1. PCI Express 基础

PCIe 总线是基于 PCI 总线发展起来的，很多基本概念都来自于PCI 总线，有必要在介绍 PCIe 之前了解 PCI 总线。

1.1 PCI 基础

PCI 总线作为处理器系统的局部总线，其主要目的是为了连接外部设备，而不是作为处理器系统的系统总线连接 Cache 和主存储器。PCI 总线作为系统总线的延伸，其设计考虑了许多与处理器相关的内容，孤立的研究PCI 总线并不可取，因此需要将PCI 作为存储器系统的一个部分来研究。

1.1.1 几个重要概念

1) PCI 总线空间与处理器空间隔离

PCI 设备具有独立的地址空间，即PCI 总线地址空间，该空间与存储器地址空间通过HOST 主桥隔离。处理器需要通过 HOST 主桥才能访问 PCI 设备，而 PCI 设备需要通过 HOST 主桥才能方位主存储器。

要注意区分存储器地址空间和 PCI 总线地址。在一个处理器系统中，存储器域、PCI 总线域与 HOST 主桥的关系如下图。

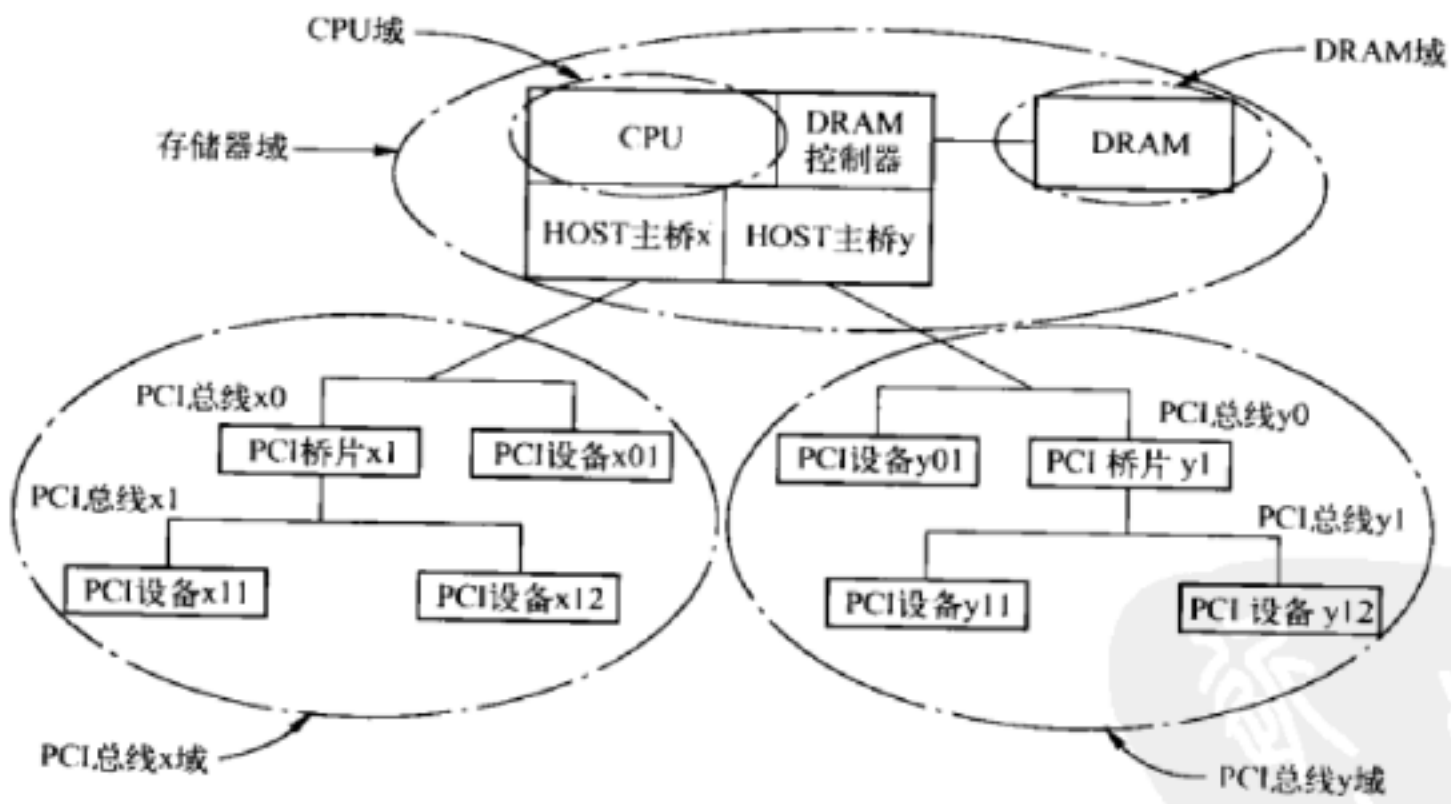


图 2-1 存储器域与 PCI 总线域的划分

图中的处理器系统由一个 CPU、一个DRAM 控制器和两个 HOST 主桥组成。在这个处理器系统中，包含 CPU 域、DRAM 域、存储器域和 PCI 总线域地址空间。其中 HOST 主桥 x 和 HOST 主桥y 分别管理 PCI 总线 x 域与 PCI 总线y 域。CPU 访问 PCI 设备，必须通过 HOST 主桥进行地址转换，而 PCI 设备访问存储器设备，也需要 HOST 主桥进行地址转换。HOST 主桥

的一个重要作用就是将存储器访问的存储器地址转换成PCI 总线地址。

CPU 域地址空间是指 CPU 所能直接访问的地址空间集合。

DRAM 域地址空间是指 DRAM 控制器所能访问的地址空间集合，又称为主主存储器域。

存储器域是 CPU 域和 DRAM 域的集合。存储器域包括 CPU 内部的通用寄存器、存储器映射寻址的寄存器、主存储器空间和外部设备空间。在Intel 的 x86 处理系统中，外部设备空间与 PCI 总线域地址空间等效。因为在x86 处理器系统中，使用 PCI 总线同一管理全部外部设备。

