  - the Margin Type is not No Command
  - the Receiver Number is the number assigned to the Receiver or Margin Type is either Clear Error Log or Go to Normal Settings and the Receiver Number is 'Broadcast'
  - the Usage Model field is 0b
  - the Margin Type, the Receiver Number, and Margin Payload fields are consistent with the definitions in Table 4-26
- The Upstream Port must transmit the Control SKP Ordered Set with No Command.
- A target Receiver must apply and respond to the Margin Command within 1ms of receiving the valid Margin Command if the Link is still in L0 state and operating at 16.0 GT/s or higher Data Rate.
  - A target Receiver in a Retimer must send a response in the Control SKP Ordered Set in the Upstream Direction within 1 ms of receiving the Margin Command.
  - A target Receiver in the Upstream Port must update the Status field of the Lane Margin Command and Status register within 1 ms of receiving the Margin Command.
  - A target Receiver in the Downstream Port must update the Status field of the Lane Margin Command and Status register within 1 ms of receiving the Margin Command if the command is not broadcast or no Retimer(s) are present
- For a valid Margin Type, other than No Command, that is broadcast and received by a Retimer:
  - A Retimer, in position X (see Figure 4-35), forwards the response unmodified in the Upstream Control SKP Ordered Set, if the command has been applied, else it sends the No Command.
  - The Receiver Number field of the response must be set to an encoding of one of the Retimer's Pseudo Ports.
  - The Retimer must respond only after both Pseudo Ports have completed the Margin Command.
- The Retimer must overwrite Bits [4:0] of Symbol 4N+1, Bits[7, 5:0] of Symbol 4N+2 and Bits [7:0] in Symbol 4N+3 as it forwards the Control SKP Ordered Set in the Upstream direction if it is the target Receiver of a Margin Command and is executing the command.
- On receipt of a Control SKP Ordered Set, the Downstream Port must reflect the Margining Lane Status Register from the corresponding fields in the received Control SKP Ordered Set within 1 µs, if it passes the Margin CRC and Margin Parity checks and one of the following conditions apply:
  - In the Margining Lane Control Register: Receiver Number is 010b through 101b
  - In the Margining Lane Control Register: Receiver Number is 000b, Margin Command is Clear Error Log, No Command, or Go to Normal Settings, and there are Retimer(s) in the Link
  - Optionally, if the Margining Lane Control Register Usage Model field is 1b
  - Optionally, if the Margining Lane Control Register Receiver Number field is 110b or 111b

    The Margining Lane Status Register fields are updated regardless of the Usage Model bit in the received Control SKP Ordered Set.
- A component must advertise the same value for each parameter defined in Table 8-11 in Section 8.4.4 across all its Receivers. A component must not change any parameter value except for $M_{SampleCount}$ and $M_{ErrorCount}$ defined in Table 8-11 in Section 8.4.4 while LinkUp = 1b.
- A target Receiver that receives a valid Step Margin command must continue to apply that offset until any of the following occur:
  - it receives a valid Go to Normal Settings command
  - it receives a subsequent valid Step Margin command with different Margin Type or Margin Payload field

  - $M_{IndErrorSampler}$ is 0b and $M_{ErrorCount}$ exceeds Error Count Limit
  - Optionally, $M_{IndErrorSampler}$ is 1b and $M_{ErrorCount}$ exceeds Error Count Limit.
- If a Step Margin command terminates because $M_{ErrorCount}$ exceeds Error Count Limit, the target Receiver must automatically return to its default sample position and indicate this in the Margin Payload field (Step Margin Execution Status = 00b). Note: termination for this reason is optional if $M_{IndErrorSampler}$ is 1b.
- If $M_{IndErrorSampler}$ is 0b, an error is detected when:
  - The target Receiver is a Port that enters Recovery or detects a Data Parity mismatch while in L0
  - The target Receiver is a Pseudo Port that enters Forwarding training sets or detects a Data Parity mismatch while forwarding non-training sets.
- If $M_{IndErrorSampler}$ is 1b, an error is detected when:
  - The target Receiver is a Port and a bit error is detected while in L0
  - The target Receiver is a Pseudo Port and a bit error is detected while the Retimer is forwarding non-training sets
- If $M_{IndErrorSampler}$ is 0b and either (1) the target Receiver is a Port that enters Recovery or (2) the target Receiver is a Pseudo Port that enters Forwarding training sets:
  - The target Receiver must go back to the default sample position
  - If the target Receiver is a Port that is still performing margining, it must resume the margin position within 128 µs of entering L0
  - If the target Receiver is a Pseudo Port that is still performing margining, it must resume the margin position within 128 µs of Forwarding non-training sets
- A target Receiver is required to clear its accumulated error count on receiving Clear Error Log command, while it continues to margin (if it is the target Receiver of a Step Margin command still in progress), if it was doing so.
- For a target Receiver of a Set Error Count Limit command, the new value is used for all future Step Margin commands until a new Set Error Count Limit command is received.
- If no Set Error Count Limit is received by a Receiver since entering L0, the default value is 4.
- Behavior is undefined if a Set Error Count Limit command is received while a Step Margin command is in effect.
- Once a target Receiver reports a Step Margin Execution Status of 11b (NAK) or 00b ('Too many errors'), it must continue to report the same status as long as the Step Margin command is in effect.
- A target Receiver must not report a Step Margin Execution Status of 01b ('Set up for margin in progress') for more than 100 ms after it receives a new valid Step Margin command
- A target Receiver that reports a Step Margin Execution Status other than 01b, cannot report 01b subsequently unless it receives a new valid Step Margin command.
- Reserved bits in the Margin Payload must follow these rules:
  - The Downstream or Upstream Port must transmit 0s for Reserved bits
  - The retimer must forward Reserved bits unmodified
  - All Receivers must ignore Reserved bits
- Reserved encodings of the Margin Command, Receiver Number, or Margin Payload fields must follow these rules:
  - The retimer must forward Reserved encodings unmodified
  - All Receivers must treat Reserved encodings as if they are not the target of the Margin Command
- A Vendor Defined Margin Command or response, that is not defined by a retimer is ignored and forwarded normally.

- A target Receiver on a Retimer must return 00h on the response payload on Access Retimer register command, if it does not support register access. If a Retimer supports Access Retimer register command, the following must be observed:
  - It must return a non-zero value for the DWORD at locations 80h and 84h respectively.
  - It must not place any registers corresponding to Margin Payload locations 88h through 9Fh.

4.2.13.3 Receiver Margin Testing Requirements

Software must ensure that the following conditions are met before performing Lane Margining at Receiver:

- The current Link data rate must be 16.0 GT/s or higher.
- The current Link width must include the Lanes that are to be tested.
- The Upstream Port's Function(s) must be programmed to a D-state that prevents the Port from entering the L1 Link state. See Section 5.2 for more information.
- The ASPM Control field of the Link Control register must be set to 00b (Disabled) in both the Downstream Port and Upstream Port.
- The state of the Hardware Autonomous Speed Disable bit of the Link Control 2 register and the Hardware Autonomous Width Disable bit of the Link Control register must be saved to be restored later in this procedure.
- If writeable, the Hardware Autonomous Speed Disable bit of the Link Control 2 register must be Set in both the Downstream Port and Upstream Port. (If hardwired to 0b, the autonomous speed change mechanism is not implemented and is therefore inherently disabled.)
- If writeable, the Hardware Autonomous Width Disable bit of the Link Control register must be Set in both the Downstream Port and Upstream Port. (If hardwired to 0b, the autonomous width change mechanism is not implemented and is therefore inherently disabled.)

While margining, software must ensure the following:

- All Margin Commands must have the Usage Model field in the Margining Lane Control Register set to 0b. While checking for the status of an outstanding Margin Command, software must check that the Usage Model field of the status part of the Margining Lane Status Register is set to 0b.
- Software must read the capabilities offered by a Receiver and margin it within the constraints of the capabilities it offers. The commands issued and the process followed to determine the margin must be consistent with the definitions provided in Section 4.2.13 and Section 8.4.4 . For example, if the Port does not support voltage testing, then software must not initiate a voltage test. In addition, if a Port supports testing of 2 Lanes simultaneously, then software must test only 1 or 2 Lanes at the same time and not more than 2 Lanes.
- For Receivers where $M_{IndErrorSampler}$ is 1b, any combination of such Receivers are permitted to be margined in parallel.
- For Receivers where $M_{IndErrorSampler}$ is 0b, at most one such Receiver is permitted to be margined at a time. However, margining may be performed on multiple Lanes simultaneously, as long as it is within the maximum number of Lanes the device supports.
- Software must ensure that the Margin Command it provides in the Margining Lane Control Register is a valid one, as defined in Section 4.2.13.1 . For example, the Margin Type must have a defined encoding and the Receiver Number and Margin Payload consistent with it.
- After issuing a command by writing to the Margining Lane Control Register atomically, software must check for the completion of this command. This is done by atomically reading the Margining Lane Status Register and checking that the status fields match the expected response for the issued command (see Table 4-25). If 10 ms has elapsed after a new Margin Command was issued and the values read do not match the expected

response, software is permitted to assume that the Receiver will not respond, and declare that the target Receiver failed margining. For a broadcast command other than No Command the Receiver Number in the response must correspond to one of the Pseudo Ports in Retimer Y or Retimer Z, as described in Figure 4-35 .

- Any two reads of the Margining Lane Status Register should be spaced at least 10 µs apart to make sure they are reading results from different Control SKP Ordered Sets.
- Software must broadcast No Command and wait for it to complete prior to issuing a new Margin Type or Receiver Number or Margin Payload in the Margining Lane Control Register.
- At the end of margining in a given direction (voltage/ timing and up/down/left/right), software must broadcast Go to Normal Settings, No Command, Clear Error Log, and No Command in series in the Downstream and Upstream Ports, after ensuring each command has been acknowledged by the target Receiver.
- If the Data Rate has changed during margining, margining results (if any) are not accurate and software must exit the margining procedure. Software must set the Margining Lane Control Register to No Command to avoid starting margining if the Data Rate later changes to 16.0 GT/s or higher.
- Software is permitted to issue a Clear Error Log command periodically while margining is in progress, to gather error information over a long period of time.
- Software must not attempt to margin both timing and voltage of a target Receiver simultaneously. Results are undefined if a Receiver receives commands that would place both voltage and timing margin locations away from the default sample position at the same time.
- Software should allow margining to run for at least 10^8 bits margined by the Receiver under test before switching to the next margin step location (unless the error limit is exceeded).
- Software must account for the 'set up for margin in progress' status while measuring the margin time or the number of bits sampled by the Receiver.
- If a target Receiver is reporting 'set up for margin in progress' for 200 ms after issuing one of the Step Margin commands, Software is permitted to assume that the Receiver will not respond and declare that the target Receiver failed margining.
- If a Receiver reports a 'NAK' in the Margin Payload status field and the corresponding Step Margin command was valid and within the allowable range (as defined in Section 4.2.13 and Section 8.4.4), Software is permitted to declare that the target Receiver failed margining.
- When the margin testing procedure is completed, the state of the Hardware Autonomous Speed Disable bit and the Hardware Autonomous Width Disable bit must be restored to the previously saved values.

IMPLEMENTATION NOTE

Example Software Flow for Lane Margining at Receiver

For getting the invariant parameters the following steps may be followed. Once obtained, the same parameters can be used across multiple sets of margining tests as long as LinkUp=1b continues to be true. For each component in the Link, do the following Steps. Software can do these steps in parallel for different components on different Lanes of the Link.

***Step A1*:**
Issue Report Margin Control Capabilities (Margin Type = 001b, Margin Payload = 88h, Receiver Number = target device in the Margining Lane Control Register)

***Step A2*:**
Read the Margining Lane Status Register.

a. If Margin Type = 001b and Receiver Number = target Receiver: Go to Step A3

b. Else: If 10 ms has expired since command issued, declare Receiver failed margining and exit; else wait for >10 µs and Go to Step A2

***Step A3*:**
Store the information provided Margin Payload status field for use during margining.

***Step A4*:**
Broadcast No Command (Margin Type = 111b, Receiver Number = 000b, and Margin Payload = 9Ch in the Margining Lane Control Register) and wait for those to be reflected back in the Margining Lane Status Register. If 10 ms expires without getting the command completion handshake, declare the Receiver failed margining and exit.

***Step A5*:**
Repeat Step A1 through Step A4 for Report $M_{NumVoltageSteps}$, Report $M_{NumTimingSteps}$, Report $M_{MaxTimingOffset}$, Report $M_{MaxVoltageOffset}$, Report $M_{SamplingRateVoltage}$, and Report $M_{SamplingRateTiming}$. It may be noted that this step can be executed in parallel across different Lanes for different Margin Type.

Margining on each Lane across the Link can be a sequence of separate commands. Prior to launching the sequence, software should read the maximum number of Lanes it is allowed to run margining simultaneously. The steps would be similar to Step A1 through Step A4 above with the Report $M_{MaxLanes}$ command. After that software can simultaneously margin up to that many Lanes of the Link. On each Link, each Receiver is margined based on its capability, subject to the constraints described here, after ensuring the Link is operating at full width in 16.0 GT/s or higher Data Rate and the hardware autonomous width and speed change as well as ASPM power states have been disabled.

If software desires to set an Error Count Limit value different than default of 4 or whatever was programmed last, it executes the following Steps prior to going to Step C1 below.

***Step B1*:**
Issue Set Error Count Limit (Margin Type = 010b, the target Receiver Number, and Margin Payload = {11b, Error Count Limit} in the Margining Lane Control Register)

***Step B2*:**
Read the Margining Lane Status Register.

a. If Margin Type = 010b, Receiver Number = target Receiver, and Margin Payload = Margin Payload control field (Bits [14:7]), go to Step B4

b. Else: If 10 ms has expired since command issued, go to Step B3; else wait for >10 µs and Go to Step B2

Step B3:
Margining has failed. Invoke the system checks to find out if the Link degraded in width/speed due to reliability reasons.

Step B4:
Broadcast No Command and wait for those to be reflected back in the status fields. If 10 ms expires without getting the command completion handshake, declare the Receiver failed margining and exit.

The following steps is an example flow of one margin point for a given Receiver executing Step Margin to timing offset to right/left of default starting with 15 steps to the right:

Step C1:
Write Margin Type = 011b, the target Receiver Number, and Margin Payload = {0000b, 1111b} in the Margining Lane Control Register

Step C2:
Read the Margining Lane Status Register.

a. If Margin Type = 011b and Receiver Number = target Receiver, Go to Step C3
b. Else If 10 ms has expired since command issued, declare Receiver has failed margining and go to Step C7
c. Wait for >10 µs and Go to Step C2

Step C3:
In the Margining Lane Status Register:

a. If Margin Payload [7:6] = 11b:
   i. If we exceeded the 0.2 UI, that is the margin;
   ii. Else report margin failure at this point and go to Step C7;
b. Else if Margin Payload [7:6] = 00b:
   i. report margin failure at this point and go to Step C7
c. Else if Margin Payload [7:6] = 01b:
   i. If 200 ms has elapsed since entering Step C3, report that the Receiver failed margining test and exit;
   ii. else wait 1 ms, read the Margining Lane Status Register and go to Step C3
d. Else go to Step C4

Step C4:
Wait for the desired amount of time for margining to happen while sampling the Margining Lane Status Register periodically for the number of errors reported in the Margin Payload field (Bits [5:0] - $M_{ErrorCount}$).

For longer runs, issue the No Command follwed by the Clear Error Log commands, (using procedures similar to Step B1 through Step B4, with the corresponding expected status field) if the length of time will cause the error count to exceed the Set Error Count Limit even when staying within the expected BER target.

If the aggregate error count remains within the expected error count and the Margin Payload [7:6] in the status field remains 10b till the end, the Receiver has the required Margin at the timing margin step; else it fails that timing margin step go to Step C7.

***Step C5*:**
Broadcast No Command and wait for those to be reflected back in the status fields. If 10 ms expires without getting the command completion handshake, declare the Receiver failed margining and exit.

***Step C6*:**
Go to Step C1, incrementing the number of timing steps through the Margin Payload control field (Bits[5:0]) if we want to test against a higher margin amount; else go to Step C8 noting the margin value that the Receiver passed

***Step C7*:**
Margin failed; The previous margin step the Receiver passed in Step C6 is the margin of the Receiver

***Step C8*:**
Broadcast No Command, Clear Error Log, No Command, Go to Normal Settings series of commands (using a procedure similar to Step B1 through Step B4 with the corresponding expected status fields)

4.3 Retimers

This Section defines the requirements for Retimers that are Physical Layer protocol aware and that interoperate with any pair of Components with any compliant channel on each side of the Retimer. An important capability of a Physical Layer protocol aware Retimer is to execute the Phase 2/3 of the equalization procedure in each direction. A maximum of two Retimers are permitted between an Upstream and a Downstream Port.

The two Retimer limit is based on multiple considerations, most notably limits on modifying SKP Ordered Sets and limits on the time spent in Phase 2/3 of the equalization procedure. To ensure interoperability, platform designers must ensure that the two Retimer limit is honored for all PCI Express Links, including those involving form factors as well as those involving active cables. Form factor specifications may define additional Retimer rules that must be honored for their form factors. Assessing interoperability with any Extension Device not based on the Retimer definition in this section is outside the scope of this specification.

Many architectures of Extension Devices are possible, i.e., analog only Repeater, protocol unaware Retimer, etc. This specification describes a Physical Layer protocol aware Retimer. It may be possible to use other types of Extension Devices in closed systems if proper analysis is done for the specific channel, Extension Device, and end-device pair - but a specific method for carrying out this analysis is outside the scope of this specification.

Retimers have two Pseudo Ports, one facing Upstream, and the other facing Downstream. The Transmitter of each Pseudo Port must derive its clock from a 100 MHz reference clock. The reference clock(s) must meet the requirements of Section 8.6 . A Retimer supports one or more reference clocking architectures as defined in Section 8.6 Electrical Sub-block.

In most operations Retimers simply forward received Ordered Sets, DLLPs, TLPs, Logical Idle, and Electrical Idle. Retimers are completely transparent to the Data Link Layer and Transaction Layer. System software shall not enable L0s on any Link where a Retimer is present. Support of beacon by Retimers is optional and beyond the scope of this specification.

When using 128b/130b encoding the Retimer executes the protocol so that each Link Segment undergoes independent Link equalization as described in Section 4.3.6 .

The Pseudo Port orientation (Upstream or Downstream), is determined dynamically, while the Link partners are in Configuration. Both crosslink and regular Links are supported.

4.3.1 Retimer Requirements

The following is a high level summary of Retimer requirements:

- Retimers are required to comply with all the electrical specification described in Chapter 8 Electrical Sub-block. Retimers must operate in one of two modes:
  - Retimers' Receivers operate at 8.0 GT/s and above with an impedance that meets the range defined by the Z_{RX-DC} parameter for 2.5 GT/s.
  - Retimers' Receivers operate at 8.0 GT/s and above with an impedance that does not meet the range defined by the Z_{RX-DC} parameter for 2.5 GT/s. In this mode the Z_{RX-DC} parameter for 2.5 GT/s must be met with in 1 ms of receiving an EIOS or inferring Electrical Idle and while the Receivers remain in Electrical Idle.
- Forwarded Symbols must always be de-skewed when more than one Lane is forwarding Symbols (including upconfigure cases).
- Determine Port orientation dynamically.
- Perform Lane polarity inversion (if needed).
- Execute the Link equalization procedure for Phase 2 and Phase 3, when using 128b/130b encoding, on each Link Segment.
- Interoperate with de-emphasis negotiation at 5.0 GT/s, on each Link Segment.
- Interoperate with Link Upconfigure
- Pass loopback data between the Loopback Master and Loopback Slave.
  - Optionally execute Slave Loopback on one Pseudo Port.
- Generate the Compliance Pattern on each Pseudo Port.
  - Load board method (i.e., time out in Polling.Active).
- Forward Modified Compliance Pattern when the Link enters Polling.Compliance via Compliance Receive bit in TS1 Ordered Sets.
- Forward Compliance or Modified Compliance Patterns when Ports enter Polling.Compliance via the Enter Compliance bit in the Link Control 2 register is set to 1b in both the Upstream Port and the Downstream Port and Retimer Enter Compliance is set to 1b (accessed in an implementation specific manner) in the Retimer.
- Adjust the data rate of operation in concert with the Upstream and Downstream Ports of the Link.
- Adjust the Link width in concert with the Upstream and Downstream Ports of the Link.
- Capture Lane numbers during Configuration.
  - Lane numbers are required when using 128b/130b encoding for the scrambling seed.
- Dynamically adjust Retimer Receiver impedance to match end Component Receiver impedance.
- Infer entering Electrical Idle at all data rates.
- Modify certain fields of Ordered Sets while forwarding.
- Perform clock compensation via addition or removal of SKP Symbols.
- Support L1.
  - Optionally Support L1 PM Substates.
- Support Link equalization to the highest data rate.
- Support No Equalization Needed mode.

4.3.2 Supported Retimer Topologies

Figure 4-36 shows the topologies supported by Retimers defined in this specification. There may be one or two Retimers between the Upstream and Downstream Ports on a Link. Each Retimer has two Pseudo Ports, which determine their Downstream/Upstream orientation dynamically. Each Retimer has an Upstream Path and a Downstream Path. Both Pseudo Ports must always operate at the same data rate, when in Forwarding mode. Thus each Path will also be at the same data rate. A Retimer is permitted to support any width option defined by this specification as its maximum width. The behavior of the Retimer in each high level operating mode is:

- Forwarding mode:
  - Symbols, Electrical Idle, and exit from Electrical Idle; are forwarded on each Upstream and Downstream Path.
- Execution mode:
  - The Upstream Pseudo Port acts as an Upstream Port of a Component. The Downstream Pseudo Port acts as a Downstream Port of a Component. This mode is used in the following cases:
    - Polling.Compliance.
    - Phase 2 and Phase 3 of the Link equalization procedure.
    - Optionally Slave Loopback.

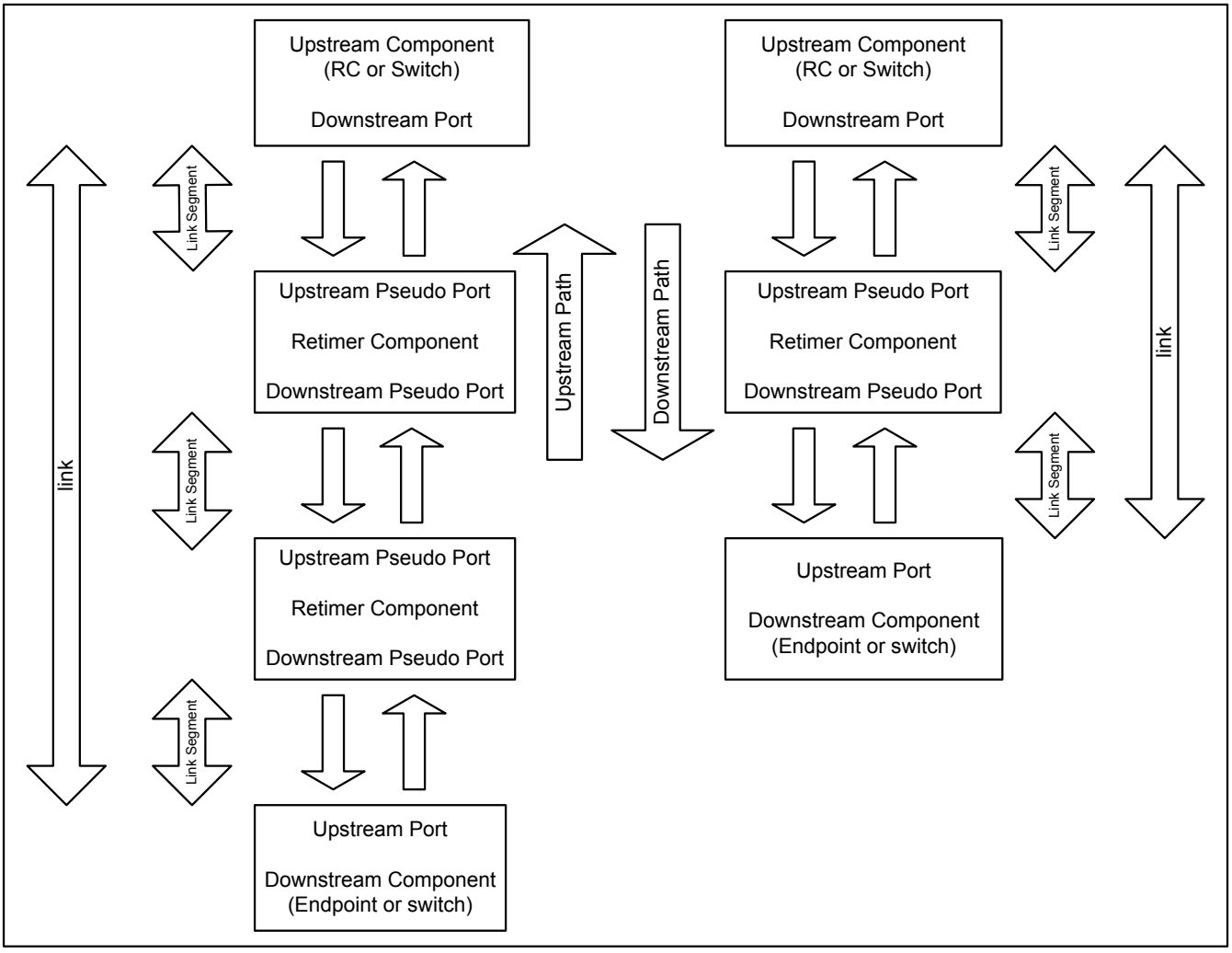

Figure 4-36 Supported Retimer Topologies

4.3.3 Variables

The following variables are set to the following specified values following a Fundamental Reset or whenever the Retimer receives Link and Lane number equal to PAD on two consecutive TS2 Ordered Sets on all Lanes that are receiving TS2 Ordered Sets on both Upstream and Downstream Pseudo Ports within a 1 µs time window from the last Symbol of the second TS2 Ordered Set on the first Lane to the last Symbol of the second TS2 Ordered Set on the last Lane.

- ***RT_port_orientation*** = undefined
- ***RT_captured_lane_number*** = PAD
- ***RT_captured_link_number*** = PAD
- ***RT_G3_EQ_complete*** = 0b
- ***RT_G4_EQ_complete*** = 0b
- ***RT_G5_EQ_complete*** = 0b
- ***RT_LinkUp*** = 0b

- ***RT_next_data_rate*** = 2.5 GT/s
- ***RT_error_data_rate*** = 2.5 GT/s

4.3.4 Receiver Impedance Propagation Rules

The Retimer Transmitters and Receivers shall meet the requirements in Section 4.2.4.9.1 while Fundamental Reset is asserted. When Fundamental Reset is deasserted the Retimer is permitted to take up to 20 ms to begin active determination of its Receiver impedance. During this interval the Receiver impedance remains as required during Fundamental Reset. Once this interval has expired Receiver impedance on Retimer Lanes is determined as follows:

- Within 1.0 ms of the Upstream or Downstream Port's Receiver meeting the Z_{RX-DC} parameter, the low impedance is back propagated, (i.e., the Retimer's Receiver shall meet the Z_{RX-DC} parameter on the corresponding Lane on the other Pseudo Port). Each Lane operates independently and this requirement applies at all times.
- The Retimer must keep its Transmitter in Electrical Idle until the Z_{RX-DC} state has been detected. This applies on an individual Lane basis.

4.3.5 Switching Between Modes

The Retimer operates in two basic modes, Forwarding mode or Execution mode. When switching between these modes the switch must occur on an Ordered Set boundary for all Lanes of the Transmitter at the same time. No other Symbols shall be between the last Ordered Set transmitted in the current mode and the first Symbol transmitted in the new mode.

When using 128b/130b the Transmitter must maintain the correct scrambling seed and LFSR value when switching between modes.

When switching between Forwarding and Execution modes, the Retimer must ensure that at least 16 TS1 Ordered Sets and at most 64 TS1 Ordered Sets are transmitted between the last EIEOS transmitted in the previous mode and the first EIEOS transmitted in the new mode.

When switching to and from the Execution Link Equalization mode the Retimer must ensure a Transmitter does not send two SKP Ordered Sets in a row, and that the maximum allowed interval is not exceeded between SKP Ordered Sets, see Section 4.2.7.3 .

4.3.6 Forwarding Rules

These rules apply when the Retimer is in Forwarding mode. The Retimer is in Forwarding mode after the deassertion of Fundamental Reset.

- If the Retimer's Receiver detects an exit from Electrical Idle on a Lane the Retimer must enter Forwarding mode and forward the Symbols on that Lane to the opposite Pseudo Port as described in Section 4.3.6.3 .
- The Retimer must continue to forward the received Symbols on a given Lane until it enters Execution mode or until an EIOS is received, or until Electrical Idle is inferred on that Lane. This requirement applies even if the Receiver loses Symbol lock or Block Alignment. See Section 4.3.6.5 for rules regarding Electrical Idle entry.
- A Retimer shall forward all Symbols unchanged, except as described in Section 4.3.6.9 and 4.3.6.7.

- When operating at 2.5 GT/s data rate, if any Lane of a Pseudo Port receives TS1 Ordered Sets with Link and Lane numbers set to PAD for 5 ms or longer, and the other Pseudo Port does not detect an exit from Electrical Idle on any Lane in that same window, and either of the following occurs:
  - The following sequence occurs:
    - An EIOS is received on any Lane that was receiving TS1 Ordered Sets
    - followed by a period of Electrical Idle, for less than 5 ms
    - followed by Electrical Idle Exit that cannot be forwarded according to Section 4.3.6.3
    - Note: this is interpreted as the Port attached to the Receiver going into Electrical Idle followed by a data rate change for a Compliance Pattern above 2.5 GT/s.
  - Compliance Pattern at 2.5 GT/s is received on any Lane that was receiving TS1 Ordered Sets.

Then the Retimer enters the Execution mode CompLoadBoard state, and follows Section 4.3.7.1 .

- If any Lane on the Upstream Pseudo Port receives two consecutive TS1 Ordered Sets with the EC field equal to 10b, when using 128b/130b encoding, then the Retimer enters Execution mode Equalization, and follows Section 4.3.7.2 .
- If the Retimer is configured to support Execution mode Slave Loopback and if any Lane on either Pseudo Port receives two consecutive TS1 Ordered Sets or two consecutive TS2 Ordered Sets with the Loopback bit set to 1b then the Retimer enters Execution mode Slave Loopback, and follows Section 4.3.7.3 .

4.3.6.1 Forwarding Type Rules

A Retimer must determine what type of Symbols it is forwarding. The rules for inferring Electrical Idle are a function of the type of Symbols the Retimer is forwarding. If a Path forwards two consecutive TS1 Ordered Sets or two consecutive TS2 Ordered Sets, on any Lane, then the Path is forwarding training sets. If a Path forwards eight consecutive Symbol Times of Idle data on all Lanes that are forwarding Symbols then the Path is forwarding non-training sets. When a Retimer transitions from forwarding training sets to forwarding non-training sets, the variable RT_error_data_rate is set to 2.5 GT/s.

4.3.6.2 Orientation and Lane Numbers Rules

The Retimer must determine the Port orientation, Lane assignment, and Lane polarity dynamically as the Link trains.

- When RT_LinkUp=0, the first Pseudo Port to receive two consecutive TS1 Ordered Sets with a non-PAD Lane number on any Lane, has its RT_port_orientation variable set to Upstream Port, and the other Pseudo Port has its RT_port_orientation variable set to Downstream Port.
- The Retimer plays no active part of Lane number determination. The Retimer must capture the Lane numbers with the RT_captured_lane_number variable at the end of the Configuration state, between the Link Components. This applies on the first time through Configuration, i.e., when RT_LinkUp is set to 0b. Subsequent trips through Configuration during Link width configure must not change the Lane numbers. Lane numbers are required for the scrambling seed when using 128b/130b. Link numbers are required in some cases when the Retimer is in Execution mode. Link numbers and Lane numbers are captured with the RT_captured_lane_number, and RT_captured_link_number variables whenever the first two consecutive TS2 Ordered Sets that contain non-PAD Lane and non-PAD Link numbers are received after RT_LinkUp variable is set to 0b. A Retimer must function normally if Lane reversal occurs. When the Retimer has captured the Lane numbers and Link numbers the variable RT_LinkUp is set to 1b. In addition if the Disable Scrambling bit in the TS2 Ordered Sets is set to 1b, in either case above, then the Retimer determines that scrambling is disabled when using 8b/10b encoding.

- Lane polarity is determined any time the Lane exits Electrical Idle, and achieves Symbol lock at 2.5 GT/s as described in Section 4.2.4.5 :
  - If polarity inversion is determined the Receiver must invert the received data. The Transmitter must never invert the transmitted data.

4.3.6.3 Electrical Idle Exit Rules

At data rates other than 2.5 GT/s, EIEOS are sent within the training sets to ensure that the analog circuit detects an exit from Electrical Idle. Receiving an EIEOS is required when using 128b/130b encoding to achieve Block Alignment. When the Retimer starts forwarding data after detecting an Electrical Idle exit, the Retimer starts transmitting on a training set boundary. The first training sets it forwards must be an EIEOS, when operating at data rates higher than 2.5 GT/s. The first EIEOS sent will be in place of the TS1 or TS2 Ordered Set that it would otherwise forward.

If no Lanes meet Z_{RX-DC} on a Pseudo Port, and the following sequence occurs:

- An exit from Electrical Idle is detected on any Lane of that Pseudo Port.
- And then if not all Lanes infer Electrical Idle, via absence of exit from Electrical Idle in a 12 ms window on that Pseu do Port and the other Pseudo Port is not receiving Ordered Sets on any Lane in that same 12 ms window.

