

基于 USB2.0 协议的大容量数据传输模块设计

陈阳新 曹其新 付 庄

(上海交通大学机器人研究所,上海 200030)

E-mail: brander@sjtu.edu.cn

摘 要 介绍了 CYPRESS 公司的一款 USB2.0 设备控制器的系统结构及其特点,并利用该芯片设计了基于 USB2.0 协议的大容量数据传输模块,介绍该模块的系统原理和实现方法,并简单介绍了相关软件的开发,最后通过测试程序和逻辑分析仪验证了传输速度和数据传输的正确性,并在大型喷墨打印机上得以实现。

关键词 通用串行总线 喷墨打印机 CPLD 数据传输

文章编号 1002-8331-(2004)13-0106-02 文献标识码 A 中图分类号 TP393.04

The Design of Transmission Module of Bulky Data Based on USB2.0 Protocol

Chen Yangxin Cao Qixin Fu Zhuang

(Research Institute of Robotics, Shanghai Jiaotong University, Shanghai 200030)

Abstract: This paper introduces the system structure and characteristics of a chip from Cypress company and fulfill the design of transmission module of bulky data based on usb2.0 protocol with the chip. the system principle and methods to realize this module are introduced and how to develop the software is simply explained. Finally the speed of transmission and accuracy of the system are testified through a test program and a logic analysis device and the system is realized in a large-scale ink-jet printer.

Keywords: USB ink-jetprinter CPLD data transmission

1 引言

现代高科技设备对数据传输提出越来越高的要求,不仅要求传输速度快,而且要求传输距离远。很多传统的通讯协议已经不能满足大容量高速数据传输。USB(通用串行总线)是从 97 年开始发展起来的通讯协议,目前基于 USB2.0 协议的设备已经广泛应用于个人电脑、工业设备^[1]。USB2.0 技术不仅可以实现高速数据传输,而且具有即插即用、功耗低、连接方便^[2]、价格便宜的特点。例如,一台中档速度和打印精度的打印机,180dpi,打印头运动速度是 0.6m/s,消耗数据量为 1.5MBYTES/s,采用传统的并口技术已经不能满足要求,而采用基于 USB2.0 协议的喷墨打印机,理论上数据传输速率可达 480Mb/s^[2],可以极大地提高打印机的速度同时也可以追求更高的打印质量。此外硬件设计简单,电缆连接简单,充分利用了 USB 即插即用技术,与当今个人电脑可以很好地接口。对这套系统稍做改进,可以应用于图像采集处理、家庭影院的音响系统、高速数据采集设备等需要高速大容量数据传输的设备上。

2 USB2.0 协议及芯片简介

USB2.0 协议是每 125 微秒组建一个微帧,与 USB1.1 协议的 1 毫秒相比要快 8 倍。每个微帧又包含几个处理事务。每个微帧的不同处理事务的最大包大小可以根据协议设置。USB 协议是将带宽分配给管道,消息和数据都是通过管道传输。传输类型和 USB1.1 协议一样,都是 4 种:中断、控制、同步和块传输。

Cypress 公司推出的 EZ-USB FX2 芯片同时也叫做

CY7C68013。它是业界第一个支持 USB2.0 同时向下兼容 USB1.1 规范的单片机。它既负责 USB 事务处理也兼具微处理器的控制功能,可用作 USB 外部设备的主控芯片。该芯片把 USB2.0 收发器、串行接口引擎 SIE(Serial Interface Engine),增强的 8051 内核、I²C 总线接口以及通用可编程接口 GPIF(General Programmable Interface)集成于一体。其小巧的体积及较高的性能价格比使得 EZ-USB FX2 芯片在海量存储器、打印机、扫描仪和 PCMCIA 等各种 USB 设备上得到了广泛的应用^[3]。图 1 是该芯片的系统结构图。其特点如下:

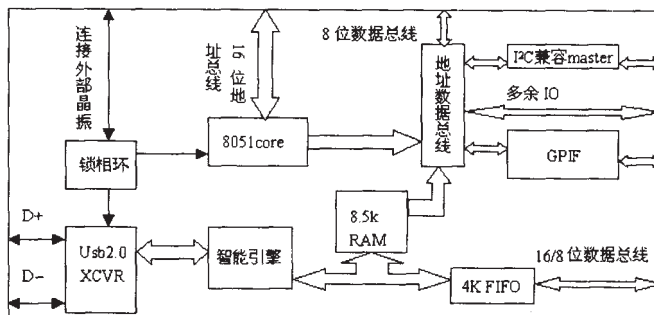


图 1 FX2 的系统结构图^[3]

