PCIE总结

目录

[PCIE接口 2](#_Toc207218206)

[PCIE和PCI的区别 2](#_Toc207218207)

[PCIE物理插口 2](#_Toc207218208)

[PCIE配置空间 3](#_Toc207218209)

[如何访问PCIe设备内存空间？ 4](#_Toc207218210)

[系统如何为PCIE设备分配内存映射空间 5](#_Toc207218211)

[系统如何读取Configuration空间? 7](#_Toc207218212)

[枚举流程 8](#_Toc207218213)

[通信模型 8](#_Toc207218214)

[PCI设备读系统内存 8](#_Toc207218215)

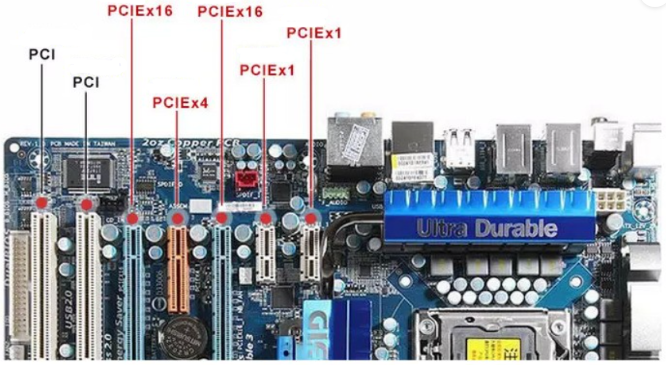
[CPU向设备内存写数据 9](#_Toc207218216)

[CPU、PCH的PCIe实现及相关需求：IIO、Bifurcation、带宽自适应、BDF上报 9](#_Toc207218217)

[BDF格式： 10](#_Toc207218218)

[PCIe相关的RAS特性：AER、UCE/CE 10](#_Toc207218219)

# PCIE接口

PCIE接口的全称是Peripheral Component Interconnect Express,是一种高速串行计算机扩展总线标准。它原来的名字叫“3GIO”，是由Intel在2001年提出，旨在替代旧的PCI，PCI-X，AGP总线标准。

PCIE属于高速串行点到点双通道高带宽传输，所连接的设备独享通道带宽，支持主动电源管理、错误报告、端对端的可靠性传输、热插拔以及服务质量等功能。

PCIE交由PCI-SIG认证发布后才改名为“PCI-Express”，简称PCIE。它的主要优势就是数据传输速率高，而且还有相当大的发展潜力。

# PCIE和PCI的区别

在概念上，PCIE总线是较旧的PCI/PCIX总线的高速串行替换。PCIE总线与旧PCI之间的主要区别之一是总线结构。PCI使用共享并行总线结构，其中PCI主机与所有设备共享一组地址线、数据线和控制线。相比之下，PCIE总线基于点到点结构，单独的串行链路将每个设备分别连接到主板。由于PCIE的独享总线结构，可以对单方向上的PCI总线进行限制，并且一次限制为一个主机。此外，PCI时钟方案将总线时钟设置为总线上最慢设备的时钟速度。相比之下，PCIE总线支持任意两个设备间的全双工通信，多个设备同时的并发访问没有固定的限制。

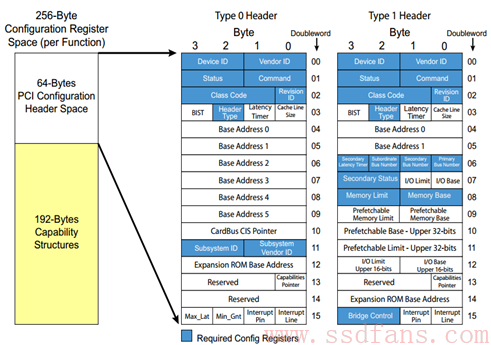
# PCIE物理插口

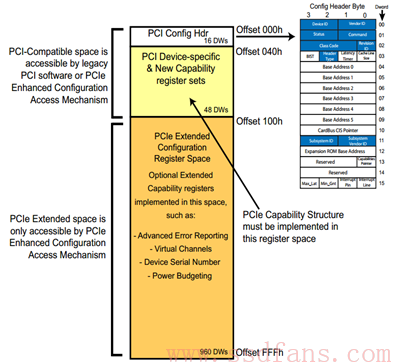
一条PCIE线缆可以将其插入更大的PCIE插槽。各种PCIE插槽如下图所示，最大的PCIE插槽为x16

