

基于FPGA的RISC微处理器的设计与实现

Design and Implement of RISC Microprocessor Based on FPGA

李强,潘明,许勇

Li Qiang, Pan Ming, Xu Yong

(桂林电子工业学院计算机科学系, 广西桂林 541004)

(Dept. of Comp. Sci., Guilin Univ. of Elec. Tech., Guilin, Guangxi, 541004, China)

摘要:基于FPGA和电子设计自动化技术,采用模块化设计的方法和VHDL语言,设计一个基于FPGA的RISC微处理器。该微处理器主要由控制器、运算器和寄存器组成,具有指令控制、操作控制、时间控制和数据加工等基本功能,其指令长度为16位定长,采用立即寻址和直接寻址两种方式。仿真结果表明,基于FPGA的RISC微处理器的时钟频率为23.02MHz,且功能完全达到设计要求。

关键词:RISC微处理器 电子设计自动化 现场可编程门阵列 VHDL语言

中图分类号:TP302 文献标识码:A 文章编号:1002-7378(2005)04-0294-04

Abstract: A RISC microprocessor is developed by using modular design method and VHDL language based on FPGA and EDA technology. The microprocessor mainly consists of controller, operator and registers, and have the basic functions of instruction control, operation control, time control and data process. Its instruction length is 16 bits. Immediately addressed and directly addressed are adopted. Its clock frequency reaches 23.02MHz and its function is satisfied.

Key words: RISC microprocessor, electronic design automation, field programmable gate array, VHDL language

20世纪80年代初兴起的RISC技术一直是计算机发展的主流^[1],RISC微处理器的一些基本理论则是计算机领域的重要基础常识,但具体实现仍有难度。电子设计自动化(Electronic Design Automation,简称EDA)是现代电子设计的核心技术。利用EDA技术进行电子系统设计的主要目标是完成专用集成电路(ASIC)的设计,而现场可编程门阵列(FPGA)和复杂可编程逻辑器件(CPLD)是实现这一途径的主流器件^[2]。现场可编程通用门阵列(Field Programmable Gate Array,简称FPGA)的内部具有丰富的可编程资源。FPGA外部连线很少、电路简单、便于控制。FPGA目前已达千万门标记(10 million-gate mark),速度可达200~400MHz。本文介绍了一种基于FPGA技术用VHDL(VHSIC Hardware Description Language)语言实现的8位RISC微处理器,并给出了仿真综合结果。

收稿日期:2005-06-18

修回日期:2005-07-30

作者简介:李强(1979-),男,硕士研究生,主要从事单片机应用和EDA技术研究。

1 微处理器功能、组成及指令集

本文设计的RISC微处理器遵循了RISC机器的一般原则:指令条数少而高效、指令长度固定、寻址方式不超过两种、大量采用寄存器、为提高指令执行速度、指令的解释采用硬连线控制等等^[3]。

1.1 RISC微处理器的功能和组成

微处理器是整个计算机系统的核心,它具有如下基本功能:指令控制、操作控制、时间控制、数据加工。本文设计的微处理器主要由控制器、运算器和寄存器组成,还包括程序计数器、译码器等一些其他的必要逻辑部件。控制器是发布命令的“决策机构”,即完成协调和指挥整个计算机系统的操作。相对于控制器而言,运算器接受控制器的命令而进行动作,即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的,所以它是执行部件。存储器是记忆设备,存储单元长度是8位,用来存放程序和数据。

1.2 微处理器的指令集

微处理器的指令长度为16位定长,每条指令占

两个存储单元,寻址方式仅有立即寻址、直接寻址两种。该RISC微处理器选取了使用频度较高的8种指令LDA、STO、JMP、ADD、AND、XOR、SKZ、HLT等。指令操作码占用指令字的高4位,预留了空间,便于以后指令集的扩展。指令周期是由8个时钟组成,每个时钟都要完成固定的操作。部分典型指令的操作流程图如图1所示。

