

# 基于 VHDL 的 PCI 总线数据采集卡的研究

林青松,王光辉

(河南科技大学 电子信息工程学院,河南 洛阳 471003)

**摘要:**提出了应用 VHDL 语言实现 PCI 数据采集卡的设计方法,对 PCI 总线主模式控制器的结构、时序和 Windows XP 环境下 PCI 设备 WDM 驱动程序的开发进行了研究;在单片 FPGA 中实现了 A/D 转换器的时序控制、FIFO 存储器和 PCI 总线接口控制器的设计。仿真结果表明,该数据采集卡符合 PCI 2.2 协议,具有 DMA 传输功能。

**关键词:**PCI 总线;现场可编程逻辑阵列(FPGA);DMA 传输;VHDL;WDM 驱动

中图分类号:TP274.2

文献标识码:A

## Research of PCI bus data acquisition card based on VHDL

LIN Qing Song, WANG Guang Hui

(Electronic Information Engineering College, Henan University of Science and Technology, Luoyang 471003, China)

**Abstract:** A new solution is introduced to design a PCI data acquisition card based on VHDL, and studies development of the PCI bus master mode controller's structure, timing and Windows XP environment's PCI device WDM driver. The design implementation of A/D converter timing control, FIFO memory and PCI bus interface controller in a single FPGA chip. The result of test indicates that this design accord with sequence of PCI 2.2 protocol and has the DMA transfer function.

**Key words:** PCI bus; FPGA; DMA transfer; VHDL; WDM driver

目前,利用个人计算机作为数据采集和处理的平台,通过数据总线将采集的数据高速地传输到计算机的存储设备中,是实现数据采集存储和处理的主要手段。最常用的个人计算机总线有 ISA 总线、USB 总线及 PCI 总线。3 种总线的特点是 ISA 总线采用程序请求 I/O 方式与 CPU 通信, CPU 资源占用大,传输速率低;USB2.0 标准的理论传输率虽然高达 480 Mb/s,但是它的 CPU 占用率较高;PCI 总线理论传输速率可达 132 Mb/s,稳定传输速率一般也可达 80 Mb/s<sup>[1]</sup>,而且可靠性高。因此 PCI 总线已经广泛应用于数据采集、测控等领域。

随着超大规模集成电路技术的发展, FPGA (现场可编程门阵列)的集成度和运行速度得到提高,并且 FPGA 易于实现逻辑控制,时钟频率高、内部时延小,可以实现乘法器、地址发生器、状态机、译码器等功能,具有比单片机和 DSP 更高的灵活性<sup>[2]</sup>。基于以上优点本文提出了一种在单片 FPGA 上实现 PCI 总线数据采集的设计方案。

### 1 PCI 数据采集系统构成

本设计中数据采集卡包括 A/D 转换单元、数据缓冲存

储单元和基于 PCI 总线的数据传输单元。被测信号经过前置预处理部分,直接进入 A/D 采样,采样后的数据经过 FIFO(先进先出存储器)缓冲,通过 PCI 总线传输到 PC 机中,最后通过计算机软件平台对采集到的数据进行信号分析和处理。整个数据采集卡的工作时序是在单片 FPGA 控制下进行的,其内部包含 A/D 和 FIFO 的时序控制模块、FIFO 存储器模块和 PCI 接口逻辑模块。其硬件设计框图如图 1 所示。

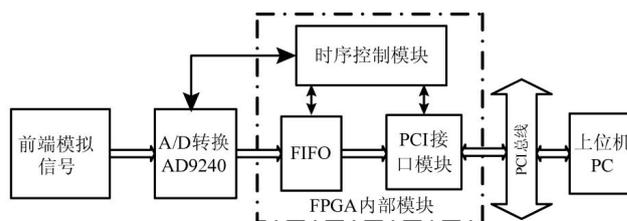


图 1 系统结构图

为了使 FPGA 能够实现上述三大功能并具有升级空间,本设计选用了 Altera 公司的 Cyclone 系列,具体型号为 EP1C6Q240C8,该 FPGA 具有较高的性价比,主要特点如下:

## 应用奇葩

Example of Application

(1) 逻辑单元 LE (Logic Elements) 数目达到了 5 980 个, 除去电源和配置引脚, 为用户提供了 185 个 I/O 引脚;

