cont2…
	Cont2…	txdata36	txdata37	txdata38	txdata39	txdata40	txdata41	txdata42	txdata43	txdata44	txdata45	txdata46	txdata47	txdata48	txdata49	txdata50	Cont3…
	Cont3…	txdata51	txdata52	txdata53	txdata54	txdata55	txdata56	txdata57	txdata58	txdata59	txdata60	txdata61	txdata62	txdata63	txdataRD3		
Rx Breakout	Left to Right																
		rxdataRD3	rxdata63	rxdata62	rxdata61	rxdata60	rxdata59	rxdata58	rxdata57	rxdata56	rxdata55	rxdata54	rxdata53	rxdata52	rxdata51	rxdata50	Cont…
	Cont…	rxdata49	rxdata48	rxdata47	rxdata46	rxdata45	rxdata44	rxdata43	rxdata42	rxdata41	rxdata40	rxdata39	rxdata38	rxdata37	rxdata36	rxdata35	Cont1…
	Cont1…	rxdata34	rxdata33	rxdata32	rxdataRD2	rxvldRD	rxvld	rxtrk	rxckp	rxckn	rxckRD	rxdataRD1	rxdata31	rxdata30	rxdata29	rxdata28	Cont2…
	Cont2…	rxdata27	rxdata26	rxdata25	rxdata24	rxdata23	rxdata22	rxdata21	rxdata20	rxdata19	rxdata18	rxdata17	rxdata16	rxdata15	rxdata14	rxdata13	Cont3…
	Cont3…	rxdata12	rxdata11	rxdata10	rxdata9	rxdata8	rxdata7	rxdata6	rxdata5	rxdata4	rxdata3	rxdata2	rxdata1	rxdata0	rxdataRD0		

5.7.2.3 x32 Advanced Package Module Bump Map

UCIe also defines a x32 Advanced Package Module that supports 32 Tx and 32 Rx data signals and two redundant bumps each for Tx and two for Rx (total of four) for lane-repair functions. All other signals, including the sidebands, are the same as those of the x64 Advanced Package.

Figure 5-23, Figure 5-24, and Figure 5-25 show the reference bump matrix for the 10-column, 16-column, and 8-column x32 Advanced Package Modules, respectively. The lower left corner of the bump map will be considered "origin" of a bump matrix and the leftmost column is Column 0.

It is strongly recommended to follow the bump matrices provided in Figure 5-23, Figure 5-24, and Figure 5-25 for x32 Advanced Package Modules.

The following rule must be followed for the 10-column x32 Advanced Package bump matrix:

- The signals order within a column must be preserved. For example, Column 0 must contain the signals: **txdataRD0**, **txdata0**, **txdata1**, **txdata2**, **txdata3**, **txdata4**, and **txdatasbRD**. Similarly, 16-column and 8-column x32 Advanced Packages must preserve the signal order within a column of the respective bump matrices.

It is strongly recommended to follow the supply and **ground** pattern shown in the bump matrices. It must be ensured that sufficient supply and **ground** bumps are provided to meet channel characteristics (FEXT and NEXT) and power-delivery requirements.

When instantiating multiple x32 Advanced Package Modules, the same rules as defined in Section 5.7.2.2 must be followed.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-23. 10-column x32 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
vss		vss		vccio		vccio		vss	
	vss		vccio		vccio		vss		vss
vss		vss		vccio		vccio		vss	
	rxcksbRD		rxcksb		vccio		rxdatasb		rxdatasbRD
txdatasbRD		txdatasb		vss		txcksb		txcksbRD	
	vss		txdata22		rxdata31		vccio		vccio
vss		txdata21		txckp		rxdata30		rxdata13	
	txdata5		txdata23		vss		rxdata14		vccio
vccio		txdata20		txckn		rxdata29		rxdata12	
	txdata6		vss		rxdataRD1		rxdata15		rxdataRD0
txdata4		vss		txckRD		rxdata28		rxdata11	
	txdata7		txdata24		rxvldRD		vss		vss
vss		txdata19		txtrk		rxdata27		rxdata10	
	txdata8		txdata25		rxvld		rxdata16		rxdata0
txdata3		txdata18		vss		rxdata26		vss	
	vss		txdata26		vss		rxdata17		rxdata1
txdata2		txdata17		txvld		rxdata25		rxdata9	
	txdata9		vss		rxtrk		rxdata18		vss
vccio		vccio		vccio		vccfwdio		vccfwdio	
	txdata10		txdata27		rxckRD		rxdata19		rxdata2
txdata1		txdata16		txvldRD		rxdata24		rxdata8	
	txdata11		txdata28		rxckn		rxdata20		rxdata3
txdata0		vss		vss		vss		rxdata7	
	txdata12		txdata29		rxckp		vss		vss
vss		txdata15		txdataRD1		rxdata23		rxdata6	
	txdata13		txdata30		vss		rxdata21		rxdata4
txdataRD0		txdata14		txdata31		rxdata22		rxdata5	
	vccio		vccio		vccfwdio		vccfwdio		vccfwdio
vccio		vccio		vccio		vccfwdio		vccfwdio	
Die Edge									

Note: In Figure 5-23, at 45-um pitch, the module depth of the 10-column reference bump matrix as shown is approximately 680.5 um.

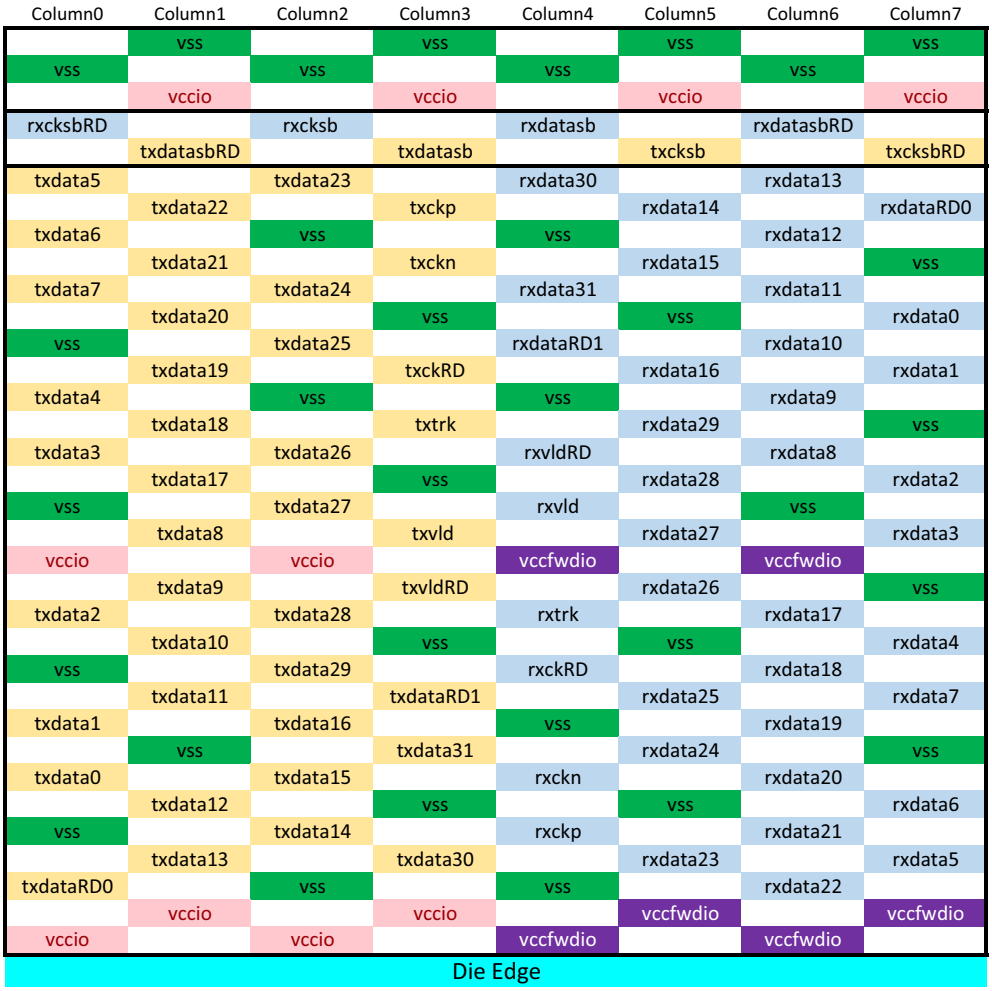
Figure 5-24. 16-column x32 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11	Column12	Column13	Column14	Column15
	vss		vss		vss		vss		vss		vss		vss		vss
vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio	
	txdatasbRD		rxcksbRD		txdatasb		rxcksb		txcksb		rxdatasb		txcksbRD		rxdatasbRD
vss		txdata5		vss		vss		vss		vss		vss		vss	
	txdata4		txdata20		txdata22		txtrk		rxdata30		rxdata26		rxdata13		rxdataRD0
txdata2		vss		txdata21		txckp		rxdata31		rxdata27		rxdata14		rxdata11	
	txdata3		txdata17		txdata23		vss		rxdata29		vss		rxdata12		vss
txdata1		txdata6		txdata19		txckn		vss		rxdata28		rxdata15		rxdata10	
	txdata8		vss		vss		txvld		rxdataRD1		rxdata25		vss		rxdata0
vss		txdata7		txdata18		txckRD		rxvldRD		rxdata24		rxdata16		rxdata9	
	txdata9		txdata16		txdata24		txvldRD		rxckRD		rxdata18		rxdata7		vss
txdata0		vss		txdata25		txdataRD1		rxvld		vss		rxdata17		rxdata8	
	txdata10		txdata15		txdata28		vss		rxckn		rxdata19		rxdata6		rxdata1
vss		txdata12		vss		txdata29		vss		rxdata23		vss		rxdata3	
	txdata11		txdata14		txdata27		txdata31		rxckp		rxdata21		rxdata5		rxdata2
txdataRD0		txdata13		txdata26		txdata30		rxtrk		rxdata22		rxdata20		rxdata4	
	vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio
vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	
Die Edge															

Note: In Figure 5-24, at 25-um pitch, the module depth of the 16-column reference bump matrix as shown is approximately 237.5 um.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-25. 8-column x32 Advanced Package Bump Map



Note: In Figure 5-25, at 55-um pitch, the module depth of the 8-column reference bump matrix as shown is approximately 962.5 um.

Figure 5-26 shows the signal exit order for the 10-column x32 Advanced Package bump map.

Figure 5-26. 10-column x32 Advanced Package Bump Map: Signal Exit Order

Tx Breakout	**Left to Right**											
		txdataRD0	txdata0	txdata1	txdata2	txdata3	txdata4	txdata5	txdata6	txdata7	txdata8	Cont…
	Cont…	txdata9	txdata10	txdata11	txdata12	txdata13	txdata14	txdata15	txdata16	txdata17	txdata18	Cont1…
	Cont1…	txdata19	txdata20	txdata21	txdata22	txdata23	txdata24	txdata25	txdata26	txdata27	txdata28	Cont2…
	Cont2…	txdata29	txdata30	txdata31	txdataRD1	txvldRD	txvld	txtrk	txckRD	txckn	txckp	
Rx Breakout	**Left to Right**											
		rxckp	rxckn	rxckRD	rxtrk	rxvld	rxvldRD	rxdataRD1	rxdata31	rxdata30	rxdata29	Cont…
	Cont…	rxdata28	rxdata27	rxdata26	rxdata25	rxdata24	rxdata23	rxdata22	rxdata21	rxdata20	rxdata19	Cont1…
	Cont1…	rxdata18	rxdata17	rxdata16	rxdata15	rxdata14	rxdata13	rxdata12	rxdata11	rxdata10	rxdata9	Cont2…
	Cont2…	rxdata8	rxdata7	rxdata6	rxdata5	rxdata4	rxdata3	rxdata2	rxdata1	rxdata0	rxdataRD0	

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.7.2.4 x64 and x32 Advanced Package Module Interoperability

x64 and x32 Advanced Package Module bump maps enable interoperability between all Tx and Rx combinations of x64 or x32, 10-column, 16-column, or 8-column Modules, in both Normal-to-Normal module orientation or Normal-to-Mirrored module orientation.

However, if x64 to x32 modules or x32 to x32 modules have normal and mirrored orientation as shown in Figure 5-27 and Figure 5-28, respectively, signal traces between the TX half and RX half will crisscross and require swizzling technique which refers to rearranging the physical connections between signal bumps of two chiplets to optimize the layout and routing on the interposer or substrate. It involves changing the order of the connections or route on different layers without altering the netlist or the electrical functionality of the design. Moreover, connections between 8-column, 16-column, and 10-column modules may need to be routed to adjacent columns (swizzle and go across). In all cases, the electrical spec must be met for all these connections.

It is optional for a x64 Module to support interoperability with a x32 Module. The following requirements apply when a x64 module supports x32 interoperability:

- When a x64 module connects to x32 module, the connection shall always be contained to the lower half of the x64 module. This must be followed even with x32 lane reversal described below.
- Electrical specifications must be met for combinations that require signal-routing swizzling.
- Lane reversal will not be allowed on **CKP-**, **CKN-**, **CKRD-**, **VLD-**, **VLDRD-**, **TRK-**, and sideband-related pins. These pins need to be connected appropriately. Swizzling for these connections is acceptable.
- x64 module must support a lane-reversal mode in a x32 manner (i.e., **TD_P[31:0]** = **TD_L[0:31]**. When a x64 module is connected to a x32 module, in either Normal or Mirrored orientation, the upper 32 bits are not used and should be disabled.
- It is not permitted for a single module of larger width to simultaneously interop with two or more modules of a lower width. For example, a x64 Advanced Package module physically connected to two x32 Advanced Package modules is prohibited.

Additional technological capabilities or layers may be needed to accomplish swizzling on data/auxiliary signals.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-12 summarizes the connections between combinations of x64 and x32 modules in both Normal-to-Normal and Normal-to-Mirrored module orientations. The table applies to all combinations of 10-column, 16-column, or 8-column modules on either side of the Link.

Table 5-12. x64 and x32 Advanced Package Connectivity Matrix

–			Normal Module		Mirrored Module	
			Rx			
			x64	x32	x64	x32
Normal Module	Tx	x64	TX[63:0] – RX[63:0][1]	TX[31:0] – RX[31:0][2]	rTX[63:0] – RX[0:63][3 4]	rTX[31:0] – RX[0:31][3 5]
		x32	TX[31:0] – RX[31:0][2]	TX[31:0] – RX[31:0][2]	rTX[31:0] – RX[0:31][3 5]	rTX[31:0] – RX[0:31][3 5]

1. Entry "TX[63:0] – RX[63:0]" is for Normal Module connections between two x64 modules without lane reversal. This applies to x64–to–x64 combination.
2. Entry "TX[31:0] – RX[31:0]" is for Normal Module connections between lower 32-bit half without lane reversal. This applies to x64-to-x32, x32-to-x64, and x32-to-x32 combinations.
3. The prefix "r" means lane reversal is enabled on the Transmitter lanes, and:
   - "rTX[63:0]" means TD_P[63:0] = TD_L[0:63], to be connected with RD_P[0:63]
   - "rTX[31:0]" means TD_P[31:0] = TD_L[0:31], to be connected with RD_P[0:31].
4. Entry "rTX[63:0] – RX[0:63]" = Normal-to-Mirrored Module connections between two x64 modules with TX lane reversal. This applies to x64-to-x64 Normal-to-Mirrored combinations.
5. Entry "rTX[31:0] – RX[0:31]" = Normal-to-Mirrored Module connections between lower 32-bit half with TX lane reversal. This applies to x64-to-x32, x32-to-x64, and x32-to-x32 Normal-to-Mirrored combinations.

The defined bump matrices can achieve optimal skew between bump matrices of differing depths, and the worst-case trace-reach skews are expected to be within the maximum lane-to-lane skew limit for the corresponding data rates as defined in Section 5.3 and Section 5.4.

Figure 5-27 and Figure 5-28 show examples of normal and mirrored x64-to-x32 and x32-to-x32 Advanced Package Module connections, respectively.

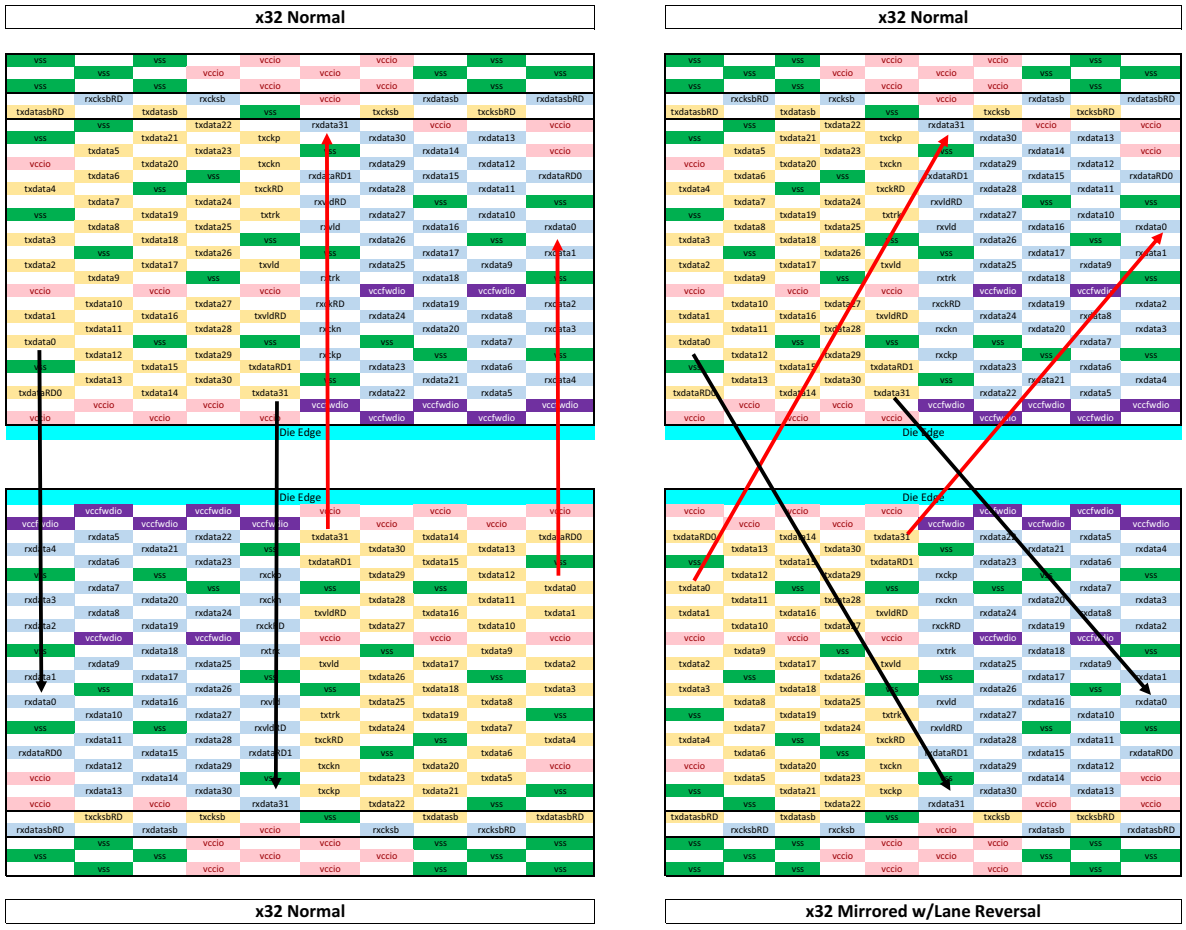
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-27. Example of Normal and Mirrored x64-to-x32 Advanced Package Module Connection



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-28. Example of Normal and Mirrored x32-to-x32 Advanced Package Module Connection



5.7.2.5 Module Naming of Advanced Package Modules

This section describes the Module naming convention of x64 and x32 Advanced Package modules in a multi-module configuration.

The Module naming is defined to help with connecting the Modules deterministically which, in turn, will help minimize the multiplexing requirements in the Multi-module PHY Logic (MMPL).

The naming of M0, M1, M2, and M3 will apply to 1, 2, or 4 Advanced Package modules that are aggregated through the MMPL.

Figure 5-29 shows the naming convention for 1, 2, or 4 Advanced Package Modules when they are connected to their "Standard Die Rotate" Module counterparts that have same number of Advanced Package Modules.

Note: The double-ended arrows in Figure 5-29 through Figure 5-32 indicate Module-to-Module connections.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-29. Naming Convention for One-, Two-, and Four-module Advanced Package Paired with "Standard Die Rotate" Configurations

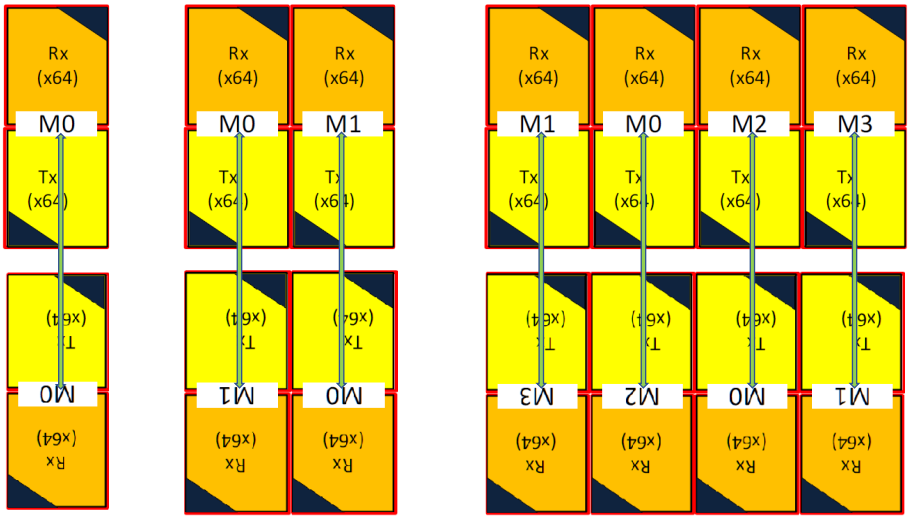
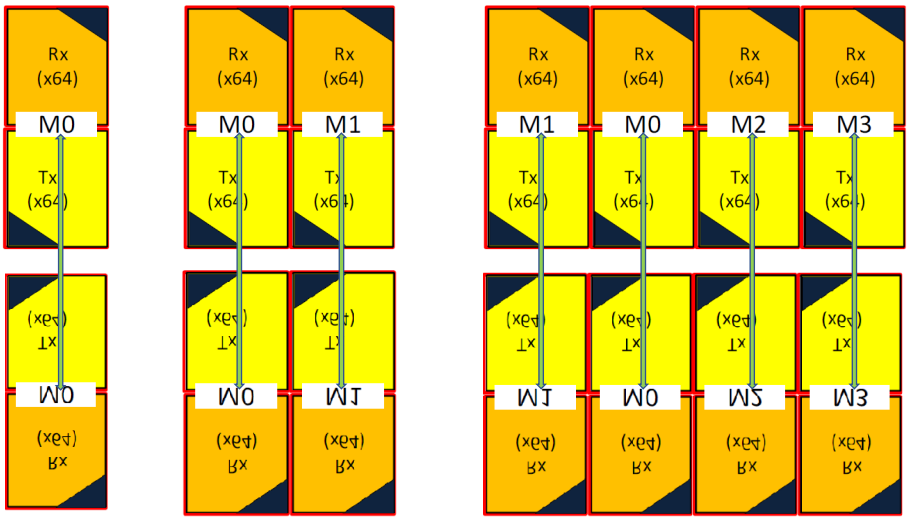


Figure 5-30 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their "Mirrored Die Rotate" counterparts with the same number of Advanced Package modules.

Figure 5-30. Naming Convention for One-, Two-, and Four-module Advanced Package Paired with "Mirrored Die Rotate" Configurations



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

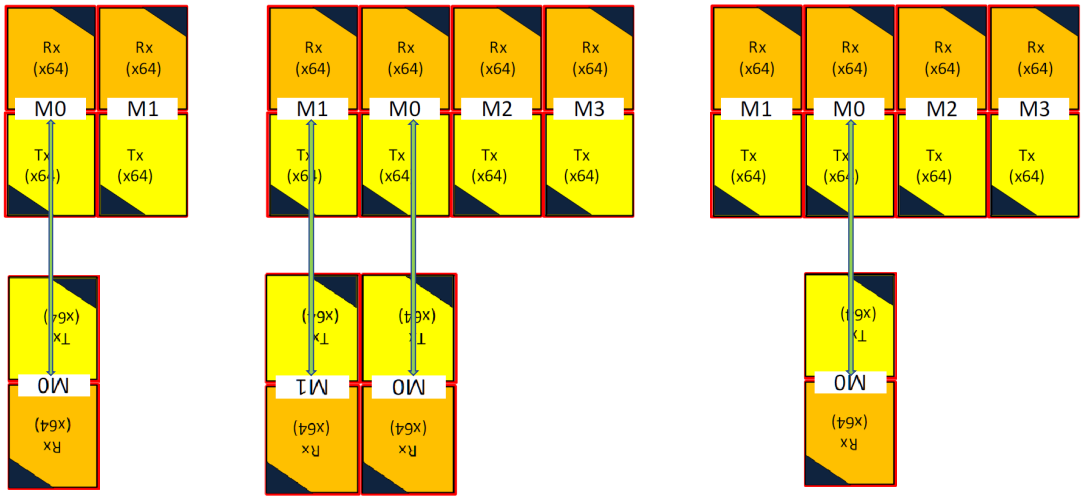
Table 5-13 summarizes the connections between the combinations shown in Figure 5-29 and Figure 5-30.

Table 5-13. Summary of Advanced Package Module Connection Combinations with Same Number of Modules on Both Sides

Advanced Package Module Connections (Same # of Modules on Both Sides)	Standard Die Rotate Counterpart	Mirrored Die Rotate Counterpart
x1 – x1	• M0 – M0	• M0 – M0
x2 – x2	• M0 – M1 • M1 – M0	• M0 – M0 • M1 – M1
x4 – x4	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3

Figure 5-31 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their "Standard Die Rotate" counterparts that have a different number of Advanced Package modules.

Figure 5-31. Examples for Advanced Package Configurations Paired with "Standard Die Rotate" Counterparts, with a Different Number of Modules



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-32 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their "Mirrored Die Rotate" counterparts that have a different number of Advanced Package modules.

Figure 5-32. Examples for Advanced Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules

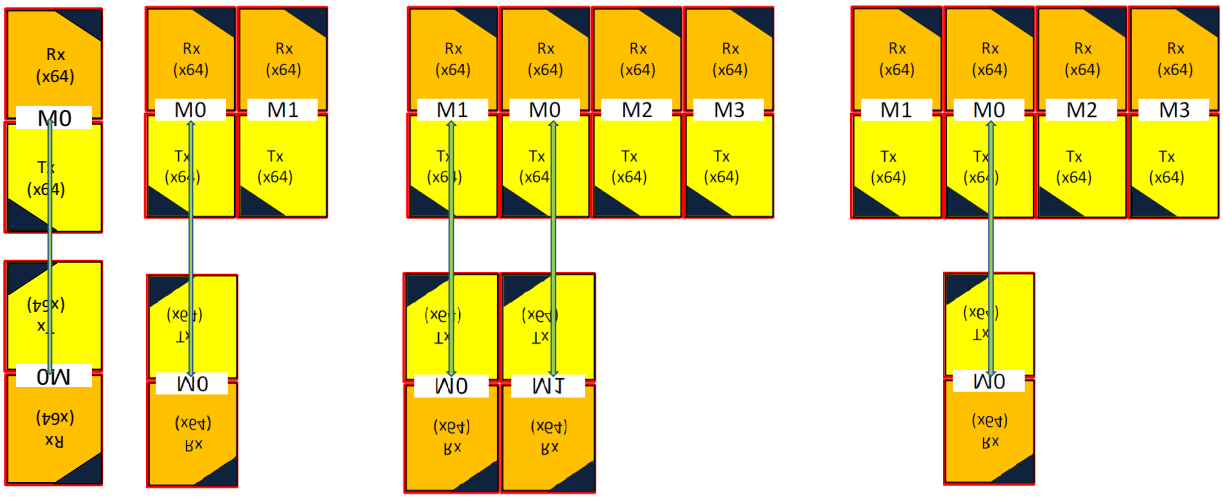


Table 5-14 summarizes the connections between the combinations shown in Figure 5-31 and Figure 5-32.

Table 5-14. Summary of Advanced Package Module Connection Combinations with Different Number of Modules on Both Sides

Advanced Package Module Connections (Different # of Modules on Both Sides)	Standard Die Rotate Counterpart[1]	Mirrored Die Rotate Counterpart[1]
x2 – x1	• M0 – M0 • M1 – NC	• M0 – M0 • M1 – NC
x4 – x2	• M0 – M0 • M1 – M1 • M3 – NC • M2 – NC	• M0 – M1 • M1 – M0 • M2 – NC • M3 – NC
x4 – x1	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	• M0 – M0 • M1 – NC • M2 – NC • M3 – NC

1. NC indicates no connection.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.7.3 Standard Package

Interconnect channel should be designed with 50 ohm characteristic impedance. Insertion loss and crosstalk for requirement at Nyquist frequency with Receiver termination is defined in Table 5-15.

Table 5-15. IL and Crosstalk for Standard Package: With Receiver Termination Enabled

Data Rate	4, 8 GT/s	12, 16 GT/s	24, 32 GT/s
VTF Loss (dB)[1 2 3]	L(0) > -4.5 $L(f_N)$ > -7.5	L(0) > -4.5 $L(f_N)$ > -6.5	L(0) > -4.5 $L(f_N)$ > -7.5
VTF Crosstalk (dB)	$XT(f_N) < 3 * L(f_N) - 11.5$ and $XT(f_N) < -25$	$XT(f_N) < 3 * L(f_N) - 11.5$ and $XT(f_N) < -25$	$XT(f_N) < 2.5 * L(f_N) - 10$ and $XT(f_N) < -26$

1. Voltage Transfer Function for 4 GT/s and 8 GT/s (Tx: 30 ohm / 0.3pF; Rx: 50 ohm / 0.3pF).
2. Voltage Transfer Function for 12 GT/s and 16 GT/s (Tx: 30 ohm / 0.2pF; Rx: 50 ohm / 0.2pF).
3. Voltage Transfer Function for 24 GT/s and 32 GT/s (Tx: 30 ohm / 0.125pF; Rx: 50 ohm / 0.125pF).

IL and crosstalk for requirement at Nyquist frequency without Receiver termination is defined by Table 5-16. Loss and crosstalk specifications between DC and Nyquist f_N follow the same methodology defined in Section 5.7.2.1.

Table 5-16. IL and Crosstalk for Standard Package: No Rx Termination

Data Rate	4-12 GT/s	16 GT/s
VTF Loss (dB)[1 2]	$L(f_N)$ > -1.25	$L(f_N)$ > -1.15
VTF Crosstalk (dB)	$XT(f_N) < 7 * L(f_N) - 12.5$ and $XT(f_N) < -15$	$XT(f_N) < 4 * L(f_N) - 13.5$ and $XT(f_N) < -17$

1. Voltage Transfer Function for 4 GT/s and 8 GT/s (Tx: 30 ohm / 0.3pF; Rx: 0.2 pF).
2. Voltage Transfer Function for 12 GT/s and 16 GT/s (Tx: 30 ohm / 0.2pF; Rx: 0.2 pF).

Table 5-17. Standard Package Module Signal List (Sheet 1 of 2)

Signal Name	Count	Description
Data		
TXDATA[15:0]	16	Transmit Data
TXVLD	1	Transmit Data Valid; Enables clocking in corresponding module
TXTRK	1	Transmit Track signal
TXCKP	1	Transmit Clock Phase-1
TXCKN	1	Transmit Clock Phase-2
RXDATA[15:0]	16	Receive Data
RXVLD	1	Receive Data Valid; Enables clocking in corresponding module
RXTRK	1	Receive Track
RXCKP	1	Receive Clock Phase-1
RXCKN	1	Receive Clock Phase-2
Sideband		
TXDATASB	1	Sideband Transmit Data
RXDATASB	1	Sideband Receiver Data
TXCKSB	1	Sideband Transmit Clock

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-17. Standard Package Module Signal List (Sheet 2 of 2)

Signal Name	Count	Description
RXCKSB	1	Sideband Receive Clock
Power and Voltage		
VSS		Ground Reference
VCCIO		I/O supply
VCCAON		Always on Aux supply (sideband)

5.7.3.1 Standard Package Module Bump Map

Figure 5-33 and Figure 5-35 show the reference bump matrices for x16 (one module) and x32 (two module) Standard Packages, respectively.

It is strongly recommended to follow the bump matrices provided in Figure 5-33 for one module and Figure 5-35 for two module Standard Packages. The lower left corner of the bump map will be considered "origin" of a bump matrix.

Signal exit order for x16 and x32 Standard Package bump matrices are shown in Figure 5-34 and Figure 5-36, respectively.

The following rules must be followed for Standard Package bump matrices:

- The signals within a column must be preserved. For example, for a x16 (one module Standard Package) shown in Figure 5-33, Column 1 must contain the signals: **txdata0**, **txdata1**, **txdata4**, **txdata5**, and **txdatasb**.
- The signals must exit the bump field in the order shown in Figure 5-34. Layer 1 and Layer 2 are two different signal routing layers in a Standard Package.

It is strongly recommended to follow the supply and **ground** pattern shown in the bump matrices. It must be ensured that sufficient supply and **ground** bumps are provided to meet channel characteristics (FEXT and NEXT) and power-delivery requirements.

The following rules must be followed for instantiating multiple modules of Standard Package bump matrix:

- When looking at a die such that the UCIe Modules are on the south side, Tx should always precede Rx within a module along the die's edge when going from left to right.
- When instantiating multiple modules, the modules must be stepped in the same orientation and abutted. Horizontal or vertical mirroring is not allowed.

If more Die Edge Bandwidth density is required, it is permitted to stack two modules before abutting. If two modules are stacked, the package may need to support at least four routing layers for UCIe signal routing. An example of stacked Standard Package Module instantiations is shown in Figure 5-35.

- If only one stacked module is instantiated, when looking at a die such that the UCIe Modules are on the south side, Tx should always precede Rx within a module along the die's edge when going from left to right.
- When instantiating multiple stacked modules, the modules must be stepped in the same orientation and abutted. Horizontal or vertical mirroring is not allowed.

Note: An example of signal routing for stacked module is shown in Figure 5-37.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-33. Standard Package Bump Map: x16 interface

Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
	txdatasb		txcksb		vccaon		vccaon		rxcksb		rxdatasb
vccio		vccio		vccio		vccio		vccio		vccio	
	vss		vss		vss		vss		vss		vss
vccio		txdata7		txdata9		vccio		rxdata8		rxdata6	
	txdata5		txckn		txdata11		rxdata10		rxckp		rxdata4
vss		vss		vss		vss		vss		vss	
	txdata4		txckp		txdata10		rxdata11		rxckn		rxdata5
vss		txdata6		txdata8		vss		rxdata9		rxdata7	
	vss		vss		vss		vss		vss		vss
vccio		txdata3		txdata13		vccio		rxdata12		rxdata2	
	txdata1		txvld		txdata15		rxdata14		rxtrk		rxdata0
vccio		vss		vss		vccio		vss		vss	
	txdata0		txtrk		txdata14		rxdata15		rxvld		rxdata1
vss		txdata2		txdata12		vss		rxdata13		rxdata3	
Die Edge											

Figure 5-34. Standard Package x16 interface: Signal exit order

Layer 1	Tx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Rx
Layer 2	Module	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module
Sideband		txdatasb					txcksb							rxcksb					rxdatasb			

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

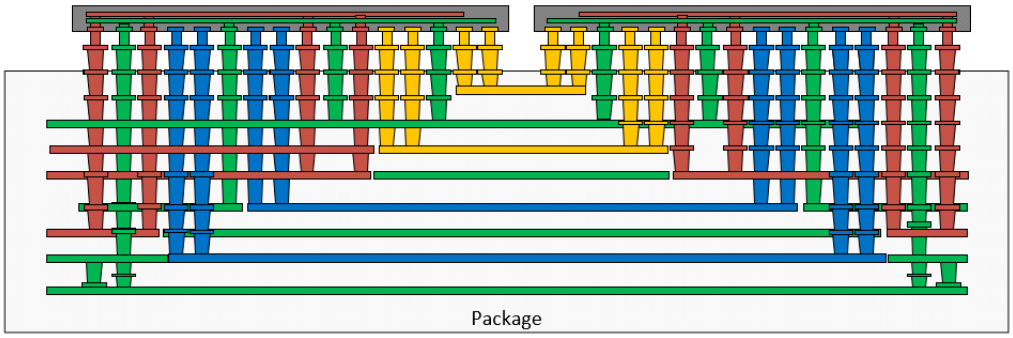
Figure 5-35. Standard Package Bump Map: x32 interface

Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
	m2rxdatasb		m2rxcksb		vccaon		m2txcksb		m2txdatasb		vccaon
m1txdatasb		m1txcksb		vccaon		vccaon		m1rxcksb		m1rxdatasb	
	vccio		vccio		vccio		vccio		vccio		vccio
vss		vss		vss		vss		vss		vss	
	m2rxdata6		m2rxdata8		vss		m2txdata9		m2txdata7		vss
m2rxdata4		m2rxckp		m2rxdata10		m2txdata11		m2txckn		m2txdata5	
	vss		vss		vss		vss		vss		vss
m2rxdata5		m2rxckn		m2rxdata11		m2txdata10		m2txckp		m2txdata4	
	m2rxdata7		m2rxdata9		vss		m2txdata8		m2txdata6		vss
vss		vss		vss		vss		vss		vss	
	m2rxdata2		m2rxdata12		vss		m2txdata13		m2txdata3		vss
m2rxdata0		m2rxtrk		m2rxdata14		m2txdata15		m2txvld		m2txdata1	
	vss		vss		vss		vss		vss		vss
m2rxdata1		m2rxvld		m2rxdata15		m2txdata14		m2txtrk		m2txdata0	
	m2rxdata3		m2rxdata13		vccio		m2txdata12		m2txdata2		vccio
vccio		vccio		vccio		vccio		vccio		vccio	
	vss		vss		vccio		vss		vss		vccio
vccio		m1txdata7		m1txdata9		vccio		m1rxdata8		m1rxdata6	
	m1txdata5		m1txckn		m1txdata11		m1rxdata10		m1rxckp		m1rxdata4
vss		vss		vss		vss		vss		vss	
	m1txdata4		m1txckp		m1txdata10		m1rxdata11		m1rxckn		m1rxdata5
vss		m1txdata6		m1txdata8		vss		m1rxdata9		m1rxdata7	
	vss		vss		vss		vss		vss		vss
vccio		m1txdata3		m1txdata13		vccio		m1rxdata12		m1rxdata2	
	m1txdata1		m1txvld		m1txdata15		m1rxdata14		m1rxtrk		m1rxdata0
vccio		vss		vss		vccio		vss		vss	
	m1txdata0		m1txtrk		m1txdata14		m1rxdata15		m1rxvld		m1rxdata1
vss		m1txdata2		m1txdata12		vss		m1rxdata13		m1rxdata3	
Die Edge											

Figure 5-36. Standard Package x32 interface: Signal exit routing

Layer 1	Tx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Rx
Layer 2	Module 1	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module 1
Layer 3	Rx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Tx
Layer 4	Module 2	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module 2
Sideband		m1txdatasb			m2rxdatasb			m1txcksb		m2rxcksb		m2txcksb		m1rxcksb		m2txdatasb			m1rxdatasb			Sideband

Figure 5-37. Standard Package cross section for stacked module



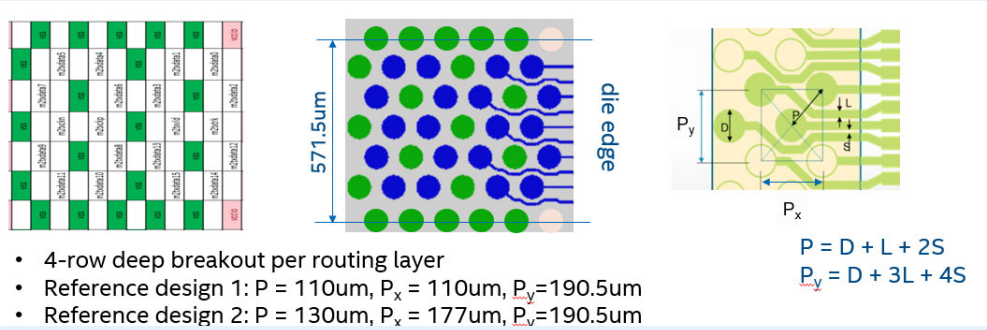
Orange – Tx signals on Die 1 talking to Rx on Die 2 in Layer 1
Orange - Tx signals on Die 1 talking to Rx on Die 2 in Layer 2
Blue - Tx signals on Die 2 talking to Rx on Die 1 in Layer 3
Blue - Tx signals on Die 2 talking to Rx on Die 1 in Layer 4
Brown – VCCIO
Green - VSS

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

IMPLEMENTATION NOTE

Figure 5-38 is a standard package channel reference configuration noting suggested design rule ratios. The 110um dimension is reference to a diagonal pitch, the horizontal and vertical pitches can vary as long as minimum design rules aren't violated. In the example, the horizontal is a multiple of 190.5um.

Figure 5-38. Standard Package reference configuration



5.7.3.2 Module Naming of Standard Package Modules

This section describes the Module naming convention of Standard Package Modules in a multi-module configuration.

The naming of M0, M1, M2, and M3 will apply to 1, 2, or 4 Standard Package modules that are aggregated through MMPL, in stacked and unstacked configuration combinations.

Figure 5-39 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their "Standard Die Rotate" module counterparts with the same number of Standard Package modules, with either same stack or same unstacked configuration.

Note: The double-ended arrows in Figure 5-39 through Figure 5-43 indicate Module-to-Module connections.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-39. Naming Convention for One-, Two-, and Four-module Standard Package Paired with "Standard Die Rotate" Configurations

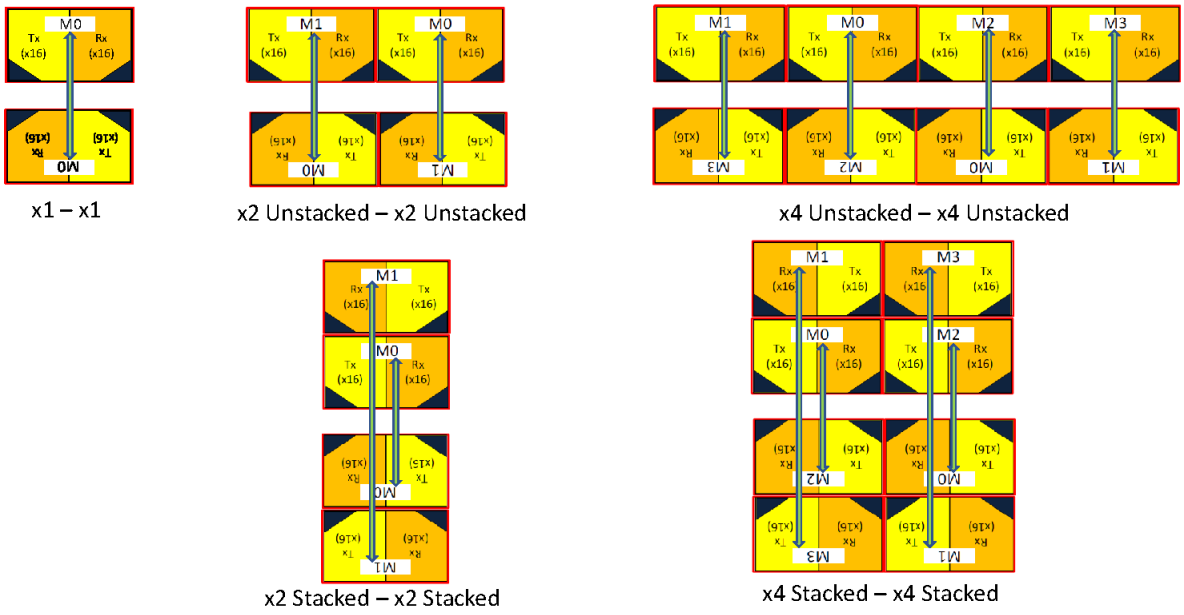
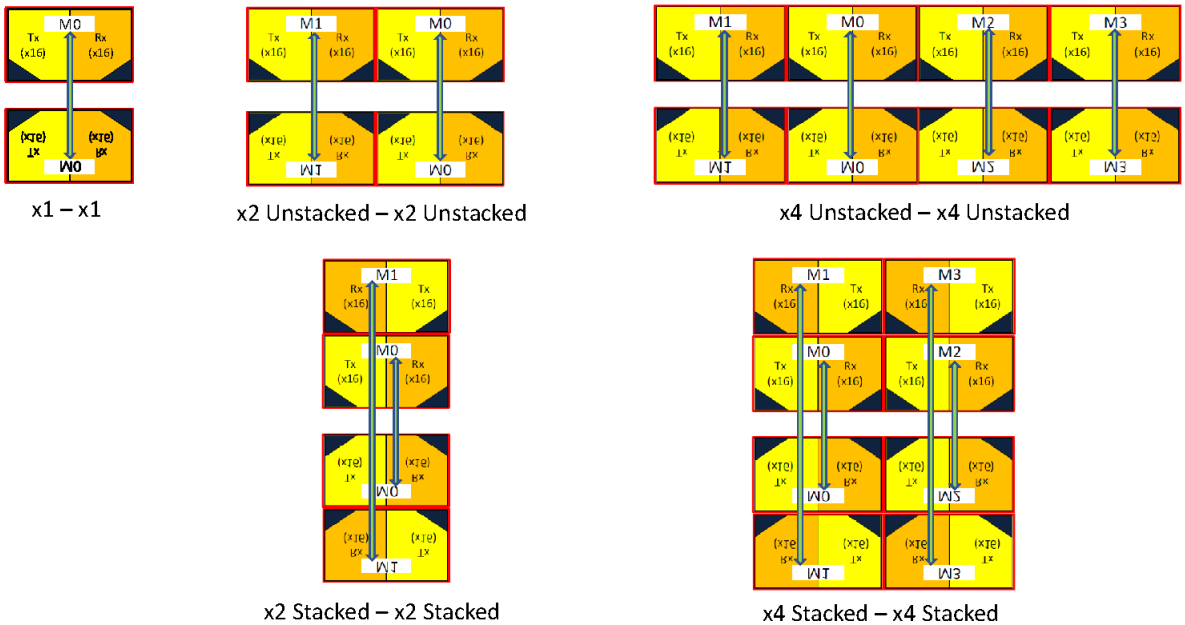


Figure 5-40 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their "Mirrored Die Rotate" counterparts that have same number of Standard Package modules, with either same stack or same unstacked configuration.

Figure 5-40. Naming Convention for One-, Two-, and Four-module Standard Package Paired with "Mirrored Die Rotate" Configurations



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-18 summarizes the connections between the combinations shown in Figure 5-39 and Figure 5-40.

Table 5-18. Summary of Standard Package Module Connection Combinations with Same Number of Modules on Both Sides [1] [2]

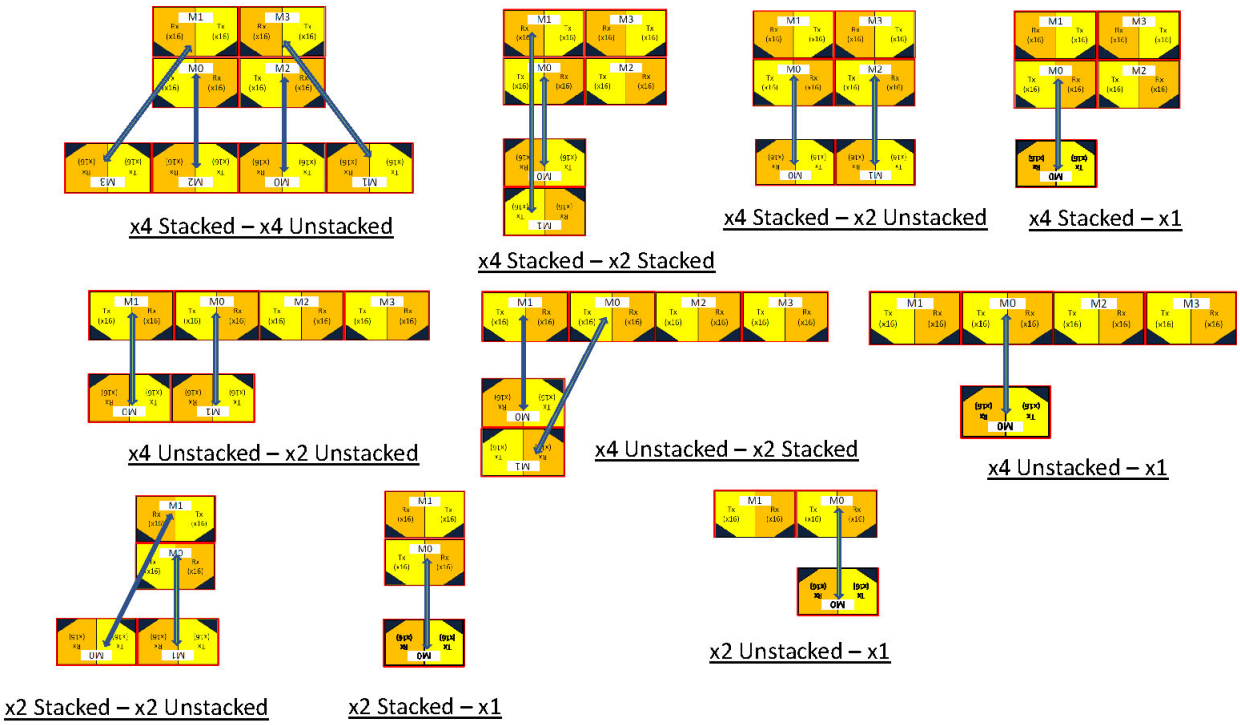
Standard Package Module Connections (Same # of Modules on Both Sides)	Standard Die Rotate Counterpart	Mirrored Die Rotate Counterpart	
		Option 1 (See Figure 5-40)	Option 2[3] (See Figure 5-43)
x1 – x1	• M0 – M0	• M0 – M0	
x2 Unstacked – x2 Unstacked	• M0 – M1 • M1 – M0	• M0 – M0 • M1 – M1	
x2 Stacked – x2 Stacked	• M0 – M0 • M1 – M1	• M0 – M0 • M1 – M1	• M0 – M1 • M1 – M0
x4 Unstacked – x4 Unstacked	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3	
x4 Stacked – x4 Stacked	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3	• M0 – M1 • M1 – M0 • M2 – M3 • M3 – M2

1. Mirror-to-Mirror connection will be same as non-mirrored case.
2. Mirror die connectivity may have jogs and need additional layers on package.
3. For some mirrored cases, there are possible alternative connections to allow design choices between more routing layers vs. max data rates, shown as Option 1 and Option 2 in Table 5-18. For x2 – x2 Stacked and x4 – x4 Stacked cases, Option 1 typically requires 2x the routing layers and enables nominal data rates, while Option 2 enables same the layer count but at reduced max data rates due to potential crosstalk. See Figure 5-42 for Option 2 connection illustrations.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-41 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their "Standard Die Rotate" counterparts that have a different number of Standard Package modules.

Figure 5-41. Examples for Standard Package Configurations Paired with "Standard Die Rotate" Counterparts, with a Different Number of Modules



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-42 shows the naming convention for 1, 2, or 4 Standard Package Modules when they are connected to their "Mirrored Die Rotate" counterparts that have a different number of Standard Package Modules.

