PCIE 5.0 规范

柏承双

邮箱：bcs_01@126.com

公众号：芯片业者柏承双

扫一扫上面的二维码图案，加我为朋友。

芯片业者柏承双

微信扫描二维码，关注我的公众号

还是得把一些背景写在这里，以免大家在理解的时候产生误会。

大概一个年头的零零碎碎的时间终于初步把 PCIE5.0 协议翻译了一遍。本身整理的时候是在做 PCIE3.0 的项目，然后做着做着不经意间冒出的想法，随后转而去做 PCIE5.0 协议的翻译。其中有些信息还是原来 PCIE3.0 的内容，有机会的时候整理个完善的版本再发。还是建议桌面上再放一个英文版，特别是做 PCIE IP 的同学。

但是不管哪个版本，PCIE 规范背后的驱动因素是我们要思考的东西，这个东西不仅能帮我们理解协议的内容，也能帮我们理解一个系统架构定义背后需要思考的东西。我们在之前的文章里也不止一次提到过 PCIE 是个系统的问题，所以对系统不了解，对架构没概念，理解中肯定会有欠缺的。所以要想精通 PCIE，没几个项目的历练，确实很难办到。

PCIE 协议只讲规则，不讲原因。而且边界很清晰，很少讲系统之外的东西，当然定义协议本身就该如此。因为 PCIE 协议边界之外的东西又变化多端，又不一定形成规则，所以很难讲。

在我翻译的协议里边尝试补充说明了一些信息，但现在看来显然不够。PCIE 协议内容太庞大了，所涉及都周边知识比协议本身多得多，所以我试图在公众号中去做一些整理，一是写一些自己的理解，聊一聊 PCIE 协议背后的逻辑；二是希望能跟业内同行一起交流分享，建立更完整的系统知识。

各位可以扫码关注首页的公众号，大家一起交流。

加我微信也是可以的。

公众号：芯片业者相关权

PCI Express® Base Specification Revision 5.0 Version 1.0

Table of Contents

PCI Express® Base Specification Revision 5.0 Version 1.0 1

Table of Contents 1

Status of This Document 40

Terms and Acronyms 41

Reference Documents 55

1 Introduction 58

1.1 A Third Generation I/O Interconnect 58

1.2 PCI Express Link 60

1.3 PCI Express Fabric Topology 60

1.3.1 RC 61

1.3.2 Endpoint 62

1.3.2.1 Legacy Endpoint Rules 62

1.3.2.2 PCI Express Endpoint Rules 62

1.3.2.3 RC Integrated Endpoint Rules 63

1.3.3 Switch 63

1.3.4 Root Complex Event Collector 64

1.3.5 PCI Express to PCI/PCI-X Bridge 65

1.4 Hardware/Software Model for Discovery, Configuration and Operation 65

1.5 PCI Express Layering Overview 65

1.5.1 Transaction Layer 66

1.5.2 Data Link Layer 67

1.5.3 Physical Layer 67

1.5.4 Layer Functions and Services 67

1.5.4.1 Transaction Layer Services 67

1.5.4.2 Data Link Layer Services 68

1.5.4.3 Physical Layer Services 68

1.5.4.4 Inter-Layer Interfaces 69

公众号·芯片业者相承及

1.5.4.4.1 Transaction/Data Link Interface 69
1.5.4.4.2 Data Link/Physical Interface 69
2 Transaction Layer Specification 71
2.1 Transaction Layer Overview 71
2.1.1 Address Spaces, Transaction Types, and Usage 72
2.1.1.1 Memory Transactions 72
2.1.1.2 I/O Transactions 72
2.1.1.3 Configuration Transactions 72
2.1.1.4 Message Transactions 72
2.1.2 Packet Format Overview 73
2.2 Transaction Layer Protocol - Packet Definition 74
2.2.1 Common Packet Header Fields 74
2.2.2 TLPs with Data Payloads - Rules 77
2.2.3 TLP Digest Rules 79
2.2.4 Routing and Addressing Rules 80
2.2.4.1 Address Based Routing Rules 80
2.2.4.2 ID Based Routing Rules 82
2.2.5 First/Last DW Byte Enables Rules 83
2.2.6 Transaction Descriptor 86
2.2.6.1 Overview 86
2.2.6.2 Transaction Descriptor – Transaction ID Field 86
2.2.6.3 Transaction Descriptor – Attributes Field 90
2.2.6.4 Relaxed Ordering and ID-Based Ordering Attributes 91
2.2.6.5 No Snoop Attribute 92
2.2.6.6 Transaction Descriptor – Traffic Class Field 92
2.2.7 Memory, I/O, and Configuration Request Rules 92
2.2.7.1 TPH Rules 95
2.2.8 Message Request Rules 97
2.2.8.1 INTx Interrupt Signaling - Rules 99
2.2.8.2 Power Management Messages 102
2.2.8.3 Error Signaling Messages 103

公众号·芯片业者相承双

2.2.8.4 Locked Transactions Support 104
2.2.8.5 Slot Power Limit Support 105
2.2.8.6 Vendor_Defined Messages 105
2.2.8.6.1 PCI-SIG-Defined VDMs 107
2.2.8.6.2 LN Messages 107
2.2.8.6.3 Device Readiness Status (DRS) Message 109
2.2.8.6.4 Function Readiness Status Message (FRS Message) 109
2.2.8.6.5 Hierarchy ID Message 111
2.2.8.7 Ignored Messages 112
2.2.8.8 Latency Tolerance Reporting (LTR) Message 112
2.2.8.9 Optimized Buffer Flush/Fill (OBFF) Message 113
2.2.8.10 Precision Time Measurement (PTM) Messages 114
2.2.9 Completion Rules 115
2.2.10 TLP Prefix Rules 117
2.2.10.1 Local TLP Prefix Processing 118
2.2.10.1.1 Vendor Defined Local TLP Prefix 118
2.2.10.2 End-End TLP Prefix Processing 119
2.2.10.2.1 Vendor Defined End-End TLP Prefix 120
2.2.10.2.2 Root Ports with End-End TLP Prefix Supported 120
2.3 Handling of Received TLPs 120
2.3.1 Request Handling Rules 123
2.3.1.1 Data Return for Read Requests 128
2.3.2 Completion Handling Rules 132
2.4 Transaction Ordering 134
2.4.1 Transaction Ordering Rules 134
2.4.2 Update Ordering and Granularity Observed by a Read Transaction 137
2.4.3 Update Ordering and Granularity Provided by a Write Transaction 138
2.5 Virtual Channel (VC) Mechanism 138
2.5.1 Virtual Channel Identification (VC ID) 140
2.5.2 TC to VC Mapping 140
2.5.3 VC and TC Rules 141

公众号·芯片业者相承双

2.6 Ordering and Receive Buffer Flow Control 142
2.6.1 Flow Control Rules 143
2.6.1.1 FC Information Tracked by Transmitter 147
2.6.1.2 FC Information Tracked by Receiver 148
2.7 Data Integrity 151
2.7.1 ECRC Rules 151
2.7.2 Error Forwarding 154
2.7.2.1 Error Forwarding Usage Model 154
2.7.2.2 Rules For Use of Data Poisoning 155
2.8 Completion Timeout Mechanism 156
2.9 Link Status Dependencies 157
2.9.1 Transaction Layer Behavior in DL_Down Status 157
2.9.2 Transaction Layer Behavior in DL_Up Status 158
2.9.3 Transaction Layer Behavior During Downstream Port Containment 158
3 Data Link Layer Specification 160
3.1 Data Link Layer Overview 160
3.2 Data Link Control and Management State Machine 161
3.2.1 Data Link Control and Management State Machine Rules 162
3.3 Data Link Feature Exchange 164
3.4 Flow Control Initialization Protocol 166
3.4.1 Flow Control Initialization State Machine Rules 166
3.4.2 Scaled Flow Control 168
3.5 Data Link Layer Packets (DLLPs) 169
3.5.1 Data Link Layer Packet Rules 170
3.6 Data Integrity Mechansisms 175
3.6.1 Introduction 175
3.6.2 LCRC, Sequence Number, and Retry Management (TLP Transmitter) 176
3.6.2.1 LCRC and Sequence Number Rules (TLP Transmitter) 176
3.6.2.2 Handling of Received DLLPs 182
3.6.3 LCRC and Sequence Number (TLP Receiver) 184
3.6.3.1 LCRC and Sequence Number Rules (TLP Receiver) 184

公众号·芯片业者相承双

4 Physical Layer Logical Block 190
4.1 Introduction 190
4.2 Logical Sub-block 190
4.2.1 Encoding for 2.5 GT/s and 5.0 GT/s Data Rates 190
4.2.1.1 Symbol Encoding 190
4.2.1.1.1 Serialization and De-serialization of Data 191
4.2.1.1.2 Special Symbols for Framing and Link Management(K code) 192
4.2.1.1.3 8b/10b Decode Rules 193
4.2.1.2 Framing and Application of Symbols to Lanes 193
4.2.1.3 Data Scrambling 195
4.2.2 Encoding for 8.0 GT/s and Higher Data Rates 196
4.2.2.1 Lane Level Encoding 197
4.2.2.2 Ordered Set Blocks 198
4.2.2.2.1 Block Alignment 198
4.2.2.3 Data Blocks 199
4.2.2.3.1 Framing Tokens 199
4.2.2.3.2 Transmitter Framing Requirements 203
4.2.2.3.3 Receiver Framing Requirements 204
4.2.2.3.4 Recovery from Framing Errors 206
4.2.2.4 Scrambling 206
4.2.2.5 Precoding 208
4.2.2.6 Loopback with 128b/130b Code 210
4.2.3 Link Equalization Procedure for 8.0 GT/s and Higher Data Rates 210
4.2.3.1 Rules for Transmitter Coefficients 219
4.2.3.2 Encoding of Presets 219
4.2.4 Link Initialization and Training 220
4.2.4.1 Training Sequences 221
4.2.4.2 Alternate Protocol Negotiation 230
4.2.4.3 Electrical Idle Sequences 232
4.2.4.4 Inferring Electrical Idle 235
4.2.4.5 Lane Polarity Inversion 236

公众号·芯片业者相承双

4.2.4.6 Fast Training Sequence (FTS) 237
4.2.4.7 Start of Data Stream Ordered Set 238
4.2.4.8 Link Error Recovery 239
4.2.4.9 Reset 239
4.2.4.9.1 Fundamental Reset 239
4.2.4.9.2 Hot Reset 240
4.2.4.10 Link Data Rate Negotiation 240
4.2.4.11 Link Width and Lane Sequence Negotiation 240
4.2.4.11.1 Required and Optional Port Behavior 240
4.2.4.12 Lane-to-Lane De-skew 241
4.2.4.13 Lane vs. Link Training 241
4.2.5 Link Training and Status State Machine (LTSSM) Descriptions 242
4.2.5.1 Detect Overview 242
4.2.5.2 Polling Overview 242
4.2.5.3 Configuration Overview 242
4.2.5.4 Recovery Overview 242
4.2.5.5 L0 Overview 242
4.2.5.6 L0s Overview 243
4.2.5.7 L1 Overview 243
4.2.5.8 L2 Overview 243
4.2.5.9 Disabled Overview 243
4.2.5.10 Loopback Overview 243
4.2.5.11 Hot Reset Overview 244
4.2.6 Link Training and Status State Rules 244
4.2.6.1 Detect 246
4.2.6.1.1 Detect.Quiet 246
4.2.6.1.2 Detect.Active 248
4.2.6.2 Polling 248
4.2.6.2.1 Polling.Active 249
4.2.6.2.2 Polling.Compliance 251
4.2.6.2.3 Polling.Configuration 254

公众号·芯片业者相承双

4.2.6.2.4 Polling.Speed 255
4.2.6.3 Configuration 255
4.2.6.3.1 Configuration.Linkwidth.Start 255
4.2.6.3.1.1 Downstream Lanes 256
4.2.6.3.1.2 Upstream Lanes 257
4.2.6.3.2 Configuration.Linkwidth.Accept 259
4.2.6.3.2.1 Downstream Lanes 259
4.2.6.3.2.2 Upstream Lanes 260
4.2.6.3.3 Configuration.Lanenum.Accept 261
4.2.6.3.3.1 Downstream Lanes 261
4.2.6.3.3.2 Upstream Lanes 262
4.2.6.3.4 Configuration.Lanenum.Wait 262
4.2.6.3.4.1 Downstream Lanes 263
4.2.6.3.4.2 Upstream Lanes 263
4.2.6.3.5 Configuration.Complete 263
4.2.6.3.5.1 Downstream Lanes 264
4.2.6.3.5.2 Upstream Lanes 265
4.2.6.3.6 Configuration.Idle 266
4.2.6.4 Recovery 268
4.2.6.4.1 Recovery.RcvrLock 268
4.2.6.4.2 Recovery.Equalization 273
4.2.6.4.2.1 Downstream Lanes 273
4.2.6.4.2.2 Upstream Lanes 275
4.2.6.4.3 Recovery.Speed 278
4.2.6.4.4 Recovery.RcvrCfg 279
4.2.6.4.5 Recovery.Idle 281
4.2.6.5 L0 283
4.2.6.6 L0s 284
4.2.6.6.1 Receiver L0s 285
4.2.6.6.1.1 Rx_L0s.Entry 285
4.2.6.6.1.2 Rx_L0s.Idle 285

公众号·芯片业者相承双

4.2.6.6.1.3 Rx_L0s.FTS 285
4.2.6.6.2 Transmitter L0s 286
4.2.6.6.2.1 Tx_L0s.Entry 286
4.2.6.6.2.2 Tx_L0s.Idle 286
4.2.6.6.2.3 Tx_L0s.FTS 286
4.2.6.7 L1 287
4.2.6.7.1 L1.Entry 287
4.2.6.7.2 L1.Idle 287
4.2.6.8 L2 288
4.2.6.8.1 L2.Idle 288
4.2.6.8.2 L2.TransmitWake 289
4.2.6.9 Disabled 289
4.2.6.10 Loopback 289
4.2.6.10.1 Loopback.Entry 290
4.2.6.10.2 Loopback.Active 292
4.2.6.10.3 Loopback.Exit 292
4.2.6.11 Hot Reset 293
4.2.7 Clock Tolerance Compensation 294
4.2.7.1 SKP Ordered Set for 8b/10b Encoding 294
4.2.7.2 SKP Ordered Set for 128b/130b Encoding 294
4.2.7.3 Rules for Transmitters 295
4.2.7.4 Rules for Receivers 296
4.2.8 Compliance Pattern in 8b/10b Encoding 296
4.2.9 Modified Compliance Pattern in 8b/10b Encoding 297
4.2.10 Compliance Pattern in 128b/130b Encoding 298
4.2.11 Modified Compliance Pattern in 128b/130b Encoding 300
4.2.12 Jitter Measurement Pattern in 128b/130b 300
4.2.13 Lane Margining at Receiver 301
4.2.13.1 Receiver Number, Margin Type, Usage Model, and Margin Payload Fields 301
4.2.13.1.1 Step Margin Execution Status 307
4.2.13.1.2 Margin Payload for Step Margin Commands 307

公众号·芯片业者相承双

4.2.13.2 Margin Command and Response Flow 307
4.2.13.3 Receiver Margin Testing Requirements 310
4.3 Retimers 314
4.3.1 Retimer Requirements 315
4.3.2 Supported Retimer Topologies 316
4.3.3 Variables 317
4.3.4 Receiver Impedance Propagation Rules 318
4.3.5 Switching Between Modes 318
4.3.6 Forwarding Rules 318
4.3.6.1 Forwarding Type Rules 319
4.3.6.2 Orientation and Lane Numbers Rules 319
4.3.6.3 Electrical Idle Exit Rules 319
4.3.6.4 Data Rate Change and Determination Rules 321
4.3.6.5 Electrical Idle Entry Rules 322
4.3.6.6 Transmitter Settings Determination Rules 323
4.3.6.7 Ordered Set Modification Rules 324
4.3.6.8 DLLP, TLP, and Logical Idle Modification Rules 326
4.3.6.9 8b/10b Encoding Rules 327
4.3.6.10 8b/10b Scrambling Rules 327
4.3.6.11 Hot Reset Rules 327
4.3.6.12 Disable Link Rules 327
4.3.6.13 Loopback 328
4.3.6.14 Compliance Receive Rules 329
4.3.6.15 Enter Compliance Rules 330
4.3.7 Execution Mode Rules 332
4.3.7.1 CompLoadBoard Rules 333
4.3.7.1.1 CompLoadBoard.Entry 333
4.3.7.1.2 CompLoadBoard.Pattern 333
4.3.7.1.3 CompLoadBoard.Exit 334
4.3.7.2 Link Equalization Rules 334
4.3.7.2.1 Downstream Lanes 334

公众号·芯片业者相承双

4.3.7.2.1.1 Phase 2 334
4.3.7.2.1.2 Phase 3 Active 335
4.3.7.2.1.3 Phase 3 Passive 335
4.3.7.2.2 Upstream Lanes 335
4.3.7.2.2.1 Phase 2 Active 335
4.3.7.2.2.2 Phase 2 Passive 336
4.3.7.2.2.3 Phase 3 336
4.3.7.2.3 Force Timeout 336
4.3.7.3 Slave Loopback 336
4.3.7.3.1 Slave Loopback.Entry 336
4.3.7.3.2 Slave Loopback.Active 337
4.3.7.3.3 Slave Loopback.Exit 337
4.3.8 Retimer Latency 337
4.3.8.1 Measurement 337
4.3.8.2 Maximum Limit on Retimer Latency 338
4.3.8.3 Impacts on Upstream and Downstream Ports 338
4.3.9 SRIS 338
4.3.10 L1 PM Substates Support 339
4.3.11 Retimer Configuration Parameters 340
4.3.11.1 Global Parameters 341
4.3.11.2 Per Physical Pseudo Port Parameters 341
4.3.12 In Band Register Access 342
5 Power Management 343
5.1 Overview 343
5.2 Link State Power Management 344
5.3 PCI-PM Software Compatible Mechanisms 348
5.3.1 Device Power Management States (D-States) of a Function 348
5.3.1.1 D0 State 348
5.3.1.2 D1 State 348
5.3.1.3 D2 State 349
5.3.1.4 D3 State 349

公众号·芯片业者相承双

5.3.1.4.1 $D3_{hot}$ State 349
5.3.1.4.2 $D3_{cold}$ State 351
5.3.2 PM Software Control of the Link Power Management State 351
5.3.2.1 Entry into the L1 State 352
5.3.2.2 Exit from L1 State 354
5.3.2.3 Entry into the L2/L3 Ready State 355
5.3.3 Power Management Event Mechanisms 356
5.3.3.1 Motivation 356
5.3.3.2 Link Wakeup 356
5.3.3.2.1 PME Synchronization 357
5.3.3.3 PM_PME Messages 358
5.3.3.3.1 PM_PME “Backpressure” Deadlock Avoidance 358
5.3.3.4 PME Rules 359
5.3.3.5 PM_PME Delivery State Machine 359
5.4 Native PCI Express Power Management Mechanisms 361
5.4.1 Active State Power Management (ASPM) 361
5.4.1.1 L0s ASPM State 362
5.4.1.1.1 Entry into the L0s State 363
5.4.1.1.2 Exit from the L0s State 363
5.4.1.2 L1 ASPM State 364
5.4.1.2.1 Entry into the L1 State 364
5.4.1.2.2 Exit from the L1 State 369
5.4.1.3 ASPM Configuration 371
5.4.1.3.1 Software Flow for Enabling or Disabling ASPM 373
5.5 L1 PM Substates 374
5.5.1 Entry conditions for L1 PM Substates and L1.0 Requirements 377
5.5.2 L1.1 Requirements 378
5.5.2.1 Exit from L1.1 378
5.5.3 L1.2 Requirements 379
5.5.3.1 L1.2.Entry 380
5.5.3.2 L1.2.Idle 381

公众号·芯片业者相承双

5.5.3.3 L1.2.Exit 381
5.5.3.3.1 Exit from L1.2 381
5.5.4 L1 PM Substates Configuration 382
5.5.5 L1 PM Substates Timing Parameters 383
5.5.6 Link Activation 383
5.6 Auxiliary Power Support 384
5.7 Power Management System Messages and DLLPs 385
5.8 PCI Function Power State Transitions 385
5.9 State Transition Recovery Time Requirements 386
5.10 PCI Bridges and Power Management 386
5.10.1 Switches and PCI Express to PCI Bridges 387
5.11 Power Management Events 387
6 System Architecture 389
6.1 Interrupt and PME Support 389
6.1.1 Rationale for PCI Express Interrupt Model 389
6.1.2 PCI Compatible INTx Emulation 389
6.1.3 INTx Emulation Software Model 389
6.1.4 Message Signaled Interrupt (MSI/MSI-X) Support 390
6.1.5 PME Support 390
6.1.6 Native PME Software Model 390
6.1.7 Legacy PME Software Model 391
6.1.8 Operating System Power Management Notification 391
6.1.9 PME Routing Between PCI Express and PCI Hierarchies 391
6.2 Error Signaling and Logging 391
6.2.1 Scope 392
6.2.2 Error Classification 392
6.2.2.1 Correctable Errors（可纠正错误） 392
6.2.2.2 Uncorrectable Errors（不可纠正错误） 393
6.2.2.2.1 Fatal Errors（致命错误） 393
6.2.2.2.2 Non-Fatal Errors（非致命错误） 393
6.2.3 Error Signaling 393

公众号·芯片业者相承双

6.2.3.1 Completion Status 393
6.2.3.2 Error Messages 393
6.2.3.2.1 Uncorrectable Error Severity Programming (Advanced Error Reporting) 394
6.2.3.2.2 Masking Individual Errors 394
6.2.3.2.3 Error Pollution 395
6.2.3.2.4 Advisory Non-Fatal Error Cases 395
6.2.3.2.4.1 Completer Sending a Completion with UR/CA Status 396
6.2.3.2.4.2 Intermediate Receiver 396
6.2.3.2.4.3 Ultimate PCI Express Receiver of a Poisoned TLP 397
6.2.3.2.4.4 Requester with Completion Timeout 397
6.2.3.2.4.5 Receiver of an Unexpected Completion 397
6.2.3.2.5 Requester Receiving a Completion with UR/CA Status 397
6.2.3.3 Error Forwarding (Data Poisoning) 398
6.2.3.4 Optional Error Checking 398
6.2.4 Error Logging 398
6.2.4.1 RC Considerations (Advanced Error Reporting) 399
6.2.4.1.1 Error Source Identification 399
6.2.4.1.2 Interrupt Generation 399
6.2.4.2 Multiple Error Handling (Advanced Error Reporting Capability) 400
6.2.4.3 Advisory Non-Fatal Error Logging 400
6.2.4.4 TLP Prefix Logging 401
6.2.5 Sequence of Device Error Signaling and Logging Operations 401
6.2.6 Error Message Controls 402
6.2.7 Error Listing and Rules 403
6.2.7.1 Conventional PCI Mapping 407
6.2.8 Virtual PCI Bridge Error Handling 407
6.2.8.1 Error Message Forwarding and PCI Mapping for Bridge - Rules 407
6.2.9 Internal Errors 408
6.3 Virtual Channel Support 409
6.3.1 Introduction and Scope 409
6.3.2 TC/VC Mapping and Example Usage 409

公众号·芯片业者相承双

6.3.3 VC Arbitration 410
6.3.3.1 Traffic Flow and Switch Arbitration Model 411
6.3.3.2 VC Arbitration - Arbitration Between VCs 413
6.3.3.2.1 Strict Priority Arbitration Model 414
6.3.3.2.2 Round Robin Arbitration Model 414
6.3.3.3 Port Arbitration - Arbitration Within VC 414
6.3.3.4 Multi-Function Devices and Function Arbitration 415
6.3.4 Isochronous Support 417
6.3.4.1 Rules for Software Configuration 417
6.3.4.2 Rules for Requesters 417
6.3.4.3 Rules for Completers 417
6.3.4.4 Rules for Switches and RCes 418
6.3.4.5 Rules for Multi-Function Devices 418
6.4 Device Synchronization 418
6.5 Locked Transactions 419
6.5.1 Introduction 419
6.5.2 Initiation and Propagation of Locked Transactions - Rules 419
6.5.3 Switches and Lock - Rules 420
6.5.4 PCI Express/PCI Bridges and Lock - Rules 421
6.5.5 RC and Lock - Rules 421
6.5.6 Legacy Endpoints 421
6.5.7 PCI Express Endpoints 421
6.6 PCI Express Reset - Rules 421
6.6.1 Conventional Reset（常规复位） 421
6.6.2 Function-Level Reset (FLR) 423
6.7 PCI Express Hot-Plug Support 425
6.7.1 Elements of Hot-Plug 426
6.7.1.1 Indicators （指示灯） 426
6.7.1.1.1 Attention Indicator 426
6.7.1.1.2 Power Indicator 427
6.7.1.2 Manually-operated Retention Latch (MRL) 428

公众号·芯片业者相承双

6.7.1.3 MRL Sensor 428
6.7.1.4 Electromechanical Interlock（机械锁扣） 428
6.7.1.5 Attention Button 428
6.7.1.6 Software User Interface 429
6.7.1.7 Slot Numbering 429
6.7.1.8 Power Controller 429
6.7.2 Registers Grouped by Hot-Plug Element Association 430
6.7.2.1 Attention Button Registers 430
6.7.2.2 Attention Indicator Registers 430
6.7.2.3 Power Indicator Registers 430
6.7.2.4 Power Controller Registers 430
6.7.2.5 Presence Detect Registers 430
6.7.2.6 MRL Sensor Registers 431
6.7.2.7 Electromechanical Interlock Registers 431
6.7.2.8 Command Completed Registers 431
6.7.2.9 Port Capabilities and Slot Information Registers 431
6.7.2.10 Hot-Plug Interrupt Control Register 431
6.7.3 PCI Express Hot-Plug Events 432
6.7.3.1 Slot Events 432
6.7.3.2 Command Completed Events 432
6.7.3.3 Data Link Layer State Changed Events 433
6.7.3.4 Software Notification of Hot-Plug Events 433
6.7.4 Firmware Support for Hot-Plug 434
6.8 Power Budgeting Capability 434
6.8.1 System Power Budgeting Process Recommendations 434
6.9 Slot Power Limit Control 435
6.10 RC Topology Discovery 436
6.11 Link Speed Management 438
6.12 Access Control Services (ACS) 439
6.12.1 ACS Component Capability Requirements 440
6.12.1.1 ACS Downstream Ports 440

公众号·芯片业者相承双

6.12.1.2 ACS Functions in Multi-Function Devices 442
6.12.1.3 Functions in Single-Function Devices 443
6.12.2 Interoperability 443
6.12.3 ACS Peer-to-Peer Control Interactions 443
6.12.4 ACS Violation Error Handling 444
6.12.5 ACS Redirection Impacts on Ordering Rules 444
6.12.5.1 Completions Passing Posted Requests 444
6.12.5.2 Requests Passing Posted Requests 445
6.13 Alternative Routing-ID Interpretation (ARI) 446
6.14 Multicast Operations 448
6.14.1 Multicast TLP Processing 448
6.14.2 Multicast Ordering 450
6.14.3 Multicast Capability Structure Field Updates 450
6.14.4 MC Blocked TLP Processing 450
6.14.5 MC_Overlay Mechanism 451
6.15 Atomic Operations (AtomicOps) 453
6.15.1 AtomicOp Use Models and Benefits 454
6.15.2 AtomicOp Transaction Protocol Summary 454
6.15.3 RC Support for AtomicOps 455
6.15.3.1 Root Ports with AtomicOp Completer Capabilities 455
6.15.3.2 Root Ports with AtomicOp Routing Capability 455
6.15.3.3 RCs with AtomicOp Requester Capabilities 456
6.15.4 Switch Support for AtomicOps 456
6.16 Dynamic Power Allocation (DPA) Capability 456
6.16.1 DPA Capability with Multi-Function Devices 457
6.17 TLP Processing Hints (TPH) 457
6.17.1 Processing Hints 458
6.17.2 Steering Tags 458
6.17.3 ST Modes of Operation 459
6.17.4 TPH Capability 460
6.18 Latency Tolerance Reporting (LTR) Mechanism 460

公众号·芯片业者相承双

6.19 Optimized Buffer Flush/Fill (OBFF) Mechanism 463
6.20 PASID TLP Layout 466
6.20.1 Managing PASID TLP Prefix Usage 466
6.20.2 PASID TLP Layout 467
6.20.2.1 PASID field 468
6.20.2.2 Execute Requested 469
6.20.2.3 Privileged Mode Requested 469
6.21 Lightweight Notification (LN) Protocol 470
6.21.1 LN Protocol Operation 471
6.21.2 LN Registration Management 472
6.21.3 LN Ordering Considerations 473
6.21.4 LN Software Configuration 473
6.21.5 LN Protocol Summary 473
6.22 Precision Time Measurement (PTM) Mechanism 474
6.22.1 Introduction 474
6.22.2 PTM Link Protocol 475
6.22.3 Configuration and Operational Requirements 477
6.22.3.1 PTM Requester Role 478
6.22.3.2 PTM Responder Role 479
6.22.3.3 PTM Time Source Role - Rules Specific to Switches 480
6.23 Readiness Notifications (RN) 481
6.23.1 Device Readiness Status (DRS) 481
6.23.2 Function Readiness Status (FRS) 482
6.23.3 FRS Queuing 483
6.24 Enhanced Allocation 483
6.25 Emergency Power Reduction State 485
6.26 Hierarchy ID Message 487
6.27 Flattening Portal Bridge (FPB) 490
6.27.1 Introduction 490
6.27.2 Hardware and Software Requirements 492
6.28 Vital Product Data (VPD) 497

公众号·芯片业者相承双

6.28.1 VPD Format 498
6.28.2 VPD Definitions 499
6.28.2.1 VPD Large and Small Resource Data Tags 499
6.28.2.2 Read-Only Fields 499
6.28.2.3 Read/Write Fields 500
6.28.2.4 VPD Example 500
6.29 Native PCIe Enclosure Management（NPEM） 502
6.30 Conventional PCI Advanced Features Operation 504
7 Software Initialization and Configuration 507
7.1 Configuration Topology 507
7.2 PCI Express Configuration Mechanisms 508
7.2.1 PCI 3.0 Compatible Configuration Mechanism 510
7.2.2 PCI Express Enhanced Configuration Access Mechanism (ECAM) 510
7.2.2.1 Host Bridge Requirements 511
7.2.2.2 PCI Express Device Requirements 512
7.2.3 RC Register Block 512
7.3 Configuration Transaction Rules 512
7.4 Configuration Register Types 512
7.5 PCI and PCIe Capabilities Required by the Base Spec for all Ports 513
7.5.1 PCI-Compatible Configuration Registers 513
7.5.1.1 Type 0/1 Common Configuration Space 513
7.5.1.1.1 Vendor ID Register (Offset 00h) 514
7.5.1.1.2 Device ID Register (Offset 02h) 514
7.5.1.1.3 Command Register (Offset 04h) 514
7.5.1.1.4 Status Register (Offset 06h) 517
7.5.1.1.5 Revision ID Register (Offset 08h) 519
7.5.1.1.6 Class Code Register (Offset 09h) 519
7.5.1.1.7 Cache Line Size Register (Offset 0Ch) 520
7.5.1.1.8 Latency Timer Register (Offset 0Dh) 520
7.5.1.1.9 Header Type Register (Offset 0Eh) 520
7.5.1.1.10 BIST Register (Offset 0Fh) 521

公众号·芯片业者相承双

7.5.1.1.11 Capabilities Pointer (Offset 34h) 522
7.5.1.1.12 Interrupt Line Register (Offset 3Ch) 522
7.5.1.1.13 Interrupt Pin Register (Offset 3Dh) 522
7.5.1.1.14 Error Registers 522
7.5.1.2 Type 0 Configuration Space Header 523
7.5.1.2.1 Base Address Registers (Offset 10h - 24h) 523
7.5.1.2.2 Cardbus CIS Pointer Register (Offset 28h) 525
7.5.1.2.3 Subsystem Vendor ID Register/Subsystem ID Register (Offset 2Ch/2Eh) 525
7.5.1.2.4 Expansion ROM Base Address Register (Offset 30h) 526
7.5.1.2.5 Min_Gnt Register/Max_Lat Register (Offset 3Eh/3Fh) 528
7.5.1.3 Type 1 Configuration Space Header 528
7.5.1.3.1 Type 1 Base Address Registers (Offset 10h-14h) 529
7.5.1.3.2 Primary Bus Number Register (Offset 18h) 529
7.5.1.3.3 Secondary Bus Number Register (Offset 19h) 529
7.5.1.3.4 Subordinate Bus Number Register (Offset 1Ah) 530
7.5.1.3.5 Secondary Latency Timer (Offset 1Bh) 530
7.5.1.3.6 I/O Base/I/O Limit Registers(Offset 1Ch/1Dh) 530
7.5.1.3.7 Secondary Status Register (Offset 1Eh) 531
7.5.1.3.8 Memory Base Register/Memory Limit Register(Offset 20h/22h) 532
7.5.1.3.9 Prefetchable Memory Base/Prefetchable Memory Limit Registers (Offset 24h/26h) 532
7.5.1.3.10 Prefetchable Base Upper 32 Bits/Prefetchable Limit Upper 32 Bits Registers (Offset 28h/2Ch) 533
7.5.1.3.11 I/O Base Upper 16 Bits/I/O Limit Upper 16 Bits Registers (Offset 30h/32h) 533
7.5.1.3.12 Expansion ROM Base Address Register (Offset 38h) 533
7.5.1.3.13 Bridge Control Register (Offset 3Eh) 534
7.5.2 PCI Power Management Capability Structure 536
7.5.2.1 Power Management Capabilities Register (Offset 00h) 536
7.5.2.2 Power Management Control/Status Register (Offset 04h) 538
7.5.2.3 Data (Offset 07h) 540
7.5.3 PCI Express Capability Structure 541
7.5.3.1 PCI Express Capability List Register (Offset 00h) 542
7.5.3.2 PCI Express Capabilities Register (Offset 02h) 543

公众号·芯片业者相承双

7.5.3.3 Device Capabilities Register (Offset 04h) 544
7.5.3.4 Device Control Register (Offset 08h) 548
7.5.3.5 Device Status Register (Offset 0Ah) 552
7.5.3.6 Link Capabilities Register (Offset 0Ch) 554
7.5.3.7 Link Control Register (Offset 10h) 558
7.5.3.8 Link Status Register (Offset 12h) 563
7.5.3.9 Slot Capabilities Register (Offset 14h) 565
7.5.3.10 Slot Control Register (Offset 18h) 566
7.5.3.11 Slot Status Register (Offset 1Ah) 570
7.5.3.12 Root Control Register (Offset 1Ch) 571
7.5.3.13 Root Capabilities Register (Offset 1Eh) 572
7.5.3.14 Root Status Register (Offset 20h) 573
7.5.3.15 Device Capabilities 2 Register (Offset 24h) 573
7.5.3.16 Device Control 2 Register (Offset 28h) 578
7.5.3.17 Device Status 2 Register (Offset 2Ah) 581
7.5.3.18 Link Capabilities 2 Register (Offset 2Ch) 582
7.5.3.19 Link Control 2 Register (Offset 30h) 584
7.5.3.20 Link Status 2 Register (Offset 32h) 588
7.5.3.21 Slot Capabilities 2 Register (Offset 34h) 591
7.5.3.22 Slot Control 2 Register (Offset 38h) 591
7.5.3.23 Slot Status 2 Register (Offset 3Ah) 591
7.6 PCI Express Extended Capabilities 592
7.6.1 Extended Capabilities in Configuration Space 592
7.6.2 Extended Capabilities in the RC Register Block 592
7.6.3 PCI Express Extended Capability Header 592
7.7 PCI and PCIe Capabilities Required by the Base Spec in Some Situations 593
7.7.1 MSI Capability Structures 593
7.7.1.1 MSI Capability Header (Offset 00h) 594
7.7.1.2 Message Control Register for MSI (Offset 02h) 595
7.7.1.3 Message Address Register for MSI (Offset 04h) 597
7.7.1.4 Message Upper Address Register for MSI (Offset 08h) 597

公众号·芯片业者相承双

7.7.1.5 Message Data Register for MSI (Offset 08h or 0Ch) 598
7.7.1.6 Extended Message Data Register for MSI (Optional) 598
7.7.1.7 Mask Bits Register for MSI (Offset 0Ch or 10h） 599
7.7.1.8 Pending Bits Register for MSI (Offset 10h or 14h) 599
7.7.2 MSI-X Capability and Table Structure 600
7.7.2.1 MSI-X Capability Header (Offset 00h) 603
7.7.2.2 Message Control Register for MSI-X (Offset 02h) 603
7.7.2.3 Table Offset/Table BIR Register for MSI-X (Offset 04h) 604
7.7.2.4 PBA Offset/PBA BIR Register for MSI-X (Offset 08h) 605
7.7.2.5 Message Address Register for MSI-X Table Entries 605
7.7.2.6 Message Upper Address Register for MSI-X Table Entries 606
7.7.2.7 Message Data Register for MSI-X Table Entries 606
7.7.2.8 Vector Control Register for MSI-X Table Entries 606
7.7.2.9 Pending Bits Register for MSI-X PBA Entries 607
7.7.3 Secondary PCI Express Extended Capability 608
7.7.3.1 Secondary PCI Express Extended Capability Header (Offset 00h) 609
7.7.3.2 Link Control 3 Register (Offset 04h) 609
7.7.3.3 Lane Error Status Register (Offset 08h) 610
7.7.3.4 Lane Equalization Control Register (Offset 0Ch) 611
7.7.4 Data Link Feature Extended Capability 613
7.7.4.1 Data Link Feature Extended Capability Header (Offset 00h) 613
7.7.4.2 Data Link Feature Capabilities Register (Offset 04h) 614
7.7.4.3 Data Link Feature Status Register (Offset 08h) 615
7.7.5 Physical Layer 16.0 GT/s Extended Capability 615
7.7.5.1 Physical Layer 16.0 GT/s Extended Capability Header (Offset 00h) 616
7.7.5.2 16.0 GT/s Capabilities Register (Offset 04h) 617
7.7.5.3 16.0 GT/s Control Register (Offset 08h) 617
7.7.5.4 16.0 GT/s Status Register (Offset 0Ch) 617
7.7.5.5 16.0 GT/s Local Data Parity Mismatch Status Register (Offset 10h) 618
7.7.5.6 16.0 GT/s First Retimer Data Parity Mismatch Status Register (Offset 14h) 619
7.7.5.7 16.0 GT/s Second Retimer Data Parity Mismatch Status Register (Offset 18h) 619

公众号·芯片业者相承双

7.7.5.8 Physical Layer 16.0 GT/s Reserved (Offset 1Ch) 620
7.7.5.9 16.0 GT/s Lane Equalization Control Register (Offsets 20h to 3Ch) 620
7.7.6 Physical Layer 32.0 GT/s Extended Capability 621
7.7.6.1 Physical Layer 32.0 GT/s Extended Capability Header (Offset 00h) 622
7.7.6.2 32.0 GT/s Capabilities Register (Offset 04h) 623
7.7.6.3 32.0 GT/s Control Register (Offset 08h) 623
7.7.6.4 32.0 GT/s Status Register (Offset 0Ch) 624
7.7.6.5 Received Modified TS Data 1 Register (Offset 10h) 626
7.7.6.6 Received Modified TS Data 2 Register (Offset 14h) 627
7.7.6.7 Transmitted Modified TS Data 1 Register (Offset 18h) 628
7.7.6.8 Transmitted Modified TS Data 2 Register (Offset 1Ch) 629
7.7.6.9 32.0 GT/s Lane Equalization Control Register (Offset 20h) 630
7.7.7 Lane Margining at the Receiver Extended Capability 631
7.7.7.1 Lane Margining at the Receiver Extended Capability Header (Offset 00h) 633
7.7.7.2 Margining Port Capabilities Register (Offset 04h) 633
7.7.7.3 Margining Port Status Register (Offset 06h) 633
7.7.7.4 Margining Lane Control Register (Offset 08h) 634
7.7.7.5 Margining Lane Status Register (Offset 0Ah) 635
7.7.8 ACS Extended Capability 636
7.7.8.1 ACS Extended Capability Header (Offset 00h) 636
7.7.8.2 ACS Capability Register (Offset 04h) 637
7.7.8.3 ACS Control Register (Offset 06h) 638
7.7.8.4 Egress Control Vector Register (Offset 08h) 640
7.8 Common PCI and PCIe Capabilities 641
7.8.1 Power Budgeting Extended Capability 641
7.8.1.1 Power Budgeting Extended Capability Header (Offset 00h) 642
7.8.1.2 Power Budgeting Data Select Register (Offset 04h) 642
7.8.1.3 Power Budgeting Data Register (Offset 08h) 642
7.8.1.4 Power Budgeting Capability Register (Offset 0Ch) 644
7.8.2 Latency Tolerance Reporting (LTR) Extended Capability 645
7.8.2.1 LTR Extended Capability Header (Offset 00h) 645

公众号·芯片业者柏承双

7.8.2.2 Max Snoop Latency Register (Offset 04h) 646
7.8.2.3 Max No-Snoop Latency Register (Offset 06h) 646
7.8.3 L1 PM Substates Extended Capability 647
7.8.3.1 L1 PM Substates Extended Capability Header (Offset 00h) 647
7.8.3.2 L1 PM Substates Capabilities Register (Offset 04h) 648
7.8.3.3 L1 PM Substates Control 1 Register (Offset 08h) 649
7.8.3.4 L1 PM Substates Control 2 Register (Offset 0Ch) 651
7.8.3.5 L1 PM Substates Status Register (Offset 10h) 652
7.8.4 Advanced Error Reporting Extended Capability 652
7.8.4.1 Advanced Error Reporting Extended Capability Header (Offset 00h) 653
7.8.4.2 Uncorrectable Error Status Register (Offset 04h) 654
7.8.4.3 Uncorrectable Error Mask Register (Offset 08h) 655
7.8.4.4 Uncorrectable Error Severity Register (Offset 0Ch) 657
7.8.4.5 Correctable Error Status Register (Offset 10h) 658
7.8.4.6 Correctable Error Mask Register (Offset 14h) 659
7.8.4.7 Advanced Error Capabilities and Control Register (Offset 18h) 660
7.8.4.8 Header Log Register (Offset 1Ch) 661
7.8.4.9 Root Error Command Register (Offset 2Ch) 662
7.8.4.10 Root Error Status Register (Offset 30h) 663
7.8.4.11 Error Source Identification Register (Offset 34h) 665
7.8.4.12 TLP Prefix Log Register (Offset 38h) 665
7.8.5 Enhanced Allocation Capability Structure (EA) 666
7.8.5.1 Enhanced Allocation Capability First DW (Offset 00h) 666
7.8.5.2 Enhanced Allocation Capability Second DW (Offset 04h)[Type 1 Functions Only] 666
7.8.5.3 Enhanced Allocation Per-Entry Format (Offset 04h or 08h) 667
7.8.6 Resizable BAR Extended Capability 671
7.8.6.1 Resizable BAR Extended Capability Header (Offset 00h) 673
7.8.6.2 Resizable BAR Capability Register 673
7.8.6.3 Resizable BAR Control Register 676
7.8.7 ARI Extended Capability 678
7.8.7.1 ARI Extended Capability Header (Offset 00h) 678

公众号·芯片业者相承双

7.8.7.2 ARI Capability Register (Offset 04h) 679
7.8.7.3 ARI Control Register (Offset 06h) 679
7.8.8 PASID Extended Capability Structure 680
7.8.8.1 PASID Extended Capability Header (Offset 00h) 680
7.8.8.2 PASID Capability Register (Offset 04h) 681
7.8.8.3 PASID Control Register (Offset 06h) 681
7.8.9 FRS Queueing Extended Capability 682
7.8.9.1 FRS Queueing Extended Capability Header (Offset 00h) 682
7.8.9.2 FRS Queueing Capability Register (Offset 04h) 683
7.8.9.3 FRS Queueing Status Register (Offset 08h) 684
7.8.9.4 FRS Queueing Control Register (Offset 0Ah) 684
7.8.9.5 FRS Message Queue Register (Offset 0Ch) 684
7.8.10 Flattening Portal Bridge (FPB) Capability 685
7.8.10.1 FPB Capability Header (Offset 00h) 686
7.8.10.2 FPB Capabilities Register (Offset 04h) 686
7.8.10.3 FPB RID Vector Control 1 Register (Offset 08h) 688
7.8.10.4 FPB RID Vector Control 2 Register (Offset 0Ch) 689
7.8.10.5 FPB MEM Low Vector Control Register (Offset 10h) 690
7.8.10.6 FPB MEM High Vector Control 1 Register (Offset 14h) 691
7.8.10.7 FPB MEM High Vector Control 2 Register (Offset 18h) 693
7.8.10.8 FPB Vector Access Control Register (Offset 1Ch) 694
7.8.10.9 FPB Vector Access Data Register (Offset 20h) 695
7.9 Additional PCI and PCIe Capabilities 695
7.9.1 Virtual Channel Extended Capability 695
7.9.1.1 Virtual Channel Extended Capability Header (Offset 00h) 696
7.9.1.2 Port VC Capability Register 1 (Offset 04h) 697
7.9.1.3 Port VC Capability Register 2 (Offset 08h) 698
7.9.1.4 Port VC Control Register (Offset 0Ch) 1
7.9.1.5 Port VC Status Register (Offset 0Eh) 1
7.9.1.6 VC Resource Capability Register 1
7.9.1.7 VC Resource Control Register 1

公众号·芯片业者相承双

7.9.1.8 VC Resource Status Register 1
7.9.1.9 VC Arbitration Table 1
7.9.1.10 Port Arbitration Table 1
7.9.2 Multi-Function Virtual Channel Extended Capability 1
7.9.2.1 MFVC Extended Capability Header (Offset 00h) 1
7.9.2.2 MFVC Port VC Capability Register 1 (Offset 04h) 1
7.9.2.3 MFVC Port VC Capability Register 2 (Offset 08h) 1
7.9.2.4 MFVC Port VC Control Register (Offset 0Ch) 1
7.9.2.5 MFVC Port VC Status Register (Offset 0Eh) 1
7.9.2.6 MFVC VC Resource Capability Register 1
7.9.2.7 MFVC VC Resource Control Register 1
7.9.2.8 MFVC VC Resource Status Register 1
7.9.2.9 MFVC VC Arbitration Table 1
7.9.2.10 Function Arbitration Table 1
7.9.3 Device Serial Number Extended Capability 1
7.9.3.1 Device Serial Number Extended Capability Header (Offset 00h) 1
7.9.3.2 Serial Number Register (Offset 04h) 1
7.9.4 Vendor-Specific Capability 1
7.9.5 Vendor-Specific Extended Capability 1
7.9.5.1 Vendor-Specific Extended Capability Header (Offset 00h) 1
7.9.5.2 Vendor-Specific Header (Offset 04h) 1
7.9.6 Designated Vendor-Specific Extended Capability (DVSEC) 1
7.9.6.1 Designated Vendor-Specific Extended Capability Header (Offset 00h) 1
7.9.6.2 Designated Vendor-Specific Header 1 (Offset 04h) 1
7.9.6.3 Designated Vendor-Specific Header 2 (Offset 08h) 1
7.9.7 RCRB Header Extended Capability 1
7.9.7.1 RCRB Header Extended Capability Header (Offset 00h) 1
7.9.7.2 RCRB Vendor ID and Device ID register (Offset 04h) 1
7.9.7.3 RCRB Capabilities register (Offset 08h) 1
7.9.7.4 RCRB Control register (Offset 0Ch) 1
7.9.8 Root Complex Link Declaration Extended Capability 1

公众号·芯片业者相承双

7.9.8.1 Root Complex Link Declaration Extended Capability Header (Offset 00h) 1
7.9.8.2 Element Self Description Register (Offset 04h) 1
7.9.8.3 Link Entries 1
7.9.8.3.1 Link Description Register 1
7.9.8.3.2 Link Address 1
7.9.8.3.2.1 Link Address for Link Type 0 1
7.9.8.3.2.2 Link Address for Link Type 1 1
7.9.9 Root Complex Internal Link Control Extended Capability 1
7.9.9.1 Root Complex Internal Link Control Extended Capability Header (Offset 00h) 1
7.9.9.2 Root Complex Link Capabilities Register (Offset 04h) 1
7.9.9.3 Root Complex Link Control Register (Offset 08h) 1
7.9.9.4 Root Complex Link Status Register (Offset 0Ah) 1
7.9.10 Root Complex Event Collector Endpoint Association Extended Capability 1
7.9.10.1 Root Complex Event Collector Endpoint Association Extended Capability Header (Offset 00h) 1
7.9.10.2 Association Bitmap for RCiEPs (Offset 04h) 1
7.9.10.3 RCEC Associated Bus Numbers Register (Offset 08h) 1
7.9.11 Multicast Extended Capability 1
7.9.11.1 Multicast Extended Capability Header (Offset 00h) 1
7.9.11.2 Multicast Capability Register (Offset 04h) 1
7.9.11.3 Multicast Control Register (Offset 06h) 1
7.9.11.4 MC_Base_Address Register (Offset 08h) 1
7.9.11.5 MC_Receive Register (Offset 10h) 1
7.9.11.6 MC_Block_All Register (Offset 18h) 1
7.9.11.7 MC_Block_Untranslated Register (Offset 20h) 1
7.9.11.8 MC_Overlay_BAR Register (Offset 28h) 1
7.9.12 Dynamic Power Allocation Extended Capability (DPA Capability) 1
7.9.12.1 DPA Extended Capability Header (Offset 00h) 1
7.9.12.2 DPA Capability Register (Offset 04h) 1
7.9.12.3 DPA Latency Indicator Register (Offset 08h) 1
7.9.12.4 DPA Status Register (Offset 0Ch) 1
7.9.12.5 DPA Control Register (Offset 0Eh) 1

公众号·芯片业者相承双

7.9.12.6 DPA Power Allocation Array 1
7.9.13 TPH Requester Extended Capability 1
7.9.13.1 TPH Requester Extended Capability Header (Offset 00h) 1
7.9.13.2 TPH Requester Capability Register (Offset 04h) 1
7.9.13.3 TPH Requester Control Register (Offset 08h) 1
7.9.13.4 TPH ST Table (Starting from Offset 0Ch) 1
7.9.14 LN Requester Extended Capability (LNR Capability) 1
7.9.14.1 LNR Extended Capability Header (Offset 00h) 1
7.9.14.2 LNR Capability Register (Offset 04h) 1
7.9.14.3 LNR Control Register (Offset 06h) 1
7.9.15 DPC Extended Capability 1
7.9.15.1 DPC Extended Capability Header (Offset 00h) 1
7.9.15.2 DPC Capability Register (Offset 04h) 1
7.9.15.3 DPC Control Register (Offset 06h) 1
7.9.15.4 DPC Status Register (Offset 08h) 1
7.9.15.5 DPC Error Source ID Register (Offset 0Ah) 1
7.9.15.6 RP PIO Status Register (Offset 0Ch) 1
7.9.15.7 RP PIO Mask Register (Offset 10h) 1
7.9.15.8 RP PIO Severity Register (Offset 14h) 1
7.9.15.9 RP PIO SysError Register (Offset 18h) 1
7.9.15.10 RP PIO Exception Register (Offset 1Ch) 1
7.9.15.11 RP PIO Header Log Register (Offset 20h) 1
7.9.15.12 RP PIO ImpSpec Log Register (Offset 30h) 1
7.9.15.13 RP PIO TLP Prefix Log Register (Offset 34h) 1
7.9.16 Precision Time Management Extended Capability (PTM Capability) 1
7.9.16.1 PTM Extended Capability Header (Offset 00h) 1
7.9.16.2 PTM Capability Register (Offset 04h) 1
7.9.16.3 PTM Control Register (Offset 08h) 1
7.9.17 Readiness Time Reporting Extended Capability 1
7.9.17.1 Readiness Time Reporting Extended Capability Header (Offset 00h) 1
7.9.17.2 Readiness Time Reporting 1 Register (Offset 04h) 1

公众号·芯片业者相承双

7.9.17.3 Readiness Time Reporting 2 Register (Offset 08h) 1
7.9.18 Hierarchy ID Extended Capability 1
7.9.18.1 Hierarchy ID Extended Capability Header (Offset 00h) 1
7.9.18.2 Hierarchy ID Status Register (Offset 04h) 1
7.9.18.3 Hierarchy ID Data Register (Offset 08h) 1
7.9.18.4 Hierarchy ID GUID 1 Register (Offset 0Ch) 1
7.9.18.5 Hierarchy ID GUID 2 Register (Offset 10h) 1
7.9.18.6 Hierarchy ID GUID 3 Register (Offset 14h) 1
7.9.18.7 Hierarchy ID GUID 4 Register (Offset 18h) 1
7.9.18.8 Hierarchy ID GUID 5 Register (Offset 1Ch) 1
7.9.19 Vital Product Data Capability (VPD Capability) 1
7.9.19.1 VPD Address Register 1
7.9.19.2 VPD Data Register 1
7.9.20 Native PCIe Enclosure Management Extended Capability (NPEM Extended Capability) 1
7.9.20.1 NPEM Extended Capability Header (Offset 00h) 1
7.9.20.2 NPEM Capability Register (Offset 04h) 1
7.9.20.3 NPEM Control Register (Offset 08h) 1
7.9.20.4 NPEM Status Register (Offset 0Ch) 1
7.9.21 Alternate Protocol Extended Capability 1
7.9.21.1 Alternate Protocol Extended Capability Header (Offset 00h) 1
7.9.21.2 Alternate Protocol Capabilities Register (Offset 04h) 1
7.9.21.3 Alternate Protocol Control Register (Offset 08h) 1
7.9.21.4 Alternate Protocol Data 1 Register (Offset 0Ch) 1
7.9.21.5 Alternate Protocol Data 2 Register (Offset 10h) 1
7.9.21.6 Alternate Protocol Selective Enable Mask Register (Offset 14h) 1
7.9.22 Conventional PCI Advanced Features Capability (AF) 1
7.9.22.1 Advanced Features Capability Header (Offset 00h) 1
7.9.22.2 AF Capabilities Register (Offset 03h) 1
7.9.22.3 Conventional PCI Advanced Features Control Register (Offset 04h) 1
7.9.22.4 AF Status Register (Offset 05h) 1
7.9.23 SFI Extended Capability 1

公众号·芯片业者柏承双

7.9.23.1 SFI Extended Capability Header (Offset 00h) 1
7.9.23.2 SFI Capability Register (Offset 04h) 1
7.9.23.3 SFI Control Register (Offset 06h) 1
7.9.23.4 SFI Status Register (Offset 08h) 1
7.9.23.5 SFI CAM Address Register (Offset 0Ch) 1
7.9.23.6 SFI CAM Data Register (Offset 10h) 1
7.9.24 Subsystem ID and Sybsystem Vendor ID Capability 1
8 Electrical Sub-Block 1
8.1 Electrical Specification Introduction 1
8.2 Interoperability Criteria 1
8.2.1 Data Rates 1
8.2.2 Refclk Architectures 1
8.3 Transmitter Specification 1
8.3.1 Measurement Setup for Characterizing Transmitters 1
8.3.1.1 Breakout and Replica Channels 1
8.3.2 Voltage Level Definitions 1
8.3.3 Tx Voltage Parameters 1
8.3.3.1 2.5 and 5.0 GT/s Transmitter Equalization 1
8.3.3.2 8.0, 16.0, and 32.0 GT/s Transmitter Equalization 1
8.3.3.3 Tx Equalization Presets 1
8.3.3.4 Measuring Tx Equalization for 2.5 GT/s and 5.0 GT/s 1
8.3.3.5 Measuring Presets at 8.0 GT/s, 16.0 GT/s, and 32.0 GT/s 1
8.3.3.6 Method for Measuring VTX-DIFF-PP at 2.5 GT/s and 5.0 GT/s 1
8.3.3.7 Method for Measuring VTX-DIFF-PP at 8.0 GT/s, 16.0 GT/s, and 32.0 GT/s 1
8.3.3.8 Coefficient Range and Tolerance 1
8.3.3.9 EIEOS and VTX-EIEOS-FS and VTX-EIEOS-RS Limits 1
8.3.3.10 Reduced Swing Signaling 1
8.3.3.11 Effective Tx Package Loss at 8.0 GT/s, 16.0 GT/s and 32.0 GT/s 1
8.3.4 Transmitter Margining 1
8.3.5 Tx Jitter Parameters 1
8.3.5.1 Post Processing Steps to Extract Jitter 1

公众号·芯片业者相承双

8.3.5.2 Applying CTLE or De-embedding 1
8.3.5.3 Independent Refclk Measurement and Post Processing 1
8.3.5.4 Embedded and Non Embedded Refclk Measurement and Post Processing 1
8.3.5.5 Behavioral CDR Characteristics 1
8.3.5.6 Data Dependent and Uncorrelated Jitter 1
8.3.5.7 Data Dependent Jitter 1
8.3.5.8 Uncorrelated Total Jitter and Deterministic Jitter (Dual Dirac Model) (TTX-UTJ and TTX-UDJDD) 1
8.3.5.9 Random Jitter (TTX-RJ) (informative) 1
8.3.5.10 Uncorrelated Total and Deterministic PWJ (TTX-UPW-TJ and TTX-UPW-DJDD) 1
8.3.6 Data Rate Dependent Parameters 1
8.3.7 Tx and Rx Return Loss 1
8.3.8 Transmitter PLL Bandwidth and Peaking 1
8.3.8.1 2.5 GT/s and 5.0 GT/s Tx PLL Bandwidth and Peaking 1
8.3.8.2 8.0 GT/s, 16.0 GT/s and 32.0 GT/s Tx PLL Bandwidth and Peaking 1
8.3.8.3 Series Capacitors 1
8.3.9 Data Rate Independent Tx Parameters 1
8.4 Receiver Specifications 1
8.4.1 Receiver Stressed Eye Specification 1
8.4.1.1 Breakout and Replica Channels 1
8.4.1.2 Calibration Channel Insertion Loss Characteristics 1
8.4.1.3 Post Processing Procedures 1
8.4.1.4 Behavioral Rx Package Models 1
8.4.1.5 Behavioral CDR Model 1
8.4.1.6 No Behavioral Rx Equalization for 2.5 and 5.0 GT/s 1
8.4.1.7 Behavioral Rx Equalization for 8.0, 16.0, and 32.0 GT/s 1
8.4.1.8 Behavioral CTLE (8.0 and 16.0 GT/s) 1
8.4.1.9 Behavioral CTLE (32.0 GT/s) 1
8.4.1.10 Behavioral DFE (8.0, 16.0, and 32.0 GT/s Only) 1
8.4.2 Stressed Eye Test 1
8.4.2.1 Procedure for Calibrating a Stressed EH/EW Eye 1
8.4.2.1.1 Post Processing Tool Requirements 1

公众号·芯片业者柏承双

8.4.2.2 Procedure for Testing Rx DUT 1
8.4.2.2.1 Sj Mask 1
8.4.2.3 Receiver Refclk Modes 1
8.4.2.3.1 Common Refclk Mode 1
8.4.2.3.2 Independent Refclk Mode 1
8.4.3 Common Receiver Parameters 1
8.4.3.1 5.0 GT/s Exit From Idle Detect (EFI) 1
8.4.3.2 Receiver Return Loss 1
8.4.4 Lane Margining at the Receiver - Electrical Requirements 1
8.4.5 Low Frequency and Miscellaneous Signaling Requirements 1
8.4.5.1 ESD Standards 1
8.4.5.2 Channel AC Coupling Capacitors 1
8.4.5.3 Short Circuit Requirements 1
8.4.5.4 Transmitter and Receiver Termination 1
8.4.5.5 Electrical Idle 1
8.4.5.6 DC Common Mode Voltage 1
8.4.5.7 Receiver Detection 1
8.5 Channel Tolerancing 1
8.5.1 Channel Compliance Testing 1
8.5.1.1 Behavioral Transmitter and Receiver Package Models 1
8.5.1.2 Measuring Package Performance (16.0 GT/s only) 1
8.5.1.3 Simulation Tool Requirements 1
8.5.1.3.1 Simulation Tool Chain Inputs 1
8.5.1.3.2 Processing Steps 1
8.5.1.3.3 Simulation Tool Outputs 1
8.5.1.3.4 Open Source Simulation Tool 1
8.5.1.4 Behavioral Transmitter Parameters 1
8.5.1.4.1 Deriving Voltage and Jitter Parameters 1
8.5.1.4.2 Optimizing Tx/Rx Equalization (8.0 GT/s, 16.0 GT/s and 32.0 GT/s only) 1
8.5.1.4.3 Pass/Fail Eye Characteristics 1
8.5.1.4.4 Characterizing Channel Common Mode Noise 1

公众号·芯片业者相承双

8.5.1.4.5 Verifying $V_{CH-IDLE-DET-DIFF-pp}$ 1
8.6 Refclk Specifications 1
8.6.1 Refclk Test Setup 1
8.6.2 REFCLK AC Specifications 1
8.6.3 Data Rate Independent Refclk Parameters 1
8.6.3.1 Low Frequency Refclk Jitter Limits 1
8.6.4 Refclk Architectures Supported 1
8.6.5 Filtering Functions Applied to Raw Data 1
8.6.5.1 PLL Filter Transfer Function Example 1
8.6.5.2 CDR Transfer Function Examples 1
8.6.6 Common Refclk Rx Architecture (CC) 1
8.6.6.1 Determining the Number of PLL BW and peaking Combinations 1
8.6.6.2 CDR and PLL BW and Peaking Limits for Common Refclk 1
8.6.7 Jitter Limits for Refclk Architectures 1
8.6.8 Form Factor Requirements for RefClock Architectures 1
9 Single Root I/O Virtualization and Sharing 1
9.1 SR-IOV Architectural Overview 1
9.1.1 PCI Technologies Interoperability 1
9.2 SR-IOV Initialization and Resource Allocation 1
9.2.1 SR-IOV Resource Discovery 1
9.2.1.1 Configuring SR-IOV Capabilities 1
9.2.1.1.1 Configuring the VF BAR Mechanisms 1
9.2.1.2 VF Discovery 1
9.2.1.3 Function Dependency Lists 1
9.2.1.4 Interrupt Resource Allocation 1
9.2.2 SR-IOV Reset Mechanisms 1
9.2.2.1 SR-IOV Conventional Reset 1
9.2.2.2 FLR That Targets a VF 1
9.2.2.3 FLR That Targets a PF 1
9.2.3 IOV Re-initialization and Reallocation 1
9.2.4 VF Migration（VF 迁移） 1

公众号·芯片业者相承双

9.2.4.1 Initial VF State 1
9.2.4.2 VF Migration State Transitions 1
9.3 Configuration 1
9.3.1 SR-IOV Configuration Overview 1
9.3.2 Configuration Space 1
9.3.3 SR-IOV Extended Capability 1
9.3.3.1 SR-IOV Extended Capability Header (Offset 00h) 1
9.3.3.2 SR-IOV Capabilities Register (04h) 1
9.3.3.2.1 VF Migration Capable 1
9.3.3.2.2 ARI Capable Hierarchy Preserved 1
9.3.3.2.3 VF 10-Bit Tag Requester Supported 1
9.3.3.2.4 VF Migration Interrupt Message Number 1
9.3.3.3 SR-IOV Control Register (Offset 08h) 1
9.3.3.3.1 VF Enable 1
9.3.3.3.2 VF Migration Enable 1
9.3.3.3.3 VF Migration Interrupt Enable 1
9.3.3.3.4 VF MSE (Memory Space Enable) 1
9.3.3.3.5 ARI Capable Hierarchy 1
9.3.3.4 SR-IOV Status Register (Offset 0Ah) 1
9.3.3.4.1 VF Migration Status 1
9.3.3.5 InitialVFs (Offset 0Ch) 1
9.3.3.6 TotalVFs (Offset 0Eh) 1
9.3.3.7 NumVFs (Offset 10h) 1
9.3.3.8 Function Dependency Link (Offset 12h) 1
9.3.3.9 First VF Offset (Offset 14h) 1
9.3.3.10 VF Stride (Offset 16h) 1
9.3.3.11 VF Device ID (Offset 1Ah) 1
9.3.3.12 Supported Page Sizes (Offset 1Ch) 1
9.3.3.13 System Page Size (Offset 20h) 1
9.3.3.14 VF BAR0 (Offset 24h), VF BAR1 (Offset 28h), VF BAR2 (Offset 2Ch), VF BAR3 (Offset 30h), VF BAR4 (Offset 34h), VF BAR5 (Offset 38h) 1

公众号·芯片业者相承双

9.3.3.15 VF Migration State Array Offset (Offset 3Ch) 1
9.3.3.15.1 VF Migration State Array 1
9.3.4 PF/VF Configuration Space Header 1
9.3.4.1 PF/VF Type 0 Configuration Space Header 1
9.3.4.1.1 Vendor ID Register Changes (Offset 00h) 1
9.3.4.1.2 Device ID Register Changes (Offset 02h) 1
9.3.4.1.3 Command Register Changes (Offset 04h) 1
9.3.4.1.4 Status Register Changes (Offset 06h) 1
9.3.4.1.5 Revision ID Register Changes (Offset 08h) 1
9.3.4.1.6 Class Code Register Changes (Offset 09h) 1
9.3.4.1.7 Cache Line Size Register Changes (Offset 0Ch) 1
9.3.4.1.8 Latency Timer Register Changes (Offset 0Dh) 1
9.3.4.1.9 Header Type Register Changes (Offset 0Eh) 1
9.3.4.1.10 BIST Register Changes (Offset 0Fh) 1
9.3.4.1.11 Base Address Registers Register Changes (Offset 10h, 14h, … 24h) 1
9.3.4.1.12 Cardbus CIS Pointer Register Changes (Offset 28h) 1
9.3.4.1.13 Subsystem Vendor ID Register Changes (Offset 2Ch) 1
9.3.4.1.14 Subsystem ID Register Changes (Offset 2Eh) 1
9.3.4.1.15 Expansion ROM Base Address Register Register Changes (Offset 30h) 1
9.3.4.1.16 Capabilities Pointer Register Changes (Offset 34h) 1
9.3.4.1.17 Interrupt Line Register Changes (Offset 3Ch) 1
9.3.4.1.18 Interrupt Pin Register Changes (Offset 3Dh) 1
9.3.4.1.19 Min_Gnt Register/Max_Lat Register Changes (Offset 3Eh/3Fh) 1
9.3.5 PCI Express Capability Changes 1
9.3.5.1 PCI Express Capabilities Register Changes (Offset 00h) 1
9.3.5.2 PCI Express Capabilities Register Changes (Offset 02h) 1
9.3.5.3 Device Capabilities Register Changes (Offset 04h) 1
9.3.5.4 Device Control Register Changes (Offset 08h) 1
9.3.5.5 Device Status Register Changes (Offset 0Ah) 1
9.3.5.6 Link Capabilities Register Changes (Offset 0Ch) 1
9.3.5.7 Link Control Register Changes (Offset 10h) 1

公众号·芯片业者相承双

9.3.5.8 Link Status Register Changes (Offset 12h) 1
9.3.5.9 Device Capabilities 2 Register Changes (Offset 24h) 1
9.3.5.10 Device Control 2 Register Changes (Offset 28h) 1
9.3.5.11 Device Status 2 Register Changes (Offset 2Ah) 1
9.3.5.12 Link Capabilities 2 Register Changes (Offset 2Ch) 1
9.3.5.13 Link Control 2 Register Changes (Offset 30h) 1
9.3.5.14 Link Status 2 Register Changes (Offset 32h) 1
9.3.6 PCI Standard Capabilities 1
9.3.6.1 VPD Capability 1
9.3.7 PCI Express Extended Capabilities Changes 1
9.3.7.1 Virtual Channel/MFVC 1
9.3.7.2 Device Serial Number 1
9.3.7.3 Power Budgeting 1
9.3.7.4 Resizable BAR 1
9.3.7.5 VF Resizable BAR Extended Capability 1
9.3.7.5.1 VF Resizable BAR Extended Capability Header (Offset 00h) 1
9.3.7.5.2 VF Resizable BAR Capability Register (Offset 04h) 1
9.3.7.5.3 VF Resizable BAR Control Register (Offset 08h) 1
9.3.7.6 Access Control Services (ACS) Extended Capability Changes 1
9.3.7.7 Alternative Routing ID Interpretation Extended Capability (ARI) Changes 1
9.3.7.8 Address Translation Services Extended Capability Changes (ATS) 1
9.3.7.9 MR-IOV Changes 1
9.3.7.10 Multicast Changes 1
9.3.7.11 Page Request Interface Changes (PRI) 1
9.3.7.12 Dynamic Power Allocation Changes (DPA) 1
9.3.7.13 TPH Requester Changes (TPH) 1
9.3.7.14 PASID Changes 1
9.3.7.15 Readiness Time Reporting Extended Capability Changes 1
9.4 SR-IOV Error Handling 1
9.4.1 Baseline Error Reporting 1
9.4.2 Advanced Error Reporting 1

公众号·芯片业者相承双

9.4.2.1 VF Header Log 1
9.4.2.2 Advanced Error Reporting Capability Changes 1
9.4.2.3 Advanced Error Reporting Extended Capability Header Changes (Offset 00h) 1
9.4.2.4 Uncorrectable Error Status Register Changes (Offset 04h) 1
9.4.2.5 Uncorrectable Error Mask Register Changes (Offset 08h) 1
9.4.2.6 Uncorrectable Error Severity Register Changes (Offset 0Ch) 1
9.4.2.7 Correctable Error Status Register Changes (Offset 10h) 1
9.4.2.8 Correctable Error Mask Register Changes (Offset 14h) 1
9.4.2.9 Advanced Error Capabilities and Control Register Changes (Offset 18h) 1
9.4.2.10 Header Log Register Changes (Offset 1Ch) 1
9.4.2.11 Root Error Command Register Changes (Offset 2Ch) 1
9.4.2.12 Root Error Status Register Changes (Offset 30h) 1
9.4.2.13 Error Source Identification Register Changes (Offset 34h) 1
9.4.2.14 TLP Prefix Log Register Changes (Offset 38h) 1
9.5 SR-IOV Interrupts 1
9.5.1 Interrupt Mechanisms 1
9.5.1.1 MSI Interrupts 1
9.5.1.2 MSI-X Interrupts 1
9.5.1.3 Address Range Isolation 1
9.6 SR-IOV Power Management 1
9.6.1 VF Device Power Management States 1
9.6.2 PF Device Power Management States 1
9.6.3 Link Power Management State 1
9.6.4 VF Power Management Capability 1
9.6.5 VF EmergencyPower Reduction State 1
10 ATS Specification 1
10.1 ATS Architectural Overview 1
10.1.1 Address Translation Services (ATS) Overview 1
10.1.2 Page Request Interface Extension 1
10.1.3 Process Address Space ID (PASID) 1
10.2 ATS Translation Services 1

公众号·芯片业者相承双

10.2.1 Memory Requests with Address Type 1
10.2.2 Translation Requests 1
10.2.2.1 Attribute Field 1
10.2.2.2 Length Field 1
10.2.2.3 Tag Field 1
10.2.2.4 Untranslated Address Field 1
10.2.2.5 No Write (NW) Flag 1
10.2.2.6 PASID TLP Prefix on Translation Request 1
10.2.3 Translation Completion 1
10.2.3.1 Translated Address Field 1
10.2.3.2 Translation Range Size (S) Field 1
10.2.3.3 Non-snooped (N) Field 1
10.2.3.4 Untranslated Access Only (U) Field 1
10.2.3.5 Read (R) and Write (W) Fields 1
10.2.3.6 Execute Permitted (Exe) 1
10.2.3.7 Privileged Mode Access (Priv) 1
10.2.3.8 Global Mapping (Global) 1
10.2.4 Completions with Multiple Translations 1
10.3 ATS Invalidation 1
10.3.1 Invalidate Request 1
10.3.2 Invalidate Completion 1
10.3.3 Invalidate Completion Semantics 1
10.3.4 Request Acceptance Rules 1
10.3.5 Invalidate Flow Control 1
10.3.6 Invalidate Ordering Semantics 1
10.3.7 Implicit Invalidation Events 1
10.3.8 PASID TLP Prefix and Global Invalidate 1
10.4 Page Request Services 1
10.4.1 Page Request Message 1
10.4.1.1 PASID TLP Prefix Usage 1
10.4.1.2 Managing PASID TLP Prefix Usage on PRG Requests 1

公众号·芯片业者相承双

10.4.1.2.1 Stop Marker Messages 1
10.4.2 Page Request Group Response Message 1
10.4.2.1 Response Code Field 1
10.4.2.2 PASID TLP Prefix Usage on PRG Responses 1
10.5 ATS Configuration 1
10.5.1 ATS Extended Capability 1
10.5.1.1 ATS Extended Capability Header (Offset 00h) 1
10.5.1.2 ATS Capability Register (Offset 04h) 1
10.5.1.3 ATS Control Register (Offset 06h) 1
10.5.2 Page Request Extended Capability Structure 1
10.5.2.1 Page Request Extended Capability Header (Offset 00h) 1
10.5.2.2 Page Request Control Register (Offset 04h) 1
10.5.2.3 Page Request Status Register (Offset 06h) 1
10.5.2.4 Outstanding Page Request Capacity (Offset 08h) 1
10.5.2.5 Outstanding Page Request Allocation (Offset 0Ch) 1
Appendixes 1
A Isochronous Applications 1
B Symbol Encoding 1
C Physical Layer Appendix 1
D Request Dependencies 1
E ID-Based Ordering Usage 1
E.1 Introduction 1
E.2 Potential Benefits with IDO Use 1
E.2.1 Benefits for MFD/RP Direct Connect 1
E.2.2 Benefits for Switched Environments 1
E.2.3 Benefits for Integrated Endpoints 1
E.2.4 IDO Use in Conjunction with RO 1
E.3 When to Use IDO 1
E.4 When Not to Use IDO 1
E.4.1 When Not to Use IDO with Endpoints 1
E.4.2 When Not to Use IDO with Root Ports 1

公众号·芯片业者相承双

E.5 Software Control of IDO Use 1
E.5.1 Software Control of Endpoint IDO Use 1
E.5.2 Software Control of Root Port IDO Use 1
F Message Code Usage 1
G Protocol Multiplexing 1
补充知识点 1
1 物理层的弹性缓冲（Elastic Buffer） 1
2 关于 head-of-line blocking 1
3 关于 de-embeding 技术 1
3.1 De-embedding 技术定义 1
3.2 De-embedding 技术理论简介 1
3.3 应用：UHF 频段 RFID 近场天线的阻抗测量方法 1
4 扩频时钟（SSC，Spread Spectrum Clocking） 1
5 PCIe 演进方向之 CCIX 简介 1
5.1 CCIX 分层架构 1
5.2 CCIX 系统拓扑样例 1
5.3 CCIX 一致性分层架构 1
5.4 CCIX 数据流样例 1
5.5 CCIX 软件 1
6 GenZ，CXL，NVLINK，OpenCAPI，CCIX 乱战 1
7 路由机制 1
7.1 地址路由 1
7.2 ID 路由 1
7.3 隐式路由 1

公众号·芯片业者相承双

Status of This Document

NOTE

High Performance Systems may run out of tags

在 32.0 GT/s 时，具有较高端到端延迟的系统，即使有 10-bit Tag，单个 Function 也可能无法有足够的 outstanding request 来达到完整性能。

支持更多 outstanding request 的更改需要与旧组件进行互操作，而不管该组件的链路速度如何。正在考虑针对 PCI Express Base 4.0 规范的 ECN 定义可选行为以解决此问题。

NOTE

Background on the new Document Process

新的 PCISIG 文档系统是 w3c Respec 工具的变体（请参阅 https://github.com/w3c/respec/wiki）。Respec 是一种广泛使用的工具，用于支持万维网规范。PCISIG 变体是 https://github.com/sglaser/respec。Respec 和 PCISIG 变体都是开源（MIT 许可）Javascript 库。它们在作者的浏览器中运行，提供快速的编辑/审查周期，无需安装任何特殊工具。

Respec 建立在 HTML5 之上，HTML5 是万维网 http://www.w3.org/TR/html5/ 的文档格式。HTML 是一种基于文本的文档格式，它允许我们部署软件开发常用的工具（git、持续集成、构建脚本等），以更好地管理和控制规范开发过程。

PCISIG 对 Respec 的增强支持更接近现有 PCISIG 实践的文档格式以及注册数字的自动创建（消除大约一半的手动绘制数字）。

公众号·芯片业务相关双

Terms and Acronyms

8b/10b

此编码策略用于 5.0GT/s 和 2.5GT/s 的 PCIe 物理层编码。

10-Bit Tags

10-Bit Tag capability，用于提供 10 比特的 Tag 字段。

Access Control Services, ACS

访问控制服务。一组能力和控制寄存器，用于在 PCI Express 组件内实现对路由的访问控制。

ACS Violation

用于描述 Completer 检测到针对 Posted 或 Non-Posted Request 的访问控制违规。

Adapter

适配器。一般指附加卡（add-in card）或类似模块。

Advanced Error Reporting, AER

高级错误报告。

Advertise (Credits)

通告。在流控制的上下文中使用，接收方发送有关其流控制信用可用性信息的行为。

Alternative Routing-ID, ARI

在 PCIe 拓扑结构中对 Requester ID 和 Completer ID 以及 Routing ID 的另一种解释。

ARI Device

ARI 设备。与 Upstream Port 相关联的设备，其每个 Function 都实现一个 ARI Extended Capability structure。

ARI Downstream Port

ARI 下游端口。这表示一个支持 ARI Forwarding 的 Switch Downstream Port 或 Root Port。

ARI Forwarding

ARI 转发。使 ARI Device 正上方的 Downstream Port 能够访问此 Device 的扩展 Function 的功能。启用 ARI Forwarding 可确保确定何时将 Type 1 Configuration Request 转换为 Type 0 Configuration Request 的逻辑不再强制限制传统的 Device Number 字段为 0。

Asserted

信号有效状态。

Async Removal

异步移除。指的是从插槽中移除适配器或电缆而无需与操作系统进行同步（即，以异步方式无需按下按钮等）。

Atomic Operation, AtomicOp

原子操作。其中针对内存空间中某个地址的单个 PCI Express 事务读该地址的值，可能会向该地址写入一个新值，然后返回原始值。这个对该地址的 read-modify-write 序列是原子地执行的。AtomicOps 包括 FetchAdd、Swap 和 CAS。

公众号：芯片业者相承双

Attribute

指的是 TLP Header 中的相关字段。

Base Address Register, BAR

基址寄存器存在于配置空间中，用于确定 Function 所需的系统内存空间量，并为映射到 Function 内存空间提供基地址。基址寄存器可以映射到内存空间或 I/O 空间。

Beacon

信标。用于退出 L2 链路电源管理状态的可选实现的 30 kHz 至 500 MHz 带内信号。这是唤醒 L2 链路的两种机制之一（请参阅 Wakeup）。

Bridge

桥。PCI-SIG 定义的系统元件之一。将 PCI/PCI-X 段或 PCI Express Port 与内部组件互连或与另一个 PCI/PCI-X 总线段或 PCI Express Port 虚拟或实际连接的 Function。Root Complex 或 Switch 中的 virtual Bridge 必须使用本规范中描述的软件配置接口。

by-1, x1

具有一个 Physical Lane 的链路或端口。注：规范中 Link 包含一条或多条 Lane。

by-8, x8

具有 8 个 Physical Lane 的链路或端口。

by-N, xN

具有 N 个 Physical Lane 的链路或端口。

Compare and Swap, CAS

将目标地址的值与指定值进行比较的 AtomicOp，如果匹配，则将另一个指定值写回该地址。无论如何，将返回地址的原始值。

Character

字符，1 个字节。注意与 Symbol 的概念区分，8b/10b 编码后的 10 比特符号叫做 Symbol。

Clear

表示清 0 操作。

cold reset

冷复位。上电引起的复位。

Completer

完成者。指的是终止或"完成"给定 Request 的 Function，它在适当时生成 Competition。通常，请求所针对的 Function 充当 Completer。当无法纠正的错误阻止请求到达其目标 Function 时，检测和处理错误的 Function 充当 Completer。

Completer Abort, CA

1. 一种用于 posted 或 non-posted Request 的状态，由于违反了 Completer 的编程模型或与 Completer 相关的不可恢复的错误，Completer 永久无法成功完成。
2. 对于在 Completer 处遭遇 Completer Abort 的 non-posted Request，状态指示与 Competition 一起返回。

公众号：芯片业者相承双

Completer ID

完成者 ID。Completer 的 Bus Number、 Device Number 和 Function Number 的组合，唯一标识拓扑结构中 Request 的 Completer。对于 ARI Completer ID，传统上用于 Device Number 字段的比特位被用来扩展 Function Number 字段，并且 Device Number 隐含为 0。

Completion

完成包。用于终结或部分终结一个事务序列的一个 Packet。Completion 始终对应于先前的 Request，并且在某些情况下，它会包括返回数据。

component

组件。表示一个（single package）物理设备。

Configuration Software

配置软件。负责访问配置空间和配置 PCI/PCIe 总线的系统软件组件。

Configuration Space

配置空间。

Configuration-Ready

配置就绪。当此 Function 确保能以 Successful Completion 的 Competition 响应针对此 Function 的一个配置请求时，就表示此 Function 是"Configuration-Ready"的。

Containment Error Recovery, CER

遏制错误恢复。在适当的软件/固件支持下，Downstream Port Containment (DPC)机制支持的通用错误遏制和恢复方法，可以在不中断应用程序的情况下处理许多无法纠正的错误。

Conventional PCI

传统 PCI。指最初在 PCI Local Bus Specification 中定义的行为或特性。PCI Express Base 4.0 和后续规范包含来自 PCI Local Bus Specification 的相关要求。

Conventional Reset

传统复位。Hot Reset、 Warm Reset 或 Cold Reset 都叫 Conventional Reset。区别于 Function Level Reset (FLR)。

Data Link Layer

数据链路层。

Data Link Layer Packet, DLLP

数据链路层数据包。

data payload

指带数据的 TLP 的有效数据负载。

deasserted

信号无效。

Design for Testability, DFT

公众号·芯片业者相承双

可测性设计。

Device (uppercase 'D')

设备，大写 D。由公共 Bus Number 和 Device Number 标识的单个拓扑结构中的一个或多个 Function 的集合。一个 SR-IOV Device 可能实现了通过一个或多个 SR-IOV Extended Capability structure 配置的附加总线编号和/或设备编号访问的附加 Function。

device (lowercase 'd')

1. 可以表示特定 I/O 类型的物理或逻辑实体。
2. PCI Express 链路任一端的组件。
3. 也可以理解为 Function 的常见不精确同义词，尤其是当设备具有单个 Function 时。

Device Readiness Status, DRS

一种确认设备是否处于 Configuration-Ready 的机制。

Downstream

1. 在 PCI Express 拓扑中，远离 Root Complex 的互连元件或系统元件（Port/component）的方向，称为 Downstream。Switch 中非 Upstream Port 的 Port 就是 Downstream Port。Root Complex 上的所有 Port 都是 Downstream Port。链路上的 Downstream component 是指离 Root Complex 更远的组件。
2. 指信息从 Root Complex 流出的信息流的方向。

Downstream Path

指从 Upstream Pseudo Port Receiver 到 Downstream Pseudo Port Transmitter 的 Retimer 的数据流。

Downstream Port

在 PCIe 总线中，Switch 是一个特殊的设备，该设备由一个上游端口和 2~n 个下游端口构成，PCIe 总线规定：在一个 switch 中可以与 RC（Root Complex）直接或者间接相连的端口为上游端口，在 PCIe 总线中，RC 的位置一般位于上方，这也是上游端口的由来。

在 switch 中除了上游端口外，其他所有的端口都被称为下游端口，下游端口一般与 EP 相连，或者链接下一级 switch 继续扩展 PCIe 链路，其中与上游端口相连的 PCIe 链路称为上游链路，与下游端口相连的 PCIe 链路称为下游链路。

Downstream Port Containment, DPC

在出现不可纠正的错误后自动禁用 Downstream Port 下方的链路，这可以防止错误后的 TLP 传播到上游或下游。

DWORD, DW

4 Byte。

Egress Port

发送端口；即往出发送业务流的 Port。

Egress Port 指发送端口，即数据离开 Switch 使用的端口，Ingress Port 指的是接收端口即数据进入 Switch 使用的端口。Egress Port 和 Ingress Port 与上下游端口没有对应关系，在 Wwitch 中，上下游端口可以作为 Engress 端口，也可以作为 Ingress 端口。

Electrical Idle

电气空闲状态，是一种链路状态，PCI Express 为 Transmitter 和 Receiver 定义了特定的需求。

公众号：芯片业者相聚汉

End-End TLP Prefix

一种 TLP 前缀。

Endpoint

几个定义的系统元件之一。具有 Type 00h Configuration Space header 的 Function。

error detection

确定错误存在的机制，由发现错误的第一个引擎（例如，Malformed TLP）或由信号错误的接收者（例如，poisoned TLP 的接收者）来判定。

error logging

检测器根据错误检测在相关寄存器中设置一个或多个比特位的行为叫做 error logging。检测器可能是错误的最初发现者或信号错误的接收者。

error reporting

在广义上，表示一般的错误通知。在 Device Control Register 的上下文中，表现为发送错误消息。在 Root Error Command Register 的上下文中，表现为由于接收到错误消息而发出中断信号。

error signaling

一个引擎通过 (1) 发送错误消息，(2) 发送带有 UR/CA Status 的 Competition，或 (3) poisoning TLP 来通知另一个引擎错误的过程叫做 error signaling。

Extension Device

用于扩展链路物理长度的组件。Retimer？Redriver？

Extended Function

指在 ARI Device 中，Function Number 大于 7 的 Function 。只有在 ARI 感知软件在 ARI 设备正上方的 Downstream Port 中使能 ARI Forwarding 后，才能访问 Extended Function。

FetchAdd, Fetch and Add

一个 AtomicOp，其中目标地址的值是使用二进制补码算法递增指定的值，忽略任何进位或溢出，并将结果写回该地址。返回地址处的原始值。

Flow Control

用于将接收缓冲区状态从 Receiver 传送到 Transmitter 以防止接收缓冲区溢出并允许 Transmitter 遵守排序规则的机制。

Flow Control Packet, FCP

用于流控制的 DLLP，用于将流控制信息从一个组件中的事务层发送到另一个组件中的事务层。

Function

在 Device 内，配置空间中与单个 Function Number 相关联的可寻址实体。用于指示 Multi-Function Device 的一个 Function，或 Single-Function Device 中的唯一 Function。具体包括 Chapter 9 中定义的特殊类型的 Function，特别是 Physical Function 和 Virtual Function。

Function Group

公众号：芯片业者相承及

在 ARI Device 中，与单个 Function Group Number 关联的一组可配置 Function。Function Group 可以选择性地作为 ARI Device 内多个 Function 之间的 VC 仲裁或访问控制的基础。

Function Level Reset, FLR

一种复位 Endpoint 的 Function 的机制。

Function Readiness Status, FRS

用于指示 Function 已处于 Configuration-Ready 的机制（请参阅 Section 6.23.2 ）。

Fundamental Reset

基本复位。一种硬件机制，用于将所有端口状态设置或返回到本文档中指定的初始条件（参见 Section 6.6 ）。

header

协议报头。

Hierarchy

指树状结构的 PCI Express I/O 互连拓扑，其中用于路由和请求者/完成者标识的配置空间地址 (ID) 是唯一的。一个系统可能包含多个 Hierarchy。

hierarchy domain

源自单个 Root Port 的 Hierarchy 的一部分。

Host Bridge

指在 Root Complex 中将一个或多个主机 CPU 连接到 Hierarchy 的部分。

Hot Reset

一种带内复位机制。

in-band signaling

与使用单独的物理（边带）信号相反，使用两个组件之间的链路来通知事件和状况的方法。本文档中定义的所有机制都可以使用带内信令来实现，尽管在某些 form factor 中可以使用边带信令来代替。

Ingress Port

接收端口。指接收业务流的端口。

Internal Error

与 PCI Express 接口相关的错误，发生在组件内，可能不是由于 PCI Express 接口本身上的数据包或事件或代表在 PCI Express 上启动的事务引起的。

I/O Space

I/O 空间。

isochronous

等时。与时间敏感应用（例如音频或视频应用程序）相关的数据。

invariant

公众号：芯片业者相承双

不变量。TLP Header 或 TLP Prefix 的一个字段，其中包含一个值，当 TLP 流经 PCI Express 拓扑结构时，该值无法合法修改。

Lane

一组差分信号对。

Layer

应用于本规范的区分单位，以帮助阐明关键元件的行为。术语层的使用并不意味着特定的实现。

Link

两个端口及其互连通道的集合。链路是两个组件之间的双单工通信路径。

Link Segment

一个 Port 和一个 Pseudo Port 或两个 Pseudo Port 及其互连通道的集合。Link Segment 是组件和 Retimer 之间或两个 Retimer（两个 Pseudo Port）之间的双单工通信路径。

Lightweight Notification, LN

一种轻量级协议，当感兴趣的 cacheline 更新时，它支持通过硬件机制向 Endpoints 发出通知。

LN Completer, LNC

主机中的服务子系统，用于接收 LN Read/Write Request，并在已注册的 cacheline 更新时发送 LN Message。

LN Completion

TLP Header 中 LN 字段设置为 1 的 Competition。

LN Message

用于通过 LN 协议进行通知的消息。

LN Read

TLP Header 中 LN 字段设置为 1 的 Memory Read Request 。

LN Requester, LNR

Endpoint 中发送 LN Read/Write Request 和接收 LN Message 的客户端子系统。

LN Write

TLP Header 中 LN 字段设置为 1 的 Memory Write Request 。

Local TLP Prefix

一种 TLP 前缀。

Logical Bus

配置空间中具有相同 Bus Number 的设备集合之间的逻辑连接。

Logical Idle

当没有信息（TLP、DLLP 或任何特殊符号）被发送或接收时，一个或多个 Symbol Time 的周期。与电气空闲不同，在逻辑空闲期间，Idle data Symbol 被发送和接收。

公众号：芯片业者拍承双

LTR

Latency Tolerance Reporting 的缩写。

Malformed Packet

违反本规范中定义的特定 TLP 结构规则的 TLP。

Memory Space

存储空间。

Message

消息。

Message Signaled Interrupt, MSI/MSI-X

两种类似但独立的机制，使 Function 能够通过使用 Memory Write Request 将系统指定的 DWORD 数据写入系统指定的地址来请求服务。与 MSI 相比，MSI-X 支持更大的最大向量数以及每个向量的独立消息地址和数据。

Message Space

消息空间。

Multicast, MC

使源发送的单个 Posted Request TLP 能够分发到多个目标的功能和相关机制。

Multicast Group, MCG

作为特定地址范围内 Multicast TLP 目标的一组 Endpoint。

Multicast Hit

接收方确定 TLP 将作为 Multicast TLP 处理称作 Multicast Hit。

Multicast TLP

TLP 可能分发到多个目标，由 TLP 传播的组件中的 Multicast Capability structure 控制。

Multicast Window

以它为目标的 Posted Request TLP 将作为 Multicast TLP 处理的 Memory Space 的区域。

Multi-Function Device, MFD

有多个 Function 的 Device。

Multi-Root I/O Virtualization, MR-IOV

支持 MR-IOV Capability 的 Function。有关其他信息，请参阅 [MR-IOV]。

naturally aligned

起始地址等于 2 的幂的整数倍的数据负载，通常是特定的 2 的幂。例如，64 字节自然对齐意味着字节地址的最低有效 6 位是 00 0000b。

NPEM

公众号·芯片业者相承双

Native PCIe Enclosure Management

OBFF

Optimized Buffer Flush/Fill

Operating System

在整个规范中，术语操作系统和系统软件是指电源管理服务程序、设备驱动程序、用户模式服务程序和/或内核模式服务程序的组合。

orderly removal

一种热插拔移除模型，当用户/操作员希望移除适配器时会通知操作系统，并且操作系统有机会在授予移除许可之前为该事件做准备（例如，停止适配器活动）。

P2P

Peer-to-peer 。

Path

在 Upstream Path 或 Downstream Path 中通过 Retimer 的数据流。

Packet

数据包。

Parts per Million, ppm

应用于频率，在规定的理想频率和测量的频率的长期平均值之间的差异，以百万分之一赫兹为单位。

PCIe®

PCI Express®

PCI Bridge

PCI 桥。

PCI Software Model

初始化、枚举、配置和使用 PCI 兼容设备所需的软件模型，如[PCI-3.0]、[PCI-X-2.0] 和 [PCI-Firmware]中指定的。

Phantom Function Number, PFN

一个无人认领的 Function Number，可用于通过将 PFN 与 Tag identifier 逻辑组合以创建唯一事务标识符来扩展 outstanding transaction identifier 的数量。

Physical Function, PF

实现 SR-IOV Extended Capability structure 并支持 Chapter 9 中定义的 SR-IOV Capability 的 PCI Function。

Physical Lane

跟 Lane 的意思一致。

Physical Layer

物理层。

公众号·芯片业者相承双

Port

1. 逻辑上，指 Component 和 PCI Express Link 之间的接口。

2. 物理上，一组 Transmitter 和 Receiver 位于定义链路的同一芯片上。

Power Management

用于最小化系统功耗、管理系统热限制和最大化系统电池寿命的软件或硬件机制。电源管理涉及系统速度、噪声、电池寿命和交流电源消耗之间的权衡。

PMUX Channel

PMUX 链路上的多路复用通道，配置为传输特定的多路复用协议。见 Appendix G。

PMUX Link

支持和启用协议复用的链路。见 Appendix G。

PMUX Packet

通过 PCI Express 链路传输的非 PCI Express 数据包。见 Appendix G。

Precision Time Measurement, PTM

用于在组件之间传送精确计时信息的可选功能。

Process Address Space ID, PASID

Process Address Space ID 与 Requester ID 一起唯一标识与事务关联的地址空间。

Programmed I/O, PIO

由主机处理器启动的事务序列，通常是执行针对特定地址范围的单个加载或存储指令的结果，但也可以由其他机制（例如 PCI 兼容配置机制）生成。值得注意的是，以 ECAM 地址范围为目标的主机处理器加载或存储会生成配置空间事务。通常存在其他内存映射范围以生成内存空间和 I/O 空间事务。

Pseudo Port

1. 逻辑上，Retimer 和 PCI Express Link Segment 之间的接口。

2. 物理上，指一组发送器和接收器位于定义的 Link Segment 的同一 Retimer 芯片上。

Quality of Service, QoS

影响不同类别业务流的带宽、延迟、抖动、相对优先级等的属性。

QWORD, QW

8 Byte。

RCiEP

Root Complex Integrated Endpoint。

Receiver, Rx

通过链路接收数据包信息的组件。

Receiving Port

公众号·芯片业者相承双

在特定 TLP 或 DLLP 的上下文中，在给定链路上接收数据包的端口。

Re-driver

一个非协议感知的、软件透明的 Extension Device。

repeater

Extension Device 的不精确术语。

Reported Error

受本文档中架构上定义的日志记录和信令要求影响的错误。

Request

请求。

Requester

发起 Request 的 Function。

Requester ID

Requester 的 Bus Number、Device Number 和 Function Number 的组合，在 Hierarchy 中唯一标识 Requester。对于 ARI Requester ID，传统上用于 Device Number 字段的比特位被用于扩展 Function Number 字段，并且 Device Number 隐含为 0。

Reserved

表示目前未定义内容、状态或信息。不允许使用任何 Reserved 区域（例如，包头位字段、配置寄存器比特位）。保留的寄存器字段必须是只读的，并且在读取时必须返回 0（多位字段全为 0）。不得使用寄存器和数据包字段的保留编码。对保留字段值或编码的任何实现依赖都将导致不符合 PCI Express 的实现。在本规范的本版本或任何未来版本中，无法保证此类实现的功能。

Refclk

Reference Clock 的简称。

Retimer

物理层协议感知、软件透明的 Extension Device，形成两个独立的电气 Link Segment。

Root Complex, RC

一个定义的系统元件，包括至少一个 Host Bridge、 Root Port 或 Root Complex Integrated Endpoint。

Root Complex Component

Root Port、 Root Complex Register Block、 Root Complex Integrated Endpoint 和 Root Complex Event Collector 的逻辑聚合。

Root Port, RP

Root Complex 上的 PCI Express Port，通过关联的虚拟 PCI-PCI Bridge 映射 Hierarchy 的一部分。

Routing Element

一个术语，指的是 Root Complex、Switch 或 Bridge 在路由、多播或阻塞 TLP 方面的能力。

Routing ID

标识 PCI Express Function 的 Requester ID 或 Completer ID。

公众号·芯片业者的承双

RP PIO

Root Port Programmed I/O. See Section 6.2.10.3 .

