

Rx 容差互连环境的阻抗目标是：对于 2.5、5.0 和 8.0 GT/s 信道，为 100Ω 差分 和 50Ω 单端；对于 16.0 GT/s 和 32.0 GT/s 信道，85Ω 差分 和 42.5Ω 单端；阻抗容差应保持在±5%或更好。

从校准信道的任一端进行测量时，用于 16.0 GT/s 的校准信道必须满足以下回波损耗掩模：

- ≤ -12 dB for < 4 GHz
- ≤ -8 dB for ≥ 4 GHz and < 12 GHz
- ≤ -6 dB for ≥ 12 GHz and ≤ 16 GHz

从校准信道的任一端进行测量时，用于 32.0 GT/s 的校准信道必须满足以下回波损耗掩模：

- ≤ -12 dB for < 4 GHz
- ≤ -10 dB for ≥ 4 GHz and < 16 GHz
- ≤ -6 dB for ≥ 16 GHz and ≤ 32 GHz

校准信道由一对差分 PCB 迹线组成，其两端通过同轴连接器端接。对于 16.0 GT/s 和 32.0 GT/s，校准信道包括一个 4.0 (16.0 GT/s) 或 5.0 (32.0 GT/s) 附加卡机械规范的连接器，并且边缘指形件放置在距离连接信号发生器的同轴连接器至少 4 dB 处。校准信道的电气特性是根据差分插入损耗掩模定义的，如 Figure 8-23 所示，其中 SDD21 是在 TP3（参见 Figure 8-22）和 TP2 之间测量的。TP4-TP5 之间的连接代表电缆连接，并包含在 SDD21 测量中。通过校准 TP3 上的差分电压摆幅和 TX EQ 可以有效地校准 TP3 之前的损耗，并且不包括在 SDD21 测量中。

虽然 8.0GT/s、16.0 GT/s 和 32.0GT/s 参数掩模的扩展范围不低于 1.0 GHz，但所有校准信道的性能必须在 1.0 GHz 以下，并且直流电阻不得超过 7.5 ohm，由 D+和 D-迹线的电阻之和测得。直流电阻的这一限制保证了校准信道的低频特性与 SDD21 掩模到 DC 的外推一致。由于参考封装模型有不同的损耗，因此包含根端口和其他设备的设备的校准损失目标不同。

对于 16.0 GT/s 和 32.0 GT/s，插入损耗范围 $F_{\text{HIGH-IL-MIN}}$ 至 $F_{\text{HIGH-IL-MAX}}$ 为标称损耗。校准信道必须有一系列的损耗选项，覆盖范围从 $F_{\text{HIGH-IL-MIN}}$ 以下至少 2 dB 到 $F_{\text{HIGH-IL-MAX}}$ 以上 3 dB（例如，对于非根情况，这意味着损耗范围从 -22.5 到 -28.5 dB），连续选项之间的损耗增量为 0.5 dB 或更小。

IMPLEMENTATION NOTE

16.0 GT/s Calibration Channel Reference Design

本节给出了一个为满足本规范要求而构建和测试的 16.0 GT/s 校准信道的示例。校准信道的高级框图如 Figure 8-24 所示。请注意，此示例夹具涵盖了比规范要求更大的损耗范围，并且可以涵盖有 Root 和无 Root 情况。测试夹具包括四个 PCB：

16.0 GT/s Rx Calibration Base Boards

从 SMA 连接器到 CEM 通孔连接器的 16 个差分对（85 欧姆标称阻抗）。有三种不同的基板可实现以下插入损耗范围——基板差分对的插入损耗变化如下：@8.0 GHz，步长为 0.5 dB。

Low-Loss Base Board：4-11.5 dB

Mid-Loss Base Board：12-19.5 dB

High-Loss Base Board：20-27.5 dB

所有走线都在底层以微带形式布线。SMA 连接器和 CEM 连接器采用 8.0GHz 的布局技术进行了优化。

16.0 GT/s Rx Calibration Riser Board

从 SMA 连接器到 Gold Edge Fingers 的 16 个差分对（标称阻抗为 85 Ohm）。对于所有十六对线对，差分对的插入损耗在 8.0 GHz 时都固定为 4 dB 标称值。所有走线都在底层以微带形式布线。SMA 连接器和 CEM 连接器通过 8.0 GHz 的布局技术进行了优化。

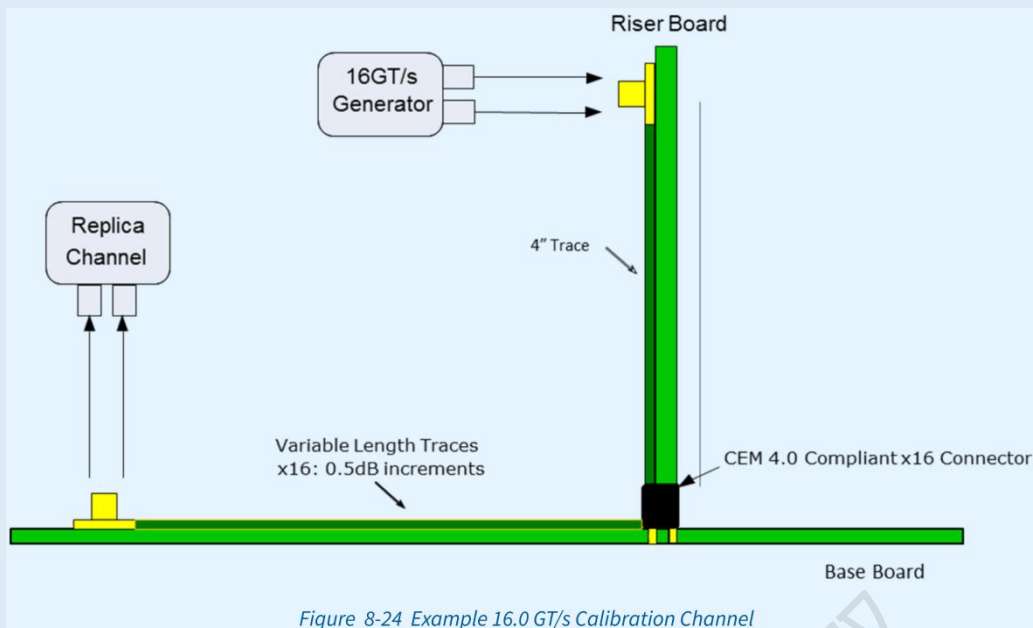


Figure 8-24 Example 16.0 GT/s Calibration Channel

Figure 8-25 显示了两块板的堆积，其中 65%是估计的铜填充百分比。Figure 8-25 包括标称 85Ω和 100Ω堆叠——85Ω堆叠用于校准信道示例。

Material	Layer	Dielectric	Copper Fill/DK	Starting Copper oz	Finished Copper oz	Copper Thickness	Single Ended Impedance	Finished Trace Single Ended	Calculated Impedance	35 Ref. Layers	Differential Impedance	Finished Trace/Gap Differential	Calculated Impedance	Diff Ref. Layers
1-1080, 1-2113	1	.0062	Top	.5	1.5	0.00210	42.5Ω +/-5%	.0135	42.77	2	85Ω +/-5%	.0075 / .005	84.95	2
	2	.0062	Plane	65%	1	0.00130	50Ω +/-5%	.010	50.11	2	100Ω +/-5%	.0052 / .006	100.38	2
FILLER	3	.043	Plane	65%	1	0.00130								
1-1080, 1-2113	4	.0062	Bot	.5	1.5	0.00210	42.5Ω +/-5%	.0135	42.77	3	85Ω +/-5%	.0075 / .005	84.95	3
							50Ω +/-5%	.010	50.11	3	100Ω +/-5%	.0052 / .006	100.38	3
		.0622	Final Thickness (After Plating)											

Figure 8-25 Stackup for Example 16.0 GT/s Calibration Channel

CEM 连接器通孔的焊盘堆叠如 Figure 8-26 所示，SMA 通孔的焊盘堆叠如 Figure 8-27 所示。

Padstack: C40P28P3M3-A59 Type: through Inner pads: Optional											
Layer	Pad Type	Geometry	Width	Height	Offset X	Offset Y	Flash Name	Shape Name			
TOP	ANTI	CIRCLE	59.00	59.00	0.00	0.00					
TOP	THERMAL	RECTANGLE	80.00	80.00	0.00	0.00	TH80X60X15_4X45				
TOP	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
GND-2	ANTI	CIRCLE	59.00	59.00	0.00	0.00					
GND-2	THERMAL	RECTANGLE	80.00	80.00	0.00	0.00	TH80X60X15_4X45				
GND-2	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
GND-3	ANTI	CIRCLE	59.00	59.00	0.00	0.00					
GND-3	THERMAL	RECTANGLE	80.00	80.00	0.00	0.00	TH80X60X15_4X45				
GND-3	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
BOTTOM	ANTI	CIRCLE	59.00	59.00	0.00	0.00					
BOTTOM	THERMAL	RECTANGLE	80.00	80.00	0.00	0.00	TH80X60X15_4X45				
BOTTOM	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
internal_pad_def	ANTI	CIRCLE	59.00	59.00	0.00	0.00					
internal_pad_def	THERMAL	RECTANGLE	80.00	80.00	0.00	0.00	TH80X60X15_4X45				
internal_pad_def	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
SOLDERMASK_TOP	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
SOLDERMASK_BOTTOM	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
PASTEMASK_TOP	REGULAR	NULL	0.00	0.00	0.00	0.00					
PASTEMASK_BOTTOM	REGULAR	NULL	0.00	0.00	0.00	0.00					
FILMMASKTOP	REGULAR	NULL	0.00	0.00	0.00	0.00					
FILMMASKBOTTOM	REGULAR	NULL	0.00	0.00	0.00	0.00					
Drill Data for C40P28P3M3-A59											
Hole Type	Drill Dia	Plating	Figure	Characters	Width	Height	Offset X	Offset Y	Pos Tolerance	Neg Tolerance	Non-Standard
CIRCLE DRILL	28.00	PLATED	DIAMOND		50.00	50.00	0.00	0.00	3.00	3.00	

Figure 8-26 CEM Connector Drill Hole Pad Stack

Padstack: C60_BOT75P20_SSM40P3M3 Type: through Inner pads: Optional											
Layer	Pad Type	Geometry	Width	Height	Offset X	Offset Y	Flash Name	Shape Name			
TOP	ANTI	CIRCLE	45.00	45.00	0.00	0.00					
TOP	THERMAL	RECTANGLE	5.00	5.00	0.00	0.00	5MIL_PAD				
TOP	REGULAR	CIRCLE	60.00	60.00	0.00	0.00					
GND-2	ANTI	CIRCLE	45.00	45.00	0.00	0.00					
GND-2	THERMAL	RECTANGLE	5.00	5.00	0.00	0.00	5MIL_PAD				
GND-2	REGULAR	CIRCLE	60.00	60.00	0.00	0.00					
GND-3	ANTI	CIRCLE	45.00	45.00	0.00	0.00					
GND-3	THERMAL	RECTANGLE	5.00	5.00	0.00	0.00	5MIL_PAD				
GND-3	REGULAR	CIRCLE	60.00	60.00	0.00	0.00					
BOTTOM	ANTI	CIRCLE	45.00	45.00	0.00	0.00					
BOTTOM	THERMAL	RECTANGLE	5.00	5.00	0.00	0.00	5MIL_PAD				
BOTTOM	REGULAR	CIRCLE	75.00	75.00	0.00	0.00					
internal_pad_def	ANTI	CIRCLE	45.00	45.00	0.00	0.00					
internal_pad_def	THERMAL	RECTANGLE	5.00	5.00	0.00	0.00	5MIL_PAD				
internal_pad_def	REGULAR	CIRCLE	60.00	60.00	0.00	0.00					
SOLDERMASK_TOP	REGULAR	CIRCLE	60.00	60.00	0.00	0.00					
SOLDERMASK_BOTTOM	REGULAR	CIRCLE	40.00	40.00	0.00	0.00					
PASTEMASK_TOP	REGULAR	NULL	0.00	0.00	0.00	0.00					
PASTEMASK_BOTTOM	REGULAR	NULL	0.00	0.00	0.00	0.00					
FILMMASKTOP	REGULAR	NULL	0.00	0.00	0.00	0.00					
FILMMASKBOTTOM	REGULAR	NULL	0.00	0.00	0.00	0.00					
Drill Data for C60_BOT75P20_SSM40P3M3											
Hole Type	Drill Dia	Plating	Figure	Characters	Width	Height	Offset X	Offset Y	Pos Tolerance	Neg Tolerance	Non-Standard
CIRCLE DRILL	20.00	PLATED	CIRCLE	E	60.00	60.00	0.00	0.00	3.00	3.00	

Figure 8-27 Pad Stack for SMA Drill Holes

IMPLEMENTATION NOTE

32.0 GT/s Calibration Channel Reference Design

本节提供了一个 32.0 GT/s 校准信道的示例，该信道已构建并经过测试以满足本规范的要求。校准信道的高级框图如 Figure 8-28 所示。请注意，此示例夹具涵盖了比规范要求更宽的损耗范围，并且可以涵盖有 Root 和 non-Root 的情况。测试夹具包括两个 PCB：

32.0 GT/s Rx Calibration Base Boards

在 Megtron-6 PCB 上从 MMPX 连接器到 CEM 表面贴装连接器的 16 个差分对（85 欧姆标称阻抗）。有三种不同的基板可实现以下插入损耗范围——基板差分对的插入损耗变化如下@16.0 GHz，步长为 0.5 dB。

Low-Loss Base Board： 4.0 - 11.5 dB

Mid-Loss Base Board： 12.0 - 19.5 dB

High-Loss Base Board： 20.0 - 27.5 dB

所有的走线都在顶层以微带线形式布线。MMPX 连接器和 CEM 连接器采用 16.0GHz 的布局技术进行了优化。

有关 MMPX 连接器的信息，请参阅 Huber+Suhner 微型连接器。

32.0 GT/s Rx Calibration Riser Board

在 Megtron-6 PCB 上从 MMPX 连接器到 Gold Edge Finger 布线的 16 个差分对（标称阻抗为 85 Ohm）。对于所有十六对线对，差分对的插入损耗在 16.0 GHz 时都固定为 8 dB 标称值。所有的走线都在顶层以微带线形式布线。MMPX 连接器和 CEM 连接器已通过 16.0 GHz 的布局技术进行了优化。

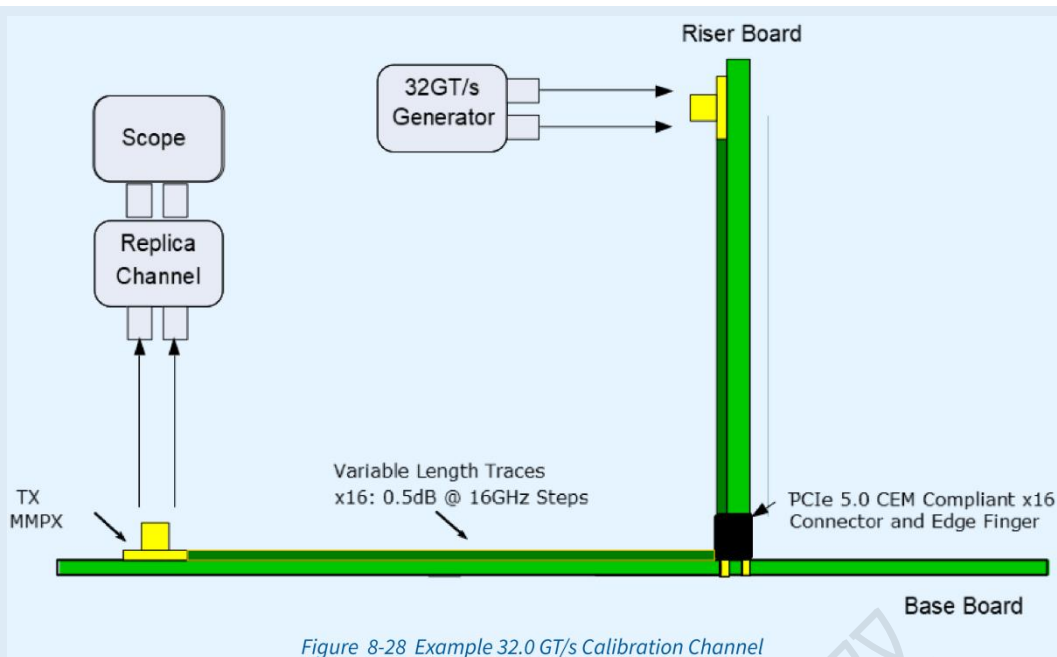


Figure 8-28 Example 32.0 GT/s Calibration Channel

这两个 85 欧姆板的堆叠情况如 Figure 8-29 所示，其中 65%是估计的铜填充百分比。

Material	Layer	Starting Cu oz	Finished Cu oz	Dielectric	Single Ended Lyr	Single Ended Impedance	Designed Trace Width	Finished Trace Width	Calculated Impedance	Ref. Lyr A	Ref. Lyr B	Differential Impedance	Designed Trace/Space	Finished Trace/Space	Calculated Impedance	Ref. Lyr A	Ref. Lyr B
1-3313(54%), 1-1078(75%)	1	Signal	.5	1.5		42.5 Ω	.00945	.0155	42.18 Ω			85 Ω	.00675/.00525	.0115/.0095	85.94 Ω	2	
	2	Plane	1	1		50 Ω	.008	.0115	49.7 Ω			100 Ω	.005/.007	.006/.006	100.45 Ω	2	
Filler Material	3	Plane	1	1													
1-3313(54%), 1-1078(75%)	4	Signal	.5	1.5		50 Ω	.008	.0115	49.7 Ω			100 Ω	.005/.007	.006/.006	100.45 Ω	2	
						42.5 Ω	.00945	.0155	42.18 Ω			85 Ω	.00675/.00525	.0115/.0095	85.94 Ω	2	

Final Thickness (After Plating): 0.062

Figure 8-29 Stack-up for Example 32.0 GT/s Calibration Channel

8.4.1.3 Post Processing Procedures

接收器测试要求在 TP2 (可访问) 处测量应力眼特征，然后进行后处理，以产生 8.0、16.0 和 32.0 GT/s 的 TP2 后处理 (TP2P) (不可访问) 处出现的信号。TP2P 定义了一个参考点，该参考点包含 behavioral Rx package 和 Rx 均衡的影响，并表示可以定义有意义的 EH 和 EW 限值的唯一位置。

8.4.1.4 Behavioral Rx Package Models

Behavioral Rx Package Model 包含在后处理中，以使校准后的眼图能够包含封装的插入损耗信息。分别为 8.0GT/s、16.0 GT / s 和 32.0 GT / s 的眼图校准定义了一对单独的封装模型。在 8.0 GT/s 的速度下，为 TX 和 RX 端口定义了单独的封装模型，以反映大多数接收器实现中典型的较小 CPAD 电容。在 16.0 GT/s 和 32.0 GT/s 的速度下，为包含 Root 端口的设备和所有其他设备定义了单独的封装模型。这对于留出合理的信道解决方案空间是必要的，并且基于这样的假设：含有 Root Complex 的设备通常较大且装有插槽，而所有其他设备往往没有插槽且较小。已构建了 16.0 GT/s 的 Root 和 non-Root 的 Behavioral Rx Package Model，以表示高损耗 (而不是最坏情况下的损耗) 封装的相应封装损耗特性。已构建了 32.0 GT/s 的 Root 和 non-Root 的 Behavioral Rx Package Model，以表示最坏情况下的封装损耗特性。

所有设备的 8.0 GT/s 和 32.0 GT/s 压力眼测试以及支持自带信道的 Non-Root Package 设备的 16.0 GT/s 压力眼测试都必须使用适当的行为封装 (请参阅 Section 8.3.3.11)。对于所有其他设备类型，如果实际 Rx 封装的性能比行为封装的性能差，则允许使用实际的封装模型。如果使用实际封装模型，则必须调整校准通道，以使包括嵌入式实际封装在内的总信道损耗保持在标称值为 28 dB。请注意，仍然需要满足外形规格的总体要求。使用 Section 8.5.1.2 中定义的方法评估 Rx 封装的性能。

信道容差部分的 Section 8.5.1.1 提供了行为封装的详细信息。Behavioral Rx Package Model 的 S 参数模型可用作设计辅助资料。假定封装模型焊盘侧的参考阻抗为 $2 \times 50 \Omega$ 。

8.4.1.5 Behavioral CDR Model

后处理应包括具有数据速率相关传递函数的 Behavioral CDR Model。一阶 CDR 传输函数用于除 32.0 GT/s 和在不支持 32.0 GT/s 时（可选）以 16.0 GT/s 运行的以 CC Refclk 架构运行的接收器。对于在 IR Refclk 模式下运行的接收器，需要备用的 CDR 传输函数。对于给定的数据速率，用于 Rx 测试的 Behavioral CDR 与用于 Tx 测试的相应 CDR 相同。有关 Behavioral CDR 功能的详细信息，请参见 Section 8.3.5.5。

8.4.1.6 No Behavioral Rx Equalization for 2.5 and 5.0 GT/s

当设置了适当的 Tx 预设时，最坏情况的通道、behavioral Rx package 和 2.5 GT/s 和 5.0 GT/s 的 Tx 抖动的组合将产生打开的眼图。因此，无需定义 Behavioral Rx Equalization 或调整 Tx 均衡设置。2.5 GT/s 和 5.0 GT/s 接收器的实际实现当然可以包括均衡。

8.4.1.7 Behavioral Rx Equalization for 8.0, 16.0, and 32.0 GT/s

在 TP2 上进行测量时，通常会关闭 8.0 GT/s、16.0 GT/s 和 32.0 GT/s 的压力眼，因此无法直接测量压力眼抖动参数。通过使用 Behavioral Rx Equalization 可以解决此问题，该均衡器同时实现 CTLE 和 1-tap (8.0 GT/s) 或 2-tap DFE (16.0 GT/s) 或 3-tap DFE (32.0 GT/s)。

CTLE 和 DFE 的 Rx 均衡算法仅旨在用作存在校准通道 ISI 加上其他信号损伤项的情况下获得打开眼图的方法以及信道一致性的一种方法。行为 Rx 均衡算法不打算用作实现实际接收机均衡的指南。例如，额外的 DFE 抽头在 CTLE 可能与行为均衡器不同和/或 CTLE 选择可能并不总是最佳的实际实现中可能具有显著优势。信道损耗特性可能会随温度和湿度而显著变化，并且实际的接收器必须能够通过这种变化继续在目标 BER 下运行。

8.4.1.8 Behavioral CTLE (8.0 and 16.0 GT/s)

8.0 GT/s 和 16.0 GT/s 的 Behavioral Rx Equalization 定义了具有固定 LF 和 HF 极点的一阶 CTLE，以及根据 Figure 8-31 所示的曲线系列指定的可调 DC 增益 (ADC)。对于 8.0 GT/s 的速率，ADC 可以在 -6 dB 至 -12 dB 的最小范围内以 1.0 dB 的步长进行调节。

$$H(s) = \omega_{p2} \times \frac{s + \omega_{p1} \times A_{DC}}{(s + \omega_{p1}) \times (s + \omega_{p2})}$$
$$\omega_{p1} = \text{pole 1} = 2\pi \times 2 \text{ GHz}$$
$$\omega_{p2} = \text{pole 2} = 2\pi \times 8 \text{ GHz}$$
$$A_{DC} = \text{dc gain}$$

Figure 8-30 Transfer Function for 8.0 GT/s Behavioral CTLE

下图说明了当 ADC 在其最小到最大范围内以 1.0 dB 为步长变化时，CTLE 的增益与频率的关系。

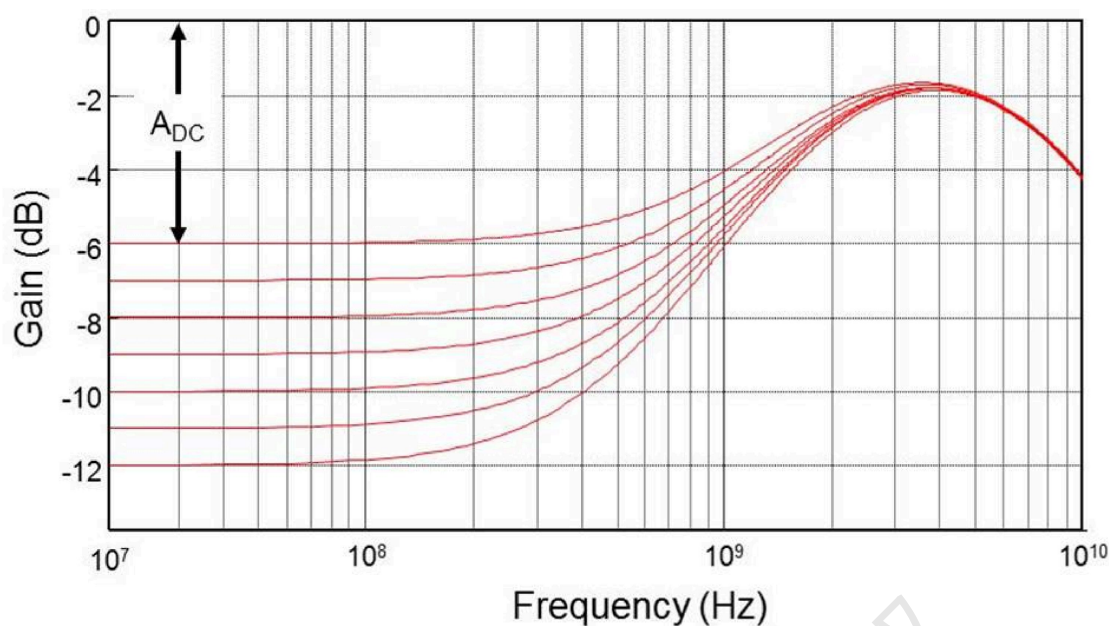


Figure 8-31 Loss Curves for 8.0 GT/s Behavioral CTLE

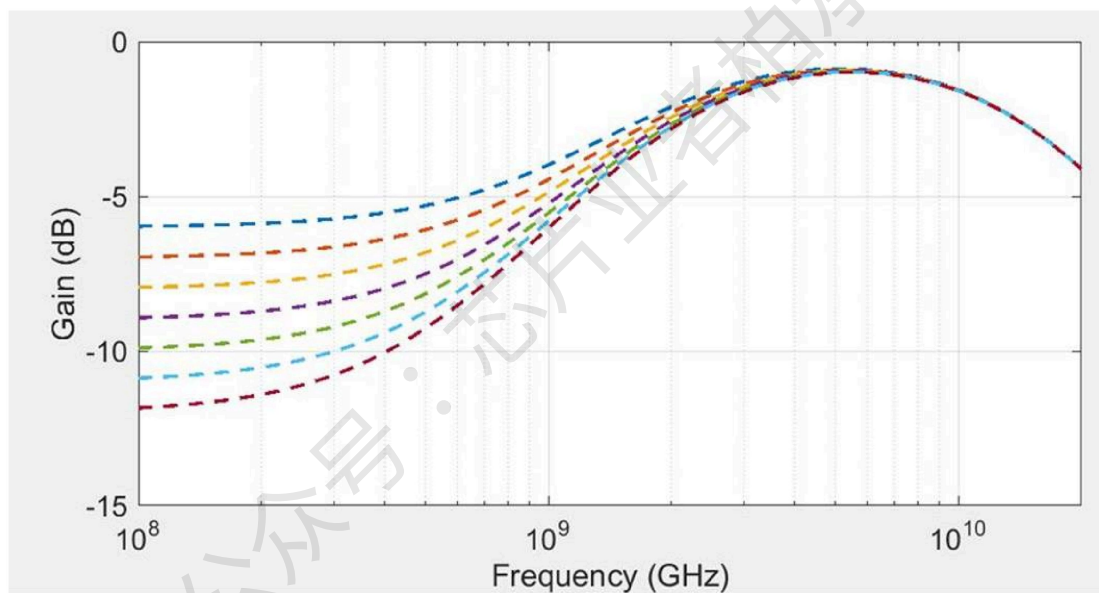


Figure 8-32 Loss Curves for 16.0 GT/s Behavioral CTLE

以 16.0 GT/s 的速度运行的接收器利用一组类似的 CTLE 曲线，但具有不同的极点位置。区别在于， $\omega_{p1} = \text{pole } 1 = 2\pi \cdot 2 \text{ GHz}$ ，而 $\omega_{p2} = \text{pole } 2 = 2\pi \cdot 16.0 \text{ GHz}$ 。ADC 的范围与 8.0 GT/s 的范围相同。

8.4.1.9 Behavioral CTLE (32.0 GT/s)

32.0 GT/s 的 Behavioral Rx Equalization 定义了一个具有固定极点的二阶 CTLE，并根据 Figure 8-33 中所示的曲线系列指定了一个可调的直流增益（ADC）。ADC 可在 -5 dB 至 -15 dB 的范围内以 1.0 dB 的步长进行调节。

$$H(s) = \frac{\omega_{P1} \times \omega_{P3} \times \omega_{P4}}{\omega_{Z1}} \times \frac{(s + \omega_{Z1})(s + \omega_{P2} \times A_{DC})}{(s + \omega_{P1})(s + \omega_{P2})(s + \omega_{P3})(s + \omega_{P4})}$$

$$\omega_x = 2\pi \times F_x$$

$$F_{P1} = 1.65 \times F_{Z1}$$

$$F_{P2} = 9.5 \text{ GHz}$$

$$F_{P3} = 28 \text{ GHz}$$

$$F_{P4} = 28 \text{ GHz}$$

$$F_{Z1} = 450 \text{ MHz}$$

$$F_{Z2} = \text{mag}(\text{DC gain}) \times F_{P2}$$

Equation 8-7 Behavioral CTLE at 32.0 GT/s

Figure 8-33 说明了当 ADC 在其最小到最大范围内以 1.0 dB 为步长变化时，CTLE 的增益与频率的关系。请注意，CTLE 曲线必须在后处理工具中扩展到足够高的频率，以使 CTLE 曲线不会因最大频率而明显限制仿真性能。

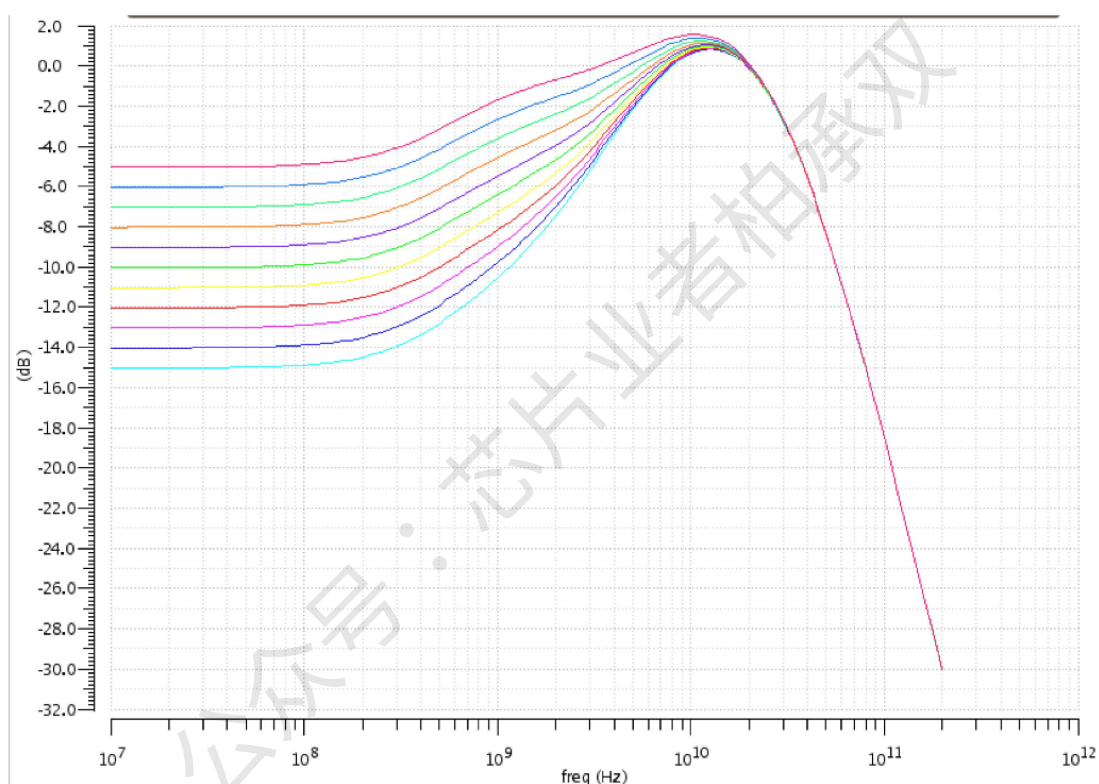


Figure 8-33 Loss Curves for 32.0 GT/s Behavioral CTLE

8.4.1.10 Behavioral DFE (8.0, 16.0, and 32.0 GT/s Only)

在采用最大长度校准通道时，在校准速率为 8.0 GT/s 时，需要结合使用一阶 CTLE 和 1-tap DFE 算法来校准压力眼。DFE 可由以下方程式和流程图表示。对于 8.0 GT/s 和 16.0 GT/s，d1 的限值为±30 mV。对于 32.0 GT/s，d1 的限制定义为 tap 幅度（h1）与 cursor 强度（h0）的比率。h1/h0 比率必须小于或等于 0.8。请注意，h1/h0 限制为 0.8 仅仅是为了限制参考接收器的行为，并不表示实际的实现方案，对于由于 h1/h0 较大而引起的突发错误是安全的，只要 h1/h0 的比率会导致未检测到的数据错误低于 0.8。实现者必须针对最大安全 h1/h0 比率对自己的特定设计进行分析。请注意，以 32.0 GT/s 的速度提供了可选的预编码机制，接收器可以选择启用该机制以降低高转换数据模式中引起静默数据损坏的 DFE 相关错误突发的风险。对于 16.0 GT/s，d2 的限制为±20 mV。对于 32.0 GT/s，d2 和 d3 的限制为±20 mV。

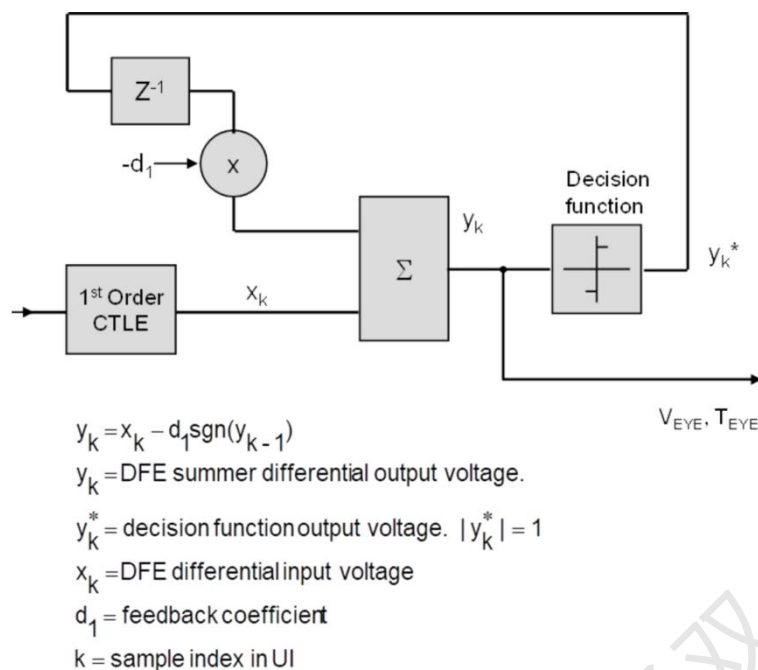


Figure 8-34 Variables Definition and Diagram for 1-tap DFE

16.0 GT/s 接收器容限利用 CTLE 和 2-tap behavioral DFE，如下所示。除了包含第二个抽头以外，它与上面显示的 1-tap DFE 相同。
 32.0 GT/s 接收器容限利用 CTLE 和 3-tap behavioral DFE。

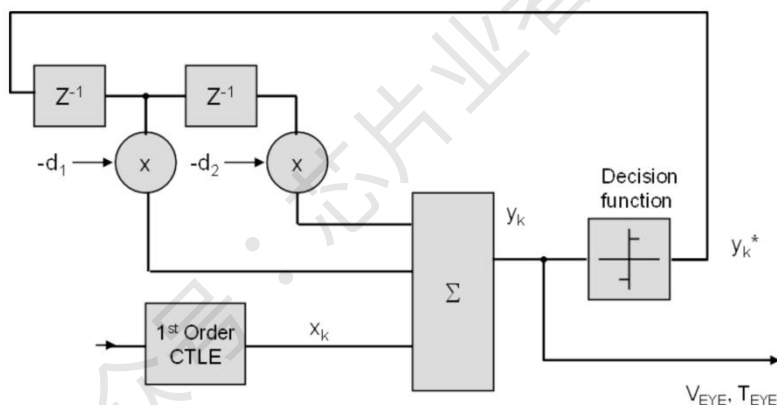


Figure 8-35 Diagram for 2-tap DFE

8. 4. 2 Stressed Eye Test

以 16.0 GT/s 和 32.0 GT/s 的速率进行 Rx 测试，每个数据速率只需要进行一个单独的压力电压/压力抖动测试。

测试接收器时，要求 DUT 上有其他 PCI Express 通道发送或接收数据。同样，如果设备支持其他 I/O，则它也应该在这些接口上进行发送或接收。目的是使 Rx 测试环境尽可能接近地复制真实系统中发现的噪声环境。

8. 4. 2. 1 Procedure for Calibrating a Stressed EH/EW Eye

校准应力电压/抖动眼的目的是为被测接收器同时提供最坏情况裕度，其失真特性类似于真实信道产生的眼。大部分失真由校准信道产生的 ISI 组成。允许 Rj 和差分电压的增量变化分别在 8.0 GT/s 时调整 EW 和 EH。Sj、VRX-DIFF-INT 和差分电压摆幅的增量变化可用于在 16.0GT/s 和 32.0 GT/s 时调整 EW 和 EH。

定义 EH/EW 的参考点对应于 8.0、16.0 和 32.0 GT/s 下接收器锁存器的输入。由于该点在物理上不可访问，因此有必要通过后处理程序构造其等效点。首先对 Compliance Pattern 的 200 万个单位间隔数据进行记录或在 TP2 处平均 1024 次的步长进行后处理，以在数学上包括由行为接收器封装引起的额外信号失真。如果使用 Compliance Pattern Waveform，则如果使用阶跃则除 $V_{RX-CM-INT}$ 外的所有应力都将打开，然后所有应力都将关闭。然后通过 Rx 均衡和行为 CDR 功能来恢复得到的信号，从而得到等效的眼图。用于 EH/EW 校准的波形后处理工具的要求在 Section 8.4.2.1.1 中作了进一步说明。

由于信号发生器输出的校准过程包含发生器直接连接到测量仪器的步骤，因此输出波形的过渡时间可能非常快。因此，用于校准生成器的仪器的带宽必须与生成器输出的边沿速率适当匹配，这一点很重要。本规范要求使用生成器进行 16.0GT/s 测试，其输出的上升时间为 14ps-19ps (20% / 80%)，这也要求示波器的最小带宽为 25GHz。该示波器带宽也是 16.0 GT/s 下发送器测量所需的最小带宽。对于 32.0 GT/s 测试，规范要求使用输出上升时间为 7.5-15.0 ps (用 P4 测量为 20% / 80%) 的生成器，这要求示波器的最小带宽为 50 GHz。该示波器带宽也是 32.0 GT/s 下发送器测量所需的最小带宽。所有数据速率都需要一个最小示波器采样率，该采样率每单位间隔至少捕获 4 个样本。

对于眼图校准过程，将 Tx 均衡固定为预设值，使用后处理工具进行校准时，该预设值可提供最佳的眼图区域。一旦测试过程开始，就可以调整 Tx 预设，以使 DUT 产生最佳的眼图边界。在 EH/EW 校准期间，Sj 设置为 100 MHz 和 0.1 UI。Tx EQ 和差分电压摆幅校准在 TP3 上完成，如 Figure 8-22 所示，并且 TP3 之前的损耗不包括在校准信道损耗中。为了以 16.0 GT/s 的速度进行校准，可以使用以下过程来校准眼图：

1. 将应力值校准为 Table 8-9 中的标称值。
2. 选择一个初始测试信道长度，该长度在 TP2P 时在 8 GHz 时会造成 $27\text{ dB} \pm 0.5\text{ dB}$ 的损耗。
3. 使用预设 $\pm 0.1\text{ dB}$ 的标称 TX EQ 测量每个 TX EQ 预设的眼图，然后选择提供最大眼图面积的 TX EQ 预设。

对于在接收器校准中执行的所有 EH，EW 和眼图面积测量，参考接收器 CTLE 中的 ADC 在其最小到最大范围内以 0.25 dB 的步长变化。这样做是为了提高自动 Rx 校准软件的可重复性和准确性，并且仅用于压力眼图校准（不适用于信道一致性等）。

4. 将校准信道损耗增加到下一个可用的长度/损耗，并在选定的预设位置测量新的眼图。继续增加长度/损耗，直到眼高或眼宽低于 Table 8-9 中的目标，然后选择先前的校准信道的长度/损耗。如果眼高或眼宽均未低于目标值，并且 8 GHz 时 TP2P 的 TP3 (Figure 8-22) 损耗达到 30.0 dB，则进行下一步。
5. 对于选定的校准信道的长度/损耗，测量每个 TX EQ 预设的眼图，然后选择具有最大眼图面积的预设。请注意，由于长度/损耗的变化，这可能是与步骤 3 不同的预设。
6. 调整 Sj， $V_{RX-DIFF-INT}$ 和 Voltage Swing，通过将它们扫过以下范围对眼图进行最终调整：
 - a. Sj 5 to 10 ps PP。
 - b. $V_{RX-DIFF-INT}$ 10 to 25 mV at TP2。
 - c. Differential Voltage Swing 720 to 800 mV PP at TP1。
7. 如果最终的 Sj 值小于 0.1 UI，则 Rj 电平会降低，因此眼图宽度以 100 MHz Sj 的 0.1 UI 满足目标眼图宽度。
8. 如果提供有效解决方案的 Sj、 $V_{RX-DIFF-INT}$ 和 Voltage Swing 的多个组合，则首先选择最接近目标眼宽 (18.75 ps) 的组合。如果存在多个 Sj， $V_{RX-DIFF-INT}$ 和 Voltage Swing 组合，这些组合都同样接近目标眼宽，则选择 Sj 最接近标称宽度的那个。所选的值必须在 BER E-12 的以下范围内给出平均眼图的高度和宽度（至少 5 次测量，稳定值所需的确切测量数量将取决于实验室环境和工具）：
 - a. Eye height 15 mV $\pm 1.5\text{ mV}$
 - b. Eye width 18.75 ps $\pm 0.5\text{ ps}$

为了以 32.0 GT/s 的速度进行校准，可以使用以下过程来校准眼图：

1. 使用预设 ± 0.3 dB 的标称 TX EQ 测量每个 TX EQ 预设的眼图，然后选择提供最大眼图面积的 TX EQ 预设。

对于在接收器校准中执行的所有 EH, EW 和眼图面积测量，参考接收器 CTLE 中的 ADC 在其最小到最大范围内以 1.0 dB 为步长变化。测量眼图，改变以下参数，每个可变参数的最大指示步长为：

- 从 TP1 到 TP2P 的信道损耗从 34.0 dB 到 37.0 dB 不等，最大步长为 0.5 dB。请注意，如果您的实际信道损耗略高于 37 或略低于 34，则将排除这些情况（损耗必须在 34.0 和 37.0 dB 之间）。
 - S_j 从 1 到 5 ps PP 变化，最大步长为 0.25 ps， S_j 在 TP1 处测量。
 - $V_{RX-DIFF-INT}$ 在 TP2 处为 5 至 30 mV，最大步长为 2.5 mV。
2. 如果最终的 S_j 值小于 0.1 UI，则 R_j 电平会降低，因此眼图宽度以 100 MHz S_j 的 0.1 UI 满足目标眼图宽度。
 3. 如果存在多个提供有效解决方案的 S_j , $V_{RX-DIFF-INT}$ 和信道损耗的组合，请首先选择信道损耗最高的组合。如果有多个组合具有最高的信道损耗，请选择最接近目标眼高（15.0 mV）的组合。所选的值必须在 BER E-12 的以下范围内给出平均眼图的高度和宽度（至少 5 次测量，稳定值所需的确切测量数量将取决于实验室环境和工具）。查找满足这些条件的应力值组合的特定方法超出了本规范的范围。当且仅当找不到应力组合时，电压摆幅才能在 720 至 800 mV 之间变化：

请注意，因为第一个 tie-breaker 的损耗最高，所以大多数方法都是从允许的最大信道损耗开始的。

- a. Eye height 15 mV $\pm$ 1.5 mV
- b. Eye width 9.375 ps $\pm$ 0.5 ps

根据测试 Rx 的数据速率，将 R_j 或 S_j 以及差分干扰源调整到 V_{RX-ST} 和 T_{RX-ST} 限制之内。EH 和 EW 范围旨在解决后处理或测量错误。Figure 8-36 显示了以 8.0 GT/s 的速度校准抖动眼图的过程。

Figure 8-37 显示了以 16.0 GT/s 的速度校准抖动眼图的过程。

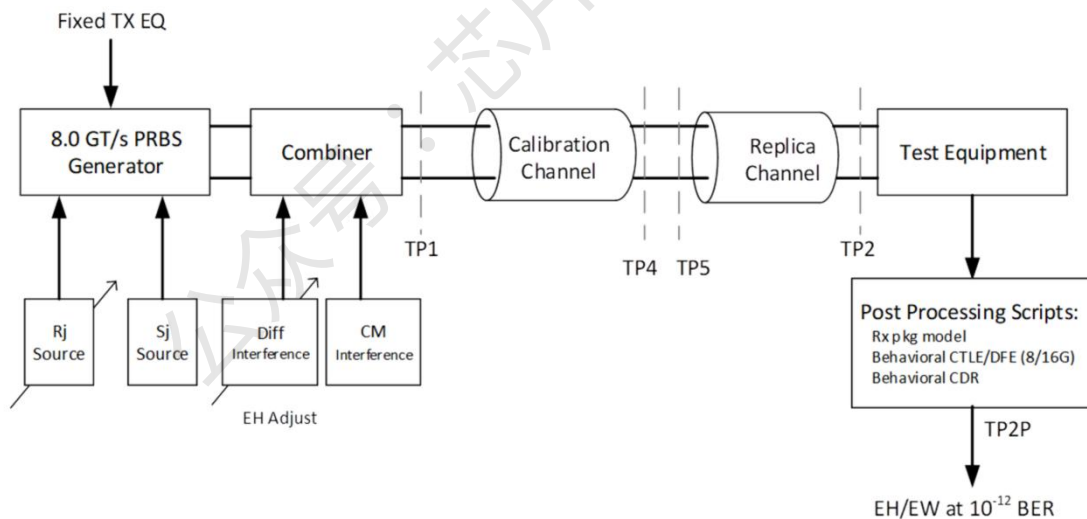


Figure 8-36 Layout for Calibrating the Stressed Jitter Eye at 8.0 GT/s

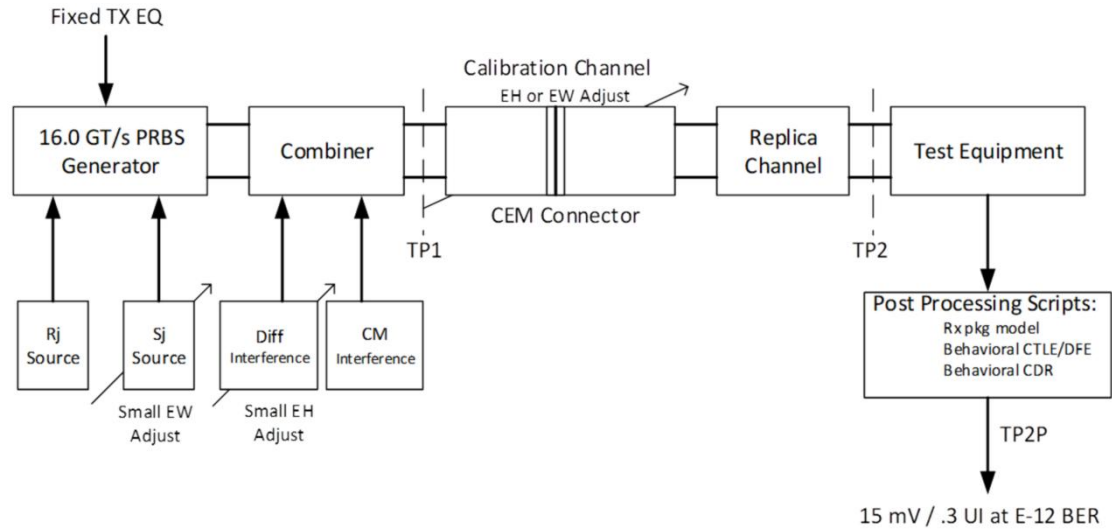


Figure 8-37 Layout for Calibrating the Stressed Jitter Eye at 16.0 GT/s

Table 8-9 Stressed Jitter Eye Parameters

Symbol	Parameter	2.5 GT/s	5.0 GT/s	8.0 GT/s	16.0 GT/s	32.0 GT/s	Units	Details
$V_{RX-LAUNCH}$	Generator launch voltage	800 to 1200	800 to 1200	800 to 1200	720 to 800	720 to 800	mV PP	Note 1
T_{RX-UI}	Unit Interval	400	200	125	62.5	31.25	ps	
T_{RX-ST}	Eye width	≤ 0.4	≤ 0.32	≤ 0.30	≤ 0.30	≤ 0.30	UI	Note 3, 4, 8, 10
V_{RX-ST}	Eye height	≤ 175	≤ 100	≤ 25	≤ 15	≤ 15	mV PP	Note 2, 4, 8, 9
$T_{RX-ST-SJ}$	Swept Sj	N/A	75 ps (max) See Note 11	See Section 8.4.2.2.1	See Section 8.4.2.2.1	See Section 8.4.2.2.1	ps	Note 5
$T_{RX-ST-RJ}$	Random Jitter	N/A	3.4	(max) 3.0	1.0	0.5	ps RMS	Note 6, 7
$V_{RX-DIFF-INT}$	Differential noise	N/A	N/A	14	14	10	mV PP	Note 7, 12 Adjust to set EH. Frequency = 2.1 GHz .
$V_{RX-CM-INT}$	Common mode noise	150	150	150	150	150	mV PP	Note 8

Symbol	Parameter	2.5 GT/s	5.0 GT/s	8.0 GT/s	16.0 GT/s	32.0 GT/s	Units	Details
$V_{SSC-RES}$	SSC Residual	N/A	75	N/A	500	N/A	ps	Note 11, 13

Notes:

1. 可以调整 $V_{RX-LAUNCH}$ 以满足 V_{RX-ST} ，只要 TP2 上的外部眼电压不超过 1300 mVPP（以 2.5、5.0 和 8.0 GT/s 进行校准）即可。对于 16.0 和 32.0 GT/s 的校准， $V_{RX-LAUNCH}$ 从 720 调整到 800 mV。
2. 2.5 GT/s 和 5.0 GT/s 的电压在 Rx 引脚上。
3. 2.5 GT/s 和 5.0 GT/s 的眼宽在 Rx 引脚上。

4. V_{RX-ST} 和 T_{RX-ST} 分别以 TP2P 8.0 GT/s, 16.0 GT/s 和 32.0 GT/s 以及 TP2 2.5 GT/s 和 5.0 GT/s 的参考。对于 8.0 GT/s, 16.0 GT/s 和 32.0 GT/s 的行为均衡将应用于 TP2 上的数据。
5. $T_{RX-ST-SJ}$ 可以在 TP1 或 TP2 处测量。使用 Sj 模板仅测试 8.0 GT/s, 16.0 GT/s 和 32.0 GT/s 的接收器。
6. 可以调整 $T_{RX-ST-RJ}$ 以使其达到 8.0 GT/s 的 T_{RX-ST} 目标值。在 TP1 处测量 R_j , 以防止数据通道交互对 R_j 校准的准确性产生不利影响。在以下范围内应用 R_j : 低频限制可能在 1.5 到 10 MHz 之间, 上限是 1.0 GHz。
7. $T_{RX-ST-RJ}$ 和 $V_{RX-DIFF-INT}$ 都受到限制, 以防止受压的眼睛包含过多的抖动或噪声失真, 这些抖动或噪声失真无法代表真实信道。这些失真分量太多会产生实际接收器无法均衡的信号。
8. 定义为 120 MHz 的单音。在 TP2 上进行的测量无需后处理。在 T_{RX-ST} 和 V_{RX-ST} 校准期间, 共模关闭, 然后在受压的眼抖动测试中打开。
9. 对于 2.5 GT/s 和 5.0 GT/s Rx 校准, 使用可变信道损耗来达到目标眼高。
10. 对于 2.5 GT/s Rx 校准, 使用 100 MHz Sj 达到目标眼宽。
11. 仅当以 5 GT/s 的速度进行测试时, 才有 33 kHz SSC 残余用于 Common Clock architecture 测试。
12. $V_{RX-DIFF-INT}$ 的频率选择为略高于参考 CTLE 的第一极点。
13. 仅适用于 CC 测试, 当在 16.0 GT/s (不支持 32.0GT/s) 和 Figure 8-43 的 Sj 掩模和一阶 CDR 传输函数时, 频率介于 30 kHz 至 33 kHz 之间的三角相位调制。

8.4.2.1.1 Post Processing Tool Requirements

波形后处理工具或信道一致性方法工具可用于以 16.0 或 32.0 GT/s 的 Rx 压力眼图校准。如果使用波形后处理工具来校准 Rx 压力眼图测试的 EH/EW, 则该工具必须与基于以下定义的一致性测试的信道一致性方法学工具一致:

- 测试信道是个长 Rx 校准信道, Root reference package 应用于后处理, 在 8GHz (16.0GT/s) 或 16 GHz (32.0GT/s) 下, 总损耗为 28.0dB。这意味着物理信道损耗为 23.0dB (16.0 GT/s) 或 27.0 dB (32.0 GT/s)。
- 所有的测试都是在 TP2 完成的。
- 通过在实时示波器上平均 1024 次, 在测试信道中捕获具有 256 个 1 和 0 的步进模式。该步进以 1 ps 或更小的 x 轴分辨率保存, 用作信道一致性方法的发送波形。使用每个预设的标称 Tx EQ 为每个预设捕获步进模式。
- 信道一致性方法在仿真中未应用 Tx EQ 的情况下运行, 使用了针对每个捕获步骤的 Rx 压力眼校准的标称应力值。选择产生最大眼图面积的 Tx EQ 预设, 以进行精确的眼图高度和宽度校准。
- 信道一致性方法用于选定的 Tx EQ 预设, 通过调整发送器输出处的 Sj、 $V_{RX-DIFF-INT}$ 和电压摆幅, 产生 15 mV 和 0.3 UI@E-12 BER (16.0 GT/s) 或 15 mV 和 0.3 UI@E-12 BER (32.0 GT/s) 的 EH/EW。
- 校准码型发生器, 使其具有与产生目标眼高和眼宽的信道一致性仿真中所使用的抖动应力水平和 Tx 摆幅相同的抖动应力水平。
- 在信道末端的每个 Tx EQ 处捕获具有 Compliance Pattern 的 200 万个单位间隔波形。在进行捕获之前, 将 Tx EQ 校准为码型发生器输出处每个预设的标称值。
- 对于使用波形后处理工具提供最大眼睛区域的预设, EH 和 EW@E-12 BER 必须在 +/-15% 范围内与信道一致性方法中的目标 EH 和 EW 匹配。

8.4.2.2 Procedure for Testing Rx DUT

8.4.2.2.1 Sj Mask

一旦获得校准的 EH 和 EW，就移动电缆就那个 Rx DUT 连接到校准信道的远端。然后，可以假设 DUT Rx 也将优化其均衡来优化 Tx 均衡。将 Sj 设置为 100 MHz 时的初始值 0.1 UI，并且接收器 CDR 必须实现锁定。在 8.0 GT/s、16.0 GT/s 和 32.0 GT/s 下，去除 100 MHz Sj 初始 tone，然后测试适当的扫描 Sj 曲线。在 16.0 GT/s 和 32.0 GT/s 时，所有测试都会出现 210 MHz 的附加的 Sj tone。这个附加的 tone 的振幅等于达到目标眼宽减去 0.1 UI 所需的 100 MHz Sj 的振幅。如果校准 Sj 电平小于 0.1UI，则不使用 210MHz 的附加 tone。根据数据速率以及被测 Rx 是在 CC 模式还是 IR Refclk 模式下工作，使用不同的 Sj 配置文件。请参阅 Table 8-9。

Figure 8-38, Figure 8-39, Figure 8-40, Figure 8-41 和 Figure 8-42 所示的 SJ 曲线由一个固定的 33 kHz tone 和一个从 400 kHz 至 100 MHz 的扫频 tone 组成，代表了扫频 Sj 频率范围。对于具有 5000 ppm SSC 的 8.0 GT/s 和 16.0 GT/s SRIS 模式，33 kHz 杂散的幅度为 25 ns pp。对于具有 3000 ppm SSC 的 32.0 GT/s SRIS 模式，33 kHz 杂散的幅度为 15 ns pp。接收器必须在整个扫频 Sj 频率范围内满足 10-12 BER。不必在整个 Sj 频率范围内测试接收器，但应测试足够数量的频率点，以确保 Rx 在某个谐振频率下不会使目标 BER 失效。必须测试 33 kHz 频率点。请注意，没有在源应用 SSC。仅在以 8.0 GT/s，16.0 GT/s 和 32.0 GT/s 的速率测试接收器时才需要扫描 Sj。使用单个 33 kHz Sj tone 测试接收器在 2.5 GT/s 和 5.0 GT/s 下的操作。

在 IR 模式下以 8.0 GT/s 的速度运行的接收器使用 Figure 8-38 所示的 Sj 掩模配置文件。33 kHz 杂散的幅度为 25 ns pp，或 200 UIpp。扫描 Sj 曲线的公式如 Figure 8-38 所示。

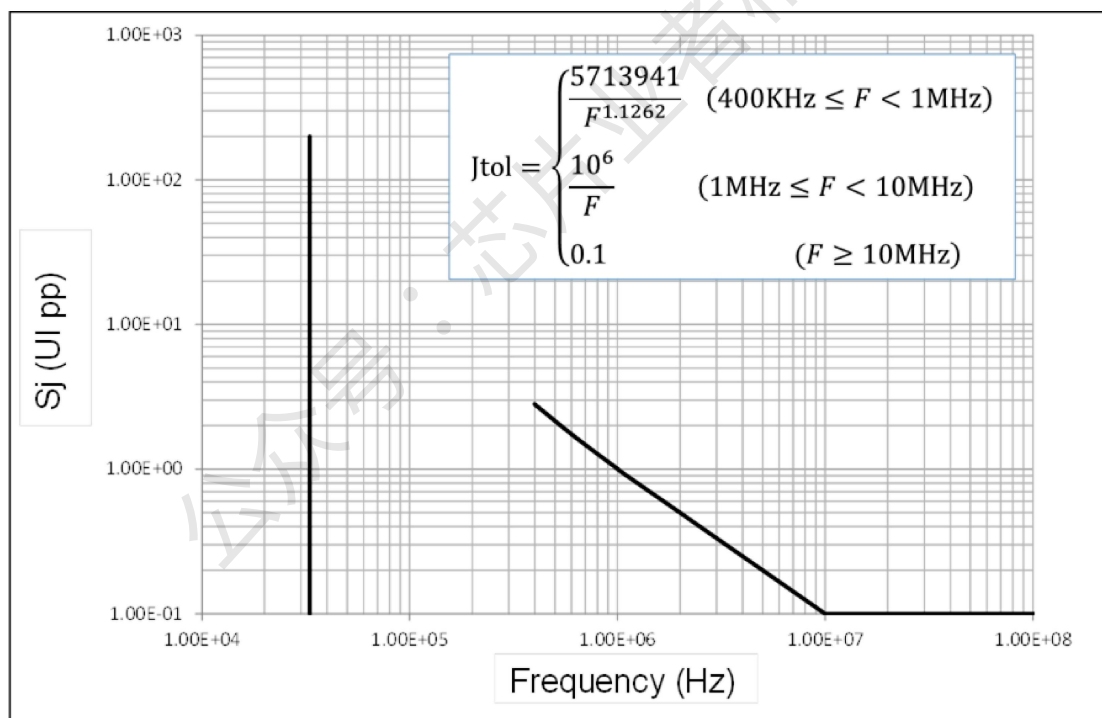


Figure 8-38 Sj Mask for Receivers Operating in IR mode at 8.0 GT/s

在 Independent Refclk (IR) 模式下以 16.0 GT/s 的速度运行的接收器使用 Figure 8-39 所示的 Sj 掩模配置文件。33 kHz 杂散的幅度为 25 ns pp，或 400 UIpp。扫描 Sj 曲线的方程式如 Figure 8-39 所示。

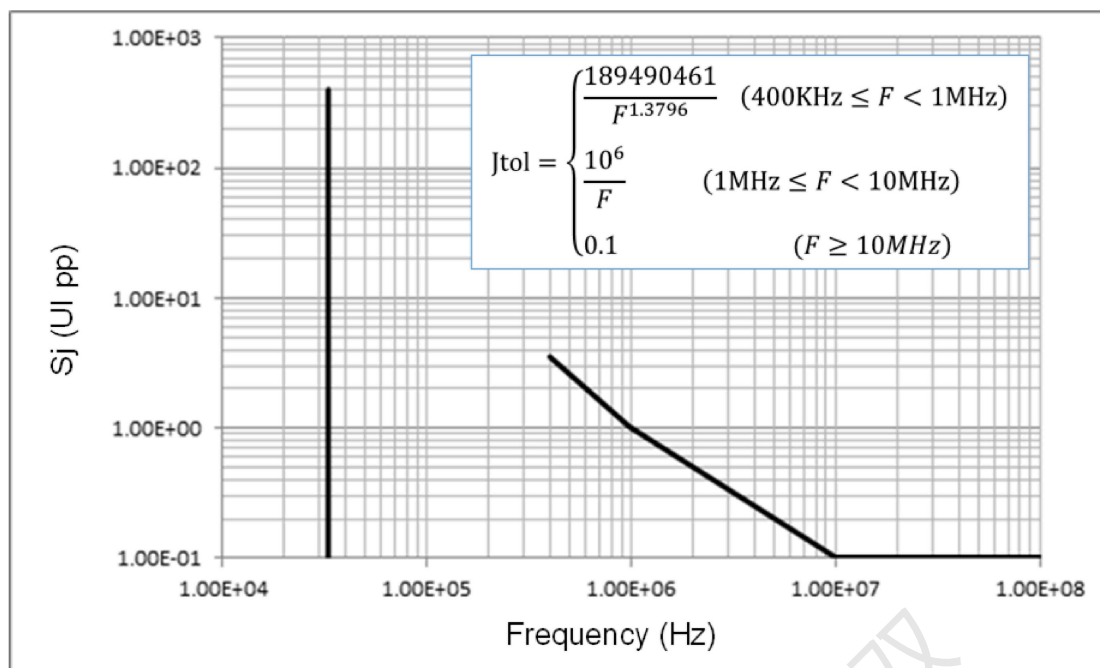


Figure 8-39 Sj Mask for Receivers Operating in SRIS mode at 16.0 GT/s

在 Common Clock (CC) Refclk 模式下以 16.0 GT/s 的速度运行的接收器使用 Figure 8-40 所示的 Sj 掩模配置文件。33 kHz 杂散的幅度为 1 ns pp, 或 16 UIpp。扫描 Sj 曲线的方程式如 Figure 8-40 所示。不支持 32.0 GT/s 的设备可以选择使用 Figure 8-43 中定义的 Sj 掩模。在这种情况下, 必须在源处应用 500 ps-pp 的三角 SSC 调制, 以实现信道校准和 RX 一致性, 如 Table 8-9 所示。

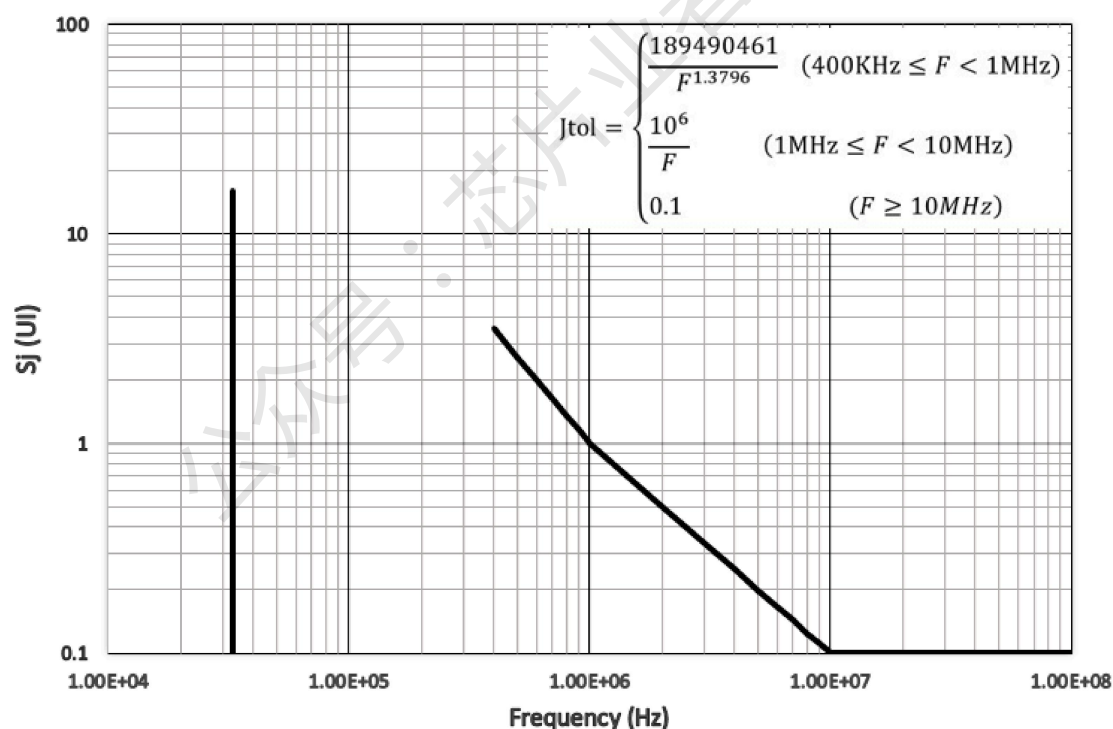


Figure 8-40 Sj Mask for Receivers Operating in CC mode at 16.0 GT/s

在 IR 模式下以 32.0 GT/s 的速度运行的接收器使用 Figure 8-41 所示的 Sj 掩模配置文件。33 kHz 杂散的幅度为 15 ns pp, 或 480 UIpp。扫描 Sj 曲线的方程式如 Figure 8-41 所示。

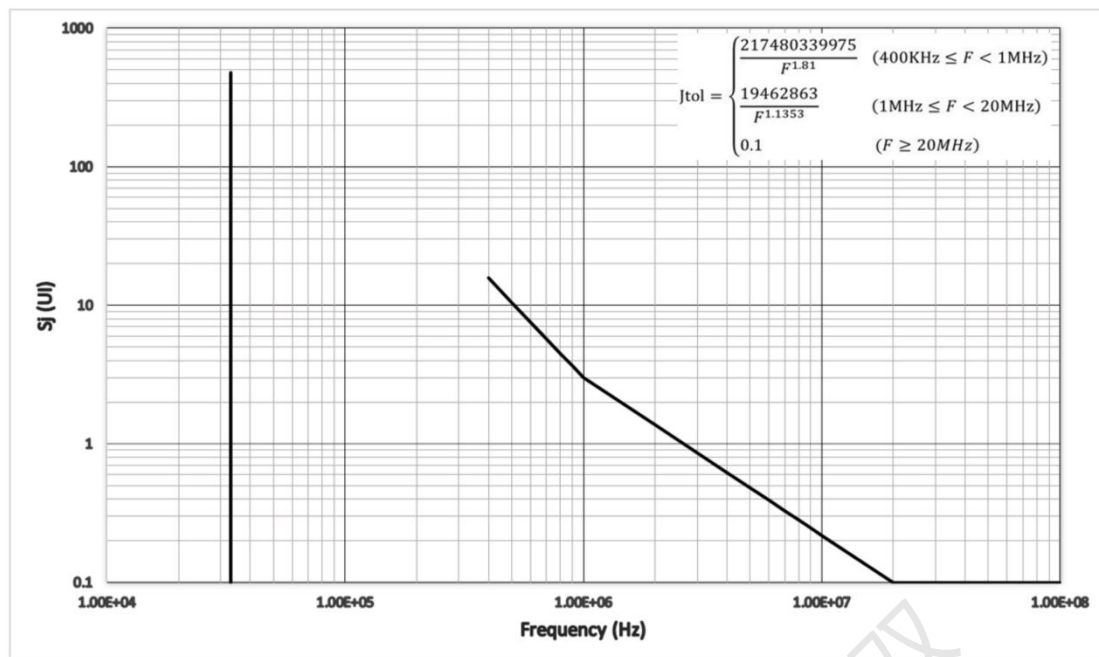


Figure 8-41 Sj Mask for Receivers Operating in SRIS mode at 32.0 GT/s

在 CC Refclk 模式下以 32.0 GT/s 的速度运行的接收器使用 Figure 8-42 所示的 Sj 掩模配置文件。33 kHz 杂散的幅度为 1 ns pp，即 32 UIpp。扫描 Sj 曲线的方程式如 Figure 8-42 所示。

