# 5 power manage

## 5.1 综述

电源管理状态如下:

D状态与特定功能相关联;

* D0为运行状态，功耗最大;
* D1和D2为中间省电状态;
* D3Hot为极低功耗状态;
* D3Cold为掉电状态;

L状态与特定链路相关联;

* L0为运行状态;
* L0s、L1、L1.0、L1.1和L1.2为各种较低功耗状态

其他规范定义了相关的功率状态(例如S状态)。本规范没有描述这些状态和D/L/B状态之间的关系。

PM提供以下服务:

* 识别给定功能的电源管理功能的机制
* 将功能转换为特定电源管理状态的能力
* 功能当前电源管理状态的通知
* 在特定事件时唤醒系统的选项

PM兼容PCI总线电源管理接口规范、高级配置和电源接口规范。本章还定义了PCI Express本地电源管理扩展。

PM定义了链路电源管理状态，允许PCI Express物理链路进入响应软件驱动的d状态转换或活动状态链路电源管理活动。PCI Express Link状态对传统总线驱动软件不直接可见，而是从驻留在这些链路上的组件的电源管理状态派生出来的。定义的链路状态有L0、L0s、L1、L2和L3。随着链路状态从L0到L3的转换，节省的功耗会增加。

组件可以使用唤醒机制唤醒系统，然后发送电源管理事件(PME)消息。PCI Express系统可以提供可选的辅助电源(Vaux)，用于从主电源关闭的状态唤醒操作。

与Vaux相关的具体定义和要求是特定于形状因素的，在本文档中，术语“辅助电源”和“Vaux”应根据使用的特定形状因素理解。

PCI Express-PM PME机制的另一个区别是它分离了以下两个PME任务:

* 关联资源的重新激活(唤醒)(即，重新建立PCI Express组件的参考时钟和主电源轨)
* 向根复合体发送PME消息

**活动状态电源管理(ASPM)**是一种基于硬件的自主活动状态机制，即使连接的组件处于D0状态，也可以节省电力。在链路空闲一段时间后，ASPM物理层协议将空闲链路置于低功耗状态。一旦进入低功耗状态，就会由链路两侧出现的流量触发过渡到完全有效的L0状态。ASPM可能被软件禁用。

有关ASPM的更多信息，请参见第5.4.1节。

## 5.2链路状态电源管理

PCI Express定义了链路电源管理状态，取代了PCI总线电源管理接口规范中定义的总线电源管理状态。链路状态对于PCI-PM遗留兼容软件是不可见的，并且要么是从连接到该链路的相应组件的电源管理d状态派生出来的，要么是通过ASPM协议派生出来的(参见第5.4.1节)。

注意，PCI Express物理层可以定义额外的中间状态。请参阅第4章了解更多关于每种状态的细节以及物理层如何处理状态之间的转换。

PCI Express-PM定义了以下链路电源管理状态:

•L0 -激活状态。

对于ASPM和PCI-PM兼容的电源管理都需要支持L0。

所有PCI Express事务和其他操作都是启用的。

•L0s -低恢复延迟，节能“待机”状态。

对于ASPM来说，L0s支持是可选的，除非适配Link的形状因素明确要求支持L0s。

在L0s期间，所有主电源、元件参考时钟和元件内部锁相环必须始终处于活动状态。链路为Tx\_L0s的端口禁用TLP和DLLP传输。

物理层提供了从此状态快速转换到L0状态的机制。当在链路的两端使用公共(分布式)参考时钟时，从L0到L0的转换时间希望小于100个符号时间。

链路上一个组件的发送端可能处于L0s，而链路上另一个组件的发送端处于L0。

L1 -更高的延迟，更低的功耗“待机”状态。

PCI-PM兼容的电源管理需要L1支持。L1对于ASPM是可选的，除非特定的形状因素特别要求。

当L1 PM子状态通过在L1 PM子状态控制1寄存器中设置一个或多个使能位而使能时，这种状态被称为L1.0子状态。

所有主电源在L1期间必须保持活动状态。只要它们满足所规定的L1退出延迟，就可以明确地允许通过应用以下技术来降低功耗，这些技术包括(但不限于)定期而不是连续地检查电气空闲退出、检查仅在一个Lane上的电气空闲退出以及关闭不需要的电路。所有平台提供的组件参考时钟在L1期间必须保持活动状态，时钟电源管理(使用CLKREQ#)和/或L1 PM Substates允许的情况除外。组件的内部锁相环可以在L1期间关闭，从而以增加退出延迟为代价实现更大的功耗节省。

当给定链路上的下游组件的所有功能被编程为不是D0的D-state时，就进入L1状态。如果下游组件请求L1进入(ASPM)并收到请求的肯定确认，也会进入L1状态。

从L1退出是由上游发起的针对下游组件的事务发起的，或者由下游组件发起的指向上游的事务发起的。从L1到L0的转换需要几微秒。

L1中的链路的TLP和DLLP传输被禁用。

**•L1 PM子状态**-可选L1低功耗Link状态的L1.1和L1.2子状态，用于PCI-PM和ASPM。

在L1.1子状态下，保持Link共模电压。当链路处于L1.0子状态，且满足进入L1.1子状态的条件时，进入L1.1子状态。详细信息请参见5.5.1节。

在L1.2子状态中，不需要维持Link共模电压。当链路处于L1.0子状态，且满足进入L1.2子状态的条件时，进入L1.2子状态。参见5.5.1节。获取详细信息。

当clkreq#信号被断言时(参见5.5.2.1和5.5.3.3节)，将启动从所有L1 PM子状态的退出。

**•L2/L3 ready**- L2或L3的过渡点。

需要L2/L3 Ready转换协议支持。

L2/L3 Ready是一种伪状态(对应于LTSSM L2状态)，当准备从下游组件或从两个附加组件移除电源和时钟时，给定链路进入这种状态。

这个过程是在PM软件将设备转换到D3状态后启动的，随后调用电源管理软件启动电源和时钟的移除。当链路状态变为L2/L3 Ready时，部件可以下电。移除主电源后，如果提供并使用Vaux，链路将切换到L2;如果没有提供或使用Vaux，链路将切换到L3。请注意，这些是链路的PM伪状态;在这些条件下，LTSSM通常仅在主电源下运行，因此在主电源移除时将断电。

在确认PME\_Turn\_Off消息(即注入PME\_TO\_Ack TLP)之后，L2/L3 Ready状态进入转换过程必须尽快开始。下游组件通过发送PM\_Enter\_L23 DLLP发起L2/L3 Ready表项。有关电源管理系统消息的详细信息，请参见第5.7节。

链路处于L2/L3 Ready状态时，禁止TLP和DLLP传输。

注意:从L2/L3 Ready返回L0将通过中间LTSSM状态退出。详细信息参见第4章。

•L2 -辅助供电链路，深度节能状态。

L2支持是可选的，并且依赖于辅助电源的存在。

如第5.6节所述，组件只能消耗辅助电源。

在L2中，组件的主电源输入和参考时钟输入关闭。

当处于L2时，任何链路重新激活唤醒逻辑(Beacon或WAKE#)、PME上下文和任何其他“保持活跃”逻辑都由辅助电源供电。

链路在L2中禁止TLP和DLLP传输。

**•L3** -链路断开状态。

当无电源存在时，器件处于L3状态。

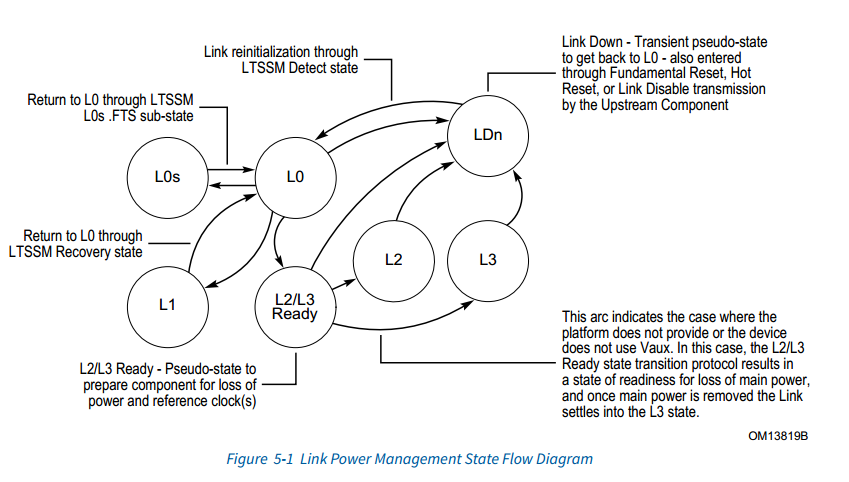
**•LDn** - L0之前的过渡Link Down伪状态。

此伪状态与LTSSM状态Detect、Polling和Configuration相关联，如果适用，还与Disabled、Loopback和Hot Reset相关联。

有关进入和退出L0和L2/L3 Ready之间的每个L-states的详细信息，请参阅第4.2节(（L2.Idle从第4章的角度来看）)。L2状态是通过辅助电源的存在来区分的用于电源管理目标的抽象，并且不应该被解释为暗示LTSSM保持活动的要求。

电气部分规定了驱动器和接收器在无电源情况下的电气特性。这是L3状态，但电气部分不涉及L3。

图5-1显示了可能发生的l状态转换的概述。



L1和L2/L3 Ready进入协商发生在L0状态。“L1”和“L2/L3 Ready”只有在协商完成后才会进入。除非发生LDn，否则在协商过程结束之前，Link Power Management一直保持L0状态。请注意，这些状态和状态转换并不直接对应于物理层LTSSM的操作。

如图5-1所示，L0状态包括LTSSM L0状态、Recovery状态、during LinkUp状态、Configuration状态。此外，LTSSM通常由主电源供电(而不是Vaux)，因此LTSSM不会在L2或L3状态下供电。

下面的示例序列说明了进入系统睡眠状态的多步骤链路状态转换过程:

1. 系统软件指导下游组件的所有功能到D3Hot。

2. 然后，Downstream组件根据需要启动Link到L1的转换。

3.然后，系统软件使根复体广播PME\_Turn\_Off消息，为移除主电源做准备。

4. 该消息导致subject Link转换回L0，以便发送该消息，并使下游组件能够使用PME\_TO\_Ack进行响应。

5. 下游组件发送PME\_TO\_Ack报文后，启动L2/L3 Ready转换协议。

L0→L1→L0→L2/L3 ready

如下面的例子所示，也可以在不首先将所有函数放入D3Hot的情况下移除电源:

1. 系统软件使根复发器广播PME\_Turn\_Off消息，为移除主电源做准备。

2. 下游组件使用PME\_TO\_Ack进行响应。

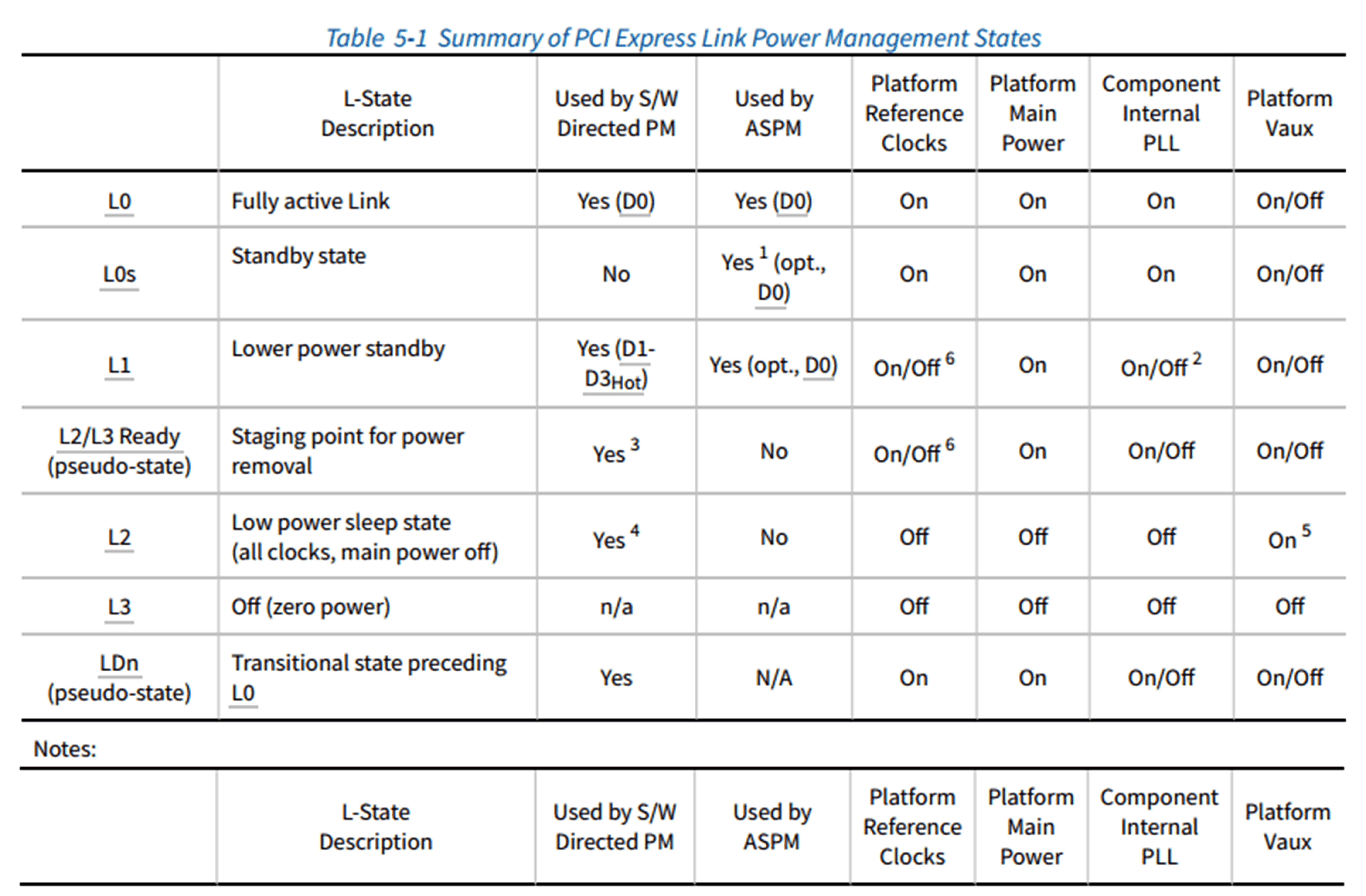
3.下游组件发送PME\_TO\_Ack报文后，启动L2/L3 Ready转换协议。