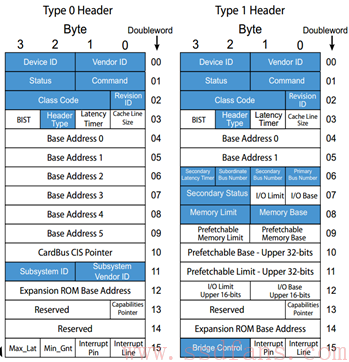
此外，M.2插口也是借助PCIE总线来传输数据的，其能够插入固态硬盘或者无线网卡。虽然M.2插口和PCIE插口外观不一样，但M.2插口依然借助PCIE总线传输数据。现在最新的M.2插口支持PCIE 5.0版本。

PCIE的基本结构包括根组件（Root Complex）、交换器（Switch）和各种终端设备（Endpoint）。

# PCIE配置空间

每个PCIE设备都有一段内存空间，主机系统可以读取这段内存来获得该设备的信息，也可以通过这段内存来配置该设备，这段内存空间就叫PCIE的配置空间。不同于其他设备的配置空间，PCIE设备的配置空间是规定好的。早在PCI和PCI-X时代就有配置空间的概念，那时的配置空间如右图。整个配置空间是一系列寄存器的集合，其中Type 0是终端设备的配置，Type 1是Bridge（PCIe时代的交换器）的配置。Type 0和Type 1都由两部分组成：64 字节的Header和192字节的Capability，后者包含设备的功能。

进入PCIe时代，192 字节不足以容纳设备的功能。为了后向兼容，PCIe扩展了Capability的空间，整个配置空间由256 字节扩展到了成4KB。在这4KB中，前面256 字节保持不变。如右图白色和浅黄色部分是原PCI配置空间，橙色部分是PCIE拓展的空间。



右上图中最核心的部分是前64字节的PCI Config Hdr白色部分，即Header。Header具体内容如右图所示。

第一行的Device ID，Vendor ID，Class Code和Revision ID都是只读寄存器，PCIe设备通过这些寄存器告诉主机系统，这是哪个厂家的设备、设备ID是多少、以及是什么类型的设备。

在Header中最重要的属BAR（Base Address Register）。对设备配置（Type 0）提供了最多6个BAR，而对交换器（Type 1）只提供了2个BAR。

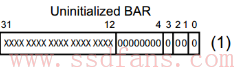
## 如何访问PCIe设备内存空间？

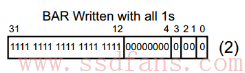
由于CPU只能访问系统内存空间，不能访问PCIe设备内存空间，所以解决办法是在CPU和PCIe中间添加一个根组件Root Complex (RC)。CPU如果要访问某个设备内存，先令RC把数据从设备内存读到系统内存，然后CPU就可以从系统内存读数据。如果CPU要往外设内存写数据，则先把数据存到系统内存中，然后令RC将数据写入到设备内存。

具体实现就是上电的时候，系统把开放的（系统可见的）设备内存映射到系统内存。CPU要访问PCIe设备内存，只需访问对应的系统内存。RC检查该系统内存地址，如果发现该系统内存地址是某个PCIe设备内存的映射，就会产生TLP，去访问对应的PCIe设备，读取或者写入PCIe设备。