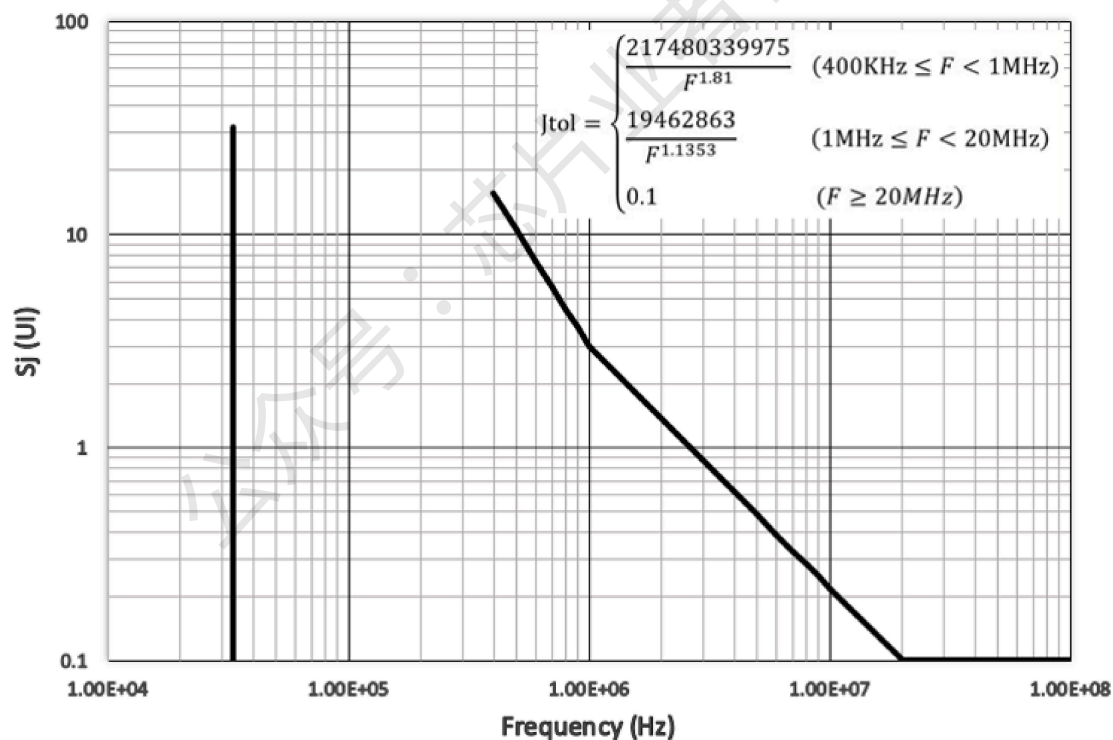


Figure 8-42 Sj Mask for Receivers Operating in CC mode at 32.0 GT/s

在 CC Refclk 模式下以 8.0 GT/s 的速度运行的接收器应使用 Figure 8-43 所示的 Sj 配置文件。除了时钟拓扑不同之外，测试过程与用于 IR 模式的测试过程相同。有关详细信息，请参见 Section 8.4.2.3。在不支持 32.0 GT/s 的设备中，在 CC Refclk 模式下以 16.0 GT/s 的速度运行的接收器还可以选择使用 Figure 8-43 所示的 Sj 掩模配置文件，并按照 Table 8-9 施加额外的剩余 SSC。

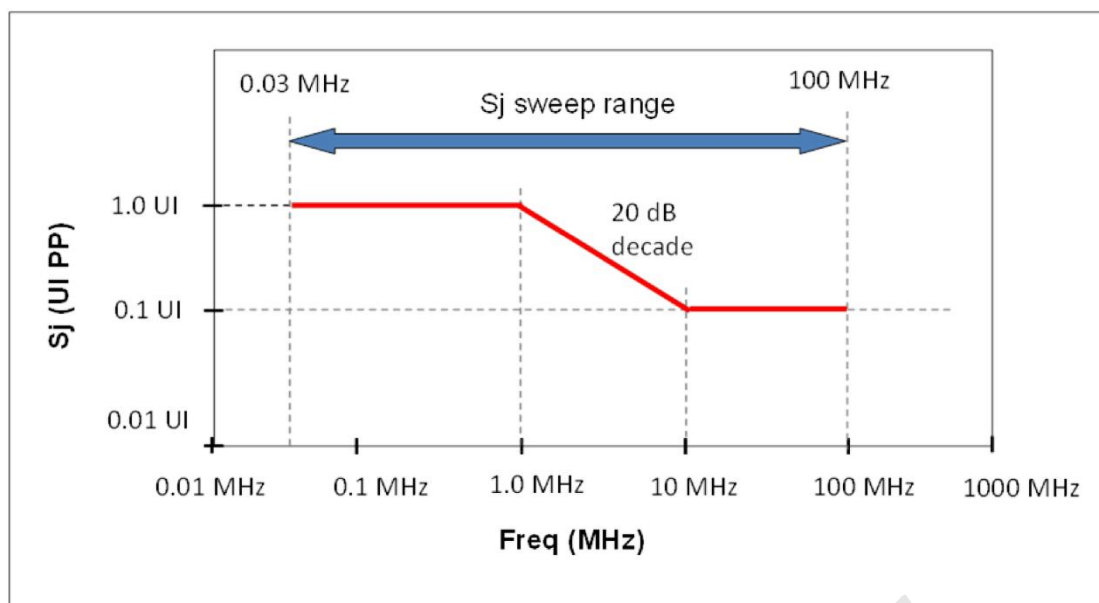


Figure 8-43 Sj Masks for Receivers Operating in CC Mode at 8.0 GT/s

8.4.2.3 Receiver Refclk Modes

允许 Rx 支持以下两种时钟模式中的一种或两种：CC 和 IR，尽管在给定时间只能运行一种时钟模式。接收器可以通过在上电时选择一种模式或通过控制引脚等方式来支持一种以上的 Refclk 模式。

8.4.2.3.1 Common Refclk Mode

Figure 8-44 显示了 Common Clock Refclk 模式下接收器的 Refclk 连接。单个 Refclk 源驱动生成器和 DUT。此测试使用 Section 8.4.2.2.1 中指定的 Sj 掩模。

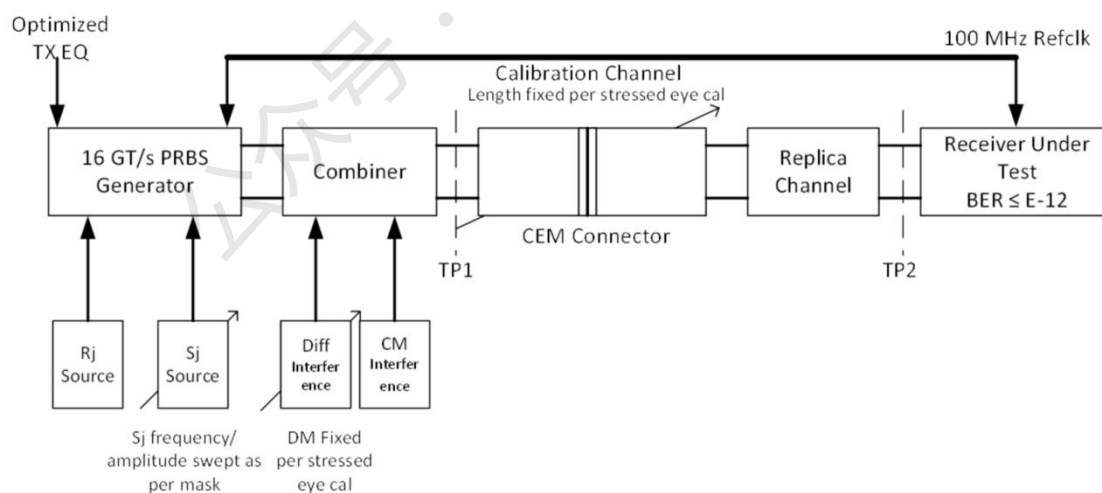


Figure 8-44 Layout for Jitter Testing Common Refclk Rx at 16.0 GT/s

8.4.2.3.2 Independent Refclk Mode

Figure 8-45 说明了在 IR Refclk 模式下测试接收器的配置。DUT 需要带 SSC 的 Refclk 源。该测试使用 Section 8.4.2.2.1 中指定的 Sj 掩模。如 Section 8.4.2.2.1 所示，扫频 Sj 时，生成器必须能够产生 33 KHz 的大 Sj tone。

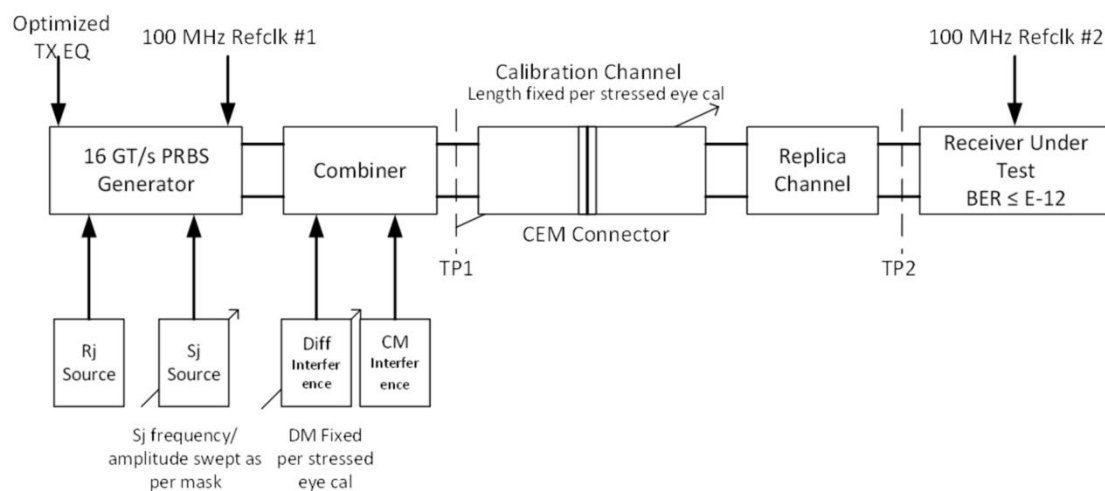


Figure 8-45 Layout for Jitter Testing for Independent Refclk Rx at 16.0 GT/s

8.4.3 Common Receiver Parameters

Table 8-10 列出了与压力眼容忍度不直接相关的常见接收器参数。为四个数据速率分别定义了规范值。

Table 8-10 Common Receiver Parameters

Symbol	Parameter	2.5 GT/s value	5.0 GT/s value	8.0 GT/s value	16.0 GT/s value	32.0 GT/s value	Units	Notes
UI (Rx)	Unit Interval	(min) 399.88 (max) 400.12 (300 PPM)	(min) 199.94 (max) 200.06 (300 PPM)	(min) 124.9625 (max) 125.0375 (300 PPM)	(min) 62.48125 (max) 62.51875 (300 PPM)	(min) 31.246875 (max) 31.253125 (100 PPM)	ps	UI tolerance does not include SSC effects
BW_{RX-PKG-PLL1}	Rx PLL bandwidth corresponding to PKGRX-PLL1	(max) 22 (min) 1.5	(max) 16.0 (min) 8	(max) 4.0 (min) 0.5	(max) 4.0 (min) 0.5	(max) 1.8 (min) 0.5	MHz	Second order PLL transfer bounding function. See Note 1.
BW_{RX-PKG-PLL2}	Rx PLL bandwidth corresponding to PKGRX-PLL2	Not Specified	(max) 16.0 , (min) 5.0	(max) 5.0 (min) 0.5	(max) 5.0 (min) 0.5	N/A	MHz	Second order PLL transfer bounding function. See Note 1.
PKGRX-PLL1	Maximum Rx PLL peaking corresponding to BW _{RX-PKG-PLL1}	(max) 3.0	3.0	2.0	2.0	2.0	dB	Second order PLL transfer bounding function. See Note 1.
PKGRX-PLL2	Maximum Rx PLL peaking corresponding to BW _{RX-PKG-PLL2}	Not specified	1.0	1.0	1.0	N/A	dB	Second order PLL transfer bounding function. See Note 1.
RL_{RX-DIFF}	Differential receiver return loss	See Figure 8-20	See Figure 8-20	See Figure 8-20	See Figure 8-20	See Figure 8-20	dB	Note 2
RL_{RX-CM}	Common mode receiver return loss	See Figure 8-21	See Figure 8-21	See Figure 8-21	See Figure 8-21	See Figure 8-21	dB	Note 2
TRX-GND-FLOAT	Rx termination float time			(max) 500	(max) 500	(max) 500	μs	Note 5
VRX-CM-AC-P	Rx AC common Mode Voltage	(max) 150	(max) 150	(max) 75 for EH < 100 mVPP (max) 125 for EH ≥ 100 mVPP	(max) 75 for EH < 100 mVPP (max) 125 for EH ≥ 100 mVPP	(max) 75 for EH < 100 mVPP (max) 125 for EH ≥ 100 mVPP	mVP	Measured at Rx pins into a pair of 50Ω terminations to ground

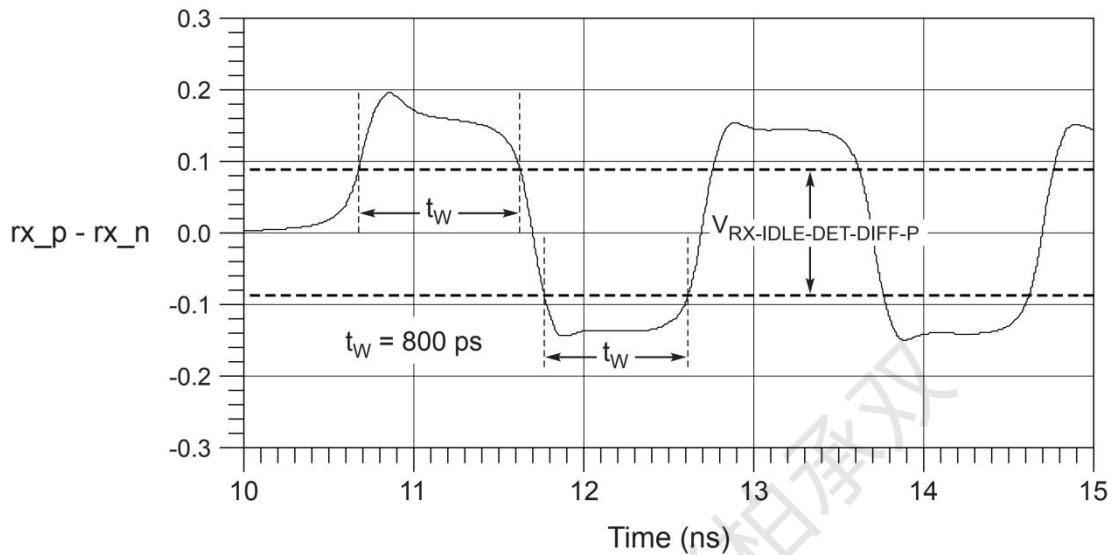
Symbol	Parameter	2.5 GT/s value	5.0 GT/s value	8.0 GT/s value	16.0 GT/s value	32.0 GT/s value	Units	Notes
Z_{RX-DC}	Receiver DC single ended impedance	(min) 40 (max) 60	(min) 40 (max) 60	Not specified	Not specified	Not specified	Ω	DC impedance limits are needed to guarantee Receiver detect. For 8.0, 16.0, and 32.0 GT/s is bounded by <u>RL_{RX-CM}</u> . See Note 3.
Z_{RX-HIGH-IMP-DC-POS}	DC input CM input impedance for V ≥ 0 during Reset or power-down	≥ 10K (0-200 mV) ≥ 20K (> 200 mV)	≥ 10K (0-200 mV) ≥ 20K (> 200 mV)	≥ 10K (0-200 mV) ≥ 20K (> 200 mV)	≥ 10K (0-200 mV) ≥ 20K (> 200 mV)	≥ 10K (0-200 mV) ≥ 20K (> 200 mV)	Ω	Voltage measured wrt. ground. Parameters may not scale with process technology. See Note 4.
Z_{RX-HIGH-IMP-DC-NEG}	DC input CM input impedance for V < 0 during Reset or power-down	1.0K (min)	1.0K (min)	1.0K (min)	1.0K (min)	1.0K (min)	Ω	Voltage measured wrt. ground. Parameters may not scale with process technology. See Note 4.
V_{RX-IDLE-DET-DIFF-PP}	Electrical Idle Detect threshold	(min) 65 (max) 175	(min) 65 (max) 175	(min) 65 (max) 175	(min) 65 (max) 175	(min) 65 (max) 175	mV	V_{RX-IDLE-DET-DIFF-p} = $2 \times V_{RX-D+} - V_{RX-D-} $
T_{RX-IDLE-DET-DIFF-ENTER-TIME}	Unexpected Electrical Idle Enter Detect Threshold Integration Time	(max) 10	(max) 10	(max) 10	(max) 10	(max) 10	ms	An unexpected Electrical Idle (V _{RX-IDLE-DET-DIFF-PP} < V _{RX-IDLE-DET-DIFF-p}) must be recognized no longer than <u>T_{RX-IDLE-DET-DIFF-ENTER-TIME}</u> to signal an unexpected idle condition.
L_{RX-SKEW}	Lane to Lane skew	(max) 20	(max) 8	(max) 6	(max) 5	(max) 5	ns	Across all Lanes on a Port. <u>L_{RX-SKEW}</u> comprehends Lane-Lane variations due to channel and repeater delay differences.

Notes:

- PLL BW 和峰值的两种组合在 5.0GT/s 下指定，以允许设计师在两个参数之间进行权衡。如果 PLL 的最小 BW ≥ 8 MHz，则允许高达 3.0 dB 的峰值。如果 PLL 的最小 BW 放宽到 ≥ 5.0 MHz，则必须满足 1.0 dB 的更严格峰值。注：PLL BW 从零扩展到上表中定义为最小值或最大值的值。对于 2.5GT/s，定义了 1.5-22MHz 和 3.0dB 的单 PLL 带宽和峰值。
- 必须同时测量共模损耗和差分回波损耗。在这两种情况下，DUT 都必须加电并进行直流隔离，并且其 D+/D- 输入必须处于低阻态。
- 首次启用接收器终端时，必须存在 Rx DC 单端阻抗，以确保正确进行接收器检测。允许立即开始对此阻抗的补偿，并且在输入检测时间之前，Rx 共模阻抗（受 RL_{RX-CM} 限制为 50Ω±20%）必须在指定范围内。
- Z_{RX-HIGH-IMP-DC-NEG} 和 Z_{RX-HIGH-IMP-DC-POS} 分别针对接收器输入处的负电压和正电压定义。发送器设计人员在设计接收器检测电路时，需要理解 > 0 和 < 0 Rx 阻抗之间的较大差异。
- 定义接收器的输入板在 2.5/5.0 GT/s 过渡到 8.0、16.0 和 32.0 GT/s 时稳定到新共模的时间。

8.4.3.1 5.0 GT/s Exit From Idle Detect (EFI)

很难用数据速率来扩展 EFI 检测电路的能力，因此 5.0、8.0、16.0 和 32.0 GT/s 规范在 FTS 和 TS1 和 TS2 有序集中定义了不同于为 2.5 GT/s 操作定义的 Data Pattern。特别是，重复的 K28.7 Pattern 被定义为保证足够的电压和时间要求，如下图所示。串联的 EIE 符号产生交替的 1/0 运行长度，每个运行长度为 5 个 UI。



A-0564

Figure 8-46 Exit from Idle Voltage and Time Margins

8.4.3.2 Receiver Return Loss

差模和共模 Rx RL 的测量方法和频率合并与 Tx 相同。有关详细信息，请参见 Figure 8-20 和 Figure 8-21。

8.4.4 Lane Margining at the Receiver - Electrical Requirements

当以 16.0 或 32.0 GT/s 的速度运行时，需要 PCI Express 组件（包括支持 16.0 GT/s 速率的 Retimer）来支持接收器的通道裕度。通道边界允许系统软件在链路处于 L0 状态时获取给定通道的裕度信息。裕度信息包括从当前接收器位置开始的任一方向上的电压和时间。裕度功能不允许要求任何其他外部硬件才能运行。电压的通道裕度支持在 16.0 GT/s 时是可选的，在 32.0 GT/s 时是必需的，向左或向右提供独立的时序裕度是可选的。为简单起见，本规范的协议一章中描述的裕度命令和要求是根据移动数据样本位置的方式进行描述的，但实际的裕度方法是特定于实现的。例如，时序裕度可以在实际数据采样器或独立/误差采样器上实现。此外，可以通过向数据样本位置注入适当量的压力/抖动或通过实际移动数据/错误样本位置来实现时序裕量。Table 8-11 中的参数是针对 16.0 GT/s 和 32.0 GT/s 的报告，并且对于每个速度都可以有所不同。

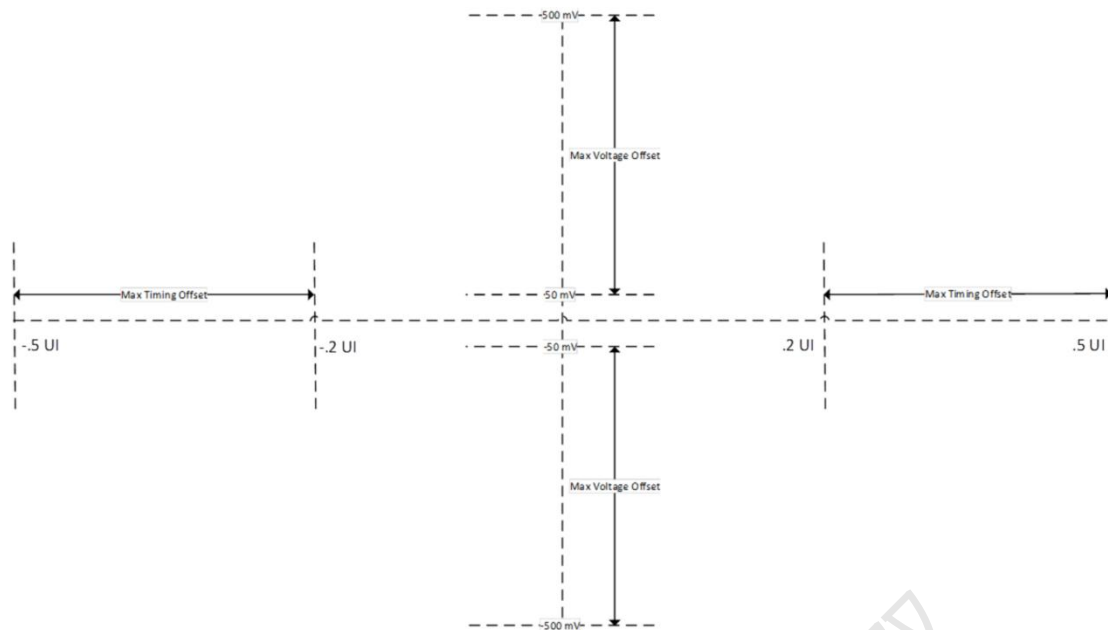


Figure 8-47 Allowed Ranges for Maximum Timing and Voltage Margins

Table 8-11 Lane Margining Timing

Parameter Name	Min	Max	Description
<i>MNumTimingSteps</i>	6	63	Number of time steps from default (to either left or right), range must be at least ± 0.2 UI Timing offset must increase monotonically The number of steps in both positive (toward the end of the unit interval) and negative (toward the beginning of the unit interval) must be identical

Parameter Name	Min	Max	Description
<i>MMaxTimingOffset</i>	20	50	Offset from default at maximum step value as percentage of a nominal UI A 0 value may be reported if the vendor chooses not to report the offset
<i>MNumVoltageSteps</i>	32	127	Number of voltage steps from default (either up or down), minimum range +/-50 mV as measured by the reference equalizer Voltage offset must increase monotonically The number of steps in both positive and negative direction from the default sample location must be identical This value is undefined if <i>MVoltageSupported</i> is 0b
<i>MMaxVoltageOffset</i>	5	50	Offset from default at maximum step value as percentage of one volt A 0 value may be reported if the vendor chooses not to report the offset when <i>MVoltageSupported</i> is 1b This value is undefined if <i>MVoltageSupported</i> is 0b
<i>MSamplingRateVoltage</i>	0	63	The ratio of bits tested to bits received during voltage margining. A value of 0 is a ratio of 1:64 (1 bit of every 64 bits received), and a value of 63 is a ratio of 64:64 (all bits received).
<i>MSamplingRateTiming</i>	0	63	The ratio of bits tested to bits received during timing margining. A value of 0 is a ratio of 1:64 (1 bit of every 64 bits received), and a value of 63 is a ratio of 64:64 (all bits received).
<i>MVoltageSupported</i>	0	1	1b indicates that voltage margining is supported
<i>MIndLeftRightTiming</i>	0	1	1b indicates independent left/right timing margin supported
<i>MIndUpDownVoltage</i>	0	1	1b independent up and down voltage margining supported
<i>MIndErrorSampler</i>	0	1	1b Margining will not produce errors (change in the error rate) in data stream (ie. error sampler is independent) 0b Margining may produce errors in the data stream
<i>MMaxLanes</i>	0	31	Maximum number of Lanes minus 1 that can be margined at the same time. It is recommended that this value be greater than or equal to the number of Lanes in the Link minus 1. Encoding Behavior is undefined if software attempts to margin more than <i>MMaxLanes</i> +1 at the same time. Note: This value is permitted to exceed the number of Lanes in the Link minus 1.
<i>MSampleReportingMethod</i>	0	1	Indicates whether sampling rates (<i>MSamplingRateVoltage</i> and <i>MSamplingRateTiming</i>) are supported (1) or a sample count is supported (0). One of the two methods is supported by each device.
<i>MErrorCount</i>	0	63	If <i>MIndErrorSampler</i> is 1b this is a count of the actual bit errors since margining started. If <i>MIndErrorSampler</i> is 0b this is the actual count of the logical errors since margining started. See the Physical Layer Logical Block chapter for the definition of what errors are counted.

Parameter Name	Min	Max	Description
			The count saturates at 63.
<i>MSampleCount</i>	0	127	Value = $3 \cdot \log_2$ (number of bits margined). Where number of bits margined is a count of the actual number of bits tested during margining. The count stops when margining stops. The count saturates at 127 (after approximately 5.54×10^{12} bits). The count resets to zero when a new margin command is received.

8.4.5 Low Frequency and Miscellaneous Signaling Requirements

8.4.5.1 ESD Standards

必须根据[ESDA-JEDEC-JS-001-2010] (适用于 HBM) 对所有 PCI Express 信号和电源引脚的 ESD 防护等级进行测试, 以达到人体模型 (HBM) 和充电设备模型 (CDM) 标准, 并符合[JEDEC-JESD22-C101] (适用于 CDM)。引脚必须达到或超过[JEDEC-JEP155-JEP157] (HBM/CDM) 或 JEDEC 认可的取代文件中建议的最低水平。

8.4.5.2 Channel AC Coupling Capacitors

PCI Express 链路的每个通道必须交流耦合。Table 8-7 给出了电容的最小值和最大值。电容器只能放置在允许插入和拔出适配器的接口的一侧。外形规格必须定义电容器的所需位置。在一切都位于单个基板上的拓扑中, 电容器可以位于通道的任何位置。之所以使用外部电容器, 是因为所需的值太大而无法在芯片内集成。

8.4.5.3 Short Circuit Requirements

所有发送器和接收器必须支持意外的热插入/拔出操作, 而不会损坏组件。发送器和接收器必须能够承受 D+和 D-的持续接地短路。

8.4.5.4 Transmitter and Receiver Termination

- 每当发送差分信号时, 发送器都必须满足 $R_{LTX-DIFF}$ 和 R_{LTX-CM} (请参见 Figure 8-20 和 Figure 8-21)。
- 仅在不发送差分信号时, 才需要发送器满足 $I_{TX-SHORT}$ (请参见 Table 8-7)。
- 注意: 未定义同一时间的差分阻抗。
- 在所有 LTSSM 状态期间, 接收器都必须满足 $R_{LRX-DIFF}$ 和 R_{LRX-CM} (请参见 Table 8-10), 仅在设备断电, 基本复位有效或明确指定时除外。
- 没有向接收器提供足够功率、基本复位有效或明确规定时, 接收器需要满足 $Z_{RX-HIGH-IMP-DC-NEG}$ 和 $Z_{RX-HIGH-IMP-DC-POS}$ (见 Table 8-7)。

8.4.5.5 Electrical Idle

电气空闲是稳态条件, 其中发送器 D+和 D-电压保持恒定在相同值。电气空闲主要用于节电和非活动状态 (例如, Disabled)。

在发送器进入电气空闲之前, 它必须始终发送所需数量的 EIOS, 但明确免除此要求的 LTSSM 子状态除外。发送完所需数量的 EIOS 的最后一个符号后, 发送器必须在 Table 8-7 中的 $T_{TX-IDLE-SET-TO-IDLE}$ 指定的时间内处于有效的电气空闲状态。

EIOS 的成功接收是根据“Physical Layer Logical Block”一章中定义的规则进行的。应当注意, 在期望多个连续 EIOS 的子状态 (例如, 对于 Loopback Slave 的 Loopback.Active) 中, 接收器必须接收适当数量的 EIOS 序列, 包括 COM, IDL, IDL, IDL。

在电气空闲状态下必须满足低阻抗共模和差分接收器端接值 (请参见 Table 8-7 和 Table 8-10)。电气空闲期间, 发送器可以处于低阻抗模式或高阻抗模式。

发送器每次进入电气空闲状态时, 都必须保持电气空闲状态至少 $T_{TX-IDLE-MIN}$ 。接收器应等待最后的 EIOS, 然后等待电气空闲 ($T_{TX-IDLE-MIN}$) 的最短时间, 以准备其电气空闲退出检测器。

当发送器从电气空闲状态转变为有效的差分信号电平时，它必须满足 Figure 8-20 和 Figure 8-21 中所述的输出回波损耗规范。

如果在接收器上检测到小于 $V_{RX-IDLE-DET-DIFFp-p}$ 最小值的信号，则不会发生电气空闲退出。如果在接收器上检测到大于 $V_{RX-IDLE-DET-DIFFp-p}$ 最大值的信号，则会发生电气空闲退出。可以在接收信号上检测到电气空闲，而不考虑其频率分量，或者仅在接收信号以 125 MHz 或更高的频率切换时才检测到电气空闲。

8.4.5.6 DC Common Mode Voltage

当以 2.5 GT/s 或 5.0 GT/s 的速度工作时，接收器 DC 共模电压标称值为 0V。

除非另有说明，否则发送器 DC 共模电压在所有状态下均保持相同的值。Table 8-10 ($V_{TX-DC-CM}$) 中指定了允许的发送器 DC 共模值的范围。

8.4.5.7 Receiver Detection

对端接收器检测是作为本端发送器实现的一部分的（这是因为我们是通过发送器充放电的时间来检测对端是否连接了设备的），必须正确检测是否存在与 Z_{RX-DC} 参数（40Ω-60Ω）所隐含的直流阻抗等效的负载阻抗。注意：仅在 2.5 GT/s 时才支持 Rx 检测，这就是为什么指定 DC 时接收器阻抗为 2.5 GT/s 的原因。（我们也只需在 2.5GT/s 下检测对端设备是否存在，更高速都需要先经过 2.5GT/s 来建立基本的链路）。

建议的接收器检测过程描述如下：

Step 1：在检测共模偏置之前，发送器必须以稳定的电压启动。

Step 2：发送器改变 D+ 和 D- 上的共模电压，该电压既要符合 $V_{TX-RCV-DETECT}$ 参数，又要符合受 Table 8-10 中参数 $Z_{RX-HIGH-IMP-DC-POS}$ 和 $Z_{RX-HIGH-IMP-DC-NEG}$ 限制的接收器高阻抗的检测的限制。根据链路变为新电压的速率来判定接收器是否被检测到。

- a. 如果发送器上的电压以仅由发送器断电阻抗以及互连和串联电容器的电容决定的速率充电，则接收器不存在。
- b. 如果发送器上的电压以发送器断电阻抗，串联电容器，互连电容和接收器端接所决定的速率充电，则接收器存在。

如果接收器检测电路对差分对的每个导体（D+ 和 D-）执行检测过程，并且检测到任一导体上的负载阻抗均大于 Z_{RX-DC} ，则接收器检测电路应将此解释为不存在终端负载，并且好像没有负载一样进行响应。

要求在差分对的两个导体上执行检测过程。

8.5 Channel Tolerancing

8.5.1 Channel Compliance Testing

规范的这部分内容用于描述系统中发射器设备引脚和接收器设备引脚之间的相关信道的信息。这种类型的系统称为“captive channel”。非 captive channel 的设计应参考适当的外形规格（CEM 是一个示例）规范，因为在这种情况下，外形规格规范优先于此规范，并将信道划分为两种不同类型的组件。

Figure 8-48 和 Figure 8-49 说明了信道容差的关键部件和过程。主要区别在于 Behavioral Tx 和 Rx 均衡的复杂性，这取决于数据速率。2.5GT/s 和 5.0 GT/s 使用固定的 Tx preset 并假定没有 Rx 均衡，而 8.0、16.0 和 32.0 GT/s 假定有多值、可调的 Tx preset 以及 Rx DFE 和 CTLE 的组合。

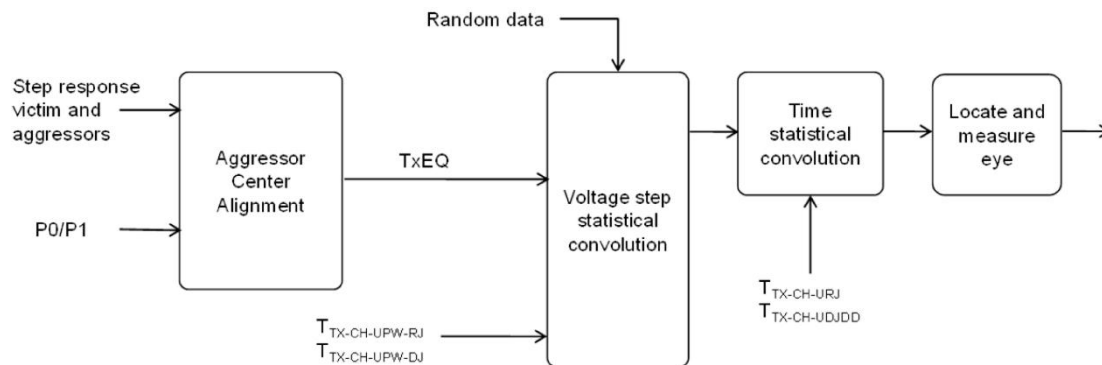


Figure 8-48 Flow Diagram for Channel Tolerancing at 2.5 and 5.0 GT/s

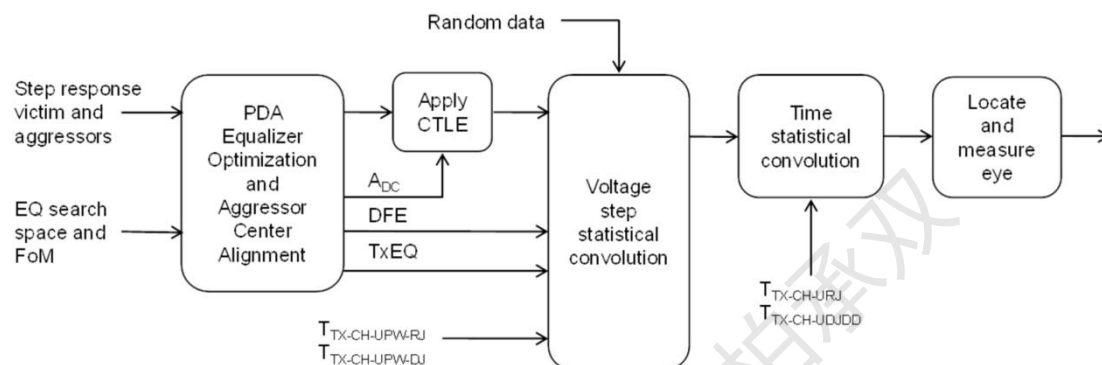


Figure 8-49 Flow Diagram for Channel Tolerancing at 8.0 and 16.0 GT/s

基本的信道一致性方法是通常首先通过 s-parameter 或等效模型来获取信道的特征。然后将 Behavioral Tx 和 Rx 封装模型添加到信道模型，以生成 Die PAD 到 Die PAD 的拓扑。该模型应包括受害路径和足够数量的侵害路径，以准确捕获信道串扰效应。使用发送器规范部分中定义的 Tx 电压和抖动限制，可以将这些参数转换为 Tx 的 Die PAD 上出现的参数。

通过仿真分析生成的模型，在等于接收器输入锁存器的点上产生电压和抖动。在接收器的锁存器上观察到的信号可以作为恢复时钟的参考，并且此数据可以用于构造眼图。

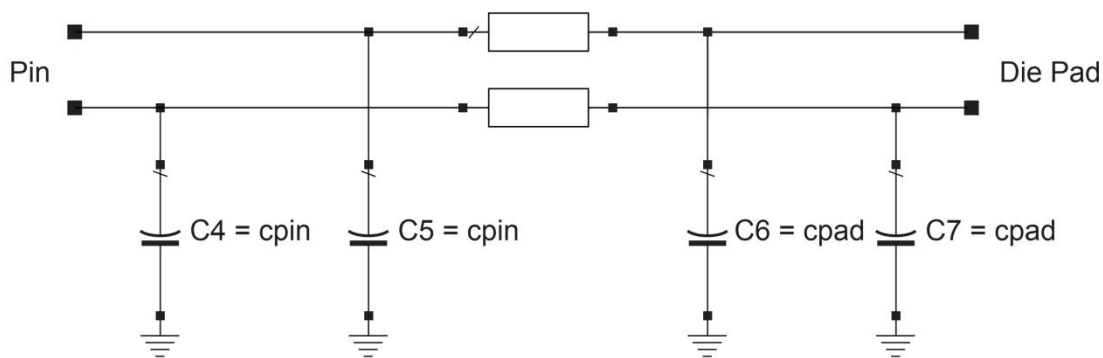
对于 8.0 GT/s、16.0GT/s 和 32.0 GT/s 的测试，仿真过程必须正确考虑 Tx 和 Rx 均衡优化，因为必须由最小能力的 Tx / Rx 对来支持。这意味着仿真过程必须能够根据 1 阶 CTLE 和 1 抽头 DFE（8.0 GT/s）、1 阶 CTLE 和 2 抽头 DFE（16.0 GT/s）以及 2 阶 CTLE 和 3 抽头 DFE（32.0 GT/s）为 Tx 预设或系数和 Rx 均衡设置选择最佳值。

8.5.1.1 Behavioral Transmitter and Receiver Package Models

在本规范中定义了行为封装模型，用来分析裸片到封装之间信道的最大损耗。请注意，在 16.0 GT/s 的速度下，行为封装对许多设备而言都是很高的损失，但不是最坏的情况（see Section 8.3.3.11 和 Section 8.4.1.5）。

规范分别为 8.0、16.0 和 32.0 GT/s 定义了一对封装模型。在 8.0 GT/s 的速度下，为 TX 和 RX 端口定义了单独的封装模型，因为此速率下大多数接收器实现中使用了典型的较小的 C_{PAD} 电容。在 16.0 和 32.0 GT/s 的速度下，为包含 Root 端口的设备和所有其他设备定义了单独的封装模型，因为大多数包含 Root Complex 的设备都需要有大电容和嵌插特性。两种数据速率的信道测试通常都需要在两个方向上测试。

封装模型包含在规范中，作为设计辅助资料。8.0 和 16.0 GT/s 的每个模型都包含 C_{PIN} 和 C_{PAD} 寄生电容以及差分 t-line 元件，如 Figure 8-50 所示。



A-0838A

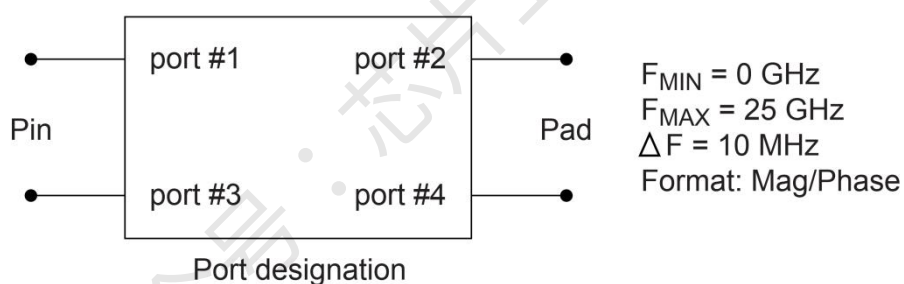
Figure 8-50 Tx/Rx Behavioral Package Models

封装模型生成中使用的 C_{PIN} 和 C_{PAD} 值仅供参考。

Table 8-12 Package Model Capacitance Values

	8.0 GT/s Tx	8.0 GT/s Rx	16.0 GT/s
C_{PIN}	0.25 pF	0.25 pF	0.25 pF
C_{PAD}	1.0 pF	0.8 pF	0.5 pF

为了便于合并到后处理流程中，将 8.0GT/s 和 16.0GT/s 行为封装模型指定为 4-port s-parameter 文件。这些文件是用端口名称、频率范围和粒度指定的，如下所示。s-parameter 的参考阻抗为 50Ω。文件格式为 Touchstone。



A-0839A

Figure 8-51 Behavioral Tx and Rx S-Port Designation for 8.0 and 16.0 GT/s Packages

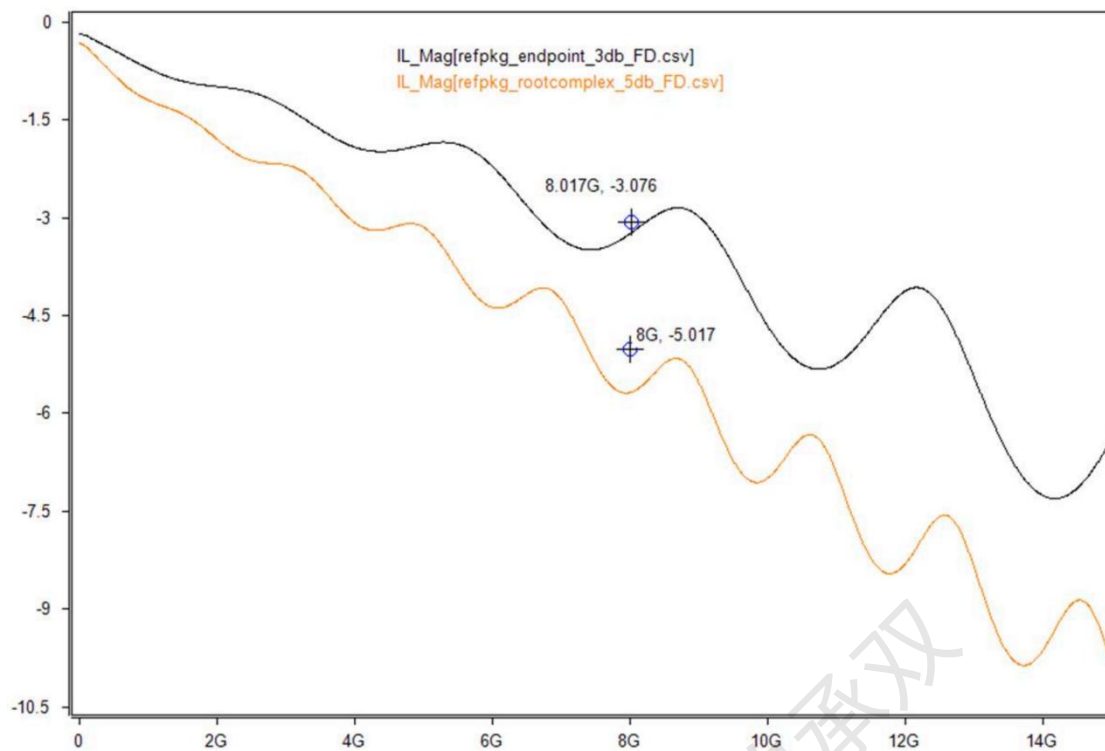


Figure 8-52 SDD21 Plots for Root and Non-Root Packages for 16.0 GT/s

对于 32.0 GT/s, root 封装模型基于实际封装和嵌插模型, non-root 封装模型基于实际的封装和 BGA 模型。

对于所有参考封装模型, 模型均包含芯片电容负载。

Insertion loss

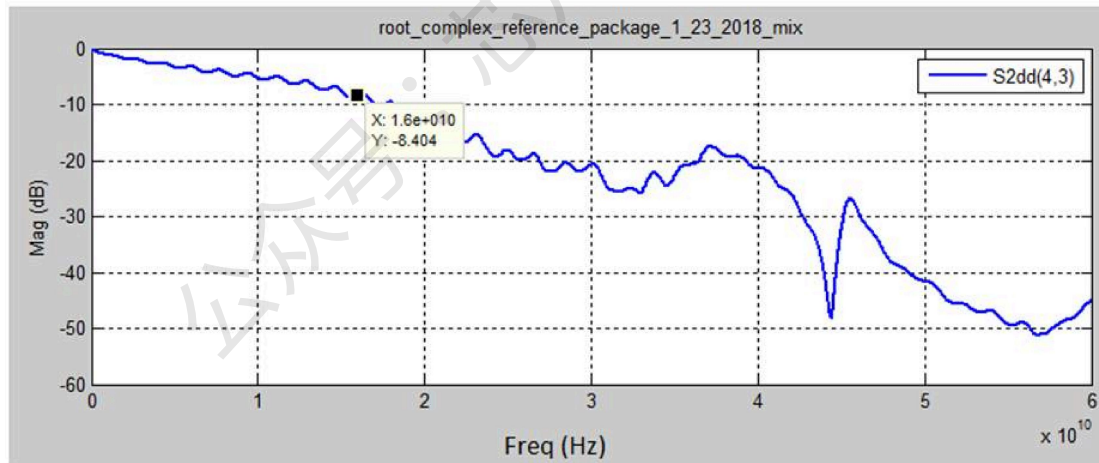


Figure 8-53 Insertion Loss for Root Reference Package for 32.0 GT/s

Return Loss (Board-side)

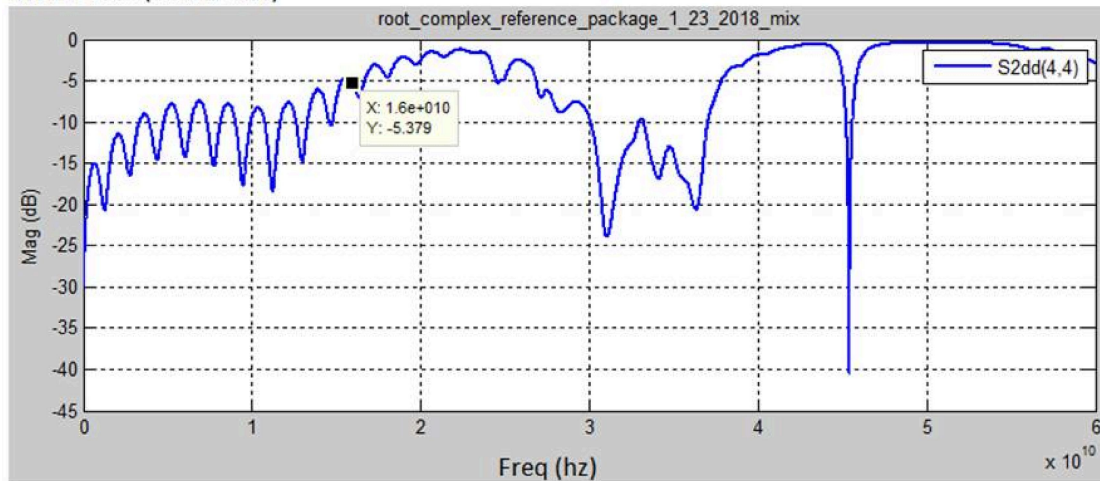


Figure 8-54 Return Loss for Root Reference Package for 32.0 GT/s

NEXT 2

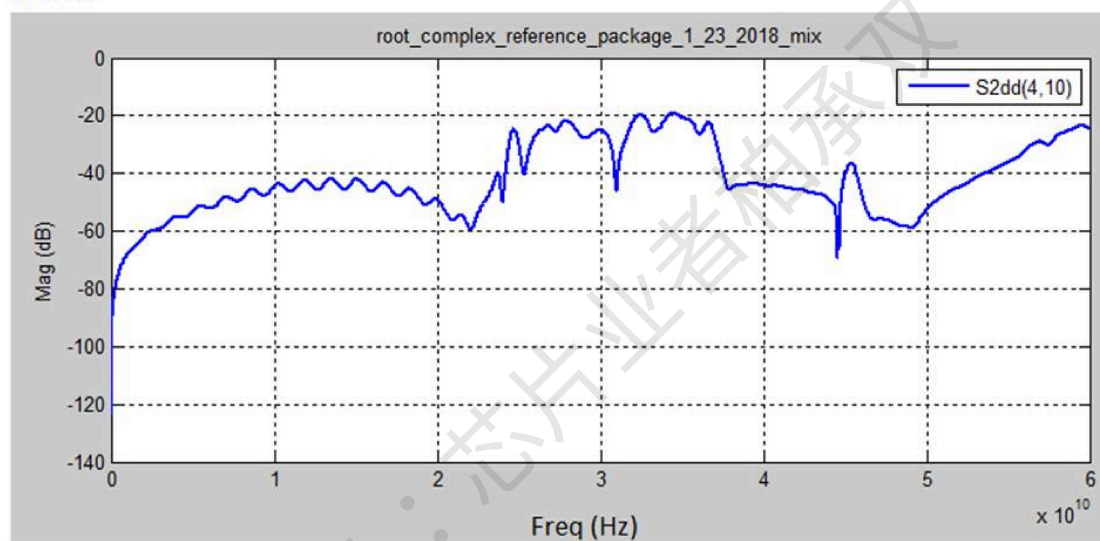


Figure 8-55 NEXT for Root Reference Package (Worst Case) for 32.0 GT/s

FEXT 1

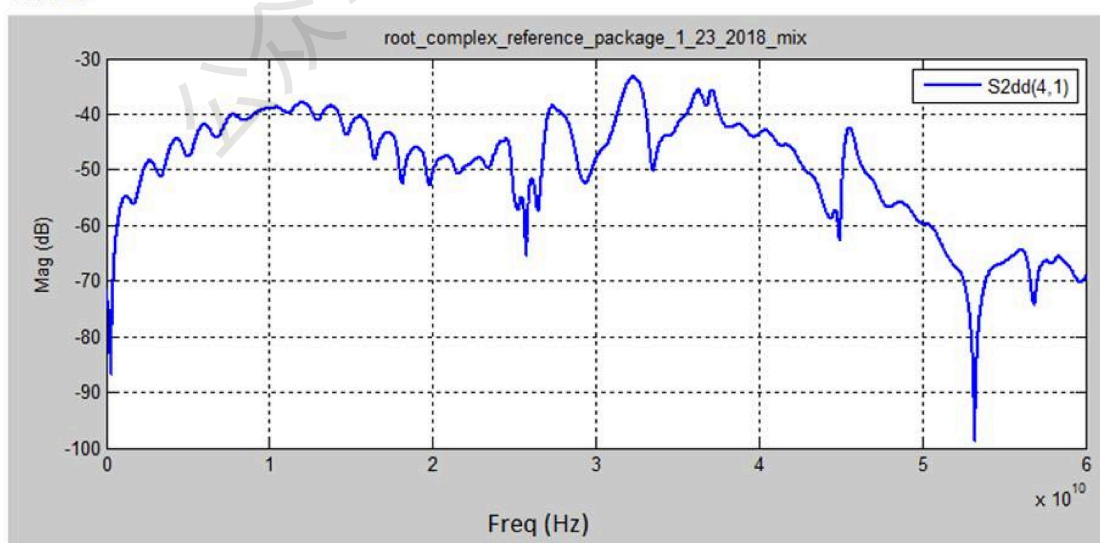


Figure 8-56 FEXT for Root Reference Package (Worst Case) for 32.0 GT/s

Insertion loss

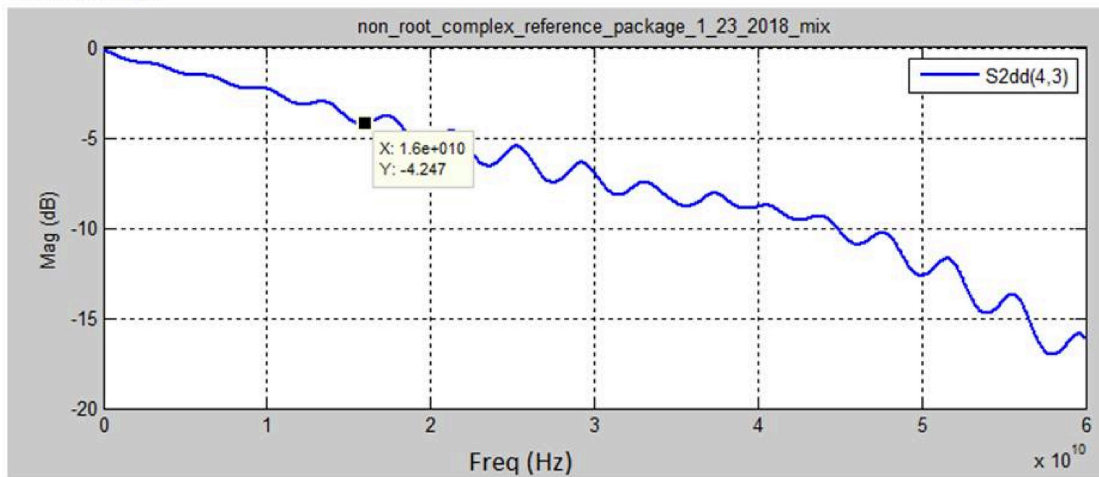


Figure 8-57 Insertion Loss for Non-Root Reference Package for 32.0 GT/s

Return Loss (Board-side)

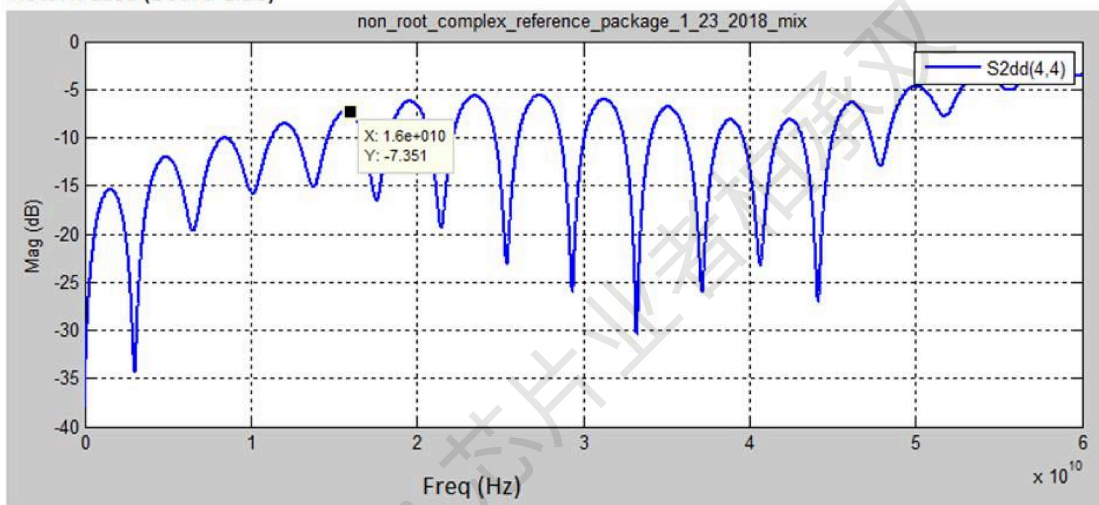


Figure 8-58 Return Loss for Non-Root Reference Package for 32.0 GT/s

NEXT 1

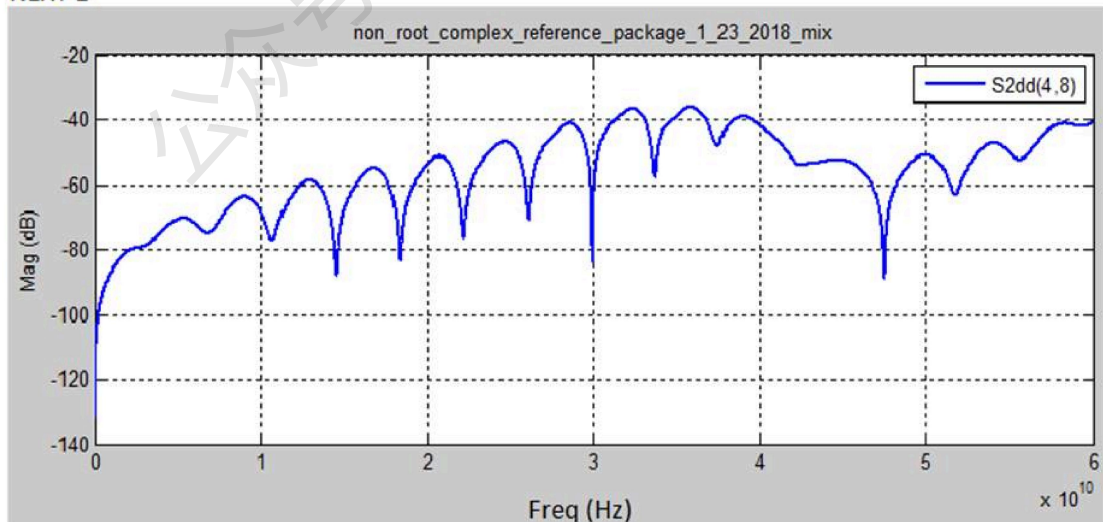


Figure 8-59 NEXT for Non-Root Reference Package (Worst Case) for 32.0 GT/s

FEXT 2 (Die-side)

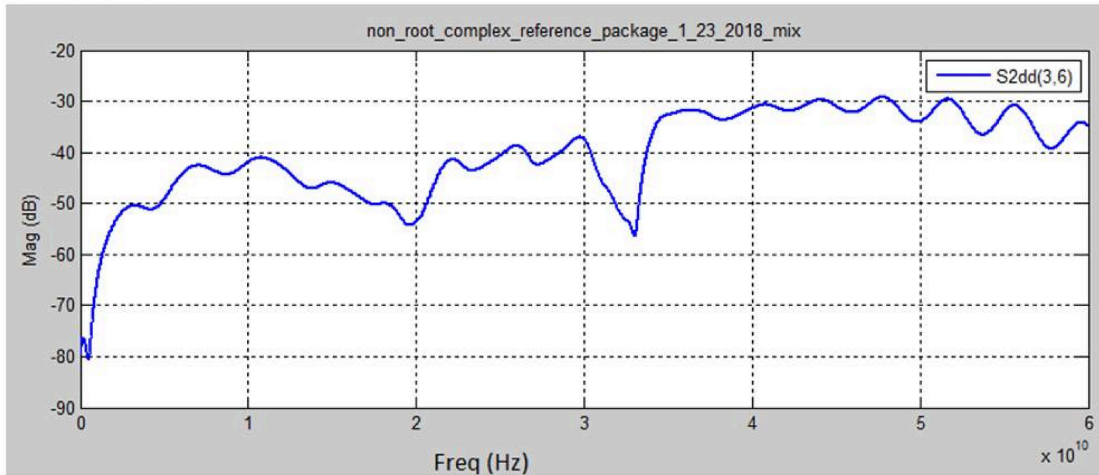
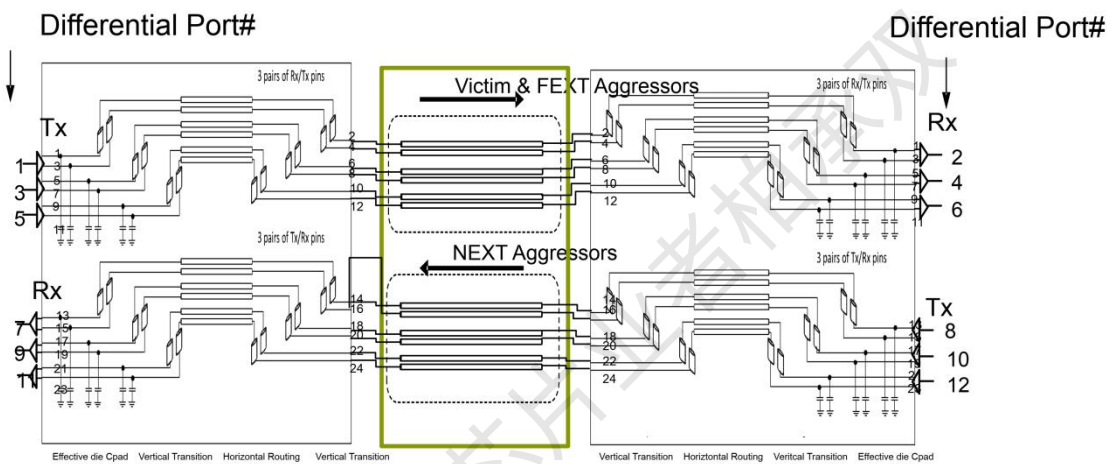


Figure 8-60 FEXT for Non-Root Reference Package (Worst Case) for 32.0 GT/s



Reference Package Channel (Pin to Pin) Reference Package

Figure 8-61 32.0 GT/s Reference Package Port Connections for Pin to Pin Channel Evaluation

Figure 8-61 给出了使用 32.0 GT/s 参考套件使用信道一致性方法评估 pin-to-pin 信道的端口连接。必须在两个方向上都做评估（分别对应 root 发送和 non-root 发送）。标记为 channel 3 到 channel 4 的信道旨在作为最坏情况的受害信道，必须在两个方向上进行评估。参考封装模型中的其他信道仅打算用作串扰干扰信道。

8.5.1.2 Measuring Package Performance (16.0 GT/s only)

16.0 GT/s 的封装插入损耗是一个有用的规范参数。某些以 16.0 GT/s（Section 8.3.3.11）的实现允许其封装在插入损耗和/或串扰方面超过参考封装。必须通过执行 PCI-SIG 网站上规范所提供的参考信道的信道一致性来评估实际的封装性能。在参考信道上运行一组信道一致性仿真，其中一个参考封装被正在评估的封装替换。如果要评估的封装的任何信道的眼高或眼宽都较小，则认为该封装的性能比参考封装差。具有比参考封装性能更差的封装的实现必须使用信道一致性方法中的实现封装模型，并且可以选择在接收器强调眼图校准中使用该实现封装。请注意，无论封装特性如何，仍然需要满足外形规格和通信道一致性（对于 captive channel）的总体要求。

8.5.1.3 Simulation Tool Requirements

信道容限是通过仿真实现的，其中 pass/fail 的判定是根据时域眼图定义的。仿真工具必须接受一组规定的输入，包括信道，然

后基于一组后处理需求进行仿真。基本规范没有规定使用任何特定的工具来仿真信道。但是，任何仿真工具都必须满足以下章节描述的要求。

8.5.1.3.1 Simulation Tool Chain Inputs

- **Channel characteristics defined as s-parameters or equivalent model:** 该模型必须包括受害者差分信道以及为准确捕获串扰所需的任意数量的侵害者。在大多数情况下，这需要 2 至 4 个额外的差分通道。请注意，32.0 GT/s 下可能需要几个的侵扰者来准确捕获最坏情况的串扰，可能需要几个 NEXT 侵扰者来准确捕获串扰。
- **Behavioral Root and Non-Root package models:** 这些模型将以 s-parameter 文件的形式作为规范的一部分包括在内（Section 8.5.1.1）。
- **Transmitter Jitter and voltage:** 输入到仿真器的电压和抖动参数可以直接从发射机和 Refclk 抖动的组合中获得。由于这些参数是固定的，因此仿真工具可以选择对它们的值进行硬编码。
- **Transmitter and Receiver Termination Impedance:** 仿真器应为发射器和接收器使用 $2 \times 50\Omega$ 终端电阻。该值与在生成和测量 Rx 容忍度的压力眼中所隐含的假设相符。

8.5.1.3.2 Processing Steps

- **Time domain representation of the end-to-end connectivity:** 包括 Tx 和 Rx 行为封装模型以及被测信道。
- **Tx voltage and jitter:** 电压和抖动在 Table 4-27 列出。它们来自为发射机定义的电压和抖动参数，但经过重新计算以正确区分高频和低频抖动分量，并且还包括 Refclk 抖动的信息。
- **Behavioral TxEq:** 仿真器应直接把发射机定义的 TxEq 功能拷贝过来做仿真。
- **Behavioral Rx CTLE:** 仿真工具应实现一个 CTLE 行为模型，用于模拟 Rx 的 CTLE 功能，并用以分析 Rx 容限。
- **Behavioral DFE:** 仿真工具应实现 1 抽头（8.0 GT/s），2 抽头（16.0 GT/s）或 3 抽头（32.0 GT/s）DFE，其中反馈系数的动态范围在 Section 8.4.1.10 定义。
- **Optimizing Tx equalization and Rx DFE/CTLE settings:** 仿真工具应实现一种优化算法，该算法选择 Tx 均衡和 Rx CTLE 和 DFE 设置的组合，以在通道的远端产生最大眼宽。有关详细信息，请参见第 8.4.1.8 节。
- **Statistical Treatment of jitter:** 为了避免过高估计信道数据和信道抖动相互作用的影响，该工具应使用这些参数的统计分析来生成电压/抖动眼图裕量。

8.5.1.3.3 Simulation Tool Outputs

输出眼图参数：仿真器应生成统计定义的输出，该输出显示眼宽（EW）和眼高（EH）。EH 将作为数据样本位置的峰值眼高测量，而 EW 将在零交叉线处测量。此外，仿真器应具有从 UI 的平均中心开始以 ± 0.1 UI 的方式将数据采样点调整为 8.0 和 16.0 GT/s 的能力，如 Figure 8-63 所示。对于 32.0 GT/s，仿真器应以 0.05 UI 的增量将采样点调整到 UI 样本位置的平均中心左侧的 0.30 UI 处，计算每个采样位置的 DFE 系数，并选择产生眼睛高度（在数据采样点）乘以眼睛宽度的最大值的結果。

8.5.1.3.4 Open Source Simulation Tool

规范应提供开源仿真工具作为设计辅助。该工具将提供一站式功能，用户可以在接收器的芯片 PAD 上提供信道特性作为阶跃响应，并且该工具可以计算也统计眼图，该眼图能显示 pass/fail 信息。

8.5.1.4 Behavioral Transmitter Parameters

8.5.1.4.1 Deriving Voltage and Jitter Parameters

本节仅用于提供信息。电压和抖动参数可以从发送器电压和抖动参数中得出，但以 Die PAD 为参考。这是必需的，因为信道仿真包括行为 Tx 封装并从 Die PAD 驱动封装。此外，必须将 Tj 项分解为单独的 Rj 和 DjDD 项。

- $V_{TX-CH-FS-NO-EQ}$ 和 $V_{TX-CH-RS-NO-EQ}$ ：这两个参数定义了与 Figure 8-5 中的 Vd 相对应的最小峰峰值电压。
- 抖动参数是基于以下方程组得出的。代数运算用于提取由 Tj 和 DjDD 项的组合隐式定义的 Rj。以下数字基于 8.0 GT/s Tx 抖动参数，该参数基于 Table 8-6 中指定的值。使用相同的方法提取 2.5、5.0、16.0 和 32.0 GT/s 的抖动参数，其中假定 PCIe 5.0 设备支持的最大数据速率为 32.0 GT/s，并且 Table 8-6 中的抖动值用于提取过程。

$$\begin{aligned}jit_hfrj_nui &= (T_{TX-UTJ} - T_{TX-UDJ-DD})/14.06 = 1.11ps \\T_{TX-CH-UPW-RJ} &= (T_{TX-UPWJ-TJ} - T_{TXUPWJ-DJDD})/14.06 = 1.00ps \\T_{TX-CH-UPW-DJ} &= T_{TXUPWJ-DJDD} = 10.0ps \\T_{TX-CH-URJ} &= \sqrt{jit_hfrj_nui^{**2} - (T_{TX-CH-UPW-RJ}*0.707)^{**2} + T_{REFCLK-RMS}^{**2}} = 1.31ps \\T_{TX-CH-UDJDD} &= T_{TX-UDJ-DD} - (T_{TXUPWJ-DJDD})/2 = 7.00ps\end{aligned}$$

A-0840A

Figure 8-62 Example Derivation of 8.0 GT/s Jitter Parameters for Table 8-13

必须以正常操作期间要支持的所有数据速率以及相应的 Tx 抖动参数对信道进行测试。例如，旨在支持最大数据速率为 8.0 GT/s 的信道必须在 2.5、5.0 和 8.0 GT/s 下进行测试。

Table 8-13 Jitter/Voltage Parameters for Channel Tolerancing

Symbol	Parameter	Value	Units	Notes
$V_{TX-CH-FS-NO-EQ}$	Full swing Tx voltage	804	mVPP	Full swing, No Tx Eq.
$V_{TX-CH-RS-NO-EQ}$	Reduced swing Tx voltage	402	mVPP	Reduced swing, No Tx Eq.
2.5 GT/s Jitter Parameters and Voltage Parameters				
$T_{TX-CH-URJ-2.5G}$	Tx uncorrelated Rj	3.45	ps RMS	See Note 1
Symbol	Parameter	Value	Units	Notes
$T_{TX-CH-UDJDD-2.5G}$	Tx uncorrelated DjDD	20	ps PP	
$T_{TX-CH-UPW-RJ-2.5G}$	Uncorrelated PW Rj	1.42	ps RMS	See Note 2
$T_{TX-CH-UPW-DJ-2.5G}$	PW DDj	80	ps PP	
$T_{TX-DIEPAD-EDGERATE-2.5G}$	Signal edge rate at behavioral Tx die pad	140	ps	Measured 10% to 90% using a gaussian lowpass filter to shape the edge. See Note 3.
5.0 GT/s Jitter Parameters and Voltage Parameters				
$T_{TX-CH-URJ-5G}$	Tx uncorrelated Rj	3.45	ps RMS	See Note 1
$T_{TX-CH-UDJDD-5G}$	Tx uncorrelated DjDD	20-	ps PP	
$T_{TX-CH-UPW-RJ-5G}$	Uncorrelated PW Rj	1.42	ps RMS	
$T_{TX-CH-UPW-DJ-5G}$	PW DDj	40	ps PP	See Note 2.
$T_{TX-DIEPAD-EDGERATE-5G}$	Signal edge rate at behavioral Tx die pad	70	ps	Measured 10% to 90% using a gaussian lowpass filter to shape the edge. See Note 3.

8.0 GT/s Jitter Parameters and Voltage Parameters

$T_{TX-CH-URJ-8G}$	Tx uncorrelated Rj	1.31	ps RMS	No DDj of HF jitter. See Note 1.
$T_{TX-CH-UDJDD-8G}$	Tx uncorrelated DjDD	7.0	ps PP	No DDj of HF jitter
$T_{TX-CH-UPW-RJ-8G}$	Uncorrelated PW Rj	1.0	ps RMS	
$T_{TX-CH-UPW-DJ-8G}$	PW DDj	10	ps PP	See Note 2.
$T_{TX-DIEPAD-EDGERATE-8G}$	Signal edge rate at behavioral Tx die pad	43.75	ps	Measured 10% to 90% using a gaussian lowpass filter to shape the edge. See Note 3.

16.0 GT/s Jitter Parameters and Voltage Parameters

$T_{TX-CH-URJ-16G}$	Tx uncorrelated Rj	0.71	ps RMS	See Note 1.
$T_{TX-CH-UDJDD-16G}$	Tx uncorrelated DjDD	3.75	ps PP	
$T_{TX-CH-UPW-RJ-16G}$	Uncorrelated PW Rj	0.54	ps RMS	
$T_{TX-CH-UPW-DJ-16G}$	PW DDj	5.0	ps PP	See Note 2.
$T_{TX-DIEPAD-EDGERATE-16G}$	Signal edge rate at behavioral Tx die pad	21.875	ps	Measured 10% to 90% using a gaussian lowpass filter to shape the edge. See Note 3.

