基于 FPGA 的数据采集系统的设计*

刘 军,岳兴莲,何国国,罗 石,吴硕开
(江苏大学,江苏 镇江 212013)

摘 要:介绍了一种用于汽车姿态测量的数据采集系统的设计,该系统基于 FPGA+USB 架构,采用 FPGA 控制整个系统的采集时序,USB 芯片作为数据采集通道,上位机完成姿态解算和数据显示功能。

关键词:FPGA;USB;数据采集

中图分类号:TP274+.2 **文献标识码**:A

Design of data acquisition system based on FPGA

LIU Jun, YUE Xing Lian, HE Guo Guo, LUO Shi, WU Shuo Kai
(Jiangsu University, Zhenjiang 212013, China)

Abstract: This paper introduced the data acquisition system using in a vehicle attitude measurement system. The acquisition system uses FPGA + USB architecture, FPGA control the whole system of collecting time-series, USB chip act as a data acquisition channel, and PC profile solver and data display.

Key words: FPGA; USB; data acquisition

现代化生产和科学研究对采集系统的要求日益提高。传统的采集卡速度慢、处理功能简单、采用分立元件、电路非常复杂;而且可靠性差、不易调试、不能很好地满足特殊要求。现场可编程门阵列(FPGA)是专用集成电路中集成度最高的一种,用户可对 FPGA 内部的逻辑模块和 I/O 模块重新配置,以实现用户所需逻辑功能。用户对 FPGA 的编程数据放入芯片,通过上电加载到 FPGA 中,对其进行初始化;也可在线对其编程,实现系统在线重构[1]。本系统设计采用 USB2.0 CY7C68013 通信接口芯片作为数据采集通道,由 FPGA 芯片 EP1C6Q240C8N 作为采集设备的控制单元,由 PC 机完成姿态的解算及结果的显示。

1 系统的组成及原理

该采集系统主要由前端调理模块、A/D 转换控制模块、SRAM 存储模块及 USB 接口模块组成,系统框图如图 1 所示。

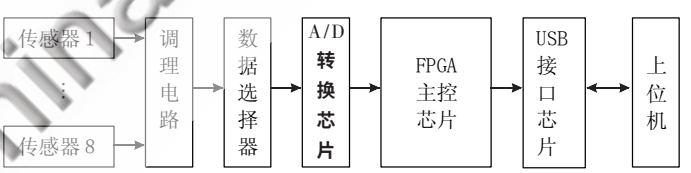


图 1 系统设计方案

2 FPGA 主控器内部模块

2.1 A/D 转换控制模块

A/D 转换模块选用 AD7685 芯片。AD7685 是 Analog Device 公司生产的一款 16 位、电荷再分配、高速、低功耗的逐次逼近型模数转换器(ADC),具有 250 kS/s 采样速率。芯片在 Verilog 编程语言的控制下,完成模拟信号到数字信号的转换。

2.2 FIFO 缓存

调用 FPGA 片上资源实现片上 FIFO 缓存,由于 A/D 采样频率与 SRAM 的读写频率不一致,所以采用读写时钟不同的 FIFO,达到数据缓存和转换时钟域的双重目的。

2.3 SRAM 乒乓缓存模块

选用 2 片 IS61LV25616 存储芯片,该芯片存储容量为 256K×16,采用 Verilog 硬件描述语言控制实现乒乓缓存,控制过程如图 2 所示。从片上 FIFO 输出的数据经选择开关后,分别进入缓冲模块 1 和缓冲模块 2。当数据写入缓冲模块 1 时,USB 模块从缓冲模块 2 读取数据;当数据写入缓冲模块 2 时,USB 模块从缓冲模块 1 读取数据以传到上位机进行处理。

* 基金项目:汽车动态模拟国家重点实验室开放基金资助项目(20071104)

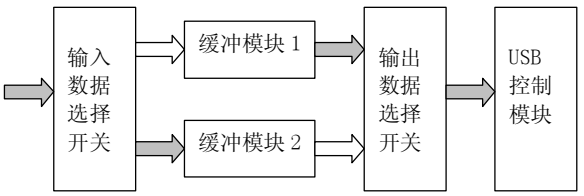
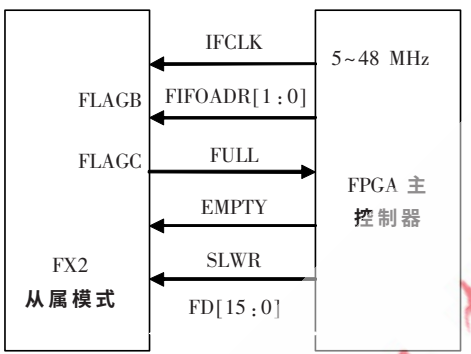