L0→L2/L3 ready

L1入口协商(无论是通过PCI-PM还是ASPM机制调用)和L2/L3 Ready入口协商映射到一个状态机，该状态机对应于本章后面描述的操作。此状态机将重置为空闲状态。对于下游组件，状态机在离开空闲状态后采取的第一个动作是根据协商的类型开始发送适当的入口DLLP。如果协商被中断，例如通过Recovery中断，则两个组件中的状态机将被回到空闲状态。上游组件必须一直处于空闲状态，等待接收表项dllp。下游组件必须始终处于空闲状态，并且必须始终继续发送条目dllp以重新启动协商。

表5-1总结了每种L-state，描述了它们的使用时间，以及与每种L-state对应的平台和组件行为

“是”项表示需要支持(除非另有说明)。“On”和“Off”条目表示所需提供时钟和电源。“开/关”表示可选的设计选择。



1. 对于连接到给定链路两端的组件，在具有独立参考时钟输入的链路配置中，L0s退出延迟将是最大的(与公共或分布式参考时钟相比)。

2. 对于在此状态期间内部关闭其锁相环的组件来说，L1退出延迟最大。

3.L2/L3 Ready进入顺序在PME\_Turn\_Off/PME\_TO\_Ack协议握手完成时启动。它不直接与D-state的跳转或根据ASPM策略和程序的过渡关联。

4. 根据平台的实现，系统的休眠状态可能1）使用L2状态，2）转换到完全关闭(L3)，3）或者它可能使链路处于L2/L3ready状态。L2/L3 Ready状态转换协议由下游组件在接收到PME\_Turn\_Off TLP消息并得到TLP确认后发起。虽然平台对L2休眠状态配置的支持是可选的(取决于Vaux的可用性)，但需要组件协议支持将Link转换为L2/L3 ready状态。

5. L2状态与L3状态的区别仅在于Vaux的存在和使用。在完成L2/L3 ready状态转换协议之后，在主电源被移除之前，链路已经表示它准备好移除主电源。

6. 低功耗的移动或手持设备可以通过“时钟请求”(CLKREQ#)机制对参考时钟进行时钟门控来降低功耗。因此，针对这些设备的组件应该能够容忍在低功耗状态退出期间重新激活参考时钟所需的额外延迟。

## 5.3 PCI-PM软件兼容机制

### 5.3.1功能的设备电源管理状态

虽然这些电源状态的概念对于系统中的所有function都是通用的，但是当转换到给定的电源管理状态时，其含义或预期的功能行为取决于function的类型(或类)。

D0电源管理状态为功能正常运行状态。其他状态是各种级别的降低功率，功能不运行或支持有限的一组操作。D1和D2是中间状态，旨在为系统设计人员提供更大的灵活性，以平衡给定器件类别的节能、恢复时间和低功耗特性可用性权衡。例如，D1状态可以被支持为比D2稍高的功耗状态，但是它产生的恢复时间比D2更快。

D3电源管理状态是一种特殊的电源管理状态，可以通过软件或物理移除其电源将function转换为D3。从这个意义上讲，两个D3变体被指定为D3Hot和D3Cold，其中下标分别表示主电源的存在或不存在。

D3Hot中的功能可以通过软件写入函数的PMCSR寄存器，从而转换为D0状态。处于D3Cold状态的功能可以通过重新应用主电源并断言Fundamental Reset来转换到d3uninitialized状态。

所有函数必须支持D0和D3状态(D3Hot和D3Cold)。D1和D2状态可选。

|  |
| --- |
| 实现注意  非d0状态下交换机和根端口虚拟桥行为  当与交换机/根端口(“虚拟桥”)关联的Type 1 Function处于非d0电源状态时，它将模拟传统PCI桥在处理内存、I/O和配置请求和完成方面的行为。所有流经下游的内存和I/O请求都作为不支持的请求终止。所有type1的配置请求都作为不支持的请求终止，但是type0的配置请求处理不受虚拟桥D状态的影响。流经虚拟桥的任何方向的完成都不受虚拟桥D状态的影响。  注意，消息的处理不受虚拟桥的PM状态的影响。 |

#### 5.3.1.1 D0 State

所有函数必须支持D0状态。D0分为两个不同的子状态，“未初始化”子状态和“活动”子状态。当组件Conventional Reset撤销时，该组件的所有函数都进入D0uninitialized状态。当一个function完成FLR时，它进入到D0uninitialized状态。配置完成后，功能进入D0active状态——PCI Express Function的完全操作状态。当某个function的内存空间使能、I/O空间使能或总线master使能位被设置为单个或组合时，该function进入D0active状态。

#### 5.3.1.2 D1 State

D1支持是可选的。当处于D1状态时，除了2.2.8节中定义的消息外，function不能在链路上发起任何请求TLP。配置请求和消息请求是function在D1状态下唯一接受的TLP。所有其他收到的请求必须作为不支持的请求处理，所有收到的完成可以选择作为意外完成处理。如果在D1中检测到由收到的TLP引起的错误(例如，不支持的请求)，并且启用了报告功能，如果链路尚未在L0中，则必须返回到L0，并且必须发送错误消息。如果在D1中检测到由收到的TLP以外的事件引起的错误(例如，完成超时)，则在将function编程回D0状态时必须发送错误消息。

注意，函数的软件驱动程序参与了function从D0到D1的转换过程。它通过保存任何function状态(如果有必要)来帮助这个过程，否则就准备将function转换到D1。作为这个静止过程的一部分，函数的软件驱动程序必须确保任何事务中间的TLP(即，未完成的请求)在将控制权交给系统配置软件之前被终止，然后完成向D1的过渡。

#### 5.3.1.3 D2 State

D2支持是可选的。当一个function目前没有被使用，并且可能在一段时间内不会被使用时，它可能会被放入D2中。这种状态要求function提供显著的功耗节约，同时仍然保留完全恢复到先前状态的能力。当处于D2状态时，除了2.2.8节定义的消息外，function不能在链路上发起任何请求TLPs。配置和消息请求是处于D2状态的功能所接受的唯一TLP。所有其他收到的请求必须作为不支持的请求处理，所有收到的完成可以选择作为意外完成处理。如果在D2中检测到由收到的TLP引起的错误(例如，不支持的请求)，并且启用了报告功能，如果链路尚未在L0中，则必须返回到L0，并且必须发送错误消息。如果在D2中检测到由收到的TLP以外的事件引起的错误(例如，完成超时)，则在将function编程回D0状态时必须发送错误消息。

注意，函数的软件驱动程序参与了函数从D0到D2的转换过程。它通过保存任何function状态(如果有必要)来为流程做出贡献，否则就准备将function转换到D2。作为这个静止过程的一部分，function的软件驱动程序必须确保在将控制权交给系统配置软件之前终止任何事务中间的tlp(即未完成的请求)，然后完成向D2的过渡。

在访问内存或I/O空间之前，系统软件必须将功能恢复到D0active状态。启动的操作，如总线主控和中断请求生成，只有在功能恢复到D0active后才能开始。

从功能从D2编程到D0，到下一次向该功能发出请求，最小恢复时间要求为200μs。未定义在此恢复时间窗口中收到的请求的行为（请参阅第7.9.17节）。

#### 5.3.1.4 D3 State

需要支持D3（D3Cold和D3Hot状态）。

如果PMCSR中的No\_Soft\_Reset字段为Set，则D3Hot状态下的功能需要维护function context。在这种情况下，从D3Hot转换到D0后，系统软件不需要重新初始化function（功能将处于D0active状态）。如果No\_Soft\_Reset位为Clear，则function不需要在D3Hot状态下维护function context，但不能保证function context将被清除，软件不得依赖于此类行为。因此，在这种情况下，系统软件需要在转换到D0后完全重新初始化function，因为function将处于D0uninitialized状态。

如果链路状态已转换为L2/L3 ready状态，则无论No\_Soft\_reset位的值如何，该function都将复位。

|  |
| --- |
| 实施说明  转换到L2/L3 ready  如第5.2节所述，平台电源管理软件启动向L2/L3 ready状态的转换，以便开始从设备中移除主电源和时钟的过程。因此，预计设备将在其Link转换为L2/L3 Ready后不久转换为D3Cold，使得仅适用于D3Hot的No\_Soft\_Reset位无关紧要。虽然不能保证L2/L3 Ready和D3Cold之间的这种相关性，但系统软件应确保L2/L3 Ready状态仅在意图移除设备主电源时才进入。设备功能，包括那些能够在D3Hot(即设置No\_Soft\_Reset位)时保持功能环境的功能，当由于所需的DL\_Down状态指示而退出L2/L3 Ready时，需要重新初始化内部状态，如第2.9.1节所述。 |

除非PCI-PM电源管理功能寄存器中的Immediate\_Readiness\_on\_Return\_to\_D0位被设置，否则系统软件必须允许在D3Hot→D0转换后至少10 ms的最小恢复时间(参见章节7.9.17)，才能访问该功能。例如，D3Hot->D0转换组件可以使用该恢复时间来在可访问之前引导其任何组件接口（例如，从串行ROM）。在恢复期间(包括配置请求包)尝试以Function为目标将导致未定义的行为。

#### 5.3.1.4.1 D3Hot State

配置和消息请求是D3Hot状态下function所接受的唯一TLP。所有其他接收到的请求都必须作为不支持的请求进行处理，所有接收到的完成都可以作为意外完成进行处理。如果在D3Hot中检测到由接收到的TLP（例如，不支持的请求）引起的错误，并且启用了报告，则如果链路不在L0中，则必须将其返回到L0，并且必须发送错误消息。如果在D3Hot中检测到由接收到的TLP以外的事件引起的错误（例如，完成超时），则当function被编程回D0状态时，可以选择性地发送错误消息。一旦进入D3Hot，function稍后可以转换为D3Cold（通过从其主机组件断电）。

请注意，function的软件驱动程序参与将function从D0转换为D3Hot的过程。它通过保存任何function状态来帮助该过程，否则这些function状态将随着主电源的移除而丢失，并为功能转换到D3Hot做准备。作为该静止过程的一部分，功能的软件驱动程序必须确保在将控制权移交给系统配置软件之前终止任何未完成的事务（即，未完成的请求），然后系统配置软件将完成向D3Hot的转换。

请注意，D3Hot也是一种有用的状态，用于减少在其他运行系统中空闲组件的功耗。

软件允许D3Hot中的Function（写入其PMCSR PowerState字段）转换为D0active状态或D0uninitialized状态。只要提供了电源和时钟，D3Hot中的function就必须响应配置空间访问，以便通过软件将其返回到D0。请注意，在从D3Hot转换到D0期间或之后，function不需要生成内部硬件复位（请参阅PMCSR中No\_Soft\_reset位的使用）。

如果不需要内部复位，在D3Hot到D0active状态完成后，除了写入PowerState字段外，不需要额外的操作系统干预。如果需要内部复位，设备将返回D0uninitialized，并且需要对设备进行完全重新初始化。完整的重新初始化序列将设备返回到D0active。

如果设备支持PME事件，并且PME\_En已设置，则必须在D3Hot中保留PME上下文。PME上下文也必须保留在PowerState命令转换回D0中。

|  |
| --- |
| 实施说明  未执行内部复位的设备  总线控制器到非PCIe总线，并从PCIe总线上的D3Hot总线控制器恢复，作为非PCIe总线的接口，(例如，CardBus, USB和IEEE 1394)是总线控制器的例子，在从D3Hot恢复时不需要内部复位。如果不需要这种内部复位，总线控制器将不需要在从其次要(非pcie)总线上的D3Hot恢复时执行下游总线复位。 |

|  |
| --- |
| 软复位的多功能设备问题  对于多功能设备(mfd)，影响整个设备行为的某些控制设置要么由所有function中的集合设置确定，要么严格取决于function 0 中的设置。以下是一些关键示例：  •对于非ARI MFD，设备控制寄存器和链路控制寄存器中的某些控制在所有function的集体设置下运行（见第7.5.3.4节和第7.5.3.7节）。  •对于 ARI 设备，设备控制寄存器和链路控制寄存器中的某些控制功能严格按照function 0 的设置运行（参见第 7.5.3.4 节和第 7.5.3.7 节）。  •对于所有MFD，设备控制2和链路控制2寄存器中的某些控制严格按照function0中的设置进行操作（见第7.5.3.16节和第7.5.3.19节）。  对任何function（尤其是function0）执行软复位可能会干扰MFD中其他处于活跃状态的function的正常操作。由于某些操作系统在D3Hot和D0之间转换给定function，期望其他function不会受到影响，因此强烈建议MFD中的每个function都使用电源管理控制/状态寄存器中的No\_Soft\_Reset位集来实现。这样，将给定函数从D3Hot转换为D0不会中断其他处于活跃状态的function的正常操作。  强烈建议MFD中的每个endpoint功能执行Function Level Reset（FLR）。  FLR可用于复位单个endpoint功能，而不会影响可能影响其他function的设置，特别是当这些function处于活动状态时。由于FLR的暂停、错误恢复和清除可重用属性，因此还建议将FLR用于单功能endpoint设备。 |

#### 5.3.1.4.2 D3Cold State

当主电源断开时，function转换到D3Cold状态。通电序列及其相关的冷复位将function从D3Cold状态转换为D0Unitialized状态，硬件将恢复function的开机默认值，就像初始开机时一样。此时，软件必须对功能进行完全初始化，以重新建立所有功能环境，完成功能恢复到D0active状态。

当PME\_En设置时，支持D3Cold唤醒功能的function必须维护其PME环境（在PMCSR中），以便在恢复过程中由PME服务例程软件进行检查。附加环境的保留是特定于实施的。

|  |
| --- |
| 实施说明  PME Context  PME 环境的示例包括但不限于function的PME\_Status位、请求代理的请求者ID、呼叫者ID（如果调制解调器支持）、触发恢复事件的IP定向网络分组的IP信息等。 |

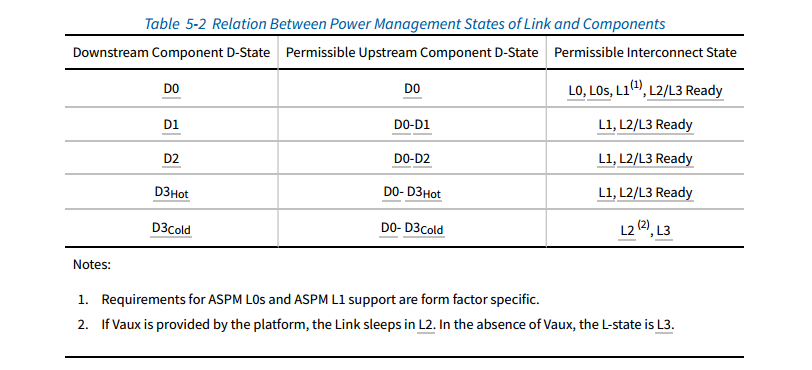
当系统软件对断言Function的 PCI-PM 兼容 PMCSR 的 PME\_Status 位执行 "写 1 清除 "配置事务时，Function的 PME 断言将被确认。