32.0 GT/s Jitter Parameters and Voltage Parameters

$T_{TX-CH-URJ-32G}$	Tx uncorrelated Rj	0.276	ps RMS	See Note 1.
$T_{TX-CH-UDJDD-32G}$	Tx uncorrelated DjDD	1.875	ps PP	
$T_{TX-CH-UPW-RJ-32G}$	Uncorrelated PW Rj	0.27	ps RMS	
$T_{TX-CH-UPW-DJ-32G}$	PW DDj	2.5	ps PP	See Note 2.

Symbol	Parameter	Value	Units	Notes
$T_{TX-DIEPAD-EDGERATE-32G}$	Signal edge rate at behavioral Tx die pad	10.94	ps	Measured 10% to 90% using a gaussian lowpass filter to shape the edge. See Note 3.

Notes:

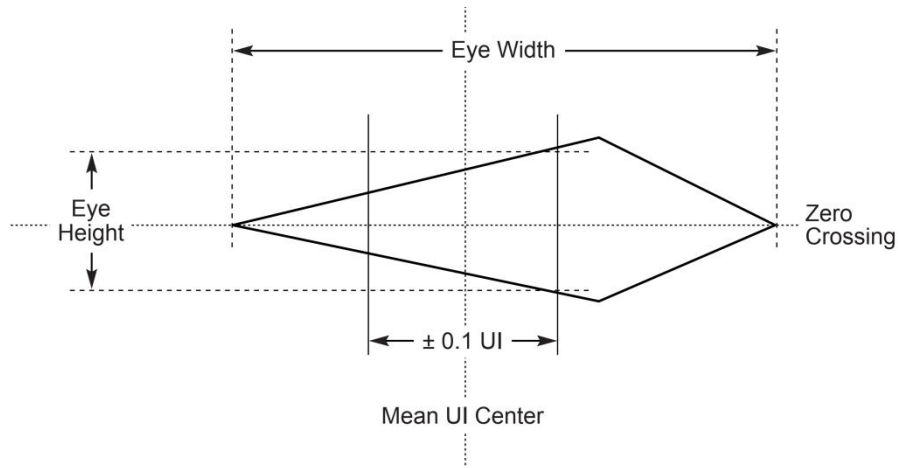
1. Includes low frequency (non F/2) Rj components from the Transmitter and Rj from the Refclk.
2. Applied on a per edge basis as a dual Dirac model.
3. Does not include parasitic die pad capacitance. See Figure 8-50 for details of behavioral package.

8.5.1.4.2 Optimizing Tx/Rx Equalization (8.0 GT/s, 16.0 GT/s and 32.0 GT/s only)

行为接收机选择 Transmitter Equalization、CTLE、DFE 和样本位置（仅 32.0 GT/s）的组合产生最佳的眼睛区域（眼睛宽度乘以眼睛高度）。

8.5.1.4.3 Pass/Fail Eye Characteristics

仿真工具的输出应采用 pass/fail 特性的形式，该特性由 eye mask 定义，如 Figure 8-63 所示。EH 和 EW 必须分别满足 Table 8-14 中定义的电压和抖动参数。在应用了适当的 Tx 和 Rx 均衡算法之后，在接收器的 Die PAD 处定义了眼图裕度。在信道被设计用于特定的一对硅器件并且这些硅器件的封装模型（包括串扰侵略者）是已知的情况下，在运行 pad-to-pad 盘信道 pass/fail 一致性仿真时，可以使用实际的器件封装而不是参考封装。



A-0841A

Figure 8-63 EH, EW Mask

注意, Figure 8-63 所示的 pass/fail EH 和 EW 限值与 Table 8-9 中针对 8.0 和 16.0 GT/s 的 Rx 测试所定义的限值相同。对于 32.0GT/s, pass/fail EH 和 EW 限制仍然相同, 但它们是在最佳样本位置计算的。对于 2.5 和 5.0 GT/s, 在应用与 8.0 GT/s 信道容差相同的参考封装后, Rx 测试的限制以封装引脚为参考, 并且该表中通道容差的极限以接收器 PAD 为参考。

Table 8-14 Channel Tolerancing Eye Mask Values

Symbol	Parameter	Value	Units	Comments
2.5 GT/s Eye Margins				
$V_{RX-CH-EH-2.5G}$	Eye height	<130 (min)	mVPP	Eye height at BER= 10^{-12} . Note 1
$T_{RX-CH-EW-2.5G}$	Eye width at zero crossing	<0.35 (min)	UI	Eye width at BER= 10^{-12}
$T_{RX-DS-OFFSET-2.5G}$	Peak EH offset from UI center	± 0.1	UI	See Figure 8-63 for details.
5.0 GT/s Eye Margins				
$V_{RX-CH-EH-5G}$	Eye height	< 85 (min)	mVPP	Eye height at BER= 10^{-12} . Note 1
$T_{RX-CH-EW-5G}$	Eye width at zero crossing	<0.30 (min)	UI	Eye width at BER= 10^{-12}
$T_{RX-DS-OFFSET-5G}$	Peak EH offset from UI center	± 0.1	UI	See Figure 8-63 for details.
8.0 GT/s Eye Margins				
$V_{RX-CH-EH-8G}$	Eye height	25 (min)	mVPP	Eye height at BER= 10^{-12} . Note 1.
$T_{RX-CH-EW-8G}$	Eye width at zero crossing	0.3 (min)	UI	Eye width at BER= 10^{-12}
$T_{RX-DS-OFFSET-8G}$	Peak EH offset from UI center	± 0.1	UI	See Figure 8-63 for details.
$V_{RX-DFE-D1-8G}$	Range for DFE d ₁ coefficient	± 30	mV	

16.0 GT/s Eye Margins

$V_{RX-CH-EH-16G}$	Eye height	15 (min)	mVPP	Eye height at BER= 10^{-12} . Note 1.
$T_{RX-CH-EW-16G}$	Eye width at zero crossing	0.3 (min)	UI	Eye width at BER= 10^{-12}
$T_{RX-DS-OFFSET-16G}$	Peak EH offset from UI center	± 0.1	UI	See Figure 8-63 for details.
$V_{RX-DFE-D1-16G}$	Range for DFE d ₁ coefficient	± 30	mV	
$V_{RX-DFE-D2-16G}$	Range for DFE d ₂ coefficient	± 20	mV	

32.0 GT/s Eye Margins

Symbol	Parameter	Value	Units	Comments
$V_{RX-CH-EH-32G}$	Eye height	15 (min)	mVPP	Eye height at BER="10" ⁻¹² . Note 1.
$T_{RX-CH-EW-32G}$	Eye width at zero crossing	0.3 (min)	UI	Eye width at BER="10" ⁻¹²
$T_{RX-DS-OFFSET-32G}$	Peak EH offset from UI center	N/A	UI	See Figure 8-63 for details.
$T_{RX-SAMPLE-OFFSET-32G}$	Max sample location offset to the left from UI center	0.30	UI	Note 2
$T_{RX-SAMPLE-GRANULARITY-32G}$	Granularity for sample location offset	0.05	UI	Note 2
$V_{RX-DFE-D1-32G}$	Range for DFE d ₁ coefficient	Abs(D1)/D0 (cursor amplitude) ≤ 0.8		
$V_{RX-DFE-D2-32G}$	Range for DFE d ₂ coefficient	± 20	mV	
$V_{RX-DFE-D3-32G}$	Range for DFE d ₃ coefficient	± 20	mV	

Notes:

1. $V_{RX-CH-EH}$ is defined as max EH within an aperture of ± 0.1 UI from mean UI center.
2. The optimal eye area is computed at each offset from mean UI center -0.30 UI to mean UI center with the specified granularity.

8.5.1.4.4 Characterizing Channel Common Mode Noise

信道必须满足在接收器上定义的共模参数要求。通常，由于存在大量会产生共模噪声的因素，这其中也包括发送器，因此无法精确仿真所有的共模噪声贡献。典型情况下，信道共模噪声是估算参数，下面给出一个简单的估算过程。信道的 CM 限制定义为在最坏 Tx CM 的情况下，一个信道可以添加并仍然满足 Rx CM 限制的 CM 噪声量。对于 EH < 100 mV，此限制为 75 mVPP；对于 EH ≥ 100 mVPP，此限制为 125 mVPP。

注意，Tx 和信道 CM 噪声参数不能简单的叠加以获取 Rx CM 限制。这是基于这样一个事实，即一个信道高频 Tx CM 噪声衰减大，Tx LF CM 噪声衰减小。该信道自身还可以贡献出一些高频和低频的 CM 分量。

8.5.1.4.5 Verifying $V_{CH-IDLE-DET-DIFF-PP}$

$V_{CH-IDLE-DET-DIFF-PP}$ 被定义为当发送器发出 EIEOS 序列时接收器能够检测到它的电压规范。在 8.0，16.0 和 32.0 GT/s 的情况下，潜在的更大 TxEq 升压比需要验证该参数。对于 2.5 或 5.0 GT/s，最大 TxEq 提升较小的情况下，无需执行此过程。定义 Tx / Rx 芯片 PAD 上的发射电压和检测电压可以使 $V_{CH-IDLE-DET-DIFF-PP}$ 与用于确定眼图裕量的信道模型相同。从 Tx 引脚到 Rx 引脚进行仿真（不包括 Tx

和 Rx 行为封装模型) 也是可以接受的, 在这种情况下, 适用于 Tx 和 Rx 部分中定义的 EIEOS 和空闲检测参数。

对于长信道, 使用 $V_{TX-EIEOS-FS}$ 来描述, 它表示信道测试在 EIEOS pattern、-11.0 dB 去加重和 0 dB preshoot 下的测试参数限制。对于短信道, 使用 $V_{TX-EIEOS-RS}$ 来描述, 它表示在-4.5dB 去加重和 0 dB preshoot 下的测试参数限制。

Table 8-15 EIEOS Signaling Parameters

Parameter	Description	Value	Units	Comments
$V_{CH-IDLE-EXIT-PP}$	Idle detect voltage seen at the Rx die pad	172	mVPP	Assuming Rx RTERM of $2 \times 50 \Omega$
$V_{CH-EIEOS-FS-Vb}$	PP voltage during Vb interval at behavioral Tx die pad for full swing signaling	255	mVPP	Assuming Tx RS of $2 \times 50 \Omega$
$V_{CH-EIEOS-RS-Vb}$	PP voltage during Vb interval at behavioral Tx die pad for reduced swing signaling	237	mVPP	Assuming Tx RS of $2 \times 50 \Omega$

8.6 Refclk Specifications

此版本的规范巩固并简化了 Refclk 的要求。2.5 GT/s Refclk 参数已从 CEM 规范移至该规范, 因此该小节现在包含了所有数据速率 (2.5、5.0、8.0、16.0 和 32.0 GT/s) 的所有 Refclk 参数。

8.6.1 Refclk Test Setup

Refclk 的测试环境假定只有 Refclk 生成器本身存在。测试设置中规定了实际系统中 Refclk 发生器的引脚与发送器或接收器的引脚之间发生的信号劣化。上述设置模拟了在 PCI Express 设备的引脚上可能发生的最坏情况下的信号劣化。请注意, Refclk 信号在代表 PCIe 设备 Refclk 输入引脚处出现的串联 (开路) 终端的负载中进行测试, 以满足除 32.0 GT/s 参考时钟抖动以外的所有要求。使用相位噪声分析仪测得的参考时钟抖动和用示波器测得的 32.0 GT/s 参考时钟抖动, 用 50 ohm 端接的无信道基准时钟进行测试。

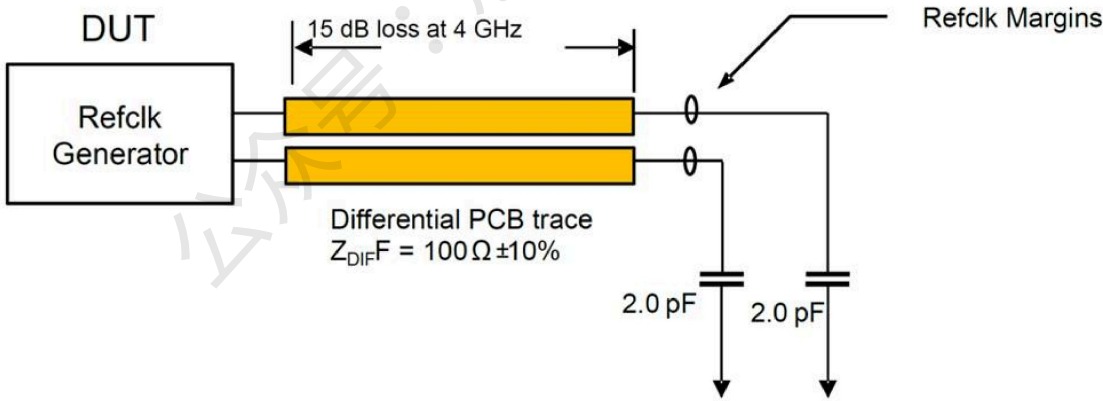


Figure 8-64 Oscilloscope Refclk Test Setup For All Cases Except Jitter at 32.0 GT/s

8.6.2 REFCLK AC Specifications

Table 8-16 中的所有规格均应使用 Note 11 中所述的测试配置和 Figure 8-64 所示的电路进行测量。

Table 8-16 REFCLK DC Specifications and AC Timing Requirements

Symbol	Parameter	100 MHz Input		Unit	Note
		Min	Max		
Rising Edge Rate	Rising Edge Rate	0.6	4.0	V/ns	2, 3
Falling Edge Rate	Falling Edge Rate	0.6	4.0	V/ns	2, 3
V_{IH}	Differential Input High Voltage	+150		mV	2
V_{IL}	Differential Input Low Voltage		-150	mV	2
V_{CROSS}	Absolute crossing point voltage	+250	+550	mV	1, 4, 5
V_{CROSS DELTA}	Variation of V _{CROSS} over all rising clock edges		+140	mV	1, 4, 9
V_{RB}	Ring-back Voltage Margin	-100	+100	mV	2, 12
T_{STABLE}	Time before V _{RB} is allowed	500		ps	2, 12
T_{PERIOD AVG}	Average Clock Period Accuracy	-300	+2800	ppm	2, 10, 13
T_{PERIOD AVG_32G_CC}	Average Clock Period Accuracy for devices that support 32.0 GT/s in CC Mode at any speed	-100	+2600	ppm	2, 10, 13
T_{PERIOD AVG_32G_SRIS}	Average Clock Period Accuracy for devices that support 32.0 GT/s in SRIS Mode at any speed	-100	+1600	ppm	2, 10, 13
T_{PERIOD ABS}	Absolute Period (including Jitter and Spread Spectrum modulation)	9.847	10.203	ns	2, 6
T_{PERIOD ABS_32G_CC}	Absolute Period (including Jitter and Spread Spectrum modulation) for devices that support 32.0 GT/s in CC Mode at any speed	9.849	10.201	ns	2, 6
T_{PERIOD ABS_32G_SRIS}	Absolute Period (including Jitter and Spread Spectrum modulation) for devices that support 32.0 GT/s in SRIS Mode at any speed	9.849	10.181	ns	2, 6
T_{CCJITTER}	Cycle to Cycle jitter		150	ps	2
V_{MAX}	Absolute Max input voltage		+1.15	V	1, 7
V_{MIN}	Absolute Min input voltage		-0.3	V	1, 8

Symbol	Parameter	100 MHz Input		Unit	Note
		Min	Max		
Duty Cycle	Duty Cycle	40	60	%	2
Rise-Fall Matching	Rising edge rate (REFCLK+) to falling edge rate (REFCLK-) matching		20	%	1, 14
Z_{C-DC}	Clock source DC impedance	40	60	W	1, 11

Notes:

1. 从单端波形上进行测量获取结果。
2. 从差分波形上进行测量获取结果。
3. 在 -150 mV 至 +150 mV 的差分波形上测量（源自 REFCLK + 减去 REFCLK -）。对于上升和下降时间，信号在测量区域中必须是单调的。300 mV 测量窗口以差分过零为中心。参见 Figure 8-69。

4. 在交叉点测量，该交叉点的 REFCLK+上升沿的瞬时电压值等于 REFCLK-的下降沿。参见 Figure 8-65。
5. 指的是从最低的交叉点到最高的交叉点的总变化，而不管是哪条边交叉。参考此测量的所有交叉点。参见 Figure 8-65。
6. 定义为绝对的最小或最大瞬时周期。这包括逐周期抖动，相对 PPM 容差和扩频调制。参见 Figure 8-68。
7. 定义为最大瞬时电压，包括过冲。参见 Figure 8-65。
8. 定义为包括下冲在内的最小瞬时电压。参见 Figure 8-65。
9. 定义为上升 REFCLK+和下降 REFCLK-的所有交叉电压的总变化。这是任何特定系统在 VCROSS 中允许的最大方差。参见 Figure 8-66。
10. 此 Nore 已删除。
11. REFCLK+和 REFCLK-将在负载电容器 CL 处测量。单端探头必须用于需要单端测量的测量。具有数学运算功能的单端探头或差分探头均可用于差分测量。测试负载 CL = 2 pF。
12. T_{STABLE} 是差分时钟在上升沿/下降沿之后必须保持最小 ± 150 mV 差分电压的时间，然后才允许其回落至 $VRB \pm 100$ mV 差分范围。参见 Figure 8-70。
13. PPM 指百万分之一，是 DC 绝对周期精度指标。1 PPM 恰好是 100.000000 MHz 或 100 Hz 的 $1/1,000,000$ 。例如，对于 300 PPM，我们的误差预算为 $100 \text{ Hz} / \text{PPM} \times 300 \text{ PPM} = 30 \text{ kHz}$ 。该周期将使用频率窗口（其测量窗口设置为 100 ms 或更大）进行测量。
14. 匹配适用于 REFCLK+的上升沿速率和 REFCLK-的下降沿速率。它是使用 ± 75 mV 窗口进行测量的，该窗口以 REFCLK+上升与 REFCLK-下降相交的中点交叉点为中心。中点交叉点用于计算示波器要用于边沿速率计算的电压阈值。REFCLK+的上升沿速率应与 REFCLK-的下降沿速率进行比较；最大允许差异不应超过最慢边缘速率的 20%。参见 Figure 8-67。

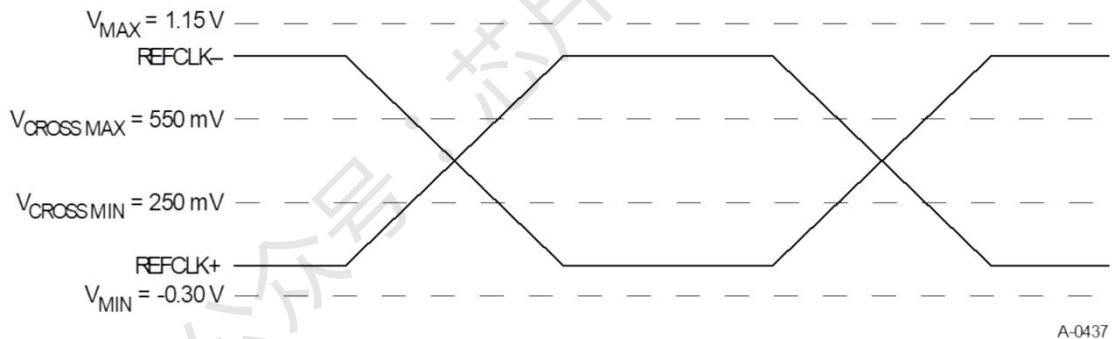


Figure 8-65 Single-Ended Measurement Points for Absolute Cross Point and Swing

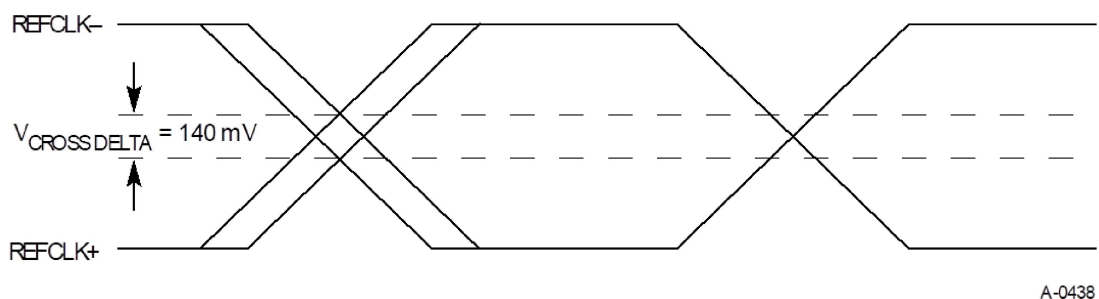


Figure 8-66 Single-Ended Measurement Points for Delta Cross Point

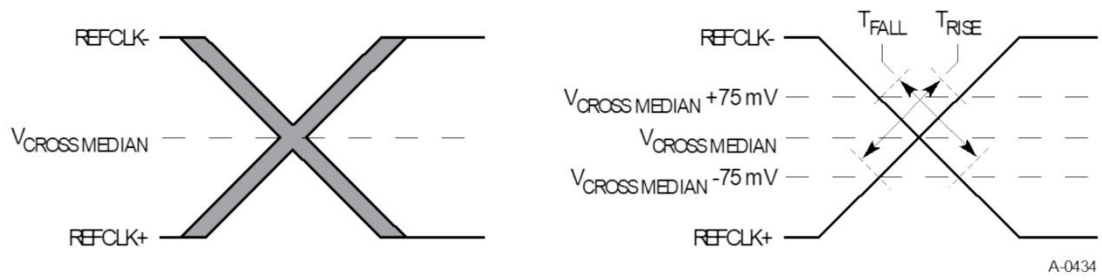


Figure 8-67 Single-Ended Measurement Points for Rise and Fall Time Matching

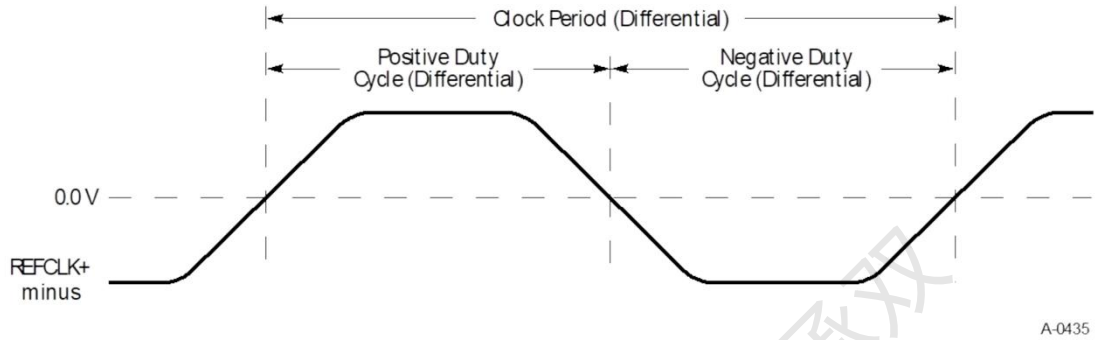


Figure 8-68 Differential Measurement Points for Duty Cycle and Period

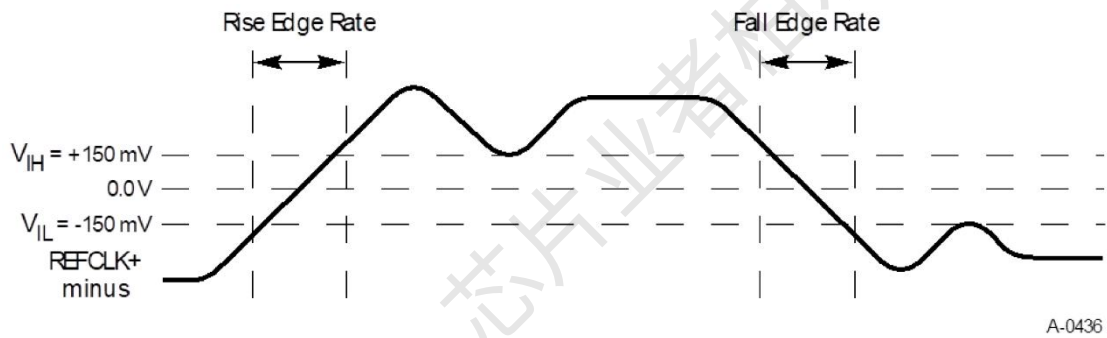


Figure 8-69 Differential Measurement Points for Rise and Fall Time

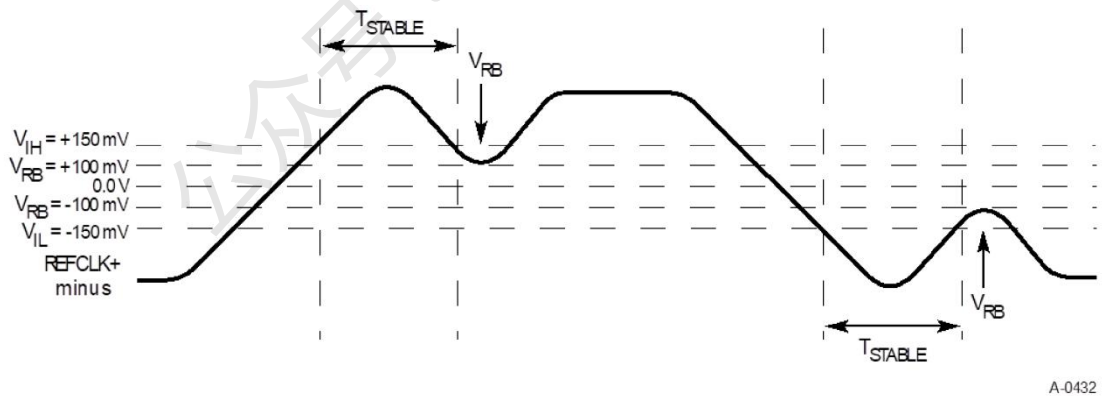


Figure 8-70 Differential Measurement Points for Ringback

8.6.3 Data Rate Independent Refclk Parameters

多个 Refclk 参数与数据速率无关，并在下表中列出。 $T_{\text{TRANSPORT-DELAY}}$ 在 Section 8.6.6 中定义，并在 Figure 8-73 中进行了说明。它仅与 Common Refclk 结构有关。对于 SRIS 模式，SSC 调制的来源取决于实现方式。

Table 8-17 Data Rate Independent Refclk Parameters

Symbol	Description	Limits	Units	Notes
<i>F_{REFCLK}</i>	Refclk Frequency	99.97 (min) 100.03 (max)	MHz	
<i>F_{REFCLK_32G}</i>	Refclk Frequency for devices that support 32.0 GT/s	99.99 (min) 100.01 (max)	MHz	
<i>F_{SSC}</i>	SSC frequency range	30 (min) 33 (max)	kHz	3
<i>T_{SSC-FREQ-DEVIATION}</i>	SSC deviation	-0.5 (min) 0.0 (max)	%	3
<i>T_{SSC-FREQ-DEVIATION_32G_SRIS}</i>	SSC deviation for devices that support 32.0 GT/s and SRIS when operating in SRIS mode at all speeds	-0.3 (min) 0.0 (max)	%	3

Symbol	Description	Limits	Units	Notes
<i>T_{TRANSPORT-DELAY}</i>	Tx-Rx transport delay	12 (max)	ns	1, 4
<i>T_{SSC-MAX-FREQ-SLEW}</i>	Max SSC df/dt	1250	ppm/ μs	2, 3

Notes:

1. 该参数仅与 Common Refclk 结构有关。
2. 使用一阶 LPF 在 0.5μs 的时间间隔内进行测量，其 f_c 为调制频率的 60 倍。
3. 在测试配置为 IR 参考时钟体系结构的设备时，必须使用 Tx 输出数据而不是参考时钟来测试 SSC 相关参数。
4. 有些外形规格（例如，包括长电缆在内的拓扑）可能会超出传输延迟限制。这些尺寸规格说明必须考虑到较大的传输延迟带来的额外抖动。

8.6.3.1 Low Frequency Refclk Jitter Limits

支持 SSC 的反射器必须在一定范围的低频范围内满足额外的抖动限制。低频 Refclk 抖动极限定义为从 30 kHz 到 500 kHz 的连续的分段线性图，如下所示。在感兴趣的频率范围内，未滤波的 Refclk 相位抖动必须低于此图。

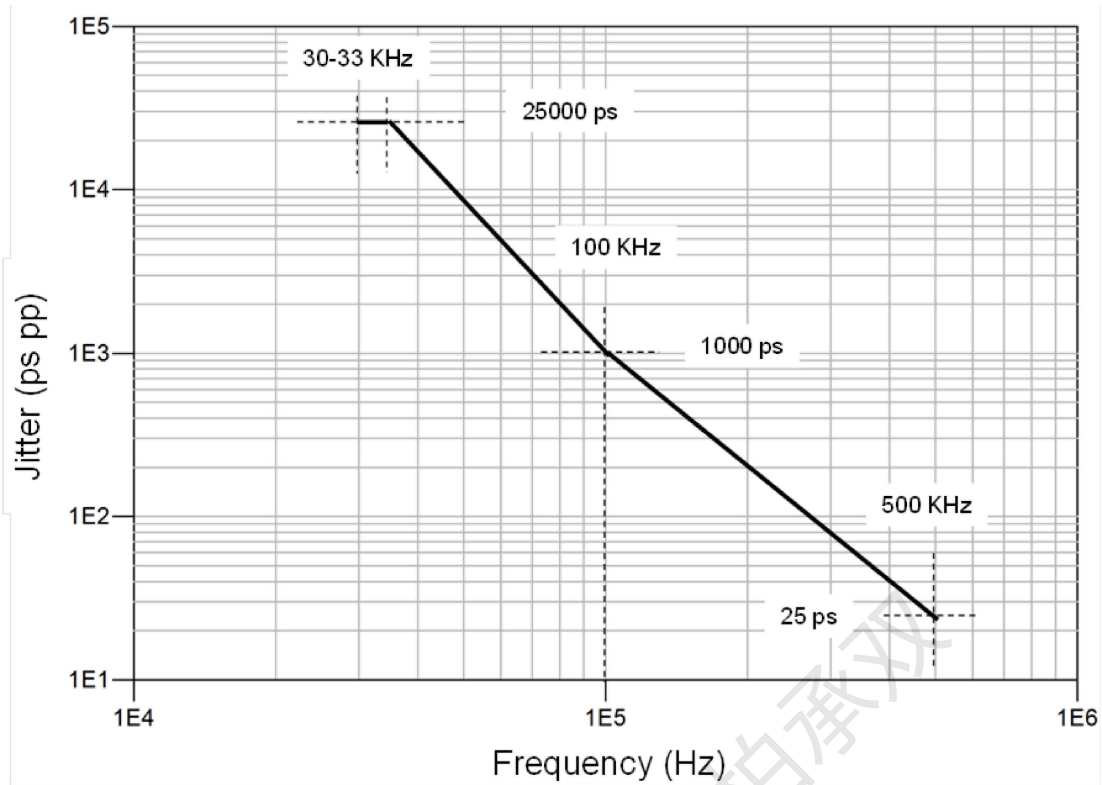


Figure 8-71 Limits for phase jitter from the Reference with 5000 ppm SSC

8.6.4 Refclk Architectures Supported

规范定义了两种 Refclk 架构：Common Refclk (CC)和 Independent Refclk (IR)。根据 CC 时钟的拓扑结构和基于 PLL 和 CDR 特性的相应抖动传递函数，对 CC 时钟体系结构进行了描述。最后，为每个数据速率给出了相应的 Refclk 抖动限值。没有为 IR 时钟架构定义抖动传递函数和相应的抖动限制。实现者需要权衡参考时钟抖动和 PLL 特性，以确保在 IR 时钟模式下满足发射器要求。

8.6.5 Filtering Functions Applied to Raw Data

两种类型的过滤应用于原始 Refclk 数据。首先，边缘滤波可将由于测试设备的有限采样率引起的测量引起的抖动降至最低。第二，复制在 Tx/Rx PLL, Rx CDR 和（如果适用）传输延迟的组合中固有的抖动过滤。前面的滤波器功能的组合产生有效的 Refclk 抖动，就像它出现在接收器的采样锁存器上一样。

请注意，PLL 和 CDR 滤波器功能代表了实际接收器实现的最小能力近似值，并且没有定义实际 PLL 或 CDR 实现。

8.6.5.1 PLL Filter Transfer Function Example

如 Figure 8-73 所示，所有 PLL 的行为均采用二阶传递函数 $H(s)$ 进行建模。定义传递函数的参数包括阻尼系数 ζ 和固有频率 ω_n 。

二阶 PLL（低通）固有频率 ω_n 和 3 dB 点 ω_{3dB} 之间的关系由以下表达式给出：

$$\frac{\omega_{3dB}}{\omega_n} = \sqrt{\sqrt{(2\zeta^2 + 1)^2 + 1} + 2\zeta^2 + 1}$$

Equation 8-8 Relationship between 2nd order PLL natural frequency and 3 dB point

下面的二阶 PLL 图说明了 f_{3dB} 为 5.0 MHz 和峰值为 1.0 dB 时的传递函数。这对应于 $\zeta = 1.15$ ，并且 $\omega_n = 11.55$ Mrad/sec。

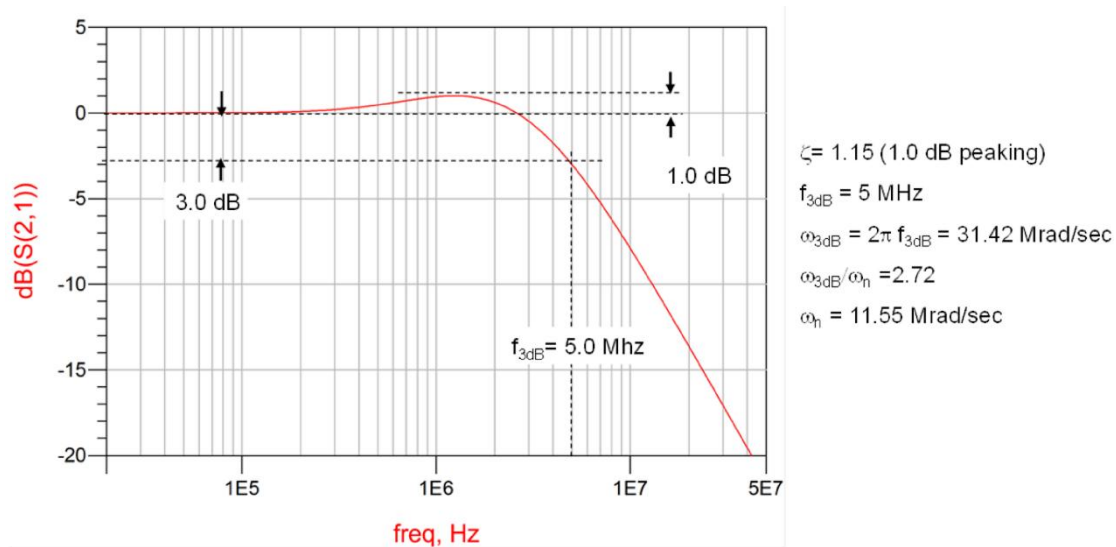


Figure 8-72 5 MHz PLL Transfer Function Example

8.6.5.2 CDR Transfer Function Examples

根据 Refclk 的结构和数据速率，一阶或更高阶的传递函数应被用作行为 CDR 边界限制。

有关行为 CDR 功能，请参见 Section 8.3.5.5。

8.6.6 Common Refclk Rx Architecture (CC)

该结构使用了一个同时分配到 Tx 和 Rx 的 Refclk 源。Refclk 产生的大多数 SSC 抖动通过 Tx 和 Rx PLL 均等传播，因此本质上跟踪 LF 抖动。对于倾向于低频的 SSC 尤其如此。Figure 8-73 给出了常见的 Common Refclk Rx 架构，显示了除 32.0 GT/s 以外的所有数据速率的关键抖动、延迟、PLL 和 CDR 传递函数源。在 32.0 GT/s 下，图中的唯一区别是 Behavioral CDR 传递函数，如 Section 8.3.5.5 中所定义。然后，由 Tx 和 Rx PLL 之间的差函数乘以 CDR 高通特性来定义 CDR 上出现的抖动量。

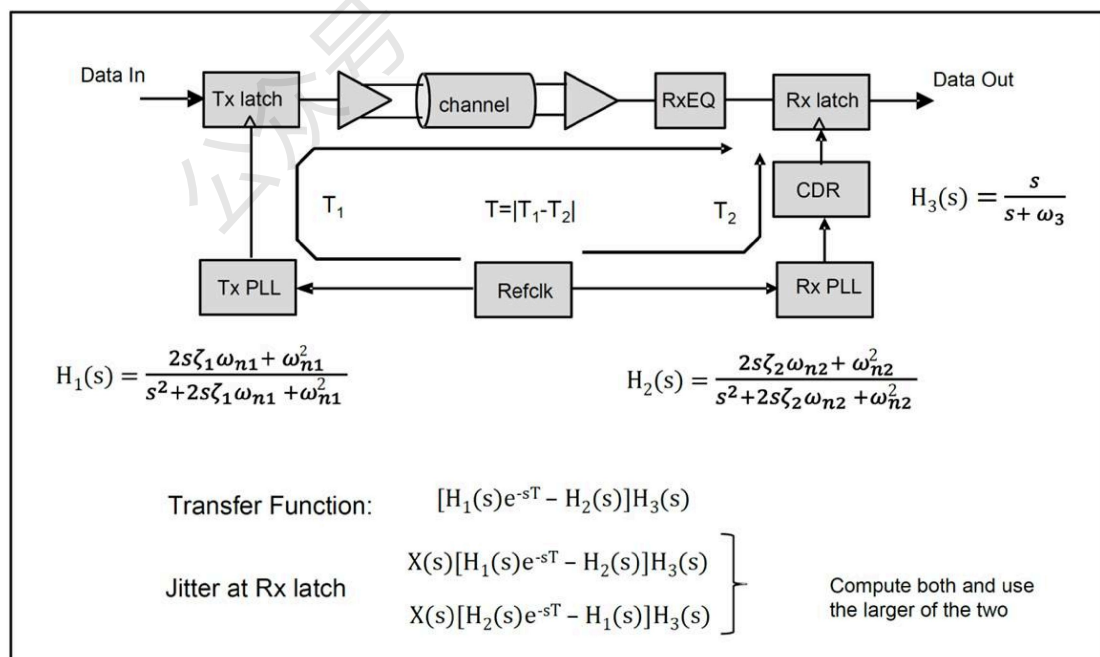


Figure 8-73 Common Refclk Rx Architecture for all Data Rates Except 32.0 GT/s

基于上述时钟架构，可以定义一个差分函数，该函数对应于 Tx 和 Rx PLL 之间的最坏不匹配情况。假定使用二阶 PLL 传递函数（即使大多数 PLL 传递函数为三阶或更高阶），因为与大多数实际 PLL 实现相比，二阶函数倾向于产生稍微保守的差分函数。

在 Common Refclk Rx 时钟结构中，还必须理解最大的发送器到接收器传输延迟差。该延迟增量如 Figure 8-73 所示，代表了发送器数据与恢复的接收器时钟之间的延迟差，如接收数据锁存器的输入所示。

8. 6. 6. 1 Determining the Number of PLL BW and peaking Combinations

Tx 或 Rx PLL 由最小/最大带宽和峰值的组合定义，总共有四个可能的限制。在 CC 架构中，Tx 和 Rx PLL 均有助于抖动传递函数。在 2.5 GT/s 下，仅定义了一组带宽/峰值限值。如果 Tx 和 Rx PLL 限值的组合由集合 (ATX, BTX, CTX, DTX)，(ARX, BRX, CRX, DRX) 定义，那么很容易证明总共有十种从每个集合选择一个元素的方法。延迟项 e^{-sT} 可以应用于 H1(s) 或 H2(s)，这增加了另外六个可能性。当考虑延迟项时，仅添加六个项，而不是十个项，因为诸如 ATX, ARX 之类的项与 ARX, ATX 相同。

在 5.0 GT/s 和更高的数据速率下，定义了 PLL 带宽和峰值的两个可能的限制集，它们可以由 (ATX, BTX, CTX, DTX) 和 (ERX, FRX, GRX, HRX) 集来定义。在这种情况下，来自上述 4 个元素集的唯一 2 个元素组合的数量为 36，当考虑延迟项时，该数量增加到 64。

8. 6. 6. 2 CDR and PLL BW and Peaking Limits for Common Refclk

Common Refclk 时钟结构的滤波器功能取决于 BW 和 Tx 和 Rx PLL 的峰值以及 CDR 高通特性所定义的差分特性。必须考虑 Tx 和 Rx PLL 峰值和带宽以及其最小和最大峰值下的 CDR 特性的所有特殊情况组合。

此过程必须应用于所有五个数据速率。

Figure 8-74 列出了需要用作 2.5 GT/s 数据速率的滤波器功能的 PLL BW 和 CDR BW /峰值。必须为 2.5 GT/s PLL 分配峰值的最小值和最大值。0.01 dB 和 3.0 dB 的值代表实际 PLL 实现的最佳估计。

2.5 和 5.0 GT/s 的最小峰值已降低至 0.01 dB，以使其与 8.0 和 16.0 GT/s 的情况保持一致。请注意，Rx CDR 是一阶的，因此它的固有频率 ω_n 可以直接从其 BW 获得，而二阶 CDR 则不同，其中 ω_n 是 BW 和峰值的函数。对于 32.0 GT/s，必须使用 Equation 8-6 所述的具有 20 MHz BW 的二阶 CDR 滤波器。

PLL #1, PLL #2	0.01 dB peaking	3.0 dB peaking	BW_{CDR(min)} = 1.5 MHz, 1st order CDR
BW _{PLL(min)} = 1.5 MHz	$\omega_{n1} = .336 \text{ Mrad/s}$ $\zeta_1 = 14$	$\omega_{n1} = 5.09 \text{ Mrad/s}$ $\zeta_1 = 0.54$	
BW _{PLL(max)} = 22 MHz	$\omega_{n1} = 4.93 \text{ Mrad/s}$ $\zeta_1 = 14$	$\omega_{n1} = 74.68 \text{ Mrad/s}$ $\zeta_1 = 0.54$	
16 combinations			2.5 GT/s

Figure 8-74 Common Refclk PLL and CDR Characteristics for 2.5 GT/s

5.0、8.0 和 16.0 GT/s 的 PLL 和 CDR 抖动和峰值特性产生了大量可能的组合，因为给出了两组 PLL BW 和峰值限制。选择这种支持两组带宽和峰值的方法是为了给设计人员在设计 PLL 电路时提供尽可能多的自由度。

PLL #1	0.01 dB peaking	1.0 dB peaking	PLL #2	0.01 dB peaking	3.0 dB peaking
BW _{PLL(min)} = 5.0 MHz	$\omega_{n1} = 1.12 \text{ Mrad/s}$ $\zeta_1 = 14$	$\omega_{n1} = 11.01 \text{ Mrad/s}$ $\zeta_1 = 1.16$	BW _{PLL(min)} = 8.0 MHz	$\omega_{n2} = 1.79 \text{ Mrad/s}$ $\zeta_2 = 14$	$\omega_{n2} = 26.86 \text{ Mrad/s}$ $\zeta_2 = 0.54$
BW _{PLL(max)} = 16 MHz	$\omega_{n1} = 3.58 \text{ Mrad/s}$ $\zeta_1 = 14$	$\omega_{n1} = 35.26 \text{ Mrad/s}$ $\zeta_1 = 1.16$	BW _{PLL(max)} = 16 MHz	$\omega_{n2} = 3.58 \text{ Mrad/s}$ $\zeta_2 = 14$	$\omega_{n2} = 53.73 \text{ Mrad/s}$ $\zeta_2 = 0.54$
BW_{CDR(min)} = 5 MHz, 1st order			64 combinations		
			5 GT/s		

Figure 8-75 Common Refclk PLL and CDR Characteristics for 5.0 GT/s

PLL #1	0.01 dB peaking	2.0 dB peaking	PLL #2	0.01 dB peaking	1.0 dB peaking
$BW_{PLL(min)} = 2.0$ MHz	$\omega_{n1} = 0.448$ Mrad/s $\zeta_1 = 14$	$\omega_{n1} = 6.02$ Mrad/s $\zeta_1 = 0.73$	$BW_{PLL(min)} = 2.0$ MHz	$\omega_{n2} = 0.448$ Mrad/s $\zeta_2 = 14$	$\omega_{n2} = 4.62$ Mrad/s $\zeta_2 = 1.15$
$BW_{PLL(max)} = 4.0$ MHz	$\omega_{n1} = 0.896$ Mrad/s $\zeta_1 = 14$	$\omega_{n1} = 12.04$ Mrad/s $\zeta_1 = 0.73$	$BW_{PLL(max)} = 5.0$ MHz	$\omega_{n2} = 1.12$ Mrad/s $\zeta_2 = 14$	$\omega_{n2} = 11.53$ Mrad/s $\zeta_2 = 1.15$
$BW_{CDR(min)} = 10$ MHz, 1 st order	64 combinations				8.0, 16.0 GT/s

Figure 8-76 Common Refclk PLL and CDR Characteristics for 8.0 and 16.0 GT/s

PLL #1, PLL #2	0.01 dB peaking	2.0 dB peaking	32.0 GT/s CC	CDR
$BW_{PLL(min)} = 0.5$ MHz	$\omega_{n1} = .112$ Mrad/s $\zeta_1 = 14$	$\omega_{n1} = 1.51$ Mrad/s $\zeta_1 = 0.73$	16 combinations	
$BW_{PLL(max)} = 1.8$ MHz	$\omega_{n1} = .403$ Mrad/s $\zeta_1 = 14$	$\omega_{n1} = 5.42$ Mrad/s $\zeta_1 = 0.73$		
				32.0 GT/s

Figure 8-77 Common Refclk PLL and CDR Characteristics for 32.0 GT/s

8.6.7 Jitter Limits for Refclk Architectures

Table 8-18 列出了四种数据速率下 CC Refclk 时钟结构的抖动限制。

2.5 GT/s 的抖动被测量为峰峰值抖动值，因为抖动的很大一部分是 SSC 谐波，在接收器上显示为 D_j 。2.5 GT/s PLL 和 CDR 带宽的组合会通过大量 SSC 残差，表现为 D_j 。

对于 5.0、8.0 和 16.0 GT/s，抖动被指定为 RMS (R_j) 值。这些信号速度利用较低的 PLL BW 和较高的 CDR BW，其作用是抑制 SSC 谐波，从而几乎所有的抖动都表现为 R_j 。

Table 8-18 Jitter Limits for CC Architecture

Data Rate	CC jitter Limit	Notes
2.5 GT/s	86 ps pp	1, 2
Data Rate	CC jitter Limit	Notes
5.0 GT/s	3.1 ps RMS	1, 2
8.0 GT/s	1.0 ps RMS	1, 2
16.0 GT/s	0.5 ps RMS	1, 2, 3, 4
32.0 GT/s	0.15 ps RMS	1, 2, 3, 5

Notes:

1. Refclk 抖动是在应用了 Figure 8-73 中的滤波器功能后测量的。
2. 抖动测量应通过实时示波器 (RTO) 以至少 20 GS/s 或更高的采样率捕获至少 100,000 个时钟周期来进行。测量时必须将宽带示波器的噪声降至最低。测得的 PP 抖动用于 RTO 测量 (无外推)。或者抖动测量可以与相位噪声分析仪 (PNA) 结合使用 (扩展) (平坦)，并整合和折叠频率成分，直到与奈奎斯特频率以下至少 200 MHz (绝对频率为 300 MHz) 的载波频率偏移。对于 2.5 GT/s 数据速率的 PNA 测量，使用 8.83 的倍增系数将 RMS 抖动转换为峰到峰抖动。如果同时进行了实时示波器和 PNA 测量并产生了不同的结果，则必须使用 RTO 结果。
3. 对于 16.0 GT/s 和 32.0 GT/s CC 测量，从基波和谐波中去除 SSC 杂散，直到 2 MHz 的截止频率，同时注意将任何非 SSC 成分的去除降至最低。

4. 请注意，信道仿真中将使用 0.7 ps RMS 值，以解决实际系统中的额外噪声。
5. 请注意，信道仿真中将使用 0.25 ps RMS 的值，以解决实际系统中的额外噪声。

8.6.8 Form Factor Requirements for RefClock Architectures

每个外形规格说明都必须包括下表（Table 8-19），以提供对支持规格说明的设备的时钟体系结构要求的清晰摘要。对于每种时钟体系结构，该表均指出此外形尺寸是该体系结构是必需的，可选的还是不允许的。请注意，这是指设备的操作而不是底层的硅功能。

如果支持 L1 PM 子状态，则外形规格必须提供 CLKREQ # 信号。外形规格规范必须指出 CLKREQ # 信号是必需的，可选的还是不允许的。

Table 8-19 Form Factor Clocking Architecture Requirements

Clock Architecture	System Board (Motherboard)	Add-in Card (Module)	Retimer
Common	**	**	**
SRNS	**	**	**
SRIS	**	**	**

** Each entry in the table must be filled in with one of: Required, Optional, or Not Allowed

如果外形规格定义 Common Reference Clock 结构是必需的或可选的，则必须有一个附加表（Table 8-20），其中提供了 common clock 的详细信息。表中的每个条目都标记为必填，可选，不允许或不适用。“Clock Source”表示公用参考时钟的源（如果适用）。“SSC”指示时钟源是否扩展。

Table 8-20 Form Factor Common Clock Architecture Details

Common Clock Details	System Board (Motherboard)	Add-in Card (Module)	Retimer
Clock Source			
SSC			

如果某种形式的时钟要求不能以这种简单的一两张表格的形式提供，则必须仔细考虑以确保该规范支持这种形式的要求。

例如，显示的填充表是一个假设的外形规格，该外形规格要求所有组件都使用 common clock 架构，并且不允许使用任何其他时钟架构（Table 8-21 和 Table 8-22）。common clock source 需要由主板组件提供，并且可以选择具有 SSC。不支持 L1 PM 子状态，因此在此外形规格中，CLKREQ # 未定义为连接器信号。

Table 8-21 Form Factor Clocking Architecture Requirements Example

Clock Architecture	System Board (Motherboard)	Add-in Card (Module)	Retimer
Common	Required	Required	Required
SRNS	Not Allowed	Not Allowed	Not Allowed
SRIS	Not Allowed	Not Allowed	Not Allowed

Table 8-22 Form Factor Common Clock Architecture Details Example

Common Clock Details	System Board (Motherboard)	Add-in Card (Module)	Retimer
Clock Source	Required	Not Allowed	Not Allowed
SSC	Optional	N/A	N/A

对于外形规格规范,重要的是要认识到:如果要支持 L1 PM 子状态,则需要 CLKREQ # 信号;对于 L1 PM 子状态,即使没有 common reference clock,也要使用 CLKREQ # 信号。

如果某种外形规格的时钟要求不能以这种简单的一两张表格的形式提供,则必须仔细考虑以确保该规范支持这种外形规格的要求。

公众号: 芯片业者柏承双

9 Single Root I/O Virtualization and Sharing

9.1 SR-IOV Architectural Overview

在工业上，已经花费了大量的努力来通过使用虚拟化技术来提高有效的硬件资源利用率。Single Root I/O Virtualization and Sharing (SR-IOV) 使多个系统镜像 (SI, System Image) 可以共享 PCI 硬件资源。

为了说明如何使用此技术来提高有效资源利用率，请参考 Figure 9-1 中所示的通用平台配置。

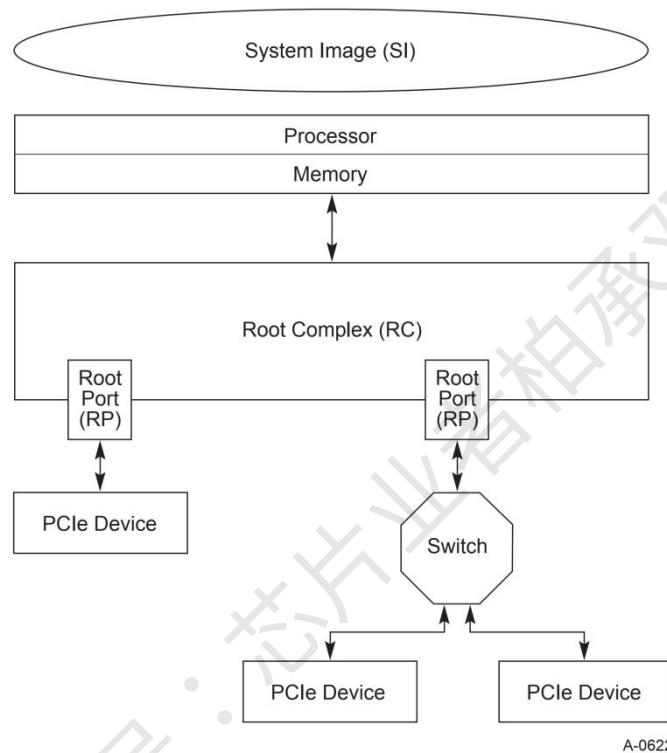


Figure 9-1 Generic Platform Configuration

通用平台配置由以下组件组成：

- PCIe RC (RC)，包括：
 - Processor —— 通用、嵌入式或专用处理元件。
 - Memory —— 通用或嵌入式。
 - RC Integrated Endpoints (RCiEPs)。
 - PCIe Root Ports (RP) —— 每个 RP 代表一个单独的层次结构。每个层次结构都称为单个根层次结构，以将其与[MR-IOV]中定义的多重层次结构技术区分开。
- PCIe Switch——提供 I/O 扇出和连接：
 - PCIe Device —— 多种 I/O 设备类型（例如，网络，存储等）。
 - System Image —— 一种用于执行应用程序或受信任服务的操作系统之类的软件（例如，共享或非共享 I/O 设备驱动程序。）

为了增加有效的硬件资源利用率而无需修改硬件，可以执行多个 SI。如 Figure 9-2 所示，在硬件和 SI 之间插入了称为虚拟化媒介（VI，Virtualization Intermediary）的软件。

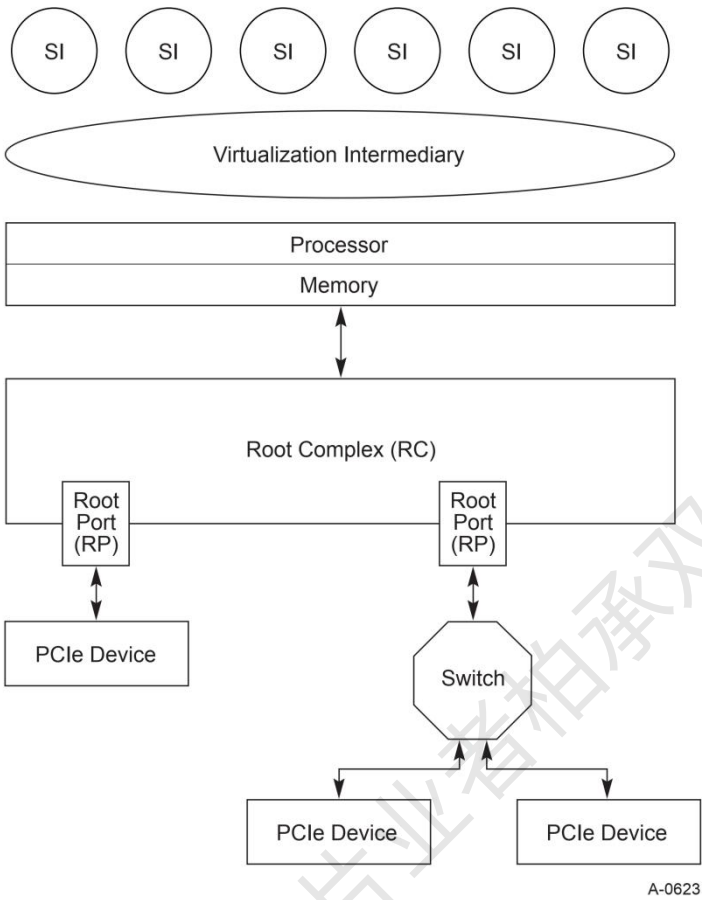


Figure 9-2 Generic Platform Configuration with a VI and Multiple SI

VI 仅拥有底层硬件的所有权。VI 使用了超出本规范范围的各种方法，对硬件进行了抽象化处理，为每个 SI 提供了自己的虚拟系统。每个 SI 可用的实际硬件资源可能会根据工作负载或特定于客户的策略而有所不同。尽管此方法在许多环境中都很好用，但是 I/O 密集型工作负载可能会导致性能严重下降。VI 必须拦截和处理每个 I/O 操作（inbound or outbound），这会增加大量平台资源的开销。

SR-IOV 提供了减少这些平台资源开销的工具。SR-IOV 的优点是：

- 消除 VI 参与主要数据搬运动作，即 DMA，内存空间访问，中断处理等的的能力。消除 VI 拦截和每个 I/O 操作的处理可以显著提高应用程序和平台的性能。
- 通过 Single Root PCI Manager (SR-PCIM)控制 SR-IOV 资源配置和管理的标准化方法。
 - 由于实现选项多种多样，包括系统固件，VI，操作系统，I/O 驱动程序等。SR-PCIM 实现不在本规范范围内。
- 通过在设备内预配大量 I/O Function 来降低硬件要求和相关成本的能力。
- 能够将 SR-IOV 与其他 I/O 虚拟化技术集成在一起，例如 ATS 技术、ATPT 技术以及中断重新映射技术，以创建强大、完整的 I/O 虚拟化解决方案。

Figure 9-3 给出了一个支持 SR-IOV 的示例平台。

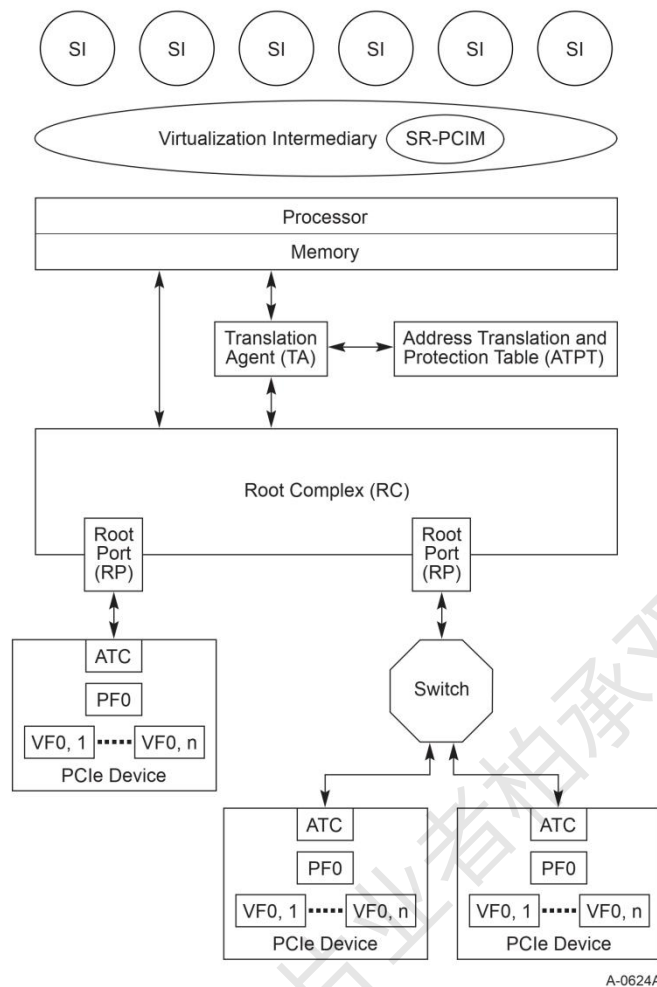


Figure 9-3 Generic Platform Configuration with SR-IOV and IOV Enablers

SR-IOV 通用平台配置由以下额外的功能元件组成：

- **SR-PCIM** —— 负责配置 SR-IOV Extended Capability, 管理 Physical Function 和 Virtual Function 以及处理相关的错误事件和整体设备控制（例如电源管理和热插拔服务）的软件。
- 可选实现的 **Translation Agent (TA)** —— TA 是硬件或硬件和软件的组合，负责将 PCIe 事务中的地址转换为关联的平台物理地址。一个 TA 可能包含一个地址转换缓存 (ATC)，以加速转换表的访问。TA 还可以支持地址转换服务 (ATS)，该地址转换服务使 PCIe Function 能够获得对相关存储器进行 DMA 访问的先验地址转换。有关 ATS 的好处和操作的更多详细信息，请参见 Chapter 10。
- 可选实现的 **Address Translation and Protection Table (ATPT)** —— ATPT 包含由 TA 访问以处理 PCIe 请求（DMA 读取，DMA 写入或中断请求）的地址转换集。有关其他详细信息，请参见 ATS (Chapter 10)。
 - 在 PCIe 中，中断被视为内存写操作。通过请求者标识符和 PCIe 事务中包含的地址的组合，可以将中断路由到相应的 I/O Function 透明的任何目标（例如，处理器内核）。
 - DMA 读和写请求通过 Routing ID 和 PCIe 事务中包含的地址的组合进行转换。
- 可选实现的 **Address Translation Cache (ATC)** —— ATC 可以存在于平台内的两个位置：在 TA 内实现，即可以集成在 RC 内或位于 RC 上方；也可以在 PCIe 设备内实现 ATC。在 RC 中实现 ATC，ATC 可以加快翻译查找的速度。在设备内实现 ATC，通过 ATS 技术填充 ATC。指示其包含已转换地址的 PCIe 事务可绕过平台的 ATC，以提高性能而不会损害与 ATPT 技术相关的收益。有关其他详细信息，请参见 ATS (Chapter 10)。

- 可选实现的 Access Control Services (ACS) —— ACS 在 PCI Express 拓扑中定义了一组控制点，以确定是否应正常路由，阻塞或重定向 TLP。在支持 SR-IOV 的系统中，ACS 可用于防止分配给 VI 或不同 SI 的 device Function 彼此通信或与对等设备通信。重定向可以允许转换代理在做出对等转发决定之前转换上游存储器 TLP 地址。可选的 ACS P2P Egress Control 可以提供选择性阻塞。ACS 必须与 ATS 进行交互。有关更多详细信息，请参见 Section 9.3.7.6。
- Physical Function (PF) —— PF 是一种支持 SR-IOV Extended Capability 的 PCIe Function，并且 SR-PCIM，VI 或 SI 均可访问。
- Virtual Function (VF) —— VF 是 SI 可以直接访问的“轻量级” PCIe Function。
 - 最低限度下，与 Function 的主要数据搬运相关的资源可用于 SI。配置资源应仅限于受信任的软件组件，例如 VI 或 SR-PCIM。
 - 一个 VF 可以由不同的 SI 串行共享（即，一个 VF 可以分配给一个 SI，然后重置并分配给另一个 SI。）
 - 可以将 VF 从一个 PF 迁移到另一个 PF。迁移过程本身不在本规范的范围之内，但可以通过本规范内定义的配置控件来简化。
- 与 PF 关联的所有 VF 必须与 PF 具有相同的设备类型（例如，相同的网络设备类型或相同的存储设备类型。）

要将 PCIe 设备与具有 PCIe SR-IOV 的设备进行比较和对比，请检查以下几组图。Figure 9-4 说明了一个 PCIe 兼容设备的示例。

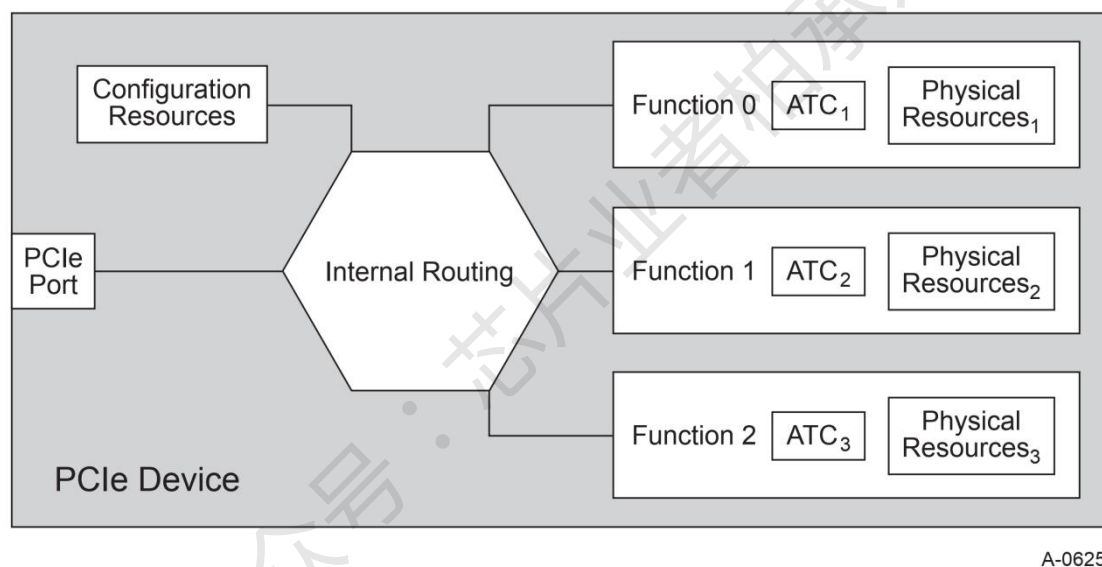


Figure 9-4 Example Multi-Function Device

该图说明了具有以下特征的 multi-Function PCIe Device 的示例：

- PCIe 设备共享一个公共 PCIe 链路。所有 Function 共享的链路和 PCIe Feature 通过 Function 0 进行管理。
- 在此示例中，每个 PCIe Function 都支持 ATS 功能，因此实现了相关的 ATC 来管理 ATS 获得的转换地址。
 - 虽然该图仅说明了三个 Function，但使用 Alternative Routing Identifier (ARI) capability，PCIe 设备最多可以支持 256 个 Function。
 - 所有 Function 都通过 PCI 枚举过程捕获的单个 Bus Number。
- 每个 PCIe Function 都有一组唯一的物理资源，包括单独的配置空间和 BAR。
- 每个 PCIe Function 都可以分配给一个 SI。为防止一个 SI 影响另一个 SI，VI 应拦截并处理所有 PCIe 配置操作。

如此图所示，硬件资源随所提供 Function 的数量而扩展。根据设备的复杂性和尺寸，每个 Function 的增量成本将有所不同。为了减少增加的硬件成本，可以使用 SR-IOV 构建设备以支持单个 PF 和多个 VF，如 Figure 9-5 所示（即 PF 的硬件资源远远大于 VF

的硬件资源)。

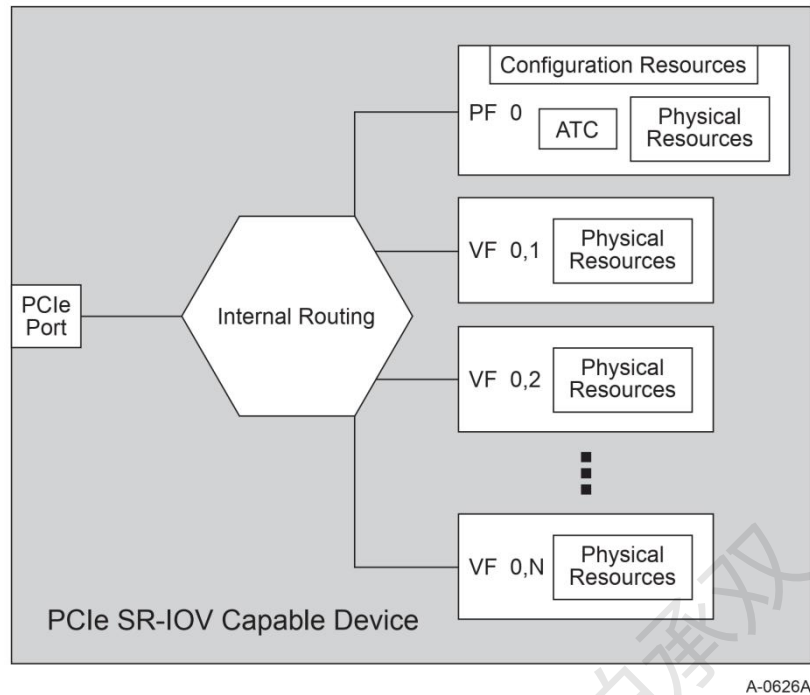


Figure 9-5 Example SR-IOV Single PF Capable Device

Figure 9-5 中的示例说明了具有 N 个 VF 的单个 PF。需要注意的主要观察结果：

- PF 是 PCIe 兼容的。
 - 最初且在常规复位之后，支持本规范中定义的 SR-IOV Capability 的 PCIe Function 应禁用 SR-IOV Capability。
 - 要发现 PF 及其关联的 VF 支持的页面大小，请读取 Supported Page Size 配置字段。有关如何使用此字段在系统页面边界上对齐 PF 或 VF 内存空间 hole 的更多信息，请参见 Section 9.2.1.1.1。
- PF 的命名法为：PF M，即 PF 的 Function Number 为 M。
- VF 的命名法为：VF M, N，表示与 PF M 相关的第 N 个 VF。VF 的编号从 1 开始，因此与 PF M 相关的第一个 VF 为 VF M, 1。
- 每个 VF 与 PF 共享许多公共配置空间字段。（即，这些字段适用于所有 VF 并通过单个 PF 控制。共享减少了实现每个 VF 的硬件资源要求。）
 - VF 使用与 PF 相同的配置机制和 header type。
 - 与给定 PF 相关的所有 VF 共享一个 VF BAR 集（请参见 Section 9.3.3.14），并共享 SR-IOV Extended Capability 中的一个 VF Memory Space Enable（MSE）字段（请参见 Section 9.3.3.3.4），它用于控制 VF 的存储空间访问能力。也就是说，如果 VF MSE 位为 0，则将禁用为所有 VF 分配的内存映射空间。
 - InitialVFs 和 TotalVFs 字段（请参见 Section 9.3.3.5 和 Section 9.3.3.6）用于发现可以与 PF 关联的 VF 的最大数量。
 - 如果设备不支持 VF 迁移，则 TotalVF 和 InitialVF 必须包含相同的值。如果设备支持 VF 迁移，则在读取 TotalVF 时，PF 必须返回可以分配给 PF 的 VF 数量。对于这样的设备，当读取 InitialVF 时，PF 必须返回分配给 PF 的 VF 的初始数量。
- 每个 Function，PF 和 VF 均分配有唯一的 Routing ID。每个 PF 的 Routing ID 是根据 Section 2.2.4.2 构造的。每个 VF 的 Routing ID 是使用其对应的 PF 的 Routing ID 和该 PF 的 SR-IOV Extended Capability 中的字段确定的。
- 假定所有 PCIe 和 SR-IOV 配置访问都是通过可信软件组件（如 VI 或 SR-PCIM）进行的。

