

基于 DSP 和 PCI 的通用数据采集及处理卡实现

陈 斌, 王成华, 夏永君

(南京航空航天大学 信息科学与技术学院, 江苏 南京 210016)

摘 要: 提出及实现了一个基于 PCI 局部总线的通用数据采集及处理卡。采用 TMS320VC5409 作为外围核心处理单元、PCI2040 作为 PCI 桥芯片, 通过软件开发实现了三相交流电频相测试仪。着重介绍了 PCI 卡的物理设计和基于 Windows 2000 的 WDM 驱动程序设计。

关键词: DSP; PCI 总线; WDM 驱动程序; 通用数据采集与处理卡

中图法分类号: TP336 文献标识码: A 文章编号: 1001-3695(2005)01-0171-03

Universal Card Realization of Data Acquisition and Processing Based on DSP and PCI

CHEN Bin, WANG Cheng-hua, XIA Yong-jun

(School of Information Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing Jiangsu 210016, China)

Abstract: Presented and realized an universal card of data acquisition and processing based on PCI local bus. In this card, applied TMS320VC5409 as a peripheral processing unit and applied PCI2040 as PCI-DSP bridge, then performed a frequency and phase testing instrument of three-phase AC power by software design. Furthermore, this paper focused on the board design of the PCI card and the WDM program application based on Windows 2000 OS.

Key words: DSP; PCI Bus; WDM; Universal Card of Data Acquisition and Processing

在数据采集与处理领域, 不论达到何种目的、采用何种算法, 通用的数据采集处理卡都有着几乎一致的流程, 因此本文提出及实现了面向中低数据速率和较复杂算法的数据采集处理卡, 并以之为通用处理平台实现了三相交流电频相测试仪。

在该设计中, A/D 采集及 D/A 输出数据先被 PCI 插卡上的 DSP 处理, 面向用户的数据则通过 PCI 总线交互传输。由于该设计较少占用主机资源, 实时响应速度快, 具有灵活的可编程性和模块化性, 因此在工业应用上有着广泛的用途。

中进行处理, 将采集所得数据处理后送往 PC 或将 PC 送来的数据处理后对外输出。CPLD 工作于状态机方式作为系统的全局控制器。

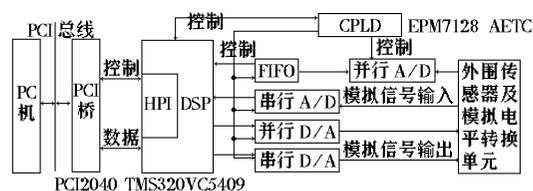


图 1 系统总体方案

1 总体方案

如图 1 所示, 数据采集和处理部分都集成在一块 PCI 插卡上, 采用 TI 公司的 PCI 专用接口芯片 PCI2040 实现 PCI 总线协议, 以半双工模式完成主机与 DSP 之间的数据交换。考虑到数据采集卡的通用性与设计灵活性, A/D 采用了并行的六通道 16 位 ADS8364 与串行的 ADS8361; D/A 采用了并行的 DAC8541 与串行的 DAC8531; DSP 选用了 TI 公司性价比较高的 TMS320VC5409; CPLD 选用 ALTERA 公司的低功耗 EPM7128AETC。这些外围器件电平彼此兼容, 数字部分均可采用 3.3V 供电, 系统功耗较低。

并行 A/D 通过 FIFO: SN74V225 接到 DSP 数据线; 并行 D/A 则直接与 DSP 数据线相连; 而串行器件则与 DSP 的 MCB-SP 连接, 通过将它们映射到不同的 I/O 空间而由 CPLD 进行逻辑选择。DSP 作为 PC 机与外界之间的协处理器对数据集

2 PCI 卡设计及实现

PCI 局部总线是微机中处理器/存储器与外围控制部件、扩展卡之间的互连接口, 由于它独立于处理器的独特设计及其在高性能、低成本、开放性等方面的优势, 使其得到迅速普及和发展, 目前已成为微机事实上的总线标准, 并在嵌入式计算机和工业控制计算机方面具有广泛的应用前景。其最新协议为 PCI2.3 规范, 在操作频率 66MHz 下, 64 位地址/数据同步总线可以提供高达 528MBps 的峰值吞吐率。

