

4.1.4 Free Running Clock Mode

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

4.1.5 Sideband transmission

Each module supports a sideband interface with a serial data and clock pin pair. Sideband packet formats and encodings are shown in [Chapter 7.0](#) and the electrical characteristics are shown in [Section 5.13](#).

As shown in [Section 7.1.2](#), the sideband message formats are defined as a 64-bit header with no data, with 32 bits or 64 bits of data. A 64-bit serial packet is defined on the I/O interface to the remote die as shown in [Figure 4-8](#). 32-bit data is sent using the 64-bit serial packet with MSBs padded with 0b. Two sideband serial packets on the I/O interface are separated by a minimum of 32 bits low as shown in [Figure 4-9](#). A sideband message with data would be transferred as a 64-bit header followed by 32 bits of low followed by 64-bit data followed by 32 bits of low.

Figure 4-8. Example 64-bit Sideband Serial Packet Transfer

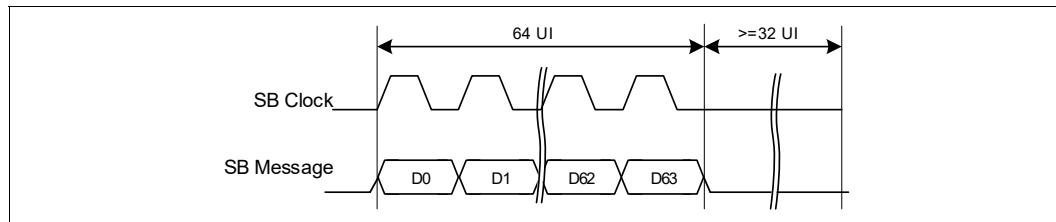
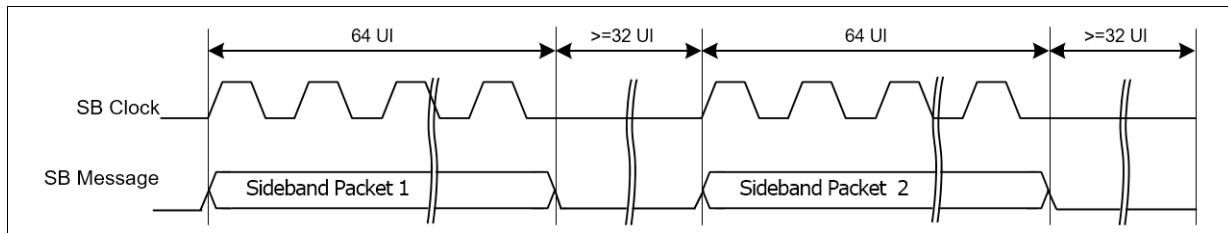


Figure 4-9. Sideband Packet Transmission: Back-to-Back



4.1.5.1 Sideband Performant Mode Operation (PMO)

Sideband designs can negotiate support for ‘Sideband Performant Mode Operation (PMO)’ by way of the Sideband Feature Extensions (SBFE) mechanism defined in [Section 4.5.3.3.1.1](#). The Sideband PMO bit of the {MBINIT.PARAM SBFE req/respond} sideband message (bit 1) is defined for negotiating this operation (see [Table 7-11](#)). When supporting this feature, a UCIe Link must support this capability on both its transmit and receive direction or not support the capability at all. In a multi-module Link, all modules must advertise the same capability. A UCIe Link must set the Sideband PMO bit to 1 on the {MBINIT.PARAM SBFE respond} sideband message that the UCIe Link transmits, but only if the corresponding Req message also has that bit set to 1 and its receiver is capable of Sideband PMO. Otherwise, the Sideband PMO bit must be cleared to 0.

When Sideband PMO capability is enabled, the 32-UI dead time between the 64-UI data transfers on the sideband is no longer applicable and the sideband can transmit 64-UI data back-to-back with no gaps. See [Figure 4-10](#) and [Figure 4-11](#) for illustration. The transmitter must follow this new mode after the transmitter has sent and received the {MBINIT.PARAM SBFE respond} sideband message with

the Sideband PMO bit set to 1, across all modules. The receiver must be ready to accept packets in this mode after the receiver has transmitted the {MBINIT.PARAM SBFE resp} sideband message with the Sideband PMO bit set to 1. After Sideband PMO is enabled, the transmitter operates in Performant Mode in all states until entry into the RESET state with the SB_MGMT_UP flag cleared to 0. Additionally, PMO can then only be renegotiated on a training sequence with the SB_MGMT_UP flag cleared to 0. Note that from a receiver perspective, due to timing differences, packets might be received without the Sideband Performant Mode even after the chiplet has transmitted an {MBINIT.PARAM SBFE resp} sideband message with the PMO bit set to 1. The sideband receiver must be backward compatible and be able to handle 32 UI of gaps between consecutive 64-UI transfers over the sideband Link.

Figure 4-10. Example 64-bit Sideband Serial Packet Transfer in Sideband Performant Mode

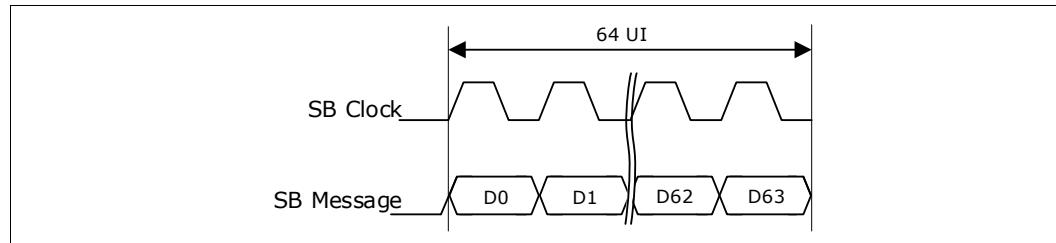
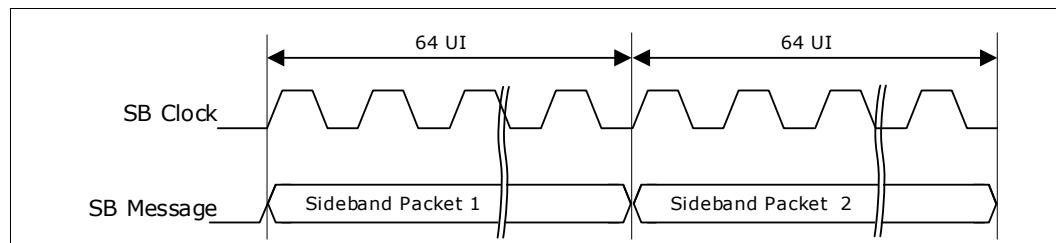


Figure 4-11. Sideband Packet Transmission: Back-to-Back in Sideband Performant Mode



4.2 Lane Reversal

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

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

- **TRDCK_L:** Logical Lane for Redundant Clock/Track Transmitter
- **RRDCK_L:** Logical Lane for Redundant Clock/Track Receiver
- **TVLD_P, RVLD_P:** Physical Lane for Valid Transmitter and Receiver
- **TRDVLD_P, RRDVLD_P:** Physical Lane for Redundant Valid Transmitter and Receiver
- **TVLD_L, RVLD_L:** Logical Lane for Valid Transmitter and Receiver
- **TRDVLD_L, RRDVLD_L:** Logical Lane for Redundant Valid Transmitter and Receiver

Devices must support Lane reversal of data Lanes within a Module. An example of Lane reversal is when physical Data Lane 0 on local die is connected to physical Data Lane (N-1) on the remote die (physical Data Lane 1 is connected to physical Data Lane N-2 and so on) where N = 8 for a x8 Standard Package, N = 16 for a x16 Standard Package, N = 32 for a x32 Advanced Package, and N = 64 for a x64 Advanced Package. Redundant Lanes, in case of Advanced Package, are also reversed. Lane reversal must be implemented on the Transmitter only. The Transmitter reverses the logical Lane order on Data and Redundant Lanes.

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

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

4.2.1 Lane ID

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

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

In [Table 4-3](#), for a x16 Standard Package module, the Lane ID for `TD_L[0:15]/RD_L[0:15]` will be represented by the set of Lane ID {0...15} respectively. For a x8 Standard Package module, the Lane ID for `TD_L[0:7]/RD_L[0:7]` will be represented by the set of Lane ID {0...7} respectively.

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

Logical Lane Number	Lane ID	Logical Lane Number	Lane ID
0	00000000b	34	00100010b
1	00000001b	35	00100011b
2	00000010b	36	00100100b
3	00000011b	37	00100101b
4	00000100b	38	00100110b
5	00000101b	39	00100111b
6	00000110b	40	00101000b
7	00000111b	41	00101001b
8	00001000b	42	00101010b

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

Logical Lane Number	Lane ID	Logical Lane Number	Lane ID
9	00001001b	43	00101011b
10	00001010b	44	00101100b
11	00001011b	45	00101101b
12	00001100b	46	00101110b
13	00001101b	47	00101111b
14	00001110b	48	00110000b
15	00001111b	49	00110001b
16	00010000b	50	00110010b
17	00010001b	51	00110011b
18	00010010b	52	00110100b
19	00010011b	53	00110101b
20	00010100b	54	00110110b
21	00010101b	55	00110111b
22	00010110b	56	00111000b
23	00010111b	57	00111001b
24	00011000b	58	00111010b
25	00011001b	59	00111011b
26	00011010b	60	00111100b
27	00011011b	61	00111101b
28	00011100b	62	00111110b
29	00011101b	63	00111111b
30	00011110b	64	01000000b
31	00011111b	65	01000001b
32	00100000b	66	01000010b
33	00100001b	67	01000011b

Table 4-3. Lane ID: Standard Package Module

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

4.3 Interconnect redundancy remapping

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

4.3.1 Data Lane repair

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

Lane remapping is accomplished by “shift left” or “shift right” operation. A “shift left” is when data traffic of logical Lane $\text{TD_L}[n]$ on $\text{TD_P}[n]$ is multiplexed onto $\text{TD_P}[n-1]$. A shift left puts $\text{TD_L}[0]$ onto $\text{TRD_P}0$ or $\text{TD_L}[32]$ onto $\text{TRD_P}2$. A shift right operation is when data traffic $\text{TD_L}[n]$ is multiplexed onto $\text{TD_P}[n+1]$. A shift right puts $\text{TD_L}[31]$ onto $\text{TRD_P}1$ or $\text{TD_L}[63]$ onto $\text{TRD_P}3$. See the pseudo codes in [Section 4.3.3.1](#) and [Section 4.3.3.2](#) that show the changes in mapping post repair.

Note: If the lower index redundant Lane ($\text{TRD_P}[0]$ or $\text{TRD_P}[2]$) is faulty, no data lanes can be repaired for its group. Note that if the higher index redundant lane ($\text{TRD_P}[1]$ or $\text{TRD_P}[3]$) is faulty, one data lane can be repaired for its group.

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

[Figure 4-12](#) shows transmit bump side of data Lane remapping for the first group of 32 Lanes. Both “shift left” and “shift right” remapping is needed to optimally repair up to any two Lanes within the group. [Figure 4-13](#) shows details of the mux structure used for data Lane repair.

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

Figure 4-12. Data Lane remapping possibilities to fix potential defects

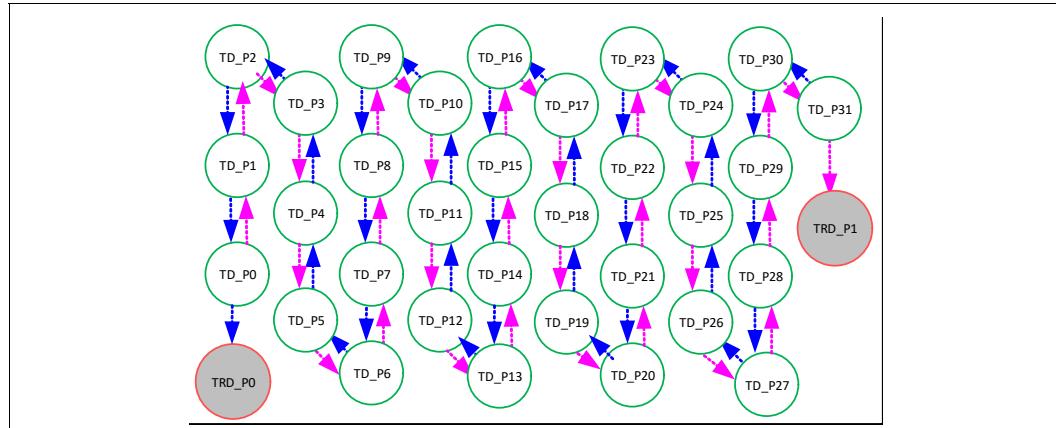
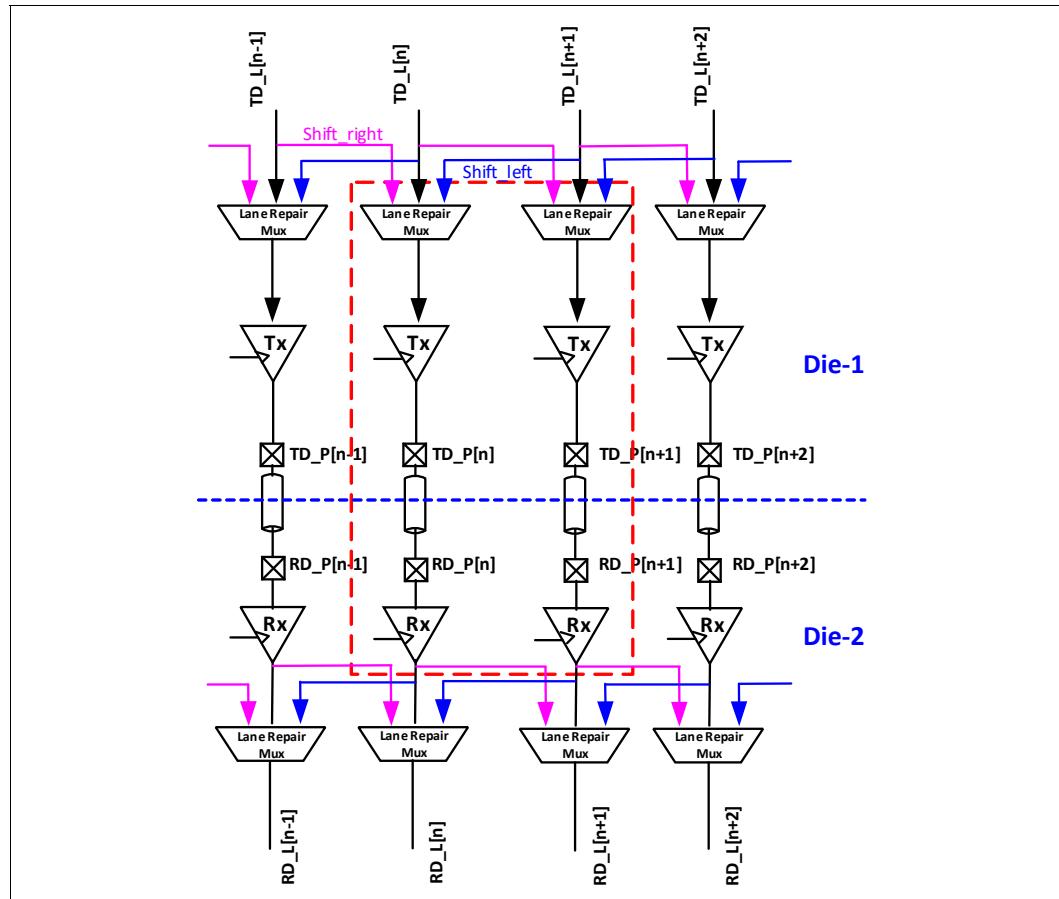


Figure 4-13. Data Lane remapping: Mux chain

4.3.2 Data Lane repair with Lane reversal

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

4.3.3 Data Lane repair implementation

4.3.3.1 Single Lane repair

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

Pseudo code for repair in `TD_P[31:0] (RD_P[31:0])` ($0 \leq x \leq 31$):

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

Pseudo code for repair in `TD_P[63:32] (RD_P[63:32])` ($32 \leq x \leq 63$) (this does not apply to x32 Advanced Package Link):

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

As shown in Figure 4-14 TD_P[29] is remapped in the direction to use TRD_P[0] as the repair resource. Figure 4-15 shows the circuit implementation.

Figure 4-14. Example of Single Lane failure remapping

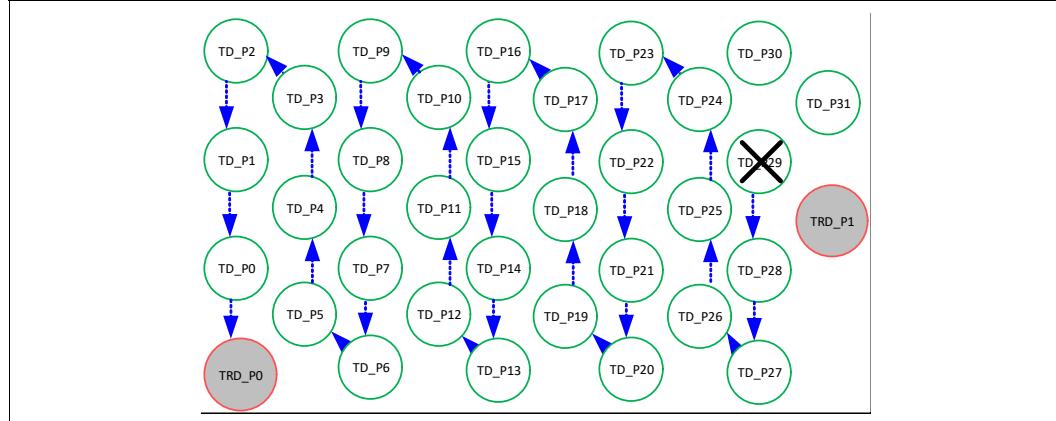
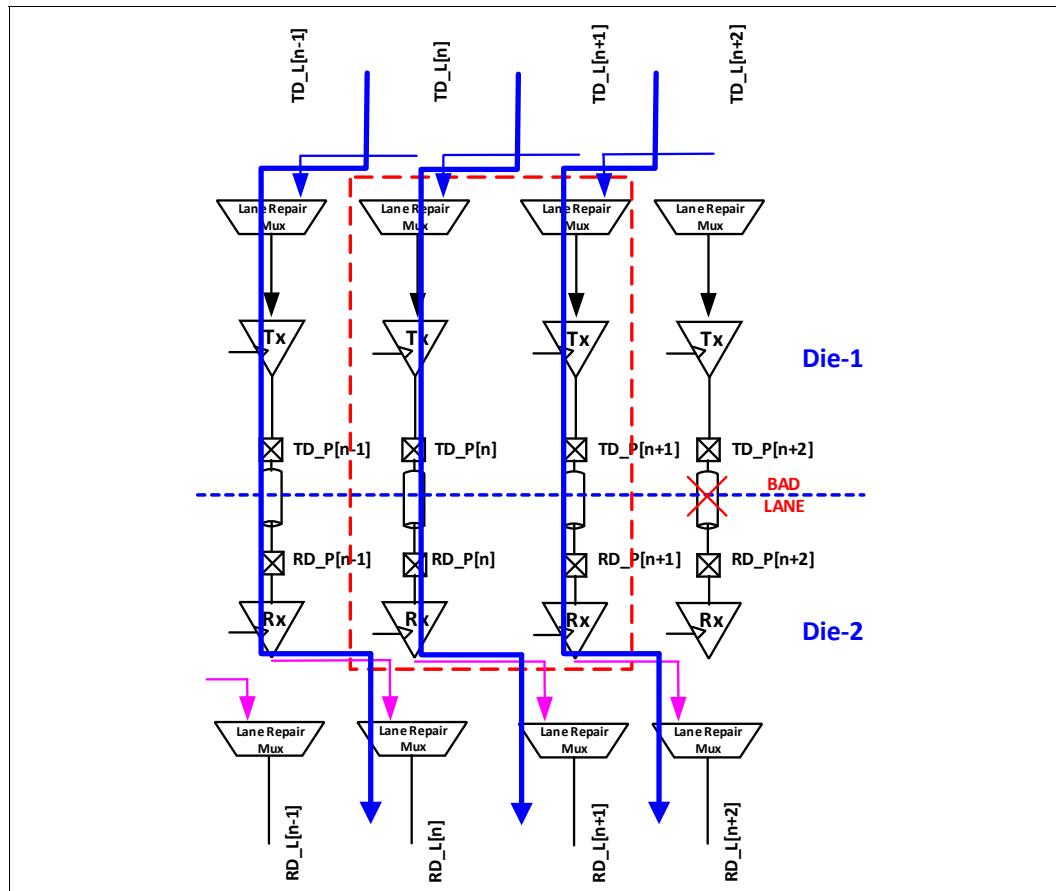


Figure 4-15. Example of Single Lane remapping implementation



4.3.3.2 Two Lane repair

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

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

```

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

```

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

```

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

```

Shown in Figure 4-16 is an example of two (physical Lanes 25 and 26) Lane remapping. Figure 4-17 shows the circuit implementation. Both Transmitter and Receiver must apply the required remapping.

Figure 4-16. Example of Two Lane failure remapping

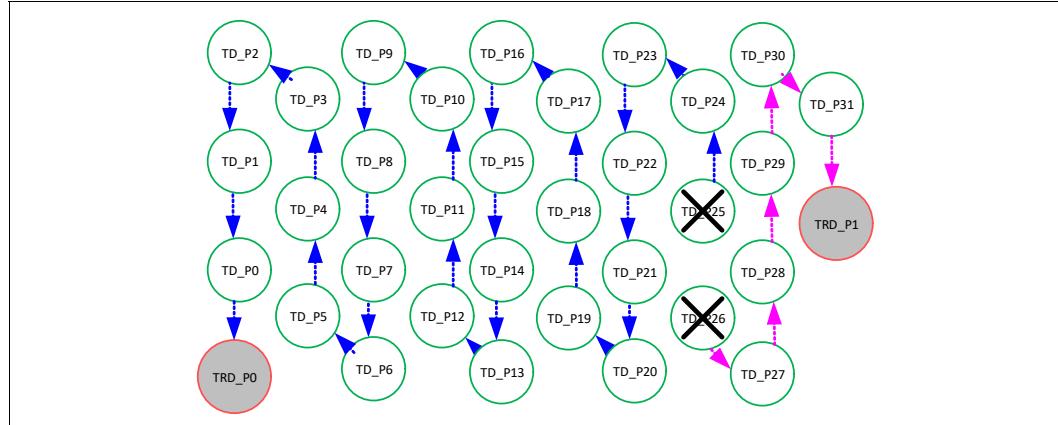
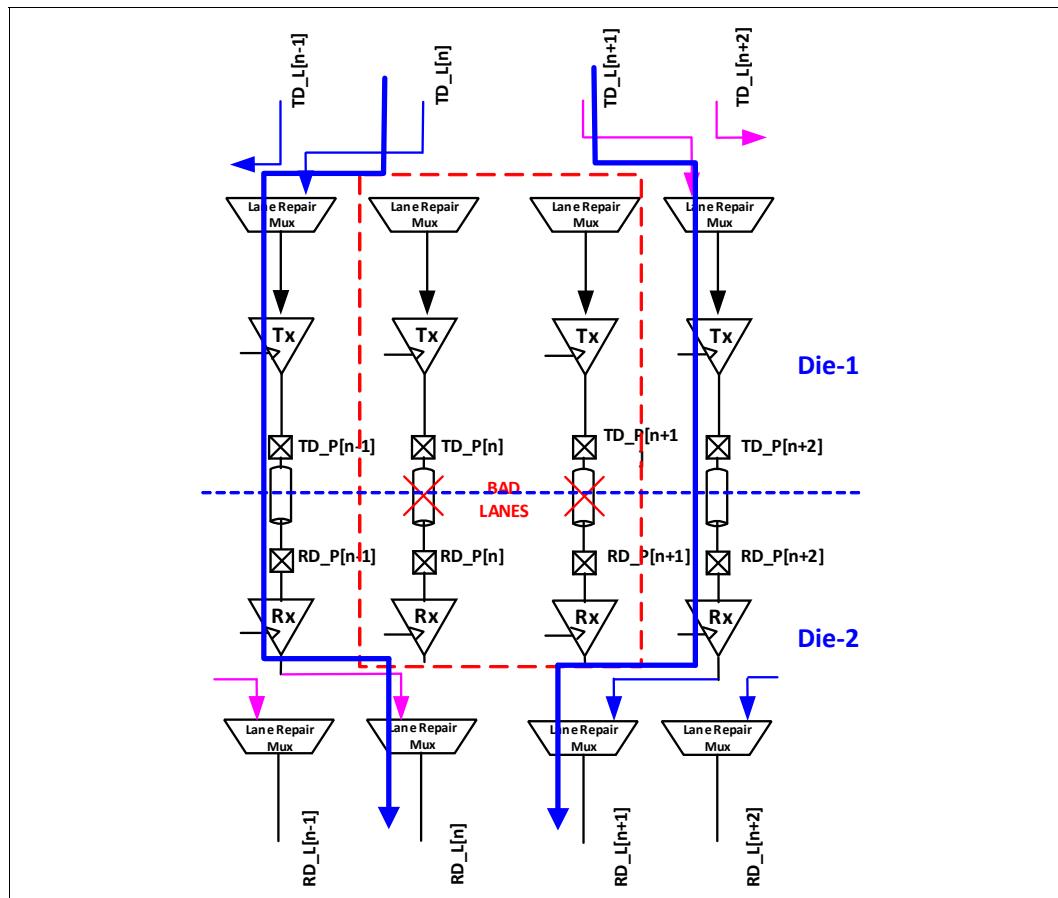


Figure 4-17. Example of Two Lane remapping implementation



4.3.3.3 Single Lane repair with Lane reversal

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

4.3.3.3.1 x64 Advanced Package Pseudo Codes

Pseudo code for one Lane failure in `TD_P[31:0]` (`RD_P[32:63]`) ($0 \leq x \leq 31$):

```

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

Pseudo code for one Lane failure in `TD_P[63:32]` (`RD_P[0:31]`) ($32 \leq x \leq 63$):

```

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

4.3.3.3.2 x32 Advanced Package Pseudo Codes

Pseudo code for one-Lane failure in `TD_P[31:0]` (`RD_P[0:31]`) ($0 \leq x \leq 31$):

```

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

4.3.3.4 Two Lane repair with Lane reversal

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

4.3.3.4.1 x64 Advanced Package Pseudo Codes

Pseudo code for two Lane failure in `TD_P[31:0]` (`RD_P[32:63]`) ($0 \leq x \leq 31$):

```

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

```

Pseudo code for two-Lane failure in `TD_P[63:32]` (`RD_P[0:31]`) ($32 \leq x \leq 63$):

```

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

```

4.3.3.4.2 x32 Advanced Package Pseudo Codes

Pseudo code for two-Lane failure in TD_P[31:0] (RD_P[0:31]) ($0 \leq x \leq 31$):

```

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

```

4.3.4 Clock and Track Lane remapping

The specification supports remapping of one broken Lane for **TCKP_P/RCKP_P**, **TCKN_P/RCKN_P** and **TTRK_P/RTRK_P** physical Lanes. The repair scheme is shown in [Figure 4-18](#). Clock Lane remapping allows repair of single Lane failure for both differential and pseudo-differential implementation of the clock Receiver. The circuit details are shown in [Figure 4-19](#) and [Figure 4-20](#) for differential and pseudo-differential clock Receivers respectively.

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

Figure 4-18. Clock and Track repair

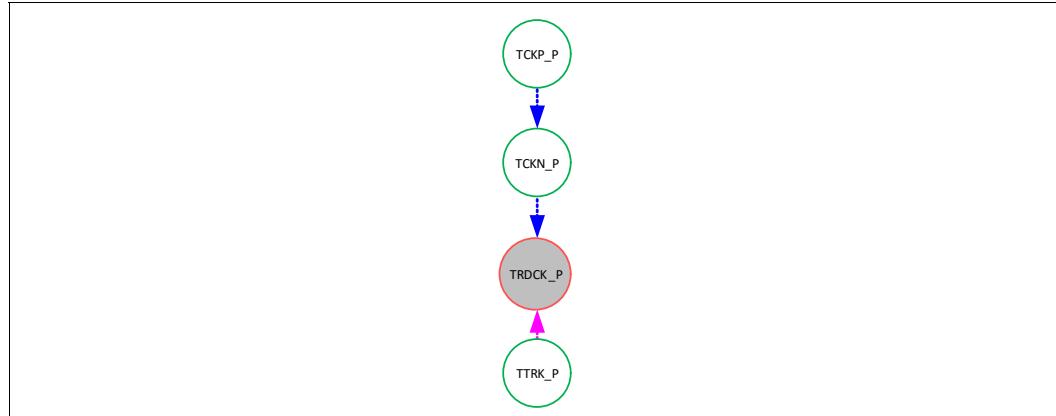


Figure 4-19. Clock and track repair: Differential Rx

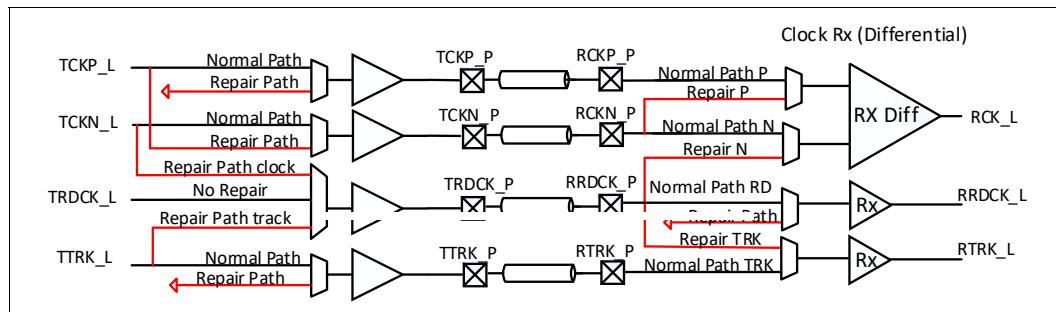
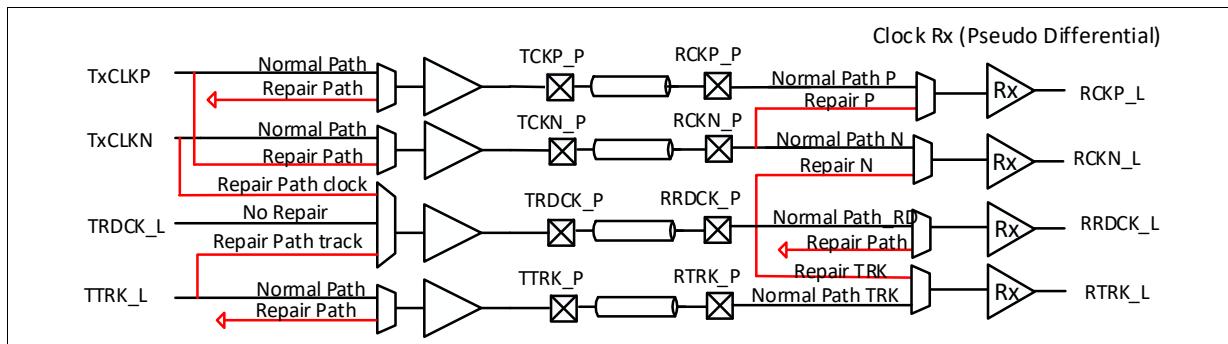


Figure 4-20. Clock and track repair: Pseudo Differential Rx



4.3.5 Clock and Track Lane repair implementation

Pseudo code for Clock and Track Lane repair:

```

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

```

The implementation of Clock and Track Lane remapping is shown in [Figure 4-21\(a\)](#), [Figure 4-21\(b\)](#) and [Figure 4-21\(c\)](#) respectively. The corresponding circuit level details of remapping implementation are shown in [Figure 4-22](#), [Figure 4-23](#) and [Figure 4-24](#).

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

Figure 4-21. Clock and Track Lane repair scheme

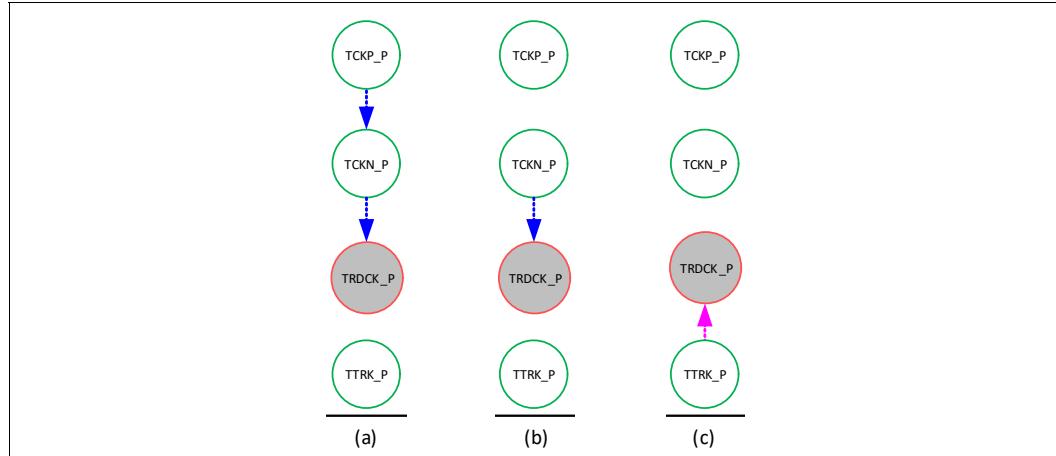


Figure 4-22. Clock and track repair: CKP repair

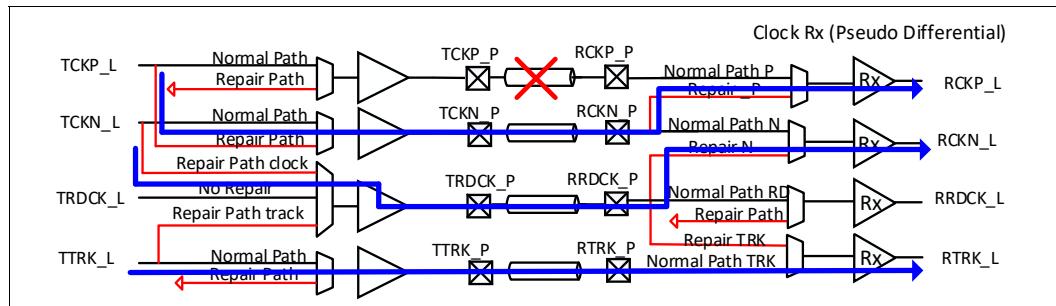
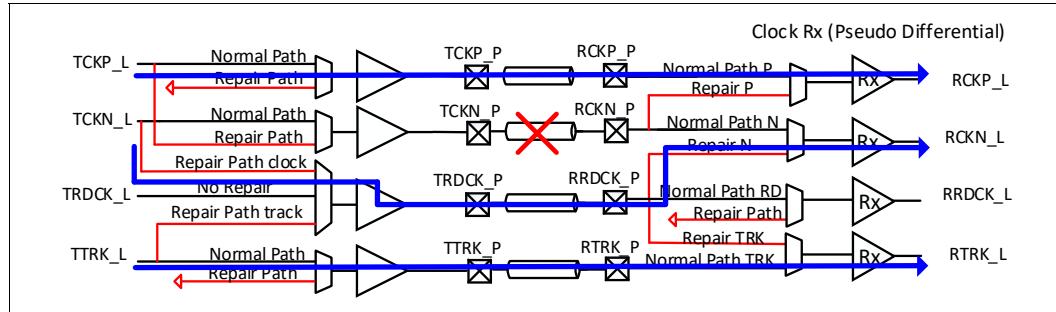
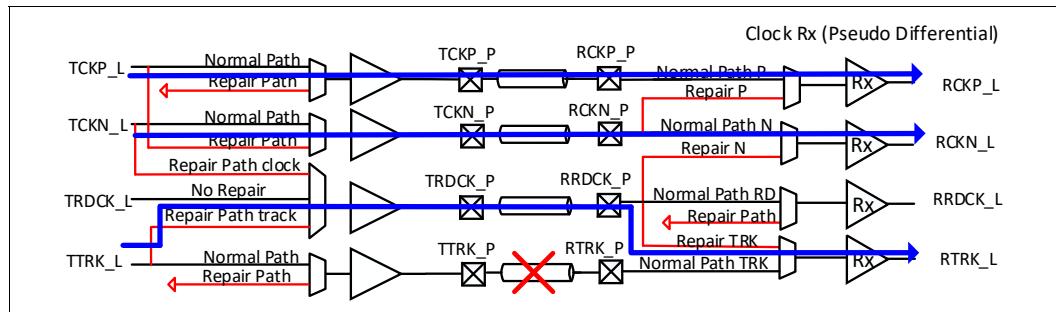


Figure 4-23. Clock and track repair: CKN repair**Figure 4-24. Clock and track repair: Track repair**

4.3.6 Valid Repair and implementation

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

Figure 4-25 shows the normal path for Valid and redundant valid Lanes. Figure 4-26 shows the repair path for Valid Lane failure.

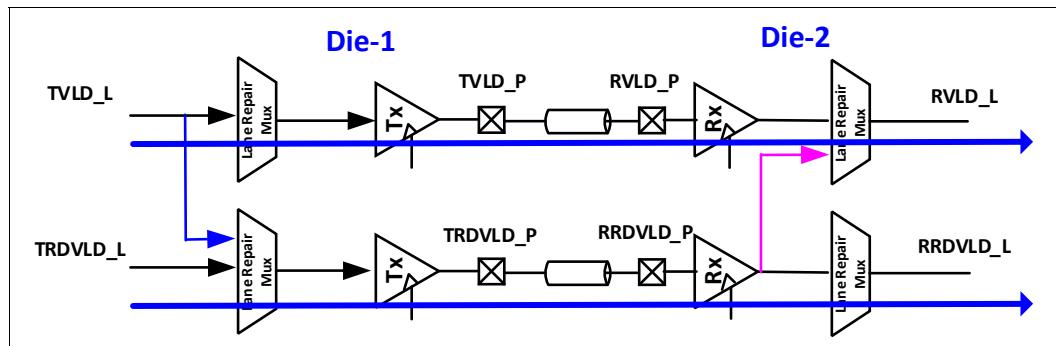
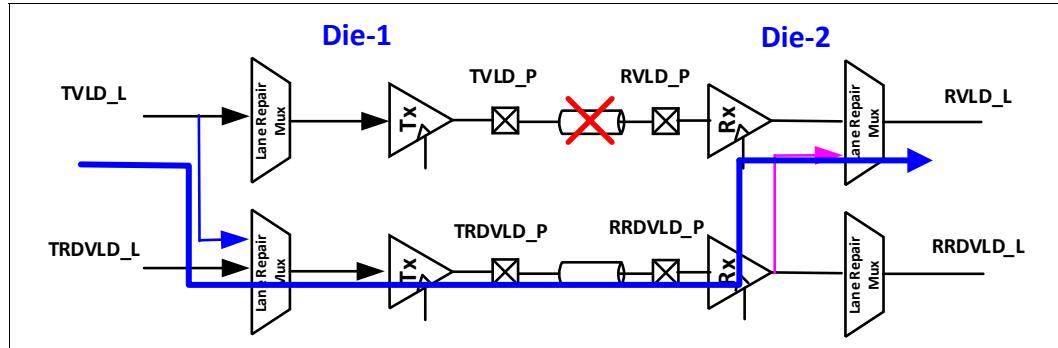
Figure 4-25. Valid repair: Normal Path

Figure 4-26. Valid Repair: Repair Path

4.3.7 Width Degrade in Standard Package Interfaces

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

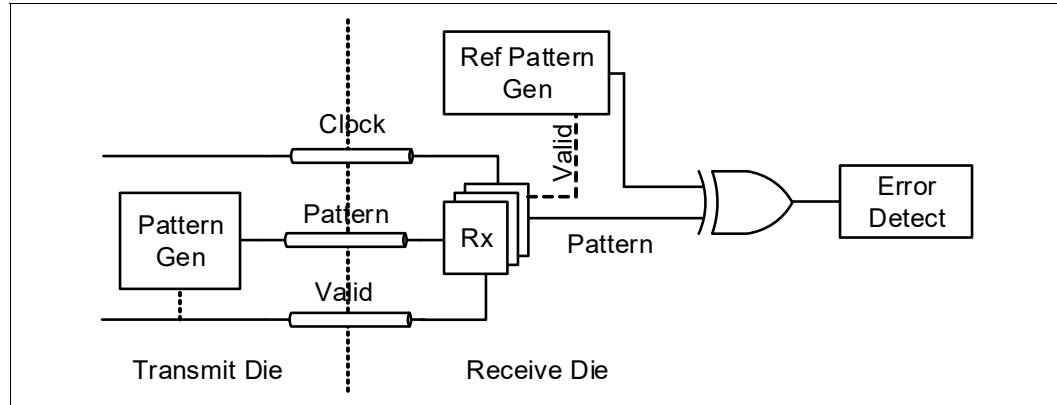
In the case of x8 Standard Package modules, resilience against faulty Lanes is provided by configuring the Link to a x4 width (Logical Lanes 0 to 3 or Logical Lanes 4 to 7, which exclude the faulty Lanes). The configuration is done during Link initialization or retraining. Transmitters of the disabled Lanes are tri-stated and Receivers are disabled.

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

4.4 Data to Clock Training and Test Modes

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

Figure 4-27 shows the infrastructure for interface training and testing. The Transmit Die and Receive Die implement the same Linear Feedback Shift Register (LFSR) described in Section 4.4.1. The pattern sent from the Transmitter along with forwarded clock and Valid is compared with locally generated reference pattern. Both transmit and receive pattern generators must start and advance in sync. The compare circuitry checks for matching data each UI. Any mismatch between the received pattern and pattern predicted by the local pattern generator is detected as an error.

Figure 4-27. Test and training logic

The receive die must implement two types of comparison schemes

- **Per-Lane comparison:** Per-Lane comparison is used to identify the number of failing Lanes between the two dies. Any mismatch, above the set threshold between the received pattern and the expected pattern on each Lane, will set an internal error detect bit for each Lane. Once a pattern mismatch on a particular Lane is found, this error bit is set for the remainder of the test. [Figure 4-28](#) illustrates Per-Lane comparison mode. This mode will indicate per-Lane errors within the module ($N = 68$ ($64 + 4$ RD) for a x64 Advanced Package Module, $N = 34$ ($32 + 2$ RD) for a x32 Advanced Package Module, $N = 16$ for a x16 Standard Package Module, and $N = 8$ for a x8 Standard Package Module, respectively). The per-Lane comparison results can be read via sideband.
- **Aggregate comparison:** In this mode, pattern mismatches each UI on any Lane within the module are accumulated into a 16-bit error counter. The Lane errors are ORed to generate a module-level error and counted as shown in [Figure 4-29](#). This scheme can be used for module level margin and BER.

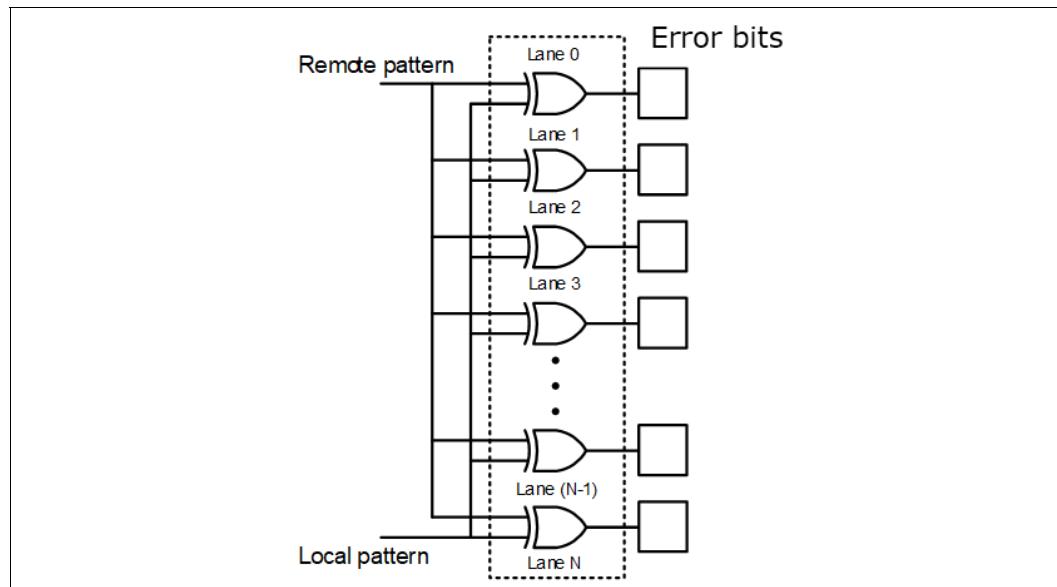
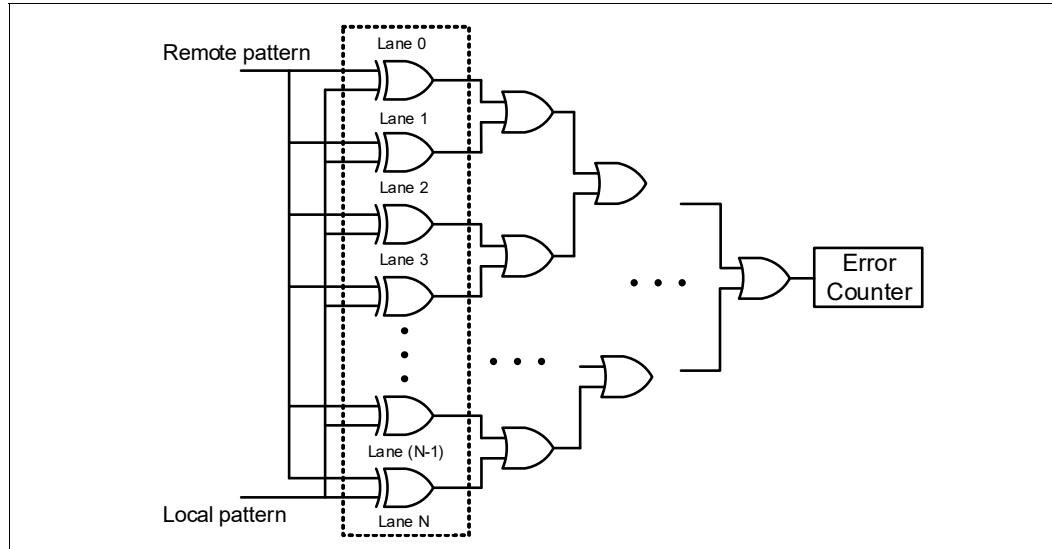
Figure 4-28. Lane failure detection

Figure 4-29. All Lane error detection

4.4.1 Scrambling and training pattern generation

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

The LFSR uses the same polynomial as PCIe: $G(X)=X^{23} + X^{21} + X^{16} + X^8 + X^5 + X^2 + 1$. Each Transmitter is permitted to implement a separate LFSR for scrambling and pattern generation. Each Receiver is permitted to implement a separate LFSR using the same polynomial for de-scrambling and pattern comparison. The implementation is shown in [Figure 4-30](#). The seed of the LFSR is Lane dependent, and based on the Logical Lane number and the seed value for Lane number is modulo 8 as shown in [Table 4-4](#).

Alternatively, implementations can choose to implement one LFSR with different tap points for multiple Lanes as shown in [Figure 4-31](#). This is equivalent to individual LFSR per-Lane with different seeds.

Table 4-4. LFSR seed values

Lane	Seed
0	23'h1DBFBC
1	23'h 0607BB
2	23'h1EC760
3	23'h18C0DB
4	23'h010F12
5	23'h19CF9
6	23'h0277CE
7	23'h1BB807
RD0, RD2 ^a	23'h18C0DB
RD1, RD3 ^b	23'h010F12

a. Same as Lane 3. These are not currently used.

b. Same as Lane 4. These are not currently used.

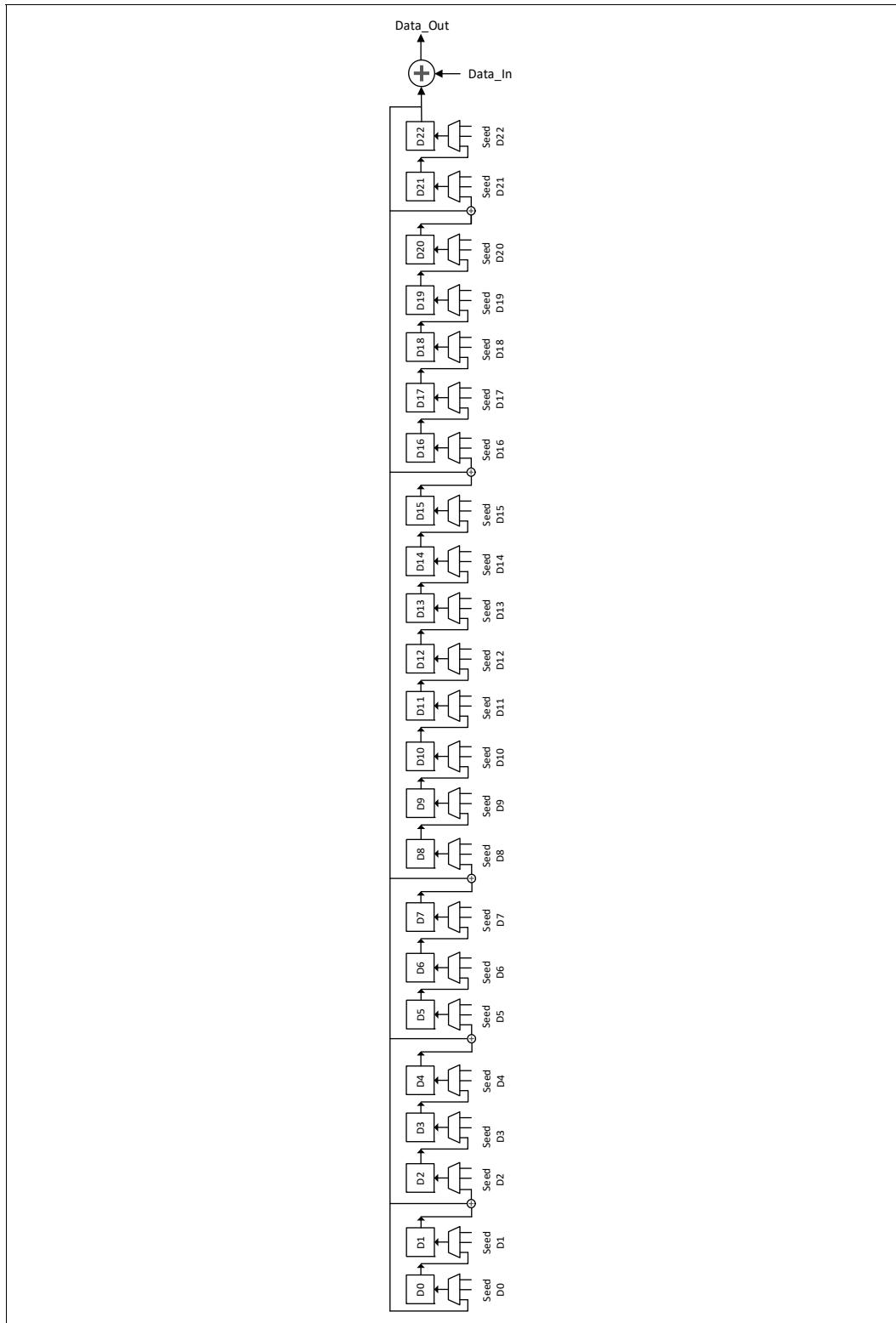
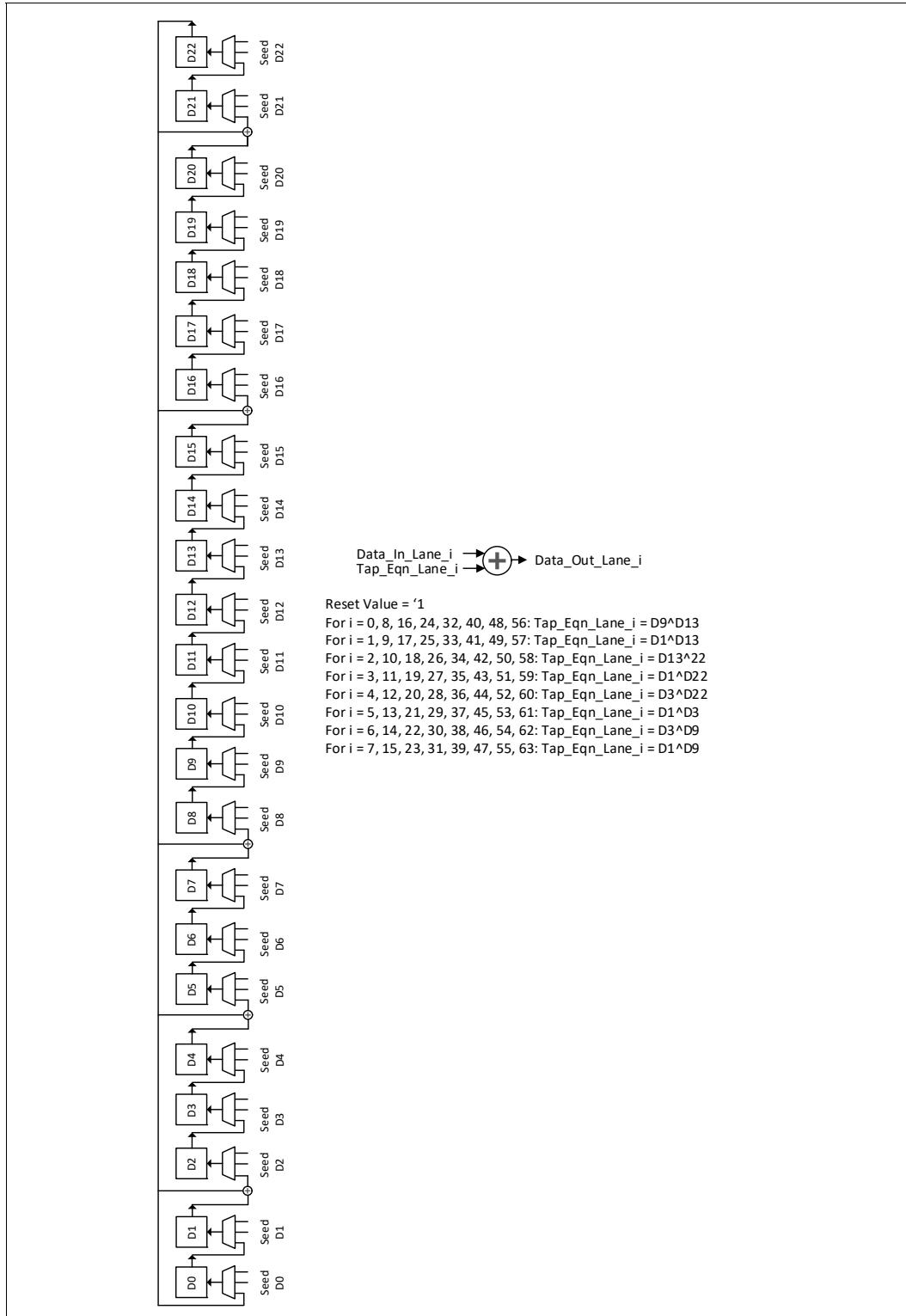
Figure 4-30. LFSR implementation

Figure 4-31. LFSR alternate implementation

4.5 Link Initialization and Training

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

- UCIe Module Partner: A UCIe Module Partner is the corresponding UCIe Module on the remote UCIe Die to which the UCIe Module connects. For example, if two UCIe Dies A and B are connected in the standard Die rotate configuration by a 2-module UCIe Link, Modules 0 and 1; UCIe Die A's Module 0 (UCIe Module A0) is connected to UCIe Die B's Module 1 (UCIe Module B1); then A0 is the UCIe Module Partner of B1 and vice-versa.
- Phase Interpolator (PI) in this specification is used to refer any method of generating and selecting sampling clock phase.
- Timeout: Every state except RESET, Active, L1/L2, and TRAINERROR in the Link Training State Machine has a residency timeout of 8 ms. For states with sub-states, the timeout is per sub-state (i.e., if Physical Layer is in a state for greater than 8 ms, then it transitions to TRAINERROR and eventually to RESET). Physical Layer sideband handshakes for RDI state transitions with remote Link partner also timeout after 8 ms. The timeout counters must be reset if a sideband message with "Stall" encoding is received. All timeout values specified are -0% and +50% unless explicitly stated otherwise. All timeout values must be set to the specified values after Domain Reset exit. All counter values must be set to the specified values after Domain Reset.
- Electrical idle is described in [Section 5.12](#).
- When Management Transport protocol is supported over the sideband, the term SB_MGMT_UP refers to an internally maintained flag in the PHY that indicates whether the Management Transport path has been successfully initialized with the partner chiplet. If the flag is cleared to 0, the Management Transport path is not up. If the flag is set to 1, the Management Transport path is up. The flag is cleared on a Management Reset or when a Heartbeat timeout is detected (see [Section 8.2.5.1.3](#)). The flag is set after successful completion of the initialization phase of Management Transport path setup as described in [Section 8.2.3.1.2](#).

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

- Triggers for initiating Link Training for Management Transport path setup on the sideband are:
 - Software writes 1 to the Retrain Link bit in the Sideband Management Port Structure register (see [Section 8.1.3.6.2.1](#))
 - HW-autonomous trigger by the Management Port Gateway to automatically train the Management Transport path on the sideband without SW intervention, as occurs after Management Reset
 - SBINIT pattern (two consecutive iterations of 64-UI clock pattern and 32-UI low) is observed on any sideband Receiver clock/data pair, when the SB_MGMT_UP flag is cleared to 0
- The triggers for initiating Link Training for the mainband are:
 - Software writes 1 to Start UCIe Link Training bit in UCIe Link Control register in the UCIe Link DVSEC (see [Section 9.5.1.5](#))
 - When Management Transport protocol is supported on the UCIe link, software writes 1 to the Retrain Link bit in the Management Port Structure register of either the Sideband or Mainband Management Port that is associated with the UCIe link (see [Section 8.1.3.6.2.1](#))
 - Adapter triggers Link Training on the RDI (RDI status is Reset and there is a NOP to Active transition on the state request)

- If [Management Transport protocol is not supported] OR [the SB_MGMT_UP flag is cleared to 0], the SBINIT pattern (two consecutive iterations of 64-UI clock pattern and 32-UI low) is observed on any sideband Receiver clock/data pair
- If [Management Transport protocol is supported] AND [the SB_MGMT_UP flag is set to 1], the {SBINIT done req} sideband message was received from the module partner

If hardware fails training after an implementation-specific number of attempts, the hardware must transition to RESET and wait for a subsequent Link Training trigger. Physical Layer must escalate a fatal error to the D2D Adapter on the RDI if mainband Software-triggered or RDI-triggered Link training fails or there is a Link-up-to-Link-down transition due to a Physical Layer timeout.

Throughout this section, references to mainband transmitter and receiver behavior are called out in various state machine states. These references do not apply when the port does not have a mainband (i.e., the port is a sideband-only port without a physically present mainband, or the port is a UCIe-S port with only the sideband used. In the latter scenario, the mainband transmitters are held in tri-state throughout and the mainband receivers are disabled.

4.5.1 Link Training Basic Operations

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

4.5.1.1 Transmitter initiated Data to Clock Point Test

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

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

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

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

5. The UCIe Module requests its UCIe Module Partner for the logged results in [Step 4](#) by sending {Tx Init D to C results req} sideband message. The UCIe Module Partner stops comparison on its mainband Receivers and responds with the logged results {Tx Init D to C results resp} sideband message.
6. The UCIe Module stops sending the pattern on its Transmitters and sends the {End Tx Init D to C point test req} sideband message and the UCIe Module Partner responds with {End Tx Init D to C point test resp}. When a UCIe Module has received the {End Tx Init D to C point test resp} sideband message, the corresponding sequence has completed.

Table 4-5. Data-to-Clock Training Parameters^a

Parameter	Sideband Message Field	Options
Clock sampling /PI phase control	Clock Phase control at Transmitter	Eye Center found by training
		Eye Left Edge found by training
		Eye Right Edge found by training
Tx Pattern Generator Setup	Data Pattern (for Data Lanes)	LFSR Pattern for Data Lanes
		Per-Lane ID pattern for Data Lanes as defined in Table 4-7
	Valid Pattern (for Valid Lanes)	VALTRAIN pattern four 1s followed by four 0s
Tx Pattern Mode Setup	Pattern Mode	Burst Mode: Uses Burst Count/Idle Count/Iteration Count
		Continuous Mode: Uses Burst count to indicate the number of UI of transmission. Idle Count = 0, Iteration Count = 1
Rx Compare Setup	Maximum Comparison error threshold	Maximum comparison error threshold
	Comparison Mode	Per-Lane or aggregate error compare

- a. See [Table 7-11](#) for the encodings. See also the registers in [Section 9.5.3.26](#) and [Section 9.5.3.27](#). See also the Implementation Note below this table.

IMPLEMENTATION NOTE

The Training Setup 1 and Training Setup 2 registers (see [Section 9.5.3.26](#) and [Section 9.5.3.27](#), respectively) are applicable for compliance or debug. For regular operation, implementations must follow the comparison mode, iteration or UI count specified in the corresponding states (for example, [Section 4.5.3.4.8](#) specifies 4K UI of continuous mode LFSR pattern and total (aggregate) error count); and because these patterns are fixed, Rx is permitted to ignore Burst Count/Idle Count/Iteration Count values in the sideband messages for regular operation. Training Setup 3 and Training Setup 4 registers (see [Section 9.5.3.28](#) and [Section 9.5.3.29](#), respectively) are applicable for regular operation as well as compliance and debug.

Additionally, implementation notes for the parameters in [Table 4-5](#):

- Clock sampling/PI phase control: For Tx-initiated Data to Clock point tests, this parameter is for informational purposes only. For Rx-initiated Data to Clock point tests, the Transmitter sets the PI phase for the requested setting (Eye Left edge, Eye Right edge or Eye Center) based on the best known values from the most recent training for the corresponding speed. For the states of MBTRAIN.VALVREF and MBTRAIN.DATAVREF, this would be the best known values at the Transmitter at 4 GT/s. This field is not applicable and is ignored for Data to Clock eye sweep tests.
- Tx Pattern Generator Setup and Tx Pattern Mode Setup:
 - For Continuous mode LFSR pattern, “Burst Count” indicates the number of UI of transmission (for example, it is 4096 (or 4K) in [Section 4.5.3.4.8](#)). “Idle Count” is always 0 and “Iteration Count” is always 1.
 - For Continuous mode Per-Lane ID pattern, “Burst Count” indicates the number of UI of transmission (e.g., it is 2048 for 128 iterations of the 16 bit pattern). “Idle Count” is always 0 and “Iteration Count” is always 1.
 - For Burst mode, the transmission is a “Burst Count” number of UI followed by an “Idle Count” number of UI of 0 repeating for an “Iteration Count” number of times. In the case of Per-Lane ID pattern, the “Burst Count” must be a multiple of the pattern size in UI. Burst mode is currently not used for regular operation of Link Training.
 - VALTRAIN pattern is used for training of the Valid lane. In Continuous mode, “Burst Count” indicates the number of UI of transmission (for example, it is 1024 for 128 iterations of the 8-bit pattern); “Idle Count” is always 0 and “Iteration Count” is always 1. In Burst mode, the transmission is a “Burst Count” number of UI followed by an “Idle Count” number of UI of 0 repeating for an “Iteration Count” number of times; the “Burst Count” must be a multiple of the pattern size in UI.

4.5.1.2 Transmitter initiated Data to Clock Eye Width Sweep

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

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

The sequence of steps for this test are as follows:

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

4.5.1.3 Receiver initiated Data to Clock point test

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

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

The sequence of steps for this test are as follows:

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

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

4.5.1.4 Receiver initiated Data to Clock Eye Width Sweep

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

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

The sequence of steps for this test are as follows:

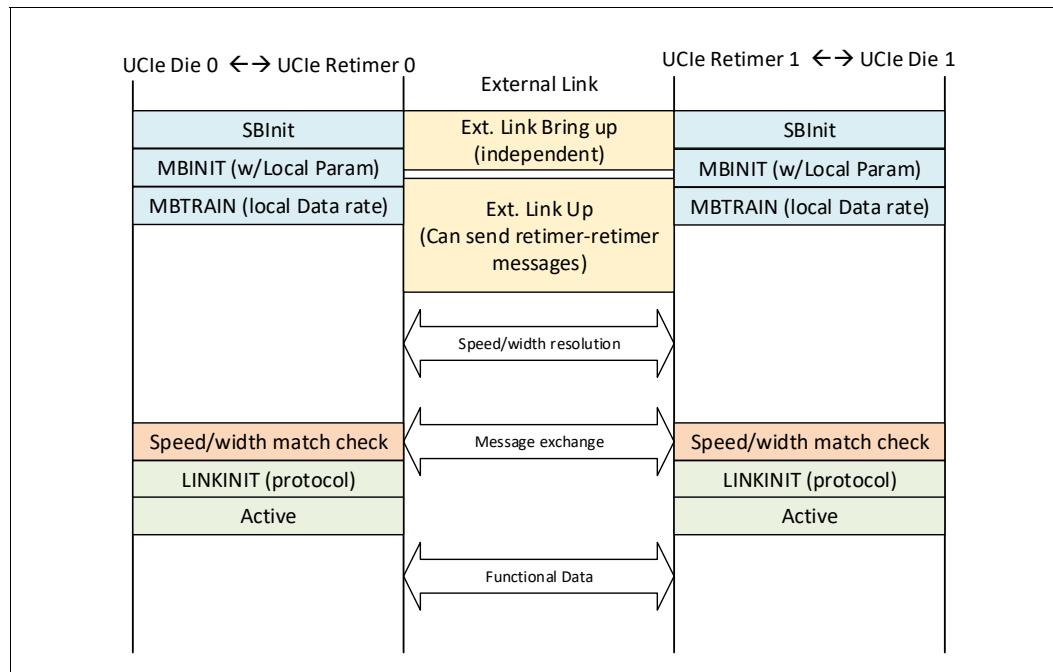
1. The UCIe Module enables the pattern comparison circuits to compare incoming mainband data to the locally generated expected pattern, sets up the Receiver parameters (shown in [Table 4-5](#)), sends a {Start Rx Init D to C eye sweep req} sideband message to its UCIe Module Partner, and then waits for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

7. The UCIe Module Partner sends an {Rx Init D to C sweep done with results} sideband message with results for its mainband Transmitter. The UCIe Module can use the sweep results for its mainband Receivers.
8. The UCIe Module sends an {End Rx Init D to C eye sweep req} sideband message and the UCIe Module Partner responds with an {End Rx Init D to C eye sweep resp} sideband message. When a UCIe Module has received the {End Rx Init D to C eye sweep resp} sideband message, the corresponding sequence has completed.

4.5.2 Link Training with Retimer

The following diagram explains the initialization flow with UCIe Retimer. As shown in Figure 4-32, external and UCIe (UCIe Die to UCIe Retimer) Links are permitted to come up independently. When a UCIe Link trains up to local data rate and width, the remote UCIe information is requested over the external Link. If there is a data rate and width difference, each UCIe Link is permitted to be retrained to achieve a speed (data rate) and width match (if the UCIe Retimer requires this for operation, it must initiate Retraining from LinkSpeed state). This can happen multiple times during initial Link bring up or retraining. Once the UCIe Retimer has determined that the UCIe Link configuration is suitable for successful operation with remote Retimer partner, the UCIe Link proceeds to ACTIVE through protocol level Link initialization (LINKINIT).

Figure 4-32. Example Retimer bring up when performing speed/width match

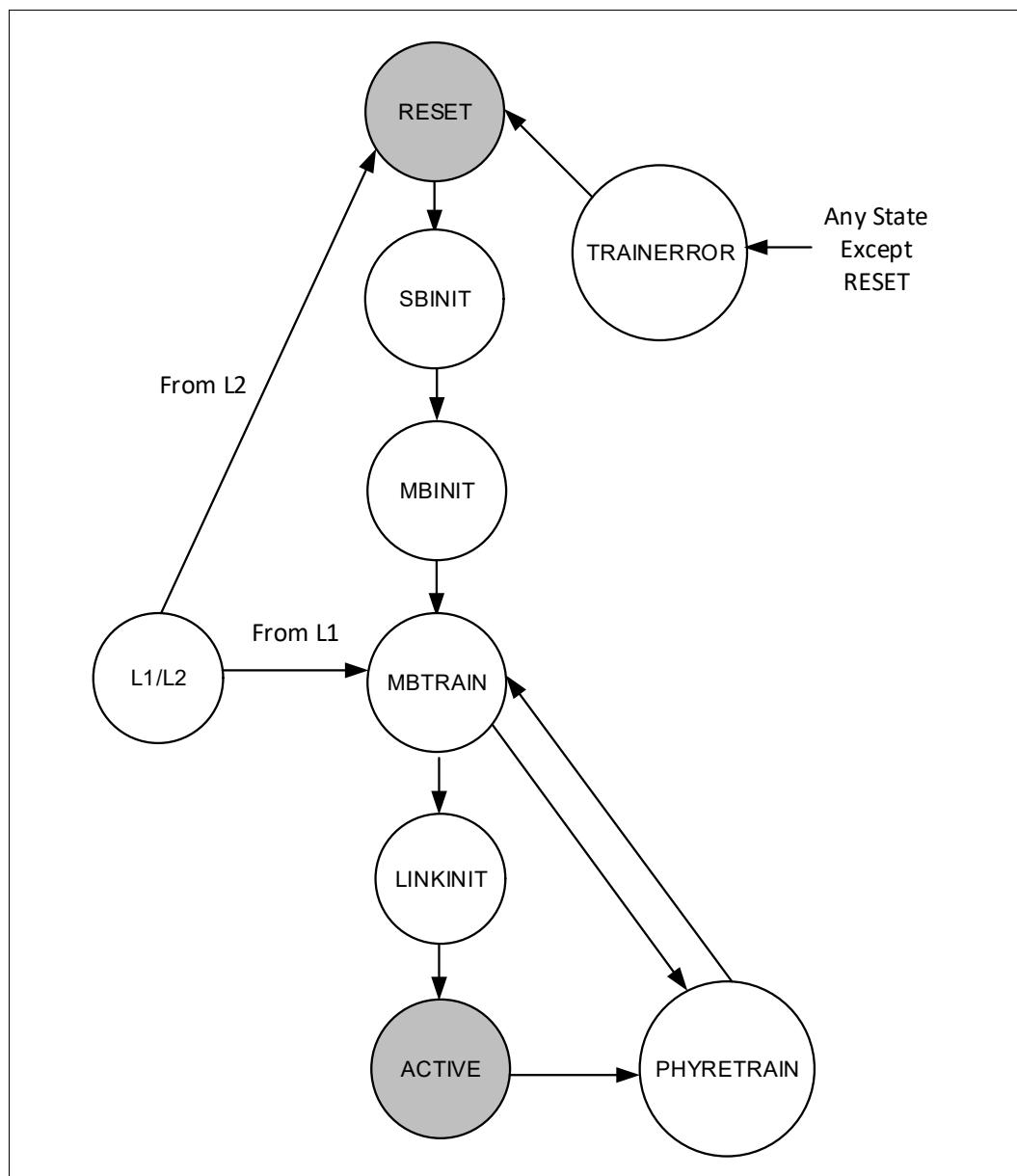


4.5.3 Link Training State Machine

A high level initialization flow is shown in Figure 4-33. A high level description of each state is shown in Table 4-6. The details and actions performed in each state are described in the following sections.

Table 4-6. State Definitions for Initialization

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

Figure 4-33. Link Training State Machine

4.5.3.1 RESET

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

- Power supplies are stable
- Sideband clock is available and running at 800 MHz
- If {
 - Physical Layer and Die to Die Adapter internal clocks are stable and available
 - Mainband clock speed is set to the slowest I/O data rate (2 GHz for 4 GT/s)
 - Local SoC/Firmware not keeping the Physical Layer in RESET
 - Link training trigger for the mainband has occurred (triggers are defined in the beginning of [Section 4.5](#))
} OR
 - {
 - Sideband Management Transport protocol is supported
 - SB_MGMT_UP = 0
 - Local SoC/Firmware is not keeping the sideband in RESET
 - Management Port Gateway indicates that it is ready for Management Transport path initialization
 - Link Training trigger for the Management Transport path has occurred (triggers are defined in the beginning of [Section 4.5](#))
}
- Data, Valid, Clock, and Track Transmitters are tri-stated
- Data, Valid, Clock, and Track Receivers are permitted to be disabled
- If [Management Transport protocol is not supported] OR [SB_MGMT_UP=0], Sideband Transmitters are held low
- Sideband Receivers are enabled

If [Management Transport protocol is supported] AND [SB_MGMT_UP=1], the Physical Layer is available for sideband packet (any sideband packets including Management Port messages) transmission/reception in the RESET state.

If [Management Transport protocol is not supported] OR [SB_MGMT_UP=0]:

- While the LTSM is in the RESET state, a Physical Layer implementation is permitted to handle any new or pending untransmitted sideband register access requests as Unsupported Requests, and if doing so, it must return a corresponding completion as well as end-to-end credit and RDI credit to the Adapter. Similarly, for register access completions, a Physical Layer implementation is permitted to discard any new or pending untransmitted completions and because completions do not consume a credit, there is no credit return back to the Adapter for these. While the LTSM is in the RESET state, an implementation may handle any new or pending untransmitted Vendor-defined messages in an implementation dependent manner. This could include dropping the message and logging an error.

4.5.3.2 Sideband Initialization (SBINIT)

In this state, the sideband (SB) interface is initialized and repaired (when applicable).

If Management Transport is supported AND the SB_MGMT_UP flag is set to 1:

- Initialization and repair of the sideband ([Step 1](#) through [Step 9](#) for Advanced Package and [Step 1](#) through [Step 7](#) for Standard Package as shown later in this section) are skipped, and only an SBINIT Done Req/Resp handshake is performed in this state (see [Step 10](#) for Advanced Package flow and [Step 8](#) for Standard Package flow as shown later in this section); otherwise, all the steps are traversed
- Physical Layer sideband is available for sideband packet (any sideband packets including Management Port messages) transmission/reception in this state

When not performing the actions relevant to this state:

- Data, Valid, Clock, and Track Transmitters remain tri-stated
- Data, Valid, Clock, and Track Receivers are permitted to be disabled
- If [Management Transport protocol is not supported] OR [the SB_MGMT_UP flag is cleared to 0] OR [a Sideband Packet is not being sent], SB Transmitters continue to be held Low
- SB Receivers continue to be enabled

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

If [Management Transport protocol is not supported] OR [SB_MGMT_UP=0]:

- While the LTSM is in the SBINIT state, a Physical Layer implementation is permitted to wait for an implementation specific amount of time to see whether SBINIT is progressing or completing, and subsequently handle any new or pending untransmitted sideband register access requests as Unsupported Requests, and if treating the request as an Unsupported Request, it must return a corresponding completion as well as end-to-end credit and RDI credit to the Adapter. Similarly, for register access completions, a Physical Layer implementation is permitted to wait for an implementation specific amount of time to see whether SBINIT is progressing or completing, and subsequently discard any new or pending untransmitted completions and because completions do not consume a credit, there is no credit return back to the Adapter for these. While the LTSM is in the SBINIT state, an implementation may handle any new or pending untransmitted Vendor-defined messages in an implementation dependent manner. This could include dropping the message and logging an error.

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

- The UCIe Module must start and continue to send iterations of a 64-UI clock pattern (a clock pattern is defined as starting with 1 and toggling every UI of transmission, i.e., 1010...) and 32-UI low on both sideband data Transmitters (**TXDATASB** and **TXDATASBRD**). The UCIe Module must send strobes on both **TXCKSB** and **TXCKSBRD** during the 64-UI clock pattern transmission and be gated (held low) otherwise.
- UCIe Module Partner must sample each incoming data patterns on its sideband Receivers with both incoming sideband clocks (this forms four Receiver/clock combinations).
- A sideband data-clock Receiver combination detection is considered successful if 128 UI clock pattern is detected.

4. If a UCIe Module Partner detects the pattern successfully on at least one of its sideband data-clock Receiver combination, it must stop sending data and clock on its sideband Transmitters after four more iterations of 64-UI clock pattern and 32-UI low. This will allow for any time differences in both UCIe Module and UCIe Module Partner coming out of RESET state. The sideband Transmitter and Receiver on the UCIe Module must now be enabled to send and receive sideband messages
5. If pattern is not detected on its sideband Receiver, the UCIe Module must continue to alternate between sending the pattern on its sideband Transmitters for 1 ms, and holding low for 1 ms, for a total of 8 ms. The sideband Receiver of the UCIe Module must remain enabled during this time. Timeout occurs after 8 ms. If a timeout occurs, the UCIe Module enters TRAINERROR state.
6. If detection is successful on more than one sideband data/clock combination, the device can pick a combination based on a priority order. Pseudocode for sideband assignment:

```

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

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

```

7. If the sideband on the UCIe Module is enabled to send and receive sideband messages ([Step 4](#)), the UCIe Module must start and continue to send {SBINIT Out of Reset} sideband message on both **TXDATASB** and **TXDATASBRD** while sending both **TXCKSB** and **TXCKSBRD** until it detects the same message in its sideband Receivers or a timeout occurs at 8 ms.
8. If {SBINIT Out of Reset} sideband message detection is successful on its sideband Receivers, the UCIe Module stops sending the sideband message. Before sending any further sideband messages, both UCIe Module and UCIe Module Partner must apply Sideband Data/Clock assignment (called the functional sideband) based on the information included in the {SBINIT Out of Reset} sideband message.
9. Any further sideband messages must be sent and received on the functional sideband. Any sideband message exchange can now be performed.
10. The UCIe Module sends the {SBINIT done req} sideband message and waits for a response. If this message is received successfully, UCIe Module Partner responds with {SBINIT done resp} sideband message. When a UCIe Module has sent and received the {SBINIT done resp} sideband message, the UCIe Module must exit to MBINIT. The following additional rules apply when the transmitting/receiving chiplet supports Management Transport protocol. These additional rules are required to initiate mainband link training for the scenario in which the Management Transport path has already been established prior (i.e., SB_MGMT_UP flag is set to 1) and one of the mainband link training triggers (see [Section 4.5](#)) occurred with the Link Training State Machine in RESET state.

- If the Module partner that is receiving the {SBINIT done req} sideband message is in RESET state and is ready to proceed with mainband initialization, the module partner must transition to SBINIT state, respond with an {SBINIT done resp} sideband message, and then send a {SBINIT done req} sideband message.
- Module partner must ignore a received {SBINIT done req} sideband message if the module partner is EITHER [in RESET state but not yet ready to proceed with mainband initialization] OR [in a state machine state other than RESET/SBINIT].
- UCIe Module that is transmitting the {SBINIT done req} sideband message must transition to RESET state (by way of TRAINERROR state) if the UCIe Module did not receive a response within the regular 8-ms time window. In that scenario, the UCIe Module can choose to re-issue the {SBINIT done req} sideband message after transitioning to SBINIT state again from RESET state. The UCIe Module can repeat this process N number of times before waiting in RESET for a new training trigger. (The value of N is implementation-dependent.) The UCIe Module partner must collapse multiple outstanding {SBINIT done req} sideband messages and respond only with a single {SBINIT done resp} sideband message.

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

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

1. The UCIe Module must start and continue to send iterations of 64 UI clock pattern (a clock pattern is defined as starting with 1b and toggling every UI of transmission, i.e., 1010...) and 32 UI low on its sideband Transmitter (**TXDATASB**). The UCIe Module must send strobe on its sideband clock (**TXCKSB**) during the 64-UI clock pattern duration and gated (held low) otherwise.
2. The UCIe Module Partner must sample incoming data pattern with incoming clock.
3. Sideband pattern detection is considered successful if 128 UI clock pattern is detected.
4. If the UCIe Module successfully detects the pattern, it stops sending data and clock on its sideband Transmitters after four more iterations of pattern in [Step 1](#). This will allow for any time differences in both UCIe Modules coming out of RESET. The UCIe Module sideband Transmitter and Receiver must now be enabled to send and receive sideband messages, respectively.
5. If pattern is not detected on its sideband Receiver, the UCIe Module continues to alternate between sending the pattern on its Transmitters for 1 ms, and holding low for 1 ms, for a total of 8 ms. The sideband Receiver must be enabled during this time. Timeout occurs after 8 ms. If a timeout occurs, the UCIe Module must exit to TRAINERROR. If a pattern is detected successfully at any time, as described in [Step 3](#), the UCIe Module enables sideband message transmission as described in [Step 4](#) and continues to [Step 6](#).
6. Once sideband detection is successful ([Step 5](#)), the UCIe Module must start and continue to send {SBINIT Out of Reset} sideband message on **TXDATASB** while sending **TXCKSB** until it detects the same message in its sideband Receivers or a timeout occurs.
7. If {SBINIT Out of Reset} sideband message detection is successful, the UCIe Module must stop sending the message. Any sideband message exchange can now be performed.
8. The UCIe Module must send the {SBINIT done req} sideband message. If this message is received successfully, the UCIe Module Partner responds with the {SBINIT done resp} sideband message. When the UCIe Module has sent and received the {SBINIT done resp} sideband message, the UCIe Module must exit to MBINIT. The following additional rules apply when the transmitting/receiving chiplet supports Management Transport protocol. These additional rules are required to initiate mainband link training for the scenario in which the Management Transport path has already been established prior (i.e., SB_MGMT_UP flag is set to 1) and one of the mainband link training triggers (see [Section 4.5](#)) occurred with the Link Training State Machine in RESET state.

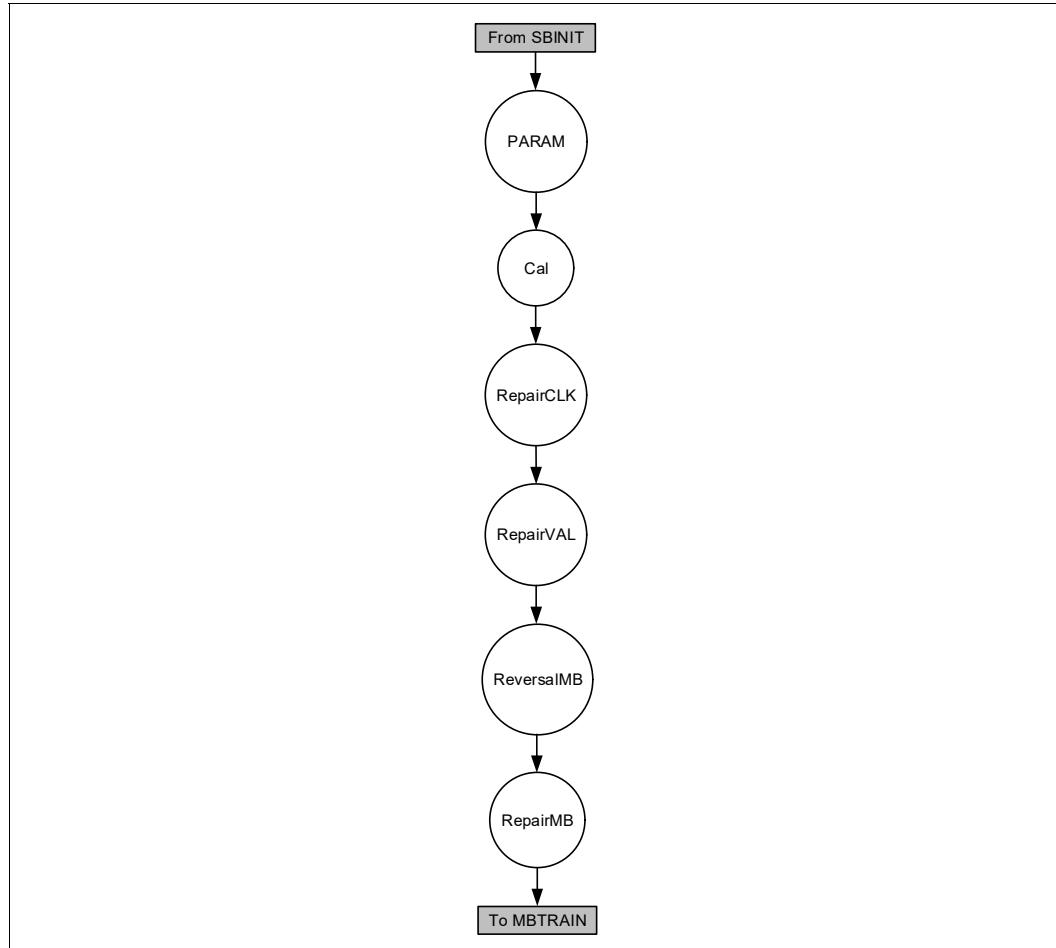
- If the Module partner that is receiving the {SBINIT done req} sideband message is in RESET state and is ready to proceed with mainband initialization, the module partner must transition to SBINIT state, respond with an {SBINIT done resp} sideband message, and then send a {SBINIT done req} sideband message.
- Module partner must ignore a received {SBINIT done req} sideband message if the module partner is EITHER [in RESET state but not yet ready to proceed with mainband initialization] OR [in a state machine state other than RESET/SBINIT].
- UCIe Module that is transmitting the {SBINIT done req} sideband message must transition to RESET state (by way of TRAINERROR state) if the UCIe Module did not receive a response within the regular 8-ms time window. In that scenario, the UCIe Module can choose to re-issue the {SBINIT done req} sideband message after transitioning to SBINIT state again from RESET state. The UCIe Module can repeat this process N number of times before waiting in RESET for a new training trigger. (The value of N is implementation-dependent.) The UCIe Module partner must collapse multiple outstanding {SBINIT done req} sideband messages and respond only with a single {SBINIT done resp} sideband message.

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

4.5.3.3 MBINIT

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

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

Figure 4-34. MBINIT: Mainband Initialization and Repair Flow

4.5.3.3.1 MBINIT.PARAM

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

- “Voltage swing”: The five bit value indicates the Transmitter voltage swing to the UCIe Module Partner. The UCIe Module Partner must use this value and its Receiver termination information to set the reference voltage (Vref) for its Receivers. The corresponding bits in the {MBINIT.PARAM configuration resp} sideband message are reserved.
- “Maximum Data Rate”: The four bit value indicates the Maximum supported Data rate to the UCIe Module Partner. This value must take into consideration all the required features at the data rate (BER, CRC/Retry, quadrature clock phase support etc.). The UCIe Module Partner must compare this value with its supported maximum data rate and must respond with the maximum common data rate encoding in the {MBINIT.PARAM configuration resp} sideband message. For example, a UCIe Module is 8 GT/s capable while the UCIe Module Partner advertises 16 GT/s, the UCIe Module must pick 8 GT/s and send it back in response.
- “Clock Mode”: The one bit value indicates the UCIe Module’s request to the UCIe Module Partner for a strobe or continuous clock. The UCIe Module Partner must use this information to set up the

clock mode on its clock Transmitter. The {MBINIT.PARAM configuration resp} sideband message must reflect the same value. Continuous clock mode requires the clock to be free running and is enforced after receiving the {MBTRAIN.RXCLKCAL start req} sideband message from the UCIe Module Partner. The clock remains free running through the remainder of MBTRAIN (unless MBTRAIN.LINKSPEED is exited due to errors) and in ACTIVE.

- “Clock Phase”: The one bit value indicates the UCIe Module’s request to UCIe Module Partner for the Clock Phase support on UCIe Module’s forwarded clock. This should only be set 1b if the maximum data rate advertised is permitted to do so (see [Table 5-8](#)). The corresponding bit in the {MBINIT.PARAM configuration resp} sideband message must be set to 1b if this was requested and the operational data rate allows it.
- “Module ID”: The UCIe Module sends its “Module ID”. This can be used by the UCIe Module Partner if in a multi-module configuration for Byte mapping, Module enable/disable information etc. The corresponding bits in the {MBINIT.PARAM configuration resp} sideband message are reserved.
- “UCIe-A x32”: This bit is set to 1b when APMW bit in DVSEC UCIe Link Capability register (see [Section 9.5.1.4](#)) is set to 1b (OR) ‘Force x32 width mode in x64 Module’ in the PHY Control register (see [Section 9.5.3.23](#)) is set; otherwise, the bit is set to 0b. If a x64 Advanced Package module supports width reduction to interoperate with a x32 Advanced Package Module, it uses this information from its link partner to condition the results during MBINIT.REVERSALMB. The corresponding bits in the {MBINIT.PARAM configuration resp} sideband message are reserved.
- “UCIe-S x8”: This bit is set to 1 in message {MBINIT.PARAM configuration req} when bit 20, SPMW, in the DVSEC UCIe Link Capability register (see [Section 9.5.1.4](#)) is set to 1 (OR) ‘Force x8 Width’ bit is set to 1 in the PHY Control register (see [Section 9.5.3.23](#)). Otherwise, this bit is set to 0. See [Section 4.5.3.3.6](#) for how this bit is used. The corresponding bit in the {MBINIT.PARAM configuration resp} sideband message is reserved.
- “Sideband Feature Extensions”: This bit is set to 1 if the transmitter supports sideband feature extensions (see [Section 4.5.3.3.1.1](#)).

Following is the sequence for parameter exchange:

1. The UCIe Module sends the {MBINIT.PARAM configuration req} sideband message to exchange parameters with the UCIe Module Partner.
2. After the {MBINIT.PARAM configuration req} sideband message is received, the UCIe Module Partner resolves and responds with the {MBINIT.PARAM configuration resp} sideband message.
3. After the UCIe Module has sent and received the {MBINIT.PARAM configuration resp} sideband message, the UCIe Module must exit to MBINIT.CAL.

It is strongly recommended that if interoperable parameters are not negotiated, then hardware maps this scenario to an Internal Error in the Error Log 1 register and transition the LTS to TRAINERROR, RDI to LINKERROR, and assert **p1_trainerror** on RDI. For a multi-module Link, all the parameters except “Module ID” must be the same for all the modules, and if this is not the case, it is strongly recommended that hardware maps this scenario to the same error escalation path.

When management transport is supported, the additional conditions required for the Link training state machine to exit MBINIT.PARAM state to MBINIT.CAL are:

- Management Transport was not negotiated.
- (OR) Management Transport was negotiated and it was either initialized successfully or an error was detected during initialization.
- (OR) SB_MGMT_UP flag is already set on entry into MBINIT state

AND

There is no MBINIT Stall condition (see [Section 4.5.3.3.1.2](#))

AND

Port is ready to train mainband.

4.5.3.3.1.1 Management Transport Path Negotiation

Management Transport protocol over the sideband is optional and chiplets use the mechanism described in this section to negotiate support for it. A sample negotiation flow is shown in [Figure 4-35](#) for a single module design. Sideband Feature Extensions Supported (SFES) is bit 14 in the {MBINIT.PARAM configuration req} sideband message (see [Table 7-11](#)). Note that MBINIT.PARAM state handshake relating to management path negotiation described in this section, is performed on all transitions through the MBINIT state. If SB_MGMT_UP flag is set (see [Section 8.2.3.1.2](#) for when this happens) at entry into MBINIT state, management transport traffic continues without interruption in the MBINIT.PARAM state.

Figure 4-35. Example Sideband Management Transport Protocol Negotiation – Single-module Scenario

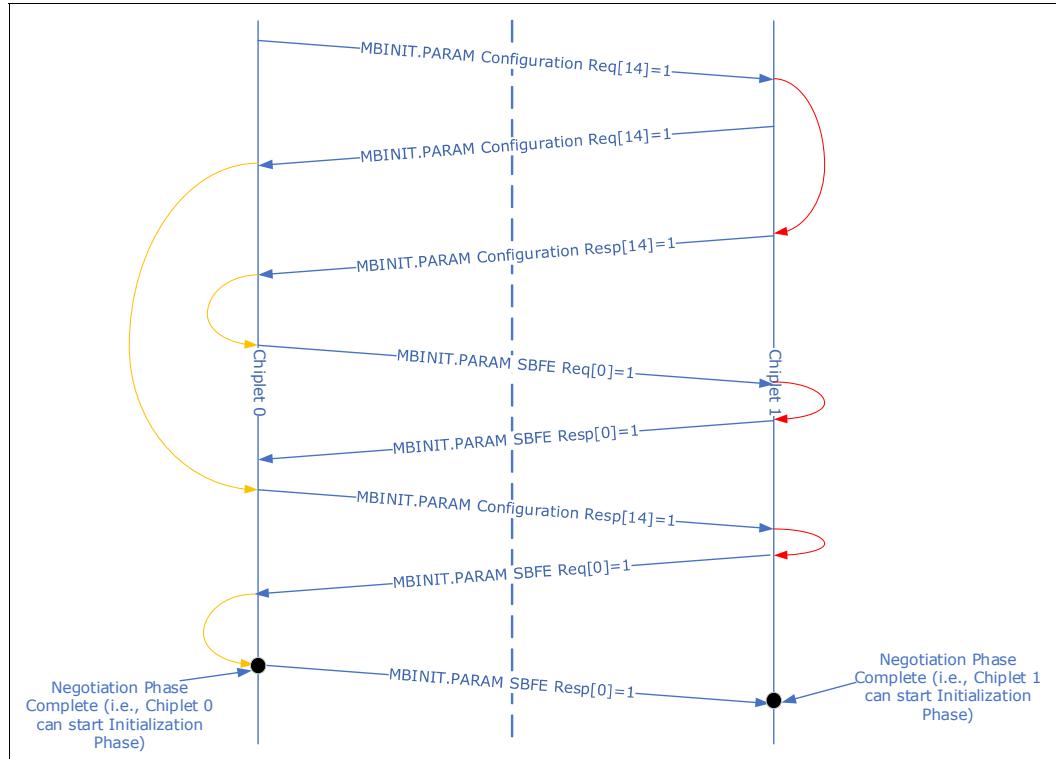
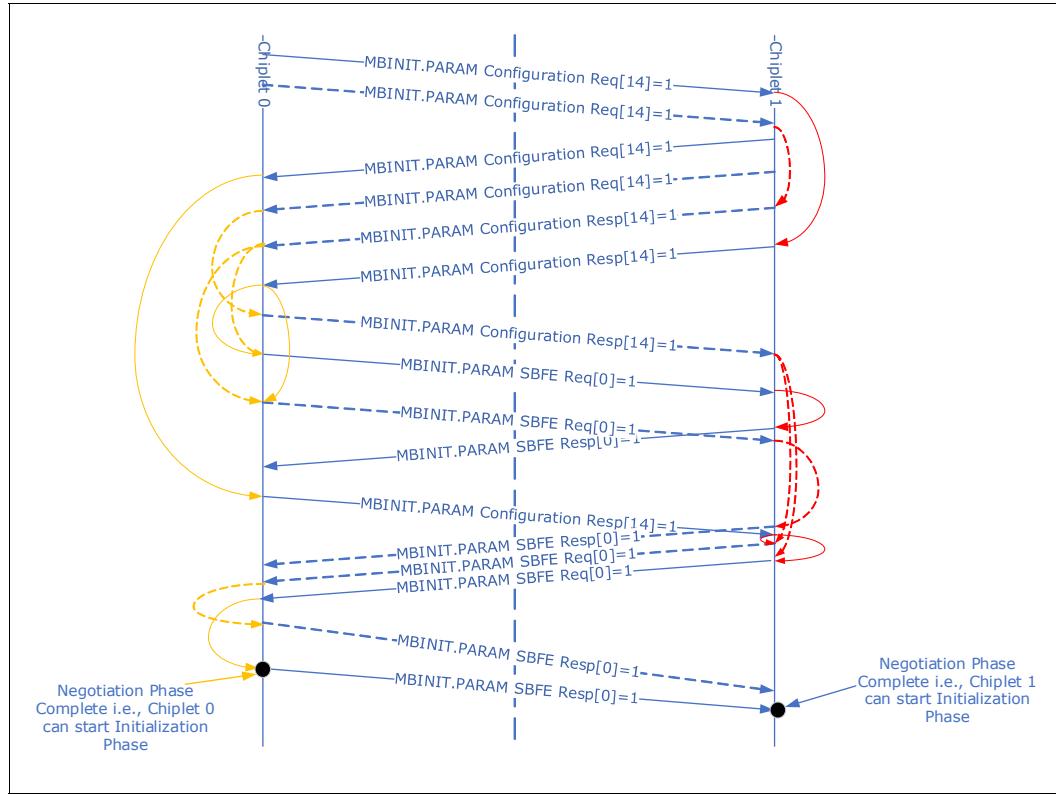


Figure 4-36. Example Sideband Management Transport Protocol Negotiation – Two-module Scenario^a



a. Solid lines are for Module 0. Dashed lines are for Module 1.

- Unless otherwise specified, a chiplet can optionally check for violation of any Negotiation Phase rules (discussed in the subsequent bullets), and when a violation is detected, the chiplet initiates a TRAINERROR handshake (see [Section 4.5.3.8](#)) to return the LTSI to RESET state.
- Sideband Feature Extensions Supported (SFES) bit in the {MBINIT.PARAM configuration req} sideband message (see [Table 7-11](#), it is bit [14] in the message) is defined to indicate support for extended sideband features (of which Management Transport is one), during MBINIT.PARAM state of link training.
 - 0 => Sideband Feature Extensions are not supported, 1 => Sideband Feature Extensions are supported.
 - All modules in a multi-module design must have the same value for this bit in the Req message.
 - After the SB_MGMT_UP flag is set, the value of this bit must remain the same on all subsequent transitions through the MBINIT.PARAM state, until that flag is cleared.
- If the Remote link partner supports sideband feature extensions and it received the SFES bit set to 1 in the {MBINIT.PARAM configuration req} sideband message, the Remote link partner will set SFES bit in the {MBINIT.PARAM configuration resp} sideband message that it sends out; otherwise, the bit is cleared to 0 in the resp message.
 - All modules in a multi-module design must have the same value for this bit in the resp message.

- If the SFES bit in the {MBINIT.PARAM configuration resp} sideband message is received as set to 1 across all modules, then the chiplet negotiates the next level of details of extended sideband features supported with remote link partner. If the SFES bit in the {MBINIT.PARAM configuration resp} sideband message is received as cleared to 0 in any module, then MBINIT.PARAM state is exited to MBINIT.CAL.
 - {MBINIT.PARAM SBFE req} sideband message (see [Table 7-11](#)) is sent to the remote link partner on all modules which then sends back an {MBINIT.PARAM SBFE resp} sideband message on all modules. This handshake happens independently in each direction.
 - SBFE Req[0]/Resp[0] (also referred to as the MTP bit) indicates support for transmission/reception of Management Port Messages. Remote link partner must set the MTP bit in the {MBINIT.PARAM SBFE resp} sideband message if it was set in the {MBINIT.PARAM SBFE req} sideband message, and it supports receiving Management Encapsulation messages.
 - After the SB_MGMT_UP flag is set to 1, the value of this bit must remain the same on all subsequent transitions through the MBINIT.PARAM state, until that flag is cleared to 0.
- When negotiating SFES in {MBINIT.PARAM configuration req/resp}, if a chiplet advertised SFES support in the Req message, the chiplet must also advertise that support in the Resp message provided the associated Req message had that capability advertised. If the chiplet did not advertise SFES support in the Req message, then the chiplet must not advertise that support in the Resp message.
- For a multi-module UCIe Link, the negotiation is performed independently per module.
 - A Physical Layer implementation may advertise MTP bit in the SBFE Req message only on a subset of the modules.

Note: A message sent on a given Module ID could be received on a different Module ID on the partner sideband Receiver. Hence all sideband links in a multi-module design must be capable of receiving MPMs even if they are limited to only supporting transmit of these messages on a subset of sideband links. See [Figure 4-37](#) and [Figure 4-38](#) for examples of multi-module transmit/receive scenarios that illustrate this point.

- After the {MBINIT.PARAM SBFE resp} sideband message has been transmitted to the remote link partner and {MBINIT.PARAM SBFE resp} sideband message has been received from the remote link partner are complete (successfully or unsuccessfully) during MBINIT.PARAM across all modules, the PHY informs the Management Port Gateway of the following:
 - Negotiated link count with management transport support on the transmit side, using the **pm_param_local_count[N-1:0]** signals (see [Section 10.1](#) for more details) at the end of the negotiation phase. This is the value RxQ-Local. A link is considered to have negotiated management transport support on the transmit side if the link transmitted the {MBINIT.PARAM SBFE req} sideband message with the MTP bit set to 1 and received the corresponding {MBINIT.PARAM SBFE resp} sideband message with its MTP bit also set to 1.
 - Negotiated link count with management transport support on the receive side, using the **pm_param_remote_count[N-1:0]** signals (see [Section 10.1](#) for more details). This is the value RxQ-Remote. A link is considered to have negotiated management transport support on the receive side if the link received the {MBINIT.PARAM SBFE req} sideband message with the MTP bit set to 1 and transmitted the corresponding {MBINIT.PARAM SBFE resp} sideband message with its MTP bit also set to 1.
- A module must be able to receive initialization phase-related messages (see [Section 8.2.3.1.2](#)) once it has transmitted {MBINIT.PARAM SBFE resp}.
- Negotiation phase ends when {MBINIT.PARAM SBFE resp} has been sent and received across all modules.

- While in SBINIT state, if the SB_MGMT_UP flag transitioned from 1 to 0, the chiplet must move the LTSM to TRAINERROR state -> RESET state.
- While in MBINIT state, if the SB_MGMT_UP flag transitioned from 1 to 0, the chiplet must perform a TRAINERROR handshake and move the LTSM to TRAINERROR state -> RESET state.

Figure 4-37. Example Sideband MPM Logical Flow with Two Modules and No Module Reversal

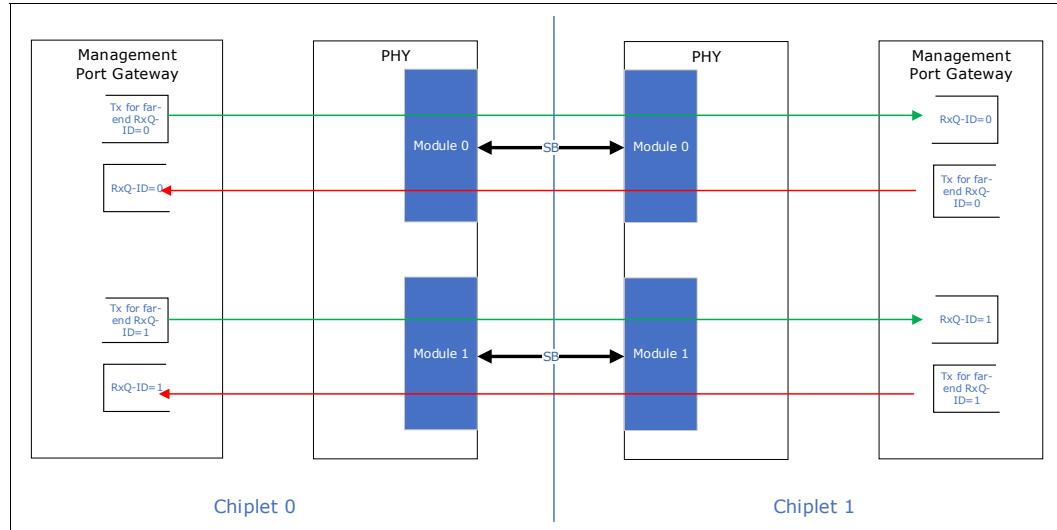
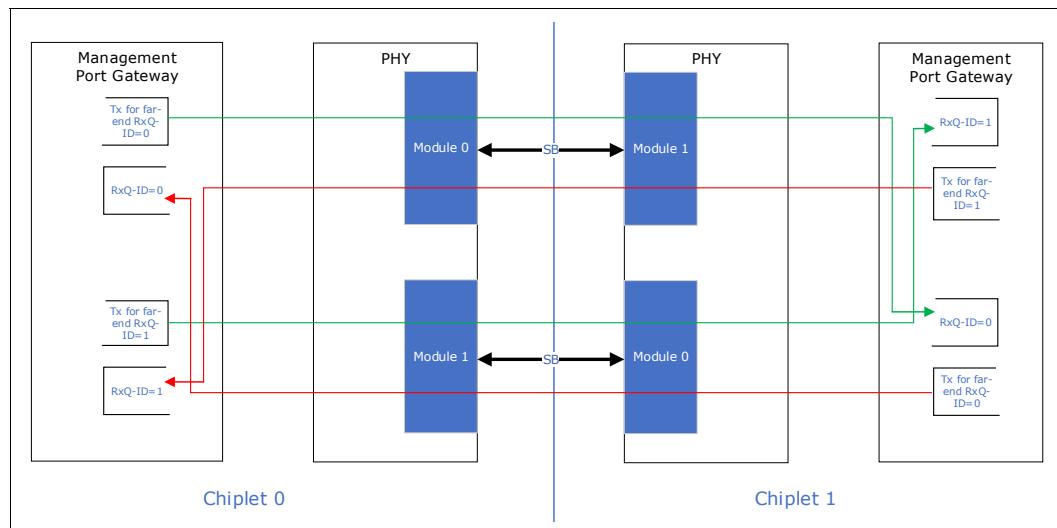


Figure 4-38. Example Sideband MPM Logical Flow with Two Modules and Module Reversal



4.5.3.3.1.2 MBINIT Stall Capability

To support firmware download and other functionality that might have to be configured before the mainband link can start training, a capability is provided to “pause” the MBINIT state machine after the PARAM sub-state has completed.

This is an optional capability that is enabled only when both chiplets have indicated support for Sideband Feature Extensions as described in Section 4.5.3.3.1.1.

To “pause” link training after MBINIT.PARAM, either side can send a “Stall” encoding of FFFFh in the MsgInfo field of the {MBINIT.PARAM SBFE resp} sideband message. For example, if a chiplet needs to download firmware by way of the partner port before the chiplet can bring up the mainband, the partner port can respond with “stall” encoding as stated above. Stall encoding instructs the other side to pause and not move beyond the MBINIT.PARAM state. The payload in the {MBINIT.PARAM SBFE resp} sideband message with stall encoding is still valid and must be accurate to the responder’s capabilities. An {MBINIT.PARAM SBFE resp} sideband message with “Stall” encoded must be sent once every 4 ms until the sender determines that it no longer needs to stall, at which time the sender either sends the {MBINIT.PARAM SBFE resp} message without the stall encoding (in which case the state machine advances to MBINIT.CAL state if other conditions allow (see [Section 8.2.3.1.2](#))), OR the sender does not send an {MBINIT.PARAM SBFE resp} sideband message, the link times out, and the link transitions from TRAINERROR state to RESET state and starts over again.

It is legal for one side to indicate a stall and the other side to not indicate a stall. In that case, both sides are stalled. It is also valid for either side to explicitly request an entry to TRAINERROR state while in MBINIT.PARAM state. This can occur if either side is not yet ready to train the mainband.

See [Section 4.5.3.3.1.3](#) for details of what happens at the end of MBINIT.PARAM when either end is a sideband-only port.

Support for receiving an {MBINIT.PARAM SBFE resp} sideband message with “stall” encoding is required in all ports that advertise the SFES bit set to 1 in the {MBINIT.PARAM configuration req} sideband message. Support for transmitting the {MBINIT.PARAM SBFE resp} sideband message with “stall” encoding is implementation-dependent. For example, if a design needs to support the firmware download feature, the design can support this capability if the design cannot complete firmware download within 8 ms.

[Figure 4-39](#) shows a scenario in which the link training state machine initially moves through RESET -> SBINIT -> MBINIT.PARAM with “stall” -> TRAINERROR -> RESET. During this phase, a chiplet “stalls” in MBINIT.PARAM for additional Init time, such as for downloading chiplet firmware. When the chiplet Init is complete, the chiplet either initiates entry to TRAINERROR state with a TRAINERROR handshake message, OR the chiplet can stop sending the {MBINIT.PARAM SBFE resp} sideband message with “stall” encoding, which would eventually trigger entry to TRAINERROR state initiated by the partner chiplet. In the second phase, the state machine moves through to SBINIT -> MBINIT.PARAM without “stall” -> MBINIT.CAL, thus training the mainband. This flow is useful for scenarios in which a chiplet potentially needs to change the advertised parameters for Link training after chiplet Init. Note that the transition to TRAINERROR in this case does not escalate to RDI transitioning to LinkError (or `pl_trainerror` assertion on RDI).

Figure 4-39. MBINIT “Stall” Example 1

Sideband Management path setup occurs first, followed by mainband training but with an entry to RESET in between.

“Stall” in MBINIT.PARAM is used to gain additional time for chiplet initialization during the first entry to MBINIT.PARAM.

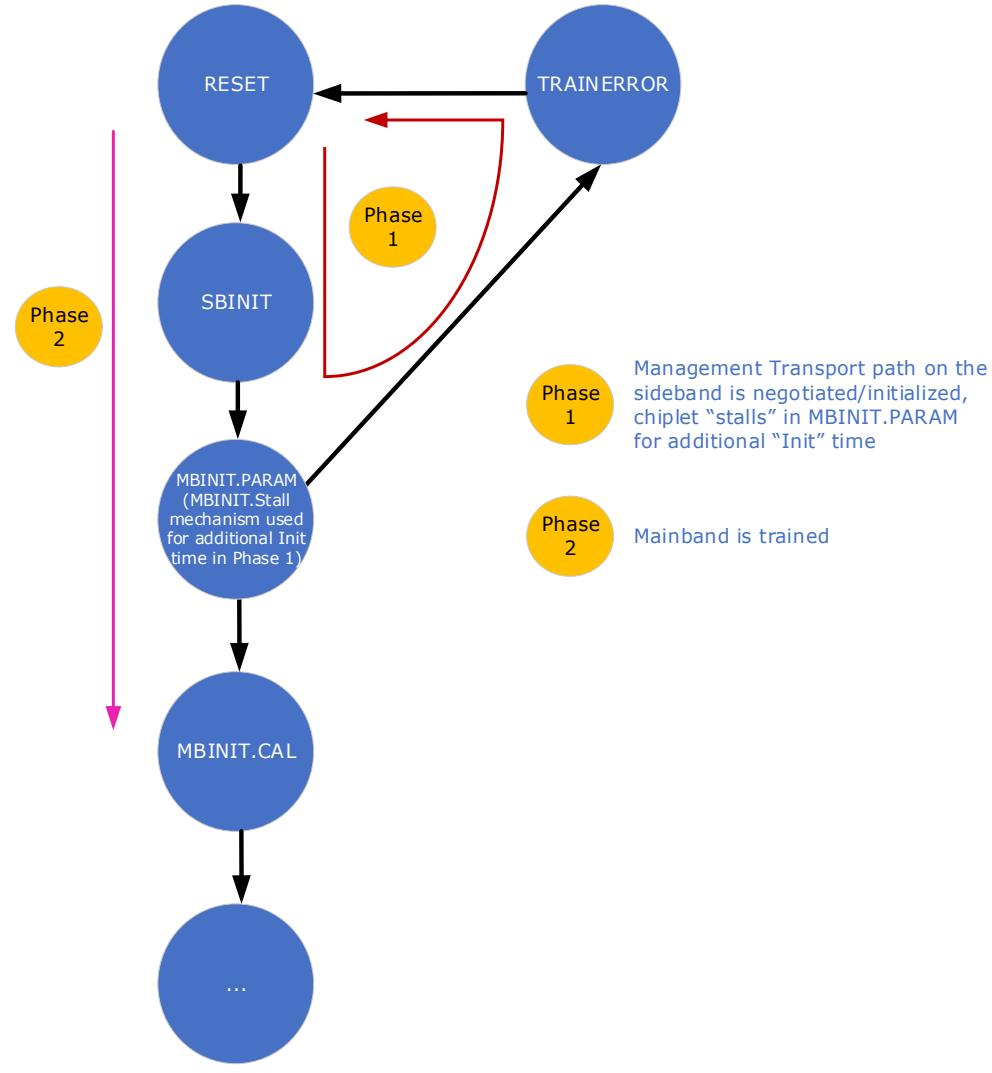
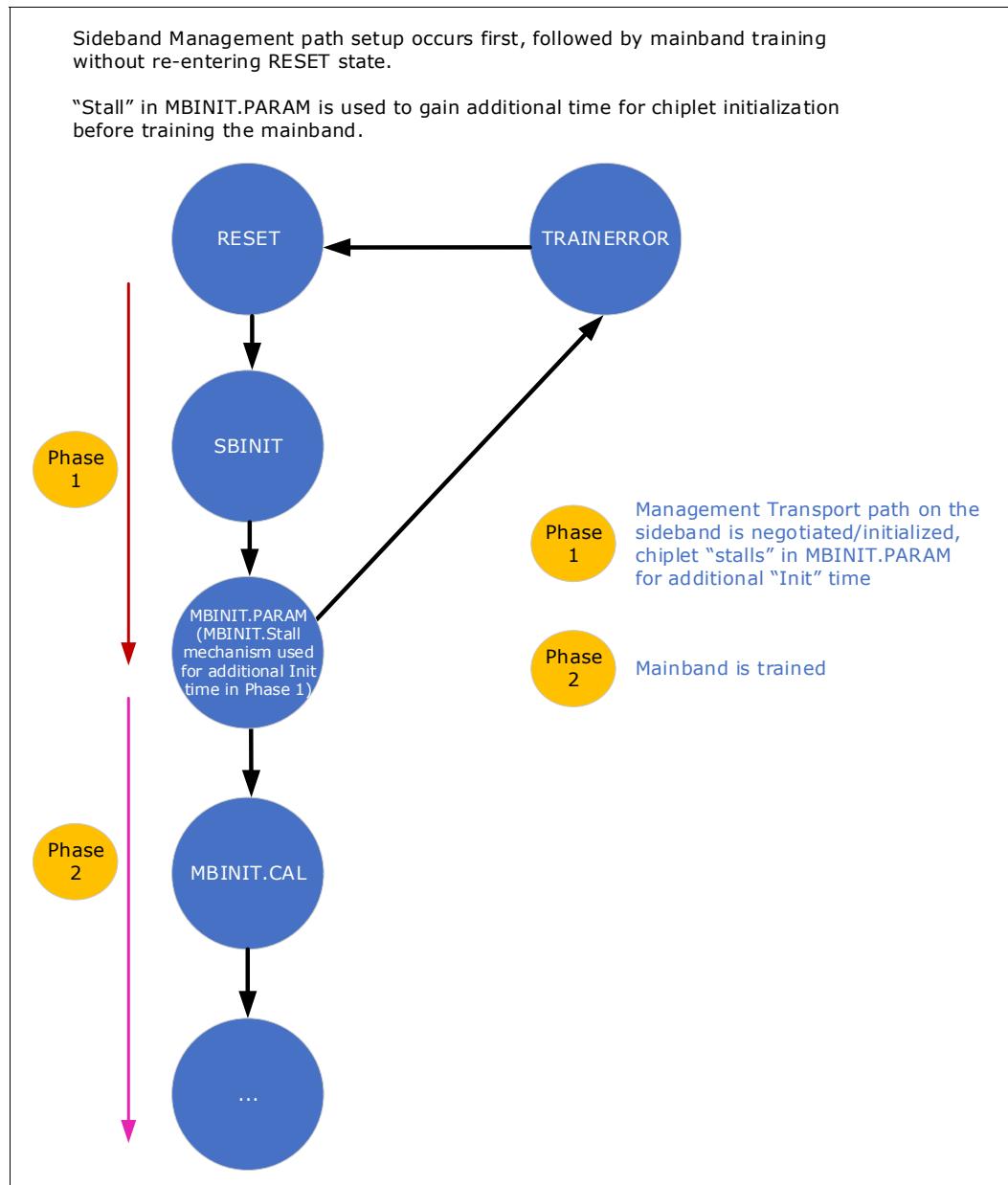


Figure 4-40 shows a scenario in which the link training state machine initially moves through RESET -> SBINIT -> MBINIT.PARAM with “stall”. The chiplet “stalls” in MBINIT.PARAM for additional Init time. After the chiplet Init is complete, the chiplet sends an {MBINIT.PARAM SBFE resp} sideband message without a “stall” encoding that triggers state machine entry to MBINIT.CAL. Mainband training resumes from that point.

Figure 4-40. MBINIT “Stall” Example 2



4.5.3.3.1.3 UCIe-S Sideband Only (SO) Port

A UCIe-S Sideband-only port will advertise the SFES bit in an {MBINIT.PARAM configuration req} sideband message. If that negotiation is successful, the port advertises bit 2 (SO bit) in an {MBINIT.PARAM SBFE req} sideband message to indicate that the port is a "Sideband-only port". If the port receives an {MBINIT.PARAM SBFE resp} sideband message with the SO bit set to 1, training pauses on both sides at the exit of the MBINIT.PARAM phase until the next Management Reset or a transition to the TRAINERROR state is triggered (see [Section 4.5.3.8](#)). State residency timeout is disabled in MBINIT.PARAM. If the remote link partner did not set the SO bit in the {MBINIT.PARAM SBFE resp} sideband message, the link goes to the TRAINERROR state. This is an SiP integration error and is fatal. All links that support management transport over sideband (i.e., those links that set bit 0 as 1 in the {MBINIT.PARAM SBFE req} sideband message that they transmit) must set the SO bit to 1 in the {MBINIT.PARAM SBFE resp} sideband message that they transmit, provided that the corresponding bit was set to 1 in the {MBINIT.PARAM SBFE req} sideband message that they received.

In multi-sideband-only Link configuration (this refers to a multi-module configuration with sideband-only modules; see [Section 8.2.1](#)), the chiplet must advertise the same value for the SO bit in the {MBINIT.PARAM SBFE req} or {MBINIT.PARAM SBFE resp} sideband message across all links. Receivers can optionally check for violation of this rule and then trigger a transition to the TRAINERROR state if the transmitter violates this rule.

4.5.3.2 MBINIT.CAL

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

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

4.5.3.3 MBINIT.REPAIRCLK

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

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

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

1. The UCIe Module sends the {MBINIT.REPAIRCLK init req} sideband message and waits for a response. The UCIe Module Partner when ready to receive pattern on `RCKP_L`, `RCKN_L`, `RTRK_L`, and `RRDCK_L` responds with {MBINIT.REPAIRCLK init resp}.
2. The UCIe Module must now send 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on its `TCKP_L` only (`TCKN_L`, `TTRK_L`, and `TRDCK_L` are tri-stated). Clock repair pattern must not be scrambled.

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

15. The UCIe Module Partner responds with {MBINIT.REPAIRCLK result resp} sideband message with logged result of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. If detection is successful only on **RRDCK_L** and not on **RCKP_L**, **RTRK_L**, **RCKN_L**, **TRDCK_P/RRDCK_P** is available as a repair resource. Else if detection is unsuccessful on any of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, the physical Lane **TRDCK_P/RRDCK_P** is not available as a repair resource. Else an electrical short is implied.
16. After receiving the {MBINIT.REPAIRCLK result resp} sideband message, the UCIe Module sends the sideband message {MBINIT.REPAIRCLK init req} and waits for a response. The UCIe Module Partner when ready to receive pattern on **RCKP_L**, **RCKN_L**, **RTRK_L**, **RRDCK_L** responds with {MBINIT.REPAIRCLK init resp}.
17. After receiving the {MBINIT.REPAIRCLK init resp} sideband message, the UCIe Module sends 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on **TTRK_L** only. (**TCKP_L**, **TCKN_L**, and **TRDCK_L** are tri-stated).
