

Figure 2-49 Calculation of 32-bit ECRC for TLP End to End Data Integrity Protection

## IMPLEMENTATION NOTE

### Protection of TD Bit Inside Switches

It is of utmost importance that Switches insure and maintain the integrity of the TD bit in TLPs that they receive and forward (i.e., by applying a special internal protection mechanism), since corruption of the TD bit will cause the ultimate target device to misinterpret the presence or absence of the TLP Digest field.

Similarly, it is highly recommended that Switches provide internal protection to other Variant fields in TLPs that they receive and forward, as the end-to-end integrity of Variant fields is not sustained by the ECRC.

## IMPLEMENTATION NOTE

### Data Link Layer Does Not Have Internal TLP Visibility

Since the Data Link Layer does not process the TLP header (it determines the start and end of the TLP based on indications from the Physical Layer), it is not aware of the existence of the TLP Digest field, and simply passes it to the Transaction Layer as a part of the TLP.

## 2.7.2 Error Forwarding

Error Forwarding (also known as data poisoning), is indicated by Setting the EP bit. The rules for doing this are specified in Section 2.7.2.2. Here are some examples of cases where Error Forwarding might be used:

- Example #1: A read from main memory encounters an uncorrectable error
- Example #2: Parity error on a PCI write to main memory
- Example #3: Data integrity error on an internal data buffer or cache.

### 2.7.2.1 Error Forwarding Usage Model

- Error Forwarding is only used for Read Completion Data, AtomicOp Completion Data, AtomicOp Request Data, or Write Data, never for the cases when the error is in the “header” (request phase, address/command, etc.). Requests/Completions with header errors cannot be forwarded in general since true destination cannot be positively known and, therefore, forwarding may cause direct or side effects such as data corruption, system failures, etc.
- Error Forwarding is used for controlled propagation of errors through the system, system diagnostics, etc.
- Note that Error forwarding does not cause Link Layer Retry - Poisoned TLPs will be retried only if there are transmission errors on the Link as determined by the TLP error detection mechanisms in the Data Link Layer.
  - The Poisoned TLP may ultimately cause the originator of the request to re-issue it (at the Transaction Layer or above) in the case of read operation or to take some other action. Such use of Error Forwarding information is beyond the scope of this specification.

### **2.7.2.2 Rules For Use of Data Poisoning**

- Support for TLP poisoning in a Transmitter is optional.
- Data poisoning applies only to the data within a Write Request (Posted or Non-Posted), a Message with Data, an AtomicOp Request, a Read Completion, or an AtomicOp Completion.
  - Poisoning of a TLP is indicated by a Set EP bit.
  - Transmitters are permitted to Set the EP bit only for TLPs that include a data payload. The behavior of the Receiver is not specified if the EP bit is Set for any TLP that does not include a data payload.
- If a Transmitter supports data poisoning, TLPs that are known to the Transmitter to include bad data must use the poisoning mechanism defined above.
- If a Downstream Port supports Poisoned TLP Egress Blocking, the Poisoned TLP Egress Blocking Enable bit is Set, and a poisoned TLP targets going out the Egress Port, the Port must handle the TLP as a Poisoned TLP Egress Blocked error unless there is a higher precedence error. See [Section 6.2.3.2.3](#), [Section 6.2.5](#), and [Section 7.9.15.2](#). Further:
  - The Port must not transmit the TLP.
  - If DPC is not triggered and the TLP is a Non-Posted Request, the Port must return a Completion with Unsupported Request Completion Status.
  - If DPC is triggered the Port must behave as described in [Section 2.9.3](#).
- The following Requests with Poisoned data must not modify the value of the target location:
  - Configuration Write Request
  - Any of the following that target a control register or control structure in the Completer: I/O Write Request, Memory Write Request, or non-vendor-defined Message with data
  - AtomicOp Request

Unless there is a higher precedence error, a Completer must handle these Requests as a Poisoned TLP Received error<sup>48</sup>, and the Completer must also return a Completion with a Completion Status of Unsupported Request (UR) if the Request is Non-Posted (see [Section 6.2.3.2.3](#), [Section 6.2.3.2.4](#), and [Section 6.2.5](#)). Regardless of the severity of the reported error, the reported error must be handled as an uncorrectable error, not an Advisory Non-Fatal Error.

A Switch must route the Request the same way it would route the same Request if it were not poisoned, unless the Request targets a location in the Switch itself, in which case the Switch is the Completer for the Request and must follow the above rules.

For some applications it may be desirable for the Completer to use poisoned data in Write Requests that do not target control registers or control structures - such use is not forbidden. Similarly, it may be desirable for the Requester to use data marked poisoned in Completions - such use is also not forbidden. The appropriate use of poisoned information is application specific, and is not discussed in this document.

This document does not define any mechanism for determining which part or parts of the data payload of a Poisoned TLP are actually corrupt and which, if any, are not corrupt.

48. Due to ambiguous language in earlier versions of this specification, a component is permitted to handle this error as an Unsupported Request, but this is strongly discouraged.

## 2.8 Completion Timeout Mechanism

In any split transaction protocol, there is a risk associated with the failure of a Requester to receive an expected Completion. To allow Requesters to attempt recovery from this situation in a standard manner, the Completion Timeout mechanism is defined. This mechanism is intended to be activated only when there is no reasonable expectation that the Completion will be returned, and should never occur under normal operating conditions. Note that the values specified here do not reflect expected service latencies, and must not be used to estimate typical response times.

PCI Express device Functions that issue Requests requiring Completions must implement the Completion Timeout mechanism. An exception is made for Configuration Requests (see below). The Completion Timeout mechanism is activated for each Request that requires one or more Completions when the Request is transmitted. Since Switches do not autonomously initiate Requests that need Completions, the requirement for Completion Timeout support is limited only to Root Complexes, PCI Express-PCI Bridges, and Endpoints.

The Completion Timeout mechanism may be disabled by configuration software. The Completion Timeout limit is set in the Completion Timeout Value field of the Device Control 2 register. A Completion Timeout is a reported error associated with the Requester Function (see [Section 6.2](#)).

Note: A Memory Read Request for which there are multiple Completions must be considered completed only when all Completions have been received by the Requester. If some, but not all, requested data is returned before the Completion Timeout timer expires, the Requester is permitted to keep or to discard the data that was returned prior to timer expiration.

Completion Timeouts for Configuration Requests have special requirements for the support of PCI Express to PCI/PCI Express bridges. PCI Express to PCI/PCI-X Bridges, by default, are not enabled to return Configuration Request Retry Status (CRS) for Configuration Requests to a PCI/PCI-X device behind the Bridge. This may result in lengthy completion delays that must be comprehended by the Completion Timeout value in the Root Complex. System software may enable PCI Express to PCI/PCI-X Bridges to return CRS by setting the Bridge Configuration Retry Enable bit in the Device Control register, subject to the restrictions noted in the [\[PCIe-to-PCI-PCI-X-Bridge-1.0\]](#).

### IMPLEMENTATION NOTE

#### Completion Timeout Prefix/Header Log Capable

The prefix/header of the Request TLP associated with a Completion Timeout may optionally be recorded by Requesters that implement the AER Capability. Support for recording of the prefix/header is indicated by the value of the Completion Timeout Prefix/Header Log Capable bit in the Advanced Error Capabilities and Control register.

A Completion Timeout may be the result of improper configuration, system failure, or Async Removal (see [Section 6.7.6](#)). In order for host software to distinguish a Completion Timeout error after which continued normal operation is not possible (e.g., after one caused by improper configuration or a system failure) from one where continued normal operation is possible (e.g., after an Async Removal), it is strongly encouraged that Requesters log the Request TLP prefix/header associated with the Completion Timeout.

## 2.9 Link Status Dependencies

### 2.9.1 Transaction Layer Behavior in DL\_Down Status

DL\_Down status indicates that there is no connection with another component on the Link, or that the connection with the other component has been lost and is not recoverable by the Physical or Data Link Layers. This section specifies the Transaction Layer's behavior if DPC has not been triggered and the Data Link Layer reports DL\_Down status to the Transaction Layer, indicating that the Link is non-operational. [Section 2.9.3](#) specifies the behavior if DPC has been triggered.

- For a Port with DL\_Down status, the Transaction Layer is not required to accept received TLPs from the Data Link Layer, provided that these TLPs have not been acknowledged by the Data Link Layer. Such TLPs do not modify receive Flow Control credits.

For a Downstream Port, DL\_Down status is handled by:

- Initializing back to their default state any buffers or internal states associated with outstanding requests transmitted Downstream
  - Note: Port configuration registers must not be affected, except as required to update status associated with the transition to DL\_Down.
- For Non-Posted Requests, forming completions for any Requests submitted by the device core for Transmission, returning Unsupported Request Completion Status, then discarding the Requests
  - This is a reported error associated with the Function for the (virtual) Bridge associated with the Port (see [Section 6.2](#)). For Root Ports, the reporting of this error is optional.
  - Non-Posted Requests already being processed by the Transaction Layer, for which it may not be practical to return Completions, are discarded.

Note: This is equivalent to the case where the Request had been Transmitted but not yet Completed before the Link status became DL\_Down.

- These cases are handled by the Requester using the Completion Timeout mechanism.

Note: The point at which a Non-Posted Request becomes “uncompletable” is implementation specific.

- The Port must terminate any PME\_Turn\_Off handshake Requests targeting the Port in such a way that the Port is considered to have acknowledged the PME\_Turn\_Off request (see the Implementation Note in [Section 5.3.3.2.1](#)).
- The Port must handle Vendor Defined Message Requests as described in [Section 2.2.8.6](#) (e.g., silently discard Vendor Defined Type 1 Messages Requests that it is not designed to receive) since the DL\_Down prevents the Request from reaching its targeted Function.
- For all other Posted Requests, discarding the Requests
  - This is a reported error associated with the Function for the (virtual) Bridge associated with the Port (see [Section 6.2](#)), and must be reported as an Unsupported Request. For Root Ports, the reporting of this error is optional.
  - For a Posted Request already being processed by the Transaction Layer, the Port is permitted not to report the error.

Note: This is equivalent to the case where the Request had been Transmitted before the Link status became DL\_Down

Note: The point at which a Posted Request becomes “unreportable” is implementation specific.

- Discarding all Completions submitted by the device core for Transmission

For an Upstream Port, DL\_Down status is handled as a reset by:

- Returning all PCI Express-specific registers, state machines and externally observable state to the specified default or initial conditions (except for registers defined as sticky - see [Section 7.4](#))
- Discarding all TLPs being processed
- For Switch and Bridge propagating hot reset to all associated Downstream Ports. In Switches that support Link speeds greater than 5.0 GT/s, the Upstream Port must direct the LTSSM of each Downstream Port to the Hot Reset state, but not hold the LTSSMs in that state. This permits each Downstream Port to begin Link training immediately after its hot reset completes. This behavior is recommended for all Switches.

## 2.9.2 Transaction Layer Behavior in DL\_Up Status

DL\_Up status indicates that a connection has been established with another component on the associated Link. This section specifies the Transaction Layer's behavior when the Data Link Layer reports entry to the DL\_Up status to the Transaction Layer, indicating that the Link is operational. The Transaction Layer of a Port with DL\_Up status must accept received TLPs that conform to the other rules of this specification.

For a Downstream Port on a Root Complex or a Switch:

- When transitioning from a non-DL\_Up status to a DL\_Up status and the Auto Slot Power Limit Disable bit is Clear in the Slot Control Register, the Port must initiate the transmission of a Set\_Slot\_Power\_Limit Message to the other component on the Link to convey the value programmed in the Slot Power Limit Scale and Value fields of the Slot Capabilities register. This Transmission is optional if the Slot Capabilities register has not yet been initialized.

## 2.9.3 Transaction Layer Behavior During Downstream Port Containment

During Downstream Port Containment (DPC), the LTSSM associated with the Downstream Port is directed to the Disabled state. Once it reaches the Disabled state, it remains there as long as the DPC Trigger Status bit in the [DPC Status Register](#) is Set. See [Section 6.2.10](#) for requirements on how long software must leave the Downstream Port in DPC. This section specifies the Transaction Layer's behavior once DPC has been triggered, and as long as the Downstream Port remains in DPC.

- Once DPC has been triggered, no additional (Upstream) TLPs are accepted from the Data Link Layer.
- If the condition that triggered DPC was associated with an Upstream TLP, any subsequent Upstream TLPs that were already accepted from the Data Link Layer must be discarded silently.

The Downstream Port handles (Downstream) TLPs submitted by the device core in the following manner.

- If the condition that triggered DPC was associated with a Downstream TLP, any prior Downstream TLPs are permitted to be dropped silently or transmitted before the Link goes down. Otherwise, the following rules apply.
  - For each Non-Posted Request, the Port must return a Completion and discard the Request silently. The Completer ID field must contain the value associated with the Downstream Port.
    - If the [DPC Completion Control](#) bit is Set in the [DPC Control Register](#), then Completions are generated with Unsupported Request (UR) Completion Status.

- If the DPC Completion Control bit is Clear, Completions are generated with Completer Abort (CA) Completion Status.
- The Port must terminate any PME\_Turn\_Off handshake Requests targeting the Port in such a way that the Port is considered to have acknowledged the PME\_Turn\_Off Request (see the Implementation Note in Section 5.3.3.2.1 ).
- The Port must handle Vendor Defined Message Requests as described in Section 2.2.8.6 . (e.g., silently discard Vendor Defined Type 1 Message Requests that it is not designed to receive) since the DL\_Down prevents the Request from reaching its targeted Function.
- For all other Posted Requests and Completions, the Port must silently discard the TLP.

For any outstanding Non-Posted Requests where DPC being triggered prevents their associated Completions from being returned, the following apply:

- For Root Ports that support RP Extensions for DPC, the Root Port may track certain Non-Posted Requests, and when DPC is triggered, synthesize a Completion for each tracked Request. This helps avoid Completion Timeouts that would otherwise occur as a side-effect of DPC being triggered. Each synthesized Completion must have a UR or CA Completion Status as determined by the DPC Completion Control bit. The set of Non-Posted Requests that get tracked is implementation-specific, but it is strongly recommended that all Non-Posted Requests that are generated by host processor instructions (e.g., “read”, “write”, “load”, “store”, or one that corresponds to an AtomicOp) be tracked. Other candidates for tracking include peer-to-peer Requests coming from other Root Ports and Requests coming from RCiEPs.
- Otherwise, the associated Requesters may encounter Completion Timeouts. The software solution stack should comprehend and account for this possibility.

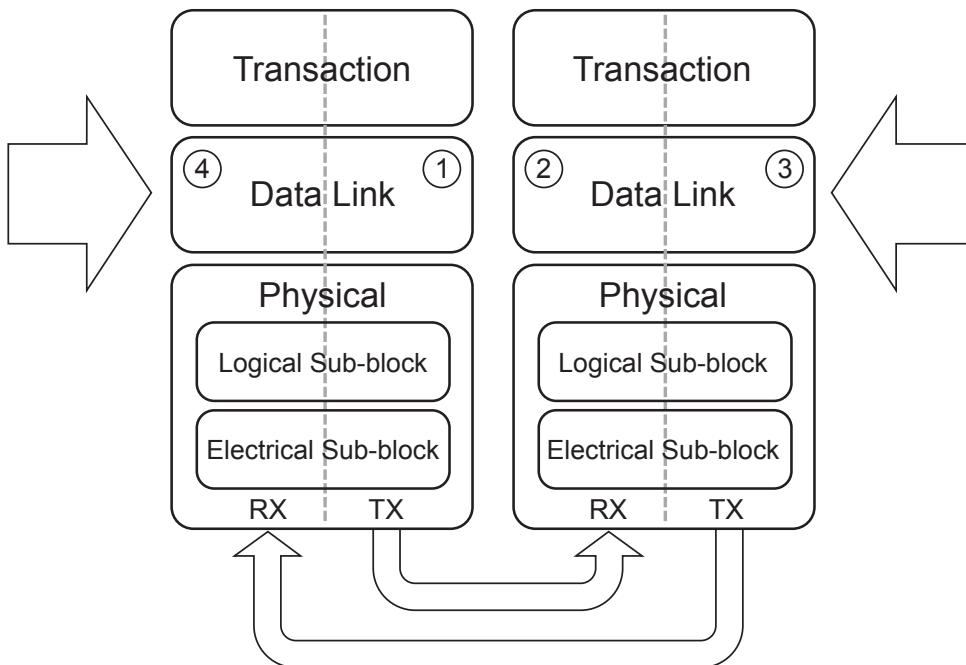


## Data Link Layer Specification

The Data Link Layer acts as an intermediate stage between the Transaction Layer and the Physical Layer. Its primary responsibility is to provide a reliable mechanism for exchanging Transaction Layer Packets (TLPs) between the two components on a Link.

3.

### 3.1 Data Link Layer Overview



OM13778A

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

The Data Link Layer is responsible for reliably conveying TLPs supplied by the Transaction Layer across a PCI Express Link to the other component's Transaction Layer. Services provided by the Data Link Layer include:

Data Exchange:

- Accept TLPs for transmission from the Transmit Transaction Layer and convey them to the Transmit Physical Layer
- Accept TLPs received over the Link from the Physical Layer and convey them to the Receive Transaction Layer

Error Detection and Retry:

- TLP Sequence Number and LCRC generation
- Transmitted TLP storage for Data Link Layer Retry

- Data integrity checking for TLPs and Data Link Layer Packets (DLLPs)
- Positive and negative acknowledgement DLLPs
- Error indications for error reporting and logging mechanisms
- Link Acknowledgement Timeout replay mechanism

Initialization and power management:

- Track Link state and convey active/reset/disconnected state to Transaction Layer

DLLPs are:

- used for Link Management functions including TLP acknowledgement, power management, and exchange of Flow Control information.
- transferred between Data Link Layers of the two directly connected components on a Link

DLLPs are sent point-to-point, between the two components on one Link. TLPs are routed from one component to another, potentially through one or more intermediate components.

Data integrity checking for DLLPs and TLPs is done using a CRC included with each packet sent across the Link. DLLPs use a 16-bit CRC and TLPs (which can be much longer than DLLPs) use a 32-bit LCRC. TLPs additionally include a sequence number, which is used to detect cases where one or more entire TLPs have been lost.

Received DLLPs that fail the CRC check are discarded. The mechanisms that use DLLPs may suffer a performance penalty from this loss of information, but are self-repairing such that a successive DLLP will supersede any information lost.

TLPs that fail the data integrity checks (LCRC and sequence number), or that are lost in transmission from one component to another, are re-sent by the Transmitter. The Transmitter stores a copy of all TLPs sent, re-sending these copies when required, and purges the copies only when it receives a positive acknowledgement of error-free receipt from the other component. If a positive acknowledgement has not been received within a specified time period, the Transmitter will automatically start re-transmission. The Receiver can request an immediate re-transmission using a negative acknowledgement.

The Data Link Layer appears as an information conduit with varying latency to the Transaction Layer. On any given individual Link all TLPs fed into the Transmit Data Link Layer (1 and 3) will appear at the output of the Receive Data Link Layer (2 and 4) in the same order at a later time, as illustrated in [Figure 3-1](#). The latency will depend on a number of factors, including pipeline latencies, width and operational frequency of the Link, transmission of electrical signals across the Link, and delays caused by Data Link Layer Retry. As a result of these delays, the Transmit Data Link Layer (1 and 3) can apply backpressure to the Transmit Transaction Layer, and the Receive Data Link Layer (2 and 4) communicates the presence or absence of valid information to the Receive Transaction Layer.

## 3.2 Data Link Control and Management State Machine

The Data Link Layer tracks the state of the Link. It communicates Link status with the Transaction and Physical Layers, and performs Link management through the Physical Layer. The Data Link Layer contains the Data Link Control and Management State Machine (DLCMSM) to perform these tasks. The states for this machine are described below, and are shown in [Figure 3-2](#).

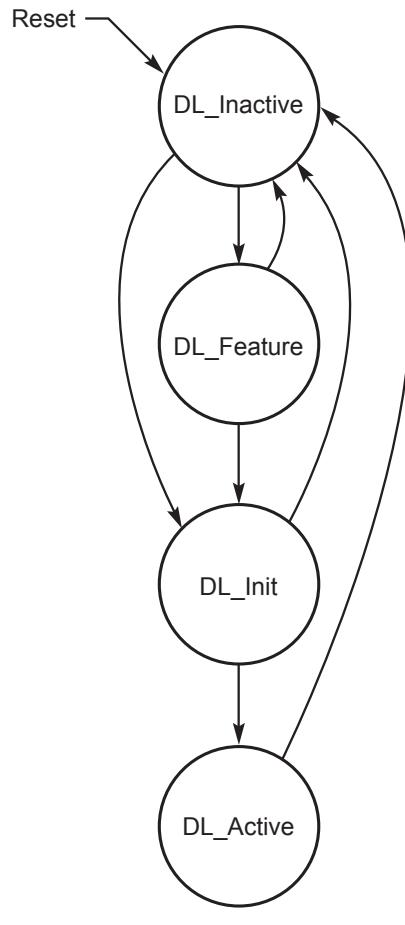
States:

- DL\_Inactive - Physical Layer reporting Link is non-operational or nothing is connected to the Port
- DL\_Feature (optional) - Physical Layer reporting Link is operational, perform the Data Link Feature Exchange

- DL\_Init - Physical Layer reporting Link is operational, initialize Flow Control for the default Virtual Channel
- DL\_Active - Normal operation mode

Status Outputs:

- ***DL\_Down*** - The Data Link Layer is not communicating with the component on the other side of the Link.
- ***DL\_Up*** - The Data Link Layer is communicating with the component on the other side of the Link.



OM13779A

Figure 3-2 Data Link Control and Management State Machine

### 3.2.1 Data Link Control and Management State Machine Rules

Rules per state:

- ***DL\_Inactive***
  - Initial state following PCI Express hot, warm, or cold reset (see [Section 6.6](#)). Note that DL states are unaffected by an FLR (see [Section 6.6](#)).
  - Upon entry to DL\_Inactive
    - Reset all Data Link Layer state information to default values

- If the Port supports the optional Data Link Feature Exchange, the Remote Data Link Feature Supported, and Remote Data Link Feature Supported Valid fields must be cleared.
- Discard the contents of the Data Link Layer Retry Buffer (see [Section 3.6](#))
- While in DL\_Inactive:
  - Report DL\_Down status to the Transaction Layer as well as to the rest of the Data Link Layer  
Note: This will cause the Transaction Layer to discard any outstanding transactions and to terminate internally any attempts to transmit a TLP. For a Downstream Port, this is equivalent to a “Hot-Remove”. For an Upstream Port, having the Link go down is equivalent to a hot reset (see [Section 2.9](#)).
  - Discard TLP information from the Transaction and Physical Layers
  - Do not generate or accept DLLPs
- Exit to DL\_Feature if:
  - The Port supports the optional Data Link Feature Exchange, the Data Link Feature Exchange Enable bit is Set, the Transaction Layer indicates that the Link is not disabled by software, and the Physical Layer reports Physical LinkUp = 1b
- Exit to DL\_Init if:
  - The Port does not support the optional Data Link Feature Exchange, the Transaction Layer indicates that the Link is not disabled by software and the Physical Layer reports Physical LinkUp = 1b  
*or*
  - The Port supports the optional Data Link Feature Exchange, the Data Link Feature Exchange Enable bit is Clear, the Transaction Layer indicates that the Link is not disabled by software, and the Physical Layer reports Physical LinkUp = 1b
- ***DL\_Feature***
  - While in DL\_Feature:
    - Perform the Data Link Feature Exchange protocol as described in [Section 3.3](#)
    - Report DL\_Down status
    - The Data Link Layer of a Port with DL\_Down status is permitted to discard any received TLPs provided that it does not acknowledge those TLPs by sending one or more Ack DLLPs
  - Exit to DL\_Init if:
    - Data Link Feature Exchange completes successfully, and the Physical Layer continues to report Physical LinkUp = 1b,  
*or*
    - Data Link Feature Exchange determines that the remote Data Link Layer does not support the optional Data Link Feature Exchange protocol, and the Physical Layer continues to report Physical LinkUp = 1b
  - Terminate the Data Link Feature Exchange protocol and exit to DL\_Inactive if:
    - Physical Layer reports Physical LinkUp = 0b
- ***DL\_Init***
  - While in DL\_Init:
    - Initialize Flow Control for the default Virtual Channel, VC0, following the Flow Control initialization protocol described in [Section 3.4](#)
    - Report DL\_Down status while in state FC\_INIT1; DL\_Up status in state FC\_INIT2

- The Data Link Layer of a Port with DL\_Down status is permitted to discard any received TLPs provided that it does not acknowledge those TLPs by sending one or more Ack DLLPs
- Exit to DL\_Active if:
  - Flow Control initialization completes successfully, and the Physical Layer continues to report Physical LinkUp = 1b
- Terminate attempt to initialize Flow Control for VC0 and exit to DL\_Inactive if:
  - Physical Layer reports Physical LinkUp = 0b
- ***DL\_Active***
  - DL\_Active is referred to as the normal operating state
  - While in DL\_Active:
    - Accept and transfer TLP information with the Transaction and Physical Layers as specified in this chapter
    - Generate and accept DLLPs as specified in this chapter
    - Report DL\_Up status to the Transaction and Data Link Layers
  - Exit to DL\_Inactive if:
    - Physical Layer reports Physical LinkUp = 0b
    - Downstream Ports that are Surprise Down Error Reporting Capable (see [Section 7.5.3.6](#)) must treat this transition from DL\_Active to DL\_Inactive as a Surprise Down error, except in the following cases where this error detection is blocked:
      - If the Secondary Bus Reset bit in the Bridge Control register has been Set by software, then the subsequent transition to DL\_Inactive must not be considered an error.
      - If the Link Disable bit has been Set by software, then the subsequent transition to DL\_Inactive must not be considered an error.
      - If a Switch Downstream Port transitions to DL\_Inactive due to an event above that Port, that transition to DL\_Inactive must not be considered an error. Example events include the Switch Upstream Port propagating Hot Reset, the Switch Upstream Link transitioning to DL\_Down, and the Secondary Bus Reset bit in the Switch Upstream Port being Set.
      - If a PME\_Turn\_Off Message has been sent through this Port, then the subsequent transition to DL\_Inactive must not be considered an error.
      - Note that the DL\_Inactive transition for this condition will not occur until a power off, a reset, or a request to restore the Link is sent to the Physical Layer.
      - Note also that in the case where the PME\_Turn\_Off/PME\_TO\_Ack handshake fails to complete successfully, a Surprise Down error may be detected.
      - If the Port is associated with a hot-pluggable slot (the Hot-Plug Capable bit in the Slot Capabilities register Set), and the Hot-Plug Surprise bit in the Slot Capabilities register is Set, then any transition to DL\_Inactive must not be considered an error.
      - If the Port is associated with a hot-pluggable slot (Hot-Plug Capable bit in the Slot Capabilities register Set), and Power Controller Control bit in Slot Control register is Set (Power-Off), then any transition to DL\_Inactive must not be considered an error.

Error blocking initiated by one or more of the above cases must remain in effect until the Port exits DL\_Active and subsequently returns to DL\_Active with none of the blocking cases in effect at the time of the return to DL\_Active.

Note that the transition out of DL\_Active is simply the expected transition as anticipated per the error detection blocking condition.

If implemented, this is a reported error associated with the detecting Port (see [Section 6.2](#) ).

## IMPLEMENTATION NOTE

### Physical Layer Throttling

Note that there are conditions where the Physical Layer may be temporarily unable to accept TLPs and DLLPs from the Data Link Layer. The Data Link Layer must comprehend this by providing mechanisms for the Physical Layer to communicate this condition, and for TLPs and DLLPs to be temporarily blocked by the condition.

## 3.3 Data Link Feature Exchange

The Data Link Feature Exchange protocol is optional. Ports that implement this protocol contain the Data Link Feature Extended Capability (see [Section 7.7.4](#) ). This capability contains four fields:

- The Local Data Link Feature Supported field indicates the Data Link Features supported by the local Port
- The Remote Data Link Feature Supported field indicates the Data Link Features supported by the remote Port
- The Remote Data Link Feature Supported Valid bit indicates that the Remote Data Link Feature Supported field contains valid data
- The Data Link Feature Exchange Enable field permits systems to disable the Data Link Feature Exchange. This can be used to work around legacy hardware that does not correctly ignore the DLLP.

The Data Link Feature Exchange protocol transmits a Port's Local Feature Supported information to the Remote Port and captures that Remote Port's Feature Supported information.

Rules for this protocol are:

- On entry to DL\_Feature:
  - The Remote Data Link Feature Supported and Remote Data Link Feature Supported Valid fields must be Cleared
- While in DL\_Feature:
  - Transaction Layer must block transmission of TLPs
  - Transmit the Data Link Feature DLLP
    - The transmitted Feature Supported field must equal the Local Data Link Feature Supported field.
    - The transmitted Feature Ack bit must equal the Remote Data Link Feature Supported Valid bit.
  - The Data Link Feature DLLP must be transmitted at least once every 34 µs. Time spent in the Recovery or Configuration LTSSM states does not contribute to this limit.
  - Process received Data Link Feature DLLPs:

- If the Remote Data Link Feature Supported Valid bit is Clear, record the Feature Supported field from the received Data Link Feature DLLP in the Remote Data Link Feature Supported field and Set the Remote Data Link Feature Supported Valid bit.
- Exit DL\_Feature if:
  - An InitFC1 DLLP has been received.
  - An MR-IOV MRIInit DLLP (encoding 0000 0001b) has been received.
  - or
  - While in DL\_Feature, at least one Data Link Feature DLLP has been received with the Feature Ack bit Set.

Each Data Link Feature has an associated bit in the Feature Supported field. A Data Link Feature is activated when that bit is Set in both the Local Data Link Feature Supported and Remote Data Link Feature Supported fields.

Data Link Features and their corresponding bit locations are shown in Table 3-1.

*Table 3-1 Data Link Feature Supported Bit Definition*

Bit Location	Description
0	<u>Scaled Flow Control</u> - indicates support for Scaled Flow Control. Scaled Flow Control must be supported in Ports that support 16.0 GT/s or higher operation.
22:1	Reserved

## 3.4 Flow Control Initialization Protocol

Before starting normal operation following power-up or interconnect reset, it is necessary to initialize Flow Control for the default Virtual Channel, VC0 (see Section 6.6). In addition, when additional Virtual Channels (VCs) are enabled, the Flow Control initialization process must be completed for each newly enabled VC before it can be used (see Section 2.6.1). This section describes the initialization process that is used for all VCs. Note that since VC0 is enabled before all other VCs, no TLP traffic of any kind will be active prior to initialization of VC0. However, when additional VCs are being initialized there will typically be TLP traffic flowing on other, already enabled, VCs. Such traffic has no direct effect on the initialization process for the additional VC(s).

There are two states in the VC initialization process. These states are:

- FC\_INIT1
- FC\_INIT2

The rules for this process are given in the following section.

### 3.4.1 Flow Control Initialization State Machine Rules

- If at any time during initialization for VCs 1-7 the VC is disabled, the flow control initialization process for the VC is terminated
- Rules for state FC\_INIT1:
  - Entered when initialization of a VC (VCx) is required
    - When the DL\_Init state is entered (VCx = VC0)

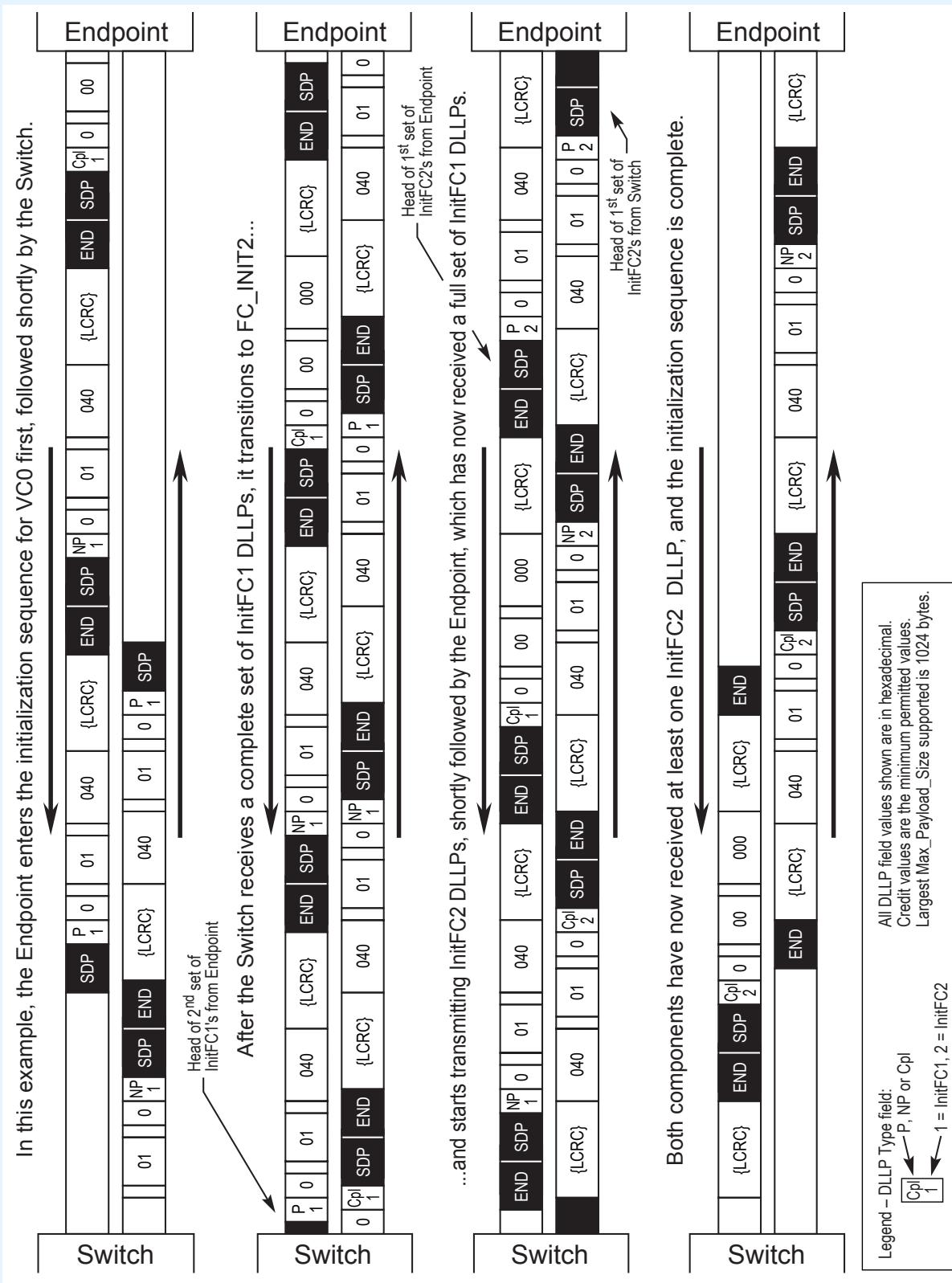
- When a VC (VCx = VC1-7) is enabled by software (see [Section 7.9.1](#) and [Section 7.9.2](#))
  - While in FC\_INIT1:
    - Transaction Layer must block transmission of TLPs using VCx
    - Transmit the following three InitFC1 DLLPs for VCx in the following relative order:
      - InitFC1-P (first)
      - InitFC1-NP (second)
      - InitFC1-Cpl (third)
    - The three InitFC1 DLLPs must be transmitted at least once every 34 µs.
      - Time spent in the Recovery or Configuration LTSSM states does not contribute to this limit.
      - It is strongly encouraged that the InitFC1 DLLP transmissions are repeated frequently, particularly when there are no other TLPs or DLLPs available for transmission.
    - If Scaled Flow Control is activated on the Link, set the HdrScale and DataScale fields in the InitFC1 DLLPs to 01b, 10b, or 11b to indicate the scaling factor it is using on the corresponding HdrFC and DataFC values.
    - If the Transmitter does not support Scaled Flow Control or if Scaled Flow Control is not activated on the Link, set the HdrScale and DataScale fields to 00b.
    - Except as needed to ensure at least the required frequency of InitFC1 DLLP transmission, the Data Link Layer must not block other transmissions.
      - Note that this includes all Physical Layer initiated transmissions (for example, Ordered Sets), Ack and Nak DLLPs (when applicable), and TLPs using VCs that have previously completed initialization (when applicable)
  - Process received InitFC1 and InitFC2 DLLPs:
    - Record the indicated HdrFC and DataFC values
    - If the Receiver supports Scaled Flow Control, record the indicated HdrScale and DataScale values.
    - Set flag FI1 once FC unit values have been recorded for each of P, NP, and Cpl for VCx
  - Exit to FC\_INIT2 if:
    - Flag FI1 has been Set indicating that FC unit values have been recorded for each of P, NP, and Cpl for VCx
- Rules for state FC\_INIT2:
  - While in FC\_INIT2:
    - Transaction Layer must block transmission of TLPs using VCx
    - Transmit the following three InitFC2 DLLPs for VCx in the following relative order:
      - InitFC2-P (first)
      - InitFC2-NP (second)
      - InitFC2-Cpl (third)
    - The three InitFC2 DLLPs must be transmitted at least once every 34 µs.
      - Time spent in the Recovery or Configuration LTSSM states does not contribute to this limit.

