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

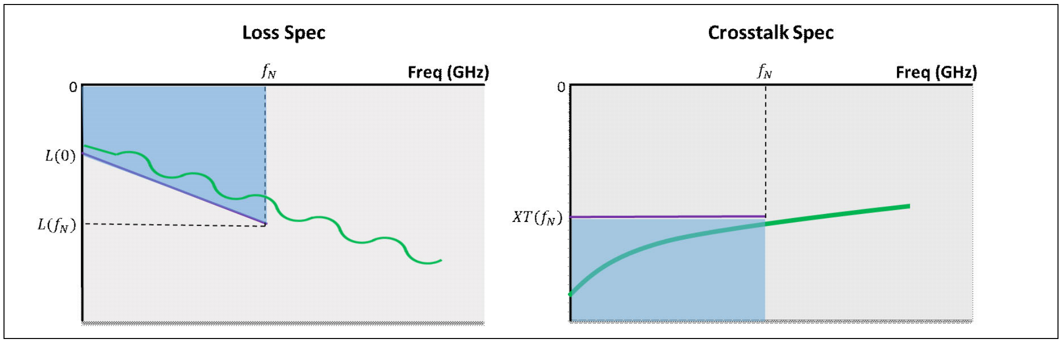
Signal Name	Count	Description
RXDATASBRD	1	Redundant Sideband Receiver Data
TXCKSBRD	1	Redundant Sideband Transmit Clock
RXCKSBRD	1	Redundant Sideband Receive Clock
Power and Voltage		
VSS		Ground Reference
VCCIO		I/O supply
VCCFWDIO		Forwarded power supply from remote Transmitter supply to local Receiver 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 5-18. 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 5-18 is a representative passing signal.

Figure 5-18. 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 5-19.

Figure 5-19. Viewer Orientation Looking at the Defined UCIe Bump Matrix

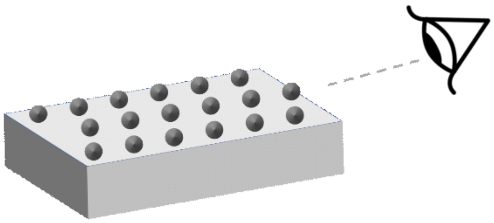


Figure 5-20, Figure 5-21, and Figure 5-22 show the reference bump matrix for the 10-column, 16-column, and 8-column x64 Advanced Package Modules, respectively. The lower left corner of the bump map will be considered “origin” of a bump matrix and the leftmost column is Column 0.

It is strongly recommended to follow the bump matrices provided in Figure 5-20, Figure 5-21, and Figure 5-22 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 5-21 and Figure 5-22, 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 5-20. 10-column x64 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
vss		vss		vccio		vccio		vss	
	vss		vccio		vccio		vss		vss
vss		vss		vccio		vccio		vss	
	rxcksbRD		rxcksb		vccio		rxdatasb		rxdatasbRD
txdatasbRD		txdatasb		vccio		txcksb		txcksbRD	
	rxdata50		rxdata35		rxdata29		rxdata14		rxdataRD0
rxdataRD3		rxdata49		rxdata34		rxdata28		rxdata13	
	rxdata51		rxdata36		rxdata30		rxdata15		vss
rxdata63		vccio		rxdata33		vccio		rxdata12	
	rxdata52		vss		rxdata31		vss		rxdata0
vss		rxdata48		rxdata32		rxdata27		rxdata11	
	rxdata53		rxdata37		rxdataRD1		rxdata16		rxdata1
rxdata62		rxdata47		rxdataRD2		rxdata26		rxdata10	
	rxdata54		rxdata38		vss		rxdata17		vss
rxdata61		rxdata46		vccio		rxdata25		rxdata9	
	rxdata55		rxdata39		rxckRD		rxdata18		rxdata2
vss		rxdata45		rxvldRD		rxdata24		rxdata8	
	rxdata56		vss		rxckn		rxdata19		rxdata3
rxdata60		rxdata44		rxvld		vss		rxdata7	
	rxdata57		rxdata40		rxckp		rxdata20		vss
rxdata59		rxdata43		rxtrk		rxdata23		rxdata6	
	rxdata58		rxdata41		vss		rxdata21		rxdata4
vss		rxdata42		vccio		rxdata22		rxdata5	
	vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio
vccio		txdata21		vccio		txdata41		txdata58	
	txdata5		txdata22		vss		txdata42		vss
txdata4		txdata20		txckp		txdata40		txdata57	
	txdata6		txdata23		txtrk		txdata43		txdata59
vss		txdata19		txckn		vss		txdata56	
	txdata7		vss		txvld		txdata44		txdata60
txdata3		txdata18		txckRD		txdata39		txdata55	
	txdata8		txdata24		txvldRD		txdata45		vss
txdata2		txdata17		vccio		txdata38		txdata54	
	txdata9		txdata25		vss		txdata46		txdata61
vccio		vccio		vccio		vccio		vccio	
	txdata10		txdata26		txdataRD2		txdata47		txdata62
txdata1		txdata16		txdataRD1		txdata37		txdata53	
	txdata11		txdata27		txdata32		txdata48		vss
txdata0		vss		txdata31		vss		txdata52	
	txdata12		vss		txdata33		vss		txdata63
vss		txdata15		txdata30		txdata36		txdata51	
	txdata13		txdata28		txdata34		txdata49		txdataRD3
txdataRD0		txdata14		txdata29		txdata35		txdata50	
	vccio		vccio		vccio		vccio		vccio
vccio		vccio		vccio		vccio		vccio	
Die Edge									

Note: In Figure 5-20, at 45-um pitch, the module depth of the 10-column reference bump matrix as shown is approximately 1043 um.

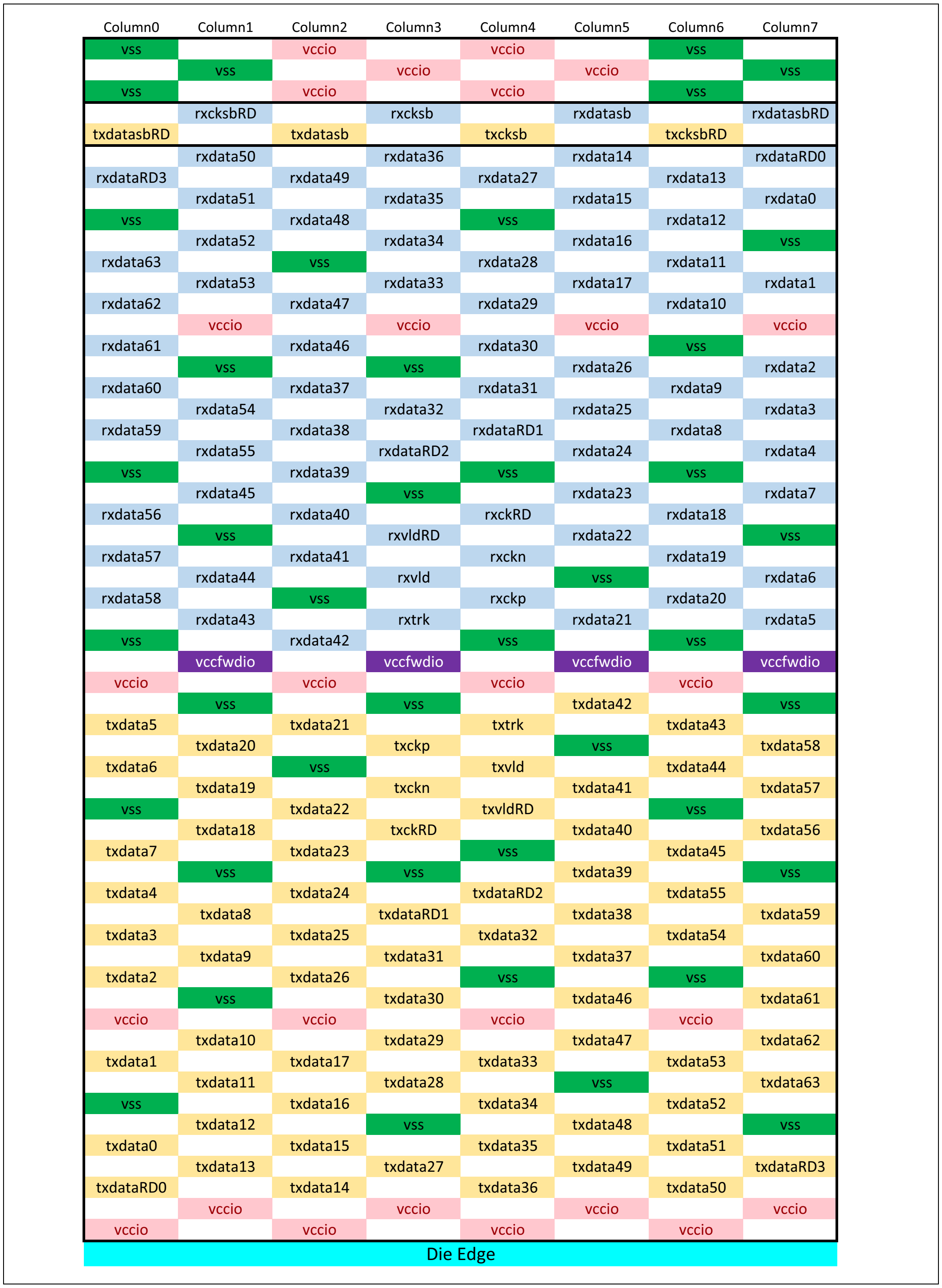
Figure 5-21. 16-column x64 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11	Column12	Column13	Column14	Column15
	vss		vss		vccio		vccio		vccio		vccio		vss		vss
vss		vss		vccio		vccio		vccio		vccio		vss		vss	
	vss		rxcksbRD		rxcksb		vss		rxdatasb		rxdatasbRD		vss		vss
vss		txdatasbRD		txdatasb		vss		vss		txcksb		txcksbRD		vss	
	rxdata54		rxdata50		rxdata35		vss		rxdata29		rxdata14		rxdata11		rxdataRD0
vss		rxdata52		rxdata49		rxdata34		rxdataRD1		rxdata28		rxdata13		rxdata9	
	rxdata55		rxdata51		rxdata36		rxdataRD2		rxdata30		rxdata15		rxdata10		vss
rxdataRD3		rxdata53		rxdata48		rxdata33		vss		rxdata27		rxdata12		rxdata8	
	rxdata61		vss		rxdata37		vss		rxdata31		rxdata16		vss		rxdata0
rxdata63		rxdata56		rxdata47		rxdata32		rxckRD		rxdata26		rxdata17		rxdata2	
	rxdata60		rxdata46		vss		rxvldRD		rxdata25		vss		rxdata7		vss
vss		rxdata57		rxdata43		rxdata38		rxckn		rxdata23		rxdata18		rxdata3	
	rxdata59		rxdata45		rxdata40		rxvld		vss		rxdata20		rxdata6		rxdata1
rxdata62		rxdata58		rxdata42		rxdata39		rxckp		rxdata22		rxdata19		rxdata4	
	vss		rxdata44		rxdata41		rxtrk		rxdata24		rxdata21		rxdata5		vss
vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	
	vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio
vss		txdata5		txdata21		txdata24		txtrk		txdata41		txdata44		vss	
	txdata4		txdata19		txdata22		txckp		txdata39		txdata42		txdata58		txdata62
txdata1		txdata6		txdata20		vss		txvld		txdata40		txdata45		txdata59	
	txdata3		txdata18		txdata23		txckn		txdata38		txdata43		txdata57		vss
vss		txdata7		vss		txdata25		txvldRD		vss		txdata46		txdata60	
	txdata2		txdata17		txdata26		txckRD		txdata32		txdata47		txdata56		txdata63
txdata0		vss		txdata16		txdata31		vss		txdata37		vss		txdata61	
	txdata8		txdata12		txdata27		vss		txdata33		txdata48		txdata53		txdataRD3
vss		txdata10		txdata15		txdata30		txdataRD2		txdata36		txdata51		txdata55	
	txdata9		txdata13		txdata28		txdataRD1		txdata34		txdata49		txdata52		vss
txdataRD0		txdata11		txdata14		txdata29		vss		txdata35		txdata50		txdata54	
	vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio
vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio	

Die Edge

Note: In Figure 5-21, at 25-um pitch, the module depth of the 16-column reference bump matrix as shown is approximately 388 um.

Figure 5-22. 8-column x64 Advanced Package Bump Map



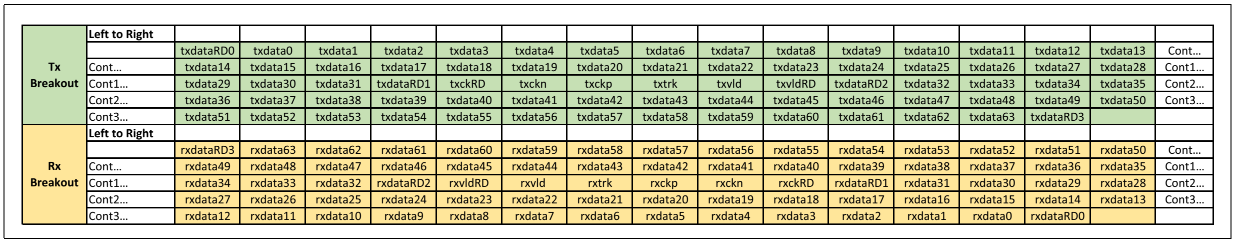
Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7
vss		vccio		vccio		vss	
	vss		vccio		vccio		vss
vss		vccio		vccio		vss	
	rxcksbRD		rxcksb		rxdatasb		rxdatasbRD
txdatasbRD		txdatasb		txcksb		txcksbRD	
	rxdata50		rxdata36		rxdata14		rxdataRD0
rxdataRD3		rxdata49		rxdata27		rxdata13	
	rxdata51		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		vss		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 5-22, at 55-um pitch, the module depth of the 8-column reference bump matrix as shown is approximately 1,585 um.

Figure 5-23 shows the signal exit order for the 10-column x64 Advanced Package bump map.

Figure 5-23. 10-column x64 Advanced Package Bump map: Signal exit order



Tx Breakout	**Left to Right**																
		txdataRD0	txdata0	txdata1	txdata2	txdata3	txdata4	txdata5	txdata6	txdata7	txdata8	txdata9	txdata10	txdata11	txdata12	txdata13	Cont…
	Cont…	txdata14	txdata15	txdata16	txdata17	txdata18	txdata19	txdata20	txdata21	txdata22	txdata23	txdata24	txdata25	txdata26	txdata27	txdata28	Cont1…
	Cont1…	txdata29	txdata30	txdata31	txdataRD1	txckRD	txckn	txckp	txtrk	txvld	txvldRD	txdataRD2	txdata32	txdata33	txdata34	txdata35	Cont2…
	Cont2…	txdata36	txdata37	txdata38	txdata39	txdata40	txdata41	txdata42	txdata43	txdata44	txdata45	txdata46	txdata47	txdata48	txdata49	txdata50	Cont3…
	Cont3…	txdata51	txdata52	txdata53	txdata54	txdata55	txdata56	txdata57	txdata58	txdata59	txdata60	txdata61	txdata62	txdata63	txdataRD3		
Rx Breakout	**Left to Right**																
		rxdataRD3	rxdata63	rxdata62	rxdata61	rxdata60	rxdata59	rxdata58	rxdata57	rxdata56	rxdata55	rxdata54	rxdata53	rxdata52	rxdata51	rxdata50	Cont…
	Cont…	rxdata49	rxdata48	rxdata47	rxdata46	rxdata45	rxdata44	rxdata43	rxdata42	rxdata41	rxdata40	rxdata39	rxdata38	rxdata37	rxdata36	rxdata35	Cont1…
	Cont1…	rxdata34	rxdata33	rxdata32	rxdataRD2	rxvldRD	rxvld	rxtrk	rxckp	rxckn	rxckRD	rxdataRD1	rxdata31	rxdata30	rxdata29	rxdata28	Cont2…
	Cont2…	rxdata27	rxdata26	rxdata25	rxdata24	rxdata23	rxdata22	rxdata21	rxdata20	rxdata19	rxdata18	rxdata17	rxdata16	rxdata15	rxdata14	rxdata13	Cont3…
	Cont3…	rxdata12	rxdata11	rxdata10	rxdata9	rxdata8	rxdata7	rxdata6	rxdata5	rxdata4	rxdata3	rxdata2	rxdata1	rxdata0	rxdataRD0		

IMPLEMENTATION NOTE — x64 Bump Maps for Max Speed

Three reference bump maps in Figure 5-20, Figure 5-21, and Figure 5-22 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.

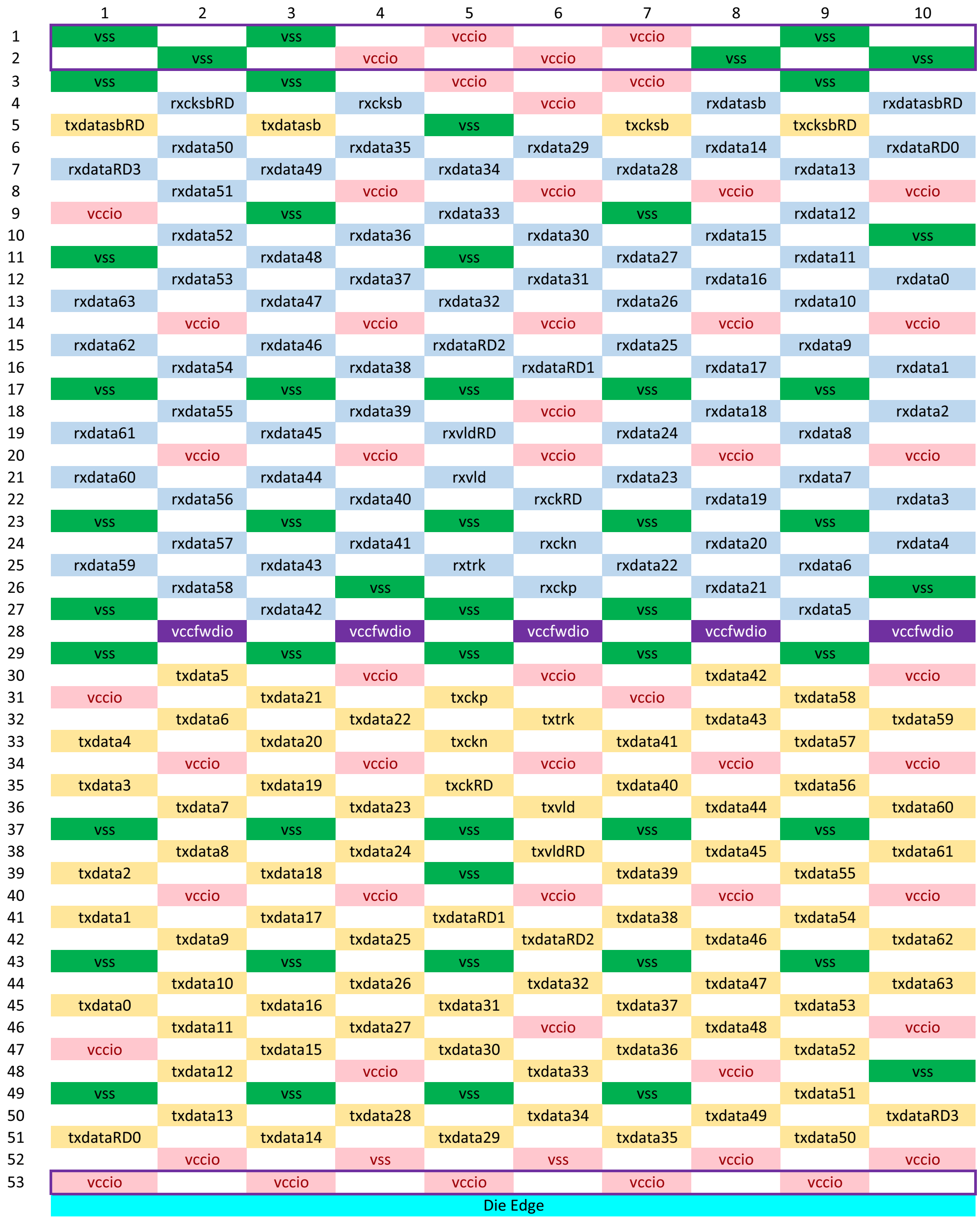
Table 5-13. Bump Map Options and the Recommended Bump Pitch Range and Max Speed

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 5-13 summarizes the corresponding max speed for these bump map options and their recommended bump pitch ranges.

Bump maps in Figure 5-24, Figure 5-25, and Figure 5-26 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.

Figure 5-24. 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		vccio		rxdatasb		rxdatasbRD
5	txdatasbRD		txdatasb		vss		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		vss		vss		vccio		vccio
53	vccio		vccio		vccio		vccio		vccio	
	Die Edge									

Note: In Figure 5-24, 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.

Figure 5-25. 16-column x64 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	txdatasbRD		rxcksbRD		txdatasb		rxcksb		txcksb		rxdatasb		txcksbRD		rxdatasbRD	
4		rxdata54		rxdata50		rxdata35		vss		rxdata29		rxdata14		rxdata11		rxdataRD0
5	rxdataRD3		rxdata52		rxdata49		rxdata34		vss		rxdata28		rxdata13		rxdata9	
6		rxdata55		rxdata51		rxdata36		rxdataRD2		rxdata30		rxdata15		rxdata10		vss
7	vss		rxdata53		rxdata48		rxdata33		rxdataRD1		rxdata27		rxdata12		rxdata8	
8		rxdata61		vss		rxdata37		vss		rxdata31		vss		rxdata7		vss
9	rxdata63		rxdata56		vss		rxdata32		vss		rxdata26		vss		rxdata2	
10		vccio		rxdata46		vccio		rxvldRD		vccio		rxdata16		vccio		rxdata0
11	vss		vccio		rxdata47		vccio		rxckRD		vccio		rxdata17		vccio	
12		rxdata60		rxdata45		rxdata40		rxvld		rxdata25		rxdata20		rxdata6		vss
13	rxdata62		rxdata57		rxdata43		rxdata38		rxckn		rxdata23		rxdata18		rxdata3	
14		rxdata59		rxdata44		rxdata41		rxtrk		rxdata24		rxdata21		rxdata5		rxdata1
15	vss		rxdata58		rxdata42		rxdata39		rxckp		rxdata22		rxdata19		rxdata4	
16		vccfwdio		vss		vccfwdio		vss		vccfwdio		vss		vccfwdio		vss
17	vss		vccfwdio		vss		vccfwdio		vss		vccfwdio		vss		vccfwdio	
18		txdata4		txdata19		txdata22		txckp		txdata39		txdata42		txdata58		vss
19	txdata1		txdata5		txdata21		txdata24		txtrk		txdata41		txdata44		txdata59	
20		txdata3		txdata18		txdata23		txckn		txdata38		txdata43		txdata57		txdata62
21	vss		txdata6		txdata20		txdata25		txvld		txdata40		txdata45		txdata60	
22		vccio		txdata17		vccio		txckRD		vccio		txdata47		vccio		vss
23	txdata0		vccio		txdata16		vccio		txvldRD		vccio		txdata46		vccio	
24		txdata2		vss		txdata26		vss		txdata32		vss		txdata56		txdata63
25	vss		txdata7		vss		txdata31		vss		txdata37		vss		txdata61	
26		txdata8		txdata12		txdata27		txdataRD1		txdata33		txdata48		txdata53		vss
27	vss		txdata10		txdata15		txdata30		txdataRD2		txdata36		txdata51		txdata55	
28		txdata9		txdata13		txdata28		vss		txdata34		txdata49		txdata52		txdataRD3
29	txdataRD0		txdata11		txdata14		txdata29		vss		txdata35		txdata50		txdata54	
30		vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio
31	vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio	
	Die Edge															

Note: In Figure 5-25, 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.

Figure 5-26. 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	
2		vss		vccio		vccio		vss
3	vss		vccio		vccio		vss	
4		rxcksbRD		rxcksb		rxdatasb		rxdatasbRD
5	txdatasbRD		txdatasb		txcksb		txcksbRD	
6		rxdata50		vss		rxdata14		rxdataRD0
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		vccio		rxdata18	
24		rxdata45		rxvldRD		rxdata23		rxdata7
25	vss		rxdata40		vss		vss	
26		vccio		rxvld		rxdata22		rxdata6
27	rxdata57		vss		rxckRD		rxdata19	
28		rxdata44		vccio		vccio		vccio
29	rxdata58		rxdata41		rxckn		rxdata20	
30		rxdata43		rxtrk		rxdata21		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	txdataRD0		txdata14		vss		txdata50	
60		vccio		vccio		vccio		vccio
61	vccio		vccio		vccio		vccio	
	Die Edge							

Note: In Figure 5-26, 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.