- 每个 VF 都包含特定服务所需的一组非共享物理资源（例如，工作队列，数据缓冲区等资源）。SI 可以直接访问这些资源，而无需 VI 或 SR-PCIM 干涉。
- 一个或多个 VF 可以分配给每个 SI。分配策略不在本规范的范围之内。
- 尽管此示例说明了 PF 中的单个 ATC，但是否实现 ATC 是可选的。另外，本规范并不排除设备内每个 VF 都实现 ATC。
- Internal Routing 是特定于实现的。
- 尽管存在关于 PF 操作的许多潜在使用模型，但常见的使用模型是使用 PF 严格在 VI 的控制下引导设备或平台。一旦配置了 SR-IOV Extended Capability，就可以将 VF 分配给各个 SI，则 PF 将承担更多的监督作用。例如，PF 可用于管理设备特定的功能，例如对每个 VF 的内部资源分配，对共享资源（例如 PCIe 链路或功能特定的链路，例如网络或存储链路）的 VF 仲裁等。这些策略，管理和资源分配操作不在本规范的范围之内。

Figure 9-6 给出了另一个示例使用模型。在此示例中，设备支持多个 PF，每个 PF 都有自己的一组 VF。

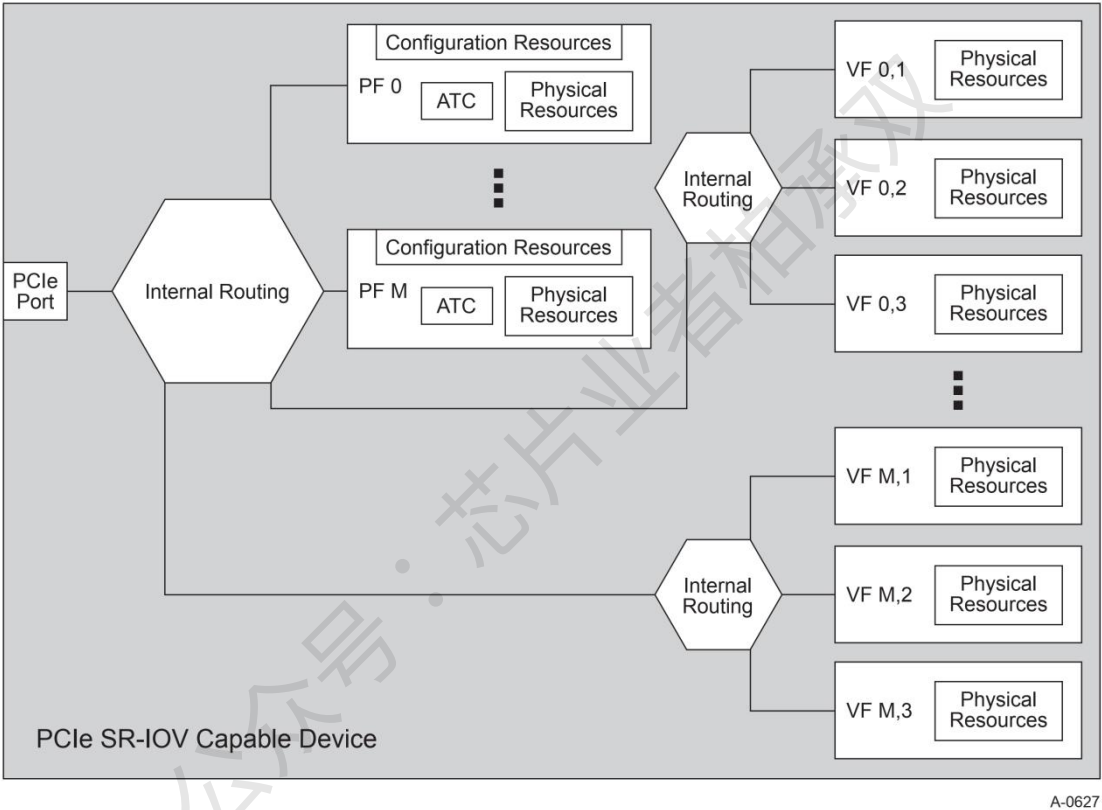


Figure 9-6 Example SR-IOV Multi-PF Capable Device

需要注意的主要观察结果：

- 可以为每个 PF 分配零个或多个 VF。对于设备内的所有 PF，每个 PF 的 VF 数量不必相同。
 - ARI Extended Capability 使 Function 可以分配给 Function Group，并定义如何配置 Function Group 仲裁。PF 和 VF 可以分配给 Function Group，并利用相关的仲裁功能。但是，在每个 Function Group 中，仲裁仍然是特定于实现的。
 - PF 和 VF 之间的 Internal Routing 是特定于实现的。
 - 对于某些使用模型，所有 PF 可能是相同的设备类型。（例如，所有 PF 为相同的网络设备或全部为相同的存储设备。）对于其他使用模型，每个 PF 可能代表不同的设备类型；（例如，在 Figure 9-6 中，一个 PF 可能代表网络设备，而另一个 PF 代表加密设备。）
- 在设备类型之间存在使用模型依赖性的情况下，例如对于每个 VF 是网络设备类型，每个 SI 都还需要一个 VF，它是加

密设备类型。SR-IOV Extended Capability 提供了一种指示这些依赖关系的方法。用于构造这些依赖关系以及将 VF 依赖关系集分配给给定的 SI 的策略不在本规范的范围之内。

从前面的示例中可以看出，PF 和 VF 的数量可以根据使用模型要求而变化。为了支持各种选项，SR-IOV 设备可以支持以下数量以及 PF 和 VF 的混合：

- 使用 Alternative Routing Identifier (ARI) capability，设备最多可以支持 256 个 PF。Function Number 分配是特定于实现的，并且在整个 256 个 Function Number 空间中可能采用稀疏方法。
- PF 只能与设备捕获的 Bus Number 相对应，如 Figure 9-7 所示。
- SR-IOV 设备可能消耗多个 Bus Number。VF 可以与设备的 Bus Number 范围内的任何 Bus Number 相关联——捕获的 Bus Number 加上软件配置的任何其他 Bus Number。有关详细信息，请参见 Section 9.2.1.2。
 - 使用多个 Bus Number 可使设备支持非常多的 VF——多达 Routing ID 空间的大小减去用于标识中间总线的位的个数。
 - 如果软件未配置足够的附加 Bus Number，则为附加 Bus Number 实现的 VF 可能不可见。

Function Co-location

ARI Extended Capability 使设备能够支持多达 256 个与捕获的 Bus Number 相关联的 Function——Function，PF 或 VF 的任意组合。如果使用模型不需要多于 256 个 Function，则强烈建议实现将所有 Function，PF 和 VF 共同放置在捕获的 Bus Number 内，并且不需要其他 Bus Number 即可访问 VF。

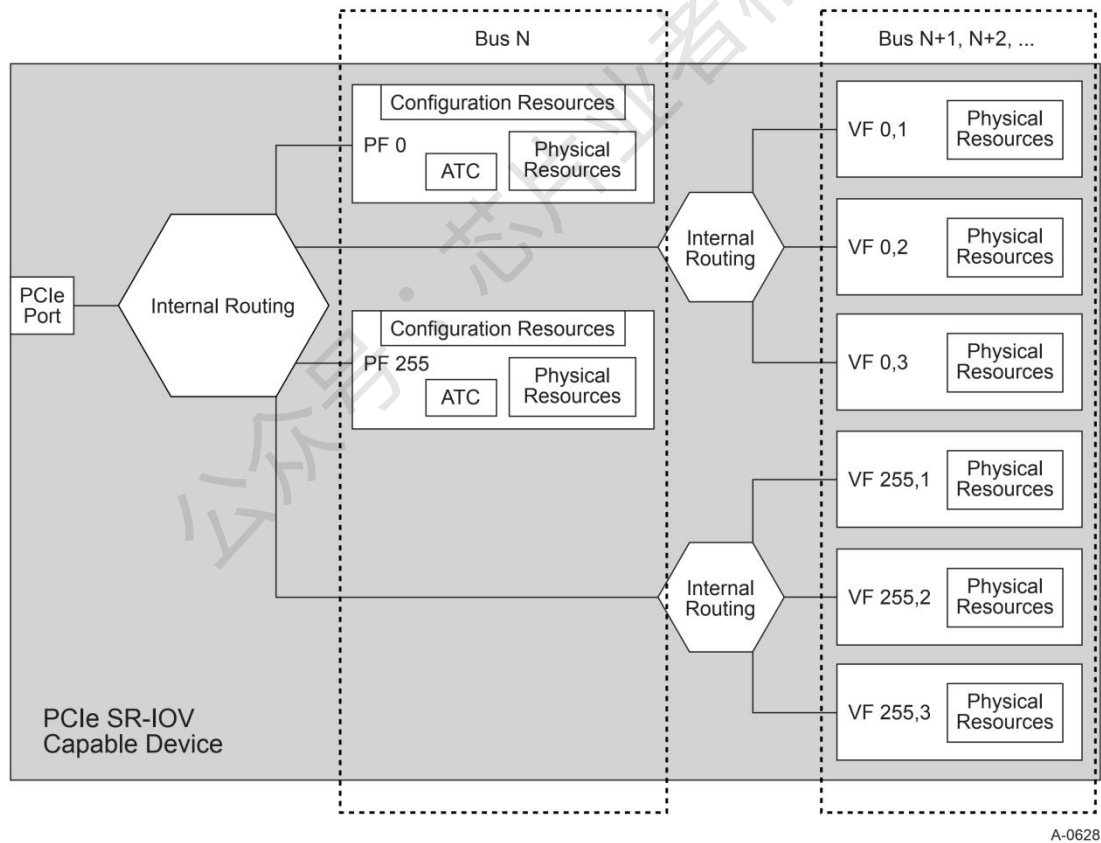


Figure 9-7 Example SR-IOV Device with Multiple Bus Numbers

在 Figure 9-8 最后一个示例中，设备实现可以混合任意数量的 Function，PF 和 VF。

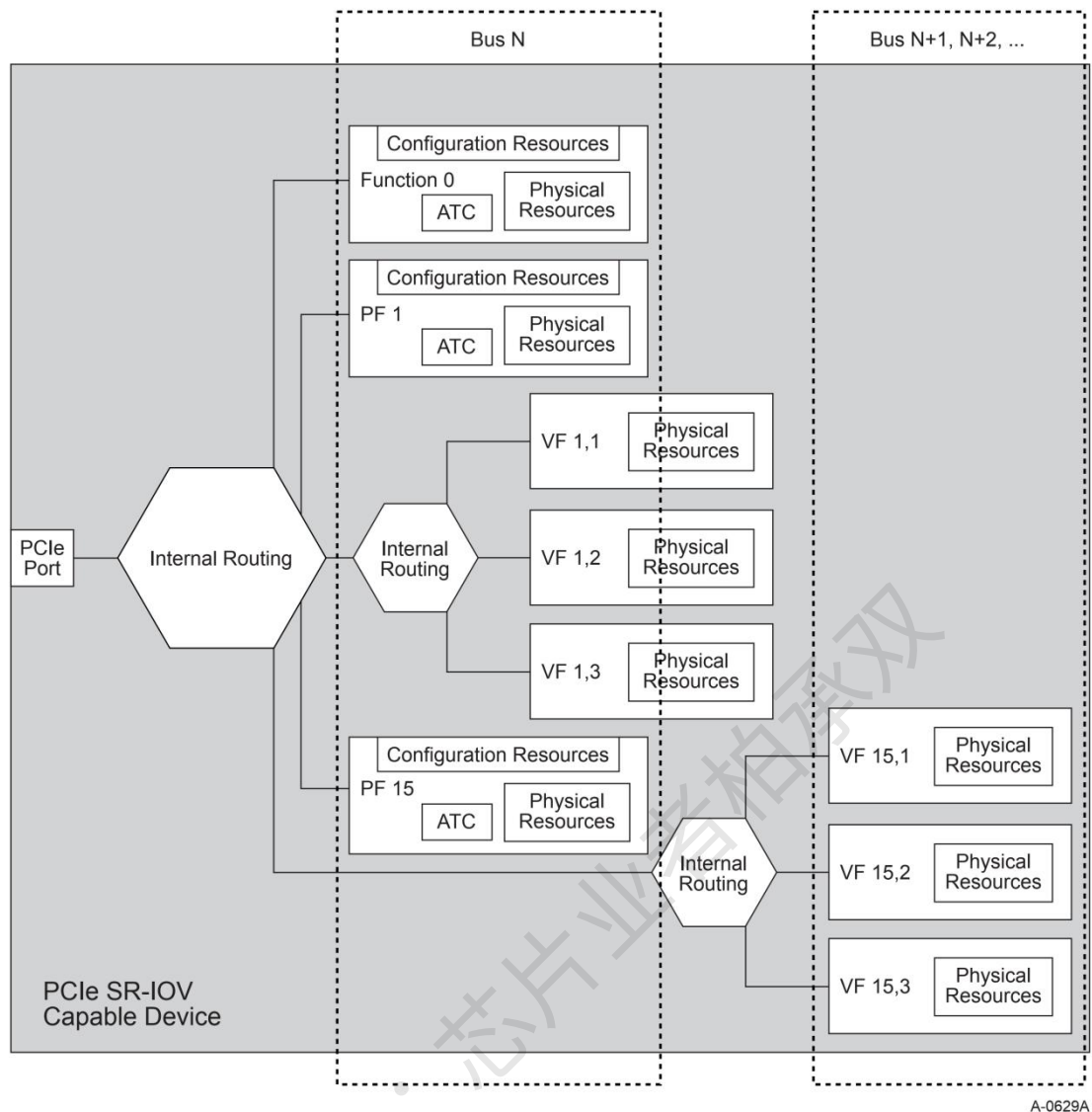


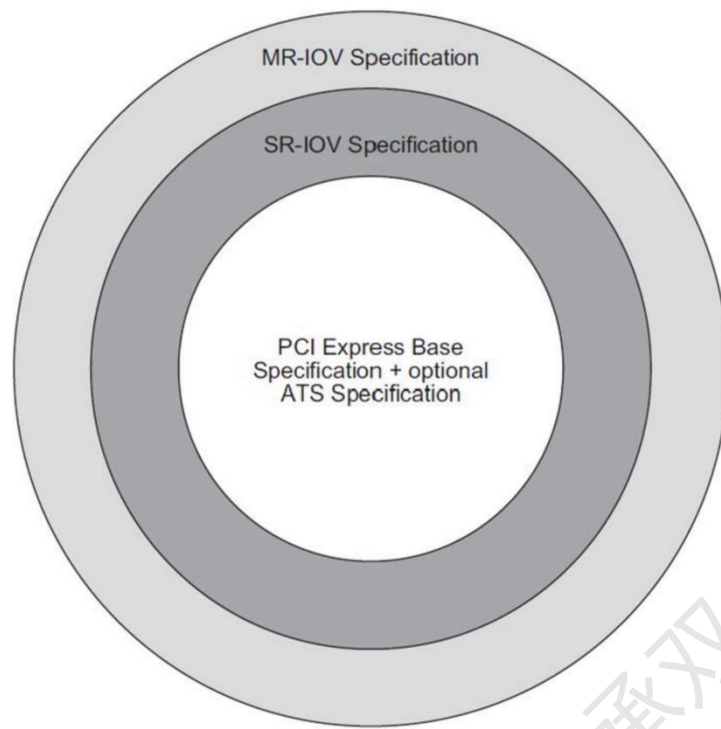
Figure 9-8 Example SR-IOV Device with a Mixture of Function Types

需要注意的主要观察结果：

- 每个设备必须包含 Function 0。Function 0 可以是 PF（即，它可以实现 SR-IOV 扩展功能）。
- 任何 Function 组合都可以与捕获的 Bus Number 关联。
 - Non-VF 只能与捕获的 Bus Number 关联。
- 如果支持 ARI Extended Capability，则可以将 Function 分配给 Function Group。分配策略超出了本规范的范围。如果不支持 ARI Extended Capability，Function 仍可以使用 Section 6.3.3.4 中定义的 Function 仲裁功能。

9.1.1 PCI Technologies Interoperability

建立明确的互操作性要求对于任何技术的成功都是至关重要的。为此，组织了 PCI-SIG I/O 虚拟化规范，以最大程度地提高一致性实现的互操作性。从概念上讲，这被视为一组同心圆，它们定义了如何分层功能以构建具有 IOV 功能的组件，如 Figure 9-9 所示。



A-0630

Figure 9-9 I/O Virtualization Interoperability

需要注意的主要观察结果：

- I/O 虚拟化扩展的核心是基于此规范。所有 IOV 实现都必须符合[PCIe-1.1]或更高版本。如果适用，IOV 规范会记录这些版本之间的相关增量。
 - IOV 规范均未涉及到物理层。
 - SR-IOV 不会违反此规范中定义的数据链路或事务层。
 - [MR-IOV]不涉及本规范中指定的事务层。
 - 所有 I/O 虚拟化功能都通过在 PCI Express 扩展配置空间中实现的新 PCI Express Capability 进行通信。
 - I/O 虚拟化规范对 PCI 或 PCI-X 规范没有影响。
 - 层次结构可以由 PCI Express 和 PCI Express-to-PCI/PCI-X Bridge 的混合组成。
 - ◆ 一个 PCI Express-to-PCI/PCI-X Bridge 和 PCI/PCI-X 设备可以由多个 SI 串行共享。
- ATS 定义了适用于任何 Function 的可选功能。SR-IOV 组件中可以支持 ATS。
- 要实现 SR-IOV 设备，SR-IOV 要求设备完全兼容[PCIe]。
 - 层次结构可以由 SR-IOV 和非 SR-IOV 组件的混合组成。例如，层次结构可以包含 SR-IOV 和非 SR-IOV Endpoint 设备的任意组合。

9.2 SR-IOV Initialization and Resource Allocation

9.2.1 SR-IOV Resource Discovery

以下各节描述软件如何确定设备具有 SR-IOV 能力，并随后通过 Virtual Function Configuration Space 识别 VF 资源。

9.2.1.1 Configuring SR-IOV Capabilities

本部分描述了使能 PF 的 IOV Capability 之前必须配置的字段。通过将 SR-IOV Extended Capability 中的 PF 的 VF Enable 字段置 1 (请参阅 Section 9.3.3.3.1) 来启用 VF。

NumVFs 字段 (请参阅 Section 9.3.3.7) 定义在对应的 PF 中设置 VF Enable 时启用的 VF 的数量。

9.2.1.1.1 Configuring the VF BAR Mechanisms

本节介绍如何将 VF BAR 配置到映射内存空间。VF 不支持 I/O 空间，因此 VF BAR 不应映射到 I/O 空间。

System Page Size 字段 (请参阅 Section 9.3.3.13) 定义启用 PF 的 IOV Capability 时系统将用来映射 VF 的 PCIe 内存地址的页面大小。PF 使用 System Page Size 字段将每个 VF BAR 定义的内存空间 aperture 与系统页面边界对齐。为系统页面大小选择的值必须是 SR-IOV Extended Capability 中支持的页面大小之一 (请参见 Section 9.3.3.12)。

VF BAR 的行为与普通 PCI 内存空间 BAR 相同 (请参见 Section 7.5.1.2.1)，不同之处在于 VF BAR 描述了每个 VF 的 aperture，而 PCI BAR 描述了单个 Function 的 aperture。VF BAR 中某些字段的属性受 VF Resizable BAR Extended Capability 的影响 (请参见 Section 9.3.7.5)。

- Section 7.5.1.2.1 中描述的用于确定 Function 的 BAR 的存储范围的行为适用于每个 VF BAR。即，可以通过写入所有“1”然后读取 VF BAR 来确定每个 VF BAR 所需的存储孔的大小。回读的结果必须按照 Section 7.5.1.2.1 中的说明进行解释。
- 分配与第一个 VF 关联的每个 BAR 的起始存储空间地址的行为也如 Section 7.5.1.2.1 所述。即，设备将写入每个 VF BAR 的地址用作第一个 VF 的起始地址。
- 在 Section 7.5.1.2.1 中描述的 VF BAR 和 BAR 之间的区别在于，对于每个 VF BAR，与第二个和更高 VF 相关联的存储空间是从第一个 VF 的起始地址和存储空间孔径得出的。对于任何给定的 VFv，根据以下公式计算其对于任何已实现的 BARb) 的内存空间孔径的起始地址：

$$\text{BARb VFv 的起始地址} = \text{VF BARb} + (v - 1) \times (\text{VF BARb aperture size})$$

其中 VF BARb aperture size 是 VF BARb 的大小，由 Section 9.3.3.14 中描述的常规 BAR 探测算法确定。

在 VF Enable 和 VF MSE 设置为 1 之前，不会启用 VF 存储空间 (请参见 Section 9.3.3.3.1 和 Section 9.3.3.3.4)。请注意，更改 System Page Size (请参阅 Section 9.3.3.13) 可能会影响 VF BAR 的孔径大小。

Figure 9-10 显示了 PF 和 VF 内存空间孔的示例。

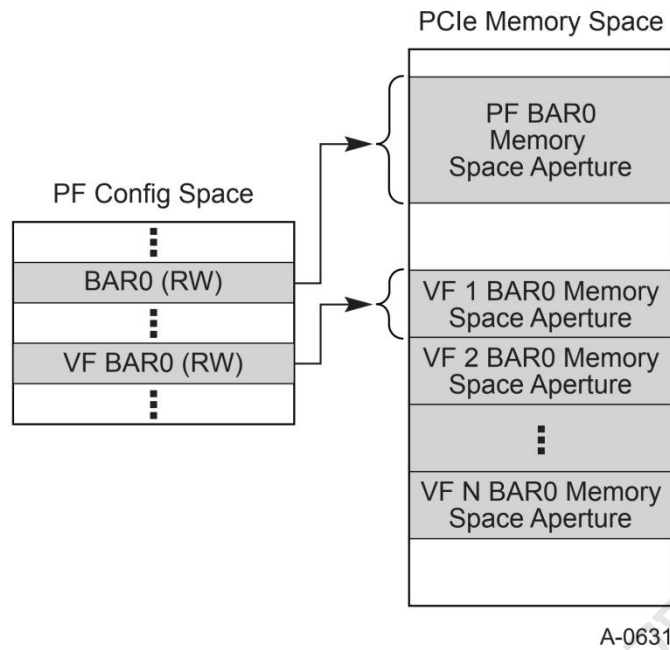


Figure 9-10 BAR Space Example for Single BAR Device

9.2.1.2 VF Discovery

SR-IOV Extended Capability 中的 First VF Offset 和 VF Stride 字段是 16 位的 Routing ID 偏移。这些偏移量用于计算具有以下限制的 VF 的 Routing ID：

- PF 中的 NumVF 值（Section 9.3.3.7）可能会影响该 PF 的 First VF Offset（Section 9.3.3.9）和 VF Stride（Section 9.3.3.10）的值。
- 设备编号最低的 PF（例如 PF0）中 ARI Capable Hierarchy 中的值（Section 9.3.3.3.5）可能会影响设备所有 PF 中的 First VF Offset 和 VF Stride 的值。
- 仅当该 PF 的 VF Enable（Section 9.3.3.3.1）为 0 时，才可以更改 PF 的 NumVF。
- 仅当设备的所有 PF 中的 VF Enable 清 0 时，才可以更改 ARI Capable Hierarchy（Section 9.3.3.3.5）。

NumVFs and ARI Capable Hierarchy

在配置了 NumVF 和 ARI Capable Hierarchy 之后，软件可能会读取 First VF Offset 和 VF Stride，以确定 PF 的 VF 将消耗多少 Bus Number。除非 VF Enable 设置为 1，否则不会实际使用额外的 Bus Number。

Table 9-1 描述了用于确定与每个 VF 相对应的 Routing ID 的算法。

Table 9-1 VF Routing ID Algorithm

VF Number	VF Routing ID
VF 1	$(\text{PF Routing ID} + \text{First VF Offset}) \text{ Modulo } 2^{16}$
VF 2	$(\text{PF Routing ID} + \text{First VF Offset} + \text{VF Stride}) \text{ Modulo } 2^{16}$
VF 3	$(\text{PF Routing ID} + \text{First VF Offset} + 2 * \text{VF Stride}) \text{ Modulo } 2^{16}$
...	...
VF N	$(\text{PF Routing ID} + \text{First VF Offset} + (N-1) * \text{VF Stride}) \text{ Modulo } 2^{16}$
...	...
VF NumVFs (last one)	$(\text{PF Routing ID} + \text{First VF Offset} + (\text{NumVFs}-1) * \text{VF Stride}) \text{ Modulo } 2^{16}$

此 Routing ID 计算中使用的所有算法都是 16 位无符号丢弃所有进位。

所有 VF 和 PF 必须具有不同的 Routing ID。任何 PF 或 VF 的 Routing ID 不得与设备的所有 PF 中的 NumVF 的任何有效设置的其他 PF 或 VF 的 Routing ID 重叠。

VF Stride 和 First VF Offset 是常数。除本节前面所述外，它们的值可能不受设备此 Function 或其他 Function 中设置的影响。

VF 可能与关联的 PF 驻留在不同的 Bus Number 上。例如，如果 First VF Offset 的值为 0100h，则会发生这种情况。VF 不应位于在数字上小于其关联 PF 的 Bus Number 上。位于与其关联的 PF 相同的 Bus Number 上的 VF 不应位于在数字上小于 PF 的 Device Number 上¹⁵⁸。

注 158：紧邻下游端口的 SR-IOV 设备始终具有 Device Number 0，因此始终满足此条件。

SR-IOV RCiEP 设备的 VF 与相同的 Root Complex Event Collector 与其 PF 相关联。此类 VF 不在 Root Complex Event Collector 的 Root Complex Event Collector Endpoint Association Extended Capability 中报告。

根据 Section 2.2.6.2，与上游端口关联的具有 SR-IOV Capability 的设备会从任何 Type 0 配置写入请求中获取 Bus Number。具有 SR-IOV Capability 的设备不会从任何 Type 1 配置写入请求中捕获 Bus Number。具有 SR-IOV Capability 的 RCiEP 使用特定于实现的机制来分配其 Bus Number。

Switch 处理 Bus Number 与基本 PCIe 相同。在基本 PCIe 中，Switch 将 Secondary Bus Number（包括）到 Subordinate Bus Number（包括）中的所有配置请求发送到设备。寻址 Secondary Bus Number 的 Type 1 配置请求将转换为 Type 0 配置请求，而寻址 Secondary Bus Number（不包括）到 Subordinate Bus Number（包括）之间的总线号的 Type 1 配置请求将作为 Type 1 配置请求转发到设备。

注意：Bus Number 是受限制的资源。强烈建议设备避免在 Bus Number 使用中留下“hole”，以免浪费 Bus Number。

所有 PF 必须位于设备捕获的 Bus Number 上。

VFs Spanning Multiple Bus Numbers

例如，考虑一个支持单个 PF 的 SR-IOV 设备。最初，仅 PF 0 可见。软件设置 ARI Capable Hierarchy。根据 SR-IOV Extended Capability，可以确定：InitialVF 为 600，First VF Offset 为 1，VF Stride 为 1。

- 如果软件将 NumVF 设置在[0...255]范围内，则设备将使用单个 Bus Number。
- 如果软件将 NumVF 设置在[256...511]范围内，则设备将使用两个 Bus Number。
- 如果软件将 NumVF 设置在[512...600]范围内，则设备将使用三个 Bus Number。

PF 0 和 VF 0,1 到 VF 0,255 始终位于第一个（捕获的）Bus Number 上。VF 0,256 至 VF 0,511 始终位于第二个 Bus Number 上（捕获

的 Bus Number 加 1)。VF 0,512 至 VF 0,600 始终位于第三个 Bus Number 上 (捕获的 Bus Number 加 2)。

软件应配置 Switch 的 Secondary 和 Subordinate Bus Number 字段, 以将足够的 Bus Number 路由到设备。如果没有足够的 Bus Number, 则软件应在不启用 SR-IOV 的情况下通过不启用 SR-IOV 和/或减少设备的部分或全部 PF 的 NumVF 来降低设备的 Bus Number 要求。

在某些 PF n 中设置了 VF Enable 后, 设备必须启用 VF n,1 到 VF n,m (含), 其中 m 是 InitialVF 和 NumVF 中的较小者。接收到针对位于捕获的 Bus Number 上的已启用 VF 的 Type 0 配置请求的设备必须正常处理该请求。接收到目标 Type 1 配置请求的设备 (其目标不是位于捕获的总线号上的已启用 VF 的设备) 必须正常处理该请求。接收到针对其捕获的总线号的 Type 1 配置请求的设备必须遵循处理不支持的请求的规则。此外, 如果设置了 VF MSE, 则每个已启用的 VF 必须响应 PCIe 内存事务, 以寻址与该 VF 相关联的内存空间。

PCI Express 结构中不存在未启用的 Function (即, 用于 m 以上的 VF 的 Function)。根据 Section 2.3.1, 寻址不存在的 Function 将导致返回不支持的请求 (UR)。这包括附加总线号上的 Function。

Multi-Function Devices with PFs and Switch Functions

SR-IOV 设备可能消耗多个 Bus Number。除第一个 Bus Number 以外的其他 Bus Number 是连续的, 并紧跟在分配给设备的第一个 Bus Number 之后。如果 SR-IOV 设备还包含 PCI-PCI 桥 (具有 Type 1 配置空间头), 则在为这些桥编程 Secondary Bus Number 时必须考虑 SR-IOV 使用情况。软件应首先确定 VF 使用的最后一个 Bus Number, 然后配置任何位于同一位置的桥以使用高于该值的 Bus Number。

9.2.1.3 Function Dependency Lists

PCI 设备在 Function 之间可能具有特定于供应商的依赖关系。例如, Function 0 和 1 可能会提供不同的机制来控制相同的基础硬件。在这种情况下, 设备编程模型可能要求将这些有依赖关系的 Function 作为一组分配给 SI。

Function Dependency List 用于描述这些 Function 间依赖关系 (或指示没有 Function 依赖关系)。软件应将 PF 和 VF 分配给 SI, 以便满足依赖性。

有关详细信息, 请参见 Section 9.3.3.8。

9.2.1.4 Interrupt Resource Allocation

如果分配了中断资源, 则 PF 和 VF 支持 MSI, MSI-X 中断或两者都支持。VF 不应实现 INTx。中断在 Section 9.5 中描述。

9.2.2 SR-IOV Reset Mechanisms

本节介绍 PCI 定义的复位机制如何影响支持 SR-IOV 的设备。它还描述了用于复位单个 VF 和单个 PF 及其关联的 VF 的机制。

9.2.2.1 SR-IOV Conventional Reset

对支持 SR-IOV 的设备的常规复位应使所有 Function (包括 PF 和 VF) 都按照 Section 6.6.1 中的规则复位到其原始的开机状态。Section 9.3 描述了所定义字段的行为。

注意: 常规复位会清除 PF 中的 VF Enable。因此, 常规复位后, VF 不再存在。

9.2.2.2 FLR That Targets a VF

VF 必须支持 Function Level Reset (FLR)。

注意：软件可以使用 FLR 复位 VF。对 VF 执行 FLR 会影响 VF 的状态，但不影响其在 PCI 配置空间或 PCI 总线地址空间中的存在。PF 的 SR-IOV Extended Capability 中的 VF 的 BAR 的值（请参阅第 9.3.3.14 节）和 VF MSE (Section 9.3.3.3.4)，以及 VF Resizable BAR capability 的值（Section 9.3.7.5）不会受到针对 VF 的 FLR 的影响。

9.2.2.3 FLR That Targets a PF

PF 必须支持 FLR。

针对 PF 的 FLR 会重置 PF 状态以及 SR-IOV Extended Capability，包括 VF Enable，这意味着 VF 不再存在。

9.2.3 IOV Re-initialization and Reallocation

如果 VF Enable 设置为 1 后清 0，则与 PF 关联的所有 VF 将不再存在，并且不再必须发出 PCIe 事务或响应配置空间或内存空间访问。清除 VF Enable（包括粘性位）后，VF 不得保留任何状态。

9.2.4 VF Migration (VF 迁移)

VF 迁移是[MR-IOV]中的可选 feature。不支持 MR-IOV Extended Capability 的设备不支持 VF 迁移功能。

在 Multi-Root 系统中，可以在 Virtual Hierarchy 之间迁移 VF。

要进行 VF 迁移，必须同时设置 VF Migration Capable 和 VF Migration Enable。VF 迁移功能向 SR-PCIM 指示存在 VF 迁移硬件，并且 Multi-Root PCI Manager (MR-PCIM) 已启用它。VF 迁移的硬件支持是可选的。SR-PCIM 对 VF 迁移的支持也是可选的。

VF Migration Enable 向设备硬件和 MR-PCIM 指示 SR-PCIM 也已启用 VF 迁移。

支持 VF 迁移对 SR-PCIM 提出以下要求：

- 需要确定 VF 是处于 Active（活动状态）、Dormant（休眠状态）或 Inactive（非活动状态）。Single Root (SR)可以使用处于活动状态的 VF。SR 可以配置为休眠的 VF，但不会发出事务。SR 无法使用处于非活动状态的 VF。
- 需要参与 Migrate In 操作。Migrate In 用于向 SR 提供 VF。由 MR-PCIM 启动 Migrate In。SR-PCIM 可以接受 Migrate In 操作，并将 VF 移到 Active.Available 状态。
- 需要参与 Migrate Out 操作。Migrate Out 操作用于请求从 SR 正常删除活动的 VF。SR-PCIM 可以接受 Migrate Out 并将 VF 移至 Inactive.Unavailable 状态。
- 需要处理 Migrate In Retraction。这是当 MR-PCIM 撤消了向 SR 提供 VF 的迁移，并且 VF 转移回 Inactive.Unavailable 状态。

9.2.4.1 Initial VF State

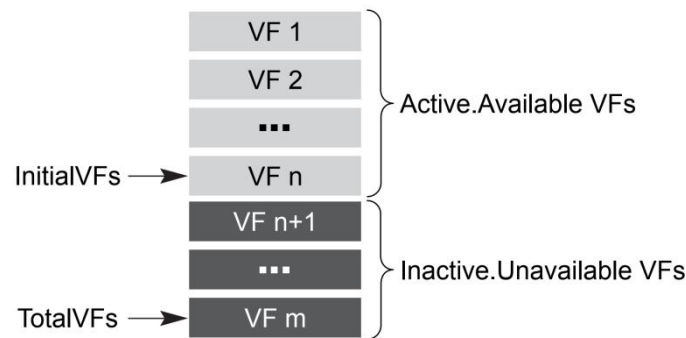
本节描述了 VF Migration State Array 中的初始值（请参见 Section 9.3.3.15.1）。

如果 InitialVF (Section 9.3.3.5) 不为零，则 VF1 至 VFInitialVF 处于 Active.Available 状态。如果 TotalVF (Section 9.3.3.6) 大于 InitialVF，则 VFInitialVF + 1 至 VFTotalVF 处于 Inactive.Unavailable 状态。如果 VF Migration Enable 清 0 (Section 9.3.3.3.2)，则不使用高于 InitialVF

的 VF。

如果 InitialVF 为 0，则没有任何 VF 处于 Active.Available 状态。如果 TotalVF 等于 InitialVF，则所有 VF 都处于 Active.Available 状态。
如果 TotalVF 为 0，则没有任何 VF 与该 PF 关联，并且没有 VF Migration State Array。

Figure 9-11 描述了此初始 VF 状态。

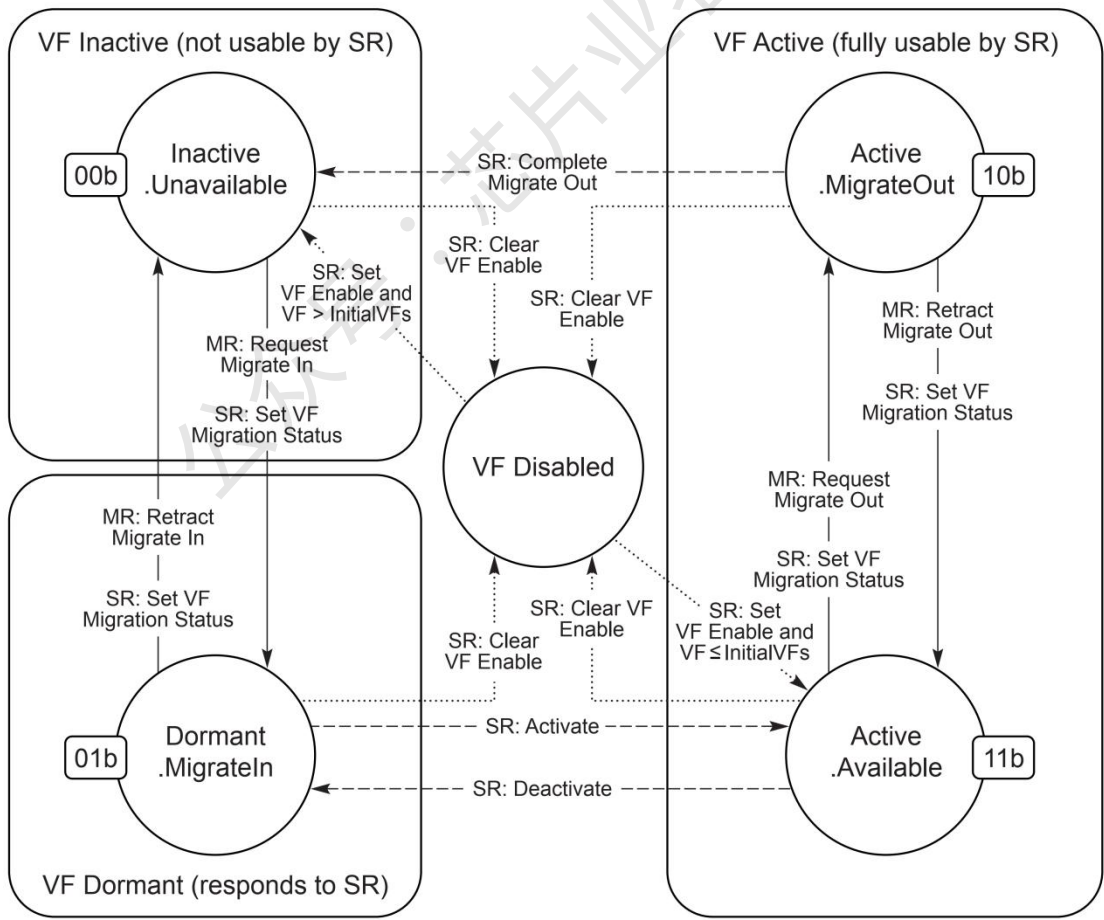


A-0632

Figure 9-11 Initial VF Migration State Array

9. 2. 4. 2 VF Migration State Transitions

VF 迁移遵循 Figure 9-12 所示的状态图。显示的状态值包含在与 VF 相关联的 VF State array entry 中。实线表示的状态转换由 MR-PCIM 启动。虚线表示的状态转换由 SR-PCIM 启动。



A-0633

Figure 9-12 VF Migration State Diagram

SR-PCIM 通过将新值写入 VF Migration State Array 来启动状态转换。当 SR-PCIM 尝试启动 Table 9-2 以外的任何状态转换时，设备将忽略写事务，并且不会发生状态转换。

Table 9-2 SR-IOV VF Migration State Table

Current VF Enable	Current VF State	Written VF Enable	Written VF State	Meaning
1	Dormant.MigrateIn	1	Active.Available	SR Activate VF.
1	Active.Available	1	Dormant.MigrateIn	SR Deactivate VF.
1	Active.MigrateOut	1	Inactive.Unavailable	SR Complete Migrate Out.
1	Any	0	Any	SR Disables all VFs.
0	Any	1	Any	SR Enables all VFs. VFs transition to either Active.Available or Inactive.Unavailable based on the VF number and InitialVFs.

Software State Migration Change Detection

SR-PCIM 通常需要在写入状态后通过重新读取 VF Migration State 来验证状态更改是否生效。

处于 Inactive.Unavailable 的 VF 不能以任何方式被软件使用。目标为 Inactive VF 的请求将收到 Unsupported Request (UR)。在转换为非活动或休眠状态的 100 毫秒内，设备必须确保不会使用指示的 Routing ID 发出新的事务。

MR-PCIM 通过使用[MR-IOV]中指定的不同数据结构来启动状态转换。这种转换的效果在 VF Migration State Array 和 VF Migration Status 中可见。由 MR-PCIM 发起的所有状态转换都会导致 VF Migration Status 被置 1。

每个支持 VF 迁移的 VF 都存在此迁移状态机。迁移状态机不受 Function Dependency List 的影响 (请参见 Section 9.2.1.3 和 Section 9.3.3.8)。

VF Migration Status 不影响 Function 的状态。如果在迁移和/或迁移操作中需要重置 VF 状态，则 SR-PCIM 必须发出 FLR 来完成此操作。未实现 FLR 的 VF 迁移时的 VF 行为是不确定的。

FLR and VF Migration

从系统 A 到系统 B 的 VF 迁移通常会在完成 MigrateOut 操作之前涉及系统 A 中的一个 FLR，在接受 MigrateIn 操作之后但使用 VF 之前，涉及系统 B 中的第二个 FLR。系统 A 使用第一个 FLR 来确保没有任何数据泄漏。系统 B 使用第二个 FLR 来确保它以干净的状态开始。

9.3 Configuration

9.3.1 SR-IOV Configuration Overview

本节提供了实现 PF 和 VF 的 SR-IOV 添加的要求。

9.3.2 Configuration Space

9.3.3 SR-IOV Extended Capability

Figure 9-13 显示了 SR-IOV Extended Capability structure。

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	Byte Offset
PCI Express Extended Capability Header																																+000h
SR-IOV Capabilities Register (RO)																																+004h
SR-IOV Status Register																SR-IOV Control Register (RW)																+008h
TotalVFs (RO)																InitialVFs (RO)																+00Ch
RsvdP								Fcn Dep Link (RO)								NumVFs (RW)																+010h
VF Stride (RO)																FirstVF Offset (RO)																+014h
VF Device ID (RO)																RsvdP																+018h
Supported Page Sizes (RO)																																+01Ch
System Page Size (RW)																																+020h
VF BAR0 (RW)																																+024h
VF BAR1 (RW)																																+028h
VF BAR2 (RW)																																+02Ch
VF BAR3 (RW)																																+030h
VF BAR4 (RW)																																+034h
VF BAR5 (RW)																																+038h
VF Migration State Array Offset (RO)																																+03Ch

886

9.3.3.1 SR-IOV Extended Capability Header (Offset 00h)

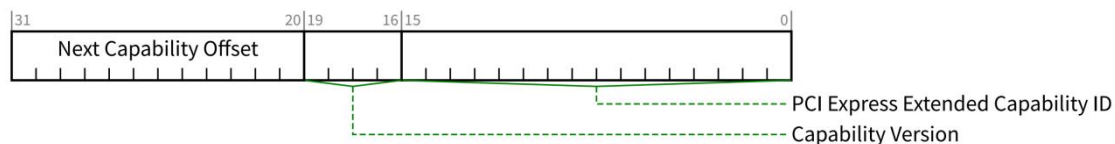


Figure 9-14 SR-IOV Extended Capability Header

Bit Location	Register Description	Attributes
15:0	PCI Express Extended Capability ID - 该字段是 PCI-SIG 定义的 ID 号，值为 0010h，表示这是一个 SR-IOV Extended Capability。	RO
19:16	Capability Version - 该字段是 PCI-SIG 定义版本号，指示存在的功能结构的版本。 此版本该值必须为 1h。	RO
31:20	Next Capability Offset - 此字段指向下一个 PCI Express Capability structure 的偏移量；如果在 Capability 的链表中不存在其他项，则为 000h。	RO

9.3.3.2 SR-IOV Capabilities Register (04h)

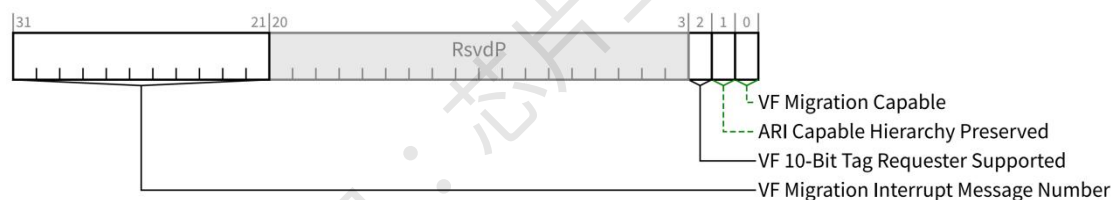


Figure 9-15 SR-IOV Capabilities Register

Bit Location	Register Description	Attributes
0	VF Migration Capable - 如果设置为 1，则 PF 具有迁移的能力，并在具有 Migration Capabl 的 MR-PCIM 下运行。	RO
1	ARI Capable Hierarchy Preserved - 对于 PCI Express Endpoint： 如果置 1，则在某些电源状态转换期间会保留 ARI Capable Hierarchy 字段的值。 对于 RCIEP： 不适用 —— 强烈建议将该位硬连线到 0b。	RO
2	VF 10-Bit Tag Requester Supported - 如果置 1，则所有 VF 必须支持 10-Bit Tag Requester capability。	HwInit

如果清 0，则 VF 必须不支持 10-Bit Tag Requester capability。

如果 PF 的 Device Capabilities 2 寄存器中的 10-Bit Tag Requester Supported 字段为 0，则该位必须为 0。

31:20	VF Migration Interrupt Message Number - 指示用于迁移中断的 MSI/MSI-X 向量。 如果 VF Migration Capable 为 0，则此字段中的值是不确定的。	RO
-------	--	----

9.3.3.2.1 VF Migration Capable

VF Migration Capable 设置为 1 表示 PF 支持 VF 迁移。如果清 0，则 PF 不支持 VF 迁移。

VF 迁移是[MR-IOV]的可选功能。如果 PF 没有实现用于 VF 迁移的硬件，则该比特应被实现为 RO 的 0b。如果 PF 实现了用于 VF 迁移的硬件，则该位由[MR-IOV]中定义的机制控制，并指示是否存在对 VF 迁移的支持。

仅实现 SR-IOV 的设备应将此字段实现为 RO 的 0b 值。

支持 VF 迁移的 PF 必须实现 MSI 或 MSI-X 中断（或两者都实现）。

9.3.3.2.2 ARI Capable Hierarchy Preserved

ARI Capable Hierarchy Preserved 设置为 1 表示 PF 在某些电源状态转换中保留 ARI Capable Hierarchy 中的字段（请参见 Section 9.3.3.3.5）。组件必须设置该位或设置 No_Soft_Reset 位（请参见 Section 9.6.2 节）。即使组件设置了 No_Soft_Reset，也建议组件设置该位。

仅在设备的编号最小的 PF（例如 PF0）中实现 ARI Capable Hierarchy Preserved。在设备的其他 PF 中，ARI Capable Hierarchy Preserved 为只读且为 0。

ARI Capable Hierarchy Preserved 不适用于 RCiEP，并且其值未定义（请参见 Section 9.3.3.3）。

9.3.3.2.3 VF 10-Bit Tag Requester Supported

如果 PF 支持 10-Bit Tag Requester capability，则允许其关联的 VF 也支持 10-Bit Tag Requester capability，但这是可选的。特别是对于将大量突发业务流同时分布在多个 VF 上的使用模型，单个 VF 可能不必使用 10-Bit Tag，因此它们可以支持 256 个以上的 outstanding Non-Posted Request。

对于给定的 PF，关联的 VF 要么全部支持 10-Bit Tag Requester capability 要么一个都不支持。这避免了不必要的实现和管理复杂性。参见 Section 9.3.5.9。

支持 10-Bit Tag Requester capability 的 VF 除具有其他 Function 类型外，还具有其他要求，以简化错误处理并减少 10-Bit Tag 相关错误的可能性，其中一个 VF 会影响其他业务流。

- 如果 SR-IOV Control 寄存器中的 VF 10-Bit Tag Requester Enable 字段被置 1，则每个 VF 必须为其生成的所有 Non-Posted Request 都使用 10-Bit Tag。
- 对于每个 outstanding 10-Bit Tag Request，如果 VF 收到与 Tag [9:8]为 00b 以外的 outstanding Request 相匹配的 Completion，则 VF 必须生成 Completion Timeout 错误以阻塞该请求（最终），作为处理此错误的替代策略，需要采用特定于设备的机制处理该错误以避免数据损坏。

强烈建议软件不要将 Unexpected Completion 错误配置为不可纠正错误。这样可以避免它们触发系统错误或硬件错误包含机制，

例如 Downstream Port Containment (DPC)。

No VF 10-Bit Tag Completer Supported Bit

没有 VF 10-Bit Tag Completer Supported 字段。如果 PF 支持 10-Bit Tag Completer capability，则要求其所有关联的 VF 支持 Section 9.3.5.9 中所述的 10-Bit Tag Completer capability。这有助于避免某些 Completer 支持 10-Bit Tag capability 而某些不支持的复杂的 PCIe 层次结构。

9. 3. 3. 2. 4 VF Migration Interrupt Message Number

VF Migration Interrupt Message Number 必须包含用于与某些 VF 迁移事件相关联生成的中断消息的 MSI 或 MSI-X 向量号。

对于 MSI，VF Migration Interrupt Message Number 必须指示用于引用本节中描述的某些 SR 事件的 MSI 消息号[0 .. 31]。该 Function 需要更新此字段，以便当软件写 MSI 的 Message Control 寄存器中的 Multiple Message Enable 字段时，分配给该 Function 的 MSI 消息的数量发生更改时，此 Function 是正确的。

对于 MSI-X，VF Migration Interrupt Message Numbe 必须指示用于生成中断消息的 MSI-X 表条目[0 .. 2047]。

如果同时实现了 MSI 和 MSI-X，则允许它们使用不同的向量，尽管软件一次只能启用一种机制。如果同时启用了 MSI 和 MSI-X 中断，则此字段未定义。

如果 VF Migration Capable 为 0，则此字段未定义。

Migration and MSI

如果 PF 实现 MSI，并且软件将 Multiple Message Enable 设置为小于 Multiple Message Capable 的值，则可能会发生 MSI 向量的某些共享。由于可能需要将单个向量同时定向到 SR-PCIM 和 PF 设备驱动程序，因此这可能会创建复杂的软件结构。

9. 3. 3. 3 SR-IOV Control Register (Offset 08h)

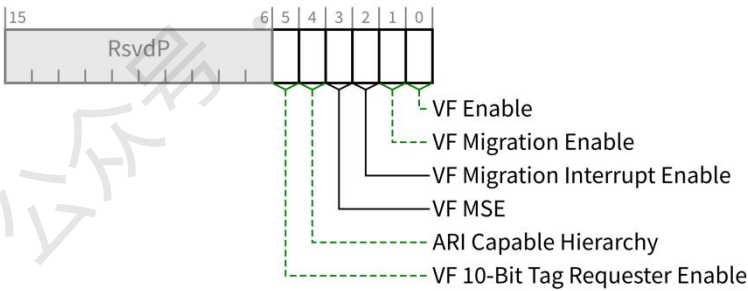


Figure 9-16 SR-IOV Control Register

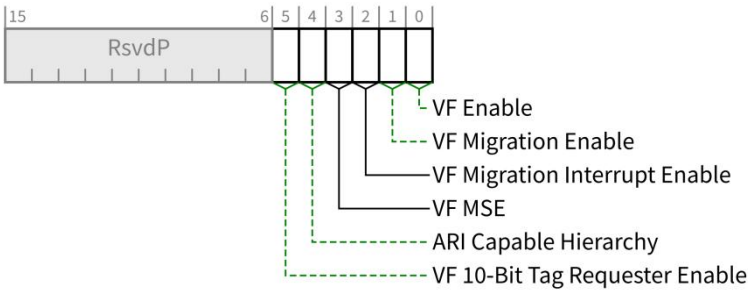


Figure 9-16 SR-IOV Control Register

Bit Location	Register Description	Attributes
0	VF Enable - Enables/Disables VFs. Default value is 0b 。	RW
1	VF Migration Enable - Enables/Disables VF Migration Support. Default value is 0b. See Section 9.3.3.3.2 。	RW or RO
2	VF Migration Interrupt Enable - Enables/Disables VF Migration State Change Interrupt. Default value is 0b 。	RW
3	VF MSE - Memory Space Enable for Virtual Functions. Default value is 0b 。	RW
4	ARI Capable Hierarchy - 对于 PCI Express Endpoint： 在设备的最低编号的 PF 中，该位必须为 RW，而在所有其他 PF 中，该位必须硬连线至 0b。 如果该位的值为 1b，则允许设备在捕获的 Bus Number 的 Function Number 为 8 到 255 中定位 VF。否则，设备必须将 VF 定位为 non-ARI 设备。 该位不受任何 PF 或 VF 的 FLR 影响。 默认值为 0b。 对于 RCiEP： 不适用 —— 强烈建议将该位硬连线到 0b。 在 RC 中，始终允许将 VF 分配给由 First VF Offset 和 VF Stride 规则限定的任何 Function Number（请参见 Section 9.3.3.9 和 Section 9.3.3.10）。	RW or RO
5	VF 10-Bit Tag Requester Enable - 如果置 1，则所有 VF 必须对其生成的所有 Non-Posted Request 使用 10-Bit Tag。如果清 0，则 VF 不得对它们生成的 Non-Posted Request 使用 10-Bit Tag。参见 Section 9.3.3.2.3。 如果在任何 VF 还有 outstanding Non-Posted Request 的时软件更改此位的值，则结果不确定。 如果 SR-IOV Capability 寄存器中的 VF 10-Bit Tag Requester Supported 字段清零，则允许将该位硬接线为 0b。 默认值为 0b。	RW or RO

9.3.3.3.1 VF Enable

VF Enable 管理 VF 到关联 PF 的分配。如果 VF Enable 设置为 1，则可以在 PCI Express 架构中访问与 PF 关联的 VF。置为 1 时，VF 会按照 PCI Express Endpoint Function 的规则进行响应并发出 PCI Express 事务。

如果 VF Enable 为 0，则将禁用 VF，并且在 PCI Express 架构中不可见；向这些 VF 发出的请求将收到 UR，并且这些 VF 不应发出 PCI Express 事务。

为了允许组件执行内部初始化，在 VF Enable 从 0 变为 1 后，不允许系统向由该 VF Enable 使能的 VF 发出请求，直到满足以下条件之一：

- 超过 100ms。
- 已从 PF 接收到 FRS Message，并且该 FRS Message 的 Reason 字段编码值为 VF Enable。
- 至少 VF Enable 时间长度已过。VF Enable 时间长度是（1）与 VF 相关的 Readiness Time Reporting capability 中的 Reset Time 值，或（2）由系统软件/固件确定的值（如 ACPI 表定义）。

RC 和/或系统软件必须在设置 VF Enable 之后至少等待 1.0 s，然后才能确定没有为有效配置请求返回 Successful Completion Status 的 VF 实际上已损坏。设置 VF Enable 后，如果尚未准备好为有效的配置请求提供 Successful 完成状态，则允许由该 VF Enable 使能的 VF 为该配置请求返回 CRS 状态，最长为 1.0 s。PF 发送带有 VF Enable 的 Reason 编码的 FRS Message 后，如果没有 VF Disabled 或其他有效的复位条件的介入，则与该 PF 关联的 VF 不允许返回 CRS。将 Successful Completion 返回给任何请求后，如果没有 VF Disabled 或其他有效的复位条件的介入，则不允许 VF 返回 CRS。

由于 VF 没有 MSE 字段（VF 中的 MSE 由 PF 中 SR-IOV Extended Capability 中的 VF MSE 字段控制），因此软件可以在 VF 准备好处理它之前发出存储请求。因此，在至少满足以下条件之一之前，不得将内存请求发送给 VF：

- VF 已成功响应一个配置请求（不返回 CRS）。
- 向 VF 发出 FLR 后，至少以下一项为真：
 - 从 FLR 发出以后至少过了 1.0 s。
 - VF 支持 FRS，并且在发出 FLR 后，已从 VF 收到 FRS Message，其中 Reason 编码为 FLR Completed。
 - 自发送 FLR 以来，至少已经过了 FLR Time。FLR Time 是（1）与 VF 相关的 Readiness Time Reporting capability 中的 FLR Time 值，或（2）由系统软件/固件确定的值（如 ACPI 表定义）。
- 在 PF 设置 VF Enable 后，至少以下一项为真：
 - VF Enable 置 1 后至少过了 1.0 s。
 - VF 支持 FRS，并且在 VF Enable 置 1 后，已从 VF 收到 FRS Message，其中 Reason 编码为 VF Enable。
 - 自 VF Enable 置 1 以来，至少已经过了 VF Enable Time。VF Enable Time 是（1）与 VF 相关的 Readiness Time Reporting capability 中的 Reset Time 值，或（2）由系统软件/固件确定的值（如 ACPI 表定义）。

在向 VF 发出 FLR 或在相关 PF 的 SR-IOV Extended Capability 中设置 VF Enable 之后，VF 可以静默地丢弃内存请求，直到 VF 成功响应任何请求（不返回 CRS）。

VF Enable 置 0 可以有效地销毁 VF。VF Enable 置 1 可以有效地创建 VF。在 VF Enable 为 0 的情况下将 VF Enable 设置为 1 将产生一组新的 VF。如果 PF 处于 D0 电源状态，则新 VF 处于 D0_{uninitialized} 状态。如果 PF 处于低功耗状态，则行为未定义（见 Section 9.6.1 和 Section 9.6.2）。

VF Enable 清 0 时，支持 FRS 的 PF 应发送 Reason 编码为 VF Disabled 的 FRS Message 指示此操作何时完成。如果 PF 或与 PF 相关的任何 VF 发出了 outstanding Non-Posted Request，则 PF 不允许发送此消息。FRS Message 只能在这些请求完成（或超时）后发送。

VF Enable 清 0 后，SR-IOV Extended Capability 或 VF Migration State Array（Section 9.3.3.15.1）中的任何字段都不能访问，除非：

- VF Enable 清 0 后至少经过 1.0 s。
- PF 支持 FRS 并在 VF Enable 清 0 后从 PF 收到了 Reason 编码为 VF Disabled 的 FRS Message。

Section 9.3.3.7 的 NumVF、Section 9.3.3.5 的 InitialVF、Section 9.3.3.6 的 TotalVF、Section 9.3.3.9 的 First VF Offset、Section 9.3.3.13 的 System Page Size 和 Section 9.3.3.14 的 VF BARx 描述了与此字段相关的其他语义。

9.3.3.3.2 VF Migration Enable

必须将 VF Migration Enable 设置为允许在此 PF 上进行 VF 迁移。详见[MR-IOV]。

如果 VF Migration Capable 设置为 1，则该位为 RW，默认值为 0b。否则，该位为 RO 的零值。

当 VF Enable 设置为 1，此字段为 RO。

9.3.3.3.3 VF Migration Interrupt Enable

每当下列条件中的逻辑 AND 从 False 转换为 True 时，必须发送 MSI 或 MSI-X 中断消息：

- VF 的 Bus Master Enable 设置为 1。
- 对应的中断向量未被 Mask 掉。
- VF Migration Interrupt Enable 设置为 1。
- VF Migration Status 被设置为 1。

使用的 MSI 或 MSI-X 中断向量由 VF Migration Interrupt Message Number 字段指示。

如果 VF Migration Capable 为 0，则此字段未定义。

注：Section 9.2.4 描述了 VF 迁移事件。

9.3.3.3.4 VF MSE (Memory Space Enable)

VF MSE 控制与此 PF 关联的所有活动 VF 的内存空间的使能，就像 Function 的 Command 寄存器中的 Memory Space Enable 字段一样。此位的默认值为 0b。

设置 VF Enable 后，VF 内存空间仅在设置了 VF MSE 后才响应。如果在设置 VF Enable 且 VF MSE 清 0 时尝试访问 Virtual Function 的内存空间，则 VF 应遵循[PCIe]中定义的不同错误报告规则。

VF MSE and VF Enable

当 VF Enable 清 0 时，VF 内存空间将响应 Unsupported Request。因此，当 VF Enable 清 0 时，VF MSE 是 don't care 的；但是，软件可以选择在编程 VF BARn 寄存器之后，在设置 VF Enable 之前设置 VF MSE。

9.3.3.3.5 ARI Capable Hierarchy

对于与上游端口关联的设备，ARI Capable Hierarchy 向设备提示 ARI 已在设备正上方的根端口或 Switch 下游端口中启用。软件应将此位设置为与设备正上方的根端口或 Switch 下游端口中的 ARI Forwarding Enable 字段匹配。

ARI Capable Hierarchy 仅存在于设备的最低编号 PF 中（例如 PF0），并影响设备的所有 PF。ARI Capable Hierarchy 在设备的其他 PF 中为只读的零。

设备可以使用 ARI Capable Hierarchy 的设置来确定 First VF Offset（Section 9.3.3.9）和 VF Stride（Section 9.3.3.10）的值。如果在任

何 PF 中设置了 VF Enable，则更改 ARI Capable Hierarchy 的效果是不确定的。常规复位时，必须将该位设置为其默认值。该位不受任何 PF 或 VF 的 FLR 影响。如果 ARI Capable Hierarchy 设置为 1（Section 9.3.3.2.2）或 No_Soft_Reset 设置为 1（Section 9.6.2），则此 PF 从 D3_{Hot} 到 D0 的电源状态转换不会影响该位的值（Section 9.6.2）。

ARI Capable Hierarchy 不应用于 RCiEP。

ARI Capable Hierarchy

对于与上游端口关联的设备，该设备无法知道是否已启用 ARI。如果启用了 ARI，则设备可以通过将 VF 分配给捕获的 Bus Number 上大于 7 的 Function Number 来节省 Bus Number。ARI 在 Section 6.13 中定义。

由于 RCiEP 与上游端口无关，因此 ARI 不适用，VF 可分配给 First VF Offset 和 VF Stride 允许的根复合体内的任何 Function Number。

9.3.3.4 SR-IOV Status Register (Offset 0Ah)

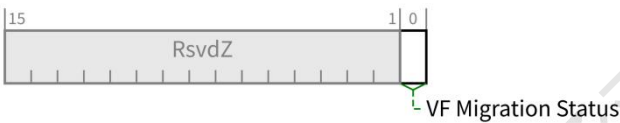


Figure 9-17 SR-IOV Status

Bit Location	Register Description	Attributes
0	VF Migration Status - 表示 MR-PCIM 已发出 Migration In 或 Migration Out 请求。为了确定事件的原因，软件可以扫描 VF State Array。默认值为 0b。	RW1C

9.3.3.4.1 VF Migration Status

当设置了 VF Enable、VF Migration Capable 和 VF Migration Enable 并且 VF Migration State Array 的 entry 发生某些状态更改时，VF Migration Status 会设置为 1（详见 Section 9.2.4 和 Section 9.3.3.15.1）。

VF Migration Status 的设置不受 VF Migration Interrupt Enable 或 MSI 或 MSI-X 的任何控制的影响。

9.3.3.5 InitialVFs (Offset 0Ch)

InitialVF 向 SR-PCIM 指示最初与 PF 关联的 VF 的数量。

InitialVF 的最小值为 0。

对于工作在 Single-Root 的设备，此字段的值为 Hwinit 的，且值与 TotalVF 相等。

对于工作在 Multi-Root 的设备，如果 VF Enable 为 0，此字段的值会随 MR-PCIM 的设置而变化。

注意：[MR-IOV]中描述了 MR-PCIM 用于影响此字段的机制。

如果 VF Migration Enable 设置为 1，并且 VF Enable 从 0 变为 1，则 InitialVF 的值可能会更改。这是必要的，因为某些 VF 可能已迁移到其他 PF，并且对于该 PF 可能不再可用。

9.3.3.6 TotalVFs (Offset 0Eh)

TotalVF 表示可以与 PF 关联的 VF 的最大数量。

TotalVF 的最小值为 0。

对于工作在 Single-Root 的设备，此字段的值为 Hwinit 的，且值与 InitialVF 相等。

对于工作在 Multi-Root 的设备，此字段的值会随 MR-PCIM 的设置而变化。

注意：[MR-IOV]中描述了 MR-PCIM 用于影响此字段的机制。

9.3.3.7 NumVFs (Offset 10h)

NumVF 控制可见的 VF 数量。SR-PCIM 将 NumVF 设置为创建 VF 的过程的一部分。在将两个 NumVF 都设置为有效值并把 VF Enable 设置为 1 之后，该数量的 VF 将在 PCI Express 架构中可见。可见的 VF 具有为其保留的 Function Number，但可能不存在。如果 VF 存在，它将响应针对它们的 PCI Express 事务，并应遵循本规范定义的所有规则。如果存在以下任一情况，则存在 VF：

- VF Migration Capable 为 0，并且 VF 的数量小于或等于 TotalVF。
- VF Migration Capable 为 1，并且关联的 VF 处于 Active.Available 或 Dormant.MigrateIn 状态（Section 9.2.4）。

NumVF 设置为大于 TotalVF 的结果是不确定的。

仅当 VF Enable 为 0 时才能写入 NumVF。如果在 VF Enable 为 1 时写入 NumVF，则结果不确定。

NumVF 的初始值是未定义的。

9.3.3.8 Function Dependency Link (Offset 12h)

设备的编程模型在一组 Function 之间可能具有特定于供应商的依赖性。Function Dependency Link 字段用于描述这些依赖关系。

此字段描述 PF 之间的依赖性。VF 依赖关系与其关联的 PF 的依赖关系相同。

如果一个 PF 独立于设备的其他 PF，则此字段应包含其自己的 Function Number。

如果一个 PF 依赖于设备的其他 PF，则此字段应包含同一 Function Dependency List 中下一个 PF 的 Function Number。Function Dependency List 中的最后一个 PF 应包含 Function Dependency List 中第一个 PF 的 Function Number。

如果 PF_p 和 PF_q 在相同的 Function Dependency List 中，则任何分配了 VF_{p,n} 的 SI 也应分配给 VF_{q,n}。

Function Dependency Link Example

考虑以下场景：

SR-IOV Field	PF 0	PF 1	PF 2
Function Dependency Link	1	0	2
NumVFs	4	4	6
First VF Offset	4	4	4
VF Stride	3	3	3

Function Number	Description	Independent
0	PF 0	No
1	PF 1	No
2	PF 2	Yes
3	Function not present	
4	VF 0,1 (aka PF 0 VF 1)	No
5	VF 1,1 (aka PF 1 VF 1)	No
6	VF 2,1 (aka PF 2 VF 1)	Yes
7	VF 0,2	No
8	VF 1,2	No
9	VF 2,2	Yes
10	VF 0,3	No
11	VF 1,3	No
12	VF 2,3	Yes
13	VF 0,4	No
14	VF 1,4	No
15	VF 2,4	Yes
16 to 17	Functions not present	
18	VF 2,5	Yes
19 to 20	Functions not present	
21	VF 2,6	Yes
22 to 255	Functions not present	

在此示例中，必须将 Function 4 和 5 分配给相同的 SI。同样，Function 7 和 8、10 和 11 以及 13 和 14 必须一起分配。如果将 PF 分配给 SI，则 Function 0 和 1 也必须一起分配。Function 2、6、9、12、15、18 和 21 是独立的，可以以任何方式分配给 SI。

Function Dependency List 中的所有 PF 对于 InitialVF，TotalVF 和 VF Migration Capable 字段应具有相同的值。

SR-PCIM 必须确保在将 Function Dependency List 中的任何 VF 分配给 SI 之前，Function Dependency List 中的所有 PF 的 NumVF，VF Enable 和 VF Migration Enable 字段都具有相同的值。

VF Migration 和 VF Mapping 操作对于每个 VF 独立发生。SR-PCIM 在它分配所有从属的 VF 之前不能将 VF 分配给 SI。例如，使用上述情况，如果 VF 0,2 (Function 7) 和 VF 1,2 (Function 8) 都处于 Inactive.Unavailable 或 Dormant.MigrateIn 状态，则 SR-PCIM 不得将任何 VF 分配给 SI 直到两个 VF 都达到 Active.Available 状态。

同样，SR-PCIM 除非可以删除所有从属的 VF，否则不得从 SI 中删除 VF。例如，使用上述情况，仅当 VF 0,2 和 VF 1,2 都达到 Active.MigrateOut 状态时，才应将它们从 SI 中删除。SR-PCIM 不得将 Function 转换为无效。在 SI 停止使用所有相关 Function 之前，此 Function 不可用。

9.3.3.9 First VF Offset (Offset 14h)

First VF Offset 是一个常数，它定义与包含此 Capability structure 的 PF 关联的第一个 VF 的 Routing ID 的偏移量。第一个 VF 的 16 位 Routing ID 是通过使用无符号 16 位算术将该字段的内容添加到包含该字段的 PF 的 Routing ID（忽略任何进位）来计算的。

VF 不应位于在数字上小于其关联 PF 的 Bus Number 上。

当编号最小的 PF 的 ARI Capable Hierarchy 的值更改或此 PF 的 NumVF 值更改时，此字段的值可能会更改。

注意：如果 NumVF 为 0，则 First VF Offset 未使用。如果 NumVF 大于 0，则 First VF Offset 不得为零。

9.3.3.10 VF Stride (Offset 16h)

VF Stride 为与包含此 Capability structure 的 PF 关联的所有 VF 定义从一个 VF 到下一个 VF 的 Routing ID 偏移。下一个 VF 的 16 位 Routing ID 是通过使用无符号 16 位算术将此字段的内容添加到当前 VF 的 Routing ID 中而忽略任何进位来计算的。

当编号最小的 PF 的 ARI Capable Hierarchy 的值更改或此 PF 的 NumVF 值更改时，此字段的值可能会更改。

注意：如果 NumVF 为 0 或 1，则不使用 VF Stride。如果 NumVF 大于 1，则 VF Stride 不能为零。

9.3.3.11 VF Device ID (Offset 1Ah)

该字段包含应该为每个 VF 提供给 SI 的 Device ID。

VF Device ID 可能不同于 PF Device ID。VF Device ID 必须由供应商管理。供应商必须确保所选的 VF Device ID 不会导致使用不兼容的设备驱动程序。

9.3.3.12 Supported Page Sizes (Offset 1Ch)

该字段指示 PF 支持的 Page Size。如果设置了 Bit n ，则此 PF 支持 2^{n+12} 的页面大小。例如，如果设置了 Bit 0，则 PF 支持 4 KB 页面大小。PF 必须支持 4 KB，8 KB，64 KB，256 KB，1 MB 和 4 MB 页面大小。所有其他页面大小都是可选的。

Page Size 描述了 VF BAR 资源的最低对齐要求，如 Section 9.3.3.13 中所述。

Non-pre-fetch Address Space

非预取地址空间仅限于 4 GB 以下的地址。32 位系统中的预取地址空间也受到限制。强烈建议供应商使用 System Page Size feature 来节省地址空间，同时也支持具有更大页面的系统。

9.3.3.13 System Page Size (Offset 20h)

此字段定义系统将用来映射 VF 的内存地址的页面大小。软件必须将 System Page Size 的值设置为 Supported Page Sizes 字段中设置的页面大小之一（请参阅 Section 9.3.3.12）。与 Supported Page Sizes 一样，如果在 System Page Size 中设置了 Bit n ，则要求与此 PF 关联的 VF 支持 2^{n+12} 的页面大小。例如，如果设置了 Bit 1，则系统正在使用 8 KB 页面大小。如果 System Page Size 为零，则结果不确定。如果在 System Page Size 中设置了多个比特，则结果是不确定的。如果在 System Page Size 中设置了在 Supported Page Sizes 中不支持的字段，则结果是不确定的。

设置 System Page Size 后，与此 PF 关联的 VF 需要以 System Page Size 为边界对齐所有 BAR 资源。每个 VF BAR n 或 VF BAR n pair（请参阅 Section 9.3.3.14）应在 System Page Size 边界上对齐。定义非零地址空间的每个 VF BAR n 或 VF BAR n pair 的大小应确定

为消耗 System Page Size 字节的整数倍。在 VF 中要求页面大小对齐的所有数据结构应在 System Page Size 边界上对齐。

写入 System Page Size 时，VF Enable 必须为零。如果在 VF Enable 为 1 时写 System Page Size，则结果不确定。

默认值为 00000001h（4K）。

9.3.3.14 VF BAR0 (Offset 24h), VF BAR1 (Offset 28h), VF BAR2 (Offset 2Ch), VF BAR3 (Offset 30h), VF BAR4 (Offset 34h), VF BAR5 (Offset 38h)

这些字段定义了 VF 的 Base Address Register (BAR)。这些字段的行为与普通的 PCI BAR 相同，如 Section 7.5.1 所述。可以通过写全 1 并读回 BAR 的内容来确定它们的大小，如 Section 7.5.1.2.1 中所述，并符合定义 BAR 类型字段的低位。

这些字段的属性可能会受到 VF Resizable BAR Extended Capability（见 Section 9.3.7.5）的影响（如果实现）。

每个 BAR 解码的地址空间大小应为 System Page Size 的整数倍。

每个 VF BAR_n，当通过写 1 并读回来确定大小时，每个 BAR 描述了单个 Virtual Function 所消耗的地址空间量和所需的对齐方式。当用实际地址值写入，并设置 VF Enable 和 VF MSE 时，BAR 会映射为 NumVF 个 BAR。换句话说，基址是与该 PF 相关联的第一个 VF BAR 的地址，所有后续 VF BAR 地址范围如下所述。

VF BAR 应仅支持 32 位和 64 位内存空间。VF 不支持 I/O 空间。除用于映射 64 位内存 VF BAR pair 的高 32 位的 VF BAR 外，任何已实现的 VF BAR 的 Bit 0 都必须是只读的 0b。

对齐要求和读取的大小是针对单个 VF 的，但当设置了 VF Enable 和 VF MSE 时，这个 BAR 包含所有（NumVF）VF BAR_n 的基址。

确定 VF BAR_n 映射的地址空间量的算法与标准 BAR 算法不同，如下所示：

1. 如果实现了 VF Resizable BAR Extended Capability（见 Section 9.3.7.5），则可以通过此功能调整 VF BAR 的大小。
2. 在读取低位以确定 BAR 是 32 位 BAR 还是 64 位 BAR pair 后，通过将所有 1 写入 VF BAR_n（或 VF BAR_n 和 VF BAR_{n+1} 用于 64 位 BAR pair）并读回 BAR 或 BAR pair 的内容来确定大小和对齐要求。将读取返回的掩码转换为 Section 7.5.1.2.1 所述的大小和对齐值。该值是单个 VF 的大小和对齐方式。
3. 将 Step 2 中的值乘以 NumVF 中设置的值，以确定在设置 VF Enable 和 VF MSE 后，BAR 或 BAR pair 将映射的总空间量。

对于每个 VF BAR_n 字段，n 对应于其中一个 VF BAR 空间。Table 9-7 显示了 n 和 Function 的 BAR 之间的关系。

Table 9-7 BAR Offsets	
<i>n</i>	BAR Offset in a Type 0 Header
0	10h
1	14h
2	18h
3	1Ch
4	20h
5	24h

System Page Size 更改后，所有 VF BAR_n 寄存器的内容都是不确定的。

9. 3. 3. 15 VF Migration State Array Offset (Offset 3Ch)

如果 VF Migration Capable 设置为 1 (Section 9.3.3.2.1) , 且 TotalVF (Section 9.3.3.6) 不为零, 则该寄存器应包含指向 VF Migration State Array 的 PF BAR 的相对指针。如果 VF Migration Capable 为 0, 则该寄存器为只读的零。VF Migration State Array 的定义见第 9.3.3.15.1。VF Migration State Array Offset 的布局如 Table 9-8 所示。

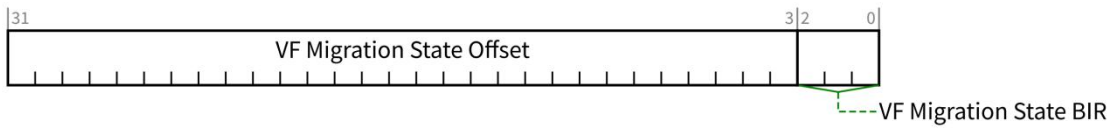


Figure 9-18 VF Migration State Array Offset

Bit Location	Register Description	Attributes																
2:0	VF Migration State BIR - 指示在配置空间中从 10h 开始的 Function 基址寄存器中的哪一个用于将 Function 的 VF Migration State Array 映射到内存空间。 编码值为： <table><tr><td>0</td><td>BAR0 10h</td></tr><tr><td>1</td><td>BAR0 14h</td></tr><tr><td>2</td><td>BAR0 18h</td></tr><tr><td>3</td><td>BAR0 1Ch</td></tr><tr><td>4</td><td>BAR0 20h</td></tr><tr><td>5</td><td>BAR0 24h</td></tr><tr><td>6</td><td>Reserved</td></tr><tr><td>7</td><td>Reserved</td></tr></table> 对于 64 位 BAR，VF Migration State BIR 表示较低的 DW。 如果 TotalVF 为 0，则此字段未定义。	0	BAR0 10h	1	BAR0 14h	2	BAR0 18h	3	BAR0 1Ch	4	BAR0 20h	5	BAR0 24h	6	Reserved	7	Reserved	RO
0	BAR0 10h																	
1	BAR0 14h																	
2	BAR0 18h																	
3	BAR0 1Ch																	
4	BAR0 20h																	
5	BAR0 24h																	
6	Reserved																	
7	Reserved																	
31:3	VF Migration State Offset - 用作从 Function 的基址寄存器中包含的地址的偏移量，以指向 VF Migration State Array 的基址。低 3 比特的 VF Migration State BIR 字段被软件屏蔽（设置为零）以形成 32 位 QW 对齐偏移。	RO																

如果为 VF Migration State Array 映射地址空间的 BAR 也映射了与 VF 迁移不相关的其他可用地址空间, 则在其他地址空间中使用的地址不得与 VF Migration State Array 所在的地址共享任何自然对齐的 8 KB 地址范围。

9. 3. 3. 15. 1 VF Migration State Array

SR-IOV Extended Capability 的 VF Migration State Array Offset 寄存器用于定位 VF Migration State Array。

VF Migration State Array 为每个 VF 都定义一个 VF Migration State Entry。VF Migration State Array 的总大小为 NumVF 个字节。如果

TotalVF 为 0，则 VF Migration State Array 将不存在。Table 9-9 定义了一个 VF Migration State Array entry 的布局。

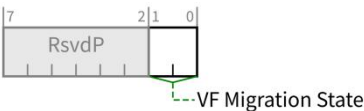


Figure 9-19 VF Migration State Entry

Table 9-9 VF Migration State Entry

Bit Location	Register Description	Attributes
1:0	VF Migration State - State of the associated VF. This field is undefined when VF Enable is Clear. The initial values of the VF Migration State Array are described in Section 9.2.4.1 .	<u>RW</u>

VF Migration State 包含 Table 9-10 中显示的值。

Table 9-10 VF Migration State Descriptions

VF State	VF Exists	Description
00b	No	Inactive.Unavailable - VF does not exist to SR nor is it being migrated in or out.
01b	No	Dormant.MigrateIn - VF is available for use by SR. VF exists but cannot initiate transactions.
10b	Yes	Active.MigrateOut - SR has been requested to relinquish use of the VF.
11b	Yes	Active.Available - Fully functional. Could be assigned to an SI.

