

A Data Acquisition System Based on PCI

基于 PCI 总线的数据采集系统

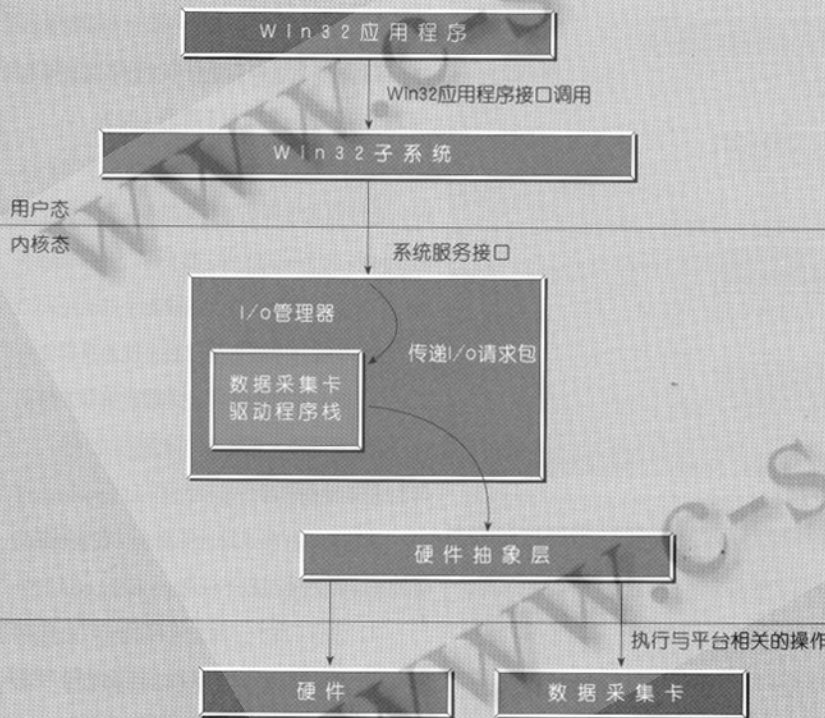


图1 数据采集系统体系结构

摘要: 本文提出了一种基于PCI总线的可变速率数据采集系统的结构以及该系统的实现,并给出了即插即用可变速率数据采集卡的WDM驱动程序设计方法,以及构建在采集卡和WDM驱动程序基础之上的Win32应用程序开发实例。

关键词: PCI总线接口 可变速率 数据采集 WDM驱动程序

1 数据采集系统体系结构

基于PCI总线接口的数据采集系统基本上由3部分构成,即数据采集卡、数据采集卡驱动程序和Win32应用程序。如果出于系统简化的考虑,可以去除数据采集卡驱动程序部分,而由Win32应用程序直接通过I/O指令来操作数据采集卡。不过这样一来系统的层次比较混乱,而且受操作系统的限制(如只能应用于Windows 9x系列中),不具有好的拓展性。

基于上述考虑,整个数据采集系统构建在Windows 98/2000系统之上,呈现如图1所示的体系结构(以Windows 2000系统为例)。

从上图可以看出,构成数据采集系统的3个组成部分分别位于操作系统的不同位置。其中Win32应用程序是整个采集系统的最高层,直接和客户进行交互,处理采集到的数据,并将结果显示给客户。它位于Windows 2000操作系统的用户态下,是采集系统的对外接口。

数据采集卡驱动程序栈是衔接Win32应用程序和数据采集卡的桥梁,是整个数据采集系统的中间层。它提供连接到数据采集卡的软件接口,在装入后成为操作系统内核的一部分,供Win32应用程序使用标准的Win32 API(如CreateFile、ReadFile、WriteFile和CloseFile等)对硬件进行I/O操作。