- It is strongly encouraged that the InitFC2 DLLP transmissions are repeated frequently, particularly when there are no other TLPs or DLLPs available for transmission.
- If Scaled Flow Control is activated on the Link, set the HdrScale and DataScale fields in the InitF2 DLLPs to 01b, 10b, or 11b to indicate the scaling factor it is using on the corresponding HdrFC and DataFC values.
- If the Transmitter does not support Scaled Flow Control or if Scaled Flow Control is not activated on the Link, set the HdrScale and DataScale fields to 00b.
- Except as needed to ensure at least the required frequency of InitFC2 DLLP transmission, the Data Link Layer must not block other transmissions
  - Note that this includes all Physical Layer initiated transmissions (for example, Ordered Sets), Ack and Nak DLLPs (when applicable), and TLPs using VCs that have previously completed initialization (when applicable)
- Process received InitFC1 and InitFC2 DLLPs:
  - Ignore the received HdrFC, HdrScale, DataFC, and DataScale values
  - Set flag FI2 on receipt of any InitFC2 DLLP for VCx
  - Set flag FI2 on receipt of any TLP on VCx, or any UpdateFC DLLP for VCx
- Signal completion and exit if:
  - Flag FI2 has been Set
  - If Scaled Flow Control is activated on the Link, the Transmitter must send 01b, 10b, or 11b for HdrScale and DataScale in all UpdateFC DLLPs for VCx.
  - If the Scaled Flow Control is not supported or if Scaled Flow Control is not activated on the Link, the Transmitter must send 00b for HdrScale and DataScale in all UpdateFC DLLPs for VCx.

## IMPLEMENTATION NOTE

### Example of Flow Control Initialization

Figure 3-3 illustrates an example of the Flow Control initialization protocol for VC0 between a Switch and a Downstream component. In this example, each component advertises the minimum permitted values for each type of Flow Control credit. For both components the largest Max\_Payload\_Size value supported is 1024 bytes, corresponding to a data payload credit advertisement of 040h. All DLLPs are shown as received without error.



OM14548

*Figure 3-3 VC0 Flow Control Initialization Example with 8b/10b Encoding-based Framing*

### 3.4.2 Scaled Flow Control

Link performance can be affected when there are insufficient flow control credits available to account for the Link round trip time. This effect becomes more noticeable at higher Link speeds and the limitation of 127 header credits and 2047 data credits can limit performance. The Scaled Flow Control mechanism is designed to address this limitation.

All Ports are permitted to support Scaled Flow Control. Ports that support 16.0 GT/s and higher data rates must support Scaled Flow Control. Scaled Flow Control activation does not affect the ability to operate at 16.0 GT/s and higher data rates.

The following rules apply when Scaled Flow Control is not activated for the Link:

- The InitFC1, InitFC2, and UpdateFC DLLPs must contain 00b in the HdrScale and DataScale fields.
- The HdrFC counter is 8 bits wide and the HdrFC DLLP field includes all bits of the counter.
- The DataFC counter is 12 bits wide and the DataFC DLLP field includes all bits of the counter.

The following rules apply when Scaled Flow Control is activated for the Link:

- The InitFC1 and InitFC2 DLLPs must contain 01b, 10b, or 11b in the HdrScale field. The value is determined by the maximum number of header credits that will be outstanding of the indicated credit type as defined in [Table 3-2](#).
- The InitFC1 and InitFC2 DLLPs must contain 01b, 10b, or 11b in the DataScale field. The value is determined by the maximum number of data payload credits that will be outstanding of the indicated credit type as defined in [Table 3-2](#).
- If the received HdrScale and DataScale values recorded in state FC\_INIT1 were non-zero, then Scaled Flow Control is enabled on this VC and UpdateFC DLLPs must contain 01b, 10b, or 11b in the HdrScale and DataScale fields.
- If the received HdrScale and DataScale values recorded in state FC\_INIT1 were zero, then Scaled Flow Control is not enabled on this VC and UpdateFC DLLPs must contain 00b in the HdrScale and DataScale fields.

*Table 3-2 Scaled Flow Control Scaling Factors*

Scale Factor	Scaled Flow Control Supported	Credit Type	Min Credits	Max Credits	Field Width	FC DLLP field	
						Transmitted	Received
00b	No	Hdr	1	127	8 bits	HdrFC	HdrFC
		Data	1	2,047	12 bits	DataFC	DataFC
01b	Yes	Hdr	1	127	8 bits	HdrFC	HdrFC
		Data	1	2,047	12 bits	DataFC	DataFC
10b	Yes	Hdr	4	508	10 bits	HdrFC >> 2	HdrFC << 2

Scale Factor	Scaled Flow Control Supported	Credit Type	Min Credits	Max Credits	Field Width	FC DLLP field	
						Transmitted	Received
11b	Yes	Data	4	8,188	14 bits	DataFC >> 2	DataFC << 2
		Hdr	16	2,032	12 bits	HdrFC >> 4	HdrFC << 4
		Data	16	32,752	16 bits	DataFC >> 4	DataFC << 4

## 3.5 Data Link Layer Packets (DLLPs)

The following DLLPs are used to support Link operations:

- Ack DLLP: TLP Sequence Number acknowledgement; used to indicate successful receipt of some number of TLPs
- Nak DLLP: TLP Sequence Number negative acknowledgement; used to initiate a Data Link Layer Retry
- InitFC1, InitFC2, and UpdateFC DLLPs; used for Flow Control
- DLLPs used for Power Management

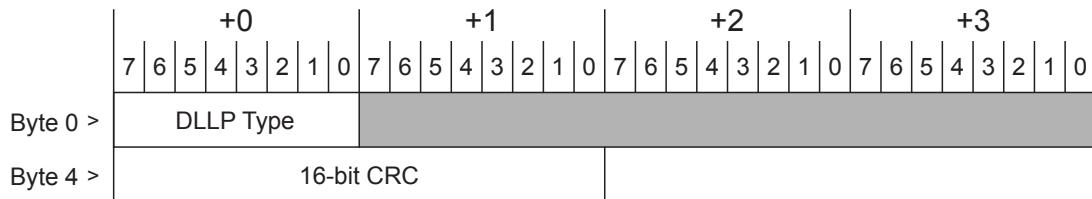
### 3.5.1 Data Link Layer Packet Rules

All DLLP fields marked Reserved (sometimes abbreviated as R) must be filled with all 0's when a DLLP is formed. Values in such fields must be ignored by Receivers. The handling of Reserved values in encoded fields is specified for each case.

All DLLPs include the following fields:

- DLLP Type - Specifies the type of DLLP. The defined encodings are shown in [Table 3-3](#).
- 16-bit CRC

See [Figure 3-4](#) below.



OM14303A

*Figure 3-4 DLLP Type and CRC Fields*

*Table 3-3 DLLP Type Encodings*

Encodings (b)	DLLP Type
0000 0000	Ack
0000 0001	MRInit - See the MR-IOV Specification <sup>49</sup>

Encodings (b)	DLLP Type
0000 0010	Data_Link_Feature
0001 0000	Nak
0010 0000	<b>PM_Enter_L1</b>
0010 0001	<b>PM_Enter_L23</b>
0010 0011	<b>PM_Active_State_Request_L1</b>
0010 0100	<b>PM_Request_Ack</b>
0011 0000	Vendor-specific
0011 0001	NOP
0100 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	InitFC1-P (v[2:0] specifies Virtual Channel)
0101 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	InitFC1-NP
0110 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	InitFC1-Cpl
0111 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	MRInitFC1 (v[2:0] specifies Virtual Link) - See the MR-IOV Specification <sup>50</sup>
1100 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	InitFC2-P
1101 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	InitFC2-NP
1110 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	InitFC2-Cpl
1111 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	MRInitFC2 - See the MR-IOV Specification <sup>51</sup>
1000 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	UpdateFC-P
1001 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	UpdateFC-NP
1010 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	UpdateFC-Cpl
1011 0v <sub>2</sub> v <sub>1</sub> v <sub>0</sub>	MRUpdateFC - See the MR-IOV Specification <sup>52</sup>
All other encodings	Reserved

- For Ack and Nak DLLPs (see [Figure 3-5](#)):
  - The AckNak\_Seq\_Num field is used to indicate what TLPs are affected
  - Transmission and reception is handled by the Data Link Layer according to the rules provided in [Section 3.6](#).
- For InitFC1, InitFC2, and UpdateFC DLLPs:
  - The HdrFC field contains the credit value for headers of the indicated type (P, NP, or Cpl).
  - The DataFC field contains the credit value for payload Data of the indicated type (P, NP, or Cpl).

49. The MR-IOV protocol uses this encoding for the MRInit negotiation. The MR-IOV protocol assumes that non-MR-IOV components will silently ignore these DLLPs..

50. The MR-IOV protocol uses these encodings after the successful completion of MRInit negotiation.

51. The MR-IOV protocol uses these encodings after the successful completion of MRInit negotiation.

52. The MR-IOV protocol uses these encodings after the successful completion of MRInit negotiation.

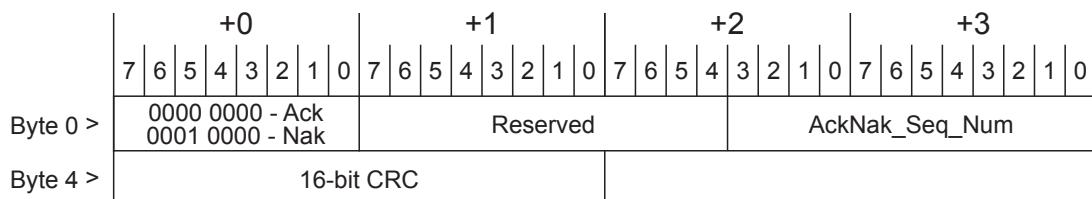
- The HdrScale field contains the scaling factor for headers of the indicated type. Encodings are defined in [Table 3-4](#).
  - The DataScale field contains the scaling factor for payload data of the indicated type. Encodings are defined in [Table 3-4](#).
  - If Scaled Flow Control is activated, the HdrScale and DataScale fields must be set to 01b, 10b, or 11b in all InitFC1, InitFC2, and UpdateFC DLLPs transmitted.
  - In UpdateFCs, a Transmitter is only permitted to send non-zero values in the HdrScale and DataScale fields if it supports Scaled Flow Control and it received non-zero values for HdrScale and DataScale in the InitFC1s and InitFC2s it received for this VC.
  - The packet formats are shown in [Figure 3-7](#), [Figure 3-8](#), and [Figure 3-9](#).
  - Transmission is triggered by the Data Link Layer when initializing Flow Control for a Virtual Channel (see [Section 3.4](#)), and following Flow Control initialization by the Transaction Layer according to the rules in [Section 2.6](#).
  - Checked for integrity on reception by the Data Link Layer and if correct, the information content of the DLLP is passed to the Transaction Layer. If the check fails, the information is discarded.
- Note: InitFC1 and InitFC2 DLLPs are used only for VC initialization

*Table 3-4 HdrScale and DataScale Encodings*

HdrScale or DataScale Value	Scaled Flow Control Supported	Scaling Factor	HdrFC DLLP Field	DataFC DLLP Field
00b	No	1	HdrFC[7:0]	DataFC[11:0]
01b	Yes	1	HdrFC[7:0]	DataFC[11:0]
10b	Yes	4	HdrFC[9:2]	DataFC[13:2]
11b	Yes	16	HdrFC[11:4]	DataFC[15:4]

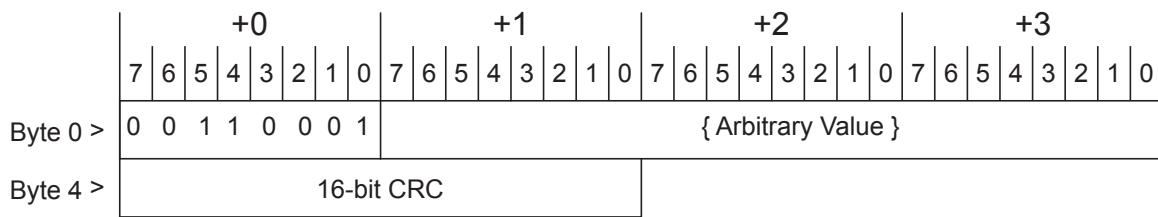
- For Power Management (PM) DLLPs (see [Figure 3-10](#)):
  - Transmission is triggered by the component's power management logic according to the rules in [Chapter 5](#)
  - Checked for integrity on reception by the Data Link Layer, then passed to the component's power management logic
- For Vendor-specific DLLPs (see [Figure 3-11](#))
  - It is recommended that receivers silently ignore Vendor Specific DLLPs unless enabled by implementation specific mechanisms.
  - It is recommended that transmitters not send Vendor Specific DLLPs unless enabled by implementation specific mechanisms.
- For NOP DLLPs (see [Figure 3-6](#))
  - Receivers shall discard this DLLP without action after checking it for data integrity.<sup>53</sup>
- For Data Link Feature DLLPs (see [Figure 3-12](#))
  - The Feature Ack bit is set to indicate that the transmitting Port has received a [Data Link Feature DLLP](#).
  - The Feature Supported bits indicate the features supported by the transmitting Port. These bits equal the value of the Local Data Link Feature Supported field (see [Section 7.7.4.2](#)).

53. This is a special case of the more general rule for unsupported DLLP Type encodings (see [Section 3.6.2.2](#)).



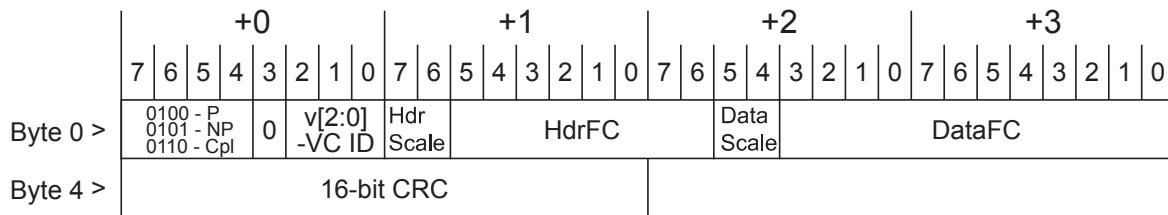
OM13781A

Figure 3-5 Data Link Layer Packet Format for Ack and Nak



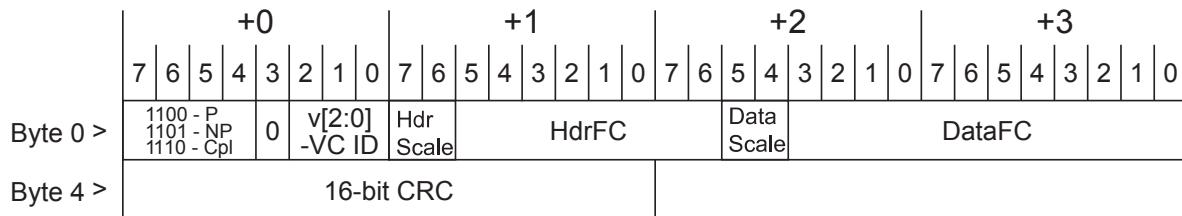
NOPDataLinkPktFmt

Figure 3-6 NOP Data Link Layer Packet Format



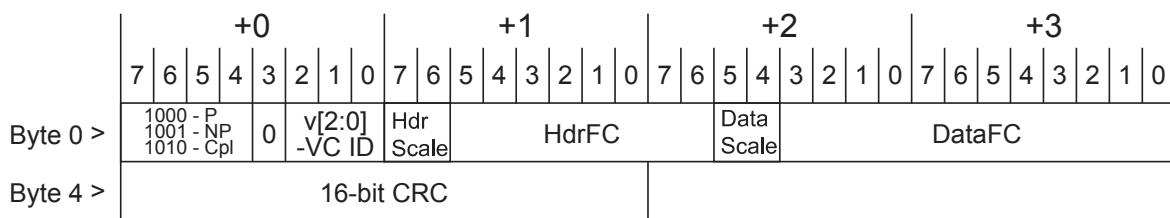
OM13782B

Figure 3-7 Data Link Layer Packet Format for InitFC1



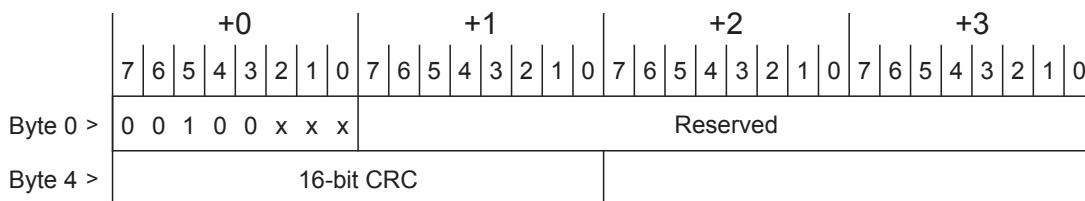
OM13783B

Figure 3-8 Data Link Layer Packet Format for InitFC2



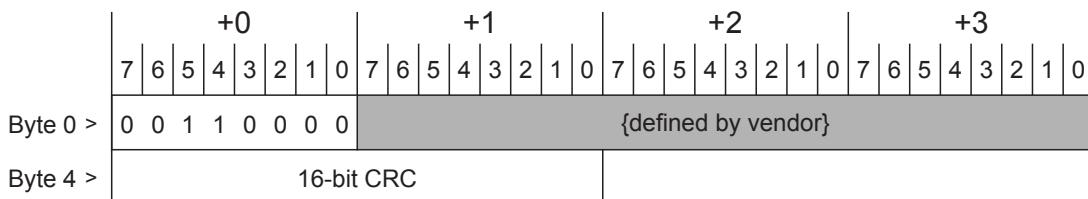
OM13784B

Figure 3-9 Data Link Layer Packet Format for UpdateFC



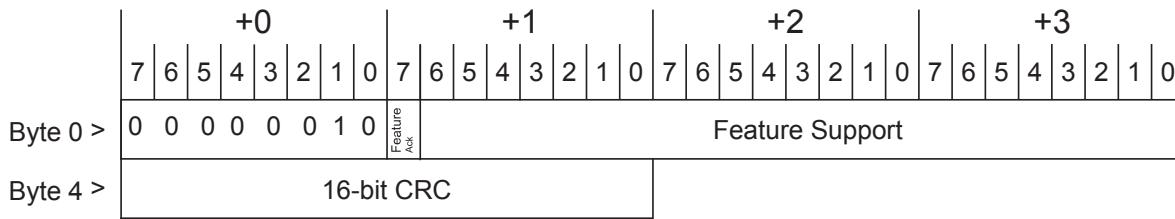
OM14304A

Figure 3-10 PM Data Link Layer Packet Format



OM14305A

Figure 3-11 Vendor-specific Data Link Layer Packet Format



DataLinkFeatureDLLP

Figure 3-12 Data Link Feature DLLP Format

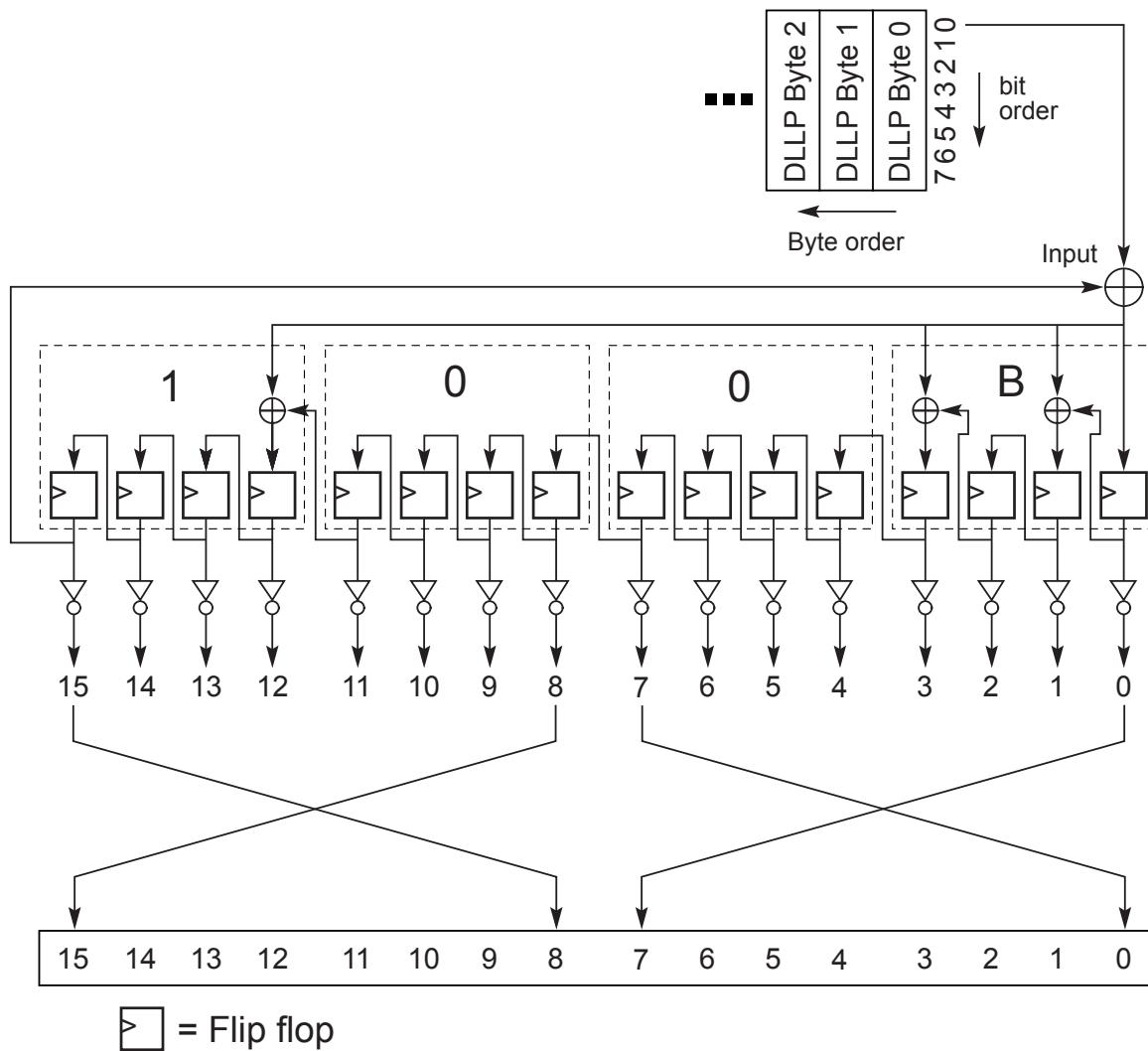
The following are the characteristics and rules associated with Data Link Layer Packets (DLLPs):

- DLLPs are differentiated from TLPs when they are presented to, or received from, the Physical Layer.

- DLLP data integrity is protected using a 16-bit CRC
- The CRC value is calculated using the following rules (see [Figure 3-13](#)):
  - The polynomial used for CRC calculation has a coefficient expressed as 100Bh
  - The seed value (initial value for CRC storage registers) is FFFFh
  - CRC calculation starts with bit 0 of byte 0 and proceeds from bit 0 to bit 7 of each byte
  - Note that CRC calculation uses all bits of the DLLP, regardless of field type, including Reserved fields.
  - The result of the calculation is complemented, then placed into the 16-bit CRC field of the DLLP as shown in [Table 3-5](#).

*Table 3-5 Mapping of Bits into CRC Field*

CRC Result Bit	Corresponding Bit Position in the 16-Bit CRC Field
0	7
1	6
2	5
3	4
4	3
5	2
6	1
7	0
8	15
9	14
10	13
11	12
12	11
13	10
14	9
15	8



OM13785

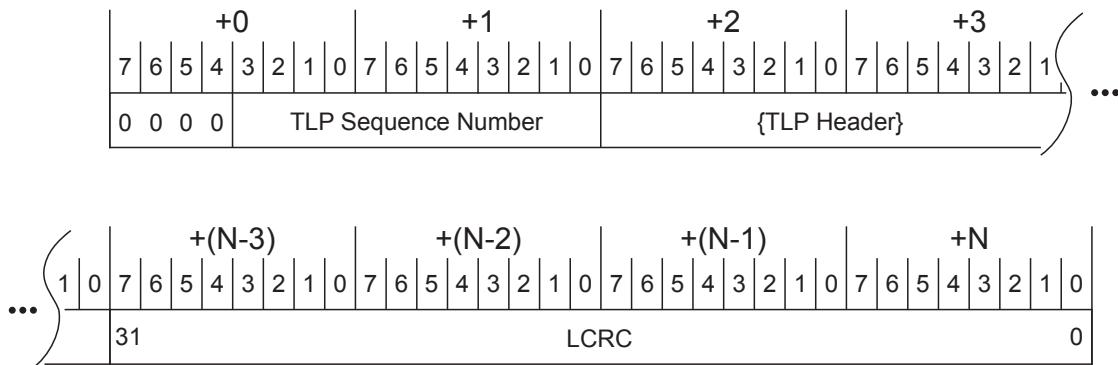
Figure 3-13 Diagram of CRC Calculation for DLLPs

## 3.6 Data Integrity Mechanisms

### 3.6.1 Introduction

The Transaction Layer provides TLP boundary information to the Data Link Layer. This allows the Data Link Layer to apply a TLP Sequence Number and a Link CRC (LCRC) for error detection to the TLP. The Receive Data Link Layer validates received TLPs by checking the TLP Sequence Number, LCRC code and any error indications from the Receive Physical Layer. In case any of these errors are in a TLP, Data Link Layer Retry is used for recovery.

The format of a TLP with the TLP Sequence Number and LCRC code applied is shown in Figure 3-14 .



OM13786A

*Figure 3-14 TLP with LCRC and TLP Sequence Number Applied*

On Ports that support Protocol Multiplexing, packets containing a non-zero value in Symbol +0, bits 7:4 are PMUX Packets. For TLPs, these bits must be 0000b. See [Appendix G](#) for details.

On Ports that do not support Protocol Multiplexing, Symbol +0, bits 7:4 are Reserved.

### 3.6.2 LCRC, Sequence Number, and Retry Management (TLP Transmitter)

The TLP transmission path through the Data Link Layer (paths labeled 1 and 3 in [Figure 3-1](#)) prepares each TLP for transmission by applying a sequence number, then calculating and appending a Link CRC (LCRC), which is used to ensure the integrity of TLPs during transmission across a Link from one component to another. TLPs are stored in a retry buffer, and are re-sent unless a positive acknowledgement of receipt is received from the other component. If repeated attempts to transmit a TLP are unsuccessful, the Transmitter will determine that the Link is not operating correctly, and will instruct the Physical Layer to retrain the Link (via the LTSSM Recovery state, [Section 4.2.6](#)). If Link retraining fails, the Physical Layer will indicate that the Link is no longer up, causing the DLCMSM to move to the DL\_Inactive state.

The mechanisms used to determine the TLP LCRC and the Sequence Number and to support Data Link Layer Retry are described in terms of conceptual “counters” and “flags”. This description does not imply nor require a particular implementation and is used only to clarify the requirements.

#### 3.6.2.1 LCRC and Sequence Number Rules (TLP Transmitter)

The following counters and timer are used to explain the remaining rules in this section:

- The following 12-bit counters are used:
  - NEXT\_TRANSMIT\_SEQ - Stores the packet sequence number applied to TLPs
    - Set to 000h in DL\_Inactive state
  - ACKD\_SEQ - Stores the sequence number acknowledged in the most recently received Ack or Nak DLLP.
    - Set to FFFh in DL\_Inactive state
- The following 2-bit counter is used:
  - REPLAY\_NUM - Counts the number of times the Retry Buffer has been re-transmitted

- Set to 00b in DL\_Inactive state
- The following timer is used:
  - REPLAY\_TIMER - Counts time that determines when a replay is required, according to the following rules:
    - Started at the last Symbol of any TLP transmission or retransmission, if not already running
    - For each replay, reset and restart REPLAY\_TIMER when sending the last Symbol of the first TLP to be retransmitted
    - Resets and restarts for each Ack DLLP received while there are more unacknowledged TLPs outstanding, if, and only if, the received Ack DLLP acknowledges some TLP in the retry buffer.
      - Note: This ensures that REPLAY\_TIMER is reset only when forward progress is being made
    - Reset and hold until restart conditions are met for each Nak received (except during a replay) or when the REPLAY\_TIMER expires
    - Not advanced during Link retraining (holds its value when the LTSSM is in the Recovery or Configuration state). Refer to [Section 4.2.5.3](#) and [Section 4.2.5.4](#).
    - If Protocol Multiplexing is supported, optionally not advanced during the reception of PMUX Packets (see [Appendix G](#)).
    - Resets and holds when there are no outstanding unacknowledged TLPs

The following rules describe how a TLP is prepared for transmission before being passed to the Physical Layer:

- The Transaction Layer indicates the start and end of the TLP to the Data Link Layer while transferring the TLP
  - The Data Link Layer treats the TLP as a “black box” and does not process or modify the contents of the TLP
- Each TLP is assigned a 12-bit sequence number when it is accepted from the Transmit side of the Transaction Layer
  - Upon acceptance of the TLP from the Transaction Layer, the packet sequence number is applied to the TLP by:
    - prepending the 12-bit value in NEXT\_TRANSMIT\_SEQ to the TLP
    - prepending 4 Reserved bits to the TLP, preceding the sequence number (see [Figure 3-15](#))
  - If the equation:

$$(NEXT\_TRANSMIT\_SEQ - ACKD\_SEQ) \bmod 4096 \geq 2048$$

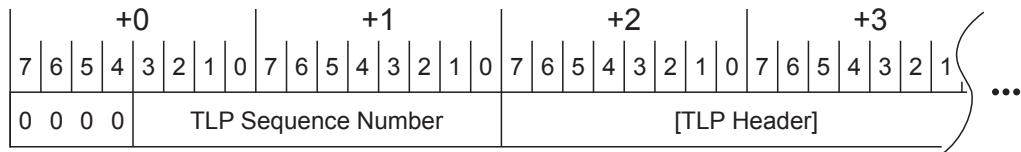
*Equation 3-1 Tx SEQ Stall*

is true, the Transmitter must cease accepting TLPs from the Transaction Layer until the equation is no longer true

- Following the application of NEXT\_TRANSMIT\_SEQ to a TLP accepted from the Transmit side of the Transaction Layer, NEXT\_TRANSMIT\_SEQ is incremented (except in the case where the TLP is nullified):

$$\text{NEXT\_TRANSMIT\_SEQ} := (\text{NEXT\_TRANSMIT\_SEQ} + 1) \bmod 4096$$

Equation 3-2 Tx SEQ Update



OM13787A

Figure 3-15 TLP Following Application of TLP Sequence Number and Reserved Bits

- TLP data integrity is protected during transfer between Data Link Layers using a 32-bit LCRC
- The LCRC value is calculated using the following mechanism (see Figure 3-16):
  - The polynomial used has coefficients expressed as 04C1 1DB7h
  - The seed value (initial value for LCRC storage registers) is FFFF FFFFh
  - The LCRC is calculated using the TLP following sequence number application (see Figure 3-15)
  - LCRC calculation starts with bit 0 of byte 0 (bit 8 of the TLP sequence number) and proceeds from bit 0 to bit 7 of each successive byte.
  - Note that LCRC calculation uses all bits of the TLP, regardless of field type, including Reserved fields
  - The remainder of the LCRC calculation is complemented, and the complemented result bits are mapped into the 32-bit LCRC field as shown in Table 3-6 .

