http://myfj.cnjournals.com myfj_sadri@163.com (021)20866796

DOI: 10.19416/j.cnki.1674-9804.2024.01.005

PCI 总线空间配置原理及具体实现

王 宁* 王珍珍 崔西宁

(中国航空工业集团公司西安航空计算技术研究所,西安 710076)

摘 要: PCI 总线属于处理器系统的局部总线范畴,作为系统总线的延伸,PCI 总线的主要目的是为了连接处理器与外部设备。自从 PCI 总线规范提出以后,迅速得到了广大厂商的认可与推广,进而在处理器体系结构中始终占据着重要的位置。从软件使用的角度来看,PCI 总线的空间配置属于一项重要内容。通过分析 PCI 总线空间配置的基本原理,并且结合 PPC 处理器架构下 PCI 设备空间配置的具体要求,详细介绍了软件使用层面的 PCI 配置原理及技术实现细节,并且给出了相关的示例代码。

关键词: PCI 总线;PCI 空间配置;PPC 处理器架构

中图分类号: TP332 **文献标识码**: A **OSID**:

0 引言

PowerPC (performance optimization with enhanced RISC-performance computing,简称 PPC)是一种精简指令集(RISC)架构的中央处理器(CPU),其基本的设计源自 IBM(国际商用机器公司)的 POWER (performance optimized with enhanced RISC)。PowerPC 芯片凭借其出色的性能和高度整合技术先进特性广泛应用于网络通信、工业控制、家用数字化、网络存储、军工以及电力系统控制等领域。

PowerPC 的 E500 是飞思卡尔(FreeScale)基于 Power Architecture 的 32 位微处理器核心。E500 系列核心有三个版本,即 E500v1,E500v2 的 E500mc。64 位版本的 E500mc 演变为 E5500 核心,并于 2010 年推出。E500 提供 32 位有效地址和 8 位、16 位和 32 位整数数据类型,它还为 SPE APU 和嵌入式向量浮点 APU 提供双元素 64 位数据类型,其中包括对由两个 32 位元素组成的操作数进行操作的指令。

PCI 总线是一种高速串行总线标准,可用于连接计算机系统中的各种设备,例如:图形卡、网络适配器、存储控制器等。PCIE 总线提供比传统的并行总线更高的数据传输速率和更低的传输延迟,因此在计算机系统中得到广泛的应用。PCI 使用中一项重要内容就是 CPU 对 PCI 设备空间的访问,本文主要介绍 CPU 对 PCI 设备空间的访问原理,以及在 PPC 处理器系统中的具体实现。

1 PCI 设备的空间配置

PCI 设备的空间配置是 PCI 协议中的重要组成部分,它定义了在 PCI 系统中各个设备所使用的不同地址空间范围。PCI 设备空间分为三个类型:配置空间、IO 空间和内存空间。PCI 设备的配置空间包含设备的 ID、制造商信息、中断信息以及 BAR 空间配置信息等。IO 空间用于 IO 操作,例如读写外部设备。内存空间用于主机和 PCIE 设备之间的数据交换。每个 PCI 设备都有一个唯一的 Bus 号、Dev 号以及 Func 号,用于标识其在 PCI 总线上的位置。通过这些编号系统就可以访问每个设备的配置空间。通过设备的配置空间,可以进一步配置 PCI 设备的 IO 空间和内存空间[1-2]。PCI 设备的配置空间所包含的信息如图 1 所示。

* 通信作者. E-mail: wwnn1_class@sina.com

引用格式: 王宁,王珍珍,崔西宁. PCI 总线空间配置原理及具体实现[J]. 民用飞机设计与研究,2024(1):28-33. WANG N, WANG Z Z, CUI X N. The principle and specific implementation of PCI bus space configuration[J]. Civil Aircraft Design and Research, 2024(1):28-33(in Chinese).

