

基于 FPGA 的多功能数据采集系统设计

刘欢,朱建鸿

(江南大学轻工过程先进控制教育部重点实验室,江苏无锡 214122)

摘要:介绍了一种高速实时数据采集系统的设计,该系统以 FPGA 作为逻辑控制的核心,以 USB 作为与上位机数据传输的接口,内置一路 AD 转换,最高采样速率 40 MHz,预留 40 针 IDE 接口,可以扩展各种不同的传感器与 AD 板,数据传输速度超过 40 MB/s。给出了系统内部结构设计图,设计思路、实施过程、仿真结果和实验结果、固件(Firmware)和基于 C# 的应用程序开发,以期对各种传感器的调试与数据采集等提供有益的参考。

关键词: 电子设计;数据采集;传感器;电压转换;USB;FPGA

中图分类号:TN919.6

文献标识码:A

文章编号:1674-6236(2016)13-0167-05

Design of multifunction data acquisition system based on FPGA

LIU Huan,ZHU Jian-hong

(Key Laboratory of Advanced Process Control for Light Industry,Ministry of Education,Jiangnan University,Wuxi 214122,China)

Abstract: The design of a high-speed real-time data acquisition system with the FPGA as a core logic control is introduced. The system has the following characteristics: using USB2.0 as data transfer interface to the host computer, building in an AD conversion, the maximum sampling rate 40 MHz, 40-pin IDE reserved for interface, expansion of a variety of sensors and AD board, and data transmission speed being over 40 MB/S. The internal structure design of the system, design ideas, implementation, simulation and experimental results, and the firmware and application based on C# are given for a variety of sensors to provide a useful reference for debugging and data acquisition.

Key words: electronic design; data acquisition; sensor; voltage conversion; USB; FPGA

传统的数据采集系统中通常采用单片机或 DSP 作为控制模块,控制 AD 转换、存储和其它外围电路的工作,利用串口、并口等接口与上位机进行通信。但随着数据采集对速度、简易性和精确度的要求日益增加,传统的数据采集系统逐渐无法满足应用要求。

PCI 接口^[1],SATA 接口^[2]和 USB 接口^[3]是高速数据采集与传输领域三大主流选择,但本文采用了 USB2.0 接口,因为 USB2.0 稳定性高、通用性好、兼容性强和速度快。基于 FPGA 的多功能高速实时数据采集系统采用了 FPGA 逻辑主控和 USB2.0 传输的思路。FPGA 芯片具有体积小、频率高、延时小,且能够使用 Verilog HDL 语言来编程的优点。本系统结合了两者的优点,具有速度快、容易拓展的特点。

1 系统总体结构

本文设计的基于 FPGA 的多功能数据采集系统采用 FPGA 芯片作为系统的控制核心,通过 USB2.0 端口与上位机通信,实现多功能、实时的高速数据采集。本采集系统划分为采集驱动电路、FPGA 采集控制电路和 USB 传输控制电路等部分。如图 1 所示。

本系统内置 1 路模拟信号输入,含 12bit 的 AD 转换器,

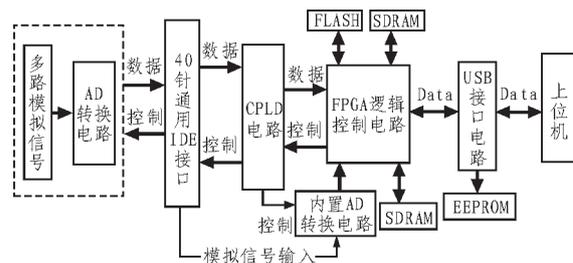


图 1 数据采集系统整体框图

最高采样率为 40 MHz, 适合用于 CCD (Charge Coupled Device, 电荷耦合器件)等图像传感器的数据采集。本数据采集系统支持外置扩展采样接口 (IDE 接口), 此接口由 CPLD 驱动,起到多路电压转换的功效,能兼容 5 V 或 3.3 V 的系统或传感器。该系统的工作过程为:主机应用程序向 USB 控制器发出采样控制包,在控制包中设置采集频率、传感器时序驱动等,进而 USB 控制器触发 FPGA 驱动传感器并采样,FPGA 根据控制包的要求驱动 AD 进行数模转换,边转换边将数据缓存到 SDRAM,数据达到一定域值后,被打包传送至 USB 控制器,再由 USB 控制器将数据高速传送至上位机处理。