Table 3-6 Mapping of Bits into LCRC Field

LCRC Result Bit	Corresponding Bit Position in the 32-Bit LCRC Field
0	7
1	6
2	5
3	4
4	3
5	2
6	1
7	0
8	15
9	14
10	13
11	12

LCRC Result Bit	Corresponding Bit Position in the 32-Bit LCRC Field
12	11
13	10
14	9
15	8
16	23
17	22
18	21
19	20
20	19
21	18
22	17
23	16
24	31
25	30
26	29
27	28
28	27
29	26
30	25
31	24

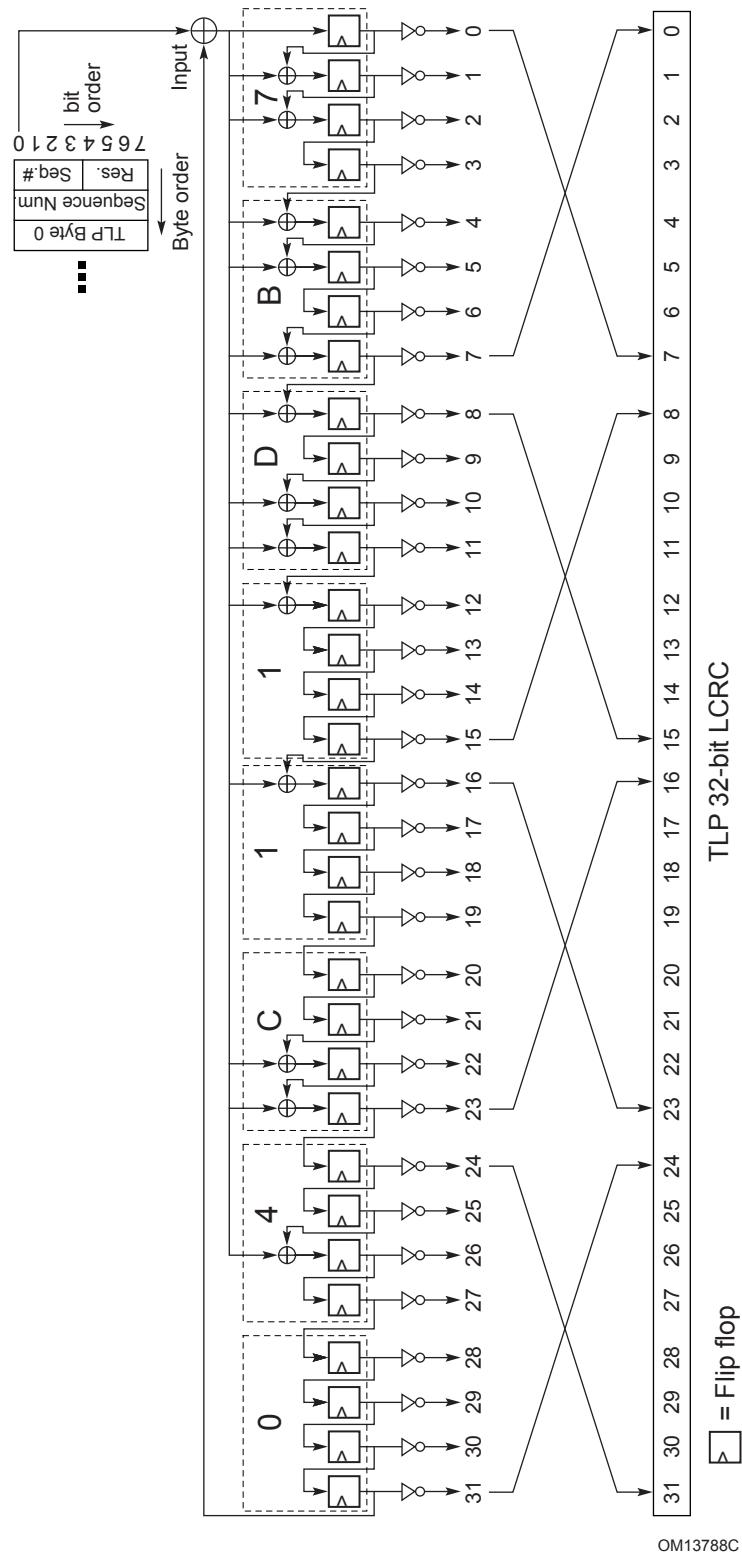


Figure 3-16 Calculation of LCRC

The 32-bit LCRC field is appended to the TLP following the bytes received from the Transaction Layer (see [Figure 3-14](#) ).

To support cut-through routing of TLPs, a Transmitter is permitted to modify a transmitted TLP to indicate that the Receiver must ignore that TLP (“nullify” the TLP).

- A Transmitter is permitted to nullify a TLP being transmitted. To do this in a way that will robustly prevent misinterpretation or corruption, the Transmitter must do the following:
  - Transmit all DWs of the TLP when the Physical Layer is using 128b/130b encoding (see [Section 4.2.2.3.1](#))
  - Use the remainder of the calculated LCRC value without inversion (the logical inverse of the value normally used)
  - Indicate to the Transmit Physical Layer that the TLP is nullified
- When this is done, the Transmitter does not increment NEXT\_TRANSMIT\_SEQ

The following rules describe the operation of the Data Link Layer Retry Buffer, from which TLPs are re-transmitted when necessary:

- Copies of Transmitted TLPs must be stored in the Data Link Layer Retry Buffer, except for nullified TLPs.

When a replay is initiated, either due to reception of a Nak or due to REPLAY\_TIMER expiration, the following rules describe the sequence of operations that must be followed:

- If all TLPs transmitted have been acknowledged (the Retry Buffer is empty), terminate replay, otherwise continue.
- Increment REPLAY\_NUM. When the replay is initiated by the reception of a Nak that acknowledged some TLPs in the retry buffer, REPLAY\_NUM is reset. It is then permitted (but not required) to be incremented.
  - If REPLAY\_NUM rolls over from 11b to 00b, the Transmitter signals the Physical Layer to retrain the Link, and waits for the completion of retraining before proceeding with the replay. This is a reported error associated with the Port (see [Section 6.2](#) ).

Note that Data Link Layer state, including the contents of the Retry Buffer, are not reset by this action unless the Physical Layer reports Physical LinkUp = 0b (causing the Data Link Control and Management State Machine to transition to the DL\_Inactive state).

- If REPLAY\_NUM does not roll over from 11b to 00b, continue with the replay.
- Block acceptance of new TLPs from the Transmit Transaction Layer.
- Complete transmission of any TLP currently being transmitted.
- Retransmit unacknowledged TLPs, starting with the oldest unacknowledged TLP and continuing in original transmission order
  - Reset and restart REPLAY\_TIMER when sending the last Symbol of the first TLP to be retransmitted
  - Once all unacknowledged TLPs have been re-transmitted, return to normal operation.
  - If any Ack or Nak DLLPs are received during a replay, the Transmitter is permitted to complete the replay without regard to the Ack or Nak DLLP(s), or to skip retransmission of any newly acknowledged TLPs.
    - Once the Transmitter has started to resend a TLP, it must complete transmission of that TLP in all cases.
  - Ack and Nak DLLPs received during a replay must be processed, and may be collapsed
    - Example: If multiple Acks are received, only the one specifying the latest Sequence Number value must be considered - Acks specifying earlier Sequence Number values are effectively “collapsed” into this one

- Example: During a replay, Nak is received, followed by an Ack specifying a later Sequence Number - the Ack supersedes the Nak, and the Nak is ignored  
Note: Since all entries in the Retry Buffer have already been allocated space in the Receiver by the Transmitter's Flow Control gating logic, no further flow control synchronization is necessary.
- Re-enable acceptance of new TLPs from the Transmit Transaction Layer.

A replay can be initiated by the expiration of REPLAY\_TIMER, or by the receipt of a Nak. The following rule covers the expiration of REPLAY\_TIMER:

- If the Transmit Retry Buffer contains TLPs for which no Ack or Nak DLLP has been received, and (as indicated by REPLAY\_TIMER) no Ack or Nak DLLP has been received for a period exceeding the REPLAY\_TIMER Limit, the Transmitter initiates a replay.
  - Simplified REPLAY\_TIMER Limits are:
    - A value from 24,000 to 31,000 Symbol Times when the Extended Synch bit is Clear.
    - A value from 80,000 to 100,000 Symbol Times when the Extended Synch bit is Set.
    - If the Extended Synch bit changes state while unacknowledged TLPs are outstanding, implementations are permitted to adjust their REPLAY\_TIMER Limit when the Extended Synch bit changes state or the next time the REPLAY\_TIMER is reset.
  - Implementations that support 16.0 GT/s or higher data rates must use the Simplified REPLAY\_TIMER Limits for operation at all data rates.
  - Implementations that only support data rates less than 16.0 GT/s are strongly recommended to use the Simplified REPLAY\_TIMER Limits for operation at all data rates, but they are permitted to use the REPLAY\_TIMER Limits described in the [PCIe-3.1].

This is a Replay Timer Timeout error and it is a reported error associated with the Port (see [Section 6.2](#) ).

## IMPLEMENTATION NOTE

### Determining REPLAY\_TIMER Limit Values

Replays are initiated primarily with a Nak DLLP, and the REPLAY\_TIMER serves as a secondary mechanism. Since it is a secondary mechanism, the REPLAY\_TIMER Limit has a relatively small effect on the average time required to convey a TLP across a Link. The Simplified REPLAY\_TIMER Limits have been defined so that no adjustments are required for ASPM L0s, Retimers, or other items as in previous revisions of this specification.

TLP Transmitters and compliance tests must base replay timing as measured at the Port of the TLP Transmitter. Timing starts with either the last Symbol of a transmitted TLP, or else the last Symbol of a received Ack DLLP, whichever determines the oldest unacknowledged TLP. Timing ends with the First Symbol of TLP retransmission.

When measuring replay timing to the point when TLP retransmission begins, compliance tests must allow for any other TLP or DLLP transmission already in progress in that direction (thus preventing the TLP retransmission).

## IMPLEMENTATION NOTE

### Recommended Priority of Scheduled Transmissions

When multiple DLLPs of the same type are scheduled for transmission but have not yet been transmitted, it is possible in many cases to “collapse” them into a single DLLP. For example, if a scheduled Ack DLLP transmission is stalled waiting for another transmission to complete, and during this time another Ack is scheduled for transmission, it is only necessary to transmit the second Ack, since the information it provides will supersede the information in the first Ack.

In addition to any TLP from the Transaction Layer (or the Retry Buffer, if a replay is in progress), Multiple DLLPs of different types may be scheduled for transmission at the same time, and must be prioritized for transmission. The following list shows the preferred priority order for selecting information for transmission. Note that the priority of the NOP DLLP and the Vendor-Specific DLLP is not listed, as usage of these DLLPs is completely implementation specific, and there is no recommended priority. Note that this priority order is a guideline, and that in all cases a fairness mechanism is highly recommended to ensure that no type of traffic is blocked for an extended or indefinite period of time by any other type of traffic. Note that the Ack Latency Limit value and REPLAY\_TIMER Limit specify requirements measured at the Port of the component, and the internal arbitration policy of the component must ensure that these externally measured requirements are met.

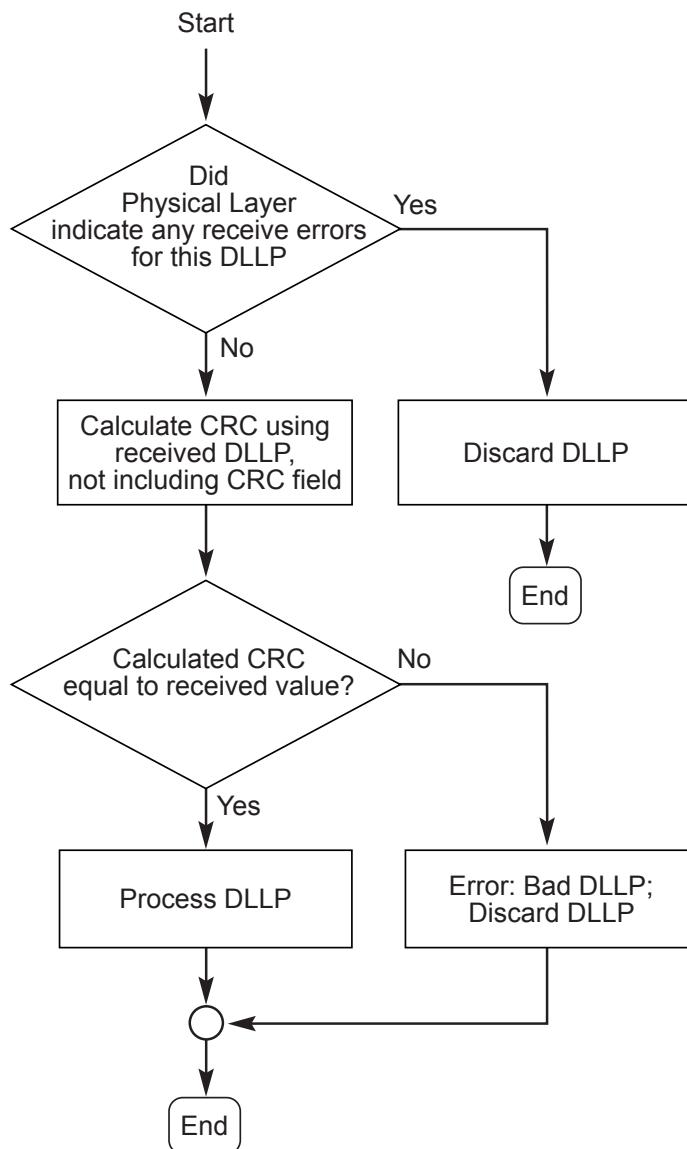
1. Completion of any transmission (TLP or DLLP) currently in progress (highest priority)
2. Nak DLLP transmissions
3. Ack DLLP transmissions scheduled for transmission as soon as possible due to: receipt of a duplicate TLP -OR- expiration of the Ack latency timer (see [Section 3.6.3.1](#))
4. FC DLLP transmissions required to satisfy [Section 2.6](#)
5. Retry Buffer re-transmissions
6. TLPs from the Transaction Layer
7. FC DLLP transmissions other than those required to satisfy [Section 2.6](#)
8. All other DLLP transmissions (lowest priority)

#### **3.6.2.2 Handling of Received DLLPs**

Since Ack/Nak and Flow Control DLLPs affect TLPs flowing in the opposite direction across the Link, the TLP transmission mechanisms in the Data Link Layer are also responsible for Ack/Nak and Flow Control DLLPs received from the other component on the Link. These DLLPs are processed according to the following rules (see [Figure 3-17](#)):

- If the Physical Layer indicates a Receiver Error, discard any DLLP currently being received and free any storage allocated for the DLLP. Note that reporting such errors to software is done by the Physical Layer (and, therefore, are not reported by the Data Link Layer).
- For all received DLLPs, the CRC value is checked by:
  - Applying the same algorithm used for calculation of transmitted DLLPs to the received DLLP, not including the 16-bit CRC field of the received DLLP
  - Comparing the calculated result with the value in the CRC field of the received DLLP
    - If not equal, the DLLP is corrupt

- A corrupt received DLLP is discarded. This is a Bad DLLP error and is a reported error associated with the Port (see [Section 6.2](#) ).
- A received DLLP that is not corrupt, but that uses unsupported DLLP Type encodings is discarded without further action. This is not considered an error.
- Non-zero values in Reserved fields are ignored.
- Receivers must process all DLLPs received at the rate they are received



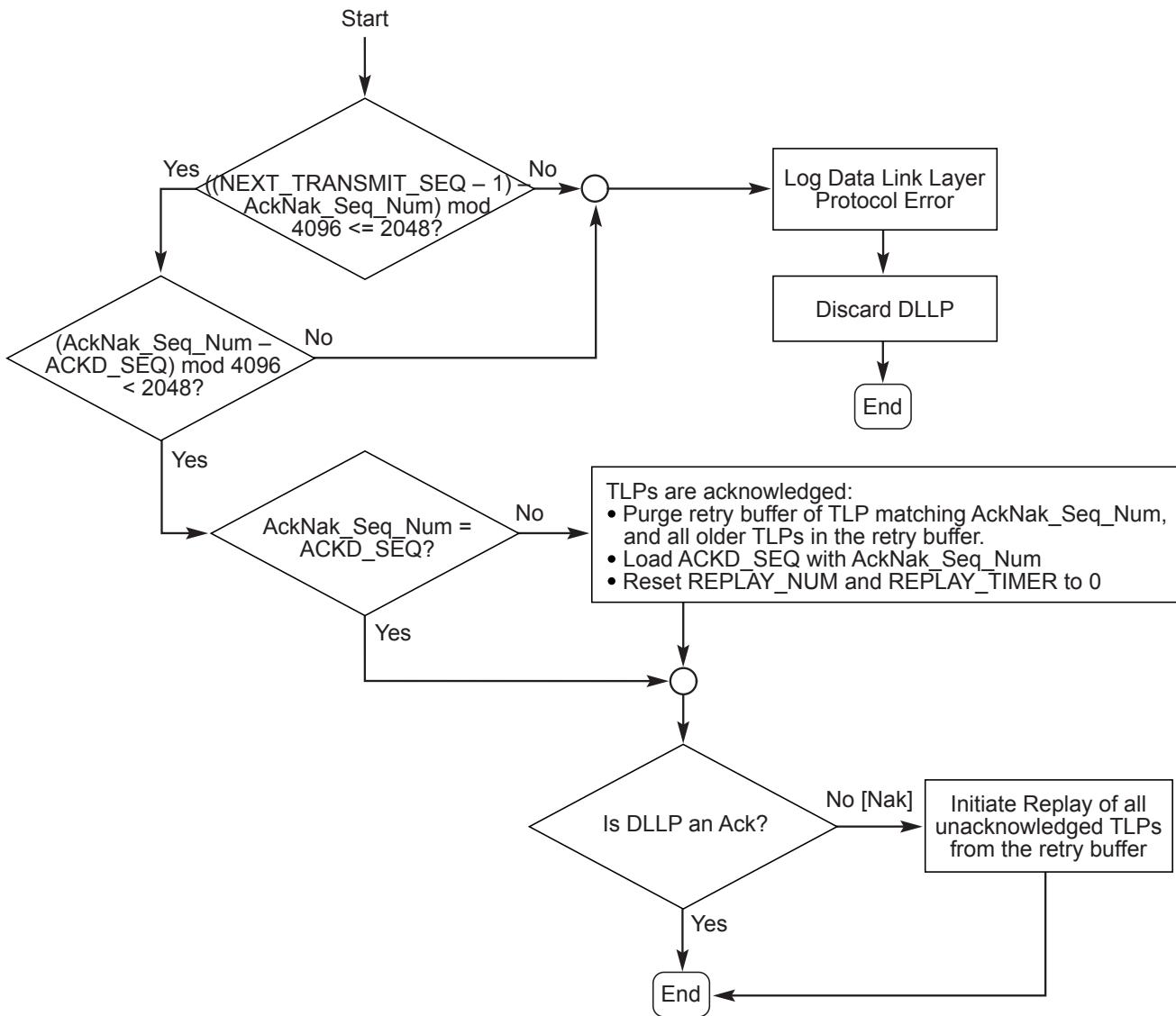
OM13789A

*Figure 3-17 Received DLLP Error Check Flowchart*

- Received NOP DLLPs are discarded
- Received FC DLLPs are passed to the Transaction Layer
- Received PM DLLPs are passed to the component's power management control logic

- For Ack and Nak DLLPs, the following steps are followed (see [Figure 3-18](#)):
  - If the Sequence Number specified by the AckNak\_Seq\_Num does not correspond to an unacknowledged TLP, or to the value in ACKD\_SEQ, the DLLP is discarded
    - This is a Data Link Protocol Error, which is a reported error associated with the Port (see [Section 6.2](#)).
  - Note that it is not an error to receive an Ack DLLP when there are no outstanding unacknowledged TLPs, including the time between reset and the first TLP transmission, as long as the specified Sequence Number matches the value in ACKD\_SEQ.
  - If the AckNak\_Seq\_Num does not specify the Sequence Number of the most recently acknowledged TLP, then the DLLP acknowledges some TLPs in the retry buffer:
    - Purge from the retry buffer all TLPs from the oldest to the one corresponding to the AckNak\_Seq\_Num
    - Load ACKD\_SEQ with the value in the AckNak\_Seq\_Num field
    - Reset REPLAY\_NUM and REPLAY\_TIMER
  - If the DLLP is a Nak, initiate a replay (see above)

Note: Receipt of a Nak is not a reported error.



OM13790B

Figure 3-18 Ack/Nak DLLP Processing Flowchart

The following rules describe the operation of the Data Link Layer Retry Buffer, from which TLPs are re-transmitted when necessary:

- Copies of Transmitted TLPs must be stored in the Data Link Layer Retry Buffer

### 3.6.3 LCRC and Sequence Number (TLP Receiver)

The TLP Receive path through the Data Link Layer (paths labeled 2 and 4 in Figure 3-1) processes TLPs received by the Physical Layer by checking the LCRC and sequence number, passing the TLP to the Receive Transaction Layer if OK and requesting a replay if corrupted.

The mechanisms used to check the TLP LCRC and the Sequence Number and to support Data Link Layer Retry are described in terms of conceptual “counters” and “flags”. This description does not imply or require a particular implementation and is used only to clarify the requirements.

### **3.6.3.1 LCRC and Sequence Number Rules (TLP Receiver)**

The following counter, flag, and timer are used to explain the remaining rules in this section:

- The following 12-bit counter is used:
  - NEXT\_RCV\_SEQ - Stores the expected Sequence Number for the next TLP
    - Set to 000h in DL\_Inactive state
- The following flag is used:
  - NAK\_SCHEDULED
    - Cleared when in DL\_Inactive state
- The following timer is used:
  - AckNak\_LATENCY\_TIMER - Counts time that determines when an Ack DLLP becomes scheduled for transmission, according to the following rules:
    - Set to 0 in DL\_Inactive state
    - Restart from 0 each time an Ack or Nak DLLP is scheduled for transmission; Reset to 0 when all TLPs received have been acknowledged with an Ack DLLP
    - If there are initially no unacknowledged TLPs and a TLP is then received, the AckNak\_LATENCY\_TIMER starts counting only when the TLP has been forwarded to the Receive Transaction Layer

The following rules are applied in sequence to describe how received TLPs are processed, and what events trigger the transmission of Ack and Nak DLLPs (see [Figure 3-19](#) ):

- If the Physical Layer indicates a Receiver Error, discard any TLP currently being received and free any storage allocated for the TLP. Note that reporting such errors to software is done by the Physical Layer (and so are not reported by the Data Link Layer).
  - If a TLP was being received at the time the Receiver Error was indicated and the NAK\_SCHEDULED flag is clear,
    - Schedule a Nak DLLP for transmission immediately
    - Set the NAK\_SCHEDULED flag
- If the Physical Layer reports that the received TLP was nullified, and the LCRC is the logical NOT of the calculated value, discard the TLP and free any storage allocated for the TLP. This is not considered an error.
- If TLP was nullified but the LCRC does not match the logical NOT of the calculated value, the TLP is corrupt - discard the TLP and free any storage allocated for the TLP.
  - If the NAK\_SCHEDULED flag is clear,
    - Schedule a Nak DLLP for transmission immediately
    - Set the NAK\_SCHEDULED flag
  - This is a Bad TLP error and is a reported error associated with the Port (see [Section 6.2](#) ).
- The LCRC value is checked by:
  - Applying the same algorithm used for calculation (above) to the received TLP, not including the 32-bit LCRC field of the received TLP
  - Comparing the calculated result with the value in the LCRC field of the received TLP

- if not equal, the TLP is corrupt - discard the TLP and free any storage allocated for the TLP
  - If the NAK\_SCHEDULED flag is clear,
    - schedule a Nak DLLP for transmission immediately
    - set the NAK\_SCHEDULED flag

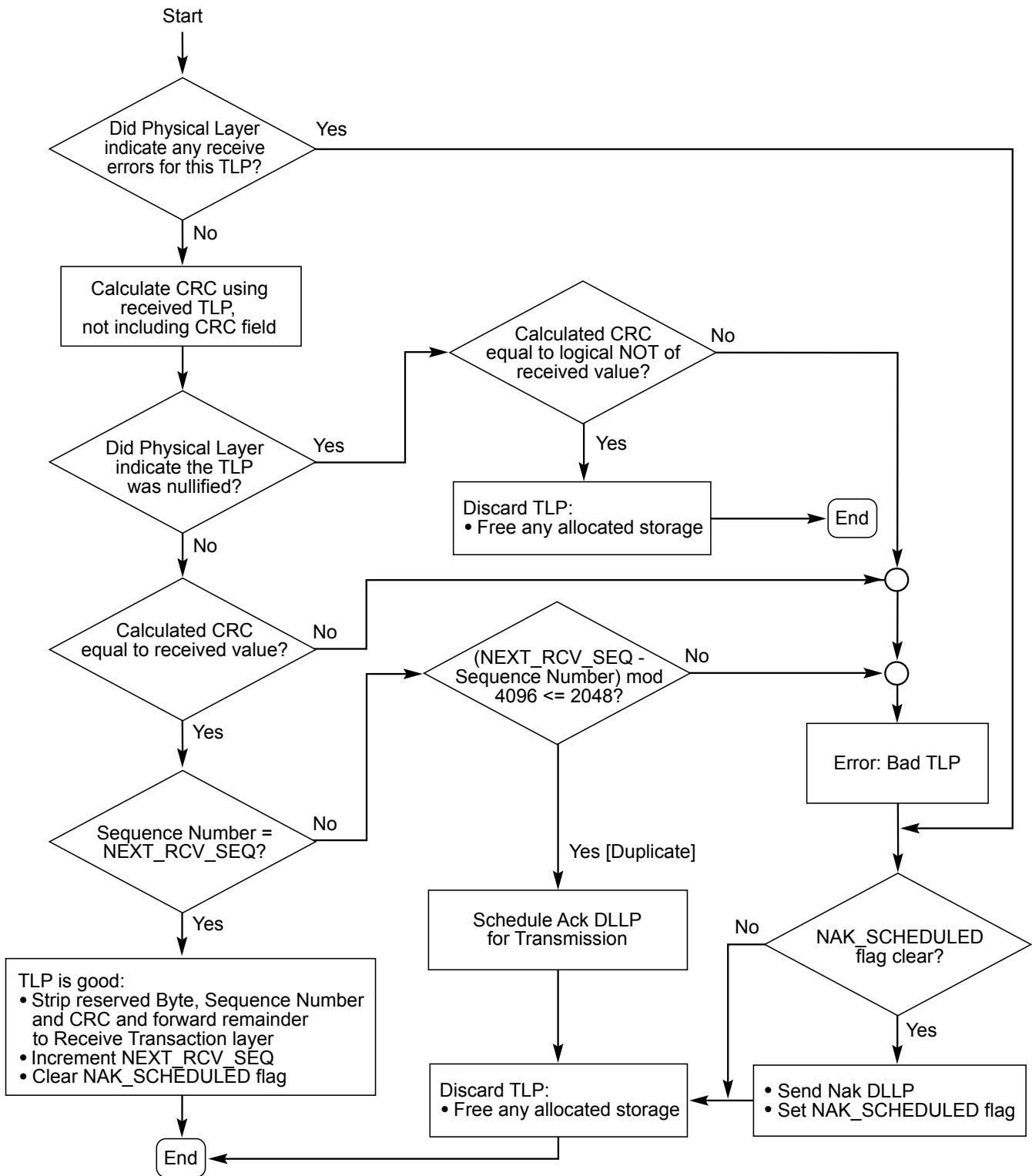
This is a Bad TLP error and is a reported error associated with the Port (see [Section 6.2](#) ).

- If the TLP Sequence Number is not equal to the expected value, stored in NEXT\_RCV\_SEQ:
  - Discard the TLP and free any storage allocated for the TLP
  - If the TLP Sequence Number satisfies the following equation:  
 $(\text{NEXT\_RCV\_SEQ} - \text{TLP Sequence Number}) \bmod 4096 \leq 2048$   
 the TLP is a duplicate, and an Ack DLLP is scheduled for transmission (per transmission priority rules)
  - Otherwise, the TLP is out of sequence (indicating one or more lost TLPs):
    - if the NAK\_SCHEDULED flag is clear,
      - schedule a Nak DLLP for transmission immediately
      - set the NAK\_SCHEDULED flag

This is a Bad TLP error and is a reported error associated with the Port (see [Section 6.2](#) ).

Regardless of the state of the NAK\_SCHEDULED flag, it is permitted for this to be a reported error associated with the Port (see [Section 6.2](#) ), and this permitted behavior is illustrated in [Figure 3-17](#). However, in order to prevent error pollution it is recommended that the Port only report such an error when the NAK\_SCHEDULED flag is clear.

- If the TLP Sequence Number is equal to the expected value stored in NEXT\_RCV\_SEQ:
  - The four Reserved bits, TLP Sequence Number, and LCRC (see [Figure 3-14](#) ) are removed and the remainder of the TLP is forwarded to the Receive Transaction Layer
    - The Data Link Layer indicates the start and end of the TLP to the Transaction Layer while transferring the TLP
      - The Data Link Layer treats the TLP as a “black box” and does not process or modify the contents of the TLP
    - Note that the Receiver Flow Control mechanisms do not account for any received TLPs until the TLP(s) are forwarded to the Receive Transaction Layer
  - NEXT\_RCV\_SEQ is incremented
  - If Set, the NAK\_SCHEDULED flag is cleared



OM13791B

Figure 3-19 Receive Data Link Layer Handling of TLPs

- A TLP Receiver must schedule an Ack DLLP such that it will be transmitted no later than when all of the following conditions are true:
  - The Data Link Control and Management State Machine is in the DL\_Active state
  - TLPs have been forwarded to the Receive Transaction Layer, but not yet acknowledged by sending an Ack DLLP
  - The AckNak\_LATENCY\_TIMER reaches or exceeds the value specified in [Table 3-7](#) for 2.5 GT/s mode operation, [Table 3-8](#) for 5.0 GT/s mode operation, [Table 3-9](#) for 8.0 GT/s and higher mode operation
  - The Link used for Ack DLLP transmission is already in L0 or has transitioned to L0
  - Note: if not already in L0, the Link must transition to L0 in order to transmit the Ack DLLP
  - Another TLP or DLLP is not currently being transmitted on the Link used for Ack DLLP transmission
  - The NAK\_SCHEDULED flag is clear
  - Note: The AckNak\_LATENCY\_TIMER must be restarted from 0 each time an Ack or Nak DLLP is scheduled for transmission
- Data Link Layer Ack DLLPs may be scheduled for transmission more frequently than required
- Data Link Layer Ack and Nak DLLPs specify the value (NEXT\_RCV\_SEQ - 1) in the AckNak\_Seq\_Num field

[Table 3-7](#), [Table 3-8](#), and [Table 3-9](#) define the threshold values for the AckNak\_LATENCY\_TIMER, which for any specific case is called the Ack Latency Limit.

TLP Receivers and compliance tests must base Ack Latency timing as measured at the Port of the TLP Receiver, starting with the time the last Symbol of a TLP is received to the first Symbol of the Ack DLLP being transmitted.

When measuring until the Ack DLLP is transmitted, compliance tests must allow for any TLP or other DLLP transmission already in progress in that direction (thus preventing the Ack DLLP transmission). If L0s is enabled, compliance tests must allow for the L0s exit latency of the Link in the direction that the Ack DLLP is being transmitted. If the Extended Synch bit of the Link Control register is Set, compliance tests must also allow for its effect on L0s exit latency.

TLP Receivers are not required to adjust their Ack DLLP scheduling based upon L0s exit latency or the value of the Extended Synch bit.

*Table 3-7 Maximum Ack Latency Limits for 2.5 GT/s (Symbol Times)*

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

*Table 3-8 Maximum Ack Latency Limits for 5.0 GT/s (Symbol Times)*

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

*Table 3-9 Maximum Ack Latency Limits for 8.0 GT/s and higher data rates (Symbol Times)*

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

## IMPLEMENTATION NOTE

### Retry Buffer Sizing

The Retry Buffer should be large enough to ensure that under normal operating conditions, transmission is never throttled because the retry buffer is full. In determining the optimal buffer size, one must consider the Ack Latency value, Ack delay caused by the Receiver already transmitting another TLP, the delays caused by the physical Link interconnect, and the time required to process the received Ack DLLP.

Given two components A and B, the L0s exit latency required by A's Receiver should be accounted for when sizing A's transmit retry buffer, as is demonstrated in the following example:

- A exits L0s on its Transmit path to B and starts transmitting a long burst of write Requests to B
- B initiates L0s exit on its Transmit path to A, but the L0s exit time required by A's Receiver is large
- Meanwhile, B is unable to send Ack DLLPs to A, and A stalls due to lack of Retry Buffer space
- The Transmit path from B to A returns to L0, B transmits an Ack DLLP to A, and the stall is resolved

This stall can be avoided by matching the size of a component's Transmitter Retry Buffer to the L0s exit latency required by the component's Receiver, or, conversely, by matching the Receiver L0s exit latency to the desired size of the Retry Buffer.

Ack Latency Limit values were chosen to allow implementations to achieve good performance without requiring an uneconomically large retry buffer. To enable consistent performance across a general purpose interconnect with differing implementations and applications, it is necessary to set the same requirements for all components without regard to the application space of any specific component. If a component does not require the full transmission bandwidth of the Link, it may reduce the size of its retry buffer below the minimum size required to maintain available retry buffer space with the Ack Latency Limit values specified.

Note that the Ack Latency Limit values specified ensure that the range of permitted outstanding TLP Sequence Numbers will never be the limiting factor causing transmission stalls.

