PCI 总线 WDM 驱动程序的设计方法与实例

陈大科，李军予
（江苏自动化研究所 江苏 连云港 222006）

摘 要：探讨基于 Windows 操作系统的 WDM 型 PCI 总线接口卡驱动程序设计方法。DriverStudio 软件是一套设备驱动程序的开发、调试和测试的工具包，介绍 DriverStudio 环境的配置方法及开发 WDM 驱动程序的详细步骤，设计 PCI 总线的驱动程序以及调用驱动程序的应用程序，实现了 32 位数据的传输。实验结果表明，采用 DriverStudio 软件能够简化驱动程序的编写，达到较高的编程效率。

关键词：WDM 驱动模型；设备驱动程序；PCI 总线；DriverStudio
中图分类号：TP336 **文献标识码**：B **文章编号**：1004－373X(2006)22－099－02

Method and Example of PCI Bus WDM Design

CHEN Dake, LI Junyu
(Jiangsu Automation Research Institution of CSIC, Lianyungang, 222006, China)

Abstract: How to design a PCI bus driver program based on Windows Driver Model(WDM) is introduced. The DriverStudio is a design kit to develop and debug device driver program. The specific steps of design a WDM and how to configure the DriverStudio are illustrated in this paper. Then a PCI bus driver program and its application are complied, and the device transfers data achieves 32bit. Test results indicate that it can simplify programming WDM driver and can efficient the program by the DriverStudio.

Keywords: WDM driver model; device driver model; PCI bus; DriverStudio

1 引 言

WDM(Windows Driver Model)模型是从 WinNT 驱动程序结构上发展起来的一个跨平台的驱动程序模型，能够在 Windows 98，NT 及 Windows 2000 操作系统中运行。WDM 的通用性强，并且随着软件技术的发展，将逐步取代原先的 VxD 类驱动程序，成为一种通用的驱动程序的构架。WDM 驱动程序可以采用 Microsoft 公司提供的工具 DDK 来设计，但是由于 DDK 中的指令很复杂，并且很难使用，只有专业的程序设计人员才能掌握，因此目前有许多公司提供专用的开发工具软件，如表 1 所示，这些工具软件相对于 DDK 来说，具有易学易用的特点。

表 1 设备驱动程序软件工具

公 司	驱动程序类	通用驱动程序
Blue Water Systems	WinDK	WinRT
KRF Tech		WinDriver
Compuware Numega	DriverWorks	DriverAgent
Open Systems Research	WinDK	

收稿日期：2006－07－12

Compuware Numega 公司的 DriverStudio 是一套完整的设备驱动程序的设计、调试软件包，用来开发驱动程序非常方便，只需根据其向导逐步添加代码就可完成驱动程序的框架设计。用 DriverStudio 开发出的程序是基于内核的，代码效率与用 DDK 开发的相同。

2 DriverStudio 软件简介

DriverStudio 是一套用来简化微软 Windows 平台下设备驱动程序的开发、调试和测试的工具包。主要包括下列工具模块：VToolsD，SoftICE，DriverWorks。

VToolsD 是一个用来开发 Win9X（Windows 95 和 Windows 98）操作系统下设备驱动程序（VxD）的工具，VToolsD 中包含了生成驱动程序源代码的各种工具库以及一些驱动程序样本。SoftICE 是用来进行设备驱动程序的调试。

DriverWorks 用来开发 Windows NT 下、Windows 98 与 Windows 2000 操作系统下驱动模型（WDM）设备驱动程序。他是通过调用 DDK 的函数来设计驱动程序的，DriverWorks 中包含一个非常完善的源代码生成工具（DriverWizard）以及相应的类库和驱动程序样本。用 DriverStudio 设计 WDM 程序时还同时生成一个测试程序，用来对硬件功能的测试。

3 DriverStudio软件的安装

DriverWorks是通过调用DDK的函数来设计驱动程序的,并且他的生成代码的修改和编译环境是VC6.0。因此,在安装DriverStudio开发软件之前,首先要安装VC6.0,再装Windows 2000 DDK。

当按以上顺序安装好软件后,还需要对DriverStudio的库文件进行编译。首先启动Numega DriverStudio\Tools菜单中的Setu PDDK and Start MSVC设置BASEDIR环境变量。然后在VC6.0中编译DriverStudio\DriverWorks\Source中的vdwlibs.dsw工程文件。在VC6.0中嵌入DriverStudio的DriverWorks,就可以进行WDM程序的开发了。

4 PCI总线WDM程序的设计

PCI是一种即插即用的接口总线,因此,每块PCI接口板都要配有驱动程序,使计算机启动时能进行资源分配。所设计的PCI接口板用于数据采集,他的参数配置如下:

device ID:0004H;设备标识符为0004H。

vendor ID:1173H;销售商标识符为1173H。

BAR0:FFFF0000H;基地址0的偏置点是FFFF0000 H,存储器地址映射。

BAR1:FFFF0001H;基地址1的偏置点是FFFF0000 H.IO口地址映射。

NUMBER_OF_BARS:2;选择2个基地址寄存器。

PCI总线驱动程序的设计步骤如下:

步骤1:启动DriverWorks出现DriverWizard的向导对话框,首先输入PCI数据采集板驱动程序的文件名:MPCI;

步骤2:选择驱动程序的类型。选择其中的WDM项,表示生成WDM类型文件。

步骤3:选择接口板的类型以及填写Device ID和Vender ID。在接口板类型项中选择PCI。Device ID和Vender ID是设备标识符和销售商标识符,要与PCI接口板中的一致,因此填写"0004"和"1173"作为Device ID和Vender ID,Subsystem ID和Revision ID项采用缺省值。

步骤4:添加Add Memory Range,表示通过缓冲区来读写IO,并选择其中的缺省设置。

步骤5:剩下的几个步骤全部选择缺省值设置。这样就完成了WDM驱动程序的框架代码的设计。

用上面的步骤产生的WDM驱动程序的框架代码(Mpci.dsw)需要添加代码,才能完成读、写和控制功能。在VC中打开Mpci.dsw,可以看到他是由2部分组成:一个是用于生成驱动程序的MPCI文件,一个是用于测试驱动程序的TEST-MPCI文件。

因为PCI接口板的功能是读取数据,需要在MPCI文件中添加"读"控制代码。首先打开MpciDevice.cpp文件,找到NTSTATUS MpciDevice::Read(KIr PI)函数并做如下修改,其中的黑体字部分是添加的代码用于IO的读写控制。

```
NTSTATUS MpciDevice::Read(KIr PI)
{
  t << "Entering MpciDevice::Read,"      << I << EOL;
if (FALSE)  // If (Request is invalid)
  {
  I.Information() = 0;
  return I.PnpComplete(this,STATUS_INVALID_PARAM-
  ETER);
  }
  if (I.ReadSize() == 0)
  {
  I.Information() = 0;
  return I.PnpComplete(this,STATUS_SUCCESS);
  }
  if (I.ReadSize() == 4)
  {
  PULONG  pBuffer  = (PULONG)I.BufferedReadDest();
  I.Information() = 4;
  m_MemoryRange0.ind(0x04,(PULONG )I.BufferedRead-
  Dest(),0x04);
  return I.PnpComplete(this,STATUS_SUCCESS);
  }
  return m_DriverManagedQueue.QueueIrp(I);
}
```

程序中的I.ReadSize() =4是指每次读取的字节数是4,即每次读32位数据,PULONG pBuffer = (PULONG)I.BufferedReadDest()表示通过缓冲区读写IO。修改好的程序经过编译后,生成文件为Mpci.sys,存放在sys/i386的目录下。

当把PCI接口板插入计算机后启动计算机,系统显示找到新硬件,按照提示将Mpci.sys装入。然后进入DOS操作系统,运行测试驱动程序Test-Mpci.exe,则屏幕上显示如下信息:

```
 Test_mpci [r n] [w n] [i n]
r initiates a read of specified number of bytes
w initiates a write of specified number of bytes
i initiates an IO Control Code message with specified index val-
ue
Example:
Test_mpci r 32 w 32
read 32 bytes,then write 32 bytes
```

这时用键盘输入:r 4,则显示读入的数据,说明PCI接口板和WDM驱动程序的功能正确。

5 结　语

通过上面的设计实例可以看到,用DriverStudio软件

(下转第103页)

$$g(t)=h(t+T_S/2)+h(t-T_S/2)$$
$$=S_a(\pi t/T_S+\pi/2)\cdot\frac{\cos(\pi t/T_S+\pi/2)}{1-4(t+T_S/2)^2/T_S}+$$
$$S_a(\pi t/T_S-\pi/2)\cdot\frac{\cos(\pi t/T_S-\pi/2)}{1-4(t-T_S/2)^2/T_S}$$
$$=\sin\pi t/T_S\cdot\cos\pi t/T_S$$
$$\left[\frac{1}{(\pi/2-\pi t/T_S)\cdot(1-4(t-T_S/2)^2/T_S^2)}\right.$$
$$\left.-\frac{1}{(\pi/2+\pi t/T_S)\cdot(1-4(t+T_S/2)^2/T_S^2)}\right]$$
$$=\frac{1}{2}\sin\omega_S t\left[\frac{1}{(\pi/2-\pi t/T_S)\cdot(1-4(t-T_S/2)^2/T_S^2)}-\right.$$
$$\left.\frac{1}{(\pi/2+\pi t/T_S)\cdot(1-4(t+T_S/2)^2/T_S^2)}\right]$$