Figure 5-42. Examples for Standard Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules

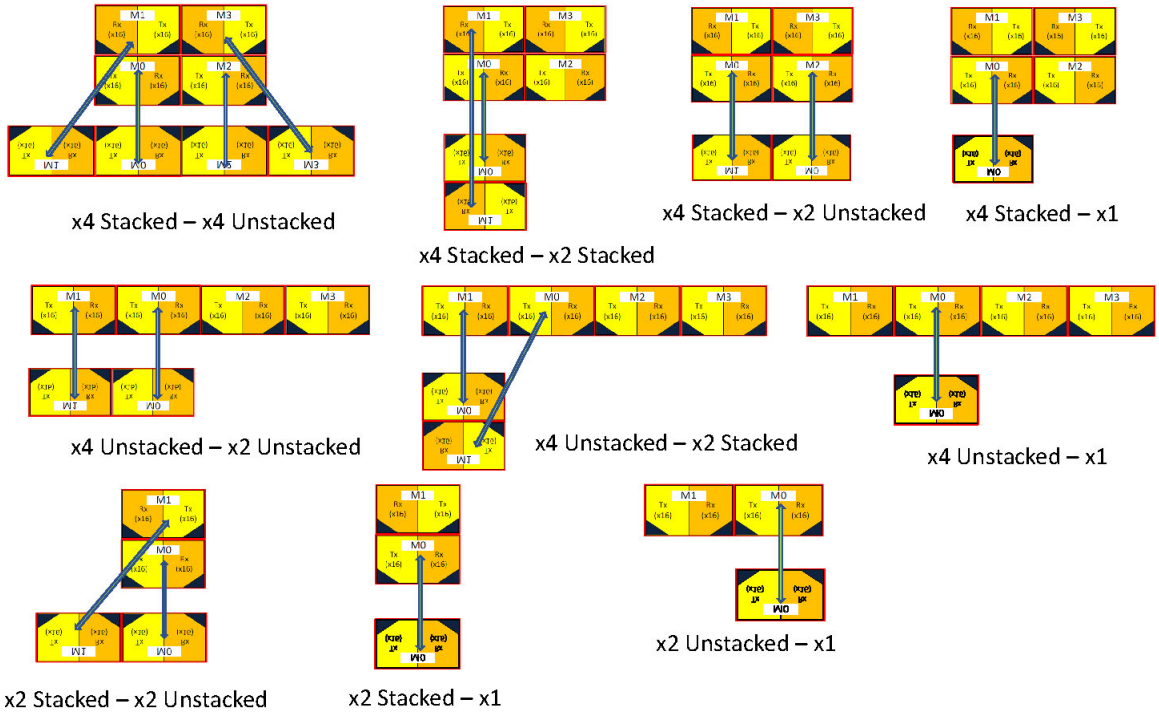
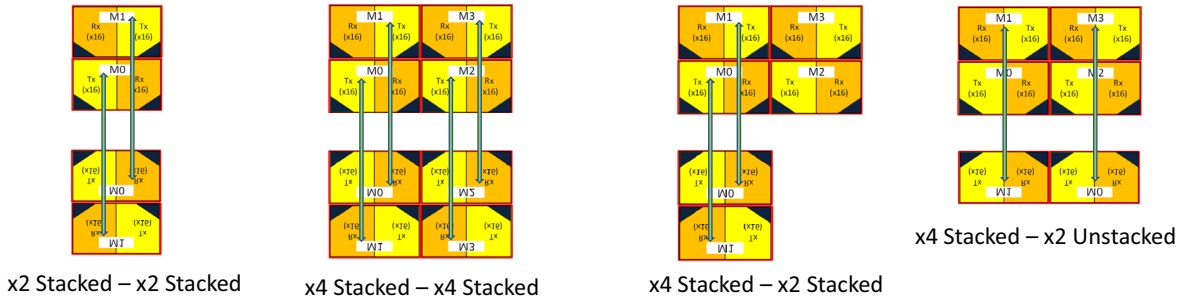


Figure 5-43 illustrates the possible alternative connections for some mirrored cases to allow design choices between more routing layers vs. reduced max data rates due to potential crosstalk, shown as Option 2 in Table 5-18 and Table 5-19.

Figure 5-43. Additional Examples for Standard Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 5-19 summarizes the connections between the combinations shown in Figure 5-41, Figure 5-42, and Figure 5-43.

Table 5-19. Summary of Standard Package Module Connection Combinations with Different Number of Modules on Both Sides

Standard Package Module Connections (Different # of Modules on Both Sides)	Standard Die Rotate Counterpart[1]	Mirrored Die Rotate Counterpart[1]	
		Option 1 (See Figure 5-42)	Option 2 (See Figure 5-43)
x4 Stacked – x4 Unstacked	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3	
x4 Stacked – x2 Stacked	• M0 – M0 • M1 – M1 • M3 – NC • M2 – NC	• M0 – M0 • M1 – M1 • M2 – NC • M3 – NC	• M0 - M1 • M1 - M0 • M2 – NC • M3 – NC
x4 Stacked – x2 Unstacked	• M0 – M0 • M1 – NC • M3 – NC • M2 – M1	• M0 – M1 • M1 – NC • M2 – M0 • M3 - NC	• M0 – NC • M1 – M1 • M2 – NC • M3 – M0
x4 Stacked – x1	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	
x4 Unstacked – x2 Unstacked	• M0 – M1 • M1 – M0 • M3 – NC • M2 – NC	• M0 – M0 • M1 – M1 • M2 – NC • M3 – NC	
x4 Unstacked – x2 Stacked	• M0 – M1 • M1 – M0 • M3 – NC • M2 – NC	• M0 – M1 • M1 – M0 • M2 – NC • M3 – NC	
x4 Unstacked – x1	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	
x2 Stacked – x2 Unstacked	• M0 – M1 • M1 – M0	• M0 – M0 • M1 – M1	
x2 Stacked – x1	• M0 – M0 • M1 – NC	• M0 – M0 • M1 – NC	
x2 Unstacked – x1	• M0 – M0 • M1 – NC	• M0 – M0 • M1 – NC	

1. NC indicates no connection.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.7.3.2.1 Module Degrade Rules

On a 2-module or 4-module link, if one or more module-pairs have failed, the link will be degraded and shall comply with the following rules:

1. The degraded link shall be either one or two modules, and shall not be three modules.
   a. For a 4-module link:
      i. If any one module-pair failed, it shall be degraded to a 2-module link.
      ii. If any two module-pairs failed, it shall be degraded to a 2-module link.
      iii. If any three module-pairs failed, it shall be degraded to a 1-module link.
   b. For a 2-module link:
      i. If any one module-pair failed, it shall be degraded to a 1-module link.
2. For a 4-module link, if only one module-pair failed, one additional module-pair that belongs to the "same half" (along the Die Edge) of the 4-module will be disabled/degraded.

Figure 5-44 illustrates an example with a x4 Unstacked connected to a x4 Unstacked "Standard Die Rotate" counterpart with one M0 – M2 pair failed. The M1 – M3 pair on its left shall be disabled accordingly to comply with the rules defined above, which will be denoted as "x (d)" in Table 5-20.

Note: The double-ended arrows in Figure 5-44 indicate Module-to-Module connections.

Figure 5-44. Example of a Configuration for Standard Package, with Some Modules Disabled

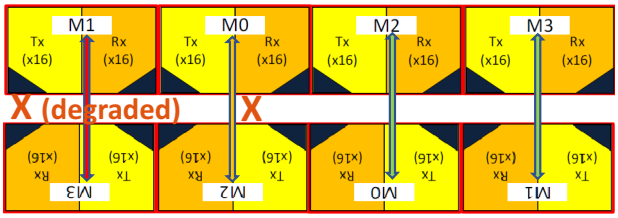


Table 5-20 summarizes the resulting degraded link if there are one, two, or three failed module-pairs for the x4 Unstacked to x4 Unstacked configuration.

Table 5-20. Summary of Degraded Links when Standard Package Module-pairs Fail

Module – Module Partner Pair	Number of Module-pairs Failed[1]													
	1-fail				2-fail						3-fail			
M0 – M2	x	x (d)	✓	✓	x	x	x	✓	✓	✓	x	x	x	✓
M1 – M3	x (d)	x	✓	✓	x	✓	✓	x	x	✓	x	x	✓	x
M3 – M1	✓	✓	x	x (d)	✓	x	✓	x	✓	x	x	✓	x	x
M2 – M0	✓	✓	x (d)	x	✓	✓	x	✓	x	x	✓	x	x	x

1. x = Failed Module – Module Partner Pair.
   x (d) = Disabled Module – Module Partner Pair to comply with Degrade rules.
   ✓ = Functional Module – Module Partner Pair.

All other module configurations shall follow the same Module Degrade rules as defined above.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.8 Tightly Coupled mode

Tightly Coupled PHY mode is defined as when both of the following conditions are met:

- Shared Power Supply between TX and RX, or Forwarded Power Supply from TX to RX
- Channel supports larger eye mask defined in Table 5-21

In this mode, there is no Receiver termination and Transmitter must provide full swing output. In this mode further optimization of PHY circuit and power reduction is possible. For example, tuned inverter can potentially used instead of a front-end amplifier. Training complexity such as voltage reference can be simplified.

Table 5-21. Tightly coupled mode: Eye mask

Data Rate	4-16 GT/s
Overall (Eye Closure due to Channel)[1]	
Eye Height[2]	250mV
Eye Width (rectangular eye mask with specified eye height)	0.7 UI

1. With 750 mV Transmitter signal swing.
2. Centered around VCCFWDIO/2.

Loss and crosstalk requirement follow the same VTF method, adjusting to the eye mask defined in Table 5-21. Table 5-22 shows the specification at Nyquist frequency.

Table 5-22. Tightly coupled mode channel for Advanced Package

Data Rate	4-12 GT/s	16 GT/s
VTF Loss[1] (dB)	$L(f_N) > -3$	-
VTF Crosstalk[1] (dB)	$XT(f_N) < 1.5 * L(f_N) - 21.5$ and $XT(f_N) < -23$	-

1. Based on Voltage Transfer Function (Tx: 25 ohm / 0.25pF; Rx: 0.2pF).

Loss and crosstalk specifications between DC and Nyquist f_N follow the same methodology defined in Section 5.7.2.1.

Although the use of this mode is primarily for Advanced Package, it may also be used for Standard Package when two Dies are near one another and Receiver must be unterminated.

5.9 Interconnect redundancy remapping

5.9.1 Advanced Package Lane remapping

Interconnect Lane remapping is supported in Advanced Package Module to improve assembly yield and recover functionality. Each module supports:

- Four redundant bumps for Data
- One redundant bump for Clock and Track
- One redundant bump for Valid

For x64 Advanced Package modules, the four redundant bumps for data repair are divided into two groups of two. Figure 5-45 shows an illustration of x64 Advanced package module redundant bump assignment for data signals. TRD_P0 and TRD_P1 are allocated to the lower 32 data Lanes and TRD_P2 and TRD_P3 are allocated to the upper 32 data Lanes. Each group is allowed to remap up to

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

two Lanes. For example, TD15 is a broken Lane in the lower half and TD_P32 and TD_P40 are broken Lanes in the upper 32 Lanes. Figure 5-46 illustrates Lane remapping for the broken Lanes.

For x32 Advanced Package modules, only the lower 32 data lanes and TRD_P0 and TRD_P1 apply in Figure 5-45 and Figure 5-46.

Details and implementation of Lane remapping for Data, Clock, Track, and Valid are provided in Section 4.3.

Figure 5-45. Data Lane repair resources

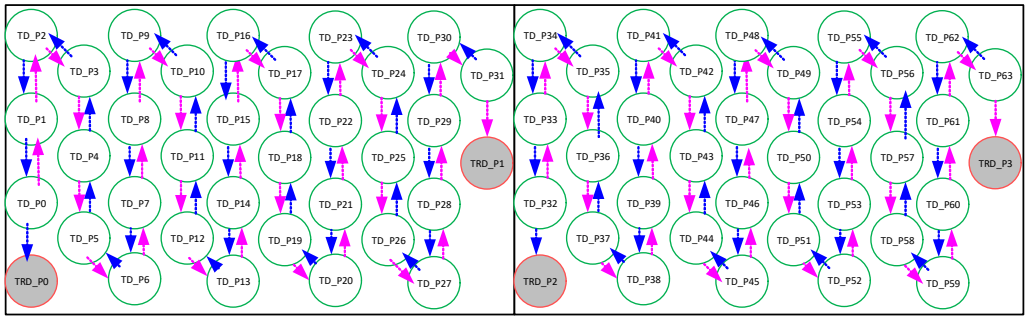
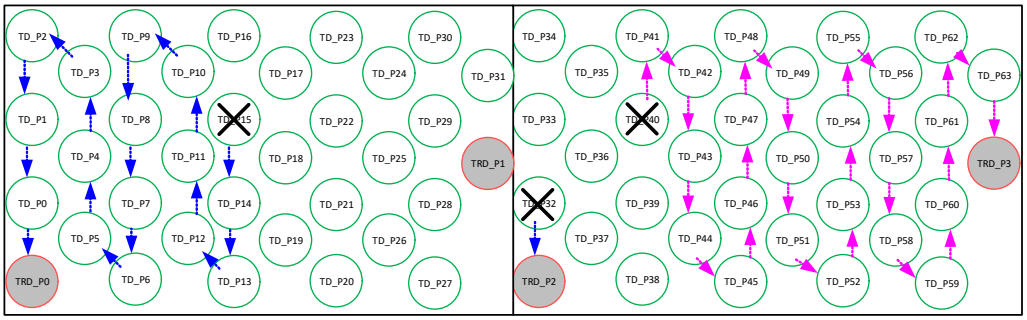


Figure 5-46. Data Lane repair



5.9.2 Standard Package Lane remapping

Lane repair is not supported in Standard Package modules.

5.10 BER requirements, CRC and retry

The BER requirement based on channel reach defined in Section 5.7 is shown in Table 5-23. Error detection and correction mechanisms such as CRC and retry are required for BER for 1E-15 to achieve the required Failure In Time (FIT) rate of significantly less than 1 (1 FIT = 1 device failure in 10^9 Hours). The UCIe spec defined CRC and retry is detailed in Chapter 3.0. For the BER of 1E-27, either parity or CRC can be used and the appropriate error reporting mechanism must be invoked to ensure a FIT that is significantly less than 1.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

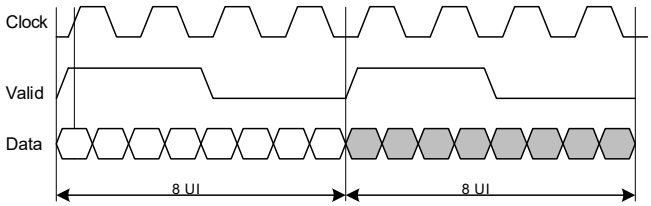
Table 5-23. Raw BER requirements

Package Type	Data Rate (GT/s)					
	4	8	12	16	24	32
Advanced Package	1E-27	1E-27	1E-27	1E-15	1E-15	1E-15
Standard Package	1E-27	1E-27	1E-15	1E-15	1E-15	1E-15

5.11 Valid and Clock Gating

Valid is used to frame transmit data. For a single transmission of 8 UI data packet, Valid is asserted for the first 4 UI and de-asserted for the second 4 UI. Figure 5-47 shows the transfer of two 8 UI data packets back to back.

Figure 5-47. Valid Framing

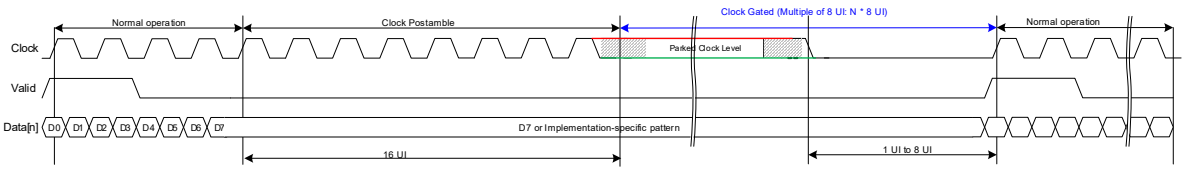


As described in Section 4.1.3, clock must be gated only after Valid signal remains low for 16 UI (8 cycles) of postamble clock for half-rate clocking and 32 UI (8 cycles) of postamble clock for quarter-rate clocking, unless free running clock mode is negotiated.

Idle state is when there is no data transmission on the mainband. During Idle state, Data, Clock, and Valid Lanes must hold values as follows:

- If the Link is unterminated (all Advanced Package and unterminated Standard Package Links), some Data Lane Transmitters are permitted to remain toggling up to the same transition density as the scrambled data without advancing the scrambler state. The remaining Data Lane Transmitters must hold the data of the last transmitted bit. Valid Lane must be held low until the next normal transmission.
  - In Strobe mode, the clock level in a clock-gated state for half-rate clocking (after meeting postamble requirement) must alternate between differential high and differential low during consecutive clock-gating events. For quarter-rate clocking, the clock level in a clock-gated state must alternate between high and low for both phases (Phase-1 and Phase-2) simultaneously. Clock must drive a differential (simultaneous) low for half- (quarter-) rate clocking for at least 1 UI or a maximum of 8 UI before normal operation. The total clock-gated period must be an integer multiple of 8 UI. Example shown in Figure 5-48 and Figure 5-49.
  - In Continuous mode, the clock remains free running (examples shown in Figure 5-50). Total idle period must be an integer multiple of 8 UI.

Figure 5-48. Data, Clock, Valid Levels for Half-rate Clocking: Clock-gated Unterminated Link



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-49. Data, Clock, Valid Levels for Quarter-rate Clocking: Clock-gated Unterminated Link

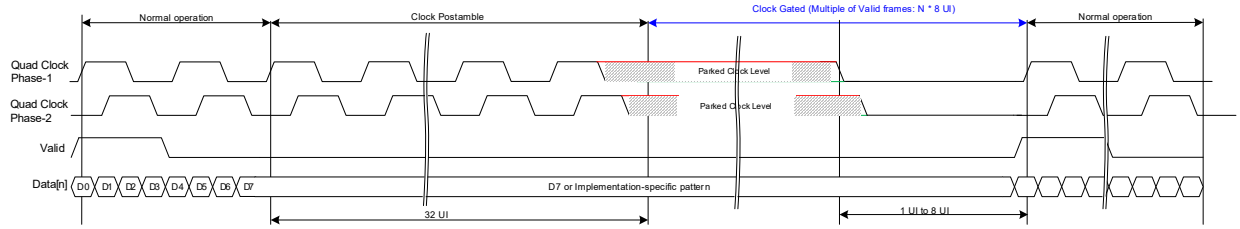
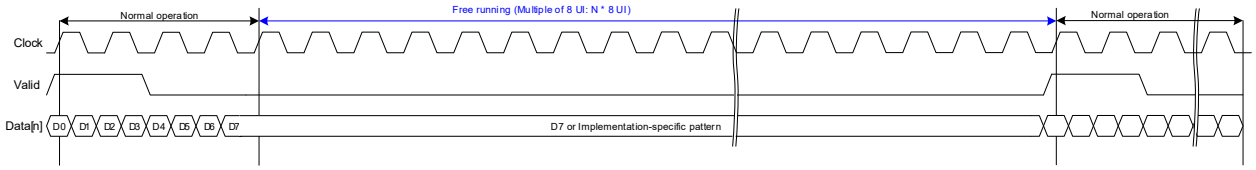


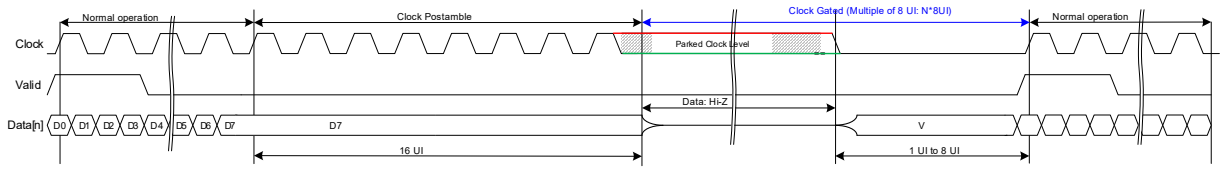
Figure 5-50. Data, Clock, Valid Levels for Half-rate Clocking: Continuous Clock Unterminated Link



- If the Link is terminated (Standard Package terminated Links), Data Lanes Transmitters must hold the data of the last transmitted bit for 16 UIs (32 UIs) for half- (quarter-) rate clocking and then go Hi-Z. Valid Lane must be held low until the next normal transmission.
  - — In Strobe mode, the clock level in a clock-gated state for half-rate clocking (after meeting postamble requirement) must alternate between differential high and differential low during consecutive clock-gating events. For quarter-rate clocking, the clock level in a clock-gated state must alternate between high and low for both phases (Phase-1 and Phase-2) simultaneously. Transmitters must precondition the Data Lanes to a 0 or 1 (V) and clock must drive a differential low for at least 1 UI or up to a maximum of 8 UIs for half- (quarter-) rate clocking before the normal transmission. The total clock-gated period must be an integer multiple of 8 UI. Example shown in Figure 5-51 and Figure 5-53.
  - — In Continuous mode, the clock remains free running (examples shown in Figure 5-54). Transmitters must precondition the Data Lanes to a 0 or 1 (V) for at least 1 UI or up to a maximum of 8 UI. Total idle period must be an integer multiple of 8 UI.

Note: Entry into and Exit from Hi-Z state are analog transitions. Hi-Z represents Transmitter state and the actual voltage during this period will be pulled Low due to termination to **ground** at the Receiver.

Figure 5-51. Data, Clock, Valid Gated Levels for Half-rate Clocking: Terminated Link



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 5-52. Data, Clock, Valid Gated Levels for Quarter-rate Clocking: Terminated Link

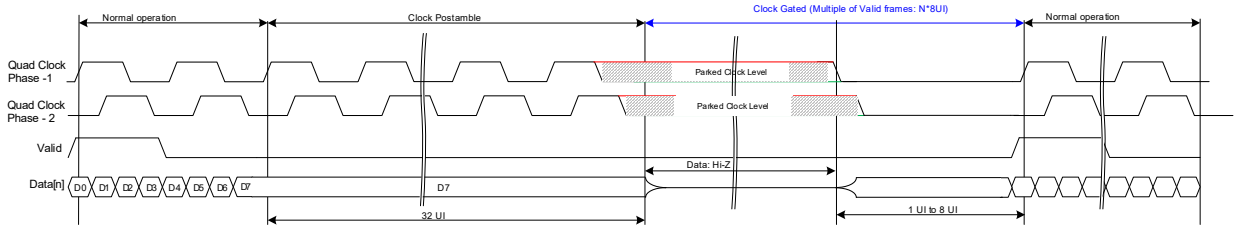
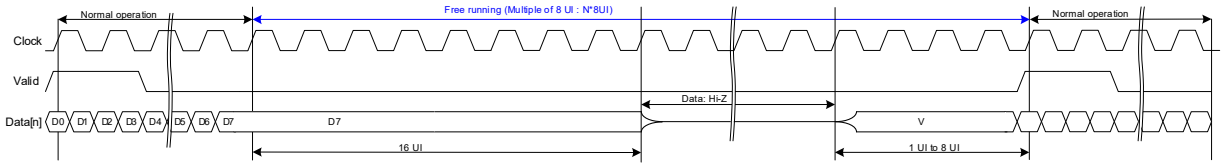


Figure 5-53. Data, Clock, Valid Gated Levels for Half-rate Clocking: Continuous Clock Terminated Link



5.12 Electrical Idle

Some training states need electrical idle when Transmitters and Receivers are waiting for generate and receive patterns.

- Electrical idle on the mainband in this Specification is described as when Transmitters and Receivers are enabled; Data, Valid and Track Lanes are held low and Clock is parked at high and low.

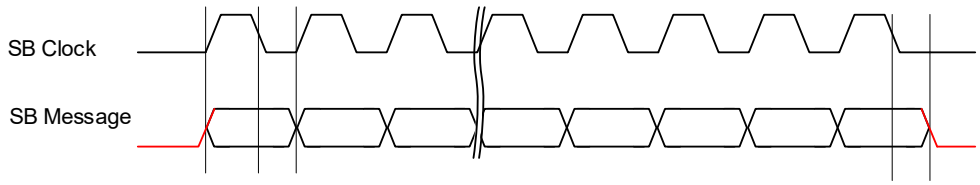
5.13 Sideband signaling

Each module supports a sideband interface. The sideband is a two-signal interface that is used for the transmit and receive directions. The sideband data is an 800 MT/s single data rate signal with an 800-MHz source. Sideband must run on power supply and auxiliary clock source which are always on (VCCAON).

Sideband data is sent edge aligned with the strobe. The Receiver must sample the incoming data with the strobe. For example, the negative edge of the strobe can be used to sample the data as the data uses single data rate signaling as shown in Figure 5-54.

For Advanced Package modules, redundancy is supported for the sideband interface. Sideband initialization and repair are described in Section 4.5.3.2. There is no redundancy and no Lane repair support on Standard Package modules.

Figure 5-54. Sideband signaling



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

5.13.1 Sideband Electrical Parameters

Table 5-24 shows the sideband electrical parameters.

It is strongly recommended that the two sides of the sideband I/O Link share the same power supply rail.

Table 5-24. Sideband Parameters summary

Parameter	Min	Typ	Max	Unit
Supply voltage (VCCAON)[1]	0.65			V
TX Swing	0.8*VCCAON	-	-	V
Input high voltage (V_{IH})	0.7*VCCAON			V
Input low voltage (V_{IL})			0.3*VCCAON	V
Output high voltage (V_{OH})	0.9*VCCCAON			V
Output low voltage (V_{OL})			0.1*VCCAON	V
Sideband Data Setup Time	200	-	-	ps
Sideband Data Hold Time	200	-	-	ps
Rise/Fall time for Advanced Package[2]	50	-	280	ps
Rise/Fall time for Standard Package[3]	80	-	175	ps

1. Always On power supply. The guidelines for maximum Voltage presented in Section 1.5 apply to sideband signaling.
2. 20 to 80% of VCCAON level with Advanced Package reference channel load.
3. 20 to 80% of VCCAON level with Standard Package reference channel load.

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.0 Sideband

6.1 Protocol specification

The usage for the sideband Link is to provide a out of band channel for Link training and an interface for sideband access of registers of the Link partner. It is also used for Link management packets and parameter exchanges with remote Link partner.

The same protocol is also used for local die sideband accesses over FDI and RDI. When relevant, FDI specific rules are pointed out using "FDI sideband:". When relevant, RDI specific rules are pointed out using "RDI sideband:". When relevant, UCIe Link specific rules are pointed out using "UCIe Link sideband:". If no prefix is mentioned, it is a common rule across FDI, RDI and UCIe Link.

The Physical Layer is responsible for framing and transporting sideband packets over the UCIe Link. Direct sideband access to remote die can originate from the Adapter or the Physical Layer. The Adapter forwards a remote die sideband access over RDI to the Physical Layer for framing and transport. These include register access requests, completions or messages.

The Protocol Layer has indirect access to remote die registers using the sideband mailbox mechanism. The mailbox registers reside in the Adapter, and it is the responsibility of the Adapter to initiate remote die register access requests when it receives the corresponding access trigger for the mailbox register over FDI.

FDI sideband: In the case of multi-protocol stacks, the Adapter must track which protocol stack sent the original request and route the completion back to the appropriate protocol stack.

FDI sideband: Because the Protocol Layer is only allowed indirect access to remote die registers, and direct access to local die registers, currently only Register Access requests and completions are permitted on the FDI.

All sideband requests that expect a response have an 8ms timeout. A "Stall" encoding is provided for the relevant packets for Retimers, to prevent timeouts if the Retimer needs extra time to respond to the request. When stalling to prevent timeouts, it is the responsibility of the Retimer to send the corresponding Stall response once every 4ms. The Retimer must also ensure that it does not Stall indefinitely, and escalates a Link down event after a reasonable attempt to complete resolution that required stalling the requester. If a requester receives a response with a "Stall" encoding, it resets the timeout counter.

In certain cases, it is necessary for registers to be fragmented between the different layers; i.e., certain bits of a given register physically reside in the Protocol Layer, other bits reside in the Adapter, and other bits reside in the Physical Layer. UCIe takes a hierarchical decoding for these registers. For fragmented registers, if a bit does not physically reside in a given Layer, it implements that bit as Read Only and tied to 0. Hence reads would return 0 for those bits from that Layer, and writes would have no effect on those bits. As an example, for reads, Protocol Layer would forward these requests to the Adapter on FDI and the Protocol Layer will OR the data responded by the Adapter with its local register before responding to software. The Adapter must do the same if any bits of that register reside in the Physical Layer before responding to the Protocol Layer.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.1.1 Packet Types

Three different categories of packets are permitted:

- Register Accesses: These can be Configuration (CFG) or Memory Mapped accesses for both Reads or Writes are supported. These can be 32-bit or 64-bit in length.
- Messages without data: These can be Link Management (LM), or Vendor Defined Packets. These don't carry additional data payloads.
- Messages with data: These can be Parameter Exchange (PE), Link Training related or Vendor Defined, and carry 64b of data.

Every packet carries a 5-bit opcode, 3-bit source identifier (srcid), and a 3-bit destination identifier (dstid). The 5-bit opcode indicates the packet type, as well as whether it carries 32b of data or 64b of data.

Table 6-1 gives the mapping of opcode encodings to Packet Types.

Table 6-1. Opcode encodings mapped to Packet Types

Opcode Encoding	Packet Type
00000b	32b Memory Read
00001b	32b Memory Write
00100b	32b Configuration Read
00101b	32b Configuration Write
01000b	64b Memory Read
01001b	64b Memory Write
01100b	64b Configuration Read
01101b	64b Configuration Write
10000b	Completion without Data
10001b	Completion with 32b Data
10010b	Message without Data
11001b	Completion with 64b Data
11011b	Message with 64b Data
Other encodings	Reserved

Table 6-2, Table 6-3, and Table 6-4 give the encodings of source and destination identifiers. It is not permitted for Protocol Layer from one side of the Link to directly access Protocol Layer of the remote Link partner over sideband (this should be done via mainband).

Table 6-2. FDI sideband: srcid and dstid encodings on FDI

Field[1]	Description
srcid[2:0]	000b: Stack 0 Protocol Layer 100b: Stack 1 Protocol Layer other encodings are reserved.
dstid[2:0]	001b: D2D Adapter 010b: Physical Layer other encodings are reserved.

1. srcid and dstid are Reserved for completion messages transferred over FDI. The Protocol Layer must correlate the completions to original requests using the Tag field. Currently, no requests are permitted from Adapter to Protocol Layer over FDI sideband.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-3. RDI sideband: srcid and dstid encodings on RDI

Field[1]	Description
srcid[2:0]	000b: Stack 0 Protocol Layer 001b: D2D Adapter 100b: Stack 1 Protocol Layer other encodings are reserved.
dstid[2]	0b: Local die terminated request 1b: Remote die terminated request
dstid[1:0]	For Local die terminated requests: 10b: Physical Layer other encodings are reserved. For Remote die terminated Register Access Requests: dstid[1:0] is Reserved For Remote die terminated Register Access Completions: 01b: D2D Adapter other encodings are reserved. For Remote die terminated messages: 01b: D2D Adapter message 10b: Physical Layer message

1. srcid and dstid are Reserved for completion messages transferred over RDI for local Register Access completions. For Register Access completions, the Adapter must correlate the completions to original requests using the Tag field regardless of dstid field. Both local and remote Register Access requests are mastered by the Adapter with unique Tag encodings.

Table 6-4. UCIe Link sideband: srcid and dstid encodings for UCIe Link

Field	Description
srcid[2:0]	001b: D2D Adapter 010b: Physical Layer other encodings are reserved
dstid[2]	1b: Remote die terminated request other encodings are reserved
dstid[1:0]	For Register Access requests: dstid[1:0] is Reserved. For Remote die terminated Register Access Completions: 01b: D2D Adapter other encodings are reserved. For Remote die terminated messages: 01b: D2D Adapter message 10b: Physical Layer message

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.1.2 Packet Formats

All the figures in this section show examples assuming a 32-bit interface of RDI/FDI transfer for sideband packets, hence the headers and data are shown in Phases of 32 bits.

6.1.2.1 Register Access Packets

Figure 6-1 shows the packet format for Register Access requests. Table 6-5 gives the description of the fields other than the opcode, srcid and dstid.

Table 6-5. Field descriptions for Register Access Requests

Field	Description
CP	Control Parity (CP) is the even parity of all the header bits excluding DP.
DP	Data Parity is the even parity of all bits in the data payload. If there is no data payload, this bit is set to 0b.
Cr	If 1b, indicates one credit return for credited sideband messages. This field is only used by the Adapter for remote Link partner's credit returns for E2E credits. It is not used for local FDI or RDI credit loops.
Addr[23:0]	Address of the request. Different opcodes use this field differently. See Table 6-6 for details. The following rules apply for the address field: For 64-bit request, Addr[2:0] is reserved. For 32-bit request, Addr[1:0] is reserved.
BE[7:0]	Byte Enables for the Request. It is NOT required to be contiguous. BE[7:4] are reserved if the opcode is for a 32-bit request.
EP	Data Poison. If poison forwarding is enabled, the completer can poison the data on internal errors.
Tag[4:0]	Tag is a 5-bit field generated by the requester, and it must be unique for all outstanding requests that require a completion. The original requester uses the Tag to associate returning completions with the original request.
Data	Payload. Can be 32 bits or 64 bits wide depending on the Opcode.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-6. Mapping of Addr[23:0] for different requests

Opcode	Description
Memory Reads/Writes	{RL[3:0], Offset[19:0]} Offset is the Byte Offset. RL[3:0] encodings are as follows: 0h: Register Locator 0 1h: Register Locator 1 2h: Register Locator 2 3h: Register Locator 3 Fh: Accesses for Protocol specific MMIO registers that are shadowed in the Adapter (e.g., ARB/MUX registers defined in the *CXL Specification*). The offsets for these registers are implementation specific, and the protocol layer must translate accesses to match the offsets implemented in the Adapter. Other encodings are reserved. For accesses to Reserved RL encodings, the completer must respond with a UR.
Configuration Reads/Writes	{RL[3:0], Rsvd[7:0], Byte Offset[11:0]}, where RL[3:0] encodings are as follows: 0h: UCIe Link DVSEC Fh: Accesses for Protocol specific configuration registers that are shadowed in the Adapter (e.g., ARB/MUX registers defined in the *CXL Specification*). The offsets for these registers are implementation specific, and the protocol layer must translate accesses to match the offsets implemented in the Adapter. Other encodings are reserved. For accesses to Reserved RL encodings, the completer must respond with a UR.

Figure 6-1. Format for Register Access request

Register Access Request																															
Bytes	**3**								**2**								**1**								**0**							
Bits	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Header / Data	**Header**																															
Phase0	srcid				rsvd				tag[4:0]								be[7:0]								rsvd					ep	opcode[4:0]	
Phase1	dp	cp	cr		rsvd				dstid								addr[23:0]															
Header / Data	**Data (if applicable, can be 32 bits or 64 bits)**																															
Phase2	data[31:0]																															
Phase3	data[63:32]																															

Figure 6-2 gives the format for Register Access completions.

Figure 6-2. Format for Register Access completions

Register Access Completions																																
Bytes	**3**								**2**								**1**								**0**							
Bits	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Header / Data	**Header**																															
Phase0	srcid				rsvd				tag[4:0]								be[7:0]								rsvd					ep	opcode[4:0]	
Phase1	dp	cp	cr		rsvd				dstid								rsvd														Status	
Header / Data	**Data (if completion with data, can be 32 bits or 64 bits)**																															
Phase2	data[31:0]																															
Phase3	data[63:32]																															

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-7 gives the field descriptions for a completion.

Table 6-7. Field Descriptions for a Completion

Field	Description
Tag[4:0]	Completion Tag associated with the corresponding Request. The requester uses this to associate the completion with the original request.
CP	Control Parity. All fields other than "DP" and "CP" in the Header are protected by Control Parity, and the parity scheme is even (including reserved bits)
DP	Data Parity. All fields in data are protected by data parity, and the parity scheme is even.
Cr	If 1b, indicates one credit return for credited sideband messages. This field is only used by the Adapter for remote Link partner's credit returns for E2E credits. It is not used for local FDI or RDI credit loops.
EP	Data Poison. If poison forwarding is enabled, the completer can poison the data on internal errors.
BE[7:0]	Byte Enables for the Request. Completer returns the same value that the original request had (this avoids the requester from having to save off the BE value). BE[7:4] are reserved if the opcode is for a 32-bit request.
Status[2:0]	Completion Status 000b - Successful Completion (SC). This can be a completion with or without data, depending on the original request (it must set the appropriate Opcode). If the original request was a write, it is a completion without data. If the original request was a read, it is a completion with data. 001b - Unsupported Request (UR). On UCIe, this is a completion with 64b Data when a request is aborted by the completer, and the Data carries the original request header that resulted in UR. This enables easier header logging at the requester. Register Access requests that timeout must also return UR status, but for those the completion is without Data. 100b - Completer Abort (CA). On UCIe, this is a completion with 64b Data, and the Data carries original request header that resulted in UR. This enables easier header logging at the requester. 111b - Stall. Receiving a completion with Stall encoding must reset the timeout at the requester. Completer must send a Stall once every 4ms if it is not ready to respond to the original request. Other encodings are reserved. An error is logged in the Sideband Mailbox Status if a CA was received or if the number of timeouts exceed the programmed threshold. For timeouts below the programmed threshold, a UR is returned to the requester.
Data	Payload. 32 bits or 64 bits depending on the Opcode.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.1.2.2 Messages without Data

Figure 6-3 shows the Format for Messages without data payloads. These can be Link Management packets, NOPs or Vendor Defined message packets.

Figure 6-3. Format for Messages without Data

Messages without Data																															
Bytes	3								2								1								0							
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header / Data	Header																															
Phase0	srcid			rsvd		rsvd			msgcode[7:0]								rsvd								opcode[4:0]							
Phase1	dp	cp		rsvd		dstid			MsgInfo[15:0]																MsgSubcode[7:0]							

The definitions of opcode, srcid, dstid, dp, and cp fields are the same as Register Access packets. Table 6-8 and Table 6-9 give the encodings of the different messages without data that are send on UCIe. Some Notes on the different message categories are listed below:

- {NOP.Crd} - These are used for E2E Credit returns. The destination must be D2D Adapter.
- {LinkMgmt.RDI.*} - These are used to coordinate RDI state transitions, the source and destination is Physical Layer.
- {LinkMgmt.Adapter0.*} - These are used to coordinate Adapter LSM state transitions for the Adapter LSM corresponding to Stack 0 Protocol Layer. The source and destination is D2D Adapter.
- {LinkMgmt.Adapter1.*} - These are used to coordinate Adapter LSM state transitions for the Adapter LSM corresponding to Stack 1 Protocol Layer. The source and destination is D2D Adapter.
- {ParityFeature.*} - This is used to coordinate enabling of the Parity insertion feature. The source and destination for this must be the D2D Adapter.
- {ErrMsg} - This is used for error reporting and escalation from the remote Link Partner. This is sent from the Retimer or Device die to the Host, and the destination must be the D2D Adapter.

Table 6-8. Message Encodings for Messages without Data (Sheet 1 of 3)

Name	Msgcode	Msgsubcode	MsgInfo	Description
{Nop.Crd}	00h	00h	0000h:Reserved 0001h:1 Credit return 0002h: 2 Credit returns 0003h: 3 Credit returns 0004h: 4 Credit returns	Explicit Credit return from Remote Link partner for credited messages.
{LinkMgmt.RDI.Req.Active}	01h	01h	Reserved	Active Request for RDI SM.
{LinkMgmt.RDI.Req.L1}		04h		L1 Request for RDI SM.
{LinkMgmt.RDI.Req.L2}		08h		L2 Request for RDI SM.
{LinkMgmt.RDI.Req.LinkReset}		09h		LinkReset Request for RDI SM.
{LinkMgmt.RDI.Req.LinkError}		0Ah		LinkError Request for RDI SM.
{LinkMgmt.RDI.Req.Retrain}		0Bh		Retrain Request for RDI SM.
{LinkMgmt.RDI.Req.Disable}		0Ch		Disable Request for RDI SM.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-8. Message Encodings for Messages without Data (Sheet 2 of 3)

Name	Msgcode	Msgsubcode	MsgInfo	Description
{LinkMgmt.RDI.Rsp.Active}	02h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for RDI SM.
{LinkMgmt.RDI.Rsp.PMNAK}		02h		PMNAK Response for RDI SM
{LinkMgmt.RDI.Rsp.L1}		04h		L1 Response for RDI SM.
{LinkMgmt.RDI.Rsp.L2}		08h		L2 Response for RDI SM.
{LinkMgmt.RDI.Rsp.LinkReset}		09h		LinkReset Response for RDI SM.
{LinkMgmt.RDI.Rsp.LinkError}		0Ah		LinkError Response for RDI SM.
{LinkMgmt.RDI.Rsp.Retrain}		0Bh		Retrain Response for RDI SM.
{LinkMgmt.RDI.Rsp.Disable}		0Ch		Disable Response for RDI SM.
{LinkMgmt.Adapter 0.Req.Active}	03h	01h	0000h: Regular Request FFFFh: Stall	Active Request for Stack 0 Adapter LSM. The Stall encoding is provided for Retimers to avoid the Adapter LSM transition to Active timeout as described in Section 7.5.3.8.
{LinkMgmt.Adapter 0.Req.L1}		04h	Reserved	L1 Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Req.L2}		08h		L2 Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Req.LinkReset}		09h		LinkReset Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Req.Disable}		0Ch		Disable Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.Active}	04h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.PMNAK}		02h		PMNAK Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.L1}		04h		L1 Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.L2}		08h		L2 Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.LinkReset}		09h		LinkReset Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.Disable}		0Ch		Disable Response for Stack 0 Adapter LSM.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-8. Message Encodings for Messages without Data (Sheet 3 of 3)

Name	Msgcode	Msgsubcode	MsgInfo	Description
{LinkMgmt.Adapter 1.Req.Active}	05h	01h	0000h: Regular Request FFFFh: Stall	Active Request for Stack 1 Adapter LSM. The Stall encoding is provided for Retimers to avoid the Adapter LSM transition to Active timeout as described in Section 7.5.3.8.
{LinkMgmt.Adapter 1.Req.L1}		04h	Reserved	L1 Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Req.L2}		08h		L2 Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Req.LinkReset}		09h		LinkReset Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Req.Disable}		0Ch		Disable Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Rsp.Active}	06h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Rsp.PMNAK}		02h		PMNAK Response for Stack 1 Adapter LSM
{LinkMgmt.Adapter 1.Rsp.L1}		04h		L1 Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Rsp.L2}		08h		L2 Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Rsp.LinkReset}		09h		LinkReset Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Rsp.Disable}		0Ch		Disable Response for Stack 1 Adapter LSM.
{ParityFeature.Req}	07h	00h	Reserved	Parity Feature enable request.
{ParityFeature.Ack}	08h	00h	0000h: Regular Response FFFFh: Stall Response	Parity Feature enable Ack.
{ParityFeature.Nak}		01h		Parity Feature enable Nak.
{ErrMsg}	09h	00h	Reserved	Correctable Error Message.
		01h		Non-Fatal Error Message.
		02h		Fatal Error Message.
{Vendor Defined Message}	FFh	--	Vendor ID	Vendor Defined Messages. These can be exchanged at any time after sideband is functional post SBINIT. Interoperability is vendor defined. Unsupported vendor defined messages must be discarded by the receiver. Note that this is NOT the UCIe Vendor ID, but rather the unique identifier of the chiplet vendor that is defining and using these messages.
All other encodings not mentioned in this table are reserved.				

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-9. Link Training State Machine related Message encodings for messages without data (Sheet 1 of 4)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{Start Tx Init D to C point test resp}	0000h	8Ah	01h
{LFSR_clear_error req}	0000h	85h	02h
{LFSR_clear_error resp}	0000h	8Ah	02h
{Tx Init D to C results req}	0000h	85h	03h
{End Tx Init D to C point test req}	0000h	85h	04h
{End Tx Init D to C point test resp}	0000h	8Ah	04h
{Start Tx Init D to C eye sweep resp}	0000h	8Ah	05h
{End Tx Init D to C eye sweep req}	0000h	85h	06h
{End Tx Init D to C eye sweep resp}	0000h	8Ah	06h
{Start Rx Init D to C point test resp}	0000h	8Ah	07h
{Rx Init D to C Tx Count Done req}	0000h	85h	08h
{Rx Init D to C Tx Count Done resp}	0000h	8Ah	08h
{End Rx Init D to C point test req}	0000h	85h	09h
{End Rx Init D to C point test resp}	0000h	8Ah	09h
{Start Rx Init D to C eye sweep resp}	0000h	8Ah	0Ah
{Rx Init D to C results req}	0000h	85h	0Bh
{End Rx Init D to C eye sweep req}	0000h	85h	0Dh
{End Rx Init D to C eye sweep resp}	0000h	8Ah	0Dh
{SBINIT out of Reset}	[15:4] : Reserved [3:0] : Result[1]	91h	00h
{SBINIT done req}	0000h	95h	01h
{SBINIT done resp}	0000h	9Ah	01h
{MBINIT.CAL Done req}	0000h	A5h	02h
{MBINIT.CAL Done resp}	0000h	AAh	02h
{MBINIT.REPAIRCLK init req}	0000h	A5h	03h
{MBINIT.REPAIRCLK init resp}	0000h	AAh	03h
{MBINIT.REPAIRCLK result req}	0000h	A5h	04h
{MBINIT.REPAIRCLK apply repair resp}	0000h	AAh	05h
{MBINIT.REPAIRCLK check repair init req}	0000h	A5h	06h
{MBINIT.REPAIRCLK check repair init resp}	0000h	AAh	06h
{MBINIT.REPAIRCLK check results req}	0000h	A5h	07h
{MBINIT.RepairCLK done req}	0000h	A5h	08h
{MBINIT.RepairCLK done resp}	0000h	AAh	08h
{MBINIT.REPAIRVAL init req}	0000h	A5h	09h
{MBINIT.REPAIRVAL init resp}	0000h	AAh	09h
{MBINIT.REPAIRVAL result req}	0000h	A5h	0Ah
{MBINIT.REPAIRVAL apply repair resp}	0000h	AAh	0Bh
{MBINIT.RepairVAL done req}	0000h	A5h	0Ch
{MBINIT.RepairVAL done resp}	0000h	AAh	0Ch

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-9. Link Training State Machine related Message encodings for messages without data (Sheet 2 of 4)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBINIT.REVERSALMB init req}	0000h	A5h	0Dh
{MBINIT.REVERSALMB init resp}	0000h	AAh	0Dh
{MBINIT.REVERSALMB clear error req}	0000h	A5h	0Eh
{MBINIT.REVERSALMB clear error resp}	0000h	AAh	0Eh
{MBINIT.REVERSALMB result req}	0000h	A5h	0Fh
{MBINIT.ReversalMB done req}	0000h	A5h	10h
{MBINIT.RversalMB done resp}	0000h	AAh	10h
{MBINIT.REPAIRMB start req}	0000h	A5h	11h
{MBINIT.REPAIRMB start resp}	0000h	AAh	11h
{MBINIT.REPAIRMB Apply repair resp}	0000h	AAh	12h
{MBINIT.REPAIRMB end req}	0000h	A5h	13h
{MBINIT.REPAIRMB end resp}	0000h	AAh	13h
{MBINIT.REPAIRCLK result resp}	[15:4]: Reserved [3]: Compare Results from **RRDCK_L** [2]: Compare Results from **RTRK_L** [1]: Compare Results from **RCKN_L** [0]: Compare Results from **RCKP_L**	AAh	04h
{MBINIT.REPAIRCLK apply repair req}	[15:4] : Reserved [3:0] : Repair Encoding Fh : No Repair 0h : Repair **RCLKP_L** 1h : Repair **RCLKN_L** 2h : Repair **RTRK_L** 7h : Unrepairable	A5h	05h
{MBINIT.REPAIRCLK check results resp}	[15:4] : Reserved [3] : Compare Results from **RRDCK_L** [2] : Compare Results from **RTRK_L** [1] : Compare Results from **RCKN_L** [0] : Compare Results from **RCKP_L**	AAh	07h
{MBINIT.REPAIRVAL result resp}	[15:2] : Reserved [1] : Compare Results from **RRDVLD_L** [0] : Compare Results from **RVLD_L**	AAh	0Ah
{MBINIT.REPAIRVAL apply repair req}	[15:2] : Reserved [1:0] : Repair Encoding 3h : No Repair 0h : Repair **RVLD_L** 1h : Unrepairable	A5h	0Bh
{MBINIT.REPAIRMB apply degrade req}	[15:2]: Reserved [1:0]: Standard package logical Lane map	A5h	14h
{MBINIT.REPAIRMB apply degrade resp}	0000h	AAh	14h

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-9. Link Training State Machine related Message encodings for messages without data (Sheet 3 of 4)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBTRAIN.VALVREF start req}	0000h	B5h	00h
{MBTRAIN.VALVREF start resp}	0000h	BAh	00h
{MBTRAIN.VALVREF end req}	0000h	B5h	01h
{MBTRAIN.VALVREF end resp}	0000h	BAh	01h
{MBTRAIN.DATAVREF start req}	0000h	B5h	02h
{MBTRAIN.DATAVREF start resp}	0000h	BAh	02h
{MBTRAIN.DATAVREF end req}	0000h	B5h	03h
{MBTRAIN.DATAVREF end resp}	0000h	BAh	03h
{MBTRAIN.SPEEDIDLE done req}	0000h	B5h	04h
{MBTRAIN.SPEEDIDLE done resp}	0000h	BAh	04h
{MBTRAIN.TXSELFCAL Done req}	0000h	B5h	05h
{MBTRAIN.TXSELFCAL Done resp}	0000h	BAh	05h
{MBTRAIN.RXCLKCAL start req}	0000h	B5h	06h
{MBTRAIN.RXCLKCAL start resp}	0000h	BAh	06h
{MBTRAIN.RXCLKCAL done req}	0000h	B5h	07h
{MBTRAIN.RXCLKCAL done resp}	0000h	BAh	07h
{MBTRAIN.VALTRAINCENTER start req}	0000h	B5h	08h
{MBTRAIN.VALTRAINCENTER start resp}	0000h	BAh	08h
{MBTRAIN.VALTRAINCENTER done req}	0000h	B5h	09h
{MBTRAIN.VALTRAINCENTER done resp}	0000h	BAh	09h
{MBTRAIN.VALTRAINVREF start req}	0000h	B5h	0Ah
{MBTRAIN.VALTRAINVREF start resp}	0000h	BAh	0Ah
{MBTRAIN.VALTRAINVREF done req}	0000h	B5h	0Bh
{MBTRAIN.VALTRAINVREF done resp}	0000h	BAh	0Bh
{MBTRAIN.DATATRAINCENTER1 start req}	0000h	B5h	0Ch
{MBTRAIN.DATATRAINCENTER1 start resp}	0000h	BAh	0Ch
{MBTRAIN.DATATRAINCENTER1 end req}	0000h	B5h	0Dh
{MBTRAIN.DATATRAINCENTER1 end resp}	0000h	BAh	0Dh
{MBTRAIN.DATATRAINVREF start req}	0000h	B5h	0Eh
{MBTRAIN.DATATRAINVREF start resp}	0000h	BAh	0Eh
{MBTRAIN.DATATRAINVREF end req}	0000h	B5h	10h
{MBTRAIN.DATATRAINVREF end resp}	0000h	BAh	10h
{MBTRAIN.RXDESKEW start req}	0000h	B5h	11h
{MBTRAIN.RXDESKEW start resp}	0000h	BAh	11h
{MBTRAIN.RXDESKEW end req}	0000h	B5h	12h
{MBTRAIN.RXDESKEW end resp}	0000h	BAh	12h
{MBTRAIN.DATATRAINCENTER2 start req}	0000h	B5h	13h
{MBTRAIN.DATATRAINCENTER2 start resp}	0000h	BAh	13h
{MBTRAIN.DATATRAINCENTER2 end req}	0000h	B5h	14h
{MBTRAIN.DATATRAINCENTER2 end resp}	0000h	BAh	14h

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-9. Link Training State Machine related Message encodings for messages without data (Sheet 4 of 4)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBTRAIN.LINKSPEED start req}	0000h	B5h	15h
{MBTRAIN.LINKSPEED start resp}	0000h	BAh	15h
{MBTRAIN.LINKSPEED error req}	0000h	B5h	16h
{MBTRAIN.LINKSPEED error resp}	0000h	BAh	16h
{MBTRAIN.LINKSPEED exit to repair req}	0000h	B5h	17h
{MBTRAIN.LINKSPEED exit to repair resp}	0000h	BAh	17h
{MBTRAIN.LINKSPEED exit to speed degrade req}	0000h	B5h	18h
{MBTRAIN.LINKSPEED exit to speed degrade resp}	0000h	BAh	18h
{MBTRAIN.LINKSPEED done req}	0000h: for regular response FFFFh: for stall	B5h	19h
{MBTRAIN.LINKSPEED done resp}	0000h: for regular response FFFFh: for stall	BAh	19h
{MBTRAIN. module disable req}	0000h	B5h	1Ah
{MBTRAIN. module disable resp}	0000h	BAh	1Ah
{MBTRAIN.LINKSPEED exit to phy retrain req}	0000h	B5h	1Fh
{MBTRAIN.LINKSPEED exit to phy retrain resp}	0000h	BAh	1Fh
{MBTRAIN.REPAIR init req}	0000h	B5h	1Bh
{MBTRAIN.REPAIR init resp}	0000h	BAh	1Bh
{MBTRAIN.REPAIR Apply repair resp}	0000h	BAh	1Ch
{MBTRAIN.REPAIR end req}	0000h	B5h	1Dh
{MBTRAIN.REPAIR end resp}	0000h	BAh	1Dh
{MBTRAIN.REPAIR Apply degrade req}	[15:2]: Reserved [1:0]: Standard package logical Lane map[2]	B5h	1Eh
{MBTRAIN.REPAIR Apply degrade resp}	0000h	BAh	1Eh
{PHYRETRAIN.retrain init req}	[15:3]: Reserved [2:0]: Retrain Encoding	C5h	00h
{PHYRETRAIN.retrain init resp}	[15:3]: Reserved [2:0]: Retrain Encoding	CAh	00h
{PHYRETRAIN.retrain start req}	[15:3]: Reserved [2:0]: Retrain Encoding	C5h	01h
{PHYRETRAIN.retrain start resp}	[15:3]: Reserved [2:0]: Retrain Encoding	CAh	01h
{RECAL.track pattern init req}	0000h	D5h	00h
{RECAL.track pattern init resp}	0000h	DAh	00h
{RECAL.track pattern done req}	0000h	D5h	01h
{RECAL.track pattern done resp}	0000h	DAh	01h
{TRAINERROR Entry req}	0000h	E5h	00h
{TRAINERROR Entry resp}	0000h	EAh	00h

1. See Section 4.5.3.2
2. See Table 4-9

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.1.2.3 Messages with data payloads

Figure 6-4 shows the formats for Messages with data payloads. The definitions of opcode, srcid, dstid, dp, and cp fields are the same as Register Access packets.