由于 PCI 总线协议的复杂性, 一般开发 PCI 接口卡都使用现成的 PCI 接口芯片, 常用的专用接口芯片如 S5933, PLX9052, PLX9054 等应用比较成熟, 但是不便于与 DSP 相连。因此选用 TI 公司推出的支持 PCI2.2 规范的 PCI 桥控制器 PCI2040 与 TI 的 DSP 进行无缝连接, 由于无须进行任何电平或逻辑转换, 不但加快了开发速度, 且提高了产品稳定性。

PCI2040 与 DSP 及 PCI 插槽金手指之间的连接比较简单, 需要注意的是 PCI 卡的电源设计与布局布线要求。

(1) 电源设计: 每个 Vcc 引脚上必须有退耦电容, 且容量的平均值至少为 0.01uF 以保证对地去耦; 从电源金手指根部到电容器焊盘的走线长度不大于 250mil, 线宽至少为 20mil, 以发挥有效的交流参考作用; 为了尽量降低 PCI 卡功耗, 应选用低电压、低功耗器件, 暂时不工作的器件则应处于节电模式。

(2) 布局布线要求: 所有 32 位接口信号的最大走线长度为 1.5inch, 以得到尽量短的传输延迟时间; 卡上时钟 CLK 信号走线长度为(2.5 ±0.1) inch, 而且只能连到一个负载上;

与金手指相连的 PCI 元件其引脚电容必须小于 10pF 以保证信号完整性。

只有满足上述的电源要求、布线长度、负载限制等约束条件, 才能保证系统信号的完整性, 以满足 PCI 局部总线的电气规范。

3 基于 WDM 的 PCI 卡驱动程序设计

3.1 WDM 驱动程序结构及原理

WDM 作为新一代的驱动程序构架, 是一个跨平台的驱动程序模型。由于它在 Windows 98 以上的操作系统中实现了源代码级兼容, 仅需要重新编译就可以运行于其他平台上, 因此为驱动程序开发人员提供了极大的方便。

WDM 驱动程序是分层的, 在不同层上的驱动程序有不同的优先级; 同时, 它还提供了一个 128 位的全局唯一标识符 GUID 来识别驱动程序; 另外, WDM 还引入了功能设备对象 FDO 与物理设备对象 PDO 两个新概念来描述硬件。开发驱动, 实际上就是开发 FDO。

3.2 应用程序、驱动程序与 DSP 之间的通信

在 Windows 2000 操作系统下, 驱动程序主要由 Windows 2000 DDK 与 VC++6.0 设计, DriverStudio 2.7 对 DDK 进行封装后, 利用向导可生成驱动程序框架, 大大提高开发速度。

当驱动程序被载入时, 内核调用初始化入口点 DriverEntry 例程。在用户插入新设备时, PnP 管理器采用即插即用模块 AddDevice 例程用以创建 WDM 设备对象。在 PnP 启动例程 OnStartDevice 中, 驱动程序初始化 DS 提供的 KmemoryRange 与 KIoRange 类实例, 顺序分配基地址寄存器分别对应于 PCI2040 操作寄存器的内存映射地址与 I/O 映射地址, 实现 PCI 设备卡配置空间的分配, 以达到系统识别及分配资源的目的。

在 Windows 2000 中, 应用程序实现与 WDM 通信的过程是: 应用程序先用 CreateFile 函数通过 GUID 接口或符号链接名打开设备, 然后一般用 DeviceIoControl 通过应用程序数据缓冲区和 WDM 进行双向通信, 包括从 WDM 中读取数据及输出数据给 WDM 两种情况, 也可以用 ReadFile 从 WDM 中读数据或用 WriteFile 写数据给 WDM。当应用程序退出时, 用 CloseHandle 关闭设备。在 Windows 2000 以前的 VXD 驱动程序中, 驱动程序可以直接给应用程序发消息、传递内存而得到快速响应, 但这样容易带来系统工作的不稳定, Win98 中常出现的

蓝屏就是该操作的后果。Win2000 中, 为了提高系统稳定性, 驱动程序与应用程序的通信被 DeviceIoControl 异步完成和 Win32 事件通知两种方法所代替。常用 Win32 事件通知: 应用程序首先创建一个事件, 将该事件句柄通过 DeviceIoControl 传递给 WDM, 然后等待 WDM 发送事件消息; WDM 通过类 KEvent 获取该事件的对象指针后, 就可以在中断优先级低于线程调度级的例程中设置事件信号态通过 WaitForSingleObject 进入应用线程以触发应用程序, 最终达到双向通信。