Set

表示信号置 1。

sideband signaling

边带信令。一种使用与形成两个组件之间的链路的信号分开的物理信号来发送事件和状况信息的方法。本文档中定义的所有机制都可以使用带内信令来实现，尽管在某些外形规格中可以使用边带信令来代替。

Single-Function Device, SFD

仅有一个 Function 的 Device。

Single Root I/O Virtualization, SR-IOV

支持本规范定义的 SR-IOV Extended Capability 的一个 Function 。

Single Root PCI Manager, SR-PCIM

负责配置和管理 SR-IOV Extended Capability 和 PF/VF 以及处理相关错误处理的软件。存在多种实现方案；因此，SR-PCIM 的实现超出了本规范的范围。

SR-IOV Device

实现了一个或多个具有 SR-IOV Extended Capability structure 的 Function 的设备。

SSD

Solid State Drive（固态硬盘）。

Swap, Unconditional Swap

一个 AtomicOp，其中将指定值写入目标地址，并返回该地址的原始值。

Switch

连接两个或多个端口以允许数据包从一个端口路由到另一个端口的已定义系统元件。对于配置软件，Switch 表现为 virtual PCI-to-PCI Bridge 的集合。

Symbol

当使用 8b/10b 编码时指的是编码后 10 bit 符号。使用 128b/130b 编码时指的是 Block 中的 8 比特符号。理解时注意"编码后"三个字。

Symbol Time

在 Lane 上放置 Symbol 所需的时间（使用 8b/10b 编码时为 Unit Interval 的 10 倍，使用 128b/130b 编码时为 Unit Interval 的 8 倍）。

System Element

根据不同规则集运行的已定义设备或设备集合。规范定义了以下系统元件：Root Complex、Endpoint、Switch 和 Bridge。

System Image, SI

公众号·芯片业者相承双

在虚拟系统上运行的软件组件，可以为其分配特定的 Function、PF 和 VF。SI 的行为和架构的规范超出了本规范的范围。SI 的示例包括 guest operating system 和共享/非共享受保护域的设备驱动程序。

System Software

包括系统固件（BIOS、UEFI）、操作系统、VMM、管理软件、平台供应商的操作系统附加组件。

Tag

分配给给定 Non-Posted Request 的编号，用于区分该请求与其他请求的 Completion。

TLP Prefix

可以选择性地附加到 TLP 的附加信息。TLP Prefix 分为 Local 或 End-End。一个 TLP 可以有多个 TLP Prefix。见 Section 2.2.10 。

TPH

TLP Processing Hints 的简称。

Transaction Descriptor

数据包头的一个元素，除了 Address、Length 和 Type 之外，还描述了此 Transaction 的属性。

Transaction ID

Transaction Descriptor 的一个组成部分，包括 Requester ID 和 Tag。

Transaction Layer

事务层

Transaction Layer Packet, TLP

事务包。

transaction sequence

单个 Request 和零个或多个与 Requester 执行单个逻辑传输相关的 Competition。

Transceiver

单个芯片上的物理发射器和接收器对。

Transmitter, Tx

通过链路发送数据包信息的组件。

Transmitting Port

在特定 TLP 或 DLLP 的上下文中，在给定链路上发送数据包的端口。

Type 0 Function

实现 Type 0 Configuration Space Header 的 Function。

Type 1 Function

实现 Type 1 Configuration Space Header 的 Function。

Unconditional Swap, Swap

公众号：芯片业者相承双

一个 AtomicOp，其中将指定值写入目标地址，并返回该地址的原始值。

Unit Interval, UI

给定交替 1 和 0 值的重复码型的数据流，Unit Interval 是通过平均电压转换之间的时间间隔来测量的值，在足够长的时间间隔内可以忽略源时钟的所有有意频率调制（见 RX:UI 和 TX:UI）。

Unsupported Request, UR

1. 应用于 posted 或 non-posted Request 的状态，该请求指定 Completer 不支持的某些操作或对某些空间的访问。
2. 对于在 Completer 处遇到 Unsupported Request 的 non-posted Request，状态指示与 Completion 一起返回。

Upstream

1. 靠近 Root Complex 的互连/系统元件(端口/组件)的相对位置。在拓扑上最接近 Root Complex 的 Switch 上的端口是 Upstream Port。仅包含 Endpoint 或 Bridge Function 的组件上的端口是 Upstream Port。链路上的 Upstream component 是更靠近 Root Complex 的组件。
2. 信息流向 Root Complex 的信息流方向为 Upstream。

Upstream Path

通过 Retimer 从 Downstream Pseudo Port Receiver 到 Upstream Pseudo Port Transmitter 的数据流。

variant

当 TLP Header 的一个字段的值在流经 PCI Express 拓扑时，根据本规范的规则可能会修改该值，这种字段叫做 variant。

Virtual Function, VF

与 Physical Function 相关联的 Function。VF 与 PF 和与同一 PF 关联的其他 VF 共享一个或多个物理资源，例如链路。

Virtualization Intermediary, VI

支持一个或多个 SI 的软件组件——通俗地称为管理程序或虚拟机监视器。VI 的行为和体系结构的规范超出了本规范的范围。

wakeup

当处于 L2 链路状态时，组件用于请求重新起用主电源的可选机制。 定义了两种这样的机制：Beacon（使用带内信令）和 WAKE#（使用边带信令）。

warm reset

不需要主电源的基本复位。

公众号·芯片业者相承双

Reference Documents

PCI

PCI-3.0

PCI Local Bus Specification, Revision 3.0

PCIe

PCIe-5.0

PCI Express Base Specification, Revision 5.0

PCIe-4.0

PCI Express Base Specification, Revision 4.0

PCIe-3.1

PCIe-3.1a

PCI Express Base Specification, Revision 3.1a

PCIe-3.0

PCI Express Base Specification, Revision 3.0

PCIE-2.1

PCI Express Base Specification, Revision 2.1

PCIe-2.0

PCI Express Base Specification, Revision 2.0

PCIe-1.1

PCI Express Base Specification, Revision 1.1

PCIe-1.0

PCIe-1.0a

PCI Express Base Specification, Revision 1.0a

CEM

CEM-4.0

PCI Express Card Electromechanical Specification, Revision 4.0

CEM-3.0

PCI Express Card Electromechanical Specification, Revision 3.0

CEM-2.0

PCI Express Card Electromechanical Specification, Revision 2.0

公众号·芯片业者柏承双

ECN-CEM-THERMAL

PCIe CEM Thermal Reporting ECN to the PCI Express Card Electromechanical Specification, Revision 3.0

PCIe-to-PCI-PCI-X-Bridge

PCIe-to-PCI-PCI-X-Bridge-1.0

PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0

Mini-Card

PCI Express Mini Card Electromechanical Specification, Revision 2.1

OCuLink

PCI Express OCuLink Specification, Revision 1.0

M.2

PCI Express M.2 Specification, Revision 1.1

U.2

SFF-8639

PCI Express SFF-8639 Module Specification, Revision 3.0, Version 1.0

Ext-Cabling

PCI Express External Cabling Specification, Revision 2.0

ExpressModule

PCI Express ExpressModule Electromechanical Specification, Revision 1.0

PCI-Hot-Plug

PCI-Hot-Plug-1.1

PCI Hot-Plug Specification, Revision 1.1

PCI-PM

PCI Bus Power Management Interface Specification, Revision 1.2

PCI-Code-and-ID

PCI Code and ID Assignment Specification, Revision 1.11 (or later)

Firmware

PCI Firmware Specification, Revision 3.2

ACPI

Advanced Configuration and Power Interface Specification, Revision 6.2

UEFI

Unified Extensible Firmware Interface (UEFI) Specification, Version 2.8

公众号·芯片业者相承双

EUI-48

EUI-64

Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID)

JEDEC-JESD22-C101

JEDEC JESD22-C101F: Field-Induced Charged-Device Model Test Method for Electrostatic Discharge Withstand Thresholds of Microelectronic Components

JEDEC-JEP155-JEP157

JEDEC JEP155: Recommended ESD Target Levels for HBM/MM Qualification and JEP157 Recommended ESD-CDM Target Levels

ESDA-JEDEC-JS-001-2010

ESDA/JEDEC JS-001-2010: Joint JEDEC/ESDA Standard for Electrostatic Discharge Sensitivity Test - Human Body Model (HBM) - Component Level

ITU-T-Rec.-X.667

ITU T-Rec. X.667: Information technology - Procedures for the operation of object identifier registration authorities: Generation of universally unique identifiers and their use in object identifiers

ISO/IEC-9834-8

ISO/IEC 9834-8: Information technology -- Procedures for the operation of object identifier registration authorities -- Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers

RFC-4122

IETF RFC-4122: A Universally Unique IDentifier (UUID) URN Namespace

PICMG

PICMG

PLUG-PLAY-ISA-1.0a

Plug and Play ISA Specification, Version 1.0a, May 5, 1994

PC-Card

PC-Card

公众号·芯片业者相承双

1 Introduction

本章概述了 PCI Express 架构和关键概念。PCI Express 是一种高性能的通用 I/O 互连，用于各种未来的计算和通信平台。保留了关键的 PCI 属性，例如其 usage model，load-store architecture 和 software interfaces，而其并行总线实现则被高度可扩展的完全串行接口所取代。PCI Express 利用点对点互连，基于 Switch 的技术和分组协议的最新进展来提供新级别的性能和功能。PCI Express 支持的一些高级功能包括电源管理，服务质量（QoS），热插拔支持，数据完整性和错误处理。

1.1 A Third Generation I/O Interconnect

此第三代 I/O 互连的高级需求如下：

- 支持多个细分市场和新兴应用：
  - 桌面，移动设备，工作站，服务器，通信平台和嵌入式设备的统一 I/O 架构。
- 能够提供低成本，多种解决方案：
  - 在系统级别达到或低于 PCI 的成本。
- 支持多种平台互连：
  - 通过连接器或电缆进行芯片级互连或板级互连。
- 新的机械外形规格（mechanical form factors）：
  - [M.2]，[CEM] (Card Electro-Mechanical)，[U.2]，[OCuLink]

  注：Form Factor（外形规格），指的是 PCIe 机械接口元件。

  M.2 接口是一种新的主机接口方案，可以兼容多种通信协议，如 sata、PCIe、USB、HSIC、UART、SMBus 等。

  CEM 指的就是插槽接口。

  M.2 接口的固态硬盘主要优点在于体积小巧、性能出色，比较广泛的用于台式电脑、笔记本、超级本等便携设备中。而 U.2 接口则具备速度更快，2.5 英寸更好的与目前 SATA3.0 接口固态硬盘兼容，适合主流笔记本、台式电脑，未来潜力较大。不过配备 U.2 接口的固态硬盘比较少，尚等待成熟。

  OCuLink 是 PCI-SIG 定义的支持 PCIe 4 Lane 的接口，体积更小巧，用来取代刚刚成为标准的 U.2 SF-8639。最关键的是，它已经为 PCI-E 4.0 和 PCI-E 5.0 标准做好了准备，数据传输率再次翻番至 16GT/s 至 32GT/s。
- PCI 兼容软件模型：
  - PCI 系统配置软件可以无修改的用于枚举和配置 PCI Express 设备。
  - 可以无修改的在现有操作系统上 boot 起来。
  - 无需修改即可支持现有 I/O 设备驱动程序。
  - 通过采用 PCI configuration paradigm 来配置/启用新的 PCI Express Function 的能力。
- 性能：
  - 低开销，低延迟的通信可最大程度地提高应用程序有效负载带宽和链路效率。

公众号·芯片业者柏承双

- 每个引脚的高带宽可最大程度地减少每个设备和连接器接口的引脚数。
- 通过聚合通道和信令频率扩展性能。

- 高级特性：
  - 理解不同的数据类型和排序规则。
  - 电源管理和预算。
    - 识别给定 Function 的电源管理功能的能力。
    - 控制给定 Function 进入相应低功耗状态的能力。
    - 能够接收 Function 当前电源状态的报告。
    - 能够生成从主电源的电源关闭状态唤醒的请求。
    - 能够对设备加电进行排序，以允许在功率预算中采用合适的平台策略。
  - 支持差异化服务的能力，即不同的服务质量（QoS）。
    - 面对队头阻塞，能够为每个 QoS 数据流提供专用的链路资源，以提高结构效率和有效的应用程序级性能。（VC 技术）
    - 每个组件中 QoS 仲裁策略可配置。
    - 能够标记每个数据包的端到端 QoS。
    - 能够创建端到端同步（基于时间的注入速率控制）解决方案。
  - 热插拔支持。
    - 支持已有的 PCI 热插拔解决方案。
    - 支持原生的热插拔解决方案（不使用任何边带信号）。
    - 支持异步移除。
    - 所有外形规格的设备支持统一的软件模型。
  - 数据完整性。
    - 对于所有的 TLP 和 DLLP 都支持链路级数据完整性。（LCRC 技术）
    - 支持高可用性解决方案的端到端数据完整性。（ECRC 技术）
  - 错误处理。
    - 支持 PCI 级别的错误处理。
    - 支持高级错误报告和处理的能力，以改进故障隔离和恢复的解决方案。
  - 工艺技术独立性。
    - 发送器和接收器支持不同的 DC 共模电压。
  - 易于测试。
    - 通过简单地连接到测试设备即可测试电气一致性。

公众号·芯片业者相承双

1.2 PCI Express Link

链路表示两个组件之间的双单工（dual-simplex）通信通道。基本的 PCI Express 链路由两个低压差分驱动信号对组成：发送信号对和接收信号对，如 Figure 1-1 所示。

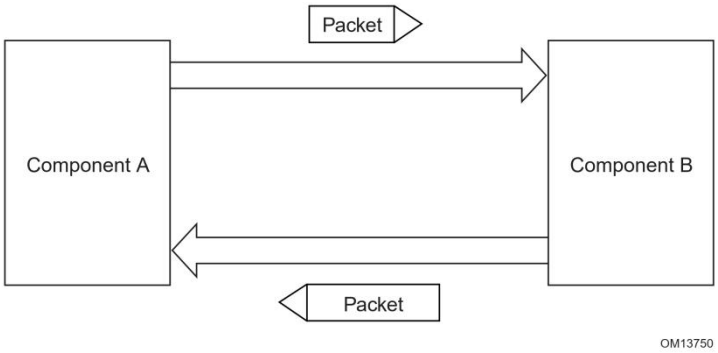


Figure 1-1: PCI Express Link

主要链路特征有：

- 基本链路：PCI Express 链路由双单向差分链路组成，实现为发送对和接收对。使用编码方案（参见 Chapter 4）嵌入数据时钟，以实现很高的数据速率。
- 信号速率：初始化后，每个链路只能在支持的一种信号速率。
  - 对于第一代 PCI Express 技术，2.5 Gigabits/second/Lane/direction 的原始带宽（raw bandwidth）。
  - 第二代提供有效的 5.0 Gigabits/second/Lane/direction 的原始带宽。
  - 第三代提供了有效的 8.0 Gigabits/second/Lane/direction 的原始带宽。
  - 第四代提供了有效的 16.0 Gigabits/second/Lane/direction 的原始带宽。
  - 第五代提供了有效的 32.0 Gigabits/second/Lane/direction 的原始带宽。
- Lanes：链路必须支持至少一个 Lane-每个 Lane 代表一组差分信号对（一对用于传输，一对用于接收）。 为了扩展带宽，链路可以设计成 xN Lane，其中 N 可以是任何支持的链路宽度。以 2.5 GT/s 数据速率运行的 x8 链路表示每个方向的原始带宽总带宽为 20Gigabits/s。本规范描述了 x1，x2，x4，x8，x12，x16 和 x32 通道宽度的操作。
- 初始化：在硬件初始化期间，每个 PCI Express 链路是在链路两端的两个设备协商通道宽度和操作频率之后建立的。不涉及固件或操作系统软件。
- 对称性：每个链路必须在每个方向上支持对称数量的通道，即 x16 链接表示每个方向上有 16 个差分信号对。

1.3 PCI Express Fabric Topology

Figure 1-2 给出了一个拓扑结构的例子，其中涉及的设备类型有 RC (RC)、Endpoints (I/O devices)、 Switch、PCI Express to PCI/PCI-X Bridge 。

公众号·芯片业者相承双

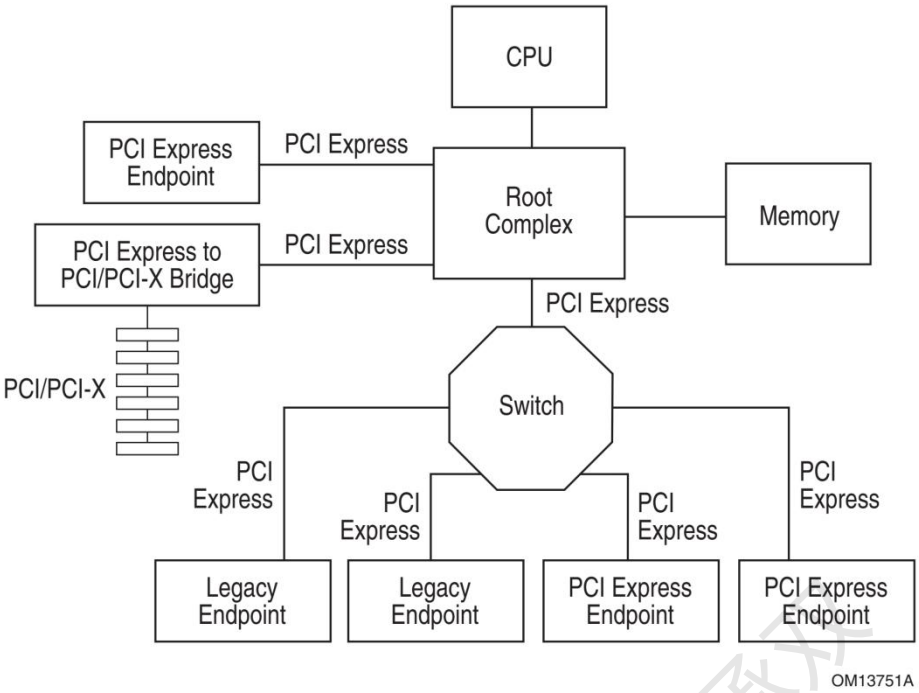


Figure 1-2: Example Topology

1.3.1 RC

- RC 表示将 CPU /内存子系统连接到 I/O 的 I/O 层次结构的根。
- RC 可以支持多个 Port。每个接口定义了一个独立的分层结构。
- 通过 RC 在层次结构 hierarchy domain 之间路由对等事务的功能是可选实现的，并且取决于实现。例如，一个实现可以在 RC 内部内部合并一个真实的或虚拟的 Switch，从而以软件透明的方式实现对等传输的全面支持。与 Switch 的规则不同，在 hierarchy domain 之间对等路由事务时，通常允许 RC 将数据包拆分为较小的数据包，例如，将一个 256 字节有效负载分成两个 128 字节有效负载的数据包。产生的数据包要遵守此规范中包含的一般数据包定义规则（例如，Max_Payload_Size，Read Completion Boundary 等）。组件设计人员应注意，将数据包拆分为较小的数据包可能会对性能造成负面影响，尤其是对于将寻址 PCI Express 至 PCI Express to PCI/PCI-X Bridge 后面的设备的事务进行寻址时。

  例外：支持 Vendor_Defined Message 的对等路由的 RC 不允许将 Vendor_Defined Message 数据包拆分为较小的数据包（除了 128 字节边界以外，即除最后一个数据包外的所有数据包必须是 128 的整数倍），以保证能通过 PCI Express 将消息转发到 PCI/PCI-X Bridge 。有关更多信息，请参阅《*PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0*》。
- RC 可以发起配置请求。
- RC 可以发起 I/O 请求。
  - 允许 RC 生成到选定根端口的 80h 和 84h 位置之一或两个位置的 I/O Request，而不考虑该 Root Port 的 PCI 桥 I/O 解码配置；建议仅在特别需要时才启用此机制。
- RC 一定不能作为 Completer 支持 Lock 语义。
- RC 可以发起锁定请求。

公众号：芯片业者相承双

1.3.2 Endpoint

Endpoint 是这样一种 Function：它可以是 PCI Express 事务的请求者或完成者，可以代表它自己，也可以代表不同的 non-PCI Express 设备（PCI 设备或主机 CPU 除外），例如 PCI Express 连接的图形控制器或 PCI Express USB 主机控制器。Endpoint 分为 Legacy、PCI Express 或 RCIEP。

1.3.2.1 Legacy Endpoint Rules

- Legacy Endpoint 必须是具有 Type 00h Configuration Space header 的一个 Function。
- Legacy Endpoint 可以接受配置请求。
- Legacy Endpoint 可以接受 I/O 请求。
  - Legacy Endpoin 允许接收对位 80h 和 84h 的任意 I/O 请求，而无需考虑该 Endpoint 的 I/O 解码配置。
- Legacy Endpoint 可以生成 I/O 请求。
- 如果设备的旧版软件支持要求所需，则 Legacy Endpoint 可以作为 Completer 支持锁定存储器语义。
- Legacy Endpoint 不能发送 Locked Request 。
- Legacy Endpoint 可以实现 Extended Configuration Space Capabilities，但是会被软件忽略。
- Legacy Endpoint 不能发出超过 4GB 地址空间的请求。
- Legacy Endpoint 如果需要中断源支持，则需要支持 MSI 或 MSI-X 或两者都支持。如果实现了 MSI，则允许 Legacy Endpoint 支持 MSI Capability 的 32 位或 64 位 Message Address 版本。
- Legacy Endpoint 可以通过 32bit Base Address register 访问存储器资源。
- Legacy Endpoint 必须出现在由 RC 发起的层次结构域的其中之一里边。

1.3.2.2 PCI Express Endpoint Rules

- PCI Express Endpoint 必须是具有 Type 00h Configuration Space header 的一个 Function。
- PCI Express Endpoint 可以接受配置请求。
- PCI Express Endpoint 不得依赖操作系统通过 BAR 声明的 I/O 资源分配。
- PCI Express Endpoint 不能生成 I/O 请求。
- PCI Express Endpoint 不能作为 Completer 支持 Locked Request，也不能作为 Requester 生成该请求。兼容 PCI Express 的软件驱动程序和应用程序要防止在访问 PCI Express Endpoint 时使用锁定语义。
- PCI Express Endpoint 可以作为 Requester 生成超过 4GB 地址空间的 Memory 请求。
- PCI Express Endpoint 如果需要支持中断源，则需要支持 MSI 或 MSI-X 或两者都支持。如果实现了 MSI，则允许 Legacy Endpoint 支持 MSI Capability 为 64 位 Message Address 版本。
- 通过 BAR 请求内存资源的 PCI Express Endpoint 必须设置 BAR 的 Prefetchable 字段，除非该范围包含具有读取副作用的地址或有该 Function 不允许写入合并的地址。有关设置 Prefetchable 字段的更多信息，请参见 Section 7.5.1.2.1。

公众号·芯片业者柏承双

- 对于 PCI Express Endpoint，所有设置了 Prefetchable 字段的 BAR 都必须支持 64 位寻址。对于未设置 Prefetchable 字段的所有 BAR，允许使用 32 位寻址。
- BAR 请求的最小内存地址范围是 128 个字节。
- PCI Express Endpoint 必须出现在由 RC 发起的层次结构域的其中之一里边。

1.3.2.3 RC Integrated Endpoint Rules

RCiEP 的应用场景是集成在 RC 内部，专门用于访问存储器。

- RC Integrated Endpoint（RCiEP）是在包含 Root Port 的 RC 的内部逻辑上实现的。
- RCiEP 必须是具有 Type 00h Configuration Space header 的一个 Function。
- RCiEP 必须可以作为 Completer 支持配置请求。
- RCiEP 不得要求通过 BAR 声明 I/O 资源。
- RCiEP 不能生成 I/O 请求。
- RCiEP 不能作为 Completer 支持 Locked Request，也不能作为 Requester 生成该请求。兼容 PCI Express 的软件驱动程序和应用程序要防止在访问 RCiEP 时使用锁定语义。
- 要求作为内存事务的请求者运行的 RCiEP 必须能够生成等于或大于主机能够作为 Completer 处理的地址。
- RCiEP 如果需要中断源支持，则需要支持 MSI 或 MSI-X 或两者都支持。如果实现了 MSI，则允许 RCiEP 支持 MSI Capability 的 32 位或 64 位 Message Address 版本。
- RCiEP 可以通过 32 位 BAR 地址访问存储器资源。
- RCiEP 不能实现 Link Capabilities、Link Status、Link Control、Link Capabilities 2、Link Status 2 和 Link Control 2 寄存器。
- 如果 RCiEP 与一个可选实现的 Root Complex Event Collector 相关联，则 RCiEP 必须通过 Root Complex Event Collector 发送 PME 信令和错误条件信令。
- RCiEP 不能与一个以上的 Root Complex Event Collector 相关联。
- RCiEP 不能实现 ASPM。
- RCiEP 可能无法独立于整个 RC 进行热插拔。
- RCiEP 不得出现在 RC 可见的任何层次结构域中。
- RCiEP 不得出现在 Switch 中。

1.3.3 Switch

Switch 定义为多个虚拟 PCI-to-PCI Bridge 设备的逻辑组合，如 Figure 1-3 所示。所有 Switch 均受以下基本规则约束。

公众号·芯片业者相随双

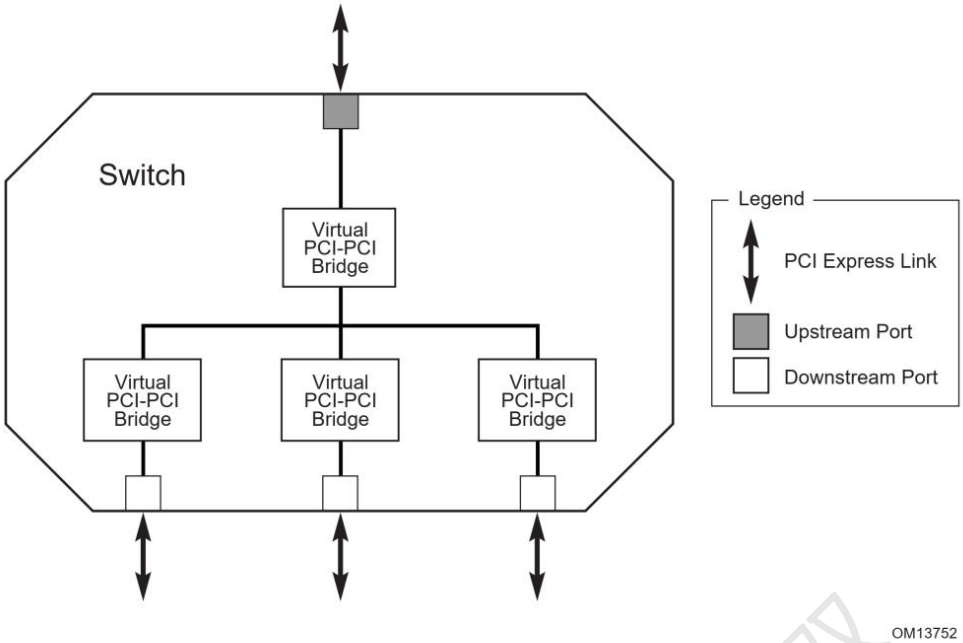


Figure 1-3: Logical Block Diagram of a Switch

- 从配置软件的视角，Switch 是两个或多个逻辑 PCI-to-PCI Bridge 。
- Switch 使用 PCI 桥机制转发事务，例如基于地址的路由，如 Section 6.14。
- 除非本文档中另有说明，否则 Switch 必须支持在任何一组 Port 之间转发所有类型的事务层数据包。
- 必须支持 Locked Request，如 Section 6.5 描述。Switch 的下游端口不需要支持作为锁定请求的发射端口。
- 每个使能的 Switch 端口都要遵循流控制协议。
- 不允许 Switch 将数据包拆分为较小的数据包，例如，不得将具有 256 字节有效负载的单个数据包分为两个 128 字节有效负载的数据包。
- 当在同一虚拟通道上发生争用时，可以使用轮询或加权轮询来实现 Switch 的 Ingress Port 之间的仲裁。 规范在后续章节会对此进行更详细的描述。
- 端点（由 Type 00h Configuration Space header 表示）不得在 Switch 内部总线上的配置软件中显示为表示 Switch 下游端口的虚拟 PCI-to-PCI Bridge 的对等方。

1.3.4 Root Complex Event Collector

- Root Complex Event Collector 为来自 RCiEP 的错误和 PME 消息提供支持。
- Root Complex Event Collector 需要遵循所有的 RCiEP 的规则。
- Root Complex Event Collector 不需要解码任何 Memory 请求或 I/O 请求。
- Root Complex Event Collector 由其 Device/Port Type 值标识（请参阅 Section 7.5.3.2）。
- Root Complex Event Collector 的 Base Class 为 08h，Sub-Class 为 06h，Programming Interface 为 00h。
- Root Complex Event Collector 驻留在 RC 中的总线上。多个 Root Complex Event Collector 允许驻留在单个总线上。
- ~~Root Complex Event~~ Collector ~~与其支持的 RCiEP 位于同一逻辑总线上。~~

公众号：芯片业者柏承双

- ~~多个 Root Complex Event Collector 允许驻留在单个逻辑总线上。~~
- Root Complex Event Collector 通过 Root Complex Event Collector Endpoint Association Capability 显式声明支持的 RCiEP 。
- Root Complex Event Collector 的实现是可选的。

1.3.5 PCI Express to PCI/PCI-X Bridge

- PCI Express to PCI/PCI-X Bridge 设备用于连接 PCI Express 和 PCI/PCI-X 外设。
- ~~PCI Express 至 PCI/PCI-X Bridge 的 PCI Express 端口必须符合本文档的要求。~~

1.4 Hardware/Software Model for Discovery, Configuration and Operation

PCI/PCIe 硬件/软件模型包括 discover、configure 和使用一个 Function 所必需的体系结构，而无需特定于 Function 的知识。关键要素包括：

- 一种配置模型，为系统软件提供发现系统中可用硬件 Function 的方法。（Capability）
- 为诸如内存空间和中断之类的可寻址资源执行基本资源分配的机制。（BAR）
- 启用/禁用 Function 响应收到的 Request 以及 Function 发起 Request 的控制信息。（BME）
- 定义明确的排序和流控制模型，以支持一致/稳健的硬件/软件接口实现。

PCI Express 配置模型支持两种机制：

- PCI 兼容配置机制：PCI 兼容机制支持与 PCI 3.0 或更高版本的操作系统以及它们相应的总线枚举和配置软件的 100%兼容性。
- PCI Express 增强的配置机制：提供了增强的机制以增加可用配置空间的大小并优化访问机制。

每个 PCI Express 链路均通过虚拟 PCI-to-PCI Bridge 结构进行映射，并具有与其关联的逻辑 PCI 总线。虚拟 PCI-to-PCI Bridge 可以是 PCI Express RC Port、Switch Upstream Port 或 Switch Downstream Port 的一部分。Root Port 是一种虚拟的 PCI-to-PCI Bridge 结构，它是从 PCI ExpressRC 中产生 PCI Express 层次结构域的。设备被映射到配置空间，以便每个设备都将响应特定的设备编号。

1.5 PCI Express Layering Overview

PCI Express 结构如 Figure 1-4。

公众号：北大毕业者相承双

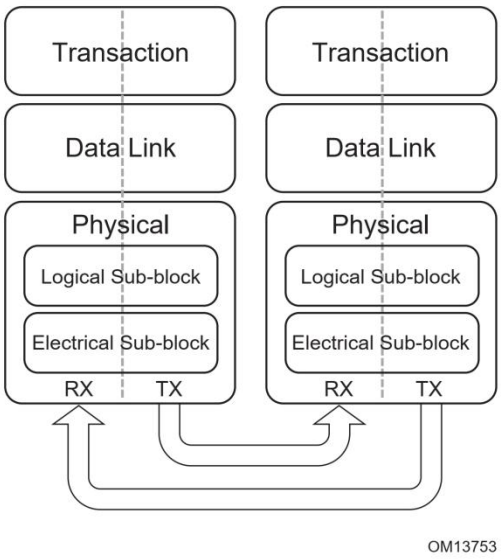


Figure 1-4: High-Level Layering Diagram

该分层定义的目的是促进读者对规范的理解。请注意，此分层并不意味着特定的 PCI Express 实现。

PCI Express 使用数据包在组件之间传递信息。数据包在事务和数据链路层中形成，以将信息从发送组件传送到接收组件。当传输的数据包流经其他层时，它们会以处理这些层的数据包所需的附加信息进行扩展。 在接收端发生相反的过程，数据包从其物理层表示转换为数据链路层表示，最后（对于事务层数据包）转换为可以由接收设备的事务层处理的形式。Figure 1-5 显示了通过层的事务级别数据包信息的概念性流程。

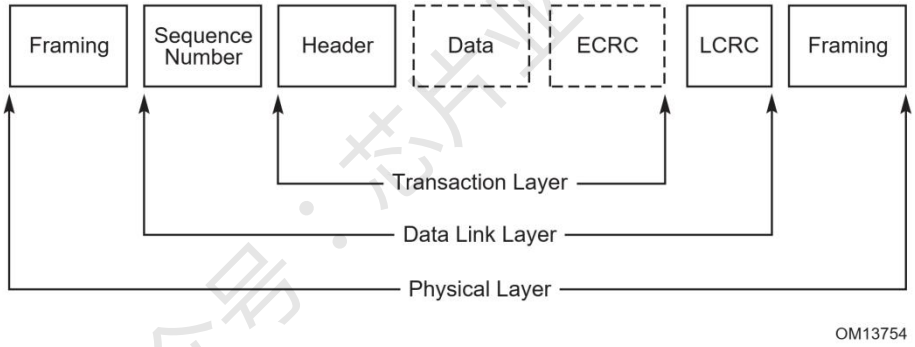


Figure 1-5: Packet Flow Through the Layers

注意，出于链路管理的目的，在两个数据链接层之间支持一种更简单的分组通信形式。

1.5.1 Transaction Layer

事务层的主要责任是事务层数据包（TLP）的组装和拆卸。TLP 用于通信事务，例如读和写以及某些类型的事件。事务层还负责管理 TLP 的基于信用的流控制功能。

每个需要响应包的请求包都实现为 split transaction。每个数据包都有一个唯一的标识符，该标识符使响应数据包可以定位到正确的请求者。数据包格式支持不同形式的寻址，具体取决于事务的类型（Memory、I/O、Configuration 和 Message）。Packet 还可以具有 No Snoop，Relaxed Ordering 和 ID-Based Ordering (IDO)等属性。

事务层支持四个地址空间：它包括三个 PCI 地址空间（memory、I/O 和 configuration），并添加了 Message Space。该规范使用 Message Space 来支持所有先前的边带信号，例如中断，电源管理请求等。您可以将 PCI Express 消息事务视为“虚拟线”，因为它们的作用是消除当前平台实现中使用的各种边带信号。

1.5.2 Data Link Layer

数据链路层处于 PCIe 协议栈的中间层，充当事务层和物理层之间的中间阶段。数据链路层的主要职责包括链路管理和数据完整性，包括错误检测和错误纠正。

数据链路层的传输端接受由事务层组装的 TLP，计算并应用数据保护代码和 TLP 序列号，并将它们提交给物理层，以在链路上传输。接收数据链路层负责检查接收到的 TLP 的完整性，并将其提交给事务层进行进一步处理。在检测到 TLP 错误后，此层负责请求重发 TLP，直到正确接收到信息或确定链路已失败为止。

数据链路层还生成并使用用于链路管理功能的数据包。为了将这些数据包与事务层（TLP）使用的数据包区分开，当引用数据链路层生成和使用的数据包时，将使用术语“数据链路层数据包（DLLP）”。

1.5.3 Physical Layer

物理层包括用于接口操作的所有电路，包括驱动器和输入缓冲器，并行到串行和串行到并行转换，PLL 和阻抗匹配电路。它还包括与接口初始化和维护有关的逻辑功能。物理层以实现特定的格式与数据链路层交换信息。该层负责将从数据链路层接收的信息转换为适当的序列化格式，并以与连接到链路另一端的设备兼容的频率和宽度在 PCI Express 链路上传输该信息。

PCI Express 架构具有“钩子”特性，可以通过速度升级和高级编码技术来支持将来的性能增强。未来的速度，编码技术或媒体可能只会影响物理层的定义。

1.5.4 Layer Functions and Services

1.5.4.1 Transaction Layer Services

事务层在生成和接收 TLP 的过程中，与链路另一侧的事务层交换流控制信息。它还负责支持软件和硬件启动的电源管理。

初始化和配置功能要求事务层执行以下操作：

- 存储由处理器或管理设备生成的链路配置信息。
- 通过物理层关于宽度和工作频率的硬件协商而生成的 Store Link 功能。

事务层的数据包生成和处理服务要求它：

- 从设备核心请求生成 TLP。
- 将收到的请求 TLP 转换为对设备核心的请求。
- 将收到的完成数据包转换为可交付给核心的有效负载或状态信息。
- 检测不受支持的 TLP，并调用适当的机制来处理它们。
- 如果支持端到端数据完整性，请生成端到端数据完整性 CRC 并相应地更新 TLP header。

流控制服务：

- 事务层跟踪整个链路上 TLP 的流量控制信用。
- 使用数据链路层的传输服务将事务信用状态定期发送到对端事务层。
- 对端流控制信息用于限制本端 TLP 传输。

公众号·芯片业者相聚

排序规则：

- 符合 PCI/PCI-X 的生产者消费者排序模型。
- 额外支持 Relaxed Ordering 。
- 额外支持 ID-Based Ordering 。

电源管理服务：

- ACPI/PCI 电源管理。
- 硬件控制的自主电源管理可在全开机状态下最大程度地降低功耗。

Virtual Channels 和 Traffic Class：

- 提供虚通道机制和流量类别标识的组合以支持针对某些类别的应用程序的区分服务和 QoS 支持。
- 虚通道提供了一种在链路的给定公共物理资源上支持多个独立逻辑数据流的方法。从概念上讲，这涉及将不同的数据流复用到单个物理链路上。
- 流量类别是一个事务层数据包标签，它通过系统未经修改地进行端对端的传输。在系统内的每个服务点（例如，Switch），流量类别标签用于应用适当的服务策略。每个流量类别标签都定义一个唯一的排序域-不为包含不同流量类别标签的数据包提供排序保证。

1.5.4.2 Data Link Layer Services

数据链路层负责与链路另一侧的设备可靠地交换信息。

初始化和电源管理服务：

- 接受来自事务层的电源状态请求并传送到物理层。
- 将 active/reset/disconnected/power 等链路状态传达给事务层。

数据保护，错误检查和重试服务：

- CRC 生成。
- 存储 TLP 用于数据链路级别重试。
- 错误检查。
- TLP 应答和消息重试。
- 错误报告和记录。

1.5.4.3 Physical Layer Services

接口初始化，维护控制和状态跟踪：

- 复位、热插拔控制、状态信息。
- 互联电源管理。
- 链路宽度和通道映射。

公众号·芯片业者相关及

- 通道极性翻转。

符号和特殊有序集生成：

- 8b/10b 编解码。
- 嵌入式时钟调节和对齐。

符号传输和对齐：

- 传输电路。
- 接收电路。
- 接收侧的弹性缓冲器。
- 接受测的 Lane-to-Lane de-skew。

DFT 支持的功能：

- Compliance pattern 。
- Modified Compliance pattern 。

1.5.4.4 Inter-Layer Interfaces

1.5.4.4.1 Transaction/Data Link Interface

事务层到数据链路层提供：

- 要通过链路发送的数据。
  - 本地的 TLP 传输握手机制。
  - TLP 边界信息。
- 请求的链路电源状态。

数据链路层到事务层提供：

- 从链路接收的数据。
- 接收到的数据的组包信息。
- 链路实际的电源状态。
- 链路状态信息。

1.5.4.4.2 Data Link/Physical Interface

数据链路层到物理层提供：

- 要通过链路发送的数据。
  - 数据传输握手机制。

公众号：芯片业者柏承双

    - TLP 和 DLLP 边界信息。
- 请求的链路电源状态。

物理层到数据链路层提供：

- 从链路接收到的数据。
- TLP 和 DLLP 数据的组包信息。
- 物理层的错误检测信息。
- 实际链路状态。
- 连接状态信息。

公众号·芯片业者柏承双

2 Transaction Layer Specification

2.1 Transaction Layer Overview

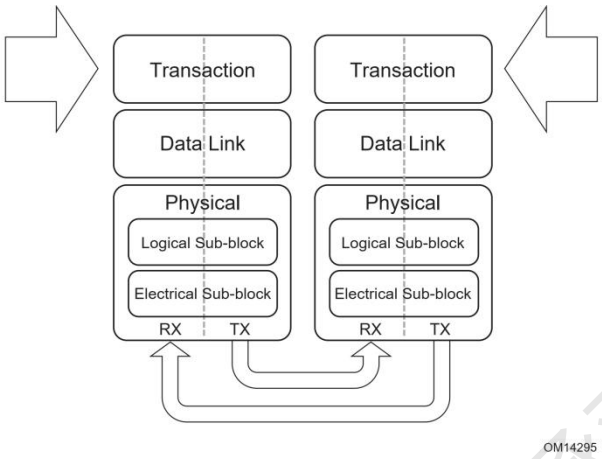


Figure 2-1: Layering Diagram Highlighting the Transaction Layer

从上层逻辑看，事务层的关键点是：

- 流水线式的完整的 split-transaction 协议。
- 事务层数据包（TLP）的排序和处理。
- 基于信用的流控制机制。
- 可选支持的数据中毒功能和端到端数据完整性检测功能。

事务层包含以下内容：

- TLP 构造和处理。
- 流控机制和虚通道管理。
- TLP 排序和管理。
  - PCI/PCI-X 兼容的排序。
  - TC 处理。

公众号·芯片业者相关双

2.1.1 Address Spaces, Transaction Types, and Usage

Table 2-1 Transaction Types for Different Address Spaces

Address Space	Transaction Types	Basic Usage
Memory	Read Write	Transfer data to/from a memory-mapped location
I/O	Read Write	Transfer data to/from an I/O-mapped location
Configuration	Read Write	Device Function configuration/setup
Message	Baseline (including Vendor-Defined)	From event signaling mechanism to general purpose messaging

2.1.1.1 Memory Transactions

包括：读请求/完成、写请求、原子操作请求/完成；分 32bit 地址格式和 64bit 地址格式。

某些内存事务可以带有 Process Address Space ID (PASID)的 PASID TLP Prefix。详细请参见 Section 6.20。

2.1.1.2 I/O Transactions

PCI Express 支持 I/O 空间，以与传统设备兼容。本规范的未来修订版可能会弃用 I/O 空间。I/O 事务包括：读请求/完成、写请求/完成。I/O 事务只有 32 bit 地址格式。

2.1.1.3 Configuration Transactions

配置请求包括：读请求/完成、写请求/完成。

2.1.1.4 Message Transactions

消息事务或简称为消息，用于支持设备之间事件的带内通信。

除了指定的消息外，PCI Express 还使用指定的消息代码为供应商定义的消息提供支持。特定的供应商定义的消息的定义不在本文档的范围之内。

该规范建立了一个标准框架，供应商可以在其中指定自己的供应商定义的消息，这些消息经定制以适应其平台的特定要求（请参阅 Section 2.2.8.5 和 Section 2.2.8.7）。

请注意，不能保证这些供应商定义的消息可与其他供应商的组件互操作。

公众号：芯片业者相承双

2.1.2 Packet Format Overview

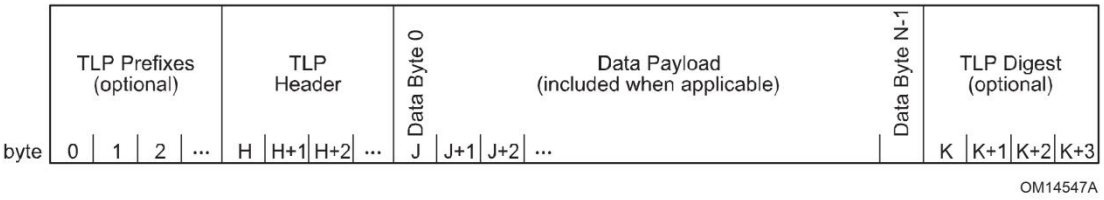


Figure 2-2: Serial View of a TLP

Figure 2-2 表示了在链路上串行传输的 TLP 的顺序，最左边一个字节最先发送或接收。

Figure 2-3 中表示的 TLP Prefix，TLP Header 和 TLP Digest 的位置，其中低编号的字节绘制在左侧，而 PCI 规范低编号的字节绘制在右侧。Header 的布局针对串行互连上的性能进行了优化，这是因为首先传输时间最紧迫的信息所驱动的。例如，在 TLP header 中，首先传输地址字段的最高有效字节，以便可以将其用于早期地址解码。

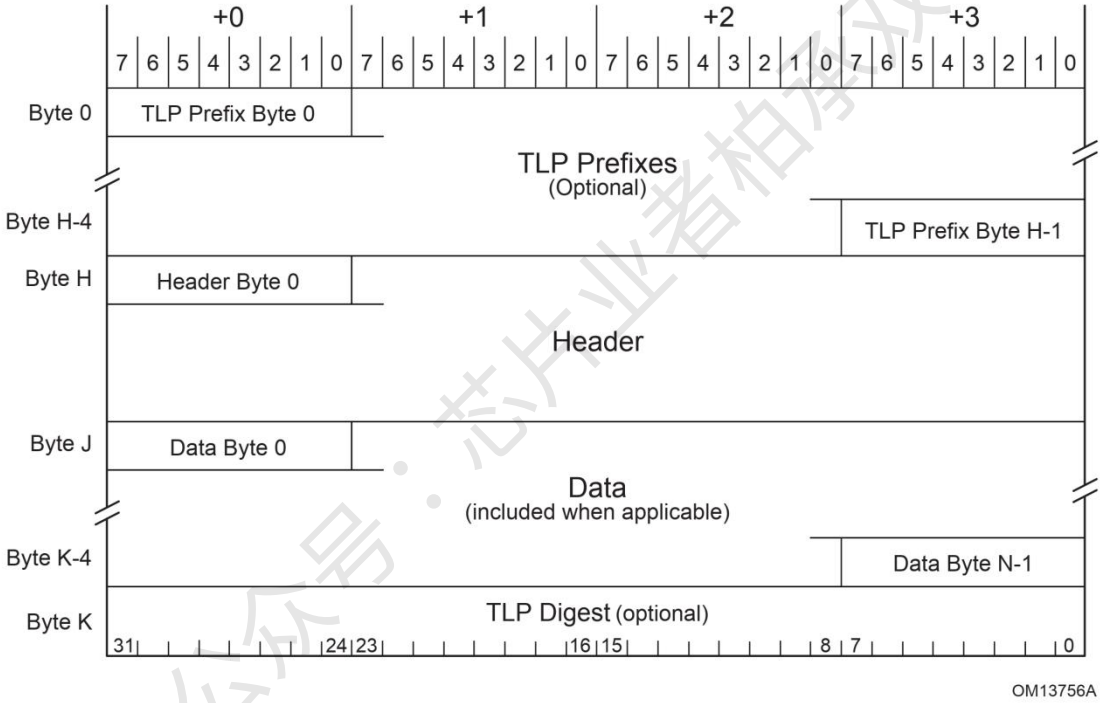


Figure 2-3: Generic TLP Format

TLP 内的有效载荷数据以最低寻址字节（Figure 2-3 所示）显示在左上方。 描述数据结构组织的详细布局（例如 Chapter 7 中描述的配置空间）保留了传统的 PCI 字节布局，并在右侧显示了最低的寻址字节。无论如何描述，所有字节在概念上都以递增的字节数顺序在链路上传输。

根据数据包的类型，该 Packet header 包含以下字段：

- Format of the packet
- Type of the packet
- Length for any associated data
- Transaction Descriptor, including：

公众号·芯片业者相濡以沫

- Transaction ID
- Attributes
- Traffic Class

- Address/routing information
- Byte Enables
- Message encoding
- Completion status

2.2 Transaction Layer Protocol - Packet Definition

四种事务类型：Memory、I/O、Configuration 和 Messages 。两种地址格式：32bit 和 64bit。

构成 TLP 时，所有标记为 Reserved 的字段（有时缩写为 R）都必须全为 0。接收者必须忽略此字段中的值，Switch 必须对其进行原封不动的转发。请注意，对于某些字段，既有指定值又有 Reserved 值–在这些情况下，按 Reserved 值的方式处理。

2.2.1 Common Packet Header Fields

所有 TLP 的 prefix 和 header 都包含以下定义字段，如 Figure 2-4 所示：

- Fmt[2:0] —— Format of TLP (see Table 2-2) – bits 7:5 of byte 0
- Type[4:0] —— Type of TLP – bits 4:0 of byte 0

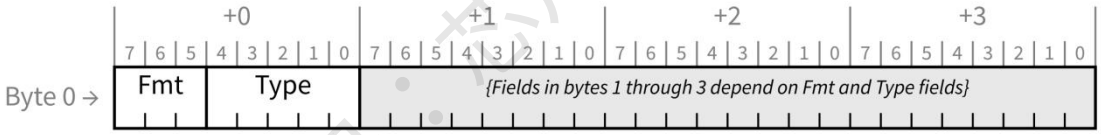


Figure 2-4 Fields Present in All TLPs

Fmt 字段还指示是否存在一个或多个 TLP Prefix，Type 字段指示相关的 TLP Prefix 的类型。

TLP header 的 Fmt 和 Type 字段确定了 TLP header 是 3DW 还是 4DW，还确定了 header 之后是否包含数据有效载荷。

TLP header 的 Fmt，Type，TD 和 Length 字段确定了非前缀 TLP 的大小所需的所有信息。Type 字段除了定义 TLP 的类型外，还决定了 Switch 如何路由 TLP。以下各节将详细讨论不同类型的 TLP。

- Fmt[2:0]和 Type[4:0]字段的值定义如 Table 2-3 。
  - 所有其他编码都是保留的（see Section 2.3）。
- TC[2:0] —— Traffic Class (see Section 2.2.6.6) – bits [6:4] of byte 1 。
- Lightweight Notification (LN) —— 此 1bit 表示内存请求是 LN Read 或 LN Write，或者完成是 LN Completion。
- TLP Hints (TH) —— 1bit 值表示在 TLP header 中存在 TPH（TLP Processing Hints）信息以及可选的 TPH TLP Prefix，– bit 0 of byte 1 (see Section 2.2.7.1) 。
- Attr[1:0] —— Attributes (see Section 2.2.6.3) – bits [5:4] of byte 2 。

公众号·芯片业者相承汉

- Attr[2] —— Attribute (see Section 2.2.6.3) – bit 2 of byte 1 。
- TD —— 1bit 值表示在 TLP 末尾以单个 DW 的形式存在 TLP 摘要(see Section 2.2.3) – bit 7 of byte 2 。
- Error Poisoned (EP) —— 表示该 TLP 是中毒的 TLP (see Section 2.7) – bit 6 of byte 2 。
- Length[9:0] —— 表示以 DW 为单位的 Data Payload 的长度（see Table 2-4 ）– byte 2[1:0]+byte 3[7:0] 。
  - TLP 数据必须自然对齐 4 字节，并以 4 字节双字（DW）为增量。
  - Cpl，CplLk 和一些 Message 保留此字段值。

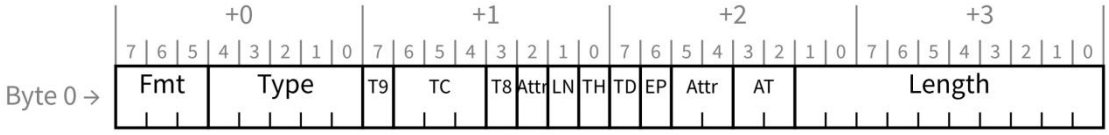


Figure 2-5 Fields Present in All TLP Headers

Table 2-2 Fmt[2:0] Field Values

Fmt[2:0]	Corresponding TLP Format
000b	3 DW header, no data
001b	4 DW header, no data
010b	3 DW header, with data
011b	4 DW header, with data
100b	TLP Prefix
	All encodings not shown above are Reserved (see Section 2.3).

公众号·芯片业者柏承双

Table 2-3 Fmt[2:0] and Type[4:0] Field Encodings

TLP Type	Fmt [2:0][3] (b)	Type [4:0] (b)	Description
MRd	000 001	0 0000	Memory Read Request
MRdLk	000 001	0 0001	Memory Read Request-Locked

TLP Type	Fmt [2:0] (b)	Type [4:0] (b)	Description
MWr	010 011	0 0000	Memory Write Request
IORd	000	0 0010	I/O Read Request
IOWr	010	0 0010	I/O Write Request
CfgRd0	000	0 0100	Configuration Read Type 0
CfgWr0	010	0 0100	Configuration Write Type 0
CfgRd1	000	0 0101	Configuration Read Type 1
CfgWr1	010	0 0101	Configuration Write Type 1
TCfgRd	000	1 1011	Deprecated TLP Type[4]
TCfgWr	010	1 1011	Deprecated TLP Type[5]
Msg	001	1 $0r_2r_1r_0$	Message Request - The sub-field r[2:0] specifies the Message routing mechanism (see Table 2-17).
MsgD	011	1 $0r_2r_1r_0$	Message Request with data payload - The sub-field r[2:0] specifies the Message routing mechanism (see Table 2-17).
Cpl	000	0 1010	Completion without Data - Used for I/O and Configuration Write Completions with any Completion Status. Also used for AtomicOp Completions and Read Completions (I/O, Configuration, or Memory) with Completion Status other than Successful Completion.
CplD	010	0 1010	Completion with Data - Used for Memory, I/O, and Configuration Read Completions. Also used for AtomicOp Completions.
CplLk	000	0 1011	Completion for Locked Memory Read without Data - Used only in error case.
CplDLk	010	0 1011	Completion for Locked Memory Read - Otherwise like CplD.
FetchAdd	010 011	0 1100	Fetch and Add AtomicOp Request
Swap	010 011	0 1101	Unconditional Swap AtomicOp Request
CAS	010 011	0 1110	Compare and Swap AtomicOp Request
LPrfx	100	0 $L_3L_2L_1L_0$	Local TLP Prefix - The sub-field L[3:0] specifies the Local TLP Prefix type (see Table 2-36).
EPrfx	100	1 $E_3E_2E_1E_0$	End-End TLP Prefix - The sub-field E[3:0] specifies the End-End TLP Prefix type (see Table 2-37).
			All encodings not shown above are Reserved (see Section 2.3).

公众号：芯片业者相承双

Table 2-4 Length[9:0] Field Encoding

Length[9:0]	Corresponding TLP Data Payload Size
00 0000 0001b	1 DW
00 0000 0010b	2 DW
…	…
11 1111 1111b	1023 DW
00 0000 0000b	1024 DW

注释 3：两个 Fmt 值的意思是其中一个使用 32bit 地址格式，另一个使用 64bit 地址格式。

注释 4,5：不推荐使用的 TLP 类型：以前用于受信任的配置空间（Trusted Configuration Space ，TCS），此规范不再支持。如果接收方未实现 TCS，则接收方必须将此类请求视为格式错误的数据包。

2.2.2 TLPs with Data Payloads - Rules

- Length 字段用于指定有多少个 DW。
- 对于所有 Message，Length[9:0]为 Reserved 值，除了那些显式引用数据长度的 Message 。
  - 参阅 Section 2.2.8 的消息编码表获取更多信息。
- 发送器不允许发送的 TLP 中的 Length 字段的值超过 Device Control 寄存器的 Max_Payload_Size 字段定义的值的 TLP（see Section 7.5.3.4）。
  - 对于 ARI 设备，Max_Payload_Size 仅由 Function 0 中设置的值确定。其他 Function 中的 Max_Payload_Size 将被忽略。
  - 对所有 Function 上的 Max_Payload_Size 设置值都相同的 non-ARI Multi-Function Device（MFD）的 Upstream Port，传输的 TLP 的数据有效载荷不得超过通用的 Max_Payload_Size 设置值。
  - 对所有 Function 上的 Max_Payload_Size 设置值不相同的 non-ARI multi-Function 设备的 Upstream Port，传输的 TLP 的数据有效载荷不得超过 Max_Payload_Size 设置值，具体根据实现定义。
    - 建议发送器的实现使用生成事务的 Function 中的 Max_Payload_Size 设置值，或者使用所有 Function 中最小的 Max_Payload_Size 设置值。
    - 除非软件知道特定的实现，否则软件不应将不同 Function 中的 Max_Payload_Size 设置为不同的值。
  - Note：Max_Payload_Size 仅适用于具有数据有效负载的 TLP。Memory Read Request 的长度不受 Max_Payload_Size 的限制。Memory Read Request 的大小由 Length 字段控制。
- 接收器接收到的 TLP 中的 Length 字段的值不得超过 Device Control 寄存器的 Max_Payload_Size 字段定义的值（see Section 7.5.3.4）。
  - 接收器必须检查是否违反此规则。如果接收器确定 TLP 违反此规则，则该 TLP 是一个 Malformed TLP。
    - 这是报告的与接收端口相关的错误（请参见 Section 6.2）。
  - 对于 ARI 设备，Max_Payload_Size 仅由 Function 0 中设置的值确定。其他 Function 中的 Max_Payload_Size 将被忽略。

公众号·芯片业者相承双

- 对所有 Function 上的 Max_Payload_Size 设置值都相同的 non-ARI multi-Function 设备的 Upstream Port，接收器将按通用的 Max_Payload_Size 设置检查 TLP 的数据有效载荷大小。
- 对所有 Function 上的 Max_Payload_Size 设置值不相同的 non-ARI multi-Function 设备的 Upstream Port，接收器如何检查 Max_Payload_Size 设置值是根据实现定义的。
  - 建议接收器实现使用事务目标的 Function 中的 Max_Payload_Size 设置，或者使用所有 Function 中最大的 Max_Payload_Size 设置。
  - 除非软件知道特定的实现，否则软件不应将不同 Function 中的 Max_Payload_Size 设置为不同的值。

- 对于包含数据的 TLP，Length 字段中的值和 TLP 中包含的实际数据量必须匹配。
  - 接收器必须检查是否违反此规则。 如果接收器确定 TLP 违反此规则，则该 TLP 是一个 Malformed TLP。
    - 这是报告的与接收端口相关的错误（请参见 Section 6.2）。
- Length 字段中的值仅适用于数据 —— Length 字段中不包括 TLP Digest 。
- 当除 AtomicOp Request 或 AtomicOp Completion 之外的其他 TLP 中包含数据有效载荷时，header 后的数据的第一个字节对应于最接近零的字节地址，并且后续字节按递增的字节地址顺序排列。
  - 示例：向地址 100h 写入 16 字节，header 后的第一个字节将是要写入地址为 100h 的字节，第二个字节将被写入地址 101h，依此类推，最后一个字节被写入地址 10Fh。
- 必须对 AtomicOp Request 和 AtomicOp Completion 中的数据有效载荷进行格式化，以使 TLP header 之后的数据的第一个字节是第一个数据的最低有效字节，而随后的数据字节的重要性严格增加。对于 CAS Request，第二个数据紧跟在第一个数据之后，并且必须采用相同的格式。
  - AtomicOp Completer 读取和写入目标地址的数据的字节序格式是特定于实现的，并且允许为 Completer 指定适合目标内存的任何格式（例如，little endian 或 big endian）。AtomicOp Completer 的字节序格式功能的报告和控制不在本规范的范围之内。
  - Little endian 的例子：对于目标内存为 Little endian 的 64bit 的 Swap Request 的目标地址 100h，header 后的第一个字节写入地址 100h，第二个字节写入地址 101h，并且依此类推，最后一个字节写入位置 107h。请注意，在执行写操作之前，Completer 首先读取目标存储器地址，以便它可以在完成中返回原始值。Completion 中与数据对应的字节地址与 Request 中的相同。
  - Big endian 的例子：对于目标内存为 Big endian 的 64bit 的 Swap Request 的目标地址 100h，header 后的第一个字节写入地址 107h，第二个字节写入地址 106h，依此类推，将最后一个字节写入地址 100h。请注意，在执行写操作之前，Completer 首先读取目标存储器地址，以便它可以在完成中返回原始值。Completion 与数据对应的字节地址与 Request 中的相同。
  - Figure 2-6 显示了针对 64bit 的 FetchAdd 的 Completer 目标内存访问的小端和大端示例。操作数和结果中的字节编号为 0-7，字节 0 为最低有效位，字节 7 为最高有效位。在每种情况下，Completer 都使用适当的字节序格式获取目标内存操作数。接下来，Completer 中的 AtomicOp 计算逻辑使用原始目标内存值和 FetchAdd 请求中的“add 值执行 FetchAdd 操作。最后，Completer 使用与读取相同的字节序格式将 FetchAdd 结果存储回目标内存。

公众号：芯片业者相承双

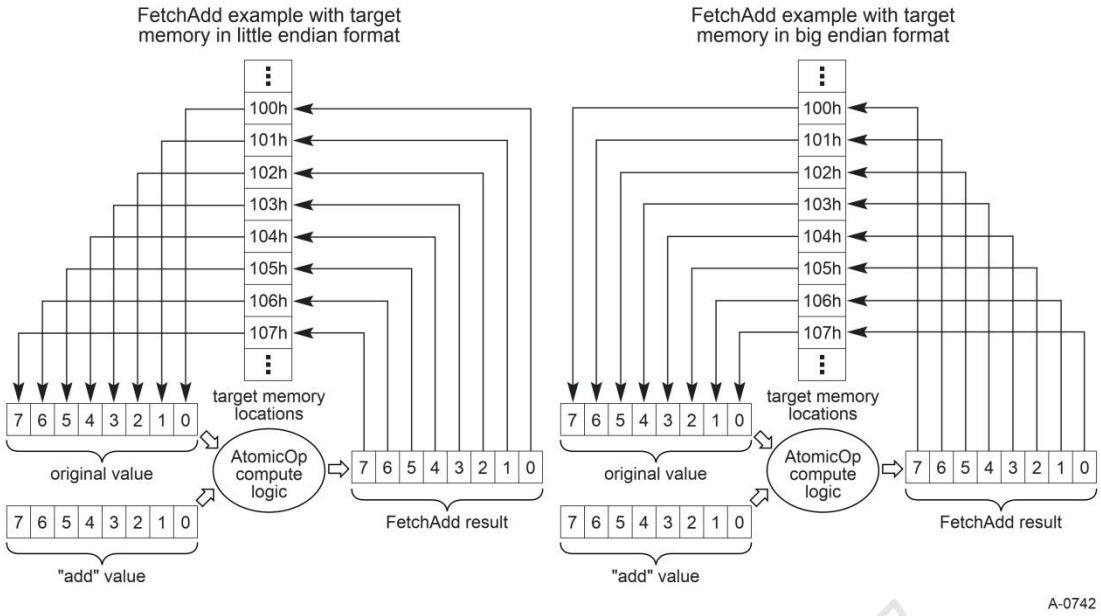


Figure 2-6 Examples of Completer Target Memory Access for FetchAdd

IMPLEMENTATION NOTE

Endian Format Support by RC AtomicOp Completers

允许 AtomicOp Completer 使用其选择的字节序格式访问目标内存的一个关键原因是，使以 AtomicOps 为目标内存的 PCI Express 设备可以与使用原子操作指令（或指令序列）的主机软件进行互操作。某些主机环境对原子操作的字节序格式支持有限，并且通过支持“正确”字节序格式，RC AtomicOp Completer 可以显着提高互操作性。

对于在仅支持 Little Endian 字节序处理器的平台上具有 AtomicOp Completer 功能的 RC，RC AtomicOp Completer 支持除 Little Endian 以外的任何字节序格式几乎没有好处。对于在支持双字节序处理器的平台上具有 AtomicOp Completer 功能的 RC，在支持 Little Endian 字节序和 Big Endian 字节序格式以及可能为主机内存的不同区域配置字节序格式方面可能会有好处。

没有 PCI Express 设备要求 RC AtomicOp Completer 支持主机处理器的“本机”字节序格式（如果有的话），这样做也不会有任何好处。 例如，某些处理器可以使用 load-link/store-conditional 或类似的指令序列以非本机字节序格式进行原子操作，因此不需要 RC AtomicOp Completer 支持其他字节序格式。

IMPLEMENTATION NOTE

Maintaining Alignment in Data Payloads

Section 2.3.1.1 讨论了有关形成某些自然地址边界的读取完成的规则。通过在 Write Request 添加遵守相似的地址边界的信息，可以显著提高内存写入性能。具体而言，形成 Write Request 以遵守 64 或 128 字节的自然地址边界将有助于提高系统性能。这是一个再常见不过的经验法则，这样做会降低硬件处理字节不对齐的复杂度，因而提高系统性能。

2.2.3 TLP Digest Rules

- 对于任何 TLP，TD 字段中的值为 1b 表示存在 TLP Digest 字段，该字段包括 TLP 末尾的 ECRC 值。
  - TD 字段值与观察到的大小不对应（考虑到数据有效载荷，如果存在的话）的 TLP 是一个 Malformed 的 TLP。
    - 这是报告的与接收端口相关的错误（请参见 Section 6.2）。

公众号：芯片业者相承双

- 如果中间设备或最终 PCI Express 接收器不支持 ECRC 检查，则接收器必须忽略 TLP Digest。
  - 如果 TLP 的接收方支持 ECRC 检查，则接收方会根据 Section 2.7.1 中的规则将 TLP Digest 字段中的值解释为 ECRC 值。

2.2.4 Routing and Addressing Rules

TLP 有三种路由机制：地址路由、ID 路由和隐式路由。本节定义了地址路由和 ID 路由机制的规则。隐式路由仅与消息请求一起使用，并在 Section 2.2.8 中介绍。

2.2.4.1 Address Based Routing Rules

- 地址路由用于 Memory 请求和 I/O 请求。
- 有两种地址格式：32bit 地址的 3DW header（see Figure 2-8）和 64bit 地址的 4DW header（see Figure 2-7）。

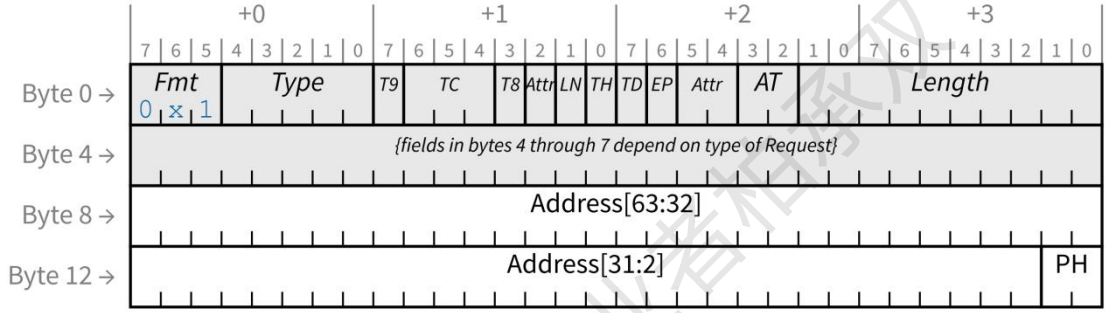


Figure 2-7 64-bit Address Routing

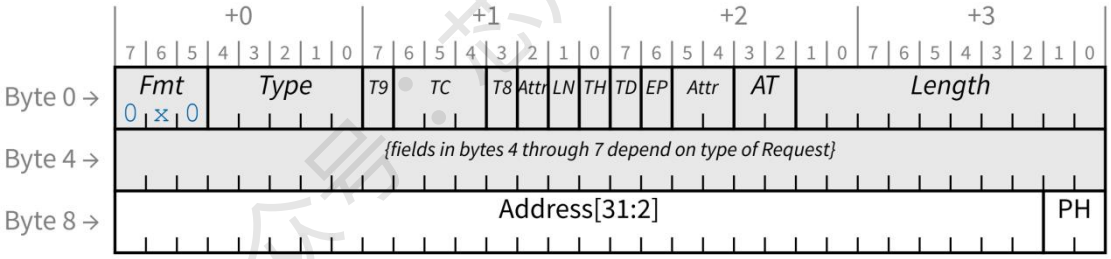


Figure 2-8 32-bit Address Routing

- 对于 Memory Read、Memory Write 和 AtomicOp Request，Address Type（AT）字段的编码如 Table 10-1 所示。对于所有其他请求，除非另有明确说明，否则 AT 字段为保留字段。LN Read 和 LN Write 对 AT 字段有特殊的要求，参见 Section 6.21.5 节。

公众号·芯片业者相关

Table 10-1 Address Type (AT) Field Encodings

AT[1:0] Coding	Mnemonic	Meaning
00b	Untranslated	A TA may treat the address as either virtual or physical.
01b	Translation Request	The TA will return the translation of the address contained in the address field of the request as a read completion. This value only has meaning for an explicit Translation Request (see Section 10.2.2). The TA will signal an Unsupported Request (UR) if it receives a TLP with the AT field set to 01b in a Memory Request other than Memory Read.
10b	Translated	The address in the transaction has been translated by an ATC. If the Function associated with the SourceID is allowed to present physical addresses to the system memory, then the TA might not translate this address. If the Function is not allowed to present physical addresses, then the TA may treat this as an UR.
11b	Reserved	The TA will signal an Unsupported Request (UR) if it receives a Memory Request TLP with the AT field set to 11b.

- 如果 TH 字段设置为 1，则 PH 字段编码如 Table 2-15 所示。如果 TH 字段清 0，则 PH 字段为保留值。

Table 2-15 Processing Hint Encoding

PH[1:0] (b)	Processing Hint	Description
00	Bi-directional data structure	Indicates frequent read and/or write access to data by Host and device
01	Requester	Indicates frequent read and/or write access to data by device
10	Target	Indicates frequent read and/or write access to data by Host
11	Target with Priority	Indicates frequent read and/or write access by Host and indicates high temporal locality for accessed data

- TLP header 中地址字段的分布如 Table 2-5 所示。

Table 2-5 Address Field Mapping

Address Bits	32-bit Addressing	64-bit Addressing
63:56	Not Applicable	Bits 7:0 of Byte 8
55:48	Not Applicable	Bits 7:0 of Byte 9
47:40	Not Applicable	Bits 7:0 of Byte 10
39:32	Not Applicable	Bits 7:0 of Byte 11
31:24	Bits 7:0 of Byte 8	Bits 7:0 of Byte 12
23:16	Bits 7:0 of Byte 9	Bits 7:0 of Byte 13
15:8	Bits 7:0 of Byte 10	Bits 7:0 of Byte 14
7:2	Bits 7:2 of Byte 11	Bits 7:2 of Byte 15

- Memory Read、Memory Write 和 AtomicOp Request 可以使用 3DW header，也可以使用 4DW header。
  - 对于低于 4 GB 的地址，请求者必须使用 32 位格式。如果接收到地址低于 4 GB 的 64 位格式请求（即地址的高 32 位全为 0），则未指定接收器的行为。
- I/O Read Request 和 I/O Write Request 只能用 32bit 地址格式。
- 所有设备都必须解码 header 中的所有地址字段-不允许使用别名。

公众号·芯片业者相承及

IMPLEMENTATION NOTE

Prevention of Address Aliasing

为了软件能正确操作，即使系统硬件架构师或设计人员知道丢 64 位地址格式在系统中有实际意义，也需要进行全地址解码。

2.2.4.2 ID Based Routing Rules

- 配置请求、基于 ID 路由的消息以及 Completion TLP 会使用 ID 路由。该规范定义的使用 ID 路由的消息如 Table F-1 所示。允许其他规范定义其他 ID 路由消息（如 ATS）。
- ID 路由使用总线号，设备号和功能号来指定 TLP 的目的地：
  - 对于 non-ARI Routing ID，总线号、设备号和 3-Bit 功能号到 TLP header 的映射如 Table 2-6，Figure 2-9 和 Figure 2-11 所示。
  - 对于 ARI Routing ID，总线号和 8-Bit 的功能号到 TLP header 的映射如 Table 2-7，Figure 2-10 和 Figure 2-12 所示。
- 规范指定了两种 ID 路由格式，一种用于 4 DW 头（Figure 2-9 和 Figure 2-10），一种用于 3 DW 头（Figure 2-12 和 Figure 2-10）。
  - 参阅 Figure 2-5 获取 TLP Header 的各字段分布。

Table 2-6 Header Field Locations for non-ARI ID Routing

Field	Header Location
Bus Number[7:0]	Bits 7:0 of Byte 8
Device Number[4:0]	Bits 7:3 of Byte 9
Function Number[2:0]	Bits 2:0 of Byte 9

Table 2-7 Header Field Locations for ARI ID Routing

Field	Header Location
Bus Number[7:0]	Bits 7:0 of Byte 8
Function Number[7:0]	Bits 7:0 of Byte 9

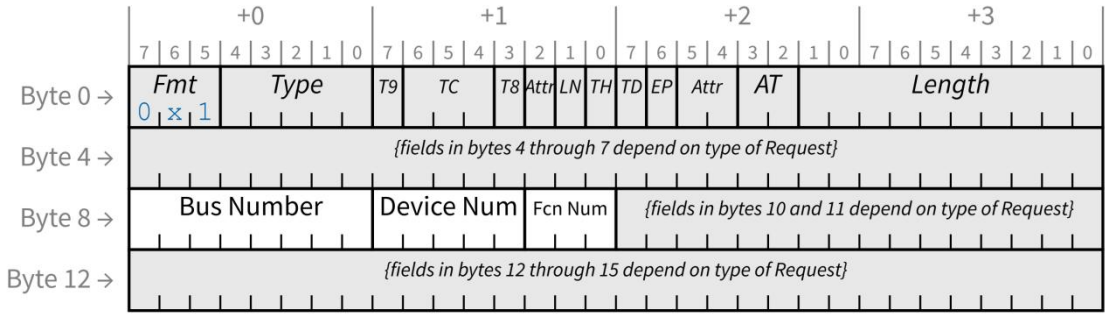


Figure 2-9 Non-ARI ID Routing with 4 DW Header

公众号·芯片业者相承双

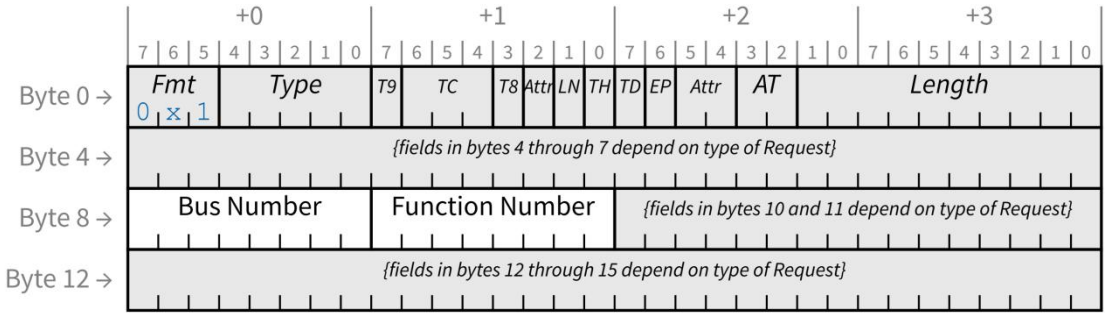


Figure 2-10 ARI ID Routing with 4 DW Header

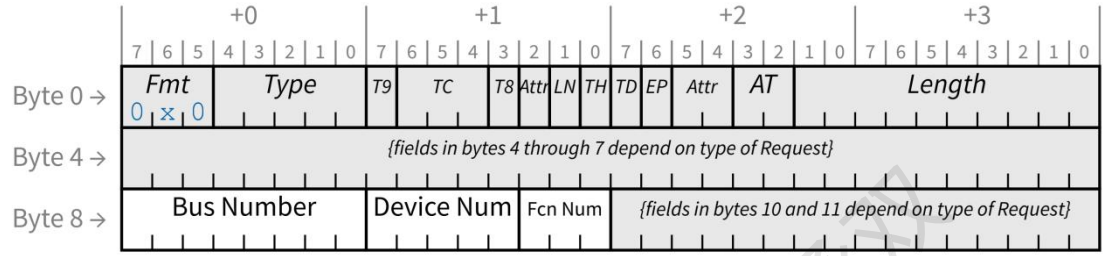


Figure 2-11 Non-ARI ID Routing with 3 DW Header

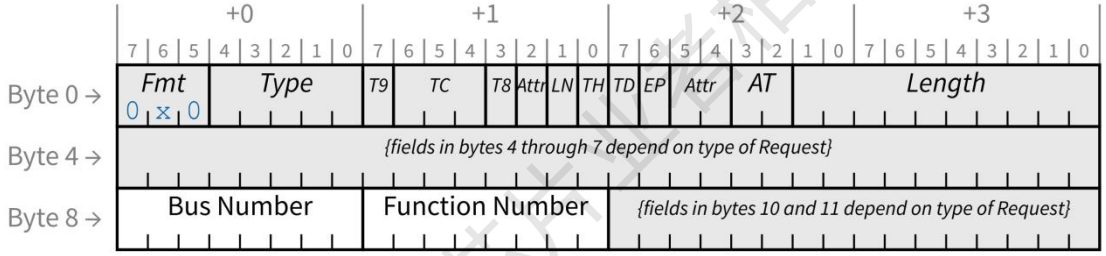


Figure 2-12 ARI ID Routing with 3 DW Header

2.2.5 First/Last DW Byte Enables Rules

Byte Enable 包含在 Memory、I/O 和 Configuration Request 中。本节定义了相应的规则。Byte Enable 位于 header 的 byte 7（see Figure 2-13）。对于 TH 字段值为 1 的 Memory Read Request，Byte Enable 字段被重新指定为携带 ST [7:0]字段（有关详细信息，请参见 Section 2.2.7.1），并且如下隐含的定义了 Byte Enable 的值。Memory Read Request 中的 TH 字段在这种情况下必须设置为 1：当看起来像是可以完成这些请求（就好像所请求数据的所有字节都使能一样）。

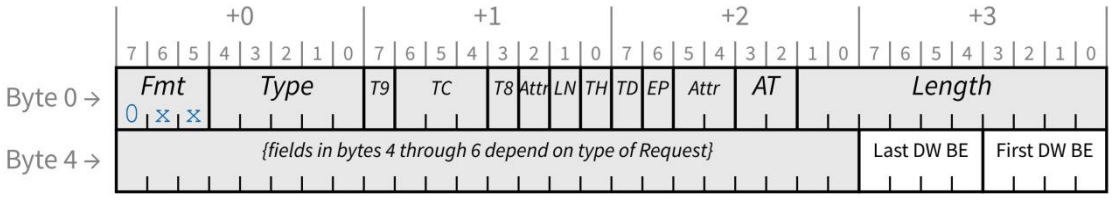


Figure 2-13 Location of Byte Enables in TLP Header

- 对于 TH 字段置 1 的 Memory Read Request，Byte Enable 隐含以下值。有关其他要求，请参见 Section 2.2.7。
  - 如果此请求的 Length 字段指示此请求的数据为 1 个 DW 的长度，则 First DW Byte Enable 的值隐含为 1111b，Last DW Byte Enable 的值隐含为 0000b。

- 如果此请求的 Length 字段指示此请求的数据大于 1 个 DW 的长度，First DW Byte Enable 和 Last DW Byte Enable 的值隐含为 1111b。

IMPLEMENTATION NOTE

Read Request with TPH to Non-Prefetchable Space

仅在可以确保完成此类读取不会产生不良副作用的情况下，才应发出将 TH 位置 1 的 Memory Read Request，并且用于访问 Non-Prefetchable Memory Space。即使某些 BAR 映射了具有读取副作用的某些地址，也请考虑 Section 7.5.2.1 中某些 BAR 可能会考虑把 Prefetchable 字段设置为 1。

- 1st DW BE [3:0]字段指明所请求的第一个（Length = 1 时为唯一一个）DW 的 Byte Enable 。
  - 如果 Request 的 Length 字段指示的长度大于 1 DW，则此字段不得等于 0000b。
- Last DW BE [3:0]字段指明请求的最后一个 DW 的 Byte Enable 。
  - 如果 Request 的 Length 字段指示的长度等于 1 DW，则此字段必须等于 0000b。
  - 如果 Request 的 Length 字段指示的长度大于 1 DW，则此字段不得等于 0000b。
- 对于 Byte Enable 字段的每个比特：
  - 如果值为 0b 表示不得写入相应的数据字节，或者如果 non-prefetchable，则不得读取。
  - 如果值为 1b 表示必须在写入或读取相应的数据字节。
- 对于长度为 1 DW 的所有请求类型，1st DW BE [3:0]字段中允许使用非连续的 Byte Enable 。
  - 非连续的 Byte Enable 指的是 1010b，0101b，1001b，1011b，1101b 这种形式。
- 对于长度为 2 DW 的 QW 对齐的 Memory Request，1st DW BE [3:0]和 Last DW BE [3:0]中均允许非连续的 Byte Enable。
- 长度为 2 DW 的所有非 QW 对齐的 Memory Request 和长度为 3 DW 或更大的 Memory Request 必须仅启用与请求的第一个 DW 和最后一个 DW 之间的数据连续的字节。
  - 连续的 Byte Enable 的例子：

    1st DW BE: 1100b，Last DW BE: 0011b

    1st DW BE: 1000b，Last DW BE: 0111b
- Table 2-8 给出了 Byte Enable 字段的比特分布，它们在 Request header 中的位置以及所引用数据的相应字节之间的对应关系。

公众号·芯片业者相承双

Table 2-8 Byte Enables Location and Correspondence

Byte Enables	Header Location	Affected Data Byte[7]
First DW BE[0]	Bit 0 of Byte 7	Byte 0
First DW BE[1]	Bit 1 of Byte 7	Byte 1
First DW BE[2]	Bit 2 of Byte 7	Byte 2
First DW BE[3]	Bit 3 of Byte 7	Byte 3
Last DW BE[0]	Bit 4 of Byte 7	Byte N-4
Last DW BE[1]	Bit 5 of Byte 7	Byte N-3
Last DW BE[2]	Bit 6 of Byte 7	Byte N-2
Last DW BE[3]	Bit 7 of Byte 7	Byte N-1

- 允许长度为 1 DW 且未启用任何字节的 Write Request，除非特别说明，否者这对 Completer 没有影响。

IMPLEMENTATION NOTE

Zero-Length Write

在某些协议下，设备可能会使用 1 DW 的未启用字节的 Memory Write Request 或“zero-length Write”，以实现某种预期作用。LN 协议就是一个例子。参见 Section 6.21。

- 如果 1 DW 的 Read Request 指定未启用任何字节读取（1st DW BE [3:0] = 0000b），则相应的 Completion 必须指定 1 DW 的长度，并包括 1 DW 的数据有效载荷。
  - 未指定 Completion packet 中数据有效载荷的内容，可以是任何值。
- 对于违反本节中指定的 Byte Enable 规则的 TLP，未定义接收方/完成方行为。
- 接收器可以选择检查是否违反了本节中指定的 Byte Enable 规则。如果执行此类检查的接收器确定 TLP 违反一个或多个 Byte Enable 规则，则该 TLP 是一个 Malformed TLP。这些检查是独立可选实现的（请参见 Section 6.2.3.4）。
  - 如果要检查 Byte Enable 规则，则报告与接收端口相关的错误（请参见 Section 6.2）。

IMPLEMENTATION NOTE

Zero-Length Read

设备可以将 no bytes enabled 的 1 DW 的 Memory Read Request 或“zero-length Read ”用作刷新请求的一种。对于请求者，刷新语义允许设备确保先前发送的 Posted Write 已在其 PCI Express 目标位置完成。为了在所有情况下均有效，zero-length Read 的地址必须与要刷新的 Posted Write 针对同一设备。一种推荐的方法是使用与 Posted Write 相同的地址。

刷新语义具有广泛的应用，所有 Completer 必须实现与此语义相关联的功能。由于请求者可以在不理解 Completer 的特征的情况下使用刷新语义，因此 Completer 必须确保 zero-length Read 没有副作用。这实际上只是规则的一种特殊情况，即在 non-prefetchable 空间中，一定不能在 Completer 中读取未启用的字节。请注意，刷新仅适用于与 zero-length Read 相同的 Traffic Class 的业务。

公众号·芯片业者相承双

2.2.6 Transaction Descriptor

2.2.6.1 Overview

事务描述符（Transaction Descriptor）是一种在请求者和完成者之间传送 Transaction 的信息的机制。事务描述符由三个字段组成，see Figure 2-14：

- Transaction ID —— 指明识别 Transaction。
- Attributes —— 指明 Transaction 的特征。
- Traffic Class (TC) —— 指明 Transaction 的业务类型。

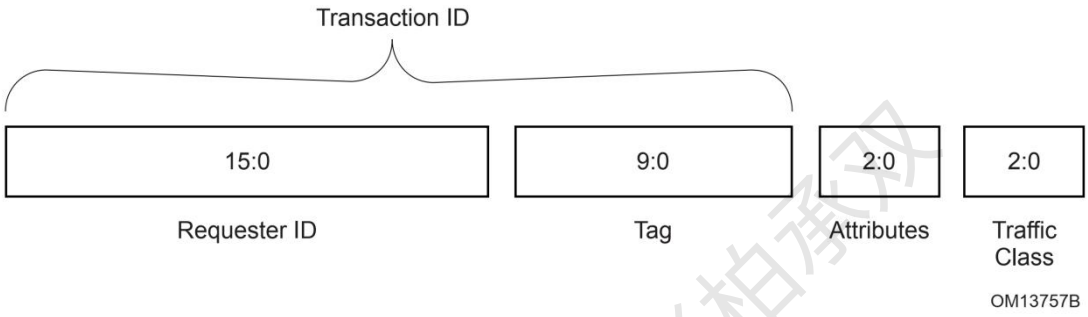


Figure 2-14 Transaction Descriptor

2.2.6.2 Transaction Descriptor – Transaction ID Field

Transaction ID 包括两个子字段：Requester ID 和 Tag，如 Figure 2-15 所示。

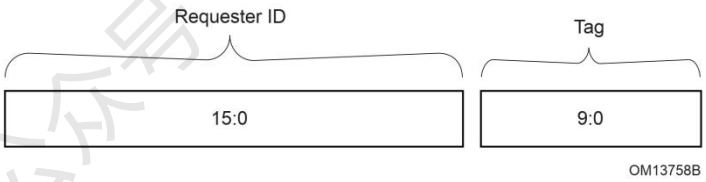


Figure 2-15 Transaction ID

在 PCIe 协议中，事务发起者(Requester)发送的 TLP 需要有身份 ID，这个就叫做 Transaction ID，而 Transaction ID 由两部分构成，Requester ID 和 Tag。其中，Requester ID 包含了 Bus/Device/Function 信息。而 Tag 则表示的是同一发起者(Requester)同时暂存 TLP 的数量（Tag 字段定义支持的 outstanding Non-Posted Request 的数量）。

注意：

1. 对于 Posted TLP 来说，Transaction ID 其实没有多大意义，因为 Requester 也不需要有反馈，但是设备还得要支持 Transaction ID 的。
2. 对于 Non-Posted TLP 来说，Transaction ID 就是必须的。不然在有 Cpl 包返回时，也不知道是哪个 Requester 的。

关于 Tag：

公众号·芯片业者柏承汉

1. Requester 发送 Non-Posted TLP 之后，在没有收到 Cpl 报文之前，对应的 Transaction ID 不能被释放。同一 Requester 发送的 Non-Posted TLP 的 Header 中, Requster ID 肯定是一样的。当一个 Requster 要发起多个 Non-Posted TLPs 时，Tag 字段就有了施展才华的时刻。

2. Tag 字段的大小决定了发送端可以暂存同一类型 TLP 的数量。PCIe 3.0 Spec 中显示，默认是 bit[4:0]，就是默认长度是 5，当 enable extend Tag bit 后，Tag 长度为 8。也就是说在 PCIe 3.0 中，每个 PCIe 设备的发送端最多只能暂存 256 个 TLPs，换句话说，对于同一发起者而言，此时 PCIe 链路上有其 256 个 TLPs 在传输。

3. PCIe 设备的 Function 设定中也可以扩展 Tag 字段。

4. Tag[9:8]的有效取值包括 01b，10b，11b。00b 表示 Requester 不支持 10-Bit Tag，如下所述。因而，10-bit Tag 最大暂存 TLP 数目由原来的 256 扩大到了 768。

[PCIe-4.0]中引入的 10-Bit Tag Capability 将总的 Tag 字段从 8-Bit 增加到 10-Bit。其中 Tag [8]（T8）和 Tag [9]（T9）与 TLP Header 中的其他 Tag [7:0]位不相邻。在本规范的先前版本中，保留了另外两个比特。

- Tag [9:0]是每个请求者生成的 10-Bit 字段，并且对于该请求者的所有需要一个 Completion 的 outstanding Request ，它必须是唯一的。不支持 10-Bit Tag Requester Capability 的请求者必须将 Tag [9:8]设置为 00b。
  - 支持 16.0 GT/s 或更高数据速率的 Function（包括 Switch 中的 Function）必须支持 10-Bit Tag Completer Capability。如果某个 Function 支持 10-Bit Tag Completer Capability，则可以选择支持 10-Bit Tag Requester Capability。请参阅 Section 7.5.3.15 和本节后面的“Considerations for Implementing 10-Bit Tag Capabilities”的实现说明。
  - 如果 RC 包含支持 10-Bit Tag Completer Capability 的元件，则 RC 必须正确处理所有作为 PCIe 请求者目标的寄存器和内存区域的 10-Bit Tag Request；例如，RCiEP 中的 DMA 请求或 MMIO 区域作为目标的主机内存。
    - 每个支持 10-Bit Tag Completer Capability 的 RP 必须由其 Ingress Port 处理收到的此类请求。
    - 每个支持 10-Bit Tag Completer Capability 的 RCiEP 必须处理来自内部路径（包括来自 RP 的请求）的此类请求。
  - 如果 RC 包含的 RCiEP 支持 10-Bit Tag Requester Capability，则 RC 必须由作为这些 RCiEP 目标的所有寄存器和存储区正确处理来自这些 RCiEP 的 10-Bit Tag Request。例如，RCiEP 中的 DMA 请求或 MMIO 区域作为目标的主机内存。
  - Receiver/Completer 必须正确处理 8-bit Tag 值，而不管其 Extended Tag Field Enable 字段的设置如何（Section 7.5.3.4）。关于桥处理扩展标签（extended tag）的详细信息，请参阅 *PCI Express to PCI/PCI-X Bridge Specification*。
  - 支持 10-Bit Tag Completer Capability 的 Receiver/Completer 必须正确处理 10-Bit Tag 值，而不管 10-Bit Tag Requester Enable 字段如何设置。参见 Section 7.5.3.16。
  - 10-Bit Tag Capability 不是为 PCI Express to PCI/PCI-X Bridge 设计的，它们不得支持 10-Bit Tag Requester Capability 或 10-Bit Tag Completer Capability。
  - 如果 10-Bit Tag Requester Enable = 0 且 Extended Tag Field Enable = 0，则每个 Function 的最大 outstanding Request 的数目应限制为 32，并且仅使用 Tag 字段的低 5 比特，其余高 5 比特必须为 00000b。
  - 如果 10-Bit Tag Requester Enable = 0 且 Extended Tag Field Enable = 1，则最大值增加到 256，并且仅使用 Tag 字段的低 8 比特，其余的高 2 比特为 00b 。
  - 如果 10-Bit Tag Requester Enable = 1，则针对单个 Completer 的最大限制提高到 768。在向其认为合适的 Completer 发送 10-Bit Tag Request 时，允许 Requester 使用整个 10-Bit Tag 字段，尽管仍然允许 Requester 将较小 Tag 的请求发送给其他 Completer。以下内容适用于 10-Bit Tag Requester Enable 字段被置 1 的具有 10-Bit Tag capable 的 Requester。
    - 如果某个 Endpoint 支持将请求发送到其他 Endpoint（而不是主机内存），则该 Endpoint 不得将 10-Bit Tag Request 发送到另一个给定的 Endpoint，除非有特定于实现的机制能确定该 Endpoint 支持 10-Bit Tag Completer Capability。

公众号：芯片业者相关权

对于某些实现，完全不向其他 Endpoint 发送 10-Bit Tag Request 可能是可以接受的。更复杂的机制不在本规范的范围之内。

- ◆ 如果 PIO Requester 支持 10-Bit Tag Requester Capability，则该 Requester 如何确定何时使用 10-Bit Tag 与较小 Tag 的机制超出了本规范的范围。
- ◆ 对于 10-Bit Tag，有效的 Tag [9:8]值为 01b，10b 或 11b。Tag [9:8]等于 00b 的 10-Bit Tag 值无效，并且请求者不能生成这样的请求。这使 Requester 可以判断接收到的应该具有 10-Bit Tag 的 Completion 是否包含无效标记，这通常是由于 Completer 不支持 10-Bit Tag Completer Capability 引起的。
- ◆ 如果 Requester 向不支持 10-Bit Completer Capability 的 Completer 发送 10-Bit Tag Request，则返回的 Completion 将携带 Tag [9:8]等于 00b 的 Tag。由于 Requester 被禁止为 10-Bit Tag 生成这些 Tag 值，因此此类完成将作为 Unexpected Completion 处理，默认情况下是 Advisory Non-Fatal Error。Requester 必须遵循标准的 PCI Express 错误处理要求。
- ◆ 当 Requester 处理带有无效 10-Bit Tag 的 Completion 作为 Unexpected Completion 时，原始请求可能会导致完成超时。如果 Requester 以某种特定于设备的方式处理完成超时条件，以避免数据损坏，则允许 Requester 按标准要求通过标准 PCI Express 错误处理机制抑制对完成超时的处理。
- ◆ 如果 Requester 支持将 10-Bit Tag Request 同时发送给某些 Completer，将较小 Tag 的 Request 同时发送给其他 Completer，则 Requester 必须遵守发送较小 Tag Request 的 Extended Tag Field Enable 字段的设置。也就是说，如果该位为 0，则仅 Tag 字段的低 5 比特可以为非零；否则为 0。如果该位置 1，则仅 Tag 字段的低 8 比特可以为非零。
- ◆ 如果 Requester 支持同时向某些 Completer 发送 10-Bit Tag Request，并向其他 Completer 发送较小 Tag 的 Request，则 Requester 必须确保，如果任何 10-Bit Tag Request 是由不支持 10-Bit Tag Completer Capability 的 Completer 完成的，则任何 outstanding 的 10-Bit Tag 都不能别名为 outstanding 的较小 Tag。请参阅本节后面的“Using 10-Bit Tags and Smaller Tags Concurrently”的实现说明。

- ■ Extended Tag Field Enable 字段的默认值是特定于实现的。 10-Bit Tag Requester Enable 字段的默认值为 0b。
- ■ 如果向多个未完成的请求发出了不唯一的 Tag 值，则 Receiver/Completer 的行为是不确定的。
- ■ 如果使用 Phantom Function Number 扩展 outstanding request 的数目，则对于所有需要该请求者完成的 outstanding Request，Phantom Function Number 和 Tag 字段的组合必须唯一。

- 对于 Posted Request，Tag[9:8]为保留值。
- 对于 TH 字段置 1 的 Posted Request，将 Tag [7:0]字段改用于 ST [7:0]字段（有关详细信息，请参见 Section 2.2.7.1）。对于 TH 字段为 0 的 Posted Request，Tag [7:0]字段未定义并且可以包含任何值。（某些 Vendor_Defined Message 是此规则的例外，请参阅 Table F-1。）
  - ■ 对于 TH 字段为 0 的 Posted Request，Tag [7:0]字段中的值不得影响接收方对请求的处理。
  - ■ 对于 TH 字段为 1 的 Posted Request，ST [7:0]字段中的值可能会影响 Completer 处理请求。
- Requester ID 和 Tag 的组合形成一个全局标识符，即层次结构中每个事务的 Transaction ID。
- 所有的 Request 和 Completion TLP 中都包含 Transaction ID。
- Requester ID 是一个 16 比特值，对于层次结构中的每个 PCI Express Function 都是唯一的。
- Function 必须捕获 Function 完成的所有 Type 0 Configuration Write Request 携带的 Bus Number 和 Device Number，并在由 Device/Function 发起的所有请求的 Requester ID 的 Bus Number 和 Device Number 字段中使用这些 Number。建议仅为成功完

公众号·芯片工者相随双

成的请求捕获这些 Number。

例外：可以通过特定于实现的方式将总线和设备号分配给 RC 中的设备，并将设备号分配给 Switch 中的下游端口。

请注意，总线号和设备号可能会在运行时更改，因此有必要在每个 Configuration Write Request 中重新捕获此信息。

建议寻址未实现的 Function 的 Configuration Write Request 不影响捕获的总线号和设备号。

注 1：在 ARI 设备中，Function 只需捕获 Bus Number。ARI 设备可以按设备或按 Function 保留捕获的 Bus Number。如果捕获的 Bus Number 是按设备保留的，则需要所有 Function 来更新和使用公共的 Bus Number。

注 2：ARI 设备的 Requester ID 不包含 Device Number 字段。参见 Sectin 2.2.4.2。

注 3：ARI 设备只有 Bus Number 可以改变。

- 代表自己生成请求时（例如，用于错误报告），Switch 必须使用与逻辑上与端口相关联的桥的 Primary Side（请参阅 Section 7.1）相关联的 Requester ID 发起请求。
- 在发起对 Function 的配置写之前，不允许该 Function 发送 Non-Posted Request（需要有效的 Requester ID 才能正确路由所产生的完成 TLP）。
  - 例外：允许 RC 中的 Function 在对系统引导设备进行访问的软件启动配置之前发起请求。

  请注意，此规则和例外与用于系统初始化和配置的现有 PCI 模型一致。
- 与设备关联的每个 Function 都必须设计唯一 Function Number。注意：每个 non-ARI Device 最多可以包含八个 Function。每个 ARI 设备最多可包含 256 个 Function。
- Switch 不能改动经过的 TLP 的 Transaction ID 。
- 在某些情况下，需要 PCI Express to PCI/PCI-X Bridge 才能为从 PCI 或 PCI-X 总线转发的请求生成 Transaction ID。

IMPLEMENTATION NOTE

Increasing the Number of Outstanding Requests

为了将需要完成的 outstanding Request 的最大可能数目增加到超过 256，如果 Phantom Functions Enable 字段置 1（参见 Section 7.8.4），则设备可以使用未分配给已实现 Function 的 Function Number 来逻辑扩展 Tag 标识符。对于 single-Function 设备，这可以使 outstanding Request 的最大数量最多增加 8 倍。

Unclaimed Function Number 就叫 Phantom Function Number（PFN）。

IMPLEMENTATION NOTE

Considerations for Implementing 10-Bit Tag Capabilities

10-Bit Tag 的使用使请求者可以将 outstanding Non-Posted Request（NPR）的数量从 256 增加到 768，这对于非常高速率的 NPR 可以避免 Tag 的可用性成为瓶颈。以下公式给出了有效负载带宽、outstanding NPR 数量和其他因素之间的基本关系：

BW = S * N / RTT，其中

BW = payload bandwidth

S = transaction payload size

N = number of outstanding NPR

公为昌·芯片业者相承双

RTT = transaction round-trip time

通常，只有使用相对较小 transaction 的高速率链路上的高速率请求者才能从其 outstanding NPR 数量增加到超过 256 中受益，尽管这也可以帮助在事务往返时间较长的配置中保持性能。

在支持 10-Bit Tag Requester Capability 的请求者需要以多个完成者为目标的配置中，需要确保请求者仅向支持 10-Bit Tag Completer Capability 的完成者发送 10-Bit Tag Request。如果所有完成者都支持此 Capability，则将大大简化此过程。

为了使业界普遍使用 10-Bit Tag，强烈建议所有 Function 都支持 10-Bit Tag Completer Capability。通过新的实现，不需要本身同时使用大量 NPR 的完成者通常可以在内部跟踪 10-Bit Tag，并以适度的增量在 Completion 中返回。

实际上同时处理大量 NPR 的完成者可能需要大量额外的硬件资源（存储空间），但是通常无法实现 10-Bit Tag 的全部性能优势，除非完成者实际同时处理更多的 NPR。

对于 RC 支持 10-Bit Tag Completer Capability 的平台，强烈建议配置 PCIe 层次结构的平台固件或操作软件在支持 10-Bit Tag Requester Capability 的 Endpoint 中自动设置 10-Bit Tag Requester Enable 字段。这将启用一类重要的支持 10-Bit Tag 的适配器，这些适配器只向主机内存发送内存读取请求。

对于 RCiEP 以外的 Endpoint，可以通过检查其关联 RP 中的 10-Bit Tag Completer Supported 字段来确定 RC 是否为每个 Port 支持 10-Bit Tag Completer Capability。RCiEP 没有关联的 RP，因此，由于这个原因，除非 RC 支持它们的 10-Bit Tag Completer Capability，否则不允许它们设置其 10-Bit Tag Completer Supported 字段。 因此，软件不需要对 RCiEP 执行单独的检查。

不支持 10-Bit Tag Completer Capability 的 Switch 仍能够正确转发带有 10-Bit Tag 的 NPR 和 Completion，因为这两个新的 Tag 字段位于以前保留的 TLP Header 字段中，并且要求 Switch 无需修改的转发 TLP Header 中的这些字段。但是，如果这样的 Switch 检测到带有 10-Bit Tag 的 NPR 的错误，并且该 Switch 通过充当 NPR 的完成者来处理错误，则完成的结果将具有无效的 10-Bit Tag。因此，强烈建议使用 10-Bit Tag 的所有组件之间的 Switch 支持 10-Bit Tag Completer Capability。请注意，支持 16.0 GT/s 或更高数据速率的 Switch 必须支持 10-Bit Tag Completer Capability。

支持 10-Bit Tag Requester Capability 的请求者发送的配置请求的目的地可能支持 10-Bit Tag Completer Capability，也可能不支持。此请求者如何确定发送的 NPR 哪些包含 10-Bit Tag 的实现不在本规范范围之内。

IMPLEMENTATION NOTE

Using 10-Bit Tags and Smaller Tags Concurrently

如本节前面所述，如果请求者支持同时向某些完成者发送 10-Bit Tag Request，并向其他完成者发送较小的 Tag 请求，则如果任何 10-Bit Tag Request 是由不支持 10-Bit Tag Completer Capability 的完成者完成的，则请求者必须确保没有 outstanding 10-Bit Tag 可以别名到 outstanding 的较小 Tag 能力。

一种实现方法是让请求者将其 8-bit Tag 空间划分为两个区域：一个区域仅用于较小的标记（8 位或 5 位 Tag），另一个区域仅用于 10-Bit Tag 中较低的 8 位。请注意，这将强制在 10-Bit Tag 可用的标记空间和较小的标记之间进行权衡。

例如，如果请求者将其 8-bit Tag 空间划分为仅将最低的 4 位用于较小的标记，则这最多支持 16 个未完成的较小标记，并将 10-Bit Tag 空间减少 3×16 个值，支持 768-48=720 个未完成的 10-Bit Tag。许多其他分区选项都是可能的，所有这些选项都减少了未完成请求的总数。一般来说，为较小的标记保留 N 个值会将 10-Bit Tag 空间减少 3*N 个值，较小的标记加上 10-Bit Tag 的总数最终为 768-2*N。

2.2.6.3 Transaction Descriptor – Attributes Field

Attributes 字段用于提供额外的信息以允许修改 Transaction 的默认处理。这些修改适用于处理系统内 Transaction 的不同方面，例

如：

- Ordering
- 硬件一致性管理（snoop）

请注意，Attributes 是提示，可用于优化业务处理。支持级别取决于特定 PCI Express 外围设备和平台构件的应用需求。有关这些属性的更多详细信息，请参阅 PCI-X 2.0。请注意，Attributes 字段的 bit 2 与 bit 1 和 bit 0 不相邻（见 FIgure 2-17 和 Figure 2-18）。

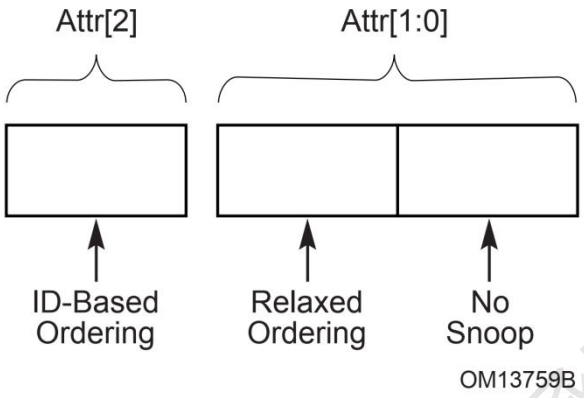


Figure 2-16 Attributes Field of Transaction Descriptor

2.2.6.4 Relaxed Ordering and ID-Based Ordering Attributes

Table 2-9 定义了 Relaxed Ordering 和 ID-Based Ordering 属性字段的状态。这些属性将在 Section 2.4 中讨论。请注意，Relaxed Ordering 和 ID-Based Ordering 属性在位置上不相邻（请参见 Figure 2-5）。

Table 2-9 Ordering Attributes

Attribute Bit [2]	Attribute Bit [1]	Ordering Type	Ordering Model
0	0	Default Ordering	PCI Strongly Ordered Model
0	1	Relaxed Ordering	PCI-X Relaxed Ordering Model
1	0	ID-Based Ordering	Independent ordering based on Requester/Completer ID
1	1	Relaxed Ordering plus ID-Based Ordering	Logical “OR” of Relaxed Ordering and IDO

对于 Configuration Request，I/O Request，作为 MSI 的 Memory Request 和 Message Request（除非特别允许），Attribute bit [1]不使用且必须将其设置为 0b。

Attribute bit [2]，即 IDO，保留用于 Configuration Request 和 I/O Request。IDO 并非为所有 Memory Request 保留，包括 MSI。除非明确禁止，否则 IDO 不保留用于消息请求。仅当设置了 Device Control 2 寄存器中的 IDO Request Enable 字段时，才允许请求者设置 IDO。

在确定 TLP 是否为 Malformed Packet 时，接收方不得检查 IDO 字段的值。

仅当 Device Control 2 寄存器中的 IDO Request Enable 字段设置为 1 时，允许完成者设置 IDO。不需要将 IDO 的值从请求复制到该请求的完成中。如果完成者启用了 IDO，则建议完成者为所有完成设置 IDO，除非有特定原因不这样做（请参阅 Appendix E）。

不需要支持在根端口之间对等转发 TLP 的 RC 来保留从入口到出口的 IDO 位。

2.2.6.5 No Snoop Attribute

Table 2-10 定义了 No Snoop 属性字段的状态。请注意，No Snoop 属性不会更改事务顺序。

Table 2-10 Cache Coherency Management Attribute

No Snoop Attribute (b)	Cache Coherency Management Type	Coherency Model
0	Default	Hardware enforced cache coherency expected
1	No Snoop	Hardware enforced cache coherency not expected

对于 Configuration Request，I/O Request，作为 MSI 的 Memory Request 和 Message Request（除非特别允许），此属性不使用且必须将其设置为 0b。

2.2.6.6 Transaction Descriptor – Traffic Class Field

业务类别（TC，Traffic Class）是一个 3 比特字段，可将 Transaction 区分为八个业务类别。

TC 机制与 PCI Express 虚通道机制一起是实现差异化业务服务的基本要素。每个 PCI Express 事务层数据包都使用 TC 信息作为在 PCI Express 中首尾相连的不变标签。当数据包在整个拓扑中传输时，此信息将在每个链路上以及每个 Switch 内使用，以对业务服务做出正确决策。服务的关键方面是根据数据包的 TC 标签通过相应的虚通道对数据包进行路由。Section 2.5 节详细介绍了 VC 机制。

Table 2-11 给出了 TC 字段的编码。

Table 2-11 Definition of TC Field Encodings

TC Field Value (b)	Definition
000	TC0: Best Effort service class (General Purpose I/O) (Default TC - must be supported by every PCI Express device)
001 to 111	TC1 toTC7: Differentiated service classes (Differentiation based on Weighted-Round-Robin (WRR) and/or priority)

系统软件决定 TC 标签和 TC/VC 映射，以便提供满足目标平台要求的差异化服务。

业务类别的概念仅适用于 PCI Express 互连结构。如何将 PCI Express TC 服务策略转换为非 PCI Express 互连上的策略的特定要求超出了本规范的范围。

2.2.7 Memory, I/O, and Configuration Request Rules

下述规则适用于 Memory 请求、IO 请求和配置请求。

- 除了公共的 header 字段外，所有 Memory 请求、IO 请求和配置请求还包括以下字段：
  - Requester ID[15:0] 和 Tag[9:0]，组成了 Transaction ID 。
  - Last DW BE[3:0]和 1st DW BE[3:0]字段。对于 TH 字段置 1 的 Memory Read Request 和 AtomicOp Request，header 中的 Last DW BE[3:0]和 1st DW BE[3:0]字段被重新用于携带 ST [7:0]字段 。对于 TH 字段设置为 0 的 Memory Read Request，请参见 Section 2.2.5 的 First/Last DW Byte Enable 的规则。 对于 TH 字段置 1 的 AtomicOp Request，DW BE 字段的值隐

含为保留值。对于 TH 字段设置为 0 的 AtomicOp Request，DW BE 字段为保留值。

Memory Request 要遵循以下规则：

- Memory Request 采用地址路由，有 32bit 地址格式和 64bit 地址格式，见 Figure 2-17 和 Figure 2-18。

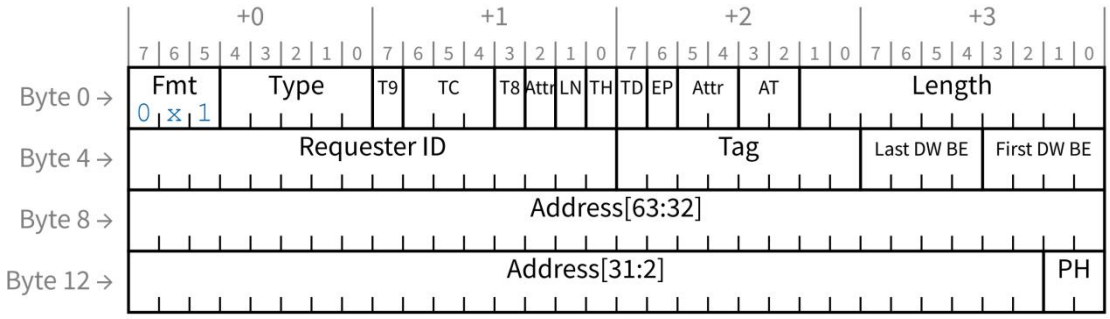


Figure 2-17 Request Header Format for 64-bit Addressing of Memory

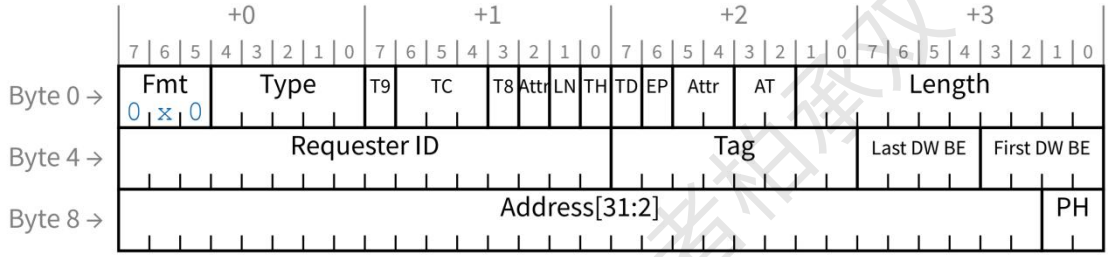


Figure 2-18 Request Header Format for 32-bit Addressing of Memory

- Memory Read Request 的 Length 字段的值不能超过 Max_Read_Request_Size（见 Section 7.5.3.4）。
- 对于 AtomicOp Request，Table 2-12 中指定了架构化的操作数大小及其关联的 Length 字段值。完成者必须检查 Length 字段值。如果该值与设计值不匹配，则完成者必须将 TLP 作为 Malformed TLP 处理。 否则，如果该值与完成者支持的操作数大小不匹配，则完成者必须将 TLP 作为不支持的请求（UR）处理。这将报告的与接收端口相关的错误（请参见 Section 6.2）。
  - FetchAdd Request 包含一个操作数，即要“ add”的值。
  - Swap Request 包含一个操作数，即要“swap”的值。
  - CAS Request 包含两个操作数。数据区域中的第一个是“compare”值，第二个是“swap”值。

Table 2-12 Length Field Values for AtomicOp Requests

AtomicOp Request	Length Field Value for Architected Operand Sizes		
	32 Bits	64 Bits	128 Bits
FetchAdd, Swap	1 DW	2 DW	N/A
CAS	2 DW	4 DW	8 DW

- 对于 AtomicOp Request，地址必须自然地与操作数大小对齐。完成者必须检查是否违反此规则。如果 TLP 违反此规则，则该 TLP 为 Malformed TLP。这将报告的与接收端口相关的错误（请参见 Section 6.2）。
- Request 指定的 Address/Length 组合不能导致一个跨 4 KB 边界的内存空间访问。
  - 接收者可以选择检查是否违反此规则。如果执行此检查的接收方确定 TLP 违反了此规则，则该 TLP 是 Malformed TLP。

- 对于 AtomicOp Request，会强制 Completer 检查用于地址自然对齐，所以它已经保证访问不会跨 4KB 边界，因此不需要单独的 4KB 边界检查。
- 请注意，如果对 AtomicOp CAS Request 执行了 4KB 边界检查，则此检查必须能够理解基于两个操作数的 TLP Length 值的大小，而对内存空间的访问则基于一个操作数的大小。

IMPLEMENTATION NOTE

Generation of 64-bit Addresses

强烈建议 PCI Express Endpoint 能够生成完整范围的 64 位地址。但是，如果 PCI Express Endpoint 支持较小的地址范围，并且无法达到给定平台环境所需的完整地址范围，则相应的设备驱动程序必须确保所有内存事务目标缓冲区都在 Endpoint 支持的地址范围内。确保这一点的方法是特定于平台和操作系统的，并且超出了本规范的范围。

I/O 请求要遵循以下规则：

- I/O Request 采用地址路由，且只能使用 32 bit 地址格式（see Figure 2-19）。

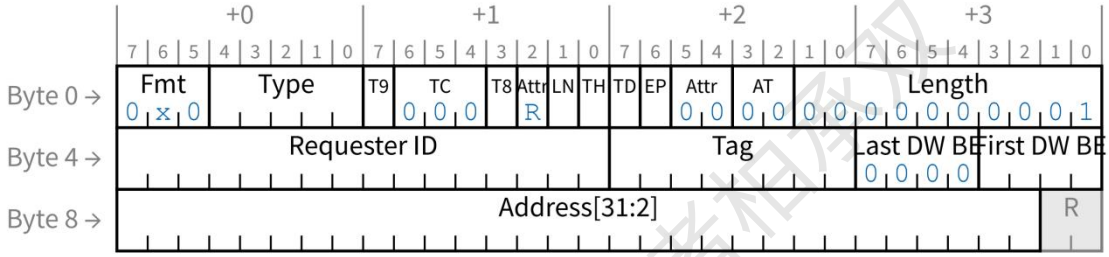


Figure 2-19 Request Header Format for I/O Transactions

- I/O Request 有以下限制：
  - TC[2:0] 必须等于 000b 。
  - LN 不适用于 I/O Request，必须设置为保留值。
  - TH 不适用于 I/O Request，必须设置为保留值。
  - Attr[2]为保留值。
  - Attr[1:0]必须等于 00b。
  - AT[1:0]必须等于 00b。不建议接收者检查这两比特。
  - Length[9:0]必须等于 00 0000 0001b 。
  - Last DW BE[3:0]必须等于 0000b。

  接收者可以选择检查是否违反了这些规则（但不能检查保留位）。这些检查是独立可选的（请参见 Section 6.2.3.4）。 如果实施这些检查的接收方确定 TLP 违反了这些规则，则该 TLP 是 Malformed TLP。
    - 如果实现，这将报告的与接收端口相关的错误（请参见 Section 6.2）。

配置请求要遵循以下规则：

- Configuration Request 采用 ID 路由，并使用 3DW header。
- 除了所有 Memory、I/O 和 Configuration 请求中包含的 header 字段以及 ID 路由字段之外，配置请求还包含以下字段（请参见 Figure 2-20）：

公众号·芯片业者相关

- Register Number[5:0]
- Extended Register Number[3:0]

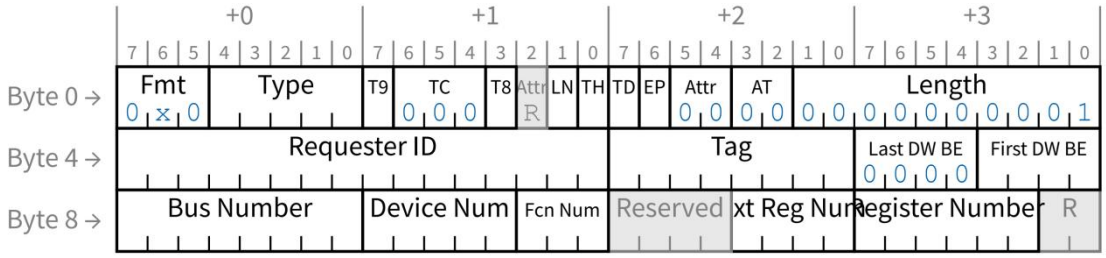


Figure 2-20 Request Header Format for Configuration Transactions

- Configuration Request 必须遵守以下限制：
  - TC[2:0] 必须等于 000b 。
  - LN 字段对 Configuration Request 没用，必须设置为保留值。
  - TH 字段对 Configuration Request 没用，必须设置为保留值。
  - Attr[2]为保留值。
  - Attr[1:0]必须等于 00b。
  - AT[1:0]必须等于 00b。
  - Length[9:0]必须等于 00 0000 0001b 。
  - Last DW BE[3:0]必须等于 0000b。

  接收者可以选择检查是否违反了这些规则（但不能检查保留位）。这些检查是独立可选的（请参见 Section 6.2.3.4）。 如果实施这些检查的接收方确定 TLP 违反了这些规则，则该 TLP 是 Malformed TLP。
    - 如果实现，这将报告的与接收端口相关的错误（请参见 Section 6.2）。

消息信号中断（MSI/MSI-X）机制使用 Memory Write Request 来表示中断消息（请参见 Section 6.1.4）。用于 MSI/MSI-X 事务的请求格式与上面定义的 Memory Write Request 格式相同，在排序，流控制和数据完整性方面，MSI/MSI-X 请求与 Memory Write 没有区别。

2.2.7.1 TPH Rules

- 为 TPH 定义了两种格式。所有提供 TPH 的请求都必须使用 Baseline TPH format（见 Figure 2-20 和 Figure 2-21）。带有 Optional TPH TLP Prefix 的格式扩展了 TPH 字段（请参见 Figure 2-19），以为 Steering Tag （ST）字段提供其他字段。

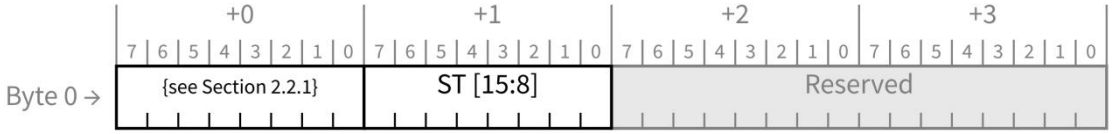


*Figure 2-21 **TPH TLP Prefix***

- Optional TPH TLP Prefix 用于扩展 TPH 字段。
  - 通过解码 byte 0 可以确定 TPH TLP Prefix 的存在（byte 0[7:5]= 111b，表示当前 TLP 为一个 TLP Prefix）。

公众号：芯片业者相承双

Table 2-13 TPH TLP Prefix Bit Mapping

Fields	TPH TLP Prefix
ST(15:8)	Bits 7:0 of byte 1
Reserved	Bits 7:0 of byte 2
Reserved	Bits 7:0 of byte 3

- 对于以内存空间为目标的请求，TH 字段值为 1b 表示 TLP header 中存在 TPH 和可选的 TPH TLP Prefix。
  - 必须将提供 TPH 的请求的 TH 字段置 1。
  - 对于带有 TPH TLP Prefix 的请求，必须将 TH 字段置 1。
  - 当 TH 字段为 0，必须将 PH 字段置为保留值。
  - 对于其他 Request，TH 位和 PH 字段不适用并为保留值。
- Processing Hints（PH）字段映射如 Figure 2-22，Figure 2-23 和 Table 2-14 所示。

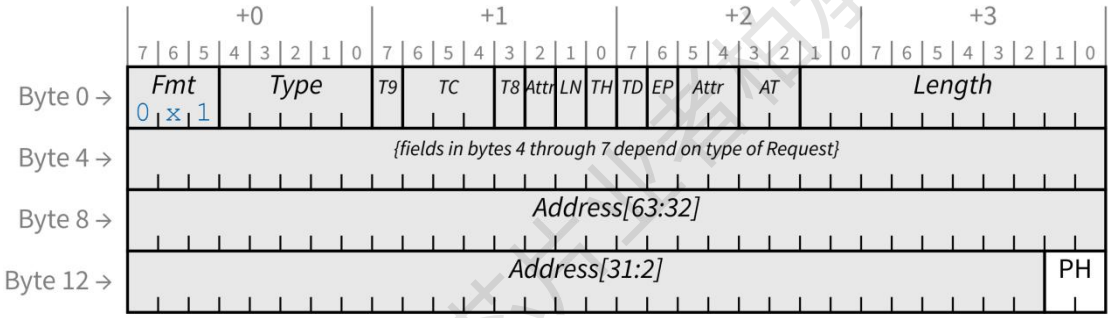


Figure 2-22 Location of PH[1:0] in a 4 DW Request Header

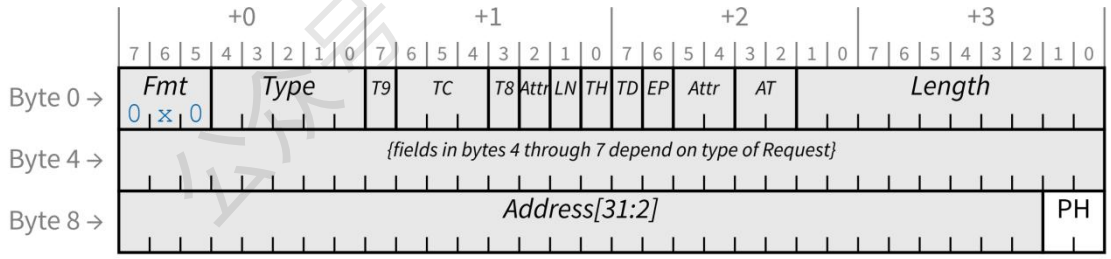


Figure 2-23 Location of PH[1:0] in a 3 DW Request Header

Table 2-14 Location of PH[1:0] in TLP Header

PH	32-bit Addressing	64-bit Addressing
1:0	Bits 1:0 of Byte 11	Bits 1:0 of Byte 15

- PH [1:0]字段提供有关数据访问模式的信息，Table 2-15 给出定义。

Table 2-15 Processing Hint Encoding

PH[1:0] (b)	Processing Hint	Description
00	Bi-directional data structure	Indicates frequent read and/or write access to data by Host and device
01	Requester	Indicates frequent read and/or write access to data by device
10	Target	Indicates frequent read and/or write access to data by Host
11	Target with Priority	Indicates frequent read and/or write access by Host and indicates high temporal locality for accessed data

- 如 Figure 2-24，Figure 2-25 和 Table 2-16 所示，Steering Tag（ST）字段映射到 TLP header。

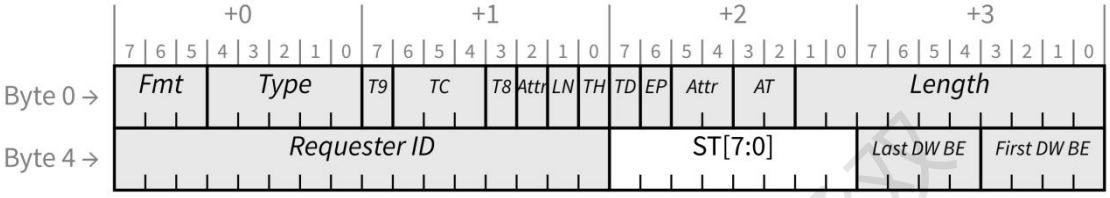


Figure 2-24 Location of ST[7:0] in the Memory Write Request Header

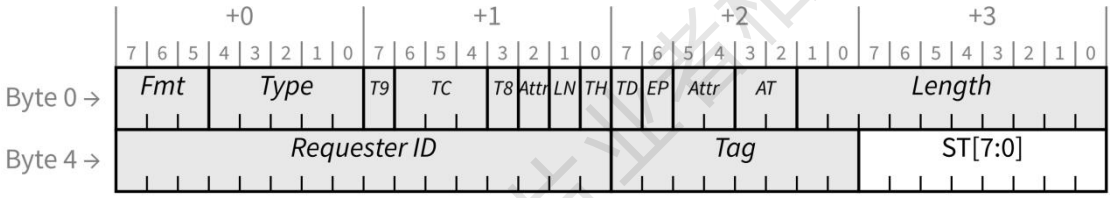


Figure 2-25 Location of ST[7:0] in Memory Read and AtomicOp Request Headers

Table 2-16 Location of ST[7:0] in TLP Headers

ST Bits	Memory Write Request	Memory Read Request or AtomicOp Request
7:0	Bits 7:0 of Byte 6	Bits 7:0 of Byte 7

- ST[7:0]字段携带 Steering Tag 值。
  - 全零值表示无 Steering Tag preference 。
  - 总共提供 255 个唯一的 Steering Tag 值。
- 不支持 TPH Completer 或路由能力并且接收到 TH 位为 1 的事务的 Function 需要忽略 TH 字段，并以与 TH 没有设置为 1 的相同事务类型的请求相同的方式处理请求。

2.2.8 Message Request Rules

本规范定义了一下 Message 类型：

- INTx Interrupt Signaling
- Power Management

公众号：芯片业务拍摄众

- Error Signaling
- Locked Transaction Support
- Slot Power Limit Support
- Vendor-Defined Messages
- Latency Tolerance Reporting (LTR) Messages
- Optimized Buffer Flush/Fill (OBFF) Messages
- Device Readiness Status (DRS) Messages
- Function Readiness Status (FRS) Messages
- Precision Time Measurement (PTM) Messages

以下规则适用于所有 Message Request。还为每种特定的 Message Request 指定了规则。

- 除了公共的 header 字段之外，所有 Message Request 都包括以下字段（请参见 Figure 2-26）：
  - Requester ID[15:0]和 Tag[9:0]，组成了 Transaction ID 。
  - Message Code[7:0]，指明了请求包含的特定消息。

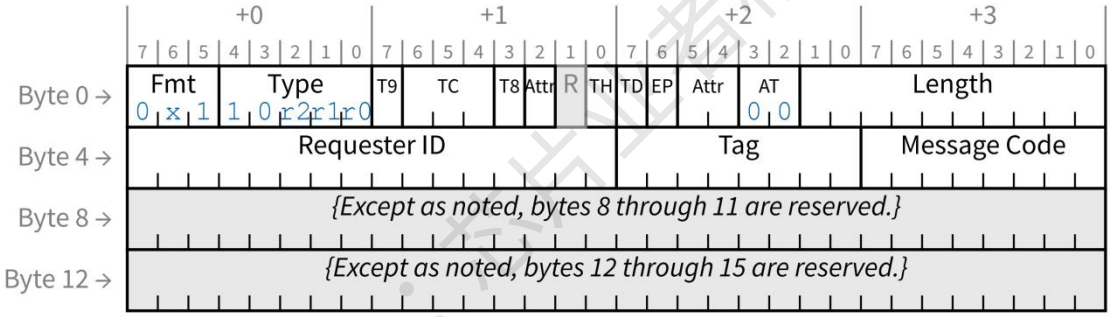


Figure 2-26 Message Request Header

- 除 Vendor_Defined Message（可以使用 Msg 或 MsgD）和 Set_Slot_Power_Limit Message（使用 MsgD）外，所有 Message Request 均使用 Msg Type 字段编码。
- 消息代码字段必须完整解码（不允许使用消息别名）。
- 除非特别指出，否则 Attr [2]字段不是保留值。
- 除非另有说明，否则 Attr [1:0]字段为保留值。
- LN 不适用于消息请求，并且该位为保留值。
- 除非另有说明，TH 不适用于消息请求，并且该位为保留值。
- AT [1:0]必须为 00b。接收者不需要进行检查。
- 除非另有说明，byte 8 至 byte 15 为保留值。
- 消息请求是 posted 请求，不需要 Completion。
- 消息请求遵循与内存写请求相同的排序规则。

公众号·未出生者相承双

许多类型的消息（包括 Vendor-Defined Message）可能在 non-D0 状态下可用，并且强烈建议当 Port 的 Bridge Function 处于 D1，D2 和 $D3_{Hot}$ 中时，对 Port 的消息处理应与在 D0 时相同。强烈建议 Type 0 Function 支持在 non-D0 状态下生成和接收消息。

除了地址路由和 ID 路由外，消息还支持其他几种路由机制。这些机制被称为“隐式路由”，因为没有地址或 ID 指定目的地，而是路由类型隐含了目的地。以下给出了消息路由机制的规则：

- 使用 Type 字段的 r [2:0]子字段确定消息路由。
  - 消息路由的编码如 Table 2-17。
  - 在以下各节中为每个消息定义允许的值。

Table 2-17 Message Routing

r[2:0] (b)	Description	Bytes 8 to 15[15]
000	Routed to Root Complex	Reserved
001	Routed by Address[16]	Address
010	Routed by ID	See Section 2.2.4.2
011	Broadcast from Root Complex	Reserved
100	Local - Terminate at Receiver	Reserved
101	Gathered and routed to Root Complex[17]	Reserved
110 to 111	Reserved - Terminate at Receiver	Reserved

注 15：除非另有说明，例如 Vendor_Defined Message 这些字节可能会自己定义。

注 16：本规范没有定义任何使用地址路由的消息。

注 17：此路由类型仅用于 PME_TO_Ack，在 Section 5.3.3.2.1 中进行了描述。

2.2.8.1 INTx Interrupt Signaling - Rules

MSI 或 MSI-X 是 PCI Express 中首选的中断信令机制（请参见 Section 6.1）。但是，在某些系统中，可能有一些功能不支持 MSI 或 MSI-X 机制。在无法使用 MSI 或 MSI-X 机制的情况下，可以使用 INTx 虚拟中断线信令机制来支持 Legacy Endpoint 和 PCI Express/PCI(-X) Bridge 。Switch 必须支持此机制。以下规则适用于 INTx 中断信令机制：

- INTx 机制使用八个不同的消息（请参见 Table 2-18）。
- Assert_INTx/Deassert_INTx Message 不包含数据载荷（TLP Type is Msg）。
- Length 字段为保留值。
- 对于 Assert_INTx/Deassert_INTx Message，Requester ID 的 Function Number 字段为 0。请注意，对于 non-ARI 和 ARI 的 Requester ID，Function Number 字段的比特数不同。
- Assert_INTx/Deassert_INTx Message 只会由 Upstream Port 发送。
  - 接收者可以选择检查是否违反此规则。如果执行此检查的接收方确定 Assert_INTx/Deassert_INTx 违反了此规则，则它必须将 TLP 视为 Malformed TLP 。
- Assert_INTx/Deassert_INTx Message 使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP 。

Table 2-18 INTx Mechanism Messages

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support[18]				Description/Comments
			RC	Ep	Sw	Br	
Assert_INTA	0010 0000	100	All: r tr As Required: t t				Assert INTA virtual wire Note: These Messages are used for Conventional PCI-compatible INTx emulation.
Assert_INTB	0010 0001	100	All: r tr As Required: t t				Assert INTB virtual wire
Assert_INTC	0010 0010	100	All: r tr As Required: t t				Assert INTC virtual wire
Assert_INTD	0010 0011	100	All: r tr As Required: t t				Assert INTD virtual wire
Deassert_INTA	0010 0100	100	All: r tr As Required: t t				Deassert INTA virtual wire
Deassert_INTB	0010 0101	100	All: r tr As Required: t t				Deassert INTB virtual wire
Deassert_INTC	0010 0110	100	All: r tr As Required: t t				Deassert INTC virtual wire
Deassert_INTD	0010 0111	100	All: r tr As Required: t t				Deassert INTD virtual wire

注 18： RC = RC ； Sw = Switch (only used with “Link” routing) ； Ep = Endpoint ； Br = PCI Express (primary) to PCI/PCI-X (secondary) Bridge ； r = Supports as Receiver ； t = Supports as Transmitter

公众号·芯片业者柏承双

Assert_INTx/Deassert_INTx Message 构成每个称为 A、B、C 和 D 的传统 PCI 中断的四个“虚拟线”。以下规则描述了这些虚拟线的操作：

- 每个链路两端的组件必须使用 Assert/Deassert Message 来跟踪四条虚拟线的逻辑状态，以分别表示每条相应虚拟线的有效和无效的变化。
  - Assert_INTx 表示 INTx（x = A、B、C 或 D）虚拟线的有效状态。
  - Deassert_INTx 表示 INTx（x = A、B、C 或 D）虚拟线的无效状态。
- 当 INTx 虚拟线的本地逻辑状态在上游端口更改时，该端口必须使用适当的 Assert_INTx 或 Deassert_INTx 消息将此状态的更改传达给同一链路另一侧的下游端口。

注意：重复的 Assert_INTx/Deassert_INTx Message 无效，但不是错误。

- 当 Command 寄存器的 Interrupt Disable 字段（Section 7.5.1.1）设置为 1b 时，INTx 中断信令被禁用。
  - 必须通过发送 Deassert_INTx 消息来取消任何有效的 INTx 虚拟线。
- 虚拟和实际的 PCI to PCI Bridge 必须根据桥接器次级侧上设备的 Device Number 映射在桥接器次级侧上跟踪的虚拟线，如 Table 2-19 所示。
- Switch 必须为每个下游端口独立跟踪四根虚拟线的状态，并在其上游端口上显示一组“折叠”的虚拟线。
- 如果下游端口变为 DL_Down 状态，则必须把与该端口关联的 INTx 虚拟线路置为无效，并且相应更新上游端口虚拟线路状态。
  - 如果这导致取消任何上游 INTx 虚拟线有效，则必须由上游端口发送 Deassert_INTx 消息。
- RC 必须针对其每个下游端口独立跟踪四根 INTx 虚拟线的状态，并将这些虚拟信号映射到系统中断资源。
  - 此映射的详细信息是特定于系统实现的。
- 如果 RC 的下游端口进入 DL_Down 状态，则必须把与该端口关联的 INTx 虚拟线路置为无效，并且必须丢弃任何关联的系统中断资源请求。

公众号·芯片业者拍承双

Table 2-19 Bridge Mapping for INTx Virtual Wires

Requester ID[7:3] from the Assert_INTx/Deassert_INTx Message received on Secondary Side of Bridge (Interrupt Source[19]) If ARI Forwarding is enabled, the value 0 must be used instead of Requester ID[7:3].	INTx Virtual Wire on Secondary Side of Bridge	Mapping to INTx Virtual Wire on Primary Side of Bridge
0,4,8,12,16,20,24,28	INTA	INTA
	INTB	INTB
	INTC	INTC
	INTD	INTD
1,5,9,13,17,21,25,29	INTA	INTB
	INTB	INTC
	INTC	INTD
	INTD	INTA
2,6,10,14,18,22,26,30	INTA	INTC
	INTB	INTD
	INTC	INTA
	INTD	INTB
3,7,11,15,19,23,27,31	INTA	INTD
	INTB	INTA
	INTC	INTB
	INTD	INTC

注 19：Assert_INTx/Deassert_INTx Message 的 Requester ID 将与该链路上的消息的发送方相对应，而不必与中断的原始源相对应。

IMPLEMENTATION NOTE

System Interrupt Mapping

请注意，系统软件（包括 BIOS 和操作系统）需要理解系统整个拓扑结构（包括分层连接的 Switch 和从属的 PCI Express/PCI Bridge）中的传统中断（INTx 机制）的重新映射，以建立 PCI Express 设备中断和系统中断控制器中的相关中断资源的关联。 Table 2-20 所描述的重新映射应用于每个 Switch 的层次结构。另外，PCI Express/PCI Bridge 和 PCI/PCI Bridge 执行类似的映射功能。

IMPLEMENTATION NOTE

Virtual Wire Mapping for INTx Interrupts From ARI Devices

ARI 设备的隐含设备号为 0。当 ARI 感知软件（包括 BIOS 和操作系统）在 ARI 设备正上方的下游端口中启用 ARI 转发以访问其扩展功能时，软件必须理解下游端口，将使用设备编号 0 来表示 ARI 设备所有功能产生的 INTx 中断的虚拟线路映射。如果不支持 ARI 的软件尝试确定扩展功能的虚拟线路映射，则可以通过检查传统的“设备编号”字段并将其查找为非 0 来提出错误的映射。

2.2.8.2 Power Management Messages

这些消息用于支持 PCI Express 电源管理，这将在 Section 5 中详细介绍。以下规则定义了电源管理消息：

- Table 2-20 给出了所有的电源管理消息。
- 电源管理消息不包括数据有效负载（TLP 类型为 Msg）。

公众号·芯片业者相承双

- Length 字段保留。
- 对于 PM_Active_State_Nak 消息， Requester ID 中的 Function Number 字段必须包含发送该消息的下游端口的 Function Number，或者为 000b，以便与该规范的早期版本兼容。
- 对于 PME_TO_Ack 消息，Requester ID 中的 Function Number 字段保留，否则，为了与本规范的早期版本兼容，必须包含与上游端口关联的 Function 之一的 Function Number。请注意，对于 non-ARI 和 ARI 的 Requester ID，Function Number 的比特数不同。
- 电源管理消息必须使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP 。

Table 2-20 Power Management Messages

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
PM_Active_State_Nak	0001 0100	100	t	r	tr	r	Terminate at Receiver
PM_PME	0001 1000	000	All:				Sent Upstream by PME-requesting component. Propagates Upstream.
			r		tr	t	
			If PME supported:				
				t			
PME_Turn_Off	0001 1001	011	t	r		r	Broadcast Downstream
PME_TO_Ack	0001 1011	101	r	t		t	Sent Upstream by Upstream Port. See Section 5.3.3.2.1 .
			(Note: Switch handling is special)				

2.2.8.3 Error Signaling Messages

错误信令消息用于信令发生在特定事务上的错误和不一定与特定事务相关联的错误。这些消息由检测到错误的设备发出。

- Table 2-21 给出了所有的错误信令消息。
- 错误信令消息不包括数据有效负载（TLP 类型为 Msg）。
- Length 字段保留。
- 对于错误信令消息，Requester ID 中的 Function Number 字段必须指示哪个 Function 正在发出错误信令。请注意，对于 non-ARI 和 ARI 的 Requester ID，Function Number 的比特数不同。
- 错误信令消息必须使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP 。

公众号·芯片业者怕承双

Table 2-21 Error Signaling Messages

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
ERR_COR	0011 0000	000	r	t	tr	t	This Message is issued when the Function or Device detects a correctable error on the PCI Express interface.
ERR_NONFATAL	0011 0001	000	r	t	tr	t	This Message is issued when the Function or Device detects a Non-Fatal, uncorrectable error on the PCI Express interface.
ERR_FATAL	0011 0011	000	r	t	tr	t	This Message is issued when the Function or Device detects a Fatal, uncorrectable error on the PCI Express interface.

