

基于 PC 的微机继电保护测试仪通信控制卡的设计及其 WDM 驱动程序的实现

唐兵, 周建中, 邹拖武, 赵炳, 何宇

(华中科技大学, 湖北 武汉 430074)

摘要: 微机继电保护测试仪作为一种高速测试装置, 要求具有很高的实时性和稳定性。阐述了基于 PCI 总线的微机继电保护测试仪通信控制卡的硬件结构及实现方法, 介绍了 WDM 驱动程序的特点及 PCI 通信控制卡驱动程序主要模块的开发, 完成了整个系统的整合设计。实际应用表明: 该系统有效地完成了各种继电测试试验。

关键词: 微机继电保护测试仪; PCI; DSP; DRAM; 驱动程序; WDM

中图分类号: TM774 文献标识码: A 文章编号: 1003-4897(2006)15-0050-04

0 引言

PCI 局部总线是微型计算机中处理器与外围控制部件、扩展卡之间的互连接口, PCI 局部总线规范是互连机构的协议。PCI 总线具有负载能力强、传输速率高、自动配置、支持即插即用等众多优点。所以 PCI 总线一出现就受到众多厂商和研究机构的关注, 在自控、电力等领域得到越来越广泛的应用。

继电器保护测试仪是一种高速测试电力装置, 要求很高的实时性, 而鉴于 PCI 总线以上众多优点, 尤其是其传输速率高达 132 M/s, 我们选用 PCI 总线作为上位机和下位机的通信接口。

在设计和使用 PCI 设备时, Windows 操作系统为了保证系统的安全性和可靠性, 对应用程序访问硬件资源加以限制。这就要求设计驱动程序以实现上位机程序对 PCI 设备的访问。

1 PCI 控制卡的设计

1.1 PCI 控制卡的功能和结构

在继电保护测试仪中, PCI 控制卡担任着重要的通信控制作用。该控制卡的整体框架如图 1 所示。

上位机将试验数据通过 PCI 总线传送至通信控制卡, 控制卡通过 PCI 接口桥芯片将数据存放至本地双端口 RAM 中, 并在上位机程序的控制下, 产生

对控制卡主控制器 DSP 的中断。DSP 在中断服务程序中读取双端口 RAM 中的数据, 从而达到与上位机通信的目的。反过来, 当 DSP 写入双端口 RAM 新数据时, 在 DSP 程序的控制下, 产生对上位机的中断。上位机在中断服务程序中读取双端口 RAM 中的数据, 从而达到与 DSP 通信的目的。

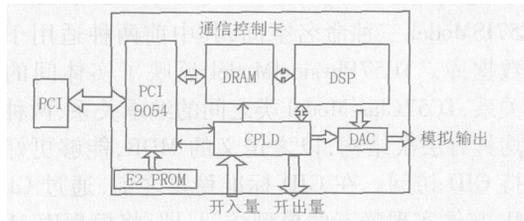


图 1 控制卡结构图

Fig 1 Structure of control card

1.2 PCI 接口设计

实现 PCI 总线协议一般有两种方法, 一是采用 PCI 总线控制芯片; 二是用 FPGA 设计实现, 但 PCI 协议比较复杂, 难度较大。因而本设计采用第二种方法, 选择 PLX 公司的 PC9054 作为总线控制芯片。PC9054 符合 PCI 本地总线规范, 突发传输速率高达 132 MB/s。

我们选取 Cypress 公司的 CY7C025AV 双端口 RAM 作为上位机和 DSP 相互通信的数据缓冲区, 它存取速度高达 15 ns, 并且具有如下特性: 写入左中断寄存器时, 将产生左中断; 写入右中断寄存器时, 将产生右中断。利用这一特性可以达到上位机和 DSP 相互通信的目的。

设计过程中所考虑到的问题:

1) PCI 插槽包含了为 PCI 元件定义的全部信

基金项目: 国家自然科学基金项目 (50579022); 国家自然科学基金重点项目 (50539140); 高等学校博士学科点专项科研基金 (20050487062)

