5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

对于MSI-X，允许函数缓存来自未屏蔽的MSI-X表条目的地址和数据值。但是，无论何时，软件通过清除其屏蔽位或通过清除功能屏蔽位来取消当前屏蔽的MSI-X表条目的屏蔽，功能必须更新从该条目缓存的任何地址或数据值。如果软件在未屏蔽条目时更改了条目的地址或数据值，则结果未定义。



**执行说明**

使用MSI/MSI-X的每矢量掩蔽

使用MSI或MSI-X的设备和驱动程序面临的挑战是，

消息被生成。如果硬件未能发送软件期望的中断消息，则中断事件可能“丢失“。如果硬件发送软件不期望的中断消息，则“虚假”中断

可能导致。

每矢量掩蔽（PVM）可用于协助这种协调。例如，当软件中断服务例程开始时，它可以屏蔽向量以帮助避免“虚假”中断。 在中断服务例程服务了它所知道的所有中断条件之后，它可以解除对向量的屏蔽。如果有任何中断

如果条件仍然存在，则需要硬件来生成新的中断消息，以保证没有中断事件丢失。

PVM是MSI-X的标准功能，也是MSI的可选88功能。对于实现MSI的设备，强烈建议也实现PVM。

<6.1.4.6>硬件/软件同步

如果一个函数在被软件确认之前多次向我发送相同向量的消息，

消息被保证被服务。如果必须服务所有的内存，则需要设备驱动程序握手。换句话说，一旦函数发送了向量A，它就不能再次发送向量A，直到它的设备显式地启用它这样做

驱动程序（前提是必须服务所有消息）。如果某些消息可能丢失，则不需要设备驱动程序握手。对于支持多个向量的函数，一个函数可以发送多个唯一的向量，并保证每个向量

将提供唯一消息。例如，函数可以发送向量A，然后发送向量B，而无需任何设备驱动程序握手（向量A和向量B都将得到服务）。



88.例外：在SR-IOV设备中，实现MSI的任何PF或VF必须实现MSI PVM

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0



执行说明

维修MSI和MSI-X中断

当系统软件分配给函数的MSI或MSI-X向量少于其请求时，

函数内的每个源都需要唯一的向量，可能需要共享单个向量。没有

硬件和软件之间的适当握手，硬件可能发送比软件期望的更少的消息，或者硬件可能发送软件认为是无关的消息。

一种相当复杂但资源密集型的方法是将一个专用的事件队列与每个

分配向量，具有用于管理每个事件队列的生产者和消费者指针这样的事件队列

通常驻留在主机存储器中。函数充当生产者，软件充当消费者。一个函数中的多个中断源可以根据需要分配给每个事件队列。每次中断

如果源需要发出中断信号，则函数将条目放置在适当的偶数t队列中（假设

有空间），更新生产者指针的副本（通常在主机内存中），并在必要时发送具有相关向量的中断消息，以通知软件事件队列需要服务。给定事件队列的中断服务例程处理在其事件队列上找到的所有条目，如

生产者指针。每个事件队列条目标识中断源，并可能标识有关事件性质的附加信息。事件队列和生产者/消费者指针的使用可用于

保证当多个中断源被迫共享一个向量时，中断事件不会被丢弃。 在发送与同一事件队列相关联的多个消息之间不需要额外的握手，以保证每个消息都得到服务。事实上，各种“中断”的标准技术

“合并”可以用于避免为发生的每个事件发送单独的消息，特别是在事件的密集突发期间。

在更适度的实现中，函数的MSI或MSI-X逻辑的硬件设计 如果未启用MSI或MSI-X，则在虚拟INTx线路上将发生到断言的转换。例如，考虑一个场景，其中两个中断事件（可能来自函数内的不同中断源）快速连续发生。 第一个事件导致发送消息。中断服务之前

当例程有机会服务第一事件时，第二事件发生。 在这种情况下，只发送一条消息，因为第一个事件在第二个事件发生时仍然有效（虚拟INTx线信号将只有一次转换到ssertion）。

用于上述实现的一种握手方法是使用标准的Per-Vector Masking，并允许

多个中断源与每个向量相关联。给定向量

vector在服务任何相关中断事件之前清除Mask位，并在服务所有已知事件之后清除Mask位。(This可以是任何数量的事件）。当发生新事件时，

掩码位为置位导致挂起位为置位。如果一个或多个相关事件在矢量的掩码位被清除时仍处于未决状态，则函数立即发送另一条消息。

不实现每向量屏蔽的MSI函数的握手方法用于向量

一个服务程序，用于在清除最后一个中断事件后重新检查所有相关的中断事件 挂起中断事件。如果发现另一个事件处于活动状态，则在同一中断服务例程调用中为其提供服务，并重复完整的重新检查，直到没有发现未决事件。这确保了如果在前一个中断事件被清除之前发生了额外的中断事件，则函数不会发送额外的中断消息，新事件将作为当前中断服务的

例行调用。

这种替代方案具有一个向量的中断服务例程处理已经生成新中断消息的中断事件的潜在副作用。中断服务例程调用导致 从新消息中可以发现没有未决中断事件。这种情况有时被称为 伪中断，并且使用这种方法的软件必须准备好容忍它们。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

由于MSI或MSI-X消息是已发布请求，事务排序规则禁止MSI或MSI-X消息传递函数先前发送的已发布请求。系统必须保证，作为给定消息的结果而调用的中断服务例程将观察到在该消息之前到达的Posted Request所执行的任何更新。因此，设备驱动程序的中断服务例程不需要从设备寄存器读取以确保数据

与恶意发布请求的一致性。但是，如果多个MSI-X表条目共享同一向量，则中断

服务例程可能需要从某个设备专用寄存器读取以确定哪些中断源需要服务。

<6.1.4.7>消息交易接收和订购要求

与所有内存写入事务一样，包括中断消息（中断）目标的设备

接收者）被要求作为完成者完成所有中断消息事务，而不要求作为完成者首先完成其他事务。通常，这意味着消息接收器必须独立于CPU何时服务中断来完成中断消息事务。例如，每次中断接收器接收到中断消息时，它可以在内部寄存器中设置一个位，指示该消息已被接收，然后 在总线上完成交易。适当的中断服务例程将在以后被调度，因为 位已设置。不允许消息接收方延迟总线上中断消息的完成

等待来自处理器的中断正在被服务的确认。当多个设备同时发送中断消息时，此类依赖关系可能会导致死锁。

虽然中断消息在整个PCIExpress层次结构中保持严格的顺序

中断消息不保证中断将被服务的任何顺序。因为信息接收者

必须完成所有中断消息事务，而不考虑实际服务中断的时间，

消息接收者通常不会维护有关接收中断的顺序的任何信息。从不同设备接收的中断消息和从同一设备接收的多个消息都是如此。如果一个设备需要一个中断消息在另一个之前被服务，则该设备不得发送第二个中断 直到第一个消息被服务。

6.1.5 PME支持

PCI Express支持来自本机PCI Express设备以及支持PME的PCI设备的电源管理事件。 PME信令是使用带内事务层PME消息（PM\_PME）完成的，如第5章所述。

6.1.6原生PME软件模型

PCI中断感知软件可以启用其中根联合体经由中断发信号通知PME的模式。 当配置本机PME支持时，根端口接收PME消息并在其根状态寄存器中设置PME状态位。如果 软件已将根控制寄存器中的PME启用位设置为1b，根端口随后生成

打扰了

如果使用INTx消息为电平触发中断信令启用根端口，则只要满足以下所有条件，就必须断言虚拟INTx

·命令寄存器中的禁用位设为0b。

·根控制寄存器中的PME复位使能位设为1b。

·根状态寄存器中的PME状态位置1。

请注意，在请求服务时，同一功能内的所有其他中断源将断言同一虚拟INTx线。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

如果使用MSI或MSI-X为边沿触发中断信令启用根端口，则每次以下条件的逻辑与从“否”转换为“真”时，必须发送中断消息

·相关联的向量未被掩蔽（如果MSI不支持PVM，则不适用）。

·根控制寄存器中的PME复位使能位设为1b。

·根状态寄存器中的PME状态位置1。

请注意，PME和热插拔事件中断（当两者都实现时）始终共享相同的MSI或MSI-X向量，如PCI Express功能寄存器中的“PME消息编号”字段所示。

此中断的软件处理程序可以通过读取根端口中的根状态寄存器中的PME转发器ID字段来确定哪个设备发送了PME消息。它通过向

根状态寄存器。更多详情请参见第7.5.3.14节。

根复合体事件收集器为根复合体集成端点（RCiEP）的上述功能提供支持。

6.1.7遗留PME软件模型

然而，遗留软件将不理解用于发信号通知PME的这种机制。在存在合法系统软件的情况下，根联合体中的系统功率管理逻辑接收PME消息并通过实现特定的机制通知系统软件。根复合体可以利用在根复合体中的转发器ID PM\_PME通知系统软件哪个设备导致电源管理事件。

由于PME是通过消息传递的，因此PME在PCI Express中具有边缘触发的语义，这与

用于常规PCI的电平触发PME机制。根联合体的责任是从系统软件中提取这种差异，以保持与常规PCI系统的兼容性。

6.1.8操作系统电源管理通知

为了保持与非PCI PCI PCI感知系统软件的兼容性，系统电源管理逻辑必须由固件配置为默认使用传统的PME信令机制。在启用本机的基于中断的PME信令之前，PCI可感知的服务器软件必须通知固件。响应于该通知，如果需要，系统固件必须重新配置根复合体以禁用用信号通知PME的传统机制。的细节 该固件通知超出了本说明书的范围，但是由于它将在系统运行时执行，因此对该通知的响应必须不干扰系统软件。因此，在控制切换到

操作系统，固件不能写入可用的系统存储器或任何PCI Express资源（例如，配置空间结构）。

6.1.9PCI Express和PCI层次之间的PME路由

支持PME的传统PCI和PCI-X设备断言PME#引脚以发出电源管理事件的信号。来自PCI或PCI-X设备的PME#信号可以由PCI Express带内PME消息转换为 桥接或直接路由到根复合体。

如果来自PCI或PCI-X设备的PME#信号被直接路由到根联合体，则它使用与在当前PCI系统中使用的相同的机制来发信号通知系统软件。根复合体可以可选地提供对经由中断从PCI或PCI-X设备向系统软件发信号通知PME的支持。在这种情况下，建议RootComplex

检测断言PME#的PCI或PCI-X设备的总线、设备和功能号，并使用此信息

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

填写根端口中的PME路由器ID字段，该根端口发起包含PCI或PCI-X设备的层次结构。如果这是不可能的，根复合体可以可选地将根端口的路由器ID写入该字段。

由于RCiEP不包含在由根端口发起的任何层次结构域中，因此不与根端口相关联的RCiEP不包含在由根端口发起的任何层次结构域中 根复合体事件收集器使用与当前PCI系统中使用的相同的机制来向PME的系统软件发送信号。根复合体事件收集器（如果实现的话）为相关联的RCiEP启用PCI Express本机PME模型。

6.2错误信号和日志记录

在本文件中，确定了必须检查的错误和可以选择检查的错误。每个此类错误都与端口或特定设备（或多功能设备中的功能）相关，

关联连同错误的描述一起给出。 本节将讨论如何对错误进行分类和报告。

6.2.1范围

本节介绍PCI Express组件的错误信号和日志记录要求其中包括错误

这些错误发生在PCI Express接口本身上，这些错误代表PCI上发起的传输

Express，以及组件内发生的与PCIExpress接口相关的错误。 本节不关注组件内发生的与PCI Express接口无关的错误。这种类型的错误

通过采用设备专用干扰的专有方法来更好地处理信令。

PCI Express定义了两种错误报告模式：基线上限能力和高级错误报告能力。所有PCI Express设备都需要基线错误报告功能，并定义了最小错误报告 要求. 高级错误报告功能是为更健壮的错误报告而定义的 具有特定的PCI Express功能结构（有关此可选功能的定义，请参阅第7章）。 本节明确指出了基线和高级错误报告功能之间的所有错误处理差异。

所有PCI Express设备都支持现有的非PCIExpress感知软件，通过将PCI Express错误映射到现有的PCI报告机制以及PCI EExpress特定机制来进行错误处理。

6.2.2错误分类

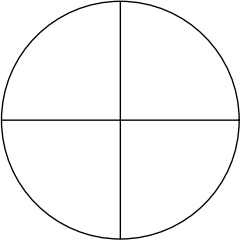
PCI Express错误可以分为两种类型：不可纠正的错误和可纠正的错误。这种分类

将导致功能故障的错误与导致性能下降的错误分开。无法纠正的错误可以进一步分为致命或非致命（见[图6-1）](#bookmark2)。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0



ERR\_COR



数据链路 物理

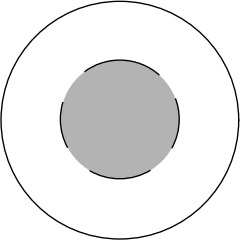
内部 交易

可校正错误



错误\_非致命 错误\_致命

|  |  |
| --- | --- |
| 数据链路  Fa | 物理tal |
| 内部 | 交易 |



不可纠正的差错

OM13827A

图6-1错误分类

将错误严重性分类为致命、不可纠正和可纠正为平台提供了以下机制：

将错误映射到合适的处理机制。例如，平台可以选择使用低优先级的性能监控软件来响应可纠正的错误。这样的软件可以计算出可纠正的频率

错误并提供链接完整性信息。另一方面，平台设计者可能会选择将致命错误映射到系统范围的reset。将这些PCI Express严重性级别映射到平台级严重性是平台设计人员的决定。

<6.2.2.1>可更正的错误

可纠正的错误包括硬件可以在不丢失任何信息的情况下恢复的错误情况。

硬件可以纠正这些错误，不需要软件干预。例如，TLP中可能由数据链路级别错误纠正的LCRC错误被视为可纠正错误。测量链路级可纠正错误的频率可能有助于分析链路的完整性。

可纠正的错误还包括事务级别的情况，其中一个代理检测到TLP错误，但另一个代理负责在需要时采取任何恢复操作，例如使用单独的后续

交易日由于恢复代理可能能够纠正错误，因此可以将检测代理配置为将错误报告为可纠正。如果确实需要恢复操作，则如果恢复代理决定不尝试恢复，则恢复代理必须将错误报告为不可纠正。

下游端口遏制（DPC）的触发不会作为n错误处理，但可以像错误一样发出信号 由于利用DPC的软件有时可以从触发DPC的不可纠正的错误中重新覆盖，因此， 参见[第6.2.10节](#bookmark4)。用于DPC信令的ERR\_CORM消息

固件，并且可以经由ERR\_COR子类字段来指示。

类似地，ERR\_COR可以由系统固件中介（SFI）能力用于发信号通知系统固件，并且必须通过ERR\_COR子类字段来指示。 参见[第6.7.4节。](#bookmark5)

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.2.2.2>无法纠正的错误

不可纠正的错误是那些影响接口功能的错误条件。没有机制

在本规范中定义，以纠正这些错误。报告不可纠正的错误类似于在PCI/PCI-X中断言SERR#。为了使系统更稳健地处理错误，本规范进一步将不可纠正的错误分类为致命错误和非致命错误。

6.2.2.2.1致命错误

致命错误是不可纠正的错误条件，使特定的链接和相关硬件不可靠。为

致命错误，可能需要重置链路上的组件以返回可靠操作。平台对致命错误的处理以及限制这些错误影响的努力是平台实现特定的。

6.2.2.2.2非致命错误

非致命错误是不可纠正的错误，导致特定交易不可靠，但链路在其他方面完全正常。将非致命错误与致命错误隔离在设备或系统中提供了触发器/接收器逻辑

管理软件有机会从错误中恢复，而无需重置链路上的组件 干扰其他正在进行的交易。与错误事务不关联的设备不受错误影响。

6.2.3错误信号

存在三种互补机制，其允许代理检测错误以警告系统或另一设备错误已经发生。第一种机制是通过完成状态，第二种方法是使用 带内错误消息，第三种是错误转发（也称为数据中毒）。

请注意，检测错误的代理有责任适当地发出错误信号。

[第6.2.7节](#bookmark9)描述了所有错误以及当检测到错误时要求硬件如何响应

<6.2.3.1>完成状态

完成报头中的完成状态字段（当状态不是成功完成时

关联请求失败（请参见www.example.com部分2.2.8.10）。这是错误报告的一种方法，它使调试器能够 将错误与特定请求相关联。换句话说，由于在完成返回之前未发布的请求不被认为是完成的，因此状态完成字段为调试器提供了一个“修复”问题的机会，

一些更高级别的协议（在本规范的范围例如，如果向可预取的

内存空间，并且完成返回不支持的请求完成状态，则调试器不会 如果它选择重新发出读取请求，则将违反本规范。注意，从PCI Express的角度来看，重新发布的读取请求是不同的请求，并且在初始请求和重新发布的请求之间没有关系（在PCI Express上）。

<6.2.3.2>错误消息

错误消息被发送到根复合体，用于根据错误的严重性报告错误的检测。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

来自PCI Express或传统端点的错误消息将发送到相应的根端口。源自根端口本身的错误通过同一根端口报告。

如果实现了可选的根复杂事件收集器，则源自RCiEP的错误将被发送到

根复合体事件收集器。源自根复杂事件收集器本身的错误将通过同一根复杂事件收集器报告。根复杂事件收集器必须将支持的RCiEP声明为

RCiEP是其功能的一部分;每个RCiEP必须与至多一个根复合体事件收集器相关联。

当检测到多个相同严重性的错误时，可以针对相同严重性的不同错误合并具有相同错误标识符的相应错误消息。对于检测到的以下错误，必须至少发送一条错误消息：

每个严重程度。然而，请注意，在某些情况下，检测到给定错误将排除某些错误的报告。参见[第6.2.3.2.3](#bookmark11)。还要注意[第6.2.4](#bookmark12)中关于非函数特定错误的特殊规则，

多功能设备。

表6-1错误消息

|  |  |
| --- | --- |
| 错误消息 | 描述 |
| ERR\_COR | 当功能或设备在PCI Express接口上检测到可纠正的错误时，会发出此消息。有关[6.2.2.1](#bookmark3)可纠正错误的定义，请参阅第www.example.com节。 |
| 错误\_非致命 | 当功能或设备在PCI Express接口上检测到非致命、无法纠正的错误时，会发出此消息。参见[第6.2.2.2.2 f节](#bookmark8)或非致命性、不正确表错误的定义。 |
| 错误\_致命 | 当功能或设备在PCI Express接口上检测到致命的、无法纠正的错误时，会发出此消息。参见[第6.2.2.2.1 f节](#bookmark7)或致命、不可纠正错误的定义。 |

对于这些消息，根Complex通过消息报头的发起者ID来标识消息的发起者。根复合体将se错误消息转换为平台级事件。

执行说明

ERR\_COR、ERR\_NONFATAL和ERR\_FATAL

在[PCIe-1.0]和[PCIe-1.0a]中，给定的错误可以是可纠正的、非致命的或致命的。 假设启用了信号发送，可纠正的错误总是用ERR\_COR发送信号，非致命错误总是用ERR\_NONFATAL发送信号，致命错误总是用ERR\_FATAL发送信号。

在支持基于角色的错误报告的后续规范中，有时会发出

使用ERR\_NONFATAL，有时使用ERR\_COR发出信号，有时根本不发出信号，这取决于检测错误的代理的角色以及代理是否实现AER（参见[第6.2.3.2.4节）](#bookmark13)。对一些 平台，发送ERR\_NONFATAL将阻止其他代理尝试恢复或确定

错误的最终处理。对于检测代理不是确定错误的最终处置的适当代理的情况，具有AER的检测代理可以用ERR\_COR发信号通知非致命错误

其用作对软件的咨询通知。对于检测代理是适当的代理的情况，代理使用ERR\_NONFATAL发出非致命错误的信号。

对于一个给定的不可纠正的错误， 在检测到该错误的操作中，软件可以配置实现AER的检测代理，以将该错误的严重性升级为致命的。检测代理（如果启用）将始终使用ERR\_FATAL发出致命错误的信号，而不管代理的角色如何。

软件应该认识到，单个事务可以由多个代理使用不同类型的 错误消息。例如，中毒的TLP可能由中间接收器用ERR\_COR发信号通知，而最终目的地接收器可能用ERR\_NONFATAL发信号通知。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.2.3.2.1不可纠正的错误严重性编程（高级错误报告）

对于实现高级错误报告功能的设备功能， 允许将每个无法纠正的错误编程为致命或非致命。无法纠正的错误无法使用定义的PCI Express机制恢复。但是，某些平台或设备可能会认为特定错误对链路或设备是致命的，而另一个平台则认为该错误不是致命的。无法纠正的错误严重性的默认值

寄存器作为本规范的起点，但如果设备驱动程序或平台软件需要更健壮的错误处理，则可以对寄存器进行重新编程。

基线错误处理不支持严重性编程。

6.2.3.2.2掩盖个别错误

[第6.2.7节](#bookmark15)列出了受本规范管辖的所有错误，并描述了上述每个错误消息的发布时间。通过使用报告功能，可以按类别（可纠正、非致命、致命）传输 设备控制寄存器的使能位（参见第www.example.com节7.5.3.4）或PCI逗号寄存器中的SERR#使能位（参见第7.5.1.1.3节）。

对于实现高级错误报告功能的设备，

可纠正错误屏蔽寄存器允许独立屏蔽每个错误条件。如果设备控制寄存器和PCI命令寄存器中的组合设置未启用特定错误类别的消息，则在没有相应掩码寄存器的值的情况下，不会发送该类消息。

如果在检测到单个错误时屏蔽了该错误，则其错误状态位仍然有效，但是没有错误报告消息被发送到根复合体，并且该错误不被记录在报头日志、TLP前缀日志或第一错误指针中。

6.2.3.2.3错误污染

如果给定事务的错误条件没有与最重要的事件隔离，则可能发生错误污染。 例如，假设物理层检测到接收器错误。在物理层检测到该错误，并向根复合体报告错误。为了避免此错误传播并在上层引起后续错误 例如，数据链路层的TLP错误），使得更难以确定错误的根本原因

数据链路层或事务层将不报告对同一分组发生的后续错误。类似地，当数据链路层检测到错误时，对于同一分组发生的后续错误将不会被报告。

交易层。此行为仅适用于与特定数据包关联的错误-每次发生时都会报告其他错误。

已纠正的内部错误是其影响已被组件掩盖或解决的错误;请参阅[第6.2.9节](#bookmark17)[f](#bookmark18)或详细信息。因此，已纠正的内部错误不会造成错误污染，并且应在检测到错误时报告。

对于在事务层中检测到的错误和不可纠正的内部错误，允许并建议对单个接收到的TLP报告不超过一个错误，并使用以下优先级（从最高到最低）

· 无法纠正的内部错误

· 接收器溢出

· 变形TLP

· ECRC检查失败

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·AtomicOp出口BI锁定

·TLP前缀被阻止

·ACS违规

·[MC封闭式TLP](#bookmark19)

· 不支持的请求（UR）、完成程序中止（CA）或意外完成

· 接收到有毒TLP或有毒TLP出口受阻

由于处理接收到的TLP时不会检测到完成错误，因此该错误不在上述优先级列表中。同一项目符号下列出的错误是互斥的，因此它们的相对顺序无关紧要。

6.2.3.2.4咨询性非致命错误案例

在某些情况下，非致命错误的检测器并不是确定错误是否可恢复或者是否需要任何恢复操作的最合适的代理。例如，如果软件试图执行

如果从不存在的设备或功能读取配置，则完成中产生的UR状态将向软件发出错误信号，并且除了通过发送ERR\_NONFATAL消息来发出错误信号之外，软件不需要向完成者发出错误信号。事实上，在某些平台上，使用ERR\_NONFATAL发出错误信号会导致系统错误，

正常的软件探测。

“Advisory Non-Fatal Error”

完成者或接收者）和特定错误。在这种情况下，具有AERs的代理通过向软件发送ERR\_COR消息（而不是发送ERR\_NONFATAL）来通知非致命错误（如果启用）。没有AER的代理

对于这些情况不发送错误消息，因为接收ERR\_COR的软件将无法区分咨询性非致命错误情况与用于评估链路完整性的可纠正错误情况。

以下是咨询性非致命错误的具体案例。请注意，同一个TLP中可能存在来自相同或不同错误类别（可纠正、非致命、致命）的多个错误。例如，意外的完成可能

也会中毒。参见[第6.2.3.2.3节f](#bookmark16)或关于报告多个错误的要求和建议。对于前面的示例，建议报告意外完成，而不报告接收到中毒TLP。

如果软件希望具有AER的代理处理通常属于咨询性非致命错误的情况， 更严重的是，软件可以将无法纠正的错误的严重性升级为致命错误，在这种情况下，代理（如果启用）将使用ERR\_FATAL发出错误信号。

本节介绍PCI Express扩展功能管理的错误的咨询非脂肪错误处理

AER。[www.example.com部分6.2.10.3](#bookmark21)介绍了支持DPC RP扩展的根端口的RP PIO错误处理机制。RPPIO咨询错误在概念上与AER咨询非致命错误相似，但适用于不同的错误情况，

由不同的公司管理。

6.2.3.2.4.1完成者发送具有UR/CA状态的完成

完成者通常发送带有不支持的请求或完成者中止（UR/CA）状态的完成，以表示未发布请求的不可纠正的错误。89如果UR/CA错误90的严重性是非致命的，则完成者必须



89.如果完成者在完成中返回数据，并且数据是坏的或可疑的，则允许完成者用信号通知错误或使用错误转发（数据中毒）机制，而不是将其作为UR或CA来处理。

90.某些其他错误（例如， ACS违规）也会导致完成者发送带有UR或CA状态的完成。如果错误的严重性（例如， ACS违规）是非致命的，完成者也必须将此情况作为建议性非致命错误处理。但是，请参见www.example.com部分2.7.2.2， 某些带有中毒数据的请求必须作为不可纠正的错误处理。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

把这个案子当作一个非致命性的错误来处理。91带有AER的完成程序通过发送ERR\_COR消息来通知非致命错误（如果启用）。没有AER的完成程序在这种情况下不发送错误消息。

即使该特定交易存在不可纠正的错误，完成人也必须将此情况作为咨询性非致命错误处理，因为完成人在收到具有UR/CA状态的完成后，

使用特定于制造商的机制报告错误（如有必要）（参见[第6.2.3.2.5节）](#bookmark23)。

6.2.3.2.4.2中间接收器

当不作为TLP的最终PCI Express目的地的接收器检测到92非致命错误时 TLP，这个“中间”接收器必须将这种情况作为咨询性非致命错误处理。93具有AER的接收器通过发送ERR\_COR消息来发出错误信号（如果启用）。没有AER的接收器在这种情况下不发送错误消息。一个

根复合体（RC）的中间接收机情况的例外被注意为低。

发生中间接收器情况的示例是交换机在其正在路由的TLP中检测到有毒或坏ECRC。尽管这是一个无法纠正的（但非致命的）错误，在这一点上，在TLP的路线，中间 接收方将其作为咨询性非致命错误处理，因此TLP的最终接收方（即，a的完成者

请求RLP或完成RLP的请求者）并不排除更适当地处理

根据其错误设置。例如，检测存储器写请求94中的有害的给定完成器可以 屏蔽错误（因此不发出信号），而同一层次结构中的不同完成程序可能会用ERR\_NONFATAL发出错误信号。

中毒TLP出口阻塞错误从不作为中间接收器情况来处理，因为它不被检测为处理接收到的TLP的一部分。

如果RC利用TLP检测到非致命错误，则它通常将在根端口之间对等转发，但是RC不支持传播错误相关信息（例如，TLP摘要、EP位或等效物），

事务，RC必须使用ERR\_NONFATAL发出错误信号（如果启用），并且不得转发事务。 一个示例是RC需要在根端口之间对等转发中毒的TLP，但是RC的内部结构不支持中毒指示。

6.2.3.2.4.3中毒TLP的终极PCI Express接收器

当中毒的TLP被其最终PCI Express目的地接收时，如果严重性是非致命的，并且接收方

如果以允许继续操作的方式处理中毒数据，则接收器必须将这种情况95作为 非致命错误。96具有AER的接收器通过发送ERR\_COR消息来发出错误信号（如果启用）。在这种情况下，没有AER的接收器不发送错误消息。请参阅www.example.com部分2.7.2.2了解适用于中毒

内存写入请求。

一个示例是接收以主机存储器为目标的中毒存储器写入TLP的根复合体。如果根复合体

将中毒数据连同其指示一起传播到主机内存，它用

ERR\_COR.如果根复合体没有将病毒传播到主机内存，则它会用ERR\_NONFATAL发出错误信号（如果启用）。

另一个示例是接收中毒的存储器读取完成TLP的解密器。如果解密器在内部传播中毒的数据或者像处理具有UR/CA状态的完成那样处理错误，则它发出错误信号（如果

使用ERR\_COR启用）。如果病毒没有以允许继续操作的方式处理病毒，它将用ERR\_NONFATAL发出错误信号（如果启用d）。

|  |  |
| --- | --- |
| 91.如果严重性是致命的，则错误不是建议性非致命错误，并且必须  92.如果接收方不实施ECRC检查或ECRC检查不  93.如果严重性是致命的，则错误不是建议性非致命错误，并且必须 | 用ERR\_F\_ATAL发出信号（如果启用）。  启用后，接收器将不会检测到ECRC错误。用ERR\_FATAL发出信号（如果启用）。 |
| 94. 请参阅www.example.com部分2.7.2.2了解适用于中毒内存写入请求的特殊规则。 | |
| 95.但是，请参阅www.example.com部分2.7.2.2了解某些必须作为不可纠正错误处理的带有中毒数据的请求。 | |
| 96.如果严重性是致命的，则错误不是建议性非致命错误，并且必须使用ERR\_FATAL发出信号（如果启用）。 | |