必须使用辅助电源来支持功能中的PME事件检测、链路重新激活，并在D3Cold中保留PME环境。注意，一旦I/O层次结构已经恢复到完全通信状态，作为链路重新激活的结果，唤醒代理然后将PME消息传播到层次结构的根，指示PME事件的源。有关PME的详细信息，请参阅第5.3.3节。

### 5.3.2链路功率管理状态的PM软件控制

链路的功率管理状态由其下游组件的D状态决定。

表5-2描述了组件（具有上行端口）的功率状态与其上行链路之间的关系。



以下规则与PCI-PM兼容电源管理有关：

•D0、D1、D2和D3Hot中的设备必须通过发送PME\_to\_Ack消息来响应收到的PME\_Turn\_Off消息。

•在任何设备D状态下，执行PME\_Turn\_Off/PME\_TO\_Ack握手序列后，下游组件必须使用PM\_Enter\_L23 DLLP请求链路转换到L2/L3 Ready。根据L2/L3 Ready进入转换协议，下游组件必须为主电源和参考时钟的丢失做好准备。

•单个功能设备的上行端口必须仅根据其功能编程为D1、D2或D3Hot，启动链路状态转换至L1。对于Switch，系统软件有责任确保Switch的上游端口的任何D状态编程都是以符合层级范围PM策略的方式进行的（(即上游端口的 D 状态不能低于最活跃的下游端口和下游连接组件/Function的 D 状态）。

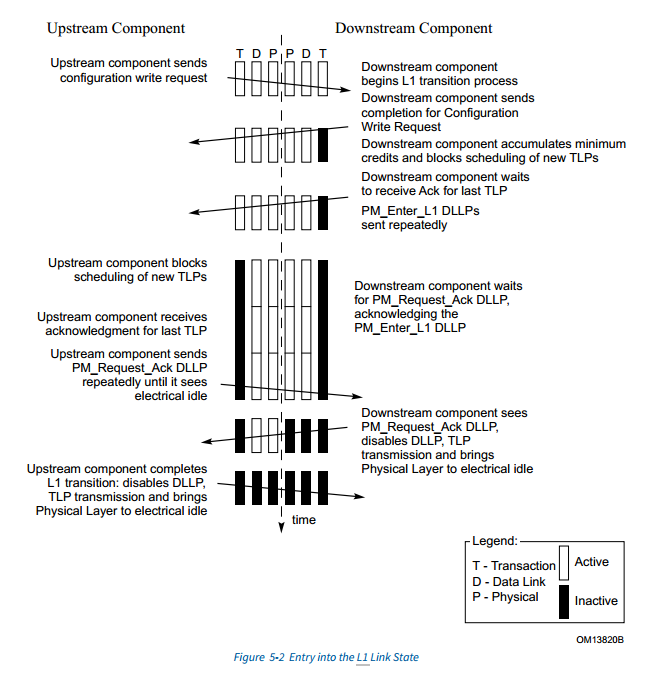
•非ARI-MFD的上游端口在其所有Function被编程为非D0 状态之前，不得启动链路状态转换到L1(代表PCI-PM)。

•ARI设备的上游端口在至少一个Function被编程为非D0状态，并且其所有Function都处于非D0状态或D0uninitialized状态之前，不得启动到L1的链路状态转换(代表PCI-PM)。

#### 5.3.2.1进入L1状态

图5-2描述了链路转换为L1状态的过程，这是电源管理软件将下游连接组件编程为较低功率状态（D1、D2或D3Hot状态）的直接结果。

该图和随后的描述概述了被编程为非D0状态的单功能下游组件的转换过程。



以下文本为图5-2所示的链路状态转换过程提供了更多细节。

PM软件请求：

1.PM软件向下游Function的PMCSR发送配置写请求TLP，以改变下游Function的D状态（例如从D0到D1）。

下游组件链路状态转换启动过程：

2. 下游组件将与配置写请求相对应的完成安排到其PMCSR PowerState字段，并计算所需的完成信用。

3. 然后，下游组件必须等待，直到累积到为所有使能的 VC 的任何 FC 类型发送最大可能的数据包时所需的最小信用。（如果它还没有这样的信用）。然后暂停所有事务层TLP调度。

4.下游组件然后等待，直到它接收到针对PMCSR写入完成的链路层确认，以及它之前发送的任何其他TLP。如果数据链路层规则要求，组件必须从其数据链路层重试缓冲区中重新传输TLP。

5.一旦所有下游组件的TLP都被确认，下游组件就开始发送PM\_Enter\_L1 DLLP。下游组件重复发送该DLLP，两次PM\_Enter\_L1 DLLP传输之间的空闲符号时间不超过8（当使用8b/10b编码时）或32（当使用128b/130b编码时）。在PM\_Enter\_L1传输之间的任何时间都允许传输其他DLLP和SKP有序集，并且不影响此空闲时间限制。

下游组件继续如上所述发送PM\_Enter\_L1 DLLP，直到它接收到来自上游组件的响应（PM\_Request\_Ack）。

下游组件必须继续接受来自上游组件的TLP和DLLP，并根据需要继续使用DLLP进行响应，包括FC更新DLLP和Ack/Nak DLLP。任何被阻止传输的TLP（包括对接收到的TLP的响应）必须被存储以备以后传输，并且必须使下游组件在L1进入后尽快启动L1退出。

上游组件链路状态转换过程：

6.在接收到PM\_Enter\_L1 DLLP时，上游组件阻止所有TLP传输的调度。

7.然后，上游组件必须等待，直到其接收到针对其先前发送的最后TLP的链路层确认。如果链路层规则要求，上游组件必须从其链路层重试缓冲区重新传输TLP。

8.一旦所有上游组件的TLP被确认，上游组件必须向下游发送PM\_Request\_Ack DLLP，而不管有任何未完成的请求。上游组件重复发送该DLLP，两次PM\_Request\_Ack DLLP传输之间的空闲符号时间不超过8（当使用8b/10b编码时）或32（当使用128b/130b编码时）。允许在PM\_Request\_Ack传输间隙的任何时间传输SKP有序集，并且不影响该空闲时间限制。

上游组件继续如上所述发送PM\_Request\_Ack DLLP，直到其观察到其接收通道进入电空闲状态。有关物理层行为的更多详细信息，请参阅第4章。

正在完成L1链路状态转换：

9．一旦下游组件在其接收通道上捕获了PM\_Request\_Ack DLLP（用信号表示上游组件确认了向L1请求的转换），它就禁用DLLP传输，并使上游定向物理链路进入电空闲状态。

10.当上游组件上的接收通道进入电空闲状态时，上游组件停止发送PM\_Request\_Ack DLLP，禁用DLLP传输，并使其传输通道处于电空闲，从而完成链路到L1的转换。

当两个组件的互连链路由于下游组件被编程为非D0状态而处于L1时，两个组件都会暂停其流量控制更新和更新 FCP Timer（见第 2.6.1.2 节）计数器机制的运行。有关物理层行为的更多详细信息，请参阅第4章。

如果与L1的协商中断，请参阅第5.2节。

L1中链路任一端的组件可以选择性地禁用其内部PLL，以便节省更多的能量。

然而，请注意，平台提供的主电源和参考时钟必须在L1的L1.0子状态下继续提供给L1链路两端的组件。

关于进入L1 PM子状态的信息，请参阅第5.5节。

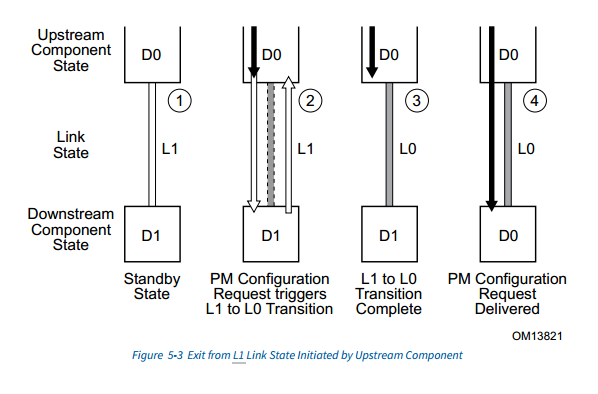
#### 5.3.2.2从L1状态退出

L1退出可以由链路任一端的组件启动。

从L1退出后，建议下游组件在L1退出后1μs内为所有使能的VC和FC类型发送流量控制更新DLLP。

第4章详细描述了将链路从L1转换为L0的物理机制。

如果某个组件需要在链路上传输TLP，则该组件必须启动L1退出。上游组件必须在下游端口上启动L1退出，即使它不具有传输它需要传输的TLP所需的流量控制信用。在L1退出之后，上游组件必须等待从下游组件接收所需的信用。图5-3概述了一个示例序列，该序列将触发上游组件启动链路向L0状态的转换。



事件顺序：

1.电源管理软件启动一个配置周期，目标是一个PM配置寄存器(在本例中是PMCSR的PowerState字段)，该寄存器位于下游组件的一个Function中(例如，将该Function恢复到D0状态)。

2.上游组件检测到配置周期是针对当前处于低功率状态的链路的，因此，启动该链路向L0状态的转换。

3.如果链路处于L1的L1.1或L1.2子状态，则上游组件启动链路向L1的L1.0子状态的转换。

4.根据第4章的定义，链路的两个方向都进入链路训练，导致链路转换到L0状态。第4章详细讨论了L1→L0转换。

5. 当链路的两个方向都恢复到active L0状态后，上行端口将配置报文发送到下行端口。

#### 5.3.2.3 进入 L2/L3 Ready状态

向L2/L3 Ready状态的转换遵循类似于L1进入过程的过程。这两者之间有一些细微的区别，如下所示。

•L2/L3 Ready进入转换协议不会立即导致L2或L3链路状态。向L2/L3 Ready的转换实际上是一次握手，以建立下游组件的断电准备状态。当平台移除组件的电源和参考时钟时，最终实现L2或L3。

•进入L2/L3 Ready的时间由PME\_Turn\_Off/PME\_TO\_Ack握手序列的完成来表示。下游组件必须在开始过渡到L2/L3 ready之前完成任何必要的准备工作，以应对断电。一旦所有断电和时钟丢失的准备工作完成，下游组件通过向上游发送PM\_Enter\_L23 DLLP发起L2/L3 Ready entry。

•L2/L3 Ready入口转换协议使用PM\_Enter\_L23 DLLP。

注意，PM\_Enter\_L23 dllp会持续发送，直到收到确认或断电。

•如果L2/L3 Ready的协商中断，请参见5.2节。

### 5.3.3电源管理事件机制

#### 5.3.3.1动机

PCI Express PME机制与PCI总线电源管理接口规范定义的PME机制在软件上兼容。电源管理事件由Function生成，作为请求PM状态更改的一种方式。电源管理事件通常用于从低功耗状态恢复系统或单个Function。

电源管理软件可以将层次结构转换为低功率状态，并将这些设备的上游链路转换为非通信L2状态。因此，PCI Express PME生成机制分为两个部分：

•唤醒非通信层次（唤醒）。仅当发起PME的设备的上游链路处于非通信L2状态时才需要该步骤，因为在该状态下设备不能向上游发送PM\_PME消息。

•向层次结构的根发送PM\_PME消息

源自PCI Express端点或PCI Express传统端点的PME指示以TLP消息的形式传播到根复合体。PM\_PME消息标识层次结构内的请求代理（通过PME消息报头的请求者ID）。PM\_PME消息中的显式标识旨在促进更快的PME服务例程响应，从而缩短恢复时间。

如果RCiEP与根复合体事件收集器相关联，则源于该RCiEP的任何PME指示都必须由该根复合体事件采集器报告。

源自根端口本身的PME指示通过同一根端口报告。

#### 5.3.3.2链路唤醒

链路唤醒机制提供了一种向平台发信号以重新建立其域内组件的功率和参考时钟的方式。有两种定义的唤醒机制：信标和WAKE#。信标机制使用带内信令来实现唤醒功能。对于支持唤醒功能的组件，对唤醒机制的支持要求由形状因素实现决定。针对在交换机的某些端口上使用Beacon而在其他端口上使用WAKE#的应用程序的交换机组件必须适当地转换唤醒机制（请参阅第5.3.3.2节中题为“WAKE#到Beacon转换的示例”的实施说明）。在WAKE#是唯一使用的唤醒机制的应用程序中，根复合体不需要支持Beacon的接收。

WAKE#机制使用边带信令来实现唤醒功能。WAKE#是由请求唤醒的组件断言并由相关电源控制器观察到的“开漏”信号。WAKE#仅针对特定形状因子进行定义，WAKE#的详细规范包含在相关形状因子规范中。

特定的形状因子规范可能需要使用Beacon或WAKE#作为唤醒机制。

当WAKE#被用作唤醒机制时，一旦WAKE#被断言，断言函数必须继续驱动信号低电平，直到主电源被恢复到组件，正如基本复位不活动所指示的那样。

系统不需要以这样的方式路由或缓冲WAKE#，即保证端点能够检测到信号已被另一个功能断言。

在使用任何唤醒机制之前，软件必须通过在PMCSR中设置功能的PME\_En位来启用功能。PME\_Status位是缓慢变化的，如果辅助电源可用并且功能被启用用于唤醒事件，则功能必须通过复位来保持PME\_Status的值（此要求也适用于PMCSR中的PME\_En位和设备控制寄存器中的辅助电源PM启用位）。

当主系统电源轨关闭时，允许从D3Cold状态产生PME的系统必须提供辅助电源以支持链路唤醒。只有当软件允许组件消耗辅助电源时，组件才能消耗辅助电源，如第5.6节所述。软件需要在参与链路唤醒的所有组件中启用辅助电源消耗，包括必须传播信标信号的所有组件。在存在遗留系统软件的情况下，这是系统固件的责任。

无论使用何种唤醒机制，一旦链路被重新激活和训练，请求代理就会向根复合体上游传播PM\_PME消息。从电源管理的角度来看，这两种唤醒机制提供相同的功能，在本章的其他地方没有区别。

|  |
| --- |
| WAKE#到信标的转换示例  针对连接“信标域”和“WAKE#域”的应用程序的交换机组件必须适当地转换唤醒机制。图5-4显示了两个示例系统，每个系统都包括使用WAKE#唤醒机制的插槽。在情况1中，WAKE#被直接输入到电源管理控制器，并且不需要转换。在情况2中，WAKE#是交换机的输入，并且响应于WAKE#被断言，交换机必须生成传播到根复合体/功率管理控制器的信标 |

##### 5.3.3.2.1 PME同步

PCI Express PM引入了一种围栏机制，用于启动电源移除序列，同时协调平台电源管理控制器的行为和PCI Express代理的PME处理。

PME\_Turn\_Off广播消息在关闭主要组件电源和参考时钟之前，根复合体或交换机下游端口必须发布一条广播消息，指示层次结构中该点下游的所有代理停止发起任何后续PM\_PME消息，在收到PME\_Turn\_Off消息后立即生效。