Then the same Pseudo Port, where no Lanes meet Z_{RX-DC}, sends the Electrical Idle Exit pattern described below for 5 µs on all Lanes.

If operating at 2.5 GT/s and the following occurs:

- any Lane detects an exit from Electrical Idle
- and then receives two consecutive TS1 Ordered Sets with Lane and Link numbers equal to PAD
- and the other Pseudo Port is not receiving Ordered Sets on any Lane

Then Receiver Detection is performed on all Lanes of the Pseudo Port that is not receiving Ordered Sets. If no Receivers were detected then:

- The result is back propagated as described in Section 4.3.4 , within 1.0 ms.
- The same Pseudo Port that received the TS1 Ordered Sets with Lane and Link numbers equal to PAD, sends the Electrical Idle Exit pattern described below for 5 µs on all Lanes.

If a Lane detects an exit from Electrical Idle then the Lane must start forwarding when all of the following are true:

- Data rate is determined, see Section 4.3.6.4 , current data rate is changed to RT_next_data_rate if required.
- Lane polarity is determined, see Section 4.3.6.2 .
- Two consecutive TS1 Ordered Sets or two consecutive TS2 Ordered Sets are received.
- Two consecutive TS1 Ordered Sets or two consecutive TS2 Ordered Sets are received on all Lanes that detected an exit from Electrical Idle or the max Retimer Exit Latency has occurred, see Table 4-27 .
- Lane De-skew is achieved on all Lanes that received two consecutive TS1 or two consecutive TS2 Ordered Sets.
- If a data rate change has occurred then 6 µs has elapsed since Electrical Idle Exit was detected.

All Ordered Sets used to establish forwarding must be discarded. Only Lanes that have detected a Receiver on the other Pseudo Port, as described in Section 4.3.4 , are considered for forwarding.

Otherwise after a 3.0 ms timeout, if the other Pseudo Port is not receiving Ordered Sets then Receiver Detection is performed on all Lanes of the Pseudo Port that is not receiving Ordered Sets, the result is back propagated as described in Section 4.3.4 , and if no Receivers were detected:

- Then the same Pseudo Port that was unable to receive two consecutive TS1 or TS2 Ordered Sets on any Lane sends the Electrical Idle Exit pattern described below for 5 µs on all Lanes.
- Else the Electrical Idle Exit pattern described below is forwarded on all Lanes that detected an exit from Electrical Idle.
- When using 128b/130b encoding:
  - One EIEOS
  - 32 Data Blocks, each with a payload of 16 Idle data Symbols (00h), scrambled, for Symbols 0 to 13.
  - Symbol 14 and 15 of each Data Block either contain Idle data Symbols (00h), scrambled, or DC Balance, determined by applying the same rules in Section 4.2.4.1 to these Data Blocks.
- When using 8b/10b encoding:
  - The Modified Compliance Pattern with the error status Symbol set to 00h.
- This Path now is forwarding the Electrical Idle Exit pattern. In this state Electrical Idle is inferred by the absence of Electrical Idle Exit, See Table 4-28 . The Path continues forwarding the Electrical Idle Exit pattern until Electrical Idle is inferred on any lane, or a 48 ms time out occurs. If a 48 ms time out occurs then:
  - RT_LINK_UP is set to 0b
  - The Pseudo Port places its Transmitter in Electrical Idle
  - The RT_next_data_rate and the RT_error_data_rate must be set to 2.5 GT/s for both Pseudo Ports
  - Receiver Detect is performed on the Pseudo Port that was sending the Electrical Idle Exit pattern and timed out, the result is back propagated as described in Section 4.3.4 .

The Transmitter, on the opposite Pseudo Port that was sending the Electrical Idle Exit Pattern and timed out, sends the Electrical Idle Exit Pattern described above for 5 µs.

IMPLEMENTATION NOTE
Electrical Idle Exit

Forwarding Electrical Idle Exit occurs in error cases where a Retimer is unable to decode training sets. Upstream and Downstream Ports use Electrical Idle Exit (without decoding any Symbols) during Polling.Compliance, and Recovery.Speed. If the Retimer does not forward Electrical Idle Exit then the Upstream and Downstream Ports will misbehave in certain conditions. For example, this may occur after a speed change to a higher data rate. In this event forwarding Electrical Idle Exit is required to keep the Upstream and Downstream Ports in lock step at Recovery.Speed, so that the data rate will return to the previous data rate, rather than a Link Down condition from a time out to Detect.

When a Retimer detects an exit from Electrical Idle and starts forwarding data, the time this takes is called the Retimer Exit Latency, and allows for such things as data rate change (if required), clock and data recovery, Symbol lock, Block Alignment, Lane-to-Lane de-skew, Receiver tuning, etc. The maximum Retimer Exit Latency is specified below for several conditions:

- The data rate before and after Electrical Idle and Electrical Idle exit detect does not change.
- Data rate change to a data rate that uses 8b/10b encoding.

- Data rate change to a data rate that uses 128b/130b encoding for the first time.
- Data rate change to a data rate that uses 128b/130b encoding not for the first time.
- How long both transmitters have been in Electrical Idle when a data rate change occurs.

Retimers are permitted to change their data rate while in Electrical Idle, and it is recommended that Retimers start the data rate change while in Electrical Idle to minimize Retimer Exit latency.

Table 4-27 Maximum Retimer Exit Latency

Condition	Link in EI For X µs, where, X < 500 µs	Link in EI for For X ≥ 500 µs
No data rate change	4 µs	4 µs
When forwarding TS1 Ordered Sets at 2.5 GT/s with Lane and Link number equal to PAD.	1 ms	1 ms
Any data rate change to 8b/10b encoding data rate	504 - X µs	4 µs
First data rate change to 128b/130b encoding date rate	1.5 -X ms	1 ms
Subsequent data rate change to 128b/130b encoding date rate	504 -X µs	4 µs

4.3.6.4 Data Rate Change and Determination Rules

The data rate of the Retimer is set to 2.5 GT/s after deassertion of Fundamental Reset.

Both Pseudo Ports of the Retimer must operate at the same data rate. If a Pseudo Port places its Transmitter in Electrical Idle, then the Symbols that it has just completed transmitting determine the variables RT_next_data_rate and RT_error_data_rate. Only when both Pseudo Ports have all Lanes in Electrical Idle shall the Retimer change the data rate. If both Pseudo Ports do not make the same determination of these variables then both variables must be set to 2.5 GT/s.

- If both Pseudo Ports were forwarding non-training sequences, then the RT_next_data_rate must be set to the current data rate. The RT_error_data_rate must be set to 2.5 GT/s. Note: this covers the case where the Link has entered L1 from L0.
- If both Pseudo Ports were forwarding TS2 Ordered Sets with the speed_change bit set to 1b and either:
  - the data rate, when forwarding those TS2s, is greater than 2.5 GT/s or,
  - the highest common data rate received in the data rate identifiers in both directions is greater than 2.5 GT/s,
    then RT_next_data_rate must be set to the highest common data rate and the RT_error_data_rate is set to current data rate. Note: this covers the case where the Link has entered Recovery.Speed from Recovery.RcvrCfg and is changing the data rate according to the highest common data rate.
- Else the RT_next_data_rate must be set to the RT_error_data_rate. The RT_error_data_rate is set to 2.5 GT/s. Note this covers the two error cases:
  - This indicates that the Link was unable to operate at the current data rate (greater than 2.5 GT/s) and the Link will operate at the 2.5 GT/s data rate or,
  - This indicates that the Link was unable to operate at the new negotiated data rate and will revert back to the old data rate with which it entered Recovery from L0 or L1.

4.3.6.5 Electrical Idle Entry Rules

The Rules for Electrical Idle entry in Forwarding mode are a function of whether the Retimer is forwarding training sets or non-training sets. The determination of this is described in Section 4.3.6.1 .

Before a Transmitter enters Electrical Idle, it must always send the Electrical Idle Ordered Set Sequence (EIOSQ), unless otherwise specified.

If the Retimer is forwarding training sets then:

- If an EIOS is received on a Lane, then the EIOSQ is forwarded on that Lane and only that Lane places its Transmitter in Electrical Idle.
- If Electrical Idle is inferred on a Lane, then that Lane places its Transmitter in Electrical Idle, after EIOSQ is transmitted on that Lane.

Else if the Retimer is forwarding non-training sets then:

- If an EIOS is received on any Lane, then the EIOSQ is forwarded on all Lanes that are currently forwarding Symbols and all Lanes place their Transmitters in Electrical Idle.
- If Electrical Idle is inferred on a Lane, then that Lane places its Transmitter in Electrical Idle, and EIOSQ is not transmitted on that Lane.

The Retimer is required to infer Electrical Idle. The criteria for a Retimer inferring Electrical Idle are described in Table 4-28 .

Table 4-28 Inferring Electrical Idle

State	2.5 GT/s	5.0 GT/s	8.0 GT/s	16.0 GT/s or higher
Forwarding: Non Training Sequence	Absence of a SKP Ordered Set in a 128 µs window	Absence of a SKP Ordered Set in a 128 µs window	Absence of a SKP Ordered Set in a 128 µs window	Absence of a SKP Ordered Set in a 128 µs window
Forwarding: Training Sequence	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	Absence of a TS1 or TS2 Ordered Set in a 4680 UI interval
Forwarding: Electrical Idle Exit Executing: Force Timeout	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	Absence of an exit from Electrical Idle in a 16000 UI interval
Forwarding: Loopback Executing: Loopback Slave	Absence of an exit from Electrical Idle in a 128 µs window	N/A	N/A	N/A

4.3.6.6 Transmitter Settings Determination Rules

When a data rate change to 32.0 GT/s occurs the Retimer transmitter settings are determined as follows:

- If the RT_G5_EQ_complete variable is set to 1b:
  - The Transmitter must use the coefficient settings agreed upon at the conclusion of the last equalization procedure applicable to 32.0 GT/s operation.
- Else:
  - An Upstream Pseudo Port must use the 128b/130b Transmitter preset values it registered from the eight consecutive 128b/130b EQ TS2 Ordered Sets received while operating at 16.0 GT/s in its Transmitter preset setting as soon as it starts transmitting at the 32.0 GT/s data rate and must ensure that it meets the preset definition in Section 4.2.3.2 . Lanes that received a Reserved or unsupported Transmitter preset value must use an implementation specific method to choose a supported Transmitter preset setting for use as soon it starts transmitting at 32.0 GT/s.
  - A Downstream Pseudo Port determines its Transmitter Settings in an implementation specific manner when it starts transmitting at 32.0 GT/s.

The RT_G5_EQ_complete variable is set to 1b when:

- Two consecutive TS1 Ordered Sets are received with EC = 01b at 32.0 GT/s.

The RT_G5_EQ_complete variable is set to 0b when any of the following occur:

- RT_LinkUp variable is set to 0b.
- The Pseudo Port is operating at 16.0 GT/s and eight consecutive 128b/130b EQ TS2 Ordered Sets are received on any Lane of the Upstream Pseudo Port. The value in the 128b/130b Transmitter Preset field is registered for later use at 32.0 GT/s for that Lane.

When a data rate change to 16.0 GT/s occurs the Retimer transmitter settings are determined as follows:

- If the RT_G4_EQ_complete variable is set to 1b:
  - The Transmitter must use the coefficient settings agreed upon at the conclusion of the last equalization procedure applicable to 16.0 GT/s operation.
- Else:
  - An Upstream Pseudo Port must use the 128b/130b Transmitter preset values it registered from the received eight consecutive 128b/130b EQ TS2 Ordered Sets in its Transmitter preset setting as soon as it starts transmitting at the 16.0 GT/s data rate and must ensure that it meets the preset definition in Section 8.3.3.3 . Lanes that received a Reserved or unsupported Transmitter preset value must use an implementation specific method to choose a supported Transmitter preset setting for use as soon it starts transmitting at 16.0 GT/s.
  - A Downstream Pseudo Port determines its Transmitter Settings in an implementation specific manner when it starts transmitting at 16.0 GT/s.

The RT_G4_EQ_complete variable is set to 1b when:

- Two consecutive TS1 Ordered Sets are received with EC = 01b at 16.0 GT/s.

The RT_G4_EQ_complete variable is set to 0b when any of the following occur:

- RT_LinkUp variable is set to 0b.

- Eight consecutive 128b/130b EQ TS2 Ordered Sets are received on any Lane of the Upstream Pseudo Port. The value in the 128b/130b Transmitter Preset field is registered for later use at 16.0 GT/s for that Lane.

When a data rate change to 8.0 GT/s occurs the Retimer transmitter settings are determined as follows:

- If the RT_G3_EQ_complete variable is set to 1b:
  - The Transmitter must use the coefficient settings agreed upon at the conclusion of the last equalization procedure applicable to 8.0 GT/s operation.
- Else:
  - An Upstream Pseudo Port must use the 8.0 GT/s Transmitter preset values it registered from the received eight consecutive EQ TS2 Ordered Sets in its Transmitter preset setting as soon as it starts transmitting at the 8.0 GT/s data rate and must ensure that it meets the preset definition in Section 8.3.3 . Lanes that received a Reserved or unsupported Transmitter preset value must use an implementation specific method to choose a supported Transmitter preset setting for use as soon it starts transmitting at 8.0 GT/s. The Upstream Pseudo Port may optionally use the 8.0 GT/s Receiver preset hint values it registered in those EQ TS2 Ordered Sets.
  - A Downstream Pseudo Port determines its Transmitter preset settings in an implementation specific manner when it starts transmitting at 8.0 GT/s.

The RT_G3_EQ_complete variable is set to 1b when:

- Two consecutive TS1 Ordered Sets are received with EC = 01b at 8.0 GT/s.

The RT_G3_EQ_complete variable is set to 0b when any of the following occur:

- RT_LinkUp variable is set to 0b
- Eight consecutive EQ TS1 or eight consecutive EQ TS2 Ordered Sets are received on any Lane of the Upstream Pseudo Port. The value in the 8.0 GT/s Transmitter Preset and optionally the 8.0 GT/s Receiver Preset Hint fields are registered for later use at 8.0 GT/s for that Lane.

When a data rate change to 5.0 GT/s occurs the Retimer transmitter settings are determined as follows:

- The Upstream Pseudo Port must sets its Transmitters to either -3.5 dB or -6.0 dB, according to the Selectable De-emphasis bit (bit 6 of Symbol 4) received in eight consecutive TS2 Ordered Sets, in the most recent series of TS2 Ordered sets, received prior to entering Electrical Idle.
- The Downstream Pseudo Port sets its Transmitters to either -3.5 dB or -6.0 dB in an implementation specific manner.

4.3.6.7 Ordered Set Modification Rules

Ordered Sets are forwarded, and certain fields are modified according to the following rules:

- The Retimer shall not modify any fields except those specifically allowed/required for modification in this specification.
- LF: the Retimer shall overwrite the LF field in TS1 Ordered Sets transmitted in both directions. The new value is determined in an implementation specific manner by the Retimer.
- FS: the Retimer shall overwrite the FS field in TS1 Ordered Sets transmitted in both directions. The new value is determined in an implementation specific manner by the Retimer.

- Pre-Cursor Coefficient: the Retimer shall overwrite the Pre-Cursor Coefficient field in TS1 Ordered Sets transmitted in both directions. The new value is determined by the current Transmitter settings.
- Cursor Coefficient: the Retimer shall overwrite the Cursor Coefficient field in TS1 Ordered Sets transmitted in both directions. The new value is determined by the current Transmitter settings.
- Post-Cursor Coefficient: the Retimer shall overwrite the Post-Cursor Coefficient field in the TS1 Ordered Sets transmitted in both directions. The new value is determined by the current Transmitter settings.
- Parity: the Retimer shall overwrite the Parity bit of forwarded TS1 Ordered Sets. This bit is the even parity of all bits of Symbols 6, 7, and 8 and bits 6:0 of Symbol 9.
- Transmitter Preset: the Retimer shall overwrite the Transmitter Preset field in TS1 Ordered Sets transmitted in both directions. If the Transmitter is using a Transmitter preset setting then the value is equal to the current setting, else it is recommended that the Transmitter Preset field be set to the most recent Transmitter preset setting that was used for the current data rate.

  The Retimer is permitted to do the following:

  - overwrite the Transmitter Preset in EQ TS1 Ordered Sets in either direction
  - overwrite the 8.0 GT/s Transmitter Preset field in EQ TS2 Ordered Sets in the Downstream direction.
  - overwrite the 128b/130b Transmitter Preset field in 128b/130b EQ TS2 Ordered Sets, in the Downstream direction.

    The new values for the 8.0 GT/s Transmitter Preset and 128b/130b Transmitter Preset fields are determined in an implementation specific manner by the Retimer.

    During phase 0 of Equalization to 16.0 GT/s (i.e., the current Data Rate is 8.0 GT/s) or phase 0 of Equalization to 32.0 GT/s (i.e., the current Data Rate is 16.0 GT/s) the Retimer is permitted to do the following in the Upstream direction:

  - Forward received TS2 Ordered Sets.
  - Convert TS2 Ordered Sets to 128b/130b EQ TS2 Ordered Sets, the value for the 128b/130b Transmitter Preset field is determined in an implementation specific manner by the Retimer.
  - Forward received 128b/130b EQ TS2 Ordered Sets with modification, the value for the 128b/130b Transmitter Preset field is determined in an implementation specific manner by the Retimer.
  - Convert 128b/130b EQ TS2 Ordered Sets to TS2 Ordered Sets.
- Receiver Preset Hint: the Retimer is permitted to do the following:
  - overwrite the Receiver Preset Hint in EQ TS1 Ordered Sets in either direction
  - overwrite the 8.0 GT/s Receiver Preset Hint field in EQ TS2 Ordered Sets in the Downstream direction.

    The new values, for the Receiver Preset Hint and 8.0 GT/s Receiver Preset Hint fields are determined in an implementation specific manner by the Retimer.
- SKP Ordered Set: The Retimer is permitted to adjust the length of SKP Ordered Sets transmitted in both directions. The Retimer must perform the same adjustment on all Lanes. When operating with 8b/10b encoding, the Retimer is permitted to add or remove one SKP Symbol of a SKP Ordered Set. When operating with 128b/130b encoding, a Retimer is permitted to add or remove 4 SKP Symbols of a SKP Ordered Set.
- Control SKP Ordered Set: The Retimer must modify the First Retimer Data Parity, or the Second Retimer Data Parity, of the Control SKP Ordered Set when the Retimer is in forwarding mode at 16.0 GT/s or above, according to its received parity. The received even parity is computed independently on each Lane as follows:
  - Parity is initialized when a data rate change occurs.
  - Parity is initialized when a SDS Ordered Set is received.

- Parity is updated with each bit of a Data Block's payload before de-scrambling has been performed.
- Parity is initialized when a Control SKP Ordered Set is received. However, parity is NOT initialized when a Standard SKP Ordered Set is received.
  If a Pseudo Port detects the Retimer Present bit was 0b in the most recently received two consecutive TS2 or EQ TS2 Ordered Sets received by that Pseudo Port when operating at 2.5 GT/s then that Pseudo Port receiver modifies the First Retimer Data Parity as it forwards the Control SKP Ordered Set, else that Pseudo Port receiver modifies the Second Retimer Data Parity as it forwards the Control SKP Ordered Set.

  The Retimer must modify symbols 4*N+1, 4*N+2, and 4*N+3 of the Control SKP Ordered Set in the Upstream direction as described in Section 4.2.13 .

  See Section 4.2.7.2 for Control SKP Ordered Set definition.

- Selectable De-emphasis: the Retimer is permitted to overwrite the Selectable De-emphasis field in the TS1 or TS2 Ordered Set in both directions. The new value is determined in an implementation specific manner by the Retimer.
- The Data Rate Identifier: The Retimer must set the Data Rate Supported bits of the Data Rate Identifier Symbol consistent with the data rates advertised in the received Ordered Sets and its own max supported Data Rate, i.e., it clears to 0b all Symbol 4 bits[5:0] Data Rates that it does not support. A Retimer must support all data rates below and including its maximum supported data rate. A Retimer makes its determination of maximum supported Data Rate once, after fundamental reset.
- DC Balance: When operating with 128b/130b encoding, the Retimer tracks the DC Balance of its Pseudo Port transmitters and transmits DC Balance Symbols as described in Section 4.2.4.1 .
- Retimer Present: When operating at 2.5 GT/s, the Retimer must set the Retimer Present bit of all forwarded TS2 and EQ TS2 Ordered Sets to 1b.
- Two Retimers Present: If the Retimer supports 16.0 GT/s, then when operating at 2.5 GT/s, the Retimer must set the Two Retimers Present bit of all forwarded TS2 and EQ TS2 Ordered Sets if it receives a TS2 or EQ TS2 Ordered Set with the Retimer Present bit set to 1b. If the Retimer does not support 16.0 GT/s, then when operating at 2.5 GT/s, the Retimer is permitted to set the Two Retimers Present bit of all forwarded TS2s and EQ TS2s if it receives a TS2 or EQ TS2 Ordered Sets with the Retimer Present bit set to 1b.
- Loopback: When optionally supporting Slave Loopback in Execution mode, the Loopback bit must be cleared to 0b when forwarding training sets.
- Enhanced Link Behavior Control: If the Retimer supports 32.0 GT/s, then when operating at 2.5GT/s, the Retimer must set the Enhanced Link Behavior Control bits of all forwarded TS1, TS2, EQ TS1 and EQ TS2 Ordered Sets as follows:
  - Set to 11b when Retimer supports Modified TS1/TS2 Ordered Sets and the Enhanced Link Behavior Control bits set to 11b in the Ordered Sets received for forwarding.
  - Set to 10b when Retimer supports no equalization and the Enhanced Link Behavior Control bits is set to 10b in the Ordered Sets received for forwarding.
  - Set to 01b when Retimer supports equalization bypass to the highest rate and the Enhanced Link Behavior Control field is set to 01b in the Ordered Sets received for forwarding.
  - Otherwise, set to 00b.

4.3.6.8 DLLP, TLP, and Logical Idle Modification Rules

DLLPs, TLPs, and Logical Idle are forwarded with no modifications to any of the Symbols unless otherwise specified.

4.3.6.9 8b/10b Encoding Rules

The Retimer shall meet the requirements in Section 4.2.1.1.3 except as follows:

- When the Retimer is forwarding and an 8b/10b decode error or a disparity error is detected in the received data, the Symbol with an error is replaced with the D21.3 Symbol with incorrect disparity in the forwarded data.
- This clause in Section 4.2.1.1.3 does not apply: If a received Symbol is found in the column corresponding to the incorrect running disparity or if the Symbol does not correspond to either column, the Physical Layer must notify the Data Link Layer that the received Symbol is invalid. This is a Receiver Error, and is a reported error associated with the Port (see Section 6.2).

IMPLEMENTATION NOTE

Retimer Transmitter Disparity

The Retimer must modify certain fields of the TS1 and TS2 Ordered Sets (e.g., Receiver Preset Hint, Transmitter Preset), therefore the Retimer must recalculate the running disparity. Simply using the disparity of the received Symbol may lead to an error in the running disparity. For example some 8b/10b codes have 6 ones and 4 zeros for positive disparity, while other codes have 5 ones and 5 zeros.

4.3.6.10 8b/10b Scrambling Rules

A Retimer is required to determine if scrambling is disabled when using 8b/10b encoding as described in Section 4.3.6.2 .

4.3.6.11 Hot Reset Rules

If any Lane of the Upstream Pseudo Port receives two consecutive TS1 Ordered Sets with the Hot Reset bit set to 1b and both the Disable Link and Loopback bits set to 0b, and then both Pseudo Ports either receive an EIOS or infer Electrical Idle on any Lane, that is receiving TS1 Ordered Sets, the Retimer does the following:

- Clears variable RT_LinkUp = 0b.
- Places its Transmitters in Electrical Idle on both Pseudo Ports.
- Set the RT_next_data_rate variable to 2.5 GT/s.
- Set the RT_error_data_rate variable to 2.5 GT/s.
- Waits for an exit from Electrical Idle on every Lane on both Pseudo Ports.

The Retimer does not perform Receiver detection on either Pseudo Port.

4.3.6.12 Disable Link Rules

If any Lane of the Upstream Pseudo Port receives two consecutive TS1 Ordered Sets with the Disable Link bit set to 1b and both the Hot Reset and Loopback bits set to 0b, and then both Pseudo Ports either receive an EIOS or infer Electrical Idle on any Lane, that is receiving TS1 Ordered Sets, the Retimer does the following:

- Clears variable RT_LinkUp = 0b.
- Places its Transmitters in Electrical Idle on both Pseudo Ports.
- Set the RT_next_data_rate variable to 2.5 GT/s.
- Set the RT_error_data_rate variable to 2.5 GT/s.
- Waits for an exit from Electrical Idle on any Lane on either Pseudo Port.

The Retimer does not perform Receiver detection on either Pseudo Port.

4.3.6.13 Loopback

The Retimer follows these additional rules if any Lane receives two consecutive TS1 Ordered Sets with the Loopback bit equal to 1b and both the Hot Reset and Disable Link bits set to 0b and the ability to execute Slave Loopback is not configured in an implementation specific way. The purpose of these rules is to allow interoperation when a Retimer (or two Retimers) exist between a Loopback master and a Loopback slave.

- The Pseudo Port that received the TS1 Ordered Sets with the Loopback bit set to 1b acts as the Loopback Slave (the other Pseudo Port acts as Loopback Master). The Upstream Path is defined as the Pseudo Port that is the Loopback master to the Pseudo Port that is the Loopback slave. The other Path is the Downstream Path.
- Once established, if a Lane loses the ability to maintain Symbol Lock or Block alignment, then the Lane must continue to transmit Symbols while in this state.
- When using 8b/10b encoding and Symbol lock is lost, the Retimer must attempt to re-achieve Symbol Lock.
- When using 128b/130b encoding and Block Alignment is lost, the Retimer must attempt to re-achieve Block Alignment via SKP Ordered Sets.
- If Loopback was entered while the Link Components were in Configuration.Linkwidth.Start, then determine the highest common data rate of the data rates supported by the Link via the data rates received in two consecutive TS1 Ordered Sets or two consecutive TS2 Ordered Sets on any Lane, that was receiving TS1 or TS2 Ordered Sets, at the time the transition to Forwarding.Loopback occurred. If the current data rate is not the highest common data rate, then:
  - Wait for any Lane to receive EIOS, and then place the Transmitters in Electrical Idle for that Path.
  - When all Transmitters are in Electrical Idle, adjust the data rate as previously determined.
  - If the new data rate is 5.0 GT/s, then the Selectable De-emphasis is determined the same as way as described in Section 4.2.6.10.1 .
  - If the new data rate uses 128b/130b encoding, then the Transmitter preset setting is determined the same as way as described in Section 4.2.6.10.1 .
  - In the Downstream Path; wait for Electrical Idle exit to be detected on each Lane and then start forwarding when two consecutive TS1 Ordered Sets have been received, on a Lane by Lane basis. This is considered the first time to this data rate for the Retimer exit latency.
  - In the Upstream Path; if the Compliance Receive bit of the TS1 Ordered Sets that directed the slave to this state was not asserted, then wait for Electrical Idle exit to be detected on each Lane, and start forwarding when two consecutive TS1 Ordered Sets have been received, on a Lane by Lane basis. This is considered the first time to this data rate for the Retimer exit latency.
- In the Upstream Path; if the Compliance Receive bit of the TS1 Ordered Sets that directed the slave to this state was set to 1b, then wait for Electrical Idle exit to be detected on each Lane, and start forwarding immediately, on a Lane by Lane basis. This is considered the first time to this data rate for the Retimer exit latency.
- If four EIOS (one EIOS if the current data rate is 2.5 GT/s) are received on any Lane then:

    - Transmit eight EIOS on every Lane that is transmitting TS1 Ordered Sets on the Pseudo Port that did not receive the EIOS and place the Transmitters in Electrical Idle.
- When both Pseudo Ports have placed their Transmitters in Electrical Idle then:
    - Set the RT_next_data_rate variable to 2.5 GT/s.
    - Set the RT_error_data_rate variable to 2.5 GT/s.
    - The additional rules for Loopback no longer apply unless the rules for entering this Section are met again.

4.3.6.14 Compliance Receive Rules

The Retimer follows these additional rules if any Lane receives eight consecutive TS1 Ordered Sets (or their complement) with the Compliance Receive bit set to 1b and the Loopback bit set to 0b. The purpose of the following rules is to support Link operation with a Retimer when the Compliance Receive bit is Set and the Loopback bit is Clear in TS1 Ordered Sets, transmitted by the Upstream or Downstream Port, while the Link is in Polling.Active.

- Pseudo Port A is defined as the first Pseudo Port that receives eight consecutive TS1 Ordered Sets (or their complement) with the Compliance Receive bit is Set and the Loopback bit is Clear. Pseudo Port B is defined as the other Pseudo Port.
- The Retimer determines the highest common data rate of the Link by examining the data rate identifiers in the TS1 Ordered Sets received on each Pseudo Port, and the max data rate supported by the Retimer.
- If the highest common data rate is equal to 5.0 GT/s then:
    - The Retimer must change its data rate to 5.0 GT/s as described in Section 4.3.6.4 .
    - The Retimer Pseudo Port A must set its de-emphasis according to the selectable de-emphasis bit received in the eight consecutive TS1 Ordered Sets.
    - The Retimer Pseudo Port B must set its de-emphasis in an implementation specific manner.
- If the highest common data rate is equal to 8.0 GT/s or higher then:
    - The Retimer must change its data rate to as applicable, as described in Section 4.3.6.4 .
    - Lane numbers are determined as described in Section 4.2.11 .
    - The Retimer Pseudo Port A must set its Transmitter coefficients on each Lane to the Transmitter preset value advertised in Symbol 6 of the eight consecutive TS1 Ordered Sets and this value must be used by the Transmitter (use of the Receiver preset hint value advertised in those TS1 Ordered Sets is optional). If the common data rate is 8.0 GT/s or higher, any Lanes that did not receive eight consecutive TS1 Ordered Sets with Transmitter preset information can use any supported Transmitter preset setting in an implementation specific manner.
    - The Retimer Pseudo Port B must set its Transmitter and Receiver equalization in an implementation specific manner.
- The Retimer must forward the Modified Compliance Pattern when it has locked to the pattern. This occurs independently on each Lane in each direction. If a Lane's Receiver loses Symbol Lock or Block Alignment, the associated Transmitter (i.e., same Lane on opposite Pseudo Port) Continues to forward data.
- Once locked to the pattern, the Retimer keeps an internal count of received Symbol errors, on a per-Lane basis. The pattern lock and Lane error is permitted to be readable in an implementation specific manner, on a per-Lane basis.
- When operating with 128b/130b encoding, Symbols with errors are forwarded unmodified by default, or may optionally be corrected to remove error pollution. The default behavior must be supported and the method of selecting the optional behavior, if supported, is implementation specific.

- When operating with 8b/10b encoding, Symbols with errors are replaced with the D21.3 Symbol with incorrect disparity by default, or may optionally be corrected to remove error pollution. The default behavior must be supported and the method of selecting the optional behavior, if supported, is implementation specific.
- The error status Symbol when using 8b/10b encoding or the Error_Status field when using 128b/130b encoding is forwarded unmodified by default, or may optionally be redefined as it is transmitted by the Retimer. The default behavior must be supported and the method of selecting the optional behavior, if supported, is implementation specific.
- If any Lane receives an EIOS on either Pseudo Port then:
  - Transmit EIOS on every Lane of the Pseudo Port that did not receive EIOS and place the Transmitters in Electrical Idle. Place the Transmitters of the other Pseudo Port in Electrical Idle; EIOS is not transmitted by the other Pseudo Port.
  - Set the RT_next_data_rate variable to 2.5 GT/s.
  - Set the RT_error_data_rate variable to 2.5 GT/s.
  - The Compliance Receive additional rules no longer apply unless the rules for entering this Section are met again.

4.3.6.15 Enter Compliance Rules

The Retimer follows these additional rules if the Retimer is exiting Electrical Idle after entering Electrical Idle as a result of Hot Reset, and the Retimer Enter Compliance bit is set in the Retimer. The purpose of the following rules is to support Link operation with a Retimer when the Link partners enter compliance as a result of the Enter Compliance bit in the Link Control 2 Register set to 1b in both Link Components and a Hot Reset occurring on the Link. Retimers do not support Link operation if the Link partners enter compliance when they exit detect if the entry into detect was not caused by a Hot Reset.

Retimers must support the following register fields in an implementation specific manner:

- Retimer Target Link Speed
  - One field per Retimer
  - Type = RWS
  - Size = 3 bits
  - Default = 001b
  - Encoding:
    - 001b = 2.5 GT/s
    - 010b = 5.0 GT/s
    - 011b = 8.0 GT/s
    - 100b = 16.0 GT/s
    - 101b - 32.0 GT/s
- Retimer Transmit Margin
  - One field per Pseudo Port
  - Type = RWS
  - Size = 3 bits
  - Default = 000b
  - Encoding:

        - 000b = Normal Operating Range
        - 001b-111b = As defined in Section 8.3.4 , not all encodings are required to be implemented
- Retimer Enter Compliance
    - One bit per Retimer
    - Type = RWS
    - Size = 1 bit
    - Default = 0b
    - Encoding:
        - 0b = do not enter compliance
        - 1b = enter compliance
- Retimer Enter Modified Compliance
    - One bit per Retimer
    - Type = RWS
    - Size = 1 bit
    - Default = 0b
    - Encoding:
        - 0b = do not enter modified compliance
        - 1b = enter modified compliance
- Retimer Compliance SOS
    - One bit per Retimer
    - Type = RWS
    - Size = 1 bit
    - Default = 0b
    - Encoding:
        - 0b = Send no SKP Ordered Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern with 8b/10b encoding.
        - 1b = Send two SKP Ordered Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern with 8b/10b encoding.
- Retimer Compliance Preset/De-emphasis
    - One field per Pseudo Port
    - Type = RWS
    - Size = 4 bits
    - Default = 0000b
    - Encoding when Retimer Target Link Speed is 5.0 GT/s:
        - 0000b -6.0 dB
        - 0001b -3.5 dB
    - Encoding when Retimer Target Link Speed is 8.0 GT/s or higher: the Transmitter Preset.

