

基于 USB2.0 的高速数据通信接口设计

An Interface Design of High-speed Data Communication based on USB 2.0

(湖南城建职业技术学院)肖燕娟 杨俊峰

XIAO YANJUAN YANG JUNFENG

摘要: 利用 USB2.0 接口芯片 CY7C68013 实现了语音信号处理系统中 DSP 与 PC 机的高速通信, 简要介绍了 CY7C68013 的功能结构以及 TMS320C6203B 的接口设计, 深入研究了该芯片的固件、设备驱动和应用程序开发。

关键词: EZ-USB FX2; CY7C68013; TMS320C6203B; 数据传输

中图分类号: TP334

文献标识码: A

ABSTRACT: Using CY7C6801 as the control chip, the communication between DSP and PC in the voice signals processing system is realized. A brief introduction of the basic features of EZ2USB CY7C68013 and the designing of interface to TMS320C6203B is provided. The development of Programming Firmware, device driver and application are thoroughly researched.

Key words: EZ-USB FX2, CY7C68013, TMS320C6203B, data transmission

引言

由于 DSP 的工作频率较高, 如 TMS320C6203B 时钟频率高达 250MHz, 故其数据读写周期很短, 然而 PC 机串口读写速度较低, 最大数据吞吐量约为 115kbps, 尽管 DSP 在与这些慢速外设进行数据交换时可以加入额外的等待周期, 但是在实时性要求苛刻, 算法复杂的场合, 将 DSP 从这些冗长的等待周期中解放出来, 将其时间重点放在处理关键的实时任务中去, 有着重要的实际意义。为了提高系统的实时性, 高速接口电路设计显得尤为重要。目前流行的 EZ-USB FX2 芯片的传输速率为 480Mbps, 较好地解决了 DSP 与 PC 机间高速数据传输时的带宽瓶颈问题。本文设计并实现了 TMS320C6000 系列 DSP 上, 利用 EZ-USB FX2 系列芯片 CY7C68013 的接口技术, 实现了 DSP 与计算机之间的高速数据传输。

1 CY7C68013 结构特点

CY7C68013 是 Cypress 公司研制开发的 EZ-USB FX2 系列芯片, 该芯片符合 USB2.0 规范, 同时兼容 USB1.0 的全速工作模式, 它的功能框图主要包括 5 个部分(图 1): (1)收发器。USB 和 PC 机通过 D+ 和 D- 两根信号线进行数据传输, 数据以 480Mbit/s 的全速信号或者以 240Mbit/s 的半速信号在 D+ 和 D- 信号线上差分传输。(2)智能接口引擎(SIE)。SIE 通过包排序、信号产生、信号检测、CRC 产生、CRC 校验、NRZI 数据编码、位填充、包标识产生和解码等功能来处理 USB 通信协议, 并保证传送到 USB 电缆上的数据字节以 LSB 开头。(3)通用微控制器以及片上 RAM。USB 控制器采用通用的 8051 微控制器, 即编程语言为 51 系列单片机的通用语言, 可使实际操作更加灵活方便。(4)I²C 总线控制器。主要为完成将 USB 控制器作为主设备时必须的配置。(5)片内 FIFO。其容量为 4KB, 可以快速实现与不同速度的外设通信。

2 CY7C68013 与 TMS320C6203B 的接口电路设计

CY7C68013 是一个非常方便的 USB2.0 实现方案, 它提供与 DSP 连接的接口, 连接方式有两种: Slave FIFOs 和 Master 可编程接口 GPIF。本文采用 Slave FIFOs 异步读写方式。Slave FIFOs 方式是从机方式, DSP 可以象读写普通 FIFO 一样对 CY7C68013 内部的多层缓冲进行读写。具体的电路连接如图 2 所示:

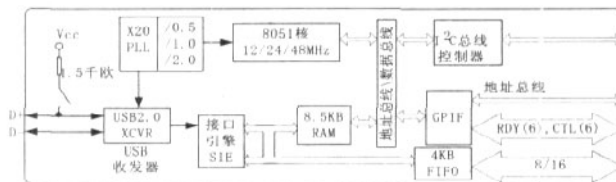


图 1 CY7C68013 功能框图

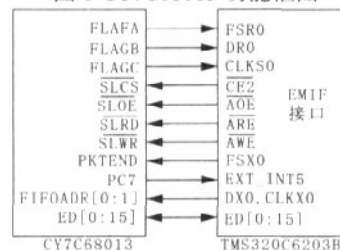


图 2 CY7C68013 与 TMS320C6203B 电路连接图

FLAGA、FLAGB 和 FLAGC 是 CY7C68013 内部 FIFO 的状态标志, TMS320C6203B 通过通用 I/O 口获得 CY7C68013 内部 FIFO 的空、半满(由用户设定半满阈值)和满这 3 个状态信号, EMIF 接口的 CE2 空间对 CY7C68013 进行读写操作。其工作过程为: DSP 通过 USB 向 PC 发送数据时, 首先查看空、半满和满这 3 个状态信号, 然后向 USB 写入适当大小的数据, 以保证数据不会溢出; PC 机通过 USB 向 DSP 发送命令字时, USB 通过中断方式通知 DSP 读取命令字。