消息的发起者由消息头的 Requester ID 标识。RC 将这些错误消息转换为平台级事件。有关这些消息的用法，请参阅 Section 6.2。

- ERR_COR Message 在消息头中有一个 ERR_COR Subclass（ECS）字段，该字段使不同的子类能够相互区分。 参见 Figure 2-27。ERR_NONFATAL 和 ERR_FATAL 消息没有 ECS 字段。

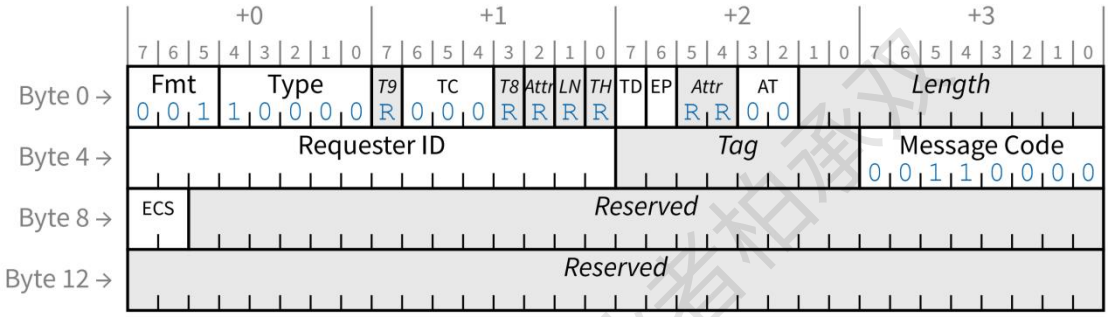


Figure 2-27 ***ERR_COR Message***

- ERR_COR Subclass（ECS）字段的编码如 Table 2-22 所示，指示 ERR_COR 消息子类。

Table 2-22 ***ERR_COR Subclass (ECS)*** Field Encodings

ECS Coding	Description
00	***ECS Legacy*** - The value inherently used if a Requester does not support ECS capability. ECS-capable Requesters must not use this value. See see Section 7.5.3.3 .
01	***ECS SIG_SFW*** - Must be used by an ECS-capable Requester when signaling a DPC or SFI event with an ERR_COR Message.
10	***ECS SIG_OS*** - Must be used by an ECS-capable Requester when signaling an AER or RP PIO event with an ERR_COR Message.
11	***ECS Extended*** - Intended for possible future use. Requesters must not use this value. Receivers must handle the signal internally the same as ECS SIG_OS.

2.2.8.4 Locked Transactions Support

Unlock Message 用于支持 Lock Transaction 的处理过程。有关 Lock Transaction 处理过程的详细信息，请参见 Section 6.5。以下规则适用于 Unlock Message 的构造：

- Table 2-23 给出了 Unlock Message 的定义。
- Unlock Message 不包括数据有效负载（TLP 类型为 Msg）。
- Length 字段保留。
- 对于 Unlock Message，Requester ID 中的 Function Number 字段为保留值。

- Unlock Message 必须使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP 。

Table 2-23 Unlock Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
Unlock	0000 0000	011	t	r	tr	r	Unlock Completer

2.2.8.5 Slot Power Limit Support

此消息用于将插槽电源限制值从 RC 或 Switch 的下游端口传送到该链路另一侧设备的（Endpoint、Switch 或 PCI Express-PCI Bridge Function）的上游端口 。

- Table 2-24 定义了 Set_Slot_Power_Limit Message。
- Set_Slot_Power_Limit Message 包含一个 1DW 的有效数据载荷（TLP Type is MsgD）。
- Set_Slot_Power_Limit Message 使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP。

Table 2-24 ***Set_Slot_Power_Limit Message***

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
Set_Slot_Power_Limit	0101 0000	100	t	r	tr	r	Set Slot Power Limit in Upstream Port

Set_Slot_Power_Limit Message 包括一个 DW 数据有效载荷。数据有效载荷是从下游端口的 Slot Capabilities 寄存器复制的，并写入链路另一侧的上游端口的 Device Capabilities 寄存器。数据有效载荷的 byte 1[1:0]映射到 Slot Power Limit Scale 字段，byte 0[7:0]映射到 Slot Power Limit Value 字段。数据有效载荷的 byte 3[7:0]，byte 2[7:0]和 byte 1[7:2]必须由发送器设置为全 0，且接收器要忽略这些值。当发生以下事件之一时，此消息必须由 RC 或 Switch 的下游端口自动发送：

- 当数据链路层报告 DL_Up 状态后，如果发送配置写访问 Slot Capabilitie 寄存器（请参见 Section 7.5.3.9）。
- 链路从 non-DL_Up 状态转换为 DL_Up 状态的任何传输（请参阅 Section 2.9.2）。如果尚未初始化 Slot Capabilitie 寄存器，则此传输是可选的。

接收 Set_Slot_Power_Limit 消息的链路另一端的组件（Endpoint、Switch 或 PCI Express-PCI Bridge Function）必须将数据有效负载中的值复制到与组件的上游端口关联的 Device Capabilities 寄存器中。 PCI Express 组件专门用于在系统平面上集成（例如，系统板），以及集成在卡/模块上的组件，其中整个卡/模块的功耗低于指定的最低功耗限制。卡/模块的外形尺寸（在相应的外形尺寸规格中定义）被允许在 Device Capabilities 寄存器的 Slot Power Limit Scale 和 Slot Power Limit Value 字段中硬接线为全 0 的值，并且不需要复制 Set_Slot_Power 有效负载的值到该寄存器中。

有关电源限制控制机制的更多详细信息，请参见 Section 6.9。

2.2.8.6 Vendor_Defined Messages

Vendor_Defined Message 允许扩展 PCI Express 消息传递功能，既可以作为 PCI Express 规范的常规扩展，也可以作为特定于供应商的扩展。尽管本规范的未来修订版可能会使用此机制来定义新的消息（请参阅下文），但本文档未特别涵盖此类扩展。本节通常定义与这些消息关联的规则。

- Vendor_Defined Message（Table 2-25）的 header 格式如 Figure 2-25。
  - Requester ID 是实现相关的。
  - 如果使用 ID 路由，则 byte 8 和 byte 9 构成目标 ID 的 16 字段。
    - 否则，这些字节将保留。
  - byte 10 和 byte 11 构成了定义消息的供应商的 PCI-SIG®定义的 Vendor ID 的 16 比特字段。
  - byte 12 至 byte 15 可用于供应商定义。

Table 2-25 Vendor_Defined Messages

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
Vendor_Defined Type 0	0111 1110	000, 010, 011, 100	See Note 1.				Triggers detection of UR by Completer if not implemented.
Vendor_Defined Type 1	0111 1111	000, 010, 011, 100	See Note 1.				Silently discarded by Completer if not implemented.

1. Note 1: Transmission by Endpoint/Root Complex/Bridge is implementation specific. Switches must forward received Messages using Routing r[2:0] field values of 000b, 010b, and 011b.

Note 1：Endpoint/RC/Bridge 的传输是特定于实现的。Switch 必须使用 Routing [2:0]字段值 000b，010b 和 011b 转发接收到的消息。

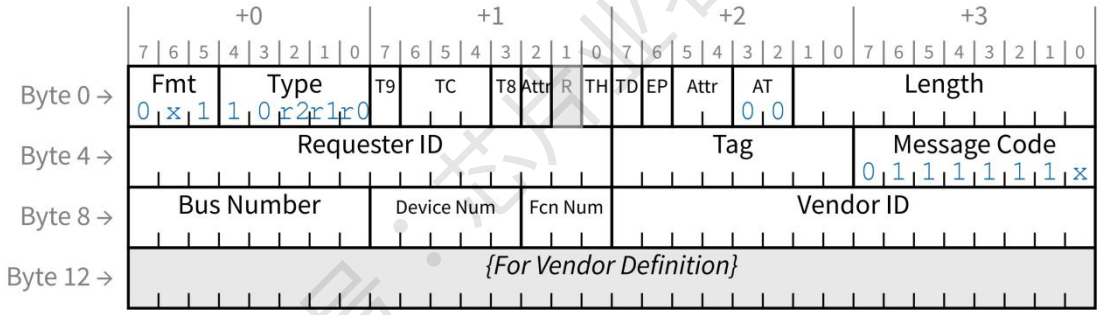


Figure 2-28 Header for Vendor-Defined Messages

- Vendor_Defined Message 包含数据有效载荷（如果不包括数据有效载荷，则 TLP 类型为 Msg；如果包括数据有效载荷，则为 MsgD）。
- Vendor_Defined Message 的 Attr[1:0]和 Attr[2]字段值为保留值。
- 由不同供应商或 PCI-SIG 定义的消息由 Vendor ID 字段中的值来区分。
  - 由特定供应商定义的消息的进一步区分超出了本文档的范围。
  - 对特定供应商定义的消息的支持是特定于实现的，超出了本文档的范围。
- 完成者会丢弃他们不打算接收的 Vendor_Defined Type 1 Message-这不是错误。
- 完成者将收到不支持的 Vendor_Defined Type 0 Message 作为 Unsupported Request 进行处理，并且根据 Section 6.2 报告错误。

PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0 定义了 Vendor_Defined Message 的其他要求，这些消息旨在与 PCI-X Device ID Message 互操作。这包括对 Tag [7:0]字段和 Length [9:0]字段的内容的限制，以及对消息头的 byte 12 到 byte 15 的特定使用。仅在 PCI Express 环境中使用的 Vendor_Defined Message（即不旨在解决 PCI Express to PCI/PCI-X Bridge 后面的目标）不受其他规则

的约束。 有关详细信息，请参阅 *PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0*。

2. 2. 8. 6. 1 PCI-SIG-Defined VDMs

PCI-SIG 定义的 VDM 是使用 PCI-SIG® Vendor ID（0001h）的 Vendor-Defined Type 1 Message。作为 Vendor-Defined Type 1 Message，如果完成者未实现，则每个消息都会被完成者静默丢弃。

除了其他 Vendor-Defined Type 1 Message 的规则之外，以下规则适用于 PCI-SIG-Defined VDM 的格式：

- PCI-SIG-Defined VDM 的 Header 的结构如 Figure 2-29 所示。
- Requester ID 字段必须填充为 Requester 对应的值。
- Message Code 必须为 01111111b 。
- Vendor ID 必须为 0001h，已分配给 PCI-SIG 。
- Subtype 字段区分特定的 PCI-SIG 定义的 VDM。有关 PCI-SIG-Defined VDM 的列表，请参见 Appendix F。

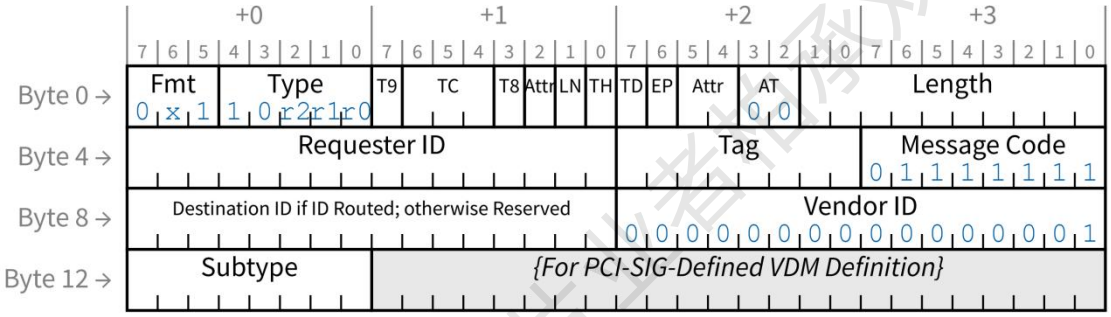


Figure 2-29 Header for PCI-SIG-Defined VDMs

2. 2. 8. 6. 2 LN Messages

LN 协议(见 Section 6.21)定义 LN Message,即 PCI SIG 定义的 VDM。每条消息的有效负载通常包含已更新或逐出的已注册 cacheline 的 64 位地址。单个 64 位地址格式同时用于 64 位和 32 位地址。由于每个 LN Message 都是一个 Vendor-Defined Type 1 Message，因此如果完成程序无法识别消息，则完成者会自动丢弃该消息。

可以使用基于 ID 的路由将 LN Message 定向到单个 Endpoint，或将其广播到给定 Root Port 下的所有设备。 一个广播 LN Message 是否发送到 RC 中的所有 Root Port 是特定于实现的。

除了其他 PCI-SIG 定义的 VDM 的规则之外，以下规则适用于 LN Message 的构成：

- Table 2-27 和 Figure 2-30 定义了 LN Message 。
- 每个 Message 必须包含 2-DW 的数据负载。
- Fmt 字段必须为 011b（4DW，带数据）。
- TLP Type 必须为 MsgD 。
- Length 字段必须为 2 。
- TC[2:0]字段必须为 000b 。

- Attr[2]，即 ID-Based Ordering (IDO)，不能为保留字段。
- Attr[1]，即 Relaxed Ordering (RO)，不能为保留字段。
- Attr[0]，即 No Snoop 字段，必须为保留字段。
- LN 字段为保留字段（相反，必须为 LN Read，LN Write 和 LN Completion 设置 LN 字段）。
- Tag 字段为保留字段。
- 如果 LN Message 是一个广播版本，则 Destination ID 字段为保留字段。
- Subtype 字段必须为 00b 。
- 如果对系统有效的 cacheline 大小为 128 字节，则 Cacheline Address 中的 Bit 6 必须为 0。对于接收 LN Message 的 Lightweight Notification Requester（LNR），如果 LNR Control 寄存器中的 LNR CLS 字段置 1，则将 LNR 配置为 128 字节 cacheline，则 LNR 必须忽略 Cacheline Address 中 Bit 6 的值。
- Notification Reason（NR）字段的编码如 Table 2-26 所示，指示发送 LN Message 的特定原因。这些编码适用于 LN Message 的定向和广播版本。

Table 2-26 Notification Reason (NR) Field Encodings

NR Coding (b)	Description
00	LN Message was sent due to a cacheline update.
01	LN Message was sent due to the eviction of a single cacheline.
10	LN Message was sent due to the eviction of all cachelines registered to this Function. For this case, the Cacheline Address is Reserved.
11	Reserved

Table 2-27 LN Messages

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
LN Message (directed)	0111 1111	010	t	r	tr	r	RC directs to a single Endpoint.
LN Message (broadcast)	0111 1111	011	t	r	tr	r	RC broadcasts to all devices under a given Root Port.

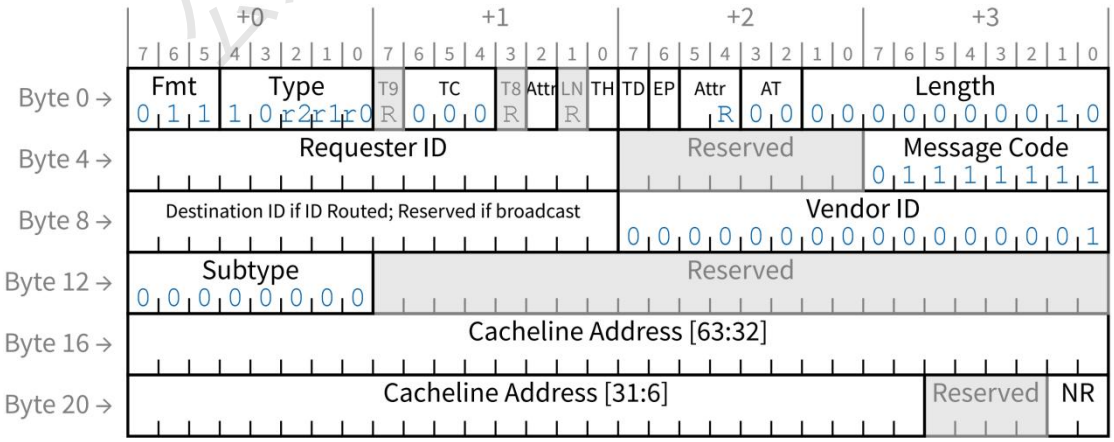


Figure 2-30 LN Message

公众号：芯片业者相承双

2. 2. 8. 6. 3 Device Readiness Status (DRS) Message

设备就绪状态（DRS）协议（请参阅 Section 6.23.1）使用 PCI-SIG-Defined VDM 机制（请参阅 Section 2.2.8.6.1）。 DRS Message 是没有数据负载的 PCI-SIG-Defined VDM（Vendor-Defined Type 1 Message）。

除了其他 PCI-SIG-Defined VDM 的规则之外，以下规则适用于 DRS Message 的构成：

- Table 2-28 和 Figure 2-31 定义了 DRS Message 。
- TLP Type 必须为 Msg 。
- TC[2:0]字段必须为 000b 。
- Attr[2:0]为保留字段。
- Tag 字段为保留字段。
- Subtype 字段为 08h 。
- Message Routing 字段必须设置为 100b - Local - 即在接收端终止。

接收者可以选择检查是否违反了这些规则（但不能检查保留位）。 这些检查是独立可选的（请参见 Section 6.2.3.4）。如果执行这些检查的接收方确定 TLP 违反了这些规则，则该 TLP 是 Malformed TLP。

- 如果执行此检查，则这是与接收端口相关的报告错误（请参阅 Section 6.2）。

Table 2-28 DRS Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
DRS Message	0111 1111	100	r	t	tr		Device Readiness Status

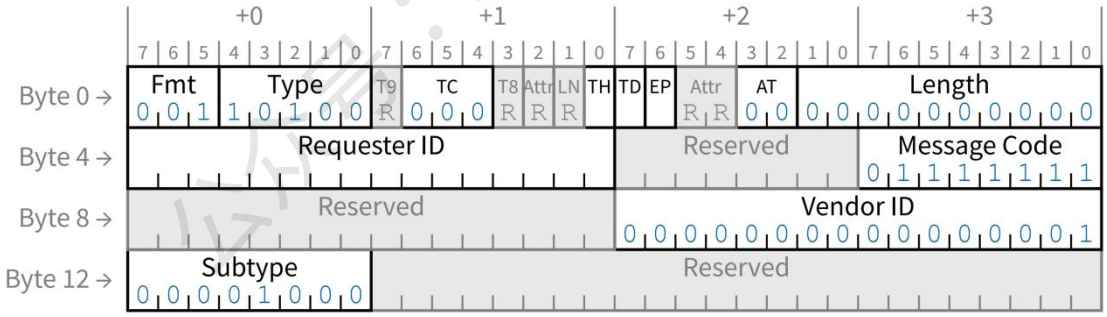


Figure 2-31 DRS Message

2. 2. 8. 6. 4 Function Readiness Status Message (FRS Message)

功能就绪状态（FRS）协议（请参阅 Section 6.23.2）使用 PCI-SIG-Defined VDM 机制（请参阅 Section 2.2.8.6.1）。 FRS Message 是没有数据负载的 PCI-SIG-Defined VDM（Vendor-Defined Type 1 Message）。

除了其他 PCI-SIG-Defined VDM 的规则之外，以下规则适用于 FRS 消息的构成：

- Table 2-29 和 Figure 2-32 定义了 FRS Message 。

- TLP Type 必须为 Msg 。
- TC[2:0]字段必须为 000b 。
- Attr[2:0]为保留字段。
- Tag 字段为保留字段。
- Subtype 字段为 09h 。
- FRS Reason[3:0]字段指示生成 FRS Message 的原因：
  - 0001b: DRS Message Received
    - 由 Message Requester ID 指示的下游端口接收到 DRS Message，并且在 Link Control 寄存器中将 DRS Signaling Control 字段设置为 DRS to FRS Signaling Enabled。
  - 0010b: $D3_{Hot}$ to D0 Transition Completed
    - 从 $D3_{Hot}$ 到 D0 的转换已经完成，并且由 Message Requester ID 指示的 Function 现在已准备就绪，并且已根据 No_Soft_Reset 位的设置返回到 $D0_{uninitialized}$ 或 $D0_{active}$ 状态（请参见 Section 7.5.2.2）。
  - 0011b: FLR Completed
    - FLR 已完成，并且 Message Requester ID 指示的 Function 现在已 Configuration-Ready。
  - 1000b: VF Enabled
    - Message Requester ID 指示了一个 Physical Function（PF）- 与该 PF 关联的所有 Virtual Function（VF）现在都已 Configuration-Ready。
  - 1001b: VF Disabled
    - Message Requester ID 指示了一个 Physical Function（PF）- 与该 PF 关联的所有 Virtual Function（VF）现在都已被禁止，且 PF 中的 SR-IOV data structure 可能不能访问。
  - Others:
    - 所有其他值保留。
- Message Routing 字段必须清为 000b - 即路由到 RC。

接收者可以选择检查是否违反了这些规则（但不能检查保留位）。 这些检查是独立可选的（请参见 Section 6.2.3.4）。如果执行这些检查的接收方确定 TLP 违反了这些规则，则该 TLP 是一个 Malformed TLP。

- 如果执行此检查，则这是与接收端口相关的报告错误（请参阅 Section 6.2）。

Table 2-29 FRS Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
FRS Message	0111 1111	000	r	t	tr		Function Readiness Status

公众号·芯片业者相承及

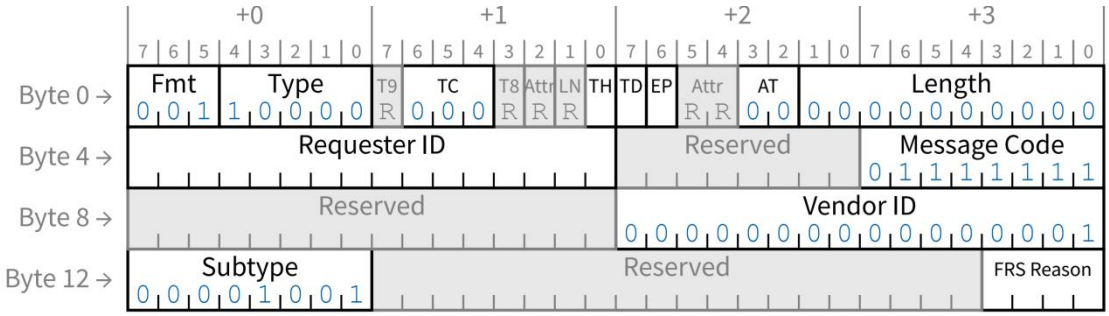


Figure 2-32 FRS Message

2.2.8.6.5 Hierarchy ID Message

Hierarchy ID Message 使用 PCI-SIG-Defined VDM 机制（请参见 Section 2.2.8.6.1）。Hierarchy ID Message 是携带有效负载（MsgD）的 PCI-SIG-Defined VDM（Vendor-Defined Type 1 Message）。

除了其他 PCI-SIG-Defined VDM 的规则之外，以下规则适用于 Hierarchy ID Message 的形成：

- Table 2-30 和 Figure 2-33 定义了 Hierarchy ID Message 。
- TLP Type 必须为 MsgD 。
- 每个 Message 必须包含 4-DW 的数据负载。
- Length 字段必须为 4 。
- TC[2:0]字段必须为 000b 。
- Attr[2:0]为保留字段。
- Tag 字段为保留字段。
- Subtype 字段必须为 01b 。
- Message Routing 字段必须为 011b - 即 RC 发起的广播。

接收者可以选择检查是否违反了这些规则（但不能检查保留位）。 这些检查是独立可选的（请参见 Section 6.2.3.4）。如果执行这些检查的接收方确定 TLP 违反了这些规则，则该 TLP 是一个 Malformed TLP。

- 如果执行此检查，则这是与接收端口相关的报告错误（请参阅 Section 6.2）。

每个 Hierarchy ID Message 的有效负载包含 System GUID 的低 128 位。

有关 Hierarchy ID、GUID Authority ID 和 System GUID 字段的详细信息，请参见 Section 6.26。

Table 2-30 Hierarchy ID Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
Hierarchy ID Message	0111 1111	011	t	r	tr		Hierarchy ID

公众号：芯片业者相承双

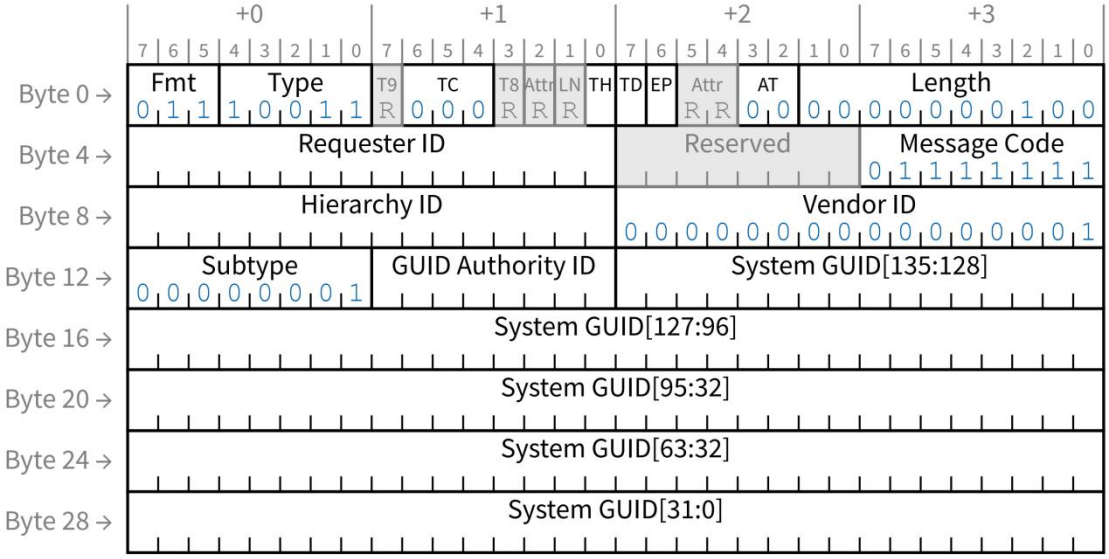


Figure 2-33 Hierarchy ID Message

2.2.8.7 Ignored Messages

Table 2-31 中列出的消息以前用于不再支持的机制（热插拔信令）。强烈建议发送者不要发送这些消息，但是如果实现了这些消息的发送，则它必须符合本规范 1.0a 版本的要求。

强烈建议接收者忽略这些消息的接收，但允许接收者按照本规范 1.0a 版的要求处理这些消息。

Table 2-31 中列出的被忽略的消息由接收方处理如下：

- 物理层和数据链路层必须像处理任何其他 TLP 一样处理这些消息。
- 事务层必须考虑流量控制，但不能对这些消息采取任何其他措施。

Table 2-31 Ignored Messages

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
			RC	Ep	Sw	Br	
Ignored Message	0100 0001	100					
Ignored Message	0100 0011	100					
Ignored Message	0100 0000	100					
Ignored Message	0100 0101	100					
Ignored Message	0100 0111	100					
Ignored Message	0100 0100	100					
Ignored Message	0100 1000	100					

2.2.8.8 Latency Tolerance Reporting (LTR) Message

LTR Message 可选地用于报告有关其读/写服务等待时间容限的设备行为。有关 LTR 的详细信息，请参见 Section 6.18。以下规则

公众号·芯片业者拍承双

适用于 LTR Message 的构造：

- Table 2-32 定义了 LTR Message。
- LTR Message 没有数据负载(the TLP Type is Msg) 。
- Length 字段为保留值。
- LTR Message 使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP。

Table 2-32 LTR Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support[1]				Description/Comments
			RC	Ep	Sw	Br	
LTR	0001 0000	100	r	t	tr		Latency Tolerance Reporting

Note 1：对 LTR 的支持是可选的。支持 LTR 的 Function 必须实现 Chapter 7 中描述的报告和启用机制。

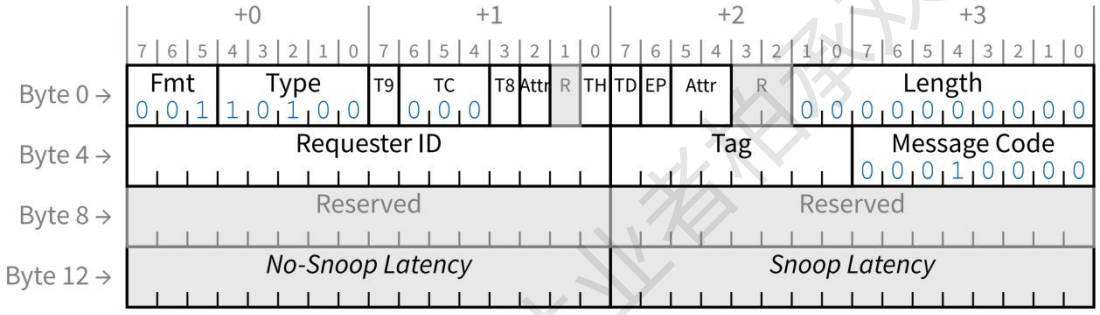


Figure 2-34 LTR Message

2.2.8.9 Optimized Buffer Flush/Fill (OBFF) Message

OBFF Message 可选择用于向 Endpoint 报告平台主要资源状态。该机制在 Section 6.19 中有详细描述。

以下规则适用于 OBFF 消息的构造：

- Table 2-33 定义了 OBFF Message。
- OBFF Message 没有数据负载(the TLP Type is Msg) 。
- Length 字段为保留值。
- Requester ID 必须设置为发送器的端口 ID。
- OBFF Message 使用 TC0。接收者必须检查是否违反此规则。如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为 Malformed TLP。

Table 2-33 OBFF Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support[1]				Description/Comments
			RC	Ep	Sw	Br	
OBFF	0001 0010	100	t	r	tr		Optimized Buffer Flush/Fill

公众号·芯片业者相关

注 1：对 OBFF 的支持是可选的。支持 OBFF 的 Function 必须实现 Chapter 7 中描述的报告和启用机制。

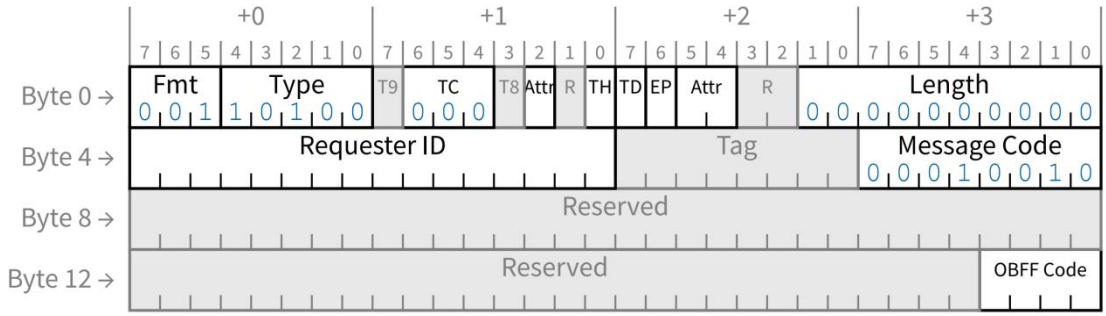


Figure 2-35 OBFF Message

2.2.8.10 Precision Time Measurement (PTM) Messages

Table 2-34 定义了 PTM Message。

- PTM Request Message 和 PTM Response Message 的 TLP 类型为 Msg，并且不能包含有效数据载荷。长度字段处理为保留。
  - Figure 2-36 说明了 PTM Request Message 和 PTM Response Message 的格式。
- PTM ResponseD Message 的 TLP 类型为 MsgD，并且必须在 TLP Header 的 Byte 8 到 Byte 15 中包括一个 64 位的 PTM Master Time 字段和一个包含 32 位的 Propagation Delay 字段的 1 DW 有效数据载荷。
  - Figure 2-37 说明了 PTM ResponseD Message 的格式。
  - 有关如何填充 PTM ResponseD Message 的详细信息，请参见 Section 6.22.3.2。
- 必须将 Requester ID 设置为发送端口的 ID。
- PTM 对话定义为由 PTM Request 和相应的 PTM Response Message 或 PTM ResponseD Message 组成的匹配消息对。
- PTM 消息必须使用默认的业务类别指示符（TC0）。 实现 PTM 的接收者必须检查是否违反此规则。 如果接收方确定 TLP 违反此规则，则它必须将 TLP 视为一个 Malformed TLP。
  - 这是一个与接收端口相关的报告错误（参见 Section 6.2）。

Table 2-34 Precision Time Measurement Messages

Name	TLP Type	Code[7:0] (b)	Routing r[2:0] (b)	Support				Description/Comments
				RC	EP	Sw	Br	
PTM Request	Msg	0101 0010	100	r	t	tr		Initiates PTM dialog
PTM Response	Msg	0101 0011	100	t	r	tr		Completes current PTM dialog - does not carry timing information
PTM ResponseD	MsgD	0101 0011	100	t	r	tr		Completes current PTM dialog - carries timing information

公众号：芯片业者相承双

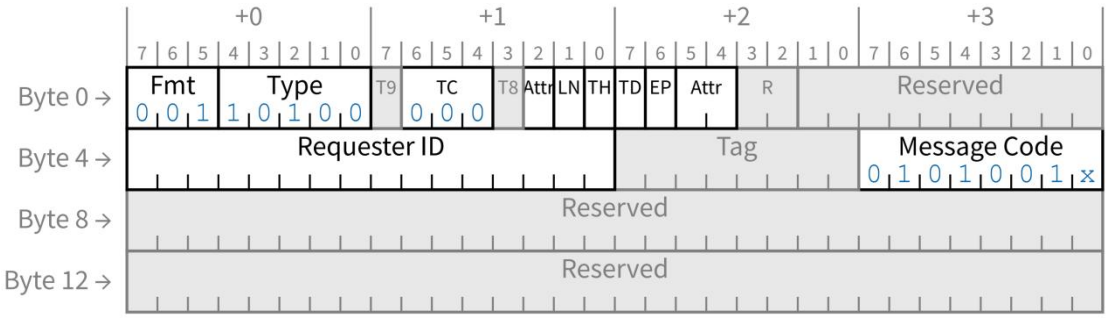


Figure 2-36 PTM Request/Response Message

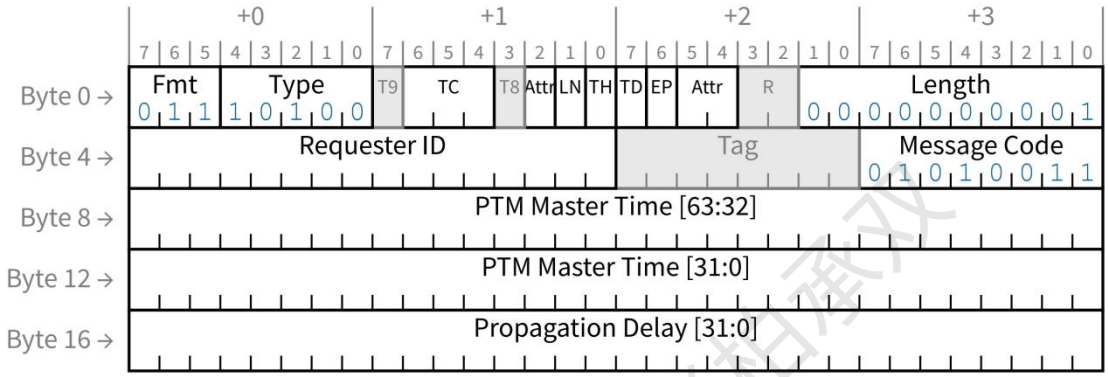


Figure 2-37 PTM ResponseD Message (4 DW header and 1 DW payload)

2.2.9 Completion Rules

所有的 Read、Non-Posted Write 和 AtomicOp Request 都需要返回一个 Completion。Completion 有两种类型：一种带数据负载的，一种不带数据负载的。以下各节定义了 Completion header 中每个字段的规则。

- Completion 使用 ID 路由，使用 3DW 头。
  - 注意，routing ID 字段直接对应相应请求者的 Requester ID 。因此，对于 Completion，这些字段将统称为 Requester ID，而不是用于 ID 路由的不同字段。
- 除了所有 TLP 中包含的 Header 字段和 ID 路由字段之外，Completion 还包含以下其他字段（请参阅 Figure 2-38）：
  - Completer ID[15:0] – Completer 的 ID。
  - Completion Status[2:0] - 见 Table 2-35 的描述。
    - 在 Section 2.3.1 中描述 Completion Status[2:0] 值的规则。
  - BCM – Byte Count Modified：此位不得由 PCI Express Completer 设置，只能由 PCI-X completer 设置。
  - Byte Count[11:0] – 请求的剩余字节数。
    - 字节计数值指定为二进制数，其中 0000 0000 0001b 表示剩余 1 字节，1111 1111 1111b 表示剩余 4095 个字节，而 0000 0000 0000b 表示剩余 4096 个字节。
    - 对于 Memory Read Completion，Byte Count[11:0]要遵循 Section 2.3.1.1 的规则。
    - 对于 AtomicOp Completion，Byte Count 的值等于相关的 AtomicOp 操作数的 size。

公众号·芯片业者相建双

- 对于其他的 Completion，Byte Count 等于 4。

- Tag[9:0] – 结合 Requester ID 字段，一起组成 Transaction ID 。

- Lower Address[6:0] – Completion 起始字节的低字节地址。

  - 对于 Memory Read Completion，此字段中的值是随 Completion 返回的数据的第一个有效字节的地址（请参见 Section 2.3.1.1 中的规则）。

  - 对于 AtomicOp Completion，Lower Address 字段值为保留值。

  - 对于所有其他的 Completion 类型，此字段均设置为全 0。接收者可以选择检查是否违反此规则。有关详细信息，请参见 Section 2.3.2。

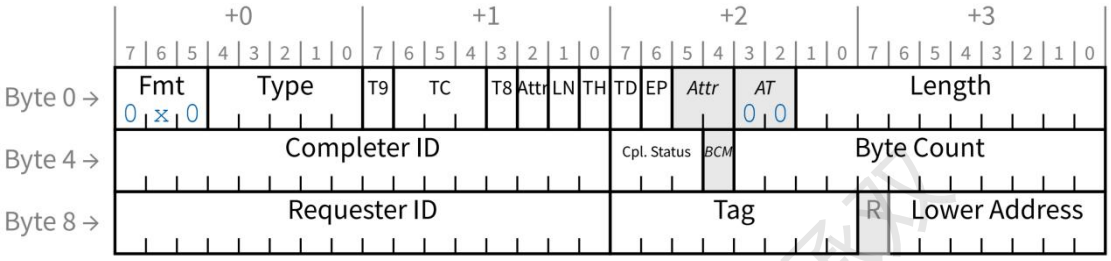


Figure 2-38 Completion Header Format

Table 2-35 Completion Status Field Values

Cpl. Status[2:0] Field Value (b)	Completion Status
000	Successful Completion (SC)
001	Unsupported Request (UR)
010	Configuration Request Retry Status (CRS)
100	Completer Abort (CA)
all others	Reserved

- Completer ID [15:0]是一个 16 位值，对于层次结构中的每个 PCI Express 的 FUnction 都是唯一的（请参见 Figure 2-39 和 Figure 2-40）。

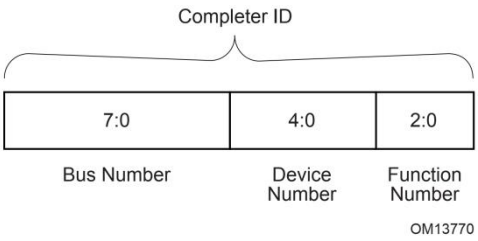


Figure 2-39 (Non-ARI) Completer ID

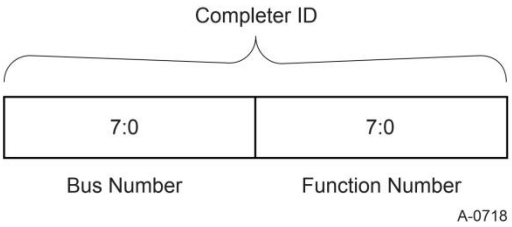


Figure 2-40 ARI Completer ID

- Function 必须能捕获 Type 0 Configuration Write Request 中的 Bus 和 Device Number，并把这些值用于 Completion 的 Completer ID 。

  - 如果 Function 必须在发送设备 Configuration Write Request 之前生成 Completion，则必须在 Bus Number 和 Device Number 字段中输入 0。

- 请注意，Bus Number 和 Device Number 可能会在运行时更改，因此有必要在每个 Configuration Write Request 中重新捕获此信息。
- 例外：可以通过特定于实现的方式将 Bus Number 分配给 RC 中的设备。

注意：对于 ARI 设备，Function 只需要捕获 Bus Number。ARI 设备可以按设备或按功能保留捕获的 Bus Number。参见 Section 2.2.6.2。

注意：一个 ARI 设备的 Completer ID 不包括 Device Number 字段。参见 Section 2.2.4.2。

注意：一个 ARI 设备的 Completer ID 其 Function number 字段为 8 bit。

- 在某些情况下，multi-Function 设备可能会生成具有 UR 状态的 Completion，而不会将该 Completion 与设备内的特定 Function 相关联-在这种情况下，Function Number 字段是保留的。
  - 示例：multi-Function 设备收到的 Read Request 的目标不是与该设备任何 Function 相关的任何资源，该设备会生成具有 UR 状态的 Completion，并在 Completer ID 的 Function Number 字段中将其值设置为全 0。
- Completion header 必须为 Requester ID 提供同样的值，Tag 和 Traffic Class 提供与相应 Request header 中相同的值。
- Completion Header 必须为 Attribute 提供与相应 Request header 中相同的值，除非以下操作明确允许：
  - 当使用了 IDO 时（请参阅 Section 2.2.6.4）。
  - 当在 Translation Completion 中使用了 RO 时（参阅 Section 10.2.3）。
- 如果完成者是 LN Completer（LNC），并且目标存储区支持 registration，则适用以下规则；否则，LN 位必须清零。
  - 如果 Completion Status 为 Successful Completion 并且关联的请求是 LN Read，则必须将 LN 位置 1。
  - 否则 LN 必须设置为 0。
- Completion 的 TH 字段设置为保留值。
- 在完成设备的软件初始化和配置之前（至少使用一个 Configuration Write Request），Completer ID 字段没有意义，对于这种情况，请求者必须忽略 Completer ID 字段中返回的值。
- 包含数据的 Completion 必须指定该 Completion 中返回的实际数据量，并且必须包含指定的数据量。
  - 包含比 Length 字段中指定的更多或更少的数据的 TLP 会形成错误，并且生成的 TLP 是 Malformed TLP 。

注意：这只是一般规则的一个特殊情况，要求 TLP 数据有效载荷长度与 Length 字段中的值匹配。

2.2.10 TLP Prefix Rules

以下规则应用于包含一个 TLP Prefix 的任何 TLP：

- 对于任何 TLP，TLP byte 0 中的 Fmt [2:0]字段值为 100b 表示存在 TLP Prefix，而 Type [4]字段表示 TLP Prefix 的类型。
  - Type [4] = 0b，为一个 Local TLP Prefix 。
  - Type [4] = 0b，为一个 End-End TLP Prefix 。
- TLP 的 byte 1 到 byte 3 的格式根据 TLP Prefix 的类型来定义。
- 包含 TLP Prefix 的 TLP 有一个潜在的 TLP Header。收到的 TLP 如果违反了此规则，则被视为一个 Malformed TLP。相关的接收端口会报一个错误（see Section 6.2）。

公众号：芯片业者相承攻

- 允许一个 TLP 与多个不同类型的 TLP Prefix。
  - 当 TLP 中包含 Local TLP Prefix 和 End-End TLP Prefix 的组合时，要求所有 Local TLP Prefix 出现任何 End-End TLP Prefix 之前。收到的 TLP 如果违反了此规则，则被视为一个 Malformed TLP。相关的接收端口会报一个错误（see Section 6.2）。
- 每个 TLP Prefix 的大小为 1 DW。TLP Prefix 可以重复以提供更多数据。
- 如果 Fmt 和 Type 字段定义了一个 Local TLP Prefix，则按 Local TLP Prefix 的规则处理（see Section 2.2.10.1）。
- 如果 Fmt 和 Type 字段定义了一个 End-End TLP Prefix，则按 End-End TLP Prefix 的规则处理（see Section 2.2.10.2）。

2.2.10.1 Local TLP Prefix Processing

以下规则应用于 Local TLP Prefix：

- Type 字段的子字段 L[3:0]定义了 Local TLP Prefix 的类型。
  - Type[4]必须为 0b。
  - L[3:0]定义如 Table 2-36。

Table 2-36 Local TLP Prefix Types

Local TLP Prefix Type	L[3:0] (b)	Description
MR-IOV	0000	**MR-IOV TLP Prefix** - Refer to [MR-IOV] specification for details.
VendPrefixL0	1110	**Vendor Defined Local TLP Prefix** - Refer to Section 2.2.10.1.1 for further details.
VendPrefixL1	1111	**Vendor Defined Local TLP Prefix** - Refer to Section 2.2.10.1.1 for further details.
		All other encodings are Reserved.

- 不同的 Local TLP Prefix 类型有特定的大小、路由方式和流控机制。
- 接收器收到的 TLP 的 Local TLP Prefix 如果是不支持的类型就会触发一个错误。如果 Extended Fmt Field Supported 字段设置为 1b，则除非另有明确规定，否则违反此规则的 TLP 将被视为 Malformed TLP。相关的接收端口会报一个错误（see Section 6.2）。如果 Extended Fmt Field Supported 字段设置为 0b，则行为是特定于设备的。
- TLP 可能会受 ECRC 保护，但其 Local TLP Prefix 不会受 ECRC 保护。

2.2.10.1.1 Vendor Defined Local TLP Prefix

如 Table 2-36 所述，类型 VendPrefixL0 和 VendPrefixL1 保留用作供应商定义的 Local TLP Prefix。为了最大程度地提高互操作性和灵活性，将以下规则应用于此类前缀：

- 除非已明确启用（使用供应商特定的机制），否则组件不得发送包含 Vendor Defined Local TLP Prefixe 的 TLP。
- 支持任何使用 Vendor Defined Local TLP Prefixe 的组件都必须支持 3bit 的 Fmt 字段的定义，并设置了 Extended Fmt Field Supported 字段（请参见 Section 7.5.3.15）。
- 建议组件是可配置的（使用特定于供应商的机制），以便可以使用两种 Vendor Defined Local TLP Prefixe 编码中的任何一种来发送所有供应商定义的前缀。这样的配置不必是对称的（例如，链路的每一端都可以使用不同的编码来发送相同的前缀）。

公众号：芯片业者的承双

2.2.10.2 End-End TLP Prefix Processing

- Type 字段的子字段 E[3:0]定义了 End-End TLP Prefix 的类型。
  - Type[4]必须为 1b。
  - E[3:0]定义如 Table 2-37。

Table 2-37 End-End TLP Prefix Types

End-End TLP Prefix Type	E[3:0] (b)	Description
TPH	0000	**TPH** - Refer to Section 2.2.7.1 and Section 6.17 for further details.
PASID	0001	**PASID** - Refer to Section 6.20 for further details.
VendPrefixE0	1110	**Vendor Defined End-End TLP Prefix** - Refer to Section 2.2.10.2.1 for further details.
VendPrefixE1	1111	**Vendor Defined End-End TLP Prefix** - Refer to Section 2.2.10.2.1 for further details.
		All other encodings are Reserved.

- 一个 TLP 允许的 End-End TLP Prefix 最大数目为 4。
  - 支持 TLP Prefix 的接收方必须检查此规则。如果接收方确定 TLP 违反此规则，则该 TLP 为 Malformed TLP。这将报告与接收端口相关的错误（请参见 Section 6.2）。
- End-End TLP Prefix 的存在不会改变 TLP 的路由。TLP 是根据 Section 2.2.4 中介绍的路由规则进行路由的。
- Function 通过 Device Capabilities 2 寄存器中的 Max End-End TLP Prefixes 字段指示它们支持多少个 End-End TLP Prefix（请参见 Section 7.5.3.15）。
  - 对于 Root Port，允许 Max End-End TLP Prefixes 字段返回一个值，该值指示支持的 End-End TLP Prefix 少于 Root Port 硬件实际实现的数量。但是，错误处理语义仍然必须基于字段中包含的值。如果收到的 TLP 包含的 End-End TLP Prefix 多于 Root Port 支持的 TLP，则必须按以下方式处理。建议将 Request 作为 Unsupported Request 处理，否则，必须将它们作为 Malformed TLP 处理。建议将 Completion 作为 Unexpected Completion 处理，否则，必须将它们作为 Malformed TLP 处理。对于由 Ingress Port 接收的 TLP，这是与 Ingress Port 相关联的报告错误。对于内部接收的 TLP 要从 Egress Port 发送出去，这将报告与 Egress Port 相关的错误。请参阅 Section 6.2。
  - 对于所有其他 Function，接收到的 TLP 包含的 End-End TLP Prefix 多于 Function 所支持的 TLP，则必须将其视为 Malformed TLP。这将报告的与接收端口相关的错误（请参见 Section 6.2）。

  按照 Section 6.2.4.4 节的规定进行 AER 日志记录（如果支持）。
- 如果 End-End TLP Prefix Supported 字段设置为 1b，则 Switch 必须支持最多 4 个 End-End TLP Prefix 的 TLP 转发。
- End-End TLP Prefix Supported 字段设置为 1b 的各种 Root Port 允许设置不同的 Max End-End TLP Prefixes 值。
- 如果 TLP 支持 ECRC 保护，则其所有 End-End TLP Prefix 也收 ECRC 保护。
- 接收器收到的 TLP 的 End-End TLP Prefix 如果是不支持的类型就会触发一个错误。违反此规则的 TLP 将被视为 Malformed TLP。相关的接收端口会报一个错误（see Section 6.2）。
- 软件应确保不将包含 End-End TLP Prefix 的 TLP 发送到不支持它们的组件。 Extended Fmt Field Supported 字段为 0b 的组件可能会错误解释包含 TLP Prefix 的 TLP。
- 如果上游端口的一个 Function 把 End-End TLP Prefix Supported 字段设置为 1b，则该上游端口的所有 Function 必须处理接收到的请求，如果该请求包含不支持的 End-End TLP Prefix，则该请求作为 Unsupported Request 。相关的接收端口会报一个错

公众号：芯片业者柏承双

误（see Section 6.2）。

- 如果上游端口的一个 Function 把 End-End TLP Prefix Supported 字段设置为 1b，则该上游端口的所有 Function 必须处理接收到的 Completion，如果该 Completion 包含不支持的 End-End TLP Prefix，则该 Completion 作为 Unsupported Completion。相关的接收端口会报一个错误（see Section 6.2）。
- 对于路由元素，每个 Egress Port 中的 End-End TLP Prefix Blocking 字段确定是否可以通过该 Egress Port 发送包含 End-End TLP Prefix 的 TLP（请参见 Section 7.5.3.16）。如果转发被阻止，则整个 TLP 将被丢弃，并报告一个 TLP Prefix Blocked Error。如果被阻止的 TLP 是 Non-Posted Request，则 Egress Port 将返回带有 Unsupport Request 完成状态的 Completion。TLP Prefix Blocked Error 是报告的与 Egress Port 相关的错误（请参见 Section 6.2）。
- 对于启用了多播的路由元素（请参见 Section 6.14）。End-End TLP Prefix 在 TLP 的所有多播副本中复制。 在每个 Egress Port 独立执行多播数据包的 TLP Prefix Egress Blocking 。

2.2.10.2.1 Vendor Defined End-End TLP Prefix

如 Table 2-37 所述，类型 VendPrefixE0 和 VendPrefixE1 保留用作 Vendor Defined End-End TLP Prefix 。为了最大程度地提高互操作性和灵活性，将以下规则应用于此类前缀：

- 除非已明确启用（使用供应商特定的机制），否则组件不得发送包含 Vendor Defined End-End TLP Prefix 的 TLP。
- 建议组件是可配置的（使用特定于供应商的机制），以便可以使用两种 Vendor Defined End-End TLP Prefix 编码中的任何一种来发送所有供应商定义的前缀。这样的配置不必是对称的（例如，链路的每一端都可以使用不同的编码来发送相同的前缀）。

2.2.10.2.2 Root Ports with End-End TLP Prefix Supported

支持在 Root Port 之间包含 End-End TLP Prefix 的 TLP 的对等路由（peer-to-peer routing）是可选的，并且取决于实现。如果 RC 支持两个或多个 Root Port 之间的 End-End TLP Prefix 的路由功能，则必须在每个相关的 Root Port 中通过 Device Capabilities 2 寄存器中的 End-End TLP Prefix Supported 字段指示支持该功能。

不需要 RC 来支持在所有 End-End TLP Prefix Supported 字段设置为 1b 的 Root Port 之间的 End-End TLP Prefix 路由。带有需要在不支持的 Root Port 之间进行路由的带有 End-End TLP Prefix 的请求，必须作为 Unsupported Request 处理。需要在不支持的 Root Port 之间进行路由的带有 End-End TLP Prefix 的 Completion，必须作为 Unexpected Completion 处理。在两种情况下，此错误均由“发送”端口报告。

必须为任何支持通过主机软件或 RC 集成端点发起的带有 End-End TLP Prefix 的 TLP 转发的根端口设置 End-End TLP Prefix Supported 字段。必须为所有支持将在 Ingress Port 上接收到的带有 End-End TLP Prefix 的 TLP 转发到 RCiEP 的根端口设置 End-End TLP Prefix Supported 字段。

设置了 End-End TLP Prefix Supported 字段的各种 Root Port 可以拥有不同的 Max End-End TLP Prefixes 值。

当在根端口之间执行对等路由时，将 TLP 拆分为较小 TLP 的 RC 必须在每个较小 TLP 中复制原始 TLP 的 End-End TLP Prefix（请参阅 Section 1.3.1）。

2.3 Handling of Received TLPs

本节介绍接收到的 TLP 在数据链路层经过完整性验证之后，这些 TLP 在事务处理层时的处理方式。这些规则如 Figure 2-41 所示。

- 接收侧会忽略保留字段。

- 如果 Fmt 字段显示存在至少一个 TLP Prefix ：
  - 通过检查后续 DWORD 的第一个字节中的 Fmt 字段，直到 Fmt 字段与 TLP Prefix 的字段不匹配，来检测 header 中是否存在其他 TLP Prefix。（每个 TLP Prefix 为一个 DWORD 长度。）
  - 根据 Section 2.2.10 的规则来处理 TLP Prefix。
- 如果 Extended Fmt Field Supported 字段设置为 1，则收到的 Fmt 和 Type 字段编码为保留值的 TLP 是 Malformed TLP（请参阅 Table 2-1 和 Table 2-3）。
- 如果 Extended Fmt Field Supported 字段设置为 0，则收到的 Fmt[2]为 1 的 TLP 的处理方式未定义。
- 收到的 Fmt[2]为 0，且 Type 字段为保留值的 TLP 为 Malformed TLP。
- 收到的 TLP 为 Malformed TLP，则该 TLP 被丢弃。
  - 如果接收到的 Malformed TLP 导致不知道释放哪个缓冲区，也不知道怎么映射到未初始化的虚通道，则必须要丢弃且不更新接收侧的流控制信息。
  - 必须丢弃所有其他类型的 Malformed TLP，可以选择不更新接收侧的流控制信息。
- 否则，更新流控制更新逻辑（Section 2.6）。
- 如果 Type 字段中的值指示 TLP 是个 Request，请按照 Section 2.3.1 进行处理；否则，收到的 TLP 是一个 Completion，请按照 Section 2.3.2 进行处理。

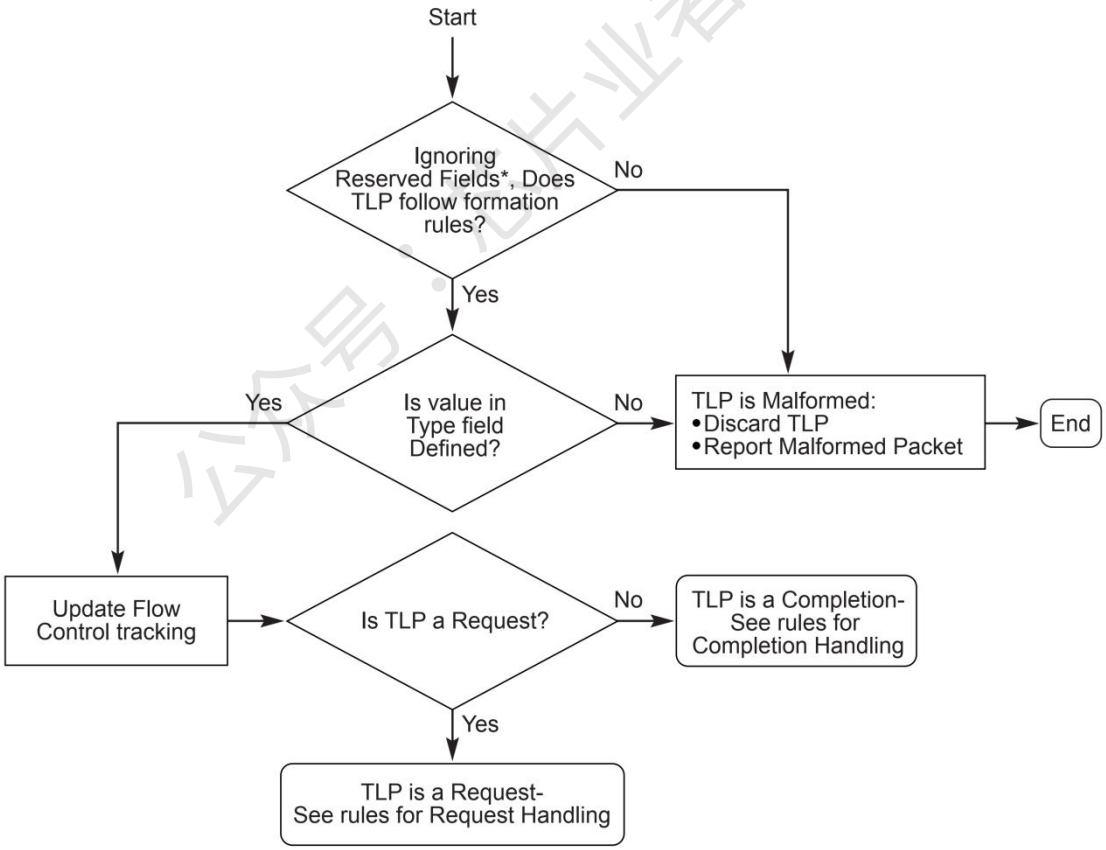


Figure 2-41 Flowchart for Handling of Received TLPs

Switch 必须能处理寻址 Switch 本身资源的 TLP 与寻址 Switch 挂接的设备资源的 TLP。Switch 根据上述规则处理所有寻址 Switch

本身资源的 TLP。根据以下规则处理通过 Switch 的 TLP（请参见 Figure 2-42）：

- 如果收到的 TLP 的 Type 字段显示该 TLP 不是 Msg 或 MsgD 请求，则该 TLP 按照 Section 2.2.4.1 和 Section 2.2.4.2 的路由规则进行处理。
- Switch 中 Completion 的路由使用 Completion 中的 Requester ID 字段来完成。
- 如果收到的 TLP 的 Type 字段显示该 TLP 是一个 Msg 或 MsgD 请求，则该请求的路由规则根据 Type 子字段 r[2:0]来进行。
  - 如果 r[2:0]显示 Msg/MsgD 请求路由到 RC（000b），Switch 就把它路由到该 Switch 的 Upstream Port。
    - 在 Switch 的 Upstream Port 上收到一个 r[2:0]为 000b 的 Msg/MsgD 请求被认为是一个错误。Switch 可能检查是否违反了这个规则——违反该规则的 TLP 是 Malformed TLP。
  - 如果 r[2:0]显示 Msg/MsgD 请求按地址路由（001b），Switch 就按 Memory 请求的规则来路由该 Msg/MsgD。
  - 如果 r[2:0]显示 Msg/MsgD 请求按 ID 路由（010b），Switch 就按 Completion 的规则来路由 Msg/MsgD。
  - 如果 r[2:0]显示 Msg/MsgD 是来自 RC 的广播路由（011b），Switch 就把它路由到该 Switch 的所有 Downstream Port。
    - 在 Switch 的 Downstream Port 上收到一个 r[2:0]为 011b 的 Msg/MsgD 请求被认为是一个错误。Switch 可能检查是否违反了这个规则——违反该规则的 TLP 是 Malformed TLP。
  - 如果 r[2:0]显示 Msg/MsgD 终结于接收侧（100b 或保留值），或者如果 Message Code 字段值已定义且对应于 Switch 必须理解的消息，则 Switch 必须根据消息处理规则处理该消息。
  - 如果 r[2:0]显示 Msg/MsgD 请求路由到 RC（101b）且被收集，参阅 Section 5.3.3.2.1 节的消息处理规则。
  - 除 PME_TO_Ack 以外，收到任何 r[2:0]为 101b 的 Msg/MsgD 请求都被认为是一个错误。在 Switch 的 Upstream Port 上收到一个 PME_TO_Ack 是个错误。Switch 可以选择检查是否违反了这些规则（参阅 Section 6.2.3.4）。如果要做检查，则违反该规则的 TLP 是 Malformed TLP。

公众号·芯片业者相承攻

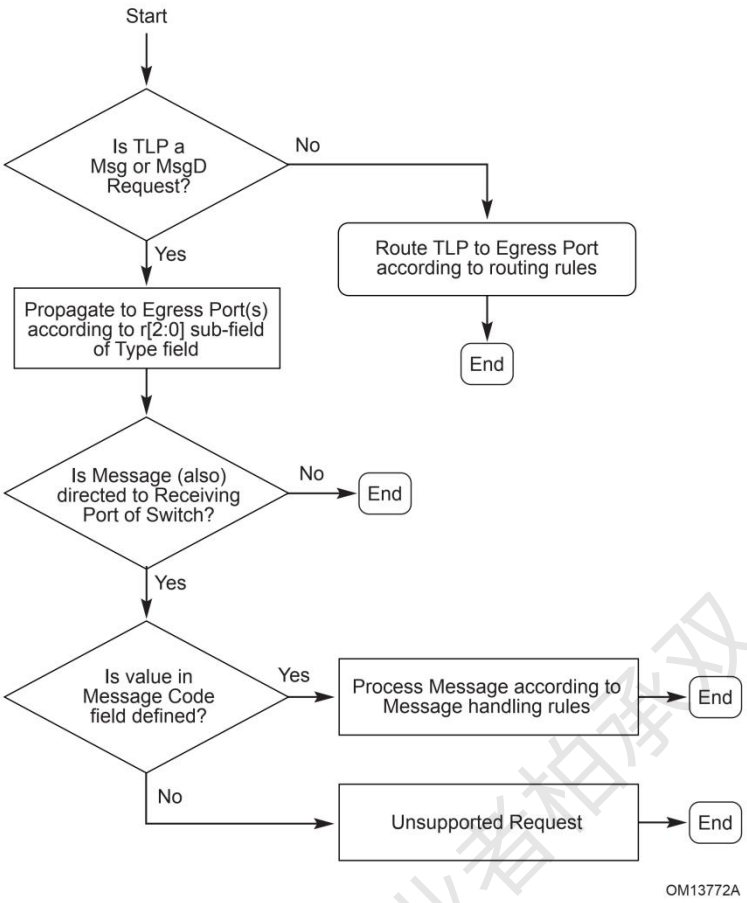


Figure 2-42 Flowchart for Switch Handling of TLPs

2.3.1 Request Handling Rules

本节介绍如何处理接收到的 Request。这些规则如 Figure 2-43 所示。

- 如果 Request 的类型是设备不支持的（因为设计或配置的原因），则该请求被处理为 Unsupported Request，这会根据 Section 6.2 的描述来报告。
  - 如果该请求需要返回 Completion，则返回一个 Completion Status 字段为 UR 的 Completion（see Section 2.2.9）。

IMPLEMENTATION NOTE

When Requests are Terminated Using Unsupported Request

在传统的 PCI 中，设备通过声明 DEVSEL # 在总线上“声明”请求。如果没有设备在设置的时钟数后发出一个请求，则该请求将作为一个 Master Abort 而终止。由于 PCI Express 是点对点互连，它没有等效的机制可以在链路上声明请求，因此一个组件的所有传输总是发送到链路上的另一个组件。因此，请求的接收者有必要确定是否应“声明”该请求。如果未声明该请求，则将其作为 Unsupported Request 处理，这种处理方式使得 PCI Express 与传统的 PCI 的 Master Abort 终结等效。通常，可以通过提出以下问题来确定正确的行为：*Would the device assert DEVSEL# for this request in conventional PCI* ？

对于所有类型的 Endpoin 设备的 Function，回答此问题相对简单。对于 Memory 请求和 I/O 请求，这个问题的确定基于 Function 已编程响应的地址范围。对于 Configuration 请求，Type 0 请求的目标设备从定义上来说就是“目标”，尽管如果设备处理未实现的 Function 时设备依然不会声明配置请求。

对于具有 Type 1 header（Root Port、Switch 和 Bridge）的设备的 Function，通常可以应用相同的问题，但是由于传统 PCI 桥的行为比 Type 0 Function 的行为更为复杂，因此在某种程度上很难确定答案。必须考虑 Root Port 和 Switch Port，就像它们实际上是

公众号：芯片业者相关双

由传统的 PCI 到 PCI 桥组成的一样，然后在每个阶段都要考虑虚拟桥的配置设置，以确定正确的行为。

PCI Express 消息在传统 PCI 中不存在，因此上述规则无法应用。当设备必须将请求作为 Unsupported Request 处理时，本规范专门针对每种消息类型进行了描述。消息通过 Root Port 和 Switch Port 不受传统 PCI 控制机制（包括 Bus Master Enable 和电源状态设置）的影响。

请注意，PCI Express 的 Completer Abort（等同于 Target Abort）仅用于指示严重错误，该错误使 Completer 永久无法响应原本应正常响应的请求。由于仅当目标已声明 DEVSEL # 时才在传统 PCI 中使用 Target Abort，因此对于传统 PCI 目标通过不声明 DEVSEL # 会忽略请求的任何情况，使用 Completer Abort 是不正确的。

- 如果 Request 是一条消息，但消息编码指定了一个未定义的值，或者是一个与设备的 Function 不支持的消息对应的值（除了 Vendor_Defined Type 1 之外，该类型不被视为错误–see Table F-1），则该 Request 是 Unsupported Request，这会根据 Section 6.2 的描述来报告。
  - 如果消息编码是受支持的值，则需按照相应的消息处理规则进行处理；如果消息编码是被忽略的消息，则接收侧忽略该消息且不用报告任何错误（请参阅 Section 2.2.8.7）。
- 如果该请求是一条消息，其路由字段指示按 ID 路由，并且如果该请求是由具有 Type 0 Header 的 device Function 接收的，则强烈建议将该设备视为该消息的目标，而不考虑在该请求的 Destination ID 字段中指定的总线号和设备号。
  - 如果 Destination ID 指定的 Function 是未实现的，强烈建议将该请求作为 Unsupported Request 处理，并按照 Section 6.2 的规定进行报告。
- 如果该请求违反了设备的 Function 的编程模型，则 Function 可以选择将请求视为一个 Completer Abort，而不是正常处理请求。
  - 如果该请求被视为一个 Completer Abort，则会为该 Function 报一个错误（see Section 6.2）。
  - 如果该请求需要一个 Completion，则返回一个 Completion Status 为 CA 的 Completion（see Section 2.2.8.10）。

公众号：芯片业者相承攻

IMPLEMENTATION NOTE

Optimizations Based on Restricted Programming Model

当设备的编程模型限制了（相对于 PCI Express 允许的限制）Request 的特征时，允许该设备“Completer Abort”任何违反编程模型的请求。这包括对寄存器块的 unaligned 访问或 wrong-size 访问以及对存储空间的 unsupported size 的访问。

通常，当所有通信都在设备驱动程序和设备本身之间进行时，设备可以采用受限的编程模型（Restricted Programming Model）。可以由操作系统软件或可能不理解该设备的受限编程模型的应用程序直接访问的设备（通常是实现传统功能的设备）应设计为支持所有类型的请求，这些请求可能存在于现有的使用模型中。如果不这样做，设备可能无法使用现有软件运行。

如果请求在发出 FLR 的到目标 Function 完成 FLR 之间到达，则允许该请求被丢弃（在更新流控制信用之后），而无需将其记录为错误或用信令通知一个错误。建议将该请求作为 Unsupported Request (UR) 处理。

- 否则（是个受支持的请求类型，且不是一个消息），则按以下方式处理：
  - 如果因为设备特定的错误条件，完成者无法处理该请求，则完成者必须尽可能将请求作为 Completer Abort 来处理。
    - 如果可以将错误分离给组件中的特定 Function，则将报告与 Function 接收侧相关的错误；如果无法分离错误，则报告与该接收端口相关的错误（请参见 Section 6.2）。
  - 仅对于配置请求，复位后设备可能会终止请求，但它必须表明设备只是暂时无法处理该请求，在将来需要能够处理该请求——在这种情况下，会返回一个 Completion Status 为 Configuration Request Retry Status (CRS) 的 Completion （请参阅 Section 6.6）。允许设备返回 CRS 的有效复位条件为：
    - Cold、Warm 和 Hot Reset 。
    - FLR。
    - 为响应从 $D3_{hot}$ 到 $D0_{uninitialized}$ 设备状态转换而执行的复位。

    严禁设备的 Function 在设备的软件发起的复位（FLR 除外）之后返回 CRS，例如，通过设备的软件驱动程序写入设备特定的复位字段来实现。另外，在没有中间有效复位（即，FLR 或常规复位）条件的情况下，设备的 Function 在先前返回一个 Successful Completion 之后不允许返回 CRS。
  - 在设备的软件发出复位（FLR 除外）之后，例如通过设备的软件驱动程序写入设备特定的复位字段，则 device Function 不允许返回 CRS。在没有有效复位（即 FLR 或常规复位）条件的情况下，或者如果设置了 Function 的 Status 寄存器中的 Immediate Readiness 字段，则 device Function 在指示其 Configuration-Ready （见 Section 6.23）后不允许返回 CRS。此外，device Function 不允许在没有有效复位（即，FLR 或常规复位）条件的情况下先前返回 Successful Completion 之后返回 CRS。
  - 在为请求提供服务的过程中，完成者可以确定请求是否必须作为错误处理，在这种情况下，将根据错误的类型来处理请求。
    - 一个示例：一个 PCI Express/PCI Bridge 刚开始可能接受请求，因为它指定了一个映射到桥的二级总线的内存空间范围，但是该请求可能会在桥的 PCI 侧执行 Master Abort 或 Target Abort。从 PCI Express 角度来看，在这种情况下，请求的状态为 UR（for Master Abort）或 CA（for Target Abort）。如果请求要求在 PCI Express 上完成，则返回相应的 Completion Status。
- 如果该请求需要返回一个 Completion，则根据 Completion 的构成规则生成该 Completion（请参阅 Section 2.2.9）。
  - Completion Status 根据处理收到的 Request 的结果来确定。
  - 如果请求有 ECRC Check Failed 错误，则是否返回完成取决于特定于实现，如果是，则返回用于完成状态的架构值。但是，强烈建议完成者返回带有 UR 完成状态的完成。

公众号：芯片业者相承双

- 在正常工作条件下，PCI Express Endpoint 和 Legacy Endpoint 接收 Posted Request 的延迟绝不能超过 10 µs，这称为 Posted Request Acceptance Limit。设备必须（a）设计为在必要的时限内处理收到的 Posted Request 并返回相关的流控制信用，或（b）依靠受限的编程模型来确保 Posted Request 不会通过以下方式发送到设备：软件或设备无法在必要的时间内接受新的 Posted Request。
  - 在应用 Posted Request Acceptance Limit 的情况下，以下情况不被认为是正常的操作条件：
    - Fundamental Reset 后的一段时间（see Section 6.6）。
    - TLP 重传或链路重新训练。
    - 一个或多个被丢弃的 FCP（Flow Control Update）。
    - 设备处于诊断模式。
    - 设备处于一个设备特定的无法正常工作的模式。
  - 以下情况被认为是正常的操作条件，但它们引起的任何延迟均不会计入 Posted Request Acceptance Limit：
    - 上行的 TLP 业务延迟了上行的 FCP。
    - 链路从一个低功耗状态退出。
    - VC 在做业务冲裁。
    - 尽管不是必需条件，但强烈建议 RC Integrated Endpoint 也遵守 Posted Request Acceptance Limit。
- 如果设备支持作为 I/O 写入的目标（即 Non-Posted Request），则强烈建议采用与 Posted Request 相同的时限返回每个对应的 Completion，尽管这不是必需的。

IMPLEMENTATION NOTE

Restricted Programming Model for Meeting the Posted Request Acceptance Limit

某些硬件设计可能无法在要求的接收时限内处理每个 Posted Request。一个示例是写入命令队列，其中有的命令的完成所花费的时间可能超过接收时限。当该设备当前正在处理先前的写入时，对该设备的后续写入可能会超过接收时限。这样的设备可能依赖于受限的编程模型，其中设备驱动程序会限制写设备的速度，驱动程序会在发出写事务之前轮询设备以确定缓冲区可用性，或者驱动程序会实现一些其他基于软件的流控机制。

公众号：芯片业者相承双

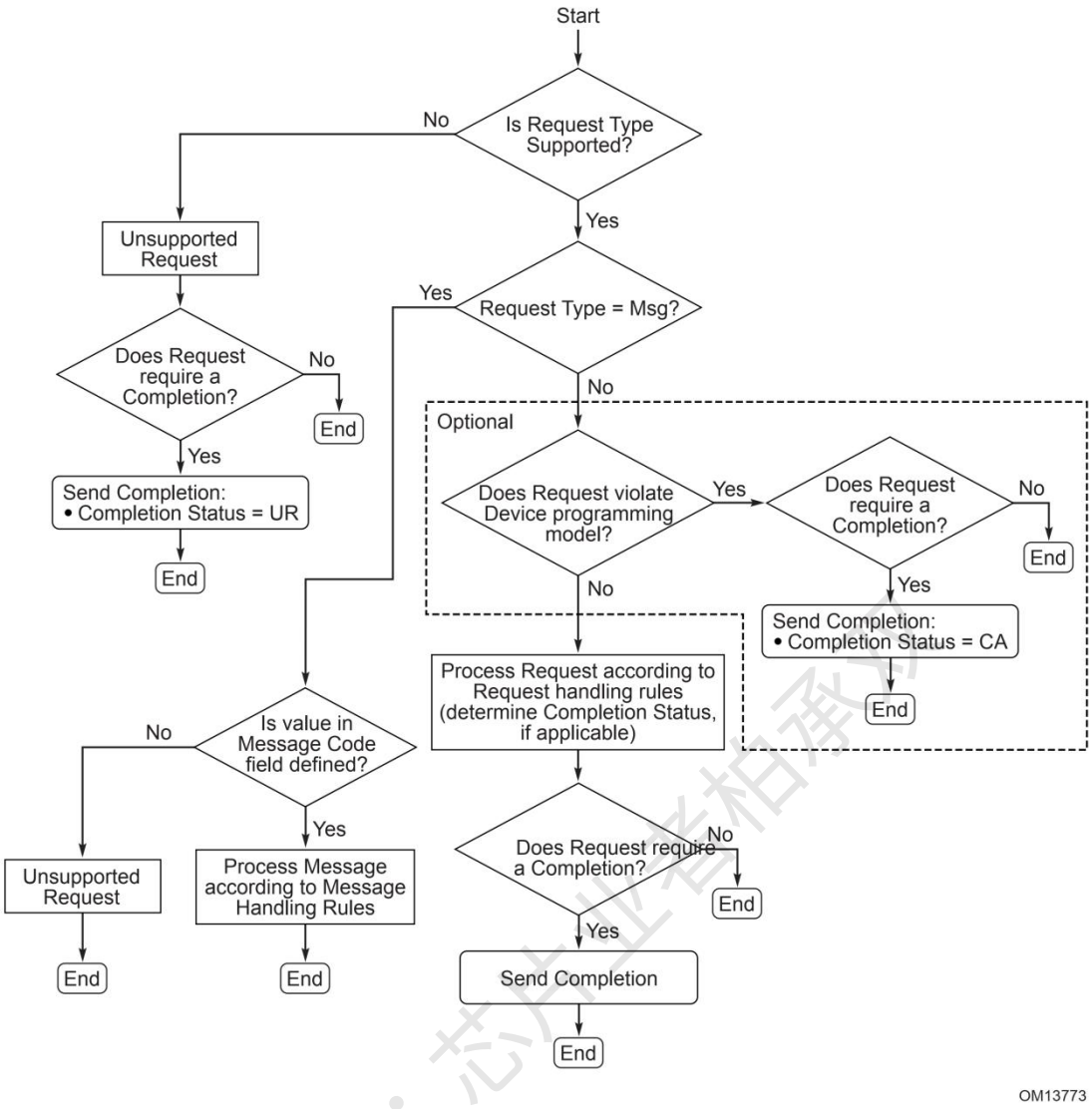


Figure 2-43 Flowchart for Handling of Received Request

IMPLEMENTATION NOTE

Configuration Request Retry Status

某些设备需要很长的自初始化过程才能完成，然后才能处理配置请求（与 PCI 上的智能 I/O 解决方案相同）。 PCI/PCI-X 体系结构已指定复位后的 2^{25}（PCI）或 2^{26}（PCI-X）时钟恢复时间 T_{rhfa}，以提供此类设备所需的自初始化时间。Section 6.6.1 为 PCIe 设备定义了一个 1.0 s 的恢复周期。PCIe 架构也提供了一种可以作为替换的策略，即通过实现 Configuration Request Retry Status (CRS) Completion Status，来等待这种最坏情况下的恢复时间。在有效复位之后接收到配置请求的设备可以以 CRS Completion Status 响应以终结该请求，从而有效地停止配置请求，直到子系统完成本地初始化并准备与主机通信为止。请注意，以 CRS Completion Status 响应配置请求是合法的。发送此 Completion Status 响应任何其他请求类型是非法的（请参阅 Section 2.3.2）。在某些情况下，Readiness Notifications (see Section 6.23) 和 Immediate Readiness (see Section 7.5.1.1.4 和 Section 7.5.2.1) 也禁止使用 CRS Completion Status。

请求者收到具有 CRS Completion Status 的 Completion 会终止发送配置请求。RC 针对最开始的配置请求的进一步措施在 Section 2.3.2 中指定。

实现 CRS 软件可见性的 RC 可以向软件报告收到的 CRS Completion Status，从而使软件可以执行其他任务，而不会在设备完成其自我初始化时停滞。打算利用此机制的软件必须确保在有效的复位之后对设备的首次访问是配置读请求，该请求将访问设备配置空间头中的 Vendor ID 字段的两个字节。仅在这种情况下，RC 将为 Vendor ID 字段合成一个特殊的读取数据值，以向软件指示

设备已返回 CRS Completion Status。对于其他配置请求，或在未启用 CRS 软件可见性的情况下，RC 通常会重新发出配置请求，直到它以 Section 2.3.2 中所述的除 CRS 以外的 Completion Status 为止。

当系统中包含 PCI Express to PCI/PCI-X Bridge 时，系统软件和/或 RC 必须遵守 Section 2.8 和 Section 6.6 所述的为 PCI/PCI-X agent 定义的 T_{rhfa} 的限制，以避免一些不当行为。同样，使用 PCI Express 组件且需要额外的自我初始化时间（超过最小保证时间）的系统，必须提供某种机制来重新发出以 CRS 状态终止的配置请求。在运行基于传统 PCI/PCI-X 的软件的系统中，RC 必须使用硬件机制重新发出配置请求，以确保正确枚举系统。

参阅 Section 6.6 获取更多关于复位的信息。

2.3.1.1 Data Return for Read Requests

- Memory Read Request 的各个 Completion 可能会提供少于该请求的数据量，只要该请求的所有 Completion 携带的数据量在合并后等于该请求的数据量即可。
  - 不同其请求的 Completion 不能合并。
  - I/O 读和配置读只用返回一个 Completion。
  - Completion 的 Completion Status 仅对应于与该 Completion 返回的数据相关的状态。
    - 状态为 Successful Completion 以外的 Completion 会终止单个 Read Request 的所有 Completion。
      - ✧ 这种情况下，Length 字段的值是未定义的，且必须被接收器忽略。
- Completion 携带的数据量不能大于 Max_Payload_Size 指定的值。
  - 接收器必须检查此规则，参阅 Section 2.2。

  Note：这只适用于所有携带有效负载的 TLP。
- Memory Read Request 可以有一或多个 Completion。
- 在用多个 Completion 为 Memory Read Request 提供服务时 Read Completion Boundary（RCB）确定了自然对齐的地址边界。
  - 对于一个 RC，RCB 为 64 byte 或 128 byte。
    - 该值在 Link Control 寄存器中报告（see Section 7.5.3.7）。

    Note：Bridge 和 Endpoint 可以实现一个相应的命令字段，该字段可以由系统软件设置，以指示 RC 的 RCB 值，从而允许 Bridge 或 Endpoint 在 RC 的 RCB 为 128 byte 时优化其行为。
  - 对其他设备类型，RCB 为 128 byte。
- 不以 RCB 字节整数倍跨越自然对齐地址边界的 Request 的 Completion，必须包括请求中指定的所有数据。
- 可以使用多个 Completion 来处理以 RCB 字节整数倍的方式跨越地址边界的请求，但是除沿以下地址边界外，不得对数据进行分段：
  - 第一个 Completion 必须以请求中指定的地址开始，并且必须以以下其中一个结束：
    - 请求中指定的地址加上请求中指定的长度（即整个请求）。
    - 请求的开始和结束之间的地址边界，为 RCB 字节的整数倍。
  - 最后一个 Completion 必须以请求中指定的地址加上请求中指定的长度结尾。

公众号·芯片业者相承双

- 第一个和最后一个 Completion 之间的所有 Completion 必须是 RCB 字节长度的整数倍。

- 接收侧可以选择检查是否违反了 RCB。如果执行此检查的接收者确定 Completion 违反了此规则，则它必须将该 Completion 作为一个 Malformed TLP 处理。
- 一个 Memory Read Request 的多个 Completion 必须以地址递增的方式返回数据。
- 对于每个 Memory Read Request 的 Completion，除非 BCM 位为 1b，否则 Byte Count 字段必须指示完成该请求所需的剩余字节数，包括随该 Completion 返回的字节数。（注：只有 PCI-X 的完成者的 BCM 会设置为 1b。）
  - 按 Table 2-38 所示计算完成 Memory Read Request 所需的字节总数。
  - 如果使用多个 Completion 完成了 Memory Read Request，则每个连续 Completion 的字节计数值是前一个 Completion 指示的值减去前一个 Completion 返回的字节数。
- Completion Data 的范围从请求指定的 DW 地址开始。在第一个或唯一完成的第一个或唯一 Data DW 中，只有在 Request 的 First BE 字段中配置为有效的字节才包含有效数据。在请求的 First BE 字段中配置为无效的字节将返回未定义的内容。
- 在最后一次成功完成的最后一个 Data DW 中，仅在 Request 的 Last BE 字段中配置为有效的字节包含有效数据。在 Request 的 Last BE 字段中配置为无效的字节将返回未定义的内容。
- CRC 运算包括所有完成数据字节，也包括其中内容不确定的字节。
- Figure 2-44 给出了以上示例。该示例假定返回单个 Completion TLP。

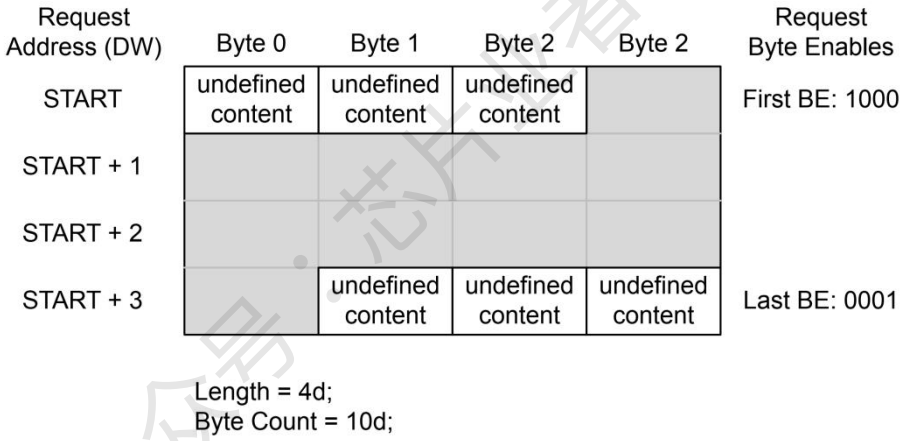


Figure 2-44 Example Completion Data when some Byte Enables are 0b

IMPLEMENTATION NOTE

BCM Bit Usage

为了满足某些 PCI-X 协议约束，在某些情况下，用于 PCI-X 突发读取的 PCI-X Bridge 或 PCI-X Completer 将在拆分的 Completion 序列的第一个 PCI-X 事务中设置 Byte Count 字段，以指示第一个事务的大小，而不是整个突发读取的大小。在这种情况下，PCI-X Bridge/PCI-X Completer 将在第一个 PCI-X 事务中将 BCM 位置 1，以指示 Byte Count 字段已从其正常使用中被修改。有关更多详细信息，请参阅《*PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0*》。

一个 PCI Express 的 Memory Read Requester 必须能够正确处理 PCI-X Bridge/PCI-X Completer 将 BCM 设置为 1 的这种情况。当这种情况发生时，第一个读请求的 Completion 会把 BCM 设置为 1，指示 Byte Count 字段报告的只是该第一个数据包的大小，而不是

整个剩余字节数。在这一点上，请求者不应该得出读完成的其他分组丢失的结论。

BCM 字段永远不会在读完成的后续数据包中设置为 1，因此这些后续数据包中的 Byte Count 字段将始终指示每个 Completion 的剩余字节数。 因此，请求者可以使用这些数据包中的 Byte Count 字段来确定是否缺少读完成的其他数据包。

PCI Express Completer 永远不会将 Byte Count 字段设置为 1。

Table 2-38 Calculating Byte Count from Length and Byte Enables

First DW BE[3:0] (b)	Last DW BE[3:0] (b)	Total Byte Count
1xx1	0000[26]	4
01x1	0000	3
1x10	0000	3
0011	0000	2
0110	0000	2
1100	0000	2
0001	0000	1
0010	0000	1
0100	0000	1
1000	0000	1
0000	0000	1
xxx1	1xxx	Length[27] * 4
xxx1	01xx	(Length * 4) - 1
xxx1	001x	(Length * 4) - 2

公众号·芯片业者相承双

First DW BE[3:0] (b)	Last DW BE[3:0] (b)	Total Byte Count
xxx1	0001	(Length * 4) - 3
xx10	1xxx	(Length * 4) - 1
xx10	01xx	(Length * 4) - 2
xx10	001x	(Length * 4) - 3
xx10	0001	(Length * 4) - 4
x100	1xxx	(Length * 4) - 2
x100	01xx	(Length * 4) - 3
x100	001x	(Length * 4) - 4
x100	0001	(Length * 4) - 5
1000	1xxx	(Length * 4) - 3
1000	01xx	(Length * 4) - 4
1000	001x	(Length * 4) - 5
1000	0001	(Length * 4) - 6

注 26：Last DW BE 为 0000b 只允许 Length 字段为 1DW 的情形。

注 27：长度是 DW 的数量，如 Length 字段中的值所指示，并乘以 4 以得出一个以字节为单位的数字。

- 对所有的 Memory Read Completion，Lower Address 字段必须指示该 Completion 返回的数据的第一个启用字节的字节地址的低位。
  - 对于第一个（或唯一一个）Completion，完成者可以从 Request 的地址的最低有效 5 位与 2 位字节级地址（如 Table 2-33 所示）连接起来，生成此字段。
  - 对于任何后续 Completion，Lower Address 字段将始终为零，但由 RCB 值为 64 byte 的 RC 生成的 Completion 除外。在这种情况下，Lower Address 字段的最低有效 6 位将始终为零，而 Lower Address 字段的最高有效位将根据 64 byte 数据有效载荷的对齐方式进行计算。

Table 2-39 Calculating Lower Address from First DW BE

First DW BE[3:0] (b)	Lower Address[1:0] (b)
0000	00
xxx1	00
xx10	01
x100	10
1000	11

- 当一个 Read Completion 携带一个非 Successful Completion 的 Completion Status：
  - 如果该 Completion 中不携带数据：
    - 用 Cpl 或 CplLk 替代 CplD 或 CplDLk。

- 如果该 Completion 是 Request 的最后一个 Completion：
  - Completer 必须不再为该 Request 返回另外的 Completion。
    - 示例：完成者将请求分为四个部分进行服务；第二个 Completion 携带 Completer Abort 的 Completion Status；完成者终止了对请求的服务，并且没有传输其余的两个完成。
- Byte Count 字段必须指示完成该请求所需的剩余字节数（就像 Completion Status 是 Successful Completion 一样）。
- 如果 Completion Status 为 Successful Completion，则 Lower Address 字段必须指示将随 Completion 返回的数据的第一个启用字节的字节地址的低位比特。

IMPLEMENTATION NOTE

Restricted Programming Model

当设备的编程模型限制了（相对于 PCI Express 所允许的限制）向该设备发出的 Read Request 的大小和/或对齐方式时，允许该设备对违反编程模型的 Read Request 使用 Completer Abort 的 Completion Status。这暗示着这样的设备，通常是所有通信将在设备的驱动程序和设备本身之间进行通信的设备，不一定需要执行生成长度 RCB 的完成所需的缓冲。但是，在所有情况下，对于设备将以 Successful Completion 状态完成的所有读取，都必须遵守 RCB 指定的边界。

Examples：

1. 地址为 1 0000h 且长度为 C0h 字节（十进制 192）的 Memory Read Request 可以由具有 64 byte 的 RCB 值的 RC 来完成，并具有以下 Completion 的组合之一：

   192 –or– 128, 64 –or– 64, 128 –or– 64, 64, 64

2. 地址为 1 0000h 且长度为 C0h 字节（十进制 192）的 Memory Read Request 可以由具有 128 byte 的 RCB 值的 RC 来完成，并具有以下 Completion 的组合之一：

   192 –or– 128, 64

3. 地址为 1 0020h 且长度为 100h 字节的 Memory Read Request 可以由具有 64 byte 的 RCB 值的 RC 来完成，并具有以下 Completion 的组合之一：

   256 –or– 32, 224 –or– 32, 64, 160 –or– 32, 64, 64, 96 –or– 32, 64, 64, 64, 32 –or– 32, 64, 128, 32 –or– 32, 128, 96 –or– 32, 128, 64, 32 –or– 96, 160 –or– 96, 128, 32 –or– 96, 64, 96 –or– 96, 64, 64, 32 –or– 160, 96 –or– 160, 64, 32 –or– 224, 32

4. 地址为 1 0020h 且长度为 100h 字节的 Memory Read Request 可以 Endpoint 来完成，并具有以下 Completion 的组合之一：

   256 –or– 96, 160 –or– 96, 128, 32 –or– 224, 32

2.3.2 Completion Handling Rules

- 设备发出了 outstanding Request，当该设备收到了与这些请求的 Transaction ID 不匹配的 Completion 时，该 Completion 称为“Unexpected Completion”。
- 如果收到的 Completion 与 outstanding Request 的 Transaction ID 相匹配，但其他字段与相应请求不匹配（例如，Attributes、Traffic Class、Byte Count、Lower Address 等），则强烈建议接收方以 Malformed TLP 处理该 Completion。
  - 完成者不得在 Completion 中检查 IDO Attribute (Attribute Bit 2)，因为请求者不需要按照 Section 2.2.6.4 和 Section 2.2.9 的规定将 IDO 值从请求复制到该请求的 Completion 中。
  - 但是，如果以其他方式正确构造了 Completion，则允许接收方将完成作为 Unexpected Completion 进行处理。

公众号：芯片业者相承双

- 当 Switch 的 Ingress Port 收到无法转发的 Completion 时，该 Ingress Port 必须将该 Completion 作为 Unexpected Completion 处理。这包括针对以下目标的 Completion：
  - 与 Upstream Port 关联的设备中不存在的 Function，
  - 与 Upstream Port 关联的总线上不存在的设备，
  - 内部交换结构上不存在的设备或功能，或
  - Downstream Port 的 Bus Number 在 Upstream Port 的 Bus Number 范围内，但该 Downstream Port 未声明。
- 收到 Unexpected Completion 是一个错误，必须根据以下规则进行处理：
  - 收到 Unexpected Completion 的设备必须丢弃该 Completion。
  - Unexpected Completion 会由相关接收端口来报告该错误（see Section 6.2）。

  Note：假定发生 Unexpected Completion 的原因主要是由于 Switch 对 Completion 的错误路由。在这种情况下，发出请求的请求者可能不会收到其请求的完成，并且请求者的完成超时机制（请参见 Section 2.8）将终止请求。
- 具有除 Successful Completion 或 Configuration Request Retry Status（这个只会用来响应配置请求）之外的其他 Completion Status 的 Completion 会导致 Requester 必须执行下述操作：
  - 释放完成缓冲区空间并请求关联的其他资源。
  - 通过请求者专用机制处理该错误（请参见 Section 6.2.3.2.5）。

  如果 Completion 是在启动 FLR 到目标 Function 完成 FLR 的时间之间到达的，则允许将该完成作为 Unexpected Completion 进行处理，或者将其静默丢弃（在更新流控制信用之后），而无需进行日志记录或将其表示为错误。一旦完成 FLR，除非已重新启用该 Function 以发出请求，否则必须将与在 FLR 之前发出的请求相对应的收到完成作为 Unexpected Completion 处理。
- RC 怎么处理携带 Configuration Request Retry Status 的 Completion 是特定于实现的，但系统复位后的时间除外（请参阅 Section 6.6）。对于支持 CRS 软件可见性的 RC，以下规则要遵守：
  - 如果未启用 CRS 软件可见性，则 RC 必须重新发出配置请求作为新请求。
  - 如果启用了 CRS 软件可见性功能，则：
    - 针对设备的 Function 的配置空间头的 Vendor ID 字段的两个字节的配置读请求，RC 必须通过为 Vendor ID 字段返回 0001h 的读取数据值，并且为其余字节返回全 1 来完成该请求并送往主机。此读取数据值已被 PCI-SIG 专门为此保留，并不与任何分配的 Vendor ID 对应。
    - 对于 Configuration Write Request 或任何其他 Configuration Read Request，RC 必须重新发出配置请求作为新请求。

  RC 实现可以选择在确定请求的目标有问题之前采取限制 Configuration Request/ CRS 的 Completion Status 循环的数量，并采取适当的措施，例如将请求作为失败的事务完成发送给主机。

  可以通过 Root Control 寄存器中的 CRS Software Visibility Enable 字段启用 CRS 软件可见性（请参见 Section 7.5.3.12），以基于单个 Root Port 来控制 RC 行为。或者，可以通过 Section 7.9.7.4 所述的 RCRB 控制寄存器中的 CRS Software Visibility Enable 字段来管理 RC 行为，从而允许一个或多个 Root Port 或 RC Integrated Endpoint 的行为由单个字段来控制。对于这种替代情况，每个 Root Port 或 RC Integrated Endpoint 通过 RC Link Declaration Capability 中的 RCRB 头关联，声明其与特定的使能位的关联（请参见 Section 7.9.8）。每个 Root Port 或 RC Integrated Endpoint 最多允许一个使能位进行控制。因此，例如，禁止其 Root Control 寄存器包含使能字段的根端口声明与 RCRB 的 RCRB 头关联，而 RCRB 的 RCRB 头功能中也包含使能位。根端口或 RCRB 头功能中是否存在使能位由相应的 CRS Software Visibility Enable 字段指示（分别参见 Section 7.5.3.13 和 Section

公众号：芯片设者相承效

7.9.7.3）。

- 配置请求以外的请求返回 Configuration Request Retry Status 的 Completion 是非法的。接收者可以选择将这些违规行为报告为 Malformed TLP。
- Completion Status 为保留值的 Completion 被视为 Unsupported Request (UR)。
- Completion Status 为 Unsupported Request 或 Completer Abort 的 Completion 使用传统 PCI 报告机制来报告（see Section 7.5.1.1.4）。
  - 请注意，由此错误条件触发生成的 Completion 由 Completer 报告，如 Section 6.2 所述。
- 如果收到了一个 Completion Status 为非 Successful Completion 的 Read Completion 或 AtomicOp Completion，则：
  - 如果该 Completion 没有携带数据：
    - 用 CpI（或 CpILk）替代 CpID（或 CpIDLk）。
  - 如果该 Completion 是该请求的最后一个 Completion：
    - 请求者必须认为该请求已终止，并且不能期望接收到其他 Completion。

    示例：请求者收到已发出的 128 字节读请求的 32 字节读数据，然后接收到 Completion Status 为 Completer Abort 的 Completion。然后，请求者必须释放为该特定读取请求分配的内部资源。

IMPLEMENTATION NOTE

Read Data Values with UR Completion Status

当配置读请求终止为 Unsupported Request 时，某些系统配置软件依赖于读取全 1 的数据值，尤其是在探查系统中是否存在设备时。当针对配置读请求返回 UR 完成状态时，旨在与软件配合使用的 RC 必须合并此值，该读取数据的值必须为全 1。

2.4 Transaction Ordering

2.4.1 Transaction Ordering Rules

Table 2-40 定义了 PCI Express Transactions 的排序要求。下表中定义的规则统一适用于 PCI Express 上的所有事务类型，包括 Memory、I/O、Configuration 和 Message。该表中定义的排序规则适用于单个业务类别（TC）。 具有不同 TC 值的 Transaction 之间没有排序要求。请注意，这也意味着在流经不同虚通道的业务之间不需要排序，因为不允许将具有相同 TC 值的事务映射到任何 PCI Express 链路上的多个 VC。

对于 Table 2-40，列代表首次发出的 Transaction，行代表随后发出的 Transaction。该表条目指示两个 Transaction 之间的排序关系。表条目定义如下：

注：规范中的"pass"指的是超过的意思，即把顺序提前。那么对于 Table 2-34 的理解需要问这么一个问题：即行代表的 TLP 能否调度到列代表的 TLP 之前传输？

- Yes —— 必须允许第二个事务（行）在第一个事务（列）之前传输，以避免死锁。（发生阻塞时，第二笔事务需要在第一笔事务之前传输。必须理解公平性，以防止饥饿。）
- Y/N —— 没有任何要求。第二笔事务在第一笔事务之前或之后传输都可以，或者被其阻塞。
- No —— 不允许第二笔事务在第一笔事务之前传输。这是支持生产者-消费者强顺序模型所必需的。

公众号：芯片业者相承双

Table 2-40 Ordering Rules Summary

Row Pass Column?		Posted Request (Col 2)	Non-Posted Request		Completion (Col 5)
			Read Request (Col 3)	NPR with Data (Col 4)	
Posted Request (Row A)		a) No b) Y/N	Yes	Yes	a) Y/N b) Yes
Non-Posted Request	Read Request (Row B)	a) No b) Y/N	Y/N	Y/N	Y/N
	NPR with Data (Row C)	a) No b) Y/N	Y/N	Y/N	Y/N
Completion (Row D)		a) No b) Y/N	Yes	Yes	a) Y/N b) No

Table 2-40 中行头和列头信息介绍如下：

Posted Request 指 Memory Write Request 或 Message Request。

Read Request 指 Configuration Read Request、 I/O Read Request 或 Memory Read Request。

NPR (Non-Posted Request) with Data 指 Configuration Write Request、I/O Write Request 或 AtomicOp Request。

Non-Posted Request 指 Read Request 或 NPR with Data。

Table 2-34 中每个条目的解释如下：

A2a 除非 A2b 适用，否则 Posted Request 不得调度到另一个 Posted Request 之前传输。

A2b 设置了 RO 字段的 Posted Request 允许调度到另一个 Posted Request 之前传输。如果两个 Requester ID 不同，则允许设置了 IDO 字段的 Posted Request 调度到另一个 Posted Request 之前传输。

其中 RO 指的是 Relaxed Ordering Attribute 字段。

某些用法通过不实现这种传递来启用（请参见 Section 7.8.15 节中的 No RO-enabled PR-PR Passing 字段）。

A3,A4 Posted Request 允许调度到 Non-Posted Request 之前传输。

A5a Posted Request 是允许调度到 Completion 之前的，但是除非应用了 A5b，否则不这么做。

A5b 在一个 PCI Express to PCI/PCI-X Bridge 中工作在传统 PCI 模式的 PCI/PCI-X 总线，对于从 PCI Express 到 PCI 方向的传输事务，Posted Request 必须要能够调度到 Completion 之前以防止死锁。

B2a 除非应用了 B2b，否则 Read Request 不能调度到 Posted Request 之前传输。

B2b 如果 Read Request 和 Posted Request 的 Requester ID 不一样，则设置了 IDO 字段的 Read Request 可以调度到 Posted Request 之前传输。

公众号·芯片业者相承双

C2a 除非应用了 C2b，否则 NPR with Data 不能调度到 Posted Request 之前传输。

C2b 设置了 RO 字段的 NPR with Data 允许调度到另一个 Posted Request 之前传输。如果两个 Requester ID 不同，则允许设置了 IDO 字段的 NPR with Data 调度到 Posted Request 之前传输。

Note：不是所有的 NPR with Data 都允许把 RO 字段设置为 1b。

B3,B4,C3,C4 一个 Non-Posted Request 允许被调度到另一个 Non-Posted Request 之前传输。

B5,C5 Non-Posted Request 允许调度到 Completion 之前传输。

D2a 除非应用了 D2b，否则 Completion 不能调度到 Posted Request 之前传输。

D2b I/O 或 Configuration Write Completion 允许调度到 Posted Request 之前传输。RO 字段设置为 1b 的 Completion 允许调度到 Posted Request 之前传输。如果 Completion 的 Completion ID 与 Posted Request 的 Requester ID 不同，则该 Completion 允许调度到 Posted Request 之前传输。

Note：并非所有组件都能将 I/O 或 Configuration Write Completion 与其他 Completion 区分开。特别是，不用作关联请求者或完成者的路由元素通常无法进行此区分。除非确定关联的请求类型，否则组件不得将此规则应用于 I/O 或 Configuration Write Completion。

D3,D4 Completion 必须能够调度到 Non-Posted Request 之前传输以避免死锁。

D5a 两个 Completion 拥有不同的 Completion ID 就允许把其中一个调度到另一个之前传输。

D5b 两个 Completion 拥有相同的 Completion ID 就不允许改变顺序。这能保证一个 Memory Read Request 的多个 Completion 可以保持地址增长的顺序。

附加规则：

- PCI Express Switch 允许将 Relaxed Ordering 字段设置为 1 的 Memory Write 或 Message Request 调度到任何之前的朝同一方向传输的 Memory Write 或 Message Request 之前。Switch 转发时不能修改 Relaxed Ordering 字段的值。RC 也被允许以任何顺序将请求中的数据字节写入系统内存（必须将字节写入正确的系统内存地址。他们的写入顺序是未定义的。）。
- 对 RC 和 Switch，Memory Write 合并是禁止的（如 *PCI Local Bus Specification* 描述）。
  - Note：这是必需的，设备可以针对与其期望的大小相匹配的 Memory Write size 优化其接收缓冲区和控制逻辑，而无需支持最大可能的 Memory Write 的 payload size。
- 合并 Memory Read Request，或合并不同 Request 的 Completion 是被禁止的。
- No Snoop 字段不影响顺序。
- 对于 Root Port 和 Switch Downstream Port，Posted Request 或 Completion 的接收不依赖同一业务类别中 Non-Posted Request 的传输 [27]。
- 对于 Endpoint、Bridge 和 Switch Upstream Port，Posted Request 的接收必须不依赖于来自同一业务类别中的同一 Upstream Port 的任何 TLP 的传输 [27]。
- 对于 Endpoint、Bridge 和 Switch Upstream Port，Non-posted Request 的接收必须不依赖于来自同一业务类别中的同一 Upstream