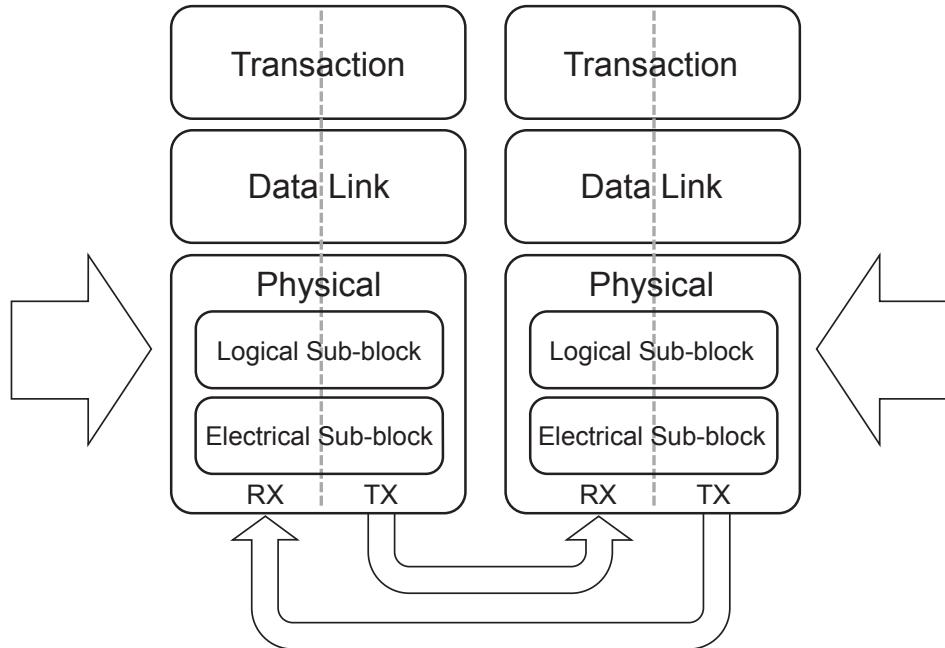
Retimers add latency (see Section 4.3.8 ) and operating in SRIS can add latency. Implementations are strongly encouraged to consider these effects when determining the optimal buffer size.

# Physical Layer Logical Block

4.

## 4.1 Introduction

The Physical Layer isolates the Transaction and Data Link Layers from the signaling technology used for Link data interchange. The Physical Layer is divided into the logical and electrical sub-blocks (see [Figure 4-1](#)).



OM13792A

*Figure 4-1 Layering Diagram Highlighting Physical Layer*

[Chapter 4](#) describes the logical sub-block and [Chapter 8](#) describes the electrical sub-block.<sup>54</sup>

## 4.2 Logical Sub-block

The logical sub-block has two main sections: a Transmit section that prepares outgoing information passed from the Data Link Layer for transmission by the electrical sub-block, and a Receiver section that identifies and prepares received information before passing it to the Data Link Layer.

The logical sub-block and electrical sub-block coordinate the state of each Transceiver through a status and control register interface or functional equivalent. The logical sub-block directs control and management functions of the Physical Layer.

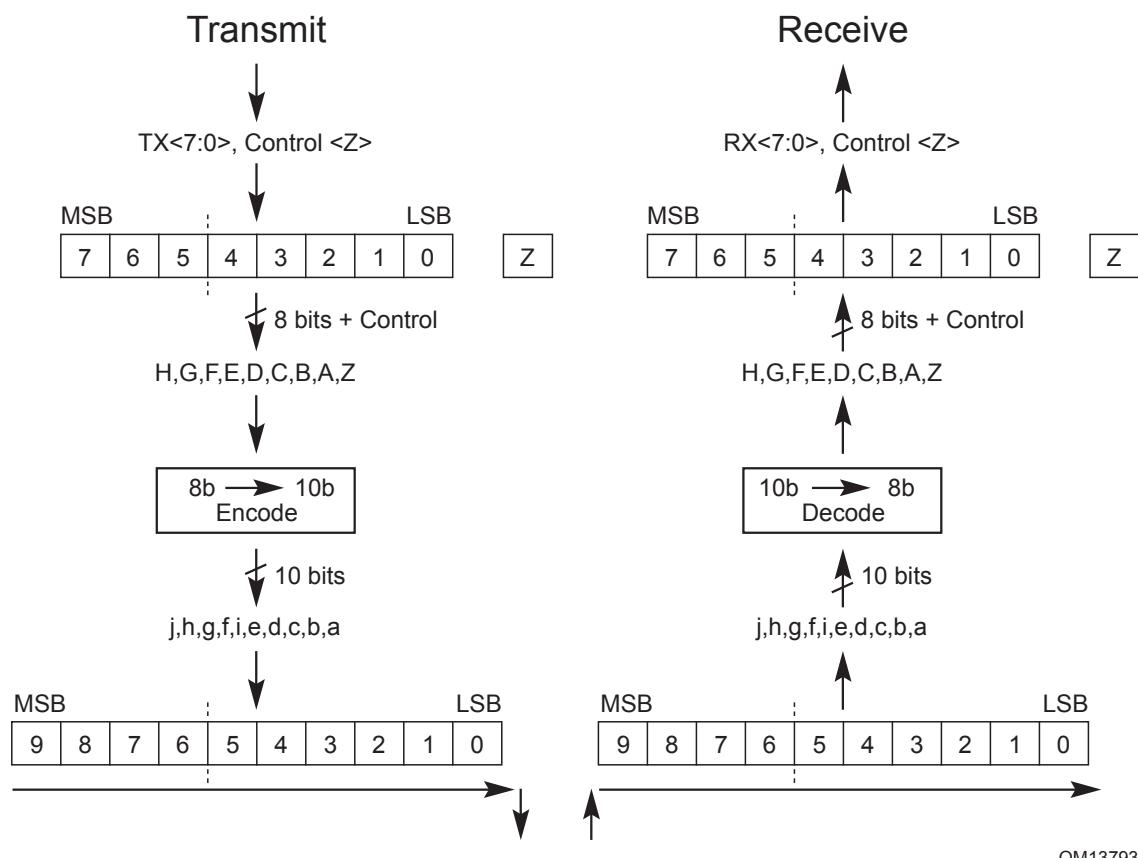
54. Prior to [PCIe-4.0] Chapter 4 described both logical and electrical sub-blocks. With [PCIe-4.0], section 4.3 was moved to a new [Chapter 8](#) and a new section 4.3 was added containing Retimer information.

PCI Express uses 8b/10b encoding when the data rate is 2.5 GT/s or 5.0 GT/s. For data rates greater than or equal to 8.0 GT/s, it uses a per-Lane code along with physical layer encapsulation.

## 4.2.1 Encoding for 2.5 GT/s and 5.0 GT/s Data Rates

### 4.2.1.1 Symbol Encoding

At 2.5 and 5.0 GT/s, PCI Express uses an 8b/10b transmission code. The definition of this transmission code is identical to that specified in ANSI X3.230-1994, clause 11 (and also IEEE 802.3z, 36.2.4). Using this scheme, 8-bit data characters are treated as 3 bits and 5 bits mapped onto a 4-bit code group and a 6-bit code group, respectively. The control bit in conjunction with the data character is used to identify when to encode one of the 12 Special Symbols included in the 8b/10b transmission code. These code groups are concatenated to form a 10-bit Symbol. As shown in [Figure 4-2](#), ABCDE maps to abcdei and FGH maps to fghj.

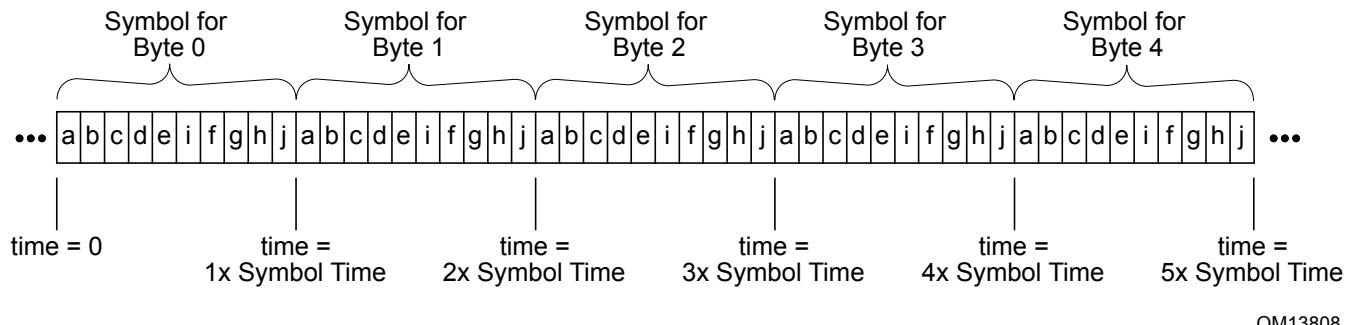


OM13793

[Figure 4-2 Character to Symbol Mapping](#)

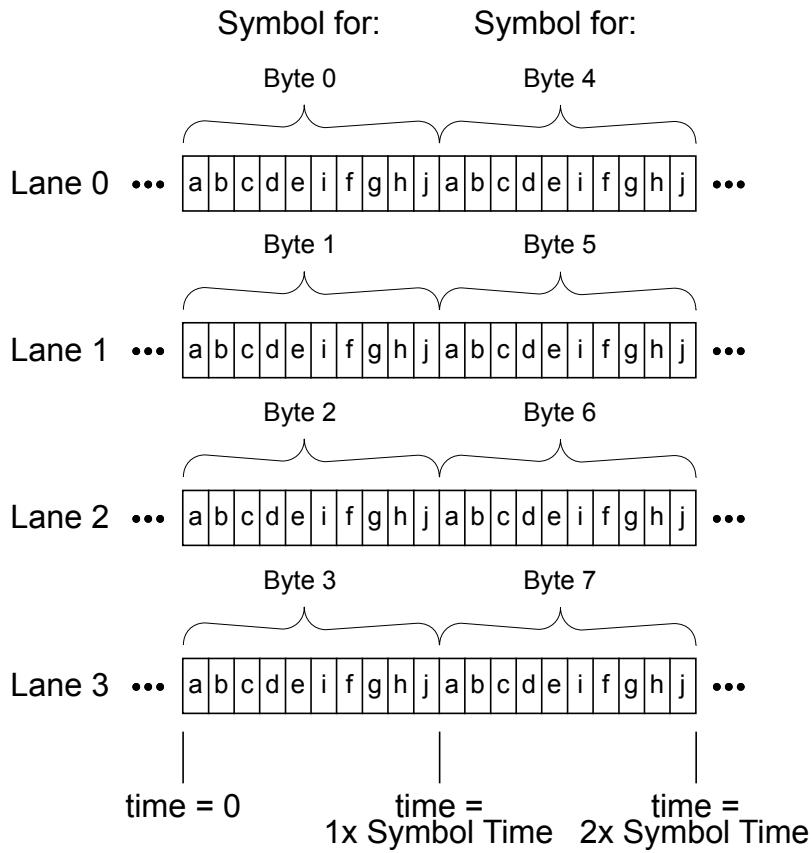
### 4.2.1.1.1 Serialization and De-serialization of Data

The bits of a Symbol are placed on a Lane starting with bit “a” and ending with bit “j”. Examples are shown in [Figure 4-3](#) and [Figure 4-4](#).



OM13808

Figure 4-3 Bit Transmission Order on Physical Lanes - x1 Example



OM13809

Figure 4-4 Bit Transmission Order on Physical Lanes - x4 Example

#### 4.2.1.1.2 Special Symbols for Framing and Link Management (K Codes)

The 8b/10b encoding scheme provides Special Symbols that are distinct from the Data Symbols used to represent Characters. These Special Symbols are used for various Link Management mechanisms described later in this chapter. Special Symbols are also used to frame DLLPs and TLPs<sup>55</sup>, using distinct Special Symbols to allow these two types of Packets to be quickly and easily distinguished.

55. In Chapter 4, PMUX packets follow the TLP framing rules.

Table 4-1 shows the Special Symbols used for PCI Express and provides a brief description for each. These Symbols will be discussed in greater detail in following sections. Each of these Special Symbols, as well as the data Symbols, must be interpreted by looking at the 10-bit Symbol in its entirety.

*Table 4-1 Special Symbols*

Encoding	Symbol	Name	Description
K28.5	COM	Comma	Used for Lane and Link initialization and management
K27.7	STP	Start TLP	Marks the start of a Transaction Layer Packet
K28.2	SDP	Start DLLP	Marks the start of a Data Link Layer Packet
K29.7	END	End	Marks the end of a Transaction Layer Packet or a Data Link Layer Packet
K30.7	EDB	EnD Bad	Marks the end of a nullified TLP
K23.7	PAD	Pad	Used in Framing and Link Width and Lane ordering negotiations
K28.0	SKP	Skip	Used for compensating for different bit rates for two communicating Ports
K28.1	FTS	Fast Training Sequence	Used within an Ordered Set to exit from L0s to L0
K28.3	IDL	Idle	Used in the Electrical Idle Ordered Set (EIOS)
K28.4			Reserved
K28.6			Reserved
K28.7	EIE	Electrical Idle Exit	Reserved in 2.5 GT/s
			Used in the Electrical Idle Exit Ordered Set (EIEOS) and sent prior to sending FTS at data rates other than 2.5 GT/s

#### 4.2.1.1.3 8b/10b Decode Rules

The Symbol tables for the valid 8b/10b codes are given in Appendix B. These tables have one column for the positive disparity and one column for the negative disparity.

A Transmitter is permitted to pick any disparity, unless otherwise required, when first transmitting differential data after being in an Electrical Idle state. The Transmitter must then follow proper 8b/10b encoding rules until the next Electrical Idle state is entered.

The initial disparity for a Receiver that detects an exit from Electrical Idle is set to the disparity of the first Symbol used to obtain Symbol lock. Disparity may also be reinitialized if Symbol lock is lost and regained during the transmission of differential information due to an implementation specific number of errors. All following received Symbols after the initial disparity is set must be found in the proper column corresponding to the current running disparity.

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).

#### **4.2.1.2 Framing and Application of Symbols to Lanes**

There are two classes of framing and application of Symbols to Lanes. The first class consists of the Ordered Sets and the second class consists of TLPs and DLLPs. Ordered Sets are always transmitted serially on each Lane, such that a full Ordered Set appears simultaneously on all Lanes of a multi-Lane Link.

The Framing mechanism uses Special Symbol K28.2 “SDP” to start a DLLP and Special Symbol K27.7 “STP” to start a TLP. The Special Symbol K29.7 “END” is used to mark the end of either a TLP or a DLLP.

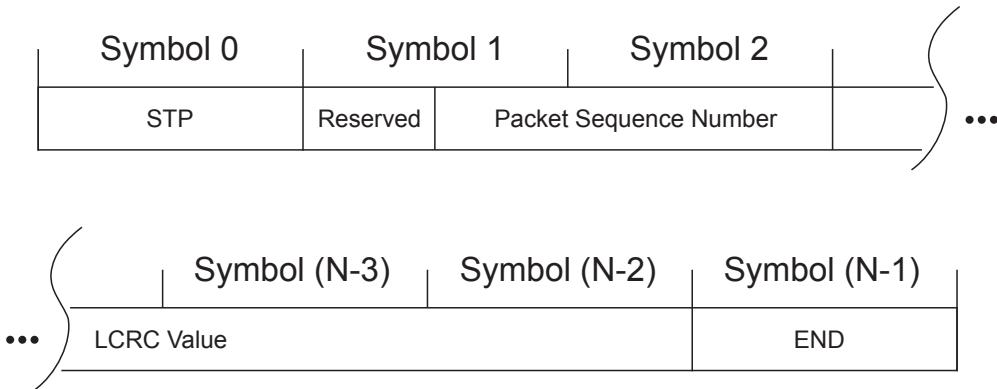
The conceptual stream of Symbols must be mapped from its internal representation, which is implementation dependent, onto the external Lanes. The Symbols are mapped onto the Lanes such that the first Symbol (representing Character 0) is placed onto Lane 0; the second is placed onto Lane 1; etc. The x1 Link represents a degenerate case and the mapping is trivial, with all Symbols placed onto the single Lane in order.

When no packet information or special Ordered Sets are being transmitted, the Transmitter is in the Logical Idle state. During this time idle data must be transmitted. The idle data must consist of the data byte 0 (00 Hexadecimal), scrambled according to the rules of [Section 4.2.1.3](#) and 8b/10b encoded according to the rules of [Section 4.2.1.1](#), in the same way that TLP and DLLP Data Symbols are scrambled and encoded. Likewise, when the Receiver is not receiving any packet information or special Ordered Sets, the Receiver is in Logical Idle and shall receive idle data as described above. During transmission of the idle data, the SKP Ordered Set must continue to be transmitted as specified in [Section 4.2.7](#).

For the following rules, “placed” is defined to mean a requirement on the Transmitter to put the Symbol into the proper Lane of a Link.

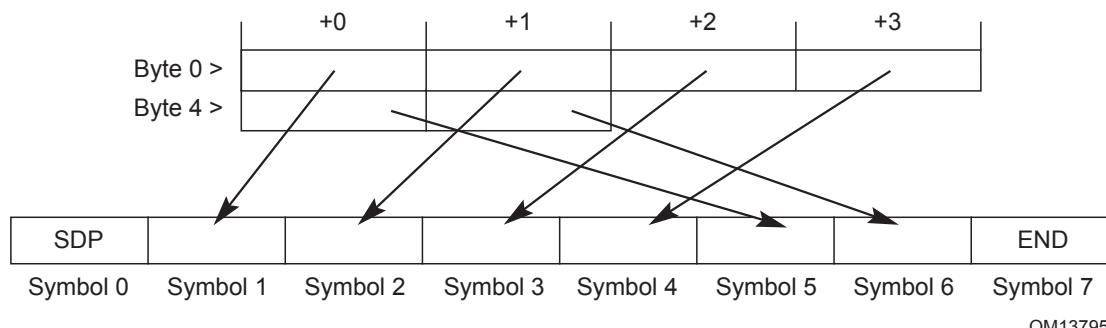
- TLPs must be framed by placing an STP Symbol at the start of the TLP and an END Symbol or EDB Symbol at the end of the TLP ([see Figure 4-5](#)).
- A properly formed TLP contains a minimum of 18 symbols between the STP and END or EDB Symbols. If a received sequence has less than 18 symbols between the STP and END or EDB symbols, the receiver is permitted to treat this as a Receiver Error.
  - If checked, this is a reported error associated with the Receiving Port ([see Section 6.2](#)).
- DLLPs must be framed by placing an SDP Symbol at the start of the DLLP and an END Symbol at the end of the DLLP ([see Figure 4-6](#)).
- Logical Idle is defined to be a period of one or more Symbol Times when no information: TLPs, DLLPs or any type of Special Symbol is being Transmitted/Received. Unlike Electrical Idle, during Logical Idle the Idle Symbol (00h) is being transmitted and received.
  - When the Transmitter is in Logical Idle, the Logical Idle data (00h) shall be transmitted on all Lanes. This is scrambled according to the rules in [Section 4.2.1.3](#).
  - Receivers must ignore incoming Logical Idle data, and must not have any dependency other than scramble sequencing on any specific data patterns.
- For Links wider than x1, the STP Symbol (representing the start of a TLP) must be placed in Lane 0 when starting Transmission of a TLP from a Logical Idle Link condition.
- For Links wider than x1, the SDP Symbol (representing the start of a DLLP) must be placed in Lane 0 when starting Transmission of a DLLP from a Logical Idle Link condition.
- The STP Symbol must not be placed on the Link more frequently than once per Symbol Time.
- The SDP Symbol must not be placed on the Link more frequently than once per Symbol Time.
- As long as the above rules are satisfied, TLP and DLLP Transmissions are permitted to follow each other successively.
- One STP Symbol and one SDP Symbol may be placed on the Link in the same Symbol Time.

- Links wider than x4 can have STP and SDP Symbols placed in Lane  $4^*N$ , where N is a positive integer. For example, for x8, STP and SDP Symbols can be placed in Lanes 0 and 4; and for x16, STP and SDP Symbols can be placed in Lanes 0, 4, 8, or 12.
- For  $xN$  Links where N is 8 or more, if an END or EDB Symbol is placed in a Lane K, where K does not equal N-1, and is not followed by a STP or SDP Symbol in Lane K+1 (i.e., there is no TLP or DLLP immediately following), then PAD Symbols must be placed in Lanes K+1 to Lane N-1.
  - For example, on a x8 Link, if END or EDB is placed in Lane 3, PAD must be placed in Lanes 4 to 7, when not followed by STP or SDP.
- The EDB Symbol is used to mark the end of a nullified TLP. Refer to [Section 3.6.2.1](#) for information on the usage of EDB.
- Receivers may optionally check for violations of the rules of this section. These checks are independently optional (see [Section 6.2.3.4](#)). If checked, violations are Receiver Errors, and are reported errors associated with the Port (see [Section 6.2](#)).



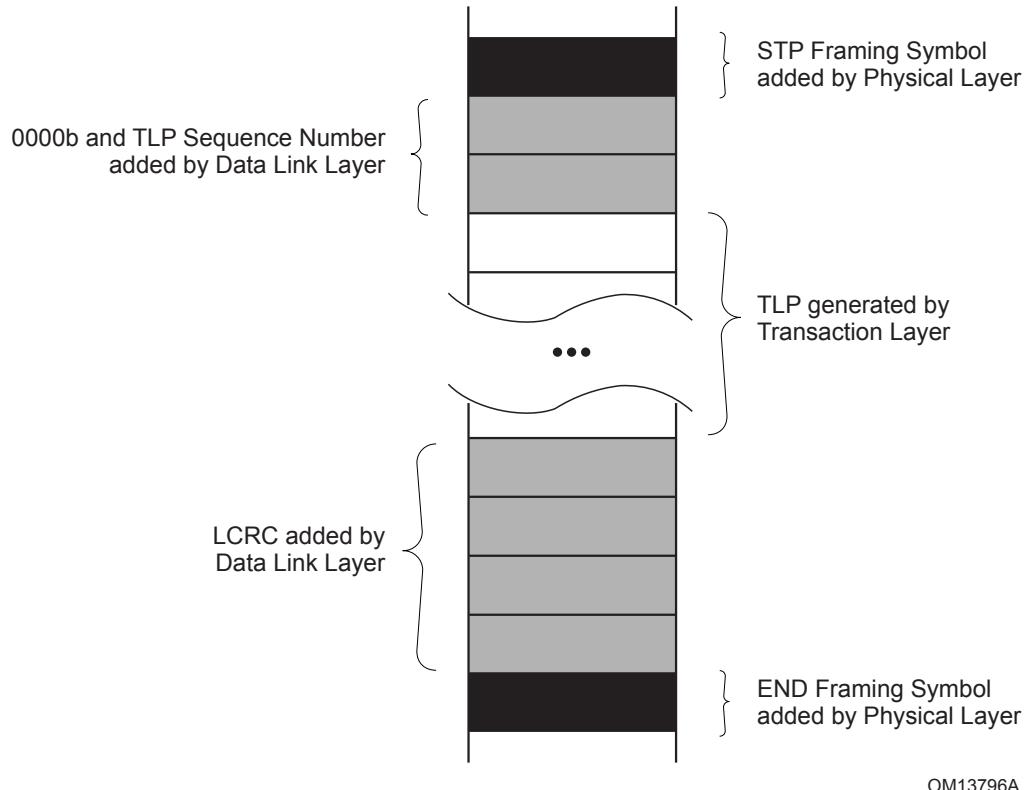
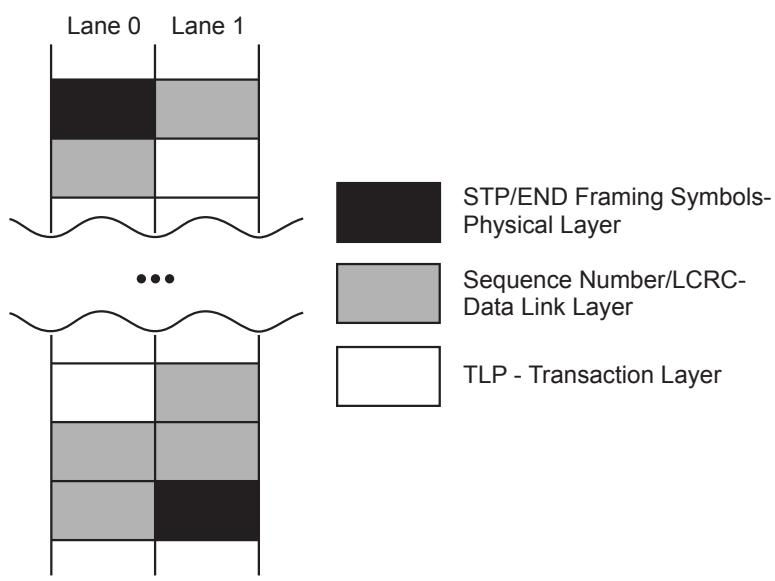
OM13794

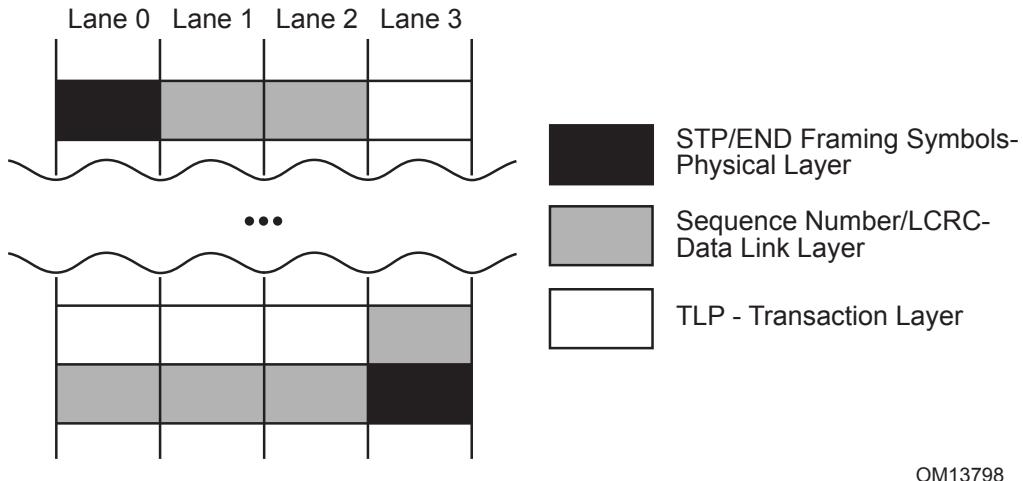
Figure 4-5 TLP with Framing Symbols Applied



OM13795

Figure 4-6 DLLP with Framing Symbols Applied

*Figure 4-7 Framed TLP on a  $\times 1$  Link**Figure 4-8 Framed TLP on a  $\times 2$  Link*



OM13798

Figure 4-9 Framed TLP on a x4 Link

#### 4.2.1.3 Data Scrambling

In order to improve electrical characteristics of a Link, data is typically scrambled. This involves XORing the data stream with a pattern generated by a Linear Feedback Shift Register (LFSR). On the Transmit side, scrambling is applied to characters prior to the 8b/10b encoding. On the Receive side, de-scrambling is applied to characters after 8b/10b decoding.

On a multi-Lane Link, the scrambling function can be implemented with one or many LFSRs. When there is more than one Transmit LFSR per Link, these must operate in concert, maintaining the same simultaneous (Lane-to-Lane Output Skew) value in each LFSR. When there is more than one Receive LFSR per Link, these must operate in concert, maintaining the same simultaneous (Lane-to-Lane Skew) value in each LFSR. Regardless of how they are implemented, LFSRs must interact with data on a Lane-by-Lane basis as if there was a separate LFSR as described here for each Lane within that Link.

The LFSR is graphically represented in Figure 4-10. Scrambling or unscrambling is performed by serially XORing the 8-bit (D0-D7) character with the 16-bit (D0-D15) output of the LFSR. An output of the LFSR, D15, is XORed with D0 of the data to be processed. The LFSR and data register are then serially advanced and the output processing is repeated for D1 through D7. The LFSR is advanced after the data is XORed. The LFSR implements the polynomial:

$$G(X)=X^{16}+X^5+X^4+X^3+1$$

The mechanism(s) and/or interface(s) utilized by the Data Link Layer to notify the Physical Layer to disable scrambling is implementation specific and beyond the scope of this specification.

The data scrambling rules are the following:

- The COM Symbol initializes the LFSR.
- The LFSR value is advanced eight serial shifts for each Symbol except the SKP.
- All data Symbols (D codes) except those within Ordered Sets (e.g., TS1, TS2, EIEOS), the Compliance Pattern (see Section 4.2.8), and the Modified Compliance Pattern (see Section 4.2.9) are scrambled.
- All special Symbols (K codes) are not scrambled.

- The initialized value of an LFSR seed (D0-D15) is FFFFh. Immediately after a COM exits the Transmit LFSR, the LFSR on the Transmit side is initialized. Every time a COM enters the Receive LFSR on any Lane of that Link, the LFSR on the Receive side is initialized.
- Scrambling can only be disabled at the end of Configuration (see [Section 4.2.6.3.5](#) ).
- Scrambling does not apply to a [Loopback Slave](#).
- Scrambling is always enabled in Detect by default.

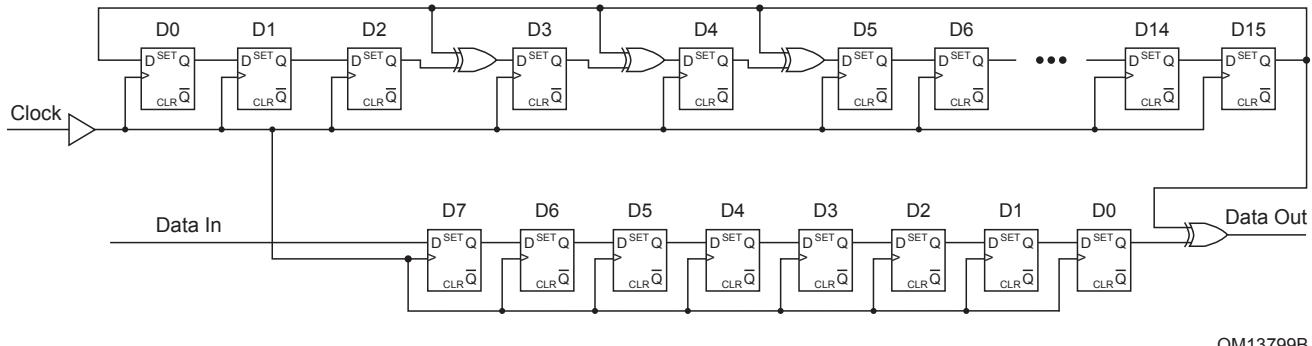
## IMPLEMENTATION NOTE

### Disabling Scrambling

Disabling scrambling is intended to help simplify test and debug equipment. Control of the exact data patterns is useful in a test and debug environment. Since scrambling is reset at the Physical Layer there is no reasonable way to reliably control the state of the data transitions through software. Thus, the Disable Scrambling bit in the TS1 and TS2 Ordered Sets is provided for these purposes.

The mechanism(s) and/or interface(s) utilized by the Data Link Layer to notify the Physical Layer to disable scrambling is implementation specific and beyond the scope of this specification.

For more information on scrambling, see [Appendix C](#) .



OM13799B

*Figure 4-10 LFSR with 8b/10b Scrambling Polynomial*

### 4.2.2 Encoding for 8.0 GT/s and Higher Data Rates

When a PCI Express Link is operating at a data rate of 8.0 GT/s or higher, it uses the encoding rules described in this subsection: 128b/130b encoding. For backwards compatibility, the Link initially trains to L0 at the 2.5 GT/s data rate using 8b/10b encoding as described in [Section 4.2.1](#) , then when the data rate is changed to 8.0 GT/s or higher, 128b/130b encoding is used. 128b/130b encoding is a Link-wide packetization mechanism and a per-Lane block code with scrambling. The basic entity of data transmission is an 8-bit data character, referred to as a Symbol, as shown in [Figure 4-11](#) and [Figure 4-12](#) .

## IMPLEMENTATION NOTE

### Symbol in 128b/130b Encoding Scheme

In the 128b/130b encoding scheme, the Symbol is one byte long, similar to the 10-bit Symbol of 8b/10b encoding.

#### 4.2.2.1 Lane Level Encoding

The physical layer uses a per-Lane block code. Each Block consists of a 2-bit Sync Header and a payload. There are two valid Sync Header encodings: 10b and 01b. The Sync Header defines the type of payload that the Block contains.

A Sync Header of 10b indicates a Data Block. Each Data Block has a 128 bit payload, resulting in a Block size of 130 bits. The payload is a Data Stream described in [Section 4.2.2.3](#).

A Sync Header of 01b indicates an Ordered Set Block. Each Ordered Set Block has a 128 bit payload, resulting in a Block size of 130 bits except for the SKP Ordered Set which can be of variable length.

All Lanes of a multi-Lane Link must transmit Blocks with the same Sync Header simultaneously, except when transmitting Jitter Measurement Pattern in Polling.Compliance.

The bit transmission order is as follows. A Sync Header represented as ‘H<sub>1</sub>H<sub>0</sub>’ is placed on a Lane starting with ‘H<sub>0</sub>’ and ending with ‘H<sub>1</sub>’. A Symbol, represented as ‘S<sub>7</sub>S<sub>6</sub>S<sub>5</sub>S<sub>4</sub>S<sub>3</sub>S<sub>2</sub>S<sub>1</sub>S<sub>0</sub>’, is placed on a Lane starting with ‘S<sub>0</sub>’ and ending with ‘S<sub>7</sub>’. In the diagrams that show a time scale, bits represent the transmission order. In layout diagrams, bits are arranged in little-endian format, consistent with packet layout diagrams in other chapters of this specification.