值得注意的是，存储器域的外部设备空间，在PCI 总线域中还有一个地址映射。当处理器访问 PCI 设备时，首先访问的是这个设备在存储器域上的PCI 设备空间，之后 HOST 主桥将这个存储器域的 PCI 总线地址转换成 PCI 总线域的物理地址，然后通过 PCI 总线事务访问PCI 总线域的地址空间。

2) 可扩展性

PCI 总线具有很强的扩展性。在PCI 总线中，HOST 主桥可以直接推出一条 PCI 总线，这条总线也是该 HOST 主桥管理的第一条 PCI 总线，该总线还可以通过 PCI 桥扩展一系列 PCI 总线，并以 HOST 主桥作为根节点，形成 1 棵 PCI 总线树。这些 PCI 总线都可以连接 PCI 设备，但是一棵 PCI 设备树上，最多只能挂接 256 个 PCI 设备（包括 PCI 桥）。

3) 动态配置机制

PCI 设备使用的地址可以根据需要由系统软件动态分配。PCI 总线使用这种方式合理地解决设备间的地址冲突，从而实现了“即插即用”功能。每一个PCI 设备都有独立的配置空间，在配置空间中包含该设备在 PCI 总线中使用的基地址即 BAR 地址，从而保证每一个 PCI 设备使用的物理地址并不相同。PCI 桥的配置空间中包含有其下 PCI 子树所能使用的地址范围。

x86 系统的工作流程是：主板上的 BIOS 程序会扫描 PCI/PCIE 设备，读取其 BAR 空间的大小，动态地为 PCI/PCIE 设备分配地址空间。在调试过中发现，假如将 BAR 空间设置成 2G，x86 系统会报 no bootable device 的错误，原因应该是 BIOS 给 PCIE 设备分配了 2G 的地址空间，暂用了硬盘的地址空间，导致无法加载操作系统。

4) 总线带宽

PCI 总线与之前的局部总线相比，极大提高了数据传送带宽，32 位/33MHz 的 PCI 总线可以提供132MB/s 的峰值带宽，而64 位/66MHz 的PCI 总线可以提供的峰值带宽为532MB/s。虽然 PCI 总线所能提供的峰值带宽远不能和PCIe 总线相比，但是与之前的局部总线 ISA、EISA 和 MCA 总线相比，仍然具有极大的优势。

ISA 总线的最高主频为 8MHz，位宽为16，其峰值带宽为16MB/s；EISA 总线的最高主频为 8.33MHz，位宽为 32，其峰值带宽为 33MB/s；而 MCA 总线的最高主频为 10MHz，最高位宽为 32，其峰值带宽为 40MB/s。PCI 总线提供的峰值带宽远高于这些总线。

表 1-1 PCI 总线频率、带宽与负载之间的关系

总线类型	总线频率/MHz	峰值带宽/（MB/s）	负载能力
PCI	33	133	4～5 个插槽
	66	266	1～2 个插槽
PCI-X	66	266	4 个插槽
	133	533	2 个插槽
	266	1066	1 个插槽
	533	2131	1 个插槽

5) 共享总线机制

PCI 设备通过仲裁获得 PCI 总线的使用权后，才能进行数据传送，在PCI 总线上进行数据传送，并不需要处理器进行干预。PCI 总线仲裁器不在 PCI 总线规范定义的范围内，也不一定是 HOST 主桥和 PCI 桥的一部分，虽然绝大多数 HOST 主桥和 PCI 桥都包含 PCI 总线仲裁器，但是在某些处理器系统设计中也可以使用独立的PCI 总线仲裁器。

PCI 设备使用共享总线方式进行数据传递，在同一条总线上，所有PCI 设备共享同一总线带宽，这将极大地影响 PCI 总线的利用率。这种机制显然不如 PCIe 总线采用的交换结构。

6) 中断机制

PCI 总线上的设备可以通过四根中断请求信号INTA~D#向处理器提交中断请求。与 ISA 总线上的设备不同，PCI 总线上的设备可以共享这些中断请求信号，不同的PCI 设备可以将这些中断请求信号线与后，与中断控制器的中断请求引脚连接。PCI 设备的配置空间记录了该设备使用这四根中断请求信号的信息。

PCI 总线还进一步提出了 MSI（Message Signal Interrupt）机制，该机制使用存储器写总线事务传递中断请求，并可以使用 x86 处理器 FSB（Front Side Bus）总线提供的 Interrupt Message 总线事务，从而提高了 PCI 设备的中断请求效率。