肖燕娟: 讲师

3 软件设计

系统的软件设计包括 3 部分: C8051 固件设计、设备驱动程序设计和应用软件设计。CY7C68013 的功能具体实现过程如下: (1) 当设备连接时, Windows 的设备管理器自动检测到该设备, 并读取由 CY7C68013 核心(并非 C8051 固件)传回的设备描述符, 根据它提供的厂商和产品 ID 号 VID/PID (存放在 FX2 外接 E²PROM 中)与 .inf 文件进行比较, 找到固件下载驱动程序(ezloader.sys), 从主机将 C8051 固件下载到 CY7C68013 的 RAM 中。

(2) C8051 固件下载到 RAM 后, CY7C68013 仿真一个脱离再连接至 USB 总线的过程, 执行重新枚举过程, 让 C8051 固件控制 USB 传输。Windows 的设备管理器会检测到新的 USB 设备的连接, 并根据 C8051 固件提供的厂商 ID(VID)/产品 ID(PID) (不同于 CY7C68013 核心提供的 VID/PID) 来加载该 USB 设备的设备驱动程序。(3) 应用软件调用 CreateFile() API 函数, 使用识别该设备的符号链接来取得该设备的代号。取得设备的句柄后, 应用软件通过 DeviceIoControl() 向 I/O 系统服务发出要求传入数据的 I/O 请求, I/O 管理器将此请求构造成一个合适的 IRP 传递给 CY7C68013 设备驱动程序。CY7C68013 设备驱动程序根据该 IRP 中包含的具体操作来构造相应的 USB 请求块, 并据此形成新的 IRP 传递给 USB 总线驱动程序。USB 总线驱动程序根据该 USB 请求块从 CY7C68013 设备读取数据。操作的结果再用 IRP 一层一层地返回给应用程序, 应用程序再对数据进行分析 and 处理。

3.1 C8051 固件设计

Cypress 公司为了简化固件开发, 缩短时间, 加速 USB 外围的开发, 在 EZ-USB FX2 软件开发包里提供了固件架构。该固件架构利用 C8051 的程序代码来实现 CY7C68013 芯片起始设置、USB 标准设备请求的处理以及 USB 闲置模式的电源管理服务。用户仅需提供其 USB 描述符表(即 dscr.a51)及实现外围功能的程序代码(periph.c), 即可完成完全兼容的 USB 设备。

固件的编译在 Keil 的集成开发环境中进行, 在建立的项目中包含 5 个文件 ez_usb.lib、usbjmplb.obj、dscr.a51 和 periph.c, 其中:

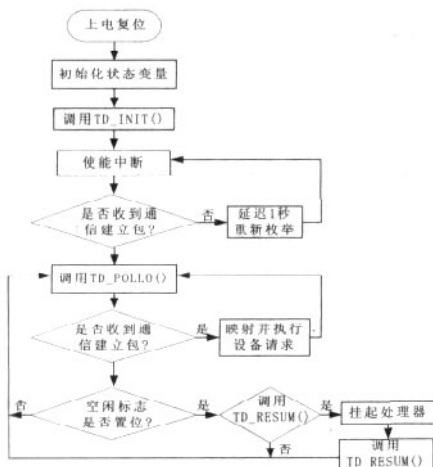


图 3 固件架构程序流程

固件架构程序流程如图 3 所示。该程序首先初始化所有的内部状态变量, 然后调用 TD_Init 函数进行初始化, 初始化主要包括: (1) 使 CY7C68013 工作于异步从 FIFO 模式。相应的寄存器操作为 IFCONFIG=0xCB。(2) 将 4KB 的 FIFO 对应到两个端点

(EndPoint), 即 EndPoint2 和 EndPoint6。相应的寄存器操作为: EP2: FG=0xA0, EP6: FG=0xE2。EndPoint2 与 EndPoint6 分别对应 21KB 的内部 FIFO (下面分别称作 FIFO2, FIFO6), 存放 LUSB 需要上传与接收的数据。其中 EndPoint2 为 OUT 型, 负责从主机接收数据; EndPoint6 为 IN 型, 负责向主机发送数据。另外 EndPoint2 与 EndPoint6 均采用批量(BULK)传输方式, 这种方式相对于其他 LUSB2.0 定义的传输方式具有数据可靠、传输速率高等特点, 是最常用的传输方式。(3) 对 FIFO 进行配置。相应的寄存器操作为 E192FIFO (FG=0x1, EP6FIFO CFG=0x0D)。本案例将 FIFO2, FIFO6 设置成自动方式。