Figure 6-4. Format for Messages with data payloads

Messages with data																															
Bytes	3								2								1								0							
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header / Data	Header																															
Phase0	srcid			rsvd		rsvd			msgcode[7:0]								rsvd								opcode[4:0]							
Phase1	dp	cp		rsvd		dstid			MsgInfo[15:0]																MsgSubcode[7:0]							
Header / Data	Data																															
Phase2	data[31:0]																															
Phase3	data[63:32]																															

Table 6-10 and Table 6-11 give the message encodings.

Table 6-10. Message encodings for Messages with Data (Sheet 1 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{AdvCap.Adapter}	01h	00h	0000h: Regular Message FFFFh: Stall Message	[0]: "Raw Format" [1]: "68B Flit Mode" [2]: "CXL 256B Flit Mode" [3]: "PCIe Flit Mode" [4]: "Streaming" [5]: "Retry" [6]: "Multi_Protocol_Enable" [7]: "Stack0_Enable" [8]: "Stack1_Enable" [9]: "CXL_LatOpt_Fmt5" [10]: "CXL_LatOpt_Fmt6" [11]: "Retimer" [20:12]: "Retimer Credits" [21]: "DP" [22]: "UP" [23]: "68B Flit Format" [24]: "Standard 256B End Header Flit Format" [25]: "Standard 256B Start Header Flit Format" [26]: "Latency-Optimized 256B without Optional Bytes Flit Format" [27]: "Latency-Optimized 256B with Optional Bytes Flit Format" [28]: "Enhanced_Multi_Protocol_Enable" [29]: "Stack 0 Maximum Bandwidth_Limit" [30]: "Stack 1 Maximum Bandwidth_Limit" [63:31]: Reserved	Advertised Capabilities of the D2D Adapter
{FinCap.Adapter}	02h	00h	0000h: Regular Message FFFFh: Stall Message		Finalized Capability of the D2D Adapter

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-10. Message encodings for Messages with Data (Sheet 2 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{AdvCap.CXL}	01h	01h	0000h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 0b, or it is 1b and the message is for Stack 0. 0001h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 1b and the message is for Stack 1. FFFFh: Stall Message	[23:0] : Flexbus Mode negotiation usage bits as defined for Symbols 12-14 of Modified TS1/TS2 Ordered Set in *CXL Specification*, with the following additional rules: • [0]: PCIe capable/enable - this must be 1b for PCIe Non-Flit Mode. • [1]: CXL.io capable/enable - this must be 0b for PCIe Non-Flit Mode. • [2]: CXL.mem capable/enable - this must be 0b for PCIe Non-Flit Mode. • [3]: CXL.cache capable/enable - this must be 0b for PCIe Non-Flit Mode. • [4]: CXL 68B Flit and VH capable; must be set for ports that support CXL protocols, as specified in the Protocol Layer interoperability requirements. • [8]: Multi-Logical Device - must be set to 0b for PCIe Non-Flit Mode. • [9]: Reserved. • [12:10]: these bits do not apply for UCIe, must be 0b. • [14]: Retimer 2 - does not apply for UCIe, must be 0b. • [15]: CXL.io Throttle - must be 0b for PCIe Non-Flit Mode. • [17:16]: NOP Hint Info - does not apply for UCIe, and must be 0.	Advertised Capabilities for CXL protocol.
{FinCap.CXL}	02h	01h	0000h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 0b, or it is 1b and the message is for Stack 0. 0001h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 1b and the message is for Stack 1. FFFFh: Stall Message		Finalized Capabilities for CXL protocol.
{MultiProtAdvCap.Adapter}	01h	02h	0000h: Reserved FFFFh: Stall Message	[0]: "68B Flit Mode" [1]: "CXL 256B Flit Mode" [2]: "PCIe Flit Mode" [3]: "Streaming" [63:4]: Reserved	Protocol Advertisement for Stack 1 when Enhanced Multi_Protocol_Enable is negotiated
{MultiProtFinCap.Adapter}	02h	02h	0000h: Reserved FFFFh: Stall Message	[0]: "68B Flit Mode" [1]: "CXL 256B Flit Mode" [2]: "PCIe Flit Mode" [63:3]: Reserved	Finalized Capability for Protocol negotiation when Enhanced Multi_Protocol_Enable is negotiated and Stack 1 is PCIe or CXL

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-10. Message encodings for Messages with Data (Sheet 3 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{Vendor Defined Message}	FFh	--	Vendor ID		Vendor Defined Messages. These can be exchanged at any time after sideband is functional post SBINIT. Interoperability is vendor defined. Unsupported vendor defined messages must be discarded by the receiver. Note that this is NOT the UCIe Vendor ID, but rather the unique identifier of the chiplet vendor that is defining and using these messages.
All other encodings not mentioned in this table are reserved.					

Table 6-11. Link Training State Machine related encodings (Sheet 1 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{Start Tx Init D to C point test req}	[15:0]: Maximum comparison error threshold	85h	01h	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43]: Iteration Count Settings [42:27]: Idle Count settings [26:11]: Burst Count settings [10]: Pattern Mode (0: continuous mode, 1: Burst Mode) [9:6] : Clock Phase control at Tx Device (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3] : Valid Pattern (0h: Functional pattern) [2:0]: Data pattern (0h: LFSR, 1h: Per Lane ID)

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-11. Link Training State Machine related encodings (Sheet 2 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{Tx Init D to C results resp}	[15:6]: Reserved [5]: Valid Lane comparison results [4]: Cumulative Results of all Lanes (0: Fail (Errors > Max Error Threshold), 1: Pass (Errors <= Max Error Threshold)). [3:0]: UCIe-A: Compare results from Redundant Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) (RRD_L[3], RRD_L[2], RRD_L[1], RRD_L[0]) UCIe-S: Reserved RRD_L[3] and RRD_L[2] are reserved for UCIe-A x32 as a transmitter of this message.	8Ah	03h	[63:0]: Compare Results of individual Data Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) UCIe-A {RD_L[63], RD_L[62], …., RD_L[1], RD_L[0]} UCIe-S {48'h0, RD_L[15], RD_L[14], …, RD_L[1], RD_L[0]} UCIe-A x32 {32'h0, RD_L[31], RD_L[30], …, RD_L[0]}
{Start Tx Init D to C eye sweep req}	[15:0]: Maximum comparison error threshold	85h	05h	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43] : Iteration Count Settings [42:27] : Idle Count settings [26:11] : Burst Count settings [10] : Pattern Mode (0 : continuous mode, 1 : Burst Mode) [9:6] : Clock Phase control at Tx Device (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3] : Valid Pattern (0h : Functional pattern) [2:0] : Data pattern (0h : LFSR, 1h : Per Lane ID)
{Start Rx Init D to C point test req}	[15:0]: Maximum comparison error threshold	85h	07h	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43] : Iteration Count Settings [42:27] : Idle Count settings [26:11] : Burst Count settings [10] : Pattern Mode (0 : continuous mode, 1 : Burst Mode) [9:6] : Clock Phase control at Transmitter (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3] : Valid Pattern (0h : Functional pattern) [2:0] : Data pattern (0h : LFSR, 1h : Per Lane ID)

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-11. Link Training State Machine related encodings (Sheet 3 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{Start Rx Init D to C eye sweep req}	[15:0]: Maximum comparison error threshold	85h	0Ah	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43] : Iteration Count Settings [42:27] : Idle Count settings [26:11] : Burst Count settings [10] : Pattern Mode (0 : continuous mode, 1 : Burst Mode) [9:6] : Clock Phase control at Transmitter (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3] : Valid Pattern (0h : Functional pattern) [2:0] : Data pattern (0h : LFSR, 1h : Per Lane ID)
{Rx Init D to C results resp}	[15:6]: Reserved [5]: Valid Lane comparison result [4]: Cumulative Results of all Lanes (0: Fail (Errors > Max Error Threshold), 1: Pass (Errors <= Max Error Threshold)). [3:0]: UCIe-A: Compare results from Redundant Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) (RRD_L[3], RRD_L[2], RRD_L[1], RRD_L[0]) UCIe-S: Reserved RRD_L[3] and RRD_L[2] are reserved for UCIe-A x32 as a transmitter of this message.	8Ah	0Bh	[63:0]: Compare Results of individual Data Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) UCIe-A {RD_L[63], RD_L[62], …., RD_L[1], RD_L[0]} UCIe-S {48'h0, RD_L[15], RD_L[14], …, RD_L[1], RD_L[0]} UCIe-A x32 {32'h0, RD_L[31], RD_L[30], …, RD_L[0]}
{Rx Init D to C sweep done with results}	0000h	81h	0Ch	[63:16]: Reserved [15:8]: Right Edge [7:0]: Left Edge
{MBINIT.PARAM Configuration req}	0000h	A5h	00h	[63:14]: Reserved [13]: UCIe-A x32 [12:11]: Module ID: 0h: 0, 1h: 1, 2h: 2, 3h:3 [10]: Clock Phase: 0b: Differential clock, 1b: Quadrature phase [9]: Clock Mode - 0b: Strobe mode; 1b: Continuous mode [8:4]: Voltage Swing - The encodings are the same as the "Supported Vswing encodings" field of the PHY Capability register [3:0]: Max IO Link Speed - The encodings are the same as "Max Link Speeds" field of the UCIe Link Capability register

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-11. Link Training State Machine related encodings (Sheet 4 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{MBINIT.PARAM configuration resp}	0000h	AAh	00h	[63:11]: Reserved [10]: Clock Phase: 0b: Differential clock, 1b: Quadrature phase [9]: Clock Mode - 0b: Strobe mode; 1b: Continuous mode [8:4]: Reserved [3:0]: Max IO Link Speed - The encodings are the same as "Max Link Speeds" field of the UCIe Link Capability register
{MBINIT.REVERSALMB result resp}	[15:6]: Reserved [5]: Valid Lane comparison result [4]: Cumulative Results of all Lanes (reserved for this message) [3:0]: UCIe-A: Compare results from Redundant Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) (RRD_L[3], RRD_L[2], RRD_L[1], RRD_L[0]) UCIe-S: Reserved RRD_L[3] and RRD_L[2] are reserved for UCIe-A x32 as a transmitter of this message.	AAh	0Fh	[63:0]: Compare Results of individual Data Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) UCIe-A {RD_L[63], RD_L[62], …., RD_L[1], RD_L[0]} UCIe-S {48'h0, RD_L[15], RD_L[14], …, RD_L[1], RD_L[0]} UCIe-A x32 {32'h0, RD_L[31], RD_L[30], …, RD_L[0]}

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-11. Link Training State Machine related encodings (Sheet 5 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{MBINIT.REPAIRMB Apply repair req}	0000h	A5h	12h	[31:24] : **Repair Address for TRD_P[3]**: Indicates the physical Lane repaired when TRD_P[3] is used in remapping scheme. This is reserved for UCIe-A x32 as a transmitter of this message. 20h: Invalid 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired …… 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Unrepairable FFh: No Repair [23:16]: **Repair Address for TRD_P[2]**: Indicates the physical Lane repaired when TRD_P[2] is used in remapping scheme. This is reserved for UCIe-A x32 as a transmitter of this message. 20h: TD_P[32] Repaired 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired …… 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Unrepairable FFh: No Repair [15:8]: **Repair Address for TRD_P[1]**: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme. 00h: Invalid 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired …… 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Unrepairable FFh: No Repair [7:0]: **Repair Address for TRD_P[0]**: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme. 00h: TD_P[0] Repaired 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired …… 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Unrepairable FFh: No Repair

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 6-11. Link Training State Machine related encodings (Sheet 6 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{MBTRAIN.REPAIR Apply repair req}	0000h	BAh	1Ch	[31:24] : **Repair Address for TRD_P[3]**: Indicates the physical Lane repaired when TRD_P[3] is used in remapping scheme. This is reserved for UCIe-A x32 as a transmitter of this message. 20h: Invalid 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired …… 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Unrepairable FFh: No Repair [23:16]: **Repair Address for TRD_P[2]**: Indicates the physical Lane repaired when TRD_P[2] is used in remapping scheme. This is reserved for UCIe-A x32 as a transmitter of this message. 20h: TD_P[32] Repaired 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired …… 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Unrepairable FFh: No Repair [15:8]: **Repair Address for TRD_P[1]**: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme. 00h: Invalid 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired …… 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Unrepairable FFh: No Repair [7:0]: **Repair Address for TRD_P[0]**: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme. 00h: TD_P[0] Repaired 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired …… 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Unrepairable FFh: No Repair

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.1.3 Flow Control and Data Integrity

Sideband packets can be transferred across FDI, RDI or the UCIe sideband Link. Each of these have independent flow control.

6.1.3.1 Flow Control and Data Integrity over FDI and RDI

For each Transmitter associated with FDI or RDI, a design time parameter of the interface is used to determine the number of credits advertised by the Receiver, with a maximum of 32 credits. Each credit corresponds to 64 bits of header and 64 bits of potentially associated data. Thus, there is only one type of credit for all sideband packets, regardless of how much data they carry. Every Transmitter/Receiver pair has an independent credit loop. For example, on RDI, credits are advertised from Physical Layer to Adapter for sideband packets transmitted from the Adapter to the Physical Layer; and credits are also advertised from Adapter to the Physical Layer for sideband packets transmitted from the Physical Layer to the Adapter.

The Transmitter must check for available credits before sending Register Access requests and Messages. The Transmitter must not check for credits before sending Register Access Completions, and the Receiver must guarantee unconditional sinking for any Register Access Completion packets. Messages carrying requests or responses consume a credit on FDI and RDI, but they must be guaranteed to make forward progress by the Receiver and not get blocked behind Register Access requests. Both RDI and FDI give a dedicated signal for sideband credit returns across those interfaces.

All Receivers associated with RDI and FDI must check received messages for data or control parity errors, and these errors must be mapped to Uncorrectable Internal Errors (UIE) and transition RDI to LinkError state.

6.1.3.2 Flow Control and Data Integrity over UCIe sideband Link between dies

The BER of the sideband Link is 1e-27 or better. Hence, no retry mechanism is provided for the sideband packets. Receivers of sideband packets must check for Data or Control parity errors, and any of these errors is mapped to a fatal UIE.

6.1.3.3 End-to-End flow control and forward progress for UCIe Link sideband

It is important for deadlock avoidance to ensure that there is sufficient space at the Receiver to sink all possible outstanding requests from the Transmitter, so that the requests do not get blocked at any intermediate buffers that would thereby prevent subsequent completions from making progress.

Sideband access for Remote Link partner's Adapter or Physical Layer registers is only accessible via the indirect mailbox mechanism, and the number of outstanding transactions is limited to four at a time. Although four credits are provisioned, there is only a single mailbox register, and this limits the number of outstanding requests that can use this mechanism to one at a time. The extra credits allow additional debug-related register access requests in case of register access timeouts. These credits are separate from local FDI or RDI accesses, and thus the Physical Layer must provision for sinking at least one register access request and completion each from remote die and local Adapter in addition to other sideband request credits (see Implementation Note below). The Adapter provisions for at least four remote register access requests from remote die Adapter. Each credit corresponds to 64b of header and 64b of data. Even requests that send no data or only send 32b of data consume one credit. Register Access completions do not consume a credit and must always sink. The Adapter credit counters for register access request transmission are initialized to 4 on domain reset exit. It is permitted to send an extra (N-4) credit returns to remote Link partner if a UCIe implementation is capable of sinking a total of N requests once RDI has transitioned to Active state. The Adapter must

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

implement a saturating credit counter capable of accumulating at least 4 credits, and hence prevent excess credit returns from overflowing the counter.

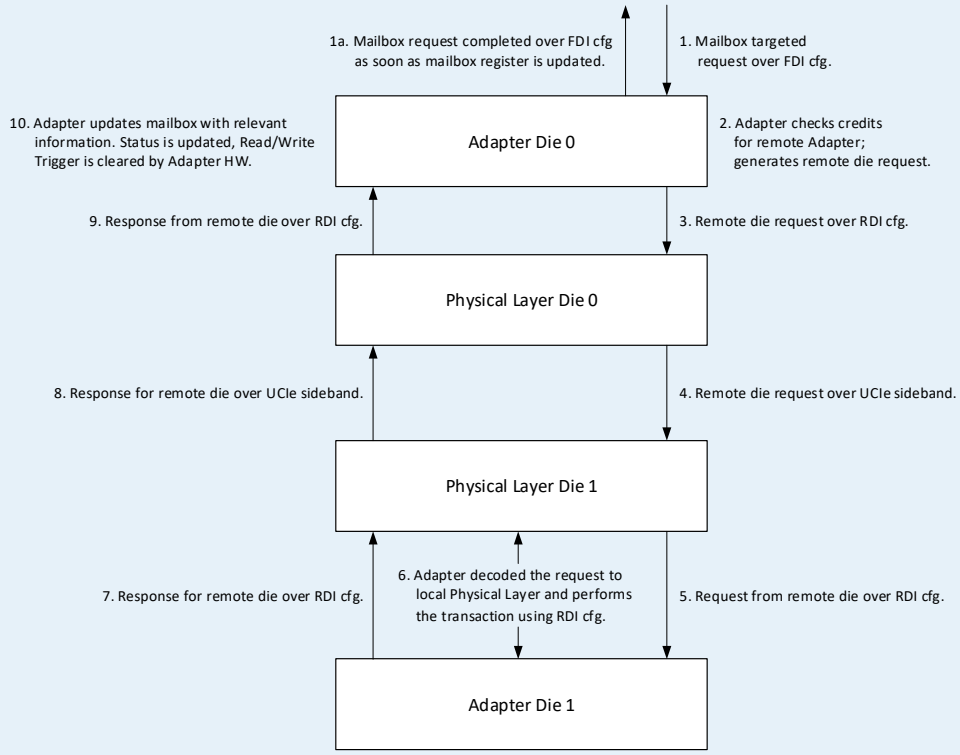
All other messages except Vendor Defined messages must always sink and make forward progress, and not block any messages on the sideband interface behind them. All Link Management message requests have an associated response, and the source of these messages must only have one outstanding request at a time (i.e., one outstanding message per "Link Management Request" MsgCode encoding).

For vendor defined messages, there must be a vendor defined cap on the number of outstanding messages, and the Receiver must guarantee sufficient space so as to not block any messages behind the vendor defined messages on any of the interfaces.

IMPLEMENTATION NOTE

Figure 6-5 shows an example of an end-to-end register access request to remote die and the corresponding completion returning back.

Figure 6-5. Example Flow for Remote Register Access Request (Local FDI/RDI Credit Checks Are Not Explicitly Shown)



In Step 1 shown in Figure 6-5, the Protocol Layer checks for FDI credits before sending the request to Adapter Die 0. Adapter Die 0 completes the mailbox request as soon as the mailbox register is updated (shown in Step 1a). FDI credits are returned once its internal buffer space is free. In Step 2, Adapter Die 0 checks credits for remote Adapter as well as credits for local RDI before sending the remote die request to Physical Layer Die 0 in Step 3. Physical Layer schedules the request over UCIe sideband and returns the RDI credit to Adapter Die 0 once it has freed up its internal buffer space.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

IMPLEMENTATION NOTE

Continued

In Step 5, Physical Layer Die 1 checks for Adapter Die 1 credits on RDI before sending the request over RDI. Adapter Die 1 decodes the request to see that it must access a register on Physical Layer Die 1; Adapter Die 1 checks for RDI credits of Physical Layer Die 1 before sending the request over RDI in Step 6. Adapter Die 1 must remap the tag for this request, if required, and save off the original tag of the remote die request as well as pre-allocate a space for the completion. Physical Layer Die 1 completes the register access request and responds with the corresponding completion. Because a completion is sent over RDI, no RDI credits need to be checked or consumed. Adapter Die 1 generates the completion for the remote die request and sends it over RDI (no credits are checked or consumed for completion over RDI) in Step 7. The completion is transferred across the different hops as shown in Figure 6-5 and finally sunk in Adapter Die 0 to update the mailbox information. No RDI credits need to be checked for completions at the different hops.

For forward progress to occur, the Adapters and Physical Layers on both die must ensure that they can sink sufficient requests, completions, and messages to guarantee that there is no Link Layer level dependency between the different types of sideband packets (i.e., remote register access requests, remote register access completions, Link state transition messages for Adapter LSM(s), Link state transition messages for RDI, and Link Training related messages). In all cases, because at most one or two outstanding messages are allowed for each operation, it is relatively easy to provide greater than or the same number of buffers to sink from RDI. For example, in the scenario shown in Figure 6-5, Physical Layer Die 1 must ensure that it has dedicated space to sink the request in Step 6 independent of any ongoing remote register access request or completion from Die 1 to Die 0, or any other sideband message for state transition, etc. Similarly, Physical Layer Die 1 must have dedicated space for remote die register access completion in Step 7.

Dynamic coarse clock gating is permitted in Adapter or Physical Layer in a subset of the RDI states (see Chapter 8.0). Thus, when applicable, any sideband transfer over RDI or FDI must follow the clock gating exit handshake rules as defined in Chapter 8.0. It is recommended to always perform the clock exit gating handshakes for sideband transfers if implementations need to decouple dependencies between the interface status and sideband transfers.

Implementations of the Physical Layer and Adapter must ensure that there is no receiver buffer overflow for messages being sent over the UCIe sideband Link. This can be done by either ensuring that the time to exit clock gating is upper bounded and less than the time to transmit a sideband packet over the UCIe sideband Link, OR that the Physical Layer has sufficient storage to account for the worst-case backup of each sideband message function (i.e., remote register access requests, remote register access completions, Link state transition messages for Adapter LSM(s), Link state transition messages for RDI, and Link Training related messages). The latter offers more-general interoperability at the cost of buffers.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

6.1.4 Operation on RDI and FDI

The same formats and rules of operation are followed on the RDI and FDI. The protocol is symmetric — requests, completions, and messages can be sent on **lp_cfg** as well as on **pl_cfg** signals. Implementations must ensure deadlock-free operation by allowing a sufficient quantity of sideband packets to sink and unblock the sideband bus for other packets. At the interface, these transactions are packetized into multiple phases depending on the configuration interface width (compile time parameter). Supported interface widths are 8, 16, or 32 bits. **lp_cfg_vld** and **pl_cfg_vld** are asserted independently for each phase. They must be asserted on consecutive clock cycles for transferring consecutive phases of the same packet. They may or may not assert on consecutive clock cycles when transferring phases of different packets. For packets with data, 64b of data is always transmitted over RDI or FDI; for 32b of valid payload, the most-significant 32b (Phase 4) of the packet are assigned to 0b before transfer.

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.0 Configuration and Parameters

7.1 High level Software view of UCIe

A key goal of UCIe is to leverage all the software investments made for PCIe and CXL while still defining the interface in an extensible way for future innovative solutions. To that end, UCIe SW view of the protocol layer is consistent with the associated protocol. For example, the host Downstream Port for UCIe that is capable of supporting CXL protocols will appear to software as a Root Port with CXL DVSEC capability and relevant PCIe capabilities. Similarly, a host downstream port for UCIe that is capable of supporting PCIe protocol only, will appear to software as a Root Port with relevant PCIe capabilities only. Host side or device side view of software for Streaming protocol is implementation-specific since the protocol itself is implementation-specific. It is though strongly recommended that ecosystem implementations define streaming solutions leveraging the SW hooks already in place for supporting CXL and PCIe. The Upstream Ports that connect to a UCIe Root Port can be a PCIe endpoint, PCIe Switch, a CXL endpoint-device, or a CXL Switch. This allows for UCIe solution to be fully backward compatible to pre-UCIe software. The remainder of this chapter talks about SW view of UCIe when paired with PCIe or CXL protocol layers.

UCIe specification allows for a single UCIe Link to be shared by multiple protocol stacks. In this version of the spec, this sharing is limited to at most 2 protocol stacks. Shared Link layer is a new concept from Software perspective and requires new discovery/control mechanisms. The mechanism by which UCIe-aware SW discovers UCIe capability is described in the next section.

Table 7-1 shows the legal/illegal combinations of Upstream and Downstream devices/ports at a given UCIe interface, from a SW viewpoint.

Table 7-1. Software view of Upstream and Downstream Device at UCIe interface

Downstream Component: SW View	Upstream Component: SW View			
	PCIe RP, PCIe Switch DSP[1]	CXL-RP, CXL Switch DSP[2]	CXL Downstream Port RCRB[3]	Streaming Device
PCIe EP, PCIe Switch USP	Valid	Valid	Illegal	Vendor defined
CXL Upstream Port RCRB[4]	Illegal	Illegal	Illegal	
CXL EP	Valid	Valid	Illegal	
Streaming Device	Vendor defined			

1. PCIe RP = As defined in *PCIe Base Specification*
2. CXL RP/Switch DSP = Standard PCIe RP/Switch-DSP with additional CXL Flexbus Port DVSEC capability
3. CXL Downstream Port RCRB = CXL Link at host or at Switch DSP that is enumerated via CXL defined Downstream Port RCRB (instead of via a Root Port)
4. CXL Upstream Port RCRB = CXL upstream port that is enumerated via CXL defined RCRB with CXL Upstream Port RCRB and that has a RCiEP below it.

All the CXL/PCIe legacy/advanced capabilities/registers defined in the respective specifications apply to UCIe host and devices as well. Some Link and PHY layer specific registers in *PCIe Base Specification* do not apply in UCIe context and these are listed in the appendix. In addition, two new

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

DVSEC capabilities and four other MMIO mapped register blocks are defined to deal with UCIe-specific Adapter and Physical Layer capabilities.

7.2 SW Discovery of UCIe Links

UCIe-aware Firmware/Software may discover the presence and capabilities of UCIe Links in the system per Table 7-2.

Table 7-2. SW discovery of UCIe Links

UCIe Links	How discovered?	Salient Points
In Host	Host specific Register Block called UiRB, containing UCIe Link DVSEC Capability	• UiRB is at a host defined static location. • Each UCIe Link has a separate UiRB Base address and these are enumerated to OS via UCIe Early discovery table (UEDT)[1] • Association of a UCIe Link to 1 or more Root ports is described in UEDT, allowing for UCIe-aware SW to understand the potential shared nature of the UCIe Link.
In Endpoints	Dev0/Fn0 of the device carries a UCIe Link DVSEC Capability.	• In multi-stack implementations, Dev0/Fn0 of the endpoint in only one of the stacks carries the UCIe Link DVSEC Capability.
In Switch USP	Dev0/Fn0 of the USP carrying a UCIe Link DVSEC Capability	• In multi-stack implementations, Dev0/Fn0 of the USP in only one of the stacks carries the UCIe Link DVSEC Capability.
In Switch DSP	Dev0/Fn0 of the Switch USP carrying one ore more UiSRB DVSEC Capability	• UCIe Links below the switch are described in UiSRB whose base address is provided in the UiSRB DVSEC Capability • A UCIe Link DVSEC capability per downstream UCIe Link is present in the UiSRB • Association of a UCIe Link to 1 or more Switch DSPs is described as part of the UCIe Link DVSEC Capability, allowing for UCIe-aware SW to understand the potential shared nature of the UCIe interface **Note:** It is legal for a Switch USP to carry the UiSRB DVSEC capability but not a UCIe Link DVSEC Capability

1. UEDT structure is standardized as part of the ACPI specification.

7.3 Register Location Details and Access Mechanism

- 2 UCIe DVSEC capabilities (UCIe Link DVSEC, UiSRB DVSEC) and four other MMIO-mapped register blocks are defined in this version of the Specification.
- UCIe Link DVSEC capability is located in UiRB for host root ports and in UiSRB for Switch downstream ports.
- UiRB region is defined at a static location on the host side and its size is enumerated in the UEDT structure. Only UCIe Link related registers are permitted in this region and designs must not implement non-UCIe related functionality in this region.
- There is a unique UiRB base address for each UCIe Link, in the host
- UiSRB region base address is provided in the UiSRB DVSEC capability. This region is part of a BAR region of Switch Dev0/Fn0 USP.
- For scalability/flexibility reasons, multiple UiSRB DVSEC capabilities can exist in a Switch USP function. In case of multiple UiSRB DVSEC capabilities in the USP. a given DSP UCIe Link can only be described in one of the UiSRB structures.
- Configuration space registers are accessed using configuration reads and configuration writes. Register Blocks are in memory mapped regions and are accessed using standard memory reads and memory writes.
- UCIe Retimer registers are not directly accessible from host SW. They can be accessed only via a window mechanism over the sideband interface (hence the terms *SB-MMIO* and *SB-Config* in

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-3). The window mechanism is available via RP/DSP UCIe Link DVSEC Capability to access the UCIe Retimer registers on the Retimer closest to the host. For accessing UCIe Retimer registers on the far end Retimer, the same window mechanism is also available in the UCIe Link DVSEC capability of EP/USP. See Section 7.5.1.11 and Section 7.5.1.12 for details of the window mechanism.

- For debug and runtime Link health monitoring reasons, host SW can also access the UCIe related registers in any partner die on the sideband interface, using the same window mechanism. For brevity purposes, that is not shown in Table 7-3. Note that register accesses over sideband are limited to only the UCIe-related Capability registers (the two DVSECs currently defined in the spec) and the four defined UCIe Register Blocks. Nothing else on the remote die are accessible via the sideband mechanism.

Table 7-3 summarizes the location of various register blocks in each native UCIe port/device. Henceforth a "UCIe port/device/EP/Switch" is used to refer to a standard PCIe or CXL port/device/EP/Switch with UCIe Link DVSEC Capability.

Table 7-3. Summary of location of various UCIe Link related registers

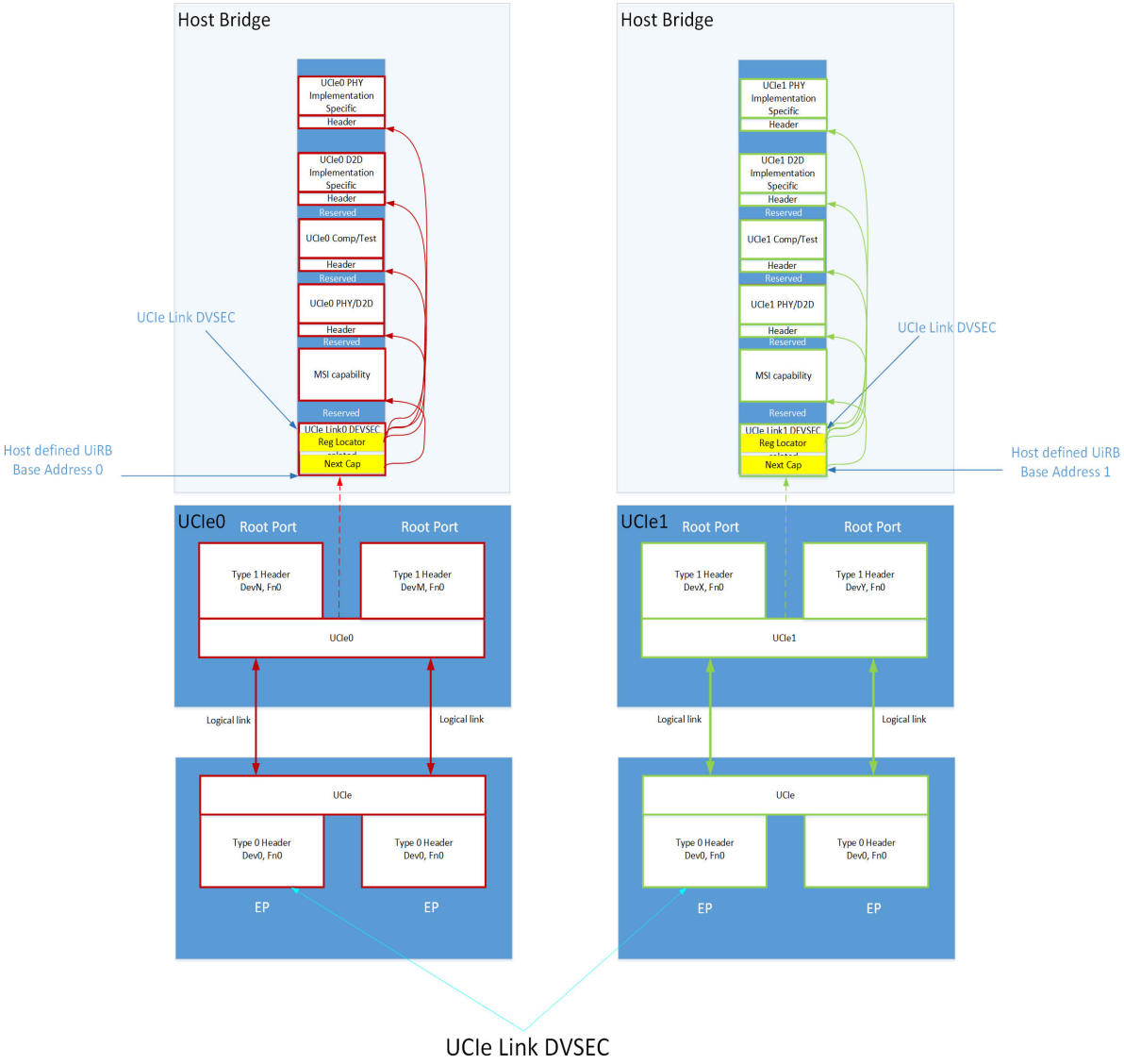
Register	Where the Register Resides					Comments
	RP	Switch USP	Switch DSP	EP	UCIe Retimer	
UCIe Link DVSEC	UiRB	Config space	UiSRB	Config Space	Sideband Config Space	Registers that define the basic UCIe interface related details
UCIe D2D/PHY Register Block	UiRB	Switch USP-BAR Region	UiSRB	EP-BAR Region	SB-MMIO Space	Registers that define lower-level functionality for the D2D/PHY interface of a typical UCIe implementation
UCIe Test/ Compliance Register Block	UiRB	Switch USP-BAR Region	UiSRB	EP-BAR Region	SB-MMIO Space	Registers for Test/Compliance of UCIe interface
UCIe Implementation Specific Register Block	UiRB	Switch USP-BAR Region	UiSRB	EP-BAR Region	SB-MMIO Space	Registers for vendor specific implementation

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.4 Software view Examples

Figure 7-1 summarizes all the details of UCIe related DVSEC Capabilities and SW discovery, for an implementation consisting of Root Ports and Endpoints. This example has a host with 2 UCIe downstream Links that each carry traffic from 2 Root Ports.

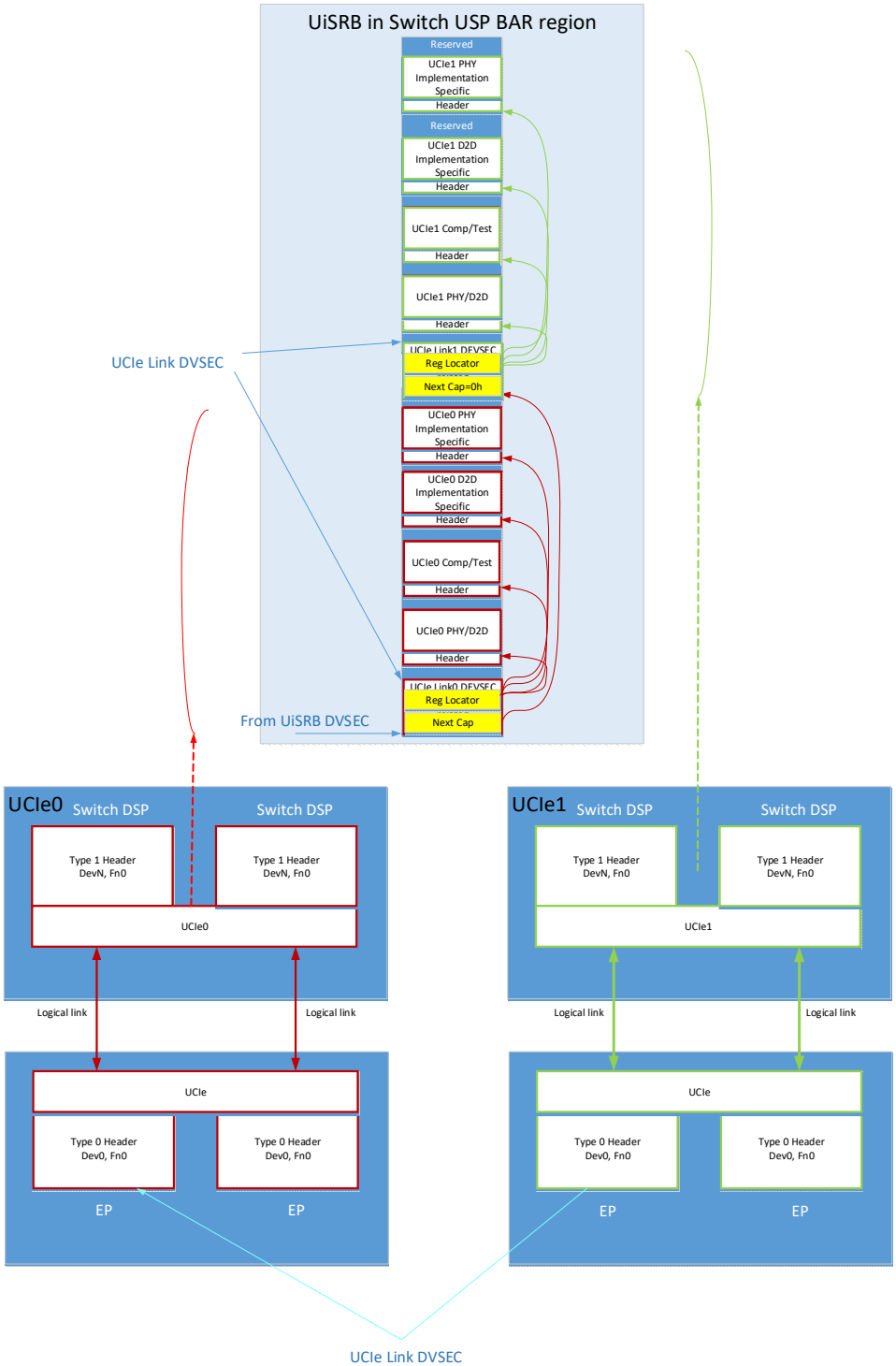
Figure 7-1. Software view Example with Root Ports and Endpoints



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Example in Figure 7-2 has a Switch with 2 UCIe Links on its downstream side and each UCIe Link carries traffic from 2 Switch DSPs.

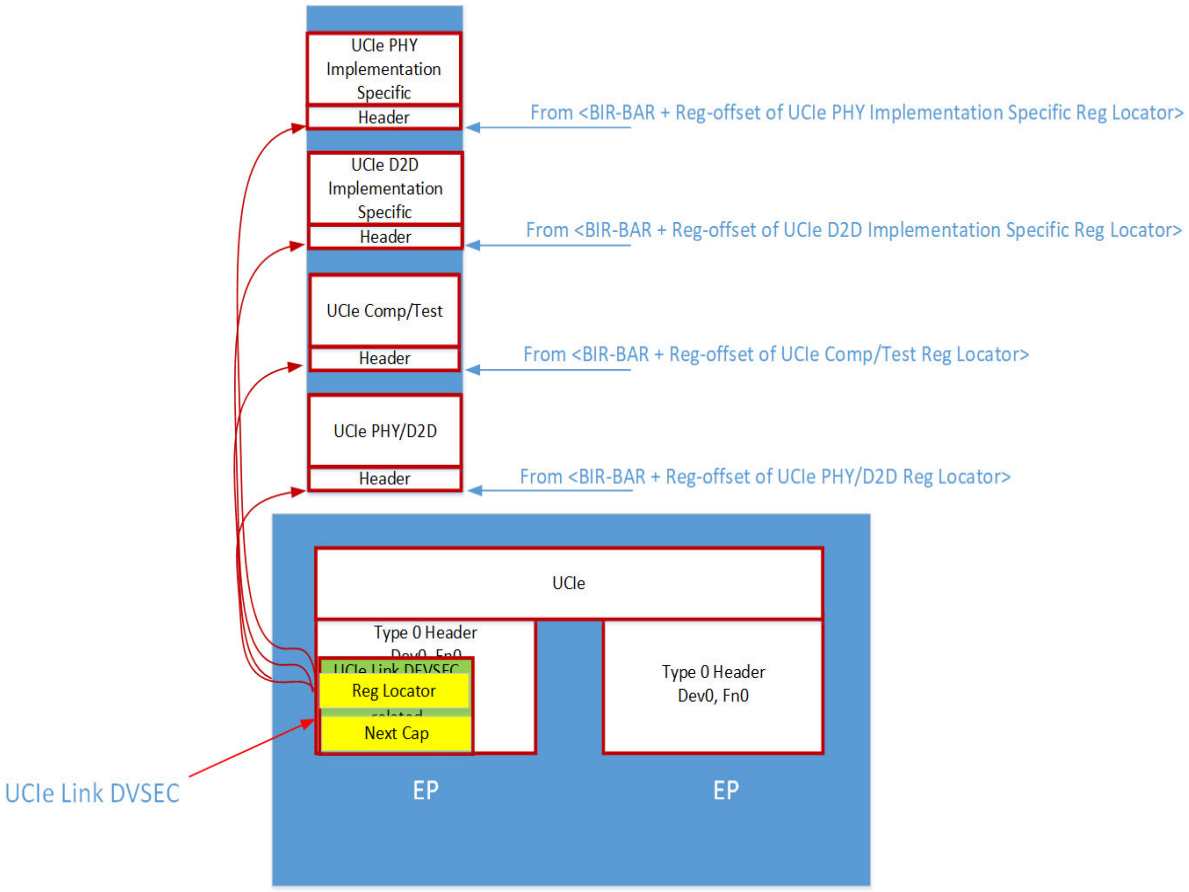
Figure 7-2. Software view Example with Switch and Endpoints



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Example in Figure 7-3 shows details UCIe registers in an implementation where two EPs are sharing a common UCIe Link.

Figure 7-3. Software view Example of UCIe Endpoint



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5 UCIe Registers

Table 7-4 summarizes the attributes for the register bits defined in this chapter. Unless otherwise specified, the definition of these attributes is consistent with *PCIe Base Specification* and *CXL Specification*.

Table 7-4. Register Attributes

Attribute	Description
RO	Read Only.
ROS	Read Only Sticky
RW	Read-Write
RWS	Read Write Sticky
RW1C	Read-Write-One-To-Clear.
RW1CS	Read-Write-One-To-Clear-Sticky. Not affected by hot reset of the Link. Otherwise, the behavior follows PCIe Base Specification.
HWInit	Hardware Initialized[1]
RsvdP	Reserved and Preserved
RsvdZ	Reserved and Zero

1. Typically, this register attribute is used for functionality/capability that can vary with package integration. For example, a chiplet that is capable of 32 GT/s maximum speed might be routed to achieve a maximum speed of 16 GT/s in a given package implementation. To account for such scenarios, the Max link speed field in the UCIe Link Capability register has the HWInit attribute and its value could be configured by a package-level strap or device/system firmware to reflect the maximum speed of that implementation.

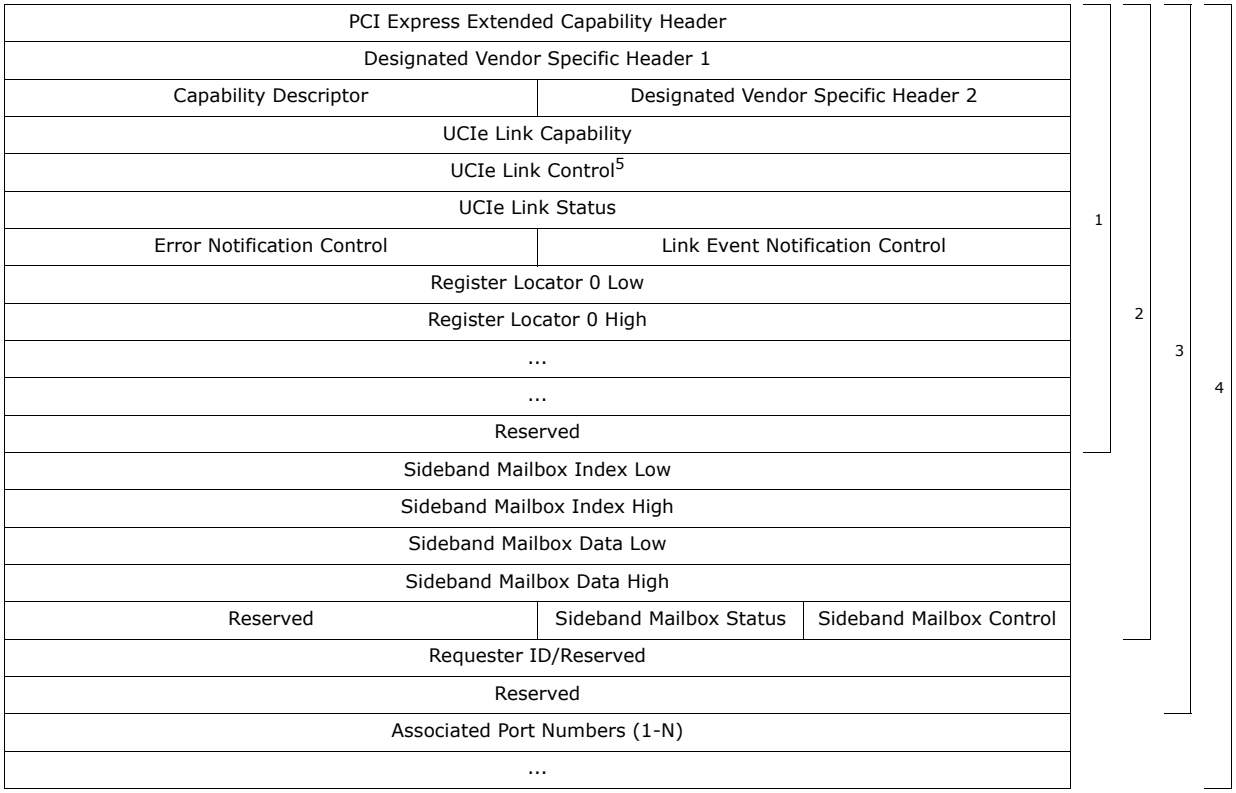
All numeric values in various data structures, individual registers and register fields defined in this chapter are always encoded in little endian format, unless stated otherwise.

7.5.1 UCIe Link DVSEC

This is the basic capability register set that is required to operate a UCIe Link. And this is one of two DVSEC capabilities defined for UCIe in the first generation. Not all the registers in the capability are applicable to all device/port types. The applicable registers for each device/port type are indicated in the right-hand side of Figure 7-4. Software may use the presence of this DVSEC to differentiate between a UCIe device vs. a standard PCIe or CXL device. Software may use this DVSEC to differentiate between a UCIe Root Port and a standard PCIe or CXL Root Port.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 7-4. UCIe Link DVSEC



1. Applies to UCIe-EP, UCIe-USP, UCIe-Retimer.
2. Applies to UCIe-EP, UCIe-USP when paired with a retimer.
3. Applies to UCIe-RP.
4. Applies to UCIe-DSP.
5. Software writes to this register need to be broadcast to both D2D Adapter and PHY blocks because some registers could be implemented in either block or both blocks.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.1 PCI Express Extended Capability Header (Offset 0h)

Set as follows for UCIe Link DVSEC. All bits in this register are RO.

Table 7-5. UCIe Link DVSEC - PCI Express Extended Capability Header

Field	Bit Location	Value	Comments
Capability ID	15:0	0023h	Value for PCI Express DVSEC capability
Revision ID	19:16	1h	Latest revision of the DVSEC capability
Next Capability Offset	31:20	Design Dependent	**For UCIe Link DVSEC in UiRB**: Set to point to MSI capability associated with this UCIe Link. The offset is in granularity of Bytes from the base address of UiRB. For example, if this is set to 100h, the MSI capability is located at offset of 100h from the base of UiRB. MSI capability must set its "Next Capability offset" to 0h to indicate end of capability chain for the specific UCIe Link. **UCIe Link DVSEC in UiSRB**: Set to point to the UCIe Link DVSEC capability of the next UCIe Link associated with a downstream port of the switch. The last UCIe Link DVSEC capability will set this offset to 0h indicating there are no more UCIe Links on downstream ports. The offset is in granularity of Bytes from the base address of UiSRB. For example, if this is set to 100h, the next DVSEC capability for the next Link is located at offset of 100h from the base of UiSRB. **Retimer:** Set to 0h **Others**: design dependent

7.5.1.2 Designated Vendor Specific Header 1, 2 (Offsets 4h and 8h)

A few things to note on the various fields described in Table 7-6. DVSEC Revision ID field represents the version of the DVSEC structure. The DVSEC Revision ID is incremented whenever the structure is extended to add more functionality. Backward compatibility shall be maintained during this process. For all values of n, DVSEC Revision ID n+1 structure may extend Revision ID n by replacing fields that are marked as reserved in Revision ID n, but must not redefine the meaning of existing fields. Software that was written for a lower Revision ID may continue to operate on UCIe DVSEC structures with a higher Revision ID, but will not be able to take advantage of new functionality.