5.7.2.3 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 5-27, Figure 5-28, and Figure 5-29 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 5-27, Figure 5-28, and Figure 5-29 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 5-27. 10-column x32 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
vss		vss		vccio		vccio		vss	
	vss		vccio		vccio		vss		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		rxdataRD1		rxdata15		rxdataRD0
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
txdataRD0		txdata14		txdata31		rxdata22		rxdata5	
	vccio		vccio		vccfwdio		vccfwdio		vccfwdio
vccio		vccio		vccio		vccfwdio		vccfwdio	

Die Edge

Note: In Figure 5-27, at 45-um pitch, the module depth of the 10-column reference bump matrix as shown is approximately 680.5 um.

Figure 5-28. 16-column x32 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11	Column12	Column13	Column14	Column15
	vss		vss		vss		vss		vss		vss		vss		vss
vccio		vccio		vccio		vccio		vccio		vccio		vccio		vccio	
	txdatasbRD		rxcksbRD		txdatasb		rxcksb		txcksb		rxdatasb		txcksbRD		rxdatasbRD
vss		txdata5		vss		vss		vss		vss		vss		vss	
	txdata4		txdata20		txdata22		txtrk		rxdata30		rxdata26		rxdata13		rxdataRD0
txdata2		vss		txdata21		txckp		rxdata31		rxdata27		rxdata14		rxdata11	
	txdata3		txdata17		txdata23		vss		rxdata29		vss		rxdata12		vss
txdata1		txdata6		txdata19		txckn		vss		rxdata28		rxdata15		rxdata10	
	txdata8		vss		vss		txvld		rxdataRD1		rxdata25		vss		rxdata0
vss		txdata7		txdata18		txckRD		rxvldRD		rxdata24		rxdata16		rxdata9	
	txdata9		txdata16		txdata24		txvldRD		rxckRD		rxdata18		rxdata7		vss
txdata0		vss		txdata25		txdataRD1		rxvld		vss		rxdata17		rxdata8	
	txdata10		txdata15		txdata28		vss		rxckn		rxdata19		rxdata6		rxdata1
vss		txdata12		vss		txdata29		vss		rxdata23		vss		rxdata3	
	txdata11		txdata14		txdata27		txdata31		rxckp		rxdata21		rxdata5		rxdata2
txdataRD0		txdata13		txdata26		txdata30		rxtrk		rxdata22		rxdata20		rxdata4	
	vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio
vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	

Die Edge

Note: In Figure 5-28, at 25-um pitch, the module depth of the 16-column reference bump matrix as shown is approximately 237.5 um.

Figure 5-29. 8-column x32 Advanced Package Bump Map

Column0	Column1	Column2	Column3	Column4	Column5	Column6	Column7
	vss		vss		vss		vss
vss		vss		vss		vss	
	vccio		vccio		vccio		vccio
rxcksbRD		rxcksb		rxdatasb		rxdatasbRD	
	txdatasbRD		txdatasb		txcksb		txcksbRD
txdata5		txdata23		rxdata30		rxdata13	
	txdata22		txckp		rxdata14		rxdataRD0
txdata6		vss		vss		rxdata12	
	txdata21		txckn		rxdata15		vss
txdata7		txdata24		rxdata31		rxdata11	
	txdata20		vss		vss		rxdata0
vss		txdata25		rxdataRD1		rxdata10	
	txdata19		txckRD		rxdata16		rxdata1
txdata4		vss		vss		rxdata9	
	txdata18		txtrk		rxdata29		vss
txdata3		txdata26		rxvldRD		rxdata8	
	txdata17		vss		rxdata28		rxdata2
vss		txdata27		rxvld		vss	
	txdata8		txvld		rxdata27		rxdata3
vccio		vccio		vccfwdio		vccfwdio	
	txdata9		txvldRD		rxdata26		vss
txdata2		txdata28		rxtrk		rxdata17	
	txdata10		vss		vss		rxdata4
vss		txdata29		rxckRD		rxdata18	
	txdata11		txdataRD1		rxdata25		rxdata7
txdata1		txdata16		vss		rxdata19	
	vss		txdata31		rxdata24		vss
txdata0		txdata15		rxckn		rxdata20	
	txdata12		vss		vss		rxdata6
vss		txdata14		rxckp		rxdata21	
	txdata13		txdata30		rxdata23		rxdata5
txdataRD0		vss		vss		rxdata22	
	vccio		vccio		vccfwdio		vccfwdio
vccio		vccio		vccfwdio		vccfwdio	
Die Edge							

Note: In Figure 5-29, at 55-um pitch, the module depth of the 8-column reference bump matrix as shown is approximately 962.5 um.

Figure 5-30 shows the signal exit order for the 10-column x32 Advanced Package bump map.

Figure 5-30. 10-column x32 Advanced Package Bump Map: Signal Exit Order

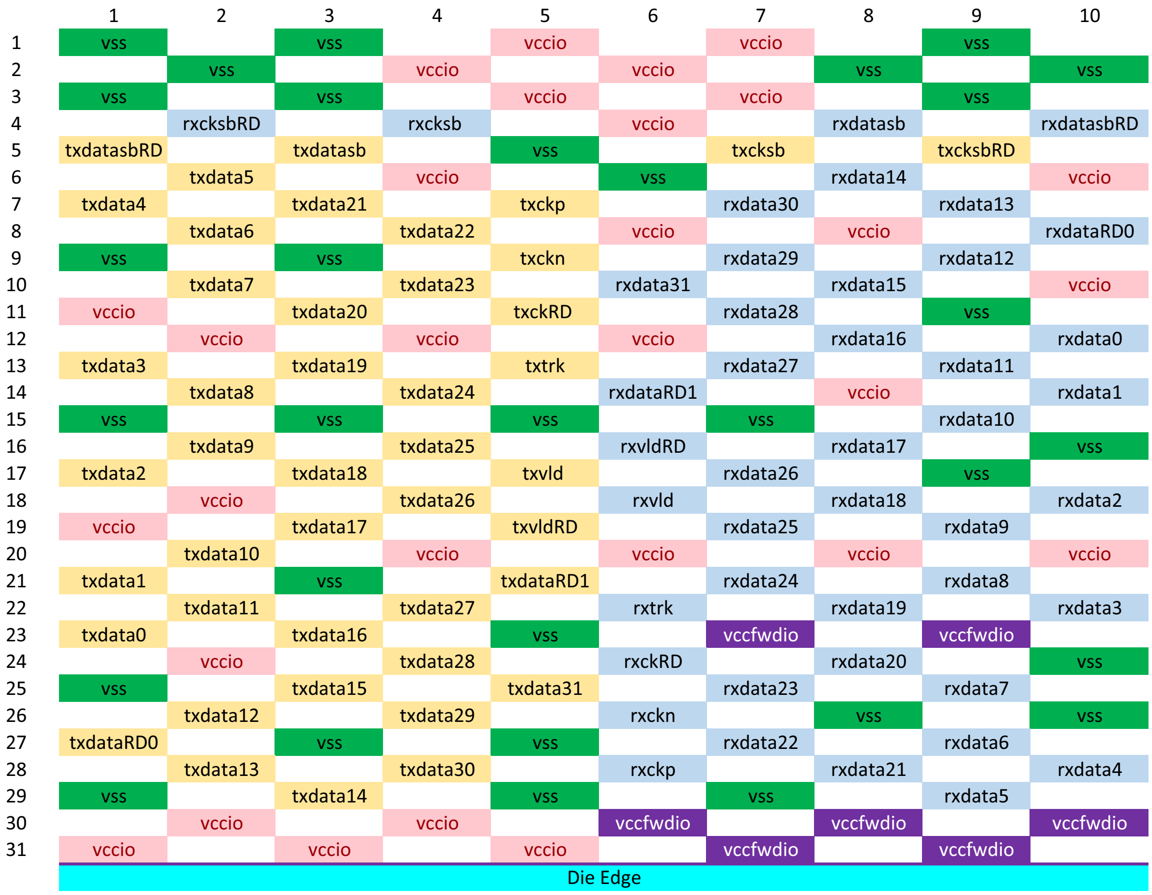
Tx Breakout	Left to Right											
		txdataRD0	txdata0	txdata1	txdata2	txdata3	txdata4	txdata5	txdata6	txdata7	txdata8	Cont…
	Cont…	txdata9	txdata10	txdata11	txdata12	txdata13	txdata14	txdata15	txdata16	txdata17	txdata18	Cont1…
	Cont1…	txdata19	txdata20	txdata21	txdata22	txdata23	txdata24	txdata25	txdata26	txdata27	txdata28	Cont2…
	Cont2…	txdata29	txdata30	txdata31	txdataRD1	txvldRD	txvld	txtrk	txckRD	txckn	txckp	
Rx Breakout	Left to Right											
		rxckp	rxckn	rxckRD	rxtrk	rxvld	rxvldRD	rxdataRD1	rxdata31	rxdata30	rxdata29	Cont…
	Cont…	rxdata28	rxdata27	rxdata26	rxdata25	rxdata24	rxdata23	rxdata22	rxdata21	rxdata20	rxdata19	Cont1…
	Cont1…	rxdata18	rxdata17	rxdata16	rxdata15	rxdata14	rxdata13	rxdata12	rxdata11	rxdata10	rxdata9	Cont2…
	Cont2…	rxdata8	rxdata7	rxdata6	rxdata5	rxdata4	rxdata3	rxdata2	rxdata1	rxdata0	rxdataRD0	

IMPLEMENTATION NOTE — x32 Bump Maps for Max Speed

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 5-31, Figure 5-32, and Figure 5-33 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.

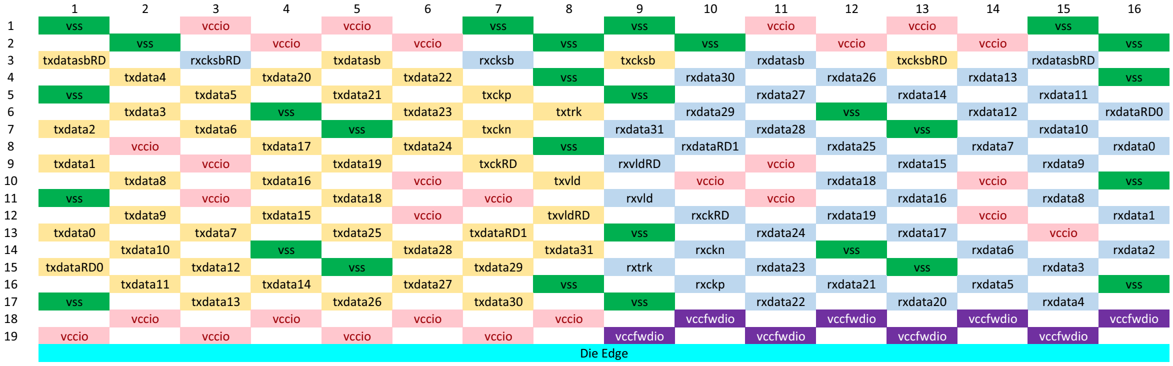
Figure 5-31. 10-column x32 Advanced Package Bump Map Example for 32 GT/s Implementation



	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		vccio		rxdatasb		rxdatasbRD
5	txdatasbRD		txdatasb		vss		txcksb		txcksbRD	
6		txdata5		vccio		vss		rxdata14		vccio
7	txdata4		txdata21		txckp		rxdata30		rxdata13	
8		txdata6		txdata22		vccio		vccio		rxdataRD0
9	vss		vss		txckn		rxdata29		rxdata12	
10		txdata7		txdata23		rxdata31		rxdata15		vccio
11	vccio		txdata20		txckRD		rxdata28		vss	
12		vccio		vccio		vccio		rxdata16		rxdata0
13	txdata3		txdata19		txtrk		rxdata27		rxdata11	
14		txdata8		txdata24		rxdataRD1		vccio		rxdata1
15	vss		vss		vss		vss		rxdata10	
16		txdata9		txdata25		rxvldRD		rxdata17		vss
17	txdata2		txdata18		txvld		rxdata26		vss	
18		vccio		txdata26		rxvld		rxdata18		rxdata2
19	vccio		txdata17		txvldRD		rxdata25		rxdata9	
20		txdata10		vccio		vccio		vccio		vccio
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		rxdata23		rxdata7	
26		txdata12		txdata29		rxckn		vss		vss
27	txdataRD0		vss		vss		rxdata22		rxdata6	
28		txdata13		txdata30		rxckp		rxdata21		rxdata4
29	vss		txdata14		vss		vss		rxdata5	
30		vccio		vccio		vccfwdio		vccfwdio		vccfwdio
31	vccio		vccio		vccio		vccfwdio		vccfwdio	
	Die Edge									

Note: In Figure 5-31, 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.

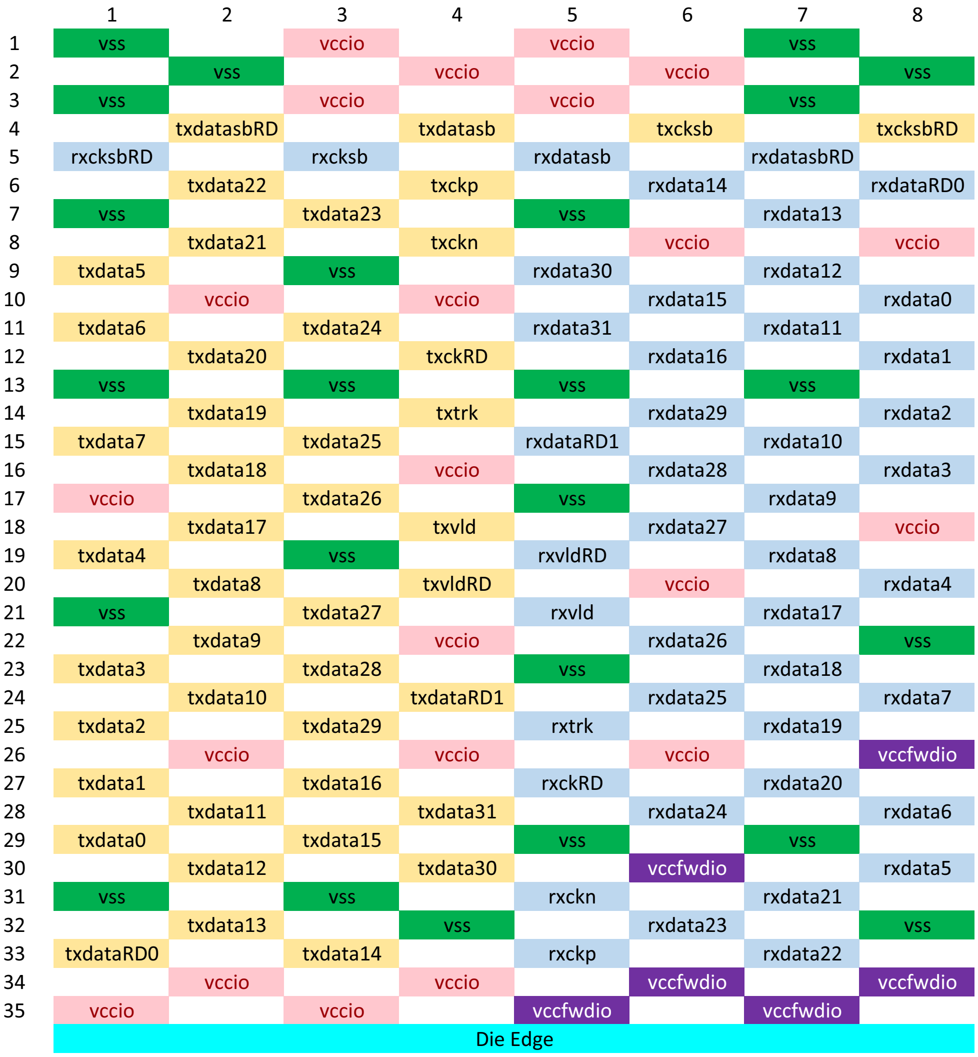
Figure 5-32. 16-column x32 Advanced Package Bump Map Example for 16 GT/s Implementation



	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	txdatasbRD		rxcksbRD		txdatasb		rxcksb		txcksb		rxdatasb		txcksbRD		rxdatasbRD	
4		txdata4		txdata20		txdata22		vss		rxdata30		rxdata26		rxdata13		vss
5	vss		txdata5		txdata21		txckp		vss		rxdata27		rxdata14		rxdata11	
6		txdata3		vss		txdata23		txtrk		rxdata29		vss		rxdata12		rxdataRD0
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		rxckRD		rxdata19		vccio		rxdata1
13	txdata0		txdata7		txdata25		txdataRD1		vss		rxdata24		rxdata17		vccio	
14		txdata10		vss		txdata28		txdata31		rxckn		vss		rxdata6		rxdata2
15	txdataRD0		txdata12		vss		txdata29		rxtrk		rxdata23		vss		rxdata3	
16		txdata11		txdata14		txdata27		vss		rxckp		rxdata21		rxdata5		vss
17	vss		txdata13		txdata26		txdata30		vss		rxdata22		rxdata20		rxdata4	
18		vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio
19	vccio		vccio		vccio		vccio		vccfwdio		vccfwdio		vccfwdio		vccfwdio	
	Die Edge															

Note: In Figure 5-32, at 25-um pitch, the module depth of the 16-column bump map as shown is approximately 250 um. 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.

Figure 5-33. 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		vccfwdio
27	txdata1		txdata16		rxckRD		rxdata20	
28		txdata11		txdata31		rxdata24		rxdata6
29	txdata0		txdata15		vss		vss	
30		txdata12		txdata30		vccfwdio		rxdata5
31	vss		vss		rxckn		rxdata21	
32		txdata13		vss		rxdata23		vss
33	txdataRD0		txdata14		rxckp		rxdata22	
34		vccio		vccio		vccfwdio		vccfwdio
35	vccio		vccio		vccfwdio		vccfwdio	
	Die Edge							

Note: In Figure 5-33, 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.

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 5-14 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 5-14. Maximum Systematic Lane-to-lane Length Mismatch in um between the Reference Bump Maps in the Implementation Note

Rx Tx	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
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

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 5-34 and Figure 5-35, 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-**, **VLDRD-**, **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 5-15 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 5-15. x64 and x32 Advanced Package Connectivity Matrix

–			Normal Module		Mirrored Module	
			Rx			
			x64	x32	x64	x32
Normal Module	Tx	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 5-34 and Figure 5-35 show examples of normal and mirrored x64-to-x32 and x32-to-x32 Advanced Package Module connections, respectively.

Figure 5-34. Example of Normal and Mirrored x64-to-x32 Advanced Package Module Connection

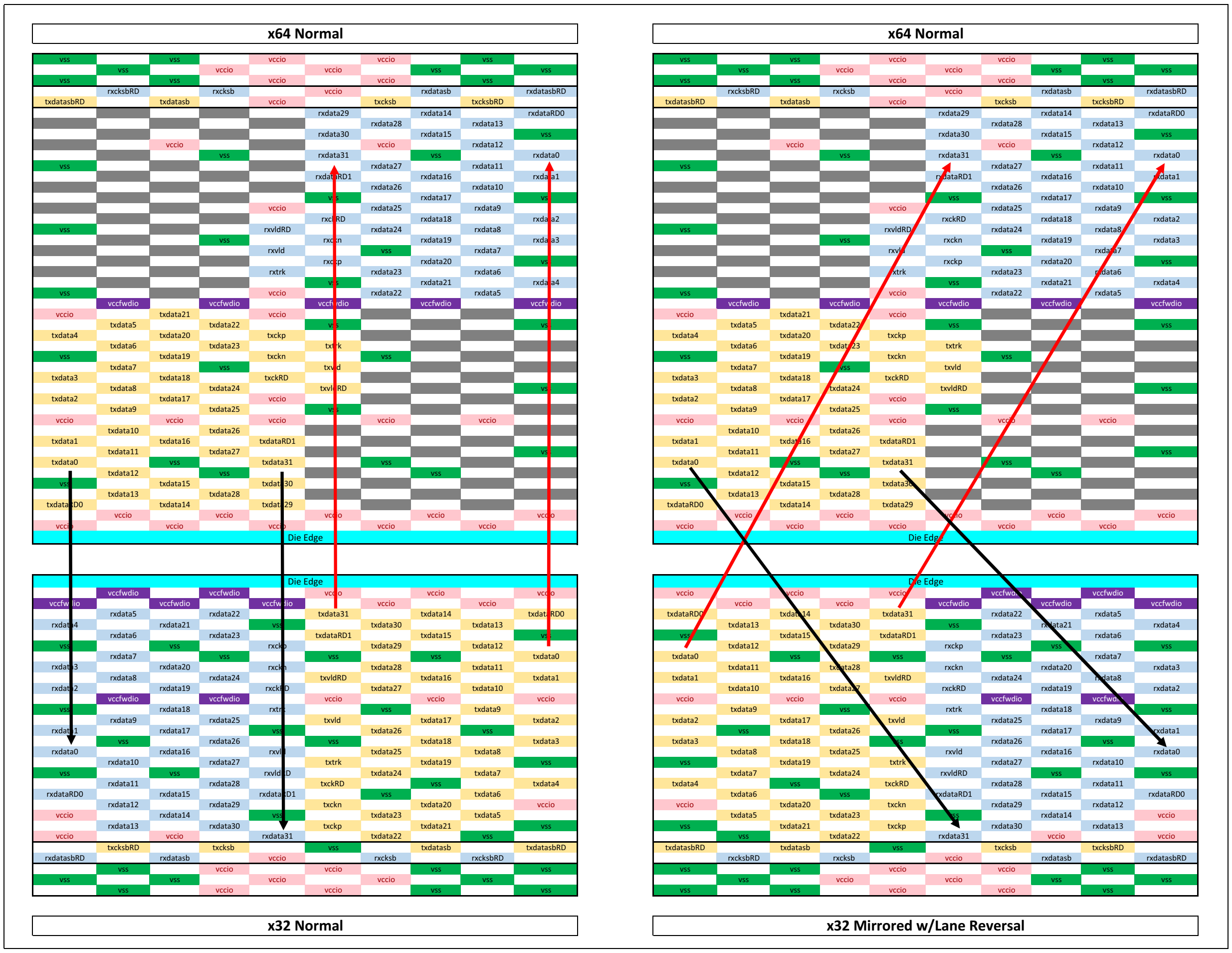
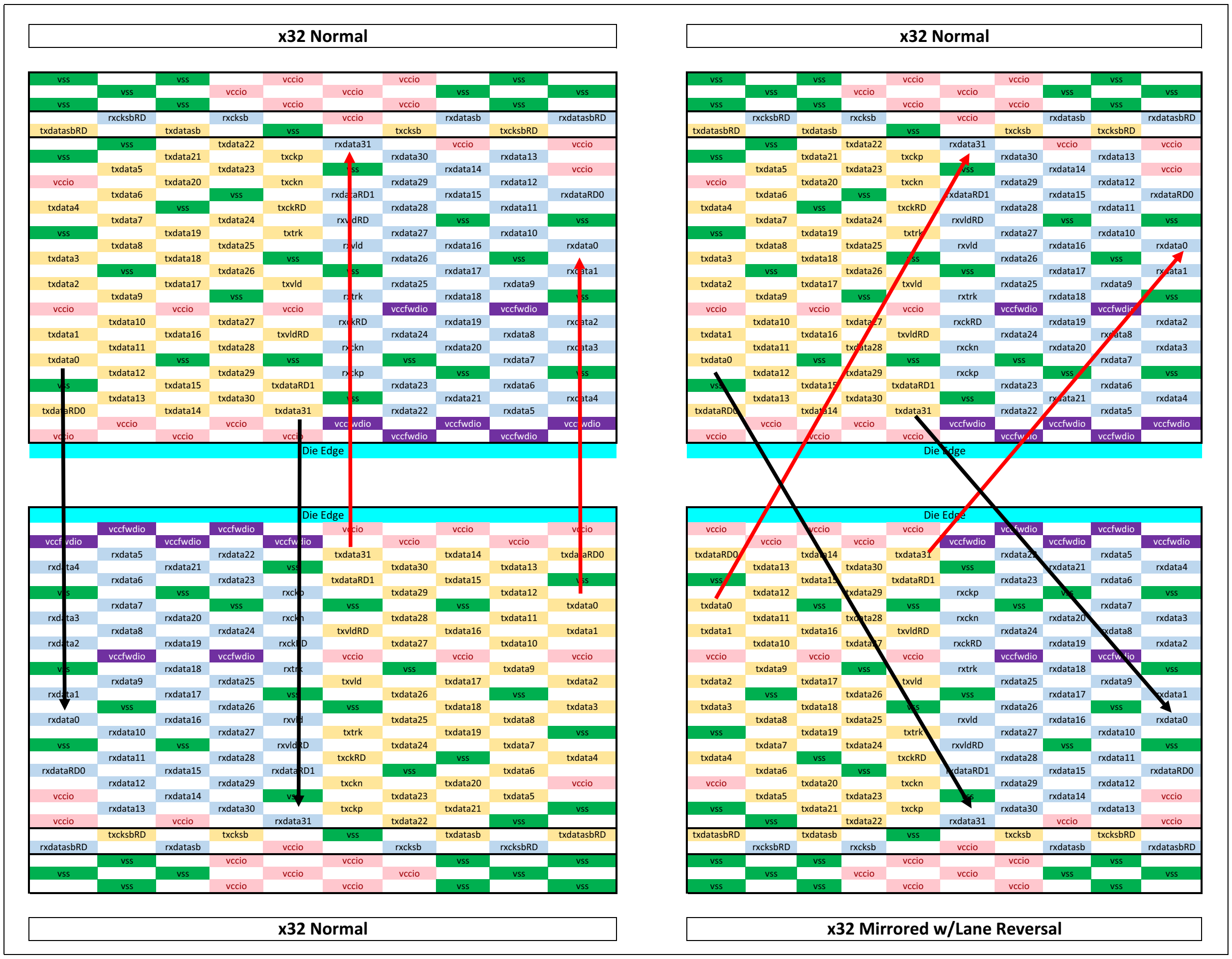


Figure 5-35. Example of Normal and Mirrored x32-to-x32 Advanced Package Module Connection



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 5-36 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 5-36 through Figure 5-39 indicate Module-to-Module connections.