1.1.2 PCI 总线的组成结构。

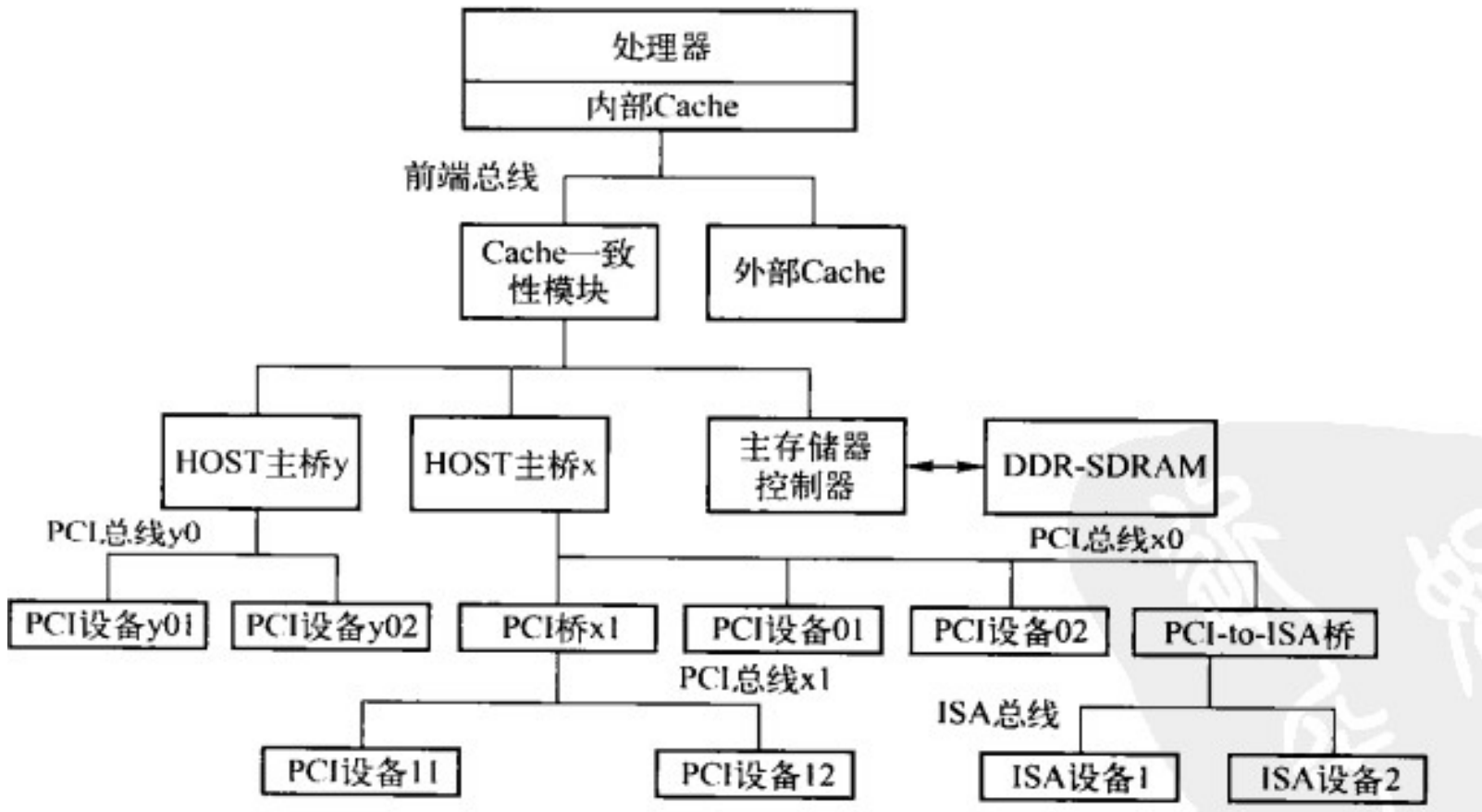


图 1-1 基于 PCI 总线的处理器系统

图中与 PCI 总线相关的模块包括：HOST 主桥、PCI 总线、PCI 桥和 PCI 设备。PCI 总线是由 HOST 主桥和 PCI 桥推出，HOST 主桥与主存储器控制器在同一级总线上，因此 PCI 设备可以方便通过 HOST 主桥访问存储器，即进行 DMA 操作。在一些简单的处理器系统中，可能不包含 PCI 桥，此时所有 PCI 设备都是连接再 HOST 主桥上推出的 PCI 总线上。在一些处理器系统中有可能有多个 HOST 主桥，如图 1-1 所示处理器系统中含有 HOST 主桥x 和 HOST 主桥y。

X86 处理器的 HOST 主桥

X86 处理器使用南北桥结构连接 CPU 和 PCI 设备。其中北桥连接快速设备，如显卡和内存条，并推出 PCI 总线，HOST 主桥包含在北桥中。而南桥连接慢速设备。

