

FPGA 的高速多通道 数据采集控制器 IP 核设计

■ 厦门大学 潘梁垚 姚铭

摘要

介绍基于 FPGA 嵌入式系统的多通道高速数据采集模块控制器的 IP 核设计。采用 TI 公司的 6 通道同步采集 A/D 转换器件(ADS8364), 针对该器件使用硬件描述语言设计 IP 核, 实现对采集数据的处理, 同时设计了 IP 核与嵌入式系统的接口。在 Xilinx 公司的 ISE 开发工具中, 利用 FPGA 器件中的硬 FIFO 控制器辅助设计 IP 核, 利用嵌入式开发工具 EDK 建立 FPGA 嵌入式系统, 并添加和修改了用户自定义 IP 核, 通过仿真验证了该方法的实效性。

关键词 FPGA 数据采集 ADS8364 IP 核 FIFO

随着可编程逻辑器件的不断进步和发展, FPGA 在嵌入式系统中发挥着越来越重要的作用。本文介绍的在电能质量监测系统中信号采集模块控制器的 IP 核, 是采用硬件描述语言来实现的。首先它是以 ADS8364 芯片为控制对象, 结合实际电路, 将 6 通道同步采样的 16 位数据存储到 FIFO 控制器。当 FIFO 控制器存储一个周期的数据后, 产生一个中断信号, 由 PowerPC 对其进行高速读取。这样能够减轻 CPU 的负担, 不需要频繁地对 6 通道的采样数据进行读取, 节省了 CPU 运算资源。

1 ADS8364 芯片的原理与具体应用

A/D 转换芯片 ADS8364 是 TI 公司推出的专为高速同步数据采集系统设计的高速度、低功耗、6 通道(三相电压、三相电流)同步采样的 16 位 A/D 转换芯片。采用模拟和数字分别供电, 在模拟输入端, 有模拟参考电压输入、输出引脚和信号六通道正反相输入引脚; 在数字端, 主要包括控制 ADS8364 的读/写、复位、片选引脚和转换结果输出总线。

ADS8364 芯片的转换过程为: 当 ADS8364 的 $\overline{\text{HOLDX}}$ 保持至少 20 ns 的低电平时, 转换开始。当转换结果被存入输出寄存器后, 引脚 EOC 的输出将保持半个时钟周期的低电平, 以提示数据分析处理器进行转换结果的接收, 处理器通过置 $\overline{\text{RD}}$ 和 $\overline{\text{CS}}$ 为低电平可使数据通过并行输出总线读出。在转换数据的接收过程中, ADS8364 芯片各引脚工作的时序达到协调一致, 才能保证监测设备良好工作, 具体时序安排如图 1 所示。

ADS8364 芯片的数据输出方式分别由 BYTE、ADD

与地址线 A2、A1、A0 组合控制, 转换结果的读取方式由电能质量监测系统中采用的数据分析处理器决定, 一般可直接读取、循环读取和 FIFO 方式的任何一种。根据 BYTE 为 0 或者为 1 可确定每次读取时得到的数据位数, 根据 ADD 为 0 或者为 1 可确定第一次读取的是通道地址信息还是通道 A/D 转换结果。在实际应用中, 我们结合了 ADS8364 模数转换器中的 6 个 16 位 ADC 可以成对同步工作的能力, 3 个保持信号($\overline{\text{HOLDA}}$ 、 $\overline{\text{HOLDB}}$ 、 $\overline{\text{HOLDC}}$)可以同时被选通, 其转换结果将保存在 6 个寄存器中。对于每一个读操作, ADS8364 均输出 16 位数据, 最高位为符号位。根据图 2 所示的 ADS8364 循环读取方式工作时序, 需设置 BYTE 为 0, A2、A1、A0 分别为 1、1、0。

