

Equation 5-3.

$$\omega_p = \min \left(\frac{\omega_n}{\sqrt[4]{1 - A_{DC}^2}}, 2\omega_n \right)$$

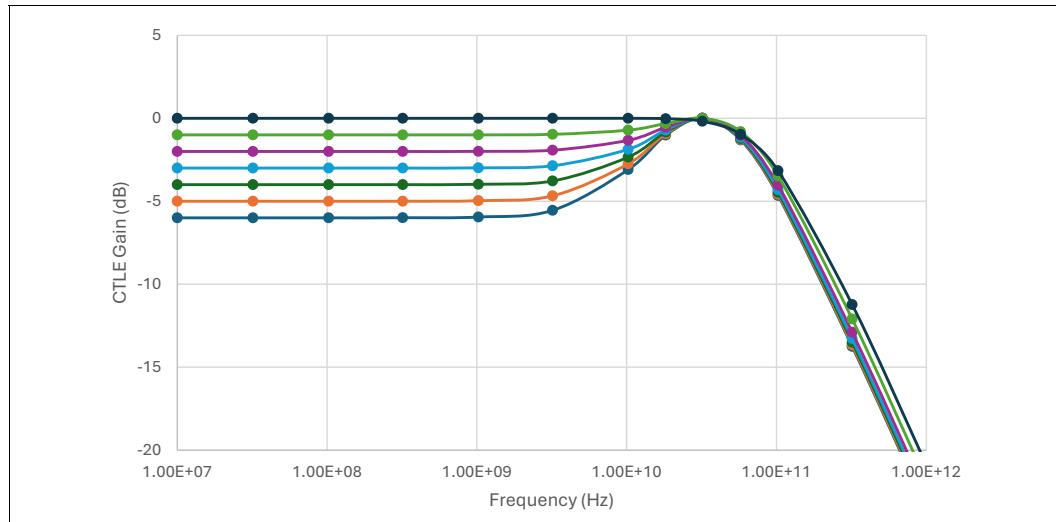
Equation 5-4.

$$\omega_z = \omega_p \sqrt{\frac{1 - \sqrt{1 - A_{DC}^2}}{2}}$$

A_{DC} is the DC gain value. f_n is the peak frequency, 24 GHz and 32 GHz, for 48 GT/s and 64 GT/s, respectively.

It is recommended to provide a minimum of seven CTLE settings to span the DC gain range from -6 dB to 0 dB, adjustable in 1-dB increments. [Figure 5-14](#) shows the Reference Rx CTLE responses for 64 GT/s.

Figure 5-14. Reference Rx CTLE for 64 GT/s



Additionally, designers have the option to implement a 1-tap DFE alongside the CTLE, which can potentially enhance signal integrity at a data rate of 64 GT/s. To maintain control over error propagation in the DFE, it is recommended that the coefficient ratio h_1/h_0 be maintained at less than 0.35.

5.5 Clocking

Figure 5-15 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 5-15](#). 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 5-12](#) 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.

At data rates of 48 GT/s and 64 GT/s, the forwarded clock operates at a quarter rate and is free-running. Additionally, in-phase/quadrature (I/Q) training can be conducted during the Rx clock calibration phase. I/Q training is utilized to fine-tune the relative timing between Phase-1 and Phase-2 of the forwarded clock. It is recommended that the phase adjustment resolution and range adhere to [Table 5-13](#) to ensure sufficient precision and coverage. For additional information, see [Section 4.5.3.4.5](#).

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-like pattern that matches the frequency and is phase-aligned with Phase-1 of the forwarded clock on its Track Transmitter, as shown in [Figure 5-15](#). The pattern is 1010... in the half-rate forwarded clock mode and 1100... in the quarter-rate forwarded clock mode.

Figure 5-15. Clocking architecture

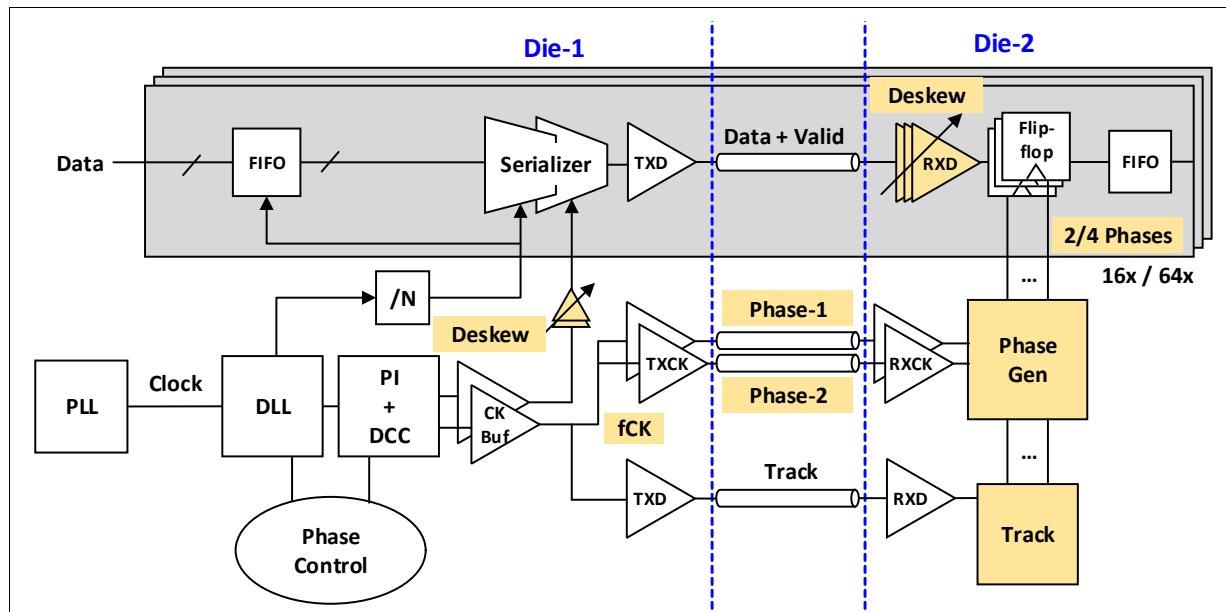


Table 5-12. Forwarded Clock Frequency and Phase

Data rate (GT/s)	Clock freq. (fCK) (GHz)	Phase-1	Phase-2	Deskew (Req/Opt)
64	16	45	135	Required
48	12	45	135	Required
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

Table 5-13. I/Q Correction for 48 GT/s and 64 GT/s

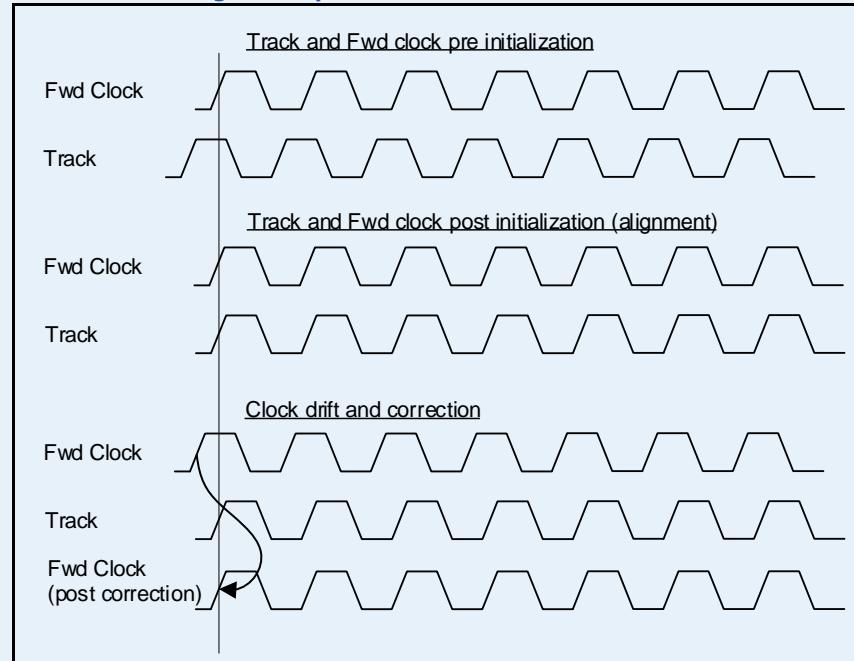
Parameter	Min	Typ	Max	Unit
I/Q Correction Step		1/64		UI
I/Q Correction Range	-12		12	step
	-0.1875		0.1875	UI

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 5-15](#)). 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 5-16. 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 5-14.

Table 5-14. 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, 32, 48, and 64 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

The UCIe interconnect channel is required to achieve a minimum eye opening that conforms to a specified eye mask, under channel compliance simulation conditions with noiseless and jitter-less behavioral Tx and Rx models.

For data rates ≤ 32 GT/s, the eye mask conforms to a rectangular shape (see [Figure 5-17](#)), with the minimum eye height and width specified in [Table 5-15](#).

Figure 5-17. Example Rectangular Eye Mask Diagram for ≤ 32 GT/s

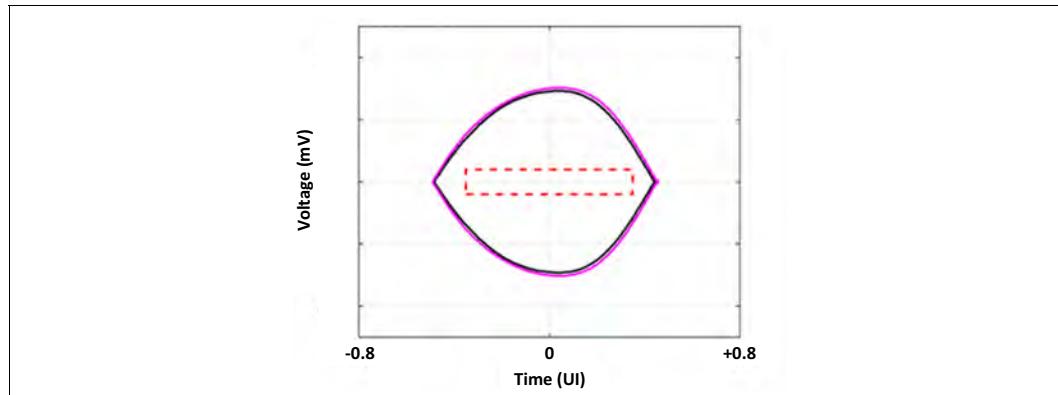
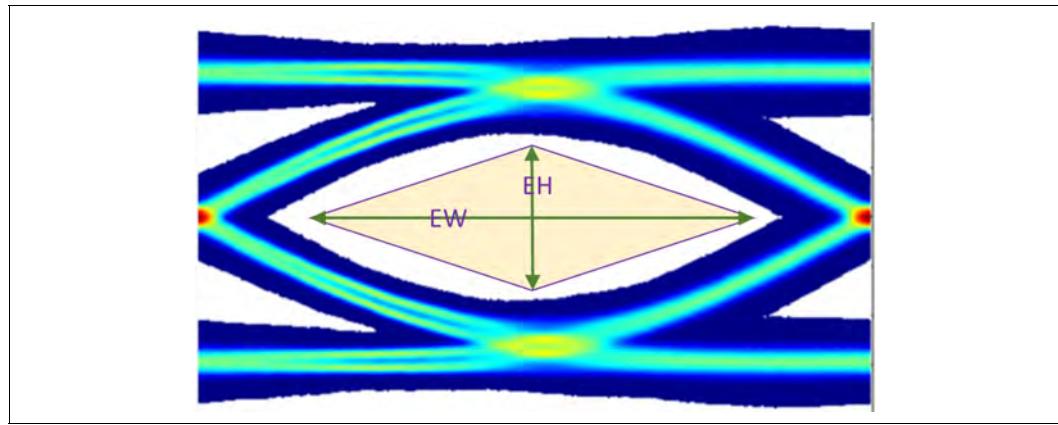


Table 5-15. Rectangular Eye Mask Requirements for <= 32 GT/s

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

a. Based on minimum Tx swing specification.
b. With equalization enabled.

At the higher data rates of 48 GT/s and 64 GT/s, the eye mask adopts a diamond shape (see Figure 5-18). During the channel analysis, the search space for Tx equalization is confined to the presets outlined in Table 5-7. Concurrently, Rx equalization utilizes a 1st order CTLE, as detailed in Section 5.4.4, with a DC gain range from -6 dB to 0 dB, adjustable in 1-dB increments. The minimum eye height and eye width defined for this specification are 65 mV and 0.65 UI, respectively.

Figure 5-18. Diamond Eye Mask for 48 GT/s and 64 GT/s

IMPLEMENTATION NOTE

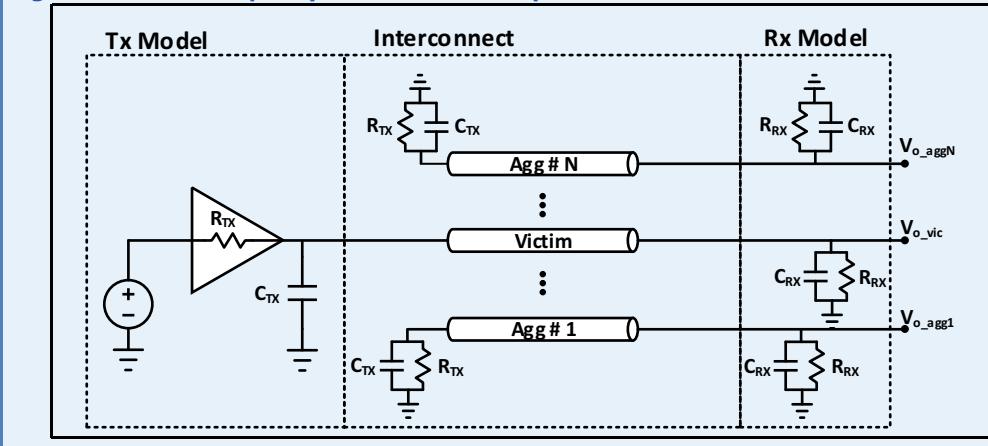
[Figure 5-19](#) shows an example circuit setup that can be used to generate the eye diagrams shown in [Figure 5-17](#) and [Figure 5-18](#). R_{TX} is the Transmitter impedance and R_{RX} represents the Receiver termination. C_{TX} and C_{RX} represent effective Transmitter and Receiver capacitance, respectively. For crosstalk, the 19-largest aggressors need to be included. At data rates of 24 GT/s and above, the corresponding Transmitter and Receiver equalization are enabled.

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

1. Generate ISI and XTALK channel step response using circuit setup shown in [Figure 5-19](#).
2. Use the generated channel response in a signal-integrity or channel-simulation tool to generate a statistical eye diagram (see [Figure 5-17](#)).

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

Figure 5-19. Example Eye Simulation Setup



The Tx Lane-to-Lane Skew Correction Range is the range that the Tx can compensate for interconnect channel mismatches and Rx lane-to-lane delay variations (within the specified limits). Therefore, the difference between Tx Lane-to-Lane Skew Correction Range in [Section 5.3.2](#) and the Rx Lane-to-Lane Skew in [Section 5.4.1](#) and [Section 5.4.2](#) represents interconnect channel-matching tolerance. The tolerance defined with respect to the center of distribution across Lanes is quantified in [Table 5-16](#). The maximum-allowable mismatch between any two Lanes is constrained by the tolerance span listed in the table.

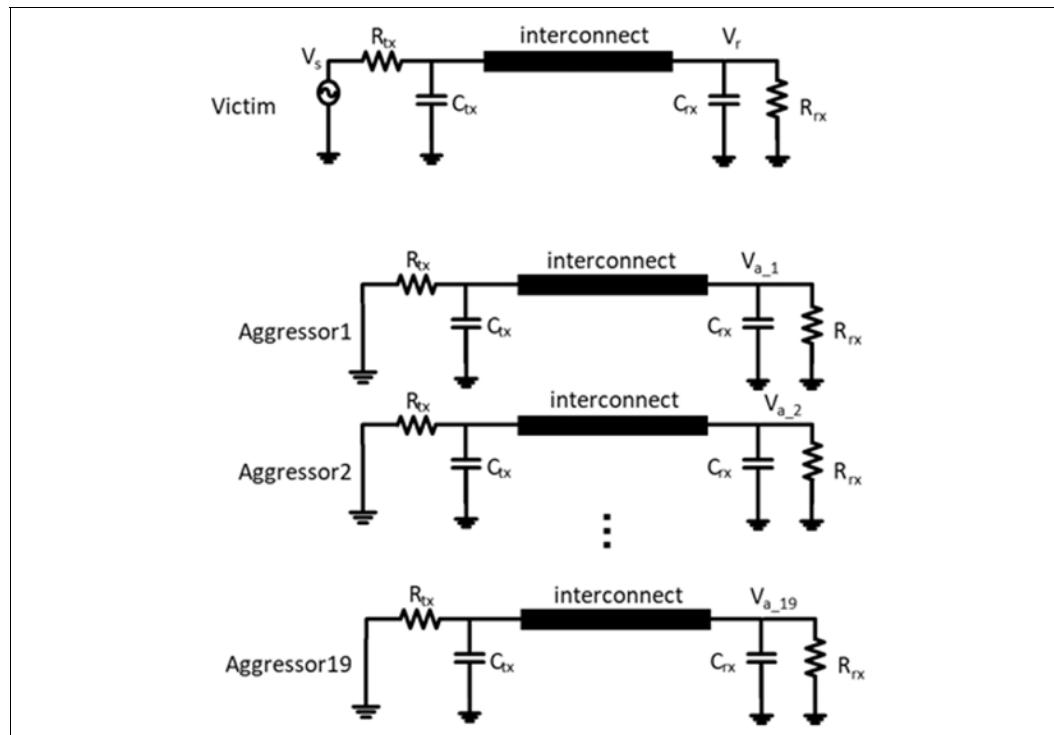
Additionally, the deskew circuit in the Tx can be used to correct Tx clock distribution skew. In this scenario, the combined correction range for Tx, Rx, and channel mismatches depends on the Tx design implementation; however, the combined correction range must exceed the specified Tx Lane-to-Lane Skew Correction Range value listed in [Table 5-5](#).

Table 5-16. Channel Matching Tolerance of Tx or Rx within a Module

Channel Matching Tolerance		Min	Max	Tolerance Span	Unit
Package Type	Data Rate				
Advanced	4 to 32 GT/s	-0.03	0.03	0.06	UI
	48 and 64 GT/s	-0.06	0.06	0.12	UI
Standard	4 to 16 GT/s	-0.07	0.07	0.14	UI
	24 and 32 GT/s	-0.10	0.10	0.20	UI
	48 and 64 GT/s	-0.20	0.20	0.40	UI

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 5-20 shows the circuit diagram for VTF calculations.

Figure 5-20. Circuit for VTF calculation

VTF loss is defined as the ratio of the Receiver voltage and the Source voltage, as shown in [Equation 5-5](#) and [Equation 5-6](#).

Equation 5-5.

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

Equation 5-6.

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

$L(f)$ 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 5-7](#).

Equation 5-7.

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

5.7.2 Advanced Package

Table 5-17. 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 5-18. 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

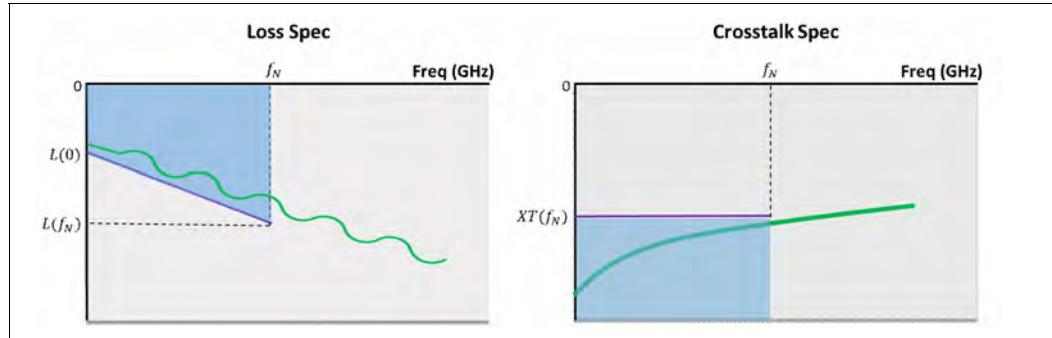
Table 5-18. x64 Advanced Package Module Signal List (Sheet 2 of 2)^a

Signal Name	Count	Description
RXVLID	1	Receive Data Valid; Enables clocking in corresponding module
RXTRK	1	Receive Track.
RXCKP	1	Receive Clock Phase-1
RXCKN	1	Receive Clock Phase-2
RXDATARD[3:0]	4	Redundant for Data Lane repair
RXCKRD	1	Redundant for Clock Lane repair
RXVLDRD	1	Redundant for Valid
Sideband		
TXDATASB	1	Sideband Transmit Data
RXDATASB	1	Sideband Receiver Data
TXCKSB	1	Sideband Transmit Clock
RXCKSB	1	Sideband Receive Clock
TXDATASBRD	1	Redundant Sideband Transmit Data
RXDATASBRD	1	Redundant Sideband Receiver Data
TXCKSBRD	1	Redundant Sideband Transmit Clock
RXCKSBRD	1	Redundant Sideband Receive Clock
Power and Voltage		
VSS		Ground Reference
VCCIO		I/O supply
VCCFWDIO		Forwarded power supply from remote Transmitter supply to local Receiver AFE (see Tightly Coupled mode in Section 5.8)
VCCAON		Always on Aux supply (sideband)

a. For x32 Advanced Package module, the **TXDATA[63:32]**, **TXRD[3:2]**, **RXDATA[63:32]**, and **RXRD[3:2]** signals do not apply. All other signals are the same as the x64 Advanced Package Module signals.

5.7.2.1 Loss and Crosstalk Mask

Loss and crosstalk are specified by a mask defined by the $L(f_N)$ and $XT(f_N)$ at Nyquist frequency. It is a linear mask from DC to f_N for loss and flat mask for crosstalk, illustrated by [Figure 5-21](#). Loss from DC to f_N needs to be above the spec line. Crosstalk from DC to f_N needs to be below the spec line. The green line in [Figure 5-21](#) is a representative passing signal.

Figure 5-21. Loss and Crosstalk Mask

5.7.2.2 x64 Advanced Package Module Bump Map

All bump matrices in this section and hereinafter are defined with “dead bug” view which means the viewer is looking directly at the UCIe micro bumps facing up, with the die flipped like a “dead bug” as illustrated in Figure 5-22.

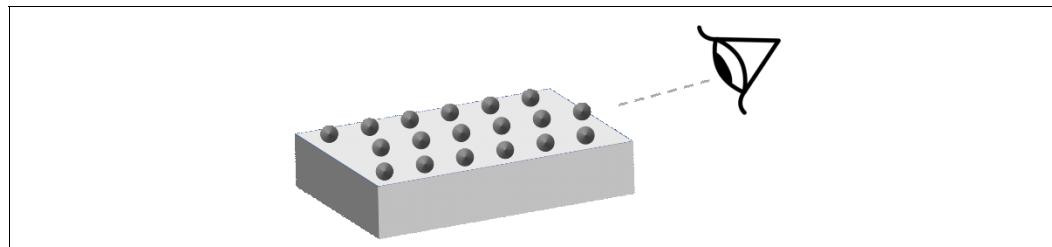
Figure 5-22. Viewer Orientation Looking at the Defined UCIe Bump Matrix

Figure 5-23, Figure 5-24, and Figure 5-25 show the reference bump matrix for the 10-column, 16-column, and 8-column x64 Advanced Package Modules, respectively. The lower left corner of the bump map will be considered “origin” of a bump matrix and the leftmost column is Column 0.

It is strongly recommended to follow the bump matrices provided in Figure 5-23, Figure 5-24, and Figure 5-25 for x64 Advanced Package interfaces.

The 10-column bump matrix is optimal for bump pitch range of 38 to 50 um. To achieve optimal area scaling with different bump pitches, the optional 16-column and 8-column bump matrices are defined for bump ranges of 25 to 37 um and 51 to 55 um, respectively, which will result in optimal Module depth while maintaining Module width of 388.8 um, as shown in Figure 5-24 and Figure 5-25, respectively.

The following rule must be followed for the 10-column x64 Advanced Package bump matrix:

- The signal order within a column must be preserved. For example, Column 0 must contain the signals: **`txdataRD0`, `txdata0`, `txdata1`, `txdata2`, `txdata3`, `txdata4`, ..., `rxdata59`, `rxdata60`, `rxdata61`, `rxdata62`, `rxdata63`, `rxdataRD3`, and `txdatasbRD`**. Similarly, 16-column and 8-column x64 Advanced Packages must preserve the signal order within a column of the respective bump matrices.

It is strongly recommended to follow the supply and **ground** pattern shown in the bump matrices. It must be ensured that sufficient supply and **ground** bumps are provided to meet channel characteristics (FEXT and NEXT) and power-delivery requirements.

The following rules must be followed when instantiating multiple modules of Advanced Package bump matrix:

- Modules must be stepped in the same orientation and abutted.
- Horizontal or vertical mirroring is not permitted.
- Module stacking is not permitted.

Additionally, in multi-module instantiations it is strongly recommended to add one column of **VSS** bumps on each outside edge of the multi-module instantiation.

Mirror die implementation may necessitate a jog or additional metal layers for proper connectivity.

Figure 5-23. 10-column x64 Advanced Package Bump Map for <= 32 GT/s

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
vss		vss	vccio		vccio		vss		vss
vss	vss	vss	vccio	vccio	vccio	vss	vss		
rxcksbRD	rxcksb	rxcksb	vccio	vccio	txcksb	rxdata8b	rxcksbRD		rxdata8bRD
rxdata8bRD	rxdata8b	rxdata8b	vccio	vccio	rxcksb	rxdata14			rxdata8bRD
rxdata8b3	rxdata49	rxdata34	rxdata30	rxdata28	rxdata15	rxdata13			
rxdata63	rxdata51	rxdata36	rxdata33	vccio	rxdata12	rxdata12	vss		
vss	vccio	vss	rxdata31	rxdata27	vss	rxdata11		rxdata0	
rxdata62	rxdata53	rxdata37	rxdataRD1	rxdata26	rxdata16	rxdata10		rxdata1	
rxdata61	rxdata47	rxdata38	vss	rxdata25	rxdata17	rxdata9	vss		
vss	rxdata48	rxdata32	rxckRD	rxdata24	rxdata19	rxdata8		rxdata2	
rxdata60	rxdata53	rxdata37	rxvldRD	rxckn	vss	rxdata7		rxdata3	
rxdata59	rxdata47	rxdata46	rxvld	rxckp	rxdata20		vss		
rxdata58	rxdata45	rxdata39	rxtrk	rxdata23	rxdata21	rxdata6		rxdata4	
vss	rxdata45	vss	rxvldRD	rxckn	rxdata22	rxdata5		vccfwdio	
rxdata60	rxdata56	rxdata44	rxvld	vss	rxdata19	rxdata7			
rxdata59	rxdata57	rxdata40	rxtrk	rxckp	rxdata20		vss		
rxdata58	rxdata43	rxdata46	rxvld	rxdata23	rxdata21	rxdata6			
vss	rxdata45	rxdata41	vss	rxckRD	rxdata22	rxdata5		vccfwdio	
vccio	vccfwdio	vccfwdio	vccio	rxvldRD	rxckn	rxdata19			
vccio	txdata5	txdata21	vccio	rxckRD	rxvldRD	rxckn		txdata58	
txdata4	txdata5	txdata22	vss	rxvldRD	rxckn	rxdata41		vss	
txdata4	txdata6	txdata20	txckp	rxckn	rxdata40	txdata42		txdata58	
vss	txdata6	txdata19	txdata23	txckn	vss	txdata43		txdata59	
txdata3	txdata7	txdata18	vss	txckRD	txvldRD	txdata44		txdata56	
txdata2	txdata8	txdata24	txvld	txvldRD	txdata39	txdata45	vss	txdata60	
vccio	txdata9	txdata17	vccio	txvldRD	txdata38	txdata46		txdata54	
vccio	txdata10	txdata25	vccio	vss	txdata38	txdata46	vccio	txdata61	
txdata1	txdata11	txdata16	txdata26	txdataRD1	txdata37	txdata47	txdata53	txdata62	
txdata0	txdata12	txdata27	txdata27	txdata32	txdata37	txdata48	txdata52	vss	
vss	txdata12	vss	txdata31	txdata33	vss	txdata49	txdata51	txdata63	
txdataRD0	txdata13	txdata28	txdata29	txdata34	txdata36	txdata49	txdata50	txdataRD3	
vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	
vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	
Die Edge									

Note:

In Figure 5-23, at 45-um pitch, the module depth of the 10-column reference bump matrix as shown is approximately 1043 um.

Figure 5-24. 16-column x64 Advanced Package Bump Map for <= 32 GT/s

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11	Column12	Column13	Column14	Column15
vss	vss	vss	vccio	vccio	vccio	vss	vccio	vccio	vccio	vss	vss	vss	vss	vss	vss
vss	vss	rxcksbRD	txdatasbRD	rxcksb	rxdata35	vss	rxcksb	rxdata35	rxdata29	rxdata28	rxdata14	rxdata11	rxdata10	rxdata9	rxdataRDO
vss	rxdata54	rxdata50	rxdata55	rxdata51	rxdata49	rxdata36	rxdata34	rxdataRD1	rxdata30	rxdata27	rxdata15	rxdata13	rxdata12	rxdata11	rxdata8
rxdataRD3	rxdata55	rxdata53	rxdata52	rxdata51	rxdata48	rxdata37	rxdata33	vss	rxdata31	rxdata26	rxdata16	vss	rxdata7	rxdata2	rxdata0
rxdata63	rxdata61	rxdata60	rxdata59	rxdata56	rxdata47	rxdata32	rxdata32	rxckRD	rxdata25	vss	rxdata23	rxdata20	rxdata18	rxdata6	rxdata1
rxdata62	rxdata59	rxdata58	rxdata57	rxdata45	rxdata43	rxdata40	rxdata38	rxvld	rxckn	rxdata22	rxdata19	rxdata19	rxdata18	rxdata4	rxdata3
vccfwdio	vccfwdio	vccfwdio	vccfwdio	vccfwdio	vccfwdio	rxdata41	rxtrk	rxdata24	rxdata21	rxdata21	rxdata5	vccfwdio	vccfwdio	vccfwdio	vccfwdio
vss	vccio	vccio	vccio	vccio	vccio	txdata5	txdata21	txdata24	txtrk	txdata41	vccio	vccio	vccio	vccio	vccio
txdata1	txdata4	txdata6	txdata4	txdata19	txdata19	txdata20	txdata22	txckp	txdata39	txdata42	txdata40	txdata45	txdata58	txdata62	txdata63
txdata0	txdata3	txdata7	txdata2	txdata17	txdata17	txdata23	txdata25	txckn	txvld	txdata38	txdata43	txdata46	txdata57	txdata60	txdata59
vss	txdata2	vss	txdata8	txdata12	txdata16	txdata26	txdata31	txckRD	txdata32	txdata37	txdata48	txdata51	txdata53	txdata61	txdata63
txdataRDO	txdata9	txdata10	txdata11	txdata13	txdata15	txdata28	txdata30	txdataRD1	txdata34	txdata36	txdata49	txdata50	txdata52	txdata54	txdata55
vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio	vccio
Die Edge															

Note:

In [Figure 5-24](#), at 25-um pitch, the module width of the 16-column reference bump matrix as shown is approximately 388.8 um.

Figure 5-25. 8-column x64 Advanced Package Bump Map for <= 32 GT/s

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7
vss		vccio		vccio		vss	
	vss		vccio		vccio		vss
vss		vccio		vccio		vss	
	rxcksbRD		rxcksb	txcksb	rxdatasb		rxdatasbRD
txdatasbRD		txdatasb		txcksb		txcksbRD	
	rxdata50		rxdata36		rxdata14		rxdataRD0
rxdataRD3		rxdata49		rxdata27		rxdata13	
	rxdata51		rxdata35		rxdata15		rxdata0
vss		rxdata48		vss		rxdata12	
	rxdata52		rxdata34		rxdata16		vss
rxdata63		vss		rxdata28		rxdata11	
	rxdata53		rxdata33		rxdata17		rxdata1
rxdata62		rxdata47		rxdata29		rxdata10	
	vccio		vccio		vccio		vccio
rxdata61		rxdata46		rxdata30		vss	
	vss		vss		rxdata26		rxdata2
rxdata60		rxdata37		rxdata31		rxdata9	
	rxdata54		rxdata32		rxdata25		rxdata3
rxdata59		rxdata38		rxdataRD1		rxdata8	
	rxdata55		rxdataRD2		rxdata24		rxdata4
vss		rxdata39		vss		vss	
	rxdata45		vss		rxdata23		rxdata7
rxdata56		rxdata40		rxckRD		rxdata18	
	vss		rxvldRD		rxdata22		vss
rxdata57		rxdata41		rxckn		rxdata19	
	rxdata44		rxvld		vss		rxdata6
rxdata58		vss		rxckp		rxdata20	
	rxdata43		rxtrk		rxdata21		rxdata5
vss		rxdata42		vss		vss	
	vccfwdio		vccfwdio		vccfwdio		vccfwdio
vccio		vccio		vccio		vccio	
	vss		vss		txdata42		vss
txdata5		txdata21		txtrk		txdata43	
	txdata20		txckp		vss		txdata58
txdata6		vss		txvld		txdata44	
	txdata19		txckn		txdata41		txdata57
vss		txdata22		txvldRD		vss	
	txdata18		txckRD		txdata40		txdata56
txdata7		txdata23		vss		txdata45	
	vss		txdata30		txdata39		vss
txdata4		txdata24		txdataRD2		txdata55	
	txdata8		txdataRD1		txdata38		txdata59
txdata3		txdata25		txdata32		txdata54	
	txdata9		txdata31		txdata37		txdata60
txdata2		txdata26		vss		vss	
	vss		txdata30		txdata46		txdata61
vccio		vccio		vccio		vccio	
	txdata10		txdata29		txdata47		txdata62
txdata1		txdata17		txdata33		txdata53	
	txdata11		txdata28		vss		txdata63
vss		txdata16		txdata34		txdata52	
	txdata12		vss		txdata48		vss
txdata0		txdata15		txdata35		txdata51	
	txdata13		txdata27		txdata49		txdataRD3
txdataRD0		txdata14		txdata36		txdata50	
	vccio		vccio		vccio		vccio
vccio		vccio		vccio		vccio	
Die Edge							

Note:

In Figure 5-25, at 55-um pitch, the module depth of the 8-column reference bump matrix as shown is approximately 1,585 um.