每个PCI Express代理都需要响应TLP“确认”消息PME\_TO\_Ack，该消息总是路由到上游。在任何情况下，PME\_TO\_Ack消息都必须在PME\_Turn\_Off消息的始发点终止

交换机只有在从其每个下游端口接收到PME\_TO\_Ack消息后才能报告“聚合”确认。一旦PME\_TO\_Ack消息到达每个下游端口，交换机必须在其上游端口上发送PME\_TO\_Ack数据包。出现以下任何一种情况都必须重置聚合机制：从上游端口传输PME\_TO\_Ack消息、在上游端口接收任何TLP、移除交换机的主电源或Fundamental Reset。

具有上行端口的所有组件都必须接受并确认PME\_Turn\_Off消息，而不管相关设备的D状态或多功能设备的任何功能如何。一旦组件发送了PME\_TO\_Ack消息，它就必须准备通过启动到L2/L3 Ready状态的转换来移除其电源和参考时钟。

交换机必须在其所有下行端口都进入L2/L3 Ready状态后，将其上行链路转换为L2/L3 Ready状态。

连接到PME\_Turn\_Off消息发起方的链路是最后一个呈现L2/L3 Ready状态的链路。该状态转换用作对功率传递管理器的指示，即层次结构的该部分内的所有链路已经成功地将所有飞行中的PME消息引退到PME\_Turn\_Off消息的来源点，并且已经执行了任何必要的本地调节以准备电源移除。

为了避免在一个或多个设备没有响应PME\_TO\_Ack消息并将其链路置于L2/L3 Ready状态的情况下发生死锁，电源管理器必须在等待一定时间后实现超时，之后它就像已经收到消息并且所有链路都进入L2/L3 Ready状态一样继续进行。

该定时器的建议限制在1毫秒到10毫秒之间。

在观察到与PME\_Turn\_Off消息的原点对应的所有链路进入L2/L3就绪状态后，电源传输管理器必须等待至少100 ns，然后才能移除组件的参考时钟和主电源。此要求不适用于上述计时器触发的情况（计时器在哪里出现的？）。

|  |
| --- |
| PME\_TO\_Ack消息由交换机代理  PME\_Turn\_Off/PME\_TO\_Ack握手的关键作用之一是确保在睡眠状态断电之前从PCI Express结构中清除所有飞行中的PME消息。这是保证发生的，因为PME消息和PME\_to\_Ack消息都使用VC0内的发布请求队列，并且因此在根复合体处接收PME\_to\_Ack之前，所有先前注入的PME消息将对系统可见。一旦根复合体的所有下游端口接收到PME\_TO\_Ack消息，根复合体就可以向电源管理器发信号，表示在不丢失任何PME消息的情况下移除电源是安全的。  交换机创建分层扩展点，因此，必须等待其所有连接的下游端口接收PME\_to\_Ack消息，然后才能代表其创建的子分层向上游发送PME\_to\_Ack消息。这可以非常简单地使用常见的记分板技术来实现。例如，一旦PME\_Turn\_Off广播消息已经在交换机的下游广播，交换机就简单地检查已经接收PME\_TO\_Ack的每个下游端口。一旦其最后一个活动的下游端口接收到PME\_TO\_Ack，则交换机将代表其下游的整个子层次向上游发送单个PME\_TO\_Ack消息作为代理。注意，一旦下游端口接收PME\_TO\_Ack消息并且交换机对其到达进行评分，然后，端口可以自由地从其内部队列中丢弃数据包，并释放相应的已发布请求队列FC信用。 |

#### 5.3.3.3 PM\_PME消息

PM\_PME消息是发布的事务层数据包（TLP），其通知电源管理软件层次结构中的哪个代理请求PM状态改变。PM\_PME消息与所有其他电源管理系统消息一样，必须使用通用业务类别TC0。

PM\_PME消息总是在根复合体的方向上路由。要在其上行链路上发送PM\_PME消息，设备必须将链路转换为L0状态（如果链路尚未处于该状态）。除非另有说明，否则在发送PM\_PME消息之后，设备将使链路保持在L0状态。

##### 5.3.3.3.1 PM\_PME“反压”死锁的避免

根复合体通常通过本地缓冲来实现，以临时存储可能同时通过层次结构传播的有限数量的PM\_PME消息。给定可以存储在根复合体内的PM\_PME消息的有限数量，在超过该临时PM\_PME信息缓冲区的容量的情况下，可能会对上游定向发布队列施加反压。

死锁可能根据以下示例场景发生：

1. 传入的PM\_PME消息将根复体的临时存储填满其容量，而在层次结构中仍有额外的PM\_PME消息向上游移动。

2. 根复合体，代表系统软件，针对PME请求者的PMCSR之一发出一个配置读请求(例如，读取其PME\_Status位)。

3.根据生产者/消费者排序规则，需要相应的分割完成包来将之前发布的所有PM\_PME消息推到它前面，在这种情况下，这些PM\_PME消息没有地方可去。

4. PME服务程序无法进行;PM\_PME消息存储情况没有改善。

5. 死锁发生。

要防止潜在的死锁，根联合体需要在这些情况下始终启用向前进程。这必须通过接受已发布队列流控制信用允许的任何PM\_PME消息，并丢弃创建溢出条件的任何PM\_PME消息来完成。此必需行为可确保在这些情况下不会发生死锁;但是，PM\_PME消息将被丢弃，从而在进程中丢失。

这种必需的行为确保在这些情况下不会发生死锁；然而，PM\_PME消息将被丢弃并因此在该过程中丢失。

为确保不会永久丢失PM\_PME消息，所有能够生成PM\_PME的代理都必须实现PME服务请求机制，以确保在合理的时间内为其PME请求提供服务。

如果在100ms（+50%/-5%）之后，请求代理的PME\_Status位尚未被清除，则PME服务超时机制到期，触发PME请求代理重新发送暂时丢失的PM\_PME消息。如果此时链路处于非通信状态，则在重新发送PM\_PME消息之前，代理必须重新激活第5.3.3.2节中定义的链路。

#### 5.3.3.4 PME规则

•所有设备功能必须根据PCI-PM规范实现PCI-PM电源管理功能（PMC）寄存器和PMCSR。这些寄存器采用符合PCI-PM的PCI能力列表格式。

◦ PME功能必须在其PMCSR中实现PME\_Status位和底层功能行为。

◦ 当功能启动链路唤醒或发出PM\_PME消息时，必须设置其PME\_Status位。

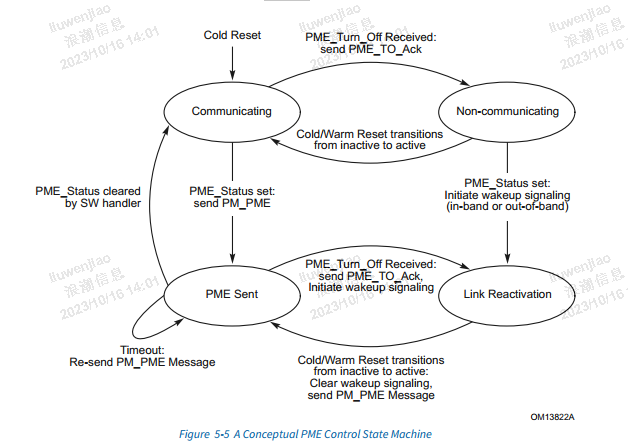
•交换机必须将在任何下游端口上接收到的PM\_PME路由到其上游端口

•**在接收到PME\_Turn\_Off消息时，设备必须阻止PM\_PME消息的传输，并向上游发送PME\_to\_Ack消息。**在链路通过LDn返回到L0状态之后，允许组件发送PM\_PME消息。

•**在链路或层次结构的一部分转移到非通信状态（即，不能发出PM\_PME消息的状态）之前，必须向下游广播PME\_Turn\_Off消息。**

#### 5.3.3.5 PM\_PME投递状态机

下图从概念上概述了PM\_PME投递控制状态机。该状态机通过立即发出PM\_PME而不是要求链路唤醒来确定链路为PME事件提供服务的能力。



**通信状态(Communicating State)**：

在初始上电和相关复位时，上行链路进入通信状态

•·如果PME\_Status被断言（假设PME传递被启用），则PM\_PME消息将被上行发出，在层次结构的根处终止。

·如果接收到PME\_Turn\_Off消息，则链路在其对消息的确认以及随后进入L2/L3就绪状态之后进入非通信状态

**非通信状态(Non-communicating State)**：

•电源和时钟恢复以及相关重置后，下一个状态为通信状态。

•如果PME\_Status被断言，链路将转换到链路重新激活状态，并激活唤醒机制。

**PME发送状态(PME Sent State)：**

•如果PME\_Status被清除，功能将再次变为PME功能。下一个状态是“通信状态（Communicating State）”。

•如果PME服务超时到期时PME\_Status位未清除，则向上游重新发送PM\_PME消息。有关超时机制的说明，请参阅第5.3.3.3.1节。

•如果已发出PME消息，但当链路即将转换为非通信状态(Non-communicating State)（收到PME\_Turn\_Off消息）时，软件尚未清除PME\_Status，则在发送PME\_to\_Ack消息后，链路将转换为链路重新激活状态。该设备还激活唤醒机制。

**链路重新激活状态（Link Reactivation State）：**

•在电源和时钟恢复以及相关重置之后，链路恢复到可进行事务处理的状态。如果需要，设备清除唤醒信令，并发出PM\_PME Upstream并转换到PME Sent状态。

## 5.4本机PCI Express电源管理机制

以下部分定义了需要新软件的电源管理功能。虽然这些功能在新的PCI Express设计中的存在不会破坏传统软件的兼容性，但充分利用它们需要新的代码来管理它们。

这些功能是使用PCI Express本机配置机制枚举和配置的，如本规范第7章所述。有关与这些PCI Express PM功能相关的特定寄存器位置、位分配和访问机制，请参阅第7章。

### 5.4.1有源状态电源管理（ASPM）

所有与内部根复合体链路或系统出口端口无关的端口都需要支持此处定义的活动状态链路PM的最低要求。从最低要求的角度来看，该功能必须被视为与PCI-PM软件兼容功能正交。例如，根复合体不受PCI-PM软件兼容功能要求的约束；但是，它必须执行ASPM的最低要求。

D0状态（即完全活动状态）的组件通常将其上游链路保持在活动L0状态，如第5.3.2节所述。ASPM为D0状态的组件定义了一个协议，通过将其链路置于低功率状态并指示链路的另一端也这样做来降低链路功率。这种能力允许硬件自主、动态链路功率降低，超出了仅由软件控制（即PCI-PM软件驱动）的功率管理所能实现的范围。

为ASPM定义了两种低功耗“待机”链路状态。L0s的低功耗链路状态针对较短的进入和退出延迟进行了优化，同时提供了显著的功耗节约。如果设备中启用了L0s状态，建议设备在链路未使用时将其传输链路置于L0s状态（有关L0s调用策略的详细信息，请参阅第5.4.1.1.1节）。D0设备状态中L0s链路状态的组件支持是可选的，除非链路的适用形状因子规范明确要求。

L1链路状态经过优化，以达到以更长的进入和退出延迟为代价的最大功率节省。对于需要非常低的功率并且可以接受更长的转换时间的情况，L1将链路功率降低到L0s状态之外。对L1链路状态的ASPM支持是可选的，除非特定形状因素特别要求。

定义了可选的L1 PM子状态L1.1和L1.2。对于需要非常低的空闲功率并且可以接受更长的转换时间的情况，这些子状态可以进一步降低链路功率。

每个组件必须在ASPM支持字段中报告其对ASPM的支持级别。如适用，每个组件还应报告其L0s和L1退出延迟（从L0s或L1状态转换到L0状态所需的时间）。终端还必须报告它们在面临风险之前能够承受的最坏情况下的延迟，例如，由于从L0s或L1到L0状态的转换延迟而导致的内部FIFO溢出。电源管理软件可以使用所提供的信息来启用适当级别的ASPM。

如果给定链路的相对侧的参考时钟是从相同的源提供的，或者从不同的源传递到每个组件，则L0s退出延迟可能会显著不同。PCI Express PM软件通过其能力结构的链路控制寄存器中的通用时钟配置位通知每个设备其时钟配置。该位用作设备报告的L0s退出延迟值的决定因素。ASPM可以默认启用或禁用，这取决于具体实施的标准和/或相关形状因子规范的要求。软件可以使用第5.4.1.3.1节中描述的过程启用或禁用ASPM。

电源管理软件通过对ASPM控制字段进行编程，启用或禁用组件的每个端口中的ASPM。请注意，在使用传统操作系统运行时，新的BIOS代码可以有效地启用或禁用ASPM功能，但支持PCI Express的操作系统可能会选择覆盖BIOS配置的ASPM设置。

|  |
| --- |
| 实施说明  同步traffic和ASPM  同步traffic需要有限的服务延迟。ASPM可能会给超过预期限制的等时事务增加延迟。一个可能的解决方案是为配置了等时虚拟通道的设备禁用ASPM。 |

对于ARI设备，ASPM控制仅由function0中的设置确定，而与function0的D状态无关。其他功能中的ASPM控制设置被组件忽略。

非ARI多功能设备的上行端口可以在每个功能的相应ASPM控制字段中用不同的值编程。根据以下规则，此类组件的策略将由所有D0函数中最活跃的公约数决定：

•在确定ASPM策略时，忽略处于非D0状态（D1及更深）的函数

•如果D0状态的任何一个功能的ASPM被关闭(ASPM控制字段= 00b)，或者至少有一个D0状态的功能只对L0启用(ASPM控制字段= 01b)，并且至少有一个D0状态的功能只对L1启用(ASPM控制字段= 10b)，则整个组件的ASPM被关闭

•否则，如果D0状态下的至少一个函数仅为L0s启用（ASPM控制字段=01b），则ASPM仅为L0启用

•否则，若D0状态中的至少一种函数仅为L1启用（ASPM-控制字段=10b），则ASP仅为L1启用

•否则，ASPM同时为L0s和L1状态启用

请注意，当设备功能进入和退出低功耗设备状态时，组件必须能够在运行时更改其行为。例如，如果多功能设备中的一个功能被编程为禁用ASPM，则当该功能处于D0状态时，必须禁用该设备的ASPM。一旦函数转换到非D0状态，如果所有其他函数都为ASPM启用，则可以启用ASPM。