2 使用硬件描述语言设计数据采集控制模块

2.1 生成硬 FIFO 控制器

可编程逻辑器件采用的是 Xilinx 公司 Virtex-4 系列 FPGA, 在核生成器中可以产生硬 FIFO 控制器。主要生

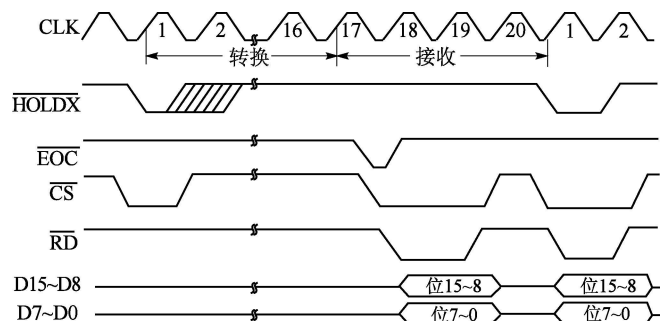


图 1 ADS8364 工作时序^[1]



成过程如下:

① 选择 FIFO 生成器,接着在 Read/Write Clock Domains 选项中选择异步 FIFO: FIFO 从 ADS8364 中存储数据时,写入时钟与 A/D 转换芯片的时钟(5 MHz)同步;当向 FIFO 中写入 1 个周期的数据后,CPU 根据 FIFO 发送的请求,用 100 MHz 系统时钟对数据进行读取。

② Memory Type: 选择使用内部的块 RAM。

③ 位宽与深度:A/D 转换芯片是 16 位,位宽设置为 16。由于电网频率是 50 Hz,且监测系统要求每个周期采样 256 点,故 A/D 芯片的采样频率为 12.8 kHz。每个周期 FIFO 要存储的数据是 256×6 (ADS8364 是 6 通道数据同步采集),所以深度可设置为大于 256×6 ,再根据需要设置 FIFO 空/满标志阈值。

完成设计,最后生成.v 原文件和.ngc 网表。FIFO 的数据输出宽度是 16 位,而 PowerPC 的数据总线宽度是 32 位,所以对 FIFO 的宽度进行扩展后并对其实例化,使其可被综合和仿真。

2.2 时钟分频设计

FPGA 的系统时钟频率是 100 MHz,A/D 转换芯片与一些控制模块需要 5 MHz 的输入时钟,所以需要时钟进行 20 分频。这里采用 Virtex 器件内置的 DCM(Digital Clock Manager,数字时钟管理)单元。DCM 中包含一个 DLL(Delay-Locked Loop,延迟锁定电路),可以提供对时钟信号分频功能,并且能够维持各输出时钟之间的相位关系即零时钟偏差。对系统时钟进行 10 分频,再 2 分频,这样可以节省分频寄存器。以下代码是时钟分频部分的 Verilog 描述:

```

BUFG CLKDV_BUF_INST (.I(CLKDV_BUF),.O(CLKDV_OUT));
IBUFG CLKIN_IBUFG_INST (.I(CLKIN_IN),.O(CLKIN_IBUFG));

```

```

BUFG CLK0_BUF_INST (.I(CLK0_BUF),.O(CLKFB_IN));
DCM_ADV DCM_ADV_INST (.CLKFB(CLKFB_IN),.CLKIN(CLKIN_IBUFG),
.DADDR(GND_BUS_7[6:0]),.DCLK(GND_BIT),.DEN(GND_BIT),
.DI(GND_BUS_16[15:0]),.DWE(GND_BIT),.PSCLK(GND_BIT),
.PSEN(GND_BIT),.PSINCDEC(GND_BIT),.RST(RST_IN),
.CLKDV(CLKDV_BUF),.CLKFX(),.CLKFX180(),.CLK0(CLK0_BUF),
.CLK2X(),.CLK2X180(),.CLK90(),.CLK180(),.CLK270(),
.DO(),.DRDY(),.LOCKED(),.PSDONE());

```

2.3 A/D 转换芯片控制模块及顶层文件的设计

控制器模块的设计:

① 根据 ADS8364 的工作原理: $\overline{\text{HOLDX}}$ 保持至少 20 ns 的低电平,转换开始,所以控制器需根据时序要求产生 $\overline{\text{HOLD}}$ 周期信号。

② 转换结束后根据 $\overline{\text{EOC}}$ 的响应状态,需要置 $\overline{\text{RD}}$ 和 $\overline{\text{CS}}$ 为低电平,使数据通过并行输出总线读出。下面是根据 $\overline{\text{EOC}}$ 的状态改变 $\overline{\text{RD}}$ 值的 Verilog 描述:

```

always @(negedge clk)
begin
if (EOC_n==0) count2<=0;
if (count2<=11)
begin
count2<= count2 + 1b1;
RD_n <= ~RD_n;
end
else RD_n <= 1;
end

```

根据图 1 的工作时序和图 2 的循环读取方式以及对数据采集频率(12.8 kHz)的要求,对芯片相应的引脚进行控制,并和 FIFO 进行连接使采集的数据能够按照循环方式写入 FIFO。采用 Verilog 硬件描述语言实现上述功能,

缩 略 语

BSB	Base System Builder	底层系统生成器
DCM	Digital Clock Manager	数字时钟管理
EPC	Electronic Product Code	电子产品编码
FFD	Full Function Device	完整功能设备
LIN	Local Interconnect Network	局部互连网络
ONS	Object Name Service	对象域名服务
OSAL	Operating System Abstraction Layer	操作系统抽象层
PLMN	Public Land Mobile-communication Network	公众陆地移动通信网
RFD	Reduced Function Device	简化功能设备
SAE	Society of Automotive Engineers	汽车工程师协会
TFT	Thin Film Transistor	薄膜晶体管

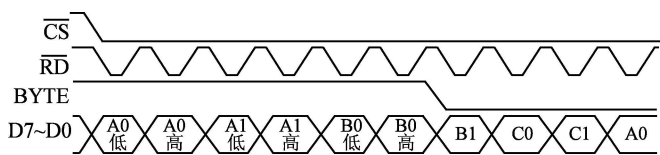


图2 循环读取方式工作时序^[1]

并建立顶层文件正确连接各个功能模块。

顶层文件的 Verilog 描述如下:

```
CLKDV clk_divide (.clk(clk), .clk_5m(clk_5m));
FIFO_ads8364 fifo
(.RD_clk_in(Bus2IP_Clk), .WR_clk_in(clk_5m),
 .RD_EN(fifo_RD_EN), .WR_EN(fifo_WR_EN),
 .Data_in(Data_in), .empty(empty), .full(full),
 .prog_full(prog_full), .Data_out(Data_out));
ads_ctrl ads_control (.clk(clk_5m), .EOC_n(EOC_n),
 .holdx_n(holdx_n), .RST_n(RST_n),
 .RD_n(RD_n));
```

如图3所示,时钟分频部分的输出与FIFO的数据写入时钟、AD_Ctrl的时钟和A/D转换芯片的时钟相连接。AD_Ctrl部分主要对ADS8364芯片进行控制,其中输出RD也连接到FIFO的写使能端,对FIFO的数据写入进行控制。FIFO的读时钟接到系统时钟,读使能由CPU控制。当FIFO写入一个周期的数据后,由prog_full产生中断信号,CPU响应并对FIFO进行读取。