A Retimer must examine the values in the above registers when the Retimer exits from Hot Reset. If the Retimer Enter Compliance bit is Set the following rules apply:

- The Retimer adjusts its data rate as defined by Retimer Target Link Speed. No data is forwarded until the data rate change has occurred.
- The Retimer configures its Transmitters according to Retimer Compliance Preset/De-emphasis on a per Pseudo Port basis.
- The Retimer must forward the Compliance or Modified Compliance Pattern when it has locked to the pattern. The Retimer must search for the Compliance Pattern if the Retimer Enter Modified Compliance bit is Clear or search for the Modified Compliance Pattern if the Retimer Enter Modified Compliance bit is Set. This occurs independently on each Lane in each direction.
- When using 8b/10b encoding, a particular Lane's Receiver independently determines a successful lock to the incoming Modified Compliance Pattern or Compliance Pattern by looking for any one occurrence of the Modified Compliance Pattern or Compliance Pattern.
  - An occurrence is defined above as the sequence of 8b/10b Symbols defined in Section 4.2.8 .
  - In the case of the Modified Compliance Pattern, the error status Symbols are not to be used for the lock process since they are undefined at any given moment.
  - Lock must be achieved within 1.0 ms of receiving the Modified Compliance Pattern.
- When using 128b/130b encoding each Lane determines Pattern Lock independently when it achieves Block Alignment as described in Section 4.2.2.2.1 .
  - Lock must be achieved within 1.5 ms of receiving the Modified Compliance Pattern or Compliance Pattern.
- When 128b/130b encoding is used, Symbols with errors are forwarded unmodified by default, or may optionally be corrected to remove error pollution. The default behavior must be supported and the method of selecting the optional behavior, if supported, is implementation specific.
- When 8b/10b encoding is used, Symbols with errors are replaced with the D21.3 Symbol with incorrect disparity by default, or may optionally be corrected to remove error pollution. The default behavior must be supported.
- Once locked, the Retimer keeps an internal count of received Symbol errors, on a per-Lane basis. If the Retimer is forwarding the Modified Compliance Pattern then the error status Symbol when using 8b/10b encoding or the Error_Status field when using 128b/130b encoding is forwarded unmodified by default, or may optionally be redefined as it is transmitted by the Retimer. The default behavior must be supported and the method of selecting the optional behavior, if supported, is implementation specific. The Retimer is permitted to make the pattern lock and Lane error information available in an implementation specific manner, on a per-Lane basis.
- If an EIOS is received on any Lane then:
  - All Lanes in that direction transmit 8 EIOS and then all Transmitters in that direction are placed in Electrical Idle.
  - When both directions have sent 8 EIOS and placed their Transmitters in Electrical Idle the data rate is changed to 2.5 GT/s.
  - Set the RT_next_data_rate variable to 2.5 GT/s.
  - Set the RT_error_data_rate variable to 2.5 GT/s.
  - The Retimer Enter Compliance bit and Retimer Enter Modified Compliance bit are both set to 0b.
  - The above additional rules no longer apply unless the rules for entering this Section and clause are met again.

4.3.7 Execution Mode Rules

In Execution mode, Retimers directly control all information transmitted by the Pseudo Ports rather than forwarding information.

4.3.7.1 CompLoadBoard Rules

While the Retimer is in the CompLoadBoard (Compliance Load Board) state both Pseudo Ports are executing the protocol as regular Ports, generating Symbols as specified in the following sub-sections on each Port, rather than forwarding from one Pseudo Port to the other.

> **IMPLEMENTATION NOTE**
> Passive Load on Transmitter
>
> This state is entered when a passive load is placed on one Pseudo Port, and the other Pseudo Port is receiving traffic.

4.3.7.1.1 CompLoadBoard.Entry

- RT_LinkUp = 0b.
- The Pseudo Port that received Compliance Pattern (Pseudo Port A) does the following:
  - The data rate remains at 2.5 GT/s.
  - The Transmitter is placed in Electrical Idle.
  - The Receiver ignores incoming Symbols.
- The other Pseudo Port (Pseudo Port B) does the following:
  - The data rate remains at 2.5 GT/s.
  - The Transmitter is placed in Electrical Idle. Receiver detection is performed on all Lanes as described in Section 8.4.5.7 .
  - The Receiver ignores incoming Symbols.
- If Pseudo Port B's Receiver detection determines there are no Receivers attached on any Lanes, then the next state for both Pseudo Ports is CompLoadBoard.Exit.
- Else the next state for both Pseudo Ports is CompLoadBoard.Pattern.

4.3.7.1.2 CompLoadBoard.Pattern

When The Retimer enters CompLoadBoard.Pattern the following occur:

- Pseudo Port A does the following:
  - The Transmitter remains in Electrical Idle.
  - The Receiver ignores incoming Symbols.

- Pseudo Port B does the following:
  - The Transmitter sends out the Compliance Pattern on all Lanes that detected a Receiver at the data rate and de-emphasis/preset level determined as described in Section 4.2.6.2.2 , (i.e., each consecutive entry into CompLoadBoard advances the pattern), except that the Setting is not set to Setting #1 during Polling.Configuration. Setting #26 and later are not used if Pseudo Port B has received a TS1 or TS2 Ordered Set (or their complement) since the exit of Fundamental Reset. If the new data rate is not 2.5 GT/s, the Transmitter is placed in Electrical Idle prior to the data rate change. The period of Electrical Idle must be greater than 1 ms but it is not to exceed 2 ms.
- If Pseudo Port B detects an Electrical Idle exit of any Lane that detected a Receiver, then the next state for both Pseudo Ports is CompLoadBoard.Exit.

4.3.7.1.3 CompLoadBoard.Exit

When The Retimer enters CompLoadBoard.Exit the following occur:

- The Pseudo Port A:
  - Data rate remains at 2.5 GT/s.
  - The Transmitter sends the Electrical Idle Exit pattern described in Section 4.3.6.3 , on the Lane(s) where Electrical Idle exit was detected on Pseudo Port B for 1 ms. Then the Transmitter is placed in Electrical Idle.
  - The Receiver ignores incoming Symbols.
- Pseudo Port B:
  - If the Transmitter is transmitting at a rate other than 2.5 GT/s the Transmitter sends eight consecutive EIOS.
  - The Transmitter is placed in Electrical Idle. If the Transmitter was transmitting at a rate other than 2.5 GT/s the period of Electrical Idle must be at least 1.0 ms.
  - Data rate is changed to 2.5 GT/s, if not already at 2.5 GT/s.
- Both Pseudo Ports are placed in Forwarding mode.

IMPLEMENTATION NOTE

TS1 Ordered Sets in Forwarding mode

Once in Forwarding mode one of two things will likely occur:

- TS1 Ordered Sets are received and forwarded from Pseudo Port's B Receiver to Pseudo Port's A Transmitter. Link training continues.
- Or: TS1 Ordered Sets are not received because 100 MHz pulses are being received on a lane from the compliance load board, advancing the Compliance Pattern. In this case the Retimer must transition from Forwarding mode to CompLoadBoard when the device attached to Pseudo Port A times out from Polling.Active to Polling.Compliance. The Retimer advances the Compliance Pattern on each entry to CompLoadBoard.

4.3.7.2 Link Equalization Rules

When in the Execution mode performing Link Equalization, the Pseudo Ports act as regular Ports, generating Symbols on each Port rather than forwarding from one Pseudo Port to the other. When the Retimer is in Execution mode it must use the Lane and Link numbers stored in RT_captured_lane_number and RT_captured_link_number.

This mode is entered while the Upstream and Downstream Ports on the Link are in negotiation to enter Phase 2 of the Equalization procedure following the procedure for switching to Execution mode described in Section 4.3.5 .

4.3.7.2.1 Downstream Lanes

The LF and FS values received in two consecutive TS1 Ordered Sets when the Upstream Port is in Phase 1 must be stored for use during Phase 3, if the Downstream Pseudo Port wants to adjust the Upstream Port's Transmitter.

4.3.7.2.1.1 Phase 2

Transmitter behaves as described in Section 4.2.6.4.2.1.2 except as follows:

- If the data rate of operation is 16.0 GT/s or above, the Retimer Equalization Extend bit of the transmitted TS1 Ordered Sets is set to 1b when the Upstream Pseudo Port state is Phase 2 Active, and it is set to 0b when the Upstream Pseudo Port state is Phase 2 Passive.
- Next phase is Phase 3 Active if all configured Lanes receive two consecutive TS1 Ordered Sets with EC=11b.
- Else, next state is Force Timeout after a 32 ms timeout with a tolerance of -0 ms and +4 ms.

4.3.7.2.1.2 Phase 3 Active

If the data rate of operation is 8.0 GT/s then the transmitter behaves as described in Section 4.2.6.4.2.1.3 except the 24 ms timeout is 2.5 ms and as follows:

- Next phase is Phase 3 Passive if all configured Lanes are operating at their optimal settings.
- Else, next state is Force Timeout after a timeout of 2.5 ms with a tolerance of -0 ms and +0.1 ms

If the data rate of operation is 16.0 GT/s or above then the transmitter behaves as described in Section 4.2.6.4.2.1.3 except the 24 ms timeout is 22 ms and as follows:

- The Retimer Equalization Extend bit of transmitted TS1 Ordered Sets is set to 0b.
- Next phase is Phase 3 Passive if all configured Lanes are operating at their optimal settings and all configured Lanes receive two consecutive TS1 Ordered Sets with the Retimer Equalization Extend bit set to 0b.
- Else, next state is Force Timeout after a timeout of 22 ms with a tolerance of -0 ms and +1.0 ms.

4.3.7.2.1.3 Phase 3 Passive

- Transmitter sends TS1 Ordered Sets with EC = 11b, Retimer Equalization Extend = 0b, and the Transmitter Preset field and the Coefficients fields must not be changed from the final value transmitted in Phase 3 Active.
- The transmitter switches to Forwarding mode when the Upstream Pseudo Port exits Phase 3.

4.3.7.2.2 Upstream Lanes

The LF and FS values received in two consecutive TS1 Ordered Sets when the Downstream Port is in Phase 1 must be stored for use during Phase 2, if the Upstream Pseudo Port wants to adjust the Downstream Port's Transmitter.

4.3.7.2.2.1 Phase 2 Active

If the data rate of operation is 8.0 GT/s then the transmitter behaves as described in Section 4.2.6.4.2.2.3 except the 24 ms timeout is 2.5 ms and as follows:

- Next state is Phase 2 Passive if all configured Lanes are operating at their optimal settings.
- Else, next state is Force Timeout after a 2.5 ms timeout with a tolerance of -0 ms and +0.1 ms

If the data rate of operation is 16.0 GT/s or above then the transmitter behaves as described in Section 4.2.6.4.2.2.3 except the 24 ms timeout is 22 ms and as follows:

- The Retimer Equalization Extend bit of transmitted TS1 Ordered Sets is set to 0b.
- Next phase is Phase 2 Passive if all configured Lanes are operating at their optimal settings and all configured Lanes receive two consecutive TS1 Ordered Sets with the Retimer Equalization Extend bit set to 0b.
- Else, next state is Force Timeout after a 22 ms timeout with a tolerance of -0 ms and +1.0 ms.

4.3.7.2.2.2 Phase 2 Passive

- Transmitter sends TS1 Ordered Sets with EC = 10b, Retimer Equalization Extend = 0b, and the Transmitter Preset field and the Coefficients fields must not be changed from the final value transmitted in Phase 2 Active.
- If the data rate of operation is 8.0 GT/s, the next state is Phase 3 when the Downstream Pseudo Port has completed Phase 3 Active.
- If the data rate of operation is 16.0 GT/s or above, the next state is Phase 3 when the Downstream Pseudo Port has started Phase 3 Active.

4.3.7.2.2.3 Phase 3

Transmitter follows Phase 3 rules for Upstream Lanes in Section 4.2.6.4.2.2.4 except as follows:

- If the data rate of operation is 16.0 GT/s or above, the Retimer Equalization Extend bit of the transmitted TS1 Ordered Sets is set to 1b when the Downstream Pseudo Port state is Phase 3 Active, and it is set to 0b when the Downstream Pseudo Port state is Phase 3 Passive.
- If all configured Lanes receive two consecutive TS1 Ordered Sets with EC=00b then the Retimer switches to Forwarding mode.
- Else, next state is Force Timeout after a timeout of 32 ms with a tolerance of -0 ms and +4 ms

4.3.7.2.3 Force Timeout

- The Electrical Idle Exit Pattern described in Section 4.3.6.3 is transmitted by both Pseudo Ports at the current data rate for a minimum of 1.0 ms.
- If on any Lane, a Receiver receives an EIOS or infers Electrical Idle via not detecting an exit from Electrical Idle (see Table 4-28) then, the Transmitters on all Lanes of the opposite Pseudo Port send an EIOSQ and are then placed in Electrical Idle.
- If both Paths have placed their Transmitters in Electrical Idle then, the RT_next_data_rate is set to the RT_error_data_rate, and the RT_error_data_rate is set to 2.5 GT/s, on both Pseudo Ports, and the Retimer enters Forwarding mode.
  - The Transmitters of both Pseudo Ports must be in Electrical Idle for at least 6 µs, before forwarding data.
- Else after a 48 ms timeout, the RT_next_data_rate is set to 2.5 GT/s and the RT_error_data_rate is set to 2.5 GT/s, on both Pseudo Ports, and the Retimer enters Forwarding mode.

IMPLEMENTATION NOTE

Purpose of Force Timeout State

The purpose of this state is to ensure both Link Components are in Recovery.Speed at the same time so they go back to the previous data rate.

4.3.7.3 Slave Loopback

Retimers optionally support Slave Loopback in Execution mode. By default Retimers are configured to forward loopback between Loopback Master and Loopback Slave. Retimers are permitted to allow configuration in an implementation specific manner to act as a Loopback Slave on either Pseudo Port. The other Pseudo Port that is not the Loopback Slave, places its Transmitter in Electrical Idle, and ignores any data on its Receivers.

4.3.7.3.1 Slave Loopback.Entry

The Pseudo Port that did not receive the TS1 Ordered Set with the Loopback bit set to 1b does the following:

- The Transmitter is placed in Electrical Idle.
- The Receiver ignores incoming Symbols.

The Pseudo Port that did receive the TS1 Ordered Set with the Loopback bit set to 1b behaves as the Loopback Slave as described in Section 4.2.6.10.1 with the following exceptions:

- The statement “LinkUp = 0b (False)” is replaced by “RT_LinkUp = 0b”.
- The statement “If Loopback.Entry was entered from Configuration.Linkwidth.Start” is replaced by “If Slave.Loopback.Entry was entered when RT_LinkUp =0b”.
- References to Loopback.Active become Slave Loopback.Active.

4.3.7.3.2 Slave Loopback.Active

The Pseudo Port that did not receive the TS1 Ordered Set with the Loopback bit set to 1b does the following:

- The Transmitter remains in Electrical Idle.
- The Receiver continues to ignore incoming Symbols.

The Pseudo Port that did receive the TS1 Ordered Set with the Loopback bit set to 1b behaves as the Loopback Slave as described in Section 4.2.6.10.2 with the following exception:

- References to Loopback.Exit become Slave Loopback.Exit.

4.3.7.3.3 Slave Loopback.Exit

The Pseudo Port that did not receive the TS1 Ordered Set with the Loopback bit set to 1b must do the following:

- Maintain the Transmitter in Electrical Idle.
- Set the data rate to 2.5 GT/s.
- The Receiver continues to ignore incoming Symbols.

The Pseudo Port that did receive the TS1 or TS2 Ordered Set with the Loopback bit set to 1b must behave as the Loopback Slave as described in Section 4.2.6.10.3 with the following exception:

- The clause “The next state of the Loopback Master and Loopback Slave is Detect” becomes “The Data rate is set to 2.5 GT/s and then both Pseudo Ports are placed in Forwarding mode”.

4.3.8 Retimer Latency

This Section defines the requirements on allowed Retimer Latency.

4.3.8.1 Measurement

Latency must be measured when the Retimer is in Forwarding mode and the Link is in L0, and is defined as the time from when the last bit of a Symbol is received at the input pins of one Pseudo Port to when the equivalent bit is transmitted on the output pins of the other Pseudo Port.

Retimer vendors are strongly encouraged to specify the latency of the Retimer in their data sheets.

Retimers are permitted to have different latencies at different data rates, and when this is the case it is strongly recommended the latency be specified per data rate.

4.3.8.2 Maximum Limit on Retimer Latency

Retimer latency shall be less than the following limit, when not operating in SRIS.

Table 4-29 Retimer Latency Limit not SRIS (Symbol times)

	2.5 GT/s	5.0 GT/s	8.0 GT/s	16.0 GT/s	32.0 GT/s
Maximum Latency	32	32	64	128	256

4.3.8.3 Impacts on Upstream and Downstream Ports

Retimers will add to the channel latency. The round trip delay is 4 times the specified latency when two Retimers are present. It is recommended that designers of Upstream and Downstream Ports consider Retimer latency when determining the following characteristics:

- Data Link Layer Retry Buffer size
- Transaction Layer Receiver buffer size and Flow Control Credits
- Data Link Layer REPLAY_TIMER Limits

Additional buffering (replay or FC) may be required to compensate for the additional channel latency.

4.3.9 SRIS

Retimers are permitted but not required to support SRIS. Retimers that support SRIS must provide a mechanism for enabling the higher rate of SKP Ordered Set transmission, as Retimers must generate SKP Ordered Sets while in Execution mode. Retimers that are enabled to support SRIS will incur additional latency in the elastic store between receive and transmit clock domains. The additional latency is required to handle the case where a Max_Payload_Size TLP is transmitted and SKP Ordered Sets, which are scheduled, are not sent. The additional latency is a function of Link width and Max_Payload_Size. This additional latency is not included in Table 4-29 .

A SRIS capable Retimer must provide an implementation specific mechanism to configure the supported Max_Payload_Size while in SRIS, that must be configured to be greater than or equal to the Max_Payload_Size for the Transmitter in the Port that the Pseudo Port is receiving. Retimer latency must be less than the following limit for the current supported Max_Payload_Size, with SRIS.

Table 4-30 Retimer Latency Limit SRIS (Symbol times)

Max_Payload_Size	2.5 GT/s	5.0 GT/s	8.0 GT/s	16.0 GT/s	32.0 GT/s
128 Bytes	34 (max)	34 (max)	66 (max)	130 (max)	194 (max)
256 Bytes	36 (max)	36 (max)	68 (max)	132 (max)	196 (max)
512 Bytes	39 (max)	39 (max)	71 (max)	135 (max)	199 (max)
1024 Bytes	46 (max)	46 (max)	78 (max)	142 (max)	206 (max)
2048 Bytes	59 (max)	59 (max)	91 (max)	155 (max)	219 (max)
4096 Bytes	86 (max)	86 (max)	118 (max)	182 (max)	246 (max)

IMPLEMENTATION NOTE

Retimer Latency with SRIS Calculation:

Table 4-30 is calculated assuming that the link is operating at x1 Link width. The max Latency is the sum of Table 4-29 and the additional latency required in the elastic store for SRIS clock compensation. The SRIS additional latency in symbol times required for SRIS clock compensation is described in the following equation:

$$2 * \left(\frac{\left(\left(\text{SRIS Link Payload Size+TLP Overhead} \right) / \text{Link Width} \right)}{\text{SKP\_rate}} \right)$$

Equation 4-1 Retimer Latency with SRIS

Where:

SRIS Link Payload Size
is the value programmed in the Retimer.

TLP Overhead
Represents the additional TLP components which consume Link bandwidth (TLP Prefix, header, LCRC, framing Symbols) and is treated here as a constant value of 28 Symbols.

Link Width
The operating width of the Link.

SKP_rate
The rate that a transmitter schedules SKP Ordered Sets when using 8b/10b encoding, 154, see Section 4.2.7.3 . When using the 128b/130b encoding the effective rate is the same.

The nominal latency would be ½ of the SRIS additional latency, and is the nominal fill of the elastic store. This makes a worse case assumption that every blocked SKP Ordered Set requires an additional symbol of latency in the elastic store. When a Max Payload Size TLP is transmitted the actual fill of the elastic store could go to zero, or two times the nominal fill depending on the relative clock frequencies. Link width down configure may occur at any time, a lane fails for example, and this down configure may occur faster than the Retimer is able to adjust its nominal elastic store. By default Retimer's will configure its nominal fill based on x1 link width, regardless of the actual current link width.

Retimers that optionally support SRIS, may optionally support a dynamic elastic store. Dynamic elastic store changes the nominal buffer fill as the link width changes. Retimers are permitted delay the Link LTSSM transitions, only while the Link down configures, in Configuration, for up to 40us. Retimers are permitted to delay the TS1 Order Set to TS2 Ordered Set transition between Configuration.Lanenum.Accept and Configuration.Complete to increase their elastic store.

4.3.10 L1 PM Substates Support

The following Section describes the Retimer's requirements to support the optional L1 PM Substates.

The Retimer enters L1.1 when CLKREQ# is sampled as deasserted. The following occur:

- REFCLK to the Retimer is turned off.
- The PHY remains powered.
- The Retimer places all Transmitters in Electrical Idle on both Pseudo Ports (if not already in Electrical Idle, the expected state). Transmitters maintain their common mode voltage.
- The Retimer must ignore any Electrical Idle exit from all Receivers on both Pseudo Ports.

The Retimer exits L1.1 when CLKREQ# is sampled as asserted. The following occur:

- REFCLK to the Retimer is enabled.
- Normal operation of the Electrical Idle exit circuit is resumed on all Lanes of both Pseudo Ports of the Retimer.
- Normal exit from Electrical Idle exit behavior is resumed, See Section 4.3.6.3 .

Retimers do not support L1.2, but if they support L1.1 and the removal of the reference clock then they must not interfere with the attached components ability to enter L1.2.

Retimer vendors must document specific implementation requirements applying to CLKREQ#. For example, a Retimer implementation that does not support the removal of the reference clock might require an implementation to pull CLKREQ# low.

IMPLEMENTATION NOTE

CLKREQ# Connection Topology with a Retimer Supporting L1 PM Substates

In this platform configuration Downstream Port (A) has only a single CLKREQ# signal. The Upstream and Downstream Ports' CLKREQ# (A and C), and the Retimer's CLKREQB# signals are connected to each other. In this case, Downstream Port (A), must assert CLKREQ# signal whenever it requires a reference clock. Component A, Component B, and the Retimer have their REFCLKs removed/restored at the same time.

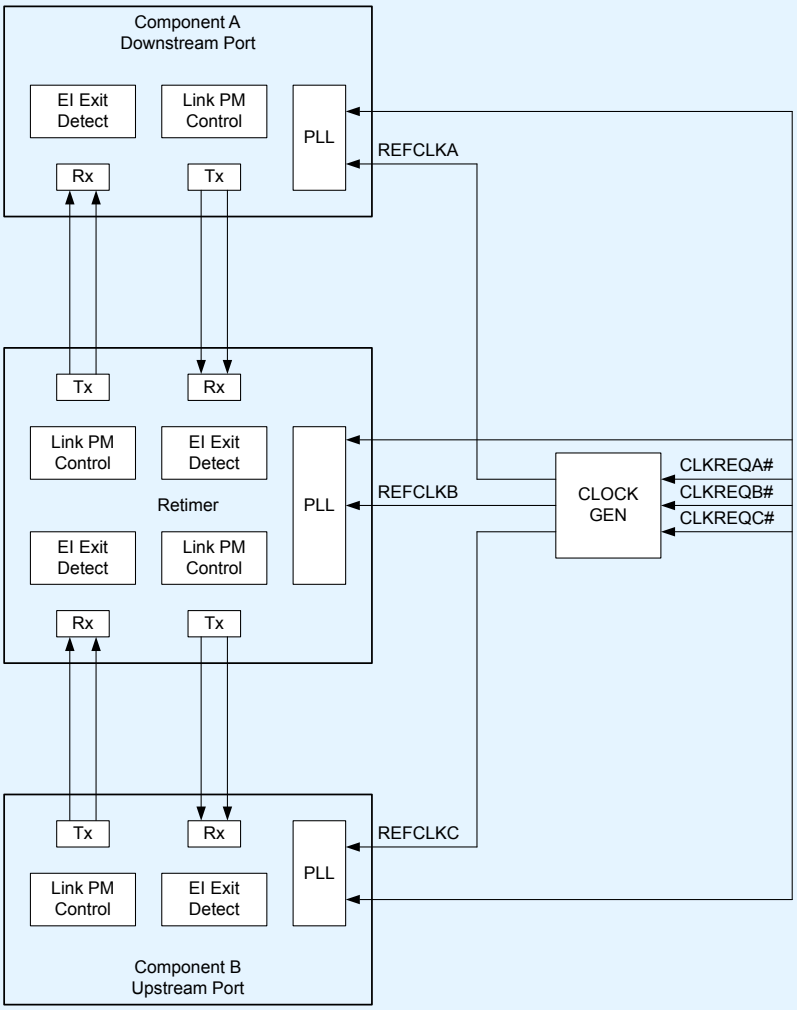

Figure 4-37 Retimer CLKREQ# Connection Topology

4.3.11 Retimer Configuration Parameters

Retimers must provide an implementation specific mechanism to configure each of the parameters in this Section.

The parameters are split into two groups: parameters that are configurable globally for the Retimer and parameters that are configurable for each physical Retimer Pseudo Port.

If a per Pseudo Port parameter only applies to an Upstream or a Downstream Pseudo Port the Retimer is not required to provide an implementation specific mechanism to configure the parameter for the other type of Pseudo Port.

4.3.11.1 Global Parameters

- **Port Orientation Method**. This controls whether the Port Orientation is determined dynamically as described in Section 4.3.6.2 , or statically based on vendor assignment of Upstream and Downstream Pseudo Ports. If the Port Orientation is set to static the Retimer is not required to dynamically adjust the Port Orientation as described in Section 4.3.6.2 . The default behavior is for the Port Orientation to be dynamically determined.
- **Maximum Data Rate**. This controls the maximum data rate that the Retimer sets in the Data Rate Identifier field of training sets that the Retimer transmits. Retimers that support only the 2.5 GT/s speed are permitted not to provide this configuration parameter.
- **SRIS Enable**. This controls whether the Retimer is configured for SRIS and transmits SKP Ordered sets at the SRIS mode rate when in Execution mode. Retimers that do not support SRIS and at least one other clocking architecture are not required to provide this configuration parameter.
- **SRIS Link Payload Size**. This controls the maximum payload size the Retimer supports while in SRIS. The value must be selectable from all the Maximum Payload Sizes shown in Table 4-29 . The default value of this parameter is to support a payload size of 4096 bytes. Retimers that do not support SRIS are not required to provide this configuration parameter.

  The following are example of cases where it might be appropriate to configure the SRIS Link Payload Size to a smaller value than the default:

  - A Retimer is part of a motherboard with a Root Port that supports a maximum payload size less than 4096 bytes.
  - A Retimer is part of an add-in card with an Endpoint that supports a Maximum Payload Size less than 4096 bytes.
  - A Retimer is located Downstream of the Downstream Port of a Switch integrated as part of a system, the Root Port silicon supports a Maximum Payload Size less than 4096 bytes and the system does not support peer to peer traffic.
- **Enhanced Link Behavior Control**. This controls the ability for the Retimer to either bypass equalization to the highest data rate or completely bypass equalization when it supports 32.0 GT/s.

4.3.11.2 Per Physical Pseudo Port Parameters

- **Port Orientation**. This is applicable only when the Port Orientation Method is configured for static determination. This is set for either Upstream or Downstream. Each Pseudo Port must be configured for a different orientation, or the behavior is undefined.
- **Selectable De-emphasis**. When the Downstream Pseudo Port is operating at 5.0 GT/s this controls the transmit de-emphasis of the Link to either -3.5 dB or -6 dB in specific situations and the value of the Selectable De-emphasis field in training sets transmitted by the Downstream Pseudo Port. See Section 4.2.6 for detailed usage information. When the Link Segment is not operating at the 5.0 GT/s speed, the setting of this bit has no effect. Retimers that support only the 2.5 GT/s speed are permitted not to provide this configuration parameter.
- **Rx Impedance Control**. This controls whether the Retimer dynamically applies and removes 50 Ω terminations or statically has 50 Ω terminations present. The value must be selectable from Dynamic, Off, and On. The default behavior is Dynamic.

- **Tx Compliance Disable**. This controls whether the Retimer transmits the Compliance Pattern in the CompLoadBoard.Pattern state. The default behavior is for the Retimer to transmit the Compliance Pattern in the CompLoadBoard.Pattern state. If TX Compliance Pattern is set to disabled, the Retimer Transmitters remain in Electrical Idle and do not transmit Compliance Pattern in CompLoadBoard.Pattern - all other behavior in the CompLoadBoard state is the same.
- **Pseudo Port Slave Loopback**. This controls whether the Retimer operates in a Forwarding mode during loopback on the Link or enters Slave Loopback on the Pseudo Port. The default behavior is for the Retimer to operate in Forwarding mode during loopback. Retimers that do not support optional Slave Loopback are permitted not to provide this configuration parameter. This configuration parameter shall only be enabled for one physical Port. Retimer behavior is undefined if the parameter is enabled for more than one physical Port.
- **Downstream Pseudo Port 8GT TX Preset**. This controls the initial TX preset used by the Downstream Pseudo Port transmitter for 8.0 GT/s transmission. The default value is implementation specific. The value must be selectable from all applicable values in Table 4-4 .
- **Downstream Pseudo Port 16GT TX Preset**. This controls the initial TX preset used by the Downstream Pseudo Port transmitter for 16.0 GT/s transmission. The default value is implementation specific. The value must be selectable from all applicable values in Table 4-4 .
- **Downstream Pseudo Port 32GT TX Preset**. This controls the initial TX preset used by the Downstream Pseudo Port transmitter for 32.0 GT/s transmission. The default value is implementation specific. The value must be selectable for all applicable values in Table 4-4 .
- **Downstream Pseudo Port 8GT Requested TX Preset**. This controls the initial transmitter preset value used in the EQ TS2 Ordered Sets transmitted by the Downstream Pseudo Port for use at 8.0 GT/s. The default value is implementation specific. The value must be selectable from all values in Table 4-4 .
- **Downstream Pseudo Port 16GT Requested TX Preset**. This controls the initial transmitter preset value used in the 128b/130b EQ TS2 Ordered Sets transmitted by the Downstream Pseudo Port for use at 16.0 GT/s. The default value is implementation specific. The value must be selectable from all values in Table 4-4 .
- **Downstream Pseudo Port 32GT Requested TX Preset**. This controls the initial transmitter preset value used in the 128b/130b EQ TS2 Ordered Sets transmitted by the Downstream Pseudo Port for use at 32.0 GT/s. The default value is implementation specific. The value must be selectable from all values in Table 4-4 .
- **Downstream Pseudo Port 8GT RX Hint**. This controls the Receiver Preset Hint value used in the EQ TS2 Ordered Sets transmitted by the Downstream Pseudo Port for use at 8.0 GT/s. The default value is implementation specific. The value must be selectable from all values in Table 4-5 .

4.3.12 In Band Register Access

- Retimers operating at 16.0 GT/s or higher may optionally support inband read only access. Control SKP Ordered Sets at 16.0 GT/s or higher provide the mechanism via the Margin Command ‘Access Retimer Register’, see Table 4-26 . Retimers that support inband read only access must return a non-zero value for the DWORD at Registers offsets 80h and 84h. Retimers that do not support inband read only access must return a zero value.
- Register offsets between A0h and FFh are designated as Vendor Defined register space.
- Register offsets from 00h to 7Fh and 85H to 9Fh are Reserved for PCI-SIG future use.

Power Management

5.

This chapter describes power management (PM) capabilities and protocols.

5.1 Overview

Power Management states are as follows:

- D states are associated with a particular Function
  - D0 is the operational state and consumes the most power
  - D1 and D2 are intermediate power saving states
  - $D3_{Hot}$ is a very low power state
  - $D3_{Cold}$ is the power off state
- L states are associated with a particular Link
  - L0 is the operational state
  - L0s, L1, L1.0, L1.1, and L1.2 are various lower power states

Other specifications define related power states (e.g. S states). This specification does not describe relationships between those states and D/L/B states.

PM provides the following services:

- A mechanism to identify power management capabilities of a given Function
- The ability to transition a Function into a certain power management state
- Notification of the current power management state of a Function
- The option to wakeup the system on a specific event

PM is compatible with the *PCI Bus Power Management Interface Specification*, and the *Advanced Configuration and Power Interface Specification*. This chapter also defines PCI Express native power management extensions.

PM defines Link power management states that a PCI Express physical Link is permitted to enter in response to either software driven D-state transitions or active state Link power management activities. PCI Express Link states are not visible directly to legacy bus driver software, but are derived from the power management state of the components residing on those Links. Defined Link states are L0, L0s, L1, L2, and L3. The power savings increase as the Link state transitions from L0 through L3.