VF Migration State Array 的初始值在 Section 9.2.4.1 中描述。在 VF Enable 清 0 后的 1.0 s 内，VF Migration State Array 将返回到 Section 9.2.4.1 中所述的配置。

软件通过将新的状态值写入条目来启动状态转换。

Table 9-11 和 Table 9-12 中列出了有效的状态转换，Figure 9-12 中显示了有效状态转换。SR 只能请求 Table 9-11 中的更改。Table 9-12 中的更改是由 MR-PCIM 使用[MR-IOV]中所述的机制启动的，并且具有此处所述的 SR 可见效果。SR-PCIM 发出的，未在 Table 9-11 中列出的任何转换都将被忽略，并且不会更改 VF 状态。

Table 9-11 SR-PCIM Initiated VF Migration State Transitions

Current State	New State	Change Initiated By	SR Visible Effects of Change
<u>Dormant.MigrateIn</u>	<u>Active.Available</u>	SR-PCIM	VF Activate VF now exists.
<u>Active.Available</u>	<u>Dormant.MigrateIn</u>	SR-PCIM	VF Deactivate VF no longer exists.
<u>Active.MigrateOut</u>	<u>Inactive.Unavailable</u>	SR-PCIM	VF Migrate Out Complete
			VF no longer exists.

Table 9-12 MR-PCIM Initiated VF Migration State Transitions

Current State	New State	Change Initiated By	SR Visible Effects of Change
<u>Active.Available</u>	<u>Active.MigrateOut</u>	MR-PCIM	VF Migrate Out Request VF continues to exist. Sets <u>VF Migration Status</u> .
<u>Inactive.Unavailable</u>	<u>Dormant.MigrateIn</u>	MR-PCIM	VF Migrate In Request VF remains non-existent. Sets <u>VF Migration Status</u> .
<u>Dormant.MigrateIn</u>	<u>Inactive.Unavailable</u>	MR-PCIM	VF Migrate In Retract VF remains non-existent. Sets <u>VF Migration Status</u> .
<u>Active.MigrateOut</u>	<u>Active.Available</u>	MR-PCIM	VF Migrate Out Retract VF continues to exist. Sets <u>VF Migration Status</u> .

9.3.4 PF/VF Configuration Space Header

本部分定义了对 PF 和 VF 配置空间字段的要求。

本章中列出的寄存器定义在现有 PCI-SIG 规范和支持 PCIe SR-IOV 的设备的 PF/VF 定义之间建立了映射。

9.3.4.1 PF/VF Type 0 Configuration Space Header

Figure 9-20 详细列出了 PCI Express Type 0 Configuration Space Header 的寄存器字段。

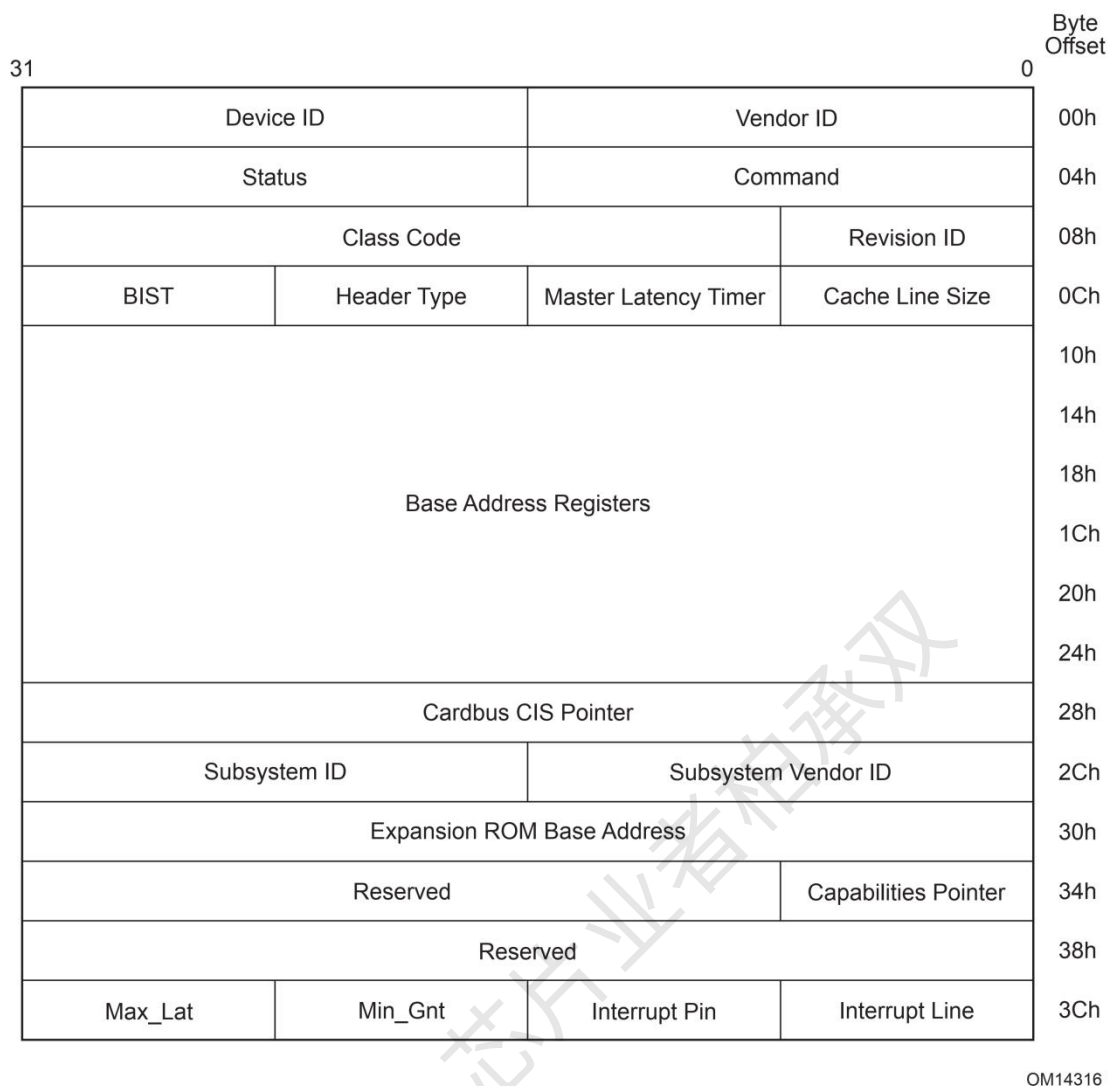


Figure 9-20 PF/VF Type 0 Configuration Space Header

本节中定义了 PCI Express Type 0 Configuration Space Header 的特定于 PCI Express 的寄存器解释。

Error Reporting 字段在 Section 9.4 中有更详细的描述。

9.3.4.1.1 Vendor ID Register Changes (Offset 00h)

该只读寄存器标识设备的制造商。

读取时，所有 VF 中的该字段均返回 FFFFh。VI 软件应把相应的 PF 的 Vendor ID 值作为 VF 的 Vendor ID 值。

9.3.4.1.2 Device ID Register Changes (Offset 02h)

该只读寄存器标识特定的设备。

读取时，所有 VF 中的该字段均返回 FFFFh。VI 软件应把相应的 PF 的 Device ID 值作为 VF 的 Device ID 值。

Legacy PCI Probing Software

为 Device ID 和 Vendor ID 返回 FFFFh 允许某些旧版软件忽略 VF。参见 Section 7.5.1.1.1。

9.3.4.1.3 Command Register Changes (Offset 04h)

PF 和 VF 功能在 Section 7.5.1.1.3 中定义，Table 9-13 中另有说明的地方除外。对于标记为 RsvdP 的 VF 字段，PF 设置适用于 VF。

Table 9-13 Command Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
0	I/O Space Enable - Does not apply to VFs. Must be hardwired to 0b for VFs.	Base	0b
1	Memory Space Enable - Does not apply to VFs. Must be hardwired to 0b for VFs. VF Memory Space is controlled by the VF MSE bit in the VF Control register.	Base	0b
2	Bus Master Enable - Transactions for a VF that has its Bus Master Enable Set must not be blocked by transactions for VFs that have their Bus Master Enable Cleared.	Base	Base
6	Parity Error Response	Base	RsvdP
8	SERR# Enable	Base	RsvdP
10	Interrupt Disable - Does not apply to VFs.	Base	0b

9.3.4.1.4 Status Register Changes (Offset 06h)

PF 和 VF 功能在 Section 7.5.1.1.4 中定义，除了 Table 9-14 描述的内容。

Table 9-14 Status Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
3	Interrupt Status - Does not apply to VFs.	Base	0b

9.3.4.1.5 Revision ID Register Changes (Offset 08h)

该寄存器指定设备特定的修订标识符。

VF 中报告的值可能与 PF 中报告的值不同。

9.3.4.1.6 Class Code Register Changes (Offset 09h)

类代码寄存器为 RO，用于标识设备的通用功能，在某些情况下还用于标识特定的寄存器级编程接口。读取时，PF 和关联的 VF 中的字段必须返回相同的值。

9.3.4.1.7 Cache Line Size Register Changes (Offset 0Ch)

PCI Express 设备将此字段实现为可读写字段，以实现旧版兼容性，但对任何 PCI Express 设备行为没有影响。Physical Function 继续将该字段实现为 RW。对于 Virtual Function，此字段为只读的零。

9.3.4.1.8 Latency Timer Register Changes (Offset 0Dh)

该字段不适用于 PCI Express。该寄存器必须为只读的零。

9.3.4.1.9 Header Type Register Changes (Offset 0Eh)

该字段标识预定义头的第二部分的布局（从配置空间中的字节 10h 开始），还标识设备是否包含多个 Function。该寄存器中的 Bit 7 用于标识一个 Multi-Function Device。对于 SR-IOV 设备，仅当存在多个 Function 时，此寄存器中的 Bit 7 才置 1。VF 不会影响 Bit 7 的值。Bit 6 至 0 标识了预定义头的第二部分的布局。对于 VF，该寄存器必须为只读的零。

9.3.4.1.10 BIST Register Changes (Offset 0Fh)

VF 不支持 BIST，并且必须将此字段定义为只读的零。

如果在设备的任何 PF 中打开了 VF Enable，则软件不得在与该设备关联的任何 Function 中调用 BIST。

9.3.4.1.11 Base Address Registers Register Changes (Offset 10h, 14h, ... 24h)

对于 VF，这些寄存器要实现为只读的零。

Note: See also Section 9.2.1.1.1 and Section 9.3.3.14 .

9.3.4.1.12 Cardbus CIS Pointer Register Changes (Offset 28h)

对于 VF，这些寄存器要实现为只读的零。

9.3.4.1.13 Subsystem Vendor ID Register Changes (Offset 2Ch)

此只读字段标识子系统的制造商。读取时，PF 和关联的 VF 中的字段必须返回相同的值。

9.3.4.1.14 Subsystem ID Register Changes (Offset 2Eh)

此只读字段标识特定子系统。设备的 PF 和 VF 值可能不同。

9.3.4.1.15 Expansion ROM Base Address Register Register Changes (Offset 30h)

Expansion ROM Base Address Register 允许在 PF 中实现。Expansion ROM Base Address Register 在 VF 中为只读的零。VI 可以选择通过仿真为 VF 提供对 Expansion ROM Base Address Register 的访问。

9.3.4.1.16 Capabilities Pointer Register Changes (Offset 34h)

与 Section 7.5.1.1.11 描述的无区别。

9.3.4.1.17 Interrupt Line Register Changes (Offset 3Ch)

此字段不应用于 VF。应实现为只读的零。

9.3.4.1.18 Interrupt Pin Register Changes (Offset 3Dh)

此字段不应用于 VF。应实现为只读的零。

9.3.4.1.19 Min_Gnt Register/Max_Lat Register Changes (Offset 3Eh/3Fh)

这些寄存器不适用于 PCI Express。应实现为只读的零。

9.3.5 PCI Express Capability Changes

PCI Express Capability (请参阅 Section 7.5.3) 用于识别 PCI Express 设备, 并表示支持 PCI Express feature。

Figure 7-21 详细介绍了 PCI Express Capability 中寄存器字段的分配。PF 和 VF 需要实现此 Capability, 但要遵守以下所述的例外情况和

9.3.5.1 PCI Express Capabilities Register Changes (Offset 00h)

Section 7.5.3.2 介绍了 PCI Express Capabilities Register, 其中描述的功能适用于 PF 和 VF。

9.3.5.2 PCI Express Capabilities Register Changes (Offset 02h)

PCI Express Capabilities Register 用于识别 PCI Express 设备类型和相关 Capability。

PF 和 VF 功能在 Section 7.5.3.2 中定义。

9.3.5.3 Device Capabilities Register Changes (Offset 04h)

Device Capabilities Register 标识 PCI Express 设备特定的功能。Figure 7-24 详细说明了 Device Capabilities Register 中寄存器字段的分配; Table 9-15 提供了相应的位定义。

除非 Table 9-15 中另有说明, 否则 PF 和 VF 功能在 Section 7.5.3.3 中定义。

Table 9-15 Device Capabilities Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
4:3	Phantom Functions Supported - When the VF Enable is Set, use of Phantom Function numbers by this PF and associated VFs is not permitted and this field must return 00b when read.	Base	00b
25:18	Captured Slot Power Limit Value	Base	undefined
27:26	Captured Slot Power Limit Scale	Base	undefined
28	Function Level Reset Capability - Required for PFs and for VFs.	1b	1b

9.3.5.4 Device Control Register Changes (Offset 08h)

Device Control Register 控制 PCI Express 设备特定的参数。Figure 7-25 详细说明了 Device Control Register 中的寄存器字段分配; Table

9-16 提供了相应的位定义。

除非 Table 9-16 另有说明，否则 PF 和 VF 功能在 Section 7.5.3.4 中定义。对于标记为 RsvdP 的 VF 字段，PF 设置适用于 VF。

Table 9-16 Device Control Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
0	<u>Correctable Error Reporting Enable</u> See Section 9.4 for details on error reporting.	Base	RsvdP
1	<u>Non-Fatal Error Reporting Enable</u>	Base	RsvdP
Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
	See Section 9.4 for details on error reporting.		
2	<u>Fatal Error Reporting Enable</u> See Section 9.4 for details on error reporting.	Base	RsvdP
3	<u>Unsupported Request Reporting Enable</u> See Section 9.4 for details on error reporting.	Base	RsvdP
4	<u>Enable Relaxed Ordering</u>	Base	RsvdP
7:5	<u>Max_Payload_Size</u>	Base	RsvdP
8	<u>Extended Tag Field Enable</u>	Base	RsvdP
9	<u>Phantom Functions Enable</u>	Base	RsvdP
10	<u>Aux Power PM Enable</u>	Base	RsvdP
11	<u>Enable No Snoop</u>	Base	RsvdP
14:12	<u>Max_Read_Request_Size</u>	Base	RsvdP
15	<u>Initiate Function Level Reset</u> - Required for PFs and for VFs. Note: Setting Initiate Function Level Reset in a PF resets VF Enable which means that VFs no longer exist after the FLR is complete.	Base	Base

9.3.5.5 Device Status Register Changes (Offset 0Ah)

Device Status Register 提供有关 PCI Express 设备特定参数的信息。Figure 7-26 详细说明了 Device Status Register 中的寄存器字段。Table 9-17 提供了相应的位定义。

除非 Table 9-17 另有说明，否则 PF 和 VF 功能在 Section 7.5.3.5 中定义。

Table 9-17 Device Status Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
4	<u>AUX Power Detected</u>	Base	0b
6	<u>Emergency Power Reduction Detected</u>	Base	0b

9.3.5.6 Link Capabilities Register Changes (Offset 0Ch)

Link Capabilities Register 标识 PCI Express 链路特定的功能。Figure 7-27 详细说明了 Link Capabilities Register 中的寄存器字段分配。

PF 和 VF 功能在 Section 7.5.3.6 中定义。

9.3.5.7 Link Control Register Changes (Offset 10h)

Link Control Register 控制 PCI Express 链路的特定参数。Figure 7-28 详细说明了 Link Control Register 中的寄存器字段。

PF 和 VF 功能在 Section 7.5.3.7 中定义，除非 Table 9-18 另有说明。对于标记为 RsvdP 的 VF 字段，PF 设置适用于 VF。

Table 9-18 Link Control Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
1:0	Active State Power Management (ASPM) Control	Base	RsvdP
3	Read Completion Boundary (RCB) Must be hardwired to 0b for VFs.	Base	RsvdP
6	Common Clock Configuration	Base	RsvdP
7	Extended Synch	Base	RsvdP
8	Enable Clock Power Management	Base	RsvdP
9	Hardware Autonomous Width Disable	Base	RsvdP

9.3.5.8 Link Status Register Changes (Offset 12h)

Link Status Register 提供有关 PCI Express Link 特定参数的信息。Figure 7-29 详细说明了 Link Status Register 中的寄存器字段。

PF 功能在 Section 7.5.3.8 中定义。对于 VF，此寄存器中的所有字段均为 RsvdZ，并且 PF 设置适用于 VF。

9.3.5.9 Device Capabilities 2 Register Changes (Offset 24h)

PF 和 VF 功能在 Section 7.5.3.15 中定义，除非 Table 9-19 中另有说明。

Table 9-19 Device Capabilities 2 Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
3:0	Completion Timeout Ranges Supported VF value must be identical to PF value.	Base	Base
4	Completion Timeout Disable Supported VF value must be identical to PF value.	Base	Base
6	AtomicOp Routing Supported Not applicable to Endpoints.	RsvdP	RsvdP
7	32-bit AtomicOp Completer Supported	Base	Base

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
	VF value must be identical to PF value.		
8	64-bit AtomicOp Completer Supported VF value must be identical to PF value.	Base	Base
9	128-bit CAS Completer Supported VF value must be identical to PF value.	Base	Base
16	10-Bit Tag Completer Supported VF value must be identical to PF value.	Base	Base
17	10-Bit Tag Requester Supported VF value must equal the value of the VF 10-Bit Tag Requester Supported bit in the SR-IOV Capabilities register. See Section 9.3.3.2.3	Base	Base
25:24	Emergency Power Reduction Supported VF value must be identical to PF value.	Base	Base
26	Emergency Power Reduction Initialization Required VF value must be identical to PF value.	Base	Base

9.3.5.10 Device Control 2 Register Changes (Offset 28h)

PF 和 VF 功能在 Section 7.5.3.16 节中定义，Table 9-20 中另有说明。

Table 9-20 Device Control 2 Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
3:0	Completion Timeout Value The PF value applies to all associated VFs.	Base	RsvdP
4	Completion Timeout Disable The PF value applies to all associated VFs.	Base	RsvdP
6	AtomicOp Requester Enable The PF value applies to all associated VFs.	Base	RsvdP
8	IDO Request Enable The PF value applies to all associated VFs.	Base	RsvdP
9	IDO Completion Enable The PF value applies to all associated VFs.	Base	RsvdP
11	Emergency Power Reduction Request This bit is only present in one Function associated with an Upstream Port. That Function can never be a VF.	Base	RsvdP

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
12	10-Bit Tag Requester Enable The value in the VF 10-Bit Tag Requester Enable bit in the SR-IOV Control register applies to all associated VFs.	Base	RsvdP

9.3.5.11 Device Status 2 Register Changes (Offset 2Ah)

PF 和 VF 功能在 Section 7.5.3.17 中定义。

9.3.5.12 Link Capabilities 2 Register Changes (Offset 2Ch)

PF 和 VF 功能在 Section 7.5.3.18 中定义。

9.3.5.13 Link Control 2 Register Changes (Offset 30h)

PF 和 VF 功能在 Section 7.5.3.19 中定义。

9.3.5.14 Link Status 2 Register Changes (Offset 32h)

PF 和 VF 功能在 Section 7.5.3.20 中定义，除非 Table 9-21 中另有说明。标记为 RsvdZ 的 VF 字段使用关联的 PF 的值。

Table 9-21 Link Status 2 Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
0	Current De-emphasis Level	Base	RsvdZ

9.3.6 PCI Standard Capabilities

Table 9-22 中描述了 PCI Standard Capabilities 的 SR-IOV 使用情况。标有 n/a 的项不适用于 PF 或 VF。

Table 9-22 SR-IOV Usage of PCI Standard Capabilities

Capability ID	Description	PF Attributes	VF Attributes
00h	Null Capability	Base	Base
01h	PCI Power Management Interface	Base	Optional. See Section 9.6 .
02h	AGP	n/a	n/a
03h	VPD	Base	Optional. See Section 9.3.6.1 .
04h	Slot Identification	n/a	n/a

Capability ID	Description	PF Attributes	VF Attributes
05h	MSI	Base	See Section 9.5.1.1 .
06h	CompactPCI Hot Swap	n/a	n/a
07h	PCI-X	n/a	n/a
08h	HyperTransport	n/a	n/a
09h	Vendor-specific	Base	Base
0Ah	Debug Port	Base	Base
0Bh	CompactPCI Central Resource Control	n/a	n/a
0Ch	PCI Hot Plug	Base	n/a
0Dh	PCI Bridge Subsystem ID	n/a	n/a
0Eh	AGP 8x	n/a	n/a
0Fh	Secure Device	n/a	n/a
10h	PCI Express	Base	See Section 9.3.5 .
11h	MSI-X	See Section 9.5.1.2 and Section 9.5.1.3 .	See Section 9.5.1.2 and Section 9.5.1.3 .
12h	Serial ATA Data/Index Configuration	Base	n/a
13h	Advanced Features	n/a	n/a
14h	Enhanced Allocation	Base	Must not implement.
15h	Flattening Portal Bridge (FPB)	n/a	n/a

9.3.6.1 VPD Capability

VPD Capability 在 PCI 中是可选的。在 SR-IOV 中，它仍然是可选的。

实现 VPD Capability 的 VF 和 PF 必须确保通过 VPD Capability 在 VF 和/或 PF 之间没有“数据泄漏”。

9.3.7 PCI Express Extended Capabilities Changes

Table 9-23 中描述了 PCI Express Extended Capabilities 的 SR-IOV 使用情况。标记为 n/a 的项不适用于 PF 或 VF（例如，仅适用于 RC 中存在的功能或仅适用于 Function 0 中的功能）。

Table 9-23 SR-IOV Usage of PCI Express Extended Capabilities

Extended Capability ID	Description	PF Attributes	VF Attributes
0000h	Null Capability	Base	Base
0001h	<u>Advanced Error Reporting Extended Capability (AER)</u>	Base	See Section 9.4.2 .
0002h	<u>Virtual Channel Extended Capability (02h)</u>	Base	Must not implement. See Section 9.3.7.1 .
0003h	<u>Device Serial Number Extended Capability</u>	Base	See Section 9.3.7.2
0004h	<u>Power Budgeting Extended Capability</u>	Base	Must not implement. See Section 9.3.7.3
0005h	<u>Root Complex Link Declaration Extended Capability</u>	n/a	n/a
0006h	<u>Root Complex Internal Link Control Extended Capability</u>	n/a	n/a
0007h	<u>Root Complex Event Collector Endpoint Association Extended Capability</u>	n/a	n/a
0008h	<u>Multi-Function Virtual Channel Extended Capability</u>	Base	Must not implement. See Section 9.3.7.1 .
0009h	<u>Virtual Channel Extended Capability (09h)</u>	Base	Must not implement. See Section 9.3.7.1 .
000Ah	<u>RCRB Header Extended Capability</u>	n/a	n/a
000Bh	<u>Vendor-specific Extended Capability</u>	Base	Base
000Ch	<u>Configuration Access Correlation Extended Capability</u>	n/a	n/a
000Dh	<u>ACS Extended Capability</u>	See Section 9.3.7.6 .	See Section 9.3.7.6 .
000Eh	<u>ARI Extended Capability (ARI)</u>	See Section 9.3.7.7 .	See Section 9.3.7.7 .
000Fh	<u>ATS Extended Capability</u>	See Section 9.3.7.8 .	See Section 9.3.7.8 .
0010h	<u>SR-IOV Extended Capability</u>	See Section 9.3.3 .	Must not implement. See Section 9.3.3
0011h	<u>MR-IOV Extended Capability (MR-IOV)</u>	Must not implement. See Section 9.3.7.9 .	Must not implement. See Section 9.3.7.9 .
0012h	<u>Multicast Extended Capability</u>	See Section 9.3.7.10 .	See Section 9.3.7.10 .
0013h	<u>Page Request Extended Capability (PRI)</u>	See Section 9.3.7.11 .	See Section 9.3.7.11
0014h	Reserved for AMD	Base	Base
0015h	<u>Resizable BAR Extended Capability</u>	Base	Must not implement. See Section 9.3.7.4
0016h	<u>Dynamic Power Allocation Extended Capability (DPA)</u>	See Section 9.3.7.12 .	Must not implement. See Section 9.3.7.12 .

Extended Capability ID	Description	PF Attributes	VF Attributes
0017h	<u>TPH Requester Extended Capability (TPH)</u>	See Section 9.3.7.13 .	See Section 9.3.7.13
0018h	<u>LTR Extended Capability</u>	Base	Must not implement. LTR is controlled using Function 0 which is never a VF.
0019h	<u>Secondary PCI Express Extended Capability</u>	Base	Must not implement. 8.0 GT/s is controlled using Function 0 which is never a VF.
001Ah	<u>PMUX Extended Capability</u>	Base	Must not implement. PMUX is controlled using Function 0 which is never a VF.
001Bh	<u>PASID Extended Capability</u>	Base	See Section 9.3.7.14
001Ch	<u>LN Requester Extended Capability (LNR)</u>	Base	Base
001Dh	<u>DPC Extended Capability</u>	n/a	n/a.
001Eh	<u>L1 PM Substates Extended Capability</u>	Base	Must not implement. L1 PM Substates is controlled using Function 0 which is never a VF.
001Fh	<u>Precision Time Management Extended Capability (PTM)</u>	Base	Must not implement. PTM controls the Port and must not be implemented in a VF.
0020h	<u>PCI Express over M-PHY Extended Capability (M-PCle)</u>	Base	Must not implement. M-PHY is controlled using Function 0 which is never a VF.
0021h	<u>FRS Queueing Extended Capability</u>	n/a	n/a
0022h	<u>Readiness Time Reporting Extended Capability</u>	Base	See Section 9.3.7.15
0023h	<u>Designated vendor-specific Extended Capability</u>	Base	Base
0024h	<u>VF Resizable BAR Extended Capability</u>	See Section 9.3.7.5	Not Implemented. See Section 9.3.7.5
0025h	<u>Data Link Feature Extended Capability</u>	Base	Must not implement.
0026h	<u>Physical Layer 16.0 GT/s Extended Capability</u>	Base	Must not implement.
0027h	<u>Lane Margining at the Receiver Extended Capability</u>	Base	Must not implement.
0028h	<u>Hierarchy ID Extended Capability</u>	Base	Base
0029h	<u>Native PCIe Enclosure Management Extended Capability (NPEM)</u>	Base	Must not implement

9.3.7.1 Virtual Channel/MFVC

VF 使用关联的 PF 的虚通道。因此，VF 不包含任何虚通道功能。

VF 不应包含 MFVC Capability。此 Capability 仅存在于 Function 0 中，而 Function 0 绝不能是 VF。

9.3.7.2 Device Serial Number

PF 中可能实现 Device Serial Number Extended Capability。如果 PF 实现该 Capability，则其值将应用于所有关联的 VF。允许使用 VF，但不建议使用 VF 来实现此 Capability。实现此 Capability 的 VF 必须返回与其关联的 PF 报告的 Device Serial Number 相同的 Device Serial Number。

9.3.7.3 Power Budgeting

PF 中可能实现 Power Budgeting Extended Capability，但是 VF 一定不能实现它。如果 PF 包含该 Capability，则它必须报告涵盖所有关联的 VF 的值。

9.3.7.4 Resizable BAR

PF 可能实现 Resizable BAR Extended Capability。由于 VF 不实现标准 BAR，因此该 Capability 不能在 VF 中存在。PF 的 Resizable BAR 的设置不会影响 SR-IOV Extended Capability 中的任何设置。

9.3.7.5 VF Resizable BAR Extended Capability

VF Resizable BAR Extended Capability 仅允许在实现至少一个 VF BAR 的 PF 中实现，并且会影响 PF 的 VF BAR 的大小和行为。由于 VF 本身并不实现 BAR，因此该 Capability 不能出现在 VF 中。PF 可以同时实现 VF Resizable BAR Extended Capability 和 Resizable BAR Capability，因为每种 Function 独立运行。

VF Resizable BAR Extended Capability 是个可选的 Capability，允许 PF 调整其 VF BAR 的大小。VF Resizable BAR Extended Capability 允许硬件通过 VF Resizable BAR Extended Capability 和 Control 寄存器传达系统可接受的资源大小，以及系统软件通过 VF Resizable BAR Control 寄存器的 VF BAR Size 字段将最佳大小传递回硬件。

硬件立即在相应的 VF BAR 的只读位中反映此大小推断。推断的大小是从 System Page Size 和 VF BAR Size 字段中解码的值中较大的值。硬件必须清除任何从读写更改为只读的位，以便后续读取返回零。在写入 VF BAR Size 字段之前，软件必须清除 SR-IOV Control 寄存器中的 VF MSE 字段。写入 VF BAR Size 字段后，未定义相应 VF BAR 的内容。为了在调整 VF BAR 大小后确保它包含有效地址，系统软件必须重新编程 VF BAR，并设置 VF MSE 字段（除非未分配资源）。

仅当关联的 VF BAR 是 64 位 BAR 时，才允许 VF Resizable BAR Extended Capability 和 Control 寄存器指示以 4 GB 或更大的容量运行的能力。

强烈建议 Function 不要在此 Function 中通告任何支持的 VF BAR size 值，该值要大于分配时将有效利用的空间。

Using the Capability During Resource Allocation

系统软件使用此 Capability 的方式与 Resizable BAR Capability 类似。系统软件必须首先配置 System Page Size 寄存器（请参阅 Section 9.2.1.1.1）。PF 会报告潜在的可用存储器 aperture 大小，并从 VF Resizable BAR Extended Capability 和 Control 寄存器中读取。由于分配的地址空间少于报告的最大大小可能导致性能降低，因此，该软件应分配其可以报告的最大大小。然后，软件将大小写入 Function 的适当 VF BAR 的 VF Resizable BAR Control 寄存器中。然后，基地址被写入 VF BAR。

出于互操作性的原因，硬件可能会将 VF BAR 的默认大小设置为较小的大小，此大小小于 VF Resizable BAR Capability 寄存器中报告的最大大小。不使用此 Capability 来分配资源大小的软件可能会导致资源分配不足，导致资源分配不理想，或者由于无法容纳资源而无法分配资源。建议在资源受限的环境中负责分配资源的系统软件将有限的地址空间适当地分配给所有内存映射的硬件，包括系统 RAM。

VF Resizable BAR Extended Capability structure 是一种 PCI Express Extended Capability，该 Capability 位于 PCI Express Extended Configuration Space 中，即在前 256 个字节以上，如 Figure 9-21 所示。这种结构可以识别和控制具有此 Capability 的 PF。为每个可调整大小的 VF BAR 实现了 Capability 寄存器和 Control 寄存器。由于任何 PF 最多可以实现 6 个 VF BAR，因此 VF Resizable BAR Extended Capability structure 的范围可以从 12 个字节长（对于单个 VF BAR）到 52 个字节长（对于所有 6 个 VF BAR）。

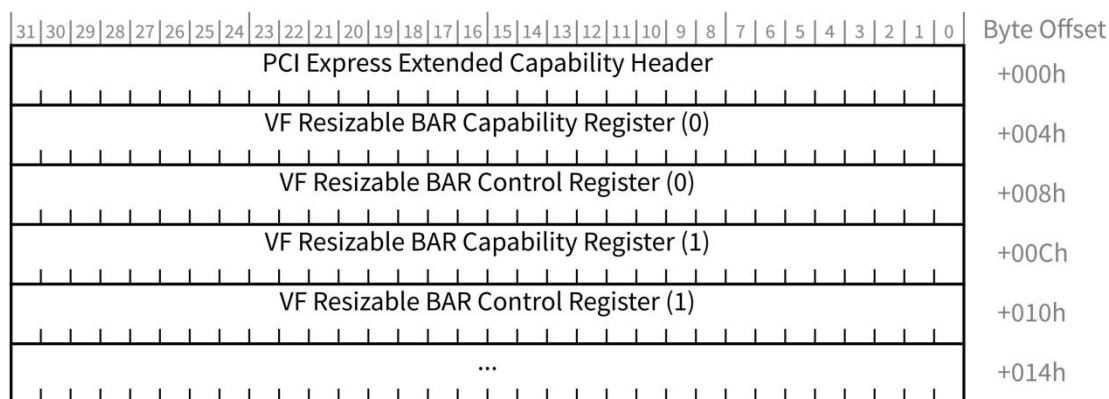


Figure 9-21 VF Resizable BAR Extended Capability

9.3.7.5.1 VF Resizable BAR Extended Capability Header (Offset 00h)

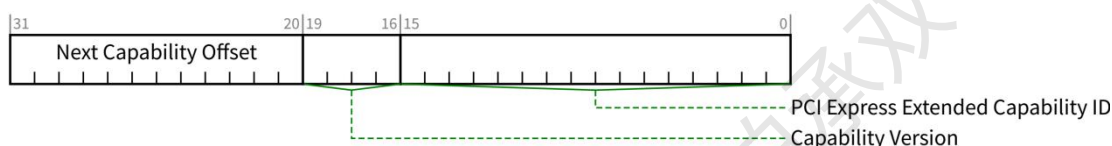


Figure 9-22 VF Resizable BAR Extended Capability Header

Table 9-24 VF Resizable BAR Extended Capability Header

Bit Location	Register Description	Attributes
15:0	PCI Express Extended Capability ID - This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability. PCI Express Extended Capability ID for the VF Resizable BAR Extended Capability is 0024h.	RO
19:16	Capability Version - This field is a PCI-SIG defined version number that indicates the version of the capability structure present. Must be 1h for this version of the specification.	RO
31:20	Next Capability Offset - This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities	RO

9.3.7.5.2 VF Resizable BAR Capability Register (Offset 04h)

VF Resizable BAR Capability 寄存器的字段描述与 Table 7-116 中 Resizable BAR Capability 寄存器的定义相同。这些描述中注明“BAR”的地方，此寄存器的描述则对应“VF BAR”。这些描述中注明“Function”的地方，该寄存器的描述对应“PF”。否则，字段描述，比特位，位置和属性都是相同的。因此，Figure 7-145 同样在该寄存器中分配了寄存器字段。

9.3.7.5.3 VF Resizable BAR Control Register (Offset 08h)

VF Resizable BAR Control 寄存器位 31:16 的定义与 Table 7-117 中的 Resizable BAR Control 寄存器的定义相同。

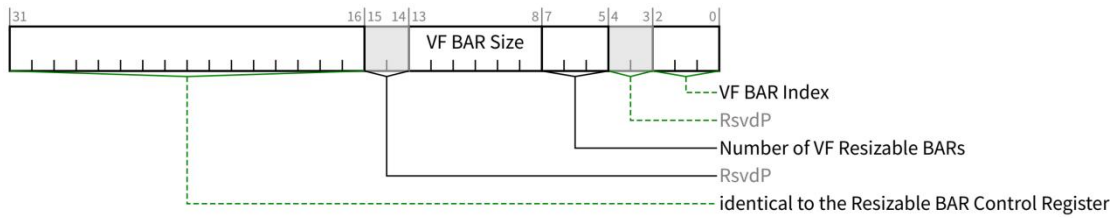


Figure 9-23 VF Resizable BAR Control Register

Bit Location	Register Description	Attributes
--------------	----------------------	------------

2:0 **VF BAR Index** - 该编码值指向位于 SR-IOV Extended Capability 中的特定 VF BAR 的开始。 RO

编码的值为：

- 0** VF BAR located at offset 24h
- 1** VF BAR located at offset 28h
- 2** VF BAR located at offset 2Ch
- 3** VF BAR located at offset 30h
- 4** VF BAR located at offset 34h
- 5** VF BAR located at offset 38h
- others** All other encodings are reserved.

对于 64 位基址寄存器，VF BAR Index 指示较低的 DWORD。

此值指示哪个 VF BAR 支持可协商的大小。

7:5 **Number of VF Resizable BARs** - 指示 Function 的 capability structure 中可调整大小的 VF BAR 的总数。参见 Figure 9-21。 RO/RsvdP

该字段的值必须在 01h 到 06h 的范围内。该字段在 VF Resizable BAR Control 寄存器（0）中有效（偏移量为 08h），其他所有字段均为 RsvdP。

13:8 **VF BAR Size** - 支持的编码值有： RW

- 0** 1 MB (2^{20} bytes)
- 1** 2 MB (2^{21} bytes)
- 2** 4 MB (2^{22} bytes)
- 3** 8 MB (2^{23} bytes)
- ...
- 43** 8 EB (2^{63} bytes)

此字段的默认值等于 VF BAR 资源通过 VF BAR 的只读位请求的地址空间的默认大小。

软件只能写入与 VF Resizable BAR Capability 和 Control 寄存器中支持的值相对应的值。写入不受支持的值得产生不确定的结果。

对该寄存器字段进行编程后，该值将立即反映在资源的大小中，如 VF BAR 中只读位的数量所编码。

31:16 **identical to the Resizable BAR Control Register** - 这些位与 Figure 7-146 中定义的 Resizable BAR Control 寄存器的[31:16]相同。这些描述中注明“BAR”的地方，此寄存器的描述则对应“VF BAR”。这些描述中注明“Function”的地方，该寄存器的描述对应“PF”。参见 Figure 7-14。 See Figure

9.3.7.6 Access Control Services (ACS) Extended Capability Changes

ACS 是可选的扩展功能。如果非 RC 中集成的设备之外的支持 SR-IOV 的设备实现了内部对等事务，则需要 ACS，并具有以下所述的其他要求。

PF 和 VF 功能在 Section 7.7.8.2 中定义，除非 Table 9-26 另有说明。

支持 SR-IOV 的设备 (实现至少一个 PF 的设备) 中的所有功能,除了支持设备内对等事务的 RCiEP 之外,均应实现 ACS Egress Control。

允许但不要求在 RCiEP 中实现 ACS。明确允许在单个 RC 中，某些 RCiEP 实现 ACS，而某些则不实现。强烈建议 RC 实现确保所有源自 RCiEP (PF 和 VF) 且不具有 ACS 功能的访问都先由 RC 中的转换代理 (TA) 进行处理，然后再进行进一步的解码和处理。这种“RC”处理的详细信息不在本规范的范围之内。

Table 9-26 ACS Capability Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
5	ACS P2P Egress Control - Required to be 1b for Functions that are not RCiEPs if peer-to-peer transactions within the Device are supported.	Base	Base

Access Control Services in Systems that Support Direct Assignment of Functions

通用 VI 通常为每个 SI 和 VI 本身实现单独的地址空间。如果此类 VI 还支持将 Function 直接分配给 SI，则直接分配的 Function 发出的 Untranslated Memory Request 事务将在相关 SI 内运行的软件的完全控制下，并且通常引用与该 SI 关联的地址空间。相反，主机 (MMIO 请求) 和未直接分配的 Function 发送的 Memory Request 事务会受 VI 控制，并且通常会引用一个或多个系统地址空间 (例如，PCIe 物理地址空间，与 VI 关联的地址空间，或者与某些特定 SI 相关的地址空间)。通常 VI 不应在这各种地址空间之间建立依赖关系。因此，这些地址空间可以自由地重叠，这可能导致 Switch 错误的进行 TLP 的路由。例如，如果请求中的地址落在与该下游端口相关联的 MMIO 地址区域之内，则由直接分配的 Function 发起并用于主存储器的上游存储器请求 TLP 可以路由到下游端口。这样的意外路由会对 SI 和/或 VI 的稳定性和完整性造成威胁。

为避免这种威胁，强烈建议供应商在支持直接分配 Function 的通用 VI 的平台上实现 ACS。此类支持应包括：

- 在位于 TA 下方的 Switch 或 RC 中，ACS 支持级别应遵循本文档中为实现 ACS Extended Capability 结构的下行 Switch 端口建立的准则。

Note: 位于 TA 上方的组件只能看到 Translated Memory Request，因此，此问题不适用于那些组件。

- 在能够进行对等事务的 SR-IOV 设备中，需要 ACS 支持。
- 在能够进行对等事务的 Multi-Function Device 中，强烈建议供应商使用 ACS P2P Egress Control 来实现 ACS。

此外，平台供应商应测试是否存在 ACS，并在直接分配该 Function 之前，在从 TA 到 Function 的路径上的 RC 和 Switch 从中启用 ACS。如果 Function 具有点对点的能力，则也应在 Function 中启用 ACS。

9.3.7.7 Alternative Routing ID Interpretation Extended Capability (ARI) Changes

ARI 不适用于 RCiEP；所有其他支持 SR-IOV 的设备 (包括至少一个 PF 的设备) 应在每个 Function 中实现 ARI Extended Capability。PF 和 VF 的相关功能在 Section 7.8.7.2 中定义，除非 Table 9-27 另有说明。

Table 9-27 ARI Capability Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
0	MFVC Function Groups Capability (M) - Additional requirements described below.	Base	Base
1	ACS Function Groups Capability (A) - Additional requirements described below.	Base	Base
15:8	Next Function Number - VFs are located using First VF Offset (see Section 9.3.3.10) and VF Stride (see Section 9.3.3.11).	Base	Undefined

任何通过可选的 Function Arbitration Table 实现 MFVC Capability 并且消耗多个 Bus Number 的设备都应在 Function 0 中将 MFVC Function Groups Enable (M)字段实现为 1b。

任何使用可选的 Egress Control Vector 实现 ACS Capability 并且消耗多个 Bus Number 的设备都应在 Function 0 中将 ACS Function Groups Enable (A)字段实现为 1b。

9.3.7.8 Address Translation Services Extended Capability Changes (ATS)

在 SR-IOV 设备中实现 ATS 是可选的。如果 VF 实现 ATS Capability，则其关联的 PF 必须实现 ATS Capability。VF 及其关联的 PF 中的 ATS Capability 可以独立启用。

PF 和 VF 的相关功能在 Section 10.5.1 中定义，除非在 Table 9-28 和 Table 9-29 中另有说明。显示为 ATS 的属性与 Address Translation Services (Chapter 10) 中指定的属性相同。

Table 9-28 ATS Capability Register

Bit Location	PF and VF Register Differences From ATS	PF Attributes	VF Attributes
4:0	Invalidate Queue Depth - Hardwired to 0 for VFs. Depth of shared PF input queue.	ATS	RO

Table 9-29 ATS Control Register Changes

Bit Location	PF and VF Register Differences From ATS	PF Attributes	VF Attributes
4:0	Smallest Translation Unit (STU) - Hardwired to 0 for VFs. PF value applies to all VFs.	ATS	RO

ATS 行为在 Address Translation Services (Chapter 10)中指定。以 Function 为目标的 ATS 请求是无效的。ATS 响应将具有与目标 Function 匹配的 Routing ID 字段。每个 Function 都需要实现足够的排队，以确保它可以容纳来自 TA 的最大数量的 outstanding Invalidation Request（使用输入或输出排队）。

但是，与 PF 关联的所有 VF 在 PF 中共享一个输入队列。为了实现 Invalidation 流控制，TA 必须确保对共享 PF 队列（以 PF 及其关联的 VF 为目标）的 outstanding Invalidate Request 的总数不超过 PF Invalidate Queue Depth 字段中的值。

9.3.7.9 MR-IOV Changes

PF 和 VF 不应实现 MR-IOV Capability。

如果存在，则 MR-IOV Capability 包含在组件的其他 Function 中。有关详细信息，请参见[MR-IOV]。

9.3.7.10 Multicast Changes

在 SR-IOV 设备中，多播支持是可选的。如果 VF 实现了 Multicast Capability，则其关联的 PF 必须实现 Multicast Capability。PF 和 VF 相关的功能在 Section 7.9.11 中定义，除非在 Table 9-30，Table 9-31 和 Table 9-32 中有说明。

Table 9-30 Multicast Capability Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
5:0	MC_Max_Group - PF value applies to all VFs.	Base	RsvdP
13:8	MC_Window_Size_Requested - PF value applies to all VFs.	Base	RsvdP
15	MC_ECRC_Regeneration_Supported - Not applicable for Endpoints.	Base	RsvdP

Table 9-31 Multicast Control Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
5:0	MC_Num_Group - PF value applies to all VFs.	Base	RsvdP

Table 9-32 Multicast Base Address Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
5:0	MC_Index_Position - PF value applies to all VFs.	Base	RsvdP
63:12	MC_Base_Address - PF value applies to all VFs.	Base	RsvdP

9.3.7.11 Page Request Interface Changes (PRI)

Page Request Interface 的相关功能在 Address Translation Services (Chapter 10) 中定义。

允许 PF 实现 Page Request Interface。关联的 VF 可以使用 PF 的 Page Request Interface。PF 被允许实现 PF 的 Page Request Interface，而 VF 不应实现它。

即使 Page Request Interface 在 PF 和 VF 之间共享，它也会在 Page Request Message 的 Requester ID 字段中发送发起请求的 Function 的 ID（PF 或 VF），并在 PRG Response Message 的 Destination Device ID 字段中获取发起请求的 Function 的 ID。

9.3.7.12 Dynamic Power Allocation Changes (DPA)

VF 可以不实现 Dynamic Power Allocation Capability。

VF 的电源分配通过 PF 的 DPA Capability（如果已实现）进行管理。

9.3.7.13 TPH Requester Changes (TPH)

TPH Requester Extended Capability 可能同时存在于 VF、PF 中，或者两者都不存在。如果与给定 PF 关联的任何 VF 实现该 Capability，则与该 PF 关联的所有 VF 也必须支持 TPH。

PF 和 VF 中的所有 TPH Requester Extended Capability 字段均按 Section 7.9.13 中的规定操作。

对于 TPH Requester Capability Register（偏移量 04h）中的字段，PF 及其关联的 VF 可能具有不同的值，但是与同一 PF 关联的所有 VF 在所有字段中必须具有相同的值。

9.3.7.14 PASID Changes

允许 Endpoint 设备支持 PASID。代表设备的单个 Function（Function 或 PF）的 PASID 配置也被设备中的所有 VF 使用。允许 PF 实现 PASID Capability，但 VF 不能实现它。

即使在 Function，PF 和 VF 之间共享了 PASID 配置，设备也会在包含 PASID 的 TLP 的 Requester ID 字段中发送发出请求的 Function 的 ID（Function，PF 或 VF）。

9.3.7.15 Readiness Time Reporting Extended Capability Changes

Readiness Time Reporting Extended Capability 可能同时存在于 VF，PF 中，或者两者都不存在。如果与给定 PF 关联的任何 VF 实现该 Capability，则与该相同 PF 关联的所有 VF 也必须支持 Readiness Time Reporting。

Reset Time 字段包含 VF Enable 生效所需的时间（请参见 Section 9.6.1）。

DL_Up Time 字段是 RsvdP。

与同一 PF 相关的所有 VF 应报告相同的时间值。

9.4 SR-IOV Error Handling

SR-IOV 设备使用 Section 6.2 中定义的错误报告机制。定义为 non-function-specific 的错误仅记录在 PF 中。

Section 6.2 定义了两个错误报告范式：baseline Capability 和 Advanced Error Reporting Capability。所有 PCI Express 设备都需要实现 baseline error reporting capability，并定义了最低错误报告要求。Advanced Error Reporting Capability 是可选的，并且定义用于更健壮的错误报告，并使用特定的 PCI Express Capability 结构实现。

9.4.1 Baseline Error Reporting

所有 SR-IOV 设备都必须支持 baseline error reporting capability，并进行一些修改以解决降低成本和实现复杂性的目标。

这些控制字段仅在 PF 中有意义。在决定生成错误消息时，VF 应使用相关 PF 中的错误报告控制字段。

下述字段在 VF 中为 RsvdP：

- Command register (see Section 9.3.4.1.3)
 - SERR# Enable
 - Parity Error Response
- Device Control register (see Section 9.3.5.4)
 - Correctable Reporting Enable
 - Non-Fatal Reporting Enable
 - Fatal Reporting Enable
 - Unsupported Request (UR) Reporting Enable

每个 VF 必须实现一种机制，以提供独立于任何其他 Function 的错误状态。为特定于 Function 的错误提供 SI 隔离是必要的。

每个 VF 中必须实现以下 baseline error reporting status 字段：

- Status register (see Section 9.3.4.1.4)
 - Master Data Parity Error
 - Signaled Target Abort
 - Received Target Abort
 - Received Master Abort
 - Signaled System Error
 - Detected Parity Error
- Device Status register (see Section 9.3.5.5)
 - Correctable Error Detected
 - Non-Fatal Error Detected
 - Fatal Error Detected
 - Unsupported Request Detected

当发错误信令时，每个 VF 应使用其自己的 Routing ID。

9. 4. 2 Advanced Error Reporting

高级错误报告功能是可选的。如果未在 PF 中实现 AER，则一定不能在关联的 VF 中实现它。如果在 PF 中实现了 AER，则在 VF 中是可选的。

Section 6.2.4 将错误分为 Function-specific 和 non-Function-specific。实现 Advanced Error Reporting Capability 的每个 VF 必须针对 Function-specific 的错误维护其自己的错误报告状态。

9. 4. 2. 1 VF Header Log

在 VF 中实现 AER 的设备可以在与单个 PF 关联的 VF 之间共享 Header Log Register。有关详细信息，请参见 Section 9.4.2.10。

PF 的 Header Log Register 独立于其关联的 VF，并且必须使用专用存储空间来实现。

当实现一组低配的 Header Log Register 时，Function 可能没有空间记录与错误相关的 Header。在这种情况下，Function 应按照 Section 6.2.4 的要求更新 Uncorrectable Error Status Register 和 Advanced Error Capabilities and Control Register。但是，当读取 Header Log Register 时，它应返回全 1 以指示溢出情况，并且没有 Header 被记录。

9. 4. 2. 2 Advanced Error Reporting Capability Changes

Figure 7-122 描述了 AER Extended Capability structure。

9.4.2.3 Advanced Error Reporting Extended Capability Header Changes (Offset 00h)

该寄存器包含 PCI Express Extended Capability ID, Capability Version 和 Next Capability Offset。这些字段并无改变, 在 Section 7.8.4.1 中进行描述。

9.4.2.4 Uncorrectable Error Status Register Changes (Offset 04h)

Uncorrectable Error Status Register 指示单个错误的错误检测状态。为 non-Function-specific 定义的错误记录在 PF 中。VF 中仅记录 Function-specific 的错误。

PF 和 VF 的对应功能在 Section 7.8.4.2 节中定义, 除非 Table 9-33 给出 VF 的不同实现。

Table 9-33 Uncorrectable Error Status Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
4	Data Link Protocol Error Status	Base	0b
5	Surprise Down Error Status	Base	0b
13	Flow Control Protocol Error Status	Base	0b
17	Receiver Overflow Status	Base	0b
18	Malformed TLP Status	Base	0b
19	ECRC Error Status	Base	0b

9.4.2.5 Uncorrectable Error Mask Register Changes (Offset 08h)

PF 和 VF 的对应功能在 Section 7.8.4.3 节中定义, 除非 Table 9-34 给出 VF 的不同实现。对于标记为 RsvdP 的 VF 字段, PF 设置适用于 VF。对于标记为 0b 的 VF 字段, 该错误不适用于 VF。

Table 9-34 Uncorrectable Error Mask Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
4	Data Link Protocol Error Mask	Base	0b
5	Surprise Down Error Mask	Base	0b
12	Poisoned TLP Received Mask	Base	RsvdP

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
13	<u>Flow Control Protocol Error Mask</u>	Base	0b
14	<u>Completion Timeout Mask</u>	Base	RsvdP
15	<u>Completer Abort Mask</u>	Base	RsvdP
16	<u>Unexpected Completion Mask</u>	Base	RsvdP
17	<u>Receiver Overflow Mask</u>	Base	0b
18	<u>Malformed TLP Mask</u>	Base	0b
19	<u>ECRC Error Mask</u>	Base	0b
20	<u>Unsupported Request Error Mask</u>	Base	RsvdP
21	<u>ACS Violation Mask</u>	Base	RsvdP

9. 4. 2. 6 Uncorrectable Error Severity Register Changes (Offset 0Ch)

PF 和 VF 的对应功能在 Section 7.8.4.4 节中定义，除非 Table 9-35 给出 VF 的不同实现。对于标记为 RsvdP 的 VF 字段，PF 设置适用于 VF。对于标记为 0b 的 VF 字段，该错误不适用于 VF。

Table 9-35 Uncorrectable Error Severity Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
4	<u>Data Link Protocol Error Severity</u>	Base	0b
5	<u>Surprise Down Error Severity</u>	Base	0b
12	<u>Poisoned TLP Received Severity</u>	Base	RsvdP
13	<u>Flow Control Protocol Error Severity</u>	Base	0b
14	<u>Completion Timeout Error Severity</u>	Base	RsvdP
15	<u>Completer Abort Error Severity</u>	Base	RsvdP
16	<u>Unexpected Completion Error Severity</u>	Base	RsvdP
17	<u>Receiver Overflow Severity</u>	Base	0b
18	<u>Malformed TLP Severity</u>	Base	0b
19	<u>ECRC Error Severity</u>	Base	0b
20	<u>Unsupported Request Error Severity</u>	Base	RsvdP
21	<u>ACS Violation Severity</u>	Base	RsvdP

9. 4. 2. 7 Correctable Error Status Register Changes (Offset 10h)

Correctable Error Status Register 指示各个可纠正错误的错误检测状态。为 non-Function-specific 定义的错误记录在 PF 中。VF 中仅记录 Function-specific 的错误。

PF 和 VF 的对应功能在 Section 7.8.4.5 节中定义，除非 Table 9-36 给出 VF 的不同实现。

Table 9-36 Correctable Error Status Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
0	<u>Receiver Error Status</u>	Base	0b
6	<u>Bad TLP Status</u>	Base	0b
7	<u>Bad DLLP Status</u>	Base	0b
8	<u>REPLAY_NUM Rollover Status</u>	Base	0b
12	<u>Replay Timer Timeout Status</u>	Base	0b
15	<u>Header Log Overflow Status</u> If the VF implements Header Log sharing (see Section 9.4.2.1), this bit is hardwired to 0b.	Base	Base / 0b

9.4.2.8 Correctable Error Mask Register Changes (Offset 14h)

PF 和 VF 的对应功能在 Section 7.8.4.6 节中定义，除非 Table 9-37 给出 VF 的不同实现。对于标记为 RsvdP 的 VF 字段，PF 设置适用于 VF。

Table 9-37 Correctable Error Mask Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
0	<u>Receiver Error Mask</u>	Base	<u>RsvdP</u>
6	<u>Bad TLP Mask</u>	Base	<u>RsvdP</u>
7	<u>Bad DLLP Mask</u>	Base	<u>RsvdP</u>
8	<u>REPLAY_NUM Rollover Mask</u>	Base	<u>RsvdP</u>
12	<u>Replay Timer Timeout Mask</u>	Base	<u>RsvdP</u>
13	<u>Advisory Non-Fatal Error Mask</u>	Base	<u>RsvdP</u>
15	<u>Header Log Overflow Mask</u> - If the VF implements Header Log sharing (see Section 9.4.2.1), this bit is RsvdP.	Base	Base / <u>RsvdP</u>

9.4.2.9 Advanced Error Capabilities and Control Register Changes (Offset 18h)

PF 和 VF 的对应功能在 Section 7.8.4.7 节中定义，除非 Table 9-38 给出 VF 的不同实现。对于标记为 RsvdP 的 VF 字段，PF 设置适用于 VF。

Table 9-38 Advanced Error Capabilities and Control Register Changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
6	<u>ECRC Generation Enable</u>	Base	<u>RsvdP</u>
8	<u>ECRC Check Enable</u>	Base	<u>RsvdP</u>
9	<u>Multiple Header Recording Capable</u> - If the VF implements Header Log sharing (see Section 9.4.2.1), this bit is hardwired to 0b.	Base	Base / 0b
10	<u>Multiple Header Recording Enable</u> - If the VF implements Header Log sharing (see Section 9.4.2.1), this bit is RsvdP.	Base	Base / <u>RsvdP</u>
11	<u>TLP Prefix Log Present</u> - If the VF implements Header Log Sharing (see Section 9.4.2.1), this bit is 0b when the Header Log contains all 1s due to an overflow condition.	Base	Base (see description)

9.4.2.10 Header Log Register Changes (Offset 1Ch)

Header Log Register 捕获与检测到的错误相对应的 TLP Header。另见 Section 9.4.2.1。

在 VF 中实现 AER 的设备可以在与单个 PF 关联的 VF 之间共享 Header Log Register。共享 Header Log 必须至少存储一个 Header。

PF 的 Header Log Register 独立于其关联的 VF，并且必须使用专用存储空间来实现。

当在 VF 中检测到错误时，应按照 Section 6.2 的规定记录错误。如果实现了一组共享的 Header Log Register，则 VF 可能没有空间记录 Header。在这种情况下，VF 必须按照 Section 6.2 节的要求更新其 Uncorrectable Error Status Register 和 Advanced Error Capabilities and Control Register。但是，当读取该 VF 的 Header Log Register 时，它应返回全 1 来指示溢出情况。

VF 的 header log entry 应被锁定并在 VF 的 First Error Pointer 有效时保持有效。如 Section 6.2 节所述，当 Uncorrectable Error Status 寄存器的相应位置 1 时，First Error Pointer 寄存器有效。当 header log entry 被锁定时，其他错误不得覆盖此或任何其他 VF 的锁定条目。当 header log entry 被解锁时，对于 Header Log Register 的任何 VF，它应可以记录新的错误。

PF 和 VF 的对应功能在 Section 7.8.4.8 中定义，除非 Table 9-39 给出 VF 的不同实现。

Table 9-39 Header Log Register changes

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
127:0	Header of TLP associated with error (additional requirements are described above)	Base	Base

9.4.2.11 Root Error Command Register Changes (Offset 2Ch)

该寄存器不适用于设备。

9.4.2.12 Root Error Status Register Changes (Offset 30h)

该寄存器不适用于设备。

9.4.2.13 Error Source Identification Register Changes (Offset 34h)

该寄存器不适用于设备。

9.4.2.14 TLP Prefix Log Register Changes (Offset 38h)

对于 PF 和 VF，如果支持 End-End TLP Prefix，则实现该寄存器。

对于 VF，如果实现了一组共享的 Header Log Register（Section 9.4.2.1），则由于溢出条件导致 Header Log 包含全 1 时，该寄存器的内容未定义。

9.5 SR-IOV Interrupts

实现了 SR-IOV 的设备使用 Section 6.1 中定义的中断信号机制。

9. 5. 1 Interrupt Mechanisms

发出中断信令的方法有三种：

- INTx
- MSI
- MSI-X

PF 可以实现 INTx。VF 一定不能实现 INTx。如果请求中断资源，则 PF 和 VF 必须实现 MSI 或 MSI-X 或两者都实现。每个 PF 和 VF 必须实现自己的唯一的的中断功能。

9. 5. 1. 1 MSI Interrupts

除 Table 9-40 另有说明外，Section 7.7 节定义了 MSI Capability 以及 PF 和 VF 的相关功能。

Table 9-40 MSI Capability: Message Control

Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
8	Per-Vector Masking Capable	1b	1b

9. 5. 1. 2 MSI-X Interrupts

MSI-X Capability 在 Section 7.7 节中定义，并在 Figure 9-24 中进行了描述。

31	16	15	8	7	3	2	0	
Message Control				Next Pointer		Capability ID		CP + 00h
Table Offset							Table BIR	CP + 04h
PBA Offset							PBA BIR	CP + 08h

A-0383

Figure 9-24 MSI-X Capability

PF 和 VF 的对应功能与 Section 7.7.2 中定义的 Function 相同。

请注意，对于 VF，Table Offset 和 PBA Offset 值是相对于 VF 的内存地址空间的。

9. 5. 1. 3 Address Range Isolation

如果为 MSI-X 表或 MSI-X PBA 映射地址空间的 BAR 也映射了与 MSI-X 结构不相关的其他可用地址空间，则在其他地址空间中使用的位（例如，用于 CSR）不得共享 MSI-X 结构所在的任何自然对齐的 System Page Size 地址范围。MSI-X 表和 MSI-X PBA 可以共存于自然对齐的 System Page Size 地址范围内，尽管它们不能相互重叠。

9. 6 SR-IOV Power Management

本节定义了 PCI Express SR-IOV 电源管理功能和协议。

如 Chapter 5 所述, PF 必须具备电源管理功能。

对于 VF, 电源管理功能是可选的。

9.6.1 VF Device Power Management States

如果 VF 没有实现电源管理功能, 则 VF 的表现的电源状态与其关联的 PF 一样。

如果 VF 实现了电源管理功能, 则该功能参阅 Section 7.5 中的描述, 除非在 Section 9.6.4 中另有说明。

如果 VF 实现了电源管理功能, 并且 PF 处于低于 VF 的电源状态, 则设备行为不确定。软件应通过降低所有 VF 的功率, 然后再降低其相关 PF 的功率状态来避免这种情况。

当 VF 完成其内部初始化并且 VF 的 Bus Master Enable 字段置 1 (请参见第 9.3.4.1.3 节) 或 SR-IOV Control (see Section 9.3.3.3) Extended Capability 中的 VF MSE 置 1 时, 其处于 D0_{active} 状态。发生以下任何一种情况时, VF 的内部初始化必须完成:

- VF 已成功响应配置请求 (不返回 CRS) 。
- 向 VF 发出 FLR 后, 以下情况之一为真:
 - 自发送 FLR 以来至少已过 1.0 s。
 - VF 支持 Function Readiness Status, 并且在发出 FLR 之后, 已收到来自 VF 的 FRS Message, Reason Code 为 FLR Completed。
 - 自发送 FLR 以来, 至少已经过去了 FLR time。FLR Time 是 (1) 与 VF 相关联的 Readiness Time Reporting Capability 中的 FLR Time 值, 或者 (2) 由系统软件/固件确定的值 (如 ACPI 定义)。
- 在 PF 中设置 VF Enable 为 1 后, 至少满足以下条件之一:
 - 自 VF Enable 设置为 1 以来已过了至少 1.0 s。
 - PF 支持 Function Readiness Status, 并且在设置了 VF Enable 后, 已收到来自 PF 的 Reason Code 为 VF Enable 的 FRS Message。
- 将 VF 从 D3_{Hot} 转换为 D0 后, 至少满足以下条件之一:
 - 自发出进入 D0 的请求以来已过去至少 10 ms。
 - VF 支持 Function Readiness Status, 并且在发出进入 D0 的请求之后, 收到了来自 VF 的 FRS Message, Reason Code 为 D3_{Hot} to D0 Transition Completed。
 - 自发出进入 D0 的请求以来, 至少已经过了 D3_{Hot} to D0 Time。D3_{Hot}to D0 Time 是 (1) 与 VF 相关联的 Readiness Time Reporting Capability 中的 D3_{Hot}to D0 Time, 或者 (2) 由系统软件/固件确定的值 (如 ACPI 定义)。