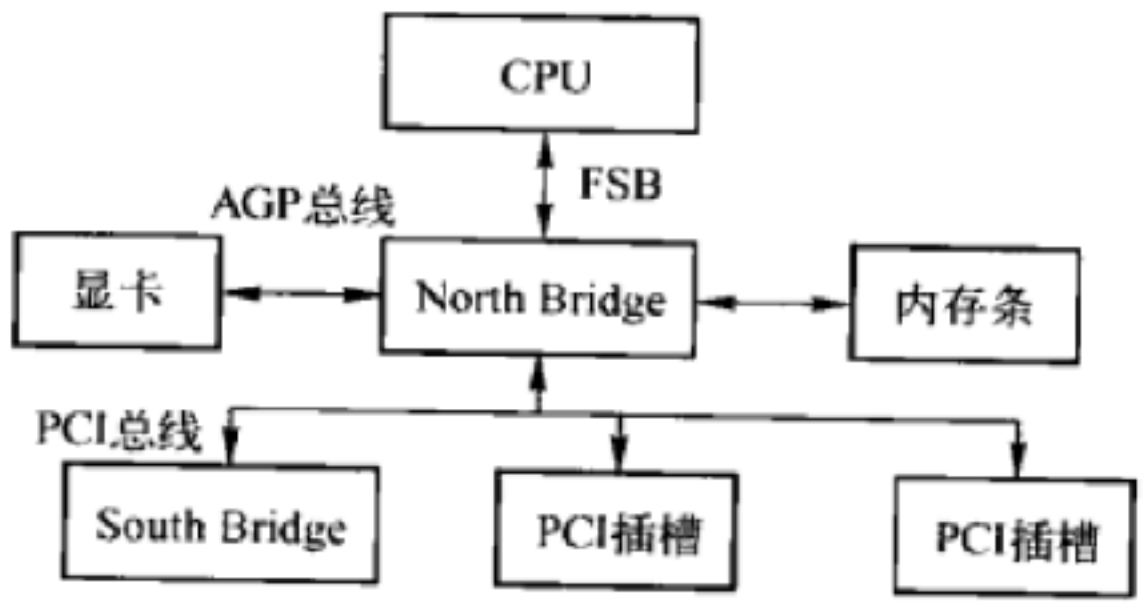


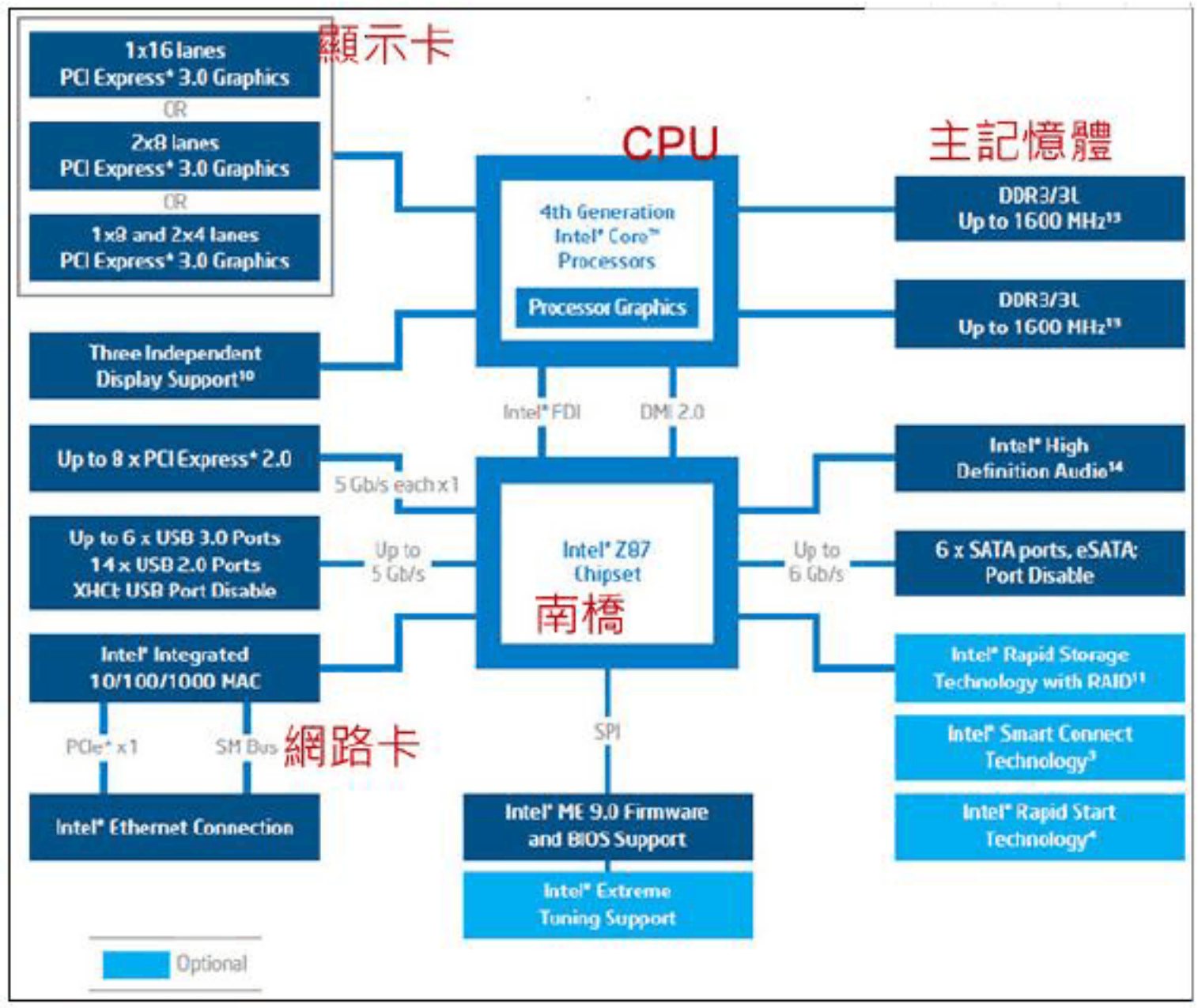
图 2-6　x86 处理器的南北桥结构

1.2 PCIE 总线概述

PCI 总线使用并行总线结构，在同一条总线上的所有外部设备共享总线带宽，而 PCIe 总线使用高速查分总线，采用端对端的连接方式，因此在每一条 PCIe 链路中只能连接两个设备。这使得 PCIe 与 PCI 总线采用的拓扑结构有所不同。PCIe 总线除了在连接方式上与 PCI 总线不同之外，还使用一些在网络通信中使用的技术，如支持多种数据路由方式，基于多通路的数据传递方式，和基于报文的数据传送方式，并充分考虑在数据传送中出现的服务质量QoS（Quality of Service）问题。

1.2.1 基于 PCIe 的系统结构

前期的 Intel 主板中会集成单独的南北桥芯片，北桥负责连接速度较快的 CPU、主存储器以及显卡等元件，南桥负责连接速度较慢的设备，包括硬盘、USB、网卡等。只要CPU 读取主存储器，还需要北桥的支持，也就是 CPU 与主存储器的交流，会占用北桥的带宽。因此新一代的 Intel 主板架构，大多将北桥存储控制器整合到CPU 封装中，CPU 直接与主存储器交互，速度较快。



基于 PCIe 总线的 Intel 处理器架构

主存储器的速度

SDRAM/DDR	型号	数据位宽	内部时钟	频率速度	带宽
SDRAM	PC100	64	100	100	800MBytes/sec
SDRAM	PC133	64	133	133	1064MBytes/sec
DDR	DDR-266	64	133	266	2.1GBytes/sec
DDR	DDR-400	64	200	400	3.2GBytes/sec
DDR	DDR2-900	64	200	800	6.4GBytes/sec
DDR	DDR3-1600	64	200	1600	12.8GBytes/sec

PCIe 总线带宽

规格	1x 带宽	16x 带宽
PCIe1.0	250Mbytes/sec	4GBytes/sec
PCIe2.0	500Mbytes/sec	8GBytes/sec
PCIe3.0	~1GBytes/sec	16GBytes/sec
PCIe4.0	~2GBytes/sec	32GBytes/sec

SATA总线带宽

版本	带宽
SATA1.0	150Mbytes/sec
SATA2.0	300Mbytes/sec
SATA3.0	600Mbytes/sec

USB 总线带宽

版本	带宽
USB1.0	1.5Mbytes/sec
USB2.0	60Mbytes/sec
USB3.0	500Mbytes/sec
USB3.1	1000Mbytes/sec