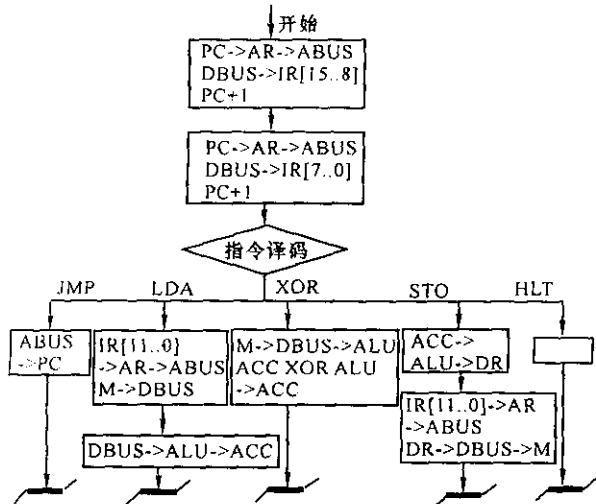


图1 部分指令操作流程

2 关键模块的设计

RISC微处理器是一个复杂的数字逻辑电路,但其基本部件的逻辑并不复杂,可以把它分为时钟产生器、指令寄存器、累加器、算术逻辑单元、数据控制器、状态控制器、程序计数器、地址多路器等单元来考虑。在硬件验证时还需要建立一些如ROM/RAM和地址译码器等必要的外围器件。以下是几个关键模块的设计。

2.1 时钟产生器的设计

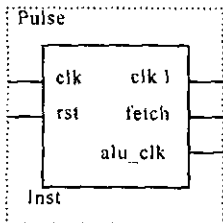


图2 时钟产生器电路的设计

时钟产生器Pulse产生的电路如图2所示,计算机的协调动作需要时间标志,它用时序信号体现,时钟产生器正是产生这些时序信号的器件。图2中时钟产生器利用外来时钟信号clk产生一系列时钟信号clk1,fetch,alu_clk等,并送往微处理器的其他部件。rst控制着微处理器的复位和启动操作,当rst一进入高电平,微处理器就结束现行操作,并且只要

rst停留在高电平状态,微处理器就维持复位状态。rst回到低电平后在接着到来的fetch上升沿启动微处理器开始工作。

由于时钟产生器对微处理器各种操作实施时间上的控制,所以其性能好坏从根本上决定了整个微处理器的运行质量。本设计采用的同步状态机的设计方法,使得clk1,fetch,alu—clk在跳变时间同步性能上有显著提升,为整个系统性能的提高打下良好的基础^[3]。

2.2 状态控制器的设计

状态控制器的电路图如图3所示。从实现的途径看,RISC微处理器与一般的微处理器的不同之处在于,它的时序控制信号的形成部件是用硬布线逻辑实现而不是采用微程序控制。由于器件本身设计比较复杂,且对各个控制信号的时序有严格要求,所以其VHDL程序用有限状态机FSM来实现。