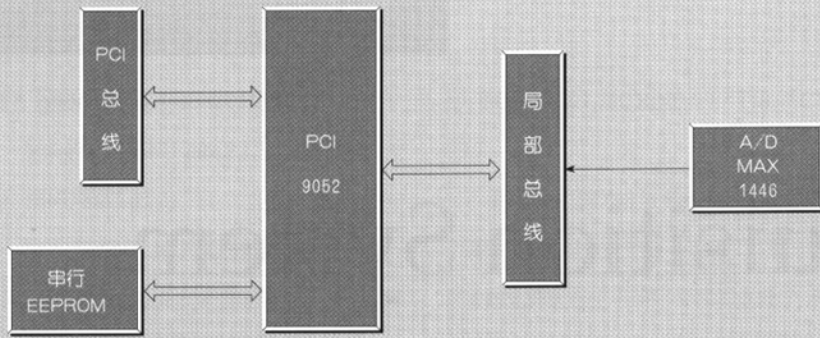


图2 数据采集卡原理图

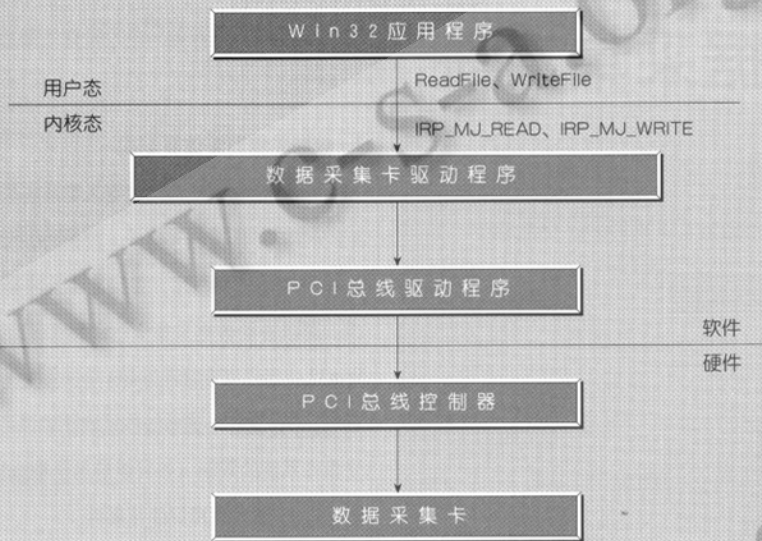


图3 应用程序、驱动程序和采集卡逻辑关系

数据采集卡是整个数据采集系统的基础,它通过传感器实时采集外部数据,经A/D变换后转化成数字信号,经过PCI接口控制芯片(PCI 9052),最后由PCI总线接口作为数据传输总线传入微机。

2 数据采集卡电路设计

数据采集系统的硬件电路主要包括采集电路、PCI接口电路、串行EEPROM和逻辑控制电路,它们的联结方式和逻辑结构如图2所示。

2.1 数据采集电路

由前端数据采集器件和A/D变换器实现,完

成数据采集和模数转换。A/D变换器采用MAX公司的MAX 1446(一路输入信号,10-Bit输出信号,60Mps, +3.0v)。

2.2 PCI接口电路

由于PCI总线规范较复杂,需采用PCI总线接口芯片以简化硬件设计,完成完整的PCI主控模块和目标模块接口功能,将复杂的PCI总线接口转换为相对简单的用户接口,设计中采用PLX公司的PCI 9052作为接口控制芯片。

2.3 串行EEPROM

使用串行EEPROM作为外部扩展ROM,存储PCI设备的配置信息,包括设备需要的内存空

间或I/O地址空间大小等资源请求信息。在PCI设备初始化过程中,串行EEPROM中的内容将会下载至PCI局部总线配置寄存器,PCI BIOS启动后读取各个PCI设备配置空间中的基地址寄存器,获取每个PCI设备分配所需的资源(I/O空间或内存空间)类型和大小等信息并且为这些PCI设备分配好所设定的资源,然后再将分配的内存空间或地址空间的基地址回写到PCI设备的各个基地址寄存器。我们采用的是Microchip公司的93LC46B。

3 WDM设备驱动程序设计

在WDM模型中,每个硬件设备至少有两个驱动程序:一个功能驱动程序和一个总线驱动程序。数据采集系统中应用程序、驱动程序和采集卡之间的逻辑关系如图3所示:

3.1 总线驱动程序

总线驱动程序是为实际的I/O总线服务,由Windows操作系统提供,负责以下的工作:

- (1) 枚举PCI总线上的设备;
- (2) 向操作系统报告总线上的动态事件;
- (3) 响应即插即用和电源管理的I/O请求;
- (4) 管理PCI总线上的设备。

3.2 功能驱动程序

功能驱动程序是设计的重点,在该系统中,功能驱动程序完成以下任务:

- (1) 设备初始化。驱动程序有一个主要的初始化入口点DriverEntry例程,当驱动程序装入时,由内核调用DriverEntry例程,在DriverEntry中,设置一系列的回调例程来完成设备运行过程中出现的各种I/O请求包(IRP)消息的处理。
- (2) 设备创建。系统启动时,总线枚举器会发现总线上的所有设备,会自动寻找并安装数据采集卡的驱动程序,并由驱动程序中的处理即插即用功能模块自动处理AddDevice例程及其他PnP消息,AddDevice使用IoCreateDevice函数创建设备对象,再使用IoRegisterDeviceInterface函