Figure 5-26 shows the signal exit order for the 10-column x64 Advanced Package bump map.

Figure 5-26. 10-column x64 Advanced Package Bump map: Signal exit order

		Left to Right																
		txdataR00	txdata0	txdata1	txdata2	txdata3	txdata4	txdata5	txdata6	txdata7	txdata8	txdata9	txdata10	txdata11	txdata12	txdata13	Cont...	
Tx Breakout		txdata14	txdata15	txdata16	txdata17	txdata18	txdata19	txdata20	txdata21	txdata22	txdata23	txdata24	txdata25	txdata26	txdata27	txdata28	Cont1...	
Cont...		txdata29	txdata30	txdata31	txdataRD1	txckRD	txckn	txrk	txvld	txvldRD	txdataRD2	txdata32	txdata33	txdata34	txdata35	Cont2...		
Cont1...		txdata36	txdata37	txdata38	txdata39	txdata40	txdata41	txdata42	txdata43	txdata44	txdata45	txdata46	txdata47	txdata48	txdata49	txdata50	Cont3...	
Cont2...		txdata51	txdata52	txdata53	txdata54	txdata55	txdata56	txdata57	txdata58	txdata59	txdata60	txdata61	txdata62	txdata63	txdataRD3			
		Left to Right																
		rxdataRD3	rxdata63	rxdata62	rxdata61	rxdata60	rxdata59	rxdata58	rxdata57	rxdata56	rxdata55	rxdata54	rxdata53	rxdata52	rxdata51	rxdata50	Cont...	
Rx Breakout		rxdata49	rxdata48	rxdata47	rxdata46	rxdata45	rxdata44	rxdata43	rxdata42	rxdata41	rxdata40	rxdata39	rxdata38	rxdata37	rxdata36	rxdata35	Cont1...	
Cont...		rxdata34	rxdata33	rxdata32	rxdataRD2	rxvldRD	rxvld	rxrk	rxckp	rxckn	rxckRD	rxdataRD1	rxdata31	rxdata30	rxdata29	rxdata28	Cont2...	
Cont1...		rxdata27	rxdata26	rxdata25	rxdata24	rxdata23	rxdata22	rxdata21	rxdata20	rxdata19	rxdata18	rxdata17	rxdata16	rxdata15	rxdata14	rxdata13	Cont3...	
Cont2...		rxdata12	rxdata11	rxdata10	rxdata9	rxdata8	rxdata7	rxdata6	rxdata5	rxdata4	rxdata3	rxdata2	rxdata1	rxdata0	rxdataRD0			

IMPLEMENTATION NOTE

High-speed Considerations for x64 Bump Maps

Three reference bump maps in Figure 5-23, Figure 5-24, and Figure 5-25 are recommended for different ranges of bump pitch, while PHY implementations have the flexibility to adjust the power and ground bumps to meet channel characteristics and power delivery requirements, which largely depend on the target speed and the advanced packaging technology capabilities.

At higher speeds, the PHY circuits draw larger current through the bumps and require better signal and power integrity of the packaging solution. This typically requires adding power and ground bumps and optimizing the distribution of them, but the implementation also needs to minimize the lane-to-lane length skew and preserve the assignment and relative order of the signals in each column to comply with the bump matrix rules in Section 5.7.2.2.

Table 5-19. Bump Map Options and the Recommended Bump Pitch Range and Corresponding Max Speed

Bump Map	Bump Pitch (um)	Max Speed (GT/s)
16 column	25-30	12
	31-37	16
10 column	38-44	24
	45-50	32
8 column	51-55	32

This Implementation Note is formulated to provide PHY implementations a set of reference x64 bump maps to encompass the max speed specified. Table 5-19 summarizes the corresponding max speed for these bump map options and their recommended bump pitch ranges.

Bump maps in Figure 5-27, Figure 5-28, and Figure 5-29 are the x64 implementation references for the corresponding max speed with an enhancement of the power and ground bumps. They all comply with the bump matrix rules in Section 5.7.2.2, and they maintain the backward compatibility in terms of signal exit order. These reference examples have been optimized for signal integrity, power integrity, lane-to-lane skew, electro-migration stress and bump area based on most of the advanced packaging technologies in the industry. Please note that technology requirements vary, and it is still required to verify the bump map with the technology provider for actual implementation requirements and performance targets.

IMPLEMENTATION NOTE

Continued

Figure 5-27. Enhanced 10-column x64 Advanced Package Bump Map Example for 32 GT/s Implementation

	1	2	3	4	5	6	7	8	9	10
1	vss		vss		vccio		vccio		vss	
2		vss		vccio		vccio		vss		vss
3	vss		vss		vccio		vccio		vss	
4		rxcksbRD		rxcksb		vss		rxdatasb		rxdatasbRD
5	txdatasbRD		txdatasb			txcksb		txcksbRD		
6		rxdata50		rxdata35		rxdata29		rxdata14		rxdataRD0
7	rxdataRD3		rxdata49		rxdata34		rxdata28		rxdata13	
8		rxdata51		vccio		vccio		vccio		vccio
9	vccio		vss		rxdata33		vss		rxdata12	
10		rxdata52		rxdata36		rxdata30		rxdata15		vss
11	vss		rxdata48		vss		rxdata27		rxdata11	
12		rxdata53		rxdata37		rxdata31		rxdata16		rxdata0
13	rxdata63		rxdata47		rxdata32		rxdata26		rxdata10	
14		vccio		vccio		vccio		vccio		vccio
15	rxdata62		rxdata46		rxdataRD2		rxdata25		rxdata9	
16		rxdata54		rxdata38		rxdataRD1		rxdata17		rxdata1
17	vss		vss		vss		vss		vss	
18		rxdata55		rxdata39		vccio		rxdata18		rxdata2
19	rxdata61		rxdata45		rxvldRD		rxdata24		rxdata8	
20		vccio		vccio		vccio		vccio		vccio
21	rxdata60		rxdata44		rxvld		rxdata23		rxdata7	
22		rxdata56		rxdata40		rxckRD		rxdata19		rxdata3
23	vss		vss		vss		vss		vss	
24		rxdata57		rxdata41		rxckn		rxdata20		rxdata4
25	rxdata59		rxdata43		rxtrk		rxdata22		rxdata6	
26		rxdata58		vss		rxckp		rxdata21		vss
27	vss		rxdata42		vss		vss		rxdata5	
28		vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio
29	vss		vss		vss		vss		vss	
30		txdata5		vccio		vccio		txdata42		vccio
31	vccio		txdata21		txckp		vccio		txdata58	
32		txdata6		txdata22		txtrk		txdata43		txdata59
33	txdata4		txdata20		txckn		txdata41		txdata57	
34		vccio		vccio		vccio		vccio		vccio
35	txdata3		txdata19		txckRD		txdata40		txdata56	
36		txdata7		txdata23		txvld		txdata44		txdata60
37	vss		vss		vss		vss		vss	
38		txdata8		txdata24		txvldRD		txdata45		txdata61
39	txdata2		txdata18		vss		txdata39		txdata55	
40		vccio		vccio		vccio		vccio		vccio
41	txdata1		txdata17		txdataRD1		txdata38		txdata54	
42		txdata9		txdata25		txdataRD2		txdata46		txdata62
43	vss		vss		vss		vss		vss	
44		txdata10		txdata26		txdata32		txdata47		txdata63
45	txdata0		txdata16		txdata31		txdata37		txdata53	
46		txdata11		txdata27		vccio		txdata48		vccio
47	vccio		txdata15		txdata30		txdata36		txdata52	
48		txdata12		vccio		txdata33		vccio		vss
49	vss		vss		vss		vss		txdata51	
50		txdata13		txdata28		txdata34		txdata49		txdataRD3
51	txdataRD0		txdata14		txdata29		txdata35		txdata50	
52		vccio		vccio		vccio		vccio		vccio
53	vccio		vccio		vccio		vccio		vccio	
					Die Edge					

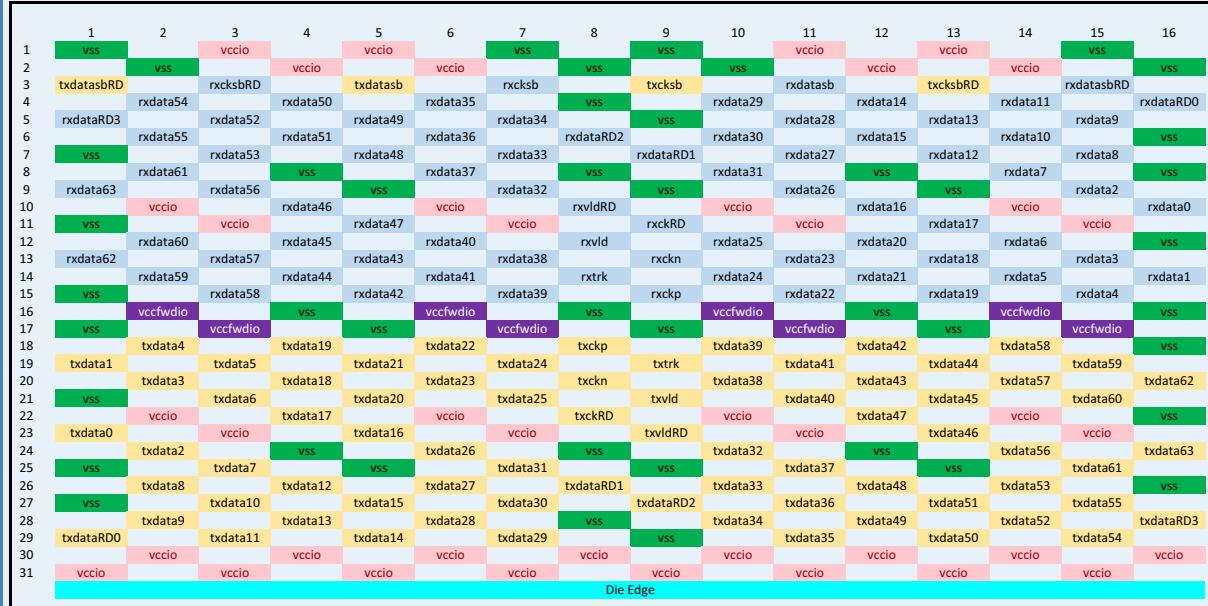
Note:

In Figure 5-27, at 45-um pitch, the module depth of the 10-column bump map as shown is approximately 1225 um. Rows 1, 2, and 53 are required for packaging solutions using floating bridges without through-silicon vias (TSVs). They can be optional for packaging solutions with TSVs.

IMPLEMENTATION NOTE

Continued

Figure 5-28. Enhanced 16-column x64 Advanced Package Bump Map Example for 16 GT/s Implementation



Note:

In Figure 5-28, at 25-um pitch, the module depth of the 16-column bump map as shown is approximately 400 um. Rows 1 and 31 are required for packaging solutions using floating bridges without TSVs. They can be optional for packaging solutions with TSVs.

IMPLEMENTATION NOTE

Continued

Figure 5-29. Enhanced 8-column x64 Advanced Package Bump Map Example for 32 GT/s Implementation

	1	2	3	4	5	6	7	8
1	vss		vccio		vccio		vss	vss
2	vss		vccio		vccio		vss	
3	vss		vccio		vccio		vss	
4		rxcksbRD		rxcksb		rxdatasb		rxdatasbRD
5	txdatasbRD		txdatasb		txcksb		txcksbRD	
6		rxdata50		vss		rxdata14		rxdataRDO
7	rxdataRD3		rxdata49		rxdata27		rxdata13	
8		vccio		rxdata36		vss		vccio
9	rxdata63		rxdata48		rxdata28		rxdata12	
10		rxdata51		rxdata35		rxdata15		rxdata0
11	vss		vss		vss		vss	
12		rxdata52		rxdata34		rxdata16		rxdata1
13	rxdata62		rxdata47		rxdata29		rxdata11	
14		vccio		vccio		vccio		vccio
15	rxdata61		rxdata46		rxdata30		rxdata10	
16		rxdata53		rxdata33		rxdata17		rxdata2
17	rxdata60		rxdata37		rxdata31		rxdata9	
18		rxdata54		rxdata32		rxdata26		rxdata3
19	vss		vss		vss		vss	
20		rxdata55		rxdataRD2		rxdata25		rxdata4
21	rxdata59		rxdata38		rxdataRD1		rxdata8	
22		vccio		vccio		rxdata24		vccio
23	rxdata56		rxdata39		rxvldRD		rxdata18	
24		rxdata45		rxvld		rxdata23		rxdata7
25	vss		rxdata40		vss		vss	
26		vccio		rxvld		rxckRD		rxdata6
27	rxdata57		vss		rxckRD		rxdata19	
28		rxdata44		vccio		vccio		vccio
29	rxdata58		rxdata41		rxtrk		rxdata20	
30		rxdata43		rxtrk		rxckP		rxdata5
31	vss		rxdata42		rxckP		vss	
32		vccfwdio		vccfwdio		vccfwdio		vccfwdio
33	vss		vss		vss		vss	
34		vccio		txckP		txdata42		vccio
35	txdata5		txdata21		txtrk		txdata43	
36		txdata20		txckN		txdata41		txdata58
37	vss		vss		vss		txdata44	
38		txdata19		txckRD		vccio		txdata57
39	txdata6		txdata22		txvld		vss	
40		vccio		vccio		txdata40		vccio
41	txdata7		txdata23		txvldRD		txdata45	
42		txdata18		vss		txdata39		txdata56
43	vss		txdata24		vss		vss	
44		txdata8		txdataRD1		txdata38		txdata59
45	txdata4		txdata25		txdataRD2		txdata55	
46		vccio		vccio		vccio		vccio
47	txdata3		txdata26		txdata32		txdata54	
48		txdata9		txdata31		txdata37		txdata60
49	txdata2		txdata17		txdata33		txdata53	
50		txdata10		txdata30		txdata46		txdata61
51	vss		vss		vss		vss	
52		txdata11		txdata29		txdata47		txdata62
53	txdata1		txdata16		txdata34		txdata52	
54		vccio		vccio		vccio		vccio
55	txdata0		txdata15		txdata35		txdata51	
56		txdata12		txdata28		txdata48		txdata63
57	vss		vss		txdata36		vss	
58		txdata13		txdata27		txdata49		txdataRD3
59	txdataRDO		txdata14		vss		txdata50	
60		vccio		vccio		vccio		vccio
61	vccio		vccio		vccio		vccio	
						Die Edge		

Note:

In Figure 5-29, at 55-um pitch, the module depth of the 8-column bump map as shown is approximately 1705 um. Rows 1, 2, and 61 are required for packaging solutions using floating bridges without TSVs. They can be optional for packaging solutions with TSVs.

5.7.2.2.1 x64 Advanced Package Module Bump Map for 48 GT/s and 64 GT/s

Figure 5-30 illustrates bump configurations for Advanced Package modules that operate at data rates of 48 GT/s and 64 GT/s. In comparison to the configuration for 32 GT/s, additional power and ground bumps have been incorporated. This is designed to mitigate the impact of increased crosstalk at higher data rates and to accommodate the higher total current. It should be noted that additional bump maps or further optimizations may be introduced in subsequent updates to this specification.

Figure 5-30. 10-column x64 Advanced Package Bump Map for 48 GT/s and 64 GT/s

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10
vss	vss	vss	vccio	vccio	vccio	vss	vss	vss	vss
	vss		vccio	vccio	vccio	vss		vss	vss
vss		vss	vccio	vccio	vccio	vss	vss	vss	vss
rxcksbRD	rxcksbRD	rxcksb	vccio	vccio	txcksb	rxdataSB	txcksbRD	rxdataSB	rxdataSB
	vss		vccio	vccio	vccio	vccio	vss	vss	vss
vss		vss	vss	vss	vss	vss	vss	vss	vss
	rxdata50		rxdata35		rxdata29		rxdata14		vss
rxdataRD3		rxdata49		rxdata34		rxdata28		rxdata13	
	vccio		vccio		vccio		vccio		vccio
vss		vss	vss	vss	vss	vss		rxdata12	
	rxdata51		rxdata36		rxdata30		rxdata15		rxdataRD0
vss		rxdata48		rxdata33		rxdata27		vss	
	rxdata52		vccio		vccio		vccio		vccio
rxdata63		vss	vss	vss	vss	vss		rxdata11	
	vccio		rxdata37		rxdata31		rxdata16		vss
vss		rxdata47		rxdata32		rxdata26		rxdata10	
	rxdata53		vccio		vccio		vccio		rxdata0
rxdata62		vss	vss	vss	vss	vss	vss		
	vccio		rxdata38		rxdataRD1		rxdata17		vccio
vss		rxdata46		rxvldRD		rxdata25		rxdata9	
	rxdata54		vccio		vccio		vccio		rxdata1
rxdata61		vss	vss	vss	vss	vss	vss		
	vccio		rxdata39		vccio		rxdata18		vccio
vss		rxdata45		rxvldRD		rxdata24		rxdata8	
	rxdata55		vccio		vccio		vccio		rxdata2
rxdata60		vss		rxvld		vss		vss	
	vccio		rxdata40		vccio		rxdata19		vss
vccio		rxdata44		vss		rxdata23		rxdata7	
	rxdata56		vccio		rxckRD		vccio		rxdata3
vss		vss		rxtrk		vss		rxdata6	
	rxdata57		rxdata41		rxckn		rxdata20		vccio
rxdata59		rxdata43		vss		vss		vss	
	vccio		vccio		rxckp		vccio		rxdata4
vss		rxdata42		vss		rxdata22		rxdata5	
	rxdata58		vss		vss		rxdata21		vss
vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	
vss		vccfwdio		vccfwdio		vccfwdio		vccfwdio	
	txdata21			vss		vss		txdata58	
txdata5		txdata22			vss		txdata42		vccio
txdata4		vss		txckp		vss		vss	
	vccio		vccio		vccio		txdata43		txdata59
vss		txdata20		txckn		txdata41		txdata57	
	txdata6		vss		txtrk		vccio		vss
txdata3		vss		txckRD		vss		txdata56	
	txdata7		txdata23		vccio		txdata44		vccio
vss		txdata19		vss		txdata40		vss	
	vccio		vccio		txvld		vccio		txdata60
txdata2		vss		vss		vss		txdata55	
	txdata8		txdata24		txvldRD		txdata45		vccio
vss		txdata18		vss		txdata39		vss	
	vccio		vccio		vccio		vccio		txdata61
txdata1		vss		vss		vss		txdata54	
	txdata9		txdata25		txdataRD2		txdata46		vccio
vss		txdata17		txdataRD1		txdata38		vss	
	vccio		vccio		vccio		vccio		txdata62
txdata0		vss		vss		vss		txdata53	
	txdata10		txdata26		txdata32		txdata47		vccio
vss		txdata16		txdata31		txdata37		vss	
	txdata11		vccio		vccio		vccio		txdata63
vss		vss		vss		vss		txdata52	
	vccio		txdata27		txdata33		txdata48		vss
txdataRD0		txdata15		txdata30		txdata36		txdata51	
	txdata12		vccio		vccio		vccio		vccio
vss		vss		vss		vss		vss	
	txdata13		txdata28		txdata34		txdata49		txdataRD3
vss		txdata14		txdata29		txdata35		txdata50	
	vccio		vss		vss		vccio		vccio
vccio		vccio		vccio		vccio		vccio	
Die Edge									

5.7.2.3 x32 Advanced Package Module Bump Map

UCIe also defines a x32 Advanced Package Module that supports 32 Tx and 32 Rx data signals and two redundant bumps each for Tx and two for Rx (total of four) for lane-repair functions. All other signals, including the sidebands, are the same as those of the x64 Advanced Package.

[Figure 5-31](#), [Figure 5-32](#), and [Figure 5-33](#) show the reference bump matrix for the 10-column, 16-column, and 8-column x32 Advanced Package Modules, respectively. The lower left corner of the bump map will be considered “origin” of a bump matrix and the leftmost column is Column 0.

It is strongly recommended to follow the bump matrices provided in [Figure 5-31](#), [Figure 5-32](#), and [Figure 5-33](#) for x32 Advanced Package Modules.

The following rule must be followed for the 10-column x32 Advanced Package bump matrix:

- The signals order within a column must be preserved. For example, Column 0 must contain the signals: **txdataRD0**, **txdata0**, **txdata1**, **txdata2**, **txdata3**, **txdata4**, and **txdatasbRD**. Similarly, 16-column and 8-column x32 Advanced Packages must preserve the signal order within a column of the respective bump matrices.

It is strongly recommended to follow the supply and **ground** pattern shown in the bump matrices. It must be ensured that sufficient supply and **ground** bumps are provided to meet channel characteristics (FEXT and NEXT) and power-delivery requirements.

When instantiating multiple x32 Advanced Package Modules, the same rules as defined in [Section 5.7.2.2](#) must be followed.

Figure 5-31. 10-column x32 Advanced Package Bump Map for <= 32 GT/s

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
vss		vss		vccio		vccio		vss	
	vss		vccio		vccio				vss
vss		vss		vccio		vccio		vss	
	rxcksbRD		rxcksb		vccio		rxdatasb		rxdatasbRD
txdatasbRD		txdatasb		vss		txcksb		txcksbRD	
	vss		txdata22		rxdata31		vccio		vccio
vss		txdata21		txckp		rxdata30		rxdata13	
	txdata5		txdata23		vss		rxdata14		vccio
vccio		txdata20		txckn		rxdata29		rxdata12	
	txdata6		vss		rxckRD		rxdata15		rxdataRDO
txdata4		vss		txckRD		rxdata28		rxdata11	
	txdata7		txdata24		rxvldRD		vss		vss
vss		txdata19		txtrk		rxdata27		rxdata10	
	txdata8		txdata25		rxvld		rxdata16		rxdata0
txdata3		txdata18		vss		rxdata26		vss	
	vss		txdata26		vss		rxdata17		rxdata1
txdata2		txdata17		txvld		rxdata25		rxdata9	
	txdata9		vss		rxtrk		rxdata18		vss
vccio		vccio		vccio		vccfwdio		vccfwdio	
	txdata10		txdata27		rxckRD		rxdata19		rxdata2
txdata1		txdata16		txvldRD		rxdata24		rxdata8	
	txdata11		txdata28		rxckn		rxdata20		rxdata3
txdata0		vss		vss		vss		rxdata7	
	txdata12		txdata29		rxckp		vss		vss
vss		txdata15		txdataRD1		rxdata23		rxdata6	
	txdata13		txdata30		vss		rxdata21		rxdata4
txdataRDO		txdata14		txdata31		rxdata22		rxdata5	
	vccio		vccio		vccfwdio		vccfwdio		vccfwdio
vccio		vccio		vccio		vccfwdio		vccfwdio	

Note: In Figure 5-31, at 45- μm pitch, the module depth of the 10-column reference bump matrix as shown is approximately 680.5 μm .

Figure 5-32. 16-column x32 Advanced Package Bump Map for <= 32 GT/s

Note: In Figure 5-32, at 25- μm pitch, the module depth of the 16-column reference bump matrix as shown is approximately 237.5 μm .

Figure 5-33. 8-column x32 Advanced Package Bump Map for <= 32 GT/s

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7
vss	vss	vss	vss	vss	vss	vss	vss
vccio		vccio	vccio		vccio	vccio	vccio
rxcksbRD	rxcksb	rxdatasb	rxdata30	txcksb	rxdata13	rxdata13	rxcksbRD
txdata5	txdata23	txdata30	rxdata14	rxdata14	rxdata13	rxdata13	txcksbRD
txdata6	txdata22	txckp	rxdata14	rxdata15	rxdata12	rxdata12	rxdataRDO
txdata7	txdata21	txckn	rxdata31	rxdata15	rxdata11	rxdata11	vss
vss	txdata20	vss	rxdata29	vss	rxdata10	rxdata10	rxdata0
txdata4	txdata25	txckRD	rxdataRD1	rxdata16	rxdata9	rxdata9	rxdata1
txdata3	txdata19	vss	rxvldRD	rxdata29	rxdata8	rxdata8	vss
vss	txdata18	txtrk	rxvld	rxdata28	rxdata2	rxdata2	rxdata2
txdata2	txdata27	txvld	rxvld	rxdata27	rxdata3	rxdata3	vss
vccio	txdata9	txvldRD	rxvld	rxdata26	rxdata17	rxdata17	vss
txdata10	txdata28	rxtrk	rxvld	rxdata26	rxdata17	rxdata17	rxdata4
vss	txdata10	vss	rxckRD	vss	rxdata18	rxdata18	rxdata7
txdata11	txdata29	txdataRD1	rxdata25	rxdata25	rxdata19	rxdata19	vss
txdata1	txdata16	vss	rxdata24	rxdata24	rxdata20	rxdata20	rxdata7
txdata0	txdata15	txdata31	rxckn	rxdata24	rxdata21	rxdata21	rxdata6
vss	txdata12	vss	rxckp	vss	rxdata23	rxdata23	rxdata5
txdataRDO	txdata13	txdata30	rxckp	vccfwdio	rxdata22	rxdata22	vccfwdio
vccio	vccio	vccio	vccfwdio	vccfwdio	vccfwdio	vccfwdio	vccfwdio
Die Edge							

Note:

In Figure 5-33, at 55-um pitch, the module depth of the 8-column reference bump matrix as shown is approximately 962.5 um.

Figure 5-34 shows the signal exit order for the 10-column x32 Advanced Package bump map.

Figure 5-34. 10-column x32 Advanced Package Bump Map: Signal Exit Order

Tx Breakout		Left to Right											
Cont...		txdataRDO	txdata0	txdata1	txdata2	txdata3	txdata4	txdata5	txdata6	txdata7	txdata8	txdata9	Cont...
Cont1...		txdata9	txdata10	txdata11	txdata12	txdata13	txdata14	txdata15	txdata16	txdata17	txdata18	txdata19	Cont1...
Cont2...		txdata19	txdata20	txdata21	txdata22	txdata23	txdata24	txdata25	txdata26	txdata27	txdata28	txdata29	Cont2...
Rx Breakout		Left to Right											
Cont...		rxckp	rxckn	rxckRD	rxtrk	rxvld	rxvldRD	rxdataRD1	rxdata31	rxdata30	rxdata29	rxdata28	Cont...
Cont1...		rxdata28	rxdata27	rxdata26	rxdata25	rxdata24	rxdata23	rxdata22	rxdata21	rxdata20	rxdata19	rxdata18	Cont1...
Cont2...		rxdata18	rxdata17	rxdata16	rxdata15	rxdata14	rxdata13	rxdata12	rxdata11	rxdata10	rxdata9	rxdata8	Cont2...

IMPLEMENTATION NOTE

High-speed Considerations for x32 Bump Maps

This Implementation Note is formulated to provide PHY implementations a set of reference x32 bump maps to encompass the max speed specified.

Bump maps in [Figure 5-35](#), [Figure 5-36](#), and [Figure 5-37](#) are the x32 implementation references for the corresponding max speed with an enhancement of the power and ground bumps. They all comply with the bump matrix rules in [Section 5.7.2.3](#), and they maintain the backward compatibility in terms of signal exit order. These reference examples have been optimized for signal integrity, power integrity, lane-to-lane skew, electro-migration stress and bump area based on most of the advanced packaging technologies in the industry. Please note that technology requirements vary, and it is still required to verify the bump map with the technology provider for actual implementation requirements and performance targets.

These bump maps have been optimized to minimize the lane to-lane routing mismatch, which is not avoidable when two different bumps at different bump pitches interoperate. [Table 5-20](#) summarizes the max skew due to bump locations for the representative cases. As a rule of thumb, each 150-um mismatch causes about 1-ps timing skew. This skew can be reduced or eliminated by the length matching effort in package channel layout design.

Table 5-20. Maximum Systematic Lane-to-lane Length Mismatch in um between the Reference Bump Maps in the Implementation Note

Rx	16-column x64 at 25 um	16-column x32 at 25 um	10-column x64 at 45 um	10-column x32 at 45 um	8-column x64 at 55 um	8-column x32 at 55 um
Tx						
16-column x64 at 25 um	0	125	351	399	560	605
16-column x32 at 25 um		0	351	393	563	618
10-column x64 at 45 um			0	159	351	463
10-column x32 at 45 um				0	428	398
8-column x64 at 55 um					0	468
8-column x32 at 55 um						0

IMPLEMENTATION NOTE

Continued

Figure 5-35. Enhanced 10-column x32 Advanced Package Bump Map Example for 32 GT/s Implementation

	1	2	3	4	5	6	7	8	9	10
1	vss		vss		vccio		vccio		vss	
2		vss		vccio		vccio		vss		vss
3	vss		vss		vccio		vccio		vss	
4		rxcksbRD		rxcksb		vss		rxdata8b		rxdata8bRD
5	txdatasbRD		txdatasb		vccio		txcksb		txcksbRD	
6		txdata5		vccio		vss		rxdata14		vccio
7	txdata4		txdata21		txdata22		txckp		rxdata13	
8		txdata6		txdata23		txckn		rxdata30		rxdataRDO
9	vss		vss				rxdata29		rxdata12	
10		txdata7		txdata20		txckRD		rxdata15		vccio
11	vccio			vccio			rxdata28		vss	
12		vccio		vccio		vccio		rxdata16		rxdata0
13	txdata3		txdata19		txdata24		txtrk		rxdata27	
14		txdata8		txdata25			rxdataRD1		vccio	
15	vss		vss		vss		vss		rxdata10	
16		txdata9		txdata25		rxvldRD		rxdata17		vss
17	txdata2		txdata18		txdata26		txvld		rxdata26	
18		vccio		txdata17		txvldRD		rxdata18		rxdata2
19	vccio			vccio			rxvld		rxdata25	
20		txdata10		vccio		vccio		rxdata25		rxdata9
21	txdata1		vss		txdataRD1		rxdata24		rxdata8	
22		txdata11		txdata27		rxtrk		rxdata19		rxdata3
23	txdata0		txdata16		vss		vccfwdio		vccfwdio	
24		vccio		txdata28		rxckRD		rxdata20		vss
25	vss		txdata15		txdata31		rxckn		rxdata23	
26		txdata12		txdata29		vss		vss		vss
27	txdataRDO		vss		vss		rxckp		rxdata22	
28		txdata13		txdata30				rxdata21		rxdata4
29	vss		txdata14		vss		vccfwdio		rxdata5	
30		vccio		vccio		vccio		vccfwdio		vccfwdio
31	vccio		vccio		vccio		vccfwdio		vccfwdio	
							Die Edge			

Note:

In Figure 5-35, at 45-um pitch, the module depth of the 10-column bump map as shown is approximately 725 um. Rows 1, 2, and 31 are required for packaging solutions using floating bridges without through-silicon vias (TSVs). They can be optional for packaging solutions with TSVs. The vccfwdio bumps are required for the tightly coupled mode up to 16 GT/s. For higher speeds, the vccfwdio bumps may be connected to the vccio bumps in package.

IMPLEMENTATION NOTE

Continued

Figure 5-36. Enhanced 16-column x32 Advanced Package Bump Map Example for 16 GT/s Implementation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	vss		vccio		vccio		vss		vss		vccio		vccio		vss	
2		vss		vccio		vccio		vss		vss		vccio		vccio		vss
3	txdatabsRD		rxcksbRD		txdatabs		txdatabs		rxcksb		txcksb		rxdata30		txcksbRD	
4		txdata4		txdata20		txdata22		vss		rxdata30		rxdata26		rxdata13		rxdata11
5	vss		txdata5		txdata21		txckp		vss		rxdata27		rxdata14			
6		txdata3		vss		txdata23		txrkn		rxdata29		vss		rxdata12		rxdataRDO
7	txdata2		txdata6		vss		txckn		rxdata31		rxdata28		vss		rxdata10	
8		vccio		txdata17		txdata24		vss		rxdataRD1		rxdata25		rxdata7		rxdata0
9	txdata1		vccio		txdata19		txckRD		rxvldRD		vccio		rxdata15		rxdata9	
10		txdata8		txdata16		vccio		txvld		vccio		rxdata18		vccio		vss
11	vss		vccio		txdata18		vccio		rxvld		vccio		rxdata16		rxdata8	
12		txdata9		txdata15		vccio		txvldRD		rxvld		rxdata19		vccio		rxdata1
13	txdata0		txdata7		txdata25		txdataRD1		vss		rxckRD		rxdata17		vccio	
14		txdata10		vss		txdata28		txdata31		rxckn		rxdata24		rxdata16		rxdata2
15	txdataRDO		txdata12		vss		txdata29		rxtrk		rxdata23		vss		rxdata3	
16		txdata11		txdata14		txdata27		vss		rxckp		rxdata21		rxdata5		rxdata4
17	vss		txdata13		txdata26		txdata30		vss		rxdata22		rxdata20		vccfwdio	
18	vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	
19	vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	

Note: In Figure 5-36, at 25- μm pitch, the module depth of the 16-column bump map as shown is approximately 250 μm . Rows 1 and 19 are required for packaging solutions using floating bridges without TSVs. They can be optional for packaging solutions with TSVs. The vccfwdio bumps are required for the tightly coupled mode up to 16 GT/s. For higher speeds, the vccfwdio bumps may be connected to the vccio bumps in package.