当该芯片工作在 SLAVE FIFO 下时,并将工作模式设置为 AUTOIN=1 和 AUTOOUT=1, CY68013 此时传输数据并没有经过 8051CPU 处理,而是通过 FIFO 和引擎直接进行主机和外部逻辑电路的数据传输,其流程图见图 2。

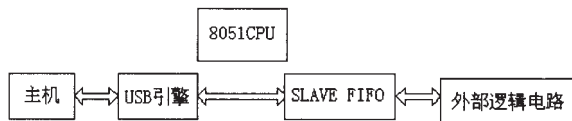


图2 FX2在SLAVE FIFO下工作于自动模式

由于数据没有经过CPU处理,而是FIFO直接与外部逻辑电路直接相连,而FIFO可以工作在48M的时钟频率下,最高传输速度可以达到96MB/s。并且SLAVE FIFO使用的是量子FIFO结构,因此能够实现高速传输。

此外,FX2的2A68端点是大量、宽带宽数据传输端点,可以使用多缓存方式(包括双缓存、三缓存和四缓存)。在数据吞吐率相近的情况下,多缓存可以使带宽运行平稳,减少了数据爆发,减少或者消除了主机与外部逻辑电路相互等待的需要。例如,双缓存的情况下,主机对其中一个512个字节的缓存读或者写的时候,外部逻辑可以同时另外一个512个字节的缓存进行写或者读,读空或写满缓存时,系统进行自动切换。图3是四缓存的示意图。利用上述的方式,可以实现FX2与主机以及外部逻辑电路的高速通讯。

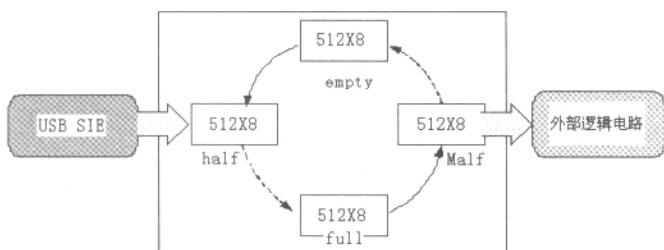


图3 FX2的端点工作在四缓存方式下切换状态图

3 打印机系统结构

打印机消耗数据的速度大约为1.5MB/s,CY68013这款USB2.0设备控制器的端点缓存最大为4K,在USB工作在2.0的协议下,该缓存同时被主机和外部逻辑CPLD(复杂可编程逻辑阵列)读写,因为缓存小,容易造成主机和外部逻辑CPLD频繁中断,甚至相互等待对方完成上一个操作才能进行下一步操作。不利于高速数据传输。通过外接FIFO建立两级缓存,提高缓存深度,降低主机和CPLD中断频率,可以极大提高数据的传输速度,更好地利用USB2.0协议的高速性。从而实现PC机上配置好的数据通过USB的FIFO(先进先出)可以快速准确地传到喷头进行打印。同时也可以通过它将命令传送到DSP(数字信号处理器)的寄存器中,控制两路电机的运动,而不需要额外的串口来进行命令传输,达到简化系统,降低成本的目的。此外,可以方便地实现打印机的状态数据,如环境温度、湿度、喷头温度、纸宽等数据,通过相关指令由CY68013上传到主机并在界面上显示出来,或者与设置参数进行比较,进而闭环控制相关设备。从而使得打印机更加智能化。系统结构参见图4。

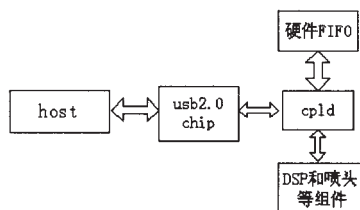


图4 喷墨打印机系统图

4 USB与CPLD接口工作原理

将EP2设置为OUT端点,负责传送打印的数据,大小为1024字节。EP6设置为OUT端点,负责传送从HOST来的命令,大小为1024个字节。EP8设置为IN端点,负责传送来自设备的状态信息到HOST,大小为1024个字节。这样,通过CPLD中的仲裁模块,就可以实现数据和命令的分离以及状态数据的传输,而不要在USB2.0协议之上组建自己的协议以区分数据和命令。图5是CPLD和USB的接口通讯图。