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.2.3.2.4.4带竣工验收的竣工验收

本节适用于执行编程I/O（PIO）的根端口以外的请求程序。有关[6.2.10.3支持DPC RP扩展的R oot Ports中](#bookmark26)的相关RP PIO功能，请参见第www.example.com节。

当一个未发布请求的发送者在等待相关的完成时超时时，发送者 允许通过发出单独的后续请求来尝试从错误中恢复。允许故障恢复器尝试恢复零次、一次或多次（有限），但如果不进行进一步的恢复尝试，则必须使用不可纠正的错误消息来通知错误（如果启用）。

如果完成故障的严重程度为非致命性，并且故障排除者选择通过发出新的

请求时，调试器必须首先将当前错误情况作为咨询性非脂肪错误处理。97A带AER的 通过发送ERR\_COR消息发出错误信号（如果启用）。在这种情况下，没有AER的路由器不会发送错误消息。

请注意，只有在Non-PostedRequest没有副作用的情况下，才有可能通过完成故障自动恢复，但也可能取决于本规范范围之外的其他考虑因素。

6.2.3.2.4.5未预期完工

当接收器接收到意外完成并且意外完成错误的严重性是非致命的时，接收器必须将这种情况作为通知非致命错误来处理。98具有AER的接收器通过发送ERR\_COR消息来发出错误信号（如果启用）。没有AER的接收器在这种情况下不发送错误消息。

如果意外的完成是由于错误路由，则在相关联的

触发器最终将触发，并且触发器可以选择尝试恢复。通过让未预期完成的接收器将错误作为咨询性非致命错误处理，可以避免对加密器恢复的干扰。

6.2.3.2.5接收具有UR/CA状态的完工的制造商

当完成者收到带有UR/CA状态的完成时，通常完成者已将错误作为建议性非致命错误处理，假设错误严重性在完成者处是非致命的（参见[第6.2.3.2.4.1节）](#bookmark22)。的

调试人员必须确定是否需要任何错误恢复操作、要采取的恢复操作类型以及是否报告错误。

如果授权者需要报告错误，则授权者必须仅通过授权者特定的机制来报告错误。例如，许多设备具有可以将错误报告给软件的关联设备驱动程序。作为另一个重要的

例如，如果配置读取完成具有UR/ CA状态，则某些平台上的根复合体将所有1返回给软件。

[6.2.10.3部分](#bookmark28)介绍了支持DPC RP扩展的根端口的RP PIO外部

机制，不允许调试器使用PCI Express日志记录和错误消息信令报告错误。

<6.2.3.3>错误转发（数据中毒）

错误转发，也称为数据中毒，通过设置TLP中的EP位来指示。参见第2.7.2节。这是PCI Express中的另一种错误报告方法，它使TLP的接收器能够将错误与特定的



97.如果严重性是致命的，则错误不是建议性非致命错误，并且必须使用ERR\_FATAL发出信号（如果启用）。强烈不建议删除者尝试恢复，因为发送ERR\_FATAL通常会导致整个层次结构下降。

98.如果严重性是致命的，则错误不是建议性非致命错误，并且必须使用ERR\_FATAL发出信号（如果启用）。

页512

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

请求或完成。与完成状态机制不同，错误转发可用于包含数据的请求或完成。此外，张力腿平台路线上的“中间”接收器，而不仅仅是 最终目的地，需要检测和报告（如果启用）接收中毒的TLP。这可以帮助软件

确定路径上的特定交换机是否使TLP中毒。

<6.2.3.4>可选错误检查

此规范包含许多可选的错误检查。除非另有说明，否则如果未执行可选的错误检查并且发生错误，则行为未定义。

当可选错误检查涉及多个规则时，除非另有说明，否则每个规则都是独立可选的。一个实现可以检查所有的规则，也可以不检查任何一个规则，或者是任何组合。

除非另有说明，否则特定于实现的标准用于确定是否执行可选的错误或检查。

错误日志

[第6.2.7节](#bookmark29)列出了本规范涵盖的所有错误，并为每个错误指定了日志记录要求。不支持高级错误报告功能的设备功能仅记录设备状态寄存器位

指示已经检测到错误。请注意，某些错误也会使用PCI兼容（类型00和01h）配置寄存器中的报告机制进行报告。第7.5.1节介绍了这些寄存器位的配置方式

受本节中描述的不同类型错误条件的

对于支持高级错误报告功能的器件功能，[表6-3、](#bookmark30)[表6-4](#bookmark31)和[表6-5](#bookmark32)中的每个错误对应于不可纠正错误状态寄存器或可纠正错误状态寄存器中的特定位

注册. 软件使用这些寄存器来更精确地确定发生的错误和严重程度。对于特定的事务层错误和不可纠正的内部错误，记录关联的TLP报头。

在多功能设备中，与设备中任何特定功能无关的PCI Express错误记录在该设备中所有功能的相应状态和记录寄存器中。

以下PCI Express错误不是特定于功能的：

·所有物理层错误

·所有数据链路层错误

·这些事务层错误：ECRC检查失败

. 不支持的请求，这是由于没有函数声明TLP而导致的。接收器溢出

. 流控制协议错误或。变形TLP

. 意外完成，当没有函数声明完成时

. 意外完成，由交换机无法转发的完成引起，且入口端口是与多功能设备关联的交换机上游端口

. 一些事务层错误（例如，接收到的中毒TLP）可以是功能特异性的或非功能特异性的，

这取决于相关联的TLP是以该设备中的单个功能还是所有功能为目标。

·一些内部错误

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

. 确定内部错误是否是特定于函数的是特定于实现的。

在检测到这些错误中的一个时，Multi-Fun智能设备应当生成最多一个给定严重性的错误报告消息，其中该消息必须报告被启用以报告的设备的功能的智能ID。

这种特殊的错误。如果没有功能被启用以发送报告消息，则设备不发送报告消息。

留言该如果所有启用报告的功能都为错误设置了相同的严重级别，则仅发送一条错误消息。如果所有启用报告的功能没有为错误设置相同的严重级别，则每个功能都有

严重性级别已发送。软件负责在检测到其中一个错误时扫描多功能器械中的所有功能。

<6.2.4.1>根复合体注意事项（高级错误报告）

6.2.4.1.1错误源识别

除了上述日志记录之外，还需要支持高级错误

需要报告功能来实现错误源ID标识寄存器，该寄存器记录根端口或根复杂事件收集器接收的第一个ERR\_NONFATAL/ERR\_FATAL（不可纠正的错误）和ERR\_COR（可纠正的错误）消息的纠正器ID。为支持高级错误报告而编写的系统软件可以使用 根错误状态寄存器，用于确定哪些字段保存有效信息。

如果RCiEP与根复合体事件收集器相关联，则RCiEP必须通过该根复合体事件收集器报告其错误。

对于根端口和根复杂事件收集器，为了将接收到的错误消息或内部生成的错误消息记录在根错误状态寄存器和错误源标识寄存器中，错误

消息必须是[6.2.8.1](#bookmark33)“传输“的，有关如何转发和传输接收到的消息的信息，请参阅第www.example.com节。内部生成的错误消息可通过SERR# Enable位进行传输

命令寄存器（ERR\_NONFATAL和ERR\_FATAL）或设备控制寄存器中的报告启用位（ERR\_COR、ERR\_NONFATAL和ERR\_FATAL）。

6.2.4.1.2数据流生成

根错误命令寄存器允许进一步控制根复合体对可纠正、非致命和致命错误消息的响应，而不是根复合体的基本功能，即根据错误消息生成系统错误。位

字段启用或禁用三种类型的错误消息的中断生成。 响应于错误消息的系统错误生成可以经由PCI Express能力结构被禁用。

如果根端口或根复合事件收集器启用了使用INTx消息的电平触发中断信号，则只要满足以下所有条件，就必须断言虚拟INTx

·命令寄存器中的禁用位设为0b。

·根错误命令寄存器中的至少一个错误报告使能位及其根错误状态寄存器中的相关错误消息接收位均设置为1b。

请注意，在请求服务时，同一功能内的所有其他中断源将断言同一虚拟INTx线。

如果使用MSI或MSI-X为边沿触发中断信令启用根端口或根复合事件收集器， 每次以下条件的逻辑AND从“否”转换为“真”时，必须发送

·相关联的向量未被掩蔽（如果MSI不支持PVM，则不适用）。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·根错误命令寄存器中的至少一个错误报告使能位及其根错误状态寄存器中的相关错误消息接收位均设置为1b。

请注意，高级错误报告MSI/MSI-X中断始终使用根错误状态寄存器中的高级错误报告消息编号字段指示的向量。

<6.2.4.2>多错误处理（高级错误报告功能）

对于高级错误报告功能，不可纠正错误状态寄存器和可纠正错误状态寄存器累积对应于该特定PCI Express接口的错误集合。位保持设置，直到

由软件明确清除或重置。由于在不可纠正错误状态寄存器中可能设置

错误指针（有效时）指向记录的最早的不可纠正的错误。 当不可纠正错误状态寄存器的相应位被设置时，第一个错误指针有效。当第一个错误指针无效

不可纠正错误状态寄存器的相应位未设置，或者是未定义位。

高级错误报告能力提供了为需要标题日志的错误记录标题99的能力。实现可能支持记录多个标头，但至少必须支持以下功能

记录至少一个。记录多个标头的能力由多标头记录能力位的状态指示，并由高级错误功能和控制的多标头记录启用位 注册. 如果支持并启用了多标题记录，则将按照检测到错误的顺序记录错误。

如果在检测到未屏蔽的不可纠正错误时没有标题记录资源可用，则其错误状态位为 设置，但不记录错误。如果一个不可纠正的错误在被检测到时被屏蔽，它的错误状态位被置位，但错误不被记录。

当软件准备消除第一错误指针指示的记录错误时，软件将1b写入

指示的错误状态位将其清除，这将导致硬件释放相关的记录资源。如果另一

该错误的实例仍然被记录，允许但不需要硬操作来保留该错误状态位设置。如果有的

错误实例仍被记录，硬件必须立即更新标头日志、TLP前缀日志、TLP前缀日志

当前位、第一个错误指针和不可纠正错误状态寄存器，以反映下一个记录的错误。如果没有记录其他错误，则建议硬件更新第一错误指针，以指示它永远不会设置的状态位，例如，a保留状态位。请参见下面的实施说明。

如果支持并启用了多个标头记录，并且第一个错误指针有效，则建议

软件不将1b写入除了第一错误指针100所指示的状态之外的任何状态b。如果软件将1b写入此类非指示位，则允许硬件清除任何相关的记录错误，但不需要这样做。

如果软件观察到第一个错误指针无效，并且软件希望清除由于先前的头记录资源溢出而设置的任何未屏蔽状态位，则软件应了解以下竞争

条件如果在软件清除这些状态位之前碰巧记录了这些错误的任何新情况，则一个或多个新记录的错误可能丢失。

如果支持并启用了多个标头记录，则软件在清除多个标头记录时必须特别小心。

标题记录使能位。如果软件在第一个错误指针有效时清除该位，则硬件行为未定义。在清除多标题记录使能位之前，建议软件暂时屏蔽所有无法纠正的错误，然后重复消除第一个错误指针指示的每个错误。

由于一个实现只能记录有限数量的报头，因此软件必须及时地为第一个错误指针、报头日志和TLP前缀日志寄存器提供服务，以限制丢失这些报头的风险

后续错误的信息当检测到需要头日志记录的错误时，会发生头日志溢出



99.如果函数支持TLP前缀，则其AER能力还会记录任何随附的TLP前缀以及每个记录的标头。对标头记录的引用也意味着TLP前缀记录。

100. 屏蔽错误的状态位是一个例外。 如果软件确定它们没有记录的标头，软件可以安全地清除它们，就像如果它们自第一个错误指针上次无效以来一直被屏蔽的情况一样。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

或者已达到实现所支持的记录头数，或者多头记录启用位未置位且第一个错误指针有效。

实现可以选择检查此条件并报告头日志溢出错误。这是与检测功能相关的报告错误。

多标题记录能力的设置和标题日志溢出的检查是独立可选的。



**执行说明**

第一个错误指针寄存器有效

第一个错误指针（FEP）字段定义为在不可纠正错误状态寄存器的相应位置1时有效。为避免某些情况下的歧义，建议如下：

·在记录了不可纠正的错误之后，当不可纠正的错误中的相关联的位被 状态寄存器通过软件向其写入1b来清除，硬件应更新FEP以指向它永远不会设置的状态位，例如，a保留状态位。(This假设函数不 我已经有另一个记录的错误要报告，如果它支持多个标题，可能就是这种情况

录音。）

·FEP的默认值应该指向硬件永远不会设置的状态位，例如，a保留状态位。

下面是一个不遵循上述建议的不支持请求（UR）的模糊性

· 当系统固件进行其配置空间探测时，UR和建议非致命错误未被屏蔽。

·函数由于正常探测而遇到UR，记录它，并将FEP设置为指向UR。

·系统固件清除UR状态位，硬件保留指向UR的FEP。

·操作系统启动后

· 正常探测设置UR状态位，但由于UR被屏蔽，因此不记录错误。

在这一点上，存在FEP指向设置的状态位（因此有效）的模糊性，而实际上，没有需要由软件处理的记录错误。

如果硬件依赖于FEP的此定义来确定何时

资源依赖于其他内部状态来确定何时可以记录新错误的硬件实现可能不会有此问题;但是，硬件实现仍应遵循上述要求

建议避免将这种模糊性呈现给软件。

<6.2.4.3>咨询性非致命错误或日志记录

[第6.2.3.2.4节](#bookmark13)描述了咨询性非致命错误情况，在这种情况下，具有AER的座席

非致命严重性的错误使用ERR\_COR而不是ERR\_NONFATAL发出错误信号（如果启用）。对于相同的情况，没有AER的代理不发送错误消息。本节剩下的讨论是在代理的上下文

实施AER。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

对于咨询非致命错误情况，由于使用可纠正错误消息来发信号通知不可纠正错误，因此适用涉及不可纠正错误和可纠正错误的控制/状态/掩码位。[图6-2所](#bookmark35)示为

顺序以下是记录数据库非致命错误的一些独特方面。

首先，不可纠正错误的严重性必须是非致命的，这由不可纠正错误严重性寄存器中的相关位确定。如果严重性是致命的，则错误不符合“建议性非致命错误”的条件，

如果启用，则以ERR\_FATAL发出信号。

接下来，特定错误案例需要是第6.2.3.2.4节中记录的咨询性非致命错误案例[之一。](#bookmark20)如果不是，则错误不符合咨询性非致命错误的条件，并且将使用不可纠正的错误消息进行签名（如果启用）。

接下来，在可纠正错误状态寄存器中设置咨询非致命错误状态位，以指示

检查可纠正错误掩码寄存器中的咨询错误和咨询非致命错误掩码位，如果设置，则不进行进一步处理。

如果咨询非致命错误掩码位被清除，则根据报告为咨询错误的特定不可纠正错误，通过设置不可纠正错误状态寄存器中的“相应”位来进行日志记录。如果

“corresponding” uncorrectable

需要头日志记录，然后记录前缀和头，取决于资源的可用性。 参见www.example.com[部分6.2.4.2。](#bookmark34)

最后，如果在设备控制寄存器中设置了可纠正错误报告启用位，则会发送ERR\_COR消息。

<6.2.4.4>TLP前缀日志

对于支持TLP前缀和高级错误报告两者的任何设备功能，根据与报头日志寄存器相同的规则，将与出错的TLP相关联的TLP前缀记录在TLP前缀日志寄存器中（使得当第一错误指针有效时，TLP前缀日志和报头日志寄存器两者总是对应于第一错误指针中指示的错误）。

TLP前缀日志存在位（参见www.example.com节7.8.4.7）表示TLP前缀日志寄存器（参见www.example.com节7.8.4.12）包含信息。

AER仅记录端-端TLP前缀。本地TLP前缀的日志记录可以使用前缀特定机制在其他地方发生。101

端-端TLP前缀记录在TLP前缀日志寄存器中。底层TLP报头记录在报头日志寄存器中，但有两种例外情况：

· 如果设置了扩展Fmt字段支持端口位（参见7.5.3.15部分），则不支持TLP

前缀和接收包含TLP Prefix的TLP将用信号通知格式错误的TLP，并且报头日志寄存器将包含TLP的前四个DW（TLP前缀后面是适合的TLP报头）。

·接收到包含比由功能的M\_ax\_End-End\_TLP前缀字段指示的更多的端-端TLP前缀的TLP的功能2.2.10.2

寄存器未定义

6.2.5设备错误标记和记录操作

[图6-2](#bookmark38)显示了与设备检测到的错误的信号和日志记录相关的操作序列。



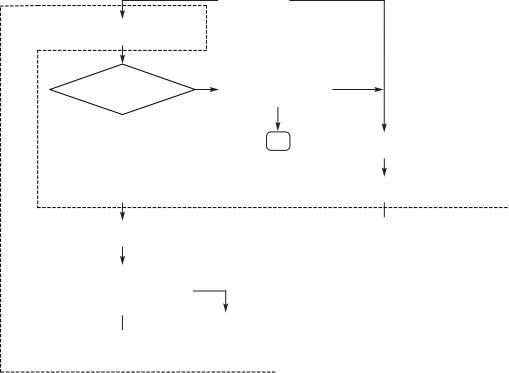
101.例如，涉及MRI-IOVTLP前缀的错误将记录在MR-IOV结构中，而不会记录在AER功能中。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

检测到表6-2、6-3、6-4或6-5



错误类型？

不可校正 可校正 缩略语：

reg

寄存器

Cmd

命令寄存器SERR#使能

设备控制寄存器不支持的请求 报告启用

致命错误报告启用

非致命错误

报告启用可更正错误报告启用

|  |
| --- |
| 根据无法纠正的错误严重性法规确定严重性 |

SERR\_ EnDCR

URREFERENERECERE

咨询

非致命错误？（第6.2.3.2.4节）

是的 是否实施？ 是的

没有

没有

y

|  |
| --- |
| 在设备状态寄存器中设置致命/非致命错误检测位 |

|  |
| --- |
| 在设备状态寄存器中设置可纠正的错误检测位 |

|  |
| --- |
| 如果为UR，则在设备状态寄存器中设置检测到不支持的请求位 |

|  |
| --- |
| 如果为UR，则在设备状态寄存器中设置检测到不支持的请求位 |

先进

|  |
| --- |
| 设置相应位  无法纠正的错误状态注册表 |

|  |
| --- |
| 设置相应位 可更正的错误状态注册表 |

误差

报告

只

在C？误差 是的

蒙面

无法纠正的错误

面具登记？

没有

是的

我不 端

端

（错误为UR）

|  |
| --- |
| 适当时，记录前缀和  header和update prefix，  标题报告字段和标题 |

V

<>

|  |
| --- |
| 如果出现建议性非致命错误：（1）设置  不可纠正中的对应位  错误状态reg，以及（2）如果  在不可纠正的错误中未屏蔽  面罩注册（如适用）  前缀和头和update前缀  和标题报告字段， |

是的

AND（ DCR： URRE=0）

AND（Cmd：SERR\_En =0）？

我不

<>

没有

端

是的

（错误为UR）和

(DCR：URRE=0）？

DPC触发器是否启用？

<>

没有

是的

端

V

<>我

没有

(DCR：CERE=1）？

端

|  |
| --- |
| 设置DPC触发状态DPC触发原因 |

致命 非致死

是的

严重性？



端

|  |
| --- |
| 发送ERR\_COR |

（Cmd： SERR\_ En=1）

（Cmd： SERR\_ En=1）

或

(DCR：FERE=1）？

没有

没有

或

（DCR：NERE=1）？

V

端

V

端

端

是的

是的

发送错误\_致命

|  |
| --- |
| 发送错误\_非致命 |

端

OM14546E

端

图6-2设备错误信号和日志操作顺序流程图



第518

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

错误消息控制

错误消息有一组复杂的相关控制位和状态位。[图6-3](#bookmark40)以伪逻辑图的形式提供了错误消息如何生成、记录、转发并最终通知系统的高级摘要。并非所有的控制位和状态位都显示出来。图中所示的对数ic门用于传送

一般概念，而不是直接实施。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | 系统特定  错误代码（INTx或MSI）   |  |  | | --- | --- | | Q  X | AER整合（可选AER的RP/RCEC部分）  启用AER报告（每个错误类一个）  根错误命令寄存器  根错误状态寄存器 |  |  | | --- | | 误差  消息接收 |   Q  DPC未包含  如果ERR\_FATAL或ERR\_NONFATAL，则设置  触发DPC  DPC状态寄存器   |  | | --- | | DPC触发状态DPC触发原因 |   SQ  SERR#使能（用于转发）控制寄存器  仅在ERR\_FATAL或ERR\_NONFATAL时设置  辅助状态寄存器  桥    系统错误 （具体平台）   |  |  |  | | --- | --- | --- | |  | 系统错误启用（每个错误类一个） 根控制寄存器 |  |   错误\_\* 消息  仅在ERR\_NONFATAL或ERR\_FATAL状态寄存器   |  | | --- | | 信号通知  系统  误差 |   启用错误报告（每个错误类一个）  Q  QA  SERR#使能（用于传输）命令寄存器  仅限ERR\_NONFATAL和  设备控制寄存器  ERR\_COR，  ERR\_NONFATAL和ERR\_FATAL  内部生成的ERR\_\*  消息   |  | | --- | | 接收  系统  误差 |   接收  错误\_\*  消息  A-0479B |

图6-3所选错误消息控制和状态位的伪逻辑图



第519

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.2.7错误列表和规则

[表6-2](#bookmark41)至[表6-4](#bookmark42)列出了本规范定义的所有PCI Express错误。列出了每个错误

使用缩写名称，在硬件中如何检测错误，错误的默认严重性，以及检测错误的代理所采取的预期操作。这些操作构成了PCI Express错误报告和日志记录的规则。

默认严重性列指定错误的默认严重性，无需任何软件重新编程。对于支持高级错误报告功能的设备功能，不可纠正的错误可编程为致命错误或 错误严重性寄存器为非致命。没有高级错误报告功能的设备功能使用默认关联，并且不可重新编程。

如果错误触发DPC，则实现下游端口控制（DPC）并启用该控制的下游端口的检测代理操作将有所不同。下表中未对DPC进行描述。 参见[第](#bookmark43)[6.2.10f节](#bookmark44)或DPC行为的描述。

表6-2一般PCI Express错误列表

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型  （默认严重性） | 检测代理操作102 | 引用 |
| 更正内部错误 | 可校正  （默认为屏蔽） | 组件：  将ERR\_COR发送到根复合体。 | [第](#bookmark45)[6.2.9](#bookmark46) |
| 无法纠正的内部错误 | 不可校正  （默认情况下为致命和屏蔽 | 组件：将ERR\_FATAL发送到根复合体。  也可以记录与错误关联的第一个TLP的前缀/标头。 | [第](#bookmark47)[6.2.9](#bookmark48) |
| 标题日志溢出 | 可校正  （默认为屏蔽） | 组件：  将ERR\_COR发送到根复合体。 | [6.2.4.2部分](#bookmark34) |

表6-3物理层错误列表

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型（默认严重性） | 检测代理操作103 | 引用 |
| 接收器错误 | 可校正 | 接收方：  将ERR\_COR发送到根复合体。 | 第4.2.1.1.3节4.2.1.2 4.2.4.8部分 第4.2.6 |

表6-4数据链路层错误列表

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型  （默认严重性） | 检测代理操作104 | 引用 |
| 不良TLP | 可校正 | 接收方：  将ERR\_COR发送到根复合体。 | 3.6.3.1部分 |
| 不良DLLP | 接收方：  将ERR\_COR发送到根复合体。 | 3.6.2.2部分 |



102.对于这些表，检测代理操作就像所有启用位都设置为“启用”一样，对于高级错误处理，掩码位被禁用，严重性位被设置为默认值。 必须根据这些位的实际设置修改操作。

103.对于这些表，检测代理操作就像所有启用位都设置为“启用”一样，对于高级错误处理，掩码位被禁用，严重性位被设置为默认值。 必须根据这些位的实际设置修改操作。

104.对于这些表，检测代理操作就像所有启用位都设置为“启用”一样，对于高级错误处理，掩码位被禁用，严重性位被设置为默认值。 必须根据这些位的实际设置修改操作。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型  （默认严重性） | 检测代理操作 | 引用 |
| 重播计时器 |  | 发送器：  将ERR\_COR发送到根复合体。 | 3.6.2.1部分 |
| RESIDENT\_NUM滚动 | 发送器：  将ERR\_COR发送到根复合体。 | 3.6.2.1部分 |
| 数据链路协议错误 | 无法纠正（致命） | 如果选中，请将ERR\_FATAL发送给Root Compl ex。 | 3.6.2.2部分 |
| 惊喜下载 | 如果选中，则将ERR\_FATAL发送到根Compl ex。 | 第3.2.1 |

表6-5事务层错误列表

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型（默认 严重程度） | 检测代理操作105 | 引用 |
| 中毒TLP  接收 | 无法纠正（非致命） | 接收方：  将ERR\_NONFATAL发送到根目录 复杂或ERR\_COR，用于 第[6.2.3.2.4.1节](#bookmark22)和[第6.2.3.2.4.2节中描述的咨询性非致命错误病例。](#bookmark24)  记录Poisoned TLP的前缀/头。106 | 2.7.2.2部分 |
| 中毒 TLP出口阻塞 | 下游端口变送器：  将ERR\_NONFATAL发送到根复合体。  记录中毒TLP的前缀/标头。 | 2.7.2.2部分 |
| ECRC检查失败 | 接收方（如果支持ECRC检查  将ERR\_NONFATAL发送到根目录 复杂或ERR\_COR，用于 咨询性非致命错误案例 如[第6.2.3.2.4.1节](#bookmark22)和[第6.2.3.2.4.2节所述。](#bookmark24)  记录遇到ECRC的TLP的前缀/报头  错误. | 第2.7.1 |
| 不支持的请求（UR） | 无法纠正（非致命） | 请求接收方：  将ERR\_NONFATAL发送到根复合体或将ERR\_COR发送到  咨询性非致命错误案例 如[第6.2.3.2.4.1节所述。](#bookmark22) 记录导致错误的TLP的前缀/标头。 | 表F-1，第2.3.1节，第2.3.2节，第2.7.2.2节，  第2.9.1节、第5.3.1节、[第6.2.3.1节、](#bookmark10)[第6.2.6节、](#bookmark39)[第6.2.8.1节、](#bookmark49)[第6.5.7节、](#bookmark50)第7.3.1节、第7.3.3节、第7.5.1.1.3节、第7.5.1.1.4节 |



105.对于这些表，检测代理操作就像所有启用位都设置为“启用”一样，对于高级错误处理，掩码位被禁用，严重性位被设置为默认值。 必须根据这些位的实际设置修改操作。

106. 仅限高级错误处理。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型（默认 严重程度） | 检测代理操作 | 引用 |
| 完成时间 | 无法纠正（非致命） | 翻译：  将ERR\_NONFATAL发送到根复合体或将ERR\_COR发送到  咨询性非致命错误案例 如[第6.2.3.2.4.4节所述。](#bookmark25)  如果完成  在高级错误中设置前缀/标头日志功能位  能力和控制  注册，记录前缀/标题  请求TLP， 遇到了错误。 | 第2.8 |
| 完成者中止 | 完成人：  将ERR\_NONFATAL发送到根复合体或将ERR\_COR发送到  咨询性非致命错误案例 如[第6.2.3.2.4.1节所述。](#bookmark22)  记录 遇到错误的请求。 | 第2.3.1 |
| 意外完成 | 接收方：  将ERR\_COR发送到根  复杂. 这是一个咨询 第6.2.3.2.4.5节中描述的非致命错误案例[。](#bookmark27)  记录 遇到错误的完成。 | 第2.3.2 |
| ACS  违反 | 接收方（如果勾选）：  将ERR\_NONFATAL发送到根复合体或将ERR\_COR发送到  咨询性非致命错误案例 如[第6.2.3.2.4.1节所述。](#bookmark22)  记录 请求导致错误的TLP。 |  |
| [MC封闭](#bookmark51)[式TLP](#bookmark52) | 接收方（如果勾选）：  将ERR\_NONFATAL发送到根复合体。  记录 请求导致错误的TLP。 | [第6.14.4](#bookmark53) |



第522

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |  |  |  |
| --- | --- | --- | --- |
| 错误名称 | 错误类型（默认 严重程度） | 检测代理操作 | 引用 |
| AtomicOp出口  封锁 | 无法纠正（非致命） | 出口端口：  将ERR\_COR发送到根  复杂. 这是一个咨询 第6.2.3.2.4.1节中描述的非致命错误案例[。](#bookmark22)  记录AtomicOp请求的前缀/头，  遇到了错误。 | 第6.15.2 |
| TLP前缀已阻止 | 出口端口：  将ERR\_NONFATAL发送到根复合体或将ERR\_COR发送到  咨询性非致命错误案例 如[第6.2.3.2.4.1节所述。](#bookmark22) 记录遇到错误的TLP的前缀/标头。 | 2.2.10.2部分 |
| 接收器溢出 | 无法纠正（致命） | 接收方（如果勾选）：  将ERR\_FATAL发送到根复合体。 | 2.6.1.2部分 |
| 流控制协议  误差 | 接收方（如果勾选）：  将ERR\_FATAL发送到根复合体。 | 第2.6.1 |
| 变形TLP | 接收方：  将ERR\_FATAL发送到根复合体。  记录遇到错误的T LP的前缀/标头。 | 第2.2.2节，第2.2.3节，第2.2节。5，第2.2.7节，  第www.example.com节2.2.8.1，第2.2.8.2节2.2.8.3，第2.2.8.4节，第2.2.8.5，第2.2.8.10节，第2.2.10节，第  2.2.10.1，第2.3节，第2.3.1节，第2.2.10.22.3.1.1，第2.3.2节，第2.5节，第2.5.3节，第2.6.1节，第2.6.1.2[6.2.4.4，](#bookmark36)[第6.3.2](#bookmark54)节 |