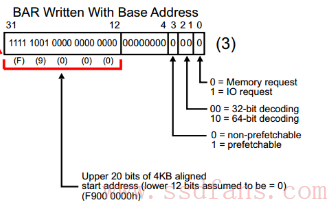
一个PCIe设备，可能有若干个内部空间（有些可预读，有些不可预读）需要映射到系统内存空间，设备出厂时，这些空间的大小和属性都写在Configuration BAR寄存器里面。上电后，主机系统读取这些BAR寄存器，分别为其分配对应的系统内存空间，并把相应的内存基地址写回到BAR寄存器。BAR寄存器的地址实际上是PCI总线的地址，CPU访问的是内存的地址，CPU访问PCIe设备时，需要把总线地址转换成内存地址。

## 系统如何为PCIE设备分配内存映射空间

上电时，系统读取PCIE设备的BAR0寄存器。如下图X表示可读可写位，不确定是1还是0；0表示生产商设好的只读位。

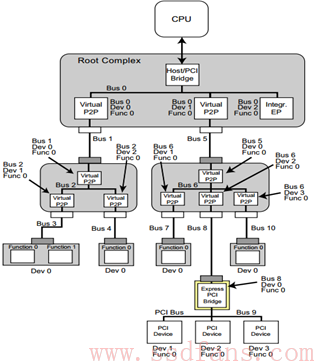
尝试将所有位全部置0，然后尝试全部置1，得到：

由于低12位为只读位，所以没有被置1，这表示该设备要占用2^12字节的空间，即4KB。在这12位中的低4位，表示该存储空间的一些属性：是IO映射还是内存映射、32位地址还是64位地址、能否读取（有些寄存器一旦被读取，其中的数据就会被清除，因此这些寄存器不能被读取）。这些是PCIE设备在出厂前设置好的，用于为系统提供信息。

然后系统根据这些信息，在系统内存找到合适的地方来映射这4KB的空间，把分配的基地址写入到BAR0：

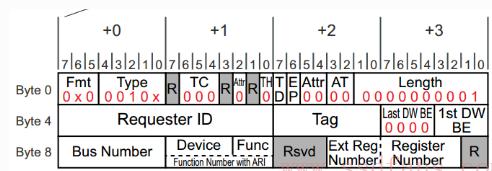
最终完成该PCIe设备内存空间的映射。

一个PCIe设备可能有若干个内部空间需要开放出来，系统依次读取BAR1，BAR2…，直到完成所有内部空间的映射。多个配置空间表示PCIe设备具有多个功能，比如既能当硬盘，还能当网卡，每个功能对应一个配置空间。

在一个PCIe结构里，一条总线下可以有多个设备，而每个设备可以具有多个功能，如右图所示。因此，在整个PCIe系统中，只要知道了Bus、Device和Function，就能找到对应的Function。寻址基本单元是功能（function），它的ID就由Bus、Device和Function组成 （BDF)。一个PCIe系统，最多可以有256条Bus，每条Bus上最多有32个Device，而每个Device最多能实现8个Function，每个Function需要4KB的内存。因此上电的时候，需要占用内存是：256 \* 32 \* 8 \* 4KB = 256MB

## 系统如何读取Configuration空间?

前面讲系统为所有可能的Configuration预留了256MB内存空间。若系统想访问某个Configuration，只需指定相应Function对应的内存地址，RC检查到这个地址是Configuration的映射地址，就会产生相应的Configuration Read TLP去获取相应Function的Configuration。

Configuration Read TLP的格式如右图最后一行所示，Bus Number + Device + Function决定了唯一的设备，Ext Reg Number + Register Number相当于配置空间的偏移。找到了设备地址，然后指定了偏移量，就能找到要访问的位置。

