

PCIE 数据采集系统的驱动程序开发

梁国龙, 何 昕, 魏仲慧, 王 军

LIANG Guo-long, HE Xin, WEI Zhong-hui, WANG Jun

中国科学院 长春光学精密机械与物理研究所, 长春 130033

Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

LIANG Guo-long, HE Xin, WEI Zhong-hui et al. Driver development of PCIE data acquisition card. Computer Engineering and Applications 2009 45(31) 63-65.

Abstract: In order to improve data transmission speed and accuracy a set of data collection system based on the PCIE interface is developed. The system uses a modular design including the data reception card, the data transport card and the software driver of three parts. A briefing on the basic principles and the composition of self-developed data acquisition card is given. The processes of emulating device driver using DriverStudio, data transferring with DMA controller and event notification in Windows XP operation system are studied. Through the computer test stability and reliability of the system, the driver developed can achieve the high-speed data transmission.

Key words: PCI Express(PCIE); modularization; DriverStudio; Direct Memory Access(DMA); driver

摘 要: 为了提高数据传输速度和准确性, 研制了一套基于 PCIE 接口的数据采集系统。该系统运用了模块化设计思路, 包括数据接收卡、数据传输卡和软件驱动三部分。简要介绍了自行研制的数据采集卡的基本原理和构成, 重点研究了在 Windows XP 系统环境下利用 DriverStudio 开发 PCIE 设备驱动程序的主要方法步骤、DMA 方式进行数据传输和事件通知的实现方法。经过上位机测试, 该系统稳定可靠, 所开发的驱动程序完全可以实现数据的高速传输。

关键词: PCI Express(PCIE); 模块化; DriverStudio; 直接存储器访问(DMA); 驱动

DOI: 10.3778/j.issn.1002-8331.2009.31.020 **文章编号:** 1002-8331(2009)31-0063-03 **文献标识码:** A **中图分类号:** TP31

1 引言

20 世纪末, PCI 总线技术应用至今已 10 多年, 其间没有根本性的发展, 而处理器、内存等设备却在不断地更新, 高速数据采集和实时信息处理领域的飞速发展也使 PCI 总线逐渐成为整个系统的瓶颈^[1]。

PCIE (PCI Express) 又称为第三代 I/O 互连技术(3GIO)。2001 年初, 由 Intel 公司最先提出采用新一代的技术取代 PCI 总线和多种芯片的内部连接。2001 年底, 包括 Intel、AMD、Dell、IBM 在内的 20 多家业界主导公司开始起草新技术的规范, 并在 2002 年完成, 对其正式命名为 PCI Express。它采用与全双工通信技术类似的双通道传输模式, 具有速度快, 并且点对点串行传输, 使两端设备可以独享带宽, 扩展灵活方便等优点。不但可被用在台式机、笔记本电脑以及服务器平台上, 甚至会延伸到网络设备的内部连接设计中^[4]。

PCIE 体系结构如图 1 所示, PCIE 规范定义了四层结构^[2]:

物理层是模型的最底层, 定义了 PCIE 的电器特性, 负责接口和设备间的连接。主要用于多种传输模式和 8b/10b 编码。

数据链路层作用是保证 PCIE 总线传输数据包的正确性和可靠性, 将冗余校验码采用某种算法协议技术添加到数据中进

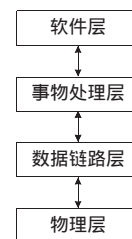


图 1 PCIE 体系结构

行校验。

处理层的主要工作是接收来自软件层的读写请求并构造发送到数据链路层的请求数据包, 此外还可能要接收来自数据链路层的响应数据包, 并还原为原来的软件请求。

软件层是分层结构最高层, 保持和 PCI 总线兼容。也就是说, PCIE 的软件在 PCI 平台中执行不必改变。

2 数据采集卡基本结构与工作原理

采集卡用基模式的 Camera Link 接口接收外来数据, 经过专用芯片转换把数据传递给 FPGA。FPGA 是整个板卡的核心, 采用 ALTERA 公司的 cyclone 系列芯片, 主要实现 PEX8311

基金项目: 长春光机所研究生种子基金(the Graduate Starting Seed Fund of Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences)。

作者简介: 梁国龙(1983-) 男, 博士生, 主要研究领域数字图像处理; 何昕, 男, 研究员, 博士生导师, 主要研究领域为数字图像处理。

收稿日期: 2008-09-12 修回日期: 2008-10-08

芯片的工作逻辑时序, 和内建 FIFO 实现对数据流的逻辑控制传输。采用 PLX 公司的 PEX8311 芯片来实现 PCIE 总线协议^[5], 该芯片前端口是本地总线, 后端口是 PCIE 总线, 可方便高效地实现协议转换。EEPROM 用于给 PEX8311 寄存器配置。系统的基本结构框图如图 2 所示。