对于上面列出的所有错误，必须在检测到错误时设置相应的状态位。对于不支持的请求（UR），应用附加的检测和报告使能位（参见[第6.2.5节）](#bookmark37)。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

器械UR报告与旧版和1.0a软件的

如果1.0a设备函数不实现基于角色的错误报告，则107表示不支持的请求

设备控制寄存器中的报告使能位清除时，防止功能发送任何错误

通知UR错误的消息。利用基于角色的错误报告功能，如果命令寄存器中的SERR#启用位被设置，则该功能被隐式地启用108以发送ERR\_NONFATAL或ERR\_FATAL消息来用信号通知UR 错误，即使不支持的请求报告启用位已清除。这会引发向后兼容性

关注为1.0A设备编写的软件（或固件）

如果软件/固件设置了SERR#启用位，但未清除不支持请求报告启用和可纠正错误报告启用位，则基于角色的错误报告功能遇到UR错误 如果请求未提交，则不会发送错误消息，如果请求未提交，则会用ERR\_NONFATAL发出错误信号。

请求已发布。非发布请求的行为支持PC兼容的配置空间探测，而发布请求的行为恢复了与PCI和PCI-X的错误报告兼容性，避免了该区域中静默数据损坏的可能性。因此，基于角色的错误报告设备 与预想的传统和1.0a软件和固件向后兼容。



<6.2.7.1>常规PCI映射

为了支持传统的PCI驱动程序和软件兼容性，PCI Express错误条件，在适当的情况下，必须映射到PCI状态寄存器位，用于错误报告。

换句话说，当检测到某些PCI Express错误时，适当的PCI状态寄存器位被设置， 旧版PCI软件错误。 当PCI Express错误导致设置PCI状态寄存器时，清除PCI状态寄存器不会导致清除不可纠正错误状态寄存器和可纠正错误状态寄存器中的位。

同样，清除不可纠正错误状态寄存器和可纠正错误状态寄存器中的位不会导致清除PCI状态寄存器。

PCI命令寄存器具有控制PCI错误报告的位。但是，PCI命令寄存器不影响PCI EXpress错误寄存器位的设置。

6.2.8虚拟PCI桥接错误处理

虚拟PCI桥配置报头与根复合体或交换机中的每个PCI Express端口相关联。对于这些情况，PCI Express错误概念需要适当映射到PCI错误报告结构。

<6.2.8.1>错误消息转发和PCI映射桥-规则

通常，TLP或者从虚拟PCI桥的一侧传递到另一侧，或者根据应用于TLP的最终接收者的相同规则在桥的入口侧处理。以下规则涵盖PCI

表达特定错误相关的情况。参见[第6.2.6f节](#bookmark39)或错误消息控制的概念性总结。



107. 如设备功能注册表中的基于角色的错误报告位所示。 参见第7.8.3节。

108.假设如果设备实现AER，则在不可纠正错误掩码寄存器中未设置不支持请求错误掩码位。

第五百二十四

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

· 如果请求没有寻址映射到网桥的内部空间或网桥的出口侧的空间，则请求在入口侧作为不支持的请求终止

· 中毒TLP按照与非中毒TLP相同的规则转发。当向下游转发中毒请求

▪ 设置状态寄存器中的检测到奇偶校验错误位

▪ 如果设置了桥控制寄存器中的奇偶校验错误响应使能位，则设置辅助状态寄存器中的主数据奇偶校验错误位

. 当向下游转发中毒

▪ 设置状态寄存器中的检测到奇偶校验错误位

▪ 如果设置了命令寄存器中的奇偶校验错误响应位，则设置状态寄存器中的主数据奇偶校验错误位

. 当向上游转发中毒请求

▪ 设置辅助状态寄存器中的检测到奇偶校验错误位

▪ 如果设置了命令寄存器中的奇偶校验错误响应位，则设置状态寄存器中的主数据奇偶校验错误位

. 当向上游转发中毒并发症

▪ 设置辅助状态寄存器中的检测到奇偶校验错误位

▪ 如果设置了桥控制寄存器中的奇偶校验错误响应使能位，则设置辅助状态寄存器中的主数据奇偶校验错误位

·ERR\_COR、ERR\_NONFATAL和ERR\_FATAL从辅助接口转发到主接口

如果桥接控制寄存器中的SERR#使能位置1，则在接口上。 转发错误消息的网桥不得在设备状态寄存器中设置相应的错误检测位。由主接口转发的错误消息的传输由多个位控制，如[图6-3所示。](#bookmark40)

· 对于根端口，从辅助接口转发到主接口的错误消息必须

通过主接口启用

寄存器或（当高级错误报告功能存在时）通过根错误命令寄存器报告并记录在根错误状态寄存器和错误源标识寄存器中。

· 对于根复合体事件收集器（技术上不是网桥），从

必须启用RCiEP以通过根控制寄存器进行

（当高级错误报告上限功能存在时）通过根错误命令寄存器报告并记录在根错误状态寄存器和错误源标识寄存器中。

6.2.9内部错误

内部错误是与组件内发生的PCI Express接口相关的错误，可能不是由于PCI Express接口本身上的数据包或事件，也不是由于PCI上启动的事务

快车 确定什么是内部错误是特定于实现的，不在本规范的范围内。

内部错误可以分为已纠正的内部错误和无法纠正的内部错误。已纠正的内部错误是组件中发生的错误，该错误已被硬件掩盖或解决，而没有任何损失

信息或操作不当。一个可能的已纠正内部错误的示例是内部数据包缓冲器

由纠错码（ECC）纠正的内存错误。不可纠正的内部错误是指在组件内发生的错误，导致组件的不正确操作。一个可能无法纠正的内部

错误是ECC无法纠正的内存错误。 从不可纠正的内部错误中恢复的唯一方法是重置或更换硬件。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

报告已纠正的内部错误和不可纠正的内部错误是独立可选的。如果报告了任何一种情况，则必须执行AER。

对于无法纠正的内部错误，标题日志记录是可选的。当记录头时，头是第一个丢失或被不可纠正的内部错误损坏的TLP的头。当没有实现头日志记录或头r不可用时，记录全1的头。

与特定PCI Express接口相关的内部错误由与该端口相关的功能报告。在交换机中检测到的无法与特定PCI Express接口关联的内部错误由上游端口报告。报告无法与特定PCI Express关联的内部错误 所有其他多端口组件中的接口（例如，根复合体）在本说明书的范围之外。

6.2.10下游港口安全壳（DPC）

下游端口密封（DPC）是下游端口的可选规范特征。DPC暂停PCI Express

在下游端口或下游端口以下检测到未屏蔽的不可纠正错误后， 任何数据损坏的潜在传播，并支持遏制错误恢复（CER）（如果由软件实现）。下游端口通过实现DPC扩展能力结构来表示对DPC的支持，该结构包含所有DPC控制和状态位。 参见第7.9.15节。

DPC默认情况下处于禁用状态，除非由软件使用DPC触发启用字段启用，否则无法触发。当DPC触发启用字段设置为01b时，DPC被启用，并在下游端口检测到

未屏蔽的不可纠正错误或当下游端口收到一个错误\_致命消息。当DPC触发启用字段被设置为10b时，DPC被启用，并且当下游端口检测到未屏蔽

无法纠正的错误或当下游端口收到ERR\_NONFATAL或ERR\_FATAL消息时。除了 对于由PCI Express扩展能力和高级错误报告（AER）管理的类型的不可纠正的错误，RP PIO错误可以作为不可纠正的错误来处理。 参见www.example.com[部分6.2.10.3。](#bookmark55) 还有一种机械主义，

[第6.2.10.4f节](#bookmark56)或触发DPC的软件或固件。

当DPC由于接收到不可纠正的错误消息而被删除时，来自该消息的删除器ID被记录在DPC错误源ID寄存器中，并且该消息被丢弃并且不被向上游转发。当DPC由未屏蔽的不可纠正错误触发时，即使在以下情况下，也不会用不可纠正错误消息来通知该错误：

否则启用。但是，当DPC被触发时，DPC可以发出一个中断信号或发送一个ERR\_COR消息（如果启用）。 参见www.example.com[部分6.2.10.1](#bookmark57)和[6.2.10.2部分。](#bookmark58)

当DPC被触发时，下游端口立即设置DPC触发状态位和DPC触发原因字段，以指示触发条件（未屏蔽的不可纠正错误、ERR\_NONFATAL、ERR\_FATAL、RP\_PIO错误或软件触发），并通过将LTSSM引导至禁用状态来禁用其链路。一旦LTSSM达到禁用状态，它将保持该状态，直到DPC触发状态位被清除。为了确保LTSSM有时间到达

禁用状态或至少在各种错误条件下关闭链路，软件必须将下游端口保留在DPC中，直到链路状态寄存器中的数据链路层链路活动位读取为0 b;否则，结果为

未定义。参见7.5.3.8部分。 有关DPC期间事务层行为的其他重要详细信息，请参见Se 2.9.3。

在已经在支持DPC的RP扩展的根端口中触发DPC之后，根端口可能需要一些时间来停顿和清理其内部活动，诸如与DMA读取请求相关联的那些。当DPC触发器

状态位被设置且DPC RP忙碌位被设置，软件必须将根端口保留在DPC中，直到DPC RP忙碌位读取为0b。

在软件从DPC释放下游端口之后，端口 Link将尝试重新训练。软件可以使用数据链路层状态更改中断、DL\_Active ERR\_COR信令或两者，以在链路再次达到DL\_Active状态时发出信号。 参见[6.7.3.3](#bookmark59)和www.example.com[部分6.2.10.5。](#bookmark60)

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

所有1的数据值

许多平台，包括支持DPC的RP扩展的平台，可以在错误与PCI Express配置、I/O或内存读取请求相关联时向软件返回全1的数据值。在DPC期间，

下游端口丢弃发往链路的请求并以错误完成它们（即，无论是与一个

不支持的请求（UR）或完成程序中止（CA）完成状态）。通过结束一系列的MMIO或

当配置空间操作具有对具有不等于全1的已知数据值的地址的读取时，软件可以确定完成器是否已被移除或DPC是否已被触发。

另请参见实施说明



执行说明

在DPC期间选择未发布的请求响应

DPC完成控制位决定下游端口如何响应非发布请求（NPR） 在DPC期间收到。 选择需要考虑平台的其余部分如何处理遏制错误恢复（CER）。

虽然平台中的特定CER策略细节不在本规范的范围内，但以下是基于一般考虑的一些指导原则。

如果平台或驱动程序不支持CER策略，建议选择UR完成，这是设备不存在时的标准行为。

如果CER策略依赖于软件通过查找PIO读取返回的所有1来

读取返回UR完成。对于以配置空间、内存空间以及可能的I/OSpace为目标的PIO读取，需要发生全1的合成。

如果CER策略使用的机制对PIO读取的UR和CA完成进行了不同的处理，则CA完成可能是更合适的选择。从PCIe设备返回的CA完成通常指示设备编程模型违规，这可能需要触发端口遏制和错误恢复。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

选择DPC触发条件

非致命错误是指无法纠正的错误，表明特定的TLP是不可靠的，一般来说，

关联功能不应继续其正常操作。致命错误是无法纠正的错误， 特定链路及其相关硬件是不可靠的，一般而言，该链路下的整个硬件 不能继续正常运行。非致命错误和致命错误之间的这种区分以及根端口错误遏制能力有时可以用来选择适当的DPC触发条件。的 以下假设在设备之间没有对等业务。

一些RC实现了将被统称为“功能级包含”（FLC）的专有特征。这不是PCI Express的架构特性。实现FLC的根端口能够包含

在与特定功能相关的流量中检测到非致命错误时开关

具有FLC的根端口下方的下游端口应配置为在下游端口

当下游端口接收到ERR\_FATAL消息时，检测到未屏蔽的不可纠正错误本身。在此模式下，交换机下游端口传递它接收到的上游ERR\_NONFATAL消息，

触发DPC。这使根端口FLC能够处理使特定功能不可靠的 和交换机下游端口DPC来处理使层次结构域的子树不可靠的错误。下游端口仍然需要为它检测到的所有未屏蔽的不可纠正错误触发DPC，因为

它生成的ERR\_NONFATAL将有自己的路由器ID，根端口中的FLC硬件将无法确定交换机下行端口下的哪个特定功能负责非致命错误。

根端口以下的交换机下游端口（无FLC）应配置为在交换机

下游端口检测到未屏蔽的无法纠正的错误，或者交换机下游端口收到

ERR\_NONFATAL或ERR\_FATAL消息。这使得DPC能够将错误包含到链接下面的受影响层次结构，并允许层次结构域的未受影响部分继续正常操作。



执行说明

软件轮询DPC RP忙位

DPC RP Busy位是硬件向软件指示RP需要保持在DPC中的一种

当RP进行一些内部清理和静止活动时，虽然这些细节

活动是特定于实施的，活动通常会在几微秒或更短的时间内完成。然而，在最坏的情况下，例如可能发生某些内部错误的情况，

系统，繁忙时段可能会大大延长，可能延长到几秒钟。如果软件在当前的软件环境中不能容忍如此长的延迟，那么软件可能需要依赖于使用定时器 中断来调度中断下的轮询。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

DPC控制的确定

DPC可以在一些配置中由平台固件控制，而在其他配置中由操作系统控制。DPC功能与高级错误报告中的功能密切相关。为了避免平台固件或操作系统是否控制DPC的冲突，建议平台 固件和操作系统总是将DPC的控制链接到高级错误报告的控制。



<6.2.10.1>DPC中断

支持DPC的下游端口必须支持DPC中断的生成。DPC中断由DPC控制寄存器中的DPC中断使能位使能。DPC中断由DPC状态寄存器中的DPC中断状态位指示。

如果使用INTx消息使能端口的电平触发中断信号，则只要满足以下条件，就必须断言虚拟INTx

·命令寄存器中的禁用位的值为0b。

·DPC启用位的值

·DPC故障状态位的值

请注意，在请求服务时，同一功能内的所有其他中断源将断言同一虚拟INTx线。

如果使用MSI或MSI-X使能端口的边沿触发中断信号，则每次以下条件的逻辑AND从“0”转换为“真”时，必须发送中断消息

·相关联的向量未被掩蔽（如果MSI不支持PVM，则不适用）。

·DPC启用位的值

·DPC故障状态位的值

如果中断生成已被禁用，则端口可以可选地发送中断消息，并且当中断生成随后被启用时，上述条件的逻辑AND为TRUE。

中断消息将使用DPC能力寄存器中的DPC中断消息编号字段指示的向量。 此向量可能与此函数中其他中断源使用的向量相同，也可能不同。

<6.2.10.2>DPCERR\_COR信令

具有DPC功能的下游端口必须支持ERR\_COR信令，而不管它是否支持高级错误报告（AER）。DPC ERR\_COR信令由DPC控制寄存器中的DPC ERR\_COR使能位使能。 DPC触发由DPC状态寄存器中的DPC触发状态位指示。DPC ERR\_COR信令的管理独立于DPC中断，并且允许并发使用这两种机制。

如果设置了DPC ERR\_COR使能位，并且设置了设备控制寄存器中的可纠正错误报告使能位或DPC控制寄存器中的DPCSIG\_SFW使能位，则每次DPC 触发器状态位从清除转换为设置。DPC ERR\_COR信令不得设置

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

设备状态寄存器，因为此事件不作为错误处理。如果下游端口支持ERR\_COR

子类功能，此DPC ERR\_COR信令事件必须设置DPC状态寄存器中的DPCSIG\_SFW状态位，还必须设置ER R\_COR消息中的ERR\_COR子类字段，以指示ECSSIG\_SFW。

对于给定的DPC触发事件，如果端口将发送ERR\_COR消息和MSI/MSI-X事务，则端口必须在发送MSI/MSI-X事务之前发送ERR\_COR消息。没有相应

如果INTx机制被用于发送DPC中断信号，则这是不必要的，因为INTx消息在通过路由元素时不一定保持相对于ERR\_COR消息的顺序。

执行说明

DPC ERR\_COR信令的使用

建议操作系统在DPC被触发时使用DPC中断来发出信号。而 DPC ERR\_COR信令指示相同的事件，DPC ERR\_COR信令主要供系统固件使用，当需要通知系统固件以便自己记录事件或首先提供固件时

服务



<6.2.10.3>根端口编程I/O（RPPIO）错误控制

RP PIO错误控制寄存器可以对未发布的请求被

根端口跟踪遇到某些无法纠正或建议错误。 有关跟踪哪些未发布请求的说明，请参见第2.9.3节。存在一组控制和状态位，用于接收不支持的

请求状态（URCpl）、接收完成并终止状态（CACpl）和完成终止（CTO）

错误.对于配置请求、I/O请求和内存请求，存在这些错误位的独立集合。这

更精细的粒度可以更精确地处理不可纠正错误的子集（URCpl、CA Cpl和CTO）。作为一个关键的例子，UR Cpl错误与内存读取请求可以配置为触发DPC进行适当的遏制， 错误处理，而带有配置请求的UR Cpl错误可以配置为返回全1（不触发DP C）以进行正常探测和枚举。

在AER中记录的UR或CA错误是根端口以完成者的角色操作的结果，并且对于接收到的

未发布的请求，返回完成。相反，记录为RP PIO错误的UR Cpl或CACpl错误是根端口以转发器的角色运行的结果，对于未完成的未发布请求，接收

建成在AER和RP PIO中记录的CTO错误是根端口以

但RP PIO错误控制支持每空间粒度。根据控制寄存器设置，CTO错误可以记录在AER寄存器、RP PIO寄存器或两者中。如果软件在RP PIO中发现CTO错误，

建议软件屏蔽AER中的CTO错误，以避免意外交互。

RP PIO报头日志寄存器、RP PIO ImpSpec日志寄存器和RP PIOTLP前缀日志寄存器指的是

统称为RP PIO日志寄存器。必须实现RP PIO报头日志寄存器;RP PIO ImpSpec日志寄存器和RP PIOTLP前缀日志寄存器是可选的。RP PIO日志大小字段指示

分配给RP PIO日志寄存器，并且由此可以计算RP PIOTLP前缀日志寄存器的分配大小。 参见www.example.com部分7.9.15.2。RP PIO日志寄存器始终记录来自PIO请求的信息 相关完成。

RP PIOStatus、Mask和Severity寄存器的行为与AER中的Uncorrectable ErrorStatus、Mask和Severity寄存器类似。请参见www.example.com部分7.8.4.2、www.example.com7.8.4.3部分和7.8.4.4>部分。当RP PIO错误在未屏蔽时被检测到时，RP PIO状态寄存器中的相关位被置位，并且错误被记录在RP PIO日志寄存器

（假设RP PIO错误日志记录资源可用）。当RP PIO错误在被屏蔽时被检测到时， RP PIO状态寄存器中的相关位仍被置位，但错误不会触发DPC，且错误不会记录在RP PIO日志寄存器中。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

根据RP PIO严重性注册表中相应位的值，将每个未屏蔽的RP PIO错误处理为不可纠正或建议。如果关联的Severity位为Set，则将错误处理为

不可纠正，触发DPC（假设DPC已启用）并使用DPC中断和/或

ERR\_COR（如果启用）。如果关联的严重性位为清除，则错误将作为建议处理（不触发DPC），并使用ERR\_COR（如果启用）发出信号。

执行说明

RP PIO咨询错误处理

每个RP PIO错误都可以作为不可纠正或建议处理。无法纠正的错误处理通常会记录错误，触发DPC，并通过DPC中断、ERR\_COR或两者发出事件信号。咨询错误 处理通常记录错误并用ERR\_COR通知事件。

在某些情况下，软件可以使用RP PIO咨询错误处理来稳健地处理RP PIO错误， 如果在RP中触发DPC，则会导致中断。如果未为给定错误启用RP PIO异常，则每当发生错误时都会返回全1的值。如果错误未触发DPC，则软件可能不确定给定PIO读取返回的全1值是否是完成返回的实际数据值，

指示PIO读取发生错误。如果软件对该错误启用咨询错误处理，则该错误的实例将被记录，从而使软件能够区分这两种情况。

如果DPC在交换机下行端口中被触发，则RP PIO咨询错误处理的使用是特别有益的 并导致RP中的一个或多个完成超时作为副作用，如第2.9.3节所述。如果RP将完成验证错误作为建议处理，则可避免在RP中触发DPC，

继续使用其他交换机下游端口。



RP PIO第一个错误指针、RP PIO标头日志和RP PIOTLP前缀日志的行为与AER中的第一个错误指针、标头日志和TLP前缀日志类似。RPPIO第一个错误指针定义为有效，当其值指示RP PIO状态寄存器中的位被置位时。当RP PIO第一个错误指针有效时，RP PIO日志记录器包含

与所指示的错误相关联的信息。RPPIO ImpSpec日志（如果实施）包含实施特定信息，例如，请求TLP的源。

与AER相反，在AER日志寄存器中记录CTO错误信息是可选的，RP PIO实现必须支持在RP PIO日志寄存器中记录RP PIO CTO错误信息。

如果利用与未完成的PIO请求相关联的接收到的完成TLP检测到错误，则RP PIO集合

以类似的方式确定用于管理错误处理的错误控制位。DPC完成控制位确定UR或CA是否适用，空间（配置、I/O或内存）是相关PIO的空间

请求过程中例如，如果为CA配置了DPC完成控制位，并且根端口收到中毒的

PIO内存读取请求完成时，内存CA Cpl位（位17）用于RP PIO控制和状态寄存器中，以处理错误。

RPPIO系统错误寄存器提供了一种在RP PIO错误发生时生成系统错误的方法。如果在RP PIO系统错误寄存器中的相关位置位时检测到未屏蔽的RP PIO错误，则会生成系统错误。

RPPIO异常寄存器提供了一种在由处理器指令生成的某些跟踪的非发布请求发生错误时生成同步处理器异常109的手段。 参见第2.9.3节。这

异常必须支持所有此类跟踪读取请求，并且可以可选地支持配置写入、I/O写入和 原子操作请求。如果检测到110具有异常支持的非发布请求的错误或者合成了其完成，并且RP PIO异常寄存器中的其相关联位被置位，则生成该异常支持的非发布请求的处理器指令被删除。 非发布请求必须采用同步例外。 即使RP PIO或AER控制指定错误作为屏蔽或建议处理，这仍然适用。



109.“异常”是处理器使用的各种机制的通用术语，包括中断、陷阱、机器检查、指令中止等。

110. 这包括完井TLP本身的任何错误（例如，格式错误的TLP）或完成状态不是成功完成的情况。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

接受同步异常的处理器指令的细节是特定于处理器的，但至少，

该机制必须能够在生成非发布请求的指令完成之前或紧接在该指令之后中断正常处理器指令流。其目的是系统固件、操作系统或两者中的异常处理例程可以检查异常的原因，并在必要时采取纠正措施。

如果处理器生成的读取或原子操作请求发生RP PIO错误，并且RP PIO异常寄存器值不会导致异常，则必须为生成请求的指令返回全1的值。

执行说明

同步异常实现

实现同步异常的确切机制是特定于处理器和平台的。一种可能的实现是使返回到处理器的数据中毒，以用于遇到错误的读或AtomicOp请求。虽然此方法可能适用于这些请求，但可能不适用于配置和I/O 写请求，因为它们不返回数据。

另一种可能的实现方式是用请求已经失败的某种其他类型的指示来标记处理器生成的非发布请求的响应事务，例如，“硬失败”的反应。这 这种方法更有可能适用于所有处理器生成的未提交请求。



执行说明

RP PIO掩码位行为和原理

对于给定的RP PIO错误，RP PIO掩码寄存器中的相关掩码位影响其相关状态位设置、错误记录和错误信令，其方式与AER中掩码位的行为密切相关。

给定RP PIO错误的系统错误生成主要由RP PIO系统错误寄存器中的相关位控制，但也取决于相关RP PIO掩码位是否为清除。选择此行为是为了与AER保持一致，也是因为在不记录原因的情况下生成SysError是不好的做法。

给定RP PIO错误的异常生成与关联的RP PIO掩码位值无关。使用

设想了RPPIO错误需要生成异常而不记录RP PIO错误或触发DPC的模型。



根端口错误处理跟踪的未发布请求，错误不是接收UR和CA完成，

由AER和RP PIO错误控制的组合控制。例如CTO111、接收到的中毒TLP和

变形TLP。对于由AER管理的给定错误，相关联的AER掩码和严重性位确定错误是否必须被处理为不可纠正的错误、被处理为咨询性非致命错误、或被处理为掩码错误。

· 如果AER管理的错误将作为不可纠正的错误处理（请参见www.example.com[部分6.2.2.2](#bookmark6)），则会触发DPC。与请求类型和完成状态相关联的RP PIO系统错误和RP PIO异常位适用。

· 如果AER管理的错误将作为咨询性非致命错误处理（参见[第6.2.3.2.4节](#bookmark13)），则不会触发DPC。RPPIO系统错误和RP PIO异常位适用。

· 如果AER管理的错误将作为屏蔽错误处理（参见[第6.2.3.2.2节](#bookmark14)），则不会触发DPC。RPPIOSysError位不适用，但RP PIO异常位适用。



111. CTO错误在AER和RP PIO中都具有状态和掩码位，尽管RP PIO对于3个空间中的每一个都具有独立的位集合。AER中的其他错误在RP PIO中没有等效错误。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.2.10.4>DPC的软件触发

如果DPC能力寄存器中的DPC软件触发支持位被设置，则软件可以通过以下方式触发DPC

向DPC控制寄存器中的DPC软件触发位写入1b，假设DPC已启用且端口当前未处于DPC。设想该机制对于软件和/或固件开发和测试是有用的。它 还支持软件或固件检查RP PIO错误或RP PIO建议错误的使用模式，

根据情况决定触发DPC。

当此机制触发DPC时，DPC状态寄存器中的DPC触发器原因和DPC触发器原因扩展字段将指示此原因。

如果在DPC软件触发位写入1b时端口已处于DPC中，则端口仍处于DPC中，并且DPC触发原因和DPC触发原因扩展字段不会修改。

执行说明

避免禁用链接和热插拔意外使用DPC

当DPC使能但未触发时，建议软件不要设置链路控制寄存器中的链路禁用位。 设置链路禁用位将导致链路定向到DL\_Down，调用一些类似于DPC中的语义，但缺少其他语义。如果DPC已启用，随后到达的任何已发布请求将 可能会触发DPC如果使能了DPC，建议软件禁用链路的方法是向DPC控制寄存器中的可选DPC软件触发位写入1b。如果DPC软件触发位未

如果已实现，软件应禁用DPC并使用链路禁用。如果操作系统正在执行此操作，但DPC由系统固件拥有，则操作系统应协调禁用DPC与系统固件。

DPC不建议与热插拔惊喜机制同时使用，这通过插槽功能寄存器中的热插拔惊喜位被设置来表示。设置此位会阻止Surprise Down的报告 错误，防止DPC被这个重要错误触发，大大降低了DPC的好处。看到

[第6.7.4.5f](#bookmark62)或关于支持两种机制的插槽的指南



<6.2.10.5>DL\_ActiveERR\_COR信令

DPC能力寄存器中的DL\_Active ERR\_CORS信令支持位表示支持此功能。 该功能由DPC控制寄存器中的DL\_ACTIVE ERR\_COR使能位使能。DL\_ACTIVE状态为

由链路状态寄存器中的数据链路层链路活动位指示。DL\_ACTIVE ERR\_COR信令独立于数据链路层状态更改中断进行管理，并且允许同时使用这两种机制。

如果DL\_ACTIVE ERR\_COR使能位被置位，并且设备控制寄存器中的可纠正错误报告使能位或DPC控制寄存器中的DPCSIG\_SFW使能位被置位，则每次链路转换到DL\_ACTIVE状态时，端口必须发送ERR\_COR消息。DL\_ACTIVE ERR\_COR信令不得设置设备状态寄存器中的可纠正错误检测位，因为此事件不作为错误处理。如果下游端口支持ERR\_COR

子类功能，此DPC ERR\_COR信令事件必须设置DPC状态寄存器中的DPCSIG\_SFW状态

并且还设置ERR\_COR消息中的ERR\_COR子类字段以指示ECSSIG\_SFW。与数据链路层状态改变中断相反，DL\_Active ERR\_COR信令仅指示链路进入DL\_Active状态，而不是链路退出DL\_Active状态。

对于给定的DL\_ACTIVE事件，如果端口将发送ERR\_COR消息和MSI/MSI-X事务，则端口必须在发送MSI/MSI-X事务之前发送ERR\_COR消息。没有相应

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

如果INTx机制被用于发送DL\_ACTIVE中断信号，则这是不必要的，因为INTx消息在通过路由元素时不一定保持相对于ERR\_COR消息的顺序。

执行说明

DL\_ACTIVE ERR\_COR信令

建议操作系统在以下情况下使用数据链路层状态更改中断进行信令：

DL\_ACTIVE更改状态。虽然DL\_ACTIVE ERR\_COR信令指示相同事件的子集，但是DL\_ACTIVEERR\_COR信令主要旨在由系统固件使用，当它需要被通知以便执行时， 下游端口配置或提供固件优先服务。