公众号：芯片业者相承双

Port 的 Non-posted Request 的传输 [27]。

- 对于 Endpoint、Bridge 和 Switch Upstream Port，Completion 的接收必须不依赖于来自同一业务类别中的同一 Upstream Port 的任何 TLP 的传输 [27]。
  - 请注意，决不允许 Endpoint 阻塞 Completion。
- 针对 Non-Posted Request 发出的 Completion 必须与该 Non-Posted Request 有相同的业务类别。
- Switch 和支持对等操作的 RC 必须对所有转发的业务强制执行这些事务排序规则。

注 27：满足上述规则是确保无死锁操作的必要但非充分条件。无死锁操作取决于系统拓扑，支持的虚通道数以及配置的业务类别到虚通道的映射。平台和系统约束的规范，以及确保无死锁操作不在本规范的范围内（有关该问题的讨论，请参阅 Appendix D）。

为确保无死锁操作，设备不应将业务从一个虚通道转发到另一虚通道。用于避免设备在虚通道之间转发或转换事务的系统中出现死锁的约束规范不在本文档的讨论范围内（有关问题的讨论，请参阅 Appendix D）。

IMPLEMENTATION NOTE

Large Memory Reads vs. Multiple Smaller Memory Reads

请注意，与 Table 2-34 中的条目 D5b 相关联的规则确保对于一个 Memory Read Request 的多个 Completion 将按地址顺序返回完成内容。但是，与条目 D5a 关联的规则允许不同 Memory Read Request 的不同 Completion 可以以与请求的发送顺序不同的顺序返回。例如，如果设备从位置 1000h 发出一个 256 字节的单个 Memory Read Request，并且使用两个 128 字节的 Completion 返回，则可以确保两个 Completion 按以下顺序返回：

第一个 Completion 携带从地址 1000h 到 107Fh 的数据；

第二个 Completion 携带从地址 1080h 到 10FFh 的数据。

但是，如果设备发出两个 128 字节的 Memory Read Request，第一个针对地址 1000h，第二个针对地址 1080h，则两个完成操作可能以以下任一顺序返回：

第一个 Completion 携带从地址 1000h 到 107Fh 的数据；

第二个 Completion 携带从地址 1080h 到 10FFh 的数据。

or

第一个 Completion 携带从地址 1080h 到 10FFh 的数据；

第二个 Completion 携带从地址 1000h 到 107Fh 的数据。

2.4.2 Update Ordering and Granularity Observed by a Read Transaction

本节讨论一个 Memory 读请求返回的数据写入缓冲区的顺序，以及数据的粒度。

如果使用单个事务的请求者从完成者读取一个数据块，并且正在同时更新完成者的数据缓冲区，则多个更新的顺序和每个读取的返回数据中所反映的更新粒度不在此规范范围之内。 这既适用于 PCI Express 写入事务执行的更新，也适用于其他机制（例如，主机 CPU 更新主机内存）执行的更新。

如果使用单个事务的请求者从完成者读取一个数据块，并且该完成者的数据缓冲区同时由一个或多个不在 PCI Express 结构上的实体更新，则更新的顺序和每个读取返回的数据反映的更新粒度不在本规范范围之内。

作为更新顺序的示例，假设数据块位于主机内存中，并且主机 CPU 首先写入位置 A，然后写入不同的位置 B。一个请求者使用单

公众号：芯片业者相建双

个读取事务读取该数据块不能保证请求者能监测到这个更新顺序。换句话说，无论数据块中位置 A 和 B 的位置如何，请求者都可以观察到位置 B 的更新值和位置 A 的旧值。除非完成者对更新顺序做出自己的保证（在本规范之外），否则依赖更新顺序的请求者必须通过一次读取事务来观察对位置 B 的更新，然后再发起对位置 A 的后续读取以返回其更新的值。

作为更新粒度的示例，如果主机 CPU 将 QWORD 写入主机内存，则从主机内存中读取该 QWORD 的请求者可能会观察到一部分 QWORD 更新，而另一部分包含旧值。

尽管本规范没有要求，但强烈建议主机平台确保当主机 CPU 将对齐的 DWORD 或对齐的 QWORD 写入主机内存时，PCI Express 读取的更新粒度将不小于 DWORD。

IMPLEMENTATION NOTE

No Ordering Required Between Cachelines

允许从主机内存请求多个缓存行的单个内存读取的完成者的 RC，可以同时获取多个缓存行，以帮助促进多缓存行的完成，但要取决于 Max_Payload_Size。这些高速缓存行提取之间不需要排序关系。

2.4.3 Update Ordering and Granularity Provided by a Write Transaction

本节讨论一个 Memory 写请求的数据写入内存的顺序以及粒度。

如果完成者接收了包含多个 DWORD 且 Relaxed Ordering 字段为 0 的单个写事务，则观察到的完成者数据缓冲区中位置更新的顺序必须以地址顺序递增。如果沿路径的 PCI 或 PCI-X 桥将多个写事务组合到单个事务中，则需要使用此语义。但是，观察到的对 Completer 数据缓冲区更新的粒度超出了本规范的范围。

尽管本规范没有要求，但是强烈建议主机平台确保在 PCI Express 写入更新主机内存时，主机 CPU 观察到的更新粒度将不小于 DWORD。

作为更新顺序和粒度的示例，如果请求者将 QWORD 写入主机内存，则在某些情况下，从主机内存读取 QWORD 的主机 CPU 可以观察到更新的第一个 DWORD 以及包含旧值的第二个 DWORD。

2.5 Virtual Channel (VC) Mechanism

虚通道（VC，Virtual Channel）机制为在整个结构中使用 TC 字段区分的业务提供了支持。VC 机制需要独立的硬件资源（队列/缓冲区和关联的控制逻辑）支持。这些资源用于在不同 VC 之间具有完全独立的流控制的情况下通过链路传输信息。在单个业务流可能会对系统内所有业务造成瓶颈的情况下，这是解决流控制引起的阻塞问题的关键。

通过将有特定 TC 值的数据包映射到其对应的 VC，从而将业务与 VC 相关联。VC 和 MFVC 机制允许将 TC 灵活映射到 VC。最简单的形式是 TC 可以按 1:1 映射到 VC。为了权衡性能/成本，PCI Express 提供了将多个 TC 映射到单个 VC 的能力。Section 2.5.2 介绍了 TC 到 VC 映射的详细信息。

当一个或多个 TC 与由 VC ID 指定的物理 VC 资源相关联时，便建立了虚通道。该过程由配置软件控制，如 Section 6.3、Section 7.9.1 和 Section 7.9.2 所述。

默认 TC0/VC0 对的映射是必须实现的、是"硬连线"的，并且所有组件都必须支持，其他的 TC 和 VC 的映射是可选实现的。因此，最基本的 TC/VC 设置不需要任何特定于 VC 的硬件或软件配置。为了确保互操作性，未实现可选的 Virtual Channel Capability structure 或 Multi-Function Virtual Channel Capability structure 的组件必须遵守以下规则：

- Requester 只能生成带有 TC0 标签的请求。（请注意，如果请求者以 TC0 以外的 TC 标签发起请求，则该请求可能被链路另一端实现 VC capability 并执行 TC 过滤的组件视为格式错误的 TLP。）

- 完成者必须能接收 TC 标签不是 TC0 的请求，并且必须保留 TC 标签，即，它生成的任何完成都必须有与请求相同的 TC 标签。
- Switch 必须将所有 TC 映射到 VC0，并且必须转发所有事务，而不管 TC 标签是什么。

如果设备的某个 Function 能够生成 TC 标签不是 TC0 的请求，则即使它只支持默认 VC，该 Function 也必须实现合适的 VC 或 MFVC Capability structure。比如说 Endpoint 或 Root Port 的 Function 类型。为了启用默认以外的 TC 映射，这是必需的。而且这种操作必须遵循 TC/VC 映射规则用 VC 和 MFVC Capability structure 的软件编程实现。

Figure 2-45 说明了虚通道的概念。从概念上讲，流经 VC 的业务在发送端多路复用到公共物理链路资源上，并在接收端多路分解成单独的 VC 路径。

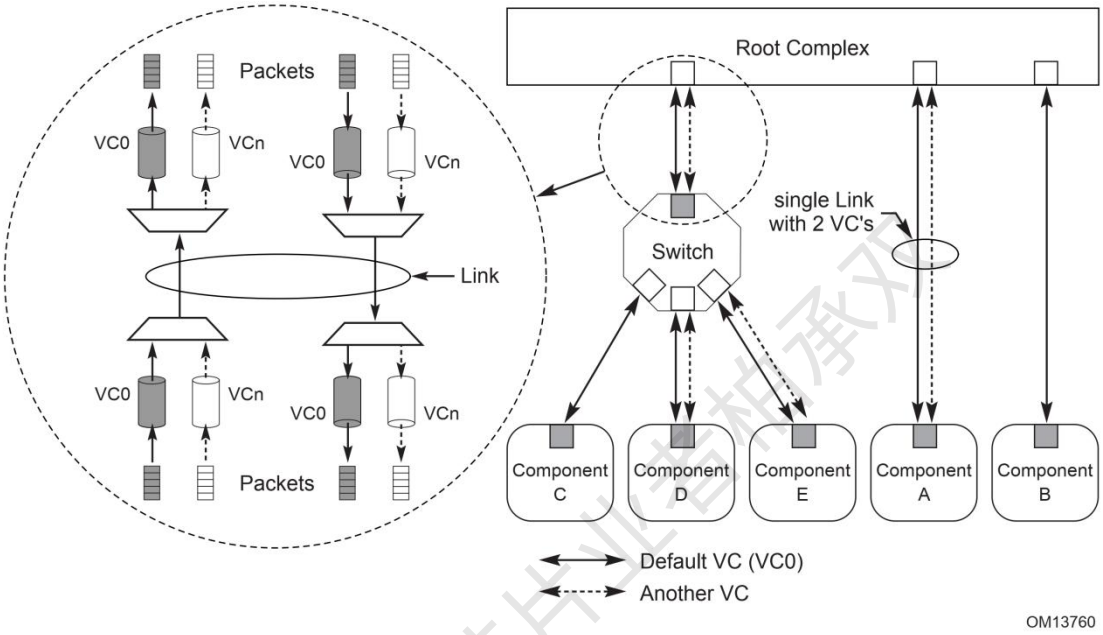


Figure 2-45 Virtual Channel Concept - An Illustration

在 Switch 内部，每个虚通道都需要专用的物理资源（队列/缓冲区和控制逻辑），以支持 Switch 内部独立的业务流。Figure 2-46 从概念上显示了 Switch 中支持上游方向业务流的 VC 资源。

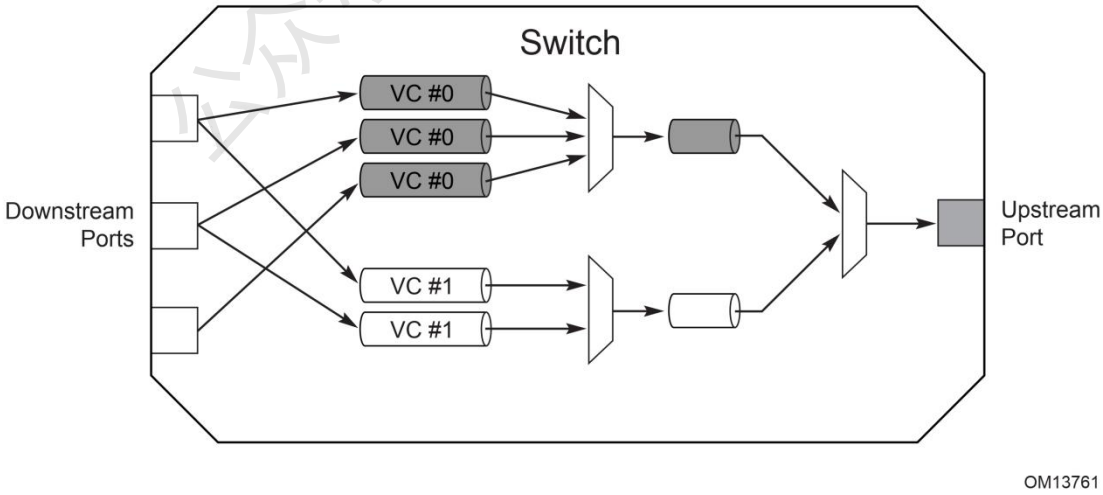


Figure 2-46 Virtual Channel Concept - Switch Internals (Upstream Flow)

multi-Function 设备可以实现类似于 Switch 中那些资源的子集的虚通道资源，目的是管理从不同 Function 到设备 Upstream Egress Port 的上行请求的 QoS。

IMPLEMENTATION NOTE

VC and VC Buffering Considerations

每个架构 VC 最小的缓冲区大小是特定于实现的。

给定架构的链路上的所有 VC 的缓冲区最小值不需要相同，即，具体实现可以根据使用模型和其他链路属性（例如，链路宽度和信令）为选定的 VC 提供合适的缓冲深度。

实现可以从配置和 VC 是否启用推导出特定于实现的策略来调整每个 VC 的缓冲，例如，如果实现了四个 VC 但仅启用了两个，则实现可以将未启用的 VC 缓冲分配给启用的 VC，这样可以降低链路流控引起的反压的可能性，提高系统的效率/性能。

多端口组件（Switch 或 RC）的每个 Port 的 VC 数量和每个 VC 的缓冲区大小可以不一样。

2.5.1 Virtual Channel Identification (VC ID)

PCI Express 的端口可以支持 1-8 个虚通道——每个端口都是独立配置/管理的，因此允许实现根据需求更改每个端口支持的 VC 数量。这些 VC 使用虚通道标识（VC ID）机制进行唯一标识。

请注意，尽管 DLLP 包含用于流控制计数的 VC ID 信息，但 TLP 却不包含。如 Section 2.5.2 所述，使用 TC 到 VC 的映射在链路的每个端口上完成 TLP 与 VC ID 的关联，以进行流控制计数。

根据 Section 7.9.1 中的定义，所有支持不止 VC0 的端口必须至少提供一个 VC Capability structure。允许 multi-Function 设备实现 Section 7.9.2 中定义的 MFVC Capability structure。对于仅支持默认 TC0/VC0 配置的端口，提供这些扩展结构是可选的。配置软件负责为这些 VC 配置链路两侧的端口。这是通过扫描层次结构并使用端口的 VC 或 MFVC Capability 寄存器（支持更多的默认 VC0）来为链路建立对应 VC 数量来实现的。为端口内的 VC 硬件资源分配 VC ID 的规则如下：

- 每个端口的 VC ID 分配必须唯一 —— 即无法将同一 VC ID 分配给同一端口内的不同 VC 硬件资源。
- 链路两侧的两个端口的 VC ID 分配必须相同（VC 数量以及 ID 都要匹配）。
- 如果 multi-Function 设备实现 MFVC Capability structure，则其 VC 硬件资源与其 Function 的任何 VC Capability structure 的 VC 硬件资源是不同的。VC ID 唯一性要求（上述第一条规则）仍单独适用于 MVFC 和任何 VC 功能结构。另外，VC ID cross-Link 的匹配要求（上述第二条规则）适用于 MFVC Capability structure，但不适用于 Function 的 VC Capability structure。
- VC ID 0 固定分配给默认 VC。

2.5.2 TC to VC Mapping

所有的业务都必须要映射到某个 VC 上。TC0 到 VC0 的映射是默认的。

把 TC 映射到非 VC0 是需要系统软件指定的。但是，映射算法必须服从以下规则：

- 一个或多个 TC 可以映射到一个 VC。
- 一个 TC 不能映射到任意 Port 或 Endpoint Function 的多个 VC 上。
- 链路两侧的 Port 上的 TC/VC 映射必须一致。

Table 2-41 给出了一个 TC/VC 映射示例。

公众号：芯片业者相承双

Table 2-41 TC to VC Mapping Example

Supported VC Configurations	TC/VC Mapping Options
VC0	TC(0-7)/VC0
VC0, VC1	TC(0-6)/VC0, TC7/VC1
VC0-VC3	TC(0-1)/VC0, TC(2-4)/VC1, TC(5-6)/VC2, TC7/VC3
VC0-VC7	TC[0:7]/VC[0:7]

注意下述规范描述习惯：

- TCn/VCk = TCn 映射到 VCk。
- TC(n-m)/VCk = 所有在 n 到 m 范围内的 TC 都映射到相同的 VCk。
- TC[n:m]/VC[n:m] = TCn/VCn，TCn+1/ VCn+1，...，TCm/VCm

Figure 2-47 提供了几种不同链路配置中 TC/VC 映射的说明。 有关 TC/VC 的其他注意事项，请参阅 Section 6.3。

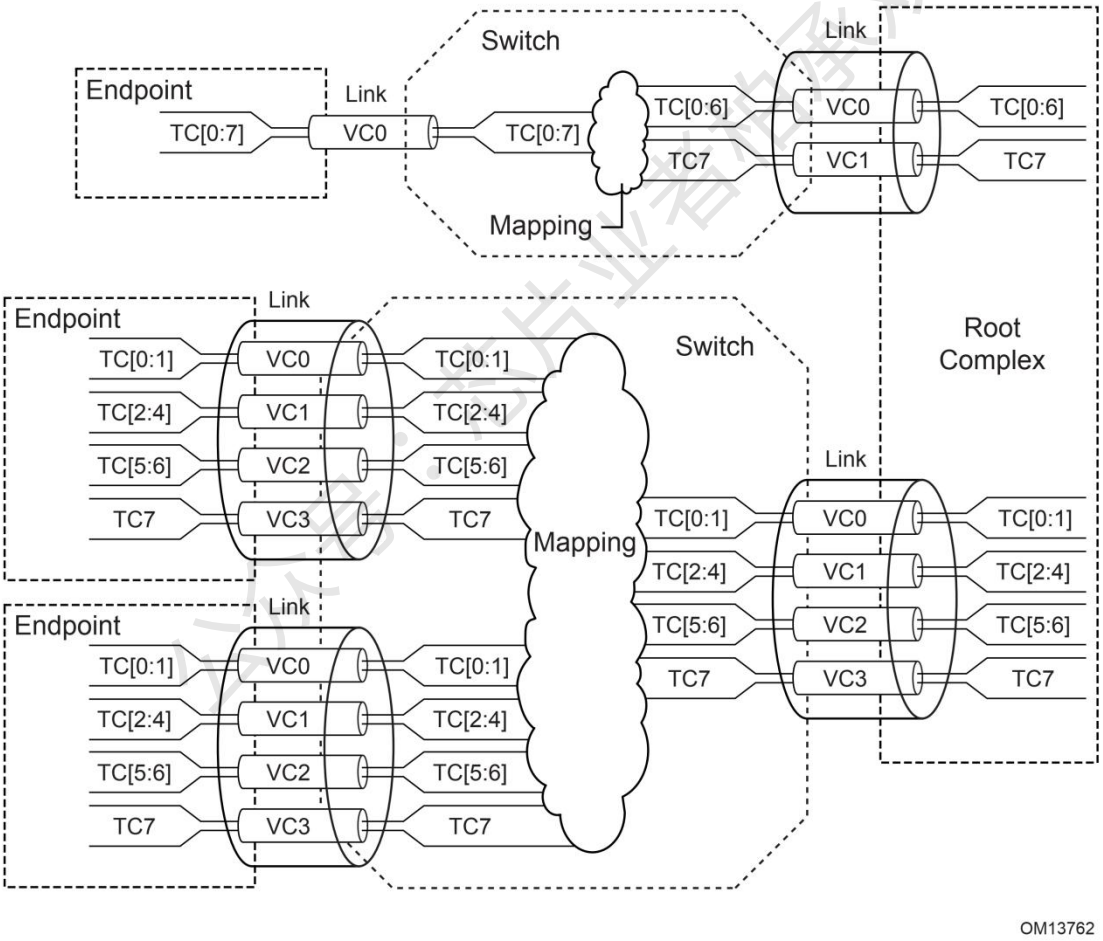


Figure 2-47 An Example of TC/VC Configurations

2.5.3 VC and TC Rules

以下是与 TC/VC 机制相关的关键规则的摘要：

- 所有设备必须支持通用 I/O 业务类别即 TC0，并且必须实现默认的 VC0。

- 每个 VC 拥有独立的流控机制。
- 不同 TC 之间不需要有排序关系。
- 不同 VC 之间不需要有排序关系。
- Switch 的对等传输能力适用于 Switch 支持的所有虚通道。
- multi-Function 设备在不同 Function 之间的对等传输能力适用于 multi-Function 设备支持的所有虚通道。
- 接收设备收到的 TLP 的 TC 没有映射到 Ingress Port 中任何已启用的 VC 上，则此 TLP 视为 Malformed TLP。
- 对于 Switch，TLP 的 TC 没有映射到目标 Egress Port 中任何已启用的 VC 上，则此 TLP 视为 Malformed TLP。
- 对于 Root Port，TLP 的 TC 没有映射到目标 RCRB 中任何已启用的 VC 上，则此 TLP 视为 Malformed TLP。
- 对于有一个 MFVC Capability structure 的 multi-Function 设备，TLP 的 TC 没有映射到 MFVC Capability structure 中任何已启用的 VC 上，则此 TLP 视为 Malformed TLP。
- Switch 的每个 Port 必须支持独立配置 TC/VC 映射。
- RC 的每个 RCRB、关联的根端口和任何 RC Integrated Endpoint 都比支持独立配置 TC/VC 映射。

有关 VC 和 TC 机制的更多详细信息，比如配置、映射和仲裁，请参阅 Section 6.3。

2.6 Ordering and Receive Buffer Flow Control

流控（FC，Flow Control）用于防止接收器缓冲区溢出并使其符合 Section 2.4 中定义的排序规则。请注意，请求者使用流控机制来跟踪链路上设备中可用的队列/缓冲区空间，如 Figure 2-48 所示。也就是说，流控是点对点的，而不是端对端的。流控并不意味着请求已到达其最终完成者。

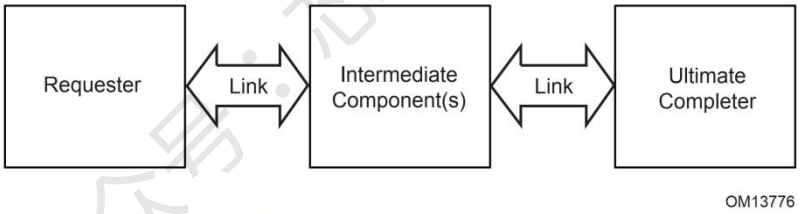


Figure 2-48 Relationship Between Requester and Ultimate Completer

流控与用于在发送器和接收器之间实现可靠的信息交换的数据完整性机制正交。流控可以将 TLP 信息从发送器流到接收器的过程视为完美，因为数据完整性机制可确保通过重传纠正已损坏和丢失的 TLP（请参阅 Section 3.6）。

每个虚通道维护一个独立的流控制信用池。FC 信息是使用 DLLP 在链路两端传递的。DLLP 的 VC ID 字段用于承载适当的流控制信用数目信息所需的虚通道标识。

Multi-Function 设备内使用的流控制机制不在本规范范围内。

流控制由事务层与数据链路层合作处理。事务层为接收的 TLP 执行流控制计数，并根据可用的传输信用选择是否阻塞 TLP 传输。

注意：流控是事务层的功能，因此，在接口上传输的以下类型的信息与流控制信用无关：LCRC、数据包成帧符号、其他特殊符号以及数据链路层之间的通信数据包。这种事实的含义是，接收者必须按照它们到达的速率来处理这些类型的信息（本规范中明确指出的除外）。

另外，从事务层传输到数据链路层和物理层的所有 TLP 必须首先经过流控制选通逻辑（Flow Control“gate”）处理。因此，如果数据链路层由于链路上的错误而反复发送 TLP，则传输和接收的流控机制才不用去关心。

公众号：芯片业者相承双

2.6.1 Flow Control Rules

在本说明书的此部分和其他部分中，“registers”是一个概念性的定义，用于描述一些规则的时候设备可以实现一些相关的逻辑。该描述不暗含或不需要特定的实现，仅用于阐明要求。

- 流控制信息是使用 Flow Control Packets (FCP)传输的，它是 DLLP 的一种（请参阅 Section 3.5）。
- 对于数据负载来说，流控制信用的计算单位为 4DW。
- 对于报头来说：
  - 不支持 TLP Prefix 的接收器的流控制信用的计算单位是一个最大的报头的 size 和 TLP Digest 的 size 的总和。
  - 支持 End-End TLP Prefix 的接收器的流控制信用的计算单位是一个最大的报头的 size、TLP Digest 的 size 和 TLP 中允许的最大 End-End TLP Prefix 的 size 的总和。
  - 支持本地 TLP Prefix 的接收器的流控制的管理取决于本地 TLP Prefix 的类型。
- 每个 VC 有自己单独的流控。
- 流控制要分三类 TLP 进行（注意与排序规则之间的关系 - 参阅 Section 2.4）：
  - Posted Requests (P) —— Messages and Memory Writes
  - Non-Posted Requests (NP) —— All Reads, I/O Writes, Configuration Writes, and AtomicOps
  - Completions (CPL) —— Associated with corresponding NP Requests
- 此外，流控制在三种类型的每一种中区分以下类型的 TLP 信息：
  - Headers (H)
  - Data (D)
- 因此，流控制为每个虚通道跟踪的信息有六种，如 Table 2-42 所示。

Table 2-42 Flow Control Credit Types

Credit Type	Applies to This Type of TLP Information
PH	Posted Request headers
PD	Posted Request Data payload
NPH	Non-Posted Request headers
NPD	Non-Posted Request Data payload
CplH	Completion headers
CplD	Completion Data payload

- 不同 TLP 消耗的流控制信用如 Table 2-43 所示。

公众号·芯片业者相承双

Table 2-43 TLP Flow Control Credit Consumption

TLP	Credit Consumed[38]
Memory, I/O, Configuration Read Request	1 NPH unit
Memory Write Request	1 PH + n PD units[39]
I/O, Configuration Write Request	1 NPH + 1 NPD Note: size of data written is never more than 1 (aligned) DW
AtomicOp Request	1 NPH + n NPD units
Message Requests without data	1 PH unit
Message Requests with data	1 PH + n PD units
Memory Read Completion	1 CplH + n CplD units
I/O, Configuration Read Completions	1 CplH unit + 1 CplD unit
I/O, Configuration Write Completions	1 CplH unit
AtomicOp Completion	1 CplH unit + 1 CplD unit Note: size of data returned is never more than 4 (aligned) DWs.

注 38：每个 header 的信用表示接受 TLP Digest 和相应 TLP 的能力。

注 39：对于所有出现"n"的情况，n = Roundup(Length/FC unit size) 。

- 组件必须为该组件支持的所有虚通道实现独立的流控制。
- 只有硬件默认虚通道（VC0）的流控制会自主初始化。
  - 在复位后数据链路层处于 DL_Init 状态时才初始化 VC0（请参见 Section 3.2 和 Section 3.4）。
- 当通过软件启用其他虚通道时，每个新启用的 VC 都将遵循流控制初始化协议（请参阅 Section 3.3）。
  - 软件通过设置链路上两个组件的 VC Enable 字段来启用虚通道（请参阅 Section 7.9.1 和 Section 7.9.2）。

  注意：多个 VC 可能同时遵循 Flow Control 初始化协议–每个 VC 都作为独立过程遵循初始化协议。
- 软件通过清除链路上两个组件中虚通道的 VC Enable 字段来禁用该虚通道。
  - 禁用组件的虚通道会复位该组件中该虚通道的流控制跟踪机制。
- InitFC1 InitFC2 FCP DLLP 只用于流控制初始化（参阅 Section 3.4）。
- 指定禁用的虚通道的 InitFC1，InitFC2 或 UpdateFC FCP 会因无效而被丢弃。
- 在任何虚通道的 FC 初始化期间，包括作为链路初始化的一部分初始化的默认 VC，接收方必须首先发送等于或大于 Table 2-44 中所示值的 VC 信用值。
  - 如果不支持 Scaled Flow Control，或支持但未激活，请使用 Scale Factor 1 列中的值。
- 如果支持并激活了 Scaled Flow Control，请将该列中的值用于与该信用类型关联的比例因子（请参见 Section 3.4.2）。

公众号：芯片业者怕承双

Table 2-44 Minimum Initial Flow Control Advertisements[40]

Credit Type	Minimum Advertisement		
	No Scaling or Scale Factor 1	Scale Factor 4	Scale Factor 16
PH	1 unit - credit value of 01h.	4 Units - credit value of 01h.	16 Units - credit value of 01h.
PD	Largest possible setting of the Max_Payload_Size for the component divided by FC Unit Size. For an MFD, this includes all Functions in the device. Example: If the largest Max_Payload_Size value supported is 1024 bytes, the smallest permitted initial credit value would be 040h.	Ceiling(Largest Max_Payload_Size / (FC Unit Size * 4)) + 1. For an MFD, this includes all Functions in the device. Example: If the largest Max_Payload_Size value supported is 1024 bytes, the smallest permitted initial credit value would be 011h.	Ceiling(Largest Max_Payload_Size / (FC Unit Size * 16)) + 1. For an MFD, this includes all Functions in the device. Example: If the largest Max_Payload_Size value supported is 1024 bytes, the smallest permitted initial credit value would be 005h.
NPH	1 unit - credit value of 01h.	4 Units - credit value of 01h.	16 Units - credit value of 01h.
NPD	Receiver that supports AtomicOp routing capability or any AtomicOp Completer capability: 2 units - credit value of 002h All other Receivers: 1 unit - credit value of 001h.	Receiver that supports AtomicOp routing capability or any AtomicOp Completer capability: 8 units - credit value of 002h All other Receivers: 4 units - credit value of 001h.	Receiver that supports AtomicOp routing capability or any AtomicOp Completer capability: 32 units - credit value of 002h All other Receivers: 16 units - credit value of 001h.
CplH	Root Complex (supporting peer-to-peer traffic between all Root Ports) and Switch: 1 FC unit - credit value of 01h Root Complex (not supporting peer-to-peer traffic between all Root Ports) and Endpoint: infinite FC units - initial credit value of all 0s.[41]	Root Complex (supporting peer-to-peer traffic between all Root Ports) and Switch: 4 FC units - credit value of 01h Root Complex (not supporting peer-to-peer traffic between all Root Ports) and Endpoint: infinite FC units - initial credit value of all 0s.[42]	Root Complex (supporting peer-to-peer traffic between all Root Ports) and Switch: 16 FC units - credit value of 01h Root Complex (not supporting peer-to-peer traffic between all Root Ports) and Endpoint: infinite FC units - initial credit value of all 0s.[43]
CplD	Root Complex (supporting peer-to-peer traffic between all Root Ports) and Switch: Largest possible setting of the Max_Payload_Size for the component divided by FC Unit Size.	Root Complex (supporting peer-to-peer traffic between all Root Ports) and Switch: Ceiling(Largest Max_Payload_Size / (FC Unit Size * 4)) + 1.	Root Complex (supporting peer-to-peer traffic between all Root Ports) and Switch: Ceiling(Largest Max_Payload_Size / (FC Unit Size * 16)) + 1.

Credit Type	Minimum Advertisement		
	No Scaling or Scale Factor 1	Scale Factor 4	Scale Factor 16
	Root Complex (not supporting peer-to-peer traffic between all Root Ports) and Endpoint: infinite FC units - initial credit value of all 0s.	Root Complex (not supporting peer-to-peer traffic between all Root Ports) and Endpoint: infinite FC units - initial credit value of all 0s.	Root Complex (not supporting peer-to-peer traffic between all Root Ports) and Endpoint: infinite FC units - initial credit value of all 0s.

注 40：*PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0* 解决了 PCI Express to PCI/PCI-X Bridge 的要求。

注 41：发送器将这个值解释为无穷大，因此它将永不停止的发送。

注 42：发送器将这个值解释为无穷大，因此它将永不停止的发送。

注 43：发送器将这个值解释为无穷大，因此它将永不停止的发送。

- 不支持所有根端口之间的对等通信的 RC，必须通告 infinite Completion credit。
- 支持所有根端口之间的对等业务流的 RC 可以有选择地通告 non-infinite Completion credit。在这种情况下，RC 必须确保避免

死锁，并保持针对 RC 的完成的转发进度。请注意，由于 RC 转发的 Non-Posted request 可能未显式分配完成业务流缓冲区空间（由于暂时缺乏信用），因此可能会暂时造成完成业务流的停顿。

- 对数据负载而言，接收方不得累积发送超过 2047 个 outstanding 的未使用信用给发送器；对于 TLP header，此值不得超过 127 个。支持 Scaled Flow Control 的接收器不得累计向发送器发送超过 Table 3-2 所示最大信用值的未使用数据或报头。
  - 组件可以选择检查是否违反此规则。如果实现此检查的组件确定违反此规则，则该错误是流控制协议错误（FCPE，Flow Control Protocol Error）。
    - 如果实现，这将报告的与接收端口相关的错误（请参阅 Section 6.2）。
- 如果在初始化过程中通告了无限信用（DLLP 中携带 00h 或 000h 值），则初始化后不需要进行流控制更新。
  - 如果发送 UpdateFC DLLP，则信用值字段必须设置为零，并且接收方必须将其忽略。接收方可以选择检查非零更新值（违反此规则）。如果实施此检查的组件确定违反此规则，则该错误是流控制协议错误（FCPE）。
    - 如果实现，这将报告的与接收端口相关的错误（请参阅 Section 6.2）。
- 如果仅仅只对给定类型（P、NP 或 CPL）的数据负载或 header 在初始化期间应用无限信用，则仍然需要传输 Update-FC DLLP，但是与 Data/header（通告为无限信用）相对应的信用字段必须设置为零，并且接收方必须将其忽略。
  - 接收方可以选择检查非零更新值是否违反此规则。如果执行此检查的接收方确定违反此规则，则该违反是流控制协议错误（FCPE）。
    - 如果实现，这将报告的与接收端口相关的错误（请参阅 Section 6.2）。
- 收到的 TLP 使用了未启用的 VC，则该 TLP 是一个 Malformed TLP。
  - VC0 时钟启用。
  - 对于 VC 1-7，当 VC Resource Control 寄存器中的相应 VC Enable 字段已设置为 1b 时，并且该 VC 的 FC 协商已退出 FC_INIT1 状态并进入 FC_INIT2 状态时，就认为该 VC 已使能（请参阅 Section 3.4）。
  - 这将报告与接收端口相关的错误（请参阅 Section 6.2）。
- 在退出 FC_INIT2 状态完成该 VC 的初始化之前，不允许使用任何 VC 0-7 进行 TLP 传输。

对于 VC 1-7，软件必须使用 VC Resource Status 寄存器中的 VC Negotiation Pendin 字段来保证链路两侧在 FC_INIT2 状态退出、链路完成协商之前不使用这些 VC。

以下各节中使用的[**Field Size**]参数在 Table 2-45 中进行了描述（请参见 Section 3.4.2）。

Table 2-45 [Field Size] Values

Scaled Flow Control Supported	HdrScale or DataScale	[Field Size] for PH, NPH, CplH	[Field Size] for PD, NPD, CplD
No	x	8	12
Yes	00b	8	12
Yes	01b	8	12
Yes	10b	10	14
Yes	11b	12	16

公众号：芯片业者相聚

2.6.1.1 FC Information Tracked by Transmitter

- 对于每种要跟踪的信息，要为流控制 TLP 发送器选通逻辑关注以下两个量：
  - **CREDITS_CONSUMED**
    - 该量指自流控制初始化以来，TLP 传输消耗的 FC unit 总数的计数，以模 $2^{[\text{Field Size}]}$计算（其中 PH，NPH 和 CPLH 的[Field Size]为 8；PD，NPD 和 CPLD 的[Field Size]为 12）。
    - 该接口初始化的时候该量设置为 0（链路训练时）。
    - 为每个事务层允许传递给发送器流控制选通的 TLP 更新该量。
      - 更新算法为：

        CREDITS_CONSUMED := (CREDITS_CONSUMED + Increment) mod $2^{[\text{Field Size}]}$

      其中 Increment 是经过发送器选通的 TLP 相应部分的 FC credit 的大小。
  - **CREDIT_LIMIT**
    - 接收端合法通告的最新 FC unit 数量。此数量表示自流控制初始化以来，接收方可用的 FC credit 总数，以模 $2^{[\text{Field Size}]}$计算（其中 PH，NPH 和 CPLH 的[Field Size]为 8；PD，NPD 和 CPLD 的为 12）。
    - 接口初始化时该值是未定的（链路训练时）。
    - 在流控制初始化期间设置该值。
    - 每收到一个 FC update，
      - 如果 CREDIT_LIMIT 不等于更新值，则将 CREDIT_LIMIT 设置为更新值。
- 如果发送端检测到准备发送的 TLP 格式不正确，则强烈建议发送方丢弃该 TLP，并将此情况视为 Uncorrectable Internal Error。
- 如果发送端检测到准备发送的 TLP 似乎格式正确，但 ECRC 是坏的，则强烈建议发送方发送 TLP 并相应地更新其内部流控制信用。
- 发送器选通逻辑（Transmitter gating）必须确定是否已通告足够的信用以允许发送给定的 TLP。如果发送方没有足够的信用来发送 TLP，则它必须阻止 TLP 的传输，可能会使正在使用同一虚通道的其他 TLP 停滞。发送器必须遵循 Section 2.4 中指定的排序和避免死锁规则，该规则要求某些类型的 TLP 必须在阻止其他特定类型的 TLP 时绕过其他特定类型的 TLP。请注意，使用不同虚通道的 TLP 没有排序关系，并且不能相互阻塞。
- 发送器选通逻辑工作过程如下：
  - 对于每种需要 credit 的类型，所需的 credit 数计算如下：

    CUMULATIVE_CREDITS_REQUIRED

    = (CREDITS_CONSUMED + <credit units required for pending TLP>) mod $2^{[\text{Field Size}]}$
  - 除非在流控制初始化期间将 CREDIT_LIMIT 指定为“无限”，否则如果对于 TLP 中的每种信息，都满足以下方程式（使用无符号算法），则允许发送器发送 TLP：

    (CREDIT_LIMIT – CUMULATIVE_CREDITS_REQUIRED) mod $2^{[\text{Field Size}]}$ <= $2^{[\text{Field Size}]}$ / 2
  - 如果在流控制初始化期间将 CREDIT_LIMIT 指定为“无限”，则对于该类型的信用无条件满足发送器选通逻辑。

公众号·芯片业者柏承双

- 请注意，某些类型的事务需要不止一种信用。（例如，Memory Write request 需要同时计算 PH 和 PD 信用。）

- 在对信用的使用和返还计算时，来自不同 TLP 的信息永远不能混入一个信用中。
- 当由于缺少 FC 信用而阻止某些 TLP 进行传输，且需要确定必须允许哪些类型的 TLP 绕过停滞的 TLP 时，发送方必须遵循 Section 2.4 中指定的排序规则。
- 事务的 FC 返还信用额一定不能解释为该事务已完成或已达到系统可见性。
  - 流控制信用返还仅用于接收缓冲区管理，并且代理不得基于流控制信息的返还或缺少返还而对事务的完成状态或系统可见性做出任何判断。
- 当发送器发送无效的 TLP 时，发送器不会为该 TLP 修改 CREDITS_CONSUMED（请参见 Section 3.6.2.1）。

2.6.1.2 FC Information Tracked by Receiver

- 对于每种要跟踪的信息，要为流控制 TLP 接收器计数逻辑关注以下两个量：
  - **CREDITS_ALLOCATED**
    - 初始化之后要通告给发送端的信用总数，以模 $2^{[\text{Field Size}]}$计算（其中 PH，NPH 和 CPLH 的[Field Size]为 8；PD，NPD 和 CPLD 的[Field Size]为 12）。
    - 根据接收端的缓冲区大小和分配策略进行初始设置。
    - 该值在 InitFC 和 UpdateFC DLLP 中发送（请参见 Section 3.5）。
    - 随着接收端事务处理层处理接收到的 TLP 的增加，可以增加可用的接收缓冲区空间。
      - 更新方式如下：

      CREDITS_ALLOCATED := (CREDITS_ALLOCATED + Increment) mod $2^{[\text{Field Size}]}$

      其中 Increment 对应于使可用的信用。
  - **CREDITS_RECEIVED**（可选-用于以下所述的可选错误检查）
    - 自流控制初始化以来收到的有效 TLP 消耗的 FC unit 总数的计数，以模 2 [Field Size]计算（其中 PH，NPH 和 CPLH 的[Field Size]为 8；PD，NPD 和 CPLD 的[Field Size]为 12）。
    - 在接口初始化的时候初始化为 0。
    - 更新方法如下：

      CREDITS_RECEIVED := (CREDITS_ RECEIVED + Increment) mod $2^{[\text{Field Size}]}$

      其中 Increment 对应于使可用的信用。

      对于每个收到的 TLP，前提是 TLP：
      - 通过了数据链路层的完整性检查。
      - 不是一个 malformed TLP；或者是一个 malformed TLP 但导致要释放哪个缓冲区是不明确的，映射到哪个已初始化的虚通道也是不明确的。后一个条件是可选实现的。
      - 消耗的信用不超过分配的信用（请参阅以下规则）。

公众号：芯片业者相承双

- ✧ 对于具有 ECRC Check Failed 错误的 TLP，但在其他方面对于释放哪个缓冲区是明确的，强烈建议更新接收到的 CREDITS_RECEIVED。

- 如果接收方实现了 CREDITS_RECEIVED 计数器，则当接收到无效的 TLP 时，接收方不会为该 TLP 修改 CREDITS_RECEIVED（请参见 Section 3.6.2.1）。

- 接收器可以选择使用无符号算术，通过检查以下公式来检查接收器溢出错误（TLP 数目超过 CREDITS_ALLOCATED）：

  (CREDITS_ALLOCATED - CREDITS_RECEIVED) mod $2^{[\text{Field Size}]}$ >= $2^{[\text{Field Size}]}$ /2

  如果执行了检查并且此等式的评估结果为 true，则接收方必须：

  - 丢弃 TLP，且不修改 CREDITS_RECEIVED。
  - 取消为 TLP 分配需要分配的任何资源。

  如果做此检查，这是报告的与接收端口相关的错误（请参阅 Section 6.2）。

  注意：发生接收方溢出错误后，接收方的行为未定义，但鼓励接收方继续运行，处理流控制更新并接受不超过分配的信用的任何 TLP。

- 对于非无限的 NPH，NPD，PH 和 CPLH 类型，每次发生以下事件序列时，都必须调度 UpdateFC FCP 进行传输：

  a. 当 Scaled Flow Control 未激活且特定类型的可用 FC 信用数为零且 TLP 处理使该类型的一个或多个单元可用时，

  b. 当 Scaled Flow Control 未激活时，NPD 信用降至 2 以下，接收器支持 AtomicOp 路由功能或 128 位 CAS Completer 功能，并且一个或多个 NPD 信用由处理的 TLP 提供，

  c. 当激活 Scaled Flow Control 且特定类型的可用 FC 信用数为零或低于 scaled threshold 时，通过 TLP 处理使该类型的一个或多个单元可用，以便可用信用数等于或大于 scaled threshold（对于 HdrScale 或 Data Scale 为 01b，0，4）对于 10b 的 HdrScale 或 DataScale，16 表示 11b 的 HdrScale 或 DataScale。

  d. 激活 Scaled Flow Control 后，用于 NPD 的 DataScale 为 01b，NPD 信用降到 2 以下，接收器支持 AtomicOp 路由功能或 128 位 CAS Completer 功能，并且 TLP 处理提供一个或多个 NPD 信用。

- 对于非无限的 PD 和 CPLD 类型，当可用信用数少于 Max_Payload_Size 时，每次处理的 TLP 使一个或多个该类型的 unit 可用时，必须调度 UpdateFC FCP 进行传输。

  - 对于 ARI 设备，Max_Payload_Size 仅由 Function 0 中的设置确定。其他 Function 中的 Max_Payload_Size 设置将被忽略。
  - 对于在所有 Function 上 Max_Payload_Size 设置相同的 non-ARI multi-Function 设备，必须使用通用的 Max_Payload_Size 设置。
  - 对于在所有 Function 上 Max_Payload_Size 设置都不同的 non-ARI multi-Function 设备，所选的 Max_Payload_Size 设置是特定于实现的，但建议在所有 Function 上使用最大的 Max_Payload_Size 设置。

- UpdateFC FCP 的调度可能比要求的要频繁。

- 当链路处于 L0 或 L0s 链路状态时，必须为每种启用的非无限 FC 信用类型调度 Update FCP，至少每 30μs（-0%/ + 50%）传输一次，除非 Control Link 寄存器的 Extended Sync 字段置 1，这种情况下这个限制为 120μs（-0%/ + 50%）。

  - 超时机制可以可选地实现。如果实现，这种机制必须：
    - ◆ 只会在链路处于 L0 或 L0s 链路状态时有效。
    - ◆ 使用限制为 200μs（-0%/ + 50%）的计时器，其中通过接收任何 Init FCP 或 Update FCP 来重置计时器，或者可以通过接收任何 DLLP 来重置计时器（请参见 Section 3.5）。

- 计时器到期后，指示物理层重新训练链路（通过 LTSSM 恢复状态，Section 4.2.6.4）。
- 如果在初始化过程中为所有三个流控制类通告了无限信用，则必须禁用此超时机制。

注意：强烈建议实现此可选机制。本规范的未来修订版可能会将这种机制从可选更改为必需。

IMPLEMENTATION NOTE

Use of "Infinite" FC Advertisement

对于给定的实现，可能并非所有队列类型都需要在硬件中把所有虚通道都物理实现。例如，由于仅在 VC0 上允许 Non-Posted Write，因此无需为 VC0 以外的虚通道实现一个 Non-Posted Data queue。对于未实现的队列，接收器可以通过在初始化期间通告无限流控制信用来消除呈现跟踪流控制信用的需要。

IMPLEMENTATION NOTE

Flow Control Update Frequency

对于要接收 TLP 流的组件，希望实现的接收缓冲区大于由于防止缺乏可用信用而导致发送器节流所需的最小大小。同样，必须返回 UpdateFC FCP，以便发送、接收和处理 UpdateFC 所需的时间足够。请注意，Table 2-46，Table 2-47 和 Table 2-48 中显示了正常操作期间 UpdateFC 传输延迟的建议最大值。请注意，这些 Table 中给出的值不考虑接收器或发送器处于 L0s、Recovery 状态中或 Retimer 引起的任何延迟（请参阅 Section 4.3.8）。为了改善性能和节能，可能期望使用比简单计时器更复杂的流控制更新策略。任何此类策略都是特定于实现的，超出了本文档的范围。

Table 中的值是从接收端事务处理层通过处理接收到的 TLP 使额外的接收缓冲区空间可用到发送相应的 UpdateFC DLLP 的第一个符号开始测量的。

Table 2-46 Maximum UpdateFC Transmission Latency Guidelines for 2.5 GT/s (Symbol Times)

		Link Operating Width						
		x1	x2	x4	x8	x12	x16	x32
Max_Payload_Size (bytes)	128	237	128	73	67	58	48	33
	256	416	217	118	107	90	72	45
	512	559	289	154	86	109	86	52
	1024	1071	545	282	150	194	150	84
	2048	2095	1057	538	278	365	278	148
	4096	4143	2081	1050	534	706	534	276

公众号·芯片业者相聚汉

Table 2-47 Maximum UpdateFC Transmission Latency Guidelines for 5.0 GT/s (Symbol Times)

		Link Operating Width						
		x1	x2	x4	x8	x12	x16	x32
Max_Payload_Size (bytes)	128	288	179	124	118	109	99	84
	256	467	268	169	158	141	123	96
	512	610	340	205	137	160	137	103
	1024	1122	596	333	201	245	201	135
	2048	2146	1108	589	329	416	329	199
	4096	4194	2132	1101	585	757	585	327

Table 2-48 Maximum UpdateFC Transmission Latency Guidelines for 8.0 GT/s and Higher Data Rates (Symbol Times)

		Link Operating Width						
		x1	x2	x4	x8	x12	x16	x32
Max_Payload_Size (bytes)	128	333	224	169	163	154	144	129
	256	512	313	214	203	186	168	141
	512	655	385	250	182	205	182	148
	1024	1167	641	378	246	290	246	180
	2048	2191	1153	634	374	461	374	244
	4096	4239	2177	1146	630	802	630	372

2.7 Data Integrity

PCI Express 中的基本数据可靠性机制（data reliability mechanism）在数据链路层实现，该机制使用 32 位 CRC（LCRC）在链路级基础上检测 TLP 中的错误，并应用 Link-by-Link retransmit 机制实现恢复。TLP 是数据传输的基本单元，由设备创建，然后有可能通过中间组件（即 Switch）路由并传输给最终的目标 PCI Express 接收者。当 TLP 经过 Switch 时，有些字段可能会被 Switch 修改，但有些影响 TLP 传输的字段不能修改。在这种情况下，Switch 会重新生成 LCRC。Switch 内部可能会发生数据损坏，为损坏的数据重新生成一个好的 LCRC 会掩盖错误的存在。为了确保在要求高数据可靠性的系统中进行端到端数据完整性检测，可以在 TLP 末尾的 TLP Digest 字段中放置事务层端到端 32 位 CRC（ECRC）。ECRC 涵盖了随着 TLP 通过的路径而不变的所有字段（不变字段）。ECRC 由源组件中的事务层生成，并由最终 PCI Express 接收器以及可选的中间接收器进行检查（如果支持）。支持 ECRC 检查的 Switch 必须在针对 Switch 本身的 TLP 上检查 ECRC。这样的 Switch 可以选择检查它转发的 TLP 上的 ECRC。在 Switch 转发的 TLP 上，无论 Switch 是否检查 ECRC 或 ECRC 检查失败，Switch 都必须将 ECRC（未转发）保持为 TLP 的组成部分 [44]。

在某些情况下，已知在生成 TLP 时 TLP 有效负载中的数据已损坏，或者在通过中间组件（如 Switch）时可能已损坏。在这些情况下，错误传递（也称为数据中毒）可用于指示消耗数据的设备的损坏情况。

2.7.1 ECRC Rules

生成和检查 ECRC 需要由软件启用（请参见 Section 7.8.4.7）。

- 如果设备的 Function 可以生成 ECRC，则它必须为该 Function 所发起的所有 TLP 计算并应用 ECRC。

- Switch 在把 TLP 从 Ingress Port 传递到 Egress Port 时不能改变 ECRC 的值 [45]。
- 如果一个设备支持 ECRC 生成和检查，则它至少有一个 Function 必须实现 Advanced Error Reporting（参阅 Section 6.2）。
- 如果启用了设备 Function 的检查 ECRC 的功能，则必须对所有带有 ECRC 的 TLP 进行检查，其中设备是最终的 PCI Express 接收器。
    - 请注意，该 Function 仍然可以在不使用 ECRC 的情况下接收 TLP，并且可以正常处理这些–这不是错误。

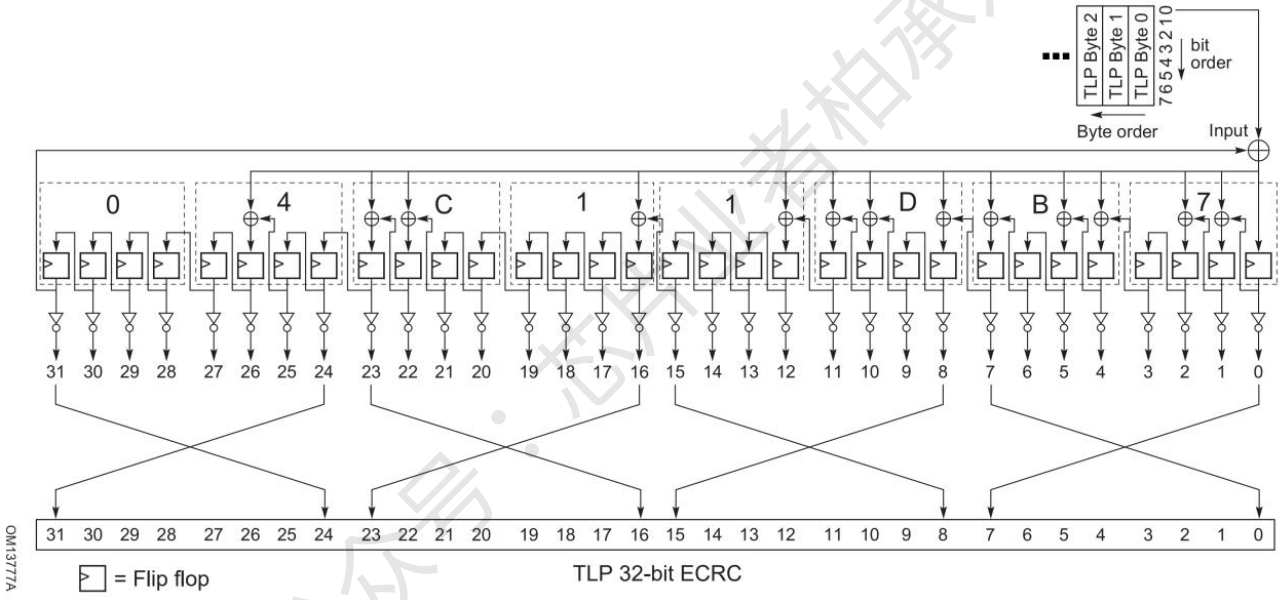
注意，Switch 可以选择对通过该 Switch 的 TLP 执行 ECRC 检查。Switch 检测到的 ECRC 错误的报告如 Table 6-5 所述，但不会改变 TLP 通过 Switch 的路径。

注 44：一个例外是由于 MC_Overlay 机制，Egress Port 需要修改 Multicast TLP。参见 Section 6.14.5。

注 45：一个例外是由于 MC_Overlay 机制，Egress Port 需要修改 Multicast TLP。参见 Section 6.14.5。

使用以下算法为 TLP（End-End TLP Prefix，header 和 data payload 部分）计算 32 位 ECRC，并将其附加到 TLP 的末尾（请参见 Figure 2-3）：

- ECRC 按下图所示算法计算。



- 使用的多项式的系数表示为 04C11DB7h。
- seed 值为 FFFFFFFFh。
- End-End Prefix（如果存在），header 和 data payload（如果存在）的所有不变字段都包含在 ECRC 计算中，对于 ECRC 计算，需改变字段都当做 1b 处理。
    - Type 字段的 bit 0 是个变量。当配置请求从 Type 1 更改为 Type 0 时，Type 字段的 bit 0 会更改。
    - EP 字段是个变量。
    - 所有其他字段是不变量。
- ECRC 计算从 byte 0 的 bit 0 开始，从 TLP 的每个字节的 bit 0 到 bit 7 进行。
- 对 ECRC 计算的结果进行补全，并将补全的结果位映射到 32 位 TLP Digest 字段中，如 Table 2-49 所示。

公众号·芯片业者相承双

Table 2-49 Mapping of Bits into ECRC Field

ECRC Result Bit	Corresponding Bit Position in the 32-bit TLP Digest Field
0	7
1	6
2	5
3	4
4	3
5	2
6	1
7	0
8	15
9	14
10	13
11	12
12	11
13	10
14	9
15	8
16	23
17	22
18	21
19	20
20	19
21	18
22	17
23	16
24	31
25	30
26	29
27	28
28	27
29	26
30	25
31	24

公众号·芯片业者相濡以沫

- 32 位 ECRC 值置于 TLP 末尾的 TLP Digest 字段中（见 Figure 2-3）。
- 对于用 TLP Digest 携带 ECRC 值的 TLP，支持端到端数据完整性检查的接收器，可通过以下方法检查 TLP Digest 字段中的 ECRC

值：

- 对接收到的 TLP 应用与 ECRC 计算相同的算法，不包括 TLP 的 32 位 TLP Digest 字段。
- 把计算值与 TLP 的 32 位 TLP Digest 字段比较。

■ 支持端到端数据完整性检查的接收器检查到 ECRC 错误会报告一个违规。报告的此错误与接收端口相关（请参见 Section 6.2）。

最终的 PCI Express 接收器如何利用 ECRC 提供的端到端数据完整性检查超出了本文档的范围。中间接收者仍然需要转发其 ECRC 检查失败的 TLP。就 ECRC 检查而言，PCI Express-to-PCI/PCI-X Bridge 被归类为最终的 PCI Express 接收器。

IMPLEMENTATION NOTE

Protection of TD Bit Inside Switches

Switch 确保并保持它们接收和转发的 TLP 中 TD 字段的完整性（即通过应用特殊的内部保护机制）是至关重要的，因为 TD 字段的损坏将导致最终目标设备不知道该 TLP 是否存在 TLP Digest 字段。

同样，强烈建议 Switch 为它们接收和转发的 TLP 中的其他变量字段提供内部保护，因为 ECRC 不能维持变量字段的端到端完整性。

IMPLEMENTATION NOTE

Data Link Layer Does Not Have Internal TLP Visibility

由于数据链路层不处理 TLP header（它根据来自物理层的指示确定 TLP 的开始和结束），因此它不知道 TLP Digest 字段的存在，只是将其作为 TLP 的一部分传递给事务处理层。

2.7.2 Error Forwarding

错误传递（Error Forwarding），也叫数据中毒（Data Poisoning），指的是在已知 TLP Data Payload 被破坏（Corrupted）的情况下，该 TLP 仍然被发送至其他的 PCIe 设备。此时，该 TLP 包头的 EP 位（Error Poisoned）被置位为 1，表明该 TLP 已经被破坏。Section 2.7.2.2 介绍了此应用的规则。以下是一些可能使用错误转发的情况的示例：

- Example #1：读取主内存时遇到不可纠正的错误。
- Example #2：PCI 写内存时遇到奇偶校验错误。
- Example #3：内部数据缓存的数据完整性错误。

2.7.2.1 Error Forwarding Usage Model

- 错误传递只是针对 TLP 中的 Data Payload 是否被破坏，和 TLP 包头的内容无关。也就是说错误传递只是针对那些带有 Data Payload 的 TLP 的，如 Memory、Configuration、I/O 写或者带有返回数据的 Completion。PCIe Spec 没有定义对没有 Data Payload 的 TLP，其 TLP 包头中的 EP 却为 1 的情况应当如何处理。
- 错误传递用于控制系统中错误的传播，以及进行系统诊断等。
- 请注意，错误传递不会导致 Link Layer Retry——仅当数据链路层中的 TLP 错误检测机制确定链路上存在传输错误时，才会重试中毒的 TLP。
  - ■ 在读取操作或采取某些其他措施的情况下，中毒 TLP 可能最终导致请求的始发者（在事务层或更高级别）重新发出请求。错误转发信息的这种使用超出了本规范的范围。

特殊应用场景举例：

1. 便于发送端（Request）和系统分析错误：假设发送端（Request）向接收端（Completer）发送了读数据请求，接收端从某个内存设备中读取数据后通过 Completion 返回数据给发送端。但是在此过程中发生了错误，接收端（Completer）因此不向发送端（Request）返回 Completion，则发送端只会产生 Completion Timeout 错误，却难以分析错误原因。如果接收端返回 Poisoned Completion TLP 给发送端（TLP 包头中 EP 为 1），则发送端至少可以确认接收端正确地接收到了其发出的请求（Request）。
2. 便于发现 Switch（或其他桥设备）中的错误：假设 TLP 中的 Data Payload 是在 Switch 中被破坏的，采用错误传递的方式有助于发现该错误。
3. 有些应用允许接收存在错误的数据：比如实时的音频或者视频传输，其宁可接收到有些许错误的数据，也需要尽量保证数据传输的实时性。
4. 数据可能通过应用层恢复：有些应用可能采用了特殊的编码 ，该编码可以恢复某些被破坏的数据（如 ECC 可恢复 1 位的错误）。

注：需要注意的是，Poisoning 操作只能在事务层进行。原因很简单：数据链路层和物理层在任何情况下，都不会检查 TLP 包头的内容，更不会修改 TLP 包头。

2.7.2.2 Rules For Use of Data Poisoning

- 在发送器中实现 TLP 中毒是可选实现的。
- 数据中毒只会应用于带数据的 Write Request (Posted or Non-Posted)、Message with Data、AtomicOp Request、Read Completion、或 AtomicOp Completion 中。
  - 中毒 TLP 的 EP 字段会设置为 1b。
  - 发送器仅可将包含数据有效载荷的 TLP 的 EP 位设置为 1b。如果不包括数据有效载荷的任何 TLP 设置了 EP 位，则未指定接收器的行为。
- 如果发送器支持数据中毒，则发送器已知包含不良数据的 TLP 必须使用上面定义的中毒机制。
- 如果下游端口支持 Poisoned TLP Egress Blocking，Poisoned TLP Egress Blocking Enable 字段置 1，并且中毒 TLP 的目的地需要从 Egress Port 流出，则除非更高的优先级，否则该端口必须将 TLP 作为一个 Poisoned TLP Egress Blocked error 处理。参见 Section 6.2.3.2.3，Section 6.2.5 和 Section 7.9.15.2。进一步：
  - 该 Port 不能发送此 TLP。
  - 如果未触发 DPC 且 TLP 是 Non-Posted Request，则端口必须返回带有 Unsupported Request Completion Status 的 Completion。
  - 如果触发了 DPC，则端口必须按照 Section 2.9.3 中所述进行操作。
- 下列带中毒数据的请求不能修改其中目标位置的值：
  - Configuration Write Request。
  - 下列以控制寄存器或 control structure 为目标的 Completer：携带数据的 I/O Write Request、 Memory Write Request 或 non-vendor-defined Message。
  - AtomicOp Request。

公众号·芯片业者相承双

除非存在更高的优先级错误，否则完成者必须将这些请求作为 Poisoned TLP Received error 来处理[48]，并且如果请求是 Non-Posted 的，则完成者还必须返回完成状态为 Unsupported Request (UR)的 Completion（请参阅 Section 6.2.3.2.3、Section 6.2.3.2.4 和 Section 6.2.5）。

Switch 必须按原先未中毒的路由方式路由请求，除非该请求的目标是 Switch 本身，在这种情况下，Switch 是请求的完成者，并且必须遵循上述规则。

注 48：由于本规范较早版本中的语言不明确，因此允许组件将这种错误作为不支持的请求来处理，但是强烈建议不要这样做。

对于某些应用程序，可能更需要 Completer 在写请求中使用不以控制寄存器或控制结构为目标的中毒数据-这种使用是禁止的。同样，请求者可能希望使用在完成中标记为中毒的数据-这种使用方式也是禁止的。中毒信息的适当使用是特定于应用程序的，本文档中不再讨论。

该文档没有定义任何机制来确定中毒 TLP 的数据有效载荷的哪些部分实际上已损坏，以及哪些未损坏。

2.8 Completion Timeout Mechanism

在任何分立事务协议（split transaction protocol）中，都存在请求者收不到预期的完成的风险。为了允许请求者以标准方式尝试从这种情况下恢复，定义了“完成超时”机制。仅在没有合理预期的 Completion 会返回的情况下才应激活此机制，并且在正常操作条件下绝对用不着该机制。请注意，此处指定的值不反映预期的服务延迟，且不得用于估计典型的响应时间。

如果某个 PCI Express 设备的 Function 能够发出需要用 Completion 响应的请求，则该 Function 必须实现完成超时机制。配置请求有一个例外（请参见下文）。发送请求时，将为需要一个或多个完成的每个请求激活完成超时机制。由于 Switch 不会自动发送需要 Completion 的请求，因此对完成超时支持的要求仅限于 RC、PCI Express-PCI Bridge 和 Endpoint。

完成超时机制可能会被配置软件禁用。 完成超时限制是在 Device Control 2 寄存器的 Completion Timeout Value 字段中设置的。请参阅 Section 2.2.9。完成超时是与请求者 Function 相关的报告错误（请参见 Section 6.2）。

注意：仅当请求者收到所有 Completion 时，才应认为 Memory Read Request 完成。如果在完成超时定时器到期之前返回了一些（但不是全部）请求的数据，则允许请求者保留或丢弃在定时器到期之前返回的数据。

配置请求的完成超时对支持 PCI Express to PCI/PCI Express bridge 有特殊要求。默认情况下，PCI Express to PCI/PCI Express bridge 不会为目标为桥上挂的 PCI/PCI-X 设备的配置请求返回 Configuration Request Retry Status (CRS)。这可能导致冗长的完成延迟，而 RC 中的“完成超时”值必须理解该延迟。系统软件可以通过设置 Device Control 寄存器中的 Bridge Configuration Retry Enable 字段来使 PCI Express to PCI/PCI Express bridge 返回 Configuration Request Retry Status，但要遵守 *PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0* 规范中所述的限制。

IMPLEMENTATION NOTE

Completion Timeout Prefix/Header Log Capable

与完成超时相关联的 Request TLP 的 Prefix/Header 可以由实现 AER Capability 的请求者有选择地记录。通过 Advanced Error Capability 和 Control 寄存器中的 Completion Timeout Prefix/Header Log Capable 字段的值指示对 Prefix/Header 的记录的支持。

完成超时可能是由于配置不正确，系统故障或异步移除引起的（Async Removal ，请参见 Section 6.7.6）。为了使主机软件能够区分“完成超时”错误，在此之后不可能继续进行正常操作（例如，由不正确的配置或系统故障引起的错误）与可能继续进行正常操作的操作（例如，在“异步移除”之后）， 强烈建议请求者记录与完成超时关联的请求 TLP Prefix/Header。

2.9 Link Status Dependencies

2.9.1 Transaction Layer Behavior in DL_Down Status

DL_Down 状态表示链路上与另一个组件没有连接，或者与其他组件的连接已丢失，并且物理层或数据链路层无法恢复。此部分指定当 DPC 没有被触发，并且数据链路层向事务层报告 DL_Down 状态时事务层的行为，这意味着链路不可操作。Section 2.9.3 规定了触发 DPC 时的行为。

- 对于处于 DL_Down 状态的端口，事务层不需要接收来自数据链路层的已接收 TLP，前提是数据链路层未确认（Ack 或 Nak）这些 TLP。这样的 TLP 不会修改接收端的流控制信用。

对于 Downstream Port，DL_Down 状态按以下方式处理：

- 将向下游传输的 outstanding request 关联的所有缓冲区或内部状态初始化回其默认状态。
  - Note：端口配置寄存器不得受到影响，除非在 DL_Down 状态发生改变时要更新一些状态。
- 对于 Non-Posted Request，为设备核心提交的用于传输的所有请求构造 Completion，然后返回 Unsupported Request 完成状态，然后丢弃这些请求。
  - 这是与端口相关的（虚拟）网桥 Function 相关的报告错误（请参见 Section 6.2）。对于根端口，此错误的报告是可选的。
  - 事务层已经在处理的 Non-Posted Request 将被丢弃。

    Note：这等效于在链路状态变为 DL_Down 之前请求已发送但尚未完成的情况。
    - 这种情况由请求者使用完成超时机制处理。

  Note：Non-Posted Request 变得“不完整”的时间点是特定于实现的。
- 端口必须终止以该端口为目标的任何 PME_Turn_Off 握手请求，以使该端口被视为已确认 PME_Turn_Off 请求（请参阅 Section 5.3.3.2.1 中的实现说明）。
- 端口必须按照 Section 2.2.8.6 中的描述处理 Vendor Defined Message Request（例如，静默丢弃它不打算接收的 Vendor Defined Type 1 Messages Request），因为 DL_Down 阻止了请求到达其目标 Function。
- 对于所有其他的 Posted Request，丢掉该 Request。
  - 这是一个与端口相关的（虚拟）网桥 Function 相关的报告错误（请参见 Section 6.2），必须报告为 Unsupported Request。对于根端口，此错误的报告是可选的。
  - 对于已由事务层处理的 Posted Request，允许端口不报告错误。

    Note：这等效于在链路状态变为 DL_Down 之前已发送请求的情况。

  Note：Posted Request 变得“不完整”的时间点是特定于实现的。
- 丢弃设备核心提交的所有 Completion 传输。

对于 Upstream Port，DL_Down 状态按以下方式处理：

- 将所有 PCI Express 特定的寄存器，状态机和外部可观察到的状态返回到指定的默认或初始条件（定义为粘性的寄存器除外-参见 Section 7.4）。

公众号：芯片业者相承效

- 丢弃所有正在处理的 TLP。
- （对于 Switch 和 Bridge）将 Hot Reset 传播到所有关联的 Downstream Port。在支持链路速度大于 5.0 GT/s 的 Switch 中，Upstream Port 必须将每个 Downstream Port 的 LTSSM 引导到 Hot Reset 状态，但不能将 LTSSM 保持在该状态。 这样就使得每个 Downstream Port 在其 Hot Reset 完成后立即开始链路训练。建议所有 Switch 使用这种行为。

2.9.2 Transaction Layer Behavior in DL_Up Status

DL_Up 状态表示已与链路上的另一个组件建立连接。本部分指明当数据链路层向事务层报告 DL_Up 状态时事务层的行为，即指明该链路可操作。具有 DL_Up 状态的端口的事务层必须接受符合本规范其他规则的已接收 TLP。

对于一个 RC 或 Switch 的 Downstream Port：

- 从非 DL_Up 状态转换为 DL_Up 状态时，端口必须发送 Set_Slot_Power_Limit 消息到链路上其他组件，以传达在 Slot Capabilities 寄存器的 Slot Power Limit Scale 和 Value 字段中编程的值。如果尚未初始化 Slot Capabilities 寄存器，则此传输是可选的。

2.9.3 Transaction Layer Behavior During Downstream Port Containment

在 Downstream Port Containment（DPC）期间，与下游端口关联的 LTSSM 定向为 Disabled 状态。一旦达到 Disabled 状态，只要 DPC Status 寄存器中的 DPC Trigger Status 字段被置 1，它就一直保持为 1。有关软件多长时间必须退出处于 DPC 中的下游端口的要求，请参见 Section 6.2.10。本部分指定了 DPC 触发后，只要下游端口保留在 DPC 中的事务层的行为。

- 一旦触发了 DPC，就不会再从数据链路层接受其他（上游）TLP。
- 如果触发 DPC 的条件与上游 TLP 相关联，则必须静默丢弃已从数据链路层接受的任何后续的上游 TLP。

下游端口以以下方式处理由设备内核提交的（下游）TLP。

- 如果触发 DPC 的条件与下游 TLP 相关，则允许在链路断开之前静默地丢弃或传输任何先前的下游 TLP。否则，以下规则适用。
  - 对于每个 Non-Posted Request，端口必须返回一个 Completion 并静默地丢弃该请求。Completer ID 字段必须包含与下游端口关联的值。
    - 如果在 DPC Control 寄存器中设置了 DPC Completion Control 字段，则生成 Unsupported Request (UR) Completion Status 的 Completion。
    - 如果 DPC Completion Control 字段清 0，则生成 Completer Abort (CA) Completion Status 的 Completion。
  - 端口必须终止任何针对该端口的 PME_Turn_Off 握手请求，以使该端口被视为已确认 PME_Turn_Off 请求（请参阅 Section 5.3.3.2.1 中的实现说明）。
  - 端口必须处理 Vendor Defined Message Request，如 Section 2.2.8.6 所述。（例如，静默地丢弃 Vendor Defined Type 1 Message Request，这不是设计用来接收的），因为 DL_Down 阻止请求到达其目的地 Function。
  - 对于所有其他 Posted Request 和 Completion，端口必须自动丢弃 TLP。

对于任何 outstanding 的 Non-Posted Request，如果触发的 DPC 阻止返回其关联的 Completion，则以下情况适用：

- 对于支持 DPC 的 RP Extension 的根端口，根端口可以跟踪某些 Non-Posted Request，并且在触发 DPC 时，为每个跟踪的请求合成一个 Completion。这有助于避免因触发 DPC 而产生副作用的完成超时。每个合成的 Completion 必须携带由 DPC Completion Control 字段确定的 UR 或 CA 完成状态。跟踪的 Non-Posted Request 的特征是特定于实现的，但强烈建议跟踪由

公众号：芯片业者相关双

主机处理器指令生成的所有 Non-Posted Request（例如，“读”、“写”、“加载”、“存储”或与原子操作相对应的请求）。跟踪的其他候选对象包括来自其他根端口的对等请求和来自 RCiEP 的请求。

- 否则，关联的请求者可能会遭遇“完成超时”。软件解决方案栈应理解并考虑这种可能性。

公众号·芯片业者柏承双

3 Data Link Layer Specification

数据链路层充当事务层和物理层之间的中间阶段。 它的主要职责是提供一种可靠的机制，用于在链路上的两个组件之间交换事务层数据包（TLP）。

3.1 Data Link Layer Overview

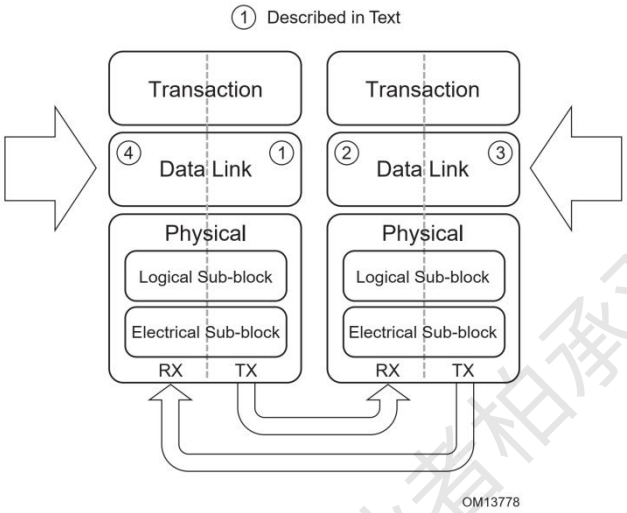


Figure 3-1: Layering Diagram Highlighting the Data Link Layer

数据链路层负责通过 PCI Express 链路可靠地将事务层提供的事务层数据包（TLP）传输到其他组件的事务层。 数据链路层提供的服务包括：

数据交换（Data Exchange）：

- 在发送端接收事务层 TLP，送到物理层。
- 在接收端把物理层接收到的 TLP 发送给事务层。

错误检测和重试（Error Detection and Retry）：

- TLP Sequence Number 和 LCRC 生成。
- 存储发送端的 TLP 用于链路层重试。
- TLP 和 DLLP 的数据链路层数据完整性检查。
- Ack/Nak 机制。
- 错误指示，用于错误报告和日志记录机制。
- 链路确认超时重播机制。

初始化和电源管理（Initialization and power management）：

- 跟踪链路状态并把 active/reset/disconnected 等状态发送给事务层。

DLLP（Data Link Layer Packet）：

公众号 · 芯片业者相承双

- 用作链路管理功能的 Ack/Nak DLLP、用作电源管理的 DLLP 和用作流控管理的 DLLP。
- DLLP 在链路两侧设备的的数据链路层交换。

DLLP 是点到点而不是端到端的传输。TLP 是端到端传输。

DLLP 和 TLP 的数据完整性检查是通过跨链路发送的每个数据包附带的 CRC 完成的。DLLP 使用 16 位 CRC，而 TLP 使用 32 位 LCRC。TLP 还包含一个序列号，该序列号用于检测一个或多个完整 TLP 是否丢失。

收到的未通过 CRC 检查的 DLLP 将被丢弃。使用 DLLP 的机制可能会因信息丢失而遭受性能损失，但会进行自我修复，以使后续的 DLLP 会取代任何丢失的信息。

数据完整性检查（LCRC 检查和序列号检查）失败，或在从一个组件到另一个组件的传输中丢失了 TLP，则由发送器重新发送。发送器存储所有已发送 TLP 的副本，并在需要时重新发送这些副本，并且仅当它从其他组件收到 Ack 确认后才清除这些副本。如果在指定时间段内未收到 Ack，则发送器将自动开始重新发送。 接收方可以使用 Nak 请求发送方立即重发。

数据链路层像是一个信息管道，对事务层的延迟有所不同。在任何给定的单个链路上，灌入发送数据链路层（1 和 3）的所有 TLP 之后将以相同顺序出现在接收数据链路层（2 和 4）的输出处，如 Figure 3-1 所示。 之间的延迟将取决于许多因素，包括流水线延迟、链路的宽度、工作频率、链路上电信号的传输以及由数据链路层重试引起的延迟。由于这些延迟，传输数据链路层（1 和 3）可以向传输事务层施加反压，而接收数据链路层（2 和 4）将有效信息的存在或不存在传达给接收事务层。

3.2 Data Link Control and Management State Machine

数据链路层跟踪链路的状态。它与事务层和物理层通信链路状态，并通过物理层执行链路管理。数据链路层包含执行这些任务的数据链路控制和管理状态机（DLCMSM）。该状态机的状态描述如下，如 Figure 3-2 所示。

状态机状态有：

- DL_Inactive ： 物理层报告链路不可用或端口无连接时数据链路层处于 DL_Inactive 状态。
- DL_Feature（可选）：物理层报告链路可操作，执行 Data Link Feature Exchange。
- DL_Init ： 物理层报告链接可操作后进入 DL_Init，数据链路层在 DL_Init 状态为默认虚通道初始化流控制。
- DL_Active ： 正常操作状态。

输出的数据链路层状态有两个：

- DL_Down ： 数据链路层未与链路另一侧的组件进行通信。
- DL_Up：数据链路层正在与链路另一侧的组件进行通信。

公众号·芯片业者相承

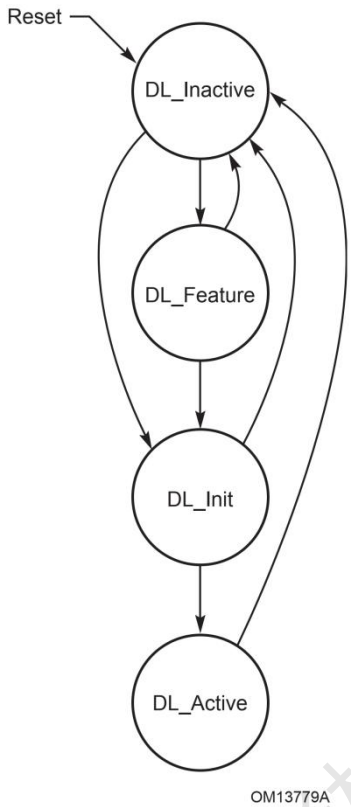


Figure 3-2 Data Link Control and Management State Machine

3.2.1 Data Link Control and Management State Machine Rules

- **DL_Inactive**
  - PCI Express Hot Reset、Warm Reset 或 Cold Reset 后数据链路层进入该初始状态（请参阅 Section 6.6）。请注意，DL 状态不受 FLR 的影响（请参阅 Section 6.6）。
  - 进入 DL_Inactive 时：
    - 所有数据链路层状态复位为默认值。
    - 如果端口支持可选实现的 Data Link Feature Exchange，则 Remote Data Link Feature Supported 字段和 Remote Data Link Feature Supported Valid 字段必须清 0。
    - 丢弃数据链路层 Retry Buffer 的值（see Section 3.6）。
  - 在 DL_Inactive 时：
    - 向事务层以及对端设备数据链路层报告 DL_Down 状态。

    这将导致事务层丢弃所有 outstanding transaction，并在内部终止尝试发送 TLP。对于下游端口，这等效于"Hot-Remove"。对于上游端口，使链路断开等效于 Hot Reset（see Section 2.9）。
    - 丢弃来自事务层和物理层的 TLP。
    - 不能生成 DLLP，也不能接收 DLLP。
  - 在以下情况会退出 DL_Inactive 并进入 DL_Feature 状态：

公众号·芯片业者相承双

- 端口支持可选的 Data Link Feature Exchange，Data Link Feature Exchange Enable 字段被置 1，事务层指示该链路未被软件禁用，并且物理层报告 Physical LinkUp = 1b。

- 退出 DL_Inactive 时：
  - 如果端口不支持可选实现的 Data Link Feature Exchange，事务层指示该链路未被软件禁用，并且物理层报告 Physical LinkUp = 1b，or
  - 如果端口支持可选实现的 Data Link Feature Exchange，则 Data Link Feature Exchange Enable 字段清 0，事务层指示该链路未被软件禁用，并且物理层报告 Physical LinkUp = 1b。

- **DL_Feature**
  - 在 DL_Feature 时，
    - 按 Section 3.3 描述的执行 Data Link Feature Exchange 协议。
    - 报告 DL_Down 状态。
    - 允许状态为 DL_Down 的端口的数据链路层丢弃任何接收到的 TLP，前提是它不通过发送一个或多个 Ack DLLP 来确认那些 TLP。
  - 在以下情况会退出 DL_Feature 并进入 DL_Init：
    - Data Link Feature Exchange 成功完成，并且物理层继续报告 Physical LinkUp = 1b，or
    - Data Link Feature Exchange 确定远程数据链路层不支持可选的 Data Link Feature Exchange 协议，并且物理层继续报告 Physical LinkUp = 1b。
  - 在下列情况下，终止 Data Link Feature Exchange 协议并退出到 DL_Inactive：
    - 物理层报告 Physical LinkUp = 0b 。

- **DL_Init**
  - 在 DL_init 时：
    - 按照 Section 3.3 中描述的流控制初始化协议，为默认虚通道 VC0 初始化流控制。
    - 发送 FC_INIT1 DLLP 的过程中报告 DL_Down 状态；发送 FC_INIT2 DLLP 的过程中报告 DL_Up 状态。
    - 允许状态为 DL_Down 的端口的数据链路层丢弃任何接收到的 TLP，前提是它不通过发送一个或多个 Ack DLLP 来确认那些 TLP。
  - 在以下情况会退出 DL_Init 并进入 DL_Active 状态：
    - 流控制初始化成功完成，并且物理层继续报告 Physical LinkUp = 1b。
  - 在以下情况下，终止尝试初始化 VC0 的流控制并退出到 DL_Inactive：
    - 物理层报告 Physical LinkUp = 0b。

- **DL_Active**
  - DL_Active 为正常操作状态。
  - 在 DL_Active 时：
    - TLP 正常传输。

公众号：芯片业者怕承双

- DLLP 正常发送和接收。
- 向事务层和物理层报告 DL_Up 状态。

- 在下述情况退出到 DL_Inactive 状态：
  - 物理层报告 Physical LinkUp = 0b。
  - 上游组件可以将从 DL_Active 到 DL_Inactive 的转换视为一个 Surprise Down error，除非以下情况阻止了此错误检测：
    - 如果通过软件将 Bridge Control 寄存器中的 Secondary Bus Reset 字段设置为 1b，则随后的向 DL_Inactive 的转换不得视为错误。
    - 如果软件已将 Link Disable 字段设置为 1b，则随后的向 DL_Inactive 的转换不得视为错误。
    - 如果由于该端口上方的事件而导致 Switch 下游端口转换为 DL_Inactive，则转换到 DL_Inactive 不得被视为一个错误。示例事件包括：Switch 上游端口发送一个 Hot Reset，Switch 上游链路转换为 DL_Down 以及设置了 Switch 上游端口中的 Secondary Bus Reset 字段。
    - 如果已通过此端口发送了 PME_Turn_Off 消息，则随后的向 DL_Inactive 的转换不得视为错误。
    - 请注意，只有在关闭电源，复位或恢复链路的请求发送到物理层后，才会发生这种向 DL_Inactive 的转换。
    - 请注意，在 PME_Turn_Off/PME_TO_Ack 握手未能成功完成的情况下，可能会检测到 Surprise Down 错误。
    - 如果端口与可热插拔插槽关联（Slot Capabilities 寄存器中的 Hot-Plug Capable 设置为 1b），并且 Slot Capabilities 寄存器中的 Hot-Plug Surprise 设置为 1b，则任何向 DL_Inactive 的转换都不视为错误。
    - 如果端口与可热插拔插槽关联（Slot Capabilities 寄存器中的 Hot-Plug Capable 设置为 1b），并且 Slot Capabilities 寄存器中的 Power Controller Control 设置为 1b（Off），则任何向 DL_Inactive 的转换都不视为错误。

  由上述情况中的一种或多种引起的错误阻止必须保持有效，直到端口退出 DL_Active 并随后返回 DL_Active 为止，而再次返回 DL_Active 时没有发生任何阻止情况。

  请注意，从 DL_Active 退出只是根据错误检测阻止条件所预期的预期过渡。

  如果实现，则是报告的与检测端口相关的错误（请参见 Section 6.2）。

IMPLEMENTATION NOTE

Physical Layer Throttling

请注意，在某些情况下，物理层可能暂时无法接受来自数据链路层的 TLP 和 DLLP。数据链路层必须通过为物理层提供传达此条件的机制，以及为 TLP 和 DLLP 被该条件暂时阻止的机制来理解这一点。

3.3 Data Link Feature Exchange

Data Link Feature Exchange 协议是可选实现的。实现此协议的端口会实现一个 Data Link Feature Extended Capability（请参阅 Section 7.7.4）。此 Capability 包含四个字段：

- Local Data Link Feature Supported 字段表示 local Port 支持此数据链路功能。

- Remote Data Link Feature Supported 字段表示 remote Port 支持此数据链路功能。
- Remote Data Link Feature Supported Valid 字段表示 Remote Data Link Feature Supported 字段包含有效数据。
- Data Link Feature Exchange Enable 字段允许系统禁用 Data Link Feature Exchange。这可用于解决无法正确忽略 DLLP 的旧硬件的问题。

Data Link Feature Exchange 协议将 Local Port 的 Feature Supported 的信息传输到 Remote Port，并捕获该 Remote Port 的 Feature Supported 的信息。

此协议的规则如下：

- 在进入 DL_Feature 时：
  - Remote Data Link Feature Supported 和 Remote Data Link Feature Supported Valid 字段必须清 0。
- 在 DL_Feature 时：
  - 事务层必须阻塞 TLP 的发送。
  - 发送 Data Link Feature DLLP。
    - 发送的 DLLP 的 Feature Supported 字段必须等于 Local Data Link Feature Supported 字段的值。
    - 发送的 DLLP 的 Feature Ack 字段必须等于 Remote Data Link Feature Supported Valid 字段值。
  - Data Link Feature DLLP 必须每 34μs 至少发送一次。在 Recovery 或 Configuration 状态中花费的时间不会导致此限制。
  - 处理收到的 Data Link Feature DLLP。
    - 如果 Remote Data Link Feature Supported Valid 字段清 0，请在 Remote Data Link Feature Supported 字段中记录接收到的 Data Link Feature DLLP 中的 Feature Supported 字段的值，然后把 Remote Data Link Feature Supported Valid 字段设置为 1。
- 在以下情况下会退出 DL_Feature：
  - 收到了一个 InitFC1 DLLP。
  - 收到了一个 MR-IOV MRInit DLLP（编码为 0000 0001b）。or
  - 在 DL_Feature 中时，至少收到一个 Feature Ack 字段设置为 1 的 Data Link Feature DLLP。

每个 Data Link Feature 在 Feature Supported 字段中都有一个对应的比特位。当在 Local Data Link Feature Supported 和 Remote Data Link Feature Supported 字段中将该位设置为“1 时，将激活此 Data Link Feature。

Table 3-1 中显示了 Data Link Feature 及其对应的比特位置。

Table 3-1 Data Link Feature Supported Bit Definition

Bit Location	Description
0	Scaled Flow Control - indicates support for Scaled Flow Control. Scaled Flow Control must be supported in Ports that support 16.0 GT/s or higher operation.
22:1	Reserved

公众号：芯片业者相承双

3.4 Flow Control Initialization Protocol

在链路正常运行之前，有必要为默认虚通道 VC0 初始化流控制（Section 6.6）。此外，启用其他虚通道（VC）后，必须为每个新启用的 VC 完成流控制初始化过程，然后才能使用它（see Section 2.6.1）。本节介绍了用于所有 VC 的初始化过程。请注意，由于在所有其他 VC 之前先启用了 VC0，所以在初始化 VC0 之前不会有任何类型的 TLP 业务流处于活动状态。但是，在初始化其他 VC 时，通常将在其他已启用的 VC 上传输 TLP 业务流。此类业务对其他 VC 的初始化过程没有直接影响。

VC 初始化有两个过程：

- FC_INIT1
- FC_INIT2

各个过程的规则如下节描述。

3.4.1 Flow Control Initialization State Machine Rules

- 如果在 VC 1-7 初始化期间的任何时间禁用了 VC，则终止 VC 的流控制初始化过程。
- FC_INIT1 状态的规则：
  - 如果需要初始化一个 VC（VCx）就进入此状态。
    - 在进入 DL_Init 状态时，VCx = VC0。
    - 当软件使能一个 VC（VCx = VC1-7）时就会进入 FC_INIT1（see Sections 7.9.1 and 7.9.2）。
  - 在 FC_INIT1 状态下：
    - 事务层必须阻止使用 VCx 传输 TLP。
    - 按下述顺序为 VCx 发送以下三个 InitFC1 DLLP：
      - InitFC1 – P (first)
      - InitFC1 – NP (second)
      - InitFC1 – Cpl (third)
    - 三个 InitFC1 DLLP 必须每 34μs 至少发送一次。
      - 在 Recovery 或 Configuration 状态消耗的时间不会导致此限制。
      - 强烈建议频繁重复 InitFC1 DLLP 传输，尤其是在没有其他 TLP 或 DLLP 可用于传输时。
    - 如果链路激活了 Scaled Flow Control，请将 InitFC1 DLLP 中的 HdrScale 和 DataScale 字段设置为 01b，10b 或 11b，以指示其在相应的 HdrFC 和 DataFC 值上使用的缩放因子。
    - 如果发送器不支持 Scaled Flow Control，或者如果链路未激活 Scaled Flow Control，则将 HdrScale 和 DataScale 字段设置为 00b。
    - 除非要保证最小的传输 InitFC1 DLLP 的频率，否则数据链路层不得阻止其他传输。
      - 请注意，这包括所有物理层发起的传输（例如，有序集）、Ack 和 Nak DLLP、以及使用先前已完成初始化的 VC 的 TLP。

公众号：芯片业者相承双

- 处理收到的 InitFC1 和 InitFC2 DLLP：
  - 记录其中携带的 HdrFC 和 DataFC 值。
  - 如果接收器支持 Scaled Flow Control，则记录其中的 HdrScale 和 DataScale 值。
  - 一旦记录了 VCx 的每个 P、NP 和 Cpl 的 FC unit 值，就设置标志 FI1。

- 下述条件满足则退出 FC_INIT1 状态进入 FC_INIT2：
  - 已设置 FI1，表示已为 VCx 的每个 P、NP 和 Cpl 记录了 FC unit 值，则进入 FC_INIT2 状态。

- FC_INIT2 状态的规则：
  - 在 FC_INIT2 状态时：
    - 事务层必须阻止使用 VCx 传输 TLP。
    - 按下述顺序为 VCx 发送以下三个 InitFC2 DLLP：
      - InitFC2 – P (first)
      - InitFC2 – NP (second)
      - InitFC2 – Cpl (third)
    - 三个 InitFC1 DLLP 必须每 34μs 至少发送一次。
      - 在 Recovery 或 Configuration 状态消耗的时间不会导致此限制。
      - 强烈建议频繁重复 InitFC2 DLLP 传输，尤其是在没有其他 TLP 或 DLLP 可用于传输时。
    - 如果链路激活了 Scaled Flow Control，请将 InitFC2 DLLP 中的 HdrScale 和 DataScale 字段设置为 01b，10b 或 11b，以指示其在相应的 HdrFC 和 DataFC 值上使用的缩放因子。
    - 如果发送器不支持 Scaled Flow Control，或者如果链路未激活 Scaled Flow Control，则将 HdrScale 和 DataScale 字段设置为 00b。
    - 除非要保证最小的传输 InitFC2 DLLP 的频率，否则数据链路层不得阻止其他传输。
      - 请注意，这包括所有物理层发起的传输（例如，有序集）、Ack 和 Nak DLLP、以及使用先前已完成初始化的 VC 的 TLP。
    - 处理收到的 InitFC1 和 InitFC2 DLLP：
      - 忽略其中携带的 HdrFC、DataFC、HdrScale 和 DataScale 的值。
      - 一旦收到任何针对 VCx 的 InitFC2 DLLP，就设置标志 FI2。
    - 在收到 VCx 上的任何 TLP 或 UpdateFC DLLP 时，设置标志 FI2。
  - 满足下述条件则完成初始化并退出 FC_INIT2：
    - FI2 被设置为 1。
    - 如果链路激活了 Scaled Flow Control，则发送器必须在 VCx 的所有 UpdateFC DLLP 中为 HdrScale 和 DataScale 发送 01b，10b 或 11b。
    - 如果支持 Scaled Flow Control 或链路未激活 Scaled Flow Control，则发送器必须在所有 VCx 的 UpdateFC DLLP 中为

公众号：芯片业者相承双

HdrScale 和 DataScale 发送 00b。

IMPLEMENTATION NOTE

Example of Flow Control Initialization

Figure 3-3 说明了 Switch 和下游组件之间 VC0 的流控制初始化协议的示例。在此示例中，每个组件都会为每种类型的流控制信用通告最小允许值。对于这两个组件，支持的最大 Max_Payload_Size 值为 1024 字节，对应于 040h 的数据有效负载信用通告。所有 DLLP 都显示为已收到，没有错误。

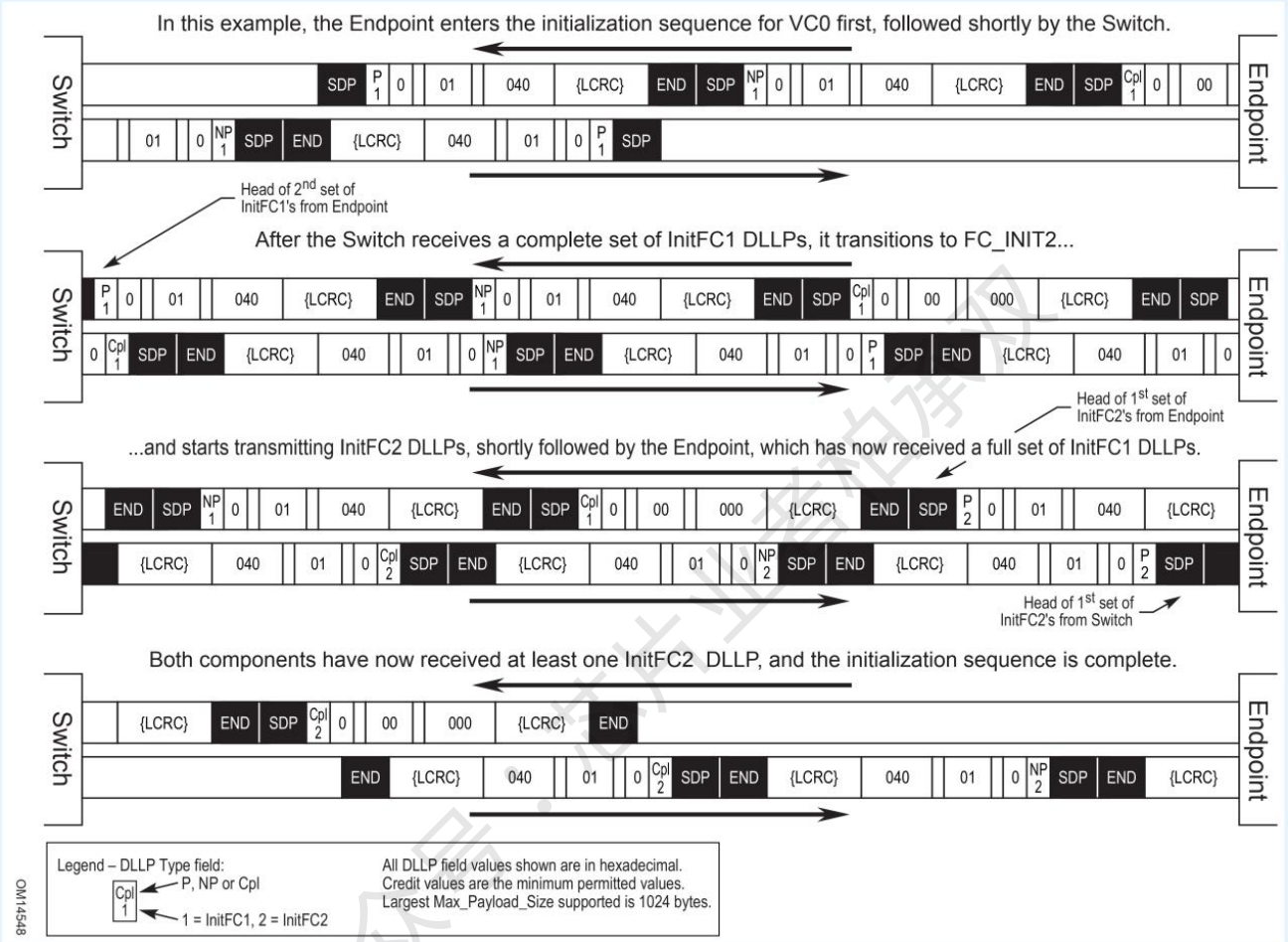


Figure 3-3 VC0 Flow Control Initialization Example with 8b/10b Encoding-based Framing

3.4.2 Scaled Flow Control

如果没有足够的流控制信用额度，则可能会影响链路性能。在较高的链路速度下，这种影响会变得更加明显，而限制 127 个 Header 信用（流控 DLLP 中 HdrFC 字段为 8-Bit）和 2047 个 Data 信用（流控 DLLP 中 DataFC 字段为 12-Bit）会限制性能。Scaled Flow Control 机制旨在解决此限制。（1 Credit = 16B）

Flow Control Credits 代表的是接收端 VC buffer 的可用空间，这个值直接影响着 PCIe Link 的性能。Scaled Flow Control 扩展了流控 DLLP，其中加入了 Hdr Scale 字段和 Data Scale 字段，用于对 Credit 的数量进行成倍的放大。见 Table3-2，Figure 3-7，Figure 3-8 和 Figure 3-9。

引入 Scaled Flow Control 功能之后，PCIe 的 Data Link 也发生了相应的变化。与 PCIe 3.0 相比, PCIe 多了一个"DL_Feature"状态。不过，要进入 DL_Feature 状态还是有条件的，需要在 Data Link Feature Capabilities Register 中使能 Data Link Feature Exchange 字段。在 Flow Control 中，DL_Feature 这个状态主要用于 Scaled Flow Control 功能的握手交互(专业名称叫做 Handshake)。在这个过程中，

会用到一个叫做 Data Link Feature DLLP。

Data Link Feature DLLP 中 Byte1 bit[7]是 Feature Ack bit，就是用来握手的，Byte 3 bit[0]也就是 Feature support 的 bit[0]代表 Scaled Flow Control 是否 support，Feature support 的其他 bit[22:1]目前是保留的，以备将来之用。

允许所有端口支持 Scaled Flow Control。支持 16.0 GT/s 和更高数据速率的端口必须支持按 Scaled Flow Control。Scaled Flow Control 激活不会影响以 16.0 GT/s 和更高数据速率运行的能力。

当未为链路激活 Scaled Flow Control 时，以下规则适用：

- 发送和接收的 InitFC1、InitFC2 和 UpdateFC DLLP 的 HdrScale 和 DataScale 字段必须为 00b。
- HdrFC 计数器为 8 比特宽，且 HdrFC DLLP 字段包括此计数器的所有比特。
- DataFC 计数器为 12 比特宽，且 DataFC DLLP 字段包括此计数器的所有比特。

当为链路激活 Scaled Flow Control 时，以下规则适用：

- InitFC1 和 InitFC2 DLLP 在 HdrScale 字段中必须包含 01b、10b 或 11b。该值由 Table 3-2 中定义的指定信用类型的 outstanding 的最大 Header 信用数确定。
- InitFC1 和 InitFC2 DLLP 在 DataScale 字段中必须包含 01b，10b 或 11b。该值由 Table 3-2 中定义的指示信用类型的 outstanding 的数据有效载荷信用的最大数量确定。
- 如果在状态 FC_INIT1 中记录的接收到的 HdrScale 和 DataScale 值不为零，则在此 VC 上启用 Scaled Flow Control，并且 UpdateFC DLLP 在 HdrScale 和 DataScale 字段中必须包含 01b，10b 或 11b。
- 如果在状态 FC_INIT1 中记录的接收到的 HdrScale 和 DataScale 值为零，则在此 VC 上未启用 Scaled Flow Control，并且 UpdateFC DLLP 在 HdrScale 和 DataScale 字段中必须包含 00b。

Table 3-2: Scaled Flow Control Scaling Factors

Scale Factor	Scaled Flow Control Supported	Header Credits					Data Credits				
		Min Credits	Max Credits	HdrFC Width	HdrFC DLLP field		Min Credits	Max Credits	DataFC Width	DataFC DLLP Field	
					Transmitted	Received				Transmitted	Received
00b	No	1	127	8 bits	HdrFC	HdrFC	1	2047	12 bits	DataFC	DataFC
01b	Yes	1	127	8 bits	HdrFC	HdrFC	1	2047	12 bits	DataFC	DataFC
10b	Yes	4	508	10 bits	HdrFC>>2	HdrFC<<2	4	8188	14 bits	DataFC>>2	DataFC<<2
11b	Yes	16	2032	12 bits	HdrFC>>4	HdrFC<<4	16	32,752	16 bits	DataFC>>4	DataFC<<4

3.5 Data Link Layer Packets (DLLPs)

以下 DLLP 用于支持链路操作：

- Ack DLLP ： TLP Sequence number 应答；用于指示成功接收到一定数量的 TLP。
- Nak DLLP ： TLP Sequence number 否定确认；用于启动数据链路层重试。
- InitFC1、InitFC2 和 UpdateFC DLLP ： 用于流控制。
- 电源管理 DLLP。

3.5.1 Data Link Layer Packet Rules

形成 DLLP 时，所有标记为“Reserved”（缩写为 R）的 DLLP 字段都必须用全 0 填充。接收者必须忽略此类字段中的值。每种情况都指定了编码字段中保留值的处理。

所有 DLLP 都包括以下字段，如 Figure 3-4：

- DLLP Type —— DLLP 类型编码字段，如 Table 3-3。
- 16-bit CRC 字段。

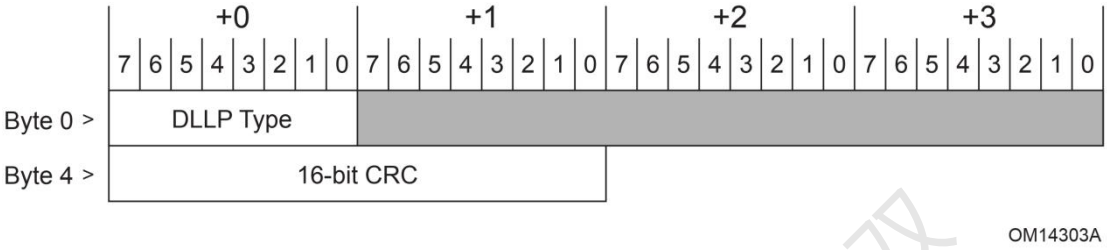


Figure 3-4 DLLP Type and CRC Fields

Table 3-3: DLLP Type Encodings

Encodings (b)	DLLP Type
0000 0000	Ack
0000 0001	MRInit – See the MR-IOV Specification[41]
0000 0010	Data_Link_Feature
0001 0000	Nak
0010 0000	PM_Enter_L1
0010 0001	PM_Enter_L23
0010 0011	PM_Active_State_Request_L1
0010 0100	PM_Request_Ack
0011 0000	Vendor-specific
0011 0001	NOP
0100 $0v_2v_1v_0$	InitFC1-P (v[2:0] specifies Virtual Channel)
0101 $0v_2v_1v_0$	InitFC1-NP
0110 $0v_2v_1v_0$	InitFC1-Cpl
0111 $0v_2v_1v_0$	MRInitFC1 (v[2:0] specifies Virtual Link) – See the MR-IOV Specification[42]
1100 $0v_2v_1v_0$	InitFC2-P
1101 $0v_2v_1v_0$	InitFC2-NP
1110 $0v_2v_1v_0$	InitFC2-Cpl
1111 $0v_2v_1v_0$	MRInitFC2 – See the MR-IOV Specification[42]
1000 $0v_2v_1v_0$	UpdateFC-P
1001 $0v_2v_1v_0$	UpdateFC-NP
1010 $0v_2v_1v_0$	UpdateFC-Cpl
1011 $0v_2v_1v_0$	MRUpdateFC – See the MR-IOV Specification[42]
All other encodings	Reserved

- 对 Ack 和 Nak DLLP（see Figure 3-5）：
  - AckNak_Seq_Num 字段用于指示哪些 TLP 受到影响。

- 数据链路层根据 Section 3.6 中提供的规则发送和接收。

- 对于 InitFC1、InitFC2 和 UpdateFC DLLP：
  - HdrFC 字段指 P、NP 或 Cpl 类型报头的信用值。
  - DataFC 字段指 P、NP 或 Cpl 类型数据负载的信用值。
  - HdrScale 字段包含对应类型的 Header 的缩放因子。编码在 Table 3-4 中定义。
  - DataScale 字段包含对应类型的有效负载数据的缩放因子。编码在 Table 3-4 中定义。

Table 3-4 HdrScale and DataScale Encodings

HdrScale or DataScale Value	Scaled Flow Control Supported	Scaling Factor	HdrFC DLLP Field	DataFC DLLP Field
00b	No	1	HdrFC[7:0]	DataFC[11:0]
01b	Yes	1	HdrFC[7:0]	DataFC[11:0]
10b	Yes	4	HdrFC[9:2]	DataFC[13:2]
11b	Yes	16	HdrFC[11:4]	DataFC[15:4]

  - 如果激活了 Scaled Flow Control，则必须在所有传输的 InitFC1，InitFC2 和 UpdateFC DLLP 中将 HdrScale 和 DataScale 字段设置为 01b，10b 或 11b。
  - 在 UpdateFC 中，仅允许发送器在支持 Scaled Flow Control 的情况下在 HdrScale 和 DataScale 字段中发送非零值，并且在此 VC 的 InitFC1 和 InitFC2 中接收到 HdrScale 和 DataScale 的非零值。
  - InitFC1、InitFC2 和 UpdateFC DLLP 包格式如 Figure 3-7、Figure 3-8 和 Figure 3-9 所示。
  - 当数据链路层初始化虚通道的流控制（请参见 Section 3.4），事务层根据 Section2.6 中的规则对流控制进行初始化后，触发传输。
  - 在数据链路层接收时检查完整性，如果正确，则将 DLLP 的信息内容传递到事务层。如果检查失败，则信息将被丢弃。

- 电源管理 DLLP（see Figure 3-10）：
  - 根据 Chapter 5 中的电源管理逻辑规则触发。
  - 在数据链路层接收时检查完整性，然后传递到组件的电源管理逻辑。

- Vendor Specific DLLP（see Figure 3-11）：
  - 建议接收者静默丢弃 Vendor Specific DLLP，除非由特定于实现的机制启用。
  - 建议发送者不要发送 Vendor Specific DLLP，除非由特定于实现的机制启用。

- NOP DLLP（see Figure 3-6）
  - 接收者应在检查 DLLP 的数据完整性后不采取任何措施并将其丢弃。

注：这是不支持 DLLP Type 编码的更通用规则的特殊情况（请参见 Section 3.6.2.2）。

- Data Link Feature DLLP（see Figure 3-12）
  - Feature Ack 字段设置为 1 表示发送端口已接收到一个 Data Link Feature DLLP。

公众号：芯片业者相关权

- Feature Supported 字段表示发送端口支持的 feature。这些位等于 Local Data Link Feature Supported 字段的值（请参见 Section 7.7.4.2）。

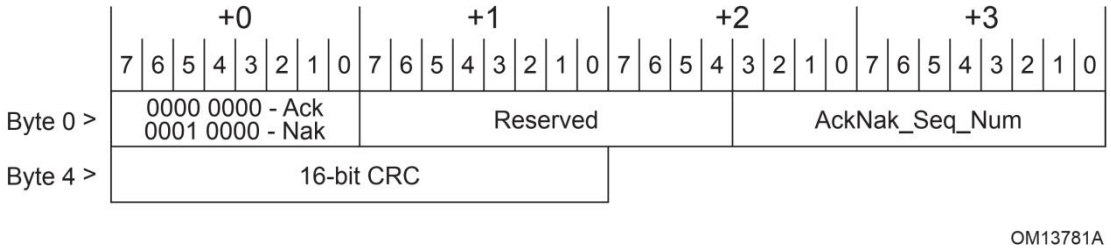


Figure 3-5 Data Link Layer Packet Format for Ack and Nak

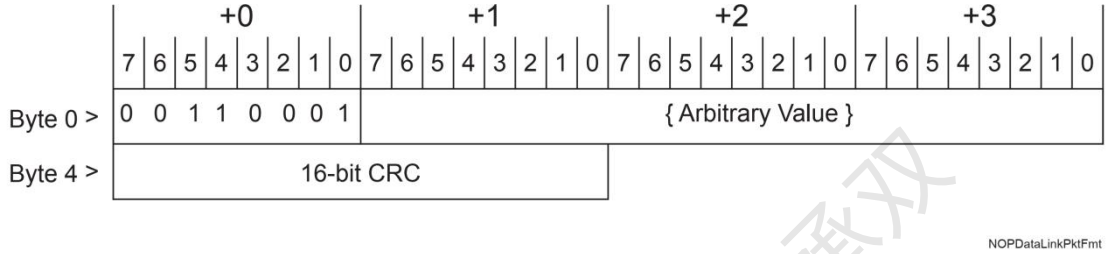


Figure 3-6 NOP Data Link Layer Packet Format

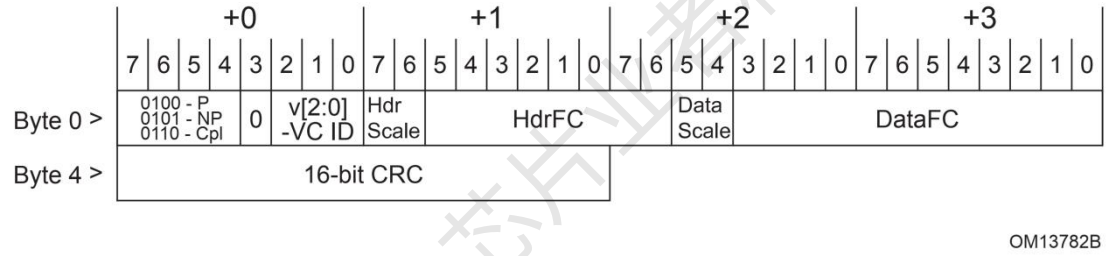


Figure 3-7 Data Link Layer Packet Format for InitFC1

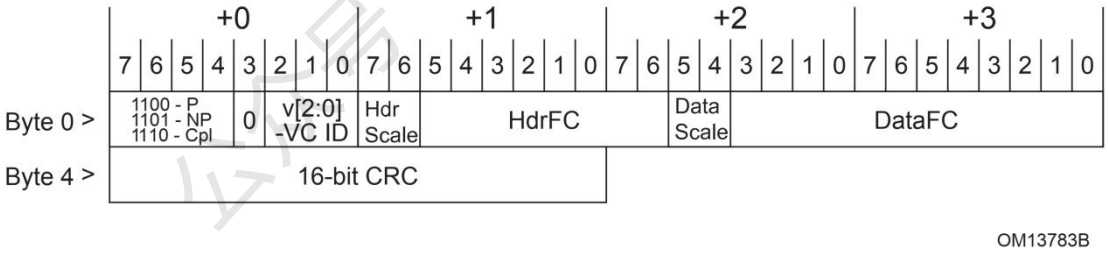


Figure 3-8 Data Link Layer Packet Format for InitFC2

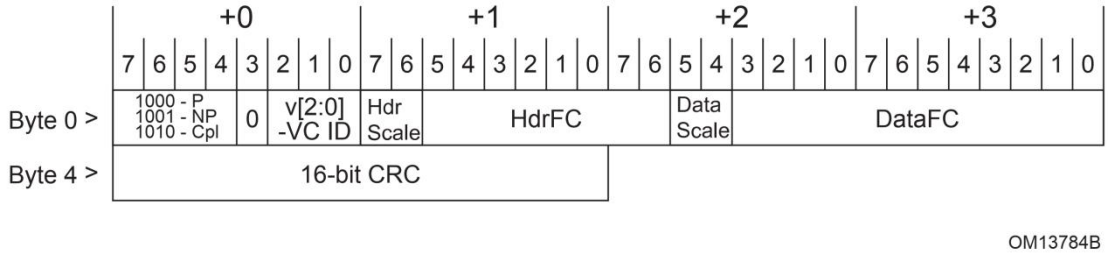


Figure 3-9 Data Link Layer Packet Format for UpdateFC

公众号·芯片业者相承双

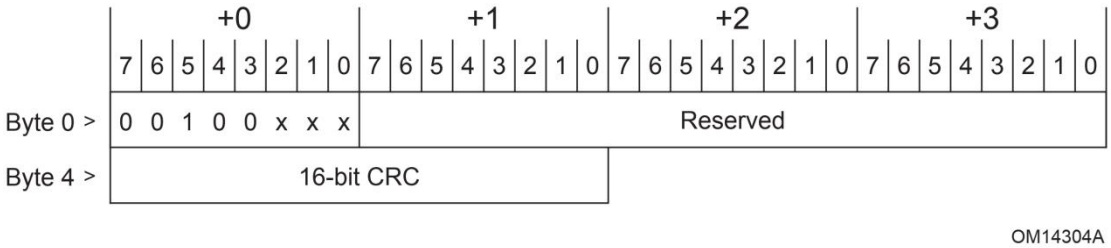


Figure 3-10 PM Data Link Layer Packet Format

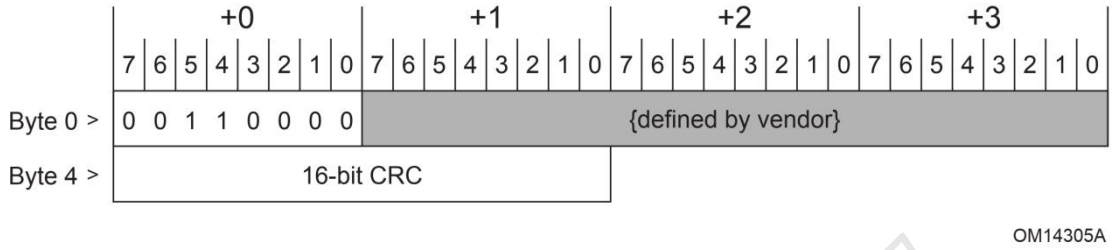


Figure 3-11 Vendor-specific Data Link Layer Packet Format

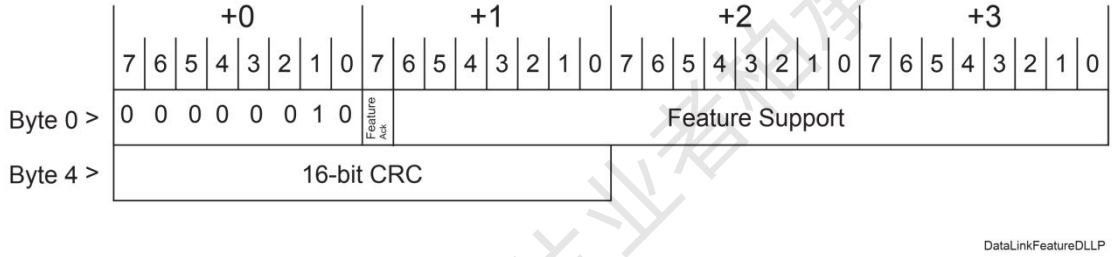


*Figure 3-12 **Data Link Feature DLLP** Format*

以下是与数据链路层数据包（DLLP）相关的特征和规则：

- 当 DLLP 呈现给物理层或从物理层接收时，它们与 TLP 不同。
- DLLP 使用一个 16-bit 的 CRC 保护 DLLP 完整性。
- 该 16-bit CRC 计算规则如下（see Figure 3-13）：
  - 用于 CRC 计算的多项式的系数表示为 100Bh。
  - seed 值为 FFFFh。
  - CRC 计算从 byte 0 的 bit 0 开始，从每个字节的 bit 0 到 bit 7。
  - 请注意，CRC 计算使用 DLLP 的所有位，而与字段类型无关，包括保留字段。计算结果经过补充，然后放入 DLLP 的 16 位 CRC 字段中，如 Table 3-5 所示。