1.1.1 端到端的数据传递

PCI 总线不同，PCIe 总线采用端到端的连接方式，在一条 PCIe 链路的两端只能各连接一个设备，这两个设备互为数据发送端和数据接收端。

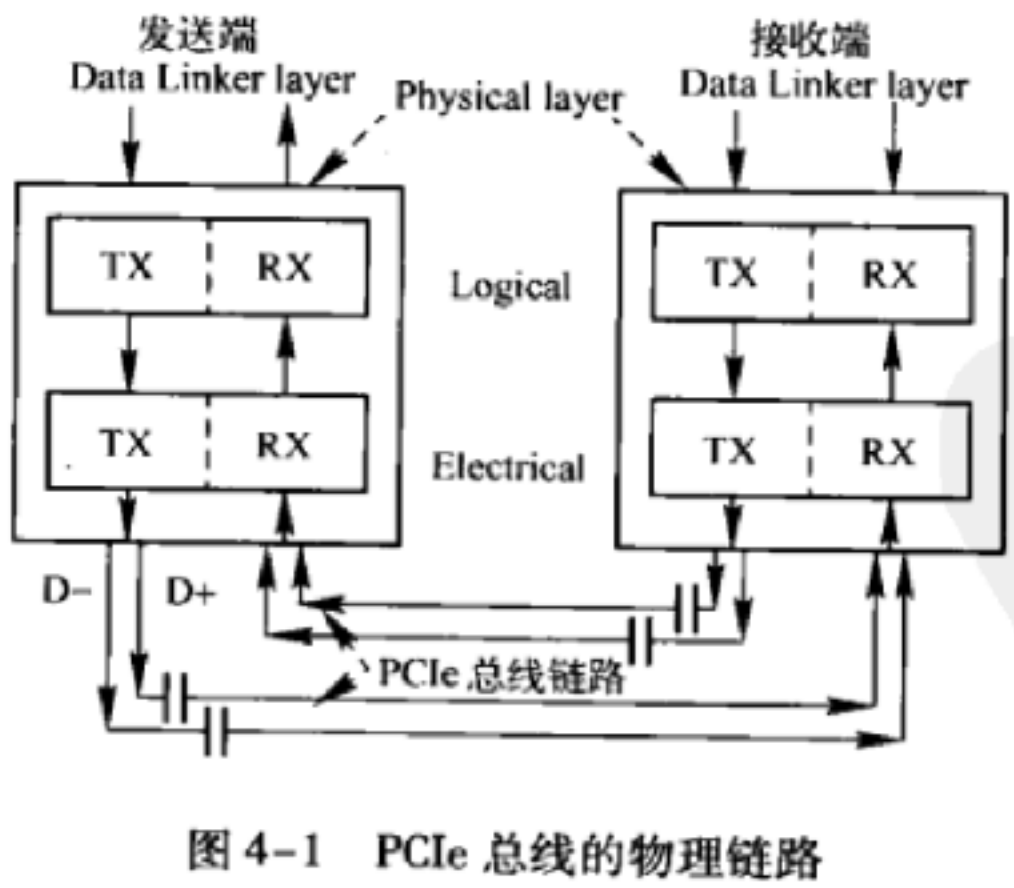


图 4-1 PCIe 总线的物理链路

在 PCIe 总线的物理链路的一个数据通路（Lane）中，有两组查分信号，共 4 根信号线。其中发送端的 TX 部件与接收端的 RX 部件使用一组差分信号连接，该链路也被称为发送端的发送链路，也是接收端的接收链路；而发送端的 RX 部件与接收端 TX 部件使用另一组查分信号连接，该链路也被称为发送端的接收链路，也是接收端的发送链路。

一个 PCIe 链路可以由多个数据通路 Lane 组成，目前 PCIe 链路可以支持 1、2、4、8、16 和 32Lane，即 x1、x2、x4、x8、x16、x32 宽度的 PCIe 链路。

表 4-1 PCIe 总线规范与总线频率和编码的关系

PCIe 总线规范	总线频率①/GHz	单 Lane 的峰值带宽/(GT/s)	编码方式
1. x	1. 25	2. 5	8/10b 编码
2. x	2. 5	5	8/10b 编码
3. 0	4	8	128/130b 编码

① 这里的总线频率指差分信号按照逻辑“0”和“1”连续变化时的频率。

PCIe 总线物理链路间的数据传送使用基于时钟的同步传送机制，但是在物理链路上并没有时钟线，PCIE 总线的接收端含有时钟恢复模块 CDR（Clock Data Recovery），CDR 将从接收报文中提取接收时钟，从而进行同步数据传递，PCIe 设备进行链路训练时将完成时钟的提取

工作。

1.2.2 PCIe 总线的层次结构

PCIe 总线采用串行连接方式，并使用数据包（Packet）进行数据传输。在 PCIe 总线中，数据报文在接收和发送过程中，需要通过多个层次，包括事务层、数据链路层和物理层。