初始化完成后, 打开中断, 之后固件程序开始枚举 USB 设备, 直至端点 0 上接收到通信建立包时为止。一旦收到通信建立包, 将重复执行下面的任务分配过程: (1) 调用 TD_Poll() 函数, 以完成用户指定的任务。(2) 判断是否有 USB 设备请求? 如果有, 则调用 ParseControlTransfer() 函数进行相应的处理; 如果没有, 则继续向下执行。(3) 检测 USB 总线是否空闲。如果空闲, 则调用 TD_Suspend() 函数, 交由用户处理。在 TD_Suspend 返回真值后, 8051 处于空闲状态。这时, 只有 USB 总线活动或芯片 Wakeup 管脚活动, 才能将 8051 重新激活。(4) 8051 被激活后, 固件程序将首先调用 TD_Resume() 函数, 以使 8051 从空闲状态中恢复出来, 然后再处理用户指令。

3.2 设备驱动程序设计

开发 USB 设备驱动程序, 可采用 Numege 公司的开发包 DriverWorks 和 Mitmsort 公司的 Windows 2000 DDK, 并以 VC++ 6.0 作为辅助开发环境。

CY7C68013 的设备驱动程序有两种: 一种用来在设备接入时从主机下载固件到 RAM 中, 称为固件下载驱动程序(ezloader.sys); 另一种是在设备重新枚举后加载的设备驱动程序(ezusb.sys), 应用软件通过该设备驱动程序与 CY7C68013 通信。固件的程序代码既可以通过外部 E²PROM 下载, 也可通过主机下载。本文以从主机下载为例进行说明, 若选择主机下载方式, 则此时外接 E²PROM 中存放的是固件下载的 PID/VID, 且第 1 个字节是 0xC0H, 系统通过该 PID/VID 找到固件下载驱动程序(ezloader.sys), 将相应的固件代码下载到 FX2 的 RAM 中。

在将固件下载到 RAM 中后, 系统清除内存中的固件下载驱动程序, 并进行重新枚举, 让 C8051 固件控制 CY7C68013。此时, 系统获得由 C8051 固件提供的 PID/VID (与 E²PROM 中的不同), 认为有新的 USB 设备接入, 并据此加载相应的设备驱动程序。

3.3 应用软件设计

应用软件满足系统与用户的接口, 它通过通用驱动程序完成对外设的控制和通信, 如开启/关闭 USB 设备、检测 USB 设备、设置 USB 数据管道等。它由动态链接库和应用程序两部分组成: 动态链接库负责与 USB 设备驱动程序通信, 响应应用程序的各种请求; 应用程序则则以定时查询的方式将 TMS320C6203B 后的数据读入计算机事先开辟的大容量内存缓冲区, 同时从缓冲区读出数据实时显示, 处理结束后再将内存缓冲区的数据存盘以及向 TMS320C6203B 发送控制命令。

动态链接库对设备驱动程序的访问都是通过 I/O Control 调用来实现。首先调用 Win32 函数 CreateFile() 得到设备驱动程序的句柄, 然后调用 Win32 函数 DeviceIoControl(), 通过得到的句柄把 IOCTL 和相关的输入输出缓冲区提交给驱动程序。当应用软件退出时, 用 CloseHandle() 函数关闭设备。

在编写应用程序时,先建立与外设的连接,然后实施数据的传输。在系统启动信号处理算法后,为保证不丢失数据,用户程序应建立一个新的工作线程专门获取外设传来的数据。程序中主要用到两个 API 函数 CreateFile() 和 DeviceIoControl()。CreatFile()取得设备句柄后,DeviceIoControl()根据该句柄完成数据传输。

部分程序代码如下:

```
hDevice=CreateFile("\\\\.\\EZUSB_0",
GENERIC_READ|GENERIC_WRITE,
FILE_SHARE_WRITE,NULL,OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,NULL);
If(hDevice!=INVALID_HANDLE_VALUE)
{ Application->MessageBoxA("创建设备失败,请确认设备是否
连接上!",NULL,IDOK);}
else{DeviceControl(hDevice,IOCTL_EZUSB_BULK_WRITE,
&blkctl,sizeof(BULK_TRANSFER_CONTROL),
&inBuffer,
sizeof(inBuffer),
&nBytes,
NULL);
...}
```

4 结论

采用两种方法对本系统的数据传输速度进行测试:(1) DSP 将 ADC 采集的语音信号进行处理后,将其频率、相位等数据上传给计算机进行分析处理;(2)计算机将控制指令传给 DSP。结果显示,两种方法的传输速度均可达到 64Mb/s 以上,完全可以满足实时数据处理的需要。