6.3虚拟通道支持

6.3.1引言和范围

虚拟通道机制为在PCI Express中支持差异化服务提供了基础

织物.它支持部署独立的物理资源，这些资源与流量标签一起，

优化处理差异化流量。 使用流量类TLP级别标签支持流量标签。流量区分的策略由TC/VC映射以及基于VC、基于端口和基于功能的策略 仲裁机制。TC/VC映射取决于平台应用需求。这些要求驱动仲裁算法的选择，并且仲裁器的可配置性/可编程性允许流量服务策略的详细调整。

虚拟信道和相关业务类机制的定义在第2章中介绍。第7.9.1节和第7.9.2节定义了VC配置/编程模型。

本节从系统的角度介绍VC机制它解决了以下方面的下一个细节：

·支持TC/VC配置

·基于VC的仲裁-算法和规则

·流量排序注意事项

· 同步支持作为特定使用模型

6.3.2 TC/VC映射和示例

当一个或多个TC与由VC ID指定的物理资源相关联时，建立虚拟信道。结构内给定路径上支持的每个流量类都必须映射到其中一个启用的虚拟 渠道每个端口都必须支持默认的TC 0/VC 0对-这是“硬连线“任何额外的TC映射或

附加VC资源启用是选项al，并由系统软件使用第7.9.1节和第7.9.2节中描述的编程模型进行控制。

在组件内供应的或在给定结构内启用的VC资源的数量可能由于以下因素而变化：

实现和使用建模需求，这是由于具有不同资源不同组件的热插拔 能力，或者由于系统软件限制在结构内的给定路径上可以启用什么资源。

一些例子来说明：

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·一组组件（根复合体、端点、交换机）可以仅支持强制VC 0资源， 必须将TC 0映射到VC 0。 系统软件可以根据应用程序使用要求，将一个或所有非零TC映射到VC 0以及结构内的任何或所有路径上。

·一组组件可以支持两个VC资源，例如， V C0和VC1。系统软件必须映射TC 0/VC 0

并且另外，可以将一个或所有非零TC标签映射到VC 0或VC 1。如上所述，可以在结构内的任何或所有路径上启用这些映射。请参考以下示例以获取更多信息。

·交换机可以用八个端口实现-七个xl链路与两个VC资源，一个x16链路与一个VC资源。系统软件可以在x1链路上启用两个VC资源，并且分配一个或多个VC资源。

附加TC，从而允许交换机区分在任何端口之间流动的流量。 x16链路还必须配置为将任何非TC 0流量映射到VC 0（如果此类流量要在此链路上流动）。注意：

需要多端口组件（交换机和根复合体）来支持每个端口的独立TC/VC映射。

在以上示例中的任一个中，系统软件具有将TC中的一个、全部或子集映射到给定VC的能力。 如果系统软件希望限制可以流过给定链路的业务类别的数量，则它可以仅将TC的子集配置到启用的VC资源。任何表示TC尚未映射到已启用VC

资源必须被视为格式错误的TLP。这被称为TC滤波。此TLP的流控制信用将丢失，并且将生成无法纠正的错误，因此通常需要软件干预来恢复正确的 TC过滤事件发生后的操作。

TC过滤的图形示例如[图6-4所示](#bookmark63)，其中TC（2：6）未映射到连接端点A和交换机的链路。这意味着在交换机和端点A之间不允许具有TC（2：6）的TLP。

开关



TC7

|  |
| --- |
| 根  复杂  TC[0：1]  TC[2：4]  TC[5：6] |

链路

|  |
| --- |
| 端点A  TC[0：1]    TC7 |



VC0

VC3

TC[0：1]

链路

TC7



VC0

TC[0：1]



VC1

TC[2：4]

链路

|  |
| --- |
| 端点B  TC[0：1]  TC[2：4]  TC[5：6]  TC7 |



VC2

TC[5：6]

VC0

TC[0：1]

VC3

TC7

VC1

TC[2：4]

VC2

TC[5：6]

TC7

VC3



映射

OM13828

图6-4 TC滤波示例

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

[图6-5](#bookmark64)显示了TC到VC映射的示例。 具有一个下游端口和一个上游端口的简单交换机将端点连接到根复合体。在上游端口，两个VC（VC 0和VC 1）通过以下方式启用

映射：TC（0-6）/VC 0，TC 7/VC 1。 在下游端口t处，仅使能VC 0，并且所有TC被映射到VC 0。在这

例如，当TC 7在下游端口被映射到VC 0时，它在上游端口被重新映射到VC 1。 虽然端点仅支持VC 0，但当它用不同的TC标记事务时，与TC 7相关联的来自/去往端点的事务可以利用在交换机和根联合体之间启用的第二虚拟信道。

链路

开关

链路

|  |
| --- |
| 端点  TC[0：7] |



VC0

TC[0：6] TC7



VC0

TC[0：7]

VC1

映射

|  |
| --- |
| 根  复杂  TC[0：6]  TC7 |

OM13829

图6-5 TC到VC映射示例

执行说明

单个VC上的多个

单个VC实现可以受益于使用多个TC。TC提供可用于区分端点或根联合体内的流量的排序规则，而与所支持的VC的数量无关。

在仅支持VC 0的简单配置中，可能无法在

这是最佳方式，因为不同的TC不能物理分离。然而，携带的好处

多个TC仍然可以部分地在小型和

直接连接到根复合体，而不是通过级联开关。在这些拓扑中，

以根复合体为目标只需要遍历单个链路，并在两个链路上优化分组调度

与使用单个TC的情况相比，基于TC的端点和根复合体（端点和根复合体）可以实现显著的改进。尽管如此，无法通过完全独立的资源路由不同的流量，

独立的流控制和独立的排序将所有业务暴露于潜在的线头阻塞条件。优化端点内部体系结构以最大限度地减少阻塞条件的暴露可以降低这些风险。



6.3.3 VC仲裁

仲裁是虚拟通道机制的关键方面之一，其定义方式完全支持

具体应用的可配置性。一般而言，基于VC的仲裁机制的定义由以下目标驱动：

·为了防止错误的事务超时并保证数据流向前进展

·在结构内的数据流之间提供差异化服务

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·在组件之间提供具有确定性（并且合理较小）的端到端延迟的保证带宽

链路是双向的，即，每个端口可以是入口端口或出口端口，这取决于业务流的方向。[图6-6](#bookmark65)中以3端口交换机为例说明了这一点，其中交换机端口之间的通信流路径 用不同类型的线条突出显示。在下面的部分中，VC仲裁是使用交换机仲裁模型定义的，因为交换机表示从仲裁角度来看的功能超集。

此外，在描述中使用单向数据流。



pciexpress链路

|  |
| --- |
| TX |

|  |
| --- |
| RX |

出口

|  |
| --- |
| TX |

|  |
| --- |
| RX |

出口

|  |  |  |
| --- | --- | --- |
| 出口 |  | 入口 |

|  |
| --- |
| TX |

|  |
| --- |
| RX |

表达 表达

入口

PCI

PCI链路

入口

链路

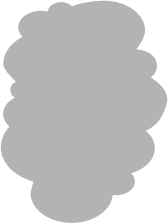
3端口交换机

OM13830

图6-6入口和出口的交通流示例

<6.3.3.1>业务流与交换仲裁模型

下图（[图6-7](#bookmark66)和[图6-8）](#bookmark67)说明了通过交换机的流量，并总结了仲裁的关键方面。



|  |
| --- |
| 2.1b |

|  |
| --- |
| 2 |

|  |
| --- |
| 2.1b |

入端口

出端口

|  |  |  |
| --- | --- | --- |
| 2.1a |  | 3.1 |
|  |

|  |
| --- |
| 3 |

|  |
| --- |
| 1 |

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |
| --- |
| 2.0b |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2.0 |  | 3.0 |
|  |  |

|  |
| --- |
| 2.0a |

|  |
| --- |
| 0 |

|  |  |  |
| --- | --- | --- |
| 3.1b |  | 3.1a |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2.0 |  | 2.0b |
|  |  |

|  |  |  |
| --- | --- | --- |
| 2.1a |  | 2.0a |
|  |

|  |
| --- |
| 3.0 |

|  |
| --- |
| 3.1b |

|  |  |  |
| --- | --- | --- |
| 3.1a |  | 3.1 |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  |  |  | | --- | | 3.0 | |  | 优先级顺序 |
| 入口  端口号  出口  端口号 | |

OM14284

图6-7通过交换机的区分流量示例

[图6-7](#bookmark66)中的小方框表示每个入口端口的传入流量这些盒子代表了

在不同VC内携带的分组使用不同的灰度级来进行消隐。表示属于不同VC的分组的每个框包括入口和出口端口的指定，以指示分组在何处被发送。

数据包来自何处以及它将去往何处。例如，指定“3.0”意味着该数据包将到达

端口#0（入口），目的地为端口#3（出口）。 在交换机内，分组根据交换机内部仲裁机制进行路由和服务。

交换机仲裁模型定义交换机内所需的仲裁基础设施和功能这

需要功能来支持一组仲裁策略，该仲裁策略控制来自多个入口端口的对出口端口的业务竞争。

[图6-8](#bookmark68)显示了交换机的概念模型，突出显示了入口到出口方向的资源和相关功能。请注意，交换机中的每个端口都可以充当入口或出口端口。因此，这一数字仅

显示了一个特定场景，其中本示例中的4端口交换机在端口#0和端口#1上具有入口流量，

将端口#2作为出口端口。另一个示例可能显示不同的流量流，这意味着交换机上端口的角色不同。PCI Express架构通过交换机实现对等通信，因此，

使用相同实例的可能情形可包括多个单独且同时的入口到出口流（例如，端口0到端口2和端口1到端口3）。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

港口仲裁

在出口端口中

VC仲裁

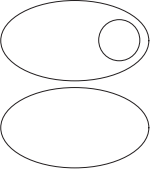
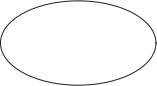
TC/ VC

映射

的

出口

端口

ARB

对于出口端口

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
|  |
| > 0 |
|  |
| > 1 |
|  |

|  |
| --- |
| 2 |

VC0-

VC0

ARB

入端口

VC1一

出端口



ARB

TC/ VC

映射

的

出口

端口



VC 1

|  |
| --- |
| 3 |

这些结构被复制到每个出口端口

OM14493B

图6-8交换机仲裁结构

以下两个步骤从概念上描述了交换机在端口0和端口1上接收到的、目的地为端口2的流量的路由。首先，根据TLP报头中的地址/路由信息确定目标Egr端口。其次， 基于出口端口的TC/VC映射来确定出口端口的目标VC。对象的事务

在出口端口中的相同VC，但来自不同的入口端口，必须在它们可以被转发到出口端口中的相应资源之前进行仲裁。该仲裁被称为港口仲裁。

一旦流量到达出口端口中的目的地VC资源，它就受到共享链路的仲裁。 从出口端口的角度来看，该仲裁可以在概念上被定义为简单形式的复用，其中复用控制基于固定的或可编程的/可编程的仲裁策略。在出口端口处的不同VC之间的仲裁的这个阶段被称为出口端口的VC仲裁。

独立于VC仲裁策略，与每个VC相关联的管理/控制逻辑必须遵守事务排序和流控制规则，然后才能使仲裁机制访问未决流量。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

出口VC控制逻辑

每个出口的VC控制逻辑包括：

·VC流控制逻辑

·VC订购控制逻辑

流控制信用在连接到同一链路的两个端口之间交换。流量控制的可用性

信用是VC控制逻辑必须使用的限定符之一，以决定何时允许VC竞争

共享链路资源（即，数据链路层传输/重试缓冲器）。如果由于缺少足够数量的流控制信用而不能提交候选分组，则VC控制逻辑必须屏蔽未决分组的存在。

数据包，以防止来自其他VC的流量阻塞。注意，由于每个VC包括用于执行以下操作的缓冲资源：

对于已发布请求、未发布请求和完成分组，VC控制逻辑还必须考虑特定候选分组的流控制信用的可用性。此外，VC控制逻辑必须遵守 已过帐/未过帐/完成交易的排序规则（更多详情请参见第2.4节），以防止

死锁和违反生产者/消费者订购模型。



<6.3.3.2>VC仲裁-VC之间的仲裁

本规范通过VC标识（VC ID）分配定义了默认的VC优先级VC ID是

在虚拟信道容量结构或多功能虚拟信道容量结构中，

渠道能力结构。 [图6-9中的示例](#bookmark70)说明了一个支持8个VC的端口，VC 0被视为最低优先级，VC 7被视为最高优先级。

VC VC ID

资源

相对优先级

VC仲裁 使用

例如

扩展VC计数= 7

对于等容通信

|  |
| --- |
| VC7 |
| VC6 |
| VC5 |
| VC4VC3 |
| VC2 |
| VC1 |
| VC0 |

高

第8名VC第7名VC第6名VC第5名VC第4名VC第3名VC第2名VC第1名VC



严格

优先

低优先级扩展VC计数= 3

管辖 VC仲裁能力字段（例如WRR）

对于QoS使用

默认VC（PCI Express/PCI）

低

OM14287

图6-9 VCID和优先级顺序-示例

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

默认优先级的可用性并不限制可以实现以支持VC仲裁的算法的类型-特定于实现的方法或架构定义的方法之一

·严格优先级-基于固有的优先级，即，VC 0=最低，VC 7=最高

· Round Robin（RR）-最简单的仲裁形式，所有VC具有相同的优先级

·加权RR-可编程权重因子确定服务级别

如果硬件支持VC资源子集的严格优先级仲裁，则软件可以将VC配置为两个优先级组-较低和较高组。上层群被视为严格优先仲裁群

而仅当在上组中没有分组要处理时才仲裁到下[图6-9](#bookmark70)

示出了支持分成两组的八个VC的示例配置，下面的组包括 VC 0-VC 3和由VC 4-VC 7组成的上组。较低组内的仲裁可以配置为支持的仲裁方法之一。端口VC能力寄存器1中的低优先级扩展VC计数

表示该组的大小。仲裁方法列在端口VC中的VC仲裁能力字段 能力寄存器2。详情请参见第7.9.1节和第7.9.2节。 当低优先级扩展VC计数字段被设置为零时，所有VC由严格优先级VC仲裁管理;当该字段等于扩展VC计数时，所有VC由VC仲裁能力字段指示的VC仲裁管理。

6.3.3.2.1严格优先仲裁模式

严格的优先级仲裁使高优先级事务的延迟最小化。然而，如果不正确应用，则存在带宽饥饿的潜在危险。使用严格优先级要求所有高优先级流量

根据最大峰值带宽和链路使用持续时间进行调节规则必须适用于以下两种情况：

事务注入端口/功能内或在数据流争用公共链路的后续出口端口内。 系统软件必须配置流量，以便以足够的速率为较低优先级的事务提供服务，以避免事务超时。

6.3.3.2.2循环仲裁模式

循环仲裁用于在事务级向所有业务提供相等的机会。注意，该方案用于需要以相同的优先级服务不同的无序流的情况。

在需要区分的情况下，可以使用加权循环方案。WRR方案是

通常在带宽调节不被业务源强制执行的情况下使用，因此不可能在不冒较低优先级业务饥饿的风险的情况下使用优先级方案。关键是这个方案

通过允许每个仲裁循环至少有一个仲裁成功，在业务竞争期间提供公平性。在争用期间，每个VC的最小允许带宽和最大突发性都由带宽权重来调节。这意味

它限制了来自不同VC的流量的仲裁延迟。请注意，延迟还取决于映射到这些VC上的流量所允许的最大数据包大小。

WRR方案的关键使用模型之一是支持QoS策略，其中可以使用不同的权重提供不同的QoS级别。

虽然某些应用的权重可以固定（通过硬件实现），但为了为不同应用提供更通用的支持，建议使用支持WRR方案的组件来实现可编程WRR。使用第7.9.1节和第7.9.2节中定义的软件界面控制WRR的编程。



112.请注意，这并不意味着在使用bandwidth方面的等价性和公平性。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.3.3.3>Port Arbitration-Arbitration Within VC

对于交换机，端口仲裁指的是来自映射到同一VC的其他入口端口的流量之间在出口端口处的仲裁。对于根端口，端口仲裁是指在根出口端口处的仲裁， 来自映射到同一VC的其他根入口端口的对等流量。对于RCRB，港口仲裁

指的是RCRB处的仲裁（例如，对于主内存）之间的流量，这些流量来自根端口， 同样的VC 在此上下文中，用于VC之间的仲裁的固有优先级排序方案是不适用的，因为它将意味着不同端口的严格仲裁优先级。来自不同端口的流量可以使用以下方式

支持的计划：

· 硬件固定仲裁方案，例如，循环赛

· 可编程WRR仲裁方案

· 可编程时基数据WRR仲裁方案

硬件固定RR或RR类方案是最简单的实现，因为它不需要任何可编程性。它使所有端口具有相同的优先级，这对于没有软件管理的差异化的应用程序是可以接受的， 需要基于每个端口的带宽预算。

可编程WRR允许灵活性，因为它可以作为平坦RR操作，或者如果需要区分，则可以以与第www.example.com节中描述的类似方式将可编程权重应用于来自不同端口[6.3.3.2的流量。](#bookmark69) 该方案是 在需要为不同端口提供不同带宽分配的情况下使用。

基于时间的WRR用于不仅需要不同的带宽分配而且需要严格控制带宽使用的应用程序。该方案允许控制可以从

在一定的时间内，不同的港口。这是某些应用程序（如等离子服务）的要求，其中流量需要满足严格的截止日期要求。[第6.3.4节](#bookmark71)提供了支持等值线的基本规则

应用. 有关基于时间的仲裁和作为此仲裁方案的使用模型的isocommunication服务的更多详细信息，请参阅附录A。

<6.3.3.4>多功能设备和功能仲裁

多功能仲裁模型定义了一个可选的仲裁基础设施和功能，

多功能设备。需要这种功能来支持一组仲裁策略，这些仲裁策略控制来自设备的多个功能的设备的上游出口端口的流量争用。

[图6-10](#bookmark72)显示了一个多功能设备的概念模型，突出显示了资源和相关功能。注意，每个功能可选地包含VC能力结构，如果存在，则其管理TC/VC映射，

可选的端口仲裁和可选的VC仲裁，都在功能内。MFVC能力结构管理

TC/VC映射、可选功能仲裁和设备上行出口端口的可选VC仲裁。

这些资源共同实现了对上游请求的增强QoS管理。然而，与在其下游端口上具有设备的完整交换机不同，多功能设备模型不支持针对以下设备的完整QoS管理：

功能之间的对等请求或下游请求。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

VC能力结构 MFVC能力结构

托管资源 托管资源





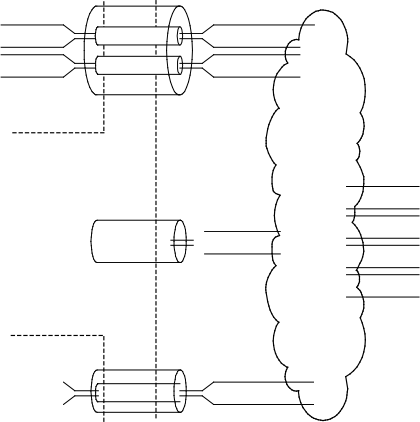
内部链接

功能0

多功能胶水逻辑

、

I I I I I I I I I I I I I I I I I I I I I I

 可选端口

TC/VC 映射

TC[0：1] TC7

VC0 VC3

TC[0：1] TC7

仲裁

外部端口

 或VC 

仲裁

TC/VC映射

功能1

VC0

 TC/VC 

内部链接

TC[0：1]

TC[2：4]

TC[5：6]

TC7

映射 

可选端口i

我我

 VC1





任择 功能仲裁

TC0， TC[2：4]

 VC0

TC0， TC[2：4]

我我

仲裁

VC2





我

( 或VC 、仲裁

我

VC3

任择

我我

VC

仲裁

我

内部链接

我

功能2

我我

我我

|  |
| --- |
| TC0 |



TC0

我

VC0



我



L

A-0411B

图6-10多功能仲裁模块

如下管理在功能处发起的上行流请求的QoS。首先，特定于功能的机制将TC应用于请求。例如，一个设备驱动程序可能会配置一个函数来用TC7标记它的所有请求。

接下来，如果函数包含VC能力结构，则它将TC/VC映射指定到函数的VC之

资源（可能是函数的单个VC资源）。此外，VC能力结构支持功能VC资源的启用和配置。

如果功能是交换机，并且目标VC资源支持端口仲裁，则该机制控制

交换机的多个下游入口端口为该VC资源进行仲裁。如果端口仲裁机制支持基于时间的WRR，则这还控制来自每个下游入口端口的请求的注入速率。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

如果功能支持VC仲裁，则该机制管理功能的多个VC资源如何仲裁到MFVC资源的概念性内部链接。

一旦请求分组概念上到达MFVC资源，TLP报头中的地址/路由信息

确定请求是上行还是对等到另一个函数。对于对等网络，QoS 管理留给非体系结构的设备专用机制。对于上游的情况，MFVC能力结构中的TC/VC映射确定请求将以哪个VC资源为目标。MFVC能力结构还支持 在多功能粘合逻辑中启用和配置VC资源。如果目标VC资源支持

功能仲裁，该机制控制多个功能如何仲裁该VC资源。如果功能仲裁机制支持基于时间的WRR，则这控制每个功能的请求到该VC资源中的注入速率。

最后，如果MFVC能力结构支持VC仲裁，则该机制控制MFVC的多个VC如何竞争设备的上游出口端口。独立于VC仲裁策略、管理/控制逻辑

与每个VC相关联的处理器必须遵守事务排序和流控制规则，然后才能使未决业务对仲裁机制可见。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0



执行说明

多功能仲裁错误行为

[表6-6](#bookmark73)显示了与[图6-10所示的示例拓扑相关的预期错误行为。](#bookmark72)

表6-6多功能仲裁错误模型示例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 源 | TC | 目的地 | | | |
| 功能0 | 功能1 | 功能2 | 外部端口 |
| 功能0 | 0 | n/a | 好 | 好 | 好 |
| 1 | MF@F1 | MF@F2 | 好 |
| 2 - 6 | MF@F0 | MF@F0 | MF@F0 |
| 7 | MF@F1 | MF@F2 | 好 |
| 功能1 | 0 | 好 | n/a | 好 | 好 |
| 1 | MF@F1 | MF@F1 | MF@F1 |
| 2 - 4 | MF@F0 | MF@F2 | 好 |
| 5 - 7 | MF@F1 | MF@F1 | MF@F1 |
| 功能2 | 0 | 好 | 好 | n/a | 好 |
| 1 - 7 | MF@F2 | MF@F2 | MF@F2 |
| 外部端口 | 0 | 好 | 好 | 好 | n/a |
| 1 | 好 | MF@F1 | MF@F2 |
| 2 - 4 | MF@F0 | 好 |
| 5 - 6 | MF@F1 |
| 7 | 好 |

图例：

|  |  |
| --- | --- |
| 好 | 成功 |
| MF@F0 | TLP变形，在随访0时 |
| MF@F1 | 成形不良的TLP，在随访1时 |
| MF@F2 | 成形不良的TLP，在随访2时 |
| n/a | 不适用（功能/端口发送到自身） |

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

无MFVC能力结构的多功能器件

如果多功能设备缺乏MFVC能力结构，则来自不同设备的数据流的仲裁将被中断

多功能设备的功能超出了本规范的范围。但是，如果多功能设备支持TC 0以外的TC，并且不实现MFVC能力结构，则必须实现单个 功能0中的VC能力结构，为链路提供架构TC/VC映射。



6.3.4等静压支承

服务于异构数据传输要求系统不仅提供有保证的数据带宽，

确定性服务延迟。定义了对等支持机制，以确保对等流量

在相关的时间段内接收其分配的带宽，同时还防止系统中的其它业务的饥饿。对等网络支持机制适用于端点和根复合体之间的通信，以及 点对点通信。

通过合理使用TC事务标记、VC数据传输等机制实现异构

协议和TC到VC映射。端到端的异构化服务要求软件在异构化器和完成器之间的路径上设置正确的配置。本节介绍软件配置规则和 硬件组件必须遵循的规则来提供端到端的异构服务。有关等离子体应用和等离子体服务设计指南的更多信息和背景材料，请参见附录A。

<6.3.4.1>软件配置规则

系统软件必须遵守以下规则，才能配置PCI Express交换矩阵以进行等容通信：

·软件必须指定一个或多个TC用于等容交易。

·软件必须确保针对同一完成者的所有isoport请求的属性字段是固定和相同的。

·软件必须配置所有用于支持以必要的带宽和延迟服务（仲裁）等离子通信的VC资源，以满足应用目标。这可以使用严格的

优先级、WRR或硬件固定仲裁。

·软件不应在给定VC上将等容通信量与非等容通信量

·软件必须遵守端口或RCRB报告的最大TimeSlots功能。

·软件不得将所有链路容量分配给对等流量。这是为了确保其他非对等事务的必要向前进展，以避免错误的事务超时。

·软件必须限制每个支持ISO的路径的Max\_Payload\_Size，以满足ISO延迟。例如，在从具有isocommunication能力的设备到根联合体的路径上流动的所有流量 应将其限制为不超过满足等负载要求的Max\_Payload\_Size

延迟要求。

·软件必须将等时配置设备的Max\_Read\_Request\_Size设置为不超过为设备设置的Max\_Payload\_Size的值。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.3.4.2>请求规则

要求提供等离子体服务的制造商必须遵守以下规则：

·读取请求的长度字段中的值必须永远不超过Max\_Payload\_Size。

·如果等容通信以根联合体为目标并且RCRB指示其不能满足等容通信，

带宽和等待时间要求，而不要求所有事务设置无监听属性位，

如果通过设置监听监听事务位来指示，则必须在TLP报头内设置该位，否则将拒绝事务。

<6.3.4.3>填写规则

提供对等服务的完成者必须遵守以下规则：

·在正常操作条件下，完成人不应将流量控制引起的背压应用于均匀注入的等渗液请求。

·完成者必须在VC资源能力寄存器中的最大时隙字段中报告其等带宽能力。请注意，完整r必须考虑部分写入。

·完成者必须遵守最大的isoconversion事务延迟。

·作为编译器的根联合体必须实现至少一个RCRB，并且支持用于相关联的VC的基于时间的端口仲裁。请注意，基于时间的端口仲裁仅适用于请求事务。

<6.3.4.4>交换机和根复合体规则

提供在线服务的交换机必须遵守以下规则。同样的规则也适用于支持根端口之间对等的异构数据流的根复合体，在本节中缩写为

·在正常操作条件下，等时配置的交换机或P2P-RC端口不应将流量控制引起的背压应用于均匀注入的等时请求。

·等时配置的交换机或P2P-RC端口必须遵守最大等时事务延迟。

·交换机或P2P-RC组件必须支持每个端口的基于时间的端口仲裁，该端口支持能够支持对等业务的一个或多个VC。请注意，基于时间的端口仲裁适用于请求

交易，而不是完成交易。

<6.3.4.5>Multi-Fun游戏机规则

包含提供等容服务的MFVC能力结构的多功能设备必须遵守以下规则：

·在正常操作条件下，为等容操作配置的MFVC胶合逻辑不应向来自其功能的均匀注入的等容请求施加背压。

·MFVC能力结构必须支持每个VC的基于时间的功能仲裁，

支持对等网业务。注意，基于时间的功能仲裁仅适用于上行请求

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

它不适用于任何下游或对等请求交易，也不适用于任何完成交易。

缺乏MFVC能力结构的多功能设备没有架构机制来为其多个功能同时提供同步服务。

6.4设备同步

系统软件需要一个

系统中的设备。例如，如果没有这样的机制，则在系统操作期间对总线号重新编号可能 当给定设备的请求或完成仍在进行中时，导致该设备的授权器ID（包括总线号）改变，并且因此可能由于授权器ID的改变而变得无效。还希望能够确保在热插拔有序移除期间没有未完成的事务。

停止机制实现的细节取决于设备硬件、设备驱动程序软件和系统

软件然而，必须支持以允许结构的系统软件管理的基本要求包括以下能力：

· 阻止设备生成新请求

· 阻止生成向设备发出

· 确定设备正在处理的所有请求均已完成

· 确定设备发起的所有未发布请求已完成

· 确定由设备发起的所有已发布请求是否已到达其目的地

驱动程序和/或系统软件阻止来自器件的新请求的能力由每个器件功能的命令寄存器（第7.5.1.1.3节）中的总线主机使能、SERR#使能和UART禁用位以及其他此类控制位支持。

发送到设备的请求通常在驱动程序的直接控制下，因此系统软件可以通过指示驱动程序停止生成这些请求来阻止这些请求（此通信的详细信息由系统软件决定

具体）。类似地，由设备服务的请求通常在设备驱动程序的控制下，因此确定这样的请求的完成通常是微不足道的。

事务挂起位在每个功能的基础上为软件提供一致的方法，以确定设备发出的所有未发布请求已完成（请参见第www.example.com节7.5.3.5）。

通过生成一个事务来“刷新”任何未完成的请求，从而确定已发布的请求是否已到达其目的地。使用TC 0对系统内存的写入将被设备的主机读取刷新，因此不需要显式刷新协议。 使用TC 0以外的TC进行写入需要某种类型的刷新同步机制。的

该机制本身是特定于设备及其驱动程序软件的实现。然而，在所有情况下，

硬件和软件实现者应该彻底理解2.4节中描述的排序规则。如果为设备发起的任何请求设置了“宽松排序”或“基于ID的排序”属性，则尤其如此。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