此外，只有RC才能发起Configuration访问请求，其他设备不允许对别的设备进行Configuration读写。

# 枚举流程

PCI枚举是个不断递归发现新设备的过程，简单来说包括以下几个步骤：

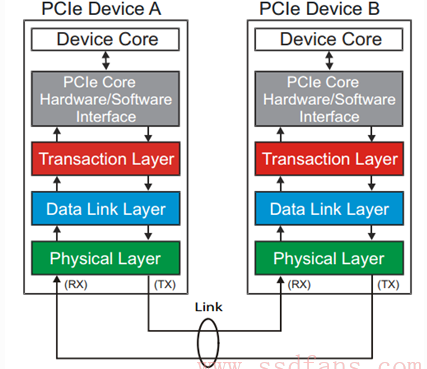
1．深度优先遍历整个PCI设备树。从Root Complex出发，寻找设备和桥。发现桥后设置Bus,表示发现一个PCI设备子树，回到第1步

2．递归的过程中通过读取BARs，记录所有MMIO和IO的需求情况并予以满足。

3．设置必要的Capabilities

整个过程结束后，一棵完整的PCI设备树就建立好了。

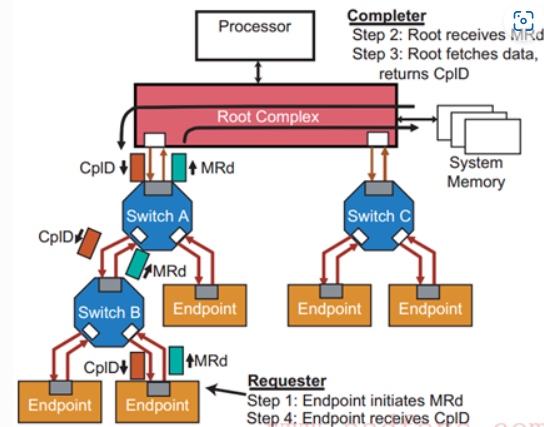
## 通信模型

**事务层**的主要职责是创建（发送）和解析（接收）TLP (Transaction Layer packet)，流量控制，QoS，事务排序等。

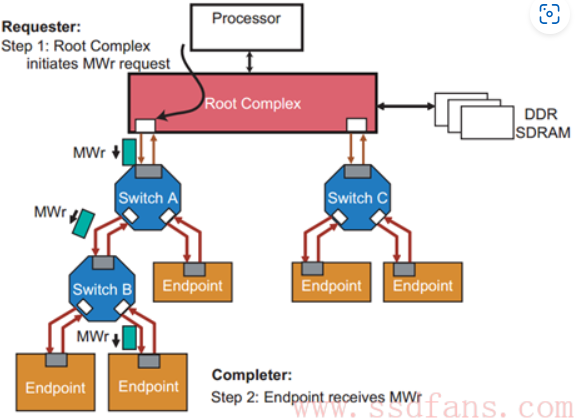
**数据链路层**的主要职责是创建（发送）或者解析（接收）DLLP(Data Link Layer packet)，Ack/Nak协议（链路层检错和纠错），流控，电源管理等。

**物理层**的主要职责是处理所有的Packet数据物理传输，发送端数据分发到各个Lane传输（stripe），接收端把各个Lane上的数据汇总起来（De-stripe），每个Lane上加扰（Scramble，目的是让0和1分布均匀，去除信道的电磁干扰EMI）去扰（De-scramble)，以及8/10或者128/130编码解码等。

### PCI设备读系统内存

如右图，若Switch B下面的某个Endpoint想读系统内存的数据，它会在事务层上生成一个Memory Read TLP并发送出去，该MRd向上经过Switch B和Switch A，最终到达RC。RC收到这个请求后到系统内存中取到对应的数据，再通过Completion with Data TLP（CplD)返回数据，原路返回到开始的Endpoint

### CPU向设备内存写数据

如右图，处理器想往某个Endpoint写入数据，因此RC在其事务层生成一个Memory Write TLP并发送，经过Switch A和Switch B，最终到达目的地。由于Memory Write TLP是Posted的，因此Endpoint收到数据后不需要返回Completion TLP