公众号·芯片业者相承双

Table 3-5 Mapping of Bits into CRC Field

CRC Result Bit	Corresponding Bit Position in the 16-Bit CRC Field
0	7
1	6
2	5
3	4
4	3
5	2
6	1
7	0
8	15
9	14
10	13
11	12
12	11
13	10
14	9
15	8

公众号·芯片业者相承双

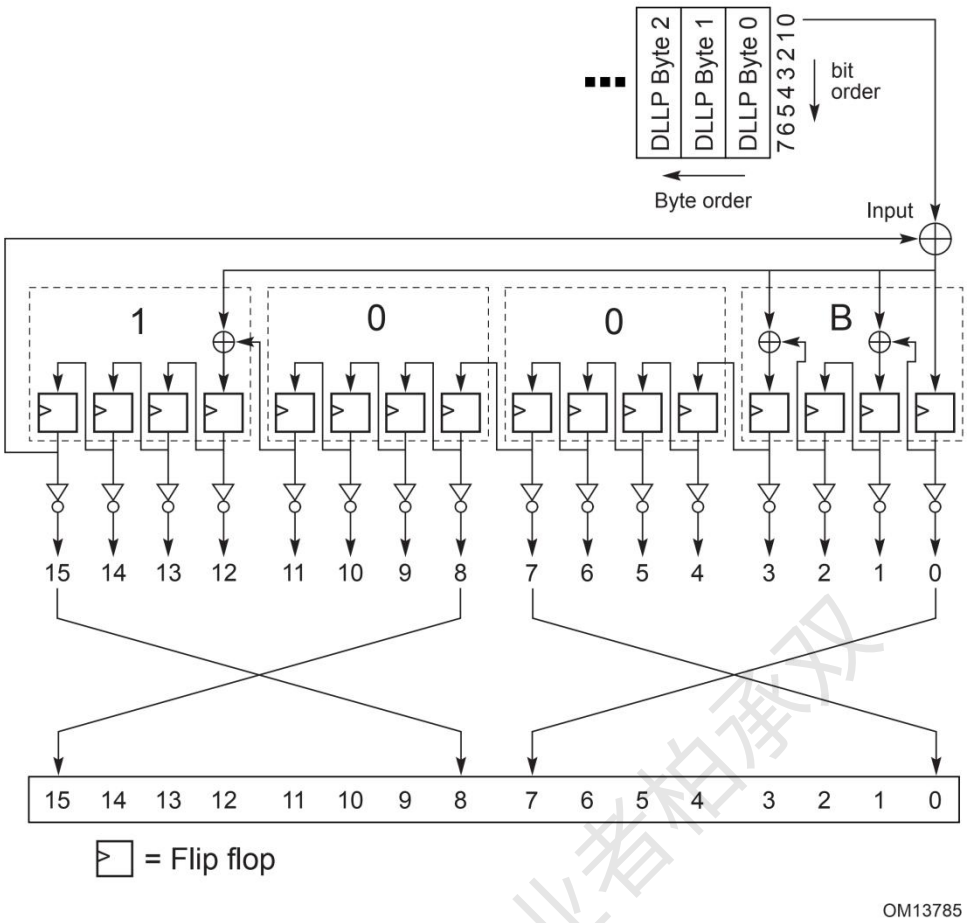


Figure 3-13 Diagram of CRC Calculation for DLLPs

3.6 Data Integrity Mechanisms

3.6.1 Introduction

事务层向数据链路层提供 TLP 边界信息。这允许数据链路层将 TLP Sequence Number 和 LCRC 应用于 TLP 的错误检测。接收数据链路层通过检查 TLP Sequence Number，LCRC 和来自接收物理层的任何错误指示来验证接收到的 TLP。如果这些错误中的任何一个在 TLP 中，则使用数据链路层重试进行恢复。

Figure 3-14 显示了添加了 TLP Sequence Number 和 LCRC 的 TLP 格式。

在支持协议复用的端口上，在 Symbol +0 中包含非零值的数据包，bit 7:4 是 PMUX 数据包。对于 TLP，这些位必须为 0000b。有关详细信息，请参见 Appendix G。

在不支持协议多路复用的端口上，Symbol +0 的 bit 7:4 被保留。

公众号：芯片业者怕承双

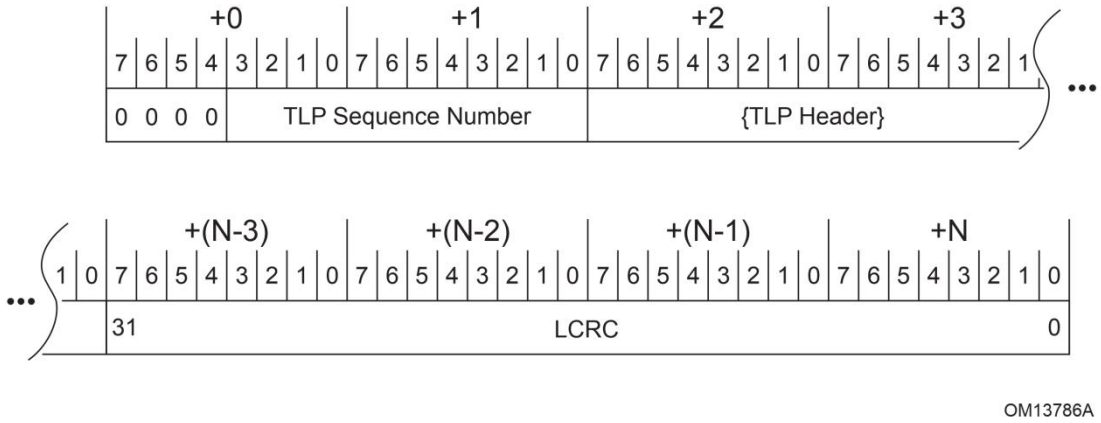


Figure 3-14 TLP with LCRC and TLP Sequence Number Applied

3.6.2 LCRC, Sequence Number, and Retry Management (TLP Transmitter)

数据链路层通过添加 TLP Sequence Number 和 LCRC 来确保每个 TLP 传输的完整性。TLP 存储在一个重试缓冲区中，除非从其他组件收到了肯定的接收确认，否则数据链路层会尝试重传。如果重复尝试发送 TLP 失败，则发送器认为链路不能正确运行，于是指示物理层重新训练链路。如果链路重新训练失败，则物理层将指示链路不再可用，从而导致 DLCMSM 转移到 DL_Inactive 状态。

概念上的“counters”和“flags”来描述 TLP LCRC 和 TLP Sequence Number 以及支持数据链路层重试的机制。此描述不暗示也不要求特定的实现，仅用于阐明要求。

3.6.2.1 LCRC and Sequence Number Rules (TLP Transmitter)

以下计数器和计时器用于解释本节中的其余规则：

- 使用了两个 12-bit 的计数器：
  - NEXT_TRANSMIT_SEQ —— 存储用于 TLP 的 Sequence Number。该计数器为每个将发送的新的 TLP 分配序列号，每个新 TLP 其值加 1，计数到 4095 时重新返回 0。
    - 在 DL_Inactive 状态设置为 0。
  - ACKD_SEQ —— 存储最近收到的 Ack 或 Nak DLLP 中的序列号（即 AckNak_Seq_Num 字段的值，NEXT_TRANSMIT_SEQ 与此值的差值的模表示有多少 TLP 未被响应）。
    - 在 DL_Inactive 状态设置为 FFFh。
- 使用了一个 2-bit 的计数器：
  - REPLAY_NUM —— 计算重试缓冲区已重新发送该 TLP 的次数。当 REPLAY_NUM 从 2’b11 翻转到 2’b00 时会触发一次物理层的链路重新训练，在重新训练之后，再次尝试重发该 TLP。无论何时收到 ACK，此计数器也会复位为 2’b00。
    - 在 DL_Inactive 状态设置为 0。
- 使用了下述计时器：
  - REPLAY_TIMER —— 根据以下规则计算需要重试的时间。或者理解为用来度量从发出 TLP 到收到对应的 ACK 或 NAK DLLP 的时间。
    - 从任何 TLP 传输或重传的最后一个符号开始。

公众号·芯片业者相关

- 对于每个重试，在发送要重传的第一个 TLP 的最后一个符号时，重置并重新启动 REPLAY_TIMER。
- 在还有更多未确认的 TLP 要发送时，当且仅当接收到的 Ack DLLP 确认重试缓冲区中的某些 TLP 时，对于收到的每个 Ack DLLP 重置并重新启动 REPLAY_TIMER。
  - ✧ 注意：这样可确保仅在 TLP 正在传递过程时重置 REPLAY_TIMER。
- 当收到 Nak、或当 REPLAY_TIMER 到期，需要重置 REPLAY_TIMER 直到重发条件满足。
- 链路重新训练期间 REPLAY_TIMER 值不增加（即当 LTSSM 处于 Recovery 或 Configuration 状态时保持其值）。请参阅 Section 4.2.5.3 和 Section 4.2.5.4。
- 如果支持协议多路复用，则可以选择在接收 PMUX 数据包期间不增加 REPLAY_TIMER 的值（PMUX 请参阅 Appendix G）。
- 如果以 Nak 响应的 TLP 都解决了，则 REPLAY_TIMER 重置并保持。

以下规则描述了 TLP 在传递到物理层之前如何准备进行传输：

- 事务层在传输 TLP 时会向数据链路层指示 TLP 的开始和结束。
  - 数据链路层将 TLP 视为“黑盒”，并且不处理或修改 TLP 的内容。
- 当数据链路层从事务层的发送端接受每个 TLP 时，会为其分配一个 12 位序列号。
  - 在从事务层接受 TLP 时，通过以下方式将数据包序列号应用于 TLP：
    - 把 12-bit 的 NEXT_TRANSMIT_SEQ 添加到 TLP 的头部。
    - 在序列号之前添加 4 bit 保留位（请参见 Figure 3-15）。
  - 如果 (NEXT_TRANSMIT_SEQ – ACKD_SEQ) mod 4096 >= 2048 为真，发送器必须停止从事务层接受 TLP，直到等式不再成立为止。这还会报告一个致命的、无法修正的数据链路层协议错误。
  - 在将 NEXT_TRANSMIT_SEQ 应用于从事务层的发送方接受的 TLP 之后，NEXT_TRANSMIT_SEQ 递增（除非 TLP 为空）：

    NEXT_TRANSMIT_SEQ := (NEXT_TRANSMIT_SEQ + 1) mod 4096

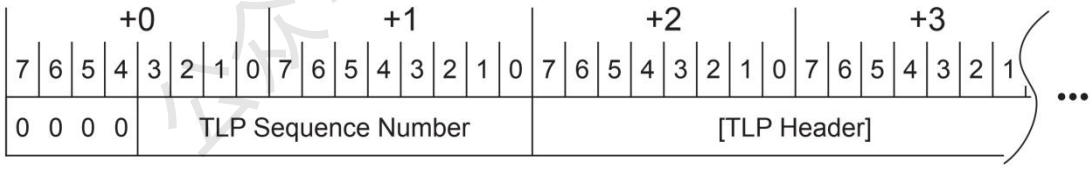


Figure 3-15 TLP Following Application of TLP Sequence Number and Reserved Bits

- 数据链路层使用 32-bit LCRC 保护 TLP 数据完整性。
- 使用以下机制计算 LCRC 值（see Figure 3-16）：
  - 多项式系数为 04C11DB7h。
  - seed 值为 FFFFFFFFh。
  - 序列号和 TLP 用于计算 LCRC（see Figure 3-15）。
  - LCRC 计算从 byte 0 的 bit 0（TLP 序列号的 bit 8）开始，并从每个连续字节的 bit 0 到 bit 7 进行。

公众号：芯片业者相承双

- 请注意，LCRC 计算使用 TLP 的所有位，而与字段类型无关，包括保留字段。

■ 补全 LCRC 计算的其余部分，并将补全后的结果位映射到 32 位 LCRC 字段中，如 Table 3-6 所示。

Table 3-6 Mapping of Bits into LCRC Field

LCRC Result Bit	Corresponding Bit Position in the 32-Bit LCRC Field
0	7
1	6
2	5
3	4
4	3
5	2
6	1
7	0
8	15
9	14
10	13
11	12

公众号·芯片业者相承双

LCRC Result Bit	Corresponding Bit Position in the 32-Bit LCRC Field
12	11
13	10
14	9
15	8
16	23
17	22
18	21
19	20
20	19
21	18
22	17
23	16
24	31
25	30
26	29
27	28
28	27
29	26
30	25
31	24

- 从事务层收到这些字节之后，将 32 位 LCRC 字段附加到 TLP（请参见 Figure 3-16）。

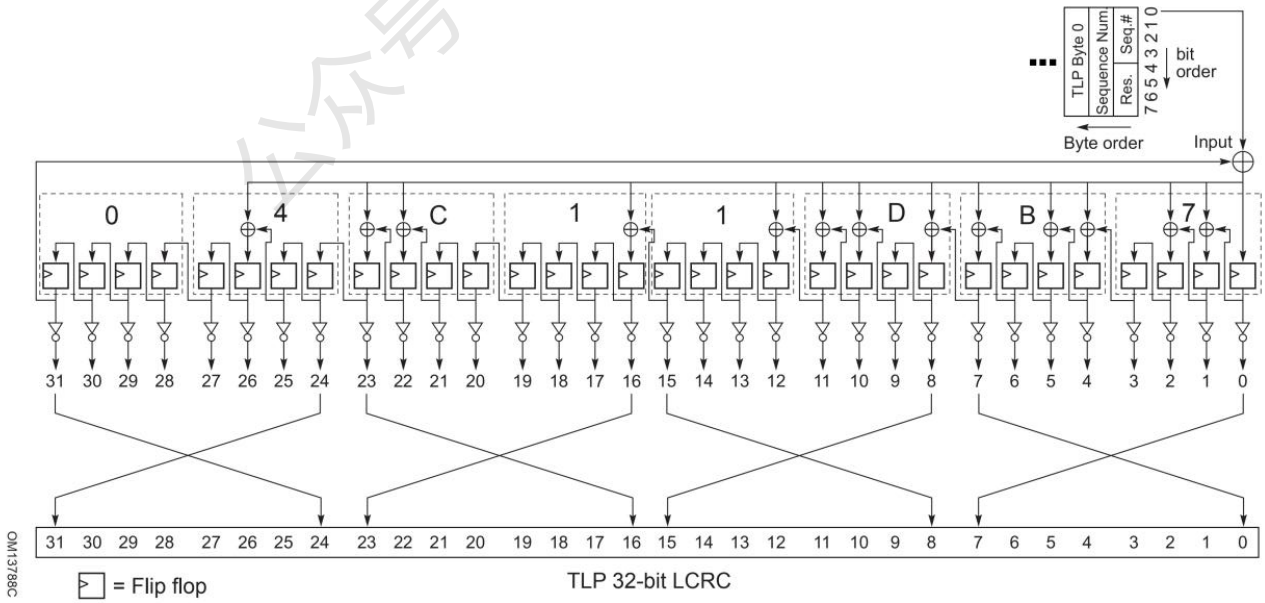


Figure 3-16 Calculation of LCRC

为了支持 TLP 的直通路由，允许发送器修改发送的 TLP，以指示接收器必须忽略该 TLP（“作废” TLP）。

- 允许发送器作废正在发送的 TLP；为了防止误解或损坏 TLP，发送器必须执行以下操作：

公众号·芯片业者相承双

- 当物理层使用 128b/130b 编码时，发送 TLP 的所有 DW（请参阅 Section 4.2.2.3.1）。
- 使用计算出的 LCRC 值的其余部分并且不进行反转（正常使用时 LCRC 值会逻辑反转）。
- 向物理层指示该 TLP 无效。

- 完成此操作后，发送器不会递增 NEXT_TRANSMIT_SEQ。

以下规则描述了数据链路层重试缓冲区的操作，必要时将从中重新发送 TLP：

- 传输的 TLP 副本必须存储在数据链路层重试缓冲区中，但无效的 TLP 除外。

当由于接收到 Nak 或由于 REPLAY_TIMER 到期而触发重发时，以下规则描述了必须遵循的操作顺序：

- 如果已确认所有传输的 TLP（Retry Buffer 为空），请终止重发，否则继续。
- REPLAY_NUM 递增。当通过接收确认重试缓冲区中某些 TLP 的 Nak 触发重发时，将重置 REPLAY_NUM。 然后允许（但不是必需）递增。
  - 如果 REPLAY_NUM 从 11b 翻转到 00b，发送器将向物理层发送信号以重新训练链路，并在重新进行重发之前等待重新训练的完成。这会报告与端口相关的错误（请参见 Section 6.2）。

  Note：除非物理层报告 Physical LinkUp = 0b（导致数据链路控制和管理状态机转换到 DL_Inactive 状态），否则此操作不会重置数据链路层状态（包括重试缓冲区的内容）。

  - 如果 REPLAY_NUM 不能从 11b 翻转到 00b，请继续重发。
- 阻止从事务层接受新的 TLP。
- 完成当前正在传输的任何 TLP 的传输。
- 重传响应了 Nak 的 TLP，从最早响应 Nak 的 TLP 开始，以最初传输顺序发送。
  - 发送第一个要重传的 TLP 的最后一个符号时，重置并重新启动 REPLAY_TIMER。
  - 一旦所有响应 Nak 的 TLP 被重新发送，请返回正常操作。
  - 如果在重发期间收到任何 Ack 或 Nak DLLP，则允许发送器完成重发而无需考虑 Ack 或 Nak DLLP，或者跳过任何新确认的 TLP。
    - 发送器一旦开始重新发送 TLP，则在所有情况下都必须完成该 TLP 的发送。
  - 重发期间收到的 Ack 和 Nak DLLP 必须进行处理，并且可能会折叠。
    - 如果收到多个 Ack，则仅应考虑指定最新序列号值的 Ack——指定较早序列号值的 Ack 实际上会“折叠”到该 Ack 中。
    - 在重发期间，会收到 Nak，接着是一个 Ack，该 Ack 指定了后面的序列号—— Ack 取代了 Nak，而 Nak 被忽略。

    注意：由于发送器的流控制选通逻辑已在接收器中为重试缓冲区中的所有条目分配了空间，因此无需进一步进行流控制同步。
- 事务层重新启用 TLP 的接收。

重发可以通过 REPLAY_TIMER 到期或接收到 Nak 来启动。以下规则涵盖 REPLAY_TIMER 的有效期：

TLP 传输过程中有一个重传(Replay)机制，当经过"一段时间"后，发送端仍然没有收到接收端返回的 Ack 或者 Nak 信息，就会触发重新发送 TLP。这里的"一段时间"有个专业名字叫做"REPLAY_TIMER"。在 PCIe 3.0 Spec 中，对于 REPLAY_TIMER 的定义,不偏不倚，

公众号·芯片业者相濡以沫

2.5GT/s, 5GT/s, 8GT/s 三个传输速率均有专属的定义，参【PCIE3.0】Table 3-4、Table 3-5 和 Table 3-6。

到了 PCIe 4.0，PCI-SIG 协会自己都不想再为 16GT/s 单独增加另外一个 REPLAY_TIMER Table。于是，提出了更加简便的 REPLAY_TIMER Limits。这下，对于所有的速率就都只有两个选择了，如下 Extended Synch 字段所描述。Extended Synch 字段在 Link Control Register 中设置。

对于这个简化的 REPLAY_TIMER，当 PCIe 链路运行在 2.5GT/5.0GT/8.0GT 时，实现时可以选择无视，依然选择原本的定义方式。但是，当 PCIe 链路运行在 16GT/s 或以上时，就必须实现 Simplified REPLAY_TIMER Limit。

- 如果发送重试缓冲区包含未收到 Ack 或 Nak DLLP 的 TLP，并且（如 REPLAY_TIMER 所示）在超过 REPLAY_TIMER 限制的时间内未收到 Ack 或 Nak DLLP，则发送器将启动 Replay。
  - Simplified REPLAY_TIMER Limit 为：
    - 当 Extended Synch 字段清 0 时，为 24000 到 31000 个 Symbol Time 。
    - 当 Extended Synch 字段置 1 时，为 80000 到 100000 个 Symbol Time 。
    - 如果 Extended Synch 字段在未确认的 TLP 处于 outstanding 时改变状态，则允许实现在 Extended Synch 字段改变状态时或下次复位 REPLAY_TIMER 时调整其 REPLAY_TIMER Limit。
  - 支持 16.0 GT/s 或更高数据速率的实现必须使用 Simplified REPLAY_TIMER Limit，以便在所有数据速率下进行操作。
  - 强烈建议仅支持低于 16.0 GT/s 的数据速率的实现在所有数据速率下使用 Simplified REPLAY_TIMER Limit，但依然允许它们使用[PCIe-3.1]中描述的 REPLAY_TIMER Limit。

  这是一个重发计时器超时错误，是与端口相关的报告错误（请参见 Section 6.2）。

IMPLEMENTATION NOTE

Determining REPLAY_TIMER Limit Values

Replay 主要由 Nak DLLP 启动，而 REPLAY_TIMER 作为辅助机制。由于它是辅助机制，因此 REPLAY_TIMER Limit 对跨链路传输 TLP 所需的平均时间影响相对较小。本规范又定义了 Simplified REPLAY_TIMER Limit，因此与本规范的先前版本一样，无需对 ASPM L0s，Retimer 或其他项进行调整。

TLP 发送器和一致性测试必须根据在 TLP 发送器端口上测量的重发时序。该时序以发送的 TLP 的最后一个符号或接收到的 Ack DLLP 的最后一个符号开始，主要看哪个是最早响应 Nak 的 TLP。该时序以 TLP 重传的第一个符号结尾。

在测量重发时序到 TLP 重发开始的时间点时，一致性测试必须允许在该方向上已经在进行任何其他 TLP 或 DLLP 传输（从而防止 TLP 重发）。

IMPLEMENTATION NOTE

Recommended Priority of Scheduled Transmissions

如果计划传输多个相同类型的 DLLP 但尚未传输，则在多数情况下有可能将它们“折叠”为单个 DLLP。例如，如果已调度的 Ack DLLP 传输停顿以等待另一次传输完成，并且在此期间已调度另一 Ack 进行传输，则仅需要传输第二个 Ack，因为它提供的信息将取代第一个 Ack 中的信息。

除了来自事务层（或重试缓冲区，如果正在进行重发）的任何 TLP 之外，还可以同时调度多个不同类型的 DLLP 进行传输，此时必须对这些 DLLP 区分优先级进行传输。以下列表显示了选择传输信息的首选优先级顺序。请注意，未列出特定于供应商的 DLLP 的优先级，因为这完全是特定于实现的，因此没有建议的优先级。请注意，此优先级顺序是一个准则，并且在所有情况下都强烈建议采用一种公平机制，以确保任何类型的业务流不会导致任何其他类型的业务流长时间或无限期地被阻塞。请注意，Ack Latency 值和 REPLAY_TIMER 限制指定了在组件的端口上测量的要求，并且组件的内部仲裁策略必须确保满足这些外部测量的要

公众号：芯片业者相濡以沫

求。

1. 当前正在进行的任何传输（TLP 或 DLLP）的完成包（最高优先级）。
2. Nak DLLP。
3. 由于以下原因，需要尽快安排 Ack DLLP 传输：收到一个副本 TLP（重传过来的 TLP）或 Ack 延迟计时器到期（see Section 3.6.3.1）。
4. 满足 Section 2.6 要求的 FC DLLP 传输。
5. 重试缓冲区重传。
6. 来自事务层的 TLP。
7. 不满足 Section 2.6 要求的 FC DLLP 传输。
8. 所有其他 DLLP 的传输（最低优先级）。

3.6.2.2 Handling of Received DLLPs

由于 Ack/Nak 和 Flow Control DLLP 影响对端发送过来的 TLP，因此数据链路层中的 TLP 传输机制还负责处理从链路上其他组件接收的 Ack/Nak 和 Flow Control DLLP。这些 DLLP 是根据以下规则处理的（请参阅 Figure 3-17）：

- 如果物理层指示接收器错误，则丢弃当前正在接收的任何 DLLP 并释放为该 DLLP 分配的任何存储。请注意，将此类错误报告给软件是由物理层完成的（因此，数据链路层不会报告）。
- 对所有收到的 DLLP 执行以下 CRC 检查过程：
  - 对接收的 DLLP 应用于发送相同的 DLLP 的算法，不包括接收的 DLLP 的 16-bit CRC 字段。
  - 比较该计算值与 DLLP 中携带的 16-bit CRC 的值。
    - 如果不相等，则 DLLP 被损坏。
  - 接收到损坏了的 DLLP 将被丢弃，并且是与端口相关的报告错误（请参见 Section 6.2）。
- 接收到的未损坏但使用不受支持的 DLLP Type 编码的 DLLP 会被丢弃，而无需采取进一步措施。这不被视为错误。
- 保留字段中的非零值将被忽略。
- 接收者必须按照接收到的速率处理所有接收到的 DLLP。

公众号：芯片业者相关双

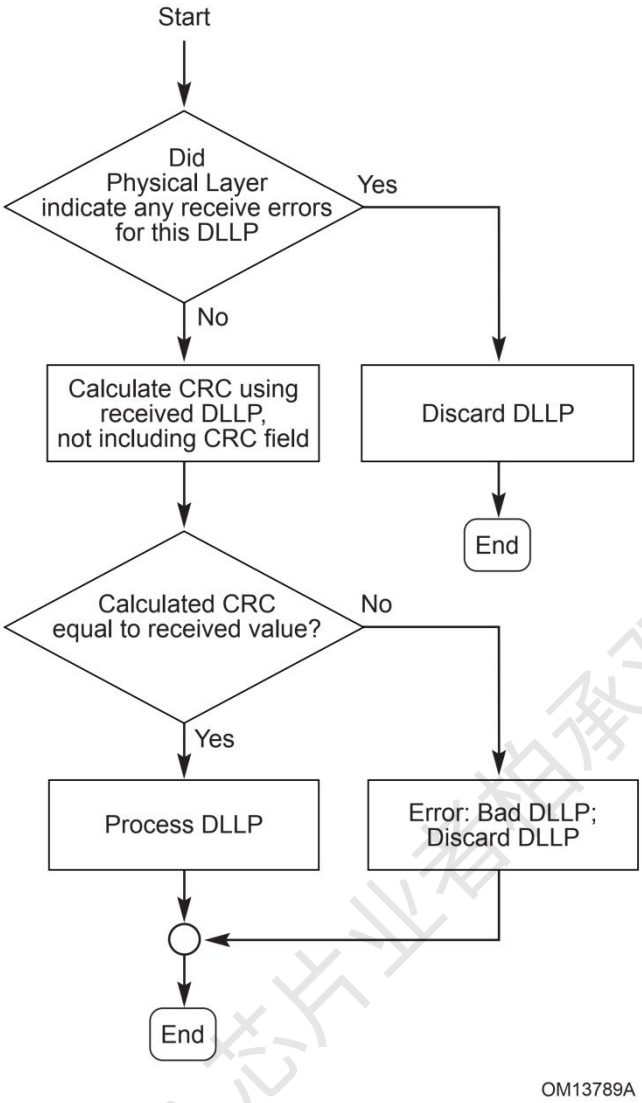


Figure 3-17 Received DLLP Error Check Flowchart

- 把接收到的 FC DLLP 传递到事务层。
- 把接收到的 PM DLLP 传递到组件的电源管理控制逻辑。
- 对于 Ack 和 Nak DLLP，请遵循以下处理步骤（请参阅 Figure 3-18）：
  - 如果由 AckNak_Seq_Num 指示的 Sequence Number 不与任何响应 Nak 的 TLP 对应，或不与 ACKD_SEQ 值对应，则 DLLP 将被丢弃。
    - 这是一个数据链路协议错误，是一个与端口相关的报告错误（参见 Section 6.2）。

  Note：只要指示的 Sequence Number 与 ACKD_SEQ 值匹配，当 TLP（包括复位和第一次 TLP 传输之间的时间）全部被确认，接收 Ack DLLP 并不是错误。

  - 如果 AckNak_Seq_Num 没有指示为一个最近响应 Ack 的 TLP 的 Sequence Number，则用 Ack DLLP 响应重试缓冲区中的某些 TLP：
    - 从重试缓冲区中清除所有 TLP，从最早的到对应于 AckNak_Seq_Num 的 TLP。
    - 使用 AckNak_Seq_Num 字段中的值加载 ACKD_SEQ。
    - 重置 REPLAY_NUM 和 REPLAY_TIMER。

公众号·芯片业者相承双

- 如果 DLLP 是 Nak，则触发 Replay 过程（请参见上文）。

Note：收到 Nak 不报告错误。

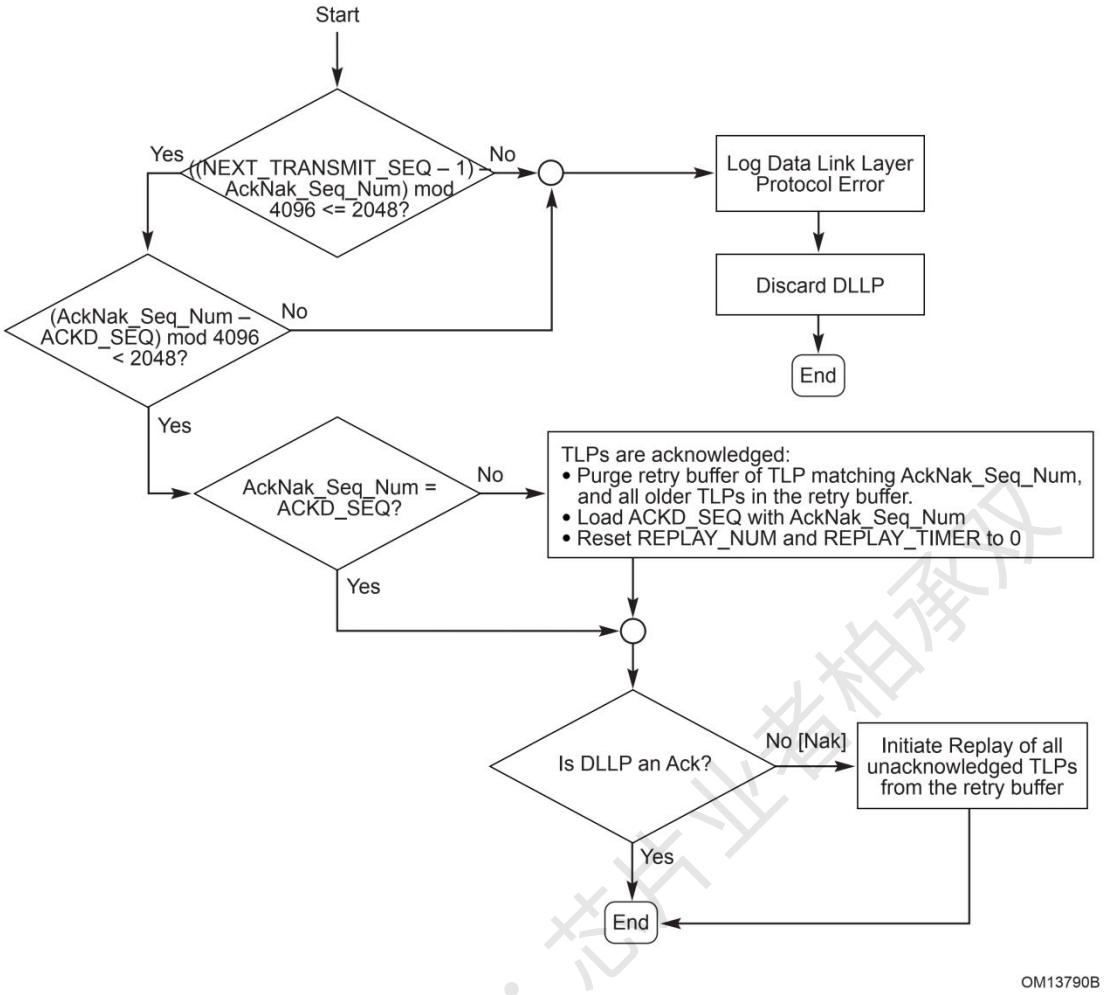


Figure 3-18 Ack/Nak DLLP Processing Flowchart

以下规则描述了数据链路层重试缓冲区的操作，必要时将从中重新发送 TLP：

- 发送的 TLP 副本必须存储在数据链路层重试缓冲区中。

3.6.3 LCRC and Sequence Number (TLP Receiver)

数据链路层的 TLP Receiver 负责检查 LCRC 和 TLP Sequence Number，并决定是否传递给事务层、或者请求重发。

概念上的“counters”和“flags”来描述 TLP LCRC 和 TLP Sequence Number 以及支持数据链路层重试的机制。此描述不暗示也不要求特定的实现，仅用于阐明要求。

3.6.3.1 LCRC and Sequence Number Rules (TLP Receiver)

- 使用了一个 12-bit 的计数器：
  - NEXT_RCV_SEQ —— 存储下一个 TLP 的期望 Sequence Number 值。
    - 在 DL_Inactive 状态重置为 0。

- 使用了下述 flag：
  - NAK_SCHEDULED
    - 在 DL_Inactive 状态重置为 0b。
- 使用了下述计时器：
  - AckNak_LATENCY_TIMER —— 根据以下规则对 Ack DLLP 何时调度传输的时间进行计数：
    - 在 DL_Inactive 状态重置为 0。
    - 每次调度发送 Ack 或 Nak DLLP 时该值都从 0 重新开始；用 Ack DLLP 确认收到的所有 TLP 时重置为 0。
    - 如果最初没有未确认的 TLP，然后接收到 TLP，则只有当 TLP 已转发到接收事务层时，AckNak_LATENCY_TIMER 才开始计数。

依次应用以下规则来描述如何处理接收到的 TLP，以及哪些事件触发了 Ack 和 Nak DLLP 的传输（请参见 Figure 3-19）：

- 如果物理层报告一个 Receiver Error，则丢弃当前正在接收的任何 TLP，并释放为该 TLP 分配的任何存储。请注意，将此类错误报告给软件是由物理层完成的（因此，数据链路层不会报告）。
  - 如果物理层在报告 Receiver Error 时正在接收 TLP，并且 NAK_SCHEDULED 标志清为 0，
    - 立即调度一个 Nak DLLP 发送出去。
    - 把 NAK_SCHEDULED 是为 1。
- 如果物理层报告接收到的 TLP 为空，并且 LCRC 为计算值的逻辑非，则丢弃 TLP 并释放为 TLP 分配的任何存储。这不被视为错误。
- 如果 TLP 无效，但 LCRC 与计算值的逻辑非匹配，则 TLP 损坏——丢弃 TLP 并释放为 TLP 分配的任何存储。
  - 如果 NAK_SCHEDULED 标志清为 0，
    - 立即调度一个 Nak DLLP 发送出去。
    - 把 NAK_SCHEDULED 是为 1。

    这是 Bad TLP 错误，是报告的与端口相关的错误（请参见 Section 6.2）。
- 按以下方式检查 LCRC：
  - 对接收到的 TLP 应用与上述计算相同的算法，不包括接收到的 TLP 的 32 位 LCRC 字段。
  - 把计算的值与数据链路层 TLP 携带的 LCRC 比较:
    - 如果不相等，则 TLP 损坏——丢弃 TLP 并释放为 TLP 分配的任何存储。
      - 如果 NAK_SCHEDULED 标志清为 0，
        - 立即调度一个 Nak DLLP 发送出去。
        - 把 NAK_SCHEDULED 是为 1。

      这是 Bad TLP 错误，是报告的与端口相关的错误（请参见 Section 6.2）。
- 如果 TLP Sequence Number 不等于期望值，则存储在 NEXT_RCV_SEQ 中：
  - 丢弃 TLP 并释放为 TLP 分配的任何存储。

公众号·芯片业者相承双

- 如果 TLP Sequence Number 满足以下方程式：

  (NEXT_RCV_SEQ - TLP Sequence Number) mod 4096 <= 2048

  则 TLP 是对方重发过来的，需要为此传输 Ack DLLP（根据传输优先级规则）。

- 否则，TLP 顺序不正确（指示一个或多个丢失的 TLP）：

  - ✧ 如果 NAK_SCHEDULED 标志清为 0，
    - ➢ 立即调度一个 Nak DLLP 发送出去。
    - ➢ 把 NAK_SCHEDULED 是为 1。

    这是 Bad TLP 错误，是报告的与端口相关的错误（请参见 Section 6.2）。

  无论 NAK_SCHEDULED 的状态如何，都可以将其报告为与端口相关的错误（请参见 Section 6.2 节），并且这种允许的行为如 Figure 3-17 所示。但是，为了防止错误污染，建议仅当 NAK_SCHEDULED 标志清零时，端口才报告此类错误。

- 如果 TLP Sequence Number 等于存储在 NEXT_RCV_SEQ 中的期望值：

  - 删除 4-bit 的保留位、TLP Sequence Number 和 LCRC（见 Figure 3-12），并发送到事务处理层。
    - ◆ 数据链路层在传输 TLP 时向事务层指示 TLP 的开始和结束。
      - ✧ 数据链路层将 TLP 视为“黑盒”，并且不处理或修改 TLP 的内容。
    - ◆ 请注意，在将 TLP 转发到事务层之前，接收方流控制机制不会考虑任何接收到的 TLP。
  - NEXT_RCV_SEQ 递增。
  - 如果 NAK_SCHEDULED 为 1，则清为 0。

公众号·芯片业者怕承双

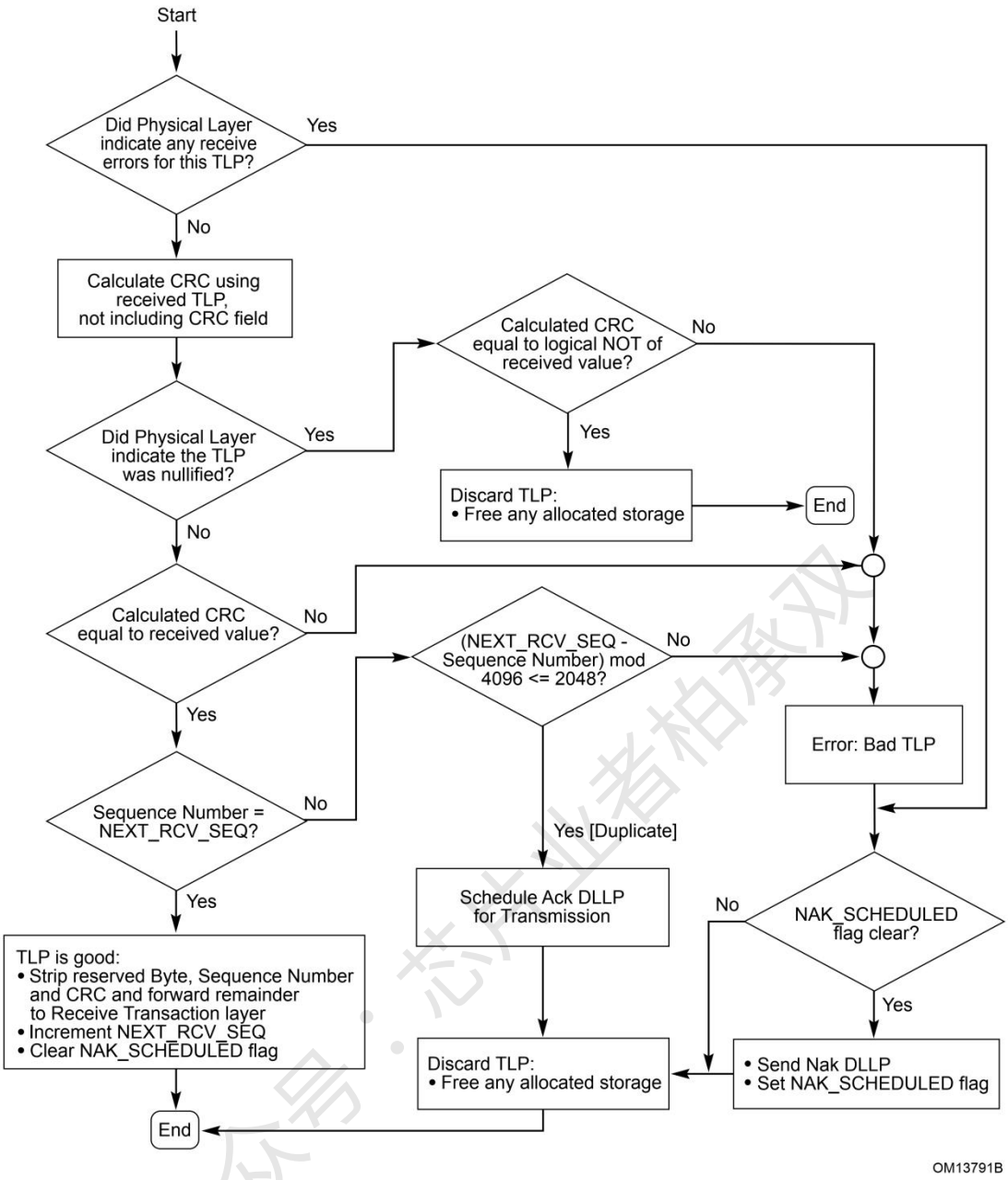


Figure 3-19 Receive Data Link Layer Handling of TLPs

- TLP 接收器必须调度一个 Ack DLLP，以便在满足以下所有条件的情况下进行传输：
  - 数据链路控制和管理状态机处于 DL_Active 状态。
  - TLP 已转发到接收事务层，但尚未通过发送 Ack DLLP 进行确认。
  - AckNak_LATENCY_TIMER 达到或超过 Table 3-7（对于 2.5 GT/s 操作模式），Table 3-8（对于 5.0 GT/s 操作模式）和 Table 3-9（对于 8.0 GT/s 以上操作模式）指定的值。
  - 用于 Ack DLLP 传输的链路已处于 L0 状中或已转换到 L0 状态。
  - Note：如果链路尚未在 L0 中，则链路必须转换为 L0 才能传输 Ack DLLP。
  - 当前没有在用于 Ack DLLP 传输的链路上传输另一个 TLP 或 DLLP。
  - NAK_SCHEDULED 清为 0。

公众号·芯片业者相承双

- Note：每次调度发送 Ack 或 Nak DLLP 时，都必须从 0 重新启动 AckNak_LATENCY_TIMER。

- 数据链路层 Ack DLLP 的调度时间可能比所需的时间更频繁。
- 为数据链路层 Ack 和 Nak DLLP 的 AckNak_Seq_Num 字段指定（NEXT_RCV_SEQ-1）值。

Table 3-7，Table 3-8 和 Table 3-9 定义了 AckNak_LATENCY_TIMER 的阈值，在任何特定情况下，该阈值都称为 Ack Latency Limit。

TLP 接收器和一致性测试必须根据在 TLP Receiver 的端口上测量的 Ack Latency 时序为基础，从接收到 TLP 的最后一个符号开始到 Ack DLLP 第一个符号发送出去结束。

在发送 Ack DLLP 之前进行测量时，一致性测试必须允许已经有 TLP 或其他 DLLP 在该方向上传输（从而保证了 Ack DLLP 传输）。如果 L0s 使能，则一致性测试必须允许 Ack DLLP 传输方向上链路的 L0s 退出延迟。如果 Link Control 寄存器的 Extended Synch 字段被设置，则一致性测试还必须考虑其对 L0s 退出延迟的影响。

不需要 TLP 接收器根据 L0s 退出延迟或 Extended Synch 的值来调整其 Ack DLLP 的调度。

Table 3-7 Maximum Ack Latency Limits for 2.5 GT/s (Symbol Times)

		Link Operating Width						
		x1	x2	x4	x8	x12	x16	x32
Max_Payload_Size (bytes)	128	237	128	73	67	58	48	33
	256	416	217	118	107	90	72	45
	512	559	289	154	86	109	86	52
	1024	1071	545	282	150	194	150	84
	2048	2095	1057	538	278	365	278	148
	4096	4143	2081	1050	534	706	534	276

Table 3-8 Maximum Ack Latency Limits for 5.0 GT/s (Symbol Times)

		Link Operating Width						
		x1	x2	x4	x8	x12	x16	x32
Max_Payload_Size (bytes)	128	288	179	124	118	109	99	84
	256	467	268	169	158	141	123	96
	512	610	340	205	137	160	137	103
	1024	1122	596	333	201	245	201	135
	2048	2146	1108	589	329	416	329	199
	4096	4194	2132	1101	585	757	585	327

公众号：芯片业者租赁网

Table 3-9 Maximum Ack Latency Limits for 8.0 GT/s and higher data rates (Symbol Times)

		Link Operating Width						
		x1	x2	x4	x8	x12	x16	x32
Max_Payload_Size (bytes)	128	333	224	169	163	154	144	129
	256	512	313	214	203	186	168	141
	512	655	385	250	182	205	182	148
	1024	1167	641	378	246	290	246	180
	2048	2191	1153	634	374	461	374	244
	4096	4239	2177	1146	630	802	630	372

IMPLEMENTATION NOTE

Retry Buffer Sizing

Retry Buffer 应足够大，以确保在正常操作条件下不会因为重试缓冲区满而导致传输被限制。在确定最佳缓冲区大小时，必须考虑 Ack Latency 值、由接收器已经发送另一 TLP 引起的 Ack 延迟、由物理链路互连引起的延迟以及处理接收到的 Ack DLLP 所需的时间。

给定两个组件 A 和 B，在调整 A 的发送重试缓冲区的大小时，应考虑 A 的 Receiver 要求的 L0s 退出延迟，如下例所示：

- A 在其到 B 的传输路径上退出 L0s，然后开始向 B 发送长突发的写请求。
- B 会在其通往 A 的传输路径上发起 L0s 退出，但是 A 的接收方所需的 L0s 退出时间很长。
- 同时，B 无法将 Ack DLLP 发送给 A，并且 A 由于缺少重试缓冲区空间而停止。
- 从 B 到 A 的传输路径返回到 L0，B 向 A 传输 Ack DLLP，并且停顿得到解决。

通过使组件的发送器重试缓冲区的大小与组件的接收器所需的 L0s 退出延迟匹配，或者从相反角度考虑，通过使接收器 L0s 的退出延迟与所需的重试缓冲区大小匹配，可以避免这种停顿。

选择 AckFactor 值是为了允许实现良好的性能，而不需要高成本的大重试缓冲区。为了在具有不同实现和应用程序的通用互连中实现一致的性能，有必要对所有组件设置相同的要求，而不考虑任何特定组件的应用程序空间。 如果组件不需要链路的全部传输带宽，则可以将其重试缓冲区的大小减小到指定的 Ack Latency 值保持可用的重试缓冲区空间所需的最小大小以下。

请注意，指定的 Ack Latency 值可确保允许的 outstanding TLP 序列号的范围永远不会成为导致传输停顿的限制因素。

公众号：芯片业者相承攻

4 Physical Layer Logical Block

4.1 Introduction

物理层规范分为逻辑子层和电气子层，如 Figure 4-1 所示。

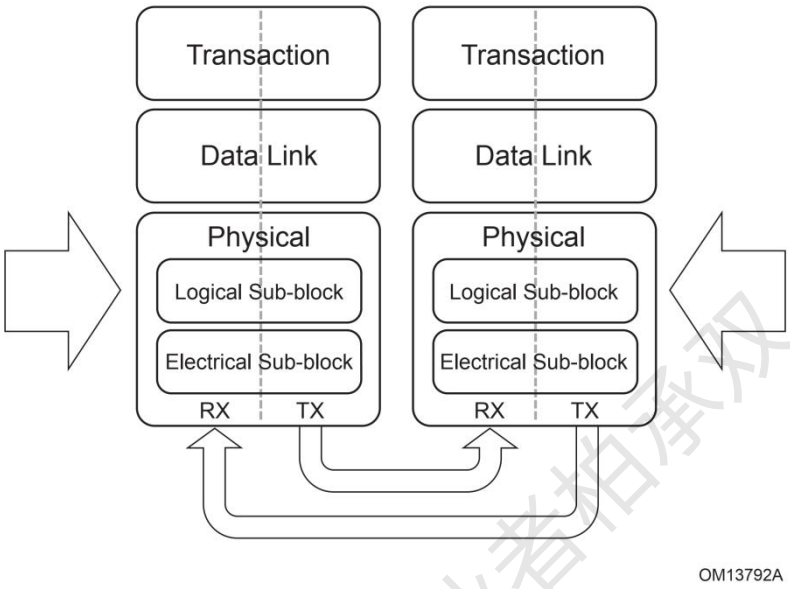


Figure 4-1 Layering Diagram Highlighting Physical Layer

4.2 Logical Sub-block

逻辑子层介于数据链路层与电气子层之间。

4.2.1 Encoding for 2.5 GT/s and 5.0 GT/s Data Rates

4.2.1.1 Symbol Encoding

在 2.5 GT/s 和 5.0 GT/s 下，PCIe 用 8b/10b 编码传输。这套传输码的定义符合 ANSI X3.230-1994，clause 11（同样也符合 IEEE 802.3z，36.2.4）的定义。这种方案将 8 bit 数据分成两个部分：3 bit 部分和 5 bit 部分，分别映射到一个 4 bit 编码和一个 6 bit 编码。控制位（Control<Z>）与数据字符（TX<7:0>和 RX<7:0>）一起用于识别何时对 8b / 10b 传输代码中包含的 12 个特殊符号之一进行编码。这些代码组串联在一起形成 10bit 的 symbol。如 Figure 4-2，ABCDE 映射到 abcdei，FGH 映射到 fghj。

注：在控制位的作用下，相同的 8bit 数据映射为不同的 10bit 符号；编码前的 8bit 数据可以称作 character（字符），编码后的 10 bit 数据可以称作 symbol（符号）。

公众号：芯片业者拍承双

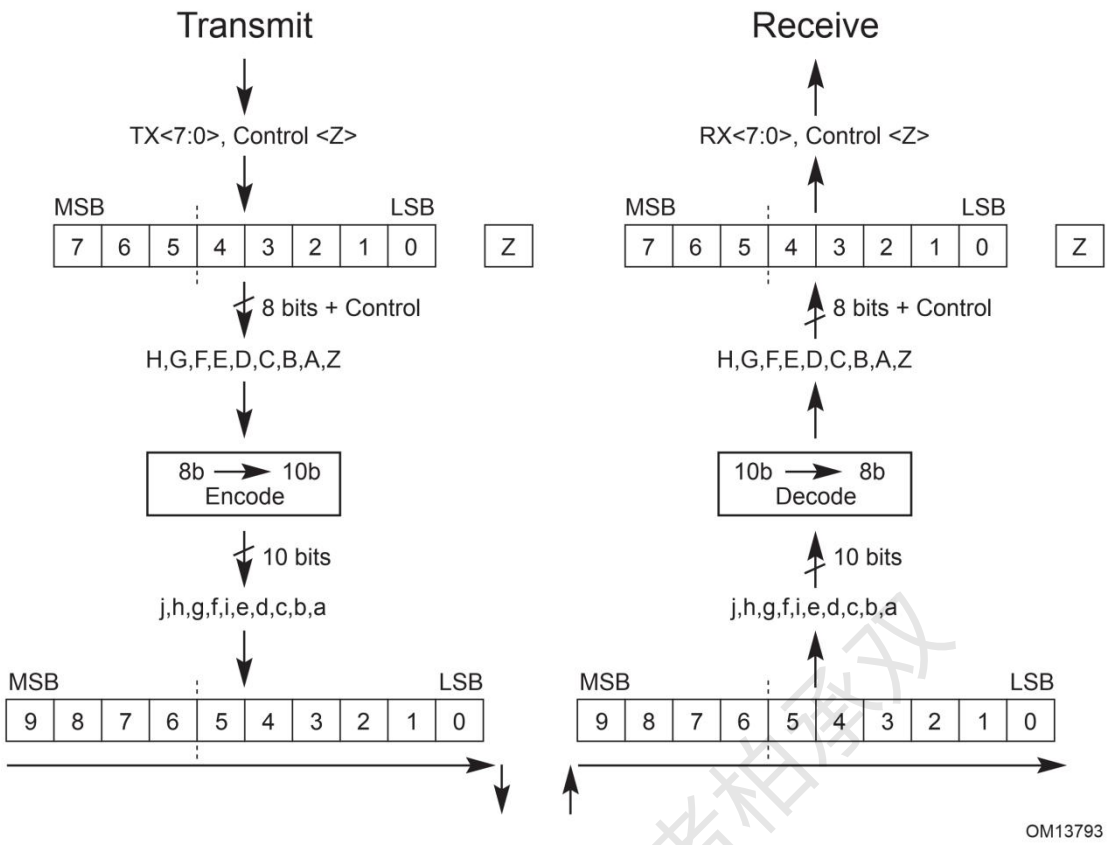


Figure 4-2 Character to Symbol Mapping

4.2.1.1.1 Serialization and De-serialization of Data

编码后的符号的比特流序如 Figure 4-3 与 Figure 4-4。

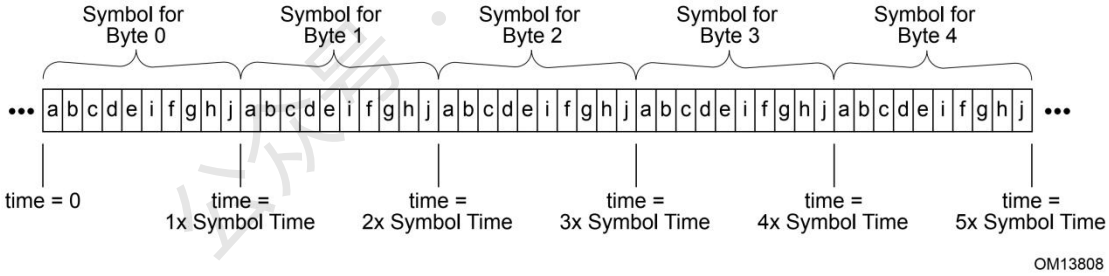


Figure 4-3 Bit Transmission Order on Physical Lanes - x1 Example

公众号：芯片业者柏承双

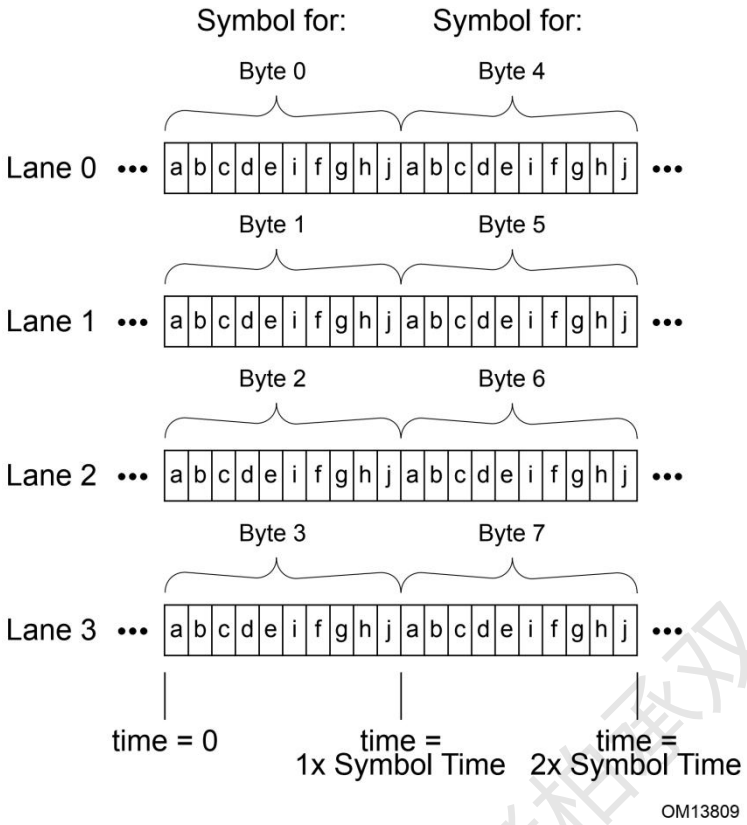


Figure 4-4 Bit Transmission Order on Physical Lanes - x4 Example

4.2.1.1.2 Special Symbols for Framing and Link Management(K code)

控制字符（K code）用于管理物理层数据传输，如 Table 4-1 。参考附录 B 的 Table B-2 获取这些控制字符的编码值。

Table 4-1 Special Symbols

Encoding	Symbol	Name	Description
K28.5	COM	Comma	Used for Lane and Link initialization and management
K27.7	STP	Start TLP	Marks the start of a Transaction Layer Packet
K28.2	SDP	Start DLLP	Marks the start of a Data Link Layer Packet
K29.7	END	End	Marks the end of a Transaction Layer Packet or a Data Link Layer Packet
K30.7	EDB	EnD Bad	Marks the end of a nullified TLP
K23.7	PAD	Pad	Used in Framing and Link Width and Lane ordering negotiations
K28.0	SKP	Skip	Used for compensating for different bit rates for two communicating Ports
K28.1	FTS	Fast Training Sequence	Used within an Ordered Set to exit from L0s to L0
K28.3	IDL	Idle	Used in the Electrical Idle Ordered Set (EIOS)
K28.4			Reserved
K28.6			Reserved
K28.7	EIE	Electrical Idle Exit	Reserved in 2.5 GT/s Used in the Electrical Idle Exit Ordered Set (EIEOS) and sent prior to sending FTS at data rates other than 2.5 GT/s

公众号·芯片业者相联双

4.2.1.1.3 8b/10b Decode Rules

8b/10b 解码按照 Appendix B 进行。如果收到的符号与 Appendix B 中任一个都不对应，就是符号错误，会向链路层报告此错误，参阅 Section 6.2 的描述。

4.2.1.2 Framing and Application of Symbols to Lanes

本节主要讲述物理层的组帧，以及将符号映射到各 Lane 上。

Symbol 的作用有两个：一个是有序集的传输，用于管理物理层；一个是用于传输 DLLP 和 TLP 信息的 Symbol。

K28.2 “SDP”用于开始一个 DLLP 传输，K27.7 “STP”用于开始一个 TLP 传输。K29.7 “END”用于一个 DLLP 或一个 TLP 传输的结束。

当没有数据包信息或特殊的有序集正在传输时，发送器处于逻辑空闲状态。在这段时间内，必须传输空闲数据。空闲数据必须由 0 组成，该数据按照 Section 4.2.1.3 的规则加扰，并且是按照 Section 4.2.1.1 的规则采用与 TLP 和 DLLP 相同的数据编码方式进行 8b/10b 编码。 同样，当接收方未接收到任何分组信息或特殊的有序集时，接收方处于逻辑空闲状态，并且应如上所述接收空闲数据。逻辑空闲期间，必须传输 SKP 有序集，参阅 Section 4.2.7 获取相关内容。

以下规则描述中的“place”的定义是指要求发送器将符号放入链路的对应 Lane 中。

- TLP 必须以一个 STP 符号开始，以一个 END（正常结束）符号或 EDB（异常结束）符号结束，如 Figure 4-5。
- DLLP 必须以一个 SDP 符号开始，以一个 END 符号结束，如 Figure 4-6。
- 逻辑空闲状态与电气空闲状态不同，逻辑空闲状态传输 Idle Symbol（00h）。
  - 逻辑空闲状态时，发送器必须在所有 Lane 上发送 Idle Symbol，参阅 Section 4.2.1.3。
  - 接收器必须忽略在逻辑空闲状态时接收到的信息。
- 对于宽度大于 x1 的链路，当从逻辑空闲链路条件开始发送 TLP 时，必须将 STP 符号放置在 Lane 0 中。
- 对于宽度大于 x1 的链接，从逻辑空闲链接条件开始发送 DLLP 时，必须将 SDP 符号放置在 Lane 0 中。
- 每个符号期间，STP 符号在链接上放置的频率不得超过一次。
- 每个符号期间，SDP 符号在链接上放置的频率不得超过一次。
- 只要满足上述规则，就可以允许 TLP 和 DLLP 的传输连续不断的进行。
- 一个 STP 符号和一个 SDP 符号可以在同一符号时间内放置在链路上。
  - 宽于 x4 的链路可以在 Lane 4*N 中放置 STP 和 SDP 符号，其中 N 是一个正整数。 例如，对于 x8，可以将 STP 和 SDP 符号放置在 Lane 0 和 4 中。对于 x16，可以将 STP 和 SDP 符号放置在 Lane 0、4、8 或 12 中。
- 对于 N 为 8 或更大的 xN 链路，如果将 END 或 EDB 符号放在 Lane K 中，其中 K 不等于 N-1，并且在 Lane K+1 中不跟 STP 或 SDP 符号（即，之后没有立即跟 TLP 或 DLLP），则必须将 PAD 符号放置在 Lane K+1 到 Lane N-1 上。
  - 例如，在 x8 链路上，如果将 END 或 EDB 放置在 Lane 3 上，如果其后跟的不是 STP 或 SDP，则必须将 PAD 放置在 Lane 4 至 7 中。
- EDB 符号用于标记无效 TLP 的结尾。有关 EDB 的用法，请参阅 Section 3.6.2.1。
- 接收者可以选择是否检查违反了本部分的规则。这些检查是独立可选的（参阅 Section 6.2.3.4）。如果实现此功能，则要报接收器错误，请参阅 Section 6.2。

公众号：芯片业者相承双

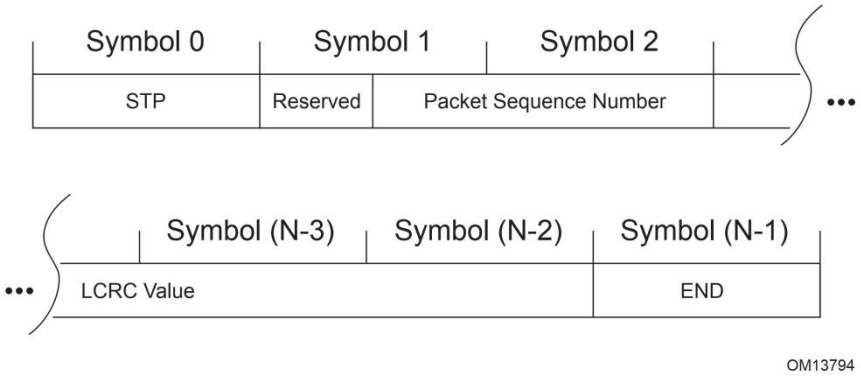


Figure 4-5 TLP with Framing Symbols Applied

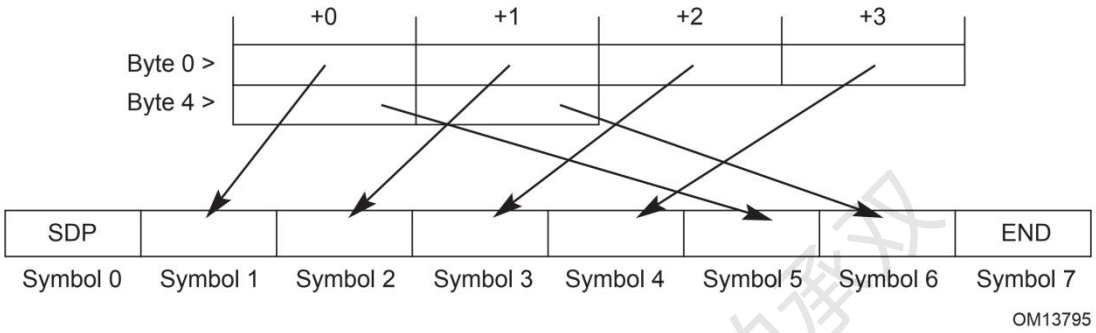


Figure 4-6 DLLP with Framing Symbols Applied

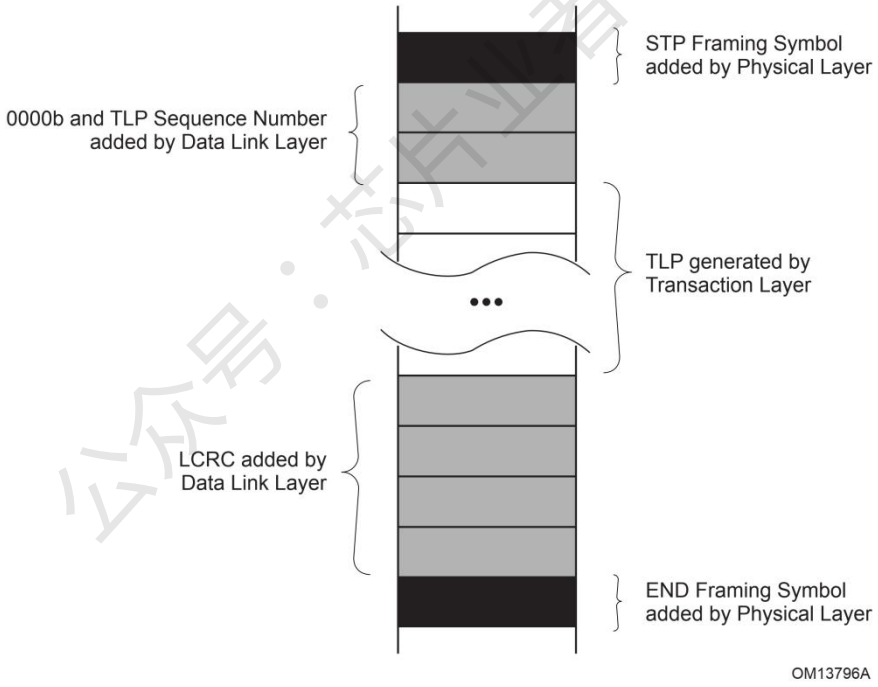


Figure 4-7 Framed TLP on a x1 Link

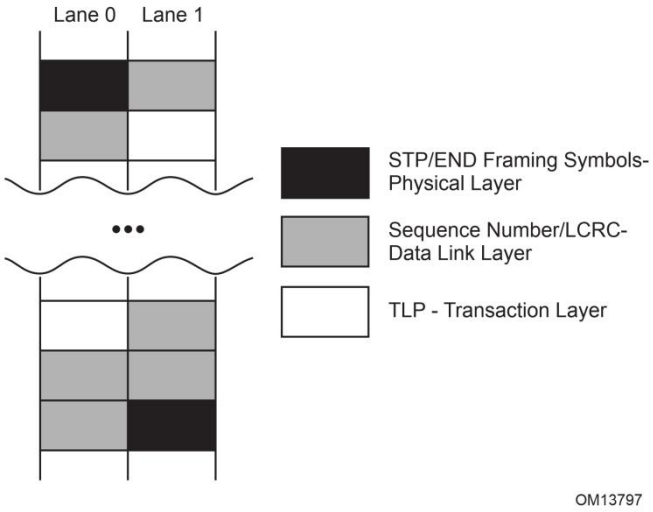


Figure 4-8 Framed TLP on a x2 Link

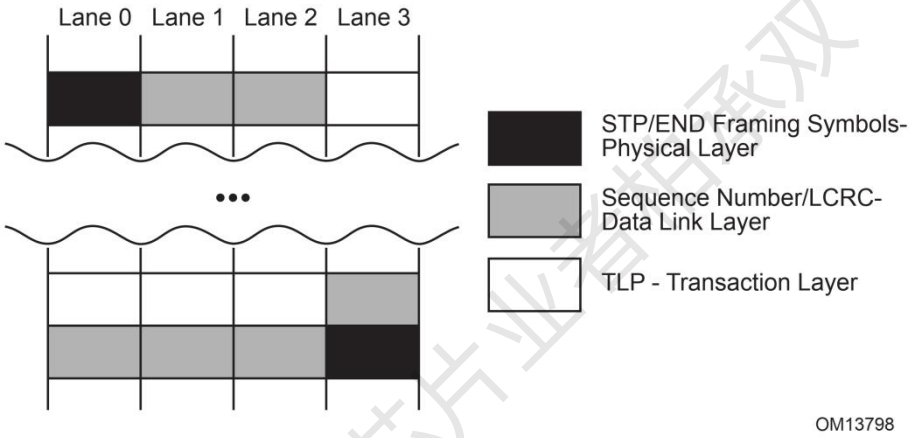


Figure 4-9 Framed TLP on a x4 Link

公众号：芯片业者相承双

4.2.1.3 Data Scrambling

数据加扰在 8b/10b 编码之前进行。Scrambler 的主要作用就是通过"加扰"的方法消减 EMI 噪声，所谓加扰是将源数据流与一个随机序列异或后，再发送出去。此时被发出的数据流也基本是伪随机的，从而降低了发送数据时产生的 EMI 噪声。

PCIe 定义的加扰功能可以通过多 Lane 链路上的一个或多个线性反馈移位寄存器（Linear Feedback Shift register，LFSR）来实现。 如果每个链路发送端实现了多个 LFSR，则这些 LFSR 必须协同工作，即在同一时刻每个 LFSR 生成的值相同（请参阅 Table 4-18，Lane-to-Lane Output Skew）。 如果接收器中实现了多个 LFSR，则它们也必须协同工作，即在同一时刻每个 LFSR 生成的值相同（请参阅 Table 4-24，Lane-to-Lane Skew）。 无论如何实现，LFSR 都必须在逐个 Lane 的基础上与数据进行交互，就像在此链路中每个 Lane 所描述的那样，都有一个单独的 LFSR。在发送侧，加扰应用于 8b / 10b 编码之前的字符。在接收端，对 8b / 10b 解码后的字符进行解扰。

LFSR 如 Figure 4-10 所示。通过将 8 位（D0-D7）字符与 LFSR 的 16 位（D0-D15）输出串行异或来执行加扰或解扰。LFSR 的输出 D15 与要处理的数据的 D0 进行异或。然后，对 LFSR 和数据寄存器进行串行高级处理，并对 D1 至 D7 重复输出处理。 数据进行异或后，LFSR 将前进。LFSR 实现多项式为：

$$G(X)=X16+X5+X4+X3+1$$

数据链路层用来通知物理层禁用加扰的机制和接口是特定于实现的，不在本规范的范围之内。

加扰规则如下：

- COM 字符来指示初始化 LFSR。
- 除 SKP 外，每个符号的 LFSR 值都提前了八个串行移位。
- 训练序列有序集（Training Sequence Ordered，如 TS1 和 TS2）、一致性 Pattern（参阅 Section 4.2.8）和修改后的一致性 Pattern（Section 4.2.9）不需要加扰，其余所有数据符号（D codes）都要加扰。
- 所有的控制符号（K codes）都不需要加扰。
- LFSR 的种子（D0-D15）的初始值为 FFFFh。COM 发出后，立即初始化发送侧的 LFSR。每当 COM 在该链路的任何 Lane 上进入接收 LFSR 时，都会初始化接收端的 LFSR。
- 只能在配置结束时禁用加扰（请参阅 Section 4.2.6.3.5）。
- 加扰不用于 loopback。
- 在 Detect 状态下，加扰默认开启。

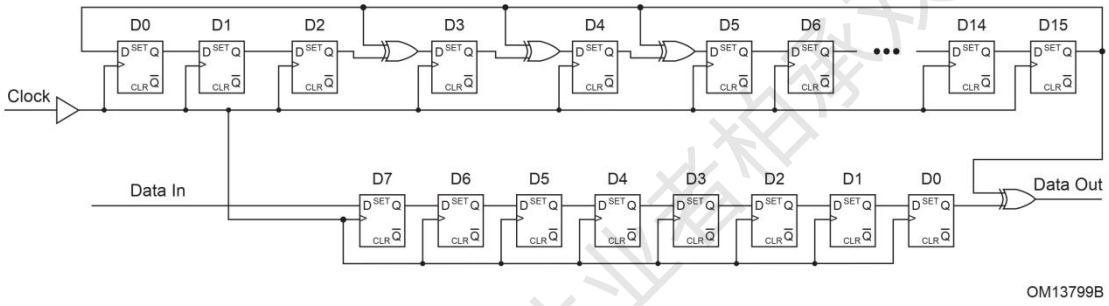


Figure 4-10 LFSR with 8b/10b Scrambling Polynomial

IMPLEMENTATION NOTE

Disabling Scrambling

禁用加扰旨在帮助简化测试和调试设备。精确 data pattern 的控制在测试和调试环境中非常有用。由于在物理层复位了加扰，因此没有合理的方法通过软件可靠地控制数据转换的状态。因此，为了这些目的而在 TS1 和 TS2 有序集中添加 Disable Scrambling 字段。

数据链路层用来通知物理层禁用加扰的机制和/或接口是特定于实现的，超出了本规范的范围。

关于加扰的更多信息，参阅 Appendix C，Appendix C 给出了 8b/10b 加扰的 C 代码实现实例和 128b/130b 加扰的 C 代码实例。

4.2.2 Encoding for 8.0 GT/s and Higher Data Rates

当 PCI Express 链路以 8.0 GT/s 或更高的数据速率运行时，它将使用本小节中描述的编码规则：128b/130b 编码。 为了向后兼容，链路最初按照 Section 4.2.1 节中所述使用 8b/10b 编码以 2.5 GT/s 的数据速率训练到 L0，然后将数据速率更改为 8.0 GT/s 或更高时，使用 128b / 130b 编码。128b/130b 编码是一种链路范围的打包机制，并且是具有加扰功能的每 Lane 的 block 编码机制。数据传输的基本实体是 8 位数据字符，称为符号，如 Figure 4-11 和 Figure 4-12 所示。

IMPLEMENTATION NOTE

Symbol in 128b/130b Encoding Scheme

在 128b/130b 编码方案中，Symbol 定义为一个字节长度，类似于 8b/10b 编码的 10-bit Symbol。

4.2.2.1 Lane Level Encoding

物理层使用每 Lane 的 block 编码机制。每个 block 均包含一个 2 位同步头（Sync Header）和一个有效负载。有效的同步头编码有两种：10b 和 01b。同步头定义了该 block 包含的有效载荷的类型。

同步头为 10b 表示数据块（Data block）。每个数据块具有 128 bit 有效载荷，因而块大小为 130 bit。有效负载是 Section 4.2.2.3 中描述的数据流。

同步头为 01b 表示有序集块（Order Set Block）。每个有序集块都有一个 128 bit 有效载荷，因此块大小为 130 bit，但 SKP 有序集可以是可变长度的。

多 Lane 链路的所有 Lane 必须同时发送具有相同同步头的块，但在 Polling.Compliance 中传输 Jitter Measurement Pattern 时除外。

比特发送顺序如下。用“H1H0”表示的 Sync Header 放置在以“H0”开头并以“H1”结尾的 Lane 上。用“S7S6S5S4S3S2S1S0”表示的 Symbol 放置在以“S0”开头并以“ S7”结尾的 Lane 上。在显示时间刻度的图中，比特表示传输顺序。在布局图中，比特以低字节序格式排列，这与本规范其他章节中的数据包布局图一致。

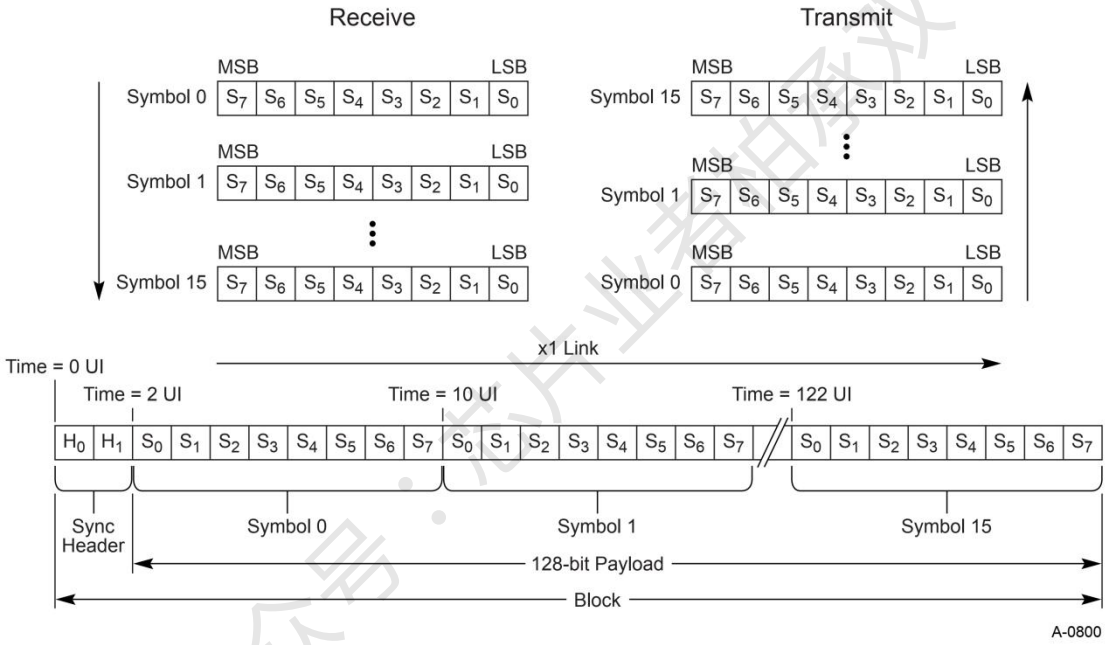


Figure 4-11 Example of Bit Transmission Order in a x1 Link Showing 130 Bits of a Block

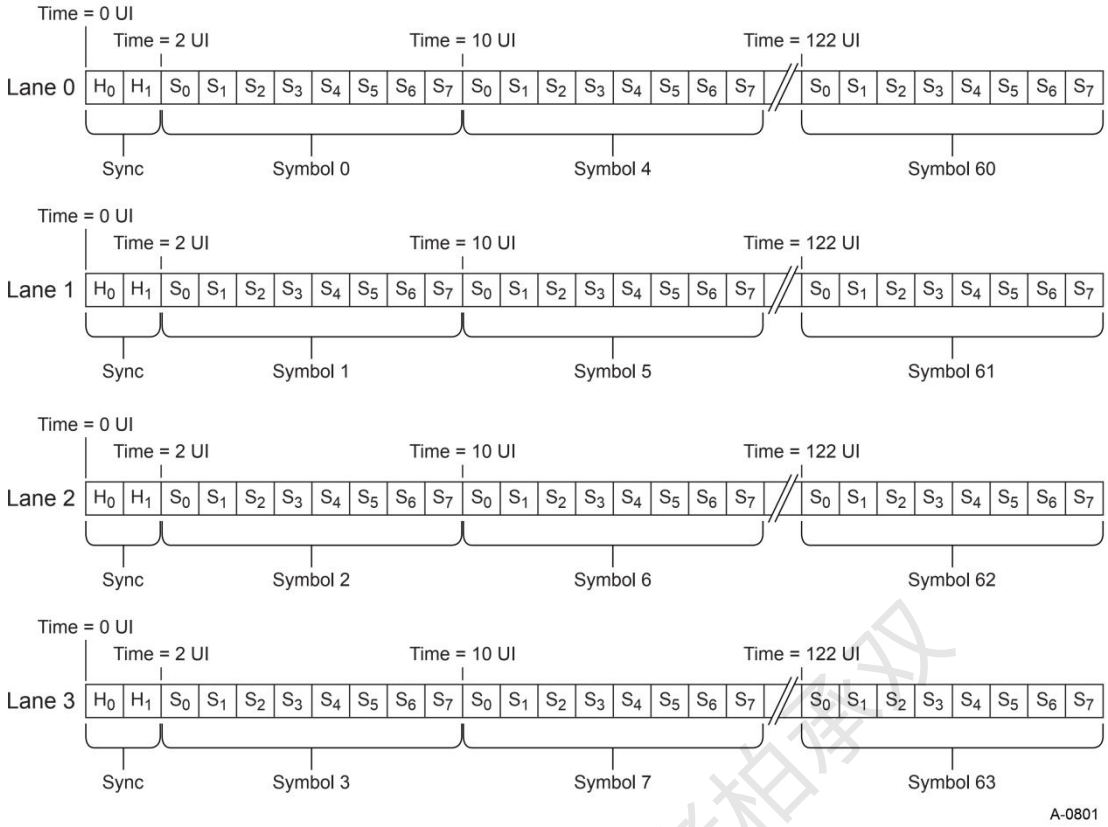


Figure 4-12 Example of Bit Placement in a x4 Link with One Block per Lane

4.2.2.2 Ordered Set Blocks

一个 Ordered Set Block 包含一个同步头，后跟一个有序集。多通道链路的所有 Lane 必须同时传输相同的有序集。有序集的第一个符号定义有序集的类型。有序集在 Section 4.2.4 和 Section 4.2.7 中有详细描述。

4.2.2.2.1 Block Alignment

在链路训练期间，130bit 的 Electrical Idle Exit Ordered Set (EIEOS)是接收器用来确定 Block Sync Headers 的唯一比特流数据。从概念上讲，接收端可以处于 Block Alignment 的三个不同阶段：Unaligned，Aligned 和 Locked 。定义这些阶段是为了说明所需的行为，但并不意味着要这样实现。

Unaligned Phase ：接收器经过一段时间的电气空闲后进入此阶段，例如，当数据速率更改为使用 128b/130b 编码的速率时，或者当它们退出低功耗链路状态时，或者在链路定向时。在此阶段，接收器检测接收到的比特流中的 EIEOS bit pattern。当检测到一个时，他们调整其对齐方式并进入 Aligned Phase。

Aligned Phase ：接收器检测接收到的 EIEOS 比特流和接收到的 Start of Data Stream（SDS）有序集的 Block。如果在与当前对齐方式不匹配的对齐方式上检测到 EIEOS 比特流，则接收方必须将其对齐方式调整为新接收到的 EIEOS 比特流。如果接收到 SDS 有序集，则接收器进入 Locked Phase。如果接收到未定义的同步头（00b 或 11b），则允许接收端返回 Unaligned Phase 。

Locked Phase ：在此阶段，接收方不得调整其块对齐方式。希望在 SDS 有序集之后接收数据块，并且调整块对齐方式会干扰这些块的处理。如果接收到未定义的同步头，则接收器必须返回到 Unaligned Phase 或 Aligned Phase。

IMPLEMENTATION NOTE

Detection of Loss of Block Alignment

在训练序列中发送的 EIEOS 和 TS 有序集的序列将导致未对准的接收器检测到未定义的 Sync Header。

其他要求：

- 在对齐或锁定阶段，接收器必须在收到 SKP 有序集时根据需要调整其对齐方式。有关 SKP 有序集的更多信息，请参见 Section 4.2.7。
- 从任何 LTSSM 状态转换到 Recovery 状态之后，接收器必须忽略所有收到的 TS 有序集，直到收到 EIEOS。从概念上讲，接收 EIEOS 可以验证接收器的对齐方式，并允许进行 TS 有序集处理。 如果收到的 EIEOS 发起了从 L0 到 Recovery 的 LTSSM 转换，则允许接收器处理跟随 EIEOS 的任何 TS 有序集，或者忽略它们，直到收到另一个 EIEOS。
- 只要停止数据流处理，就可以将接收方从 Locked Phase 定向到 Unaligned Phase 或 Aligned Phase。有关数据流要求的更多信息，请参见 Section 4.2.2.3。
- Loopback Masters ： 在 Loopback.Entry 状态下， Master 必须能够调整接收器的块对齐方式以接收 EIEOS 比特流。进入 Loopback.Active 状态时，允许主机发送 EIEOS 并将其接收方的块对齐方式调整为环回比特流。
- Loopback Slaves ： 在进入 Loopback.Entry 时， Slave 必须能够调整接收器的块对齐方式以接收 EIEOS 比特流。Slave 处于 Loopback.Active 时，Slave 不得调整其接收方的块对齐方式。从概念上讲，当从设备开始环回接收到的 bite 流时，将接收器定向到 Locked Phase。

4.2.2.3 Data Blocks

Data Block 的有效载荷是符号流（Symbol Stream），定义为“Data Stream”，它由 Framing Tokens，TLP 和 DLLP 组成。数据流的每个符号都放置在链路的单个 Lane 中，并且符号流在链路的所有 Lane 上分布，并跨越 Block 边界。

数据流从紧随 SDS 有序集的 Data Block 的第一个符号开始。当检测到帧错误时，或者以除 SKP 有序集以外的有序集之前的数据块的最后一个符号时，该数据流结束。 数据流中出现的 SKP 有序集具有特定要求，如以下各节所述。

4.2.2.3.1 Framing Tokens

Table 4-2 中显示了物理层使用的 Framing Tokens。 每个 Framing Tokens 指定或暗示与该 Tokens 相关联的符号的数量，并因此指定下一个 Framing Tokens 的位置。 Figure 4-15 显示了在 x8 链路上传输的 TLP，DLLP 和 IDL 的示例。

数据流的第一个 Framing Token 始终位于数据流的第一个数据块的 Lane 0 的 Symbol 0 中。在本章的其余部分，术语 Framing Token 和 Token 可互换使用。

Table 4-2 Framing Token Encoding

Framing Token Type	Description
IDL	Logical Idle. The Framing Token is 1 Symbol. This Token is transmitted when no TLPs or DLLPs or other Framing Tokens are being transmitted.
SDP	Start of DLLP. The Framing Token is 2 Symbols long and is followed by the DLLP information.
STP	Start of TLP. The Framing Token is 4 Symbols long and includes the 12-bit TLP Sequence Number. It is followed by the TLP information.
EDB	EnD Bad. The Framing Token is 4 Symbols long and is used to confirm that the previous TLP was nullified.
EDS	End of Data Stream. The Framing Token is four Symbols long and indicates that the next Block will be an Ordered Set Block.

公众号：芯片业者相承双

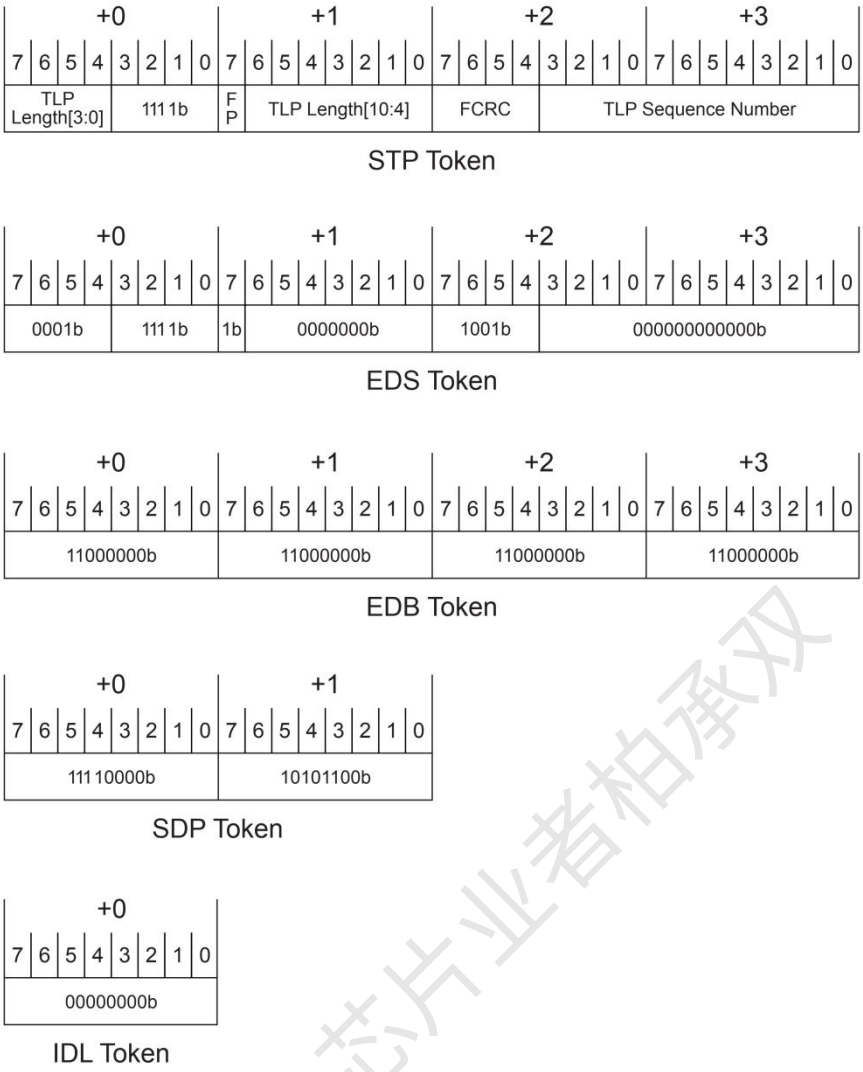


Figure 4-13 Layout of Framing Tokens

物理层 DLLP 布局如 Figure 4-14 所示。Symbol 0 和 1 是 SDP Token，Symbol 2 至 7 是数据链路层 DLLP 信息。

物理层的 TLP 布局如 Figure 4-14 所示。详细的信息要结合 Figure 4-13 一起理解。TLP length 的单位是 DW，包括 Framing Tokens、TLP Prefixes（如果有）、TLP Header、TLP data payload（如果有）、TLP digest（如果有）以及 TLP LCRC 等信息。例如，如果 TLP 具有一个 3DW 头，1 个 DW 数据有效载荷，并且不包含 TLP 摘要，则 TLP length 字段值为 6：它包括 1（Framing Token）+ 0（TLP 前缀）+ 3（TLP 头） ）+ 1（TLP 数据有效负载）+ 0（TLP 摘要）+ 1（TLP LCRC）。如果同一 TLP 包含 TLP 摘要，则 TLP 长度字段值将为 7。当 TLP 无效时，EDB 令牌被视为 TLP 的扩展，但不包括在 TLP 长度字段的计算中。

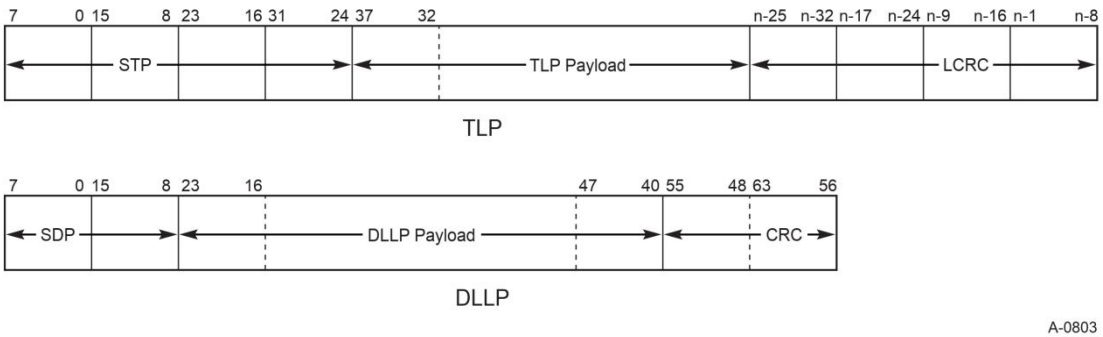


Figure 4-14 TLP and DLLP Layout

公众号：芯片业者相承双

TLP length 字段受 4 位 CRC（Frame CRC，即上述 FCRC 字段）保护，偶校验位（Frame Parity，即上述 FP 字段）保护 TLP 长度和 FCRC 字段。FCRC 和 FP 计算如下：

C[0] = L[10] ^ L[7] ^ L[6] ^ L[4] ^ L[2] ^ L[1] ^ L[0]

C[1] = L[10] ^ L[9] ^ L[7] ^ L[5] ^ L[4] ^ L[3] ^ L[2]

C[2] = L[9] ^ L[8] ^ L[6] ^ L[4] ^ L[3] ^ L[2] ^ L[1]

C[3] = L[8] ^ L[7] ^ L[5] ^ L[3] ^ L[2] ^ L[1] ^ L[0]

P = L[10] ^ L[9] ^ L[8] ^ L[7] ^ L[6] ^ L[5] ^ L[4] ^ L[3] ^ L[2] ^ L[1] ^ L[0] ^ C[3] ^ C[2] ^ C[1] ^C[0]

FP 字段的值简化为 P = L[10] ^ L[9] ^ L[8] ^ L[6] ^ L[5] ^ L[2] ^ L[0] 。

在上述等式中，TLP 长度字段表示为 L [10:0]，其中 L [0]是最低有效位，L [10]是最高有效位。发送器在发送之前计算 Frame CRC 和 Frame Parity。接收器必须使用与发送器相同的算法来计算 Frame CRC 和 Frame Parity，然后将计算出的值与接收到的值进行比较。

STP Token 的 TLP length 字段值不为 1。如果接收到的符号序列与 TLP length 字段值为 1 的 STP Token 格式匹配，则将评估这些符号以确定它们是否与 EDS Token 匹配。

IMPLEMENTATION NOTE

Frame CRC and Frame Parity

Frame CRC 有效地计算为（L [0] X^{14} + L [1] X^{13} +…+ L [9] X^5 + L [10] X^4）mod（X^4 + X +1）。应该注意的是，X^4 + X + 1 是一个原始多项式，CRC 可以检测到两个误码。 Frame Parity 字段可以检测到奇数个误码。因此，Frame CRC 和 Frame Parity 共同保证了 TLP length 字段的三比特错误检测。必须注意的是，即使在简化的帧奇偶方程中，并非所有项都存在，它仍然保持了检测奇数位错误的特性。计算中仅使用偶数个 CRC 项中存在的那些 TLP 长度字段位。

请注意，对于 TLP，在计算 LCRC 之前，数据链路层会在 TLP 序列号字段前添加 4 个保留位（0000b）。当使用 128b/130b 编码时，这些保留位未明确发送，并且接收器在计算 LCRC 时假定接收到的 4 位为 0000b。

TLP length 字段大于 1535 的数据包是 PMUX 数据包。 对于此类数据包，实际数据包长度的计算方法有所不同，STP Token 中的 TLP 序列号字段包含其他信息，并且使用不同的规则计算链路 CRC。有关详细信息，请参见 Appendix G。

TLP length 字段介于 1152 和 1535（含）之间的的数据包保留以供将来标准化。

发射机必须发送由 STP Frame Token 的 TLP length 字段指定个数的 DW。使用 128b/130b 编码时，即使 TLP 无效，也不会将它截断。Figure 4-16 显示了无效的 23 DW TLP 的示例。

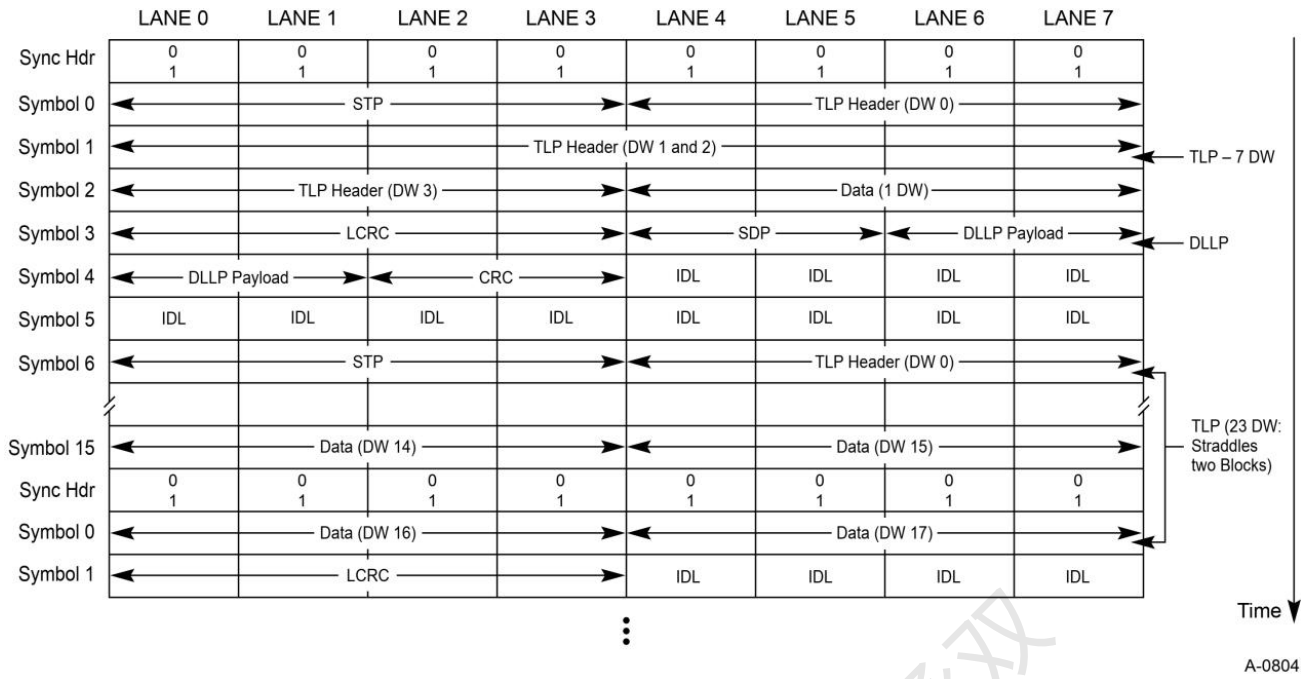


Figure 4-15 Packet Transmission in a x8 Link

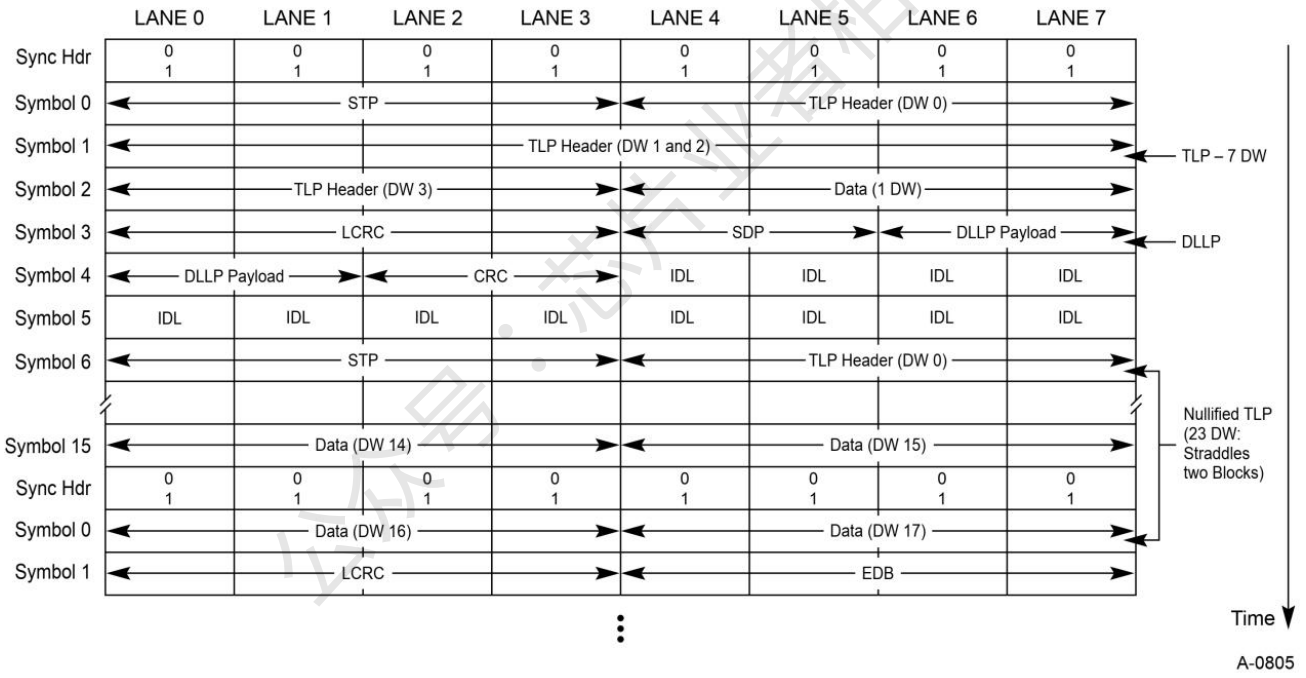


Figure 4-16 Nullified TLP Layout in a x8 Link with Other Packets

Figure 4-17 显示了 TLP，DLLP，IDL 和 EDS 令牌以及后跟 SKP 有序集的示例。SKP 有序集在 Section 4.2.7.2 中定义。

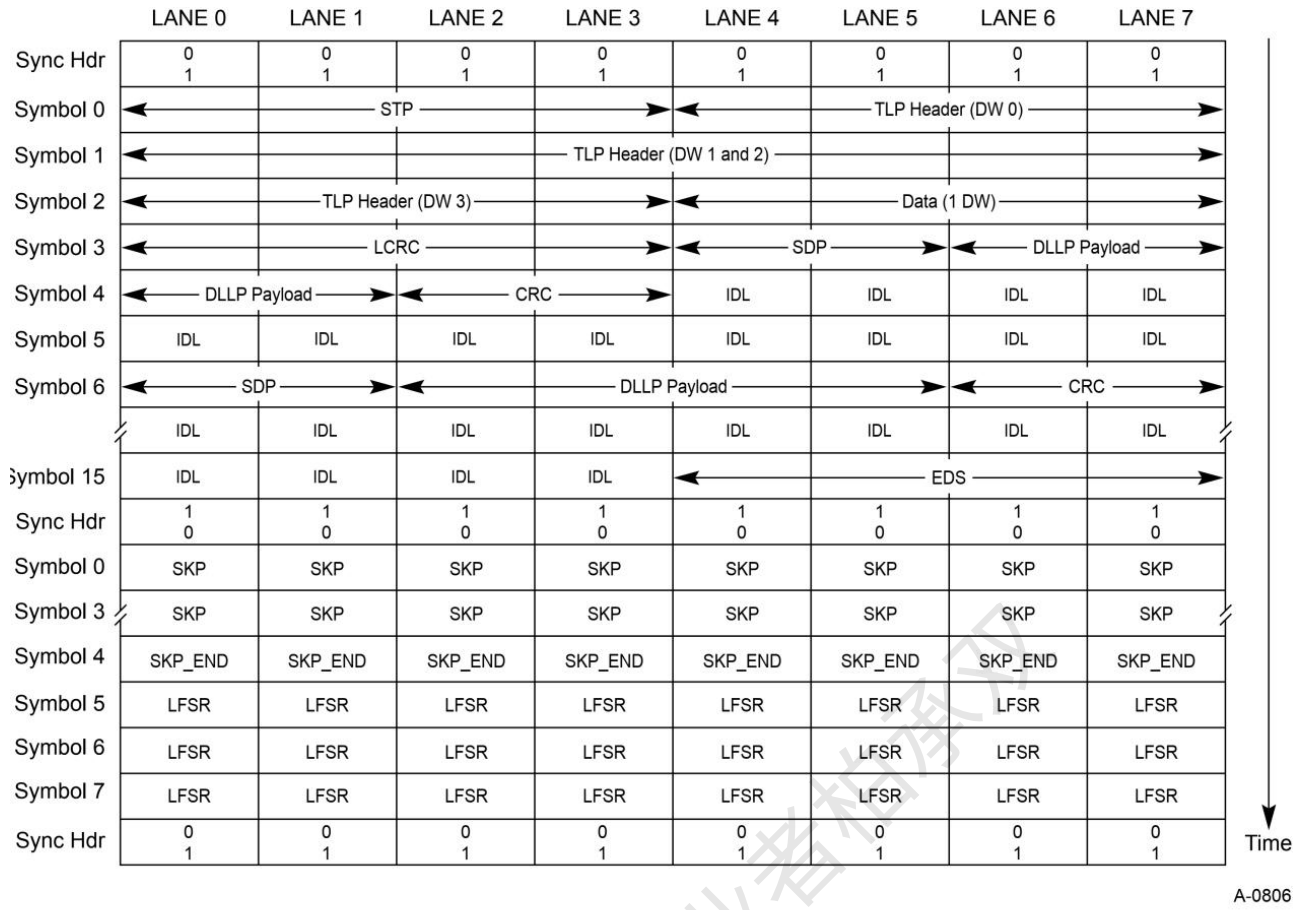


Figure 4-17 SKP Ordered Set of Length 66-bit in a x8 Link

4.2.2.3.2 Transmitter Framing Requirements

发送端的 Data Stream 需要遵循以下规则。

- 为了发送一个 TLP：
  - 发送 STP Token，然后立即发送数据链路层提供的完整 TLP 信息。
  - 即使 TLP 无效，也必须发送 STP Token 中 TLP length 字段所指定的 TLP 的所有 DW。
  - 如果 TLP 为空，则必须在 TLP 之后立即发送 EDB Token。TLP 的最后一个符号和 EDB Token 的第一个符号之间不得有别的符号。考虑到 EDB Token 的原理，一个无效的 TLP 的 STP Token 的 TLP length 字段的值不能进行调整。
  - 每个 Symbol Time 发送 STP Token 的频率不得超过一次。
- 为了发送一个 DLLP：
  - 发送 SDP 令牌，然后立即发送数据链路层提供的完整 DLLP 信息。
  - 必须传输 DLLP 的所有的 6 个符号。
  - 每个 Symbol Time 发送 SDP 令牌的频率不得超过一次。
- 为了在 Data Stream 中发送一个 SKP 有序集：
  - 在当前数据块的最后一个 DW 中发送 EDS Token。例如，对于 x1 链路，Token 在块的 Symbol Time 为 12-15 的 Lane 0 上传输，对于 x16 链路在块的 Symbol Time 为 15 的 Lane 12-15 上传输。

公众号·芯片业者检索

- 在当前数据块之后发送 SKP 有序集。
- 在 SKP 有序集之后发送数据块。数据流从数据块的第一个符号恢复。如果计划发送多个 SKP 有序集，则每个 SKP 有序集必须在其后带有一个 EDS Token 的数据块。

- 如果要结束一个 Data Stream：
  - 在当前数据块的最后一个 DW 中发送 EDS Token，然后在下一个块中发送 EIOS 或 EIEOS。对于 LTSSM 电源管理状态转换，需要发送 EIOS；对于所有其他情况需要发送 EIEOS。例如，对于 x1 链路，Token 在块的 Symbol Time 为 12-15 的 Lane 0 上传输，对于 x16 链路，在块的 Symbol Time 为 15 的 Lane 12-15 上传输。
- 不传输 TLP，DLLP 或其他 Frame Tokens 时，必须在所有 Lane 上传输 IDL Token。
- 对于多 Lane 的链路：
  - 传输一个 IDL Token 后，必须在之后的 Symbol Time 的 Lane 0 中传输下一个 STP 或 SDP Token 的第一个符号。可以在相同的 Symbol Time 内在 IDL Token 之后发送 EDS Token，因为它必须在块的最后 DW 中发送。
  - 对于 N 为 8 或更大的 xN 链路，如果 EDB Token、TLP 或 DLLP 结束于 Lane K，其中 K 不等于 N-1，并且其后没有 STP，SDP 或 EDB Token 位于 Lane K+1 中，那么 IDL Token 必须放在 Lane K+1 到 N-1 中。例如，在 x8 链路上，如果 TLP 或 DLLP 在 Lane 3 中结束，则必须将 IDL Token 放在 Lane 4 至 Lane 7 中。EDS Token 是此要求的例外，可以在 IDL Token 之后传输。
  - Tokens、TLP 和 DLLP 允许相继出现，以便可以在同一 Symbol Time 内传输一个以上的 Token，只要它们的传输符合本节所述的其他要求即可。
    - 宽度大于 x4 的链路可以将 Token 放置在 Lane 4*N 上，其中 N 是一个正整数。例如，可以将 Token 放置在 x8 链路的 Lane 0 和 Lane 4 中，并且可以将 Token 放置在 x16 链路的 Lane 0、Lane 4、Lane 8 或 Lane 12 中。

4.2.2.3.3 Receiver Framing Requirements

以下要求适用于接收到的数据流以及发生在数据流开始和结束处的块类型切换。

- 当处理预期为 Framing Token 的 Symbol 时，接收与 Framing Token 的定义不匹配的符号或者符号序列会被定义为是 Framing Error。 强烈建议 multi-Lane 链路的接收器在该 Lane 的 Lane Error Status 寄存器中报告一个错误，该错误会在该符号与 STP、IDL、SDP、EDB 或 EDS Token 的 Symbol +0 不匹配的情况下接收到预期的 Framing Token 的第一个符号（仅 bit[3:0]）（请参见 Figure 4-13）。
- 本节中的所有可选错误检查和错误报告都是可选择实现的（请参阅 Section 6.2.3.4）。
- 当收到一个 STP Token 时：
  - 接收器必须计算接收到的 TLP length 字段的 Frame CRC 和 Frame Parity，并将结果与接收到的 Frame CRC 和 Frame Parity 字段进行比较。Frame CRC 或帧 Frame Parity 不匹配被定义为 Framing Error。
    - 为了向数据链路层报告，具有 Framing Error 的 STP Token 不被视为 TLP 的一部分。
  - 如果 TLP length 字段为 1，则这些符号不是 STP Token，此时需要对其进行评估以确定它们是否为 EDS Token。
  - 接收器可选择实现是否检查 TLP length 字段的值是否为 0。如果实现，则接收到 TLP length 字段为 0 则表示 Framing Error。
  - 允许接收者检查 TLP length 字段的值是 2、3 还是 4。如果实现，则接收这样的 TLP length 字段被视为 Framing Error。
  - 接收器可以选择实现是否检查 TLP length 字段的值是否在 1152 和 1535（含）之间。如果实现，则这个 TLP 被视为 Framing

公众号·芯片业者怕承改

Error。

- 不支持协议多路复用的端口上的接收器可以有选择地实现是否检查 TLP length 字段的值是否大于 1535。如果实现，则接收到这样的 TLP 被视为是 Framing Error。
- 支持协议复用的端口上的接收器应处理 TLP length 字段大于 1535 的 STP Token，作为 Appendix G 中定义的 PMUX 数据包的开始。
- 下一个要处理的 Token 以紧随 TLP 的最后一个 DW 之后的符号开始，这由 TLP 的 Length 字段确定。
  - 接收者必须评估此符号，并确定它是否是 EDB Token 的第一个符号，并因此确定 TLP 是否无效。请参阅 EDB Token 要求。
- 接收端可以选择实现是否检查在单个 Symbol Time 内是否收到了多个 STP Token。如果实现这个功能，则在单个 Symbol Time 内接收多个 STP Token 被视为是 Framing Error。