Figure 5-36. Naming Convention for One-, Two-, and Four-module Advanced Package Paired with "Standard Die Rotate" Configurations

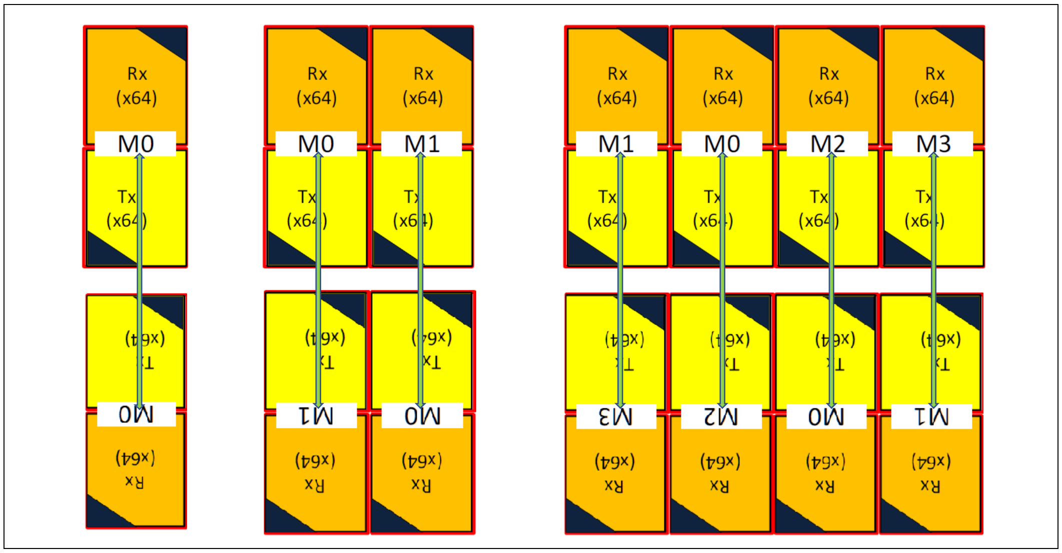


Figure 5-37 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their "Mirrored Die Rotate" counterparts with the same number of Advanced Package modules.

Figure 5-37. Naming Convention for One-, Two-, and Four-module Advanced Package Paired with "Mirrored Die Rotate" Configurations

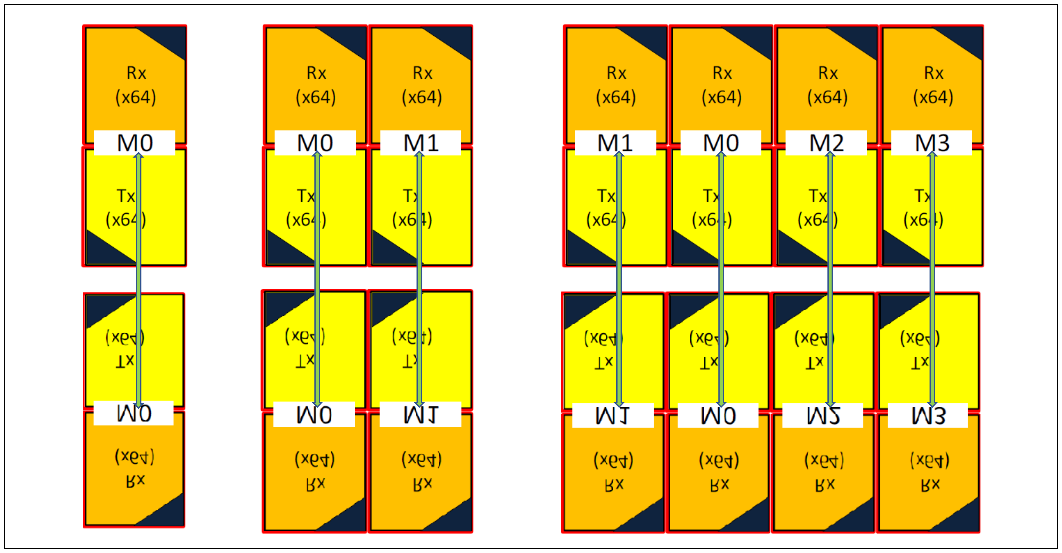


Table 5-16 summarizes the connections between the combinations shown in Figure 5-36 and Figure 5-37.

Table 5-16. 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 5-38 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 5-38. Examples for Advanced Package Configurations Paired with “Standard Die Rotate” Counterparts, with a Different Number of Modules

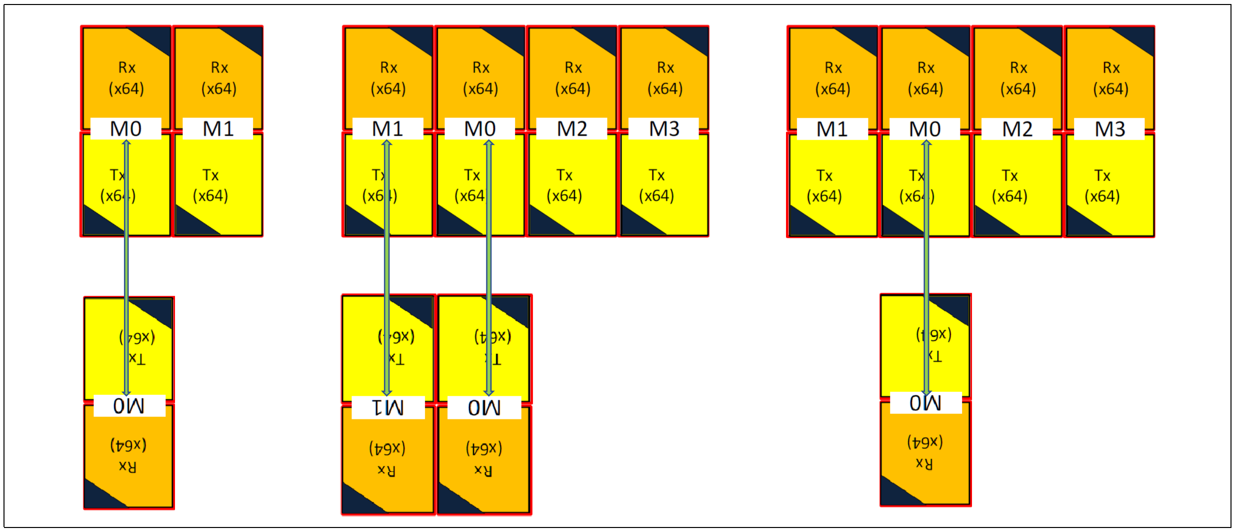


Figure 5-39 shows the naming convention for 1, 2, or 4 Advanced Package modules when they are connected to their “Mirrored Die Rotate” counterparts that have a different number of Advanced Package modules.

Figure 5-39. Examples for Advanced Package Configurations Paired with “Mirrored Die Rotate” Counterparts, with a Different Number of Modules

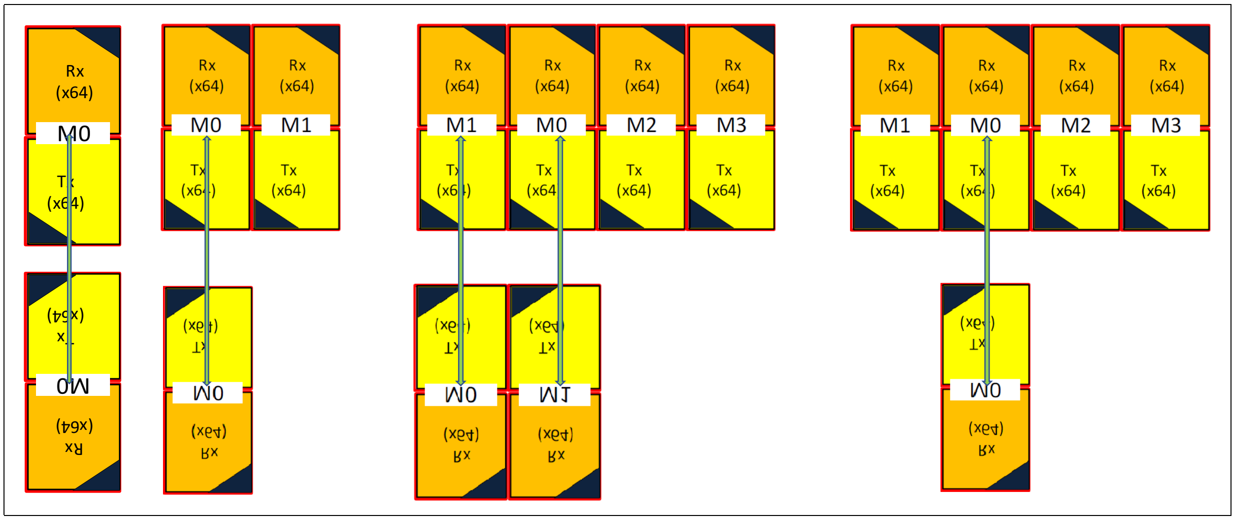


Table 5-17 summarizes the connections between the combinations shown in Figure 5-38 and Figure 5-39.

Table 5-17. 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	• M0 – M0 • M1 – NC	• M0 – M0 • M1 – NC
x4 – x2	• M0 – M0 • M1 – M1 • M3 – NC • M2 – NC	• M0 – M1 • M1 – M0 • M2 – NC • M3 – NC
x4 – x1	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	• 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 5-18.

Table 5-18. 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 5-19. Loss and crosstalk specifications between DC and Nyquist f_N follow the same methodology defined in Section 5.7.2.1.

Table 5-19. 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 5-20. 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 5-20. 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 5-40 and Figure 5-42 show the reference bump matrices for x16 (one module) and x32 (two module) Standard Packages, respectively.

It is strongly recommended to follow the bump matrices provided in Figure 5-40 for one module and Figure 5-42 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 5-41 and Figure 5-43, 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 5-40, 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 5-41. 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:

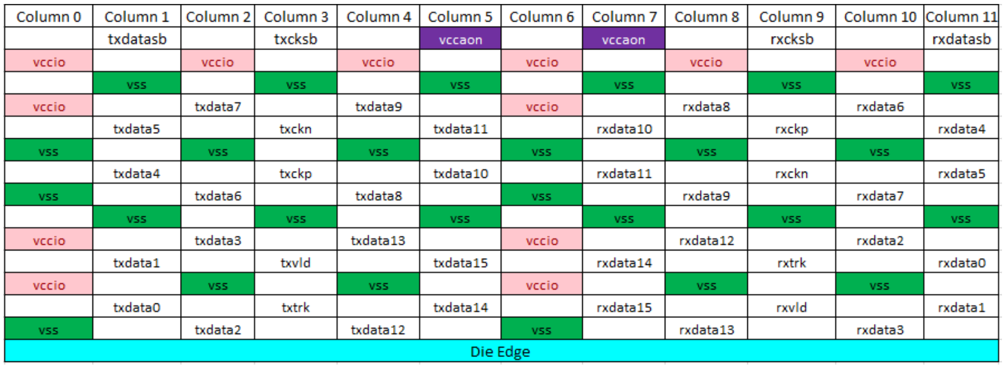
- When looking at a die such that the UCIe Modules are on the south side, Tx should always precede Rx within a module along the die's edge when going from left to right.
- When instantiating multiple modules, the modules must be stepped in the same orientation and abutted. Horizontal or vertical mirroring 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 UCIe signal routing. An example of stacked Standard Package Module instantiations is shown in Figure 5-42.

- If only one stacked module is instantiated, when looking at a die such that the UCIe 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.

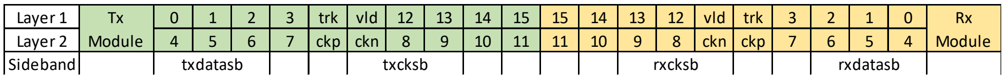
Note: An example of signal routing for stacked module is shown in Figure 5-44.

Figure 5-40. Standard Package Bump Map: x16 interface



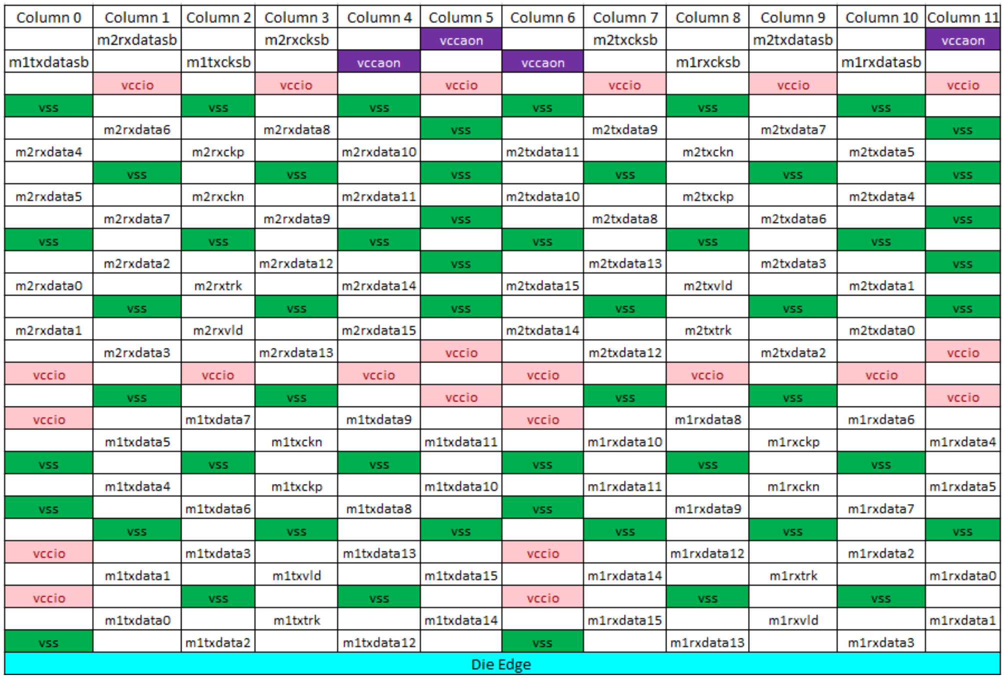
Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
	txdatasb		txcksb		vccaon		vccaon		rxcksb		rxdatasb
vccio		vccio		vccio		vccio		vccio		vccio	
	vss		vss		vss		vss		vss		vss
vccio		txdata7		txdata9		vccio		rxdata8		rxdata6	
	txdata5		txckn		txdata11		rxdata10		rxckp		rxdata4
vss		vss		vss		vss		vss		vss	
	txdata4		txckp		txdata10		rxdata11		rxckn		rxdata5
vss		txdata6		txdata8		vss		rxdata9		rxdata7	
	vss		vss		vss		vss		vss		vss
vccio		txdata3		txdata13		vccio		rxdata12		rxdata2	
	txdata1		txvld		txdata15		rxdata14		rxtrk		rxdata0
vccio		vss		vss		vccio		vss		vss	
	txdata0		txtrk		txdata14		rxdata15		rxvld		rxdata1
vss		txdata2		txdata12		vss		rxdata13		rxdata3	
Die Edge											

Figure 5-41. Standard Package x16 interface: Signal exit order



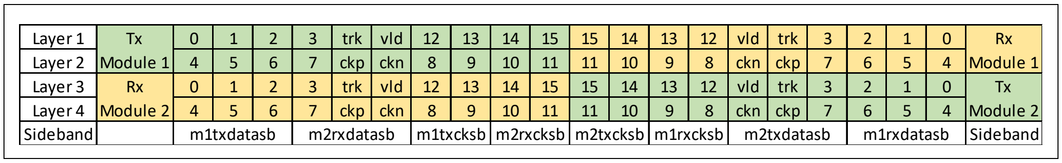
Layer 1	Tx	0	1	2	3	trk	vld	12	13	14	15	15	14	13	12	vld	trk	3	2	1	0	Rx
Layer 2	Module	4	5	6	7	ckp	ckn	8	9	10	11	11	10	9	8	ckn	ckp	7	6	5	4	Module
Sideband		txdatasb					txcksb							rxcksb					rxdatasb			

Figure 5-42. Standard Package Bump Map: x32 interface



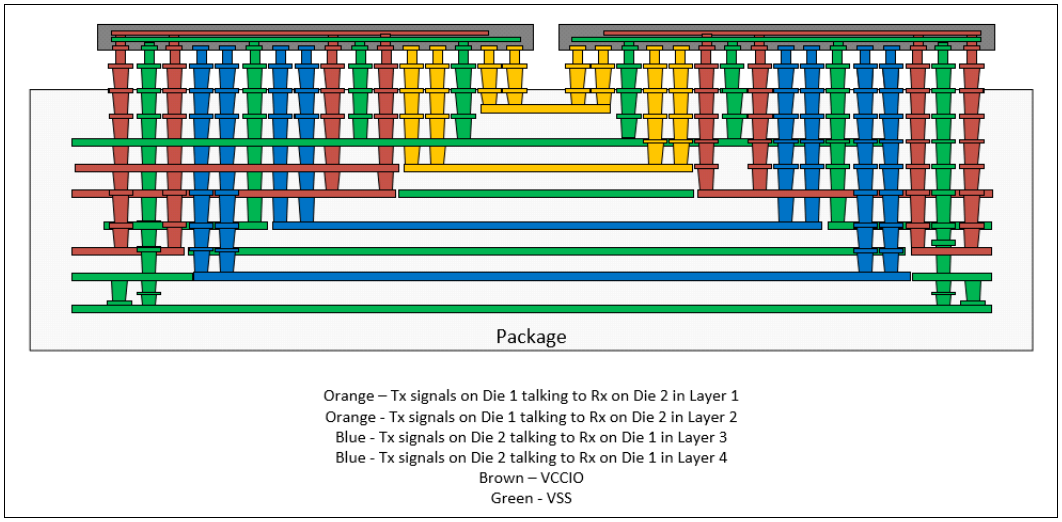
Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
	m2rxdatasb		m2rxcksb		vccaon		m2txcksb		m2txdatasb		vccaon
m1txdatasb		m1txcksb		vccaon		vccaon		m1rxcksb		m1rxdatasb	
	vccio		vccio		vccio		vccio		vccio		vccio
vss		vss		vss		vss		vss		vss	
	m2rxdata6		m2rxdata8		vss		m2txdata9		m2txdata7		vss
m2rxdata4		m2rxckp		m2rxdata10		m2txdata11		m2txckn		m2txdata5	
	vss		vss		vss		vss		vss		vss
m2rxdata5		m2rxckn		m2rxdata11		m2txdata10		m2txckp		m2txdata4	
	m2rxdata7		m2rxdata9		vss		m2txdata8		m2txdata6		vss
vss		vss		vss		vss		vss		vss	
	m2rxdata2		m2rxdata12		vss		m2txdata13		m2txdata3		vss
m2rxdata0		m2rxtrk		m2rxdata14		m2txdata15		m2txvld		m2txdata1	
	vss		vss		vss		vss		vss		vss
m2rxdata1		m2rxvld		m2rxdata15		m2txdata14		m2txtrk		m2txdata0	
	m2rxdata3		m2rxdata13		vccio		m2txdata12		m2txdata2		vccio
vccio		vccio		vccio		vccio		vccio		vccio	
	vss		vss		vccio		vss		vss		vccio
vccio		m1txdata7		m1txdata9		vccio		m1rxdata8		m1rxdata6	
	m1txdata5		m1txckn		m1txdata11		m1rxdata10		m1rxckp		m1rxdata4
vss		vss		vss		vss		vss		vss	
	m1txdata4		m1txckp		m1txdata10		m1rxdata11		m1rxckn		m1rxdata5
vss		m1txdata6		m1txdata8		vss		m1rxdata9		m1rxdata7	
	vss		vss		vss		vss		vss		vss
vccio		m1txdata3		m1txdata13		vccio		m1rxdata12		m1rxdata2	
	m1txdata1		m1txvld		m1txdata15		m1rxdata14		m1rxtrk		m1rxdata0
vccio		vss		vss		vccio		vss		vss	
	m1txdata0		m1txtrk		m1txdata14		m1rxdata15		m1rxvld		m1rxdata1
vss		m1txdata2		m1txdata12		vss		m1rxdata13		m1rxdata3	
Die Edge											

Figure 5-43. 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 5-44. Standard Package cross section for stacked module

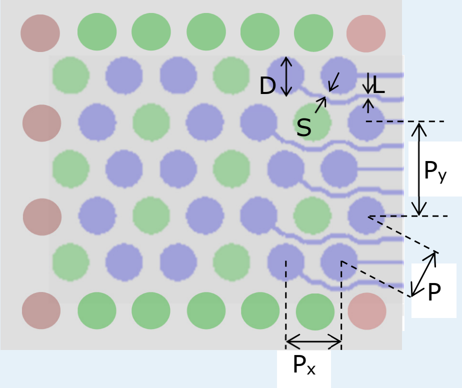


Orange – Tx signals on Die 1 talking to Rx on Die 2 in Layer 1
Orange - Tx signals on Die 1 talking to Rx on Die 2 in Layer 2
Blue - Tx signals on Die 2 talking to Rx on Die 1 in Layer 3
Blue - Tx signals on Die 2 talking to Rx on Die 1 in Layer 4
Brown – VCCIO
Green - VSS

IMPLEMENTATION NOTE

Figure 5-45 shows a breakout design reference with the Standard Package channel based on the bump pitch and on routing design rules.

Figure 5-45. Standard Package reference configuration



$$P = D + L + 2S$$

$$P = \frac{1}{2}\sqrt{P_x^2 + P_y^2}$$

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

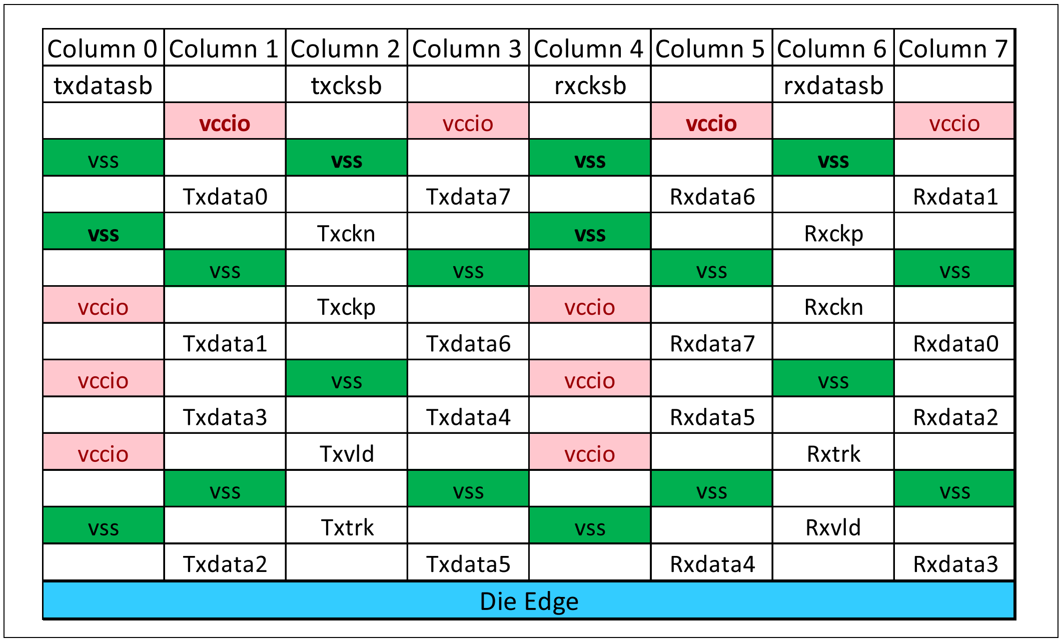
5.7.3.2 x8 Standard Package Module Bump Map

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

UCIe-S x8 support is limited to a single module configuration. When a UCIe-S x8 port is connected to a multi-module x16 port, it is always connected to Module 0 UCIe-S x16.