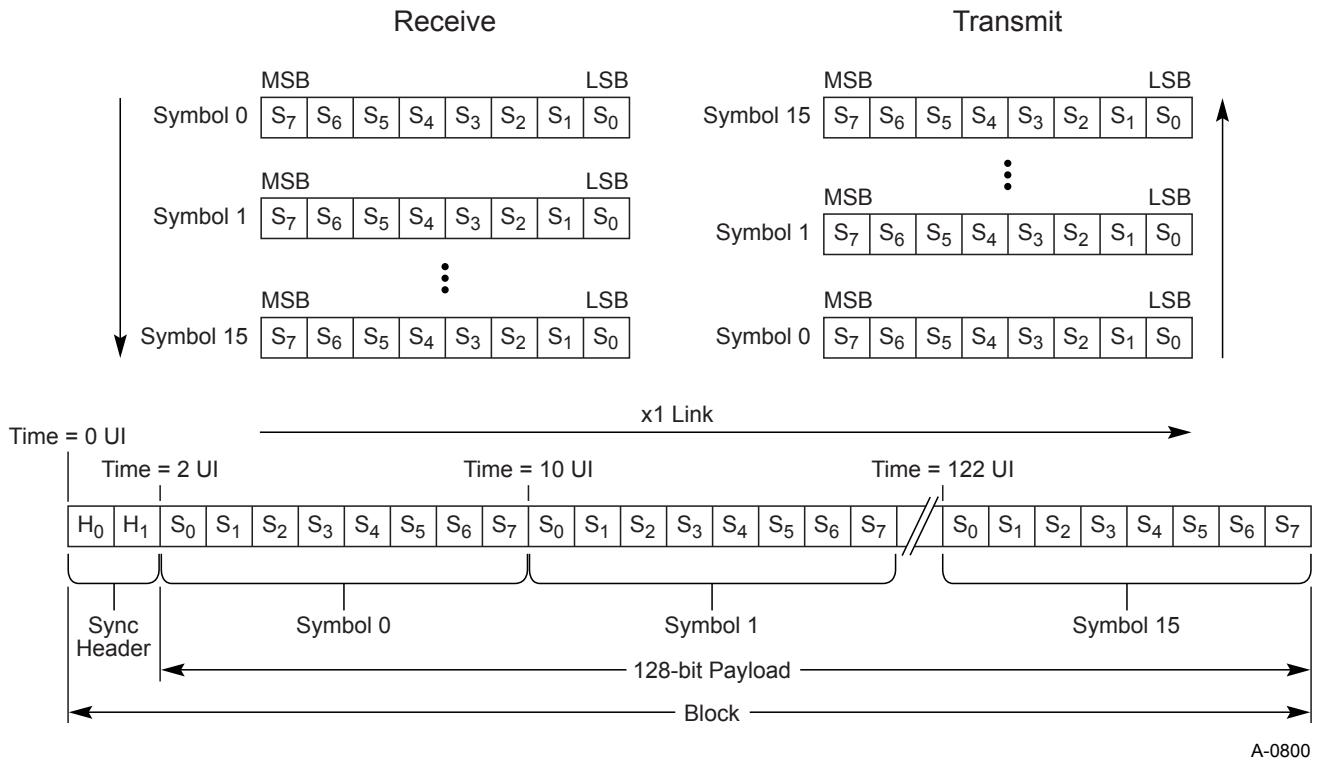
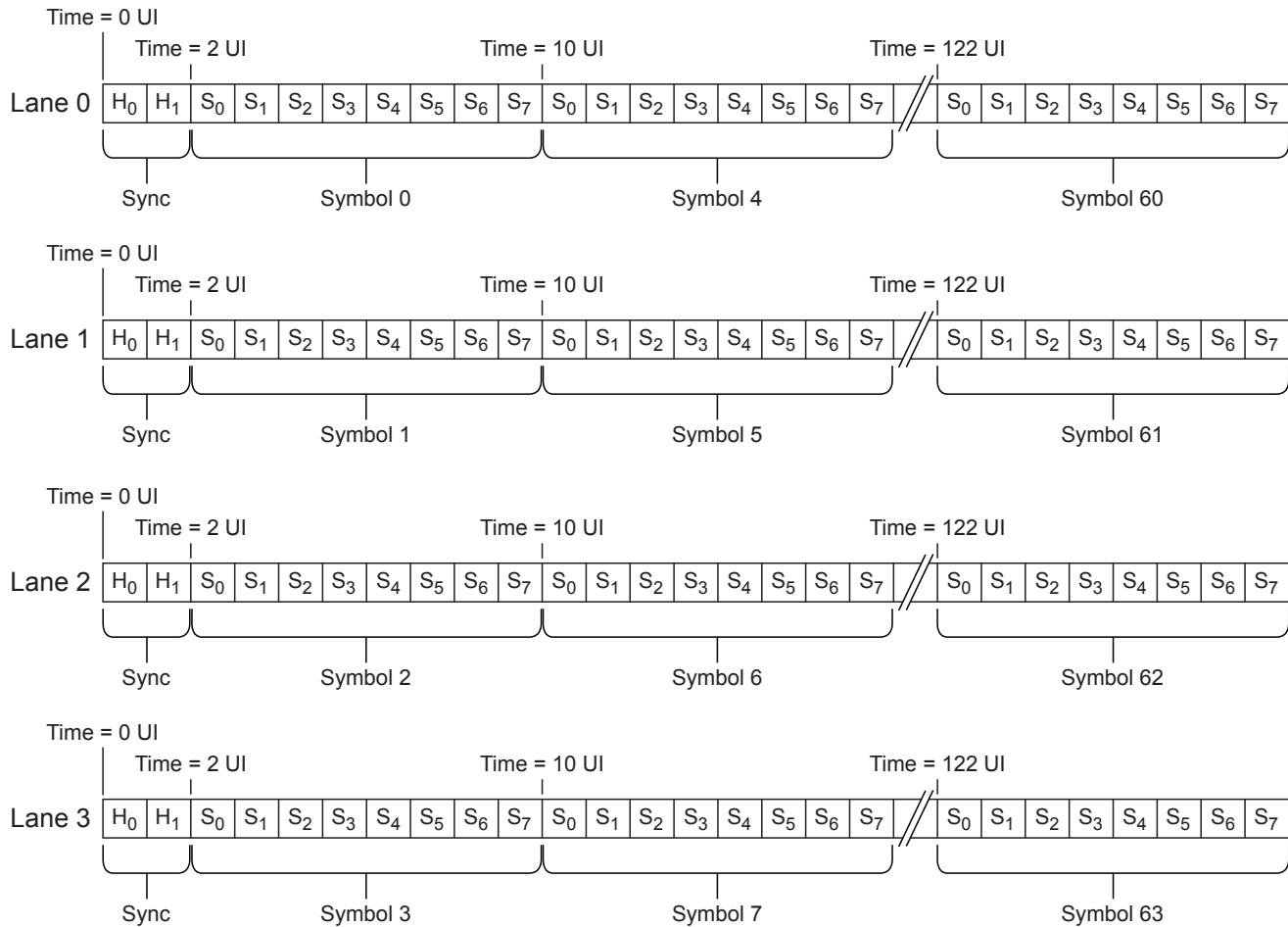


Figure 4-11 Example of Bit Transmission Order in a  $x1$  Link Showing 130 Bits of a Block



A-0801

Figure 4-12 Example of Bit Placement in a x4 Link with One Block per Lane

#### 4.2.2.2 Ordered Set Blocks

An Ordered Set Block contains a Sync Header followed by one Ordered Set. All Lanes of a multi-Lane Link must transmit the same Ordered Set type simultaneously. The first Symbol of the Ordered Set defines the type of Ordered Set. Subsequent symbols of the Ordered Set are defined by the Ordered Set type and need not be identical across lanes of a multi-Lane Link. The Ordered Sets are described in detail in [Section 4.2.4](#) and [Section 4.2.7](#).

##### 4.2.2.2.1 Block Alignment

During Link training, the 130 bits of the Electrical Idle Exit Ordered Set (EIEOS) are a unique bit pattern that Receivers use to determine the location of the Block Sync Headers in the received bit stream. Conceptually, Receivers can be in three different phases of Block alignment: Unaligned, Aligned, and Locked. These phases are defined to illustrate the required behavior, but are not meant to specify a required implementation.

###### Unaligned Phase

Receivers enter this phase after a period of Electrical Idle, such as when the data rate is changed to one that uses 128b/130b encoding or when they exit a low-power Link state, or if directed (by an implementation specific

method). In this phase, Receivers monitor the received bit stream for the EIEOS bit pattern. When one is detected, they adjust their alignment to it and proceed to the Aligned phase.

#### **Aligned Phase**

Receivers monitor the received bit stream for the EIEOS bit pattern and the received Blocks for a Start of Data Stream (SDS) Ordered Set. If an EIEOS bit pattern is detected on an alignment that does not match the current alignment, Receivers must adjust their alignment to the newly received EIEOS bit pattern. If an SDS Ordered Set is received, Receivers proceed to the Locked phase. Receivers are permitted to return to the Unaligned phase if an undefined Sync Header (00b or 11b) is received.

#### **Locked Phase**

Receivers must not adjust their Block alignment while in this phase. Data Blocks are expected to be received after an SDS Ordered Set, and adjusting the Block alignment would interfere with the processing of these Blocks. Receivers must return to the Unaligned or Aligned phase if an undefined Sync Header is received.

## **IMPLEMENTATION NOTE**

### **Detection of Loss of Block Alignment**

The sequence of EIEOS and TS Ordered Sets transmitted during training sequences will cause misaligned Receivers to detect an undefined Sync Header.

Additional Requirements:

- While in the Aligned or Locked phase, Receivers must adjust their alignment as necessary when a SKP Ordered Set is received. See [Section 4.2.7](#) for more information on SKP Ordered Sets.
- After any LTSSM transition to Recovery, Receivers must ignore all received TS Ordered Sets until they receive an EIEOS. Conceptually, receiving an EIEOS validates the Receiver's alignment and allows TS Ordered Set processing to proceed. If a received EIEOS initiates an LTSSM transition from L0 to Recovery, Receivers are permitted to process any TS Ordered Sets that follow the EIEOS or ignore them until another EIEOS is received after entering Recovery.
- Receivers are permitted to transition from the Locked phase to the Unaligned or Aligned phase as long as Data Stream processing is stopped. See [Section 4.2.2.3](#) for more information on Data Stream requirements.
- Loopback Masters: While in Loopback.Entry, Masters must be capable of adjusting their Receiver's Block alignment to received EIEOS bit patterns. While in Loopback.Active, Masters are permitted to transmit an EIEOS and adjust their Receiver's Block alignment to the looped back bit stream.
- Loopback Slaves: While in Loopback.Entry, Slaves must be capable of adjusting their Receiver's Block alignment to received EIEOS bit patterns. While in Loopback.Active, Slaves must not adjust their Receiver's Block alignment. Conceptually, the Receiver is directed to the Locked phase when the Slave starts to loop back the received bit stream.

#### **4.2.2.3 Data Blocks**

The payload of Data Blocks is a stream of Symbols defined as a “Data Stream” that consists of Framing Tokens, TLPs, and DLLPs. Each Symbol of the Data Stream is placed on a single Lane of the Link, and the stream of Symbols is striped across all Lanes of the Link and spans Block boundaries.

A Data Stream starts with the first Symbol of the Data Block that follows an SDS Ordered Set. It ends either when a Framing Error is detected or with the last Symbol of the Data Block that precedes an Ordered Set other than a SKP

Ordered Set. SKP Ordered Sets that occur within a Data Stream have specific requirements as described in the following sections.

#### 4.2.2.3.1 Framing Tokens

The Framing Tokens used by the Physical Layer are shown in [Table 4-2](#). Each Framing Token specifies or implies the number of Symbols associated with the Token and therefore the location of the next Framing Token. [Figure 4-15](#) shows an example of TLPs, DLLPs, and IDLs transmitted on a x8 link.

The first Framing Token of a Data Stream is always located in Symbol 0 of Lane 0 of the first Data Block of the Data Stream. For the rest of this chapter, the terms Framing Token and Token are used interchangeably.

*Table 4-2 Framing Token Encoding*

Framing Token Type	Description
<b>IDL</b>	Logical Idle. The Framing Token is 1 Symbol. This Token is transmitted when no TLPs or DLLPs or other Framing Tokens are being transmitted.
<b>SDP</b>	Start of DLLP. The Framing Token is 2 Symbols long and is followed by the DLLP information.
<b>STP</b>	Start of TLP. The Framing Token is 4 Symbols long and includes the 12-bit TLP Sequence Number. It is followed by the TLP information.
<b>EDB</b>	EnD Bad. The Framing Token is 4 Symbols long and is used to confirm that the previous TLP was nullified.
<b>EDS</b>	End of Data Stream. The Framing Token is four Symbols long and indicates that the next Block will be an Ordered Set Block.

+0	+1	+2	+3
7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0
TLP Length[3:0]	1111b	F P	TLP Length[10:4]

STP Token

+0	+1	+2	+3
7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0
0001b	1111b	1b	0000000b

EDS Token

+0	+1	+2	+3
7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0
11000000b	11000000b	11000000b	11000000b

EDB Token

+0	+1
7   6   5   4   3   2   1   0	7   6   5   4   3   2   1   0
11110000b	10101100b

SDP Token

+0
7   6   5   4   3   2   1   0
00000000b

IDL Token

A-0802

Figure 4-13 Layout of Framing Tokens

The Physical Layer DLLP layout is shown in Figure 4-14. Symbols 0 and 1 are the SDP Token, and Symbols 2 through 7 are the Data Link Layer DLLP information.

The Physical Layer TLP layout is shown in [Figure 4-14](#). Details of the STP Framing Token are shown in [Figure 4-13](#). The length of the TLP (in DWs) being transmitted is specified by an 11-bit field called TLP Length. The TLP Length field is the total amount of information transferred, including the Framing Token, TLP Prefixes (if any), TLP Header, TLP data payload (if any), TLP digest (if any), and TLP LCRC. For example, if a TLP has a 3 DW header, a 1 DW data payload, and does not include a TLP digest, the TLP Length field value is 6: 1 (Framing Token) + 0 (TLP Prefixes) + 3 (TLP header) + 1 (TLP data payload) + 0 (TLP digest) + 1 (TLP LCRC). If the same TLP included a TLP digest, the TLP Length field value would be 7. When a TLP is nullified, the EDB Token is considered an extension of the TLP but is not included in the calculation of the TLP Length field.

The TLP Length field is protected by a 4-bit CRC (Frame CRC), and an even parity bit (Frame Parity) protects both the TLP Length and Frame CRC fields. The Frame CRC and Frame Parity are calculated as follows:

$$C[0] = L[10] \wedge L[7] \wedge L[6] \wedge L[4] \wedge L[2] \wedge L[1] \wedge L[0]$$

$$C[1] = L[10] \wedge L[9] \wedge L[7] \wedge L[5] \wedge L[4] \wedge L[3] \wedge L[2]$$

$$C[2] = L[9] \wedge L[8] \wedge L[6] \wedge L[4] \wedge L[3] \wedge L[2] \wedge L[1]$$

$$C[3] = L[8] \wedge L[7] \wedge L[5] \wedge L[3] \wedge L[2] \wedge L[1] \wedge L[0]$$

$$P = L[10] \wedge L[9] \wedge L[8] \wedge L[7] \wedge L[6] \wedge L[5] \wedge L[4] \wedge L[3] \wedge L[2] \wedge L[1] \wedge L[0] \wedge C[3] \wedge C[2] \wedge C[1] \wedge C[0]$$

The Frame Parity reduces to  $P = L[10] \wedge L[9] \wedge L[8] \wedge L[6] \wedge L[5] \wedge L[2] \wedge L[0]$

The TLP Length field is represented in the above equations as  $L[10:0]$ , where  $L[0]$  is the least significant bit and  $L[10]$  is the most significant bit. Transmitters calculate the Frame CRC and Frame Parity before transmission. Receivers must calculate the Frame CRC and Frame Parity using the same algorithm as the transmitter and then compare the calculated values to the received values.

STP Tokens do not have a TLP Length field value of 1. If a received sequence of Symbols matches the format of an STP Token with a TLP Length field value of 1, the Symbols are evaluated to determine whether they match the EDS Token.

## IMPLEMENTATION NOTE

### Frame CRC and Frame Parity

The Frame CRC bits are effectively calculated as  $(L[0] X^{14} + L[1] X^{13} + \dots + L[9] X^5 + L[10] X^4) \bmod (X^4 + X + 1)$ . It should be noted that  $X^4 + X + 1$  is a primitive polynomial and the CRC can detect two bit errors. The Frame Parity bit can detect an odd number of bit errors. Thus, the Frame CRC and Frame Parity together guarantee three bit error detection for the TLP Length field. It must be noted that even though in the reduced Frame Parity equation all terms are not present, it still maintains the property of detecting odd bit errors. Only those TLP Length field bits which are present in an even number of CRC terms are used in the calculation.

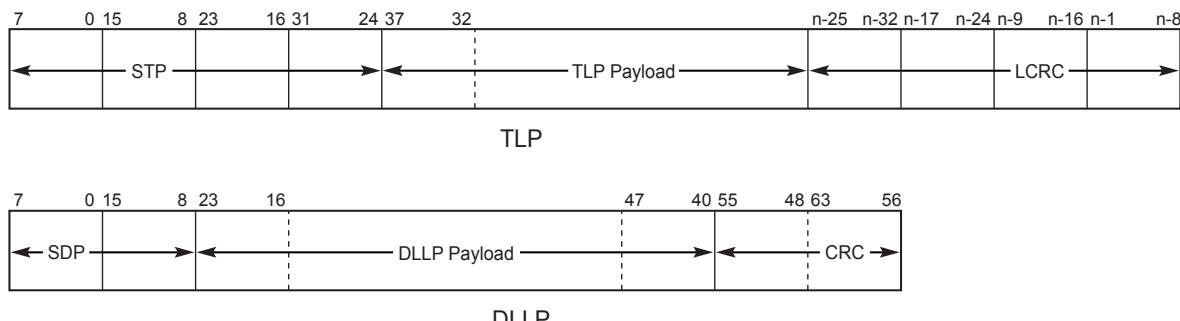
Note that, for TLPs, the Data Link Layer prepends 4 Reserved bits (0000b) to the TLP Sequence Number field before it calculates the LCRC. These Reserved bits are not explicitly transmitted when using 128b/130b encoding, and Receivers assume that the 4 bits received are 0000b when calculating the LCRC.

Packets containing a TLP Length field that is greater than 1535 are PMUX Packets. For such packets, the actual packet length is computed differently, the TLP Sequence Number field in the STP Token contains other information, and the Link CRC is computed using different rules. See [Appendix G](#) for details.

Packets containing a TLP Length field that is between 1152 and 1535 (inclusive) are reserved for future standardization.

Transmitters must transmit all DWs of a TLP specified by the TLP Length field of the STP Framing Token. TLPs are never truncated when using 128b/130b encoding - even when nullified. [Figure 4-16](#) shows an example of a nullified 23 DW TLP.

Figure 4-17 shows an example of TLPs, DLLPs, IDLs, and an EDS Token followed by a SKP Ordered Set. SKP Ordered Sets are defined in Section 4.2.7.2.



A-0803

Figure 4-14 TLP and DLLP Layout

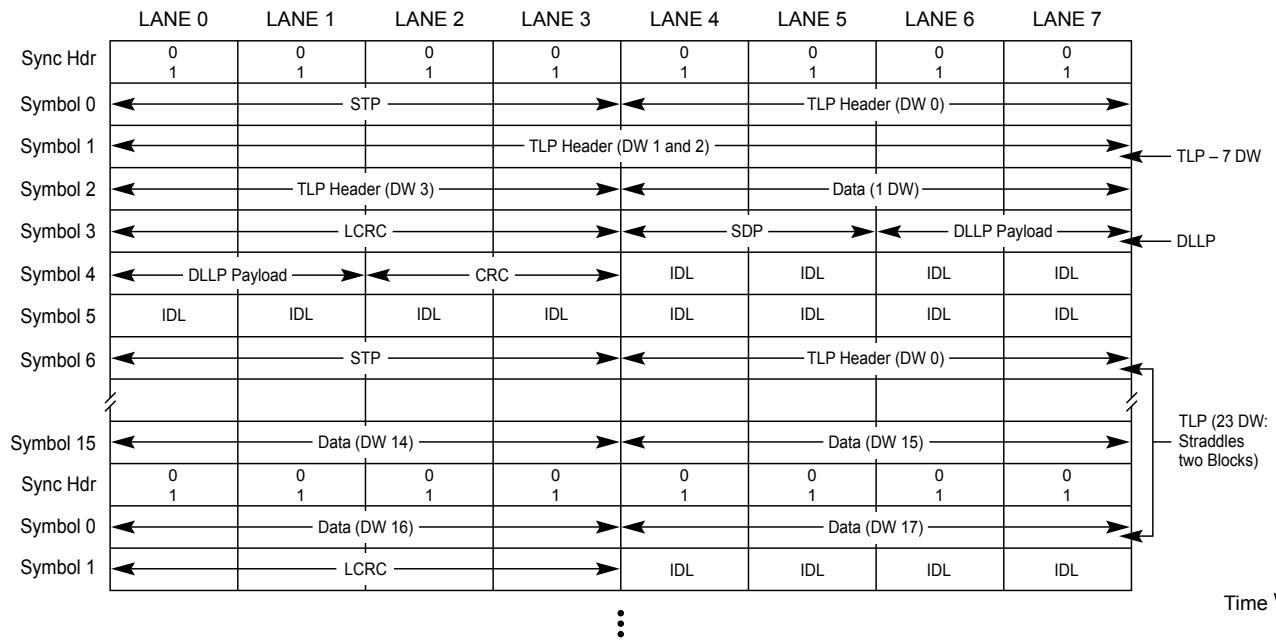


Figure 4-15 Packet Transmission in a x8 Link

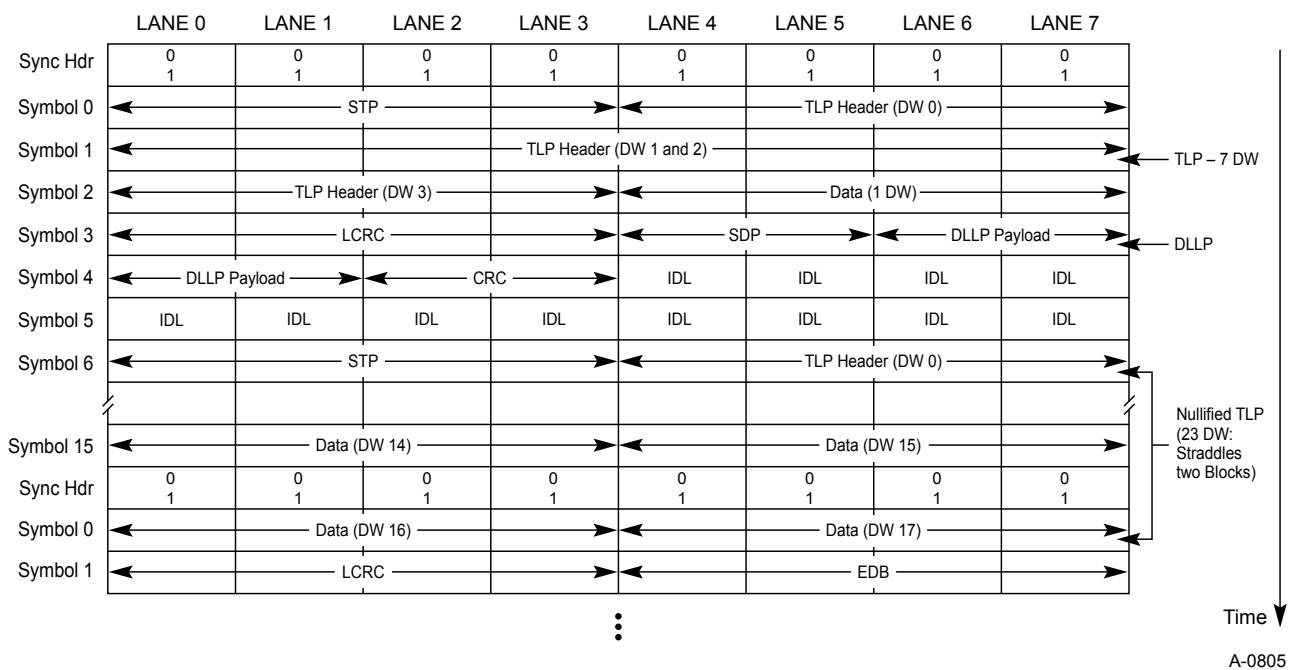
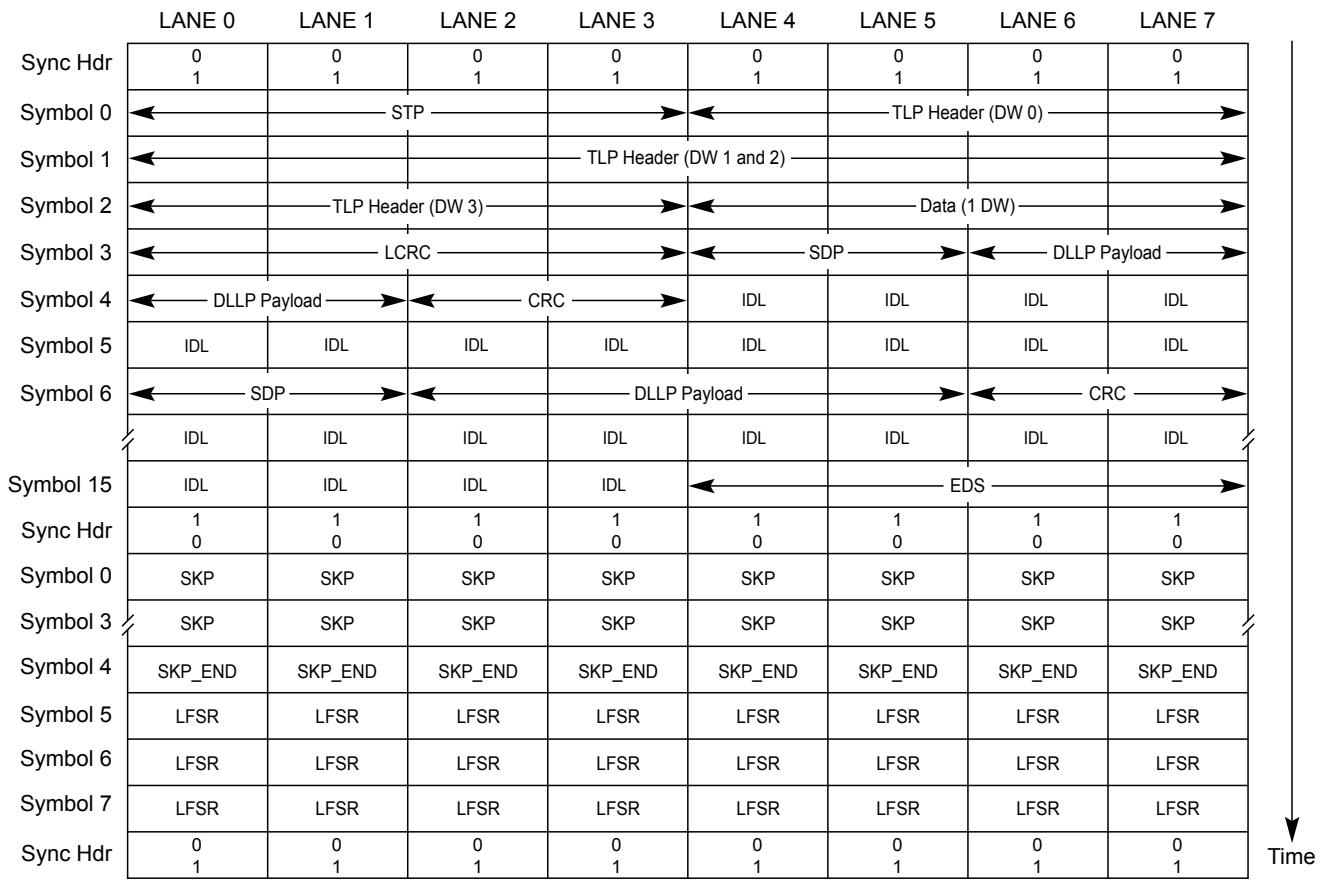


Figure 4-16 Nullified TLP Layout in a x8 Link with Other Packets



A-0806

Figure 4-17 SKP Ordered Set of Length 66-bit in a x8 Link

#### 4.2.2.3.2 Transmitter Framing Requirements

The following requirements apply to the transmitted Data Stream.

- To Transmit a TLP:
  - Transmit an STP Token immediately followed by the complete TLP information provided by the Data Link Layer.
  - All DWs of the TLP, as specified by the TLP Length field of the STP Token, must be transmitted, even if the TLP is nullified.
  - If the TLP is nullified, an EDB Token must be transmitted immediately following the TLP. There must be no Symbols between the last Symbol of the TLP and the first Symbol of the EDB Token. The value of the TLP Length field of a nullified TLP's STP Token is NOT adjusted to account for the EDB Token.
  - The STP Token must not be transmitted more frequently than once per Symbol Time.
- To Transmit a DLLP:
  - Transmit an SDP Token immediately followed by the complete DLLP information provided by the Data Link Layer.
  - All 6 Symbols of the DLLP must be transmitted.

- The SDP Token must not be transmitted more frequently than once per Symbol Time.
- To Transmit a SKP Ordered Set within a Data Stream:
  - Transmit an EDS Token in the last DW of the current Data Block. For example, the Token is transmitted on Lane 0 in Symbol Times 12-15 of the Block for a x1 Link, and on Lanes 12-15 of Symbol Time 15 of the Block for a x16 Link.
  - Transmit the SKP Ordered Set following the current Data Block.
  - Transmit a Data Block following the SKP Ordered Set. The Data Stream resumes with the first Symbol of the Data Block. If multiple SKP Ordered Sets are scheduled for transmission, each SKP Ordered Set must be preceded by a Data Block with an EDS Token.
- To end a Data Stream:
  - Transmit an EDS Token in the last DW of the current Data Block, followed in the next block by an EOS or an EEOS. An EOS is transmitted for LTSSM power management state transitions, and an EEOS is transmitted for all other cases. For example, the Token is transmitted on Lane 0 in Symbol Times 12-15 of the Block for a x1 Link, and on Lanes 12-15 of Symbol Time 15 of the Block for a x16 Link.
- The IDL Token must be transmitted on all Lanes when not transmitting a TLP, DLLP, or other Framing Token.
- Multi-Lane Links:
  - After transmitting an IDL Token, the first Symbol of the next STP or SDP Token must be transmitted in Lane 0 of a future Symbol Time. An EDS Token can be transmitted after an IDL Token in the same Symbol Time, since it must be transmitted in the last DW of a Block.
  - For xN Links where N is 8 or more, if an EDB Token, TLP, or DLLP ends in a Lane K, where K does not equal N-1, and it is not followed by the first Symbol of an STP, SDP, or EDB Token in Lane K+1, then IDL Tokens must be placed in Lanes K+1 to N-1. For example, on a x8 Link, if a TLP or DLLP ends in Lane 3, IDL Tokens must be placed in Lanes 4 to 7. The EDS Token is an exception to this requirement, and can be transmitted following IDL Tokens.
  - Tokens, TLPs, and DLLPs are permitted to follow each other successively such that more than one Token may be transmitted in the same Symbol Time as long as their transmission conforms with the other requirements stated in this section.
    - Links wider than x4 can have Tokens placed starting on Lane  $4^*N$ , where N is a positive integer. For example, Tokens can be placed in Lanes 0 and 4 of a x8 Link, and Tokens can be placed in Lanes 0, 4, 8, or 12 of a x16 Link.

#### 4.2.2.3.3 Receiver Framing Requirements

The following requirements apply to the received Data Stream and the Block type transitions that occur at the beginning and end of the Data Stream.

- When processing Symbols that are expected to be a Framing Token, receiving a Symbol or sequence of Symbols that does not match the definition of a Framing Token is a Framing Error. It is strongly recommended that Receivers of a multi-Lane Link report an error in the Lane Error Status Register for the Lane that receives the first Symbol of an expected Framing Token when that Symbol does not match Symbol +0 of an STP (bits [3:0] only), IDL, SDP, EDB, or EDS Token (see [Figure 4-13](#)).
- All optional error checks and error reports in this section are independently optional (see [Section 6.2.3.4](#)).
- When an STP Token is received:
  - Receivers must calculate the Frame CRC and Frame Parity of the received TLP Length field and compare the results to the received Frame CRC and Frame Parity fields. A Frame CRC or Frame Parity mismatch is a Framing Error.

- An STP Token with Framing Error is not considered part of a TLP for the purpose of reporting to the Data Link Layer.
  - If the TLP Length field is 1, the Symbols are not an STP Token and are instead evaluated to determine whether they are an EDS Token.
  - Receivers are permitted to check whether the TLP Length field has a value of 0. If checked, receiving a TLP Length field of 0 is a Framing Error.
  - Receivers are permitted to check whether the TLP Length field has a value of 2, 3, or 4. If checked, receiving such a TLP Length field is a Framing Error.
  - Receivers are permitted to check whether the TLP Length field has a value between 1152 and 1535 (inclusive). If checked, receiving such a TLP Length field is a Framing Error.
  - Receivers on Ports that do not support Protocol Multiplexing are permitted to check whether the TLP Length field has a value greater than 1535. If checked, receiving such a TLP Length field is a Framing Error.
  - Receivers on Ports that support Protocol Multiplexing, shall process STP Tokens with a TLP Length field that is greater than 1535 as the start of a PMUX Packet as defined in [Appendix G](#).
  - The next Token to be processed begins with the Symbol immediately following the last DW of the TLP, as determined by the TLP Length field.
    - Receivers must evaluate this Symbol and determine whether it is the first Symbol of an EDB Token and therefore whether the TLP is nullified. See the EDB Token requirements.
  - Receivers are permitted to check whether more than one STP Token is received in a single Symbol Time. If checked, receiving more than one STP Token in a single Symbol Time is a Framing Error
- When an EDB Token is received:
    - If an EDB Token is received immediately following a TLP (there are no Symbols between the last Symbol of the TLP and the first Symbol of the EDB Token), receivers must inform the Data Link Layer that an EDB Token has been received. Receivers are permitted to inform the Data Link Layer that an EDB Token has been received after processing the first Symbol of the EDB Token or after processing any or all of the remaining Symbols of the EDB Token. Regardless of when they inform the Data Link Layer of a received EDB Token, Receivers must check all Symbols of the EDB Token. Receiving a Symbol that does not match the definition of an EDB Token is a Framing Error.
    - Receiving an EDB Token at any time other than immediately following a TLP is a Framing Error.
    - The next Token to be processed begins with the Symbol immediately following the EDB Token.
  - When an EDS Token is received in the last four Symbols of the Data Block across the Link:
    - Receivers must stop processing the Data Stream.
    - Receiving an Ordered Set other than SKP, EIOS, or EIEOS in the Block following the EDS Token is a Framing Error.
    - If a SKP Ordered Set is received in the Block following the EDS Token, Receivers resume Data Stream processing with the first Symbol of the Data Block that follows the SKP Ordered Set unless a Framing Error has been detected.
  - When an SDP Token is received:
    - The next Token to be processed begins with the Symbol immediately following the last Symbol of the DLLP.
    - Receivers are permitted to check whether more than one SDP Token is received in a single Symbol Time. If checked, receiving more than one SDP Token in a single Symbol Time is a Framing Error.
  - When an IDL Token is received:
    - For a x1 Link, the next Token to be processed begins with the next Symbol received.