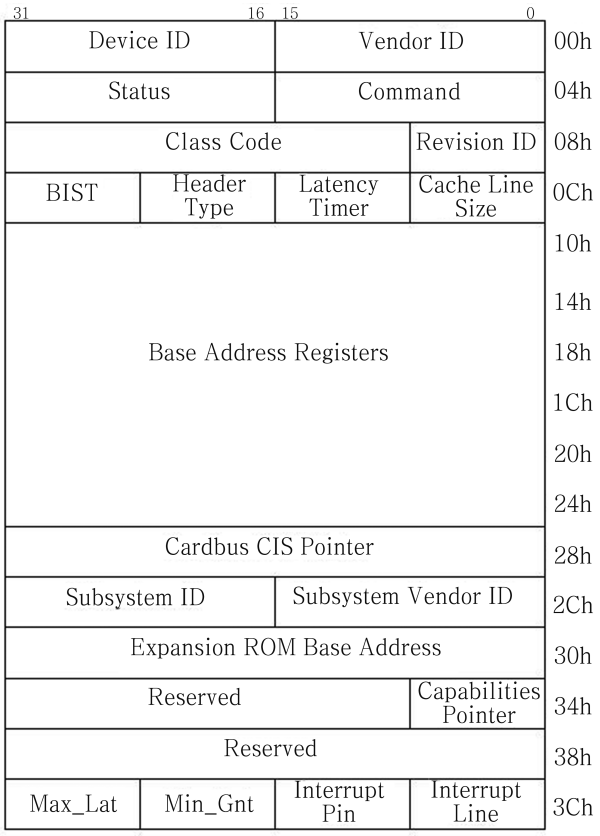

图 1　PCI 设备配置空间信息

PCI 设备的 IO 空间和内存空间都是通过图 1 中的 Base Address Registers(BAR 寄存器)进行配置。BAR 地址寄存器的第 0 位如果是 1,就表示该段地址空间为 IO 地址空间;如果是 0,就表示该段地址空间为内存空间。每一个 PCI 设备的配置空间中,都含有该设备在 PCI 地址空间内所使用的基地址,系统软件可以动态地配置该基地址,但是要确保各个 PCI 设备的物理地址互不冲突。当 PCI 设备的配置空间被系统软件初始化之后,该 PCI 设备在 PCI 总线树上将拥有自己的 PCI 总线地址空间,即 BAR 寄存器所描述的地址空间。在每个 PCI 设备的配置空间中,都有 6 个 BAR 寄存器。每一个 BAR 寄存器都与该 PCI 设备所使用的一段 PCI 地址空间相对应。

配置 PCI 设备 IO 空间和内存空间时,最容易混淆的两个概念就是 PCI 地址空间和 CPU 地址空间。以 32 位地址空间为例,PCI 地址空间和 CPU 地址空间的范围都是 0x00000000—0xFFFFFFFF。主机端只能发起/响应 CPU 地址空间的地址访问操作,PCI 设备也只能发起/响应对于 PCI 地址空间的访问请求,因此,在 PCI 地址空间和 CPU 地址空间之间就存在一层映射关系,如图 2 所示。