Figure 5-46 shows the reference bump matrix for a x8 Standard Package.

Figure 5-46. 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 5-46. 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 5-47 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 5-47 through Figure 5-51 indicate Module-to-Module connections.

Figure 5-47. Naming Convention for One-, Two-, and Four-module Standard Package Paired with “Standard Die Rotate” Configurations

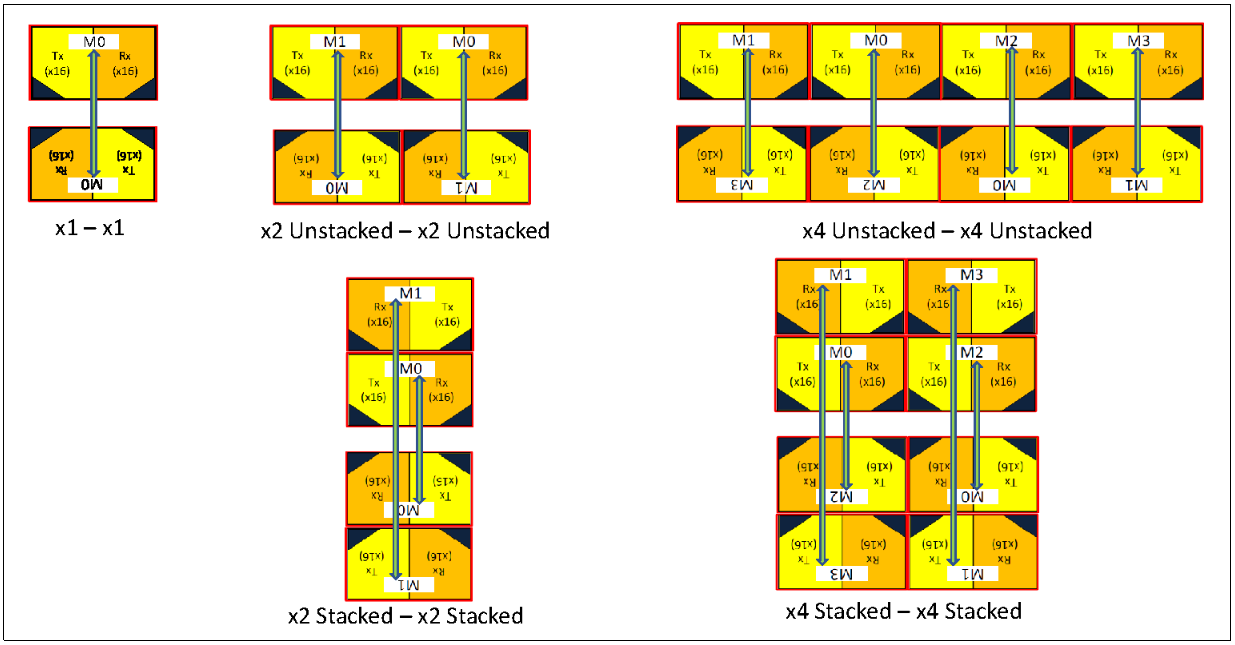


Figure 5-48 shows the naming convention for 1, 2, or 4 Standard Package modules when they are connected to their "Mirrored Die Rotate" counterparts that have same number of Standard Package modules, with either same stack or same unstacked configuration.

Figure 5-48. Naming Convention for One-, Two-, and Four-module Standard Package Paired with "Mirrored Die Rotate" Configurations

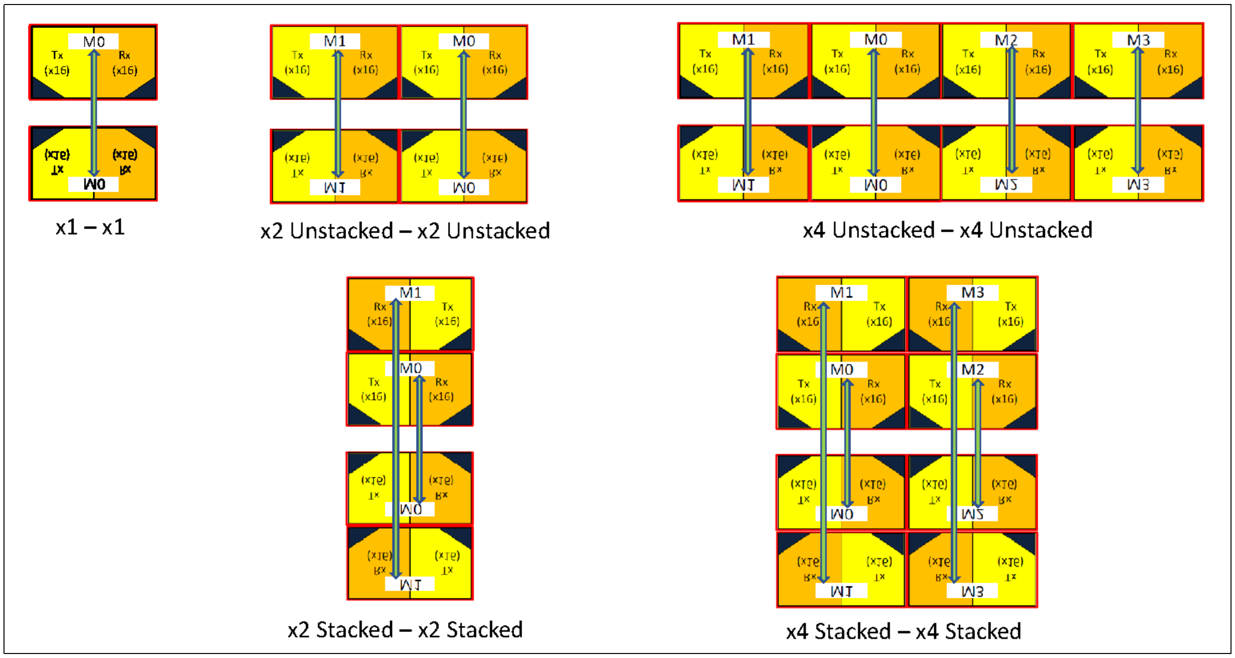


Table 5-21 summarizes the connections between the combinations shown in Figure 5-47 and Figure 5-48.

Table 5-21. 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 5-48)	Option 2[c] (See Figure 5-51)
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 5-21. 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 5-50 for Option 2 connection illustrations.

Figure 5-49 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 5-49. Examples for Standard Package Configurations Paired with “Standard Die Rotate” Counterparts, with a Different Number of Modules

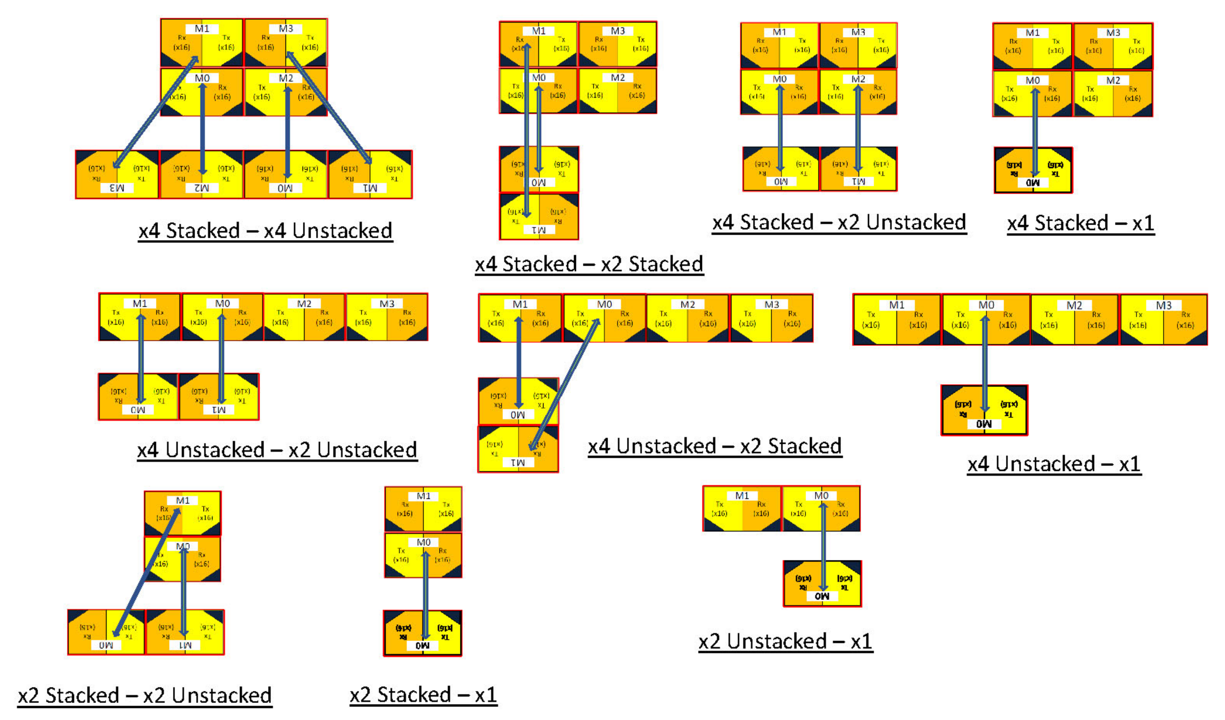

M0
M1
M2
M3
Rx (x16)
Tx (x16)
x4 Stacked – x4 Unstacked
x4 Stacked – x2 Stacked
x4 Stacked – x2 Unstacked
x4 Stacked – x1
x4 Unstacked – x2 Unstacked
x4 Unstacked – x2 Stacked
x4 Unstacked – x1
x2 Stacked – x2 Unstacked
x2 Stacked – x1
x2 Unstacked – x1

Figure 5-50 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 5-50. Examples for Standard Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules

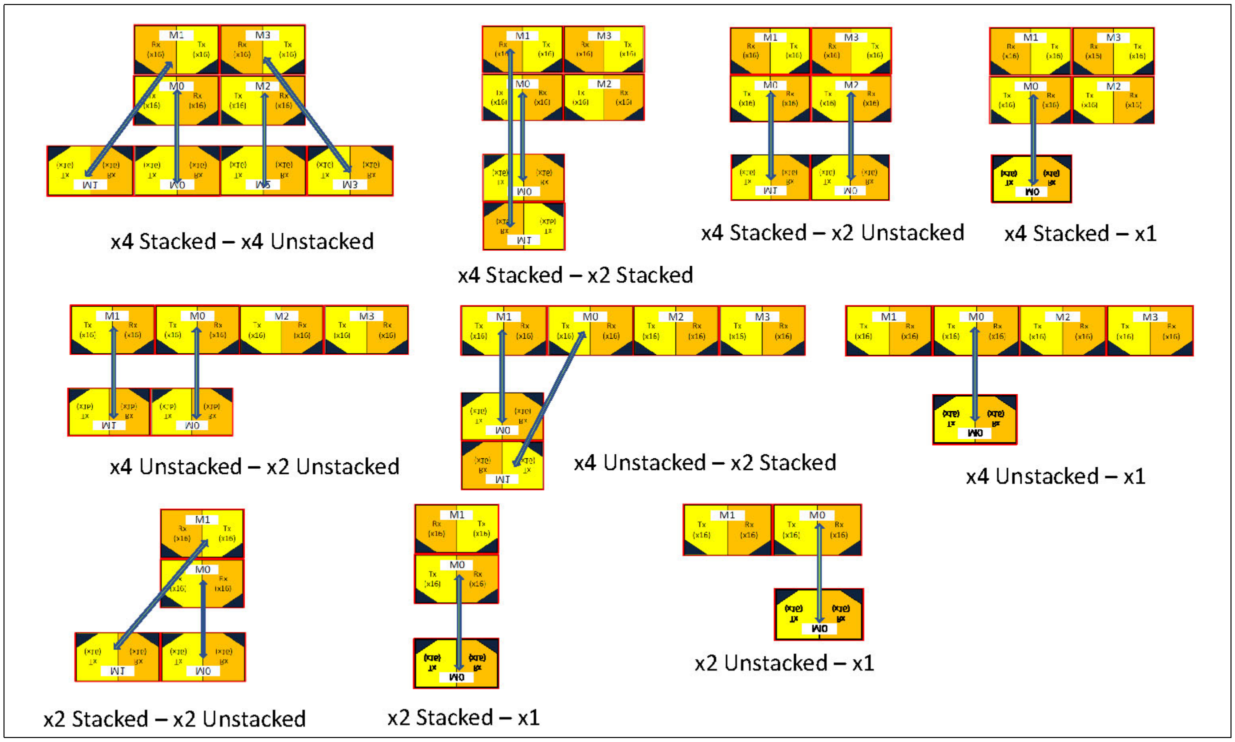

M0
M1
M2
M3
Rx (x16)
Tx (x16)
x4 Stacked – x4 Unstacked
x4 Stacked – x2 Stacked
x4 Stacked – x2 Unstacked
x4 Stacked – x1
x4 Unstacked – x2 Unstacked
x4 Unstacked – x2 Stacked
x4 Unstacked – x1
x2 Stacked – x2 Unstacked
x2 Stacked – x1
x2 Unstacked – x1

Figure 5-51 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 5-21 and Table 5-22.

Figure 5-51. Additional Examples for Standard Package Configurations Paired with "Mirrored Die Rotate" Counterparts, with a Different Number of Modules

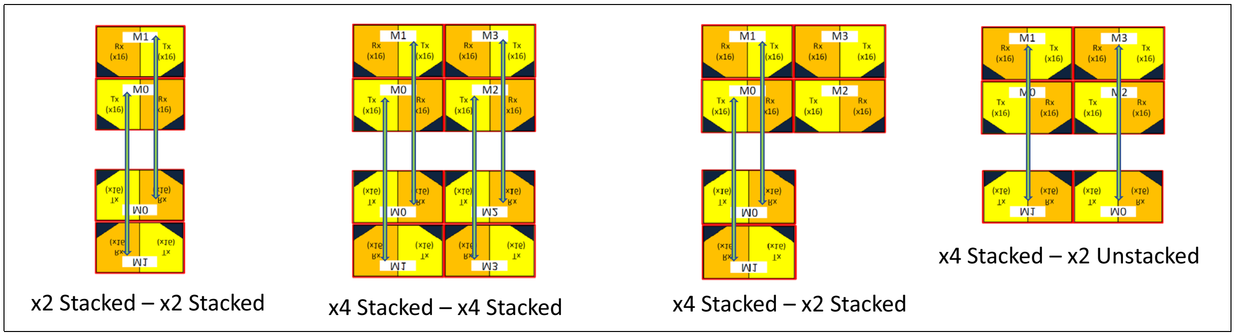


Table 5-22 summarizes the connections between the combinations shown in Figure 5-49, Figure 5-50, and Figure 5-51.

5.7.3.4.1 Module Degrade Rules

Table 5-22. 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 5-50)	Option 2 (See Figure 5-51)
x4 Stacked – x4 Unstacked	• M0 – M2 • M1 – M3 • M3 – M1 • M2 – M0	• M0 – M0 • M1 – M1 • M2 – M2 • M3 – M3	
x4 Stacked – x2 Stacked	• M0 – M0 • M1 – M1 • M3 – NC • M2 – NC	• M0 – M0 • M1 – M1 • M2 – NC • M3 – NC	• M0 - M1 • M1 - M0 • M2 – NC • M3 – NC
x4 Stacked – x2 Unstacked	• M0 – M0 • M1 – NC • M3 – NC • M2 – M1	• M0 – M1 • M1 – NC • M2 – M0 • M3 - NC	• M0 – NC • M1 – M1 • M2 – NC • M3 – M0
x4 Stacked – x1	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	
x4 Unstacked – x2 Unstacked	• M0 – M1 • M1 – M0 • M3 – NC • M2 – NC	• M0 – M0 • M1 – M1 • M2 – NC • M3 – NC	
x4 Unstacked – x2 Stacked	• M0 – M1 • M1 – M0 • M3 – NC • M2 – NC	• M0 – M1 • M1 – M0 • M2 – NC • M3 – NC	
x4 Unstacked – x1	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	• M0 – M0 • M1 – NC • M3 – NC • M2 – NC	
x2 Stacked – x2 Unstacked	• M0 – M1 • M1 – M0	• M0 – M0 • M1 – M1	
x2 Stacked – x1	• M0 – M0 • M1 – NC	• M0 – M0 • M1 – NC	
x2 Unstacked – x1	• M0 – M0 • M1 – NC	• 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 5-52 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 5-23.

Note: The double-ended arrows in Figure 5-52 indicate Module-to-Module connections.

Figure 5-52. Example of a Configuration for Standard Package, with Some Modules Disabled

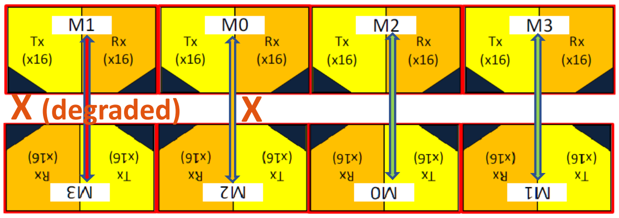


Table 5-23 summarizes the resulting degraded link if there are one, two, or three failed module-pairs for the x4 Unstacked to x4 Unstacked configuration.

Table 5-23. 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			
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 5-53 shows the bump map for a UCIe-S sideband-only port. Figure 5-54 shows the supported configurations for a UCIe-S sideband-only port.

Figure 5-53. UCIe-S Sideband-only Port Bump Map

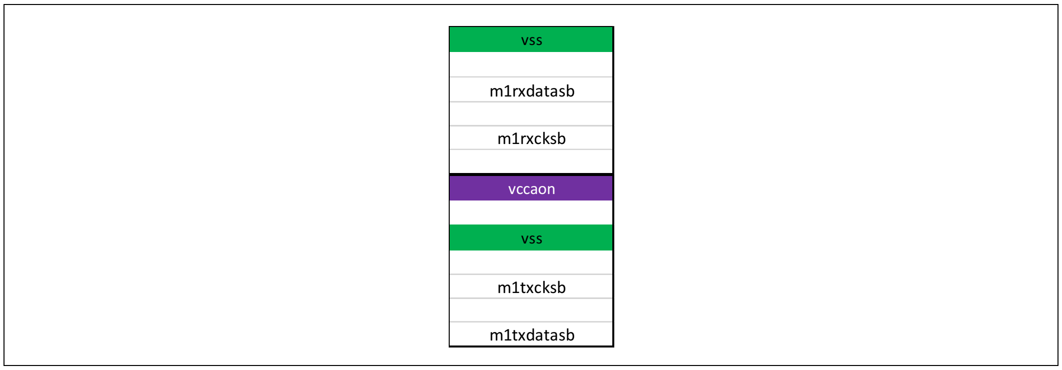
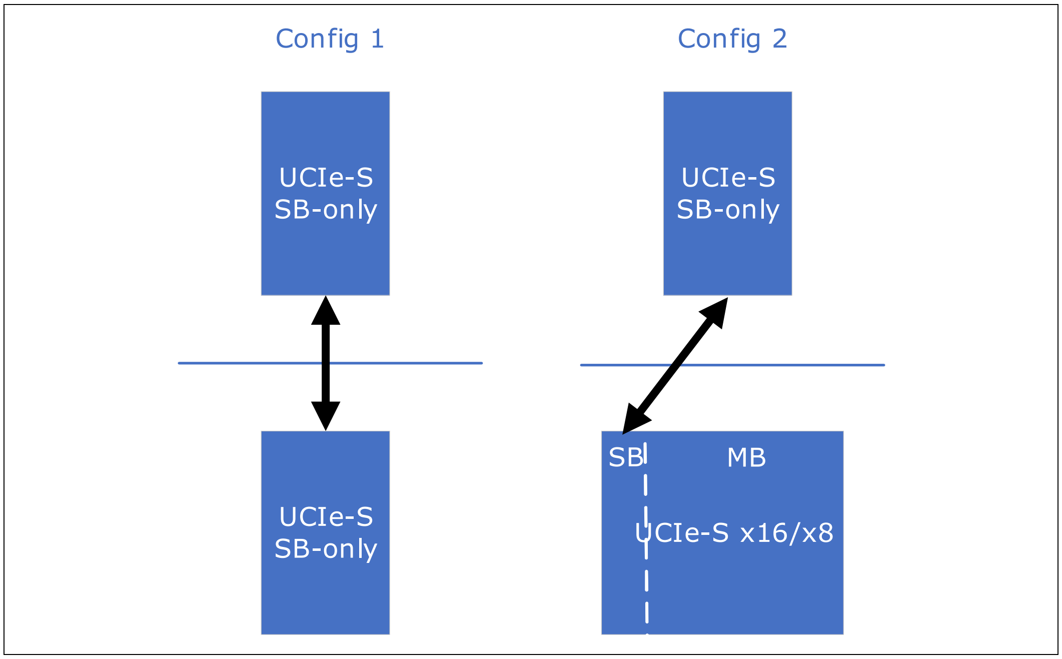


Figure 5-54. UCIe-S Sideband-only Port Supported Configurations



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 5-24

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 5-24. 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 5-24. Table 5-25 shows the specification at Nyquist frequency.

Table 5-25. 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 5-55 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 5-56 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 5-55 and Figure 5-56.

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

Figure 5-55. Data Lane repair resources

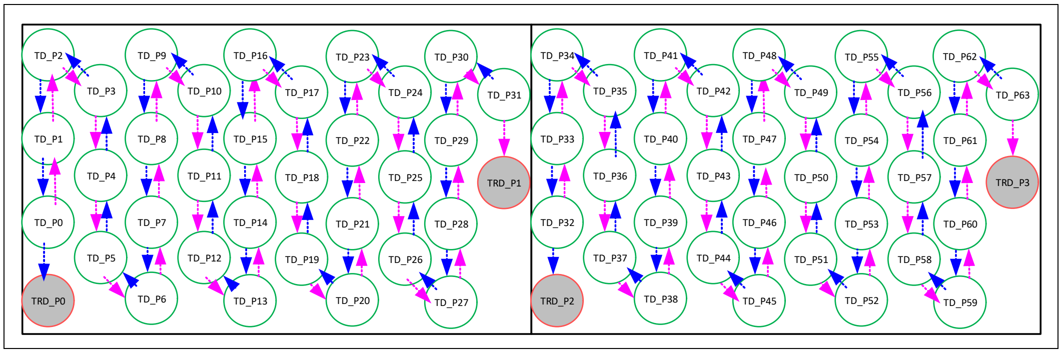
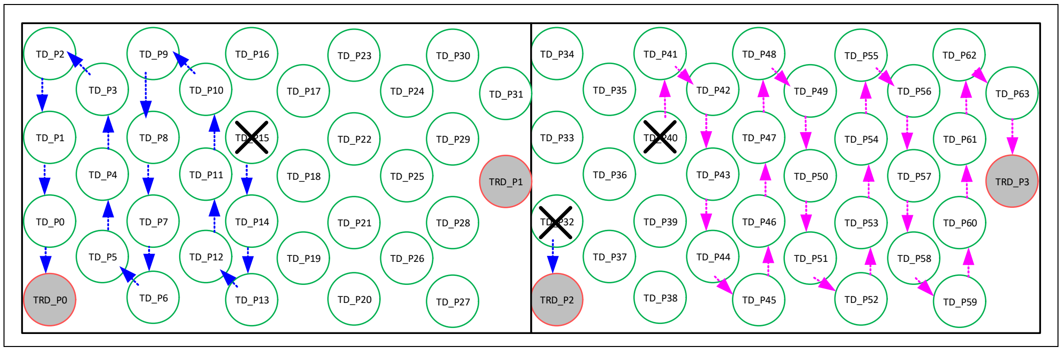


Figure 5-56. Data Lane repair



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 5-26. Error detection and correction mechanisms such as CRC and retry are required for BER for 1E-15 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.

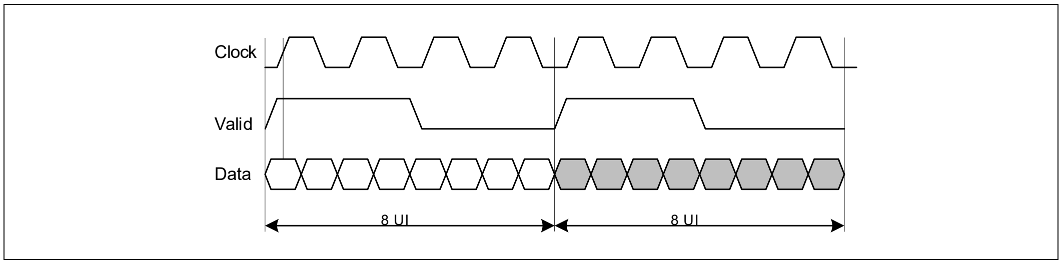
Table 5-26. Raw BER requirements

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

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 5-57 shows the transfer of two 8 UI data packets back to back.