All bits in this register are RO.

Table 7-6. UCIe Link DVSEC - Designated Vendor Specific Header 1, 2

Register	Field	Bit Location	Value
Designated Vendor-Specific Header 1 (offset 04h)	DVSEC Vendor ID	15:0	D2DEh
	DVSEC Revision	19:16	0h
	Length	31:20	Device dependent. See Section 7.5.1.19 for some examples.
Designated Vendor-Specific Header 2 (offset 08h)	DVSEC ID	15:0	0h

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.3 Capability Descriptor (Offset Ah)

Provides a way for SW to discover which optional capabilities are implemented by the UCIe Port/ Device.

Table 7-7. UCIe Link DVSEC - Capability Descriptor

Bit	Attribute	Description
2:0	RO	**Number of Register locators:** 0h: 2 Register Locators 1h: 3 Register Locators 2h: 4 Register Locators .. 6h: 8 Register locators 7h: 1 Register Locator For this revision of UCIe, only values 0h, 1h, 2h and 7h are valid.
3	RO(RP/DSP), HWInit(EP/USP), RsvdP(Retimer)	**Sideband mailbox Registers Present:** 0h: No sideband mailbox register set present in this capability 1h: Sideband mailbox register set present in this capability For RP/DSP, default value of this is 1. EP/USP must set this bit when they are paired with a retimer and must clear this bit in all other scenarios.
7:4	RO(DSP), RsvdP (Others)	**Number of Switch DSPs associated with this UCIe Link** Applies only to UCIe Link DVSEC in UiSRB. The specific 'port number' values of each Switch downstream port associated with this UCIe Link is called out in the Associated Port Number register(s) in this capability. 0h – 1 Port 1h – 2 ports .. Fh – 16 ports 'Port Number' is bits 31:24 of the PCIe Link capabilities register of the downstream port. For first generation of UCIe, only values 0h and 1h are legal.
15:8	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.4 UCIe Link DVSEC - UCIe Link Capability (Offset Ch)

Basic characteristics of the UCIe Link are discovered by SW using this register.

Table 7-8. UCIe Link DVSEC - UCIe Link Capability (Sheet 1 of 2)

Bit	Attribute	Description
0	RO	**Raw Format:** If set, indicates the Link can support Raw Format.
3:1	HWInit	**Max Link Width** 0h: x16 1h: x32 2h: x64 3h: x128 4h: x256 Others - Reserved
7:4	HWInit	**Max Link Speeds** 0h: 4GT/s 1h: 8GT/s 2h: 12GT/s 3h: 16GT/s 4h: 24GT/s 5h: 32GT/s Others: Reserved
8	RO(Retimer), RsvdP(others)	**Retimer** - Set by retimer to indicate it to SW
9	RsvdP(Retimer), RO(others)	**Multi-stack capable** 0 - single stack capable 1 - multi stack capable In first rev of spec, only 2 stacks max is possible
10	RO	**Advanced Packaging** 0 = Standard package mode for UCIe Link 1 = Advanced package mode for UCIe Link
11	RO	**68B Flit Format for Streaming Protocol** If set, indicates 68B Flit Format is supported for Streaming Protocol. This is only set if at least one of the Protocol Layers is Streaming Protocol.
12	RO	**Standard 256B End Header Flit Format for Streaming Protocol** If set, indicates Standard 256B End Header Flit Format is supported for Streaming Protocol. This is only set if at least one of the Protocol Layers is Streaming Protocol.
13	RO	**Standard 256B Start Header Flit Format for Streaming Protocol** If set, indicates Standard 256B Start Header Flit Format is supported for Streaming Protocol. This is only set if at least one of the Protocol Layers is Streaming Protocol.
14	RO	**Latency-Optimized 256B Flit Format without Optional Bytes for Streaming Protocol** If set, indicates Latency-Optimized 256B without Optional Bytes Flit Format is supported for Streaming Protocol. This is only set if at least one of the Protocol Layers is Streaming Protocol.
15	RO	**Latency-Optimized 256B Flit Format with Optional Bytes for Streaming Protocol** If set, indicates Latency-Optimized 256B with Optional Bytes Flit Format is supported for Streaming Protocol. This is only set if at least one of the Protocol Layers is Streaming Protocol.
16	RO	**Enhanced Multi-protocol Capable** 0 = Not capable of multi-protocol with different protocols 1 = Capable of multi-protocol with different protocols
17	RO	**Standard Start Header Flit for PCIe Protocol** If set, indicates Standard Start Header 256B Flit Format is supported for PCIe protocol. This is only set if at least one of the Protocol Layers is PCIe protocol.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-8. UCIe Link DVSEC - UCIe Link Capability (Sheet 2 of 2)

Bit	Attribute	Description
18	RO	**Latency-Optimized Flit with Optional Bytes for PCIe Protocol** If set, indicates that the Latency-Optimized Flit Format with Optional Bytes is supported for PCIe. This is only set if at least one of the Protocol Layers is PCIe protocol.
19	RO	**'Runtime Link Testing Parity' Feature Error Signaling** If set, design supports signaling errors detected during Runtime link testing with parity as Correctable errors. If cleared, this error signaling mechanism is not supported.
20	HWInit	**APMW (Advanced Package Module Width)** If set, indicates the Advanced Package Module size is x32 or a x64 module operating in x32 mode (decided at integration time). If reset, indicates x64 Advanced Package Module.
21	RO/RsvdP	**x32 Width Support in x64 Module** If set, indicates that a x64 Advanced Package Module can operate in x32 mode; otherwise, it cannot operate in x32 mode. For x32 Advanced Package Module, this bit is reserved.
31:22	RsvdP	Reserved

7.5.1.5 UCIe Link DVSEC - UCIe Link Control (Offset 10h)

Basic UCIe Link control bits are in this register.

Table 7-9. UCIe Link DVSEC - UCIe Link Control (Sheet 1 of 3)

Bit	Attribute	Description
0	RW(RP/DSP), HWInit(Others)	**Raw Format Enable**: If set, enables the Link to negotiate Raw Format during Link training. Default value of this is 0b for RP and firmware/SW sets this bit based on system usage scenario. Switch DSP can set the default via implementation-specific mechanisms such as straps/FW/etc., to account of system usage scenario (like UCIe retimer). This allows for the DSP Link to train up without Software intervention and be UCIe-unaware-OS compatible.
1	RW(RP/DSP), RO(EP/DSP), RsvdP (Retimer)	**Multi-stack enable**: When set, multi-stack training is enabled else not. In first gen of UCIe, only 2 stacks max are possible. Should not be set to 1b if Enhanced Multi-Protocol enable is set to 1b. Default is same as 'Multi-stack Capable' bit in UCIe Link Capability register.
5:2	RW(RP/DSP), RsvdP (Others)	**Target Link Width** 0h: Reserved 1h: Reserved 2h: x16 3h: x32 4h: x64 5h: x128 6h: x256 Others are Reserved. Default is same as 'Max Link Width' field in UCIe Link Capability Register.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-9. UCIe Link DVSEC - UCIe Link Control (Sheet 2 of 3)

Bit	Attribute	Description
9:6	RW(RP/DSP), RsvdP (Others)	**Target Link Speed** 0h: 4GT/s 1h: 8GT/s 2h: 12GT/s 3h: 16GT/s 4h: 24GT/s 5h: 32GT/s Others: Reserved Default is same as 'Max Link speed' field in UCIe Link Capability Register.
10	RW, with auto clear(RP/DSP), RsvdP (Others)	**Start UCIe Link training** - When set to 1, Link training starts with Link Control bits programmed in this register and with the protocol layer capabilities. Bit is automatically cleared when Link training completes with either success or error. The status register captures the final status of the Link training. Note that if the Link is up when this bit is set to 1 from 0, the Link will go through full training through Link Down state thus resetting everything beneath the Link. Primary usage intended for this bit is for initial Link training out of reset on the host side. **Note:** For downstream ports of a switch with UCIe, local HW/FW has to autonomously initiate Link training after a conventional reset, without waiting for higher level SW to start the training via this bit, to ensure backward compatibility. Default is 0.
11	RW with auto clear (RP/DSP), RsvdP (Others)	**Retrain UCIe Link** - When set to 1, Link that is already up (Link_status=up) will be retrained without going through Link Down state. SW can use this bit to potentially recover from Link errors. If the Link is down (Link_status=down) when this bit is set, there is no effect from this bit being set. SW should use the 'Start UCIe Link training' bit in case the Link is down. The Link_status bit in the status register can be read by software to determine whether to use this bit or not. Note that when retrain happens, the Link speed or width can change because of reliability reasons, and it will be captured through the appropriate status bit in the Link Status register. Bit is automatically cleared when Link retraining completes with either success or error (as reported via the appropriate status bits in the Link Status register) or if the Link retrain did not happen at all for the reason stated earlier. Default is 0.
12	RW	**PHY layer Clock gating enable** - When set, the dynamic clock gating of the forwarded clock is enabled. Otherwise it is free running. Default is 1.
13	RW	**68B Flit Format for Streaming Protocol** If set, enables 68B Flit Format advertisement if the corresponding capability is supported. Default is same as the '68B Flit Format for Streaming Protocol' bit in the UCIe Link Capability register.
14	RW	**Standard 256B End Header Flit Format for Streaming Protocol** If set, enables Standard 256B End Header Flit Format advertisement if the corresponding capability is supported. Default is same as the 'Standard 256B End Header Flit Format for Streaming Protocol' bit in the UCIe Link Capability register.
15	RW	**Standard 256B Start Header Flit Format for Streaming Protocol** If set, enables Standard 256B Start Header Flit Format advertisement if the corresponding capability is supported. Default is same as the 'Standard 256B Start Header Flit Format for Streaming Protocol' bit in the UCIe Link Capability register.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-9. UCIe Link DVSEC - UCIe Link Control (Sheet 3 of 3)

Bit	Attribute	Description
16	RW	**Latency -Optimized 256B Flit Format without Optional Bytes for Streaming Protocol** If set, enables Latency-Optimized 256B Flit Format without Optional bytes advertisement if the corresponding capability is supported. Default is same as the 'Latency-Optimized 256B Flit Format without for Streaming Protocol' bit in the UCIe Link Capability register.
17	RW	**Latency-Optimized 256B Flit Format with Optional Bytes for Streaming Protocol** If set, enables Latency-Optimized 256B Flit Format with Optional bytes advertisement if the corresponding capability is supported. Default is same as the 'Latency-Optimized 256B Flit Format for Streaming Protocol' bit in the UCIe Link Capability register.
18	RW (RP/DSP), RO (EP/DSP), RsvdP (Retimer)	**Enhanced Multi-Protocol Enable** When set, enhanced multi-protocol training is enabled else not. Enhanced Multi-Protocol permits 2 stacks with the same or different protocols. Default is same as 'Enhanced Multi-Protocol Capable' bit in UCIe Link Capability register.
19	RW	**Standard Start Header Flit for PCIe Protocol** If set, enables Standard Start Header 256B Flit Format for PCIe protocol. Default is same as 'Standard Start Header Flit for PCIe Protocol' bit in UCIe Link Capability register.
20	RW	**Latency-Optimized Flit with Optional Bytes for PCIe Protocol** If set, enables the Latency-Optimized Flit Format with Optional Byte for PCIe. Default is same as 'Latency-Optimized Flit with Optional Bytes for PCIe Protocol' bit in UCIe Link Capability register.
31:21	RsvdP	Reserved

7.5.1.6 UCIe Link DVSEC - UCIe Link Status (Offset 14h)

Basic UCIe Link status bits are in this register.

Table 7-10. UCIe Link DVSEC - UCIe Link Status (Sheet 1 of 2)

Bit	Attribute	Description
0	RO	**Raw Format Enabled**: If set, indicates the Adapter negotiated Raw Format operation with remote Link partner. This bit is only valid when Link Status bit in this register indicates 'Link Up'.
1	RsvdZ (Retimer), RO (Others)	**Multi-stack enabled**: When set, multi-stack training has been enabled with remote training partner. This bit is only valid when Link Status bit in this register indicates 'Link Up'.
2	RsvdZ (Retimer), RO (Others)	**Enhanced Multi-protocol Enabled** When set, multi-protocol training has been enabled with remote training partner. This bit is only valid when Link Status bit in this register indicates 'Link Up'.
3	RO	**x32 Advanced Package Module Enabled** When set, indicates that the Advanced Package operating module size is x32.
6:4	RsvdZ	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-10. UCIe Link DVSEC - UCIe Link Status (Sheet 2 of 2)

Bit	Attribute	Description
10:7	RO	**Link Width enabled** 0h: Reserved 1h: x8 2h : x16 3h : x32 4h : x64 5h : x128 6h : x256 This has meaning only when Link status bit shows Link is up.
14:11	RO	**Link Speed enabled** 0h: 4GT/s 1h: 8GT/s 2h: 12GT/s 3h: 16GT/s 4h: 24GT/s 5h: 32GT/s Others: Reserved This field has meaning only when Link status field shows Link is up
15	RO	**Link Status** 0 - Link is down. 1 - Link is up Transitioning a Link from down to up requires a full Link training, which can be achieved using one of these methods: • Start Link training via the bits in the UCIe Link Control register of the upstream device • Using the protocol layer reset bit associated with the Link, like the SBR bit in the BCTL register of the RP P2P space • Using the protocol layer Link Disable bit associated with the Link, like the Link Disable bit in the Link CTL register of the PCIe capability register in the RP P2P space, and then releasing the disable. **Note:** If the Link is actively retraining, this bit reflects a value of 1b.
16	RO	**Link Training/Retraining** 1b - Currently Link is training or retraining 0b - Link is not training or retraining
17	RW1C (RP/DSP), RsvdZ (Others)	**Link Status changed** 1b - Link either transitioned from up to down or down to up. 0b - No Link status change since the last time SW cleared this bit
18	RW1C (RP/DSP), RsvdZ (Others)	**HW autonomous BW changed** UCIe autonomously changed the Link width or speed to correct Link reliability related issues.
19	RW1CS	**Detected UCIe Link correctable error** Further details of specific type of correctable error is found in Table 7-30 register.
20	RW1CS	**Detected UCIe Link Uncorrectable Non-fatal error** Further details of specific type of correctable error is found in Table 7-27 register.
21	RW1CS	**Detected UCIe Link Uncorrectable Fatal error** Further details of specific type of correctable error is found in Table 7-27 register.
25:22	RO	**Flit Format Status** This field and the Flit Format field in the Header Log 2 register in the D2D/PHY register block (see Section 7.5.3.8) are mirror copies. This field indicates the negotiated Flit Format. This field is only valid when Link Status bit in this register indicates 'Link Up'.
31:26	RsvdZ	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.7 UCIe Link DVSEC - Link Event Notification Control (Offset 18h)

Link event notification related controls are in this register.

Table 7-11. UCIe Link DVSEC - Link Event Notification Control

Bit	Attribute	Description
0	RW(RP/DSP), RsvdP (Others)	**'Link Status changed' UCIe Link Event Interrupt enable** 0: Reporting of this event via interrupt is not enabled 1: Reporting of this event via interrupt is enabled. Default is 0
1	RW(RP/DSP), RsvdP (Others)	**'HW autonomous BW changed' UCIe Link Event Interrupt enable** 0: Reporting of this event via interrupt is not enabled 1: Reporting of this event via interrupt is enabled Default is 0
10:2	RsvdP	Reserved
15:11	RO(RP/DSP), RsvdP(Others)	**Link Event Notification Interrupt number** This field indicates which MSI vector (for host UCIe Links), or MSI/MSI-X vector (for switch DSP UCIe Links) is used for the interrupt message generated in association with the events that are controlled via this register. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the Message Control Register for MSI. For first generation of UCIe, maximum 2 interrupt vectors could be requested for UCIe related functionality and the 'Link event' is one of them. For MSI-X (applicable only for interrupts from Switch DSPs with UCIe Links), the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. For UCIe related interrupts, a switch should request its interrupt requirements from either MSI or MSI-X capability but not both.

7.5.1.8 UCIe Link DVSEC - Error Notification Control (Offset 1Ah)

Link error notification related controls are in this register.

Note: This register only controls the propagation of the error condition and it has no impact on the setting of the appropriate status bits in the Link Status register, when the relevant error happens.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-12. UCIe Link DVSEC - Error Notification Control (Sheet 1 of 3)

Bit	Attribute	Description
0	RW(RP/DSP), RsvdP (Others)	**'Correctable error detected' protocol layer based reporting enable** 0: Reporting of this error via protocol layer mechanism is not enabled 1: Reporting of this error via protocol layer mechanism is enabled Default is 0 When enabled, the reported PCIe/CXL protocol layer correctable error type is 'Correctable internal error'. This bit is applicable for only RP/DSP.
1	RW	**'Correctable error detected' UCIe Link Error Interrupt enable** RP/DSP 0: Reporting of this error via UCIe Link Error interrupt is not enabled 1: Reporting of this error via UCIe Link Error interrupt is enabled EP/USP 0: Reporting of this error via sideband error message is not enabled 1: Reporting of this error via sideband error message is enabled Note that in the case of EP/USP connected to a retimer, their sideband error message targets the retimer and how the retimer sends it across to the partner retimer is vendor specific. Retimer connected to RP/DSP 0: Reporting of this error via sideband error message to RP/DSP is not enabled 1: Reporting of this error via sideband error message to RP/DSP is enabled Retimer connected to EP/USP 0: Reporting of this error to the partner retimer is disabled. 1: Reporting of this error to the partner retimer is enabled. The specific mechanism for reporting the error to the partner retimer is vendor-specific. Default is 0
2	RW(RP/DSP), RsvdP (Others)	**'Uncorrectable non-fatal error detected' protocol layer based reporting enable** 0: Reporting of this error via protocol layer mechanism is not enabled 1: Reporting of this error via protocol layer mechanism is enabled Default is 0 This bit is applicable for only RP/DSP.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-12. UCIe Link DVSEC - Error Notification Control (Sheet 2 of 3)

Bit	Attribute	Description
3	RW	**'Uncorrectable non-fatal error detected' UCIe Link Error Interrupt enable** RP/DSP 0: Reporting of this error via UCIe Link Error interrupt is not enabled 1: Reporting of this error via UCIe Link Error interrupt is enabled EP/USP 0: Reporting of this error via sideband error message is not enabled 1: Reporting of this error via sideband error message is enabled Note that in the case of EP/USP connected to a retimer, their sideband error message targets the retimer and how the retimer sends it across to the partner retimer is vendor specific. Retimer connected to RP/DSP 0: Reporting of this error via sideband error message to RP/DSP is not enabled 1: Reporting of this error via sideband error message to RP/DSP is enabled Retimer connected to EP/USP 0: Reporting of this error to the partner retimer is disabled. 1: Reporting of this error to the partner retimer is enabled. The specific mechanism for reporting the error to the partner retimer is vendor specific. Default is 0
4	RW (RP/DSP), RsvdP (Others)	**'Uncorrectable fatal error detected' protocol layer based reporting enable** 0: Reporting of this error via protocol layer mechanism is not enabled 1: Reporting of this error via protocol layer mechanism is enabled Default is 0 When enabled, the reported PCIe/CXL protocol layer uncorrectable error type is 'Uncorrectable internal error' This bit is applicable for only RP/DSP.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-12. UCIe Link DVSEC - Error Notification Control (Sheet 3 of 3)

Bit	Attribute	Description
5	RW	**'Uncorrectable fatal error detected' UCIe Link Error Interrupt enable** RP/DSP 0: Reporting of this error via UCIe Link Error interrupt is not enabled 1: Reporting of this error via UCIe Link Error interrupt is enabled EP/USP 0: Reporting of this error via sideband error message is not enabled 1: Reporting of this error via sideband error message is enabled Note that in the case of EP/USP connected to a retimer, their sideband error message targets the retimer and how the retimer sends it across to the partner retimer is vendor specific. Retimer connected to RP/DSP 0: Reporting of this error via sideband error message to RP/DSP is not enabled 1: Reporting of this error via sideband error message to RP/DSP is enabled Retimer connected to EP/USP 0: Reporting of this error to the partner retimer is disabled. 1: Reporting of this error to the partner retimer is enabled. The specific mechanism for reporting the error to the partner retimer is vendor specific. Default is 0
10:6	RsvdP	Reserved
15:11	RW	**Link Error Notification Interrupt number** This field indicates which MSI vector (for host UCIe Links), or MSI/MSI-X vector (for switch DSP UCIe Links) is used for the interrupt message generated in association with the events that are controlled via this register. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the Message Control Register for MSI. For first generation of UCIe, maximum 2 interrupt vectors could be requested for UCIe related functionality and the 'Error' is one of them. For MSI-X (applicable only for interrupts from Switch DSPs with UCIe Links), the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. For UCIe related interrupts, a switch should request its interrupt requirements from either MSI or MSI-X capability but not both.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.9 UCIe Link DVSEC - Register Locator 0, 1, 2, 3 Low (Offset 1Ch and when Register Locators 1, 2, 3 are present Offsets 24h, 2Ch, and 34h respectively)

The starting address of the MMIO-mapped register blocks for D2D/PHY, Compliance/Test and Implementation-specifics are located by SW via these registers.

Note: All register blocks start with a header section that indicates the size of the block in multiples of 4 KB.

Table 7-13. UCIe Link DVSEC - Register Locator 0, 1, 2, 3 Low

Bit	Attribute	Description
2:0	RO	**Register BIR** For UCIe DVSEC capability in host UiRB, Switch UiSRB and in UCIe Retimer, this field is reserved. For others, its defined as follows: Indicates which one of a Dev0/Fn0 Base Address Registers, located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BAR Equivalent Indicator (BEI), is used to map the UCIe Register blocks into Memory Space. Defined encodings are: • 0 Base Address Register 10h • 1 Base Address Register 14h • 2 Base Address Register 18h • 3 Base Address Register 1Ch • 4 Base Address Register 20h • 5 Base Address Register 24h All other Reserved. The Registers block must be wholly contained within the specified BAR. For a 64-bit Base Address Register, the Register BIR indicates the lower DWORD.
6:3	RO	**Register Block Identifier** • Identifies the type of UCIe register blocks. Defined encodings are:0h UCIe D2D/PHY Register Block • 1h UCIe Test/Compliance Register Block • 2h D2D Adapter Implementation specific register block • 3h PHY Implementation specific register block • All other encodings are reserved The same register block identifier value cannot be repeated in multiple Register Locator entries.
11:7	RsvdP	Reserved
31:12	RO	**Register Block Offset** Addr[31:12] of the 4-KB aligned offset from the starting address of the Dev0/Fn0 BAR pointed to by the Register BIR field (for EP, Switch USP) or from the start of UiRB/UiSRB region (for hosts/Switch). This field is reserved for retimers.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.10 UCIe Link DVSEC - Register Locator 0, 1, 2, 3 High (Offset 20h and when Register Locators 1, 2, 3 Are Present Offsets 28h, 30h, and 38h respectively)

Addr[63:32] of the starting address of the MMIO-mapped register blocks for D2D/PHY, Compliance/ Test and Implementation-specifics are located by SW via these registers.

Note: All register blocks start with a header section that indicates the size of the block in multiples of 4 KB.

Table 7-14. UCIe Link DVSEC - Register Locator 0, 1, 2, 3 High

Bit	Attribute	Description
63:32	RO	**Register Block Offset** Addr[63:32] of the 4-KB aligned offset from the starting address of the Dev0/Fn0 BAR pointed to by the Register BIR field (for EP, Switch USP) or from the start of UiRB/UiSRB region (for hosts/Switch). This field is reserved for retimers.

7.5.1.11 UCIe Link DVSEC - Sideband Mailbox Index Low (Offset is design dependent)

Mailbox registers are to be implemented by all hosts with UCIe Links. Switches with downstream UCIe Links and EP/USP, when paired with UCIe Retimer, should also implement this register. Note that accesses to mailbox are inherently non-atomic in nature and hence it is up to higher-level software to coordinate access to any mailbox-related register so that one agent does not step on another agent using the mailbox mechanism. Those mechanisms for software coordination are beyond the scope of this specification.

Table 7-15. UCIe Link DVSEC - Sideband Mailbox Index Low

Bit	Attribute	Description
4:0	RW	**Opcode** 00000b 32b Memory Read 00001b 32b Memory Write 00100b 32b Configuration Read 00101b 32b Configuration Write 01000b 64b Memory Read 01001b 64b Memory Write 01100b 64b Configuration Read 01101b 64b Configuration Write OthersReserved Default 00100
12:5	RW	**BE[7:0]** Default Fh
31:13	RW	**Addr[18:0] of Sideband Accesses** Format for this field is as defined in the sideband interface definition in Chapter 6.0. **Note:** The address offset defined as part of this address field is DWORD aligned for 32bit accesses and QWORD aligned for 64bit accesses. Default is 0.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.12 UCIe Link DVSEC - Sideband Mailbox Index High (Offset is design dependent)

Mailbox registers are to be implemented by all hosts with UCIe Links. Switches with downstream UCIe Links and EP/USP, when paired with UCIe Retimer, should also implement this register. Note that accesses to mailbox are inherently non-atomic in nature and hence it is up to higher-level software to coordinate access to any mailbox-related register so that one agent does not step on another agent using the mailbox mechanism. Those mechanisms for software coordination are beyond the scope of this specification.

Table 7-16. UCIe Link DVSEC - Sideband Mailbox Index High

Bit	Attribute	Description
4:0	RW	**Addr[23:19] of Sideband Accesses** Format for this field is as defined in the sideband interface definition in Chapter 6.0. Default is 0.
31:5	RsvdP	Reserved

7.5.1.13 UCIe Link DVSEC - Sideband Mailbox Data Low (Offset is design dependent)

Table 7-17. UCIe Link DVSEC - Sideband Mailbox Data Low

Bit	Attribute	Description
31:0	RW	For sideband write opcodes, this carries the write data [31:0] to the destination. For sideband read opcodes, this carries the data read from the destination when the Write/Read Trigger bit in the Mailbox Control register is cleared, after it was initially set. This field's value is undefined until the Write/Read trigger bit is cleared on reads.

7.5.1.14 UCIe Link DVSEC - Sideband Mailbox Data High (Offset is design dependent)

Table 7-18. UCIe Link DVSEC - Sideband Mailbox Data High

Bit	Attribute	Description
31:0	RW	For sideband write opcodes, this carries the write data [63:32] to the destination. For sideband read opcodes, this carries the data read from the destination when the Write/Read Trigger bit in the Mailbox Control register is cleared, after it was initially set. This field's value is undefined until the Write/Read trigger bit is cleared on reads. For 32b Writes/Reads, this register does not carry valid data.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.15 UCIe Link DVSEC - Sideband Mailbox Control (Offset is design dependent)

Table 7-19. UCIe Link DVSEC - Sideband Mailbox Control

Bit	Attribute	Description
0	RW, with auto clear	**Write/Read trigger**: When this bit is written to a 1 from a value of 0, the mailbox generates traffic on the sideband interface, using the contents of the Mailbox Header and Data registers. This bit automatically clears when the write or read access triggered by this bit being set, is complete on the sideband bus. SW can poll this bit to know when the write/read has actually completed at the destination. It can then go read the Mailbox data register for the read data.
7:1	RsvdP	Reserved

7.5.1.16 UCIe Link DVSEC - Sideband Mailbox Status (Offset is design dependent)

Table 7-20. UCIe Link DVSEC - Sideband Mailbox Status

Bit	Attribute	Description
1:0	RW1C(RP/DSP), RW1C(EP/USP), when implemented	**Write/Read status:** 00 - CA received 01 - UR received 01 - Reserved 11 - Success This bit has valid value only when the Write/Read Trigger bit is cleared from being a 1 prior to it.
7:2	RsvdZ	Reserved

7.5.1.17 UCIe Link DVSEC - Requester ID (Offset is design dependent)

Table 7-21. UCIe Link DVSEC - Requester ID

Bit	Attribute	Description
23:0	RW(RP)/RsvdP (Others)	Applicable only for host side UCIe Links. Segment No: Bus No: Dev No: Fn No for MSIs triggered on behalf of the associated UCIe Link **Note:** For MSIs issued on behalf of UCIe Links on downstream ports of switches, the Switch USP BDF is used. UCIe Link DVSEC capabilities in UiSRB implement this as RO 0.
31:24	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.1.18 UCIe Link DVSEC - Associated Port Numbers (Offset is design dependent)

These registers apply only to UCIe Link DVSEC capabilities present in UiSRB.

Table 7-22. UCIe Link DVSEC - Associated Port Numbers

Bit	Attribute	Description
7:0	RO	**Port Number 1** - 'Port number' of the 1st switch DSP associated with this UCIe. This value is from the Link Capabilities register of that switch DSP.
15:8	RO	**Port Number 2** - 'Port number' of the 2nd switch DSP associated with this UCIe, if any. If there is no 2nd switch DSP associated with this UCIe Link, this field is treated as reserved and should not be included as part of the "length" field of the 'Designated Vendor specific Header 1' register and SW should not consider this as part of the DVSEC capability. **Note:** Only a maximum of two Port numbers can be associated with a UCIe Link in the current revision of the specification.

7.5.1.19 Examples of setting the Length field in DVSEC for various Scenarios

Example#1: UCIe EP supporting 2 Register Locators and not associated with a UCIe-Retimer, would set the length field in DVSEC capability to indicate 48B.

Example#2: Host UiRB supporting 3 register locators would set the length to indicate 84B.

Example#3: Switch UiSRB supporting 3 register locators and associated with just 1 DSP port to a UCIe Link, would set the length to indicate 85B.

7.5.2 UCIe Switch Register Block (UiSRB) DVSEC Capability

This capability can only be present in the config space of the upstream port of a Switch. There can be multiple of these in the same USP config space.

7.5.2.1 PCI Express Extended Capability Header (Offset 0h)

Set as follows for UCIe Switch Register Block DVSEC. All bits in this register are RO.

Table 7-23. UiSRB DVSEC - PCI Express Extended Capability Header

Field	Bit Location	Value	Comments
Capability ID	15:0	0023h	Value for PCI Express DVSEC capability
Revision ID	19:16	1h	Latest revision of the DVSEC capability
Next Capability Offset	31:20	Design Dependent	

7.5.2.2 Designated Vendor Specific Header 1, 2 (Offsets 4h and 8h)

A few things to note on the various fields described in Table 7-6. DVSEC Revision ID field represents the version of the DVSEC structure. The DVSEC Revision ID is incremented whenever the structure is extended to add more functionality. Backward compatibility shall be maintained during this process. For all values of n, DVSEC Revision ID n+1 structure may extend Revision ID n by replacing fields that are marked as reserved in Revision ID n, but must not redefine the meaning of existing fields. Software that was written for a lower Revision ID may continue to operate on UCIe DVSEC structures with a higher Revision ID, but will not be able to take advantage of new functionality.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

All bits in this register are RO.

Table 7-24. UiSRB DVSEC - Designated Vendor Specific Header 1, 2

Register	Field	Bit Location	Value
Designated Vendor-Specific Header 1 (offset 04h)	DVSEC Vendor ID	15:0	D2DEh
	DVSEC Revision	19:16	0h
	Length	31:20	14h
Designated Vendor-Specific Header 2 (offset 08h)	DVSEC ID	15:0	1h

7.5.2.3 UCIe Switch Register Block (UiSRB) Base Address (Offset Ch)

All bits in this register are RO.

Table 7-25. UiSRB DVSEC - UiSRB Base Address

Bit	Attributes	Description
0	RO	**Register BIR** Indicates which one of a Switch USP Function's Base Address Registers, located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BAR Equivalent Indicator (BEI), is used to locate the UCIe Switch Register Block. Defined encodings are: • 0 Base Address Register 10h • 1 Base Address Register 14h • All other Reserved. The Registers block must be wholly contained within the specified BAR. For a 64-bit Base Address Register, the Register BIR indicates the lower DWORD.
11:1	RsvdP	Reserved
63:12	RO	**Register Block Offset** A 4-KB-aligned offset from the starting address of the Switch USP BAR indicated by the Register BIR field. The BAR value + Offset indicated in this register is where the UCIe Switch Register Block (UiSRB) starts. Ex: If this register is 100, UiSRB starts at the <64bit BAR value + 100000h>

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3 D2D/PHY Register Block

These registers occupy 8 KB of register space. The first 4 KB are for the D2D Adapter, and the next 4 KB are for the Physical Layer. In the PHY register block, extended capabilities start at Offset 200h. If an implementation does not support any extended capabilities, it must implement a NULL capability at Offset 200h (which implements 0h for the DWORD at that offset). The D2D Adapter registers are enumerated below. The location of these registers in the system MMIO region is as described in Section 7.3.

7.5.3.1 UCIe Register Block Header

Table 7-26. D2D/PHY Register Block - UCIe Register Block Header (Offset 0h)

Bit	Attributes	Description
15:0	RO	**Vendor ID** Default is set to Vendor ID assigned for UCIe Consortium - D2DEh
31:16	RO	**Vendor ID Register Block** Set to 0h to indicate D2D/PHY register block
35:32	RO	**Vendor Register Block Version** Set to 0h
63:36	RsvdP	Reserved
95:64	RO	**Vendor Register Block Length** - The number of bytes in the register block including the UCIe Register block header. Default is 2000h.
127:96	RsvdP	Reserved

7.5.3.2 Uncorrectable Error Status Register (Offset 10h)

Table 7-27. Uncorrectable Error Status Register (Sheet 1 of 2)

Bit	Attribute	Description
0	RW1CS	**Adapter Timeout:** Set to 1b by hardware if greater than 8ms has elapsed for Adapter handshakes with its remote Link partner. The Header Log 2 register captures the reason for a timeout. This error will bring the main Link Down. Default Value is 0b.
1	RW1CS	**Receiver Overflow**: Set to 1b by hardware if Receiver overflow errors are detected. The Header Log 2 register captures the encoding to indicate the type of Receiver overflow. This error will bring the Link Down. Default Value is 0b.
2	RW1CS	**Internal Error**: Set to 1b by hardware if an internal Data path error is detected. Examples of such errors include (but not limited to) uncorrectable error correcting code (ECC) error in the Retry buffer, sideband parity errors etc. This error will bring the Link Down. It includes fatal error indicated by the Physical Layer that brought the Link Down. Default Value is 0b.
3	RW1CS(RP/DSP/ Retimer), RsvdZ(Others)	**Sideband Fatal Error Message received**: Set to 1b by hardware if the Adapter received a Fatal {ErrMsg} sideband message. Default Value is 0b.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-27. Uncorrectable Error Status Register (Sheet 2 of 2)

Bit	Attribute	Description
4	RW1CS(RP/DSP/ Retimer), RsvdZ(Others)	**Sideband Non-Fatal Error Message received**: Set to 1b by hardware if the Adapter received a Non-Fatal {ErrMsg} sideband message. Default Value is 0b.
5	RW1CS	**Invalid Parameter Exchange**: Set to 1b if the Adapter was not able to determine a valid protocol or Flit Format for operation.
31:6	RsvdZ	Reserved

7.5.3.3 Uncorrectable Error Mask Register (Offset 14h)

The Uncorrectable Error Mask Register controls reporting of individual errors. When a bit is 1b in this register, the corresponding error status bit in the Uncorrectable Error Status register is not forwarded to the Protocol Layer for escalation/signaling but it does not impact error logging in the "First Fatal Error Indicator" field in the Header Log 2 register.

Table 7-28. Uncorrectable Error Mask Register

Bit	Attribute	Description
0	RWS	**Adapter Timeout Mask** Default Value is 1b.
1	RWS	**Receiver Overflow Mask** Default Value is 1b.
2	RWS	**Internal Error Mask** Default Value is 1b.
3	RWS	**Sideband Fatal Error Message received Mask** Default Value is 1b.
4	RWS	**Sideband Non-Fatal Error Message received Mask** Default Value is 1b.
5	RWS	**Invalid Parameter Exchange Mask** Default Value is 1b.
31:6	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.4 Uncorrectable Error Severity Register (Offset 18h)

The Uncorrectable Error Severity register controls whether an individual error is reported as a Non-fatal or Fatal error. An error is reported as a fatal uncorrectable error when the corresponding bit in the severity register is 1b. If the bit is 0b, the corresponding error is reported as a non-fatal uncorrectable error.

Table 7-29. Uncorrectable Error Severity Register

Bit	Attribute	Description
0	RWS	**Adapter Timeout Severity** Default Value is 1b.
1	RWS	**Receiver Overflow Severity** Default Value is 1b.
2	RWS	**Internal Error Severity** Default Value is 1b.
3	RWS	**Sideband Fatal Error Message received Severity** Default Value is 1b.
4	RWS	**Sideband Non-Fatal Error Message received Severity** Default Value is 0b.
5	RWS	**Invalid Parameter Exchange Severity** Default Value is 1b
31:6	RsvdP	Reserved

7.5.3.5 Correctable Error Status Register (Offset 1Ch)

Table 7-30. Correctable Error Status Register

Bit	Attribute	Description
0	RW1CS	**CRC Error Detected**: Set to 1b by hardware if the Adapter detected a CRC Error when Adapter Retry was negotiated with remote Link partner. Default Value is 0b.
1	RW1CS	**Adapter LSM transition to Retrain**: Set to 1b by hardware if the Adapter LSM transitioned to Retrain state. Default Value is 0b.
2	RW1CS	**Correctable Internal Error**: Set to 1b by hardware if an internal correctable Data path error is detected. Examples of such errors include (but are not limited to) correctable error correcting code (ECC) error in the Retry buffer, Physical Layer indicated correctable error on RDI, etc. LinkError state transition on RDI and FDI must also be logged in this bit. Default Value is 0b.
3	RW1CS(RP/DSP/ Retimer), RsvdZ(Others)	**Sideband Correctable Error Message received**: Set to 1b by hardware if the Adapter received a Correctable {ErrMsg} sideband message with Device origin encoding in the message information. Default Value is 0b.
4	RW1CS	**'Runtime Link Testing Parity' Error**
31:5	RsvdZ	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.6 Correctable Error Mask Register (Offset 20h)

The Correctable Error Mask Register controls the reporting of individual errors. When a bit is 1b in this register, setting of the corresponding error status bit is not forwarded to the Protocol Layer for escalation/signaling.

Table 7-31. Correctable Error Mask Register

Bit	Attribute	Description
0	RWS	**CRC Error Detected Mask** Default Value is 1b.
1	RWS	**Adapter LSM transition to Retrain Mask** Default Value is 1b.
2	RWS	**Correctable Internal Error Mask** Default Value is 1b.
3	RWS	**Device Correctable Error Message received Mask** Default Value is 1b.
4	RWS	**'Runtime Link Testing Parity' Error Mask** Default Value is 1b.
31:5	RsvdP	Reserved

7.5.3.7 Header Log 1 Register (Offset 24h)

This register is used to log the header on sideband register accesses that receive UR/CA error status.

Table 7-32. Header Log 1 Register

Bit	Attribute	Description
63:0	ROS	**Header Log 1:** This logs the header for the first sideband mailbox register access that received a completion with Completer Abort status or received a completion with Unsupported Request. Note that register accesses that time out are not required to be logged at the requester. If the 'Received UR' and 'Received CA' status bits are both cleared in the 'Sideband Mailbox Status' register, this field's value is undefined. Default Value is 0.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.8 Header Log 2 Register (Offset 2Ch)

This register is used to log syndrome of various mainband and sideband errors and specific status logging on link training.

Table 7-33. Header Log 2 Register (Sheet 1 of 2)

Bit	Attribute	Description
3:0	ROS	**Adapter Timeout encoding**: Captures the reason for the first Adapter Timeout that was logged in Uncorrectable Error Status. Default Value is 0000b. The encodings are interpreted as follows: 0001b: Parameter Exchange flow timed out 0010b: Adapter LSM request to remote Link partner did not receive a response after 8ms. Bits 11:9 capture the specific state request that did not receive a response. Bit 12 of this register captures which Adapter LSM timed out. 0011b: Adapter LSM transition to Active timeout. This is recorded in case the Adapter never received Active Request from remote Link partner for 8ms after sending an Active Request on sideband even though it received an Active Response. Bit 12 of this register captures which Adapter LSM timed out. 0100b: Retry Timeout - no Ack or Nak received after 8ms, when Retry was enabled. Timeout counter is only incremented while RDI is in Active and Adapter's Retry buffer is not empty. 0101b: Local sideband access timeout 0110b: Retimer credit return timeout - no Retimer credit received for greater than 8ms if one or more Retimer credits have been consumed by the Adapter. This timer is only counting during Active state. If RDI moves to Retrain, this timer must be Reset since the Retimer credits are also Reset. 0111b: Remote Register Access timeout. This is triggered when if the Adapter has observed N timeouts for Register Accesses where N is >= register access timeout threshold. other encodings are reserved. If the Adapter Timeout status bit is cleared in the 'Uncorrectable Error Status' register, this field's value is undefined.
6:4	ROS	**Receiver Overflow encoding**: Captures the encoding for the first Receiver overflow error that occurred. Default value is 000b. The encodings are interpreted as follows: 001b: Transmitter Retry Buffer overflow 010b: Retimer Receiver Buffer overflow 011b: FDI sideband buffer overflow 100b: RDI sideband buffer overflow other encodings are reserved. If the Receiver overflow status bit is cleared in the 'Uncorrectable Error Status' register, this field's value is undefined.
9:7	ROS	**Adapter LSM response type:** 001b: Active 010b: L1 011b: L2 100b: LinkReset 101b: Disable Other encodings are reserved If the Adapter Timeout status bit is cleared in the 'Uncorrectable Error Status' register, this field's value is undefined.
10	ROS	**Adapter LSM id :** 0b : Adapter LSM 0 timed out 1b : Adapter LSM 1 timed out
12:11	RsvdZ	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-33. Header Log 2 Register (Sheet 2 of 2)

Bit	Attribute	Description
13	RO	**Parameter Exchange Successful** : Hardware updates this bit to 1b after successful Parameter exchange with remote Link partner, on every link training.
17:14	ROS	**Flit Format**: This field logs the negotiated Flit Format, it is the current snapshot of the format the Adapter is informing to the Protocol Layer. See Chapter 3.0 for the definitions of these formats. The encodings are: 0001b - Format 1 0010b - Format 2 0011b - Format 3 0100b - Format 4 0101b - Format 5 0110b - Format 6 Other encodings are Reserved
22:18	ROS	**First Fatal Error Indicator**: 5-bit encoding that indicates which bit of Uncorrectable Error Status errors was logged first. If no bits are 1b in the Uncorrectable Error Status register, then the value of this field has no meaning.
31:23	RsvdZ	Reserved

7.5.3.9 Error and Link Testing Control Register (0ffset 30h)

Table 7-34. Error and Link Testing Control Register (Sheet 1 of 2)

Bit	Attribute	Description
3:0	RW	**Remote Register Access Threshold**: Indicates the number of consecutive timeouts for remote register accesses that must occur before the Register Access timeout is logged and the error escalated to a Link_Status=Down condition. Default Value is 0100b.
4	RW	**Runtime Link Testing Tx Enable**: Software writes to this bit to enable Parity byte injections in the data stream as described in Section 3.9. Runtime Link Rx Enable must be set to 1b for remote Link Partner for successful enabling of this mode. Default Value is 0b.
5	RW	**Runtime Link Testing Rx Enable**: Software writes to this bit to enable Parity byte checking in the data stream as described in Section 3.9. Runtime Link Tx Enable must be set to 1b for remote Link Partner for successful enabling of this mode. Default Value is 0b.
8:6	RW	**Number of 64 Byte Inserts**: Software writes to this to indicate the number 64 Byte inserts are done at a time for Runtime Link Testing. Software programs this based on the current Link width (taking into account how many modules per Adapter are currently in operation). The encodings are: 000b: one 64B insert 001b: two 64B inserts 010b: four 64B inserts Other encodings are reserved. Default value is 000b.
9	RW1C	**Parity Feature Nak received**: Hardware updates this bit if it receives a Nak from remote Link partner when attempting to enable Runtime Link Testing.
12:10	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-34. Error and Link Testing Control Register (Sheet 2 of 2)

Bit	Attribute	Description
14:13	RW	**CRC Injection Enable** : Software writes to this bit to trigger CRC error injections, The error is injected by inverting 1, 2 or 3 bits in the CRC bytes. The specific bits inverted are implementation specific. The CRC injection must not happen for Flits that are already inverting CRC bits for Viral handling. The encodings are interpreted as : 00b : CRC Injection is Disabled. 01b : 1 bit is inverted 10b : 2 bits are inverted 11b : 3 bits are inverted. Default Value is 00b.
16:15	RW	**CRC Injection Count** : Software writes to this bit to program the number of CRC injections. It only takes effect if CRC injection Enable is not Disabled. 00b : Single Flit is corrupted. CRC Injection Busy is reset to 0b after single Flit corruption. 01b: A CRC error is injected every 8 Flits. Hardware continues to inject a CRC error every 8 Flits until CRC Injection Enable is 00b. CRC Injection Busy is reset to 0b only after CRC Injection Enable is 00b. 10b: A CRC error is injected every 16 Flits. Hardware continues to inject a CRC error every 16 Flits until CRC Injection Enable is 00b. CRC Injection Busy is reset to 0b only after CRC Injection Enable is 00b. 11b: A CRC error is injected every 64 Flits. Hardware continues to inject a CRC error every 64 Flits until CRC Injection Enable is 00b. CRC Injection Busy is reset to 0b only after CRC Injection Enable is 00b.
17	RO	**CRC Injection Busy** : Hardware loads a 1b to this bit once it has begun CRC Injection. Software is permitted to poll on this bit. See CRC Injection Count description to see how this bit returns to 0b.
31:18	RsvdP	Reserved

7.5.3.10 Runtime Link Testing Parity Log 0 (Offset 34h)

Table 7-35. Runtime Link Testing Parity Log 0 Register

Bit	Attribute	Description
63:0	RW1C	**Parity Log for Module 0**: Hardware updates the bit corresponding to the parity error byte with error over the period when Runtime Link Testing was enabled at Rx. Default Value is 0.

7.5.3.11 Runtime Link Testing Parity Log 1 (Offset 3Ch)

Table 7-36. Runtime Link Testing Parity Log 1 Register

Bit	Attribute	Description
63:0	RW1C	**Parity Log for Module 1**: Hardware updates the bit corresponding to the parity error byte with error over the period when Runtime Link Testing was enabled at Rx. Default Value is 0. This is register is only applicable if the Adapter is designed for handling two or more Physical Layer modules. It is reserved otherwise.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.12 Runtime Link Testing Parity Log 2 (Offset 44h)

Table 7-37. Runtime Link Testing Parity Log 2 Register

Bit	Attribute	Description
63:0	RW1C	**Parity Log for Module 2**: Hardware updates the bit corresponding to the parity error byte with error over the period when Runtime Link Testing was enabled at Rx. Default Value is 0. This is register is only applicable if the Adapter is designed for handling four Physical Layer modules. It is reserved otherwise.

7.5.3.13 Runtime Link Testing Parity Log 3 (Offset 4Ch)

Table 7-38. Runtime Link Testing Parity Log 3 Register

Bit	Attribute	Description
63:0	RW1C	**Parity Log for Module 3**: Hardware updates the bit corresponding to the parity error byte with error over the period when Runtime Link Testing was enabled at Rx. Default Value is 0. This is register is only applicable if the Adapter is designed for handling four Physical Layer modules. It is reserved otherwise.

7.5.3.14 Advertised Adapter Capability Log (Offset 54h)

Table 7-39. Advertised Adapter Capability Log Register

Bit	Attribute	Description
63:0	RW1C	**Advertised Adapter Capability**: Hardware updates the bits corresponding to the data bits it sent in the {AdvCap.Adapter} sideband message. Default Value is 0.

7.5.3.15 Finalized Adapter Capability Log (Offset 5Ch)

Table 7-40. Finalized Adapter Capability Log Register

Bit	Attribute	Description
63:0	RW1C	**Finalized Adapter Capability**: Hardware updates the bits corresponding to the data bits it sent (DP) or received (UP) in the {FinCap.Adapter} sideband message. Default Value is 0.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.16 Advertised CXL Capability Log (Offset 64h)

Table 7-41. Advertised CXL Capability Log Register

Bit	Attribute	Description
63:0	RW1C	**Advertised CXL Capability**: Hardware updates the bits corresponding to the data bits it sent in the {AdvCap.CXL} sideband message, when it is sent with MsgInfo=0000h. Default Value is 0.

7.5.3.17 Finalized CXL Capability Log (Offset 6Ch)

Table 7-42. Finalized CXL Capability Log Register

Bit	Attribute	Description
63:0	RW1C	**Finalized CXL Capability**: Hardware updates the bits corresponding to the data bits it sent (DP) or received (UP) in the {FinCap.CXL} sideband message, when it is sent with MsgInfo=0000h. Default Value is 0.

7.5.3.18 Advertised Multi-Protocol Capability Log Register (Offset 78h)

This register is reserved for designs that do not implement the Enhanced Multi-protocol capability.

Table 7-43. Advertised Multi-Protocol Capability Log Register

Bit	Attribute	Description
63:0	RW1C	**Advertised Multi-Protocol Capability**: Hardware updates the bits corresponding to the data bits it sent in the {MultiProtAdvCap.Adapter} sideband message. Default value is 0.

7.5.3.19 Finalized Multi-Protocol Capability Log Register (Offset 80h)

This register is reserved for designs that do not implement the Enhanced Multi-protocol capability.

Table 7-44. Finalized Multi-Protocol Capability Log Register

Bit	Attribute	Description
63:0	RW1C	**Finalized Multi-Protocol Capability**: Hardware updates the bits corresponding to the data bits it sent in the {MultiProtFinCap.Adapter} sideband message. Default value is 0.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.20 Advertised CXL Capability Log Register for Stack 1 (Offset 88h)

This register is reserved for designs that do not implement the Enhanced Multi-protocol capability.

Table 7-45. Advertised CXL Capability Log Register for Stack 1

Bit	Attribute	Description
63:0	RW1C	**Advertised CXL Capability**: Hardware updates the bits corresponding to the data bits it sent in the {AdvCap.CXL} sideband message when it is sent with MsgInfo=0001h. Default value is 0.

7.5.3.21 Finalized CXL Capability Log Register for Stack 1 (Offset 90h)

This register is reserved for designs not implementing the Enhanced multi-protocol capability.

Table 7-46. Finalized CXL Capability Log Register for Stack 1

Bit	Attribute	Description
63:0	RW1C	**Finalized CXL Capability**: Hardware updates the bits corresponding to the data bits it sent in the {FinCap.CXL} sideband message when it is sent with MsgInfo=0001h. Default value is 0.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.22 PHY Capability (Offset 1000h)

This register is global, and not per module.

Table 7-47. Physical Layer Capability Register

Bit	Attribute	Description
2:0	RO	**Module to Byte mapping** 00h: no Module reversal (LSB Module to LSB Module) 01h: MSB Module to LSB module connectivity Reserved
3	RO	**Terminated Link**: If set to 1b, it indicates that the Receiver is terminated.
4	RO	**TX Equalization support** 0: TXEQ not supported 1: TXEQ supported
10:5	RO	**Supported Vswing encodings** 01h: 0.4V 02h: 0.45V 03h: 0.5V 04h: 0.55V 05h: 0.6V 06h: 0.65V 07h: 0.7V 08h: 0.75V 09h: 0.8V 0Ah: 0.85V 0Bh: 0.9V 0Ch: 0.95V 0Dh: 1.0V 0Eh: 1.05 0Fh: 1.1V 10h: 1.15V Others Reserved
12:11	RO	**Rx Clock Mode support** 0h: Supports both free running and strobe modes 1h: Strobe mode only 2h: Free running mode only This is the Clock mode supported by a Rx that it advertises to its Tx partner during training.
14:13	RO	**Rx Clock phase support** 0h: Differential clock only 1h: Quadrature clock only 2h: Both differential and quadrature clock This is the Clock mode supported by a Rx that it advertises to its Tx partner during training.
15	RO	**Package type** 0b: Advanced Package 1b: Standard Package
16	RO	**Tightly coupled mode (TCM) support** 0b: TCM not Supported 1b: TCM supported
31:17	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.23 PHY Control (Offset 1004h)

This register is global, and not per module.