PCI2040 作为 PCI 总线与 DSP 之间的接口芯片, 使得 WDM 与 DSP 双方的操作都面向于 PCI2040。PCI2040 对 WDM 提供了内存映射的 HPI 控制与状态寄存器 (CSR)、面向 DSP 的 HPI 寄存器以及面向通用 (GP) 器件并与 HPI 口引脚复用的 GP 总线寄存器, 其对应的管脚信息如图 2 所示。PCI2040 的 CSR 起状态控制作用, 其中标志位的设置会影响它面向 DSP 引脚的工作状态。C5409 提供了增强型的 HPI8 接口, 微机作为主方 (Host) 可通过 HPIA (地址)、HPIC (控制)、HPID0 (数据方式 0) 和 HPID1 (数据方式 1) 四个 HPI 寄存器实现对 DSP 片内空间的完全访问; 相比较而言, DSP 作为 HPI 口的从方, 只能先将送往主机的数据写到片内空间, 然后对 PCI2040 发 HINT 中断以实现被动的数据交换。

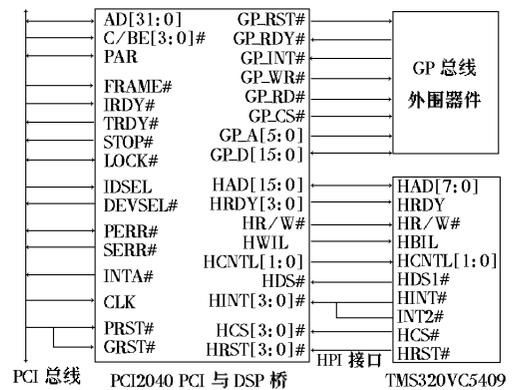


图 2 PCI2040 接口图

3.3 DSP 程序上载

C5409 的在片 ROM 内含有 DSP 系统的自举加载 Boot-Loader 程序, 系统上电后按照 BootLoader 可将用户代码以多种方式送入 DSP 片内高速程序空间执行。本系统是基于 PC 机的 DSP 系统, 因此选用主机下载 HPI 引导模式。

通过 CCS 的 Simulator 将编译好的可执行程序 .out 文件加载进 DSP 的程序空间, 参照 .map 文件将该程序空间以 .dat 文件方式读出, 去除多余报头后即得到 TI 格式的程序代码, 最后以内存映射方式调用 KmemoryRange 类的成员函数将代码写入 DSP 程序空间。其流程如图 3 所示。

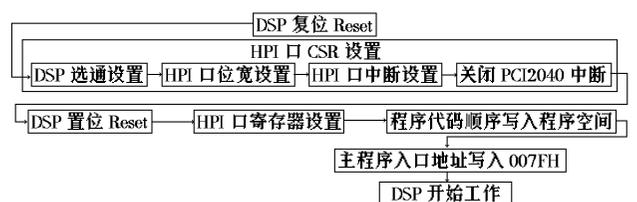


图 3 DSP 的 HPI 口加载方式

3.4 中断处理

对于 WDM 的中断, 在 DS 中提供了 KInterrupt 类实现硬件中断的处理, 其成员函数包括中断初始化, 将一个中断服务例程连接到一个中断和解除其连接等。由于中断服务例程 Isr_Irq 运行于硬件中断级, 很多函数都不能调用, 为此操作系统要求中断服务例程的处理时间尽可能短, 于是在内核中就引入了延迟过程调用 DpcFor_Irq 即 KdeferredCall 类完成大部分的中断处理工作。

对于 DSP 系统, 其核心程序是一个个的中断服务处理子程序, 包括 HPI 中断、定时/计数器中断、DMA 中断、MCBSP 数据接收和发送中断、用户自定义中断等, 其中断入口在中断向量表中得到对应。对中断处理程序的编程实现了一个完整的 DSP 应用系统。

4 应用及结论

通过以上的设计, 搭建了一个初步的 DSP&PCI 硬件应用平台, 并在该平台的基础上实现了三相交流电频相测试仪, 其中 DSP 主要的中断处理流程如图 4 所示。



图 4 DSP 主要中断处理流程

由于 A/D 转换精度较高且数字信号处理部分的算法较成熟, 系统检测精度主要取决于电压互感器、电流互感器和电平转换单元的精度。以高精度谐波发生器作为信号源, 检测后所得的精度如表 1 所示。

在该平台的基础上, 通过开发功能各异的 MFC 应用程序、WDM 驱动程序、DSP 应用程序和 CPLD 的逻辑控制程序, 可实现丰富多彩的功能。在将通用功能建成函数库的基础上, 又较

方便地开发了中速率数据压缩、语音信号压缩存储及回放等新功能, 适应了目前系统集成开发上的通用性、便捷性和可复用性, 突出了系统集成中软件开发的重要性。

表 1 所得的精度

部分检测项目	检测精度
各相电压、电流的基波有效值	> 0.01%
各相电压、电流的 30 次以下谐波有效值	> 0.05%
各相电压、电流的基波频率及相位	> 0.05%
各相电压、电流的 15 次以下谐波频率及相位	> 0.5%

参考文献:

[1] CI Special Interest Group. PCI Local Bus Specification [M]. 1999.

