Windows 2000下PLX9050 PCI接口设备驱动程序开发

别其璋

(重庆工商大学计算机科学与信息工程学院 重庆 400033)

【摘要】在Windows主机系统中使用PLX 9050芯片实现了PC主机与Motorola PowerPC 860微处理系统之间的外设部件互联(PCI)接口连接，提供了解决微处理器系统与计算机系统之间通过PCI接口互联的一种方法。介绍了在Windows 2000系统下开发PCI设备驱动程序的基本方法和步骤。这种方法还可推广用于Windows 2000下开发压缩的PCI(CompactPCI)接口设备的驱动程序，从而为使用PC机开发工业控制类应用提供一种便捷的编程方法。

关 键 词 外设部件互联接口； 设备驱动程序； 双端口随机存储器； 直接内存存取

中图分类号 TP311.52 文献标识码 A

Development of PLX 9050 PCI Device Driver on Windows 2000

BIE Qi-zhang

(College of Computer and Information Engineering, Chongqing Technology and Business Univ. Chongqing 400033)

Abstract PCI interface connection between Motorola PowerPC 860 MPU system and PC host is complemented using PLX 9050 chip based on Windows 2000.A solution of interconnection between MPU and computer system by means of PCI interface is given. Our discussion centre on the basic method and procedure of developing PCI interface device driver on Windows 2000 system in this paper. This approach can further be applied to develop CompactPCI interface device driver and provides us a convenient and fast way to program for PC in industrial control applications.

Key words peripheral component interconnect interface; device driver; dual port random access memory; direct memory access

随着计算机工业的发展，越来越多的PC架构的计算机系统应用于通信，自动控制等领域，设计这类工业控制系统时，常常需要实现PC主机系统和微处理器系统之间的通信控制。最初常常采用串口或者工业标准结构 (Industry Standard Architecture，ISA)接口实现两者的通信。这两种方式由于速率较低，误码率高，不可能提供高速和可靠的连接。随着外设部件互联(Peripheral Component Interconnect，PCI)接口的出现和普及，越来越多的工业控制系统采用PCI接口与微处理器系统相连，因此，开发PCI设备驱动程序，成为许多厂商和研究机构探讨的课题。

1 系统结构

以一个非结构化补充业务数据(Unstructured Supplementary Service Data，USSD)设备开发为例，介绍在Windows 2000下开发PCI接口驱动程序的方法。该设备以计算机为核心，链路接口板收发7号信令网信息。系统结构如图1所示。

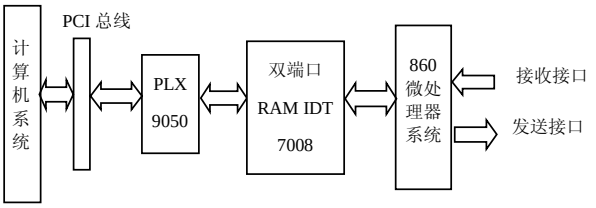

图1 系统结构

收稿日期：2004 – 07 – 12
作者简介：别其璋(1945 –)，男，副教授，主要从事电子与信息技术应用方面的研究.

1.1 PowerPC 860微处理器系统

系统采用Motorola公司的PowerPC 860微处理器，控制DS21554芯片(E1单通道收发器)来连接7号信令网链路，实现信令接口，同时控制MITEL MT90823AB(数据交叉芯片)实现E1接口中链路的时隙控制。860内部的通信处理模块接入信令链路数据，PowerPC处理器完成7号信令链路的消息传输部分(Message Transfer Part，MTP)的第二层逻辑控制。MTP第三层的消息，如信令网管理(Signaling Network Management，SNM)，电话用户部分(Telephone User Part，TUP)等消息，则通过双端口RAM IDT 7008发送到PLX9050中，PC主机通过PCI接口从PLX9050中读取这些消息。

1.2 IDT 7008双端口RAM芯片

IDT公司的IDT 7008双端口RAM芯片提供对一段RAM空间的双端口控制，两侧都可独立地读写RAM中的任何位置，同时提供同步保护机制，实现860微处理器和PC主机的信息交互。

1.3 PLX 9050接口卡

PLX Technology出品的PLX 9050芯片是一个高性能的PCI接口芯片，用于连接低速的微处理器本地总线和主机系统的高速PCI总线。在本应用中，PLX 9050卡一方面与PC主机系统的PCI总线连接，另一方面通过16位本地总线连接到双端口RAM IDT 7008，控制双端口RAM，与860微处理器相连。

1.4 计算机系统

采用普通PC计算机或者PC服务器，操作系统采用Windows 2000或Windows XP。

2 PLX 9050卡的分析和程序处理

2.1 PLX 9050特性及内部结构

PLX 9050芯片是连接低速本地总线和PCI总线的高性能的PCI接口芯片，提供6个可编程寄存器空间。PC主机通过编程控制这6个寄存器来完成该接口卡的工作模式配置和初始化过程。以下结合PCI设备驱动程序特点，作进一步分析。