Table 7-48. Physical Layer Control Register

Bit	Attribute	Description
2:0	RW	**Module to Byte mapping** 00h: no Module reversal (LSB Module to LSB Module) 01h: MSB Module to LSB module connectivity Others: Reserved Module to Byte mapping hardware writes it post hardware training Default is 0h
3	RW	**Rx Terminated Control** 0b: Rx Termination disabled 1b: Rx Termination enabled Default is same as 'Terminated Link' bit in PHY capability register
4	RW	**Tx Eq Enable:** 0b: Eq Disabled 1b: Eq Enabled Default is 0
5	RW	**Clock Mode Select** 0h: Strobe Mode 1h: Free running mode Default is 0
6	RW	**Clock phase support select:** 0h: Differential clock 1h: Quadrature clock
7	RW/RsvdP	**Force x32 Width Mode in x64 Module** This bit is used only for test and debug purposes. In normal operation, this bit should be reset to 0. When set, this bit will force the x64 module to present "UCIe-A x32 bit =1" during MBINIT.PARAM exchange phase independent of the value of bit 20, APMW, in the UCIe Link Capability register. This bit applies to all modules in a multi-module link. For x32 Advanced Package modules, this bit is reserved.
31:8	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.24 PHY Status (Offset 1008h)

This register is global and not per module.

Table 7-49. Physical Layer Status Register

Bit	Attribute	Description
2:0	RO	**Module to Byte mapping** 00h: no Module reversal (LSB Module to LSB Module) 01h: MSB Module to LSB module connectivity Others: Reserved Module to Byte mapping hardware writes it post hardware training Default is 0h
3	RO	**Rx Terminated Status** 0b: Rx Termination disabled 1b: Rx Termination enabled Default is same as 'Terminated Link' bit in PHY capability register
4	RO	**Tx Eq Status:** 0b: Eq Disabled 1b: Eq Enabled Default is 0
5	RO	**Clock Mode Status:** 0h: Strobe Mode 1h: Free running mode Default is 0
6	RO	**Clock phase Status:** 0h: Differential clock 1h: Quadrature clock
7	RO	**Lane Reversal within Module:** Indicates if Lanes within a module are reversed 0b: Lanes within module not reversed 1b: Lanes within module are reversed
31:8	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.25 PHY Initialization and Debug (Offset 100Ch)

This register is global, and not per module.

Table 7-50. Phy Init and Debug Register

Bit	Attribute	Description
2:0	RW	**Initialization control** 000b: Initialize to Active. This is the regular Link bring up. 001b: Initialize to MBINIT (Debug mode) (i.e., pause training after completing step-2 of MBINIT.PARAM). 010b: Initialize to MBTRAIN (Debug/compliance mode) (i.e., pause training after entering MBTRAIN after completing step-1 of MBTRAIN.VALVREF). 011b = Pause after completing step-1 of MBTRAIN.RXDESKEW; regardless of entering for initial bring up or from Retrain. 100b = Pause after completing step-1 of MBTRAIN.DATATRAINCENTER2; regardless of entering for initial bring up or from Retrain. All other encodings are reserved. When training has paused, the corresponding state timeouts must be disabled, and hardware resumes training on any of the following triggers: • A 0b-to-1b transition on 'Resume Training' bit in this register • Sideband message for the corresponding state is received from remote link partner (e.g., if paused in MBINIT, receiving {MBINIT.CAL Done req} from remote link partner is also a trigger to move forward) A device that does not support the UCIe Test and Compliance register block is permitted to only implement encodings 000b through 010b. Default is 000b.
4:3	RsvdP	Reserved
5	RW	**Resume Training** A 0b-to-1b transition on this bit triggers hardware to resume training from the last link training state, achieved via 'Initialization Control' field in this register until ACTIVE. A device that does not support the UCIe Test and Compliance register block is permitted to hardwire this bit to 0b. Default is 0b.
31:6	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.26 Training Setup 1 (Offset 1010h)

This register is replicated per module. Offsets 1010h to 101Ch are used in 4B increments for multi-module scenarios

Table 7-51. Training Setup 1 Register

Bit	Attribute	Description
2:0	RW	**Data pattern used during training** 000b: Per-Lane LFSR pattern 001b: Per-Lane ID pattern 010b: If @PHY-Compliance {Per-Lane Clock pattern AA pattern} Else Reserved 011b: If @PHY-Compliance {Per-Lane all 0 pattern} Else Reserved 100b: If @PHY-Compliance {Per-Lane all 1 pattern} Else Reserved 101b: If {@PHY-Compliance Per-Lane inverted Clock pattern} Else Reserved All other encodings are reserved Default is 000b.
5:3	RW	**Valid Pattern used during training** 000b: Functional valid pattern (1111 0000 (lsb first)) All other encodings are reserved Default is 000b.
9:6	RW	**Clock Phase control** 0h: Clock PI center found by Transmitter 1h: Left edge found through Data to clock training 2h: Right edge found through Data to clock training All other encodings are reserved Default = 0
10	RW	**Training mode:** 0b: Continuous mode 1b: Burst Mode Default = 0
26:11	RW	**16 bit Burst Count**: Indicates the duration of selected pattern (UI count) Default = 4h
31:27	RsvdP	Reserved

7.5.3.27 Training Setup 2 (Offset 1020h)

This register is replicated per module. Offsets 1020h to 102Ch are used in 4B offset increments for multi-module scenarios.

Table 7-52. Training Setup 2 Register

Bit	Attribute	Description
15:0	RW	**16 bit idle count**: Indicates the duration of low following the burst (UI count) Default = 4h
31:16	RW	**16 bit iterations**: Indicates the iteration count of bursts followed by idle (UI count) Default = 4h

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.28 Training Setup 3 (Offset 1030h)

This register is replicated per module. Offsets 1030h to 1048h are used in 8B offset increments for multi-module scenarios.

Table 7-53. Training Setup 3 Register

Bit	Attribute	Description
63:0	RW	**Lane mask**: Indicated the Lanes to mask during Rx comparison. Example 1h = Lane 0 is masked during comparison Default = 0 (no mask)

7.5.3.29 Training Setup 4 (Offset 1050h)

This register is replicated per module. Offsets 1050h to 105Ch are used in 4B offset increments for multi-module scenarios.

Table 7-54. Training Setup 4 Register

Bit	Attribute	Description
3:0	RW	**Repair Lane mask**: Indicated the Redundant Lanes to mask during Rx comparison. Example 1h =RD0 is masked during comparison 2h: RD1 mask Default = 0 (no mask)
15:4	RW	**Max error Threshold in per Lane comparison**: Indicates threshold for error counting to start Default = 0 (all errors are counted)
31:16	RW	**Max error Threshold in aggregate comparison**: Indicates threshold for error counting to start Default = 0 (all errors are counted)

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.30 Current Lane Map Module 0 (Offset 1060h)

Table 7-55. Current Lane Map Module 0 Register

Bit	Attribute	Description
63:0	RW	**Current Rx Lane map (CLM) for Module-0**: If a bit is 1 it indicates the corresponding physical Lane is operational. For Standard package modules, bits 63:16 of this register are not applicable. For UCIe-A x32 implementations (i.e., APMW bit in UCIe Link Capability register is set), bits 63:32 of this register are not applicable. Default Value is all 0s.

7.5.3.31 Current Lane Map Module 1 (Offset 1068h)

Table 7-56. Current Lane Map Module 1 Register

Bit	Attribute	Description
63:0	RW	**Current Rx Lane map (CLM) for Module-1**: If a bit is 1 it indicates the corresponding physical Lane is operational. For Standard package modules, bits 63:16 of this register are not applicable. For UCIe-A x32 implementations (i.e., APMW bit in UCIe Link Capability register is set), bits 63:32 of this register are not applicable. Default Value is all 0s. This register is reserved if Module 1 is not present

7.5.3.32 Current Lane Map Module 2 (Offset 1070h)

Table 7-57. Current Lane Map Module 2 Register

Bit	Attribute	Description
63:0	RW/RsvdP	**Current Rx Lane map (CLM) for Module-2**: If a bit is 1 it indicates the corresponding physical Lane is operational. For Standard package modules, bits 63:16 of this register are not applicable. For UCIe-A x32 implementations (i.e., APMW bit in UCIe Link Capability register is set), bits 63:32 of this register are not applicable. Default Value is all 0s. This register is reserved if Module 2 is not present

7.5.3.33 Current Lane Map Module 3 (Offset 1078h)

Table 7-58. Current Lane Map Module 3 Register

Bit	Attribute	Description
63:0	RW/RsvdP	**Current Rx Lane map (CLM) for Module-3**: If a bit is 1 it indicates the corresponding physical Lane is operational. For Standard package modules, bits 63:16 of this register are not applicable. For UCIe-A x32 implementations (i.e., APMW bit in UCIe Link Capability register is set), bits 63:32 of this register are not applicable. Default Value is all 0s. This register is reserved if Module 3 is not present

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.34 Error Log 0 (Offset 1080h)

This register is replicated per module. Offsets 1080h to 108Ch are used in 4B offset increments for multi-module scenarios.

Table 7-59. Error Log 0 Register

Bit	Attribute	Description
7:0	ROS	**State N**: Captures the current Link training state machine status. State Encodings are given by: 00h RESET 01h SBINIT 02h MBINIT.PARAM 03h MBINIT.CAL 04h MBINIT.RepairCLK 05h MBINIT.RepairVAL 06h MBINIT.ReversalMB 07h MBINIT.RepairMB 08h MBTRAIN.VALVREF 09h MBTRAIN.DATAVREF 0Ah MBTRAIN.SPEEDIDLE 0Bh MBTRAIN.TXSELFCAL 0Ch MBTRAIN.RXSELFCAL 0Dh MBTRAIN.VALTRAINCENTER 0Eh MBTRAIN.VALTRAINVREF 0Fh MBTRAIN.DATATRAINCENTER1 10h MBTRAIN.DATATRAINVREF 11h MBTRAIN.RXDESKEW 12h MBTRAIN.DATATRAINCENTER2 13h MBTRAIN.LINKSPEED 14h MBTRAIN.REPAIR 15h PHYRETRAIN 16h LINKINIT 17h ACTIVE 18h TRAINERROR 19h L1/L2 All other encodings are reserved Default is 0
8	ROS	**Lane Reversal**: 1b indicates Lane Reversal within the module. Default is 0
9	ROS	**Width Degrade**: 1b indicates Module width Degrade. Applicable to Standard package only. Default is 0.
15:10	RsvdZ	Reserved
23:16	ROS	**State (N-1)**: Captures the state before State N was entered for Link training state machine. State encodings are the same as State N field. Default is 0
31:24	ROS	**State (N-2)**: Captures the state before State (N-1) was entered for Link training state machine. State encodings are the same as State N field. Default is 0

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.35 Error Log 1 (Offset 1090h)

This register is replicated per module. Offsets 1090h to 109Ch are used in 4B offset increments for multi-module scenarios.

Table 7-60. Error Log 1 Register

Bit	Attribute	Description
7:0	ROS	**State (N-3)**: Captures the state status before State (N-2) was entered. State encodings are the same as State N field. Default is 0
8	RW1CS	**State Timeout Occurred**: Hardware sets this to 1b if a Link Training State machine state or sub-state timed out and it was escalated as a fatal error. Default value is 0b.
9	RW1CS	**Sideband Timeout Occurred**: Hardware sets this to 1b if a sideband handshake timed out, for example, if a RDI request did not get a response for 8ms. Sideband handshakes related to Link Training messages are not included here. Default value is 0b.
10	RW1CS	**Remote LinkError received**: Hardware sets this to 1b if remote Link partner requested LinkError transition through RDI sideband. Default value is 0b.
11	RW1CS	**Internal Error**: Hardware sets this to 1b if any implementation specific internal error occurred in the Physical Layer. Default value is 0b.
31:12	RsvdZ	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.36 Runtime Link Test Control (Offset 1100h)

Table 7-61. Runtime Link Test Control (Sheet 1 of 2)

Bit	Attribute	Description
0	RW	**Apply Valid Repair**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Valid repair in the next Retrain cycle. This bit should have no effect for hardware on Standard Package. Default is 0
1	RW	**Apply Clock Repair**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Clock repair in the next Retrain cycle. This bit should have no effect for hardware on Standard Package. Default is 0
2	RW	**Apply Module 0 Lane Repair**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical module id at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 0, if possible and relevant. Default value is 0b
3	RW	**Apply Module 1 Lane Repair**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical module id at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 1, if possible and relevant. Default value is 0b These bits are reserved if Module 1 is not present.
4	RW	**Apply Module 2 Lane Repair**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical module id at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 2, if possible and relevant. Default value is 0b These bits are reserved if Module 2 is not present.
5	RW	**Apply Module 3 Lane Repair**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical module id at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 3, if possible and relevant. Default value is 0b These bits are reserved if Module 3 is not present.
6	RW	**Start**: Software writes to this bit before hitting Link Retrain bit to inform hardware that the contents of this register are valid.
7	RW	**Inject Stuck-at fault**: Software writes 1b to this bit to indicate hardware must inject a stuck at fault for the Lane id identified in Lane Repair id for the corresponding field. Injecting the fault at Tx or Rx is implementation specific. Default value is 0b.
14:8	RW	**Module 0 Lane repair id**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical transmit Lane id in logical Module 0 at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 0, if possible and relevant. Default is 0. These bits are reserved if Module 0 is not present.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-61. Runtime Link Test Control (Sheet 2 of 2)

Bit	Attribute	Description
21:15	RW	**Module 1 Lane repair id**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical transmit Lane id in logical Module 1 at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 1, if possible and relevant. Default is 0. These bits are reserved if Module 1 is not present.
28:22	RW	**Module 2 Lane repair id**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical transmit Lane id in logical Module 2 at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 2, if possible and relevant. Default is 0. These bits are reserved if Module 2 is not present.
35:29	RW	**Module 3 Lane repair id**: For Advanced Package, software programs this bit to inform Physical Layer hardware to apply Lane repair for this logical transmit Lane id in logical Module 3 at the next Retrain cycle, if this Module is operational. For Standard Package, this bit will trigger a width degrade for logical Module 3, if possible and relevant. Default is 0. These bits are reserved if Module 3 is not present.
63:36	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.37 Runtime Link Test Status (Offset 1108h)

Table 7-62. Runtime Link Test Status Register

Bit	Attribute	Description
0	RO	**Busy**: Hardware loads 1b to this bit once Start bit is written by software. Hardware loads 0b to this bit once it has attempted to complete the actions requested in Runtime Link Test Control register. Default is 0
31:1	RsvdZ	Reserved

7.5.3.38 Mainband Data Repair (Offset 110Ch)

This register is replicated per advanced module. For Standard package, this register is not applicable. Offsets 110Ch to 1124h are used in 8B offset increments for multi-module scenarios.

Table 7-63. Mainband Data Repair Register (Sheet 1 of 2)

Bit	Attribute	Description
7:0	RO	**Repair Address for TRD_P[0]**: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme 00h: TD_P[0] Repaired 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired ……. 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Unrepairable FFh: No Repair
15:8	RO	**Repair Address for TRD_P[1]**: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme 00h: Invalid 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired ……. 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Unrepairable FFh: No Repair
23:16	RO	**Repair Address for TRD_P[2]**: Indicates the physical Lane repaired when TRD_P[2] is used in remapping scheme 20h: TD_P[32] Repaired 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired ……. 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Unrepairable FFh: No Repair This field is reserved for UCIe-A x32 module implementations.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-63. Mainband Data Repair Register (Sheet 2 of 2)

Bit	Attribute	Description
31:24	RO	**Repair Address for TRD_P[3]**: Indicates the physical Lane repaired when TRD_P[3] is used in remapping scheme 20h: Invalid 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired …… 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Unrepairable FFh: No Repair This field is reserved for UCIe-A x32 module implementations.
39:32	RO	**Repair Address for RRD_P[0]**: Indicates the physical Lane repaired when RRD_P[0] is used in remapping scheme 00h: RD_P[0] Repaired 01h: RD_P[1] Repaired 02h: RD_P[2] Repaired …… 1Eh: RD_P[30] Repaired 1Fh: RD_P[31] Repaired F0h: Unrepairable FFh: No Repair
47:40	RO	**Repair Address for RRD_P[1]**: Indicates the physical Lane repaired when RRD_P[1] is used in remapping scheme 00h: Invalid 01h: RD_P[1] Repaired 02h: RD_P[2] Repaired …… 1Eh: RD_P[30] Repaired 1Fh: RD_P[31] Repaired F0h: Unrepairable FFh: No Repair
55:48	RO	**Repair Address for RRD_P[2]**: Indicates the physical Lane repaired when RRD_P[2] is used in remapping scheme 20h: RD_P[32] Repaired 21h: RD_P[33] Repaired 22h: RD_P[34] Repaired …… 3Eh: RD_P[62] Repaired 3Fh: RD_P[63] Repaired F0h: Unrepairable FFh: No Repair This field is reserved for UCIe-A x32 implementations.
63:56	RO	**Repair Address for RRD_P[3]**: Indicates the physical Lane repaired when RRD_P[3] is used in remapping scheme 20h: Invalid 21h: RD_P[33] Repaired 22h: RD_P[34] Repaired …… 3Eh: RD_P[62] Repaired 3Fh: RD_P[63] Repaired F0h: Unrepairable FFh: No Repair This field is reserved for UCIe-A x32 module implementations.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.39 Clock, Track, Valid and Sideband Repair (Offset 1134h)

This register is replicated per module. Offsets 1134h to 1140h are used in 4B offset increments for multi-module scenarios.

Table 7-64. Clock, Track, Valid and Sideband Repair Register

Bit	Attribute	Description
3:0	RO	**Repair Address for TRDCK_P**: Indicates the physical Lane repaired when **TRDCK_P** is used in remapping scheme 0h: **TCKP_P** Repaired 1h: **TCKN_P** Repaired 2h: **TTRK_P** Repaired 3h to 6h: Reserved 7h: Unrepairable Fh: No Repair
7:4	RO	**Repair Address for RRDCK_P**: Indicates the physical Lane repaired when **RRDCK_P** is used in remapping scheme 0h: **RCKP_P** Repaired 1h: **RCKN_P** Repaired 2h: **RTRK_P** Repaired 3h to 6h: Reserved 7h: Unrepairable Fh: No Repair
9:8	RO	**Repair Address for TRDVLD_P**: Indicates the physical Lane repaired when **TRDVLD_P** is used in remapping scheme 0h: **TVLD_P** Repaired 1h: Unrepairable 2h: Reserved 3h: No Repair
11:10	RO	**Repair Address for RRDVLD_P**: Indicates the physical Lane repaired when **RRDVLD_P** is used in remapping scheme 0h: **RVLD_P** Repaired 1h: Unrepairable 2h: Reserved 3h: No Repair
15:12	RsvdP	Reserved
19:16	RO	**Repair Address for Sideband Transmitter**: Indicates sideband repair result for the Transmitter Result[3:0]
23:20	RO	**Repair Address for Sideband Receiver**: Indicates sideband repair result for the Transmitter Result[3:0]
31:24	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.40 UCIe Link Health Monitor (UHM) DVSEC

This DVSEC is an extended Capability. It is required for all devices that support Compliance testing (as indicated by the presence of Compliance/Test Register Locator) and optional otherwise. This DVSEC contains the required registers for SW to read eye margin values per lane. SW Flow for Eye Margining is as follows:

- SW ensures that the Eye Margin Valid (EMV) bit in UHM_STS register is cleared
- SW triggers a retrain of the link
  - When the retrain completes (as indicated by bit 16 in the UCIe Link Status register) and the EMV bit is set in the UHM_STS register, SW can read the EM*_Ln*_Mod* registers in UHM DVSEC to know the margins. Receive margins are logged in the Tx UHM registers.

Note that HW may also measure Eye Margins during HW-autonomous retraining and/or initial training and if measured, is permitted to report it in the Eye Margin registers whenever the EMV bit is cleared.

Figure 7-5. UCIe Link Health Monitor (UHM) DVSEC

PCI Express Extended Capability Header			
Designated Vendor Specific Header 1			
Reserved		Designated Vendor Specific Header 2	
UHM_STS		Reserved	
Reserved			
Reserved			
EMR_Ln1_Mod0	EML_Ln1_Mod0	EMR_Ln0_Mod0	EML_Ln0_Mod0
EMR_Ln3_Mod0	EML_Ln3_Mod0	EMR_Ln2_Mod0	EML_Ln2_Mod0
…			
…			
EMR_Ln1_Mod1	EML_Ln1_Mod1	EMR_Ln0_Mod1	EML_Ln0_Mod1
EMR_Ln3_Mod1	EML_Ln3_Mod1	EMR_Ln2_Mod1	EML_Ln2_Mod1
…			
…			

Table 7-65. UHM DVSEC - Designated Vendor Specific Header 1, 2

Register	Field	Bit Location	Value
Designated Vendor-Specific Header 1 (offset 04h)	DVSEC Vendor ID	15:0	D2DEh
	DVSEC Revision	19:16	0h
	Length	31:20	Design dependent
Designated Vendor-Specific Header 2 (offset 08h)	DVSEC ID	15:0	1h

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.3.40.1 UHM Status (Offset Eh)

Table 7-66. UHM Status

Bit	Attribute	Description
7:0	RO	**Step Count** Step count used in the reporting of margin information. A value of 0 indicates 256. For example, a value of 32 indicates that the UI is equally divided into 32 steps and Eye Margin registers provide the left and right margins in multiples of UI/32.
8	RW1C	**Eye Margin Valid (EMV)** This bit, when set, indicates that margin registers carry valid information from the last retrain. SW must clear this bit before initiating link retrain, if it intends to measure eye margins during the retrain. On a SW-initiated link retrain, if after retrain, this bit is cleared, then SW should infer that there was some error in margin measurement. Note that HW logs any new Eye Margin measurements (whether it is measured during SW-initiated retrain, during HW-autonomous retraining, or during initial training) in the Eye Margin registers only when this bit is cleared.
15:9	RsvdP	Reserved

7.5.3.40.2 Eye Margin (Starting Offset 10h)

Table 7-67. EML_Lnx_Mody

Bit	Attribute	Description
7:0	RO	**Eye Margin Left for Lane x and Module y** Provides the left eye margin relative to the PI center, in units of UI/Step Count.

Table 7-68. EMR_Lnx_Mody

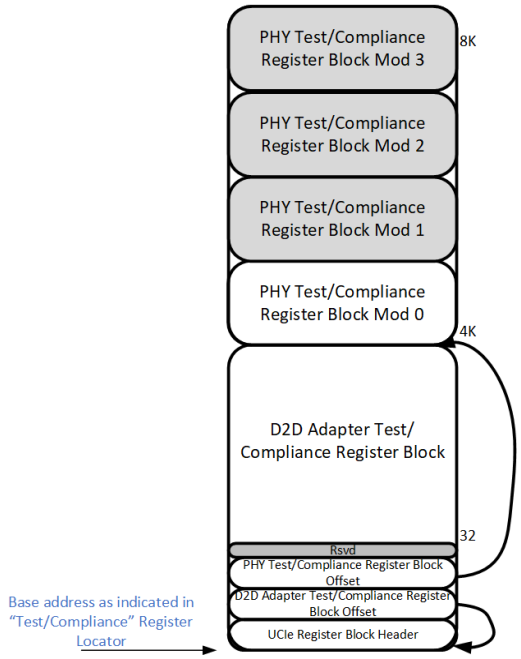
Bit	Attribute	Description
7:0	RO	**Eye Margin Right for Lane x and Module y** Provides the right eye margin relative to the PI center, in units of UI/Step Count.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4 Test/Compliance Register Block

The Test/Compliance register block is 8 KB in size with first 4 KB from base address (as enumerated via register locator with Register Block Identifier of 1h) used for D2D Adapter-related Test/Compliance registers and the second 4 KB used for PHY-related Test/Compliance registers. For future extensibility, these offsets are enumerable via the associated Register Block Offset registers, as shown in Figure 7-6.

Figure 7-6. UCIe Test/Compliance Register Block



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.1 UCIe Register Block Header

Table 7-69. UCIe Register Block Header (Offset 0h)

Bit	Attributes	Description
15:0	RO	**Vendor ID** Default is set to Vendor ID assigned for UCIe Consortium - D2DEh.
31:16	RO	**Vendor ID Register Block** Set to 1h to indicate Test Compliance register block.
35:32	RO	**Vendor Register Block Version** Set to 0h.
63:36	RsvdP	Reserved
95:64	RO	**Vendor Register Block Length** The number of bytes in the register block including the UCIe register block header. Default is 2000h.
127:96	RsvdP	Reserved

7.5.4.2 D2D Adapter Test/Compliance Register Block Offset

Table 7-70. D2D Adapter Test/Compliance Register Block Offset (Offset 10h)

Bit	Attributes	Description
7:0	RO	**D2D Adapter Test/Compliance Register Block Offset (D2DOFF)** 4-KB granular offset from Test/Compliance Register Block base address for D2D Adapter Test/Compliance registers. This field should be set to 0. However, SW must read this field to know the actual offset, for future compatibility reasons.
15:8	RO	**D2D Adapter Test/Compliance Register Block Length** 4-KB granular length of the D2D Adapter Test/Compliance registers. This field should be set to 1 to indicate 4-KB length. However, SW must read this field to know the actual length, for future compatibility reasons.
31:16	RsvdP	Reserved

7.5.4.3 PHY Test/Compliance Register Block Offset

Table 7-71. PHY Test/Compliance Register Block Offset (Offset 14h)

Bit	Attributes	Description
7:0	RO	**PHY Test/Compliance Register Block Offset (PHYOFF)** 4-KB granular offset from Test/Compliance Register Block base address for PHY Adapter Test/Compliance registers. This field should be set to 1, indicating that the registers start at 4 KB from the base address. However, SW must read this field to know the actual offset, for future compatibility reasons.
15:8	RO	**PHY Test/Compliance Register Block Length** 4-KB granular length of the PHY Test/Compliance registers. This field should be set to 1 to indicate 4-KB length. However, SW must read this field to know the actual length, for future compatibility reasons.
31:16	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.4 D2D Adapter Test/Compliance Register Block

7.5.4.4.1 Adapter Compliance Control

Table 7-72. Adapter Compliance Control (Offset 20h from D2DOFF)

Bit	Attributes	Description
1:0	RW	**Compliance Mode** Any write to this register takes effect after the next entry of RDI state status to Retrain. • 00b = Normal mode of operation • 01b = PHY only Link Training or Retraining — Adapter performs the necessary RDI handshakes to bring RDI to Active but does not perform Parameter exchanges or Adapter vLSM handshakes and keeps FDI in Reset to prevent mainband traffic. — Adapter must still trigger RDI to Retrain if software programmed the Retrain bit in Link Control. — Sideband Register Access requests and completions are operational in this mode. • 10b = Adapter Compliance — Adapter performs the necessary RDI handshakes to bring RDI to Active but does not perform Parameter exchanges or Adapter vLSM handshakes (unless triggered by software) and keeps FDI in Reset. — Adapter only performs actions based on the triggers and setup according to the registers defined in Section 7.5.4.4.2 to Section 7.5.4.4.6. — Adapter must still trigger RDI to Retrain if software programmed the Retrain bit in Link Control. — Sideband Register Access requests and completions are operational in this mode. • 11b = Reserved Any RDI transition to LINKERROR when this field is either 01b or 10b does not reset any registers. Default is 00b.
2	RW	**Force Link Reset** If set to 1b, Adapter transitions RDI to LinkError state. This bit is used by Compliance software to re-initialize the DUT anytime during Compliance testing. If SW expectation is that the DUT reinitializes to normal mode at the end of link reset, the Compliance Mode field in this register must be 00b and the Compliance Enable for PHY bit in the PHY Compliance Control Register must be 0b.
31:3	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.4.2 Flit Tx Injection Control

Table 7-73. Flit Tx Injection Control (Offset 28h from D2DOFF) (Sheet 1 of 2)

Bit	Attributes	Description
0	RW	**Flit Tx Injection Enable** Setting this bit to 1b starts Flit injection from the Adapter to the PHY at the Transmitter. Clearing this bit to 0b stops Flit injection on the Link. Default is 0b.
3:1	RW	**Flit Type** Type of Flit injected. • 000b = Adapter NOP Flits. These bypass TX retry buffer. • 001b = Test Flits. • 010b = Alternate between NOP Flits and Test Flits. • All other encodings are reserved. Default is 000b.
5:4	RW	**Injection mode** • 00b = Continuous injection of Flits as specified by Flit Type field. • 01b = Inject 'Flit Inject Number' of Flits contiguously without any intervening Protocol Flits. • 10b = Inject 'Flit Inject Number' of Flits while interleaving with Protocol Flits. If Protocol Flits are available, alternate between Protocol Flits and Injected Flits. If no Protocol Flits are available then, inject consecutively. • 11b = Reserved. Default is 00b.
13:6	RW	**Flit Inject Number** If the Injection mode is not 00b, this field indicates the number of Flits injected. Default is 00h.
17:14	RW	**Payload Type** This field determines the payload type used if Test Flits are injected. Payload includes all bits in the Flit with the exception of Flit Header, CRC, and Reserved bits. • 0h = Fixed 4B pattern picked up from 'Payload Fixed Pattern' field of this register, inserted so as to cover all the Payload bytes (with the same pattern replicated in incrementing 4B chunks) • 1h = Random 4B pattern picked up from a 32b LFSR (linear feedback shift register used for pseudo random pattern generation), inserted so as to cover all the Payload bytes (with the same pattern replicated in incrementing 4B chunks) • 2h = Fixed 4 byte pattern picked up from 'Payload Fixed Pattern' field of this register, inserted once at the 'Flit Byte Offset' location within the Flit • 3h = Random 4B pattern picked up from a 32b LFSR, inserted once at the 'Flit Byte Offset' location within the Flit and the rest of the payload is assigned 0b • 4h = Same as 2h, except the 4B pattern is injected every 'Pattern Repetition' bytes starting with 'Flit Byte Offset' • 5h = Same as 3h, except the 4B pattern is injected every 'Pattern Repetition' bytes starting with 'Flit Byte Offset' and the rest of the payload is assigned 0b • All other encodings are reserved Default is 0h. LFSR seed and primitive polynomial choice is implementation specific. **Note:** While in mission mode, because scrambling is always enabled, changing the Payload Type may have no benefit. This may, however, be useful during compliance testing with scrambling disabled.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-73. Flit Tx Injection Control (Offset 28h from D2DOFF) (Sheet 2 of 2)

Bit	Attributes	Description
25:18	RW	**Flit Byte Offset** See 'Payload Type'. Default is 00h.
31:26	RW	**Pattern Repetition** See 'Payload Type'. A value of 00h or 01h must be interpreted as a single pattern occurrence. Default is 00h.
63:32	RW	**Payload Fixed Pattern** See 'Payload Type'. Default is 0000 0000h.

7.5.4.4.3 Adapter Test Status (Offset 30h from D2DOFF)

Table 7-74. Adapter Test Status (Sheet 1 of 2)

Bit	Attributes	Description
0	RO	**Compliance Status** If Adapter is in 'PHY only Link Training or Retraining' or 'Adapter Compliance' mode, it is set to 1b; otherwise, it is 0b.
2:1	RO	**Flit Tx Injection Status** • 00b = No Flits injected. • 01b = At least one Flit was injected, but not completed. For Continuous Injection mode, this will be the status until Flit Injection Enable transitions from 1b to 0b. • 10b = Completed Flit Injection, for cases in which a finite number of Flit injections was set up. • 11b = Flit Injection Enable transitioned from 1b to 0b before Flit injections were complete. This field is cleared to 00b on a 0b-to-1b transition of Flit Injection Enable bit. Default is 00b.
4:3	RW1C	**Flit Rx Status** • 00b = No Test Flits received • 01b = Received at least one Test Flit without CRC error • All other encodings are reserved Default is 00b.
5	RO	**Link State Request Injection Status for Stack** • 0b = No request injected • 1b = Completed Request Injection This bit is cleared to 0b on a 0b-to-1b transition of 'Link State Request or Response Injection Enable'.
6	RO	**Link State Response Injection Status for Stack** • 0b = No response injected • 1b = Completed Response Injection This bit is cleared to 0b on a 0b-to-1b transition of 'Link State Request or Response Injection Enable'.
7	RO	**Link State Request Injection Status for Stack** • 0b = No request injected • 1b = Completed Request Injection This bit is cleared to 0b on a 0b-to-1b transition of 'Link State Request or Response Injection Enable'.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-74. Adapter Test Status (Sheet 2 of 2)

Bit	Attributes	Description
8	RO	**Link State Response Injection Status for Stack** • 0b = No response injected • 1b = Completed Response Injection This bit is cleared to 0b on a 0b-to-1b transition of 'Link State Request or Response Injection Enable'.
10:9	RO	**Retry Injection Status** • 00b = No errors injected on Transmitted Flits • 01b = Injected error on at least one transmitted Flit • 10b = Finished error injection sequence on transmitted Flits • 11b = Reserved This field is cleared to 00b on a 0b-to-1b transition of 'Retry Injection Enable'.
11	RO	**Number of Retries Exceeded Threshold** Set to 1b if the number of independent retry events exceed the threshold defined in 'Tx Retry Error Threshold'. This bit is cleared to 0b on a 0b-to-1b transition of 'Retry Injection Enable'.
31:12	RsvdZ	Reserved

7.5.4.4.4 Link State Injection Control Stack 0 (Offset 34h from D2DOFF)

As mentioned in Section 9.2, this register only takes effect when the Adapter is in Adapter Compliance Mode.

Table 7-75. Link State Injection Control Stack 0

Bit	Attributes	Description
0	RW	**Link State Request or Response Injection Enable at Tx** • 0b = Link State Request or Response Injection not enabled at Tx • 1b = Link State Request or Response Injection enabled at Tx
1	RW	**Injection Type** • 0b = Inject a request packet with the request matching "Link Request" field • 1b = Inject a response packet with the response matching "Link Response" field when a request matching "Link Request" field is received
5:2	RW	**Link Request** The encodings match the State request encodings of FDI.
9:6	RW	**Link Response** The encodings match the State response encodings of FDI.
31:10	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.4.5 Link State Injection Control Stack 1 (Offset 38h from D2DOFF)

As mentioned in Section 9.2, this register only takes effect when the Adapter is in Adapter Compliance Mode.

Table 7-76. Link State Injection Control Stack 1

Bit	Attributes	Description
0	RW	**Link State Request or Response Injection Enable at Tx** • 0b = Link State Request or Response Injection not enabled at Tx • 1b = Link State Request or Response Injection enabled at Tx
1	RW	**Injection Type** • 0b = Inject a request packet with the request matching "Link Request" field • 1b = Inject a response packet with the response matching "Link Response" field when a request matching "Link Request" field is received
5:2	RW	**Link Request** The encodings match the State request encodings of FDI.
9:6	RW	**Link Response** The encodings match the State response encodings of FDI.
31:10	RsvdP	Reserved

7.5.4.4.6 Retry Injection Control (Offset 40h from D2DOFF)

Table 7-77. Retry Injection Control (Sheet 1 of 2)

Bit	Attributes	Description
0	RW	**Retry Injection Enable** Setting this bit to 1b enables and starts error injections at Tx to force Retry on the UCIe Link. Clearing this bit to 0b stops Flit injection on the Link. Default is 0b.
3:1	RW	**Error Injection Type on Transmitted Flits** • 000b = No errors injected on Transmitted Flits • 001b = 1-bit error injected in 'Byte Offset' of the Flit, it is permitted to invert any bit in the corresponding byte position • 010b = 2-bit error injected in 'Byte Offset' of the Flit, it is permitted to invert any two bits in the corresponding byte position • 011b = 3-bit error injected in 'Byte Offset' of the Flit, it is permitted to invert any three bits in the corresponding byte position • All other encodings are reserved Default is 000b.
11:4	RW	**Byte Offset** See 'Error Injection Type on Transmitted Flits'. 00h means error is injected on Byte 0, 01h means error is injected in Byte 1, and so on. Default is 00h.
19:12	RW	**Number of Flits between Injected Errors** A nonzero value indicates the exact number of Flits after which a subsequent error is injected. A value of 0 will inject errors after a pseudo-random number of Flits between 1 and 31, chosen from a 32b LFSR output. Default is 00h.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-77. Retry Injection Control (Sheet 2 of 2)

Bit	Attributes	Description
27:20	RW	**Number of Errors Injected** Represents the number of errors injected on the Transmitted Flits. A value of 0 indicates that the error injection continues until the Retry Injection Enable is disabled. Default is 00h.
29:28	RW	**Flit Type for Error Injection** • 000b = Inject errors on any Flit type. • 001b = Only inject errors on NOP Flits. • 010b = Only inject errors on Payload Flits (Protocol Flits or Test Flits). • 011b = Only inject errors on Test Flits. • 100b = Only inject errors on Payload Flits. Subsequent errors injected on the same sequence number ('Number of Flits between Injected Errors' is ignored for this case). **Note:** The 100b value can be used to test Replay number Rollover rules. • All other encodings are reserved Default value is 00h.
31:30	RsvdP	Reserved
35:32	RW	**Tx Retry Error Threshold** If the number of independent retry events exceeds this threshold, Adapter must log this in 'Number of Retries Exceeded Threshold' and trigger Retrain on RDI. RDI state status going to Retrain also clears the internal count of independent retry events. Default value is 0h.
63:36	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.5 PHY Test/Compliance Register Block

Certain register bits described in this section take effect only when the PHY enters "PHY Compliance" mode. This mode is entered when bit 0 of 'Physical Layer Compliance Control 1' register is written and PHY subsequently enters PHYRETRAIN state. The latter happens when SW retrains the link. These register bits are tagged with @PHY-Compliance for easy readability and intuitive understanding.

Transition to TRAINERROR @PHY-Compliance does not reset any registers.

SW is required to place the Adapter in one of the Compliance modes (defined in the Adapter Compliance Control register) before enabling @PHY-Compliance.

All modules of a Link must be in @PHY-Compliance at the same time. The Link behavior is undefined if a subset of modules of a Link are in @PHY-Compliance and others are not. All registers in this section are replicated, one per module, as follows:

- Module 0 registers start at Offset 000h from PHYOFF
- Module 1 registers start at Offset 400h from PHYOFF
- Module 2 registers start at Offset 800h from PHYOFF
- Module 3 registers start at Offset C00h from PHYOFF

If certain modules are not implemented, those registers become reserved (as shown with gray boxes in Figure 7-6).

7.5.4.5.1 Physical Layer Compliance Control 1 (Offsets 000h, 400h, 800h, and C00h from PHYOFF)

Table 7-78. Physical Layer Compliance Control 1 (Sheet 1 of 2)

Bit	Attributes	Description
0	RW	**Compliance Enable for Physical Layer** Setting this bit to 1b puts the Physical Layer in "PHY Compliance" on the next entry into PHYRETRAIN state. Even if RDI status moves to Active, it does not assert **`pl_trdy`** to the Adapter in this mode. Default is 0b.
1	RW	**Scrambling Disabled** @PHY-Compliance, when set to 1b, Physical Layer disables scrambling. Default is 0b.
2	RW	**PHY Compliance Operation Trigger** @PHY-Compliance, transitioning this bit from 0b-to-1b starts one iteration of the Link training basic operations set by 'PHY Compliance Operation Type'. 'PHY Compliance Operation Type' field identifies which of the Link training basic operations is performed. 'Training Setup 1', 'Training Setup 2', 'Training Setup 3', and 'Training Setup 4' registers determine the parameters to be used for this. Default is 0b.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-78. Physical Layer Compliance Control 1 (Sheet 2 of 2)

Bit	Attributes	Description
5:3	RW	**PHY Compliance Operation Type** @PHY-Compliance, where the Link training basic operation (see Section 4.5.1) is performed when 'PHY Compliance Operation Trigger' transitions from 0b to 1b • 000b = No operation • 001b = Transmitter initiated Data-to-Clock point test (see Section 4.5.1.1) • 010b = Transmitter initiated Data-to-Clock eye width sweep (see Section 4.5.1.2) • 011b = Receiver initiated Data-to-Clock point training (see Section 4.5.1.3) • 100b = Receiver initiated Data-to-Clock width sweep training (see Section 4.5.1.4) • All other encodings are reserved
7:6	RsvdP	Reserved
9:8	RW	**Rx Vref Offset Enable** @PHY-Compliance: • 00b = No change to trained Rx Vref value • 01b = Add Rx Vref offset to trained Rx Vref value (up to maximum permitted Vref value) • 10b = Subtract Rx Vref offset to trained Rx Vref value (down to minimum permitted Rx Vref, any negative value to be terminated at 0) • 11b = Reserved
17:10	RW	**Rx Vref Offset** @PHY-Compliance, when 'Rx Vref Offset Enable' is set to 01b or 10b, this is the value that needs to be added or subtracted as defined in 'Rx Vref Offset Enable'. The Rx Vref value, after applying the Rx Vref offset, is expected to be monotonically increasing/decreasing with increasing/decreasing values of Rx Vref offset relative to the trained value and must have sufficient range to cover the input eye mask range defined in Chapter 5.0. Rx Vref Offset will be applied during Tx or Rx Data to Point Training and the Physical Layer must compare the per Lane errors with 'Max error Threshold in per-Lane comparison', and aggregate Lane errors with 'Max Error Threshold in Aggregate Comparison' in the 'Training Setup 4' register. If the errors measured are greater than the corresponding threshold, then the device must set the Rx Vref offset status register to "failed". Software must increase or decrease the Rx Vref Offset by one from the previous value. Default is 00h.
63:18	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.5.2 Physical Layer Compliance Control 2 (Offsets 008h, 408h, 808h, and C08h from PHYOFF)

Table 7-79. Physical Layer Compliance Control 2

Bit	Attributes	Description
0	RW	**Even UI Compare Mask** @PHY-Compliance, if this bit is set, any compare results for even UIs are masked (i.e., not counted toward error in per Lane or aggregate comparison (see Section 4.4)), where Even UI refers as to a Unit Interval data eye, the first data UI and every subsequent alternate UI. • 0b = No even UI compare result masking • 1b = Even UI compare result masked Default is 0b.
1	RW	**Odd UI Compare Mask** @PHY-Compliance, if this bit is set, any compare results for odd UIs are masked (i.e., not counted toward error in per Lane or aggregate comparison (see Section 4.4)), where Odd UI refers as to a Unit Interval data eye, the second data UI and every subsequent alternate UI). • 0b = No odd UI compare result masking • 1b = Odd UI compare results masked Default is 0b.

2	RW	**Track Enable**
		If @PHY-Compliance {
		If this bit is set, Track Transmission is enabled during one of the operations set by 'PHY compliance operation type'. Track transmission complies with descriptions in Section 5.5.1.
		}
		Else {
		The appropriate sideband handshakes as described in Section 4.6 needs to be followed irrespective of the value of this bit
		}

3	RW	**Compare Setup** • 0b = Aggregate comparison • 1b = Per Lane comparison Default is 0b. See Section 4.4 for more details.
31:4	RsvdP	Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.5.3 Physical Layer Compliance Status 1 (Offsets 010h, 410h, 810h, and C10h from PHYOFF)

Table 7-80. Physical Layer Compliance Status 1

Bit	Attributes	Description
0	RO	**PHY in Compliance mode** If (@PHY-Compliance) 1b. Else 0b.

1	RO	**PHY Compliance operation status**
		If (@PHY-Compliance) {
		This bit is set to 1b if 'PHY compliance operation type' in 'Physical Layer Compliance Control 1' register is 001b, 010b, 011b, or 100b and hardware has performed the required operation.
		Else the bit is cleared to 0b.
		}

3:2	RW1C	**Rx Vref Offset Operation Status** @PHY-Compliance: • 00b = Device does not support applying any Rx Vref Offset value • 01b = 'Rx Vref Offset' has not been applied • 10b = Rx Vref Offset has been successfully applied • 11b = Did not apply 'Rx Vref Offset' as the resulting value exceeds the value supported by hardware Default is 00b.
31:4	RsvdZ	Reserved

7.5.4.5.4 Physical Layer Compliance Status 2 (Offsets 018h, 418h, 818h, and C18h from PHYOFF)

Table 7-81. Physical Layer Compliance Status 2

Bit	Attributes	Description
31:0	RW1C	**Aggregate Error Count** @PHY-Compliance, this is the Error count of aggregate error comparison when 'PHY Compliance Operation Type' is 001b or 011b (performing point tests). Default is 0000 0000h.
39:32	RO	**Supported Rx Vref Range Up** Max step count supported up from the trained Rx Vref value for Vref margining.
47:40	RO	**Supported Rx Vref Range Down** Max step count supported down from the trained Rx Vref value for Vref margining.
55:48	RO	**Trained Value for Rx Vref** Rx Vref as trained, in resolution counts.
63:56	RO	**Vref Step Count Resolution** Increase in Vref value in mV between two consecutive encodings in ascending order.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.5.4.5.5 Physical Layer Compliance Status 3 (Offsets 020h, 420h, 820h, and C20h from PHYOFF)

Table 7-82. Physical Layer Compliance Status 3

Bit	Attributes	Description
63:0	RO	**Per Lane Comparison Result** Per Lane comparison result in PHY Compliance when 'PHY Compliance Operation Type' is 001b or 011b (performing point tests) and 'Comparison Setup' is 1b (Per Lane comparison) {[63:0]: Compare Results of all Logical Data Lanes (0h Fail (Errors > Max Error Threshold), 1h Pass (Errors < Max Error Threshold)) UCIe-A {RD_L[63], RD_L[62], …., RD_L[1], RD_L[0]} UCIe-A x32 {32'h0, RD_L[31], RD_L[30], …, RD_L[1], RD_L[0]} UCIe-S {48'h0, RD_L[15], RD_L[14], …, RD_L[1], RD_L[0]} Default is all 0s.

7.5.5 Implementation Specific Register Blocks

These are left to be vendor defined. There is a separate implementation specific register Block for D2D Adapter and PHY. These register blocks should carry the same header as defined in Table 7-26, at offset 0h of the register block. And the VendorID should be set to the specific vendor's ID and the 'VendorID register block' field set to 2h or 3h to indicate that it is a vendor specific register block. The other fields in that header are set by the vendor to track their revision number and the block length. Max length cannot exceed 1MB in size and length is always in multiples of 4KB. Implementations are highly encouraged to pack registers and reduce length of the region as much as possible.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.6 UCIe Link Registers in Streaming Mode and System SW/ FW Implications

IMPLEMENTATION NOTE

While the SW view of Protocol Layer for streaming protocols is implementation-specific, it is strongly recommended that UCIe link-related registers defined in this chapter be implemented as-is for streaming mode solutions as well. If a streaming mode solution chooses to support the industry-standard PCIe hierarchical tree model for enumeration/control, it must be compliant with the enumeration model and registers defined in this chapter. A UCIe port in such an implementation would expose UCIe link registers consistent with the RP/DSP or EP/USP functionality it represents.

In some streaming mode solutions, it might be desirable to implement UCIe link as a fully symmetric link, such as in a Symmetric Multi-Processing system that uses UCIe as a D2D interconnect. In such solutions, there is no notion of Upstream Port or Downstream Port on a UCIe link and also typically system firmware knows the D2D link connections a priori and it is able to configure them without requiring any "link discovery" mechanisms. It is recommended that both ends of the link implement UCIe registers defined for a Root Port, in such streaming mode solutions. Note that in this model, several link-related features become fully symmetric as well. For example, link training, mailbox trigger, and direct link-event/error reporting to Software are now possible from either end of the link. Whether such symmetric UCIe links are exposed to OS for native management, or system FW fully manages these links, is a system-architecture choice. Exposing such links natively to the OS could be in the form of exposing each side as an ACPI device or in the form of an FW intermediary that emulates a traditional PCIe hierarchical tree model for the symmetric link. Such choices are implementation-specific and could depend on the extent of OS support for symmetric topology.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

7.7 MSI and MSI-X Capability in Hosts/Switches for UCIe interrupt

Follow the base spec for details, but MSI/MSI-X capability implemented in host and switch must request 2 vectors for UCIe usage - 1 for Link status events and 1 for Link error events. Note that in MSI scenario, OS might not always allot both the requested vectors and in that case both the Link Status and Link error events use the same MSI vector number. The MSI designs must also support the Pending and Mask bits.

7.8 UCIe Early Discovery Table (UEDT)

Table 7-83. UEDT Header

Field	Byte Offset	Length in Bytes	Description
Signature	00h	4	Signature for the UCIe Early Discovery Table (UEDT).
Length	04h	4	Length, in bytes, of the entire UEDT.
Revision	08h	1	Value is 1h for the first UCIe instance.
Checksum	09h	1	Entire table must sum to 0.
OEM ID	0Ah	6	OEM ID
OEM Table ID	10h	8	Manufacturer Model ID
OEM Revision	18h	4	OEM Revision
Creator ID	1Ch	4	Vendor ID of the utility that created this table.
Creator Revision	20h	4	Revision of the utility that created this table.
UEDT Structure[n]	24h	Varies	A list of UEDT structures for this implementation. • 0h = UCIe Link structure (UCLS) • All other encodings are reserved

Table 7-84. UCIe Link Structure (UCLS) (Sheet 1 of 2)

Field	Byte Offset	Length in Bytes	Description
Type	00h	1	Signature for the UCIe Early Discovery Table (UEDT).
Revision	01h	1	Value is 1h for the first UCLS definition.
Record Length	02h	2	Length of this record, in bytes.
UID	04h	4	Host Bridge Unique ID. Used to associate a UCLS instance with a Host Bridge instance. The value of this field shall match the output of UID under the associated Host Bridge in ACPI namespace.
UCIe Stack Size	08h	4	• 1h = One RP • 2h = Two RPs
Reserved	0Ch	4	Reserved
Base	10h	8	Base address of UiRB, aligned to a 4-KB boundary.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 7-84. UCIe Link Structure (UCLS) (Sheet 2 of 2)

Field	Byte Offset	Length in Bytes	Description
Length	18h	8	Can range anywhere from 12 KB to 2 MB, in multiples of 4 KB.
DF1	20h	1	Device Function of the PCIe/CXL RP 1 associated with the UCLS.
DF2	21h	1	Device Function of the PCIe/CXL RP 2 (if multi-stack implementation) associated with the UCLS.

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.0 Interface Definitions

This chapter will cover the details of interface operation and signal definitions for the Raw Die-to-Die Interface (RDI), as well as the Flit-Aware Die-to-Die Interface (FDI). Common rules across RDI and FDI are covered as a separate section. The convention used in this chapter is that "assertion" of a signal is for 0b to 1b transition, and "de-assertion" of a signal is for 1b to 0b transition.

8.1 Raw Die-to-Die Interface (RDI)

This section defines the signal descriptions and functionality associated with a single instance of Raw Die-to-Die Interface (RDI). A single instance could be used for a configuration associated with a single Die-to-Die module (i.e., one Die-to-Die Adapter for one module), or a single instance is also applicable for configurations where multiple modules are grouped together for a single logical Die-to-Die Link (i.e., one Die-to-Die Adapter for multiple modules). Figure 8-1 shows example configurations using RDI.