9.6.2 PF Device Power Management States

PF 的电源管理状态 (D-state) 对其关联的 VF 具有全局影响。如果 VF 没有实现电源管理功能, 则 F 的表现的电源状态与其关联的 PF 一样。

如果 VF 实现了电源管理功能, 则如果 PF 的电源状态低于 VF, 则设备行为是不确定的。软件应通过降低所有 VF 的功率, 然后再降低其相关 PF 的功率状态来避免这种情况。

当 PF 进入 D3_{Hot} 状态时:

- 如果 No_Soft_Reset 字段为 0，则 PF 在从 D3_{Hot} 到 D0 的切换上执行内部复位，并且其所有配置状态均返回默认值。
Note：复位 PF 会复位 VF Enable，这意味着 VF 不再存在，并且在完成从 D3_{Hot} 到 D0 的转换之后，任何 VF 特定的上下文都会丢失。
 - 如果 No_Soft_Reset 字段置 1，则不会发生内部复位。SR-IOV Extended Capability 保持状态，并且关联的 VF 保持使能。
- 当 PF 置于 D3_{Cold} 状态时，VF 不再存在，任何 VF 特定的上下文都将丢失，并且 PME 事件只能由 PF 发起。

No_Soft_Reset Strongly Recommended

强烈建议在 Multi-Function Device 的所有 Function 中将 No_Soft_Reset 位置 1。该建议适用于 PF。

9. 6. 3 Link Power Management State

VF 电源状态不会影响链路电源状态。
不管 VF 的 D-state 如何，Link Power State 都仅由 PF 中的设置控制。

9. 6. 4 VF Power Management Capability

下表列出了 PF 和 VF 电源管理功能的要求。
PF 和 VF 的对应功能在 Section 7.5 中定义，Table 9-41 和 Table 9-42 中注明的是 VF 实现的不同点。

Table 9-41 SR-IOV Power Management Control/Status (PMCSR)			
Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
14:13	Data_Scale	Base	00b
12:9	Data_Select	Base	0000b
3	No_Soft_Reset -If a VF implements the Power Management Capability, the VF's value of this field must be identical to the associated PF's value.	Base	Base

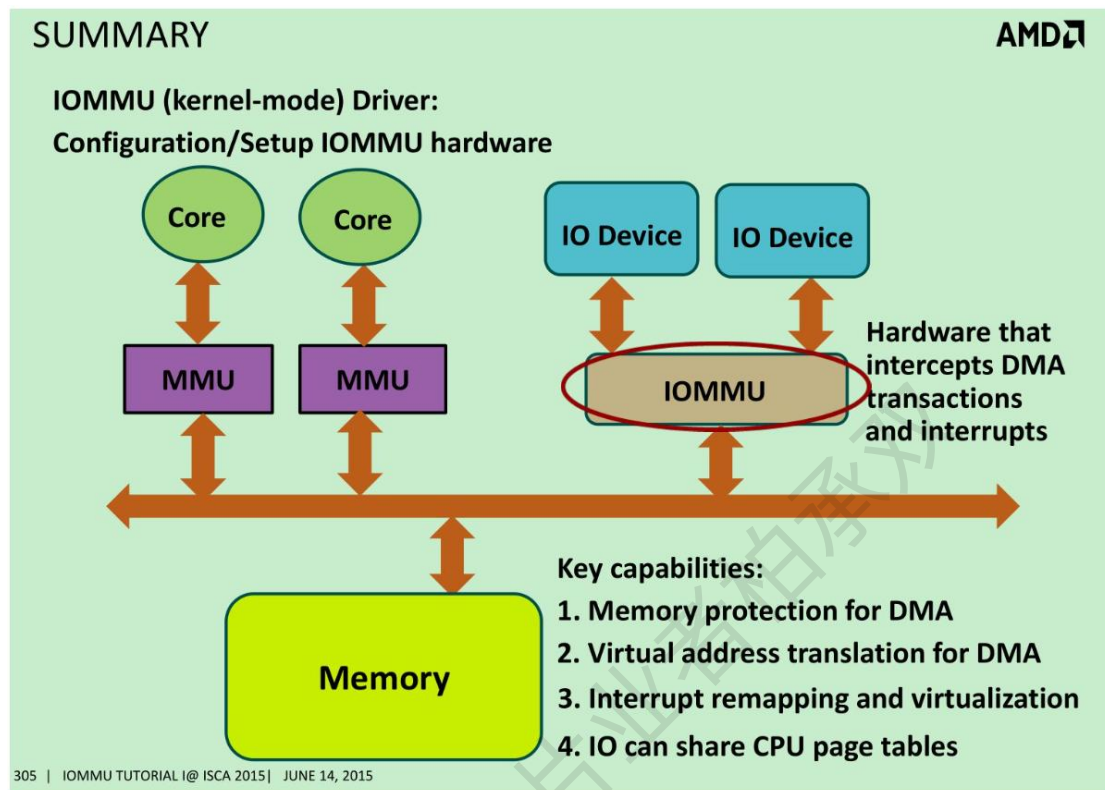
Table 9-42 SR-IOV Power Management Data Register			
Bit Location	PF and VF Register Differences From Base	PF Attributes	VF Attributes
7:0	Data	Base	00000000b

9. 6. 5 VF EmergencyPower Reduction State

如果 VF 中的 Emergency Power Reduction Supported 字段不为零,则该 VF 与关联的 PF 同时进入和退出 Emergency Power Reduction State。软件可以使用 PF 中的 Emergency Power Reduction Detected 字段来模拟 VF 中的相应字段。

10 ATS Specification

补充 IOMMU 简单介绍



1. MMU 把 CPU 访问的虚拟地址转换为内存物理地址并进行地址保护；IOMMU 把设备访问的虚拟地址转换为内存物理地址。
2. MMU + TLB 使得 CPU 可以支持多任务；IOMMU + TLB 使得 IO 可以支持多设备（I/O Function），此处的 I/O Function 可以类比为 CPU 的多任务。
3. TLB 是页表的缓存，用于提高页表访问的效率。IOMMU 也必须实现类似 TLB 的缓存用于提高 IO 访问页表的效率。PCIe ATS 功能中 ATC 就是类似于 TLB 机制的缓存。
4. IOMMU 的一个重要用途是虚拟化技术（virtualization）：虚拟机上运行的操作系统（guest OS）通常不知道它所访问的 host-physical 内存地址。如果要进行 DMA 操作，就有可能破坏内存，因为实际的硬件（hardware）不知道 guest-physical 和 host-physical 内存地址之间的映射关系。IOMMU 根据 guest-physical 和 host-physical 内存地址之间的转换表（translation table），re-mapping 硬件访问的地址，就可以解决这个问题。引入虚拟化后，带来的问题是：设备看到的是 GPA(guest physical address)，但是访问的是 HPA(host physical address)。
5. 结合主机 MMU 和 IOMMU 可以使得 CPU 与设备访问的是同一块虚拟地址。
6. IOMMU 使得不在 BOOT 的时候为设备分配存储空间成为可能。

10.1 ATS Architectural Overview

大多数现代系统的体系结构都提供了对设备 I/O Function 的 DMA 的地址转换的支持。在许多实现中，通常的做法是假定 CPU 和 I/O Function 看到的物理地址空间是等效的。在支持 DMA 地址转换的系统中则不是这样：编程到 I/O Function 中的地址是由 RC

处理的“句柄”，该处理的结果通常是转换为中央聚合系统中的物理内存地址。通常，该处理包括访问权限检查，以确保允许 DMA 访问此内存地址。

DMA 地址转换的目的各不相同，包括：

- 限制“broken”或未正确编程的 DMA I/O Function 的破坏性。
- 提供分散/聚集（scatter/gather）功能。
- 能够将消息信号中断（例如 MSI 或 MSI-X）重映射到不同的地址范围，而无需与基础 I/O Function 进行协调。

在虚拟化透传设备使用中，或者主机侧用户态驱动框架中还有一个问题就是 msi 中断的安全保护。msi 的特点就是只要发起特定的 pci write 消息，机器(在 x86 上是 LAPIC)就能够将此 pci write 消息翻译为一次中断信号。在 msi 中断信息里，含有中断源信息，中断 vector 信息以及中断发往哪里，中断模式等等信息。

如果设备透传到虚拟机，而虚拟机里有恶意驱动，那么恶意驱动完全可以操作一个透传设备发起 dma write 访问，带有 msi 中断信息，导致 hypervisor 被攻击。

在用户态驱动层面也是相同的，用户态驱动可以触发 dma 带有 msi 中断导致主机系统被攻击。

为了防止这种情况发生，Intel 的 iommu 技术（vt-d）里实现了中断重映射技术（intremap）。设备发起的 msi 中断先进入 iommu 的 intremap 模块进行映射得到真正的 msi 中断信息。主机 os 会在这个中断重映射表里限定，某个设备只能发起哪种中断向量，同时会验证 msi 中的信息位（如此提升了恶意软件触发真实 msi 的难度）。

这也是在公司李同学遇到了某款有硬件缺陷的主机支持 iommu 的 dmar 但是不支持 intremap，开启 vfio 的时候 linux kernel 不会开启设备的 vfio，提示只有开启 vfio 模块的 allow_unsafe_interrupts 配置后才能使用 vfio 的原因。

- 地址空间转换（将 32 位 I/O Function 转换为更大的系统地址空间）。只需要在 iommu 地址映射表上配置 32bit pci 设备 dma 地址-->64 位机器物理地址即可。
- 虚拟化支持。

不管动机如何，主机系统中 DMA 地址转换的存在对 DMA 访问都有一定的性能影响。

根据实现，由于存在解析实际物理地址所需的时间，DMA 访问时间可能会大大延长。如果实现要求访问驻留在主存中的转换表，则访问时间可能比未转换的访问时间长得多。另外，如果每个事务需要多个存储器访问（例如，用于表遍历），则与 DMA 相关联的内存事务开销会很高。

为了减轻这些影响，设计通常在执行地址转换的实体中实现地址转换缓存。在 CPU 中，地址转换缓存最常被称为转换后备缓冲器（TLB）。对于 I/O TA 而言，术语 ATC 用于将其与 CPU 使用的转换缓存区分开。

尽管 TLB 和 ATC 之间有一些相似之处，但仍存在重要区别。TLB 可满足名义上一次运行一个线程的 CPU 的需求。但是，ATC 通常用于处理来自多个 I/O Function 的请求，每个 Function 都可以视为一个单独的线程。这种差异使得根据成本模型和跨广泛系统配置的预期技术重用两个因素无法确定 ATC 的尺寸。

本规范中描述的机制使得 I/O 设备可以参与地址转换，并为其自身的内存访问提供 ATC。在设备内实现 ATC 的好处包括：

- 通过分配地址转换缓存减轻 TA 资源压力的能力（降低了 TA 中“thrashing”的概率）。
- 使 ATC 设备对系统的 ATC 大小的性能依赖性降低。
- 通过将预转换的请求发送到中央聚合系统来确保最佳访问延迟。

关于 thrashing

在更换页面时，如果更换的页面是一个很快会被再次访问的页面，则再次缺页中断后又很快会发生新的缺页中断。这会造成整

个系统的效率急剧下降-----这种现象称为 **thrashing**（颠簸，或者抖动）：

内存抖动的解决策略是：

- 1. 如果是因为页面替换策略失误，可以修改替换算法来解决这个问题；
- 2. 如果是因为运行的程序太多，造成程序无法同时将所有频繁访问的页面调入内存，则要降低多道程序的数量。
- 3. 否则，还剩下两个办法：1 终止该进程；2 增加物理内存容量。

该规范将会提供 PCIe 设备与 TA 结合使用的互操作性，但是 TA 及其 Address Translation and Protection Table (ATPT) 被视为特定于实现的，不在本规范范围之内。尽管可能在其他 PCIe 组件中实现 ATS，但此规范仅限于 PCIe 设备和 PCIe RC Integrated Endpoint(RCiEP)。

Figure 10-1 给出了一个具有 TA 和 ATPT 的示例平台，以及一组 PCIe 设备和集成了 ATC 的 RC Integrated Endpoint。TA 和 ATPT 是特定于实现的，在给定的系统设计中可以是不同或集成的组件。

其中 TA (Translation Agent)，ATC (Address Translation Cache)，ATS (Address Translation Services)，ATPT (Address Translation and Protection Table)。理解 ATS 最好先建立 IOMMU 的概念。TA 实现的应该就是 IOMMU 的功能，ATPT 实现的应该就是 TLB 的功能。

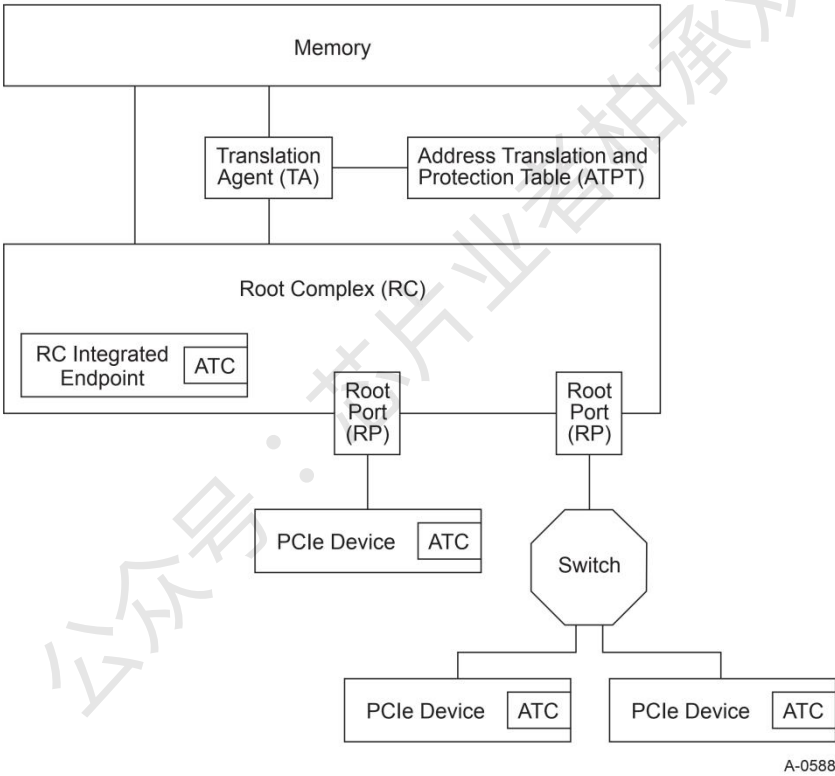


Figure 10-1 Example Illustrating a Platform with TA, ATPT, and ATC Elements

10. 1. 1 Address Translation Services (ATS) Overview

ATS 章节提供了一组新的 TLP 和其语义。ATS 使用 request-completion 协议为设备¹⁶³和 RC 之间提供地址转换服务。另外，在 Memory Read 和 Memory Write TLP 中定义了新的 AT 字段。AT 字段使 RC 能够通过 ATS 协议确定给定的请求其地址是否已经发生转换。

注 163：本章中对设备的所有引用同样适用于 PCIe 设备或 RC Integrated Endpoint。ATS 并未在需求，语义，配置，错误处理等方面对这两种类型进行区分。从软件的角度来看，具有 ATS 功能的 RC Integrated Endpoint 必须与具有 ATS 功能的非集成设备具有相同的行为。

Figure 10-2 说明了 ATS Translation Request 操作的基本流程。

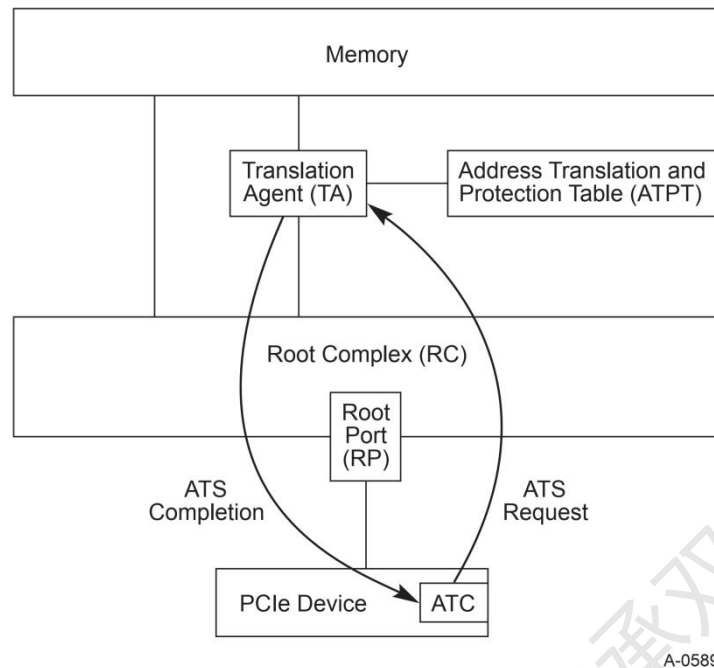


Figure 10-2 Example ATS Translation Request/Completion Exchange

在此示例中，single-Function PCIe Device 收到一个 Function 特定的工作请求。该 Function 通过特定于实现的方法来判断在其 ATC 中缓存转换地址会提高效率。Function 或软件在进行此类判定时可以考虑许多因素，例如：

- 长时间内频繁访问的内存地址范围，或者其关联的缓冲区内容更新速率明显很快。
- 内存地址范围，例如工作和完成队列结构、用于低延迟通信的数据缓冲区、图形帧缓冲区、用于缓存 Function 特定的内容的主机内存等。

考虑到设计和访问模式的可变性，没有可以应用的单一标准。

该 Function 生成一个 ATS Translation Request，该请求通过向上游发送给 RC，然后由 RC 转发给 TA。ATS Translation Request 使用与本规范中定义的相同的路由和排序规则。此外，在任何给定时间都可以处理多个 outstanding ATS Translation Request；也就是说，您可以在一个或多个 TC 上流水线处理多个请求。每个 TC 代表一个唯一的排序域，并定义相关的 ATS Translation Completion 必须使用的域。

收到 ATS Translation Request 后，TA 将执行以下基本步骤：

1. 验证 Function 已配置为可以发出 ATS Translation Request 的状态。
2. 确定 Function 是否可以访问 ATS Translation Request 指示的内存并具有相关的访问权限。
3. 确定是否可以为 Function 提供地址转换。如果是，则 TA 向 Function 发出一个转换。
 - a. ATS 需要支持各种页面大小，以适应各种 ATPT 和处理器实现。
 - 1) 页面大小必须为 2 的幂并自然对齐。
 - 2) 支持的最小页面大小为 4096 字节。支持 ATS 的组件才能支持此最小页面大小。
 - b. 必须告知 Function 它需要支持的最小转换或无效大小，以使 Function 有机会优化其资源利用率。最小的转换大小必须为 4096 字节。

4. TA 将请求的成功或失败传达给 RC, RC 生成 ATS Translation Completion, 并通过通过 RP 传输到 Function。
 - a. 要求 RC 为每个 ATS Translation Request 生成至少一个 ATS Translation Completion; 这与请求的成功或失败无关, 最少存在 1:1 对应关系。
 - 1) 成功的转换会导致每个请求返回一个或两个 ATS Translation Completion。Translation Completion 指示所涵盖的转换范围。
 - 2) RC 可以流水线化多个 ATS Translation Completion; 即, RC 可以返回多个 ATS Translation Completion, 并且这些 ATS Translation Completion 可以相对于 ATS Translation Request 采用任何顺序。
 - 3) 要求 RC 使用与相应的 ATS Translation Request 相同的 TC 来发送 ATS Translation Completion。
 - b. 请求的地址可能无效。这时 RC 要发一个 Translation Completion, 指示无法访问所请求的地址。

当 Function 收到 ATS Translation Completion 信息, Function 会选择更新其 ATC 以反映该转换, 或通知此转换不存在。Function 继续处理其工作请求, 并根据此 Completion 的结果使用转换后的地址或未转换的地址生成后续请求。

- a. 与 Read Completion 类似, 需要一个 Function 为每个 Completion 分配资源空间, 而又不会对 PCIe 链路造成反压。
- b. Function 需要丢弃可能“过时”的 Translation Completion。发生过时转换的原因可能多种多样。

可以推测, ATS Translation Request 和 Translation Completion 处理在概念上是相似的, 并且在许多方面与 PCIe 读请求和读完成处理相同。这是为了降低设计复杂性并简化 ATS 与现有和新的基于 PCIe 的解决方案的集成。请记住, ATS 需要满足以下条件:

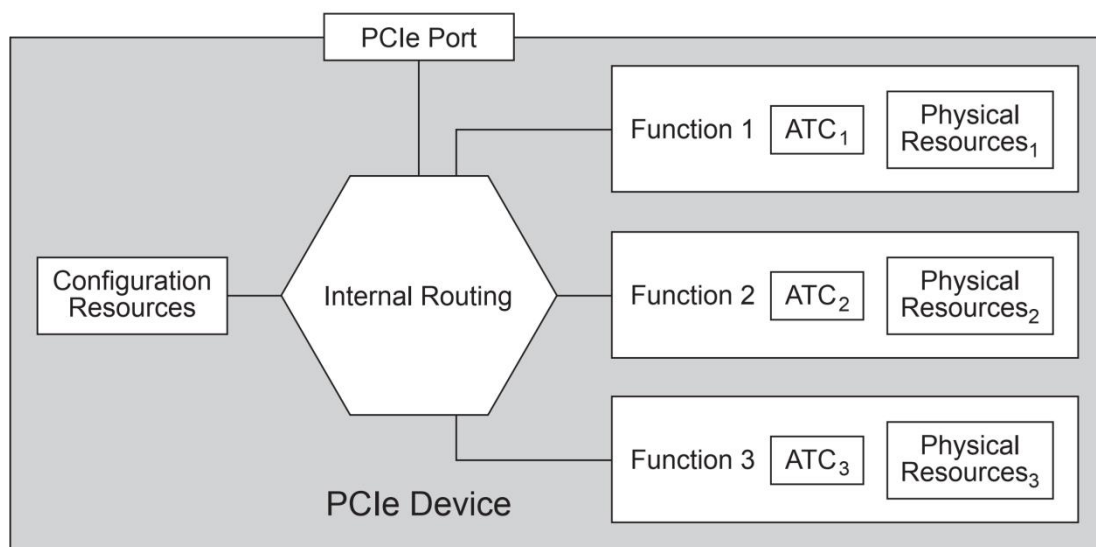
- 具有 ATS 功能的组件必须能与[PCIe-1.1]规范兼容的组件互操作。
- 通过新的 Capability 和相关的配置结构启用 ATS。要启用 ATS, 软件必须能检测到此 Capability 并使 Function 能够发送 ATS TLP。如果未启用此 Capability, 则要求该 Function 不发出 ATS Translation Request, 并且其发出的所有 TLP 请求为 AT 字段为 0 的 DMA 读和写请求。
- ATS TLP 使用基于地址的路由或基于 Requester ID (RID)的路由进行路由。
- ATS TLP 必须使用与本规范中指定的排序规则相同的排序规则。
- 要求 ATS TLP 不经修改地通过[PCIe-1.1]兼容的 Switch。
- 允许 Function 混用已转换和未转换的请求。
- 要求 ATS 事务不依赖内存请求的地址字段来传递 PCI-SIG 定义的当前用途以外的其他信息。

Address Range Overlap

已转换和未转换的地址范围可能会完全重叠。这不是 ATS 的要求, 而是对 TA 的实现限制, 以便可以正确路由内存请求。

与之前的示例相反, Figure 10-3 给出了一个 Multi-Function Device 的示例。在此示例设备中, 有三个 Function。Figure 10-3 中需要注意的关键点是:

- 每个 ATC 与一个 Function 关联。每个支持 ATS 的 Function 必须能够发出和接收每种 ATS Translation Request 和 Translation Completion 类型中的至少一种。
- 每个 ATC 在每个 Function 的基础上进行配置和访问。不需要 Multi-Function Device 在每个 Function 上都实现 ATS。
- 如果 ATC 实现在一组 Function 之间共享资源, 则逻辑行为必须与完全独立的 ATC 实现一致。



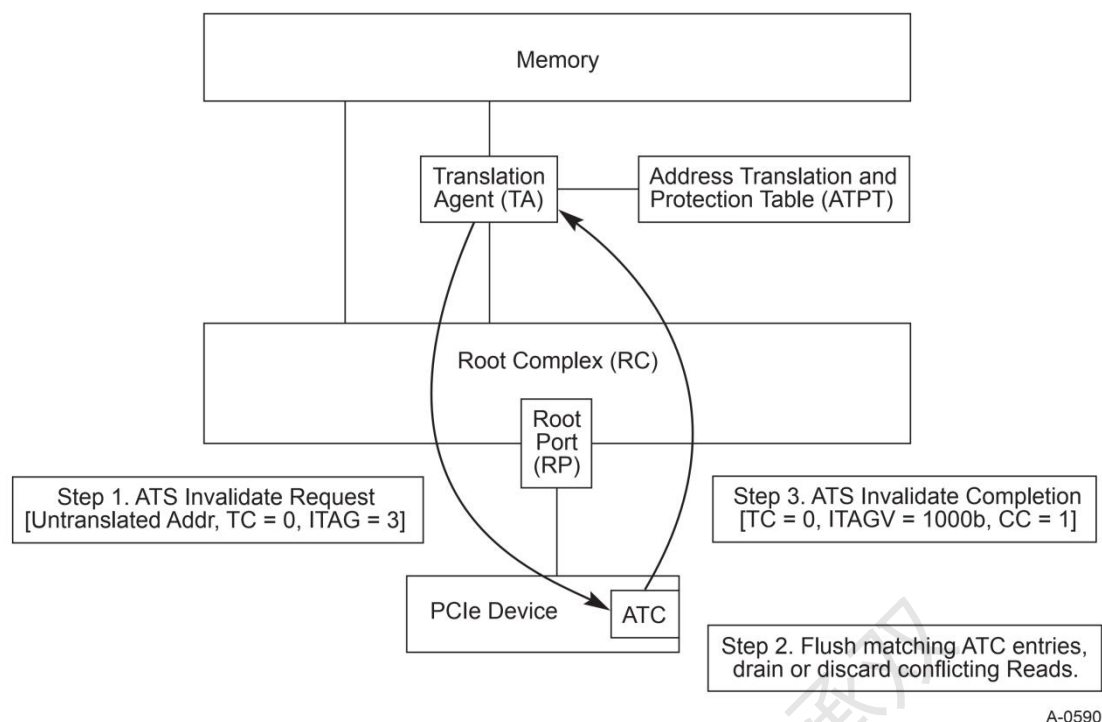
A-0592

Figure 10-3 Example Multi-Function Device with ATC per Function

与设备内的 Function 数量无关，以下各项是必需的：

- 除非通过 ATS Translation Request 和 Translation Completion 协议获得了 TLP 内的地址，否则该 Function 不得发送 AT 字段为 1 的 TLP。
- 每个 ATC 要求使用 ATS 协议进行唯一填充；即，每个 ATC 的条目必须通过用于响应 Function 针对特定地址发出的 ATS Translation Request 的 ATS Translation Completion 来填充。
- 每个 ATC 只能通过 ATS 协议来修改。即：
 - ✧ 主机系统软件只能通过本规范中定义的协议来修改 ATC，除非使 ATC 中的一个或多个转换无效。设备或 Function 复位是由软件执行的更改 ATC 内容的操作的示例，但是复位只允许使条目无效，而不能修改其内容。
 - ✧ 主机系统软件不能使用设备上执行的软件来修改 ATC。

当 TA 确定某个 Function 不应再在其 ATC 中维护转换条目时，TA 会启动 ATS Invalidation 协议。ATS 失效协议由一个 Invalidation Request 和一个或多个 Invalidate Completion 组成。



A-0590

Figure 10-4 Invalidation Protocol with a Single Invalidation Request and Completion

如 Figure 10-4 所示，ATS Invalidation 协议实质上包含三个步骤：

1. 系统软件更新 TA 使用的表中的条目。更改表后，TA 确定应在 ATC 中使转换无效，并发送一个 Invalidation Request TLP，该请求将从 RP 发送到示例中的 single-Function Device。Invalidation Request 传送未转换的地址范围、TC 和 RP 唯一 Tag，该 Tag 用于将 Invalidate Completion 与 Invalidation Request 相关联。
2. Function 接收 Invalidation Request 并使所有匹配的 ATC 条目无效。收到 Invalidation Request 后，不需要 Function 立即刷新所有挂起的请求。如果事务在队列中等待发送，则即使这些事务使用无效的地址，Function 也不必删除队列中的请求。
 - a. Function 不能表示无效过程已完成，直到所有 outstanding 的引用关联转换地址的 Read Request 或 Translation Request 都被淘汰或无效。
 - b. 要求有一个 Function 来确保到 RC 的 Invalidate Completion 指示将在先前发布的任何使用“陈旧”地址的写入之后到达 RC。
3. 当某个 Function 确定所有已转换地址的使用均已完成时，它将发出一个或多个 ATS Invalidate Completion。
 - a. 为可能引用了无效范围的每个 TC 发出 Invalidate Completion。这些 Completion 充当刷新机制，以确保清除层次结构中所有可能包含对已转换地址的引用的正在进行中的事务。
 - 1) 所需的 Completion 次数在每个 Invalidate Completion 中传达。TA 或 RC 实现可以维护一个计数器，以确保在考虑不再使用转换之前已收到所有 Invalidate Completion。
 - 2) 如果发送了多个 Invalidate Completion，则每个 TC 中发送的 Invalidate Completion 必须与在 Section 10.3.2 中详述的字段中相同。
 - b. Invalidate Completion 包含来自 Invalidation Request 的 ITAG，以使 RC 能够关联 Invalidation Request 和 Completion。

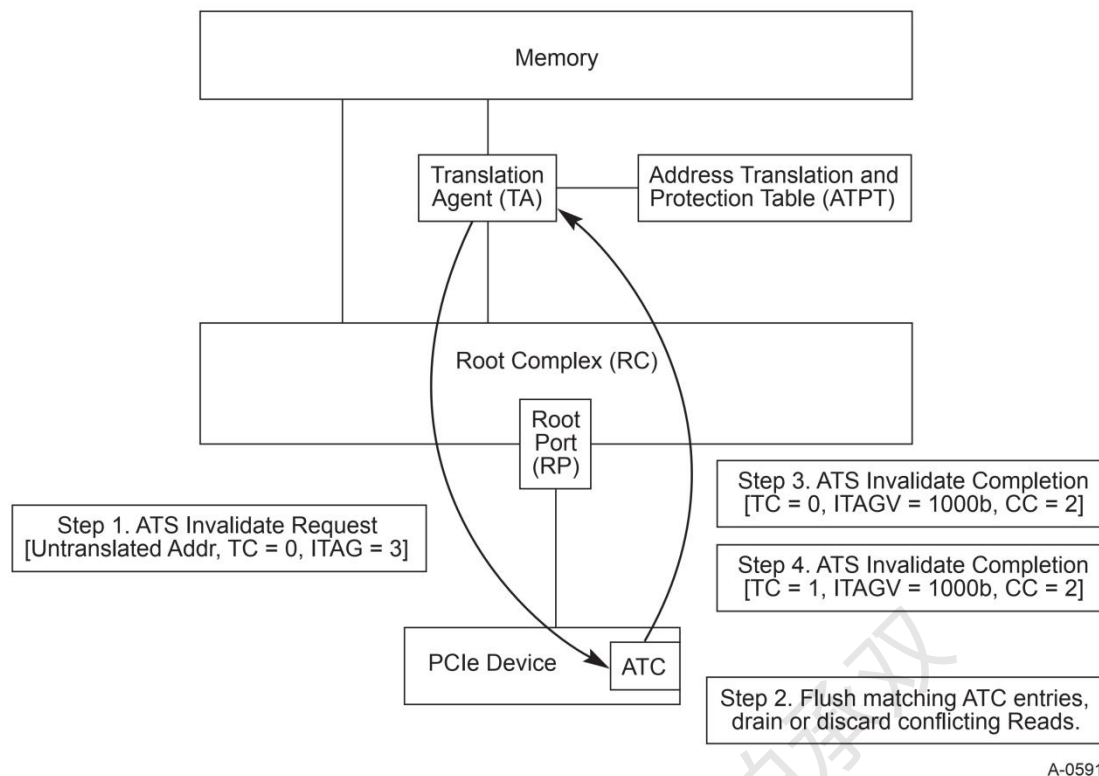


Figure 10-5 Single Invalidate Request with Multiple Invalidate Completions

10.1.2 Page Request Interface Extension

ATS 改善了基于 DMA 的数据搬运的行为。关联的页面请求接口 (PRI, Page Request Interface) 通过允许启动 DMA 操作而不需要固定所有要移入或移出系统内存的数据而提供了其他优点¹⁶⁴。与固定内存相关的开销可能不大, 但是从可分页池中删除大量内存对系统性能的负面影响可能非常大。

注 164: 这里的**固定** (原文 Pinned, Pinning) 指锁定在适当的位置, 以使其不能被系统的动态分页机制替换掉。也就是指传统的在 BOOT 时就为设备分配好的内存区域, 基本上这块内存一旦分配给设备就不会再用于分页机制。

PRI 在功能上独立于 ATS 的其他方面。也就是说, 支持 ATS 的设备不需要支持 PRI, 但是 PRI 却依赖于 ATS。

可以构造智能 I/O 设备以充分利用动态内存接口。从设备的角度来看, **固定**将始终具有最佳的性能-即能保证要访问的所有内存都存在。但是, 确保设备可能访问到的所有存储器的驻留可能会出现问題, 并在主机上强制达到次优级别的设备感知能力。允许设备更独立地运行 (当它需要不存在的内存资源时出现页面错误) 可以提供设备与主机之间更高级的耦合¹⁶⁵。

注 165: 另一种选择是设备与其驱动程序之间的专用接口, 用于传递设备状态, 以便驱动程序可以确保固定内存资源的可用性。

利用页面请求接口的机制是特定于设备的。作为这种接口可以改善整体系统性能的模型的示例, 让我们研究一下高速 LAN 设备。这样的设备知道其突发速率, 并且仅需要在某个量子范围内可以接收的物理缓冲区空间就可以用于入站数据。可以使未固定虚拟内存页面的向量可用于设备, 然后设备根据需要请求以维持其突发窗口。这样可以最大程度地减少所需的设备内存占用, 并简化与主机的接口, 而这两者都不会对性能产生负面影响。

进行分页的能力引发了页表状态标志管理的问题。典型的 TA 将标志 (例如, 脏和访问指示) 与每个未转换的地址相关联。如果没有关于如何管理映射到 Function 的页面的任何其他提示, 则此类 TA 将需要保守地假定, 当它们授予 Function 读取或写入页面的权限时, 该 Function 将使用该权限。此类可写页面在将其转换的地址提供给 Function 之前需要标记为脏。

对于不支持分页的 Function 来说, 这种保守的写许可授予脏许可行为通常不是什么大问题, 分页是固定在页面上的, 很少需要支付将干净页面保存到内存的成本。但是, 如果所有可写页面都被视为脏页, 则支持页面请求接口的 Function 可能会遭受重大

损失，因为此类 Function 在运行时不会固定其可访问的内存占用空间，并且可能会发出推测性的页面请求以提高性能。在这样的系统中保存干净页面（而不只是丢弃它们）的成本会降低原本有吸引力的分页技术的价值。在无法写入后备存储的情况下（例如 CD-ROM），这可能会导致严重的性能问题和功能风险。

Translation Requests 中的 No Write (NW) 标志表示某个 Function 愿意将其用法限制为仅读取页面，而与原本应授予的访问权限无关。

如果设备通过发出设置了 NW 标志的转换请求选择仅请求读取访问，然后又确定需要将其写入页面，则该设备必须发出新的转换请求。

在收到带有 NW 标记为 Clear 的转换请求后，允许 TA 将相关页面标记为脏。强烈建议 Function 除非发出明确的写许可，否则不要发出此类请求。写入许可的一个示例是主机向 Function 发出命令以从存储设备加载数据并将该数据写入内存。

10.1.3 Process Address Space ID (PASID)

某些 TLP 可以选择与 Process Address Space ID (PASID) 相关联。该值是使用 PASID TLP Prefix 传送的。PASID TLP Prefix 在 Section 6.20 中定义。

PASID TLP Prefix 可用于：

- Memory Requests (including Untranslated AtomicOp Requests) with Untranslated Addresses
- Address Translation Requests
- Page Request Messages
- ATS Invalidation Requests
- PRG Response Messages

Section 6.20 中定义了 PASID TLP Prefix 用于 Untranslated Memory Request 的用法。本节介绍了其余 TLP 的 PASID TLP Prefix 的用法。

当请求没有包含 PASID TLP Prefix 时，未转换的地址表示与 Requester ID 关联的地址空间。

当请求包含 PASID TLP Prefix 时，未转换的地址表示与 Requester ID 和 PASID 值都关联的地址空间。

当 Response 包含 PASID TLP Prefix 时，PASID 值反映与相应请求关联的地址空间。

每个 Function 都有一组独立的 PASID 值。PASID 字段的宽度为 20 位，但是有效宽度受到 RC (TA) 和 Function (ATC) 支持的宽度中较小者的限制。PASID 值的未使用的高位必须为 0b。

对于存在虚拟中介 (VI) 的系统中的 Endpoint，地址是未转换的。

具有关联的 PASID 的地址通常用于表示 Guest Virtual Addresses (GVA)，而与 PASID 不关联的未转换地址则表示 Guest Physical Addresses (GPA)。可以设计 TA，以便 VI 管理用于执行从 GPA 到转换地址的转换表，而各个 Guest Operating System 管理用于执行从 GVA 到 GPA 的转换表。当使用关联的 PASID 转换地址时，TA 会执行两次转换并返回最终的转换地址（即，将 GVA 转换为 GPA，然后将 GPA 转换为转换地址）。中间 GPA 值对 ATC 不可见。

当 ATC 使缓存的 GPA 映射无效时，它将使 GPA 映射无效，并使 ATC 中的所有 GVA 映射也无效。GPA 失效完成后，VI 可以安全地从 Guest Operating System 中删除支持 GPA 内存范围的页面。VI 不需要知道哪些 GVA 映射涉及 GPA 映射。

10.2 **ATS Translation Services**

TA 负责转换。ATC 可以缓存这些转换。如果通过 PCIe 将 ATC 与 TA 分开，则来自 ATC 的内存请求将需要能够指示事务中的地址是否已转换。在本节中将描述对内存事务的修改，以及用于在远程 ATC 和中央 TA 之间传递转换的事务。

10.2.1 **Memory Requests with Address Type**

实现了 ATC 的 Function 可以发送已转换或未转换地址的内存读/写请求。如 Figure 10-6 和 Figure 10-7 所示，Address Type (AT) 字段用于指示请求报头中的地址类型。

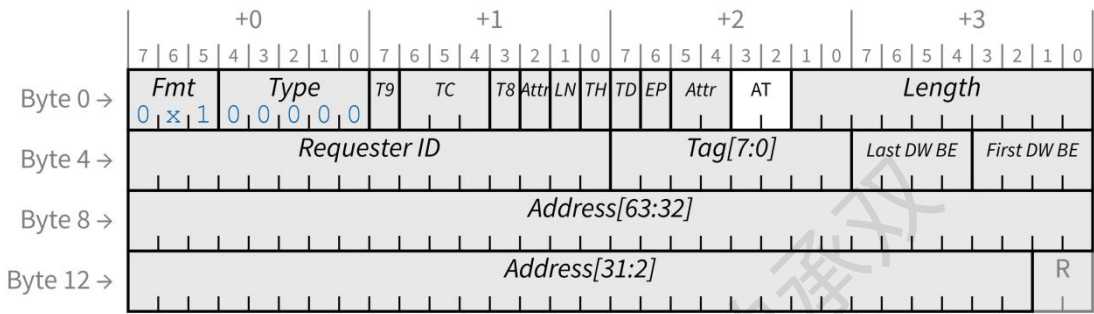


Figure 10-6 Memory Request Header with 64-bit Address

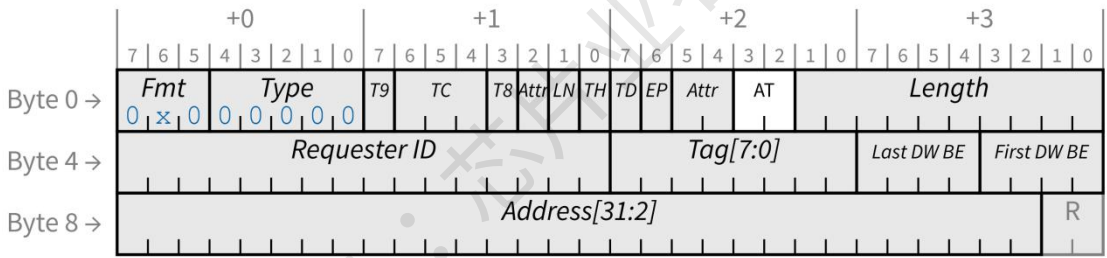


Figure 10-7 Memory Request Header with 32-bit Address

请求中的 AT 字段是本规范早期版本中保留字段的重新定义。未实现 ATC 的 Function 将继续将 AT 字段设置为其定义的保留值 (00b)。实现 ATC 的 Function 将如 Table 10-1 所示设置 AT 字段。

Table 10-1 Address Type (AT) Field Encodings

AT[1:0] Coding	Mnemonic	Meaning
00b	Untranslated	A TA may treat the address as either virtual or physical.
01b	Translation Request	The TA will return the translation of the address contained in the address field of the request as a read completion. This value only has meaning for an explicit Translation Request (see Section 10.2.2). The TA will signal an Unsupported Request (UR) if it receives a TLP with the AT field set to 01b in a Memory Request other than Memory Read.
10b	Translated	The address in the transaction has been translated by an ATC. If the Function associated with the SourceID is allowed to present physical addresses to the system memory, then the TA might not translate this address. If the Function is not allowed to present physical addresses, then the TA may treat this as an UR.
11b	Reserved	The TA will signal an Unsupported Request (UR) if it receives a Memory Request TLP with the AT field set to 11b.

AT 字段仅为“内存请求”定义。对其他 TLP 为保留值。

10. 2. 2 Translation Requests

Translation Request 的格式类似于内存读请求。AT 字段用于区分是 Translation Request 和普通的 Memory Read Request。

Translation Request header 的格式如 Figure 10-8 和 Figure 10-9。

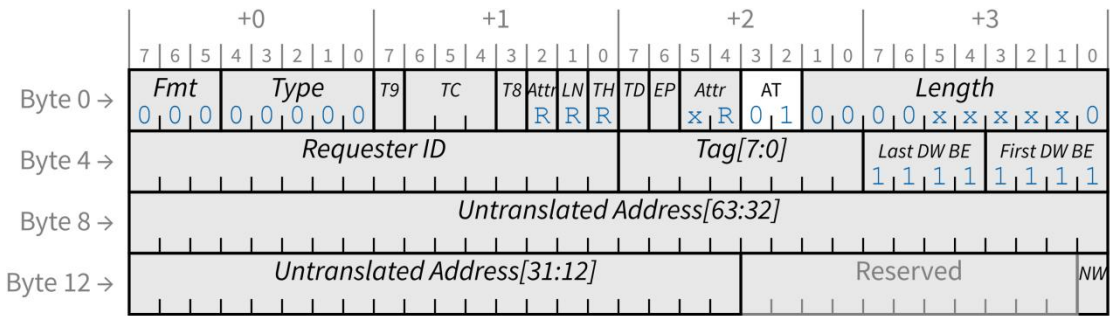


Figure 10-8 64-bit Translation Request Header

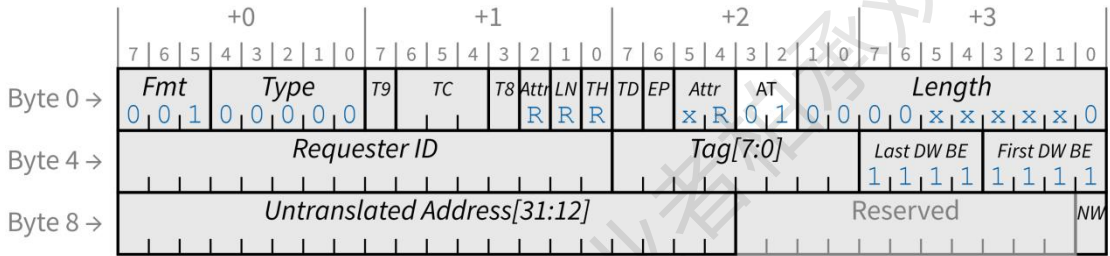


Figure 10-9 32-bit Translation Request Header

Translation Request 与 Memory Read Request 具有相同的完成超时间隔。

10. 2. 2. 1 Attribute Field

对于 Translation Request，Relaxed Ordering (RO) 字段允许使用，这会影响其关联的 Translation Completion 的顺序。Attr 字段的其余部分为保留值。Translation Request 的请求者不得依赖 TA 来保证 Translation Completion 与任何其他 Request 或 Completion 之间的任何特定排序关系。Translation Request 没有排序要求。TA 可以针对任何其他请求重新排序 Translation Request。

Translation Request Ordering

因为在 Translation Request 和其他类型的请求之间不能假定排序，所以 Translation Request 不会产生有效的刷新/排序原语。

10. 2. 2. 2 Length Field

Length 字段的设置表示为响应该请求可以返回多少 Translation。每个 Translation 的长度为 8 个字节，代表一个或多个 STU (Smallest Translation Unit, 最小转换单元)。Length 字段的最大值是 RCB。Translation Request 中的 Length 字段必须始终指示偶数个 DWORD。如果将 Length 设置为大于允许的值，或者如果 Length 字段的最低有效位不为零，则 TA 将请求视为 Malformed Packet。

如果 Length 字段的值大于 2，则 Function 正在请求转换范围大于单个 STU 的内存。如果提供了其他转换，则假定这些转换是从请求的地址开始，按顺序增加，大小相等的 STU 对齐区域。

10.2.2.3 Tag Field

Tag 字段具有与 Memory Read Request 相同的含义。

10.2.2.4 Untranslated Address Field

Translation Request 包括 32 位或 64 位 Untranslated Address 字段。该字段表示要转换的地址。TA 将根据 Translation Request 中的地址来决定请求的有效性。TA 返回的转换数允许少于请求的转换，但不能返回多余的转换。

当请求多个转换时，如果该转换的范围不与 Translation Request 的隐含范围重叠（仅适用于初始值之后的转换），则 TA 将不返回转换。Translation Request 的隐含范围是 $[2^{STU+12} * (Length / 2)]$ 个字节。

Translation Request 中的 Untranslated Address 字段是第一个 STU 范围内的任何地址。地址位 11:0 在 Translation Request 中不存在，并且隐含为零。如果请求者 Page Aligned Request 字段为 1（请参见 Section 7.8.8.2），则必须确保位 11:2 为零。如果请求者 Page Aligned Request 字段为 0，则允许为位 11:2 提供任何值。TA 必须忽略 Bit 11:2 以及确定转换所需的任何低位。

例如，如果 Page Aligned Request 设置为 1 的 Function 使用 64 位寻址，并且其 STU 设置为 1（即 8192 字节页面），则 Bit 63:13 是有效的，TA 将忽略隐含为 0 的 Bit 11:0。

注：Page Aligned Request 字段是在 ATS 规范的 Revision 1.1 中添加的。

10.2.2.5 No Write (NW) Flag

当 No Write (NW) 设置为 1 时，表示该 Function 正在请求对该转换的只读访问。

TA 可能会忽略 No Write Flag，但是，如果 TA 用标记为只读的转换来响应，则该 Function 不得使用该转换发出存储写事务。在这种情况下，Function 可能会发出另一个 NW 设置为 0 的转换请求，无论是否设置了 W（写入）位，都可能导致新的转换完成。

在收到 NW 为 0 的 Translation Request 后，允许 TA 将相关页面标记为脏。强烈建议 Function 除非发出明确的写许可，否则不要发出此类请求。

注：NW 字段是在 ATS 规范的 Revision 1.1 中添加的。

10.2.2.6 PASID TLP Prefix on Translation Request

如果 Translation Request 包含 PASID TLP Prefix，则 Untranslated Address 字段是由 PASID 字段指示的进程地址空间内的地址。

如果 Translation Request 的 PASID TLP Prefix 的 Privileged Mode Requested 或 Execute Requested 字段设置为 1，则可以在构造 Translation Completion Data Entry 时使用它们。

PASID Extended Capability 指示 Function 是否支持并启用了发送和接收带有 PASID TLP Prefix 的 TLP 的能力。

10.2.3 Translation Completion

TA 为每个 Translation Request 发送 Translation Completion (Cpl 或 CplID)。本规范描述了 Translation Completion 中各字段的含义。在本规范中未定义的字段具有与本规范中的 Read Completion 相同的含义。对于 Translation Completion，如果相应的 Translation Request 的 RO 设置为 1，则该 Translation Completion 的 Relaxed Ordering (RO) 字段允许设置为 1。Attr 字段的其余部分为保留值。

如果 TA 无法执行请求的转换，则使用 Figure 10-10 中所示格式的 Completion。

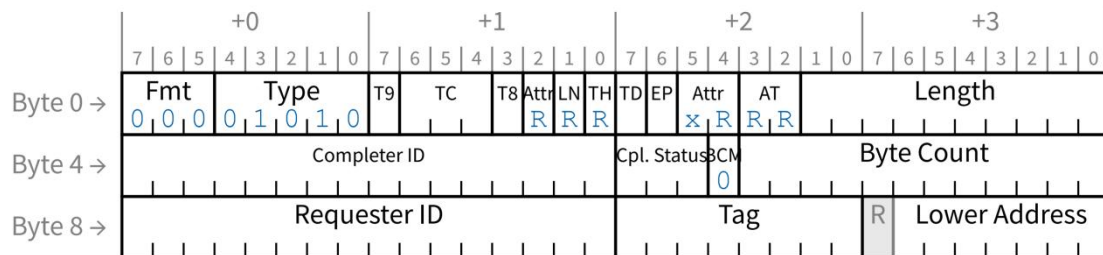


Figure 10-10 Translation Completion with No Data

Byte Count Field for Unsuccessful Translation Completions with No Data

该规范的先前版本指示对于没有数据的 Unsuccessful Translation Completion，Byte Count 和 Lower Address 字段应为 0000 0000 0000b。强烈建议实现不要依赖于把没有数据的 Unsuccessful Translation Completion 中的 Byte Count 和 Lower Address 字段设置为任何特定值这种方式。

Table 10-2 中列出了 Completion Status 字段的值和含义。

Table 10-2 Translation Completion with No Data Status Codes

Value	Status	Meaning
000b	Success	This Completion Status has a nominal meaning of “success”. The TA will not return this value in a Cpl.
001b	Unsupported Request (UR)	Translation Requests from this Function are not supported by the TA. If a Function receives this Completion code, it must disable its ATC and not send requests using translated addresses until the ATC is re-enabled. For transactions the Function may internally have in flight, the Function may either terminate or complete them. The mechanism a Function receiving this code uses to report this condition is outside the scope of this specification. The TA detecting this error is a “Completer Sending a Completion with UR/CA Status” and shall behave as defined in this specification.
010b	CRS	This value is not allowed in any Completion to a request initiated by a PCI Express Function. If received by a Function, it shall be treated as a Malformed TLP.
100b	Completer Abort (CA)	The TA was not able to translate the address because of an error in the TA. This nominally causes an error to be reported to the device driver associated with the ATC. See AER in this specification.
All others	Reserved	A Translation Completion with a Reserved Completion Status value is treated as if the Completion Status was Unsupported Request (001b).

注意：如果值不是 Success 就意味着一个 Error。

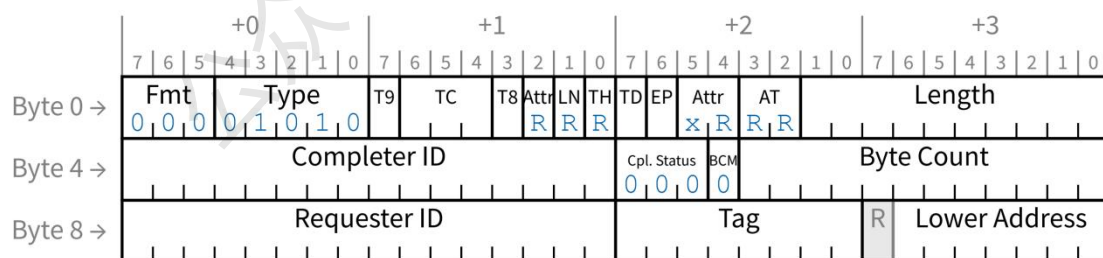


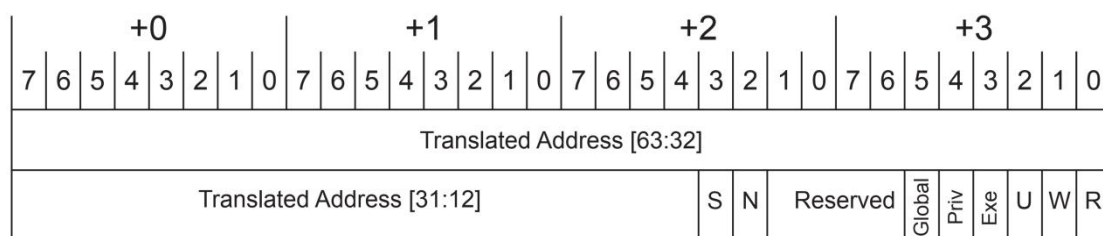
Figure 10-11 Successful Translation Completion

根据 Section 2.2.9 和 Section 2.3 设置各个字段。

必须使用与 Translation Request 相同的 TC 发送 Translation Completion。不需要 Function 验证是否使用了相同的 TC。

Lower Address 字段的值得此 packet 与 RCB 语义保持一致。如果结果在单个 packet 中返回，则 Lower Address 设置为 RCB 减去 Byte Count。如果在多个 packet 中返回结果，则第一个 packet 的 Lower Address 字段为 RCB 减去 Total Completion Length * 4，随后的 packet 的 Lower Address 字段为 000 0000b。有关多个 packet 的 Completion 的其他要求，请参阅 Section 10.2.4。

如果 Completion Status 字段为 000b，则转换成功，并且数据有效载荷将跟随报头。数据有效负载的内容如 Figure 10-12 所示。



A-0583A

Figure 10-12 Translation Completion Data Entry

Field	Meaning
S	Size of translation - 如果转换应用于 4096 字节的内存范围，则此字段为 0b。如果此字段为 1b，则转换应用于大于 4096 字节的内存范围（请参见 Section 10.2.3.1）。
N	Non-snooped accesses - 如果此字段为 1b，则使用此转换的读写请求必须清除 Attribute 字段中的 No Snoop 位。如果此字段为 0b，则该 Function 可以使用其他方式来判定 No Snoop 是否应该设置。
Reserved	ATC 会忽略该字段的值。
Global	Global Mapping - 如果该位置 1，则允许 ATC 在所有 PASID 中缓存此映射条目。如果为 0，则允许 ATC 仅在与发出请求的 PASID 关联的 PASID 中缓存此映射条目。仅当相关的 Translation Request 包含 PASID TLP Prefix 时，才可以设置此位。
Exe	Execute Permitted - 如果该位置 1，则允许发起请求的 Function 执行关联存储器范围中包含的代码。 仅当关联的 Translation Request 的 PASID TLP Prefix 的 Execute Requested 设置为 1 时，才可以设置该位。如果该位置 1，则还必须将 R 字段置 1。 Priv 位指示与 Exe 位关联的特权级别。如果 Priv 设置为 1，则 Exe 位指示与 Function 中的特权模式实体关联的权限。如果 Priv 为 0，则 Exe 位指示与 Function 中的非特权模式实体关联的权限。 如果 R 字段设置为 1，则可以缓存该值。
Priv	Privileged Mode Access - 如果该位置 1，则 R，W 和 Exe 表示与特权模式实体关联的权限。如果此位为 0，则 R，W 和 Exe 表示与非特权模式实体关联的权限。 仅当关联的 Translation Request 的 PASID TLP Prefix 的 Privileged Mode Requested 设置为 1 时，才可以设置该位。 R，W 或 Exe 值中的任何一个都必须被缓存。
U	Untranslated access only - 设置此字段时，只能使用未转换的地址访问指示的范围，并且在 AT 设置为 Translated 的后续读取/写入请求中，不得使用此 Translation Completion Data Entry 的 Translated Address 字段。如果设置了 R 或 W，则可以缓存该值。
R,W	Read, Write - 这两个字段指示使用转换的请求所允许的事务类型。编码是： 00b 不允许读写操作。该转换被认为是无效的。Translated Address，N，U 和 Exe 字段的内容未定义。带有

	此值的转换一定不能缓存在 ATC 中（请参阅 Section 10.2.3.5）。
01b	允许以该范围为目标写请求，但是除非读请求为零长度读取，否则不允许。
10b	允许针对该范围的读取请求（包括零长度读取），但不允许写入请求。
11b	允许以该范围为目标读写请求。
	Priv 位指示与 R 和 W 关联的权限级别。如果设置了 Priv，则 R 和 W 指示与 Function 中的 Privileged Mode 实体关联的权限。如果清为 0，则 R 和 W 表示与 Function 中 Non Privileged Mode 实体相关联的权限。

10.2.3.1 Translated Address Field

如果 R 和 W 字段均为 0，或者如果 U 字段设置为 1，则该 Function 可能不会出于任何目的使用 Translated Address 字段。

如果 R 或 W 字段已设置为 1，而 U 字段为 0，则 Translated Address 字段包含该 Function 可以在 Memory Request 中使用的地址，且 AT 字段设置为 Translated，并且该 Function 可以缓存已转换的地址。缓存后，必须将 R 和 W 字段存储为与 Translation Completion 条目相同的值。缓存的地址必须是 Translation Completion 中指示的地址范围的子集。

当转换后的地址缓存在 Function 的 ATC 中时，该 Function 只能删除该条目，而不能修改它。当收到的 Invalidation Request 的指示范围与缓存地址的任何部分重叠时，必须从 ATC 中删除该条目。

除非该 entry 在 Translation Completion 中并且 ATS Capability 的 E (Enable) 字段设置为 1，否则不允许某个 Function 把该 entry 缓存进其 ATC。设置 E 字段之前写入的 ATC 缓存中的 entry 不得在 Memory Request 中使用。当设置了 E 字段时，它们必须无效或被忽略而不使用。

10.2.3.2 Translation Range Size (S) Field

如果 S 字段设置为 1，则转换要应用于大于 4096 字节的范围。如果 S = 1b，则转换地址的第 12 位用于指示范围是否大于 8192 字节。如果第 12 位为 0b，则范围大小为 8192 字节，但如果为 1 则大于 8192 字节。如果 S = 1b 且 Bit 12 = 1b，则 Bit 13 用于确定范围是否大于 16384 个字节。如果 Bit 13 为 0b，则范围大小为 16384 字节，但如果为 1 则它大于 16384 字节。

按序消耗的低地址位用以指示与转换关联的范围的大小。

注意：此编码方法还用于指示无效的内存范围的大小。

Table 10-4 中显示了不同转换大小的示例。

Table 10-4 Examples of Translation Size Using S Field

Address Bits																				S	Translation Range Size in Bytes		
63:32 *	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13			12	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	4 K	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	8 K
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	16 K
x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	1	1	1	1	1	1	1	1	2 M
x	x	x	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 G
x	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4 G

Note:

* Upper address bits are used to indicate the size for ranges larger than 4 GB.

无论 STU 设置如何，都要将 S 字段设置为以 4096 字节的倍数表示的范围大小（即为 1）。例如，如果将 STU 设置为指示最小转换为 8192 字节，则应在 Translation Completion 和所有 Invalidate Request 中返回的所有转换上设置 S。如果将 STU 设置为指示最小 16384 字节，则 S 和 Bit 12 都将在所有转换范围和无效范围内设置。

如果设置了 S 且位 63:12 均为 1b，则行为未定义。如果设置了 S 且位 63 为 0b，而位 62:12 全部为 1b，则请求将使所有转换无效。

如果 Function 收到的 Translation Completion 的 Translation Size 字段小于该 Function 的已编程的 STU 值，则它将 Translation Completion 视为一个具有 UR 状态的 Completion。

10.2.3.3 Non-snooped (N) Field

设置此字段表示在此转换范围内目标内存的读取和写入请求必须清除请求报头中的 No Snoop 属性位。当此字段为 0b 时，允许 Function 以特定的方式设置 No Snoop 属性位。

注：清除此字段后，如果把 Device Control 寄存器中的 Enable No Snoop 字段清为 0，则该 Function 不允许在内存请求中设置 No Snoop。

如果设置了 R 或 W，则 ATC 可以缓存 N 字段。

U 设置为 1 时，此字段的含义不确定，TA 可以将该字段设置为任何值。转换对 N 字段具有单个值，该值不受特权级别的影响。允许 ATC 缓存 N 字段，而不考虑 Priv 位的值。

10.2.3.4 Untranslated Access Only (U) Field

当不允许该 Function 使用转换后的地址访问隐含的内存范围时，将把此字段设置为 1（该范围由 Translation Request 中的未转换地址和 Translation Completion 中的转换偏移量隐含）。该 Function 可以使用未转换的地址来访问范围，只要 R 和 W 字段允许访问即可。如果设置了 R 或 W，则 Function 可以缓存该转换值。如果设置了 U 字段，则转换中的 Translated Address 字段不一定是有效的内存地址，并且 Function 不能在 AT 设置为 Translated 的读或写请求中使用该值。

注意：此字段的一种可能用途是避免不必要的无效。如果某个 Function 对内存的某些部分使用转换的请求，而对其他部分不使用，则可以在不使用转换请求的部分上使用 U 字段。如果在 U 字段中设置了转换，则转换发生变化时，则不一定要求将 Invalidate Request 发送到 Function。这种用法的一个示例是带有用于命令的环形缓冲区的 Function。环形缓冲区可能会分配很长时间，并且具有很高的重用性（局部性）。因此，Function 在其内存请求中使用以命令缓冲区为目标的转换地址很有用。相同的 Function 可能会访问局部性差且重用率低的数据缓冲区。最好使用未转换的请求来处理对数据缓冲区的访问。设置数据缓冲区转换的 U

字段可确保转换不会尝试使用转换后的值来访问数据缓冲区，因此，当更改数据缓冲区映射时，不需要 Invalidate Request。当设置了 U 且设置了 R 或 W 时，允许 ATC 缓存 U，R，W Exe 和 Priv 以及转换范围大小（请参见 Section 10.2.3.2）。如果这些值发生更改，则需要 Invalidate Request。

10.2.3.5 Read (R) and Write (W) Fields

这些字段指示返回的转换值是否可以在读或写内存请求中使用。如果清除了 R 字段，则 ATC 可能不会使用转换值发出非零读取请求。如果 W 字段被清除，则 ATC 可能不会使用转换值发出写请求。如果 R 和 W 字段均未设置，则 ATC 可能不会使用转换值发出任何类型的请求。如果同时清除了 R 和 W 字段，则仍指示转换范围，但是转换中其他值的含义不确定。

注意：即使 $R = W = 0b$ ，也要指明转换条目的范围，以便在 Translation Completion 中留出一个“hole”。例如，如果 Translation Request 的长度为六个 DW，则 Translation Completion 中最多可以包含三个转换。第一个和第三个转换可能把 R 或 W 设置为 1，但第二个转换可能设置 $R = W = 0b$ 。为了避免对指示的 hole 的大小产生歧义，即使 $R = W = 0b$ ，在 Translation Completion 中也会指示 hole 的范围。

$R = 0b$ ， $W = 0b$ 状态用于指示转换中的地址字段可能不用于形成后续请求的转换后的地址值。

当主机改变 TA 中的转换以使得转换存在时，主机不需要向 ATC 发送无效指示，以便主机知道转换状态的变化。由于可能无法将转换的变化通知 ATC，因此不得缓存转换值 $R = W = 0b$ 。

如果未找到所请求地址的 table entry，则 TA 将返回具有单个转换值的 CplID，其中 $R = W = 0b$ 。

注意：实现不应假定接收到设置了 R 或 W 位（与 U 位的值无关）的转换响应意味着随后的具有相同未转换地址的读取或写入请求将成功。尽管设备及其控制软件可能会确保此属性，但是这样做的方法不在本规范的范围之内。

Priv 位用于限定 R 和 W。如果设置了 Priv，则 R 和 W 表示授予该 Function 中特权模式实体的权限。如果清为 0，则 R 和 W 表示在 Function 中授予非特权模式实体的权限。两个特权级别的 R 和 W 值是独立的。ATC 不得在与转换相关联的“特权模式”和“非特权模式”权限之间假定任何关联。

10.2.3.6 Execute Permitted (Exe)

如果设置了 Exe，则允许发出请求的 Function 在隐含的内存范围内执行代码。如果 Exe 为 0，则不允许发出请求的 Function 在隐含的内存范围内执行代码。

Function 执行代码的含义的定义超出了本规范的范围。各种系统组件可能具有不同的指令集。当发出请求的 Function 尝试执行此位不允许的代码时，其行为不属于本规范的范围。

只能设置 Exe 位，TA 支持执行权限，相关的 Translation Request 的 PASID TLP Prefix 的 Execute Requested¹⁶⁸ 字段的有效值为 1b，并且在 Translation Completion Data Entry 中设置 R。否则，Exe 位必须清零。

注 168：该位的有效值为 0b，除非在 PASID TLP Prefix 中把该位设置为 1b，并且已为请求使能该位的使用。

如果 R 设置为 1，则可以缓存该值。

Priv 位用于限定 Exe 位。如果设置了 Priv，则 Exe 位表示授予该 Function 中的特权模式实体的权限。如果清为 0，则 Exe 位表示该 Function 中授予非特权模式实体的权限。两个特权级别的 Exe 位值是独立的。ATC 不得在与转换相关联的“特权模式”和“非特权模式”权限之间假定任何关联。

Function 可以选择检查：

- 如果 Translation Request 中的 Execute Requested 字段为 0，则关联的 Translation Completion Data Entry 中的 Exe 位也为 0。

- 如果 Exe 设置为 1，则 R 也设置为 1。

如果任一项可选检查均失败，则该 Function 将发出 Unexpected Completion (UC)。这些检查是独立可选的。

10.2.3.7 Privileged Mode Access (Priv)

如果 Priv 设置为 1，则 R、W 和 Exe 表示的是发出请求的 Function 中以特权模式操作的实体的权限。如果 Priv 设置为 0，则 R、W 和 Exe 是指在发出请求的 Function 中在非特权模式下运行的实体的权限。

Privileged Mode 和 Non-Privileged Mode 的含义以及实体以 Privileged Mode 和 Non-Privileged Mode 运行的含义取决于系统的保护模型，不在本规范范围之内。

当发出请求的 Function 中的实体尝试访问不允许其访问的内存时，行为超出了本规范的范围。

仅当 TA 支持特权模式并且相关的 Translation Request 的 PASID TLP Prefix 的 Privileged Mode Requested¹⁶⁹ 字段为有效值 1b 时，才可以将 Priv 位置 1。否则，Priv 位必须清零。

注 169：该位的有效值为 0b，除非前缀中的该位为 1b，并且已使能该位用于请求。

R、W 和 Exe 的特权和非特权模式版本是独立的。ATC 可以缓存 R、W 和 Exe 的一个或两个版本。对于一个特权级别，接收到 R = W = 0b 的转换的 ATC 可能不会假设其对于另一特权级别可能收到的内容。

如果设置了 R 或 W，则可以缓存该值。缓存相应的 R、W 或 Exe 值时，必须缓存该值。

注意：由于仅当发出请求的 Function 设置了 Privileged Mode Requested 字段时才将 Priv 位置 1，因此从未设置该位的 Function 应始终将 Priv 位清零，因此无需对其进行缓存。

Function 可以可选地进行检查，当 Translation Request 中的 Privileged Mode Requested 字段为 0 时，关联的 Translation Completion Data Entry 中的 Priv 位也为 0。如果该可选检查失败，则该 Function 将发出 Unexpected Completion (UC)。

Execute Permission and Privilege Mode Enforcement

发出请求的 Function 确定特定的内存请求是否需要执行权限或与特权模式实体或非特权模式实体相关联。ATC 执行由 Exe 和 Priv 位指示的保护检查。

10.2.3.8 Global Mapping (Global)

如果 Global 字段设置为 1，则允许发出请求的 Function 在 ATC 中为此转换创建 Global Mapping 条目。如果 Global 为 0，则不允许发出请求的 Function 在 ATC 中为此转换创建 Global Mapping 条目。Global Mapping entry 适用于该 Function 的所有 PASID。它们允许 ATC 减少所需的转换请求的数量，并减少缓存结果所需的内存。

允许 Function 忽略该位，并始终在 ATC 中创建非 Global Mapping entry。这可能会导致在不同的 PASID 下针对同一未转换地址请求进行多次转换。

使用该位的 Function 还必须把 Global Invalidate Supported 字段设置为 1（Section 10.5.1.2）。

10.2.4 Completions with Multiple Translations

允许 ATC 请求 TA 为实际上连续的地址范围提供转换。只要将请求大小 (Total Completion Length * 4) 不大于 Max_Read_Request_Size 或 RCB，就可以通过将 Translation Request 中的 Length 字段设置为所请求转换次数的两倍的值来实现此目的。

如果请求多个转换，则 TA 可以返回一个或多个转换，只要转换数量不超过请求的转换数量即可。TA 返回的转换数量少于请求的

数量，这不是错误，除非访问数据有错误，否则不会发送错误指示。

如果 Translation Completion 包含多个转换，则所有转换必须指示相同的大小。同样，连续的转换必须应用于与同一 Completion 中的先前转换相邻的虚拟地址范围。

如果转换同时具有 $R = 0b$ 和 $W = 0b$ ，则 TA 必须仍将 Size 字段和用于对完成大小进行编码的 Translated Address 字段的低位设置为适当的值。

Translation Completion 中的每个转换将与 Translation Request 的隐含存储范围有一些重叠（see Section 10.2.2）。