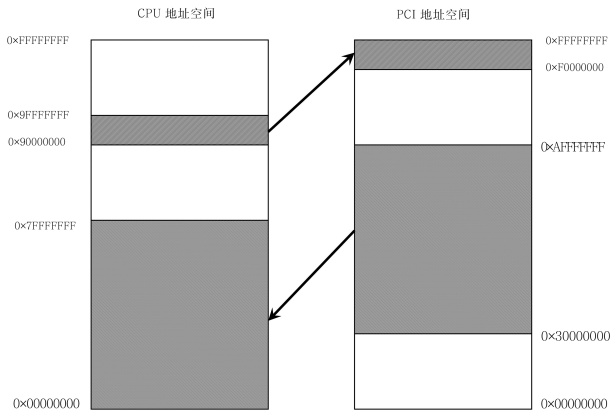

图 2　CPU 地址空间和 PCI 地址空间之间的映射

以图 2 的地址空间映射为例,当 CPU 需要访问 PCI 地址空间 0xF0000000—0xFFFFFFFF 这段地址时,根据图 2 中的映射关系,CPU 需要发出对 0x90000000—0x9FFFFFFF 这段地址空间中相应地址的访问请求。当 CPU 发出对 0xF0000008 地址的访问请求时,实际访问的则是 CPU 地址空间的 0xF0000008 地址,并不能访问到 PCI 地址空间的 0xF0000008 地址。只有当 CPU 发出对 0x90000008 地址的访问请求时,才能真正访问到 PCI 地址空间的 0xF0000008 地址。反之亦然,当 PCI 设备需要访问 CPU 地址空间 0x00000000—0x7FFFFFFF 这段地址时,PCI 设备实际发出的是对 PCI 地址空间 0x30000000—0xAFFFFFFF 这段地址空间的访问操作。

由于在 CPU 地址空间和 PCI 地址空间需要进行地址映射,因此就需要一个具有地址转换功能的角色,该角色就是 PCI 的 HOST 主桥。以 CPU 访问 PCI 设备为例,当 CPU 需要发起对某一个 PCI 设备的访问时,会发出对相应 CPU 地址空间的访问请求,然后由 HOST 主桥将该 CPU 地址空间转换为对应的 PCI 地址空间,然后再由对应的 PCI 设备对该访问请求进行响应。因此,任何一个 PCI 设备只有在 CPU 地址空间具有相应的映像地址时,才能被 CPU 所访问[3-4]。

由以上分析可知,HOST 主桥负责隔离 CPU 地址空间和 PCI 地址空间,并完成 CPU 与 PCI 设备之间的地址映射和数据交换。CPU 与 PCI 设备之间的数据交换则主要由 CPU 访问 PCI 设备的地址

空间”和 PCI 设备通过 DMA 机制访问主机端的 RAM”两部分组成。HOST 主桥的具体实现,在不同的处理器系统会有较大差异,下面以集成了 E500 核的 PPC 处理器系统为例,具体介绍 PCI 的配置过程。

2　PPC 处理器系统中 PCI 的配置

在集成了 E500 核的 PPC 处理器系统中,对于 PCI 的配置全部通过寄存器完成。由于集成了 E500 核的 PPC 处理器系统没有 IO 端口的概念,所以,对寄存器都是通过内存映射的方式进行访问,即所有寄存器在 CPU 地址空间都有自己的地址。因此,为了完成对 PCI 的配置,首先必须找到对应的寄存器地址值。在集成了 E500 核的 PPC 处理器系统中,PCI 配置相关的寄存器隶属于系统配置/控制/状态寄存器(CCSR)空间。CCSR 空间的基地址则通过 CCSRBAR 寄存器进行配置。与 PCI 配置相关的寄存器则属于 CCSR 空间的 General Utilities 部分[5]。在集成了 E500 核的 PPC 处理器系统中,典型的内存空间及 CCSR 空间划分如图 3 所示。