冲洗机构

在诸如端点通过TC 0仅与主机存储器通信的简单情况 通过从端点读取来实现。如果端点使用TC 0以外的TC向主存储器发出写入，则可以通过在定向到主存储器的相应TC上的存储器读取来实现“刷新”。 内存读取需要在端点正在使用的所有TC上执行。

如果使用内存读取来“刷新”未完成的事务，但不需要实际读取，则可能需要使用第2.2.5节中描述的零长度读取语义。



设备之间的对等交互需要所涉及的设备之间的显式

即使所有通信都是通过TC 0。对于给定的系统，管理对等交互的模型必须是 确立了习 系统软件、设备硬件和软件必须符合此模型。阻止请求生成和确定请求完成的要求与非对等交互的要求相匹配，但是确定发布的请求已经到达对等目的地设备需要明确的

同步机制该机制本身是特定于设备、其驱动程序软件以及用于建立和取消对等通信的模型的实现。

6.5锁定事务

6.5.1引言

需要锁定事务支持，以防止在使用遗留软件的系统中出现死锁，

访问I/O设备。请注意，某些CPU可能会因执行隐式触发锁定的指令而生成锁定访问。有些遗留软件滥用这些事务，即使不需要独占访问也会生成锁定序列。 由于对I/O设备的锁定访问会引入除上述之外的潜在死锁以及严重的性能下降，因此PCI Express端点禁止支持锁定访问，并且新软件不得使用将导致对I/O设备的锁定访问的指令。旧版端点仅为了与现有软件兼容而支持锁定访问。

仅允许根联合体在PCI Express上发起锁定请求。端点发起的锁定请求

桥梁不支持。 这与[PCI]（附录F-独占访问）中列出的锁定事务使用限制一致。

本节指定与支持从主机CPU到传统端点的锁定访问相关的规则，包括通过交换机和PCI Express/PCI网桥传播这些事务。

6.5.2锁定交易的发起和传播-规则

锁定事务序列由主机CPU生成，作为对同一位置的一个或多个读取，然后是多个写入。当一个锁被建立时，所有其他流量都被阻止使用根

Complex和锁定的Legacy Endpo int或Bridge。

·使用“锁”型读取请求/完成（MRdLk/CplDLk）在PCI Express上发起锁定事务序列或尝试锁定事务序列，

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

. 以成功完成以外的状态完成的锁定请求不会建立锁定（在以下部分中详细解释）

. 无论与锁定序列关联的任何完成的状态如何， 序列和试图锁定的序列必须通过发送一个锁定消息来终止。

. MRdLk、CplDLk和RISK语义仅允许用于默认流量类（TC 0）

. 在一个层次结构域内，在给定时间只能进行一次锁定事务

·从根复合体沿着锁定的事务路径向下发送到完成器的同步消息，并且可以从根复合体广播到所有端点和网桥

. 任何未参与锁定序列的设备必须忽略此消息

·任何违反锁定事务的发起和传播规则的行为都可能导致未定义的设备和/或系统行为。

. 通过PCI Express的锁定事务序列的发起和传播如下执行：

·锁定的事务序列以MRdLk请求开始

. 锁定事务序列的任何连续读取

. 任何MRdLk请求的完成对成功请求使用CplDLk完成类型，对不成功请求使用CplLk完成类型

· 如果与锁定序列相关的任何读取未成功完成，则验证器必须确保锁的原子性不再有保障，并且验证器和完成器之间的路径不再锁定

·锁定序列的所有写入都

·锁定消息用于指示锁定序列的结束。交换机向已锁定的出口端口

· 在接收到一个HTTP消息后，如果旧端点或网桥处于锁定状态，则它必须解锁自身。如果未锁定，或者接收器是不支持锁定的PCI Express端点或桥接器，

已忽略消息并丢弃

6.5.3开关和锁-规则

交换机必须将与锁定序列相关的事务与其他事务区分开来，以防止其他事务

事务干扰锁并可能导致死锁。以下规则涵盖了如何做到这一点。请注意，锁定的访问仅限于TC 0，它始终映射到VC 0。

·当交换机将MRdLk请求从入口端口（最接近根复合体）传播到出口端口时，它必须阻止映射到默认虚拟通道（VC 0）的所有请求被传播到出口端口

. 如果在此入口端口接收到寻址不同出口端口的后续MRdLk请求，则交换机的行为未定义

注：PCI Express不支持这种分割锁定访问，软件不得导致这种锁定访问。这种访问可能导致系统死锁。

·当返回第一个MRdLk请求的CplDLk时，如果完成指示成功完成状态，则交换机必须阻止来自所有其他端口的所有请求被传播到端口中的任一个 除了映射到出口端口上的非VC 0的请求之外，

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·锁定序列中涉及的两个端口必须如上所述保持被阻塞，直到交换机接收到MRL消息（在入口端口处，用于初始MRdLk请求）。

. 必须将该消息转发到锁定的Egres端口。该消息可以广播到所有其他端口

. 一旦解锁消息到达，入口端口被解锁，并且在从出口端口传输解锁消息之后，被封锁的出口端口

▪ 未在锁定的访问中使用的端口不受此消息的

6.5.4 PCI Express/PCI桥和锁-规则

PCIExpress/PCI网桥的要求与交换机的要求类似，不同之处在于，由于PCI Express/PCI网桥仅使用默认的虚拟通道和流量类，因此在锁定访问期间，所有其他流量都将被阻止。PCI Express/PCI桥的PCI总线侧的要求与PCI/PCI桥的要求相

[PCI-to-PCI-Bridge-1.2]和[PCIe-to-PCI-PCI-X-Bridge-1.0]）。

6.5.5根复合体和锁-规则

根复合体被允许作为一个加密器来支持锁定的事务。如果支持锁定事务，根复合体必须遵循第6.5.2节中描述的顺序来执行锁定访问。根联合体用于将PCI Express连接到主机CPU的机制不在本文档的范围内。

6.5.6遗留终点

允许遗留端点支持锁定访问，但不鼓励使用它们。如果支持锁定访问，则遗留端点必须按以下方式处理它们：

·当传统端点以成功完成状态为锁定访问的第一个读取请求的第一个完成进行验证时，传统端点变为锁定

. 如果完成状态不是“成功完成”，则传统结束点不会被锁定。锁定后，旧终结点必须保持锁定状态，直到它收到“消息”

·锁定时，传统端点不得使用映射到默认虚拟通道（VC 0）的TC

请注意，在存在多个可能的请求源的情况下，此要求适用于端点内所有可能的请求源。

. 可以使用映射到默认虚拟通道以外的VC的 TC

6.5.7 PCI Express终端

PCI Express终端不支持锁定。PCIExpress Endp点必须将MRdLk请求视为不支持的请求（参见第2章）。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.6 PCI Express重置-规则

本节指定PCI Express重置机制。 本节介绍本文档中定义的体系结构机制与本文档中定义的重置机制之间的关系。之间的任何关系 PCI Express常规复位和组件或平台复位是组件或平台特定的（除了

明确指出）。

6.6.1常规复位

常规复位包括除功能级复位以外的所有复位机制。有两类

常规复位：基本复位和非基本复位的复位。 本节适用于所有类型的常规复位。

在所有形式因素和系统硬件配置中，在某种程度上，必须存在用于将所有端口状态设置或返回到本文档中指定的初始条件的硬件机制-该机制称为 复位2.4.9.1“该机制可以采用系统提供给组件或适配卡的辅助信号的形式，在这种情况下，信号必须称为PERST#，并且必须符合第4节中规定的规则。当

PERST#提供给组件或适配器，该信号必须由组件或适配器用作基本复位。当未向组件或适配器提供PERST#时，组件或适配器会自动生成基本重置，有关如何执行此操作的详细信息不在本手册的范围内。如果一个基本

复位由组件或适配器自主生成，并且如果平台向

组件/适配器，如果提供的电源超出了为外形规格或系统指定的限制，则组件/适配器必须生成对自身的基本复位。

·常规复位有三种不同的类型：冷、暖和热：

. 在向组件通电后，必须进行基本复位。这被称为冷复位。

. 在某些情况下，基本复位机制可以由硬件触发，而无需移除和重新施加到组件的电力。这被称为热重置。 本文档未指定生成热重置的方法。

. 存在用于跨链路传播常规复位的带内机制。 这称为热复位，在第4.2.4.9.2节中进行了描述。

软件有一种带内机制，可强制链路进入电气ID，从而“禁用”链路。禁用LTSSM状态在第www.example.com节中描述4.2.5.9，链路禁用控制位在第www.example.com节7.5.3.7中描述，下游端口遏制机制在[第6.2.10节中描述。](#bookmark44)禁用链路会导致下游组件进行热重置。

还要注意，数据链路层报告DL\_Down状态在某些方面与热复位相同-参见第2.9节。

·在退出任何类型的常规复位（冷、热或热）时，所有端口寄存器和状态机必须设置为本文档中指定的初始化值，粘性寄存器除外（参见第7.4节）。

. 请注意，从设备的角度来看，任何类型的常规复位（冷、热、热或DL\_Down）在事务层及以上具有与 常规PCI。

·在从基本复位退出时，物理层将尝试启动L链路（参见第4.2.5节）。

一旦链路上的两个组件都进入初始链路训练状态，它们将继续通过链路 物理层的初始化，然后通过VC 0的流控制初始化，使数据链路和事务层准备好使用链路。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

. 在VC 0的流控制初始化之后，TLP和DLLP可以跨链路传输。

在常规复位之后，某些设备可能需要额外的时间才能响应请求

他们收到。特别是对于配置请求，组件和设备必须以确定性的方式运行，以下规则解决了这一问题。

第一组规则涉及组件和设备的

·元件必须在基本复位结束后20 ms内进入LTSSM检测状态（链路训练在第4.2.4节中描述）。

. 注意：在某些系统中，链路上的两个组件可能会在不同的时间退出基本重置。每个组件必须遵守进入初始活动链路的要求

从其自身的角度来看，基本复位结束后20 ms内的训练状态。

·在完成链路训练（进入DL\_Active状态，参见第3.2节）时，组件必须能够接收和处理TLP和DLLP。

第二组规则解决了对系统的要求：

·为了允许组件执行内部初始化，系统软件必须在一个或多个设备的常规复位结束后等待指定的最短时间，然后才允许发出配置。 对这些设备的请求，除非使用了准备就绪机制（参见第6.23节）。

. 对于不支持超过5.0 GT/s的链路速度的下游端口，软件必须等待至少100 ms，然后才能向该端口下的设备发送配置请求。

. 对于支持链路速度大于5.0 GT/s的下游端口，软件必须等待

链路训练完成后，在将配置请求发送到

该端口下方的设备。 软件可以通过轮询数据链路层链路活动位或设置相关中断来确定链路训练何时完成（参见[第www.example.com节6.7.3.3）](#bookmark77)。

. 系统必须保证所有旨在软启动的组件在启动时可见，并准备好在应用程序结束后的最短时间内

在根复合体处的常规复位-如何完成这超出了本规范的范围

. 注：仅当软件启用CRS软件可见性时，软件才应使用100 ms的等待时间。 否则，可能会导致完成超时、平台超时或冗长的处理器指令暂停。 参见第2.3.1节中的配置请求状态更新实施说明。

· 设备常规复位后，设备必须能够在1.0 s内接收配置

如果请求有效，则请求并返回成功完成。这段时间与快速链接训练的完成方式无关。如果使用了准备就绪时间表机制（参见第6.23节），则该时间段可能会更短。

· 除非使用就绪性重置机制，否则根联合体和/或系统软件必须在设备的常规重置后至少允许1.0 s，如果设备未能返回有效配置请求的成功完成状态，则在确定设备损坏之前。这段时间与

链接训练完成。

注：此延迟类似于PCI/PCI-X指定的参数T rhf，旨在为需要自我初始化的设备提供足够的时间。

·当尝试对PCI Express/PCI（-X）桥后面的PCI或PCI-X总线段上的设备进行配置访问时，必须遵守定时参数T rhf a。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

对于该第二组规则，如果系统软件不具有对基本复位的状态的直接可见性（例如，热插拔;参见[第6.7节](#bookmark78)），软件必须将这些定时参数基于基本复位结束后发生的已知事件。

当链路正常运行时，以下规则适用：

· 无论出于何种原因，如果正常运行的链路断开，则事务和数据链路层将进入DL\_Inactive状态（参见第2.9和3.2.1节）。

· 对于任何根或交换机下游端口，设置与端口相关的桥控制寄存器的辅助总线复位位必须导致发送热复位（参见第4.2.4.9.2节）。

· 对于交换机，以下情况必须导致在所有下游端口上发送热复位

. 设置与上游端口相关的桥控制寄存器的辅助总线复位位

. 上游端口的数据链路层报告DL\_Down状态。在支持链路速度大于5.0 GT/s的交换机中，上游端口必须将每个下游端口的LTSSM引导到热复位状态，但不能将LTSSM保持在该状态。这允许每个下游端口

在其热重置完成后立即开始链路训练。 建议所有交换机都采用此行为。

. 在上游端口上接收热重置

本文档中指定了基本复位的某些方面，其他方面则特定于平台、外形和/或实现。特定的平台、形状因素或应用空间可能需要对系统组件之间的时序和/或顺序关系进行额外规范，以进行基本复位。为

例如，可能要求机箱内的所有PCI Express组件同时观察基本复位的断言和解除断言（在一定范围内）。在多机箱环境中，可能需要 指定包含根复合体的机箱是最后退出基本重置的机箱。

在提供电源和PERST#的所有情况下，必须定义以下参数：

·TPvperl-PERST# must在电源变为有效

·Tperst-当断言时，PERST#必须保持断言至少这么长时间

·Tfail-当电源变为有效时，必须在此时间

可以指定附加参数。

在提供参考时钟的所有情况下，必须定义以下参数：

·在任何提供的参考时钟稳定之后，T perst-perst- PERST#必须至少保持有效。

6.6.2功能电平复位（FLR）

FLR机制使软件能够以功能级粒度静默和重置端点硬件。三个示例使用模型说明了此功能：的好处

· 在某些系统中，有可能控制功能的软件实体将停止正常操作。 To prevent data corruption, it is necessary to stop all PCI Express and external I/O (not PCI Express) operations being performed by the Function.其他定义的复位操作不保证外部I/O操作将停止。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

· 在硬件从一个分区迁移到另一个分区的分区环境中，必须

确保硬件不保留先前分区的剩余“知识”，例如，委托给第一分区而不委托给第二分区的用户秘密信息。此外，由于功能范围广泛，因此有必要以功能独立的方式完成。

·当系统软件正在为函数建立软件栈并然后重建该栈时，有时需要在重建函数的软件栈之前将状态返回到未初始化状态。

FLR的实现是可选的（不是必需的），但强烈建议。

FLR适用于每个功能。只有目标函数受FLR操作的影响。 链路状态不得受FLR影响。

FLR修改本规范所述的功能状态，如下所示：

· 函数寄存器和特定于函数的状态机必须设置为本文档中指定的初始化值，但以下情况除外：

. 粘性类型寄存器（ROS、RWS、RW1CS）。定义为HwInit类型的寄存器

. 这些其他字段或寄存器：

在设备功能寄存器中记录捕获的时隙功率限制值

在设备功能寄存器中记录捕获的时隙功率限制S标度

设备控制寄存器中的最大有效负载大小

链路控制寄存器中的活动状态功率管理（ASPM）控制

在链路控制寄存器中重新读取完成边界

链路控制寄存器中的公共时钟配置

链路控制寄存器中的扩展同步

▪在链路控制寄存器中启用时钟电源管理

链路控制寄存器中的硬件自主宽度禁用

链路控制2寄存器中的硬件自主速度禁用

链路状态2寄存器中的链路均衡请求8.0 GT/s

在16.0 GT/s状态寄存器中的链路均衡请求16.0 GT/s

在链路控制3寄存器中启用较低SKP OS生成向量

辅助PCI Express扩展能力结构中的通道均衡控制寄存器

物理层16.0 GT/s E扩展能力结构中的

虚拟通道E扩展能力结构中的所有寄存器

多功能虚拟通道扩展功能结构中的

数据链路特性扩展能力结构中的所有寄存器

物理层16.0 GT/s扩展功能结构中的

物理层32.0 GT/s扩展能力结构中的所有寄存器

接收器扩展能力结构中通道边缘中的所有寄存器

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

· 强烈建议不要将以下寄存器重置为初始值：

. ARI扩展能力结构中的

. L1 PM子状态扩展能力结构中的所有寄存器 . 延迟容限报告功能结构中的所有寄存器。精密时间管理功能结构中的所有寄存器

本规范的未来修订版可能会将此建议更改为要求。

请注意，清除了使该功能能够在PCI Express上发起请求的控件，包括总线主机使能、MSI使能等，从而有效地使该功能在链路上变得静止。

请注意，与链路功能（包括物理和数据链路层中的链路功能）相关联的端口状态机器不会被FLR重置，并且VC 0在FLR之后保持初始化。

·在启动FLR之前，必须通过发送相应的Deassert\_INTx消息来解除功能断言的任何未完成INTx中断。

注意，当向多功能设备的功能发起FLR时，如果另一功能继续断言匹配的INTx，则将不发送Deassert\_INTx消息。

通过向启动功能级复位位写入1b启动FLR后，功能必须完成

100 ms内的FLR。如果软件在事务挂起位为1b时启动FLR，则软件不得初始化该函数，直到允许任何相关组件有足够的时间到达，或达到合理的确定性 任何剩余的完成永远不会到来。为此，建议软件允许设备上的完成时间与FLR前值所提供的时间一样长。如果在

Function when FLR was issued, then the delay is system dependent but must be no less than 100 ms. If Function

实现就绪状态（FRS-参见第6.23.2节），然后允许系统软件发布配置

在收到表示“准备就绪”的FRS请求后立即向职能部门提出请求，但这并不一定表示职能部门发起的未完成请求已完成。

请注意，在收到FLR后，设备功能可以清除所有事务状态，包括事务 挂起或将完成超时设置为默认值，以便所有挂起的事务在FLR执行期间超时。无论如何，事务挂起位必须在完成FLR时清除。

由于FLR修改了本规范未描述的功能状态（除了本规范描述的状态之外），

规范），有必要使用一组标准来指定FLR的行为，当应用于函数时，

表明该功能已满足FLR的要求。必须使用特定于功能的知识应用以下标准，以评估功能

·该函数不得在由该函数控制的任何外部接口上给出具有活动主机的初始化适配器的外观。终止外部接口上的活动所需的步骤超出了本规范的范围。

. 例如，网络适配器必须不响应于将需要主机系统进行适配器初始化或与活动主机系统交互的查询，但是被允许执行其被设计为在不需要主机初始化或交互的情况下执行的动作。如果网络适配器

包括在同一外部网络接口上运行的多个功能，则此规则仅影响与FLR重置的特定功能相关的方面。

·函数不得在其自身内保持软件可读状态，该软件可读状态可能包括与函数的任何先前使用相关联的秘密信息。分配给函数的主机内存不得 修改功能。

. 例如，具有可由主机软件直接或间接读取的内部存储器的函数必须清除或随机化该存储器。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·该功能必须返回到一种状态，使得该功能的PCI Express接口的正常配置

使其可由通常与功能相关的驱动程序使用当启动FLR时，目标功能必须表现如下：

·函数必须返回启动FLR操作的配置写入的完成，然后启动FLR。

·在进行FLR时：

. 如果请求到达，则允许静默地丢弃请求（在流控制信用的更新之后），而不将其记录或发信号通知为错误。

. 如果完成到达，则允许将完成作为意外完成来处理或静默地丢弃（在流控制信用的更新之后），而不将其记录或将其作为错误发信号通知。

. 虽然要求函数在上述时间限制内完成FLR操作，但后续函数特定初始化序列可能需要额外的时间。如果额外的时间 则该功能必须返回配置请求状态（CRS）完成状态 当在上述时间限制之后接收到配置请求时。本功能响应状态为完成的非CRS的配置请求后，不允许返回CRS

直到再次重置

执行说明

避免陈旧完成

FLR导致职能部门无法跟踪任何未完成的未发布请求。 任何随后到达的相应完成都被称为“过时”。如果软件在存在未完成请求时发出FLR，

然后重新启用该函数进行操作，而无需等待潜在的陈旧完成，

之后到达的完成可能会导致数据损坏，因为功能错误地认为属于自FLR以来发出的请求。

软件可以通过多种方式避免陈旧完成造成的数据损坏下面是一个可能的算法：

1. 与可能直接访问函数的其他软件一起执行FLR编译的软件，并确保在此算法期间不会发生此类访问。

2. 软件清除输入命令寄存器，禁止功能发出任何新请求。

3. 软件轮询设备状态寄存器中的事务挂起位，直到它被清除或直到它已经足够长，软件可以合理地确定与任何

剩余的未完成交易将永远不会到达。在许多平台上，事务挂起位通常会在几毫秒内清除，因此软件可能会选择在此初始期间进行轮询

使用紧密的软件循环。在极少数情况下，此时事务挂起位未被清除，软件将需要轮询更长的特定于平台的时间（可能为秒），因此软件 可以选择使用基于定时器的中断轮询机制来进行该轮询。

4. 软件启动FLR。

5. 软件等待100 ms。

6. 软件重新配置该功能并使其正常运行。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.7 PCI Express本机热插拔

PCI Express架构被设计成主动地支持电缆、附加卡和模块的热添加和热移除（“热插拔”）。PCI Express本机热插拔提供了一个“工具箱”机制，允许不同的用户/操作员 使用自一致的基础设施支持的模型。这些机制可用于有序

依赖于与操作系统的协调的添加/移除（例如，传统的PCI热插拔），以及在不与操作系统进行锁定步骤同步的情况下移除PCI。这个序列定义了 热插拔机制，并指定如果在系统中实现，热插拔的元素（如指示器和按钮）必须如何工作。

6.7.1热插拔元件

[表6-7](#bookmark79)列出了本规范中包含的支持热插拔模型的物理元件。 外形规格必须定义这些元素在该外形规格中的使用方式。对于给定的外形规格，

可能的是，仅需要可用热插拔元件中的一些，或者甚至不需要这些元件中的

必需的.在所有情况下，形状因素规范必须定义通过选择所包含的元素而对系统或用户施加的所有假设和限制。 旨在仅与选定的形状因子一起使用的硅组件实现被允许仅支持相关联的形状因子所需的那些元件。

表6-7热插拔要素

|  |  |
| --- | --- |
| 元件 | 目的 |
| 指标 | 显示插槽的电源和连接状态 |
| 手动操作 保持闩锁（MRL） | 将适配器固定到位 |
| MRL传感器 | 允许端口和系统软件检测正在打开的MRL |
| 机电联锁 | 防止适配器从插槽 |
| 注意按钮 | 允许用户请求热插拔操作 |
| 软件用户界面 | 允许用户请求热插拔操作 |
| 插槽编号 | 提供插槽的视觉识别 |
| 功率控制器 | 由软件控制的电子部件，用于控制插槽或适配器的电源，并监控电源的故障情况 |
| 带外存在检测 | 确定插槽中适配器的物理存在的方法，不依赖于物理层 |

<6.7.1.1>指标

有两个指示器：电源指示器和注意指示器。每个指示器处于以下三种状态之一：开， 关闭或闪烁。热插拔系统软件通过写入与指示器相关联的命令寄存器而具有对指示器状态的排他控制（除了下面指出的一个例外）。指示器要求必须包含在所有外形规格中。对于给定的形状因子，指示符可以是必需的或可选的或根本不适用。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

支持热插拔的端口控制指示灯的闪烁频率、占空比和相位。闪烁指示器必须 工作频率在1和2 Hz之间，占空比为50%（+/-5%）。闪烁指示器不需要在端口之间同步或同相。

指示灯可以位于机箱或适配器上（有关指示灯位置要求，请参阅相关的外形规格）。不考虑物理位置，指标的逻辑控制由

链路上上游组件的下游端口

下游端口不得改变指示器的状态，除非软件命令这样做，除非

能够检测固定电源故障的平台（仅在实现电源控制器时相关）。在他

如果出现固定电源故障，则允许平台超控下游端口并强制电源指示灯亮起（作为不应移除适配器的指示）。系统软件对固定故障的处理

可选的并且在本说明书中没有描述。因此，平台供应商必须确保此功能（如果实现）通过其他软件、平台文档或其他方式解决。

6.7.1.1.1注意指示器

注意指示灯必须为黄色或琥珀色，表示存在操作问题或正在识别热插拔插槽，以便操作人员可以轻松定位。

表6-8注意指示器状态

|  |  |
| --- | --- |
| 指示器外观 | 意义 |
| 关闭 | 正常-正常操作 |
| 对 | 注意-此插槽 |
| 闪烁 | 定位-根据用户 |

注意指示灯关闭

处于关闭状态的注意指示灯表示适配器（如果存在）和热插拔插槽都不需要注意。

注意指示灯亮起

注意指示灯处于开状态表示适配器或插槽存在操作问题

操作问题是阻止适配器继续操作的情况。操作系统或

其它系统软件确定特定条件是否阻止适配器的继续操作

注意指示灯是否适当。操作问题的示例包括与外部布线、适配器、软件驱动程序和电源故障相关的问题。通常，处于开启状态的注意力指示器

指示尝试操作失败或发生意外事件

注意指示器不用于报告在验证热插拔请求

操作验证是一个术语，适用于系统软件执行的任何检查，以确保所请求的

操作是可行的，允许的，不会引起问题。验证失败的示例包括拒绝

执行热插拔操作的许可、功率预算不足以及在接受热插拔请求之前可能检测到的其它条件。

注意指示灯闪烁

闪烁的注意指示器指示系统软件正在识别此插槽以供操作员查找。此行为由用户控制（例如，从软件用户界面或管理工具）。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.7.1.1.2电源指示灯

电源指示灯（必须为绿色）指示插槽的电源状态。[表6-9](#bookmark80)列出了电源指示灯状态。

表6-9电源指示灯状态

|  |  |
| --- | --- |
| 指示器 外观 | 意义 |
| 关闭 | 电源关闭-允许插入或移除数据。 |
| 对 | 通电-不允许插入或拔出适配器。 |
| 闪烁 | 电源转换-正在进行热插拔操作，不允许插入或取出适配器。 |

电源指示灯关闭

电源指示灯处于关闭状态表示允许插入或取出适配器。 如果形状因素要求，插槽的主电源关闭。注意，取决于形状因子，其它功率/信号可以

即使主电源关闭且电源指示灯熄灭，也会保持亮起。在使用[CEM]形状因子的示例中，如果平台为热插拔插槽提供Vaux，并且MRL关闭，则即使电源指示灯关闭，MRL切换的任何信号也会连接到插槽。当MRL为

打开当插槽未通电和/或允许插入或拔出适配器时，系统软件必须关闭插槽的电源指示灯。有关详细信息，请参阅相应的外形规格。

电源指示灯亮起

处于打开状态的电源指示灯表示热插拔操作已完成，插槽的主电源已打开，并且不允许插入或移除适配器。

电源指示灯闪烁

闪烁的电源指示灯表示插槽正在通电或断电，不允许插入或取出适配器。

当按下注意按钮或通过热插拔软件界面启动热插拔操作时，闪烁的电源指示灯也向操作员提供视觉反馈。

<6.7.1.2>手动锁闩（MRL）

MRL是一种手动操作的保持机制，可将数据夹固定在插槽中，并防止用户

移除设备。MRL将适配器牢固地固定在插槽中，以便可以连接电缆，而不会产生间歇性接触的风险。在平台中允许同时抑制两个或多个适配器的MRL

不提供MRL传感器。

<6.7.1.3>MRL传感器

MRL传感器是一种开关、光学设备或其他类型的传感器，其向

下游端口。当MRL完全关闭时，MRL传感器报告关闭，而在所有其他时间（即，如果MRL完全打开或处于中间位置）则报告打开。

如果为插槽安装了电源控制器，则当MRL传感器指示MRL打开时，必须自动从插槽中断开插槽主电源。如果Vaux和SMBus等信号由MRL切换，则这些

当MRL传感器或指示MRL打开时，信号必须自动从插槽

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

当MRL传感器指示MRL已再次关闭时，恢复到插槽。请参考相应的外形规格，以识别由MRL切换的信号（如果有）。

请注意，热插拔控制器不会根据MRL传感器的变化自动改变电源指示灯或注意指示灯的状态。

执行说明

MRL传感器处理

在没有MRL传感器的情况下，对于某些形状因素，可以使用带外存在检测来处理 切换信号。在这种情况下，当带外存在检测指示插槽中没有适配器时，交换信号将自动从插槽中删除。

如果在热插拔控制器上没有相应的MRL传感器输入的情况下实现MRL传感器，则建议将MRL传感器路由到热插拔控制器的电源故障输入。 这允许在MRL打开时关闭活动适配器的电源。



<6.7.1.4>机电联锁

机电互锁是一种将适配器或MRL物理锁定到位的机制，直到系统软件将其释放。机电联锁的状态由软件设置，除非响应后续软件命令，否则不得改变。特别是，机电联锁的状态必须保持，

当热插拔插槽的电源被移除时。

机电互锁的当前状态必须始终反映在插槽状态寄存器的机电互锁状态位中，该状态必须在任何命令更改后200 ms内更新。在发出命令切换机电联锁状态后，软件必须等待至少1秒，然后再发出另一个命令。 可以发出切换状态的命令。系统可以可选地扩展对互锁的控制，以提供物理 适配器的安全性。