Components may wakeup the system using a wakeup mechanism followed by a power management event (PME) Message. PCI Express systems may provide the optional auxiliary power supply (Vaux) needed for wakeup operation from states where the main power supplies are off.

The specific definition and requirements associated with Vaux are form-factor specific, and throughout this document the terms “auxiliary power” and “Vaux” should be understood in reference to the specific form factor in use.

Another distinction of the PCI Express-PM PME mechanism is its separation of the following two PME tasks:

- Reactivation (wakeup) of the associated resources (i.e., re-establishing reference clocks and main power rails to the PCI Express components)

- Sending a PME Message to the Root Complex

Active State Power Management (***ASPM***) is an autonomous hardware-based, active state mechanism that enables power savings even when the connected components are in the D0 state. After a period of idle Link time, an ASPM Physical-Layer protocol places the idle Link into a lower power state. Once in the lower-power state, transitions to the fully operative L0 state are triggered by traffic appearing on either side of the Link. ASPM may be disabled by software. Refer to Section 5.4.1 for more information on ASPM.

5.2 Link State Power Management

PCI Express defines Link power management states, replacing the bus power management states that were defined by the *PCI Bus Power Management Interface Specification*. Link states are not visible to PCI-PM legacy compatible software, and are either derived from the power management D-states of the corresponding components connected to that Link or by ASPM protocols (see Section 5.4.1).

Note that the PCI Express Physical Layer may define additional intermediate states. Refer to Chapter 4 for more detail on each state and how the Physical Layer handles transitions between states.

PCI Express-PM defines the following Link power management states:

- L0 - Active state.

  L0 support is required for both ASPM and PCI-PM compatible power management.

  All PCI Express transactions and other operations are enabled.

- L0s - A low resume latency, energy saving “standby” state.

  L0s support is optional for ASPM unless the applicable form factor specification for the Link explicitly requires L0s support.

  All main power supplies, component reference clocks, and components' internal PLLs must be active at all times during L0s. TLP and DLLP transmission is disabled for a Port whose Link is in Tx_L0s.

  The Physical Layer provides mechanisms for quick transitions from this state to the L0 state. When common (distributed) reference clocks are used on both sides of a Link, the transition time from L0s to L0 is desired to be less than 100 Symbol Times.

  It is possible for the Transmit side of one component on a Link to be in L0s while the Transmit side of the other component on the Link is in L0.

- L1 - Higher latency, lower power “standby” state.

  L1 support is required for PCI-PM compatible power management. L1 is optional for ASPM unless specifically required by a particular form factor.

  When L1 PM Substates is enabled by setting one or more of the enable bits in the L1 PM Substates Control 1 Register this state is referred to as the L1.0 substate.

  All main power supplies must remain active during L1. As long as they adhere to the advertised L1 exit latencies, implementations are explicitly permitted to reduce power by applying techniques such as, but not limited to, periodic rather than continuous checking for Electrical Idle exit, checking for Electrical Idle exit on only one Lane, and powering off of unneeded circuits. All platform-provided component reference clocks must remain active during L1, except as permitted by Clock Power Management (using CLKREQ#) and/or L1 PM Substates when enabled. A component's internal PLLs may be shut off during L1, enabling greater power savings at a cost of increased exit latency.[79]

The L1 state is entered whenever all Functions of a Downstream component on a given Link are programmed to a D-state other than D0. The L1 state also is entered if the Downstream component requests L1 entry (ASPM) and receives positive acknowledgement for the request.

Exit from L1 is initiated by an Upstream-initiated transaction targeting a Downstream component, or by the Downstream component's initiation of a transaction heading Upstream. Transition from L1 to L0 is desired to be a few microseconds.

TLP and DLLP transmission is disabled for a Link in L1.

- ***L1 PM Substates*** - optional L1.1 and L1.2 substates of the L1 low power Link state for PCI-PM and ASPM.

  In the L1.1 substate, the Link common mode voltages are maintained. The L1.1 substate is entered when the Link is in the L1.0 substate and conditions for entry into L1.1 substate are met. See Section 5.5.1 for details.

  In the L1.2 substate, the Link common mode voltages are not required to be maintained. The L1.2 substate is entered when the Link is in the L1.0 substate and conditions for entry into L1.2 substate are met. See Section 5.5.1 . for details.

  Exit from all L1 PM Substates is initiated when the CLKREQ# signal is asserted (see Section 5.5.2.1 and Section 5.5.3.3).

- ***L2/L3 Ready*** - Staging point for L2 or L3.

  L2/L3 Ready transition protocol support is required.

  L2/L3 Ready is a pseudo-state (corresponding to the LTSSM L2 state) that a given Link enters when preparing for the removal of power and clocks from the Downstream component or from both attached components. This process is initiated after PM software transitions a device into a D3 state, and subsequently calls power management software to initiate the removal of power and clocks. After the Link enters the L2/L3 Ready state the component(s) are ready for power removal. After main power has been removed, the Link will either transition to L2 if Vaux is provided and used, or it will transition to L3 if no Vaux is provided or used. Note that these are PM pseudo-states for the Link; under these conditions, the LTSSM will in, general, operate only on main power, and so will power off with main power removal.

  The L2/L3 Ready state entry transition process must begin as soon as possible following the acknowledgment of a PME_Turn_Off Message, (i.e., the injection of a PME_TO_Ack TLP). The Downstream component initiates L2/L3 Ready entry by sending a PM_Enter_L23 DLLP. Refer to Section 5.7 for further detail on power management system Messages.

  TLP and DLLP transmission is disabled for a Link in L2/L3 Ready.

  Note: Exit from L2/L3 Ready back to L0 will be through intermediate LTSSM states. Refer to Chapter 4 for detailed information.

- L2 - Auxiliary-powered Link, deep-energy-saving state.

  L2 support is optional, and dependent upon the presence of auxiliary power.

  A component may only consume auxiliary power if enabled to do so as described in Section 5.6 .

  In L2, the component's main power supply inputs and reference clock inputs are shut off.

  When in L2, any Link reactivation wakeup logic (Beacon or WAKE#), PME context, and any other “keep alive” logic is powered by auxiliary power.

  TLP and DLLP transmission is disabled for a Link in L2.

- ***L3*** - Link Off state.

79. For example, disabling the internal PLL may be something that is desirable when in $D3_{Hot}$, but not so when in D1 or D2.

When no power is present, the component is in the L3 state.

- ***LDn*** - A transitional Link Down pseudo-state prior to L0.

  This pseudo-state is associated with the LTSSM states Detect, Polling, and Configuration, and, when applicable, Disabled, Loopback, and Hot Reset.

Refer to Section 4.2 for further detail relating to entering and exiting each of the L-states between L0 and L2/L3 Ready (L2.Idle from the Chapter 4 perspective). The L2 state is an abstraction for PM purposes distinguished by the presence of auxiliary power, and should not be construed to imply a requirement that the LTSSM remain active.

The electrical section specifies the electrical properties of drivers and Receivers when no power is applied. This is the L3 state but the electrical section does not refer to L3.

Figure 5-1 shows an overview of L-state transitions that may occur.

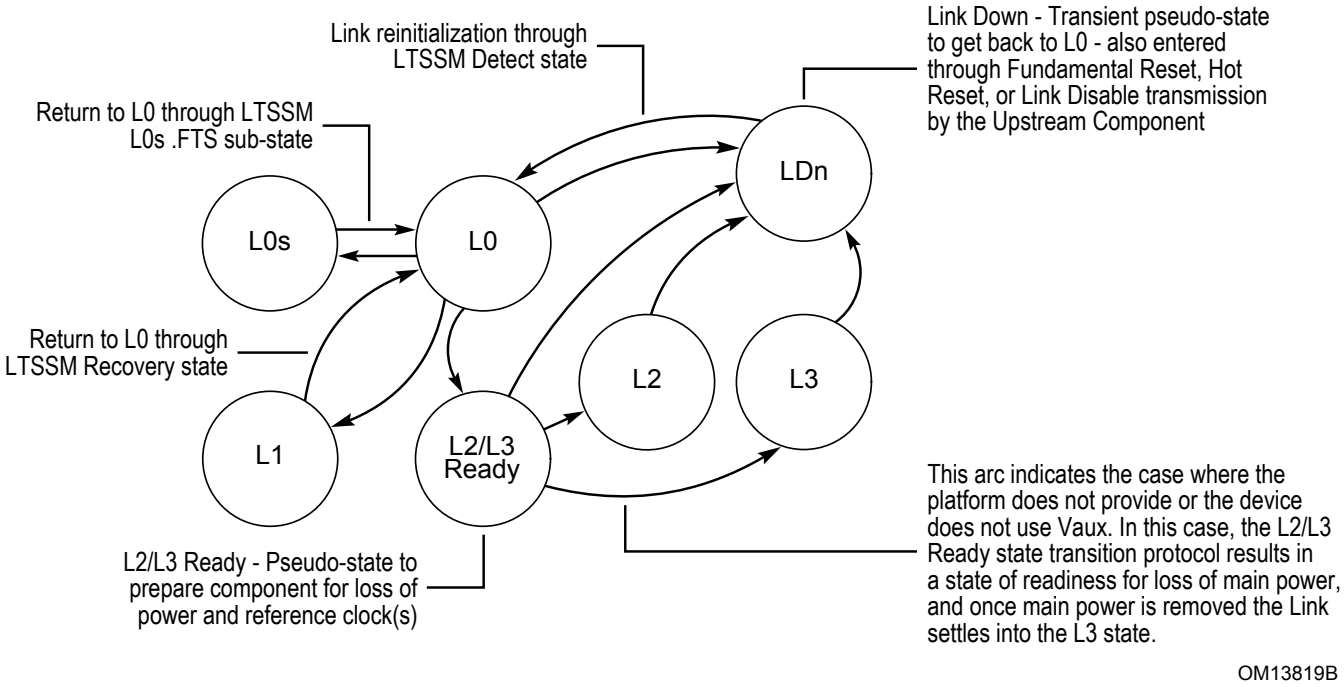

Figure 5-1 Link Power Management State Flow Diagram

The L1 and L2/L3 Ready entry negotiations happen while in the L0 state. L1 and L2/L3 Ready are entered only after the negotiation completes. Link Power Management remains in L0 until the negotiation process is completed, unless LDn occurs. Note that these states and state transitions do not correspond directly to the actions of the Physical Layer LTSSM. For example in Figure 5-1 , L0 encompasses the LTSSM L0, Recovery, and, during LinkUp, Configuration states. Also, the LTSSM is typically powered by main power (not Vaux), so LTSSM will not be powered in either the L2 or the L3 state.

The following example sequence illustrates the multi-step Link state transition process leading up to entering a system sleep state:

1. System software directs all Functions of a Downstream component to $D3_{Hot}$.
2. The Downstream component then initiates the transition of the Link to L1 as required.
3. System software then causes the Root Complex to broadcast the PME_Turn_Off Message in preparation for removing the main power source.

4. This Message causes the subject Link to transition back to L0 in order to send it and to enable the Downstream component to respond with PME_TO_Ack.
5. After sending the PME_TO_Ack, the Downstream component initiates the L2/L3 Ready transition protocol.

L0 → L1 → L0 → L2/L3 Ready

As the following example illustrates, it is also possible to remove power without first placing all Functions into $D3_{Hot}$:

1. System software causes the Root Complex to broadcast the PME_Turn_Off Message in preparation for removing the main power source.
2. The Downstream components respond with PME_TO_Ack.
3. After sending the PME_TO_Ack, the Downstream component initiates the L2/L3 Ready transition protocol.

L0 → L2/L3 Ready

The L1 entry negotiation (whether invoked via PCI-PM or ASPM mechanisms) and the L2/L3 Ready entry negotiation map to a state machine which corresponds to the actions described later in this chapter. This state machine is reset to an idle state. For a Downstream component, the first action taken by the state machine, after leaving the idle state, is to start sending the appropriate entry DLLPs depending on the type of negotiation. If the negotiation is interrupted, for example by a trip through Recovery, the state machine in both components is reset back to the idle state. The Upstream component must always go to the idle state, and wait to receive entry DLLPs. The Downstream componentmust always go to the idle state and must always proceed to sending entry DLLPs to restart the negotiation.

Table 5-1 summarizes each L-state, describing when they are used, and the platform and component behaviors that correspond to each.

A “Yes” entry indicates that support is required (unless otherwise noted). “On” and “Off” entries indicate the required clocking and power delivery. “On/Off” indicates an optional design choice.

Table 5-1 Summary of PCI Express Link Power Management States

	L-State Description	Used by S/W Directed PM	Used by ASPM	Platform Reference Clocks	Platform Main Power	Component Internal PLL	Platform Vaux
L0	Fully active Link	Yes (D0)	Yes (D0)	On	On	On	On/Off
L0s	Standby state	No	Yes [1] (opt., D0)	On	On	On	On/Off
L1	Lower power standby	Yes ($D1$-$D3_{Hot}$)	Yes (opt., D0)	On/Off [6]	On	On/Off [2]	On/Off
L2/L3 Ready (pseudo-state)	Staging point for power removal	Yes [3]	No	On/Off [6]	On	On/Off	On/Off
L2	Low power sleep state (all clocks, main power off)	Yes [4]	No	Off	Off	Off	On [5]
L3	Off (zero power)	n/a	n/a	Off	Off	Off	Off
LDn (pseudo-state)	Transitional state preceding L0	Yes	N/A	On	On	On/Off	On/Off

Notes:

	L-State Description	Used by S/W Directed PM	Used by ASPM	Platform Reference Clocks	Platform Main Power	Component Internal PLL	Platform Vaux

1. L0s exit latency will be greatest in Link configurations with independent reference clock inputs for components connected to opposite ends of a given Link (vs. a common, distributed reference clock).
2. L1 exit latency will be greatest for components that internally shut off their PLLs during this state.
3. L2/L3 Ready entry sequence is initiated at the completion of the PME_Turn_Off/PME_TO_Ack protocol handshake. It is not directly affiliated with either a D-State transition or a transition in accordance with ASPM policies and procedures.
4. Depending upon the platform implementation, the system's sleep state may use the L2 state, transition to fully off (L3), or it may leave Links in the L2/L3 Ready state. L2/L3 Ready state transition protocol is initiated by the Downstream component following reception and TLP acknowledgement of the PME_Turn_Off TLP Message. While platform support for an L2 sleep state configuration is optional (depending on the availability of Vaux), component protocol support for transitioning the Link to the L2/L3 Ready state is required.
5. L2 is distinguished from the L3 state only by the presence and use of Vaux. After the completion of the L2/L3 Ready state transition protocol and before main power has been removed, the Link has indicated its readiness for main power removal.
6. Low-power mobile or handheld devices may reduce power by clock gating the reference clock(s) via the “clock request” (CLKREQ#) mechanism. As a result, components targeting these devices should be tolerant of the additional delays required to re-energize the reference clock during the low-power state exit.

5.3 PCI-PM Software Compatible Mechanisms

5.3.1 Device Power Management States (D-States) of a Function

While the concept of these power states is universal for all Functions in the system, the meaning, or intended functional behavior when transitioned to a given power management state, is dependent upon the type (or class) of the Function.

The D0 power management state is the normal operation state of the Function. Other states are various levels of reduced power, where the Function is either not operating or supports a limited set of operations. D1 and D2 are intermediate states that are intended to afford the system designer more flexibility in balancing power savings, restore time, and low power feature availability tradeoffs for a given device class. The D1 state could, for example, be supported as a slightly more power consuming state than D2, however one that yields a quicker restore time than could be realized from D2.

The D3 power management state constitutes a special category of power management state in that a Function could be transitioned into D3 either by software or by physically removing its power. In that sense, the two D3 variants have been designated as ***D3$_{Hot}$*** and ***D3$_{Cold}$*** where the subscript refers to the presence or absence of main power respectively. Functions in $D3_{Hot}$ are permitted to be transitioned to the D0 state via software by writing to the Function's PMCSR register. Functions in the $D3_{Cold}$ state are permitted to be transitioned to the $D0_{uninitialized}$ state by reapplying main power and asserting Fundamental Reset.

All Functions must support the D0 and D3 states (both $D3_{Hot}$ and $D3_{Cold}$). The D1 and D2 states are optional.

IMPLEMENTATION NOTE
Switch and Root Port Virtual Bridge Behavior in Non-D0 States

When a Type 1 Function associated with a Switch/Root Port (a “virtual bridge”) is in a non-D0 power state, it will emulate the behavior of a conventional PCI bridge in its handling of Memory, I/O, and Configuration Requests and Completions. All Memory and I/O requests flowing Downstream are terminated as Unsupported Requests. All Type 1 Configuration Requests are terminated as Unsupported Requests, however Type 0 Configuration Request handling is unaffected by the virtual bridge D state. Completions flowing in either direction across the virtual bridge are unaffected by the virtual bridge D state.

Note that the handling of Messages is not affected by the PM state of the virtual bridge.

5.3.1.1 D0 State

All Functions must support the D0 state. D0 is divided into two distinct substates, the “un-initialized” substate and the “active” substate. When a component comes out of Conventional Reset all Functions of the component enter the ***D0$_{uninitialized}$*** state. When a Function completes FLR, it enters the $D0_{uninitialized}$ state. After configuration is complete a Function enters the $D0_{active}$ state, the fully operational state for a PCI Express Function. A Function enters the ***D0$_{active}$*** state whenever any single or combination of the Function's Memory Space Enable, I/O Space Enable, or Bus Master Enable bits have been Set[80] .

5.3.1.2 D1 State

D1 support is optional. While in the D1 state, a Function must not initiate any Request TLPs on the Link with the exception of Messages as defined in Section 2.2.8 . Configuration and Message Requests are the only TLPs accepted by a Function in the D1 state. All other received Requests must be handled as Unsupported Requests, and all received Completions may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g., an Unsupported Request) is detected while in D1, and reporting is enabled, the Link must be returned to L0 if it is not already in L0 and an error Message must be sent. If an error caused by an event other than a received TLP (e.g., a Completion Timeout) is detected while in D1, an error Message must be sent when the Function is programmed back to the D0 state.

Note that a Function's software driver participates in the process of transitioning the Function from D0 to D1. It contributes to the process by saving any functional state (if necessary), and otherwise preparing the Function for the transition to D1. As part of this quiescence process the Function's software driver must ensure that any mid-transaction TLPs (i.e., Requests with outstanding Completions), are terminated prior to handing control to the system configuration software that would then complete the transition to D1.

5.3.1.3 D2 State

D2 support is optional. When a Function is not currently being used and probably will not be used for some time, it may be put into D2. This state requires the Function to provide significant power savings while still retaining the ability to fully recover to its previous condition. While in the D2 state, a Function must not initiate any Request TLPs on the Link with the exception of Messages as defined in Section 2.2.8 . Configuration and Message requests are the only TLPs

80. A Function remains in $D0_{active}$ even if these enable bits are subsequently cleared.

accepted by a Function in the D2 state. All other received Requests must be handled as Unsupported Requests, and all received Completions may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g., an Unsupported Request) is detected while in D2, and reporting is enabled, the Link must be returned to L0 if it is not already in L0 and an error Message must be sent. If an error caused by an event other than a received TLP (e.g., a Completion Timeout) is detected while in D2, an error Message must be sent when the Function is programmed back to the D0 state.

Note that a Function's software driver participates in the process of transitioning the Function from D0 to D2. It contributes to the process by saving any functional state (if necessary), and otherwise preparing the Function for the transition to D2. As part of this quiescence process the Function's software driver must ensure that any mid-transaction TLPs (i.e., Requests with outstanding Completions), are terminated prior to handing control to the system configuration software that would then complete the transition to D2.

System software must restore the Function to the $D0_{active}$ state before memory or I/O space can be accessed. Initiated actions such as bus mastering and interrupt request generation can only commence after the Function has been restored to $D0_{active}$.

There is a minimum recovery time requirement of 200 µs between when a Function is programmed from D2 to D0 and the next Request issued to the Function. Behavior is undefined for Requests received in this recovery time window (see Section 7.9.17).

5.3.1.4 D3 State

D3 support is required, (both the $D3_{Cold}$ and the $D3_{Hot}$ states).

Functional context is required to be maintained by Functions in the $D3_{Hot}$ state if the No_Soft_Reset field in the PMCSR is Set. In this case, System Software is not required to re-initialize the Function after a transition from $D3_{Hot}$ to D0 (the Function will be in the $D0_{active}$ state). If the No_Soft_Reset bit is Clear, functional context is not required to be maintained by the Function in the $D3_{Hot}$ state, however it is not guaranteed that functional context will be cleared and software must not depend on such behavior. As a result, in this case System Software is required to fully re-initialize the Function after a transition to D0 as the Function will be in the $D0_{uninitialized}$ state.

The Function will be reset if the Link state has transitioned to the L2/L3 Ready state regardless of the value of the No_Soft_Reset bit.

IMPLEMENTATION NOTE

Transitioning to L2/L3 Ready

As described in Section 5.2 , transition to the L2/L3 Ready state is initiated by platform power management software in order to begin the process of removing main power and clocks from the device. As a result, it is expected that a device will transition to $D3_{Cold}$ shortly after its Link transitions to L2/L3 Ready, making the No_Soft_Reset bit, which only applies to $D3_{Hot}$, irrelevant. While there is no guarantee of this correlation between L2/L3 Ready and $D3_{Cold}$, system software should ensure that the L2/L3 Ready state is entered only when the intent is to remove device main power. Device Functions, including those that are otherwise capable of maintaining functional context while in $D3_{Hot}$ (i.e., set the No_Soft_Reset bit), are required to re-initialize internal state as described in Section 2.9.1 when exiting L2/L3 Ready due to the required DL_Down status indication.

Unless the Immediate_Readiness_on_Return_to_D0 bit in the PCI-PM Power Management Capabilities register is Set, System Software must allow a minimum recovery time following a $D3_{Hot}$ → D0 transition of at least 10 ms (see Section7.9.17), prior to accessing the Function. This recovery time may, for example, be used by the $D3_{Hot}$ → D0

transitioning component to bootstrap any of its component interfaces (e.g., from serial ROM) prior to being accessible. Attempts to target the Function during the recovery time (including configuration request packets) will result in undefined behavior.

5.3.1.4.1 $D3_{Hot}$ State

Configuration and Message requests are the only TLPs accepted by a Function in the $D3_{Hot}$ state. All other received Requests must be handled as Unsupported Requests, and all received Completions may optionally be handled as Unexpected Completions. If an error caused by a received TLP (e.g., an Unsupported Request) is detected while in $D3_{Hot}$, and reporting is enabled, the Link must be returned to L0 if it is not already in L0 and an error Message must be sent. If an error caused by an event other than a received TLP (e.g., a Completion Timeout) is detected while in $D3_{Hot}$, an error Message may optionally be sent when the Function is programmed back to the D0 state. Once in $D3_{Hot}$ the Function can later be transitioned into $D3_{Cold}$ (by removing power from its host component).

Note that a Function's software driver participates in the process of transitioning the Function from D0 to $D3_{Hot}$. It contributes to the process by saving any functional state that would otherwise be lost with removal of main power, and otherwise preparing the Function for the transition to $D3_{Hot}$. As part of this quiescence process the Function's software driver must ensure that any outstanding transactions (i.e., Requests with outstanding Completions), are terminated prior to handing control to the system configuration software that would then complete the transition to $D3_{Hot}$.

Note that $D3_{Hot}$ is also a useful state for reducing power consumption by idle components in an otherwise running system.

Functions that are in $D3_{Hot}$ are permitted to be transitioned by software (writing to their PMCSR PowerState field) to the $D0_{active}$ state or the $D0_{uninitialized}$ state. Functions that are in $D3_{Hot}$ must respond to Configuration Space accesses as long as power and clock are supplied so that they can be returned to D0 by software. Note that the Function is not required to generate an internal hardware reset during or immediately following its transition from $D3_{Hot}$ to D0 (see usage of the No_Soft_Reset bit in the PMCSR).

If not requiring an internal reset, upon completion of the $D3_{Hot}$ to $D0_{active}$ state, no additional operating system intervention is required beyond writing the PowerState field. If the internal reset is required, devices return to $D0_{uninitialized}$ and a full reinitialization is required on the device. The full reinitialization sequence returns the device to $D0_{active}$.

If the device supports PME events, and PME_En is Set, PME context must be preserved in $D3_{Hot}$. PME context must also be preserved in a PowerState command transition back to D0.

> **IMPLEMENTATION NOTE**
>
> **Devices Not Performing an Internal Reset**
>
> Bus controllers to non-PCIe buses and resume from $D3_{Hot}$ bus controllers on PCIe buses that serve as interfaces to non-PCIe buses, (e.g., CardBus, USB, and IEEE 1394) are examples of bus controllers that would benefit from not requiring an internal reset upon resume from $D3_{Hot}$. If this internal reset is not required, the bus controller would not need to perform a downstream bus reset upon resume from $D3_{Hot}$ on its secondary (non-PCIe) bus.

IMPLEMENTATION NOTE
Multi-Function Device Issues with Soft Reset

With Multi-Function Devices (MFDs), certain control settings affecting overall device behavior are determined either by the collective settings in all Functions or strictly off the settings in Function 0. Here are some key examples:

- With non-ARI MFDs, certain controls in the Device Control register and Link Control registers operate off the collective settings of all Functions (see Section 7.5.3.4 and Section 7.5.3.7).
- With ARI Devices, certain controls in the Device Control register and Link Control registers operate strictly off the settings in Function 0 (see Section 7.5.3.4 and Section 7.5.3.7).
- With all MFDs, certain controls in the Device Control 2 and Link Control 2 registers operate strictly off the settings in Function 0 (see Section 7.5.3.16 and Section 7.5.3.19).

Performing a soft reset on any Function (especially Function 0) may disrupt the proper operation of other active Functions in the MFD. Since some Operating Systems transition a given Function between $D3_{Hot}$ and D0 with the expectation that other Functions will not be impacted, it is strongly recommended that every Function in an MFD be implemented with the No_Soft_Reset bit Set in the Power Management Control/Status register. This way, transitioning a given Function from $D3_{Hot}$ to D0 will not disrupt the proper operation of other active Functions.

It is also strongly recommended that every Endpoint Function in an MFD implement Function Level Reset (FLR). FLR can be used to reset an individual Endpoint Function without impacting the settings that might affect other Functions, particularly if those Functions are active. As a result of FLR's quiescing, error recovery, and cleansing for reuse properties, FLR is also recommended for single-Function Endpoint devices.

5.3.1.4.2 $D3_{Cold}$ State

A Function transitions to the $D3_{Cold}$ state when its main power is removed. A power-on sequence with its associated cold reset transitions a Function from the $D3_{Cold}$ state to the $D0_{uninititialized}$ state, and the power-on defaults will be restored to the Function by hardware just as at initial power up. At this point, software must perform a full initialization of the Function in order to re-establish all functional context, completing the restoration of the Function to its $D0_{active}$ state.

Functions that support wakeup functionality from $D3_{Cold}$ must maintain their PME context (in the PMCSR), When PME_En is Set, for inspection by PME service routine software during the course of the resume process. Retention of additional context is implementation specific.

IMPLEMENTATION NOTE
PME Context

Examples of PME context include, but are not limited to, a Function's PME_Status bit, the requesting agent's Requester ID, Caller ID if supported by a modem, IP information for IP directed network packets that trigger a resume event, etc.

A Function's PME assertion is acknowledged when system software performs a “write 1 to clear” configuration transaction to the asserting Function's PME_Status bit of its PCI-PM compatible PMCSR.

An auxiliary power source must be used to support PME event detection within a Function, Link reactivation, and to preserve PME context from within $D3_{Cold}$. Note that once the I/O Hierarchy has been brought back to a fully communicating state, as a result of the Link reactivation, the waking agent then propagates a PME Message to the root of the Hierarchy indicating the source of the PME event. Refer to Section 5.3.3 for further PME specific detail.

5.3.2 PM Software Control of the Link Power Management State

The power management state of a Link is determined by the D-state of its Downstream component.

Table 5-2 depicts the relationships between the power state of a component (with an Upstream Port) and its Upstream Link.

Table 5-2 Relation Between Power Management States of Link and Components

Downstream Component D-State	Permissible Upstream Component D-State	Permissible Interconnect State
D0	D0	L0, L0s, L1 (1), L2/L3 Ready
D1	D0-D1	L1, L2/L3 Ready
D2	D0-D2	L1, L2/L3 Ready
$D3_{Hot}$	D0- $D3_{Hot}$	L1, L2/L3 Ready
$D3_{Cold}$	D0- $D3_{Cold}$	L2 (2), L3

Notes:

1. Requirements for ASPM L0s and ASPM L1 support are form factor specific.
2. If Vaux is provided by the platform, the Link sleeps in L2. In the absence of Vaux, the L-state is L3.

The following rules relate to PCI-PM compatible power management:

- Devices in D0, D1, D2, and $D3_{Hot}$ must respond to the receipt of a PME_Turn_Off Message by the transmission of a PME_TO_Ack Message.
- In any device D state, following the execution of a PME_Turn_Off/PME_TO_Ack handshake sequence, a Downstream component must request a Link transition to L2/L3 Ready using the PM_Enter_L23 DLLP. Following the L2/L3 Ready entry transition protocol the Downstream component must be ready for loss of main power and reference clock.
- The Upstream Port of a single-Function device must initiate a Link state transition to L1 based solely upon its Function being programmed to D1, D2, or $D3_{Hot}$. In the case of the Switch, system software bears the responsibility of ensuring that any D-state programming of a Switch's Upstream Port is done in a compliant manner with respect to hierarchy-wide PM policies (i.e., the Upstream Port cannot be programmed to a D-state that is any less active than the most active Downstream Port and Downstream connected component/ Function(s)).
- The Upstream Port of a non-ARI Multi-Function Device must not initiate a Link state transition to L1 (on behalf of PCI-PM) until all of its Functions have been programmed to a non-D0 D-state.
- The Upstream Port of an ARI Device must not initiate a Link state transition to L1 (on behalf of PCI-PM) until at least one of its Functions has been programmed to a non-D0 state, and all of its Functions are either in a non-D0 state or the $D0_{uninitialized}$ state.

5.3.2.1 Entry into the L1 State

Figure 5-2 depicts the process by which a Link transitions into the L1 state as a direct result of power management software programming the Downstream connected component into a lower power state, (either D1, D2, or $D3_{Hot}$ state). This figure and the subsequent description outline the transition process for a single -Function Downstream component that is being programmed to a non-D0 state.

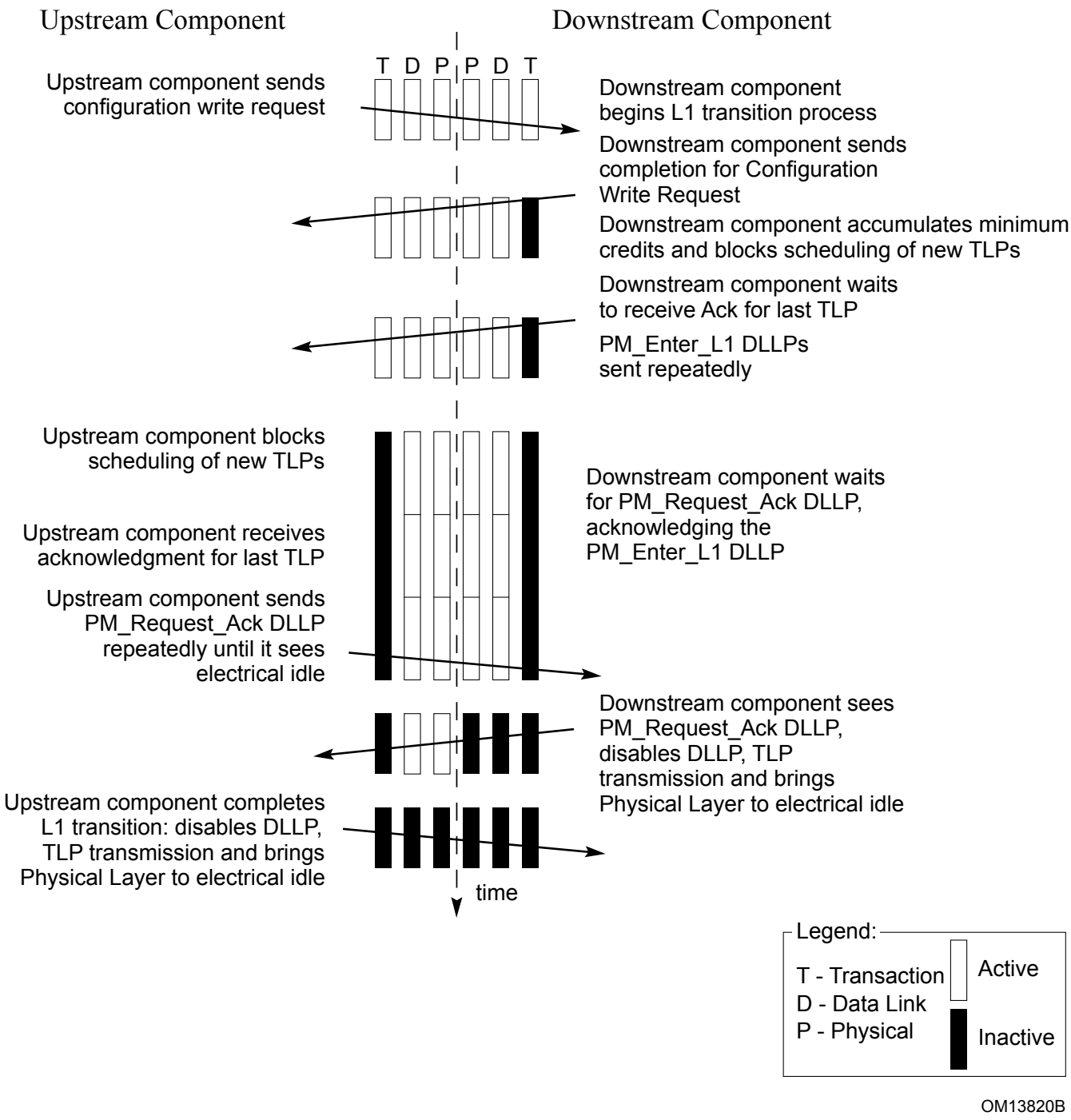

Figure 5-2 Entry into the L1 Link State

The following text provides additional detail for the Link state transition process shown in Figure 5-2 .