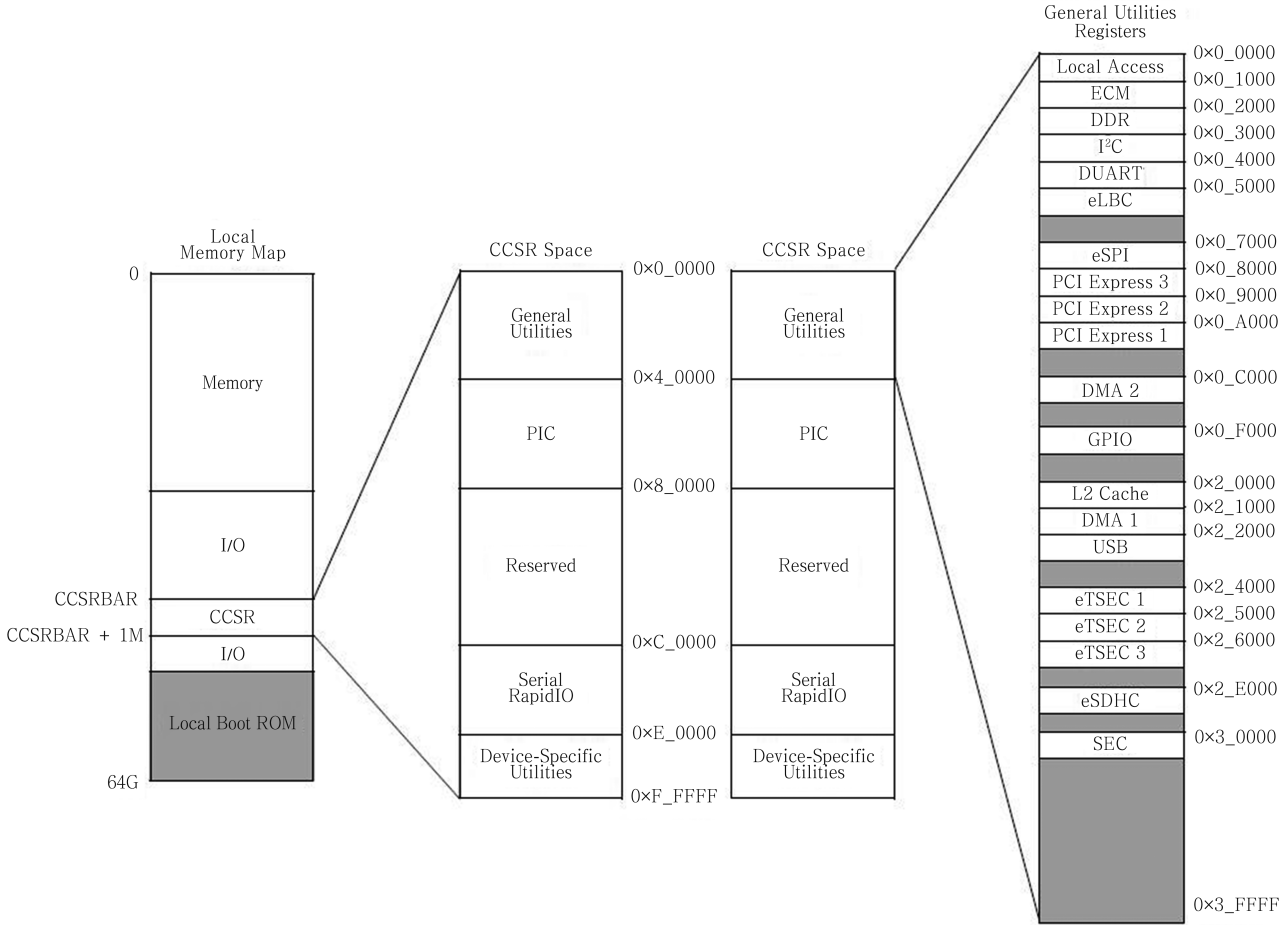

图 3　CCSR 空间映射示意图

由图 3 可知,在 CCSR 空间的 General Utilities 中,为 PCI 配置分配了三段独立的寄存器空间,起始偏移地址依次为:0x8000、0x9000、0xA000。这三段寄存器地址空间分别对应三个独立的 PCI 总线控制器,在这三段地址空间内,寄存器的分布情况则完全相同,具体结构如图 4 所示。

图 4 中与 CPU 地址空间和 PCI 地址空间地址映射相关的配置寄存器隶属于 ATMU 寄存器段,该段寄存器地址偏移从 0xC00 开始,共包含 5 组 outbound 寄存器和 3 组 inbound 寄存器。其中,outbound 寄存器组负责 CPU 地址空间到 PCI 地址空间的映射配置;inbound 寄存器组则负责 PCI 地址空间到 CPU 地址空间的映射配置。以图 2 为例,只有在 outbound 寄存器组中配置了相关的映射关系之后,CPU 发出的对 0x90000000—0x9FFFFFFF 地址段的访问,才能够被成功转换成对 PCI 地址