<6.7.1.5>注意按钮

注意按钮是一个瞬时接触按钮开关，位于每个热插拔插槽附近或

由用户按下以在该插槽上启动热插拔操作的适配器。无论按钮的物理位置如何，信号都由与下游端口相关的热插拔硬件处理并指示给软件 对应于S批次。

注意按钮必须允许用户在不考虑按钮物理位置的情况下启动热添加和热删除操作。

如果存在，电源指示灯将通过闪烁向操作人员提供视觉反馈（如果系统软件接受由注意按钮发起的请求）。一旦动力指示器开始闪烁，则存在5秒的中止间隔，在此期间，第二次按下注意按钮将取消操作。

如果由注意按钮n启动的操作因任何原因失败，建议系统软件通过软件用户界面显示错误消息，解释失败原因，或将错误消息添加到系统日志中。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.7.1.6>软件用户界面

系统软件提供了一个用户界面，允许启动热插入和热移除，并允许监控占用的插槽。对热插拔用户界面的详细讨论是特定于操作系统 因此超出了本文件的范围。

在具有多个热插拔插槽的系统上，系统软件必须允许用户在每个插槽上启动

独立于所有其他插槽的状态。因此，允许用户使用软件用户界面或注意按钮在一个插槽上启动热插拔操作，而另一个插槽上的热插拔操作正在进行中，而不管使用哪个界面启动第一个操作。

<6.7.1.7>插槽编号

物理插槽标识符（如[PCI-Hot-Plug-1.1]第1.5节中所定义）由可选的机箱号和插槽的物理插槽号组成。 物理插槽号是插槽的机箱唯一标识符。系统软件

从端口中的寄存器确定物理插槽号。机箱编号0是为主机架保留的。的 其他机箱的机箱编号必须是从PCI到PCI桥的机箱编号寄存器获得的非零值（请参见[ PCI-to-PCI-Bridge-1.2 ]，第13.4节）。

无论与每个插槽相关联的外形尺寸如何，每个物理插槽号在机箱内都必须是唯一的。

<6.7.1.8>电源控制器

功率控制器是由一个或多个分立组件组成的元件，其在软件的控制下动作以将热插拔槽的功率状态设置为适合于特定形状因子。电源控制器还必须

监控插槽的主电源轨和辅助电源轨（如果支持）上出现的电源故障情况（如相关外形规格中所定义）。

如果不存在电源控制器，则必须由热插拔控制器自动设置热插拔插槽的电源状态，以响应插槽中适配器的变化。

电源控制器独立监控主电源和辅助电源故障。如果电源控制器检测到主电源

如果热插拔插槽出现电源故障，则必须自动设置其内部主电源故障锁存器并断开主电源

从热插拔插槽（不影响辅助电源）。类似地，如果电源控制器检测到热插拔插槽上的辅助电源故障，则必须自动设置其内部辅助电源故障插槽，并从

热插拔插槽（不影响主电源）。只要电源故障条件保持锁定，插槽的电源就必须保持关闭，而不管软件是否写入以打开热插拔插槽的电源。当软件关闭热插拔插槽的电源时，主电源故障锁存器将被清除。辅助电源故障闭锁的机理

清除是形状因子特定的，但是一般需要从热插拔槽移除辅助电源。例如，一个形状因子可以在插槽的MRL打开时移除辅助电源，而另一个形状因子可能需要从插槽物理地移除适配器。有关具体要求，请参阅相关的外形规格。

由于时隙控制寄存器中的电源控制器控制位反映的是最后写入的值，而不是实际状态 这意味着在电源故障条件下，在电源控制器控制位的值和槽的电源状态之间可能存在不一致。为了确定插槽是否由于电源故障而关闭，

软件必须使用电源故障软件通知来检测电源故障。要确定请求的上电操作是否失败，软件必须使用第www.example.com节中所述的热插拔插槽上电超时机制[6.7.3.3。](#bookmark82)

软件不得假定写入插槽控制寄存器以更改热插拔插槽的电源状态会导致立即发生电源状态转换。打开电源后，软件必须等待数据链路层状态更改事件，如第www.example.com节所述[6.7.3.3](#bookmark83)。关闭电源后，软件必须等待至少1秒才能采取任何行动

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

它依赖于从热插拔插槽中移除的电源。例如，在完成1秒等待期之前，不允许软件关闭电源指示器（如果存在）或尝试打开电源控制器。

6.7.2按热插拔元素关联

本节中描述的寄存器按热插拔元素分组，以传送与

实现每一个元素。与实现热插拔功能的每个下游端口关联的注册表字段 插槽位于PCI Express能力结构中的设备能力、插槽能力、插槽控制、插槽状态和插槽上限能力2寄存器中（参见第7.5.3节）。报告热插拔元件存在的寄存器

与适配器上的设备功能相关联的功能位于设备功能寄存器中（也位于PCI Express功能结构中）。

<6.7.2.1>注意按钮寄存器

注意按钮存在（插槽功能寄存器和设备功能寄存器）-此位表示注意按钮是否存在 按钮由机箱（插槽功能寄存器）或适配器（设备功能寄存器）进行电气控制。

按下注意按钮（插槽状态寄存器-按下由底盘电控的注意按钮时，该位被设置。

按下注意按钮启用（插槽控制寄存器-设置时，该位启用有关按下注意按钮事件的软件通知（请参阅[第6.7.3.4节）](#bookmark85)。

<6.7.2.2>注意力指标寄存器

注意指示器存在（插槽功能寄存器和设备功能寄存器）-此位指示注意指示器是由机箱（插槽功能寄存器）还是由适配器（设备功能）进行电气

Register）。

注意指示灯控制（插槽控制寄存器）-写入时，将由机箱电气控制的注意指示灯设置为写入状态。

<6.7.2.3>电源指示寄存器

电源指示灯存在（插槽功能寄存器和设备功能寄存器）-该位指示电源指示灯是否由机箱（插槽功能寄存器）或适配器（设备功能寄存器）进行电气控制。

电源指示灯控制（插槽控制寄存器）-写入时，将由机箱电气控制的电源指示灯设置为写入状态。

<6.7.2.4>电源控制器寄存器

电源控制器存在（插槽功能寄存器）-此位表示是否实现了电源控制器

电源控制器控制（插槽控制寄存器）-根据写入的值打开或关闭电源控制器。检测到电源故障（插槽状态寄存器）-当插槽或适配器检测到电源故障时，该位置1。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

电源故障检测使能（插槽控制寄存器）-设置时，此位使能电源故障事件的软件通知（参见[第www.example.com节6.7.3.4）](#bookmark86)。

<6.7.2.5>在线检测寄存器

带内PD禁用支持（插槽功能2寄存器）-此位指示插槽是否支持禁用带内存在检测，这允许独立于带内存在检测状态报告带外存在检测状态。

带内PD禁用（时隙控制寄存器）-设置时，该位禁用带内存在检测机制，使其不影响存在检测状态位，从而允许该位专用于报告带外存在检测。

存在检测状态（插槽状态寄存器）-此位表示插槽中存在数据。

存在检测已更改（插槽状态寄存器）-当检测到存在检测状态更改时，该位置1

存在检测更改启用（插槽控制寄存器）-设置时，此位启用存在检测更改事件的软件通知（参见[第www.example.com节6.7.3.4）](#bookmark87)。

<6.7.2.6>MRL传感器寄存器

MRL传感器存在（插槽功能寄存器）-此位指示是否使用MRL

MRL传感器已更改（插槽状态寄存器）-当MRL传感器状态值更改时，该位置1

MRL传感器更改使能（插槽控制寄存器）-设置时，该位启用MRL传感器更改事件的软件通知（参见www.example.com[节6.7.3.4）](#bookmark88)。

MRL传感器状态（插槽状态寄存器）-此寄存器报告MRL传感器的状态（如果已实现）。

<6.7.2.7>机电互锁寄存器

机电联锁存在（插槽功能寄存器）-该位指示是否实施机电联锁。

机电联锁状态（插槽状态寄存器）-此位反映机电联锁的当前状态。

机电互锁控制（插槽控制寄存器）-当设置为1b时，此b将切换机电互锁的状态。

<6.7.2.8>命令完成的寄存器

无命令完成支持（插槽功能寄存器）-此位设置为1b时，表示当热插拔控制器完成发出的命令时，此插槽不生成软件通知。

命令完成（Slot状态寄存器）-当热插拔控制器完成发出的命令并准备好接受下一个逗号ND时，该位被设置。

命令已完成启用（Slot控制寄存器）-设置时，当热插拔控制逻辑完成命令时，此位启用软件通知（参见[第www.example.com节6.7.3.4](#bookmark89)）。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.7.2.9>端口功能和插槽信息寄存器

已实现插槽（PCI Express功能寄存器）-设置时，此位表示与此下游端口相关的链路已连接到插槽。

物理插槽号（插槽功能寄存器）-此硬件初始化字段指示连接到端口的物理插槽号。

热插拔能力（插槽能力寄存器）-设置时，此位表示此插槽能够支持热插拔。 热插拔惊喜（插槽功能寄存器）-设置时，此位表示已为此插槽启用了用于处理热插拔删除的热插拔惊喜机制。 参见[第6.7.6节。](#bookmark90)

<6.7.2.10>热插拔控制注册

热插拔中断使能（插槽控制寄存器）-设置时，该位使能在启用的热插拔事件上生成热插拔中断。

6.7.3 PCI Express热插拔事件

具有热插拔功能的下游端口支持以下热插拔事件：

·插槽事件：

. 按下注意按钮。检测到电源故障

. MRL传感器已更换

。存在检测已更改

·命令完成前夕nts

· 数据链路层状态更改事件

这些事件中的每一个都有一个状态字段，该字段指示事件已经发生，但还没有被sed处理 软件，以及启用字段，其指示事件是否被启用用于软件通知。 某些事件还有一个capability字段，用于指示端口是否支持该事件类型。按事件类型对这些字段进行的分组见[第6.7.2节](#bookmark84)，每个单独字段的描述见第7.5.3节。

<6.7.3.1>老虎机活动

具有热插拔功能的下游端口会监控其控制的插槽是否出现上面列出的插槽事件。当其中一 当检测到这些插槽事件时，端口通过设置与事件相关联的状态字段来指示事件已经发生。在这一点上，前夕是挂起，直到软件清除状态字段。

一旦某个插槽事件在特定插槽上挂起，该插槽上所有该类型的后续事件都将被忽略，直到

事件被清除。端口必须继续监视插槽中的所有其他插槽事件类型，并在发生时报告它们。

如果通过关联的启用字段启用，插槽事件必须生成软件通知。如果所述事件不是

如相关功能字段所示，端口上不支持，软件不得启用事件的软件通知。将此通知报告给软件的机制在第www.example.com节中描述[6.7.3.4。](#bookmark92)

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.7.3.2>命令完成事件

由于改变某些热插拔元件的状态可能不会立即发生，PCI Express支持热插拔命令和命令完成事件。所有支持热插拔的端口都需要支持热插拔命令，如果报告了该功能，则还需要逗号和已完成事件。

软件通过发出一个写事务来向具有热插拔功能的下游端口发出一个命令，

端口插槽控制寄存器的一部分对插槽控制寄存器的单次写入被视为单次

命令，即使写入影响插槽控制寄存器中的多个字段。为了响应此事务，端口必须执行请求的操作，然后为命令完成事件设置关联的状态字段。即使在发出命令时状态字段已经设置，端口也必须正常处理命令。如果单个命令导致启动多个操作，则执行这些操作的顺序为

未指明。 与单个命令执行相关的所有操作不得超过1秒。

如果不支持命令完成事件，如插槽能力寄存器的无命令完成支持字段中的值f1b所示，则具有热插拔能力的端口必须处理以端口的插槽控制寄存器的任何部分为目标的写入事务，而不依赖于先前的插槽控制写入。软件允许发布 multipleSlot控制按顺序写入，在写入站点之间没有任何延迟。

如果支持命令完成事件，则软件必须在发出下一个命令之前等待命令完成。然而，如果在命令执行的1秒限制之后未设置状态字段，则软件

允许重复该命令或发出下一个命令。如果软件在端口

完成前一个命令的处理，并且在1秒的时间限制到期之前，允许端口接受或丢弃写入。这样的写入被认为是编程错误，并可能导致差异

插槽控制寄存器和热插拔元件状态之间的关系。为了从这种编程错误中恢复并使控制器返回到一致状态，软件必须向符合

命令完成规则。

如果通过相关的启用字段启用，则命令的完成必须生成软件通知。

此规则的例外情况是，写入插槽控制寄存器后出现的命令会禁用

命令完成事件的软件通知。 这样命令必须如上所述地被处理，但不能生成软件通知。

<6.7.3.3>数据链路层状态更改事件

数据链路层状态已更改事件指示链路状态寄存器中数据链路层链路活动位的状态已更改。支持数据链路层状态更改事件和软件通知 支持热插拔的下行链路端口需要事件。如果支持此事件，则端口将设置状态字段

当数据链路层链路活动位中的值改变时，

此事件允许软件间接确定新热插拔适配器的通电时间。

在数据链路层链路活动位读取1b之后，软件必须等待100 ms，然后才能启动对热添加设备的配置访问（参见[第6.6节）](#bookmark75)。数据链路层链路活动位读取1b后，软件必须允许1 在允许确定未能返回有效的成功完成的热插拔设备

配置请求是损坏的设备（参见[第6.6节）](#bookmark75)。

数据链路层状态已更改事件必须在启动热插入的事件发生后1秒内发生。如果

如果支持电源控制器，则超时间隔是从软件启动对插槽控制寄存器的写入以打开电源开始测量的。如果不支持电源控制器，则从存在开始测量超时间隔

检测插槽事件。 如果数据链路层状态更改事件未在1秒内发生，则软件在热添加操作中将全部超时。软件在这样的超时之后采取的动作是特定于实现的。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

<6.7.3.4>热插拔设备的软件通知

支持热插拔的下游端口必须支持在热插拔事件中生成中断。中所述 在<6.7.3.1>和<6.7.3.2>部分，每个热插拔事件都有一个用于生成中断的使能位和一个状态位，该状态位指示事件何时发生但尚未被软件处理。re也是插槽控制寄存器中的热插拔启用位，用作所有热插拔事件的主机启用/禁用位。

如果使用INTx消息使能端口的电平触发中断信号，则只要满足以下条件，就必须断言虚拟INTx

·命令寄存器中的禁用位设为0b。

·插槽控制寄存器r中的热插拔启用位设置为1b。

·插槽状态寄存器中的至少一个热插拔事件状态位及其在插槽控制寄存器中的关联使能位均设置为1b。

请注意，在请求服务时，同一功能内的所有其他中断源将断言同一虚拟INTx线。

如果使用MSI或MSI-X使能端口的边沿触发中断信号，则每次以下条件的逻辑AND从“0”转换为“真”时，必须发送中断消息

·相关联的向量未被掩蔽（如果MSI不支持PVM，则不适用）。

·插槽控制寄存器r中的热插拔启用位设置为1b。

·插槽状态寄存器中的至少一个热插拔事件状态位及其在插槽控制寄存器中的关联使能位均设置为1b。

请注意，PME和热插拔事件中断（当两者都实现时）始终共享相同的MSI或MSI-X向量，如PCI Express功能寄存器中的“PME消息编号”字段所示。

当发生热插拔事件时，端口可以选择性地发送MSI，而中断生成被禁用，并且中断生成随后被启用。

如果相关的表单事实或规范要求生成尾流

支持在系统处于睡眠状态或端口处于设备状态D1、D2或D3 Hot时发生热插拔事件时生成唤醒事件（使用PME机制）。

软件通过启用事件的软件通知，使热插拔事件能够生成唤醒事件，

在[6.7.3.1部分](#bookmark91)中描述。请注意，为了使软件在保持唤醒的同时禁用中断生成，

如果启用了生成，则必须清除热插拔可扩展位。对于支持唤醒生成的外形规格，如果出现以下三种情况，则必须生成唤醒事件：

·已启用事件的状态寄存器从清除转换为设置

·端口处于设备状态D1、D2或D3热，并且

·端口的电源管理控制/状态寄存器中的PME\_En

请注意，热插拔控制器代表热插拔设备生成唤醒，并且该设备不需要具有辅助（或主）电源。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.7.4系统固件中间件（SFI）支持

系统固件中介（SFI）功能是下游端口的可选标准功能。一些SFI SFI功能集中在热插拔插槽上，如插槽功能寄存器中的热插拔功能位被设置所指示的，而一些SFI功能在该上下文之外是有用的。如果下游端口支持SFI功能 结构，必须设置以下位：

·链路能力寄存器中的数据链路层链路活动报告能力位

·链路能力2寄存器中的DRS支持

·设备能力寄存器中的ERR\_COR子类能力

<6.7.4.1>SFIERR\_COR事件信令

SFI功能不支持生成INTx或MSI/MSI-X中断，因为该功能旨在由系统固件使用。

具有SFI的下游端口必须支持ERR\_COR信令，而不管它是否支持高级错误

报告（AER）或不报告。SFIERR\_COR事件信令由SFI控制寄存器中的SFI OOB PD更改使能、SFIDLL状态更改使能和SFI DRS信令使能位独立使能。这些事件由SFI状态寄存器中的SFI OOB PD Changed、SFI DLLStateChanged和SFI DRS Received位指示。

如果设备控制寄存器中的可纠正错误报告启用位被设置，则端口必须发送ERR\_COR 每次满足其中一个启用条件时都显示消息。SFIERR\_COR事件信令不得设置设备状态寄存器中的可纠正错误检测位，因为此事件不作为错误处理。

执行说明

DPC DL\_Active与SFI DLL状态的ERR\_COR信令已更改

DPC实现用于DL\_Active的ERR\_COR信令，而SFI实现用于SFI DLL状态改变的ERR\_COR信令，这是相关但不相同的条件。当数据链路

链路状态寄存器中的层链路活动位从0b变为1b，并且此位可由SFI DLL SFI控制寄存器中的状态掩码位。当SFIDLL状态位 在SFI状态寄存器中，通过变为设置或变为清除来改变其值，并且该条件总是基于实际的数据链路层状态。



<6.7.4.2>SFI下游端口过滤（DPF）

下游端口过滤（DPF）是一种机制，其中下游端口可以在其下方的目标组件处处理指定的请求TLP，就好像链路处于DL\_Down中一样。 参见第2.9.1节。

DPF有两种过滤请求TLP的模式，目标是下游端口以下的组件。 第一模式过滤所有这样的请求TLP;第二模式仅过滤配置请求TLP。不得过滤其他TLP，或

被DPF拦截。

DPF的一个关键用例是保证异步系统软件活动（如总线扫描）不会

无意中将配置请求发送到常规复位后尚未就绪的设备，因为此类访问会导致未定义的硬件行为。 参见[第6.6.1节。](#bookmark76)

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

DPF的另一个关键用例是支持固件优先功能，当通知系统固件发生故障时， 在使设备对操作系统可见之前，配置新添加的设备。为此 在这种使用情况下，[SFI CAM](#bookmark94)机制使得下游端口本身能够生成以下游组件为目标的配置请求TLP，并且这些TLP不被DPF机制过滤或阻止。参见[6.7.4.3部分，](#bookmark95)

部分7.9.21.5和部分7.9.21.6。

<6.7.4.3>SFI CAM

SFI配置访问方法（CAM）为SFI感知系统固件提供了一种方法，以便在DPF被

启用. SFI[CAM](#bookmark95)始终启用。

要使用[SFI CAM](#bookmark95)，软件首先写入SFI CAM地址寄存器，指定目标配置地址。然后，软件读取或写入SFI CAM数据寄存器，以使被代理的配置请求被生成并被发送到下行链路组件。

以下规则适用：

·用于被代理的配置请求的所有TLP字段与以SFI CAM数据寄存器为目标的配置请求中的那些TLP字段相同，除了以下例外：

. 目标总线号、设备号和功能号来自SFI CAM地址寄存器。

. 扩展寄存器号和寄存器号来自SFI CAM地址寄存器。.再生LCRC。

. 如果存在，则ECRC被再生。

·[SFI CAM](#bookmark95)不得将完成验证机制应用于请求。

·系统固件必须确保在其写入SFI CAM地址寄存器的时间与其

如果SFI CAM数据寄存器的后续读取或写入完成，则没有其他线程修改SFI CAM地址寄存器;否则，结果未定义。

· 如果存在与被代理的配置请求相关联的检测到的错误，则这是与实现SFI CAM的下游端口[相关联的报告的错误](#bookmark95)（参见[示例6.2）](#bookmark1)。

·流向上游的完井液必须通过下游管线，且不得更改。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

SFI CAM地址和数据寄存器的串行化使用

如上所述，系统固件必须确保在其写入SFI CAM地址寄存器的时间与其随后完成SFI CAM数据寄存器的读取或写入的时间之间，没有其它线程修改SFI CAM

地址寄存器。例如，可以使用semaphore或其他同步机制来确保该串行化。

对于其中将处理器存储指令有效地发布到配置空间的平台，软件仍必须确保在另一软件线程修改SFI CAM之前完成所得到的配置写入

数据寄存器。在这样的平台上，用于确定配置写入何时完成的机制是特定于平台的。

在适当的序列化条件下，[SFI CAM](#bookmark95)可正确处理导致CRS的配置请求

完成，即使根联合体自动将配置请求作为新请求重新发出。重新发出的配置请求将再次被发送到SFI CAM数据寄存器，并且相关联的

下游端口将再次生成针对下游组件的配置请求。只要在配置请求完成之前，SFI CAM地址寄存器

序列可以无限重复，直到返回非CRS完成或发生完成错误。

当启用CRS软件可见性时，[SFI CAM](#bookmark95)仍能正确处理导致CRS完成的配置请求。任何具有CRS完成状态的完成流回原始的排序器， 根据CRS软件可见性语义的要求处理它们。 参见第2.3.2节。



执行说明

在SFI CAM中使用USB总线编号

当下游端口启用DPF时，SFI感知系统固件可以使用[SFI CAM](#bookmark95)来配置 并访问端口下面的子层次结构，而其他软件不能这样做。巴士号码 端口下方的配置通常对其他软件不可见，配置用于端口下方的总线号应限于已分配给端口的总线号，因为通过端口进入上游的TLP 包含具有配置的总线编号的ID。如果检测到任何错误并使用这些TLP进行记录，则总线

其他软件可能会看到编号，如果它们与系统中其他地方使用的总线编号重叠，就会造成混乱。



<6.7.4.4>SFIInteractions with ReadinessNotifications

SFI功能能够屏蔽接收到的设备就绪状态（DRS）消息的验证，并仿真 他们被接收。当使用SFI的下游端口过滤（DPF）机制来阻止操作系统对D下游端口以下的设备或子层次结构的可见性时，该功能是有用的。

规则：

·当SFI DRS屏蔽位被设置时，链路状态2寄存器值中的DRS消息接收位必须为0b。

·SFI DRS接收位必须始终指示DRS消息接收条件的实际状态。

·当SFI DRS掩码位为清除并且1b被写入SFI DRS触发位时，下游端口必须表现得好像接收到DRS消息一样。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

针对设备就绪性的SFI透明优化

某些设备在执行热添加操作后可能需要比允许的时间更长的时间才能进入“备份就绪”状态。 参见[第6.6节。1.](#bookmark76)

如果系统固件缺少这样的设备，则它可以使用SFI DPF机制来阻止新添加的设备的操作系统可见性，等待必要的时间量以使设备变为可扩展就绪，并且 然后将设备暴露给操作系统。

为了避免操作系统不必要地等待新暴露的设备的额外时间，

当系统固件变为“准备就绪”时，系统固件可以使用SFI DRS触发位来使下游端口

仿真DRS消息的接收。然后，支持DRS的操作系统可以立即发现并配置新暴露的设备。

新曝光的设备本身不一定需要具备DRS功能。由于上行端口被明确允许发送DRS消息，即使当其DRS支持位为清除时，它上面的下行端口也可以

合法地模拟从其接收DRS消息，即使其不能发送DRS消息。

还应当注意，在系统固件意识到设备提前变为备转就绪的情况下，系统固件可以使用SFI DRS触发机制将其暴露给操作系统。

虽然SFI不适用于操作系统软件，但建议在支持SFI的平台中使用的操作系统实现对DRS的支持，以便系统作为一个整体可以享受这种优化的设备就绪时间的好处。



执行说明

SFI DPF和功能就绪状态（FRS）消息

[下游端口过滤（DPF）不](#bookmark93)影响FRS消息的生成或传播。当设备准备好作为DMAC热添加操作的一部分时，设备不会生成FRS消息。但是，如果系统

当固件在设备上执行导致FRS事件的操作时，所产生的FRS消息可以对操作系统可见。 参见第2.2.8.6.4节和第6.23.2节。



<6.7.4.5>SFI抑制热插拔意外功能

如果插槽支持热插拔惊喜（HPS）功能，如插槽上限能力寄存器中的热插拔惊喜位被置位所指示的，则SFI控制寄存器中的SFI HPS抑制位可用于强制热插拔惊喜位被清除，并禁用相关的热插拔惊喜功能。

当下游端口/插槽组合同时支持HPS和下游端口时，HPS抑制非常有用

遏制（DPC）。不建议将DP C与HPS并发使用，因此如果插槽启用了HPS功能，则不应启用DPC。如果软件希望使用DPC，软件应首先设置SFI HPS抑制位，以便

禁用HPS功能，允许DPC正常工作。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

热插拔惊喜功能的软件协商

假设系统固件拥有SFI能力结构，建议向后

与旧的操作系统兼容，默认情况下在插槽上启用热插拔惊喜功能

支持移除P2P。然后，如果插槽也支持DPC并且操作系统希望使用它，则操作系统将请求通过系统固件抑制HPS，并且系统固件将

确定是否设置或清除SFI HPS抑制位。



6.7.5热插拔

某些系统包括支持热插拔的根端口和交换机（在具有本机热插拔支持的ACPI兼容操作系统可用之前发布），可以使用ACPI固件传播热插拔事件。固件

如果使用具有本机支持的操作系统，则必须禁用对热插拔寄存器的控制平台形式，

提供ACPI固件来传播热插拔事件还必须提供将控制转移到操作系统的机制。该方法的细节在PCI固件规范中描述。

6.7.6异步删除

异步删除是指由于错误遏制而删除适配器或禁用下游端口

而不事先警告操作系统。这与有序移除形成对比，其中移除操作是

通过用户动作和系统管理设施的明确定义的序列以与操作系统锁步的方式执行。例如，用户按下注意按钮以请求来自操作系统的许可。

系统移除适配器，但是直到操作系统已经停止对适配器的活动并授予移除许可，用户才实际从插槽中移除适配器。

由于PCI Express层次结构或操作系统的其余部分必须在PCI Express层次结构或操作系统的其余部分之前移除PCI Express，

意识到该事件，除了标准PCI热插拔所需的特殊说明之外，还需要特殊说明。 本节概述了PCI Express事件，这些事件可能作为PCI Express删除的副作用发生，以及处理PCI Express删除的机制。

由于对于与插槽相关联的下游端口的物理和数据链路层而言，可纠正错误可能是意外的，因此可纠正错误可能被报告为事件的副作用（即，接收器错误、不良TLP和不良错误）。

DLLP）。如果报告了这些错误，软件应将其作为此事件的预期部分进行处理。

调度员可能会遇到与已接受的请求相关联的完成超时，但永远不会

由删除的完成者完成。在此上下文中产生的任何完成验证错误都应作为此事件的预期部分进行处理。

异步移除可能导致下游端口中从DL\_Active转换为DL\_Down。这种转变可能导致

在一个意外的错误。此外，PCI Express架构域中的完成者可能不会立即意识到这种转换，并继续向必须由PCI Express架构域处理的已删除完成者发出请求。

与插槽关联的下游端口

下游端口遏制（DPC）或热插拔惊喜（HPS）机制可用于支持作为整个热插拔架构的一部分的热插拔移除。相关参考模型见附录I。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

异步热插拔的热插拔惊喜机制

热插拔惊喜（HPS）机制，如插槽功能寄存器中的热插拔惊喜位被设置所指示的，不建议用于热插拔。DPC是建议采用的支持业务流程的机制。 热插拔。 [6.7.4.4](#bookmark96)有关支持这两种机制的插槽的指南，请参见第www.example.com节。

随着HPS的去除，使用HPS有严重的缺点。不可纠正的错误，除了那些固有的

使链路中断需要被配置为使系统崩溃、由软件异步处理或被忽略。这些错误包括与Posted MemoryWrites、带有中毒数据的TLP和完成超时相关的无法纠正的错误。软件忽略或异步处理的不可纠正的错误可能会使其

驱动程序无法确定哪些高级操作成功完成，而哪些没有成功完成。

DPC提供了一个强大的机制来支持去除病毒。TLP流完全停止时，

触发DPC的不可纠正错误。支持遏制错误恢复（CER）的操作系统/驱动程序堆栈可以从许多短暂的PCIe不可纠正错误中完全、透明地恢复。DPC可同时支持ECOLC删除和CER



6.8功率放大能力

随着适配器的热插拔能力的增加，系统需要能够正确地

向添加到系统的任何新设备分配功率。 此功能是与电源管理不同的独立功能，并且需要基本级别的支持以确保系统的正确操作。功率预算 concept将允许设备与系统交互以实现这些目标的构建块放在适当的位置。有 系统可以实现实际功率预算能力的许多方式，因此，它们是第二种 本规范的范围。

对于采用不需要热插拔支持的外形规格或集成在系统主板上的设备，可选择实施电源转换功能。外形规格可以

需要支持电源预算。设备和/或适配器需要保持在相应机电规范中规定的配置功率限制之下，直到它们被系统配置和启用。 系统应保证在启用适配器之前已正确预算了电源。

6.8.1系统电源调试过程建议

建议系统固件向电源预算管理代理提供以下信息：

·总系统功率预算（电源信息）。

·由系统固件（系统板设备）分配的总功率。

·插槽总数和插槽类型。

系统固件负责为系统板上所有没有电源的设备分配电源

预算能力。固件可以包括或可以不包括连接到标准电源轨的设备。 当固件为实现功率分配能力的设备分配功率时，必须将功率预算能力寄存器中的系统分配位设置为1b，以指示已正确分配。的