[2] Texas Instruments. TMS320C54X DSP Reference Set volume 1 ~volume 5 [M]. 1999.

[3] Texas Instruments. PCI2040 PCI-DSP Bridge Controller Data Manual [M]. 1999.

[4] Microsoft 公司. Microsoft Windows 2000 驱动程序设计指南 [M]. 北京: 北京大学出版社, 2000.

[5] 武安河, 郇铭, 于洪涛. Windows 2000/XP WDM 设备驱动程序开发 [M]. 北京: 电子工业出版社, 2003.

[6] 陈光日, 裴继红. 基于 DSP + PCI 结构的高速数据采集系统 [J]. 雷达与对抗, 2002, (4): 34-38.

[7] 肖伟, 吕扬生, 印震峰. DSP 应用系统的 PCI 接口设计 [J]. 医疗卫生装备, 2003, (1): 19-21.

[8] 李平, 张云麟. Windows 2000 内核模式驱动程序设计 [J]. 计算机应用研究, 2003, 20(1): 122-124.

[9] 陈君, 赵伟, 初仁欣. 测算电网频率和谐波的新方法 [J]. 清华大学学报 (自然科学版), 2000, 40(1): 25-27.

作者简介:

陈斌(1979-), 男, 四川开江人, 硕士, 主要研究方向为数字系统设计与计算机应用; 王成华(1963-), 男, 江苏扬中人, 教授, 主要研究方向为电路与系统、数字系统设计与计算机应用; 夏永君(1979-), 男, 江苏苏州人, 主要研究方向为电路与系统、数字系统设计与计算机应用。

(上接第 157 页)

3 结论

本文提出的方法能将应用程序开发工具使用于操作系统内核的开发中, 操作方法完全与应用程序开发相同。对于许多嵌入式系统内核, 可在本文构建的环境中得到完全调试。本文所讨论的技术可应用于不同类型系统内核的开发。该环境被用于一个通用内核的开发工作中, 调试了绝大部分代码, 包括进程/线程控制、文件系统上层、部分虚存代码、全部物理内存管理代码、内核内存管理代码, 并在裸机上验证通过, 调试代码量占内核总代码量的 93%。实践证明, 该方法使内核开发效率得到了极大的提高。

参考文献:

[1] Peter Kacsuk, et al. A Graphical Development and Debugging Environment for Parallel Programs [J]. Parallel Computing, 1997, (22): 1747-1770.

[2] 张磊, 王学慧. Linux 内核调试技术 [J]. 计算机工程, 2003, 29(10): 81-83.

[3] Kevin Lawton, et al. Bochs User Manual [EB/OL]. <http://bochs.sourceforge.net/doc/docbook/user/book1.html>, 2003-01.

[4] Greg C Lee, Jackie C Wu. Debug It: A Debugging Practicing System [J]. Computers & Education, 1999, (32): 165-179.

[5] 周明德, 等. 保护方式下的 80386 及其编程 [M]. 北京: 清华大学出版社, 1993.

[6] 毛德操, 胡希明. Linux 内核源代码情景分析 [M]. 杭州: 浙江大学出版社, 2000.

[7] Matt Pietrek. A Crash Course on the Depths of Win32™ Structured Exception Handling [EB/OL]. <http://www.microsoft.com/msj/0197/exception/exception.aspx>, 1997-01.

[8] Microsoft. SetThreadContext [EB/OL]. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/debug/base/setth-readcontext.asp>, 2003-01.

作者简介:

石怀东(1979-), 男, 浙江宁波人, 硕士研究生, 主要研究方向为实时系统、嵌入式操作系统; 杨子江(1975-), 男, 安徽安庆人, 讲师, 博士, 主要研究方向为实时系统、嵌入式操作系统等; 董金祥, 男, 浙江温州人, 教授, 博士生导师, 主要研究方向为计算机辅助设计、人工智能、工程数据库、嵌入式系统。