

高速数据采集卡 WDM 驱动程序的开发

李正平¹, 徐超¹, 陈丽娟², 谭守标¹

(1. 安徽大学 电子科学与技术学院, 安徽 合肥 230039;

2. 合肥工业大学, 安徽 合肥 230009)

摘要: WDM 是 Microsoft Windows 2000/XP 操作系统的标准驱动程序模型。文中在简要介绍一款基于 PCI 总线高速数据采集卡硬件设计的基础上, 详细阐述了基于 WDM 模型的驱动程序设计。并结合项目的实际情况, 对设计中一些策略的考虑和编程技巧做了讨论, 给出了利用 DDK 开发 WDM 驱动程序的实例。

关键词: PCI 总线; WDM 驱动程序; 共享内存

中图分类号: TP316.7

文献标识码: A

文章编号: 1673-629X(2006)09-0123-03

Development of WDM Driver for High-Speed Data Acquisition Card

LI Zheng-ping¹, XU Chao¹, CHEN Li-juan², TAN Shou-biao¹

(1. School of Electronic Science & Technology, Anhui University, Hefei 230039, China;

2. Hefei University of Technology, Hefei 230009, China)

Abstract: WDM is the standard driver model in Microsoft Windows 2000/XP operating system. Briefly introduces the hardware design of one PCI bus high-speed data acquisition card, addresses the design of WDM driver in detailed. According to the requirement of project, discuss the strategy and programming tricks adopted in the device driver. It provides a good instance of WDM device driver development using DDK.

Key words: PCI bus; WDM driver; share memory

0 引言

PCI 总线即外围部件互连总线, 是一种高性能的 32/64 位地址数据复用总线, 已经成为个人计算机的标准总线^[1]。目前业界已经推出许多性能优异、技术成熟、价格便宜的 PCI 接口芯片, 其中 PLX 公司的 PCI9054 就是其中一款。Windows 2000/XP 是当前流行的操作系统, 具有优异的性能和良好的用户界面。基于 PCI 总线的数据采集卡要在 Windows 2000/XP 下正常工作, 必须开发设备驱动程序, 应用程序通过调用驱动程序提供的接口实现对采集卡的操作。

Windows 2000/XP 支持统一的 WDM 驱动程序模型。WDM 驱动程序支持即插即用, 支持电源管理和 WMI (Windows Management Instrumentation)。开发 WDM 驱动程序基本工具是 Microsoft 公司提供的 DDK, 相比于第三方公司提供的快速开发工具, 利用 DDK 开发的驱动程序具有代码简洁、结构清晰、效率高的优点。文中开发的一款基于 PCI9054^[2] 高速数据采集卡, 由于其应用场合对效率要求非常高, 因此采用 DDK 开发该采集卡的驱动程序。

收稿日期: 2005-12-12

作者简介: 李正平 (1979-), 男, 安徽宣城人, 博士, 副教授, 研究领域为数据采集与处理、操作系统内核、核技术应用以及嵌入式系统。

1 高速数据采集卡硬件设计

文中开发的高速数据采集卡是用来处理摄像头采集的图像数据, 一块板卡可以同时处理两路摄像头图像数据。图 1 为采集卡的硬件结构框图。其 PCI 控制逻辑由 PCI9054 芯片提供, 其配置信息由一块 EEPROM 提供。摄像头采集的信号首先经过高速运算放大器放大, 再经 AD 转换成数字信号传入板卡上的 RAM 存储器中, 然后由 PCI9054 启动 DMA 传输, 将数据从板卡上的 RAM 通过 PCI 总线传送到 PC 机内存中。板卡上 RAM 由一块 FPGA 实现, 该芯片同时还产生各种控制信号, 如控制 AD 转换的 STB 信号和控制摄像头工作的控制信号。

2 WDM 驱动程序的开发

Windows 2000/XP 是基于 NT 内核的真正 32 位可抢先可重入的虚拟内存操作系统。其 I/O 管理器负责对设备的管理, 包括文件系统、中间介质和设备驱动。WDM 驱动程序是操作系统可信赖的组件, 运行于 Intel X86 处理器的 Ring 0 级, 可以执行特权指令, 并可以访问任何 IO 端口。