图3 各功能模块连接示意图

2.4 仿真

对顶层文件进行综合,并在 Modelsim 中对其进行仿真。数据采集控制器的仿真结果如图4所示。当 holdx_n 为低电平时,启动 A/D 转换,完成后

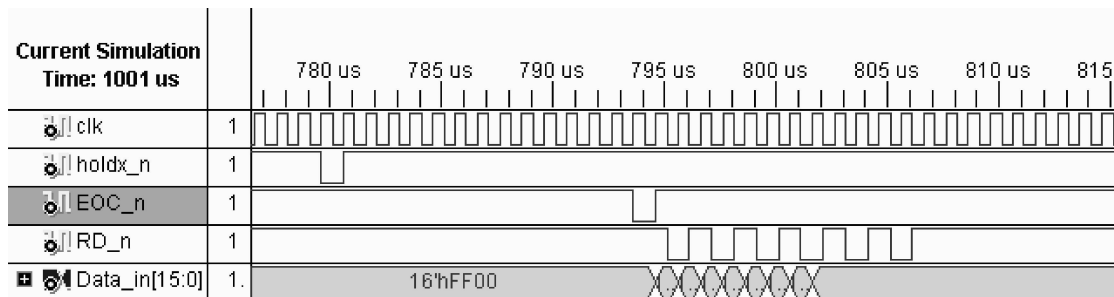


图4 A/D转换过程与数据读取仿真波形

根据 EOC_n 的低电平信号产生 6 个 RD_n 的低电平信号,循环读取数据。当 FIFO 存储了一个周期的数据后,CPU 置 FIFO 的读使能端口为高电平,对 FIFO 中的数据进行高速读取。若 FIFO 中数据为空,empty 为高电平。

3 使用 Xilinx 嵌入式开发工具 EDK 设计 IP 核

嵌入式开发软件 EDK 为设计人员提供了自动化设计向导——Base System Builder(BSB),可以指引工程师快速完成整个设计过程。使用 BSB 创建工程,在创建完成之后使用 EDK 自带的 CIP(Create and Import Peripheral Wizard)添加用户自定义 IP 核^[2],生成的用户 IP 核保存在 EDK 工程目录下的 pcore 文件夹。用户 IP 核目录如图5所示。

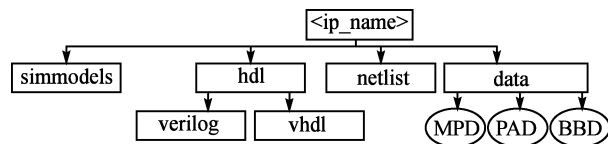


图5 用户 IP 核目录结构图

其中文件夹 data 用于存放用户 IP 的配置文件,如 .prj 文件、.mpd 文件和 .pao 文件等;文件夹 hdl 用于存放用户 IP 的 HDL 代码,即 .v 或者 .vhd 文件;而 dev1(simmodels)文件夹中的工程可以使用户在 ISE 平台对工程进行设计、综合与仿真,如果设计需要加入网表,可以放在 netlist 文件夹。CIP 在建立用户 IP 核时,使用了一种专用接口规范(IPIF)。IPIF 是一个验证并优化的高度参数化的定制接口,它提供了一个简化的总线协议 IPIC^[3](IP Interconnect)。操作这个总线与直接操作 PLB 及 OPB 这些总线相比要简单很多。通过 IPIF 模块,对其进行参数化定制来满足设计需求,将降低设计与测试的工作量。

将设计的 Verilog 文件复制到 IP 核目录下相对应的 hdl 文件夹下,启动 ISE 开发平台并打开 dev1 文件夹中的工程文件,在 Sources for Implementation 中显示的结构如图6所示。图中,adsfifo.vhd 是 IPIC 的描述文件,user_logic.v(或 user_logic.vhd)可以实现用户 IP 核功能设计。

需要在 adsfifo.vhd 中加入必要的端口声明与逻辑设

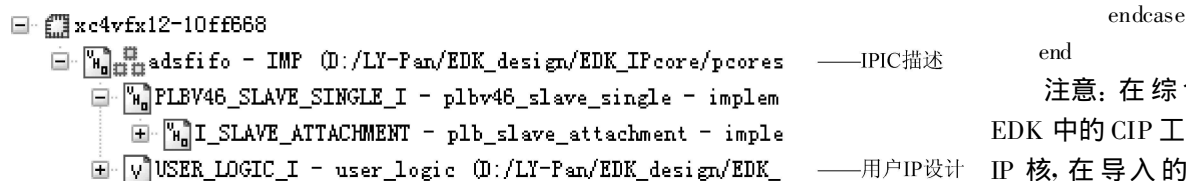


图6 文件结构

计,使 PLB 控制器与用户 IP 设计端口进行相应的连接。设计完成后在 ISE 平台中对该 IP 核进行综合并仿真。综合后查看 FPGA 器件的资源使用情况,如表 1 所列。