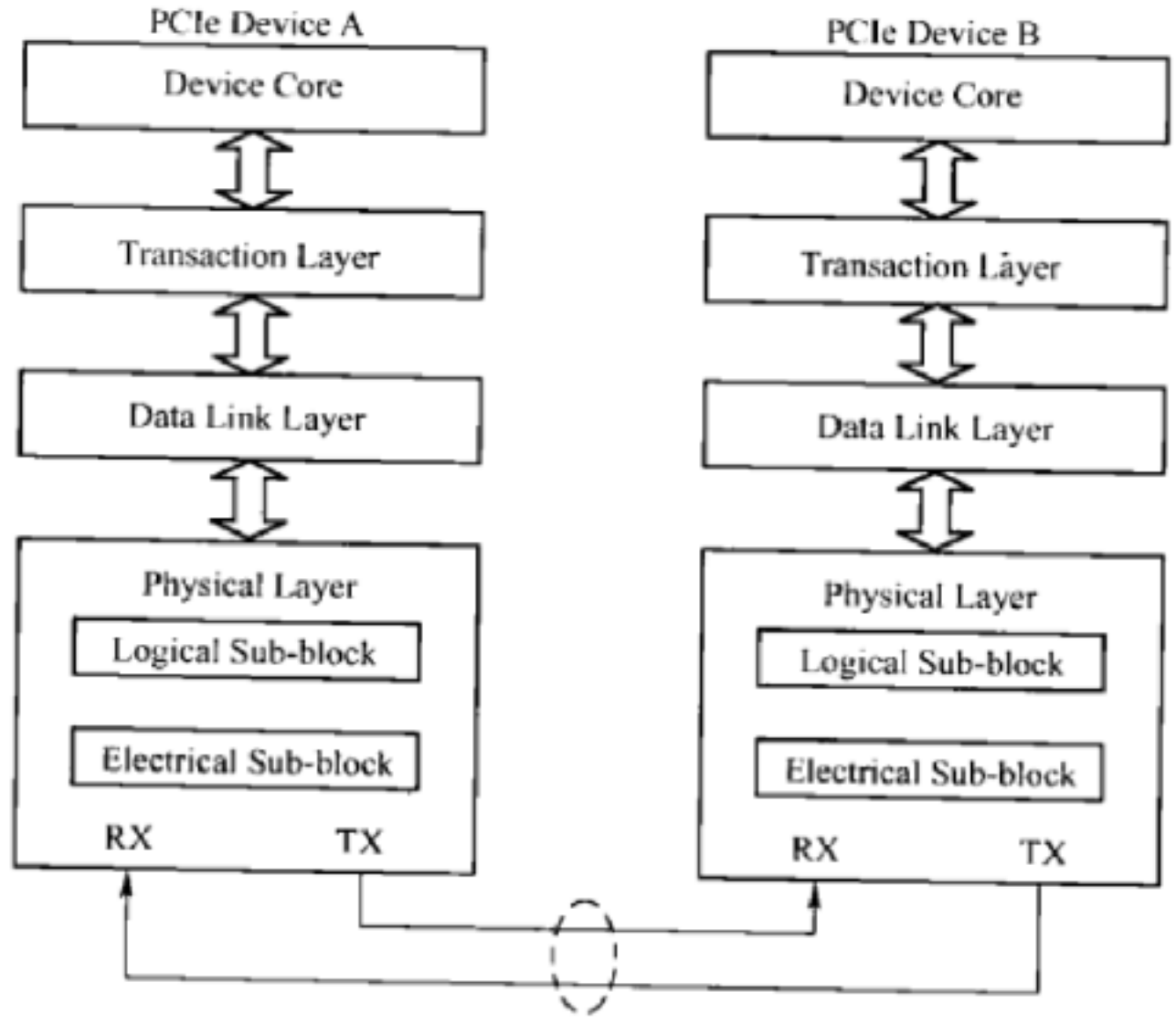


图 4-4　PCI Express 总线的层次组成结构

PCIe 总线的层次组成结构与网络中的层次结构有类似之处，但 PCIe 总线的各个层次都是用硬件逻辑实现的。在 PCIe 体系结构中，根据报文首先在设备的核心层（Device Core）中产生，然后再经过该设备的事务层（Transaction Layer）、数据链路层（Data Link Layer）和物理层（Physical Layer），最终发送出去。而接收端的数据也需要经过物理层、数据链路层和事务层，最终到达核心层。

1. 事务层

事务层定义了 PCIe 总线使用总线事务，其中多数总线事务与 PCI 总线兼容。这些总线事务可以通过 Switch 等设备传送到其他 PCIe 设备或者 RC 设备。RC 设备也可以使用这些总线事务访问 PCIe 设备。

事务层接收来自 PCIe 设备核心层的数据，并将其封装成 TLP（Transaction Layer Packet）后，发向数据链路层。此外事务层还可以从数据链路层中接收数据报文，然后转发至 PCIe 设备的核心层。

2. 数据链路层

数据链路层保证来自发送端事务层的报文可以可靠、完整地发送到接收端的数据链路层。来自事务层的报文在通过数据链路层时，被添加 Sequence Number 前缀和 CRC 后缀。数据链路层使用 ACK/NAK 协议保证报文的可靠传递。

PCIe 总线的数据链路层还定义了多种DLLP（Data Link Layer Packet），DLLP 产生于数据链

路层，终止与数据链路层。

1. 物理层

物理层是 PCIe 的最底层，将 PCIe 设备连接再一起。PCIe 总线的物理电气特性决定了PCIe 链路只能使用端到端的连接方式。PCIe 总线的物理层为 PCIe 设备间的数据通信提供传送介质，为数据提供可靠的物理环境。

1.2.3 PCIe 体系结构的组成结构

PCIe 总线作为处理器系统的局部总线，其作用于 PCI 总线类似，主要目的是为了连接处理器系统中的外部设备。在大多数处理器系统中，都使用RC Switch 和 PCIe-to-PCI 桥这些基本模块连接 PCIe 和 PCI 设备。在 PCIe 总线中，基于 PCIe 总线的设备，也成为 EP（Endpoint）。

基于 PCIe 总线的通用处理器系统如下图

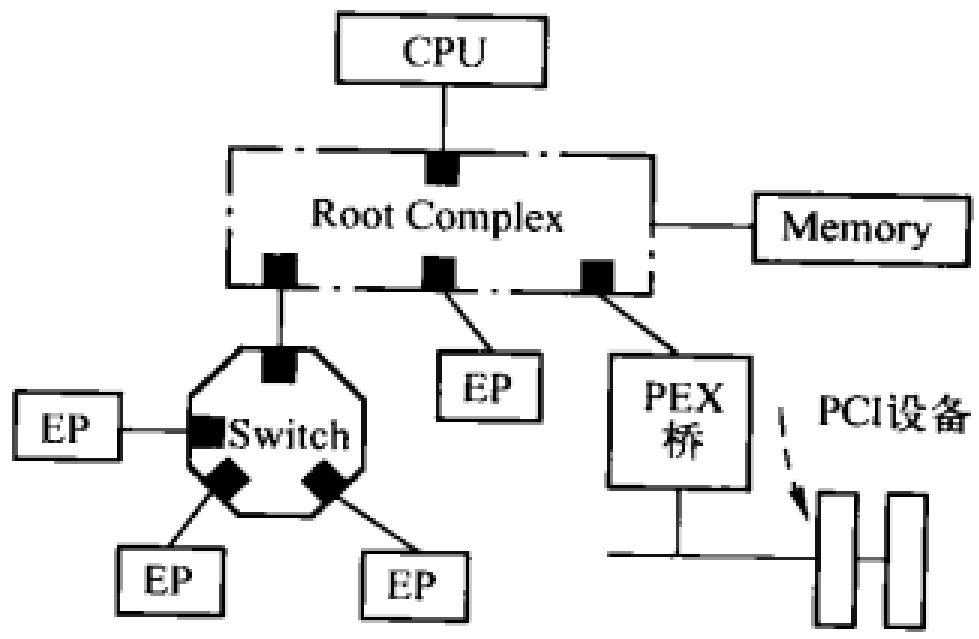


图 4-9　基于 PCIe 总线的通用处理器系统

图中所示的结构将 PCIe 总线端口、存储器控制器等一系列与外部设备有关的接口都集成在一起，并统称为 RC。RC 具有一个或者多个 PCIe 端口，可以连接各类 PCIe 设备。PCIe 设备包括（网卡、显卡等设备）、Switch 和 PCIe 桥。