遵照标准的 WDM 驱动程序框架^[3-6], 编写了下面的例程:

(1) DriverEntry 例程^[3]。

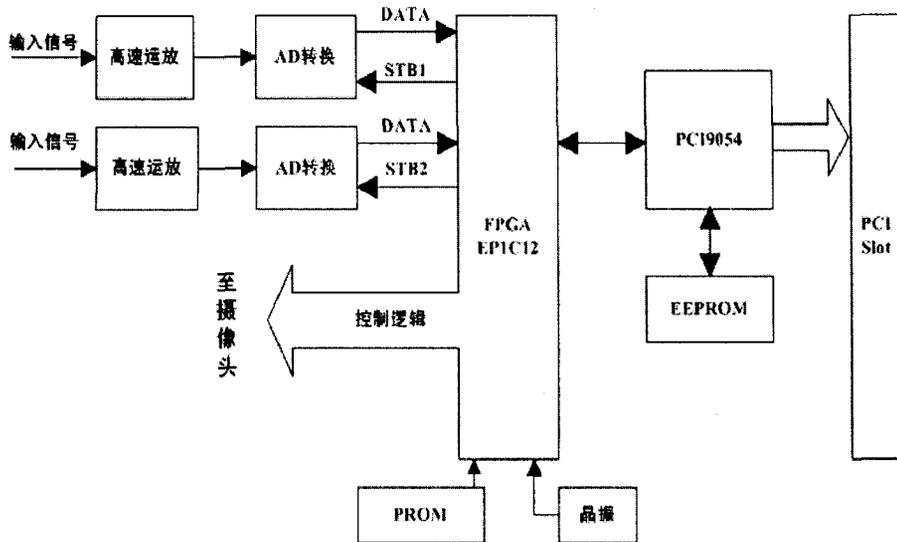


图 1 系统硬件框图

当 IO 管理器装入驱动程序时,首先调用这一例程。DriverEntry 需要执行一系列全局初始化和硬件复位等工作。主要是向系统注册其他例程的调用入口地址,如 Create, Close, PnP 等分派例程。

(2) AddDevice 例程^[4,5]。

由于一台计算机上可能同时插入多块采集卡,因此驱动程序需要具备同时管理多个设备的能力。IO 管理器在枚举到每块板卡时都会为其调用 AddDevice,该例程完成设备相关的初始化工作。包括调用 IoCreateDevice 创建设备对象,注册设备接口,创建符号连接,这样用户态程序可以通过它访问驱动程序提供的服务。此外 AddDevice 还初始化设备标志为缓冲 IO 方式、设置电源状态。

(3) PnP 分派例程^[4,5]。

在驱动中,关键的分派例程有 PnP 分派例程和 read 分派例程。PnP 例程处理即插即用 IRP 请求,其中最为关键的是副功能码为 IRP_MN_START_DEVICE 的请求。IO 管理器在取得该 PCI 卡的资源需求后即向驱动发送 IRP_MN_START_DEVICE 请求,其处理例程 CameraStartDevice 完成 PCI 资源配置。

CameraStartDevice 首先调用 IoGetCurrentIrpStackLocation 得到当前 IO 堆栈,在其中包含着原始的资源分配信息和转换后的资源分配信息。CameraStartDevice 将这些资源保存在设备扩展中,对于内存资源调用 MmMapIoSpace 将其转换为内核可访问的虚拟地址,而对于中断资源则调用 IoConnectInterrupt 注册设备驱动的中断服务程序 ISR。由于一块板卡支持对两路摄像头图像数据的处理,因此在 CameraStartDevice 例程中调用 KeInitializeDpc 初始化两个 DPC 对象,并将其存储在设备扩展中,这些 DPC 用于驱动和应用程序的通信。