- 当收到一个 EDB Token 时：
  - 如果在 TLP 之后立即收到 EDB Token（指 TLP 的最后一个符号和 EDB Token 的第一个符号之间没有其它符号），则接收器必须通知数据链路层已收到 EDB Token。在处理了 EDB Token 的第一个符号之后或在处理了 EDB Token 的任何或所有其余符号之后，允许接收器通知数据链路层已收到 EDB Token。无论何时将接收到的 EDB Token 通知数据链路层，接收器都必须检查 EDB Token 的所有符号。收到与 EDB Token 的定义不匹配的符号被视为是 Framing Error。
  - 除了紧跟在 TLP 之后这种情况外，在任何其他时间接收 EDB Token 都被视为是 Framing Error。
  - EDB Token 之后的符号是下一个要处理的 Token。
- 当在链路在数据块的后四个符号中收到 EDS 时：
  - 接收器必须停止处理数据流。
  - 在 EDS Token 之后的块中接收除 SKP，EIOS 或 EIEOS 以外的有序集被视为是 Framing Error。
  - 如果在 EDS Token 之后的块中接收到 SKP 有序集，则除非检测到帧错误，否则接收方将使用 SKP 有序集后的数据块的第一个符号恢复数据流处理（这是数据流中嵌入 SKP 有序集的场景）。
- 当收到一个 SDP Token 时：
  - 紧跟在 DLLP 的最后一个符号之后的符号是下一个要处理的 Token。
  - 接收器可以选择检查在单个 Symbol Time 内是否收到了多个 SDP Token。如果实现该功能，则在单个 Symbol Time 内接收多个 SDP Token 将被视为是 Framing Error。
- 当收到 IDL Token 时：
  - 对于 x1 链路，下一个要处理的 Token 是接收到的下一个符号。
  - 对于 x2 链路，下一个要处理的 Token 是在下一个 Symbol Time 的 Lane 0 中接收到的符号。接收器可以选择检查在 IDL Token 之后在 Lane 1 中接收到的符号是否也是 IDL。 如果实现这个功能，则接收 IDL 以外的符号将被视为是 Framing Error。
  - 对于 x4 链路，下一个要处理的 Token 是在下一个 Symbol Time 的 Lane 0 中接收到的符号。接收器可以选择检查 IDL Token 后在 Lane 1-3 中接收到的符号是否也是 IDL。如果实现这个功能，则接收 IDL 以外的符号将被视为是 Framing Error。
  - 对于 x8，x12，x16 和 x32 链路，下一个要处理的 Token 是 IDL Token 后的下一个 DW 对齐 Lane 中接收到的符号。例如，如果在 x16 链路的 Lane 4 中接收到 IDL Token，则下一个 Token 位置是同一 Symbol Time 的 Lane 8。但是，如果在 x8

公众号：芯片业者拓荒

链路的 Lane 4 上接收到 IDL Token，则下一个 Token 位置是下一个 Symbol Time 的 Lane 0。 接收器可以选择检查在 IDL Token 和下一个 Token 位置之间接收到的符号是否也是 IDL。如果实现这个功能，则接收 IDL 以外的符号将被视为是 Framing Error。

- Note：预期在 IDL Token 之后的同一 Symbol Time 内收到的唯一 Token 是其他 IDL Token 或 EDS Token。

- 在处理数据流时，在考虑到 lane-to-lane skew 时，接收器还必须检查每个 Lane 接收到的块类型，其条件如下：
  - 在紧接 SDS 有序集的任何 Lane 上收到有序集块应该被视为 Framing Error。
  - 接收到具有未定义块类型的块（同步头为 11b 或 00b）被视为 Framing Error。接收器可以选择在 Lane Error Status 寄存器中报告此错误。
  - 在任何 Lane 上接收到有序集块而在前一个块中未接收到 EDS Token 的情况被视为 Framing Error。 例如，接收到前面没有 EDS Token 的 SKP 有序集就是 Framing Error。另外，在数据流中立即接收到一个 SKP 有序集，紧接着是另一个有序集块（包括另一个 SKP 有序集），这也是 Framing Error。强烈建议，如果有序集的第一个符号是 SKP，如果接收到的 Symbol 1 到 4N 与 Table 4-22 或 Table 4-23 中的相应符号不匹配，则 multi-Lane 链路的接收器在 Lane Error Status 寄存器中为此 Lane 报告一个错误。
  - 当前一个数据块包含 EDS Token 时，在任何 Lane 上接收数据块都是 Framing Error。
  - 接收器可以选择在不同 Lane 上检查不同的有序集。例如，Lane 0 接收 SKP 有序集，Lane 1 接收 EIOS。如果实现该功能，则接收不同的有序集是一个 Framing Error。

4.2.2.3.4 Recovery from Framing Errors

如果接收器在处理数据流时检测到 Framing Error，则它必须：

- 如 Section 4.2.4.7 所述报告接收器错误。
- 停止处理数据流。如前所述，当接收到下一个 SDS 有序集时，将启动新数据流的处理。
- 按照 Section 4.2.4.7 中所述启动错误恢复过程。如果 LTSSM 状态为 L0，则将 LTSSM 定向到 Recovery。 如果在检测到 Framing Error 时 LTSSM 状态为 Configuration.Complete 或 Configuration.Idle，则可能会由于超时导致从 Configuration.Idle 转移到 Recovery.RcvrLock，或者直接从 L0 转移到 Recovery。如果在检测到 Framing Error 时 LTSSM 状态为 Recovery.RcvrCfg 或 Recovery.Idle，则可能会由于超时导致导致从 Recovery.Idle 转换到 Recovery.RcvrLock，或者直接从 L0 转移到 Recovery。如果 LTSSM 为 Recovery.RcvrLock 或 Configuration.Linkwidth.Start 子状态，则错误恢复过程会依赖这些子状态退出时后的转移状态，即不会直接把 LTSSM 转移到 Recovery 状态；
  - Note：预期 Framing Error 恢复机制不会直接导致任何数据链路层启动的恢复操作，例如 NAK。

IMPLEMENTATION NOTE

Time Spent in Recovery Due to Detection of a Framing Error

使用 128b/130b 编码时，所有 Framing Error 都需要实现链路恢复功能。从所有端口都进入恢复状态开始，从 Framing Error 中恢复所需的时间预计少于 1 us。

4.2.2.4 Scrambling

多 Lane 链路中发射机的每个 Lane 可以实现单独的 LFSR 进行加扰。多 Lane 链路中接收机的每个 Lane 也可以实现单独的 LFSR 用于解扰。当然也可以选择复用 LFSR 的方式。

公众号：芯片业者相承双

8.0GT/s 加扰的多项式为：G(X) = X23 + X21 + X16 + X8 + X5 + X2 + 1，它的实现如 FIgure 4-18 所示。

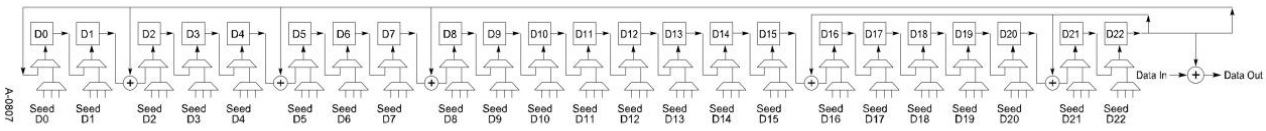


Figure 4-18: LFSR with Scrambling Polynomial in 8.0 GT/s and Above Data Rate

加扰规则如下：

- Sync Header 的两个比特不加扰，也不执行 LFSR（这两个不一样：不加扰意味着还需要进 LFSR 处理，这会占用 cycle，而不执行 LFSR 意味着不占 cycle）。
- 电气空闲退出有序集（EIEOS）的所有 16 个符号都要 bypass 掉加扰。在发送时，发送 EIEOS 的最后一个符号后，将对加扰的 LFSR 进行初始化；在接收时，接收到 EIEOS 的最后一个符号后，将对解扰的 LFSR 进行初始化。
- 对于 TS1 和 TS2 有序集：
  - TS1 和 TS2 有序集的第一个符号要 bypass 掉加扰。
  - TS1 和 TS2 有序集的 1-13 个符号需要加扰。
  - 如果要实现 DC 平衡，TS1 和 TS2 有序集第 14、15 个符号要 bypass 掉加扰；否则要执行加扰。
- FTS（Fast Training Sequence）有序集的 16 个符号都要 bypass 掉加扰。
- SDS（Start of Data Stream）有序集的 16 个符号都要 bypass 掉加扰。
- EIOS（Electrical Idle Ordered Set）有序集的 16 个符号都要 bypass 掉加扰。
- SKP 有序集的所有符号都要 bypass 掉加扰。
- 除 SKP 有序集外，发送器会对所有有序集的所有符号都会推进 LFSR。对于 SKP 有序集的任何符号，都不推进 LFSR。这里推进的意思是让 LFSR 产生下一个伪随机码。
- 接收器解析有序集块的 Symbol 0，以确定是否推进其 LFSR。 如果该块的 Symbol 0 为 SKP（请参见 Section 4.2.7.2），则该块的任何符号都不推进 LFSR。否则，将对块的所有符号推进 LFSR。
- 数据块的所有 16 个符号均要加扰，并使加扰器推进。
- 对于需要加扰的符号，最低有效位首先被加扰，最高有效位最后被加扰。
- LFSR 的种子值取决于当链路首次进入 Configuration.Idle（即从 LinkUp = 0b 的 Detect 进入 Polling 时）时分配给该 Lane 的 Lane number。
  - 根据 Lane number 模 8 获取的种子值如下：
    - 0: 1DBFBCh
    - 1: 0607BBh
    - 2: 1EC760h
    - 3: 18C0DBh
    - 4: 010F12h

公众号·芯片业者相承双

- 5: 19CFC9h
- 6: 0277CEh
- 7: 1BB807h

加扰的伪代码在 Appendix C 的 Section C.2 给出。

- LinkUp = 1 时，LFSR 的种子值不会更改。只要链路是 LinkUp 状态，通过 LTSSM 的 Configuration 状态进行的链路重新配置不会修改初始的 Lane number 分配（即使 Lane 分配在 Configuration 状态可能会更改）。
- 在使用 128b/130b 编码的情况下，Configuration.Complete 状态不会禁用加扰功能。
- Loopback Slave 不允许对环回的比特流进行加扰或解扰。

IMPLEMENTATION NOTE

LFSR Implementation with a Shared LFSR

具体实现时可以选择实现一个 LFSR，并采用不同的分接点，如 Figure 4-19 所示，这等效于具有不同种子的单个 LFSR 每 Lane，如 Figure 4-18 所示。还应注意，四个 Lane 的抽头方程是两个相邻 Lane 的抽头方程的 XOR。 例如，可以通过对 Lane1 和 Lane 7 的输出进行 XOR 运算来获得 Lane 0。 Lane 2 是 Lane 1 和 Lane 3 的 XOR； Lane 4 是 Lane 3 和 Lane 5 的 XOR；Lane 6 是 Lane 5 和 Lane 7 的 XOR。这可以用来帮助减少门数，但由于两个 Lane 的 XOR 结果而可能导致延迟。

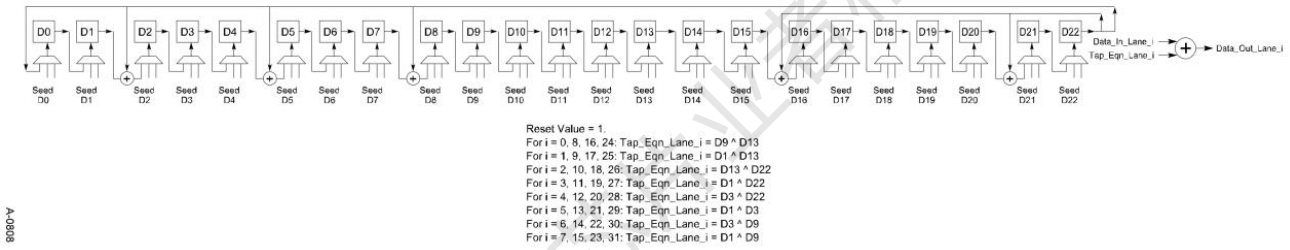


Figure 4-19: Alternate Implementation of the LFSR for Descrambling

4.2.2.5 Precoding

预编码（precoding）是 2018 年全国科学技术名词审定委员会公布的计算机科学技术名词。它是一种利用前后码元相关性，而在很大程度上避免误码扩散的编码技术。

接收器可能会要求其发送器进行预编码，以便以 32.0 GT/s 或更高的数据速率工作。预编码规则如下：

- Port 或 Pseudo-Port 必须在链路的所有已配置通道上请求预编码。如果通过 Port 或 Pseudo-Port 在某些通道上请求预编码，而在其他通道上未请求预编码，则行为是不确定的。
- Port 或 Pseudo-Port 可以请求独立于其他 Port 或 Pseudo-Port 的预编码。例如，在 Figure 4-35 中没有 Retimer 的情况下，或者仅在两个 Retimer 中的 Tx（A）和 Tx（E）的所有通道上，都可能仅在上游端口中打开预编码。示例如 Figure 4-35 所示。
- 当 LTSSM 处于 Detect 状态时，对于数据速率为 32.0 GT/s 以及更高的数据速率，将关闭预编码。
- 在设置 32.0 GT/s 或更高数据速率之前，必须通过设置 EQ TS2 Ordered Set 或 128b/130b EQ TS2 Ordered Set 中的相应比特来发出预编码请求。必须通过在 EQ TS2 或 128b/130b EQ TS2 Ordered Set 中设置 Transmitter Precode Request 来发出预编码请求，然后才能在转换到 Recovery.Speed，从而达到目标数据速率。对于高于 32.0 GT/s 的每个数据速率，必须独立发出预编码请求。

- 如果链路以 32.0 GT/s 或更高的数据速率运行，而没有通过（在已修改的）TS1/TS2 有序集中协商的 No Equalization Needed 机制执行均衡，则必须强制执行来自正在使用的最后均衡结果的预编码请求 。因此，如果每个(pseudo) Port 通告 No Equalization Needed 机制，则必须在每个通道中存储预编码请求以及 Tx Eq 值。如果没有对链路进行过任何均衡（在当前链路之前），则预编码将不会打开。
- 如果目标数据速率为 32.0 GT/s 或更高，则如果在从 Recovery.RcvrCfg 切换到 Recovery.Speed 过程中接收到的八个连续 EQ TS2 或 128b/130b EQ TS2 Ordered Set 中的 Transmitter Precode Request 都设置为 1b，发送器必须为链路在退出 Recovery.Speed 时的目标数据速率开启预编码。一旦开启，预编码将对该目标数据速率有效，直到发送器在从 Recovery.RcvrCfg 切换到 Recovery.Speed 期间接收到另一组八个连续 EQ TS2 或 128b/130b EQ TS2 Ordered Set，且目标速率与之前相同。
- 对于低于 32.0 GT/s 的任何数据速率，发送器不得开启预编码。
- 对于 32.0 GT/s 或更高的数据速率，如果预编码开启，则发送器必须将处于 Recovery 状态的 TS1 Ordered Set 的 Transmitter Precoding On 字段设置为 1；否则，发送器必须将其设置为 0b。
- 启用预编码后，仅对已加扰的比特位进行预编码。
- 用于预编码的“previous bit”在每个块边界上均设置为 1b，并通过当前块边界内发送的最后加扰和预编码比特位进行更新。
- 开启预编码后，对于已加扰的符号，接收器必须先对预编码的比特进行解码，然后再将其发送给解扰器。
- 已在 Lane 0 上以 32.0 GT/s 数据速率开启预编码的发送器，必须将 32.0 GT/s Status 寄存器中的 Transmitter Precoding On 字段设置为 1b；否则必须将其设置为 0b。已请求或将要求以 32.0 GT/s 数据速率打开预编码的接收器，必须在 32.0 GT/s Status 寄存器中将 Transmitter Precoding On 字段设置为 1b； 否则必须将其设置为 0b。

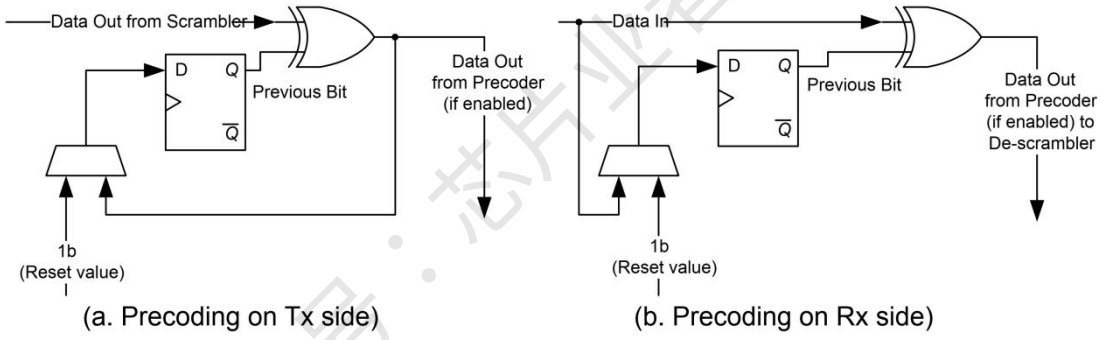


Figure 4-20 Precoding working the scrambler/ de-scrambler

IMPLEMENTATION NOTE

Parity in the SKP Ordered Set when Precoding is turned on

根据 Section 4.2.4.1 和 Section 4.2.7.2 的规则，当启用预编码时，应在发送侧应用预编码之前计算 SKP Ordered Set 中的奇偶校验。因此，发送器中的顺序为：

1. scrambling，
2. followed by parity bit calculation，
3. followed by precoding for the scrambled bits 。

相应的，接收侧的顺序为：

1. precoding (if turned on by the Transmitter)，
2. followed by parity bit calculation，

3. followed by descrambling 。

此顺序的基本原理是，在具有一个或两个 Retimer 的链路中，不同的链路部分可能会启用或禁用预编码。让我们考虑一个在根端口和 Endpoint 之间有一个 Retimer 的示例的系统来说明这一点。在上游方向上，由于 Retimer Receiver 需要在 Endpoint 上的发送器通道中进行预编码，但是由于根端口不需要在其接收器上进行预编码，因此 Retimer 到 Root Port Link 这一短已关闭了预编码。由于 Retimer 不会改变奇偶校验位，因此，如果在预编码后由（Endpoint 的）发送器进行奇偶校验计算，则根端口将出现奇偶校验错误。

IMPLEMENTATION NOTE

Loopback Master behavior if Precoding is on in any Link Segment

根据本节和 Section 4.2.6.4 中提到的预编码规则，以 32.0 GT/s 或更高的数据速率运行的 Loopback Master 应考虑到预编码在某些链路段上处于打开状态，而在其他链路段上处于关闭状态。当 Loopback Slave 从发送 TS1 Ordered Set 切换到环回这些比特时，这一点尤其重要。这是 Loopback Master 接收器上的预编码可以切换的地方（在预编码打开和关闭之间）。允许 Loopback Master 使用特定于实现的机制来处理这种情况。

IMPLEMENTATION NOTE

TS1/TS2 Ordered Sets when Precoding is turned on

根据本节中的规则，当预编码被打开时，用于预编码的“previous bit”对于 Symbol 1 的第一比特是 1b，因为 Symbol 0 没有被加扰并且“previous bit”在块边界处被设置为 1b。

4.2.2.6 Loopback with 128b/130b Code

使用 128b/130b 编码时，Loopback Masters 必须发送具有特定定义的 01b 和 10b 同步头的块。但是，从有序集块过渡到数据块时，不需要传输 SDS 有序集，从数据块过渡到有序集块时，也不需要传输 EDS Token。 Master 必须按照 Section 4.2.7 中的规定定期发送 SKP 有序集，并且它们必须能够处理长度变化的环回的 SKP 有序集。Master 可以发送 Section 4.2.2.2.1 中定义的电气空闲退出有序集（EIEOS）。允许 Master 发送它们期望环回的数据块和有序集块中的任何有效负载。但是，应避免与 SKP OS，EIEOS 或 EIOS 的定义相匹配的有序集块，因为它们在环回时已经有明确的用途。

当使用 128b/130b 编码时，Loopback Slave 必须转发所有接收到的未被修改的比特流，除了 SKP 有序集，可以根据需要对它们进行调整以进行时钟补偿。如果需要时钟补偿，则 Slave 必须为每个有序集添加或删除 4 个 SKP 符号。修改后的 SKP 有序集必须符合 Section 4.2.7.2 的定义（即，它必须具有 4 至 20 个 SKP 符号，后跟 SKP_END 符号，以及 Loopback Masters 发送的紧随其后的三个符号）。如果 Slave 无法获得块对齐或未对齐，则它可能无法执行时钟补偿，因此无法环回所有接收到的比特。在这种情况下，可以根据需要添加或删除符号以继续操作。当检测到从有序集块到数据块的切换时，Slave 不得检查接收到的 SDS 有序集，并且当检测到从数据块到有序集块的切换时，Slave 也不得检查接收到的 EDS Token。

4.2.3 Link Equalization Procedure for 8.0 GT/s and Higher Data Rates

当以 8.0 GT/s 或更高的数据速率运行时，链路均衡过程使组件能够调整每个 Lane 的发送器和接收器设置，以改善信号质量并满足 Chapter 8 中指定的要求。与 LTSSM 关联的所有 Lane 必须参与均衡过程。该过程必须在首次将数据速率更改为 8.0 GT/s 或以上的过程中执行。当 LinkUp = 1b 时，组件必须针对将来将要遇到的所有操作条件到达适当的发送器设置。组件不得要求重复均衡过程来确保可靠运行，尽管规定可以重复该过程。组件必须存储均衡过程中协商好的的发射机设置，并将其用于之后的 8.0 GT/s 或更高的数据速率。甚至在均衡过程完成后，组件仍可以微调其接收器设置，因为这样做不会导致链路不可靠（即不符合 Chapter

公众号·芯片业者相承双

8 中的要求）或进入 Recovery 状态。

任何数据速率都不需要链路均衡过程，如果链路中的所有组件都已通告其 TS1/TS2 Ordered Set 或 modified TS1/TS2 Ordered Set（see Table 4-6 ，Table 4-7 和 Table 4-8）中不需要均衡，则可以完全 Bypass 链路均衡过程（请参阅 Table 4-6，Table 4-7 和 Table 4-8）。如果组件支持 32.0 GT/s 或更高的数据速率，并且可以使用先前的均衡过程中存储的均衡设置可靠地运行，或者不需要均衡来可靠地操作，那么组件可以选择通告它不需要以高于 5.0Gt/s 的任何速率进行均衡。

均衡过程可以自主启动，也可以通过软件启动。强烈建议组件使用自主机制来处理其打算在 5.0 GT/s 以上的所有数据速率。但是，对于所有高于 5.0 GT/s 的数据速率，选择不参与自主机制的组件必须具有其相关软件，以确保在以该数据速率运行之前，基于软件的机制应用于未应用自主机制的高于 5.0 GT/s 的数据速率。

通常，仅当以高于 5.0 GT/s 的所有较低数据速率成功完成均衡时，才以较高数据速率执行均衡。例如，链路将以 8.0 GT/s 成功完成均衡，然后才能以 16.0 GT/s 成功完成均衡，然后才能以 32.0 GT/s 的速率成功完成均衡。但是，如果所有组件都支持 32.0 GT/s 或更高的数据速率，则可以使用一种可选的机制（且链路中的所有组件都支持该机制）来跳过均衡，以达到 32.0 GT/s 或更高的公共的数据速率，就如在 TS1/TS2 Ordered Set 或 modified TS1/TS2 Ordered Set 中通告的那样。启用此可选机制并在组件之间成功协商后，除最高公共数据速率外，不会以任何其他速率执行均衡。对于未执行均衡的所有高于 5.0 GT/s 的数据速率，可以预期的是链路不得以这些数据速率运行。例如，一条链路可以 2.5 GT/s 的速度训练到 L0，进入 Recovery 并以 32.0 GT/s 的速度执行均衡，而跳过 8.0 GT/s 和 16.0 GT/s 的均衡。在这种情况下，链路的预期数据传输速率为 2.5 GT/s，5.0 GT/s 或 32.0 GT/s。如果在最高公共数据速率下的均衡过程即使在重新均衡尝试之后也不成功，并且链路需要以较低的数据速率进行均衡，则下游端口必须停止通告 Equalization bypass to highest rate support 字段，并确保链路以 2.5 GT/s 或 5.0 GT/s 的速率恢复运行。如果不支持跳过均衡到最高公共数据速率的可选机制，则将执行所需的全部均衡过程。如果较低数据速率下的均衡过程由软件驱动，则必须将 Equalization bypass to highest rate Disable 设置为 1，并将 No Equalization Needed Disable 字段设置为 1b；设置目标链路速度，使链路以 2.5Gt/s 或 5.0Gt/s 的速度运行；然后将目标链路速度设置为以 8.0 GT/s 起的较低速率均衡。如果 Equalization bypass to highest rate Disable 设置为 1b，则端口不得通告 Equalization bypass to highest rate support 字段。

如果所有组件都支持 32.0 GT/s 或更高的数据速率，并且链路中的所有组件都支持 No Equalization Needed 机制，则可以使用另一种可选机制跳过整个均衡过程并直接达到 32.0 GT/s 或更高的最高公共数据速率，此速率为 TS1/TS2 或 modified TS1/TS2 Ordered Set 中通告的最高速率。如果一个组件能够从之前的均衡中以所有数据速率检索均衡和其他电路设置（该均衡将对该组件起作用），或者在高于 5.0 GT/s 的所有数据速率中根本不需要均衡，则可以执行此操作。如果 Equalization bypass to highest rate Disable 设置为 1，则组件不得通告此 Capability。

如果链路的一个方向在通告 No Equalization Needed，而另一方向在 TS1/TS2 Ordered Set 中通告 Equalization bypass to highest rate support，则链路将在 Equalization bypass to highest rate support 下工作，因为 No Equalization Needed 字段还指示设备能够绕过均衡以获得最高数据速率。在 modified TS1/TS2 Ordered Set 中，将 No Equalization Needed 字段设置为 1b 的设备还必须将 Equalization bypass to highest rate support 设置为 1b。如果链路的一个方向通告 No Equalization Needed，而另一方向仅在 modified TS1/TS2 Ordered Set 中通告 Equalization bypass to highest rate support，则链路将以 Equalization bypass to highest rate support 工作。 如果设备在其发送的 modified TS1/TS2 Ordered Set 中将 No Equalization Needed 通告为 1b，并且将 Equalization bypass to highest rate support 设置为 0b，则链路操作是不确定的。

如果两个组件在初次链路协商期间（当 LinkUp 设置为 1b 时）通告它们至少能够达到 8.0GT/s 的数据速率（通过 TS1 和 TS2 有序集），并且下游端口选择以高于 5.0GT/s 的预期工作数据速率执行均衡过程，则执行自主机制。虽然不推荐这样做，但下游端口可以选择仅在 5.0 GT/s 以上的预期数据速率的子集上执行自主机制。在这种情况下，必须执行基于软件的机制，以便对 5.0 GT/s 以上的预期数据速率执行均衡过程，而该机制不在自主机制的范围内。例如，如果两个组件都通告了 8.0 GT/s、16.0 GT/s 和 32.0 GT/s 数据速率，但仅对 8.0 GT/s 和 16.0 GT/s 数据速率执行了自主均衡，则必须采用基于软件的机制以 32.0 GT/s 数据速率进行均衡。

在自主机制中，在进入 L0 之后，无论当前链路速度如何，如果必须执行均衡过程并且直到均衡过程完成，则任何组件都不必发送任何 DLLP。一旦每个通道的发送器和接收器设置已针对每个支持的高于 5.0 GT/s 的公共数据速率进行了调整，下游端口打算使用自主机制执行均衡，则认为均衡过程已完成。下游端口需要启动对需要执行均衡的数据速率的速度改变。在任何均衡（自

主或软件启动或重新均衡）期间，下游端口不得通告对高于 Recovery 中需要执行均衡的数据速率的任何数据速率的支持。提供以下示例来说明均衡流程。

Example：考虑一个需要在 8.0 GT/s 和 16.0 GT/s 下自动执行均衡的链路。下游端口通过不通告高于 8.0 GT/s 的任何数据速率而进入 Recovery 以 8.0 GT/s 的速率执行均衡。如果 Link Status 2 寄存器的 Equalization 8.0 GT/s Phase 3 Successful 和 Equalization 8.0 GT/s Complete 字段均设置为 1b，则认为 8.0 GT/s 均衡过程已成功执行。 从 Recovery 切换到 L0 后，在初始数据速率更改为 8.0 GT/s 之后，需要下游端口从 L0 切换到 Recovery，通告对 16.0 GT/s 数据速率支持（但不通告对 32.0 GT/s 的支持，即使它能够支持 32.0 GT/s），将数据速率更改为 16.0 GT/s，然后执行 16.0 GT/s 均衡过程。

如果下游端口检测到均衡问题或上游端口发出均衡重做请求（通过将 Request Equalization 设置为 1b），则下游端口可以在继续以均衡失败的数据速率操作或以更高的数据速率执行均衡之前重做均衡。在给定的数据速率下，背靠背均衡重做的次数是特定于实现的，但必须是有限的。如果在初始或后续均衡过程结束时以及在执行特定数量的均衡重做时，链路不能以执行均衡的数据速率可靠地操作，那么它必须恢复到较低的数据速率操作。

在均衡过程完成之前，使用自主机制的组件不得启用任何自主链路宽度缩小机制。在从下游端口接收到 DLLP 之前，上游端口不得传输任何 DLLP。如果下游端口再次执行均衡，则在完成均衡过程之前，它不得传输任何 DLLP。如果下游端口能够满足其系统要求，则可以基于其自身的需要或者基于来自上游端口的请求再次执行均衡。如 Section 6.6 所述，多次执行均衡可能会干扰软件对链路和设备状态的确定。

IMPLEMENTATION NOTE

DLLP Blocking During Autonomous Equalization

当使用自主机制以 8.0 GT/s 或更高的数据速率进行均衡时，下游端口需要阻止 DLLP 的传输，直到均衡完成，并且上游端口需要阻止 DLLP 的传输，直到从下游端口接收到 DLLP 为止。如果两个组件都通告它们能够以 16.0 GT/s（或 32.0 GT/s）的数据速率传输数据，但下游端口仅使用自主机制以 8.0 GT/s 的速率进行均衡，则下游端口仅需要在 8.0 GT/s 均衡完成之前阻止 DLLP 传输。类似地，如果两个组件都通告它们能够以 32.0 GT/s 的数据速率传输，但下游端口仅使用自主机制以 16.0 GT/s 的速率进行均衡，则下游端口仅需要阻塞 DLLP 传输，直到 16.0 GT/s 均衡完成。如果在 DLLP 传输被阻止时，下游端口延迟从 L0 进入 Recovery，则 L0 推断的电气空闲超时（见 Section 4.2.4.4）或 DLLP 超时（见 Section 2.6.1.2）可能在上游或下游端口中过期。如果这两个超时中的任何一个发生，它将导致启动一个进入 Recovery 来执行链路重新训练。这两个超时都不是可报告的错误情况，并且由此产生的链路重新训练对正确的链路操作没有影响。

当使用基于软件的机制时，软件必须保证不会对正在进行的传输事务产生任何副作用（例如，不能超时）如果有的话，是因为链路正在执行均衡过程。软件可以将 1b 写入 Link Control 3 的 Perform Equalization 字段，然后写 Link Control 2 的 Target Link Speed 字段，以使链路以 8.0 GT/s 或更高的速度运行，然后将 1b 写到下行端口的 Link Control 的 Retrain Link 字段以执行均衡过程。如果未成功执行以 8.0 GT/s 开始的所有较低数据速率的均衡过程，则在软件启动的均衡过程中，软件不得使链路以 8.0 GT/s 以上的数据速率运行并将均衡 bypass 到更高的数据速率（即 Equalization bypass to highest rate Supported 为 0b 或 Equalization bypass to highest rate Disable 为 1b）。对于以下数据速率，均衡过程被视为成功：

8.0 GT/s：Link Status 2 寄存器的 Equalization 8.0 GT/s Phase 3 Successful 和 Equalization 8.0 GT/s Complete 字段均设置为 1b。

16.0GT/s：16.0GT/s Status 寄存器的 Equalization 16.0 GT/s Phase 3 Successful 和 Equalization 16.0 GT/s Complete 字段均设置为 1b。

软件可能会在两个组件中都设置 Link Control 寄存器的 Hardware Autonomous Width Disable 字段，或使用其他机制来确保在设置下行端口中的 Perform Equalization 字段之前，链路的全部 Lane 的功能是完善的。启用自主 width downsizing 功能的组件负责通过把 Hardware Autonomous Width Disable 设置为 1b 之后的 1 ms 内让 LTSSM 切换到 Recovery 和 Configuration，来将链路配置为对应宽度。如果上游端口最初并未通告 8.0 GT/s、16.0GT/s 或 32.0GT/s 数据速率并参与自主均衡机制，则其相关软件必须确保对均衡期间执行的任何传输事务没有任何副作用（如果有的话），然后再指示上游端口转换到 Recovery 状态并通告之前未通告的数据速率并启动一个速度切换。下游端口在转换到 Recovery 状态时，在初始速度更改为通告的数据速率时，随后将启动均衡过程。

Hardware Autonomous Width Disable = 0，表示在链路某条 Lane 异常时链路会自主协商，减小 Lane 的宽度，以规避异常 Lane。当切换到 8.0 GT/s 时，可能存在某条 Lane 设置的不恰当导致该 Lane 支持不了 8.0 GT/s 而其他 Lane 可以支持 8.0 GT/s 的现象，这时候这个功能就起作用了。Hardware Autonomous Width Disable = 1 的话这个功能就被禁用了。

需要上游端口在 Recovery.RcvrLock 状态下检查均衡设置问题（参阅 Section 4.2.6.4.1）。但是，允许下游端口和上游端口在任何时候都使用特定于实现的方法来检测均衡问题。对于以下每个数据速率，如果检测到其均衡设置有问题则端口需要执行以下操作：

8.0GT/s ： Link Status 2 寄存器的 Link Equalization Request 8.0 GT/s 字段要设置为 1b 。

16.0GT/s ： 16.0 GT/s Status 寄存器的 Link Equalization Request 16.0 GT/s 字段要设置为 1b 。

32.0GT/s ： 32.0 GT/s Status 寄存器的 Link Equalization Request 32.0 GT/s 字段要设置为 1b 。

除了把相应的 Link Equalization Request 设置为 1b，上游端口必须让 LTSSM 切换到 Recovery 状态，并使用其在 Recovery.RcvrCfg 子状态中发送的 TS2 有序集中的 Request Equalization 位来发出均衡请求，并把 Equalization Request Data Rate 设置为检测到问题的数据速率。对于每个检测到的问题，它只能发出一次均衡请求。当请求均衡时，也允许但不要求上游端口将 Quiesce Guarantee 字段设置为 1b，以通知下游端口在 1ms 内启动的均衡过程不会对其操作造成任何副作用。

当下游端口接收到来自上游端口的均衡请求时(处于 Recovery.RcvrCfg 状态并接收到 8 个连续的 TS2 有序集且 Request Equalization 设置为 1b 时），它必须在完成下一个 Recovery 到 L0 转换的 1ms 内以请求的数据速率（由接收的 Equalization Request Data Rate 字段定义）启动均衡过程，或必须在其 Link Status 2 寄存器中设置适当的 Link Equalization Request 8.0 GT/s 值，或在其 16.0 GT/s Status 寄存器中设置 Link Equalization Request 16.0 GT/s，或在其 32.0 GT/s Status 寄存器中设置 Link Equalization Request 32.0 GT/s。只有当它能够保证执行均衡过程不会对其操作或上游端口的操作造成任何副作用时，它才应该启动均衡过程。允许但不要求下游端口使用接收到的 Quiesce Guarantee 位来确定上游端口在没有副作用的情况下执行均衡过程的能力。

如果下游端口要启动均衡过程并可以保证不会对其自身操作产生副作用，但是无法直接确定均衡过程是否会对上游端口的操作产生副作用，则允许请求上游端口发起均衡请求。下游端口通过转换为 Recovery 并在 Recovery.RcvrCfg 状态下将其传输的 TS2 有序集的 Request Equalization 位设置为 1b，将 Equalization Request Data Rate 位设置为所需数据速率，并将 Quiesce Guarantee 位设置为 1b 来实现此目的。当上游端口从下游端口接收到这样的均衡请求时（处于 Recovery.RcvrCfg 状态，并且接收到 8 个连续的 TS2 有序集，且 Request Equalization 和 Quiesce Guarantee 位设置为 1b），这是允许的，但不是必需的，以静默其操作并准备以下游端口请求的数据速率执行均衡过程，然后以相同的数据速率（使用前面介绍的报告均衡设置问题的方法）并要求将 Quiesce Guarantee 位设置为 1b。上游端口可以响应的时间没有时间限制，但是应该尝试尽快做出响应。如果下游端口发出请求并从上游端口接收到此类响应，则它仍必须保证在执行完下一次从 Recovery 到 L0 的转换后的 1 毫秒内，以协商的数据速率启动均衡过程。均衡过程不会对其操作产生任何副作用，或者必须在其 Link Status 2 寄存器中设置适当的 Link Equalization Request 8.0 GT/s 或在 16.0 GT/s Status 寄存器中设置 Link Equalization Request 16.0 GT/s 位或在 32.0 GT/s Status 寄存器中设置 Link Equalization Request 32.0 GT/s 位。

IMPLEMENTATION NOTE

Using Quiesce Guarantee Mechanism

数据链路层处于 DL_Active 状态后，由于执行均衡而产生的副作用可能发生在端口，设备或系统级别。例如，执行均衡过程所需的时间可能会导致 Completion Timeout 错误发生-这指的是在其他可能的系统组件中。Quiesce Guarantee Mechanism 可以帮助端口决定是否发出一个均衡请求动作。

在报告其均衡问题之后，组件可以通过 Recovery.Speed 超时或通过将数据速率更改为较低的数据速率来以较低的数据速率运行。执行均衡过程所需的任何数据速率更改均不受 Section 6.11 节中 200 ms 要求的限制。Table 4-3 描述了重做均衡的机制。有时可能需要将速度更改为中间数据速率以重做均衡。例如，如果下游端口要以 16.0 GT/s 的速度重新进行均衡，且不支持 bypass equalization，并且当前数据速率为 2.5 GT/s 或 5.0 GT/s，则下游端口必须首先将速度更改为 8.0 GT/s（除非必要，否则不会执行 8.0 GT/s 均衡过程），然后它将以 16.0 GT/s 的速度启动重做均衡。通过 Recovery 状态，每一跳最多只能执行一次均衡过程。

Table 4-3 Equalization requirements under different conditions

从 2.5GT/s 或 5.0GT/s 到 8.0GT/s 均衡	此处描述的机制对于所有均衡方式都是相同的：初次、重做、自主或软件启动。 下游端口使用 8b/10b 编码将每个通道的发送器的 preset 值和接收器的 preset hint 发送到上游端口（如果适用）。在协商了将数据速率更改为较高数据速率之前，在切换到较高数据速率以执行均衡之前，使用 Recovery.RcvrCfg 中的 EQ TS2 有序集（在 Section 4.2.4.1 中定义）发送这些值。在 EQ TS2 有序集中发送的 preset 的推导如下： ● 对于 8.0 GT/s 速率下的均衡：Lane Equalization Control Register Entry 的 Upstream Port 8.0 GT/s Transmitter Preset 和 Upstream Port 8.0 GT/s Receiver Preset Hint 字段的值。 在数据速率更改为需要进行均衡的较高数据速率之后，上游端口将使用其接收到的 preset 值发送 TS1 有序集。如果发送器使用 reduced swing，则 preset 值必须在 Section 8.3.3.3 中定义的可操作范围内。 在数据速率改变到需要执行均衡的更高数据速率后，下游端口发送具有如下 preset 值的 TS1 有序集，假设 preset 值必须在 Section 8.3.3.3 中定义的可操作范围内，前提是发送器将使用 reduced swing： ● 对于 8.0 GT/s 速率下的均衡：Lane Equalization Control Register Entry 的 Downstream Port 8.0 GT/s Transmitter Preset 和可选的 Downstream Port 8.0 GT/s Receiver Preset Hint 字段的值。 要执行重做均衡，下游端口必须在 2.5 GT/s 或 5.0 GT/s 通过 Recovery.RcvrLock 中的 EQ TS1 有序集发起 Speed Change 的请求，以通知上游端口它打算以更高的数据速率进行重做均衡。如果上游端口接收到 EQ TS1 有序集且 Speed Change 位设置为 1b，并且打算将来以更高的数据速率运行，则上游端口应在 Recovery 中通告更高的数据速率。
从 8.0GT/s 到 16.0GT/s 均衡 or 从 16.0GT/s 到 32GT/s 均衡	此处描述的机制对于所有均衡方式都是相同的：初次、重做均衡、自主或软件启动。 当协商更高的数据速率时，下游端口使用 128b/130b 编码将每个通道的发送器的 preset 值传送到上游端口。在协商将数据速率更改为更高数据速率之前，在切换到更高数据速率之前，使用 Recovery.RcvrCfg 中的 128b/130b EQ TS2 Ordered Set（在 Section 4.2.4.1 中定义）发送了这些值。在 128b/130b EQ TS2 Ordered Set 中发送的 preset 值的推导如下： ● 对于以 16.0 GT/s 的速度进行的均衡：对应 Lane 的 16.0 GT/s Lane Equalization Control Register Entry 的 Upstream Port 16.0 GT/s Transmitter Preset 字段的值。 ● 对于以 32.0 GT/s 的速度进行的均衡：对应 Lane 的 32.0 GT/s Lane Equalization Control Register Entry 的 Upstream Port 32.0 GT/s Transmitter Preset 字段的值。 （可选）在协商将数据速率更改为较高数据速率之前，在转换为需要执行均衡的速率之前，上游端口使用 Recovery.RcvrCfg 中发送的 128b/130b EQ TS2 Ordered Set 将初始发送器 preset 设置发送到下游端口。这些 preset 值是通过特定于实现的方式确定的。数据速率更改为较高数据速率后，上游端口将使用其接收到的 preset 值发送 TS1 有序集。如果下游端口未在 Recovery.RcvrCfg 中接收到 preset 值，则在数据速率更改为较高数据速率后，它将发送具有以下 preset 的 TS1 有序集： ● 对于以 16.0 GT/s 的速度进行的均衡：对应 Lane 的 16.0 GT/s Lane Equalization Control Register Entry 的 Downstream Port 16.0 GT/s Transmitter Preset 字段的值。 ● 对于以 32.0 GT/s 的速度进行的均衡：对应 Lane 的 32.0 GT/s Lane Equalization Control Register

公众号：芯片业者相承双

Entry 的 Downstream Port 32.0 GT/s Transmitter Preset 字段的值。

如果发送器将使用 reduced swing，则 preset 值必须在 Section 8.3.3.3 中定义的可操作范围内。

要执行重做均衡，下游端口必须通过 Recovery.RcvrLock 中的 TS1 有序集请求 Speed Change，并且将 Equalization Redo 位设置为 1b，以通知上游端口它打算重做均衡。如果上游端口接收到 TS1 有序集且 Speed Change 位设置为 1b，Equalization Redo 位设置为 1b，并且打算将来以更高的数据速率进行操作，则上游端口应在 Recovery 中通告较高的数据速率。

从 2.5GT/s 或 5.0GT/s 到 32.0GT/s 均衡

仅当链路能够 bypass 均衡到较高数据速率的情况下（即，32.0 GT/s Capability 寄存器的 Equalization bypass to highest rate Supported 为 1 且 32.0 GT/s Control 寄存器的 Equalization bypass to highest rate Disable 为 0），均衡才能从 2.5GT/s 或 5.0GT/s 直接到 32.0GT/s 或更高速率。

此处描述的机制对于所有均衡方式都是相同的：初始、重做均衡、自主或软件启动。

下游端口使用 8b/10b 编码将每个通道的发送器 preset 值和接收器 preset hint 发送到上游端口（如果适用）。在协商了将数据速率更改为较高数据速率之前，在切换到较高数据速率以执行均衡之前，使用 Recovery.RcvrCfg 中的 EQ TS2 有序集（在 Section 4.2.4.1 中定义）发送这些值。在 EQ TS2 有序集中发送的 preset 值的推导如下：

- 要以 32.0 GT/s 的速度进行均衡：对应 Lane 的 32.0 GT/s Lane Equalization Control Register Entry 的 Upstream Port 32.0 GT/s Transmitter Preset 的值。Receiver Preset Hint 字段必须设置为 000b。

在数据速率更改为需要进行均衡的较高数据速率之后，上游端口将使用其接收到的 preset 值发送 TS1 有序集。如果发送器将使用 reduced swing，则 preset 值必须在 Section 8.3.3.3 中定义的可操作范围内。

在数据速率改变到需要执行均衡的更高数据速率后，下游端口发送具有如下 preset 值的 TS1 有序集，假设 preset 值必须在 Section 8.3.3.3 中定义的可操作范围内，前提是发射器将使用 reduced swing：

- 要以 32.0 GT/s 的速度进行均衡：对应 Lane 的 32.0 GT/s Lane Equalization Control Register Entry 的 Downstream Port 32.0 GT/s Transmitter Preset 值。

要执行重做均衡，下游端口必须在 2.5 GT/s 或 5.0 GT/s 通过 Recovery.RcvrLock 中的 EQ TS1 有序集请求 Speed Change，以通知上游端口它打算以更高的数据速率进行重做均衡。 如果上游端口接收到 EQ TS1 有序集且 Speed Change 位设置为 1b，并且打算将来以更高的数据速率运行，则上游端口应在 Recovery 中通告更高的数据速率。

在数据速率等于目标均衡数据速率下进行均衡。

这只有通过重做均衡才有可能。此处涉及的组合包括：8.0 GT/s 数据速率下做 8.0 GT/s 均衡，16.0 GT/s 数据速率下做 16.0 GT/s 均衡和在 32.0 GT/s 数据速率下做 32.0 GT/s 均衡。

在这种情况下，在均衡期间使用的初始 preset 等于在最后一次以进行均衡的数据速率执行均衡时使用的初始 preset。

均衡过程包括以下四个 Phase。当以 8.0 GT/s 的数据速率运行时，将使用 TS1 有序集中的 Equalization Control（EC）字段来传输 Phase 信息。

Phase 0

这个阶段是在协商（和之前）将要执行均衡的数据速率时执行的。如 Table 4-3 所述确定用于均衡的 preset 值。

Phase 1

这两个组件都使链路能够以当前数据速率运行，并可以交换 TS1 有序集，以完成剩余阶段的发送器/接收器对的微调。预计在组件准备好进入下一阶段之前，链路将以小于 10^{-4} 的 BER 运行。每个发送器使用 Table 4-3 中所述的 preset 值。

下游端口通过发送 EC=01b 的 TS1 有序集（表示 Phase 1）来启动 Phase 1。上游端口在调整其接收器（如有必要）以确保其可以进行均衡处理后，接收这些 TS1 有序集并转换到 Phase 1（在该 Phase 中，它以 EC=01b 发送 TS1 有序集）。当下游端口在进入 Phase 2 之前从上游端口接收到 EC=01b 的 TS1 有序集时，下游端口确保能够可靠地从上游端口接收比特流以继续进行其余 Phase。

Phase 2

在此阶段中，上游端口在每个 Lane 上独立地调整下游端口的发送器设置及其自身的接收器设置，以确保它接收到符合 Chapter 8 要求的比特流（比如，所有可用的 Lane 上的 BER 小于 10^{-12}）。下游端口通过将 EC = 10b 的 TS1 有序集发送到上游端口来启动进入 Phase 2。下游端口按照下面的规则通告发送器系数及其所使用的 preset 值，Phase 1 仅发送 preset 值，Phase 2 会发送 preset 值和 coefficients 值。上游端口接收这些有序集，并可能要求使用不同的 preset 和 coefficients，并继续评估每组值，直到达到下游端口 Lane 的最佳设置为止。上游端口完成此阶段后，它将链路 EC = 11b 的 TS1 有序集发送到下游端口，从而将链路切换到 Phase 3。

Phase 3

在此阶段中，下游端口使用类似于 Phase 2 的握手和评估过程，独立地在每个 Lane 上调整上游端口的发送器设置及其自身的接收器设置，只不过 EC 值要设置为 11b。下游端口通过发送 EC = 00b 的 TS1 有序集来发信号通知 Phase 3（以及均衡过程）结束。

组件用来调整其 Link partner 的发送器的算法以及对该发送器设置及其接收器设置的评估是特定于实现的。 组件可以请求更改任意数量的 Lane，并且可以为每个 Lane 请求不同的设置。每个请求的设置可以是 preset 值或一组符合 Section 4.2.3.1 中定义要求的系数。每个组件负责确保在微调结束时（上游端口的 Phase 2 和下游端口的 Phase 3），其 Link partner 在每个 Lane 中都具有最佳的 Transmitter 设置，这将使链路满足 Chapter 8 中的要求。

接收到调整其发送器请求的 Link partner 必须评估该请求并对其执行操作。如果请求有效的 preset 值，并且发送器在 full swing 模式下运行，则必须将其反映在发送器设置中，然后反映在 Link partner 传输的 TS1 有序集的 preset 字段和 coefficients 字段中。如果请求一个预设值，且发送器在 reduced swing 模式下运行，并且如 Section 8.3.3.3 中所定义的那样支持所请求的 preset，它必须反映在发送器设置中，然后反映在 Link partner 传输的 TS1 有序集的 preset 字段和 coefficients 字段中。允许工作在 reduced swing 模式下的发送器拒绝 Section 8.3.3.3 中定义的不支持的预设请求。调整系数的请求可以被接受或拒绝。如果为 Lane 请求的系数集被接受，则必须在发送器设置中反映出来，然后在传输的 TS1 有序集中反映出来。如果为 Lane 请求的系数集被拒绝，则不会更改发送器设置，但发送的 TS1 有序集必须反映所请求的系数以及将 Reject Coefficient 字段设置为 1b。在响应系数请求的任何一种情况下，传输的 TS1 有序集的预设字段都不会从传输的最后一个预设值更改。仅当请求的系数集不符合 Section 4.2.3.1 中定义的规则时，Link partner 才可以拒绝调整系数的请求。

在执行 crosslink 的均衡时，在较早的 crosslink 初始化期间以较低数据速率扮演下游端口角色的组件也承担了下游端口进行均衡的责任。

如果某 Lane 在 Polling.Compliance、Loopback、Recovery.Equalization 的 Phase 0 或 Phase 1 直接使用了保留的或不支持的发送器的 preset 值，则允许该 Lane 以特定的实现方式来支持发送器的 preset 值的设置。保留的或不支持的发送器预设值将在任何后续的 Compliance Patterns 或有序集中传输，而不是由 Lane 选择的特定于实现的预设值。例如，如果上游端口的 Lane 被定向为使用它在 Recovery.RcvrCfg 中接收到的 EQ TS2 有序集使用的发送器的保留值 1111b，则在更改数据后允许将任何受支持的发发送器预设值用于其发射机设置 速率达到 8.0 GT/s，但它必须在 Recovery.Equalization 的 Phase 0 和 Phase 1 发送的 TS1 有序集中发送 1111b 作为其发送器预设值。

在环回状态下，Loopback Master 负责通过 EQ TS1 有序集传送其希望从设备使用的发送器和接收器设置，它以 2.5 GT/s 或 5.0 GT/s 的数据速率以及预设或系数进行传输，希望被测设备在 TS1 有序集下以 8.0 GT/s 数据速率传输的设置。同样，如果通过 TS1 有序集输入了 Polling.Compliance 状态，则执行测试的实体必须根据 Section 4.2.6.2 节中定义的机制发送适当的 EQ TS1 有序集和系数

以供被测设备使用。

IMPLEMENTATION NOTE

Equalization Example

下图是说明两个设备如何完成均衡过程的示例。如果两个端口支持的最大公共数据速率为 8.0 GT/s，则在 8.0 GT/s 均衡过程结束时，均衡过程就完成了。如果两个端口支持的最大公共数据速率为 16.0 GT/s，则在 8.0 GT/s 均衡过程之后执行 16.0 GT/s 均衡过程。如果在链路处于 8.0 GT/s 数据速率（对于 8.0 GT/s 均衡）或 16.0 GT/s（对于 16.0 GT/s 均衡）时，重复执行 8.0 GT/s 或 16.0 GT/s 均衡过程，并且在链路处于执行状态时执行，则可以跳过 Phase 0，因为链路不需要回到 2.5 GT/s 或 5.0 GT/s（对于 8.0 GT/s 均衡）或 8.0 GT/s（对于 16.0 GT/s 均衡）以重新发送相同的 EQ TS2 有序集以传送 preset 值。如果下游端口根据平台中的通道和组件确定不需要对发送器进行微调，则可以选择跳过 Phase 2 和 Phase 3。

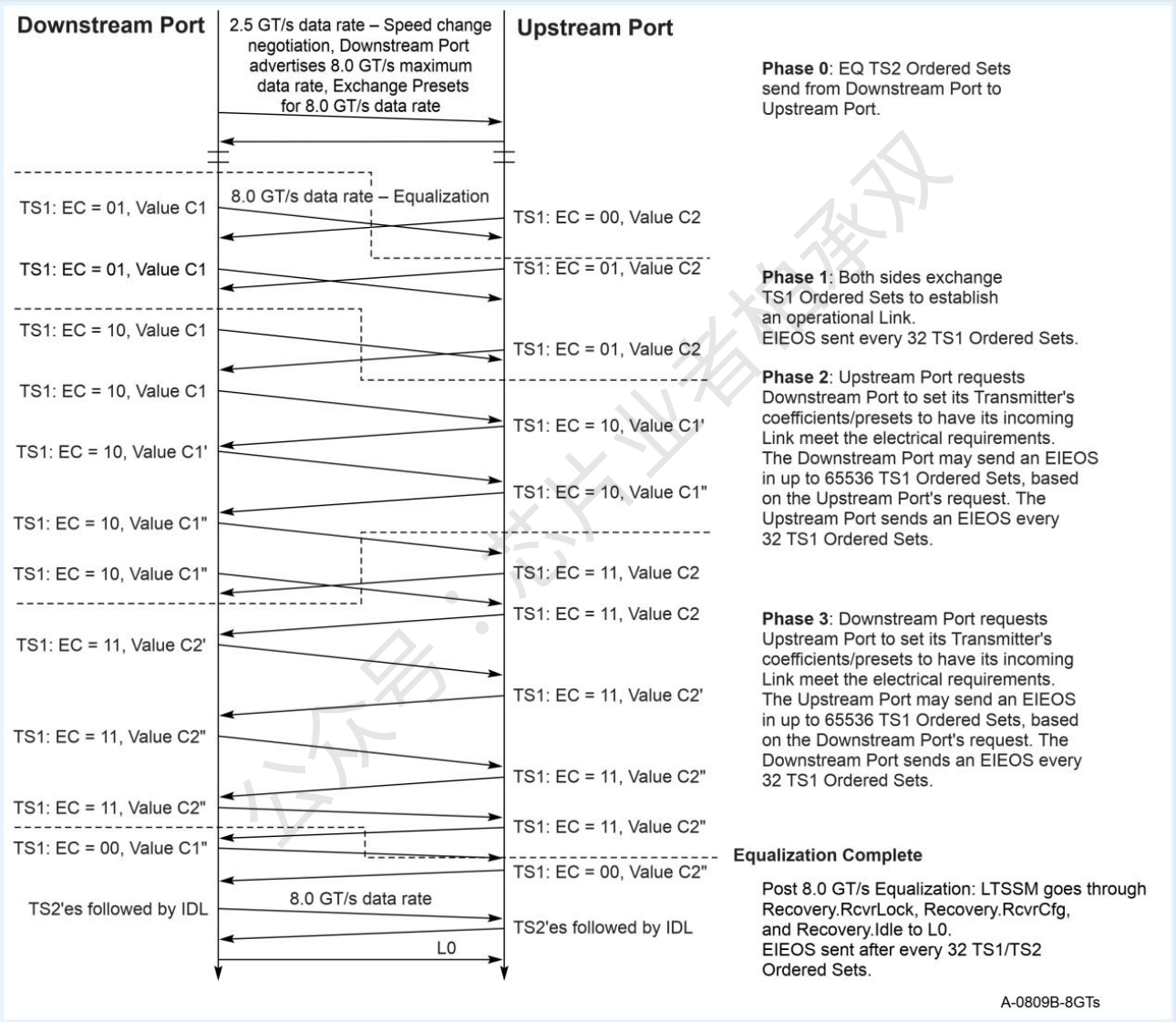


Figure 4-21 8.0 GT/s Equalization Flow

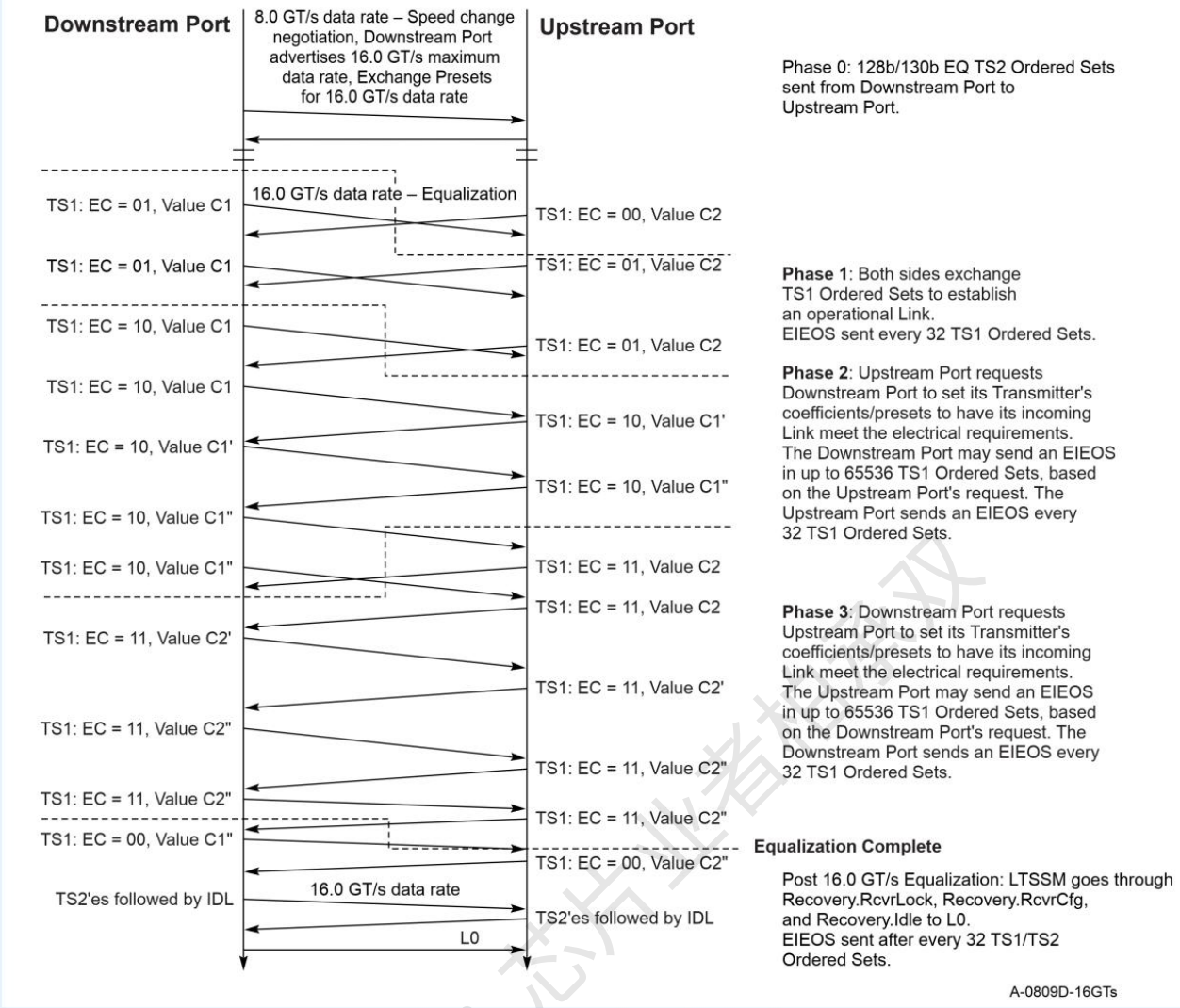


Figure 4-22 16.0 GT/s Equalization Flow

IMPLEMENTATION NOTE

Equalization Bypass Example

下面的流程图提供了一个示例流程，其中链路可以以较低的数据速率 bypass 均衡，并以最高的支持数据速率进行均衡。例如，当 n = 5 时，链路可以将 Gen 1 数据速率训练为 L0，确定所有组件（包括 Retimer，如果有的话）都可以 bypass 的 Gen 5 的均衡，则将数据速率更改为 Gen 5。

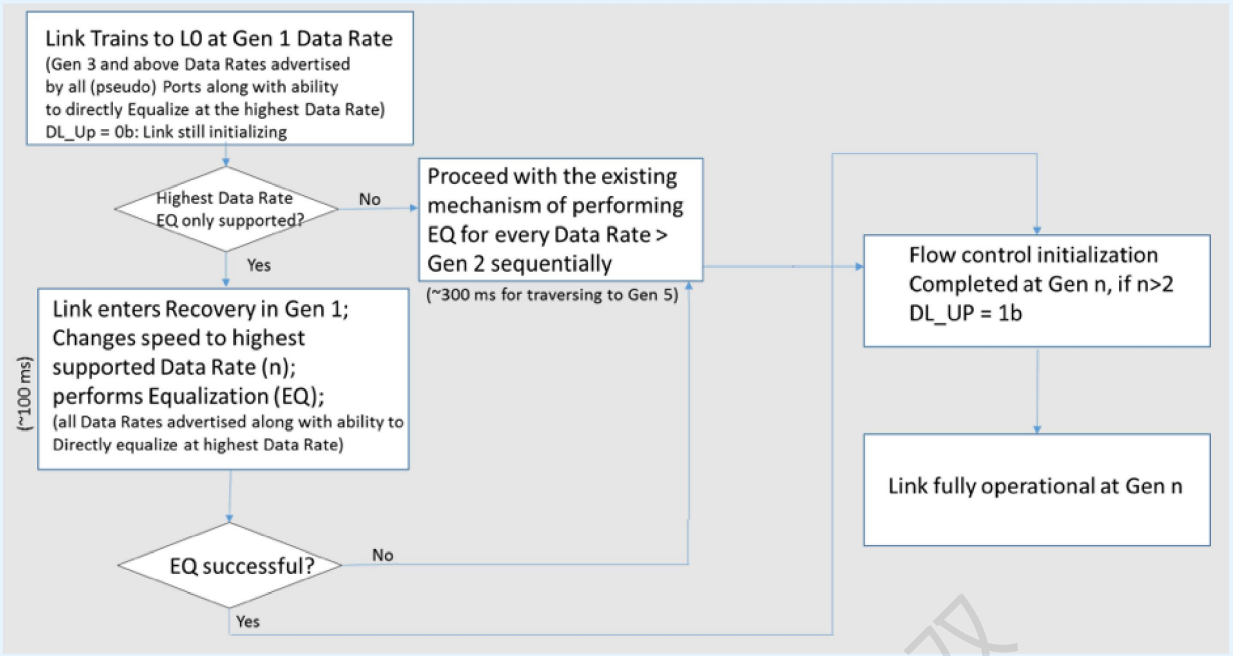


Figure 4-23 Equalization Bypass Example

4.2.3.1 Rules for Transmitter Coefficients

有关系数及其表示的 FIR 滤波器的说明，请参见 Section 8.3.3.1。以下规则适用于通告系数设置和请求系数设置。

1. C_{-1} 和 C_{+1} 是 FIR 方程中使用的系数，分别代表前驱和后驱。TS1 有序集中传递的 pre-cursor 和 post-cursor 值表示其绝对值。C_0 表示 cursor 系数设置，并且是一个正数。

2. 系数的绝对值之和定义了 FS（Full Swing；FS = $|C_{-1}| + C_0 + |C_{+1}|$）。在 Phase 1 中，FS 会通知给 Link partner。发送器 FS 范围定义如下：

   - 对于 Full Swing 模式，FS 值在 24~63 之间。
   - 对于 Reduced Swing 模式，FS 值在 12~63 之间。

3. 发送器在 Phase 1 期间通告其 LF（Low Frequency）值。这对应于发送器可产生的最小差分电压，该最小差分电压是 LF/FS 乘以发送最大差分电压的值。发送器必须确保当下面的公式 c）等于 LF 时，它必须满足 Section 8.3.3.9 中定义的 $V_{TXEIEOS-FS}$ 和 $V_{TX-EIEOS-RS}$ 的电气要求。

4. 在请求 Link partner 的发送器的一组系数之前，必须满足以下规则。接收到 TX 系数设置的更新请求后，端口必须验证新请求满足以下条件，如果违反以下任何条件，则拒绝该请求：

   a) $|C_{-1}| <= Floor (FS/4)$

   b) $|C_{-1}|+C_0+|C_{+1}| = FS$ (Do not allow peak power to change with adaptation.)

   c) $C_0 -|C_{-1}|-|C_{+1}| >= LF$

4.2.3.2 Encoding of Presets

发送器和接收器 Preset Hint 的定义在 Chapter 8 中描述。 Table 4-4 和 Table 4-5 提供了发送器 Preset 和接收器 Preset Hint 的编码。接收器 Preset Hint 是可选的，仅针对 8.0 GT/s 的数据速率进行定义。

公众号：芯片业者相承双

Table 4-4 Transmitter Preset Encoding

Encoding	Preset Number in Table 8-1
0000b	P0
0001b	P1
0010b	P2
0011b	P3
0100b	P4
0101b	P5
0110b	P6
0111b	P7
1000b	P8
1001b	P9
1010b	P10
1011b through 1111b	Reserved

Table 4-5 Receiver Preset Hint Encoding for 8.0 GT/s

Encoding	Receiver Preset Value
000b	-6 dB
001b	-7 dB
010b	-8 dB
011b	-9 dB
100b	-10 dB
101b	-11 dB
110b	-12 dB
111b	Reserved

公众号·芯片业者相承双

4.2.4 Link Initialization and Training

本节定义了物理层控制过程，该过程配置和初始化每个链接以进行正常操作。本节涵盖以下功能：

- configuring and initializing the Link
- supporting normal packet transfers
- supported state transitions when recovering from Link errors
- restarting a Port from low power states

链路训练过程会协商好以下信息：

- Link width
- Link data rate
- Lane reversal
- Lane polarity

链路训练包括以下功能：

- Link data rate negotiation
- Bit lock per Lane
- Lane polarity
- Symbol lock or Block alignment per Lane
- Lane ordering within a Link
- Link width negotiation
- Lane-to-Lane de-skew within a multi-Lane Link

4.2.4.1 Training Sequences

训练序列有序集用于比特对齐、符号对齐以及交换物理层参数。2.5GT/s 和 5GT/s 速率时，训练序列有序集从不加扰，只用 8b/10b 编码。到 8GT/s 及以上速率时，采用 128b/130b 编码，符号有可能加扰有可能不加扰，具体参阅 Section 4.2.2.4。

训练序列（TS1 或 TS2 或 Modified TS1 或 Modified TS2）是连续传输的，并且只能被 SKP 有序集（Section 4.2.7）或对于 2.5 GT/s 以外的数据速率由 EIEOS 有序集中断（Section 4.2.4.3）。

当支持 8.0 GT/s 或更高的数据速率时，使用 8b/10b 编码的 TS1（或 TS2）有序集（即 2.5 或 5.0 GT/s 数据速率）可以是标准 TS1（或 TS2）有序集（即，对于 TS1 有序集，Symbol 6 是 D10.2，对于 TS2 有序集，是 D5.2）或 EQ TS1 有序集（或 EQ TS2 有序集）（即，Symbol 6 的 bit 7 是 1b）。传输 EQ TS1 有序集的能力是特定于实现的。支持 8.0 GT/s 或更高数据速率的端口必须接受 LTSSM 状态中的 TS1（或 TS2）类型，除非明确要求查找特定类型。不支持 8.0 GT/s 数据速率的端口允许（但不是必需）接受 EQ TS1（或 TS2）有序集。

当支持 16.0 GT/s 和更高数据速率时，使用 128b/130b 编码（即 8.0GT/s 或更高数据速率）的 TS2 可以是标准 TS2 有序集（即 Symbol 7 为 45h）或 128b/130b EQ TS2（即，Symbol 7 的 bit 7 为 1b）。除非明确要求查找特定类型，否则支持 16.0 GT/s 或更高数据速率的端口必须在 LTSSM 状态中接受任一 TS2 类型。允许（但不是必需）不支持 16.0 GT/s 数据速率的端口接受 128b/130b EQ TS2 有序集。

使用 8b/10b 编码时，仅当 Symbol 6 与先前的 TS1 或 TS2 有序集的 Symbol 6 相匹配时，才将 TS1 或 TS2 有序集视为连续（传输过程中不能 Symbol 6 的值不能随意改动）。

打算协商备用协议或传递一个 Training Set Message 的组件必须在 Configuration.Lanenum.Wait，Configuration.Lanenum.Accept 和 Configuration.Complete 子状态中使用 Modified TS1/TS2 Ordered Set，而不是 TS1/TS2 Ordered Set。为了可以发送 Modified TS1/TS2 Ordered Set，组件必须在 Polling.Active，Polling.Configuration，TS1/TS2 Ordered Set 中将 TS1 和 TS2 有序集的 Enhanced Link behavior Control 字段设置为 11b， Configuration.Linkwidth.Start 和 Configuration.Linkwidth.Accept 子状态，并执行 LinkUp = 0b 时过渡到 Configuration.Lanenum.Wait 子状态中概述的步骤。 如果链路伙伴不支持 Modified TS1/TS2 Ordered Set，则从 Configuration.LaneNum.Wait 开始，标准 TS 应停止在 Enhanced Link behavior Control 字段中发送 11b 并切换到适当的编码。

当使用 8b/10b 编码时，仅当所有 Symbol 与先前 Modified TS1 或 Modified TS2 Ordered Set 的对应符号匹配且 Symbol 15 中的奇偶

公众号：芯片业者相承双

校验与期望值匹配时，才将 Modified TS1 或 Modified TS2 Ordered Set 视为连续。在链路的所有通道上的每个 Modified TS1/TS2 Ordered Set 中，Symbol 8-14 必须相同。

当使用 128b/130b 编码时，仅当 Symbol 6-9 与先前的 TS1 或 TS2 有序集的 Symbol 6-9 匹配时，才将 TS1 或 TS2 有序集视为连续，并按如下所述处理保留位。

TS1 和 TS2 中保留位的处理规则如下：

- 发送端的保留位为 0。
- 对于接收端：
  - 不得根据接收到的保留位的值确定 TS1 或 TS2 有序集是否无效。
  - 如果保留位包含在奇偶校验计算中，则必须使用接收到的保留位的值进行奇偶校验计算。
  - 可以选择比较在 TS1 或 TS2 有序集中被明确要求相同的符号中的保留位的接收值，以确定它们是否连续。
  - 不得根据接收到的任何保留位的值采取任何功能性动作。

当使用 128b/130b 编码时，要求发送器跟踪 TS1 和 TS2 有序集在链路上（加扰之后）传输的比特流的连续 DC 平衡。正在运行的 DC 平衡是已发送的 1s 数与已发送的 0s 数之差。每个 Lane 必须独立跟踪其运行的 DC 平衡，并且必须能够跟踪任一方向上至少 511 位的差。 所使用的任何计数器都必须在其极限处饱和（而不是翻滚），并在达到极限后继续跟踪减少量。 例如，如果检测到 513 的差异，则可以跟踪 511 位差异的计数器将在 511 饱和，如果将来将其减小 2，则计数器将变为 509。

通过两个事件将运行中的 DC 平衡设置为 0：1）变送器退出电气空闲状态； 2）在数据块之后传输 EIEOS。

对于每个发送的 TS1 或 TS2 有序集，发送器必须评估其 DC 平衡，并发送为 Symbol 14 和 Symbol 15 定义的用于 DC 平衡的 Symbol 之一，如以下算法所定义。如果需要减少 1 的数量，则发送 DC 平衡符号 20h（对于 Symbol 14）和 08h（对于 Symbol 15）。如果需要减少 0，则发送 DC 平衡 Symbol DFh（用于 Symbol 14）和 F7h（用于 Symbol 15）。如果不需要更改，则发送适当的 TS1 或 TS2 标识符符号。为 Symbol 14 或 Symbol 15 传输的任何 DC 平衡符号都会绕过加扰，而 TS1 和 TS2 标识符则遵循标准加扰规则。必须使用以下算法来控制 DC 平衡：

- 如果在 TS 有序集的 Symbol 11 的末尾，运行中的 DC 平衡> 31，则发送 Symbol 14 为 DFh 和 Symbol 15 为 F7h 的 TS 有序集以减少 0 的数量，或者 Symbol 14 为 20h 和 Symbol 15 为 08h 的 TS 有序集以减少 1 的数量。
- 否则，如果在 TS 有序集的 Symbol 11 的末尾正在运行的 DC 平衡> 15，则发送 Symbol 15 为 F7h 的 TS 以减少 0 的数目，或发送 Symbol 15 为 08h 的 TS 以减少 1 的数目。Symbol 14 传输正常的加扰的 TS 标识符。
- 否则，Symbol 14 和 Symbol 15 都传输正常的加扰的 TS 标识符。

当确定 TS1 或 TS2 有序集有效时，允许但不要求接收者检查 Symbol 14 和 Symbol 15 的下列值：解扰后的适当 TS 标识符符号，或在对 Symbol 14 进行解扰之前有效的 DC 平衡符号 DFh 或 20h ，或在对 Symbol 15 进行解扰之前将有效的 DC 平衡符号 F7h 或 08h。

如果接收方在 Symbol 14 中收到 DC balance pattern，则该 pattern 可能会被加扰（并预编码）。因此，如果接收器正在执行此可选检查，则它必须保持解扰器并接收用于检查 Symbol 15 的预编码，该符号可以加扰（并预编码）或 DC balance pattern。

IMPLEMENTATION NOTE

Sync Header and DC Balance

Block Sync Header 以及 TS1 和 TS2 有序集的第一个符号不会影响运行中的 DC 平衡，因为它们具有相等的 1 和 0。

Hot Reset，Disable Link 和 Enable Loopback 的控制字段是互斥的，一次只能设置其中一个，并在已配置的所有 Lane（L0 中的所有 Lane）上传输这些字段。如果同时设置了 Hot Reset，Disable Link 和 Enable Loopback，则链路行为不确定。

通过上游端口或下游端口发送时，TS1 有序集的 Retimer Equalization Extend 字段始终设置为 0b。Retimer 按照 Section 4.3.7.2 中的说明将其设置为 1b。

Table 4-6 ***TS1 Ordered Set***

Symbol Number	Description
0	• When operating at 2.5 or 5.0 GT/s: COM (K28.5) for Symbol alignment. • When operating at 8.0 GT/s or above: Encoded as 1Eh (TS1 Ordered Set).

Symbol Number	Description
1	Link Number. • Ports that do not support 8.0 GT/s or above: 0-255, PAD. • Downstream Ports that support 8.0 GT/s or above: 0-31, PAD. • Upstream Ports that support 8.0 GT/s or above: 0-255, PAD. • When operating at 2.5 or 5.0 GT/s: PAD is encoded as K23.7. • When operating at 8.0 GT/s or above: PAD is encoded as F7h.
2	Lane Number within Link. • When operating at 2.5 or 5.0 GT/s: 0-31, PAD. PAD is encoded as K23.7. • When operating at 8.0 GT/s or above: 0-31, PAD. PAD is encoded as F7h.
3	N_FTS. The number of Fast Training Sequences required by the Receiver: 0-255.
4	Data Rate Identifier **Bit 0** Reserved for future Data Rate. **Bit 1** 2.5 GT/s Data Rate Supported. Must be set to 1b. **Bit 2** 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. See Section 8.2 . **Bit 3** 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b. **Bit 4** 16.0 GT/s Data Rate Supported. Must be set to 1b if Bit 5 is 1b. **Bit 5** 32.0 GT/s Data Rate Supported. **Bit 6** ***Autonomous Change*/*Selectable De-emphasis***. • Downstream Ports: This bit is defined for use in the following LTSSM states: Polling.Active, Configuration.Linkwidth.Start, and Loopback.Entry. In all other LTSSM states, it is Reserved. • Upstream Ports: This bit is defined for use in the following LTSSM states: Polling.Active, Configuration, Recovery, and Loopback.Entry. In all other LTSSM states, it is Reserved. **Bit 7** speed_change. This bit can be set to 1b only in the Recovery.RcvrLock LTSSM state. In all other LTSSM states, it is Reserved.
5	Training Control **Bit 0** ***Hot Reset bit*** **0b** Deassert **1b** Assert **Bit 1** ***Disable Link bit*** **0b** Deassert **1b** Assert **Bit 2** ***Loopback bit*** **0b** Deassert **1b** Assert **Bit 3** ***Disable Scrambling bit*** (2.5 GT/s and 5.0 GT/s data rates)

公众号：志片业者相承双

Symbol Number	Description
	0b Deassert **1b** Assert Reserved (other data rates) **Bit 4** ***Compliance Receive bit*** (5.0 GT/s and above data rates, optional at 2.5 GT/s) **0b** Deassert **1b** Assert Ports that support 5.0 GT/s and above data rate(s) must implement the Compliance Receive bit. Ports that support only 2.5 GT/s data rate may optionally implement the Compliance Receive bit. If not implemented, the bit is Reserved. **Bit 5** ***Transmit Modified Compliance Pattern in Loopback***. This bit is defined for use in Loopback by the Loopback Master when 32.0 GT/s or higher data rates are supported. See Section 4.2.6.10.1 . In all other cases, this bit is Reserved. **Bit 7:6** ***Enhanced Link Behavior Control*** **00b** ***Full Equalization required*** Modified TS1/TS2 Ordered Sets not supported. **01b** ***Equalization bypass to highest rate support*** Modified TS1/TS2 Ordered Sets not supported. Indicates intention to perform 32.0 GT/s equalization when set by Loopback Master. See Section 4.2.3 and Section 4.2.6.10.1 . **10b** ***No Equalization Needed*** Modified TS1/TS2 Ordered Sets not supported A device advertising this capability must support Equalization bypass to highest rate. See Section 4.2.3 . **11b** ***Modified TS1/TS2 Ordered Sets supported*** Equalization bypass options specified in Modified TS1/TS2 Ordered Sets. These bits are defined for use in Polling and Configuration when LinkUp=0b and 32.0 GT/s or higher data rates are supported and in Loopback by the Loopback Master when 32.0 GT/s or higher data rates are supported. In all other cases, these bits are Reserved.
6	When operating at 2.5 or 5.0 GT/s: • Standard TS1 Ordered Sets encode this Symbol as a TS1 Identifier, D10.2 (4Ah). • EQ TS1 Ordered Sets encode this Symbol as follows: ◦ For Equalization at 8.0 GT/s Data Rate: **Bits 2:0** ***Receiver Preset Hint***. See Section 4.2.3.2 . **Bit 6:3** ***Transmitter Preset***. See Section 4.2.3.2 . **Bit 7** Set to 1b. ◦ For Equalization at 32.0 GT/s or higher Data Rate: **Bit 0** Transmitter Precode Request - See Section 4.2.2.5 . This bit has no defined usage in receivers at this time. **Bit 2:1** Reserved **Bit 6:3** Transmitter Preset. See Section 4.2.3.2 . **Bit 7** Set to 1b. When operating at 8.0 GT/s or higher data rate:

Symbol Number	Description
	Bit 1:0 Equalization Control (EC). These bits are only used in the Recovery.Equalization and Loopback LTSSM states. See Section 4.2.6.4.2 and Section 4.2.6.10 . In all other LTSSM states, they must be set to 00b. **Bit 2** Reset EIEOS Interval Count. This bit is defined for use in the Recovery.Equalization LTSSM state. See Section 4.2.6.4.2 and Section 4.2.4.3 . In all other LTSSM states, it is Reserved. **Bit 6:3** Transmitter Preset. See Section 4.2.3 and Section 4.2.6 . **Bit 7** ***Use Preset***/Equalization Redo. This bit is defined for use in the Recovery.Equalization, Recovery.RcvrLock and Loopback LTSSM states. See Section 4.2.6.4.1 , Section 4.2.6.4.2 and Section 4.2.6.10 . In all other LTSSM states, it is Reserved.
7	• When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah). • When operating at 8.0 GT/s or higher: **Bit 5:0** FS when the EC field of Symbol 6 is 01b (see Section 4.2.3.1). Otherwise, Pre-cursor Coefficient for the current data rate of operation. **Bit 6** ***Transmitter Precoding on***. This bit is defined for use in the Recovery state for use at 32.0 GT/s or higher. See Section 4.2.2.5 . In all the other cases, it is Reserved. **Bit 7** ***Retimer Equalization Extend*** bit. This bit is defined for use in the Recovery.Equalization LTSSM state when operating at 16.0 GT/s or higher data rate. In all other LTSSM states and when operating at 8.0 GT/s, it is Reserved.
8	• When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah). • When operating at 8.0 GT/s or higher data rate: **Bit 5:0** LF when the EC field of Symbol 6 is 01b (see Section 4.2.3.1). Otherwise, Cursor Coefficient for the current data rate of operation. **Bit 7:6** Reserved.
9	• When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah). • When operating at 8.0 GT/s or higher data rate: **Bit 5:0** Post-cursor Coefficient for the current data rate of operation. **Bit 6** ***Reject Coefficient Values bit***. This bit can only be set to 1b in specific Phases of the Recovery.Equalization LTSSM State. See Section 4.2.6.4.2 . In all other LTSSM states, it must be set to 0b. **Bit 7** Parity (P). This bit is the even parity of all bits of Symbols 6, 7, and 8 and bits 6:0 of Symbol 9. Receivers must calculate the parity of the received bits and compare it to the received Parity bit. Received TS1 Ordered Sets are valid only if the calculated and received Parity match.
10 - 13	• When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah). • When operating at 8.0 GT/s or above: TS1 Identifier. Encoded as 4Ah.
14 - 15	• When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah). • When operating at 8.0 GT/s or above: TS1 Identifier (encoded as 4Ah) or a DC Balance Symbol.

公众号：芯片业者相承双

Table 4-7 ***TS2 Ordered Set***

Symbol Number	Description
0	• When operating at 2.5 or 5.0 GT/s: COM (K28.5) for Symbol alignment. • When operating at 8.0 GT/s or above: Encoded as 2Dh (TS2 Ordered Set).
1	Link Number. • Ports that do not support 8.0 GT/s or above: 0-255, PAD. • Downstream Ports that support 8.0 GT/s or above: 0-31, PAD. • Upstream Ports that support 8.0 GT/s or above: 0-255, PAD. • When operating at 2.5 or 5.0 GT/s: PAD is encoded as K23.7. • When operating at 8.0 GT/s or above: PAD is encoded as F7h.
2	Lane Number within Link. • When operating at 2.5 or 5.0 GT/s: 0-31, PAD. PAD is encoded as K23.7. • When operating at 8.0 GT/s or above: 0-31, PAD. PAD is encoded as F7h.
3	N_FTS. The number of Fast Training Sequences required by the Receiver: 0-255.
4	Data Rate Identifier **Bit 0** Reserved for future Data Rate **Bit 1** 2.5 GT/s Data Rate Supported. Must be set to 1b. **Bit 2** 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. See Section 8.2 . **Bit 3** 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b. **Bit 4** 16.0 GT/s Data Rate Supported. Must be set to 1b if Bit 5 is 1b. **Bit 5** 32.0 GT/s Data Rate Supported **Bit 6** Autonomous Change/Selectable De-emphasis/Link Upconfigure Capability. This bit is defined for use in the following LTSSM states: Polling.Configuration, Configuration.Complete, and Recovery. In all other LTSSM states, it is Reserved. **Bit 7** speed_change. This bit can be set to 1b only in the Recovery.RcvrCfg LTSSM state. In all other LTSSM states, it is Reserved.
5	Training Control **Bit 0** Hot Reset bit **0b** Deassert **1b** Assert **Bit 1** Disable Link bit **0b** Deassert **1b** Assert **Bit 2** Loopback bit

公众号·芯片业者相承双

Symbol Number	Description
	0b Deassert **1b** Assert **Bit 3** Disable Scrambling bit in 2.5 GT/s and 5.0 GT/s data rates; Reserved in other data rates **0b** Deassert **1b** Assert **Bit 4** ***Retimer Present bit*** in 2.5 GT/s data rate. Reserved in other data rates. **0b** No Retimers present **1b** One or more Retimers present **Bit 5** ***Two Retimers Present bit*** in 2.5 GT/s data rate. Reserved in other data rates. Ports that support 16.0 GT/s data rate or higher must implement this bit. Ports that support only 8.0 GT/s data rate or lower are permitted to implement this bit. **0b** Zero or one Retimers present **1b** Two or more Retimers present **Bit 7:6** Enhanced Link Behavior Control **00b** Full Equalization required, Modified TS1/TS2 Ordered Sets not supported. **01b** Equalization bypass to highest rate support Modified TS1/TS2 Ordered Sets not supported. See Section 4.2.3 . **10b** No Equalization Needed, A device advertising this capability must support Equalization bypass to highest rate. See Section 4.2.3 . Modified TS1/TS2 Ordered Sets not supported **11b** Modified TS1/TS2 Ordered Sets supported, Equalization bypass options specified in Modified TS1/TS2 Ordered Sets. These bits defined for use in Polling and Configuration when LinkUp=0 and 32.0 GT/s or higher data rate is supported. In all other cases, Bits 7:6 are Reserved.
6	• When operating at 2.5 or 5.0 GT/s: ◦ Standard TS2 Ordered Sets encode this Symbol as a TS2 Identifier, D5.2 (45h). ◦ EQ TS2 Ordered Sets encode this Symbol as follows: ▪ For Equalization at 8.0 GT/s Data Rate: **Bit 2:0** Receiver Preset Hint. See Section 4.2.3.2 . **Bit 6:3** Transmitter Preset. See Section 4.2.3.2 . **Bit 7** Set to 1b. ▪ For Equalization at 32.0 GT/s or higher Data Rate: **Bit 0** ***Transmitter Precode Request***. See Section 4.2.2.5 . **Bit 2:1** Reserved **Bit 6:3** Transmitter Preset. See Section 4.2.3.2 . **Bit 7** Set to 1b. • When operating at 8.0 GT/s or higher Data Rate: **Bit 3:0** Reserved. **Bit 5:4** ***Equalization Request Data Rate***.

公众号：芯片业者相关双

Symbol Number	Description
	00b 8.0 GT/s **10b** 16.0 GT/s **01b** 32.0 GT/s **11b** Reserved These bits are defined for use in the Recovery.RcvrCfg LTSSM state. In all other LTSSM states, they are Reserved. See Section 4.2.3 for usage and recognize that these bits are non-sequentially encoded for purposes of backwards compatibility **Bit 6** ***Quiesce Guarantee***. This bit is defined for use in the Recovery.RcvrCfg LTSSM state. In all other LTSSM states, it is Reserved. **Bit 7** ***Request Equalization***. This bit is defined for use in the Recovery.RcvrCfg LTSSM state. In all other LTSSM states, it is Reserved.
7	• When operating at 2.5 or 5.0 GT/s: TS2 Identifier. Encoded as D5.2 (45h). • When operating at 8.0 GT/s or above: ◦ Standard TS2 Ordered Sets encode this Symbol as a TS2 Identifier, 45h. ◦ 128b/130b EQ TS2 Ordered Sets encode this Symbol as follows: **Bit 0** Transmitter Precode Request for operating at 32.0 GT/s or higher Data Rate. See Section 4.2.2.5 . This bit is Reserved if the 128b/130b EQ TS2 is sent for equalization at data rates of 8.0 GT/s or 16.0 GT/s. **Bit 2:1** Reserved **Bit 6:3** 128b/130b Transmitter Preset. See Section 4.2.3.2 . **Bit 7** Set to 1b. This definition is only valid in the Recovery.RcvrCfg LTSSM state when Preset values are being communicated.
8 - 13	• When operating at 2.5 or 5.0 GT/s: TS2 Identifier. Encoded as D5.2 (45h). • When operating at 8.0 GT/s or above: TS2 Identifier. Encoded as 45h.
14-15	• When operating at 2.5 or 5.0 GT/s: TS2 Identifier. Encoded as D5.2 (45h). • When operating at 8.0 GT/s or above: TS2 Identifier (encoded as 45h) or a DC Balance Symbol.

Table 4-8 ***Modified TS1/TS2 Ordered Set*** *(8b/10b encoding)*

Symbol Number	Description
0	COM (K28.5) for Symbol alignment.
1	Link Number. Downstream Ports: 0-31, PAD (K23.7). Upstream Ports: 0-255, PAD (K23.7).

公众号·芯片业者相承攻

Symbol Number	Description
2	Lane Number within Link : 0-31, PAD. PAD is encoded as K23.7.
3	N_FTS. The number of Fast Training Sequences required by the Receiver: 0-255.
4	Data Rate Identifier **Bit 0** Reserved for future Data Rate. **Bit 1** 2.5 GT/s Data Rate Supported. Must be set to 1b. **Bit 2** 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. See Section 8.2 . **Bit 3** 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b. **Bit 4** 16.0 GT/s Data Rate Supported. Must be set to 1b if Bit 5 is 1b. **Bit 5** 32.0 GT/s Data Rate Supported. **Bit 6** ***Link Upconfigure Capability*** **Bit 7** Reserved.
5	Training/ Equalization Control **Bit 0** Equalization bypass to highest rate support. See Section 4.2.3 **Bit 1** ***No Equalization Needed bit***. See Section 4.2.3 **Bit 3:2** Reserved **Bit 4** Retimer Present bit **0b** No Retimers present **1b** One Retimer is present **Bit 5** Two or more Retimers Present **Bit 6** 1b **Bit 7** 1b
6	For Modified TS1: TS1 Identifier, encoded as D10.2 For Modified TS2: TS2 Identifier, encoded as D5.2
7	For Modified TS1: TS1 Identifier, encoded as D10.2 For Modified TS2: TS2 Identifier, encoded as D5.2
8-9	**Bits 2:0** ***Modified TS Usage*** **000b** PCIe protocol only **001b** PCIe protocol only with vendor defined ***Training Set Messages*** **010b** ***Alternate Protocol Negotiation*** **011b through 111b** Reserved The values advertised in these bits must be consistent with the Modified TS Usage Mode Selected field of the 32.0 GT/s Control register and the capabilities of the device. These are bits[2:0] of Symbol 8. **Bits 15:3** ***Modified TS Information 1***

公众号·芯片业者相承双

Symbol Number	Description
	If Modified TS Usage = 001b or 010b; else Reserved.
10-11	***Training Set Message Vendor ID*** if Modified TS Usage = 001b. ***Alternate Protocol Vendor ID*** if Modified TS Usage = 010b. Reserved for other cases.
12-14	If Modified TS Usage = 001b or 010b, ***Modified TS Information 2*** Else, Reserved
15	Bit-wise even parity of Symbols 4 through 14. . For example: Bit 0 = Symbol 4 Bit [0] ^ Symbol 5 Bit [0] ^ …. ^ Symbol 14 Bit [0], … , Bit [7] = Symbol 4 Bit [7] ^ Symbol 5 Bit [7] ^ …. ^ Symbol 14 Bit [7]

扩展到多个符号上的 Modified TS1/TS2 Ordered Set 中的字段使用 little endian 格式，该格式使用这些多个符号上的所有位。例如，Modified TS1/TS2 Ordered Set 的 Symbol 8 和 9 总共 16 位。Modified TS Usage 字段位于 Symbol 8 的 bit[2:0]中，Modified TS Usage 字段的 bit 0 位于 Symbol 8 的 bit 0 中，Modified TS Usage 字段的 bit 1 位于 Symbol 8 的 bit 1 中，Modified TS Usage 字段的 bit 2 位于 Symbol 8 的 bit 2 中。类似地，Modified TS Information 1 字段的 13 位的 bit 12 被放置在 Symbol 9 的 bit 7 中，其中 Modified TS Information 1 的 bit 0 被放置在 Symbol 8 的 bit 3 中。

4.2.4.2 Alternate Protocol Negotiation

除了跳过均衡外，还可以在 LinkUp = 0b 时的 Configuration.Lanenum.Wait，Configuration.Lanenum.Accept 和 Configuration.Complete 子状态期间在 8b/10b 编码中通过交换 Modified TS1/TS2 Ordered 协商备用协议（Alternate Protocol）。

PCIe PHY 可以在 128b/130b 编码支持备用协议。备用协议是使用 PCIe PHY 层定义的一种 non-PCIe 协议。在备用协议模式下，可以选择在运行 PCIe 协议之外运行一种或多种备用协议。Ordered Set 块与 SKP Ordered Set 插入的规则以及 Ordered Set 和 Data Blocks 之间的过渡一起使用。但是，可以根据备用协议的规则修改数据块的内容。

IMPLEMENTATION NOTE

Alternate Protocols should have an EDS Token Equivalent

EDS Tok 在 PCI Express 中用于指示从数据块到有序集块的切换。此附加的"冗余"信息可确保 2 比特的 block header 中的随机位错误不会被错误地解释为数据流的末尾。这是 PCI Express 使用的一种机制，可实现 4 单位的无误数据错误汉明距离。

备用协议应具有等效的机制。

下图表示发生备用协议和均衡旁路协商的状态：

公众号·芯片业者相承攻

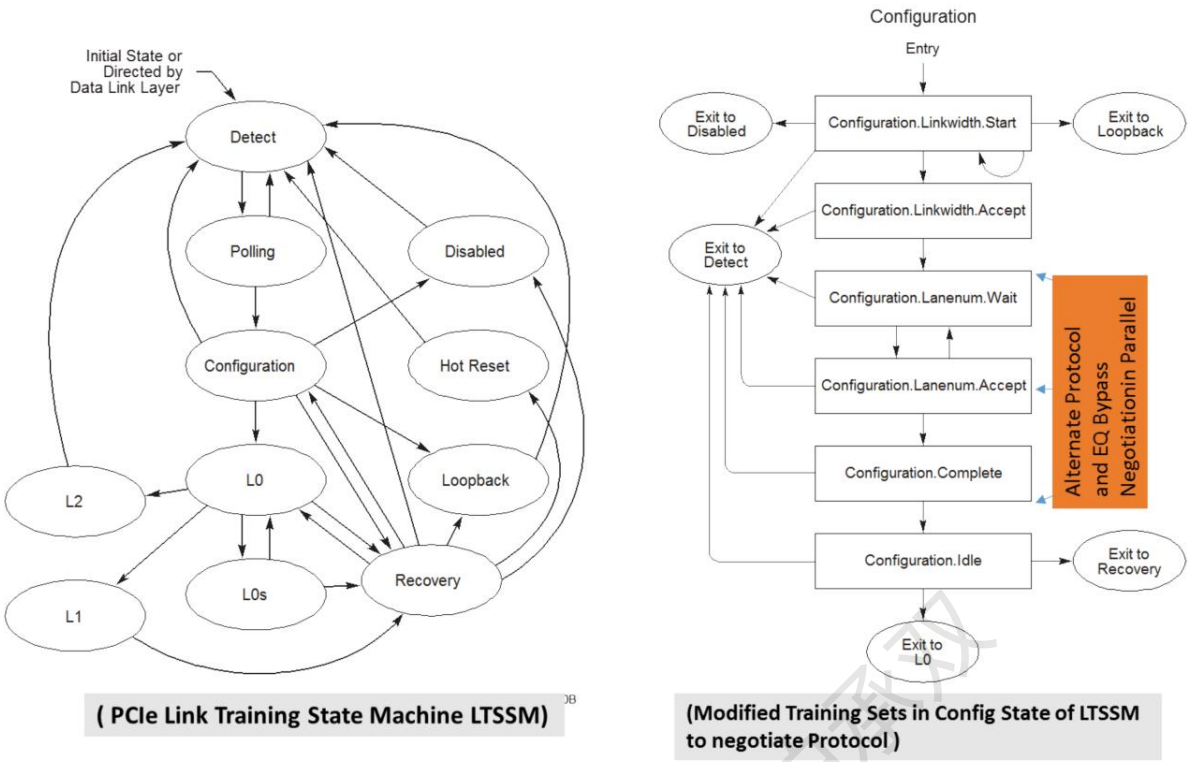


Figure 4-24 Alternate Protocol Negotiation and Equalization Bypass LTSSM States

当端口处于 LinkUp = 0 的 Configuration.Lanenum.Wait、Configuration.Lanenum.Accept 和 Configuration.Complete 子状态时，下游端口根据 Modified TS Usage Mode Selected 字段的值来管理 Alternate Protocol Negotiation 和 Training Set Message。

上游端口必须通过传输 000b 的 Modified TS Usage 值来响应不受支持的 Modified TS Usage 。

如果 Modified TS Usage Mode Selected 值为：

000b

没有备用协议协商或训练集消息发生。该链路将用作 PCI Express 链接。

001b

Training Set Message 被启用。Modified TS Information 1 和 Modified TS Information 2 字段携带由 Training Set Message Vendor ID 字段定义的特定于供应商的消息。

010b

Alternate Protocol Negotiation 被启用。Modified TS Information 1 和 Modified TS Information 2 字段携带由 Alternate Protocol Vendor ID 字段定义的备用协议详细信息。协议请求或响应与 Alternate Protocol Detail 和 Alternate Protocol Vendor ID 确定的协议相关联。协议请求或响应与 Alternate Protocol Vendor ID 字段定义的协议相关联。

Alternate Protocol Negotiation Status 字段指示协商协议的进度。

others

Reserved

支持 Alternate Protocol Negotiation 的下游端口在首次进入 Configuration.Lanenum.Wait， LinkUp = 0 且 Modified TS Usage Mode Selected 字段为 010b 时，将启动协商过程。开始协商包括发送 Modified TS Usage = 010b 的 Modified TS1/TS2 Ordered Set。

公众号·芯片业者相关权

Table 4-9 Modified TS Information 1 field in Modified TS1/TS2 Ordered Sets if Modified TS Usage = 010b (Alternate Protocol)

Bits	Field	Description
4:3	***Alternate Protocol Negotiation Status***	For Modified TS1 Ordered Sets: **00b** **DP** Indicates a protocol request from the Downstream Port asking whether the Upstream Port supports a particular alternate protocol. **UP** Indicates that the Upstream Port does not have an answer for a protocol request yet. This occurs either when it is evaluating the protocol request or it has not received two consecutive Modified TS1s to perform the evaluation. In the former case, Alternate Protocol Vendor ID and Alternate Protocol Details reflect what it received, while Modified TS Information 2 is protocol specific. In the latter case, all 3 fields must be 0. **01b** **DP** Reserved **UP** Indicates that the Upstream Port does not support the requested protocol. Alternate Protocol Vendor ID and Alternate Protocol Details reflect what it received. Modified TS Information 2 must be all 0s. **10b** **DP** Reserved **UP** Indicates that the Upstream Port supports the requested protocol. Alternate Protocol Vendor ID and Alternate Protocol Details reflect what it received, while Modified TS Information 2 field is protocol specific. **11b** Reserved For Modified TS2 Ordered Sets: **00b** Indicates a protocol confirmation from the Downstream Port as well as the Upstream Port. Behavior is undefined if the Downstream Port had not earlier received status 10b for this protocol in this instance of protocol negotiation during the Modified TS1 Ordered Sets. Similarly, behavior is undefined if the Upstream Port had not earlier transmitted status 10b for this protocol in this instance of protocol negotiation during the Modified TS1 Ordered Sets. No protocol is selected unless the Downstream Port sends and receives a protocol confirmation in the Modified TS2 Ordered Sets. If the Downstream Port decides not to use any Alternate Protocol, it may optionally indicate this by transmitting Modified TS2 Ordered Set with Modified TS Usage of 000b. **01b, 10b, 11b** Reserved
15:5	***Alternate Protocol Details***	Alternate Protocol Details is Modified TS Usage = 010b.

如果 Modified TS Usage = 001b，则 Modified TS Information 1 和 Modified TS Information 2 包含训练集消息的详细信息。

备用协议协商必须与通道编号协商同时进行。DP 负责确保在过渡到 Configuration.Complete 子状态之前，他们就备用协议协商达成共识。如果备用协议协商未达成共识，则允许回退到 PCIe 协议。在成功协商备用协议后，链路将以 2.5 GT/s 的速率切换到 L0，之后将数据速率切换至更高的数据速率，并在需要时执行均衡，并以所需的最高数据速率进入 L0。在执行均衡后，以最高数据速率发送 SDS 有序集后，数据块将携带替代协议，并且链路将处于替代协议的控制之下。

4.2.4.3 Electrical Idle Sequences

除非另有规定，否则发送器进入电气空闲前，必须始终发送 Electrical Idle Ordered Set Sequence（EIOSQ）。如果当前数据速率为 2.5 GT/s、8.0 GT/s、16.0 GT/s 或 32.0 GT/s，则定义为一个 EIOS；如果当前数据速率为 5.0 GT/s，则定义为两个连续 EIOS。

使用 8b/10b 编码时，EIOS 是 K28.5（COM），后跟三个 K28.3（IDL）符号。发送器必须发送 EIOS 的所有符号。当收到 COM 和三个 IDL 符号中的两个时，就认为收到 EIOS。当使用 128b/130b 编码时，EIOS 是一个有序集块，如 Table 4-11 所定义。如果要在其后发送其他 EIOS，则发送端必须发送 EIOS 的所有符号。否则，发送器必须发送 EIOS 的 Symbol 0-13，但允许在 Symbol 14 或 Symbol 15 中的任何地方终止 EIOS，之后链路过渡到电气空闲状态。当有序集块的 Symbol 0-3 与 EIOS 的定义匹配时，则认为已接收到 EIOS。

IMPLEMENTATION NOTE

Truncation of EIOS Ordered Set

最后的 EIOS 的截断可帮助实现以下实现：发送器可能终止于内部时钟边界，由于 128b/130b 编码，该内部时钟边界可能无法与符号边界对齐。截断是可以的，因为接收方只需查看前四个符号即可得出它是 EIOS 的结论。

传输完最后一个电气空闲有序集的最后一个符号后，发送器必须处于 $T_{TX\text{-}IDLE\text{-}SET\text{-}TO\text{-}IDLE}$ 指定的有效电气空闲状态（见 Table 8-7）。

Table 4-10 Electrical Idle Ordered Set (EIOS) for 2.5 GT/s and 5.0 GT/s Data Rates

Symbol Number	Encoded Values	Description
0	K28.5	COM for Symbol alignment
1	K28.3	IDL
2	K28.3	IDL
3	K28.3	IDL

Table 4-11 Electrical Idle Ordered Set (EIOS) for 8.0 GT/s and Above Data Rates

Symbol Numbers	Value	Description
0-15	66h	EIOS Identifier and Payload

Table 4-12 Electrical Idle Exit Ordered Set (EIEOS) for 5.0 GT/s Data Rate

Symbol Number	Encoded Values	Description
0	K28.5	COM for Symbol alignment
1-14	K28.7	EIE - K Symbol with low frequency components for helping achieve exit from Electrical Idle
15	D10.2	TS1 Identifier (See Note 1)

Table 4-13 Electrical Idle Exit Ordered Set (EIEOS) for 8.0 GT/s Data Rates

Symbol Numbers	Value	Description
0, 2, 4, 6, 8, 10, 12, 14	00h	Symbol 0: EIEOS Identifier A low frequency pattern that alternates between eight 0s and eight 1s.
1, 3, 5, 7, 9, 11, 13, 15	FFh	A low frequency pattern that alternates between eight 0s and eight 1s.

Table 4-14 Electrical Idle Exit Ordered Set (EIEOS) for 16.0 GT/s Data Rate

Symbol Numbers	Value	Description
0, 1, 4, 5, 8, 9, 12, 13	00h	Symbol 0: EIEOS Identifier A low frequency pattern that alternates between sixteen 0s and sixteen 1s.
2, 3, 6, 7, 10, 11, 14, 15	FFh	A low frequency pattern that alternates between sixteen 0s and sixteen 1s.

公众号：芯片业者相承双

Table 4-15 Electrical Idle Exit Ordered Set (EIEOS) for 32.0 GT/s Data Rate

Symbol Numbers	Value	Description
0, 1, 2, 3, 8, 9, 10, 11	00h	Symbol 0: EIEOS Identifier A low frequency pattern that alternates between thirty-two 0s and thirty-two 1s.
4, 5, 6, 7, 12, 13, 14, 15	FFh	A low frequency pattern that alternates between thirty-two 0s and thirty-two 1s.