Figure 8-1. Example configurations using RDI

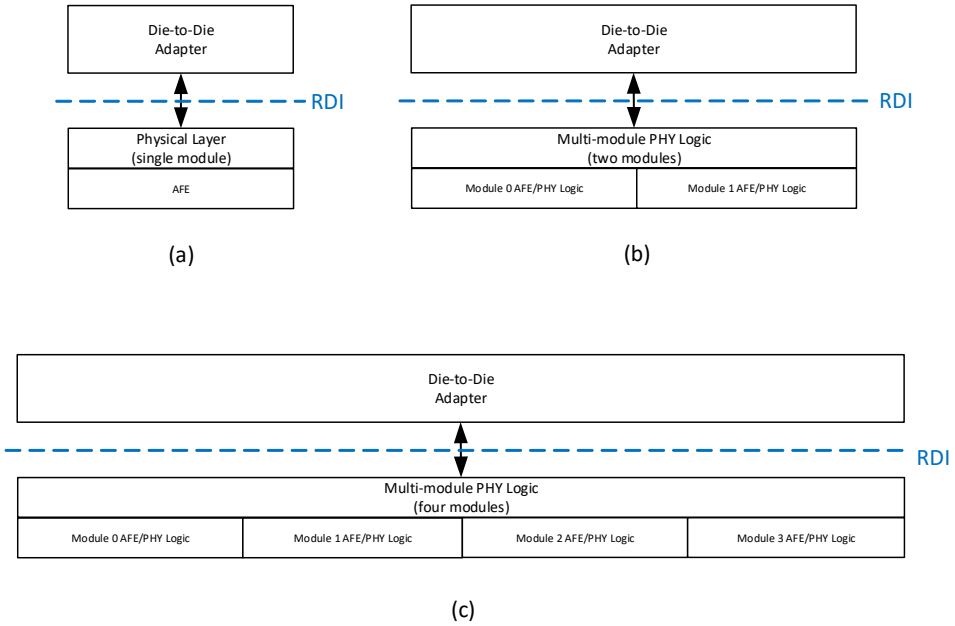


Table 8-1 lists the RDI signals and their descriptions. All signals are synchronous with **`lclk`**.

In Table 8-1:

- pl_* indicates that the signal is driven away from the Physical Layer to the Die-to-Die Adapter.
- lp_* indicates that the signal is driven away from the Die-to-Die Adapter to the Physical Layer.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-1. RDI signal list (Sheet 1 of 4)

Signal Name	Signal Description
lclk	The clock at which RDI operates.
lp_irdy	Adapter to Physical Layer signal indication that the Adapter has data to send. This must be asserted if **lp_valid** is asserted and the Adapter wants the Physical Layer to sample the data. **lp_irdy** must not be presented by the Adapter when **pl_state_sts** is Reset except when the status transitions from LinkError to Reset. On a LinkError to Reset transition, it is permitted for **lp_irdy** to be asserted for a few clocks but it must be de-asserted eventually. Physical Layer must ignore **lp_irdy** when status is Reset.
lp_valid	Adapter to Physical Layer indication that data is valid on the corresponding **lp_data** bytes.
lp_data[NBYTES-1:0][7:0]	Adapter to Physical Layer data, where 'NBYTES' equals number of bytes determined by the data width for the RDI instance.
lp_retimer_crd	When asserted at a rising clock edge, it indicates a single credit return from the Adapter to the Physical Layer for the Retimer Receiver buffers. Each credit corresponds to 256B of mainband data. This signal must NOT assert for dies that are not UCIe Retimers.
pl_trdy	The Physical Layer is ready to accept data. Data is accepted by the Physical Layer when **pl_trdy**, **lp_valid**, and **lp_irdy** are asserted at the rising edge of **lclk**.
pl_valid	Physical Layer to Adapter indication that data is valid on **pl_data**.
pl_data[NBYTES-1:0][7:0]	Physical Layer to Adapter data, where NBYTES equals the number of bytes determined by the data width for the RDI instance.
pl_retimer_crd	When asserted at a rising clock edge, it indicates a single credit return from the Retimer to the Adapter. Each credit corresponds to 256B of mainband data. This signal must NOT assert if the remote Link partner is not a Retimer.
lp_state_req[3:0]	Adapter request to Physical Layer to request state change. Encodings as follows: 0000b : NOP 0001b : Active 0100b : L1 1000b : L2 1001b : LinkReset 1011b : Retrain 1100b : Disabled All other encodings are reserved.
lp_linkerror	Adapter to Physical Layer indication that an error has occurred which requires the Link to go down. Physical Layer must move to LinkError state and stay there as long as **lp_linkerror**=1. The reason for having this be an indication decoupled from regular state transitions is to allow immediate action on part of the Adapter and Physical Layer in order to provide the quickest path for error containment when applicable (for example, a viral error escalation must map to the LinkError state). The Adapter must OR internal error conditions with **lp_linkerror** received from Protocol Layer on FDI.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-1. RDI signal list (Sheet 2 of 4)

Signal Name	Signal Description
pl_state_sts[3:0]	Physical Layer to Adapter Status indication of the Interface. Encodings as follows: 0000b : Reset 0001b : Active 0011b : Active.PMNAK 0100b : L1 1000b : L2 1001b : LinkReset 1010b : LinkError 1011b : Retrain 1100b : Disabled All other encodings are reserved. The status signal is permitted to transition from Physical Layer autonomously when applicable. For example the Physical Layer asserts the Retrain status when it decides to enter retraining either autonomously or when requested by remote agent.
pl_inband_pres	Physical Layer to the Adapter indication that the Die-to-Die Link has finished training and is ready for RDI transition to Active and Stage 3 of bring up. Once it transitions to 1b, this must stay 1b until Physical Layer determines the Link is down (i.e., the Link Training State Machine transitions to TrainError or Reset).
pl_error	Physical Layer to the Adapter indication that it has detected a framing related error which is recoverable through Link Retrain. It is pipeline matched with the receive data path. Physical Layer is expected to go through Retrain flow after this signal has been asserted and it must not send valid data to Adapter until the Link has retrained. If **pl_error**=1 and **pl_valid**=1 in the same clock cycle, the Adapter must discard the corresponding Flit (even if it is only partially received when **pl_error** asserted) and trigger a retry (if retry is enabled).
pl_cerror	Physical Layer to the Adapter indication that a correctable error was detected that does not affect the data path and will not cause Retrain on the Link. The Adapter must OR the **pl_error** and **pl_cerror** signals for Correctable Error Logging.
pl_nferror	Physical Layer to the Adapter indication that a non-fatal error was detected.
pl_trainerror	Indicates a fatal error from the Physical Layer. Physical Layer must transition **pl_state_sts** to LinkError if not already in LinkError state. This must be escalated to upper Protocol Layers. Implementations are permitted to map any fatal error to this signal that require upper layer escalation (or interrupt generation) depending on system level requirements.
pl_phyinrecenter	Physical Layer indication to Adapter that the Physical Layer is training or retraining. If this is asserted during a state where clock gating is permitted, the **pl_clk_req/ lp_clk_ack** handshake must be performed with the upper layer. The upper layers are permitted to use this to update the "Link Training/Retraining" bit in the UCIe Link Status register.
pl_stallreq	Physical Layer request to Adapter to align Transmitter at Flit boundary and not send any new Flits to prepare for state transition. See Section 8.3.2.
lp_stallack	Adapter to Physical Layer indication that the Flits are aligned and stalled (if **pl_stallreq** was asserted). It is strongly recommended that this response logic be on a global free running clock, so the Adapter can respond to **pl_stallreq** with **lp_stallack** even if other significant portions of the Adapter are clock gated. See Section 8.3.2.
pl_speedmode[2:0]	Current Link speed. The following encodings are used: 000b: 4GT/s 001b: 8GT/s 010b: 12GT/s 011b: 16GT/s 100b: 24GT/s 101b: 32GT/s other encodings are reserved. The Adapter must only consider this signal to be relevant when the RDI state is Active or Retrain. For multi-module configurations, all modules must operate at the same speed.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-1. RDI signal list (Sheet 3 of 4)

Signal Name	Signal Description
pl_lnk_cfg[2:0]	Current Link Configuration. Indicates the current operating width of a module. 001b: x8 010b: x16 011b: x32 100b: x64 101b: x128 110b: x256 other encodings are reserved. This is the width of the UCIe physical die-to-die Link which may be comprised of one to four modules. For UCIe-S the maximum encoding would be x64, for UCIe-A the maximum encoding would be x128 for UCIe-A x32 and x256 for UCIe-A x64. The Adapter must only consider this signal to be relevant when the RDI state is Active or Retrain. This signal indicates the total width across all Active Modules corresponding to the RDI instance.
pl_clk_req	Request from the Physical Layer to remove clock gating from the internal logic of the Adapter. This is an asynchronous signal relative to **lclk** from the Adapter's perspective since it is not tied to **lclk** being available in the Adapter. Together with **lp_clk_ack**, it forms a four-way handshake to enable dynamic clock gating in the Adapter. When dynamic clock gating is supported, the Adapter must use this signal to exit clock gating before responding with **lp_clk_ack**. If dynamic clock gating is not supported, it is permitted for the Physical Layer to tie this signal to 1b.
lp_clk_ack	Response from the Adapter to the Physical Layer acknowledging that its clocks have been ungated in response to **pl_clk_req**. This signal is only asserted when **pl_clk_req** is asserted, and de-asserted after **pl_clk_req** has de-asserted. When dynamic clock gating is not supported by the Adapter, it must stage **pl_clk_req** internally for one or more clock cycles and turn it around as **lp_clk_ack**. This way it will still participate in the handshake even though it does not support dynamic clock gating.
lp_wake_req	Request from the Adapter to remove clock gating from the internal logic of the Physical Layer. This is an asynchronous signal from the Physical Layer's perspective since it is not tied to **lclk** being available in the Physical Layer. Together with **pl_wake_ack**, it forms a four-way handshake to enable dynamic clock gating in the Physical Layer. When dynamic clock gating is supported, the Physical Layer must use this signal to exit clock gating before responding with **pl_wake_ack**. If dynamic clock gating is not supported, it is permitted for the Adapter to tie this signal to 1b.
pl_wake_ack	Response from the Physical Layer to the Adapter acknowledging that its clocks have been ungated in response to **lp_wake_req**. This signal is only asserted after **lp_wake_req** has asserted, and is de-asserted after **lp_wake_req** has de-asserted. When dynamic clock gating is not supported by the Physical Layer, it must stage **lp_wake_req** internally for one or more clock cycles and turn it around as **pl_wake_ack**. This way it will still participate in the handshake even though it does not support dynamic clock gating.
pl_cfg[NC-1:0]	This is the sideband interface from the Physical Layer to the Adapter. See Chapter 6.0 for packet format details. NC is the width of the interface. Supported values are 8, 16, and 32.
pl_cfg_vld	When asserted, indicates that **pl_cfg** has valid information that should be consumed by the Adapter.
pl_cfg_crd	Credit return for sideband packets from the Physical Layer to the Adapter for sideband packets. Each credit corresponds to 64 bits of header and 64 bits of data. Even transactions that don't carry data or carry 32 bits of data consume the same credit and the Physical Layer returns the credit once the corresponding transaction has been processed or deallocated from its internal buffers. See Section 6.1.3.1 for additional flow control rules. Because the advertised credits are design parameters, the Adapter transmitter updates the credit counters with initial credits on domain reset exit, and no initialization credits are returned over the interface. Credit returns must follow the same rules of clock gating exit handshakes as the sideband packets to ensure that no credit returns are dropped by the receiver of the credit returns.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-1. RDI signal list (Sheet 4 of 4)

Signal Name	Signal Description
`lp_cfg[NC-1:0]`	This is the sideband interface from Adapter to the Physical Layer. See Chapter 6.0 for details. NC is the width of the interface. Supported values are 8, 16, and 32.
`lp_cfg_vld`	When asserted, indicates that **`lp_cfg`** has valid information that should be consumed by the Physical Layer.
`lp_cfg_crd`	Credit return for sideband packets from the Adapter to the Physical Layer for sideband packets. Each credit corresponds to 64 bits of header and 64 bits of data. Even transactions that don't carry data or carry 32 bits of data consume the same credit and the Adapter returns the credit once the corresponding transaction has been processed or deallocated from its internal buffers. See Section 6.1.3.1 for additional flow control rules. Because the advertised credits are design parameters, the Physical Layer transmitter updates the credit counters with initial credits on domain reset exit, and no initialization credits are returned over the interface. Credit returns must follow the same rules of clock gating exit handshakes as the sideband packets to ensure that no credit returns are dropped by the receiver of the credit returns.

8.1.1 Interface reset requirements

RDI does not define a separate interface signal for reset; however, it is required that the logic entities on both sides of RDI are in the same reset domain and the reset for each side is derived from the same source.

8.1.2 Interface clocking requirements

RDI requires both sides of the interface to be on the same clock domain. Moreover, the clock domain for sideband interface (**`*cfg*`**) is the same as the mainband signals.

Each side is permitted to instantiate clock crossing FIFOs internally if needed, as long as it does not violate the requirements at the interface itself.

It is important to note that there is no back pressure possible from the Adapter to the Physical Layer on the main data path. So any clock crossing related logic internal to the Adapter must take this into consideration.

For example, for a 64 Lane module with a max speed of 16GT/s, RDI could be 64B wide running at 2GHz to be exactly bandwidth matched.

8.1.3 Dynamic clock gating

Dynamic coarse clock gating is permitted in the Adapter and Physical Layer when **`pl_state_sts`** is Reset, LinkReset, Disabled, or PM. This section defines the rules around entry and exit of clock gating. Note that clock gating is not permitted in LinkError state; it is expected that for UCIe usages, error handlers will be enabled to make sure the Link is not stuck in LinkError state if the intent is save power for Links in error state.

8.1.3.1 Rules and description for lp_wake_req/pl_wake_ack handshake

Adapter can request removal of clock gating of the Physical Layer by asserting **`lp_wake_req`** (asynchronous to **`lclk`** availability in the Physical Layer). All Physical Layer implementations must respond with a **`pl_wake_ack`** (synchronous to **`lclk`**). The extent of internal clock ungating when **`pl_wake_ack`** is asserted is implementation-specific, but **`lclk`** must be available by this time to

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

enable RDI signal transitions from the Adapters. The Wake Req/Ack is a full handshake and it must be used for state transition requests (on **lp_state_req** or **lp_linkerror**) when moving away from Reset or PM states. It must also be used for sending packets on the sideband interface.

Rules for this handshake:

1. Adapter asserts **lp_wake_req** to request ungating of clocks by the Physical Layer.
2. The Physical Layer asserts **pl_wake_ack** to indicate that clock gating has been removed. There must be at least one clock cycle bubble between **lp_wake_req** assertion and **pl_wake_ack** assertion.
3. **lp_wake_req** must de-assert before **pl_wake_ack** de-asserts. It is the responsibility of the Adapter to control the specific scenario of de-assertion. As an example, when performing the handshake for a state request, it is permitted to keep **lp_wake_req** asserted until it observes the desired state status. Adapter is also permitted to keep **lp_wake_req** asserted through states where clock gating is not allowed in the Physical Layer (i.e., Active, LinkError, or Retrain).
4. **lp_wake_req** should not be the only consideration for Physical Layer to perform clock gating, it must take into account **pl_state_sts** and other internal or Link requirements before performing global and/or local clock gating.
5. When performing **lp_wake_req/pl_wake_ack** handshake for **lp_state_req** transitions or **lp_linkerror** transition, the Adapter is permitted to not wait for **pl_wake_ack** before changing **lp_state_req** or **lp_linkerror**.
6. When performing **lp_wake_req/pl_wake_ack** handshake for **lp_cfg** transitions, Adapter must wait for **pl_wake_ack** before changing **lp_cfg** or **lp_cfg_vld**. Because **lp_cfg** can have multiple transitions for a single packet transfer, it is necessary to make sure that the Physical Layer clocks are up before transfer begins.

8.1.3.2 Rules and description for pl_clk_req/lp_clk_ack handshake

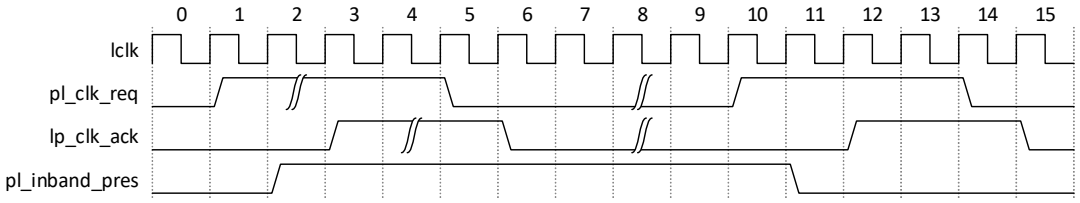
Physical Layer is allowed to initiate **pl_clk_req/lp_clk_ack** handshake at any time and the Adapter must respond.

Rules for this handshake:

1. Physical Layer asserts **pl_clk_req** to request removal of clock gating by the Adapter. This can be done anytime, and independent of current RDI state.
2. The Adapter asserts **lp_clk_ack** to indicate that clock gating has been removed. There must be at least one clock cycle bubble between **pl_clk_req** assertion and **lp_clk_ack** assertion.
3. **pl_clk_req** must de-assert before **lp_clk_ack**. It is the responsibility of the Physical Layer to control the specific scenario of de-assertion, after the required actions for this handshake are completed.
4. **pl_clk_req** should not be the only consideration for the Adapter to perform clock gating, it must take into account **pl_state_sts** and other protocol-specific requirements before performing trunk and/or local clock gating.
5. The Physical Layer must use this handshake to ensure transitions of **pl_inband_pres** have been observed by the Adapter. Since **pl_inband_pres** is a level oriented signal (once asserted it stays asserted during the lifetime of Link operation), the Physical Layer is permitted to let the signal transition without waiting for **lp_clk_ack**. When this is done during initial Link bring up, it is strongly recommended for the Physical Layer to keep **pl_clk_req** asserted until the state status transitions away from Reset to a state where clock gating is not permitted.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-2. Example Waveform Showing Handling of Level Transition



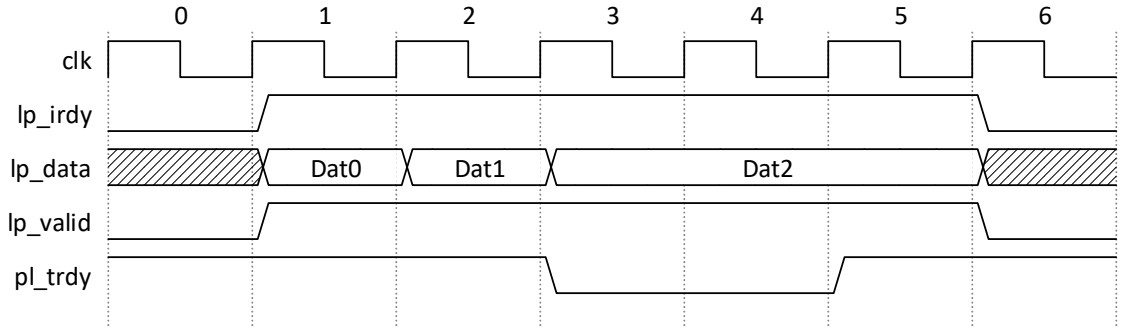
6. The Physical Layer must also perform this handshake before transition to LinkError state from Reset or PM state (when the LinkError transition occurs by the Physical Layer without being directed by the Adapter). It is permitted to assert **pl_clk_req** before the state change, in which case it must stay asserted until the state status transitions. It is also permitted to assert **pl_clk_req** after the state status transition, but in this case Physical Layer must wait for **lp_clk_ack** before performing another state transition.
7. The Physical Layer must also perform this handshake when the status is PM and remote Link partner is requesting PM exit. For exit from Reset or PM states to a state that is not LinkError, it is required to assert pl_clk_req before the status change, and in this case it must stay asserted until the state status transitions away from Reset or PM.
8. When clock-gated in RESET states, Adapters that rely on dynamic clock gating to save power must wait in clock gated state for **pl_inband_pres**=1. The Physical Layer will request clock gating exit when it transitions **pl_inband_pres**, and the Adapter must wait for **pl_inband_pres** assertion before requesting **lp_state_req** = ACTIVE. If **pl_inband_pres** de-asserts while **pl_state_sts** = RESET, then the Adapter is permitted to return to clock-gated state after moving **lp_state_req** to NOP.
9. Physical Layer must also perform this handshake for sideband traffic to Adapter. When performing the handshake for **pl_cfg** transitions, Physical Layer must wait for **lp_clk_ack** before changing **pl_cfg** or **pl_cfg_vld**. Because **pl_cfg** can have multiple transitions for a single packet transfer, it is necessary to make sure that the Adapter clocks are up before transfer begins.

8.1.4 Data Transfer

As indicated in the signal list descriptions, when Adapter is sending data to the Physical Layer, data is transferred when **lp_irdy**, **pl_trdy**, and **lp_valid** are asserted. Figure 8-3 shows an example waveform for data transfer from the Adapter to the Physical Layer. Data is transmitted on clock cycles 1, 2, and 5. No assumption should be made by Adapter about when **pl_trdy** can de-assert or for how many cycles it remains de-asserted before it is asserted again, unless explicitly guaranteed by the Physical Layer. If a Flit transfer takes multiple clock cycles, the Adapter is not permitted to insert bubbles in the middle of a Flit transfer. This means that **lp_valid** and **lp_irdy** must be asserted continuously until the Flit transfer is complete. Of course, data transfer can stall because of **pl_trdy** de-assertion.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-3. Data Transfer from Adapter to Physical Layer

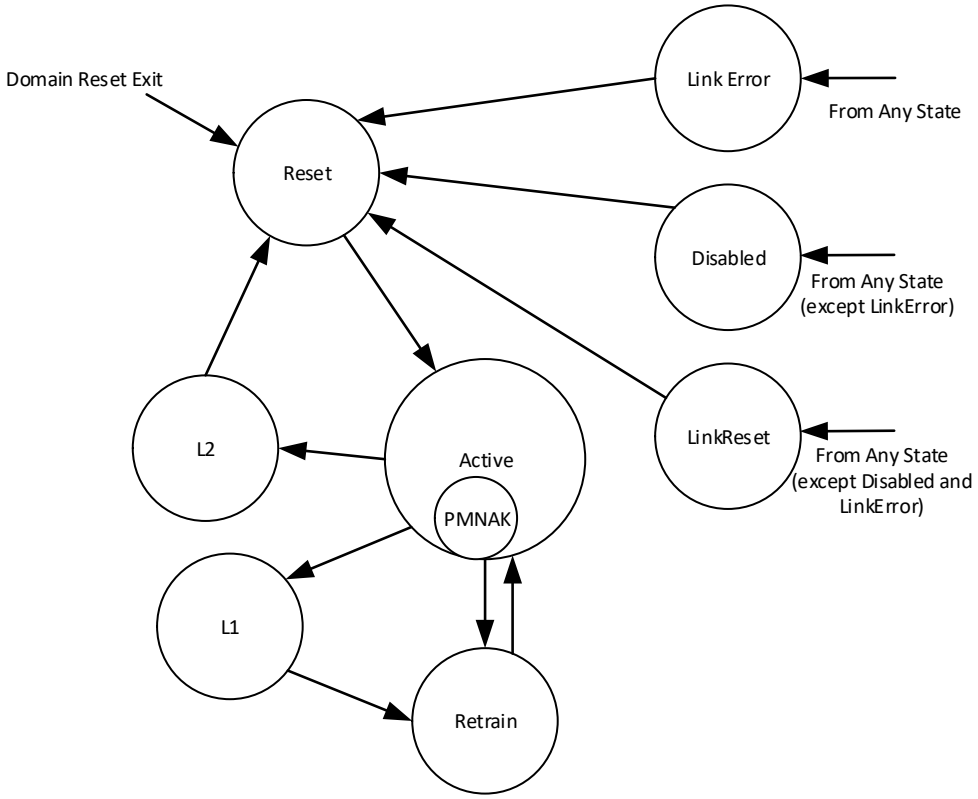


As indicated in the signal list descriptions, when the Physical Layer is sending data to the Adapter, there is no backpressure mechanism, and data is transferred whenever **`pl_valid`** is asserted. The Physical Layer is permitted to insert bubbles in the middle of a Flit transfer and the Adapter must be able to handle that.

8.1.5 RDI State Machine

Figure 8-4 shows the RDI state machine.

Figure 8-4. RDI State Machine



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

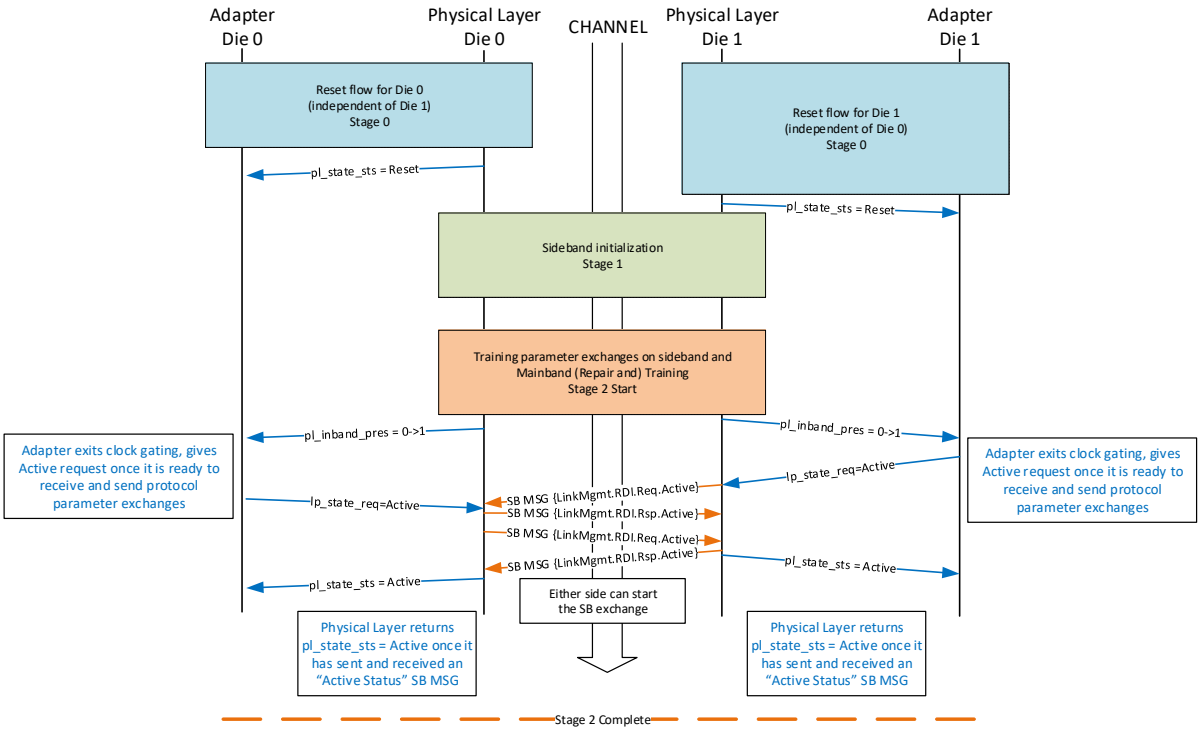
8.1.6 RDI bring up flow

Figure 8-5 shows an example flow for Stage 2 of the Link bring up highlighting the transitions on RDI. This stage requires sequencing on RDI that coordinates the state transition from Reset to Active.

1. Once Physical Layer has completed Link training, it must do the **pl_clk_req** handshake with the Adapter and reflect **pl_inband_pres**=1 on RDI. Note that the **pl_clk_req** handshake is not shown in the example flow in Figure 8-5
2. This is the trigger for Adapter to request Active state. It must perform the **lp_wake_req** handshake as described in Section 8.1.3. Note that the **lp_wake_req** handshake is not shown in the example flow in Figure 8-5.
3. Only after sampling **lp_state_req** = Active, the Physical Layer must send the {LinkMgmt.RDI.Req.Active} sideband message to remote Link partner's Physical Layer.
4. The Physical Layer must respond to the {LinkMgmt.RDI.Req.Active} sideband message with a {LinkMgmt.RDI.Rsp.Active} sideband message. The {LinkMgmt.RDI.Rsp.Active} sideband message must only be sent after the Physical Layer has sampled **lp_state_req** = Active from its local RDI.
5. Once the Physical Layer has sent and received the {LinkMgmt.RDI.Rsp.Active} sideband message, it must transition **pl_state_sts** to Active.
6. This opens up the Adapter to transition to Stage 3 of the bring up flow.

Steps 3 to 5 are referred to as the "Active Entry handshake" and must be performed for every entry to Active state. Active.PMNAK to Active transition is not considered here because Active.PMNAK is only a sub-state of Active.

Figure 8-5. Example flow of Link bring up on RDI



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.1.7 RDI PM flow

This section defines the rules for PM entry, exit and abort flows as they apply to handshakes on the RDI. The rules for L1 and L2 are the same, except that exit from L2 is to Reset state, whereas exit from L1 is to Retrain state. This section uses PM to denote L1 or L2. A "PM Request" sideband message is {LinkMgmt.RDI.Req.L1} or {LinkMgmt.RDI.Req.L2}. A "PM Response" sideband message is {LinkMgmt.RDI.Rsp.L1} or {LinkMgmt.RDI.Rsp.L2}.

- Regardless of protocol, the PM entry or exit flow is symmetric on RDI. Both Physical Layer must issue PM entry request through a sideband message once the conditions of PM entry have been satisfied. PM entry is considered successful and complete once both sides have received a valid "PM Response" sideband message. Figure 8-6 shows an example flow for L1. Once the RDI status is PM, the Physical Layer can transition itself to a power savings state (turning off the PLL for example). Note that the sideband logic and corresponding PLL needs to stay on even during L1 state.
- All the Adapter state machines (Adapter LSMs) in the Adapter must have moved to the corresponding PM state before the Adapter requests PM entry from remote Link partner. Adapter LSM in PM implies the retry buffer of the Adapter must be empty, and it must not have any new Flits (or Ack/Nak) pending to be scheduled. Essentially there should be no traffic on mainband when PM entry is requested by the Adapter to the Physical Layer. The Adapter is permitted to clock gate its sideband logic once RDI status is PM and there are no outstanding transactions or responses on sideband. Physical Layer must do **pl_clk_req** handshake (if **pl_clk_req** is not already asserted or status is not Active) before forwarding sideband requests from the Link to the Adapter.
- Adapter requests PM entry by transitioning **lp_state_req** to the corresponding PM encoding. Once requested, the Adapter cannot change this request until it observes PM, Active.PMNAK or LinkError state on **pl_state_sts**. While requesting PM state, if the Adapter receives Active request from the Protocol Layer, or a PM exit request for the Adapter LSM on sideband, it must sink the message but delay processing it until **pl_state_sts** has resolved. Once the RDI state is resolved, the Adapter must first bring it back to Active before processing the other requests.
  - If the resolution is PM (upon successful PM entry) and the Protocol Layer needs to exit PM (or there is a pending Protocol Layer Active request from remote Link partner), then the Adapter must initiate PM exit flow on RDI by requesting **lp_state_req** = Active. All PM entry-related handshakes must have finished prior to this (this is when both sides Physical Layer have received a valid "PM Response" sideband message).
  - If the resolution is Active.PMNAK, the Adapter must initiate a request of Active on RDI. Once the status moves to Active, the Adapter is permitted to re-request PM entry (if all conditions of PM entry are still met). Figure 8-7 shows an example of PM abort flow. The PM request could have been from either side.
  - If the resolution is LinkError, then the Adapter must propagate this to Protocol Layers. This also resets any outstanding PM handshakes.
- Physical Layer initiates a "PM Request" sideband message once it samples the corresponding PM encoding on **lp_state_req** and has completed the StallReq/Ack handshake with its Adapter.
- Once a Physical Layer receives a "PM request" sideband message, it must respond to it within 2 us:
  - If its local Adapter is requesting the corresponding PM state, it must respond with the corresponding "PM Response" sideband message. If the current status is not PM, it must transition **pl_state_sts** to PM after responding to the sideband message.
  - If the current **pl_state_sts** = PM, it must respond with "PM Response" sideband message.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

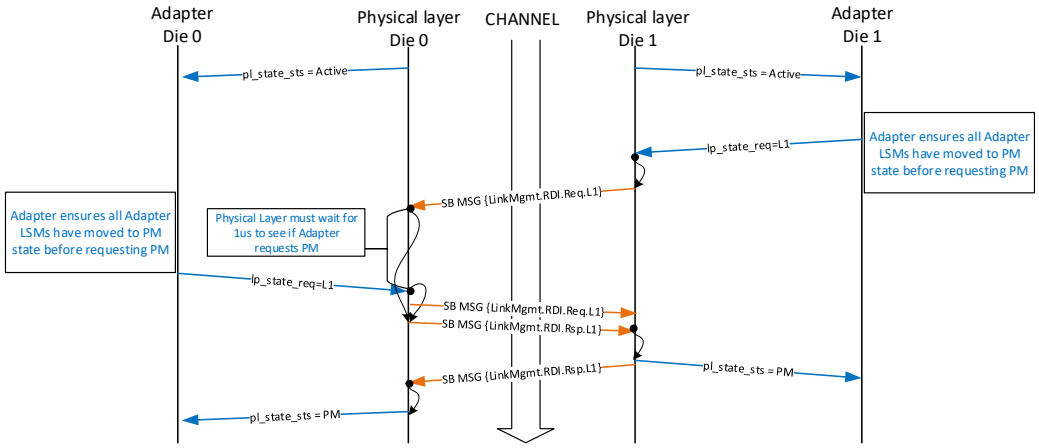
  - If **pl_state_sts** = Active and **lp_state_req** = Active and it remains this way for 1us after receiving the "PM Request" sideband message, it must respond with {LinkMgmt.RDI.Rsp.PMNAK} sideband message.

- If a Physical Layer receives a "PM Response" sideband message in response to a "PM Request" sideband message, it must transition **pl_state_sts** on its local RDI to PM (if it is not currently in a PM state).
- If a Physical Layer receives a {LinkMgmt.RDI.Rsp.PMNAK} sideband message in response to a "PM Request" sideband message, it must transition **pl_state_sts** on its local RDI to Active.PMNAK state. The Physical Layer is permitted to retry PM entry handshake (if all conditions of PM entry are satisfied) at least 2 us after receiving the {LinkMgmt.RDI.Rsp.PMNAK} sideband message OR if it received a corresponding "PM Request" sideband message from the remote Link partner.
- PM exit is initiated by the Adapter requesting Active on RDI. This triggers the Physical Layer to initiate PM exit by sending a {LinkMgmt.RDI.Req.Active} sideband message. Physical Layer must make sure it has finished any Link retraining steps before it responds with the {LinkMgmt.RDI.Rsp.Active} sideband message. Figure 8-8 shows an example flow of PM exit on RDI.
  - PM exit handshake completion requires both Physical Layers to send as well as receive a {LinkMgmt.RDI.Rsp.Active} sideband message. Once this has completed, the Physical Layer is permitted to transition **pl_state_sts** to Active on RDI.
  - If **pl_state_sts** = PM and a {LinkMgmt.RDI.Req.Active} sideband message is received, the Physical Layer must initiate **pl_clk_req** handshake with the Adapter, and transition **pl_state_sts** to Retrain. This must trigger the Adapter to request Active on **lp_state_req** (if not already doing so), and this in turn triggers the Physical Layer to send {LinkMgmt.RDI.Req.Active} sideband message to the remote Link partner. Figure 8-9 shows an example of the L1 exit flow on RDI and its interaction with the LTSM in the Physical Layer. It is permitted for the LTSM to begin the Link PM exit and retraining flow when a {LinkMgmt.RDI.Req.Active} sideband message is received or when the Adapter requests Active on RDI. The timeout counters for the Active Request sideband message handshake must begin only after LTSM is in the LINKINIT state. L2 exit follows a similar flow for cases in which graceful exit is required without domain reset; however, the L2 exit is via Reset state on RDI, and not Retrain. Exit conditions from Reset state apply for L2 exit, i.e., a NOP -> Active transition is required on **lp_state_req** for the Physical Layer to exit Reset state on RDI.

Note that the following figures are examples for L1, and do not show the **lp_wake_req**, **pl_clk_req** handshakes. Implementations must follow the rules outlined for these handshakes in previous sections.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-6. Successful PM entry flow



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-7. PM Abort flow

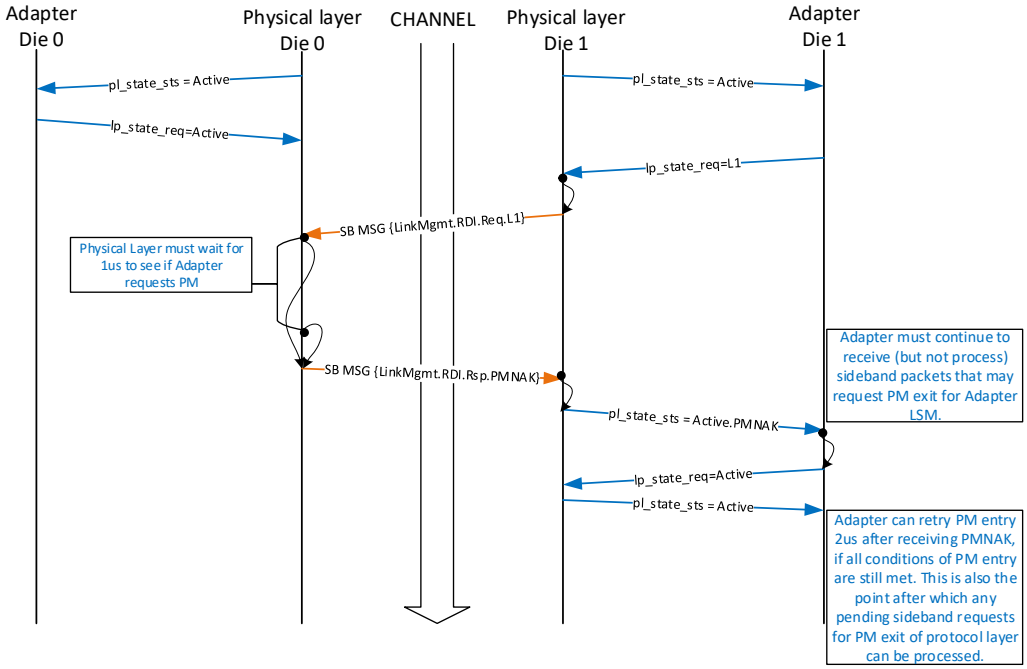
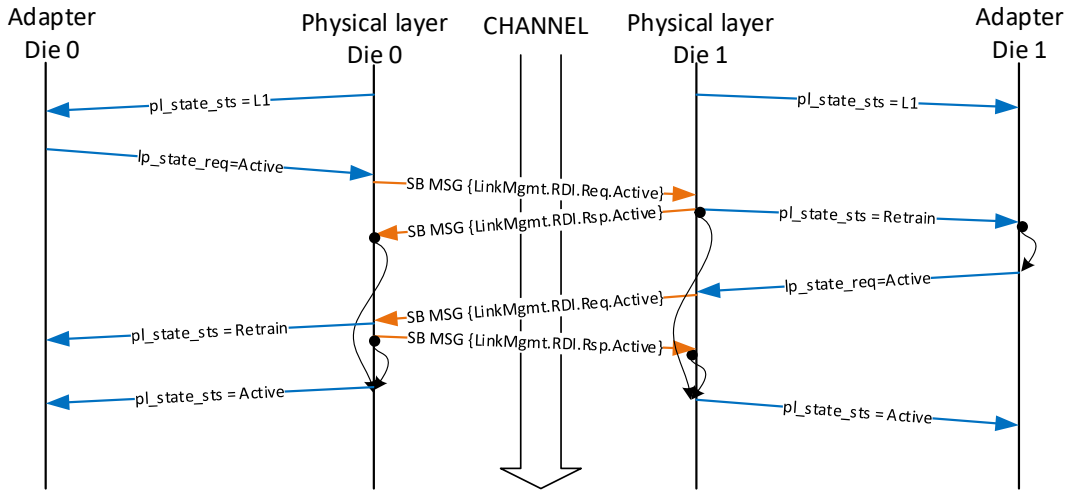
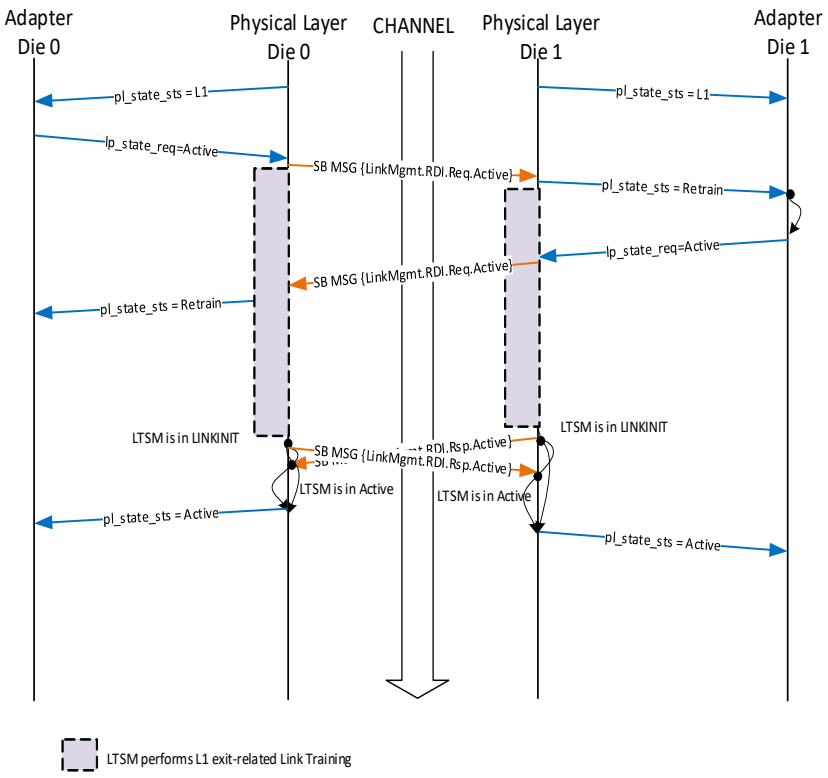


Figure 8-8. PM Exit flow



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-9. RDI PM Exit Example Showing Interactions with LTSM



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.2 Flit-Aware Die-to-Die Interface (FDI)

This section defines the signal descriptions and functionality associated with a single instance of Flit-Aware Die-to-Die Interface (FDI). A single instance is used for a Protocol Layer to Adapter connection. However, a single Adapter can host multiple protocol stacks using multiple instances of FDI. Figure 8-10 shows example configurations using multiple instances of FDI.

Figure 8-10. Example configurations using FDI

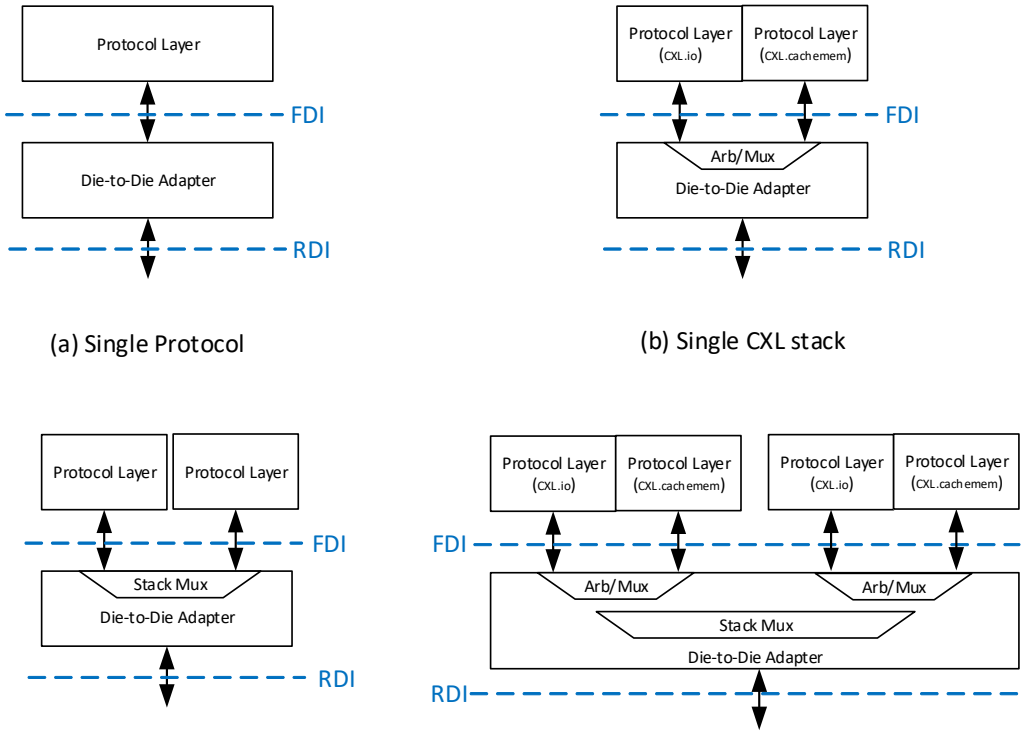


Table 8-2 lists the FDI signals and their descriptions. All signals are synchronous with **`lclk`**.

In Table 8-2:

- pl_* indicates that the signal is driven away from the Die-to-Die Adapter to the Protocol Layer.
- lp_* indicates that the signal is driven away from the Protocol Layer to the Die-to-Die Adapter.