PCI9054 提供两个 DMA 控制器,分别对应一路摄像头的输入数据。这里采用基于公用缓冲区的 MASTER 模式 DMA 传输方式。在 CameraStartDevice 例程中,首先为每路摄像头调用 IoGetDmaAdapter 得到一个 DMA 适配器

结构,再调用 AllocateCommonBuffer 分配一个公用缓冲区,然后 CameraStartDevice 初始化设备扩展,并调用 HAL 提供的 WRITE_PORT_ULONG 初始化 PCI9054 寄存器。

CameraStopDevice, CameraRemoveDevice 和 CameraSurpriseRemoveDevice 分别处理 IRP_MN_STOP_DEVICE, IRP_MN_REMOVE_DEVICE 和 IRP_MN_SURPRISE_REMOVAL 请求。它们的功能相似:

首先设置 PCI9054 寄存器终止当前未完成的数据传输,然后依次调用 FreeCommonBuffer, PutDmaAdapter 和 IoDisconnectInterrupt 释放公用缓冲区、DMA 适配器以及注销中断服务。

对于其余 PnP 请求,调用 DefaultPnpHandler 简单地将其传递给下层驱动处理。

(4) Create 和 Close 例程^[6]。

驱动的 Create 和 Close 例程 CameraCreateClose 处理 IRP_MJ_CREATE 和 IRP_MJ_CLOSE 请求。当上层应用程序调用 CreateFile 和 CloseHandle 时,IO 管理器会向驱动发送 IRP_MJ_CREATE 和 IRP_MJ_CLOSE 请求。为了实现应用程序与驱动共享内存,CameraCreateClose 例程在处理 IRP_MJ_CREATE 请求时为每路摄像头执行下面的代码:

```
devExt->pmdd0 = IoAllocateMdl ( devExt->comBufAddress0, 0x4000, FALSE, FALSE, NULL);
if ( devExt->pmdd0 == NULL )
{
    Irp->IoStatus.Status = STATUS_UNSUCCESSFUL;
    Irp->IoStatus.Information = 0;
    IoCompleteRequest (Irp, IO_NO_INCREMENT);
    return STATUS_UNSUCCESSFUL;
}
MmBuildMdlForNonPagedPool (devExt->pmdd0);
devExt->UserSpaceAddress0 = MmMapLockedPages (devExt->pmdd0, UserMode);
```

这样,应用程序可以通过保存在设备扩展中的 UserSpaceAddress0 和 UserSpaceAddress1 直接访问两路 DMA 的公用缓冲区。而在对 IRP_MJ_CLOSE 的处理中 CameraCreateClose 例程首先终止当前还没完成的 DMA 传输,然后再释放上面代码中分配的 MDL。

(5) DeviceIoControl 和 Read 例程。

Win32 API DeviceIoControl 产生主功能码为 IRP_MJ_DEVICE_CONTROL 的 IRP,而 ReadFile 则会产生

IRP_MJ_READ IRP。应用程序通过 DeviceIoControl 例程实现对采集卡的特定操作,如通过设置 FPGA 内指定寄存器而启动一次数据采集。Read 例程将 DMA 公用缓冲区的数据拷贝到用户程序提供的用户态缓冲区中。

(6) Unload 例程。

当需要将驱动从系统中卸载时,IO 管理器调用其 Unload 例程。Unload 例程需要取消由 DriverEntry 例程所作的任何操作。

(7) 中断服务例程(ISR)和 DPC 例程^[5]。

当设备产生中断时,系统调用驱动程序的 ISR。由于 PCI 中断可以被多个设备共享,中断服务程序首先判断该中断是否该驱动程序管理的设备产生,如果不是则立即返回,内核会接着调用其他中断服务程序处理该中断。如果是,则 ISR 负责响应该中断,并排队相应的 DPC 例程作进一步的处理。DPC 例程很简单,通过调用 KeSetEvent 来通知应用程序读取数据。

为了支持轮询方式,在驱动中实现两个标志位分别对应两路摄像头,在 ISR 则判断哪路摄像头的数据采集完成,并置位对应的标志。这样应用程序可以通过 DeviceIoControl 查询这个标志来实现轮询方式的数据采集。