#### 5.4.1.1 L0s ASPM State

L0s低功率链路状态的设备支持是可选的，除非链路的适用外形规格明确要求。

|  |
| --- |
| 实施说明  L0s不受支持时遗留软件的潜在问题  在本规范的早期版本中，L0s的设备支持是强制性的，软件可以合法地假设所有设备都支持L0s。不支持L0s的较新硬件组件可能会遇到此类“遗留软件”的问题。这样的软件甚至可能不会检查链路功能寄存器中的ASPM Support字段，可能不会识别随后为ASPM Support域定义的值（00b和10b），或者可能不会遵循仅在链路两侧的组件都支持L0s的情况下才启用L0s的策略。  遇到以前保留的值00b（无ASPM支持）的旧软件（操作系统或固件）很可能不会启用L1，这是预期行为。传统软件也很可能不会为该组件的发射机启用L0s（也是预期行为），但尚不清楚此类软件是否也会为链路另一侧的组件启用L0s。如果软件在一侧启用L0s，而另一侧的组件没有指示它支持L0s，则结果是未定义的。可能需要通过更新遗留软件、采用“黑名单”或类似机制指示遗留软件不启用L0s，或简单地不支持有问题的系统配置来处理由此产生的行为不可接受的情况。  在一些平台上，固件控制ASPM，操作系统可以保留或覆盖固件建立的ASPM设置。这将受到操作系统是否支持控制ASPM以及在某些情况下固件是否允许操作系统控制ASPM的影响。此外，热插拔操作的ASPM控制可能会受到是否使用本机PCI Express热插拔与ACPI热插拔的影响。用L0s解决任何遗留软件问题可能需要更新固件、操作系统或两者。  当组件没有通告它支持L0s时，如其ASPM Support字段值为00b或10b所示，建议组件的L0s Exit Latency字段返回值111b，指示最大延迟范围。公布这个最大延迟范围可能有助于阻止传统软件启用L0s（如果不是这样的话），从而有助于避免传统软件错误地启用该组件或链路另一侧的组件上的L0所引起的问题。 |

事务层和链路层计时器不受L0s状态转换的影响（即，它们必须遵循各自章节中定义的规则）。

|  |
| --- |
| 实施说明  最小化L0s退出延迟  L0s退出延迟主要取决于接收器快速获取比特和符号同步的能力。  对于高频时钟解决方案存在不同的方法，这些方法可能在L0s退出延迟方面存在显著差异，因此在ASPM的效率方面也存在显著差异。为了使用ASPM实现最大的节能效率，应通过适当选择时钟解决方案来保持L0s的退出延迟较低 |

##### 5.4.1.1.1进入L0s状态

进入L0s状态是针对链路的每个方向单独管理的。链路任一端的每个设备都有责任在其传输通道上发起进入L0s状态。软件不得在给定链路上的任何方向启用L0s，除非链路两侧的组件都支持L0s；否则，结果是未定义的。

针对L0s状态禁用的端口不得将其传输通道转换为L0s状态。但是，如果端口通告其支持L0s，则端口必须能够容忍其接收端口通道进入L0s（由于另一端的设备将其发送通道带入L0s状态），然后再返回L0状态。

L0s调用策略

如果在一段时间内（建议不超过7μs）满足定义的空闲条件（如下），为L0s入口启用的端口通常应将其传输通道转换为L0s状态。在此时间段内，端口用于确定何时进入L0s的策略是特定于实现的。Transmitter从不强制进入L0s。

空闲的定义

“空闲”上行端口的定义因设备功能类别而异。多功能设备的上行端口只有在其所有功能都空闲时才被视为空闲。

如果满足以下条件，则非交换机端口被确定为空闲：

•没有待通过链路传输的TLP，或者没有可用于传输任何TLP的FC信用

•没有待传输的DLLP

如果满足以下条件，则交换机上行端口功能被确定为空闲：

•交换机的下行端口接收通道均未处于L0、恢复或配置状态

•没有待传输的TLP，或没有可用于传输任何内容的FC信用

•没有待传输的DLLP

如果满足以下条件，则交换机的下游端口被确定为空闲：

•交换机的上游端口的接收通道不处于L0、恢复或配置状态

•没有要在此链路上传输的未决TLP，或者没有可用的FC信用

•没有待传输的DLLP

请参阅第4.2节，了解物理层进入L0s的详细信息。

##### 5.4.1.1.2退出L0s状态

当具有要在链路上传输的TLP或DLLP时，发射器位于L0s的组件必须启动L0s退出。请注意，从L0s链路状态的转换并不取决于FC信用的状态（或可用性）。链路必须能够达到L0状态，并能够在链路上交换FC信用。例如，如果某一类型的所有信用在链路进入L0时都已消耗，则当需要通过链路发送新信用时，链路两侧的任何组件都必须能够将链路转换为L0状态。注意，对于组件来说，预测空闲状态的结束并启动L0s发送退出可能是合适的；例如当接收到NP请求时。

下游启动退出

如果需要通过链路进行通信，则允许组件的上游端口在其传输链路上启动从L0s低功率状态的退出（在下游交换机的情况下为上游端口传输通道）。如第4.2节所述，组件在上游方向的线路上启动向L0状态的转换。

如果上游组件是交换机（即，它不是根复合体），则它必须在检测到任何下游端口上的L0s退出后立即在其上游端口传输通道上启动转换（如果上游端口的传输通道处于低功率状态）。

上游启动退出

如果下行端口需要通过链路进行通信，则允许其在任何传输链路上启动从L0s低功率状态的退出。如第4章所述，组件在下游方向的车道上启动向L0状态的转换。

如果下游组件包含交换机，则在其上游端口上检测到从L0s退出后，必须立即在其所有处于L0s的下游端口传输通道上启动转换。已处于L0状态的链路不受此转换的影响。下游组件处于低功率状态（即D1-D3Hot状态）的链路也不受退出转换的影响。

例如，考虑一个交换机，其上行端口处于L0s，下行设备处于D1状态。配置请求包向下游传送到交换机，最终意图对下游设备从D1到D0进行重新编程。交换机的上行端口链路必须转换到L0状态，以允许数据包到达交换机。连接到处于D1状态的设备的下行链路还不会转换到L0状态；它将保持在L1状态。捕获的数据包被检查并路由到与D1中的下游设备共享链路的下游端口。如第4.2节所述，交换机现在将下行链路转换为L0状态。请注意，向L0状态的转换是由路由到该特定下行L1链路的数据包触发的，而不是由上行端口的链路向L0态的转换触发的。如果数据包的目的地针对不同的下行链路，则该特定的下行链路将保持在L1状态。

#### 5.4.1.2 L1 ASPM状态

组件可以可选地支持ASPM L1状态；一种以更长的退出延迟为代价提供更大功率节省的状态。L1退出延迟对软件可见，并通过L1退出延迟字段报告。

|  |
| --- |
| 实施说明  仅支持L1时遗留软件的潜在问题  在本规范的早期版本中，L0s的设备支持是强制性的，并且没有体系结构的ASPM支持字段值来指示在没有L0s支持的情况下对L1的支持。仅支持L1的较新硬件组件可能会遇到“遗留软件”的问题，即软件无法识别随后为ASPM支持字段定义的值。  遇到先前保留的值10b（L1支持）的旧软件可能会避免启用L0和L1，不幸的是，这避免了将L1与仅支持L1的新组件一起使用。虽然这可能会导致额外的电力消耗，但不应导致任何功能不当。但是，在这种10b情况下，与启用L0s的遗留软件存在相同的问题，如第5.4.1.1节中的实施说明“不支持L0s时遗留软件的潜在问题”中所述。 |

如果支持，则默认情况下在ASPM控制字段中禁用L1条目。只有当链路上的两个组件都支持ASPM L1时，软件才能在下游组件上启用ASPM L1。软件必须对ASPM L1的启用和禁用进行排序，以便在下游组件之前启用上游组件，在下游组件之后禁用上游组件。

##### 5.4.1.2.1 ASPM进入L1状态

使能L1 ASPM表项的组件的上游端口可以发起进入L1 Link状态的表项。

关于转换到L1.1或L1.2子状态的详细信息，请参见第5.5.1节。

|  |
| --- |
| 实施说明  启动L1  本规范没有规定具有上行端口的组件何时必须启动到L1状态的转换。在本规范中定义了用于转换到L1和转换出L1的可互操作机制；然而，管理何时过渡到L1的具体ASPM策略留给实现者。  一种可能的方法是，一旦下行设备的接收机和发射机都处于L0s状态（RxL0s和TxL0s）达设定的时间量，则下行设备启动向L1状态的转换。另一种方法是，一旦链路在L0中空闲了设定的时间量，下游设备就发起向L1状态的转换。如果L0s条目未启用，这一点尤其有用。还有另一种方法是，如果下游设备已经完成其分配的任务，则下游设备发起向L1状态的转换。请注意，组件的一级调用策略决不受以下几个示例的限制。 |

三条电源管理消息提供对ASPM L1状态的支持：

•PM\_Active\_state\_Request\_L1（DLLP）

•PM\_Request\_Ack（DLRP）

•PM\_Active\_state\_Nak（TLP）

为ASPM L1条目启用的下游组件与链路上的上游组件协商L1条目。

如果以下所有条件都成立，则下游端口必须接受进入L1的请求：

•端口支持ASPM L1条目，并且启用了ASPM L1条目的功能

•未计划传输TLP

•未计划发送Ack或Nak DLLP

如果满足以下所有条件，交换机上行端口可以在其链路上请求L1条目：

•上行端口支持ASPM L1条目并已启用

•交换机的所有下行端口链路都处于L1状态（或更深）

•没有待传输的TLP

•没有待传输的DLLP

•上行端口的接收器在特定于实现的设置数量时间

请注意，交换机在其任何下行端口上启用ASPM L1链路状态，并在其上行端口上禁用甚至不支持ASPM L1是合法的。在这种情况下，下行端口可能会进入L1链接状态，但交换机永远不会在其上行端口上启动ASPM L1入口转换。

ASPM L1协商规则（见图5-6和图5-7）：

•下游组件不得启动ASPM L1条目，直到它至少累积了为所有启用的VC发送任何FC类型的最大可能数据包所需的最小信用数。

•在决定进入低功率链路状态时，下游组件必须阻止所有TLP从事务层移动到数据链路层进行传输（包括完成数据包）。如果在L1协商过程中，任何TLP从事务层变为可用于传输，则必须首先完成向L1的转换，然后下游组件必须发起向L0的返回。如果与L1的协商中断，请参阅第5.2节。

•下游组件必须等待，直到它收到之前发送的最后一个TLP的链路层确认（即重试缓冲区为空）。如果数据链路层规则要求，组件必须从其数据链路层重试缓冲区中重新传输TLP。

•然后下游组件通过发送PM\_Active\_State\_Request\_L1 DLLP到它的发送通道上发起ASPM协商。下游组件重复发送此DLLP，不超过8个(当使用8b/10b编码时)或32个(当使用128b/130b编码时)PM\_Active\_State\_Request\_L1 DLLP的后续传输之间的空闲符号时间。其他dllp和SKP Ordered set的传输必须在PM\_Active\_State\_Request\_L1传输之间的任何时间根据需要发生，并且不影响此空闲时间限制。L1表项期间的SKP有序集传输遵循4.2.7节中时钟容差补偿规则。

•下游组件继续如上所述发送PM\_Active\_State\_Request\_L1 DLLP，直到接收到来自上游设备的响应（见下文）。下游组件保持在该循环中，等待来自上游组件的响应。

在此等待期间，下游组件不得启动任何事务层传输。它仍然必须接受来自上游组件的TLP和DLLP，存储所需的任何TLP响应以供以后传输。根据链路层协议的需要，它继续使用DLLP（包括FC更新DLLP）进行响应。

如果下游组件出于任何原因需要在链路上传输TLP，则它必须首先完成向低功率链路状态的转换。一旦处于较低功率链路状态，下游组件必须启动低功率链路的退出以处理传输。如果与L1的协商中断，请参阅第5.2节。

•上游组件必须立即（在遵守本规范中的所有其他规则的情况下）对请求作出响应，接受或拒绝请求。如果上游组件无法接受请求，则必须立即（在遵守本规范中的所有其他规则的情况下）拒绝该请求。

•如果与L1的协商中断，请参阅第5.2节。

拒绝的规则：

•如果拒绝，上游组件必须通过向下游组件发送PM\_Active\_State\_Nak消息来尽快安排拒绝。发送PM\_Active\_State\_Nak消息后，允许上游组件启动任何TLP或DLLP传输。

•如果请求被拒绝，通常建议下游组件立即将其传输通道转换为L0s状态，前提是L0s已启用并且满足L0s进入的条件。

•在发送与后续ASPM L1协商序列相关联的PM\_Active\_State\_Request\_L1 DLLP之前，下游组件必须在其发送端进入和退出L0s，或者至少等待上一次发送与前一个ASPM L1协商序列相关联的PM\_Active\_State\_Request\_L1 DLLP的10 μs。这个10 μs定时器必须只计算LTSSM L0和L0s状态所花费的时间。计时器必须保持在LTSSM恢复状态。如果链路断开后又恢复，则忽略定时器，并允许组件在链路恢复后发出新的ASPM L1请求。

|  |
| --- |
| 实施说明  ASPM L1上游组件接受/拒绝注意事项  当上游组件以PM\_Request\_Ack DLLP响应下游组件的ASPM L1请求，接受L1进入请求时，ASPM L1协商协议明确无误地以Link进入L1结束。但是，如果上游组件响应PM\_Active\_State\_Nak消息拒绝L1表项请求，则ASPM L1协商协议的终止就不那么明确了。因此，两个组件都需要被设计成明确地终止协议交换。如果不这样做，则存在两个组件彼此不同步的风险，并且结果可能是未定义的。例如，考虑以下情况：  •下游组件通过传输一系列PM\_Active\_State\_Request\_L1 DLLP请求ASPM L1条目。  •由于临时情况，上游组件以PM\_Active\_State\_Nak消息进行响应，以拒绝L1请求。  •下游组件继续传输PM\_Active\_State\_Request\_L1 DLLP一段时间，然后才能响应PM\_Active\_State\_Nak消息。  •同时，先前导致上游组件拒绝L1请求的临时条件得到解决，上游组件错误地将持续的PM\_Active\_State\_request\_L1 DLLP视为进入L1的新请求，并通过向下游发送PM\_request\_Ack DLLP进行响应。  在这一点上，结果是未定义的，因为下游组件将L1请求视为被拒绝和正在完成，但上游组件将这种情况视为正在接受的第二个L1请求。  为了避免这种情况，下游组件需要提供一种机制来区分一个ASPM L1请求和另一个。下游组件通过进入L0s或在开始传输与第二个请求相关的PM\_Active\_State\_Request\_L1 DLLP之前，从与第一个ASPM L1请求相关的最后一个PM\_Active\_State\_Request\_L1 DLLP开始等待至少10μs来完成此操作（如上所述）。  如果上游组件能够表现出上述行为，则上游组件有必要通过在其接收器上检测到向L0s的转换或在L0/L0s或更长时间内测量到的9.5μs的PM\_Active\_State\_request\_L1 DLLP的接收中断来识别L1请求序列的结束，作为下游组件的ASPM L1请求之间的分离。  如果存在歧义的可能性，上游组件应拒绝L1请求，以避免潜在地造成上述歧义情况。 |