该接口电路在本人开发的基于 DSP 的语音信号采集与处理系统中,实现了数据高速传输的任务,工作性能良好。

本文作者创新点:传统的信号处理系统中数据传输接口在低速时常采用标准串行口或并行口,高速时一般采用 PCI 总线接口。但标准串行口或并行口数据传输速率较低;PCI 总线尽管数据传输速率可以达到 1Gbps,但是硬件设计和驱动开发难度较大,USB 总线具有快速、双向、低价并可热插拔等特性,而且支持各种 PC 与外设之间的连接,本文将 USB2.0 接口芯片 CY7C68013 用于信号处理系统中,处理后的数据经 USB 向计算机传输,速度得到了大大提高,并且系统可靠性也达到了理想效果。本文所述的方法具有一定的普遍性,可用于多种微处理扩展 USB 接口的方案中。

参考文献:

[1] 徐庆元,张天序,钟胜.基于 USB 总线的高速视频采集系统设计[J].微计算机信息,2006,22(10-1):247-249.

作者简介:肖燕娟(1967-),女,湖南武冈人,学士,讲师,主要从事计算机教学与研究;杨俊峰(1968-),男,江西上饶人,学士,讲师,主要从事计算机教学与研究。

(411101 湖南湘潭 湖南城建职业技术学院)肖燕娟 杨俊峰
通讯地址:(411101 湖南湘潭市岳塘区书院路 42 号湖南城建职业技术学院)肖燕娟

(收稿日期:2007.11.13)(修稿日期:2008.1.15)

(上接第 182 页)

2.5 在 CPLD 芯片中顶层文件的原理图,如图 3 所示。

图中 JPXZ 模块为节拍产生器,输入端 CLK 接振荡器的输

出时钟,频率为 4Hz,PC 为节拍选择信号,PP 为输出节拍。CAI_LIZI 模块为地址码产生器,CAI_BMQ 模块为编码发生器,输出端 QOUT[31..0] 输出 32 位状态编码信号,接缓冲驱动电路。

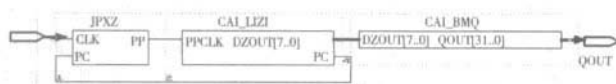


图 3 CPLD 芯片中顶层文件的原理图

2.6 缓冲驱动电路

该模块为彩灯提供需要的工作电压和电流,隔离负载对系统工作的影响。首先根据每路彩灯的功率选择继电器或双向可控硅,再根据继电器或双向可控硅所需驱动电压和电流设计驱动电路。详细设计此处不再介绍。

3 系统仿真测试与实物测试

在基于 MAX+PLUS 软件平台,对该系统程序各模块进行仿真测试,图 4 所示为彩灯控制器顶层文件仿真波形图。



图 4 彩灯控制器顶层文件仿真波形图

图中 CLK 为振荡器产生系统的主时钟,周期为 0.25 秒,qout[31..0]为彩灯控制器输出的 32 路数字控制信号(32 位状态编码),图中所示当系统以慢节拍(0.5 秒,2 个周期一拍)输出最后一组编码后,马上以快节拍(0.25 秒,1 个周期一拍)开始下一轮花型循环,达到设计要求。

将该系统程序下载到 MAX7000S 系列的 EPM7128SLC84-15 目标芯片上,并配以外围电路进行实物测试(以 LED 灯代替彩灯),满足到设计要求。

4 结束语

本文作者创新点:以 VHDL 硬件描述语言进行设计,将彩灯控制器的核心部分集成在可编程逻辑器件 CPLD 芯片上,大大简化了外部电路,较以前的传统设计,既减少了所用芯片的种类和数量,缩小了体积,降低了功耗,提高了系统的整体性能,对系统在使用中的故障率大为减少。特别是,对花型设计方案的修改、增加花型数量都很方便,可以在不修改硬件电路的基础上,仅通过更改软件就可实现,具有广阔的应用前景。

参考文献

[1]李景华 杜玉远.可编程逻辑器件与 EDA 技术[M].东北大学出版社 2001.

[2]徐志军 徐光辉.CPLD/FPGA 的开发与应用[M].电子工业出版社 2002.

[3]赵明富.基于 CPLD 的彩灯控制器[J].微计算机信息 2005 5-2:49-50.

作者简介:陈华容,(1965-),女,湖南祁东人,高级工程师,研究方向:EDA 技术、PLC 控制。

Biography:Chen Hua-rong (1965-), Female (Han nationality), Hunan Province, Electric and Information Engineering Department of Hunan Industry College, senior engineer, research area: EDA technology, PLC control.

(421008 湖南 衡阳 湖南工学院电气与信息工程系) 陈华容 姚胜兴

通讯地址:(421002 湖南省衡阳市珠晖区湖南工学院酃湖校区湖南工学院电气与信息工程系)陈华容

(收稿日期:2007.12.03)(修稿日期:2008.1.05)