PCIe 总线采取端到端的连接方式，每一个 PCIe 端口只能连接一个 EP，当然 PCIe 端口也可以连接 Switch 进行链路扩展。通过 Switch 扩展出的 PCIe 链路可以继续挂接 EP 或者其它 Switch。

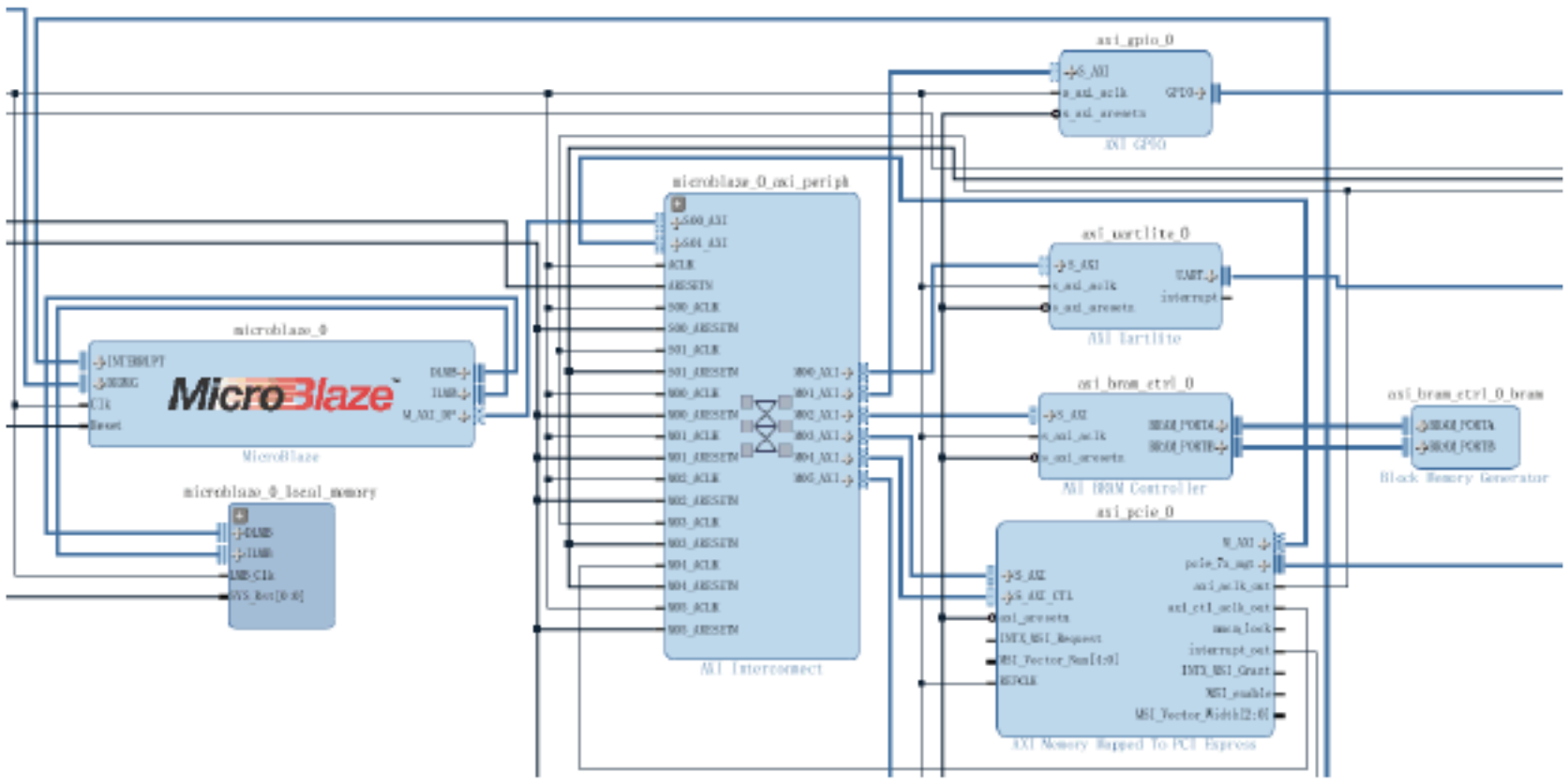
2. 基于 FPGA 的 PCIe 总线分析

2.1 硬件系统设计

本章将采用 xilinx 的 FPGA 芯片搭建一个 RC 端和 EP 端的硬件平台，用于仿真学习 PCIe 总线相关的知识包括DMA 操作、地址映射等，以及xilinx 的 AXI Memoy Mapped TO PCI Express、AXI DMA 等 IP 使用。

2.1.1 RC 端系统设计

RC 端硬件系统由MicroBlaze 处理器、RC（采用 xilinx 的 AXI Memony Mapped To PCI Express 配置成 RC 模式）、内存（采用内置 BRAM 块）、常用外设包括 GPIO、串口。



