

DSP 和 FPGA 并行通信方法研究

姜楠 马迎建 冯翔

(北京航空航天大学仪器科学与光电工程学院 北京 100083)

摘要: 本文介绍了一种 DSP 与 FPGA 之间的并口通信方法,它实际应用于陀螺数字检测电路。此方法使用 TMS320F2812 芯片的外部数据总线接口(XINTF)与 FPGA 相连接,利用它的中断和读写选通信号(XRD 与 XWE)作为判断位与使能位完成通信。文中给出了它们的硬件连接方式和通信部分的软件程序,并用 modelsim 对 FPGA 通信程序进行了仿真。通过仿真结果以及在实际电路中的实验调试,表明此方法可以实现 DSP 与 FPGA 之间的并行通信,能有效提高通信速度。

关键词: DSP; FPGA; 外部接口; 并行通信; modelsim

中图分类号: TP2 **文献标识码:** A

Research on parallel communication between DSP and FPGA

Jiang Nan Ma Yingjian Feng Xiang

(School of Instrument Science & Opto-Electronic Engineering, Beihang University, Beijing 100083)

Abstract: Method of communication between DSP and FPGA was introduced, which used in the gyroscope digital detection circuit. The external interface(XINTF) data bus of the TMS320F2812 chip is connected with FPGA in the method. The interrupt signals of the DSP are used as judgment signals, and the read-write select (XRD and XWE) signals are used as enable signals to complete communication. The hardware interface circuit between DSP and FPGA and the software program of communication part are introduced. Then the FPGA communication program is simulated by modelsim. Through the result of the simulation and the experiment debugging in the real circuit, it is indicated that the method can achieve communication between DSP and FPGA, and the speed of communication is improved availably.

Keywords: DSP; FPGA; External Interface; parallel communication; modelsim

0 引言

在航空领域中,陀螺在导航中起着重要作用。随着数字化陀螺精度的提高,所需要处理的数据速度和数据量也越来越高。DSP 与 FPGA 之间的并口通信相对于串口通信更能满足速度的要求。本文基于数字化陀螺信号检测系统,选用 TI 公司的 TMS320F2812(以下简称为 DSP)和 XILINX 公司 Virtex-II 系列的 XC2V250(以下简称为 FPGA)芯片搭建硬件平台,实现 16 位并行总线的通信,从而提高了通信速度,并简化电路,节约成本,使得系统易于修改和升级^[1]。

1 设计实现

1.1 设计思想

在此系统中,DSP 作为主要运算芯片,是实现算法的核心;而 FPGA 作为它的外围器件,可以看成硬件设计的一种延续。信号的流程图如图 1 所示^[2]。

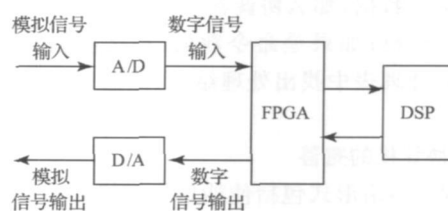


图 1 信号流程图

模拟信号通过高频率 A/D 转换后,转成高速的数字信号发送给 FPGA,在 FPGA 中进行累加求平均后,将较低频率的信号传送给 DSP,在 DSP 中进行核心算法的处理后,再把信号发回给 FPGA。通过 FPGA 把发回的信号进行逻辑组合后通过 D/A 转换成模拟信号输出。可以看出 FPGA 和 DSP 的通信是数据传输的核心部分。其中,DSP 的输入和写出采用中断控制,当 FPGA 准备好写数后,向 DSP 发送读中断,DSP 响应中断后读取数据。当 FPGA 准备好读数时,也向 DSP 发送中断请求,DSP 就会把处理好的数据放在总线上,等待 FPGA 读取。这样就能够完成两

个芯片的并行通信。

1.2 硬件部分连接

TMS320F2812 型 DSP 芯片有 16 位 XINTF 数据线^[3], 为系统外部接口, 它可以与各种外部存储器或者设备实现无缝连接。在此系统中, 将它与 FPGA 的 16 个用户可定义 I/O 管脚相连, 用于实现 16 位并行数据的通信。TMS320F2812 芯片还有 56 个可编程通用输入/输出引脚 (GPIO) 引脚^[4], 任选两个和 FPGA 的用户 I/O 管脚相连, 作为 DSP 的读写中断。