接受时的规则：

•如果上游组件准备接受请求，则必须阻止事务层对任何TLP的调度。

•上游组件必须等待，直到接收到其先前发送的最后一个TLP的数据链路层确认。如果数据链路层规则要求，上游组件必须重新传输TLP

•一旦所有的tlp都被确认，上游组件发送一个PM\_Request\_Ack DLLP Downstream。上游组件重复发送此DLLP，不超过8个(当使用8b/ 10b编码时)或32个(当使用128b/130b编码时)在后续传输PM\_Request\_Ack DLLP之间的空闲符号时间。SKP有序集的传输必须在PM\_Request\_Ack传输之间的任何时间按照要求进行，并且不会导致此空闲时间限制。L1表项期间的SKP有序集传输遵循4.2.7节中的时钟容差补偿规则。

•上游组件继续如上所述发送PM\_Request\_Ack DLLP，直到其观察到其接收通道进入电气空闲状态。有关物理层行为的更多详细信息，请参阅第4章。

•如果上游组件在发送PM\_Request\_Ack DLLP后出于任何原因需要在链路上传输TLP，则它必须首先完成向低功率状态的转换，然后在链路返回L0后启动从低功率状态退出以处理传输。如果与L1的协商中断，请参阅第5.2节。

◦ 在这种情况下，上游组件必须启动从L1的退出，即使它不具有传输TLP所需的流量控制信用。

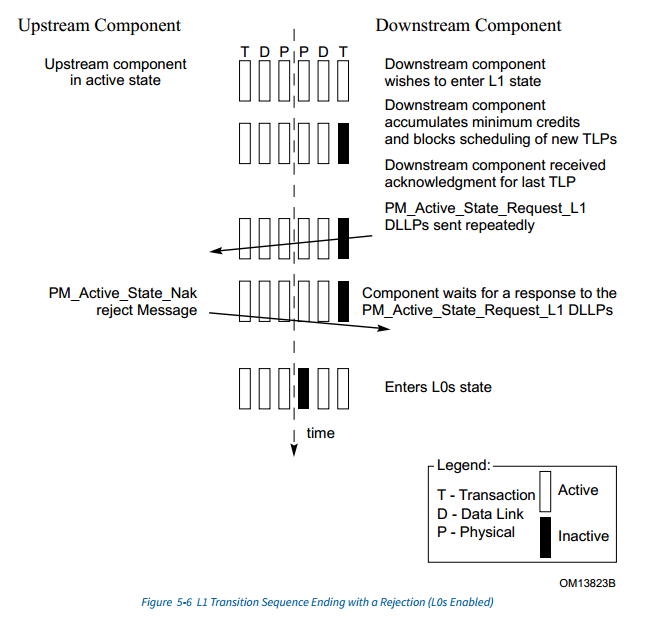
•当下游组件在其接收通道上检测到PM\_Request\_Ack DLLP（表明上游设备确认转换为L1请求）时，下游组件停止发送PM\_Active\_State\_Request\_L1 DLLP，禁用DLLP、TLP传输，并使其传输通道进入电空闲状态。

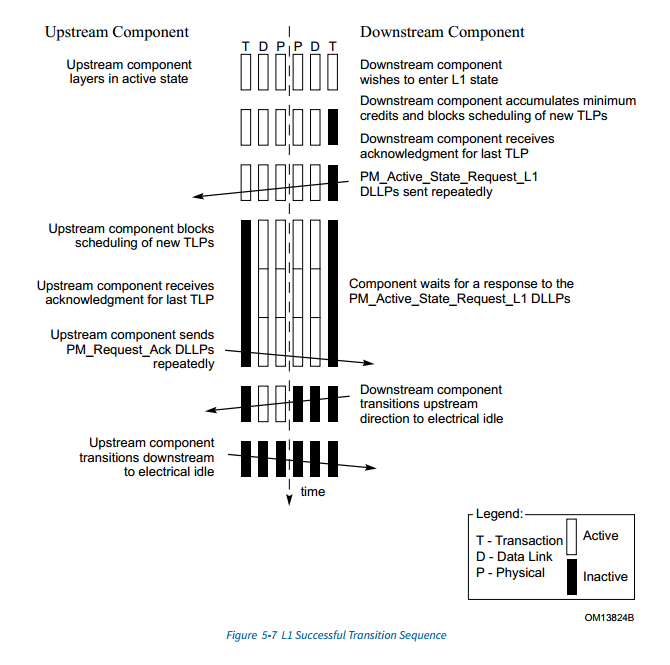
•当上游组件在其接收通道上检测到电气空闲（表明下游组件已进入L1状态）时，它停止发送PM\_Request\_Ack DLLP，禁用DLLP、TLP传输，并使链路的下游方向进入电气空闲状态。

注：

1. 事务层完成超时机制不受向L1状态转换的影响(即，它必须保持计数)。

2. 当链路处于L1状态时，流控制更新计时器被冻结，以防止计时器过期，这将不必要地将链路转换回L0状态。





##### 5.4.1.2.2退出L1状态

链路任一端的组件可以启动从L1链路状态的退出。

有关过渡到L1.1或L1.2子状态的详细信息，请参见第5.5.1节。

从L1退出后，建议下游组件在L1退出后1μs内发送所有启用的VC和FC类型的流量控制更新DLLP

下游组件启动退出

如果上行端口需要通过链路进行通信，则必须在其传输通道上启动从L1的退出。组件启动到L0状态的转换，如第4章所述。上游组件必须通过启动其传输通道的类似转换来做出响应。

如果上游组件是交换机下游端口（即，它不是根复合体根端口），则交换机必须在其任何下游端口链路上检测到L1退出活动后，立即在其上游端口的传输通道上启动L1退出转换（如果上游端口的链路处于L1状态）。由于L1退出延迟相对较长，交换机在其上行端口链路上启动L1退出转换之前，不得等到其下行端口链路完全退出到L0。等待下行链路完成L0转换将导致消息在经过多个交换机时经历累积延迟。

交换机需要在其任何下行端口链路上的L1退出转换开始后不超过1μs后，在其上行端口链路上启动L1退出转换。有关L1退出期间物理层信令的详细信息，请参阅第4.2节。

考虑图5-8中的示例。附加到每个端口的数字表示相应端口报告的传输通道L1出口延迟，单位为微秒。

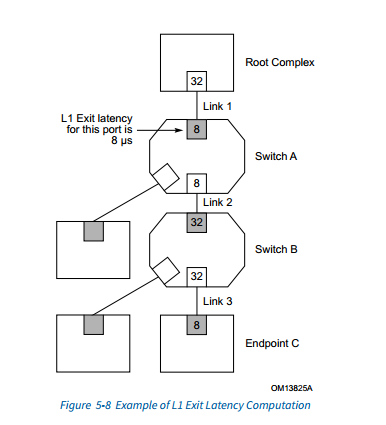
链路1、2和3都处于L1状态，端点C在时间T启动向L0状态的转换。由于交换机B在其端口上退出L1需要32μs，因此链路3将在T+32转换到L0状态（考虑到端点C的T+8和交换机B的T+32，时间最长）。

交换机B需要在从链路3上的L1状态开始转换不超过1μs后，在其上行端口链路（链路2）上启动从L1状态的转换。因此，向L0状态的转换将在链路2的T+1处开始。类似地，链路1将在时间T+2开始其向L0状态的转换。

如上所述，链路2将在时间T+33完成其向L0状态的转换（因为交换机B需要更长的时间来转换并且它在时间T+1开始）。链路1将在时间T+34完成向L0状态的转换（因为根复合体需要32μs才能转换，并且它在时间T+2开始）。

因此，在链路1、2和3中，最后完成向L0状态转换的链路是具有34μs延迟的链路1。

这是在端点C中发起转换的数据包所经历的延迟。



交换机不需要在其任何其他下行端口链路上启动L1退出转换。

上游组件启动退出

根复合体，或者交换机必须在其任何根端口或下游端口链路上启动从L1的退出（如果需要通过该链路进行通信）。交换机或根复合体必须能够启动L1退出，即使它不具有传输给定TLP所需的流量控制信用。组件启动到L0状态的转换，如第4章所述。下游组件必须通过在其传输通道上启动类似的转换来做出响应。

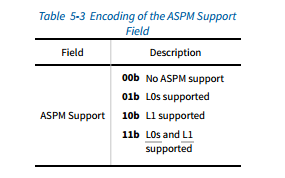
如果下行组件包含交换机，则它必须在其上行端口链路上检测到从L1状态退出时，立即在其所有下行链路上启动转换（假设下行链路处于ASPM L1状态）。由于L1退出延迟相对较长，交换机在其下游端口链路上启动L1退出转换之前，不得等到其上游端口链路完全退出到L0。如果是这样的话，通过多个交换机的消息在通过每个交换机时会经历累积的延迟。

交换机需要在其上行端口上从L1状态转换开始不超过1μs后，在其当前处于L1的所有下行端口链路上启动从L1状态的转换。有关L1退出期间物理层信令的详细信息，请参阅第4.2节。已处于L0状态的下游端口链路不参与退出转换。下游组件处于低功率D状态（D1-D3Hot）的下游端口链路也不受L1出口转变的影响（即，此类链路不得转变为L0状态）。

#### 5.4.1.3 ASPM配置

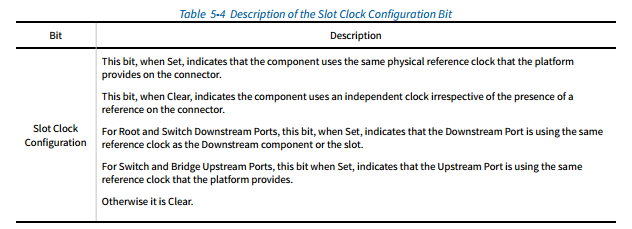
所有函数必须实现以下配置位以支持ASPM。有关配置寄存器分配和访问机制，请参阅第7章。

每个组件在下面的ASPM支持字段中报告其对ASPM的支持级别



软件不得在给定链路上的任何方向启用L0s，除非链路两侧的组件都支持L0s；否则，结果是未定义的。

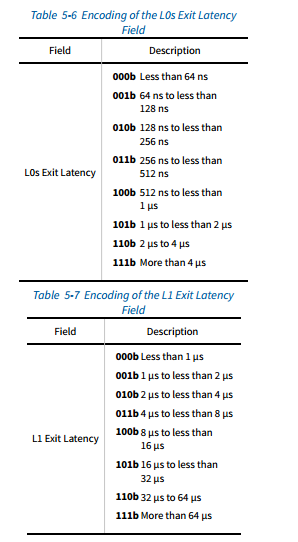
每个组件在其能力结构的链路状态寄存器中的插槽时钟配置位中报告其参考时钟的来源



每个组件必须支持其能力结构的链路控制寄存器中的公共时钟配置位。软件写入该寄存器位，以向设备指示其是否与链路另一端的设备共享相同的时钟源。

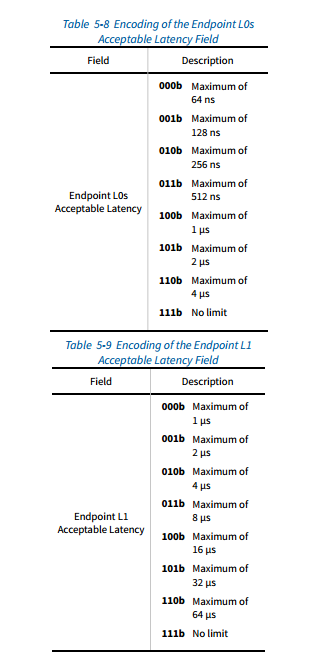


每个端口分别在L0s exit latency（L0s退出延迟）和L1 exit Latncy（L1退出延迟）配置字段中报告L0s和L1退出延迟（将接收通道从L0s或L1状态转换为L0状态所需的时间）。如果端口不支持L0s或ASPM L1，则相应的退出延迟字段的值未定义。



端点还报告由于从L0s状态或L1状态到L0状态的转换而可以吸收的额外延迟。这分别在端点L0s可接受延迟和端点L1可接受延迟字段中报告。

电源管理软件使用层次结构中所有组件报告的延迟信息，可以通过将从根到端点的每个给定路径的退出延迟与每个相应端点可以承受的可接受延迟进行比较来启用适当级别的ASPM。



电源管理软件通过对ASPM控制字段进行编程来启用或禁用每个组件中的ASPM。



**ASPM Control = 00b**

端口的Transmitter不得进入L0s。

连接到链路下游端的端口不得在其上游链路上发出PM\_Active\_State\_Request\_L1 DLLP。

连接到接收L1请求的链路上游端的端口必须以否定确认进行响应

**ASPM Control = 01b**

如果满足所有条件，则端口必须使链路进入L0s状态。

连接到链路下游端的端口不得在其上游链路上发出PM\_Active\_State\_Request\_L1 DLLP。

连接到接收L1请求的链路上游端的端口必须以否定确认进行响应

**ASPM Control = 10b**

端口的Transmitter不得进入L0s。

连接到链路下游端的端口可能会发出PM\_Active\_State\_Request\_L1 DLLP。

如果满足第5.4.1.2.1节中的根复合体根端口或交换机下游端口的条件，则连接到链路上游端的端口必须对L1请求做出肯定确认，并转换为L1。

**ASPM Control = 11b**

如果满足所有条件，则端口必须使链路进入L0s状态。

连接到链路下游端的端口可能会发出PM\_Active\_State\_Request\_L1 DLLP。

如果满足第5.4.1.2.1节中的根复合体根端口或交换机下游端口的条件，则连接到链路上游端的端口必须对L1请求做出肯定确认，并转换为L1。

##### 5.4.1.3.1启用或禁用ASPM的软件流程

以下是一个示例软件算法，重点介绍了如何在组件中启用或禁用ASPM。

•PCI Express组件通过其插槽时钟配置位中的适当值通电。他们初始化该位的方法是特定于设备的。

•PCI Express系统软件扫描每个链路两端组件中的插槽时钟配置位，以确定两者是否使用相同的参考时钟源或来自不同源的参考时钟。如果两个设备中的插槽时钟配置位都已设置，则它们都使用相同的参考时钟源，否则它们不会。

•PCI Express软件更新每个链路两端组件中的公共时钟配置位，以指示这些设备是否共享相同的参考时钟，并通过向上游组件的链路控制寄存器中的Retrain Link位写入1b来触发链路再训练。

•根据通用时钟配置位的设置，设备必须在其L0s/L1退出延迟字段中反映适当的L0s/L1退出延迟。

•PCI Express系统软件然后根据每个端口报告的延迟读取并计算每个端点的L0s/L1退出延迟。有关示例，请参阅第5.4.1.2.2节。

•对于具有一个或多个端点功能的每个组件，PCI Express系统软件检查端点L0s/L1可接受延迟，如每个端点功能在其链路功能寄存器中报告的那样，并相应地启用或禁用该层次结构上的部分或所有干预设备端口中的L0s/L1条目（通过链路控制寄存器中的ASPM控制字段）