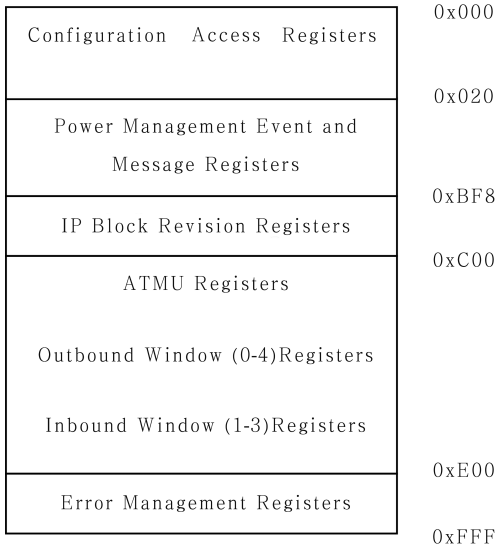

图 4　PCI 配置空间寄存器

空间 0xF0000000—0xFFFFFFFF 地址段的访问;同样地,也只有在 inbound 寄存器组中配置了相关的映射关系之后,PCI 设备发出的对0x30000000—0xAFFFFFFF 地址段的访问,才能够被成功转换成对 CPU 地址空间 0x00000000—0x7FFFFFFF 地址段的访问。

在集成了 E500 核的 PPC 处理器系统中访问 PCI 地址空间,除了配置上述的映射关系以外,还需要在 Local Access Window(LAW)寄存器中,对映射关系中的 CPU 地址空间进行访问配置,将 CPU 对于该段地址空间的访问处理指派给 PCI 总线控制器[6]。实际上,在集成了 E500 核的 PPC 处理器系统中,所有的地址空间的访问都需要在 LAW 寄存器中进行配置,以决定该段地址空间的访问目标。由图 3 可知,LAW 寄存器组位于 CCSR 寄存器空间的起始位置(偏移为 0),大小为 0x1000。

LAW 寄存器组共包含 12 组 LAWBAR 和 LAWAR 寄存器,每组[LAWBAR,LAWAR]寄存器配套使用。其中,LAWBAR 寄存器负责配置地址空间的起始地址,LAWAR 寄存器负责配置地址空间的大小、目标以及使能位,如图 5 所示。