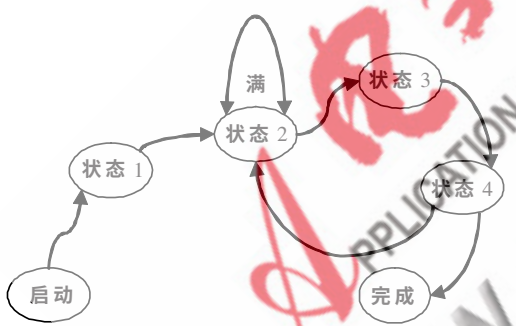
图 2 乒乓缓存工作原理

2.4 USB 控制模块

USB 接口芯片采用 EZ-USB FX2(CY7C68013),FX2 作为 USB2.0 数据通道来实现与主机的高速通信。FPGA 能够满足 Slave FIFO 要求的传输时序[2]作为 Slave FIFO 主控制器。图 3 分别给出了 FX2 与 FPGA 的接口图和状态转换图。



(a)引脚接口:同步 FIFO 写



(b)同步写状态设计

图 3 FX2 与 FPGA 接口和状态转换图

同步 Slave FIFO 写时序如下:

IDLE:当写事件发生时,转到状态 1。

状态 1:指向 IN FIFO,激活 FIFOADR[1:0],转向状态 2。

状态 2:如果 FIFO 满标志为"假"(FIFO 不满),则转向状态 3;否则停留在状态 2。

状态 3:传送总线驱动数据,为 1 个 IFCLK 激活 SLWR,转向状态 4。

状态 4:如果有更多的数据要写,则转向状态 2;否则转向 IDLE。

3 USB 芯片固件程序及驱动程序

3.1 FX2 的固件程序设计

CY7C68013 芯片固件程序负责处理 PC 机发来的各种 USB 请求,以完成主机与外围电路间的数据传输。写固件程序是比较复杂的,需要用到大量的函数,但其基本结构却相对简单,包括下面 3 个过程[3]:

(1)初始化:处理器和外围电路的初始化。

(2)主函数:完成符合设备特定要求的代码。

(3)中断处理:处理各种中断的程序代码。

Cypress 公司的 EZ-USB FX2 开发套件提供给用户 1 个固件函数库(Ezusb.lib)和固件框架(Framework),两者均是基于 KEIL C51 进行开发的。固件函数库提供了一系列的函数来加速 USB 固件程序的开发,使用时只需在程序中包含 EZUSB.H 和 EZREGS.H 两个头文件,并在项目中链接 Ezusb.lib,就可以直接使用固件库中的各个函数[4]。

在程序起始时,固件框架将执行如下步骤:

(1)首先设置所有的内部状态变量,即设置起始的初值。

(2)调用用户的初始设置函数 TD_Init(),待返回后,固件框架会设置 USB 为未配置的状态,并且使能中断。

(3)紧接着在 1 s 的间隔内,开始重新列举设备,并直到设置(SETUP)封包收到端点 0 为止。

(4)一旦 SETUP 包被检测到,固件程序结构框架就开始进行任务分配。任务分配就是依次重复地执行下面的过程:

①调用用户函数 TD_Poll()。

②检测是否有标准的设备请求,如果有,则执行指令并做出相应的操作。

③检测 USB 核是否有 USB 挂起信号,如果收到,则调用用户程序 TD_Suspend(),从该函数成功返回后(返回值为 TRUE),再检测是否发生 USB 唤醒事件。如果未检测到,则处理器处于挂起方式;如果检测到,则调用用户程序 TD_Resume(),程序继续运行。如果从 TD_Suspend() 返回为 FALSE 时,则程序继续进行。

标准请求和 vendor 专用请求由框架分析和执行。默认情况下,对标准请求执行 USB 规定的响应,无论如何,框架提供交互的连接,以允许用户程序处理或覆盖指定的设备请求。EZ-USB 中断也交给框架进行处理,任务循环的流程图如图 4 所示。

在 FX2 芯片的固件程序设计中,最关键的就是系统初始化 TD_Init(void)[3-5],下面是其部分代码。

```
void  TD_Init(void)
{
    CPUCS=((CPUCS  &  ~bmCLKSPD)|bmCLKSPD1);
                          //设置 CPU 时钟频率为 48 MHz
    SYNCDELAY;
                  //设置 68013 工作于 Slave  FIFO 模式
    REVCTL=0x03;
```