PM Software Request:

1. PM software sends a Configuration Write Request TLP to the Downstream Function's PMCSR to change the Downstream Function's D-state (from D0 to D1 for example).

Downstream Component Link State Transition Initiation Process:

2. The Downstream component schedules the Completion corresponding to the Configuration Write Request to its PMCSR PowerState field and accounts for the completion credits required.
3. The Downstream component must then wait until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type for all enabled VCs (if it does not already have such credits). All Transaction Layer TLP scheduling is then suspended.
4. The Downstream component then waits until it receives a Link Layer acknowledgement for the PMCSR Write Completion, and any other TLPs it had previously sent. The component must retransmit a TLP out of its Data Link Layer Retry buffer if required to do so by Data Link Layer rules.
5. Once all of the Downstream components' TLPs have been acknowledged, the Downstream component starts to transmit PM_Enter_L1 DLLPs. The Downstream component sends this DLLP repeatedly with no more than eight (when using 8b/10b encoding) or 32 (when using 128b/130b encoding) Symbol times of idle between subsequent transmissions of the PM_Enter_L1 DLLP. The transmission of other DLLPs and SKP Ordered Sets is permitted at any time between PM_Enter_L1 transmissions, and do not contribute to this idle time limit.

   The Downstream component continues to transmit the PM_Enter_L1 DLLP as described above until it receives a response from the Upstream component[81] (PM_Request_Ack).

   The Downstream component must continue to accept TLPs and DLLPs from the Upstream component, and continue to respond with DLLPs, including FC update DLLPs and Ack/Nak DLLPs, as required. Any TLPs that are blocked from transmission (including responses to TLP(s) received) must be stored for later transmission, and must cause the Downstream component to initiate L1 exit as soon as possible following L1 entry.

Upstream Component Link State Transition Process:

6. Upon receiving the PM_Enter_L1 DLLP, the Upstream component blocks the scheduling of all TLP transmissions.
7. The Upstream component then must wait until it receives a Link Layer acknowledgement for the last TLP it had previously sent. The Upstream component must retransmit a TLP from its Link Layer retry buffer if required to do so by the Link Layer rules.
8. Once all of the Upstream component's TLPs have been acknowledged, the Upstream component must send PM_Request_Ack DLLPs Downstream, regardless of any outstanding Requests. The Upstream component sends this DLLP repeatedly with no more than eight (when using 8b/10b encoding) or 32 (when using 128b/130b encoding) Symbol times of idle between subsequent transmissions of the PM_Request_Ack DLLP. The transmission of SKP Ordered Sets is permitted at any time between PM_Request_Ack transmissions, and does not contribute to this idle time limit.

   The Upstream component continues to transmit the PM_Request_Ack DLLP as described above until it observes its receive Lanes enter into the Electrical Idle state. Refer to Chapter 4 for more details on the Physical Layer behavior.

Completing the L1 Link State Transition:

9. Once the Downstream component has captured the PM_Request_Ack DLLP on its Receive Lanes (signaling that the Upstream component acknowledged the transition to L1 request), it then disables DLLP transmission and brings the Upstream directed physical Link into the Electrical Idle state.
10. When the Receive Lanes on the Upstream component enter the Electrical Idle state, the Upstream component stops sending PM_Request_Ack DLLPs, disables DLLP transmission, and brings its Transmit Lanes to Electrical Idle completing the transition of the Link to L1.

81. If at this point the Downstream component needs to initiate a transfer on the Link, it must first complete the transition to L1. Once in L1 it is then permitted to initiate an exit L1 to handle the transfer.

When two components' interconnecting Link is in L1 as a result of the Downstream component being programmed to a non-D0 state, both components suspend the operation of their Flow Control Update and, if implemented, Update FCP Timer (see Section 2.6.1.2) counter mechanisms. Refer to Chapter 4 for more detail on the Physical Layer behavior.

Refer to Section 5.2 if the negotiation to L1 is interrupted.

Components on either end of a Link in L1 may optionally disable their internal PLLs in order to conserve more energy. Note, however, that platform supplied main power and reference clocks must continue to be supplied to components on both ends of an L1 Link in the L1.0 substate of L1.

Refer to Section 5.5 for entry into the L1 PM Substates.

5.3.2.2 Exit from L1 State

L1 exit can be initiated by the component on either end of a Link.

Upon exit from L1, it is recommended that the Downstream component send flow control update DLLPs for all enabled VCs and FC types starting within 1 µs of L1 exit.

The physical mechanism for transitioning a Link from L1 to L0 is described in detail in Chapter 4 .

L1 exit must be initiated by a component if that component needs to transmit a TLP on the Link. An Upstream component must initiate L1 exit on a Downstream Port even if it does not have the flow control credits needed to transmit the TLP that it needs to transmit. Following L1 exit, the Upstream component must wait to receive the needed credit from the Downstream component. Figure 5-3 outlines an example sequence that would trigger an Upstream component to initiate transition of the Link to the L0 state.

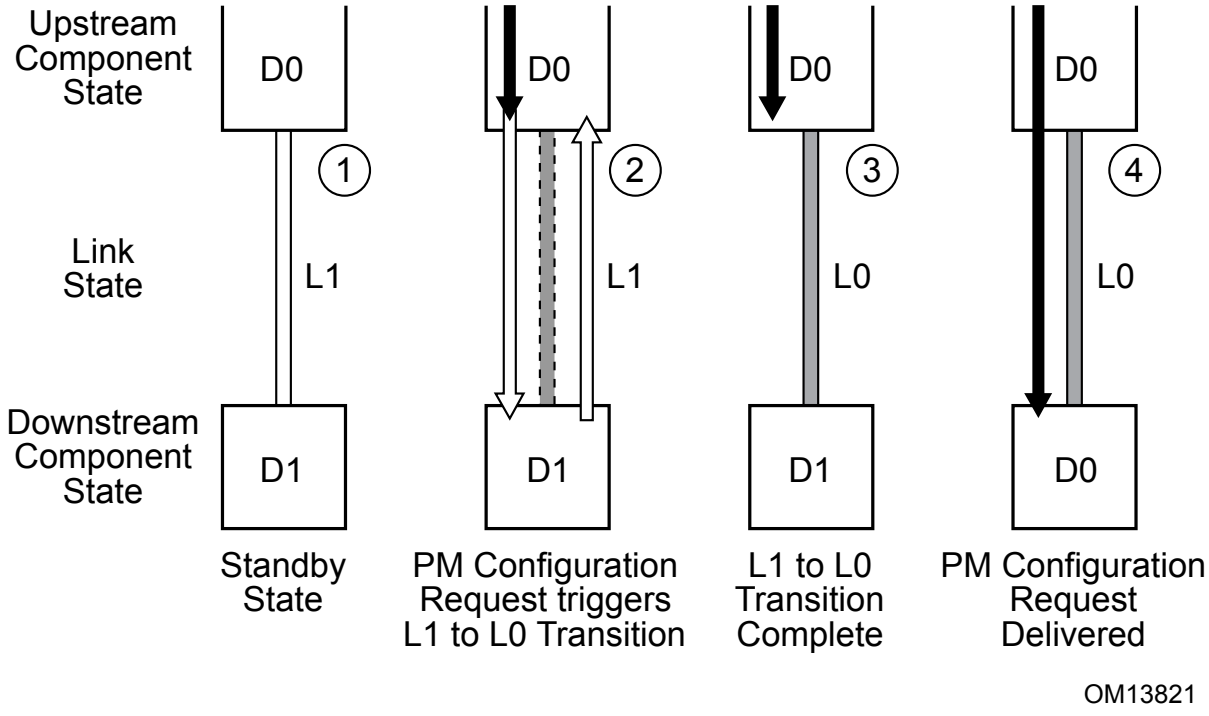

Figure 5-3 Exit from L1 Link State Initiated by Upstream Component

Sequence of events:

1. Power management software initiates a configuration cycle targeting a PM configuration register (the PowerState field of the PMCSR in this example) within a Function that resides in the Downstream component (e.g., to bring the Function back to the D0 state).
2. The Upstream component detects that a configuration cycle is intended for a Link that is currently in a low power state, and as a result, initiates a transition of that Link into the L0 state.
3. If the Link is in either L1.1 or L1.2 substates of L1, then the Upstream component initiates a transition of the Link into the L1.0 substate of L1.
4. In accordance with the Chapter 4 definition, both directions of the Link enter into Link training, resulting in the transition of the Link to the L0 state. The L1 → L0 transition is discussed in detail in Chapter 4 .
5. Once both directions of the Link are back to the active L0 state, the Upstream Port sends the configuration Packet Downstream.

5.3.2.3 Entry into the L2/L3 Ready State

Transition to the L2/L3 Ready state follows a process that is similar to the L1 entry process. There are some minor differences between the two that are spelled out below.

- L2/L3 Ready entry transition protocol does not immediately result in an L2 or L3 Link state. The transition to L2/L3 Ready is effectively a handshake to establish the Downstream component's readiness for power removal. L2 or L3 is ultimately achieved when the platform removes the components' power and reference clock.
- The time for L2/L3 Ready entry transition is indicated by the completion of the PME_Turn_Off/PME_TO_Ack handshake sequence. Any actions on the part of the Downstream component necessary to ready itself for loss of power must be completed prior to initiating the transition to L2/L3 Ready. Once all preparations for loss of power and clock are completed, L2/L3 Ready entry is initiated by the Downstream component by sending the PM_Enter_L23 DLLP Upstream.
- L2/L3 Ready entry transition protocol uses the PM_Enter_L23 DLLP.

Note that the PM_Enter_L23 DLLPs are sent continuously until an acknowledgement is received or power is removed.

- Refer to Section 5.2 if the negotiation to L2/L3 Ready is interrupted.

5.3.3 Power Management Event Mechanisms

5.3.3.1 Motivation

The PCI Express PME mechanism is software compatible with the PME mechanism defined by the *PCI Bus Power Management Interface Specification*. Power Management Events are generated by Functions as a means of requesting a PM state change. Power Management Events are typically utilized to revive the system or an individual Function from a low power state.

Power management software may transition a Hierarchy into a low power state, and transition the Upstream Links of these devices into the non-communicating L2 state.[82] The PCI Express PME generation mechanism is, therefore, broken into two components:

82. The L2 state is defined as “non-communicating” since component reference clock and main power supply are removed in that state.

- Waking a non-communicating Hierarchy (wakeup). This step is required only if the Upstream Link of the device originating the PME is in the non-communicating L2 state, since in that state the device cannot send a PM_PME Message Upstream.
- Sending a PM_PME Message to the root of the Hierarchy

PME indications that originate from PCI Express Endpoints or PCI Express Legacy Endpoints are propagated to the Root Complex in the form of TLP messages. PM_PME Messages identify the requesting agent within the Hierarchy (via the Requester ID of the PME Message header). Explicit identification within the PM_PME Message is intended to facilitate quicker PME service routine response, and hence shorter resume time.

If an RCiEP is associated with a Root Complex Event Collector, any PME indications that originate from that RCiEP must be reported by that Root Complex Event Collector.

PME indications that originate from a Root Port itself are reported through the same Root Port.

5.3.3.2 Link Wakeup

The Link wakeup mechanisms provide a means of signaling the platform to re-establish power and reference clocks to the components within its domain. There are two defined wakeup mechanisms: Beacon and WAKE#. The Beacon mechanism uses in-band signaling to implement wakeup functionality. For components that support wakeup functionality, the form factor specification(s) targeted by the implementation determine the support requirements for the wakeup mechanism. Switch components targeting applications where Beacon is used on some Ports of the Switch and WAKE# is used for other Ports must translate the wakeup mechanism appropriately (see the implementation note entitled “Example of WAKE# to Beacon Translation”in Section, 5.3.3.2). In applications where WAKE# is the only wakeup mechanism used, the Root Complex is not required to support the receipt of Beacon.

The WAKE# mechanism uses sideband signaling to implement wakeup functionality. WAKE# is an “open drain” signal asserted by components requesting wakeup and observed by the associated power controller. WAKE# is only defined for certain form factors, and the detailed specifications for WAKE# are included in the relevant form factor specifications. Specific form factor specifications may require the use of either Beacon or WAKE# as the wakeup mechanism.

When WAKE# is used as a wakeup mechanism, once WAKE# has been asserted, the asserting Function must continue to drive the signal low until main power has been restored to the component as indicated by Fundamental Reset going inactive.

The system is not required to route or buffer WAKE# in such a way that an Endpoint is guaranteed to be able to detect that the signal has been asserted by another Function.

Before using any wakeup mechanism, a Function must be enabled by software to do so by setting the Function's PME_En bit in the PMCSR. The PME_Status bit is sticky, and Functions must maintain the value of the PME_Status bit through reset if auxiliary power is available and they are enabled for wakeup events (this requirement also applies to the PME_En bit in the PMCSR and the Aux Power PM Enable bit in the Device Control Register).

Systems that allow PME generation from $D3_{Cold}$ state must provide auxiliary power to support Link wakeup when the main system power rails are off. A component may only consume auxiliary power if software has enabled it to do so as described in Section 5.6 . Software is required to enable auxiliary power consumption in all components that participate in Link wakeup, including all components that must propagate the Beacon signal. In the presence of legacy system software, this is the responsibility of system firmware.

Regardless of the wakeup mechanism used, once the Link has been re-activated and trained, the requesting agent then propagates a PM_PME Message Upstream to the Root Complex. From a power management point of view, the two wakeup mechanisms provide the same functionality, and are not distinguished elsewhere in this chapter.

IMPLEMENTATION NOTE

Example of WAKE# to Beacon Translation

Switch components targeting applications that connect “Beacon domains” and “WAKE# domains” must translate the wakeup mechanism appropriately. Figure 5-4 shows two example systems, each including slots that use the WAKE# wakeup mechanism. In Case 1, WAKE# is input directly to the Power Management Controller, and no translation is required. In Case 2, WAKE# is an input to the Switch, and in response to WAKE# being asserted the Switch must generate a Beacon that is propagated to the Root Complex/Power Management Controller.

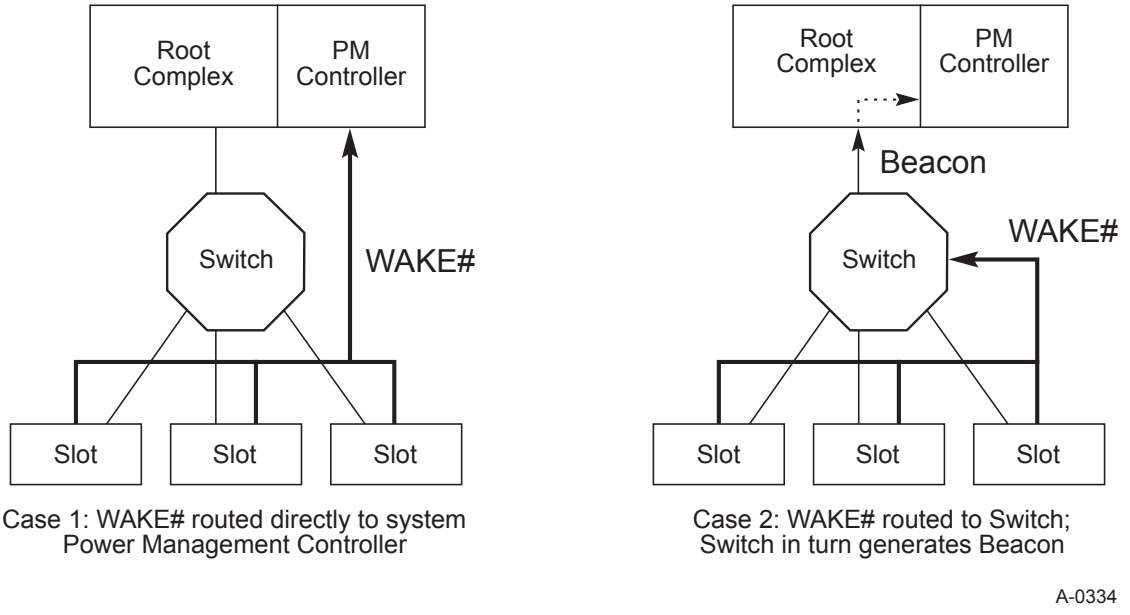

Figure 5-4 Conceptual Diagrams Showing Two Example Cases of WAKE# Routing

5.3.3.2.1 PME Synchronization

PCI Express-PM introduces a fence mechanism that serves to initiate the power removal sequence while also coordinating the behavior of the platform's power management controller and PME handling by PCI Express agents.

PME_Turn_Off Broadcast Message

Before main component power and reference clocks are turned off, the Root Complex or Switch Downstream Port must issue a broadcast Message that instructs all agents Downstream of that point within the hierarchy to cease initiation of any subsequent PM_PME Messages, effective immediately upon receipt of the PME_Turn_Off Message.

Each PCI Express agent is required to respond with a TLP “acknowledgement” Message, PME_TO_Ack that is always routed Upstream. In all cases, the PME_TO_Ack Message must terminate at the PME_Turn_Off Message's point of origin.[83]

83. Point of origin for the PME_Turn_Off Message could be all of the Root Ports for a given Root Complex (full platform sleep state transition), an individual Root Port, or a Switch Downstream Port.

A Switch must report an “aggregate” acknowledgement only after having received PME_TO_Ack Messages from each of its Downstream Ports. Once a PME_TO_Ack Message has arrived on each Downstream Port, the Switch must then send a PME_TO_Ack packet on its Upstream Port. The occurrence of any one of the following must reset the aggregation mechanism: the transmission of the PME_TO_Ack Message from the Upstream Port, the receipt of any TLP at the Upstream Port, the removal of main power to the Switch, or Fundamental Reset.

All components with an Upstream Port must accept and acknowledge the PME_Turn_Off Message regardless of the D state of the associated device or any of its Functions for a Multi-Function Device. Once a component has sent a PME_TO_Ack Message, it must then prepare for removal of its power and reference clocks by initiating a transition to the L2/L3 Ready state.

A Switch must transition its Upstream Link to the L2/L3 Ready state after all of its Downstream Ports have entered the L2/L3 Ready state.

The Links attached to the originator of the PME_Turn_Off Message are the last to assume the L2/L3 Ready state. This state transition serves as an indication to the power delivery manager[84] that all Links within that portion of the Hierarchy have successfully retired all in flight PME Messages to the point of PME_Turn_Off Message origin and have performed any necessary local conditioning in preparation for power removal.

In order to avoid deadlock in the case where one or more devices do not respond with a PME_TO_Ack Message and then put their Links into the L2/L3 Ready state, the power manager must implement a timeout after waiting for a certain amount of time, after which it proceeds as if the Message had been received and all Links put into the L2/L3 Ready state. The recommended limit for this timer is in the range of 1 ms to 10 ms.

The power delivery manager must wait a minimum of 100 ns after observing all Links corresponding to the point of origin of the PME_Turn_Off Message enter L2/L3 Ready before removing the components' reference clock and main power. This requirement does not apply in the case where the above mentioned timer triggers.

IMPLEMENTATION NOTE

PME_TO_Ack Message Proxy by Switches

One of the PME_Turn_Off/PME_TO_Ack handshake's key roles is to ensure that all in flight PME Messages are flushed from the PCI Express fabric prior to sleep state power removal. This is guaranteed to occur because PME Messages and the PME_TO_Ack Messages both use the posted request queue within VC0 and so all previously injected PME Messages will be made visible to the system before the PME_TO_Ack is received at the Root Complex. Once all Downstream Ports of the Root Complex receive a PME_TO_Ack Message the Root Complex can then signal the power manager that it is safe to remove power without loss of any PME Messages.

Switches create points of hierarchical expansion and, therefore, must wait for all of their connected Downstream Ports to receive a PME_TO_Ack Message before they can send a PME_TO_Ack Message Upstream on behalf of the sub-hierarchy that it has created Downstream. This can be accomplished very simply using common score boarding techniques. For example, once a PME_Turn_Off broadcast Message has been broadcast Downstream of the Switch, the Switch simply checks off each Downstream Port having received a PME_TO_Ack. Once the last of its active Downstream Ports receives a PME_TO_Ack, the Switch will then send a single PME_TO_Ack Message Upstream as a proxy on behalf of the entire sub-hierarchy Downstream of it. Note that once a Downstream Port receives a PME_TO_Ack Message and the Switch has scored its arrival, the Port is then free to drop the packet from its internal queues and free up the corresponding posted request queue FC credits.

84. Power delivery control within this context relates to control over the entire Link hierarchy, or over a subset of Links ranging down to a single Link and associated Endpoint for sub hierarchies supporting independently managed power and clock distribution.

5.3.3.3 PM_PME Messages

PM_PME Messages are posted Transaction Layer Packets (TLPs) that inform the power management software which agent within the Hierarchy requests a PM state change. PM_PME Messages, like all other Power Management system Messages, must use the general purpose Traffic Class, TC0.

PM_PME Messages are always routed in the direction of the Root Complex. To send a PM_PME Message on its Upstream Link, a device must transition the Link to the L0 state (if the Link was not in that state already). Unless otherwise noted, the device will keep the Link in the L0 state following the transmission of a PM_PME Message.

5.3.3.3.1 PM_PME "Backpressure" Deadlock Avoidance

A Root Complex is typically implemented with local buffering to store temporarily a finite number of PM_PME Messages that could potentially be simultaneously propagating through the Hierarchy. Given a limited number of PM_PME Messages that can be stored within the Root Complex, there can be backpressure applied to the Upstream directed posted queue in the event that the capacity of this temporary PM_PME Message buffer is exceeded.

Deadlock can occur according to the following example scenario:

1. Incoming PM_PME Messages fill the Root Complex's temporary storage to its capacity while there are additional PM_PME Messages still in the Hierarchy making their way Upstream.
2. The Root Complex, on behalf of system software, issues a Configuration Read Request targeting one of the PME requester's PMCSR (e.g., reading its PME_Status bit).
3. The corresponding split completion Packet is required, as per producer/consumer ordering rules, to push all previously posted PM_PME Messages ahead of it, which in this case are PM_PME Messages that have no place to go.
4. The PME service routine cannot make progress; the PM_PME Message storage situation does not improve.
5. Deadlock occurs.

Precluding potential deadlocks requires the Root Complex to always enable forward progress under these circumstances. This must be done by accepting any PM_PME Messages that posted queue flow control credits allow for, and discarding any PM_PME Messages that create an overflow condition. This required behavior ensures that no deadlock will occur in these cases; however, PM_PME Messages will be discarded and hence lost in the process.

To ensure that no PM_PME Messages are lost permanently, all agents that are capable of generating PM_PME must implement a PME Service Timeout mechanism to ensure that their PME requests are serviced in a reasonable amount of time.

If after 100 ms (+50%/-5%), the PME_Status bit of a requesting agent has not yet been cleared, the PME Service Timeout mechanism expires triggering the PME requesting agent to re-send the temporarily lost PM_PME Message. If at this time the Link is in a non-communicating state, then, prior to re-sending the PM_PME Message, the agent must reactivate the Link as defined in Section 5.3.3.2 .

5.3.3.4 PME Rules

- All device Functions must implement the PCI-PM Power Management Capabilities (PMC) register and the PMCSR in accordance with the PCI-PM specification. These registers reside in the PCI-PM compliant PCI Capability List format.

- PME capable Functions must implement the PME_Status bit, and underlying functional behavior, in their PMCSR.
- When a Function initiates Link wakeup, or issues a PM_PME Message, it must set its PME_Status bit.

- Switches must route a PM_PME received on any Downstream Port to their Upstream Port
- On receiving a PME_Turn_Off Message, the device must block the transmission of PM_PME Messages and transmit a PME_TO_Ack Message Upstream. The component is permitted to send a PM_PME Message after the Link is returned to an L0 state through LDn.
- Before a Link or a portion of a Hierarchy is transferred into a non-communicating state (i.e., a state from which it cannot issue a PM_PME Message), a PME_Turn_Off Message must be broadcast Downstream.

5.3.3.5 PM_PME Delivery State Machine

The following diagram conceptually outlines the PM_PME delivery control state machine. This state machine determines the ability of a Link to service PME events by issuing PM_PME immediately vs. requiring Link wakeup.

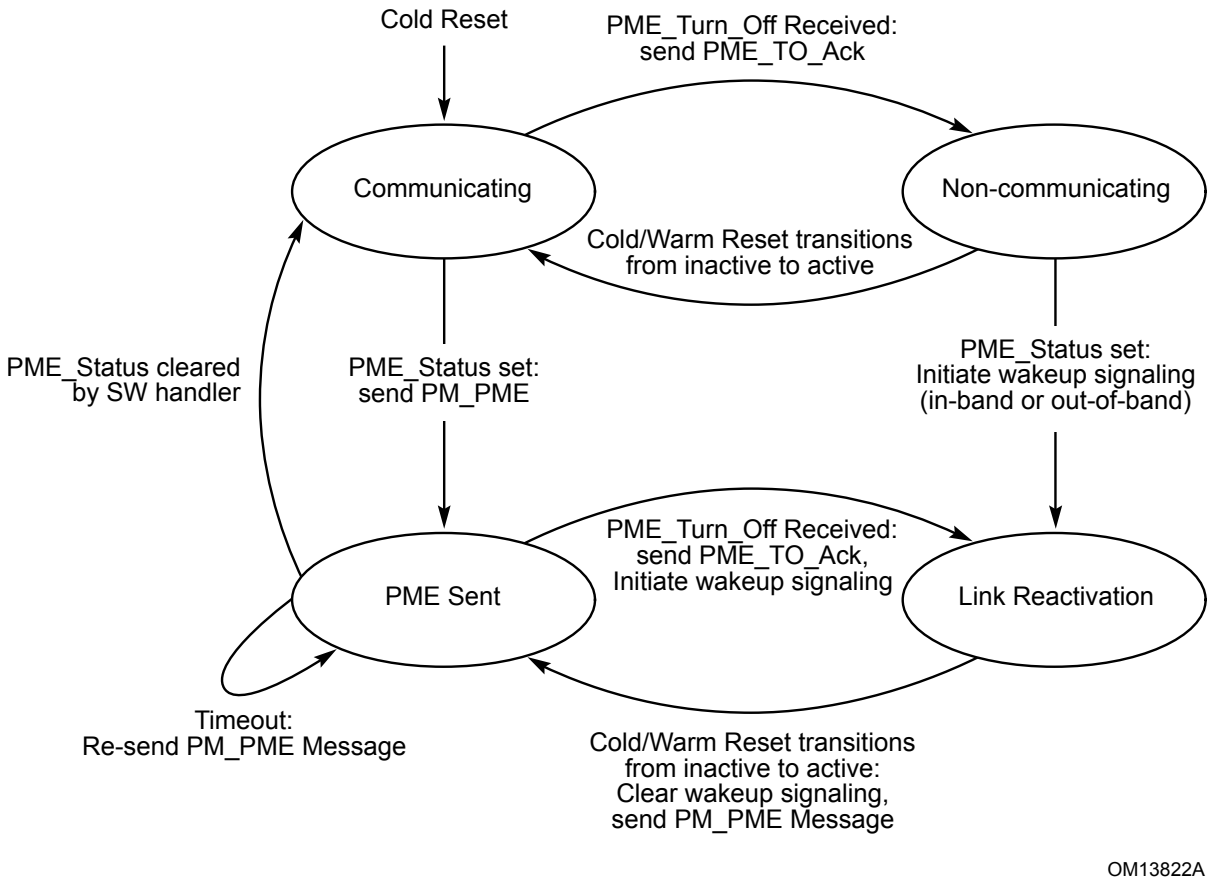

Figure 5-5 A Conceptual PME Control State Machine

Communicating State:

At initial power-up and associated reset, the Upstream Link enters the Communicating state

- If PME_Status is asserted (assuming PME delivery is enabled), a PM_PME Message will be issued Upstream, terminating at the root of the Hierarchy. The next state is the PME Sent state
- If a PME_Turn_Off Message is received, the Link enters the Non-communicating state following its acknowledgment of the Message and subsequent entry into the L2/L3 Ready state.

Non-communicating State:

- Following the restoration of power and clock, and the associated reset, the next state is the Communicating state.
- If PME_Status is asserted, the Link will transition to the Link Reactivation state, and activate the wakeup mechanism.

PME Sent State

- If PME_Status is cleared, the Function becomes PME Capable again. Next state is the Communicating state.
- If the PME_Status bit is not Clear by the time the PME service timeout expires, a PM_PME Message is re-sent Upstream. Refer to Section 5.3.3.3.1 for an explanation of the timeout mechanism.
- If a PME Message has been issued but the PME_Status has not been cleared by software when the Link is about to be transitioned into a messaging incapable state (a PME_Turn_Off Message is received), the Link transitions into Link Reactivation state after sending a PME_TO_Ack Message. The device also activates the wakeup mechanism.

Link Reactivation State

- Following the restoration of power and clock, and the associated reset, the Link resumes a transaction-capable state. The device clears the wakeup signaling, if necessary, and issues a PM_PME Upstream and transitions into the PME Sent state.

5.4 Native PCI Express Power Management Mechanisms

The following sections define power management features that require new software. While the presence of these features in new PCI Express designs will not break legacy software compatibility, taking the full advantage of them requires new code to manage them.

These features are enumerated and configured using PCI Express native configuration mechanisms as described in Chapter 7 of this specification. Refer to Chapter 7 for specific register locations, bit assignments, and access mechanisms associated with these PCI Express-PM features.

5.4.1 Active State Power Management (ASPM)

All Ports not associated with an Internal Root Complex Link or system Egress Port are required to support the minimum requirements defined herein for Active State Link PM. This feature must be treated as being orthogonal to the PCI-PM software compatible features from a minimum requirements perspective. For example, the Root Complex is exempt from the PCI-PM software compatible features requirements; however, it must implement the minimum requirements of ASPM.

Components in the D0 state (i.e., fully active state) normally keep their Upstream Link in the active L0 state, as defined in Section 5.3.2 . ASPM defines a protocol for components in the D0 state to reduce Link power by placing their Links into a low power state and instructing the other end of the Link to do likewise. This capability allows hardware-autonomous,

dynamic Link power reduction beyond what is achievable by software-only controlled (i.e., PCI-PM software driven) power management.

Two low power “standby” Link states are defined for ASPM. The L0s low power Link state is optimized for short entry and exit latencies, while providing substantial power savings. If the L0s state is enabled in a device, it is recommended that the device bring its Transmit Link into the L0s state whenever that Link is not in use (refer to Section 5.4.1.1.1 for details relating to the L0s invocation policy). Component support of the L0s Link state from within the D0 device state is optional unless the applicable form factor specification for the Link explicitly requires it.

The L1 Link state is optimized for maximum power savings at a cost of longer entry and exit latencies. L1 reduces Link power beyond the L0s state for cases where very low power is required and longer transition times are acceptable. ASPM support for the L1 Link state is optional unless specifically required by a particular form factor.

Optional L1 PM Substates L1.1 and L1.2 are defined. These substates can further reduce Link power for cases where very low idle power is required, and longer transition times are acceptable.

Each component must report its level of support for ASPM in the ASPM Support field. As applicable, each component shall also report its L0s and L1 exit latency (the time that it requires to transition from the L0s or L1 state to the L0 state). Endpoint Functions must also report the worst-case latency that they can withstand before risking, for example, internal FIFO overruns due to the transition latency from L0s or L1 to the L0 state. Power management software can use the provided information to then enable the appropriate level of ASPM.

The L0s exit latency may differ significantly if the reference clock for opposing sides of a given Link is provided from the same source, or delivered to each component from a different source. PCI Express-PM software informs each device of its clock configuration via the Common Clock Configuration bit in its Capability structure's Link Control register. This bit serves as the determining factor in the L0s exit latency value reported by the device. ASPM may be enabled or disabled by default depending on implementation specific criteria and/or the requirements of the associated form factor specification(s). Software can enable or disable ASPM using a process described in Section 5.4.1.3.1 .

Power management software enables or disables ASPM in each Port of a component by programming the ASPM Control field. Note that new BIOS code can effectively enable or disable ASPM functionality when running with a legacy operating system, but a PCI Express-aware operating system might choose to override ASPM settings configured by the BIOS.

IMPLEMENTATION NOTE

Isochronous Traffic and ASPM

Isochronous traffic requires bounded service latency. ASPM may add latency to isochronous transactions beyond expected limits. A possible solution would be to disable ASPM for devices that are configured with an Isochronous Virtual Channel.

For ARI Devices, ASPM Control is determined solely by the setting in Function 0, regardless of Function 0's D-state. The ASPM Control settings in other Functions are ignored by the component.