## 5.5 L1 PM子状态

L1 PM子状态建立了一个链路功率管理机制，创建L1链路状态的低功率子状态(见图5-9)，以及使用这些子状态的相关机制。L1 PM子状态是:

•L1.0 substate

◦L1.0 substate对应传统L1 Link state。当链路进入L1时，就进入这个子状态。L1 PM子状态机制定义了从该子状态到L1.1和L1.2子状态之间的转换。

◦根据4.2.6.7.2节的要求，上游和下游端口必须启用以检测电气空闲出口。

• L1.1 substate

◦保持链路共模电压。

◦使用双向开漏时钟请求(clkreq#)信号进入和退出此状态。

◦上游和下游端口不需要启用以检测电气空闲出口。

• L1.2 substate

◦不需要维护链路共模电压。

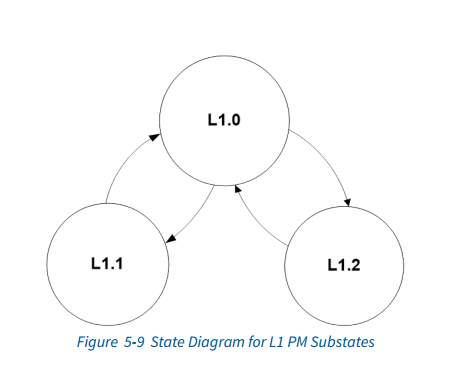
◦使用双向开漏时钟请求(clkreq#)信号进入和退出此状态。

◦上游和下游端口不需要启用以检测电气空闲出口。

支持L1 PM子状态的端口在L1.0以外的L1 PM子状态下不需要参考时钟。

支持L1 PM子状态和支持SRIS模式的端口在SRIS模式下工作时需要支持L1 PM子状态。在这种情况下，clkreq#信号由本节中定义的L1 PM Substates协议使用，但与链路上任何端口使用的任何本地时钟没有定义的关系，并且这些本地时钟的管理是特定于实现的。

支持ASPM L1的L1.2子状态的端口必须支持LTR (Latency Tolerance Reporting)。



•当启用时，L1 PM子状态机制对clkreq#信号应用以下额外要求:clkreq#信号必须被链路的上游和下游端口支持为双向开漏信号。每个端口必须有一个唯一的信号实例，并且上游和下游端口clkreq#信号必须连接。

•当链路处于PCI-PM L1或ASPM L1状态，或者链路处于L2/L3 Ready伪状态时，允许上游端口解除CLKREQ#;当链路处于任何其他状态时，上游端口必须断言CLKREQ#。

•与clkreq#信号相关的所有其他规格，如果没有由L1 PM Substates明确定义或修改，则继续适用

如果在一个特定的系统中不能满足这些需求，那么就不能启用L1 PM Substates。

|  |
| --- |
| 实现说明  clkreq#连接拓扑  对于上游组件，clkreq#信号的连接拓扑可以有所不同。下面描述了一些clkreq#连接拓扑的示例。对于下游组件，这些情况本质上是相同的，但是从上游组件的角度来看，有一些关键的区别，如下所述。  例1: 单个锁相环的单个下行接口连接单个上行接口(如图5-10所示)。  在这个平台配置中，上游组件(A)只有一个clkreq#信号。上游和下游端口的clkreq# (A和B)信号相互连接。在这种情况下，上游组件(A)必须在需要参考时钟时断言clkreq#信号。    例2:上行组件有多个下行接口，共用一个锁相环，连接到独立的下行组件(如图5-11所示)。  在这个示例配置中，上游组件(A)有三个clkreq#信号实例，每个下游端口有一个clkreq#信号实例，上游组件(A)有一个公共共享clkreq#信号实例。CLKREQC#信号用于连接下游组件(B和C)上游端口的clkreq#信号，上游组件的公共共享clkreq# (CLKREQA#)信号用于请求共享锁相环的参考时钟。上游组件(A)中的PLL控制逻辑只能在下游端口都处于L1.1或L1.2子状态时关闭并解除CLKREQA#，并且PLL的所有内部(A)消费者都不需要时钟。    电路板实现者有必要考虑组件将支持哪些clkreq#拓扑，以便进行适当的电路板级连接以支持L1 PM子状态和参考时钟生成。 |

|  |
| --- |
| 实现说明  避免L1 PM子状态和LTSSM之间的意外交互  通常情况下，节省电力的实现技术也会增加恢复正常操作的延迟。在实现L1 PM子状态时，实现者必须确保任何增加的延迟不会与平台的其他元素产生负面交互。确保不会意外触发LTSSM超时条件尤其重要。尽管典型的实现不会接近导致此类交互的延迟，但实现者有责任确保实现正确的整体操作。 |

### 5.5.1 L1 PM子态的进入条件和L1.0要求

当L1 PM子状态为L1.0，并且LTSSM通过PCI-PM兼容电源管理进入L1时，该链路被认为处于PCI-PM L1.0。当L1 PM Substate在L1.0，并且LTSSM通过ASPM进入L1时，认为该链路处于ASPM L1.0中。

以下规则定义了L1.1和L1.2子状态的进入方式:

•上游和下游端口都必须监视clkreq#信号的逻辑状态。

•当PCI-PM L1.0中PCI-PM L1.2使能位被设置时，clkreq#去断言时必须进入L1.2子状态。

•当在PCI-PM L1.0中，PCI-PM L1.1使能位为Set时，当clkreq#去断言且PCI-PM L1.2使能位为Clear时，必须进入L1.1子状态。

•当在ASPM L1.0中，设置了ASPM L1.2 Enable位时，当CLKREQ#去断言且满足以下所有条件时，必须进入L1.2子状态:

◦该端口最近发送或接收的窥探LTR值大于等于LTR\_L1.2\_THRESHOLD value和Scale字段设置的值，或者没有窥探服务延迟要求。

◦该端口最近一次发送或接收的非窥探LTR值大于等于“LTR\_L1.2\_THRESHOLD value”和“Scale”字段设置的值，或者没有非窥探服务延迟要求。

•在ASPM L1.0中，设置了ASPM L1.1使能位，当CLKREQ#解除断言且不满足进入L1.2子状态的条件时，必须进入L1.1子状态。

当L1.2的进入条件满足时，遵循以下规则:

•上游和下游端口都必须监视clkreq#输入信号的逻辑状态。

•在链路进入L1.0之前，上游端口不能解除clkreq#。

•允许任意端口断言CLKREQ#以防止Link进入L1.2。

•下游端口想要阻止进入L1.2，必须在链路进入L1之前断言clkreq#。

•当clkreq#解除断言时，端口进入L1.2。进入L1.2子状态。

如果“下行端口”为PCI-PM L1.0，且设置了“PCI-PM L1.1 Enable”和/或“PCI-PM L1.2 Enable”，或者“下行端口”为ASPM L1.0，且设置了“ASPM L1.1 Enable”和/或“ASPM L1.2 Enable”，且“下行端口”在没有进入L1.1或L1.2的情况下发起退出到“恢复”，则“下行端口”必须断言CLKREQ#，直到链路退出“恢复”。

### 5.5.2 L1.1 Requirements

如果实现了电空闲(EI)出口检测和Refclk活动检测，则允许上游和下游端口停用机制，但是两个端口必须保持公共模式。

#### 5.5.2.1 Exit from L1.1

如果上游或下游端口需要从L1.1启动退出，则必须断言CLKREQ#，直到链路退出Recovery。上游端口必须在进入Recovery时断言clkreq#，并且必须继续断言clkreq#，直到下一个进入L1的表项，或者其他允许clkreq#解除断言的状态。

•如果断言CLKREQ#，则下一个状态为L1.0。

◦Refclk最终将被打开，在PCI Express Mini CEM规范中定义，这可能会延迟根据上游端口发布的LTR。

图5-12显示了进入L1.1的情况，退出由Upstream Port驱动。

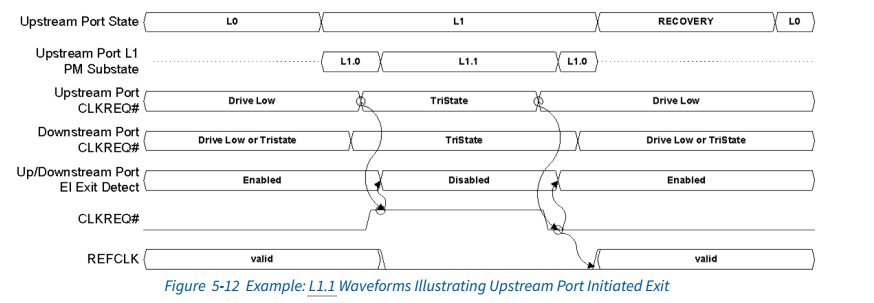
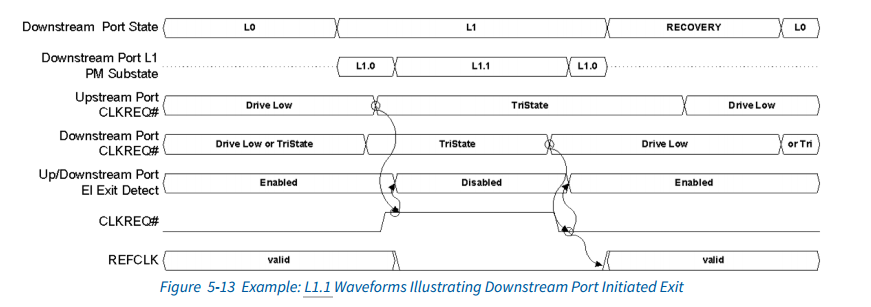


图5-13显示了进入L1.1，退出由下行端口驱动。

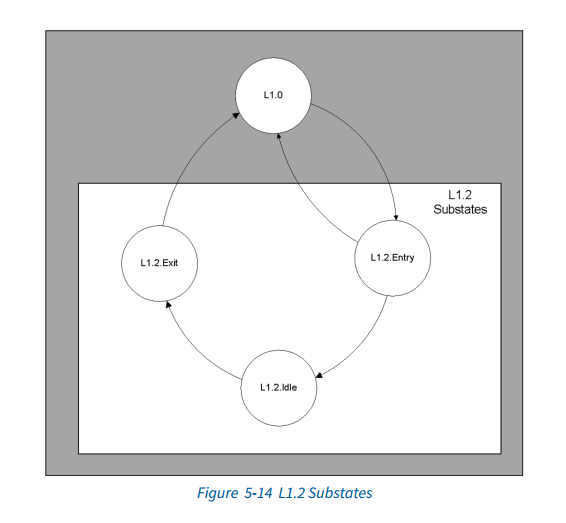


### 5.5.3 L1.2 Requirements

在L1.2期间必须维护所有的Link和PHY状态，或者在退出时必须使用特定于实现的方法恢复，并且从L1.2退出时的LTSSM和相应的Port state必须与L1.0的LTSSM和Port state无法区分。

L1.2有不适用于L1.1的附加要求。这些要求在本节中记录。

L1.2有三个子状态，定义如下(见图5-14)。



#### 5.5.3.1 L1.2.Entry

L1.2.Entry是进入L1.2时的过渡状态，以便有时间关闭Refclk，并确保两个端口都观察到clkreq# deasserted。以下规则适用于L1.2.Entry:

•上游和下游端口继续保持公共模式。

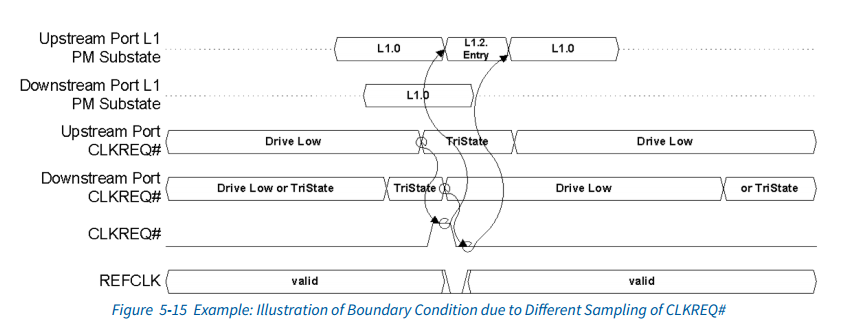
•上游和下游端口都可以关闭其电气空闲(EI)出口检测电路。

•在这种状态下，上游和下游端口不能断言CLKREQ#。

•在TL1O\_REFCLK\_OFF中必须关闭Refclk。

•如果断言CLKREQ#，则下一个状态为L1.0，否则下一个状态为L1.2。等待TPOWER\_OFF后空闲。

请注意，当一个端口断言clkreq#后不久，另一个端口解除断言clkreq#，但在第一个端口观察到clkreq#解除断言之前，会出现一个边界条件。这是一个不可避免的边界条件，实现必须正确处理。示例如图5-15所示。



#### 5.5.3.2 L1.2.Idle

当要求进入L1.2.Idle (参见5.5.1节)满足后，端口进入L1.2.Idle子状态。以下规则适用于L1.2.Idle：

•上行和下行端口都可以关闭任何活动逻辑，包括维持共模所需的电路。

•上游和下游端口的PHY都可能断电。

以下规则适用于使用基于CLKREQ#的机制时的L1.2.Idle:

•如果上游或下游端口需要退出L1.2，则必须在确保满足TL1.2后断言CLKREQ#。

•如果下游端口从L1开始退出，则必须断言CLKREQ#，直到链路退出恢复。上游端口必须在进入Recovery时断言clkreq#，并且必须继续断言clkreq#，直到下一个进入L1的表项，或者其他允许clkreq#解除断言的状态。

•如果上游端口从L1开始退出，它必须继续断言clkreq#，直到下一个进入L1，或者其他允许clkreq#解除断言的状态。

•上游和下游端口都必须监控clkreq#输入信号的逻辑状态。

•下一个状态是L1.2。如果断言了clkreq#，则退出。

#### 5.5.3.3 L1.2.Exit

这是从L1.2退出时的过渡状态，以便两个设备都有时间上电。在L1.2退出时需遵循以下原则:

•上行端口和下行端口的物理电源均已上电。

•不能假定已维护了共模。

##### 5.5.3.3.1 Exit from L1.2

•以下规则适用于使用基于CLKREQ#的机制的L1.2.Exit:

•上游和下游端口必须上电L1.0所需的任何电路，包括维持共模所需的电路。

•在此状态下，上游和下游端口不能改变其clkreq#的驱动状态。

•Refclk必须不早于TL10\_REFCLK\_ON最小时间打开，并且可能需要根据端点发布的LTR允许的时间量才能生效。

•等待TPOWER\_ON后，下一个状态为L1.0。

◦普通模式允许在L1.0阶段被动建立，在Recovery阶段主动建立。为了确保公共模式已经建立，下游端口必须保持一个定时器，下游端口必须继续发送TS1训练序列，直到至少TCOMMONMODE已经过去，因为下游端口已经开始传输TS1训练序列，并在配置的链路的任何Lane上检测到电空闲出口。