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

功率预算管理器负责分配所有PCIExpress设备，包括具有功率分配能力并具有系统分配位清除的系统板设备。电源预算管理器负责

确定热插拔设备是否可以在系统中被预算和启用。

存在可以提供相同功能的替代方法，并且不要求以这种方式实现功率预算过程。

6.9时隙功率限制控制

PCI Express提供了一种机制，用于软件控制限制适配器（与该插槽相关联）可以配置的每个插槽的最大功率。如果支持，紧急功率降低状态将覆盖

这里列出的机制（见第6.25节）。这一机制的关键要素是：

·在根联合体或交换机的下游端口中实现的时隙能力寄存器的时隙功率限制值和比例字段

·在存在于上游端口中的端点、交换机或PCI接口-PCI桥功能中实现的设备能力寄存器的捕获的时隙功率限制值和比例字段

·Set\_Slot\_Power\_Limit消息，其将（根复合体或交换机的）下游端口的时隙能力寄存器的时隙功率限制值和比例字段的内容传送到对应的

连接到同一链路的组件的上游端口中设备功能寄存器的捕获的时隙功率限制值和比例字段

平台上的功率限制通常由软件（例如，平台固件）控制，该软件维护平台的细节，诸如：

· 平台分区，包括使用适配器进行I/O扩展的插槽

·电力输送能力

·热性能

该软件负责正确编程插槽的插槽功率限制值和比例字段

连接到插槽的下游端口的功能寄存器值写入寄存器后，

在下游端口内，使用Set\_Slot\_Power\_Limit消息将其传送到适配器（参见第www.example.com节2.2.8.5）。消息的接收者必须使用消息数据有效负载中的值来限制整个适配器的功率使用，除非适配器永远不会超过相应的形状因子规范中指定的最低值。要求与适配器相关的设备驱动程序软件能够（通过读取捕获插槽的值

设备能力寄存器的功率限制值和比例字段）来配置适配器的硬件，以保证适配器不会超过所施加的限制。在平台规定的限额低于

最低限度需要充分的操作，设备驱动程序将能够通信这个差异，以更高层次的配置软件。需要配置软件将插槽功率限制设置为最大值之一

基于平台的能力为相应的形状因子指定。

以下规则涵盖Slot功率限制控制机制：对于适配器：

·直到并且除非接收到指示时隙功率限制值大于在适配器的形状因子的形状事实或规范中指定的最低值的Set\_Slot\_Power \_Limit消息，适配器不得消耗超过指定的最低值。

·适配器消耗的功率决不能超过最近收到的

Set\_Slot\_Power\_Limit消息或相应外形规格中指定的最小值，以较高者为准。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·具有端点、交换机或PCI接口-PCI桥接功能的组件，目标是集成到适配器上，其中总功耗低于为目标外形定义的最低限制，

允许忽略Set\_Slot\_Power\_Limit消息，并在设备功能寄存器的捕获时隙功率限制值和比例字段中返回值

. 这样的组件仍然必须能够接收Set\_Slot\_Power\_Limit消息而没有错误，

只需丢弃Message值

对于源插槽的根复合体和交换机

·配置软件不得对Set\_Slot\_Power\_Limit值进行编程，该值指示低于插槽外形规格规格中指定的最低值的限值。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0



执行说明

基于时隙功率限制控制能力的示例

下面的功率限制场景是适配器必须如何基于时隙功率限制控制能力来表现的示例。 形式系数限值是表示，不应作为实际要求。

注：外形规格#1的最大功率要求为40W和25W;外形规格#2的最大功率要求为15W。

场景1：消耗12 W的

· 如果将适配器插入formfactor#140W插槽，则遵循插槽功率限制控制机制，并且适配器正常工作。

· 如果将适配器插入规格为#1的25W插槽，则遵循插槽功率限制控制机制，适配器将正常工作。

· 如果将适配器插入规格为#2的15W插槽，则遵循插槽功率限制控制机制，适配器将正常工作。

在所有情况下，由于适配器在所有外形规格下都能正常工作，因此它可以忽略任何插槽功率限制消息。

场景2：消耗18 W的适配器

· 如果将适配器插入formfactor#140W插槽，则遵循插槽功率限制控制机制，并且适配器正常工作。

· 如果将适配器插入规格为#1的25W插槽，则遵循插槽功率限制控制机制，适配器将正常工作。

· 如果将适配器插入规格为#2的15W插槽，则插槽功率限制控制机制为

随后，适配器必须缩小到15W或禁用操作。对于给定的形状因数，在任何功率限制内都不能扩展的适配器将始终在该形状因数中被禁用，

不应该使用。

在这种情况下，如果适配器仅在形状因子#1中使用，则它可以忽略任何插槽功率限制消息。为了在外形规格#2中有用，适配器应该能够扩展到外形规格#2的功率限制。

场景3：消耗30 W的

· 如果将适配器插入formfactor#140W插槽，则遵循插槽功率限制控制机制，设备工作正常。

· 如果将适配器插入规格为#1的25W插槽，则遵循插槽功率限制控制机制，设备必须缩小到25W或禁用操作。

· 如果将适配器插入规格为#2的15W插槽，则插槽功率限制控制机制为

随后，适配器必须缩小到15W或禁用操作。对于给定的形状因数，在任何功率限制内都不能扩展的适配器将始终在该形状因数中被禁用，

不应该使用。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

在这种情况下，由于适配器消耗的功率高于插槽的最低功率限制，因此适配器必须

能够缩放或禁用以防止系统故障。必须避免在超过适配器所插入的插座的能力的功率水平下操作适配器。



执行说明

时隙功率限制控制寄存器

通常，根复合体或交换机的下游端口内的时隙功率限制寄存器字段将被

由平台专用软件编程。一些实施方式可以使用硬件方法来初始化这些寄存器的值，并且因此不需要软件支持。

具有端点、交换机或PCI接口-PCI桥接功能的组件，如果目标是集成到适配器上，并且总功耗低于为该外形规格定义的最低限值，则允许忽略Set\_Slot\_Power\_Limit消息。请注意，采用此实现方法的组件可能不会

兼容未来可能定义的形状因素。 这样的因素可能强加低于基于现有组件的新适配器所需的最小值的较低功率限制。



执行说明

自动插槽功率限制禁用

在某些环境中，主机软件可能希望直接管理Set\_Slot\_Power\_Limit的传输

通过对插槽功能寄存器执行配置写入，而不是发送 当链路从非DL\_Up状态转换为DL\_Up状态时自动发生。这允许主机软件 在链路断开后或由于电缆或适配器插入而同时热添加多个端点时，通过错开端点到更高功率状态的转换来限制电源浪涌电流。



6.10根复合体拓扑发现

根联合体可以向配置软件呈现以下拓扑之一

·单个不透明的根复合体，使得软件对于根复合体的内部操作没有可见性

根复合体。从软件的角度来看，所有根端口都是相互独立的;不存在任何机制来管理用于任何区分服务的各种根端口之间的任何仲裁。

·单个根复合组件，以便软件对内部

根复合体组件的操作。 如[图6-11所示，](#bookmark97)软件将根端口视为组件的入口端口。用于将流量聚合到系统出口端口或 内部接收单元（例如存储器）由RCRB结构表示。差异化服务的控制 通过位于R CRB中的虚拟信道能力结构来提供。

入口

入口

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **根复合体**  根端口1   |  | | --- | | 端口  仲裁 |  |  | | --- | | 水槽 |  |  | | --- | | 内部端口 |   出口  根端口2   |  | | --- | | VC  仲裁 |  |  | | --- | | RCRB A | |

A-0423

图6-11表示为单个分量的根复合体

·多个根复合体分量，使得软件不仅具有关于以下的可见性和控制：

但是也具有发现和控制给定根复合体组件的内部操作的能力

不同根复合体组件之间的仲裁。如[图6-12](#bookmark98)所示，软件查看根 端口作为给定组件的入口端口。RCRB结构控制从组件到其他组件的出口 根复合体组件（RCRBC）或诸如存储器（RCRBA）的内部接收单元。另外，RCRB结构（RCRBB）也可以存在于给定组件中以控制来自其它根联合体的业务

件. 通过分别适当地位于RCRB中的虚拟信道能力结构来提供对区分服务的控制。

更复杂的拓扑也是可能的。

根复合体拓扑可以表示为逻辑根复合体组件的集合，使得每个逻辑组件具有：

· 一个或多个入口端口。

·出口端口。

·如果根复合体支持虚拟信道，则位于配置空间（对于根端口）或RCRB（对于内部入口/出口端口）中的可选的相关联的虚拟信道能力。

·集成在根复合体中的可选设备/功能。

为了使软件正确地编程仲裁和其他控制参数的PCI Express区分

服务，软件必须能够发现根复合体的内部拓扑。根复合体拓扑发现通过根复合体链路声明能力来完成，如第7.9.8节所述。

入口

入口

入口

入口

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 出口   |  |  |  |  | | --- | --- | --- | --- | | 构成部分1  根端口1   |  | | --- | | VC  仲裁 |   内部端口  根端口2   |  | | --- | | 端口  仲裁 |  |  | | --- | | RCRB C | |   根复合体   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | |  | | 构成部分2 | | | | | | | | | 内部端口 | | |  | |  | | --- | | RCRB B |   VC  仲裁 |  | | | | | | |  |  | |  |  | | 根端口1  根端口2 |  | | | 端口  仲裁 | | | | |  | 内部端口 | | 出口 | 吸收单元 |  | |  | |  |  |  | | |  | |  | | | RCRB A | | |  | | | | |  |  | |

A-0424

图6-12根复形表示为多个分量

6.11链路速度管理

本节描述了如何在LTSSM（第4.2.6节）和软件链路观测和控制机制（参见第www.example.com节、第7.5.3.6节7.5.3.7、第7.5.3.8节、第7.5.3.18节、

7.5.3.19部分和第7部分。5.3.20）。

下游端口中链路控制2寄存器中的目标链路速度字段设置链路的上限

速度除了下面描述的，上游组件必须尝试将链路维持在目标链路速度，

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

或以链路上两个组件支持的最高速度（如训练集中的值所报告的-参见第www.example.com节4.2.4.1），以较低者为准。

允许链路控制2寄存器中的硬件自主速度禁用位清零的任何上游端口或下游端口使用特定于实现的标准自主改变链路速度。

如果链路的可靠性低得不可接受，则允许任一组件通过从组件传输的训练集中通告的支持速度列表中删除不可靠的链路速度来降低链路速度。确定可接受链路可靠性的标准是特定于实现的，并且不依赖于

硬件自动速度禁用位的设置。

在任何给定的速度协商期间，一个或两个组件都可能通告所有速度的子集 支持，作为一种手段，以限制后增强链路的速度。允许组件更改其 在不设置速度更改位的情况下，通过恢复驱动链路，在不请求链路速度更改的情况下通告支持的速度。

当组件

传输的训练集（有或没有改变链路速度的请求），以先到者为准。

允许软件限制链路操作的最大速度，并通过设置 上游组件中目标链路速度字段中的值。 修改“目标链路速度”字段中的值后，软件必须通过向“重新训练链路”位写入1b来触发链路重新训练。通过链路带宽通知机制，软件会收到任何链路速度变化（以及任何链路宽度变化）的通知。

允许软件通过在两个组件中的链路控制2寄存器中写入目标链路速度字段中的相同值并设置输入，使链路以特定速度转换到轮询合规性LTSSM状态。 合规位，然后在链路上启动热复位（通过下游端口）。

请注意，这将使链路进入DL\_Down状态，因此不能对正在使用该链路的其他软件透明。当清除Enter Compliance位时，下游端口将返回到Polling.Active。

6.12访问控制服务（ACS）

ACS在PCI Express拓扑中定义了一组控制点，用于确定TLP是正常路由、阻塞还是重定向。ACS适用于RC、交换机和多功能设备。113对于ACS要求，

具有SR-IOV能力的单功能设备必须像多功能设备一样进行处理，因为它们在启用虚拟功能（VF）之后本质上表现为多功能设备。

允许但不要求在RCiEP中实施ACS。明确允许的是，在单个根复合体内，一些RCiEP实现ACS，而一些不实现。强烈建议根复合体

实现方式确保所有源自不具有ACS能力的RCiEP（PF和VF）的访问都是第一次访问。

在进一步解码和处理之前，由根复合体中的翻译代理（TA）进行处理。此类根复合体处理的细节不在本规范的范围内。

ACS提供以下类型的访问控制：

·ACS源验证

·ACS翻译阻塞

·ACS P2P请求重定向

·ACS P2P完成重定向



113. 多功能设备中的适用功能具体包括PCI Express端点、交换机上游端口、传统PCI Express端点和根复合体集成端点。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·ACS上游转发

·ACS P2P出口控制

·ACS直接翻译P2P

·ACS I/O请求阻塞

·ACS DSP内存目标访问

·ACS USP内存目标访问

·ACS无人认领请求重定向

下一节将讨论每一项的具体要求。

默认情况下，ACS硬件功能处于禁用状态，并且仅由ACS感知软件启用。除ACS源验证外，ACS访问控制不适用于多播TLP（参见[第6.14节](#bookmark99)），并且对

他们

6.12.1 ACS组件能力R要求

ACS功能通过ACS E扩展功能结构报告和管理。PCI Express组件

允许在其部分、全部或全部适用功能中实施ACS扩展功能结构。的 所实现的范围通过每个ACS扩展能力结构中的能力位来传达。具有ACS扩展能力结构的给定功能可以被要求或禁止实现某些功能。

功能，取决于功能的具体类型以及它是否是多功能设备的一部分。

ACS从不适用于PCI Express到PCI桥功能或根复合体事件收集器功能，诸如

函数不能实现ACS扩展功能结构。

<6.12.1.1>ACS下游端口

本节适用于实现ACS扩展功能结构的根端口和交换机下游端口。 本节适用于单功能设备和多功能设备的下游端口功能。

·ACS源验证：必须执行。

当启用时，下游端口从端口接收的每个上游请求的转发器ID测试总线号，以确定它是否与关联的虚拟网桥的辅助侧关联 与下游端口，通过以下任一或两者

. 确定中继器ID是否在端口的总线号“孔径”范围内--由辅助总线号寄存器和从属总线号寄存器指定的包含范围。

. 如果实现并启用了FPB，则通过应用FPB路由ID机制来确定路由器ID与网桥的次级侧相关联。

如果请求的验证器ID中的总线号不在此范围内，则这是与接收端口相关的报告错误（ACS违规）（参见[第6.12.5节](#bookmark100)）。

ACS酸验证从不影响完成。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0



**执行说明**

上游消息和ACS源验证

允许函数在被分配总线号之前传输上游消息。等 消息将具有总线编号为00h的调度员ID。如果下游端口已启用ACS源验证，则这些消息（参见表F-1和第6.23.1节）可能会被检测为ACS违规错误。

·ACS翻译阻塞：必须执行。

启用时，下游端口检查每个上游内存请求的添加请求类型（AT）字段 港口接收。如果AT字段不是默认值，则这是与接收端口相关的报告错误（ACS违规）（参见[第6.12.5节）](#bookmark101)。此错误必须优先于ACS上游转发和任何适用的ACS P2P控制机制。

ACS转换阻止从不影响完成。

·ACS P2P请求重定向：必须由支持与其他根端口的对等流量的根端口实现;114必须由交换机下游端口实现。

ACS P2P请求重定向受ACS P2P出口控制和ACS直接转换P2P机制（如果实施）的影响。更多信息请参见[第6.1和2.3f节](#bookmark102)。

当在交换机下游端口中启用ACS P2P请求重定向时，对等请求必须向上游重定向到RC。

当在根端口中启用ACS P2P请求重定向时，对等请求必须被发送到RC内的重定向请求验证逻辑，该RC确定请求是被“反射”回其原始目标的下游，还是作为ACS违规错误被阻止。算法和特定控制

做出这个决定不是由这个规范来架构的。

下游端口从不重定向正在向下游传输的请求。ACS P2P请求重定向不会影响完成。

·ACS P2P完成重定向：必须由实现ACS P2P请求重定向的根端口实现;必须由交换机下游端口实现。

ACSP2P完成重定向的目的是避免在重定向请求时违反完成和请求之间的排序规则。参见[第6.12.6节f](#bookmark103)或更多信息。

ACS P2P完成重定向不与控制请求的ACS控制交互。

当在交换机下游端口中启用ACS P2P完成重定向时，必须将不具有宽松排序属性位设置（1b）的对等完成115向上游重定向到RC。

否则，必须正常路由对等完成。

当在根端口中启用ACS P2P完成重定向时，必须处理未设置“宽松排序”位的对等完成，使其不会传递发送到重定向的

RC中的请求验证逻辑。这样的完成最终必须向下游发送到它们的原始对等目标，而不引起额外的ACS访问控制检查。

下游端口永远不会重定向正在向下游行进的补体。ACS P2P完成重定向不会影响请求。



114. ACSP2P请求重定向或ACS P2P完成重定向支持的根端口指示并不暗示根联合体的任何特定级别的对等支持，或者根本不支持对等业务

115. 这包括读取完成、AtomicOp完成和其他有或没有数据的完成。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

·ACS上游转发：如果RC支持重定向请求验证，则必须由根端口实现;必须由切换下游端口实现。

当在交换机下游端口中启用ACS上游转发，并且其入口端口接收到

上游请求或完成TLP针对端口自己的出口端口，端口必须改为将TLP上游转发到RC。

当在根端口中启用子ACS上游转发，并且其入口端口接收到上游请求或

针对端口自身的出口端口的完成TLP，端口必须如下处理TLP。用于

请求时，根端口必须将其处理为端口

请求重定向机制。对于完成，根端口必须将其处理为与端口使用ACS P2P完成重定向机制“重定向”的完成相同。

当ACS上游转发未在下游端口上启用，并且其入口端口接收到以端口自己的出口端口为目标的上游请求或完成TLP时，TLP的处理未定义。

·ACS P2P出口控制：实施是可选的。

ACS P2P出口控制受ACS P2P请求重定向和ACS直接转换P2P机制（如果实施）的影响。更多信息请参见[第6.1和2.3f节](#bookmark104)。

支持ACS P2P出口控制的交换机可以选择性地配置为阻止其下游端口之间的对等请求。软件可配置交换机，使其不允许或仅允许其

下游端口向其他下游端口发送对等请求。这是在每个下游端口的基础上配置的。

支持ACS P2P出口控制的RC可以选择性地配置为阻止对等请求

在根端口之间。软件可以将RC配置为不允许层次结构域或仅允许层次结构域的子集向其他层次结构域发送对等请求。这是在每个Root端口的基础上配置的。

利用下游端口中的ACS P2P出口控制，入口端口中的控制（ [6.12.5.](#bookmark106)

完成永远不会受到ACS P2P出口控制的影响。

·ACS直接转换P2P：必须由支持地址转换服务（ATS）的根端口实现，并且还支持与其他根端口的对等流量;116必须由交换机下游

口岸

当在下游端口中启用ACS直接转换P2P时，

地址类型（AT）字段指示转换后的地址必须正常（“直接”）路由到对等出口端口，而不管ACS P2P请求重定向和ACS P2P出口控制设置如何。所有其他点对点

请求仍然必须服从ACS P2P请求重定向和ACS P2P出口控制设置。

完成不会受到ACS Direct Translated P2P的影响。

·ACS I/O请求阻塞：必须由支持ACS增强功能的根端口和交换机下游端口实现。

启用后，端口必须将端口入口收到的上游I/O请求作为ACS违规处理。

·ACS DSP存储器目标访问：必须由支持ACS增强能力并且具有要保护的适用存储器BAR空间的根端口和交换机下游端口来实现。



116. ACS直接转换P2P支持的根端口指示并不暗示根联合体的任何特定级别的对等支持，或者根本不支持对等流量。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

ACS DSP存储器目标访问决定如何通过Down流端口 处理与适用的下游端口相关联的任何存储器BAR空间117的入口和目标。请求可以被阻止、重定向或允许直接前进到其目标。在Switch中，所有

下游端口适用，包括接收请求的端口。在根复合体中， 适用的根端口集是特定于实现的，但总是包括在其上接收请求的根端口。

·ACS USP内存目标访问：必须由支持ACS的交换机下游端口

增强的功能，并且在交换机上游端口中有适用的内存BAR空间要保护;不适用于根端口。

ACS USP内存目标访问决定交换机下游 端口的入口并且以与交换机的上游端口相关联的任何存储器BAR空间118为目标被处理。请求可以被阻止、重定向或允许直接前进到其目标。

如果交换机上游端口以外的任何功能与上游端口相关联，则该字段对访问它们的存储器BAR空间119没有影响。这种访问由交换机上行端口中的ACS扩展能力（如果存在）控制。

·ACS无人认领请求重定向：必须由支持ACS增强功能的交换机下游端口实现;不适用于根端口。

启用时，交换机下游端口 交换机上游端口的存储器窗口内的空间不在存储器窗口内，或者交换机内任何下游端口的存储器BAR目标被重定向到交换机之外的上游。

如果未启用，则交换机下游端口会将此类请求作为不支持的请求（UR）处理。

<6.12.1.2>支持SR-IOV的多功能设备中的ACS功能

本节适用于多功能设备ACS功能，但上一节中介绍的下游端口功能除外。对于ACS要求，具有SR-IOV功能的单功能设备必须像多功能设备一样处理。

·ACS源验证：不得实施。

·ACS翻译阻塞：不得实施。

·ACS P2P请求重定向：必须由支持与其他功能的对等业务的功能来实现。这包括SR-IOV虚拟功能（VF）。

ACS P2P请求重定向受ACS P2P出口控制和ACS直接转换P2P机制（如果实施）的影响。更多信息请参见[第6.1和2.3f节](#bookmark107)。

当ACS P2P请求重定向在不是RCiEP的多功能设备中启用时，对等请求（在设备的功能之间）必须被向上游重定向到RC。

允许但不要求在RCiEP中实现ACS P2P请求重定向。当在RCiEP中启用ACS P2P请求重定向时，对等请求（定义为不以系统为目标的所有请求）

内存，必须发送到根复合体内的特定于实现的逻辑，该逻辑确定



117. 这还包括由扩展ROM基址寄存器（BAR）分配的任何存储器空间。这还包括由BEI值为0、1、7或8的EA条目分配的任何内存空间。 参见See7.8.5.3。

118. 这还包括由扩展ROM基址寄存器（BAR）分配的任何存储器空间。这还包括由BEI值为0、1、7或8的EA条目分配的任何内存空间。 参见See7.8.5.3。

119. 这还包括由扩展ROM基址寄存器（BAR）分配的任何存储器空间。这还包括由BEI值为0、1、7或8的EA条目分配的任何内存空间。 参见See7.8.5.3。

第584

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

请求被定向到其原始目标，或作为ACS违规错误被阻止。本规范未对用于进行此确定的算法和特定控制进行架构设计。

ACSP2P请求重定向不会影响完成。

·ACS P2P完成重定向：必须由实施ACS P2P请求重定向的职能部门实施。ACS P2P完成重定向的目的是避免在完成之间违反排序规则，

请求被重定向时的请求参见第6.12.6f节或更多信息。

ACS P2P完成重定向不与控制请求的ACS控制交互。

当在不是RCiEP的多功能设备中启用ACS P2P完成重定向时，必须将未设置宽松排序位的对等完成向上游重定向到RC。

否则，必须正常路由对等完成。ACS P2P完成重定向不会影响请求。

·ACS上游转发：不得实施。

·ACS P2P出口控制：实施是可选的;基于功能编号或[功能组](#bookmark109)

[编号](#bookmark110);控制多功能或支持SR-IOV的设备内不同功能之间的对等请求。

ACS P2P出口控制受ACS P2P请求重定向和ACS直接转换P2P机制（如果实施）的影响。更多信息请参见[第6.1和2.3f节](#bookmark111)。

支持ACS P2P出口控制的多功能设备内的每个功能可以被选择性地启用以阻止与设备内的其他功能或功能组120的对等通信。这是

在每个功能的基础上配置。

在多功能或支持SR-IOV的设备中使用ACS P2P出口控制，“发送”功能中的控制确定请求是否被阻止，如果是，则“发送”功能根据

[第6.12.5节。](#bookmark112)

当在ARI设备中启用ACS功能组时（ACS功能组启用已设置），ACS P2P出口控制将基于每个功能组而不是基于每个功能实施。 参见[第6.13节。](#bookmark113)

完成永远不会受到ACS P2P出口控制的影响。

·ACS直接转换P2P：如果多功能设备功能支持地址转换服务（ATS）以及与其他功能的对等流量，则必须实现ACS直接转换P2P。

当在多功能设备中启用ACS直接转换P2P时，其地址类型（AT）字段指示转换地址的对等存储器请求必须被正常地（“直接地”）路由

功能，无论ACS P2P请求重定向和ACS P2P出口控制设置如何。所有其他对等请求必须仍然服从ACS P2P请求重定向和ACS P2P出口控制设置。

完成不会受到ACS Direct Translated P2P的影响。

<6.12.1.3>单功能器件中的功能

本节适用于单功能器械功能，但不包括前一节中介绍的下游端口功能和SR-IOV功能。对于ACS要求， 必须像处理多功能器械一样处理具有SR-IOV功能的器械。

ACS能力不适用，且该职能不得实施ACS扩展能力结构。



120. ACS功能组功能对于实现ACS P2P出口控制的ARI设备是可选的。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.12.2互操作性

以下规则控制ACS和非ACS组件之间的互操作性：

·当不使用ACS P2P请求重定向和ACS P2P完成重定向时，ACS和n-ACS

组件可以在拓扑内混合并且将完全互操作。 可以在ACS组件的子集中启用ACS，而不会影响互操作性。

·当使用ACS P2P请求重定向、ACS P2P完成重定向或两者时，PCI Express层级中的某些组件必须支持（上游重定向请求的）ACS上游转发。

具体而言：

关联的根端口121必须支持ACS上游转发。否则，根端口如何处理上行重定向请求或完成TLP是未定义的。RC还必须实现重定向

请求验证。

在启用P2PTLP重定向的每个ACS组件及其关联的根端口之间，任何中间交换机都必须支持ACS上游转发。否则，此类交换机如何处理上游重定向TLP是未定义的。

6.12.3 ACS对等控制交互

对于每个对等请求，多个ACS控制机制可以交互以确定该请求是被直接路由到其对等目标、作为ACS违规被立即阻止、还是被重定向到上游

对RC进行访问验证。对等完成重定向由ACS P2P完成重定向机制排他性地确定。

如果ACS直接转换P2P在端口/FUN中启用，则对等内存请求的地址类型（AT）字段

表示转换后的地址必须正常（“直接”）路由到对等端口/功能，而不考虑ACS P2P请求重定向和ACS P2P出口控制设置。接受这些要求，无条件地接受所有其他要求。

对等请求，必须服从ACS P2P请求重定向和ACS P2P出口控制设置。具体地，适用的出口控制向量位连同ACS P2P出口控制启用位（E）和ACS P2P请求重定向启用位（R）一起确定如何处理请求。必须注意，不能对访问的拓扑性进行限制。 如果ACS对等请求重定向的目标是可能成为锁定目标

access. 有关这些控制位的说明，请参见第7.7.8[表6-10](#bookmark114)说明了相互作用。

表6-10 ACSP 2 P请求重定向和ACSP 2 P出口控制交互

|  |  |  |  |
| --- | --- | --- | --- |
| 控制位E（b） | 控制位R（b） | 关联出口交换机端口、根端口、功能或功能组的出口控制向量位 | 对等请求所需的处理 |
| 0 | 0 | X- | 直接路由到对等目标 |
| 0 | 1 | X- | 重定向上游 |
| 1 | 0 | 1 | 作为ACS违规 |
| 1 | 0 | 0 | 直接路由到对等目标 |
| 1 | 1 | 1 | 重定向上游 |



121.不适用于多功能根复合集成端点的功能之间的ACS重定向。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

|  |  |  |  |
| --- | --- | --- | --- |
| 控制位E（b） | 控制位R（b） | 关联出口交换机端口、根端口、功能或功能组的出口控制向量位 | 对等请求所需的处理 |
| 1 | 1 | 0 | 直接路由到对等目标 |

6.12.4 ACS增强能力

ACS增强功能是一套额外的ACS控制机制，用于提高ACS提供的隔离和保护水平。ACS增强功能定义了以下附加访问控制机制：

·ACS I/O请求阻塞

·ACS DSP内存目标访问

·ACS USP内存目标访问

·ACS无人认领请求重定向

通过这些机制，ACS增强能力提供对指向原始ACS机制未覆盖的区域的请求的保护和一致处理。

执行说明

ACS重定向和来宾物理路由器（GPA）

ACS重定向机制最初是为了实现对P2P内存的细粒度访问控制而设计的

请求，通过将选定的上游请求重定向到RC，其中验证逻辑确定是否

允许或拒绝访问。然而，ACS重定向机制还可以确保在VM的直接控制下的功能将它们的DMA请求正确地路由到主机中的转换代理，然后转换代理将它们的客户物理地址（GPA）转换成主机物理地址（HPA）。

用于内存空间与DMA的GPA范围不保证与PCIe结构用于内存请求路由和访问控制的HPA范围一致。如果用于DMA的任何GPA在HPA范围内

用于内存空间，可能导致合法或恶意的数据包错误路由

ACS重定向机制可以确保用于DMA的具有GPA的上游内存请求永远不会得到

路由到HPA内存范围。ACS P2P请求重定向处理（1）功能之间的对等访问 以及（2）交换机或RC内的下游端口之间的对等访问。 ACSP2P出口控制与重定向以更细粒度的方式处理这两种情况。

ACS增强功能引入的重定向机制可处理其他情况。ACS DSP

带重定向的内存目标访问处理下游端口内存资源范围的此问题。具有重定向的ACS USP内存目标访问处理交换机上游端口内存资源范围的此问题。在交换机、ACS

未声明的请求重定向为Upstream端口内存孔内未声明的任何区域处理此问题

由其他ACS重定向机制处理。这些ACS重定向机制一起可以确保

具有打算用于DMA的GPA的上游存储器请求总是被路由或重定向到主机中的翻译代理，并且具有打算用于P2P的GPA的那些上游存储器请求仍然按照原始架构被路由。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.12.5 ACS违规错误处理

由于硬件或软件缺陷/故障，可能会发生ACS违规。协助故障隔离和根本原因分析 分析表明，ACS组件中应引入AER。AER前缀/报头记录和前缀日志/报头日志寄存器可用于确定违规请求的前缀/报头。ACS违规状态，

掩码和严重性位提供错误的肯定标识，并增强对错误记录和信号的控制。

当检测到ACS违规时，作为完成器122操作的 ACS组件必须执行以下操作：

· 对于未发布的请求，完成者必须生成具有完成者中止（CA）完成状态的完成。

·完成者必须记录ACS违规并发出信号，如[图6-2所示。](#bookmark35)请注意以下事项：

. 即使完成程序在发送完成时使用CA完成状态，

必须记录ACS违规错误，而不是完成程序异常错误。

. 如果ACS违规的严重性是非致命的，并且完成者发送带有CA的完成

完成状态，此情况必须作为建议性非致命错误处理，[如第](#bookmark22)

[6.2.3.2.4.1.](#bookmark22)

·完成器123必须视情况在其状态寄存器或辅助状态寄存器中设置信令目标异常位。

6.12.6 ACS重定向对排序规则的

当ACS P2P请求重定向被启用时，部分或全部对等请求被重定向，这在某些情况下会导致违反排序规则。 本节将探讨这些情况，以及一个类似的情况，该情况发生在RC中，该RC实现“请求重定向”作为强制对等访问控制的替代机制。

<6.12.6.1>完成传递已发布的请求

当对等发布的请求被重定向时，被路由的后续对等非RO124完成被重定向。

直接可以有效地通过重定向的发布请求，违反了非RO完成不得通过发布请求的排序规则。更多信息请参见第2.4.1节。

ACS P2P完成重定向可用于避免违反此排序规则。当ACS P2P完成重定向为

启用时，将重定向所有对等非RO完成，从而采用与重定向对等相同的 发布请求。当一些或所有对等请求被直接路由时，启用ACS P2P完成重定向将不会导致任何排序规则违反，因为允许给定的完成由任何TLP传递， 具有相同交易ID的另一个完成。

作为ACS P2P请求重定向的替代机制，用于实施对等访问控制，

实现“请求重定向”，当RC支持用于“对等”业务的特殊地址范围时，RC将把经验证的上游请求重定向到对等设备。在接收到针对特定的

地址范围内，RC验证请求，将地址转换为目标适当的对等设备，并将请求发送回下游。对于未发布的重定向请求，如果RC不修改请求者ID，



122.除一种情况外，在所有情况下，检测ACS违规的ACS组件也作为完成程序运行。 例外情况是根复合体重定向请求验证逻辑不允许重定向请求。如果重定向的请求来自根端口，则该根端口必须作为完成程序运行。如果重定向的请求来自根复合体集成端点，则关联的根复合体事件收集器必须作为完成器操作。

123. 类似地，如果请求是未发布的，则当认证器接收到结果为具有CA完成状态的完成时，认证器必须将

在其状态寄存器或辅助状态寄存器（视情况而定）中接收目标中止位。注意，对于多功能设备在其功能之间的对等请求引起ACS违反错误的情况，同一功能可以用作触发器和完成器两者。

124.在本节中，“non-RO”是表征未设置其简化排序属性字段的TLP的缩写。

第588

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

最终的完成将“直接”点对点地返回 非RO完成有效地传递重定向的已发布请求，违反了与使用ACS P2P请求重定向时相同的排序规则。ACS P2P完成重定向可用于避免违反此排序规则， 好.