2 主要芯片选型

本数据采集系统采用的主要芯片包括 FPGA 芯片、CPLD

收稿日期:2015-07-15 稿件编号:201507112

作者简介:刘欢(1991—),男,湖北天门人,硕士研究生。研究方向:检测与传感。

芯片、AD 转换芯片和 USB 传输芯片。

依据本系统设计的需要选用 Altera 公司的 Cyclone IV 系列 EP4CE15F17C8 芯片。该芯片内部集成逻辑单元 15 408 个,用户 I/O 口 165 个,PLL(锁相环)4 个,系统时钟频率可高于 200 MHz,能满足设计的要求。

根据电压要求,本系统采用 Altera 公司的 MAX II 系列 EMP240T100C5 芯片^[5]。该芯片具有 80 个通用 I/O 口,支持 1.8 V、2.5 V、3.3 V 和 5 V 等多种电压输入输出,能满足设计需求。

根据采集系统的采样率、分辨率等要求,本系统选用 Analog Devices 公司的 AD9924 模数转换器件。该器件是 12 位低功耗高速并行模数转换器件,最高采样率为 40 MHz。

Cypress 公司的 EZ-USB FX2 系列芯片是一款集成了 USB2.0 协议的微处理器。本文采用了 EZ-USB FX2LP 系列低功耗芯片中的 CY7C68013A-56(下文简称为 EZ-USB)。

3 系统硬件设计

3.1 多功能数据采集电路设计

传感器的驱动电压与 AD 芯片的输出电压常为 5 V 或

3.3 V,如 SONY ILX554B CCD 和 TOSHIBA TCD1304DG CCD 的传感器驱动电压分别为 5 V 和 3.3 V;AD9220 与 AD9224 模数转换芯片的输出电压分别为 5 V 和 3.3 V。这些信号都要与主控芯片相连。为了信号能够被准确捕捉,需要进行电压转换,以往常使用专用的电压转换芯片,如 SN74LVC4245,可以实现 8 路 3.3 V 与 5 V 的相互转换,但这类芯片体积大,转换通道少,在通道需求大、PCB 空间小的情况下对电路设计者非常不利。

本系统选用 CPLD(EPM240T100C5)通过软件硬件结合控制实现多功能数据采集,如图 2 所示,其中 U1 为 CPLD 部分 I/O 口,S1 为四路硬件开关,R1-R4 为上拉电阻,P1 为 40 针通用 IDE 接口。CPLD 部分 I/O 口与 IDE 接口相连,I/O 口与 FPGA 相连(图中未画出),通过编程实现各个 I/O 口之间的同步,然后通过 QUATAUS II 对 CPLD I/O 口输入输出模式进行设定,并配合硬件开关控制上位电阻的闭合来实现 I/O 口的电压转换,实现 4 个 8 位 I/O 口分别以 3.3 V 和 5 V 独立输入输出,还有一个 8 位 I/O 口(其中一位已被内置 AD 输入占用)直接与 FPGA 相连,可以进行 3.3 V 的输入输出。具体实现过程分为 4 种情况(4 个 I/O 口同理,以 PA 口为例):