1.3 软件部分实现

1.3.1 通信中的时钟设计

本系统中, DSP 所使用的中断是 CPU 定时器定时给出的中断^[5], CPU 定时器在初始化阶段设计为每隔 $1\mu\text{s}$ 产生一个中断发给 CPU, 使其访问中断子程序, 判断中断标志位是否为 1。如果为 1, 则读数或写数; 如果为 0, 则继续循环等待。因此为了保证数据读写的正确, 既不丢数, 也不重复读数, 就要使每个运算周期只判断到一次标志位为 1。这样使 FPGA 发来的中断长度也为 $1\mu\text{s}$ 即可。

在 DSP 收到中断以后, 会从 FPGA 读或者写数据。为了数据读写的准确, TMS320F2812 芯片的外部接口区 (XINTF) 的时钟也是可以编程控制的^[6]。每个 XINTF 区的读或者写的时序都可以被分成 3 个阶段: 前导 (Lead) 有效 (Active) 和跟踪 (Trail), 3 个阶段的时间可以分别配置。在有效阶段里会实现对外部设备的访问。如果是读操作, 则读选通信号 (XRD) 被拉低, 同时将数据锁存进 DSP; 如果是写操作, 写选通信号 (XWE) 被拉低, 同时将数据送到数据总线上。因此在系统中以 XRD 和 XWE 分别作为 DSP 发送给 FPGA 的读写使能信号。

1.3.2 FPGA 收发模块设计

FPGA 的软件编写采用 Verilog HDL 进行描述。由于和 DSP 之间通信是双向的, 因此和总线相连的端口设计成 inout 型^[7]。数据的读入和写出是由时钟控制读写标志位来完成的, 时钟同样控制着发送给 DSP 的读中断和写中断。以下给出此通信部分 Verilog HDL 程序, 为了程序简明易读, 其中部分端口定义和变量定义程序被省略, 只列出了与通信有关的定义:

```
module fpga_rw (
    inout [15:0] data_inout;
    output [15:0] data_out;
    output int_flag1, int_flag2;
    wire [15:0] XD_out;

    assign XD_out = data_out;
    assign data_inout = inout_flag? XD_out : 16'hzzzz;
    always @(posedge clk_50M) // A/D 采样的时钟
    begin
        j <= j + 1;
        if (j >= 3 & & j <= 103)
```

```
begin
    if (j == 3)
        int_flag1 <= 1; // 写中断的标志位

    else if (j == 56)
        int_flag1 <= 0;
        if (XRD == 0) // DSP 读使能位有效

    时发数
        data_out <= data_sum [19:4];

    // FPGA 写数

end
else if (j >= 250 & & j <= 300)
begin
    if (j == 250)
        int_flag2 <= 1; // 读中断的标志位
    else if (j == 301)
        int_flag2 <= 0;
        if (XWE == 0) // DSP 写使能位有效

    时收数
        XD_in <= data_inout; // FPGA 读数

end
else if (j == 399)
    j <= 0;
end
always @(posedge clk_50M)
begin
    if (j == 0)
        inout_flag <= 1; // 数据读写标志位
    else if (j == 149)
        inout_flag <= 0;
end
```

1.3.3 DSP 读写数据模块设计

DSP 主程序运行起来后开始空循环, 等待写中断的到来。当 FPGA 发来写中断时, 说明 FPGA 已经把数据准备好放在总线上了。此时, DSP 将数据从数据总线读入, 然后就可进行后续的运算处理了。数据经过一定的算法处理后, 等待读中断的到来。当收到 FPGA 的读中断信号后, DSP 就把处理好的数据输出到数据总线, 由 FPGA 读入。下面给出 DSP 的通信部分程序:

```
for (;)
{
    if (flag_in == 1) // 判断读中断标志位
    {
        data_reg = data_in; // DSP 读入数据
        flag_in = 0;
        for (;)
        {
            if (flag_out == 1) // 判断写中断标志位
```

```

{ *FPGA_ADD=0x00fe; //数据写入 FPGA
  flag_out=0;
  break; }
}
};

```

下面是 TMS320F2812 的中断程序:

```

interrupt void cpu_timer0_isr(void)
{
  if( GpioDataRegs.GPADAT.bit.GPIOA11== 1)
    //判断读中断的到来
    {data_in= *FPGA_ADD; //从地址总线上读入数据
      flag_in=1;}
  if( GpioDataRegs.GPADAT.bit.GPIOA13== 1)
    //判断写中断的到来
    flag_out=1;
}