- For a x2 Link, the next Token to be processed begins with the Symbol received in Lane 0 of the next Symbol Time. It is strongly recommended that Receivers check whether the Symbol received in Lane 1, if it did not receive IDL, after an IDL Token was received in Lane 0 is also IDL and report an error for Lane 1 in the Lane Error Status Register. If checked, receiving a Symbol other than IDL is a Framing Error.
- For a x4 Link, the next Token to be processed begins with the Symbol received in Lane 0 of the next Symbol Time. It is strongly recommended that Receivers check whether the Symbols received in Lanes 1-3, after an IDL Token was received in Lane 0 are also IDL and report an error for the Lane(s) that did not receive IDL, in the Lane Error Status Register. If checked, receiving a Symbol other than IDL is a Framing Error.
- For x8, x12, x16, and x32 Links, the next Token to be processed begins with the Symbol received in the next DW aligned Lane following the IDL Token. For example, if an IDL Token is received in Lane 4 of a x16 Link, the next Token location begins with Lane 8 of the same Symbol Time. However, if an IDL Token is received on Lane 4 of a x8 Link, the next Token location begins with Lane 0 of the following Symbol Time. It is strongly recommended that Receivers check whether the Symbols received between the IDL Token and the next Token location are also IDL and report an error for the Lane(s) that did not receive IDL, in the Lane Error Status Register. If checked, receiving a Symbol other than IDL is a Framing Error.

Note: The only Tokens expected to be received in the same Symbol Time following an IDL Token are additional IDL Tokens or an EDS Token.

- While processing the Data Stream, Receivers must also check the Block type received by each Lane, after accounting for Lane-to-Lane de-skew, for the following conditions:
  - Receiving an Ordered Set Block on any Lane immediately following an SDS Ordered Set is a Framing Error.
  - Receiving a Block with an undefined Block type (a Sync Header of 11b or 00b) is a Framing Error. It is strongly recommended that Receivers of a multi-Lane Link report an error for any Lane that received the undefined Block type in the Lane Error Status register.
  - Receiving an Ordered Set Block on any Lane without receiving an EDS token in the preceding Block is a Framing Error. For example, receiving a SKP Ordered Set without a preceding EDS Token is a Framing Error. In addition, receiving a SKP Ordered Set followed immediately by another Ordered Set Block (including another SKP Ordered Set) within a Data Stream is a Framing Error. It is strongly recommended that if the first Symbol of the Ordered Set is SKP, Receivers of a multi-Lane Link report an error for the Lane(s) in the Lane Error Status register if the received Symbol number 1 through 4N does not match the corresponding Symbol in [Table 4-22](#) or [Table 4-23](#).
  - Receiving a Data Block on any Lane when the previous block contained an EDS Token is a Framing Error. It is strongly recommended that Receivers of a multi-Lane Link report an error for the Lane(s) that received the Data Block in the Lane Error Status register.
  - Receivers are permitted to check for different Ordered Sets on different Lanes. For example, Lane 0 receives a SKP Ordered Set and Lane 1 receives an EOS. If checked, receiving different Ordered Sets is a Framing Error.

#### **4.2.2.3.4 Recovery from Framing Errors**

If a receiver detects a Framing Error while processing the Data Stream, it must:

- Report a Receiver Error as described in [Section 4.2.4.8](#).
- Stop processing the Data Stream. Processing of a new Data Stream is initiated when the next SDS Ordered Set is received as previously described.

- Initiate the error recovery process as described in [Section 4.2.4.8](#). If the LTSSM state is L0, direct the LTSSM to Recovery. If the LTSSM state is Configuration.Complete or Configuration.Idle when the Framing Error is detected, the error recovery process is satisfied by either a transition from Configuration.Idle to Recovery.RcvrLock due to the specified timeout, or a transition to Recovery through L0. If the LTSSM state is Recovery.RcvrCfg or Recovery.Idle when the Framing Error is detected, the error recovery process is satisfied by either a transition from Recovery.Idle to Recovery.RcvrLock due to the specified timeout, or a directed transition from L0 to Recovery. If the LTSSM substate is either Recovery.RcvrLock or Configuration.Linkwidth.Start, the error recovery process is satisfied upon exit from these substates and no direction of the LTSSM to Recovery is required.
  - Note: The framing error recovery mechanism is not expected to directly cause any Data Link Layer initiated recovery action such as NAK.

## IMPLEMENTATION NOTE

### Time Spent in Recovery Due to Detection of a Framing Error

When using 128b/130b encoding, all Framing Errors require Link recovery. It is expected that implementations will require less than 1 microsecond to recover from a Framing Error as measured from the time that both Ports have entered the Recovery state.

#### 4.2.2.4 Scrambling

Each Lane of the transmitter in a multi-Lane Link may implement a separate LFSR for scrambling. Each Lane of the receiver in a multi-Lane Link may implement a separate LFSR for descrambling. Implementations may choose to implement fewer LFSRs, but must achieve the same functionality as independent LFSRs.

The LFSR uses the following polynomial:  $G(X) = X^{23} + X^{21} + X^{16} + X^8 + X^5 + X^2 + 1$  and is demonstrated in [Figure 4-18](#).

The scrambling rules are as follows:

- The two bits of the Sync Header are not scrambled and do not advance the LFSR.
- All 16 Symbols of an Electrical Idle Exit Ordered Set (EIEOS) bypass scrambling. The scrambling LFSR is initialized after the last Symbol of an EIEOS is transmitted, and the descrambling LFSR is initialized after the last Symbol of an EIEOS is received.
- TS1 and TS2 Ordered Sets:
  - Symbol 0 of a TS1 or TS2 Ordered Set bypasses scrambling.
  - Symbols 1-13 are scrambled.
  - Symbols 14 and 15 bypass scrambling if required for DC Balance, but they are scrambled if not required for DC Balance.
- All 16 Symbols of a Fast Training Sequence (FTS) Ordered Set bypass scrambling.
- All 16 Symbols of a Start of Data Stream (SDS) Ordered Set bypass scrambling.
- All 16 Symbols of an Electrical Idle Ordered Set (EIOS) bypass scrambling.
- All Symbols of a SKP Ordered Set bypass scrambling.
- Transmitters advance their LFSR for all Symbols of all Ordered Sets except for the SKP Ordered Set. The LFSR is not advanced for any Symbols of a SKP Ordered Set.

- Receivers evaluate Symbol 0 of Ordered Set Blocks to determine whether to advance their LFSR. If Symbol 0 of the Block is SKP (see [Section 4.2.7.2](#) ), then the LFSR is not advanced for any Symbol of the Block. Otherwise, the LFSR is advanced for all Symbols of the Block.
- All 16 Symbols of a Data Block are scrambled and advance the scrambler.
- For Symbols that need to be scrambled, the least significant bit is scrambled first and the most significant bit is scrambled last.
- The seed value of the LFSR is dependent on the Lane number assigned to the Lane when the Link first entered Configuration.Idle (i.e., having gone through Polling from Detect with [LinkUp = 0b](#)).
  - The seed values for Lane number modulo 8 are:

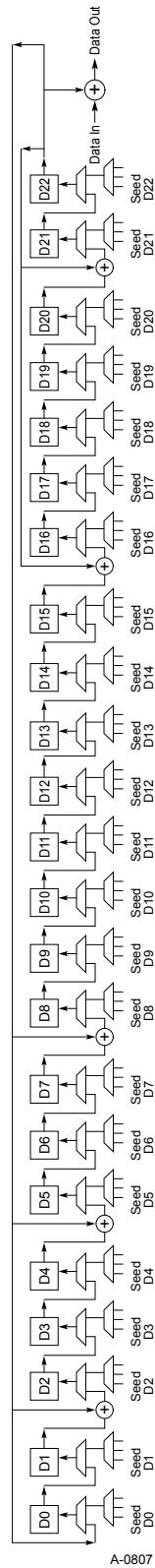
Lane	Seed
0	1DBFBCh
1	0607BBh
2	1EC760h
3	18C0DBh
4	010F12h
5	19CFC9h
6	0277CEh
7	1BB807h

## IMPLEMENTATION NOTE

### Scrambling Pseudo-code

The pseudo-code for the scrambler along with examples is provided in [Section C.2 of Appendix C](#).

- The seed value of the LFSR does not change while [LinkUp=1](#). Link reconfiguration through the LTSSM Configuration state does not modify the initial Lane number assignment as long as the [LinkUp](#) remains 1 (even though the Lane assignment may change during Configuration).
- Scrambling cannot be disabled in Configuration.Complete when using 128b/130b encoding.
- A [Loopback Slave](#) must not descramble or scramble the looped-back bit stream.



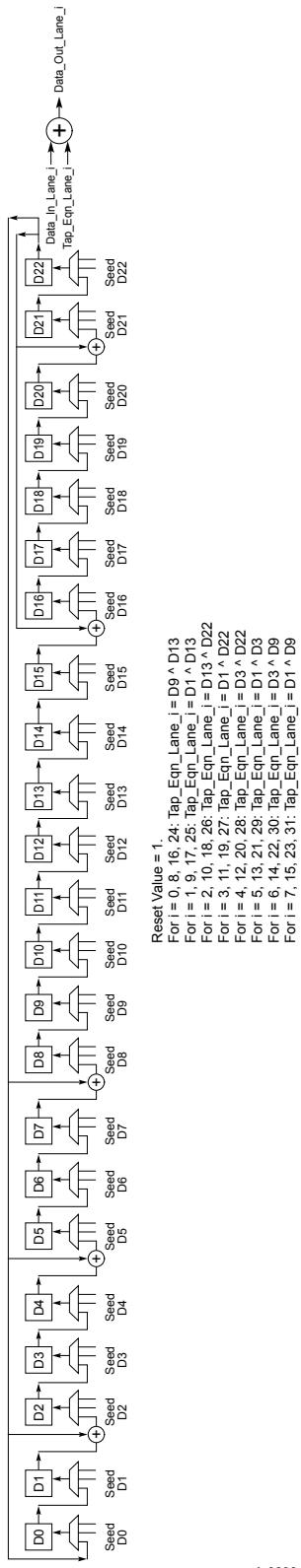
A-0807

Figure 4-18 LFSR with Scrambling Polynomial in 8.0 GT/s and Above Data Rate

## IMPLEMENTATION NOTE

### LFSR Implementation with a Shared LFSR

Implementations may choose to implement one LFSR and take different tap points as shown in Figure 4-19, which is equivalent to the individual LFSR per-lane with different seeds, as shown in Figure 4-18. It should also be noted that the tap equations of four Lanes are the XOR of the tap equations of two neighboring Lanes. For example, Lane 0 can be obtained by XORing the output of Lanes 1 and 7; Lane 2 is the XOR of Lanes 1 and 3; Lane 4 is the XOR of Lanes 3 and 5; and Lane 6 is the XOR of Lanes 5 and 7. This can be used to help reduce the gate count at the expense of potential delay due to the XOR results of the two Lanes.



A-0808

Figure 4-19 Alternate Implementation of the LFSR for Descrambling

#### 4.2.2.5 Precoding

A Receiver may request precoding from its transmitter for operating at data rates of 32.0 GT/s or higher. The precoding rules are as follows:

- A Port or Pseudo-Port must request precoding on all configured Lanes of the Link. Behavior is undefined if precoding is requested on some Lanes but not others by a Port or Pseudo-Port.
- A Port or Pseudo-Port may request precoding independent of other Ports or Pseudo-Ports. For example, it is possible that precoding may be turned on only in the Upstream Port in the case with no Retimers in [Figure 4-35](#), or on all the Lanes in Tx(A) and Tx(E) in the two Re-timer example in [Figure 4-35](#).
- Precoding is turned off for all data rates 32.0 GT/s and higher when the LTSSM is in the Detect state.
- Precoding request, if any, must be made prior to entering the 32.0 GT/s or higher data rate by setting the appropriate bit in the [EQ TS2 Ordered Sets](#) or the [128b/130b EQ TS2 Ordered Sets](#). A precoding request must be made by setting Transmitter Precode Request in the [EQ TS2](#) or [128b/130b EQ TS2 Ordered sets](#) prior to the transition to [Recovery.Speed](#) for the target data rate where the precoding will be turned on. For each data rate above 32.0 GT/s, the precoding request must be made independently.
- If the Link operates at 32.0 GT/s or higher data rate without performing equalization through the [No Equalization Needed](#) mechanism it negotiation in the (modified) TS1/TS2 Ordered Sets, the precoding requests from the last equalization results that are being used must be enforced. Thus each (pseudo) Port must store the precoding request along with the Tx Eq values in each Lanes, if it advertises the [No Equalization Needed](#) mechanism. If no equalization has ever been performed on the Link (prior to the current Link up), then precoding will not be turned on.
- If Transmitter Precode Request is set to 1b in each of the received eight consecutive EQ TS2 or [128b/130b EQ TS2 Ordered Sets](#) during [Recovery.RcvrCfg](#) prior to entry to [Recovery.Speed](#), the Transmitter must turn on the precoding for the target data rate at which the Link will operate on exit from [Recovery.Speed](#) if the target data rate is 32.0 GT/s or higher. Once turned on, the precoding will be in effect for that target data rate until the Transmitter receives another set of eight consecutive EQ TS2 or [128b/130b EQ TS2 Ordered Sets](#) during [Recovery.RcvrCfg](#) prior to entry to [Recovery.Speed](#) for the same target data rate.
- A Transmitter must not turn on precoding for any data rates lower than 32.0 GT/s.
- For data rates of 32.0 GT/s or higher, a Transmitter must set the [Transmitter Precoding On](#) bit in the TS1 Ordered Set in Recovery state to 1b if the precoding is on; else the bit must be set to 0b.
- Only scrambled bits are precoded, when precoding is turned on.
- The “previous bit” used for precoding is set to 1b on every block boundary and gets updated by the last scrambled and precoded bit transmitted within the current block boundary.
- When precoding is turned on, for symbols that are scrambled, Receivers must first decode the precoded bits before sending them to the descrambler.
- A Transmitter that has turned on precoding for 32.0 GT/s data rate on Lane 0, must set the [Transmitter Precoding On](#) bit to 1b in the [32.0 GT/s Status Register](#); else it must set the bit to 0b. A Receiver that has requested or will request to turn on precoding at 32.0 GT/s data rate, must set [Transmitter Precode Request](#) to 1b in the [32.0 GT/s Status Register](#); else it must set the bit to 0b.

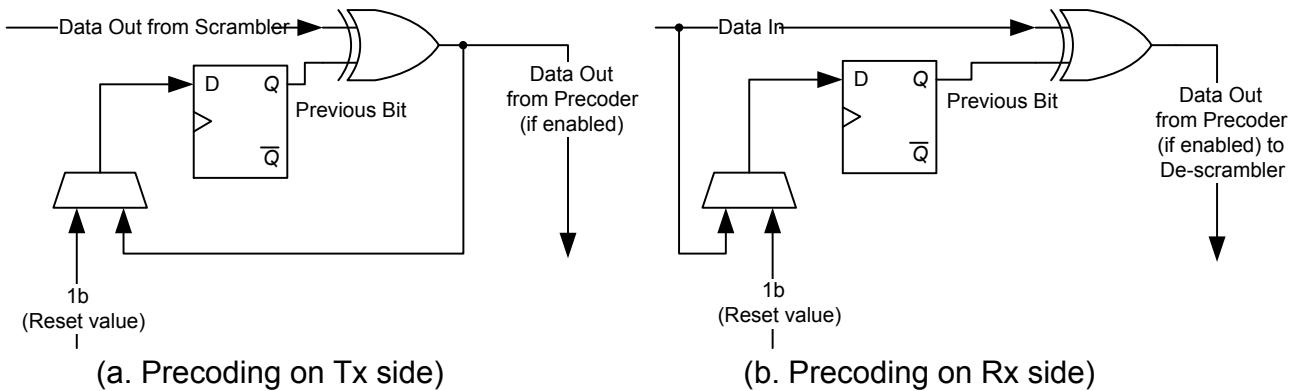


Figure 4-20 Precoding working the scrambler/de-scrambler

## IMPLEMENTATION NOTE

### Parity in the SKP Ordered Set when Precoding is turned on

As per the rules of [Section 4.2.4.1](#) and [Section 4.2.7.2](#), when precoding is turned on, the parity in the SKP Ordered Sets should be calculated before precoding is applied on the Transmit side. Thus, the order in the Transmitter is:

1. scrambling,
2. followed by parity bit calculation,
3. followed by precoding for the scrambled bits.

Accordingly, in the Receiver, the order is:

1. precoding (if turned on by the Transmitter),
2. followed by parity bit calculation,
3. followed by descrambling.

The rationale for this order is that in a Link with one or two Retimers, different Link segments may have the precoding on or off. Let us consider an example system with one retimer between the Root Port and End Point to illustrate this. In the upstream direction, the End Point has precoding on in its Transmitter Lanes since the Retimer Receiver needs it but the Retimer to Root Port Link segment has the precoding off since the Root Port does not need precoding at its Receiver. Since the Retimer does not change the parity bit, the Root Port would get a parity error if the parity calculation was done by the Transmitter (of the End Point) after precoding.

## IMPLEMENTATION NOTE

### Loopback Master behavior if Precoding is on in any Link Segment

As per the rules of precoding mentioned in this section and [Section 4.2.6.4](#), a Loopback Master operating at a data rate of 32.0 GT/s or higher should account for precoding to be on on some link segments and off in other link segments. This is particularly relevant when the Loopback Slave transitions from sending TS1 Ordered Sets to looping back the bits. This is where the precoding on the receiver of the Loopback Master may switch (between precoding on and off). The Loopback Master is permitted to use implementation specific mechanisms to handle this scenario.

## IMPLEMENTATION NOTE

### TS1/TS2 Ordered Sets when Precoding is turned on

As per the rules in this section, when precoding is turned on, the ‘previous bit’ used for precoding is 1b for the first bit of Symbol 1 since Symbol 0 is not scrambled and the ‘previous bit’ gets set to 1b at the block boundary.

#### **4.2.2.6 Loopback with 128b/130b Code**

When using 128b/130b encoding, Loopback Masters must transmit Blocks with the defined 01b and 10b Sync Headers. However, they are not required to transmit an SDS Ordered Set when transitioning from Ordered Set Blocks to Data Blocks, nor are they required to transmit an EDS Token when transitioning from Data Blocks to Ordered Set Blocks. Masters must transmit SKP Ordered Sets periodically as defined in [Section 4.2.7](#), and they must be capable of processing received (looped-back) SKP Ordered Sets of varying length. Masters are permitted to transmit Electrical Idle Exit Ordered Sets (EIEOS) as defined in [Section 4.2.2.2.1](#). Masters are permitted to transmit any payload in Data Blocks and Ordered Set Blocks that they expect to be looped-back. If the Loopback Master transmits an Ordered Set Block whose first symbol matches the first symbol of SKP OS, EIEOS, or EOS, that Ordered Set Block must be a complete and valid SKP OS, EIEOS, or EOS.

When using 128b/130b encoding, Loopback Slaves must retransmit all bits received without modification, except for SKP Ordered Sets which can be adjusted as needed for clock compensation. If clock compensation is required, slaves must add or remove 4 SKP Symbols per Ordered Set. The modified SKP Ordered Set must meet the definition of [Section 4.2.7.2](#) (i.e., it must have between 4 to 20 SKP Symbols followed by the SKP\_END Symbol and the three Symbols that follow it as transmitted by the Loopback Masters). If a slave is unable to obtain Block alignment or it is misaligned, it may be unable to perform clock compensation and therefore unable to loop-back all bits received. In this case, it is permitted to add or remove Symbols as necessary to continue operation. Slaves must not check for a received SDS Ordered Set when a transition from Ordered Set Blocks to Data Blocks is detected, and they must not check for a received EDS Token when a transition from Data Blocks to Ordered Set Blocks is detected.

#### **4.2.3 Link Equalization Procedure for 8.0 GT/s and Higher Data Rates**

The Link equalization procedure enables components to adjust the Transmitter and the Receiver setup of each Lane to improve the signal quality and meet the requirements specified in [Chapter 8](#), when operating at 8.0 GT/s and higher data rates. All the Lanes that are associated with the LTSSM (i.e., those Lanes that are currently operational or may be

operational in the future due to Link Upconfigure) must participate in the equalization procedure. The procedure must be executed during the first data rate change to any data rate at 8.0 GT/s or above, unless all components in the Link have advertised that no equalization is needed. Components must arrive at the appropriate Transmitter setup for all the operating conditions and data rates that they will encounter in the future when LinkUp=1b. Components must not require that the equalization procedure be repeated at any data rate for reliable operation, although there is provision to repeat the procedure. Components must store the Transmitter setups that were agreed to during the equalization procedures and use them for future operation at 8.0 GT/s and higher data rates. Components are permitted to fine-tune their Receiver setup even after the equalization procedure is complete as long as doing so does not cause the Link to be unreliable (i.e., does not meet the requirements in Chapter 8 ) or go to Recovery.

The Link equalization procedure is not required for any data rates and can be completely bypassed if all components in the Link have advertised that no equalization is needed in its TS1/TS2 Ordered Sets or modified TS1/TS2 Ordered Sets (see Table 4-6, Table 4-7, and Table 4-8 ). A component may choose to advertise that it does not need equalization at any rates above 5.0 GT/s if it supports 32.0 GT/s or higher data rates and can either operate reliably with equalization settings stored from a prior equalization procedure or does not need equalization for reliable operation.

The equalization procedure can be initiated either autonomously or by software. It is strongly recommended that components use the autonomous mechanism for all the data rates above 5.0 GT/s that they intend to operate in. However, a component that chooses not to participate in the autonomous mechanism for all the data rates above 5.0 GT/s must have its associated software ensure that the software based mechanism is applied to the data rates above 5.0 GT/s where the autonomous mechanism was not applied, prior to operating at that data rate.

Normally, equalization is performed at a higher data rate only if equalization has successfully completed at all lower data rates above 5.0 GT/s. For example, a Link will complete equalization successfully at 8.0 GT/s, followed by 16.0 GT/s, followed by 32.0 GT/s. However, an optional mechanism to skip over equalization to the highest common data rate of 32.0 GT/s or higher is permitted if all components support data rates of 32.0 GT/s or higher and the mechanism is supported by all components in the Link, as advertised in the TS1/TS2 Ordered sets or modified TS1/TS2 Ordered Sets. When this optional mechanism is enabled and successfully negotiated between the components, equalization is not performed at any other rate except the highest common data rate. For all the data rates above 5.0 GT/s where equalization is not performed, the expectation is that the Link must not operate in those data rates. For example, a Link may train to L0 in 2.5 GT/s, enter Recovery and perform equalization at 32.0 GT/s, skipping equalization at 8.0 GT/s and 16.0 GT/s. In this case, the intended data rates of operation of the Link are 2.5 GT/s, 5.0 GT/s, or 32.0 GT/s. If the equalization procedure at the highest common data rate is unsuccessful even after re-equalization attempts and the Link needs to equalize at lower data rates, the Downstream Port must stop advertising Equalization bypass to highest rate support and ensure that the Link returns to operation at 2.5 GT/s or 5.0 GT/s. The required equalization procedures are then performed as they would have been if the optional mechanism to skip over equalization to the highest common data rate was never supported. If the equalization procedure at the lower data rates is driven by software, it must set the Equalization bypass to highest rate Disable and No Equalization Needed Disable register bits to 1b each; set the target Link speed such that the Link will be operational at 2.5 GT/s or 5.0 GT/s; and then set the target Link speed to equalize at the lower rates starting with 8.0 GT/s onwards. A port must not advertise the Equalization bypass to highest rate support if the Equalization bypass to highest rate Disable bit is set to 1b.

Another optional mechanism to skip the entire equalization process and go directly to the highest common data rate of 32.0 GT/s or higher is permitted if all components support data rates of 32.0 GT/s or higher and the No Equalization Needed mechanism is supported by all components in the Link, as advertised in the TS1/TS2 or modified TS1/TS2 Ordered sets. This is done if a component is either able to retrieve the equalization and other circuit settings at all the data rates from a prior equalization that will work for the component or it does not need equalization at all in the all data rates above 5.0 GT/s. A component must not advertise this capability if the Equalization bypass to highest rate Disable bit is set to 1b.

If one direction of the Link is advertising No Equalization Needed and the other side is advertising Equalization bypass to highest rate support in the TS1/TS2 Ordered Sets, the Link will operate in the Equalization bypass to highest rate support since the No Equalization Needed bit also indicates that the device is capable of bypassing Equalization to the highest data rate. In the modified TS1/TS2 Ordered Sets, a device that sets No Equalization Needed bit to 1b must also set the Equalization bypass to highest rate support to 1b. If one direction of the Link is advertising No Equalization Needed bit

and the other side is advertising Equalization bypass to highest rate support only in the modified TS1/TS2 Ordered Sets, the Link will operate in the Equalization bypass to highest rate support. Link operation is undefined if a device advertises No Equalization Needed bit as 1b and Equalization bypass to highest rate support to 0b in the modified TS1/TS2 Ordered Sets it transmits.

The autonomous mechanism is executed if both components advertise that they are capable of at least the 8.0 GT/s data rate (via the TS1 and TS2 Ordered Sets) during the initial Link negotiation (when LinkUp is set to 1b) and the Downstream Port chooses to perform the equalization procedure at the intended data rates of operation above 5.0 GT/s. While not recommended, the Downstream Port may choose to perform the autonomous mechanism only on a subset of the intended data rates of operation above 5.0 GT/s. In that case, the software based mechanism must be executed in order to perform the equalization procedure for the intended data rates of operation above 5.0 GT/s, not covered by the autonomous mechanism. For example, if both components advertised 8.0 GT/s, 16.0 GT/s, and 32.0 GT/s Data Rates but autonomous equalization was performed for only 8.0 GT/s and 16.0 GT/s Data Rates, then software based mechanism must be adopted for equalization at 32.0 GT/s Data Rate.

In the autonomous mechanism, after entering L0, irrespective of the current Link speed, neither component must transmit any DLLP if the equalization procedure must be performed and until the equalization procedure completes. The equalization procedure is considered complete once the Transmitter and Receiver setup of each Lane has been adjusted for each common data rate supported above 5.0 GT/s for which the Downstream Port intends to perform equalization using the autonomous mechanism. The Downstream Port is required to initiate the speed change to the data rate where the equalization needs to be performed. During any equalization (autonomous or software initiated or re-equalization), the Downstream Port must not advertise support for any data rate above the data rate for which equalization needs to be performed in Recovery. The following example is provided to illustrate the equalization flow.

Example: Consider a Link where equalization needs to be performed autonomously at 8.0 GT/s, and 16.0 GT/s. The Downstream Port enters Recovery to perform equalization at 8.0 GT/s by not advertising any data rates above 8.0 GT/s. The 8.0 GT/s equalization procedure is deemed to have been successfully executed if the Equalization 8.0 GT/s Phase 3 Successful bit and Equalization 8.0 GT/s Complete bit of the Link Status 2 register are both set to 1b. Immediately following the transition from Recovery to L0, after the initial data rate change to 8.0 GT/s, the Downstream Port is required to transition from L0 to Recovery, advertise 16.0 GT/s data rate support (but not advertise support for 32.0 GT/s, even if it is capable of supporting 32.0 GT/s), change the data rate to 16.0 GT/s and perform the 16.0 GT/s equalization procedure.

If the Downstream Port detects equalization problems or the Upstream Port made an equalization redo request (by setting the Request Equalization bit to 1b) the Downstream Port may redo equalization prior to proceeding to operate at the data rate where the equalization failed or performing equalization at a higher data rate. The number of back-to-back equalization redos at a given data rate is implementation specific but must be finite. If at the conclusion of the initial or subsequent equalization process and the execution of an implementation specific number of equalization redo's, the Link is not able to operate reliably at the data rate where equalization was performed, then it must revert back to a lower data rate of operation.

Components using the autonomous mechanism must not initiate any autonomous Link width downsizing until the equalization procedure completes. An Upstream Port must not transmit any DLLP until it receives a DLLP from the Downstream Port. If the Downstream Port performs equalization again, it must not transmit any DLLP until it completes the equalization procedure. A Downstream Port may perform equalization again based on its own needs or based on the request from the Upstream Port, if it can meet its system requirements. Executing equalization multiple times may interfere with software determination of Link and device status, as described in Section 6.6.

## IMPLEMENTATION NOTE

### DLLP Blocking During Autonomous Equalization

When using the autonomous mechanism for equalization at 8.0 GT/s or higher data rates, the Downstream Port is required to block the transmission of DLLPs until equalization has completed, and the Upstream Port is required to block the transmission of DLLPs until a DLLP is received from the Downstream Port. If both components advertise that they are capable of the 16.0 GT/s (or 32.0 GT/s) data rate but the Downstream Port only uses the autonomous mechanism for equalization at 8.0 GT/s, the Downstream Port is only required to block DLLP transmission until 8.0 GT/s equalization has completed. Similarly, if both components advertise that they are capable of the 32.0 GT/s data rate but the Downstream Port only uses the autonomous mechanism for equalization at 16.0 GT/s, the Downstream Port is only required to block DLLP transmission until 16.0 GT/s equalization has completed. If the Downstream Port delays entering Recovery from L0 while DLLP transmission is blocked, either the L0 Inferred Electrical Idle timeout (see Section 4.2.4.4) or the DLLP timeout (see Section 2.6.1.2) may expire in the Upstream or Downstream Ports. If either of these two timeouts occurs, it will result in the initiation of an entry to Recovery to perform Link retraining. Neither of these two timeouts is a reportable error condition, and the resulting Link retraining has no impact on proper Link operation.

When using the software based mechanism, software must guarantee that there will be no side-effects for transactions in flight (e.g., no timeout), if any, due to the Link undergoing the equalization procedure. Software can write 1b to the Perform Equalization bit in the Link Control 3 Register, followed by a write to the Target Link Speed field in the Link Control 2 register to enable the Link to run at 8.0 GT/s or higher, followed by a write of 1b to the Retrain Link bit in the Link Control register of the Downstream Port to perform equalization. Software must not enable the Link to run at a data rate above 8.0 GT/s during a software initiated equalization procedure if the equalization procedure at all the lower data rates starting with 8.0 GT/s have not been successfully executed and the Link is not capable of bypassing equalization to higher data rate(s) (i.e., either Equalization bypass to highest rate Supported is 0b or Equalization bypass to highest rate Disable is 1b). The equalization procedure is deemed successful as follows for the following data rates:

- 8.0 GT/s: Equalization 8.0 GT/s Phase 3 Successful bit and Equalization 8.0 GT/s Complete bit of the Link Status 2 register are both set to 1b;
- 16.0 GT/s: Equalization 16.0 GT/s Phase 3 Successful bit and Equalization 16.0 GT/s Complete bit of the 16.0 GT/s Status Register are both set to 1b.

Software may set the Hardware Autonomous Width Disable of the Link Control register in both components or use some other mechanism to ensure that the Link is in its full functional width prior to setting the Perform Equalization bit in the Downstream Port. The component that had initiated the autonomous width downsizing is responsible to upconfigure the Link to go to its full functional width by initiating the transition to Recovery and Configuration within 1 ms of the Hardware Autonomous Width Disable bit being set to 1b. If an Upstream Port does not advertise the 8.0 GT/s data rate, the 16.0 GT/s data rate, or the 32.0 GT/s data rate initially and did not participate in the autonomous equalization mechanism for the non-advertised rates, its associated software must ensure there will be no side-effects for transactions in flight, if any, during equalization, before it instructs the Upstream Port to go to Recovery and advertise the previously non-advertised data rates and initiate a speed change. The Downstream Port subsequently initiates the equalization procedure during the initial speed change to the data rate advertised by the Upstream Port when it transitions to Recovery.

Upstream Ports are required to check for equalization setting problems in the Recovery.RcvrLock state (see Section 4.2.6.4.1). However, both Downstream and Upstream Ports are permitted to use implementation-specific methods to detect equalization problems at any time. A Port that detects a problem with its equalization settings is required to undertake the following actions, for each the following data rates:

- 8.0 GT/s: Link Equalization Request 8.0 GT/s bit in the Link Status 2 register is set to 1b;

- 16.0 GT/s: Link Equalization Request 16.0 GT/s bit in the 16.0 GT/s Status Register is set to 1b
- 32.0 GT/s: Link Equalization Request 32.0 GT/s bit in its 32.0 GT/s Status Register is set to 1b.

In addition to setting the appropriate Link Equalization Request bit to 1b, an Upstream Port must initiate a transition to Recovery (if necessary) and request equalization at the appropriate data rate in the Recovery.RcvrCfg state by setting the Request Equalization bit of its transmitted TS2 Ordered Sets to 1b and the Equalization Request Data Rate bits to the data rate of the detected problem. If it requests equalization, it must request equalization for each detected problem only once. When requesting equalization, the Upstream Port is also permitted, but not required, to set the Quiesce Guarantee bit to 1b to inform the Downstream Port that an equalization process initiated within 1 ms will not cause any side-effects to its operation.

When a Downstream Port receives an equalization request from an Upstream Port (when it is in the Recovery.RcvrCfg state and receives 8 consecutive TS2 Ordered Sets with the Request Equalization bit set to 1b), it must either initiate an equalization process at the requested data rate (as defined by the received Equalization Request Data Rate bits) within 1 ms of completing the next Recovery to L0 transition, or it must set the appropriate Link Equalization Request 8.0 GT/s in its Link Status 2 register or Link Equalization Request 16.0 GT/s bit in its 16.0 GT/s Status Register or Link Equalization Request 32.0 GT/s bit in its 32.0 GT/s Status Register. It should initiate an equalization process only if it can guarantee that executing the equalization process will not cause any side-effects to either its operation or the Upstream Port's operation. The Downstream Port is permitted, but not required, to use the received Quiesce Guarantee bit to determine the Upstream Port's ability to execute an equalization process without side-effects.

If a Downstream Port wants to initiate an equalization process and can guarantee that it will not cause side-effects to its own operation but is unable to directly determine whether the equalization process will cause side-effects to the Upstream Port's operation, then it is permitted to request that the Upstream Port initiate an equalization request. The Downstream Port does so by transitioning to Recovery and in the Recovery.RcvrCfg state setting the Request Equalization bit of its transmitted TS2 Ordered Sets to 1b, the Equalization Request Data Rate bits to the desired data rate, and the Quiesce Guarantee bit to 1b. When an Upstream Port receives such an equalization request from a Downstream Port (when it is in the Recovery.RcvrCfg state and receives 8 consecutive TS2 Ordered Sets with the Request Equalization and Quiesce Guarantee bits set to 1b), it is permitted, but not required, to quiesce its operation and prepare to execute an equalization process at the data rate requested by the Downstream Port, and then request equalization at that same data rate (using the method described previously for reporting equalization setting problems) and with the Quiesce Guarantee bit set to 1b. There is no time limit on how long the Upstream Port can take to respond, but it should attempt to do so as quickly as possible. If a Downstream Port makes a request and receives such a response from the Upstream Port, then it must either initiate an equalization process at the agreed-upon data rate within 1 ms of completing the next Recovery to L0 transition if it can still guarantee that executing the equalization process will not cause any side-effects to its operation, or it must set the appropriate Link Equalization Request 8.0 GT/s in its Link Status 2 register or Link Equalization Request 16.0 GT/s bit in its 16.0 GT/s Status Register or Link Equalization Request 32.0 GT/s bit in its 32.0 GT/s Status Register.