18. The UCIe Module Partner detects this pattern on all **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module Partner logs the detection result for **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**.
19. The UCIe Module after completing pattern transmission sends {MBINIT.REPAIRCLK result req} sideband message to get the logged result and waits for a response.
20. The UCIe Module Partner stops comparison and responds with {MBINIT.REPAIRCLK result resp} sideband message with logged result of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**. If detection is successful only on **RTRK_L** and not on **RCKP_L**, **RCKN_L**, **RRDCK_L**, no repair is needed. Else if detection is unsuccessful on any of **RCKP_L**, **RCKN_L**, **RTRK_L**, and **RRDCK_L**, repair is needed on the physical Lane **TTRK_P/RTRK_P**. Else an electrical short is implied.
21. Clock or Track is unrepairable if any of the following are true:
 - If repair is needed on any two of **RCKP_L**, **RCKN_L**, and **RTRK_L**
 - Electrical short is detected
 - **RRDCK_L** is unavailable for repair when repair is needed
 If the clock or track is unrepairable, the UCIe Module and UCIe Module Partner must exit to TRAINERROR after performing TRAINERROR handshake (see [Section 4.5.3.8](#)).
- If repair is required on only one of the clock or track lanes and a repair resource is available, then the UCIe Module applies repair on its clock/track Transmitter and sends the {MBINIT.REPAIRCLK apply repair req} sideband message with repair information. If a repair is needed for one of the clock or track pins (**CKP**, **CKN**, or **TRK**) and a repair resource is available, repair is applied as described in [Section 4.3](#). The UCIe Module Partner applies repair and sends {MBINIT.REPAIRCLK apply repair resp} sideband message.
22. If a repair is applied, UCIe Module must check the repair success by applying clock repair pattern and checking on the Receiver.
 - a. The UCIe Module sends sideband message {MBINIT.REPAIRCLK check repair init req} to initiate check repair and waits for a response. The UCIe Module Partner responds with sideband message {MBINIT.REPAIRCLK check repair init resp} and is ready to receive and check clock repair pattern.
 - b. After receiving the {MBINIT.REPAIRCLK check repair init resp} sideband message, the UCIe Module sends 128 iterations of clock repair pattern (16 clock cycles followed by 8 cycles of low) on **TCKP_L**. The UCIe Module Partner detects this pattern **RCKN_L**, **RCKP_L**, **RTRK_L**. Detection is considered successful if at least 16 consecutive cycles of clock repair pattern are detected. The UCIe Module requests for check result request using the sideband message

{MBINIT.REPAIRCLK check results req} and the UCIe Module Partner responds with the sideband message {MBINIT.REPAIRCLK check results resp}. Repair is considered successful if pattern is detected only on **RCKP_L**. If repair is unsuccessful, the UCIe Module and UCIe Module Partner must exit to TRAINERROR after performing TRAINERROR handshake (see [Section 4.5.3.8](#)).

- c. Step a and Step b are repeated for **TCKN_L** and **TTRK_L**.
- 23. If repair is successful or repair is not required, the UCIe Module sends {MBINIT.REPAIRCLK done req} sideband message and the UCIe Module Partner responds with {MBINIT.REPAIRCLK done resp} sideband message. When the UCIe Module has sent and received {MBINIT.REPAIRCLK done resp}, it must exit to REPAIRVAL.

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

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

4.5.3.3.4 MBINIT.REPAIRVAL

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

- Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- Clock Receivers are enabled
- When not transmitting the VALTRAIN pattern, the transmitters for **TVLD_L** and **TRDVLD_L** are disabled (tri-stated)
- The receivers for **RVLD_L** and **RRDVLD_L** are enabled

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

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

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

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

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

4.5.3.3.5 MBINIT.REVERSALMB

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

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

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

Table 4-7. Per Lane ID Pattern

Pattern	0	1	0	1	Lane ID ^a												0	1	0	1
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				

a. Note that bit 0 of Lane ID maps to bit 4 in the Per Lane ID pattern, bit 1 to bit 5 and so on until bit 7 to bit 11.

Table 4-8. Per Lane ID Pattern Examples

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

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

1. The UCIe Module must send the {MBINIT.REVERSALMB init req} sideband message. When ready to receive "Per Lane ID" pattern and perform per-Lane pattern comparison, the UCIe Module Partner must respond with {MBINIT.REVERSALMB init resp}.
2. On Receiving the {MBINIT.REVERSALMB init resp} sideband message or entering from [Step 8](#), the UCIe Module must send {MBINIT.REVERSALMB clear error req} sideband message. Upon receiving this message, the UCIe Module Partner clears any prior errors and responds with {MBINIT.REVERSALMB clear error resp}. After receiving {MBINIT.REVERSALMB clear error resp}, the UCIe Module sends 128 iterations of Per Lane ID pattern (see [Table 4-7](#)) on all N data Lanes with correct Valid framing on the Valid Lane (see [Section 5.11](#) and [Section 4.1.2](#)) along with the forwarded clock. [Table 4-7](#) and [Table 4-8](#) show examples of the Per Lane ID pattern. N is 68 (64 Data + 4 RD) for a x64 Advanced Package. N is 34 (32 Data + 2 RD) for a x32 Advanced Package. N is 16 for a x16 Standard Package. N is 8 for a x8 Standard Package.
3. The UCIe Module Partner must perform a per-Lane compare on its Receivers on all N Lanes (see [Section 4.4](#)). Detection on a Lane is considered successful if at least 16 consecutive iterations of "Per Lane ID" pattern are detected. The UCIe Module Partner logs the detection result for its Receiver Lanes to be used for Lane reversal Detection.
4. After sending 128 iterations of "Per Lane ID" pattern, the UCIe Module stops sending the pattern and sends {MBINIT.REVERSALMB result req} sideband message to get the logged result.
5. The UCIe Module Partner stops comparison and must respond with {MBINIT.REVERSALMB result resp} sideband message with per Lane result (see [Table 7-11](#) for the message format).
6. If majority of the Lanes show success (since some Lanes may need repair), Lane reversal is not needed. Skip to [Step 11](#). Note that if exactly 50% of the Lanes showed success, Lane reversal is applied.
7. The UCIe Module applies Lane reversal on its Transmitters (see [Section 4.2](#)).
8. Following the Lane reversal application on its Transmitters, the UCIe Module repeats [Step 2](#) through [Step 5](#).
9. If majority of Lanes show success, the Lane reversal is needed. Lane reversal preserved for rest of the device operation. Skip to [Step 11](#).
10. The UCIe Module must exit to TRAINERROR after completing the TRAINERROR handshake.
11. The UCIe Module must send {MBINIT.REVERSALMB done req} sideband message and the UCIe Module Partner responds with {MBINIT.REVERSALMB done resp}. When the UCIe Module has sent and received {MBINIT.REVERSALMB done resp} sideband message, it must exit to REPAIRMB.

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

If a x32 Advanced Package Module had received "UCIe-A x32" as 0b during parameter exchanges, it must recognize that it is connected to a x64 Advanced Package Module and appropriately interpret the

received {MBINIT.REVERSALMB result resp} sideband message, looking for majority of success in the lower 32 data lane set and 2 repair lane set of the x64 module (the x64 module will always place the results of its receiver on the lower half of its data/repair lane set).

If a x16 Standard Package Module that supports interoperation with a x8 Standard Package Module had its SPMW bit set to 1b OR has transmitted or received "UCIE-S x8" as 1b during parameter exchanges, the x16 Standard Package Module must recognize that it needs to operate in x8 mode and appropriately interpret the received {MBINIT.REVERSALMB result resp} sideband message. In this scenario, the x16 Standard Package Module applies [Step 6](#) through [Step 9](#) to the lower-8 data-lane set. Additionally, the x16 Standard Package Module applies Lane reversal (if required) within the lower-8 data-lane set.

When a x8 Standard Package Module receives the {MBINIT.REVERSALMB result resp} sideband message, the module must look for majority of success in the bits that correspond to the lower-8 data-lane set only.

4.5.3.3.6 MBINIT.REPAIRMB

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

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

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

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

1. The UCIE Module must send the {MBINIT.REPAIRMB start req} sideband message and waits for a response. The UCIE Module Partner responds with {MBINIT.REPAIRMB start resp}.
2. The UCIE Module device performs Transmitter-initiated Data-to-Clock point test as described in [Section 4.5.1.1](#) on its Transmitter Lanes (all the data Lanes including the redundant Lanes). The Receiver must check for the pass/fail criteria on the data Lanes and the redundant Lanes.
 - a. The transmit pattern must be set up to send 128 iterations of continuous mode "Per Lane ID" Pattern. "Per Lane ID" pattern must not be scrambled. The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if at least 16 consecutive iterations of "Per Lane ID" pattern are detected.
 - c. LFSR Reset has no impact in MBINIT.REPAIRMB.
3. The UCIE Module receives the per-Lane pass/fail information over sideband message at the end of Transmitter initiated data to clock point test [Step 2](#).

4. If Lane repair is required and the necessary repair resources are available, the UCIe Module applies repair on its mainband Transmitters for data Lanes as described in [Section 4.3.1](#), and sends {MBINIT.REPAIRMB Apply repair req} sideband message. Upon receiving this sideband message, the UCIe Module Partner applies repair on its mainband Receivers for data Lanes as described in [Section 4.3.1](#), and sends {MBINIT.REPAIRMB Apply repair resp} sideband message. Otherwise, if the number of Lane failures are more than repair capability (see [Section 4.3](#)), the mainband is unrepairable and the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.
5. If repair is not required, perform [Step 7](#).
6. If Lane repair is applied ([Step 4](#)), the applied repair is checked by UCIe Module by repeating [Step 2](#) and [Step 3](#). If post repair Lane errors are logged in [Step 5](#), the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake. If repair is successful, perform [Step 7](#).
7. The UCIe Module sends {MBINIT.REPAIRMB end req} sideband message and the UCIe Module Partner responds with {MBINIT.REPAIRMB end resp}. When UCIe Module has sent and received {MBINIT.REPAIRMB end resp}, it must exit to MBTRAIN.

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

1. The UCIe Module sends the {MBINIT.REPAIRMB start req} sideband message and waits for a response. The UCIe Module Partner responds with the {MBINIT.REPAIRMB start resp} sideband message when ready to receive the pattern on its mainband Receivers for data Lanes.
2. The UCIe Module performs Transmitter-initiated Data-to-Clock point test as described in [Section 4.5](#).
 - a. The transmit pattern must be set up to send 128 iterations of continuous mode "Per Lane ID" Pattern. The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if at least 16 consecutive iterations of "Per Lane ID" pattern are detected.
 - c. LFSR Reset has no impact in MBINIT.REPAIRMB.
3. The UCIe Module must send the {MBINIT.REPAIRMB apply degrade req} sideband message indicating the functional Lanes on its Transmitter using one of the logical Lane map encodings from [Table 4-9](#). Encodings 100b and 101b can be used only when the SPMW bit is set to 1 or the UCIe-S x8 bit was set to 1 in the transmitted or received {MBINIT.PARAM configuration req} sideband message. If the remote Link partner indicated a width degrade in the functional Lanes, the UCIe Module must apply the corresponding width degrade to its Receiver. If the remote Link partner indicated all Lanes are functional (i.e. a Lane map code of 011b), the UCIe Module sets its Transmitter and Receiver to the width corresponding to the functional Lane encoding determined on its Transmitter. The UCIe Module sends the {MBINIT.REPAIRMB apply degrade resp} sideband message after setting its Transmitter and Receiver lanes to the relevant width. If the width on the Transmitter or Receiver has changed, both Link partners must repeat [Step 2](#). If the width on the Transmitter or Receiver has not changed, proceed to [Step 4](#). If a "Degrade not possible" encoding is sent or received in the {MBINIT.REPAIRMB apply degrade req} sideband message, the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.
4. The UCIe Module sends the {MBINIT.REPAIRMB end req} sideband message and the UCIe Module Partner responds with the {MBINIT.REPAIRMB end resp} sideband message. When the UCIe Module has sent and received the {MBINIT.REPAIRMB end resp} sideband message, it must exit to MBTRAIN.

Table 4-9. Standard Package Logical Lane Map

Lane Map Code	Functional Lanes
000b	None (Degrade not possible)
001b	Logical Lanes 0 to 7
010b	Logical Lanes 8 to 15
011b	0 - 15
100b	Logical Lanes 0 to 3
101b	Logical Lanes 4 to 7

IMPLEMENTATION NOTE

Consider an example in which Die A is communicating with Die B over a Standard Package PCIe Link.

During the first iteration of Step 2 of MBINIT.REPAIRMB, let's say that Tx on Die A detects errors on Lane ID 1 and not on any other Lanes, but Tx on Die B detects errors on Lane ID 10 and not on any other Lanes. Thus, as per the rules in Step 3, Die A sends {MBINIT.REPAIRMB apply degrade req} with a Lane map code of 010b. Similarly, in Step 3, Die B sends {MBINIT.REPAIRMB apply degrade req} with a Lane map code of 001b. The Rx on Die B disables Lanes 0 to 7, and the Tx on Die B tri-states Lanes 8 to 15. The Rx on Die A disables Lanes 8 to 15, and the Tx on Die A tri-states Lanes 0 to 7. Following the rules in Step 3, each die goes back and repeats Step 2.

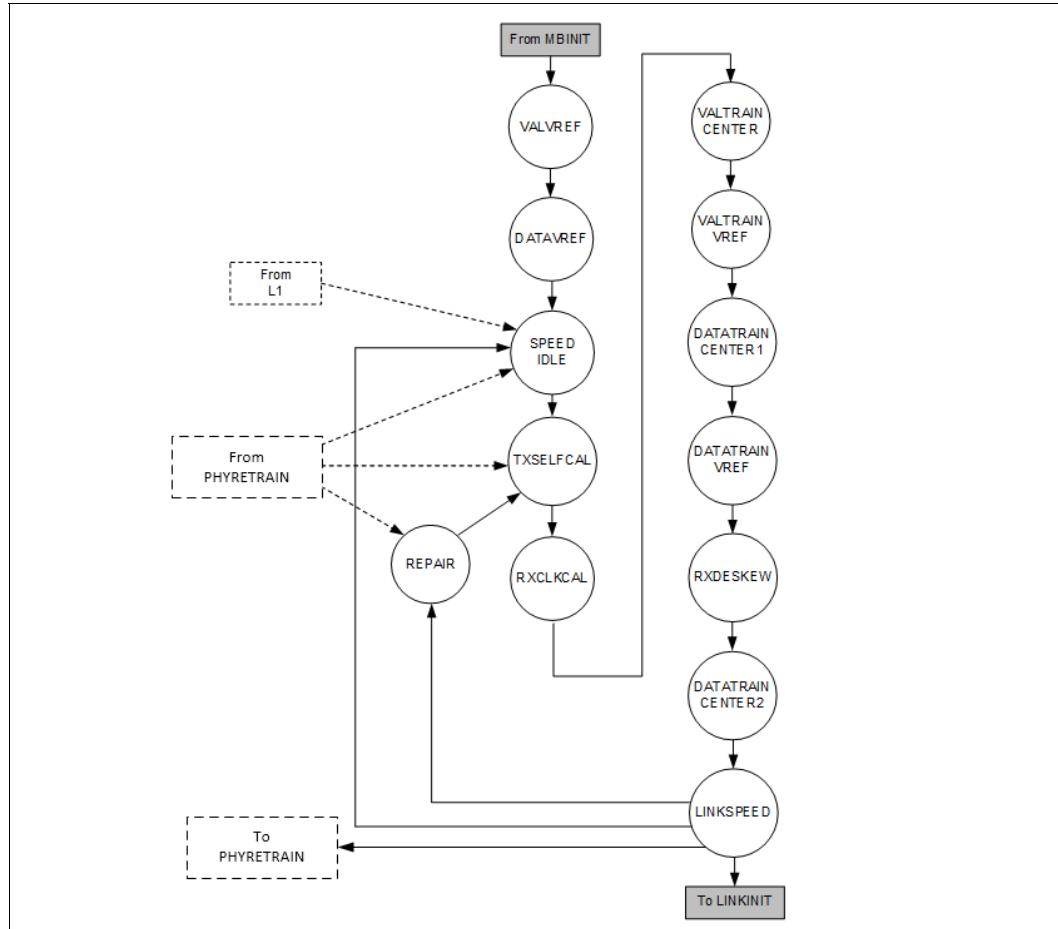
In this second iteration of Step 2, both Die know that some of the Lanes are disabled, and they will ignore the information related to the disabled Lanes in {Tx Init D to C results resp} (e.g., Die A will ignore the information related to Lanes 0 to 7 and perform only the Transmitter-initiated Data to Clock point test on Lanes 8 to 15).

Let's say that in the second iteration of Step 2, no errors are reported on the enabled Lanes. In Step 3, Die A sends {MBINIT.REPAIRMB apply degrade req} with a Lane map code of 010b. Similarly, in Step 3, Die B sends {MBINIT.REPAIRMB apply degrade req} with a Lane map code of 001b. Because the width of the Tx and Rx have not changed, both Die proceed to Step 4.

4.5.3.4 MBTRAIN

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

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

Figure 4-41. Mainband Training**4.5.3.4.1 MBTRAIN.VALVREF**

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

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

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

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

2. UCIe Module optimizes Vref on its Valid Receiver by adjusting Receiver reference voltage and performing one or more Receiver-initiated Data-to-Clock point tests (see [Section 4.5.1.3](#)) (AND/OR) one or more Receiver-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.4](#)).
 - a. The transmit pattern must be set to send 128 iterations of continuous mode “VALTRAIN” (four 1s and four 0s) pattern (see [Table 4-5](#)). This pattern must not be scrambled. The Receiver must be set up to perform comparison on the Valid Lane.
 - b. Detection on a Receiver Lane is considered successful if “VALTRAIN” pattern detection errors are less than the set threshold (per Lane comparison threshold in [Section 9.5.3.29](#)).
 - c. It should be noted that LFSR RESET has no impact in MBTRAIN.VALVREF.
3. The UCIe Module must send {MBTRAIN.VALVREF end req} sideband message after the Vref optimization (One way to perform Vref Optimization is to step through Vref and perform [Step 2](#) at each setting). When {MBTRAIN.VALVREF end req} is received, the UCIe Module Partner must respond with {MBTRAIN.VALVREF end resp}. When the UCIe Module has sent and received the sideband message {MBTRAIN.VALVREF end resp}, it must exit to MBTRAIN.DATAVREF.

4.5.3.4.2 MBTRAIN.DATAVREF

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

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

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

1. The UCIe Module sends the sideband message {MBTRAIN.DATAVREF start req}. When {MBTRAIN.DATAVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATAVREF start resp}.
2. The UCIe Module optimizes data Vref by adjusting data Receiver reference voltage and performing one or more Receiver-initiated Data-to-Clock point tests (see [Section 4.5.1.3](#)) (AND/OR) one or more Receiver-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.4](#)).
 - a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in [Section 4.4.1](#). The LFSR pattern on Data must be accompanied by correct Valid framing on the Valid Lane as described in [Section 4.1.2](#). The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see [Section 9.5.3.29](#)).
3. The UCIe Module must send {MBTRAIN.DATAVREF end req} sideband message after the Vref optimization (One way to perform Vref Optimization is to step through Vref and perform step 2 at each setting). When {MBTRAIN.DATAVREF end req} is received, the UCIe Module Partner must respond with {MBTRAIN.DATAVREF end resp}. Once the UCIe Module has sent and received the sideband message {MBTRAIN.DATAVREF end resp}, it must exit to MBTRAIN.SPEEDIDLE.

4.5.3.4.3 MBTRAIN.SPEEDIDLE

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

The following rules apply:

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

4.5.3.4.4 MBTRAIN.TXSELCAL

The UCIe Module calibrates its circuit parameters independent of the UCIe Module Partner. Data, Clock, Valid, and Track Transmitters are tri-stated. Data, Clock, Valid, and Track Receivers are permitted to be disabled.

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

4.5.3.4.5 MBTRAIN.RXCLKCAL

In this state, Data, Valid Transmitters are held low (Data and Valid Receivers are permitted to be disabled). When not performing the actions relevant to this state, if Strobe mode was advertised by the UCIe Module partner, the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking). When not performing the actions relevant to this state, if continuous clock mode was advertised by the UCIe Module partner and the {MBTRAIN.RXCLKCAL start req} sideband message has been received, then the Clock Transmitters are providing the free-running forwarded clock.

1. The UCIe Module, when ready to perform calibration on its Clock receive path, sends the {MBTRAIN.RXCLKCAL start req} sideband message. When the {MBTRAIN.RXCLKCAL start req} sideband message is received, the UCIe Module Partner starts sending the forwarded clock and Track. Subsequently, the UCIe Module Partner sends the {MBTRAIN.RXCLKCAL start resp} sideband message. The Transmitter clock must be free running and all Data Lanes and Valid must be held low. The UCIe Module is permitted to use the forwarded clock to perform any clock path-

related and Clock-to-Track-related calibration. The UCIe Module Partner must not adjust any circuit or PI phase parameters on its Transmitters within this state.

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

4.5.3.4.6 MBTRAIN.VALTRAINCENTER

To ensure the valid signal is functional, valid to clock training is performed before the data Lane training. The Receiver samples the pattern on Valid with the forwarded clock. Receiver reference voltage is set to the optimized value achieved through Vref training (see [Section 4.5.3.4.1](#) and [Section 4.5.3.4.2](#)). All data and Track Transmitters are held low during valid to clock training. When not performing the actions relevant to this state, if Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking). When not performing the actions relevant to this state, if continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock.

Following is the MBTRAIN.VALTRAINCENTER sequence:

1. The UCIe Module sends a sideband message {MBTRAIN.VALTRAINCENTER start req}. The UCIe Module Partner responds with {MBTRAIN.VALTRAINCENTER start resp}.
2. The UCIe Module must perform one or more Transmitter-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.2](#)) (AND/OR) one or more Transmitter-initiated Data-to-Clock point tests (see [Section 4.5.1.1](#)) to determine the correct Valid to clock centering.
 - a. The transmit pattern must be set to send 128 iterations of continuous mode "VALTRAIN" (four 1s and four 0s) pattern (see [Table 4-5](#)). This pattern must not be scrambled. The Receiver must be set up to perform comparison on the Valid Lane.
 - b. Detection on a Receiver Lane is considered successful if "VALTRAIN" pattern detection errors are less than set threshold (per Lane comparison threshold in [Section 9.5.3.29](#)).
 - c. It should be noted that LFSR RESET has no impact in MBTRAIN.VALVREF.
3. The UCIe Module can use the received results log to assess valid functionality and margins. Following this, step 4 must be performed.
4. The UCIe Module must send {MBTRAIN.VALTRAINCENTER done req} sideband message. When {MBTRAIN.VALTRAINCENTER done req} is received the UCIe Module Partner responds with {MBTRAIN.VALTRAINCENTER done resp}. Once the UCIe Module has sent and received {MBTRAIN.VALTRAINCENTER done resp} sideband message, the UCIe Module must exit to MBTRAIN.VALTRAINVREF.

4.5.3.4.7 MBTRAIN.VALTRAINVREF

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

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

1. The UCIe Module must send the sideband message {MBTRAIN.VALTRAINVREF start req}. When {MBTRAIN.VALTRAINVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.VALTRAINVREF start resp}.
2. UCIe Module optionally optimizes Vref by adjusting Receiver reference voltage on its Valid Receiver and performing one or more Receiver-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.4](#)) (AND/OR) one or more Receiver-initiated Data-to-Clock point tests (see [Section 4.5.1.3](#)). Step 2 is optional and implementation-specific.
 - a. If Valid centering is performed, the transmit pattern must be set to send 128 iterations of continuous mode "VALTRAIN" (four 1s and four 0s) pattern (see [Table 4-5](#)). This pattern must not be scrambled. The Receiver must be set up to perform comparison on the Valid Lane.
 - b. Detection on a Receiver Lane is considered successful if "VALTRAIN" pattern detection errors are less than set threshold (per Lane comparison threshold in [Section 9.5.3.29](#)).
 - c. It should be noted that LFSR RESET has no impact in MBTRAIN.VALVREF.
3. The UCIe Module must send {MBTRAIN.VALTRAINVREF end req} sideband message after the Vref optimization is complete. When {MBTRAIN.VALTRAINVREF end req} is received, the UCIe Module Partner must respond with {MBTRAIN.VALTRAINVREF end resp}. Once the UCIe Module has sent and received the sideband message {MBTRAIN.VALTRAINVREF end resp}, it must exit to MBTRAIN.DATATRAINCENTER1.

4.5.3.4.8 MBTRAIN.DATATRAINCENTER1

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

- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled
- If Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- If continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock

Following is the MBTRAIN.DATATRAINCENTER1 sequence:

1. The UCIe Module sends the {MBTRAIN.DATATRAINCENTER1 start req} sideband message. When {MBTRAIN.DATATRAINCENTER1 start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER1 start resp}.
2. The UCIe Module must perform one or more Transmitter-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.2](#)) (AND/OR) one or more Transmitter-initiated Data-to-Clock point tests (see [Section 4.5.1.1](#)) to determine the correct data to clock centering and adjust Transmitter per-bit deskew (if needed).
 - a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in [Section 4.4.1](#). The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in [Section 4.1.2](#). The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see [Section 9.5.3.29](#)).

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

4.5.3.4.9 MBTRAIN.DATATRAINVREF

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

- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled
- If Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- If continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock

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

1. The UCIe Module must send the sideband message {MBTRAIN.DATATRAINVREF start req}. When {MBTRAIN.DATATRAINVREF start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINVREF start resp}.
2. UCIe Module optionally optimizes Vref by adjusting Receiver reference voltage and performing one or more Receiver-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.4](#)) (AND/OR) one or more Receiver-initiated Data-to-Clock point tests (see [Section 4.5.1.3](#)). Step 2 is optional and implementation specific. If Data Vref optimization is performed:
 - a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in [Section 4.4.1](#). The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in [Section 4.1.2](#). The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see [Section 9.5.3.29](#)).
3. The UCIe Module must send {MBTRAIN.DATATRAINVREF end req} sideband message after the Vref optimization is complete. When {MBTRAIN.DATATRAINVREF end req} is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINVREF end resp}. Once the UCIe Module has sent and received the sideband message {MBTRAIN.DATATRAINVREF end resp}, it must exit to MBTRAIN.RXDESKEW.

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

4.5.3.4.10 MBTRAIN.RXDESKEW

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

- Clock Receivers are enabled

- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled
- If Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- If continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock

Following is the MBTRAIN.RXDESKEW sequence:

1. The UCIe Module must send the sideband message {MBTRAIN.RXDESKEW start req}. When {MBTRAIN.RXDESKEW start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.RXDESKEW start resp}.
2. UCIe Module optionally performs per Lane deskew on its Receivers by one or more Receiver-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.4](#)) (AND/OR) one or more Receiver-initiated Data-to-Clock point tests (see [Section 4.5.1.3](#)). Step 2 is optional and implementation specific. If per Lane deskew is performed:
 - a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in [Section 4.4.1](#). The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in [Section 4.1.2](#). The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if total error count is less than the set threshold (see [Section 9.5.3.29](#)).
3. The UCIe Module must send {MBTRAIN.RXDESKEW end req} sideband message after the deskew is performed or skipped. When {MBTRAIN.RXDESKEW end req} is received, the UCIe Module Partner must respond with {MBTRAIN.RXDESKEW end resp}. Once UCIe Module has sent and received the sideband message {MBTRAIN.RXDESKEW end resp}, it must exit to MBTRAIN.DATATRAINCENTER2.

4.5.3.4.11 MBTRAIN.DATATRAINCENTER2

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

- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled
- If Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- If continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock

Following is the MBTRAIN.DATATRAINCENTER2 sequence:

1. The UCIe Module sends the sideband message {MBTRAIN.DATATRAINCENTER2 start req}. When {MBTRAIN.DATATRAINCENTER2 start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER2 start resp}.
2. The UCIe Module must perform one or more Transmitter-initiated Data-to-Clock eye width sweeps (see [Section 4.5.1.2](#)) (AND/OR) one or more Transmitter-initiated Data-to-Clock point tests (see [Section 4.5.1.1](#)) to determine the correct data-to-eye centering.

- a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in [Section 4.4.1](#). The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in [Section 4.1.2](#). The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if total error count is less than the set error threshold (see [Section 9.5.3.29](#)).
3. The UCIe Module uses the received training results to calculate the final eye center and set the clock phase to sample the data eye at the optimal point to maximize eye margins. The UCIe Module must send the {MBTRAIN.DATATRAINCENTER2 end req} sideband message. When {MBTRAIN.DATATRAINCENTER2 end req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.DATATRAINCENTER2 end resp}. Once UCIe Module has sent and received {MBTRAIN.DATATRAINCENTER2 end resp} sideband message, it must exit to MBTRAIN.LINKSPEED.

4.5.3.4.12 MBTRAIN.LINKSPEED

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

- Clock Receivers are enabled
- Data and Valid Transmitters are held low
- Data and Valid Receivers are enabled
- If Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking)
- If continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock

The following steps must be executed sequentially, when applicable:

1. The UCIe Module sends the sideband message {MBTRAIN.LINKSPEED start req}. When {MBTRAIN.LINKSPEED start req} sideband message is received, the UCIe Module Partner responds with {MBTRAIN.LINKSPEED start resp}.
2. The UCIe Module must perform a Transmitter-initiated Data-to-Clock point test (see [Section 4.5.1.1](#)) with the final clock sampling phase calculated in the previous MBTRAIN.DATACENTER2 state. LFSR pattern described in [Section 4.4.1](#) must be used in this state.
 - a. The Transmitter must be set up to send 4K UI of continuous mode LFSR pattern described in [Section 4.4.1](#). The LFSR pattern on data must be accompanied by correct valid framing on the Valid Lane as described in [Section 4.1.2](#). The Receiver must be set up to perform per Lane comparison.
 - b. Detection on a Receiver Lane is considered successful if total error count is less than the set threshold (see [Section 9.5.3.29](#)).

3. For single-Module instantiations, if errors are encountered, the UCIe Module sets its Transmitters to an electrical idle state and sends {MBTRAIN.LINKSPEED error req} sideband message. If an {MBTRAIN.LINKSPEED error req} sideband message is received, the UCIe Module Partner must complete [Step 1](#) and [Step 2](#), evaluate the results and if not initiating an {MBTRAIN.LINKSPEED exit to phy retrain req} sideband message, the UCIe Module Partner enters electrical idle on its Receiver and sends the {MBTRAIN.LINKSPEED error resp} sideband message. If an {MBTRAIN.LINKSPEED exit to phy retrain req} sideband message is received, the UCIe Module must exit to PHYRETRAIN and send an {MBTRAIN.LINKSPEED exit to PHY retrain resp} sideband message; any outstanding messages are abandoned. Otherwise, after the {MBTRAIN.LINKSPEED error resp} sideband message is received, the PHY_IN_RETRAIN flag is cleared and the following rules apply:
 - a. Based on the number of Lanes encountering errors, the UCIe Module checks if the failing Lanes can be repaired (for Advanced package) or Width degraded (for standard package). If Lanes can be repaired (for Advanced package) or Width degraded (for standard package), the UCIe Module must send {MBTRAIN.LINKSPEED exit to repair req} to the UCIe Module Partner. The UCIe Module Partner, if not initiating a speed degrade, enters MBTRAIN.REPAIR and sends the sideband message {MBTRAIN.LINKSPEED exit to repair resp}. If {MBTRAIN.LINKSPEED exit to repair resp} is received in response to a {MBTRAIN.LINKSPEED exit to repair req}, the UCIe Module must exit to MBTRAIN.REPAIR. If a UCIe Module is initiating a speed degrade, it must not respond to {MBTRAIN.LINKSPEED exit to repair req}.
 - b. If the Lanes cannot be repaired (for Advanced package) or width degraded (for Standard package), the speed must be degraded. The UCIe Module sends {MBTRAIN.LINKSPEED exit to speed degrade req} sideband message and waits for a response from the remote Link partner. The UCIe Module Partner must respond with {MBTRAIN.LINKSPEED exit to speed degrade resp}. Following this handshake, the UCIe Module must exit to MBTRAIN.SPEEDIDLE to set data rate to next lower speed.
 - c. If the UCIe Module receives an {MBTRAIN.LINKSPEED exit to speed degrade req} any outstanding {MBTRAIN.LINKSPEED exit to repair req} must be abandoned and the UCIe Module must respond to {MBTRAIN.LINKSPEED exit to speed degrade req}.
 - d. Any outstanding {MBTRAIN.LINKSPEED done req} must be abandoned if a UCIe Module has received a {MBTRAIN.LINKSPEED error req}.
4. For single- or multi-module instantiations, if no errors are encountered, the UCIe Module must set the clock phase on its Transmitter to sample the data eye at the optimal point to maximize eye margins. If PHY_IN_RETRAIN is not set for single-module instantiations, proceed to [Step 6](#). If PHY_IN_RETRAIN is not set, for multi-module instantiations, the UCIe Module must send the {MBTRAIN.LINKSPEED done req} (if not waiting for Link match criteria as a Retimer) to the remote UCIe Module Partner and wait for multi-module PHY Logic (MMPL) resolution in [Step 5c](#). If the PHY_IN_RETRAIN variable is set, the following actions must be taken:
 - a. If a change is detected in Runtime Link Testing Control register relative to the values at previous PHYRETRAIN entry, the UCIe Module must send {MBTRAIN.LINKSPEED exit to phy retrain req} and wait for a response. Upon receiving this message, the UCIe Module Partner must exit to PHY retrain and send {MBTRAIN.LINKSPEED exit to PHY retrain resp}. Once this sideband message is received, the UCIe Module must exit to PHY retrain.
 - b. Else if no change is detected in the Runtime Link Testing Control register relative to the values at previous PHYRETRAIN entry, Busy bit in Runtime Link Testing Status and PHY_IN_RETRAIN variable must be cleared and the UCIe Module must proceed to [Step 6](#).

5. For multi-module instantiations, if errors are encountered, the UCIe Module sets its Transmitters to an electrical idle state and sends {MBTRAIN.LINKSPEED error req} sideband message. If an {MBTRAIN.LINKSPEED error req} sideband message is received, the UCIe Module Partner must complete [Step 1](#) and [Step 2](#), evaluate the results and if not initiating an {MBTRAIN.LINKSPEED exit to phy retrain req} sideband message, the UCIe Module Partner enters electrical idle on its Receiver, and sends the {MBTRAIN.LINKSPEED error resp} sideband message. If an {MBTRAIN.LINKSPEED exit to phy retrain req} sideband message is received, the UCIe Module must exit to PHYRETRAIN and send an {MBTRAIN.LINKSPEED exit to PHY retrain resp} sideband message; any outstanding messages are abandoned. Otherwise, after the {MBTRAIN.LINKSPEED error resp} sideband message is received, the PHY_IN_RETRAIN flag is cleared and the following rules apply:
 - a. Based on the number of Lanes encountering errors, the UCIe Module checks whether the failing Lanes can be repaired (for Advanced Package) or Width degraded (for Standard Package). If Lanes can be repaired (for Advanced Package) or Width degraded (for Standard Package), the UCIe Module must send {MBTRAIN.LINKSPEED exit to repair req} to the UCIe Module Partner.
 - b. If the Lanes cannot be repaired (for Advanced Package) or width degraded (for Standard Package), the speed must be degraded. The UCIe Module sends {MBTRAIN.LINKSPEED exit to speed degrade req}.
 - c. The UCIe Module informs MMPL of local and remote error requests, done requests, or speed degrade requests, and waits for resolution. It also informs MMPL of any prior width degrade (for example in MBINIT.REPAIRMB), and MMPL treats this as the corresponding module requesting width degrade from the full operational width.
 - d. Based on the resolution flow chart in [Section 4.7](#), MMPL directs each Module to send either the {MBTRAIN.LINKSPEED exit to repair resp} (indicating next state is REPAIR), {MBTRAIN.LINKSPEED exit to speed degrade resp} (indicating next state is SPEEDIDLE with target speed to next-lower speed), {MBTRAIN.LINKSPEED multi-module disable module resp} (indicating next state is TRAINERROR and eventually RESET), or {MBTRAIN.LINKSPEED done resp} (indicating next state is LINKINIT). This is done regardless of the module's original error request or done request, and indicates the result of the resolution and next state to each module. The UCIe Module transitions to next state once it has sent and received the sideband response message that matches the expected resolution. Any mismatch on received message vs. expected resolution must take all modules to TRAINERROR. For Retimer dies, the resolution must take into account any Link match requirements, and while resolving the target configuration with remote Retimer partner, each UCIe Module from the Retimer die must send {MBTRAIN.LINKSPEED done resp} with stall encoding every 4 ms. The UCIe Retimer must ensure that this stall is not perpetual, and an implementation-specific timeout must be included in the Retimer. If {MBTRAIN.LINKSPEED done resp} with stall encoding is received, it must reset timers for state transition as well as any outstanding handshakes for multi-module resolution.

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

4.5.3.4.13 MBTRAIN.REPAIR

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

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

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

For a **x16 Standard package**, if the Lanes encountering errors are all contained within the set of Lane 0 through Lane 7 or Lane 8 through Lane 15, the width must be degraded to a x8 Link using the set of Lanes without errors (Lane 0 ... Lane 7 OR Lane 8 ... Lane 15). Likewise, for a **x8 Standard package**, if the Lanes encountering errors are all contained within the set of Lane 0 through Lane 3 or Lane 4 through Lane 7, the width must be degraded to a x4 Link using the set of Lanes without errors (Lane 0 through Lane 3 or Lane 4 through Lane 7).

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

2. The UCIe Module must send the {MBTRAIN.REPAIR apply degrade req} sideband message, indicating the functional Lanes on its Transmitter using one of the logical Lane map encodings from [Table 4-9](#). Encodings 100b and 101b can be used only when the SPMW bit is set to 1 or the UCIe-S x8 bit was set to 1 in the transmitted or received {MBINIT.PARAM configuration req} sideband message. If the remote Link partner indicated a width degrade in the functional Lanes, the UCIe Module must apply the corresponding width degrade to its Receiver. If the remote Link partner indicated all Lanes are functional, the UCIe Module sets its Transmitter and Receiver to the logical lane map corresponding to the functional Lane encoding determined on its Transmitter. The UCIe Module sends the {MBTRAIN.REPAIR apply degrade resp} sideband message after setting its Transmitter and Receiver lanes to the relevant logical lane map and proceeds to [Step 3](#) if a degrade is possible or if all Lanes are functional. If a "Degrade not possible" encoding is sent or received in the {MBTRAIN.REPAIR apply degrade req} sideband message, the UCIe Module must exit to TRAINERROR after performing the TRAINERROR handshake.
3. The UCIe Module must send the {MBTRAIN.REPAIR end req} sideband message and wait for a response. The UCIe Module Partner must then respond with the {MBTRAIN.REPAIR end resp} sideband message. When UCIe Module has sent and received the {MBTRAIN.REPAIR end resp} sideband message, the UCIe Module must exit to MBTRAIN.TXSELCAL.

4.5.3.5 LINKINIT

This state is used to allow die to die adapter to complete initial Link management before entering Active state on RDI. See [Section 10.1.6](#) for more details on RDI bring up flow. Track, Data, and Valid Transmitters are held low. If Strobe mode was advertised by the UCIe Module partner, then the Clock Transmitters are held differential low (for differential clocking) or simultaneous low (for Quadrature clocking). If continuous clock mode was advertised by the UCIe Module partner, then the Clock Transmitters are providing the free-running forwarded clock. Clock Receivers are enabled.

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

The LFSR must be RESET upon entering this state.

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

4.5.3.6 ACTIVE

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

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

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

4.5.3.7 PHYRETRAIN

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

- Adapter directed PHY retrain: Adapter can direct the PHY to retrain for any reason it deems necessary (see [Section 10.3.3.4](#) Retrain State rules for more details and examples of Adapter-initiated Retrain requests).
- PHY initiated PHY retrain: Local PHY must initiate retrain on detecting a Valid framing error.

- Remote die requested PHY retrain: Local PHY must enter PHY retrain on receiving a request from the remote die.
- If a change is detected in Runtime Link Testing Control register during MBTRAIN.LINKSPEED.

A variable PHY_IN_RETRAIN must be set when entering PHYRETRAIN. For a multi-module configuration, the Retrain encoding (and hence the Retrain exit resolution) must be the same for all the modules which are part of the same multi-module Link. This is required because in a multi-module Link all modules must operate at the same speed and width (however, for Advanced package, it is possible for a module not needing repair to go through the Repair state and send the "No Repair" encoding in the corresponding {*apply repair*} messages).

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

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

Table 4-11. Retrain encoding

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

Table 4-12. Retrain exit state resolution

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

4.5.3.7.1 Adapter initiated PHY retrain

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

1. UCIe Module receives retrain request from the local Adapter (RDI state req moved to Retrain). Following this, the UCIe Module must complete the stall Req/Ack (`p1_stallreq; lp_stallack`) hand shake on RDI as described in [Chapter 10.0](#).
2. After completion of stall Req/Ack handshake and transmitting any pending data over mainband, UCIe Module must send sideband message {LinkMgmt.RDI.Req.Retrain} to UCIe Module Partner.
3. The UCIe Module Partner on receiving the sideband message {LinkMgmt.RDI.Req.Retrain} must transition its RDI state to Retrain after completion of stall Req/Ack (`p1_stallreq; lp_stallack`) handshake on its RDI and there is no mainband data pending in the Receiver pipeline. After completion of stall Req/Ack handshake and transmitting any pending data over mainband, the UCIe Module Partner responds with {LinkMgmt.RDI.Rsp.Retrain}.
4. Once {LinkMgmt.RDI.Rsp.Retrain} is received and there is no mainband data pending in the receiver pipeline, the UCIe Module must transition its RDI to Retrain.
5. UCIe Module must send {PHYRETRAIN.retrain start req} with retrain encoding reflecting the contents of Runtime Link Test Control register except the Start bit (if the Busy bit in Runtime Link Test Status Register is set). Following this, the UCIe Module Partner compares the received retrain encoding with the local retrain encoding. If received retrain encoding is the same as the local retrain encoding, the UCIe Module Partner must respond with {PHYRETRAIN.retrain start resp}. If the retrain encodings do not match, the UCIe Module Partner must resolve according to Retrain encodings and resolutions shown in [Table 4-10](#), [Table 4-11](#), and [Table 4-12](#) and then send {PHYRETRAIN.retrain start resp} with the resolved retrain encoding.
6. Once UCIe Module sends and receives the sideband message {PHYRETRAIN.retrain start resp}, it must exit to corresponding training state according to the resolved retrain register encoding.

4.5.3.7.2 PHY initiated PHY retrain

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

1. On detecting a valid framing error, the UCIe Module must assert `p1_error` when transmitting that flit (or flit chunk) on RDI. Following this the UCIe Module (PHY) must complete the stall Req/Ack (`p1_stallreq; lp_stallack`) handshake on RDI.
2. The UCIe Module must send sideband message {LinkMgmt.RDI.Req.Retrain}.
3. The UCIe Module Partner on receiving the sideband message {LinkMgmt.RDI.Req.Retrain} must transition its RDI to retrain after completion of stall Req/Ack (`p1_stallreq; lp_stallack`) handshake on its RDI. Following that the UCIe Module Partner responds with {LinkMgmt.RDI.Rsp.Retrain}.
4. Once {LinkMgmt.RDI.Rsp.Retrain} is received, the UCIe Module must transition its RDI to Retrain.
5. UCIe Module must send {PHYRETRAIN.retrain start req} with retrain encoding reflecting the contents of Runtime Link Test Control register except the Start bit (if the Busy bit in Runtime Link Test Status Register is set). Following this, the UCIe Module Partner compares the received retrain encoding with the local retrain encoding. If received retrain encoding is the same as the local retrain encoding, the UCIe Module Partner must respond with {PHYRETRAIN.retrain start resp}. If the retrain encodings do not match, the UCIe Module Partner must resolve according to Retrain encodings and resolutions shown in [Table 4-10](#), [Table 4-11](#), and [Table 4-12](#) and then send {PHYRETRAIN.retrain start resp} with the resolved retrain encoding.
6. Once a UCIe Module has sent and received the sideband message {PHYRETRAIN.retrain start resp}, it must exit to corresponding training state according to the resolved retrain encoding.

4.5.3.7.3 Remote Die requested PHY retrain

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

4.5.3.7.4 PHY retrain from LINKSPEED

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

4.5.3.8 TRAINERROR

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

The exit from TRAINERROR to RESET is implementation specific. For cases when there is no error escalation (i.e., RDI is not in LinkError), it is recommended to exit TRAINERROR as soon as possible. For cases when there is error escalation (i.e., RDI is in LinkError), it is required for Physical Layer to be in TRAINERROR as long as RDI is in LinkError. To avoid problems with entering RESET while transmitting sideband packets, any in-progress sideband packets must finish transmission before entering RESET state.

See [Chapter 10.0](#) for correctable, non-fatal, and fatal error escalation on RDI.

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

If sideband is Active, a sideband handshake must be performed to enter TRAINERROR state from any state other than SINIT. The following is defined as the TRAINERROR handshake:

- The UCIe Module requesting exit to TRAINERROR must send {TRAINERROR Entry req} sideband message and wait for a response. The UCIe Module Partner must exit to TRAINERROR and

respond with {TRAINERROR Entry resp}. Once {TRAINERROR Entry resp} sideband message is received, the UCIe Module must exit to TRAINERROR. If no response is received for 8 ms, the LTSM transitions to TRAINERROR.

4.5.3.9 L1/L2

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

- This state is entered when RDI has transitioned to PM state as described in [Chapter 10.0](#). The PHY power saving features in this state are implementation specific.
- When local Adapter requests Active on RDI or remote Link partner requests L1 exit the PHY must exit to MBTRAIN.SPEEDIDLE. L1 exit is coordinated with the corresponding L1 state exit transitions on RDI.
- When local Adapter requests Active on RDI or remote Link partner requests L2 exit the PHY must exit to RESET. L2 exit is coordinated with the corresponding L2 state exit transitions on RDI.

4.6 Runtime Recalibration

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

The following sequence is used to request track pattern:

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

4.7 Multi-module Link

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

4.7.1 Multi-module initialization

Each module in a multi-module configuration must initialize and train independently, using its sideband. If two or four modules are used, a separate multi-module PHY logic block coordinates across the modules, as described in [Section 1.2.2](#). The MMPL is responsible for orchestrating data transfer and any associated byte swizzling for the Transmitters across the multiple modules such that the remote Link partner's Receivers observe the correct byte-to-Lane mapping (i.e., for any valid transfer, bytes are laid out from LSB to MSB in ascending order of Module ID and Lane ID across all the active Lanes). [Figure 4-42](#), [Figure 4-43](#), [Figure 4-44](#), and [Figure 4-45](#) illustrate examples of the aforementioned byte swizzling for some of the Standard Package configurations. M0, M1, M2, and M3 in the figures correspond to Module ID 0, Module ID 1, Module ID 2, and Module ID 3, respectively. The figures provide the RDI byte-to-Module mapping. [Figure 4-42](#) shows the scenario in which the remote Link partner's Module ID is the same. [Figure 4-43](#) shows an example of a scenario in which the remote Link partner's Module ID is different. [Figure 4-44](#) shows an example of width degradation for Standard package in which the remote Link partner's Module ID is different (note that the bytes are laid out in ascending order of Module ID and Lane ID across all the active Lanes at the Receiver in all cases). [Figure 4-45](#) shows a scenario in which two modules are disabled. This corresponds to a case outlined in [Chapter 5.0](#) in which a stacked configuration is connected with an unstacked configuration and the M0 and M2 modules are disabled; the remaining bytes of RDI are sent over subsequent 8-UI intervals such that M1 on the remote Link partner receives the least significant bytes.

Figure 4-42. Example of Byte Mapping for Matching Module IDs

Module Name	M1	M0	M2	M3				
Data Bytes	RDI B16 --- RDI B31	RDI B16 --- RDI B31	RDI B0 --- RDI B15	RDI B0 --- RDI B15	RDI B32 --- RDI B47	RDI B32 --- RDI B47	RDI B48--- RDI B63	RDI B48--- RDI B63
Tx/Rx	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)
Tx/Rx	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)
Data Bytes	RDI B16 --- RDI B31	RDI B16 --- RDI B31	RDI B0 --- RDI B15	RDI B0 --- RDI B15	RDI B32 --- RDI B47	RDI B32 --- RDI B47	RDI B48--- RDI B63	RDI B48--- RDI B63
Module Name	M1	M0	M2	M3				

Figure 4-43. Example of Byte Mapping for Differing Module IDs

Module Name	M1	M0	M2	M3				
Data Bytes	RDI B16 --- RDI B31	RDI B48 --- RDI B63	RDI B0 --- RDI B15	RDI B32 --- RDI B47	RDI B32 --- RDI B47	RDI B0 --- RDI B15	RDI B48--- RDI B63	RDI B16--- RDI B31
Tx/Rx	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)
Tx/Rx	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)
Data Bytes	RDI B16 --- RDI B31	RDI B48 --- RDI B63	RDI B0 --- RDI B15	RDI B32 --- RDI B47	RDI B32 --- RDI B47	RDI B0 --- RDI B15	RDI B48--- RDI B63	RDI B16 --- RDI B31
Module Name	M3	M2	M0	M1				

Figure 4-44. Example of Width Degradation with Byte Mapping for Differing Module IDs

Module Name	M1	M0	M2	M3				
Data Bytes UI 15-8	RDI B40 --- RDI B47	RDI B57-- RDI B63	RDI B32 --- RDI B39	RDI B48 --- RDI B55	RDI B48 --- RDI B55	RDI B32 --- RDI B39	RDI B57-- RDI B63	RDI B40 --- RDI B47
Data Bytes UI 7-0	RDI B8 --- RDI B15	RDI B24-- RDI B31	RDI B0 -- RDI B7	RDI B16 --- RDI B23	RDI B16 --- RDI B23	RDI B0 -- RDI B7	RDI B24-- RDI B31	RDI B8 -- RDI B15
Tx/Rx	Rx (x8)	Tx (x8)	Rx (x8)	Tx (x8)	Rx (x8)	Tx (x8)	Rx (x8)	Tx (x8)
Tx/Rx	Tx (x8)	Rx (x8)	Tx (x8)	Rx (x8)	Tx (x8)	Rx (x8)	Tx (x8)	Rx (x8)
Data Bytes UI 7-0	RDI B8 -- RDI B15	RDI B24 -- RDI B31	RDI B0 -- RDI B7	RDI B16 --- RDI B23	RDI B16 --- RDI B23	RDI B0 -- RDI B7	RDI B24-- RDI B31	RDI B8 -- RDI B15
Data Bytes UI 15-8	RDI B40 -- RDI B47	RDI B57-- RDI B63	RDI B32 --- RDI B39	RDI B48 --- RDI B55	RDI B48 --- RDI B55	RDI B32 --- RDI B39	RDI B57-- RDI B63	RDI B40 -- RDI B47
Module Name	M3	M2	M0	M1				

Figure 4-45. Example of Byte Mapping with Module Disable

Module Name	M1	M0	M2	M3		
Data Bytes	RDI B0 --- RDI B15	RDI B16 --- RDI B31			RDI B16 --- RDI B31	RDI B0 --- RDI B15
Tx/Rx	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)	Rx (x16)	Tx (x16)
			Disabled		Disabled	
Tx/Rx	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)	Tx (x16)	Rx(x16)
Data Bytes	RDI B0 -- RDI B15	RDI B16 -- RDI B31			RDI B16 -- RDI B31	RDI B0 -- RDI B15
Module Name	M3	M2	M0	M1		

Each module in a multi-module Link must operate at the same width and speed. During initialization or retraining, if any module failed to train, the MMPL must ensure that the multi-module configuration degrades to the next permitted configuration for width or speed degrade (see [Figure 4-46](#) and [Figure 4-47](#)). Subsequently, any differences in width and speed between the different modules must be resolved using the following rules:

1. For Standard package multi-module configuration, if width degrade is reported for any of the modules:
 - a. If less than or equal to half the number of modules report width degrade at the current Link speed, the corresponding Modules must be disabled. The MMPL must ensure that the multi-module configuration degrades to the next permitted configuration for width or speed degrade (see [Figure 4-46](#) and [Figure 4-47](#)). For example, if three out of four modules are active, the MMPL must degrade the Link to a two-module configuration.
 - b. If the majority of modules report width degrade at the current Link speed, see the pseudo code below:

```

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

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

```

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

Figure 4-46 and Figure 4-47 provide a consolidated view of the above two rules as a flow chart that Advanced Package and Standard Package implementations, respectively, must follow. Note that the "Yes" condition for $HMLS/2 > CMLS$ question is there to cover the base case of 4 GT/s. In other words, if some module(s) passed MBINIT but failed 4 GT/s in LinkSpeed, then the "Yes" arc will result in module disable instead of TrainError (because CMLS will be 0 for that) and provide the opportunity to remain operational at 4 GT/s for the modules that were still operational at 4 GT/s.

Figure 4-46. Decision Flow Chart for Multi-module Advanced Package

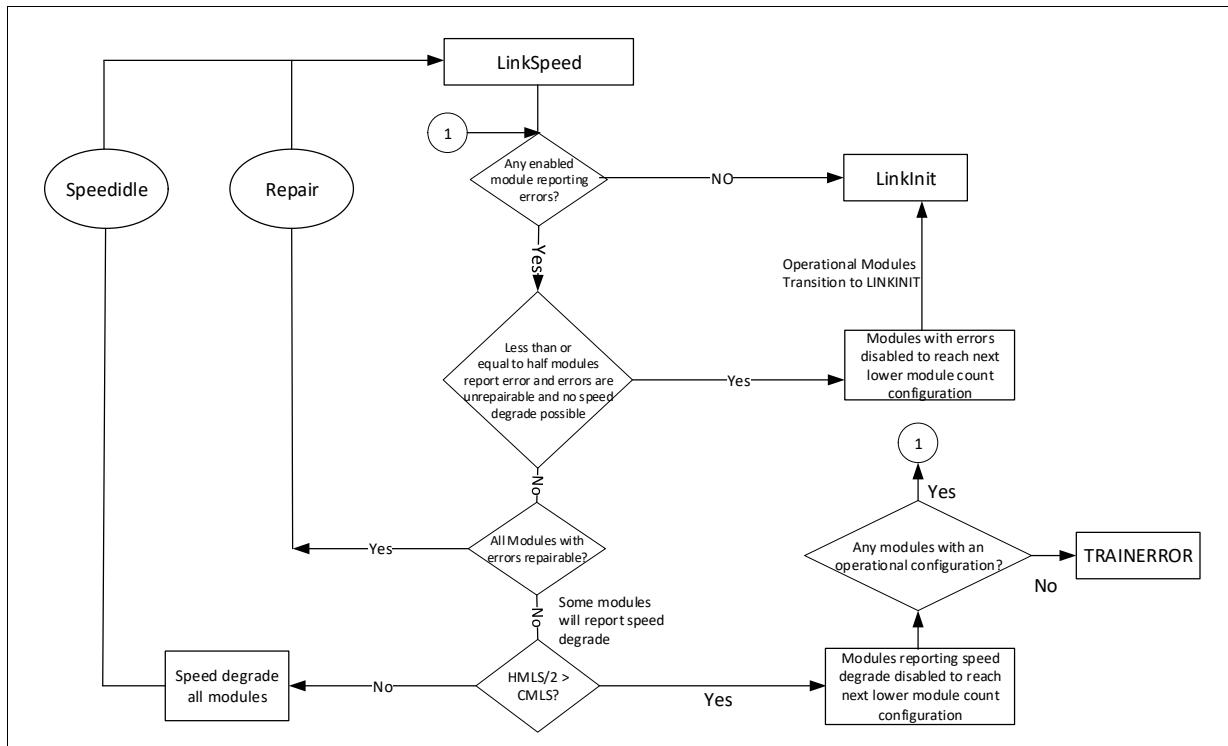
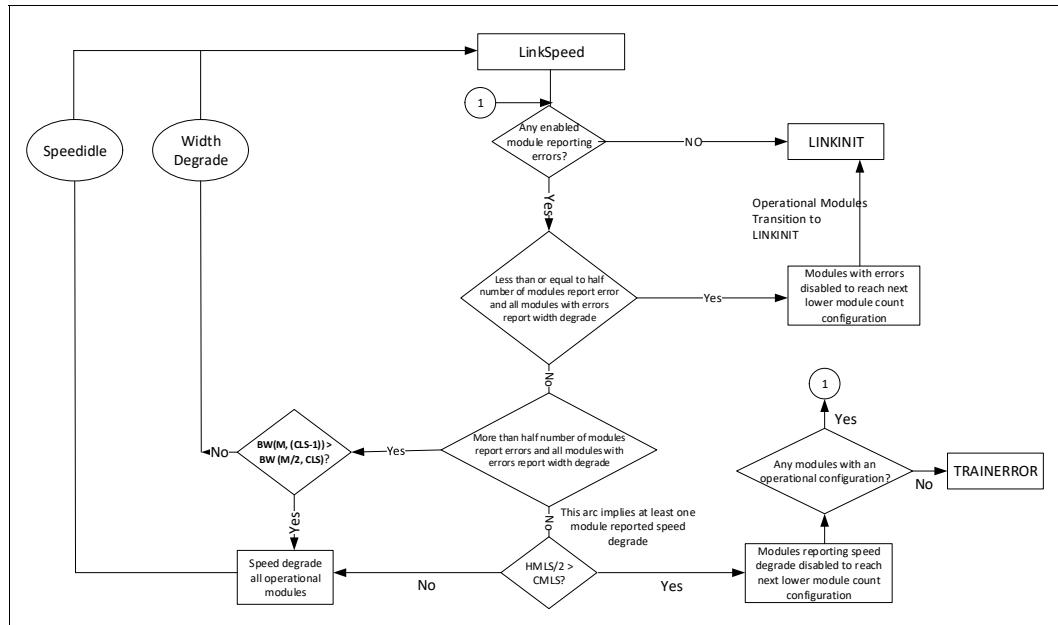


Figure 4-47. Decision Flow Chart for Multi-module Standard Package

4.7.1.1 Sideband Assignment and Retimer Credits for Multi-module Configurations

During Link initialization, training and retraining (see [Section 4.5](#)) certain sideband packets are sent on individual module sideband interfaces. These include all the messages in [Table 7-9](#) and [Table 7-11](#), and any vendor-defined messages that were defined as such.

Other sideband packets use a single sideband to send sideband packets. These include Register Access packets (requests as well as completions), all the non-vendor defined messages in [Table 7-8](#) and [Table 7-10](#), and any vendor-defined messages that were defined as such. A device must send these sideband packets on the sideband interface of the numerically least Module ID whose LTSM is not in RESET or SBINIT. A packet sent on a given Module ID could be received on a different Module ID on the sideband Receiver.

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

4.7.1.2 Examples of MMPL Synchronization

When a module is part of a multi-module UCIe Link, it performs individual training steps through [Step 2](#) of MBTRAIN.LINKSPEED independent of other modules using its own sideband Link. This is true for both Link Initialization and Link Retraining. The common Link parameters exchanged in MBINIT.PARAM are the same for all modules that are part of the multi-module Link (for example the "Maximum Data Rate"). Thus, when in MBTRAIN.LINKSPEED, all modules of a multi-module Link are operating at the same data rate.

The synchronization orchestration by MMPL happens in the MBTRAIN.LINKSPEED state based on the rules outlined in [Section 4.7.1](#). As outlined in [Section 4.5.3.4.12](#), after [Step 2](#) of MBTRAIN.LINKSPEED has completed and PHY_IN_RETRAIN is not set:

- If no errors are encountered for a module, an {MBTRAIN.LINKSPEED done req} is sent to the remote Link partner module
- If errors are encountered for a module, an {MBTRAIN.LINKSPEED error req}/ {MBTRAIN.LINKSPEED error resp} handshake is performed followed by sending an {MBTRAIN.LINKSPEED exit to repair req} or an {MBTRAIN.LINKSPEED exit to speed degrade req} on that module's sideband.

The individual modules notify the MMPL of the sent and received information from these sideband messages. MMPL collects this information from all the modules which are operational in the Link and determines the next state based on resolution of the rules outlined in [Section 4.7.1](#). Of course, the case without errors is when all modules sent and received the {MBTRAIN.LINKSPEED done req} message with no change to Link width. In this scenario, the MMPL directs them to proceed to [Step 6](#) of MBTRAIN.LINKSPEED.

The following sections cover a few examples of this resolution for a Link with four modules and Standard Package configuration where errors were encountered.

4.7.1.2.1 Example 1: MMPL Resolution results in a Width Degrade per Module

In this example, the four modules of the UCIe Link are in MBTRAIN.LINKSPEED at 8 GT/s. [Table 4-13](#) shows the exchanged messages for one die (in the case where errors are encountered, it is assumed that the {MBTRAIN.LINKSPEED error req}/ {MBTRAIN.LINKSPEED error resp} has completed before the messages shown). Because the resolution is consistent in using the sent and received messages, both die of the Link will reach the same resolution.

Table 4-13. Messages exchanged that are used to determine resolution for Example 1

Module Identifier	Sent Message	Received Message
Module 0	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED done req}
Module 1	{MBTRAIN.LINKSPEED exit to repair req}	{MBTRAIN.LINKSPEED done req}
Module 2	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED exit to repair req}
Module 3	{MBTRAIN.LINKSPEED exit to repair req}	{MBTRAIN.LINKSPEED exit to repair req}

In this example, 3 out of the 4 modules have either sent or received the {MBTRAIN.LINKSPEED exit to repair req} message which indicate a width degrade for Standard Package configurations. The value of "CLS" (Current Link Speed) is 8 GT/s, and the value of "CLS-1" is 4 GT/s. The value of "M" (Number of Active Modules) is 4. Because BW (4 Links at 4 GT/s) is not greater than BW (2 Links at 8 GT/s), the flow chart in [Figure 4-46](#) would result in the MMPL notifying all the modules to proceed with a width degrade by moving to MBTRAIN.REPAIR as the next state (i.e., {MBTRAIN.LINKSPEED exit to repair resp} will be sent on each Module). Note that "CLS" and "M" are re-computed using the updated information every time the Link is MBTRAIN.LINKSPEED and there is a corresponding MMPL resolution. Width degrade is applied per module following the steps in MBTRAIN.REPAIR for every module in this UCIe Link. For the module where no errors were encountered, the transmitter is permitted to pick either of Lanes 0 to 7 or Lanes 8 to 15 as the operational Lanes when transmitting the {MBTRAIN.REPAIR apply degrade req} to the remote Link partner module. Following the exit from MBTRAIN.REPAIR, the training continues through the substates of MBTRAIN and in the next iteration of MBTRAIN.LINKSPEED, if no errors are encountered, MMPL will direct the modules to proceed to [Step 6](#) of MBTRAIN.LINKSPEED.

Note that this example is covering the case where errors occurred during the LINKSPEED state. If the width of a module is already lower from the rest of the operational modules that are part of a multi-module Link (e.g., if a module had degraded width during MBINIT.REPAIRMB itself), it may have sent and received {MBTRAIN.LINKSPEED done req} during LINKSPEED. However, from a MMPL resolution

perspective, MMPL must treat this as a Module reporting errors requiring width degrade. This is because a multi-module Link requires all modules to operate at the same width and speed.

4.7.1.2.2 Example 2: MMPL Resolution results in a Speed Degrade

In this example, the four modules of the UCIe Link are in MBTRAIN.LINKSPEED at 16 GT/s. Table 4-14 shows the exchanged messages for one die (in the case where errors are encountered, it is assumed that the {MBTRAIN.LINKSPEED error req}/ {MBTRAIN.LINKSPEED error resp} has completed before the messages shown). Because the resolution is consistent in using the sent and received messages, both die of the Link will reach the same resolution.

Table 4-14. Messages exchanged that are used to determine resolution for Example 2

Module Identifier	Sent Message	Received Message
Module 0	{MBTRAIN.LINKSPEED exit to repair req}	{MBTRAIN.LINKSPEED done req}
Module 1	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED done req}
Module 2	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED done req}
Module 3	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED exit to speed degrade req}

In this example, Module 3 has received a message indicating that the remote partner wants to speed degrade. "CMLS" always maps to the next degraded Link speed and so in this case "CMLS" is 12 GT/s. "HMLS" always ends up mapping to current Link speed and so in this case it is 16 GT/s. Because Module 3 received a speed degrade request, following the flow chart in Figure 4-47, this would result in MMPL notifying all the modules to proceed with a speed degrade by moving to MBTRAIN.SPEEDIDLE (i.e. {MBTRAIN.LINKSPEED exit to speed degrade resp} will be sent on each Module). Following the exit from MBTRAIN.SPEEDIDLE, the training continues through the substates of MBTRAIN and in the next iteration of MBTRAIN.LINKSPEED, if no errors are encountered, MMPL will direct the modules to proceed to Step 6 of MBTRAIN.LINKSPEED. Note that "CMLS" and "HMLS" are using the updated information every time the Link is MBTRAIN.LINKSPEED and there is a corresponding MMPL resolution. In this example, for the next iteration, CMLS will be 8 GT/s and HMLS will be 12 GT/s.

4.7.1.2.3 Example 3: MMPL Resolution results in a Module Disable

In this example, the four modules of the UCIe Link are in MBTRAIN.LINKSPEED at 16 GT/s. Table 4-15 shows the exchanged messages for one die (in the case where errors are encountered, it is assumed that the {MBTRAIN.LINKSPEED error req}/ {MBTRAIN.LINKSPEED error resp} has completed before the messages shown). Because the resolution is consistent in using the sent and received messages, both die of the Link will reach the same resolution.

Table 4-15. Messages exchanged that are used to determine resolution for Example 3

Module Identifier	Sent Message	Received Message
Module 0	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED done req}
Module 1	{MBTRAIN.LINKSPEED exit to repair req}	{MBTRAIN.LINKSPEED done req}
Module 2	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED done req}
Module 3	{MBTRAIN.LINKSPEED done req}	{MBTRAIN.LINKSPEED done req}

Because less than half of the modules are reporting errors and requesting a width degrade, as per the flow chart in Figure 4-47, MMPL would take the configuration to a two module configuration. As per the rules in Section 5.7.3.4.1, Module 0 and Module 1 would send the {MBTRAIN.LINKSPEED multi-module disable module resp} to take these modules to TRAINERROR and RESET. Module 2 and Module 3 would send the {MBTRAIN.LINKSPEED done resp} to take them to LINKINIT.

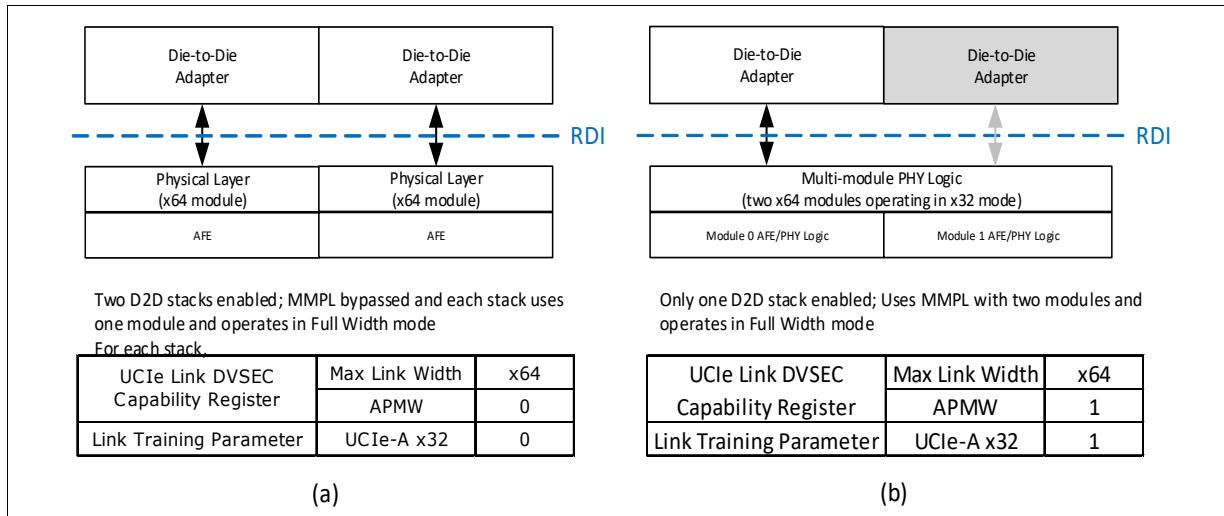
4.7.2 Multi-module Interoperability between x64 and x32 Advanced Packages

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

One example of interoperation between UCIe-A x64 and UCIe-A x32 is when the UCIe-A x64 Stack (including RDI and FDI maximum throughput) is bandwidth-matched (Full Width Mode) by the remote Link partner's maximum throughput for a given interface. [Figure 4-48](#) shows an example of two x64 modules that are capable of operating as two independent UCIe stacks with independent Adapters and Protocol Layers (bypass MMPL logic in configuration (a) in [Figure 4-48](#)) and is also capable of operating as a multi-module configuration when connected to a corresponding multi-module configuration of x32 Advanced Package to achieve the equivalent bandwidth of a single x64 module (configuration (b) in [Figure 4-48](#)). In the latter configuration, one of the Adapters (shown in gray) is disabled.

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

Figure 4-48. Implementation Example Showing Two Different Operating Modes of the Same Hardware Implementation



Another example of interoperation between UCIe-A x64 and UCIe-A x32 is when the UCIe-A x64 Stack degrades bandwidth (Degraded Width Mode) to match the remote Link partner's maximum throughput. [Figure 4-49](#) shows an example of RDI byte-to-module assignments for a four-module set of x64 Advanced Package modules interoperating with a four-module set of x32 Advanced Package modules. The example is for a 256B RDI width on the x64 set, and a 128B RDI on the x32 set. On the Transmitter side of x64 modules, the MMPL throttles RDI, as required, because the MMPL can only send half the bytes over 8 UI; and on the Receiver side, the MMPL accumulates 16 UI worth of data before forwarding it over RDI (assumes data transfers are in chunks of 256B OR appropriate pause of data stream indications are applied and detected by MMPLs/Adapters within the data stream).

Figure 4-49. RDI Byte-to-Module Assignment Example for x64 Interop with x32

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

For the example shown in Figure 4-49, a single Adapter is operating with all four Modules. The D2D stack uses MMPL with 4 modules, with each of the x64 modules operating in Degraded Width Mode, and only 32 lanes routed per module. The corresponding values in the capability register and Link Training parameter are as listed in Table 4-16.

Table 4-16. Capability Register and Link Training Parameter Values for RDI Byte-to-Module Assignment Example for x64 Interop with x32

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

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

4.8 Sideband PHY Arbitration between MPMs and Link Management Packets