An Upstream Port of a non-ARI Multi-Function Device may be programmed with different values in their respective ASPM Control fields of each Function. The policy for such a component will be dictated by the most active common denominator among all D0 Functions according to the following rules:

- Functions in a non-D0 state (D1 and deeper) are ignored in determining the ASPM policy
- If any of the Functions in the D0 state has its ASPM disabled (ASPM Control field = 00b) or if at least one of the Functions in the D0 state is enabled for L0s only (ASPM Control field = 01b) and at least one other Function in the D0 state is enabled for L1 only (ASPM Control field = 10b), then ASPM is disabled for the entire component

- Else, if at least one of the Functions in the D0 state is enabled for L0s only (ASPM Control field = 01b), then ASPM is enabled for L0s only
- Else, if at least one of the Functions in the D0 state is enabled for L1 only (ASPM Control field = 10b), then ASPM is enabled for L1 only
- Else, ASPM is enabled for both L0s and L1 states

Note that the components must be capable of changing their behavior during runtime as device Functions enter and exit low power device states. For example, if one Function within a Multi-Function Device is programmed to disable ASPM, then ASPM must be disabled for that device while that Function is in the D0 state. Once the Function transitions to a non-D0 state, ASPM can be enabled if all other Functions are enabled for ASPM.

5.4.1.1 L0s ASPM State

Device support of the L0s low power Link state is optional unless the applicable form factor specification for the Link explicitly requires it.

IMPLEMENTATION NOTE

Potential Issues With Legacy Software When L0s is Not Supported

In earlier versions of this specification, device support of L0s was mandatory, and software could legitimately assume that all devices support L0s. Newer hardware components that do not support L0s may encounter issues with such “legacy software”. Such software might not even check the ASPM Support field in the Link Capabilities register, might not recognize the subsequently defined values (00b and 10b) for the ASPM Support field, or might not follow the policy of enabling L0s only if components on both sides of the Link each support L0s.

Legacy software (either operating system or firmware) that encounters the previously reserved value 00b (No ASPM Support), will most likely refrain from enabling L1, which is intended behavior. Legacy software will also most likely refrain from enabling L0s for that component's Transmitter (also intended behavior), but it is unclear if such software will also refrain from enabling L0s for the component on the other side of the Link. If software enables L0s on one side when the component on the other side does not indicate that it supports L0s, the result is undefined. Situations where the resulting behavior is unacceptable may need to be handled by updating the legacy software, resorting to “blacklists” or similar mechanisms directing the legacy software not to enable L0s, or simply not supporting the problematic system configurations.

On some platforms, firmware controls ASPM, and the operating system may either preserve or override the ASPM settings established by firmware. This will be influenced by whether the operating system supports controlling ASPM, and in some cases by whether the firmware permits the operating system to take control of ASPM. Also, ASPM control with hot-plug operations may be influenced by whether native PCI Express hot-plug versus ACPI hot-plug is used. Addressing any legacy software issues with L0s may require updating the firmware, the operating system, or both.

When a component does not advertise that it supports L0s, as indicated by its ASPM Support field value being 00b or 10b, it is recommended that the component's L0s Exit Latency field return a value of 111b, indicating the maximum latency range. Advertising this maximum latency range may help discourage legacy software from enabling L0s if it otherwise would do so, and thus may help avoid problems caused by legacy software mistakenly enabling L0s on this component or the component on the other side of the Link.

Transaction Layer and Link Layer timers are not affected by a transition to the L0s state (i.e., they must follow the rules as defined in their respective chapters).

IMPLEMENTATION NOTE
Minimizing L0s Exit Latency

L0s exit latency depends mainly on the ability of the Receiver to quickly acquire bit and Symbol synchronization. Different approaches exist for high-frequency clocking solutions which may differ significantly in their L0s exit latency, and therefore in the efficiency of ASPM. To achieve maximum power savings efficiency with ASPM, L0s exit latency should be kept low by proper selection of the clocking solution.

5.4.1.1.1 Entry into the L0s State

Entry into the L0s state is managed separately for each direction of the Link. It is the responsibility of each device at either end of the Link to initiate an entry into the L0s state on its transmitting Lanes. Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined.

A Port that is disabled for the L0s state must not transition its transmitting Lanes to the L0s state. However, if the Port advertises that it supports L0s, Port must be able to tolerate having its Receiver Port Lanes enter L0s, (as a result of the device at the other end bringing its transmitting Lanes into L0s state), and then later returning to the L0 state.

L0s Invocation Policy

Ports that are enabled for L0s entry generally should transition their Transmit Lanes to the L0s state if the defined idle conditions (below) are met for a period of time, recommended not to exceed 7 µs. Within this time period, the policy used by the Port to determine when to enter L0s is implementation specific. It is never mandatory for a Transmitter to enter L0s.

Definition of Idle

The definition of an “idle” Upstream Port varies with device Function category. An Upstream Port of a Multi-Function Device is considered idle only when all of its Functions are idle.

A non-Switch Port is determined to be idle if the following conditions are met:

- No TLP is pending to transmit over the Link, or no FC credits are available to transmit any TLPs
- No DLLPs are pending for transmission

A Switch Upstream Port Function is determined to be idle if the following conditions are met:

- None of the Switch's Downstream Port Receive Lanes are in the L0, Recovery, or Configuration state
- No pending TLPs to transmit, or no FC credits are available to transmit anything
- No DLLPs are pending for transmission

A Switch's Downstream Port is determined to be idle if the following conditions are met:

- The Switch's Upstream Port's Receive Lanes are not in the L0, Recovery, or Configuration state
- No pending TLPs to transmit on this Link, or no FC credits are available
- No DLLPs are pending for transmission

Refer to Section 4.2 for details on L0s entry by the Physical Layer.

5.4.1.1.2 Exit from the L0s State

A component with its Transmitter in L0s must initiate L0s exit when it has a TLP or DLLP to transmit across the Link. Note that a transition from the L0s Link state does not depend on the status (or availability) of FC credits. The Link must be able to reach the L0 state, and to exchange FC credits across the Link. For example, if all credits of some type were consumed when the Link entered L0s, then any component on either side of the Link must still be able to transition the Link to the L0 state when new credits need to be sent across the Link. Note that it may be appropriate for a component to anticipate the end of the idle condition and initiate L0s transmit exit; for example, when a NP request is received.

Downstream Initiated Exit

The Upstream Port of a component is permitted to initiate an exit from the L0s low-power state on its Transmit Link, (Upstream Port Transmit Lanes in the case of a Downstream Switch), if it needs to communicate through the Link. The component initiates a transition to the L0 state on Lanes in the Upstream direction as described in Section 4.2 .

If the Upstream component is a Switch (i.e., it is not the Root Complex), then it must initiate a transition on its Upstream Port Transmit Lanes (if the Upstream Port's Transmit Lanes are in a low-power state) as soon as it detects an exit from L0s on any of its Downstream Ports.

Upstream Initiated Exit

A Downstream Port is permitted to initiate an exit from L0s low power state on any of its Transmit Links if it needs to communicate through the Link. The component initiates a transition to the L0 state on Lanes in the Downstream direction as described in Chapter 4 .

If the Downstream component contains a Switch, it must initiate a transition on all of its Downstream Port Transmit Lanes that are in L0s at that time as soon as it detects an exit from L0s on its Upstream Port. Links that are already in the L0 state are not affected by this transition. Links whose Downstream component is in a low-power state (i.e., D1- $D3_{Hot}$ states) are also not affected by the exit transitions.

For example, consider a Switch with an Upstream Port in L0s and a Downstream device in a D1 state. A configuration request packet travels Downstream to the Switch, intending ultimately to reprogram the Downstream device from D1 to D0. The Switch's Upstream Port Link must transition to the L0 state to allow the packet to reach the Switch. The Downstream Link connecting to the device in D1 state will not transition to the L0 state yet; it will remain in the L1 state. The captured packet is checked and routed to the Downstream Port that shares a Link with the Downstream device that is in D1. As described in Section 4.2 , the Switch now transitions the Downstream Link to the L0 state. Note that the transition to the L0 state was triggered by the packet being routed to that particular Downstream L1 Link, and not by the transition of the Upstream Port's Link to the L0 state. If the packet's destination was targeting a different Downstream Link, then that particular Downstream Link would have remained in the L1 state.

5.4.1.2 L1 ASPM State

A component may optionally support the ASPM L1 state; a state that provides greater power savings at the expense of longer exit latency. L1 exit latency is visible to software, and reported via the L1 Exit Latency field.

IMPLEMENTATION NOTE
Potential Issues With Legacy Software When Only L1 is Supported

In earlier versions of this specification, device support of L0s was mandatory, and there was no architected ASPM Support field value to indicate L1 support without L0s support. Newer hardware components that support only L1 may encounter issues with “legacy software”, i.e., software that does not recognize the subsequently defined value for the ASPM Support field.

Legacy software that encounters the previously reserved value 10b (L1 Support), may refrain from enabling both L0s and L1, which unfortunately avoids using L1 with new components that support only L1. While this may result in additional power being consumed, it should not cause any functional misbehavior. However, the same issues with respect to legacy software enabling L0s exist for this 10b case as are described in the Implementation Note “Potential Issues With Legacy Software When L0s is Not Supported” in Section 5.4.1.1 .

When supported, L1 entry is disabled by default in the ASPM Control field. Software must enable ASPM L1 on the Downstream component only if it is supported by both components on a Link. Software must sequence the enabling and disabling of ASPM L1 such that the Upstream component is enabled before the Downstream component and disabled after the Downstream component.

5.4.1.2.1 ASPM Entry into the L1 State

An Upstream Port on a component enabled for L1 ASPM entry may initiate entry into the L1 Link state.

See Section 5.5.1 for details on transitions into either the L1.1 or L1.2 substates.

IMPLEMENTATION NOTE
Initiating L1

This specification does not dictate when a component with an Upstream Port must initiate a transition to the L1 state. The interoperable mechanisms for transitioning into and out of L1 are defined within this specification; however, the specific ASPM policy governing when to transition into L1 is left to the implementer.

One possible approach would be for the Downstream device to initiate a transition to the L1 state once the device has both its Receiver and Transmitter in the L0s state (RxL0s and TxL0s) for a set amount of time. Another approach would be for the Downstream device to initiate a transition to the L1 state once the Link has been idle in L0 for a set amount of time. This is particularly useful if L0s entry is not enabled. Still another approach would be for the Downstream device to initiate a transition to the L1 state if it has completed its assigned tasks. Note that a component's L1 invocation policy is in no way limited by these few examples.

Three power management Messages provide support for the ASPM L1 state:

- PM_Active_State_Request_L1 (DLLP)
- PM_Request_Ack (DLLP)
- PM_Active_State_Nak (TLP)

Downstream components enabled for ASPM L1 entry negotiate for L1 entry with the Upstream component on the Link.

A Downstream Port must accept a request to enter L1 if all of the following conditions are true:

- The Port supports ASPM L1 entry, and ASPM L1 entry is enabled.[85]
- No TLP is scheduled for transmission
- No Ack or Nak DLLP is scheduled for transmission

A Switch Upstream Port may request L1 entry on its Link provided all of the following conditions are true:

- The Upstream Port supports ASPM L1 entry and it is enabled
- All of the Switch's Downstream Port Links are in the L1 state (or deeper)
- No pending TLPs to transmit
- No pending DLLPs to transmit
- The Upstream Port's Receiver is idle for an implementation specific set amount of time

Note that it is legitimate for a Switch to be enabled for the ASPM L1 Link state on any of its Downstream Ports and to be disabled or not even supportive of ASPM L1 on its Upstream Port. In that case, Downstream Ports may enter the L1 Link state, but the Switch will never initiate an ASPM L1 entry transition on its Upstream Port.

ASPM L1 Negotiation Rules (see Figure 5-6 and Figure 5-7):

- The Downstream component must not initiate ASPM L1 entry until it accumulates at least the minimum number of credits required to send the largest possible packet for any FC type for all enabled VCs.
- Upon deciding to enter a low-power Link state, the Downstream component must block movement of all TLPs from the Transaction Layer to the Data Link Layer for transmission (including completion packets). If any TLPs become available from the Transaction Layer for transmission during the L1 negotiation process, the transition to L1 must first be completed and then the Downstream component must initiate a return to L0. Refer to Section 5.2 if the negotiation to L1 is interrupted.
- The Downstream component must wait until it receives a Link Layer acknowledgement for the last TLP it had previously sent (i.e., the retry buffer is empty). The component must retransmit a TLP out of its Data Link Layer Retry buffer if required by the Data Link Layer rules.
- The Downstream component then initiates ASPM negotiation by sending a PM_Active_State_Request_L1 DLLP onto its Transmit Lanes. The Downstream component sends this DLLP repeatedly with no more than eight (when using 8b/10b encoding) or 32 (when using 128b/130b encoding) Symbol times of idle between subsequent transmissions of the PM_Active_State_Request_L1 DLLP. The transmission of other DLLPs and SKP Ordered Sets must occur as required at any time between PM_Active_State_Request_L1 transmissions, and do not contribute to this idle time limit. Transmission of SKP Ordered Sets during L1 entry follows the clock tolerance compensation rules in Section 4.2.7 .
- The Downstream component continues to transmit the PM_Active_State_Request_L1 DLLP as described above until it receives a response from the Upstream device (see below). The Downstream component remains in this loop waiting for a response from the Upstream component.

  During this waiting period, the Downstream component must not initiate any Transaction Layer transfers. It must still accept TLPs and DLLPs from the Upstream component, storing for later transmission any TLP responses required. It continues to respond with DLLPs, including FC update DLLPs, as needed by the Link Layer protocol.

85. Software must enable ASPM L1 for the Downstream component only if it is also enabled for the Upstream component.

If the Downstream component for any reason needs to transmit a TLP on the Link, it must first complete the transition to the low-power Link state. Once in a lower power Link state, the Downstream component must then initiate exit of the low-power Link state to handle the transfer. Refer to Section 5.2 if the negotiation to L1 is interrupted.

- The Upstream component must immediately (while obeying all other rules in this specification) respond to the request with either an acceptance or a rejection of the request.
  If the Upstream component is not able to accept the request, it must immediately (while obeying all other rules in this specification) reject the request.
- Refer to Section 5.2 if the negotiation to L1 is interrupted.

Rules in case of rejection:

- In the case of a rejection, the Upstream component must schedule, as soon as possible, a rejection by sending the PM_Active_State_Nak Message to the Downstream component. Once the PM_Active_State_Nak Message is sent, the Upstream component is permitted to initiate any TLP or DLLP transfers.
- If the request was rejected, it is generally recommended that the Downstream component immediately transition its Transmit Lanes into the L0s state, provided L0s is enabled and that conditions for L0s entry are met.
- Prior to transmitting a PM_Active_State_Request_L1 DLLP associated with a subsequent ASPM L1 negotiation sequence, the Downstream component must either enter and exit L0s on its Transmitter, or it must wait at least 10 µs from the last transmission of the PM_Active_State_Request_L1 DLLP associated with the preceding ASPM L1 negotiation. This 10 µs timer must count only time spent in the LTSSM L0 and L0s states. The timer must hold in the LTSSM Recovery state. If the Link goes down and comes back up, the timer is ignored and the component is permitted to issue new ASPM L1 request after the Link has come back up.

IMPLEMENTATION NOTE

ASPM L1 Accept/Reject Considerations for the Upstream Component

When the Upstream component has responded to the Downstream component's ASPM L1 request with a PM_Request_Ack DLLP to accept the L1 entry request, the ASPM L1 negotiation protocol clearly and unambiguously ends with the Link entering L1. However, if the Upstream component responds with a PM_Active_State_Nak Message to reject the L1 entry request, the termination of the ASPM L1 negotiation protocol is less clear. Therefore, both components need to be designed to unambiguously terminate the protocol exchange. If this is not done, there is the risk that the two components will get out of sync with each other, and the results may be undefined. For example, consider the following case:

- The Downstream component requests ASPM L1 entry by transmitting a sequence of PM_Active_State_Request_L1 DLLPs.
- Due to a temporary condition, the Upstream component responds with a PM_Active_State_Nak Message to reject the L1 request.
- The Downstream component continues to transmit the PM_Active_State_Request_L1 DLLPs for some time before it is able to respond to the PM_Active_State_Nak Message.
- Meanwhile, the temporary condition that previously caused the Upstream component to reject the L1 request is resolved, and the Upstream component erroneously sees the continuing PM_Active_State_Request_L1 DLLPs as a new request to enter L1, and responds by transmitting PM_Request_Ack DLLPs Downstream.

At this point, the result is undefined, because the Downstream component views the L1 request as rejected and finishing, but the Upstream component views the situation as a second L1 request being accepted.

To avoid this situation, the Downstream component needs to provide a mechanism to distinguish between one ASPM L1 request and another. The Downstream component does this by entering L0s or by waiting a minimum of 10 µs from the transmission of the last PM_Active_State_Request_L1 DLLP associated with the first ASPM L1 request before starting transmission of the PM_Active_State_Request_L1 DLLPs associated with the second request (as described above).

If the Upstream component is capable of exhibiting the behavior described above, then it is necessary for the Upstream component to recognize the end of an L1 request sequence by detecting a transition to L0s on its Receiver or a break in the reception of PM_Active_State_Request_L1 DLLPs of 9.5 µs measured while in L0/L0s or more as a separation between ASPM L1 requests by the Downstream component.

If there is a possibility of ambiguity, the Upstream component should reject the L1 request to avoid potentially creating the ambiguous situation outlined above.

Rules in case of acceptance:

- If the Upstream component is ready to accept the request, it must block scheduling of any TLPs from the Transaction Layer.
- The Upstream component then must wait until it receives a Data Link Layer acknowledgement for the last TLP it had previously sent. The Upstream component must retransmit a TLP if required by the Data Link Layer rules.

- Once all TLPs have been acknowledged, the Upstream component sends a PM_Request_Ack DLLP Downstream. The Upstream component sends this DLLP repeatedly with no more than eight (when using 8b/10b encoding) or 32 (when using 128b/130b encoding) Symbol times of idle between subsequent transmissions of the PM_Request_Ack DLLP. The transmission of SKP Ordered Sets must occur as required at any time between PM_Request_Ack transmissions, and do not contribute to this idle time limit. Transmission of SKP Ordered Sets during L1 entry follows the clock tolerance compensation rules in Section 4.2.7 .
- The Upstream component continues to transmit the PM_Request_Ack DLLP as described above until it observes its Receive Lanes enter into the Electrical Idle state. Refer to Chapter 4 for more details on the Physical Layer behavior.
- If the Upstream component needs, for any reason, to transmit a TLP on the Link after it sends a PM_Request_Ack DLLP, it must first complete the transition to the low-power state, and then initiate an exit from the low-power state to handle the transfer once the Link is back to L0. Refer to Section 5.2 if the negotiation to L1 is interrupted.
  - The Upstream component must initiate an exit from L1 in this case even if it does not have the required flow control credit to transmit the TLP(s).
- When the Downstream component detects a PM_Request_Ack DLLP on its Receive Lanes (signaling that the Upstream device acknowledged the transition to L1 request), the Downstream component then ceases sending the PM_Active_State_Request_L1 DLLP, disables DLLP, TLP transmission and brings its Transmit Lanes into the Electrical Idle state.
- When the Upstream component detects an Electrical Idle on its Receive Lanes (signaling that the Downstream component has entered the L1 state), it then ceases to send the PM_Request_Ack DLLP, disables DLLP, TLP transmission and brings the Downstream direction of the Link into the Electrical Idle state.

Notes:

1. The transaction Layer Completion Timeout mechanism is not affected by transition to the L1 state (i.e., it must keep counting).
2. Flow Control Update timers are frozen while the Link is in L1 state to prevent a timer expiration that will unnecessarily transition the Link back to the L0 state.

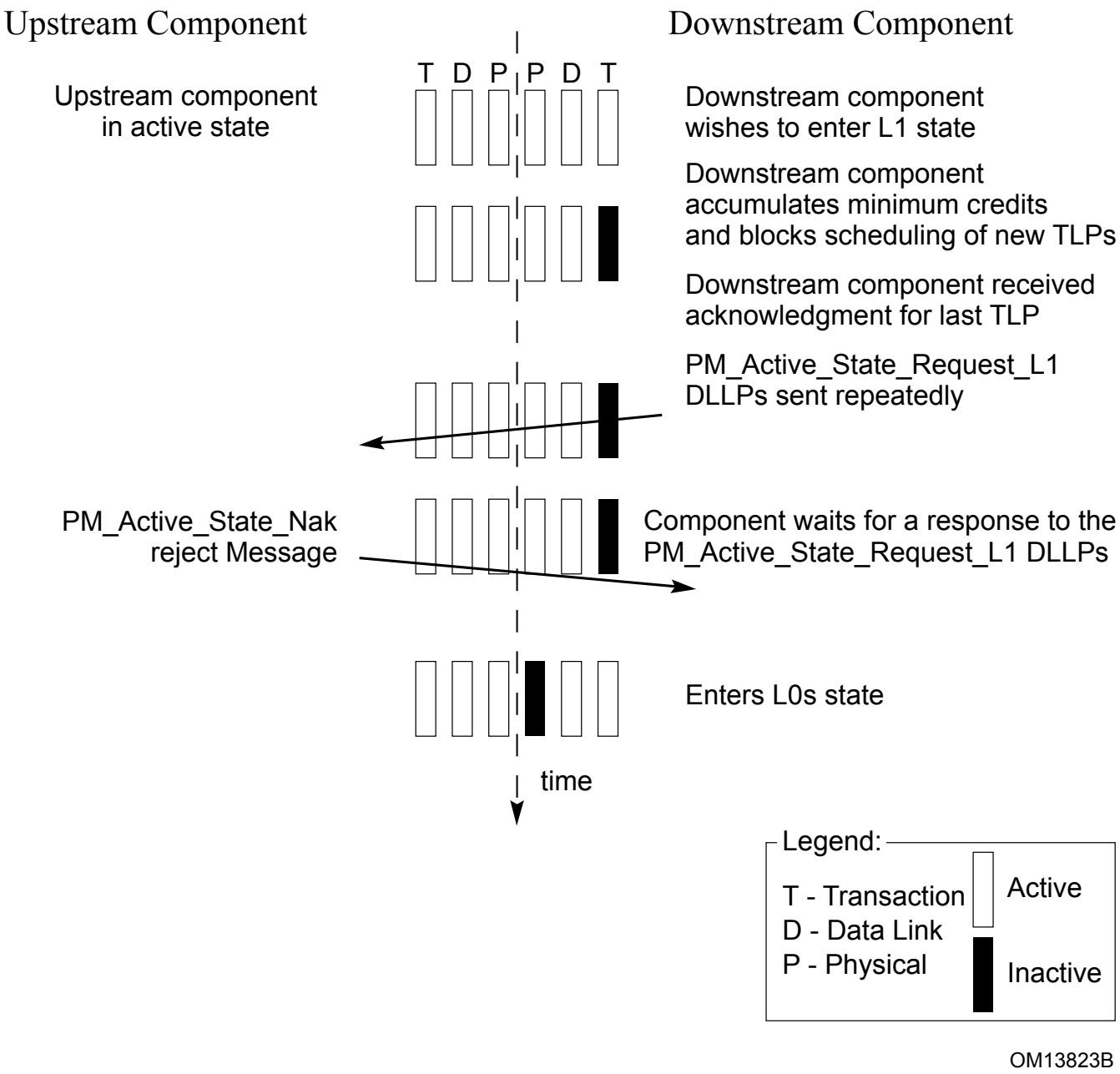

Figure 5-6 L1 Transition Sequence Ending with a Rejection (L0s Enabled)

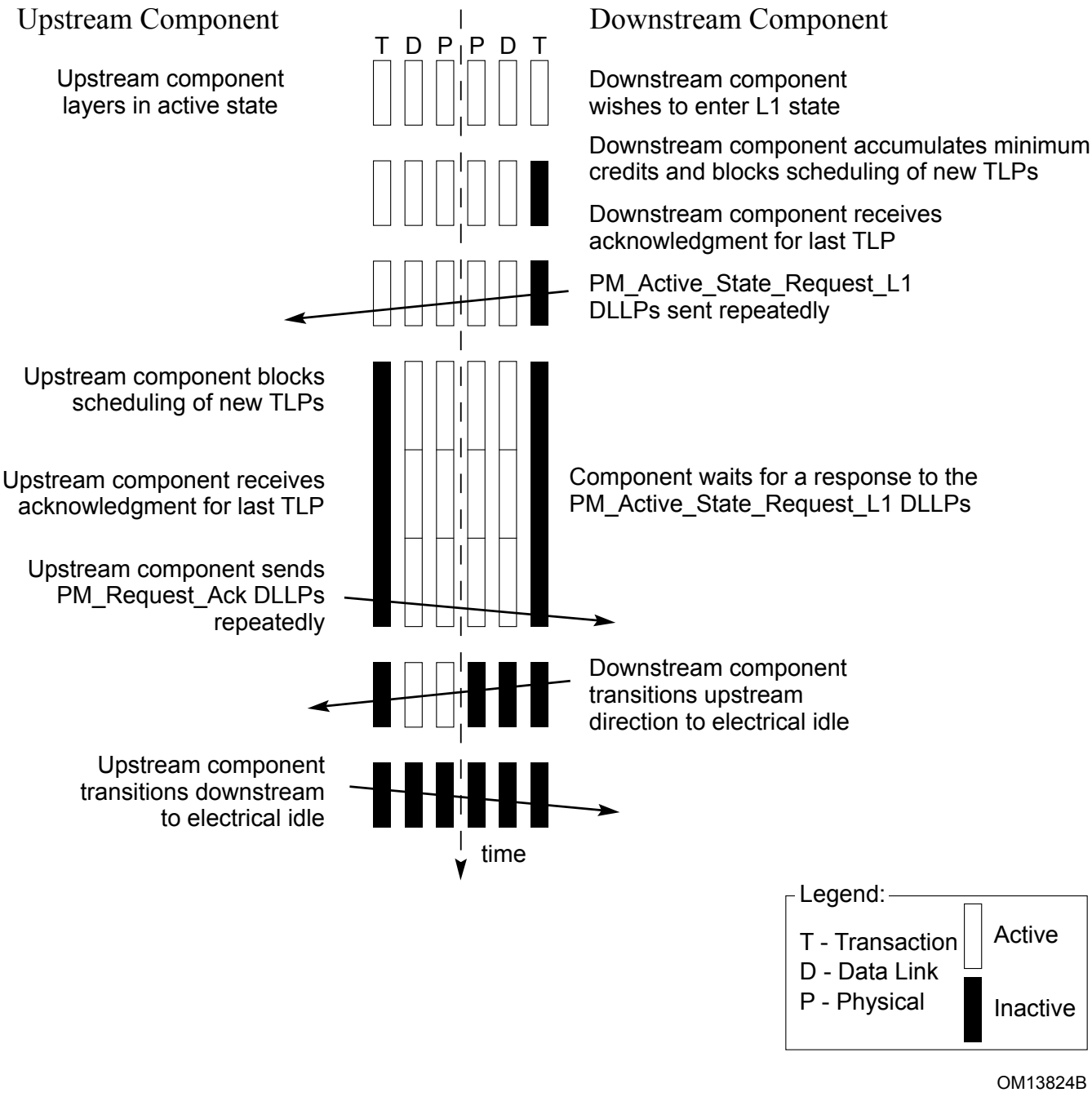

Figure 5-7 L1 Successful Transition Sequence

5.4.1.2.2 Exit from the L1 State

Components on either end of a Link may initiate an exit from the L1 Link state.

See Section 5.5.1 for details on transitions into either the L1.1 or L1.2 substates.

Upon exit from L1, it is recommended that the Downstream component send flow control update DLLPs for all enabled VCs and FC types starting within 1 µs of L1 exit.

Downstream Component Initiated Exit

An Upstream Port must initiate an exit from L1 on its Transmit Lanes if it needs to communicate through the Link. The component initiates a transition to the L0 state as described in Chapter 4 . The Upstream component must respond by initiating a similar transition of its Transmit Lanes.

If the Upstream component is a Switch Downstream Port, (i.e., it is not a Root Complex Root Port), the Switch must initiate an L1 exit transition on its Upstream Port's Transmit Lanes, (if the Upstream Port's Link is in the L1 state), as soon as it detects the L1 exit activity on any of its Downstream Port Links. Since L1 exit latencies are relatively long, a Switch must not wait until its Downstream Port Link has fully exited to L0 before initiating an L1 exit transition on its Upstream Port Link. Waiting until the Downstream Link has completed the L0 transition will cause a Message traveling through several Switches to experience accumulating latency as it traverses each Switch.

A Switch is required to initiate an L1 exit transition on its Upstream Port Link after no more than 1 µs from the beginning of an L1 exit transition on any of its Downstream Port Links. Refer to Section 4.2 for details of the Physical Layer signaling during L1 exit.

Consider the example in Figure 5-8 . The numbers attached to each Port represent the corresponding Port's reported Transmit Lanes L1 exit latency in units of microseconds.

Links 1, 2, and 3 are all in the L1 state, and Endpoint C initiates a transition to the L0 state at time T. Since Switch B takes 32 µs to exit L1 on its Ports, Link 3 will transition to the L0 state at T+32 (longest time considering T+8 for the Endpoint C, and T+32 for Switch B).

Switch B is required to initiate a transition from the L1 state on its Upstream Port Link (Link 2) after no more than 1 µs from the beginning of the transition from the L1 state on Link 3. Therefore, transition to the L0 state will begin on Link 2 at T+1. Similarly, Link 1 will start its transition to the L0 state at time T+2.

Following along as above, Link 2 will complete its transition to the L0 state at time T+33 (since Switch B takes longer to transition and it started at time T+1). Link 1 will complete its transition to the L0 state at time T+34 (since the Root Complex takes 32 µs to transition and it started at time T+2).

Therefore, among Links 1, 2, and 3, the Link to complete the transition to the L0 state last is Link 1 with a 34 µs delay. This is the delay experienced by the packet that initiated the transition in Endpoint C.

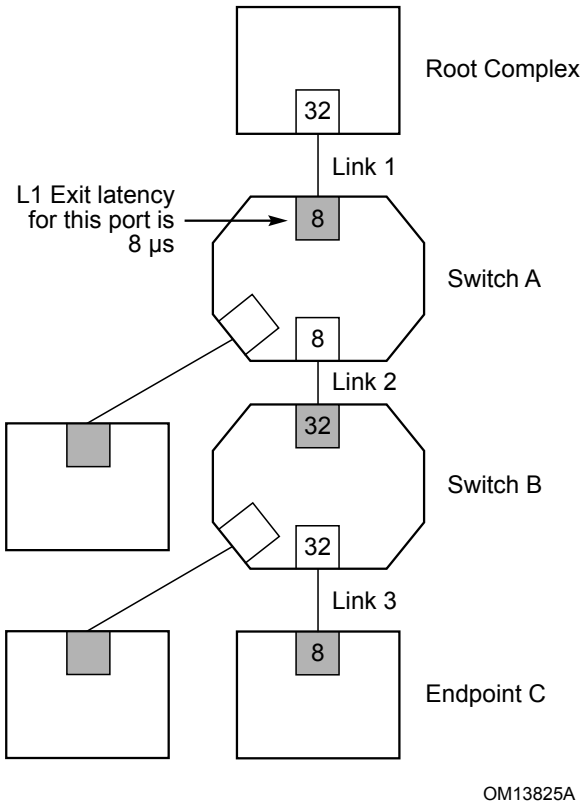

Figure 5-8 Example of L1 Exit Latency Computation

Switches are not required to initiate an L1 exit transition on any other of their Downstream Port Links.

Upstream Component Initiated Exit

A Root Complex, or a Switch must initiate an exit from L1 on any of its Root Ports, or Downstream Port Links if it needs to communicate through that Link. The Switch or Root Complex must be capable of initiating L1 exit even if it does not have the flow control credits needed to transmit a given TLP. The component initiates a transition to the L0 state as described in Chapter 4 . The Downstream component must respond by initiating a similar transition on its Transmit Lanes.

If the Downstream component contains a Switch, it must initiate a transition on all of its Downstream Links (assuming the Downstream Link is in an ASPM L1 state) as soon as it detects an exit from L1 state on its Upstream Port Link. Since L1 exit latencies are relatively long, a Switch must not wait until its Upstream Port Link has fully exited to L0 before initiating an L1 exit transition on its Downstream Port Links. If that were the case, a Message traveling through multiple Switches would experience accumulating latency as it traverses each Switch.

A Switch is required to initiate a transition from L1 state on all of its Downstream Port Links that are currently in L1 after no more than 1 µs from the beginning of a transition from L1 state on its Upstream Port. Refer to Section 4.2 for details of the Physical Layer signaling during L1 exit. Downstream Port Links that are already in the L0 state do not participate in the exit transition. Downstream Port Links whose Downstream component is in a low power D-state (D1-$D3_{Hot}$) are also not affected by the L1 exit transitions (i.e., such Links must not be transitioned to the L0 state).

5.4.1.3 ASPM Configuration

All Functions must implement the following configuration bits in support of ASPM. Refer to Chapter 7 for configuration register assignment and access mechanisms.

Each component reports its level of support for ASPM in the ASPM Support field below.

Table 5-3 Encoding of the ASPM Support Field

Field	Description
ASPM Support	**00b** No ASPM support **01b** L0s supported **10b** L1 supported **11b** L0s and L1 supported

Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined.

Each component reports the source of its reference clock in its Slot Clock Configuration bit located in its Capability structure's Link Status register.

Table 5-4 Description of the Slot Clock Configuration Bit

Bit	Description
Slot Clock Configuration	This bit, when Set, indicates that the component uses the same physical reference clock that the platform provides on the connector. This bit, when Clear, indicates the component uses an independent clock irrespective of the presence of a reference on the connector. For Root and Switch Downstream Ports, this bit, when Set, indicates that the Downstream Port is using the same reference clock as the Downstream component or the slot. For Switch and Bridge Upstream Ports, this bit when Set, indicates that the Upstream Port is using the same reference clock that the platform provides. Otherwise it is Clear.