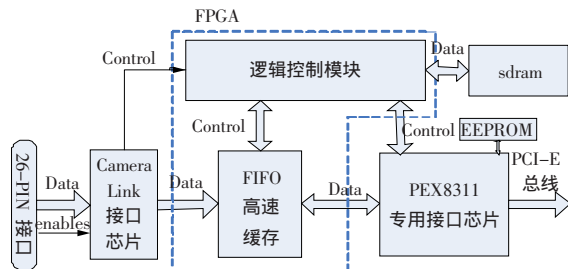


图2 数据采集卡的基本结构

由于接收的数据会有乱码或丢数据的情况, 所以加一片 SDRAM 进行数据的校验和缓存, 确保数据正确稳定传送。从数据采集卡输出的数据要进入 PC 机, 进行下一步的处理、存储或者显示, 即进行采集卡 and 上位机的通信。因此驱动程序的设计势在必行。

3 PCIE 协议总线设备驱动分析

虽然 PCIE 采用了比 PCI 总线更加先进的串行数据传输, 设备与设备之间通过交换器进行互连, 但是 PCIE 总线是在 PCI 基础上发展起来的, 沿用了现有的 PCI 总线的编程概念及通讯标准, 在软件上却是向后兼容的。并且 PCIE 设备的存储空间, IO 空间和配置空间与 PCI 设备的地址空间相同, 地址空间模型没有改变。因此, 可以依据 PCI 设备的驱动程序模型进行 PCIE 设备驱动程序的开发^[6]。

4 采集卡驱动程序实现及代码分析

4.1 驱动开发工具的选择

Jungo 公司开发的 WinDriver 难度低, 不需要熟悉操作系统的内核知识和牵涉到很底层的东西, 整个驱动程序中的所有函数都工作在用户态, 可以不用添加任何函数实现驱动, 但它可塑性差, 并且灵活性低。

由 Microsoft 公司开发的 DDK(Device Development Kit)底层代码编写工具, 代码简单, 结构清晰, 效率高, 但是大部分都是通用的基础性的工作, 而且需要对整个体系结构有很好的理解和把握, 开发难度相对较大。

DriverStudio 由 NuMega 公司开发, 以面向对象的方式, 将驱动程序编写所需的与内核访问及对硬件的访问封装成类, 加上设计的驱动程序代码生成向导 DriverWizard 以及 DriverAgent、VtoolsD、DriverWorks、SoftICE 等多款功能强大的工具, 极大地简化了驱动程序开发的难度, 减少了工作量, 而且灵活性也较好。该文选择 Driverstudio 进行数据采集卡的驱动开发^[7]。

4.2 设备驱动程序开发

利用 DriverWorks 的向导生成的框架不必一一介绍, 下面主要实现驱动程序相应的功能和进行代码分析。

4.2.1 驱动程序添加资源和初始化

驱动向导完成之后, 驱动的基本框架也就形成了。可以按照要求修改或添加相关的源代码。驱动 PCIE_LINTDevice.h

文件中可以看到添加的资源 and 例程的声明:

```
class PCIE_LINTDevice : public KPNPDevice
{
    PCIE_LINTDevice(PDEVICE_OBJECT Pdo, ULONG Unit) //构造函数
~PCIE_LINTDevice() //析构函数
    ... (省略)
    VOID StartDMA(ULONG PAddress, ULONG NBytes);
    DEVMEMBER_DISPATCHERS
    DEVMEMBER_DMAREADY(PCIE_LINTDevice OnDmaReady);
    MEMBER_ISR(PCIE_LINTDevice Isr_Irq);
    MEMBER_DPC(PCIE_LINTDevice DpcFor_Irq);
    void SerialRead(KIrp Irp); //DMA 读
    void SerialWrite(KIrp Irp); //DMA 写
    NTSTATUS PCIE_LINT_IOCTL_800_READ_Handler(KIrp Irp);
    ... (省略)
    BOOLEAN Isr_Irq(); //中断
    VOID DpcFor_Irq(PVOID Arg1, PVOID Arg2) //中断处理
    //定义类成员变量
    KMemoryRange m_MemoryRange0 //用于芯片寄存器内存映射
    KMemoryRange m_MemoryRange1 //用于芯片寄存器 I/O 映射
    KIoRange m_IoPortRange0; //用于设备卡 I/O 映射
    KDmaAdapter m_Dma; //DMA 传输
    KEvent* m_pEvent; //事件通知
    ... (省略)
}
```