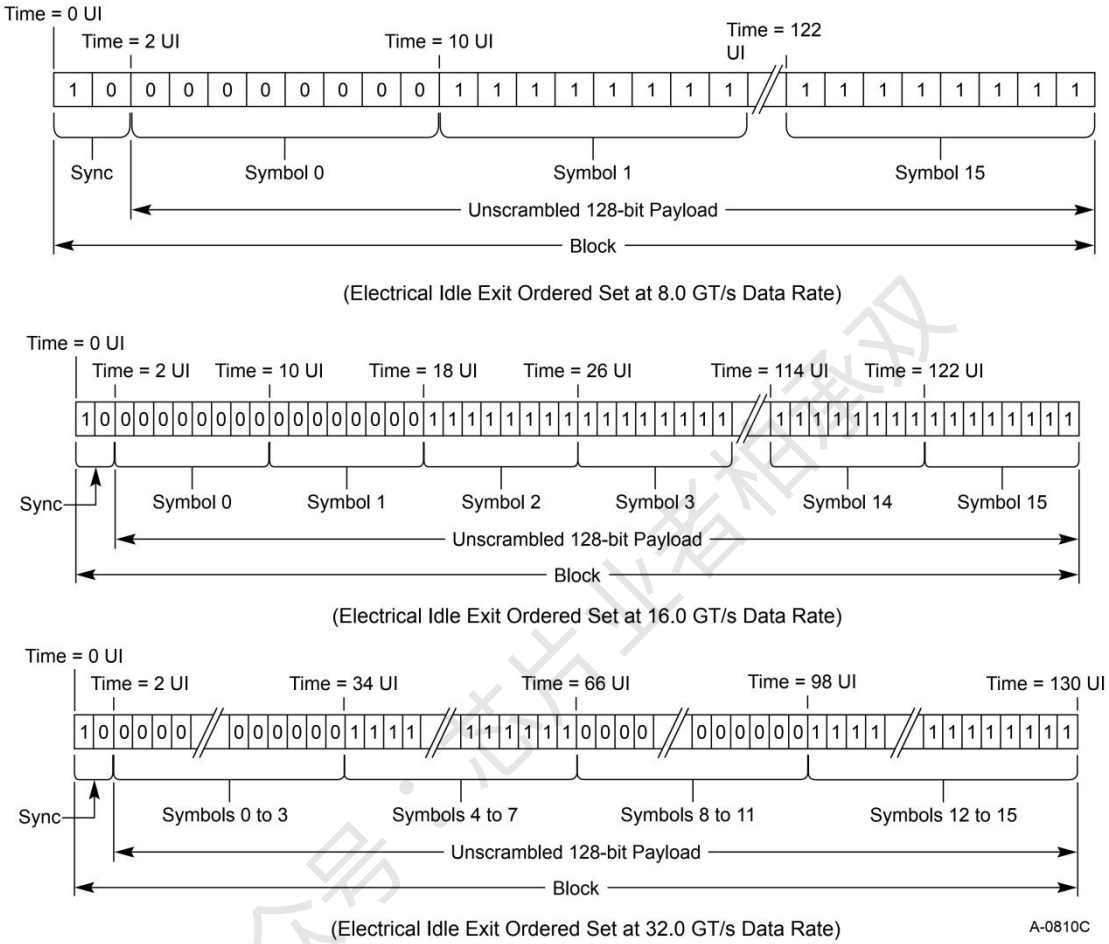


*Figure 4-25 Electrical Idle Exit Ordered Set for 8.0 GT/s and Above Data Rates (**EIEOS**)*

仅当以 2.5 GT/s 以外的速度运行时，才会发送 EIEOS。它是周期性发送的低频的 pattern，有助于确保接收机的电空闲出口电路可以检测到电空闲的退出。当使用 128b/130b 编码时，它也可用于块对齐，如 Section 4.2.2.2.1 所述。

数据速率为 32.0 GT/s 及以上时电气空闲退出有序集序列（EIEOSQ）包括两个连续的 EIEOS，数据速率为 5.0 GT/s，8.0 GT/s 和 16.0 GT/s 时为一个 EIEOS。必须将两个以 32.0 GT/s 的速度运行的 EIEOS 背对背并不能中断，才能被认为是连续的并形成一个 EIEOSQ。不管 EIEOSQ 的长度如何，在 EIEOS 上仍然会发生块对齐。

当在 5GT/s 下运行，EIEOSQ（Table 4-2）的规则如下：

- 进入 LTSSM 的 Configuration.Linkwidth.Start 状态后，在发送第一个 TS1 有序集之前会发送 EIEOS。
- 进入 LTSSM 的 Recovery.RcvrLock 状态后，在发送第一个 TS1 有序集之前会发送 EIEOS。
- 在 LTSSM 的 Configuration.Linkwidth.Start，Recovery.RcvrLock 和 Recovery.RcvrCfg 状态中每发送 32 个 TS1 或 TS2 有序集之后。在以下情况下，TS1/TS2 计数会设置为 0：

- 发送了一个 EIEOS。
- 在 LTSSM 的 Recovery.RcvrCfg 状态收到第一个 TS2 序列。

当采用 128b/130b 编码时，EIEOSQ（Table 4-13 到 Table 4-15 和 Figure 4-25）的规则如下：

- 进入 LTSSM 的 Configuration.Linkwidth.Start 状态后，在发送第一个 TS1 有序集之前会发送 EIEOS。
- 进入 LTSSM 的 Recovery.RcvrLock 状态后，在发送第一个 TS1 有序集之前会发送 EIEOS。
- 在结束数据流且不发送 EIOS 且未进入 LTSSM 的 Recovery.RcvrLock 子状态时，紧随 EDS Framing Token 之后会发送 EIEOS。
- 所有需要传输 TS1 或 TS2 有序集的 LTSSM 状态下，每 32 个 TS1 或 TS2 有序集发送之后。在以下情况下，TS1 / TS2 计数会设置为 0：
  - 发送了一个 EIEOS。
  - 在 LTSSM 的 Recovery.RcvrCfg 状态收到第一个 TS2 序列。
  - 在 LTSSM 的 Configuration.Complete 状态收到第一个 TS2 序列。
  - 下游端口处于 LTSSM 的 Recovery.Equalization 状态的 Phase 2，且在任意 Lane 上收到的两个连续的 TS1 有序集中的 Reset EIEOS Interval Count 字段设置为 1。
  - 上游端口处于 LTSSM 的 Recovery.Equalization 状态的 Phase 3，且在任意 Lane 上收到的两个连续的 TS1 有序集中的 Reset EIEOS Interval Count 字段设置为 1。
- 在 LTSSM 的 Recovery.Equalization 状态下每发送 65,536 个 TS1 有序集之后，如果 Reset EIEOS Interval Count 阻止了该间隔的发送，则将在此状态下发送。允许实现通过在加扰 LFSR 匹配其种子值时在两个 TS1 有序集中发送 EIEOS 来满足此要求。
- 作为相关部分中所述的 FTS Ordered Set、Compliance Pattern 或 Modified Compliance pattern 的一部分发送。

Example：LTSSM 以 5.0 GT/s 的数据速率从 L0 切换到 Recovery.RcvrLock。它先传输一个 EIEOS，然后传输 TS1 有序集。它发送 32 个 TS1 有序集，之后发送第二个 EIEOS。随后，它再发送两个 TS1 有序集，并进入 Recovery.RcvrCfg，在传输 30 个 TS2 有序集后，接着传输第三个 EIEOS。当收到 TS2 有序集时，它将在 Recovery.RcvrCfg 中再发送 31 个 TS2 有序集（在前 30 个 TS2 有序集之后）。由于它接收到第一个 TS2 有序集，因此它将 EIEOS interval count 清为 0，并在继续向 Recovery.Idle 转换之前继续发送另外 16 个 TS2 有序集。因此，由于 EIEOS interval count 已重置为 0b，因此在最后 47 个 TS2 有序集中间未发送 EIEOS。LTSSM 从 Recovery.Idle 切换到 Configuration.Linkwidth.Start 并传输 EIEOS，此后它开始传输 TS1 有序集。

以 2.5 GT/s 以外的速度运行时，允许实现不依赖电空闲检测电路的输出，除非在某些 LTSSM 状态期间接收 EIEOS，或在 Rx L0s 或 Polling.Compliance 期间在接收器处接收到由四个连续 EIE 符号（见 Section 4.2.4.6）预先提供的 FTS。

4.2.4.4 Inferring Electrical Idle

允许设备在所有操作速度下推断电气空闲状态，而不是使用模拟电路检测电气空闲状态。Table 4-16 总结了推断各种子状态下的电气空闲的条件。

公众号：芯片学者相承汉

Table 4-16 Electrical Idle Inference Conditions

State	2.5 GT/s	5.0 GT/s	8.0 GT/s and higher data rates
L0	Absence of Flow Control Update DLLP[56] or alternatively a SKP Ordered Set in a 128 µs window	Absence of Flow Control Update DLLP[57] or alternatively a SKP Ordered Set in a 128 µs window	Absence of Flow Control Update DLLP[58] or alternatively a SKP Ordered Set in a 128 µs window
Recovery.RcvrCfg	Absence of a TS1 or TS2 Ordered Set in a 1280 UI interval	Absence of a TS1 or TS2 Ordered Set in a 1280 UI interval	Absence of a TS1 or TS2 Ordered Set in a 4 ms window
Recovery.Speed when successful_speed_negotiation = 1b	Absence of a TS1 or TS2 Ordered Set in a 1280 UI interval	Absence of a TS1 or TS2 Ordered Set in a 1280 UI interval	Absence of a TS1 or TS2 Ordered Set in a 4680 UI interval
Recovery.Speed when successful_speed_negotiation = 0b	Absence of an exit from Electrical Idle in a 2000 UI interval	Absence of an exit from Electrical Idle in a 16000 UI interval	Absence of an exit from Electrical Idle in a 16000 UI interval
Loopback.Active (as slave)	Absence of an exit from Electrical Idle in a 128 µs window	N/A	N/A

注 56,57,58：Flow Control Update DLLP 是本规范中定义的 UpdateFC 或[MR-IOV]中定义的 MRUpdateFC。

不得基于对电气空闲条件的推断来确定电气空闲出口条件。考虑到电路面积因素，允许选择为每个 LTSSM 实现一个公共的超时计数器，并在 LTSSM 控制的每个 Lane 的公共计数器确定的公共超时窗口内查找电气空闲推断条件，而不用为每条 Lane 设置超时计数器。

IMPLEMENTATION NOTE

Inference of Electrical Idle

在 L0 状态下，期望在 128 µs 窗口中接收到一定数量的流控制更新 DLLP。同样在 L0 中，预计将平均以 2.5 GT/s 和 5.0 GT/s 的数据速率每发送 1538 个符号，并以 8.0 GT/s 的数据速率平均发送 375 个块。为简化起见，允许使用这些指示器中的一个（或两个）来推断电气空闲。因此，可以将在任何 128 µs 窗口中不存在流控制更新 DLLP 或 SKP 有序集的情况推断为电气空闲。通过成功的速度协商，在 Recovery.RcvrCfg 以及 Recovery.Speed 中，接收方应连续接收 TS1 或 TS2 有序集，但 EIEOS 和 SKP 有序集除外。因此，对于实现推断机制的组件，必须在上述指定的间隔内不存在 TS1 或 TS2 有序集。如果设备以 success_speed_negotiation = 0b 进入 Recovery.Speed，则设备可能无法接收符号。因此，电空闲推断是在没有电气空闲退出的情况下完成的。在非 2.5 GT/s 的数据速率下，只有在收到 EIEOS 时才能保证电空闲退出。因此，该窗口设置为 16000 UI，用于以 5.0 GT/s 和 8.0 GT/s 的数据速率检测从电气空闲退出。在 2.5 GT/s 的数据速率下，必须在收到每个符号后检测到电气空闲出口。因此，在 2000 UI 窗口中没有电空闲出口就构成了电气空闲条件。

4.2.4.5 Lane Polarity Inversion

在 Polling 过程中，接收器将 TS1 和 TS2 有序集的 Symbol 6-15 视为通道极性反转的指示符（D +和 D-被交换）。 如果发生通道极性反转，则收到的 TS1 的 Symbol 6-15 的值为 D21.5，而不是预期的 D10.2。同样，如果发生通道极性反转，则 TS2 有序集的 Symbol 6-15 为 D26.5，而不是预期的 D5.2。

如果检测到通道反转，则接收器必须反转接收到的数据。发送器绝不能反转发送的数据。所有通道上的所有 PCI Express 接收器都需要独立支持通道极性反转。

公众号：芯片业者相承双

4.2.4.6 Fast Training Sequence (FTS)

快速训练序列（FTS，Fast Training Sequence）是从 L0s 过渡到 L0 时用于比特锁定和符号锁定的机制。接收器使用 FTS 来检测从电气空闲状态的退出，并将接收器的 bit and Symbol receive circuitry 与输入数据对齐。 有关 L0s 和 L0 的说明，请参见 Section 4.2.5 。

- **对于 2.5GT/s 和 5GT/s 速率：**

一个 FTS 包含一个 K28.5（COM）符号，然后是三个 K28.1 符号。组件可以请求的 FTS（N_FTS）的最大数量为 255，提供 4 * 255 * 10 * UI 的时间用于 bite 锁定。如果数据速率为 5.0 GT/s，则在发送第一个 FTS 之前以有效信号电平发送四个连续的 EIE 符号。这些符号将帮助接收器检测到从电气空闲状态退出。由于退出电气空闲状态需要在其第一个 FTS 之前加上额外的 EIE 符号，因此该实现不能保证在发送器引脚上的信号在允许的时间内达到合适的电平（见 Table 4-18），因此接收器至少可以接收有效信号电平下的四个 EIE 符号。在传输第一个 FTS 之前，实现不得传输八个以上的 EIE 符号。允许组件在不同的速率下通告不同的 N_FTS 速率。在 5.0 GT/s 的速度下，组件可以选择通告合适的 N_FTS 值，因为它将接收到四个 EIE 符号。如果 Link Control 寄存器的 Extended Synch 字段被设置，则必须发送 4096 个 FTS，以便为外部链路监视工具提供足够的时间来实现比特同步和帧同步。为了满足 Section 4.2.7 中的定义，必须在 FTS 之间调度和传输 SKP 有序集，但在第一个 N_FTS FTS 期间不能传输 SKP 有序集。最后一个 FTS 传输后，始终发送单个 SKP 有序集。请注意，有可能可以背对背传输两个 SKP 有序集（一个 SKP 有序集表示 4096 个 FTS 的完成，而一个 SKP 有序集则进行了调度和传输以满足 Section 4.2.7 中描述的定义）。

注：N_FTS 表示 FTS 的个数，在 TS1 和 TS2 序列的 Symbol 3 中通告。

- **对于 8GT/s 和以上速率：**

单个 FTS 是一个 130 位不加扰的有序集块，如 Table 4-17 所示。组件可以请求的 FTS（N_FTS）的最大数量为 255，提供 130 * 255 UI（如果包括周期性 EIEOS，则为 130 * 263 UI）的时间用于比特锁定。允许组件在不同的速度下通告不同的 N_FTS 值。从 L0s 退出时，发送器首先发送一个 EIEOSQ，该块将帮助接收器检测到由于其低频成分而退出电气空闲的情况。在第一个 EIEOSQ 之后，发送器必须发送所需数量的 FTS（设置 Link Control 寄存器的 Extended Synch 字段时为 4096；否则为 N_FTS），每 32 个 FTS 之后发送一次 EIEOSQ。 FTS 序列将使接收器获得比特锁定（并且可以选择进行块对齐）。设置 Extended Synch 字段后，要根据 Section 4.2.7 描述的规则，在 FTS 和 EIEOSQ 之间调度传输 SKP 有序集。如果有的话，FTS 序列的最后一个 FTS 有序集（如果 N_FTS 等于零，则不发送 FTS 有序集）后跟最后一个 EIEOSQ，它将帮助接收器实现块对齐。如果 N_FTS 是 32 的倍数，则即使在最后一个 FTS 有序集之后的数据速率低于 32.0 GT/s 的情况下，实现也可以背对背发送两个 EIEOS。EIEOS 重置发送器和接收器中的加扰器。在最终的 EIEOSQ 之后，将发送 SDS 有序集，以帮助接收器执行 de-skew 并指示从有序集到数据流的过渡。在发送 SDS 有序集之后，必须发送一个数据块。

IMPLEMENTATION NOTE

Scrambling LFSR During FTS Transmission in 128b/130b Encoding

由于加扰器在最后一个 EIEOS 上重置，并且 FTS 序列中的任何有序集都没有被加扰，因此在接收 FTS 期间，实现是否选择推进加扰器不重要。

公众号：芯片业老柏承双

Table 4-17 FTS for 8.0 GT/s and Above Data Rates

Symbol Number	Value
0	55h
1	47h
2	4Eh
3	C7h
4	CCh
5	C6h
6	C9h
7	25h
8	6Eh
9	ECh
10	88h
11	7Fh
12	80h
13	8Dh
14	8Bh
15	8Eh

N_FTS 定义了从 L0s 切换到 L0 时必须发送的 FTS 的数量。在 2.5 GT/s 的数据速率下，组件可以请求的用于 Symbol Lock 的时间对应于 16 ns（N_FTS 设置为 0b 和一个 SKP 有序集）～4 µs（N_FTS 设置为 255），当设置 Extended Synch 字段时，需要发送 4096 个 FTS，导致 bit lock 时间为 64 µs。对于 8.0 GT/s 和更高的数据速率，将 Extended Synch 字段设置为 1 时，要求发送器发送 4096 个 FTS 有序集块。请注意，组件报告的 N_FTS 值可能会更改； 例如，由于软件修改了 Common Clock Configuration 字段（Section 7.5.3.7）的值。

如果在接收器获得 bit lock、Symbol lock 或 Block alignment 以及 Lane-to-Lane de-skew 之前，N_FTS 时间已用完，则接收器必须切换到 Recovery 状态。该过程在 LTSSM 的 Section 4.2.5 中有详细说明。

4.2.4.7 Start of Data Stream Ordered Set

Table 4-18 和 Table 4-19 中描述的 Start of Data Stream（SDS）有序集仅针对 128b/130b 编码定义。它在 Configuration.Idle，Recovery.Idle 和 Tx_L0s.FTS 等 LTSSM 状态中使用，以定义从有序集块到数据流的过渡，并且允许 Loopback Masters 按照 Section 4.2.2.6 中的描述进行传输。不得在其他时间传输该有序集。在处于非环回状态时，SDS 有序集之后的块必须是数据块，并且该数据块的第一个符号是数据流的第一个符号。

Table 4-18 SDS Ordered Set (for 8.0 GT/s and 16.0 GT/s Data Rate)

Symbol Number	Value	Description
0	E1h	SDS Ordered Set Identifier
1-15	55h	Body of SDS Ordered Set

Table 4-19 SDS Ordered Set (for 32.0 GT/s and higher Data Rate)

Symbol Number	Value	Description
0	E1h	SDS Ordered Set Identifier
1-15	87h	Body of SDS Ordered Set

4.2.4.8 Link Error Recovery

- 如果在 8b/10b 编码下发生 Link Errors：
  - 包括 8b/10b 解码错误，Framing Errors，loss of Symbol lock，Elasticity Buffer Overflow/Underflow，以及 loss of Lane-to-Lane de-skew。
  - 必须检查 8b/10b 解码错误，并在指定的 LTSSM 状态下触发 Receiver Error（请参见 Table 4-20），这是与端口相关的 reported error（请参见 Section 6.2）。任何其它 Framing Error、Loss of Symbol Lock、Lane Deskew Error 和 Elasticity Buffer Overflow/Underflow 触发的 Receiver Error 都是可选择实现的。
- 如果在 128b/130b 编码下发生 Link Errors：
  - 包括 Framing Errors、 loss of Block Alignment、 Elasticity Buffer Overflow/Underflow 或者 loss of Lane-to-Lane de-skew。
  - 必须检查 Framing Errors，并在指定的 LTSSM 状态下触发 Receiver Error（请参见 Table 4-20），这是与端口相关的 reported error（请参见 Section 6.2）。任何其它 Block Alignment、Elasticity Buffer Overflow/Underflow 和 loss of Lane-to-Lane de-skew 触发的 Receiver Error 都是可选择实现的。
- 在一个已配置完成且进入 L0 状态的链路上，可以通过指示链路切换到 Recovery 状态，这个功能需要在物理层之上的层中实现（如 Section 3.6 所述）。
  - 注意：链路错误也可能导致物理层启动从 L0 到 Recovery 的转换。
- 发生链路错误时，除 L0 以外的所有 LTSSM 状态都会执行 Link Error Recovery（根据上下文，除 Detect、 Hot Reset、 Disabled 外）。
  - 当使用 8b/10b 编码进行操作时，非 L0 的 LTSSM 状态中发生的链路错误一定不会导致物理层启动 LTSSM 状态转换。
  - 当使用 128b/130b 编码进行操作且不处理数据流时，非 L0 的 LTSSM 状态中发生的链路错误一定不会导致物理层启动 LTSSM 状态转换。
- 使用 8b/10b 编码进行操作时，如果 Lane 检测到实现特定数量的 8b/10b 错误，则必须尽快检验或重新建立符号锁定。检验和重新建立符号锁定的方法是特定于实现的。

4.2.4.9 Reset

Section 6.6 从系统角度描述了复位。

4.2.4.9.1 Fundamental Reset

当 Fundamental Reset 有效时：

- 接收器终端负载要满足 $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}POS}$ 和 $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}NEG}$（请参阅 Table 8-10）。
- 发送器只需要满足 $I_{TX\text{-}SHORT}$（Table 8-7）。
- 发送器维持一个常量的 DC 共模电压。

当 Fundamental Reset 释放后：

- 该端口的 LTSSM 完成初始化（参阅 Section 4.2.5 和 Section 6.6.1）。

公众号·芯片业者相承双

4.2.4.9.2 Hot Reset

Hot Reset 的规范在 Section 4.2.5.11 定义。

4.2.4.10 Link Data Rate Negotiation

要求所有设备在每个通道上以 2.5 GT/s 的数据速率开始链路初始化。TS1 和 TS2（请参阅 Section 4.2.4.1）中的 Symbol 4 用于通告所有支持的数据速率。链路最初以 2.5 GT/s 的数据速率训练到 L0，此后通过进入 Recovery 状态来切换数据速率。

4.2.4.11 Link Width and Lane Sequence Negotiation

PCI Express 链路必须由并行的 1、2、4、8、12、16 或 32 通道组成，分别称为 x1，x2，x4，x8，x12，x16 和 x32 链路。链路中的所有通道必须同时基于相同的频率传输数据，并且通道之间的时滞不得超过 $L_{TX-SKEW}$（Table 8-10）。协商过程有很多步骤。

协商为有效链路的每个通道建立 Link number 和 Lane number 值；每个不属于有效链路的通道都将退出协商，以成为分离的链路或保持在电气空闲状态。

在 Link width 和 Lane number 协商期间，两个通信端口必须满足 Table 8-10 中 $L_{RX-SKEW}$ 指定的最大的 Lane-to-Lane skew 值。

可选的链路协商行为包括通道反转，可变宽度的链路，将端口拆分为多个链路以及交叉链路的配置。

其他规范可能会施加其他规则和限制，而这些规则和限制必须由那些符合这些其他规范的组件来理解； 本规范的目的是理解各种组件功能的互操作性。

4.2.4.11.1 Required and Optional Port Behavior

- xN 端口需要有组成 xN 链路和 x1 链路（其中 N 可以是 32、16、12、8、4、2 和 1）的能力。
  - 设计人员必须以允许这些组件满足上述要求的方式连接两个不同组件之间的端口。如果组件之间的端口连接方式与组件的端口说明/数据表中定义的预期用途不一致，则行为未定义。
- xN 端口组成 N 到 1 之间的任何链路宽度的能力是可选的。
  - 比如一个 x16 端口，该端口只能配置为一个链路，但是链路的宽度可以配置为 x12，x8，x4，x2 以及所需的 x16 和 x1。
- 一个端口拆分成多个链路的功能是可选的。
  - 比如一个 x16 端口，该端口可能能够配置两个 x8 链接，四个 x4 链接或 16 个 x1 链接。
- 通道反转功能是可选的。
  - 如果实现，则必须对多通道链路的给定端口的发送器和接收器都进行通道反转。
  - 通道反转的示例为下游端口的 Lane N-1 连接到上游端口的 Lane 0，其中下游或上游设备可以颠倒通道顺序以配置 xN 链接。

交叉链路功能是可选的。在这种情况下，连接到下游端口的下游端口或连接到上游端口的上游端口是交叉链接。

当前和将来的机械特性和/或形状因数规格可能需要实现上面列出的一些可选功能。组件设计者必须阅读他们正在设计的组件所使用的系统的规格，以确保符合这些规格。

公众号：芯片业者相承双

4.2.4.12 Lane-to-Lane De-skew

在将数据和控制传送到数据链路层之前，接收器必须补偿多通道链路中所有通道之间允许的 skew（请参见 Table 8-7 和 Table 8-10）。

当使用 8b/10b 编码时，Lane-to-Lane de-skew 机制可以使用以下一项或多项来实现：

- The COM Symbol of a received TS1 or TS2 Ordered Set
- The COM Symbol of a received Electrical Idle Exit Ordered Set
- The COM Symbol of the first received SKP Ordered Set after an FTS sequence
- The COM Symbol of a received SKP Ordered Set during a training sequence when not using SRIS

使用 128b/130b 编码时，Lane-to-Lane de-skew 机制可以使用以下一项或多项来实现：

- A received SDS Ordered Set
- A received Electrical Idle Exit Ordered Set
- A received SKP Ordered Set

也可以采用其他 de-skew 机制。Lane-to-Lane de-skew 必须在 LTSSM 的 Configuration、Recovery、和 L0s 状态执行。

IMPLEMENTATION NOTE

Unambiguous Lane-to-Lane De-Skew:

接收器必须能够消除的 2.5 GT/s 的 max skew 为 20 ns。标准的 SKP 有序集（即没有通过 Retimer 添加或删除的 SKP 符号）的长度为 4 个符号，即 16 ns，速率为 2.5 GT/s。通常，SKP 有序集的传输使得它们间隔良好，并且不需要特别注意将它们的 skew，即，它们提供了明确的机制。如果传输了背对背的 SKP 有序集，则仅查找 SKP 有序集的 COM 出现在同一时间的每个通道上的实现可能会失败。当退出 L0s 时，发送器可以在快速训练序列的最后一个 FTS 有序集合之后发送背对背的 SKP 有序集合。必须在 L0s 中实现 De-skew，因此在这种情况下执行 De-skew 时，实现必须理解背靠背的 SKP 有序集。

Section 4.2.4.12 中明确机制的例外会发生，因为可能会发送背对背有序集，即，当使用 128b/130b 编码退出 L0s 时，可能会背对背发送 EIEOS。在这种情况下，EIEOS 仍可用于 de-skew，但是，执行 de-skew 时，实现必须包含背靠背 EIEOS。

当以 16.0 GT/s 或更高的数据速率运行时，发送器可以在训练序列结束时发送背对背的 SKP 有序集，例如 TS2 有序集，SKP 有序集，SKP 有序集，SDS 有序集。建议在这种情况下选择使用 SKP 有序集进行 de-skew 的实现，以认识到背对背 SKP 有序集是不同的，即标准 SKP 有序集和控制 SKP 有序集。

4.2.4.13 Lane vs. Link Training

链路初始化过程将端口的未关联通道构建为链路的关联通道以构造链路。 为了使通道正确配置为所需的链路，TS1 和 TS2 有序集必须在所有通道上将适当的字段（Symbol 3、4 和 5）设置为相同的值。

Configuration 状态完成后会形成链路。

如果 Port 实现了可以配置成多条链路的功能，则可以观察到以下情况：

- 每个单独的链路，都需要单独的 LTSSM。
- LTSSM 规则只用于配置一个链路。在多条链路的情况下，以串行或并行方式配置链路的决定是特定于实现的。

公众号：志业者相承双

4.2.5 Link Training and Status State Machine (LTSSM) Descriptions

LTSSM 状态如 Figure 4-26 所示。这些状态将在以下各节中介绍。

除非另有明确说明，否则为 LTSSM 指定的所有超时值均为负 0 秒加 50%。Fundamental Reset 后，所有超时值必须设置为指定的值。Fundamental Reset 后，所有计数器值必须设置为指定值。

4.2.5.1 Detect Overview

该状态的功能是检测远端设备是否存在。可以从任何状态直接定向到该状态。

4.2.5.2 Polling Overview

LTSSM 在 Polling 状态发送训练有序集并响应接收到的训练有序集。在这个状态下，完成比特锁定和符号锁定，并配置通道极性。

Polling 状态包括 Polling.Compliance（请参阅 Section 4.2.6.2.2）。此状态旨在与测试设备一起使用，用于评估被测设备中的发送器和互连是否符合 Table 8-6、Table 8-7 和 Table 8-10 中的电压和时序规范。

Polling.Compliance 状态还包括一个简化的互操作性测试方案，该方案旨在使用各种各样的测试和测量设备（即码型发生器，示波器，BERT 等）来执行测试过程。在 Polling.Active 状态发送 Compliance Receive 字段被置为 1 的 TS1 序列（注意，Loopback 字段不能置为 1）可以使状态机转移到 Polling.Compliance 状态。设置 Compliance Receive 的方法是特定于实现的。还包括将数据速率更改为由最高的公共发送和接收数据速率标识符（TS1 的符号 4）指示的速率的措施，以使此行为可扩展到各种数据速率。

IMPLEMENTATION NOTE

Use of Polling.Compliance

Polling.Compliance 是针对一致性测试环境而设计的，在正常操作期间不会进入，并且不能由任何原因禁用。由特定的物理系统环境，或配置寄存器可以导致进入 Polling.Compliance 状态，如 Section 4.2.6.2.1 所述。导致发送器输出一致性码型的任何其他机制都是特定于实现的，并且超出了本规范的范围。

4.2.5.3 Configuration Overview

在 Configuration 状态，发送器和接收器均以协商的数据速率发送和接收数据。通过 Lane 宽度协商形成链路。同时，还要执行 Lane-to-Lane de-skew 过程，如果允许的话加扰过程被禁用，N_FTS 被设置，也可能进入 Disable 或 Loopback 状态。

4.2.5.4 Recovery Overview

在 Recovery 状态，发送器和接收器都使用已配置的链路和通道号以及支持的数据速率发送和接收数据。Recovery 允许已配置的链路根据需要调整数据速率，重新建立 bit lock、Symbol lock 或 Block alignment、Lane-to-Lane de-skew 。Recovery 还用于设置新的 N_FTS 值，并进入 Loopback、Disabled、Hot Reset 和 Configuration 状态。

4.2.5.5 L0 Overview

L0 状态是正常的数据传输状态。所有的低功耗状态都需要从这个状态开始。

公众号·芯片业者相承双

4.2.5.6 L0s Overview

L0s 是一个省电状态。当使用具有独立扩频时钟（SSC）的独立参考时钟工作时（请参见 Section 4.2.7），不支持 L0s，并且不得在 Capability 寄存器中通告支持 L0s。有关 SSC 的定义，请参见 Section 4.3.7.3。

L0 允许链路快速进入省电状态并从省电状态中恢复，而无需经过 Recovery 状态。

当收到一个 EIOS 后链路进入 L0s。

从 L0s 退出然后进入到 L0 需要重新建立 bit lock、Symbol lock 或 Block alignment、Lane-toLane de-skew 。

一个端口发送器和接收器的链路不必都进入 L0s。

4.2.5.7 L1 Overview

L1 是一个省电状态。

L1 状态比 L0 节省了更多的功耗，但增加了恢复延迟。

L1 的进入是在数据链路层参与下完成的，在接收到 EIOS 之后发生。

4.2.5.8 L2 Overview

L2 可以更节省功耗。在这个状态，发送器和接收器的电源几乎全部关掉，主电源和时钟是不能保证的，但是辅助电源还在正常工作。

当系统支持 Beacon 时，支持唤醒功能的上游端口必须能够发送 Beacon；下游端口必须能够接收 Beacon； 这就是称为 Beacon 的唤醒信号。

进入 L2 需要数据链路层的参与，在接收到 EIOS 之后发生。

4.2.5.9 Disabled Overview

Disabled 状态的功能是：允许禁用已配置的链路直到定向或电气空闲（例如，由热插拔导致）退出。

链路中发送的 TS1 和 TS2 有序集中 Training Control 字段的 bit 1（Disable Link）会导致进入 Disabled 状态。

在上层逻辑的参与下，链路可以进入 Disabled 状态。链路也可以通过接收两个连续的 TS1 有序集并将其 Disable Link 置位来进入 Disabled 状态（请参阅 Section 4.2.6.3.1 和 Section 4.2.6.4.5）。

4.2.5.10 Loopback Overview

Loopback 状态的目的旨在用于测试和故障隔离。 规范仅定义了进入和退出行为，所有其他详细信息是特定于实现的。Loopback 可以基于每个通道或已配置的链路进行操作。

发起 Loopback 请求的叫做 Loopback Master。

环回数据的叫做 Loopback Slave。

设置 TS1 和 TS2 的 Loopback 字段会导致进入 Loopback 状态。

公众号·芯片业者相承双

Loopback Master 怎么实现进入 Loopback 状态是基于特定实现的。

Loopback Slave 收到两个 loopback 字段设置为 1 的 TS1 或 TS2 就进入 Loopback 状态。

IMPLEMENTATION NOTE

Use of Loopback

一旦处于 Loopback 状态，只要遵循编码规则，Master 就可以发送任意码型的符号。一旦进入 Loopback 状态，数据加扰的概念就不再相关了。发出的内容将回环回来。数据链路层用来通知物理层进入环回状态的机制和/或接口是组件实现特定的，不在本规范的范围之内。

4.2.5.11 Hot Reset Overview

设置 TS1 和 TS2 的 Hot Reset 字段会导致进入 Hot Reset 状态。

链路进入 Hot Reset 的方式由上层逻辑决定。链路只要收到两个 Hot Reset 字段设置为 1 的 TS1 或 TS2 序列就会进入 Hot Reset 状态。

4.2.6 Link Training and Status State Rules

Table 4-20 说明了如何在整个 LTSSM 中处理链路状态信息（有关更多信息，请参见 Section 3.2 中的 LinkUp；Section 7.5.3.8 中的 Link Speed、Link Width、和 Link Training ；Section 6.2 中的 Receiver Error；以及 Section 6.7 中的 In-Band Presence）。当以 8b/10b 编码运行时，在 Table 4-20 中允许接收器将错误报告为 Receiver Error 时，接收器还可以选择在 Lane Error Status 寄存器中报告 8b/10b 错误。

IMPLEMENTATION NOTE

Receiver Errors During Configuration and Recovery States

在 Configuration 状态或 Recovery 状态中设置 Receiver Errors 的功能旨在允许报告在这两个状态中数据交换时发生的链路错误。例如，如果在接收 TLP 时 LTSSM 从 L0 切换到 Recovery 状态，则可以报告 LTSSM 切换之后发生的链路错误。

Table 4-20 Link Status Mapped to the LTSSM

LTSSM State	Link Width	Link Speed	LinkUp	Link Training	Receiver Error	In-Band Presence[64]
Detect	Undefined	Undefined	0b	0b	No action	0b

公众号·芯片业者拍承双

LTSSM State	Link Width	Link Speed	LinkUp	Link Training	Receiver Error	In-Band Presence
Polling	Undefined	Set to 2.5 GT/s on entry from Detect. Link speed may change on entry to Polling.Compliance.	0b	0b	No action	1b
Configuration	Set	No action	0b/ 1b[65]	1b	Set on 8b/10b Error. Optional: Set on Link Error when using 128b/ 130b encoding.	1b
Recovery	No action	Set to new speed when speed changes	1b	1b	Optionally set on Link Error.	1b
L0	No action	No action	1b	0b	Set on Link Error.	1b
L0s	No action	No action	1b	0b	No action	1b
L1	No action	No action	1b	0b	No action	1b
L2	No action	No action	1b	0b	No action	1b
Disabled	Undefined	Undefined	0b	0b	Optional: Set on 8b/10b Error	1b
Loopback	No action	Link speed may change on entry to Loopback from Configuration.	0b	0b	No action	1b
Hot Reset	No action	No action	0b	0b	Optional: Set on 8b/10b Error	1b

注意：

1. In-Band 是指：没有边带信号用于计算链路另一端上电设备的存在。
2. LinkUp：如果通过 Detect->Polling->Configuration，则 LinkUp 始终为 0，一直到链路初始化完成才为 1；如果从任何其他状态进入 Configuration 状态，则 LinkUp 始终为 1。

LTSSM 状态机转移图如 Figure 4-26 所示。

公众号·芯片业者相承双

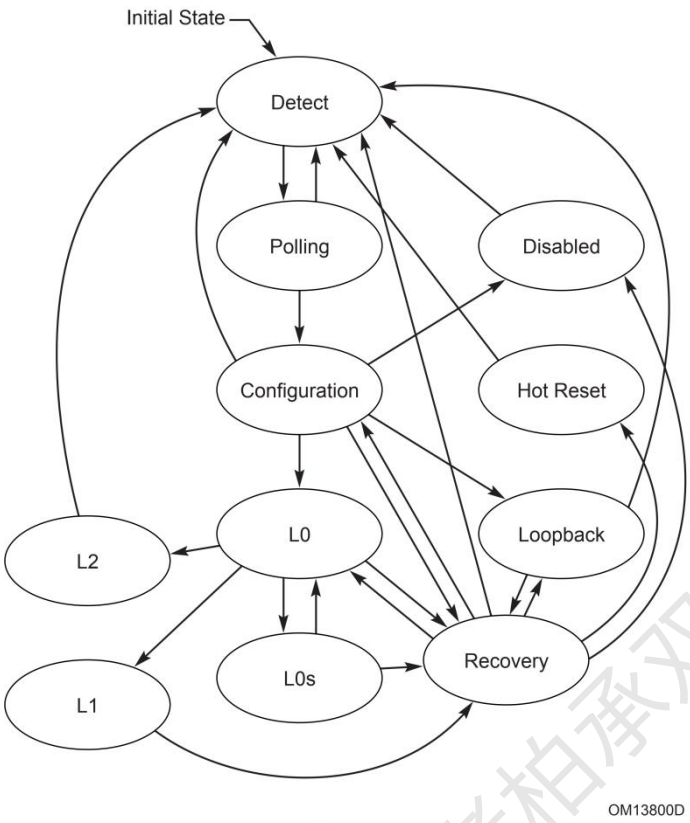


Figure 4-26 Main State Diagram for Link Training and Status State Machine

4.2.6.1 Detect

检测状态的目的是检测远端设备是否存在。Detect 子状态的状态机如 Figure 4-27 所示。

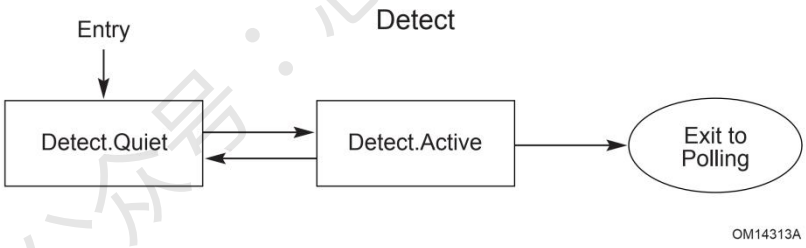


Figure 4-27 Detect Substate Machine

以下各节介绍 LTSSM 各个子状态的规则。

4.2.6.1.1 Detect.Quiet

- Detect.Quiet 子状态下，发送器应该处于电气空闲状态。
  - 此时 DC 共模电压不需要满足规范定义的 0~3.6V 的要求。
- 这个状态只支持 2.5GT/s 下的工作频率。如果从别的 LTSSM 状态跳入这个状态之前的工作频率不是数据速率为 2.5GT/s 下的工作频率，则 LTSSM 需要至少停在此状态 1ms，等工作频率切换为 2.5GT/s 下的频率。
  - Note：这不影响在 TS1 和 TS2 中通告的 data rate。
- 所有接收器都必须在进入此子状态后的 1 ms 内满足 2.5 GT/s 下的 Z_{GT-DC} 规范的要求（请参见 Table 8-10）。LTSSM 必须保持

公众号：芯片业者相承双

在此子状态，直到满足 2.5 GT/s 下的 Z_{RX-DC} 规范为止。

- 此状态下 LinkUp = 0b。这意味着不管从哪个状态跳转到该状态，都要 Link Down。
- 此状态下 Link Status 2 寄存器的 Equalization 8.0 GT/s Phase 1 Successful、Equalization 8.0 GT/s Phase 2 Successful、Equalization 8.0 GT/s Phase 3 Successful 和 Equalization 8.0 GT/s Complete 字段都为 0。16.0 GT/s Status 寄存器的 Equalization 16.0 GT/s Phase 1 Successful、 Equalization 16.0 GT/s Phase 2 Successful、 Equalization 16.0 GT/s Phase 3 Successful 和 Equalization 16.0 GT/s Complete 字段都为 0。 32.0 GT/s Status 寄存器的 Equalization 32.0 GT/s Phase 1 Successful, Equalization 32.0 GT/s Phase 2 Successful、Equalization 32.0 GT/s Phase 3 Successful 和 Equalization 32.0 GT/s Complete 字段都为 0。
- ***use_modified_TS1_TS2_Ordered_Set*** 变量复位为 0 。
- 变量 ***directed_speed_change*** 清 0； 变量 ***upconfigure_capable*** 清 0； 变量 ***idle_to_rlock_transitioned*** 清 00h； 变量 ***select_deemphasis*** 的值：对上游端口来说跟具体平台有关，对下游端口来说与 Link Status 2 寄存器的 Selectable Preset/De-emphasis 字段一致； 变量 ***equalization_done_8GT_data_rate*** 、 ***equalization_done_16GT_data_rate*** 、***equalization_done_32GT_data_rate*** 清 0。变量 ***perform_equalization_for_loopback*** 清 0。
  - 请注意，由于这些变量是 PCIe 2.0 规范定义的，因此 PCe 2.0 之前的设备将不会实现这些变量，并且将始终采用以下处理方式，即让 directed_speed_change 和 upconfigure_capable 变量始终为 0b，且 idle_to_rlock_transitioned 变量始终设置为 FFh。
- 异常情况下（比如设备未连接），LTSSM 需要在此状态停留 12ms，然后跳转到 Detect.Active 状态；正常情况下（比如设备连接上），当任意 Lane 检测到电气空闲退出就跳转到 Detect.Active 状态。

以下是 Synopsys PCIe IP 核在该状态下的代码，用于帮助我们理解该状态：

```
4198        S_DETECT_QUIET: begin
4199            if ( app_ltssm_enable_dd
4200                && (current_powerdown == `P1) // If there was a power state change, wait until phystatus is received
4201                && ((timeout_12ms && all_phystatus_deasserted) || (latchd_phystatus_fall && any_phy_mac_rxeidle_exit)) )
4202                next_lts_state   = S_DETECT_ACT;
4203            else
4204                next_lts_state   = S_DETECT_QUIET;
4205        end
```

```
4206        S_PRE_DETECT_QUIET: begin
4207            if (&xmtbyte_txelecidle
4208                && !(|latched_detect_speed)
4209                && !app_ltssm_enable_fall_edge && timeout_1us && !clear
4210                )
4211                next_lts_state   = S_DETECT_QUIET;
4212            else if (|latched_detect_speed && (current_data_rate == `GEN1_RATE) && timeout_1ms) //change speed to gen1
4213                next_lts_state   = S_DETECT_QUIET;
4214            else if ( timeout_48ms ) // This transition is not possible under normal conditions.
4215                next_lts_state   = S_DETECT_QUIET;
4216            else
4217                next_lts_state   = S_PRE_DETECT_QUIET;
4218        end
```

1. Synopsys PCIe IP 的状态机在 app_ltssm_enable 使能的情况下才能正常跳转；
2. 上述参数 P1 是 Synopsys PCIe IP 的 PHY 的状态（phy status），如下图：这给我们的参考意义是，在 P1 状态，设备已经正常上电完成，发送器的 pipeP_clk、pcs_clk 已经正常提供。但是注意，Receiver Detection 电路此时还未开始工作。

Table 3-3 Clock Availability

Clock	P0/P0s	P1	P2	Enable Required
phy_ref_output_clk	Y	Y	Y	macP_pclkreq_n = 2'b00
pcs_clk	Y	Y	N	macP_pclkreq_n = 2'b00 and pipeP_powerdown != 2'b11
pipeP_pclk	Y	Y	N	macP_pclkreq_n = 2'b00 and pipeP_powerdown != 2'b11
rx_clk	Y	N	N	macP_pclkreq_n = 2'b00 and pipeP_powerdown = 2'b00 or 2'b01

公众号：芯片业者相承及

4.2.6.1.2 Detect.Active

- 在 Detect.Active 状态，发送器在所有未配置的 Lane 上执行 Receiver Detection 过程，用于检测对端是否连接了一个设备（关于 Receiver Detection 电路原理的信息，参阅 Section 8.4.5.7）。
- 如果在所有已配置的 Lane 上检测到 Receiver 存在则 LTSSM 状态机跳转到 Polling 状态。
- 如果有任一个 Lane 上没有检测到 Receiver 则 LTSSM 状态机跳转回 Detect.Quiet 状态。
- 如果有的 Lane 检测到 Receiver 而有的 Lane 没有检测到，则需要：

1. 等 12ms。
2. 发送器需要重新在所有未配置的 Lane 上检测 Receiver，
   - 如果这次检测的结果与第一次检测的结果相同，状态机就切换到 Polling 状态。
     - 那些未检测到 Receiver 的 Lane 必须：
       1. 如果逻辑中实现了多个 LTSSM 状态机，则这些未检测到 Receiver 的 Lane 也可以交由别的 LTSSM 来检测，这是一种实现方法。
       2. 所有没有与可选的新的 LTSSM 关联上的 Lane 进入电气空闲状态。（注：不需要驱动共模电压来满足 L0 期间的 DC 直流共模和电气空闲（$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$）规范之间的绝对增量规范（请参见 Table 8-6）。
          (1) 在 LTSSM 转换回 Detect 之后，必须立即将这些 Lane 与此 LTSSM 重新关联。
          (2) 切换到电气空闲之前，无需发送 EIOS。
   - 否则状态机跳入 Detect.Quiet 状态。

以下是 Synopsys PCIe IP 核在该状态下的代码，用于帮助我们理解该状态：可以看出，它的实现还是很符合上述规则的。可以查看以下信息是否输出到一些状态寄存器，可用于调试该状态的完成情况。

```
4219        S_DETECT_ACT: begin
4220        // when all lanes detected receiver
4221        // or when the number of lanes detected receiver matched the
4222        // last time in this state
4223            if (int_rxdetect_done & ((all_phy_mac_rxdetected & ~retry_detect) | (any_phy_mac_rxdetected & retry_detect & same_detected_lanes)
4224                 | (cfg_auto_flip_en & cfg_auto_flip_using_predet_lane & int_rxdetected[0])
4225                ))
4226               next_lts_state   = S_POLL_ACTIVE;
4227            else if ((int_rxdetect_done & !any_phy_mac_rxdetected)
4228                    | (int_rxdetect_done & any_phy_mac_rxdetected & !same_detected_lanes & retry_detect)
4229                    | (int_rxdetect_done & cfg_auto_flip_en & cfg_auto_flip_using_predet_lane & !int_rxdetected[0])
4230                   )
4231               next_lts_state   = S_DETECT_QUIET;
4232            else if (int_rxdetect_done & any_phy_mac_rxdetected & !retry_detect)
4233               next_lts_state   = S_DETECT_WAIT;
4234            else
4235               next_lts_state   = S_DETECT_ACT;
4236        end
```

```
4237        S_DETECT_WAIT:
4238            if (timeout_12ms)
4239               next_lts_state   = S_DETECT_ACT;
4240            else
4241               next_lts_state   = S_DETECT_WAIT;
```

4.2.6.2 Polling

Polling 状态的子状态机如 Figure 4-28 所示。

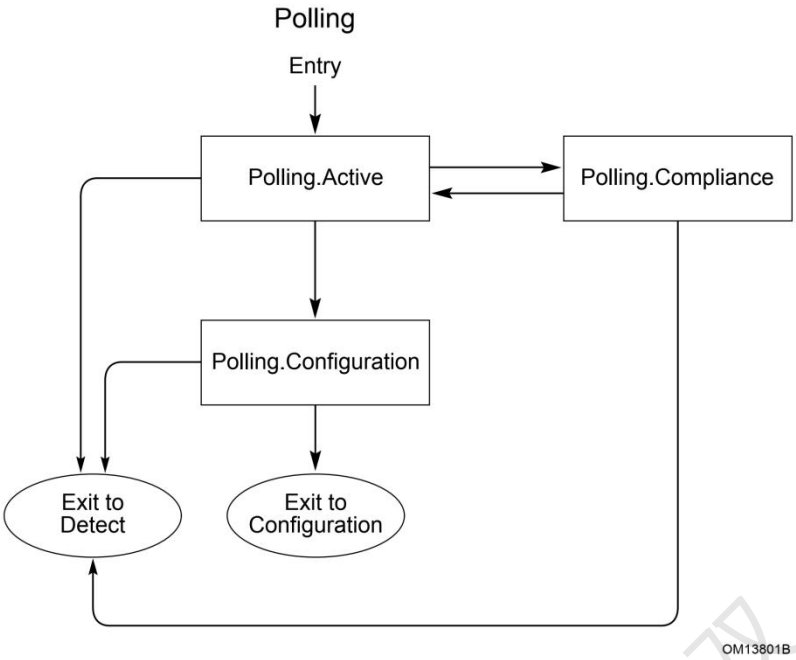


Figure 4-28 Polling Substate Machine

LTSSM 在 Polling 状态两侧设备交换 TS1 和 TS2 训练序列，用于完成比特锁定、符号锁定和配置通道极性。同时，还在 TS1 和 TS2 训练序列中携带了两端设备能支持的 data rate。

4.2.6.2.1 Polling.Active

Detect 状态完成之后（即设备检测完成），LTSSM 状态机就跳转到该状态。

- 在 Polling.Active 状态下，发送器在所有 Detect 期间检测到 Receiver 的 Lane 上发送 Lane number 和 Link number 设置为 PAD 的 TS1 Ordered Set。
  - 发送的 TS1 的 Data Rate Identifier 字段需要携带本设备能支持的 data rate。
  - 发送器在退出电气空闲和发送 TS1 序列之前设置好其 TX 的共模电压。
    - 发送器需要在进入此状态的 192ns 之内以满足 Transmit Margin（Link Control 2）字段定义的默认电压电平驱动数据。这个发送电压电平要一直持续到进入 Polling.Compliance 状态或 Recovery.RcvrLock 状态。
- 此状态下，如果把 Link Control 2 寄存器的 Enter Compliance 字段设置为 1，则状态机跳转到 Polling.Compliance 状态；如果在 Polling.Active 之前就把 Link Control 2 寄存器的 Enter Compliance 字段设置为 1，则发送器不发送 TS1 序列，直接进入 Polling.Compliance 状态（这应该也符合测试的一般流程）。
- 如果至少发送了 1024 个 TS1 并且所有检测到有 Receiver 存在的 Lane 都收到满足以下条件的 8 个连续的训练序列（TS1、TS2 或其补码），则 LTSSM 状态机就跳转到 Polling.Configuration 状态。
  - TS1 序列，其 Lane number 和 Link number 字段设置为 PAD，且 Compliance Receive 字段为 0b。
  - TS1 序列，其 Lane number 和 Link number 字段设置为 PAD，且 Loopback 字段为 0b。
  - TS2 序列，其 Lane number 和 Link number 字段设置为 PAD。
- 否则，在 24ms 超时后 LTSSM 状态机会按下述方式跳转。
  - 如果下面两个条件都满足就跳转到 Polling.Configuration 状态：

公众号：芯片业者相承双

- 任意在 Detect 期间检测到有 Receiver 存在的 Lane 上至少发送了 1024 个 TS1 并且收到满足以下条件的 8 个连续的训练序列。

  1. TS1 序列，其 Lane number 和 Link number 字段设置为 PAD，且 Compliance Receive 字段为 0b。
  2. TS1 序列，其 Lane number 和 Link number 字段设置为 PAD，且 Loopback 字段为 0b。
  3. TS2 序列，其 Lane number 和 Link number 字段设置为 PAD。

- 从进入 Polling.Active 状态起，至少在一组 predetermined set of Lanes 上至少都检测到一次从电气空闲状态退出。

  - Note：设置这个条件的目的是让发送器和接收器不会因为一个或多个坏的 Lane 而导致状态机无法正常运行。这使得状态机可以继续在 Polling.Configuration 中进行其他训练。这个功能是特定于实现的。请注意，直到 1.1 规范之前，predetermined set of Lanes 的数目还是等于检测到 Receiver 的通道的总数。
  - Note：从进入 Polling.Active 状态起，收到八个连续的 TS1 或 TS2 有序集的任何 Lane 都应至少检测到一次从电气空闲退出。

- 如果满足下面两个条件任意一个就跳转到 Polling.Compliance 状态：

  - 从进入 Polling.Active 状态起，之前定义好的 predetermined set of Lanes 中不是所有 Lane 都检测到电气空闲退出，就会跳转到 Polling.Compliance 状态。
  - 这些 Lane 中任意一条 Lane 上收到了 8 个连续的 Lane number 和 Link number 字段设置为 PAD，Compliance Receive 字段为 1b，且 Loopback 字段设置为 0b 的 TS1 序列，就跳转到 Polling.Compliance 状态。

  Note：如果在所有 Lane 上都连接一个被动测试负载，则设备将进入 Polling.Compliance。

- 如果 Polling.Configuration 和 Polling.Compliance 的条件都没满足就跳转到 Detect 状态。

这里有个术语 predetermined set of Lanes（预定通道组）：它有这样的背景，比如说 x8 链路上有 5 条 Lane 是好的，但是我们却不能用这 5 条 Lane 来工作，只能使用的模式为 x4、x2 和 x1，所以需要提前配置好要使用哪些 Lane，丢弃哪些 Lane，predetermined set of Lanes 的目的就在此。

以下给出了 Synopsys PCIe IP 在该状态下的代码，用于帮助理解。

从波形上看 Polling.Active 状态交换的 TS1 序列为：

RX 侧：TS1(BC->F7->F7->FF->0E(rate)->00->4a->4a->4a->4a->4a->4a->4a->4a->4a->4a)

TX 侧：TS1(BC->F7->F7->80->0E(rate)->00->4a->4a->4a->4a->4a->4a->4a->4a->4a->4a)

```
S_POLL_ACTIVE: begin
    if (current_powerdown == 'P0)
        if (cfg_enter_compliance)
            next_lts_state = S_POLL_COMPLIANCE;
        else
        if ( link_latched_live_all_8_ts_plinkn_planen_rcvd
             && (link_latched_live_all_8_ts1_plinkn_planen_compl_rcv_0_rcvd || link_latched_live_all_8_ts1_plinkn_planen_lpbk1_rcvd || link_latched_live_all_8_ts2_plinkn_planen_rcvd)
             && xmtbyte_1024_ts_sent )
        // 8 ts are rcvd with pad link/lane# on all lanes
            next_lts_state = S_POLL_CONFIG;
        else if (timeout_24ms) begin
            if ( link_latched_live_any_8_ts_plinkn_planen_rcvd && link_xmlh_1024_ts1_sent_after_any_1_ts_rcvd && all_predet_lane_latchd_rxeidle_exit //ts_rcvd_1024_sent: 1024 TSs sent after receiving one TS
                 && (link_latched_live_any_8_ts1_plinkn_planen_compl_rcv_0_rcvd || link_latched_live_any_8_ts1_plinkn_planen_lpbk1_rcvd || link_latched_live_any_8_ts2_plinkn_planen_rcvd) //don't check comp
               )
                next_lts_state = S_POLL_CONFIG;
            else if ( link_latched_live_any_8_ts1_plinkn_planen_compl_rcv_1_lpbk0_rcvd && ~clear ) begin
                next_lts_state = S_POLL_COMPLIANCE;
                int_rcvd_8expect_ts1 = 1'b1;
            end
            else if ( any_predet_lane_latched_rxeidle )
                // not all predet lanes exit rxelecidle
                next_lts_state = S_POLL_COMPLIANCE;
            else
                next_lts_state = S_PRE_DETECT_QUIET;
        end else // !timeout_24ms
            next_lts_state = S_POLL_ACTIVE;
    else // !current_powerdown == 'P0
        next_lts_state = S_POLL_ACTIVE;
```

4.2.6.2.2 Polling.Compliance

- Link Control 2 寄存器的 Transmit Margin 字段为 1 就会进入此状态，并且在 192ns 内作用到发送侧封装的 PIN 上。这个字段需要在 LTSSM 状态机处于此状态的整个过程中都有效。
- 使用以下算法在从 Polling.Active 到 Polling.Compliance 的转换中确定用于传输的一致性码型的数据速率和去加重级别。
  - 如果端口仅能够以 2.5 GT/s 的数据速率进行传输，则用于传输的一致性码型的数据速率为 2.5 GT/s，去加重级别为-3.5 dB。
  - 否则，如果端口由于在 Polling.Active 中检测到八个连续的 Compliance Receive 字段为 1 的 TS1 有序集而进入 Polling.Compliance 状态，则数据速率为交换的 TS1 序列集中携带的 Data Rate Identifiers 字段支持的公共数据速率中最高的速率。select_deemphasis 变量必须 TS1 有序集中的 Selectable De-emphasis 字段的值。如果公共数据速率为 8.0 GT/s 或更高，则将每个通道上的 select_preset 变量设置为 TS1 有序集的 Symbol 6 中通告的发送器的 preset 值，并且该值必须由发送器使用（使用接收器的 TS1 中通告的 preset 值是可选实现的）。如果公共数据速率为 8.0 GT/s 或更高，则任何未接收到带有发送器 preset 值的 TS1 有序集的通道可以选择使用任何受支持的发送器 preset 值。
  - Link Control 2 寄存器的 Transmit Margin 字段为 1，则使用 Link Control 2 的 Target Link Speed 字段支持的值作为一致性码型的数据速率。如果数据速率为 5GT/s，则 select_deemphasis 变量等于 Link Control 2 的 Compliance Preset/De-emphasis 字段的值等于 0001b。如果数据速率为 8GT/s 或更高，则 select_deemphasis 变量等于 Link Control 2 的 Compliance Preset/De-emphasis 字段的值，这也是此时发送器的工作状态。
  - 其他情况（应该指的就是接测试负载的场景吧），data rate、preset 和 de-emphasis 级别的定义如下，它取决于组件支持的最大数据速率和在此条件下进入 Polling.Compliance 的次数，如 Table 4-21 描述：

Table 4-21 Compliance Pattern Settings

Setting Nos	Data Rate	Transmitter De-emphasis or preset sequence
#1	2.5 GT/s	-3.5 dB
#2, #3	5.0 GT/s	-3.5 dB followed by -6 dB
#4 through #14	8.0 GT/s	Transmitter Preset Encoding 0000b through 1010b, as defined in Section 4.2.3.2 , in increasing order
#15 through #25	16.0 GT/s	Transmitter Preset Encoding 0000b through 1010b, as defined in Section 4.2.3.2 , in increasing order
#26 through #34	16.0 GT/s	Transmitter Preset Encoding 0100b as defined in Section 4.2.3.2
#35 through #45	32.0 GT/s	Transmitter Preset Encoding 0000b through 1010b, as defined in Section 4.2.3.2 , in increasing order
#46 through #54	32.0 GT/s	Transmitter Preset Encoding 0100b as defined in Section 4.2.3.2

随后按上述顺序重复进入 Polling.Compliance 状态。例如，导致端口以 5.0 GT/s 的数据速率和-6 dB 的去加重级别传输一致性码型的过程为：Polling.Active，Polling.Compliance（2.5 GT/s 和- 3.5 dB），Polling.Active，Polling.Compliance（5.0 GT/s 和-3.5 dB），Polling.Active，Polling.Compliance（5.0 GT/s 和-6 dB）。即逐个试上面的 Setting，找到一个合适的设置，一致性测试负载也是按照这个思路执行测试的。

如果端口支持 16.0 GT/s 或更高的数据速率，或当端口以 8.0 GT/s 或更高的数据速率（请参见 Table 8-10）工作时其接收器不满足 2.5 GT/s 的 Z_{RX-DC} 规范，则必须在 Polling.Configuration 状态下将该序列设置为 Setting #1。允许所有端口在

公众号：芯片业务相关

Polling.Configuration 状态下将序列设置为 Setting # 1。

IMPLEMENTATION NOTE

Compliance Load Board Usage to Generate Compliance Patterns

可以设想，一致性负载（基）板可以在差分对的一个支路上以 350mv 峰峰值发送约 1ms 的 100mhz 信号，以使设备达到到期望的速度和去加重水平。被测设备需要根据其支持的最大数据速率，依次在以下设置中循环，以便每次从 Polling.Active 进入 Polling.Compliance。从第一个进入 Polling.Compliance 的条目的第一个设置开始如 Table 4-21 中定义的基本复位所示。

- 如果一致性码型的数据速率不是 2.5 GT/s，并且在进入 Polling.Compliance 之前在 Polling.Active 中发送了 TS1 有序集，则发送器在进入电气空闲之前会发送一个 EIOS 或两个连续的 EIOS。如果一致性码型的数据速率不是 2.5 GT/s，并且在进入 Polling.Compliance 之前未在 Polling.Active 中发送 TS1 有序集，则发送器必须进入电气空闲状态，而不发送任何 EIOS。在电气空闲期间，数据速率将更改为新的速率并保持稳定。在操作频率为 5.0 GT/s 的情况下，则如果 select_deemphasis 变量为 1b，则 de-emphasis/preset 级别必须设置为-3.5 dB，否则必须将其设置为-6 dB。如果操作频率为 8.0 GT/s，则必须将发送器的 preset 值设置为 select_preset 变量中的值。电气空闲周期大于 1 ms，但不超过 2 ms。

- data rate 和 de-emphasis/preset 级别确定之后，在 Polling Compliance 期间的行为要符合以下规则：

  - 如果由于发送 TS1 序列或设置 Link Control 2 的 Enter Compliance 和 Enter Modified Compliance 字段导致状态机进入 Polling.Compliance 状态，则发送器就要以上述定义好的速率在所有检测到有 Receiver 存在的 Lane 上发送 error status Symbol 设置为 0 的 Modified Compliance Pattern（Section 4.2.9）。

    - 如果数据速率是 2.5 GT/s 或 5.0 GT/s，则特定的通道接收器通过查找任何出现的 Modified Compliance Pattern，然后将其自己的 Modified Compliance Pattern 的 error status Symbol 的 Pattern Lock（第 1 位）置为 1 来表示成功锁定输入的 Modified Compliance Pattern。

      - error status Symbols 不可在锁定过程中使用，因为它们在任何给定时刻均未定义。

      - 将上面出现的 Modified Compliance Pattern 定义为以下 8b/10b 符号序列：K28.5，D21.5，K28.5 和 D10.2 或每个单独符号的补码。

      - 被测设备必须在其接收器封装引脚上收到 Modified Compliance Pattern 后 1ms 内成功锁定到输入的 Modified Compliance Pattern，必须将其在发送器封装引脚上发送的 Modified Compliance Pattern 的 Pattern Lock 设置为 1。

    - 如果数据速率为 8.0 GT/s 或更高：在进入此子状态时，Error_Status 字段设置为 00h。如 Section 4.2.2.2.1 所述，每个通道在实现块对齐时都独立设置 Pattern Lock 字段。Pattern Lock 之后，将数据块中接收到的符号与空闲数据符号（00h）进行比较，并且每个不匹配的符号都会使 Receiver Error Count 字段增加 1。Receiver Error Count 的饱和值为 127（其他不匹配的符号不会更改 Receiver Error Count）。每个通道的 Pattern Lock 和 Receiver Error Count 信息作为在该通道的 Modified Compliance Pattern 中传输的 SKP 有序集的一部分进行传输。有关更多信息，请参见 Section 4.2.7。被测设备必须在其接收器封装引脚上收到 Modified Compliance Pattern 后的 4 ms 内，将其发送的 SKP 有序集中的 Pattern Lock 设置为 1。

    在 Section 4.2.2.4 中定义的加扰可以用于收到的 Modified Compliance Pattern。例如，为每 个通道设置加扰的 LFSR 种子，在 EIEOS 初始化 LFSR，而 SKP 有序集不推进 LFSR。

- Section 4.2.2.4 中定义的加扰要求适用于收到的 Modified Compliance Pattern。例如，为每个 Lane 设置加扰的 LFSR 种子，EIEOS 初始化 LFSR，而 SKP 有序集不推进 LFSR。

- 如果数据速率是 2.5 GT/s 或 5.0 GT/s，则一旦特定通道指示它已锁定到输入的 Modified Compliance Pattern，则每次发生接收器错误时，该通道的 Receiver Error Count 都会增加。

公众号·芯片业者相关资

- error status Symbols 使用低 7 位作为 Receiver Error Count 字段，如果计数达到 127，则该字段将始终停留在全 1。
- 如果使用 8b/10b 编码，则接收器不得对处于此子状态时将接收的 10 位模式做出任何措施。

- 如果 Link Control 2 寄存器中的 Enter Compliance 为 0b，则下一个状态定向到 Detect 。
- 否则，如果在进入 Polling.Compliance 时将 Enter Compliance 设置为 1b，则在满足以下任何条件的情况下，下一个状态为 Polling.Active：
  - Link Control 2 寄存器中的 Enter Compliance 位更改为 0b。
  - 该端口是上游端口，并且在任何通道上都接收到 EIOS。当此条件为真时，Enter Compliance 位将重置为 0b。

  如果发送器以 2.5 GT/s 以外的数据速率发送数据，或者在进入 Polling.Compliance 时将 Link Control 2 寄存器中的 Enter Compliance 位设置为 1b，则发送器将发送八个连续的 EIOS 并在进入进入电空闲状态之前先转换到 Polling.Active。 在电气空闲期间，数据速率更改为 2.5 GT/s 并稳定下来，去加重级别设置为-3.5 dB。电气空闲周期大于 1 ms，但不得超过 2 ms。
- Note：发送多个 EIOS 可以提供足够的鲁棒性，以便在使用配置寄存器机制进入时，另一个端口检测到至少一个 EIOS 并退出 Polling.Compliance 子状态。
- 否则，如果由于 Link Control 2 寄存器的 Enter Compliance 设置为 1b 并且 Link Control 2 寄存器的 Enter Modified Compliance 设置为 0b，从而导致端口进入了 Polling.Compliance，则：

  a. 发送器以上面定义的 data rate 和 de-emphasis/preset 等级在所有检测到有 Receiver 存在的 Lane 上发送一致性码型。

  b. 如果下面任意条件满足，则状态机跳转到 Polling.Active：

     1. 在进入 Polling.Compliance 之前 Link Control 2 的 Enter Compliance 字段从 1b 变为 0b。
     2. 该端口是一个上游端口，Link Control 2 的 Enter Compliance 字段为 1b，并且已在任何通道上检测到 EIOS。当此条件为真时，Enter Compliance 位重置为 0b。

     发送器发送八个连续的 EIOS，并在切换到 Polling.Active 之前进入电空闲状态。在电气空闲期间，数据速率更改为 2.5 GT/s 并保持稳定。电气空闲时间段大于 1 ms，但不得超过 2 ms。

     注意：发送多个 EIOS 可以提供足够的鲁棒性，以便在使用配置寄存器机制进入时，另一个端口检测到至少一个 EIOS 并退出 Polling 。
- 否则，

  a. 发送器在 Detect 期间以上面确定的数据速率和去加重/预设级别在检测到有接收器存在的通道上发送以下码型：

     - For Settings #1 to #25, and #35 to #45: Compliance pattern on all Lanes.
     - For Setting #26, #46: Jitter Measurement Pattern on all Lanes.
     - For Setting #27, #47: Jitter Measurement Pattern on Lanes 0/8/16/24 and Compliance pattern on all other Lanes.
     - For Setting #28, #48: Jitter Measurement Pattern on Lanes 1/9/17/25 and Compliance pattern on all other Lanes.
     - For Setting #29, #49: Jitter Measurement Pattern on Lanes 2/10/18/26 and Compliance pattern on all other Lanes.
     - For Setting #30, #50: Jitter Measurement Pattern on Lanes 3/11/19/27 and Compliance patter on all other Lanes.
     - For Setting #31, #51: Jitter Measurement Pattern on Lanes 4/12/20/28 and Compliance pattern on all other Lanes.
     - For Setting #32, #52: Jitter Measurement Pattern on Lanes 5/13/21/29 and Compliance pattern on all other Lanes.

公众号·芯片业者相承双

- For Setting #33, #53: Jitter Measurement Pattern on Lanes 6/14/22/30 and Compliance pattern on all other Lanes.
- For Setting #34, #54: Jitter Measurement Pattern on Lanes 7/15/23/31 and Compliance pattern on all other Lanes.

b. 如果在 Detect 期间检测到有接收器存在的任何通道的接收器检测到电气空闲退出，则下一个状态为 Polling.Active。如果发送器以 2.5 GT/s 以外的数据速率发送，则发送器将发送八个连续的 EIOS，并进入电空闲状态，然后再转换为 Polling.Active。在电气空闲期间，数据速率更改为 2.5 GT/s 并保持稳定。电气空闲时间段大于 1 ms，但不得超过 2 ms。

```
4272        S_POLL_COMPLIANCE: begin
4273            if (curnt_compliance_state == S_COMPL_EXIT)
4274                next_lts_state = S_POLL_ACTIVE;
4275            // move the ltssm state from Polling.Compliance to Detect
4276            else if (cdm_ras_des_direct_poll_cmp_to_det)
4277                next_lts_state = S_PRE_DETECT_QUIET;
4278            else
4279                next_lts_state = S_POLL_COMPLIANCE;
4280        end
```

4.2.6.2.3 Polling.Configuration

- 如果有需要，接收器在这个状态完成通道极性翻转（Section 4.2.4.5）。
- 进入该状态后，必须将 Link Control 2 寄存器的 Transmit Margin 字段置为 000b。
- 如果有需要，按 Section 4.2.6.2.2 的描述更新发送器的 Polling.Compliance 过程的配置。
- 发送器会在所有检测到有 Receiver 存在的 Lane 上发送 Link number 和 Lane number 为 PAD 的 TS2 序列集。
  - 发送的 TS2 的 Data Rate Identifier 字段需要携带本设备能支持的 data rate。
- 如果在任意检测到有 Receiver 存在的的 Lane 上收到 8 个 Link number 和 Lane number 为 PAD 的 TS2 有序集，且在收到一个这样的 TS2 之后发送完 16 个 TS2 有序集，则状态机跳转到 Configuration 状态。
- 否则，在经过 48ms 超时后状态机跳转到 Detect 状态。

以下给出了 Synopsys PCIe IP 在该状态下的代码，用于帮助理解。

从波形上看 Polling.Configuration 状态交换的序列为：

RX 侧：

TS1(BC->F7->F7->FF->0E(rate)->00->4a->4a->4a->4a->4a->4a->4a->4a->4a->4a)

TS2(BC->F7->F7->FF->0E(rate)->00->45->45->45->45->45->45->45->45->45->45)

TX 侧：

TS1(BC->F7->F7->80->0E(rate)->00->4a->4a->4a->4a->4a->4a->4a->4a->4a->4a)

TS2(BC->F7->F7->80->0E(rate)->00->45->45->45->45->45->45->45->45->45->45)

```
4281        S_POLL_CONFIG: begin
4282            //use latched signal because the other end may move to Configuration and start to send ts1s
4283            //while the near end is still sending 16 ts2s.
4284            if (link_latched_live_any_8_ts2_plinkn_planen_rcvd && link_xmlh_16_ts2_sent_after_1_ts2_rcvd)
4285                next_lts_state = S_CFG_LINKWD_START;
4286            else if (timeout_48ms)
4287                next_lts_state = S_PRE_DETECT_QUIET;
4288            else
4289                next_lts_state = S_POLL_CONFIG;
4290        end
```

公众号·芯片业者相承双

4.2.6.2.4 Polling.Speed

如果设备仅在 2.5GT/s 下工作，则此状态永远不会达到；这个状态本来的目的是用于切换到 Recovery 状态改变 data rate。

IMPLEMENTATION NOTE

Support for Higher Data Rates than 2.5 GT/s

链路最初将以 2.5 GT/s 的数据速率训练到 L0 状态，即使双方都能够以大于 2.5 GT/s 的数据速率进行操作。 支持的更高数据速率在 TS 有序集中通告。 对方的速度能力是在 Configuration.Complete 子状态期间注册的。 基于最高支持的共用数据速率，任何一方都可以通过切换到 Recovery 状态来启动从 L0 状态开始的速度切换。

Synopsys 在实现时已经没有这个状态了。

4.2.6.3 Configuration

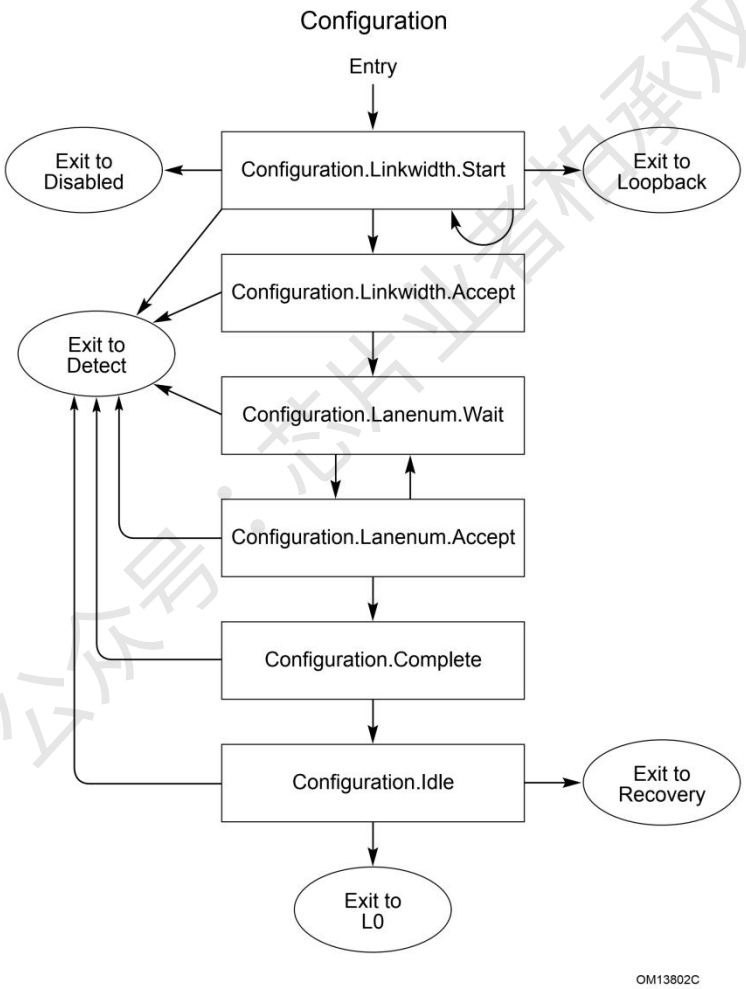


Figure 4-29 Configuration Substate Machine

这个状态完成链路速率协商、链路宽度协商、通道编号以及 lane-to-lane de-skew。RC（DSP）设备发起协商过程。

4.2.6.3.1 Configuration.Linkwidth.Start

这里先给出 Synopsys PCIe IP 里 Configuration.Linkwidth.Start 状态的代码。

```
4291        S_CFG_LINKWD_START: begin
4292            if ( ~cfg_upstream_port ) begin // DSP
4293                if ( cfg_link_dis ) begin
4294                    next_lts_state = S_DISABLED_ENTRY;
4295                end else if ( cfg_lpbk_en ) begin
4296                    next_lts_state   = S_LPBK_ENTRY;
4297                end else if ( link_latched_live_all_2_ts1_lpbk1_rcvd ) begin
4298                    next_lts_state = S_LPBK_ENTRY;
4299                end else if ( link_any_2_ts1_linknmtx_planen_rcvd ) begin
4300                    next_lts_state = S_CFG_LINKWD_ACEPT;
4301                end else if ( timeout_32ms ) begin //32ms < 24ms + 50% margin
4302                    next_lts_state = S_PRE_DETECT_QUIET;
4303                end else begin
4304                    next_lts_state = S_CFG_LINKWD_START;
4305                end
4306            end else begin // USP
4307                if ( cfg_lpbk_en ) begin
4308                    next_lts_state   = S_LPBK_ENTRY;
4309                end else if ( link_any_2_ts1_dis1_rcvd ) begin
4310                    next_lts_state = S_DISABLED_ENTRY;
4311                end else if ( link_latched_live_all_2_ts1_lpbk1_rcvd ) begin
4312                    next_lts_state = S_LPBK_ENTRY;
4313                end else if (
4314                            (link_latched_live_all_2_ts1_linkn_planen_rcvd && latched_ts1_sent ) ) begin // ensure at least 1 ts1 sent
4315                    next_lts_state = S_CFG_LINKWD_ACEPT;
4316                end else if ( timeout_32ms ) begin //32ms < 24ms + 50% margin
4317                    next_lts_state = S_PRE_DETECT_QUIET;
4318                end else begin
4319                    next_lts_state = S_CFG_LINKWD_START;
4320                end
4321            end
4322        end
```

从波形上看：其实就是一个交换 TS1 的过程，其中设置了 Link number。

RX 侧：TS1(BC->00（LinkN）->F7->FF->0E->00->4a->4a->4a->4a->4a->4a->4a->4a->4a->4a)

TX 侧：TS1(BC->00->F7->80->0E->00->45->45->45->45->45->45->45->45->45->45)

可以看到，正常情况下，RC 在所有通道上发送两个链路号为 8'h00 的 TS1 序列和通道号为 PAD 字符的序列集给 EP 表示开始进行链路号协商，EP 会返回链路号为 8'h00 通道号为 PAD 的 TS1 序列集响应，一旦这个过程完成就会进入 Configuration.Linkwidth.Accept 状态。

要深入理解一下 Link number 的意义，以便跟 Lane number 区别：链路号可以理解为与端口数目有关，一条 x8 链路如果分成两个 x4，那它可以挂两个 x4 设备，此时 Link number 就应该有两个，假设可以编号为 0 和 1；如果一条 x8 链路如果分成 4 个 x2，那它可以挂 4 个 x2 设备，此时 Link number 就应该有 4 个，假设可以编号为 0、1、2 和 3；由此 Link number 其实只在支持多 Port 的情况下起作用。

4.2.6.3.1.1 Downstream Lanes

- 在 Configuration.Linkwidth.Start 子状态，DSP 的上层逻辑可以通过在所有检测到 Receiver 的 Lane 上发送 Disable Link 字段设置为 1 的 TS1 和 TS2 有序集让 LTSSM 状态机切换到 Disabled 状态。
- 在 Configuration.Linkwidth.Start 子状态，端口的上层逻辑可以通过在所有检测到 Receiver 的 Lane 上发送 Loopback 字段设置为 1 的 TS1 和 TS2 有序集让 LTSSM 状态机切换到 Loopback 状态，此时发送器作为 Loopback Master。
- 如果支持 crosslink，在所有传输 TS1 有序集的通道接收到两个连续的 TS1 有序集且其 Disable Link 字段被声明为 1，则下一个状态为 Disabled 状态。
- 如果下述其中一个条件满足，则下个状态为 Loopback 状态：
  - 发送 TS1 有序集的所有 Lane 也在接收 TS1 有序集，收到的两个连续的 TS1 有序集的 Loopback 字段被置 1b。
  - 任意发送 TS1 有序集的 Lane 收到了两个连续的 Loopback 字段置 1b 的 TS1 有序集且 Enhanced Link Behavior Control 字段被设置为 1b 。
  - 请注意，接收到 Loopback 字段置 1 的有序集的设备将成为 Loopback Slave 。
- 如果 LinkUp 为 0b 或 LTSSM 未启动链路宽度的配置，则发送器在所有有效的 Downstream Lanes 上发送 Link number 设置为

某值且 Lane number 设置为 PAD 的 TS1 有序集。此外，如果 upconfigure_capable 为 1b 且 LTSSM 未启动链路宽度的配置，则 LTSSM 在进入 Recovery 检测到电气空闲退出且连续收到两个 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集之后，在所有无效的 Lanes 上发送 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集。

- 从 Polling 转换到此子状态时，在 Detect 站柜台检测到接收器的任何通道均被视为有效通道。
- 从 Recovery 转换到此子状态时，在之前的 Configuration.Complete 完成配置的通道被视为有效通道。
- TS1 有序集的 Data Rate Identifier 字段必须通告端口支持的所有数据速率，包括它不打算使用的数据速率。

- 如果 LinkUp 为 1b 且 LTSSM 启动了链路宽度的配置，则一开始它会在当前有效的一组 Lane 上发送 Link number 和 Lane number 都设置为 PAD 的 TS1 有序集；然后无效的 Lane 会变成有效的 Lane；然后这些 Lane 会在进入 Recovery 检测到电气空闲退出且连续收到两个 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集。LTSSM 会在发送了 TS1 序列集又收到两个 Link number 和 Lane number 都设置为 PAD 的 TS1 有序集或 1ms 超时的 Lanes 上发送 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集。
  - 在任何无效的通道变为有效后，发送器必须等待其 TX 共模电压建立完成，然后才能退出电空闲并发送 TS1 有序集。
  - 对于仅能够作为唯一链路的通道组，Link number 是可以不同的。
  - Note：Link number 分配的一个示例是有八个 Downstream Lane 的链路，当连接到一个组件时，这些通道可以协商成一个 x8 端口，而当连接到两个不同组件时，则可以协商成两个 x4 端口。Downstream Lane 在其中四个 Lane 上发送 Link number 为 N 的 TS1 有序集，而在其他四个 Lane 上发送 Link number 为 N+1 的 TS1 有序集。
- 如果有任何 Lane 首先收到至少一个或多个 Link number 和 Lane number 都设置为 PAD 的 TS1 有序集，则如果接下来在这些 Downstream Lanes 上又收到了两个连续的 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集，且此 Link number 与其之前发送的一致， 则状态机立马切换到 Configuration.Linkwidth.Accept 状态。
  - 如果不支持交叉链路，则首先 Link number 和 Lane number 都设置为 PAD 的 TS1 有序集的条件始终为 true。
- 否则：如果 LinkUp 为 0b 并且支持交叉链路，则所有检测到 Receiver 的 Downstream Lane 先发送 16-32 个 Link number 为 non-PAD 且 Lane number 为 PAD 的 TS1 序列集；然后，如果上述任一个 Downstream Lane 收到了两个连续的 Link number 不是 PAD 且 Lane number 为 PAD 的 TS1 序列集，就把 Downstream Lane 指定为 Upstream Lane 并且指定一个随机的交叉链路超时时间（参看 Table 8-7 的 $T_{crosslink}$）。同样，状态机也跳转为 Upstream Lane 的 Configuration.Linkwidth.Start。
  - Note：支持交叉链接（crosslink）指的是双方都可以尝试充当 Downstream Port。这个解决方案是使两个 Port 都作为 Upstream Port，然后设置一个随机超时时间，直到链路的一侧变为 Downstream Port，而另一侧仍为 Upstream Port。即使连接两个相同的设备，此超时也必须是随机的，以最终打破任何可能的死锁。
  - 如果支持交叉链路，则只有在未因接收 TS2 有序集而中断的情况下，接收序列为 TS1 的有序集（其 Link number 为 PAD，后跟与发送的 Link number 相匹配的非 PAD 的 Link number）才有效。

IMPLEMENTATION NOTE

Crosslink Initialization

如果下游通道同时连接到下游通道（交叉链路）和上游通道，则具有下游通道的端口可以继续使用本节中所述的单个 LTSSM，也可以分为多个 LTSSM。

- 如果上述规则都没满足，则在 24ms 超时后状态机跳转到 Detect 状态。

4.2.6.3.1.2 Upstream Lanes

- 在 Configuration.Linkwidth.Start 子状态，如果支持 crosslink Port，则上层逻辑可以把 TS1 序列中把 Disable Link 字段设置为 1，

公众号：芯片业者相关双

下个状态切换到 Disabled。

- 在 Configuration.Linkwidth.Start 子状态，端口的上层逻辑可以通过把 TS1 和 TS2 序列中的 Loopback 字段设置为 1 让 LTSSM 状态机切换到 Loopback 状态，此时发送器作为 Loopback Master。
- 在发送 TS1 序列集后又接收到两个连续的 Disable Link 字段设置为 1 的 TS1 序列集的任意 Lane 后，下个状态切换到 Disabled 状态。
  - 如果支持 crosslink，只有在所有正在发送 TS1 有序集的通道接收到两个连续的 Disable Link 字段设置为 1 的 TS1 有序集后，下一个状态切换到 Disabled 状态。
- 如果下述其中一个条件满足，则下个状态为 Loopback 状态：
  - 发送 TS1 有序集的所有 Lane 也在接收 TS1 有序集，收到的两个连续的 TS1 有序集的 Loopback 字段被置 1b。
  - 任意发送 TS1 有序集的 Lane 收到了两个连续的 Loopback 字段置 1b 的 TS1 有序集且 Enhanced Link Behavior Control 字段被设置为 1b 。
  - 请注意，接收到 Loopback 字段置 1 的有序集的设备将成为 Loopback Slave 。
- 发送器在所有有效的 Upstream Lanes 上发送 Link number 和 Lane number 设置为 PAD 的 TS1 有序集；然后无效的 Lane 开始配置其链路宽度；并且如果 upconfigure_capable 设置为 1b，则这些无效的 Lane 中的每一个都会在进入 Recovery 状态检测到从电气空闲退出，并且随后在此子状态下接收到两个 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
  - 从 Polling 转换到此子状态时，在 Detect 站柜台检测到接收器的任何通道均被视为有效通道。
  - 从 Recovery 转换到此子状态时，在之前的 Configuration.Complete 完成配置的通道被视为有效通道。
  - 从 Recovery 转换到此子状态时，如果转换不是由 LTSSM 超时引起的，如果发送器因自主原因更改链路宽度，则发送器必须将其在 Configuration 状态下发送的 TS1 有序集中的 Autonomous Change 字段设置为 1b。
  - TS1 有序集的 Data Rate Identifier 字段必须通告端口支持的所有数据速率，包括它不打算使用的数据速率。
- 如果任一 Lane 接收到两个连续的 Link number 设置为 non-PAD 且 Lane number 设置为 PAD 的 TS1 有序集，就在所有检测到 Receiver 的 Lane 上发送 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集，而在剩余 Lane 上必须发送 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。如果满足下述条件中任意一个，则状态机跳转到 Configuration.Linkwidth.Accept：
  - 如果 LTSSM 正在启动链路宽度配置，则它将在所有想要由无效变为有效的 Lane 上等待直到接收到两个连续的 Link number 设置为 non-PAD 且 Lane number 设置为 PAD 的 TS1 有序集；或者在进入此子状态 1ms 后，在发送 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集之前任意 Lane 上收到两个连续的 Link number 设置为 non-PAD 且 Lane number 设置为 PAD 的 TS1 有序集。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。
    - 激活任何无效的通道后，发送器必须等待其 TX 共模电压建立，然后退出电空闲并发送 TS1 有序集。
- 可选的，如果 LinkUp 为 0b 并且支持交叉链路，则检测到 Receiver 的所有 Upstream Lanes 必须首先发送 16-32 个 Link number 和 Lane number 设置为 PAD 的 TS1 有序集，然后如果任意上述 Lane 首先收到了两个连续的 Link number 和 Lane number 设置为 PAD 的 TS1 有序集，然后：
  - 发送器继续发送 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。

公众号：芯片业者相关权

- 如果任一 Lane 接收到两个连续的 Link number 设置为 non-PAD 且 Lane number 设置为 PAD 的 TS1 有序集，就在所有检测到 Receiver 的 Lane 上发送 Link number 设置为某值且 Lane number 设置为 PAD 的 TS1 有序集，而在剩余 Lane 上必须发送 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。之后状态机跳转到 Configuration.Linkwidth.Accept。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。
- 否则，在 Tcrosslink 超时后，将发送 16-32 个 Link number 和 Lane number 设置为 PAD 的 TS2 有序集。上游通道变为下游通道，且下一个状态为下游通道的 Configuration.Linkwidth.Start。
  - Note:对于交叉链路来说，此可选行为是必需的，在这种行为中，两个端口可能从上游端口开始，而其中一个最终将会成为下游端口。

- 如果上述规则都没满足，则在 24ms 超时后状态机跳转到 Detect 状态。

4.2.6.3.2 Configuration.Linkwidth.Accept

```
4323        S_CFG_LINKWD_ACCEPT: begin
4324            if ( ~cfg_upstream_port && link_latched_live_all_2_ts1_linknmtx_rcvd ) begin
4325                next_lts_state = S_CFG_LANENUM_WAIT;
4326            end else if ( cfg_upstream_port && link_latched_live_all_2_ts1_linknmtx_lanen_rcvd ) begin
4327                next_lts_state = S_CFG_LANENUM_WAIT;
4328            end else if ( link_latched_live_all_2_ts1_plinkn_planen_rcvd || timeout_3ms ) begin //3ms = 2ms + 50% margin
4329                next_lts_state = S_PRE_DETECT_QUIET;
4330            end else begin
4331                next_lts_state = S_CFG_LINKWD_ACCEPT;
4332            end
4333        end
```

4.2.6.3.2.1 Downstream Lanes

- 如果配置的链路由至少一组通道组成，且这些通道接收到两个连续的具有相同 Link number 的 TS1 有序集（此 Link number 为 non-PAD 且与下游通道发送的相等），则 Downstream Lane 就发送具有相同 Link number 且 Lane number 设置为 non-Pad 的 TS1 序列集。然后下一个状态跳转到 Configuration.Lanenum.Wait。
  - 分配的 non-PAD Lane number 必须在 0 到 n-1 之间，并顺序分配给接收相同 Link number 的同一通道组，并且未接收 TS1 有序集的 Downstream Lanes 也不得破坏此初始链路编号。任何剩余的通道都必须传输 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。
  - 如果满足以下所有条件，则 use_modified_TS1_TS2_Ordered_Set 变量必须设置为 1b：
    - LinkUp = 0b；
    - 自进入 Polling 状态以来，该组件在处于 Polling 和 Configuration 状态时在 TS1 和 TS2 有序集的 Symbol 5 中的 Enhanced Link Behavior Control 字段中发送了 Modified TS1/TS2 Ordered Sets supported 值（11b），并在轮询和配置状态下传输了 TS2 有序集。
    - 在当前已配置的链路的所有通道上收到的八个连续的 TS2 有序集导致从 Polling.Configuration 到 Configuration 状态的转换，在 Symbol 5 中的 Enhanced Link Behavior Control 字段中具有 Modified TS1/TS2 Ordered Sets supported（11b）。 在接收的八个连续的 TS2 有序集中，32.0 GT/s data rate is supported 字段设置为 1b。

- 如果 2ms 超时，或者链路不能配置，或者所有 Lane 都收到两个连续的 Link number 和 Lane number 设置为 PAD 的 TS1 有序

公众号·志齐业者相承双

集，下个状态就跳转到 Detect 状态。

4.2.6.3.2.2 Upstream Lanes

- 如果配置的链路可以由那些发送过 non-PAD Link number 且正在接受两个连续的 Link number 不变且 Lane number 为 non-PAD 的 TS1 序列集的 Lane 组成，若没有通道反转，则在这些 Lane 上发送 Link number 和 Lane number 与接收值一致的 TS1 序列集，若存在通道反转，则在这些 Lane 上发送 Link number 与接收值一致且 Lane number 与接收值不同的 TS1 序列集。然后下一个状态跳转到 Configuration.Lanenum.Wait。
  - 新分配的 Lane number 必须在 0 到 m-1 的范围内，且必须按顺序分配给接收 non-PAD Lane number 的那些连续分组的通道（即，那些未接收任何 TS1 有序集的通道会破坏这个连续分组，它们不能包含在此分组内）。m-1 必须等于或小于接收到的最大 Lane number，即 n-1。其余 Lane 必须传输 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。
  - 如果满足以下所有条件，则 use_modified_TS1_TS2_Ordered_Set 变量必须设置为 1b：
    - LinkUp = 0b；
    - 自进入 Polling 状态以来，该组件在处于 Polling 和 Configuration 状态时在 TS1 和 TS2 有序集的 Symbol 5 中的 Enhanced Link Behavior Control 字段中发送了 Modified TS1/TS2 Ordered Sets supported 值（11b），并在轮询和配置状态下传输了 TS2 有序集。
    - 在当前已配置的链路的所有通道上收到的八个连续的 TS2 有序集导致从 Polling.Configuration 到 Configuration 状态的转换，在 Symbol 5 中的 Enhanced Link Behavior Control 字段中具有 Modified TS1/TS2 Ordered Sets supported（11b）。 在接收的八个连续的 TS2 有序集中，32.0 GT/s data rate is supported 字段设置为 1b。
- 如果 2ms 超时，或者链路不能配置，或者所有 Lane 都收到两个连续的 Link number 和 Lane number 设置为 PAD 的 TS1 有序集，下个状态就跳转到 Detect 状态。

IMPLEMENTATION NOTE

Example Cases

Notable examples related to the configuration of Downstream Lanes:

1. 一个可以分为两个 x4 链路的 x8 的 Downstream Port，它将两个不同的 Link number 发送给两个 x4 的 Upstream Port。Upstream Port 通过选择两个 Link number 来同时响应。 Downstream Port 必须选择其中一组配置为链路，而其余的则留给另外一个 LTSSM 进行配置（这会在 Configuration.Complete 中发生）。
2. 一个可以分为两个 x8 链路的 x16 的 Downstream Port 连接到一个可以配置成 x12、或 x8 和 x4 链路的 x12 的 Upstream Port 上。在 Configuration.Linkwidth.Start 状态，Upstream Port 在所有 12 个 Lane 上返回相同的 Link number。 然后，Downstream Port 将发送相同的 Link number 给 Upstream Port，并在可以形成 x8 链路的八个 Lane 中发送 Lane number，其余四个 Lane 发送 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
3. 一个 x8 的 Downstream Port，其中只有七个通道正在接收具有相同 Link number 的 TS1 有序集，而第八个通道位于这些通道的中间或相邻位置 ，它没有收到 TS1 有序集。在这种情况下，第八通道被视为与其他七个通道相同，x8 通道的通道编号过程应如上所述进行。

Notable examples related to the configuration of Upstream Lanes:

1. 一个 x8 的 Upstream Port 是由反向编号的 Lane number 表示的。 如果上游端口支持通道反转，则 Upstream Port 会将相同

的 Lane number 发送回 Downstream Port。否则，相反的 Lane number 将发送回 Downstream Port，以选择性地确定通道顺序或退出 Configuration 状态。

通道反转必须是 Upstream Port 支持的功能，Downstream Port 不会实现此功能。具体来说，Upstream Port 的通道反转大概是这种场景：默认的 Upstream Port 的通道编号（0 到 n-1）对应正在发送的 Upstream Port 的通道编号（n-1 到 0）。

2. 一个 x8 的 Upstream Port 在其 Lane 0 上未收到 TS1 有序集，则：

a. 在 Upstream Port 只支持一个 x8 或 x1 链路并且还支持通道反转的情况下，Upstream Port 将会只把接收到的 Lane 7 分配给 Lane 0，其它七个 Lane 发送 Link number 和 Lane number 为 PAD 的 TS1 序列集。

b. 在 Upstream Port 只支持一个 x8 或 x1 链路并且不支持通道反转的情况下，不会形成任何链路并且链路会在 2ms 超时后进入 Detect 状态。

3. 一个可以分成两个 x4 链路的 x8 的 Upstream crosslink Port 连接到两个具有相同 Link nuber 的 x4 的 Downstream Port，并且每个 x4 的 Downstream Port 同时显示分别编号为 0 到 3 的 Lane number。Upstream Port 必须选择这些 Lane number 中的一组来配置为链路，而让另一组通过 Configuration 状态。

4.2.6.3.3 Configuration.Lanenum.Accept

```
4345        S_CFG_LANENUM_ACEPT: begin
4346            if ( ((~cfg_upstream_port && link_latched_live_all_2_ts1_linknmtx_lanenmtx_rcvd) || (cfg_upstream_port && link_latched_live_all_2_ts2_linknmtx_lanenmtx_rcvd)) && ~link_mode_changed ) begin
4347                next_lts_state = S_CFG_COMPLETE;
4348            end else if ( link_latched_live_all_2_ts1_linknmtx_lanen_rcvd && link_mode_changed ) begin
4349                next_lts_state = S_CFG_LANENUM_WAIT;
4350            end else if ( link_latched_live_all_2_ts1_plinkn_planen_rcvd || timeout_3ms ) begin //3ms = 2ms + 50% margin
4351                next_lts_state = S_PRE_DETECT_QUIET;
4352            end else begin
4353                next_lts_state = S_CFG_LANENUM_ACEPT;
4354            end
4355        end
```

在此子状态，如果 use_modified_TS1_TS2_Ordered_Set 变量设置为 1b，则：

- 发送器必须发送 modified TS1 Ordered Set，以替换 TS1 Ordered Set 。
- 接收器必须检查是否收到了 modified TS1 Ordered Set 而不是检查是否收到了 TS1 Ordered Set [注意：有关 identical consecutive modified TS1 Ordered Set 的定义，请参见 Section 4.2.4.1。]

4.2.6.3.3.1 Downstream Lanes

- 如果 Downstream Lanes 收到了两个连续的与其发送的 non-PAD Link number 和 non-PAD Lane number 相匹配的 TS1 序列集(如果支持通道反转，则收发的 Lane number 要翻转)，则下个状态会跳转到 Configuration.Complete 状态。注意：允许用 Retimer 来延迟进入 Configuration.Complete，此过程在 Section 4.3.8 中描述：
  - 如果当前的 Configuration 是从 Recovery 状态转换而来的，则按照下述规则更新 Link Status 寄存器的 Link Bandwidth Management Status 和 Link Autonomous Bandwidth Status 字段：
    a. 如果带宽变化是因 Downstream Port 由可靠因素发出的，则 Link Bandwidth Management Status = 1b。
    b. 否则，如果带宽变化不是因 Downstream Port 发起的且接收到的两个连续的 TS1 有序集的 Autonomous Change 字段为 0，则 Link Bandwidth Management Status = 1b。
    c. 否则，Link Autonomous Bandwidth Status = 1b。
  - 反转的通道号严格定义为 Downstream Lane 0 收到的 TS1 序列集中携带的 Lane number 为 n-1，Downstream Lane n-1 收到的 TS1 有序集中携带的 Lane number 为 0。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更

公众号·芯片业者相承双

多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。

- 如果链路可以由那些收到了两个连续的与其发送的 non-PAD Link number 和 non-PAD Lane number 相匹配的 TS1 序列集的 Lane 组成，则在这些 Lane 上发送具有相同 Link number 以及新的 Lane number 的 TS1 序列集，并且下个状态会转换到 Configuration.Lanenum.Wait 状态。
  - 新分配的 Lane number 必须在 0 到 m-1 的范围内，且必须按顺序分配给接收 non-PAD Lane number 的那些连续分组的通道（即，那些未接收任何 TS1 有序集的通道会破坏这个连续分组，它们不能包含在此分组内）。m-1 必须等于或小于接收到的最大 Lane number，即 n-1。其余 Lane 必须传输 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。
- 如果无法配置链路，或者在所有 Lane 上收到两个连续的 Link number 与 Lane number 为 PAD 的 TS1 序列集，则状态机跳转到 Detect 状态。

4.2.6.3.3.2 Upstream Lanes

- 如果 Upstream Lanes 收到了两个连续的与其发送的 non-PAD Link number 和 non-PAD Lane number 相匹配的 TS1 序列集，则下个状态会跳转到 Configuration.Complete 状态。注意：允许用 Retimer 来延迟进入 Configuration.Complete，此过程在 Section 4.3.8 中描述。
- 如果链路可以由那些收到了两个连续的与其发送的 non-PAD Link number 和 non-PAD Lane number 相匹配的 TS1 序列集的 Lane 组成，则在这些 Lane 上发送具有相同 Link number 以及新的 Lane number 的 TS1 序列集，并且下个状态会转换到 Configuration.Lanenum.Wait 状态。
  - 新分配的 Lane number 必须在 0 到 m-1 的范围内，且必须按顺序分配给接收 non-PAD Lane number 的那些连续分组的通道（即，那些未接收任何 TS1 有序集的通道会破坏这个连续分组，它们不能包含在此分组内）。m-1 必须等于或小于接收到的最大 Lane number，即 n-1。其余 Lane 必须传输 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
  - 建议在任何有可能在 TS1 有序集中接收到错误，或者丢失 128b/130b 块对齐的多通道链路；在使用 8b/10b 编码时，将上面列出的评估延迟额外两个或多个 TS1 有序集，或者在使用 128b/130b 编码时，将上述评估延迟额外 34 个或更多 TS1 有序集，但不能超过 1 ms，以免过早配置出更小的链路。
- 如果无法配置链路，或者在所有 Lane 上收到两个连续的 Link number 与 Lane number 为 PAD 的 TS1 序列集，则状态机跳转到 Detect 状态。

4.2.6.3.4 Configuration.Lanenum.Wait

```
4334        S_CFG_LANENUM_WAIT: begin
4335            if ( (~cfg_upstream_port && link_latched_live_all_2_ts1_linknmtx_lanenmtx_rcvd) || (cfg_upstream_port && link_any_2_ts2_rcvd) ) begin
4336                next_lts_state = S_CFG_LANENUM_ACEPT;
4337            end else if ( timeout_1ms && link_latched_live_any_2_ts1_lanendiff_linkn_rcvd ) begin
4338                next_lts_state = S_CFG_LANENUM_ACEPT;
4339            end else if ( link_latched_live_all_2_ts1_plinkn_planen_rcvd || timeout_3ms ) begin //3ms = 2ms + 50% margin
4340                next_lts_state = S_PRE_DETECT_QUIET;
4341            end else begin
4342                next_lts_state = S_CFG_LANENUM_WAIT;
4343            end
4344        end
```

在此子状态，如果 use_modified_TS1_TS2_Ordered_Set 变量设置为 1b，则：

- 发送器必须发送 modified TS1 Ordered Set，以替换 TS1 Ordered Set 。
- 接收器必须检查是否收到了 modified TS1 Ordered Set 而不是检查是否收到了 TS1 Ordered Set [注意： 有关 identical

consecutive modified TS1 Ordered Set 的定义，请参见 Section 4.2.4.1。]

4.2.6.3.4.1 Downstream Lanes

- 如果检测到 Receiver 的 Lane 收到两个连续的 Lane number 与刚进入 Configuration.Lanenum.Wait 时收到的 Lane 值不同的 TS1 序列集，并且不是所有 Lane 的 Link number 都设置为 PAD 或者在所有 Lane 上收到了两个连续的 Lane number 与 Link number 与其之前发送的一致的 TS1 序列集，则下个状态跳转到 Configuration.Lanenum.Accept。

  在转换到 Configuration.Lanenum.Accept 之前，允许 Upstream Lanes 延迟 1ms。目的是为了防止收到的错误或在通道之间的 skew 影响最终配置的链路宽度。

  上述条件隐含两个点：一个是两个 Port 要链路宽度协商完成，一个是在转换到 Configuration.Lanenum.Accept 时 Lane number 发生了改变。

  为了允许两个端口以协商的链路宽度为基础，要求接收与 Lane 首次进入 Configuration.Lanenum.Wait 时不同的任何 Lane Number 的条件是必要的。语句“任何一个 Lane 都接收两个连续的 TS1 有序集，其 Lane Number 与 Lane 首次进入 Configuration.Lanenum.Wait 时的 Lane Number 不同”的确切含义要求 Lane Number 必须与最近的在切换到 Configuration.Lanenum.Accept 之前进入 Configuration.Lanenum.Wait 时的改变值。

- 如果 2ms 超时，或者所有 Lane 都收到两个连续的 Link number 和 Lane number 设置为 PAD 的 TS1 有序集，下个状态就跳转到 Detect 状态。

4.2.6.3.4.2 Upstream Lanes

- 如果满足下述两个条件中的一个，状态机就跳转到 Configuration.Lanenum.Accept。

  a. 如果任意 Lane 收到两个连续的 Lane number 与刚进入 Configuration.Lanenum.Wait 时收到的 Lane 值不同的 TS1 序列集，并且不是所有 Lane 的 Link number 都设置为 PAD。

  b. 如果任意 Lane 收到两个连续的 TS2 序列。

- 如果 2ms 超时，或者所有 Lane 都收到两个连续的 Link number 和 Lane number 设置为 PAD 的 TS1 有序集，下个状态就跳转到 Detect 状态

4.2.6.3.5 Configuration.Complete

```
4356        S_CFG_COMPLETE: begin
4357            if ( link_latched_live_all_8_ts2_linknmtx_lanenmtx_rcvd && link_xmlh_16_ts2_sent_after_1_ts2_rcvd && rmlh_deskew_complete ) begin
4358                next_lts_state = S_CFG_IDLE;
4359            end else if ( timeout_2ms ) begin
4360                if ( !(&idle_to_rlock) & ((current_data_rate == `GEN3_RATE) || (current_data_rate == `GEN4_RATE)) ) //idle_to_rlock < ffh & gen3 data rate
4361                  next_lts_state = S_CFG_IDLE;
4362                else
4363                  next_lts_state = S_PRE_DETECT_QUIET;
4364            end else begin
4365                next_lts_state = S_CFG_COMPLETE;
4366            end
4367        end
```

设备在进入此子状态之前允许通告改变支持的 data rate 和 upconfigure 的能力，但是在进入此状态之后就不允许改变了。

在此子状态，如果 use_modified_TS1_TS2_Ordered_Set 变量设置为 1b，则：

- 发送器必须发送 modified TS1 Ordered Set，以替换 TS1 Ordered Set 。
- 接收器必须检查是否收到了 modified TS1 Ordered Set 而不是检查是否收到了 TS1 Ordered Set [注意：有关 identical consecutive modified TS1 Ordered Set 的定义，请参见 Section 4.2.4.1。]

公众号·芯片业者相承双

4.2.6.3.5.1 Downstream Lanes

- 在此子状态下，Downstream Lane 会发送与收到的 TS1 序列集中 Link number 和 Lane number 一致的 TS2 序列集。
  - 可以把 TS2 有序集的 Upconfigure Capability 字段置为 1，表示此 Port 可以在此分配的 Lane 0 上支持 x1 链路，并且在 LinkUp = 1 的时候可以向上配置（up-configuring）链路。通告这个能力是个可选择实现的功能。

  Upconfigure Capability 可以在 LinkUp = 1 的时候重新进入 Configuration 状态以协商是否要 upsize 或 downsize 链路宽度。
- 在此子状态退出时必须要为 L0s 设置好 N_FTS 值。
- 当使用 8b/10b 编码时，在此子状态退出时必须完成 Lane-to-Lane de-skew。
- 如果传输的 TS2 序列集中的 Disable Scrambling 字段有效，则禁用加扰功能。
  - 在所有配置的 Lane 上发送 Disable Scrambling 置 1 的训练序列的 Port 本身也需要禁用加扰。加扰功能只能在 8b/10b 编码的时候可以被禁用。
- 如果在发送 TS2 序列集的所有 Lane 上收到了 8 个有相同 Link number 和 Lane number 且 Data Rate Identifiers 字段相同（Link Upconfigure Capability 也相同）的 TS2 序列集，并且在收到一个 TS2 序列集之后发送完 16 个 TS2 序列集，状态机就立马跳转到 Configuration.Idle 状态。当数据速率为 2.5 GT/s 时，将 Link Capabilities 2 寄存器的 Retimer Presence Detect Supported 字段设置为 1b 的实现也必须接收八个连续的 Retimer Present 字段相同的 TS2 有序集。当数据速率为 2.5 GT/s 时，将 Link Capabilities 2 寄存器的 Two Retimers Presence Detect Supported 字段设置为 1b 的实现也必须接收八个连续的 Retimer Present 字段相同的 TS2 有序集。
  - 如果数据速率为 2.5GT/s：
    - 如果 Link Capabilities 2 寄存器的 Retimer Presence Detect Supported 字段设置为 1b，并且任何已配置通道在八个连续接收的 Retimer Present 字段设置为 1b 的 TS2 有序集，则必须在 Link Status 2 寄存器中将 Retimer Presence Detected 字段设置为 1b，否则设置为 0b 。
    - 如果 Link Capabilities 2 寄存器的 Two Retimer Presence Detect Supported 字段设置为 1b，并且任何已配置通道在八个连续接收的 Two Retimer Present 字段设置为 1b 的 TS2 有序集，则必须在 Link Status 2 寄存器中将 Two Retimer Presence Detected 字段设置为 1b，否则设置为 0b 。
  - 如果设备支持大于 2.5GT/s 的 data rate，它必须记录下载所有 Lane 上收到的 Data Rate Identifiers。这可能会覆盖之前记录的 data rate 值。定义了一个用于在 Recovery 状态切换 data rate 的变量 ***changed_speed_recovery***，该变量在此子状态下设置为 0b。
  - 如果设备发送的 TS2 有序集中携带的 Link Upconfigure Capability 字段为 1，并且收到了 8 个连续的 Link Upconfigure Capability 字段为 1 的 TS2 序列集，则 upconfigure_capable 变量设置为 1，否则设置为 0。
  - 所有未经配置的剩余的 Lane 不再与 LTSSM 有任何关系，他们必须做如下处理：
    1) 如果支持多个 LTSSM，则可以用新的 LTSSM 来处理。
    2) 或者，所有没有与新的 LTSSM 关联的其他 Lane 进入电气空闲状态。建议保留这些通道的 Receiver terminations。如果未将其保留，则必须在 LTSSM 进入 Recovery.RcvrCfg 时将其打开，直到达到 Configuration.Complete 子状态（如果 upconfigure_capable 设置为 1b 以允许潜在的链路宽度的 upconfiguration）。 在通过 L0 进行初始链路训练期间，任何不属于 LTSSM 的通道都不能成为 LTSSM 的一部分，这是链路宽度 upconfiguration 过程的一部分。
       - 在 crosslink 情况下，Receiver terminations 必须满足 Table 8-10 定义的 $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}POS}$ 和 $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}NEG}$ 。
       - 这些 Lane 必须在 LTSSM 进入 Detect 状态之后，与 LTSSM 重新建立联系。

公众号·芯片业者相关权

- 在进入电气空闲状态之前没必要发送 EIOS，也没必要在符号或有序集的边缘进入电气空闲状态。

- 2ms 超时后，
  - 如果当前 data rate 是 2.5GT/s 或者 5GT/s，下个状态就跳抓到 Detect 状态。
  - 如果 idle_to_rlock_transitioned < FFh，且当前 data rate 是 8GT/s 或更高，下个状态跳转到 Configuration.Idle。
    - changed_speed_recovery 复位成 0。
    - 不属于已配置链路的 Lane 不再与正在工作的 LTSSM 相关联，并且要符合上述 1)和 2)的 non-timeeout 的进入 Configuration.Idle 状态的条件。
    - upconfigure_capable 可以被设置。如果训练序列的 Link Upconfigure Capability = 1b，upconfigure_capable 就设置为 1b，反之设置为 0b。
  - 否则，下个状态就跳抓到 Detect 状态。

4.2.6.3.5.2 Upstream Lanes

- 在此子状态下，Downstream Lane 会发送与收到的 TS1 序列集中 Link number 和 Lane number 一致的 TS2 序列集。
  - 可以把 TS2 有序集的 Upconfigure Capability 字段置为 1，表示此 Port 可以在此分配的 Lane 0 上支持 x1 链路，并且在 LinkUp = 1 的时候可以向上配置（up-configuring）链路。通告这个能力是个可选择实现的功能。
- 在此子状态退出时必须要为 L0s 设置好 N_FTS 值。
- 当使用 8b/10b 编码时，在此子状态退出时必须完成 Lane-to-Lane de-skew。
- 如果传输的 TS2 序列集中的 Disable Scrambling 字段有效，则禁用加扰功能。
  - 在所有配置的 Lane 上发送 Disable Scrambling 置 1 的训练序列的 Port 本身也需要禁用加扰。加扰功能只能在 8b/10b 编码的时候可以被禁用。
- 如果在发送 TS2 序列集的所有 Lane 上收到了 8 个有相同 Link number 和 Lane number 且 Data Rate Identifiers 字段相同（Link Upconfigure Capability 也相同）的 TS2 序列集，并且在收到一个 TS2 序列集之后发送完 16 个 TS2 序列集，状态机就立马跳转到 Configuration.Idle 状态。当数据速率为 2.5 GT/s 时，将 Link Capabilities 2 寄存器的 Retimer Presence Detect Supported 字段设置为 1b 的实现也必须接收八个连续的 Retimer Present 字段相同的 TS2 有序集。当数据速率为 2.5 GT/s 时，将 Link Capabilities 2 寄存器的 Two Retimers Presence Detect Supported 字段设置为 1b 的实现也必须接收八个连续的 Retimer Present 字段相同的 TS2 有序集。
  - 如果数据速率为 2.5GT/s：
    - 如果 Link Capabilities 2 寄存器的 Retimer Presence Detect Supported 字段设置为 1b，并且任何已配置通道在八个连续接收的 Retimer Present 字段设置为 1b 的 TS2 有序集，则必须在 Link Status 2 寄存器中将 Retimer Presence Detected 字段设置为 1b，否则设置为 0b 。
    - 如果 Link Capabilities 2 寄存器的 Two Retimer Presence Detect Supported 字段设置为 1b，并且任何已配置通道在八个连续接收的 Two Retimer Present 字段设置为 1b 的 TS2 有序集，则必须在 Link Status 2 寄存器中将 Two Retimer Presence Detected 字段设置为 1b，否则设置为 0b 。
  - 如果设备支持大于 2.5GT/s 的 data rate，它必须记录下载所有 Lane 上收到的 Data Rate Identifiers。这可能会覆盖之前记录的 data rate 值。定义了一个用于在 Recovery 状态切换 data rate 的变量 changed_speed_recovery，该变量在此子状态下设置为 0b。

公众号：芯片业者相建双

- 如果设备发送的 TS2 有序集中携带的 Link Upconfigure Capability 字段为 1，并且收到了 8 个连续的 Link Upconfigure Capability 字段为 1 的 TS2 序列集，则 upconfigure_capable 变量设置为 1，否则设置为 0。
- 所有未经配置的剩余的 Lane 不再与 LTSSM 有任何关系，他们必须做如下处理：
  1) 如果支持多个 LTSSM，则可以用新的 crosslink LTSSM 来处理。
  2) 或者，所有没有与新的 crosslink LTSSM 关联的 Lane 进入电气空闲状态。建议保留这些通道的 Receiver terminations。如果未将其保留，则必须在 LTSSM 进入 Recovery.RcvrCfg 时将其打开，直到达到 Configuration.Complete 子状态（如果 upconfigure_capable 设置为 1b 以允许潜在的链路宽度的 upconfiguration）。在通过 L0 进行初始链路训练期间，任何不属于 LTSSM 的通道都不能成为 LTSSM 的一部分，这是链路宽度 upconfiguration 过程的一部分。
     - 这些 Lane 必须在 LTSSM 进入 Detect 状态之后，与 LTSSM 重新建立联系。
     - 在进入电气空闲状态之前没必要发送 EIOS，也没必要在符号或有序集的边缘进入电气空闲状态。

- 2ms 超时后，
  - 如果当前 data rate 是 2.5GT/s 或者 5GT/s，下个状态就跳抓到 Detect 状态。
  - 如果 idle_to_rlock_transitioned < FFh，且当前 data rate 是 8GT/s 或更高，下个状态跳转到 Configuration.Idle。
    - changed_speed_recovery 复位成 0。
    - 不属于已配置链路的 Lane 不再与正在工作的 LTSSM 相关联，并且要符合上述 1)和 2)的 non-timeeout 的进入 Configuration.Idle 状态的条件。
    - upconfigure_capable 可以被设置。如果训练序列的 Link Upconfigure Capability = 1b，upconfigure_capable 就设置为 1b，反之设置为 0b。
  - 否则，下个状态就跳抓到 Detect 状态。

4.2.6.3.6 Configuration.Idle

```
4368        S_CFG_IDLE: begin
4369            //not timeout from S_CFG_COMPLETE to S_CFG_IDLE for Gen3 rate
4370            if ( (((current_data_rate == `GEN3_RATE) || (current_data_rate == `GEN4_RATE)) && !latched_to_idle_timeout && rcvd_8idles && idle_16_sent) ||
4371                 (((current_data_rate == `GEN1_RATE) || (current_data_rate == `GEN2_RATE)) && rcvd_8idles && idle_16_sent) )
4372                next_lts_state = S_L0;
4373            else if (timeout_2ms) begin
4374                if ( !(&idle_to_rlock) ) //idle_to_rlock < ffh
4375                    next_lts_state = S_RCVRY_LOCK;
4376                else
4377                    next_lts_state = S_PRE_DETECT_QUIET;
4378            end else
4379                next_lts_state = S_CFG_IDLE;
4380        end
```

- 在此子状态下，当使用 8b/10b 编码，发送器会在所有已配置的 Lane 上发送 Idle data Symbol。
- 如果 LinkUp = 0b 且链路的所有组件都支持 32.0 GT/s 数据速率，就如在进入 Configuration.Idle 之前收到的八个连续的 TS2 或八个连续且相同的 modified TS2 Ordered Set 中通告的那样，则：
  - 如果在收到的八个连续且相同的 modified TS2 Ordered Set 中将 No Equalization Needed 字段设置为 1b，并且还在链路的所有已配置通道中的已发送 modified TS2 Ordered Set 中进行了设置；或者在收到的八个连续且相同的 modified TS2 Ordered Set 中将 Enhanced Link Behavior Control 字段设置为 10b，并且还在链路的所有已配置通道中的已发送 modified TS2 Ordered Set 中进行了设置，则：
    - equalization_done_8GT_data_rate、equalization_done_16GT_data_rate 和 equalization_done_32GT_data_rate 变量设置为 1b。

公众号：芯片业者怕承双

- 32.0 GT/s Status Register 中的 No Equalization Needed Received 设置为 1b 。

- 或者，如果在收到的八个连续且相同的 modified TS2 Ordered Set 中将 Equalization bypass to highest rate support 字段设置为 1b，并且还在链路的所有已配置通道中的已发送 modified TS2 Ordered Set 中进行了设置；或者在收到的八个连续且相同的 modified TS2 Ordered Set 中将 Enhanced Link Behavior Control 字段设置为 01b 或 10b，并且还在链路的所有已配置通道中的已发送 modified TS2 Ordered Set 中进行了设置，则：
  - equalization_done_8GT_data_rate、equalization_done_16GT_data_rate 变量设置为 1b 。
- 如果进入此子状态是由于接收到八个连续且相同的 modified TS2 Ordered Set 且 LinkUp = 0b：
  - 如果在接收到的八个连续的 modified TS2 Ordered Set 中将 Modified TS Usage 字段设置为 010b（备用协议），并且在在链路的所有已配置通道中，已发送的 modified TS2 Ordered Set 的 Modified TS Usage 字段设置为相同值，并且 Modified TS Information 1 和 Alternate Protocol Vendor ID 字段在发送和接收的 modified TS2 Ordered Set 中有相同的值，则：
    - 32.0 GT/s Status 寄存器中的 Modified TS Received 字段设置为 1b。基于接收到的八个连续的 modified TS2 Ordered Set，协商的详细信息将反映在 Received Modified TS Data 1 Register 和 Received Modified TS Data 2 Register 中。
- 当使用 128b/130b 编码：
  - 当数据速率为 8.0GT/s，发送器会在所有已配置的 Lane 上先发送一个 SDS 有序集，然后紧跟着发 Idle data Symbol。Lane 0 上的第一个 Idle data Symbol 就是数据流的第一个符号。
  - 当数据速率为 16.0GT/s 或更高，则发送器在所有已配置通道上发送一个 Control SKP Ordered Set，然后立即发送一个 SDS 有序集以启动数据流，然后在所有已配置通道上发送 Idle data Symbol。在 Lane 0 上发送的第一个 Idle data Symbol 是数据流的第一个符号。
- 接收器在此子状态下等待接收 Idle data。
- LinkUp = 1b。
- 当使用 8b/10b 编码，如果在所有已配置的 Lane 上接收到 8 个连续的 Idle data 并且在接收到一个 Idle data 后发送完 16 个 Idle data，状态机就跳转到 L0。
  - 如果自上次从 Recovery 或 Configuration 进入 L0 后，软件将 Link Control 寄存器的 Retrain Link 字段写成 1b，则 Downstream Port 必须设置 Link Status 寄存器的 Link Bandwidth Management Status 字段。
  - 状态机切换到 L0 后，use_modified_TS1_TS2_Ordered_Set 变量重置为 0b 。
- 当使用 128b/130b 编码，如果在所有已配置的 Lane 上接收到 8 个连续的 Idle data 并且在接收到一个 Idle data 后发送完 16 个 Idle data，并且此状态不是由 Configuration.Complete 超时而进入的，状态机就跳转到 L0。
  - 此时的 Idle data 必须在接收的 Data Block 里。
  - 数据流处理开始之前必须完成 Lane-to-Lane de-skew。
  - 如果自上次从 Recovery 或 Configuration 进入 L0 后，软件将 Link Control 寄存器的 Retrain Link 字段写成 1b，则 Downstream Port 必须设置 Link Status 寄存器的 Link Bandwidth Management Status 字段。
  - 在切换为 L0 时 idle_to_rlock_transitioned 变量复位为 00h。
- 否则，在 2ms 超时后执行下述操作：
  - 如果 idle_to_rlock_transitioned < FFh，状态机跳转到 Recovery.RcvrLock。