Each component must support the Common Clock Configuration bit in their Capability structure's Link Control register. Software writes to this register bit to indicate to the device whether it is sharing the same clock source as the device on the other end of the Link.

Table 5-5 Description of the Common Clock Configuration Bit

Bit	Description
Common Clock Configuration	This bit, when Set, indicates that this component and the component at the opposite end of the Link are operating with a common clock source. This bit, when Clear, indicates that this component and the component at the opposite end of the Link are operating with separate reference clock sources. Default value of this bit is 0b.

Bit	Description
	Components utilize this common clock configuration information to report the correct L0s and L1 Exit Latencies.

Each Port reports the L0s and L1 exit latency (the time that they require to transition their Receive Lanes from the L0s or L1 state to the L0 state) in the L0s Exit Latency and the L1 Exit Latency configuration fields, respectively. If a Port does not support L0s or ASPM L1, the value of the respective exit latency field is undefined.

Table 5-6 Encoding of the L0s Exit Latency Field

Field	Description
L0s Exit Latency	**000b** Less than 64 ns **001b** 64 ns to less than 128 ns **010b** 128 ns to less than 256 ns **011b** 256 ns to less than 512 ns **100b** 512 ns to less than 1 µs **101b** 1 µs to less than 2 µs **110b** 2 µs to 4 µs **111b** More than 4 µs

Table 5-7 Encoding of the L1 Exit Latency Field

Field	Description
L1 Exit Latency	**000b** Less than 1 µs **001b** 1 µs to less than 2 µs **010b** 2 µs to less than 4 µs **011b** 4 µs to less than 8 µs **100b** 8 µs to less than 16 µs **101b** 16 µs to less than 32 µs **110b** 32 µs to 64 µs **111b** More than 64 µs

Endpoints also report the additional latency that they can absorb due to the transition from L0s state or L1 state to the L0 state. This is reported in the Endpoint L0s Acceptable Latency and Endpoint L1 Acceptable Latency fields, respectively.

Power management software, using the latency information reported by all components in the Hierarchy, can enable the appropriate level of ASPM by comparing exit latency for each given path from Root to Endpoint against the acceptable latency that each corresponding Endpoint can withstand.

Table 5-8 Encoding of the Endpoint L0s Acceptable Latency Field

Field	Description
Endpoint L0s Acceptable Latency	**000b** Maximum of 64 ns **001b** Maximum of 128 ns **010b** Maximum of 256 ns **011b** Maximum of 512 ns **100b** Maximum of 1 µs **101b** Maximum of 2 µs **110b** Maximum of 4 µs **111b** No limit

Table 5-9 Encoding of the Endpoint L1 Acceptable Latency Field

Field	Description
Endpoint L1 Acceptable Latency	**000b** Maximum of 1 µs **001b** Maximum of 2 µs **010b** Maximum of 4 µs **011b** Maximum of 8 µs **100b** Maximum of 16 µs **101b** Maximum of 32 µs **110b** Maximum of 64 µs **111b** No limit

Power management software enables or disables ASPM in each component by programming the ASPM Control field.

Table 5-10 Encoding of the ASPM Control Field

Field	Description
ASPM Control	**00b** Disabled **01b** L0s Entry Enabled **10b** L1 Entry Enabled

Field	Description
	11b L0s and L1 Entry enabled

ASPM Control = 00b

Port's Transmitter must not enter L0s.

Ports connected to the Downstream end of the Link must not issue a PM_Active_State_Request_L1 DLLP on its Upstream Link.

Ports connected to the Upstream end of the Link receiving a L1 request must respond with negative acknowledgement.

ASPM Control = 01b

Port must bring a Link into L0s state if all conditions are met.

Ports connected to the Downstream end of the Link must not issue a PM_Active_State_Request_L1 DLLP on its Upstream Link.

Ports connected to the Upstream end of the Link receiving a L1 request must respond with negative acknowledgement.

ASPM Control = 10b

Port's Transmitter must not enter L0s.

Ports connected to the Downstream end of the Link may issue PM_Active_State_Request_L1 DLLPs.

Ports connected to the Upstream end of the Link must respond with positive acknowledgement to a L1 request and transition into L1 if the conditions for the Root Complex Root Port or Switch Downstream Port in Section 5.4.1.2.1 are met.

ASPM Control = 11b

Port must bring a Link into the L0s state if all conditions are met.

Ports connected to the Downstream end of the Link may issue PM_Active_State_Request_L1 DLLPs.

Ports connected to the Upstream end of the Link must respond with positive acknowledgement to a L1 request and transition into L1 if the conditions for the Root Complex Root Port or Switch Downstream Port in Section 5.4.1.2.1 are met.

5.4.1.3.1 Software Flow for Enabling or Disabling ASPM

Following is an example software algorithm that highlights how to enable or disable ASPM in a component.

- PCI Express components power up with an appropriate value in their Slot Clock Configuration bit. The method by which they initialize this bit is device-specific.
- PCI Express system software scans the Slot Clock Configuration bit in the components on both ends of each Link to determine if both are using the same reference clock source or reference clocks from separate sources. If the Slot Clock Configuration bits in both devices are Set, they are both using the same reference clock source, otherwise they're not.
- PCI Express software updates the Common Clock Configuration bits in the components on both ends of each Link to indicate if those devices share the same reference clock and triggers Link retraining by writing 1b to the Retrain Link bit in the Link Control register of the Upstream component.

- Devices must reflect the appropriate L0s /L1 exit latency in their L0s /L1 Exit Latency fields, per the setting of the Common Clock Configuration bit.
- PCI Express system software then reads and calculates the L0s /L1 exit latency for each Endpoint based on the latencies reported by each Port. Refer to Section 5.4.1.2.2 for an example.
- For each component with one or more Endpoint Functions, PCI Express system software examines the Endpoint L0s /L1 Acceptable Latency, as reported by each Endpoint Function in its Link Capabilities register, and enables or disables L0s /L1 entry (via the ASPM Control field in the Link Control register) accordingly in some or all of the intervening device Ports on that hierarchy.

5.5 L1 PM Substates

L1 PM Substates establish a Link power management regime that creates lower power substates of the L1 Link state (see Figure 5-9), and associated mechanisms for using those substates. The L1 PM Substates are:

- ***L1.0*** substate
  - The L1.0 substate corresponds to the conventional L1 Link state. This substate is entered whenever the Link enters L1. The L1 PM Substate mechanism defines transitions from this substate to and from the L1.1 and L1.2 substates.
  - The Upstream and Downstream Ports must be enabled to detect Electrical Idle exit as required in Section 4.2.6.7.2 .
- ***L1.1*** substate
  - Link common mode voltages are maintained.
  - Uses a bidirectional open-drain clock request (CLKREQ#) signal for entry to and exit from this state.
  - The Upstream and Downstream Ports are not required to be enabled to detect Electrical Idle exit.
- ***L1.2*** substate
  - Link common mode voltages are not required to be maintained.
  - Uses a bidirectional open-drain clock request (CLKREQ#) signal for entry to and exit from this state.
  - The Upstream and Downstream Ports are not required to be enabled to detect Electrical Idle exit.

Ports that support L1 PM Substates must not require a reference clock while in L1 PM Substates other than L1.0.

Ports that support L1 PM Substates and also support SRIS mode are required to support L1 PM Substates while operating in SRIS mode. In such cases the CLKREQ# signal is used by the L1 PM Substates protocol as defined in this section, but has no defined relationship to any local clocks used by either Port on the Link, and the management of such local clocks is implementation-specific.

Ports that support the L1.2 substate for ASPM L1 must support Latency Tolerance Reporting (LTR).

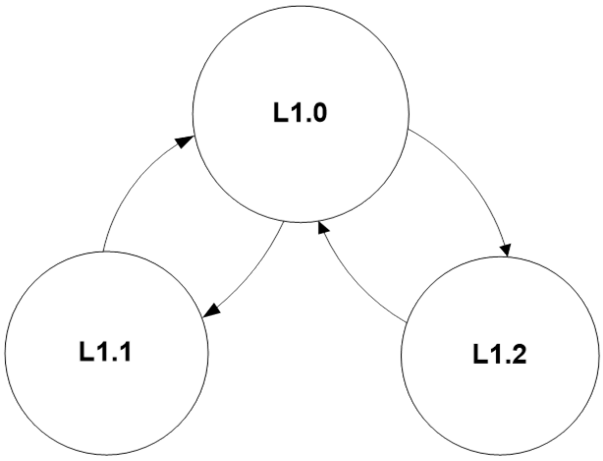

Figure 5-9 State Diagram for L1 PM Substates

- When enabled, the L1 PM Substates mechanism applies the following additional requirements to the CLKREQ# signal:The CLKREQ# signal must be supported as a bi-directional open drain signal by both the Upstream and Downstream Ports of the Link. Each Port must have a unique instance of the signal, and the Upstream and Downstream Port CLKREQ# signals must be connected.
- It is permitted for the Upstream Port to deassert CLKREQ# when the Link is in the PCI-PM L1 or ASPM L1 states, or when the Link is in the L2/L3 Ready pseudo-state; CLKREQ# must be asserted by the Upstream Port when the Link is in any other state.
- All other specifications related to the CLKREQ# signal that are not specifically defined or modified by L1 PM Substates continue to apply.

If these requirements cannot be satisfied in a particular system, then L1 PM Substates must not be enabled.

IMPLEMENTATION NOTE

CLKREQ# Connection Topologies

For an Upstream component the connection topologies for the CLKREQ# signal can vary. A few examples of CLKREQ# connection topologies are described below. For the Downstream component these cases are essentially the same, however from the Upstream component's perspective, there are some key differences that are described below.

Example 1: Single Downstream Port with a single PLL connected to a single Upstream Port (see Figure 5-10).

In this platform configuration the Upstream component (A) has only a single CLKREQ# signal. The Upstream and Downstream Ports' CLKREQ# (A and B) signals are connected to each other. In this case, Upstream component (A), must assert CLKREQ# signal whenever it requires a reference clock.

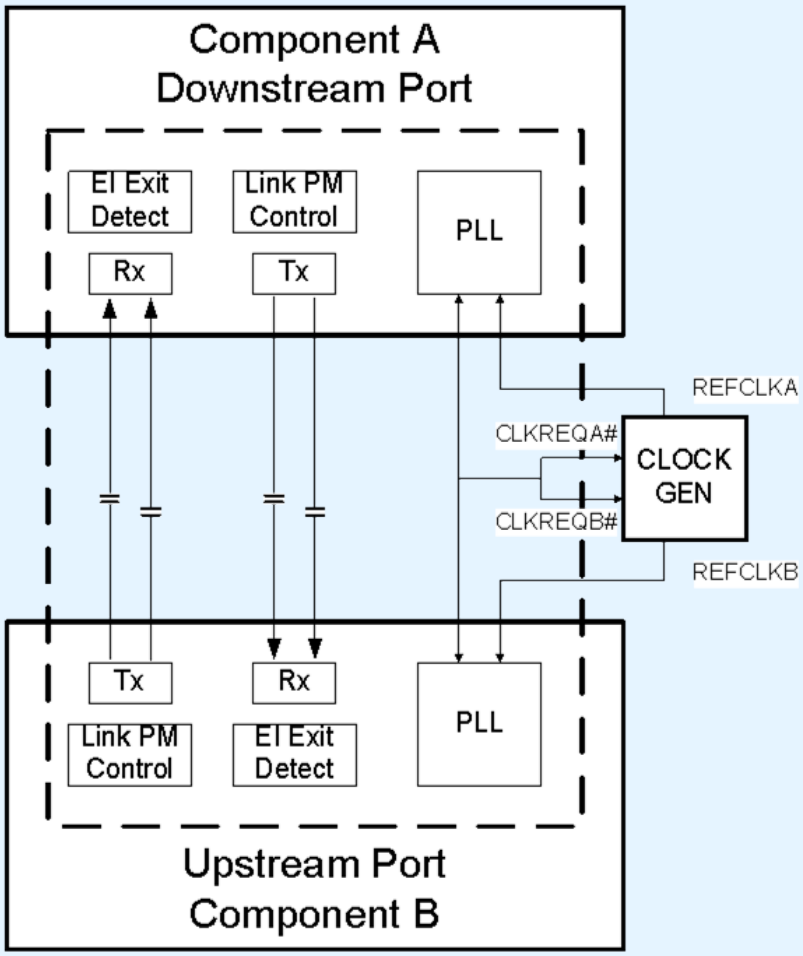

Figure 5-10 Downstream Port with a Single PLL

Example 2: Upstream component with multiple Downstream Ports, with a common shared PLL, connected to separate Downstream components (see Figure 5-11).

In this example configuration, there are three instances of CLKREQ# signal for the Upstream component (A), one per Downstream Port and a common shared CLKREQ# signal for the Upstream component (A). In this topology the Downstream Port CLKREQ# (CLKREQB#, CLKREQC#) signals are used to connect to the CLKREQ# signal of the Upstream Port of the Downstream components (B and C). The common shared CLKREQ# (CLKREQA#) signal for the Upstream component is used to request the reference clock for the shared PLL. The PLL control logic in Upstream component (A) can only be turned off and CLKREQA# be deasserted when both the Downstream Ports are in L1.1 or L1.2 Substates, and all internal (A) consumers of the PLL don't require a clock.

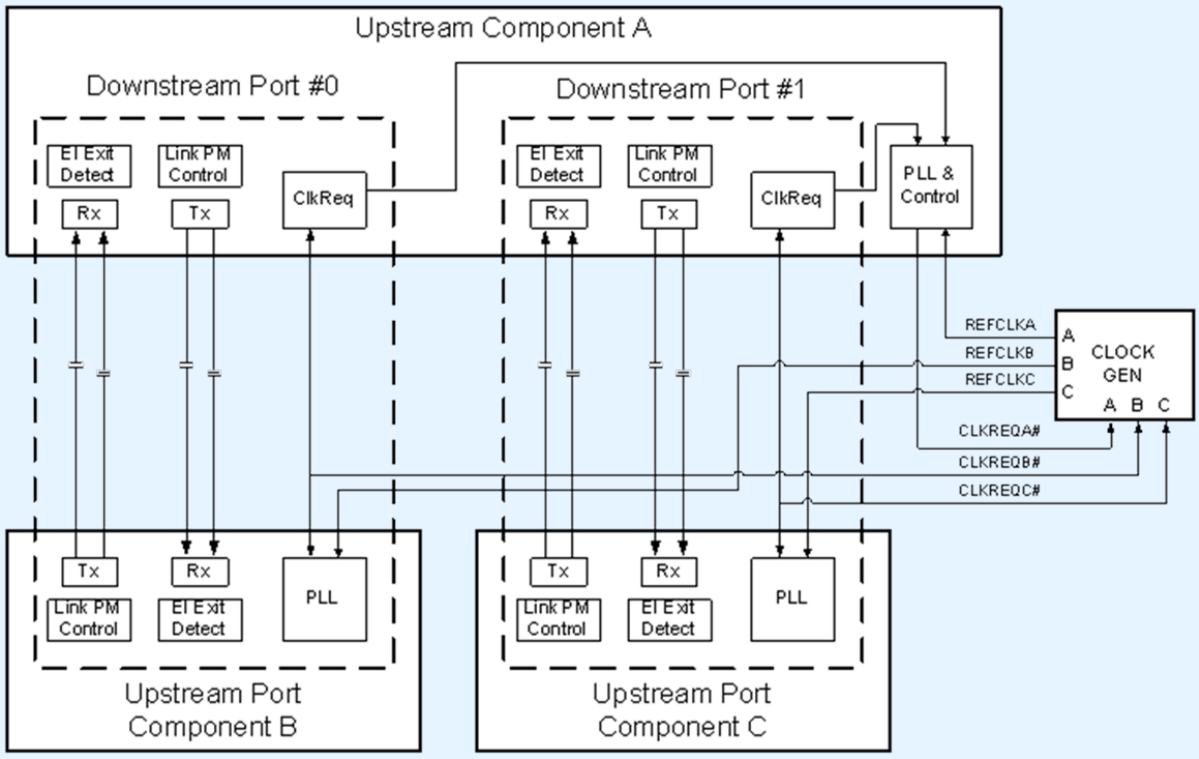

Figure 5-11 Multiple Downstream Ports with a shared PLL

It is necessary for board implementers to consider what CLKREQ# topologies will be supported by components in order to make appropriate board level connections to support L1 PM Substates and for the reference clock generation.

IMPLEMENTATION NOTE

Avoiding Unintended Interactions Between L1 PM Substates and the LTSSM

It is often the case that implementation techniques which save power will also increase the latency to return to normal operation. When implementing L1 PM Substates, it is important for the implementer to ensure that any added delays will not negatively interact with other elements of the platform. It is particularly important to ensure that LTSSM timeout conditions are not unintentionally triggered. Although typical implementations will not approach the latencies that would cause such interactions, the responsibility lies with the implementer to ensure that correct overall operation is achieved.

5.5.1 Entry conditions for L1 PM Substates and L1.0 Requirements

The Link is considered to be in PCI-PM L1.0 when the L1 PM Substate is L1.0 and the LTSSM entered L1 through PCI-PM compatible power management. The Link is considered to be in ASPM L1.0 when the L1 PM Substate is in L1.0 and LTSSM entered L1 through ASPM.

The following rules define how the L1.1 and L1.2 substates are entered:

- Both the Upstream and Downstream Ports must monitor the logical state of the CLKREQ# signal.
- When in PCI-PM L1.0 and the PCI-PM L1.2 Enable bit is Set, the L1.2 substate must be entered when CLKREQ# is deasserted.
- When in PCI-PM L1.0 and the PCI-PM L1.1 Enable bit is Set, the L1.1 substate must be entered when CLKREQ# is deasserted and the PCI-PM L1.2 Enable bit is Clear.
- When in ASPM L1.0 and the ASPM L1.2 Enable bit is Set, the L1.2 substate must be entered when CLKREQ# is deasserted and all of the following conditions are true:
  - The reported snooped LTR value last sent or received by this Port is greater than or equal to the value set by the LTR_L1.2_THRESHOLD Value and Scale fields, or there is no snoop service latency requirement.
  - The reported non-snooped LTR last sent or received by this Port value is greater than or equal to the value set by the LTR_L1.2_THRESHOLD Value and Scale fields, or there is no non-snoop service latency requirement.
- When in ASPM L1.0 and the ASPM L1.1 Enable bit is Set, the L1.1 substate must be entered when CLKREQ# is deasserted and the conditions for entering the L1.2 substate are not satisfied.

When the entry conditions for L1.2 are satisfied, the following rules apply:

- Both the Upstream and Downstream Ports must monitor the logical state of the CLKREQ# input signal.
- An Upstream Port must not deassert CLKREQ# until the Link has entered L1.0.
- It is permitted for either Port to assert CLKREQ# to prevent the Link from entering L1.2.
- A Downstream Port intending to block entry into L1.2 must assert CLKREQ# before the Link enters L1.
- When CLKREQ# is deasserted the Ports enter the L1.2.Entry substate of L1.2.

If a Downstream Port is in PCI-PM L1.0 and PCI-PM L1.1 Enable and/or PCI-PM L1.2 Enable are Set, or if a Downstream Port is in ASPM L1.0 and ASPM L1.1 Enable and/or ASPM L1.2 Enable are Set, and the Downstream Port initiates an exit to Recovery without having entered L1.1 or L1.2, the Downstream Port must assert CLKREQ# until the Link exits Recovery.

5.5.2 L1.1 Requirements

Both Upstream and Downstream Ports are permitted to deactivate mechanisms for electrical idle (EI) exit detection and Refclk activity detection if implemented, however both ports must maintain common mode.

5.5.2.1 Exit from L1.1

If either the Upstream or Downstream Port needs to initiate exit from L1.1, it must assert CLKREQ# until the Link exits Recovery. The Upstream Port must assert CLKREQ# on entry to Recovery, and must continue to assert CLKREQ# until the next entry into L1, or other state allowing CLKREQ# deassertion.

- Next state is L1.0 if CLKREQ# is asserted.
  - The Refclk will eventually be turned on as defined in the PCI Express Mini CEM spec, which may be delayed according to the LTR advertised by the Upstream Port.

Figure 5-12 illustrates entry into L1.1 with exit driven by the Upstream Port.

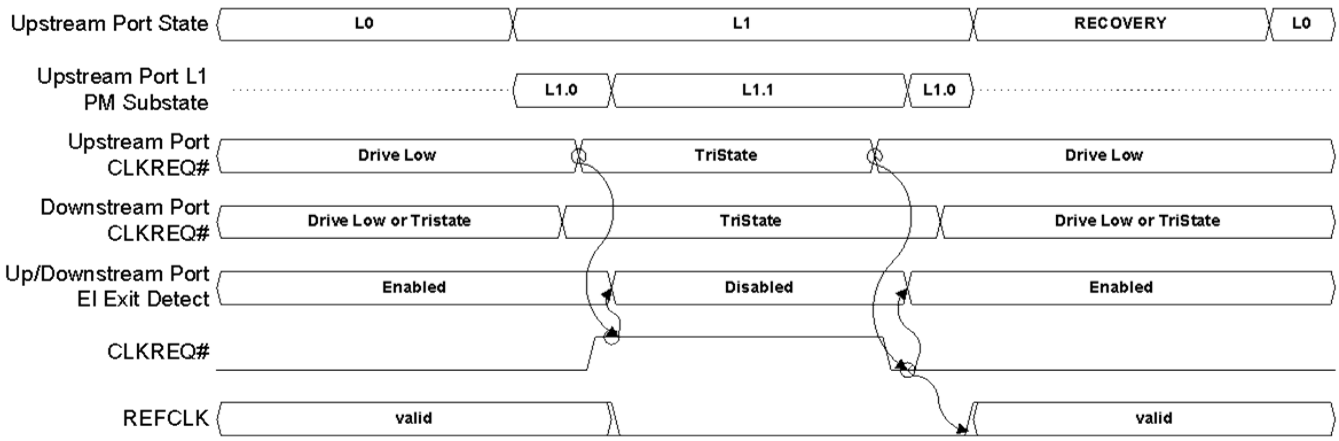

Figure 5-12 Example: L1.1 Waveforms Illustrating Upstream Port Initiated Exit

Figure 5-13 illustrates entry into L1.1 with exit driven by the Downstream Port.

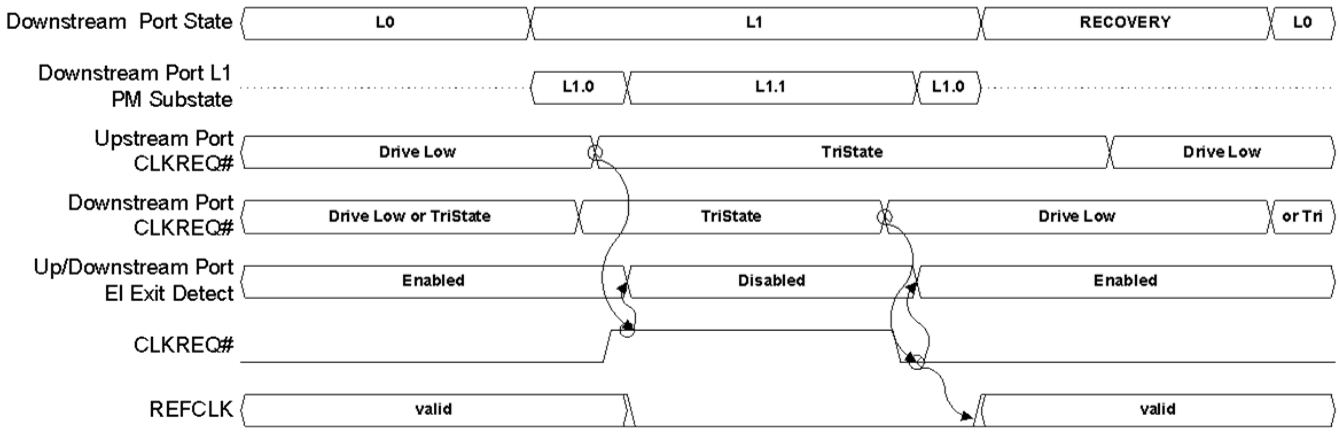

Figure 5-13 Example: L1.1 Waveforms Illustrating Downstream Port Initiated Exit

5.5.3 L1.2 Requirements

All Link and PHY state must be maintained during L1.2, or must be restored upon exit using implementation-specific means, and the LTSSM and corresponding Port state upon exit from L1.2 must be indistinguishable from the L1.0 LTSSM and Port state.

L1.2 has additional requirements that do not apply to L1.1 These requirements are documented in this section.

L1.2 has three substates, which are defined below (see Figure 5-14).

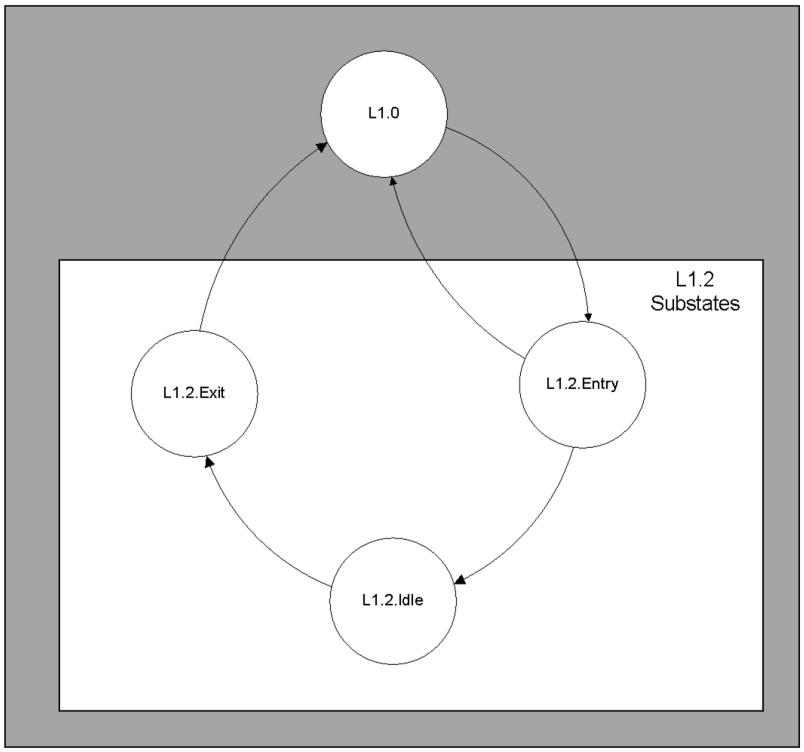

Figure 5-14 L1.2 Substates

5.5.3.1 L1.2.Entry

L1.2.Entry is a transitional state on entry into L1.2 to allow time for Refclk to turn off and to ensure both Ports have observed CLKREQ# deasserted. The following rules apply to L1.2.Entry:

- Both Upstream and Downstream Ports continue to maintain common mode.
- Both Upstream and Downstream Ports may turn off their electrical idle (EI) exit detect circuitry.
- The Upstream and Downstream Ports must not assert CLKREQ# in this state.
- Refclk must be turned off within $T_{L1O\_REFCLK\_OFF}$.
- Next state is L1.0 if CLKREQ# is asserted, else the next state is L1.2.Idle after waiting for T_{POWER_OFF}.

Note that there is a boundary condition which can occur when one Port asserts CLKREQ# shortly after the other Port deasserts CLKREQ#, but before the first Port has observed CLKREQ# deasserted. This is an unavoidable boundary condition that implementations must handle correctly. An example of this condition is illustrated in Figure 5-15 .

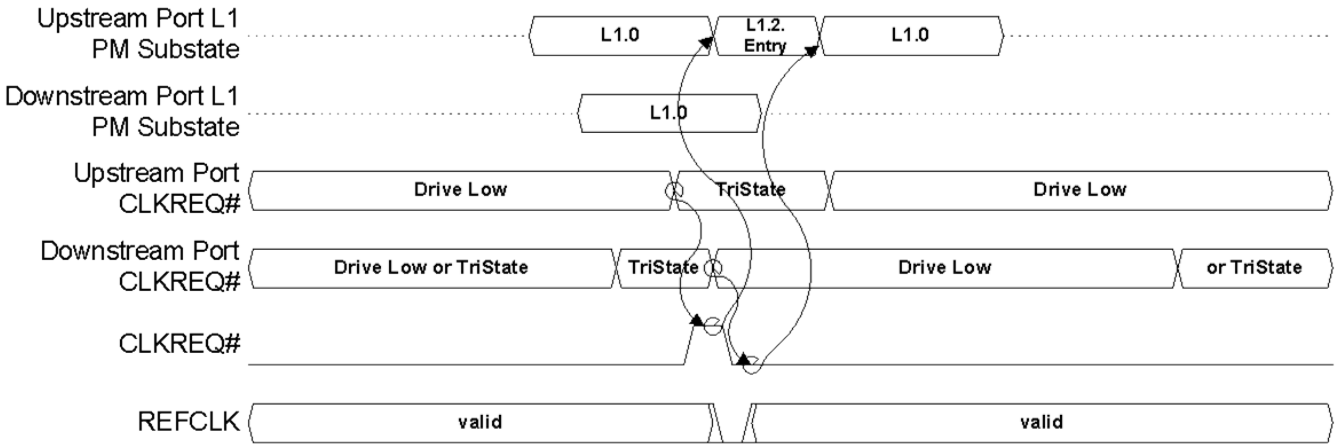

Figure 5-15 Example: Illustration of Boundary Condition due to Different Sampling of CLKREQ#

5.5.3.2 L1.2.Idle

When requirements for the entry into L1.2.Idle state (see Section 5.5.1) have been satisfied then the Ports enter the L1.2.Idle substate. The following rules apply in L1.2.Idle:

- Both Upstream and Downstream Ports may power-down any active logic, including circuits required to maintain common mode.
- The PHY of both Upstream and Downstream Ports may have their power removed.

The following rules apply for L1.2.Idle state when using the CLKREQ#-based mechanism:

- If either the Upstream or Downstream Port needs to exit L1.2, it must assert CLKREQ# after ensuring that $T_{L1.2}$ has been met.
- If the Downstream Port is initiating exit from L1, it must assert CLKREQ# until the Link exits Recovery. The Upstream Port must assert CLKREQ# on entry to Recovery, and must continue to assert CLKREQ# until the next entry into L1, or other state allowing CLKREQ# deassertion.
- If the Upstream Port is initiating exit from L1, it must continue to assert CLKREQ# until the next entry into L1, or other state allowing CLKREQ# deassertion.
- Both the Upstream and Downstream Ports must monitor the logical state of the CLKREQ# input signal.
- Next state is L1.2.Exit if CLKREQ# is asserted.

5.5.3.3 L1.2.Exit

This is a transitional state on exit from L1.2 to allow time for both devices to power up. In L1.2.Exit, the following rules apply:

- The PHYs of both Upstream and Downstream Ports must be powered.
- It must not be assumed that common mode has been maintained.

5.5.3.3.1 Exit from L1.2

- The following rules apply for L1.2.Exit using the CLKREQ#-based mechanism:
- Both Upstream and Downstream Ports must power up any circuits required for L1.0, including circuits required to maintain common mode.
- The Upstream and Downstream Ports must not change their driving state of CLKREQ# in this state.
- Refclk must be turned on no earlier than $T_{L10\_REFCLK\_ON}$ minimum time, and may take up to the amount of time allowed according to the LTR advertised by the Endpoint before becoming valid.
- Next state is L1.0 after waiting for T_{POWER_ON}.
  - Common mode is permitted to be established passively during L1.0, and actively during Recovery. In order to ensure common mode has been established, the Downstream Port must maintain a timer, and the Downstream Port must continue to send TS1 training sequences until a minimum of $T_{COMMONMODE}$ has elapsed since the Downstream Port has started transmitting TS1 training sequences and has detected electrical idle exit on any Lane of the configured Link.

Figure 5-16 illustrates the signal relationships and timing constraints associated with L1.2 entry and Upstream Port initiated exit.

Figure 5-17 illustrates the signal relationships and timing constraints associated with L1.2 entry and Downstream Port initiated exit.