IMPLEMENTATION NOTE

Continued

Figure 5-37. Enhanced 8-column x32 Advanced Package Bump Map Example for 32 GT/s Implementation

	1	2	3	4	5	6	7	8
1	vss		vccio		vccio		vss	
2		vss		vccio		vccio		vss
3	vss		vccio		vccio		vss	
4		txdatasbRD		txdatasb		txcksb		txcksbRD
5	rxcksbRD		rxcksb		rxdatasb		rxdatasbRD	
6		txdata22		txckp		rxdata14		rxdataRD0
7	vss		txdata23		vss		rxdata13	
8		txdata21		txckn		vccio		vccio
9	txdata5		vss		rxdata30		rxdata12	
10		vccio		vccio		rxdata15		rxdata0
11	txdata6		txdata24		rxdata31		rxdata11	
12		txdata20		txckRD		rxdata16		rxdata1
13	vss		vss		vss		vss	
14		txdata19		txtrk		rxdata29		rxdata2
15	txdata7		txdata25		rxdataRD1		rxdata10	
16		txdata18		vccio		rxdata28		rxdata3
17	vccio		txdata26		vss		rxdata9	
18		txdata17		txvld		rxdata27		vccio
19	txdata4		vss		rxvldRD		rxdata8	
20		txdata8		txvldRD		vccio		rxdata4
21	vss		txdata27		rxvld		rxdata17	
22		txdata9		vccio		rxdata26		vss
23	txdata3		txdata28		vss		rxdata18	
24		txdata10		txdataRD1		rxdata25		rxdata7
25	txdata2		txdata29		rxtrk		rxdata19	
26	vccio		vccio		vccio		vccfdio	
27	txdata1		txdata16		rxckRD		rxdata20	
28		txdata11		txdata31		rxdata24		rxdata6
29	txdata0		txdata15		vss		vss	
30		txdata12		txdata30		vccfdio		rxdata5
31	vss		vss		rxckn		rxdata21	
32		txdata13		vss		rxdata23		vss
33	txdataRD0		txdata14		rxckp		rxdata22	
34		vccio		vccio		vccfdio		vccfdio
35	vccio		vccio		vccfdio		vccfdio	

Note: In Figure 5-37, at 55-um pitch, the module depth of the 8-column bump map as shown is approximately 990 um. Rows 1, 2, and 35 are required for packaging solutions using floating bridges without TSVs. They can be optional for packaging solutions with TSVs. The vccfwdio bumps are required for the tightly coupled mode up to 16 GT/s. For higher speeds, the vccfwdio bumps may be connected to the vccio bumps in package.

5.7.2.4 x64 and x32 Advanced Package Module Interoperability

x64 and x32 Advanced Package Module bump maps enable interoperability between all Tx and Rx combinations of x64 or x32, 10-column, 16-column, or 8-column Modules, in both Normal-to-Normal module orientation or Normal-to-Mirrored module orientation.

However, if x64 to x32 modules or x32 to x32 modules have normal and mirrored orientation as shown in [Figure 5-38](#) and [Figure 5-39](#), respectively, signal traces between the TX half and RX half will crisscross and require swizzling technique which refers to rearranging the physical connections between signal bumps of two chiplets to optimize the layout and routing on the interposer or substrate. It involves changing the order of the connections or route on different layers without altering the netlist or the electrical functionality of the design. Moreover, connections between 8-column, 16-column, and 10-column modules may need to be routed to adjacent columns (swizzle and go across). In all cases, the electrical spec must be met for all these connections.

It is optional for a x64 Module to support interoperability with a x32 Module. The following requirements apply when a x64 module supports x32 interoperability:

- When a x64 module connects to x32 module, the connection shall always be contained to the lower half of the x64 module. This must be followed even with x32 lane reversal described below.
- Electrical specifications must be met for combinations that require signal-routing swizzling.
- Lane reversal will not be permitted on **CKP-**, **CKN-**, **CKRD-**, **VLD-**, **VLD RD-**, **TRK-**, and sideband-related pins. These pins need to be connected appropriately. Swizzling for these connections is acceptable.
- x64 module must support a lane-reversal mode in a x32 manner (i.e., **TD_P[31:0] = TD_L[0:31]**). When a x64 module is connected to a x32 module, in either Normal or Mirrored orientation, the upper 32 bits are not used and should be disabled.
- It is not permitted for a single module of larger width to simultaneously interop with two or more modules of a lower width. For example, a x64 Advanced Package module physically connected to two x32 Advanced Package modules is prohibited.

Additional technological capabilities or layers may be needed to accomplish swizzling on data/auxiliary signals.

Table 5-21 summarizes the connections between combinations of x64 and x32 modules in both Normal-to-Normal and Normal-to-Mirrored module orientations. The table applies to all combinations of 10-column, 16-column, or 8-column modules on either side of the Link.

Table 5-21. x64 and x32 Advanced Package Connectivity Matrix

Normal Module Tx	Normal Module		Mirrored Module	
	Rx			
	x64	x32	x64	x32
x64	TX[63:0] – RX[63:0] ^a	TX[31:0] – RX[31:0] ^b	rTX[63:0] – RX[0:63] ^{c d}	rTX[31:0] – RX[0:31] ^{c e}
x32	TX[31:0] – RX[31:0] ^b	TX[31:0] – RX[31:0] ^b	rTX[31:0] – RX[0:31] ^{c e}	rTX[31:0] – RX[0:31] ^{c e}

- a. Entry "TX[63:0] – RX[63:0]" is for Normal Module connections between two x64 modules without lane reversal. This applies to x64-to-x64 combination.
- b. Entry "TX[31:0] – RX[31:0]" is for Normal Module connections between lower 32-bit half without lane reversal. This applies to x64-to-x32, x32-to-x64, and x32-to-x32 combinations.
 - c. The prefix "r" means lane reversal is enabled on the Transmitter lanes, and:
 - "rTX[63:0]" means TD_P[63:0] = TD_L[0:63], to be connected with RD_P[0:63]
 - "rTX[31:0]" means TD_P[31:0] = TD_L[0:31], to be connected with RD_P[0:31].
 - d. Entry "rTX[63:0] – RX[0:63]" = Normal-to-Mirrored Module connections between two x64 modules with TX lane reversal. This applies to x64-to-x64 Normal-to-Mirrored combinations.
 - e. Entry "rTX[31:0] – RX[0:31]" = Normal-to-Mirrored Module connections between lower 32-bit half with TX lane reversal. This applies to x64-to-x32, x32-to-x64, and x32-to-x32 Normal-to-Mirrored combinations.

The defined bump matrices can achieve optimal skew between bump matrices of differing depths, and the worst-case trace-reach skews are expected to be within the maximum lane-to-lane skew limit for the corresponding data rates as defined in [Section 5.3](#) and [Section 5.4](#).

[Figure 5-38](#) and [Figure 5-39](#) show examples of normal and mirrored x64-to-x32 and x32-to-x32 Advanced Package Module connections, respectively.

Figure 5-38. Example of Normal and Mirrored x64-to-x32 Advanced Package Module Connection

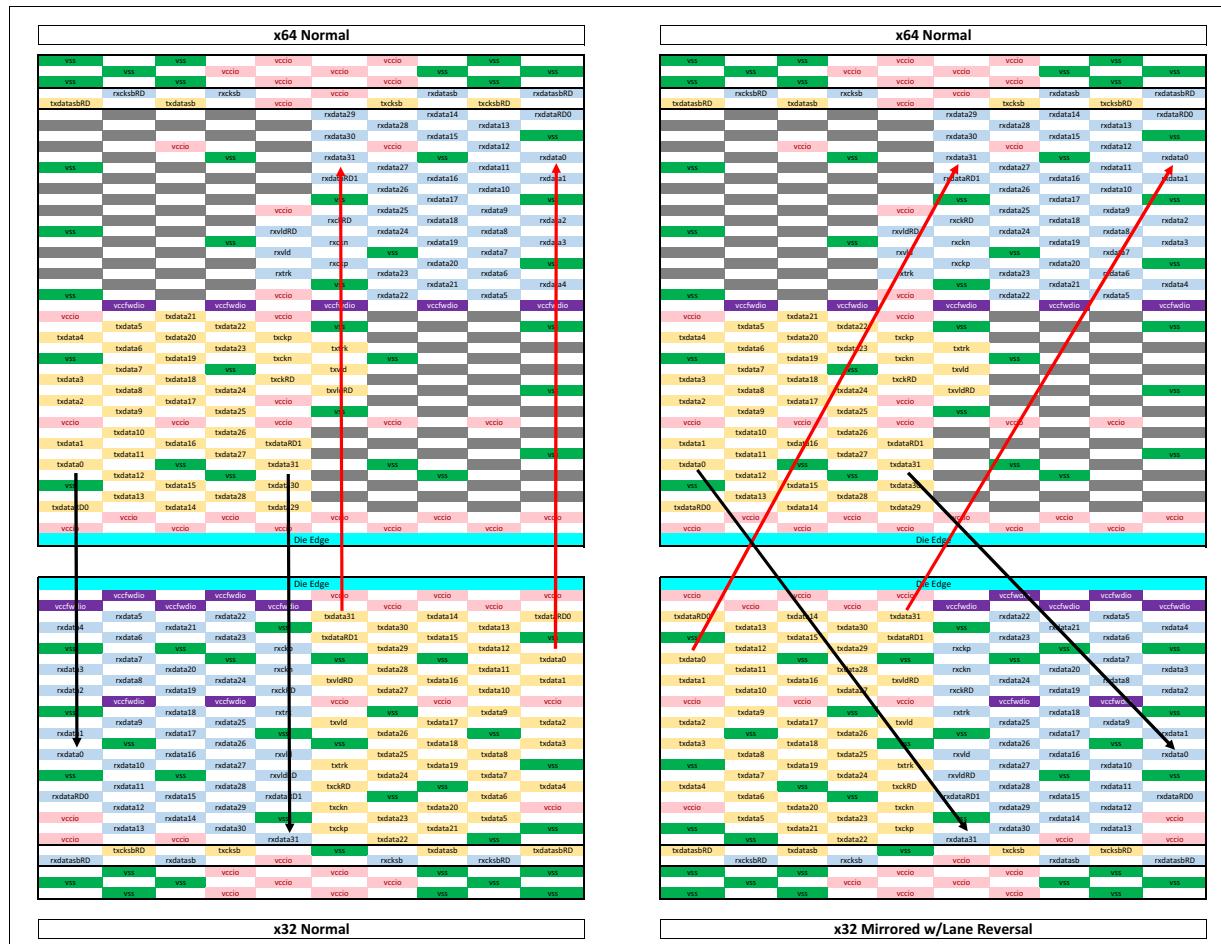
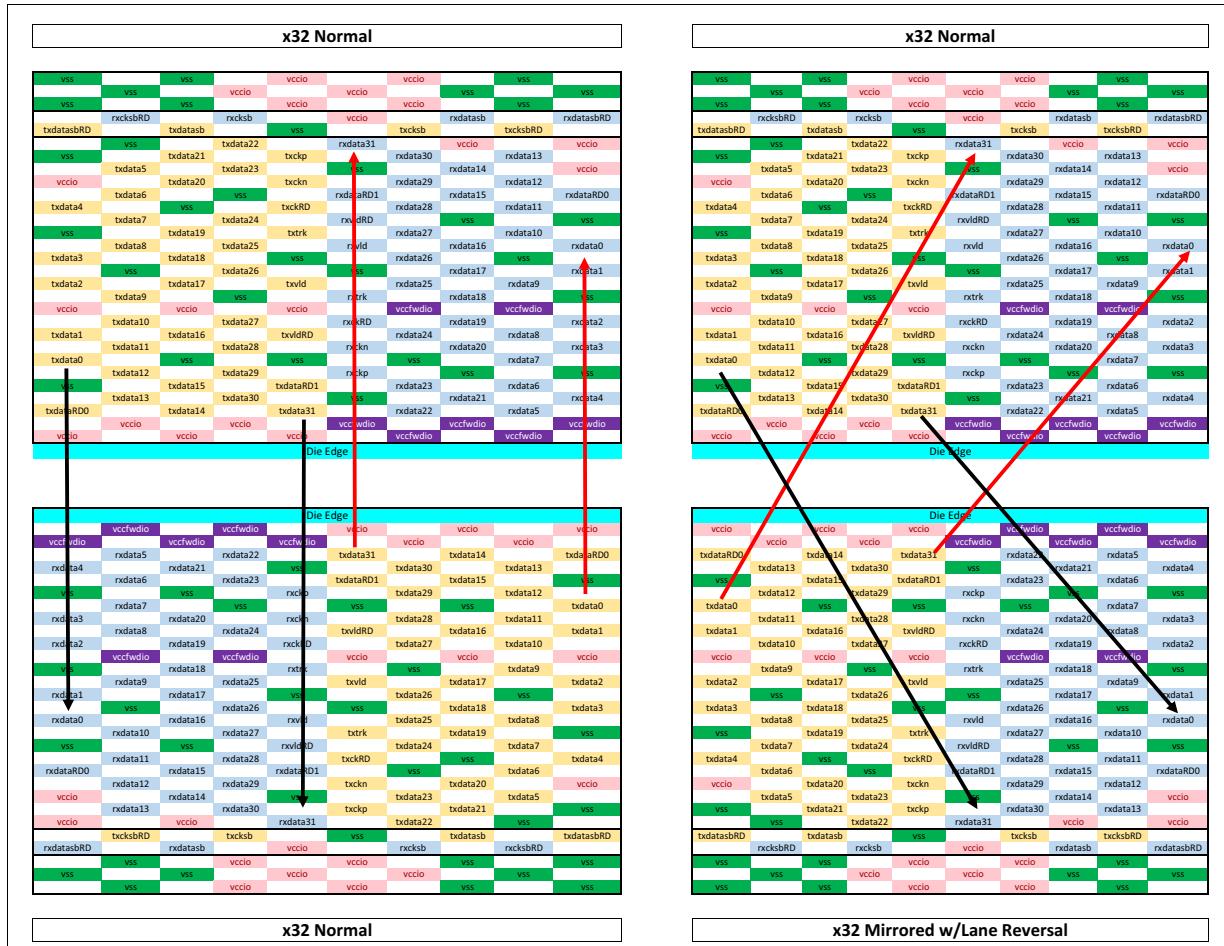


Figure 5-39. Example of Normal and Mirrored x32-to-x32 Advanced Package Module Connection



Although the depth of the bump map for 48 GT/s and 64 GT/s is greater than that for lower data rates, the module width and signal exit order remain unchanged. As a result, the bump maps for 48 GT/s and 64 GT/s are interoperable with those for lower speeds.

5.7.2.5 Module Naming of Advanced Package Modules

This section describes the Module naming convention of x64 and x32 Advanced Package modules in a multi-module configuration.

The Module naming is defined to help with connecting the Modules deterministically which, in turn, will help minimize the multiplexing requirements in the Multi-module PHY Logic (MMPL).

The naming of M0, M1, M2, and M3 will apply to 1, 2, or 4 Advanced Package modules that are aggregated through the MMPL.

Figure 5-40 shows the naming convention for 1, 2, or 4 Advanced Package Modules when they are connected to their "Standard Die Rotate" Module counterparts that have same number of Advanced Package Modules.

Note: The double-ended arrows in Figure 5-40 through Figure 5-43 indicate Module-to-Module connections.

Figure 5-40. Naming Convention for One-, Two-, and Four-module Advanced Package Paired with “Standard Die Rotate” Configurations

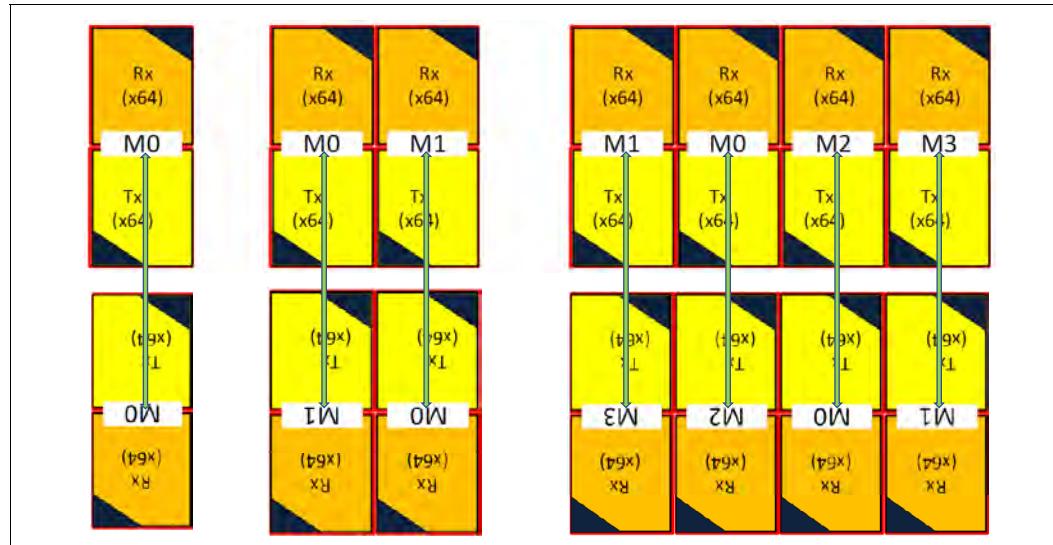


Figure 5-41 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their “Mirrored Die Rotate” counterparts with the same number of Advanced Package modules.

Figure 5-41. Naming Convention for One-, Two-, and Four-module Advanced Package Paired with “Mirrored Die Rotate” Configurations

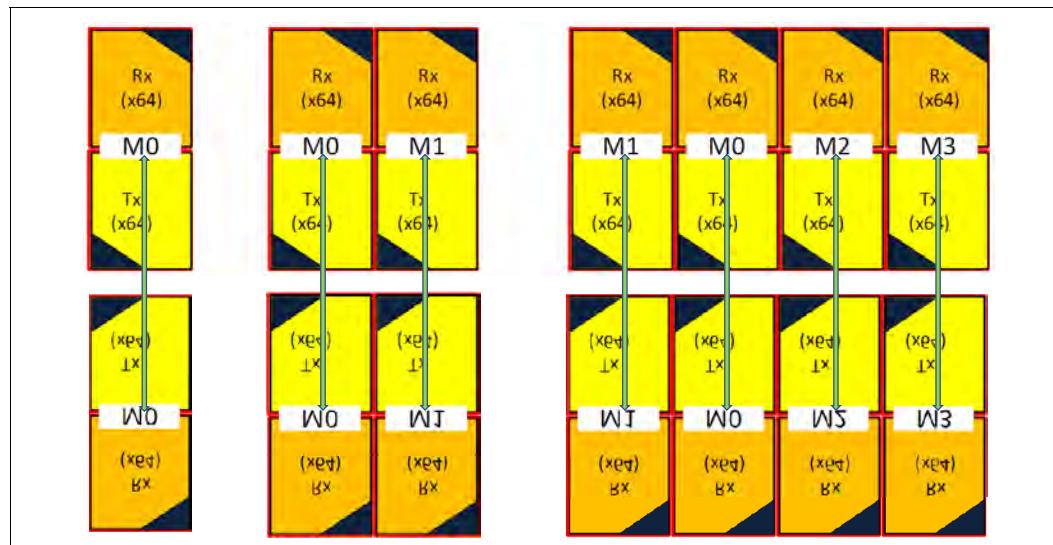


Table 5-22 summarizes the connections between the combinations shown in Figure 5-40 and Figure 5-41.

Table 5-22. Summary of Advanced Package Module Connection Combinations with Same Number of Modules on Both Sides

Advanced Package Module Connections (Same # of Modules on Both Sides)	Standard Die Rotate Counterpart	Mirrored Die Rotate Counterpart
x1 – x1	• M0 – M0	• M0 – M0
x2 – x2	• M0 – M1 • M1 – M0	• M0 – M0 • M1 – M1
x4 – x4	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3

Figure 5-42 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their “Standard Die Rotate” counterparts that have a different number of Advanced Package modules.

Figure 5-42. Examples for Advanced Package Configurations Paired with “Standard Die Rotate” Counterparts, with a Different Number of Modules

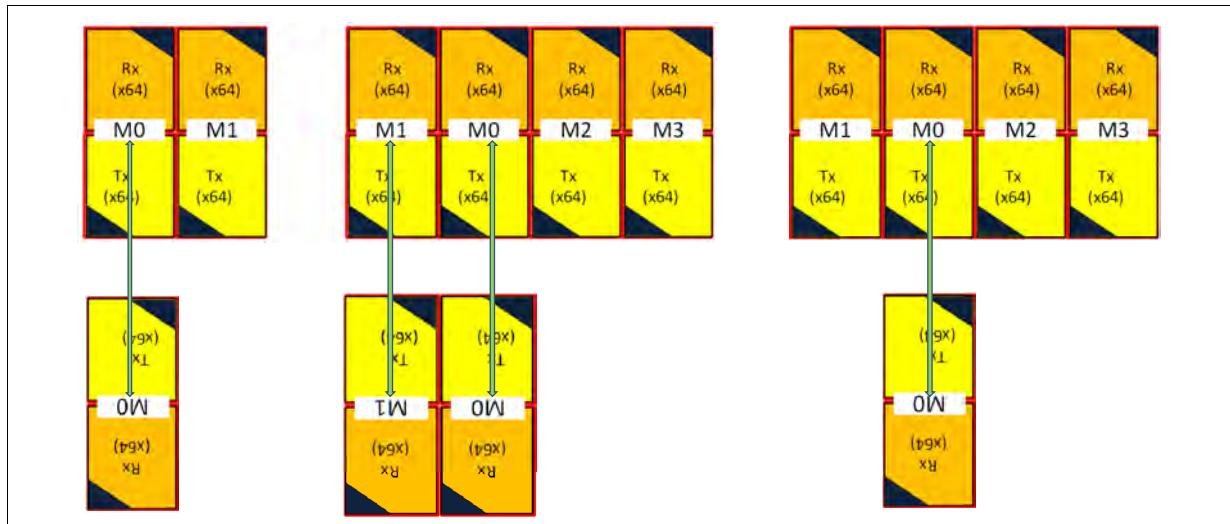


Figure 5-43 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their “Mirrored Die Rotate” counterparts that have a different number of Advanced Package modules.

Figure 5-43. Examples for Advanced Package Configurations Paired with “Mirrored Die Rotate” Counterparts, with a Different Number of Modules



Table 5-23 summarizes the connections between the combinations shown in Figure 5-42 and Figure 5-43.

Table 5-23. Summary of Advanced Package Module Connection Combinations with Different Number of Modules on Both Sides

Advanced Package Module Connections (Different # of Modules on Both Sides)	Standard Die Rotate Counterpart ^a	Mirrored Die Rotate Counterpart ^a
x2 – x1	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC
x4 – x2	<ul style="list-style-type: none"> • M0 – M0 • M1 – M1 • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M1 • M1 – M0 • M2 – NC • M3 – NC
x4 – x1	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M2 – NC • M3 – NC

a. NC indicates no connection.

5.7.3 Standard Package

Interconnect channel should be designed with 50-ohm characteristic impedance. Insertion loss and crosstalk for requirement at Nyquist frequency with Receiver termination is defined in [Table 5-24](#).

Table 5-24. IL and Crosstalk for Standard Package: With Receiver Termination Enabled

Data Rate	4, 8 GT/s	12, 16 GT/s	24, 32 GT/s
VTF Loss (dB) ^{a b c}	$L(0) > -4.5$ $L(f_N) > -7.5$	$L(0) > -4.5$ $L(f_N) > -6.5$	$L(0) > -4.5$ $L(f_N) > -7.5$
VTF Crosstalk (dB)	$XT(f_N) < 3 * L(f_N) - 11.5$ and $XT(f_N) < -25$	$XT(f_N) < 3 * L(f_N) - 11.5$ and $XT(f_N) < -25$	$XT(f_N) < 2.5 * L(f_N) - 10$ and $XT(f_N) < -26$

a. Voltage Transfer Function for 4 GT/s and 8 GT/s (Tx: 30 ohm / 0.3pF; Rx: 50 ohm / 0.3pF).

b. Voltage Transfer Function for 12 GT/s and 16 GT/s (Tx: 30 ohm / 0.2pF; Rx: 50 ohm / 0.2pF).

c. Voltage Transfer Function for 24 GT/s and 32 GT/s (Tx: 30 ohm / 0.125pF; Rx: 50 ohm / 0.125pF).

IL and crosstalk for requirement at Nyquist frequency without Receiver termination is defined by [Table 5-25](#). Loss and crosstalk specifications between DC and Nyquist f_N follow the same methodology defined in [Section 5.7.2.1](#).

Table 5-25. IL and Crosstalk for Standard Package: No Rx Termination

Data Rate	4-12 GT/s	16 GT/s
VTF Loss (dB) ^{a b}	$L(f_N) > -1.25$	$L(f_N) > -1.15$
VTF Crosstalk (dB)	$XT(f_N) < 7 * L(f_N) - 12.5$ and $XT(f_N) < -15$	$XT(f_N) < 4 * L(f_N) - 13.5$ and $XT(f_N) < -17$

a. Voltage Transfer Function for 4 GT/s and 8 GT/s (Tx: 30 ohm / 0.3pF; Rx: 0.2 pF).

b. Voltage Transfer Function for 12 GT/s and 16 GT/s (Tx: 30 ohm / 0.2pF; Rx: 0.2 pF).

Table 5-26. Standard Package Module Signal List (Sheet 1 of 2)

Signal Name	Count	Description
Data		
TXDATA[15:0]	16	Transmit Data
TXVLD	1	Transmit Data Valid; Enables clocking in corresponding module
TXTRK	1	Transmit Track signal
TXCKP	1	Transmit Clock Phase-1
TXCKN	1	Transmit Clock Phase-2
RXDATA[15:0]	16	Receive Data
RXVLD	1	Receive Data Valid; Enables clocking in corresponding module
RXTRK	1	Receive Track
RXCKP	1	Receive Clock Phase-1
RXCKN	1	Receive Clock Phase-2
Sideband		
TXDATASB	1	Sideband Transmit Data
RXDATASB	1	Sideband Receiver Data
TXCKSB	1	Sideband Transmit Clock

Table 5-26. Standard Package Module Signal List (Sheet 2 of 2)

Signal Name	Count	Description
RXCKSB	1	Sideband Receive Clock
Power and Voltage		
VSS		Ground Reference
VCCIO		I/O supply
VCCAON		Always on Aux supply (sideband)

5.7.3.1 x16 Standard Package Module Bump Map

Figure 5-44 and Figure 5-46 show the reference bump matrices for x16 (one module) and x32 (two module) Standard Packages, respectively, for data rates up to 32 GT/s.

It is strongly recommended to follow the bump matrices provided in Figure 5-44 for one module and Figure 5-46 for two module Standard Packages. The lower left corner of the bump map will be considered “origin” of a bump matrix.

Signal exit order for x16 and x32 Standard Package bump matrices are shown in Figure 5-45 and Figure 5-47, respectively.

The following rules must be followed for Standard Package bump matrices:

- The signals within a column must be preserved. For example, for a x16 (one module Standard Package) shown in Figure 5-44, Column 1 must contain the signals: **txdata0**, **txdata1**, **txdata4**, **txdata5**, and **txdatasb**.
- The signals must exit the bump field in the order shown in Figure 5-45. Layer 1 and Layer 2 are two different signal routing layers in a Standard Package.

It is strongly recommended to follow the supply and **ground** pattern shown in the bump matrices. It must be ensured that sufficient supply and **ground** bumps are provided to meet channel characteristics (FEXT and NEXT) and power-delivery requirements.

The following rules must be followed for instantiating multiple modules of Standard Package bump matrix:

- When looking at a die such that the PCIe Modules are on the south side, Tx should always precede Rx within a module along the die’s edge when going from left to right.
- When instantiating multiple modules, the modules must be stepped in the same orientation and abutted. Horizontal or vertical mirroring is not permitted.

If more Die Edge Bandwidth density is required, it is permitted to stack two modules before abutting. If two modules are stacked, the package may need to support at least four routing layers for PCIe signal routing. An example of stacked Standard Package Module instantiations is shown in Figure 5-46.

- If only one stacked module is instantiated, when looking at a die such that the PCIe Modules are on the south side, Tx should always precede Rx within a module along the die’s edge when going from left to right.
- When instantiating multiple stacked modules, the modules must be stepped in the same orientation and abutted. Horizontal or vertical mirroring is not permitted.

Note: An example of signal routing for stacked module is shown in Figure 5-48.

Figure 5-44. Standard Package Bump Map: x16 interface

Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
vccio	txdatasb	vccio	txcksb	vccio	vccaon	vccio	vccaon	vccio	rxcksb	vccio	rxdatasb
vss	vss										
vccio	txdata5	txdata7	txckn	txdata9	txdata11	txdata10	txdata11	txdata10	rxdata8	rxdata6	rxdata4
vss	rxckp	rxdata5	rxdata7								
vss	txdata4	txdata6	txckp	txdata8	txdata10	txdata11	txdata10	txdata11	rxckn	rxdata9	rxdata5
vss	rxdata9	rxdata7	rxdata4								
vccio	txdata3	txdata13	txvld	txdata15	vccio	txdata14	txdata15	txdata12	rxdata12	rxdata2	rxdata0
vccio	txdata1	vss	vss	vss	vccio	vss	vss	vss	rxvld	vss	rxdata1
vss	txdata0	txtrk	txdata12	txdata14	vss	txdata15	txdata13	txdata13	rxdata13	rxdata3	rxdata1
Die Edge											

Figure 5-45. Standard Package x16 interface: Signal exit order

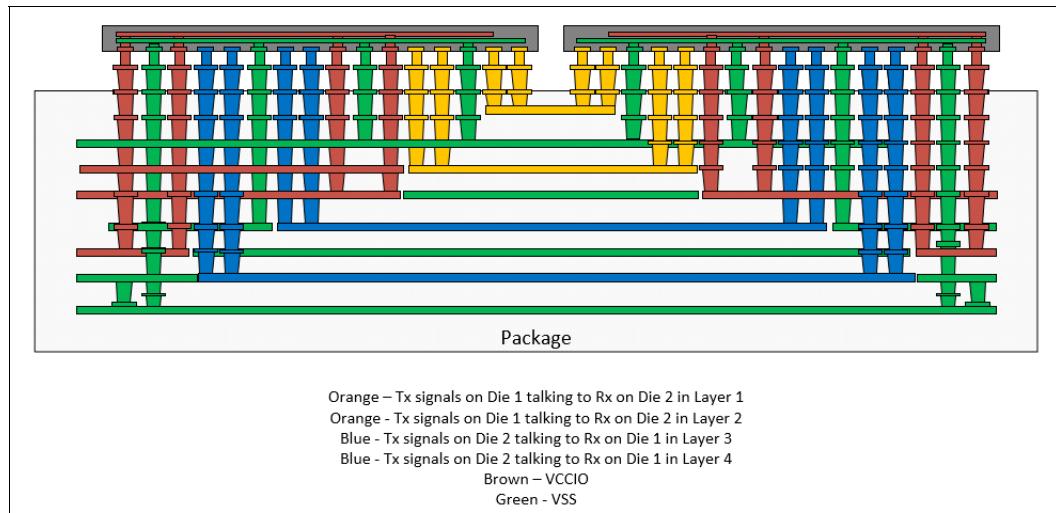
Layer 1	Tx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Rx	
Layer 2	Module	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module	
Sideband		txdatasb				txcksb										rxcksb						rxdatasb	

Figure 5-46. Standard Package Bump Map: x32 interface

Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
m1txdatasb	m1txcksb	m1txcksb	vccao	vccao	m2txcksb	m2txcksb	m1rxcksb	m1rxcksb	m1rxdatasb	vccao	vccao
vccio	vccio	vccio									
vss	vss	vss									
m2rxdata6	m2rxdata8	m2rxdata8	vss	vss	m2txdata9	m2txdata9	m2txdata7	m2txdata7	m2txdata5	vss	vss
m2rxdata4	m2rxckp	m2rxckp	m2rxdata10	vss	m2txdata11	m2txdata11	m2txckn	m2txckn	m2txdata4	vss	vss
m2rxdata5	m2rxckn	m2rxckn	m2rxdata11	vss	m2txdata10	m2txdata10	m2txckp	m2txckp	m2txdata6	vss	vss
m2rxdata7	m2rxdata9	m2rxdata9	vss	vss	m2txdata8	m2txdata8	m2txdata6	m2txdata6	m2txdata4	vss	vss
m2rxdata2	m2rxdata12	m2rxdata12	vss	vss	m2txdata13	m2txdata13	m2txdata11	m2txdata11	m2txdata9	vss	vss
m2rxdata0	m2rxtrk	m2rxtrk	m2rxdata14	vss	m2txdata15	m2txdata15	m2txvld	m2txvld	m2txdata1	vss	vss
m2rxdata1	m2rvld	m2rvld	m2rxdata15	vss	m2txdata14	m2txdata14	m2txtrk	m2txtrk	m2txdata0	vss	vss
m2rxdata3	m2rxdata13	m2rxdata13	vccio	vccio	m2txdata12	m2txdata12	m2txdata2	m2txdata2	m2txdata1	vccio	vccio
vccio	vccio	vccio									
vccio	vss	vss	vss								
m1txdata7	m1txdata7	m1txdata9	m1txdata9	m1txdata11	m1txdata11	m1txdata10	m1txdata10	m1rxdata8	m1rxdata8	m1rxdata6	m1rxdata4
m1txdata5	m1txckn	m1txckn	m1txdata11	vss	m1txdata10	m1txdata10	m1rxckp	m1rxckp	m1rxdata4	vss	vss
vss	vss	vss									
m1txdata4	m1txckp	m1txckp	m1txdata10	vss	m1txdata11	m1txdata11	m1rxckn	m1rxckn	m1rxdata5	vss	vss
vss	m1txdata6	m1txdata6	m1txdata8	vss	vss	vss	m1rxdata9	m1rxdata9	m1rxdata7	vss	vss
vccio	m1txdata3	m1txdata3	m1txdata13	vccio	vccio	vccio	m1rxdata12	m1rxdata12	m1rxdata2	vccio	vccio
m1txdata1	m1txvld	m1txvld	m1txdata15	vccio	vccio	m1rxdata14	m1rxdata14	m1rxtrk	m1rxtrk	m1rxdata0	vccio
vccio	vss	vss	vss								
m1txdata0	m1txdata2	m1txdata2	m1txdata12	vss	m1txdata14	m1txdata14	m1rxvld	m1rxvld	m1rxdata1	vss	vss
vss	m1txdata2	m1txdata2	m1txdata12	vss	vss	vss	m1rxdata13	m1rxdata13	m1rxdata3	vss	vss
Die Edge											

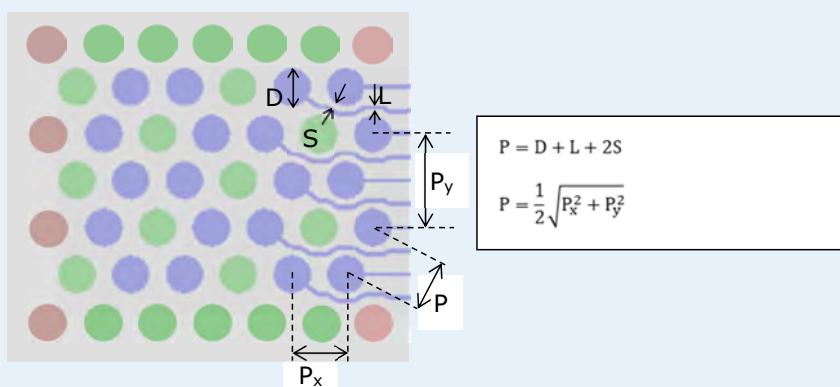
Figure 5-47. Standard Package x32 interface: Signal exit routing

Layer 1	Tx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Rx
Layer 2	Module 1	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module 1
Layer 3	Rx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Tx
Layer 4	Module 2	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module 2
Sideband		m1txdatasb	m2rxdatasb	m1txcksb	m2rxcksb	m2txcksb	m1rxcksb	m2txdatasb	m1rxdatasb													Sideband

Figure 5-48. Standard Package cross section for stacked module

IMPLEMENTATION NOTE

Figure 5-49 shows a breakout design reference with the Standard Package channel based on the bump pitch and on routing design rules.