While the exact arbitration policy is implementation specific, care should be taken to avoid delaying transmitting any pending link management packets for extended lengths of time potentially causing timeouts. See Section 8.2.5.1.2 for length restriction on MPMs with Data that allows for PHY arbitration to provide an upper bound on the amount of delay to send a link management packet or a higher-priority MPM packet that might be waiting behind an MPM with Data packet. Additionally, PHY transmitter must fully complete transmission of a MPM with Data within 512 UI max (when Sideband Performant Mode Operation is negotiated) and 768 UI max (when Sideband Performant Mode

Operation is not negotiated). See [Section 4.1.5](#) for details of sideband transmission UI notation. See [Section 4.1.5.1](#) for Performant Mode Operation (PMO) details.

[Figure 4-50](#) and [Figure 4-51](#) show the arbitration at the PHY.

Figure 4-50. Example of Encapsulated MTPs Transmitted on Sideband Link without Sideband PMO

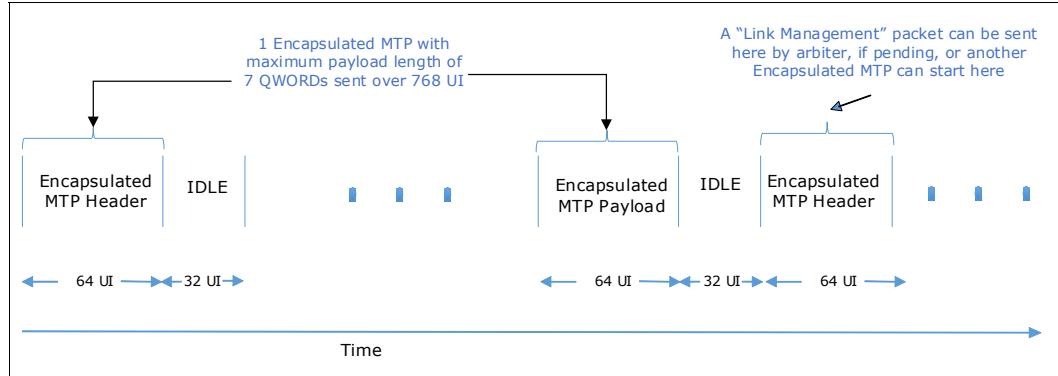
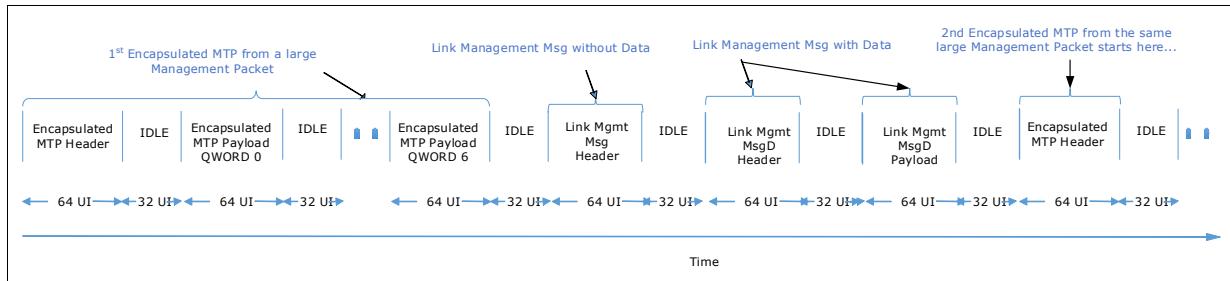


Figure 4-51. Example of a Large Management Packet Split into Two Encapsulated MTPs, with No Segmentation, No Sideband PMO, and with Two Link Management Packets between the Two Encapsulated MTPs



§ §

5.0 Electrical Layer (2D and 2.5D)

Key attributes of electrical specification include:

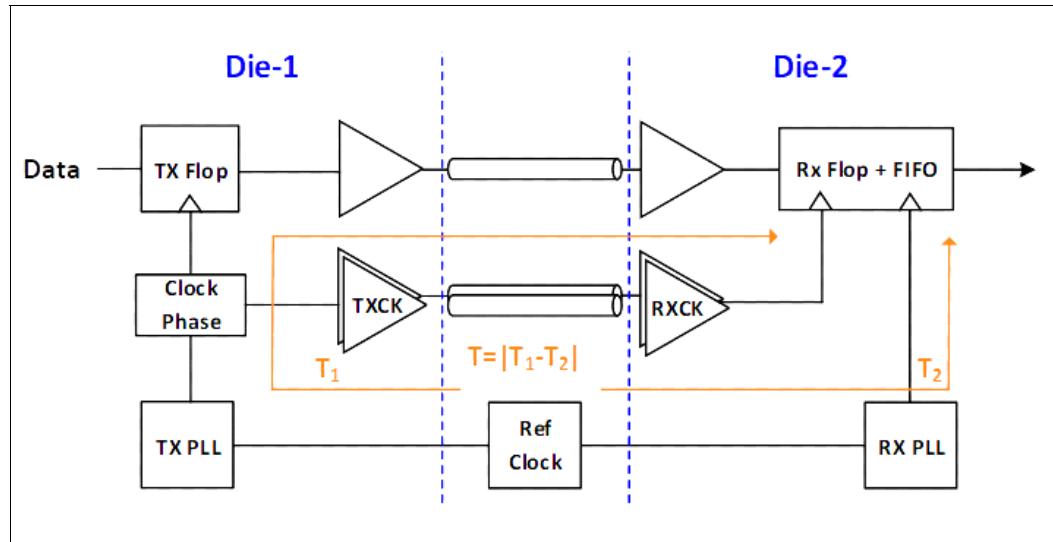
- Support for 4, 8, 12, 16, 24, and 32 GT/s data rates
- Support for Advanced and Standard package interconnects
- Support for clock and power gating mechanisms
- Single-ended unidirectional data signaling
- DC coupled point-to-point interconnect
- Forwarded clock for transmit jitter tracking
- Matched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

5.1 Interoperability

5.1.1 Data rates

A device must support 4 GT/s and all the data rates data rates between 4 GT/s and the highest supported data rate. For example, a device supporting 16 GT/s must also support 4, 8, and 12 GT/s Data rates.

Spread-Spectrum Clocking (SSC) is permitted. Common reference clock (REFCLK) is required between a UCIe Link Transmitter and the corresponding UCIe Link partner's Receiver with a transport delay difference less than 5 ns to limit the FIFO depth and minimize the latency impact. For the retimer use case, the "Local UCIe Link connection" shall use common REFCLK, while the "Off-Package Link connection" is not required to use or share the common REFCLK. [Figure 5-1](#) shows the transport delay difference and is symmetrical for both directions of a Die's UCIe Link connection. The transport delay represents the delay difference between the Transmitter data to the Receiver data latch and the clock as seen at the receiver's FIFO output data latch. See [Section 5.1.2](#) for REFCLK details.

Figure 5-1. Example Common Reference Clock**IMPLEMENTATION NOTE**

In typical implementations, the LCLK for PCIe Link Transmitter and LCLK for the corresponding link partner Receiver, are both generated from the common reference clock. In the example implementation of Figure 5-1, the LCLK for Transmitter in Die-1 can be generated from TX PLL and the LCLK for Receiver in Die-2 can be generated from the RX PLL.

5.1.2 Reference Clock (REFCLK)

Common reference clock (REFCLK) uses a single source that is distributed to both the Transmitter and the Receiver. The clock can be supplied from a package pin or be forwarded by another die on the package. In either case, the reference clock used by both dies on the same link must be from the same clock source. Although other reference clocks are possible, it is recommended that every chiplet use a 100-MHz reference clock, including both dies having different reference clock values from the same clock source. Table 5-1 lists the permitted reference clock frequency range. The minimum and maximum frequencies listed in the table indicate the limits, and do not indicate a requirement to support the entire frequency range. It is required for implementations to generate precise I/O clock frequencies for the supported data rates that use the reference clock. Note that this is possible if the I/O clock frequency is an exact integer multiple of the reference clock frequency (if different from 100 MHz). The reference clock may be disabled in low-power states (such as is done in other Standards and Specifications).

Table 5-1. REFCLK Frequency PPMs and SSC PPMs (Sheet 1 of 2)

Symbol	Description	Limits			Unit	Notes
		Min	Rec	Max		
F _{REFCLK}	REFCLK Frequency	25	100	200	MHz	
F _{REFCLKDEVIATION}	REFCLK Frequency Deviation	-300		300	ppm	Maximum deviation allowed from ideal target frequency.
F _{SSC}	SSC Modulation Frequency	30		33	kHz	

Table 5-1. REFCLK Frequency PPMs and SSC PPMs (Sheet 2 of 2)

Symbol	Description	Limits			Unit	Notes
		Min	Rec	Max		
T _{SSC-FREQ-DEVIATION}	SSC Deviation	-0.5		0	%	Tracks for different frequencies.
T _{TRANSPORT-DELAY}	Tx-to-Rx Transport Delay			5	ns	
T _{SSC-MAX-FREQ-SLEW}	SSC df/dt			1250	ppm/us	

5.2 Overview

5.2.1 Interface Overview

High-level block diagrams of UCIe PHY are shown in [Figure 5-2](#) and [Figure 5-3](#). The UCIe physical interface consists of building blocks called Modules. A Module that uses advanced packaging technology (e.g., EMIB, CoWoS) called “Advanced Package Module” consists of a pair of clocks, 64 or 32 single-ended data Lanes for x64 or x32 Advanced Package Module, respectively, a data valid Lane each direction (transmit and receive) and a Track Lane. There is a low-speed sideband bus for initialization, Link training, and configuration reads/writes. The sideband consists of a single-ended sideband data Lane and single-ended sideband clock Lane in both directions (transmit and receive).

The x16 or x8 “Standard Package Module” uses a traditional Standard packaging with larger pitch. A Standard Package Module consists of a pair of clocks, 16 or 8 single-ended data Lanes, a data valid Lane and Track Lane in each direction (transmit and receive). There is a low-speed sideband bus for initialization, Link training, and configuration reads/writes. The sideband consists of a single-ended sideband data Lane and single-ended sideband clock Lane in both directions (transmit and receive).

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

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

Figure 5-2. x64 or x32 Advanced Package Module

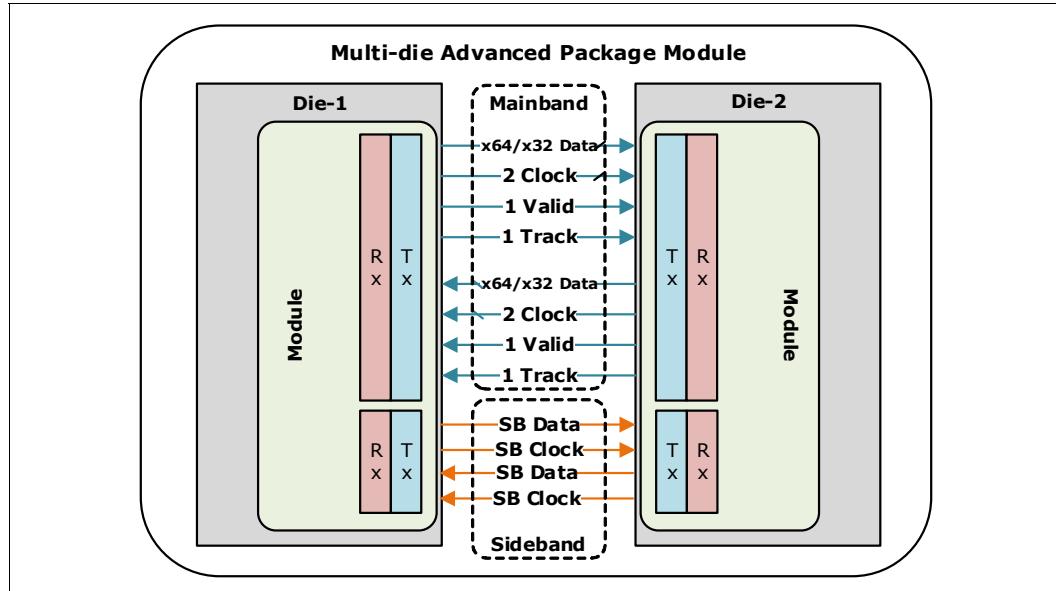
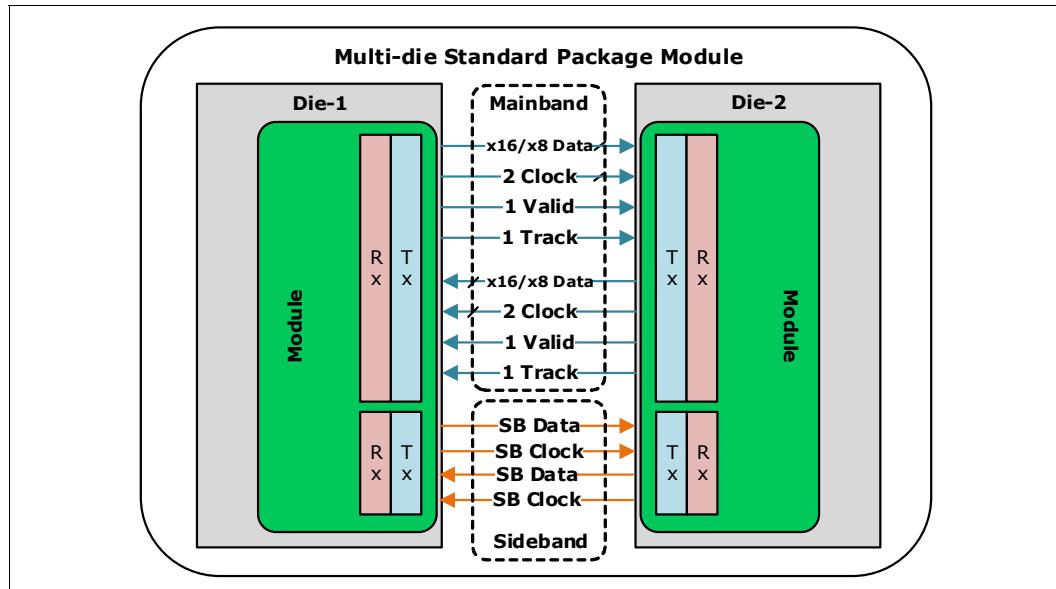


Figure 5-3. x16 or x8 Standard Package Module



5.2.2 Electrical summary

Table 5-2 defines the PHY electrical characteristics of a UCIe device.

Table 5-2. Electrical summary

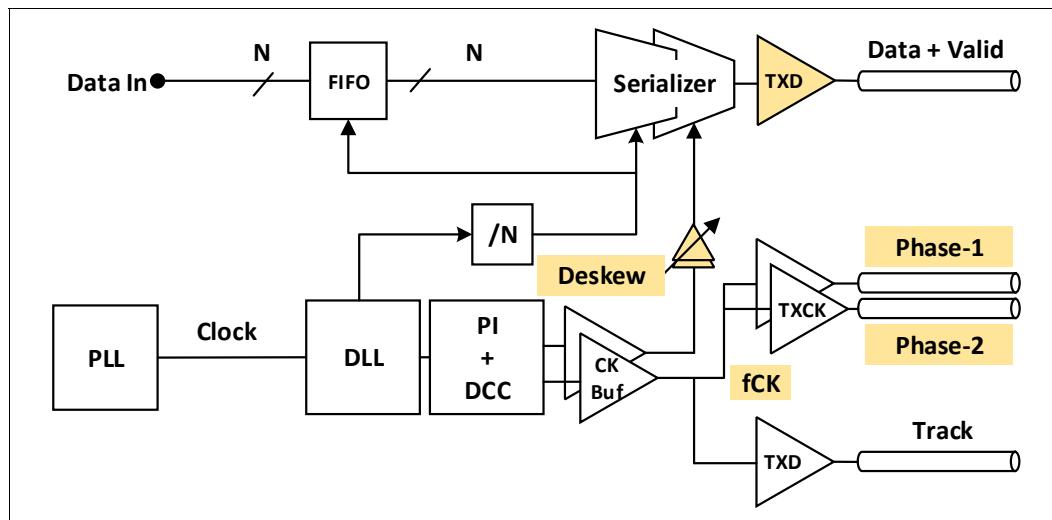
Parameter	Advanced Package (x64)			Standard Package			
Data Width (per module)	64	64	64	16	16	16	16
Data Rate (GT/s)	4/8/12	16	24/32	4-16	4/8/12	16	24/32
Power Efficiency Target (pJ/b)	See Table 1-4						
Latency Target (TX+RX) (UI) ^a (Target upper bound)	12	12	16	12	12	12	16
Idle Exit/Entry Latency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

- a. Electrical PHY latency target. For overall latency target, see Table 1-4.
- b. See Table 1-4.
- c. For compatibility, PHY dimension width must match spec for Advanced Package. Tolerance of PHY dimension width for Standard Package can be higher because there is more routing flexibility. For best channel performance, it's recommended for width to be close to spec.
- d. Standard Package PHY dimension width is the effective width of one (x16) module based on x32 interface (see Figure 5-42 and Figure 5-43).
- e. PHY dimension depth is an informative parameter and depends on bump pitch. Number in the table is based on 45-um bump pitch for 10-column x64 Advanced Package and 100-um bump pitch for Standard Package. See Section 5.7.2 for informative values of PHY dimension depth for combinations of the x64 and x32 Advanced Package modules in 10-column, 16-column, and 8-column bump matrix construction.
- f. Reference (Industry Council on ESD Target Levels): White Paper 2: A Case for Lowering Component-level CDM ESD Specifications and Requirements.

5.3 Transmitter Specification

The Transmitter topology is shown in Figure 5-4. Each data module consists of N single-ended data Transmitters plus a Valid signal. N is 68 (64 Data + 4 Redundant Data) for a x64 Advanced Package Module. N is 34 (32 Data + 2 Redundant Data) for a x32 Advanced Package Module. N is 16 for a x16 Standard Package Module. N is 8 for a x8 Standard Package Module. There is a pair of Transmitters for clocking and a Track signal in each module. The clock rates and phases are discussed in detail in Section 5.5.

Figure 5-4. Transmitter



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

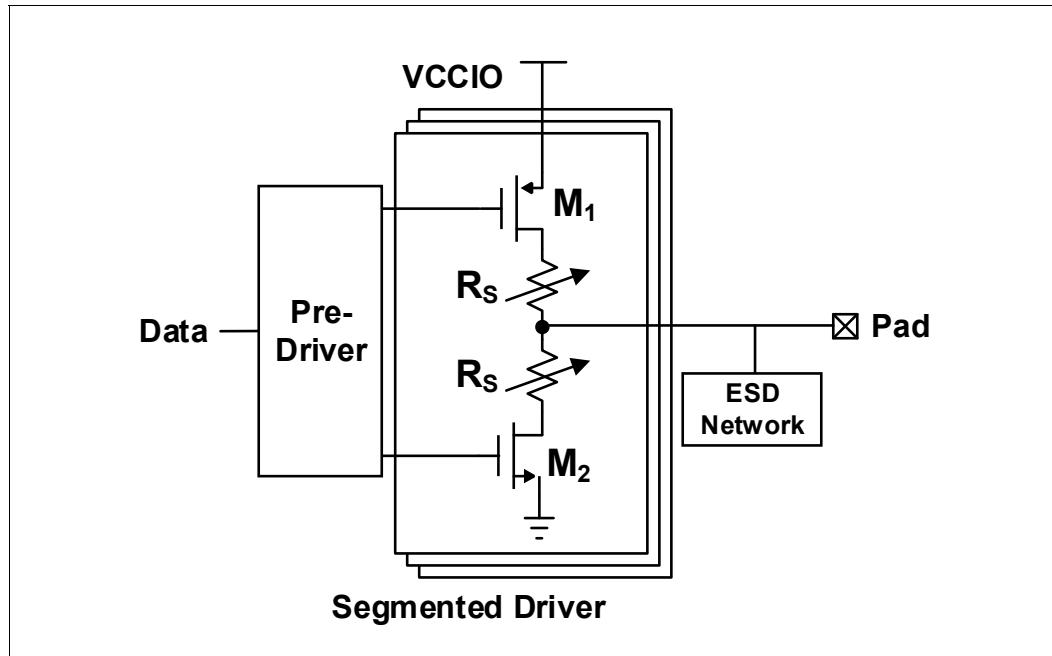
The Track signal can be used for PHY to compensate for slow-changing variables such as voltage or temperature. Track is a unidirectional signal similar to a data bit. The UCIE Module sends a clock pattern (1010...) aligned with Phase-1 of the forwarded clock signal on its Track Transmitter when requested over the sideband by the UCIE Module Partner for its Track Receiver. See [Section 4.6](#) for more details on Runtime Recalibration steps and [Section 5.5.1](#) for Track usage.

5.3.1 Driver Topology

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

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

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

Figure 5-5. Transmitter driver example circuit

5.3.2 Transmitter Electrical parameters

Table 5-3 defines the Transmitter electrical parameters.

Table 5-3. Transmitter Electrical Parameters (Sheet 1 of 2)

Parameter	Min	Typ	Max	Unit
Data Lane TX Swing ^a	0.4			V
Fwd Clock Tx Swing (single ended)	0.4			V
Incoming Clock Rise/Fall time ^b	0.1	0.22	0.25	UI
Incoming Differential Clock Overlap ^b	-	-	30	mUI
Incoming Data Rise/Fall time ^b	-	0.35	-	UI
Driver Pull-up/Pull-down Impedance for Advanced Package ^c	22	25	28	Ohms
Impedance Step Size for Advanced Package ^d			0.5	Ohms
Driver Pull-up/Pull-down Impedance for Standard Package ^e	27	30	33	Ohms
Impedance Step Size for Standard Package ^d	-	-	0.5	Ohms
1-UI Total Jitter ^f	-	-	96/113	mUI pk-pk
1-UI Deterministic Jitter (Dual Dirac) ^g	-	-	48	mUI pk-pk
Tx Data/clock Differential Jitter (Divergent Path) ^h			60	mUI pk-pk
Duty Cycle Error ⁱ	-0.02	-	0.02	UI
Lane-to-Lane Skew Correction Range (up to 16 GT/s) ^j	-0.1	-	0.1	UI
Lane-to-Lane Skew Correction Range (up to 32 GT/s) ^j	-0.15	-	0.15	UI
Lane-to-Lane Skew Correction Range (up to 16 GT/s) ^k	-0.14	-	0.14	UI
Lane-to-Lane Skew Correction Range (up to 32 GT/s) ^k	-0.22	-	0.22	UI
Lane-to-Lane Skew ⁱ	-0.02	-	0.02	UI

Table 5-3. Transmitter Electrical Parameters (Sheet 2 of 2)

Parameter	Min	Typ	Max	Unit
Clock to Mean Data Training Accuracy ^l	-0.07	-	0.07	UI
Phase Adjustment Step ^m	-	-	16	mUI
TX Pad Capacitance (for all speeds) ⁿ	-	-	250	fF
TX Pad Capacitance (8 GT/s capable design) ¹⁴	-	-	300	fF
TX Pad Capacitance (16 GT/s capable design) ^o	-	-	200	fF
TX Pad Capacitance (32 GT/s capable design) ^o	-	-	125	fF

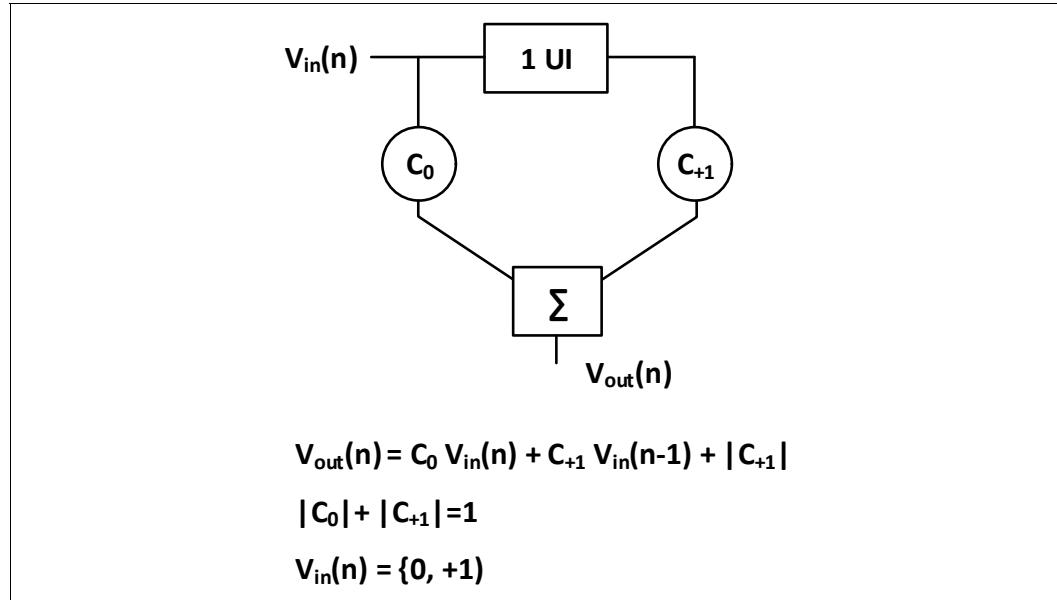
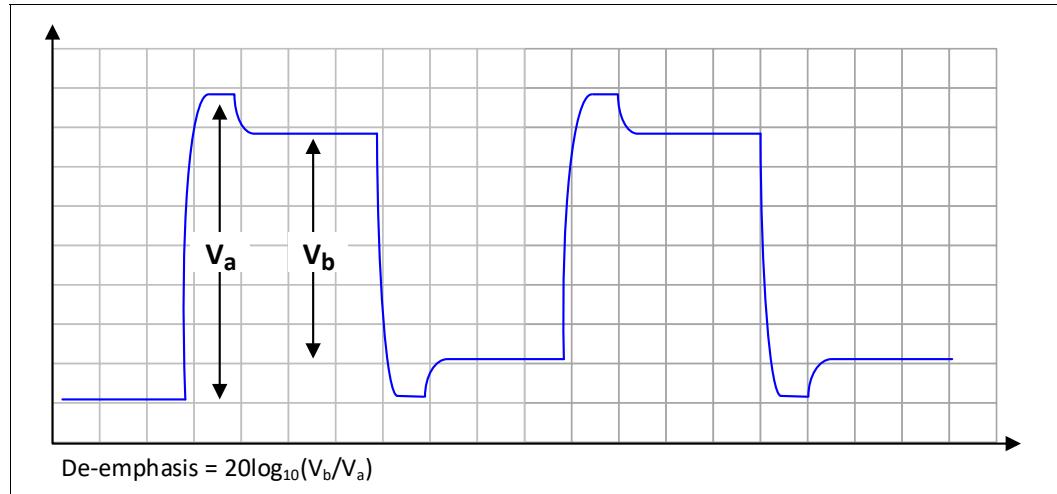
- a. For recommended maximum Transmitter voltage, see [Section 1.5](#).
- b. Expected input (informative). Measured 20% to 80%. Differential clock overlap is deviation from the ideal differential phase (180 degrees apart).
- c. Driver pull-up/down impedance is calibrated at midpoint of Transmitter signal swing.
- d. Impedance step size is an informative parameter and can be implementation specific to meet Driver pull-up/pull-down impedance.
- e. Driver pull-up/pull-down impedance is calibrated at midpoint of Transmitter signal swing (with nominal Rx termination when applicable).
- f. At BER 1E-15/1E-27.
- g. Data dependent jitter excluding Duty Cycle Error.
- h. Includes absolute random jitter and untracked deterministic jitter of the divergent path due to delay mismatch (in the matched architecture).
- i. Post correction.
- j. Advanced Package.
- k. Standard Package.
- l. Includes static and tracking error.
- m. Informative parameter. Phase adjustment step size must be chosen to meet other timing parameters, including Clock-to-Mean Data Training Accuracy, Lane-to-Lane skew, and Duty cycle error (if applicable).
- n. Effective pad capacitance Advanced Package.
- o. Effective pad capacitance Standard Package.

5.3.3 24 GT/s and 32 GT/s Transmitter Equalization

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

Tx equalization coefficients for 24 GT/s and 32 GT/s are based on the FIR filter shown in [Figure 5-6](#). Equalization coefficient is subject to maximum unity swing constraint.

The Transmitter must support the equalization settings shown in [Table 5-4](#). Determination of de-emphasis setting is based on initial configuration or training sequence, where the value with larger eye opening will be selected.

Figure 5-6. Transmitter de-emphasis**Figure 5-7.** Transmitter de-emphasis waveform**Table 5-4.** Transmitter de-emphasis values

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

5.4 Receiver Specification

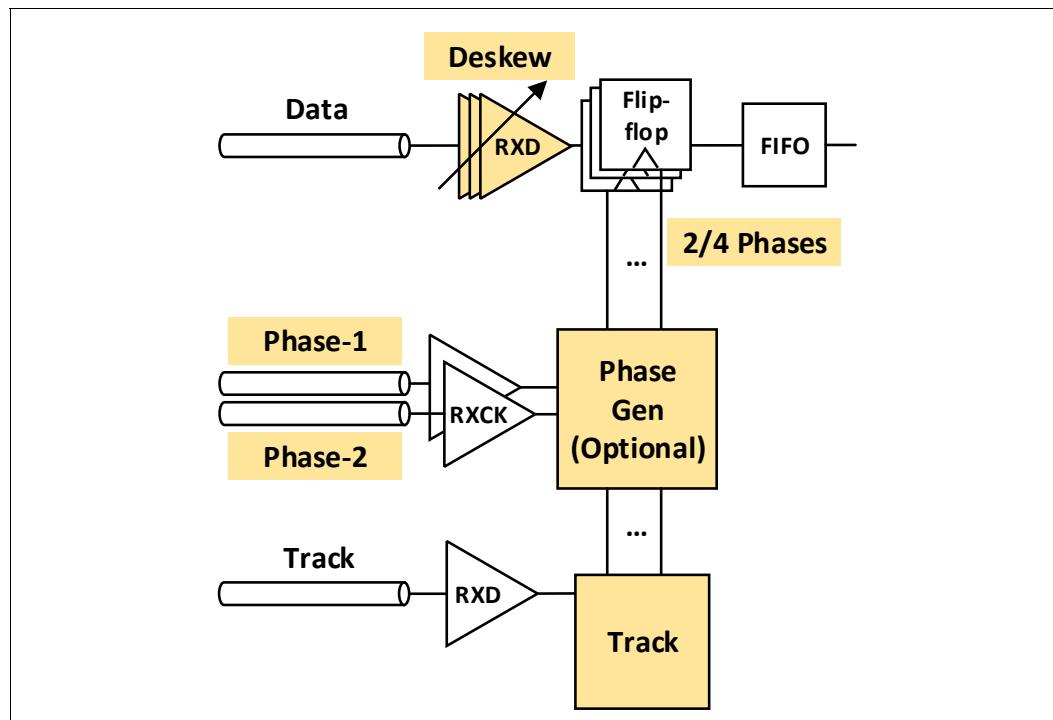
The Receiver topology is illustrated in Figure 5-8. Each Module (Advanced Package and Standard Package) consists of clocks Receivers, data Receivers, and Track Receiver.

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

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

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

Figure 5-8. Receiver topology