成功的 Translation Completion 必须包含一个或两个 CplID。

如果 Translation Completion CplID 的字节计数大于 Length 字段的四倍，则需要其他 CplID 来完成事务。

如果 Translation Completion CplID 的字节计数等于 Length 字段的四倍，则数据包将完成请求。对于这样的 CplID，如果字节计数和低位地址的总和不是 RCB 的倍数，则 CplID 是序列的最后一个。如果没有收到该请求的先前 CplID，则发生错误，并且所有转换值都应丢弃。

注意：有许多原因可导致 TA 截断完成结果。例如，该请求可能要求提供一个地址范围，但并非所有地址都已定义。如果第一个转换有效但位于转换页面的末尾，则可能会发生这种情况。TA 在查找转换的下一页时，可能会发现该页面无效，因此地址无效。将返回有效的地址范围，并且未显示任何错误。截断 Translation Completion 时，不允许 TA 用无效的条目填充响应（ $R = 0b$ ， $W = 0b$ ）。

注意：TA 可能有多种原因将 Translation Completion 分成多个 TLP。例如，如果 Translation Completion 的虚拟地址解析为跨存储器系统的 RCB 边界的表访问，则 TA 的 Completion 可能会被存储器分为多个 Completion。当从系统内存中接收到 TA 时，它可以将 Translation Completion 的每个部分发送給一个 Function，而不是要求 TA 累积结果。

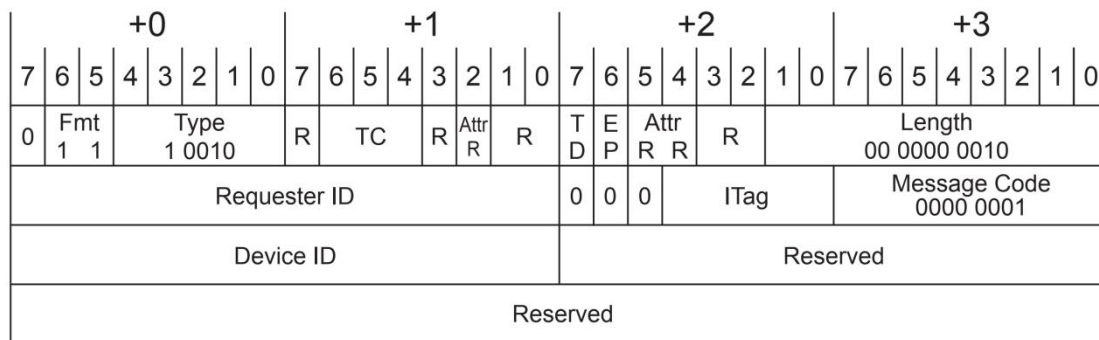
10.3 ATS Invalidation

ATS 使用本节中定义的消息来维护 TA 和 ATC 之间的一致性。该规范假设每个 ATC 都有一个关联的 TA。TA（及其关联软件）必须通过发出 Invalidate Request 来确保 ATC 中缓存的地址转换不会过时。

10.3.1 Invalidate Request

当 TA 中的转换发生变化，并且该转换可能包含在 Function 的 ATC 中时，TA（及其关联软件）必须向 ATC 发送 Invalidate Request，以保持 ATPT 和 ATC 之间的正确同步。Invalidate Request 用于从 ATC 中清除地址范围的特定子集。Invalidate Request 被限制为覆盖 4096 字节页面的 2 倍的幂。

Invalidate Request 的格式如 Figure 10-13 所示。



A-0584A

Figure 10-13 Invalidate Request Message

Invalidate Request 是具有 64 位数据的 MsgD 事务。Invalidate Request Message 可以在任何 TC 中发送。ITag 字段的值限制为 0 到 31, TA 用此字段唯一地标识它发出的请求。一个 TA 必须确保一旦使用了 ITag, 它就不会被重用, 直到被相应的 Invalidate Completion 或特定于供应商的超时机制释放为止 (请参见下文)。

对于它发出的所有无效标记, TA 可能拥有一个 ITag 值池, 或者可能为每个 Device ID 或任何其他组合拥有一个池。具有不同 Function 的多个 ATC 的设备必须为每个 Requester ID 分别管理 ITag。

Invalidate Request 中指定的地址范围可能跨越一个或多个 STU 4096 字节页面。无效范围要求自然对齐, 并且不得小于 STU 4096 字节的页面。在收到范围小于 STU 的 Invalidate Request 时, ATC 可以 (1) 发送 Unsupported Request, 或者 (2) 将请求的范围舍入为大于或等于 STU 的值。

Invalidate Completion Timeout

设备应在 1 分钟内 (+ 50%-0%) 响应 Invalidate Request。具有一定的时间允许 ATPT 执行无效完成超时并重用关联的 ITag 值。ATPT 设计是特定于实现的。因此, “无效完成超时”及其相关的错误处理不在本规范的范围之内。

有效载荷的内容是即将无效的未转换的地址范围。有效载荷格式显示在 Remove Word {RD} 标签中, 该标签用于帮助生成 ToC / ToF / ToT。

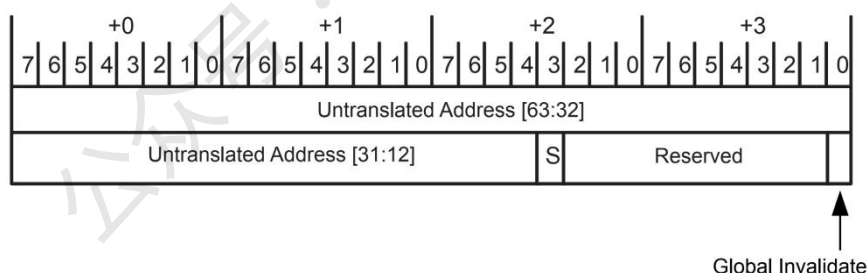


Figure 10-14 Invalidate Request Message Body

S 字段用于指示无效范围是否大于 4096 字节。其含义与 Translation Completion 中的含义相同 (请参见 Section 10.2.3.1 和 Section 10.2.3.2)。

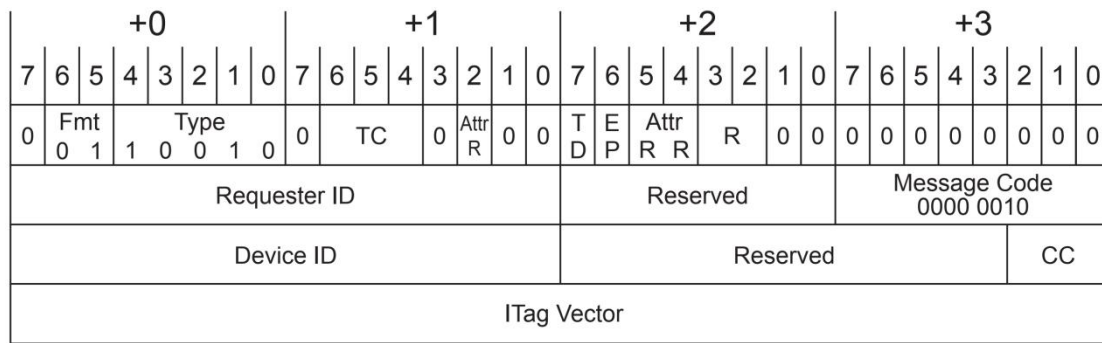
Global Invalidate 字段表示 Invalidate Request Message 影响所有 PASID 值 (请参见 Section 10.3.8)。除非 Invalidate Request 具有 PASID TLP Prefix, 否则该位保留。如果 Global Invalidate Supported 字段被清 0, 则该位将被 ATC 忽略 (请参见 Section 10.3.8)。

10.3.2 Invalidate Completion

Function 完成无效操作后, 它将向 TA 发送一个或多个 Invalidate Completion Message。必须使用从 Invalidate Request 中提取的信

息来标记这些消息，以使 TA 能够将 Invalidate Completion 与 Invalidate Request 相关联。

Invalidate Completion Message 的格式如 Figure 10-13 所示。



A-0586A

Figure 10-15 Invalidate Completion Message Format

Invalidate Completion Message 是按 ID 路由的 Msg 事务。Invalidate Completion Message 的 Requester ID 字段设置为包含 ATC 的 Function 的 Requester ID。Invalidate Completion 的 Device ID 字段设置为 TA 的 Requester ID。ATC 可以从相应的 Invalidate Request 的 Requester ID 字段获得 TA 的 Requester ID。可替代地，由于 ATC 仅与单个 TA 相关联，所以 ATC 可以在基本复位或 FLR 之后从第一个 Invalidate Request 采样并存储 Requester ID。随后的 Invalidate Completion Message 可以使用此值来设置 Invalidate Completion Message 的 Device ID 字段。

Completion Count (CC) 字段表示必须为关联的 Invalidate Request 发送的 Invalidate Completion Message 的数量。将 CC 字段设置为 0 表示必须发送八个响应。在负责完成相应的 Invalidate Request 之前，TA 负责收集与给定标签关联的所有响应。

可以在任何 TC 上发送 Invalidate Completion Message，而与接收到原始 Invalidate Request 的 TC 无关。这使实现能够利用 Invalidate Completion 将未完成的事务推送到 TA，以确保满足所需的无效语义。需要使用单个上游 TC 的实现才能在所使用的 TC 中发送单个 Invalidate Completion。

ITag 向量字段用于指示哪个 Invalidate Request 已完成。来自 Invalidate Request 的 32 个可能的 ITag 字段值中的每一个均由 ITag 向量字段中的单个位表示。ITag 向量字段的最低有效位 (Bit 0；即，Figure 10-13 中所示的 Invalidate Completion 消息的示意图中最右边的位) 对应于 ITag 字段值 0。最高有效位 (ITag 向量字段的第 31 位) 对应于 ITag 字段值 31。允许实现通过在单个消息中设置多个 ITag 向量位来合并多个 Invalidate Completion，只要满足以下条件：

- Invalidate Completion 在同一 TC 中流动。
- Invalidate Completion 具有相同的 CC 值。
- Invalidate Completion 的所有片段都必须具有相同的 Requester ID，CC 和 ITag 向量字段。

一个 TA，它收到了一个 Invalidate Completion，此 Invalidate Completion 的 ITag 表示没有 outstanding Invalidation Request 存在，则应使用特定于实现的机制报告此错误。一种可能的机制是将此 Invalidate Completion 报告为 Unexpected Completion (UC)。

不支持 ATS 的 Function 会将 Invalidation Request 视为 UR。

无论是否设置了 Bus Master Enable 字段，支持 ATS 的 Function 都需要发送 Invalidate Completion 以响应 Invalidation Request。请注意，即使 Bus Master Enable 字段为 0，也必须满足上述条件。设备实现此操作的方法取决于实现。

Bus Master Enable Change

当 Bus Master Enable 从 1 更改为 0 时，不应将其他内存请求排队。清除 BME 时，可能存在排队的写请求。这些请求可能会阻塞 Invalidate Completion。这些请求必须被发送或丢弃。这将确保所有可能依赖于 outstanding invalidation 的 outstanding write transaction 均已完成。

10.3.3 Invalidate Completion Semantics

在 ATC 可以针对给定的 Invalidation Request 返回 Invalidate Completion 之前，它必须确保满足以下条件：

- 该 Function 发起的所有新请求都不会使用过时的地址转换。
- 所有使用与无效范围匹配的转换地址的 outstanding read request 都已完成或被标记为要丢弃（丢弃方法是特定于实现的）。
- 使用与无效范围匹配的转换地址的所有 outstanding posted write 均已推送到 TA。要求 ATC 在 posted write 已发送但未将其推送到 TA 的每个 TC 中发送 Invalidate Completion 消息的副本。在 Invalidate Completion 消息的每个副本中，必须将 CC 字段设置为相同的值，以指示已发送的副本数。TA 负责收集所有已发送的响应，然后再考虑将其视为无效。

Implied TC Flushing

当做出关于哪个 TC 发送 Invalidate Completion 的决定时，ATC 可以以特定的方式推断出已发送的 posted write 已被推送到 TA。例如，已经将读取事务发送到 TA 上方的目的地并接收到其相应响应的 Function 可以推断出同一 TC 中发送的任何先前的 posted write 已被推送到 TA。

10.3.4 Request Acceptance Rules

根据本节中列举的请求接受规则，不允许 Function 创建依赖关系，其中对 posted transaction 的接受取决于 posted transaction 的传输。鉴于 Invalidation Request 和 Invalidate Completion 都是 posted transaction，Function 不得依赖 Invalidate Completion 的传输来接受 Invalidation Request。实现此目的的方法是特定于实现的。

即使未启用 ATS，在其配置空间中具有 ATS Capability 的 Function 也必须能够接受 Invalidation Request 并发送 Invalidate Completion。

Invalidate Queue Depth

ATC 仅与单个 TA 相关联。对于任何给定的 ATC，每个 TA 总共限制为 32 个 outstanding invalidation。这将单个 ATC 上未激活的 outstanding Invalidation Request 的数量限制为 32。为了避免事后依赖性，要求 ATC 最多接受 32 个 Invalidation Request。

ATC 可以选择实现容纳 Invalidation Request 的最大大小的输入队列。可替代地，ATC 可以选择实现保持 Invalidate Completion 的最大大小的输出队列。请注意，对 Invalidate Completion 进行排队对每个条目所需的状态要少得多，从而导致实现比输入队列缓冲更有效的实现。

请注意，选择实现输入排队还是输出排队（或两者混合）对确保无死锁行为没有影响。但是关于排队的实现选择可能会对性能产生重大影响（请参见 Section 10.3.5）。

10.3.5 Invalidate Flow Control

由于缓存体系结构和排队策略的多样性，关于无效等待时间和吞吐量的实现可能会有很大不同。TA 可能会以超过平均 ATC 服务速率的速率生成 Invalidation Request。发生这种情况时，基于信用的流控制机制将限制 TA 的发送速度。这样做的副作用是拥塞会通过基于信用的流控制机制传播到其他通道和链路。根据此拥塞的频率和持续时间，性能可能会受到影响。强烈建议 TA 及其相关软件实现更高级别的流控制机制。

为了协助实现 Invalidate Flow Control，ATC 必须在链路反压之前发送它可以缓冲的 Invalidate Request 数。该值会应用于该 Function 服务的所有失效，而与失效的大小无关。该值在 ATS Capability structure 的 Invalidate Queue Depth 字段中传达（请参见 Section 7.8.8）。值 00000b 表示 Invalidate Flow Control 对此 Function 不是必须的。

Invalidate Flow Control

如果满足以下一项或多项，则 Function 可能表明不需要 Invalidate Flow Control：

1. 该 Function 可以以 Invalidate Request 的最大到达率来处理无效。
2. 该 Function 不会或很少会引起链路反压（性能损失可忽略不计）。
3. 该 Function 可以完全缓冲传入的最大数量无效消息，而无需反压链路。

10.3.6 Invalidate Ordering Semantics

Invalidate Request 和 Translation Completion 可能使用不同的 TC 发送，因此彼此之间无顺序（从链路的角度来看）要求。当 Invalidate Request Bypass 掉 Translation Completion 重叠区域时，ATC 必须确保维持适当的无效行为。

ATC 必须针对所有到达的 Invalidate Request“监听”其 outstanding translation request 队列。侦听 $N * \text{STU}$ 大小的 translation request（ N 为 2 的幂）时，ATC 必须侦听地址范围，该地址范围从包含指定地址的 STU 对齐区域开始，到以后 $(N-1)$ 个 STU 大小页面结束。

如果 Invalidate Request 与 outstanding Translation Request 中的地址范围重叠，则在发送 Invalidate Completion 之前，必须将 Translation Request 标记为无效，并且必须丢弃其对应的 Translation Response 的结果。如果在发送 Invalidate Completion 之前收到 Translation Response，则实现满足 Invalidate Completion 语义（请参见 Section 10.3.3）的情况下，实现可以自由地使用转换结果发出请求。

Request Range Overlap in Invalidations

在上面的描述中， N 是 Translation Request 中请求的 STU 大小转换的数量。等于 $(\text{Length field in Translation Request}) / 2$ 。

一个示例为：

STU 为 000010b 表示一个 16384 字节的页。

outstanding Translation Request 的 Length 字段为 00 0000 0100b，表示两个 Translation，覆盖 32768 字节的范围。

Translation Request 的高 48 位为 0000 0FFF FFFFh。

请求中地址的低 16 位为 11xx xxxx xxxx b，表示 Translation Request 覆盖的范围与 32768 字节边界重叠（实际上，该请求跨越 16 TB 边界）。

如果返回两个 Translation，则它们将覆盖两个 STU 大小的区域，分别为 0000 0FFF FFFF C000h 和 0000 1000 0000 0000h。

接收到 Invalidate Request，其高 48 位为 0000 1000 0000h，低 16 位为 0001 1xxx xxxx b。

ATC 必须检测到与 Translation Request 的一部分相关的转换现在已失效，并且与无效区域关联的 Translation Completion 必须被丢弃（为简化起见，允许 ATC 丢弃所有 Translation Completion）。

应当注意，如果 Translation Request 没有越过请求的对齐边界，则简化了 Invalidate Request 的处理。上例中的 Translation Request 未与 32768 字节边界对齐。如果将其分为两个请求，则将 Invalidate Request 的范围与 Translation Request 中的地址关联起来会更简单。不需要将 Translation Request 分解为对齐的请求。

10.3.7 Implicit Invalidation Events

以下事件将导致所有 ATC 条目无效：

- Conventional Reset (all forms)

- Function Level Reset
- ATS Capability 的 E 字段从 0 变成 1。

以下事件将导致使用特定 PASID 请求的所有非全局映射 ATC 条目无效：

- 停止使用本规范中定义的 PASID。

当这些隐含的无效事件发生时，不会发送任何明确的 Invalidate Completion 消息。

Implicit Invalidation and PASID

当设置了 ATS Capability 中的 E 字段时，软件可能不会更改任何 PASID 使能位。当软件设置 E 字段时发生的无效也使具有关联的 PASID 值的 ATC 条目无效。

10.3.8 PASID TLP Prefix and Global Invalidate

本节中的要求适用于支持 PASID TLP Prefix 的 Function。对于包含 PASID TLP Prefix 的 Invalidation Request，ATC 应：

- 如果关联的 PASID 值大于或等于 $2^{\text{Max PASID Width}}$ ，则可以选择发送 Unsupported Request (UR)。每当存在超出范围的 PASID 值时，即使忽略了 PASID 值，也可能会发出此错误信令（请参阅下文）。
- 如果 PASID Enable 字段为 0，则返回 Invalidation Completion。
- 如果 Function 支持 Global Invalidate（请参阅 Section 7.8.8.2）：
 - 如果请求中的 Global Invalidate 字段被置 1，则在与任何 PASID 值相关联的指定存储范围内，使 ATC 中的全局和非全局映射条目无效，并返回 Invalidation Completion。PASID TLP Prefix 中的 PASID 值将被忽略。
 - 如果请求中的 Global Invalidate 字段被置 0，则仅使使用关联的 PASID 值请求的在指示的内存范围内的 ATC 中的非全局映射条目无效，并返回 Invalidation Completion。

对于某些或所有指示的存储范围，ATC 中的全局映射条目可能会保留。

- 如果 Function 不支持 Global Invalidate（请参阅 Section 7.8.8.2），请使用关联的 PASID 值使 ATC 中指定存储范围内的条目无效，并返回 Invalidation Completion。
- 如果 ATC 中没有匹配的条目，则使所有 ATC 条目无效并返回 Invalidation Completion。

对于没有 PASID TLP Prefix 的 Invalidation Request，ATC 应：

- 在没有 PASID 值的情况下，使指示的内存范围内的 ATC 条目无效。
- 使使用任何 PASID 值请求的所有地址的 ATC 条目无效。

10.4 Page Request Services

Page Request 的一般模型如下：

1. 一个 Function 判定它需要访问一个页面，但是该页面的 ATS translation 不可用。
2. 该 Function 使相关的 Page Request Interface 向其 RC 发送 Page Request Message。Page Request Message 包含页面地址和 Page Request Group (PRG)索引。PRG 索引用于标识事务，并用于将请求与响应进行匹配。
3. 当 RC 确定其对请求的响应（通常将使请求的页面驻留）时，它将 PRG Response Message 发送回发出请求的 Function。

4. 然后, Function 可以使用 ATS 来请求转换所请求的页面。

Page Request Message 是一种 PCIe Message Request, 它的消息代码为 4 (0000 0100b), 它是向 RC 路由的。RC 用来缓冲请求的机制是特定于实现的。唯一的要求是 RC 不能静默丢弃请求。

所有 Page Request Message 和 PRG Response Message 均以 PCIe Traffic Class 0 传输。Traffic Class 不为 0 的 Page Request Message 或 PRG Response Message 应被接收到的 RC 或 Endpoint 视为 Malformed TLP。中间路由元件 (例如, Switch) 不得检测到此错误。

可以使用 Page Request Message 和 PRG Response Message 的 Attr 字段中的 Relaxed Ordering 和 ID-Based Ordering 字段。Attr 字段中的 No Snoop 位保留。

Page Request Service 允许将页面请求分组为页面请求组 (PRG)。PRG 可以包含一个或多个页面请求。主机对 PRG 中的所有页面进行整体响应。通过独立的 Page Request Message 来请求 PRG 中的各个页面, 并通过共享相同的 PRG 索引将其识别为属于公共 PRG。PRG 的最后一个请求在其 Page Request Message 中进行了标记。每个页面请求 (而不是每个 PRG) 消耗一个请求信用。

PRG Response Message 是一种 PCIe Message Request, 它通过 ID 路由回发出请求的 Function。系统软件使用它来提醒 Function 已满足与相应 PRG 相关的页面请求。页面请求机制不保证任何请求完成顺序, 并且所有请求本质上独立于所有其他同时发生的 outstanding request。如果某个 Function 要求某个特定请求在另一个请求之前完成, 则初始请求需要在发出后续请求之前完成。对于 Function 而言, 在不确定其驻留状态的情况下推测性请求页面和/或对同一页面发出多个并发的 outstanding request 是有效的。

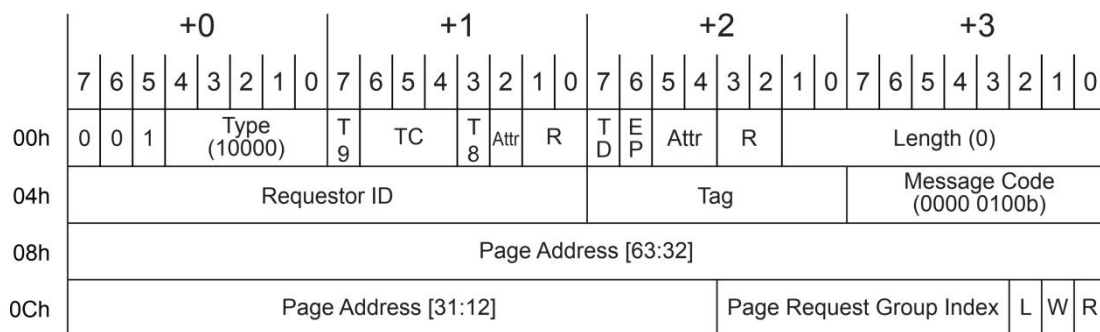
为 Page Request Interface 分配了特定数量的页面请求消息信用。RC (系统软件) 可以以任何适当的方式划分可用的信用。主机选择采取的任何措施, 以确保通过 Page Request Interface 正确计数信用 (Page Request Interface 使用的资源不多于其分配) 是一种实现选择。不允许 Page Request Interface 超额预订可用的请求数 (这样做可能会导致页面请求机制在根目录超过缓冲区限制的情况下被禁用)。Page Request Interface 的页面请求分配是静态的。确定何时启用 Page Request Interface, 并且只能通过禁用然后重新启用接口来进行更改。

10.4.1 Page Request Message

Function 使用 Page Request Message 将页面请求发送到其关联的主机。页面请求表示 Function 需要的页面。与 Function 关联的 Page Request Interface 被赋予特定的页面请求分配。Page Request Interface 不得发送超出其页面请求分配的页面请求。

页面请求包含所需页面的未转换地址, 该页面所需的访问权限以及 PRG 索引。PRG 索引是 Function 分配的 9 位标量, 用于标识关联的页面请求。可以使用单个 PRG 索引请求多个页面。当与一个给定的 PRG 关联的页面不止一个时, 除与给定 PRG 关联的最后一个请求外, 所有请求中的页面请求记录中的最后一个标志都会被清除 (该标志在最后一个请求中设置)。页面请求被大量响应。在根用户未收到 PRG 的最后一个请求之前, 没有响应是可能的 (响应失败错误除外)。Function 在任何给定时间可能未完成的 PRG 数量小于或等于关联的页面请求界面的未完成页面请求分配。一个请求组包含对同一页面的多个请求, 并且多个未完成的 PRG 请求同一页面是有效的。

Page Request Message 的前两个 DW 包含一个标准 PCIe 消息头。消息的后两个 DW 包含页面请求特定的数据字段。



A-0737A

Figure 10-16 Page Request Message

Field	Meaning
R	Read Access Requested - 为 1 时，此字段表示发出请求的 Function 对关联页面的读访问。为 0 时，此字段表示发出请求的 Function 将不会读关联的页面。 必须为带有 PASID TLP Prefix 且 Execute Requested 字段设置为 1 的 Page Request 设置 R 字段。 如果 R 和 W 都清零并且 L 被置位，则这是一个 Stop Marker（请参见 Section 10.4.1.2.1）。
W	Write Access Requested - 为 1 时，此字段表示发出请求的 Function 对关联页面的写访问和/或零长度读访问。为 0 时，此字段表示发出请求的 Function 将不会写关联的页面。 在接收到 W 字段设置为 1 的 Page Request Message 后，允许主机将关联的页面标记为脏。 因此，除非 Function 已获得明确的写许可，否则 Function 不得发出此类请求。 如果 R 和 W 都清零并且 L 被置位，则这是一个 Stop Marker（请参见 Section 10.4.1.2.1）。
L	Last Request in PRG - 为 1 时，此字段表示关联的页面请求是关联的 PRG 的最后一个请求。PRG 可以具有单个条目，在这种情况下，PRG 包含一个设置了该字段的单个请求。 为 0 时，此字段表示将使用此记录的 PRG 索引发送其他页面请求。 如果 R 和 W 都清零并且 L 被置位，则这是一个 Stop Marker（请参见 Section 10.4.1.2.1）。
Page Request Group Index	Page Request Group Index - 该字段包含 Function 提供的关联页面请求的标识符。Function 不需要使用 PRG 索引值的整个可用范围。主机绝不能使用 Function 先前未发送的 PRG 索引进行响应，并且当前还不是 outstanding request PRG Index（发出响应失败时除外，在这种情况下，主机无需保留关联请求的 PRG 索引值在错误响应中）。
Page Address	Page Address - 该字段包含要加载的页面的未转换地址。对于大于 4096 字节的页面，将忽略此字段的最低有效位。例如，当请求 8096 字节的页面时，将忽略该字段的最低有效位。

Last Bit and Relaxed Ordering

如果多个页面请求与一个 PRG 索引相关联，则 PRG 的最后一个页面请求除了应设置 Last 标志外，还应把 Relaxed Ordering 属性

位置 0。所有其他页面请求消息都可以将 Relaxed Ordering 属性位设置为任何值。

10.4.1.1 PASID TLP Prefix Usage

PASID Extended Capability 指示 Function 是否支持 PASID TLP Prefix，以及是否启用了发送和接收它们的能力。

允许支持 PASID TLP Prefix 的 Function 在 Page Request Message 上发送 PASID TLP Prefix。PASID 字段包含正在请求的页面的进程地址空间，并且 Execute Requested 和 Privileged Mode Requested 字段指示正在请求的访问。

如果 PRG 中的一个 Page Request Message 具有 PASID TLP Prefix，则该 PRG 中的所有 Page Request Message 必须包含相同的 PASID TLP Prefix。当 PASID TLP Prefix 不一致时，行为是不确定的。

支持 PASID TLP Prefix 并 PRG Response PASID Required 字段置 1（请参见 Section 10.5.2.3）的 Function，如果相关的 Page Request Message 具有 PASID TLP Prefix，则 PRG Response Message 将包含 PASID TLP Prefix。对于此类 PRG Response Message，Execute Requested 和 Privileged Mode Requested 字段的值为保留值，并且 PASID 字段包含来自关联的 Page Request Message 的 PASID。

10.4.1.2 Managing PASID TLP Prefix Usage on PRG Requests

这里给出停止和开始使用 PASID 的规则。

本节描述了适用于已停止使用 PASID 时发出 Page Request Message 的 Function 的其他规则。不需要其他规则即可开始为 PASID 使用 Page Request Interface。

当停止使用特定的 PASID 时，可以选择使用 Stop Marker Message，以避免在 Function 指示特定 PASID 的停止请求完成之前等待 PRG Response Message。

要在不使用 Stop Marker Message 的情况下停止，该 Function 应：

1. 停止为此 PASID 把新的 Page Request Message 推进队列。
2. 完成为此 PASID 的所有多页 Page Request Message 的发送（即，发送设置了 L 位的 Page Request Message）。
3. 等待 PRG Response Message 与 PASID 的任何 outstanding Page Request Message 相关联。
4. 表示已使用设备特定的机制停止了 PASID。该机制必须指示将不会生成 Stop Marker Message。

要使用 Stop Marker Message 停止，该 Function 应：

1. 停止为此 PASID 把新的 Page Request Message 推进队列。
2. 完成为此 PASID 的所有多页 Page Request Message 的发送（即，发送设置了 L 位的 Page Request Message）。
3. 在内部将此 PASID 的所有 outstanding Page Request Message 标记为 stale。与这些请求关联的 PRG Response Message 将返回页面请求分配信用和 PRG 索引值，否则将被忽略。¹⁷⁰
4. 表示已使用设备特定的机制停止了 PASID。该机制必须指示将生成 Stop Marker Message。
5. 发送 Stop Marker Message，以向主机指示此 PASID 的所有后续 Page Request Message 都将用于 PASID 值的新用途。

Note：Step 4 和 5 可以以任何顺序或并行执行。

注 170：页面请求分配在 Function 的所有 PASID 之间共享（请参见 Section 10.5.2.5）。如果 PRG Response PASID Required 为 0，则 PRG 索引值将在该 Function 的所有 PASID 之间共享（请参见 Section 10.5.2.3）。

10.4.1.2.1 Stop Marker Messages

Stop Marker Message 指示 Function 已停止使用 Page Request Interface，并已发送了针对特定 PASID 的所有挂起的 Page Request Message。Stop Marker Message 相对于 Page Request Message 是有序排列的，用于将 Page Request Message 推向主机。当主机接收到 Stop Marker Message 时，这表明与已停止的 PASID 关联的所有 Page Request Message 均已传递，并且任何具有相同 PASID 值的后续 Page Request Message 均与该 PASID 值的新形式相关联。

Stop Marker Message 没有响应。它们没有 PRG 索引，也不消耗页面请求分配（请参见 Section 10.5.2.5）。

Stop Marker Message 的布局如 Figure 10-17 所示。

	+0								+1								+2								+3										
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
00h	1	0	0	1	0	0	0	1	R	R	R	PASID																							
04h	0	0	1	Type (10000)				T 9	TC		T 8	Attr	R	T D	E P	Attr	R	Length (0)																	
08h	Requester ID														Tag								Message Code (0000 0100b)												
0Ch	Reserved																																		
10h	Reserved																								Marker Type (00000b)				L 1	W 0	R 0				

Figure 10-17 Stop Marker Message

Stop Marker Message 被编码为包含 PASID TLP Prefix 的 Page Request Message，但以下情况除外：

- L, W 和 R 字段分别为 1b, 0b 和 0b。
- Untranslated Address 字段和 PRG Index 字段的高位保留。
- Marker Type 字段为 00000b，以指示这是一个 Stop Marker Message。

PASID TLP Prefix 中的 Execute Requested 和 Privileged Mode Requested 字段为保留值。

- Traffic Class 必须为 0。
- Relaxed Ordering 属性字段必须为 0。
- ID-Based Ordering 属性字段必须为 1。

如果收到 Stop Marker Message 并且以下任何一项为真，则行为未定义：

- Marker Type 字段不等于 00000b。
- 不存在 PASID TLP Prefix。
- PASID 值与 outstanding stop request 不匹配。
- 不完备 PASID 的 Page Request Message 仍处于 outstanding 状态(即,对于某些 PRG 索引,最近接收到的 Page Request Message 没有设置 L 位)。

10.4.2 Page Request Group Response Message

系统硬件和/或软件通过 PRG Response Message 与 Function 的页面请求接口进行通信。主机使用 PRG Response Message 来表示

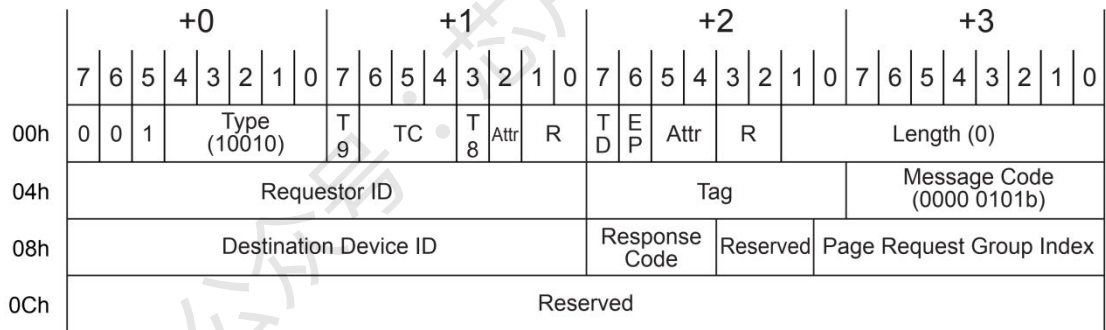
PRG 完成或接口的灾难性故障。为响应 PRG 而发出单个 PRG Response Message，这与 PRG 关联的页面请求数无关。没有机制来指示部分请求完成或部分请求失败。如果不能满足与给定 PRG 关联的任何页面，则认为该请求已失败，并且在 PRG Response Message 中提供失败的原因。主机没有义务部分满足多页请求。如果无法保留其中一个请求的页面，则可以但不必丢弃整个请求。也就是说，从 Function 的角度来看，共享 PRG 与失败的页面请求但与失败无关的页面的驻留时间是不确定的。

有四种可能的页面请求失败：

1. 请求的页面不是有效的 Untranslated Address。
2. 存在 PASID TLP Prefix 支持，页面请求带有 PASID TLP Prefix，并且未为此请求启用 PASID TLP Prefix 用法，PASID 值无效，或者当 R 清除时，Execute Requested 设置为 1。¹⁷¹
3. 请求的页面不具有请求的访问属性（当页面请求带有 PASID TLP Prefix 时，包括执行权限和/或特权模式访问）。
4. 由于未指定的原因，系统无法响应该请求。此响应是终端的（主机可能不再响应任何页面请求，并且在重置 Function 的页面请求接口之前，可能不再提供对该 Function 的任何进一步答复）。例如，违反 Function 分配请求限制或 RC 缓冲能力溢出的请求可能会导致这种类型的故障。

Function 对以上页面请求失败情况 1、2 和 3 的响应取决于实现。失败不一定是持久的，也就是说，失败的请求在某些情况下如果重新发出可能会成功。尽管基于每个 Function，对故障的响应将是与实现相关的行为，但可能性的范围排除了对通用故障行为的精确指定。

所有响应都通过 PRG Response Message 发送到其关联的 Function。一个 Function 必须能够接收多个连续的消息而不会丢失任何信息。为避免死锁，Function 必须能够处理该 Function 的所有 outstanding Page Request Message 的 PRG Response Message，而不必依赖该 Function 发送或接收任何其他 TLP。¹⁷² PRG Response Message 是 ID 路由的 PCIe 消息。该消息中唯一的页面请求接口特定字段是 Response Code 和 PRG。所有其他字段均为标准 PCIe 消息字段。（注意：这些消息是根据字节 8 和 9 中的 ID 路由的；字节 4 和 5 中包含主机的 RID。）



A-0738A

Figure 10-18 PRG Response Message

Table 10-6 PRG Response Message Data Fields

Field	Meaning
Page Request Group Index	Page Request Group Index - This field contains a Function supplied index to which the RC is responding. A given PRG Index will receive exactly one response per instance of PRG (with the possible exception of a Response Failure).
Response Code	Response Code - This field contains the response type of the associated PRG. The encodings are presented in Section 10.4.2.1.

注 171：在 Section 6.20.1 中定义了不存在 PASID TLP Prefix 支持时的行为。

注 172：例如，即使上游 TLP 被流控制延迟，处理导致 Function 发送 TLP 上游的 PRG Response Message 也不得阻塞后续下游 TLP 的处理。

10. 4. 2. 1 Response Code Field

Table 10-7 中列出了 Response Code 字段的值和含义。

Table 10-7 Response Codes		
Value	Status	Meaning
0000b	Success	All pages within the associated PRG were successfully made resident.
0001b	Invalid Request	One or more pages within the associated PRG do not exist or requests access privilege(s) that cannot be granted. Unless the page mapping associated with the Function is altered, re-issuance of the associated request will never result in success.
1110b:0010b	Unused	Unused Response Code values. A Function receiving such a message shall process it as if the message contained a Response Code of Response Failure.
1111b	Response Failure	One or more pages within the associated request group have encountered/caused a catastrophic error. This response disables the Page Request Interface at the Function. Any pending page requests for other PRGs will be satisfied at the convenience of the host. The Function shall ignore any subsequent PRG Response Messages, pending re-enablement of the Page Request Interface.

10. 4. 2. 2 PASID TLP Prefix Usage on PRG Responses

如果页面请求具有 PASID TLP Prefix，则相应的 PRG Response Message 也可以选择包含一个。

如果 PRG Response PASID Required 字段为 0，则 PRG Response Message 没有 PASID TLP Prefix 。

如果 PRG Response PASID Required 字段为 1，则如果页面请求也带有一个 PASID TLP Prefix，则 PRG Response Message 也带有一个 PASID TLP Prefix。允许 Function 使用前缀中的 PASID 值与 PRG 索引结合使用，以匹配请求和响应。

在附加到 PRG Response Message 的 PASID TLP Prefix 中，Execute Requested 和 Privileged Mode Requested 字段为保留值，并从页面请求的 PASID 值复制 PASID 值。

10. 5 ATS Configuration

10. 5. 1 ATS Extended Capability

每个支持 ATS 的 Function（能够生成 Translation Request）必须在其扩展配置空间中实现 ATS Extended Capability structure。允许 Endpoint 或 RC Integrated Endpoint 实现 ATS。

Figure 10-19 详细描述了 ATS Extended Capability structure 中寄存器及各个字段。

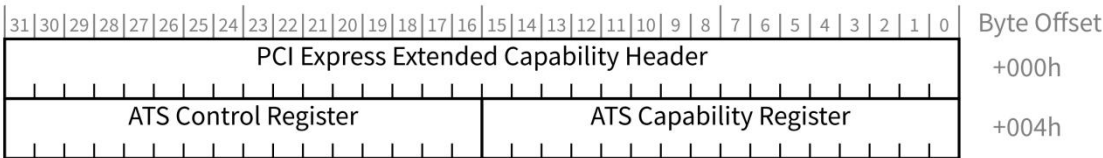


Figure 10-19 ATS Extended Capability Structure

10. 5. 1. 1 ATS Extended Capability Header (Offset 00h)

Figure 10-20 详细说明了 ATS Extended Capability Header 中寄存器字段的分配；Table 10-8 提供了相应的字段定义。

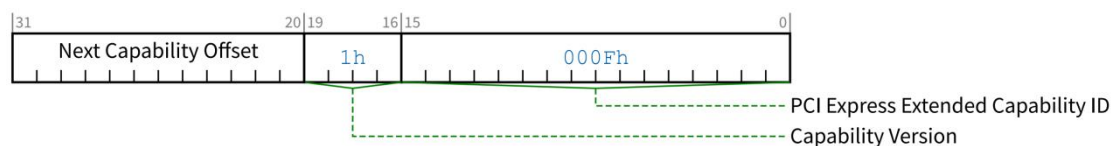


Figure 10-20 ATS Extended Capability Header

Bit Location	Register Description	Attributes
15:0	PCI Express Extended Capability ID - 该字段是 PCI-SIG 定义的 ID 号，值为 000Fh，表示这是一个 ATS Extended Capability。	RO
19:16	Capability Version - 该字段是 PCI-SIG 定义版本号，指示存在的功能结构的版本。 此版本该值必须为 1h。	RO
31:20	Next Capability Offset - 此字段指向下一个 PCI Express Capability structure 的偏移量；如果在 Capability 的链表中不存在其他项，则为 000h。	RO

10.5.1.2 ATS Capability Register (Offset 04h)

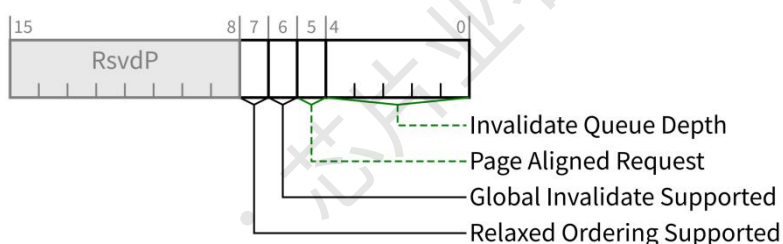


Figure 10-21 ATS Capability Register (Offset 04h)

Bit Location	Register Description	Attributes
4:0	Invalidate Queue Depth - Function 在对上游链路施加反压之前可以接受的 Invalidate Request 数。 如果为 00000b，则该 Function 可以接受 32 个 Invalidate Request。	RO
5	Page Aligned Request - 如果设置为 1，则表示 Untranslated Address 始终与 4096 字节边界对齐。 建议此字段设置为 1。该字段允许软件区分与本规范兼容的实现与与本规范的较早版本兼容的实现，在该版本中，允许请求者以位[11:2]的形式提供任何内容。	RO
6	Global Invalidate Supported - 如果设置为 1，则该 Function 支持把 Invalidate Request 中的 Global Invalidate 字段设置为 1。如果设置为 0，则 Function 将忽略所有 Invalidate Request 中的 Global Invalidate 字段（请参见 Section 10.3.8）。 如果 Function 不支持 PASID TLP Prefix，则该位为 0b。	RO

7	Relaxed Ordering Supported - 如果为 1，则表示当 Enable Relaxed Ordering 字段设置为 1 时，允许该 Function 设置 Translation Request 中的 RO 位。	RO
---	---	----

10. 5. 1. 3 ATS Control Register (Offset 06h)

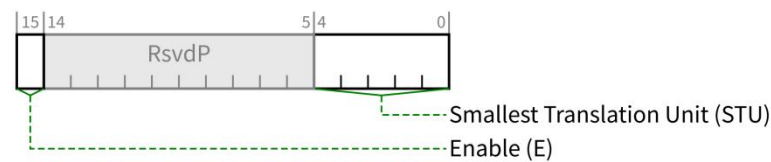


Figure 10-22 ATS Control Register

Bit Location	Register Description	Attributes
4:0	Smallest Translation Unit (STU) - 该值向 Function 指示 Translation Completion 或 Invalidate Request 中指示的最小 4096 字节块数。此值是 2 的幂，块的数量是 2^{STU} 。值 00000b 表示一个块，值 11111b 表示 2^{31} 个块（或总计 8 TB）。 默认值为 00000b。	RW
15	Enable (E) - 置为 1 时，使能 Function 以缓存 translation。 如果设置了该位并且更改了 PASID Enable、Execute Permission Enable 或 Privileged Mode Enable 字段的值，则行为未定义。	RW

10. 5. 2 Page Request Extended Capability Structure

Page Request Extended Capability Structure 用于配置 Page Request Interface 机制。Multi-Function Endpoint 或 RC Integrated Endpoint 设备可以在设备内的任何 Function 上实现 Page Request Interface 和相关 Capability。对于 SR-IOV，PF 支持一个 Page Request Interface，并且在 PF 和关联的 VF 之间共享。PF 实现了此 Capability，而 VF 没有。每个 Page Request Interface 机制都是独立运行的。

注意：对于 SR-IOV，即使 Page Request Interface 在 PF 和 VF 之间共享，它也会在 Page Request Message 的 Requester ID 字段中发送发出请求的 Function 的 ID（PF 或 VF），并期望目标产生 PRG Response Message 的 Destination Device ID 字段中的发出请求的 Function 的 ID。

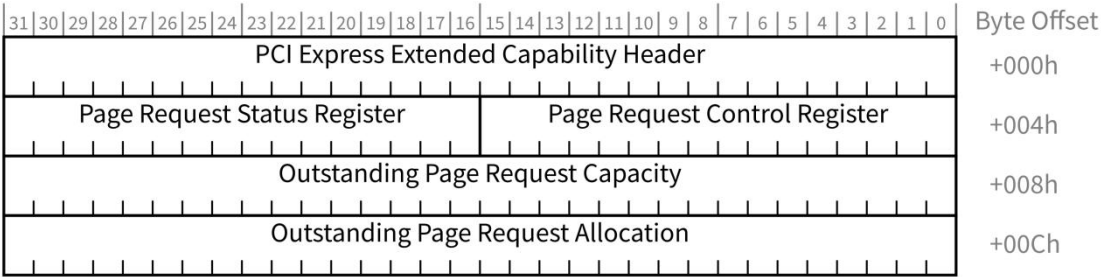


Figure 10-23 Page Request Extended Capability Structure

10. 5. 2. 1 Page Request Extended Capability Header (Offset 00h)

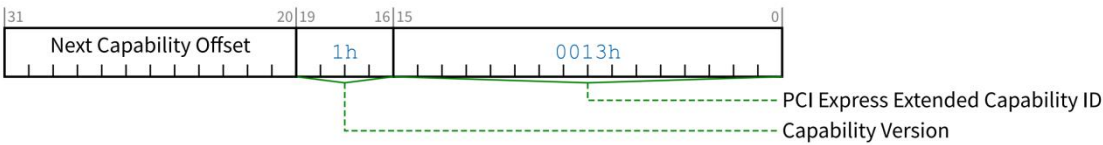


Figure 10-24 Page Request Extended Capability Header

Bit Location	Register Description	Attributes
15:0	PCI Express Extended Capability ID - 该字段是 PCI-SIG 定义的 ID 号，值为 0013h，表示这是一个 Page Request Extended Capability。	RO
19:16	Capability Version - 该字段是 PCI-SIG 定义版本号，指示存在的功能结构的版本。 此版本该值必须为 1h。	RO
31:20	Next Capability Offset - 此字段指向下一个 PCI Express Capability structure 的偏移量；如果在 Capability 的链表中不存在其他项，则为 000h。	RO

10. 5. 2. 2 Page Request Control Register (Offset 04h)

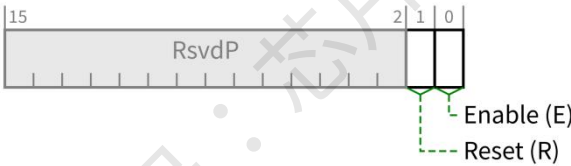


Figure 10-25 Page Request Control Register

Bit Location	Register Description	Attributes
0	Enable (E) - 设置为 1 时，表示允许 Page Request Interface 进行页面请求。如果此字段为 0，则不允许 Page Request Interface 发出页面请求。如果此字段和 Stopped 字段均为 0，则 Page Request Interface 将不会发出新的页面请求，而是具有已传输或排队等待传输的 outstanding page request。当 Page Request Interface 从 not-Enabled 切换到 Enabled 时，其状态标志（Stopped、Response Failure 和 Unexpected Response）将被清 0。使能尚未成功 Stopped 的 Page Request Interface 会产生不确定的结果。 默认值为 0b。	RW
1	Reset (R) - 当清除 Enable 字段或在设置该字段的同一寄存器更新中清除该字段时，向该字段写入 1b，将清除关联的与实现相关的页面请求信用计数器和关联的 Page Request Interface 的未决请求状态。如果将此字段写入 0b 或在设置了 Enable 字段时将该字段写入任何值，则不会启动任何操作。读取该字段将返回 0b。	RW

10. 5. 2. 3 Page Request Status Register (Offset 06h)

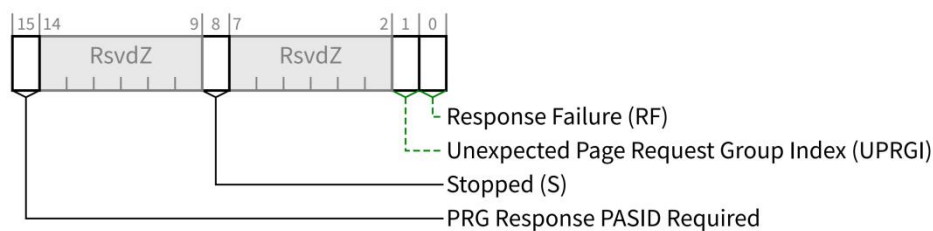


Figure 10-26 Page Request Status Register

Bit Location	Register Description	Attributes
0	Response Failure (RF) - 置为 1 时，此字段表示 Function 已接收到表示响应失败的 PRG Response Message。该 Function 不希望主机进一步响应(任何接收到的信息都将被忽略)。该字段由 Function 设置，当写入 1 时清除。 对于 SR-IOV，如果任何关联的 Function (PF 或 VF) 收到表示响应失败的 PRG Response Message，则在 PF 中设置此字段。 默认值为 0b。	RW1C
1	Unexpected Page Request Group Index (UPRGI)- 置为 1 时，此字段表示 Function 已接收到 PRG Response Message，其中包含没有匹配请求的 PRG 索引。该字段由 Function 设置，当写入 1 时清除。 对于 SR-IOV，如果任何关联的 Function (PF 或 VF) 接收到没有匹配请求的 PRG Response Message，则在 PF 中设置此字段。 默认值为 0b。	RW1C
8	Stopped (S) - 设置为 1 后，相应的 Page Request Interface 将停止发出其他页面请求，并且所有先前发出的页面请求均已完成。如果此字段为 0，则相应的 Page Request Interface 要么没有停止，要么已经停止发出新的页面请求，但是存在 outstanding Page Request。仅当 Enable 为 0 时，此字段才有意义。如果 Enable 设置为 1，则此字段未定义。 如果 Enable 设置为 0，则在先前设置之后，Interface 将转换为停止状态并清除此字段。当前在 Function 上 outstanding 的所有页面请求完成后，将设置此字段，并且 Interface 将进入禁用状态。如果没有 outstanding Page Request，则 Enable 设置为 0 后可以立即设置此字段。重置 Interface 将导致立即切换到禁用状态。在停止状态下，收到响应失败消息将导致立即转换为禁用状态(设置此字段)。 对于 SR-IOV，仅当所有关联的 Function (PF 和 VF) 都停止发出页面请求时，才设置此字段。 默认值为 1b。	RO
15	PRG Response PASID Required - 如果已设置为 1，则当相应的页面请求具有 PASID TLP Prefix 时，Function 需要 PRG Response Message 上的 PASID TLP Prefix。如果清为 0，则该 Function 不希望	RO

任何 PRG Response Message 上带有 PASID TLP Prefix。

如果该位为 0 且 Function 接收到带有 PASID TLP Prefix 的 PRG Response Message，则 Function 行为不确定。

如果此位置 1，并且当相应的页面请求带有 PASID TLP Prefix 时，Function 接收到不带 PASID TLP Prefix 的 PRG Response Message，则 Function 的行为未定义。

如果 Function 不支持 PASID TLP Prefix，则此位为 RsvdZ。

10. 5. 2. 4 Outstanding Page Request Capacity (Offset 08h)

该寄存器包含相应的 Page Request Interface 物理支持的 outstanding page request message 的数量。这是可以有效分配给 Page Request Interface 的页面数的上限。

这个寄存器是 Read Only 的。

10. 5. 2. 5 Outstanding Page Request Allocation (Offset 0Ch)

该寄存器包含允许相应的 Page Request Interface 发出的 outstanding page request message 的数量（在任何给定实例中都为 outstanding）。

Page Request Interface 的 outstanding 的 PRG 的数量小于或等于它已发出的请求消息的数量。例如，如果系统软件将 1000 条消息分配给 Page Request Interface，则单个 PRG 可以使用所有 1000 条可能的请求。相反，在每个 PRG 发出一个请求时，Page Request Interface 在耗尽所有页面请求信用之前将用完 PRG 索引（其中只有 512 个）。Page Request Interface 必须为任何给定的 PRG 预先分配其请求可用性，也就是说，给定 PRG 所需的所有请求必须在所有请求发出之前都可用。

该寄存器为 RW。如果在 Enable 设置为 1 时此寄存器发生变化，则行为不确定。如果写入寄存器的值大于 Outstanding Page Request 的 Capacity，则行为未定义。默认值为 0。

当支持 PASID TLP Prefix 时，请求分配仍与 Function 相对应，并在 Function 以及 Function 的所有 PASID 之间共享。

停止 PASID 不会影响该 PASID 使用的任何分配。系统应该继续响应 PRG Response Message，以便将页面请求和 PRG 索引资源返回给 Function（请参见 Section 10.4.2.1）。

Stop Marker Message 会消耗缓冲区，但不包括在此分配中（请参见 Section 10.4.1.2.1）。系统应为 Stop Marker Message 提供额外的缓冲，并应限制 outstanding Stop Marker Message 的数量，以避免超出此额外的缓冲。

Appendixes

A Isochronous Applications

B Symbol Encoding

Table B-1 给出数据字符的 8b/10b 编码映射。Table B-2 给出标志字符的 8b/10b 编码映射。Table B-1 的表格太多，参考规范获取，这里只给出 Table B-2。

Table B-2: 8b/10b Special Character Symbol Codes

Data Byte Name	Data Byte Value	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
K28.0	1C	000 11100	001111 0100	110000 1011
K28.1	3C	001 11100	001111 1001	110000 0110
K28.2	5C	010 11100	001111 0101	110000 1010
K28.3	7C	011 11100	001111 0011	110000 1100
K28.4	9C	100 11100	001111 0010	110000 1101
K28.5	BC	101 11100	001111 1010	110000 0101
K28.6	DC	110 11100	001111 0110	110000 1001
K28.7	FC	111 11100	001111 1000	110000 0111
K23.7	F7	111 10111	111010 1000	000101 0111
K27.7	FB	111 11011	110110 1000	001001 0111
K29.7	FD	111 11101	101110 1000	010001 0111
K30.7	FE	111 11110	011110 1000	100001 0111

C Physical Layer Appendix

D Request Dependencies

E ID-Based Ordering Usage

E.1 Introduction

IDO (ID-Based Ordering) 是一种机制，它允许放宽某些排序限制，以此作为提高性能的一种手段。通常有些场景是有些 TLP 不能调度到一些 TLP 前面，而开启 IDO 后可能允许这些 TLP 调度到其他 TLP 前面。IDO 允许的通过不是正常操作所必需的（例如，避免死锁）；这只是提高性能的一种手段。

在讨论 IDO 的时候我们引入了“TLP stream”的概念，它是指一组具有相同发起者的 TLP，即请求的 Requester ID 和完成者 Completer ID 是相同的。在通常禁止 TLP 调度到其他 TLP 前面的几种重要情况下，如果 IDO 允许这种调度发生，他们的 TLP 应该属于不同的 TLP stream。

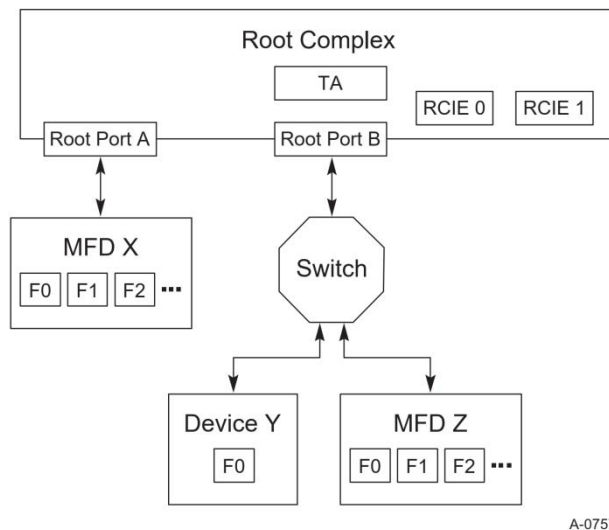


Figure E-1: Reference Topology for IDO Use

Figure E-1 给出了一个参考拓扑。该参考拓扑并不是要阻止 IDO 与其他拓扑一起使用，而是要提供具体的示例进行讨论。

设备 X 和 Z 是 multi-Function 设备（MFD）；设备 Y 是 single-Function 设备。RCIE 是 RC Integrated Endpoint Function，并且可能会或可能不会属于同一设备。我们将假定一个或多个 Function 正在使用 RC（RC）中的 Translation Agent（TA）。

参考 Section 2.4.1 中的排序表和说明，如果两个 TLP 属于不同的 TLP stream，设置了 IDO 字段的 Posted Request、Non-Posted Request 或 Completion TLP 等这些 TLP 允许调度到一个 Posted Request TLP 前面。在以下示例中，DMAR 和 DMAW 代表直接内存访问的读写。PIOR 和 PIOW 代表程序化 I/O 读写。

E.2 Potential Benefits with IDO Use

以下是使用 IDO 可以预见的一些潜在好处的示例。通常，当多个 TLP stream 共享一个公共链路并且该链路由于高利用率或由于暂时缺乏流控制（FC）信用而变得拥塞时，IDO 会提供最大的好处。

E.2.1 Benefits for MFD/RP Direct Connect

以下是 Figure E-1 中 MFD X 和 RC 之间的业务场景下的一些示例。

- Posted Request 可以调度到另一个 Posted Request 前面：当 F0 的 DMAW 由于 TA 丢失而暂停时，如果为 F1 的 DMAW 设置了 IDO，则 RC 内部允许该 DMAW 调度到 F0 的暂停的 DMAW 前面。
- Non-Posted Request 可以调度到另一个 Posted Request 前面：当 F0 的 DMAW 由于 TA 丢失而暂停时，如果为 F1 的 DMAR 设置了 IDO，则 RC 内允许此 DMAR 请求调度到 F0 的暂停的 DMAW 前面。
- Completion 可以调度到另一个 Posted Request 前面：当 F0 中的 DMAW 由于 TA 丢失而暂停时，如果为 F1 中的 PIOR Completion 设置了 IDO，则在 RC 内允许该 PIOR Completion 调度到 F0 的暂停的 DMAW 前面。

E.2.2 Benefits for Switched Environments

以下是 Figure E-1 中 Switch 内部业务场景下的一些示例。

- Non-Posted Request 可以调度到另一个 Posted Request 前面：由于缺少来自 Root Port B 的 FC 信用而导致设备 Y 的 DMAW 在

Switch 内停滞时,如果为来自 MFD Z 的 DMAR 请求设置了 IDO,则允许在 Switch 内把此 DMAR 请求调度到设备 Y 停滞的 DMAW 之前。把来自 MFD Z 中一个 Function 的 DMAR 请求调度到来自 MFD Z 中不同 Function 的停滞 DMAW 之前也是类似场景。

- Completion 可以调度到另一个 Posted Request 前面:由于缺少来自 Root Port B 的 FC 信用而导致设备 Y 的 DMAW 停滞在 Switch 中时,如果为来自 MFD Z 的 PIOR Completion 设置了 IDO,则在 Switch 中允许该 PIOR 调度到设备 Y 停滞的 DMAW 的 Completion 之前。对于来自 MFD Z 中一个 Function 的 PIOR Completion,调度到另一个 Function 的停滞 DMAW 的 Completion 也是类似场景。
- Posted Request 可以调度到另一个 Posted Request 前面:在 Switch 中,把一个 TLP Stream 中的 DMAW 调度到来自不同 TLP stream 的 DMAW 前几乎没有好处。但是,不禁止 Switch 执行这种调度。

E.2.3 Benefits for Integrated Endpoints

这是 Figure E-1 中的 RC Integrated Endpoint (RCIE) 的一些示例。好处与 MFD/RP Direct Connect 情况基本相同。

Posted Request 可以调度到另一个 Posted Request 前面:当 RCIE 0 的 DMAW 由于 TA 丢失而停顿时,如果为 RCIE 1 的 DMAW 设置了 IDO,则 RC 内部允许该 DMAW 调度到 RCIE 0 的停顿的 DMAW 前面。

Non-Posted Request 可以调度到另一个 Posted Request 前面:当 RCIE 0 的 DMAW 由于 TA 丢失而停顿时,如果为 RCIE 1 的 DMAR 设置了 IDO,则 RC 内允许此 DMAR 请求调度到 RCIE 0 的停顿的 DMAW 前面。

Completion 可以调度到另一个 Posted Request 前面:当 RCIE 0 中的 DMAW 由于 TA 丢失而停顿时,如果为 RCIE 1 中的 PIOR Completion 设置了 IDO,则在 RC 内允许该 PIOR Completion 调度到 RCIE 0 的停顿的 DMAW 前面。

E.2.4 IDO Use in Conjunction with RO

IDO 和 RO 是正交的。某些实例:例如,一个 Posted Request 可以调度到另一个 Posted Request 前面的场景可能是在设置 IDO 或 RO 或两者都设置的情况下发生。尽管 IDO 和 RO 在某些情况下有很多重叠,但强烈建议在安全的情况下 IDO 和 RO 两者都使用。RO 允许某些 TLP 在同一 TLP stream 中超前,这是 IDO 不允许的。对于不同 TLP stream 中的业务,IDO 允许控制业务超前于任何其他业务,通常用控制业务设置 RO 是不安全的。

注意,RO 指 Relaxed Ordering Attribute 字段。

E.3 When to Use IDO

对于 Endpoint Function,当 Endpoint 仅与另一个实体(最常见的是 RC)直接通信时,可以安全地在由 Endpoint 发起的所有 TLP 中设置 IDO。对于 RC 情况,“直接通信”具体包括 DMA 通信、PIO 通信和中断通信。与 RCIE 通信或使用 P2P 根端口通信构成与多个实体的通信。

对于根端口,没有可以安全设置的所有 TLP 中的 IDO 的预想的高收益使用模型。使用根端口在 TLP 子集中产生的 IDO 超出本规范范围的模型。

E.4 When Not to Use IDO

E.4.1 When Not to Use IDO with Endpoints

使用 Endpoint Function 时,如果 Endpoint 直接与多个实体进行通信,则在起发送的 TLP 中设置 IDO 并不总是安全的。在某些 TLP

中设置 IDO 可能是安全的，但是此类使用模型不在本规范的范围之内。

例如，在 Figure E-1 中，如果设备 Y 和 MFD Z 正在进行对等通信，并且还通过主机内存进行通信，则在它们发起的 TLP 中设置 IDO 并不总是安全的。作为故障情况的示例，我们假设设备 Y 在向 MFD Z 进行 P2P 写入后执行了 DMAW（到主机内存）。观察到 P2P 写入后，我们假设 MFD Z 然后对先前指定的目标位置进行了 DMAW，正常的排序规则将保证在从 MFD Z 进行 DMAW 之前，主机内存将观察到设备 Y 的 DMAW。但是，如果在 MFD Z 的 DMAW 中设置了 IDO，则允许 RC 具有将第二个 DMAW 调度到第一个前面，导致主机存储器内容的最终结果不同。

当使用 IDO 时，可以使用诸如执行零长度读取之类的同步技术来避免此类通信失败，但是特定的使用模型不在本规范的范围之内。

E.4.2 When Not to Use IDO with Root Ports

对于根端口，如果层次结构中的 Endpoint Function 执行任何 P2P 业务，则在它产生的 TLP 中设置 IDO 并不总是安全的。在某些 TLP 中设置 IDO 可能是安全的，但是此类使用模型不在本规范的范围之内。

例如，在 Figure E-1 中，如果设备 Y 和 MFD Z 正在进行 P2P 通信并且还在与主机软件进行通信，则 Root Port B 在其发起的 TLP 中设置 IDO 并不总是安全的。例如，假设设备 Y 对 MFD Z 进行了 P2P 写操作，然后进行了 DMAW（作为主机内存）。观察到 DMAW 之后，我们假设主机对 MFD Z 进行了 PIOV。正常的排序规则将保证 MFD Z 在从主机进行 PIOV 之前会观察到来自设备 Y 的 P2P 写入。但是，如果从主机在 PIOV 中设置了 IDO，则将允许 Switch 让 PIOV 调度到 P2P 写操作之前，最终使两个写操作不按顺序到达 MFDZ。

Requester and Completer IDs for RC-Originated TLPs

在 PIO 请求中的 Requester ID 与 DMAR 完成中的 Completer ID 不匹配的 RC 实现中，如果在完成中设置了 IDO，则会启用另一种潜在的通信失败情况。对于这种情况，如果在 PIOV 之后是 IDO 集已完成的 DMAR 完成，则根端口下的 Switch 可能允许 DMAR 完成通过 PIOV，这违反了正常的排序规则，即非 RO 读取完成不得通过已发布请求。尽管 PIOV 和 DMAR 完成在逻辑上属于同一 TLP 流，但它们似乎属于不同的 TLP 流。在使用来自此类 RC 的 TLP 设置 IDO 时，建议特别谨慎。

E.5 Software Control of IDO Use

E.5.1 Software Control of Endpoint IDO Use

默认情况下，在 Endpoint 发起的任何 TLP 中均不设置 IDO。

The “Simple” Policy for IDO Use

可以预见，如果 Endpoint 仅直接与另一个实体（例如 RC）直接通信，可能会发现将 IDO 设置为 “Simple” Policy 就足够了。以下是一个设想的 “Simple” Policy。如果将 IDO Request Enable 字段设置为 1，则 Endpoint 将在其发起的所有 TLP 请求中设置 IDO。如果将 IDO Completion Enable 字段设置为 1，则 Endpoint 将在其发起的所有 Completion TLP 中设置 IDO。

可以预见，Endpoint 的驱动程序将确定 Endpoint 何时安全地在其发起的 TLP 中设置 IDO。驱动程序应该能够确定 Endpoint 是否正在与多个其他实体通信，并且应该知道 Endpoint 的能力，包括设置所有 TLP 的 IDO 或者选择性地设置 IDO。如果驱动程序确定启用 IDO 是安全的，则驱动程序可以通过 OS 服务间接设置或直接设置 IDO Request Enable 和/或 IDO Completion Enable 字段，这具体取决于操作系统策略。

如果 Endpoint 不是使用 “Simple” Policy 就足够的通信模型，则 Endpoint 可以实现更复杂的策略来确定 Endpoint 何时将 IDO 设置为 1。这样的实现可能利用由设备驱动程序管理的特定于设备的控件。这样的策略和特定于设备的控制机制不在本规范的范围之内。

E.5.2 Software Control of Root Port IDO Use

因为如果把根端口发起的 TLP 的 IDO 设置为 1 可能并没有后预想的高收益，并且这可能还会导致通信失败，因此根端口很少把他们发送的 TLP 的 IDO 设置为 1。根端口设置 IDO 的使用模型和策略不在本规范的范围之内。

F Message Code Usage

Table F-1 Message Code Usage

Message Code	Routing r[2:0]	Type	Description
0000 0000	011	Msg	Unlock, see Section 2.2.8.4
0000 0001	010	MsgD	Invalidate Request Message, see Section 10.3.1
0000 0010	010	Msg	Invalidate Completion Message, see Section 10.3.2
0000 0100	000	Msg	Page Request Message, see Section 10.4.1
0000 0101	010	Msg	PRG Response Message, see Section 10.4.2
0001 0000	100	Msg	Latency Tolerance Reporting (LTR) Message, see Section 2.2.8.8
0001 0010	100	Msg	Optimized Buffer Flush/Fill (OBFF) Message, see Section 2.2.8.9
0001 0100	100	Msg	PM_Active_State_Nak, see Section 2.2.8.2
0001 1000	000	Msg	PM_PME, see Section 2.2.8.2
0001 1001	011	Msg	PME_Turn_Off, see Section 2.2.8.2
0001 1011	101	Msg	PME_TO_Ack, see Section 2.2.8.2
0010 0000	100	Msg	Assert_INTA, see Section 2.2.8.1
0010 0001	100	Msg	Assert_INTB, see Section 2.2.8.1
0010 0010	100	Msg	Assert_INTC, see Section 2.2.8.1
0010 0011	100	Msg	Assert_INTD, see Section 2.2.8.1
0010 0100	100	Msg	Deassert_INTA, see Section 2.2.8.1
0010 0101	100	Msg	Deassert_INTB, see Section 2.2.8.1
0010 0110	100	Msg	Deassert_INTC, see Section 2.2.8.1
0010 0111	100	Msg	Deassert_INTD, see Section 2.2.8.1
0011 0000	000	Msg	ERR_COR, see Section 2.2.8.3
0011 0001	000	Msg	ERR_NONFATAL, see Section 2.2.8.3
0011 0011	000	Msg	ERR_FATAL, see Section 2.2.8.3
0100 0000	100	Msg	Ignored Message, see Section 2.2.8.7

Message Code	Routing r[2:0]	Type	Description
0100 0001	100	Msg	Ignored Message, see Section 2.2.8.7
0100 0011	100	Msg	Ignored Message, see Section 2.2.8.7
0100 0100	100	Msg	Ignored Message, see Section 2.2.8.7
0100 0101	100	Msg	Ignored Message, see Section 2.2.8.7
0100 0111	100	Msg	Ignored Message, see Section 2.2.8.7
0100 1000	100	Msg	Ignored Message, see Section 2.2.8.7
0101 0000	100	MsgD	Set_Slot_Power_Limit, see Section 2.2.8.5
0101 0010	100	Msg	PTM Request, see Section 2.2.8.10
0101 0011	100	Msg/MsgD	PTM Response/PTM ResponseD, see Section 2.2.8.10
0111 1110	000, 010, 011, or 100	Msg/MsgD	Vendor_Defined Type 0, see Section 2.2.8.6
0111 1111	000, 010, 011, or 100	Msg/MsgD	Vendor_Defined Type 1, see Section 2.2.8.6

G Protocol Multiplexing

协议复用机制提供了一种在 PCI Express 链路上传输非 PCI Express 协议的标准机制。该机制支持将 PMUX 数据包和 TLP 多路复用到单个 PCI Express 链路上。