Note: The same signal naming convention as RDI is used to highlight that RDI signal list is a proper subset of FDI signal list.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-2. FDI signal list (Sheet 1 of 6)

Signal Name	Signal Description
lclk	The clock at which FDI operates.
lp_irdy	Signal indicating that the Protocol Layer potentially has data to send. This must be asserted if lp_valid is asserted and the Protocol Layer wants the Adapter to sample the data. **lp_irdy** must not be presented by the Protocol Layer when **pl_state_sts** is Reset except when the status transitions from LinkError to Reset. On a LinkError to Reset transition, it is permitted for **lp_irdy** to be asserted for a few clocks but it must be de-asserted eventually. Physical Layer must ignore **lp_irdy** when status is Reset.
lp_valid	Protocol Layer to Adapter indication that data is valid on the corresponding **lp_data** bytes.
lp_data[NBYTES-1:0][7:0]	Protocol Layer to Adapter data, where 'NBYTES' equals number of bytes determined by the data width for the FDI instance.
lp_retimer_crd	When asserted at a rising clock edge, it indicates a single credit return for the Retimer Receiver buffer. Each credit corresponds to 256B of mainband data (including Flit header and CRC etc.). This signal must NOT assert if a Retimer is not present. On FDI, this is an optional signal. It is permitted to have the Receiver buffers in the Protocol Layer for Raw Format only. If this is not exposed to Protocol Layer, Adapter must track credit at 256B granularity even for Raw Format and return credits to Physical Layer on RDI. When this is exposed on FDI, the Adapter must have the initial credits knowledge through other implementation specific means in order to advertise this to the remote Link partner during parameter exchanges.
lp_corrupt_crc	This signal is only applicable for CXL.cachemem in UCIe Flit Mode (i.e., the Adapter doing Retry) for CXL 256B Flit Mode. It is meant as a latency optimization that enables detection and containment for viral or poison using the Adapter to corrupt CRC of outgoing Flit. It is recommended to corrupt CRC by performing a bitwise XOR of the computed CRC with the syndrome 138Eh. The syndrome was computed such that no 1-bit or 2-bit errors alias to this syndrome, and it has the least probability of aliasing with 3-bit errors. For Standard 256B Flits, Protocol Layer asserts this along with **lp_valid** for the last chunk of the Flit that needs containment. Adapter corrupts CRC for both of the 128B halves of the Flit which had this set. It also must make sure to overwrite this flit (with the next flit sent by the Protocol Layer) in the Tx Retry buffer. For Latency-Optimized 256B Flits, Protocol Layer asserts this along with **lp_valid** for the last chunk of the 128B Flit half that needs containment. If **lp_corrupt_crc** is asserted on the first 128B half of the Flit, Protocol Layer must assert it on the second 128B half of the Flit as well. The very next Flit from the Protocol Layer after this signal has been asserted must carry the information relevant for viral, as defined in the CXL specification. If this was asserted on the second 128B half of the Flit only, it is the responsibility of the Protocol Layer to send the first 128B half exactly as before, and insert the viral information in the second half of the Flit. Adapter corrupts CRC for the 128B half of the Flit which had this set. It also must make sure to overwrite this flit (with the next flit sent by the Protocol Layer) in the Tx Retry buffer.
lp_dllp[NDLLP-1:0]	Protocol Layer to Adapter transfer of DLLP bytes. This is not used for 68B Flit Mode, CXL.cachemem or Streaming Protocols. For a 64B data path on lp_data, it is recommended to assign NDLLP >= 8, so that 1 DLLP per Flit can be transferred from the Protocol Layer to the Adapter on average. The Adapter is responsible for inserting DLLP into DLP bytes 2:5 if the Flit packing rules permit it. See Section 8.2.4.1 for additional rules.
lp_dllp_valid	Indicates valid DLLP transfer on **lp_dllp**. DLLP transfers are not subject to backpressure by **pl_trdy** (the Adapter must have storage for different types of DLLP and this can be overwritten so that the latest DLLPs are sent to remote Link partner). DLLP transfers are subject to backpressure by **pl_stallreq** - Protocol Layer must stop DLLP transfers at DLLP Flit aligned boundary before giving **lp_stallack** or requesting PM.
lp_dllp_ofc	Indicates that the corresponding DLLP bytes on **lp_dllp** follow the Optimized_Update_FC format. It must stay asserted for the entire duration of the DLLP transfer on **lp_dllp**.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-2. FDI signal list (Sheet 2 of 6)

Signal Name	Signal Description
lp_stream[7:0]	Protocol Layer to Adapter signal that indicates the stream ID to use with data. Each stream ID maps to a unique protocol and stack. 00h : Reserved 01h : Stack 0 : PCIe 02h : Stack 0 : CXL.io 03h : Stack 0 : CXL.cachemem 04h : Stack 0 : Streaming protocol 11h : Stack 1 : PCIe 12h : Stack 1 : CXL.io 13h : Stack 1 : CXL.cachemem 14h : Stack 1 : Streaming protocol Other encodings are Reserved.
pl_trdy	The Adapter is ready to accept data. Data is accepted by the Adapter when **pl_trdy**, **lp_valid**, and **lp_irdy** are asserted at the rising edge of **lclk**.
pl_valid	Adapter to Protocol Layer indication that data is valid on **pl_data**.
pl_data[NBYTES-1:0][7:0]	Adapter to Protocol Layer data, where NBYTES equals the number of bytes determined by the data width for the FDI instance.
pl_retimer_crd	When asserted at a rising clock edge, it indicates a single credit return from the Retimer. Each credit corresponds to 256B of mainband data (including Flit header and CRC etc.). This signal must NOT assert if a Retimer is not present. On FDI, this is an optional signal. It is permitted to expose these credits to Protocol Layer for Raw Format only. If this is not exposed to Protocol Layer, Adapter must track credit at 256B granularity even for Raw Format and back-pressure the Protocol Layer using **pl_trdy**. When this is exposed on FDI, the Adapter converts the initial credits received from the Retimer over sideband to credit returns to the Protocol Layer on this bit after Adapter LSM has moved to Active state.
pl_dllp[NDLLP-1:0]	Adapter to Protocol Layer transfer of DLLP bytes. This is not used for 68B Flit Mode, CXL.cachemem or Streaming Protocols. For a 64B data path on **pl_data**, it is recommended to assign NDLLP >= 8, so that 1 DLLP per Flit can be transferred from the Adapter to the Protocol Layer, on average. The Adapter is responsible for extracting DLLP from DLP bytes 2:5 if a Flit Marker is not present. The Adapter is also responsible for indicating Optimized_Update_FC format by setting **pl_dllp_ofc** = 1 for the corresponding transfer on FDI.
pl_dllp_valid	Indicates valid DLLP transfer on **pl_dllp**. DLLPs can be transferred to the Protocol Layer whenever valid Flits can be transferred on **pl_data**. There is no backpressure and the Protocol Layer must always sink DLLPs.
pl_dllp_ofc	Indicates that the corresponding DLLP bytes on **pl_dllp** follow the Optimized_Update_FC format. It must stay asserted for the entire duration of the DLLP transfer on **pl_dllp**.
pl_stream[7:0]	Adapter to Protocol Layer signal that indicates the stream ID to use with data. Each stream ID maps to a unique protocol. 00h : Reserved 01h : Stack 0 : PCIe 02h : Stack 0 : CXL.io 03h : Stack 0 : CXL.cachemem 04h : Stack 0 : Streaming protocol 11h : Stack 1 : PCIe 12h : Stack 1 : CXL.io 13h : Stack 1 : CXL.cachemem 14h : Stack 1 : Streaming protocol Other encodings are Reserved.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-2. FDI signal list (Sheet 3 of 6)

Signal Name	Signal Description
pl_flit_cancel	Adapter to Protocol Layer indication to dump a Flit. This enables latency optimizations on the Receiver data path when CRC checking is enabled in the Adapter. It is not applicable for Raw Format or 68B Flit Format. For Standard 256B Flit, it is required to have a fixed number of clock cycle delay between the last chunk of a Flit transfer and the assertion of **pl_flit_cancel**. This delay is fixed to be 1 cycle (i.e., the cycle after the last chunk transfer of a Flit). When this signal is asserted, Protocol Layer must not consume the associated Flit. For Latency-Optimized 256B Flits, it is required to have a fixed number of clock cycle delay between the last chunk of a 128B half Flit transfer and the assertion of **pl_flit_cancel**. This delay is fixed to be 1 cycle (i.e., the cycle after the last transfer of the corresponding 128B chunk). When this signal is asserted, Protocol Layer must not consume the associated Flit half. When this mode is supported, Protocol Layer must support it for all applicable Flit Formats associated with the corresponding protocol. Adapter must guarantee this to be a single cycle pulse when dumping a Flit or Flit half. It is the responsibility of the Adapter to ensure that the canceled Flits or Flit halves are eventually replayed on the interface without cancellation in the correct order once they pass CRC after Retry etc. See Section 8.2.5 for examples.
lp_state_req[3:0]	Protocol Layer request to Adapter to request state change. Encodings as follows: 0000b : NOP 0001b : Active 0100b : L1 1000b : L2 1001b : LinkReset 1011b : Retrain 1100b : Disabled All other encodings are reserved.
lp_linkerror	Protocol Layer to Adapter indication that an error has occurred which requires the Link to go down. Adapter must propagate this request to RDI, and move the Adapter LSMs (and CXL vLSMs if applicable) to LinkError state once RDI is in LinkError state. It must stay there as long as **lp_linkerror**=1. The reason for having this be an indication decoupled from regular state transitions is to allow immediate action on part of the Protocol Layer and Adapter in order to provide the quickest path for error containment when applicable (for example, a viral error escalation could map to the LinkError state)
pl_state_sts[3:0]	Adapter to Protocol Layer Status indication of the Interface. Encodings as follows: 0000b : Reset 0001b : Active 0011b : Active.PMNAK 0100b : L1 1000b : L2 1001b : LinkReset 1010b : LinkError 1011b : Retrain 1100b : Disabled All other encodings are reserved. The status signal is permitted to transition from Adapter autonomously when applicable. For example the Adapter asserts the Retrain status when it decides to enter retraining either autonomously or when requested by remote agent. For PCIe/Streaming protocols, the Adapter LSM is exposed as **pl_state_sts** to the Protocol Layer. For CXL protocol, the ARB/MUX vLSM is exposed as **pl_state_status** to the Protocol Layer. The Link Status is considered to be Up from Protocol Layer perspective when FDI status is Active, Active.PMNAK, Retrain, L1 or L2. The Link Status is considered Down for other states of FDI.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-2. FDI signal list (Sheet 4 of 6)

Signal Name	Signal Description
pl_inband_pres	Adapter to the Protocol Layer indication that the Die-to-Die Link has finished negotiation of parameters with remote Link partner and is ready for transitioning the FDI Link State Machine (LSM) to Active. Once it transitions to 1b, this must stay 1b until FDI moves to Active or LinkError. It stays asserted while FDI is in Retrain, Active.PMNAK, L1 or L2. It must de-assert during LinkReset, Disabled or LinkError states.
pl_error	Adapter to the Protocol Layer indication that it has detected a framing related error. It is pipeline matched with the receive data path. It must also assert if **pl_error** was asserted on RDI by the Physical Layer for a Flit which the Adapter is forwarding to the Protocol Layer. It is permitted for Protocol Layer to use **pl_error** indication to log correctable errors when Retry is enabled from the Adapter. The Adapter must finish any partial Flits sent to the Protocol Layer and assert **pl_flit_cancel** in order to prevent consumption of that Flit by the Protocol Layer. Adapter must initiate Link Retrain on RDI following this, if it was a framing error detected by the Adapter.
pl_cerror	Adapter to the Protocol Layer indication that a correctable error was detected that does not affect the data path and will not cause Retrain on the Link. The Protocol Layer must OR the **pl_error** and **pl_cerror** signals for Correctable Error Logging. The Adapter must OR any internally detected errors with the **pl_cerror** indication on RDI and forward it to the Protocol Layer.
pl_nferror	Adapter to the Protocol Layer indication that a non-fatal error was detected. This is used by Protocol Layer for error logging and corresponding escalation to software. The Adapter must OR any internally detected errors with **pl_nferror** on RDI and forward the result on FDI.
pl_trainerror	Indicates a fatal error from the Adapter. Adapter must transition **pl_state_sts** to LinkError if not already in LinkError state. Implementations are permitted to map any fatal error to this signal that require upper layer escalation (or interrupt generation) depending on system level requirements.
pl_rx_active_req	Adapter asserts this signal to request the Protocol Layer to open its Receiver's data path and get ready for receiving protocol data or Flits. The rising edge of this signal must be when **pl_state_sts** is Reset, Retrain or Active. Together with **lp_rx_active_sts**, it forms a four way handshake. See Section 8.2.7 for rules related to this handshake.
lp_rx_active_sts	Protocol Layer responds to **pl_rx_active_req** after it is ready to receive and parse protocol data or Flits. Together with **pl_rx_active_req**, it forms a four way handshake. See Section 8.2.7 for rules related to this handshake.
pl_protocol[2:0]	Adapter indication to Protocol Layer the protocol that was negotiated during training. It has the following encodings: 000b - PCIe 011b - CXL.1 [Single protocol, i.e., CXL.io] 100b - CXL.2 [Multi-protocol, Type 1 device] 101b - CXL.3 [Multi-protocol, Type 2 device] 110b - CXL.4 [Multi-protocol, Type 3 device] 111b - Streaming Protocol Other encodings are Reserved
pl_protocol_flitfmt[3:0]	This indicates the negotiated Format. See Chapter 3.0 for the definitions of these formats. 0001b - Format 1 : Raw Format 0010b - Format 2 : 68B Flit Format 0011b - Format 3 : Standard 256B End Header Flit Format 0100b - Format 4 : Standard 256B Start Header Flit Format 0101b - Format 5 : Latency-Optimized 256B without Optional Bytes Flit Format 0110b - Format 6 : Latency-Optimized 256B with Optional Bytes Flit Format Other encodings are Reserved

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-2. FDI signal list (Sheet 5 of 6)

Signal Name	Signal Description
pl_protocol_vld	Indication that **pl_protocol**, and **pl_protocol_flitfmt** have valid information. This is a level signal, asserted when the Adapter has determined the appropriate protocol, but must only de-assert again after subsequent transitions to LinkError or Reset state depending on the Link state machine transitions. Protocol Layer must sample and store **pl_protocol** and **pl_protocol_flitfmt** when **pl_protocol_vld** = 1 and **pl_state_sts** = Reset and **pl_inband_pres** = 1. It must treat this saved value as the negotiated protocol until **pl_state_sts** = Reset and **pl_inband_pres** = 0. The Adapter must ensure that if **pl_inband_pres** = 1, **pl_protocol_vld** = 1 and **pl_state_sts** = Reset, then **pl_protocol** and **pl_protocol_flitfmt** are the correct values that can be sampled by the Protocol Layer.
pl_stallreq	Adapter request to Protocol Layer to flush all Flits for state transition and not prepare any new Flits. See Section 8.2.6 for details.
lp_stallack	Protocol Layer to Adapter indication that the Flits are aligned and stalled (if **pl_stallreq** was asserted). It is strongly recommended that this response logic be on a global free running clock, so the Protocol Layer can respond to **pl_stallreq** with **lp_stallack** even if other significant portions of the Protocol Layer are clock gated.
pl_phyinrecenter	Adapter indication to Protocol Layer that the Link is doing training or retraining (i.e., RDI has **pl_phyinrecenter** asserted or the Adapter LSM has not moved to Active yet). If this is asserted during a state where clock gating is permitted, the **pl_clk_req/lp_clk_ack** handshake must be performed with the upper layer. The upper layers are permitted to use this to update the "Link Training/Retraining" bit in the UCIe Link Status register.
pl_phyinl1	Adapter indication to Protocol Layer that the Physical Layer is in L1 power management state (i.e., RDI is in L1 state).
pl_phyinl2	Adapter indication to Protocol Layer that the Physical Layer is in L2 power management state (i.e., RDI is in L2 state).
pl_speedmode[2:0]	Current Link speed. The following encodings are used: 000b: 4GT/s 001b: 8GT/s 010b: 12GT/s 011b: 16GT/s 100b: 24GT/s 101b: 32GT/s other encodings are reserved. The Protocol Layer must only consider this signal to be relevant when the FDI state is Active or Retrain. For multi-module configurations, all modules must operate at the same speed.
pl_lnk_cfg[2:0]	Current Link Configuration. Indicates the current operating width of a module. 001b: x8 010b: x16 011b: x32 100b: x64 101b: x128 110b: x256 other encodings are reserved. The Protocol Layer must only consider this signal to be relevant when the FDI state is Active or Retrain. This is the total width across all Active modules for the corresponding FDI instance.
pl_clk_req	Request from the Adapter to remove clock gating from the internal logic of the Protocol Layer. This is an asynchronous signal from the Protocol Layer's perspective since it is not tied to **lclk** being available in the Protocol Layer. Together with **lp_clk_ack**, it forms a four-way handshake to enable dynamic clock gating in the Protocol Layer. When dynamic clock gating is supported, the Protocol Layer must use this signal to exit clock gating before responding with **lp_clk_ack**. If dynamic clock gating is not supported, it is permitted for the Adapter to tie this signal to 1b.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-2. FDI signal list (Sheet 6 of 6)

Signal Name	Signal Description
lp_clk_ack	Response from the Protocol Layer to the Adapter acknowledging that its clocks have been ungated in response to **pl_clk_req**. This signal is only asserted when **pl_clk_req** is asserted, and de-asserted after **pl_clk_req** has de-asserted. When dynamic clock gating is not supported by the Protocol Layer, it must stage **pl_clk_req** internally for one or more clock cycles and turn it around as **lp_clk_ack**. This way it will still participate in the handshake even though it does not support dynamic clock gating.
lp_wake_req	Request from the Protocol Layer to remove clock gating from the internal logic of the Adapter. This is an asynchronous signal relative to **lclk** from the Adapter's perspective since it is not tied to **lclk** being available in the Adapter. Together with **pl_wake_ack**, it forms a four-way handshake to enable dynamic clock gating in the Adapter. When dynamic clock gating is supported, the Adapter must use this signal to exit clock gating before responding with **pl_wake_ack**. If dynamic clock gating is not supported, it is permitted for the Protocol Layer to tie this signal to 1b.
pl_wake_ack	Response from the Adapter to the Protocol Layer acknowledging that its clocks have been ungated in response to **lp_wake_req**. This signal is only asserted after **lp_wake_req** has asserted, and is de-asserted after **lp_wake_req** has de-asserted. When dynamic clock gating is not supported by the Adapter, it must stage **lp_wake_req** internally for one or more clock cycles and turn it around as **pl_wake_ack**. This way it will still participate in the handshake even though it does not support dynamic clock gating.
pl_cfg[NC-1:0]	This is the sideband interface from the Adapter to the Protocol Layer. See Chapter 6.0 for details. NC is the width of the interface. Supported values are 8, 16, and 32.
pl_cfg_vld	When asserted, indicates that **pl_cfg** has valid information that should be consumed by the Protocol Layer.
pl_cfg_crd	Credit return for sideband packets from the Adapter to the Protocol Layer for sideband packets. Each credit corresponds to 64 bits of header and 64 bits of data. Even transactions that don't carry data or carry 32 bits of data consume the same credit and the Receiver returns the credit once the corresponding transaction has been processed or de-allocated from its internal buffers. See Section 6.1.3.1 for additional flow control rules. Because the advertised credits are design parameters, the Protocol Layer transmitter updates the credit counters with initial credits on domain reset exit, and no initialization credits are returned over the interface. Credit returns must follow the same rules of clock gating exit handshakes as the sideband packets to ensure that no credit returns are dropped by the receiver of the credit returns.
lp_cfg[NC-1:0]	This is the sideband interface from Protocol Layer to the Adapter. See Chapter 6.0 for details. NC is the width of the interface. Supported values are 8, 16, and 32.
lp_cfg_vld	When asserted, indicates that **lp_cfg** has valid information that should be consumed by the Adapter.
lp_cfg_crd	Credit return for sideband packets from the Protocol Layer to the Adapter for sideband packets. Each credit corresponds to 64 bits of header and 64 bits of data. Even transactions that don't carry data or carry 32 bits of data consume the same credit and the Receiver returns the credit once the corresponding transaction has been processed or de-allocated from its internal buffers. See Section 6.1.3.1 for additional flow control rules. Because the advertised credits are design parameters, the Adapter transmitter updates the credit counters with initial credits on domain reset exit, and no initialization credits are returned over the interface. Credit returns must follow the same rules of clock gating exit handshakes as the sideband packets to ensure that no credit returns are dropped by the receiver of the credit returns.

8.2.1 Interface reset requirements

FDI does not define a separate interface signal for reset; however, it is required that the logic entities on both sides of FDI are in the same reset domain and the reset for each side is derived from the same source.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.2.2 Interface clocking requirements

FDI requires both sides of the interface to be on the same clock domain. Moreover, the clock domain for sideband interface (***cfg***) is the same as the mainband signals.

Each side is permitted to instantiate clock crossing FIFOs internally if needed, as long as it does not violate the requirements at the interface itself.

It is important to note that there is no back pressure possible from the Protocol Layer to the Adapter on the main data path. So any clock crossing related logic internal to the Protocol Layer must take this into consideration.

8.2.3 Dynamic clock gating

Dynamic coarse clock gating is permitted in the Adapter and Protocol Layer when **pl_state_sts** is Reset, LinkReset, Disabled or PM states. This section defines the rules around entry and exit of clock gating. Note that clock gating is not permitted in LinkError states - it is expected that the UCIe usages will enable error handlers to make sure the Link is not stuck in a LinkError state, if the intent is to save power when a Link is in an error state.

8.2.3.1 Rules and description for lp_wake_req/pl_wake_ack handshake

Protocol Layer can request removal of clock gating of the Adapter by asserting **lp_wake_req** (asynchronous to **lclk** availability in the Adapter). All Adapter implementations must respond with a **pl_wake_ack** (synchronous to **lclk**). The extent of internal clock ungating when **pl_wake_ack** is asserted is implementation-specific, but lclk must be available by this time to enable FDI transitions from the Protocol Layers. The Wake Req/Ack is a full handshake and it must be used for state transition requests (on **lp_state_req** or **lp_linkerror**) when moving away from Reset or PM states. It must also be used for sending packets on the sideband interface.

Rules for this handshake:

1. Protocol Layer asserts **lp_wake_req** to request ungating of clocks by the Adapter.
2. The Adapter asserts **pl_wake_ack** to indicate that clock gating has been removed. There must be at least one clock cycle bubble between **lp_wake_req** assertion and **pl_wake_ack** assertion.
3. **lp_wake_req** must de-assert before **pl_wake_ack** de-asserts. It is the responsibility of the Protocol Layer to control the specific scenario of de-assertion. As an example, when performing the handshake for a state request, it is permitted to keep **lp_wake_req** asserted until it observes the desired state status. Protocol Layer is also permitted to keep **lp_wake_req** asserted through states where clock gating is not allowed in the Adapter (i.e., Active, LinkError or Retrain).
4. **lp_wake_req** should not be the only consideration for Adapter to perform clock gating, it must take into account **pl_state_sts** and other internal or Link requirements before performing global and/or local clock gating.
5. When performing **lp_wake_req/pl_wake_ack** handshake for **lp_state_req** transitions or **lp_linkerror** transition, the Protocol Layer is permitted to not wait for **pl_wake_ack** before changing **lp_state_req** or **lp_linkerror**.
6. When performing **lp_wake_req/pl_wake_ack** handshake for **lp_cfg** transitions, Protocol Layer must wait for **pl_wake_ack** before changing **lp_cfg** or **lp_cfg_vld**. Because **lp_cfg** can have multiple transitions for a single packet transfer, it is necessary to make sure that the Adapter clocks are up before transfer begins.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

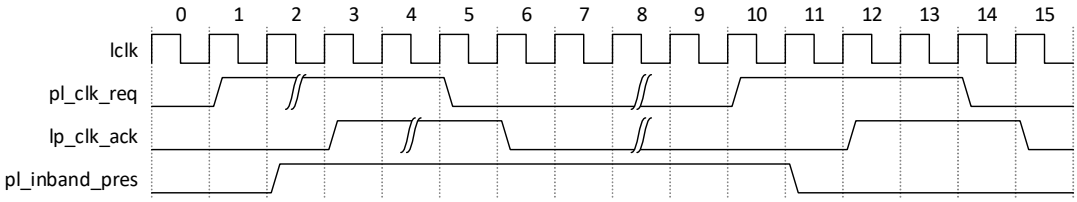
8.2.3.2 Rules and description for pl_clk_req/lp_clk_ack handshake

Adapter is allowed to initiate **pl_clk_req/lp_clk_ack** handshake at any time and the Protocol Layer must respond.

Rules for this handshake:

1. Adapter asserts **pl_clk_req** to request removal of clock gating by the Protocol Layer. This can be done anytime, and independent of current FDI state.
2. The Protocol Layer asserts **lp_clk_ack** to indicate that clock gating has been removed. There must be at least one clock cycle bubble between **pl_clk_req** assertion and **lp_clk_ack** assertion.
3. **pl_clk_req** must de-assert before **lp_clk_ack**. It is the responsibility of the Adapter to control the specific scenario of de-assertion, after the required actions for this handshake are completed.
4. **pl_clk_req** should not be the only consideration for the Protocol Layer to perform clock gating, it must take into account **pl_state_sts** and other protocol-specific requirements before performing trunk and/or local clock gating.
5. The Adapter must use this handshake to ensure transitions of **pl_inband_pres**, **pl_phyinl1**, **pl_phyinl2**, **pl_phyinrecenter**, and **pl_rx_active_req** have been observed by the Protocol Layer. Since these are level oriented signals, the Adapter is permitted to let the signal transition without waiting for **lp_clk_ack**. When this is done during initial Link bring up, it is strongly recommended for the Adapter to keep **pl_clk_req** asserted until the state status transitions away from Reset to a state where clock gating is not permitted or until the state status is Reset and **pl_inband_pres** de-asserts.

Figure 8-11. Example Waveform Showing Handling of Level Transition



6. The Adapter must also perform this handshake before transition to LinkError state from Reset, LinkReset, Disabled or PM state (especially when the LinkError transition occurs by the Adapter without being directed by the Protocol Layer). It is permitted to assert **pl_clk_req** before the state change, in which case it must stay asserted until the state status transitions. It is also permitted to assert **pl_clk_req** after the state status transition, but in this case Adapter must wait for **lp_clk_ack** before performing another state transition.
7. The Adapter must also perform this handshake when the status is PM and remote Link partner is requesting PM exit. For exit from Reset, LinkReset, Disabled or PM states to a state that is not LinkError, it is required to assert **pl_clk_req** before the status change, and in this case it must stay asserted until the state status transitions away from Reset or PM.
8. The Adapter must also perform this handshake for sideband transfers from the Adapter to the Protocol Layer. When performing the handshake for **pl_cfg** transitions, Adapter must wait for **lp_clk_ack** before changing **pl_cfg** or **pl_cfg_vld**. Because **pl_cfg** can have multiple transitions for a single packet transfer, it is necessary to make sure that the Protocol Layer clocks are up before transfer begins.

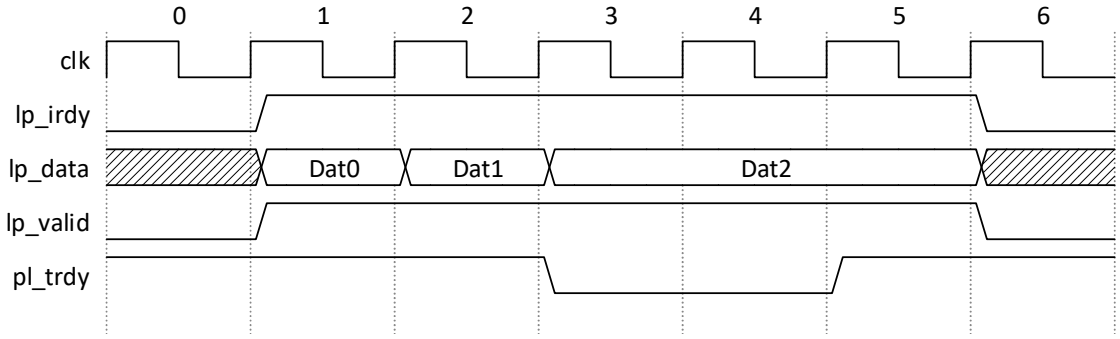
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

When clock-gated in Reset states, Protocol Layers that rely on dynamic clock gating to save power must wait in clock gated state for **pl_inband_pres**=1. The Adapter will request clock gating exit when it transitions **pl_inband_pres**, and the Protocol Layer must wait for **pl_inband_pres** assertion before requesting **lp_state_req** = ACTIVE. If **pl_inband_pres** de-asserts while **pl_state_sts** = Reset, then the Protocol Layer is permitted to return to clock-gated state after moving **lp_state_req** to NOP.

8.2.4 Data Transfer

As indicated in the signal list descriptions, when Protocol Layer is sending data to the Adapter, data is transferred when **lp_irdy**, **pl_trdy** and **lp_valid** are asserted. Figure 8-12 shows an example waveform for data transfer from the Protocol Layer to the Adapter. Data is transmitted on clock cycles 1, 2, and 5. No assumption should be made by Protocol Layer about when **pl_trdy** can de-assert or for how many cycles it remains de-asserted before it is asserted again, unless explicitly guaranteed by the Adapter. If a Flit transfer takes multiple clock cycles, the Protocol Layer is not permitted to insert bubbles in the middle of a Flit transfer (i.e., **lp_valid** and **lp_irdy** must be asserted continuously until the Flit transfer is complete. Of course, data transfer can stall because of **pl_trdy** de-assertion).

Figure 8-12. Data Transfer from Protocol Layer to Adapter



As indicated in the signal list descriptions, when Adapter is sending data to the Protocol layer, there is no back-pressure mechanism, and data is transferred whenever **pl_valid** is asserted. The Adapter is permitted to insert bubbles in the middle of a Flit transfer and the Protocol Layer must be able to handle that.

8.2.4.1 DLLP transfer rules for 256B Flit Mode

For PCIe and CXL.io 256B Flits (both Standard and Latency-Optimized), FDI provides a separate signal for DLLP transfers from the Protocol Layer to the Adapter and vice-versa. Since the DLLPs have to bypass the Retry buffer, the separate signal enables the Adapter to insert DLLPs into the Flits from the Protocol Layer or the Retry buffer, if it is permitted to do so per the Flit packing rules of the corresponding Flit Format. Rules relevant for FDI operation (per FDI instance) are outlined below:

For the Transmitting side:

- Protocol Layer is responsible for sending the relevant DLLPs at the rate defined by the underlying Protocol to prevent timeouts of DLLP exchanges. If the Protocol Layer has no TLPs to send, it must insert NOP Flits to ensure that the Adapter gets an opportunity to insert the DLLP bytes.
- When transferring DLLP or Optimized_Update_FC, the least significant byte is sent over Byte 0 of the FDI bus, the next byte over Byte 1 and so on. When the transfer is over multiple chunks across FDI, Byte 0 is transferred on the first chunk LSB, Byte 1 following it and so on.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- The Adapter must have storage for at least 1 DLLP of every unique DLLP encoding (including Optimized_Update_FC) per supported VC that is possible for transfer to remote Link partner. The Adapter tracks pending DLLPs and schedules them on the next available opportunity for the relevant Flits. Credit update DLLPs must not be reordered for a VC by the Adapter. It is however permitted to discard a pending credit DLLP if the Protocol Layer presented a new credit DLLP of the same FC and VC. This extends to Optimized_Update_FC packets; i.e., it is permitted to discard any pending NP or P Update FC DLLPs, if the Protocol Layer transferred an Optimized_Update_FC for the corresponding VC.

On the Receiving side:

- The Adapter must extract DLLPs from received Flits of the corresponding protocol and forward them to the Protocol Layer. The FDI signal width of **pl_dllp** must be wide enough to keep up with the maximum rate of DLLPs that could be received from the Link.
- When transferring DLLP over multiple chunks across FDI, Byte 0 is transferred on the first chunk LSB, Byte 1 following it and so on.
- The Protocol Identifier corresponding to D2D Adapter in the Flit Header overlaps with the Flit usage of NOP Flits defined in PCIe and CXL specifications. The Adapter must check for available DLLPs in these Flits as well. All 0 bits in the DLLP byte positions indicate a NOP DLLP, and must not be forwarded to the Protocol Layer.

8.2.5 Examples of pl_flit_cancel Timing Relationships

In all the examples shown in this section, a 64B datapath on FDI is shown, and "F0Bytes" in the figures correspond to "Flit 0 Bytes".

Figure 8-13 shows an example timing relationship for **pl_flit_cancel** and **pl_data** for Latency-Optimized Flits when the first Flit half fails CRC check. Both Flit halves are canceled by the Adapter in this example by asserting **pl_flit_cancel** one clock after the last chunk transfer of the corresponding Flit half. It is permitted for the Adapter to de-assert **pl_valid** on clock cycles 5 and 6 instead of canceling that Flit half; however, this might have implications to meeting physical design timing margins in the Adapter. The use of **pl_flit_cancel** allows the Adapter to perform the CRC check on the side without putting the CRC logic in the critical timing path of the data flow and thus permitting higher frequency operation for implementations. In the example shown, after replay flow the entire Flit is transferred to the Protocol Layer without canceling as CRC checks pass.

Figure 8-13. Example for pl_flit_cancel for Latency-Optimized Flits and CRC Error on First Flit Half

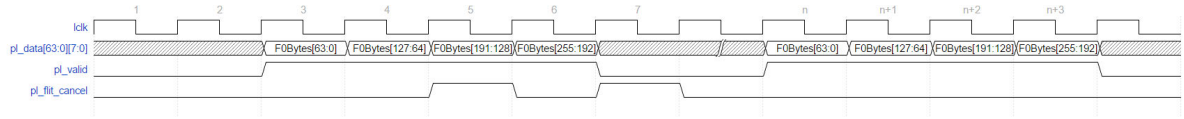


Figure 8-14 and Figure 8-15 show examples of two possible implementations of timing relationship for **pl_flit_cancel** and **pl_data** for Latency-Optimized Flits when the second Flit half fails CRC check. In both cases, the first half of the Flit is consumed by the Protocol Layer because it is not canceled by the Adapter (the data transferred on clock cycles 3 and 4).

In the first case (shown in Figure 8-14), after the replay flow, CRC passes, and the Adapter ensures that the Protocol Layer does not re-consume the first half again by asserting **pl_flit_cancel** for it. In this case, **pl_valid** asserts for the entire Flit, but only the second half is consumed because the first half was canceled on clock cycle (n+2).

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

In the second case (shown in Figure 8-15), after the replay flow, CRC passes, and the Adapter ensures that the Protocol Layer does not re-consume the first half again by not asserting **pl_valid** for it.

Figure 8-14. Example for pl_flit_cancel for Latency-Optimized Flits and CRC Error on Second Flit Half

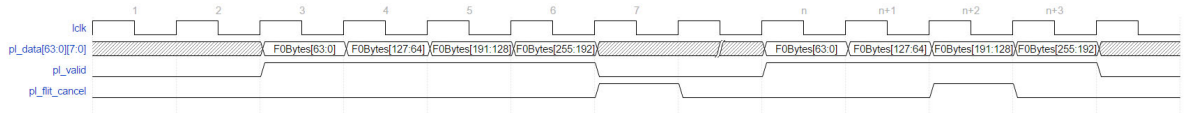


Figure 8-15. Example for pl_flit_cancel for Latency-Optimized Flits and CRC Error on Second Flit Half, Alternate Implementation Example

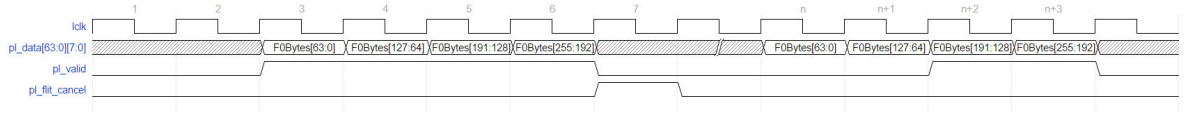
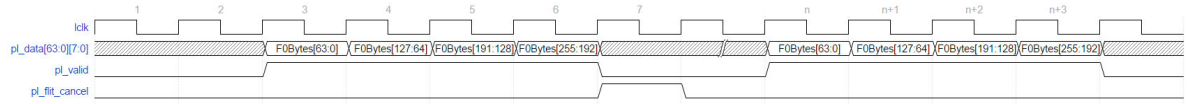


Figure 8-16 shows an example for a Standard 256B Flit. In this case, the CRC bytes are packed toward the end of the Flit and thus a CRC error on either of the two halves cancels the entire Flit. After replay flow, CRC passes, and the entire Flit is sent to the Protocol Layer without canceling it.

Figure 8-16. Example for pl_flit_cancel for Standard 256B Flits

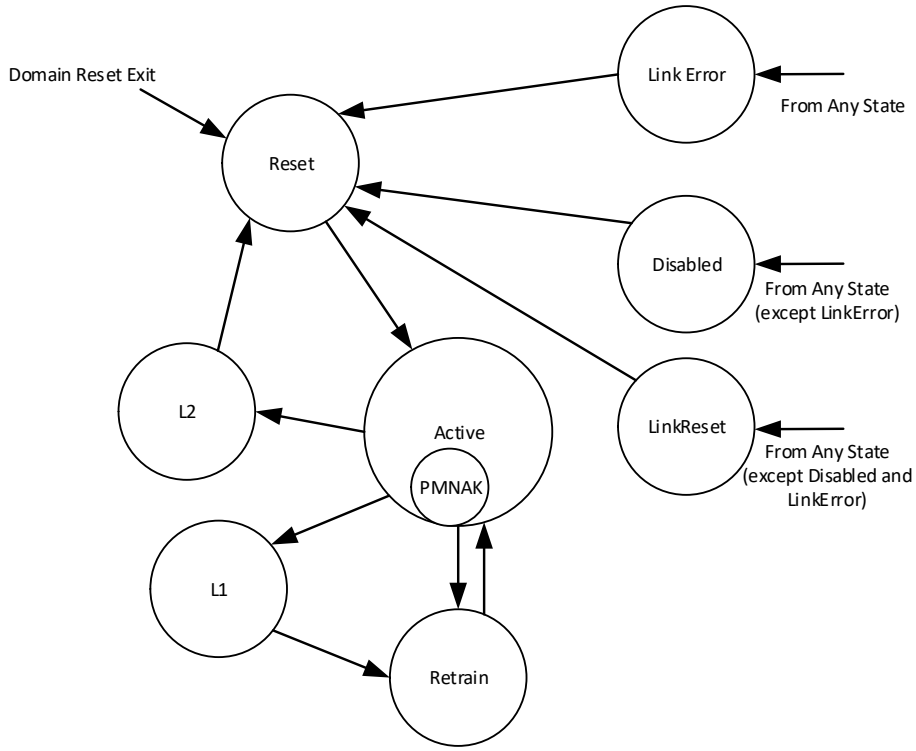


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.2.6 FDI State Machine

Figure 8-17 shows the FDI state machine.

Figure 8-17. FDI State Machine



8.2.7 Rx_active_req/Sts Handshake

The Adapter negotiates Active state transitions on FDI using sideband messages when the Adapter LSM is exposed to the Protocol Layer. Since the sideband Link is running slower than the mainband Link, the Adapter needs to make sure that the Protocol Layer's Receiver is already in Active state (even though **pl_state_sts** might not have moved to Active yet) before responding to the Active request sideband message from remote Link partner. Rx_active_req/Sts handshake facilitates this.

When CXL is sent over UCIe, ARB/MUX functionality is performed by the Adapter and CXL vLSMs are exposed on FDI. Although ALMPs are transmitted over mainband, the interface to the Protocol Layer is FDI and it follows the rules of Rx_active_req/Sts Handshake as well.

Rules for this handshake:

1. The Adapter (or ARB/MUX) asserts **pl_rx_active_req** to trigger the Protocol Layer to open its Receiver's data path for receiving protocol data or Flits. This signal does not affect the Transmitter data path (it must wait for **pl_state_sts** to move to Active and follow the rules of **pl_trdy**). **pl_rx_active_req** should have a rising edge only when **lp_rx_active_sts** = 0 and **pl_state_sts** is Reset, Retrain or Active.
2. The Protocol Layer asserts **lp_rx_active_sts** after **pl_rx_active_req** has asserted and when it is ready to receive protocol data or Flits. There must be at least one clock cycle delay between **lp_rx_active_sts** assertion and **lp_rx_active_sts** assertion to prevent a combinatorial loop.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

3. When **pl_rx_active_req** = 1 and **lp_rx_active_sts** = 1, the Receiver is in Active state if **pl_state_sts** is Reset, Retrain, or Active.
4. **pl_rx_active_req** should have a falling edge only when **lp_rx_active_sts** = 1. This must trigger Protocol Layer to de-assert **lp_rx_active_sts**, and this completes the transition of the Receiver away from Active state.
5. For graceful exit from Active state (i.e., a transition to PM, Retrain,LinkReset or Disabled states), both **pl_rx_active_req** and **lp_rx_active_sts** must de-assert before **pl_state_sts** transitions away from Active.
6. If **pl_rx_active_req** = 0 while **pl_state_sts** = Active, the Adapter must guarantee no Flits would be sent to the Protocol Layer (for example, this can happen if the Adapter LSM or RDI is in Retrain, but the vLSM exposed to Protocol Layer is still in Active). Thus, it is permitted to perform this handshake even when the state status on FDI remains Active throughout.
7. For Active to LinkError transition, it is permitted for **pl_state_sts** to transition to LinkError before **pl_rx_active_req** de-asserts, but both **pl_rx_active_req** and **lp_rx_active_sts** must de-assert before **pl_state_sts** transitions away from LinkError.

8.2.8 FDI Bring up flow

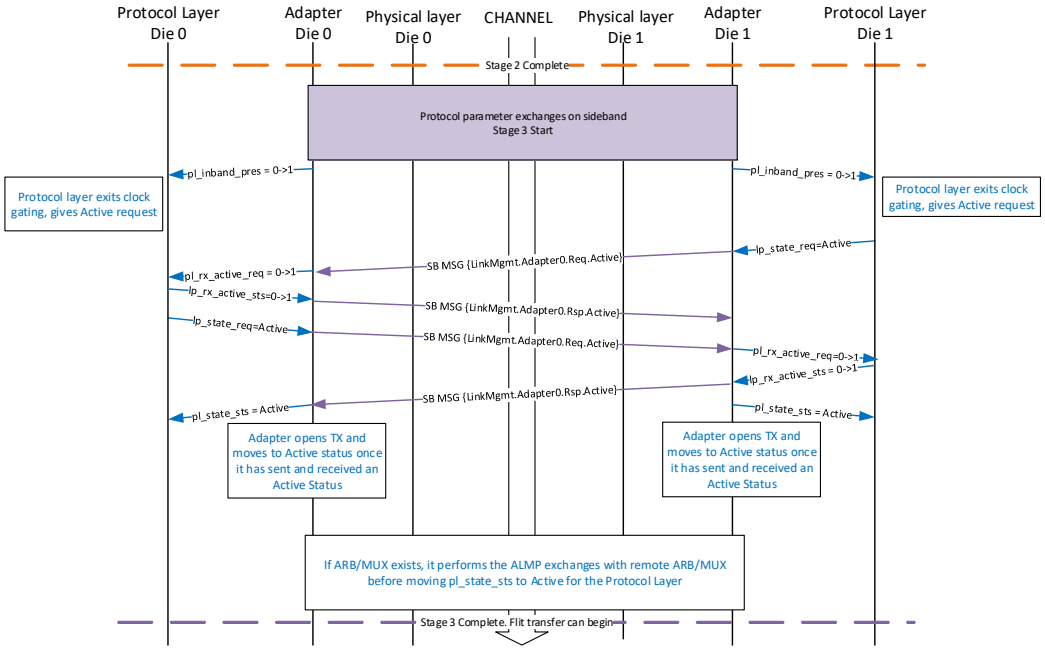
Figure 8-18 shows an example flow for Stage 3 of the Link bring up highlighting the transitions on FDI. This stage requires sequencing on FDI that coordinates the state transition from Reset to Active. If multiple stacks of protocol or ARB/MUX is present, the same sequence happens independently for each Protocol Layer stack. The flows on FDI are illustrated for Adapter 0 LSM in the sideband message encodings, however Adapter 1 LSM must send the sideband message encodings corresponding to Adapter 1 to its remote Link partner.

1. Once Adapter has completed transition to Active on RDI and successful parameter negotiation with the remote Link partner, it must do the **pl_clk_req** handshake with the Protocol Layer and reflect **pl_inband_pres**=1 on FDI. Note that the **pl_clk_req** handshake is not shown in the example flow in Figure 8-18
2. This is the trigger for Protocol Layer to request Active state. It is permitted for the Protocol Layer to wait unlit **pl_protocol_vld** = 1 before requesting Active. It must perform the **lp_wake_req** handshake as described in Section 8.2.3.1. Note that the **lp_wake_req** handshake is not shown in the example flow in Figure 8-18.
3. On sampling **lp_state_req** = Active, the Adapter must send the {LinkMgmt.Adapter0.Req.Active} sideband message to remote Link partner.
4. The Adapter must respond to the {LinkMgmt.Adapter.Req.Active} sideband message with a {LinkMgmt.Adapter.Rsp.Active} sideband message after making sure that the Protocol Layer's Receiver is ready. The {LinkMgmt.Adapter0.Rsp.Active} must only be sent after the Adapter has sampled **pl_rx_active_req** = **lp_rx_active_sts** = 1. As mentioned previously, the **pl_clk_req** handshake applies to **pl_rx_active_req** as well; it is permitted for the Adapter to keep **pl_clk_req** asserted continuously (once it has been asserted for **pl_inband_pres**) while doing the bring up flow.
5. If no ARB/MUX is present, once the Adapter has sent and received the {LinkMgmt.Adapter0.Rsp.Active} sideband message, it must transition **pl_state_sts** to Active for the Protocol Layer, and Flit transfer can begin.
6. If ARB/MUX is present, the sending and receipt of {LinkMgmt.Adapter0.Rsp.Active} sideband message opens up the ARB/MUX to perform ALMP exchanges over mainband and eventually transition the vLSMs to Active state.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Steps 3 to 6 constitute the "Active Entry Handshake" on FDI and must be performed for every entry to Active state. Active.PMNAK to Active transition is not considered here because Active.PMNAK is only a sub-state of Active.

Figure 8-18. FDI Bring up flow



8.2.9 FDI PM Flow

This section describes the sequencing and rules for PM entry and exit on FDI. The rules are the same for L1 or L2 entry. L1 exit transitions the state machine through Retrain to Active, whereas L2 exit transitions the state machine through Reset to Active. The flow illustrations in the section use L1 as an example. A "PM request" sideband message is {LinkMgmt.Adapter*.Req.L1} or {LinkMgmt.Adapter*.Req.L2}. A "PM Response" sideband message is {LinkMgmt.Adapter*.Rsp.L1} or {LinkMgmt.Adapter*.Rsp.L2}. The flows on FDI are illustrated for Adapter 0 LSM in the sideband message encodings; however, Adapter 1 LSM must send the sideband message encodings corresponding to Adapter 1 to its remote Link partner.

- The Protocol Layer requests PM entry on FDI after idle time criteria has been met. The criteria for idle time is implementation specific and could be dependent on the protocol. For PCIe and CXL.io protocols, PM DLLPs are **not** used to negotiate PM entry/exit when using D2D Adapter's Retry buffer (i.e., UCIe Flit Mode).
- If operating in UCIe Flit Mode, ARB/MUX is present within the D2D Adapter, and it follows the rules of *CXL Specification* (for 256B Flit Mode) to take the vLSMs to the corresponding PM state. Note that even for CXL 68B Flit Mode, the same ALMP rules as 256B Flit Mode are used. This is a simplification on UCIe, because ALMPs always go through the retry buffer. Once vLSMs are in the PM state, ARB/MUX requests the Adapter Link State Machine to enter the corresponding PM state. The Adapter Link State Machine transition to PM follows the same rules as outlined for Protocol Layer and Adapter below.
- If CXL or PCIe protocol has been negotiated, only the upstream port (UP) can initiate PM entry. This is done using a sideband message from the UP Adapter to the downstream port (DP) Adapter. DP Adapter must not initiate entry into PM. PM support is required for CXL and PCIe protocols. PM entry is considered successful and complete once UP receives a valid "PM Response" sideband

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

message. Figure 8-19 shows an example flow for CXL or PCIe protocol PM Entry on FDI and Adapter. Once the FDI status is PM for all Protocol Layers, the Adapter can request PM transition on RDI.

- If Streaming protocol has been negotiated or UCIe is in Raw Format, then both side Adapters must issue PM entry request through a sideband message once the conditions of PM entry have been satisfied. PM entry is considered successful and complete once both sides have received a valid "PM Response" sideband message. Figure 8-20 shows an example flow for symmetric protocols. Once the FDI status is PM for all Protocol Layers, the Adapter can request PM transition on RDI.
- Protocol Layer requests PM entry once it has blocked transmission of any new Protocol Layer Flits, by transitioning **lp_state_req** to L1 or L2 encoding. Once requested, the Protocol Layer cannot change this request until it observes the corresponding PM state, Retrain, Active.PMNAK or LinkError state on **pl_state_sts**; unless it is a DP Protocol Layer for PCIe or CXL. Once the FDI state is resolved, the Adapter must first bring it back to Active before processing any new PM requests from the Protocol Layer.
  - If the resolution is PM (upon successful PM entry) and the Protocol Layer needs to exit PM (or there is a pending Protocol Layer Active request from remote Link partner) then the Protocol Layer must initiate PM exit flow on FDI by requesting **lp_state_req** = Active. All PM entry related handshakes must have finished prior to this (for CXL/PCIe protocols, this is when UP has received a valid "PM Response" sideband message. For symmetric protocols, this is when both sides Adapter have received a valid "PM Response" sideband message).
  - If the resolution is Active.PMNAK, the Protocol Layer must initiate a request of Active on FDI. Once the status moves to Active, the Protocol Layer is permitted to re-request PM entry (if all conditions of PM entry are still met). Figure 8-21 shows an example of PM abort flow. The PM request could have been from either side depending on the configuration. Protocol Layer must continue receiving protocol data or Flits while the status is Active or Active.PMNAK.
  - DP Protocol Layer for PCIe or CXL is permitted to change request from PM to Active without waiting for PM or Active.PMNAK (the DP FDI will never have **pl_state_sts**=Active.PMNAK since it does not send "PM Request" sideband messages); however, it is still possible for the Adapter to initiate a stallreq/ack and complete PM entry if it was in the process of committing to PM entry when the Protocol Layer changed its request. In this scenario, the Protocol Layer will see **pl_state_sts** transition to PM and it is permitted to continue asking for the new state request.
  - If the resolution is LinkError, then the Link is down and it resets any outstanding PM handshakes.
- Adapter (UP port only if CXL or PCIe protocol), initiates a "PM request" sideband message once it samples a PM request on **lp_state_req** and has completed the StallReq/Ack handshake with the corresponding Protocol Layer and its Retry buffer is empty of Flits from the Protocol Layer that is requesting PM (all pending Acks have been received).
- If the Adapter LSM moves to Retrain while waiting for a "PM Response" sideband message, it must wait for the response. Once the response is received, it must transition back to Active before requesting a new PM entry. Note that the transition to Active requires Active Entry handshake with the remote Link partner, and that will cause the remote partner to exit PM. If the Adapter LSM receives a "PM Request" sideband message after it has transitioned to Retrain, it must immediately respond with {LinkMgmt.Adapter0.Rsp.PMNAK}.

Note: The precise timing of the remote Link partner that is observing Link Retrain is unknown; thus, the safer thing to do is to go to Active and redo the PM handshake when necessary for this scenario. There is a small probability that there might be an exit from PM and re-entry back in PM under certain scenarios.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

- Once the Adapter receives a "PM request" sideband message, it must respond to it within 2 us (the time is only counted during the Adapter LSM being in Active state):
  - if its local Protocol Layer is requesting PM, it must respond with the corresponding "PM Response" sideband message after finishing the StallReq/Ack handshake with its Protocol Layer and its Retry buffer being empty. If the current status is not PM, it must transition **pl_state_sts** to PM after responding to the sideband message.
  - If the current **pl_state_sts** = PM, it must respond with "PM Response" sideband message.
  - If **pl_state_sts** = Active and **lp_state_req** = Active and it remains this way for 1us after receiving the "PM Request" sideband message, it must respond with {LinkMgmt.Adapter0.Rsp.PMNAK} sideband message. The time is only counted during all the relevant state machines being in Active state.
- If the Adapter receives a "PM Response" sideband message in response to a "PM Request" sideband message, it must transition **pl_state_sts** on its local FDI to PM (if it is not currently in a PM state).
- If the Adapter receives a {LinkMgmt.Adapter0.Rsp.PMNAK} sideband message in response to a "PM Request" sideband message, it must transition **pl_state_sts** on its local FDI to Active.PMNAK state. It is permitted to retry PM entry handshake (if all conditions of PM entry are satisfied) at least 2us after receiving the {LinkMgmt.Adapter0.Rsp.PMNAK} sideband message OR if it received a corresponding "PM Request" sideband message from the remote Link partner.
- PM exit is initiated by the Protocol Layer requesting Active on FDI. After RDI is in Active, triggers the Adapter to initiate PM exit by performing the Active Entry handshakes on sideband. Figure 8-22 shows an example flow of PM exit on FDI when Adapter LSM is exposed.
  - PM exit handshake completion requires both Adapters to send as well as receive a {LinkMgmt.Adapter0.Rsp.Active} sideband message. Once this has completed, the Adapter is permitted to transition Adapter LSM to Active.
  - If **pl_state_sts** = PM and a {LinkMgmt.Adapter0.Req.Active} sideband message is received, the Adapter must initiate **pl_clk_req** handshake with the Protocol Layer, and transition Adapter LSM to Retrain (For L2 exit, the transition is to Reset). This must trigger the Protocol Layer to request Active on **lp_state_req** (if not already doing so), and this in turn triggers the Adapter to send {LinkMgmt.Adapter0.Req.Active} sideband message to the remote Link partner.