Figure 5-49. Standard Package reference configuration

- 4-row deep breakout per routing layer
- Example 1: $P_y = 190.5$ um, $P_x \approx 111.5$ um, $P \approx 110$ um
- Example 2: $P_y = 190.5$ um, $P_x \approx 177$ um, $P \approx 130$ um

[Figure 5-50](#) and [Figure 5-51](#) present recommended bump configurations for Standard Package modules for 48 GT/s and 64 GT/s. As with the Advanced Package configurations, a greater number of power and ground bumps have been integrated to support the increased data rates. Additional bump maps or further optimizations may be introduced in subsequent updates to this specification. To enhance die edge bandwidth density, the UCIe Consortium is investigating the feasibility of bump maps for 48 GT/s and 64 GT/s that have die edge equal to that of 32 GT/s and below. This exploration involves optimizing bump maps and enhancing circuits through improvements in equalization, termination network, skew, jitter, and noise reduction.

Figure 5-50. Standard Package Bump Map: 48 GT/s and 64 GT/s x32 Interface

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12	Column 13	Column 14
	rxdatasb2		rxcksb2		vccio		vccinfaon		vccio		rxdatasb2		txcksb2
txdatasb1		txcksb1		vccio		vccinfaon		vccio		rxdatasb1		rxdatack1	
	vccio		vccio		vccio		vccio		vccio		vccio		vccio
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	rxdata6		rxckp		rxdata10		txdata11		txdata8		txdata7		txdata4
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	rxdata5		rxckn		rxdata9		VSS		VSS		VSS		VSS
VSS		VSS		VSS		VSS		txdata9		txckn		txdata5	
	VSS		vccio		VSS		VSS		VSS		VSS		VSS
rxdata4		rxdata7		rxdata8		rxdata11		txdata10		txckp		txdata6	
	vccio		vccio		vccio		VSS		VSS		VSS		VSS
VSS		VSS		VSS		VSS		vccio		vccio		vccio	
	rxdata2		rxvld		rxdata14		txdata15		txdata12		txdata3		txdata0
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	rxdata1		rxtrk		rxdata13		VSS		VSS		VSS		VSS
VSS		VSS		VSS		VSS		txdata13		txtrk		txdata1	
	VSS		VSS		VSS		VSS		VSS		VSS		VSS
rxdata0		rxdata3		rxdata12		rxdata15		txdata14		txvld		txdata2	
	VSS		vccio		VSS		VSS		vccio		vccio		VSS
vccio		vccio		vccio		vccio		vccio		vccio		vccio	
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	txdata4		txdata7		txdata8		txdata11		rxdata10		rxckp		rxdata6
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	VSS		VSS		VSS		VSS		rxdata9		rxckn		rxdata5
VSS		txdata5		txckn		txdata9		VSS		VSS		VSS	
VSS		VSS		VSS		VSS		VSS		vccio		VSS	
	txdata6		txckp		txdata10		rxdata11		rxdata8		rxdata7		rxdata4
VSS		VSS		VSS		VSS		vccio		vccio		vccio	
	vccio		vccio		vccio		VSS		VSS		VSS		VSS
txdata0		txdata3		txdata12		txdata15		rxdata14		rxvld		rxdata2	
	VSS		vccio		VSS		VSS		VSS		VSS		VSS
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	txdata1		txtrk		txdata13		VSS		VSS		VSS		VSS
VSS		VSS		VSS		VSS		VSS		VSS		VSS	
	txdata2		txvld		txdata14		rxdata15		rxdata12		rxdata3		rxdata0
Die Edge													

Figure 5-51. x16 Standard Package Potential Bump Map: 48 GT/s and 64 GT/s x16

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12	Column 13	Column 14		
vccio	txdatasb1		txcksb1		vccio		vccio		vccio		vccio		rxdatasb1		rxdatack1
vss		vccio		vccio		vss		vss		vss		vss		vss	
txdata4		txdata7		txdata8		txdata11		rxdata10		rxckp		rxdata6			
vss		vss		vss		vss		rxdata9		rxckn		rxdata5			
txdata5		txckn		txdata9		vss		vss		vss		vss			
vss		vss		vss		vss		vss		vccio		vss			
txdata6		txckp		txdata10		rxdata11		rxdata8		rxdata7		rxdata4			
vss		vss		vss		vss		vccio		vccio		vccio			
vccio			vccio		vccio		vss		vss		vss		vss		
txdata0		txdata3		txdata12		txdata15		rxdata14		rxvld		rxdata2			
vss		vss		vss		vss		vss		vss		vss			
txdata1		txtrk		txdata13		vss		vss		rxtrk		rxdata1			
vss		vss		vss		vss		vss		vss		vss			
txdata2		txvld		txdata14		rxdata15		rxdata12		rxdata3		rxdata0			
Die Edge															

5.7.3.2 x8 Standard Package Module Bump Map

Designs can choose to add a PCIe-S port for sort/pre-bond test purposes in scenarios where they need the high bandwidth of PCIe, but the design is an advanced package design, or for any other reason. To reduce the chiplet's die edge when supporting such a PCIe-S usage, a x8 version of PCIe-S is provided. This is an additional option that goes beyond the available standard x16 PCIe-S port options. A PCIe-S x16 Module, including bump maps for all data rates, can optionally support a connection to a PCIe-S x8 Module. When this connection is supported, the connection is always to the lower x8 lanes (i.e., Lanes 7:0). PCIe-S x8 designs must support lane reversal and degraded mode operation to x4. PCIe-S x16 designs that support connection to a x8 Module must support lane reversal, and must support degraded mode operation to x4 on its lower 8 lanes when connected to a x8 Module.

PCIe-S x8 support is limited to a single module configuration and up to 32 GT/s. When a PCIe-S x8 port is connected to a multi-module x16 port, it is always connected to Module 0 PCIe-S x16.

Figure 5-52 shows the reference bump matrix for a x8 Standard Package.

Figure 5-52. Standard Package Bump Map: x8 Interface

Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
txdatasb		txcksb		rxcksb		rxdatasb	
	vccio		vccio		vccio		vccio
vss		vss		vss		vss	
	Txdata0		Txdata7		Rxdata6		Rxdata1
vss		Txckn		vss		Rxckp	
	vss		vss		vss		vss
vccio		Txckp		vccio		Rxckn	
	Txdata1		Txdata6		Rxdata7		Rxdata0
vccio		vss		vccio		vss	
	Txdata3		Txdata4		Rxdata5		Rxdata2
vccio		Txvld		vccio		Rxtrk	
	vss		vss		vss		vss
vss		Txtrk		vss		Rxvld	
	Txdata2		Txdata5		Rxdata4		Rxdata3
Die Edge							

It is strongly recommended to follow the bump matrix provided in Figure 5-52. The lower left corner of the bump map will be considered “origin” of a bump matrix.

The same rules as mentioned for x16 and x32 Standard Package bump matrices in Section 5.7.3.1 must be followed for the x8 bump matrix.

5.7.3.3 x16 and x8 Standard Package Module Interoperability

A x8 bump matrix will either connect to another x8 bump matrix or to bits [7:0] of a x16 bump matrix.

5.7.3.4 Module Naming of Standard Package Modules

This section describes the Module naming convention of Standard Package Modules in a multi-module configuration.

The naming of M0, M1, M2, and M3 will apply to 1, 2, or 4 Standard Package modules that are aggregated through MMPL, in stacked and unstacked configuration combinations.

Figure 5-53 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their “Standard Die Rotate” module counterparts with the same number of Standard Package modules, with either same stack or same unstacked configuration.

Note: The double-ended arrows in Figure 5-53 through Figure 5-57 indicate Module-to-Module connections.

Figure 5-53. Naming Convention for One-, Two-, and Four-module Standard Package Paired with “Standard Die Rotate” Configurations

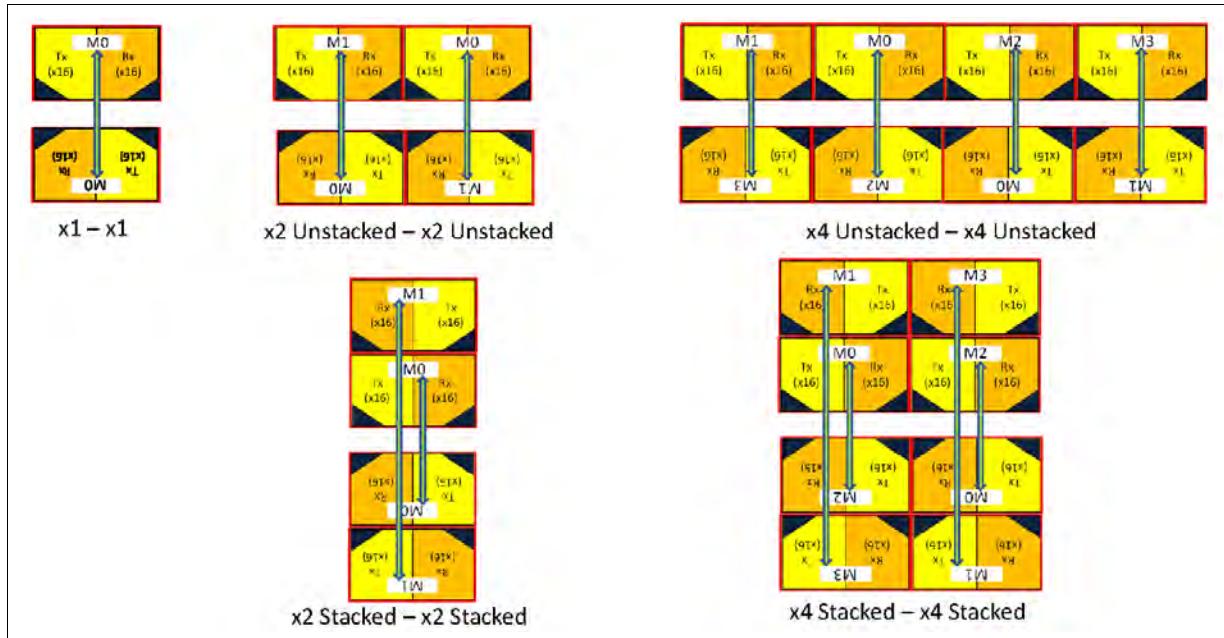


Figure 5-54 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their “Mirrored Die Rotate” counterparts that have same number of Standard Package modules, with either same stack or same unstacked configuration.

Figure 5-54. Naming Convention for One-, Two-, and Four-module Standard Package Paired with “Mirrored Die Rotate” Configurations

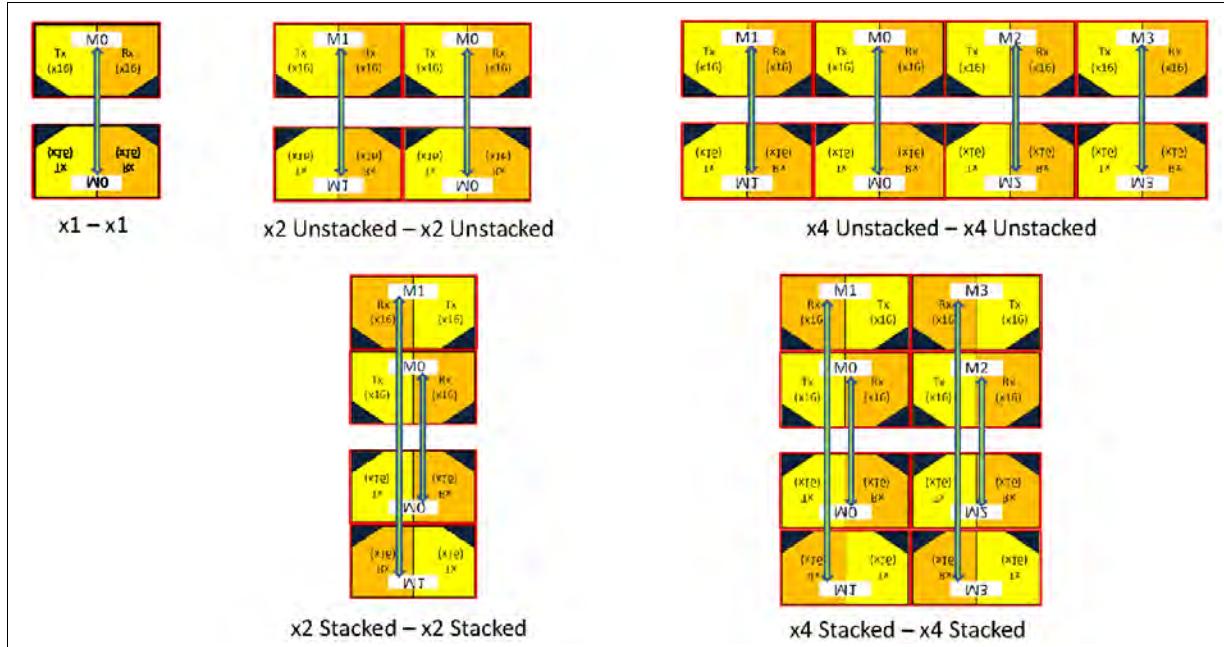


Table 5-27 summarizes the connections between the combinations shown in Figure 5-53 and Figure 5-54.

Table 5-27. Summary of Standard Package Module Connection Combinations with Same Number of Modules on Both Sides^{a b}

Standard Package Module Connections (Same # of Modules on Both Sides)	Standard Die Rotate Counterpart	Mirrored Die Rotate Counterpart	
		Option 1 (See Figure 5-54)	Option 2 ^c (See Figure 5-57)
x1 – x1	• M0 – M0	• M0 – M0	
x2 Unstacked – x2 Unstacked	• M0 – M1 • M1 – M0	• M0 – M0 • M1 – M1	
x2 Stacked – x2 Stacked	• M0 – M0 • M1 – M1	• M0 – M0 • M1 – M1	• M0 – M1 • M1 – M0
x4 Unstacked – x4 Unstacked	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3	
x4 Stacked – x4 Stacked	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3	• M0 – M1 • M1 – M0 • M2 – M3 • M3 – M2

a. Mirror-to-Mirror connection will be same as non-mirrored case.

b. Mirror die connectivity may have jogs and need additional layers on package.

c. For some mirrored cases, there are possible alternative connections to allow design choices between more routing layers vs. max data rates, shown as Option 1 and Option 2 in Table 5-27. For x2 – x2 Stacked and x4 – x4 Stacked cases, Option 1 typically requires 2x the routing layers and enables nominal data rates, while Option 2 enables same the layer count but at reduced max data rates due to potential crosstalk. See Figure 5-56 for Option 2 connection illustrations.

Figure 5-55 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their “Standard Die Rotate” counterparts that have a different number of Standard Package modules.

Figure 5-55. Examples for Standard Package Configurations Paired with “Standard Die Rotate” Counterparts, with a Different Number of Modules



Figure 5-56 shows the naming convention for 1, 2, or 4 Standard Package Modules when they are connected to their “Mirrored Die Rotate” counterparts that have a different number of Standard Package Modules.

Figure 5-56. Examples for Standard Package Configurations Paired with “Mirrored Die Rotate” Counterparts, with a Different Number of Modules



Figure 5-57 illustrates the possible alternative connections for some mirrored cases to allow design choices between more routing layers vs. reduced max data rates due to potential crosstalk, shown as Option 2 in Table 5-27 and Table 5-28.

Figure 5-57. Additional Examples for Standard Package Configurations Paired with “Mirrored Die Rotate” Counterparts, with a Different Number of Modules

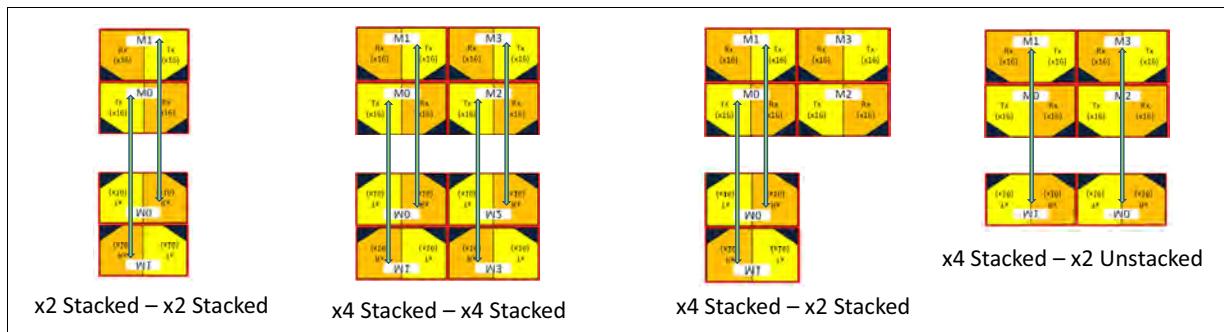


Table 5-28 summarizes the connections between the combinations shown in Figure 5-55, Figure 5-56, and Figure 5-57.

5.7.3.4.1 Module Degrade Rules

Table 5-28. Summary of Standard Package Module Connection Combinations with Different Number of Modules on Both Sides

Standard Package Module Connections (Different # of Modules on Both Sides)	Standard Die Rotate Counterpart ^a	Mirrored Die Rotate Counterpart ^a	
		Option 1 (See Figure 5-56)	Option 2 (See Figure 5-57)
x4 Stacked – x4 Unstacked	<ul style="list-style-type: none"> • M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0 	<ul style="list-style-type: none"> • M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3 	
x4 Stacked – x2 Stacked	<ul style="list-style-type: none"> • M0 – M0 • M1 – M1 • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – M1 • M2 – NC • M3 – NC 	<ul style="list-style-type: none"> • M0 – M1 • M1 – M0 • M2 – NC • M3 – NC
x4 Stacked – x2 Unstacked	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M3 – NC • M2 – M1 	<ul style="list-style-type: none"> • M0 – M1 • M1 – NC • M2 – M0 • M3 – NC 	<ul style="list-style-type: none"> • M0 – NC • M1 – M1 • M2 – NC • M3 – M0
x4 Stacked – x1	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M3 – NC • M2 – NC 	
x4 Unstacked – x2 Unstacked	<ul style="list-style-type: none"> • M0 – M1 • M1 – M0 • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – M1 • M2 – NC • M3 – NC 	
x4 Unstacked – x2 Stacked	<ul style="list-style-type: none"> • M0 – M1 • M1 – M0 • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M1 • M1 – M0 • M2 – NC • M3 – NC 	
x4 Unstacked – x1	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M3 – NC • M2 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC • M3 – NC • M2 – NC 	
x2 Stacked – x2 Unstacked	<ul style="list-style-type: none"> • M0 – M1 • M1 – M0 	<ul style="list-style-type: none"> • M0 – M0 • M1 – M1 	
x2 Stacked – x1	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC 	
x2 Unstacked – x1	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC 	<ul style="list-style-type: none"> • M0 – M0 • M1 – NC 	

a. NC indicates no connection.

On a 2-module or 4-module link, if one or more module-pairs have failed, the link will be degraded and shall comply with the following rules:

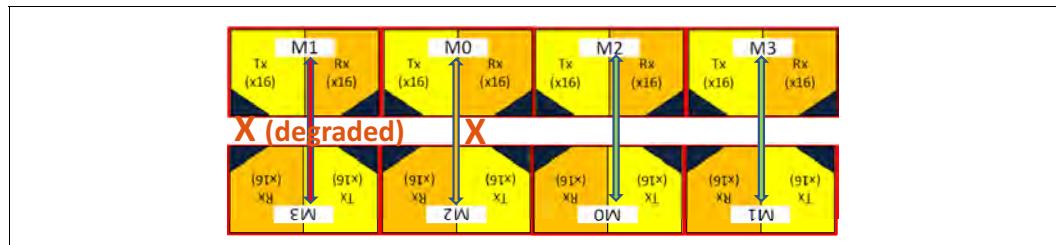
1. The degraded link shall be either one or two modules, and shall not be three modules.
 - a. For a 4-module link:
 - i. If any one module-pair failed, it shall be degraded to a 2-module link.

- ii. If any two module-pairs failed, it shall be degraded to a 2-module link.
 - iii. If any three module-pairs failed, it shall be degraded to a 1-module link.
- b. For a 2-module link:
- i. If any one module-pair failed, it shall be degraded to a 1-module link.
2. For a 4-module link, if only one module-pair failed, one additional module-pair that belongs to the "same half" (along the Die Edge) of the 4-module will be disabled/degraded.

[Figure 5-58](#) illustrates an example with a x4 Unstacked connected to a x4 Unstacked "Standard Die Rotate" counterpart with one M0 – M2 pair failed. The M1 – M3 pair on its left shall be disabled accordingly to comply with the rules defined above, which will be denoted as "x (d)" in [Table 5-29](#).

Note: The double-ended arrows in [Figure 5-58](#) indicate Module-to-Module connections.

Figure 5-58. Example of a Configuration for Standard Package, with Some Modules Disabled



[Table 5-29](#) summarizes the resulting degraded link if there are one, two, or three failed module-pairs for the x4 Unstacked to x4 Unstacked configuration.

Table 5-29. Summary of Degraded Links when Standard Package Module-pairs Fail

Module – Module Partner Pair	Number of Module-pairs Failed ^a													
	1-fail				2-fail						3-fail			
	x	x (d)	✓	✓	x	x	x	✓	✓	✓	x	x	x	✓
M0 – M2	x	x (d)	✓	✓	x	x	x	✓	✓	✓	x	x	x	✓
M1 – M3	x (d)	x	✓	✓	x	✓	✓	x	x	✓	x	x	✓	x
M3 – M1	✓	✓	x	x (d)	✓	x	✓	x	✓	x	x	✓	x	x
M2 – M0	✓	✓	x (d)	x	✓	✓	x	✓	x	x	✓	x	x	x

a. x = Failed Module – Module Partner Pair.

x (d) = Disabled Module – Module Partner Pair to comply with Degrade rules.

✓ = Functional Module – Module Partner Pair.

All other module configurations shall follow the same Module Degrade rules as defined above.

5.7.4 UCIe-S Sideband-only Port

A UCIe-S sideband-only port is also permitted for test/manageability purposes. The RDI signals to the sideband port for a sideband-only configuration are the same as for a sideband with mainband configuration (see [Chapter 10.0](#) for details of the latter).

[Figure 5-59](#) shows the bump map for a UCIe-S sideband-only port. [Figure 5-60](#) shows the supported configurations for a UCIe-S sideband-only port.

Figure 5-59. UCIe-S Sideband-only Port Bump Map

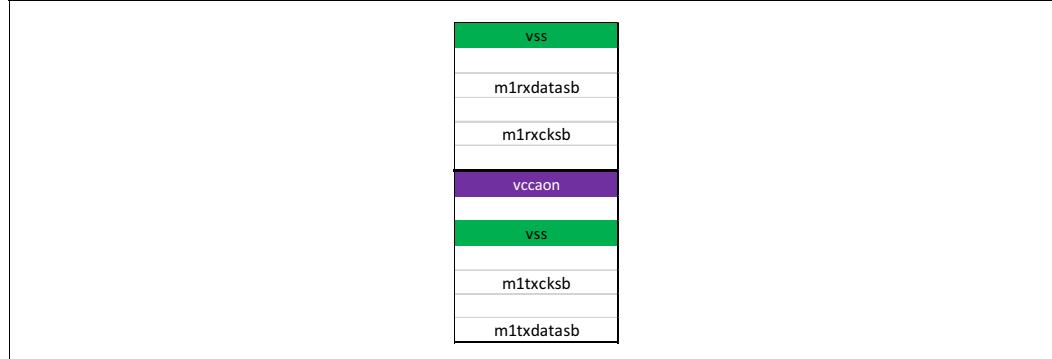
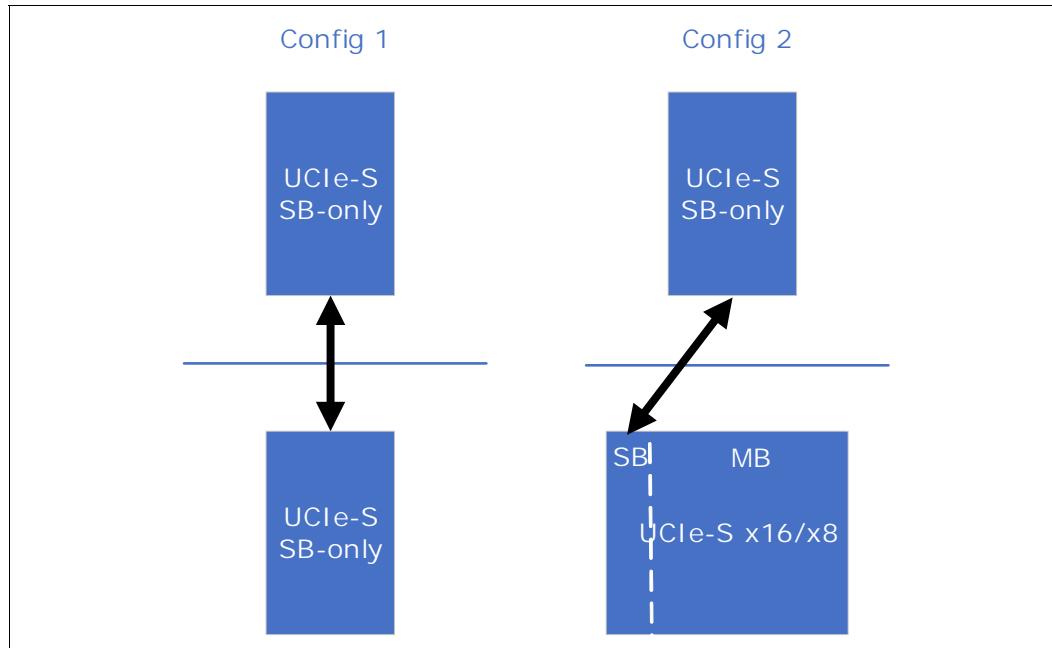


Figure 5-60. UCIe-S Sideband-only Port Supported Configurations



5.8 Tightly Coupled Mode

Tightly Coupled PHY mode is defined as when both of the following conditions are met:

- Shared Power Supply between Tx and Rx, or Forwarded Power Supply from Tx to Rx
- Channel supports larger eye mask defined in [Table 5-30](#)

In this mode, there is no Receiver termination and the Transmitter must provide full swing output. In this mode, further optimization of PHY circuit and power reduction is possible. For example, a tuned inverter can potentially be used instead of a front-end amplifier. Training complexity such as voltage reference can be simplified.

Table 5-30. Tightly Coupled Mode: Eye Mask

Data Rate	4-16 GT/s
Overall (Eye Closure due to Channel) ^a	
Eye Height ^b	250 mV
Eye Width (rectangular eye mask with specified eye height)	0.7 UI

a. With 750-mV Transmitter signal swing.

b. Centered around VCCFWDIO/2.

Loss and crosstalk requirement follow the same VTF method, adjusting to the eye mask defined in [Table 5-30](#). [Table 5-31](#) shows the specification at Nyquist frequency.

Table 5-31. Tightly Coupled Mode Channel for Advanced Package

Data Rate	4-12 GT/s	16 GT/s
VTF Loss ^a (dB)	$L(f_N) > -3$	-
VTF Crosstalk ^a (dB)	$XT(f_N) < 1.5 * L(f_N) - 21.5$ and $XT(f_N) < -23$	-

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

Loss and crosstalk specifications between DC and Nyquist f_N follow the same methodology defined in [Section 5.7.2.1](#).

Although the use of this mode is primarily for Advanced Package, it may also be used for Standard Package when two Dies are near one another and Receiver must be unterminated.

5.9 Interconnect redundancy Remapping

5.9.1 Advanced Package Lane Remapping

Interconnect Lane remapping is supported in Advanced Package Module to improve assembly yield and recover functionality. Each module supports:

- Four redundant bumps for Data
- One redundant bump for Clock and Track
- One redundant bump for Valid

For x64 Advanced Package modules, the four redundant bumps for data repair are divided into two groups of two. [Figure 5-61](#) shows an illustration of x64 Advanced package module redundant bump assignment for data signals. TRD_P0 and TRD_P1 are allocated to the lower 32 data Lanes and TRD_P2 and TRD_P3 are allocated to the upper 32 data Lanes. Each group is permitted to remap up

to two Lanes. For example, TD15 is a broken Lane in the lower half and TD_P32 and TD_P40 are broken Lanes in the upper 32 Lanes. [Figure 5-62](#) illustrates Lane remapping for the broken Lanes.

For x32 Advanced Package modules, only the lower 32 data lanes and TRD_P0 and TRD_P1 apply in [Figure 5-61](#) and [Figure 5-62](#).

Details and implementation of Lane remapping for Data, Clock, Track, and Valid are provided in Section 4.3.