在 PCIE_LINTDevice.cpp 的启动例程中的初始化:

```
NTSTATUS PCIE_LINTDevice::OnStartDevice(KIrp Irp)
{
    m_Dma.Initialize(&dd, m_Lower.TopOfStack(), 0) //初始化 DMA
    m_Buffer.Initialize(&m_Dma, 32768); //开辟 32k 公用缓冲区,
    用于 DMA 传输
    status = m_MemoryRange0.Initialize(pResListTranslated, pResListRaw, PciConfig.BaseAddressIndexToOrdinal(0)); //初始化 m_MemoryRange0
    status = m_MemoryRange1.Initialize(pResListTranslated, pResListRaw, PciConfig.BaseAddressIndexToOrdinal(2)); //初始化 m_MemoryRange1
    status = m_IoPortRange0.Initialize(pResListTranslated, pResListRaw, PciConfig.BaseAddressIndexToOrdinal(1)); //初始化 m_IoPortRange0
    status = m_Irq.InitializeAndConnect(pResListTranslated, LinkTo(Isr_Irq), this); //初始化中断对象并连接中断
    m_DpcFor_Irq.Setup(LinkTo(DpcFor_Irq), this) //初始化 DPC
}
```

4.2.2 简单的参数配置和 I/O 读写

在应用程序中调用和驱动程序连接的设备 IO 接口函数 DeviceIoControl, 用控制命令 PCIE_LINT_IOCTL_800_READ 调用驱动程序。驱动程序执行 PCIE_LINT_IOCTL_800_READ 的处理:

```
NTSTATUS PCIE_LINTDevice::PCIE_LINT_IOCTL_800_READ_Handler(KIrp Irp)
{
    ... (省略)
}
```

```

CHAR    offset=0x68; //中断状态寄存器地址
ULONG   dwTotalSize=I.ReadSize(CURRENT); //读出大小
ULONG   dwBytesRead = 0;
m_IoPortRange0.ind (offset, (PUCHAR)I.IoctlBuffer (),
dwTotalSize);
//从 0x68 寄存器里读出配置数据给应用程序
...(省略)
}

```

4.2.3 DMA 编程传输

为了实现大容量快速的数据传输,就要启动 DMA,在驱动程序里 DMA 整体流程如图 3 所示。当运行驱动程序时首先进行初始化,当应用程序 ReadFile 或者 WriteFile 发出读写命令后,驱动调用 SerialRead/SerailWrite 函数,再调用 OnDMAReady()函数,在 OnDMAReady()函数中启动 DMA 传输即 StartDMA,传输完后发生中断,在中断里调用 Continue()再次启动下次传输,直到全部数据完成^[6]。

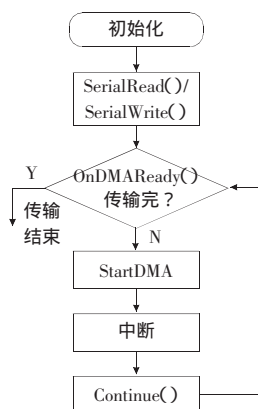


图 3 DMA 流程

```

VOID PCIE_LINTDevice : SerialRead(KIrp I)
{
    ... (省略)
    m_CurrentTransfer =
    new(NonPagedPool) KDmaTransfer(this &m_Dma);
    status=m_CurrentTransfer->Initiate(this &m_Dma I.Mdl(),
    (I.MajorFunction ()==IRP_MJ_READ)? FromDeviceToMemory :
    FromMemoryToDevice LinkTo(OnDmaReady) &m_Buffer); //该操作将
    使系统调用 OnDmaReady 例程
}

VOID PCIE_LINTDevice : OnDmaReady (KDmaTransfer *pXfer,
KIrp I)
{
    //判断 DMA 是否传输完
    ... (省略)
    if (pXfer->BytesRemaining() == 0)
    pXfer->Terminate();
    ... (省略)
    if (pXfer->BytesRemaining())
    StartDMA(PTD->td_PhysAddr.LowPart PTD->td_Length);
    /* 如果 BytesRemaining 缓冲区所存储的数据不为空,就再
    次调用 StartDMA 继续传输,
    反之没有数据,就调用 Terminate 结束本次传输。*/
}

VOID PCIE_LINTDevice : StartDMA (ULONG PAddress,ULONG