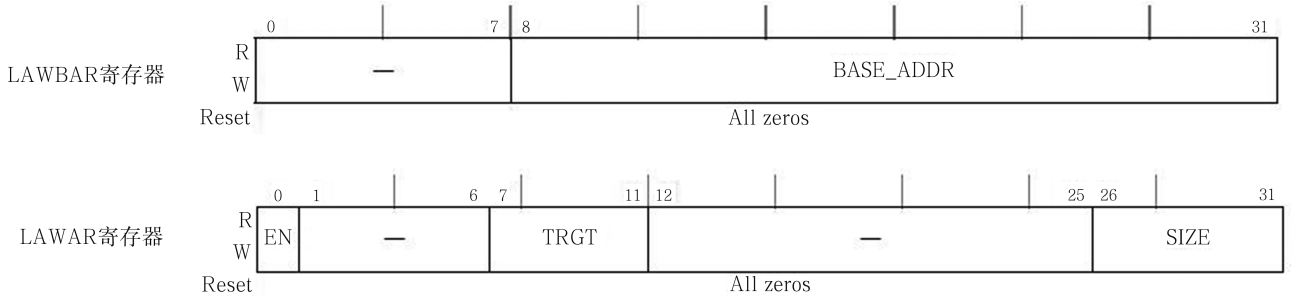

图 5　LAW 寄存器

图 5 中,LAWAR 寄存器的 TRGT 域负责指定该段地址空间所访问的实际目标。与三路 PCI 总线控制器对应的 TRGT 值分别为:0、1、2。

根据以上介绍,在集成了 E500 核的 PPC 处理器系统中对 PCI 进行配置的过程可以归纳为以下几个主要步骤。

第一步:配置 CCSRBAR 寄存器,为 CCSR 空间的寄存器分配起始地址,假设将 CCSRBAR 设置为 0x70000000。

第二步:通过 CCSRBAR 寄存器以及 LAW 配置寄存器的偏移,获取 LAW 配置寄存器的起始地址为 0x70000000。

第三步:选取一组未使用的[LAWBAR,LAWAR]寄存器,为访问 PCI 设备的 CPU 地址空间配置本地访问窗口。

第四步:通过 CCSRBAR 寄存器以及 PCI 配置寄存器的偏移,获取 PCI 配置空间寄存器的起始地址。以第一路 PCI 控制器(偏移 0x8000)为例,其配置寄存器的起始地址为 0x70008000。

第五步:通过 PCI 控制器的配置寄存器起始地址,计算 ATMU 配置寄存器的起始地址。以第一路 PCI 控制器为例,其 ATMU 配置寄存器的起始地址为 0x70008C00。

第六步:通过访问 ATMU 配置寄存器的内存地址,对 ATMU 寄存器(包括 outbound 寄存器组和 inbound 寄存器组)进行地址空间映射相关的配置。

第七步:在设备的 BAR 空间寄存器中,配置该设备需要使用的 BAR 空间。

通过以上七个步骤,就可以对 CPU 地址空间和 PCI 地址空间的映射关系进行配置。配置完成之后,主机 CPU 和 PCI 设备之间就可以通过配置好的映射关系进行数据交换。

PPC 处理器系统中 PCI 配置的示例代码如图 6 所示。

```
/* 配置 LAW 空间 */
out  long(M86xx_LAWBAR2(M86xx_CCSBAR),
            (PCIE1_OW_CPU_ADDR) >> LAWBAR_ADRS_SHIFT);
out  long(M86xx_LAWAR2(M86xx_CCSBAR),
          LAWAR_ENABLE | LAWAR_TGTIF_PCIEX1 | LAWAR_SIZE_256MB);
.....
/* 配置 ATMU */
pcie  = (struct ccsr_pcie  *) M86xx_PCIE1_CCSR;
ob_addr_pci  =PCIE1_OW_CPU_ADDR;
ob_addr_pci  =PCIE1_OW_CPU_ADDR;
ib_addr_pci  =PCIE1_IW_CPU_ADDR;
ib_addr_pci  =PCIE1_IW_PCI_ADDR;
/* 配置 outbound MEM window */
out_long((UINT32) &pcie→pow[1].potar,((ob_addr_pci)  >> 12)  & 0x000ffff) ;
out_long((UINT32) &pcie→pow[1].potar,0) ;
out_long((UINT32) &pcie→pow[1].powbar,((ob_addr_cpu)  >> 12)  & 0x000ffff) ;
out_long((UINT32) &pcie→pow[1].powar, 0x80044000  | PCIE_ATTR_WS_256M) ;
/* 配置 inbound MEM window */
out_long((UINT32) &pcie→piw[2].pitar,(ib_addr_cpu)  >> 12)  & 0x000ffff) ;
out_long((UINT32) &pcie→piw[2].piwbar,(ib_addr_pci)  >> 12)  & 0x000ffff) ;
out_long((UINT32) &pcie→piw[w].piwbar,0) ;
out_long((UINT32) &pcie→piw[2].piwar,A0F55000 | PCIE_ATTR_WS_1G) ;
.....
/* 通过设备的 VENDOR_ID和 DEVICE_ID获取其 bus、dev、func号*/
/* 从而在 PCI 配置空间中找到设备对应的配置区域*/
if(pciFindDevice(VENDOR_ID,DEVICE_ID,0, &bus, &dev, &fuac)== OK)
{
    /* 该设备需要配置两路BAR，将预分配的两个空间分别写入BAR0与BAR1寄存器*/
    pciConfigOutLong  (bus,device,func,PCI_CFG_BAR0_ REG,PCI_BAR0);
    pciConfigOutLong  (bus,device,func,PCI_CFG_BAR1_ REG,PCI_BAR1);
    /* 形成地址空间使能命令，并写入CMD 寄存器*/
    pci_cr  =  PCI_CMD_MEM_ENABLE | PCI_CMD_MASTER_ENABLE;
    pciConfigOutWord  (bus,device,func,PCI_CFG_CMD_ REG,pci_cr);
}
```