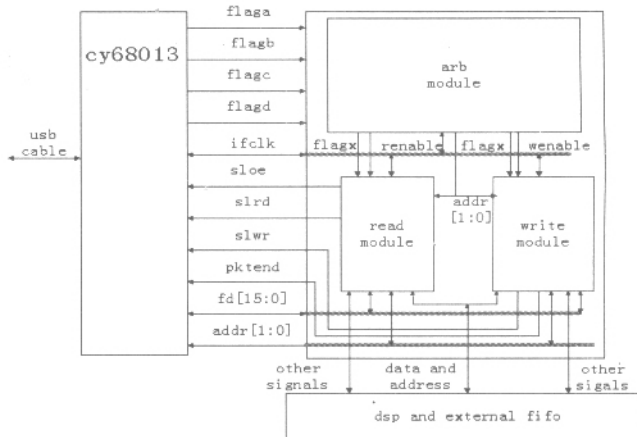


图5 USB和CPLD数据通讯接口图

CY68013与CPLD的通讯是通过仲裁模块仲裁后执行相应的读写模块实现的。从CY68013过来的FLAG标志位输入ARB模块,然后由ARB模块决定读写哪个端点,以区分数据和命令,并执行读写模块中的一个,从而将端点FIFO中的数据放置到外部FIFO中,或者将外部FIFO中的数据写进端点FIFO中。CY68013采用同步方式读写,接口时钟最大为48M,CY68013传输速度可达56MB/s。笔者采用异步读写的方式,在异步读时,接口读时钟最大为10M,异步写时,接口时钟最大为8.3M^[4]。

由于USB传输速度快,异步方式CY68013可以达到10MB/s,喷头消耗数据的速度大约为1.5MB/s,为了发挥USB2.0速度快的优势,同时使得主机可以进行其他任务的处理,CPLD须外接硬件FIFO,建立两级缓存。从CY68013接受的数据先放到硬件FIFO。需要设备状态数据时,CPLD也先将DSP的寄存器中的数据读到外部FIFO中,然后通知USB启动写模块。例如在该模块中,当主机要传送数据给喷头打印时,通过主机发送数据到CY68013端口2。当端口2有数据,在ARB模块中,就会从FLAG标志位检测出端点2的FIFO中有数据,然后做出判断,选择出2端点进行读模块操作。读模块就根据接口时序,将端点FIFO中的数据读出并送到硬件FIFO中进行缓存。当主机要读出打印机的相关状态,就主机向端点6写入命令指令,ARB模块检测到相应的FLAG标志位后,判断有数据就读出指令,并送到DSP相应的寄存器,DSP收到指令后,将有关状态寄存器中的数据集中在自己的缓存中,并通知ARB模块要写,ARB根据当前任务状态,立即响应或者延迟一段时间响应,然后启动写模块,写模块将与相关状态的寄存器数据写入端点8中。主机程序将端点8的FIFO的数据读出来,并做相应处理,从相关界面显示出来或者对相关参数做出调整。

(下转115页)

西的 SXL-100 型^[4],该实验装置是用 AMCC5933PCI 主模式接口芯片转换成 ISA-like 总线,并在该总线上完成一些 82 系列接口芯片使用的实验。其实验内容在硬件上基本和原来的基于 ISA 接口的实验装置一致,软件上主要是基于 Win9x 平台下的 VxD 技术,目前已经过时。

该文提出的实验装置硬件部分抛弃了传统的以 82 系列的片子的使用为核心的实验内容,提出用 EDA 工具来设计接口芯片。从而将原来的验证性实验转换成了设计性实验。软件部分抛弃了传统的 DOS 下汇编程序的做法,而代之以基于 Windows 平台下 (Win9X/WinNT) 的 C 语言的编程和 WDM 驱动程序编写。使学生避免了繁琐而过时的汇编语言编程。

(上接 107 页)

5 软件的编制

USB 设备要能正常工作需要三部分软件程序,第一部分是 USB 设备上运行的固件程序,第二部分是安装在主机上的驱动程序,第三部分是客户要完成具体功能并安装在主机上的客户程序^[5]。

实践证明,该装置不但解决了各大高校面临的旧实验装置不能再使用的问题,而且通过上述的软硬件开发平台,学生可以学习到最新,最实用的软硬件接口技术。

(收稿日期:2003 年 9 月)

参考文献

1.马锦鸣等.高性能个人计算机硬件结构及接口[M].北京:国防工业出版社,2001-05
2.PCI9054 Data Book Version 2.1[S].PLX Technology Inc,2000
3.熊宁.微机键盘通讯方式及其编程[J].宁夏大学学报(自然科学版),1997,18(4):361

```
#define VX_B3 0xB3 // 显示 setup 包
#define VX_B4 0xB4 // 取寄存器值
#define VX_B5 0xB5 // 设置寄存器值
```