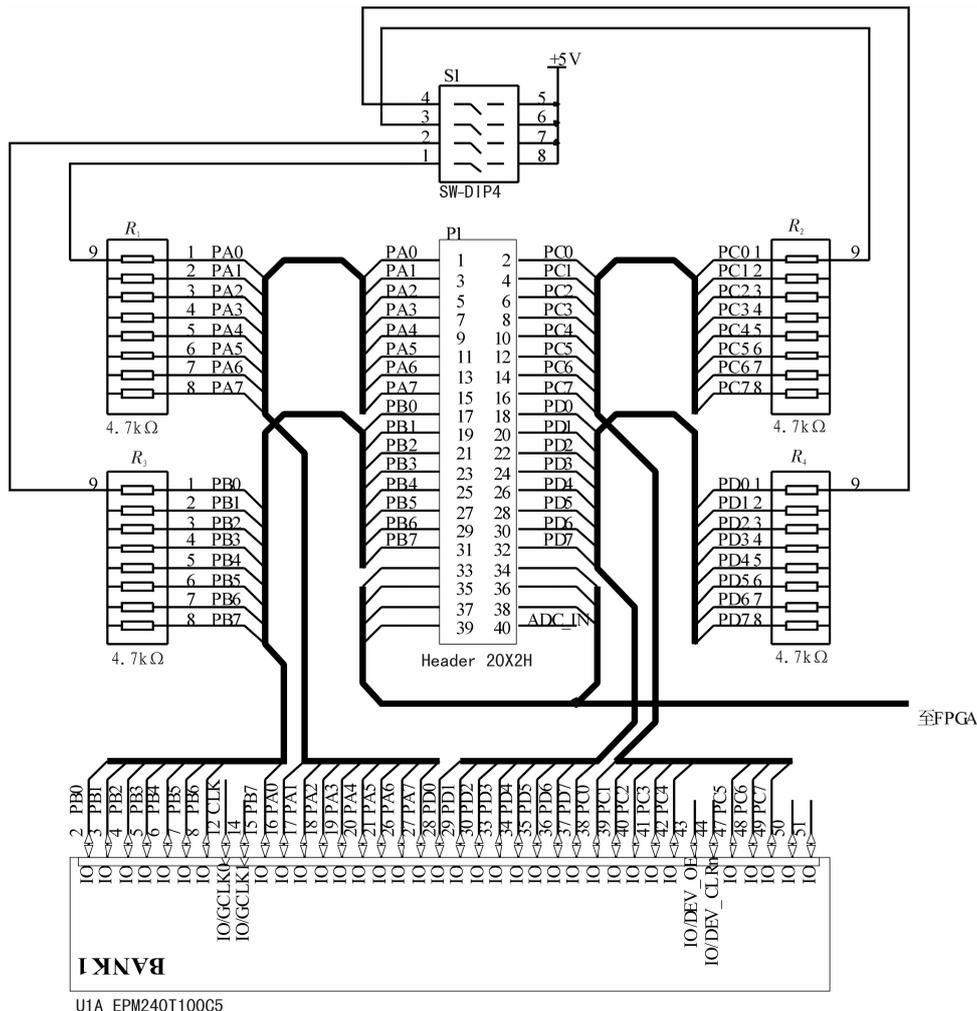


图 2 多功能数据采集的连接

①PA口3.3V输出,此时设置对应的CPLD I/O口为普通输出模式,断开S1第一位,再通过程序实现该I/O口与FPGA对应的I/O口时序同步。

②PA口5V输出,此时设置对应的CPLD I/O口为开漏输出模式,闭合S1第一位(加上拉电阻),再通过程序实现该I/O口与FPGA对应的I/O口时序同步。

③PA口3.3V输入,此时设置对应的CPLD I/O口为普通输入模式,断开S1第一位,再通过程序实现FPGA对应的I/O口与该I/O口时序同步。

④PA口5V输入,此时设置同③,EPM240T100C5产品手册表明,当VCCIO=3.3V时,能兼容5V TTL输入信号,AD数模转换芯片输出信号一般均为TTL信号,能兼容。

3.2 FPGA与SDRAM及USB芯片连接电路设计

本系统采用SDRAM(同步动态随机存储器)代替传统的FIFO进行数据缓存,此为两级数据缓冲的第一级,两片SDRAM轮流操作,进行通常所说的乒乓操作^[8]。SDRAM型号为H57V2562GTR-75C,单片容量为32M(普通FIFO芯片的容量一般为32~512K),价格较贵,但从容量的角度来考虑,SDRAM存在明显优势。SDRAM由FPGA控制读写,用来暂存AD电路所产生的数据。

EZ-USB有多种数据传输模式,如GPIF模式、SLAVE FIFO模式等,甚至端口模式都可以进行数据传输。其中传输速度最快的是SLAVE FIFO模式,可以达到40MB/S。

当EZ-USB工作于SLAVE FIFO模式时,外围电路可以像对待普通FIFO一样对EZ-USB内部FIFO进行读写^[7],此为两级数据缓冲的第二级。本系统采用FPGA作为主控制器,USB控制器采用SLAVE FIFO模式,FPGA通过控制EZ-USB内部FIFO及SDRAM进行与上位机之间的数据传输。

FPGA与SDRAM及USB芯片信号连接图如图3所示。其中,在与SDRAM连接中,A[0..12]为13位地址位,DQ[0..15]为16位输入输出数据位,INCLK_SD为FPGA PLL锁相环二倍频产生的100MHz时钟信号,是SDRAM的主时钟信号,其它为控制信号;在与EZ-USB的连接中,data[0..15]为16位数据位,其它为控制信号或状态位。