Figure 5-61. Data Lane repair resources

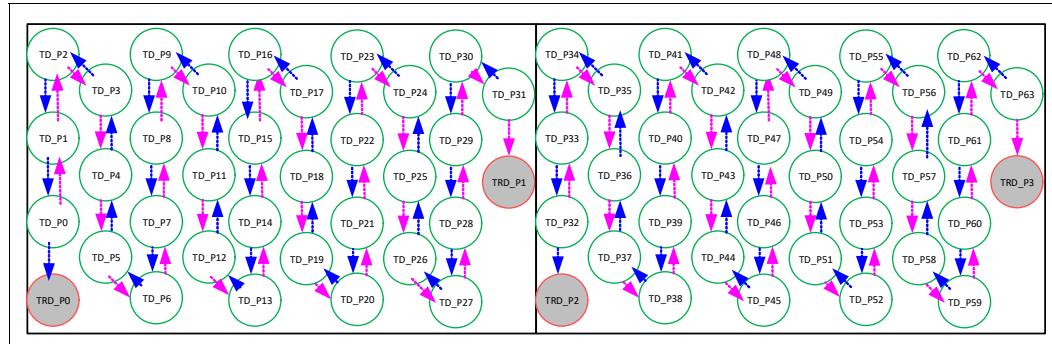
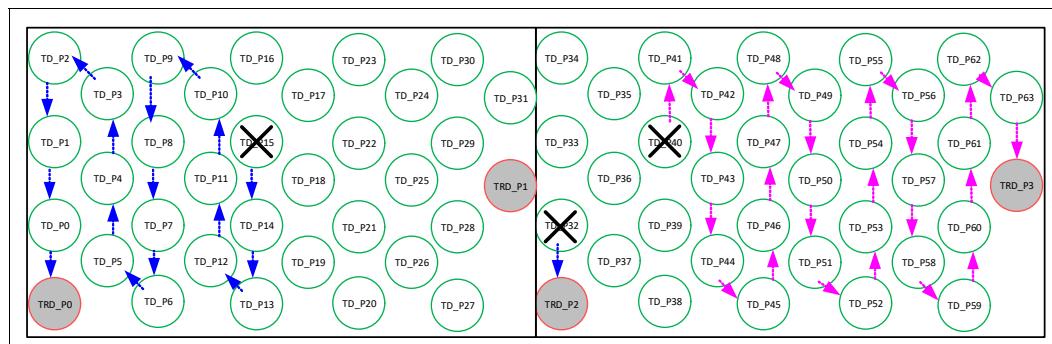


Figure 5-62. Data Lane repair



5.9.2 Standard Package Lane remapping

Lane repair is not supported in Standard Package modules.

5.10 BER Requirements, CRC, and Retry

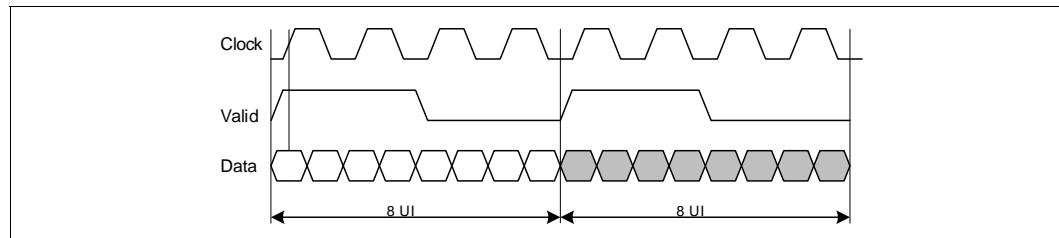
The BER requirement based on channel reach defined in [Section 5.7](#) is shown in [Table 5-32](#). Error detection and correction mechanisms such as CRC and retry are required for BER of 1E-15 or higher to achieve the required Failure In Time (FIT) rate of significantly less than 1 (1 FIT = 1 device failure in 10^9 Hours). The UCIe spec defined CRC and retry is detailed in [Chapter 3.0](#). For the BER of 1E-27, either parity or CRC can be used and the appropriate error reporting mechanism must be invoked to ensure a FIT that is significantly less than 1.

Table 5-32. Raw BER Requirements

Package Type	Data Rate (GT/s)							
	4	8	12	16	24	32	48	64
Advanced Package	1E-27	1E-27	1E-27	1E-15	1E-15	1E-15	1E-15	1E-12
Standard Package	1E-27	1E-27	1E-15	1E-15	1E-15	1E-15	1E-15	1E-12

5.11 Valid and Clock Gating

Valid is used to frame transmit data. For a single transmission of 8 UI data packet, Valid is asserted for the first 4 UI and de-asserted for the second 4 UI. Figure 5-63 shows the transfer of two 8 UI data packets back to back.

Figure 5-63. Valid Framing

As described in Section 4.1.3, clock must be gated only after Valid signal remains low for 16 UI (8 cycles) of postamble clock for half-rate clocking and 32 UI (8 cycles) of postamble clock for quarter-rate clocking, unless free running clock mode is negotiated.

Idle state is when there is no data transmission on the mainband. During Idle state, Data, Clock, and Valid Lanes must hold values as follows:

- If the Link is unterminated, some Data Lane Transmitters are permitted to remain toggling up to the same transition density as the scrambled data without advancing the scrambler state. The remaining Data Lane Transmitters must hold the data of the last transmitted bit. Valid Lane must be held low until the next normal transmission.
 - In Strobe mode, the clock level in a clock-gated state for half-rate clocking (after meeting postamble requirement) must alternate between differential high and differential low during consecutive clock-gating events. For quarter-rate clocking, the clock level in a clock-gated state must alternate between high and low for both phases (Phase-1 and Phase-2) simultaneously. Clock must drive a differential (simultaneous) low for half- (quarter-) rate clocking for at least 1 UI or a maximum of 8 UI before normal operation. The total clock-gated period must be an integer multiple of 8 UI. Example shown in Figure 5-64 and Figure 5-65.
 - In Continuous mode, the clock remains free running (examples shown in Figure 5-66). Total idle period must be an integer multiple of 8 UI.

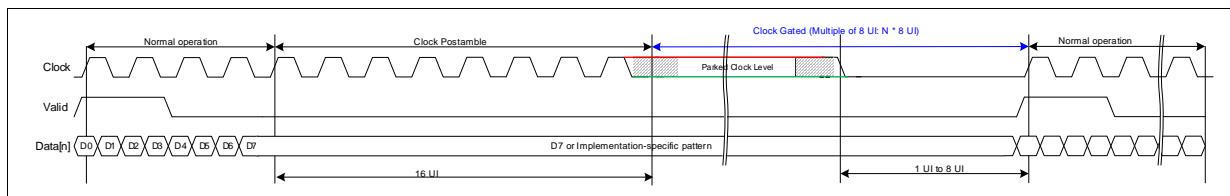
Figure 5-64. Data, Clock, Valid Levels for Half-rate Clocking: Clock-gated Untermminated Link

Figure 5-65. Data, Clock, Valid Levels for Quarter-rate Clocking: Clock-gated Untermminated Link

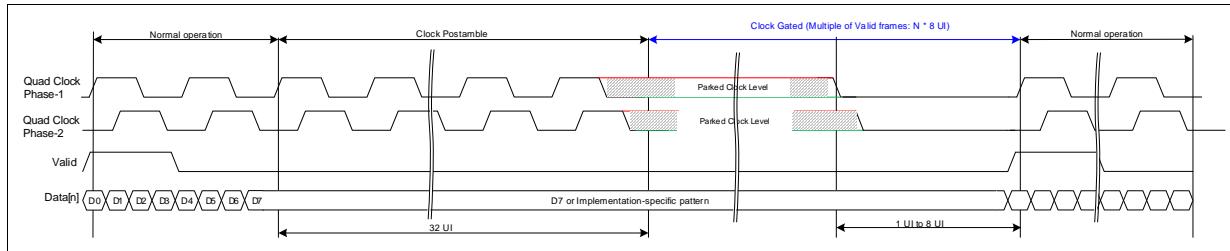
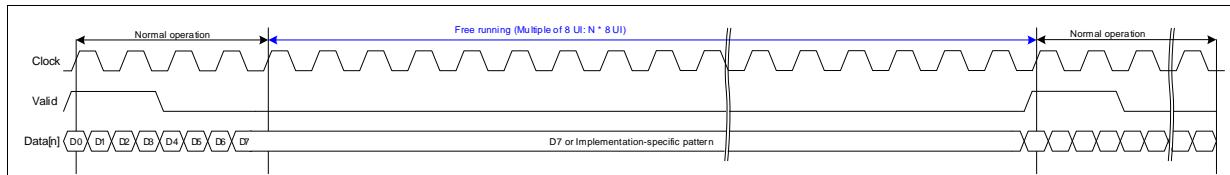


Figure 5-66. Data, Clock, Valid Levels for Half-rate Clocking: Continuous Clock Untermminated Link



- If the Link is terminated, some Data Lane Transmitters are permitted to remain toggling up to the same transition density as the scrambled data without advancing the scrambler state. The remaining Data Lanes' Transmitters hold the data of the last-transmitted bit for 16 UIs under half-rate clocking and 32 UIs under quarter-rate clocking, before transitioning to Hi-Z. Valid Lane must be held low until the next normal transmission. Note that keeping the transmitter toggling will incur extra power penalty and should be applied with discretion.
 - In Strobe mode, the clock level in a clock-gated state for half-rate clocking (after meeting postamble requirement) must alternate between differential high and differential low during consecutive clock-gating events. For quarter-rate clocking, the clock level in a clock-gated state must alternate between high and low for both phases (Phase-1 and Phase-2) simultaneously. Transmitters must precondition the Data Lanes to a 0 or 1 (V) and clock must drive a differential low for at least 1 UI or up to a maximum of 8 UIs for half- (quarter-) rate clocking before the normal transmission. The total clock-gated period must be an integer multiple of 8 UI. Example shown in [Figure 5-67](#) and [Figure 5-69](#).
 - In Continuous mode, the clock remains free running (examples shown in [Figure 5-70](#)). Transmitters must precondition the Data Lanes to a 0 or 1 (V) for at least 1 UI or up to a maximum of 8 UI. Total idle period must be an integer multiple of 8 UI.

Note: Entry into and Exit from Hi-Z state are analog transitions. Hi-Z represents Transmitter state and the actual voltage during this period will be pulled Low due to termination to **ground** at the Receiver.

Figure 5-67. Data, Clock, Valid Gated Levels for Half-rate Clocking: Terminated Link

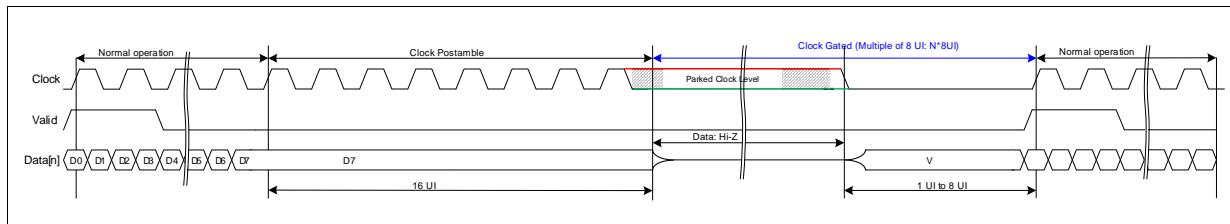


Figure 5-68. Data, Clock, Valid Gated Levels for Quarter-rate Clocking: Terminated Link

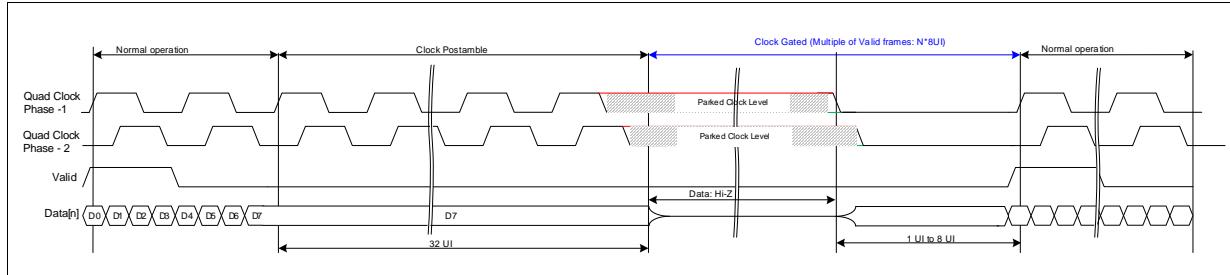
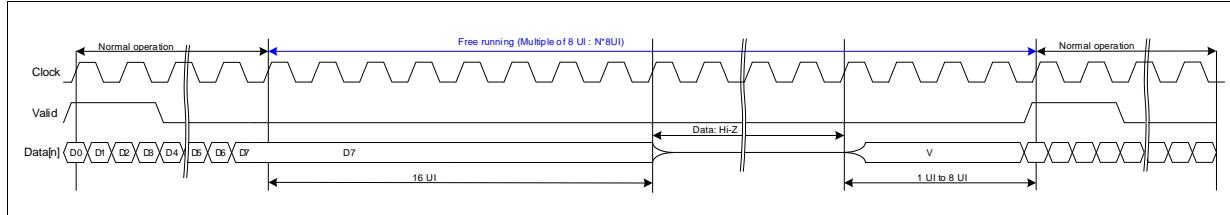


Figure 5-69. Data, Clock, Valid Gated Levels for Half-rate Clocking: Continuous Clock Terminated Link



For data rates of 48 GT/s and 64 GT/s, the forwarded clock lanes must operate in continuous mode. Hamming distance 4 encoding is utilized for Valid Framing, Retimers, and other potential applications. At these data rates, it is required to implement double-error detection and single-error correction for the Valid signal. This requirement aims to minimize the frequency of Link re-training events that are caused by Valid framing errors, while ensuring a sufficiently low probability of failure. Table 4-1 describes the four legal Valid Framing encodings for retimers. For non-retimer links, only two of those encodings are legal. A received value that is one bit different from one of those legal encodings is corrected to that legal encoding. A received value that is two or more bits different from one of those legal encodings is detected as a Valid framing error.

Track operation and Strobe mode:

- Track may be enabled during Link Training (MBTRAIN.RXCLKCAL) and when runtime recalibration is requested
- When Track is enabled, both CLKPN and Track will remain ON until the flow is complete
- Track may be enabled during mainband data transfer or during Electrical Idle state
- If Strobe mode is selected and runtime recalibration is on-going, the clock continues to toggle even if the Idle state is entered, until the flow is complete

5.12 Electrical Idle

Some training states need electrical idle when Transmitters and Receivers are waiting for generate and receive patterns.

- Electrical idle on the mainband in this Specification is described as when Transmitters and Receivers are enabled; Data, Valid and Track Lanes are held low and Clock is parked at high and low.

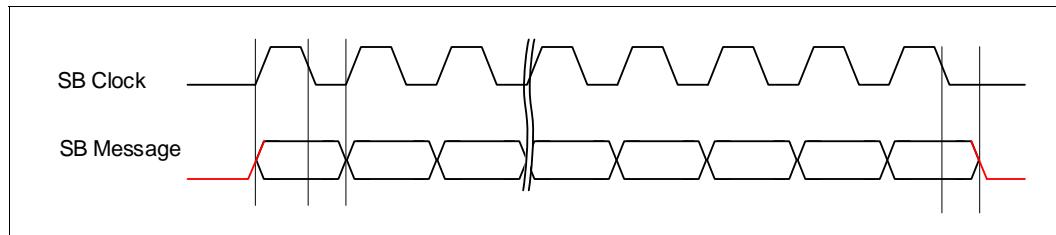
5.13 Sideband signaling

Each module supports a sideband interface. The sideband is a two-signal interface that is used for the transmit and receive directions. The sideband data is an 800 MT/s single data rate signal with an 800-MHz source. Channel reach of the sideband is the same as that of the main band, as defined in [Table 1-1](#) and [Table 1-2](#), unless explicitly stated otherwise in the extended reach setting. The extended reach setting must meet the Tx Driver Output Impedance specified in [Table 5-33](#). Sideband should run on power supply and clock derived from the auxiliary clock (AUXCLK) source which are always on (VCCAON). See [Section 5.13.2](#) for AUXCLK details.

Sideband data is sent edge aligned with the positive edge of the strobe. The Receiver must sample the incoming data with the strobe. The negative edge of the strobe is used to sample the data as the data uses single data rate signaling as shown in [Figure 5-70](#). Sideband transmission is described in [Section 4.1.5](#).

For Advanced Package modules, redundancy is supported for the sideband interface. Sideband initialization and repair are described in [Section 4.5.3.2](#). There is no redundancy and no Lane repair support on Standard Package modules.

Figure 5-70. Sideband signaling



5.13.1 Sideband Electrical Parameters

[Table 5-33](#) shows the sideband electrical parameters.

It is strongly recommended that the two sides of the sideband I/O Link share the same power supply rail.

Table 5-33. Sideband Parameters summary

Parameter	Min	Typ	Max	Unit
Supply voltage (VCCAON) ^a	0.65			V
TX Swing	0.8*VCCAON	—	—	V
Input high voltage (V_{IH})	0.7*VCCAON			V
Input low voltage (V_{IL})			0.3*VCCAON	V
Output high voltage (V_{OH})	0.9*VCCAON			V
Output low voltage (V_{OL})			0.1*VCCAON	V
Sideband Data Setup Time ^b	200	—	—	ps
Sideband Data Hold Time ^b	200	—	—	ps
Rise/Fall time for Advanced Package ^{c, d}	50	—	280	ps
Rise/Fall time for Standard Package ^{e, d}	80	—	175	ps
Extended Reach Channel Length (Standard Package)			100	mm
Tx Driver Output Impedance for Extended Reach ^f			60	Ohms

a. Always On power supply. The guidelines for maximum Voltage presented in [Section 1.5](#) apply to sideband signaling.

- b. The Setup and Hold Times are referenced to the Sideband Data in relation to the falling edge of the Sideband Clock at Sideband Rx.
- c. 20 to 80% of VCCAON level with Advanced Package reference channel load.
- d. This specification applies to sideband of channel length the same as that of the mainband.
- e. 20 to 80% of VCCAON level with Standard Package reference channel load.
- f. This specification is only required for enabling the extended reach sideband. The specification must be met across variations in process, voltage, and temperature.

5.13.2 Auxiliary Clock (AUXCLK)

Auxiliary clock (AUXCLK) may be from any clock source. Although other clock frequencies are possible, it is recommended that every chiplet should also use a 100-MHz clock source. [Table 5-34](#) lists the permitted auxiliary 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. Reference clock (REFCLK; see [Section 5.1.2](#)) can be used if it is always on. Spread-Spectrum Clocking (SSC) is permitted. AUXCLK has relaxed tolerances compared to REFCLK.

Table 5-34. AUXCLK Frequency Parameters

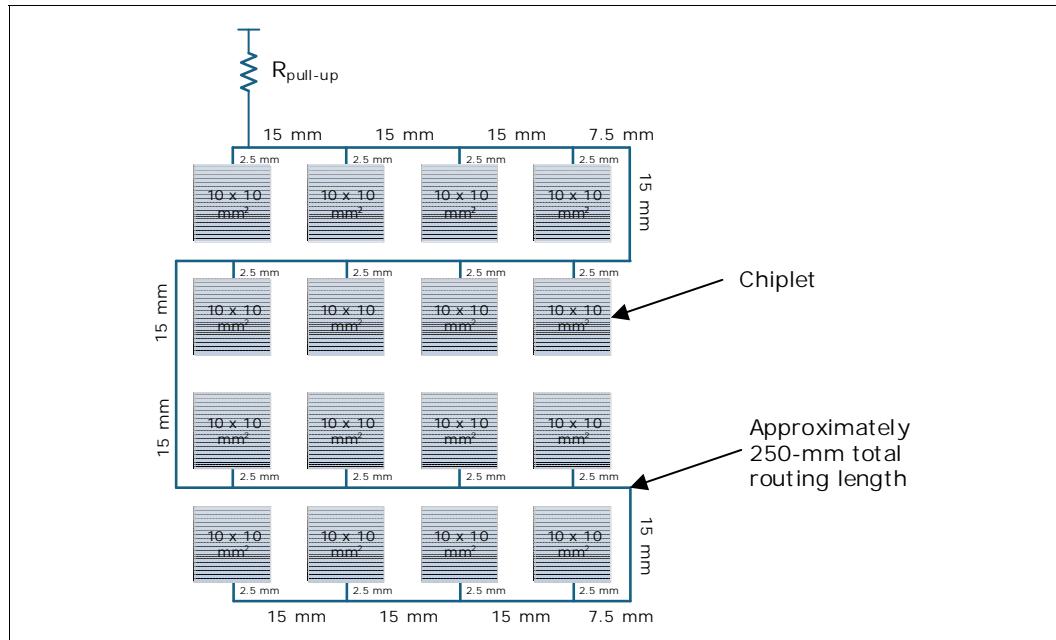
Symbol	Description	Limits			Unit	Notes
		Min	Rec	Max		
F _{AUXCLK}	AUXCLK Frequency	25	100	800	MHz	

5.14 Open Drain

An Open-drain net is resistively pulled up and one or multiple drivers can pull down the net. Any of the drivers can pull down the net such that all sinks will detect the net as low within the maximum fall time. When all drivers have stopped pulling the net low, all sinks will detect the net as high within the maximum rise time.

A chiplet can optionally support an Open Drain pin that connects to a common net. The Open Drain has been specified for a pin connected to a common package route. This pin is not required to be on the same die-edge as the UCIe macro. For an SiP with a small number of chiplets, it is possible for the Open Drain to be routed on the substrate. It is recommended that SiP integrators perform the necessary simulations to ensure functionality.

[Figure 5-71](#) shows an example using Open Drain to connect 16 modules of 10 mm x 10 mm size with 5-mm separation. Note that this example assumes worst-case routing where a default package resistor pull-up is on one end of the Open Drain net. Actual implementations may have more optimized routing compared to the figure. This should allow for meeting the specifications in [Table 5-35](#) more easily.

Figure 5-71. Example Package Route for Open Drain Signal**Table 5-35.** Open Drain Specification Summary

Parameter	Min	Typ	Max	Units	Notes
$R_{\text{pull-down}}$	120	180	240	Ohms	a
$R_{\text{pull-up}}$	5K	7.5K	10K	Ohms	a
Input high voltage (V_{IH})	0.7*VCCAON			V	
Input low voltage (V_{IL})			0.3*VCCAON	V	
Output high voltage (V_{OH})	0.9*VCCAON			V	
Output low voltage (V_{OL})			0.1*VCCAON	V	
Rise time	-	1.5	<2.0	us	b
Fall time	-	30	<100	ns	c
Pin Capacitance (C_{pin})			1	pF	

Note: Requires typical package routing and default single pull-up resistor on package.

- a. Ratio $R_{\text{pull-down}} / (R_{\text{pull-down}} + R_{\text{pull-up}})$ is important to meet V_{OL} . Any combination of Min/Max values meets the requirements.
- b. Rise time to $V_{OH} < 2$ us.
- c. Fall time to $V_{OL} < 100$ ns.

The specification must be met across variations in process, voltage, and temperature.

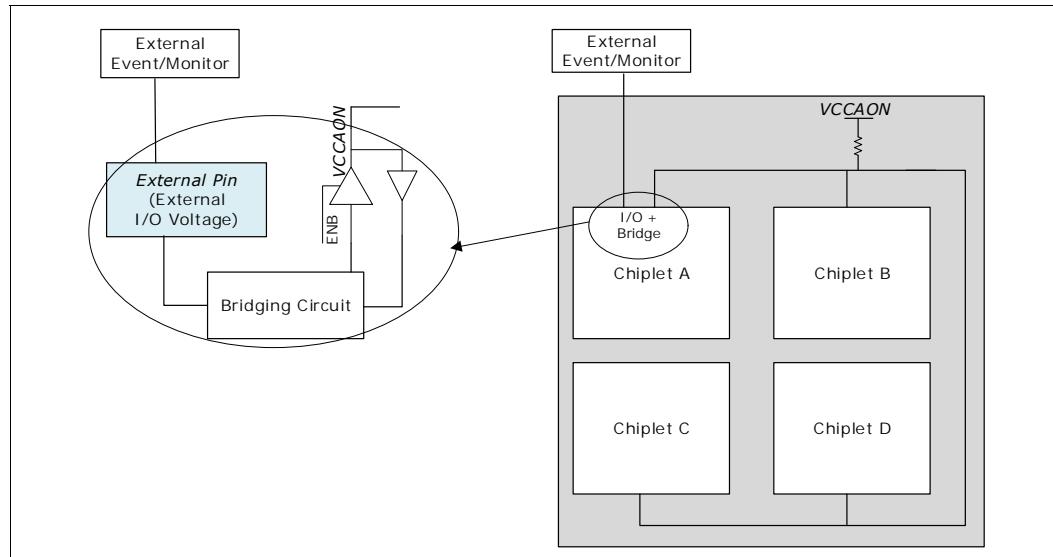
5.14.1 Open Drain Usage

Open Drain pins enable low-latency, bidirectional events. The Open Drain signals are used for some specified PCIe low-latency events (see [Section 8.4.2.1](#) and [Section 8.4.2.2](#)), as well as vendor-defined events.

5.14.2 External Open Drain Connections

The Open Drain specification is recommended to use for in-package routing. Systems that require external event connections are recommended to use a separate pin for external event communications. In this case, it is expected that one chiplet provides any associated bridging between the internal Open Drain and the external event pin. Figure 5-72 shows an example of bridging between internal Open Drain and external event pin. Although not shown in Figure 5-72, it is possible that the external event pin is also bidirectional. Details of external Open Drain implementation are beyond the scope of this specification.

Figure 5-72. Example Bridging between Internal and External Event Pin



IMPLEMENTATION NOTE

Bridging Circuit

If the event being communicated indicates that we are approaching or exceeding standard operating conditions, it is advisable that the bridging circuitry be designed to maintain functionality beyond standard operating conditions.

§ §

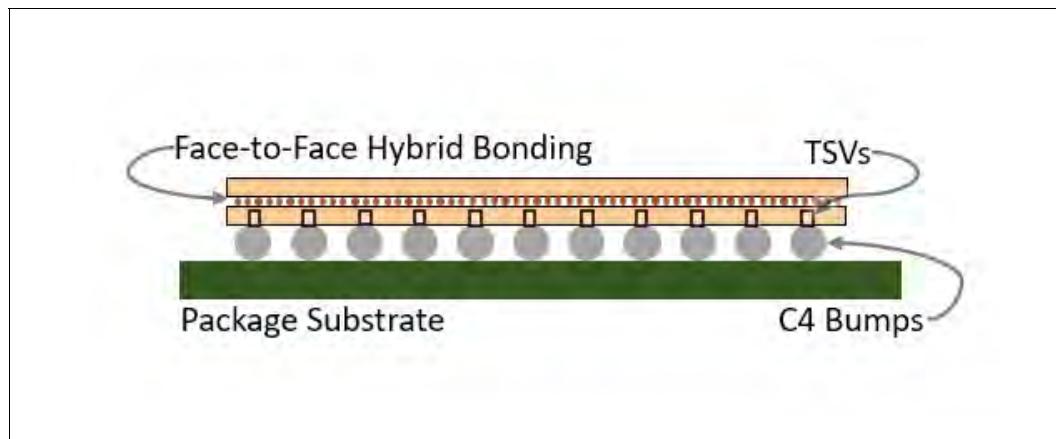
6.0 UCIE-3D

6.1 Introduction

Three-dimensional heterogeneously integrated technologies present an opportunity for the development of new electronic systems with advantages of higher bandwidth and lower power as compared to 2D and 2.5D architectures. 3D will enable applications where the scale of data movement is impractical for monolithic, 2D, or 2.5D approaches.

Universal Chiplet Interconnect express for 3D packaging (UCIE-3D) is designed as a universally applicable interface for 3D die-to-die communication. [Figure 6-1](#) illustrates an example of two dies stacked in a 3D configuration. UCIE-3D uses a two-dimensional array of interconnect bumps for data transmission between dies.

Figure 6-1. Example of 3D Die Stacking



6.2 UCIE-3D Features and Summary

While the UCIE 2D and 2.5D models strive for seamless plug-and-play interoperability, the UCIE-3D model necessitates a more-integrated approach due to the inherent characteristics of packaging technology. The objective is to offer a range of options or a “menu” from which users can select what best suits their needs. The primary objectives and general methodology for UCIE-3D are as follows:

- Circuit and logic must fit within the bump area (i.e., UCIE will continue to be bump-limited). Given the high density, this will translate to lower operating frequencies and a much-simplified circuit (e.g., at 1- μm bump-pitch, the UCIE-3D area amortized on a per-lane basis must be less than 1 μm^2).
- No D2D adapter. Low BER due to low-frequency and almost zero-channel distance — No CRC/replay is needed.
- A hardened minimal PHY such as a simple inverter/driver. The SoC Logic connects directly to the PHY.

- All debug/testability hooks are located within a common block (across all UCIE-3D Links) that is connected to the SoC Logic network inside the chiplet.
- Lane repair becomes a bundle-wide repair that is orchestrated by the SoC Logic.

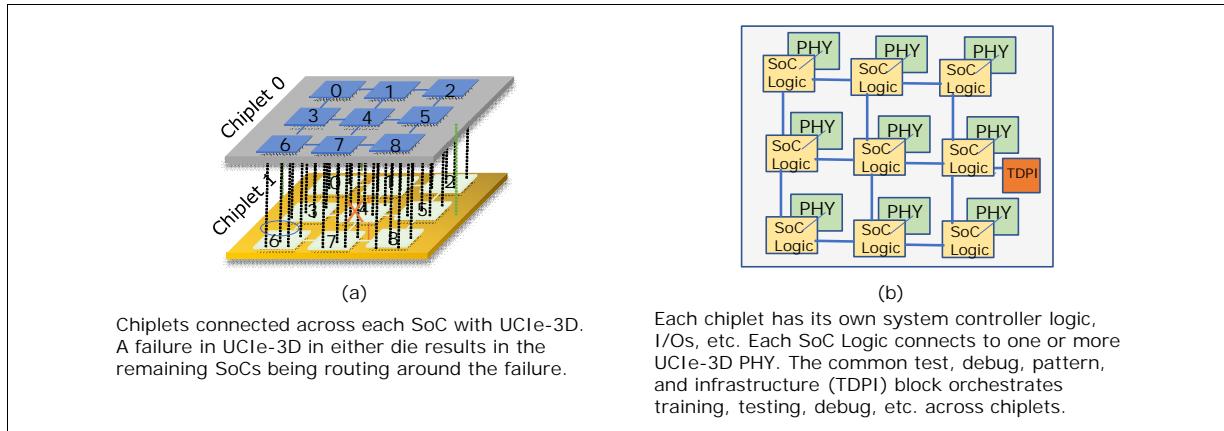
Figure 6-2. UCIE-3D Illustration

Table 6-1 summarizes the key performance indicators of the proposed UCIE-3D.

Table 6-1. UCIE-3D Key Performance Indicators (Sheet 1 of 2)

Characteristics/KPIs	UCIE-S	UCIE-A	UCIE-3D	Comments for UCIE-3D
Characteristics				
Data Rate (GT/s)	4, 8, 12, 16, 24, 32, 48, 64		up to 4	<ul style="list-style-type: none"> • Equal to SoC Logic Frequency — power efficiency is critical
Width (each cluster)	16	64	80	<ul style="list-style-type: none"> • Options of reduced width to 70, 60, ...
Bump Pitch (um)	100 to 130	25 to 55	≤ 10 (optimized) > 10 to 25 (functional)	<ul style="list-style-type: none"> • Must scale such that UCIE-3D fits within the bump area • Must support hybrid bonding
Channel Reach (mm)	≤ 25	≤ 2	3D vertical	<ul style="list-style-type: none"> • F2F bonding initially; F2B, B2B, multi-stack possible
Target for Key Metrics				
BW Die Edge (GB/s/mm)	28 to 370	165 to 2634	N/A (vertical)	
BW Density (GB/s/mm ²)	22 to 192	188 to 1646	4,000 at 9 um	<ul style="list-style-type: none"> • 4 TB/s/mm² at 9 um • Approximately 12 TB/s/mm² at 5 um • Approximately 35 TB/s/mm² at 3 um • Approximately 300 TB/s/mm² at 1 um
Power Efficiency Target (pJ/b)	0.50 to 0.75	0.25 to 0.50	< 0.05 at 9 um	<ul style="list-style-type: none"> • Conservatively estimated at 9-um pitch • < 0.02 for 3-um pitch
Low-power Entry/Exit	0.5 ns at ≤ 16 GT/s 0.5 ns to 1 ns at ≥ 24 GT/s		0 ns	<ul style="list-style-type: none"> • No preamble or postamble

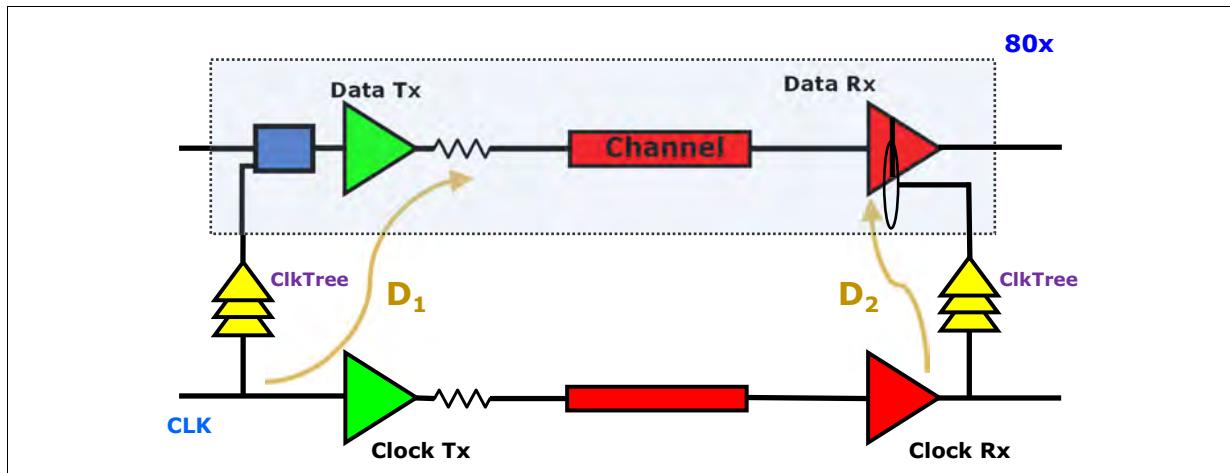
Table 6-1. UCIE-3D Key Performance Indicators (Sheet 2 of 2)

Characteristics/KPIs	UCIE-S	UCIE-A	UCIE-3D	Comments for UCIE-3D
Latency (Tx + Rx)	< 2 ns (PHY + Adapter) ≤ 0.75 ns (PHY 16 GT/s)	0.125 ns at 4 GT/s		<ul style="list-style-type: none"> 0.5 UI, half of flop to flop
Reliability (FIT ^a)	0 < FIT << 1			<ul style="list-style-type: none"> BER < 1E-27
ESD	30-V CDM	5-V CDM → ≤ 3 V		<ul style="list-style-type: none"> 5-V CDM at introduction No ESD for wafer-to-wafer hybrid bonding possible

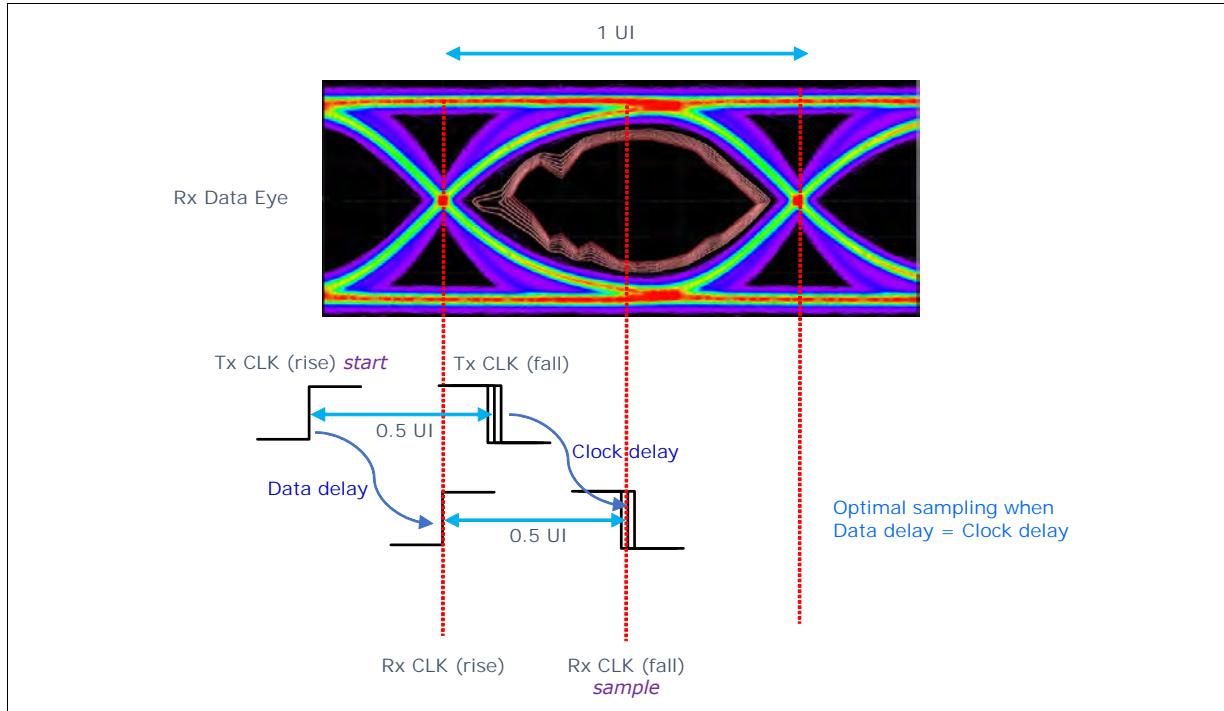
a. FIT = Failure in Time.