Figure 5-57. 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 (all Advanced Package and unterminated Standard Package Links), 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 5-58 and Figure 5-59.
    - — In Continuous mode, the clock remains free running (examples shown in Figure 5-60). Total idle period must be an integer multiple of 8 UI.

Figure 5-58. Data, Clock, Valid Levels for Half-rate Clocking: Clock-gated Unterminated Link

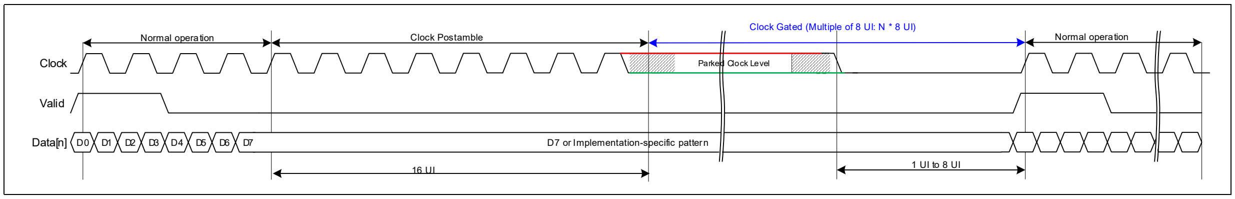


Figure 5-59. Data, Clock, Valid Levels for Quarter-rate Clocking: Clock-gated Unterminated Link

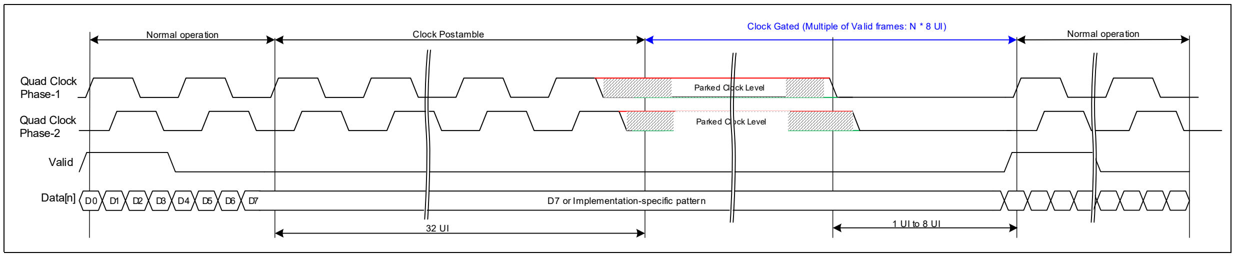
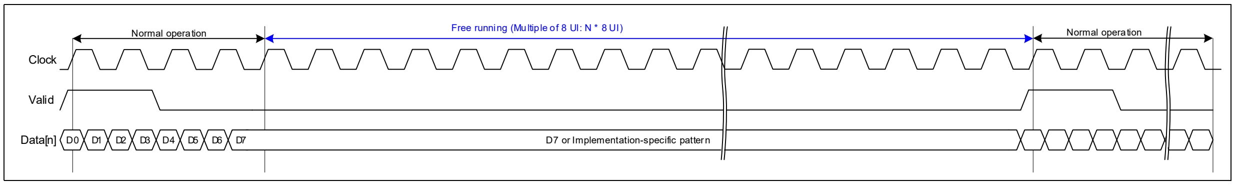


Figure 5-60. Data, Clock, Valid Levels for Half-rate Clocking: Continuous Clock Unterminated Link


Normal operation
Free running (Multiple of 8 UI: N * 8 UI)
Normal operation
Clock
Valid
Data[n]
D0
D1
D2
D3
D4
D5
D6
D7
D7 or Implementation-specific pattern

- If the Link is terminated (Standard Package terminated Links), 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. 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 5-61 and Figure 5-63.
  - In Continuous mode, the clock remains free running (examples shown in Figure 5-64). 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 5-61. Data, Clock, Valid Gated Levels for Half-rate Clocking: Terminated Link

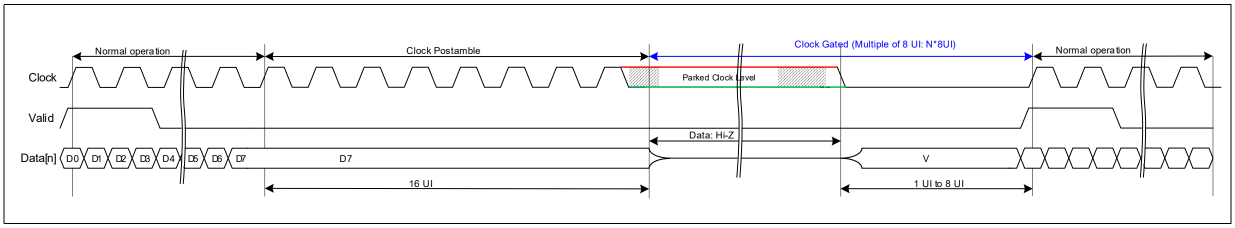


Figure 5-62. Data, Clock, Valid Gated Levels for Quarter-rate Clocking: Terminated Link

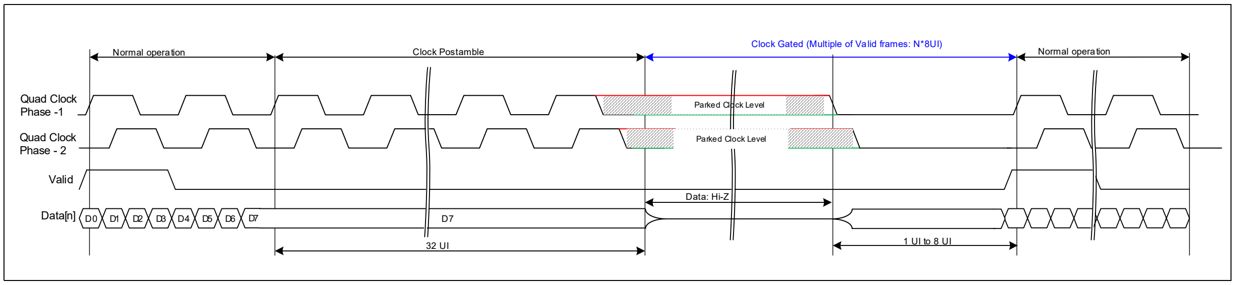
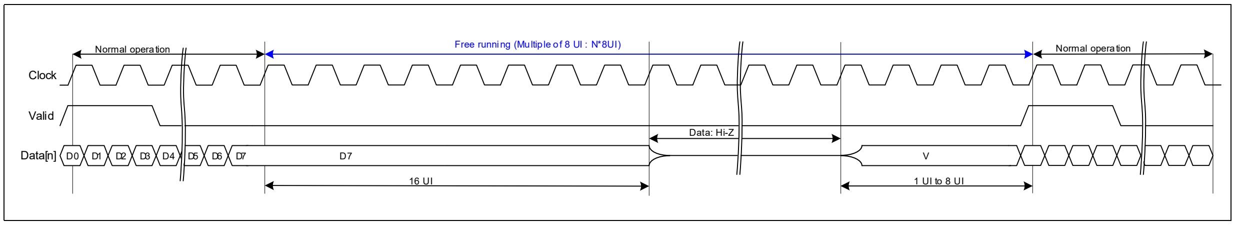


Figure 5-63. Data, Clock, Valid Gated Levels for Half-rate Clocking: Continuous Clock Terminated Link



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.

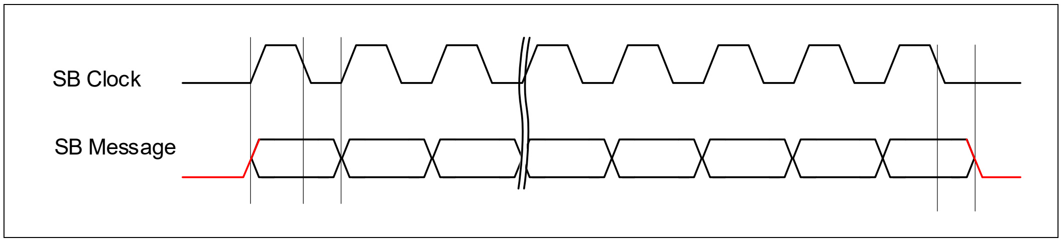
5.13 Sideband signaling

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. Sideband must 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 5-64. 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 5-64. Sideband signaling



5.13.1 Sideband Electrical Parameters

Table 5-27 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 5-27. 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*VCCCAON			V
Output low voltage (V_{OL})			0.1*VCCAON	V
Sideband Data Setup Time	200	-	-	ps
Sideband Data Hold Time	200	-	-	ps
Rise/Fall time for Advanced Package[b]	50	-	280	ps
Rise/Fall time for Standard Package[c]	80	-	175	ps

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

[b] 20 to 80% of VCCAON level with Advanced Package reference channel load.

[c] 20 to 80% of VCCAON level with Standard Package reference channel load.

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 5-28 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 reduced tolerances compared to REFCLK.

Table 5-28. AUXCLK Frequency Parameters

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

§ §

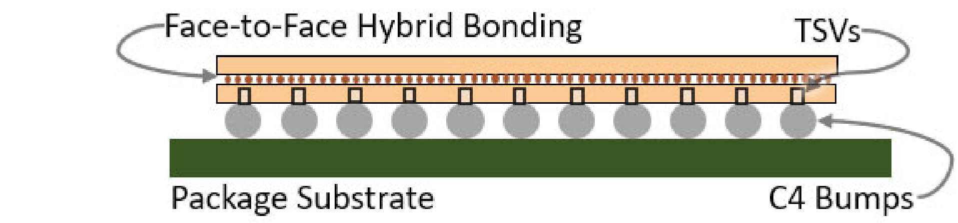
6.0 UCIe-3D

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



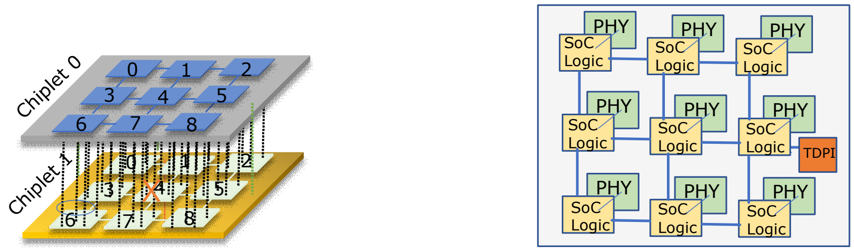
6.2 UCIe-3D Features and Summary

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-um bump-pitch, the UCIe-3D area amortized on a per-lane basis must be less than 1 um^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



(a)

Chiplets connected across each SoC with UCIe-3D. A failure in UCIe-3D in either die results in the remaining SoCs being routing around the failure.

(b)

Each chiplet has its own system controller logic, I/Os, etc. Each SoC Logic connects to one or more UCIe-3D PHY. The common test, debug, pattern, and infrastructure (TDPI) block orchestrates training, testing, debug, etc. across chiplets.

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

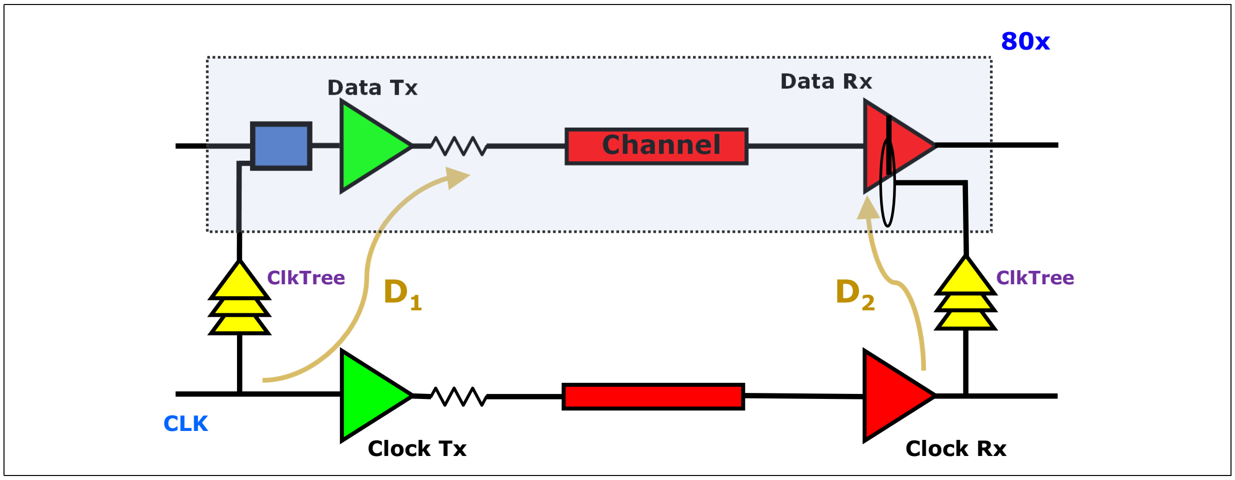
Table 6-1. UCIe-3D Key Performance Indicators

Characteristics/KPIs	UCIe-S	UCIe-A	UCIe-3D	Comments for UCIe-3D
Characteristics				
Data Rate (GT/s)	4, 8, 12, 16, 24, 32		up to 4	• Equal to SoC Logic Frequency — power efficiency is critical.
Width (each cluster)	16	64	80	• Options of reduced width to 70, 60, ….
Bump Pitch (um)	100 to 130	25 to 55	≤ 10 (optimized) > 10 to 25 (functional)	• Must scale such that UCIe-3D fits with the bump area. • Must support hybrid bonding.
Channel Reach (mm)	≤ 25	≤ 2	3D vertical	• F2F bonding initially; F2B, B2B, multi-stack possible.
Target for Key Metrics				
BW Die Edge (GB/s/mm)	28 to 224	165 to 1,317	N/A (vertical)	
BW Density (GB/s/mm^2)	22 to 125	188 to 1,350	4,000 at 9 um	• 4 TB/s/mm^2 at 9 um • Approximately 12 TB/s/mm^2 at 5 um • Approximately 35 TB/s/mm^2 at 3 um • Approximately 300 TB/s/mm^2 at 1 um
Power Efficiency Target (pJ/b)	0.50	0.25	< 0.05 at 9 um	• 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	• No preamble or postamble.
Latency (Tx + Rx)	< 2 ns (PHY + Adapter)		0.125 ns at 4 GT/s	• 0.5 UI, half of flop to flop.
Reliability (FIT)	0 < FIT << 1			• BER < 1E-27.
ESD	30-V CDM		5-V CDM –> ≤ 3V	• 5-V CDM at introduction. No ESD for wafer-to-wafer hybrid bonding possible.

6.3 UCIe-3D Tx, Rx, and Clocking

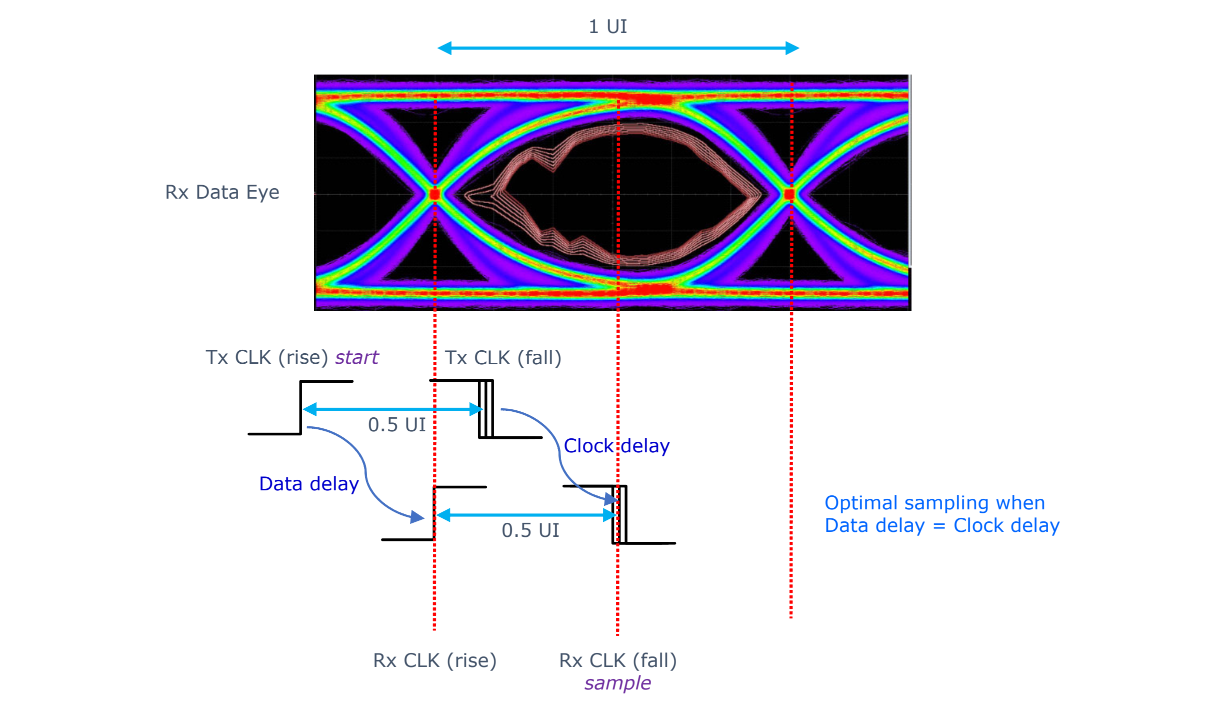
Figure 6-3 presents the Transceiver (Trx) architecture of UCIe-3D. A matched architecture as shown in Figure 5-4, Figure 5-8, and Figure 5-13 offers optimal supply noise rejection. However, this comes at the cost of increased power consumption. The architecture depicted in Figure 6-3 circumvents this 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 UCIe-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_{trx}			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:

$$\alpha_{Tx} = \frac{dD_{tx}}{D_{tx}} \Big/ \frac{dV_{cc}}{V_{cc}} \quad \alpha_{Rx} = \frac{dD_{rx}}{D_{rx}} \Big/ \frac{dV_{cc}}{V_{cc}}$$

d. This is equivalent to a variation of ±5% in Vcc. 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 Vcc-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.

$$D_{tx\_min} = \max\left(D_{tx\_typ} - 0.08\ \text{UI}, 0\right)$$

Equation 6-3.

$$D_{rx\_min} = \max\left(D_{rx\_typ} - 0.08\ \text{UI}, 0\right)$$

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

Equation 6-4.

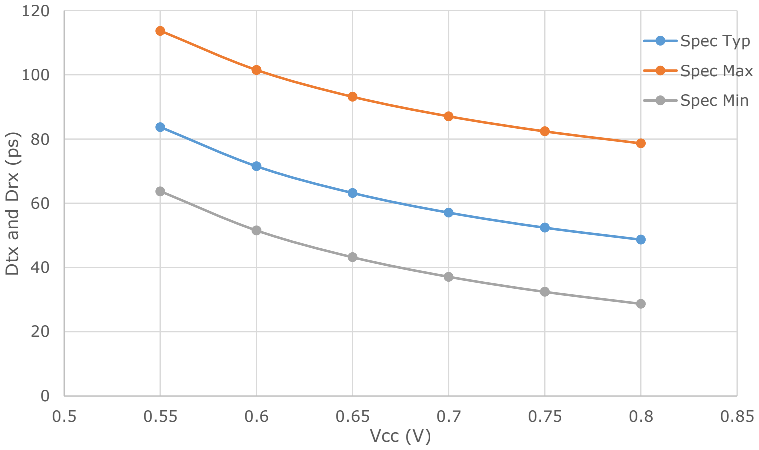
$$D_{tx\_max} = D_{tx\_typ} + 0.12\ \text{UI}$$

Equation 6-5.

$$D_{rx\_max} = D_{rx\_typ} + 0.12\ \text{UI}$$

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.

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

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.

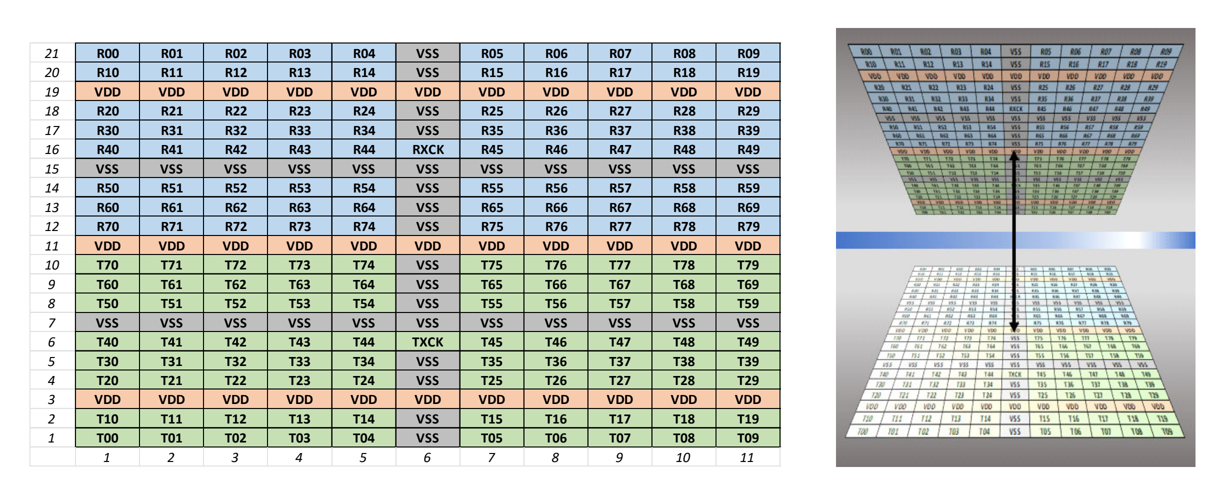
Table 6-4. Energy Efficiency Target

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 mm^2. It is important to note that the area scales with the square of the bump pitch.

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



21	R00	R01	R02	R03	R04	VSS	R05	R06	R07	R08	R09
20	R10	R11	R12	R13	R14	VSS	R15	R16	R17	R18	R19
19	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
18	R20	R21	R22	R23	R24	VSS	R25	R26	R27	R28	R29
17	R30	R31	R32	R33	R34	VSS	R35	R36	R37	R38	R39
16	R40	R41	R42	R43	R44	RXCK	R45	R46	R47	R48	R49
15	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
14	R50	R51	R52	R53	R54	VSS	R55	R56	R57	R58	R59
13	R60	R61	R62	R63	R64	VSS	R65	R66	R67	R68	R69
12	R70	R71	R72	R73	R74	VSS	R75	R76	R77	R78	R79
11	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
10	T70	T71	T72	T73	T74	VSS	T75	T76	T77	T78	T79
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

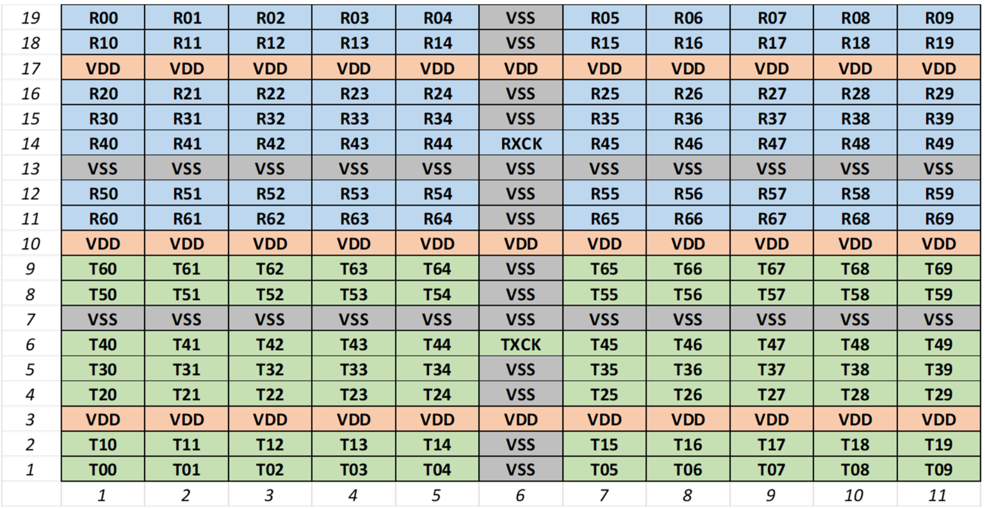
The UCIe-3D standard does not prescribe a mandatory bump pitch; however, a 9-um pitch is recommended at introduction. As the technology advances, additional specific recommended pitch values will be established.