号线。另外还有两个引脚是与连接器本身有关的,即 PRSNT1#和 PRSNT2#。它们的用途有二:其一,用来表明插槽上存在一块板;其二,是提供该板对电源要求的有关信息。表 1 给出了扩展板上 PRSNT# 引脚的设置情况。

表 1 PRSN#引脚的设置定义
Tab 1 Set definition of pin PRSN#

PRSNT1#	PRSNT2#	扩展板配置
开路	开路	不存在扩展板
地	开路	有扩展板,最大功耗为 25 W
开路	地	有扩展板,最大功耗为 15 W
地	地	有扩展板,最大功耗为 7.5 W

根据表 1 要求,本控制卡 PRSNT#的配置为 PRSNT1#接地,PRSNT2#开路,即最大功耗为 25 W。

2) PC 9054 本地地址总线硬件连接包括如下内容:数据总线和地址总线与数据交换缓冲双端口 RAM 的连接、读写信号的连接、中断引脚的连接、本地地址总线模式选择以及总线宽度选择。需要指出的是,PC 9054 中断信号与双端口 RAM 左中断信号 NTL 相连,当 DSP 写入双端口 RAM 新数据时,将通过双端口 RAM 左中断信号通知上位机读取更新的数据。

3) PC 9054 的配置

配置空间是 PC 规范所特有的,是除了 D 空间和内存空间的第三种空间。系统通过配置空间可以了解到 PC 的功能、中断和基地址,因此在系统启动时需要配置空间进行设置以分配系统资源。

PC 9054 的配置寄存器将在系统启动时由 E-2PROM 加载,其中配置信息将由烧写器写入。PC BAR0 配置为内存方式,映射寄存器至系统内存空间;PC BAR1 配置为 I/O 方式,映射寄存器至系统 I/O 空间;PC BAR2 和 PC BAR3 配置为内存方式,映射本地地址空间至系统内存空间。配置 Vendor 和 Device 为 10B5 和 9054,作为系统识别本 PCI 卡的标识信息。

1.3 DSP 及其外围电路设计

在通信控制卡中,DSP 作为主控制器芯片,是整个控制卡的核心。本测试仪采用 TI 公司的高速数字信号处理器 TMS320F2812 (简称 F2812)。F2812 将共享双端口 RAM 作为其外存使用,与 PC 9054 交换上位机传送过来的参数和命令,同时控制着数模转换器输出要求的模拟信号,与 CPLD 配合实现开入量检测及开出量输出。

F2812 与双端口 RAM 的数据总线和地址总线连接,读写信号直接连接,双端口 RAM 片选信号由

F2812 的区域选择信号 XZCS0AND1#,双端口 RAM 的右中断信号 NTR 连接至 F2812 的中断 NT1,当上位机更新双端口 RAM 时,将通过双端口 RAM 右中断通知 F2812 读取更新的数据。

CPLD 在通信控制卡中作为逻辑控制协调核心,其将 F2812 发送过来的地址信号进行译码从而片选与同步 DAC。CPLD 还负责检测开入量,通过中断通知 F2812 读取当前开入量。

2 WDM 驱动程序的实现

2.1 WDM 驱动程序模型

WDM (Windows Driver Model) 是微软公司全新的驱动程序模式,支持即插即用、电源管理和 WMI 技术,运行平台是 Windows 98/2000/XP 操作系统。

WDM 采用分层的驱动程序结构,而且是基于对象的。为了便于对硬件的管理,WDM 对每一个单一的硬件引入了一些数据结构,图 2 左部分描述了一个 DEVICE_OBJECT 数据结构栈。在数据结构栈中最底层的是物理设备对象 (Physical Device Object),用于描述我们的设备与物理总线的关系,简称为 PDO。在 PDO 的上面,有功能设备对象 (Function Device Object),用来描述设备的逻辑功能,简称为 FDO。在 FDO 的上面或下面,有过滤设备对象 (Filter Device Objects),简称为 FDO。数据结构栈中的每一个对象都属于一个特定的驱动程序,如图 2 中间的虚线所指示,PDO 属于总线驱动程序,FDO 属于功能驱动程序,FDO 属于过滤驱动程序。本设计需要实现功能驱动程序。