5.4.1 Receiver Electrical Parameters

The specified Receiver electrical parameters are shown in [Table 5-5](#).

Table 5-5. Receiver Electrical parameters

Parameter	Min	Typ	Max	Unit
RX Input Impedance ^a	45	50	55	Ohms
Impedance Step Size ^a	-	-	1	Ohms
Data/Clock Total Differential Jitter ^{b c}	-	-	60	mUI pk-pk
Lane-to-Lane skew (up to 16 GT/s) ^d	-0.07	-	0.07	UI
Lane-to Lane skew (> 16 GT/s) ^d	-0.12	-	0.12	UI
Phase error ^e (Including Duty cycle error and in-phase quadrature mismatch)	-0.04	-	0.04	UI
Per-Lane deskew adjustment step ^f	-	-	16	mUI
Output Rise Time ^g	-	-	0.1	UI
Output Fall Time ^g	-	-	0.1	UI
RX Pad Capacitance ^h	-	-	200	fF
RX Pad Capacitance (up to 8 GT/s) ^a	-	-	300	fF
RX Pad Capacitance (up to 16 GT/s) ^{a i}	-	-	200	fF
RX Pad Capacitance (24 and 32 GT/s) ^{a i}	-	-	125	fF
Rx Voltage sensitivity	-	-	40	mV

a. Standard Package mode with termination. Impedance step size is an informative parameter and can be implementation specific to meet Rx Input Impedance.

b. Based on matched architecture.

c. Includes absolute random jitter and untracked deterministic jitter of the divergent path due to delay mismatch (in the matched architecture).

d. Require Rx per-Lane deskew if limit is exceeded.

e. Residual error post training and correction.

f. When applicable (informative).

g. Expected output (informative). Measured 20% to 80%.

h. Advanced Package.

i. Effective Pad capacitance.

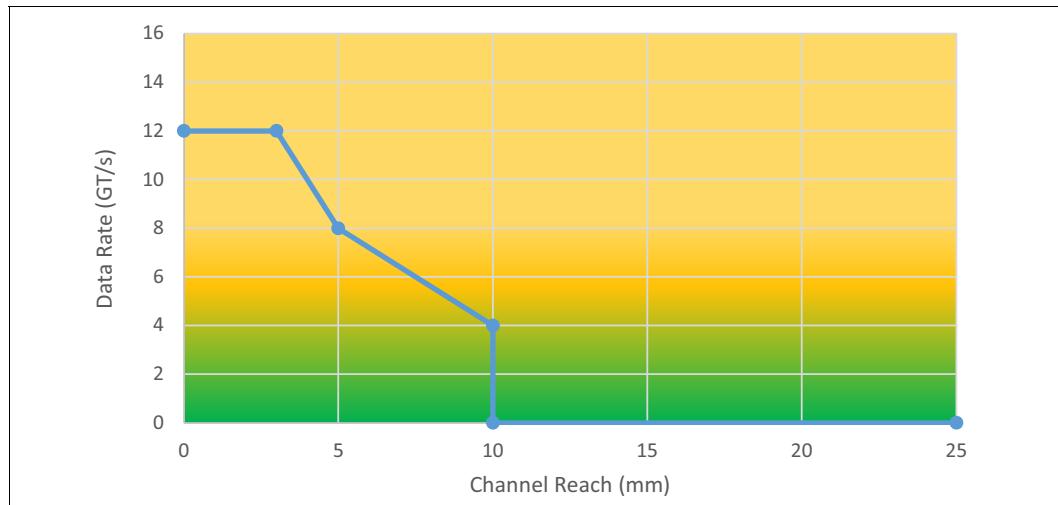
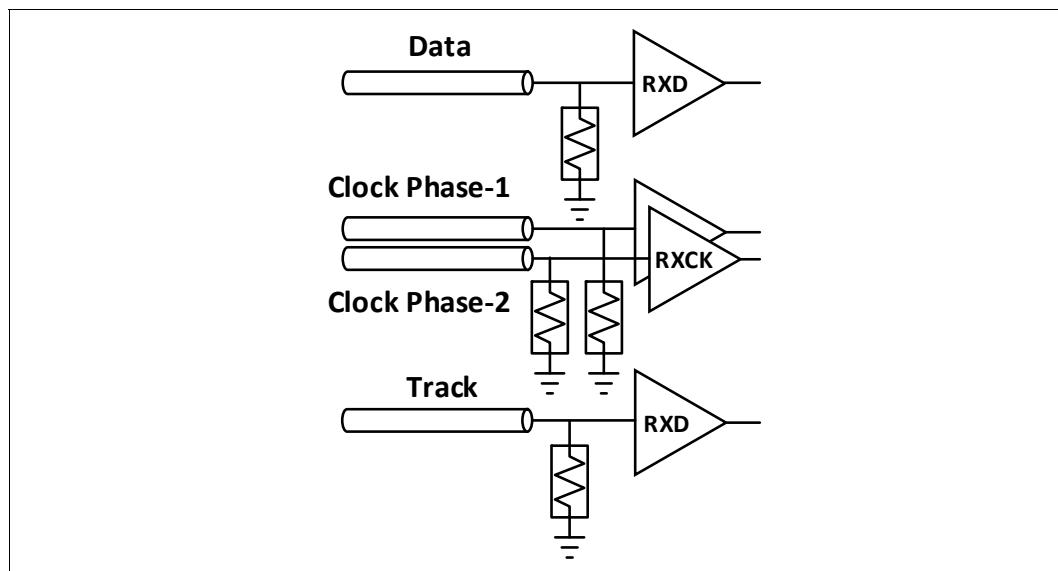
5.4.2 Rx Termination

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

Receiver termination on Standard Package is data rate and channel dependent. [Table 5-6](#) shows the maximum data rate and channel reach combinations for which the Receivers in Standard Package Modules are recommended to remain unterminated for a minimally compliant Transmitter. [Figure 5-9](#) shows an alternate representation of termination requirement. The area below the curve in [Figure 5-9](#) shows the speed and channel-reach combinations for which the Receivers in Standard Package Modules are recommended to remain unterminated. Termination is required for all other combinations. Receivers must be ground-terminated when applicable, as shown in [Figure 5-10](#).

Table 5-6. Maximum channel reach for unterminated Receiver (Tx Swing = 0.4 V)

Data Rate (GT/s)	Channel Reach (mm)
12	3
8	5
4	10

Figure 5-9. Receiver Termination Map for Table 5-6 (Tx Swing = 0.4 V)**Figure 5-10.** Receiver termination

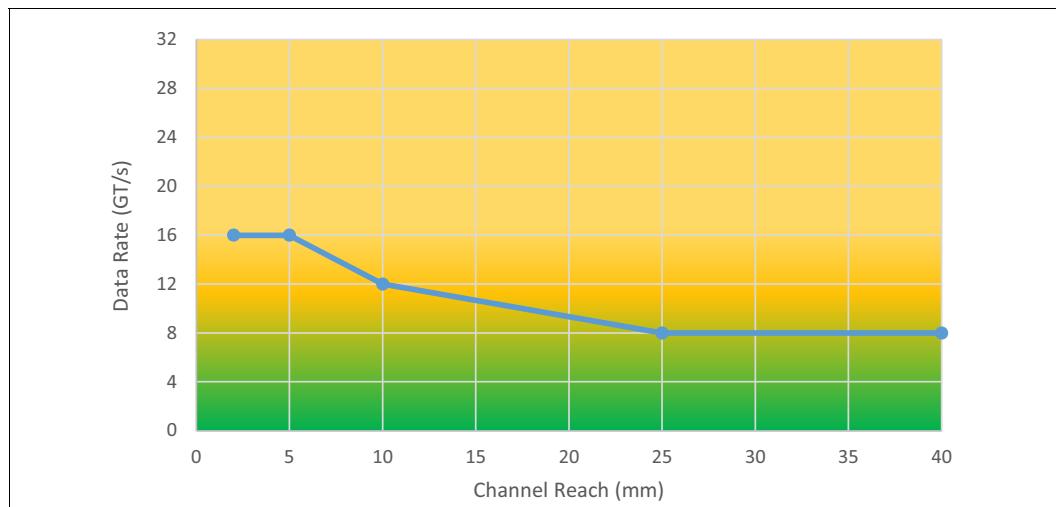
For higher Transmitter swing, unterminated Receiver can be extended to longer channel and high data rate. [Table 5-7](#) shows the maximum data rate and channel reach combinations for Transmitter swing and 0.85 V (maximum recommended swing). [Figure 5-11](#) shows an alternate representation of termination requirement. The area below the curve in [Figure 5-11](#) shows the speed and channel reach

combinations for which the Receivers in Standard Package Modules are recommended to remain unterminated.

Table 5-7. Maximum Channel reach for unterminated Receiver (TX swing = 0.85V)

Data Rate (GT/s)	Channel Reach (mm)
16	5
12	10
8 and below	All supported Lengths

Figure 5-11. Receiver termination map for Table 5-7 (TX Swing = 0.85 V)



IMPLEMENTATION NOTE

When the Transmitter is tri-stated and the Receiver is not required to be enabled (e.g., SBINIT, and some MBINIT states):

- Disabled Receivers must be tolerant of a floating input pad
- Receivers are permitted to enable weak-termination directly on the input pad to prevent crowbar current in the receiver and to lower noise sensitivity at the receiver trip point

When the Transmitter is tri-stated and the Receiver is required to be enabled (e.g., REPAIRCLK and REPAIRVAL states for Advanced Package):

- Enabled Receivers for (CLKP, CLKN, CLKRD, TRK, VLD, VLDRD) must be tolerant of a floating input signal on the pad
- Receivers are permitted to enable weak-termination directly on the input pad to prevent crowbar current in the receiver and to lower noise sensitivity at the receiver trip point

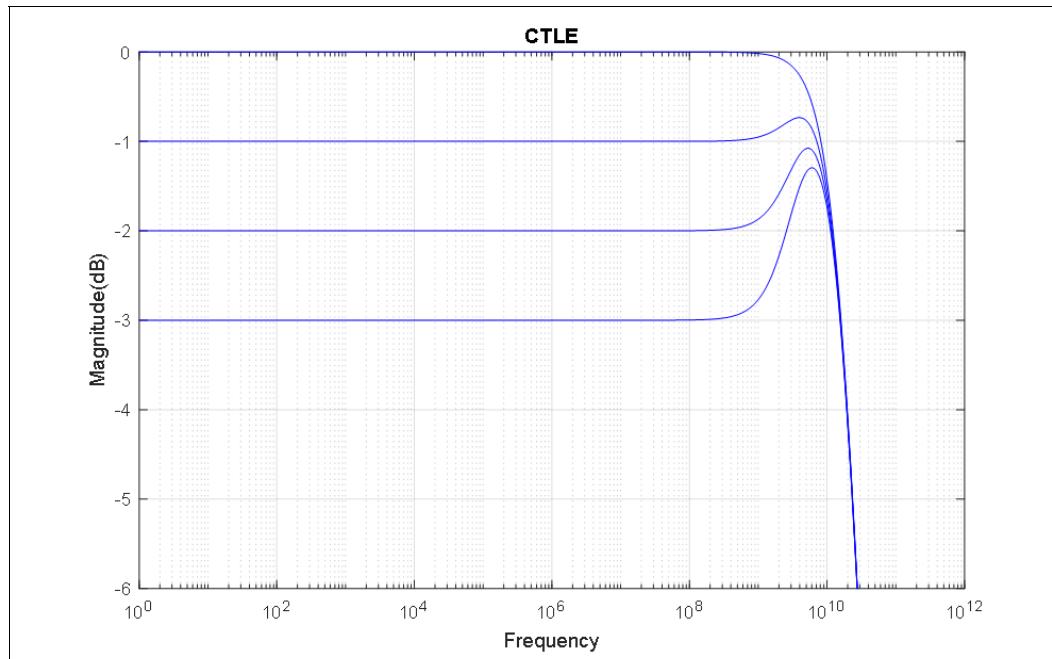
5.4.3 24 and 32 GT/s Receiver Equalization

Receiver equalization may be implemented at 24 GT/s and 32 GT/s data rates. This enables Link operation even when TX equalization is not available. Implementation can be CTLE, inductive peaking, 1-tap DFE, or others. Expected RX equalization capability is equivalent of 1st order CTLE. Example transfer function curves of a first order CTLE are shown in [Figure 5-12](#) and the corresponding equation is shown below:

$$H(s) = \omega_{p2} \left(\frac{s + A_{DC}\omega_{p1}}{(s + \omega_{p1})(s + \omega_{p2})} \right)$$

where, $\omega_{p2} = 2\pi \times \text{DataRate}$, $\omega_{p1} = 2\pi \times \text{DataRate} / 4$, and A_{DC} is the DC gain.

Figure 5-12. Example CTLE



5.5 Clocking

[Figure 5-13](#) shows the forwarded clocking architecture. Each module supports a two-phase forwarded clock. It is critical to maintain matching between all data Lanes and valid signal within the module. The Receiver must provide matched delays between the Receiver clock distribution and Data/Valid Receiver path. This is to minimize the impact of power supply noise-induced jitter on Link performance. Phase adjustment is performed on the Transmitter as shown in [Figure 5-13](#). Link training is required to set the position of phase adjustment to maximize the Link margin.

At higher data rates, Receiver eye margins may be small and any skew between the data Lanes (including Valid) may further degrade Link performance. Per-Lane deskew must be supported on the Transmitter at high data rates.

This specification supports quarter-rate clock frequencies at data rates (24 GT/s and 32 GT/s). The forwarded clock Transmitter must support quadrature phases in addition to differential clock at these data rates (to enable either quarter-rate or half-rate Receiver implementations). [Table 5-8](#) shows the clock frequencies and phases that must be supported at different data rates. Forwarded Clock Phase is negotiated during Link Initialization and Training (see [Section 4.5.3.3.1](#)). At 24 GT/s and 32 GT/s,

Receiver has the options to support differential clock or quadrature clock. The capability register is defined in [Table 9-47](#), and advertised at the beginning of link negotiation. Note that to achieve interoperability with designs of lower max data rate, differential clock must always be used at 16 GT/s and below, independent of the choice at 24 GT/s and 32 GT/s.

5.5.1 Track

Track signal can be used to perform Runtime Recalibration to adjust the Receiver clock path against slow varying voltage, temperature and transistor aging conditions.

When requested by the UCIe Module, the UCIe Module Partner sends a clock pattern (1010...) aligned with Phase-1 of the forwarded clock on its Track Transmitter, as shown in [Figure 5-13](#).

Figure 5-13. Clocking architecture

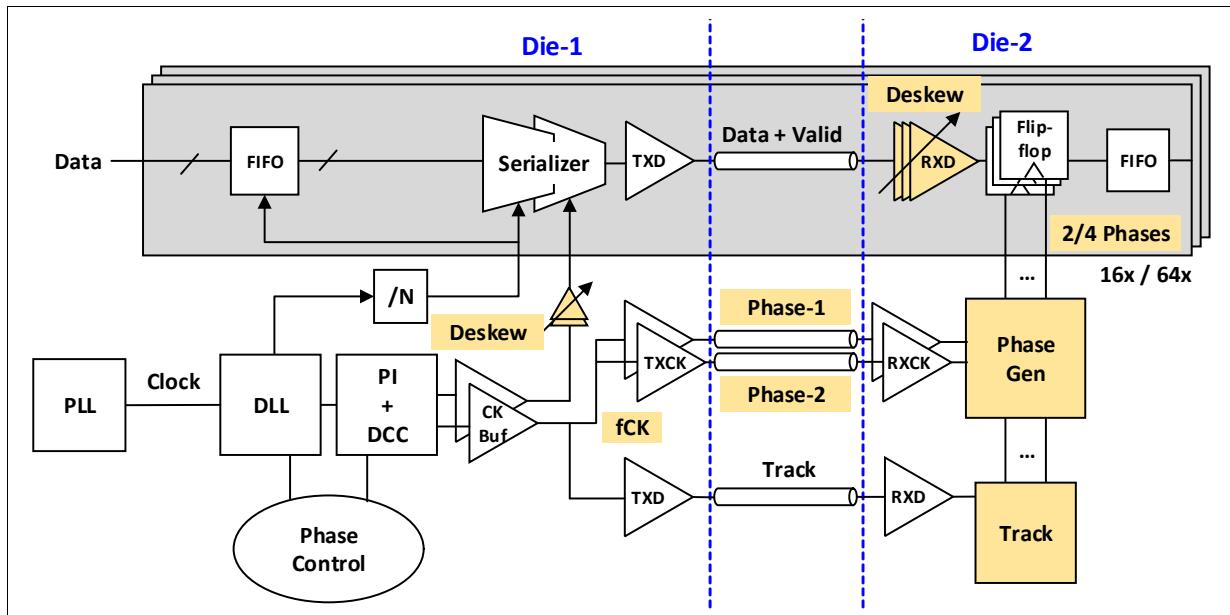


Table 5-8. Forwarded clock frequency and phase

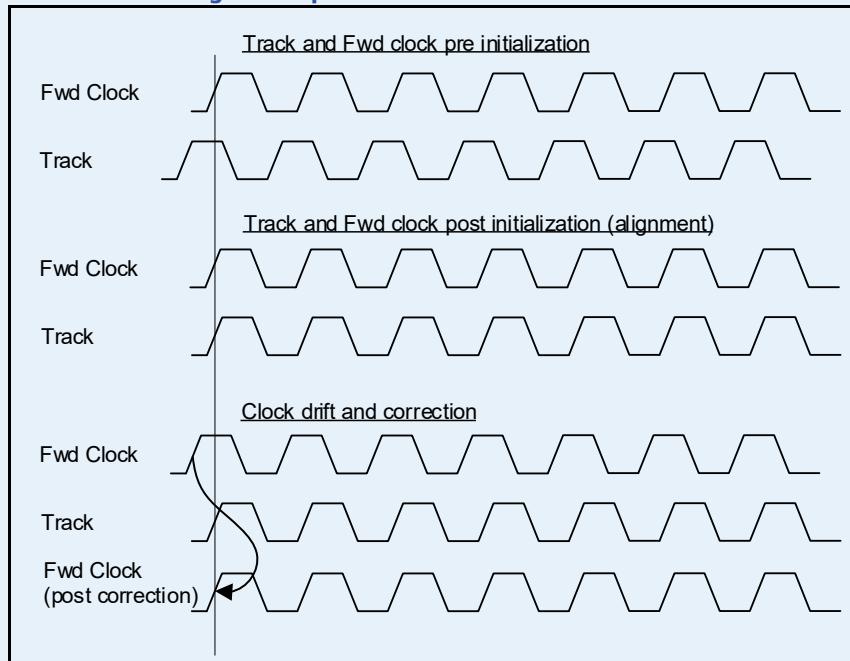
Data rate (GT/s)	Clock freq. (fCK) (GHz)	Phase-1	Phase-2	Deskew (Req/Opt)
32	16	90	270	Required
	8	45	135	Required
24	12	90	270	Required
	6	45	135	Required
16	8	90	270	Required
12	6	90	270	Required
8	4	90	270	Optional
4	2	90	270	Optional

IMPLEMENTATION NOTE

This implementation note provides an example usage for Track signal to calibrate out slow varying temperature- and voltage-related delay drift between Data and Clock on the Receiver.

Track uses the same type of Tx driver and Rx receiver as Data (see [Figure 5-13](#)). A clock pattern aligned with Phase-1 of the forwarded clock is sent from Track Transmitter and received on the Track Receiver. Any initial skew can be calibrated out during initialization and training (MBTRAIN.RXCLKCAL) on the Receiver side. During run-time, any drift between Data and the forwarded clock can be detected. One method for detecting the drift is to sample Track with the forwarded clock. An implementation-specific number of samples can be collected, averaged if needed, and used for drift detection. This drift can then be corrected on the forwarded clock (if needed).

Figure 5-14. Track Usage Example



5.6 Supply noise and clock skew

I/O Vcc noise and the clock skew between data modules shall be within the range specified in Table 5-9.

Table 5-9. I/O Noise and Clock Skew

Parameter	Min	Nom	Max	Unit
I/O Vcc noise for 4 GT/s and 8 GT/s ^a	-	-	80	mVpp
I/O Vcc noise for 12 GT/s ^a	-	-	50	mVpp
I/O Vcc noise for 16 GT/s	-	-	40	mVpp
I/O Vcc noise for 24 GT/s and 32 GT/s ^a	-	-	30	mVpp
Module to module clock skew ^b	-	-	60	ps

- a. I/O VCC noise includes all noise at the I/O supply bumps relative to VSS bumps. This noise includes all DC and AC fluctuations at all applicable frequencies.
- b. Applies only to multi-module instantiations.

IMPLEMENTATION NOTE

Due to different micro bump max current capacity and power delivery requirements, PHY in Advanced Package may have TX providing I/O power supply to RX circuits.

Due to low current draw, sideband supply voltage is strongly recommended to be on an always-on power domain.

5.7 Ball-out and Channel Specification

UCIE interconnect channel needs to meet the requirement of minimum rectangular eye open as specified in [Table 5-10](#) under channel compliance simulation conditions with noiseless and jitter-less behavioral TX and RX models.

Figure 5-15. Example Eye diagram

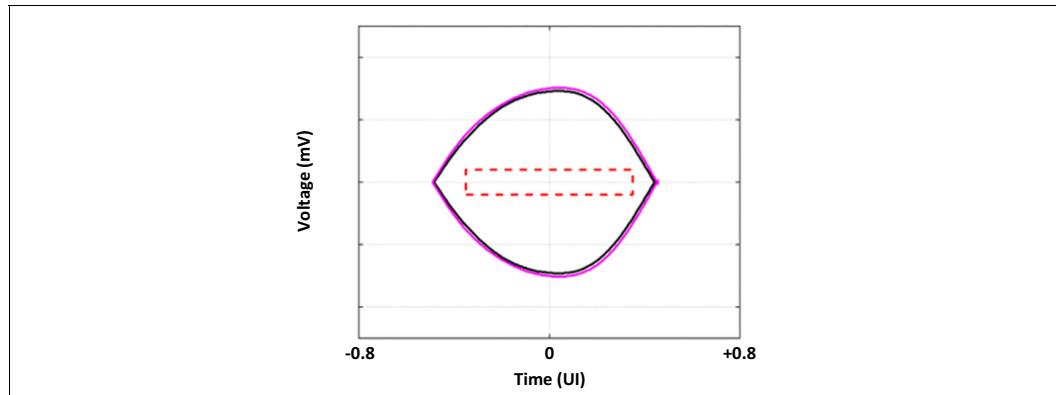


Table 5-10. Eye requirements

Data Rate (GT/s)	Eye Height (mV)	Eye width (UI)
4, 8, 12, 16 ^{a c}	40	0.75
24, 32 ^{a b c}	40	0.65

- a. Rectangular mask.
- b. With equalization enabled.
- c. Based on minimum Tx swing specification.

IMPLEMENTATION NOTE

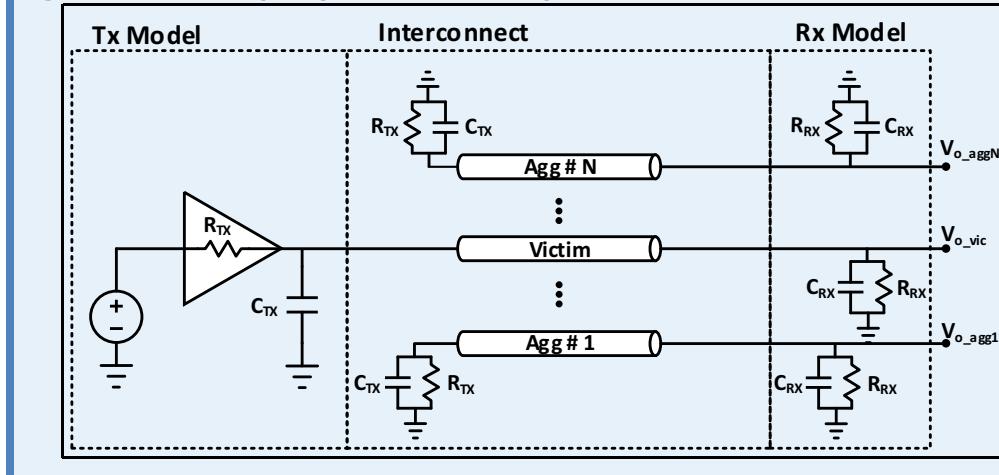
Figure 5-16 shows an example circuit setup that can be used to generate the statistical eye diagram shown in Figure 5-15. RTX is the Transmitter impedance and RRX represents the Receiver termination. CTX, CRX represent effective Transmitter and Receiver capacitance, respectively. For crosstalk, the 19-largest aggressors need to be included. Transmitter equalization (TXEQ) is enabled at 24 GT/s and 32 GT/s.

The eye diagram was generated using a two-step process.

1. Generate ISI and XTALK channel step response using circuit setup shown in Figure 5-16.
2. Use the generated channel response in a signal-integrity or channel-simulation tool to generate a statistical eye diagram (see Figure 5-15)

Other equivalent methods may be used, depending on the signal-integrity or channel-simulation tool.

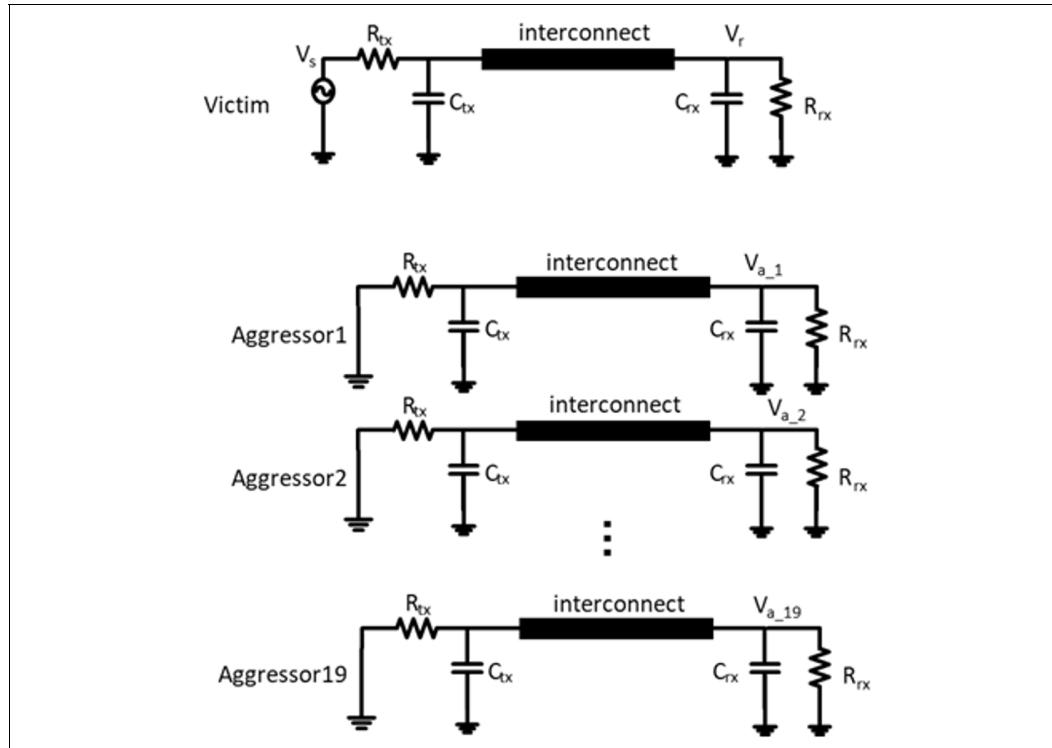
Figure 5-16. Example Eye Simulation Setup



5.7.1 Voltage Transfer Function

Voltage Transfer Function (VTF) based metrics are used to define insertion loss and crosstalk. VTF metrics incorporate both resistive and capacitive components of TX and RX terminations. [Figure 5-17](#) shows the circuit diagram for VTF calculations.

Figure 5-17. Circuit for VTF calculation



VTF loss is defined as the ratio of the Receiver voltage and the Source voltage, as shown in [Equation 5-1](#) and [Equation 5-2](#).

Equation 5-1.

$$L(f) = 20\log_{10}\left|\frac{V_r(f)}{V_s(f)}\right|$$

Equation 5-2.

$$L(0) = 20\log_{10}\left(\frac{R_{rx}}{R_{tx} + R_{channel} + R_{rx}}\right)$$

$L(f)$ is the frequency dependent loss and $L(0)$ is the DC loss. For unterminated channel, $L(0)$ is effectively 0.

VTF crosstalk is defined as the power sum of the ratios of the aggressor Receiver voltage to the source voltage. 19 aggressors are included in the calculation. Based on crosstalk reciprocity, VTF crosstalk can be expressed as shown in [Equation 5-3](#).

Equation 5-3.

$$XT(f) = 10\log_{10} \left(\sum_{i=1}^{19} \left| \frac{V_{ai}(f)}{V_s(f)} \right|^2 \right)$$

5.7.2 Advanced Package**Table 5-11. Channel Characteristics**

Data Rate	4-16 GT/s	24, 32 GT/s
VTF Loss (dB)	$L(f_N) > -3$	$L(f_N) > -5$
VTF Crosstalk (dB) ^a	$XT(f_N) < 1.5 L(f_N) - 21.5$ and $XT(f_N) < -23$	$XT(f_N) < 1.5 L(f_N) - 19$ and $XT(f_N) < -24$

a. Based on Voltage Transfer Function Method (Tx: 25 ohm / 0.25 pF; Rx: 0.2 pF).

f_N is the Nyquist frequency. The equations in the table form a segmented line in the loss-crosstalk coordinate plane, defining the pass/fail region.

Table 5-12. x64 Advanced Package Module Signal List (Sheet 1 of 2)^a

Signal Name	Count	Description
Data		
TXDATA[63:0]	64	Transmit Data
TXVLD	1	Transmit Data Valid; Enables clocking in corresponding module
TXTRK	1	Transmit Track signal
TXCKP	1	Transmit Clock Phase-1
TXCKN	1	Transmit Clock Phase-2
TXCKRD	1	Redundant for Clock and Track Lane repair
TXDATARD[3:0]	4	Redundant for Data Lane repair
TXVLDRD	1	Redundant for Valid
RXDATA[63:0]	64	Receive Data
RXVLD	1	Receive Data Valid; Enables clocking in corresponding module
RXTRK	1	Receive Track.
RXCKP	1	Receive Clock Phase-1
RXCKN	1	Receive Clock Phase-2
RXDATARD[3:0]	4	Redundant for Data Lane repair
RXCKRD	1	Redundant for Clock Lane repair
RXVLDRD	1	Redundant for Valid
Sideband		
TXDATASB	1	Sideband Transmit Data
RXDATASB	1	Sideband Receiver Data
TXCKSB	1	Sideband Transmit Clock
RXCKSB	1	Sideband Receive Clock
TXDATASBRD	1	Redundant Sideband Transmit Data