```

2 仿真实验设计

程序完成后为验证通信的正确性使用 modelsim 进行

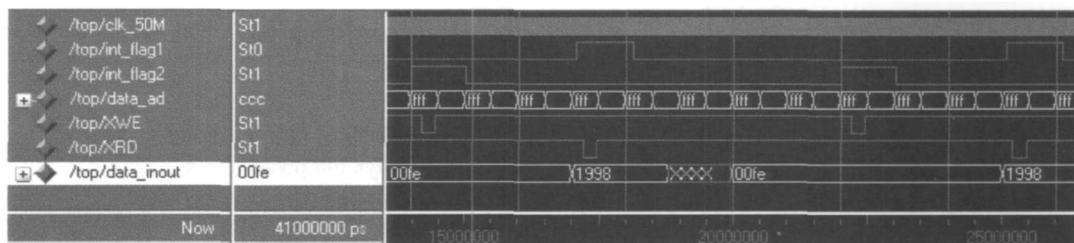


图 3 modelsim 仿真波形图

在数据线 data_inout 上, 16 进制数 1998 为 FPGA 模块发送给 DSP 模块的数据, 而 00fe 为 DSP 模块发送给 FPGA 模块的数据。由此波形图可以看出仿真结果可以完成 DSP 与 FPGA 之间的数据并行通信。

3 结 论

本文介绍了 DSP 和 FPGA 之间通过数据总线实现双向并行通信的方法。通过在实际的陀螺硬件电路板上调试, 已经可以完成数据的并行通信, 有效提高了数据传输的速度。此方法为以 DSP+FPGA 为硬件平台的系统提供了并行通信的参考方案。

参 考 文 献

- [1] 汉泽西, 孙燕妮. DSP+FPGA 技术[J]. 电子技术, 2007, 2: 18-21.
- [2] 孟德刚, 何国瑜. 基于 FPGA 数据采集系统[J]. 电子测量技术, 2004(5): 74-75.
- [3] Texas Instruments. TMS320x281x External Interface

了仿真实验。仿真程序编了 3 个模块: A/D 测试模块, FPGA 模块和 DSP 测试模块。其中 A/D 测试模块和 DSP 测试模块为行为模块, 分别用来模拟 A/D 发送信号和 DSP 收发数据; FPGA 模块为可综合模块。在 3 个模块以上, 还有一个顶层模块, 用于连接 3 个模块, 全面测试^[8]。模块的连接示意图如图 2 所示。

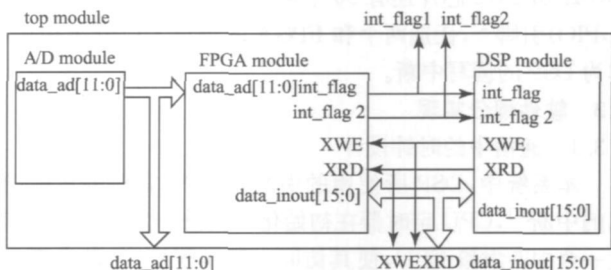


图 2 modelsim 仿真模块连接图

其中 int_flag1 和 int_flag2 分别为写中断和读中断, XWE 和 XRD 写使能位和读使能位, data_ad 为模拟 A/D 发出的数据, data_inout 为 16 位数据总线。通过连接到顶层模块的数据线可以观察到数据的仿真结果。仿真波形如图 3 所示。

(XINTF) Reference Guide[Z]. 2004: 9.

- [4] 徐科军, 张瀚, 陈智渊. TMS320x2812x DSP 原理与应用[M]. 北京: 北京航空航天大学出版社, 2006: 3.
- [5] 万山明. TMS320F281x DSP 原理及应用实例[M]. 北京: 北京航空航天大学出版社, 2007: 56-60.
- [6] 袁帅, 佟为明, 李中伟. TMS320F2812 外部接口分析与存储器扩展[J]. 单片机与嵌入式系统应用, 2007(8): 69-71.
- [7] 莫海勇, 张申科. FPGA 中双向端口 I/O 的研究[J]. 国外电子测量技术, 2005(6): 49-51.
- [8] 夏宇闻. Verilog 数字系统设计教程[M]. 北京: 北京航空航天大学出版社, 2003: 177-200.

作 者 简 介

姜楠, 男, 1983 年 1 月出生, 工学硕士, 主要研究方向为谐振陀螺的信号检测。

E-mail: jiangnan830119@163.com