## CPU、PCH的PCIe实现及相关需求：IIO、Bifurcation、带宽自适应、BDF上报

现代CPU通常集成了多个PCIe通道，这些通道直接连接到各种外设，如显卡、存储设备和网络接口卡等。这些通道的数量和版本取决于CPU具体型号。

**高性能**：CPU直接提供的PCIe通道通常用于高性能设备，因为它们可以提供最低的延迟和最高的带宽。

**可配置性**：这些通道可以通过BIOS或固件进行配置，以满足不同系统需求。

PCH是一个集成芯片，负责处理与CPU以外的大部分I/O操作。PCH也提供了多个PCIe通道，但这些通道通常用于较低带宽需求的设备，如USB控制器、音频控制器和网卡等。

**多功能性**：PCH提供了大量的I/O接口，包括PCIe、SATA、USB等。

**扩展性**：PCH的PCIe通道可以用于扩展系统功能，连接更多外设。

**IIO** 是指集成在CPU或PCH中的I/O控制单元。它负责管理和调度所有I/O操作，包括PCIe数据传输。

主要功能：

**数据路由**：管理数据在CPU、内存和外设之间的传输。

**中断处理**：处理来自各种外设的中断请求。

**资源分配**：分配和管理系统资源，如内存地址和I/O端口。

**Bifurcation** 是指将一个PCIe插槽分割成多个较小的插槽，以支持多个设备。例如，一个x16插槽可以分割成两个x8插槽或四个x4插槽。这种技术允许更灵活地使用有限数量的PCIe通道，特别是在需要连接多个设备时。

应用场景：

**服务器和工作站**：常用于高密度服务器和工作站，以最大化设备连接数量。

**多GPU配置**：在多GPU系统中，可以通过Bifurcation将一个x16插槽分割成两个x8插槽，从而支持两个GPU。

**带宽自适应** 是指根据当前负载动态调整PCIe链路带宽，以优化性能和功耗。这种技术可以在不需要高带宽时降低链路速度，从而节省功耗，在需要高带宽时自动提升链路速度以满足需求。

优点：

**功耗优化**：通过动态调整带宽，可以在低负载时节省电能。

**性能优化**：在高负载时自动提升带宽，以确保系统性能。

**BDF上报** 是指通过PCI配置空间报告每个PCI设备的总线号、设备号和功能号。这些信息用于唯一标识系统中的每个PCI设备，并进行资源分配和管理。

## BDF格式：

**Bus Number（总线号）**：标识哪个总线上的设备。

**Device Number（设备号）**：标识总线上的哪个设备。

**Function Number（功能号）**：标识设备中的哪个功能单元，一个物理设备可能有多个功能单元。

**资源管理**：操作系统使用BDF信息来分配内存地址空间、中断请求等资源。

**故障排除**：BDF信息有助于识别故障设备，便于维护和修理。

1. **CPU 和 PCH 的 PCIe 实现**
   * CPU 提供高性能、高带宽的直接PCIe通道，用于连接关键外设。
   * PCH 提供更多、更灵活的I/O接口，用于扩展系统功能。
2. **IIO（Integrated I/O）**
   * 管理所有I/O操作，包括数据路由、中断处理和资源分配。
3. **Bifurcation**
   * 将一个大容量PCIe插槽分割成多个较小插槽，以支持更多设备连接，特别是在服务器和多GPU配置中应用广泛。
4. **带宽自适应**
   * 根据负载动态调整链路带宽，优化功耗与性能平衡。
5. **BDF（Bus, Device, Function）上报**
   * 用于唯一标识并管理每个PCI设备，确保系统资源有效分配与故障排除。

## PCIe相关的RAS特性：AER、UCE/CE

在PCIe（Peripheral Component Interconnect Express）协议中，**RAS（Reliability, Availability, Serviceability）** 特性是确保系统可靠性、可用性和可维护性的关键功能。以下是PCIe相关的主要RAS特性，包括AER（Advanced Error Reporting）、UCE（Uncorrectable Error）和CE（Correctable Error）的详细解释：