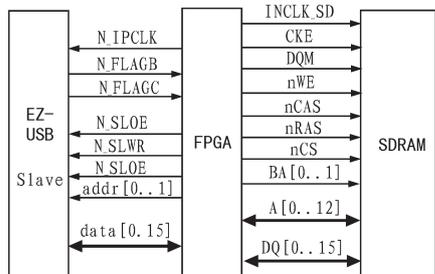


图3 FPGA与SDRAM及USB芯片信号连接图

4 系统软件设计

系统软件设计主要包括FPGA逻辑控制程序设计、CPLD程序设计、USB固件程序设计和上位机应用程序设计。

4.1 FPGA逻辑控制设计

FPGA逻辑控制程序的设计是整个数据采集系统的关键,整体分成三大模块:PLL锁相环模块、分频采样控制模块和NIOS II主控模块。本设计中使用QUARTUS II提供的IP核进行编程,降低了开发的难度并缩短了开发周期。FPGA逻辑控制模块化设计原理图如图4所示,上文已提及的外设连接也在图中显现。

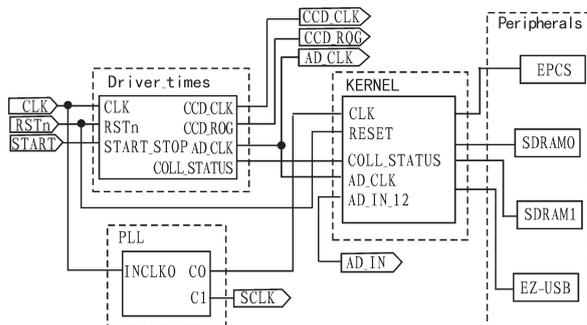


图4 FPGA模块化设计原理图

4.1.1 PLL锁相环模块

NIOS II内核和SDRAM的时钟频率高于FPGA的主时钟50MHz,因此,本系统采用PLL锁相环的倍频功能来提升时钟频率到100MHz,从而驱动NIOS II内核和SDRAM。PLL锁相环模块的建立既可以通过Verilog HDL直接编写,还可以通过QUARTUS II提供的IP内核来设置。本文采用后者,具有方便、便捷的优点。如图4中PLL模块所示。

4.1.2 分频采样控制模块

FPGA主时钟由50MHz有源晶振提供,由于A/D采样的频率要根据实际频率需求设定500KHz~40MHz,因此需要构建分频采样控制自定义模块,该模块对主时钟进行若干分频得到相应的驱动同步信号。同时,该模块还要控制数据采集、AD转换以及数据同步存储。如图4中driver_times模块所示,其中CCD_ROG、CCD_CLK分别是传感器驱动信号,AD_CLK是ADC芯片的主时钟信号并作为NIOS II内核的输入信号为数据存储提供同步信号。

4.1.3 NIOS II主控模块

NIOS II主控模块主要分为USB传输控制部分和SDRAM缓存控制部分,它们在数据和功能上有连通性,所以把它们集成在一个内核模块中,如图4中KERNEL模块所示。NIOS II同时也是一个IP内核^[9],它的作用相当于一个32位RISC嵌入式处理器。嵌入式开发所采用的高级语言C语言相对于FPGA的Verilog HDL或是VHDL等硬件描述语言具有开发难度低,逻辑明确等优点。采用嵌入式内核之后并没有损失FPGA的并行执行能力,一个工程中可以建立多个内核,依然有并行执行程序的能力,相当于多个内核同时工作,FPGA加ARM组合形式,使得代码执行效率更高。

1)SDRAM缓存控制

如图4所示,在KERNEL软核中已经建立了两个SDRAM控制器,SDRAM控制器一旦构建好,接下来对

SDRAM 的处理就像对内部 RAM 地址一样进行赋值和读取。32MB 的 SDRAM 中除了 NIOS II 系统的运行占用了少量空间(8KB),其余均可用于数据存储。用 C 语言中的指针操作可以轻松的完成 SDRAM 的读写工作,下面给出实现 ADC 转换出的数据存储到 SDRAM0 的关键代码:

```
unsigned short * ram0 = (unsigned short *) (SDRAM_BASE + 0x10000); //SDRAM 地址
void save_data_0()
{
    memset(ram0,0,16000000); //内存初始化
    while(Coll_State)
    {
        if(AD_CLK)
            *(ram0++) = ADC_ONE_DATA;
    }
}
```