图 6　LAW 寄存器

3　结论

PCI 总线作为目前使用最为广泛的局部总线,在各大厂家处理器的体系结构中占据着重要的地位。对 PCI 设备总线空间的配置是软件使用 PCI 的重中之重。本文通过对 PCI 空间配置原理的详细分析,以及结合具体的 PPC 处理器架构实现,给出了详细的软件配置示例代码,对于理解其它处理器架构下 PCI 设备空间的配置有一定借鉴意义。

参考文献:

[1] GUO J Y, HE Z W. Design of general data-processing board based on embedded PowerPC platform[J]. Advanced Materials Research,2011,403-408:1981-1984.

[2] LIU W, LIU Y F, QIAO L Y. Development of dual-channel high-speed data acquisition card based on PCI bus[C]//2013 IEEE AUTOTESTCON, September 16-19, 2013, Schaumburg, IL, USA. [S. l.]: IEEE, 2013.

[3] 刘红甫,樊双丽,曲道奎. 基于 PowerPC 的单板计算机的设计[J]. 微计算机信息, 2008, 24(29):60-62.

[4] WILSON T. PowerPC-PCI bridging in embedded applications[J]. Electronic product design, 1998,19(6):25-26,28.

[5] HOU J B. Study on PCI bus and ISA bus conversion design[J]. Computer Applications and Software, 2013. DOI:10.4156/jdcta.vol7.issue4.55.

[6] 刘博,颜丰琳,程小芩. PowerPC8270 数据处理模块的异步总线接口设计[J]. 长江信息通信,2019(6):75-76.

[7] 李智慧,张双,刘帜琦,等. PowerPC 多核处理模块性能测试方法研究[J]. 长江信息通信, 2023, 36(4):121-123.

[8] 王志强,何洋,黄子露. PowerPC755 模块常见故障分类及排除[J]. 电子世界, 2021(8):178-179.

[9] 范坤. 航空电子设备的一种多核高速处理平台设计[J]. 单片机与嵌入式系统应用, 2023, 23(5):73-77.

[10] 边庆,白晨,田征戈. 基于 P2010 处理器的嵌入式系统硬件平台设计[J]. 信息技术与信息化, 2022(4):163-166.

作者简介

王　宁　男,本科,高级工程师。主要研究方向:机载计算机软件。E-mail: wwnn1_class@sina.com

王珍珍　女,硕士,助理工程师。主要研究方向:机载计算机软件。E-mail: 1072593676@qq.com

崔西宁　男,博士,研究员。主要研究方向:机载计算机软件。E-mail: cxning@avic.com

The principle and specific implementation of PCI bus space configuration

WANG Ning*　WANG Zhenzhen　CUI Xining

(Xi'an Aeronautics Computing Technique Research Institute, AVIC, Xi'an 710076, China)

Abstract: Peripheral component interconnect(PCI) bus belongs to the local bus category of the processor system, as an extension of the system bus, the main purpose of PCI bus is to connect the processor and external devices. Since the PCI bus specification was proposed, it has been rapidly recognized and promoted by the majority of manufacturers, and has always occupied an important position in the processor architecture. From the perspective of software use, the spatial configuration of PCI bus is an important content. By analyzing the basic principle of PCI bus space configuration, and combining with the specific requirements of PCI device space configuration under PPC processor architecture, this paper introduces the PCI configuration principle and technical implementation details at the software use level in detail, and gives the relevant example code.

Keywords: PCI bus; PCI space configuration; PPC processor architecture

* Corresponding author. E-mail: wwnn1_class@sina.com