## IMPLEMENTATION NOTE

### Using Quiesce Guarantee Mechanism

Side-effects due to executing equalization after the Data Link Layer is in DL\_Active can occur at the Port, Device, or system level. For example, the time required to execute the equalization process could cause a Completion Timeout error to occur - possibly in a different system component. The Quiesce Guarantee information can help Ports decide whether to execute a requested equalization or not.

A component may operate at a lower data rate after reporting its equalization problems, either by timing out through Recovery.Speed or by initiating a data rate change to a lower data rate. Any data rate change required to perform the equalization procedure is exempt from the 200 ms requirement in Section 6.11. Table 4-3 describes the mechanism for performing redo Equalization. Sometimes it may be necessary to perform a speed change to an intermediate data rate to

redo equalization. For example, if the Downstream Port wants to redo equalization at 16.0 GT/s, bypass equalization is not supported, and the current data rate is either 2.5 GT/s or 5.0 GT/s, the Downstream Port must first initiate a speed change to 8.0 GT/s (the 8.0 GT/s equalization procedure will not be executed unless necessary) from which it will launch the redo equalization for 16.0 GT/s. The equalization procedure can be performed at most once in each trip through the Recovery state.

*Table 4-3 Equalization requirements under different conditions*

From 2.5 GT/s or 5.0 GT/s to 8.0 GT/s Equalization	<p>The mechanisms described here are identical for all flavors of equalization: initial or redo equalization; autonomous or software initiated.</p> <p>The Downstream Port communicates the Transmitter preset values and the Receiver preset hints, if applicable, for each Lane to the Upstream Port using 8b/10b encoding. These values are communicated using the <u>EQ TS2 Ordered Sets</u> (defined in <u>Section 4.2.4.1</u>) in <u>Recovery.RcvrCfg</u>, when a data rate change to the higher data rate has been negotiated, prior to transitioning to the higher data rate to perform equalization. The preset values sent in the <u>EQ TS2 Ordered Sets</u> are derived as follows:</p> <ul style="list-style-type: none"> <li>• For equalization at 8.0 GT/s: <u>Upstream Port 8.0 GT/s Transmitter Preset</u> and <u>Upstream Port 8.0 GT/s Receiver Preset Hint</u> fields of each <u>Lane Equalization Control Register Entry</u>.</li> </ul>
From 8.0 GT/s to 16.0 GT/s equalization OR from 16.0 GT/s to 32.0 GT/s equalization	<p>After the data rate change to the higher data rate where equalization needs to be performed, the Upstream Port transmits TS1 Ordered Sets with the preset values it received. The preset values must be within the operable range defined in <u>Section 8.3.3.3</u> if reduced swing will be used by the Transmitter.</p> <p>After the data rate change to the higher data rate where equalization needs to be performed, the Downstream Port transmits TS1 Ordered Sets with the preset values as follows with the assumption that the preset values must be within the operable range defined in <u>Section 8.3.3.3</u> if reduced swing will be used by the Transmitter:</p> <ul style="list-style-type: none"> <li>• For equalization at 8.0 GT/s: <u>Downstream Port 8.0 GT/s Transmitter Preset</u> and optionally <u>Downstream Port 8.0 GT/s Receiver Preset Hint</u> fields of each <u>Lane Equalization Control Register Entry</u>.</li> </ul> <p>To perform redo equalization, the Downstream Port must request speed change through <u>EQ TS1 Ordered Sets</u> in <u>Recovery.RcvrLock</u> at 2.5 GT/s or 5.0 GT/s to inform the Upstream Port that it intends to redo equalization at the higher data rate. An Upstream Port should advertise the higher data rate in <u>Recovery</u> if it receives <u>EQ TS1 Ordered Sets</u> with speed change bit set to 1b and if it intends to operate at the higher data rate in the future.</p>

- For equalization at 16.0 GT/s: Downstream Port 16.0 GT/s Transmitter Preset field of the 16.0 GT/s Lane Equalization Control Register Entry corresponding to the Lane.
- For equalization at 32.0 GT/s: Downstream Port 32.0 GT/s Transmitter Preset field of the 32.0 GT/s Lane Equalization Control Register Entry corresponding to the Lane.

The preset values must be within the operable range defined in Section 8.3.3.3 if reduced swing will be used by the Transmitter.

To perform redo equalization, the Downstream Port must request speed change through TS1 Ordered Sets in Recovery.RcvrLock with the Equalization Redo bit set to 1b to inform the Upstream Port that it intends to redo equalization. An Upstream Port should advertise the higher data rate in Recovery if it receives TS1 Ordered Sets with speed change bit set to 1b, Equalization Redo bit set to 1b and it intends to operate at the higher data rate in the future.

From  
2.5 GT/s or  
5.0 GT/s to  
32.0 GT/s  
Equalization

Equalization to 32.0 GT/s or higher data rate from 2.5 GT/s or 5.0 GT/s is possible only if the Link is capable of bypassing equalization to higher data rate(s) (i.e., Equalization bypass to highest rate Supported in 32.0 GT/s Capabilities register is 1b and Equalization bypass to highest rate Disable in the 32.0 GT/s Control Register is 0b).

The mechanisms described here are identical for all flavors of equalization: initial or redo equalization; autonomous or software initiated.

The Downstream Port communicates the Transmitter preset values and the Receiver preset hints, if applicable, for each Lane to the Upstream Port using 8b/10b encoding. These values are communicated using the EQ TS2 Ordered Sets (defined in Section 4.2.4.1) in Recovery.RcvrCfg, when a data rate change to the higher data rate has been negotiated, prior to transitioning to the higher data rate to perform equalization. The preset values sent in the EQ TS2 Ordered Sets are derived as follows:

- For equalization at 32.0 GT/s: Upstream Port 32.0 GT/s Transmitter Preset field of the 32.0 GT/s Lane Equalization Control Register Entry corresponding to the Lane. The Receiver Preset Hint field must be set to 000b.

After the data rate change to the higher data rate where equalization needs to be performed, the Upstream Port transmits TS1 Ordered Sets with the preset values it received. The preset values must be within the operable range defined in Section 8.3.3.3 if reduced swing will be used by the Transmitter.

After the data rate change to the higher data rate where equalization needs to be performed, the Downstream Port transmits TS1 Ordered Sets with the preset values as follows with the assumption that the preset values must be within the operable range defined in Section 8.3.3.3 if reduced swing will be used by the Transmitter:

- For equalization at 32.0 GT/s: Downstream Port 32.0 GT/s Transmitter Preset field of the 32.0 GT/s Lane Equalization Control Register Entry corresponding to the Lane.

To perform redo equalization, the Downstream Port must request speed change through EQ TS1 Ordered Sets in Recovery.RcvrLock at 2.5 GT/s or 5.0 GT/s to inform the Upstream Port that it intends to redo equalization at the higher data rate. An Upstream Port should advertise the higher data rate in Recovery if it receives EQ TS1 Ordered Sets with speed change bit set to 1b and it intends to operate at the higher data rate in the future.

Equalization  
at a data  
rate from a  
data rate  
equal to the  
target  
equalization  
data rate

This is only possible with a redo equalization. The combinations covered here are: 8.0 GT/s equalization from 8.0 GT/s data rate, 16.0 GT/s equalization from 16.0 GT/s data rate, and 32.0 GT/s equalization from 32.0 GT/s data rate.

In this case, the initial preset used during equalization is equal to the initial preset used during the last time the equalization was performed at the data rate where equalization is being performed.

The equalization procedure consists of up to four Phases, as described below. When operating at 8.0 GT/s or higher data rates, the Phase information is transmitted using the Equalization Control (EC) field in the TS1 Ordered Sets.

### **Phase 0**

This phase is executed while negotiating (and prior to) to the data rate where equalization would be performed. The preset to be used for equalization is determined as described in Table 4-3 .

### **Phase 1**

Both components make the Link operational enough at the current data rate to be able to exchange TS1 Ordered Sets to complete the remaining phases for the fine-tuning of their Transmitter/Receiver pairs. It is expected that the Link will operate at a BER of less than  $10^{-4}$  before the component is ready to move on to the next Phase. Each Transmitter uses the preset values as described in Table 4-3 .

The Downstream Port initiates Phase 1 by transmitting TS1 Ordered Sets with EC=01b (indicating Phase 1). The Upstream Port, after adjusting its Receiver, if necessary, to ensure that it can progress with the equalization process, receives these TS1 Ordered Sets and transitions to Phase 1 (where it transmits TS1 Ordered Sets with EC=01b). The Downstream Port ensures that it can reliably receive the bit stream from the Upstream Port to continue through the rest of the Phases when it receives TS1 Ordered Sets from the Upstream Port with EC=01b before it moves on to Phase 2.

### **Phase 2**

In this Phase the Upstream Port adjusts the Transmitter setting of the Downstream Port along with its own Receiver setting, independently, on each Lane, to ensure it receives the bit stream compliant with the requirements in Chapter 8 (e.g., each operational Downstream Lane has a BER less than  $10^{-12}$ ). The Downstream Port initiates the move to Phase 2 by transmitting TS1 Ordered Sets with EC=10b to the Upstream Port. The Downstream Port advertises the Transmitter coefficients and the preset it is using per the rules below in Phase 1 for preset only and in Phase 2 for preset and coefficients. The Upstream Port receives these Ordered Sets and may request different coefficient or preset settings and continue to evaluate each setting until it arrives at the best setting for operating the Downstream Lanes. After the Upstream Port has completed this Phase, it moves the Link to Phase 3 by transmitting TS1 Ordered Sets with EC=11b to the Downstream Port.

### **Phase 3**

In this Phase the Downstream Port adjusts the Transmitter setting of the Upstream Port along with its own Receiver setting, independently, on each Lane, using a handshake and evaluation process similar to Phase 2 with the exception that EC=11b. The Downstream Port signals the end of Phase 3 (and the equalization procedure) by transmitting TS1 Ordered Sets with EC=00b.

The algorithm used by a component to adjust the transmitter of its Link partner and the evaluation of that Transmitter set-up with its Receiver set-up is implementation-specific. A component may request changes to any number of Lanes and can request different settings for each Lane. Each requested setting can be a preset or a set of coefficients that meets the requirements defined in Section 4.2.3.1 . Each component is responsible for ensuring that at the end of the fine-tuning (Phase 2 for Upstream Ports and Phase 3 for Downstream Ports), its Link partner has the Transmitter setting in each Lane that will cause the Link to meet the requirements in Chapter 8 .

A Link partner receiving the request to adjust its Transmitter must evaluate the request and act on it. If a valid preset value is requested and the Transmitter is operating in full-swing mode, it must be reflected in the Transmitter set-up and subsequently in the preset and coefficient fields of the TS1 Ordered Set that the Link partner transmits. If a preset value is requested, the Transmitter is operating in reduced-swing mode, and the requested preset is supported as defined in Section 8.3.3.3 it must be reflected in the Transmitter set-up and subsequently in the preset and coefficient fields of the TS1 Ordered Set that the Link partner transmits. Transmitters operating in reduced-swing mode are permitted to reject preset requests that are not supported as defined in Section 8.3.3.3 . A request for adjusting the coefficients may be accepted or rejected. If the set of coefficients requested for a Lane is accepted, it must be reflected in the Transmitter set-up and subsequently in the transmitted TS1 Ordered Sets. If the set of coefficients requested for a Lane is rejected, the Transmitter set-up is not changed, but the transmitted TS1 Ordered Sets must reflect the requested coefficients along with the Reject Coefficient bit set to 1b. In either case of responding to a coefficient request, the preset field of the

transmitted TS1 Ordered Sets is not changed from the last preset value that was transmitted. A request for adjusting the coefficients may be rejected by the Link partner only if the set of coefficients requested is not compliant with the rules defined in Section 4.2.3.1.

When performing equalization of a crosslink, the component that played the role of the Downstream Port during the earlier crosslink initialization at the lower data rate also assumes the responsibility of the Downstream Port for equalization.

If a Lane receives a Transmitter Preset value (from TS1 or TS2 sequences) with a Reserved or unsupported value in Polling.Compliance, Loopback, or Phase 0 or Phase 1 of Recovery.Equalization, then the Lane is permitted to use any supported Transmitter preset setting in an implementation-specific manner. The Reserved or unsupported Transmitter preset value is transmitted in any subsequent Compliance Patterns or Ordered Sets, and not the implementation-specific preset value chosen by the Lane. For example, if a Lane of an Upstream Port receives a Transmitter preset value 1111b (Reserved) with the EQ TS2 Ordered Sets it receives in Recovery.RcvrCfg, it is permitted to use any supported Transmitter preset value for its transmitter setting after changing the data rate to 8.0 GT/s, but it must transmit 1111b as its Transmitter preset value in the TS1 Ordered Sets it transmits in Phase 0 and Phase 1 of Recovery.Equalization.

In the Loopback state, the Loopback Master is responsible for communicating the Transmitter and Receiver settings it wants the Slave to use through the EQ TS1 Ordered Sets it transmits in the 2.5 GT/s or 5.0 GT/s data rate, and the preset or coefficient settings it wants the device under test to operate under in the TS1 Ordered Sets it transmits in the 8.0 GT/s or higher data rate. Similarly, if the Polling.Compliance state for 8.0 GT/s or higher Data Rates is entered through TS1 Ordered Sets, the entity that is performing the test is required to send the appropriate EQ TS1 Ordered Sets and coefficients for the device under test to operate with, according to the mechanism defined in Section 4.2.6.2.

## IMPLEMENTATION NOTE

### Equalization Example

The following diagram is an example illustrating how two devices may complete the equalization procedure. If the maximum common data rate supported by both Ports is 8.0 GT/s, the equalization procedure is complete at the conclusion of the 8.0 GT/s equalization procedure. If the maximum common data rate supported by both Ports is 16.0 GT/s, the 8.0 GT/s equalization procedure is followed by the 16.0 GT/s equalization procedure. If either the 8.0 GT/s or 16.0 GT/s equalization procedure is repeated and is performed while the Link is in 8.0 GT/s data rate (for the 8.0 GT/s equalization) or in 16.0 GT/s (for the 16.0 GT/s equalization), Phase 0 may be skipped since there is no need for the Link to go back to 2.5 GT/s or 5.0 GT/s (for the 8.0 GT/s equalization) or 8.0 GT/s (for the 16.0 GT/s equalization) to resend the same EQ TS2 Ordered Sets to convey the presets. A Downstream Port may choose to skip Phase 2 and Phase 3 if it determines that fine-tuning of the Transmitter is not needed based on the channel and components in the platform.

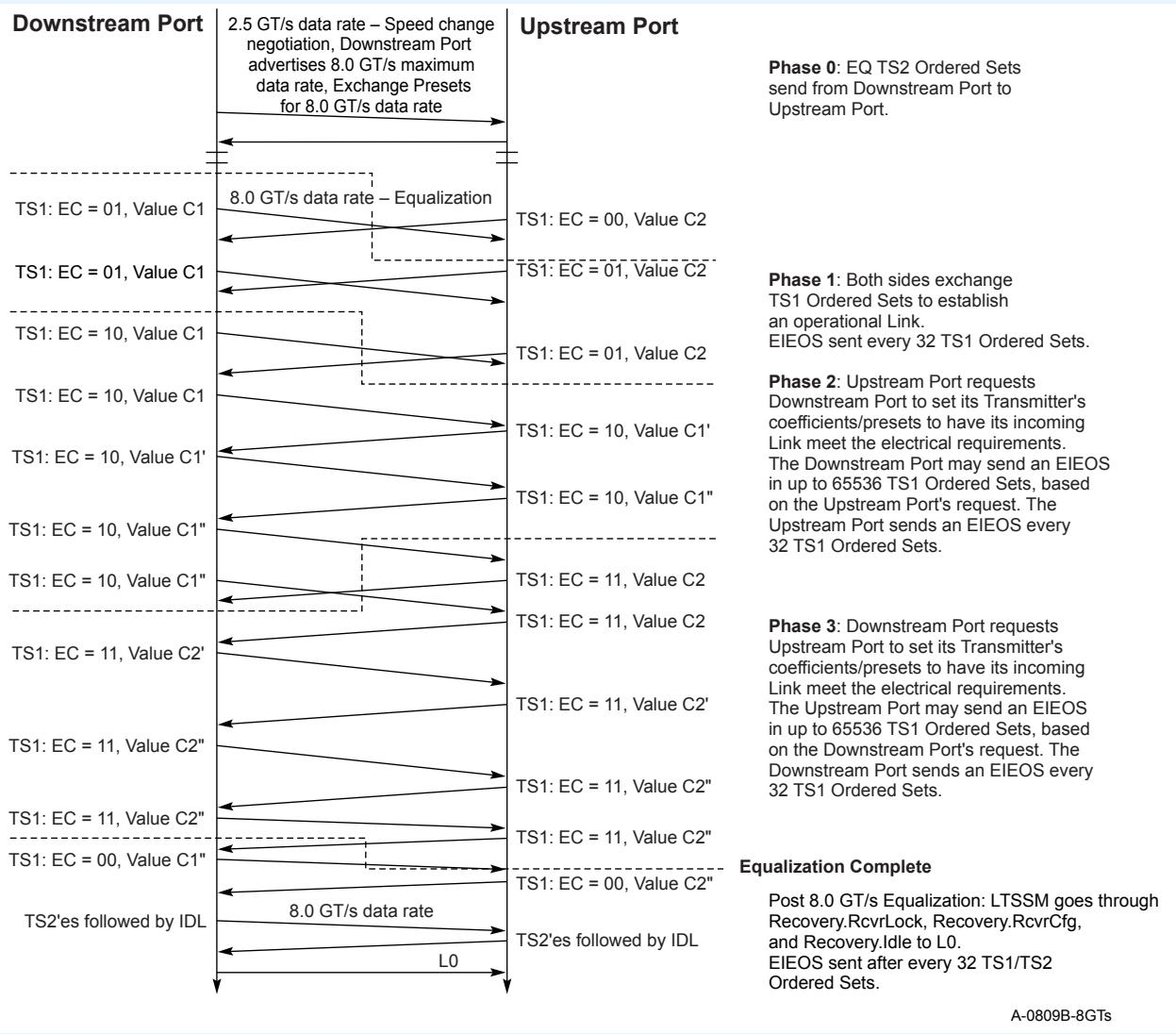


Figure 4-21 8.0 GT/s Equalization Flow

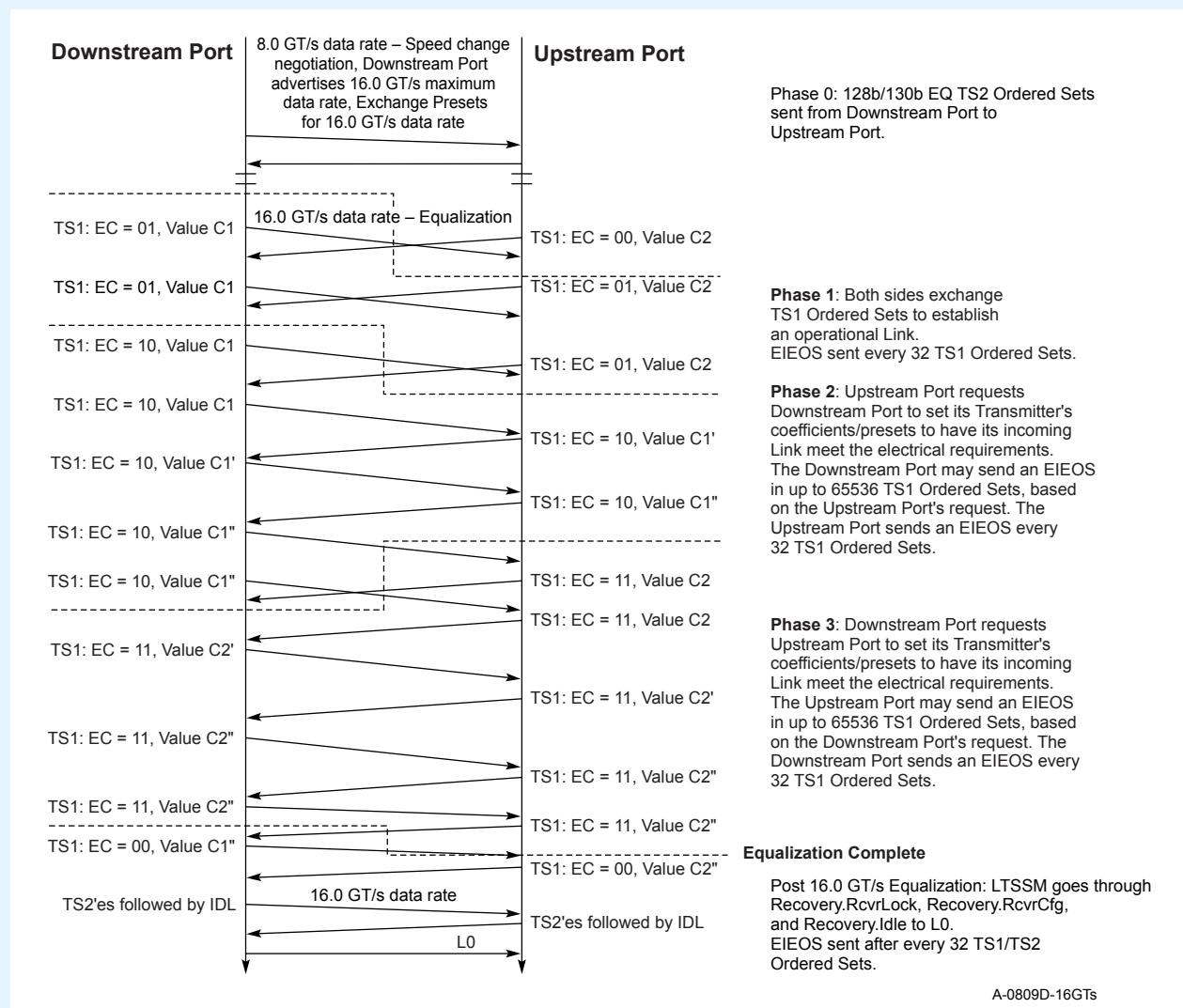


Figure 4-22 16.0 GT/s Equalization Flow

A-0809D-16GTs

## IMPLEMENTATION NOTE

### Equalization Bypass Example

The following flow-chart provides an example flow where a Link may bypass equalization at lower Data Rates and go to the highest support data rate for equalization. For example, when n=5, the Link can train to L0 in Gen 1 data rate, establish that all components (including Retimers, if any) can bypass equalization to Gen 5, change the data rate to Gen 5 and just perform equalization at Gen 5.

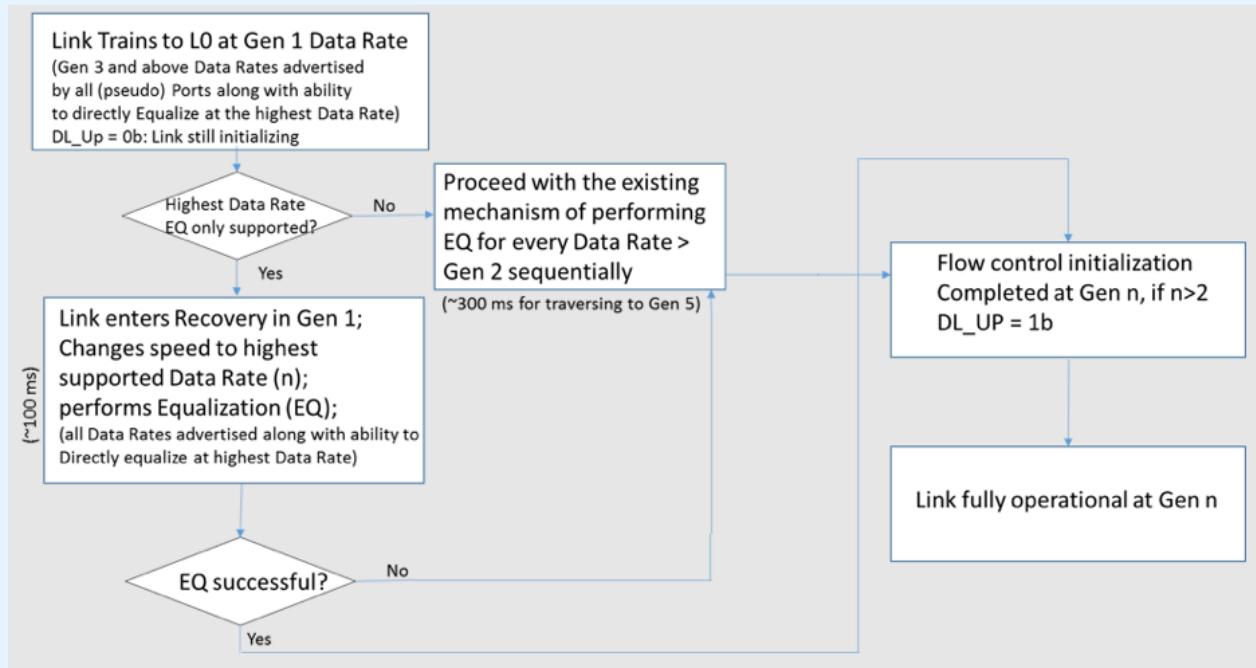


Figure 4-23 Equalization Bypass Example

#### 4.2.3.1 Rules for Transmitter Coefficients

The explanation of the coefficients and the FIR filter it represents are provided in [Section 8.3.3.1](#). The following rules apply to both the advertised as well as requested coefficient settings.

1. **C<sub>-1</sub>** and **C<sub>+1</sub>** are the coefficients used in the FIR equation and represent the pre-cursor and post-cursor, respectively. The pre-cursor and post-cursor values communicated in the TS1 Ordered Sets represent their absolute values. **C<sub>0</sub>** represents the cursor coefficient setting and is a positive entity.
2. The sum of the absolute values of the coefficients defines the FS (Full Swing;  $FS = |C_{-1}| + C_0 + |C_{+1}|$ ). FS is advertised to the Link partner in Phase 1. The Transmitter FS range is defined below:
  - $FS \in \{24, \dots, 63\}$  (i.e., FS must have a value from 24 through 63) for full swing mode.
  - $FS \in \{12, \dots, 63\}$  for reduced swing mode.
3. A Transmitter advertises its LF (Low Frequency) value during Phase 1. This corresponds to the minimum differential voltage that can be generated by the Transmitter which is LF/FS times the Transmitters maximum

differential voltage. The Transmitter must ensure that when equation c) below equals LF it must meet the electrical requirements defined in [Section 8.3.3.9](#) for  $V_{TX-EIEOS-FS}$  and  $V_{TX-EIEOS-RS}$ .

4. The following rules must be satisfied before a set of coefficients can be requested of the Link partner's Transmitter. Upon reception of an update request for TX coefficient settings, a Port must verify that the new request meets the following conditions and reject the request if any of following conditions are violated:
  - a.  $|C_{-1}| \leq \text{Floor}(\text{FS}/4)$
  - b.  $|C_{-1}| + C_0 + |C_{+1}| = \text{FS}$  (Do not allow peak power to change with adaptation)
  - c.  $C_0 - |C_{-1}| - |C_{+1}| \geq \text{LF}$

#### **4.2.3.2 Encoding of Presets**

Definition of the Transmitter and Receiver Preset Hints appears in [Chapter 8](#). The encoding for the Transmitter Preset and Receiver Preset Hint are provided in [Table 4-4](#) and [Table 4-5](#). Receiver Preset Hints are optional and only defined for the 8.0 GT/s data rate.

*Table 4-4 Transmitter Preset Encoding*

Encoding	Preset Number in <a href="#">Table 8-1</a>
0000b	P0
0001b	P1
0010b	P2
0011b	P3
0100b	P4
0101b	P5
0110b	P6
0111b	P7
1000b	P8
1001b	P9
1010b	P10
1011b through 1111b	Reserved

*Table 4-5 Receiver Preset Hint Encoding for 8.0 GT/s*

Encoding	Receiver Preset Value
000b	-6 dB
001b	-7 dB
010b	-8 dB
011b	-9 dB
100b	-10 dB

Encoding	Receiver Preset Value
101b	-11 dB
110b	-12 dB
111b	Reserved

## 4.2.4 Link Initialization and Training

This section defines the Physical Layer control process that configures and initializes each Link for normal operation. This section covers the following features:

- configuring and initializing the Link
- supporting normal packet transfers
- supported state transitions when recovering from Link errors
- restarting a Port from low power states.

The following are discovered and determined during the training process:

- Link width
- Link data rate
- Lane reversal
- Lane polarity

Training does:

- Link data rate negotiation.
- Bit lock per Lane
- Lane polarity
- Symbol lock or Block alignment per Lane
- Lane ordering within a Link
- Link width negotiation
- Lane-to-Lane de-skew within a multi-Lane Link.

### 4.2.4.1 Training Sequences

Training sequences are composed of Ordered Sets used for initializing bit alignment, Symbol alignment and to exchange Physical Layer parameters. When the data rate is 2.5 GT/s or 5.0 GT/s, Ordered Sets are never scrambled but are always 8b/10b encoded. When the data rate is 8.0 GT/s or higher, the 128b/130b encoding is used and Symbols may or may not be scrambled, according to the rules in [Section 4.2.2.4](#).

Training sequences (TS1 or TS2 or Modified TS1 or Modified TS2) are transmitted consecutively and can only be interrupted by SKP Ordered Sets (see [Section 4.2.7](#)) or, for data rates other than 2.5 GT/s, EIEOS Ordered Sets (see [Section 4.2.4.3](#)).

When 8.0 GT/s or higher data rates are supported, a TS1 (or TS2) Ordered Set using 8b/10b encoding (i.e., 2.5 or 5.0 GT/s data rate) can be either a standard TS1 (or TS2) Ordered Set (i.e., Symbol 6 is D10.2 for a TS1 Ordered Set or D5.2 for a TS2 Ordered Set) or an ***EQ TS1 Ordered Set*** (or ***EQ TS2 Ordered Set***) (i.e., Symbol 6 bit 7 is 1b). The ability to transmit EQ TS1 Ordered Sets is implementation-specific. Ports supporting 8.0 GT/s or higher data rates must accept either TS1 (or TS2) type in the LTSSM states unless explicitly required to look for a specific type. Ports that do not support the 8.0 GT/s data rate are permitted, but not required, to accept EQ TS1 (or TS2) Ordered Sets.

When the 16.0 GT/s and higher data rate is supported, a TS2 using 128b/130b encoding (i.e. 8.0 or higher data rate) can be either a standard TS2 Ordered Set (i.e., Symbol 7 is 45h) or an ***128b/130b EQ TS2*** (i.e., Symbol 7 bit 7 is 1b). Ports supporting the 16.0 GT/s or higher data rate must accept either TS2 type in the LTSSM states unless explicitly required to look for a specific type. Ports that do not support the 16.0 GT/s data rate are permitted, but not required, to accept 128b/130b EQ TS2 Ordered Sets.

When using 8b/10b encoding, TS1 or TS2 Ordered Sets are considered consecutive only if Symbol 6 matches the Symbol 6 of the previous TS1 or TS2 Ordered Set.

Components that intend to either negotiate alternate protocols or pass a Training Set Message must use Modified TS1/TS2 Ordered Sets instead of standard TS1/TS2 Ordered Sets in Configuration.Lanenum.Wait, Configuration.Lanenum.Accept, and Configuration.Complete substates. In order to be eligible to send the Modified TS1/TS2 Ordered Sets, components must set the Enhanced Link behavior Control bits (bit 7:6 of Symbol 5) in TS1 and TS2 Ordered Sets to 11b in Polling.Active, Polling.Configuration, Configuration.Linkwidth.Start, and Configuration.Linkwidth.Accept sub-states and follow through the steps outlined on transition to Configuration.Lanenum.Wait substate when LinkUp=0b. If the Link partner does not support Modified TS1/TS2 Ordered Sets, then starting with Configuration.LaneNum.Wait, the standard TSes should stop sending 11b in the Enhanced Link Behavior Control field and switch to the appropriate encodings.

When using 8b/10b encoding, modified TS1 or modified TS2 Ordered Sets are considered consecutive only if all Symbols matches the corresponding Symbols of the previous modified TS1 or modified TS2 Ordered Sets and the parity in Symbol 15 matches with the expected value. Symbols 8-14 must be identical in each Modified TS1/TS2 Ordered Sets across all Lanes of a Link.

When using 128b/130b encoding, TS1 or TS2 Ordered Sets are considered consecutive only if Symbols 6-9 match Symbols 6-9 of the previous TS1 or TS2 Ordered Set, with Reserved bits treated as described below.

Reserved bits in TS1 and TS2 Ordered Sets must be handled as follows:

- The Transmitter must transmit 0s for Reserved bits.
- The Receiver:
  - must not determine that a TS1 or TS2 Ordered Set is invalid based on the received value of Reserved bits
  - must use the received value of Reserved bits for the purpose of a parity computation if the Reserved bits are included in a parity calculation
  - may optionally compare the received value of Reserved bits within Symbols that are explicitly called out as being required to be identical in TS1 or TS2 Ordered Sets to determine if they are consecutive
  - must not otherwise take any functional action based on the value of any received Reserved bits

When using 128b/130b encoding, Transmitters are required to track the running DC Balance of the bits transmitted on the wire (after scrambling and precoding, if turned on) that constitute the TS1 and TS2 Ordered Sets only. The running DC Balance is the difference between the number of 1s transmitted and the number of 0s transmitted. Each Lane must track its running DC Balance independently and be capable of tracking a difference of at least 511 bits in either direction: 511 more 1s than 0s, and 511 more 0s than 1s. Any counters used must saturate at their limit (not roll-over) and continue to track reductions after their limit is reached. For example, a counter that can track a difference of 511 bits will saturate at 511 if a difference of 513 is detected, and then change to 509 if the difference is reduced by 2 in the future.

The running DC Balance is set to 0 by two events: 1) The Transmitter exiting Electrical Idle; 2) Transmission of an EIEOS following a Data Block.