6.3 UCIE-3D Tx, Rx, and Clocking

Figure 6-3 presents the Transceiver (Trx) architecture of UCIE-3D. A matched architecture as shown in Figure 5-4, Figure 5-9, and Figure 5-15 offers optimal supply noise rejection. However, this comes at the cost of increased power consumption. The architecture depicted in Figure 6-3 circumvents this power penalty while maintaining the same level of supply noise rejection. The UCIE-3D specification will establish target values and tolerances for clock distribution delays D_1 and D_2 .

Figure 6-3. UCIE-3D PHY

It is important to highlight that UCIE-3D uses a rise-to-fall timing approach, differing from the typical on-die logic design that uses a rise-to-rise timing approach. The primary distinction between these two scenarios is that on-die logic must factor in the delay caused by combinational logic, whereas UCIE-3D features matched data and clock buffer delays, resulting in a near-zero differential. As depicted in Figure 6-4, rise-to-fall timing yields the most-optimal timing margin for zero-delay differential.

Figure 6-4. Start Edge and Sample Edge

6.4 Electrical Specification

6.4.1 Timing Budget

Consideration of various factors such as jitter, noise, mismatch, and error terms are crucial for the link timing budget. [Table 6-2](#) outlines the PCIe-3D specification parameters that are pertinent to link timing. Pulse width deviation from 50% clock period includes both static error (duty-cycle error) and dynamic error (pulse-width jitter). Lane-to-lane skews account for the variation between data lanes, and Data/Clock differential delays account for the clock to center of distribution of data lanes.

Table 6-2. Timing and Mismatch Specification (Sheet 1 of 2)

Specification	Name	Min	Typ	Max	Unit	UI = 250 ps at 4 GT/s	Note
Eye Closure due to Channel	C_h		0.1		UI	25 ps	a
Pulse-width Deviation from 50% Clock Period	J_{pw}		0.08		UI pk-to-pk	20 ps	
Tx Lane-to-Lane Skew	S_{tx}		0.12		UI pk-to-pk	30 ps	
Rx Lane-to-Lane Skew	S_{rx}		0.12		UI pk-to-pk	30 ps	
Tx Data/Clock Differential Delay	D_{tx}	D_{tx_min}	D_{tx_typ}	D_{tx_max}	ps	max – min = 50 ps	b
Rx Data/Clock Differential Delay	D_{rx}	D_{rx_min}	D_{rx_typ}	D_{rx_max}	ps	max – min = 50 ps	
Alpha Factor (Tx and Rx)	α_{tx}			1.5			c
Vcc Noise	n_{vcc}			10	% pk-to-pk		d

Table 6-2. Timing and Mismatch Specification (Sheet 2 of 2)

Specification	Name	Min	Typ	Max	Unit	UI = 250 ps at 4 GT/s	Note
Tx Data/Clock Differential RJ	J _{rtx}			0.05	UI pk-to-pk at BER	12.5 ps	
Rx Data/Clock Differential RJ	J _{rrx}			0.05	UI pk-to-pk at BER	12.5 ps	
Sampling Aperture	A _p			0.03	UI	7.5 ps	

- a. Eye closure due to channel includes inter-symbol interference (ISI) and crosstalk.
b. Defined as clock to mean data, min/typ/max values are shown below.

c. Alpha factor is defined as follows for Tx and Rx, respectively:

$$\alpha_{Tx} = \frac{dD_{Tx}}{D_{Tx}} / \frac{dV_{CC}}{V_{CC}} \quad \alpha_{Rx} = \frac{dD_{Rx}}{D_{Rx}} / \frac{dV_{CC}}{V_{CC}}$$

- d. This is equivalent to a variation of $\pm 5\%$ in V_{CC}. Careful mitigation is particularly needed when disturbances external to UCIe occur, such as electromagnetic coupling from through-silicon vias (TSVs).

Parameters D_{tx} and D_{rx} are V_{CC}-dependent functions. [Equation 6-1](#) defines their typical values.

Equation 6-1.

$$D_{tx_typ} = D_{rx_typ} = \frac{V_{CC}}{0.0153 V_{CC}^2 + 0.0188 V_{CC} - 0.0084}$$

where, unit of D_{tx_typ} and D_{rx_typ} is ps and unit of V_{CC} is V.

[Equation 6-2](#) and [Equation 6-3](#) define the minimum spec curve of D_{tx} and D_{rx}, respectively.

Equation 6-2.

$$D_{tx_min} = \max(D_{tx_typ} - 0.08 UI, 0)$$

Equation 6-3.

$$D_{rx_min} = \max(D_{rx_typ} - 0.08 UI, 0)$$

[Equation 6-4](#) and [Equation 6-5](#) define the maximum spec curve of D_{tx} and D_{rx}, respectively.

Equation 6-4.

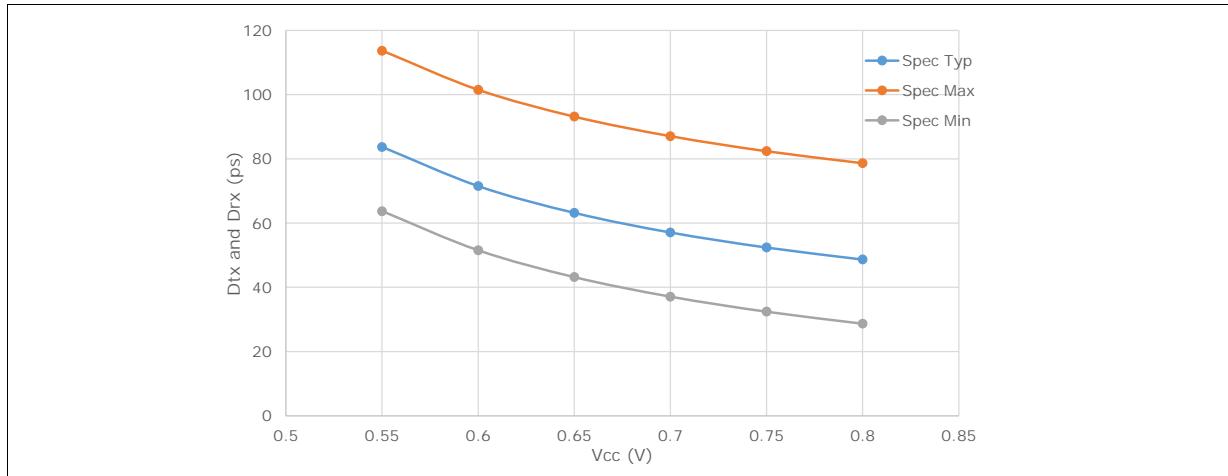
$$D_{tx_max} = D_{tx_typ} + 0.12 UI$$

Equation 6-5.

$$D_{rx_max} = D_{rx_typ} + 0.12 UI$$

Figure 6-5 illustrates a plot of the spec range for 4 GT/s.

Figure 6-5. Dtx and Drx Spec Range for 4 GT/s



The equation for delay time, derived from the general theory of buffer chain, incorporates a term proportional to Vcc and a quadratic Vcc dependence in the denominator. This equation is fitted to a specific process and design. A typical design is expected to have the same trend, and remain within the boundaries of the upper and lower curves. It is not required to align with the central curve.

Equation 6-6 is essential in closing the timing budget, subsequently leading to the defined specification limit.

Equation 6-6.

$$C_h + J_{pw} + S_{tx} + S_{rx} + \sqrt{J_{rtx}^2 + J_{rrx}^2} + A_p + [max(D_{tx}) - min(D_{tx}) + max(D_{rx}) - min(D_{rx})] (1 + \alpha_{tx}n_{vcc}) < 1 UI$$

When there is a change in Vcc, as in the case of a dynamic voltage frequency scaling (DVFS) scenario, the specification range for D_{tx} and D_{rx} adjusts correspondingly. This offers a degree of design flexibility because the delay does not need to conform to a fixed band across the entire Vcc range. Given that the range from maximum to minimum remains constant, the timing margin remains unaffected.

6.4.2 ESD and Energy Efficiency

Data and clock signals shall comply with a mask on an eye diagram that specifies the following:

- Minimum voltage swing
- Minimum duration during which the output voltage will be stable
- Maximum permitted overshoot and undershoot

The Tx output swing range is between 0.40 V and 0.75 V.

Table 6-3 defines the ESD targets.

Table 6-3. ESD Specification for ≤ 10 um Bump Pitch

Parameter	Minimum
Discharge voltage (CDM)	5 V
Discharge peak current	40 mA

The feasibility of 0-V ESD should be explored for the special case of wafer-to-wafer hybrid bonding. For more details, see the *Industry Council on ESD Targets white papers*.

For > 10 um to < 25 um bump pitches, higher ESD can be permitted. The exact target will be published in a future revision of the specification.

Table 6-4 lists the Energy Efficiency targets.

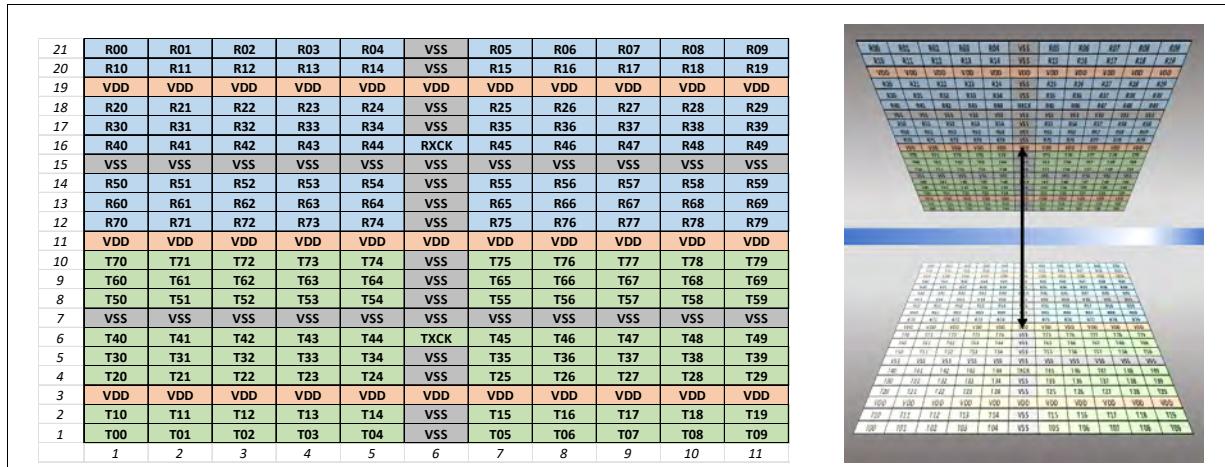
Table 6-4. Energy Efficiency Target

Bump Spacing (um)	Energy Efficiency at 4 GT/s (pJ/bit)
9	0.05
3	0.02
1	0.01
9 to 25	To be published in a future revision of the specification.

6.4.3 UCIE-3D Module and Bump Map

Figure 6-6 depicts a potential bump map for UCIE-3D. The arrangement of the signals is such that the same PHY can be utilized on both the top die and bottom die. The unit used in Figure 6-6 is the bump pitch. The estimated area for a x80 module (encompassing both Tx and Rx) in a 9-um pitch is approximately 0.02 mm². It is important to note that the area scales with the square of the bump pitch.

Figure 6-6. UCIE-3D Module Bump Map



The UCIE-3D standard does not prescribe a mandatory bump pitch; however, a 9-um pitch is recommended at introduction. As the technology advances, additional specific recommended pitch values will be established.

Although UCIE-3D does not inherently predefine an adapter, users have the flexibility to allocate some data lanes within the module for adapter functions as required, such as Valid, Data Mask, Parity, and ECC. UCIE-3D does not necessitate a sideband for initialization. If a low-bandwidth data link similar to sideband is required, it is up to the implementation to determine how to assign a group of lanes for the purpose. Bit replication or other forms of redundancy can be used to guarantee link reliability.

If modules are physically adjacent, extra VDDs can be added between them to provide physical separation, shielding, and additional power delivery.

Along with x80, the bump map of x70 Module is depicted in [Figure 6-7](#). Bump maps of additional Module widths may be incorporated in a future update to this specification if needed, using similar layout.

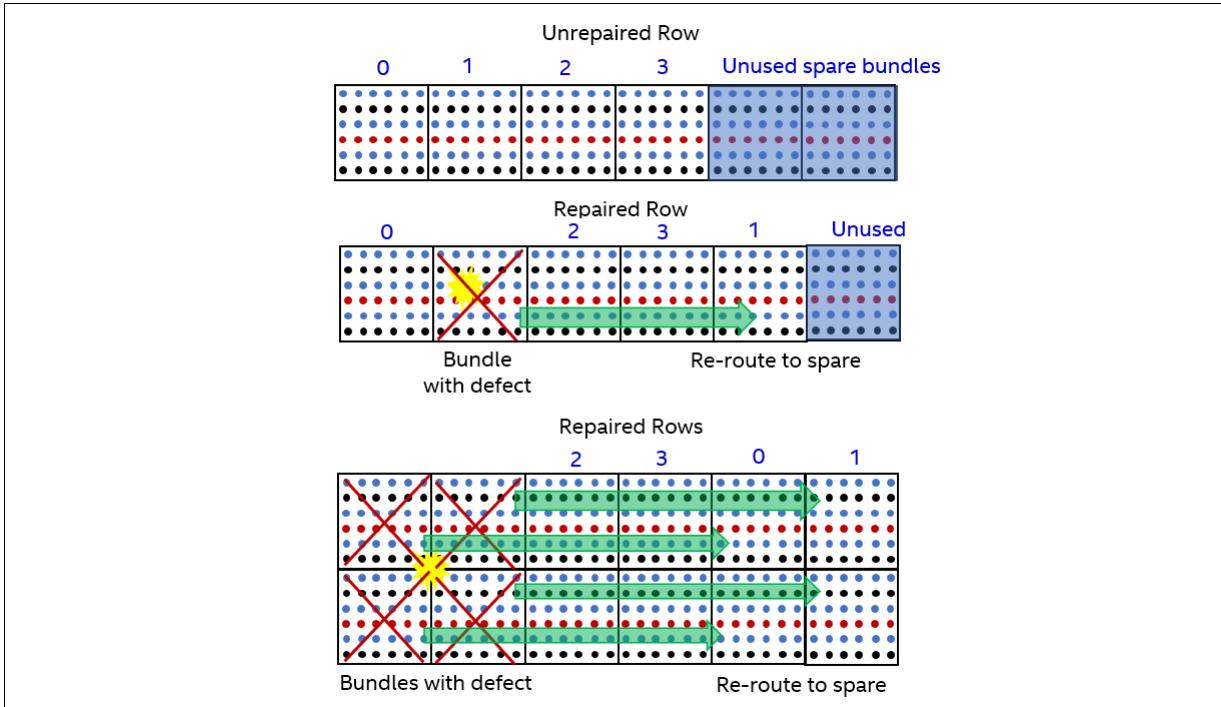
Figure 6-7. x70 Module

19	R00	R01	R02	R03	R04	VSS	R05	R06	R07	R08	R09
18	R10	R11	R12	R13	R14	VSS	R15	R16	R17	R18	R19
17	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
16	R20	R21	R22	R23	R24	VSS	R25	R26	R27	R28	R29
15	R30	R31	R32	R33	R34	VSS	R35	R36	R37	R38	R39
14	R40	R41	R42	R43	R44	RXCK	R45	R46	R47	R48	R49
13	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
12	R50	R51	R52	R53	R54	VSS	R55	R56	R57	R58	R59
11	R60	R61	R62	R63	R64	VSS	R65	R66	R67	R68	R69
10	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
9	T60	T61	T62	T63	T64	VSS	T65	T66	T67	T68	T69
8	T50	T51	T52	T53	T54	VSS	T55	T56	T57	T58	T59
7	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
6	T40	T41	T42	T43	T44	TXCK	T45	T46	T47	T48	T49
5	T30	T31	T32	T33	T34	VSS	T35	T36	T37	T38	T39
4	T20	T21	T22	T23	T24	VSS	T25	T26	T27	T28	T29
3	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
2	T10	T11	T12	T13	T14	VSS	T15	T16	T17	T18	T19
1	T00	T01	T02	T03	T04	VSS	T05	T06	T07	T08	T09
	1	2	3	4	5	6	7	8	9	10	11

6.4.4 Repair Strategy

Defect size (more exactly, Si area impacted by a single defect) is defined by a probability distribution. The size is influenced by factors such as numbers of I/Os in SoC, packaging technology used, and bump pitch. A standard needs to cover technologies from multiple companies, scalable to future bump pitches, as well as different SoC sizes. Lane repair based on fixed defect size is not practical for an effective standard.

Given these considerations, a bundle repair strategy is proposed for UCIE-3D. This involves reserving bundles within the SoC for repair purposes, which can be rerouted to serve as backup in the event of a failure, as illustrated in [Figure 6-8](#). The figure shows the cases of no repair, 1-bundle repair, and 4-bundle repairs. For a densely packed 2D UCIE Module array, it is recommended to reserve two full Modules (equivalent to four bundles) to repair a single failure. This assumes an alternating arrangement of Tx and Rx bundles in at least one direction. Each Module is equipped with one Tx bundle (comprising a x80 Tx + Clock) and one Rx bundle (comprising a x80 Rx + Clock).

Figure 6-8. Bundle Repair

To scale the general case of a large number of UCIE links, the following mathematical model can be used to compute the repair requirements:

Parameters:

- D_0 represents the defect density of the interconnect, expressed in terms of the number of failures per unit area
- A signifies the total UCIE-3D area of the chip
- δ denotes the acceptable yield loss

The model suggests reserving $2k$ full Modules, where k is determined by the subsequent equation.

Equation 6-7.

$$1 - \sum_{i=0}^k P_i(AD_0) < \delta$$

Equation 6-8.

$$P_i(x) = \frac{x^i}{i!} e^{-x}$$

The calculations in [Equation 6-7](#) and [Equation 6-8](#) assume that large interconnect defects that are comparable to bundle size are relatively rare. More spare bundles may be needed if density of large defects exceeds a limit such that [Equation 6-9](#) does not hold.

Equation 6-9.

$$1 - e^{-AD_1} < \delta$$

where, D_1 is the density of defects with diameter greater than the bundle dimension. The exact amount can be determined by simulation.

When UCIE-3D links are not densely packed, strategic placement of spacing between bundles can effectively reduce the number of repair bundles required. For example, with sufficient spacing between rows, the occurrence of a single defect eliminating four bundles can be prevented. However, the precise determination of this spacing is highly dependent on the specific technology in use, and thus, falls beyond the scope of this specification. The specification merely highlights this as a potential option.

The initiation of repair is anticipated to originate from the SoC Logic, which is external to the UCIE-3D PHY, and therefore is not elaborated on in this context. The implementation can be specific to the system.

6.4.5 Channel and Data Rate Extension

While the immediate focus of UCIE-3D is on Face-to-Face hybrid bonding, the proposed architecture is designed to be adaptable for Face-to-Back, Back-to-Back, and multi-stack configurations. Comprehensive channel and circuit simulations are necessary to determine the optimal data rate for these scenarios. Reduction of 10% or less in data rate is expected for Face-to-Back and Back-to-Back configurations.

§ §

7.0 Sideband

7.1 Protocol Specification

The usage for the sideband Link is to provide a out of band channel for Link training and an interface for sideband access of registers of the Link partner. It is also used for Link Management Packets and parameter exchanges with remote Link partner.

The same protocol is also used for local die sideband accesses over FDI and RDI. When relevant, FDI specific rules are pointed out using "FDI sideband:". When relevant, RDI specific rules are pointed out using "RDI sideband:". When relevant, PCIe Link specific rules are pointed out using "PCIe Link sideband:". If no prefix is mentioned, it is a common rule across FDI, RDI and PCIe Link.

The Physical Layer is responsible for framing and transporting sideband packets over the PCIe Link. Direct sideband access to remote die can originate from the Adapter or the Physical Layer. The Adapter forwards a remote die sideband access over RDI to the Physical Layer for framing and transport. These include register access requests, completions or messages.

The Protocol Layer has indirect access to remote die registers using the sideband mailbox mechanism. The mailbox registers reside in the Adapter, and it is the responsibility of the Adapter to initiate remote die register access requests when it receives the corresponding access trigger for the mailbox register over FDI.

FDI sideband: In the case of multi-protocol stacks, the Adapter must track which protocol stack sent the original request and route the completion back to the appropriate protocol stack.

FDI sideband: Because the Protocol Layer is only permitted indirect access to remote die registers, and direct access to local die registers, currently only Register Access requests and completions are permitted on the FDI sideband.

All sideband requests that expect a response have an 8ms timeout. A "Stall" encoding is provided for the relevant packets for Retimers, to prevent timeouts if the Retimer needs extra time to respond to the request. When stalling to prevent timeouts, it is the responsibility of the Retimer to send the corresponding Stall response once every 4ms. The Retimer must also ensure that it does not Stall indefinitely, and escalates a Link down event after a reasonable attempt to complete resolution that required stalling the requester. If a requester receives a response with a "Stall" encoding, it resets the timeout counter.

In certain cases, it is necessary for registers to be fragmented between the different layers; i.e., certain bits of a given register physically reside in the Protocol Layer, other bits reside in the Adapter, and other bits reside in the Physical Layer. PCIe takes a hierarchical decoding for these registers. For fragmented registers, if a bit does not physically reside in a given Layer, it implements that bit as Read Only and tied to 0. Hence reads would return 0 for those bits from that Layer, and writes would have no effect on those bits. As an example, for reads, Protocol Layer would forward these requests to the Adapter on FDI and the Protocol Layer will OR the data responded by the Adapter with its local register before responding to software. The Adapter must do the same if any bits of that register reside in the Physical Layer before responding to the Protocol Layer.

7.1.1 Sideband Packet Types

Five different categories of sideband packets are permitted:

- Register Accesses: These can be Configuration (CFG) or Memory Mapped accesses for both Reads or Writes are supported. These can be associated with 32b of data or 64b of data. All register accesses (Reads or Writes) have an associated completion.
- Messages without data: These can be Link Management (LM), or Vendor Defined Packets. These do not carry additional data payloads.
- Messages with data: These can be Parameter Exchange (PE), Link Training related or Vendor Defined, and carry 64b of data.
- Management Transport Messages: If Management Transport protocol is supported, Management Transport Messages with data or without data are supported (see [Section 7.1.2.4](#) and [Section 7.1.2.5](#), respectively).
- Priority Sideband Traffic Packets (PSTP): These are sideband packets that carry priority messages to the remote Link partner (see [Section 4.1.5.2](#) for the rules related to transmitting these sideband packets, and [Figure 7-14](#) for the format of a PSTP).

Every sideband packet type carries a 5-bit opcode. Every sideband packet type, with the exception of a PSTP, also carries a 3-bit source identifier (srcid), and a 3-bit destination identifier (dstid). The 5-bit opcode indicates the sideband packet type, as well as whether the sideband packet carries no data, 32b of data or 64b of data.

[Table 7-1](#) gives the mapping of opcode encodings to the Sideband Packet Types.

Table 7-1. Sideband Packet Opcode Encodings Mapped to Sideband Packet Types

Opcode Encoding	Sideband Packet Type	Opcode Encoding	Sideband Packet Type
00000b	32b Memory Read	01101b	64b Configuration Write
00001b	32b Memory Write	10000b	Completion without Data
00010b	32b DMS Register Read	10001b	Completion with 32b Data
00011b	32b DMS Register Write	10010b	Message without Data
00100b	32b Configuration Read	10111b	Management Port Messages without Data
00101b	32b Configuration Write	11000b	Management Port Message with Data
01000b	64b Memory Read	11001b	Completion with 64b Data
01001b	64b Memory Write	11011b	Message with 64b Data
01010b	64b DMS Register Read	11110b	Priority Packet (see Section 4.1.5.2 for details) ^a
01011b	64b DMS Register Write	11111b	Priority Packet (see Section 4.1.5.2 for details) ^a
01100b	64b Configuration Read	Other encodings	Reserved

a. The Priority Packet headers are of a different size compared to other sideband packets, which causes the parity bit location to be different for them. With a BER of 1E-27 on the sideband Link, it is required to prevent single bit errors on the Link from causing aliasing between priority packets and other sideband packets. This relies on the opcode field plus bit [5] and bit [6] requiring more than one bit flip to change between priority packet format and non-priority packet format types. Future packet changes/ additions must take this into account and prevent aliasing for any new packet additions/changes.

[Table 7-2](#), [Table 7-3](#), and [Table 7-4](#) give the encodings of source and destination identifiers. It is not permitted for Protocol Layer from one side of the Link to directly access Protocol Layer of the remote Link partner over sideband (this should be done via mainband).

Table 7-2. FDI sideband: srcid and dstid encodings on FDI

Field ^a	Description
srcid[2:0]	000b: Stack 0 Protocol Layer 100b: Stack 1 Protocol Layer other encodings are reserved.
dstid[2:0]	001b: D2D Adapter 010b: Physical Layer other encodings are reserved.

- a. srcid and dstid are Reserved for completion messages transferred over FDI. The Protocol Layer must correlate the completions to original requests using the Tag field. Currently, no requests are permitted from Adapter to Protocol Layer over FDI sideband.

Table 7-3. RDI sideband: srcid and dstid encodings on RDI

Field ^a	Description
srcid[2:0]	000b: Stack 0 Protocol Layer 001b: D2D Adapter 011b: Management Port Gateway (see Section 8.2) 100b: Stack 1 Protocol Layer other encodings are reserved.
dstid[2]	0b: Local die terminated request 1b: Remote die terminated request
dstid[1:0]	For Local die terminated requests: 10b: Physical Layer other encodings are reserved. For Remote die terminated Register Access Requests: dstid[1:0] is Reserved For Remote die terminated Register Access Completions: 01b: D2D Adapter other encodings are reserved. For Remote die terminated messages: 01b: D2D Adapter message 10b: Physical Layer message 11b: Management Port Gateway message (see Section 8.2)

- a. srcid and dstid are Reserved for completion messages transferred over RDI for local Register Access completions. For Register Access completions, the Adapter must correlate the completions to original requests using the Tag field regardless of dstid field. Both local and remote Register Access requests are mastered by the Adapter with unique Tag encodings.

Table 7-4. UCIe Link sideband: srcid and dstid encodings for UCIe Link

Field	Description
srcid[2:0]	001b: D2D Adapter 010b: Physical Layer 011b: Management Port Gateway (see Section 8.2) other encodings are reserved
dstid[2]	1b: Remote die terminated request other encodings are reserved
dstid[1:0]	For Register Access requests: dstid[1:0] is Reserved. For Remote die terminated Register Access Completions: 01b: D2D Adapter other encodings are reserved. For Remote die terminated messages: 01b: D2D Adapter message 10b: Physical Layer message 11b: Management Port Gateway message (see Section 8.2)

7.1.2 Sideband Packet Formats

All the figures in this section show examples assuming a 32-bit interface of RDI/FDI transfer for sideband packets, hence the headers and data are shown in Phases of 32 bits.

Note that the sideband packet format figures provided in this chapter show the packet format over multiple 32-bit Phases. This is for representation purposes only. For transport over the PCIe sideband bumps (serial interface), the transfer occurs as a 64-bit serial packet at a time. For headers, the transmission order is bit 0 of Phase 0 as bit 0 of the serial packet (D0 in [Figure 4-7](#)), bit 1 of Phase 0 as bit 1 of the serial packet, etc., followed by bit 0 of Phase 1 as bit 32 of the serial packet, bit 1 of Phase 1 as bit 33 of the serial packet, etc., until bit 31 of Phase 1 as bit 63 of the serial packet.

Data (if present) is sent as a subsequent serial packet, with bit 0 of Phase 2 as bit 0 of the serial packet (D0 in [Figure 4-7](#)), bit 1 of Phase 2 as bit 1 of the serial packet, etc., followed by bit 0 of Phase 3 as bit 32 of the serial packet, bit 1 of Phase 3 as bit 33 of the serial packet, etc., until bit 31 of Phase 3 as bit 63 of the serial packet.

7.1.2.1 Register Access Packets

[Figure 7-1](#) shows the packet format for Register Access requests. [Table 7-5](#) gives the description of the fields other than the opcode, srcid, and dstid.

Table 7-5. Field descriptions for Register Access Requests

Field	Description
CP	Control Parity (CP) is the even parity of all the header bits excluding DP.
DP	Data Parity is the even parity of all bits in the data payload. If there is no data payload, this bit is set to 0b.
Cr	If 1b, indicates one credit return for credited sideband messages. This field is only used by the Adapter for remote Link partner's credit returns for E2E credits. It is not used for local FDI or RDI credit loops.
Addr[23:0]	Address of the request. Different opcodes use this field differently. See Table 7-6 for details. The following rules apply for the address field: For 64-bit request, Addr[2:0] is reserved. For 32-bit request, Addr[1:0] is reserved.
BE[7:0]	Byte Enables for the Request. It is NOT required to be contiguous. BE[7:4] are reserved if the opcode is for a 32-bit request.
EP	Data Poison. If poison forwarding is enabled, the completer can poison the data on internal errors. Setting the EP bit is optional, the conditions for setting it to 1 are implementation-specific. Typical usages involve giving additional FIT protection against data integrity errors on internal data buffers. A Receiver must not modify the contents of the target location for requests with data payload that have the EP bit set. It must return UR for the completion status of requests with an EP bit set.
Tag[4:0]	Tag is a 5-bit field generated by the requester, and it must be unique for all outstanding requests that require a completion. The original requester uses the Tag to associate returning completions with the original request.
Data	Payload. Can be 32 bits or 64 bits wide depending on the Opcode.