在该驱动程序的开发中,并没有对 StartIo 例程做特殊处理。通常驱动程序会排队 IRP,然后在 StartIo 例程中处理这些 IRP。在此处的驱动中,上层应用程序必须调用 DeviceIoControl 向 FPGA 内一特定位置的寄存器写入适当的数启动数据采集,然后采用轮询或者中断方式等待数据采集完成。当数据采集完成后,应用程序再调用 ReadFile 函数或者 DeviceIoControl 读取数据。因此,能保证每次只有一个 IRP 等待处理,无需利用 StartIo 例程排队处理 IRP。

(上接第 122 页)

要的 XML 索引文件,同时完成课件页面的生成,进行压缩打包。

生成的压缩包解压后可直接作为网络多媒体课件访问并播放,也可以藉由支持标准的各种后期加工软件重新编辑。

3 结论

标准化实时远程教学系统综合各种直播系统的优点,针对操作性比较强的课程实现了电脑桌面操作直播,对习惯于板书的教师提供了电子白板的支持,以在线文本交流的方式实现了学生与教师、学生与学生之间的互动,充分利用现有的各种 PPT,Word 等素材资源,减轻教师负担,并实现了标准化课件的生成,有利于课程资源的交流与共享。标准化以及实时性顺应网络教育发展的潮流。系统灵活的配置和强大的扩展能力,可以适应各种实际环境的需要。系统进行了多次测试,均收到较好的效果。

应用程序读取数据有两种方式:一是利用 ReadFile win32 函数读取,最终系统会调用驱动中的 Read 例程将 DMA 公用缓冲区中的所有数据拷贝到用户态缓冲区中。另外一种方法是利用共享内存技术。应用程序通过调用 DeviceIoControl 直接返回 devExt->UserSpaceAddress0 或者 devExt->UserSpaceAddress1,通过这两个变量,应用程序可以直接访问内核态公用缓冲区中任意位置的数据。

3 结束语

文中介绍了一种基于 PCI 总线的数据采集卡驱动程序的开发,对开发中的一些特殊处理做了详细的阐述。WDM 驱动程序是 Windows 2000/XP 上的标准驱动程序模型。开发 WDM 驱动一方面需要对 Windows 内核有较深刻的理解,另一方面也需要对所控制的硬件设备和客户的需求有清晰的认识,只有这样才能开发出真正高效的驱动程序。

参考文献:

- [1] Shanley T. PCI 系统结构[M]. 北京:电子工业出版社, 2001.
- [2] PLX Technology[Z]. PCI9054 Data Book,1999.
- [3] 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.
- [4] Baker A. Windows NT 设备驱动程序设计指南[M]. 北京:机械工业出版社,1997.
- [5] Oney W. Programming the Microsoft Windows Driver Model [M]. US:Microsoft press,1999.
- [6] Microsoft Corporation. DDK Documentation[Z]. 2003.

参考文献:

- [1] 张栓记,何丕廉,胡敏.我国网络远程教学技术平台的研究现状与未来趋势[J]. 现代教育技术,2005(1):40-42.
- [2] Patel G, Tabrizi M H N. E-Class - a Multimedia and Web based Distance Learning System[A]. Proceedings International Conference on Information Technology: Coding and Computing, IEEE Computer Society 2002[C]. Las Vegas, NV, USA: [s. n.],2002. 524-528.
- [3] 杨宗凯,吴砥,刘清堂.网络教育标准与技术[M]. 北京:清华大学出版社,2003.
- [4] Shaw J. Introduction to Digital Media and Windows Media 9 Series[M]. US:Microsoft Corporation, 2004.
- [5] Huang S, Hu Hui. Integrating Windows Streaming Media Technologies Into a Virtual Classroom Environment[A]. Proceedings International Symposium on Multimedia Software Engineering, IEEE Computer Society 2000[C]. Taipei: [s. n.],2000. 411-418.
- [6] Microsoft Corporation. Windows Media Encoder SDK[M]. US:Microsoft Corporation,2003.