(2) 内嵌了 20 个 M4K RAM 块, 可方便地配置成 FIFO 或 RAM;

(3) 内嵌了 2 个锁相环, 有利于 FPGA 内部的时钟管理, 提高系统的稳定性;

(4) 支持 66 MHz、32 位 PCI 总线标准。

## 1.1 模数转换单元

采用 ADI 公司的 14 位、10M/s 高性能模数转换器 AD9240。该芯片自带采样保持电路 (SHA) 和输出缓冲器 (OUTPUT BUFFER)。AD9240 的时序控制与传统的 A/D 有所不同, 完全依靠时钟控制采样、转换和数据输出<sup>[3]</sup>。如图 2 所示, 在第 1 个时钟的上升沿开始采样转换, 第 4 个时钟上升沿到来时, 数据将出现在 D1~D14 端口上。数据输出采用直接二进制编码格式, 根据需要可以配置芯片的 DRVDD 引脚构成 +3.3 V 或 +5 V 逻辑系列接口, 而不需使用电平转换芯片, 简化了电路设计。硬件上, 采用系统自通电起, A/D 和时钟电路始终处于工作状态, 对数据不停进行转换, 独立的模拟和数字电源、模拟地和数字地设计, 减小数字信号对模拟信号的干扰。

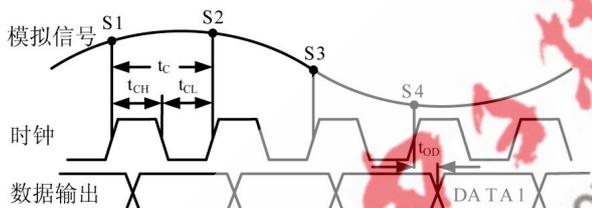


图 2 AD9240 时序图

## 1.2 数据缓冲存储单元

Cyclone 系列 FPGA 是一款高性能、低价格的可编程逻辑器件, 具有丰富的逻辑单元和存储单元。其内部的 M4K RAM 块可以配置大小不同的各种类型存储器, 如 RAM 和 FIFO, 其中 FIFO 具有两套数据线而无地址线, 可在其一端写操作而在另一端进行读操作, 数据在其中顺序移动, 从而达到很高的传输速度和效率。因此, FIFO 更适合作为 A/D 采样时数据高速写入的缓冲存储器。

在设计中采用 FPGA 来实现 FIFO 和数据传输的时序控制。使用 Quartus II 8.0 集成开发环境中的 MegaWizard Plug-In Manager 工具来构建 FIFO, 首先在向导中设置参数, 构建一个 14 位宽、512B 深的 FIFO, 并设置 FIFO 的空满标志位, 写使能、读使能等控制位, 以便实现与 A/D 转换器和 PCI 总线的逻辑接口。所构建的 FIFO 如图 3 所示。

用 FIFO 构成高速 A/D 采样缓存时, 由于转换速度较快, 直接将 ADC 采样后的数据存储到 FIFO 中, 此过程对时序配置要求非常严格, 如果两者时序关系配合不当, 就会发生数据存储出错或者掉数。针对这一问题, 把 A/D 转换时钟和 FIFO 写时钟设置为同一时钟源, 自上电起, A/D 和时钟电路一直处于工作状态, 不停地进行数据的转换, 但数据是

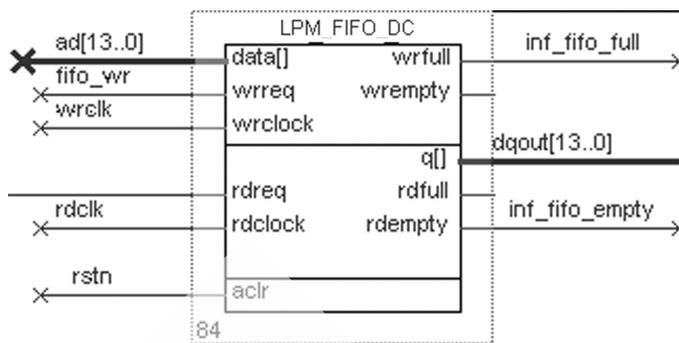


图 3 同步 FIFO

否写入到 FIFO 中, 由 FIFO 的写使能信号 (wrreq) 来决定, 当时序控制模块发出写使能信号有效时, 转换的数据才能存储到 FIFO 中。从图 2 可知, A/D 转换数据的输出和转换时钟有一定的相位差  $t_{ob}$ , 在 FPGA 内部可通过延时或时钟管理器来满足建立时间和保持时间, 保证数据不失码地传输到 FIFO 中。