2.2 PCI设备驱动程序特点

PLX 9050卡采用直接内存存取((Direct Memory Access，DMA)工作方式，采用映射主机物理内存来实现DMA工作。因此，要完成对PLX 9050卡的设备驱动程序设计，在计算机系统中需完成如下工作：

1) 完成对PLX 9050卡物理设备的标识；
2) 存取和修改PLX 9050卡的PCI配置空间；
3) 获取并映射PLX 9050卡的配置地址寄存器，实现对配置地址寄存器的读写；
4) 申请高达1 M字节连续的物理地址来完成DMA操作。

2.3 Windows 2000系统设备驱动程序设计

2.3.1 Windows 2000系统设备驱动程序特点

Windows 2000设备驱动采用WDM (Win32 Driver Mode，32位接口驱动程序模型)，由I/O管理器给驱动程序发送各种输入输出请求包 (IO Request Packet，IRP)，管理驱动程序的行为，通过一个叫做硬件设备抽象层 (Hardware Abstraction Layer, HAL)来访问底层设备[1]。

WDM驱动程序主体是一个入口函数DriverEntry，完成驱动程序对象的初始化，把各种函数指针填入驱动程序对象，这些指针对操作系统指明了驱动程序中各种子例程的位置。它包括以下指针成员：

1) DriverStartIo定义驱动程序使用的标准串行IRP，指向驱动程序的StartIo例程；
2) DriverExtension->AddDevice指向驱动程序的AddDevice函数，用于创建一个设备对象；
3) MajorFunction是一个指针数组，它主要处理各种用户的I/O请求IRP。

Windows 2000的虚拟地址和物理地址的对应关系如图2所示[2]：

在设备驱动程序中获取PCI设备寄存器的地址以及申请物理内存都是根据上述描述实现的。

Windows 2000设备驱动程序由设备驱动程序.sys文件和设备安装程序.inf文件组成，.inf文件设置设备的信息和安装信息，.sys文件为实际编译的设备驱动程序文件。

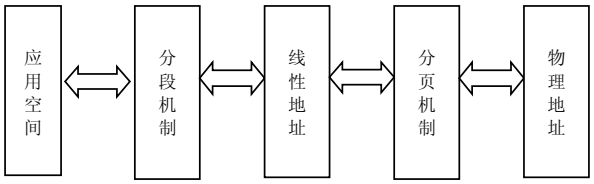

图2　Windows 2000的虚拟地址和物理地址的对应关系

2.3.2　Windows 2000系统中PCI设备驱动程序开发过程

1) 首先找到该PCI卡在Windows 2000系统的设备中的位置，使用regedit命令打开注册表，在“HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Enum\PCI”设备树中的PCI设备枚举下可发现PLX 9050卡的信息为“VEN_10B5&DEV_9050&SUBSYS_00000000&REV_01”，在设备信息中的Location Informatoin 字段中记录有PCI卡的总线信息“PCI总线1，设备15，功能0”，同时记录DeviceDesc字段[3]。

2) 编辑该PCI卡设备驱动程序的配置文件pci9050.inf，在[Manufacturer]段中指明设备信息：

[MfgDeviceSection]

%Fdo.DeviceDesc %＝DDK9050Install, PCI\VEN_10B5&DEV_9050&SUBSYS_905010B5&REV_01

然后完成设备其他操作和信息的配置。

3) 编辑PCI卡设备驱动程序文件，本例为 pci9050.c，编译后文件为 pci9050.sys， 把该文件和安装文件pci9050.inf 拷贝到任意目录。

4) 在Windows的设备管理器中查找名字为“网络适配器”的带“?”号的设备，选择“更新驱动程序”，找到pci9050.inf，即可完成设备的安装。

2.3.3　Windows 2000下编写PLX 9050卡设备驱动程序

限于篇幅，本文提供一个编制PLX 9050卡源文件pci9050.c的编程概要。

2.3.3.1　构建Windows 2000环境下设备驱动程序框架

在设备驱动程序的入口函数DriverEntry中定义各种子例程：

建立分派子例程的函数入口描述

DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL] ＝ Nt9050Dispatch；

…

Nt9050Dispatch( )为驱动程序的主过程，主要处理操作系统和应用程序发起的IRP请求，完成PCI配置空间存取，物理内存申请，设备运行信息配置等操作。下面对该过程进行详细介绍。

2.3.3.2　实现PCI配置空间存取和物理内存申请

要完成对PLX 9050卡的设备驱动，在计算机系统中需存取和修改它的PCI配置空间，获取并映射它的配置地址寄存器地址，申请高达1 M字节连续的物理地址来完成DMA操作。具体过程如下：