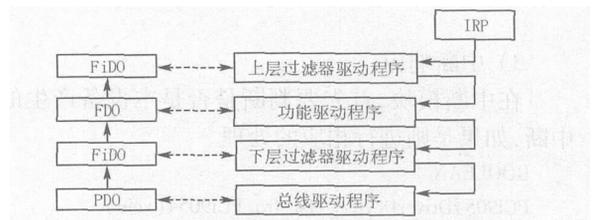


图 2 WDM 中设计对象和驱动程序的分层结构

Fig 2 Step structure of design object and driving program in WDM

2.2 具体实现

Numega Lab 公司的 DriverStudio 是一套开发、调试和检测 windows 平台下设备驱动程序的软件包。DriveWizard 是 DriveWorks 创建 WDM 框架程序的工具,其 DriveWizard 可以快速生成 WDM 驱动程序框架。在驱动程序框架中写入相对于设备的特定代码,编译后即可得所需的驱动程序。

驱动程序的关键模块如下:

1) 驱动程序调用方式

应用程序调用驱动程序有两种方式。一种是 GUID 方式,另一种是 LinkName 方式。GUID 模式需要包含 DriveWorks 的一个头文件 devintf.h,而我们开发应用程序的时候是不希望再去装 DriverStudio 的,因此采用符号连接方式打开设备较合适,代码如下:

```
char * sLinkName = "\\.\ \BBPCDevice0";
CreateFile(
    sLinkName, /按 LinkName 方式打开设备
    ..... );
```

2) 读写控制

读写控制是在 DeviceControl 函数中实现的。在此函数中根据不同控制码调用相应的处理函数,代码如下:

```
NTSTATUS
PC0054DriverDevice::DeviceControl(Kernel)
{
    .....
    switch (IoctlCode())
    {
    case PC0054DRIVER_IOCTL_Read:
        status = PC0054DRIVER_IOCTL_Read(D);
        break;
    case PC0054DRIVER_IOCTL_Write:
        status = PC0054DRIVER_IOCTL_Write(D);
        break;
    .....
    }
}
```

3) 中断响应

在中断模块,首先要判断是否是本设备产生的中断,如果是则进行相应的处理。

```
BOOLEAN
PC0054DriverDevice::Isr_Irq_PC0054(void)
{
    .....
    if((status & 0x8000) == 0) /判断是否本设备产生的中
断
    {
        return FALSE; /不是本设备的中断
    }
    .....
}
```

2.3 驱动程序的调用

应用程序调用驱动程序的过程是:应用程序先

调用 CreateFile 函数打开设备,然后调用 DeviceIoControl 函数和驱动程序进行通信,包括从驱动程序读数据和向驱动程序写入数据两种情况,最后当应用程序退出时,调用 CloseHandle 关闭设备。程序具体说明如下:

1) 打开设备

应用程序通过调用 CreateFile() API 函数来获得设备句柄,然后通过该句柄与驱动程序进行通信。为了便于应用程序的调用,将 CreateFile 函数封装成如下形式:

```
char * sLinkName = "\\.\ \BBPCDevice0";
HANDLE OpenByName(void)
{
    // Create a handle to the driver
    return CreateFile(
        sLinkName,
        ..... );
}
```