主要是使用 0xB4 和 0xB5 两条指令,
0xB4 用于读取相应寄存器的值,结合 FX2 的寄存器总表,可以很容易地查看寄存器的内容。
REQ=0xB4,Value 中为寄存器的地址。
0xB5 用于向相应寄存器写入值。
REQ=0xB5,Value 中为寄存器的地址 Index 为要写入的值。

应用程序实现三个功能:一是通过 API 函数将硬盘上的一个文件中的数据写到 CY68013 的端口 2。第二个功能是将对话框中接收到的参数写到端点 6。第三个功能是将端点 8 中的数据读出来,并做相应处理。

驱动程序是 CYPRESS 提供的样板,该驱动程序几乎涵盖了 CYPRESS 产品,是个通用的驱动程序。

6 方案验证

实验结果证实,两级 FIFO 构成的缓存机制可以进行高速数据传输,EP2 端点测试的下传速度大致为 8.2MB/s。用应用程序界面将下载到设备的数据和从设备上传到主机的数据在主机上显示出来,并将 32 通道的逻辑分析仪监测 CY68013 到 CPLD 的数据总线和 CPLD 到喷头的数据,对比发现完全正确。同时发现 CY68013 对设备到主机的数据有 128 个字节的缓存深度,出现在 CY68013 数据总线的数据要经过 128 个字节才能被应用程序读出。并经过监测,在主机和外部逻辑 CPLD 方面都没有 NAK 和等待的情况,且打印机打印效果良好。命令通过应用程序界面,可以准确地传送到控制设备的寄存器,无明显延时。从客户程序的控制面板能够对打印机进行相应操作,实时地修改相应的控制参数。打印机相关的状态数据,能够准确高速地传送到主机客户程序的界面并实时进行反馈。通过在大中型喷墨打印机上的验证,证明该接口能够进行高速数据传输,并且稳定可靠,实时性较强。(收稿日期:2003 年 8 月)

参考文献

1.www.usb.org
2.University serial bus specification revision2.0[S]
3.Cy7c68013 EZ-USB FX2 USB Microcontroller High-speed Usb Peripheral Controller[M]
4.EZ-USB FX2 Technical reference manual[M]
5.张念淮,江浩.USB 总线接口开发指南[M]

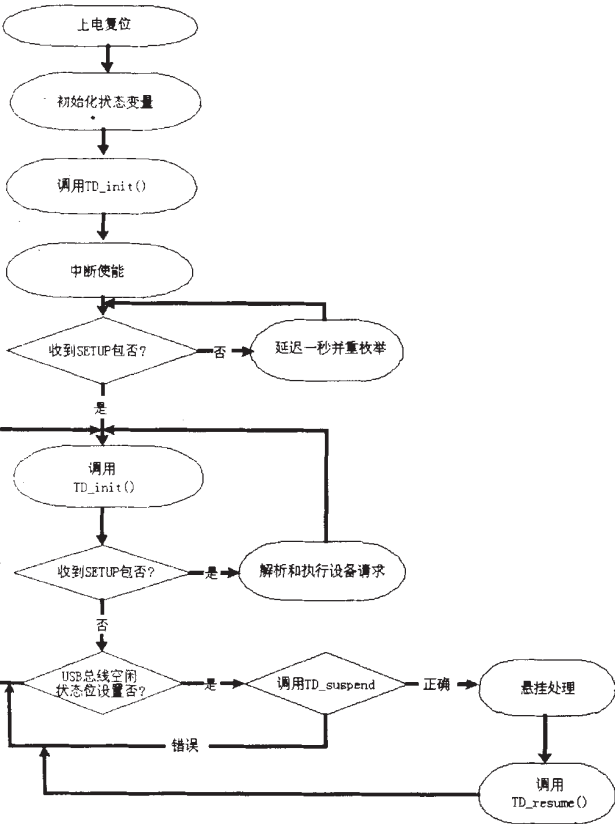


图 6 固件流程图

固件程序的流程图参见图 6。笔者使用的是 SLAVE FIFO 形式,所以在传输过程中 MCU 是没有控制权的。在该程序中设置 EP2,EP6,EP8 的方向,缓存的大小,发送数据包的大小,接口的时钟模式和频率。为了方便调试,笔者加入了一些 Vendor Request 主要有:

```
#define VX_B2 0xB2 // 测试客户请求
```