Table 7-6. Mapping of Addr[23:0] for Different Requests

Opcode	Description
Memory Reads/Writes	<p>{RL[3:0], Offset[19:0]}</p> <p>Offset is the Byte Offset.</p> <p>RL[3:0] encodings are as follows:</p> <ul style="list-style-type: none"> 0h: Register Locator 0 1h: Register Locator 1 2h: Register Locator 2 3h: Register Locator 3 <p>Fh: Accesses for Protocol specific MMIO registers that are shadowed in the Adapter (e.g., ARB/MUX registers defined in the <i>CXL Specification</i>). The offsets for these registers are implementation specific, and the protocol layer must translate accesses to match the offsets implemented in the Adapter.</p> <p>Other encodings are reserved.</p> <p>For accesses to Reserved RL encodings, the completer must respond with a UR.</p>
Configuration Reads/Writes	<p>{RL[3:0], Rsvd[7:0], Byte Offset[11:0]}, where</p> <p>RL[3:0] encodings are as follows:</p> <ul style="list-style-type: none"> 0h: PCIe Link DVSEC <p>Fh: Accesses for Protocol specific configuration registers that are shadowed in the Adapter (e.g., ARB/MUX registers defined in the <i>CXL Specification</i>). The offsets for these registers are implementation specific, and the protocol layer must translate accesses to match the offsets implemented in the Adapter.</p> <p>Other encodings are reserved.</p> <p>For accesses to Reserved RL encodings, the completer must respond with a UR.</p>
DMS Register Reads/Writes	<p>These allow for accessing the DMS registers implemented in PCIe Spoke Type 0, 1, or 2.</p> <p>Addr[21:0] provides the register offset in DMS register space, relative to the start of the Spoke's register space, that corresponds to the DevID. A maximum of 4 MB of address space is possible for PCIe D2D/PHY Spokes. These opcodes are always targeted at the local D2D or PHY registers (i.e., these opcodes never target the remote link partner).</p> <p>Addr[23:22] encodings are as follows:</p> <ul style="list-style-type: none"> 00b: Spoke registers. 01b: Reserved. 10b: Reserved. 11b: Used for other chiplet UMAP registers that are shadowed in the D2D or PHY, if any. The definitions of these registers and offsets are implementation-specific.

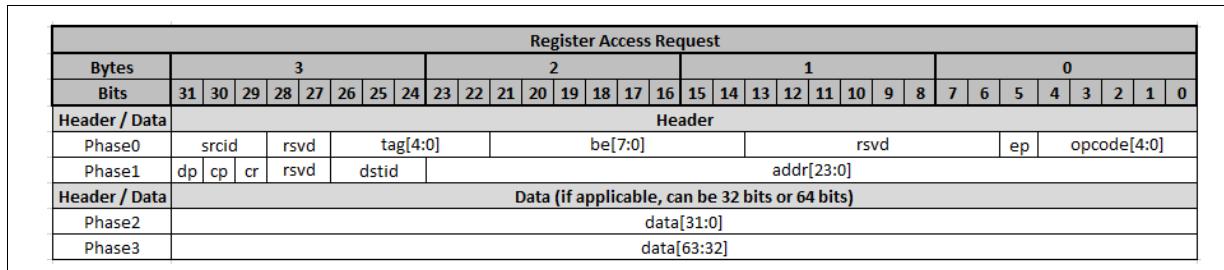
Figure 7-1. Format for Register Access Request

Figure 7-2 gives the format for Register Access completions.

Figure 7-2. Format for Register Access Completions

Register Access Completions																																																										
Bytes	3							2							1							0																																				
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																										
Header / Data	Header																																																									
Phase0	srcid			rsvd			tag[4:0]				be[7:0]				rsvd			ep	opcode[4:0]							Status																																
Phase1	dp	cp	cr	rsvd			dstid			rsvd														Status																																		
Header / Data	Data (if completion with data, can be 32 bits or 64 bits)																																																									
Phase2	data[31:0]																																																									
Phase3	data[63:32]																																																									

Table 7-7 gives the field descriptions for a completion.

Table 7-7. Field Descriptions for a Completion

Field	Description
Tag[4:0]	Completion Tag associated with the corresponding Request. The requester uses this to associate the completion with the original request.
CP	Control Parity. All fields other than "DP" and "CP" in the Header are protected by Control Parity, and the parity scheme is even (including reserved bits).
DP	Data Parity. All fields in data are protected by data parity, and the parity scheme is even.
Cr	If 1b, indicates one credit return for credited sideband messages. This field is only used by the Adapter for remote Link partner's credit returns for E2E credits. It is not used for local FDI or RDI credit loops.
EP	Data Poison. If poison forwarding is enabled, the completer can poison the data on internal errors. Setting the EP bit is optional, the conditions for setting it to 1 are implementation-specific. Typical usages involve giving additional FIT protection against data integrity errors on internal data buffers. A Receiver must not modify the contents of the target location for requests with data payload that have the EP bit set. It must return UR for the completion status of requests with an EP bit set.
BE[7:0]	Byte Enables for the Request. Completer returns the same value that the original request had (this avoids the requester from having to save off the BE value). BE[7:4] are reserved if the opcode is for a 32-bit request.
Status[2:0]	Completion Status 000b - Successful Completion (SC). This can be a completion with or without data, depending on the original request (it must set the appropriate Opcode). If the original request was a write, it is a completion without data. If the original request was a read, it is a completion with data. 001b - Unsupported Request (UR). On PCIe, this is a completion with 64b Data when a request is aborted by the completer, and the Data carries the original request header that resulted in UR. This enables easier header logging at the requester. Register Access requests that timeout must also return UR status, but for those the completion is without Data. 100b - Completer Abort (CA). On PCIe, this is a completion with 64b Data, and the Data carries original request header that resulted in CA. This enables easier header logging at the requester. 111b - Stall. Receiving a completion with Stall encoding must reset the timeout at the requester. Completer must send a Stall once every 4ms if it is not ready to respond to the original request. Other encodings are reserved. An error is logged in the Sideband Mailbox Status if a CA was received or if the number of timeouts exceed the programmed threshold. For timeouts below the programmed threshold, a UR is returned to the requester.
Data	Payload. 32 bits or 64 bits depending on the Opcode.

7.1.2.2 Messages without Data

Figure 7-3 shows the Format for Messages without data payloads. These can be Link Management Packets, NOPs or Vendor Defined message packets.

Figure 7-3. Format for Messages without Data

Messages without Data																																
Bytes	3							2							1							0										
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header / Data	Header																															
Phase0	srcid	rsvd	rsvd				msgcode[7:0]							rsvd							opcode[4:0]											
Phase1	dp	cp	rsvd	dstid		MsgInfo[15:0]							MsgSubcode[7:0]																			

The definitions of opcode, srcid, dstid, dp, and cp fields are the same as Register Access packets.

Table 7-8 and Table 7-9 give the encodings of the different messages without data that are send on UCIe. Some Notes on the different message categories are listed below:

- {NOP.Crd} — These are used for E2E Credit returns. The destination must be D2D Adapter.
- {LinkMgmt.RDI.*} — These are used to coordinate RDI state transitions, the source and destination is Physical Layer.
- {LinkMgmt.Adapter0.*} — These are used to coordinate Adapter LSM state transitions for the Adapter LSM corresponding to Stack 0 Protocol Layer. The source and destination is D2D Adapter.
- {LinkMgmt.Adapter1.*} — These are used to coordinate Adapter LSM state transitions for the Adapter LSM corresponding to Stack 1 Protocol Layer. The source and destination is D2D Adapter.
- {ParityFeature.*} — This is used to coordinate enabling of the Parity insertion feature. The source and destination for this must be the D2D Adapter.
- {ErrMsg} — This is used for error reporting and escalation from the remote Link Partner. This is sent from the Retimer or Device die to the Host, and the destination must be the D2D Adapter.

Table 7-8. Message Encodings for Messages without Data^a (Sheet 1 of 3)

Name	Msgcode	Msgsubcode	MsgInfo	Description
{Nop.Crd}	00h	00h	0000h: Reserved 0001h: 1 Credit return 0002h: 2 Credit returns 0003h: 3 Credit returns 0004h: 4 Credit returns	Explicit Credit return from Remote Link partner for credited messages.
{LinkMgmt.RDI.Req.Active}	01h	01h	Reserved	Active Request for RDI SM.
{LinkMgmt.RDI.Req.L1}		04h		L1 Request for RDI SM.
{LinkMgmt.RDI.Req.L2}		08h		L2 Request for RDI SM.
{LinkMgmt.RDI.Req.LinkReset}		09h		LinkReset Request for RDI SM.
{LinkMgmt.RDI.Req.LinkError}		0Ah		LinkError Request for RDI SM.
{LinkMgmt.RDI.Req.Retrain}		0Bh		Retrain Request for RDI SM.
{LinkMgmt.RDI.Req.Disable}		0Ch		Disable Request for RDI SM.

Table 7-8. Message Encodings for Messages without Data^a (Sheet 2 of 3)

Name	Msgcode	Msgsubcode	MsgInfo	Description
{LinkMgmt.RDI.Rsp.Active}	02h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for RDI SM.
{LinkMgmt.RDI.Rsp.PMNAK}		02h		PMNAK Response for RDI SM
{LinkMgmt.RDI.Rsp.L1}		04h		L1 Response for RDI SM.
{LinkMgmt.RDI.Rsp.L2}		08h		L2 Response for RDI SM.
{LinkMgmt.RDI.Rsp.LinkReset}		09h		LinkReset Response for RDI SM.
{LinkMgmt.RDI.Rsp.LinkError}		0Ah		LinkError Response for RDI SM.
{LinkMgmt.RDI.Rsp.Retrain}		0Bh		Retrain Response for RDI SM.
{LinkMgmt.RDI.Rsp.Disable}		0Ch		Disable Response for RDI SM.
{LinkMgmt.Adapter0.Req.Active}	03h	01h	0000h: Regular Request FFFFh: Stall	Active Request for Stack 0 Adapter LSM. The Stall encoding is provided for Retimers to avoid the Adapter LSM transition to Active timeout as described in Section 9.5.3.8 .
{LinkMgmt.Adapter0.Req.L1}		04h	Reserved	L1 Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Req.L2}		08h		L2 Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Req.LinkReset}		09h		LinkReset Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Req.Disable}		0Ch		Disable Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Rsp.Active}	04h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Rsp.PMNAK}		02h		PMNAK Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Rsp.L1}		04h		L1 Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Rsp.L2}		08h		L2 Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Rsp.LinkReset}		09h		LinkReset Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter0.Rsp.Disable}		0Ch		Disable Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter1.Req.Active}	05h	01h	0000h: Regular Request FFFFh: Stall	Active Request for Stack 1 Adapter LSM. The Stall encoding is provided for Retimers to avoid the Adapter LSM transition to Active timeout as described in Section 9.5.3.8 .
{LinkMgmt.Adapter1.Req.L1}		04h	Reserved	L1 Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter1.Req.L2}		08h		L2 Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter1.Req.LinkReset}		09h		LinkReset Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter1.Req.Disable}		0Ch		Disable Request for Stack 1 Adapter LSM.

Table 7-8. Message Encodings for Messages without Data^a (Sheet 3 of 3)

Name	Msgcode	Msgsubcode	MsgInfo	Description
{LinkMgmt.Adapter.1.Rsp.Active}	06h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter.1.Rsp.PMNAK}		02h		PMNAK Response for Stack 1 Adapter LSM
{LinkMgmt.Adapter.1.Rsp.L1}		04h		L1 Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter.1.Rsp.L2}		08h		L2 Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter.1.Rsp.LinkReset}		09h		LinkReset Response for Stack 1 Adapter LSM.
{LinkMgmt.Adapter.1.Rsp.Disable}		0Ch		Disable Response for Stack 1 Adapter LSM.
{ParityFeature.Req}	07h	00h	Reserved	Parity Feature enable request.
{ParityFeature.Ack}	08h	00h	0000h: Regular Response FFFFh: Stall Response	Parity Feature enable Ack.
{ParityFeature.Nak}		01h		Parity Feature enable Nak.
{ErrMsg}	09h	00h	Reserved	Correctable Error Message.
		01h		Non-Fatal Error Message.
		02h		Fatal Error Message.
{Vendor Defined Message}	FFh	--	Vendor ID	<p>Vendor Defined Messages. These can be exchanged at any time after sideband is functional post SBINIT. Interoperability is vendor defined. Unsupported vendor defined messages must be discarded by the receiver. Note that this is NOT the UCIe Vendor ID, but rather the unique identifier of the chiplet vendor that is defining and using these messages.</p>

a. All other encodings not mentioned in this table are reserved.

Table 7-9. Link Training State Machine related Message encodings for messages without data (Sheet 1 of 5)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{Start Tx Init D to C point test resp}	0000h	8Ah	01h
{LFSR_clear_error req}	0000h	85h	02h
{LFSR_clear_error resp}	0000h	8Ah	02h
{Tx Init D to C results req}	0000h	85h	03h
{End Tx Init D to C point test req}	0000h	85h	04h
{End Tx Init D to C point test resp}	0000h	8Ah	04h
{Start Tx Init D to C eye sweep resp}	0000h	8Ah	05h
{End Tx Init D to C eye sweep req}	0000h	85h	06h
{End Tx Init D to C eye sweep resp}	0000h	8Ah	06h
{Start Rx Init D to C point test resp}	0000h	8Ah	07h
{Rx Init D to C Tx Count Done req}	0000h	85h	08h
{Rx Init D to C Tx Count Done resp}	0000h	8Ah	08h
{End Rx Init D to C point test req}	0000h	85h	09h
{End Rx Init D to C point test resp}	0000h	8Ah	09h
{Start Rx Init D to C eye sweep resp}	0000h	8Ah	0Ah
{Rx Init D to C results req}	0000h	85h	0Bh
{End Rx Init D to C eye sweep req}	0000h	85h	0Dh
{End Rx Init D to C eye sweep resp}	0000h	8Ah	0Dh
{SBINIT out of Reset}	[15:4]: Reserved [3:0]: Result ^a	91h	00h
{SBINIT done req}	0000h	95h	01h
{SBINIT done resp}	0000h	9Ah	01h
{MBINIT.CAL Done req}	0000h	A5h	02h
{MBINIT.CAL Done resp}	0000h	AAh	02h
{MBINIT.REPAIRCLK init req}	0000h	A5h	03h
{MBINIT.REPAIRCLK init resp}	0000h	AAh	03h
{MBINIT.REPAIRCLK result req}	0000h	A5h	04h
{MBINIT.REPAIRCLK result resp}	[15:4]: Reserved [3]: Compare Results from RRDCK_L [2]: Compare Results from RTRK_L [1]: Compare Results from RCKN_L [0]: Compare Results from RCKP_L	AAh	04h
{MBINIT.REPAIRCLK apply repair req}	[15:4]: Reserved [3:0]: Repair Encoding • Fh: Reserved • Oh: Repair RCKP_L • 1h: Repair RCKN_L • 2h: Repair RTRK_L • 7h: Reserved	A5h	05h
{MBINIT.REPAIRCLK apply repair resp}	0000h	AAh	05h

Table 7-9. Link Training State Machine related Message encodings for messages without data (Sheet 2 of 5)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBINIT.REPAIRCLK check repair init req}	0000h	A5h	06h
{MBINIT.REPAIRCLK check repair init resp}	0000h	AAh	06h
{MBINIT.REPAIRCLK check results req}	0000h	A5h	07h
{MBINIT.REPAIRCLK check results resp}	[15:4]: Reserved [3]: Compare Results from RRDCK_L [2]: Compare Results from RTRK_L [1]: Compare Results from RCKN_L [0]: Compare Results from RCKP_L	AAh	07h
{MBINIT.REPAIRCLK done req}	0000h	A5h	08h
{MBINIT.REPAIRCLK done resp}	0000h	AAh	08h
{MBINIT.REPAIRVAL init req}	0000h	A5h	09h
{MBINIT.REPAIRVAL init resp}	0000h	AAh	09h
{MBINIT.REPAIRVAL result req}	0000h	A5h	0Ah
{MBINIT.REPAIRVAL result resp}	[15:2]: Reserved [1]: Compare Results from RRDVLD_L [0]: Compare Results from RVLD_L	AAh	0Ah
{MBINIT.REPAIRVAL apply repair req}	[15:2]: Reserved [1:0]: Repair Encoding <ul style="list-style-type: none">• 3h: Reserved• 0h: Repair RVLD_L• 1h: Reserved	A5h	0Bh
{MBINIT.REPAIRVAL apply repair resp}	0000h	AAh	0Bh
{MBINIT.REPAIRVAL done req}	0000h	A5h	0Ch
{MBINIT.REPAIRVAL done resp}	0000h	AAh	0Ch
{MBINIT.REVERSALMB init req}	0000h	A5h	0Dh
{MBINIT.REVERSALMB init resp}	0000h	AAh	0Dh
{MBINIT.REVERSALMB clear error req}	0000h	A5h	0Eh
{MBINIT.REVERSALMB clear error resp}	0000h	AAh	0Eh
{MBINIT.REVERSALMB result req}	0000h	A5h	0Fh
{MBINIT.REVERSALMB done req}	0000h	A5h	10h
{MBINIT.REVERSALMB done resp}	0000h	AAh	10h
{MBINIT.REPAIRMB start req}	0000h	A5h	11h
{MBINIT.REPAIRMB start resp}	0000h	AAh	11h
{MBINIT.REPAIRMB Apply repair resp}	0000h	AAh	12h
{MBINIT.REPAIRMB end req}	0000h	A5h	13h
{MBINIT.REPAIRMB end resp}	0000h	AAh	13h
{MBINIT.REPAIRMB apply degrade req}	[15:3]: Reserved [2:0]: Standard package logical Lane map	A5h	14h
{MBINIT.REPAIRMB apply degrade resp}	0000h	AAh	14h

Table 7-9. Link Training State Machine related Message encodings for messages without data (Sheet 3 of 5)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBTRAIN.VALVREF start req}	0000h	B5h	00h
{MBTRAIN.VALVREF start resp}	0000h	BAh	00h
{MBTRAIN.VALVREF end req}	0000h	B5h	01h
{MBTRAIN.VALVREF end resp}	0000h	BAh	01h
{MBTRAIN.DATAVREF start req}	0000h	B5h	02h
{MBTRAIN.DATAVREF start resp}	0000h	BAh	02h
{MBTRAIN.DATAVREF end req}	0000h	B5h	03h
{MBTRAIN.DATAVREF end resp}	0000h	BAh	03h
{MBTRAIN.SPEEDIDLE done req}	0000h	B5h	04h
{MBTRAIN.SPEEDIDLE done resp}	0000h	BAh	04h
{MBTRAIN.TXSELFICAL Done req}	0000h	B5h	05h
{MBTRAIN.TXSELFICAL Done resp}	0000h	BAh	05h
{MBTRAIN.RXCLKCAL start req}	0000h	B5h	06h
{MBTRAIN.RXCLKCAL start resp}	0000h	BAh	06h
{MBTRAIN.RXCLKCAL done req}	0000h	B5h	07h
{MBTRAIN.RXCLKCAL done resp}	0000h	BAh	07h
{MBTRAIN.VALTRAINCENTER start req}	0000h	B5h	08h
{MBTRAIN.VALTRAINCENTER start resp}	0000h	BAh	08h
{MBTRAIN.VALTRAINCENTER done req}	0000h	B5h	09h
{MBTRAIN.VALTRAINCENTER done resp}	0000h	BAh	09h
{MBTRAIN.VALTRAINVREF start req}	0000h	B5h	0Ah
{MBTRAIN.VALTRAINVREF start resp}	0000h	BAh	0Ah
{MBTRAIN.VALTRAINVREF done req}	0000h	B5h	0Bh
{MBTRAIN.VALTRAINVREF done resp}	0000h	BAh	0Bh
{MBTRAIN.DATATRAINCENTER1 start req}	0000h	B5h	0Ch
{MBTRAIN.DATATRAINCENTER1 start resp}	0000h	BAh	0Ch
{MBTRAIN.DATATRAINCENTER1 end req}	0000h	B5h	0Dh
{MBTRAIN.DATATRAINCENTER1 end resp}	0000h	BAh	0Dh
{MBTRAIN.DATATRAINVREF start req}	0000h	B5h	0Eh
{MBTRAIN.DATATRAINVREF start resp}	0000h	BAh	0Eh
{MBTRAIN.DATATRAINVREF end req}	0000h	B5h	10h
{MBTRAIN.DATATRAINVREF end resp}	0000h	BAh	10h
{MBTRAIN.RXDESKEW start req}	0000h	B5h	11h
{MBTRAIN.RXDESKEW start resp}	0000h	BAh	11h
{MBTRAIN.RXDESKEW end req}	0000h	B5h	12h
{MBTRAIN.RXDESKEW end resp}	0000h	BAh	12h
{MBTRAIN.DATATRAINCENTER2 start req}	0000h	B5h	13h
{MBTRAIN.DATATRAINCENTER2 start resp}	0000h	BAh	13h
{MBTRAIN.DATATRAINCENTER2 end req}	0000h	B5h	14h
{MBTRAIN.DATATRAINCENTER2 end resp}	0000h	BAh	14h

Table 7-9. Link Training State Machine related Message encodings for messages without data (Sheet 4 of 5)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBTRAIN.LINKSPEED start req}	0000h	B5h	15h
{MBTRAIN.LINKSPEED start resp}	0000h	BAh	15h
{MBTRAIN.LINKSPEED error req}	0000h	B5h	16h
{MBTRAIN.LINKSPEED error resp}	0000h	BAh	16h
{MBTRAIN.LINKSPEED exit to repair req}	0000h	B5h	17h
{MBTRAIN.LINKSPEED exit to repair resp}	0000h	BAh	17h
{MBTRAIN.LINKSPEED exit to speed degrade req}	0000h	B5h	18h
{MBTRAIN.LINKSPEED exit to speed degrade resp}	0000h	BAh	18h
{MBTRAIN.LINKSPEED done req}	0000h: For regular response	B5h	19h
{MBTRAIN.LINKSPEED done resp}	0000h: For regular response FFFFh: For stall	BAh	19h
{MBTRAIN.LINKSPEED multi-module disable module resp}	0000h	BAh	1Ah
{MBTRAIN.LINKSPEED exit to phy retrain req}	0000h	B5h	1Fh
{MBTRAIN.LINKSPEED exit to phy retrain resp}	0000h	BAh	1Fh
{MBTRAIN.REPAIR init req}	0000h	B5h	1Bh
{MBTRAIN.REPAIR init resp}	0000h	BAh	1Bh
{MBTRAIN.REPAIR Apply repair resp}	0000h	BAh	1Ch
{MBTRAIN.REPAIR end req}	0000h	B5h	1Dh
{MBTRAIN.REPAIR end resp}	0000h	BAh	1Dh
{MBTRAIN.REPAIR Apply degrade req}	[15:3]: Reserved [2:0]: Standard Package logical Lane map ^b	B5h	1Eh
{MBTRAIN.REPAIR Apply degrade resp}	0000h	BAh	1Eh
{MBTRAIN.RXDESKEW EQ Preset req}	[15:4]: Reserved [3:0]: EQ Preset encoding	B5h	1Fh
{MBTRAIN.RXDESKEW EQ Preset resp}	[15:1]: Reserved [0]: Status • 0: Success • 1: Fail	BAh	1Fh
{MBTRAIN.RXDESKEW exit to DATATRAINCENTER1 req}	0000h	B5h	20h
{MBTRAIN.RXDESKEW exit to DATATRAINCENTER1 resp}	0000h	BAh	20h

Table 7-9. Link Training State Machine related Message encodings for messages without data (Sheet 5 of 5)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBTRAIN.RXCLKCAL TCKN_L shift req}	[15:6]: Reserved [5:1]: Shift Value. Indicates the value of the shift from the current setting with a step size of 1/64 UI [0]: Decrement <ul style="list-style-type: none">• 0: Increment• 1: Decrement	B5h	21h
{MBTRAIN.RXCLKCAL TCKN_L shift resp}	[15:1]: Reserved [0]: Status <ul style="list-style-type: none">• 0: Shift was successfully applied• 1: Shift was not applied (most likely scenario is that it is Out of Range)	BAh	21h
{PHYRETRAIN.retrain start req}	[15:3]: Reserved [2:0]: Retrain Encoding	C5h	01h
{PHYRETRAIN.retrain start resp}	[15:3]: Reserved [2:0]: Retrain Encoding	CAh	01h
{TRAINERROR Entry req}	0000h	E5h	00h
{TRAINERROR Entry resp}	0000h	EAh	00h
{RECAL.track pattern init req}	0000h	D5h	00h
{RECAL.track pattern init resp}	0000h	DAh	00h
{RECAL.track pattern done req}	0000h	D5h	01h
{RECAL.track pattern done resp}	0000h	DAh	01h
{RECAL.track tx adjust req}	[15:9]: Reserved [8]: <ul style="list-style-type: none">• 0: Increment clock delay or decrement data delay• 1: Decrement clock delay or increment data delay [7:0]: Delay compensation value, in units of 1/64 UI	B5h	22h
{RECAL.track tx adjust resp}	[15:2]: Reserved [1:0]: Status <ul style="list-style-type: none">• 00b: Reserved• 01b: Drift compensated• 10b: Drift not compensated• 11b: Stall	BAh	22h

- a. See [Section 4.5.3.2](#).
b. See [Table 4-9](#).

7.1.2.3 Messages with data payloads

[Figure 7-4](#) shows the formats for Messages with data payloads. The definitions of opcode, srcid, dstid, dp, and cp fields are the same as Register Access packets.

Figure 7-4. Format for Messages with data payloads

Messages with data																																																																							
Bytes	3							2							1							0																																																	
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																							
Header / Data	Header																																																																						
Phase0	srcid			rsvd				rsvd				msgcode[7:0]							rsvd				opcode[4:0]																																																
Phase1	dp	cp	rsvd	dstid				MsgInfo[15:0]												MsgSubcode[7:0]																																																			
Header / Data	Data																																																																						
Phase2	data[31:0]																																																																						
Phase3	data[63:32]																																																																						

Table 7-10 and Table 7-11 give the message encodings.

Table 7-10. Message encodings for Messages with Data^a (Sheet 1 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{AdvCap.Adapter}	01h	00h	0000h: Regular Message FFFFh: Stall Message	[0]: "Raw Format" [1]: "68B Flit Mode" [2]: "CXL 256B Flit Mode" [3]: "PCIe Flit Mode" [4]: "Streaming" [5]: "Retry" [6]: "Multi_Protocol_Enable" [7]: "Stack0_Enable" [8]: "Stack1_Enable" [9]: "CXL_LatOpt_Fmt5" [10]: "CXL_LatOpt_Fmt6" [11]: "Retimer" [20:12]: "Retimer Credits" [21]: "DP" [22]: "UP" [23]: "68B Flit Format" [24]: "Standard 256B End Header Flit Format" [25]: "Standard 256B Start Header Flit Format" [26]: "Latency-Optimized 256B without Optional Bytes Flit Format" [27]: "Latency-Optimized 256B with Optional Bytes Flit Format" [28]: "Enhanced_Multi_Protocol_Enable" [29]: "Stack 0 Maximum Bandwidth_Limit" [30]: "Stack 1 Maximum Bandwidth_Limit" [31]: "Management Transport Protocol" [63:32]: Reserved	Advertised Capabilities of the D2D Adapter
{FinCap.Adapter}	02h	00h	0000h: Regular Message FFFFh: Stall Message	[0]: "Raw Format" [1]: "68B Flit Mode" [2]: "CXL 256B Flit Mode" [3]: "PCIe Flit Mode" [4]: "Streaming" [5]: "Retry" [6]: "Multi_Protocol_Enable" [7]: "Stack0_Enable" [8]: "Stack1_Enable" [9]: "CXL_LatOpt_Fmt5" [10]: "CXL_LatOpt_Fmt6" [11]: "Retimer" [20:12]: "Retimer Credits" [21]: "DP" [22]: "UP" [23]: "68B Flit Format" [24]: "Standard 256B End Header Flit Format" [25]: "Standard 256B Start Header Flit Format" [26]: "Latency-Optimized 256B without Optional Bytes Flit Format" [27]: "Latency-Optimized 256B with Optional Bytes Flit Format" [28]: "Enhanced_Multi_Protocol_Enable" [29]: "Stack 0 Maximum Bandwidth_Limit" [30]: "Stack 1 Maximum Bandwidth_Limit" [31]: "Management Transport Protocol" [63:32]: Reserved	Finalized Capability of the D2D Adapter

Table 7-10. Message encodings for Messages with Data^a (Sheet 2 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{AdvCap.CXL}	01h	01h	0000h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 0b, or it is 1b and the message is for Stack 0. 0001h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 1b and the message is for Stack 1. FFFFh: Stall Message	[23:0]: Flexbus Mode negotiation usage bits as defined for Symbols 12-14 of Modified TS1/TS2 Ordered Set in <i>CXL Specification</i> , with the following additional rules: <ul style="list-style-type: none">• [0]: PCIe capable/enable - this must be 1b for PCIe Non-Flit Mode.• [1]: CXL.io capable/enable - this must be 0b for PCIe Non-Flit Mode.• [2]: CXL.mem capable/enable - this must be 0b for PCIe Non-Flit Mode.• [3]: CXL.cache capable/enable - this must be 0b for PCIe Non-Flit Mode.• [4]: CXL 68B Flit and VH capable; must be set for ports that support CXL protocols, as specified in the Protocol Layer interoperability requirements.• [8]: Multi-Logical Device - must be set to 0b for PCIe Non-Flit Mode.• [9]: Reserved.• [12:10]: these bits do not apply for UCle, must be 0b.• [14]: Retimer 2 - does not apply for UCle, must be 0b.• [15]: CXL.io Throttle - must be 0b for PCIe Non-Flit Mode.• [17:16]: NOP Hint Info - does not apply for UCle, and must be 0.	Advertised Capabilities for CXL protocol.
{FinCap.CXL}	02h	01h	0000h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 0b, or it is 1b and the message is for Stack 0. 0001h: Post negotiation, if Enhanced_Multi_Protocol_Enable is 1b and the message is for Stack 1. FFFFh: Stall Message	[0:15]: These bits do not apply for UCle, must be 0b.	Finalized Capabilities for CXL protocol.
{MultiProtAdvCap.Adapter}	01h	02h	0000h: Reserved FFFFh: Stall Message	[0]: "68B Flit Mode" [1]: "CXL 256B Flit Mode" [2]: "PCIe Flit Mode" [3]: "Streaming Protocol" [4]: "Management Transport Protocol" [63:5]: Reserved	Protocol Advertisement for Stack 1 when Enhanced Multi_Protocol_Enable is negotiated
{MultiProtFinCap.Adapter}	02h	02h	0000h: Reserved FFFFh: Stall Message	[0]: "68B Flit Mode" [1]: "CXL 256B Flit Mode" [2]: "PCIe Flit Mode" [3]: Reserved [4]: "Management Transport Protocol" [63:5]: Reserved	Finalized Capability for Protocol negotiation when Enhanced Multi_Protocol_Enable is negotiated and Stack 1 is PCIe or CXL

Table 7-10. Message encodings for Messages with Data^a (Sheet 3 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{Vendor Defined Message}	FFh	--	Vendor ID		<p>Vendor Defined Messages.</p> <p>These can be exchanged at any time after sideband is functional post SBINIT.</p> <p>Interoperability is vendor defined.</p> <p>Unsupported vendor defined messages must be discarded by the receiver.</p> <p>Note that this is NOT the UCIE Vendor ID, but rather the unique identifier of the chiplet vendor that is defining and using these messages.</p>

a. All other encodings not mentioned in this table are reserved.