图5-16给出了L1.2Entry和Upstream Port initiated exit的信号关系和时序约束。

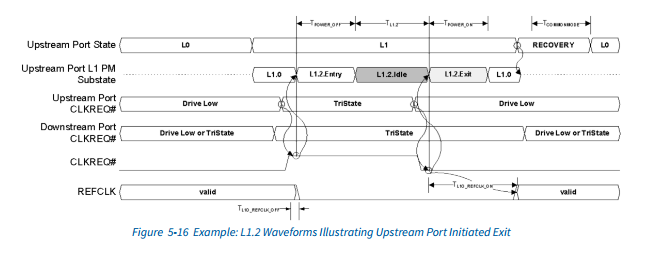
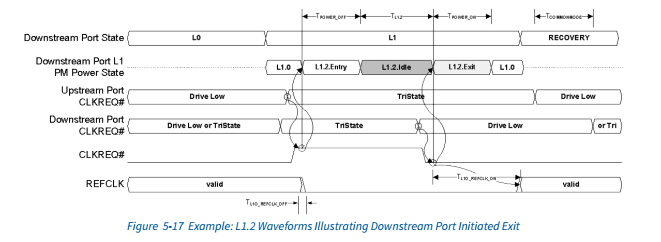


图5-17说明了L1.2Entry和Downstream Port initiated exit的信号关系和时序约束。



### 5.5.4 L1 PM子状态配置

当与该端口相关的ASPM L1.1 Enable、ASPM L1.2 Enable、PCI-PM L1.1 Enable和PCI-PM L1.2 Enable位的任意组合被设置时，则认为该端口上的L1 PM Substates已使能。

只有当一条链路的上下行端口都设置了相应的支持能力位时，该链路的上下行端口才必须设置L1 PM子状态使能位，否则行为未定义。

任何使能位的设置都必须先在下行端口进行，然后才允许在上行端口设置相应的位。如果L1 PM Substates的使能位稍后要清除，则必须先清除上游端口的使能位，然后再允许下游端口清除相应的使能位。

如果设置了ASPM L1 PM子状态的使能位中的一个或两个，则在ASPM L1未使能的情况下，两个端口都必须按照本节的说明进行配置。

如果为PCI-PM L1 PM Substates设置一个或两个使能位，则必须在D0中按本节所述配置两个端口。

在设置L1.2的使能位之前，必须设置TPOWER\_ON、Common\_Mode\_Restore\_Time的值，如果要设置ASPM L1.2的使能位，则必须编程LTR\_L1.2\_THRESHOLD (Value和Scale字段)。

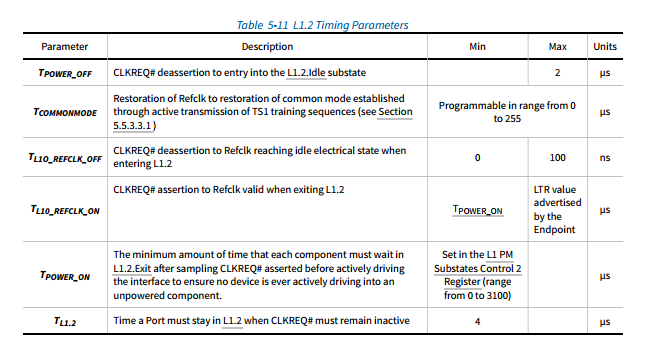
TPOWER\_ON和Common\_Mode\_Restore\_Time字段必须根据连接这两个组件所使用的组件和交流耦合电容，编程为合适的值。这些值的确定是特定于设计实现的。

当ASPM L1.2 Enable和PCI-PM L1.2 Enable位同时被清除时，不需要编程TPOWER\_ON、Common\_Mode\_Restore\_Time和LTR\_L1.2\_THRESHOLD Value和Scale字段，并且硬件不能依赖这些字段来获得任何特定的值。

在编程LTR\_L1.2\_THRESHOLD Value和Scale字段时，两个端口必须编程相同的值。

### 5.5.5 L1 PM子状态定时参数

表5-11定义了与L1.2子状态机制相关的时序参数。



### 5.5.6链路激活

链路激活是暂时禁用L1子状态的可选机制。链路激活用于使链路脱离L1.1/L1.2，避免潜在的停机。这种停顿的一个例子是与执行D3Hot到D0转换的Configuration Write相关联的停顿。链路激活也可以用于间接指示设备，在延迟敏感或时间关键操作期间，它应该避免长时间的内部电源管理。

以下规则适用于链路激活:

•允许下行端口支持链路激活，由L1 PM子状态能力寄存器中的链路激活支持位表示。

•链路激活控制位必须对端口行为没有影响，除非设置了以下一个或多个位:

◦PCI-PM L1.2 Enable

◦PCI-PM L1.1 Enable

•当链路激活控制位设置时，即将进入L1的端口必须断言，当L1处于断言状态时，clkreq#信号保持断言状态。

•如果链路激活控制位为Clear，链路激活机制不会对clkreq#信号的状态施加任何额外的要求。

•如果端口启用了使用MSI或MSI- x的边缘触发中断信令，则必须在以下条件的逻辑与从FALSE转换为TRUE时发送中断消息:

◦相关向量被解除掩码(如果MSI不支持PVM则不适用)

◦链路激活中断启用位设置

◦链路激活控制位设置

◦链路激活状态位设置。注意，链路激活中断总是使用由PCI Express能力寄存器中的中断消息号字段指示的MSI或MSI- x向量。

•如果端口为使用INTx消息的电平触发中断信令启用，只要满足以下条件，无论何时都必须断言虚拟INTx线:

◦命令寄存器中的中断禁用位为Clear。

◦设置链路激活中断使能位;

◦设置链路激活控制位;

◦设置链路激活状态位

•每次以下条件的逻辑与从FALSE转换为TRUE时，链路激活状态位必须设置:

◦PCI-PM L1.2使能位或PCI-PM L1.1使能位(或两者都设置)设置

◦链路激活控制位设置

◦链路不在L1子状态

## 5.6辅助电源支持

与辅助电源相关的具体定义和要求是特定于外形因素的，术语“辅助电源”和“Vaux”应参考使用的特定外形因素来理解。提供辅助电源的具体机构在本规范中未作规定。下面的文本定义了适用于所有形式因素的需求。

PCI Express PM在设备控制寄存器中提供了一个辅助电源PM使能位，该位提供了使功能能够绘制最大辅助电流的方法，而不依赖于其对PME产生的支持水平。

函数通过在PMC寄存器的Aux\_Current字段中指定一个非零值来请求辅助电源分配。参考第7章的Aux Power PM使能寄存器位分配和访问机制。

辅助功率PM使能的辅助功率分配确定如下:

辅助电源PM Enable = 1b:

辅助电源根据PMC寄存器的Aux\_Current字段的请求分配，独立于PMSCR中的PME\_En位。PME\_En位仍然控制着掌握PME的能力。

辅助电源PM Enable = 0b:

辅助电源分配由章节7.5.2.2中定义的PME\_En位控制。

辅助电源PM使能位是粘性的(见7.4节)，所以它的状态保留在D3Cold状态，并且不受从D3Cold状态到D0uninitialized状态转换的影响。

## 5.7电源管理系统消息和DLLP

表5-12定义了每个PM包在PCI Express堆栈中的位置。



有关电源管理dllp结构的信息，请参见第3.5节。

电源管理消息遵循所有消息的通用规则。Power Management Message字段的设置规则如下:

•长度字段保留。

•属性字段必须设置为默认值(全部为0)。

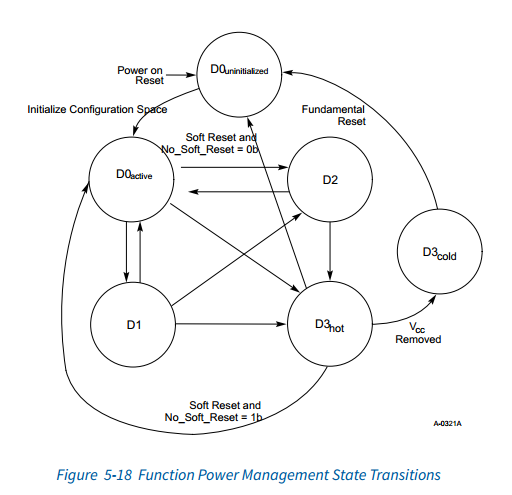
•保留地址字段。

•请求者ID -参见2.2.8.2节中的表2-20。

•流量分类字段必须使用默认的分类(TC0)。

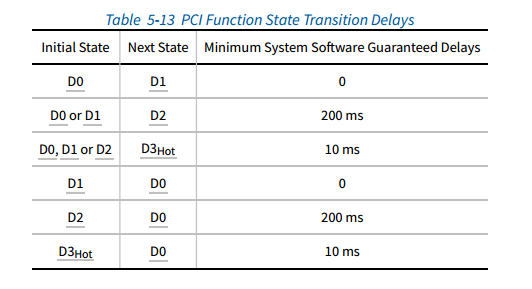
## 5.8 PCI功能电源状态转换

所有PCI-PM电源管理状态的变化都是由软件显式控制的，除了基本重置，它使所有功能恢复到未初始化状态。图5-18显示了所有支持的状态转换。未标记的弧线表示软件启动的状态转换(设置电源状态操作)。



## 5.9状态转换恢复时间要求

表5-13显示了系统软件必须允许的最小恢复时间，从一个函数被编程改变状态到该函数下一次被访问(包括配置空间)，除非准备就绪通知(见章节6.23)被用来向系统软件指示修改的值。对于桥接函数，这个延迟也构成了桥接状态改变和逻辑总线上的任何函数可以被访问之间的最小延迟。



## 5.10 PCI桥和电源管理

由于电源管理是在操作系统的指导下进行的，因此每一类函数都必须有一个明确定义的标准，以确定功能可用性，以及在每种电源管理状态下运行时必须保留哪些功能上下文。一些示例设备类规范已经被提议作为ACPI规范的一部分，用于从音频到网络外接卡的各种功能。虽然为大多数函数定义特定于设备类的行为策略超出了本规范的范围，但为PCI桥接函数定义所需的行为属于本规范的范围。这里的定义适用于所有三种类型的PCIe桥:

• Host bridge, PCI Express to expansion bus bridge, or other ACPI enumerated bridge

• Switches

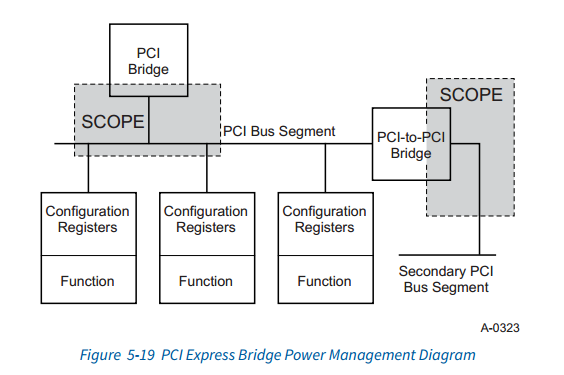
• PCI Express to PCI bridge

• PCI-to-CardBus bridge

控制这些函数状态的机制在一定程度上取决于存在哪种类型的发起设备。下面的部分描述这些机制是如何在三种类型的桥上工作的。

介绍PCI Express Bridge功能的电源管理策略。PCI快速桥接功能可以被描述为具有次级总线下游的始发设备。介绍网桥功能的电源管理状态与其从母线的电源管理状态之间的关系。

图5-19中的阴影区域说明了本节讨论的内容。



从图5-19可以看出，从操作系统的角度来看，本章描述的PCI Express Bridge行为对于主机网桥、交换机以及PCI Express到PCI网桥都是通用的。

系统软件的责任是确保对于给定的总线和驻留在该总线上的所有功能，只使用有效的、可行的总线和下游功能电源管理状态组合。

### 5.10.1交换机和PCI Express转PCI bridge

交换机或PCI Express到PCI桥的次要总线的电源管理策略与为任何桥功能定义的策略相同。

桥函数的PMCSR\_BSE寄存器中的BPCC\_En和B2\_B3#总线电源/时钟控制字段支持与任何其他桥相同的功能。

## 5.11电源管理事件

有两种类型的电源管理事件:

•唤醒事件

•PME生成

Wakeup事件用于请求打开电源。

PME生成事件用于向系统标识请求上电的函数。

在传统的PCI中，这两个事件都与pme#信号相关。pme#信号由函数断言，以请求更改其电源管理状态。当设置了PME\_En位并且事件发生时，函数设置PME\_Status位并断言pme#信号。它保持pme#信号断言，直到PME\_En位或PME\_Status被清除(通常由软件)。

在PCI Express中，Wakeup事件与wake#信号相关联。如果支持，WAKE#信号在相关的外形规格中定义，当功能处于D3Cold和PME\_En设置时，函数使用该信号来请求更改其PCI-PM电源管理状态。

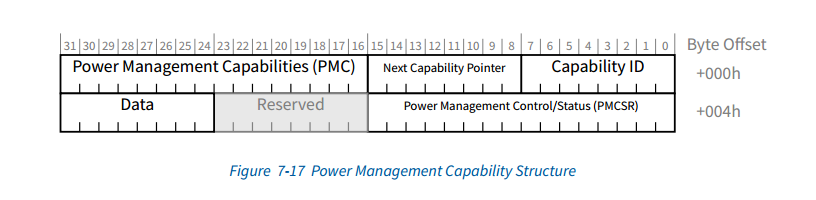
在PCI Express中，当主电源恢复并训练链路后，发起唤醒的函数(例如，断言wake#的函数)会向根Complex发送一个PM\_PME消息。PM\_PME消息向根复合体提供请求函数的标识，而不需要软件轮询正在设置的PME\_Status位。

**Configuration**

7.5.2 PCI电源管理能力结构

本节描述组成PCI电源管理接口结构的寄存器。

PCI Power Management Capability架构的组织结构如图7-17所示。这种结构是所有PCI Express函数所必需的。



注意:8位数据寄存器(偏移量07h)对于Type 0和Type 1函数都是可选的。

需要PCI Express设备功能支持D0和D3设备状态;由于PCI Express的PME消息传递的带内性质，需要在7.1节中描述的PCI-PCI桥结构表示PCI Express端口来指示PME消息传递能力。

然而，只有当PCI-PCI桥函数本身生成PME时，才会设置表示PCI Express Ports的PCI-PCI桥结构的PME\_Status位。当网桥传播PME消息，但是PCI-PCI桥函数本身没有在内部生成PME时，PME\_Status位没有设置。