For every TS1 or TS2 Ordered Set transmitted, Transmitters must evaluate the running DC Balance and transmit one of the DC Balance Symbols defined for Symbols 14 and 15 as defined by the algorithm below. If the number of 1s needs to be reduced, the DC Balance Symbols 20h (for Symbol 14) and 08h (for Symbol 15) are transmitted. If the number of 0s needs to be reduced, the DC Balance Symbols DFh (for Symbol 14) and F7h (for Symbol 15) are transmitted. If no change is required, the appropriate TS1 or TS2 Identifier Symbol is transmitted. Any DC Balance Symbols transmitted for Symbols 14 or 15 bypass scrambling, while TS1 and TS2 Identifier Symbols follow the standard scrambling rules. The following algorithm must be used to control the DC Balance:

- If the running DC Balance is > 31 at the end of Symbol 11 of the TS Ordered Set, transmit DFh for Symbol 14 and F7h for Symbol 15 to reduce the number of 0s, or 20h for Symbol 14 and 08h for Symbol 15 to reduce the number of 1s.
- Else, if the running DC Balance is > 15 at the end of Symbol 11 of the TS Ordered Set, transmit F7h for Symbol 15 to reduce the number of 0s, or 08h for Symbol 15 to reduce the number of 1s. Transmit the normal TS Identifier Symbol (scrambled) for Symbol 14.
- Else, transmit the normal TS Identifier Symbol (scrambled) for Symbols 14 and 15.

Receivers are permitted, but not required, to check Symbols 14 and 15 for the following values when determining whether a TS1 or TS2 Ordered Set is valid: The appropriate TS Identifier Symbol after de-scrambling, or a valid DC Balance Symbol of DFh or 20h before de-scrambling for Symbol 14, or a valid DC Balance Symbol of F7h or 08h before de-scrambling for Symbol 15.

If a Receiver receives a DC balance pattern in Symbol 14, it is possible that the pattern is scrambled (and precoded). Thus if the Receiver is performing this optional check, it must keep descrambler and receive precoding running for checking Symbol 15, which can be either scrambled (and precoded) or the DC balance pattern.

## IMPLEMENTATION NOTE

### Sync Header and DC Balance

Block Sync Header bits and the first Symbol of TS1 and TS2 Ordered Sets do not affect the running DC Balance, because they have equal number of 1s and 0s.

The Training control bits Hot Reset bit, Disable Link bit, and Loopback bit are mutually exclusive, only one of these bits is permitted to be set at a time as well as transmitted on all Lanes in a configured (all Lanes that were in L0) or possible (all Lanes in Configuration) Link. If more than one of Hot Reset bit, Disable Link bit, or Loopback bit are Set at the same time, the Link behavior is undefined.

The TS1 Ordered Set's Retimer Equalization Extend bit is always set to 0b when transmitted by an Upstream Port or Downstream Port. Retimers set the bit to 1b as described in Section 4.3.7.2.

*Table 4-6 TS1 Ordered Set*

Symbol Number	Description
0	<ul style="list-style-type: none"> <li>• When operating at 2.5 or 5.0 GT/s: COM (K28.5) for Symbol alignment.</li> <li>• When operating at 8.0 GT/s or above: Encoded as 1Eh (TS1 Ordered Set).</li> </ul>

Symbol Number	Description
1	<p>Link Number.</p> <ul style="list-style-type: none"> <li>• Ports that do not support 8.0 GT/s or above: 0-255, PAD.</li> <li>• Downstream Ports that support 8.0 GT/s or above: 0-31, PAD.</li> <li>• Upstream Ports that support 8.0 GT/s or above: 0-255, PAD.</li> <li>• When operating at 2.5 or 5.0 GT/s: PAD is encoded as K23.7.</li> <li>• When operating at 8.0 GT/s or above: PAD is encoded as F7h.</li> </ul>
2	<p>Lane Number within Link.</p> <ul style="list-style-type: none"> <li>• When operating at 2.5 or 5.0 GT/s: 0-31, PAD. PAD is encoded as K23.7.</li> <li>• When operating at 8.0 GT/s or above: 0-31, PAD. PAD is encoded as F7h.</li> </ul>
3	N_FTS. The number of Fast Training Sequences required by the Receiver: 0-255.
4	<p>Data Rate Identifier</p> <p><b>Bit 0</b> Reserved for future Data Rate.</p> <p><b>Bit 1</b> 2.5 GT/s Data Rate Supported. Must be set to 1b.</p> <p><b>Bit 2</b> 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. See Section 8.2 .</p> <p><b>Bit 3</b> 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b.</p> <p><b>Bit 4</b> 16.0 GT/s Data Rate Supported. Must be set to 1b if Bit 5 is 1b.</p> <p><b>Bit 5</b> 32.0 GT/s Data Rate Supported.</p> <p><b>Bit 6</b> <b><i>Autonomous Change&gt;Selectable De-emphasis.</i></b> <ul style="list-style-type: none"> <li>• Downstream Ports: This bit is defined for use in the following LTSSM states: <u>Polling.Active</u>, <u>Configuration.Linkwidth.Start</u>, and <u>Loopback.Entry</u>. In all other LTSSM states, it is Reserved.</li> <li>• Upstream Ports: This bit is defined for use in the following LTSSM states: <u>Polling.Active</u>, <u>Configuration</u>, <u>Recovery</u>, and <u>Loopback.Entry</u>. In all other LTSSM states, it is Reserved.</li> </ul> </p> <p><b>Bit 7</b> speed_change. This bit can be set to 1b only in the <u>Recovery.RcvrLock</u> LTSSM state. In all other LTSSM states, it is Reserved.</p>
5	<p>Training Control</p> <p><b>Bit 0</b> <b><i>Hot Reset bit</i></b> <ul style="list-style-type: none"> <li><b>0b</b> Deassert</li> <li><b>1b</b> Assert</li> </ul> </p> <p><b>Bit 1</b> <b><i>Disable Link bit</i></b> <ul style="list-style-type: none"> <li><b>0b</b> Deassert</li> <li><b>1b</b> Assert</li> </ul> </p> <p><b>Bit 2</b> <b><i>Loopback bit</i></b> <ul style="list-style-type: none"> <li><b>0b</b> Deassert</li> <li><b>1b</b> Assert</li> </ul> </p> <p><b>Bit 3</b> <b><i>Disable Scrambling bit</i></b> (2.5 GT/s and 5.0 GT/s data rates)</p>

Symbol Number	Description
	<p><b>0b</b> Deassert  <b>1b</b> Assert          Reserved (other data rates)</p>
<b>Bit 4</b>	<p><b>Compliance Receive bit</b> (5.0 GT/s and above data rates, optional at 2.5 GT/s)</p> <p><b>0b</b> Deassert  <b>1b</b> Assert</p> <p>Ports that support 5.0 GT/s and above data rate(s) must implement the <u>Compliance Receive bit</u>. Ports that support only 2.5 GT/s data rate may optionally implement the <u>Compliance Receive bit</u>. If not implemented, the bit is Reserved.</p>
<b>Bit 5</b>	<p><b>Transmit Modified Compliance Pattern in Loopback.</b> This bit is defined for use in Loopback by the Loopback Master when 32.0 GT/s or higher data rates are supported. See <u>Section 4.2.6.10.1</u>. In all other cases, this bit is Reserved.</p>
<b>Bit 7:6</b>	<p><b>Enhanced Link Behavior Control</b></p> <p><b>00b</b> <b>Full Equalization required</b>          Modified TS1/TS2 Ordered Sets not supported.</p> <p><b>01b</b> <b>Equalization bypass to highest rate support</b>          Modified TS1/TS2 Ordered Sets not supported.          Indicates intention to perform 32.0 GT/s equalization when set by <u>Loopback Master</u>. See <u>Section 4.2.3</u> and <u>Section 4.2.6.10.1</u>.</p> <p><b>10b</b> <b>No Equalization Needed</b>          Modified TS1/TS2 Ordered Sets not supported          A device advertising this capability must support Equalization bypass to highest rate. See <u>Section 4.2.3</u>.</p> <p><b>11b</b> <b>Modified TS1/TS2 Ordered Sets supported</b>          Equalization bypass options specified in Modified TS1/TS2 Ordered Sets.</p> <p>These bits are defined for use in <u>Polling</u> and <u>Configuration</u> when <u>LinkUp=0b</u> and 32.0 GT/s or higher data rates are supported and in <u>Loopback</u> by the <u>Loopback Master</u> when 32.0 GT/s or higher data rates are supported. In all other cases, these bits are Reserved.</p>
When operating at 2.5 or 5.0 GT/s:	<ul style="list-style-type: none"> <li>Standard TS1 Ordered Sets encode this Symbol as a TS1 Identifier, D10.2 (4Ah).</li> <li><u>EQ TS1 Ordered Sets</u> encode this Symbol as follows: <ul style="list-style-type: none"> <li>For Equalization at 8.0 GT/s Data Rate: <ul style="list-style-type: none"> <li><b>Bits 2:0</b> <b>Receiver Preset Hint</b>. See <u>Section 4.2.3.2</u>.</li> <li><b>Bit 6:3</b> <b>Transmitter Preset</b>. See <u>Section 4.2.3.2</u>.</li> <li><b>Bit 7</b> Set to 1b.</li> </ul> </li> <li>For Equalization at 32.0 GT/s or higher Data Rate: <ul style="list-style-type: none"> <li><b>Bit 0</b> <b>Transmitter Precode Request</b> - See <u>Section 4.2.2.5</u>. This bit has no defined usage in receivers at this time.</li> <li><b>Bit 2:1</b> Reserved</li> <li><b>Bit 6:3</b> Transmitter Preset. See <u>Section 4.2.3.2</u>.</li> <li><b>Bit 7</b> Set to 1b.</li> </ul> </li> </ul> </li> </ul>
When operating at 8.0 GT/s or higher data rate:	

Symbol Number	Description
	<p><b>Bit 1:0</b> Equalization Control (EC). These bits are only used in the Recovery.Equalization and Loopback LTSSM states. See Section 4.2.6.4.2 and Section 4.2.6.10 . In all other LTSSM states, they must be set to 00b.</p> <p><b>Bit 2</b> Reset EIEOS Interval Count. This bit is defined for use in the Recovery.Equalization LTSSM state. See Section 4.2.6.4.2 and Section 4.2.4.3 . In all other LTSSM states, it is Reserved.</p> <p><b>Bit 6:3</b> Transmitter Preset. See Section 4.2.3 and Section 4.2.6 .</p> <p><b>Bit 7</b> <b>Use Preset</b>/Equalization Redo. This bit is defined for use in the Recovery.Equalization, Recovery.RcvrLock and Loopback LTSSM states. See Section 4.2.6.4.1 , Section 4.2.6.4.2 and Section 4.2.6.10 . In all other LTSSM states, it is Reserved.</p>
7	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah).</li> <li>When operating at 8.0 GT/s or higher:           <p><b>Bit 5:0</b> FS when the EC field of Symbol 6 is 01b (see Section 4.2.3.1 ). Otherwise, Pre-cursor Coefficient for the current data rate of operation.</p> <p><b>Bit 6</b> <b>Transmitter Precoding on</b>. This bit is defined for use in the Recovery state for use at 32.0 GT/s or higher. See Section 4.2.2.5 . In all the other cases, it is Reserved.</p> <p><b>Bit 7</b> <b>Retimer Equalization Extend</b> bit. This bit is defined for use in the Recovery.Equalization LTSSM state when operating at 16.0 GT/s or higher data rate. In all other LTSSM states and when operating at 8.0 GT/s, it is Reserved.</p> </li> </ul>
8	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah).</li> <li>When operating at 8.0 GT/s or higher data rate:           <p><b>Bit 5:0</b> LF when the EC field of Symbol 6 is 01b (see Section 4.2.3.1 ). Otherwise, Cursor Coefficient for the current data rate of operation.</p> <p><b>Bit 7:6</b> Reserved.</p> </li> </ul>
9	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah).</li> <li>When operating at 8.0 GT/s or higher data rate:           <p><b>Bit 5:0</b> Post-cursor Coefficient for the current data rate of operation.</p> <p><b>Bit 6</b> <b>Reject Coefficient Values bit</b>. This bit can only be set to 1b in specific Phases of the Recovery.Equalization LTSSM State. See Section 4.2.6.4.2 . In all other LTSSM states, it must be set to 0b.</p> <p><b>Bit 7</b> Parity (P). This bit is the even parity of all bits of Symbols 6, 7, and 8 and bits 6:0 of Symbol 9. Receivers must calculate the parity of the received bits and compare it to the received Parity bit. Received TS1 Ordered Sets are valid only if the calculated and received Parity match.</p> </li> </ul>
10 - 13	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah).</li> <li>When operating at 8.0 GT/s or above: TS1 Identifier. Encoded as 4Ah.</li> </ul>
14 - 15	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS1 Identifier. Encoded as D10.2 (4Ah).</li> <li>When operating at 8.0 GT/s or above: TS1 Identifier (encoded as 4Ah) or a DC Balance Symbol.</li> </ul>

Table 4-7 TS2 Ordered Set

Symbol Number	Description
0	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: COM (K28.5) for Symbol alignment.</li> <li>When operating at 8.0 GT/s or above: Encoded as 2Dh (TS2 Ordered Set).</li> </ul>
1	<p>Link Number.</p> <ul style="list-style-type: none"> <li>Ports that do not support 8.0 GT/s or above: 0-255, PAD.</li> <li>Downstream Ports that support 8.0 GT/s or above: 0-31, PAD.</li> <li>Upstream Ports that support 8.0 GT/s or above: 0-255, PAD.</li> <li>When operating at 2.5 or 5.0 GT/s: PAD is encoded as K23.7.</li> <li>When operating at 8.0 GT/s or above: PAD is encoded as F7h.</li> </ul>
2	<p>Lane Number within Link.</p> <ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: 0-31, PAD. PAD is encoded as K23.7.</li> <li>When operating at 8.0 GT/s or above: 0-31, PAD. PAD is encoded as F7h.</li> </ul>
3	N_FTS. The number of Fast Training Sequences required by the Receiver: 0-255.
4	<p>Data Rate Identifier</p> <p><b>Bit 0</b> Reserved for future Data Rate</p> <p><b>Bit 1</b> 2.5 GT/s Data Rate Supported. Must be set to 1b.</p> <p><b>Bit 2</b> 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. See Section 8.2 .</p> <p><b>Bit 3</b> 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b.</p> <p><b>Bit 4</b> 16.0 GT/s Data Rate Supported. Must be set to 1b if Bit 5 is 1b.</p> <p><b>Bit 5</b> 32.0 GT/s Data Rate Supported</p> <p><b>Bit 6</b> Autonomous Change&gt;Selectable De-emphasis/Link Upconfigure Capability. This bit is defined for use in the following LTSSM states: <u>Polling</u>, <u>Configuration</u>, <u>Configuration.Complete</u>, and <u>Recovery</u>. In all other LTSSM states, it is Reserved.</p> <p><b>Bit 7</b> speed_change. This bit can be set to 1b only in the <u>Recovery.RcvrCfg</u> LTSSM state. In all other LTSSM states, it is Reserved.</p>
5	<p>Training Control</p> <p><b>Bit 0</b> Hot Reset bit</p> <p><b>0b</b> Deassert</p> <p><b>1b</b> Assert</p> <p><b>Bit 1</b> Disable Link bit</p> <p><b>0b</b> Deassert</p> <p><b>1b</b> Assert</p> <p><b>Bit 2</b> Loopback bit</p>

Symbol Number	Description
	<p><b>0b</b> Deassert</p> <p><b>1b</b> Assert</p>
<b>Bit 3</b>	<u>Disable Scrambling bit</u> in 2.5 GT/s and 5.0 GT/s data rates; Reserved in other data rates
	<p><b>0b</b> Deassert</p> <p><b>1b</b> Assert</p>
<b>Bit 4</b>	<u>Retimer Present bit</u> in 2.5 GT/s data rate. Reserved in other data rates.
	<p><b>0b</b> No Retimers present</p> <p><b>1b</b> One or more Retimers present</p>
<b>Bit 5</b>	<u>Two Retimers Present bit</u> in 2.5 GT/s data rate. Reserved in other data rates. Ports that support 16.0 GT/s data rate or higher must implement this bit. Ports that support only 8.0 GT/s data rate or lower are permitted to implement this bit.
	<p><b>0b</b> Zero or one Retimers present</p> <p><b>1b</b> Two or more Retimers present</p>
<b>Bit 7:6</b>	<u>Enhanced Link Behavior Control</u>
	<p><b>00b</b> Full Equalization required, Modified TS1/TS2 Ordered Sets not supported.</p> <p><b>01b</b> Equalization bypass to highest rate support <u>Modified TS1/TS2 Ordered Sets</u> not supported. See Section 4.2.3 .</p> <p><b>10b</b> <u>No Equalization Needed</u>, A device advertising this capability must support Equalization bypass to highest rate. See Section 4.2.3 . Modified TS1/TS2 Ordered Sets not supported</p> <p><b>11b</b> <u>Modified TS1/TS2 Ordered Sets</u> supported, Equalization bypass options specified in Modified TS1/TS2 Ordered Sets.</p>
	These bits defined for use in <u>Polling</u> and <u>Configuration</u> when <u>LinkUp=0</u> and 32.0 GT/s or higher data rate is supported. In all other cases, Bits 7:6 are Reserved.
6	<ul style="list-style-type: none"> <li>• When operating at 2.5 or 5.0 GT/s: <ul style="list-style-type: none"> <li>◦ Standard TS2 Ordered Sets encode this Symbol as a TS2 Identifier, D5.2 (45h).</li> <li>◦ EQ TS2 Ordered Sets encode this Symbol as follows: <ul style="list-style-type: none"> <li>▪ For Equalization at 8.0 GT/s Data Rate: <b>Bit 2:0</b> Receiver Preset Hint. See Section 4.2.3.2 .</li> <li><b>Bit 6:3</b> Transmitter Preset. See Section 4.2.3.2 .</li> <li><b>Bit 7</b> Set to 1b.</li> </ul> </li> <li>▪ For Equalization at 32.0 GT/s or higher Data Rate: <b>Bit 0</b> <b>Transmitter Precode Request</b>. See Section 4.2.2.5 .</li> <li><b>Bit 2:1</b> Reserved</li> <li><b>Bit 6:3</b> Transmitter Preset. See Section 4.2.3.2 .</li> <li><b>Bit 7</b> Set to 1b.</li> </ul> </li> <li>• When operating at 8.0 GT/s or higher Data Rate: <b>Bit 3:0</b> Reserved.</li> <li><b>Bit 5:4</b> <b>Equalization Request Data Rate</b>.</li> </ul>

Symbol Number	Description
	<p><b>00b</b> 8.0 GT/s  <b>10b</b> 16.0 GT/s  <b>01b</b> 32.0 GT/s  <b>11b</b> Reserved</p> <p>These bits are defined for use in the <u>Recovery.RcvrCfg</u> LTSSM state. In all other LTSSM states, they are Reserved. See <u>Section 4.2.3</u> for usage and recognize that these bits are non-sequentially encoded for purposes of backwards compatibility</p> <p><b>Bit 6</b> <b>Quiesce Guarantee.</b> This bit is defined for use in the <u>Recovery.RcvrCfg</u> LTSSM state. In all other LTSSM states, it is Reserved.</p> <p><b>Bit 7</b> <b>Request Equalization.</b> This bit is defined for use in the <u>Recovery.RcvrCfg</u> LTSSM state. In all other LTSSM states, it is Reserved.</p>
7	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS2 Identifier. Encoded as D5.2 (45h).</li> <li>When operating at 8.0 GT/s or above: <ul style="list-style-type: none"> <li>Standard TS2 Ordered Sets encode this Symbol as a TS2 Identifier, 45h.</li> <li>128b/130b EQ TS2 Ordered Sets encode this Symbol as follows: <ul style="list-style-type: none"> <li><b>Bit 0</b> Transmitter Precode Request for operating at 32.0 GT/s or higher Data Rate. See <u>Section 4.2.2.5</u>. This bit is Reserved if the <u>128b/130b EQ TS2</u> is sent for equalization at data rates of 8.0 GT/s or 16.0 GT/s.</li> <li><b>Bit 2:1</b> Reserved</li> <li><b>Bit 6:3</b> 128b/130b Transmitter Preset. See <u>Section 4.2.3.2</u>.</li> <li><b>Bit 7</b> Set to 1b.</li> </ul> </li> </ul> </li> </ul> <p>This definition is only valid in the <u>Recovery.RcvrCfg</u> LTSSM state when Preset values are being communicated.</p>
8 - 13	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS2 Identifier. Encoded as D5.2 (45h).</li> <li>When operating at 8.0 GT/s or above: TS2 Identifier. Encoded as 45h.</li> </ul>
14-15	<ul style="list-style-type: none"> <li>When operating at 2.5 or 5.0 GT/s: TS2 Identifier. Encoded as D5.2 (45h).</li> <li>When operating at 8.0 GT/s or above: TS2 Identifier (encoded as 45h) or a DC Balance Symbol.</li> </ul>

*Table 4-8 Modified TS1/TS2 Ordered Set (8b/10b encoding)*

Symbol Number	Description
0	COM (K28.5) for Symbol alignment.
1	<p>Link Number.</p> <p>Downstream Ports: 0-31, PAD (K23.7).</p> <p>Upstream Ports: 0-255, PAD (K23.7).</p>

Symbol Number	Description
2	Lane Number within Link : 0-31, PAD. PAD is encoded as K23.7.
3	N_FTS. The number of Fast Training Sequences required by the Receiver: 0-255.
4	<p>Data Rate Identifier</p> <p><b>Bit 0</b> Reserved for future Data Rate.</p> <p><b>Bit 1</b> 2.5 GT/s Data Rate Supported. Must be set to 1b.</p> <p><b>Bit 2</b> 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. See Section 8.2 .</p> <p><b>Bit 3</b> 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b.</p> <p><b>Bit 4</b> 16.0 GT/s Data Rate Supported. Must be set to 1b if Bit 5 is 1b.</p> <p><b>Bit 5</b> 32.0 GT/s Data Rate Supported.</p> <p><b>Bit 6</b> <b><i>Link Upconfigure Capability</i></b></p> <p><b>Bit 7</b> Reserved.</p>
5	<p>Training/ Equalization Control</p> <p><b>Bit 0</b> Equalization bypass to highest rate support. See Section 4.2.3</p> <p><b>Bit 1</b> <b><i>No Equalization Needed bit</i></b>. See Section 4.2.3</p> <p><b>Bit 3:2</b> Reserved</p> <p><b>Bit 4</b> Retimer Present bit</p> <ul style="list-style-type: none"> <li><b>0b</b> No Retimers present</li> <li><b>1b</b> One Retimer is present</li> </ul> <p><b>Bit 5</b> Two or more Retimers Present</p> <p><b>Bit 6</b> 1b</p> <p><b>Bit 7</b> 1b</p>
6	<p>For Modified TS1: TS1 Identifier, encoded as D10.2</p> <p>For Modified TS2: TS2 Identifier, encoded as D5.2</p>
7	<p>For Modified TS1: TS1 Identifier, encoded as D10.2</p> <p>For Modified TS2: TS2 Identifier, encoded as D5.2</p>
8-9	<p><b>Bits 2:0</b> <b><i>Modified TS Usage</i></b></p> <ul style="list-style-type: none"> <li><b>000b</b> PCIe protocol only</li> <li><b>001b</b> PCIe protocol only with vendor defined <b><i>Training Set Messages</i></b></li> <li><b>010b</b> <b><i>Alternate Protocol Negotiation</i></b></li> <li><b>011b</b> Reserved</li> <li><b>through</b> The values advertised in these bits must be consistent with the <u>Modified TS Usage Mode Selected field of the 32.0 GT/s Control register</u> and the capabilities of the device. These are bits[2:0] of Symbol 8.</li> <li><b>111b</b></li> </ul> <p><b>Bits 15:3</b> <b><i>Modified TS Information 1</i></b></p>

Symbol Number	Description
	If <u>Modified TS Usage</u> = 001b or 010b; else Reserved.
10-11	<b>Training Set Message Vendor ID</b> if <u>Modified TS Usage</u> = 001b. <b>Alternate Protocol Vendor ID</b> if <u>Modified TS Usage</u> = 010b. Reserved for other cases.
12-14	If <u>Modified TS Usage</u> = 001b or 010b, <b>Modified TS Information 2</b> Else, Reserved
15	Bit-wise even parity of Symbols 4 through 14. . For example: Bit 0 = Symbol 4 Bit [0] ^ Symbol 5 Bit [0] ^ .... ^ Symbol 14 Bit [0], ..., Bit [7] = Symbol 4 Bit [7] ^ Symbol 5 Bit [7] ^ .... ^ Symbol 14 Bit [7]

Fields in the Modified TS1/TS2 Ordered Sets that extend over multiple Symbols use the little endian format using all the bits over those multiple Symbols. For example, Symbols 8 and 9 of the Modified TS1/TS2 comprise 16 bits. The Modified TS Usage field goes in bits [2:0] of Symbol 8 with the bit 0 of Modified TS Usage field placed in bit 0 of Symbol 8, bit 1 of Modified TS Usage field placed in bit 1 of Symbol 8, and bit 2 of Modified TS Usage field placed in bit 2 of Symbol 8. Similarly, bit 12 of the 13 bits of Modified TS Information 1 field is placed in bit 7 of Symbol 9 where as bit 0 of Modified TS Information 1 is placed in bit 3 of Symbol 8.

#### 4.2.4.2 Alternate Protocol Negotiation

In addition to the decision to skip equalization, alternate protocols can also be negotiated during the Configuration.Lanenum.Wait, Configuration.Lanenum.Accept, and Configuration.Complete substates, while LinkUp=0b, through the exchange of Modified TS1/TS2 Ordered sets in the 8b/10b encoding.

Alternate protocol(s) may be supported with PCIe PHY in 128b/130b encoding. An alternate protocol is defined to be a non-PCIe protocol using the PCIe PHY layer. One may choose to run PCIe protocol in addition to one or multiple alternate protocols in the alternate protocol mode. The Ordered Set blocks are used as-is, along with the rules governing SKP Ordered Set insertion and the transition between Ordered Set and Data Blocks. The contents of the Data Blocks, however, may be modified according to the rules of the alternate protocol.

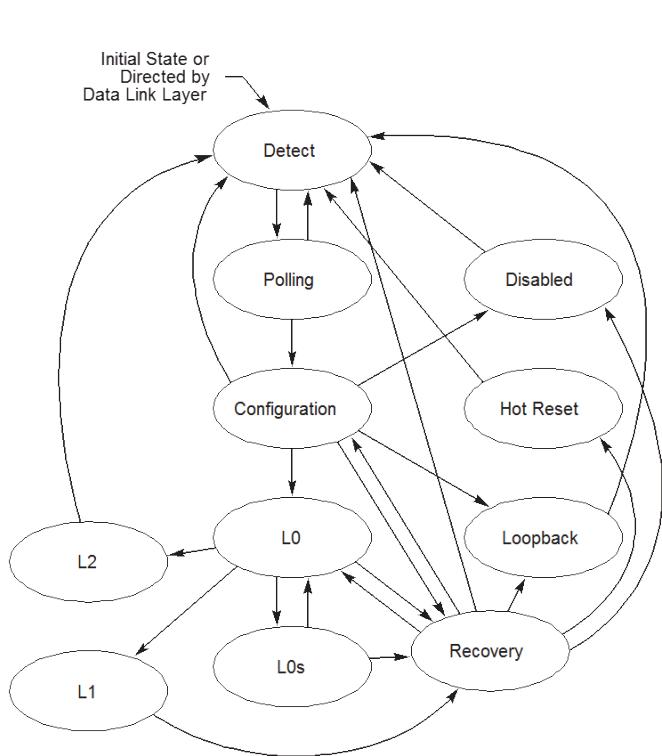
### IMPLEMENTATION NOTE

#### Alternate Protocols should have an EDS Token Equivalent

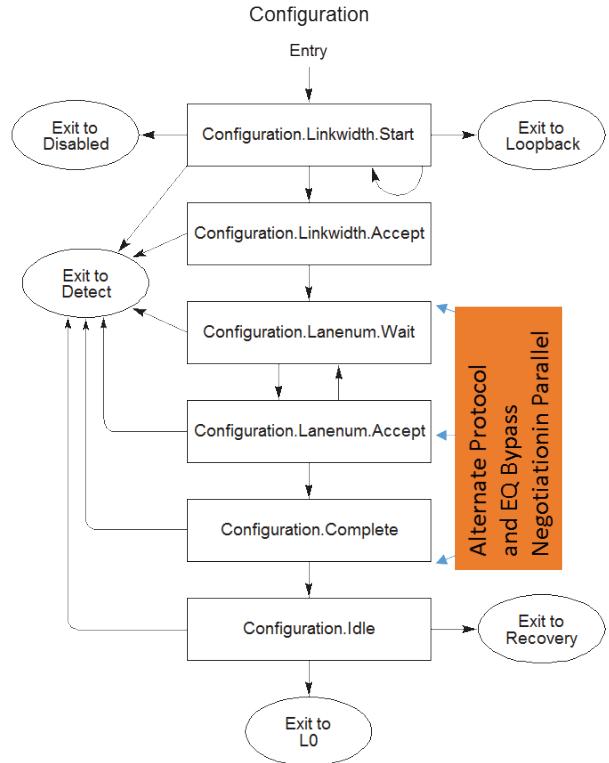
The EDS Token is used in PCI Express to indicate a switch from Data Blocks to Ordered Set blocks. This additional "redundant" information ensures that a random bit error in the 2 bit block header isn't incorrectly interpreted as the end of a data stream. This is one mechanism used by PCI Express to accomplish an undected data error Hamming Distance of 4.

Alternate protocols should have an equivalent mechanism.

The following diagram represents the states where alternate protocol and equalization bypass negotiation occurs:



( PCIe Link Training State Machine LTSSM )



( Modified Training Sets in Config State of LTSSM to negotiate Protocol )

Figure 4-24 Alternate Protocol Negotiation and Equalization Bypass LTSSM States

Downstream Ports manage Alternate Protocol Negotiation and Training Set Messages based on the value of the Modified TS Usage Mode Selected field when the Port is in Configuration.Lanenum.Wait, Configuration.Lanenum.Accept, and Configuration.Complete substates with LinkUp = 0.

Upstream Ports must respond to unsupported Modified TS Usage values by transmitting Modified TS Usage 000b.

If Modified TS Usage Mode Selected is:

**000b**

No Alternate Protocol Negotiation or Training Set Message occurs. The link will operate as a PCI Express Link.

**001b**

Training Set Messages are enabled. Modified TS Information 1 and Modified TS Information 2 fields carry the vendor specific messages defined by the Training Set Message Vendor ID field.

**010b**

Alternate Protocol Negotiation is enabled. Modified TS Information 1 and Modified TS Information 2 fields carry the alternate protocol details defined by the Alternate Protocol Vendor ID field. A protocol request or response is associated with the protocol determined by Alternate Protocol Details and Alternate Protocol Vendor ID. A protocol request or response is associated with the protocol defined by the Alternate Protocol Vendor ID field.

The Alternate Protocol Negotiation Status field indicates the progress of the negotiation protocol.

**others**

Reserved

A Downstream Port that supports Alternate Protocol Negotiation will start the negotiation process when it first enters Configuration.Lanenum.Wait, LinkUp = 0, and Modified TS Usage Mode Selected field is 010b. Starting negotiation consists of sending Modified TS1/TS2 Ordered Sets with Modified TS Usage = 010b.

*Table 4-9 Modified TS Information 1 field in Modified TS1/TS2 Ordered Sets if Modified TS Usage = 010b (Alternate Protocol)*

Bits	Field	Description	
For Modified TS1 Ordered Sets:			
4:3  <i>Alternate Protocol Negotiation Status</i>	<b>00b</b>	<b>DP</b>	Indicates a protocol request from the Downstream Port asking whether the Upstream Port supports a particular alternate protocol.
		<b>UP</b>	Indicates that the Upstream Port does not have an answer for a protocol request yet. This occurs either when it is evaluating the protocol request or it has not received two consecutive Modified TS1s to perform the evaluation. In the former case, <u>Alternate Protocol Vendor ID</u> and <u>Alternate Protocol Details</u> reflect what it received, while <u>Modified TS Information 2</u> is protocol specific. In the latter case, all 3 fields must be 0.
	<b>01b</b>	<b>DP</b>	Reserved
		<b>UP</b>	Indicates that the Upstream Port does not support the requested protocol. <u>Alternate Protocol Vendor ID</u> and <u>Alternate Protocol Details</u> reflect what it received. <u>Modified TS Information 2</u> must be all 0s.
	<b>10b</b>	<b>DP</b>	Reserved
		<b>UP</b>	Indicates that the Upstream Port supports the requested protocol. <u>Alternate Protocol Vendor ID</u> and <u>Alternate Protocol Details</u> reflect what it received, while <u>Modified TS Information 2</u> field is protocol specific.
	<b>11b</b>	Reserved	
	For Modified TS2 Ordered Sets:		
	<b>00b</b>		Indicates a protocol confirmation from the Downstream Port as well as the Upstream Port. Behavior is undefined if the Downstream Port had not earlier received status 10b for this protocol in this instance of protocol negotiation during the Modified TS1 Ordered Sets. Similarly, behavior is undefined if the Upstream Port had not earlier transmitted status 10b for this protocol in this instance of protocol negotiation during the Modified TS1 Ordered Sets.
			No protocol is selected unless the Downstream Port sends and receives a protocol confirmation in the Modified TS2 Ordered Sets. If the Downstream Port decides not to use any Alternate Protocol, it may optionally indicate this by transmitting <u>Modified TS2 Ordered Set</u> with <u>Modified TS Usage</u> of 000b.
15:5  <i>Alternate Protocol Details</i>	<b>01b, 10b,</b> <b>11b</b>	Reserved	
			Alternate Protocol Details is <u>Modified TS Usage</u> = 010b.

If Modified TS Usage = 001b, then Modified TS Information 1 and Modified TS Information 2 contain details of the training set messages.

Alternate Protocol Negotiation must be concurrent with the Lane number negotiation. The DP is responsible for ensuring that they arrive at a consensus on the Alternate Protocol Negotiation prior to transitioning to Configuration.Complete substate. It is permitted to fall back to PCIe protocol if the Alternate Protocol Negotiation does not arrive at a consensus. On a successful negotiation to alternate protocol, the Link moves to L0 at 2.5 GT/s, switches the data rate to the higher data rates, performing equalization, if needed and enters L0 at the highest data rate desired. After transmitting the SDS