Note that the following figures are examples and do not show the **lp_wake_req**, **pl_clk_req**, and/or **pl_rx_active_req** handshakes. Implementations must follow the rules outlined for these handshakes in previous sections.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-19. PM Entry example for CXL or PCIe protocols

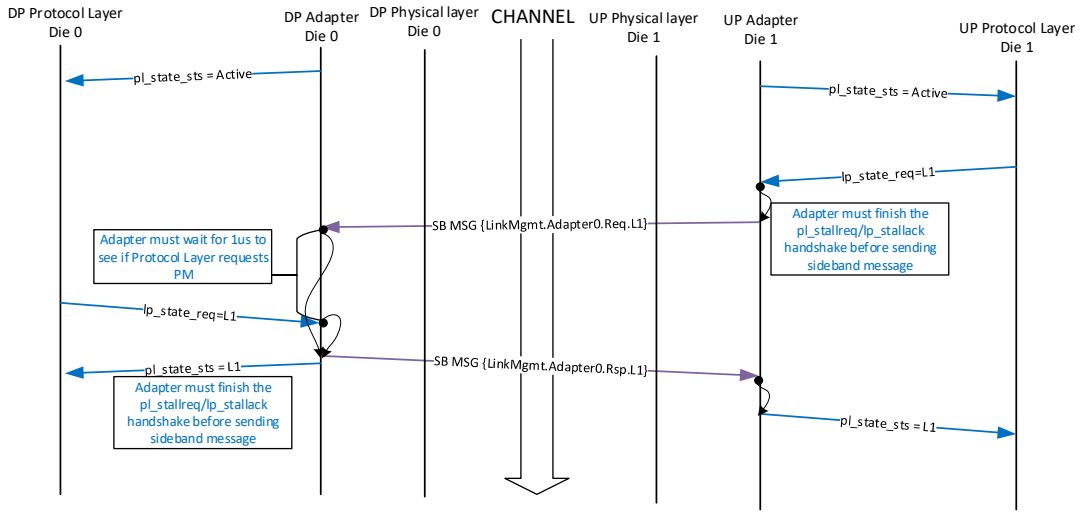
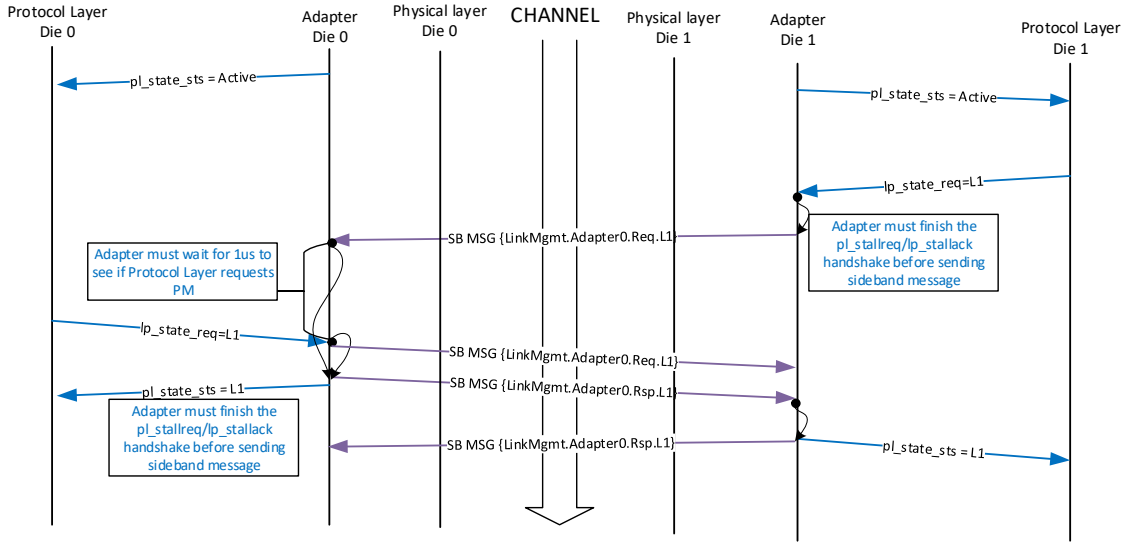


Figure 8-20. PM Entry example for symmetric protocol



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-21. PM Abort Example

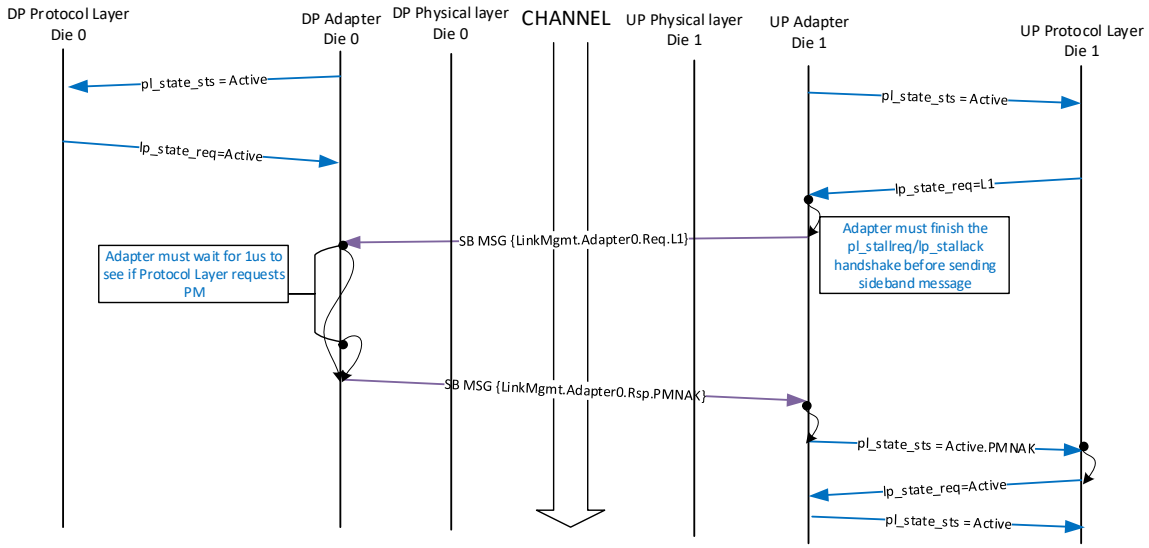
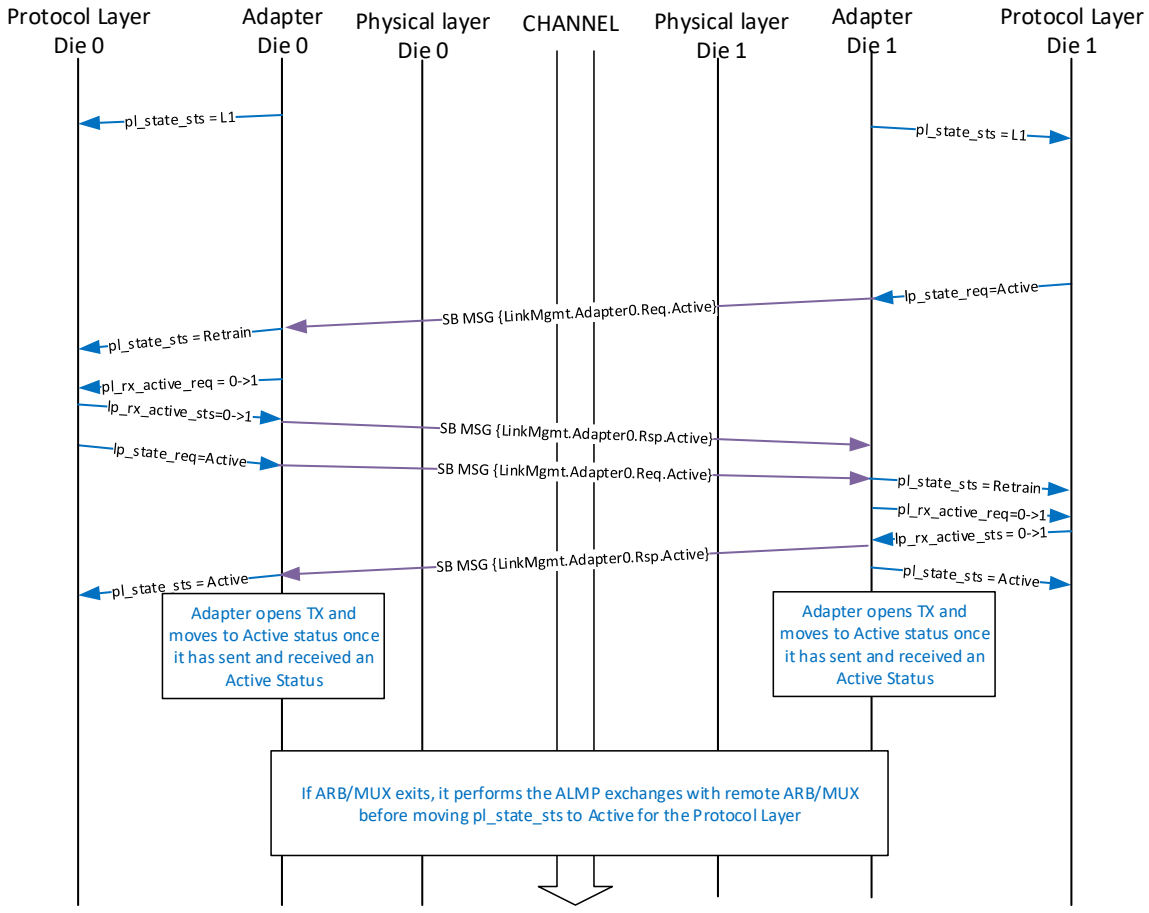


Figure 8-22. PM Exit Example



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3 Common rules for FDI and RDI

This section covers common set of rules applicable to FDI and RDI and cross-interactions between them. Any applicable differences are called out as well. To have common terminology for the common set of rules, Upper Layer is used to refer to Adapter for RDI, and Protocol Layer for FDI. Lower Layer is used to refer to Physical Layer for RDI and Adapter for FDI. In the following sections, "UCIe Flit Mode" refers to scenarios where the Retry buffer in the Adapter is being utilized and "UCIe Raw Format" refers to scenarios where the Retry buffer in the Adapter is not being utilized.

Because Active.PMNAK is a sub-state of Active, all rules that apply for Active are also applicable for Active.PMNAK; however the state status cannot move from Active.PMNAK directly to L1 or L2 due to the rules requiring the Upper Layer to request a transition to Active before requesting PM again.

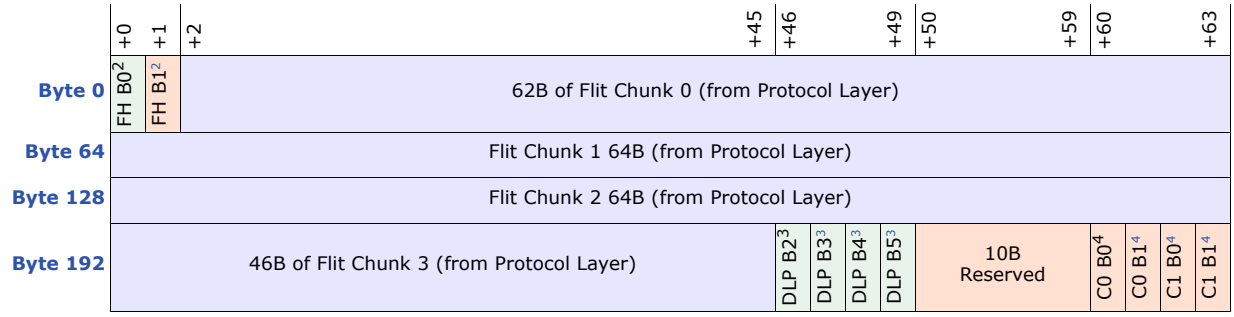
8.3.1 Byte Mapping for FDI and RDI

The Flit Format figures in Chapter 3.0 show examples of how a Flit is laid out on a 64B datapath when sent over FDI or RDI. Figure 8-23 shows an example of a CXL.io Standard 256B Start Header Flit for reference. Each Flit takes four data transfers across FDI or RDI when the data width is 64 Bytes. Each data transfer is referred to a Flit Chunk, numbered in increasing order within an entire Flit transfer.

For every data transfer, the Least Significant Byte from the corresponding Flit Chunk is mapped to Byte 0 on FDI (or RDI), the next Byte from the Flit is mapped to Byte 1 on FDI (or RDI), and so on. Within each Byte, bit 0 of the Byte from the Flit maps to bit 0 of the corresponding Byte on FDI (or RDI), and so on. The same mapping applies for both transmit and receive directions.

For example, in Transfer 0, Byte 0 of the Flit is mapped to Byte 0 of FDI (or RDI), Byte 1 of the Flit is mapped to Byte 1, and so on. In transfer 1, Byte 64 of the Flit is mapped to Byte 0 of FDI (or RDI), Byte 65 of the Flit is mapped to Byte 1 of FDI (or RDI) and so on. This example is illustrated in Figure 8-24. Data transfers follow the rules outlined in Section 8.1.4 for RDI and Section 8.2.4 for FDI and hence don't necessarily correspond to consecutive clock cycles.

Figure 8-23. CXL.io Standard 256B Start Header Flit Format Example[1]



1. See Figure 2-1 for color mapping.
2. Flit Header Byte 0 and Byte 1, respectively.
3. DLP Byte 2, Byte 3, Byte 4, and Byte 5, respectively.
4. CRC0 Byte 0, CRC0 Byte 1, CRC1 Byte 0, and CRC1 Byte 1, respectively.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 8-24. FDI (or RDI) Byte Mapping for 64B Datapath

Transfer (Rows)	FDI (or RDI) Bytes (Columns)							
	0	1	2	…	60	61	62	63
0	Flit Byte 0	Flit Byte 1	Flit Byte 2	…	Flit Byte 60	Flit Byte 61	Flit Byte 62	Flit Byte 63
1	Flit Byte 64	Flit Byte 65	Flit Byte 66	…	Flit Byte 124	Flit Byte 125	Flit Byte 126	Flit Byte 127
2	Flit Byte 128	Flit Byte 129	Flit Byte 130	…	Flit Byte 188	Flit Byte 189	Flit Byte 190	Flit Byte 191
3	Flit Byte 192	Flit Byte 193	Flit Byte 194	…	Flit Byte 252	Flit Byte 253	Flit Byte 254	Flit Byte 255

If the FDI or RDI datapath width is increased (or decreased), the Byte mapping follows the same convention of increasing order of Flit bytes mapped to increasing order of FDI (or RDI) bytes. Figure 8-25 shows an illustration of a 128B data path.

Figure 8-25. FDI (or RDI) Byte Mapping for 128B Datapath

Transfer (Rows)	FDI (or RDI) Bytes (Columns)											
	0	1	2	…	62	63	64	65	…	125	126	127
0	Flit Byte 0	Flit Byte 1	Flit Byte 2	…	Flit Byte 62	Flit Byte 63	Flit Byte 64	Flit Byte 65	…	Flit Byte 125	Flit Byte 126	Flit Byte 127
1	Flit Byte 128	Flit Byte 129	Flit Byte 130	…	Flit Byte 190	Flit Byte 191	Flit Byte 192	Flit Byte 193	…	Flit Byte 253	Flit Byte 254	Flit Byte 255

The frequency of operation of the interfaces along with the data width determines the maximum bandwidth that can be sustained across the FDI (or RDI) interface. For example, a 64B datapath at 2 GHz of clock frequency can sustain a 16 GT/s Link for an Advanced Package configuration. Similarly, to scale to 32 GT/s of Link speed operation for Advanced Package configuration, a 128B datapath running at 2 GHz would be able to support the maximum Link bandwidth.

The FDI (or RDI) byte mapping for the transmit or receive direction does not change for multi-module configurations. The MMPL logic within the Physical Layer is responsible for ensuring that the bytes are transmitted in the correct order to the correct module. Any byte swizzling or rearrangement to resolve module naming conventions, etc., is thus the responsibility of the MMPL logic.

8.3.2 Stallreq/Ack Mechanism

The Stallreq/Ack mechanism is used by the Lower Layer to interrupt the Flit transfers by the Upper Layer at a Flit boundary. On RDI, the Stallreq/Ack mechanism must be used when exiting Active state to Retrain, PM, LinkReset or Disabled states. On FDI, for UCIe Raw Format, the Stallreq/Ack mechanism must be used when exiting Active state to Retrain, PM, LinkReset or Disabled states. On FDI, for UCIe Flit Mode, the Stallreq/Ack mechanism must only be used when exiting Active State to a PM state. For other scenarios that exit Active state for UCIe Flit Mode, the Adapter must simply de-assert **`pl_trdy`** at a Flit boundary before state transition.

The Stallreq/Ack mechanism is mandatory for all FDI and RDI implementations. **`lp_stallack`** assertion implies that Upper Layer has stalled its pipeline at a Flit aligned boundary.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

The **lp_stallreq/pl_stallack** handshake is a four-phase sequence that follows the rules below:

1. The **pl_stallreq** and **lp_stallack** must be de-asserted before domain reset exit.
2. A rising edge on **pl_stallreq** must only occur when **lp_stallack** is de-asserted.
3. A falling edge on **pl_stallreq** must only occur when **lp_stallack** is asserted or when the domain is in reset.
4. A rising edge on **lp_stallack** must only occur when **pl_stallreq** is asserted.
5. A falling edge on **lp_stallack** must only occur when **pl_stallreq** is de-asserted or when domain is in reset.
6. When **lp_stallack** is asserted **lp_valid** and **lp_irdy** must both be de-asserted.
7. While **pl_stallreq** is asserted, any data presented on the interface must be accepted by the physical layer until the rising edge of **lp_stallack**. **pl_trdy** is not required to be asserted consecutively.
8. The logic path between **pl_stallreq** and **lp_stallack** must contain at least one flip-flop to prevent a combinatorial loop.
9. A complete stallreq/stallack handshake is defined as the completion of all four phases: Rising edge on **pl_stallreq**, rising edge on **lp_stallack**, falling edge on **pl_stallreq**, falling edge on **lp_stallack**.
10. It is strongly recommended that Upper Layer implements providing **lp_stallack** on a global free running clock so that it can finish the handshake even if the rest of its logic is clock gated.

To avoid performance penalties, it is recommended that this handshake be completed as quickly as possible while satisfying the above rules.

8.3.3 State Request and Status

Table 8-3 describes the Requests considered by the Lower layer in each of the interface states. The Upper layer must take into account the interface state status and make the necessary request modifications.

The requests are listed on the Row and the state status is listed in the Column.

The entries in Table 8-3 denote the following:

- Yes: Indicates that the request is considered for next state transition by the physical layer.
- N/A: Not Applicable
- Ignore: Indicates that the request is ignored and has no effect on the next state transition.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table 8-3. Requests Considered in Each State by Lower Layer

Request (Row) Versus Status (Column)	RESET	ACTIVE	L1	LinkReset	RETRAIN	DISABLE	L2	LinkError
NOP	Yes	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore
ACTIVE	Yes[1]	Ignore[2]	Yes	Yes	Yes	Yes	Yes	Yes
L1	Ignore	Yes	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore
LinkReset	Yes[1]	Yes	Yes	Ignore	Yes	Ignore	Yes	Ignore
RETRAIN	Ignore	Yes	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore
DISABLE	Yes[1]	Yes	Yes	Yes	Yes	Ignore	Yes	Ignore
L2	Ignore	Yes	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore
LinkError (sideband wire)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

1. Requires request transition from NOP
2. If the Status is Active.PMNAK, then the Lower Layer transitions to Active upon sampling the Active Request.

8.3.3.1 Reset State rules

The Reset State can be entered on de-assertion of interface reset signal or from LinkReset/Disable/ LinkError/L2 states. In Reset state, the physical layer is permitted to begin its initialization/training process.

The **pl_state_sts** is not permitted to exit Reset state until requested by the upper layer. The exit from Reset state is requested by the upper layer by changing the **lp_state_req** signal from NOP encoding value to the permitted next state encoding value.

The rules for Reset state transition are as follows:

1. Reset→Active: The lower layer triggers transitions to the Active state upon observing **lp_state_req** == Reset (NOP) for at least one clock while **pl_state_sts** is indicating Reset followed by observing **lp_state_req** == Active. The transition to Active is only completed once the corresponding Active Entry handshakes have completed on the Link. For RDI, it is when the Physical Layer has sent and received an Active Response sideband message to and from the remote Physical Layer respectively. For the Adapter LSM, it is when the Adapter has sent and received an Active Status sideband message to and from the remote Adapter respectively. For the ARB/MUX vLSM, it is when the ARB/MUX has sent and received an Active Status ALMP to and from the remote ARB/MUX respectively.
2. Reset→LinkReset: The lower layer transitions to the LinkReset state upon observing **lp_state_req** == Reset (NOP) for at least one clock while **pl_state_sts** is indicating Reset followed by observing **lp_state_req** == LinkReset OR when requested by remote Link partner through the relevant sideband message. The lower layer is permitted to transition through Active State, and when it does, Active state exit conditions apply.
3. Reset→Disabled: The lower layer transitions to the Disabled state upon observing **lp_state_req** == Reset (NOP) for at least one clock while **pl_state_sts** is indicating Reset followed by observing **lp_state_req** == Disabled OR when requested by the remote Link partner through the relevant sideband message. The lower layer is permitted to transition through Active State, and when it does, Active state exit conditions apply.
4. Reset→LinkError: The lower layer transitions to LinkError based on observing an internal request to move to the LinkError or **lp_linkerror** assertion. For RDI, this transition is permitted if requested by the remote Link partner through the relevant sideband message.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3.3.2 Active State rules

The Active state to next state transitions are described below.

The rules for Active State transition are as follows:

1. Active→Retrain: The Lower layer transitions to the Retrain state upon observing **lp_state_req** == Retrain or due to an internal request to retrain the Link while **pl_state_sts** == Active. This arc is not applicable for CXL vLSMs exposed on FDI (CXL Flit Mode with Retry in the Adapter).
2. Active→L1: The physical layer transitions to L1 based on observing **lp_state_req** == L1 while in the Active state, if other conditions of PM entry have also been satisfied.
3. Active→L2: The physical layer transitions to L2 based on observing **lp_state_req** == L2 while in the Active state, if other conditions of PM entry have also been satisfied.

It is permitted to have an Active.PMNAK to Retrain/LinkReset/Disable/LinkError transition for cases where Lower Layer is waiting for the Upper Layer to change the request to Active and the corresponding Link event triggers it. There is no scenario where there is a transition from Active.PMNAK to L1 or L2.

Section 8.3.3.8 describes the transition from Active or Active.PMNAK to LinkReset, Disable, or LinkError states.

8.3.3.3 PM Entry/Exit rules

See the PM entry and exit sequences in the RDI and FDI sections.

8.3.3.4 Retrain State rules

Adapter requests Retrain on RDI if any of the following events occur:

- Software writes to the Retrain bit and the Link is in Active state.
- Number of CRC or parity errors detected crosses a threshold. The specific algorithm for determining this is implementation specific.
- Protocol Layer requests Retrain (only applicable for Raw Format).
- any other implementation specific condition (if applicable).

Physical Layer triggers a Retrain transition on RDI if:

- Valid framing errors are observed
- Remote Physical Layer requests Retrain entry
- Adapter requests Retrain

Protocol Layer must not request Retrain on FDI, unless UCIe is operating in Raw Format.

A Retrain transition on RDI must always be propagated to Adapter LSMs that are in Active. Retrain transitions of the UCIe Link are not propagated to CXL vLSMs. Upon Retrain entry, the credit counter for UCIe Retimer (if present) must be reset to the value advertised during initial Link bring up (the value is given by the "Retimer_Credits" Parameter in the {AdvCap.Adapter} sideband message during initial Link bring up). The Retimer must drain or dump any Flits in flight or its internal transport buffers upon entry to Retrain. Additionally, the Retimer must trigger Retrain of the remote UCIe Link (across the Off-Package Interconnect).

Entry into Retrain state resets power management state for the corresponding state machine, and power management entry if required must be re-initiated after the interface enters Active state. If

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

there was an outstanding PM request that returns PM Response, the corresponding state machine must perform Active Entry handshakes to bring that state machine back to Active.

The rules for Retrain state transition are as follows:

1. Retrain→Active: If Retrain was entered from L1, the lower layer begins Active Entry handshakes upon observing **lp_state_req** == Active while the **pl_state_sts** == Retrain. If Retrain was entered from Active, the lower layer begins Active Entry handshakes only after observing a NOP-> Active transition on **lp_state_req**. Lower layer transitions to Active once the corresponding Active Entry handshakes have completed. Exit from Retrain on RDI requires the Active Entry handshakes to have completed between Physical Layers. Exit from Retrain on FDI must ensure that RDI has moved back to Active, and Active Entry handshakes have successfully completed between Adapters (for the Adapter LSM).
2. Transitional state: The lower layer is permitted to transition to the Active state upon observing **lp_state_req** == LinkReset or Disabled while the **pl_state_sts** == Retrain. Following the entry into Active the lower layer is permitted to make a transition to the requested state.

Section 8.3.3.8 describes Retrain exit to LinkReset, Disable, or LinkError states.

Note: The requirement to wait for NOP->Active transition ensures that the Upper Layer has a way to delay Active transition in case it is waiting for any relevant sideband handshakes to complete (for example the Parity Feature handshake).

8.3.3.5 LinkReset State rules

LinkReset is used for reset flows (HotReset equivalent in PCIe, Protocol Layer must use this to propagate SBR to the device as well) to convey device and/or Link Reset across the UCIe Link.

Adapter triggers LinkReset transition upon observing a LinkReset request from the Protocol Layer, on receiving a sideband message requesting LinkReset entry from the remote Link partner or an implementation specific internal condition (if applicable). Implementations must make best efforts to gracefully drain the Retry buffers when transitioning to LinkReset, however, entry to LinkReset must not timeout on waiting for the Retry buffer to drain. The Protocol Layer and Adapter must drain/flush their pipelines and retry buffer of the Flits for the corresponding Protocol Stack once the FDI state machines have entered LinkReset.

If all the FDI state machines and Adapter LSMs are in LinkReset, the Adapter triggers RDI to enter LinkReset as well.

The rules for LinkReset State transitions are as follows:

1. LinkReset→Reset: The lower layer transitions to the Reset state due to an internal request to move to Reset (example reset pin trigger) or **lp_state_req** == Active while **pl_state_sts** == LinkReset and all necessary actions with respect to LinkReset have been completed.
2. LinkReset→Disabled The lower layer transitions to Disabled based on observing **lp_state_req** == Disabled or due to an internal request to move to Disabled while **pl_state_sts** == LinkReset.
3. Transitional State: The PHY is permitted to transition through Reset State, and when it does, Reset state exit conditions apply.
4. LinkReset→LinkError: The lower layer transitions to LinkError due to an internal request to move to LinkError or **lp_linkerror** assertion while **pl_state_sts** == LinkReset.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3.3.6 Disabled State rules

Adapter triggers Disabled entry when any of the following events occur:

- Protocol Layer requests entry to Disabled state
- Software writes to the Link Disable bit corresponding to UCIe
- Remote Link partner requests entry to Disabled state through the relevant sideband message
- An implementation specific internal condition (if applicable)

Implementations must make best efforts to gracefully drain the Retry buffers when transitioning to Disabled, however, entry to Disabled must not timeout on waiting for the Retry buffer to drain. The Protocol Layer and Adapter must drain/flush their pipelines and retry buffer of the Flits for the corresponding Protocol Stack once the FDI state machines have entered Disabled.

If all the FDI state machines and Adapter LSMs are in Disabled, the Adapter triggers RDI to enter Disabled as well.

The rules for Disabled State are as follows:

- Disabled→Reset: The physical layer transitions to the Reset state due to an internal request to move to Reset (example reset pin trigger) or **lp_state_req** == Active while **pl_state_sts** == Disabled and all necessary actions with respect to Disabled transition have completed.
- Disabled→LinkError: The physical layer transitions to LinkError due to an internal request to move to LinkError or **lp_linkerror** assertion while **pl_state_sts** == Disabled.

8.3.3.7 LinkError State rules

The lower layer enters LinkError state when directed via **lp_linkerror** signal or due to Internal LinkError conditions. For RDI, the entry is also triggered if the remote Link partner requested LinkError entry through the relevant sideband message. It is not required to complete the stallreq/ack handshake before entering this state. However, for implementations where LinkError state is not a terminal state (terminal implies SoC needs to go through reset flow after reaching LinkError state), it is expected that software can come and retrain the Link after clearing error status registers, etc., and the following rules should be followed:

- If the lower layer decides to perform a **pl_stallreq/lp_stallack** handshake, it must provide **pl_trdy** to the upper layer to drain the packets. In cases where there is an uncorrectable internal error in the lower layer, these packets could be dropped and not transmitted on the Link.
- It is required for the upper layer to internally clean up the data path, even if **pl_trdy** is not asserted and it has sampled LinkError on **pl_state_sts** for at least one clock cycle.

The lower layer may enter LinkError state due to Internal LinkError requests such as when:

- Encountering uncorrectable errors due to hardware failure or directed by Upper Layer
- Remote Link partner requests entry into LinkError (RDI only)

The rules for LinkError state are as follows:

- LinkError→Reset: The lower layer transitions to Reset due to an internal request to move to Reset (example: reset pin assertion, or software clearing the error status bits that triggered the error) or **lp_state_req** == Active and **lp_linkerror** = 0, while **pl_state_sts** == LinkError and minimum residency requirements are met. Lower Layer must implement a minimum residency time in LinkError of 16 ms to ensure that the remote Link partner will be forced to enter LinkError due to timeouts (to cover for cases where the LinkError transition happened and sideband was not functional).

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3.3.8 Common State rules

This section covers some of the common conditions for exit from Active, Retrain, L1, and L2 to LinkReset, Disable and LinkError states. For RDI, PM encoding and rules correspond to L1 in the text below.

The rules are as follows:

- [Active, Retrain, L1, L2]→LinkReset: The lower layer transitions to LinkReset based on observing **lp_state_req** ==LinkReset or due to an internal request to move to LinkReset or the remote Link partner requesting LinkReset over sideband.
- [Active, Retrain, L1, L2]→Disabled: The physical layer transitions to Disabled based on observing **lp_state_req** ==Disabled or due to an internal request to move to Disabled while **pl_state_sts** == Active, or the remote Link partner requesting Disabled over sideband.
- [Active, Retrain, L1, L2]→LinkError: The physical layer transitions RDI to LinkError based on observing an internal request to move to the LinkError or **lp_linkerror** assertion, or the remote Link partner requesting LinkError over sideband. RDI must move to LinkError before propagating LinkError to all Adapter LSMs.

From a state machine hierarchy perspective, it is required for Adapter LSM to move to LinkReset, Disabled or LinkError before propagating this to CXL vLSMs. This ensures CXL rules are followed where these states are “non-virtual” from the perspective of CXL vLSMs.

Adapter LSM can transition to LinkReset or Disabled without RDI transitioning to these states. In the case of multi-protocol stacks over the same Physical Link/Adapter, each Protocol can independently enter these states without affecting the other protocol stack on the RDI.

If all the Adapter LSMs have moved to a common state of LinkReset/Disabled or LinkError, then RDI is taken to the corresponding state. If however, the Adapter LSMs are in different state combinations of LinkError, Disabled or LinkReset, the RDI is moved to the highest priority state. The priority order from highest to lowest is LinkError, Disabled, LinkReset. For a LinkError/LinkReset/Disabled transition on RDI, Physical Layer must initiate the corresponding sideband handshake to transition remote Link partner to the required state. If no response is received from remote Link partner for this message after 8ms, RDI transitions to LinkError.

If RDI moves to a state that is of a higher priority order than the current Adapter LSM, it is required for the Adapter to propagate that to the Adapter LSM using sideband handshakes to ensure the transition with the remote Link partner.

After transition from LinkError/LinkReset/Disable to Reset on RDI, the Physical Layer must not begin training unless it observes a NOP->Active transition on **lp_state_req** from the Adapter or Software writes 1b to the Start UCIe Link Training bit. The Adapter should not trigger NOP->Active unless it receives this transition from the Protocol Layer or has internally decided to bring the Link Up. The Adapter must trigger this on RDI if the Protocol Layer has triggered this even if **pl_inband_pres** = 0. Thus, if the Protocol Layer is waiting for software intervention and wants to hold back the Link from training, it can delay the NOP->Active trigger on FDI. Upper Layers are permitted to transition **lp_state_req** back to NOP after giving the NOP->Active trigger in order to clock gate while waiting for **pl_inband_pres** to assert.

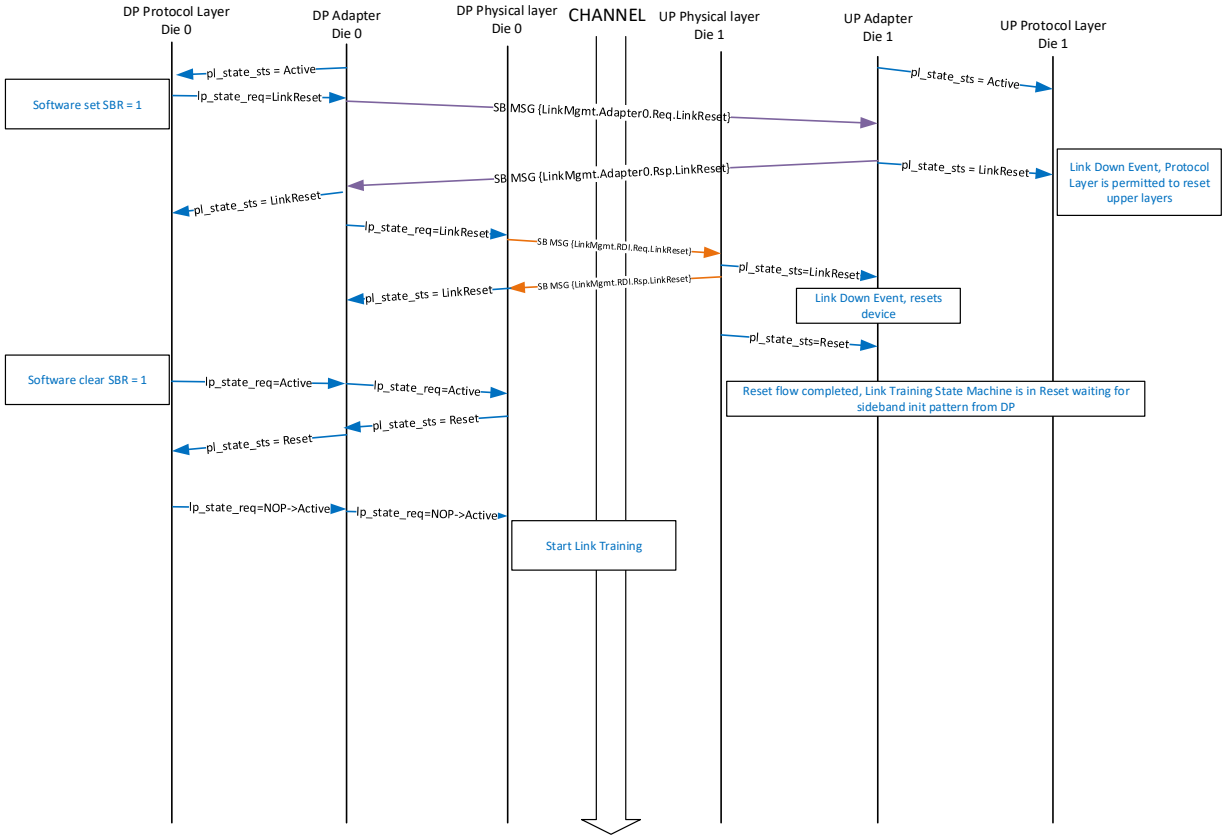
Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3.4 Example flow diagrams

8.3.4.1 LinkReset Entry and Exit

Figure 8-26 shows an example flow for LinkReset entry and exit. In the multi-protocol stack scenario, it is permitted for each protocol stack to independently transition to LinkReset. RDI is only transitioned to LinkReset if all the corresponding Adapter LSMs are in LinkReset.

Figure 8-26. LinkReset Example

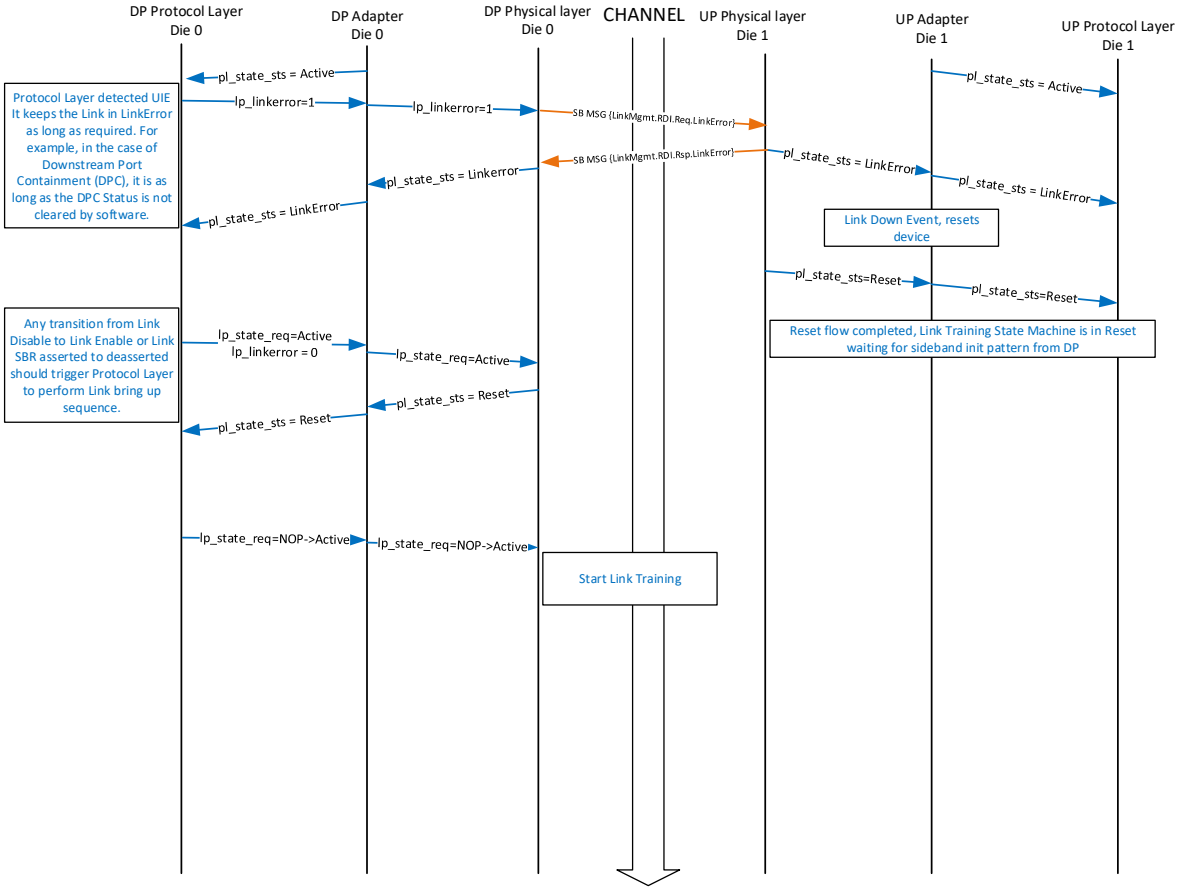


Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

8.3.4.2 LinkError

Figure 8-27 shows an example of LinkError entry and exit when the Protocol Layer detected an uncorrectable internal error.

Figure 8-27. LinkError example



§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

9.0 Compliance

The goal of Compliance testing is to validate the mainband supported features of a Device Under Test (DUT) against a known good reference UCIe implementation. Device support for Compliance Testing is optional, however a device that does not support capabilities listed in this chapter may not be able to participate in the Compliance program. Different layers of UCIe (Physical, Adapter, Protocol) will be checked independently with a suite of tests for compliance testing.

The system setup for compliance testing is composed of the following:

- Reference UCIe design (Golden Die): This is a known good UCIe implementation across all layers of the UCIe stack.
- DUT: One or more DUTs that will be tested with the reference design. It is required that these have cleared the testing requirements of die sort/pre-bond before they are brought for compliance testing.
- In the case of Advanced Package configuration, a known good silicon bridge or interposer that connects the Golden Die with the DUT. In the case of Standard Package configuration, a known good package for connecting the Golden Die to the DUT.

UCIe implementations that support compliance testing must implement the Compliance/Test Register Block as outlined in Chapter 7.0 and adhere to the requirements outlined in this chapter.

The above components are integrated together in a test package (see Figure 9-1), which is then used for running Compliance and Interoperability tests.

UCIe sideband plays a critical role for enabling compliance testing by allowing compliance software to access registers from different UCIe components (e.g., Physical Layer, D2D Adapter, etc.) for setting up tests as well as monitoring status. It is expected that UCIe sideband comes up without requiring any FW initialization.

This specification defines the required hardware capabilities of the UCIe stack in the DUT. A separate document will be published later to describe the following:

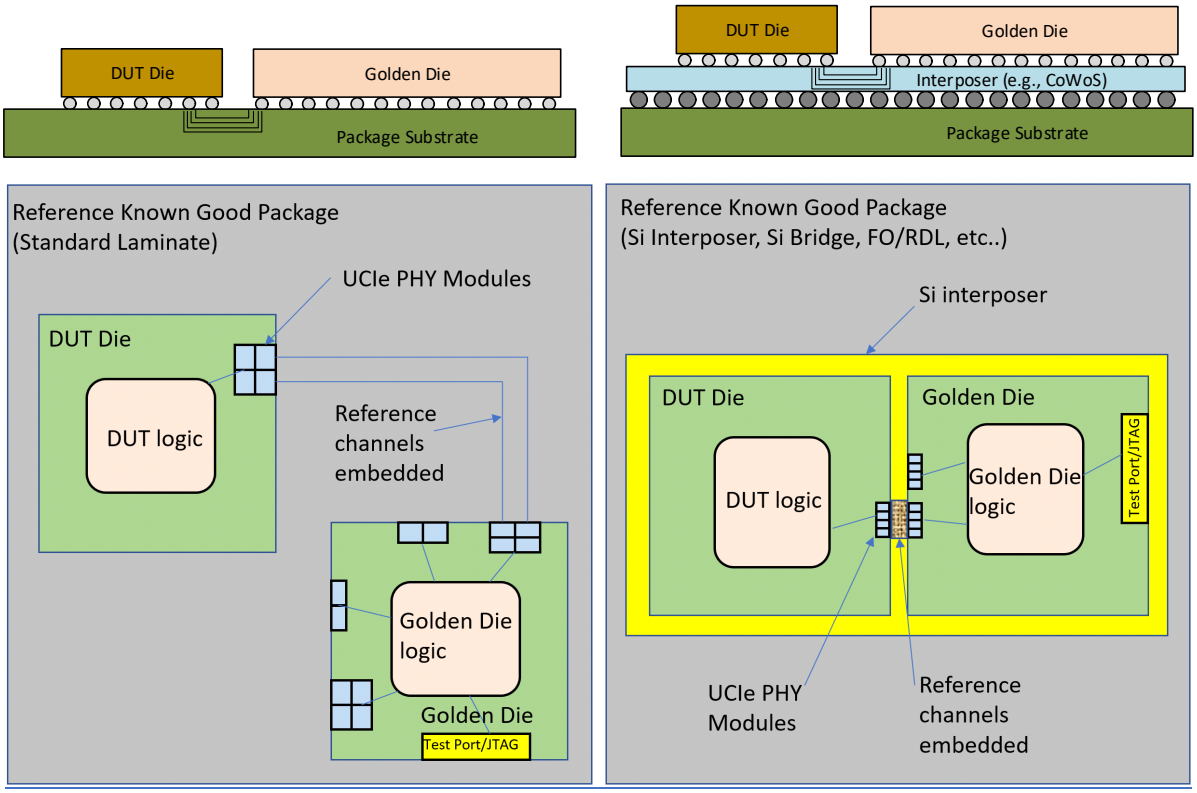
- Compliance test setup, including the channel model and package level details
- Test details
- Golden Die details including form factor and system-level behavior.

This chapter uses the terms ‘software’ and ‘compliance software’ interchangeably. Any use of the term ‘software’ in this chapter means compliance software that is either running on the Golden Die, or on an external controller that is connected to the Golden Die via test/JTAG port.

Software, prior to testing compliance for any optional UCIe capability, must read the corresponding Capability register (e.g., PHY Capability register described in Section 7.5.3.22) to ensure that the DUT implements the capability.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Figure 9-1. Examples of Standard and Advanced Package setups for DUT and Golden Die Compliance Testing



9.1 Protocol Layer Compliance

Protocol Layer Compliance testing seeks to test the UCIe protocol stack for compliance to the associated protocol layer specification.

For PCIe and CXL Protocol Layers, UCIe leverages the protocol compliance defined in those specifications for the respective transaction layers. Implementations must follow the requirements and capabilities outlined in *PCIe Base Specification* and *CXL Specification*, respectively.

For Streaming protocols, because Protocol Layer interoperability is specific to the protocol being streamed, compliance testing of the Protocol Layer is beyond the scope of this specification.

9.2 Adapter Compliance

This Specification defines the hardware capabilities that are required in the DUT for exercising and testing the different functionalities of the Adapter. The Golden Die Adapter must have all the capabilities of the DUT, support all the Flit Formats from Chapter 3.0, and must have the capability to inject both consistent and inconsistent sideband messages (for Parameter exchanges, Register accesses and state transitions) to test DUT behavior for various error scenarios (e.g., timeouts, etc.).

The capabilities listed in this section must be supported by the Adapter in the DUT if the Adapter supports any of the Flit Formats defined in Chapter 3.0. These capabilities are applicable to Adapters of all UCIe device types (including Retimers). Each of the capabilities also have their respective

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Control and Status registers, which are used to enable software to test various combinations of flows and test criteria.

- Ability to Inject Test or NOP Flits: On the Transmitter, the injection behavior is defined by the Flit Tx Injection Control register (see Table 7-73). For all injected Flits, CRC is computed, and if CRC error injection is enabled, CRC errors are injected accordingly. It is allowed for the Adapter to be set up to inject NOP Flits or Test Flits. NOP Flit follows the identical layout as defined in Chapter 3.0. Test Flits carry a special encoding of 01b in bits [7:6] of Byte 1 of the Flit Header that is applicable for all Flit Formats that the Adapter supports. Unlike NOP Flits, Test Flits go through the Tx Retry buffer if Retry is enabled. One of the purposes of defining the Test Flits is to test the Retry Flows independently, regardless of whether the Protocol Layer is enabled. The Payload in these Flits carry specific patterns that are determined by the fields in the Flit Tx Injection Control register. Software is permitted to enable flit injection in mission mode as well while interleaving with regular Protocol Flits using the appropriate programming (see the register fields in Table 7-73). At the Receiver, these Flits are not forwarded to the Protocol Layer. The Receiver cancels these using the **pl_flit_cancel** signal on FDI or any other mechanism; however, CRC must be checked the same as with regular Flits, and any errors must trigger the Retry Flows as applicable.
- Injection of Link State Request or Response sideband messages. This is controlled using the Link State Injection registers defined in the Link State Injection Control Stack 0 and Link State Injection Control Stack 1 registers (see Table 7-75 and Table 7-76, respectively). Single Protocol stack implementations use the Stack 0 register. Software must place the Adapter in Compliance mode (by writing 10b to the 'Compliance Mode' field in the Adapter Compliance Control register).
- Retry injection control as defined in the Retry Injection Control register (see Table 7-77).

9.3 PHY Compliance

This specification defines the hardware capabilities that are required in the Device Under Test (DUT) for exercising and testing the different functionalities of the Physical Layer. The Golden Die must support capabilities to force timeouts on all applicable sideband messages as well as state residence timers.

The registers and associated functionality defined in Section 7.5.4 and the UHM DVSEC Capability defined in Section 7.5.3.36 are used for Compliance testing. These registers provide the following functionality:

- Timing margining
- Voltage margining, when supported
- BER measurement
- Lane-to-Lane skew for a given module at both the Receiver and Transmitter
- TX Equalization (EQ) as defined in Section 5.3.3

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Appendix A CXL/PCIe Register applicability to UCIe

A.1 CXL Registers applicability to UCIe

All CXL-defined DVSECs fully apply in the context of UCIe when operating in Raw Format. When operating in non-Raw Format, a few register definitions need to be reinterpreted in the context of UCIe. See below for details. Note that regardless of the Raw Format or non-Raw Format, device/port configurations with CXL 1.1 compliance is not permitted as was discussed in Chapter 7.0.

Table A-1. CXL Registers for UCIe devices

Register Block	Register	Bits	Comments
DVSEC Capability	DVSEC Flex Bus Port Control	3,4	See next row for how these bits are handled
	DVSEC Flex Bus Port Status	3, 4	This bit just mirrors bits 3, 4 in Flex bus port control register, to mimic legacy behavior.
		11, 12	Hardwired to 0
	From 14h-1Fh		N/A

A.2 PCIe Register applicability to UCIe

All PCIe specifications defined DVSEC apply in the context of UCIe as well. There are a few Link and PHY layer registers/bits though that are N/A or need to be reinterpreted in the context of UCIe. They are listed below.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Table A-2. PCIe Registers for UCIe devices

Register Block	Register	Bits	Comments
PCIe capability	PCI Express Capabilities Register	8	Slot implemented – set to 0. And hence follow rules for implementing other slot related registers/bits at various locations in the PCIe capability register set.
	Device capabilities Register	8:6	N/A and can be set to any value
	Link Capabilities Register	3:0	Max Link Speed: Set to 0011b indicating 8GT/s
		9:4	Max Link Width: 01 0000b, indicating x16
		11:10	ASPM support: 01b/11b encodings disallowed
		14:12	N/A
		17:15	L1 Exit Latency: Devices/Ports must set this bit based on whether they are connected to a retimer or not, and also the retimer based exit latency might not be known at design time as well. To assist with this, these bits need to be made HWInit from a device/port perspective so system FW can set this at boot time based on the specific retimer based latencies.
		18	N/A and hardwired to 0
	Link Control Register	6	HW ignores what is written here but follow any base spec rules for bit attributes.
		7	HW ignores what is written here but follow any base spec rules for bit attributes.
		8	Set to RO 0
		9	Set to RO 0
		10, 11, 12	HW ignores what is written in these bits but follows any base spec rules for bit attributes.
	Link Status Register	3:0	Current Link speed: Set to 0011b indicating 8GT/s
		9:4	Negotiated Link width: x16
		15	Hardwired to 0
	Link Capabilities 2 Register	7:1	Set to 000 0111b
		15:9	Set to 00h
PCIe Capability	Link Capabilities 2 Register	22:16	Set to 00h
		24:23	Set to 00b just to appear compliant
	Link Control 2 Register	3:0	Target Link speed: Writes to this register are ignored by UCIe hardware, but HW follows the base spec rules for bit attributes
		4	HW ignores what is written in this bit but follows any base spec rules for bit attributes.
		5	HW autonomous speed disable – Set to RO 0
		15:6	N/A for UCIe. HW should follow base spec rules for register bit attributes.
	Link Status 2 Register	9:0	Set to RO 0
PCIe Extended Capability	Secondary PCI Express Extended Capability	All	Implement per the base spec, but HW ignores all commands from SW and also sets all equalization control registry entries to 0.

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

A.3 PCIe/CXL registers that need to be part of D2D

- PCIe Link Control Register
  - Bits 13, 5, 4, and 1:0 are relevant for D2D operation
- CXL DVSEC Flex Bus Port Received Modified TS Data Phase1 Register
- CXL DVSEC Flex Bus Port Control
- CXL DVSEC Flex Bus Port Status
- CXL ARB/MUX registers

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

Appendix B AIB Interoperability

Implementations are permitted to design a superset stack to be interoperable with UCIe/AIB PHY. This section details the UCIe interoperability criteria with AIB.

B.1 AIB Signal Mapping

B.1.1 Data path

Data path signal mapping for AIB 2.0 and AIB 1.0 are shown in Table B-1 and Table B-2 respectively. AIB sideband is sent over an asynchronous path on UCIe main band.

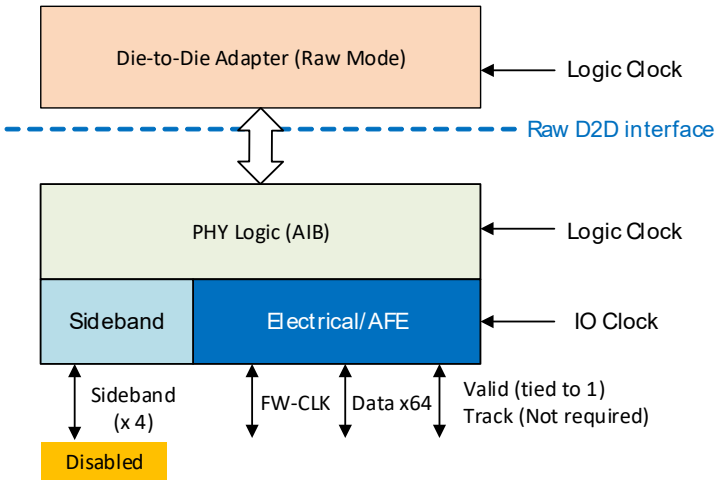
B.1.2 Always high Valid

Always high Valid is an optional feature that is only applicable to AIB interoperability applications. This must be negotiated prior to main band Link training through parameter exchange. Raw mode must be used in such applications.

B.1.3 Sideband

AIB sideband is sent using UCIe main band signals. UCIe sideband is not required in AIB interoperability mode and it is disabled (Transmitters are Hi-Z and Receivers are disabled).

Figure B-1. AIB interoperability



Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

B.1.4 Raw Die-to-Die interface

AIB Phy logic block shown in Figure B-1 presents a subset of RDI to next layer up.

Note: More details will be shown in a later revision of this specification

Table B-1. AIB 2.0 Datapath mapping for Advanced Package

UCIe Interface	AIB 2.0	Note
TXDATA[39:0]	**TX[39:0]**	
TXDATA[47:40]	AIB Sideband Tx	Asynchronous path
TXDATA[63:48]	NA	Disabled (Hi-Z)
RXDATA[39:0]	**RX[39:0]**	
RXDATA[47:40]	AIB Sideband Rx	Asynchronous path
RXDATA[63:48]	NA	
TXDATASB	NA	Disabled (Hi-Z)
RXDATASB		
TXCKSB		
RXCKSB		
TXDATASBRD		
RXDATASBRD		

Table B-2. AIB 1.0 Datapath mapping for Advanced Package

UCIe Interface	AIB 1.0	Note
TXDATA[19:0]	**TX[19:0]**	
TXDATA[42:20]	AIB Sideband Tx	Asynchronous path
TXDATA[63:43]	NA	Disabled (Hi-Z)
RXDATA[19:0]	**RX[19:0]**	
RXDATA[42:20]	AIB Sideband Rx	Asynchronous path
RXDATA[63:43]	NA	
TXDATASB	NA	Disabled (Hi-Z)
RXDATASB		
TXCKSB		
RXCKSB		
TXDATASBRD		
RXDATASBRD		

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023

B.2 Initialization

AIB Phy logic block shown in Figure B-1 contains all the AIB Link logic and state machines. Please see AIB specification (Section 2 and Section 3) for initialization flow.

B.3 Bump Map

Note: More details will be shown in a future revision this specification

§ §

Property of Universal Chiplet Interconnect Express (UCIe) © 2022–2023