```

```

NBytes))//执行 DMA 传输
{
    ... (省略)
    /* 设置 DMA 相关寄存器,启动传输。注意这里对寄存器配置
    要谨慎,需要仔细查阅数据手册的寄存器配置,否则会导致死机或不
    工作 */
}

BOOLEAN PCIE_LINTDevice : Isr_Irq() //进入中断
{
    /* 传输完后发生中断,判断是否是 DMA 中断,如果是,就要
    进行必要的寄存器设置并且请求中断延迟调用 */
    if (! m_DpcFor_Irq.Request(NULL, NULL)) { ... } //调用中断
    延迟例程
}

VOID PCIE_LINTDevice : DpcFor_Irq (PVOID Arg1, PVOID
Arg2) //中断延迟例程
{
    m_CurrentTransfer->Continue(UseTransferSize);
    //调用 Continue 函数,该操作将使系统再次调用 OnDmaReady
    例程
}

```

4.2.4 事件通知

当驱动程序捕获到特点事件(如本地中断)发生时,应该可以主动和应用程序进行通信。这主要有三种方法,DeviceIoControl 异步完成、共享命名事件、事件通知。这里只介绍事件通知方法。该文章是以本地中断触发事件来进行实验的。事件通知基本流程是应用程序中用 DeviceIoControl 传一个句柄给驱动程序,然后创建一个线程等待^[6]。

```

HANDLE hEvent=*(HANDLE*)I.IoctlBuffer(); //应用程序创
建的事件句柄

```

```

m_pEvent=new(NonPagedPool) KEvent(hEvent, OBJECT_
TYPE_ALL_ACCESS); //构造系统事件函数,等待事件发生

```

一旦有本地中断产生,调用中断延迟例程,在 Dpc 中设置事件发生通知应用程序:

```

m_pEvent->Set() //事件发生,通知应用程序

```

然后在应用程序中进行相关的处理即可。注意:

(1)INTCSR 寄存器要开启本地中断输入使能位和 PCIE 中断使能。

(2)当 DMA 中断和本地中断同时开启时,容易产生冲突,导致死机,所以当响应某一中断时,应该屏蔽其他中断,用到时再开启该中断。

5 结论

通过自行研制的 PCIe 图像采集系统阐述了 WDM 驱动的开发,经测试在 WindowsXP 下开发的驱动运行稳定,数据传输率达到了比较好的效果,在数据链路 x1 通道的条件下用 DMA 模式传输,数据率可以达到 110 MB/s,暂时完全可以满足项目要求。如果用 x4 或者以上的通道进行传输将会有更大的传输速度,具有很大的扩展空间。

参考文献:

- [1] 田玉敏,王崧,张波.PCI Express 系统体系结构标准教材[M].北京:电子工业出版社,2005.
- [2] PCI Express Card Electromechanical Specification Revision 1.0A[S]. 2003.

的诸多内容,主要包括:空间曲面拟合插值技术、空间曲面网格剖分技术、空间曲面拓扑关系描述方法,以及三维地质网格剖分技术等。通过地震资料构造解释和层序解释,得到以离散数据点表达的地质界面和断层面。这些地质界面和断层面在三维图形中就是空间曲面。结构建模的首要任务就是空间曲面的离散数据点进行拟合和插值,然后进行空间曲面的网格剖分^[10-17]。

区域地震速度模型分析与应用系统借助了 MapGIS-TDE 平台提供的多源数据耦合三维建模技术方法,研究适合城市地质空间数据多源、异构的三维地质建模技术方法。采用建模数据及建模算法交互式选择、利用自适应三维地质建模方法以及模型重构的反馈机制及修正技术,支持基于地表高程数据、钻孔数据、地质剖面、平面地质图等多源数据耦合三维地质结构建模。

4 系统实现

系统包括多种数据类型的数据建设、软件开发、系统维护等多项工作,是一项工程量大、复杂的信息系统工程,需要运用系统工程、软件工程方法和全面质量管理体系,统筹安排系统的建设工作。针对系统的需求,系统提供了 4 个不同的子系统模块,系统选用 Microsoft Windows 2000/XP 系列操作系统、国产优秀地理信息系统 MapGIS 7.0 基础平台。MapGIS 7.0 平台具有很好兼容 Windows 系列操作系统环境的特性,能进行空间数据的建模、管理、绘制及分析。并且已在地质行业得到广泛的应用,对专题来说较为合适的 GIS 平台。数据库则采用 Microsoft SQL Server,采用标准 API 接口与组件设计及实现此系统。