Table 7-11. Link Training State Machine related encodings (Sheet 1 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{Start Tx Init D to C point test req}	[15:0]: Maximum comparison error threshold	85h	01h	<p>[63:60]: Reserved</p> <p>[59]: Comparison Mode (0: Per Lane; 1: Aggregate)</p> <p>[58:43]: Iteration Count Settings</p> <p>[42:27]: Idle Count settings</p> <p>[26:11]: Burst Count settings</p> <p>[10]: Pattern Mode (0: continuous mode, 1: Burst Mode)</p> <p>[9:6]: Clock Phase control at Tx Device (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge)</p> <p>[5:3]: Valid Pattern (0h: Functional pattern)</p> <p>[2:0]: Data pattern (0h: LFSR, 1h: Per Lane ID)</p>

Table 7-11. Link Training State Machine related encodings (Sheet 2 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{Tx Init D to C results resp}	[15:6]: Reserved [5]: Valid Lane comparison results [4]: Cumulative Results of all Lanes (0: Fail (Errors > Max Error Threshold), 1: Pass (Errors <= Max Error Threshold)). [3:0]: PCIe-A: Compare results from Redundant Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) RRD_L[3], RRD_L[2], RRD_L[1], RRD_L[0] PCIe-S: Reserved RRD_L[3] and RRD_L[2] are reserved for PCIe-A x32 as a transmitter of this message.	8Ah	03h	[63:0]: Compare Results of individual Data Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) PCIe-A {RD_L[63], RD_L[62], ..., RD_L[1], RD_L[0]} PCIe-S {48'h0, RD_L[15], RD_L[14], ..., RD_L[1], RD_L[0]} PCIe-A x32 {32'h0, RD_L[31], RD_L[30], ..., RD_L[0]} PCIe-S x8 {56'h0, RD_L[7], RD_L[6], ..., RD_L[1], RD_L[0]}
{Start Tx Init D to C eye sweep req}	[15:0]: Maximum comparison error threshold	85h	05h	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43]: Iteration Count Settings [42:27]: Idle Count settings [26:11]: Burst Count settings [10]: Pattern Mode (0: continuous mode, 1: Burst Mode) [9:6]: Clock Phase control at Tx Device (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3]: Valid Pattern (0h: Functional pattern) [2:0]: Data pattern (0h: LFSR, 1h: Per Lane ID)
{Start Rx Init D to C point test req}	[15:0]: Maximum comparison error threshold	85h	07h	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43]: Iteration Count Settings [42:27]: Idle Count settings [26:11]: Burst Count settings [10]: Pattern Mode (0: continuous mode, 1: Burst Mode) [9:6]: Clock Phase control at Transmitter (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3]: Valid Pattern (0h: Functional pattern) [2:0]: Data pattern (0h: LFSR, 1h: Per Lane ID)

Table 7-11. Link Training State Machine related encodings (Sheet 3 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{Start Rx Init D to C eye sweep req}	[15:0]: Maximum comparison error threshold	85h	0Ah	[63:60]: Reserved [59]: Comparison Mode (0: Per Lane; 1: Aggregate) [58:43]: Iteration Count Settings [42:27]: Idle Count settings [26:11]: Burst Count settings [10]: Pattern Mode (0: continuous mode, 1: Burst Mode) [9:6]: Clock Phase control at Transmitter (0h: Clock PI Center, 1h: Left Edge, 2h: Right Edge) [5:3]: Valid Pattern (0h: Functional pattern) [2:0]: Data pattern (0h: LFSR, 1h: Per Lane ID)
{Rx Init D to C results resp}	[15:6]: Reserved [5]: Valid Lane comparison result [4]: Cumulative Results of all Lanes (0: Fail (Errors > Max Error Threshold), 1: Pass (Errors <= Max Error Threshold)). [3:0]: PCIe-A: Compare results from Redundant Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) (RRD_L[3], RRD_L[2], RRD_L[1], RRD_L[0]) PCIe-S: Reserved RRD_L[3] and RRD_L[2] are reserved for PCIe-A x32 as a transmitter of this message.	8Ah	0Bh	[63:0]: Compare Results of individual Data Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) PCIe-A { RD_L[63], RD_L[62], ..., RD_L[1], RD_L[0] } PCIe-S {48'h0, RD_L[15], RD_L[14], ..., RD_L[1], RD_L[0] } PCIe-A x32 {32'h0, RD_L[31], RD_L[30], ..., RD_L[0] } PCIe-S x8 {56'h0, RD_L[7], RD_L[6], ..., RD_L[1], RD_L[0] }
{Rx Init D to C sweep done with results}	0000h	81h	0Ch	[63:16]: Reserved [15:8]: Right Edge [7:0]: Left Edge
{MBINIT.PARAM configuration req}	0000h	A5h	00h	[63:16]: Reserved [15]: Tx Adjustment during Runtime Recalibration (TARR) is supported (1) or not supported (0) [14]: Sideband feature extensions is supported (1) or not supported (0) [13]: PCIe-A x32 if Advanced Package; PCIe-S x8 if Standard Package. [12:11]: Module ID: 0h: 0, 1h: 1, 2h: 2, 3h: 3 [10]: Clock Phase: 0b: Differential clock, 1b: Quadrature phase [9]: Clock Mode - 0b: Strobe mode; 1b: Continuous mode [8:4]: Voltage Swing - The encodings are the same as the "Supported Tx Vswing encodings" field of the PHY Capability register [3:0]: Max IO Link Speed - The encodings are the same as "Max Link Speeds" field of the PCIe Link Capability register

Table 7-11. Link Training State Machine related encodings (Sheet 4 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{MBINIT.PARAM configuration resp}	0000h	AAh	00h	[63:16]: Reserved [15]: Tx Adjustment during Runtime Recalibration (TARR) is negotiated (1) or not supported (0) [14]: Sideband feature extensions is negotiated (1) or not supported (0) [13:11]: Reserved [10]: Clock Phase: 0b: Differential clock, 1b: Quadrature phase [9]: Clock Mode - 0b: Strobe mode; 1b: Continuous mode [8:4]: Reserved [3:0]: Max IO Link Speed - The encodings are the same as "Max Link Speeds" field of the UCIe Link Capability register
{MBINIT.PARAM SBFE req}	0000h: Regular Message	A5h	01h	[63:5]: Reserved [4]: L2SPD is supported (1) or not supported (0) [3]: PSPT is supported (1) or not supported (0) [2]: Sideband-only (SO) port (1), full UCIe port (0) [1]: Sideband Performant Mode Operation (PMO) is supported (1) or not supported (0) [0]: Management Transport protocol is supported (1) or not supported (0)
{MBINIT.PARAM SBFE resp}	0000h: Regular Message FFFFh: Stall Message	AAh	01h	[63:5]: Reserved [4]: L2SPD is negotiated (1) or not negotiated (0) [3]: PSPT is negotiated (1) or not negotiated (0) [2]: Sideband-only (SO) port (1), full UCIe port (0) [1]: Sideband Performant Mode Operation (PMO) is negotiated (1) or not supported (0) [0]: Management Transport protocol is supported (1) or not supported (0)
{MBINIT.REVERSAL MB result resp}	The error condition for this flow is NOT observing 16 consecutive iterations of the expected pattern. The error threshold is always 0 for this test. [15:4]: Reserved [3:0]: UCIe-A: Compare results from Redundant Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) RRD_L[3], RRD_L[2], RRD_L[1], RRD_L[0] UCIe-S: Reserved RRD_L[3] and RRD_L[2] are reserved for UCIe-A x32 as a transmitter of this message.	AAh	0Fh	The error condition for this flow is NOT observing 16 consecutive iterations of the expected pattern. The error threshold is always 0 for this test. [63:0]: Compare Results of individual Data Lanes (0h: Fail (Errors > Max Error Threshold), 1h: Pass (Errors <= Max Error Threshold)) UCIe-A {RD_L[63], RD_L[62], ..., RD_L[1], RD_L[0]} UCIe-S {48'h0, RD_L[15], RD_L[14], ..., RD_L[1], RD_L[0]} UCIe-A x32 {32'h0, RD_L[31], RD_L[30], ..., RD_L[0]} UCIe-S x8 {56'h0, RD_L[7], RD_L[6], ..., RD_L[1], RD_L[0]}

Table 7-11. Link Training State Machine related encodings (Sheet 5 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{MBINIT.REPAIRMB Apply repair req}	0000h	A5h	12h	<p>[31:24]: Repair Address for TRD_P[3]: Indicates the physical Lane repaired when TRD_P[3] is used in remapping scheme. This is reserved for UCle-A x32 as a transmitter of this message.</p> <p>20h: Invalid 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Reserved FFh: No Repair</p> <p>[23:16]: Repair Address for TRD_P[2]: Indicates the physical Lane repaired when TRD_P[2] is used in remapping scheme. This is reserved for UCle-A x32 as a transmitter of this message.</p> <p>20h: TD_P[32] Repaired 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Reserved FFh: No Repair</p> <p>[15:8]: Repair Address for TRD_P[1]: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme.</p> <p>00h: Invalid 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Reserved FFh: No Repair</p> <p>[7:0]: Repair Address for TRD_P[0]: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme.</p> <p>00h: TD_P[0] Repaired 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Reserved FFh: No Repair</p>

Table 7-11. Link Training State Machine related encodings (Sheet 6 of 6)

Message	MsgInfo[15:0]	MsgCode [7:0]	MsgSubcode [7:0]	Data Field[63:0]
{ MBTRAIN.REPAIR Apply repair req}	0000h	B5h	1Ch	<p>[31:24]: Repair Address for TRD_P[3]: Indicates the physical Lane repaired when TRD_P[3] is used in remapping scheme. This is reserved for UCle-A x32 as a transmitter of this message.</p> <p>20h: Invalid 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Reserved FFh: No Repair</p> <p>[23:16]: Repair Address for TRD_P[2]: Indicates the physical Lane repaired when TRD_P[2] is used in remapping scheme. This is reserved for UCle-A x32 as a transmitter of this message.</p> <p>20h: TD_P[32] Repaired 21h: TD_P[33] Repaired 22h: TD_P[34] Repaired 3Eh: TD_P[62] Repaired 3Fh: TD_P[63] Repaired F0h: Reserved FFh: No Repair</p> <p>[15:8]: Repair Address for TRD_P[1]: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme.</p> <p>00h: Invalid 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Reserved FFh: No Repair</p> <p>[7:0]: Repair Address for TRD_P[0]: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme.</p> <p>00h: TD_P[0] Repaired 01h: TD_P[1] Repaired 02h: TD_P[2] Repaired 1Eh: TD_P[30] Repaired 1Fh: TD_P[31] Repaired F0h: Reserved FFh: No Repair</p>

7.1.2.4 Management Port Message (MPM) with Data

As with all sideband messages, Management Port Messages with Data also carry a 1-QWORD header. This is referred to as “MPM header” (see [Figure 7-5](#)) for the remainder of this section. The payload in these messages is referred to as “MPM payload” for the remainder of this section.

Bits [21:14] in the first DW of the MPM Hdr of an MPM with Data message, forms an 8b msgcode that denotes a specific MPM with Data message. [Table 7-12](#) summarizes the supported MPM with Data messages over sideband.

Support for these messages is optional and negotiated as described in [Section 8.2.3.1](#).

Table 7-12. Supported MPM with Data Messages on Sideband

msgcode	Message
01h	Encapsulated MTP Message
FFh	Vendor-defined Management Port Gateway Message
Others	Reserved

7.1.2.4.1 Common Fields in MPM Header of MPM with Data Messages on Sideband

[Figure 7-5](#) shows and [Table 7-13](#) describes the common fields in the MPM header of MPM with data messages on the sideband.

Figure 7-5. Common Fields in MPM Header of all MPM with Data Messages on Sideband

3			2						1						0										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6
srcid=011b		rsvd		re	sp	vc		msgcode						length						rs	vd	opcode = 11000b			
rs	vd	cp	rsvd	dstid=111b		msgcode-specific										rsvd	msgcode-specific		rsvd	rxqid					

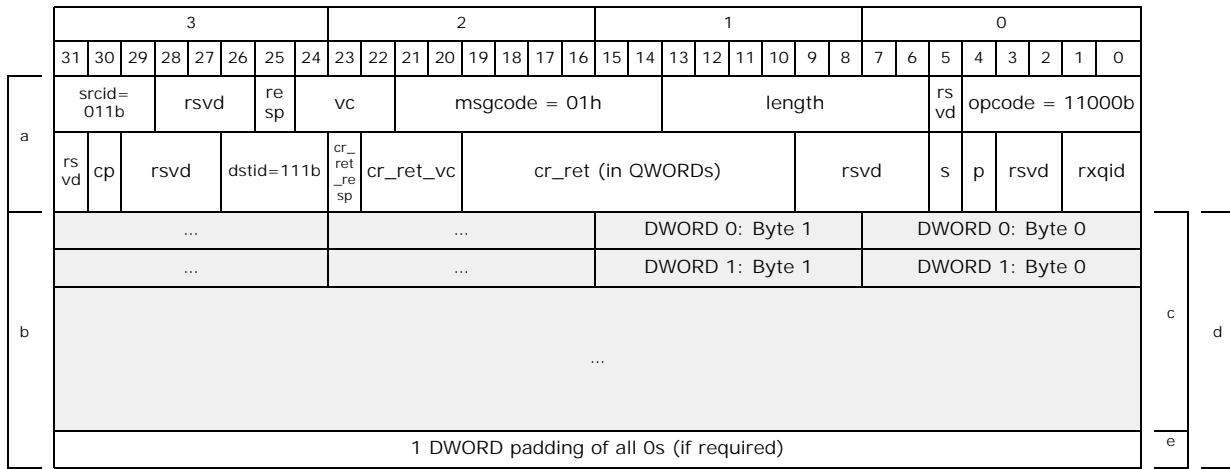
Table 7-13. Common Fields in MPM Header of all MPM with Data Messages on Sideband

Field	Description
opcode	11000b: MPM with Data.
length	MPM Payload length (i.e., 0h for 1 QWORD, 1h for 2 QWORDs, 2h for 3 QWORDs, etc.).
msgcode	Message code as defined in Table 7-12 .
vc	Virtual Channel ID.
resp	0: Request MPM. 1: Response MPM. For a Vendor-defined Management Port Gateway Message with Data, this bit is always 0 (see Section 7.1.2.4.3).
srcid	011b: Indicates Management Port Gateway as source. Other values: Not applicable to MPM. For details on other values of srcid, see Table 7-2 , Table 7-3 , and Table 7-4 .
rxqid	RxQ-ID to which this packet is destined, and RxQ-ID associated with any credits returned in the packet (see Section 8.2.3.1.2 for RxQ details).
dstid	111b: Indicates Management Port Gateway as target. Other values: Not applicable to MPM. For details on other values of dstid, see Table 7-2 , Table 7-3 , and Table 7-4 .
cp	Control parity for the sideband packet header. All fields other than “cp” in the header are protected by Control Parity, and the parity scheme is even (including reserved bits).

7.1.2.4.2 Encapsulated MTP Message

Encapsulated MTP on sideband is an MPM with Data message with a msgcode of 01h.

Figure 7-6. Encapsulated MTP on Sideband



- MPM Header.
- MPM Payload.
- Management Transport Packet (MTP).
- Length in MPM Header.
- DWORD padding.

Table 7-14. Encapsulated MTP on Sideband Fields

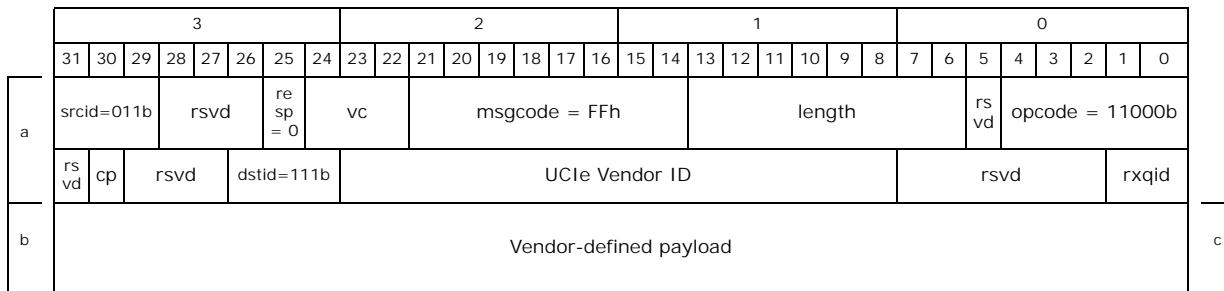
Location	Bit	Description
MPM Header ^a	s	Segmented MTP (see Section 8.2.4.2). The first and middle segments in a segmented MTP have this bit set to 1. The last segment in a segmented MTP will have this bit cleared to 0. An unsegmented MTP also has this bit cleared to 0.
	p	If this is set to 1, there is 1-DWORD padding of all Os added at the end of the packet, to align to a QWORD boundary.
	cr_ret	Value of RxQ credits being returned to the MPG receiving this message, indicated by the rxqid value and its VC: Resp channel indicated via cr_ret_vc/cr_ret_resp fields. 000h indicates 0 credits returned. 001h indicates 1 credit returned. ... 3FEh indicates 1022 credits returned. 3FFh is reserved. If there is no credit being returned, cr_ret fields must be set to 0h.
	cr_ret_vc	VC associated with the credit returned.
	cr_ret_resp	Resp value associated with the credit returned. 0=Request channel credit. 1=Response channel credit.
MPM Payload	—	See Section 8.2 for details. Note that DWORDx:Bytey in Figure 7-6 refers to the corresponding DWORD, Byte defined in the Management Transport Packet in Figure 8-5 .

- See [Section 7.1.2.4.1](#) for details of header fields common to all MPMs with data on the sideband.

7.1.2.4.3 Vendor-defined Management Port Gateway Message

The Vendor-defined Management Port Gateway message with data is defined for custom communication between MPGs on the two ends of a PCIe sideband link. These messages are not part of the Management transport protocol, and these messages start at an MPG and terminate at the MPG on the other end of the PCIe sideband link. These messages share the same RxQ-ID request buffers and credits as encapsulated MTP messages. If an MPG does not support these messages or does not support vendor-defined messages from a given vendor (identified by the PCIe Vendor ID in the header), the MPG silently drops those messages. Length of these Vendor defined messages is subject to the same rules stated in [Section 8.2.5.1.2](#). Ordering of these messages sent over multiple sideband links is subject to the same rules presented in [Section 8.2.4.3](#) for encapsulated MTPs.

Figure 7-7. Vendor-defined Management Port Gateway Message with Data on Sideband



- a. MPM Header.
- b. MPM Payload.
- c. Length in MPM Header.

Table 7-15. Vendor-defined Management Port Gateway Message with Data on Sideband Fields

Location	Field	Description
MPM Header ^a	PCIe Vendor ID	PCIe Consortium-assigned unique ID for each vendor.
MPM Payload	—	Vendor-defined.

a. See [Section 7.1.2.4.1](#) for details of header fields common to all MPMs with data on the sideband.

7.1.2.5 MPMs without Data

Bits [21:14] in the first DWORD of the MPM header of an MPM without Data message form an 8b msgcode that denotes a specific MPM without Data message. [Table 7-16](#) lists the supported msgcodes.

Table 7-16. Supported MPM without Data Messages on Sideband

msgcode	Message
01h	Management Port Gateway Capabilities Message
02h	Credit Return Message
03h	Init Done Message
04h	PM Message
FFh	Vendor-defined Management Port Gateway Message
Others	Reserved

7.1.2.5.1 Common Header Fields of MPM without Data Messages on Sideband

Figure 7-8 shows and Table 7-17 describes the common fields in the MPM header of MPM without data messages on the sideband.

Figure 7-8. Common Fields in MPM Header of all MPM without Data Messages on Sideband

3			2								1								0												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
srcid=011b		rsvd						msgcode								msgcode-specific								rs vd	opcode = 10111b						
rs vd	cp	rsvd	dstid=111b	msgcode-specific																rsvd	msgcode- specific										

Table 7-17. Common Fields in MPM Header of all MPM without Data Messages on Sideband

Field	Description
opcode	10111b: MPM without Data.
msgcode	Message code as defined in Table 7-16.
srcid	011b: Indicates Management Port Gateway as source.
cp	Control parity for the sideband packet header. All fields other than "cp" in the header are protected by Control Parity, and the parity scheme is even (including reserved bits).
dstid	111b: Indicates Management Port Gateway as target.

7.1.2.5.2 Management Port Gateway Capabilities Message

See Section 8.2.3.1.2 for usage of this message during sideband management transport path initialization.

Figure 7-9 shows and Table 7-18 describes the Management Port Gateway Capabilities message format on the sideband.

Figure 7-9. Management Port Gateway Capabilities MPM on Sideband

3			2								1								0												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
srcid=011b		rsvd						msgcode = 01h								NumVC				rsvd				opcode = 10111b							
rs vd	cp	rsvd	dstid=111b	Port ID[15:0]																rsvd											

a. MPM Header.

Table 7-18. Management Port Gateway Capabilities MPM Header Fields on Sideband^a

Field	Description
NumVC	Number of VCs supported by the Management Port Gateway that is transmitting the message.
Port ID	Port ID number value of the Management port associated with the Management Port Gateway that is issuing the message (see Section 8.1.3.6.2.1).

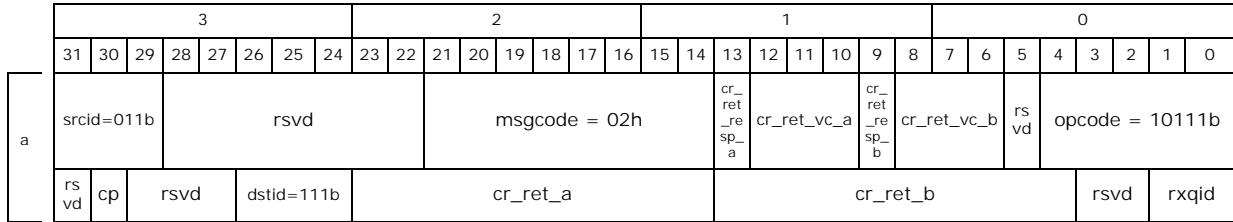
a. See Table 7-17 for details of header fields common to all MPMS without data on the sideband.

7.1.2.5.3 Credit Return Message

See Section 8.2.3.1.2 for usage of this message during sideband management transport path initialization.

Figure 7-10 shows and Table 7-19 describes the Credit Return message format on the sideband. If credit returns a and b carry the same vc:resp fields, then the total credit returned for that rxqid:vc:resp credit type is the sum of cr_ret_a and cr_ret_b.

Figure 7-10. Credit Return MPM on Sideband



a. MPM Header.

Table 7-19. Credit Return MPM Header Fields on Sideband^a

Field	Description
cr_ret_vc_a(b)	VC for which the credit is being returned.
cr_ret_resp_a(b)	Resp value associated with the credit returned. 0=Request channel credit. 1=Response channel credit.
cr_ret_a(b)	Value of credits returned for the RxQ (in the Management Port Gateway transmitting this message) indicated by the rxqid field and the associated VC:Resp channel indicated via cr_ret_vc_a(b)/cr_ret_resp_a(b) fields. 000h indicates 0 credits returned. 001h indicates 1 credit returned. ... 3FEh indicates 1022 credits returned. 3FFh indicates infinite credits. 3FFh value is legal only on credit returns that happen during VC initialization (i.e., before Init Done message is sent) and cannot be used after initialization until the transport path is renegotiated/initialized again. If a receiver detects infinite credit returns after VC initialization and during runtime, it silently ignores it.
rxqid	RxQ-ID of the receiver queue for which the credits are being returned (see Section 8.2.3.1.2 for RxQ details).

a. See Table 7-17 for details of header fields common to all MPMs without data on the sideband.

7.1.2.5.4 Init Done Message

See Section 8.2.5.1.4 for usage of this message during sideband management transport path initialization.

Figure 7-11 shows and Table 7-20 describes the Init Done message format on the sideband.

Figure 7-11. Init Done MPM on Sideband

3								2								1								0								
a	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	srcid=011b				rsvd								msgcode = 03h								rsvd								opcode = 10111b			
	rs	vd	cp	rsvd				dstid=111b				rsvd																rxqid				

a. MPM Header.

Table 7-20. Init Done MPM Header Fields on Sideband^a

Field	Description
rxqid	RxQ-ID of the receiver queue that has completed initializing credits (see Section 8.2.3.1.2 for RxQ details).

a. See Table 7-17 for details of header fields common to all MPMs without data on the sideband.

7.1.2.5.5 PM Message

See Section 8.2.5.1.4 for usage of this message during sideband management transport PM flows.

Figure 7-12 shows and Table 7-21 describes the PM message format on the sideband.

Figure 7-12. PM MPM on Sideband

3								2								1								0												
a	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	srcid=011b				rsvd								msgcode = 04h								pmcode				rsvd								opcode = 10111b			
	rs	vd	cp	rsvd				dstid=111b				rsvd																rxqid								

a. MPM Header.

Table 7-21. PM MPM Header Fields on Sideband^a

Field	Description
pmcode	1h: Wake Req; 2h: Wake ack; 3h: Sleep Req; 4h: Sleep ack; 5h: Sleep nak; Others: Rsvd.
rxqid	RxQ-ID of the receiver queue to which the message applies (see Section 8.2.3.1.2 for RxQ details).

a. See Table 7-17 for details of header fields common to all MPMs without data on the sideband.

7.1.2.5.6 Vendor-defined Management Port Gateway Message

The Vendor-defined Management Port Gateway message without data is defined for custom communication between the MPGs on both ends of a PCIe sideband link. These messages are not part of the management transport protocol, and these messages start at an MPG and terminate at the MPG on the other end of the PCIe sideband link. These messages share the same RxQ-ID request buffers as encapsulated MTP messages. If an MPG does not support these messages or does not

support these messages from a given vendor (identified by the PCIe Vendor ID in the header), the MPG silently drops those messages.

The Vendor-defined Management Port Gateway message without data on the sideband has the format shown in Figure 7-13.

Figure 7-13. Vendor-defined Management Port Gateway MPM without Data on Sideband

3																																										
31 30 29			28 27 26			25 24		23 22		21 20		19 18		17 16		15 14		13 12		11 10		9 8		7 6		5 4		3 2		1 0												
srcid=011b			rsvd			re sp = 0		vc		msgcode = FFh															Vendor-defined		rs vd		opcode = 10111b													
rs vd		cp		rsvd		dstid=111b		PCIe Vendor ID															rsvd		rxqid																	

a. MPM Header.

Table 7-22. MPM Header Vendor-defined Management Port Gateway Message without Data on Sideband^a

Field	Descriptions
Vendor-defined	Defined by the vendor specified in the PCIe Vendor ID field.
vc	Virtual Channel ID.
resp	Vendor-defined Management Port Gateway message without data always uses the Request channel. The value must be 0.
PCIe Vendor ID	PCIe Consortium-assigned unique ID for each vendor.
rxqid	RxQ-ID of the receiver queue to which the message belongs (see Section 8.2.3.1.2 for RxQ details).

a. See Table 7-17 for details of header fields common to all MPMS without data on the sideband.

7.1.2.6 Priority Sideband Traffic Packets

See Section 4.1.5.2 for the rules associated with priority sideband traffic packets. Figure 7-14 shows the packet format. Note that Byte 0[6, 5] must always be 11b for all priority sideband traffic packets to avoid aliasing with other packet types in case there are single bit errors during transmission. Bit 31 is a Parity (P) bit, calculated by XORing bits 0 through 30 of the priority sideband traffic packet (i.e., $P = \oplus \{PSPT[30:0]\}$, where PSPT is the priority sideband traffic packet).

Figure 7-14. Format for Priority Sideband Traffic Packets

Priority Sideband Traffic Packet																																
Bytes	3								2								1								0							
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header / Data																																
Phase0	P	Payload (LSB is on bit 8)															rsvd	1	1	opcode[4:0]												

7.1.3 Flow Control and Data Integrity

Sideband packets can be transferred across FDI, RDI or the PCIe sideband Link. Each of these have independent flow control.

7.1.3.1 Flow Control and Data Integrity over FDI and RDI

For each Transmitter associated with FDI or RDI, a design time parameter of the interface is used to determine the number of credits advertised by the Receiver, with a maximum of 32 credits. Each credit corresponds to 64 bits of header and 64 bits of potentially associated data. Thus, there is only one type of credit for all sideband packets, regardless of how much data they carry. Every Transmitter/Receiver pair has an independent credit loop. For example, on RDI, credits are advertised from Physical Layer to Adapter for sideband packets transmitted from the Adapter to the Physical Layer; and credits are also advertised from Adapter to the Physical Layer for sideband packets transmitted from the Physical Layer to the Adapter.

The Transmitter must check for available credits before sending Register Access requests and Messages. The Transmitter must not check for credits before sending Register Access Completions, and the Receiver must guarantee unconditional sinking for any Register Access Completion packets. Messages carrying requests or responses consume a credit on FDI and RDI, but they must be guaranteed to make forward progress by the Receiver and not get blocked behind Register Access requests. Both RDI and FDI give a dedicated signal for sideband credit returns across those interfaces.

All Receivers associated with RDI and FDI must check received messages for data or control parity errors, and these errors must be mapped to Uncorrectable Internal Errors (UIE) and transition RDI to LinkError state. All receivers of the Priority Sideband Traffic Packet (PSTP) must check for parity errors, and these errors must be mapped to Uncorrectable Internal Errors (UIE) and transition RDI to LinkError state.

When supporting Management Port Messages over sideband, the Physical Layer maintains separate credited buffers (which is a design time parameter) per RxQ-ID it supports to which it can receive Management Port Messages from Management Port Gateway over the RDI configuration bus. Whether received over FDI or RDI, Management Port Messages are always sunk unconditionally in the Management Port Gateway.

7.1.3.2 Flow Control and Data Integrity over UCIE sideband Link between dies

The BER of the sideband Link is 1e-27 or better. Hence, no retry mechanism is provided for the sideband packets. Receivers of sideband packets must check for Data or Control parity errors, and any of these errors is mapped to a fatal UIE.

7.1.3.3 End-to-End flow control and forward progress for UCIE Link sideband

It is important for deadlock avoidance to ensure that there is sufficient space at the Receiver to sink all possible outstanding requests from the Transmitter, so that the requests do not get blocked at any intermediate buffers that would thereby prevent subsequent completions from making progress.

Sideband access for Remote Link partner's Adapter or Physical Layer registers is only accessible via the indirect mailbox mechanism, and the number of outstanding transactions is limited to four at a time. Although four credits are provisioned, there is only a single mailbox register, and this limits the number of outstanding requests that can use this mechanism to one at a time. The extra credits allow additional debug-related register access requests in case of register access timeouts. These credits are separate from local FDI or RDI accesses, and thus the Physical Layer must provision for sinking at least one register access request and completion each from remote die and local Adapter in addition to other sideband request credits (see Implementation Note below). The Adapter provisions for at least four remote register access requests from remote die Adapter. Each credit corresponds to 64b of header and 64b of data. Even requests that send no data or only send 32b of data consume one credit. Register Access completions do not consume a credit and must always sink.

If Management Transport Protocol is not supported, the Adapter credit counters for register access request are initialized to 4 on Domain Reset exit OR whenever RDI transitions from Reset to Active.

If Management Transport Protocol is supported, the Adapter credit counters for register access request are initialized to 4 on [Domain Reset exit] OR whenever [RDI transitions from Reset to Active AND SB_MGMT_UP=0].

It is permitted to send an extra (N-4) credit returns to remote Link partner if a PCIe implementation is capable of sinking a total of N requests once RDI has transitioned to Active state. The Adapter must implement a saturating credit counter capable of accumulating at least 4 credits, and hence prevent excess credit returns from overflowing the counter.

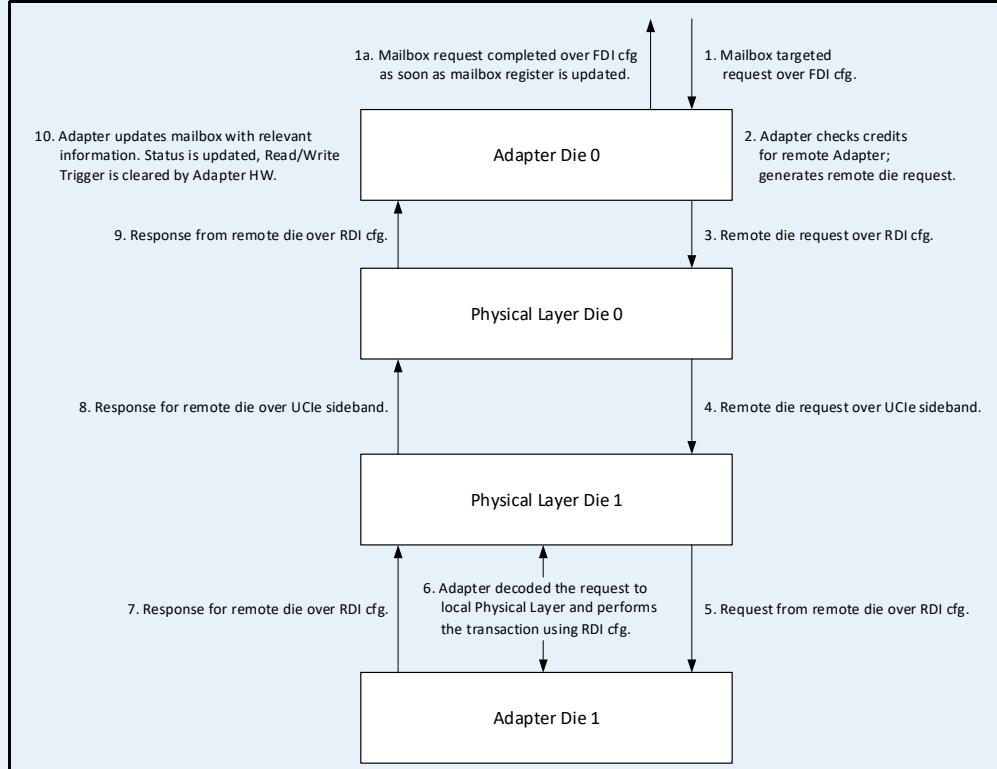
All other messages except Vendor Defined messages must always sink and make forward progress, and not block any messages on the sideband interface behind them. All Link Management message requests have an associated response, and the source of these messages must only have one outstanding request at a time (i.e., one outstanding message per "Link Management Request" MsgCode encoding). Priority Sideband Traffic Packets (PSTPs) must always be accepted and make forward progress — there is no flow control check at the Transmitter for them.

For vendor defined messages, there must be a vendor defined cap on the number of outstanding messages, and the Receiver must guarantee sufficient space so as to not block any messages behind the vendor defined messages on any of the interfaces.

IMPLEMENTATION NOTE

Figure 7-15 shows an example of an end-to-end register access request to remote die and the corresponding completion returning back.

**Figure 7-15. Example Flow for Remote Register Access Request
(Local FDI/RDI Credit Checks Are Not Explicitly Shown)**



In Step 1 shown in Figure 7-15, the Protocol Layer checks for FDI credits before sending the request to Adapter Die 0. Adapter Die 0 completes the mailbox request as soon as the mailbox register is updated (shown in Step 1a). FDI credits are returned once its internal buffer space is free. In Step 2, Adapter Die 0 checks credits for remote Adapter as well as credits for local RDI before sending the remote die request to Physical Layer Die 0 in Step 3. Physical Layer schedules the request over UCle sideband and returns the RDI credit to Adapter Die 0 once it has freed up its internal buffer space.