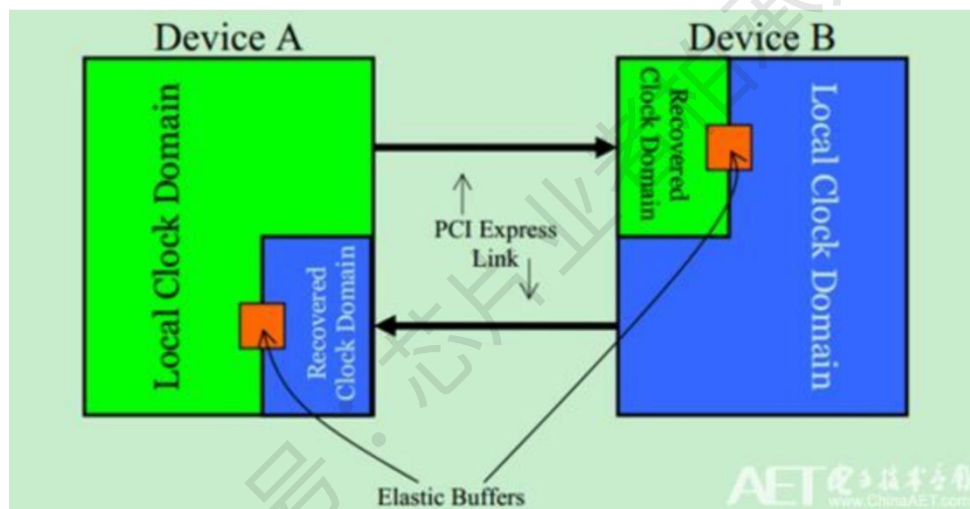
补充知识点

1 物理层的弹性缓冲 (Elastic Buffer)

PCIe 物理层逻辑子层的弹性缓存 (Elastic Buffer, 又称为 CTC Buffer 或者 Synchronization Buffer)。其本质上是一种 FIFO, 主要用于解决跨时钟域问题。当然, PCIe 的弹性缓存还用于补偿时钟误差 (Compensate for the clock differences)。实际上, 除了 PCIe, 弹性缓存还广泛应用于其它的高速串行接口——USB、InfiniBand、Fibre Channel、Gigabit Ethernet 等基于 SerDes 的应用。

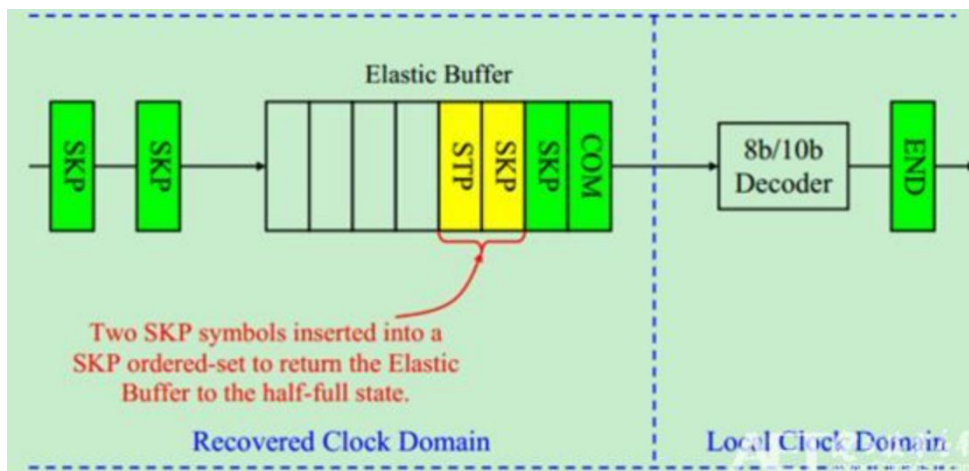
由于 PCIe 采用的基于 8b/10b 的嵌入式源同步时钟, 接收端存在两个时钟域: 一个是通过 CDR 从数据流中解析出来的时钟, 用该时钟对数据进行采样; 另一个是本地时钟域, 用于其他的逻辑的。借助弹性缓存 (FIFO), 可以实现数据在这两个时钟域的转换。

以 PCIe Gen1 为例, 链路上的数据速率为 2.5Gbps。但实际上, 任何晶振 (或者其他频率发生器) 都是有误差的, PCIe Spec 允许的误差范围为 $\pm 300\text{ppm}$ (Parts Per Million)。即链路上实际的频率范围为 $2.49925\text{GHz} \sim 2.50075\text{GHz}$ 。借助弹性缓存, 通过删除或者插入 SKP Ordered Set 可以消除链路频率误差的影响。如下图所示:

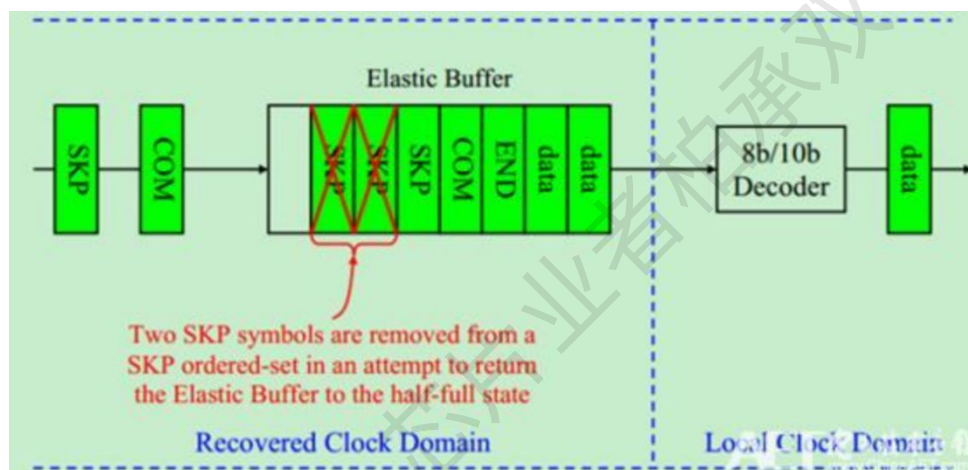


需要注意的是 PCIe Spec 并没有规定弹性缓存的具体位置, 设计者可以将弹性缓存放在 8b/10b 解码器之前, 也可以把弹性缓存放在 8b/10b 解码器之后。不过, Mindshare 的建议是将弹性缓存放置于 8b/10b 解码器之前的。

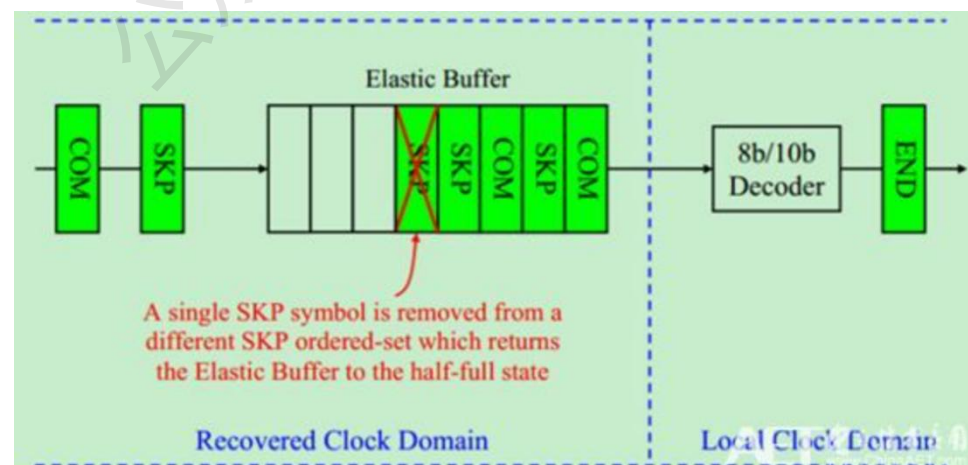
当本地时钟域的时钟 (Local Clock) 的速度比数据流通过 CDR 解析出的时钟 (Recovered Clock) 的时钟要快时, 且弹性缓存即将被读空之前, 可以向 SKP Ordered Set 中插入 1~2 个 SKP。如下图所示:



当本地时钟域的时钟（Local Clock）的速度比数据流通过 CDR 解析出的时钟（Recovered Clock）的时钟要慢时，且弹性缓存即将溢出之前，可以从 SKP Ordered Set 中移除 1~2 个 SKP。如下图所示：



需要特别注意的是，Intel 提出的 PIPE 规范（并非 PCI-SIG 强制的规范，具体参考前面关于 PIPE 的文章）中，只允许每次从一个 SKP Ordered Set 中插入或者移除一个 SKP。如果需要插入或者移除两个 SKP，则需要对两个 SKP Ordered Set 进行操作。如下图所示：



弹性缓冲器其实就是一个 FIFO(First-In-First-Out)先进先出器件。数据以某个时钟频率写入，输出却用另一个不同的时钟频率。因此，依靠这个电气特征，该回路功能方块的最主要目的是要确保高速环境下，传送及接收两端时钟差异上的补偿作用。如此一

来,数据通信的集成性可获最佳保证。

下面以两个耳熟能详的接口为例,来说明弹性缓冲器一般位于控制器中的确切位置。

1. 弹性缓冲器在不同接口中的位置

首先以 USB2.0 Hub 为例。众所周知,Hub 装置的职责就是承上启下,单进多出,以扩展连接设备的数量。USB 接口最多可连接 127 个设备。而 USB2.0 的 Hub 又有别于上一代 USB1.1 规格,它将收进来的封包去头去尾,只读取数据本身,然后以局部新的干净时钟信号将它输出。因此,480Mbps 可连续接 5 阶 Hub 装置依然保持 480Mbps 数据传送畅通。

所以,USB2.0 Hub 内部的重发器(Repeater)区块通常内含弹性缓冲器,可用来补偿 Rcv_Clk 及 Xmt_Clk 两个时钟信号差。

再以 PCI Express 为例。PCI Express 采用内置时钟的 8b/10b 编码方式,因此在接收端势必要构建一个时钟数据回复 CDR(Clock Data Recovery)回路,其中的弹性缓冲器也用来补偿传送时钟 f1 与接收端时钟 f2 的差异。

逻辑上,因为 FIFO 两端的时钟存在差异,最后很有可能发生溢位(Overflow)或下溢(Underflow)现象。为了避免这种情况,EB 弹性缓冲器通常会在特定的时间插入或删除特殊的符号(Symbol)。如此一来,用来补偿两端时钟的差异就不用考虑额外的问题了。这也是 EB 弹性缓冲器名称的由来。

弹性缓冲器的出现甚至可以回溯到 60 年代。1963 年,Maurice Karnaugh 获得这个技术的专利,只不过当时的应用是在 PCM 电话网络上。当高速传输转向串行方式时,EB 弹性缓冲器的价值再度显现。

2. PCIe 中的时钟差

PCI Express 采用的 8b/10b 编码,输出端与接收端时序的同步关系可看成是一种“源同步”,也是一种“时钟传递”的数据传输协议方法。这与过去的 PCI 或 PCI-X 采用的共通时钟方式是截然不同的。

为了方便记忆起见,最为简单的方法就是将“源同步”看成时钟与数据的合成,都来自于输出端的驱动器就可以了。

PCI Express 的传输速率是 2.5Gbps(波特率),容许的误差范围是 $\pm 300\text{ppm}$ 。将它展开数**算后,即可发现该总线输出端允许的频率范围就是 2.49925GHz ~ 2.50075GHz。如果用单元间距 UI(Unit Interval)来计算的话,2.5Gbps 的一个单元间距为 400ps,由于有 $\pm 300\text{ppm}$ 的余裕,因此,最坏的情况下应该是 399.88ps ~ 400.12ps、600ppm 的两端时钟差。换句话说,经过 1666 个周期,可能位移了一个时钟。

既然数据的写入与读取是有差异的,那么势必要追踪缓冲器里的数据存取状态,才能避免错误的发生。

3. EB 在 PCIe 中的应用

PCI Express 为 EB 弹性缓冲器可以在适当的时间插入或删除符号,特地规范了 SKP 指令集(Ordered-sets),由 COM+SKP+SKP+SKP 四个编码后的符号所组成。输出端则必须在最大容许范围(也就是 600ppm 的最坏情况下)输出 SKP 指令集,用来补偿两端时钟差异。

根据 PCI Express 规范,输出端在 1180 ~ 1538 个符号时间内(不是位时间)输出 SKP 指令集。而弹性缓冲器在接收到的 SKP 有序集内,最多可插入或删除两个 SKP 符号以补偿两端的时钟差。至于为什么不是位时间,最主要的原因是 SKP 符号的插入或删除每次都是整个符号在处理,也就是 4ns 的时间。

不难想像,有一种可能会让 SKP 指令集不能在依据规定的时间片内送出。那就是当事务层 TLP 数据包很大的时候,就必须等到整个 TLP 数据包传完之后,才能够立即将累积的待传送 SKP 指令集发送出去。

由于 PCI Express PHY 实体层的构成元素颇多,这里只针对最为重要的三个功能区块与 EB 间的连接关系进行分析。这三个要素即 De-serializer 解串器、8b/10b 解码器及 Deskew 解偏移电路。

4. EB 在 PCIe 中的具体位置

由于 EB 针对 SKP 指令集中 SKP 符号的插入或删除是以符号的层次来处理,所以 EB 肯定在 De-serializer 解串器后面。

其次,EB 放在 8b/10b 解码器之前或之后似乎都很合理。但 PCI Express 有规定一种叫做“回路从属端(Loopback Slave)”的机能,大意是说,PCI Express 设备必须将主控端送过来的 10 位符号原封不动地传送回去,不能有任何更改。但时钟补偿还是必须维持,因此 EB 只

能放在 8b/10b 解码器之前了。

最难理解的是 EB 与 Deskew 解偏移电路的前后关系。PCI Express 总线允许多条传输通道(Lane)的构成,也就是 x1/x2/x4/x8/x16/x32 的构成。PCI Express 允许数据在分割多条传输通道上传输,专用术语即“Byte Striping”。

这时候问题就来了。由于驱动器或各个传输通道跑线长度不同等因素,各个传输通道的接收端有可能提前或落后,并由此产生“Lane-to-lane Skew”时间差。这就需要 Deskew 解偏移电路的对其进行纠正。所以,EB 通常也放在 Deskew 解偏移电路后面。

5. EB 的空间大小

一般而言,EB 缓冲器的大小至少取决于三个因素,它们分别是:传送 SKP 有序集的时间片、设备支持的最大数据承载量(Payload)及链路宽度。尽管规格明文指出:每 1180 ~ 1538 个符号需要送出一个 SKP 指令集。但有时候,TLP 包传到一半正遇上丢 SKP 有序集的时间点,就必须耐心等待 TLP 包完全传送出去。这就是为什么一定要郑重考虑最多数据承载量的主要原因。

最坏的情况是:当两端 PCI Express 设备以 x1 的链路状态存在时,输出端输出最大承载量的包后刚好也是必须输出 SKP 指令集的时候,可以用图 8 所示的数学式表达。

图 8 所示的数学式中的 TLP_Overhead 是可以定性定量算出最高值的,总共 28 个符号值,罗列如下:

- ◆ 数据框的开始符号(1 个符号)
- ◆ 封包次序号码(2 个符号)
- ◆ 封包标头(16 个符号)
- ◆ ECRC(4 个符号)
- ◆ LCRC(4 个符号)
- ◆ 数据框的退出符号(1 个符号)

由于 PCI Express 允许的数据最大承载量是 4096 字节。根据上面数学式就可算出,最多的符号位移是 3.4 个;若承载量是 2048 字节,符号位移是 2.17 个(见表 1)。

6. EB 深度与数据承载量有关

EB 缓冲器实践的方法一般有两种。第一种是把 EB 维持在半满状态。

正常的情况下,EB 缓冲器维持在半满状态。SKP 指令集中的 SKP 符号可以插入或移除来补偿两端的时钟差。当一个设备能够满足 4096 字节的承载量时,其 EB 缓冲器深度应该为 8。

举例来说,当局部端的时钟比较慢,也就是读取的速度比不上符号写入的速度,为了防止溢位,就可在接收到的 SKP 有序集中移走 2 个 SKP 符号,尽可能让 EB 维持在半满状态。

另外一种实践策略是流程控制手段。这种方式并不把 EB 缓冲器维持在半满状态,而只维持在一个底端进入点的状态。

局部端时钟比较快的情况下是允许下溢的。但当局部端时钟比较慢时,处理方法就与前面一样,在 SKP 指令集中移走 SKP 符号,尽可能让 EB 维持在正常状态。

上述两种方式各有优点。比如说,流程控制的手段可降低缓冲器深度,等待时间(Latency)也比较短。在 4096 位承载量时,EB 缓冲器的深度为 5,因为必须加上一个底端的进入点。但这种情况的实现线路比较复杂。

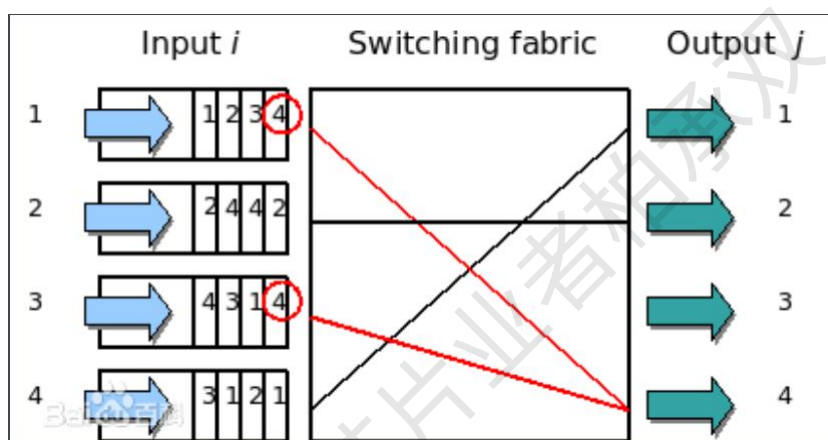
综上所述,EB 的深度与数据承载量息息相关,而 EB 宽度取决于编码方式。

2 关于 head-of-line blocking

术语 HOL，有行头阻塞、队头阻塞、派头阻塞等名称。由于 FIFO（先进先出）队列机制造成的，每个输入端的 FIFO 首先处理的是在队列中最靠前的数据，而这时队列后面的数据对应的出口缓存可能已经空闲，但因为得不到处理而只能等待，这样既浪费了带宽又降低了系统性能。

当在相同的输入端口上到达的包被指向不同的输出端口的时候就会出现排头阻塞。由于输入缓存使用了 FIFO 队列，交换结构在每一个时隙（周期）中将输入队列队头的数据包发送到其对应的输出端口中，队列后部的数据包就只能等待前面传输完后再排到队列前面，然后才可以传输。如果某一缓存头部的数据包由于拥塞而不能交换到一个输出端口，那么后面的数据包就会被它阻塞，该缓存中余下的包也会被队头包所阻塞，即使这些数据包本身目的端口并没有拥塞。

如下图所示：当前 1、3 输入队列是相互竞争的，即它们相同的输出端口口 4 发送数据包。在这种情况下，如果交换结构决定从输入队列 3 中传输数据包，则在同一时钟周期内不能处理输入队列 1 的数据包。而且处于输出队列 1 的后续数据包，如第二个数据包（输出端口为 3，当前空闲）也将无法被输出到端口 3。

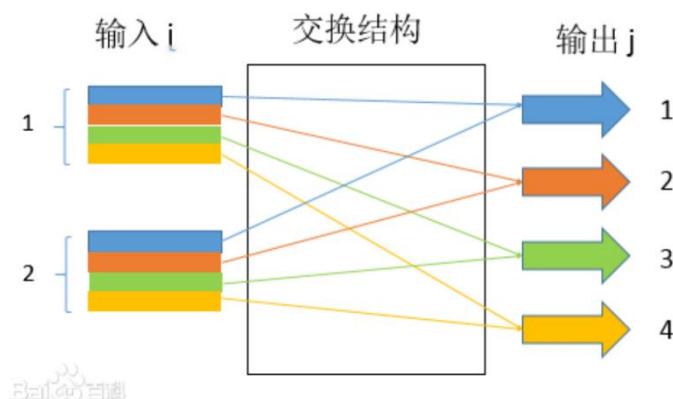


克服这种局限性的一种方法是使用虚拟输出队列（Virtual Output Queues）。因为只有具有输入缓冲的交换结构才存在排头阻塞问题。所以若有充足的内部带宽，则输入缓冲就是不必要的，进而能够避免排头阻塞的问题。这种没有输入缓冲的交换结构在小到中等规模的以太网交换机上十分常见。

虚拟输出队列（Virtual Output Queues）总体的想法十分朴素：在输入端口将发送到不同端口的数据包虚拟成不同的队列，并且彼此互不影响，这样一来即使队头数据包被阻塞也将不会影响发送到其他端口的数据包的发送。

如下图 2×4 的输入输出交换结构中，每个输入端口将根据数据包输出端口的不同而加入专属“虚拟队列”（图中以不同的颜色区分），这样一来，在同一输入端口而目的端口不同的数据包的发送将彼此互不影响。

除了虚拟输出队列（Virtual Output Queues）外，还有其他许多解决排头阻塞的算法，如神经网络、iSLIP 等等。



3 关于 de-embedding 技术

RF 工程师们依赖于矢量网络分析仪 (VNA) 来测量 RF 元件的 S 参数，从而进行描述以及后续的设计。测量中会出现一个问题，即这些元件往往是表贴式，因此不能与 VNA 直接相连。简单的印刷版 (PCB) 测试装置往往会表贴上待测器件(DUT)以便与 VNA 相连接，如图 1 所示。然而，这些测试装置本身会为 S 参数测量带来一些寄生效应，因此，需要 De-embedding 技术进行消除。



图 1

3.1 De-embedding 技术定义

微波元器件的测量必须包含两部分：第一部分是把实际被测器件 (DUT) 加上焊点 (对于在片测量系统) 或者测试夹具 (对于同轴测量系统) 当做一个被测器件；另一部分是空的焊点或者测试夹具作为被测器件。然后用第一部分测得的 S 参数扣除第二部分 (焊点或者测试夹具部分产生的寄生效应)，从而可以得到被测器件的真实性能。我们把这个步骤就叫做 De-embedding。

De-embedding 技术是微波测量的重要技术之一，主要目的是消除寄生元件、部件对实际被测器件 (DUT) 的影响。一般分为四大类：串联技术、并联技术、级联技术及混合技术。为了确保测量值为实际被测器件的本身特性，必须进行 De-embedding。

3.2 De-embedding 技术理论简介

由上面对 De-embedding 技术的定义我们知道，我们测量的 S 参数实际是 DUT、寄生元件（如探针）、其他元件（如焊点）综合下的 S 参数，因此，我们需要对这些因素进行消除，以达到精确测量的目的。

对于任何一个被测器件 DUT，我们都会使用到探针，而探针跟 DUT 是级联关系，我们采用 A 参数处理；对于焊点来说，有短路焊点与开路焊点，开路焊点相当于在 DUT 并联，短路焊点相当于串联器件，对于并联与串联，我们分别使用导纳参数与阻抗参数进行运算。

在对一个器件进行 De-embedding 精确测量时，我们一般采取由内而外消除各个影响，具体步骤如下：

- 1) 测量包括被测器件、开路、短路焊点及微波探针在内的 S 参数，将 S 参数转变为 A 参数 $[A_{total}]$ ，利用级联技术消除微波探针影响，得到被测器件及开路、短路焊点在内的 S 参数；
- 2) 将 S 参数转换为 Y 参数 $[Y_{total}]$ ，利用并联技术，消除开路焊点的影响，得到被测器件和短路互连线在内的 Y 参数 $[Y_T]$ ；
- 3) 将 Y 参数转化为 Z 参数 $[Z_T]$ ，利用串联技术，消除短路焊点的影响，得到被测器件的 Z 参数 $[Z_{DUT}]$ ；
- 4) 将得到的被测器件的 Z 参数转化为 S 参数，即可得到被测元件的 S 参数。对于级联技术的实现，可以由如下表示：

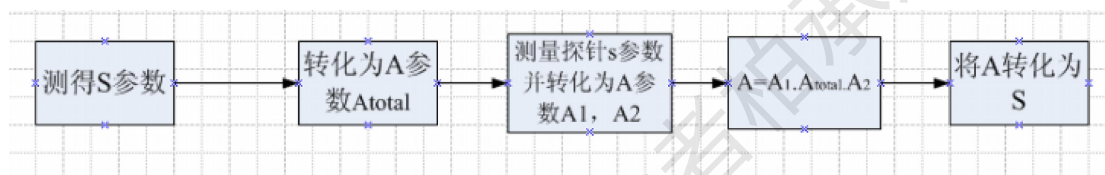


图 2 级联示意图

对于并联技术如图 3 所示：

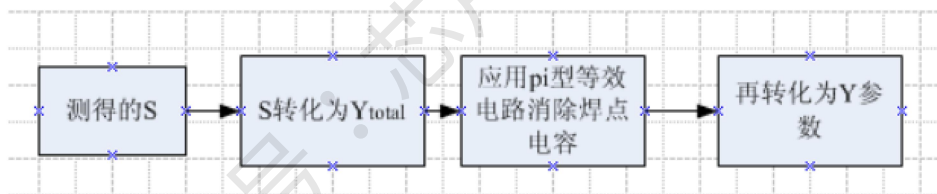


图 3 并联技术过程图

串联技术如图 4：

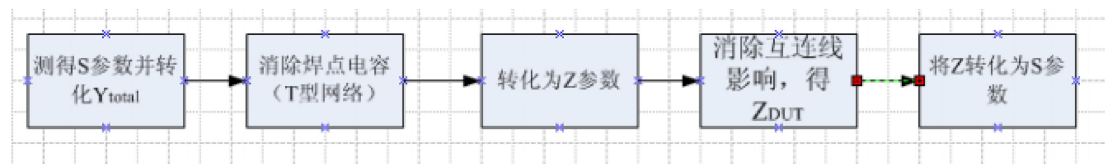


图 4 串联技术过程图

3.3 应用：UHF 频段 RFID 近场天线的阻抗测量方法

超高频（UHF）频段的射频识别（RFID）近场读写器天线（NFRA）由于其单品识别方面应用的潜力，对环境的不敏感性和比 HF 天线更高的读写速度，正引起多方面的关注。UHF 频段的 NFRA 通常采用带有平衡端口的电大环结构来实现。

对于 NFRA 来说，良好的匹配网络是至关重要的。通常 UHF 频段的 NFRA 天线都被设计成安装在金属腔体里来减小环境对天线性能的影响，如图 5 所示。但是由于金属腔体的存在，天线的阻抗会随频率的变化而剧烈变化，这将导致在仿真软件中得到的阻抗值不够精确，在此不精确的阻抗基础上很难设计出性能良好的匹配网络。

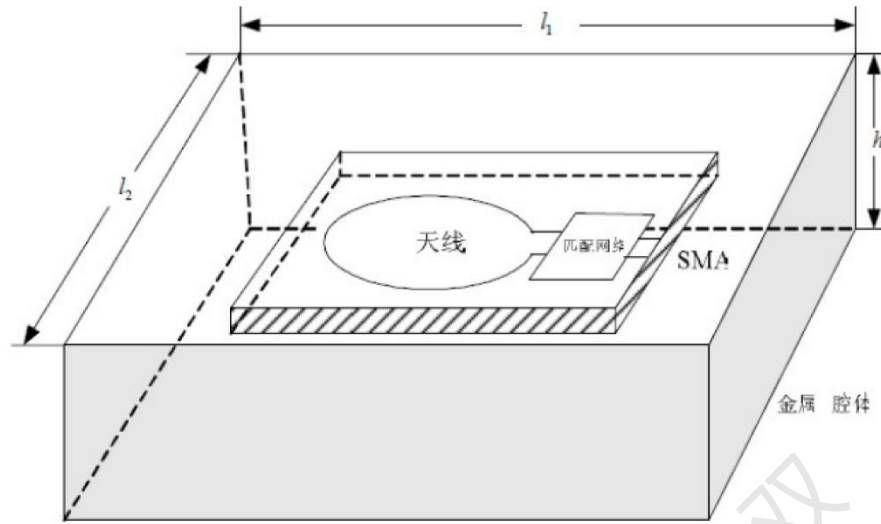


图 5 UHF NFRA 近场读写天线结构图

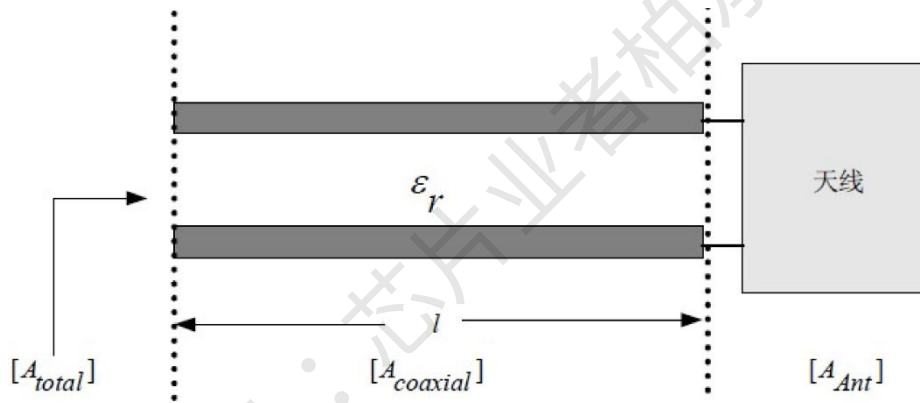


图 6 测量方法的等效电路模型

图 6 给出了使用 De-embedding 技术测量的等效电路模型，其中，同轴线被一段长为 l 的传输线等效。

应用前面说的步骤：

- 1) 测量天线和寄生部件的 S_m 参数；
- 2) 将 S_m 转化为 A 参数， A_{total} ；
- 3) 在图 6 我们可以看到，待测线跟一段同轴线串联，我们可以得出： $A_{total} = A_{coaxial} * A_{ant}$ ；
- 4) 由矩阵变换可以得到， $A_{ant} = A_{coaxial}^{-1} * A_{total}$ ；

$$A_{coaxial} = \begin{bmatrix} \cos \beta l & jZ_0 \sin \beta l \\ \frac{j \sin \beta l}{Z_0} & \cos \beta l \end{bmatrix}$$

- 5) 最后将 A_{ant} 转化为 S 参数：

$$S_{Ant} = S_m e^{j2\beta L} = S_m e^{j\frac{4\pi f \sqrt{\epsilon_r} L}{c}} = (mag S_m) e^{j(\theta + \varphi)}$$

其中， β 是相位常数， ϵ_r 为同轴线的相对介电常数， θ 为 S_m 的相位， mag 为其幅度。

S_{Ant} 是一个复数，我们令 $S_{Ant} = Re + jIm$

再由传输线的终端反射系数公式有 $S_{Ant} = \frac{Z_m - Z_0}{Z_m + Z_0}$

由上述两式，我们可以得出天线阻抗的虚部与实部：

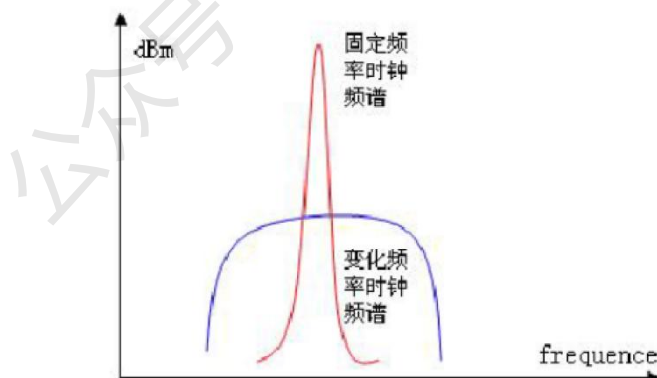
$$re(Z_m) = \frac{Z_0 \cdot (1 - Re^2 - Im^2)}{(1 - Re)^2 + Im^2}$$

$$im(Z_m) = \frac{2Z_0 \cdot Im}{(1 - Re)^2 + Im^2}$$

Z_0 为同轴线特殊阻抗。

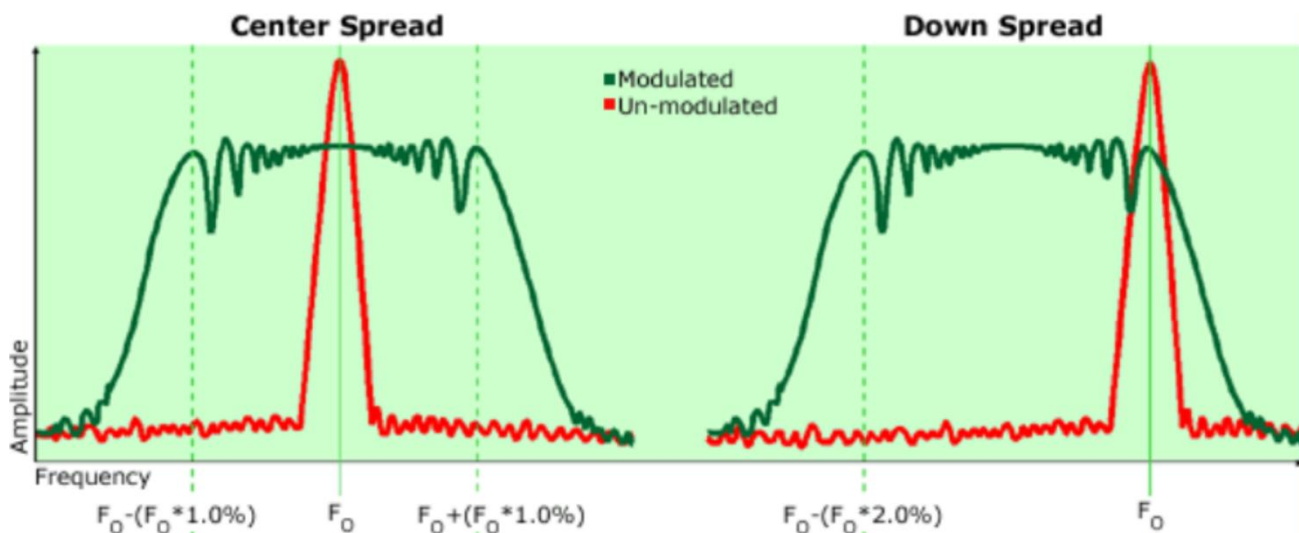
4 扩频时钟 (SSC, Spread Spectrum Clocking)

EMI 一直是高速系统设计的难点，在传统设计中，主要通过滤波、接地、屏蔽等方法来减小 EMI 辐射，这些方法都是通过改变/切断 EMI 辐射路径来达到减小 EMI 辐射的目的，往往设计成本比较高，另外还有一种更好的治本方法，那就是在 EMI 源头上做文章，减小 EMI 的产生，SSC 技术就是其中一种。学过信号与系统课程的同学都知道，对于固定频率的时钟，所有能量都集中在其基频上，其频谱很窄，但幅度很高，对外辐射能量很大，而对于频率变化的时钟，其能量会分散在一定频率范围上。当前 PCIe、SATA、SAS、USB3.0 等几乎所有的高速协议都支持 SSC 的功能。



扩频时钟类似于扩频通信，也是用一个较低的频率调制系统时钟，使得窄带的周期性系统时钟被有意扩展为宽带，基频和谐波所包含的峰值能量显著降低，在频域上的表现是产生一个具有边带谐波的频谱。扩频时钟一般有如下参数：扩展率、扩频类型、调制率和调制波形。

扩频类型指向下扩频、中心扩频或向上扩频，由于中心和向上扩频都会产生超过系统时钟的频率，会对系统造成影响，所以一般向下扩频用的最为广泛。

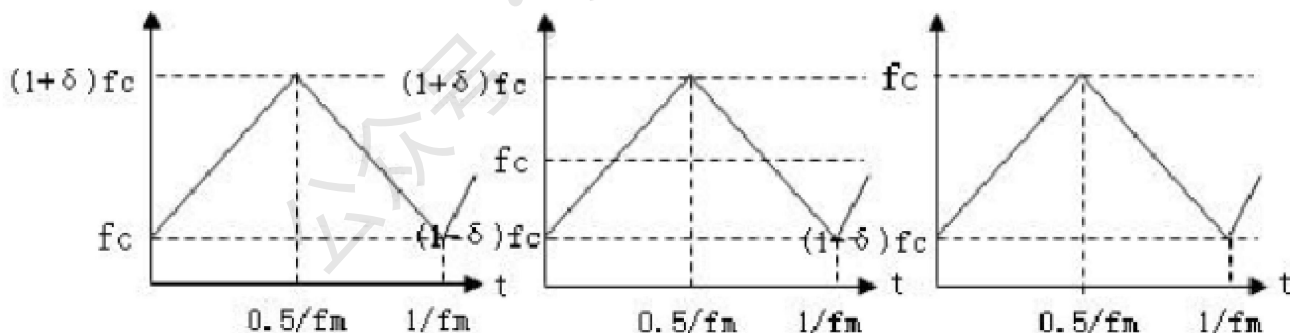


扩展率是频率抖动（或扩展）范围与原 CLK 频率（ f_c ）的比值。虽然高扩展率加强了对 EMI 的衰减程度，但是也可能会超过系统最大额定频率或低于平均频率而影响系统性能，一般扩展率在 0.5%~2.5% 之间。

调制率 f_m 用于确定 CLK 频率扩展周期率，在该周期内 CLK 频率变化 Δf 并返回到初始频率。一般来说频谱调制速率较低，可以实现频率平滑调制，减少调制后的时钟周期的抖动参数。因此当系统对时钟周期抖动参数较敏感时，降低调制速率是一种有效的设计手段。虽然通过减小频谱调制速率可以降低调制时钟源的时钟周期抖动，但是仍然会引入一定的附加的时钟周期抖动，在有些电路设计中会最终导致系统出错。

调制率实际上就是调制波形的频率。通常扩频时钟典型的调制频率范围在 30 到 60kHz 之间。由于像 PCIe 这样的告诉串行总线，1.0 的速率达到了 2.5Gbps，2.0 更是高达 5Gbps，所以 PCI SIG 规定了扩展率为 0~0.5%，调制频率为 30~33kHz，这样不但满足了 EMI 的衰减要求，也兼顾了将扩频时钟带来的周期抖动最小化的要求。

调制波形代表 CLK 频率随时间的变化曲线，通常为三角波和非线性的 Hershey kiss 波。相比三角波，hershey kiss 能提供最佳的平坦度和最大的衰减幅度。



5 PCIe 演进方向之 CCIX 简介

摩尔定律逐渐降速，业界需要一同寻找提升计算性能、同时保持低功耗的方法。CCIX 联盟的成立旨在实现一种新型互联，专注于新兴的加速应用，如机器学习、网络处理、存储卸载、内存数据库和 4G/5G 无线技术。这个标准使得基于不同指令集的处理器的，将缓存一致性、对等计算的优势扩展至许多加速设备包括 FPGA、GPU、网络或存储适配器、智能网络和定制的专用集成电路。CCIX 通过扩展现有成熟的数据中心硬件和软件基础设施来简化开发和采用。这最终能使系统设计者将合适的异构组件无缝集成来满足特定系统需求。

缓存一致性的加速器互联，即 CCIX™（读成“see 6”）是一种能够将两个或两个以上器件通过缓存一致性的方式来共享数据的片间

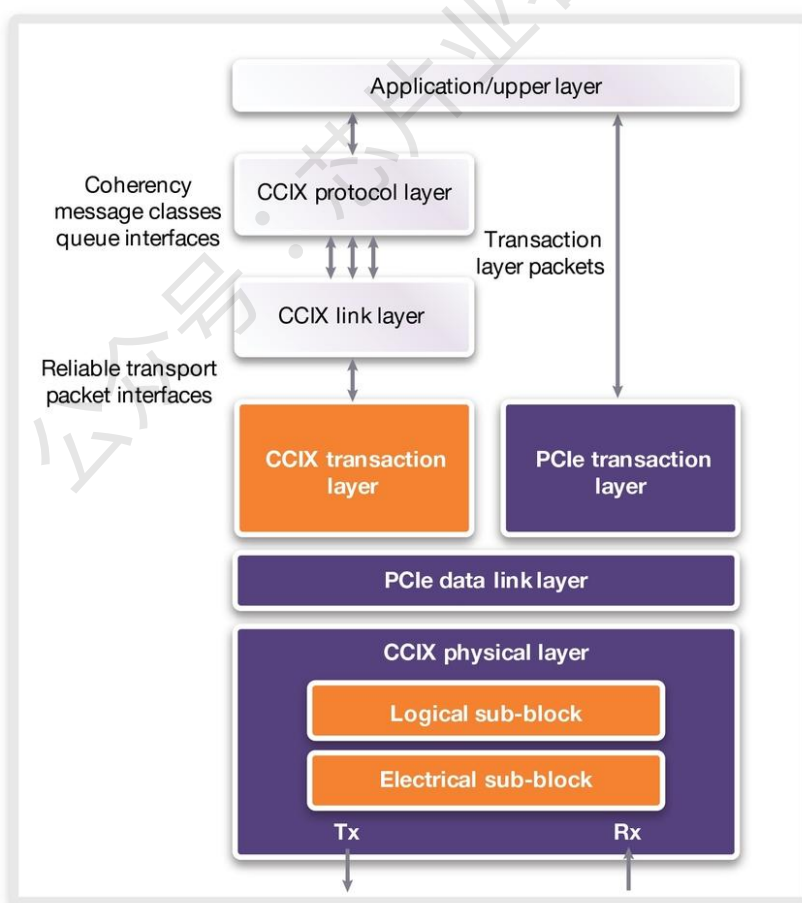
互联。机器学习和大数据应用正深刻的变革数据处理的方式。通过片外加速器的定制，传统处理器从计算到网络的应用都得到了增强；这推动了产业整体向加速器和异构计算发展。对目前很多计算任务，加速器能够比单独的处理器速度更快、功耗更低的完成所需功能。但是，不受管控的异构会带来软件复杂性。CCIX 旨在优化、简化异构系统的架构设计，同时基于不同指令集(ISA)的处理器或应用特定的加速器提升系统的带宽、降低时延。

今天，高科技的版图是由新的、大规模的消费者服务的创新构成的，例如 5G、云计算、物联网、大数据和自动驾驶。机器学习和人工智能应用根本性的改变了消费者行为。这又推进了平台和解决方案不断演进，通过高效、可扩展的方式来支持这些新应用。仅以中央处理器为中心的服务器架构的解决方案无法满足这些应用的性能需求。因此需要基于高效的异构计算架构的解决方案，包含加速器例如图形处理单元（GPU）、可编程逻辑阵列（FPGA）、智能网卡（NIC）和很多其它领域特定的可编程器件。PCI Express (PCIe) 是目前最常见的，处理器和片外加速器间传输数据的协议。尽管 PCIe 协议作为输入输出(IO)协议很有效，但不能支持 IO 设备成为对等计算模型中的一个无缝组件。

随着片外加速器的应用越来越多，**高性能、低延时和易用性**成为下一代互联的首要诉求。

5.1 CCIX 分层架构

CCIX 规范的最大优势之一是它建立在 PCI Express 规范的基础之上。CCIX 的一致性协议只需很少修改或者无需修改就可以通过 PCI Express 链路传递。如下图所示，某个现有的 PCI Express 控制器实施可以通过逻辑进行扩展，以实现 CCIX 事务层。CCIX 事务层负责携带一致性消息，而块 - 即 CCIX 协议层和链路层 -- 负责实现一致性协议本身并对其执行操作。这些块需要与用于缓存的内部片上系统（SoC）逻辑进行紧密集成，并且可能是特定于该 SoC 上使用的具体架构。在其今后设计中实现 CCIX 的 SoC 设计人员通常希望将 CCIX 协议及链路层与 CCIX 事务层分开，以使前者能够与内部 SoC 逻辑紧密集成。



CCIX 栈的最上层是 CCIX 协议层。这一层负责一致性协议，包括内存的读、写流。这一层提供了片上一致性协议（例如 Arm AMBA

CHI) 的简单映射。这一层定义的缓存状态使得硬件能够确定内存的状态。比如硬件可以确定数据是否唯一且未被修改（和内存一致），或是共享且被修改的（和内存不一致）。

CCIX 协议层之下是 CCIX 链接层。这一层负责 CCIX 协议层定义的代理（agent）之间消息的传输格式。目前 CCIX 链接层是构建在 PCIe 之上，但是基于分层架构，CCIX 将来可以映射到不同的传输层。此外，这一层负责端口聚合，使得多个端口能够聚合在一起提升带宽。

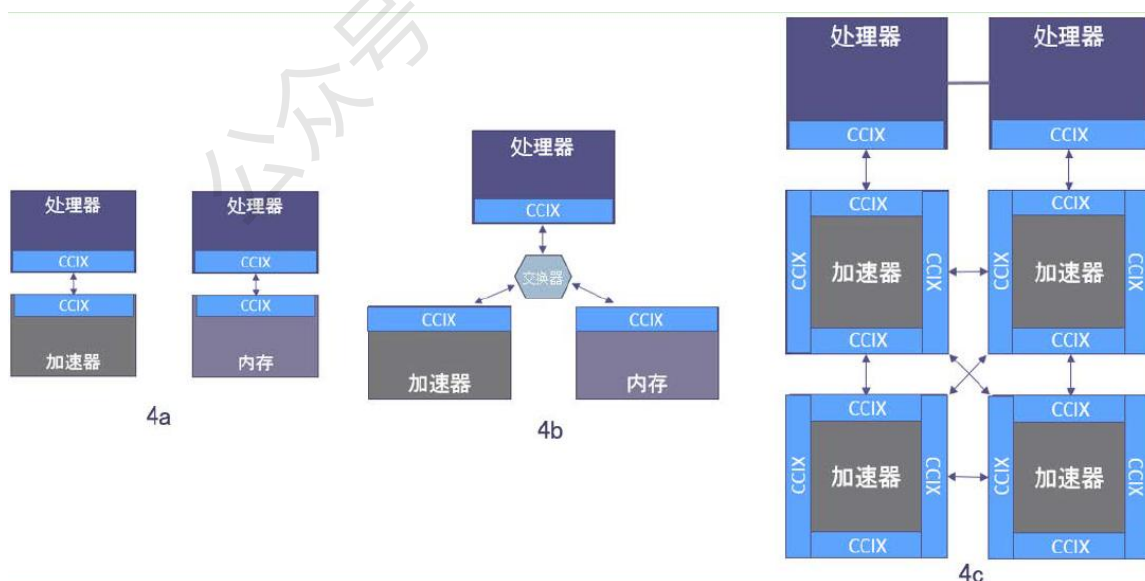
CCIX 和 PCIe 事务层负责处理它们各自的包。PCIe 协议支持部署虚拟通道，使得不同数据流通过一个 PCIe 链路。将 CCIX 和 PCIe 传输流各分到一个虚拟通道，CCIX 和 PCIe 传输可以共享相同的链路。CCIX 能够传输标准的 PCIe 包，或经过优化的 CCIX 包。经过优化的 CCIX 包删减了 PCIe 包里的几个不必要的字段。传输标准的 PCIe 包时可以采用现有的 PCIe 交换器。传输经过优化的 CCIX 包，能降低 PCIe 的额外开销，使得一致性传输的包更小、更高效。

PCIe 数据链路层执行数据链路层的所有正常功能。这些功能包括 CRC 错误校验、包确认和超时检查，和信用初始化及交换。

如前所述，CCIX 的最大的吸引力之一就是它与 PCI Express 的兼容性。实际上，CCIX 的缓存一致性协议可以通过运行 8GT/s 或更快速度的任何 PCI Express 链路来传递。PCI Express 4.0 规定的最高数据速率为 16GT/s，这在一条 16 通道链路上可以达到总双向带宽约 64GB/s，但 CCIX 联盟的一些成员需要更大的带宽。他们认为，通过将传输速率提高到 25GT/s，一条 CCIX 链路可以在相同的条件下达到 100GB/s。这导致出现了一项称为“扩展速度模式”（ESM）的 CCIX 特性。由于 PCI Express 由一个不同的标准化机构所拥有，所以 CCIX 联盟选择了一个聪明的机制用于在具有 ESM 功能的组件与 PCI Express 组件之间实现兼容性。希望彼此进行通信的两个 CCIX 组件可以通过正常的 PCI Express 链路初始化过程（通常是一个硬件自主过程）进行处理，以达到最高的相互支持的 PCI Express 速度。自此开始，在主机系统上运行的软件可以询问 CCIX 特定的配置寄存器，并确定两个组件是否都具备 ESM 能力。如果具备的话，则确定它们的最高支持速度。该软件然后在两个组件上编写其他 CCIX 特定的寄存器，以便把 PCI Express 链路速度映射为 CCIX ESM 链路速度。自此以后，链路协商将针对 CCIX ESM 速度，因此，通过强制进行链路重新训练，这两个组件现在可以以高达 25GT/s 的速度进行通信。

5.2 CCIX 系统拓扑样例

得益于分层架构，CCIX 能够支持多种灵活的拓扑结构。最常见的拓扑结构是直接附加的共享虚拟内存。但其它拓扑结构，如交换器、菊花链或网状拓扑，也很容易被构建和支持。

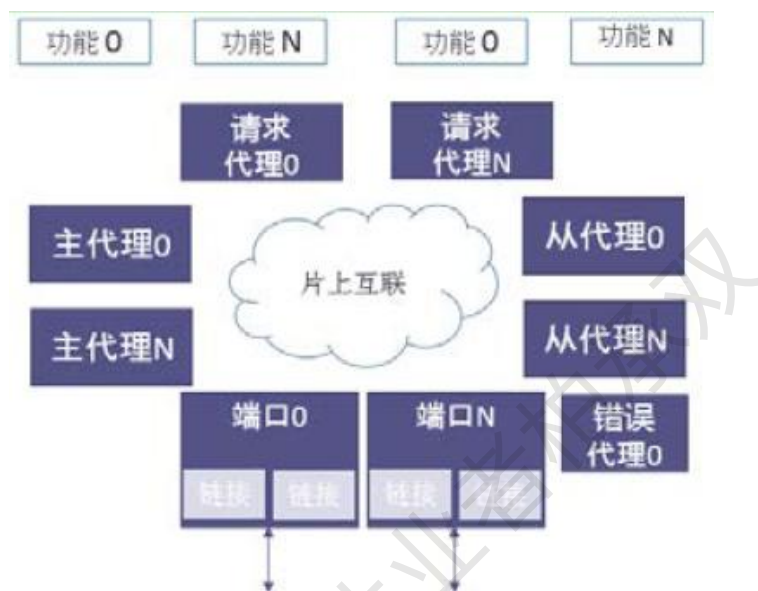


上图中，4a 为直接连接，4b 为交换器拓扑，4c 为混合菊花链。

5.3 CCIX 一致性分层架构

CCIX 协议定义了 CCIX 组成模块的内存访问协议。所有 CCIX 器件至少有一个具备 CCIX 链接的端口。一个 CCIX 端口关联一组物理管脚，用于和另一个 CCIX 端口连接，在两个或多个不同芯片间交互信息。

同时定义了一些不同的代理类型，哪种代理在哪个器件取决于器件的功能。定义的代理类型包括：请求代理（RA）、主代理（HA）、从代理（SA）和错误代理（Error Agent）。请求代理、主代理、从代理、错误代理，系统里的端口和链接统称 CCIX 组件。一个代理在协议中由一个代理 ID 标识。以下对每一种代理类型进行简要描述。



请求代理（RA） - 一个请求代理对系统内的不同地址进行读、写操作。请求代理可以对它已经访问的地址的数据进行缓存。每个 CCIX 请求代理可以有一个或多个处理单元作为内部请求的发起者，它（们）的请求由一个 CCIX 架构的请求代理执行。根本上说 CCIX 请求代理提供了加速器或 CCIX 使能的 IO 主设备向一致性系统内存的接口。此外，请求代理使得加速器的缓存具备一致性，因此编程者无需感知。

主代理（HA） - 主代理负责管理指定的一段地址的数据一致性。当一个缓存行的状态需要改变时，主代理通过向所需的请求代理发出侦听操作来保持一致性。

从代理（SA） - CCIX 支持扩展系统内存，来包含外设所附的内存。这种情形出现在主代理在一个芯片上，而这个主代理关联的一些或全部物理内存存在另一个芯片上时。这种架构组件（扩展内存）称为从代理。从代理不会被请求代理直接访问。请求代理总是访问一个主代理，然后主代理再访问从代理。

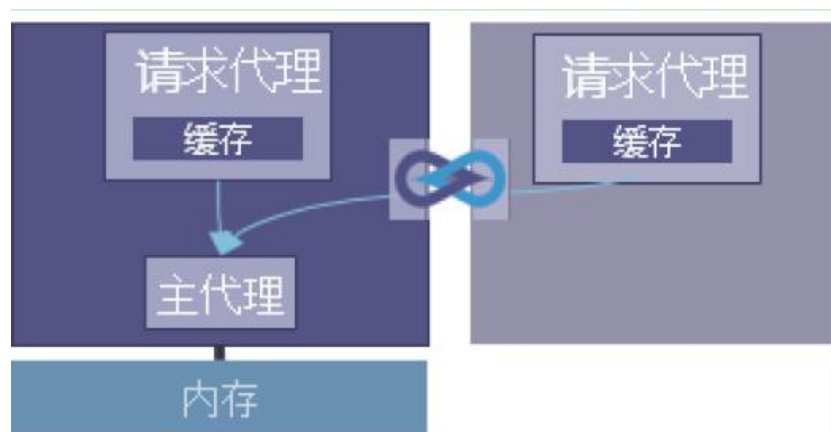
错误代理 - 一个错误代理接收并处理协议错误信息。协议错误信息由 CCIX 组件发出。

5.4 CCIX 数据流样例

基于上述的代理类别，可以描述 CCIX 可见的一些常见用例。

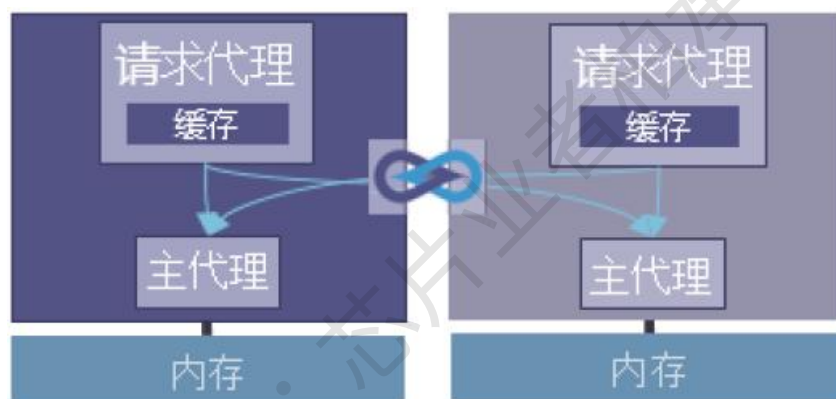
1、加速器共享处理器内存

当采用和部署 CCIX 时，最常见的初始用例是处理器和加速器共享缓存。这个用例里有两个请求代理，各自管理自己的缓存。主代理在处理器上，管理连接到该处理器的内存的访问。



2、共享的处理器和加速器内存

下一种常用模型可能是处理器和加速器共享虚拟内存。在这个用例里，加速器和处理器的内存同在一个共享虚拟内存池里。处理器只需要简单的将待处理的数据的地址指针传给加速器，而不需要复杂的 PCIe DMA 和驱动在处理器和加速器内存之间传递数据。有两个请求代理管理各自的缓存，有两个主代理管理内存。免去软件驱动开发和额外开销，可以大幅提升系统性能和简化软件。



3、拓展基本结构

得益于 CCIX 非常灵活的特性，它可以在展示的这些基本数据流之外进行拓展。从直接附带的加速器、到网状拓扑和星型网络，CCIX 具备很可观的选项来支持很多种类的拓扑结构。

5.5 CCIX 软件

CCIX 对硬件的增强极大推进了片外加速器，同时 CCIX 的一个关键优势是它能支持主设备和加速器间的数据共享采用无驱动的数据移动方式。传统的 PCIe 加速器需要驱动对加速器写入和读出数据，这增加了延时和计算开销。采用无驱动的数据移动方式，CCIX 还可以将系统内存扩展至主设备的内存之外。

基于 CCIX，每个支持 CCIX 的设备的行为与现有 NUMA(非统一内存访问)操作系统中的节点类似。这种基于内存的方法利用了现有的操作系统功能。在这种模式下，用来共享的所有数据结构都放在处理器和加速器都可访问的共享内存里。这种数据共享模型可以消除加速器特定的控制与管理驱动，允许加速器资源由一个中心调度器安排的长时间运行的任务来调用。这个调度器可以是操作系统调度程序的一部分，或和操作系统调度程序协同。这能简化在虚拟机或容器上运行的应用所用的软件库，允许开发者用任何语言、有完整的工具支持来编写常规的应用软件。

6 GenZ, CXL, NVLINK, OpenCAPI, CCIX 乱战

纵观数据中心业界，底层技术方面其实正处在一场架构变革的初始点，这场变革就是 I/O 总线的网络化以及 I/O 资源的大规模池化。

众所周知，在开放平台下，PCIE 是目前高性能 I/O 设备普遍采用的总线类型，目前已经到 Gen4，很快会到 Gen5。但是 PCIE 总线的树形拓扑以及有限的设备标识 ID 号码范围，导致其无法形成一个大规模网络，这个问题在 NVMe 盘未普及之前显得不那么是个问题，但是 NVMe 盘得道广泛应用之后，会占用大量的 PCIE 同道数量，这使得原本捉襟见肘的 PCIE 总线资源更加紧张，GPU、NIC、FPGA/ASIC 卡的接入数量就会被 NVMe 盘挤占，虽然可以用 PCIE Switch 来解决同道数量不够的问题，但是对于 PCIE 总线设备 ID 号的不足，PCIE Switch 也并没有方便的解决方案，虽然可以用 NTB 方式来解决，但是这又需要在 Host 端 OS 内核底层增加一层驱动，这种对内核的变更使得该方案只适合用在封闭系统里，比如一些企业级存储系统。

PCIE 的另外 2 个问题是，存储器地址空间隔离、不支持 Cache Coherency 事务。PCIE 网络中的地址空间虽然也是 64bit，但是其原本设计初衷是该网络的地址空间是私有的，与 CPU 的地址空间并不是原生融合的，需要地址翻译寄存器来做基址地址翻译，而对于 Intel 平台，这个翻译靠软件来执行，而且规则也很简单，就是不翻译，维持原有地址。

这样的话，虽然 CPU 可以直接访问 PCIE 网络中的地址，而 PCIE 设备也可以访问 CPU 地址空间中的地址（比如 Host RAM），但是由于 PCIE 事务层不支持 Cache Coherency 事务的处理，所以 PCIE 设备端无法缓存 CPU 地址域中的数据，所以每次都必须去访问 Host RAM 来获取数据。

设备端无法缓存的话，每次都访问 Host RAM 有什么问题么？问题就是延迟太高，通过 DDR 同道直接访问内存延迟在 40ns 左右，而通过 PCIE 访问则会在 100ns 级别，如果是小尺寸访存请求，性能将会比较差。正因如此，对于目前的 GPU、FPGA/ASIC 等加速卡，普遍采用现将数据从 Host RAM 拷贝到加速卡上的内部存储器，计算，算完了再拷贝回 Host RAM。

很显然，只要两招就能解决上述问题，第一就是将总线速率提升，降低访问延迟，第二就是在物理链路之上增加对 Cache Coherency（下简称 CC）事务的处理，也就是在设备一侧增加一个 CC Agent 与 CPU 一侧的 Agent 交互。关于 CC 缓存一致性详细底层架构和原理可阅读《大话计算机》一书第 6 章。

基于上面两个考虑，IBM 最早推出了 CAPI (Coherent Accelerator Processor Interface) 接口，当年我还写了一篇文章来介绍。CAPI1.0 接口复用了 PCIE 物理层、链路层和事务层，并利用 PCIE 数据包的 Payload 字段隧道化封装了 CC 和 CAPI 控制事务（这两者后文统称 CAPI 事务），在 CPU 一侧增加针对 CAPI 事务的解析处理模块，然后在加速卡芯片一侧，IBM 提供 CAPI 的事务解析逻辑 IP 用于集成到第三方芯片中。在后来的 CAPI 版本中，逐渐演化成了 OpenCAPI，有了自己的物理、链路、事务层，以及独立的处理模块，与 PCIE 分离。IBM 下一步甚至会用 OpenCAPI 来连接 DDR 内存，内存卡上会有一颗 OpenCAPI~DDR4/5 的桥接芯片来负责适配。IBM 不愧为业界大佬，在这方面风范犹存而且先带领大家玩了起来，不过毕竟是廉颇已老，跟风者寥寥无几，GPU/FPGA/ASIC 厂商似乎并不是十分买账，毕竟 OpenPower 的机器在数据中心占比实在是太低了。

时隔 4 年，Intel 也跟了上来，在 2019 年 3 月份推出了 Compute Express Link (CXL) 协议接口，其与 CAPI 酷似，也是将 CXL 协议封装到 PCIE 链路层数据包中传送，并在 CPU 端的 PCIE 总控后端按照事务标识分流 CXL 专属事务给 CXL 处理逻辑处理。IBM 毕竟廉颇已老，如今是牙膏厂 Intel 的天下，Intel 振臂一呼，那都得跟上来。

IBM 和 Intel 各玩各的，这业界不能就让他俩这么单独玩。早在几年前，GenZ (Generation Z) 这套协议雏形初现，如今已经是羽翼丰满。GenZ 除了具有 CAPI 和 CXL 的功能之外，还将总线拓展成了交换式网络，也就是加入了 GenZ Switch 这个角色，可以将大量的 CPU、I/O 设备、存储器挂接到 GenZ 网络上，实现更高的扩展性。GenZ 是一个开放组织，其不依赖于任何 CPU 平台，目前已经有 60 多家厂商加入了 GenZ 组织，其中包括 Intel 的老对手 AMD。然而 AMD 对 GenZ 似乎不是很上心，不过 Intel 推出了 CXL 之后，AMD 估计应该心里多少都有点涟漪，我猜测 AMD 应该不会自己另起炉灶搞一套总线，而应该斩钉截铁的站队 GenZ 阵营，直接在 CPU 里继承 GenZ 主控，或者采用 Microchip (收购了 Microsemi、PMC) 等 I/O 控制器老牌公司的外置 GenZ I/O 控制器。

再说说淫威大这边的事。Nvidia 给人感觉一直是 diaodiao 的，GenZ、CXL、人家一个也没看上，也不跟风，而是自己搞了一套 NVLINK，

而且反扑了 IBM，IBM 大佬竟然在 Power CPU 里继承了 NVLINK 控制器，奇怪的是 NV 竟然还是 OpenCAPI 组织成员，但最终却反客为主，够 diao 吧？NVLINK 可以支持 CPU-GPU 间链路也可以支持 GPU-GPU 间链路，而且 NV diao 上加 diao，连 NVLINK Switch 都自己搞出来了，而且还搞出了搭载 16 个 GPU+NVLINK Switch 的整机方案：DGX/HGX，以及即将推出的第二代整机平台方案。整个平台满配据说要数百万¥。

现在的时间线是：CAPI->GenZ->NVLINK->CXL。这还没完，ARM 平台不掺和进来这出戏就不够精彩。ARM 选择了加入另一个开放的访存和 I/O 网络平台（CCIX）Cache Coherent Interconnect for Accelerators。CCIX 酷似 GenZ。AMD 也在这个平台里。完整的时间线应该是这样的：CAPI->GenZ->CCIX->NVLINK->CXL。

那么，这几员大将，到底谁能在这场架构变革中胜出？冬瓜哥的观点是这样的，OpenCAPI 基本不用考虑，IBM 的生态目前无法与 Intel 抗衡，而且 OpenCAPI 只支持 CPU 直连，尚未支持 OpenCAPI Switch，后续估计也不会搞。CXL 嘛，Intel 的面子大家还是要给的，加上 Intel 目前的生态控制力，大家就算是为了规避风险，也必然会选择跟风，所以胜算较大。但是如 OpenCAPI 一样，CXL 目前仅仅只支持 CPU 点对点直连拓扑，尚未有 CXL Switch 的打算，不过冬瓜哥猜测，Intel 更高概率不会做外部 Switch，而 Microchip 这类外部 I/O 主控、Expander/Switch 公司在这方面更有经验和积累，这块蛋糕给它们来做似乎更加合理，再加上业界都不想让牙膏厂继续一家独大，所以 GenZ 前景还是非常明朗的，从加入它的众多厂商就可以看出。老牌 I/O 控制器和 Switch 厂商 Microchip 目前加入了 GenZ 和 OpenCAPI，而且是 GenZ 的核心贡献者，有没有可能将来是 CXL+GenZ 的融合架构呢？不是没有这个可能，因为在 CXL 发布时，Intel 官方特地邀请了 GenZ 的发言人，其中似乎有强烈暗示 CXL 和 GenZ 应该是趋于融合，前者只管点对点直连，到了外面则交给 GenZ。

至于 NVLINK，既然 NV 要 diao 到底，那谁也拦不住，毕竟 GPU 集群使用的越来越多，GPU 和 GPU 之间怎么勾搭那是人家自己的事情。CCIX，目前看来前景不是十分明朗，AMD 和 ARM 这对难兄难弟能否引领 CCIX 独占鳌头，很难说。

7 路由机制

PCIe 共有三种路由方式：基于地址（Address）路由，基于设备 ID（Bus number + Device number + Function Number）路由，还有就是隐式（Implicit）路由。

TLP类型	路由方式
Memory Read/Write TLP	地址路由
Configuration Read/Write TLP	ID路由
Completion TLP	ID路由
Message TLP	地址路由或者ID路由或者隐式路由

7.1 地址路由

在 PCIe 总线中，存储器读写和 I/O 读写 TLP 使用基于地址的路由方式。

当一个 TLP 进行数据传递时，可能会经过多级 Switch，最终到达目的地。Switch 将根据存储器读写或者 IO 读写请求 TLP 的地址将报文传递到合适的 Egress Port 上。一个 Switch 包含几个端口，就包含几个虚拟 PCI-to-PCI 桥，每个虚拟 PCI-to-PCI 桥都有一个配置空间。

在虚拟 PCI-to-PCI 桥的配置空间中，包含一个桥片能够接受的物理地址范围，PCIe 总线通过这个物理地址实现基于地址的路由，这段配置寄存器如下图所示，当系统软件初始化 PCI 总线时，将合理的设置这些寄存器，之后当 TLP 通过这些 Switch 时将根据这些寄存器选择合适的路径。

Secondary Status																I/O Limit								I/O Base								+01Ch															
Memory Limit																Memory Base																								+020h							
Prefetchable Memory Limit																Prefetchable Memory Base																								+024h							
Prefetchable Memory Base Upper 32 Bits																																+028h															
Prefetchable Memory Limit Upper 32 Bits																																+02Ch															
I/O Base Limit 16 Bits																U/O Base Upper 16 Bits																								+030h							

- 当一个 Endpoint 收到一个 Memory Read 或者 Memory Write TLP，它会把 TLP Header 中的地址跟它 Configuration 当中的所有 BAR 寄存器比较，如果 TLP Header 中的地址落在这些 BAR 的地址空间，那么它就认为该 TLP 是发给它的，于是接收该 TLP，否则就忽略。
- 当一个 Switch 上游端口收到一个 Memory Read 或者 Memory Write TLP，它首先把 TLP Header 中的地址跟它自己 Configuration 当中的所有 BAR 寄存器比较，如果 TLP Header 当中的地址落在这些 BAR 的地址空间，那么它就认为该 TLP 是发给它的，于是接收该 TLP（这个过程与 Endpoint 的处理方式一样）；如果不是，然后看这个地址是否落在其下游设备的地址范围内，这就是通过检查上述 Base 和 Limit 寄存器来实现的，如果是，说明该 TLP 是发给它下游设备的，因此它要完成路由转发；如果地址不落在下游设备的地方范围内，说明该 TLP 不是发给它下面设备的，因此不接受该 TLP。
- 如果是 TLP 从下游往上游走，则对应的 Downstream Port 首先把 TLP Header 中的地址跟它自己 Configuration 当中的所有 BAR 寄存器比较，如果 TLP Header 当中的地址落在这些 BAR 的地址空间，那么它就认为该 TLP 是发给它的，于是接收该 TLP（跟前面描述一样）；如果不是，然后看这个地址是否落在其下游设备的地址范围内（是否在 Memory Base 和 Memory Limit 之间），如果是，这个时候不是接受，而是拒绝或者转发；相反，如果地址不落在下游设备的地方范围内，Switch 则把该 TLP 向上传送。

7.2 ID 路由

在一个 PCIe 拓扑结构中，由 ID = Bus number+Device number+Function Number (BDF)能唯一找到某个设备的某个 Function。这种按设备 ID 号来寻址的方式叫做 ID 路由。Configuration TLP 和 Completion TLP（以 C 打头的 TLP）按 ID 路由，Message 在某些情况下也是 ID 路由。

- 当一个 Endpoint 收到一个基于 ID 路由的 TLP，它用自己的 ID 和收到 TLP Header 中的 BDF 比较，如果是给自己的，就收下 TLP，否则就拒绝。在 PCI Express 中，每台设备每次在主链路上检测到配置写 (Type 0) 时，就捕获 TLP Header 中的 Bus Number 和 Device Number。复位时，系统中所有总线和设备号复位成 0，因此设备将不响应任何不是配置周期的事务，直到完成至少一个配置写 (Type 0)。注意：规范并没有定义要求设备保存捕获的 Bus Number 和 Device Number，但实现时必须这么做。

对一个 Switch 来说，每个 Port 靠近 RC (上游) 的那根 Bus 叫做 Primary Bus，其 Number 写在其 Configuration Header 中的 Primary Bus Number 寄存器；每个 Port 下面的那根 Bus 叫做 Secondary Bus，其 Number 写在其 Configuration Header 中的 Secondary Bus Number 寄存器；对上游端口，Subordinate Bus 是其下游所有端口连接的 Bus 编号最大的那个 Bus，Subordinate Bus Number 写在每个 Port 的 Configuration Header 中的 Subordinate Bus Number 寄存器。

- 当一个 Switch 收到一个基于 ID 寻址的 TLP，首先检查 TLP 中的 BDF 是否与自己的 ID 匹配，如匹配，说明该 TLP 是给自己的，收下；否则，则检查该 TLP 中的 Bus Number 是否落在 Secondary Bus Number 和 Subordinate Bus Number 之间，如果是，

说明该 TLP 是发给其下游设备的，然后转发到对应的下游端口；如果其他情况，则拒绝这些 TLP。

7.3 隐式路由

只有 Message TLP 才支持隐式路由。在 PCIe 总线中，有些 Message 是与 RC 通信的，RC 是该 TLP 的发送者或者接收者，因此没有必要明明白白的指定地址或者 ID，而是采用“你懂的”的方式进行路由，这种路由方式为隐式路由。Message TLP 还支持地址路由和 ID 路由，但以隐式路由为主。

- 当一个 Endpoint 收到一个 Message TLP，检查 TLP Header，如果是 RC 的广播 Message (011b) 或者该 Message 终结于它 (100b)，它就接受该 Message。
- 当一个 Switch 收到一个 Message TLP，检查 TLP Header，如果是 RC 的广播 Message (011b)，则往它每个下游端口复制该 Message 然后转发；如果该 Message 终结于它 (100b)，则接受该 TLP；如果下游端口收到发给 RC 的 message，往上游端口转发便是。

上面说的是 Message 使用隐式路由的情况。如果是地址路由或者 ID 路由，Message TLP 的路由跟别的 TLP 一样，不赘述。