公众号：芯片工程师相承双

- ✧ 在转换到 Recovery.RcvrLock 时，
  - ❖ 如果 data rate 是 8GT/s，idle_to_rlock_transitioned 自增 1。
  - ❖ 如果 data rate 是 2.5GT/s 或 5GT/s，idle_to_rlock_transitioned 直接设置为 FFh。
- ◆ 否则，状态机跳转到 Detect。

idle_to_rlock_transitioned 定义的是从 Configuration.Idle 跳转到 Recovery.RcvrLock 的次数。

4.2.6.4 Recovery

在 Recovery 状态，发送器和接收器都使用已配置的链路和通道号以及支持的数据速率发送和接收数据。Recovery 允许已配置的链路根据需要调整数据速率，重新建立 bit lock、Symbol lock 或 Block alignment、Lane-to-Lane de-skew 。Recovery 还用于设置新的 N_FTS 值，并进入 Loopback、Disabled、Hot Reset 和 Configuration 状态。

Recovery 状态的子状态机如 Figure 4-30 所示。

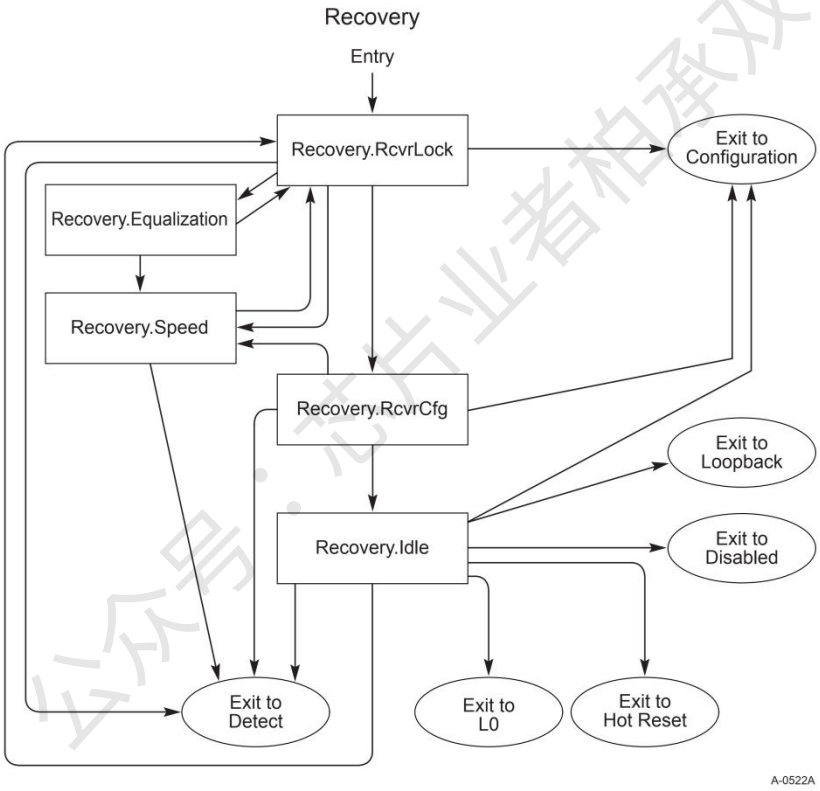


Figure 4-30 Recovery Substate Machine

4.2.6.4.1 Recovery.RcvrLock

如果链路以 8.0 GT/s 或更高的数据速率运行，则接收器必须在获得该 Lane 的 Block Alignment 后才考虑接收任何 TS1 或 TS2 有序集。 如果从 L1 或 Recovery.Speed 或 L0s 进入此子状态，则必须在退出电气空闲状态后获得 Block Alignment。如果从 L0 进入此子状态，则必须在最后一个数据流结束后获得 Block Alignment。

- 如果在 data rate 为 8.0GT/s 或更高速率下工作：
  - 如果 start_equalization_w_preset = 1b：

- Upstream Port 必须使用在 Recovery.RcvrCfg 状态从接收到的适当的八个连续 TS2 有序集中携带的发送器 preset 值（如果协商为 8.0 GT/s，则为 EQ TS2；如果协商为 32.0 GT/s 且为 equalization bypass to highest data rate，则为 EQ TS2，或者如果协商为 16.0GT/s 或 32.0GT/s，则为 128b/130b EQ TS2），如 Chapter 8 所述。接收到保留或不支持的发送器预设值的通道必须使用实现特定的方法来选择支持的发送器预设设置，以便在它开始以需要执行均衡的数据速率开始发送时使用。

- Downstream Port 开始以必须执行均衡的数据速率进行传输时，必须按照以下规则使用发送器预设设置：

  1. 如果均衡的数据速率为 16.0 GT/s 或 32.0 GT/s，并且接收到八个连续的 EQ TS2 有序集（对于要进行到 32.0 GT / s 的均衡旁路的情况）或 128b / 130b EQ TS2 有序集，且它们携带在 Recovery.RcvrCfg 进行的最新转换中具有受支持的发送器预设值，则必须使用来自那些 EQ TS2 或 128b/130b EQ TS2 有序集的发送器预设值。

  2. 否则，如果支持在适当的 Lane Equalization Control Register Entry 的 Downstream Port Transmitter Preset 字段中定义的 Transmitter Preset 值（如下所述），则必须使用它：

Data Rate of Equalization	Transmitter Preset value to be used as soon as the Link transitions to the data rate of equalization
8.0 GT/s	Transmitter Preset field defined in the Lane Equalization Control Register Entry for each Lane. The Downstream Port may optionally use the Downstream Port 8.0 GT/s Receiver Preset Hint field defined in the Lane Equalization Control Register Entry for each of its Receivers corresponding to the Lane, if they are not Reserved values.
16.0 GT/s	Downstream Port 16.0 GT/s Transmitter Preset field of the 16.0 GT/s Lane Equalization Control Register Entry
32.0 GT/s	Downstream Port 32.0 GT/s Transmitter Preset field of the 32.0 GT/s Lane Equalization Control Register Entry

  3. 否则，请使用特定于实现的方法来选择支持的发送器预设设置。

  下游端口必须确保其符合 Chapter 8 中的预设定义。

- 下个状态跳转到 Recovery.Equalization 状态。

■ 否则：

- 发送器必须使用最后一次均衡过程结束时商定的系数设置。

- 如果此子状态是从 Recovery.Equalization 中进入的，则在传输的 TS1 有序集中，下游端口必须将 Pre-cursor、Cursor 和 Post-cursor Coefficient 字段设置为当前的发送器的设置，并且如果在 Recovery.Equalization 的 Phase 2 中最后接受的请求是一个 preset 请求，它必须将 Transmitter Preset 设置为该请求的可接受预设。

- 建议在此子状态下，在传输的 TS1 有序集中，所有端口都将 Pre-cursor、Cursor 和 Post-cursor Coefficient 字段设置为当前的发送器的设置，并将 Transmitter Preset 位设置为最新的预设值。

- Upstream Port 在所有已配置的 Lane 上收到的 8 个连续的 TS1 有序集有以下特征，则状态机跳转到 Recovery.Equalization 状态。

  - 接收到的 TS1 有序集中的 Link number 和 Lane number 与每个通道上发送的 TS1 有序集中的 Link number 和 Lane number 匹配。

  - speed_change = 0b。

  - EC 字段不能等于 00b。

IMPLEMENTATION NOTE

公众号：芯片业者相关权

Redoing Equalization

Downstream Port 可以使用此条款通过软件帮助或某些其他特定于实现的方式来重做发送器均衡过程的某些部分以避免超时，同时确保链路上没有正在传输的事务。

- Downstream Port 在直接定向且 Recovery.RcvrLock 状态不是从 Configuration.Idle 或 Recovery.Idle 进入的情况下跳转到 Recovery.Equalization 状态。在开始传输 Recovery.Equalization 要求的 TS1 有序集之前，端口必须确保由于处于 Recovery.RcvrLock 状态而传输的 EC = 00b 的 TS1 有序集不超过 2 个。

- 在此子状态下，发送器在所有已配置的 Lane 上发送 Configuration 完成时设置好的 Link number 和 Lane number 的 TS1 有序集。如果变量 directed_speed_change = 1b，则 TS1 有序集的 speed_change 字段（Data Rate Identifier 字符的 bit7）就必须设置为 1b。如果在任意 Lane 上收到 8 个连续的 speed_change 字段设置为 1b 的 TS1 有序集，则 directed_speed_change 变量设置为 1b。只能通告那些能够可靠支持的大于 2.5 GT/s 的数据速率。发送的 TS1 有序集中的 N_FTS 表示在当前数据速率下快速恢复需要的 FTS 数量。允许设备更改进入此子状态时通告的支持的数据速率。

  当支持 equalization bypass to highest data rate 时，Downstream Port 想通过重做均衡把数据速率从 2.5 GT/s 或 5.0 GT/s 切换到 8.0 GT/s 或 32.0 GT/s，则必须：

  - 发送 EQ TS1 Ordered Set，并将 speed_change 位设置为 1b，并通告以下数据速率：
    - 如果重做均衡用于 8.0 GT/s 数据速率，则通告 8.0 GT/s Data Rate Identifier。
    - 如果重做均衡用于 32.0 GT/s 数据速率，则通告 32.0 GT/s Data Rate Identifier。
  - 如果均衡重做尝试是由硬件（如 Section 4.2.3 所述）启动的，则硬件必须在启动尝试之前确保数据速率为 2.5 GT/s 或 5.0 GT/s。
  - 如果按 Section 4.2.3 中所述由软件机制启动了均衡重做尝试，则在启动尝试之前，软件必须确保数据速率为 2.5 GT/s 或 5.0 GT/s。

  打算以数据速率从 8.0 GT/s 更改为 16.0 GT/s 或从 16.0 GT/s 更改为 32.0 GT/s 的方式进行重做均衡的下游端口必须：

  - 发送将 Equalization Redo 设置为 1b，speed_change 位设置为 1b 的 TS1 Ordered Set，并通告将执行均衡重做的 Data Rate Identifier（16.0 GT/s 或 32.0 GT/s）。
  - 如果均衡重做尝试是由硬件（如 Section 4.2.3 中所述）启动的，则硬件必须在启动重做均衡尝试之前确保数据速率为以下值：
    - 如果均衡重做是为 16.0GT/s 数据速率而做的，则必须保证数据速率为 8.0GT/s 。
    - 如果均衡重做是为 32.0GT/s 数据速率而做的，则必须保证数据速率为 16.0GT/s 。
  - 如果按 Section 4.2.3 中所述由软件机制启动了均衡重做尝试，则在启动重做均衡尝试之前，软件必须确保数据速率为以下值：
    - 如果均衡重做是为 16.0GT/s 数据速率而做的，则必须保证数据速率为 8.0GT/s 。
    - 如果均衡重做是为 32.0GT/s 数据速率而做的，则必须保证数据速率为 16.0GT/s 。

  上游端口必须在它在 Recovery.RcvrCfg 中传输的 TS2 有序集中通告其最高数据速率支持，并且可选地在此子状态下传输在 TS1 有序集合中通告最高的数据速率支持，除非上游端口确定与最高数据速率无关的问题，如果均衡接收到的八个连续的有序集是以下之一，则均衡将阻止它在要求执行均衡的最高数据速率下可靠运行：

  - 将 speed_change 位设置为 1b 的 EQ TS1 或 EQ TS2 有序集。
  - Equalization Redo 设置为 1b 的 TS1 Order Set 或 speed_change 设置为 1b 的 128b/130b EQ TS2 Ordered Set。

公众号：芯片业老相承双

在其他情况下，设备不得在此子状态下或在 Recovery.RcvrCfg 或 Recovery.Equalization 子状态下更改支持的数据速率值。进入该子状态后，successful_speed_negotiation 变量将重置为 0b。

IMPLEMENTATION NOTE

Handling a Request to Advertise 8.0 GT/s Data Rate Identifier

如果没有通告 8.0 GT/s Data Rate Identifiers 的 Upstream Port 接收到具有 8.0 GT/s Data Rate Identifiers 并且在 Recovery.RcvrLock 状态设置 speed_change 字段为 1 的 EQ TS，则表明 Downstream Port 正在尝试切换链路速率达到 8 GT/s，以执行 8.0 GT/s 链路均衡过程。如果由于某种原因 Upstream Port 无法或不愿意切换到在 TS2 有序集中通告 8.0 GT / s Data Rate Identifiers，则在状态机转换到 Recovery.RcvrCfg 时会传输该数据，因此在本轮 Recovery 状态将不执行 8.0 GT/s 的链路均衡过程。这可能会导致 Downstream Port 永久放弃将链路速度更改为 8.0 GT/s 并执行 8.0 GT/s 链路均衡过程的尝试，导致直到链路转换到 Detect 并重新开始一个链路训练之前，链路速率都低于 8.0 GT/s。建议如果 Upstream Port 由于某种暂时原因而无法或不愿意在上述条件下切换为通告 8.0 GT/s Data Rate Identifiers，并且不打算禁止链路以 8.0 GT/s 的速度运行，则建议它在合理的情况下执行以下某个操作：

- 如果 Upstream Port 支持用于执行链路均衡过程的停顿保证机制（Quiesce Guarantee mechanism），转换到 Recovery 并在其发送的 TS 中将 speed_change 位设置为 1b，并通告 8.0 GT/s 的 Data Rate Identifiers。如果在将速度更改为 8.0 GT/s 之后且以 8.0 GT/s 的速度进入 Recovery.RcvrCfg 之前未进入过 Recovery.Equalization（Downstream Port 没有把状态机定向到 Recovery.Equalization），则应在 Recovery.RcvrCfg 中以 8.0 GT/s 的速率发送的 Request Equalization 和 Quiesce Guarantee 字段设置为 1 的 TS2 有序集，以便请求 Downstream Port 启动链路均衡过程。
- 进入 Recovery 状态并通告 8.0 GT/s 的 Data Rate Identifiers，并将 speed_change 位清除为 0b。Downstream Port 随后可以将速度更改为 8.0 GT/s，并执行链路均衡过程，尽管不能保证会这样做。

处理通告 16.0 GT/s（或 32.0 GT/s）Data Rate Identifier 的请求的过程与 8.0 GT/s Data Rate Identifier 以 16.0 GT/s（或 32.0 GT/s）Data Rate Identifier 替换 8.0 GT/s 相似，同理用于 128b/130b EQ TS2 代替 EQ TS。

Upstream Port 必须将其传输的 TS1 有序集的 Selectable De-emphasis 字段设置为与所需的 5.0 GT/s 数据速率下的去加重级别相匹配。Upstream Port 请求一个去加重级别的机制是特定于实现的。还必须注意，由于 TS1 有序集在链路传输中可能有 bit error，所以 Upstream Port 的请求可能无法到达 Downstream Port，因此，当速度切换时，Upstream Port 可能会尝试在随后进入 Recovery 后重新请求期望的去加重级别。如果 Downstream Port 打算在 Recovery.RcvrCfg 中使用 Upstream Port 的去加重信息，则它必须记录在此状态下接收到的 Selectable De-emphasis 字段的值。

Link Control 2 寄存器的 Transmit Margin 字段在进入该子状态时进行采样，并在进入该子状态的 192 ns 之内对发送封装引脚生效，并且一直保持有效，直到之后从 L0，L0s 或 L1 进入该子状态后再重新采样。

- 激活任何无效的通道后，发送器必须等待其 TX 共模电压建立，然后退出电空闲并发送 TS1 有序集。
- 从 L1.2 L1 PM 子状态退出时，允许在 L1.0 期间被动建立共模，并在 Recovery 期间主动建立共模。为了确保在 Recovery.RcvrLock 中已建立共模，下游端口必须维护一个计时器，并且下游端口必须发送 TS2 训练序列，直到下游端口已开始传输 TS1 训练序列为止，直到经过至少 $T_{COMMONMODE}$ 为止，在已配置链路的任何通道上检测到电气空闲退出。参见 Section 5.5.3.3。
- 实现时必须注意，在对端进入 Recovery.RcvrLock 之后，新的 Transmit Margin 字段在 192 ns 内生效后，早期的 bit lock 和 Symbol or Block alignment 之后，电压电平可能会发生变化。在这些情况下，接收器需要重新获取 bit lock 和 Symbol or Block alignment。

  a. Note：对于发起速度更改的一侧，在 L0 或 L1 状态下，directed_speed_change 变量设置为 1b。对于未启动速度更改的一侧，如果接收到的 TS 有序集的 speed_change 设置为 1b，则在此子状态中将 directed_speed_change 设置为 1b。在 Recovery.Speed 子状态下，该位复位为 0b。

  b. 从 L0 进入此子状态后，设备必须接收它收到的所有 TLP 和 DLLP，然后再接收第一个 TS 有序集。如果使用 128b / 130b 编码，则任何接收到的 TLP 和 DLLP 都必须遵守 Section 4.2.2.3 中 128b / 130b 编码的成帧规则。

- 如果在所有已配置的 Lane 上收到 8 个连续的与在这些 Lane 上发送的 Link number 和 Lane number 一致的 TS1 或 TS2 有序集，

并且携带的 speed_change 字段值与 directed_speed_change 变量相等，并且在当前速率为 8.0GT/s 或更高的情况下 EC 字段值为 00b，则下个状态跳转到 Recovery.RcvrCfg 状态。

- 如果 Extended Synch 字段设置为 1b，则发送器在进入 Recovery.RcvrCfg 之前至少必须发送 1024 个连续的 TS1 有序集。
- 如果从 Recovery.Equalization 中进入此子状态，则 Upstream Port 必须评估所有接收到八个 TS1 有序集的通道所接收到的 equalization coefficient 或 preset，并注意它们是否不同于均衡过程中 Phase 2 接收的最终的 equalization coefficient 或 preset。注意：通过设置 TS2 有序集的 Request Equalization 字段，在 Recovery.RcvrCfg 中报告不匹配。

- 否则，在 24ms 超时后，执行以下操作：
  - 如果下面两个条件满足，则状态机转换到 Recovery.RcvrCfg 状态：
    - 在任何已配置的通道上接收到八个连续的 TS1 或 TS2 有序集，且这些有序集的 Link number 与 Lane number 与发送的一致，并且 speed_change 位等于 1b。
    - 当前工作的数据速率大于 2.5 GT/s；或者在发送和接收的连续的八个 TS1 或 TS2 有序集中都设置了 5.0 GT/s 或更高的 data rate identifiers。
  - 否则，如果自 L0 或 L1 进入 Recovery 以来操作速度尚未更改为相互协商的数据速率（即，changed_speed_recovery = 0b），并且当前操作速度大于 2.5 GT/s，则下一个状态是 Recovery.Speed。 离开 Recovery.Speed 后将要使用的新数据速率为 2.5 GT/s。

    Note：这表示链路无法以当前数据速率（大于 2.5 GT/s）运行，并且链路将以 2.5 GT/s 数据速　率运行。
  - 否则，如果自 L0 或 L1 进入 Recovery 以来操作速度已更改为相互协商的数据速率（changed_speed_recovery = 1b；即到此子状态的速率已从 Recovery.Speed 获取），则下一个状态是 Recovery.Speed。 离开 Recovery.Speed 后要使用的新数据速率恢复为从 L0 或 L1 进入 Recovery 时的速度。

    Note：这表明链路无法以新的协商数据速率运行，并且将恢复为从 L0 或 L1 进入 Recovery 时的　数据速率。
  - 否则，如果任何已配置通道已接收到至少一个 Link number 和 Lane number 与正在发送的相匹配 TS1 或 TS2 有序集，且自进入 Recovery 以来工作速率尚未更改为相互协商的数据速率，并且至少满足以下条件之一，则下一个状态是 Configuration 并且 directed_speed_change 变量重置为 0b：
    - directed_speed_change 变量等于 0b，并且接收到的 TS1 或 TS2 有序集上的 speed_change 位等于 0b。
    - 当前的工作数据速率为 2.5 GT/s，而 2.5 GT/s 数据速率是已发送的 TS1 有序集和已接收的 TS1 或 TS2 有序集中最高的通告的数据速率。
  - 否则，下个状态是 Detect 状态。

IMPLEMENTATION NOTE

Example Showing Speed Change Algorithm Between 2.5 GT/s and 5.0 GT/s

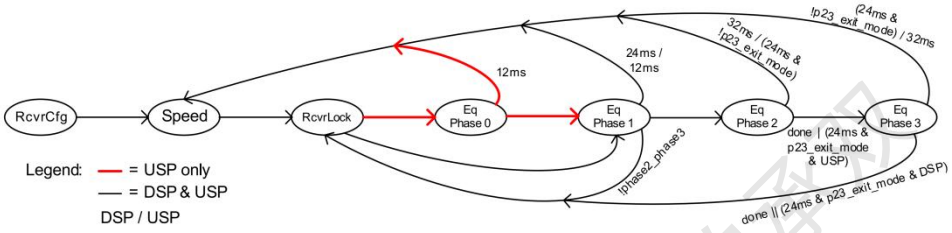
假设链路连接了两个数据速率大于 5.0 GT/s 的组件 A 和 B。链路以 2.5GT/s 的速率训练到 L0。A 准备将链路切换到大于 5.0GT/s 的速率上，于是把 directed_speed_change 设置为 1b，然后从 L0 切换到 Recovery.RcvrLock。A 发送 speed_change 字段置 1 的 TS1 有序集并通告所支持的三种数据速率。B 在 L0 状态下收到第一个 TS1 有序集然后进入 Recovery.RcvrLock 状态。最开始的时候 B 发送 speed_change 字段置 0 的 TS1 有序集。当 B 收到八个连续的 speed_change 字段置 1，B 就开始发送 speed_change 字段置 1 的 TS1 有序集且通告支持的三种数据速率。然后 B 将进入 Recovery.RcvrCfg，之后进入 Recovery.Speed。A 等待从 B 接收八个连续的 speed_change 字段置 1 的 TS1 有序集或 TS2 有序集，然后 A 将进入 Recovery.RcvrCfg，接着进入 Recovery.Speed。A 和 B 都进入 Recovery.Speed 状态且记录了 8.0GT/s 数据速率为它们可工作的最大速率。在 Recovery.Speed 状态 directed_speed_change 变量设置为 0b。当它们从 Recovery.Speed 切换到 Recovery.RcvrLock 状态时，它们会工作在 8.0GT/s 下且发送的 TS1 有序集的

公众号：芯片业者怕承及

speed_change 字段置为 0b。如果两端都稳定工作在 8.0GT/s，它们将转移到 Recovery.RcvrCfg，然后经过 Recovery.Idle 进入 L0。但是，如果 B Symbol lock 失败，它会在 Recovery.RcvrLock 状态超时，然后进入 Recovery.Speed 状态。A 将会进入 Recovery.RcvrCfg，但它会在 B 进入 Recovery.RcvrLock 后 A 收到工作在 8.0GT/s 的 TS1 有序集后看到电气空闲。这会导致 A 进入 Recovery.Speed。当第二次进入 Recovery.Speed 后，两侧就返回到之前进入 Recovery 状态时的工作速率（即 2.5GT/s）。两侧都在 2.5GT/s 的工作速率下从 Recovery 进入 L0。A 将第二次置起 directed_speed_change 变量，请求 8.0GT/s 的数据速率，经过同样的步骤，在建立 8.0GT/s 数据速率时失败并且以 2.5GT/s 的数据速率返回到 L0。但是在做第三次尝试的时候，A 将只通告 2.5GT/s 和 5GT/s 两种速率并且成功建立工作在 5GT/s 的链路且进入 L0。但是，如果因为热插拔事件导致状态机进入 Detect 状态，则必须通告支持从 2.5GT/s 到 8GT/s 的能力。

4.2.6.4.2 Recovery.Equalization

Equalization States in the LTSSM



这张图来自<DesignWare Cores PCI Express Controller Databook> 。

在该子状态，发送器发送与 Configuration 完成时相同的有 Link number 和 Lane number 的 TS1 有序集。

4.2.6.4.2.1 Downstream Lanes

在进入 Recovery.Equalization 子状态时：

- 当前的 phase 就是 Phase 1。
- Link Status 2 寄存器的 Equalization Phase 1 Successful， Equalization Phase 2 Successful， Equalization Phase 3 Successful， Link Equalization Request 和 Equalization Complete 字段，以及 Link Control 3 寄存器的 Perform Equalization 字段都设置为 0b。
- equalization_done_8GT_data_rate 变量设置为 1b。
- start_equalization_w_preset 设置为 0b。

4.2.6.4.2.1.1 Phase 1 of Transmitter Equalization

- 在该 Phase，发送器用 Downstream Port 的 Lane Equalization Control 寄存器的 Transmitter Preset 字段的值来发送 TS1 有序集。在发送的 TS1 有序集中，EC 字段为 01b，Transmitter Preset 就设置为对应 Lane 的寄存器中的值，FS、LF 和 Post-cursor Coefficient 字段的值设置为与 Transmitter Preset 字段相关的值。
- 如果 Downstream Port 需要时间来稳定其接收器逻辑，则进入 Phase 1 后，允许 Downstream Port 最多等待 500 ns，然后评估接收到的 TS1 有序集的信息。
- 如果所有已配置的上接收到两个连续的 EC = 01b 的 TS1 有序集并且 Downstream Port 想执行 Phase 2 和 Phase 3，则下个状态跳转到 Phase 2。
    - 接收器必须完成其比特锁定过程，然后在其接收器引脚上收到第一个有效有序集合的第一比特后的 2 ms 内识别有序集。

公众号·芯片业者相关

- Link Status 2 寄存器的 Equalization Phase 1 Successful 字段设置为 1b。
- 如果 Downstream Port 想要调整 Upstream Port 的发送器的参数，则两个连续的 TS1 有序集中的 FS 和 LF 值用于 Phase 3。

■ 如果所有已配置的上接收到两个连续的 EC = 01b 的 TS1 有序集并且 Downstream Port 不想执行 Phase 2 和 Phase 3，则下个状态跳转到 Recovery.RcvrLock。

- Link Status 2 寄存器的 Equalization Phase 1 Successful，Equalization Phase 2 Successful，Equalization Phase 3 Successful，Link Equalization Request 和 Equalization Complete 字段都设置为 1b。
- Note：转换到 Recovery.RcvrLock 可能用于这种场景：Downstream Port 基于平台和信道特性的原因知道自己不用执行 Phase 2 和 Phase 3 过程。

■ 否则，在 24ms 超时后，状态机跳转到 Recovery.Speed。

- successful_speed_negotiation 设置为 0b。
- Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.2.1.2 Phase 2 of Transmitter Equalization

■ 发送器根据如下规则，按 Lane 独立发送 EC = 10b 与相关 coefficient 设置的 TS1 有序集。

- 如果从进入 Phase 2 开始收到了两个连续的 EC = 10b 的 TS1 有序集，或者收到了两个 preset 或一组 coefficient（比如由 Use Preset 指定）的值与最后收到的两个 EC = 10b 的 TS1 有序集中携带的值不同的 EC = 10b 的 TS1 有序集：
  - ◆ 如果最近两个连续的 TS1 有序集合中要求的 preset 或 coefficient 是合法且受支持的（请参阅 Section 4.2.3），则执行以下操作：
    - ❖ 将发送器设置更改为所需的 preset 或 coefficient，以使新的设置在接收器引脚上收到第二个请求新设置的 TS1 有序设置的最后位置后的 500 ns 内对发送器引脚有效。更改发送器设置不得使发送器引脚上的任何非法电压电平或参数超过 1 ns。
    - ❖ 在传输的 TS1 有序集中体现新的设置，并将 Reject Coefficient Values 字段设置为 0b。
  - ◆ 否则：不用改变发送的设置值，但是需要在 TS1 有序集中体现 preset 或 coefficient 值的请求，并且把 Reject Coefficient Values 设置为 1b。
- 否则，当前的 preset 和 coefficient 值会被用于发送器。

■ 如果所有已配置的上接收到两个连续的 EC = 11b 的 TS1 有序集，则下个状态跳转到 Phase 3。

- Link Status 2 寄存器的 Equalization Phase 2 Successful 字段设置为 1b。

■ 否则，在 32ms 超时后状态机跳转到 Recovery.Speed 状态。这个超时容差为-0ms~+4ms。

- successful_speed_negotiation 设置为 0b。
- Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.2.1.3 Phase 3 of Transmitter Equalization

■ 发送器发送 EC = 11b 的 TS1 有序集。

■ Port 必须在每个通道上独立评估并获得最佳设置。根据 Section 4.2.3 和 Section 4.3 中的规则评估合法的新 preset 或 coefficient 的设置：

公众号·芯片业者相聚双

- 通过将 Transmitter Preset 字段设置为所需的值并将 Use Preset 字段设置为 1b 来请求一个新的 preset。或者，通过将 Pre-cursor、Cursor 和 Post-Cursor Coefficient 字段设置为所需的值，并将 Use Preset 字段设置为 0b，以请求一组新的 coefficient。发出请求后，必须连续请求至少 1 µs 或直到请求评估完成为止（以较晚者为准）。

- 等待所需的时间（500 ns 加上往返延迟，包括通过 Downstream Port 的逻辑延迟），以确保（如果接受）Upstream Port 正在使用请求的设置进行传输。获取 Block Alignment，然后评估传入的有序集。Note：由于在切换到请求的设置期间传入的比特流可能是非法的，因此 Downstream Port 可能忽略在此等待期间收到的任何内容。因此，在此等待周期完成之后需要验证 Block Alignment。

- 如果接收到两个连续的 TS1 有序集，且 Transmitter Preset 字段（用于 preset 请求）或 Pre-cursor、Cursor 和 Post-Cursor 字段（用于 coefficient 请求）与请求的字段相同，并且 Reject Coefficient Values 字段设置为 0b，则请求的设置被接受并且需要进行评估。如果接收到两个连续的 TS1 有序集，且 Transmitter Preset 字段或 Pre-cursor、Cursor 和 Post-Cursor 字段与请求的字段相同，并且 Reject Coefficient Values 字段设置为 1b，则请求的设置被拒绝，建议 Downstream Port 请求其他 preset 或 coefficient 的组合。但是，允许 Downstream Port 在不发出任何其他请求的情况下退出此 Phase。Downstream Port 负责根据其评估标准和要求将其发送的 TS1 有序集中的 Reset EIEOS Interval Count 字段设置为 1b。

IMPLEMENTATION NOTE

Reset EIEOS and Coefficient/ Preset Requests

当端口想要更长的 PRBS 码型时，可以将 Reset EIEOS Interval Count 字段设置为 1b，然后在需要获取块对齐时将其清除。

在此 Phase 发送的所有 TS1 有序集都是请求。第一个请求可以是新的 preset 或 coefficient 请求，也可以是通过反映两个连续的 TS1 有序集（EC = 11b）中接收到的设置来保持当前链路伙伴发射机设置的请求，这些设置导致切换到 Phase 3。

- 从发送请求到评估完成，每个 preset 或 coefficient 请求所花费的总时间必须少于 2 ms。在优化的最后阶段需要更长评估时间的实现可能会继续请求相同的 preset 或 coefficient 设置（超过 2 ms 的限制），但在此阶段必须遵守 24 ms 的超时时间，并且此例外不得超过两次。如果请求者无法在超时时间内接收到有序集，则可以假定所请求的设置在该通道中不起作用。

- 所有新的 preset 或 coefficient 设置必须同时显示在所有已配置的通道上。如果任何给定的通道不想在那时更改设置，则可以继续发送当前的 preset 或 coefficient 作为其新值。

- 如果所有已配置的 Lane 都工作在最佳设置下则下个状态为 Recovery.RcvrLock。
  - Link Status 2 寄存器的 Equalization Phase 3 Successful 和 Equalization Complete 字段设置为 1b。
- 否则，在 24ms 超时后状态机跳转到 Recovery.Speed 状态。这个超时容差为-0ms~+2ms。
  - successful_speed_negotiation 设置为 0b。
  - Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.2.2 Upstream Lanes

在进入 Recovery.Equalization 子状态时：

- 当前的 phase 就是 Phase 0。
- Link Status 2 寄存器的 Equalization Phase 1 Successful，Equalization Phase 2 Successful，Equalization Phase 3 Successful，Link Equalization Request 和 Equalization Complete 字段都设置为 0b。
- equalization_done_8GT_data_rate 变量设置为 1b。
- start_equalization_w_preset 设置为 0b。

公众号：芯片业者相承双

4.2.6.4.2.2.1 Phase 0 of Transmitter Equalization

- 在最近一次从 2.5 GT/s 或 5.0 GT/s 数据速率转换到 8.0 GT/s 数据速率期间，发送器使用 EQ TS2 有序集中接收的发送器 preset 指定的发送器设置发送的 TS1 有序集。在 TS1 有序集中，EC 字段设置为 00b，每个通道的 Transmitter Preset 字段设置为在 EQ TS2 有序集中接收的值，以及 Pre-cursor Coefficient、Cursor Coefficient 和 Post-cursor Coefficient 字段设置为与 Transmitter Preset 字段相对应的值。
  - 如果接收到的 EQ TS2 有序集合的通道的 Transmitter Preset 为保留值或不支持的值：意味着在发送的 TS1 有序集合中，Transmitter Preset 字段设置为接收到的 Transmitter Preset，但 Coefficient 字段设置为与由特定选择的与特定实现有关的 Transmitter Preset 值。
  - 如果通道没有接收到 EQ TS2 有序集：意味着在发送的 TS1 有序集中，Transmitter Preset 字段没有定义，且 Coefficient 字段设置为与由特定选择的与特定实现有关的 Transmitter Preset 值。
- 如果 Upstream Port 需要时间来稳定其接收器逻辑，则允许其在进入 Phase 0 之后等待长达 500 ns 的时间，然后才能评估 TS1 有序集的接收器信息。
- 如果在所有已配置的 Lane 上收到两个连续的 EC = 01b 的 TS1 有序集，则下个 phase 是 Phase 1。
  - 接收器必须完成其比特锁定过程，然后在其接收器引脚上收到第一个有效有序集的第一比特后的 2 ms 内识别有序集。
  - 如果 Upstream Port 想要调整 Downstream Port 的发送器的参数，则两个连续的 TS1 有序集中的 FS 和 LF 值用于 Phase 2。
- 否则，在 12ms 超时后进入 Recovery.Speed 状态。
  - successful_speed_negotiation 变量设置为 0b。
  - Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.2.2.2 Phase 1 of Transmitter Equalization

- 发送器使用在 Phase 0 中确定的发送器设置发送 TS1 有序集。在 TS1 有序集中，EC 字段设置为 01b，每个通道的 FS，LF 和 Post-cursor Coefficient 字段设置为该 Lane 的当前发送器的设置值。
- 如果在所有已配置的 Lane 上收到两个连续的 EC = 10b 的 TS1 有序集，则下个 Phase 为 Phase 2。
  - Link Status 2 寄存器的 Equalization Phase 1 Successful 字段设置为 1b。
- 如果在所有已配置的 Lane 上收到两个连续的 EC = 00b 的 TS1 有序集，则下个状态为 Recovery.RcvrLock。
  - Link Status 2 寄存器的 Equalization Phase 1 Successful 字段和 Equalization Complete 字段设置为 1b。
- 否则，在 12ms 超时后进入 Recovery.Speed 状态。
  - successful_speed_negotiation 变量设置为 0b。
  - Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.2.2.3 Phase 2 of Transmitter Equalization

- 发送器发送 EC = 10b 的 TS1 有序集。
- Port 必须在每个通道上独立评估并获得最佳设置。根据 Section 4.2.3 和 Section 4.3 中的规则评估合法的新 preset 或 coefficient 的设置：
  - 通过将 Transmitter Preset 字段设置为所需的值并将 Use Preset 字段设置为 1b 来请求一个新的 preset。或者，通过将

公众号·芯片业者相互汉

Pre-cursor、Cursor 和 Post-Cursor Coefficient 字段设置为所需的值，并将 Use Preset 字段设置为 0b，以请求一组新的 coefficient。发出请求后，必须连续请求至少 1 µs 或直到请求评估完成为止（以较晚者为准）。

- 等待所需的时间（500 ns 加上往返延迟，包括通过 Downstream Port 的逻辑延迟），以确保（如果接受）Upstream Port 正在使用请求的设置进行传输。获取 Block Alignment，然后评估传入的有序集。Note：由于在切换到请求的设置期间传入的比特流可能是非法的，因此 Downstream Port 可能忽略在此等待期间收到的任何内容。因此，在此等待周期完成之后需要验证 Block Alignment。

- 如果接收到两个连续的 TS1 有序集，且 Transmitter Preset 字段（用于 preset 请求）或 Pre-cursor、Cursor 和 Post-Cursor 字段（用于 coefficient 请求）与请求的字段相同，并且 Reject Coefficient Values 字段设置为 0b，则请求的设置被接受并且需要进行评估。如果接收到两个连续的 TS1 有序集，且 Transmitter Preset 字段或 Pre-cursor、Cursor 和 Post-Cursor 字段与请求的字段相同，并且 Reject Coefficient Values 字段设置为 1b，则请求的设置被拒绝，建议 Downstream Port 请求其他 preset 或 coefficient 的组合。但是，允许 Downstream Port 在不发出任何其他请求的情况下退出此 Phase。Downstream Port 负责根据其评估标准和要求将其发送的 TS1 有序集中的 Reset EIEOS Interval Count 字段设置为 1b。

IMPLEMENTATION NOTE

Reset EIEOS and Coefficient/ Preset Requests

当端口想要更长的 PRBS 码型时，可以将 Reset EIEOS Interval Count 字段设置为 1b，然后在需要获取块对齐时将其清除。

在此 Phase 发送的所有 TS1 有序集都是请求。第一个请求可以是新的 preset 或 coefficient 请求，也可以是通过反映两个连续的 TS1 有序集（EC = 10b）中接收到的设置来保持当前链路伙伴发射机设置的请求，这些设置导致切换到 Phase 3。

- 从发送请求到评估完成，每个 preset 或 coefficient 请求所花费的总时间必须少于 2 ms。在优化的最后阶段需要更长评估时间的实现可能会继续请求相同的 preset 或 coefficient 设置（超过 2 ms 的限制），但在此阶段必须遵守 24 ms 的超时时间，并且此例外不得超过两次。如果请求者无法在超时时间内接收到有序集，则可以假定所请求的设置在该通道中不起作用。

- 所有新的 preset 或 coefficient 设置必须同时显示在所有已配置的通道上。如果任何给定的通道不想在那时更改设置，则可以继续发送当前的 preset 或 coefficient 作为其新值。

■ 如果所有已配置的 Lane 都工作在最佳设置下则下个 phase 为 Phase 3。

- Link Status 2 寄存器的 Equalization Phase 2 Successful 字段设置为 1b。

■ 否则，在 24ms 超时后状态机跳转到 Recovery.Speed 状态。这个超时容差为-0ms~+2ms。

- successful_speed_negotiation 设置为 0b。

- Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.2.2.4 Phase 3 of Transmitter Equalization

■ 发送器发送 EC = 11b 且有如下规则设置的 coefficient 的 TS1 有序集：

- 如果从进入 Phase 3 开始收到了两个连续的 EC = 11b 的 TS1 有序集，或者收到了两个 preset 或 coefficient（比如由 Use Preset 指定）的值与最后收到的两个 EC = 11b 的 TS1 有序集中携带的值不同的 EC = 11b 的 TS1 有序集：

  ◆ 如果最近两个连续的 TS1 有序集合中要求的 preset 或 coefficient 是合法且受支持的（请参阅 Section 4.2.3），则执行以下操作：

    ❖ 将发送器设置更改为所需的 preset 或 coefficient，以使新的设置在接收器引脚上收到第二个请求新设置的 TS1 有序设置的最后位置后的 500 ns 内对发送器引脚有效。更改发送器设置不得使发送器引脚上的任何非

法电压电平或参数超过 1 ns。

- ❖ 在传输的 TS1 有序集中体现新的设置，并将 Reject Coefficient Values 字段设置为 0b。

- ◆ 否则：不用改变发送的设置值，但是需要在 TS1 有序集中体现 preset 或 coefficient 值的请求，并且把 Reject Coefficient Values 设置为 1b。

- 否则，当前的 preset 和 coefficient 值会被用于发送器。

- 如果所有已配置的 Lane 都收到两个连续的 EC = 00b 的 TS1 有序集，则下个状态为 Recovery.RcvrLock。
  - Link Status 2 寄存器的 Equalization Phase 3 Successful 和 Equalization Complete 字段设置为 1b。
- 否则，在 32ms 超时后状态机跳转到 Recovery.Speed 状态。这个超时容差为-0ms~+4ms。
  - successful_speed_negotiation 设置为 0b。
  - Link Status 2 寄存器的 Equalization Complete 字段设置为 1b。

4.2.6.4.3 Recovery.Speed

- 发送器进入电空闲状态，并一直保持此状态，直到接收器的 Lane 进入电空闲状态为止，然后在成功的速度协商的情况下（例如，success_speed_negotiation = 1b）停留至少 800 ns，或者在不成功的速度协商中至少停留 6 µs（即，success_speed_negotiation = 0b），但在那里停留的时间不得超过 1 ms。仅在接收器的 Lane 进入电气空闲状态后，才允许将操作频率更改为新的数据速率。如果协商的数据速率为 5.0 GT/s，并且在 full swing 模式下运行，则在 select_deemphasis = 0b 的情况下选择-6 dB 的去加重等级，在 select_deemphasis = 1b 的情况下选择-3.5 dB 的去加重等级。请注意，如果链路已经以两个端口支持的最高数据速率运行，则依然会执行 Recovery.Speed，但不会更改数据速率。

  如果当前数据速率为 2.5 GT/s 或 8.0 GT/s，则必须在进入电空闲之前发送 EIOS。如果当前数据速率为 5.0 GT/s，则在进入电空闲之前必须发送两个连续的 EIOS。

  在此状态下，DC 共模电压不需要符合规范定义。

  如果在任何已配置的通道上接收到 EIOS，或者如 Section 4.2.4.3 中所述检测到/推断出电气空闲，则通道上存在电气空闲条件。

  - 在成功进行速度协商（即，success_speed_negotiation = 1b）后进入此子状态时，如果在指定的时间间隔（Table 4-11 中定义）内未在任何已配置的通道中接收到 TS1 或 TS2 有序集，则可能会在接收器通道上推断出电气空闲状态。（这涵盖了链路正常运行并且双方都已成功接收 TS 有序集的情况。因此，在指定间隔内缺少 TS1 或 TS2 有序集可以解释为进入了电气空闲状态。）
  - 如果在 Table 4-11 中指定的时间间隔内未在任何已配置的通道中至少检测到一次电气空闲退出，则在速度协商失败（即，success_speed_negotiation = 0b）之后进入此子状态。（这涵盖了以下情况：至少一方在接收由另一代理发送的 TS 有序集时遇到问题，因此在较长的时间间隔内没有退出电气空闲可以被视为等同于进入电气空闲。）
- 若满足上述在发送器的 Lane 不再需要处于电气空闲状态的条件，下个状态就跳转到 Recovery.RcvrLock。
  - 如果在成功进行速度协商后从 Recovery.RcvrCfg 中进入此子状态（即，success_speed_negotiation = 1b），则新的数据速率将在所有已配置的通道上更改为链路两侧通告的最高公共数据速率。changed_speed_recovery 变量设置为 1b。
  - 否则，如果是第二次从 L0 或 L1 进入 Recovery 之后进入此子状态（即，changed_speed_recovery = 1b），则新数据速率将是 LTSSM 从 L0 或 L1 进入 Recovery 时的数据速率。 changed_speed_recovery 变量将重置为 0b。
  - 否则，新数据速率将为 2.5 GT/s。changed_speed_recovery 变量保持为 0b。

公众号·芯片业务拓展双

Note：这代表以下情况：L0 中的操作频率大于 2.5 GT/s，并且一侧无法以该频率操作，然后在 Recovery.RcvrLock 超时的情况下，它第一次从 L0 或 L1 进入该子状态。

- 48ms 超时后进入 Detect 状态。
  - Note：正常情况下不会执行这个操作。
- directed_speed_change = 0b。Link Status 寄存器的 Current Link Speed 字段要反映当前的工作速率。
  - 在发生链路带宽变化时，如果将 recovery_speed_negotiation 设置为 1b，并且在 Recovery.RcvrCfg 中接收到八个连续的 Autonomous Change 设置为 1b 的 TS2 有序集，或出于自主原因由 Downstream Port 发起速度更改（非可靠因素且不是因为 Link Retrain 导致），则 Link Status 寄存器的 Link Autonomous Bandwidth Status 字段要设置为 1b。
  - 否则，在在发生链路带宽变化时，Link Status 寄存器的 Link Bandwidth Management Status 字段要设置为 1b。

4.2.6.4.4 Recovery.RcvrCfg

在这个子状态，发送器在所有已配置的 Lane 上发送在退出 Configuration 状态时使用的 Link number 与 Lane number 值的 TS2 有序集。如果 directed_speed_change 已经为 1b，则 TS2 有序集的 speed_change 字段也要设置为 1b。TS2 中携带的 N_FTS 值要反映在当前速率下快速恢复需要的 FTS 数。

如果下述条件全满足，Downstream Port 就必须发送这样的 EQ TS2 有序集：其 Transmitter Preset 和 Receiver Preset Hint 字段的值由 Upstream Port 的 Transmitter Preset 和 Receiver Preset Hint 字段的值确定，而这些值又来自于各个 Lane 的 Equalization Control 寄存器。

a. Downstream Port 会在 Recovery.RcvrLock 状态通告对 8.0GT/s 数据速率的支持，而 Upstream Port 则在退出 Detect 状态后、在 Configuration.Complete 或者 Recovery.RcvrCfg 状态中通告对 8.0GT/s 数据速率的支持，并且会在进入此子状态之前在任意已配置的 Lane 上收到八个连续的 speed_change 字段设置为 1 的 TS1 或 TS2 有序集。

b. 如果要执行此操作，equalization_done_8GT_data_rate 变量设置为 0b。

如果当前速率是 8.0GT/s，并且在最近执行的均衡过程中检测到一个问题，则 Upstream Port 会把有序集中的 Request Equalization 字段设置为 1b。否则，Request Equalization 设置为 0b。如果希望重新执行均衡过程，则 Downstream Port 或 Upstream Port 还需要把 Request Equalization 和 Quiesce Guarantee 字段设置为 1b，参阅 Section 4.2.3。

在进入此子状态时，start_equalization_w_preset 设置为 0b。

- 在进入此子状态时，Downstream Port 必须把 select_deemphasis 变量设置为 Link Control 2 寄存器 Selectable De-emphasis 字段的值，或采用某种特定机制设置 select_deemphasis 的值，比如可以使用 Upstream Port 在八个连续的 TS1 有序集中携带的值。一个通告支持 5GT/s 数据速率的 Downstream Port 必须在 TS2 有序集中传输与 select_deemphasis 值相同的 Selectable De-emphasis 字段。一个 Upstream Port 如果因自主原因要改变链路带宽，就必须传输 Autonomous Change 字段设置为 1b 的 TS2 有序集。
  - 对于支持 Link width upconfigure 的设备，如果 directed_speed_change 为 0b，则建议在此子状态、Recovery.Idle 子状态、Configuration.Linkwidth.Start 子状态开启当前 inactive Lane 集合上的电气空闲检测电路。这样做是为了在 Link upconfigure 期间，不启动 upconfiguration 的那一侧不会在 Configuration.Linkwidth.Start 子状态期间丢失发送的第一个 EIEOS。
- 如果下述条件全满足，则状态机跳转到 Recovery.Speed 状态：
  - 满足下述其中一个条件：
    1) 如果使用 8b/10b 编码，则需要在任意已配置的通道上收到八个连续的 Data Rate Identifier 相同的 TS2 有序集，并

且其 speed_change 字段设置为 1b，且八个连续的 TS2 有序集是标准 TS2 有序集。

2) 在所有已配置的通道上收到八个连续的 Data Rate Identifier 相同的 TS2 有序集，且其 speed_change 字段设置为 1b。

3) 在任意已配置的通道上收到八个连续的 Data Rate Identifier 相同的 TS2 有序集，并且其 speed_change 字段设置为 1b，并且收到这 8 个连续的 TS2 有序集之后超过 1ms。

- 当前数据速率大于 2.5 GT/s 或者在发送和接收（连续八个）的 TS2 有序集中设置了大于 2.5 GT/s 的 Data Rate Identifiers 字段。
- 对于 8b/10b 编码，在同一个已配置的通道中接收到一个 speed_change 设置为 1b 的 TS2 有序集后，至少发送了 32 个不被 EIEOS 中断的 speed_change 设置为 1b 的 TS2 有序集。 对于 128b/130b 编码，在同一个已配置的通道中接收到一个将 speed_change 设置为 1b 的 TS2 有序集后，至少发送 128 个 speed_change 位设置为 1b 的 TS2 有序集。

在接收到的 speed_change 字段设置为 1b 的八个连续 TS2 有序集上通告的数据速率表明了另一个端口可以支持的数据速率。这些接收到的八个连续的 TS2 有序集中的 Autonomous Change 字段由 Downstream Port 给出，以便在 Recovery.Speed 子状态下由 Link Status 寄存器记录。Upstream Ports 必须用 select_deemphasis 变量记录在这八个连续的 TS2 有序集中通告的 Selectable De-emphasis 字段的值。在 Recovery.Speed 状态切换的新速率是两个 Port 支持的最高的 data rate。对于 Upstream Ports，如果这八个连续的 TS2 有序集是 EQ TS2 有序集，它必须把 start_equalization_w_preset 变量设置为 1b 并且把 Lane Equalization Control 寄存器的 Upstream Port Transmitter Preset 和 Upstream Port Receiver Preset Hint 字段的值更新为这些 EQ TS2 有序集中携带的值。在执行链路均衡之前，首次以 8.0 GT/s 的速度运行时，任何未接收到满足此条件的 EQ TS2 有序集的已配置通道将使用与实现相关的 preset 值。如果 equalization_done_8GT_data_rate 变量为 0b，或因直接定向的原因，则 Downstream Port 必须将 start_equalization_w_preset 变量设置为 1b。successful_speed_negotiation 变量要设置为 1b。请注意，如果链路已经以两个端口支持的最高数据速率运行，则将执行 Recovery.Speed，但不会更改数据速率。如果使用 128b/130b 编码，并且在八个连续的 TS2 有序集中将 Request Equalization 位设置为 1b，则设备必须将其与 Quiesce Guarantee 字段的值一起记录下来，并遵循 Section 4.2.3 中的规则。

- 如果下述两个条件全为真，则状态机跳转到 Recovery.Idle 状态：
  - 在所有已配置的 Lane 上收到 8 个连续的与在该 Lane 上发送的 Link number 与 Lane number 值相匹配、 Data Rate Identifiers 相同的 TS2 有序集，并且满足以下两个条件中的一个：
    - 收到的八个连续的 TS2 有序集的 speed_change 字段为 0b。
    - 当前数据速率为 2.5 GT/s，并且没有在接收到的八个连续 TS2 有序集中设置 5.0 GT/s 或更高的 Data Rate Identifier，或者没有在发送的 TS2 有序集中设置 5.0 GT/s 或更高的 Data Rate Identifier。
  - 在收到一个 TS2 有序集后发送完 16 个连续的不被 EIEOS 中断的 TS2 有序集。在进入 Recovery.Idle 时 changed_speed_recovery 和 directed_speed_change 变量设置为 0b。
  - 如果 N_FTS 发生变化，则新的 N_FTS 用于之后的 L0s 状态。
  - 当使用 8b/10b 编码，Lane-to-Lane de-skew 必须在退出 Recovery.RcvrCfg 时完成。
  - 设备必须记录在发生此状态转换时在任意 Lane 上发送的八个连续的 TS2 有序集中通告的 Data Rate Identifiers 值。这个值可能会覆盖之前记录的值。
  - 当使用 128b/130b 编码且八个连续的 TS2 有序集中的 Request Equalization 字段为 1b，则设备必须要记录此值并按 Section 4.2.3 的规则来处理。
- 如果在任意 Lane 上收到了八个连续的 TS1 有序集中的 Link number 与 Lane number 与在该 Lane 上发送的不一致，并且在收到一个 TS1 有序集之后发送完 16 个 TS2 有序集，并且下述任一条件满足，则状态机跳转到 Configuration 状态：
  - 收到的八个连续的 TS1 有序集的 speed_change 字段为 0b。

- 当前数据速率为 2.5 GT/s，并且没有在接收到的八个连续 TS1 有序集中设置 5.0 GT/s 或更高的 Data Rate Identifier，或者没有在发送的 TS2 有序集中设置 5.0 GT/s 或更高的 Data Rate Identifier。
- 如果 N_FTS 发生变化，则新的 N_FTS 用于之后的 L0s 状态。

在状态机切换到 Configuration 时 changed_speed_recovery 和 directed_speed_change 变量设置为 0b。

- 如果从 L0 或 L1 进入 Recovery 以来，工作速率没有变化到相互协商的速率上（changed_speed_recovery = 0b），并且现在的工作速率大于 2.5GT/s，并且检测到一个 EIOS 或者在任意已配置的 Lane 上推导/检测到电气空闲条件，并且在进入此子状态后在未配置的 Lane 上收到一个 TS2 有序集，则下个状态转换到 Recovery.Speed。退出 Recovery.Speed 后的工作速率将会是 2.5GT/s。

  如 Section 4.2.4.3 所述，如果在 Table 4-11 中指定的时间间隔内未收到 TS1 或 TS2 有序集，则可以推断出电气空闲状态。

- 如果从 L0 或 L1 进入 Recovery 以来，工作速率已经变化到相互协商的速率上（changed_speed_recovery = 1b），并且检测到一个 EIOS 或者在任意已配置的 Lane 上推导/检测到电气空闲条件，并且在进入此子状态后在未配置的 Lane 上收到一个 TS2 有序集，则下个状态转换到 Recovery.Speed。退出 Recovery.Speed 后的工作速率将会恢复到从 L0 或 L1 进入 Recovery 时的速率。

  如 Section 4.2.4.3 所述，如果在 Table 4-11 中指定的时间间隔内未收到 TS1 或 TS2 有序集，则可以推断出电气空闲状态。

  这种切换意味着另一端无法以其运行的速度实现符号锁定或块对齐。因此，双方将恢复到 2.5 GT/s 的运行速度，并且两个设备都不会在不退出 Recovery 状态的情况下再次尝试更改速度(意思是如果还需要尝试更改速度，则必须退出 Recovery 状态)。还应注意，即使此处涉及速度更改，但 changed_speed_recovery 依然为 0b。

- 在 48ms 超时后会执行以下操作：
  - 如果当前速率为 2.5GT/s 或者 5GT/s，则下个状态跳转到 Detect 状态。
  - 如果 idle_to_rlock_transitioned < FFh 且当前工作速率为 8GT/s，则下个状态跳转到 Recovery.Idle。
    - 在进入 Recovery.Idle 时，changed_speed_recovery 和 directed_speed_change 变量都重置为 0b。
  - 否则下个状态跳转到 Detect 状态。

4.2.6.4.5 Recovery.Idle

- 在直接定向作用下下个状态跳转到 Disabled 状态。
  - Note：“直接定向”的意思是 Downstream Port 或 crosslink Port 在上层逻辑中把发送的 TS1 或 TS2 有序集中的 Disable Link 字段置为 1b。
- 在直接定向作用下下个状态跳转到 Hot Reset 状态。
  - Note：“直接定向”的意思是 Downstream Port 或 crosslink Port 在上层逻辑中把发送的 TS1 或 TS2 有序集中的 Hot Reset 字段置为 1b。
- 在直接定向作用下下个状态跳转到 Configuration 状态。
  - Note：“直接定向”的意思是 Port 在上层逻辑中进行重新配置链路。
- 在直接定向作用下下个状态跳转到 Loopback 状态，此时发送器作为一个 loopback master，这个根特定实现有关。
  - Note：“直接定向”的意思是 Port 在上层逻辑中把发送的 TS1 或 TS2 有序集中的 Loopback 字段置为 1b。

公众号·芯片业者相谋双

- 如果在任意已配置的 Lane 上收到两个连续的 Disable Link 字段为 1b 的 TS1 有序集，则下个状态跳转到 Disabled 状态。
  - Note：这个行为仅应用于 Upstream Port 和 crosslink Port。
- 如果在任意已配置的 Lane 上收到两个连续的 Hot Reset 字段为 1b 的 TS1 有序集，则下个状态跳转到 Hot Reset 状态。
  - Note：这个行为仅应用于 Upstream Port 和 crosslink Port。
- 如果在任意已配置的 Lane 上收到两个连续的 Lane number 字段为 PAD 的 TS1 有序集，则下个状态跳转到 Configuration 状态。
  - Note：确保实现转换到 Configuration 以更改链路配置功能的端口在所有通道上发送 Lane number 设置为 PAD 的 TS1 有序集。
  - Note：建议 LTSSM 使用此转换来增大/缩小链接宽度，以减少更改链接宽度所需的时间。
- 如果在任意已配置的 Lane 上收到两个连续的 Loopback 字段为 1b 的 TS1 有序集，则下个状态跳转到 Loopback 状态。
  - Note：收到 Loopback 字段为 1b 的 TS1 有序集的设备作为 loopback slave。
- 当使用 8b/10b 编码，则发送器在所有已配置的 Lane 上发送 Idle data。当使用 128b/130b 编码，则发送器在所有已配置的 Lane 上发送 SDS 有序集以开始一个数据流，然后发送 Idle data Symbols。Note：在 Lane 0 上发送的第一个 Idle data Symbol 为数据流的第一个 Symbol。
  - 如果定向到其他状态，则在转换到其他状态之前不用发送 Idle Symbol，也不用发送 128b/130b 编码（Disabled，Hot Reset，Configuration，or Loopback）。

IMPLEMENTATION NOTE

EDS Usage

当使用 128b/130b 编码，在 Configuration 或 Loopback 或 Hot Reset 或 Disabled 状态时，如果数据流处于有效状态（即已发送 SDS 有序集），则必须发送 EDS。未启动 Link Upconfigure 的一方在收到 TS1 有序集时可能已经发送了 SDS 和数据流（Logical IDL）。在这种情况下，它将在发送配置中的 TS1 有序集之前在有效的通道集中发送 EDS。

- 当使用 8b/10b 编码，如果在所有已配置的 Lane 上收到 8 个连续的 Symbol Times 的 Idle data 并且在收到一个 Idle data Symbol 后发完 16 个 Idle data Symbol，则状态机跳转到 L0。
  - 如果在最后一次从 Recovery 或 Configuration 状态转换到 L0 之前软件把 Link Control 寄存器的 Retrain Link 字段写 1b，Downstream Port 就必须把 Link Status 寄存器的 Link Bandwidth Management Status 字段置为 1b。
- 当使用 128b/130b 编码，如果在所有已配置的 Lane 上收到 8 个连续的 Symbol Times 的 Idle data 并且在收到一个 Idle data Symbol 后发完 16 个 Idle data Symbol，并且此子状态不是从 Recovery.RcvrCfg 切换过来的，则状态机跳转到 L0。
  - Idle data Symbol 必须在 Data Block 中接收。
  - Lane-to-Lane de-skew 必须在数据流处理开始之前完成。
  - 如果在最后一次从 Recovery 或 Configuration 状态转换到 L0 之前软件把 Link Control 寄存器的 Retrain Link 字段写 1b，Downstream Port 就必须把 Link Status 寄存器的 Link Bandwidth Management Status 字段置为 1b。
  - 在转换到 L0 状态时 idle_to_rlock_transitioned 变量要置为 00h。
- 否则，在 2ms 超时后执行下述操作：
  - 如果 idle_to_rlock_transitioned < FFh，则下个状态为 Recovery.RcvrLock。
    - 如果 data rate 是 8.0GT/s，则每次转换到 Recovery.RcvrLock，idle_to_rlock_transitioned 变量递增 1。

公众号·芯片业者相承双

- 如果 data rate 是 5.0GT/s（或，如果支持 2.5GT/s），则在转换到 Recovery.RcvrLock 时 idle_to_rlock_transitioned 变量设置为 FFh。

- 否则，下个状态为 Detect。

4.2.6.5 L0

L0 状态是正常的数据传输状态。

- LinkUp = 1b 。
  - 在接收 STP 和 SDP 时，idle_to_rlock_transitioned 要复位为 00h。
- 如果下述两个条件其中一个满足，状态机就跳转到 Recovery 状态：1）当两侧设备都支持比 2.5GT/s 更高的 data rate 且链路处于 DL_Active 时，上层逻辑会直接发起链路速率切换（directed_speed_change = 1b）；2）当两侧设备都支持 8GT/s 且链路要执行一个 8GT/s 的均衡过程时，上层逻辑会直接发起链路速率切换（directed_speed_change = 1b）。然后还要把 changed_speed_recovery 复位成 0b。
  - 对于 Upstream Port，如果 Downstream Port 从来没有在 Configuration.Complete 或 Recovery.RcvrCfg 状态通告比 2.5GT/s 更高的数据速率，directed_speed_change 就不能设置为 1b。
  - 对于 Downstream Port，如果 Upstream Port 从来没有在 Configuration.Complete 或 Recovery.RcvrCfg 状态通告比 2.5GT/s 更高的数据速率，directed_speed_change 就不能设置为 1b。如果支持比 2.5GT/s 更高的数据速率，则如果 Link Control 寄存器的 Retrain Link 字段设置为 1b，并且 Link Control 2 寄存器中的 Target Link Speed 字段不等于当前的链路速度，则 Downstream Port 必须将 directed_speed_change 变量设置为 1b。
  - 即使链路不处于 DL_Active 状态，如果对端通过 TS1 发出请求，支持比 2.5GT/s 更高的数据速率的 Port 也必须参与链路速率切换过程。
- 如果要发生链路宽度变化，则状态机会跳转到 Recovery 状态。
  - 如果对端端口在 Configuration 状态没有通告链路宽度的 upconfigure 能力，或者链路在进入 L0 时已经是最大宽度的链路，上层逻辑就不能发出增加链路宽度的操作。
  - 正常情况下，出于可靠性原因，如果将 upconfigure_capable 重置为 0b，则上层逻辑不会减小链路宽度，因为如果 upconfigure_capable 为 0b，则链路将无法返回到最初的宽度。如果将 Link Control 寄存器中的 Hardware Autonomous Width Disable 字段设置为 1b，则端口不得出于可靠性以外的其他原因启动减小链路宽度的操作。
  - 减少或增加链路宽度是跟具体实现有关的问题。
- 如果在此状态下收到了一个 TS1 或 TS2 有序集，或在 128b/130b 编码时收到了一个 EIEOS，则状态机会跳转到 Recovery 状态。
- 状态机可以在此状态下直接定向到 Recovery 状态。如果在没有接收到 EIOS 的 Lane 上检测到了电气空闲，则端口有可能转换到 Recovery 状态，也有可能停留在 L0 状态。这种没有接收到 EIOS 的 Lane 上检测到了电气空闲的情况可能会导致 error，所以 Port 可能会直接切换到 Recovery 状态。
  - 如 Section 4.2.4.3 所述，可以在以下任一条件下推断所有通道的电气空闲条件：（i）在任何 128µs 窗口中均不存在 Flow Control Update DLLP，（ii） 在任何 128µs 窗口中的任何已配置通道中不存在 SKP 有序集，或者（iii）在任何 128µs 窗口中的任何已配置通道中不存在 Flow Control Update DLLP 或 SKP 有序集。
  - 这里的"直接定向"的意思是上层逻辑直接导致状态机进入 Recovery 状态，包括把 Link Control 寄存器的 Retrain Link 字段设置为 1b 这种方式。

公众号·芯片业余自承双

- 发送器可以在此状态下发送 TLP 和 DLLP。

- 如果发送器实现了 L0s 功能，则发送器可以直接转移到 L0s 状态，（注意只是发送器进入 L0s）。参阅 Section 4.2.6.2.2。
  - Note：这里的"直接"指的是上层逻辑直接导致状态机进入 L0s 状态，参阅 Section 4.2.6.6.2 。
  - Note：有一个点需要注意：就是在此情况下 TX 和 RX 可能处于不同的 LTSSM 状态（如 Tx 处于 L0s，而 Rx 处于 L0）。
- 如果接收器实现了 L0s 功能，那么在任意 Lane 上收到了 EIOS，则接收器直接转移到 L0s 状态，（注意只是接收器进入 L0s）。参阅 Section 4.2.6.2.1。
  - Note：有一个点需要注意：就是在此情况下 TX 和 RX 可能处于不同的 LTSSM 状态（如 Tx 处于 L0s，而 Rx 处于 L0）。
- 如果接收器没有实现 L0s，则当在任意 Lane 上接收到 EIOS 时状态机就转换到 Recovery 状态，且 Port 不能直接定向到 L1 和 L2 状态，参阅 Section 4.2.6.6.1。
- 下列条件满足，状态机就转换到 L1 状态。
  - 上层逻辑发出直接定向，且在任意 Lane 收到一个 EIOS，并且在 2.5GT/s 和 8GT/s 速率下在所有 Lane 上发送一个 EIOS 或在 5GT/s 速率下在所有 Lane 上发送两个连续的 EIOS，状态机就跳入 L1。
  - Note：上述的"直接"指的是在两端都接收和发送了 EIOS 之后都同意立即进入 L1 状态。PCI-PM(see Section 5.3.2.1) 或 ASPM(see Section 5.4.1.2.1) 方式可以导致链路进入 L1。
  - Note：当由上层逻辑指示时，链路的一侧会一致在所有通道上发送 EIOS 来启动进入 L1 的过程，然后进入电气空闲[注 1]。然后该端口会等待在任意 Lane 上收到一个 EIOS，之后状态机立即转换到 L1。相反的，在任意 Lane 上首先收到 EIOS 的一侧必须在所有 Lane 上发送 EIOS 并且立即转化到 L1。
- 下列条件之一满足，状态机就转换到 L2 状态。
  - 上层逻辑发出直接定向，且在任意 Lane 收到一个 EIOS，并且在 2.5GT/s 和 8GT/s 速率下在所有 Lane 上发送一个 EIOS 或在 5GT/s 速率下在所有 Lane 上发送两个连续的 EIOS，状态机就跳入 L2。
  - Note：上述的"直接"指的是在两端都接收和发送了 EIOS 之后都同意立即进入 L2 状态。Section 5.3.2.3 给出了更多解释。
  - Note：当由上层逻辑指示时，链路的一侧会一致在所有通道上发送 EIOS 来启动进入 L2 的过程，然后进入电气空闲[注 2]。然后该端口会等待在任意 Lane 上收到一个 EIOS，之后状态机立即转换到 L2。相反的，在任意 Lane 上首先收到 EIOS 的一侧必须在所有 Lane 上发送 EIOS 并且立即转化到 L2。

注 1：被驱动的共模电压必须满足 L0 期间直流共模电压的绝对增量（$V_{TX\text{-}CM\text{-}DC\text{-}ACTIVE\text{-}IDLE\text{-}DELTA}$）和电气空闲状态的规范（请参见 Table 4-18）。

注 2：被驱动的共模电压不用满足 L0 期间直流共模电压的绝对增量（$V_{TX\text{-}CM\text{-}DC\text{-}ACTIVE\text{-}IDLE\text{-}DELTA}$）和电气空闲状态的规范（请参见 Table 4-18）。

4.2.6.6 L0s

L0s 子状态如 Figure 4-27 所示。

公众号：芯片业者的承双

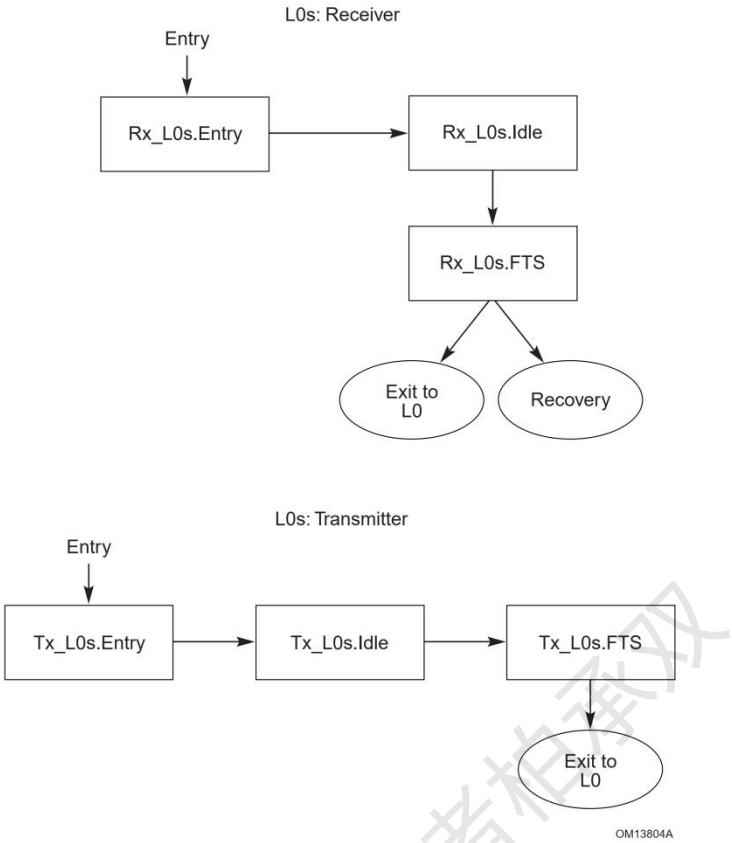


Figure 4-27: L0s Substate Machine

4.2.6.6.1 Receiver L0s

如果接收器的 Port 通告对 L0s 的支持，则接收器必须实现 L0s，Link Capability 寄存器中的 ASPM Support 字段用于指示对该功能的支持。即使接收器的 Port 不通告对 L0s 的支持，也允许该接收器实现 L0s。

4.2.6.6.1.1 Rx_L0s.Entry

- $T_{TX\text{-}IDLE\text{-}MIN}$（这个值在 Table4-18 中定义）超时后进入 Rx_L0s.Idle 状态。
  - Note：这是发送器进入电气空闲的最小时间。

4.2.6.6.1.2 Rx_L0s.Idle

- 如果接收器在任意已配置的 Lane 上检测到电气空闲退出，则状态机转换到 Rx_L0s.FTS 状态。

4.2.6.6.1.3 Rx_L0s.FTS

- 如果在链路的所有已配置的 Lane 上收到 8b/10b 编码的 SKP 有序集或者 128b/130b 编码的 SDS 有序集，则状态机会跳转到 L0。
  - 接收器必须在接收到 8b/10b 编码的 SKP 有序集后能立即接收有效数据。
  - 接收器必须在接收到 128b/130b 编码的 SSDS 有序集后能立即接收有效数据。
  - 在离开 Rx_L0s.FTS 状态时必须完成 Lane-to-Lane de-skew。

公众号·芯片业者相关权

- 否则，在 N_FTS 超时后进入 Recovery 状态。
  - 当使用 8b/10b 编码：N_FTS 超时不得小于 40 * [N_FTS + 3] * UI（3 * 40 UI 是从六个符号推导而来，以覆盖最大的 SKP 有序集+四个符号（可能有额外的 FTS）+ 2 个符号的设计余量），且不得超过此数字的两倍。 设置 Extended Synch 字段后，接收器 N_FTS 超时必须调整为不小于 40 * [2048] * UI（2048 个 FTS），并且不大于 40 * [4096] * UI（4096 个 FTS）。在规范定义的范围内选择合适的超时值时，实现时必须考虑最坏情况的 Lane to Lane skew，设计裕度以及四到八个连续的 EIE 符号（速度不超过 2.5 GT/s）。
  - 当使用 128b/130b 编码：N_FTS 超时不得小于 130 * [N_FTS + 5 + 12 + Floor（N_FTS/32）] * UI，并且不得大于此数量的两倍。5 + Floor（N_FTS/32）表示了第一个 EIEOS，最后一个 EIEOS，SDS，周期性的 EIEOS 和一个附加 EIEOS 的情况，以防实现方式选择发送两个 EIEOS，然后在 N_FTS 被 32 整除时发送一个 SDS。如果设置了 Extended Synch 字段，则此处将考虑发送 12 个 SKP 有序集的数量。当 Extended Synch 字段被置位时，超时应与 N_FTS 等于 4096 的正常情况相同。
  - 发送器也必须转移到 Recovery 状态，但是允许它完成 TLP 和 DLLP 的整个过程。
  - 建议在过渡到 Recovery 时增加 N_FTS 字段的值，以保证将来从 Rx_L0s.FTS 过渡到 Recovery。

4.2.6.6.2 Transmitter L0s

如果发送器的 Port 通告对 L0s 的支持，则接收器必须实现 L0s，Link Capability 寄存器中的 ASPM Support 字段用于指示对该功能的支持。即使接收器的 Port 不通告对 L0s 的支持，也允许该接收器实现 L0s。

4.2.6.6.2.1 Tx_L0s.Entry

- 如果当前数据速率为 2.5 GT/s 或 8.0 GT/s，则发送器发送一个 EIOS；如果当前数据速率为 5.0 GT/s，则发送两个连续的 EIOS；然后进入电空闲状态。
  - DC 共模电压的值要符合规范定义的 $T_{TX-IDLE-SET-TO-IDLE}$ 的值。

  Note:被驱动的共模电压必须满足 L0 期间 DC 共模电压和电气空闲之间的绝对增量（$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$）的规范（请参见 Table 4-18）。
- $T_{TX-IDLE-MIN}$（这个值在 Table4-18 中定义）超时后进入 Tx_L0s.Idle 状态。

4.2.6.6.2.2 Tx_L0s.Idle

- 在直接定向作用下，下个状态为 Tx_L0s.FTS。

IMPLEMENTATION NOTE

Increase of N_FTS Due to Timeout in Rx_L0s.FTS

发送器通过在 Tx_L0s.FTS 子状态发送 N_FTS 个快速训练序列，使接收器能够重新获取比特锁定、符号锁定或块对齐。在没有 N_FTS 快速训练序列的情况下，接收器将在 Rx_L0s.FTS 子状态中超时，并且可能会增加在 Recovery 状态下通告的 N_FTS 的值。

4.2.6.6.2.3 Tx_L0s.FTS

- 发送器在所有已配置的通道上发送 N_FTS 个快速训练序列（Fast Training Sequences，FTS）。
  - 在 5.0GT/s 的数据速率下，在发送 N_FTS 个（4096 if Extended Sync = 1）FTS 之前必须发送四到八个 EIE Symbol。在 128b/130b 编码下，在发送 N_FTS 个（4096 if Extended Sync = 1）FTS 之前必须发送一个 EIEOS。在 2.5GT/s 的数据速率下，在发送 N_FTS 个（4096 if Extended Sync = 1）FTS 之前需要发送最多一个完整的 FTS。

公众号·志片业者相承双

- 在发送协商好的 N_FTS 个 FTS 之前，不能插入 SKP 有序集。
- 如果 Extended Sync 字段设置为 1b，则需要发送 4096 个 FTS。

- 当使用 8b/10b 编码，发送器在所有已配置的通道上发送一个 SKP 有序集。
- 当使用 128b/130b 编码，发送器在所有已配置的通道上发送一个 EIEOS，然后再跟一个 SDS 有序集。Note：SDS 有序集发送后在 Lane 0 上发送的第一个 Symbol 是数据流的第一个 Symbol。
- 下个状态是 L0。

4.2.6.7 L1

L1 子状态如 Figure 4-28 所示。

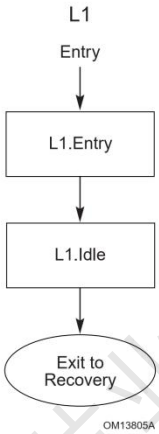


Figure 4-28: L1 Substate Machine

4.2.6.7.1 L1.Entry

- 所有已配置的发送器处于电气空闲状态。
  - DC 共模电压比现在规范定义的 $T_{TX-IDLE-SET-TO-IDLE}$ 值的范围之内。
- $T_{TX-IDLE-MIN}$ 超时后状态进入 L1.Idle 子状态。$T_{TX-IDLE-MIN}$ 在 Table 4-18 定义。
  - Note：这个时间用于保证发送器建立好电气空闲条件。

4.2.6.7.2 L1.Idle

- 发送器保持电气空闲状态。
  - DC 共模电压必须在规范定义之内。

  Note：被驱动的共模电压必须满足 L0 期间 DC 共模电压和电气空闲之间的绝对增量（$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$）的规范（请参见 Table 4-18）。
- 如果任何接收器检测到从电气空闲状态退出，或者除 2.5GT/s 速率之外在保持此子状态至少 40 ns 之后被直接定向，则下一个状态是 Recovery。
  - Note：在该子状态下，除了 2.5 GT/s 以外的速度，至少需要 40 ns 的停留时间，以解决逻辑电平上的延迟，以便在链

公众号·芯片业者相承双

路进入 L1 并立即退出 L1 状态时为电气空闲检测电路提供准备。

- 允许 Port 遵循 L0 中描述的相同规则将 directed_speed_change 变量设置为 1b。 进行此类转换时，必须将 changed_speed_recovery 变量重置为 0b。Port 也可以从 Recovery 转换回 L0，然后在从 L0 到 Recovery 的转换上将 directed_speed_change 变量设置为 1b。
- 如果使用直接定向的方式更改链路宽度，则也允许 Port 从 L1 进入 Recovery。Port 必须遵循 L0 状态所述相同的规则来更改链路宽度。

4.2.6.8 L2

L2 子状态如 Figure 4-29 所示。

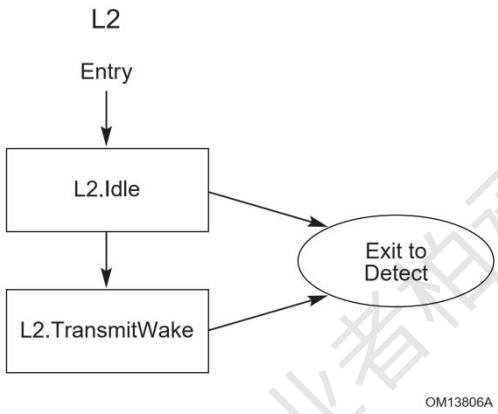


Figure 4-29: L2 Substate Machine

4.2.6.8.1 L2.Idle

- 所有 RX 的终端必须保持使能为低阻抗状态。
- 所有已配置的发送器必须处于电气空闲状态至少 $T_{TX\text{-}IDLE\text{-}MIN}$ 时间（Table 4-18）。
  - DC 共模电压必须在规范定义的范围内。
  - 接收器必须至少等待 $T_{TX\text{-}IDLE\text{-}MIN}$ 再开始查找电气空闲退出。
- 对于 Downstream Lanes：
  - 对于 Root Port，如果至少在 Lane 0 上检测到一个 Beacon 或者因直接定向作用，状态机跳转到 Detect 状态。
    - 在进入 Detect 前主电源必须恢复。
    - Note：“直接定向”指的是上层逻辑决定进入 Detect 状态。
  - 对于一个 Switch，如果至少在 Lane 0 上检测到一个 Beacon，则 Upstream Port 必须转换到 L2.TransmitWake 状态。
- 对于 Upstream Lanes：
  - 如果在任意之前预确定过的通道组上检测到电气空闲退出，则状态机进入 Detect 状态。
    - 预确定的通道组必须包括但不限于可能与链路的 Lane 0 进行协商的任何通道。对于多通道链路，预确定的通道数必须大于或等于 2。

公众号·芯片业者怕承双

- 一个 Switch 可能把某些 Downstream Lanes 转换到 Detect 状态。

- 对于 Upstream Port，如果在直接定向作用下发送一个 Beacon，则状态机转换到 L2.TransmitWake 状态。
  - Note：Beacons 只能由 Upstream Port 往 RC 方向发送。

4.2.6.8.2 L2.TransmitWake

只有 Upstream Port 有这个状态。

- 在此子状态下，至少需要在 Lane 0 上发送 Beacon。
- 如果任意接收器沿 RC 方向检测到电气空闲退出，则下个状态跳转到 Detect。
  - Note：当上游接收器看到电气空闲已退出时，需要保证电源被恢复，但是也可以在电气空闲退出之前恢复电源。

4.2.6.9 Disabled

- 在此状态下，在所有 Lane 上发送 16 到 32 个 Disable Link 字段设置为 1b 的 TS1 有序集，然后进入电气空闲。
  - 在进入电气空闲状态之前必须发送 EIOS（2.5GT/s 或 8.0GT/s 会发送一个，5.0GT/s 会发送两个）。
  - DC 共模电压不是必须要在规范定义之内。

  Note：被驱动的共模电压必须满足 L0 期间 DC 共模电压和电气空闲之间的绝对增量（$V_{TX\text{-}CM\text{-}DC\text{-}ACTIVE\text{-}IDLE\text{-}DELTA}$）的规范（请参见 Table 4-18）。
- 如果在任意 Lane 上发送一个 EIOS（2.5GT/s 或 8.0GT/s 会发送一个，5.0GT/s 会发送两个）并且接收了一个 EIOS（即使正传输 TS1 的 Disable Link 置为 1b），就会执行以下操作：
  - LinkUp = 0b。
    - 在此视角下，链路处于 Disabled 状态。
  - 对于 Upstream Port：如果接收器检测到电气空闲退出则下个状态切换到 Detect。
- 对于 Downstream Port：在直接定向作用下，下个状态为 Detect 状态（比如，软件把 Link Disable 设置为 0b）。
- 对于 Upstream Port：如果 2ms 超时后没收到 EIOS，则下个状态为 Detect。

4.2.6.10 Loopback

Loopback 子状态机如 Figure 4-30。

公众号·芯片业者相承双

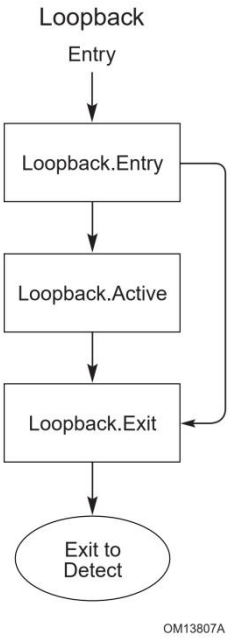


Figure 4-30: Loopback Substate Machine

4.2.6.10.1 Loopback.Entry

- LinkUp = 0b。
- 在此子状态下，接收器将忽略接收的的 TS1 或 TS2 有序集中的 Link number 和 Lane number 字段的值。
- Loopback master 要求：
  - 如果从 Configuration.Linkwidth.Start 进入 Loopback.Entry，需要确定 master 与在切换到 Loopback.Entry 时在任意有效的 Lane 上收到的两个连续 TS1 或 TS2 有序集中的数据速率的最高公共数据速率。如果当前数据速率不是最高的公共数据速率：
    - 发送 16 个 Loopback 字段置为 1b 的 TS1 有序集，然后在 2.5GT/s 或 8GT/s 速率下再发送一个 EIOS，在 5.0GT/s 下再发送两个 EIOS，然后转换到电气空闲并停留 1ms。在这段电气空闲期间内将速率切换到最高的公共速率上。
    - 如果最高公共速率是 5.0GT/s，则 slave 的发送器的 de-emphasis 由发送的 TS1 有序集中的 Selectable De-emphasis 字段控制（1b = -3.5 dB, 0b = -6 dB）。
    - 对于 5.0GT/s 和以上速率，允许发送器实现自己的发送器参数设置，而忽略它发送给 slave 的发送器参数设置。
    - Note：如果在 LinkUp 设置为 1b 之后进入 Loopback 状态，则一个 Port 可以从 Recovery 进入 Loopback 状态，另一个 Port 可以从 Configuration 进入 Loopback 状态。从 Configuration 进入 Loopback 状态的 Port 可以尝试改变数据速率而另一个不能。如果此场景发生，则结果是未定义的。测试最好避免这种场景。
  - 发送 Loopback 字段置为 1b 的 TS1 有序集。
    - 还允许 master 在 Loopback.Entry 中发送 Compliance Receive 字段置为 1b 的 TS1 有序集，包括在数据速率变化之前发送的那些内容。如果它把 Compliance Receive 字段置为 1b，则在处于 Loopback.Entry 状态时不得再次将其置为 0b。 当一个或两侧端口在数据速率更改后难以获得比特锁定、符号锁定或块对齐时，此使用模型在测试和验证场景非常有用。设置 Compliance Receive 字段的能力是特定于实现的。
  - 如果发送的 TS1 有序集中的 Compliance Receive 字段被置为 1b，则在 2ms 超时后状态切换到 Loopback.Active。

公众号·芯片业者相承双

- 如果发送的 TS1 有序集中的 Compliance Receive 字段被置为 1b，并且与实现相关的一组 Lane 上收到了两个连续的 Loopback 字段置为 1b 的 TS1 有序集，则下个状态切换到 Loopback.Active。

  如果更改了数据速率，则 master 必须考虑 slave 可能处于电空闲状态的时间，并发送足够数量的 TS1 有序集，以便 slave 在进行 Loopback.Active 之前获取符号锁定或块对齐。

IMPLEMENTATION NOTE

Lane Numbering with 128b/130b Encoding in Loopback

如果当前数据速率使用 128b/130b 编码并且尚未协商好通道编号，则 master 和 slave 可能无法解码接收到的信息，因为它们的通道使用不同的加扰 LFSR 种子值（因为 LFSR 种子值由 Lane number 确定）。这种场景可以通过以下方式避免：允许 master 和 slave 在将 master 定向至 Loopback 之前完成通道编号协商，然后指示 master 在 Loopback.Entry 期间把 Compliance Receive 字段被置为 1b，或使用其他方法来确保 LFSR 种子值匹配 。

- 如果一个小于 100ms 的与实现相关的超时发生，则下个状态跳转到 Loopback.Exit。

■ Loopback slave 要求：

- 如果从 Configuration.Linkwidth.Start 进入 Loopback.Entry，需要确定 slave 与在切换到 Loopback.Entry 时在任意有效的 Lane 上收到的两个连续 TS1 或 TS2 有序集中的数据速率的最高公共数据速率。如果当前数据速率不是最高的公共数据速率：
  - ◆ 则如果当前速率是 2.5GT/s 或 8GT/s 则需要发送一个 EIOS，如果当前速率为 5.0GT/s 则需要发送两个 EIOS，然后转换到电气空闲并停留 2ms。在电气空闲期间，速率切换到最高公共数据速率上。
  - ◆ 如果最高公共速率是 5.0GT/s，则将发送器的 de-emphasis 设置为由 TS1 有序集中的 Selectable De-emphasis 字段指定的值（1b = -3.5 dB, 0b = -6 dB）。
  - ◆ 如果最高公共速率是 8.0GT/s 而且因 EQ TS1 把 slave 定向到此子状态，则把发送器的参数设置设置为 TS1 有序集中携带的 Preset 字段指定的值。如果最高公共速率是 8.0GT/s 而且因标准 TS1 把 slave 定向到此子状态，slave 就使用其默认的发送器参数设置值。
- 如果因 TS1 有序集中的 Compliance Receive 字段置为 1b 导致进入此子状态，则下个状态为 Loopback.Active。
  - ◆ slave 的发送器无需切换为在任何边界上发送环回数据，并且可以截断任何正在进行的有序集。
- 否则，slave 的发送器发送 Link number 和 Lane number 设置为 PAD 的 TS1 有序集。
  - ◆ 如果数据速率为 2.5GT/s 或 5.0GT/s 且完成符号锁定，则下个状态是 Loopback.Active。
  - ◆ 如果数据速率为 8.0GT/s 且在所有有效 Lane 上收到两个连续的 TS1 有序集，则下个状态是 Loopback.Active。如果 EC 字段值适用于 slave 的 Port（10b 或 11b）且请求的设置是一个 preset 或一组有效的 coefficient，则由接收到的 TS1 有序集指定的均衡参数设置就必须做评估并应用于发送器（Note：这个过程等同于 Recovery.Equalization 状态的行为）。slave 可选择实现两者（指 EC 为 10b 和 11b）的值都接收。如果应用了该设置，则它们必须在接收到的 500 ns 内生效，并且不得使发送器违反任何电气规范超过 1 ns。与 Recovery.Equalization 不同，新设置不会反映在 slave 发送的 TS1 有序集中。
  - ◆ 当使用 8b/10b 编码，slave 的发送器必须切换为在一个符号边界上发送环回数据，但是它允许截断任何正在传输的有序集。当使用 128b/130b 编码，slave 的发送器没必要在任何边界上发送环回数据，并且它允许截断任何正在传输的有序集。

公众号：芯片业者柏承双

4.2.6.10.2 Loopback.Active

- 在此子状态下，发送器必须发送有效的编码数据。Loopback master 在退出 Loopback 之前应该避免将 EIOS 当做数据发送。
- 在继续执行时钟容限补偿的同时，需要回送 slave 重新发送接收到的已编码信息，并需要应用 Polling 阶段确定的极性反转：
  - 必须按照 Section 4.2.7 中的概述，在每个通道上添加或删除 SKP，不同之处在于不必跨已配置链路的通道同时添加或删除 SKP。
    - 对于 8b/10b 编码，如果重发的 SKP 有序集需要添加一个 SKP Symbol 以适应时序容限矫正，需要在 COM Symbol 后的 SKP 有序集中与 SKP Symbol 相邻的重传符号流的任意位置插入该 SKP Symbol。插入的 SKP 符号必须与 SKP 有序集中收到的 SKP 符号具有相同的差异。
    - 对于 8b/10b 编码，如果重发的 SKP 有序集需要减少一个 SKP Symbol 以适应时序容限矫正，该 SKP Symbol 只需要简单的不进行重传就行。
    - 对于 128b/130b 编码，如果重发的 SKP 有序集需要添加 SKP Symbol 以适应时序容限矫正，需要在有序集的 SKP_END Symbol 前的重传符号流中插入四个 SKP Symbol。
    - 对于 128b/130b 编码，如果重发的 SKP 有序集需要减少 SKP Symbol 以适应时序容限矫正，则 SKP 有序集中 SKP_END Symbol 前的四个 SKP Symbol 只需简单的不进行重传就行。
  - 即使它被确定为无效编码（即，对于 8b/10b 编码无法转换为合法的 K 或 D 字符；对于 128b/130b 编码无法转换为合法的 Sync Header 或有效的有序集），Loopback slave 也不允许修改接收到的编码数据（Polling 中确定的极性反转情况除外）。
- 如果下述之一条件满足则状态机跳转到 Loopback.Exit：
  - 如果直接定向或在任何通道上收到四个连续的 EIOS。必须注意的是，在 8b/10b 编码中，接收到四个连续的 EIOS 表示该通道接收到四组连续的 COM、IDL、IDL、IDL，或者四个 EIOS 中 IDL 符号占 IDL 总数的三分之二。在 128b/130b 编码中，接收四个连续的 EIOS 表示在最后一个 EIOS 中的 01b 同步头之后的前三个和前四个符号中接收完整的 130 位 EIOS。
  - 可选地，如果当前链路速度为 2.5 GT/s，并且在任何通道上都接收到 EIOS 或检测/推断出电气空闲。
  - Loopback slave 必须能够在 Loopback slave 接收到 EIOS 的 1ms 内检测到任何通道上的电气空闲状况。
  - Note：在接收到 EIOS 之后且在 Loopback slave 实际检测到电气空闲之前的时间内，Loopback slave 可能会接收到编码方案未定义的比特流，该比特流可能会被发送器送回。
  - $T_{TX\text{-}IDLE\text{-}SET\text{-}TO\text{-}IDLE}$ 参数在这种情况下不适用，因为 Loopback slave 可能在 EIOS 之后的 1ms 之内无法检测到电气空闲。
- 在直接定向作用下下个状态跳转到 Loopback.Exit。

4.2.6.10.3 Loopback.Exit

- Loopback master 在仅支持 2.5 GT/s 数据速率的端口发送一个 EIOS，并在支持大于 2.5 GT/s 数据速率的端口发送八个连续的 EIOS（可选择实现对于仅支持 2.5 GT/s 数据速率的端口也发送八个连续 EIOS），不论当前的链路速度如何，都要在所有通道上进入电气空闲状态保持 2ms。
  - 发送最后的 EIOS 后，Loopback master 必须在 $T_{TX\text{-}IDLE\text{-}SET\text{-}TO\text{-}IDLE}$ 内所有通道都转换为电气空闲状态。
  - Note：EIOS 可用于表示 Loopback master 发生的传输和比较操作结束。接收到任何 EIOS 后，Loopback master 收到的任何数据都应忽略，因为它是未定义的。

- Loopback slave 必须所有 Lane 进入电气空闲并保持 2ms。
  - 进入电气空闲之前，Loopback slave 必须对检测到电气空闲之前收到的所有符号进行环回。这样可以确保 Loopback master 可以看到 EIOS，以表示任何环回发送和比较操作的逻辑结束。
- Loopback master 和 Loopback slave 的下个状态都是 Detect 状态。

4.2.6.11 Hot Reset

DSP 通过发送两个连续的 hot reset 字段为 1 的 TS1 序列来让 USP 进入热复位状态。

```
4738        S_HOT_RESET_ENTRY: begin
4739            ltssm_in_hotrst_dis_entry = 1'b1;
4740            // For downstream ports, stay here until the link partner reaches Hot Reset.
4741            if ((!cfg_upstream_port
4742                 ) & link_any_2_ts1_linknmtx_lanenmtx_hotreset1_rcvd)
4743                next_lts_state   = S_HOT_RESET;
4744            //as long as the EP is contineously receiving TS1 with reset, stay in HOT_RESET_ENTRY state and keep linkup
4745            //When timeout_2ms happened after rcvd_2rst is deasserted, then
4746            //we will go into predetect quiet
4747            // The timeout timer is reset everytime rcvd_2rst signal is assert
4748            else if (timeout_2ms)
4749                next_lts_state   = S_HOT_RESET;
4750            else
4751                next_lts_state   = S_HOT_RESET_ENTRY;
4752        end
4753        S_HOT_RESET: begin
4754            // Once Reset goes away
4755            if (!latched_direct_rst | ds_timeout_2ms
4756                | cfg_upstream_port
4757               )
4758                next_lts_state   = S_PRE_DETECT_QUIET;
4759            else
4760                next_lts_state   = S_HOT_RESET;
4761        end
```

- 由上层逻辑引导进入 Hot Reset 的 Lane 会执行以下操作（这个多应用于 Downstream Port）：
  - 在所有已配置的 Lane 上发送 Hot Reset 字段置为 1b 且 Link number 与 Lane number 为已配置好的值的 TS1 有序集。
  - 如果在任意已配置的 Lane 上收到两个连续的 Hot Reset 字段置为 1b 的 TS1 有序集，则：
    - LinkUp = 0b。
    - 如果没有上层逻辑要求保持 Hot Reset，则下个状态为 Detect 状态。
    - 否则，继续在所有已配置的 Lane 上发送 Hot Reset 字段置为 1b 且 Link number 与 Lane number 为已配置好的值的 TS1 有序集
  - 否则，在 2ms 超时后进入 Detect 状态。
- 并非由上层逻辑引导进入 Hot Reset 的 Lane 会执行以下操作（比如在任意已配置的 Lane 上收到两个连续的 Hot Reset 字段置为 1b 的 TS1 有序集）（这个多应用于 Upstream Port）：
  - LinkUp = 0b。
  - 如果一个 Switch 的 Upstream Port 的任意 Lane 收到两个连续的 Hot Reset 字段置为 1b 的 TS1 有序集，所有已配置的 Downstream Port 必须立即切换到 Hot Reset。
    - Switch 上的 crosslink 是个此规则的例外，它的行为与具体实现有关。
  - 所有已配置的 Lane 上要发送 Hot Reset 字段置为 1b 且 Link number 与 Lane number 为已配置好的值的 TS1 有序集。
  - 如果收到了两个连续的 Hot Reset 字段置为 1b 且 Link number 与 Lane number 为已配置好的值的 TS1 有序集，将继续保持 Hot Reset 状态并重置 2ms 计时器。
  - 否则，在 2ms 超时后进入 Detect 状态。

公众号·芯片业者相承双

Note：通常，Downstream Port 或 crosslink Port 的通道将被直接定向到 Hot Reset，而 Upstream Port 或 crosslink Port 的通道将通过接收两个连续的 Hot Reset 字段置为 1b 的 TS1 有序集从 Recovery.Idle 状态进入 Hot Reset。

4.2.7 Clock Tolerance Compensation

SKP Ordered Set（如 Section 4.2.7.1 和 Section 4.2.7.2 定义）用于补偿链路两端比特率之间的频率差异。接收器物理层逻辑子层必须实现一个执行此补偿的弹性缓存（Elastic Buffer）。SKP 有序集传输之间的间隔是根据 Table 8-6，Table 8-10 和 Table 8-17 中指定的 Transmit、Receiver 和 Refclk 规范得出的。

该规范支持共享参考时钟体系结构（common Refclk），这样 Tx 和 Rx Refclk 速率之间没有差异；规范还支持两种参考时钟体系结构，其中 Tx 和 Rx Refclk 速率不同，即 Separate Reference Clocks With No SSC - SRNS 和 Separate Reference Clocks with Independent SSC - SRIS。SRNS 的最大频率差异为 600 ppm，这可能导致每 1666 个 Cycle 偏移一次时钟。SRIS 的最大差异为 5600 ppm，这可能导致每 178 个 Cycle 偏移一次时钟。

特定的外形规格规范要么实现 SRIS 时钟结构，要么实现 SRNS 时钟结构，或者实现可以在这两种时钟结构之间切换的机制。Upstream Port 允许实现对 SRIS 和 SRNS 的任何组合的支持（不包括对两者的任何支持），但必须符合相关的外形规格规范的要求。支持 SRIS 的 Downstream Port 也必须支持 SRNS，除非该 Port 仅与修改这些要求的特定外形规格相关联。满足特定外形规格要求的端口配置是特定于实现的。Section 8.6.8 节提供了外形规格的具体指南。

即使 Port 运行于 SRIS，如果接收器能够以 SRNS 中使用的速率生成 SKP Order Set，则允许该 Port 把 Link Capabilities 2 Register 中的 Lower SKP OS Reception Supported Speeds Vector 字段设置为合适的数据速率。即使 Port 运行于 SRIS，如果发送器能够以 SRNS 中使用的速率生成 SKP Order Set，则允许该 Port 把 Link Capabilities 2 Register 中的 Lower SKP OS Reception Supported Speeds Vector 字段设置为合适的数据速率。系统软件必须先把 Link Partner 的 Link Control 3 Register 的 Enable Lower SKP OS Generation Vector 字段设置为适当的数据速率，然后检查 Lower SKP OS Reception Supported Speeds Vector 域中的位是否已设置。链路上存在的任何软件透明的扩展设备（例如中继器）也必须支持 lower SKP OS Generation，系统软件才能设置 Enable Lower SKP OS Generation Vector 字段。软件如何确定此类扩展设备是否支持 lower SKP OS Generation 是特定于实现的。当在 Enable Lower SKP OS Generation Vector 字段中设置了链路正在运行的数据速率的比特位时，无论链路正运行于哪种时钟架构，发送器都会以 SRNS 中使用的速率在 L0 状态中调度 SKP 有序集的生成。在其他 LTSSM 状态中，SKP 有序集调度工作于时钟结构的合适速率下。

与仅支持 SRNS 的设计相比，支持 SRIS 的组件的弹性缓冲区中可能需要更多的条目。如果 SKP 有序集在最大有效负载的数据包之后立即掉落（latter fall），则此要求考虑了调度 SKP 有序集可能需要的额外时间。

4.2.7.1 SKP Ordered Set for 8b/10b Encoding

当使用 8b/10b 编码时，传输的 SKP 有序集是一个 COM 符号，后跟三个 SKP 符号，但 Loopback.Active 状态下的 Loopback 从设备除外。收到的 SKP 有序集是一个 COM 符号，后跟一到五个 SKP 符号。

4.2.7.2 SKP Ordered Set for 128b/130b Encoding

Table 4-15 给出了使用 128b/130b 编码时 SKP 有序集的布局。传输的 SKP 有序集为 16 个符号。一个端口可以添加和删除四个 SKP 符号。因此，接收到的 SKP 有序集可以是 8、12、16、20 或 24 个符号。SKP_END 符号表示 SKP 有序集的最后四个符号，以便接收器可以识别比特流中下一个块同步头的位置。根据 LTSSM 状态，SKP_END 符号后面的三个符号包含不同的信息。

Table 4-15 中的 Data Parity 字段是自最后一个 SDS 有序集或 SKP 有序集（以最近传输的那个有序集）开始加扰后传输的所有数据块的有效负载的偶校验。每个通道的 Data Parity 字段是独立计算的。 要求接收端使用与发送器相同的方法来计算奇偶校验，并将计算结果与接收到的 Data Parity 字段进行比较。 如果计算出的值和接收到的值不匹配，则必须设置 Lane Error Status 寄存器中的对应字段，该位与检测到不匹配的默认通道号相对应。但是，这样的不匹配不是一个 Receiver Error，也不能引起链路重新训

练。

为 Modified Compliance Pattern 定义的 Error_Status 字段包含两部分信息：Pattern Lock（Error_Status [7]）和 Receiver Error Count（Error_Status [6:0]）。有关更多信息，请参见 Section 4.2.6.2.2。

Table 4-15: SKP Ordered Set with 128b/130b Encoding

Symbol Number	Value	Description
0 through (4*N - 1) [N can be 1 through 5]	AAh	SKP Symbol. Symbol 0 is the SKP Ordered Set identifier.
4*N	E1h	SKP_END Symbol. Signifies the end of the SKP Ordered Set after three more Symbols.
4*N + 1	00-FFh	(i) If LTSSM state is Polling.Compliance: AAh (ii) Else if prior block was a Data Block: Bit[7] = Data Parity Bit[6:0] = LFSR[22:16] (iii) Else: Bit[7] = ~LFSR[22] Bit[6:0] = LFSR[22:16]
4*N + 2	00-FFh	(i) If the LTSSM state is Polling.Compliance: Error_Status[7:0] (ii) Else LFSR[15:8]
4*N + 3	00-FFh	(i) If the LTSSM state is Polling.Compliance: ~Error_Status[7:0] (ii) Else: LFSR[7:0]

IMPLEMENTATION NOTE

关于 SKP 有序集中的 LFSR

传输 LFSR 值使得跟踪工具即使在比特流中间需要重新获取块对齐时也能够运行。由于跟踪工具无法强制链路进入 Recovery 状态，因此如果需要的话它们可以重新获取比特锁定，并监视 SKP 有序集以获取 Block alignment 并执行 Lane-to-Lane de-skew。 可以将“SKP 有序集中的 LFSR 值加载到其 LFSR 中，以开始解释比特流。必须注意的是，使用比特流时，可能会将其别名为非块边界上的 SKP 有序集。跟踪工具可以通过使用特定于实现的方式（例如，接收固定数量的有效数据包或同步头或后续的 SKP 有序集）来验证其块对齐。

4.2.7.3 Rules for Transmitters

- 所有通道均应以相同的频率发送符号（所有多通道链路中的比特率之差为 0 ppm）。
- 传输时，相同长度的 SKP 有序集应在多通道链路的所有通道上同时传输，但处于 Loopback.Active 状态的 Loopback Slave 允许的除外（请参见 Section 4.2.4.10 和 Table 4-18 的上下文获取同时的意义）。
- 按 Section 4.2.7.1 描述的方式传送 8b/10b 编码的 SKP 有序集。
- 按 Section 4.2.7.2 描述的方式传送 128b/130b 编码的 SKP 有序集。
- 当使用 8b/10b 编码时：一个 SKP 有序集的发送间隔在 1180 和 1538 个 Symbol Times 之间。
- 当使用 128b/130b 编码时：当 LTSSM 不处于 Loopback 状态或处于 Loopback 状态但 Loopback Slave 尚未开始环回数据，则一个 SKP 有序集的发送间隔在 370 到 375 个 block 之间。当 LTSSM 处于 Loopback 状态时，Loopback Master 必须安排两个 SKP 有序集进行传输，彼此之间最多相隔两个块，间隔为 370 到 375 个块。
- 如果分组或有序集尚未进行，则应发送预定的 SKP 有序集，否则，将其累积，然后在下一个分组或有序集边界处连续插入。

公众号：芯片业者相承及

注意：使用 128b / 130b 编码时，不能在数据流中的连续块中传输 SKP 有序集。 有关更多信息，请参见第 4.2.2.3.2 节。

- 如果 packet（指有效数据包）或有序集尚未开始传输，则已经调度好的 SKP 有序集就可以发送，否则就将其累积，然后在下一个 packet 或有序集边界处连续插入。注意：使用 128b/130b 编码时，不能在数据流中的连续块中传输 SKP 有序集。有关更多信息，请参见 Section 4.2.2.3.2。
- 检测连续的符号或有序集（例如 Polling.Active 中的八个连续的 TS1 有序集）时，插入的 SKP 有序集不算是中断了这种"连续"的行为。
- 当使用 8b/10b 编码时：如果 Link Control 2 寄存器的 Compliance SOS 字段为 0，则在 Polling.Compliance 过程中正在发送 Compliance Pattern 或 Modified Compliance Pattern（Section 4.2.8）时，不得传输 SKP 有序集。如果 Link Control 2 寄存器的 Compliance SOS 字段为 1b，则使用 8b/10b 编码正在发送 Compliance Pattern 或 Modified Compliance Pattern 时，必须在合适的时间间隔内发送已调度好的 SKP 有序集。
- 当使用 128b/130b 编码时，Compliance SOS 字段不起作用。
- 在发送器处于电气空闲状态时所花费的所有时间都不会计入用于计划发送 SKP 有序集的计划时间间隔中。
- 建议在发送器电气空闲时随时重置用于计划 SKP 有序集的任何计数器或其他机制。

4.2.7.4 Rules for Receivers

- 接收器应在使用 8b/10b 编码时识别出 Section 4.2.7.1 中定义的 received SKP Ordered Sets，在使用 128b/130b 编码时识别出 Section 4.2.7.2 中定义的 received SKP Ordered Sets。
  - 在多通道链路中，各通道接收到的 SKP 有序集的长度必须相同。这在 Loopback.Active 时可能例外，因为 Loopback.Active 可以针对各通道单独进行。
- 当使用 8b/10b 编码时，接收器应容忍在 1180 到 1538 个 Symbol Time 的平均间隔内接收和处理 SKP 有序集，而当使用 128b/130b 编码时，接收器应容忍在 370 至 375 个块之间接收和处理 SKP 有序集。
  - 注意：因为处于空闲状态的发送器不需要复位它的基于时间调度 SKP 有序集的机制，因此，接收器应有能力在小于空闲时间的 SKP 有序集之间的平均时间间隔内接收第一个有时间安排的 SKP 有序集。
- 对于 8.0 GT/s 及以上的数据速率，在 L0 状态下，接收端必须检查每个 SKP 有序集前面是否有带有 EDS 数据包的数据块。
- 接收方应能以 2.5 GT/s 和 5.0 GT/s 的数据速率接收和处理连续的 SKP 有序集。
  - 接收器应能容忍接收和处理最大间隔取决于组件支持的 Max_Payload_Size 的 SKP 有序集。对于 2.5 GT/s 和 5.0 GT/s 的数据速率，SKP 有序集之间的最大符号数（N）的公式为：N = 1538 +（Max_payload_size_byte + 28）。 例如，如果 Max_Payload_Size 为 4096 字节，则 N = 1538 + 4096 + 28 = 5662。

4.2.8 Compliance Pattern in 8b/10b Encoding

在 Polling 状态期间，必须根据 Section 4.2.6.2.1 中所述的条件从 Polling.Active 转移到 Polling.Compliance 子状态。在 8b/10b 下，Compliance Pattern 由重复的 K28.5，D21.5，K28.5 和 D10.2 的符号序列组成。Compliance Pattern 序列如下：

Symbol	K28.5	D21.5	K28.5	D10.2
Current Disparity	Negative	Positive	Positive	Negative
Pattern	0011111010	1010101010	1100000101	0101010101

公众号·芯片业者相承双

第一个一致性序列符号必须具有负差异（negative disparity）。允许创建差异错误（disparity error），以使执行差异（running disparity）与第一个一致性序列符号的负差异对齐。

对于具有多个通道的任何给定设备，每八个通道总共延迟四个符号。在四个 Symbol Compliance Pattern 序列的开始和结尾均出现两个符号延迟。 x1 设备或在 x1 模式下运行的 xN 设备允许在 Compliance Pattern 中包含 Delay Symbols。

那么，每八个通道的延迟序列为：

Symbol:	D	D	K28.5	D21.5	K28.5	D10.2	D	D

其中 D 是 K28.5 符号。第一个 D 符号具有负差异，以使延迟差异（delay disparity）与 Compliance 序列的差异对齐。

发送八个符号后，延迟符号将前进到下一个通道，直到在所有八个通道上都发送了延迟符号。然后，延迟符号循环回到通道 0，并重复该过程。无论检测到或支持的通道数量如何，都可以在所有八个通道上提前延迟序列。下表给出了此过程的说明：

Lane 0	D	D	K28.5-	D21.5	K28.5+	D10.2	D	D	K28.5-	D21.5	K28.5+	D10.2
Lane 1	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	D	D	K28.5-	D21.5
Lane 2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
Lane 3	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
Lane 4	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
Lane 5	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
Lane 6	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
Lane 7	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2
Lane 8	D	D	K28.5-	D21.5	K28.5+	D10.2	D	D	K28.5-	D21.5	K28.5+	D10.2
Lane 9	K28.5-	D21.5	K28.5+	D10.2	K28.5-	D21.5	K28.5+	D10.2	D	D	K28.5-	D21.5

Key:

K28.5-	COM when disparity is negative, specifically: “0011111010”
K28.5+	COM when disparity is positive, specifically: “1100000101”
D21.5	Out of phase data Symbol, specifically: “1010101010”
D10.2	Out of phase data Symbol, specifically: “0101010101”
D	Delay Symbol K28.5 (with appropriate disparity)

此延迟序列可确保相邻通道之间的干扰，从而能够在接近最坏情况的符号间干扰（Inter-Symbol Interference，ISI）和串扰（cross-talk）条件下测量一致性模式。

4.2.9 Modified Compliance Pattern in 8b/10b Encoding

Modified Compliance Pattern 由 basic Compliance Pattern sequence（请参阅 Section 4.2.8）组成，只做了一处更改。两个相同的 error status Symbol，后跟两个 K28.5，被附加到 8b/10b 符号的 basic Compliance Pattern sequence（K28.5，D21.5，K28.5 和 D10.2）上，以形成（K28）的 Modified Compliance Pattern，即 K28.5，D21.5，K28.5，D10.2，error status Symbol，error status Symbol，K28.5，K28.5 。Modified Compliance Pattern 的第一个符号必须具有负差异。允许创建差异错误（disparity error），以使执行差异（running disparity）与第一个 Modified Compliance Sequence 符号的负差异对齐。对于具有多个通道的任何给定设备，每八个通道总共移动八个符号。在八个 Symbol Modified Compliance Pattern sequence 的开头，出现了 K28.5 的四个符号，在结尾出现了另一个 K28.7 的四个符号。第一个 D 符号具有负差异，以使延迟差异与 Modified Compliance Sequence 的差异对齐。发送 16 个符号后，延迟符号将前进到下一个通道，直到在所有八个通道上都发送了延迟符号。然后，延迟符号循环回到通道 0，并重复该过程。无论检测到或支持的通道数量如何，都可以在所有八个通道上提前延迟序列。x1 设备或在 x1 模式下运行的 xN 设备允许在 Modified Compliance Pattern 中包含 Delay Symbols。

Modified Compliance Pattern 如下：

Lane0	D	D	D	D	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.7-	K28.7-	K28.7-	K28.7-	K28.5-	D21.5
Lane1	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	D	D
Lane2	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5
Lane3	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5
Lane4	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5
Lane5	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5
Lane6	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5
Lane7	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5
Lane8	D	D	D	D	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.7-	K28.7-	K28.7-	K28.7-	K28.5-	D21.5
Lane9	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	K28.5-	D21.5	K28.5+	D10.2	ERR	ERR	K28.5-	K28.5+	D	D

Key:

K28.5-	COM when disparity is negative, specifically: “0011111010”
K28.5+	COM when disparity is positive, specifically: “1100000101”
D21.5	Out of phase data Symbol specifically: “1010101010”
D10.2	Out of phase data Symbol, specifically: “0101010101”
D	Delay Symbol K28.5 (with appropriate disparity)
ERR	error status Symbol (with appropriate disparity)
K28.7-	EIE when disparity is negative, specifically “0011111000”

插入两个相同的错误符号而不是一个错误符号的原因是为了确保 8b/10b 序列的差异不会受到添加错误状态符号的影响。

所有其他 Compliance pattern 规则都是相同的（即，用于添加延迟符号的规则），以便保留 Compliance pattern 的所携带的串扰特性。

error status Symbol 是一个 8b/10b 的 D Symbol，在每个通道上都有，其各字段定义如下：

- Receiver Error Count (Bits 6:0) ：当某 Pattern Lock 有效，这个字段就递增。
- Pattern Lock (Bit 7) ：当某 Lane 锁定了输入的 Modified Compliance Pattern 该字段就有效。

4.2.10 Compliance Pattern in 128b/130b Encoding

128b/130b 编码下的 Compliance Pattern 由以下 36 个 block 的重复序列组成。

1. Sync Header 为`01b 的 block，其 128bit 未加扰的负载由 64 个 1b 后跟 64 个 0b 组成。
2. Sync Header 为`01b 的 block，其 128bit 未加扰的负载构成如下：

公众号·志齐业者相承双