## 1.3 PCI 总线接口模块

目前 PCI 接口的设计有两种方法: 一种是采用现成的 PCI 总线桥接器件; 另一种方法是采用大规模可编程逻辑器件, 通过软件编程完成硬件设计<sup>[4]</sup>。考虑到使用 PCI 接口专用芯片, 如 AMCC S5933、PLX9054 等, 仍需要外部扩展 FPGA 来进行 IO 接口处理, 这种方法成本较高且占用 PCB 板面积较大。于是设计中用 VHDL (硬件描述语言) 对 FPGA 编程开发 PCI 接口逻辑, 使单片 FPGA 既包含用户逻辑又包含接口逻辑, 从而使电路大大简化, 使设计更加紧凑。其次, 当系统升级时, 只需对可编程器件重新进行逻辑设计, 而无需更新 PCB 版图。

数据采集卡中设计的 PCI 总线接口电路具有 32 位总线宽度, 工作在 33 MHz, 具有 Target/Initiator 模式, 在 Initiator 模式下可以进行 DMA 传输, 主要包含地址命令锁存、地址命令译码、奇偶校验、数据通路、配置空间和有限状态机六部分。其中 Initiator 模块扮演着总线控制者的角色, 它的核心是主模式状态机。主模式状态机处理 DMA 传输时总线上的各个状态, 控制总线的时序, 因此主模式状态机的设计是 PCI 接口模块中的设计重点, 下面将主要介绍设计中 PCI 主模式状态机的设计。

如图 4 所示主模式状态机由 IDLE、REQ、ADDR、WAIT、READ、WRITE、LAST、RETRY、ABORT9 种状态组成。

IDLE: 总线空闲状态。当未发起 DMA 传输时, 总线应停留在空闲状态, 各信号均无效。

REQ: 当接收到 start 信号, 由 IDLE 状态跳转到 REQ 状态, 将 req# 信号拉低, 申请对总线的控制权, 若得到仲裁器允许, 即 gnt# 有效, 则跳转到 ADDR 状态, 否则停留在 REQ 状态。

ADDR: 该状态使 frame# 有效, 并同时给出要访问的 Host 物理首地址和命令, 在下一个时钟给出 irdy#, 跳转到 WAIT 状态。

应用奇葩 Example of Application

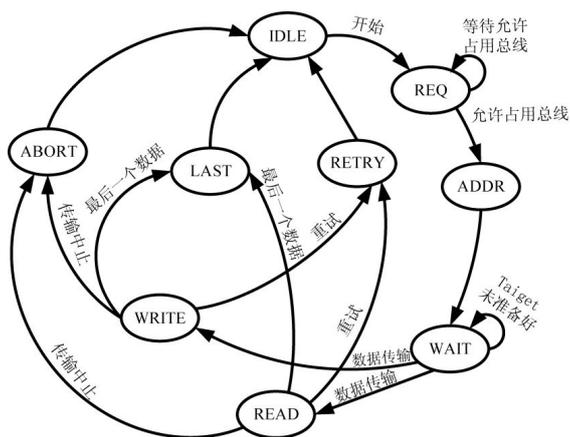


图4 主模式状态机

**WAIT:**该状态等待 Target 准备好,若 trdy# 无效则停留在 WAIT 状态;若有效则判断是读还是写,跳转到 READ 或 WRITE,并开始接收或发送数据;若 Target 在 16 个周期内未有效 trdy#,则跳转到 RETRY。

**READ:**接收 pci\_ad[31:0]上的数据,一直有效 frame# 和 irdy#,并检测当前传输是否是倒数第二次传输,若是则在下一个时钟无效 frame# 信号,并跳转到 LAST 状态;若不是则停留在 READ 状态;在此过程中,若 target 有数据的断开(disconnect with data),则跳转到 RETRY 状态;若 target 终止传输(target abort),则跳转到 ABORT 状态。