表 1 器件资源使用情况

使用的逻辑资源类型	已用/可用	利用率/(%)
Slice 资源	167/5472	3
Slice Flip Flops 资源	252/10944	2
4 input LUTs 资源	184/10944	1
FIFO16/ RAMB16s 资源	2/36	5
GCLKs 资源	4/32	12
DCM_ADVs 资源	1/4	25

根据需要修改 user_logic.v(或 user_logic.vhd),向其中添加端口声明与逻辑设计:

```
assign
    slv_reg_write_sel = Bus2IP_WrCE[0:0], fifo_RD_EN =
    Bus2IP_RdCE[0:0],
    slv_write_ack = Bus2IP_WrCE[0], slv_read_ack = Bus2IP_
    RdCE[0];
    fifo_WR_EN = ~RD_n;
always @(posedge Bus2IP_Clk)
    begin: SLAVE_REG_READ_PROC
        case (fifo_RD_EN)
            1'b1 : slv_ip2bus_data <= Data_out;
            default : slv_ip2bus_data <= 0;
```

加入相关联的.v 或.vhd 文件。导入完成后在 EDK 的 IP Catalog 的 Project Local pcores 分类中可以看到用户 IP 核,可以向 EDK 工程中加入该 IP 核,并设置其 Bus Interface、Port 和 Addresses 后生成位流文件,下载到开发板进行调试。

4 总结

利用 FPGA 和 ADS8364 设计的数据采集的 IP 核,其接口简单,采集精度高,可同时采集多路信号,而且能减轻 FPGA 嵌入式系统中 CPU 的负担,节省 CPU 的运算资源。经过仿真和下载到开发板验证,该设计能满足高速交变电压信号采集的高精度和高实时性的要求。

参考文献

- [1] 美国 TI 公司.ADS8364 数据工具手册,2006.
- [2] 赵峰.FPGA 上的嵌入式系统设计实例[M].西安:西安电子科技大学出版社,2008.
- [3] 孙航.Xilinx 可编程逻辑器件应用与系统设计[M].北京:电子工业出版社,2008.
- [4] Embedded System Tools Reference Manual[OL]. http://china.xilinx.com/ise/embedded/edk9_li_docs/est_rm.pdf.

潘梁焱(硕士研究生),主要研究方向为嵌入式系统。

(收稿日期:2009-02-18)

MIPS64 架构推动 Cavium Networks 新型 OCTEON II 处理器

MIPS 科技公司(MIPS Technologies, Inc)宣布 Cavium Networks 公司正式推出的新型 OCTEON II 因特网应用处理器(IAP)中,采用了 MIPS 科技的高性能 MIPS64 架构。OCTEON II 系列采用多达 32 个 MIPS64 内核,是 Cavium 基于 MIPS64 架构的可扩展多核 OCTEON 处理器成功产品线的最新产品,并创下了在一个芯片中采用 MIPS64 处理器内核数目的新纪录。

OCTEON II 处理器是专为需要支持新一代融合数据、音视频的“超级网络”(hyper network)设计的,适用于企业、数据中心、接入和服务提供商市场。OCTEON II 处理器的目标应用包括 3G、WiMAX、LTE 和无线网络,以及统一存储系统和适配器的交换机、路由器、装置和设备等。

OCTEON II 系列集成了 1 至 32 个 MIPS64 内核,以及多达 75 个用于服务质量、封处理、TCP、压缩、加密、RAID、重复数据删除(de-duplication)及正则表达式(regular expression)处理的应用加速引擎。此外,还有高达 400 Gbps 的 DDR3 存储器带宽、高达 100 Gbps 的网络连接,而整个系列的耗电仅有 2~60 W。