又根据傅里叶变换的时移性和线性性，可得：

$$G(\omega)=F^{-1}[g(t)]$$
$$=\frac{\pi}{\omega_S}g_{2\omega_S}(\omega)\left(1+\cos\frac{\pi}{\omega_S}\omega\right)e^{jT_S\omega/2}+\frac{\pi}{\omega_S}g_{2\omega_S}(\omega)\cdot$$
$$\left(1+\cos\frac{\pi}{\omega_S}\omega\right)\cdot e^{-jT_S\omega/2}$$
$$=\frac{\pi}{\omega_S}g_{2\omega_S}(\omega)\left(1+\cos\frac{\pi}{\omega_S}\omega\right)(e^{jT_S\omega/2}+e^{-jT_S\omega/2})$$
$$=\begin{cases}T_S(1+\cos\omega T_S/2)\cdot\cos(\omega T_S/2) & |\omega|\leqslant 2\pi/T_S\\ 0 & |\omega|>2\pi/T_S\end{cases}$$

所得部分响应系统频谱特性见图 6 所示($T_S=1$)。

其中取第一过零点为系统带宽，即 $B=\omega/2\pi=1/2T_S$(Hz)，且频带利用率 $\eta=\frac{1/T_S}{1/2T_S}=2$(Band/Hz)，又根据 $g(t)$ 表达式可得：

$$\begin{cases}g(0)=0\\ g(\pm\frac{T_S}{2})=1\\ g(\pm\frac{kT_S}{2})=0 \qquad k=\pm3,\pm5,\cdots\end{cases}$$

即：① $g(t)$ 的尾巴衰减更快，与 t^2 成反比；

② 若用 $g(t)$ 作为传输波形，且传输码元间隔为 T_S，则在抽样时刻上仅发生前一码元对当前码元的干扰，当前码元的判定仅需当前传输波形抽样值减去前一码元判定值即可；

③ 前一码元的干扰很大，但其是确定的，故仍可按 $1/T_S$ 的速率传送码元。

3.2 系统的可靠传输[3]

上述方法原理可行，但可能会造成错误的传播，即只要一个码元发生错误，则这种错误会影响以后的码元判决。为了克服这一缺陷，需对输入信号 a_k 在送入系统之前进行预处理，预处理电路框图如图 7 所示。

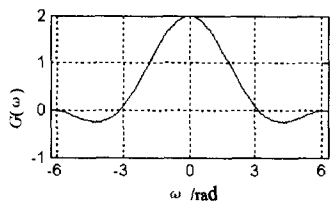

图 6　升余弦系统频率响应

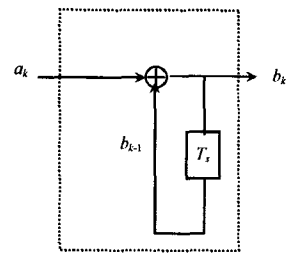

图 7　电路框图

通过对信号预处理电路的分析可知，若对当前 c_k 作模 2 处理便可直接得到发送端的 a_k，此时并不需要预先知道 a_{k-1}，同时，也不存在错误传播情况。

通过以上分析可以看出：数字基带信号在不同的基带系统中传输时，其码元传输速率是受系统频率响应 $H(\omega)$ 确定的系统带宽限制的，只有满足特定的比例关系，才能做到无码间干扰传输，且可做到任意小误差传输。

参 考 文 献

[1] 樊昌信，詹道庸，徐炳祥，等. 通信原理[M]. 北京：国防工业出版社，1998.

[2] 吴大正，杨林耀，张永瑞. 信号与线形系统分析[M]. 北京：高等教育出版社，1998.

[3] 王兴亮，达新宇，林家薇，等. 数字通信原理与技术[M]. 西安：西安电子科技大学出版社，2002.

(上接第 100 页)

能够快速地设计一个 WDM 驱动程序框架，然后再做一些小的改动就可以使用了。现在 PCI 总线已成为计算机的标准接口总线，在开发 PCI 接口板时，驱动程序的设计往往是最困难的工作，有了 DriverStudio 软件，就可以简化驱动程序的设计工作，使硬件设计人员也可以设计驱动程序，并且 DriverStudio 产生的用于测试的程序也有助于上层应用程序的设计。

参 考 文 献

[1] [美]Chris Cant. Windows WDM 设备驱动程序开发指南[M]. 孙义，马莉波，国雪飞，译. 北京：机械工业出版社，2000.

[2] [美]Tom Shanley · Don Anderson. PCI 体系结构[M]. 刘晖，冀然然，夏意军，译. 北京：电子工业出版社，2000.

[3] 李静. 激元成像系统 PCI 驱动程序开发[J]. 现代电子技术，2005，28(12)：35－36，40.

作者简介　陈大科　男，1967 年出生，江苏连云港人，高级工程师。主要从事自动检测与控制技术方向的研究。