**WRITE:**将后端待发送的数据放到 pci\_ad[31:0]总线上,一直使 frame# 和 irdy# 信号处于低电平,并检测当前传输是否是倒数第二次传输,若是则在下一个时钟无效 frame# 信号,并跳转到 LAST 状态;若不是则停留在 WRITE 状态;在此过程中,若 target 有数据的断开(disconnect with data),则跳转到 RETRY 状态;若 target 终止传输(target abort),则跳转到 ABORT 状态。

**RETRY:**该状态保存当前剩余字节数、当前内存物理地址,跳转到 IDLE 状态,等待重新发起传输。

**LAST:**完成最后一次传输,在下一时钟周期将 irdy# 无效,回到 IDLE 状态。

**ABORT:**无效所有信号,回到 IDLE 状态。

需要说明的是,PCI 配置空间定义在 Target 模块中,PCI 的配置读写操作也由 Target 模块完成。系统开始工作时,首先由主机通过从模式写操作设置 DMA 寄存器,如传输字节数、内存物理首地址,最后给出 start 信号,通知 Initiator 模块申请总线控制权,在得到仲裁其允许后启动 DMA 传输。

图 5 是在 Quartus II 环境下 PCI 总线上 DMA 传输的时序仿真图,信号配合过程是:master 接收到 slave 发送的 dma\_start# 请求信号后,发送 req# 请求占用 PCI 总线,接到 gnt# 允许信号后,发送帧信号 frame#。在 frame# 有效的第一个时钟,发送读数据块的地址 pci\_ad[31:0](addr)及读命令 CMD,在随后的时钟当 irdy#、devsel#、trdy# 有效且 stop# 无效时,在字节使能信号 c/be[3:0]# 的同步下,接收读数据块

(data)。结束后,发 intr\_a# 中断且释放 PCI 总线。以上信号带“#”的表示低电平有效。

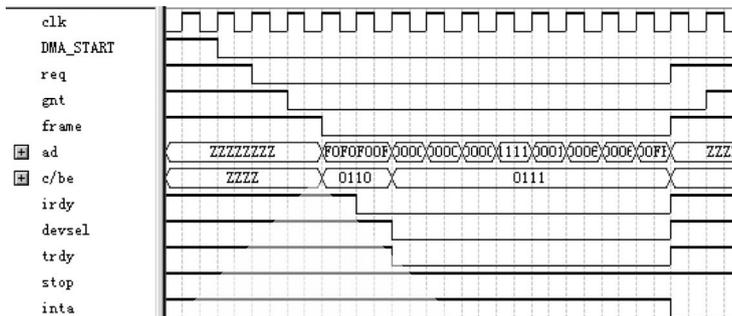


图5 DMA 传输时序仿真图

2 WDM 驱动程序的设计

WDM (Windows Driver Model) 是 Windows 32 模式驱动程序模型,这种驱动程序为 Windows 98/2000/XP 的设备驱动程序提供了统一的框架。它来源于 Windows NT 的分层 32 位色设备驱动程序模型 (Layered 32-bits Device Driver Model);支持更多的特性,如即插即用 (Plug&Play)、电源管理 (Power Management)、Windows 管理诊断 (Windows Management Instrumentation) 和 NT 事件等。

PCI 设备驱动程序除驱动程序入口例程、即插即用例程、分发例程、电源管理例程和卸载例程等基本部分外,在框架上与其他类型的设备驱动程序基本相同,是很标准的 WDM 设备驱动程序。另外,为实现 PCI 设备的系统中断等功能,还应包括中断服务 (IRP) 例程和延时过程调度 DPC (Deferred Procedure Call) 例程等其他例程。图 6 描述了 PCI 设备驱动程序的基本组成部分。