如果ACS P2P请求重定向和RC P2P请求重定向未被使用，

启用ACS P2P完成重定向，建议不要这样做，因为可能会影响性能。

执行说明

ACS P2P完成重定向对性能的影响

虽然使用ACS P2P Completion Redirect可以避免订单违规，

请求，它也可能影响性能。具体而言，所有重定向的完成将必须从重定向点向上行进到RC并返回，从而引入额外的延迟并可能增加链路和RC拥塞。

由于具有RelaxedOrdering位设置的对等完成从不被重定向（从而避免了

性能影响），因此强烈建议实施简化程序以最大限度地正确使用宽松排序，并且软件通过设置启用

设备控制寄存器中的宽松排序位。

如果软件启用ACS P2P请求重定向、RC P2P请求重定向或两者，并且软件确定正确的操作不会受到点对点非RO完成的影响，则通过点对点125发布

请求时，建议softw禁用ACS P2P完成重定向，以避免其性能影响。



<6.12.6.2>请求传递已发布的请求

当一些点对点请求被重定向，而其他点对点请求被直接路由时，存在违反排序规则的可能性，其中非发布请求或非RO发布请求不得通过发布 请求。有关更多信息，请参阅第2.4.1节。

仅当同时启用ACS P2P请求重定向和ACS直接转换P2P时，才存在这些违反排序规则的可能性。软件不应该同时启用这两种机制，除非它可以确定这种排序规则

违反不会发生，或者如果这种排序规则违反确实发生，也不会损害适当的操作。



125. 这些包括由ACS P2P请求重定向机制重定向的真正对等请求，以及路由到根联合体的“逻辑对等”请求，根联合体然后将其重定向到对等设备。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

ACS DirectTranslated P2P确保正确操作

ACSDirectTranslated P2P的目的是优化地址翻译服务（ATS）与点对点通信（其访问控制由RC强制执行）一起使用的环境中的性能。

允许直接路由具有转换地址的对等请求避免了与重定向相关联的可能的性能影响，重定向会引入额外的延迟并可能增加链路和RC拥塞。

对于允许具有转换地址的对等请求，但具有

未转换的地址将作为ACS违规而被阻止，建议软件启用ACS Direct 翻译P2P和ACS P2P请求重定向，并在RC中配置重定向请求验证逻辑，以阻止具有未翻译地址的重定向请求。此配置没有违反排序规则

与传递已发布请求的请求相关联。

对于使用模型，其中一些转换器专门使用转换后的地址进行对等请求

并且一些转发器专门将未转换地址用于对等请求，并且这两类转发器彼此不进行对等通信，因此，通过直接对等请求传递重定向的对等请求（具有未转换地址）不太可能危及正常操作

（翻译地址）。建议软件不要启用ACS直接传输P2P，除非软件确定正确的操作不会因违反排序规则而受到影响。

对于使用模型，其中单个转换器使用已转换和未转换地址，

对等请求，再次建议软件不要启用ACS Direct Translated P2P

软件确信正确的操作不会被所导致的排序规则违反所损害。这

需要对所使用的对等通信模型进行详细分析，并且超出了本说明书的范围。



6.13替代性标识解释（ARI）

路由ID、验证器ID和完成器ID是16位标识符，传统上由三个字段组成：8位总线号、5位设备号和3位功能号。使用ARI，16位字段被解释为两个字段

而不是三个：一个8位总线号和一个8位功能号-设备号字段被删除。这种新的解释使ARI设备能够支持多达256个功能[0. 255]而不是8个函数[0.. 7]。

ARI由一组新的可选功能和控制寄存器位控制。这些规定：

·软件检测组件是否支持ARI的能力。

·将配置ARI下游端口的能力软件化，使得确定何时将类型1配置请求转变为类型0配置请求的逻辑不再强制对传统设备号字段为0的限制。

·软件配置ARI设备以将每个功能分配给功能组的能力。当基于单个功能的更精细粒度控制不适用时，基于功能组的控制

必需的.

. 如果支持并启用多功能VC仲裁，则仲裁可以基于功能组而不是单个功能。

. 如果支持并启用ACS P2P出口控制，则访问控制可以可选地基于功能组而不是单个功能。

以下说明了启用这些功能的示例流程，并提供了有关其使用的其他详细信息

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

1. 软件枚举PCI Express层次结构并确定是否支持ARI扩展功能。

a. 对于ARI下游端口，该功能通过设备功能2寄存器进行通信。

B. 对于ARI设备，该能力通过ARI扩展能力结构来传达

C. ARI对当今平台中使用的基本枚举算法没有影响。

2. 软件在每个组件中启用ARI

a. 在ARI设备正上方的ARI下游端口中，软件设置ARI转发

器件控制2寄存器中的使能位。设置此位可确保用于确定何时将类型1配置请求转换为类型0配置请求的逻辑不再强制执行

传统设备编号字段的限制为0。

B. 在ARI设备中，如果使用类型0配置请求进行处理，扩展功能必须做出响应。ARI感知软件有必要在ARI设备上方的下行端口中启用ARI转发，以便ARI感知软件发现和配置扩展

功能协调发展的

C. 如果ARI设备实现具有功能仲裁的多功能VC能力结构，并且还实现MFVC功能组，则ARI感知软件将功能分类为功能组， 组

I. 每个功能被分配到由功能组编号表示的功能组。二. 最多可配置8个功能组。

三. 在多功能VC仲裁表中，在每个仲裁时隙中使用[功能组号](#bookmark115)代替功能号。

1. 仲裁以职能组为基础，而不是以单个职能为基础。

2. 多功能V C仲裁的所有其他方面保持不变。见章节

7.9.2.10了解更多详情。

四. 每个功能组中的功能仲裁是特定于实现的。

D. 如果ARI设备支持ACS P2P出口控制，则可以可选地基于功能组来实现访问控制。

e. 为了提高枚举性能并创建更确定的解决方案，软件可以通过函数编号的链接列表来枚举函数。下一个链表元素是

通过每个功能的AR I能力寄存器进行通信。

I. 函数0充当函数编号链表的头。软件检测到

将ARI能力寄存器内的非零下一函数号字段作为链表内的下一函数。软件使用设备捕获的总线号和从ARI能力寄存器中导出的功能号发出配置探测以进行定位 下一个关联函数的配置空间。

二. 函数号可以是稀疏的，并且在ARI设备的消费中是非顺序的。

对于ARI设备，每个功能的设备功能寄存器中的“支持的虚拟功能”字段

第www.example.com节7.5.3.3，表7-19）必须设置为00 b，以指示不支持Phantom Functions。扩展标记字段使能位和10位标记转换器使能位仍然可以用于使每个功能支持更高的

未完成的请求数参见2.2.6.2部分。

[图6-13](#bookmark116)显示了一个示例系统拓扑，其中有两个ARI设备，一个位于根端口下方，另一个位于交换机下方。要访问ARI设备X中的扩展功能，根端口A必须支持ARI转发，并通过以下方式启用

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

软件为了访问ARI设备Y中的扩展功能，交换机下游端口D必须支持ARI转发并通过软件启用。 使用此配置时，建议软件不要在根端口B或交换机下游端口C中启用ARI转发。

根端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 根端口A  根复合体 | | |  | |
| B |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ARI设备X | | | |
| |  | | --- | | F0 | | |  | | --- | | F1 | | |  | | --- | | F2 | |  |

多达256个功能

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  | |
| 非ARI器械 | | | | |
|  | |  | | --- | | F1 | | |  | | --- | | F2 | | |  |

|  |
| --- |
| 上游端口 |

开关

下游端口D

下游端口C

F0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  | |
| ARI器械Y | | | | |
| F0 | |  | | --- | | F1 | | |  | | --- | | F2 | | |  |

1至8个功能 多达256个功能 A-0719

图6-13 ARI D设备的系统拓扑

执行说明

ARI转发启用设置不当

强烈建议软件在一般情况下，仅当软件确定下游端口下面的设备是ARI D设备时，才在下游端口中设置ARI转发启用位。如果该位被设置

当出现非ARI设备时，非ARI设备可以在其解释为不同设备号的情况下响应配置空间访问，并且其功能可以在多个设备号下被别名化， 通常导致不期望的行为。

在下游端口下方热插拔之后，强烈建议软件清除ARI

下游端口中的转发使能位，直到软件确定新添加的组件实际上是ARI设备。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

在固件/操作系统控制切换时的ARI转发启用设置

强烈建议固件在控制切换到操作系统时不要在下行端口中设置ARI转发启用位，除非固件知道操作系统是ARI感知的。与此 位设置，非ARI感知操作系统可能能够发现和枚举下游端口下方的ARI设备中的扩展功能，但是这样的操作系统通常将不能管理

扩展功能成功，因为它将解释下游下方存在多个设备 端口而不是单个ARI设备。作为许多预想问题的一个示例，INTx虚拟线的中断绑定将与非ARI感知操作系统所期望的不一致。



6.14多播操作

多播能力结构定义了多播地址范围、将该范围分割成N个相等大小的多播窗口、以及每个多播窗口与多播组MCG的关联。每个

组件内支持多播的功能实现多播能力结构，该多播能力结构为每个MCG提供路由方向和许可检查，以用于通过或到达该功能的TLP。组播组是一个 一个宽度最多为6位的字段，嵌入在地址中，从MC\_Index\_Position开始，如第7.9.11.4节中所定义

.

|  |
| --- |
| MC\_ Base\_ Address |

|  |
| --- |
| 2MC\_索引\_位置 |



+



+



+

*+*

|  |
| --- |
| 多播组0内存空间 |
| 多播组1内存空间 |
| 组播组2内存空间 |
| 组播组3内存空间 |
|  |
| 多播组*N-1*存储空间 |

粤ICP备17057555号-1

图6-14组播地址范围分段

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.14.1组播TLP处理

如果以下所有条件均为真，则会发生多播命中

·MC\_Enable已设置

·TLP是内存写入或地址路由消息，两者都是发布请求

·地址TLP>=MC\_Base\_Address

·地址TLP（MC\_Base\_Address+（2MC\_Index\_Position \*（MC\_Base\_Group+ 1）

在该步骤中，每个SwitchIngress端口和其他组件使用MC\_Enable、MC\_Base\_Address、

MC\_Index\_Position和MC\_Index\_Group从它们的任何一个函数。需要软件来配置交换机的所有功能和多功能上游端口的所有功能，使其在每个字段中具有相同的

如果不是这种情况，则结果是不确定的。

如果非投递存储器请求中的地址在多播窗口中命中，则不发生多播命中，并且TLP在基本规范之前被正常处理-即，作为单播。

如果发生多播命中，则仍然可以应用的唯一ACS访问控制是ACS源验证。特别地，ACS重定向和ACS出口控制向量都不影响多播命中期间的操作。

如果发生多播命中，则不应用正常的地址路由规则。相反，TLP的处理如下：

使用MC\_Base\_Address和MC\_Index\_Position的任何函数值从TLP中的地址提取多播组。具体而言：

MCG =（（地址TLP-MC\_Base\_Address）>>MC\_Index\_Position）3Fh

在此过程中，组件可以使用MC\_Base\_Address和MC\_Index\_Position的任何Function值。使用哪个函数的值是设备特定的。

组件接下来检查对应于所提取的MCG的MC\_Block\_All和MC\_Block\_Untranslated位。

交换机和根端口使用与入口端口相关联的MC\_Block\_All和MC\_Block\_Untranslated寄存器检查其入口端口中的多播TLP。端点功能检查它们准备发送的多播TLP，使用

它们的MC\_Block\_All和MC\_Block\_Untranslated寄存器。如果与提取的MCG对应的MC\_Block\_All位被设置，则TLP被处理为[MC阻塞TLP](#bookmark117)。如果与所提取的MCG相对应的MC\_Block\_Untranslated位是 设置且TLP包含未翻译地址的情况下，TLP也作为[MC阻塞TLP处理。](#bookmark118)

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

MC\_Block\_未转换和PIO写入

编程I/O（PIO）写入内存空间通常具有未转换的地址，因为没有

用于软件控制PIO请求的地址类型（AT）字段的架构机制因此，如果有

对于给定的切换到多播的任何PIO写入，软件应确保适当的

该交换机上游端口中的MC\_Block\_Untranslated位为Clear。否则，交换机上游端口可能会阻止合法地以多播窗口为目标的PIO写入。由于软件可能需要为了PIO写入而清除交换机上游端口中的MC\_Block\_Untranslated位，因此强烈建议

推荐用于能够进行地址转换的根复合体：

·所有集成端点均实现多播能力结构，以提供用于发送未翻译多播TLP的访问控制。

·所有具有对等能力的根端口各自实现多播能力结构，以为对等转发的未转换多播TLP提供访问控制。

出于类似的原因，对于上行端口是多功能设备中的功能的具有多播功能的交换机组件，强烈建议将该多功能设备 实现多播能力结构。



执行说明

多播窗口大小

多播TLP的每个最终接收机可以具有不同的多播窗口大小要求。在一个

在极端情况下，可能需要多播窗口来覆盖在设备内实现的存储器范围。另一方面，它可能只需要覆盖FIFO寄存器所在的特定偏移。的

多播能力寄存器中的MC\_Window\_Size\_Requested字段由端点用于通告其所需的多播窗口的大小。

除非可用的地址空间有限，否则资源分配软件可以将每个请求视为

最小值，并通过MC\_Index\_Position设置多播窗口大小以容纳最大请求。在某些情况下，可以通过配置较小的窗口大小并将 多个连续MCG的相同成员资格。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

多播、ATS和重定向

ACS P2P请求重定向和ACS直接转换P2P机制提供了P2P

未翻译的请求可以被重定向到根复合体（RC）进行访问控制检查，而翻译的P2P请求可以被“直接”路由

性能对于多播TLP，不存在对应的重定向机制。

为了实现类似的功能，RC可以被配置为提供不在多播地址范围内的一个或多个目标存储器空间范围，但是RC映射到“受保护的”多播窗口。多播TLP

无论是否具有ATS功能，都可以将这些RC内存空间范围作为目标，以便间接访问受保护的多播窗口。当任何一种类型的发送器以这些范围为目标进行内存写入时，

满足访问控制检查的每个TLP将被RC向下反射回以受保护的多播窗口为目标的转换地址。126支持ATS的数据库可以请求和缓存已转换的数据库

使用RC存储器空间范围，然后稍后使用那些直接以受保护的多播无线网络为目标的用于备忘录写入的转换的存储器，并且可以在不经过RC的情况下进行多播。

对于只能使用翻译地址直接定位受保护多播窗口的硬件实施，软件在所有适用功能的MC\_Block\_Untranslated寄存器中设置适当的MCG位

整个平台。其位被设置的每个MCG将使其相关联的多播窗口被保护以防止使用未翻译广告的直接访问。



如果TLP在交换机或根联合体中未被阻塞，则其被转发出所有端口，除了其入口端口，其对应于所提取的MCG的MC\_Receive位被设置。在端点中，它由所有函数使用，

设置与所提取的MCG相对应的MC\_Receive位。如果没有端口转发TLP或没有函数消耗它，则TLP将被静默丢弃。

为了防止环路，禁止根端口或交换机端口将TLP转发回其入口端口，即使这样也是如此

由与端口关联的MC\_Receive寄存器一个例外是前面描述的情况

实现说明，其中RC反映了从入口根端口进入多播窗口的单播TLP。在这种情况下，当由与该入口根端口相关联的MC\_Receive寄存器指定时，要求RC发送 反射的TLP从它最初进入的同一根端口输出。

多播命中会暂停正常的地址路由，包括交换机中的默认上游路由。当多播命中发生时，TLP将仅被转发出其与从TLP中的地址提取的MCG相关联的MC\_ReceiveTM位被设置的那些出口端口。如果TLP中的地址不使用正常的

地址解码时，仅当上游端口的MC\_ReceiveRegister指定时，TLP才被复制到上游端口。

6.14.2组播排序

没有为处理MulticastTLP定义新的排序规则。所有多播TLP都是发布请求，并遵循发布请求排序规则。多播TLP相对于多播系统中的其他TLP按照正常排序规则进行排序。

组件的入口流通过复制点。一旦复制到出口流中，多播TLP将遵循与流中的其他发布请求相同的顺序。



126.如果原始发送者属于与此窗口相关联的MCG，则原始发送者还将收到反射的TLP的副本。

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

6.14.3多播能力结构字段更新

多播能力结构的某些字段可以随时改变。其他人不能改变与

除非组件的每个功能中的MC\_Enableb都为Clear，否则无法预测结果。后一组包括MC\_Base\_Address和MC\_Index\_Position。

软件可能随时更改的字段包括MC\_Enable、MC\_Block\_Group、MC\_Receive、MC\_Block\_All和MC\_Block\_Untranslated。对这些字段的更新必须自行排序。例如，考虑TLPA和B 以该顺序到达相同的入口端口和相同的TC。如果A对这些字段之一使用值X，则B必须使用相同的值或更新的值。

对于多功能上游交换机端口由一台交换机接收或由一个端点

功能被呈现给其他并行端点功能和其他并行交换机的下游交换机端口（如果功能在同一设备中，则它们被认为是并行的。转发单个多播TLP

当任何上游交换功能具有适当的MC\_ReceiveSet位时，为上游。

6.14.4 MC阻塞TLP处理

当TLP被MC\_Block\_All或MC\_Block\_Untranslated机制阻塞时，TLP将被丢弃。的

阻止TLP的函数充当完成器。完成者必须记录此[MC阻塞TLP](#bookmark19)错误并发出信号 如[图6-2所示](#bookmark35)。此外，完成器必须在其状态寄存器或辅助状态寄存器中适当地设置信号目标中止位。为了帮助进行故障隔离和根本原因分析，强烈建议在具有组播功能的函数中实现AER。

在根复合体和交换机中，如果入口端口接收到的TLP发生错误， 入口端口。如果错误发生在准备发送TLP的端点功能中，则该端点功能报告该错误。

6.14.5 MC\_叠加机制

提供[MC\_Overlay](#bookmark120)机制以允许端点中不包含多播能力结构的单个BAR用于多播和单播TLP接收。 软件可以通过在下游端口中设置MC\_Overlay\_BAR来配置[MC\_Overlay](#bookmark120)机制以影响这一点，使得多播地址范围或其一部分被重新映射（覆盖）到端点的BAR所接受的存储器空间范围上。在交换机的上游端口 该机制可用于将多播地址范围的一部分覆盖到相关联的存储器空间范围 主机内存。

下游端口上游端口

机制适用于从交换机向上游驶出的TLP。端口 多功能上游端口中的功能。

启用时，覆盖操作指定将多播TLP中地址中位数等于或大于MC\_Overlay\_Size字段的位替换为MC\_Overlay\_BAR中的相应位。换句

关键词：

5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

如果（MC\_Overlay\_Size6）

那么Egress\_TLP\_Addr = Ingress\_TLP\_Addr;

否则Egress\_TLP\_Addr = {MC\_Overlay\_BAR[63：MC\_Overlay\_Size]，

Ingress\_TLP\_Addr[MC\_Overlay\_Size-1：0] };

公式6-1 MC\_Overlay Trans表单规则

如果具有修改的地址的TLP包含可选的ECRC，则未修改的ECRC几乎肯定会指示

错误. 将包含ECRC的TLP多播复制到具有[MC\_Overlay](#bookmark120)的出口端口时要采取的操作

是否启用取决于是否实现了对ECRC再生的可选支持。 [表6-11列出了所有应急措施](#bookmark121)。如果[未启用MC\_Overlay](#bookmark120)，则不修改TLP转发。如果[MC\_Overlay](#bookmark120)为

启用并且TLP没有ECRC，则转发修改后的TLP，其地址如前一段中指定的那样被替换。如果TLP具有ECRC但不支持ECRC再生，则转发修改的TLP，丢弃其ECRC，并清除报头中的TD位以指示未附接ECRC。如果TLP有ECRC和ECRC

如果支持再生，则在转发TLP之前执行ECRC检查。如果ECRC检查通过，则TLP与重新生成的ECRC一起转发。如果ECRC检查失败，则转发具有反向重新生成的ECRC的TLP。

表6-11 MC\_Overlay的ECRC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [MC\_Overlay](#bookmark120)已启用 | TLP具有ECRC | 支持ECRC再生 | ECRC检查通过时的 | ECRC检查失败 |
| 没有 | X | X | 前向TLP未修改 | |
| 是的 | 没有 | X | 正向修正TLP | |
| 是的 | 是的 | 没有 | 丢弃ECRC并清除TD位的前向修改的 | |
| 是的 | 是的 | 是的 | 具有再生ECRC的前向修正TLP | 具有反向再生ECRC的正向修正TLP |

执行说明

MC\_Overlay和ECRC再生

交换机和根复合体端口可以选择支持ECRC再生。如果支持ECRC再生，则强烈建议通过最小化检查原始TLP的ECRC之间的时间来稳健地执行此操作 并将其替换为在修改的TLP上计算的ECRC。在此期间，TLP不受 数据完整性漏洞，如果预检查和再生没有在同一流水线阶段完成。

从通过启用了[MC\_Overlay](#bookmark120)但不支持ECRC重新生成的端口的多播TLP中剥离ECRC允许接收端点启用ECRC检查。在这种情况下，端点将享受

非多播TLP上的ECRC的益处，而不检测由[MC\_Overlay](#bookmark120)机制修改的多播TLP上的ECRC。

当多播ECRC重新生成被确认并且在TLP修改之前检测到ECRC错误时，则反转重新生成的ECRC确保ECRC错误不会被重新生成过程掩盖。



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

向不具备组播功能的端点进行组播

不包含多播能力结构的端点功能无法区分多播TLP和单播TLP。系统设计人员可以利用这一事实来使用这样的端点，

多播目标。这样做的主要要求是，设备上方的交换机端口中的虚拟PCI到PCI桥的基址寄存器和极限寄存器被配置为至少覆盖部分组播地址范围，或者采用[MC\_Overlay](#bookmark120)机制。扩展该推理，甚至有可能单个多播目标功能可以位于PCI Express到PCI/PCI-X桥的PCI/PCI-X侧。

如果没有多播能力结构的端点被用作多播目标并且[MC\_Overlay](#bookmark120)机制

用于多播TLP的地址。因此，在多播窗口 错误.在多播窗口中命中并且在RCiEP的孔径中未命中的内存读取，或者 根据标准地址路由规则，交换机的下游端口将被路由到上游，并在那里被处理为UR。



执行说明

根复合体中的组播

具有支持多播的多个根端口的根复合体可以实现与其实现要求一样多的多播能力结构。如果实现了多个字段，软件应确保[第6.14.3节](#bookmark119)中指定的某些字段配置相同。为了支持到RCiEP的多播

在一个实施方式中，需要经由MC\_Base\_Address寄存器向集成在其中的所有潜在的多播目标端点暴露被标识为多播的所有TLP。每个这样的集成端点然后使用其多播能力结构中的MC\_ReceiveRegister来确定其是否应当接收TLP。



执行说明

多播和多媒体设备

作为潜在组播目标的所有端口功能和端点功能都需要实现组播

能力结构，以便每个都有自己的MC\_ReceiveVector。在单个组件中，软件应该

以相同方式配置这些能力结构的MC\_Enable、MC\_Base\_Address、MC\_Index\_Position和MC\_Base\_Group字段。在这种情况下，仅在一个上实现地址解码逻辑就足够了。

组件中多播BAR的实例



5.0-1.0-PUB - PCI Express®基本规范修订版5.0版本1.0

执行说明

拥塞避免

多播的使用增加了交换机的输出链路利用率，其程度与所使用的多播组的大小和多播流量占总流量的比例成正比。这会增加以下风险： 当使用多播时，拥塞和拥塞扩散。

为了降低此风险，旨在用作多播目标的组件应设计为使用 以线速组播TLP。打算用作组播源的组件应考虑添加速率限制机制。

在许多应用中，应用

适应性强，不会造成拥堵。其他将需要明确的机制来限制注入速率，选择具有足以保持组播业务的必要突发而不主张流控制的缓冲器的交换机，或者选择能够以所需速率吸收组播业务的组播目标组件。系统设计者有责任选择适当的机制和组件来服务于应用程序。



执行说明

作为多播的主机

对于通用系统，预计组播地址范围通常不会配置为

与直接映射到主机内存的内存空间重叠。如果主机内存要作为

对于多播接收者，根复合体可能需要具有某种I/O存储器管理单元（IOMMU），其能够将多播窗口的部分重新映射到主机存储器，可能具有页级粒度。

或者，交换机的上游端口中的[MC\_Overlay](#bookmark120)机制可用于将多播地址范围的一部分覆盖到主机存储器上。

对于缺少IOMMU的嵌入式系统，配置与直接映射到主机存储器的存储器空间重叠的多播窗口可能是可行的，从而避免了对IOMMU的需要。该方法的具体细节超出了本说明书的范围。



6.15原子操作（AtomicOps）

原子操作（AtomicOp）是一个单一的PCI Express事务，目标是内存空间中的位置，读取

location的值，可能会向该位置写入一个新的值，并返回原始值。对该位置的这种“读-修改-写”序列是原子地执行的。AtomicOps包括以下内容

· FetchAdd（Fetch andAdd）：请求包含一个操作数，即“add”值。读取目标位置的值。

. 使用二进制补码算法将“add”值添加到它，忽略任何进位或溢出。.将和写回目标位置。

. 返回目标位置的原始值。

·Swap（无条件交换）：请求包含单个操作数，即