1) 存取和修改PCI配置空间

通过读注册表可获得PCI设备的总线号、时隙号，通过DeviceIoControl( )函数传递这些信息到驱动程序，驱动程序中设备驱动程序开发工具包 (Driver Development Kit，DDK)提供HalGetBusData( )函数获取PCI设备的配置空间，然后可调用HalSetBusData( )函数实现配置存取和修改。

2) 获取并映射配置地址寄存器地址

PLX 9050有6个配置地址寄存器，使用第一个(序号为0)地址寄存器来存放内存映射模式的配置信息，因此需映射该地址寄存器的基地址到应用程序，供应用程序使用。

首先，映射地址寄存器的基地址到线性空间

physicaladdress.LowPart＝DeviceExtension->PciConfigSpace.u.type0.BaseAddresses[0]；

physicaladdress.HighPart＝0；

DeviceExtension->pBaseVirAddr＝MmMapIoSpace (physicaladdress，size，MmNonCached)；

对指定内核虚拟地址申请MDL(Memory Describer List，内存描述表)

DeviceExtension->pBaseMdl＝IoAllocateMdl (DeviceExtension->pBaseVirAddr，size，

FALSE，FALSE，NULL)；

把该MDL指定的内存锁定在非分页空间

MmBuildMdlForNonPagedPool (DeviceExtension-＞pBaseMdl)；

将锁定的内存页映射到应用程序空间

pa-＞UserSpaceAddress＝MmMapLockedPages(pa-＞TheMdl，UserMode)；

这样，应用程序可通过访问pa-＞UserSpaceAddress地址来直接访问PCI卡的设备配置空间。

3) 申请连续的物理地址来完成DMA操作

为DMA操作申请逻辑连续的地址空间

pa-＞VirtualAddress＝DeviceExtension-＞Adapter-＞DmaOperations-＞AllocateCommonBuffer(

DeviceExtension-＞Adapter， pa-＞size，&pa-＞LogicalAddress，MmNonCached)；

申请的物理地址为pa-＞LogicalAddress的LowPart部分

pa-＞PhysicalAddress＝pa-＞LogicalAddress.LowPart；

对指定内核虚拟地址申请MDL

pa-＞TheMdl＝IoAllocateMdl(pa-＞VirtualAddress，pa-＞size，FALSE，FALSE，NULL)；

把该MDL指定的内存锁定在非分页空间

MmBuildMdlForNonPagedPool (pa-＞TheMdl)；

将锁定的内存页映射到应用程序空间

pa-＞UserSpaceAddress＝MmMapLockedPages(pa-＞TheMdl，UserMode)；

这样，应用程序可通过访问pa-＞UserSpaceAddress地址来直接访问申请的DMA内存块。

2.3.3.3 完成PLX 9050卡自身寄存器配置

根据PLX 9050卡的工作模式要求进行寄存器的配置工作，不再赘述。

2.3.3.4 建立驱动程序和用户程序间调用接口

设备驱动程序安装后，可在应用程序中进行调用。

打开设备驱动程序

CreateFile(functionClassDeviceData-＞DevicePath，GENERIC_READ | GENERIC_WRITE，0，

NULL，OPEN_EXISTING，0，NULL)；

通过DeviceIoControl()函数对该设备进行IO操作[3]

DeviceIoControl (hDevice，IOCTL_PCICFG_GETCFG，&in，sizeof (in)，&ConfigData，

sizeof (ConfigData)，&nRet，NULL)；

通过对PCI设备配置地址寄存器基地址的映射，应用程序可自由地配置PCI设备各种寄存器，完成设备的配置初始化过程，通过对PCI设备映射到用户空间的DMA内存的读写操作，完成PCI接口数据的采集和发送。

3 结 束 语

通过上述过程，在Windows 2000系统中完成了PLX 9050卡的设备驱动程序设计，实现了微处理器系统和PC主机系统之间的PCI接口连接，揭示了Windows 2000设备驱动程序的特点和内存管理的体系，提供了使用Windows 2000 DDK工具开发PCI接口设备驱动程序的一般步骤和技术要点。该开发过程还可以推广到在Windows 2000下开发CompactPCI(紧凑的外设部件互联)接口设备的驱动程序，从而为在Windows 2000系统下开发工业控制类应用提供一种便捷的方法。

参 考 文 献

[1] 郭益昆. VC++. NET开发驱动程序详解[M]. 北京：希望电子出版社, 2000

[2] Cant C. Windows WDM设备驱动程序开发指南[M]. 北京：机械工业出版社, 2000

[3] 李 进. Windows 9X/NT/2000注册表详解[M]. 北京：科学出版社, 2001

编 辑 熊思亮