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



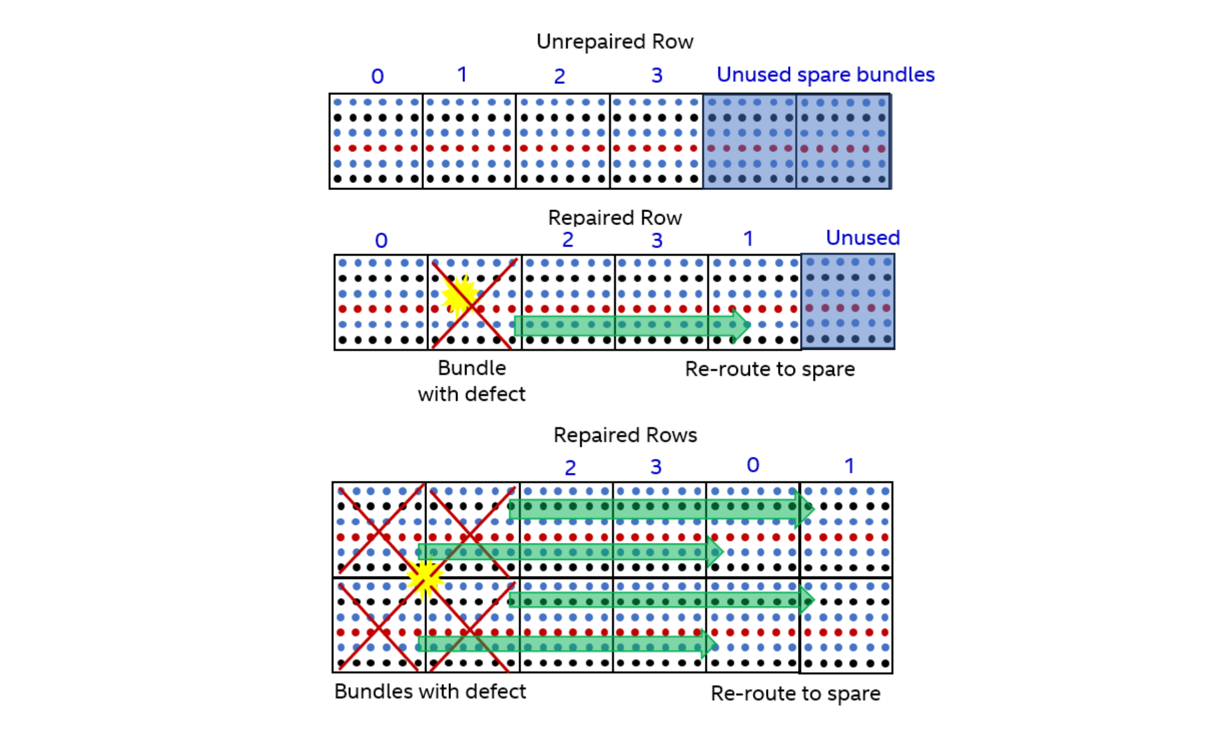
19	R00	R01	R02	R03	R04	VSS	R05	R06	R07	R08	R09
18	R10	R11	R12	R13	R14	VSS	R15	R16	R17	R18	R19
17	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
16	R20	R21	R22	R23	R24	VSS	R25	R26	R27	R28	R29
15	R30	R31	R32	R33	R34	VSS	R35	R36	R37	R38	R39
14	R40	R41	R42	R43	R44	RXCK	R45	R46	R47	R48	R49
13	VSS	VSS	VSS	VSS	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, UCIe Link specific rules are pointed out using “UCIe Link sideband:”. If no prefix is mentioned, it is a common rule across FDI, RDI and UCIe Link.

The Physical Layer is responsible for framing and transporting sideband packets over the UCIe 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. UCIe 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 Packet Types

Three different categories of 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).

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

Table 7-1 gives the mapping of opcode encodings to Packet Types.

Table 7-1. Opcode Encodings Mapped to Packet Types

Opcode Encoding	Packet Type
00000b	32b Memory Read
00001b	32b Memory Write
00010b	32b DMS Register Read
00011b	32b DMS Register Write
00100b	32b Configuration Read
00101b	32b Configuration Write
01000b	64b Memory Read
01001b	64b Memory Write
01010b	64b DMS Register Read
01011b	64b DMS Register Write
01100b	64b Configuration Read
01101b	64b Configuration Write
10000b	Completion without Data
10001b	Completion with 32b Data
10010b	Message without Data
10111b	Management Port Messages without Data
11000b	Management Port Message with Data
11001b	Completion with 64b Data
11011b	Message with 64b Data
Other encodings	Reserved

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 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 UCIe 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 4-8), 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 4-8), 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.

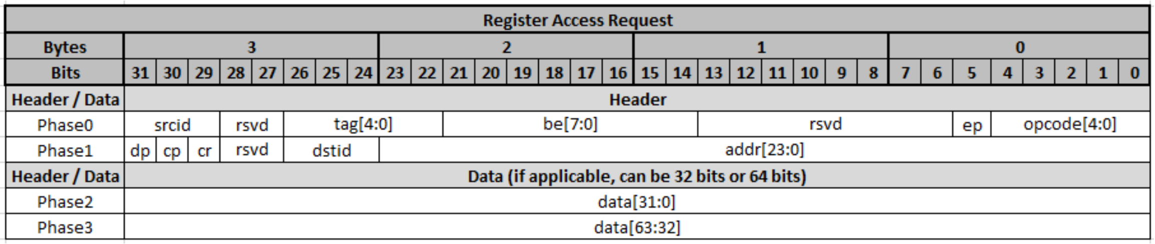
Table 7-5. Field descriptions for Register Access Requests

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	{RL[3:0], Offset[19:0]} Offset is the Byte Offset. RL[3:0] encodings are as follows: 0h: Register Locator 0 1h: Register Locator 1 2h: Register Locator 2 3h: Register Locator 3 Fh: Accesses for Protocol specific MMIO registers that are shadowed in the Adapter (e.g., ARB/MUX registers defined in the *CXL Specification*). The offsets for these registers are implementation specific, and the protocol layer must translate accesses to match the offsets implemented in the Adapter. Other encodings are reserved. For accesses to Reserved RL encodings, the completer must respond with a UR.
Configuration Reads/Writes	{RL[3:0], Rsvd[7:0], Byte Offset[11:0]}, where RL[3:0] encodings are as follows: 0h: UCIe Link DVSEC Fh: Accesses for Protocol specific configuration registers that are shadowed in the Adapter (e.g., ARB/MUX registers defined in the *CXL Specification*). The offsets for these registers are implementation specific, and the protocol layer must translate accesses to match the offsets implemented in the Adapter. Other encodings are reserved. For accesses to Reserved RL encodings, the completer must respond with a UR.
DMS Register Reads/Writes	These allow for accessing the DMS registers implemented in UCIe Spoke Type 0, 1, or 2. 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 UCIe 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). Addr[23:22] encodings are as follows: 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.

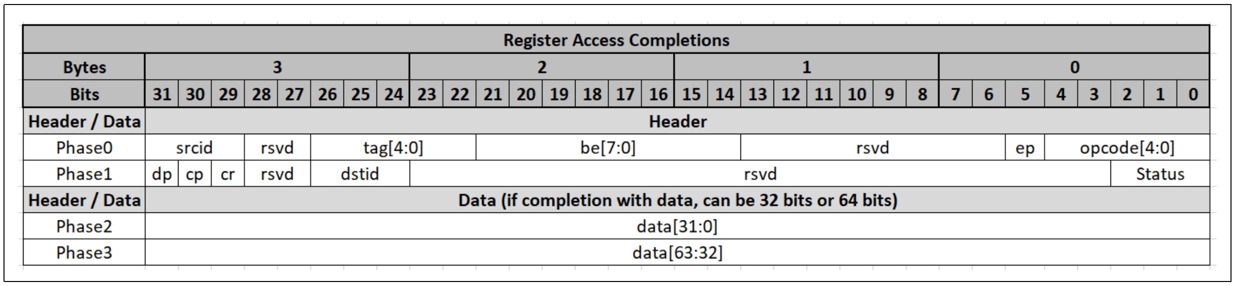
Figure 7-1. Format for Register Access Request



<table>
<tr><th colspan="33">Register Access Request</th></tr>
<tr><th>Bytes</th><th colspan="8">3</th><th colspan="8">2</th><th colspan="8">1</th><th colspan="8">0</th></tr>
<tr><th>Bits</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Header / Data</td><td colspan="32">Header</td></tr>
<tr><td>Phase0</td><td colspan="3">srcid</td><td colspan="2">rsvd</td><td colspan="5">tag[4:0]</td><td colspan="8">be[7:0]</td><td colspan="8">rsvd</td><td>ep</td><td colspan="5">opcode[4:0]</td></tr>
<tr><td>Phase1</td><td>dp</td><td>cp</td><td>cr</td><td colspan="2">rsvd</td><td colspan="3">dstid</td><td colspan="24">addr[23:0]</td></tr>
<tr><td>Header / Data</td><td colspan="32">Data (if applicable, can be 32 bits or 64 bits)</td></tr>
<tr><td>Phase2</td><td colspan="32">data[31:0]</td></tr>
<tr><td>Phase3</td><td colspan="32">data[63:32]</td></tr>
</table>

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]				
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.

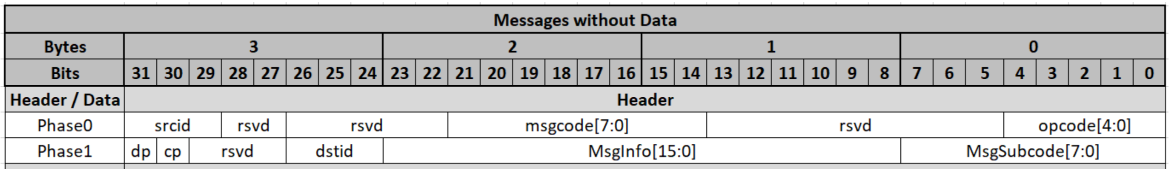
Table 7-7. Field Descriptions for a Completion

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 UCIe, 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 UCIe, this is a completion with 64b Data, and the Data carries original request header that resulted in UR. 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 (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 (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.Adapter 0.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.Adapter 0.Req.L1}		04h	Reserved	L1 Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Req.L2}		08h		L2 Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Req.LinkReset}		09h		LinkReset Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Req.Disable}		0Ch		Disable Request for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.Active}	04h	01h	0000h: Regular Response FFFFh: Stall Response	Active Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.PMNAK}		02h		PMNAK Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.L1}		04h		L1 Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.L2}		08h		L2 Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.LinkReset}		09h		LinkReset Response for Stack 0 Adapter LSM.
{LinkMgmt.Adapter 0.Rsp.Disable}		0Ch		Disable Response for Stack 0 Adapter LSM.

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

Name	Msgcode	Msgsubcode	MsgInfo	Description
{LinkMgmt.Adapter 1.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.Adapter 1.Req.L1}		04h	Reserved	L1 Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Req.L2}		08h		L2 Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Req.LinkReset}		09h		LinkReset Request for Stack 1 Adapter LSM.
{LinkMgmt.Adapter 1.Req.Disable}		0Ch		Disable Request for Stack 1 Adapter LSM.
{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	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.
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 4)

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 0h: Repair **RCLKP_L** 1h: Repair **RCLKN_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 4)

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 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 4)

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.TXSELFCAL Done req}	0000h	B5h	05h
{MBTRAIN.TXSELFCAL 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 4)

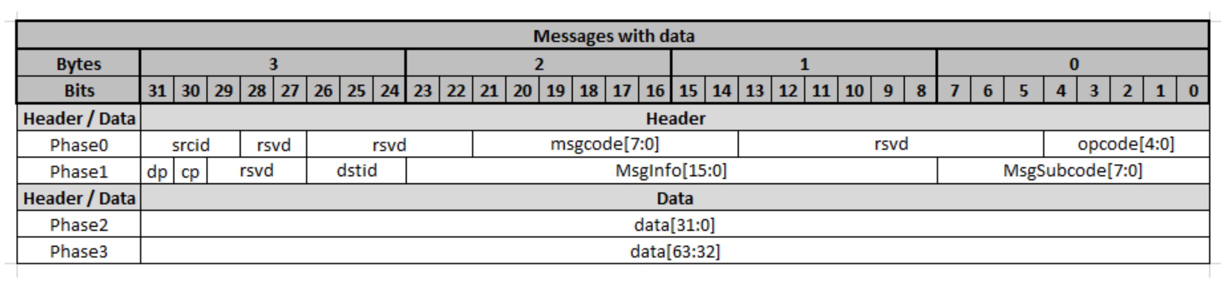
Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{MBTRAIN.LINKSPEED start req}	0000h	B5h	15h
{MBTRAIN.LINKSPEED start resp}	0000h	BAh	15h
{MBTRAIN.LINKSPEED error req}	0000h	B5h	16h
{MBTRAIN.LINKSPEED error resp}	0000h	BAh	16h
{MBTRAIN.LINKSPEED exit to repair req}	0000h	B5h	17h
{MBTRAIN.LINKSPEED 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
{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

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 (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		Finalized Capability of the D2D Adapter

Table 7-10. Message encodings for Messages with Data (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 *CXL Specification*, with the following additional rules: • [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 UCIe, must be 0b. • [14]: Retimer 2 - does not apply for UCIe, must be 0b. • [15]: CXL.io Throttle - must be 0b for PCIe Non-Flit Mode. • [17:16]: NOP Hint Info - does not apply for UCIe, 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		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 (Sheet 3 of 3)

Name	Msg code	Msgsubcode	MsgInfo	Data Bit Encodings	Description
{Vendor Defined Message}	FFh	--	Vendor ID		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.
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	[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)

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]:
	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.
		8Ah	03h	[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]**}

{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]:
	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.
		8Ah	0Bh	[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]**}

{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	[63:15]: Reserved [14]: Sideband feature extensions is supported (1) or not supported (0) [13]: UCIe-A x32 [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 UCIe 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	[63:11]: 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	[63:3]: Reserved [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	[63:3]: Reserved [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	[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 UCIe-A x32 as a transmitter of this message. 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 [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 UCIe-A x32 as a transmitter of this message. 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 [15:8]: **Repair Address for TRD_P[1]**: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme. 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 [7:0]: **Repair Address for TRD_P[0]**: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme. 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

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	[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 UCIe-A x32 as a transmitter of this message. 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 [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 UCIe-A x32 as a transmitter of this message. 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 [15:8]: **Repair Address for TRD_P[1]**: Indicates the physical Lane repaired when TRD_P[1] is used in remapping scheme. 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 [7:0]: **Repair Address for TRD_P[0]**: Indicates the physical Lane repaired when TRD_P[0] is used in remapping scheme. 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

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 a 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.

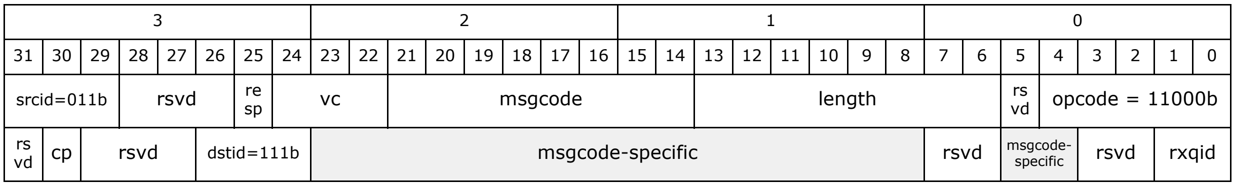
Table 7-12. Supported MPM with Data Messages on Sideband

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



<table>
<tr><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">srcid=011b</td><td colspan="3">rsvd</td><td>resp</td><td colspan="3">vc</td><td colspan="8">msgcode</td><td colspan="8">length</td><td>rsvd</td><td colspan="5">opcode = 11000b</td></tr>
<tr><td>rsvd</td><td>cp</td><td colspan="3">rsvd</td><td colspan="3">dstid=111b</td><td colspan="16">msgcode-specific</td><td colspan="2">rsvd</td><td colspan="2">msgcode-specific</td><td colspan="2">rsvd</td><td colspan="2">rxqid</td></tr>
</table>

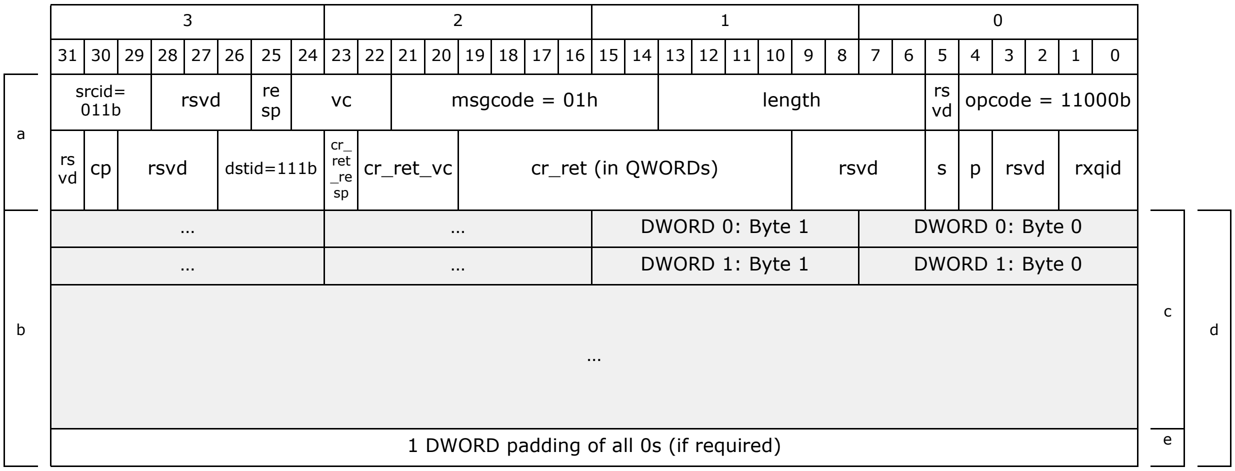
Table 7-13. Common Fields in MPM Header of all MPM with Data Messages on Sideband

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.
dstid	111b: Indicates Management Port Gateway as target.
cp	Control parity for all bits in the sideband packet header.
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).

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



a. MPM Header.
b. MPM Payload.
c. Management Transport Packet (MTP).
d. Length in MPM Header.
e. 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 0s 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.

a. See Section 7.1.2.4.1 for details of header fields common to all MPMs with data on the sideband.

7.1.2.4.3 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 UCIe 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 UCIe 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 UCIe 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

<table>
<tr><td></td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td><td></td></tr>
<tr><td></td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td rowspan="2">a</td><td colspan="3">srcid=011b</td><td colspan="3">rsvd</td><td>re sp = 0</td><td colspan="3">vc</td><td colspan="8">msgcode = FFh</td><td colspan="8">length</td><td>rs vd</td><td colspan="5">opcode = 11000b</td><td></td></tr>
<tr><td>rs vd</td><td>cp</td><td colspan="3">rsvd</td><td colspan="3">dstid=111b</td><td colspan="16">UCIe Vendor ID</td><td colspan="6">rsvd</td><td colspan="2">rxqid</td><td></td></tr>
<tr><td>b</td><td colspan="32">Vendor-defined payload</td><td>c</td></tr>
</table>

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]	UCIe Vendor ID	UCIe 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

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

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 all bits in the sideband packet header.
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

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

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

<table>
<tr><td></td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td></td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">a</td><td colspan="3">srcid=011b</td><td colspan="7">rsvd</td><td colspan="8">msgcode = 02h</td><td>cr_ret_resp_a</td><td colspan="3">cr_ret_vc_a</td><td>cr_ret_resp_b</td><td colspan="3">cr_ret_vc_b</td><td>rsvd</td><td colspan="5">opcode = 10111b</td></tr>
<tr><td>rsvd</td><td>cp</td><td colspan="3">rsvd</td><td colspan="3">dstid=111b</td><td colspan="10">cr_ret_a</td><td colspan="10">cr_ret_b</td><td colspan="2">rsvd</td><td colspan="2">rxqid</td></tr>
</table>

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

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

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

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

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 UCIe 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 UCIe 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 UCIe 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

<table>
<tr><td></td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td></td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">a</td><td colspan="2">srcid=011b</td><td colspan="4">rsvd</td><td>resp = 0</td><td colspan="3">vc</td><td colspan="8">msgcode = FFh</td><td colspan="8">Vendor-defined</td><td>rsvd</td><td colspan="5">opcode = 10111b</td></tr>
<tr><td>rsvd</td><td>cp</td><td colspan="3">rsvd</td><td colspan="3">dstid=111b</td><td colspan="16">UCIe Vendor ID</td><td colspan="6">rsvd</td><td colspan="2">rxqid</td></tr>
</table>

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 UCIe 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.
UCIe Vendor ID	UCIe 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.

7.1.3 Flow Control and Data Integrity

Sideband packets can be transferred across FDI, RDI or the UCIe 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.

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 UCIe 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.

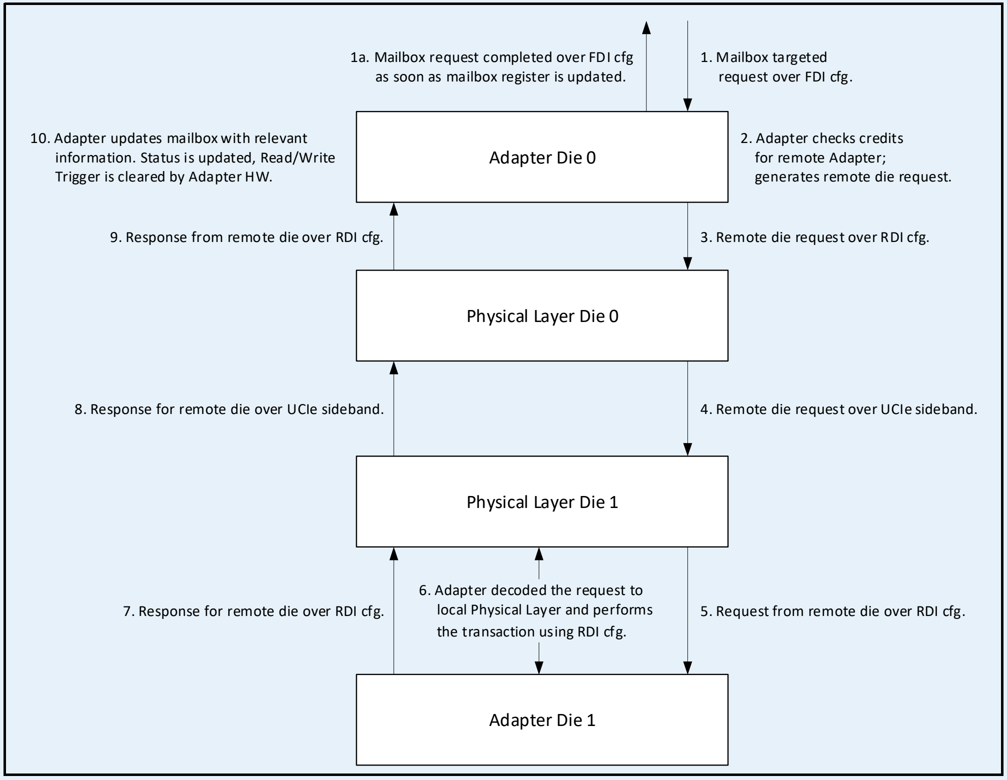
All other messages except Vendor Defined messages must always sink and make forward progress, and not block any messages on the sideband interface behind them. All Link Management message requests have an associated response, and the source of these messages must only have one outstanding request at a time (i.e., one outstanding message per "Link Management Request" MsgCode encoding).

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 7-14 shows an example of an end-to-end register access request to remote die and the corresponding completion returning back.

Figure 7-14. Example Flow for Remote Register Access Request (Local FDI/RDI Credit Checks Are Not Explicitly Shown)


1a. Mailbox request completed over FDI cfg as soon as mailbox register is updated.
1. Mailbox targeted request over FDI cfg.
10. Adapter updates mailbox with relevant information. Status is updated, Read/Write Trigger is cleared by Adapter HW.
Adapter Die 0
2. Adapter checks credits for remote Adapter; generates remote die request.
9. Response from remote die over RDI cfg.
3. Remote die request over RDI cfg.
Physical Layer Die 0
8. Response for remote die over UCIe sideband.
4. Remote die request over UCIe sideband.
Physical Layer Die 1
7. Response for remote die over RDI cfg.
6. Adapter decoded the request to local Physical Layer and performs the transaction using RDI cfg.
5. Request from remote die over RDI cfg.
Adapter Die 1

In Step 1 shown in Figure 7-14, 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 UCIe sideband and returns the RDI credit to Adapter Die 0 once it has freed up its internal buffer space.