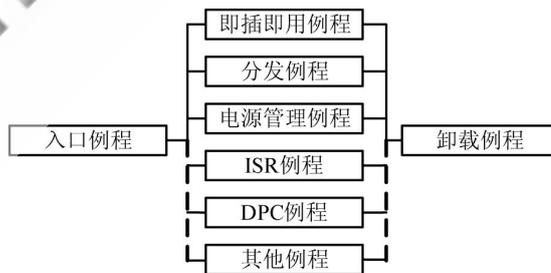


图6 PCI 驱动程序的结构

结合 WDM 驱动程序特性,在 Windows XP 操作系统下,对 PCI 数据采集卡编写驱动程序。采用 DriverStido 工具生成的 WDM 驱动程序框架,在 VC++ 6.0 中开发支持 I/O、内存和 DMA 操作的驱动程序。

在驱动程序开发中,DMA 操作的实现方法如下:首先在即插即用例程中调用 IoGetDmaAdapter 函数得到一个 DMA 适配器对象。当驱动程序的分发例程收到应用程序的 IRP\_Mj\_READ IRP 包后,调用 IoStartPacket 函数让系统启动入口例程中的 DriverStartIo 函数。在 DriverStartIo 函数中设置 DMA 寄存器,将 IRP 的 MDL (Memory Descriptor List) 所描述的内存区组装成分散/集中列表,最后启动 DMA 数

(下转第 72 页)

(上接第 68 页)

据传输,由 DMA 适配器产生数据传输周期。驱动程序收到 DMA 传输结束中断后,在 ISP 例程中首先禁止 PCI 设备的 DMA 中断,防止中断程序嵌套,然后在返回前调用 IoRequestDpc 函数请求一个 DPC。在 DPC 例程中首先清除 DMA 中断源,然后将数据从 PCI 驱动内存中复制到 IRP 的 MDL 用户缓冲区内供应用程序存储和后续处理。这样就完成了一次 DMA 操作。

随着 FPGA 技术的发展,硬件设计和软件设计的界限已经被打破。本文用 VHDL 语言实现了 PCI 总线数据采集卡的设计,在 Quartus II 中经功能仿真、时序验证,符合 PCI 总线 2.2 标准,支持 DMA 操作。本数据采集卡的各逻辑模块仅占用了 FPGA 中 1 956 个 LE,大大降低了开发成本,提高了系统集成度。同时,由于 FPGA 的体系结构和编程的灵活性,使得系统具有很强的扩展性和移植性,为将来系统功

能的改进和完善提供了便利。采用 DriveSniffer 编写驱动程序,使难度较大的 Windows 驱动开发变得容易,缩短了开发时间。

#### 参考文献

- [1] PIMG.PCI Local Bus Specification.Revision 2.2 18,1998.
- [2] 王友波,刘明业.PCI 总线接口控制器的 FPGA 实现 [J].北京理工大学学报,2004,25(5):423-426.
- [3] AD9240 Datasheet. <http://www.analog.com>. 2008, 12.
- [4] 李贵山.PCI 局部总线开发者指南[M].西安:西安电子科技大学出版社,1997.
- [5] 张惠鹃,周利华.Windows 环境下的设备驱动程序设计 [M].西安:西安电子科技大学出版社,2002.
- [6] 曹荣荣,阙沛文.PCI 数据采集卡及其 WDM 驱动程序设计 [J].计算机测量与控制,2006,14(3):415-417.

(收稿日期:2009-04-21)

## 力科推出首个 SAS-2 的物理层发送端兼容性测试方案

2009 年 11 月 2 日,在纽约州的 Chestnut Ridge 力科总部,力科作为全球串行数据测试方案的领导者,正式发布了 QualiPHY-SAS2,这是首个 SAS-2 的物理层发送端兼容性测试应用方案,SAS-2 是最新的串行连接 SCSI 标准。这个软件的发布扩展了力科自动化串行数据兼容性测试方案的组合,为 SDA8Zi 系列示波器提供了自动化控制与测量 1.5/3.0/6 Gb/s 的 SAS 信号,QualiPHY SAS-2 按照 UNH IOL 的串行 SCSI 6 Gb/s 物理层测试套件,执行所有发送端物理层测试,包括了:Group 1(发送端 OOB 信号和交流耦合要求)和 Group2(发送端展频时钟要求)、Group3(发送端不归零码数据信号要求)。

想了解更多力科技术的工程师和技术人员可以打电话到 1-800-5LeCroy (1-800-553-2769) 或者浏览 [www.lecroy.com](http://www.lecroy.com)

(力科公司供稿)