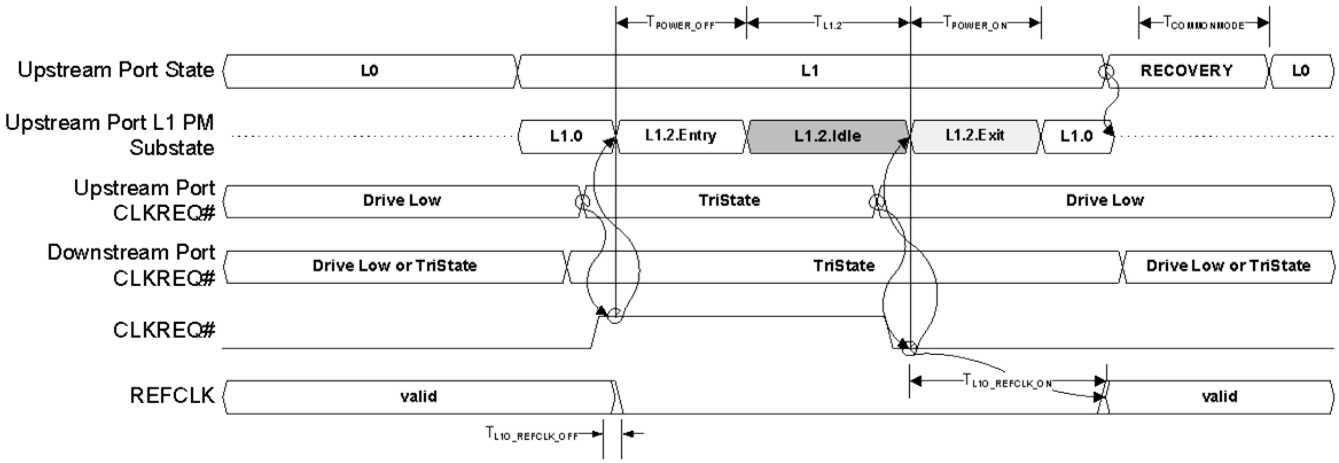

Figure 5-16 Example: L1.2 Waveforms Illustrating Upstream Port Initiated Exit

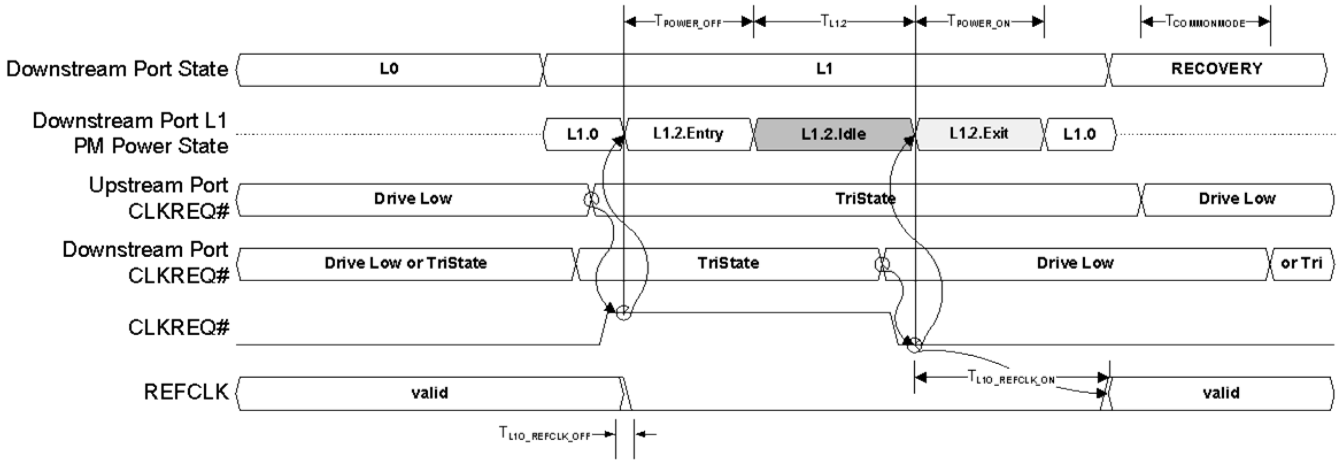

Figure 5-17 Example: L1.2 Waveforms Illustrating Downstream Port Initiated Exit

5.5.4 L1 PM Substates Configuration

L1 PM Substates is considered enabled on a Port when any combination of the ASPM L1.1 Enable, ASPM L1.2 Enable, PCI-PM L1.1 Enable and PCI-PM L1.2 Enable bits associated with that Port are Set.

An L1 PM Substate enable bit must only be Set in the Upstream and Downstream Ports on a Link when the corresponding supported capability bit is Set by both the Upstream and Downstream Ports on that Link, otherwise the behavior is undefined.

The Setting of any enable bit must be performed at the Downstream Port before the corresponding bit is permitted to be Set at the Upstream Port. If any L1 PM Substates enable bit is at a later time to be cleared, the enable bit(s) must be cleared in the Upstream Port before the corresponding enable bit(s) are permitted to be cleared in the Downstream Port.

If setting either or both of the enable bits for ASPM L1 PM Substates, both ports must be configured as described in this section while ASPM L1 is disabled.

If setting either or both of the enable bits for PCI-PM L1 PM Substates, both ports must be configured as described in this section while in D0.

Prior to setting either or both of the enable bits for L1.2, the values for T_{POWER_ON}, Common_Mode_Restore_Time, and, if the ASPM L1.2 Enable bit is to be Set, the LTR_L1.2_THRESHOLD (both Value and Scale fields) must be programmed.

The T_{POWER_ON} and Common_Mode_Restore_Time fields must be programmed to the appropriate values based on the components and AC coupling capacitors used in the connection linking the two components. The determination of these values is design implementation specific.

When both the ASPM L1.2 Enable and PCI-PM L1.2 Enable bits are cleared, it is not required to program the T_{POWER_ON}, Common_Mode_Restore_Time, and LTR_L1.2_THRESHOLD Value and Scale fields, and hardware must not rely on these fields to have any particular values.

When programming LTR_L1.2_THRESHOLD Value and Scale fields, identical values must be programmed in both Ports.

5.5.5 L1 PM Substates Timing Parameters

Table 5-11 defines the timing parameters associated with the L1.2 substates mechanism.

Table 5-11 L1.2 Timing Parameters

Parameter	Description	Min	Max	Units
T_{POWER_OFF}	CLKREQ# deassertion to entry into the L1.2.Idle substate		2	µs
$T_{COMMONMODE}$	Restoration of Refclk to restoration of common mode established through active transmission of TS1 training sequences (see Section 5.5.3.3.1)	Programmable in range from 0 to 255		µs
$T_{L1O\_REFCLK\_OFF}$	CLKREQ# deassertion to Refclk reaching idle electrical state when entering L1.2	0	100	ns
$T_{L1O\_REFCLK\_ON}$	CLKREQ# assertion to Refclk valid when exiting L1.2	T_{POWER_ON}	LTR value advertised by the Endpoint	µs
T_{POWER_ON}	The minimum amount of time that each component must wait in L1.2.Exit after sampling CLKREQ# asserted before actively driving the interface to ensure no device is ever actively driving into an unpowered component.	Set in the L1 PM Substates Control 2 Register (range from 0 to 3100)		µs
$T_{L1.2}$	Time a Port must stay in L1.2 when CLKREQ# must remain inactive	4		µs

5.5.6 *Link Activation*

Link Activation is an optional mechanism to temporarily disable L1 Substates. Link Activation is used to bring a Link out of L1.1/L1.2, avoiding potential stalls. An example of one such stall is the stall associated with a Configuration Write to perform a $D3_{Hot}$ to D0 transition. Link Activation can also be used to indirectly indicate to a Device that it should avoid long-latency internal power management during latency-sensitive or time critical operations.

The following rules apply to Link Activation:

- A Downstream Port is permitted to support Link Activation, as indicated by the Link Activation Supported bit in the L1 PM Substates Capabilities Register being Set.
- The Link Activation Control bit must have no effect on Port behavior unless one or more of the following bits are Set:
  - PCI-PM L1.2 Enable
  - PCI-PM L1.1 Enable
- When the Link Activation Control bit is Set, the Port that is about to enter L1 must assert, and while in L1 maintain as asserted, the CLKREQ# signal.
- If the Link Activation Control bit is Clear, the Link Activation mechanism does not impose any additional requirements on the state of the CLKREQ# signal.
- If the Port is enabled for edge-triggered interrupt signaling using MSI or MSI-X, an interrupt message must be sent every time the logical AND of the following conditions transitions from FALSE to TRUE:
  - The associated vector is unmasked (not applicable if MSI does not support PVM)
  - The Link Activation Interrupt Enable bit is Set
  - The Link Activation Control bit is Set
  - The Link Activation Status bit is Set. Note that Link Activation interrupts always use the MSI or MSI-X vector indicated by the Interrupt Message Number field in the PCI Express Capabilities Register.

- If the Port is enabled for level-triggered interrupt signaling using the INTx messages, the virtual INTx wire must be asserted whenever and as long as the following conditions are satisfied:
  - The Interrupt Disable bit in the Command Register is Clear.
  - The Link Activation Interrupt Enable bit is Set
  - The Link Activation Control bit is Set
  - The Link Activation Status bit is Set
- The Link Activation Status bit must be Set every time the logical AND of the following conditions transitions from FALSE to TRUE:
  - Either the PCI-PM L1.2 Enable bit or the PCI-PM L1.1 Enable bit (or both) are Set
  - The Link Activation Control bit is Set
  - The Link is not in an L1 Substate

5.6 Auxiliary Power Support

The specific definition and requirements associated with auxiliary power are form-factor specific, and the terms “auxiliary power” and “Vaux” should be understood in reference to the specific form factor in use. The specific mechanism(s) for supplying auxiliary power are not defined in this specification. The following text defines requirements that apply in all form factors.

PCI Express PM provides a Aux Power PM Enable bit in the Device Control Register that provides the means for enabling a Function to draw the maximum allowance of auxiliary current independent of its level of support for PME generation.

A Function requests auxiliary power allocation by specifying a non-zero value in the Aux_Current field of the PMC register. Refer to Chapter 7 for the Aux Power PM Enable register bit assignment, and access mechanism.

Allocation of auxiliary power using Aux Power PM Enable is determined as follows:

Aux Power PM Enable = 1b:
Auxiliary power is allocated as requested in the Aux_Current field of the PMC register, independent of the PME_En bit in the PMSCR. The PME_En bit still controls the ability to master PME.

Aux Power PM Enable = 0b:
Auxiliary power allocation is controlled by the PME_En bit as defined in Section 7.5.2.2 .

The Aux Power PM Enable bit is sticky (see Section 7.4) so its state is preserved in the $D3_{Cold}$ state, and is not affected by the transitions from the $D3_{Cold}$ state to the $D0_{uninitialized}$ state.

5.7 Power Management System Messages and DLLPs

Table 5-12 defines the location of each PM packet in the PCI Express stack.

Table 5-12 Power Management System Messages and DLLPs

Packet	Type
PM_Enter_L1	DLLP
PM_Enter_L23	DLLP

Packet	Type
PM_Active_State_Request_L1	DLLP
PM_Request_Ack	DLLP
PM_Active_State_Nak	Transaction Layer Message
PM_PME	Transaction Layer Message
PME_Turn_Off	Transaction Layer Message
PME_TO_Ack	Transaction Layer Message

For information on the structure of the power management DLLPs, refer to Section 3.5 .

Power Management Messages follow the general rules for all Messages. Power Management Message fields follow the following rules:

- Length field is Reserved.
- Attribute field must be set to the default values (all 0's).
- Address field is Reserved.
- Requester ID - see Table 2-20 in Section 2.2.8.2 .
- Traffic Class field must use the default class (TC0).

5.8 PCI Function Power State Transitions

All PCI-PM power management state changes are explicitly controlled by software except for Fundamental Reset which brings all Functions to the $D0_{uninitialized}$ state. Figure 5-18 shows all supported state transitions. The unlabeled arcs represent a software initiated state transition (Set Power State operation).

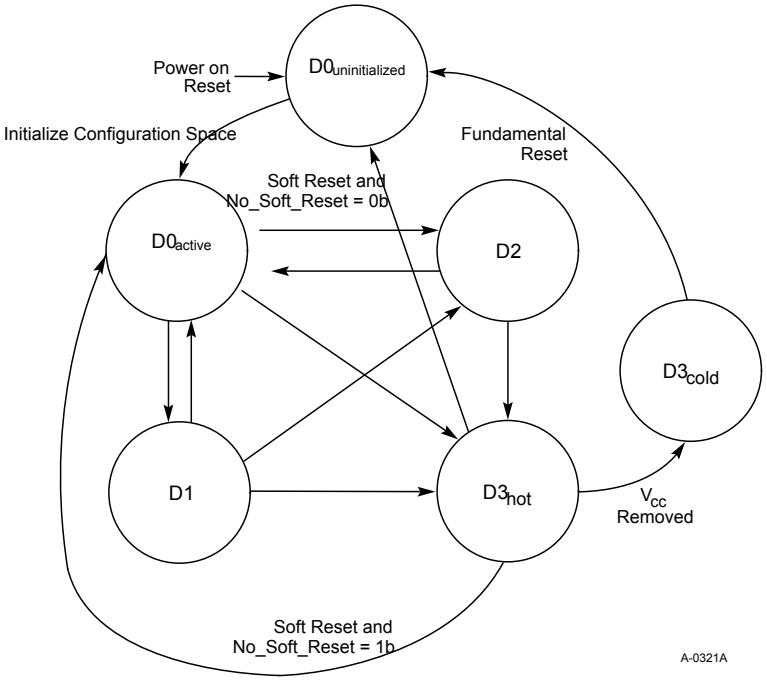

Figure 5-18 Function Power Management State Transitions

5.9 State Transition Recovery Time Requirements

Table 5-13 shows the minimum recovery times that system software must allow between the time that a Function is programmed to change state and the time that the function is next accessed (including Configuration Space), unless Readiness Notifications (see Section 6.23) is used to indicate modified values to system software. For bridge Functions, this delay also constitutes a minimum delay between when the bridge's state is changed and when any Function on the logical bus that it originates can be accessed.

Table 5-13 PCI Function State Transition Delays

Initial State	Next State	Minimum System Software Guaranteed Delays
D0	D1	0
D0 or D1	D2	200 ms
D0, D1 or D2	$D3_{Hot}$	10 ms
D1	D0	0
D2	D0	200 ms
$D3_{Hot}$	D0	10 ms

5.10 PCI Bridges and Power Management

With power management under the direction of the operating system, each class of Functions must have a clearly defined criteria for feature availability as well as what functional context must be preserved when operating in each of the power management states. Some example Device-Class specifications have been proposed as part of the ACPI specification for various Functions ranging from audio to network add-in cards. While defining Device-Class specific behavioral policies for most Functions is outside the scope of this specification, defining the required behavior for PCI bridge functions is within the scope of this specification. The definitions here apply to all three types of PCIe Bridges:

- Host bridge, PCI Express to expansion bus bridge, or other ACPI enumerated bridge
- Switches
- PCI Express to PCI bridge
- PCI-to-CardBus bridge

The mechanisms for controlling the state of these Functions vary somewhat depending on which type of Originating Device is present. The following sections describe how these mechanisms work for the three types of bridges.

This section details the power management policies for PCI Express Bridge Functions. The PCI Express Bridge Function can be characterized as an Originating Device with a secondary bus downstream of it. This section describes the relationship of the bridge function's power management state to that of its secondary bus.

The shaded regions in Figure 5-19 illustrate what is discussed in this section.

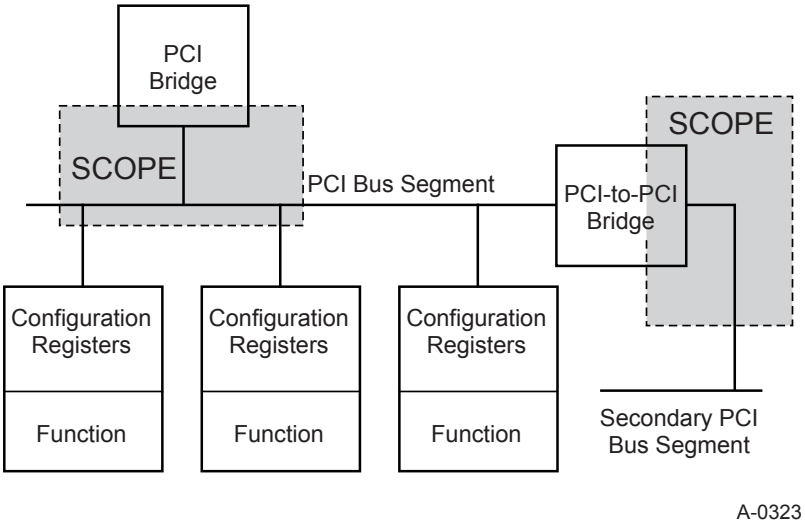

Figure 5-19 PCI Express Bridge Power Management Diagram

As can be seen from Figure 5-19 , the PCI Express Bridge behavior described in this chapter is common, from the perspective of the operating system, to host bridges, Switches, and PCI Express to PCI bridges.

It is the responsibility of the system software to ensure that only valid, workable combinations of bus and downstream Function power management states are used for a given bus and all Functions residing on that bus.

5.10.1 Switches and PCI Express to PCI Bridges

The power management policies for the secondary bus of a Switch or PCI Express to PCI bridge are identical to those defined for any Bridge Function.

The BPCC_En and B2_B3# bus power/clock control fields in the Bridge Function's PMCSR_BSE register support the same functionality as for any other Bridges.

5.11 Power Management Events

There are two varieties of Power Management Events:

- Wakeup Events
- PME Generation

A Wakeup Event is used to request that power be turned on.

A PME Generation Event is used to identify to the system the Function requesting that power be turned on.

In conventional PCI, both events are associated with the PME# signal. The PME# signal is asserted by a Function to request a change in its power management state. When the PME_En bit is Set and the event occurs, the Function sets the PME_Status bit and asserts the PME# signal. It keeps the PME# signal asserted until either the PME_En bit or the PME_Status are Cleared (typically by software).

In PCI Express, the Wakeup Event is associated with the WAKE# signal. If supported, the WAKE# signal is defined in the associated form factor specification and is used by a Function to request a change in its PCI-PM power management state when the Function is in $D3_{Cold}$ and PME_En is Set.

In PCI Express, after main power has been restored and the Link is trained, the Function(s) that initiated the wakeup (e.g., that asserted WAKE#), sends a PM_PME Message to the Root Complex. The PM_PME Message provides the Root Complex with the identity of the requesting Function(s) without requiring software to poll for the PME_Status bit being Set.

System Architecture

6.

This chapter addresses various aspects of PCI Express interconnect architecture in a platform context.

6.1 Interrupt and PME Support

The PCI Express interrupt model supports two mechanisms:

- INTx emulation
- Message Signaled Interrupt (MSI/MSI-X)

For legacy compatibility, PCI Express provides a PCI INTx emulation mechanism to signal interrupts to the system interrupt controller (typically part of the Root Complex). This mechanism is compatible with existing PCI software, and provides the same level and type of service as the corresponding PCI interrupt signaling mechanism and is independent of system interrupt controller specifics. This legacy compatibility mechanism allows boot device support without requiring complex BIOS-level interrupt configuration/control service stacks. It virtualizes PCI physical interrupt signals by using an in-band signaling mechanism.

If an implementation supports interrupts, then this specification requires support of either MSI or MSI-X or both. PCI Compatible INTx interrupt emulation is optional. Switches are required to support forwarding the INTx interrupt emulation Messages (see Section 2.2.8.1). The PCI Express MSI and MSI-X mechanisms are compatible with those originally defined in [PCI].

6.1.1 Rationale for PCI Express Interrupt Model

PCI Express takes an evolutionary approach from PCI with respect to interrupt support.

As required for PCI/PCI-X interrupt mechanisms, each device Function is required to differentiate between INTx and MSI/MSI-X modes of operation. The device complexity required to support both schemes is no different than that for PCI/PCI-X devices. The advantages of this approach include:

- Compatibility with existing PCI Software Models
- Direct support for boot devices
- Easier End of Life (EOL) for INTx legacy mechanisms.

The existing software model is used to differentiate INTx vs. MSI/MSI-X modes of operation; thus, no special software support is required for PCI Express.

6.1.2 PCI-compatible INTx Emulation

PCI Express emulates the PCI interrupt mechanism including the Interrupt Pin and Interrupt Line registers of the PCI Configuration Space for PCI device Functions. PCI Express non-Switch devices may optionally support these registers for backwards compatibility. Switch devices are required to support them. Actual interrupt signaling uses in-band Messages rather than being signaled using physical pins.

Two types of Messages are defined, Assert_INTx and Deassert_INTx, for emulation of PCI INTx signaling, where x is A, B, C, and D for respective PCI interrupt signals. These Messages are used to provide "virtual wires" for signaling interrupts across a Link. Switches collect these virtual wires and present a combined set at the Switch's Upstream Port. Ultimately, the virtual wires are routed to the Root Complex which maps the virtual wires to system interrupt resources. Devices must use assert/deassert Messages in pairs to emulate PCI interrupt level-triggered signaling. Actual mapping of PCI Express INTx emulation to system interrupts is implementation specific as is mapping of physical interrupt signals in conventional PCI.

The legacy INTx emulation mechanism may be deprecated in a future version of this specification.

6.1.3 INTx Emulation Software Model

The software model for legacy INTx emulation matches that of PCI. The system BIOS reporting of chipset/platform interrupt mapping and the association of each device Function's interrupt with PCI interrupt lines is handled in exactly the same manner as with conventional PCI systems. Legacy software reads from each device Function's Interrupt Pin register to determine if the Function is interrupt driven. A value between 01h and 04h indicates that the Function uses an emulated interrupt pin to generate an interrupt.

Note that similarly to physical interrupt signals, the INTx emulation mechanism may potentially cause spurious interrupts that must be handled by the system software.

6.1.4 MSI and MSI-X Operation

Message Signaled Interrupts (MSI) is an optional feature that enables a device Function to request service by writing a system-specified data value to a system-specified address (using a DWORD Memory Write transaction). System software initializes the message address and message data (from here on referred to as the "vector") during device configuration, allocating one or more vectors to each MSI-capable Function.

Interrupt latency (the time from interrupt signaling to interrupt servicing) is system dependent. Consistent with current interrupt architectures, Message Signaled Interrupts do not provide interrupt latency time guarantees.

MSI-X defines a separate optional extension to basic MSI functionality. Compared to MSI, MSI-X supports a larger maximum number of vectors per Function, the ability for software to control aliasing when fewer vectors are allocated than requested, plus the ability for each vector to use an independent address and data value, specified by a table that resides in Memory Space. However, most of the other characteristics of MSI-X are identical to those of MSI.

For the sake of software backward compatibility, MSI and MSI-X use separate and independent Capability structures. On Functions that support both MSI and MSI-X, system software that supports only MSI can still enable and use MSI without any modification. MSI functionality is managed exclusively through the MSI Capability structure, and MSI-X functionality is managed exclusively through the MSI-X Capability structure.

A Function is permitted to implement both MSI and MSI-X, but system software is prohibited from enabling both at the same time. If system software enables both at the same time, the behavior is undefined.

All PCI Express device Functions that are capable of generating interrupts must support MSI or MSI-X or both. The MSI and MSI-X mechanisms deliver interrupts by performing Memory Write transactions. MSI and MSI-X are edge-triggered interrupt mechanisms; neither [PCI] nor this specification support level-triggered MSI/MSI-X interrupts. Certain PCI devices and their drivers rely on INTx-type level-triggered interrupt behavior (addressed by the PCI Express legacy INTx emulation mechanism). To take advantage of the MSI or MSI-X capability and edge-triggered interrupt semantics, these devices and their drivers may have to be redesigned.

MSI and MSI-X each support Per-Vector Masking (PVM). PVM is an optional[86] extension to MSI, and a standard feature with MSI-X. A Function that supports the PVM extension to MSI is backward compatible with system software that is unaware

of the extension. MSI-X also supports a Function Mask bit, which when Set masks all of the vectors associated with a Function.

A Legacy Endpoint that implements MSI is required to support either the 32-bit or 64-bit Message Address version of the MSI Capability structure. A PCI Express Endpoint that implements MSI is required to support the 64-bit Message Address version of the MSI Capability structure.

The Requester of an MSI/MSI-X transaction must set the No Snoop and Relaxed Ordering attributes of the Transaction Descriptor to 0b. A Requester of an MSI/MSI-X transaction is permitted to Set the ID-Based Ordering (IDO) attribute if use of the IDO attribute is enabled.

Note that, unlike INTx emulation Messages, MSI/MSI-X transactions are not restricted to the TC0 traffic class.

IMPLEMENTATION NOTE

Synchronization of Data Traffic and Message Signaled Interrupts

MSI/MSI-X transactions are permitted to use the TC that is most appropriate for the device's programming model. This is generally the same TC as is used to transfer data; for legacy I/O, TC0 should be used.

If a device uses more than one TC, it must explicitly ensure that proper synchronization is maintained between data traffic and interrupt Message(s) not using the same TC. Methods for ensuring this synchronization are implementation specific. One option is for a device to issue a zero-length Read (as described in Section 2.2.5) using each additional TC used for data traffic prior to issuing the MSI/MSI-X transaction. Other methods are also possible. Note, however, that platform software (e.g., a device driver) is generally only capable of issuing transactions using TC0.

Within a device, different Functions are permitted to implement different sets of the MSI/MSI-X/INTx interrupt mechanisms, and system software manages each Function's interrupt mechanisms independently.

6.1.4.1 MSI Configuration

In this section, all register and field references are in the context of the MSI Capability structure.

System software reads the Message Control register to determine the Function's MSI capabilities.

System software reads the Multiple Message Capable field (bits 3-1 of the Message Control register) to determine the number of requested vectors. MSI supports a maximum of 32 vectors per Function. System software writes to the Multiple Message Enable field (bits 6-4 of the Message Control register) to allocate either all or a subset of the requested vectors. For example, a Function can request four vectors and be allocated either four, two, or one vector. The number of vectors requested and allocated is aligned to a power of two (that is, a Function that requires three vectors must request four).

If the Per-Vector Masking Capable bit (bit 8 of the Message Control register) is Set and system software supports Per-Vector Masking, system software may mask one or more vectors by writing to the Mask Bits register.

If the 64-bit Address Capable bit (bit 7 of the Message Control register) is Set, system software initializes the MSI Capability structure's Message Address register (specifying the lower 32 bits of the message address) and the Message Upper Address register (specifying the upper 32 bits of the message address) with a system-specified message address. System software may program the Message Upper Address register to zero so that the Function uses a 32-bit address for

86. Exception: Within an SR-IOV Device, any PFs or VFs that implement MSI must implement MSI PVM.

the MSI transaction. If this bit is Clear, system software initializes the MSI Capability structure's Message Address register (specifying a 32-bit message address) with a system specified message address.

System software initializes the MSI Capability structure's Message Data register with the lower 16 bits of a system specified data value. When the Extended Message Data Capable bit is Clear, care must be taken to initialize only the Message Data register (i.e., a 2-byte value) and not modify the upper two bytes of that DWORD location.

If the Extended Message Data Capable bit is Set and system software supports 32-bit vector values, system software may initialize the MSI capability structure's Extended Message Data register with the upper 16 bits of a system specified data value, and then Set the Extended Message Data Enable bit.

6.1.4.2 MSI-X Configuration

In this section, all register and field references are in the context of the MSI-X Capability, MSI-X Table, and MSI-X PBA structures.

System software allocates address space for the Function's standard set of Base Address registers and sets the registers accordingly. One of the Function's Base Address registers includes address space for the MSI-X Table, though the system software that allocates address space does not need to be aware of which Base Address register this is, or the fact the address space is used for the MSI-X Table. The same or another Base Address register includes address space for the MSI-X PBA, and the same point regarding system software applies.

Depending upon system software policy, system software, device driver software, or each at different times or environments may configure a Function's MSI-X Capability and table structures with suitable vectors. For example, a booting environment will likely require only a single vector, whereas a normal operating system environment for running applications may benefit from multiple vectors if the Function supports an MSI-X Table with multiple entries. For the remainder of this section, "software" refers to either system software or device driver software.

Software reads the Table Size field from the Message Control register to determine the MSI-X Table size. The field encodes the number of table entries as N-1, so software must add 1 to the value read from the field to calculate the number of table entries N. MSI-X supports a maximum table size of 2048 entries.

Software calculates the base address of the MSI-X Table by reading the 32-bit value from the Table Offset/Table BIR register, masking off the lower 3 Table BIR bits, and adding the remaining QWORD-aligned 32-bit Table offset to the address taken from the Base Address register indicated by the Table BIR. Software calculates the base address of the MSI-X PBA using the same process with the PBA Offset/PBA BIR register.

For each MSI-X Table entry that will be used, software fills in the Message Address field, Message Upper Address field, Message Data field, and Vector Control field. The Vector Control field may contain optional Steering Tag fields. Software must not modify the Address, Data, or Steering Tag fields of an entry while it is unmasked. Refer to Section 6.1.4.5 for details.

IMPLEMENTATION NOTE
Special Considerations for QWORD Accesses

Software is permitted to fill in MSI-X Table entry DWORD fields individually with DWORD writes, or software in certain cases is permitted to fill in appropriate pairs of DWORDs with a single QWORD write. Specifically, software is always permitted to fill in the Message Address and Message Upper Address fields with a single QWORD write. If a given entry is currently masked (via its Mask bit or the Function Mask bit), software is permitted to fill in the Message Data and Vector Control fields with a single QWORD write, taking advantage of the fact the Message Data field is guaranteed to become visible to hardware no later than the Vector Control field. However, if software wishes to mask a currently unmasked entry (without Setting the Function Mask bit), software must Set the entry's Mask bit using a DWORD write to the Vector Control field, since performing a QWORD write to the Message Data and Vector Control fields might result in the Message Data field being modified before the Mask bit in the Vector Control field becomes Set.

For potential use by future specifications, the Reserved bits in the Vector Control field must have their default values preserved by software. If software does not preserve their values, the result is undefined.

For each MSI-X Table entry that software chooses not to configure for generating messages, software can simply leave the entry in its default state of being masked.

Software is permitted to configure multiple MSI-X Table entries with the same vector, and this may indeed be necessary when fewer vectors are allocated than requested.

IMPLEMENTATION NOTE
Handling MSI-X Vector Shortages

For the case where fewer vectors are allocated to a Function than desired, software-controlled aliasing as enabled by MSI-X is one approach for handling the situation. For example, if a Function supports five queues, each with an associated MSI-X table entry, but only three vectors are allocated, the Function could be designed for software still to configure all five table entries, assigning one or more vectors to multiple table entries. Software could assign the three vectors {A,B,C} to the five entries as ABCCC, ABBCC, ABCBA, or other similar combinations.

Alternatively, the Function could be designed for software to configure it (using a device specific mechanism) to use only three queues and three MSI-X table entries. Software could assign the three vectors {A,B,C} to the five entries as ABC-, A-B-C, A-CB, or other similar combinations.

6.1.4.3 Enabling Operation

To maintain backward compatibility, the MSI Enable bit in the Message Control Register for MSI and the MSI-X Enable bit in the Message Control Register for MSI-X are each Clear by default (MSI and MSI-X are both disabled). System configuration software Sets one of these bits to enable either MSI or MSI-X, but never both simultaneously. Behavior is undefined if both MSI and MSI-X are enabled simultaneously. A device driver is prohibited from writing this bit to mask a Function's service request. While enabled for MSI or MSI-X operation, a Function is prohibited from using INTx interrupts (if implemented) to request service (MSI, MSI-X, and INTx are mutually exclusive).

6.1.4.4 Sending Messages

Once MSI or MSI-X is enabled (the appropriate bit in one of the Message Control registers is Set), and one or more vectors is unmasked, the Function is permitted to send messages. To send a message, a Function does a DWORD Memory Write to the appropriate message address with the appropriate message data.

For MSI when the Extended Message Data Enable bit is Clear, the DWORD that is written is made up of the value in the MSI Message Data register in the lower two bytes and zeroes in the upper two bytes. For MSI when the Extended Message Data Enable bit is Set, the DWORD that is written is made up of the value in the MSI Message Data register in the lower two bytes and the value in the MSI Extended Message Data register in the upper two bytes.

For MSI, if the Multiple Message Enable field (bits 6-4 of the Message Control Register for MSI) is non-zero, the Function is permitted to modify the low order bits of the message data to generate multiple vectors. For example, a Multiple Message Enable encoding of 010b indicates the Function is permitted to modify message data bits 1 and 0 to generate up to four unique vectors. If the Multiple Message Enable field is 000b, the Function is not permitted to modify the message data.

For MSI-X, the MSI-X Table contains at least one entry for every allocated vector, and the 32-bit Message Data field value from a selected table entry is used in the message without any modification to the low-order bits by the Function.

How a Function uses multiple vectors (when allocated) is device dependent. A Function must handle being allocated fewer vectors than requested.

6.1.4.5 Per-vector Masking and Function Masking

Per-Vector Masking (PVM) is an optional[87] feature with MSI, and a standard feature in MSI-X.

Function Masking is a standard feature in MSI-X. When the MSI-X Function Mask bit is Set, all of the Function's entries must behave as being masked, regardless of the per-entry Mask bit values. Function Masking is not supported in MSI, but software can readily achieve a similar effect by Setting all MSI Mask bits using a single DWORD write.

PVM in MSI-X is controlled by a Mask bit in each MSI-X Table entry. While more accurately termed "per-entry masking", masking an MSI-X Table entry is still referred to as "vector masking" so similar descriptions can be used for both MSI and MSI-X. However, since software is permitted to program the same vector (a unique Address/Data pair) into multiple MSI-X table entries, all such entries must be masked in order to guarantee the Function will not send a message using that Address/Data pair.

For MSI and MSI-X, while a vector is masked, the Function is prohibited from sending the associated message, and the Function must Set the associated Pending bit whenever the Function would otherwise send the message. When software unmasks a vector whose associated Pending bit is Set, the Function must schedule sending the associated message, and Clear the Pending bit as soon as the message has been sent. Note that Clearing the MSI-X Function Mask bit may result in many messages needing to be sent.

If a masked vector has its Pending bit Set, and the associated underlying interrupt events are somehow satisfied (usually by software though the exact manner is Function-specific), the Function must Clear the Pending bit, to avoid sending a spurious interrupt message later when software unmasks the vector. However, if a subsequent interrupt event occurs while the vector is still masked, the Function must again Set the Pending bit.

Software is permitted to mask one or more vectors indefinitely, and service their associated interrupt events strictly based on polling their Pending bits. A Function must Set and Clear its Pending bits as necessary to support this "pure polling" mode of operation.

87. Exception: Within an SR-IOV Device, any PFs or VFs that implement MSI must implement MSI PVM.