如果打开设备成功,则返回设备句柄,否则返回无效句柄。

2) 读、写数据

如果打开设备成功,则可以利用返回的设备句柄调用 DeviceIoControl() API 函数来进行数据的读写工作。同样为了便于应用程序的调用,将 DeviceIoControl 函数封装在 WritePC0054() 和 ReadPC0054() 中,具体如下:

```
void WritePC0054(ULONG offset, ULONG num)
{
    .....
    if (!DeviceIoControl(hDevice,
        PC0054DRIVER_IOCTL_Write,
        .... )
    )
    {
        AfxMessageBox("写入数据失败!");
        return;
    }
    .....
    ULONG ReadPC0054(ULONG offset)
    {
    .....
    if (!DeviceIoControl(hDevice,
        PC0054DRIVER_IOCTL_Read,
        .... )
    )
    {
```

```

AfxMessageBox("读取失败!");
return 0;
}
.....
}

```

3 整体调试和试验结果

3.1 整体调试

整个系统访问流程如下:由应用程序发出控制命令和数据,调用内核函数访问驱动程序,驱动程序接收命令和数据并对 PCI卡进行硬件读写,然后把数据再返回驱动程序,由应用程序核对数据的正确性。

利用 DriverStudio 自带的 SoftIE、DriveMonitor 工具可以很方便地进行调试,具体步骤如下:

- 1) 使用 DriveMonitor 加载驱动程序,启用 softIE 跟踪调试。
- 2) 对于中断响应代码,利用 SoftIE 产生虚拟中断,单步跟踪中断。
- 3) 发送数据,通过查看内存的数据,确定数据传输是否正确。

3.2 试验结果

该 PC 控制卡通过驱动程序成功地接收了来自上位机的数据,经过处理输出所需的信号,有效地完成了对外围器件的实时控制。

参考文献:

- [1] 武安河,郇铭,于洪涛. Win 2000 / XP WDM 设备驱动程序开发 [M]. 北京:电子工业出版社, 2003.
WU An-he, TAN Ming, YU Hong-tao. Win 2000/XP WDM Device Driving Program Development [M]. Beijing: Publishing House of Electronics Industry, 2003.

收稿日期: 2006-01-04; 修回日期: 2006-02-15

作者简介:

唐兵 (1980 -),男,硕士研究生,专业方向为系统分析与集成; E-mail: tangbingaaa@tom.com

周健中 (1959 -),男,教授,博士生导师,主要研究方向为工业控制及自动化系统,模糊控制与人工智能理论,计算机大型数据库及网络系统,基于 INTERNET/ INTRANET 的异质集群网络的计算机通讯技术;

邹托武 (1981 -),男,硕士研究生,专业方向为系统分析与集成。

Design of communication control card based on PCI and implementation of WDM driving program in microcomputer-based relay protection testing device

TANG Bing, ZHOU Jian-zhong, ZOU Tuo-wu, ZHAO Bing, HE Yu
(Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: As a high-speed testing device, microcomputer-based relay protection testing device demands high performance of real time and stability. This paper introduces the hardware structure and the implementation method of communication control card based on PCI bus. It also details the characteristic of WDM driving program and the design of the main modules to complete the hardware and software design of the whole system. The application indicates that this system can effectively accomplish various relay protection testing experiments.

This project is supported by National Natural Science Foundation of China (No. 50579002), Key Project of National Natural Science Foundation of China (No. 50539140) and Research Fund for the Doctoral Program of Higher Education (No. 20050407062).

Key words: microcomputer-based relay protection testing device; PCI; DSP; DRAM; driving program; WDM

(上接第 49 页 continued from page 49)

Abstract: Real-time database acts as a bridge between the control system and the information management system in power supply enterprise. Plant Information System is a commercial real-time database provided with excellent performance such as fast-access, high-reliability, and efficient-storage. It is critical to combine the characteristics of PI and the requirements of the power industry when applying PI system. On this view point, this paper introduces the system structure and key modules of PI, and then presents the typical use of PI in the power supply enterprise. To improve openness of the whole information integration system, it discusses the design of the PI system that conforms to IEC61970 in such aspects as model definition and data access. The problems should be paid attention to and relevant suggestions for implementing PI system abide by IEC61970 are pointed out. At last, a feasible solution which meets the requirements of power information integration is brought forward.

Key words: PI; power supply enterprise; IEC61970; common information model; component interface specification