IMPLEMENTATION NOTE

Continued

In Step 5, Physical Layer Die 1 checks for Adapter Die 1 credits on RDI before sending the request over RDI. Adapter Die 1 decodes the request to see that it must access a register on Physical Layer Die 1; Adapter Die 1 checks for RDI credits of Physical Layer Die 1 before sending the request over RDI in Step 6. Adapter Die 1 must remap the tag for this request, if required, and save off the original tag of the remote die request as well as pre-allocate a space for the completion. Physical Layer Die 1 completes the register access request and responds with the corresponding completion. Because a completion is sent over RDI, no RDI credits need to be checked or consumed. Adapter Die 1 generates the completion for the remote die request and sends it over RDI (no credits are checked or consumed for completion over RDI) in Step 7. The completion is transferred across the different hops as shown in Figure 7-14 and finally sunk in Adapter Die 0 to update the mailbox information. No RDI credits need to be checked for completions at the different hops.

For forward progress to occur, the Adapters and Physical Layers on both die must ensure that they can sink sufficient requests, completions, and messages to guarantee that there is no Link Layer level dependency between the different types of sideband packets (i.e., remote register access requests, remote register access completions, Link state transition messages for Adapter LSM(s), Link state transition messages for RDI, and Link Training related messages). In all cases, because at most one or two outstanding messages are permitted for each operation, it is relatively easy to provide greater than or the same number of buffers to sink from RDI. For example, in the scenario shown in Figure 7-14, Physical Layer Die 1 must ensure that it has dedicated space to sink the request in Step 6 independent of any ongoing remote register access request or completion from Die 1 to Die 0, or any other sideband message for state transition, etc. Similarly, Physical Layer Die 1 must have dedicated space for remote die register access completion in Step 7.

Dynamic coarse clock gating is permitted in Adapter or Physical Layer in a subset of the RDI states (see Chapter 10.0). Thus, when applicable, any sideband transfer over RDI or FDI must follow the clock gating exit handshake rules as defined in Chapter 10.0. It is recommended to always perform the clock exit gating handshakes for sideband transfers if implementations need to decouple dependencies between the interface status and sideband transfers.

Implementations of the Physical Layer and Adapter must ensure that there is no receiver buffer overflow for messages being sent over the UCIe sideband Link. This can be done by either ensuring that the time to exit clock gating is upper bounded and less than the time to transmit a sideband packet over the UCIe sideband Link, OR that the Physical Layer has sufficient storage to account for the worst-case backup of each sideband message function (i.e., remote register access requests, remote register access completions, Link state transition messages for Adapter LSM(s), Link state transition messages for RDI, and Link Training related messages). The latter offers more-general interoperability at the cost of buffers.

7.1.4 Operation on RDI and FDI

The same formats and rules of operation are followed on the RDI and FDI. The protocol is symmetric — requests, completions, and messages can be sent on **lp_cfg** as well as on **pl_cfg** signals. Implementations must ensure deadlock-free operation by allowing a sufficient quantity of sideband packets to sink and unblock the sideband bus for other packets. At the interface, these transactions are packetized into multiple phases depending on the configuration interface width (compile time parameter). Supported interface widths are 8, 16, or 32 bits. **lp_cfg_vld** and **pl_cfg_vld** are asserted independently for each phase. They must be asserted on consecutive clock cycles for transferring consecutive phases of the same packet. They may or may not assert on consecutive clock cycles when transferring phases of different packets. For packets with data, 64b of data is always transmitted over RDI or FDI; for 32b of valid payload, the most-significant 32b (Phase 4) of the packet are assigned to 0b before transfer.

§ §

8.0 System Architecture

8.1 UCIe Manageability

8.1.1 Overview

UCIe Manageability is optional and defines mechanisms to manage a UCIe-based SiP that is independent of UCIe mainband protocols. This accelerates the construction of a UCIe-based SiP by allowing a common manageability architecture and hardware/software infrastructure to be leveraged across implementations.

Examples of functions that may be performed using UCIe Manageability include the following:

- Discovery of chiplets that make up an SiP and their configuration,
- Initialization of chiplet structures, and parameters (i.e., serial EEPROM replacement),
- Firmware download,
- Power and thermal management,
- Error reporting,
- Performance monitoring and telemetry,
- Retrieval of log and crash dump information,
- Initiation and reporting of self-test status,
- Test and debug, and
- Various aspects of chiplet security.

UCIe manageability has been architected with the following goals:

- Support for management using mainband or sideband is mainband protocol agnostic allowing it to be used with chiplets that implement existing mainband protocols, mainband protocols that may be standardized in the future, and vendor-specific mainband protocols.
- The required core capabilities of UCIe manageability may be realized in hardware allowing simple chiplets to remain simple while providing advanced manageability capabilities (i.e., those that may require firmware implementation) to be implemented in chiplets that require these capabilities.
- UCIe Chiplets that support manageability may be used to realize products for a variety of markets. These markets may have vastly different manageability and security requirements. UCIe manageability defines a menu of optional management and security capabilities that build on top of the required core capabilities.
- UCIe manageability is intended to foster an open chiplet ecosystem where SiPs may be constructed from chiplets produced by different vendors. This means that common features and capabilities that are generally useful across market segments are standardized. Mechanisms are supplied for vendor-specific extensions.

- Manageability capabilities are discoverable and configurable, allowing a common firmware base to be rapidly used across SiPs.
- UCIe manageability builds on top of applicable industry standards.

8.1.2 Theory of Operation

A UCIe-based System-in-Package (SiP) that supports manageability contains one or more UCIe Chiplets that support UCIe manageability. Chiplets in the SiP that support UCIe manageability form a Management Domain. The SiP may also contain chiplets that do not support UCIe manageability and these chiplets are outside the Management Domain. If the Management Domain contains more than one chiplet, then the chiplets are interconnected through Management Ports using chiplet-to-chiplet management links to form a Management Network. The Management Network is fully connected meaning that there is a path from any chiplet in the Management Domain to any other chiplet in the Management Domain.

A UCIe Chiplet that supports manageability contains a Management Fabric and one or more Management Entities. A Management Entity is a Management Element, a Management Port, or a Management Bridge. An example UCIe chiplet that supports manageability is shown in Figure 8-1 and an example SiP that supports manageability consisting of four UCIe chiplets is shown in Figure 8-2.

Figure 8-1. Example UCIe Chiplet that Supports Manageability

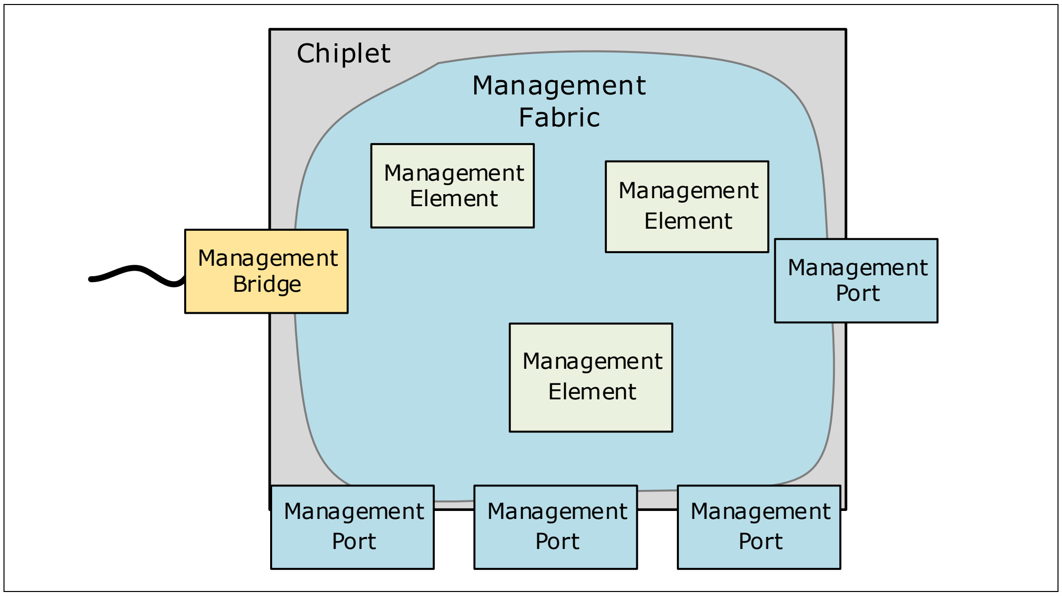
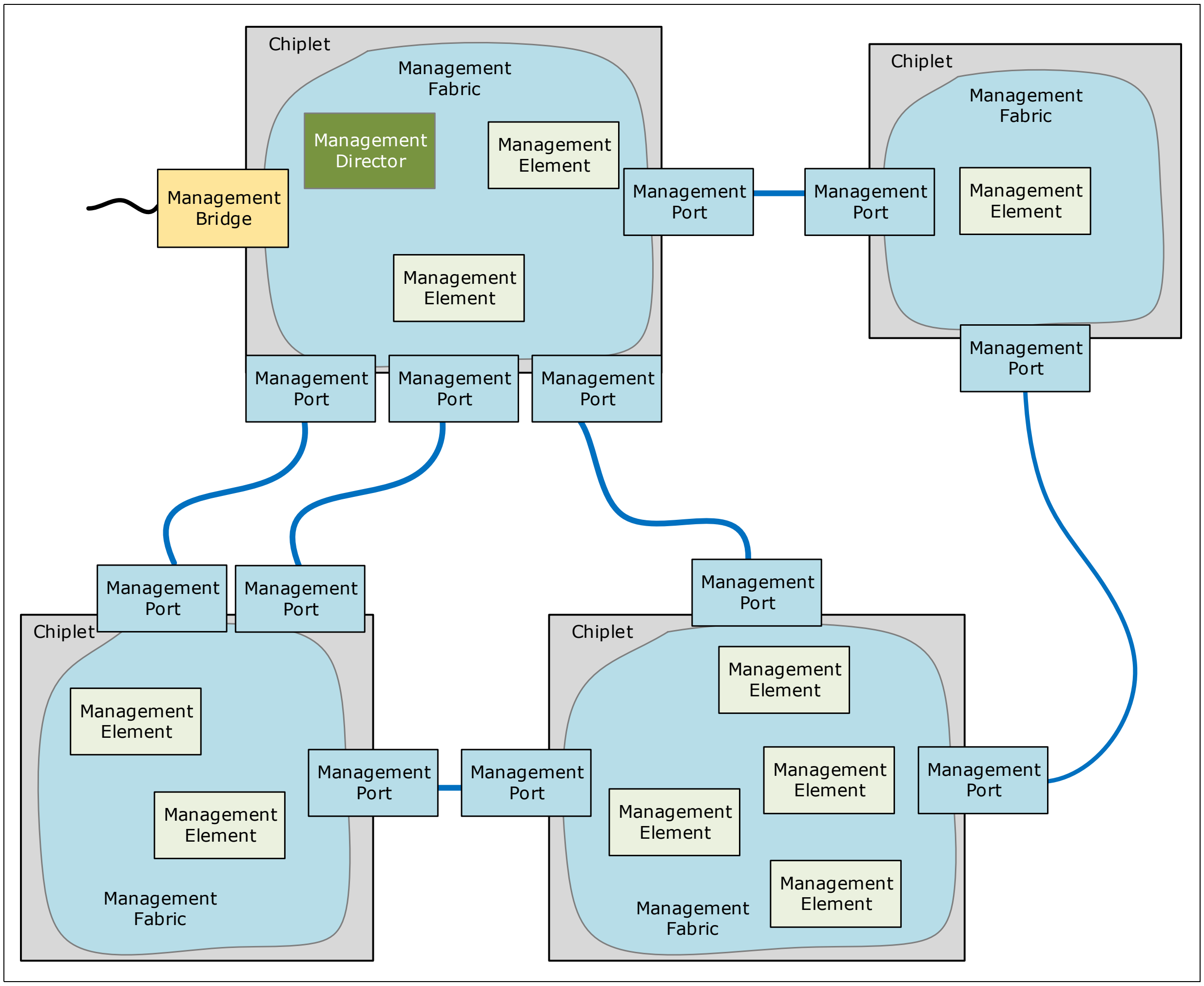
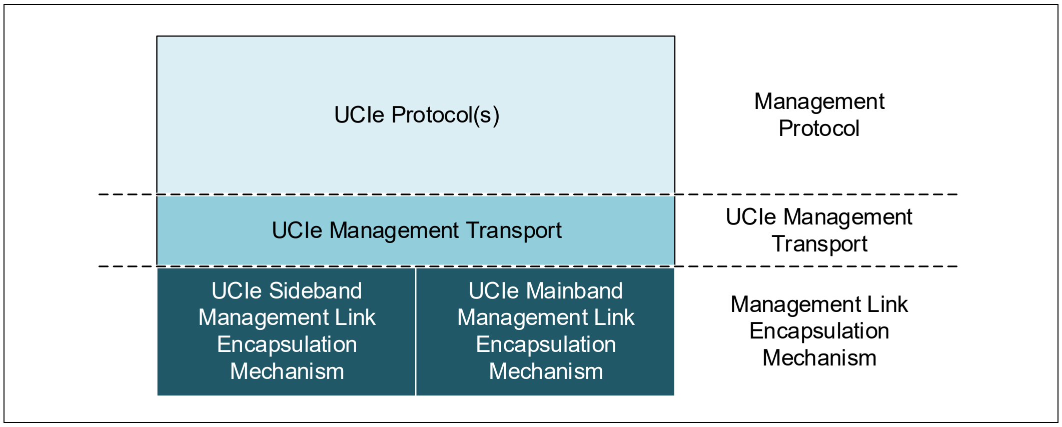


Figure 8-2. Example SiP that Supports Manageability



The UCIe Management Transport is an end-to-end media-independent protocol for management communication on the Management Network. This includes between Management Entities within a chiplet as well as between Management Entities in different chiplets. As shown in Figure 8-3, the Management Protocols above the UCIe Management Transport are used to implement management services. An example of a Management Protocol is the UCIe Memory Access Protocol.

Figure 8-3. UCIe Manageability Protocol Hierarchy


UCIe Protocol(s)
Management Protocol
UCIe Management Transport
UCIe Management Transport
UCIe Sideband Management Link Encapsulation Mechanism
UCIe Mainband Management Link Encapsulation Mechanism
Management Link Encapsulation Mechanism

A Management Port is a Management Entity that acts as the interface between the Management Fabric within a chiplet and a point-to-point management link that interconnects two chiplets. As shown in Figure 8-3. UCIe Manageability Protocol Hierarchy, below the UCIe Management Transport is a Management Link Encapsulation Mechanism that defines how UCIe Management Transport packets are transferred across a point-to-point management link. Two Management Link Encapsulation Mechanisms are defined, one for the UCIe sideband and one for the UCIe mainband. See Section 8.2 for additional details of Management Link Encapsulation Mechanisms. Whether a specific UCIe sideband or mainband link in a chiplet may function as a point-to-point management link is implementation specific.

A chiplet that supports manageability should support at least one UCIe sideband Management Port. To enable broad interoperability, it is strongly recommended that a chiplet support enough UCIe sideband Management Ports to enable construction of an SiP with a single management domain using only UCIe sideband. If a chiplet supports management applications that require high bandwidth, such as test, debug, and telemetry, then it is strongly recommended that the chiplet support UCIe mainband Management Ports.

A Management Fabric within a UCIe chiplet facilitates communication between Management Entities inside the chiplet. A Management Entity is a Management Element, a Management Port, or a Management Bridge. The Management Fabric may be realized using one or more on-die fabrics and the implementation of the Management Fabric is beyond the scope of this specification.

A Management Element is a Management Entity that implements a management service. A Management Element must support the UCIe Management Transport protocol and one or more Management protocols.

A Management Bridge is a Management Entity that interconnects the Management Network to another interconnect, allowing agents on the Management Network and the other interconnect to communicate. The interconnect associated with a bridge may be internal to an SiP or external to the SiP.

One of the Management Elements within an SiP is designated as the Management Director. An SiP may contain multiple Management Elements that may act as a Management Director; however, there can only be one active Management Director at a time. How the Management Director is selected in such SiPs is beyond the scope of this specification. The roles of the Management Director include the following:

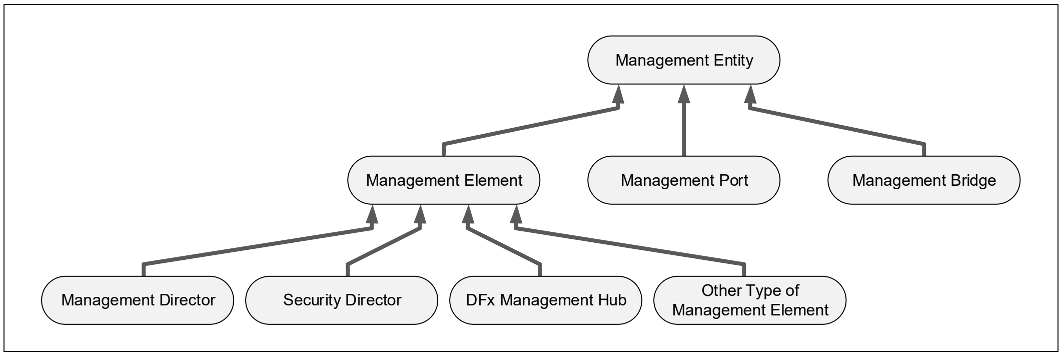
- Discovering chiplets and configuring Chiplet IDs,
- Discovering and configuring Management Elements,

- Discovering and configuring Management Ports,
- Discovering and configuring Management Bridges,
- Acting as the manageability Root of Trust (RoT), and
- Coordinating overall management of the SiP.

One or more of the Management Elements within an SiP may function as a Security Director. A Security Director is responsible for configuring security parameters.

The relationship between the various type of Management Entities is shown in Figure 8-4.

Figure 8-4. Relationship Between the Various Types of Management Entities



Unless otherwise specified, UCIe manageability, Management Entities, and all associated manageability structures in a chiplet (e.g., those in a Management Capability Structure) are reset on a Management Reset. A Management Reset occurs on initial application of power to a chiplet. Other conditions that cause a Management Reset are implementation specific.

8.1.3 UCIe Management Transport

The UCIe Management Transport is a protocol that facilitates communication between Management Entities on an SiP's Management Network. UCIe Management Transport packets are produced and consumed by Management Entities on the Management Network and the packets flow unmodified through the network.

8.1.3.1 UCIe Management Transport Packet

Figure 8-5 shows the fields that make up a UCIe Management Transport packet. The first two DWORDs contain the packet header. This is followed by a protocol specified field defined by the Management Protocol carried in the packet. Finally, a UCIe Management Transport packet may optionally contain a Packet Integrity value computed over the previous contents of the packet. If present, Packet Integrity is one DWORD in size.

Reserved fields in a UCIe Management Transport packet must be filled with 0s when the packet is formed. Reserved fields must be forwarded unmodified on the Management Network and ignored by receivers. An implementation that relies on the value of a reserved field in a packet is non-compliant.

Figure 8-5. UCIe Management Transport Packet

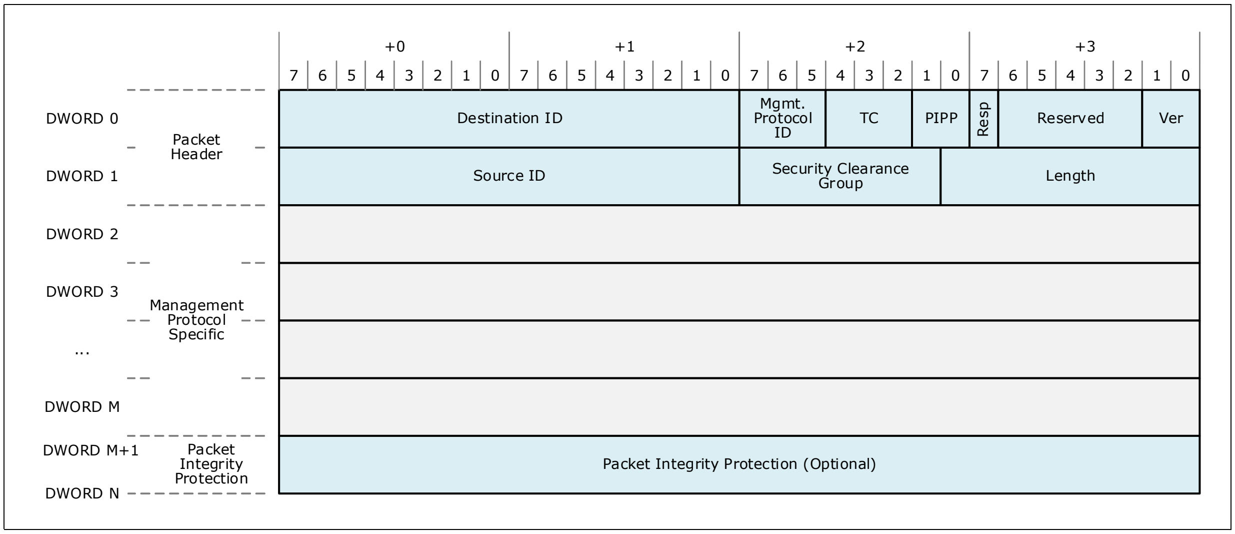


Table 8-1 defines the fields of a UCIe Management Transport packet. All fields in the table have little endian bit ordering (e.g., Destination ID bits 0 through 7 are in Byte 1 with bit 0 of the Destination ID in Byte 1 bit 0, and Destination ID bits 8 through 15 are in Byte 0 with Destination ID bit 8 in Byte 0 bit 0).

Table 8-1. UCIe Management Transport Packet Fields (Sheet 1 of 2)

Field Name	Field Size	Description
Destination ID	16 bits	**Destination ID** This field specifies the Management Network ID of the entity on the Management Network that is to receive the packet.
Mgmt. Protocol ID	3 bits	**Management Protocol ID** This field contains an ID that corresponds to a Management Protocol and specifies the type of payload contained in the Management Protocol Specific field. See Section 8.1.3.1.3 for ID values.
TC	3 bits	**Traffic Class** This field is a packet label used to enable different packet servicing policies. Each Traffic Class is a unique ordering domain with no ordering guarantees between packets in different Traffic Classes. See Section 8.1.3.1.1.
PIPP	2 bits	**Packet Integrity Protection Present (PIPP)** A nonzero value in this field indicates that the packet contains a packet integrity protection value in the Packet Integrity Protection field. The type of packet integrity is indicated by the nonzero value. See Section 8.1.3.4 for more information. 00b: Packet Integrity Protection field is not present in the packet 11b: CRC Integrity (one DWORD in size) Others: Reserved
Resp	1 bit	**Request or Response** This field indicates whether the packet is a request or a response. 0: Request packet 1: Response packet
Reserved	5 bits	**Reserved**
Ver	2 bits	**UCIe Management Transport Packet Version** This field indicates the version of the UCIe Management Transport packet. This field must be set to 00b in this version of the specification. If a Management Entity receives a packet with a UCIe Management Transport packet version that it does not support, then Management Entity must discard the packet.

Table 8-1. UCIe Management Transport Packet Fields (Sheet 2 of 2)

Field Name	Field Size	Description
Source ID	16 bits	**Source ID** This field indicates the Management Network ID of the entity on the Management Network that originated the packet.
Security Clearance Group	7 bits	**Security Clearance Group** This field is used by the UCIe Management Transport access control mechanism. In request packets, this field is set to the security clearance group value associated with the request. In response packets, this field is not used and must be set to 00h.
Length	9 bits	**Length** This field indicates the length of the entire packet in DWORDs. This includes the UCIe Management Network Header, the Management Protocol Specific field, and the Packet Integrity Protection field if present. The length of the packet in DWORDs is equal to the value of this field plus 1 (e.g., a value of 000h in this field indicates a packet length of one DWORD, a value of 001h in this field indicates a packet length of two DWORDs, and so on).
Management Protocol Specific	Varies	**Management Protocol Specific** This field contains Management Protocol specific packet contents. The format of this field is indicated by the Management Protocol field. This field must be an integral number of DWORDs.
Packet Integrity Protection	Varies	**Packet Integrity Protection** If the Packet Integrity Protection Present (PIPP) field in the packet has a nonzero value, then this field is present in the packet and corresponding packet integrity value. See Section 8.1.3.4 for more information. This field must be an integral number of DWORDs. In this version of the specification only CRC integrity protection is supported which is always one DWORD.

A request packet is a packet with the Resp field in the UCIe Management Transport packet header assigned to 0. A requester is a Management Entity that first introduces a request packet into a Management Fabric. A responder is the Management Entity that performs the actions associated with a request that consists of one or more request packets and is the ultimate destination of these request packet(s). A response packet is a packet with the Resp field in the UCIe Management Transport packet header assigned to 1. As a result of performing the actions associated with a request, the responder may introduce one or more response packets into the Management Fabric destined to the requester.

A Management entity that receives a malformed packet must discard the packet.