系统总体用户界面采用地质图或特色地理图片为背景。主界面数据图层显示,采用小比例尺的地质图或工作程度图数据作为基本支撑。系统运行主界面如图 4 所示。

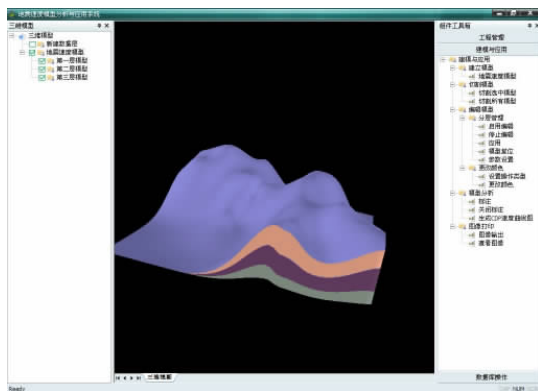


图 4 系统运行主界面

5 结语

该系统是区域地震速度数据管理和分析的有效工具。分析

人员通过该系统能够更加有效地对地震数据进行管理,通过简单的界面操作还可以对由地震数据生成的三维模型进行可视化的编辑,并随时导出兴趣数据以便进一步分析研究。系统下一步的工作将在此基础上进行扩展,研究实现地震三维模型在结构模型和属性模型上的同时切割功能,从而满足用户在专业应用方面的更广泛的需求。

参考文献:

- [1] Piazza J L, Sandjiv L, Legeron S, 等.用地质统计学改善地震速度:实例研究[C]//美国勘探地球物理学家学会第 67 届年会论文集, 1997: 253-256.
- [2] 潘宏勋, 方伍宝.地震速度分析方法综述[J].勘探地球物理进展, 2006, 29(5): 305-311.
- [3] 程红杰, 胡祥云, 田米玛, 等.地震数据网格化方法研究[J].工程地球物理学报, 2006, 3(1): 28-32.
- [4] 潘宏勋, 孙开峰, 叶勇.地震速度分析技术新进展[J].勘探地球物理进展, 2008, 31(3): 172-178.
- [5] 郑贵洲, 申永利.地质特征三维分析及三维地质模拟现状研究[J].地球科学进展, 2004, 19(2): 218-223.
- [6] 朱良峰, 潘信, 吴信才.三维地质建模及可视化系统的设计与开发[J].岩土力学, 2006, 27(5): 828-832.
- [7] 王有新, 王延光.地震数据处理中的速度换算关系[J].油气地球物理, 2005, 3(3): 1-6.
- [8] 王帮进, 张宏, 蒋文明, 等.基于插件的房产 GIS 集成研究[J].微机信息, 2007, 23(2-1): 210-212.
- [9] 魏嘉.地质建模技术[J].勘探地球物理进展, 2007, 30(1): 1-6.
- [10] 周良辰, 陈锁忠, 朱莹.地质结构三维建模及其可视化方法研究[J].计算机应用研究, 2007, 24(6): 150-164.
- [11] 张二华, 高林, 马仁安, 等.三维地震数据可视化原理及方法[J].CT理论与应用研究, 2007, 16(3): 20-28.
- [12] 徐能雄, 田红, 于沐.适于岩体结构三维建模的非连续地层界面整体重构[J].岩土工程学报, 2007, 29(9): 1361-1366.
- [13] 姚凌青, 潘懋, 屈红刚, 等.Kriging 算法在含水量三维属性模型构建中的应用研究[J].计算机应用研究, 2008, 25(8): 2554-2560.
- [14] 王俊虎, 杨锋杰.3D Surfer 在地质体属性建模及可视化中的应用[J].金属矿山, 2008(5): 117-119.
- [15] 吴焱, 马殿富, 华刚.基于体数据的三维地质属性体建模[J].计算机工程与应用, 2003, 39(4): 108-111.
- [16] 马仁安, 杨静宇, 王洪元.可视化技术及在三维地震解释中的应用[J].计算机工程, 2003, 29(5): 139-141.
- [17] 于军, 王晓梅, 吴加各, 等.利用三维地震勘探资料在 GMS 平台下实现地层实体建模的方法研究[J].水文地质工程地质, 2006, 33(1): 8-10.

(上接 65 页)

- [3] 孟会, 刘雪峰.PCI Express 总线技术分析[J].计算机工程, 2006(23): 253-258.
- [4] 许军, 李玉山, 贺占庄, 等.PCI-Express 总线技术研究[J].计算机工程与科学, 2006(5): 141-143.

- [5] Expresslane PEX8311AA PCI Express -to -generic local bus bridge data book[M/OL].http://www.plxtech.com.
- [6] 武安河.Windows 2000/XP WDM 设备驱动程序开发[M].2 版.北京: 电子工业出版社, 2005.
- [7] Oney W.Programming the Microsoft Windows driver model[M].马少华, 译.[S.L.] Microsoft Press, 2003.