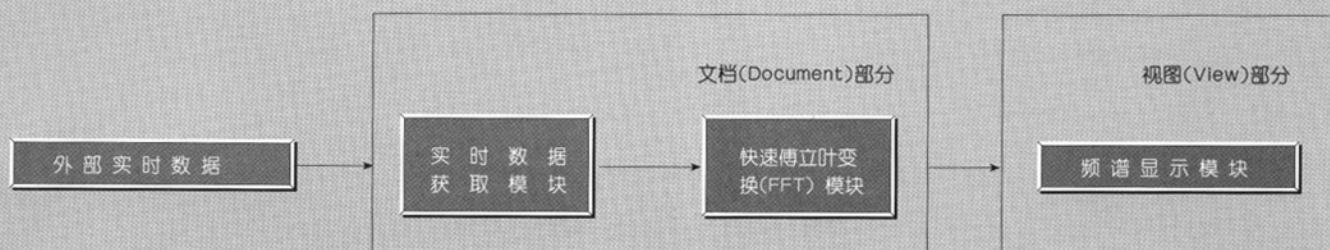


图4 Win32应用程序流程

数将设备注册为一个特定的设备接口,然后使用IoAttachDeviceToDeviceStack函数将当前驱动程序添加到设备栈中。

(3) 设备删除。收到“删除设备”请求时,通过调用IoDetachDevice函数把本驱动程序的设备对象从设备栈脱开,然后调用IoDeleteDevice函数删除设备对象和它的设备扩展。

(4) Win32程序接口。Win32程序接口是驱动程序给Win32用户程序访问该设备的一组命令,也称为设备接口。驱动程序响应CreateFile、ReadFile、WriteFile、DeviceIoControl、CloseHandle等Win32函数的调用,通过处理IRP来响应来自Win32应用程序的打开、关闭、读、写和IOCTL等请求。

(5) 访问硬件。“启动设备”请求在IRP栈的Parameters.StartDevice结构中传递AllocatedResources和AllocatedResourcesTranslated域,我们使用AllocatedResourcesTranslated连接中断向量,映射I/O空间和内存空间。

3.3 驱动程序的安装

驱动程序的安装必须采用INF文件方式。INF文件中含有一个WDM设备驱动程序所有必需的信息,包括要复制的文件列表、要创建的注册表项等。

4 Win32应用程序设计

Win32应用程序是运行在用户态下的,采用标准的Windows图形界面风格,它的主要功能就是对数据采集卡采集到的实时数据进行频谱分

析,并将分析结果显示出来。程序设计遵循VC的文档/视图模式,将数据处理部分和显示部分从逻辑上分开,基本的模块图如图4所示。

其中实时数据获取模块通过驱动程序的Win32用户接口获得采集到的实时数据,经FFT变换后得到其频谱数据,最后将频谱显示给终端用户。

5 系统流程

数据采集卡、驱动程序、应用程序这三个功能模块的数据流程如图5所示。

其中设置“采集数据的数量是否达到要求”一步的目的是调整数据采集的速率。本系统中这一步采用由CPU来主控控制电路,实现对A/D的控制,调整采集速率。

6 结束语

接口控制芯片PCI 9052只能作为Target,不能作为Master,使用如PCI 9054,或是AMCC公司的S5933等可作为Master的PCI接口控制芯片,就可实现DMA方式的数据传输,进一步提高数据采集速率。■

参考文献

- 1 刘晖译, PCI系统结构(第四版), 电子工业出版社, 2000.
- 2 PLX PCI 9052 Data Book, 2000.
- 3 Chris Cant(美)著, 孙义等译, Windows WDM设备驱动程序开发指南, 机械工业出版社, 2000.

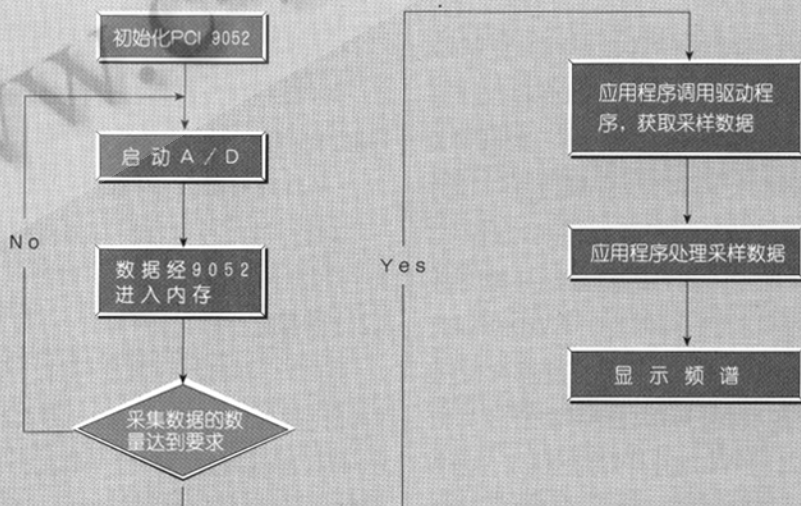


图5 数据采集系统数据流程