- A packet with an incorrect length in the Length field is malformed.
  - An example of a malformed packet with an incorrect length in the Length field is one where the Length field in the UCIe Management Transport packet header indicates a length of 65 DWORDs but the actual length of the packet is 64 DWORDs.
- A packet whose size exceeds the Management Transport Packet size supported by a chiplet is malformed.
- A packet that violates a requirement outlined in this specification is malformed.
  - An example of a malformed packet due to a requirement violation is a response packet with a nonzero value in the Security Clearance Group field.

8.1.3.1.1 Traffic Class and Packet Ordering Requirements

Traffic classes (TCs) are used to enable different packet servicing policies. The UCIe Management Transport supports eight traffic classes, and the characteristics of each traffic class are described in Table 8-2. The Management Director configures management fabric routes and may configure different routes for different traffic classes (e.g., TC0 traffic may be routed to UCIe sideband Management Port A and TC2 traffic may be routed to UCIe mainband Management Port B).

Table 8-2. Traffic Class Characteristics

Traffic Class	Description
0	**Default Ordered Lossless Traffic Class** Traffic class optimized for packets that require high reliability and availability. Example: Configuration and health monitoring packets
1	**Low Latency Ordered Lossless Traffic Class** This traffic class has the same characteristics as the Default Ordered Lossless Traffic Class but is intended to be lightly loaded and used for packets that require low latency and should bypass congested Default Ordered Lossless Traffic Class packets. Example: Debug cross trigger and low latency power management packets
2	**Bulk Lossless Ordered Performance Traffic Class** Traffic class optimized for packets associated with bulk traffic. In a properly functioning system without errors, packets in this traffic class are never dropped. Example: Firmware download packets
3	**Bulk Lossy Ordered Performance Traffic Class** Traffic class optimized for packets associated with bulk traffic that may be dropped in the presence of congestion. Example: Debug trace packets
4	**Unordered Lossless Traffic Class** Traffic class optimized for packets that require high performance. Request packets in this traffic class may be delivered out-of-order. Response packets in this traffic class may be delivered out-of-order. Example: High performance UCIe Memory Access protocol accesses.
5 to 6	**Reserved**
7	**Vendor-Specific Traffic Class**

A request packet and an associated response packet need not have the same traffic class. Which traffic classes are allowed and how they are mapped between a request and response are Management Protocol specific.

An implementation shall ensure forward progress on all traffic classes. Quality of service between traffic classes is implementation specific and beyond the scope of this specification.

Each traffic class is a unique ordering domain and there are no ordering guarantees for packets in different traffic classes and there are no ordering guarantees between request and response packets in the same traffic class. Regardless of the traffic class, a response packet associated with any traffic class must be able to bypass a blocked request packet associated with any traffic class.

Within an ordered traffic class, request packets are delivered in-order from a requester to a responder and response packets are delivered in-order from a responder to the requester. There are no ordering guarantees between requests to different responders and there are no ordering guarantees between responses from different responders to a requester.

Within an unordered traffic class there are no ordering guarantees between packets of any type and the chiplet’s Management Fabric is free to arbitrarily reorder packets. While packets may be reordered on an unordered traffic class, there is no requirement that they be reordered (i.e., an implementation is free to maintain ordering as in an ordered traffic class).

Packets associated with a lossy traffic class may be dropped during normal operation. The policy used to determine when a lossy traffic class packet is dropped is vendor defined and beyond the scope of this specification (e.g., due to exceeding a vendor defined congestion threshold). Lossless traffic classes are reliable, and packets associated with a lossless traffic class are not dropped during normal operation; however, packets associated with a lossless traffic class may be dropped due to an error condition. The detection of lost packets and recovery from lost packets is the responsibility of a Management Protocol.

To maintain forward compatibility, a UCIe Management Transport packet with a reserved traffic class value is treated in the same manner as a packet with a traffic class value of zero (i.e., Default Ordered Lossless Traffic Class).

To maintain interoperability, all implementations are required to support all traffic classes; however, an implementation is only required to maintain the ordered and lossless characteristics of a traffic class. All other traffic class characteristics may be ignored. For example, an implementation may treat all traffic classes in the same manner as the Default Ordered Lossless Traffic Class. Under no circumstances may packets in an ordered traffic class be reordered between a requester and a responder. Similarly, under no circumstances may a packet in a lossless traffic class be dropped in a properly functioning system without errors.

> **IMPLEMENTATION NOTE — CHIPLET ROUTE THROUGH**
>
> Chiplet route through occurs when a UCIe Management Transport packet flows through a chiplet (i.e., the packet enters a chiplet through a UCIe Management Port, is not destined to any Management Entity within the chiplet, and the packet matches a Route Entry that causes it to be routed out a UCIe Management Port). Due to chiplet route through it is desirable that chiplets implement the packet servicing policies associated with a TC even if none of the Management Entities within the chiplet utilize that TC. Chiplets are always required to support chiplet route through for all traffic classes.

8.1.3.1.2 Packet Length

The length of a Packet is indicated by the Length field, is an integral number of DWORDs, and consists of the entire length of the packet (i.e., the UCIe Management Network Header, the Management Protocol Specific field, and Packet Integrity Protection field if present.).

The maximum packet length architecturally supported by the UCIe Management Transport is 512 DWORDs (i.e., 2048B). A chiplet may support a maximum packet length that is less than the architectural limit. The Maximum Packet Size (MPS) field in the Chiplet Capability Structure indicates the maximum packet size supported by the chiplet. If a packet larger than that advertised by MPS is received on a Management Port or Management Bridge, then it is silently discarded and not emitted on the chiplet's Management Fabric.

The Configured Maximum Packet Size (CMPS) field in the Chiplet Capability Structure controls the maximum UCIe Management Transport packet size generated by a Management Entity within the chiplet. The initial value of this field corresponds to 8 DWORDs (i.e., 64B). The CMPS field does not affect UCIe Management Transport packets emitted by Management Ports and Management Bridges when forwarding packets into a chiplet's Management Fabric. This allows packets to be routed through the chiplet (e.g., between Management Ports) that are larger than the size of packets generated by Management Entities within the chiplet.

8.1.3.1.3 Management Protocol

The Management Protocol field in a packet contains a Management Protocol ID that indicates the format of the Management Protocol Specific field. Table 8-3 lists the Management Protocols supported by the UCIe Management Transport.

Table 8-3. Management Protocol IDs

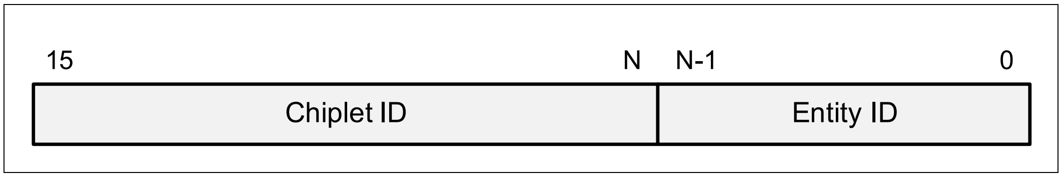
Management Protocol ID	Description
0	Reserved
1	UCIe Memory Access Protocol This protocol is used to access memory mapped structures and memory.
2	UCIe Test and Debug Protocol
3 to 6	Reserved
7	Vendor defined

8.1.3.2 Management Network ID

Management Network IDs are used to uniquely identify Management Entities in the Management Network and to route UCIe Management Transport packets between Management Entities.

As shown in Figure 8-6, a Management Network ID is a 16-bit field that consists of the concatenation of a Chiplet ID and an Entity ID. The Chiplet ID uniquely identifies a chiplet within an SiP and the Entity ID is a fixed value that uniquely identifies a Management Entity within a chiplet. Together the Chiplet ID and Entity ID uniquely identify each Management Entity in an SiP.

Figure 8-6. Management Network ID Format



The size of the Chiplet ID in bits is chiplet implementation specific and may be 2-bits to 15-bits in size. The size of the Entity ID in bits is also chiplet implementation specific and may be 1-bit to 14-bits in size. For each chiplet, the sum of the size of the Chiplet ID and the Entity ID must be 16-bits. The size of the Chiplet ID and Entity ID fields may be different in different chiplets in an SiP (i.e., it is not a requirement that all chiplets of an SiP have the same Chiplet ID field size). To facilitate interoperability an implementation should make the Chiplet ID field as large as possible (i.e., make the Entity ID field only as large as needed to address Management Entities in the chiplet).

The Management Director initializes the Chiplet ID value associated with each chiplet during SiP initialization. This is done by writing the Chiplet ID value to the Chiplet field in the Chiplet Capability Structure and setting the Chiplet ID Valid bit in that structure. The Management Director may determine the size of the Chiplet ID associated with a chiplet by examining the initial value of the Chiplet ID field in the Chiplet Capability Structure or by determining which bits are read-write in this field.

Each Management Entity within a chiplet has a fixed Entity ID. Entity ID zero within each chiplet must be a Management Element that is used to initialize and configure the chiplet. Entity IDs within a chiplet that map to Management Entities may be sparse. For example, a chiplet may contain Management Entities at Management Entity IDs zero, one, and three. The other Entity IDs are unused. A UCIe Management Packet that targets an unused Entity ID within a chiplet is silently discarded by the chiplet’s Management Fabric.

IMPLEMENTATION NOTE — CONFIGURING ROUTING IN AN SIP WITH DIFFERENT CHIPLET ID SIZES

A System-in-Package (SiP) may be composed of chiplets with varying sizes for the Chiplet ID portion of the Management Network ID. To establish routing within such an SiP, one approach is to identify the smallest Chiplet ID size among all chiplets in the SiP. Subsequently, Route Entries (as described in Section 8.1.3.6.2.2) are configured as if each chiplet possessed a Chiplet ID size equivalent to this minimum. For chiplets with a Chiplet ID size exceeding the minimum, any unused Chiplet ID bits are assigned 0 in the Base ID field of a Route Entry and set to 1 in the Limit ID field of a Route Entry.

8.1.3.3 Routing

UCIe Management Transport packets are routed based on the Destination ID field in the UCIe Management Network Header.

The routing of UCIe Management Transport packets differs depending on whether the Chiplet ID value is initialized or uninitialized. The Chiplet ID value is initialized when the Chiplet ID Valid bit is set to 1 in the Chiplet Capability Structure. The Chiplet ID value is uninitialized when the Chiplet ID Valid bit is cleared to 0 in the Chiplet Capability Structure.

A UCIe Management Transport request packet is one where the Resp field is cleared to 0. A UCIe Management Transport response packet is one where the Resp field is set to 1.

While the Chiplet ID and Entity ID size of chiplets may vary in an SiP, all packet routing associated with a chiplet is performed using the Chiplet ID size of the chiplet performing the routing.

The method used to configure UCIe Management Transport routing within an SiP is beyond the scope of this specification. The routing may be pre-determined during SiP design and this static configuration may be used by the Management Director to configure routing. Alternatively, the Management Director may discover the SiP configuration (i.e., chiplets, Management Network topology, and Chiplet ID size of each chiplet) and use this information to dynamically configure routing.

Because the Management Network may contain redundant management links between chiplets as well as links that form cycles, care must be exercised to ensure that the UCIe Management Transport routing is acyclic and deadlock free. In the absence of faults, request packets and response packets are delivered in order; however, it is possible to configure UCIe Management Transport routing such that the path used by a request packet from a requester to a responder is different from path used by a response packet from the responder back to the requester.

8.1.3.3.1 Routing of a Packet from a Management Entity within the Chiplet

This section describes routing of a UCIe Management Transport packet generated by a Management Entity within the chiplet.

- If the chiplet's Chiplet ID value is initialized, then the packet is routed as follows.
  - If the Chiplet ID portion of the packet's Destination ID matches the Chiplet ID value of the chiplet performing the routing, the packet is routed within the chiplet based on the Entity ID portion of the packet's Destination ID.
    - If the Entity ID portion of the packet's Destination ID matches that of a Management Entity within the chiplet, then the packet is routed to that Management Entity.

        - If the Entity ID portion of the packet’s Destination ID does not match that of any Management Entity within the chiplet, then the packet is discarded.
    - If the Chiplet ID portion of the packet’s Destination ID does not match the Chiplet ID value of the chiplet, then the packet is routed based on Management Port Route Entries.
        - If the packet’s Destination ID matches a Route Entry associated with a Management Port, then the packet is routed out that Management Port. See Section 8.1.3.6.2.2 for Route Entry matching rules.
        - If the packet’s Destination ID matches multiple Route Entries within the chiplet and the packet is associated with an ordered traffic class, then the packet is discarded.
        - If the packet’s Destination ID matches multiple Route Entries within the chiplet and the packet is associated with an unordered traffic class, then the packet is routed out one of the Management Ports with a matching Route Entry. Which matching Route Entry is selected in the unordered traffic class case is vendor defined.
        - If the packet’s Destination ID does not match any Route Entries within the chiplet, then the packet is discarded.
- If the chiplet’s Chiplet ID value is uninitialized, then packet is routed as follows.
    - If the packet is a UCIe Management Transport Response packet and the corresponding UCIe Management Transport Request packet was received from a Management Port, then the packet is routed as follows.
        - If the link is up that is associated with Management Port on which the corresponding UCIe Management Transport Request packet was received, then the response packet is routed out that same Management Port on virtual channel zero (VC0). This allows a Management Entity outside the chiplet to configure the chiplet before the chiplet’s Chiplet ID value is initialized.
        - If the link associated with Management Port on which the corresponding UCIe Management Transport Request packet was received is not up, then the response packet is discarded.
    - Otherwise, the packet is routed within the chiplet based on the Entity ID portion of the packet’s Destination ID.
        - If the Entity ID portion of the packet’s Destination ID matches that of a Management Entity within the chiplet, then the packet is routed to that Management Entity.
        - If the Entity ID portion of the packet’s Destination ID does not match that of any Management Entity within the chiplet, then the packet is discarded.

IMPLEMENTATION NOTE — Routing for Uninitialized Chiplet IDs

When a chiplet's Chiplet ID value is uninitialized, then a packet received on a chiplet's Management Port is routed based on the Entity ID portion of the packet's Destination ID to a Management Entity within the chiplet, if one exists. When a response packet is emitted by this Management Entity, the packet is routed out the same Management Port on which the corresponding request was received. This allows an agent external to the chiplet to configure the chiplet before the Chiplet ID and Route Entries are initialized (e.g., following a Management Reset).

When a response packet is routed out the Management Port on which the corresponding request was received, it is routed out the Management Port on virtual channel zero. While there is no requirement that requests to uninitialized chiplets use virtual channel zero (VC0), it is strongly encouraged that virtual channel zero be used.

8.1.3.3.2 Routing of a Packet Received on a Management Port

This Section describes routing of a UCIe Management Transport packet received on a chiplet's Management Port.

- If the chiplet's Chiplet ID value is initialized, then the packet is routed in the same manner is a packet generated by a Management Entity within the chiplet. See Section 8.1.3.3.1 for how such a packet is routed.
- If the chiplet's Chiplet ID value is uninitialized, then the packet is routed within the chiplet based on the Entity ID portion of the packet's Destination ID.
  - — If the Entity ID portion of the packet's Destination ID matches that of a Management Entity within the chiplet, then the packet is routed to that Management Entity.
  - — If the Entity ID portion of the packet's Destination ID does not match that of any Management Entity within the chiplet, then the packet is discarded.

8.1.3.4 Packet Integrity Protection

A UCIe Management Transport packet may optionally contain a Packet Integrity Protection field that is used to protect the integrity of the packet. The presence and type of packet integrity are indicated by the Packet Integrity Protection Present (PIPP) field in the packet.

CRC protection, defined in Section 8.1.3.4.1, is the only packet integrity protection currently defined and is one DWORD in size.

8.1.3.4.1 CRC Integrity Protection

When the PIPP field in a packet is set to 11b, then the Packet Integrity Protection field in the packet is one DWORD in size and contains a 32-bit CRC computed over the previous contents of the packet (i.e., the UCIe Management Network Header and Management Protocol Specific field). Each bit of the Packet Integrity Protection field is set to the corresponding bit of the computed CRC (e.g., bit 31 of the computed CRC corresponds to bit 31 of the Packet Integrity Protection field).

The 32-bit CRC required by this specification is CRC-32C (Castagnoli) which uses the generator polynomial 1EDC6F41h. The CRC is calculated using the following Rocksoft™ Model CRC Algorithm parameters:

```
Name : "CRC-32C"
Width : 32
```

Poly : 1EDC6F41h
Init : FFFFFFFFh
RefIn : True
RefOut : True
XorOut : FFFFFFFFh
Check : E3069283h

When the PIPP field in a packet is set to 11b and the CRC contained in the Packet Integrity Protection field is incorrect, then the packet is discarded.

8.1.3.5 Access Control

The Management Network in an SiP may contain multiple Management Entities that issue requests and multiple Management Entities that expose assets (e.g., structures, keys, or memory). The UCIe Management Transport supports an access control mechanism that may be used by a Management Protocol to prevent unauthorized access to assets by Management Entities on the Management Network.

Some Management Protocols have their own security architecture, so the use of the access control mechanism is Management Protocol specific. Table 8-4 shows which Management Protocols use the access control mechanism.

Table 8-4. Management Protocol use of Access Control Mechanism

Management Protocol	Uses Access Control Mechanism
UCIe Memory Access Protocol	Yes
UCIe Test and Debug Protocol	Yes[a]
Vendor Defined	Vendor Defined

[a] The UCIe Test and Debug Protocol uses the UCIe Memory Access Protocol for configuration and status field accesses and as a result uses the Access Control Mechanism.

The access control mechanism is based on a 7-bit security clearance group value contained in request packets. When a Management Entity emits a request packet on the Management Network, it populates the Security Clearance Group field in the packet with a value that corresponds to the security clearance group associated with the requester. When a Management Entity receives a request packet, it determines whether the asset(s) accessed by the request are allowed or prohibited by that security clearance group.

A Management Entity may emit packets with different security clearance group values. How the security clearance group values that a Management Entity emits are configured or selected is beyond the scope of this specification (see Section 8.1.3.6.4.1).

Although the security clearance group value is seven bits in size, an implementation may choose to restrict the number of security clearance groups. When an implementation restricts the number of security clearance groups to a value N, then security clearance group values from 0 to (N-1) are allowed, and security clearance group values from N to 127 are not allowed. Security Clearance Group 0 is dedicated for use by Security Directors (see Section 8.1.3.5.2).

The method a Management Entity uses to determine whether a request packet is allowed access to an asset is shown in Figure 8-7 and consists of the following steps.

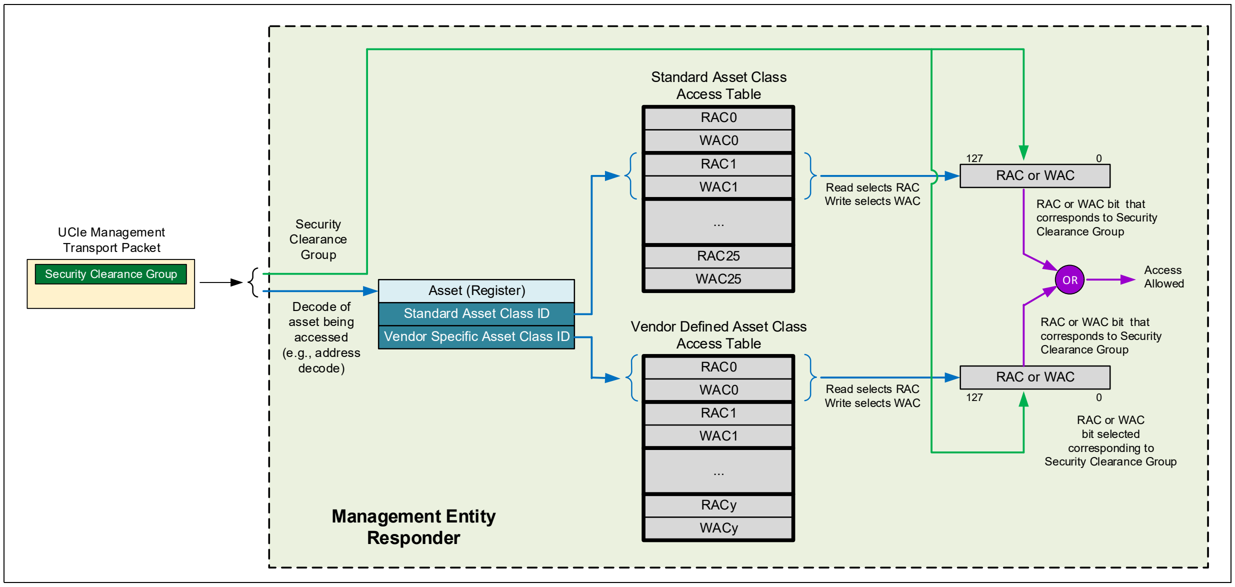
1. The standard asset class ID or the vendor defined asset class ID of the asset being accessed by the request packet is determined.

- — UCIe defines standard asset classes (see Section 8.1.3.5.1) and supports vendor defined asset classes. Each asset must map to a standard asset class, a vendor defined asset class, or both.
- — Associated with each asset class is an asset class ID. The mapping associated with this step produces a standard asset class ID, a vendor defined asset class ID, or both.
- — The manner and granularity in which an implementation maps assets to asset class IDs are beyond the scope of this specification and may be done as part of address decoding, tagging of assets, or some other mechanism.

2. Each asset class ID determined in the previous step is mapped to a 256-bit access control structure.
   - — Associated with each asset class ID is a 128-bit Read Access Control (RAC) structure and a 128-bit Write Access Control (WAC) structure. If an asset's state is being read, then the RAC structure is selected as the access control structure. If an asset's state is being written/ modified, then the WAC structure is selected as the access control structure.
     - o RAC and WAC structures are contained in the Access Control Capability Structure (see Section 8.1.3.6.3).
     - o If a Management Entity does not have any assets that correspond to an asset class ID, then the RAC and WAC structures associated with that asset class ID must be hardwired to 0 (i.e., all the bits on both the RAC and WAC structures are read-only with a value of 0) so no accesses would be allowed.
     - o If an implementation restricts the number of security clearance groups, then RAC and WAC structure bits associated with security clearance groups that are not supported must be hardwired to 0. For example, if an implementation only supports Security Clearance Groups 0 through 3, then bits 4 through 127 must be read-only with a value of 0 in all RAC and WAC structures.
3. The bit corresponding to the security clearance group value in the request packet is examined in each access control structure determined in the previous step to determine whether access to the asset is allowed.
   - — The 7-bit security clearance group value in the request packet is a value from 0 to 127 and this value maps to a corresponding bit in a 128-bit access structure (e.g., security clearance group value 27 maps to access control structure bit 27).
   - — If a bit corresponding to the security clearance group value in an access control structure is set to one, then access to that asset is allowed by that security clearance group. If a bit corresponding to the security clearance group value in an access control structure is set to 0, then access to that asset is prohibited by that security clearance group.
4. If access to the asset is allowed by either the standard asset class or the vendor defined asset, then access to the asset is granted and the request is processed. If access to the asset class is prohibited by both the standard asset class and a vendor defined asset class, then access to the asset is prohibited and the request is not processed.
   - — How a request that attempts to access a prohibited asset is handled is Management Protocol specific.

If a request packet requires access to multiple assets for the request to be serviced, then Step 1 through Step 3 are performed and the request is processed only if access is granted to all assets. If access is prohibited to any asset associated with the request, then no asset is accessed by the request.

Figure 8-7. Access Control Determination in a Responder Management Entity



8.1.3.5.1 Standard Asset Classes

The objective of the standard asset classes is to provide a consistent classification of assets across chiplet implementations and applications for access control. To achieve this, it must be possible for a chiplet implementer to map chiplet assets that are accessible over the Management Network into asset classes without understanding the underlying architecture, roles, or applications associated with an SiP that uses the chiplet.

Standard asset class 0 is for SiP security configuration (e.g., reading and writing the RAC and WAC structures). The remaining standard asset classes are constructed by taking the Cartesian product of a set of asset types and asset contexts and removing elements that are not applicable in practice. Asset types used to construct the standard asset class are shown in Table 8-5 and asset contexts used to construct the standard asset class are shown in Table 8-6. The standard asset classes are shown in Table 8-7.

In some cases, an asset may logically map to two or more standard asset classes. For example, a memory region may contain both SiP data and chiplet data. When this occurs, the asset should be mapped to the standard asset class with the lowest ID value.

In some cases, further access control granularity may be desired beyond what is offered by the standard asset classes. This granularity may be accomplished by separating the assets into different Management Elements within the chiplet. For example, a chiplet may contain two global secrets and the chiplet implementor may wish to allow one set of requesters access to the first global secret and a separate set of requesters access to the second global secret. By putting the two global assets into two different Management Elements, different security clearance groups may be given access to each global secret. In another example, a chiplet may contain an I/O controller and a memory controller and the chiplet implementor may wish to allow one security clearance group to configure the I/O controller and a different security clearance group to configure the memory controller. This granularity may be achieved by putting the I/O controller and memory controller in different Management Elements.