地址分配

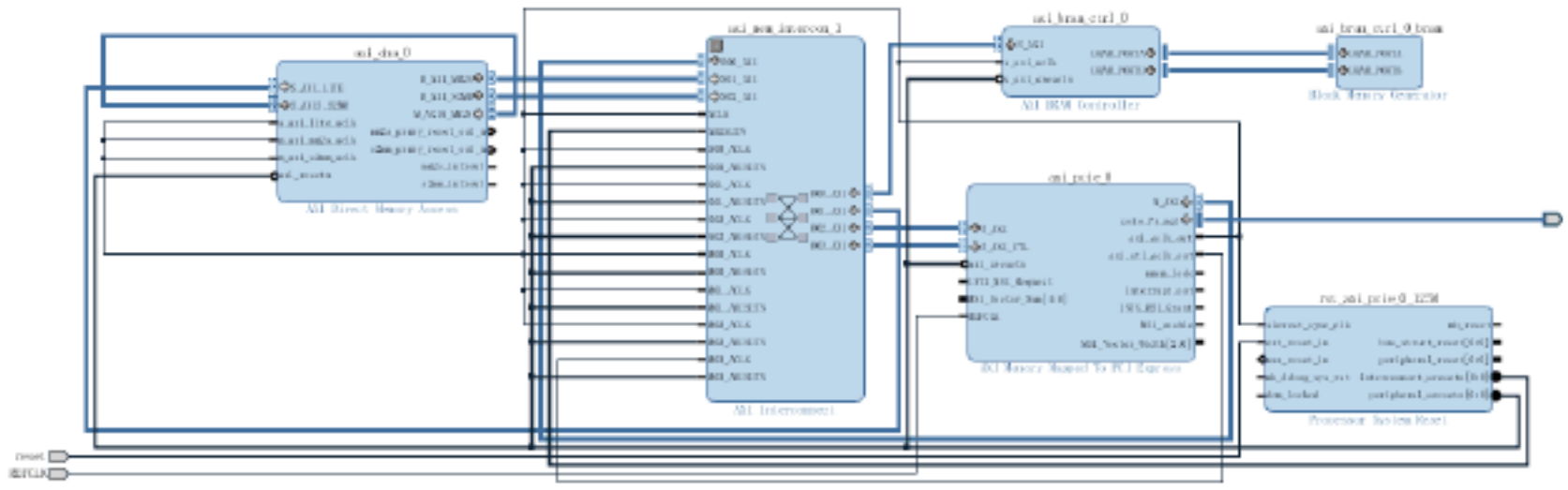
axi_pcie_0					
M_AXI (32 address bits : 4G)					
axi_bram_ctrl_0	S_AXI	Mem0	0x0600_0000	64K	0x0600_FFFF
axi_gpio_0	S_AXI	Reg	0x4000_0000	64K	0x4000_FFFF
axi_intc_0	s_axi	Reg	0x4120_0000	64K	0x4120_FFFF
axi_pcie_0	S_AXI	BAR0	0xC000_0000	64K	0xC000_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x5000_0000	256M	0x5FFF_FFFF
axi_uartlite_0	S_AXI	Reg	0x4060_0000	64K	0x4060_FFFF
microblaze_0					
Data (32 address bits : 4G)					
axi_bram_ctrl_0	S_AXI	Mem0	0x0600_0000	64K	0x0600_FFFF
axi_gpio_0	S_AXI	Reg	0x4000_0000	64K	0x4000_FFFF
axi_intc_0	s_axi	Reg	0x4120_0000	64K	0x4120_FFFF
axi_pcie_0	S_AXI	BAR0	0xC000_0000	64K	0xC000_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x5000_0000	256M	0x5FFF_FFFF
axi_uartlite_0	S_AXI	Reg	0x4060_0000	64K	0x4060_FFFF
microblaze_0_local_memory/dlmb_bram...	SLMB	Mem	0x0000_0000	128K	0x0001_FFFF
Instruction (32 address bits : 4G)					
microblaze_0_local_memory/ilmb_bram...	SLMB	Mem	0x0000_0000	128K	0x0001_FFFF

pcie 地址空间　　axi_pcie_bar0　　0xC000_0000~0xC000_FFFF

mem 地址空间　　axi_mem_0　　0x0600_0000~0x0600_FFFF

2.1.2 EP 端系统设计

EP 端硬件系统由MicroBlaze 处理器、EP（采用 xilinx 的 AXI Memony Mapped To PCI Express 配置成 EP 模式）、DMA 以及内置 BRAM 块。



地址分配

axi_dma_0					
Data_MM2S (32 address bits : 4G)					
axi_bram_ctrl_0	S_AXI	Mem0	0x0800_0000	32K	0x0800_7FFF
axi_pcie_0	S_AXI	BAR0	0x4000_0000	64K	0x4000_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x1000_0000	256M	0x1FFF_FFFF
Excluded Address Segments (1)					
axi_dma_0	S_AXI_LITE	Reg	0x2000_0000	64K	0x2000_FFFF
Data_S2MM (32 address bits : 4G)					
axi_bram_ctrl_0	S_AXI	Mem0	0x0800_0000	32K	0x0800_7FFF
axi_pcie_0	S_AXI	BAR0	0x4000_0000	64K	0x4000_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x1000_0000	256M	0x1FFF_FFFF
Excluded Address Segments (1)					
axi_dma_0	S_AXI_LITE	Reg	0x2000_0000	64K	0x2000_FFFF
axi_pcie_0					
M_AXI (32 address bits : 4G)					
axi_bram_ctrl_0	S_AXI	Mem0	0x0800_0000	32K	0x0800_7FFF
axi_dma_0	S_AXI_LITE	Reg	0x0800_8000	32K	0x0800_FFFF
axi_pcie_0	S_AXI	BAR0	0x4000_0000	64K	0x4000_FFFF
axi_pcie_0	S_AXI_CTL	CTL0	0x1000_0000	256M	0x1FFF_FFFF

pcie 地址空间　　axi_pcie_bar0　　0x4000_0000~0x4000_FFFF

mem 地址空间　　axi_mem_0　　0x0800_0000~0x0800_FFFF

2.2 读写数据分析

2.2.1 读写 MEM 数据

操作地址为 RC 端 BAR 空间存储器域地址+偏移量

例如，往 EP 端偏移量为 0x1000 的 MEM 写数据，RC 端写地址为 0xC000_000+0x1000

0xC000_000 为 RC 端 axi_pcie_bar0 基地址，即 EP 端 PCIe 域基地址映射到 RC 端存储器地址域的基地址。

将 RC 端存储器域地址 0xC000_1000，转换成 EP 端 AXI 地址 0x0800_1000，最终写入到 EP 端 MEM。