数据总线为 16 位,所以每存储一个数据,指针向后移动一次(16 位),由系统内部自动处理完成,数据的地址不需要人工干预。从 SDRAM0 中读出数据也是如此,下面给出了从 SDRAM0 读出数据送至 EZ-USB 数据 IO 口的关键代码:

```
void read_data_0()
{
    while(! flagb)
    {
        if(! ifclk)
            CY->DATA = *(--ram0);
    }
}
```

其中 CY->DATA 是用结构体定义的 IO 口,ifclk 为 EZ-USB 的 IO 口时钟。

SDRAM1 的工作过程与 SDRAM0 相同,两块 SDRAM 交替工作,在向一块 SDRAM 存入数据的同时,从另一块 SDRAM 读出数据,以保证数据的高速稳定传输。

2)USB 传输控制

本系统采用 KERNEL 软核通过判断端点 FIFO 的空、满标志位 (FLAGB、FLAGC) 来对 EZ-USB 端点 2、4、6、8 进行读、写控制,并采用异步 FIFO 读写控制模式,控制程序流程图如图 5 所示。

4.2 CPLD 程序设计

本设计中,CPLD 主要用于电压转换,其实现原理是通过 CPLD 的逻辑单元把 FPGA 对传感器的驱动信号做一个同步输入输出(从一个 I/O 口接收,送至另一个 I/O 输出),再通过外部硬件电路的配置,使其驱动电压升高或降低。其中,CPLD I/O 口的模块(推挽、开漏)通过 QUARTUS II 工程进行定义。

4.3 USB 固件程序设计

USB 固件是运行在 EZ-USB 芯片中的代码。SLAVE

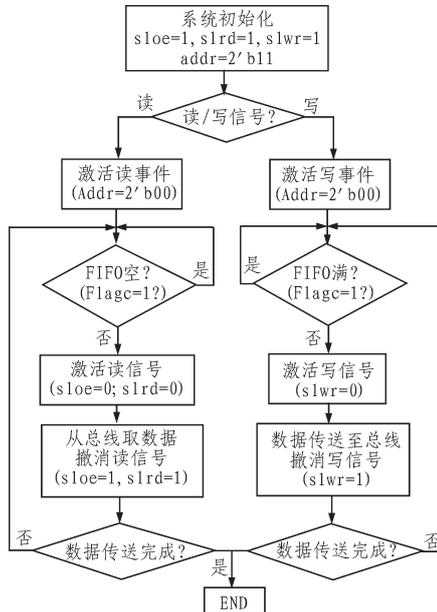


图 5 FPGA 异步 FIFO 读写控制流程图

FIFO 模式中,固件的功能配置了 SLAVE FIFO 相关的寄存器以及控制 EZ-USB 何时工作在 SLAVE FIFO 模式下,一旦固件将相关的寄存器配置完毕,FPGA 即可按照 SLAVE FIFO 的传输时序控制 EZ-USB 与主机进行高速通信,而在通信过程中不需要 EZ-USB 芯片中 8051 单片机的干预,从而保证足够的数据传输速率。固件程序采用 Cypress 公司提供的固件框架,添加自己的配置代码^[6]。添加的部分如下:

```
Void TD_Init(void)
{...
    IFCONFIG = 0x03; //设置外部时钟源、slave fifo 模式
    EP2CFG = 0xA2; //2 端点 OUT、批量传输、512 字节双缓冲
    EP4CFG=EP6CFG=EP8CFG=0xE2; // 4,6,8 端点 IN、批量传输、512 字节双缓冲
    AUTOPTSETUP |= 0x01 //使用自动指针
    ... }
```

在上面程序中,定义了 2 端点为上位机命令字下传通道,4、6、8 端点为采集数据上传通道,均设置为批量传输、512 字节双缓冲。

4.4 上位机应用程序设计

上位机应用程序的作用是提供一个人机交互的显示界面,进行实时数据交换,反映系统的运行状态并存储下位机的采集数据。该系统采用 Visual Studio 2013 进行上位机 C# 应用程序的设计,程序中采用 Cypress 公司提供的 CYAPI 控制函数类配合适当的驱动程序完成整个上下位机的通讯过程。

5 系统测试

本文对系统做了 3 个方面的测试,分别是驱动时序仿真、NIOS II 主控逻辑仿真和 USB 传输速度测试。

5.1 驱动时序仿真

以 SONY ILX554B CCD 为例,对传感器驱动时序的产生