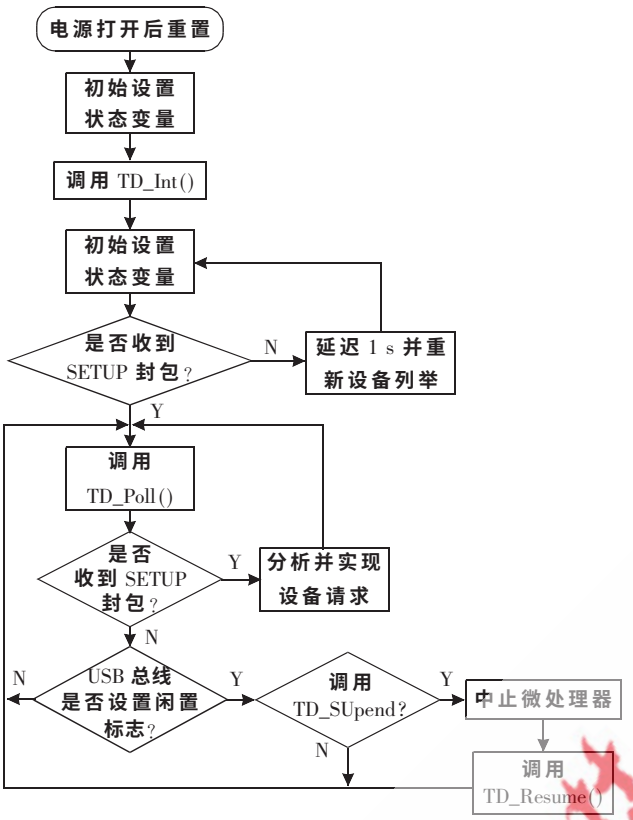


图4 固件框架流程图

```
//必须设置REVCTL.0 和 REVCTL.1 为 1
SYNCDELAY;
IFCONFIG=0x43;                    //工作于同步 FIFO 模式
SYNCDELAY;                        //配置各个端点的工作状态
EP2CFG=0xE8;          //端点 2,IN,块传输,1 024 B,4 倍
                                  //缓冲区
SYNCDELAY;
EP2FIFOCFG=0x09;      //配置端点 2 工 作于 16 位模
                      //式,自动接收 IN 令牌包
SYNCDELAY;
PINFLAGSAB=0x00;      //定义 FLAGA 为可编程级标
                      //志,FLAGB:FIFO 满标志位
SYNCDELAY;            //定义 FLAGC 为满标志,
PINFLAGSCD=0x00;
SYNCDELAY;                        //一般不需要 FLAGD
PORTACFG=0x00;        //用 PA7/FLAGD 作为端口引脚,
                      //不作为 FIFO 标志
SYNCDELAY;
FIFOPINPOLAR=0x00;          //设置所有 FIFO 接口引脚
                            //为低电平有效
SYNCDELAY;
EP2AUTOINLENH=0x02;   //端点 2 自动接收 512B 数据
SYNCDELAY;
EP2AUTOINLENH=0x00;
SYNCDELAY;
                            //复位 EP2 的 FIFO 缓冲区
FIFORESET=0x80;             //不接收主机发出的命令
SYNCDELAY;
FIFORESET=0x02;           //复位 EP2 的 FIFO 缓冲区
SYNCDELAY;
FIFORESET=0x00;                     //恢复正常工作
SYNCDELAY;
Rwuen=TRUE;                 //使能远程唤醒功能
}
```

3.2 USB 设备驱动程序

USB 设备驱动程序的主要功能是使 Win32 应用程序能正确访问本数据采集卡的硬件设备。本设计中将 CY7C68013 的固件代码存放在上位机上,当系统上电或 USB 连接时,再将它下载至芯片的 RAM 中,由增强型 8051 执行。这一过程需要使用 2 个驱动程序:1 个用于下载芯片的固件程序,另 1 个用于实现本数据采集卡的具体功能。也可以使用 EZ-USB 的通用驱动程序,很多 USB 芯片的厂商都为其 USB 芯片提供了通用驱动程序,可以满足大部分系统的需求,用户可在此基础上直接进行固件程序的开发[6]。

4 主机应用程序

应用程序主要负责读取系统硬件所输出的数据采集结果,并实时显示波形,所使用的编程语言为微软的 Visual C++6.0 语言编写的 Win32 应用程序。

主要控件包括:采集控制组按钮,USB 组按钮。采集控制组按钮负责控制硬件系统是否进行数据采集,并使用 USB 块传输来读取采集结果。USB 组按钮主要负责读取该数据采集卡的 USB 设备描述符和配置描述符。

在该数据采集系统的设计中,CY7C68013 芯片灵活的接口和 FPGA 可编程特性简化了外部硬件的设计,提高了系统的可靠性,且利于设备的生产与调度。事实证明,本文设计的系统完全满足设计和使用要求。

参考文献

[1] 严雪萍.基于 FPGA 的高速数据采集系统[J].微计算机信息,2008(1-2):209-211.
[2] 华清远见嵌入式培训中心.FPGA 应用开发入门与典型实例[M].北京:人民邮电出版社,2008.
[3] 钱峰.EZ-USB FX2 单片机原理、编程及应用[M].北京:北京航空航天大学出版社,2006.
[4] 李英伟.USB2.0 原理与工程开发(第 2 版)[M].北京:国防工业出版社,2007.
[5] 张弘.USB 接口技术[M].西安:西安电子科技大学出版社,2002.
[6] 薛园园.USB 应用开发技术大全[M].北京:人民邮电出版社,2007.

(收稿日期:2009-06-08)