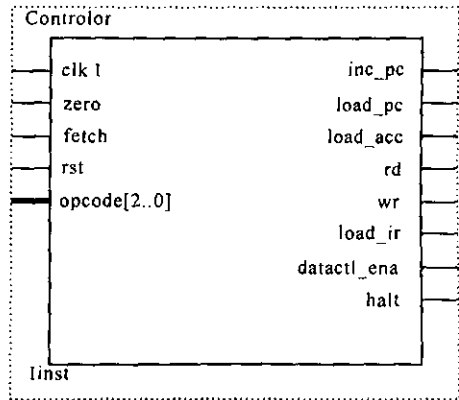


图3 状态控制器电路

结构体程序如下:

```
architecture rtl of statctl is
    type mystate is (st0, st1, st2, st3, st4, st5, st6, st7);
    signal cur_state : mystate;
begin
    process(clk1,ena)
    begin
        if falling_edge(clk1) then
            if(ena='0') then
                cur_state<=st0; inc_pc<='0';
                load_acc<='0'; load_pc<='0';
                rd<='0'; wr<='0'; load_ir<='0';
                datactl_ena<='0'; halt<='0';
            else
                case cur_state is
                    when st0 => .....
                    when st1 => .....
                    when st2 => .....
                end case
            end if
        end if
    end process
end architecture
```

```

when st3 => cur_state<=st4;
if (opcode=halt) then .....
    else .....
        when st4 => cur_state<=st5;
        if(opcode=jmp) then .....
        elsif(opcode=add or opcode=ann
or opcode=xoo or opcode=lda) then
            elsif(opcode=sto) then ... else ...
            when st5 => cur_state<=st6;
            if(opcode=add or opcode=ann or
opcode=xoo or opcode=lda)
then.....
                elsif(opcode=skz and zero='1')
then .....
                    elsif(opcode=jmp) then .....
                    elsif(opcode=sto) then .....
                    else .....
                    when st6 => cur_state<=st7;
                    if(opcode=sto) then .....
                    elsif(opcode=add or opcode=ann
or opcode=xoo or opcode=lda)
then .....
                        else .....
                        when st7 => cur_state<=st0;
                        if(opcode=skz and zero='1') then .....
                        else .....
                        when others => .....
                        end case;
                    end if;
                end if;
            end process;
        end rtl;
    
```

2.3 算术逻辑单元ALU的设计

ALU 是绝大多数指令必须经过的单元,所有的运算都在算术逻辑单元ALU进行。ALU接受指令寄存器IR送来的4位指令操作码,根据不同的指令,ALU在信号alu-clk的正跳变沿触发下完成各种算术逻辑运算。微处理器各部件结构如图4所示。

3 软件综合与仿真和硬件实现

3.1 微处理器的软件综合与仿真

该微处理器设计共有11个基本模块,除前文分析的3个模块外,还有指令寄存器IR、累加器ACC、程序计数器PC、简单的存储器ROM/RAM、地址多路器ADDR等模块。所有的模块采用Quartus II 4.2单独综合,并调试通过,且都生成有单独的*.bdf文件,最后创建一个顶层文件top.bdf,把所有基本

模块的bdf文件连接成如图4的形式。做完顶层设计

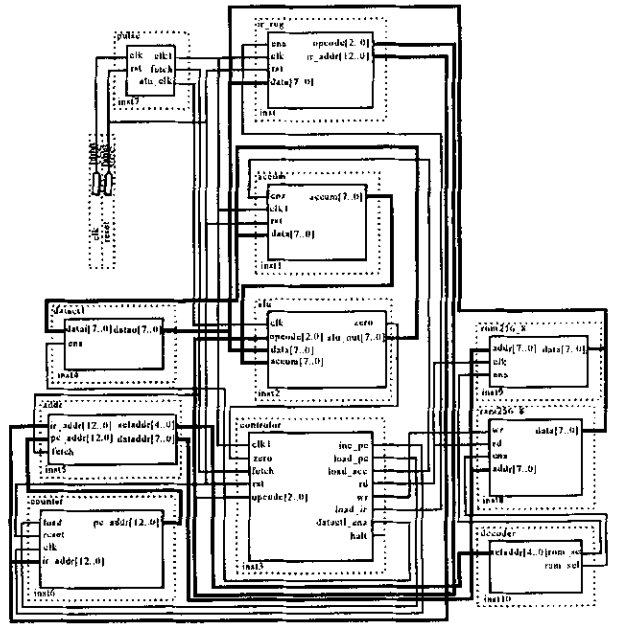


图4 微处理器各部件的结构

后,采用Quartus II 4.2进行综合与仿真。系统仿真的部分结果如图5所示,从图5可以看出,存放在存储器不同地址中的2个操作数3CH(00111100)和18H(00011000)相异或时,结果24H(00100100)在信号wr的上升沿触发下存入存储器中。我们可清楚地看到每条指令都是在一个指令周期中完成。数据总线data上记录着指令的运行情况,同时也可看到空闲时其呈高阻状态。

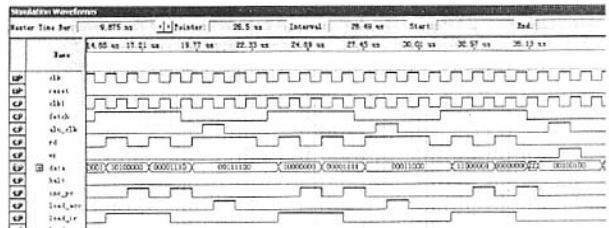


图5 仿真波形

主要的程序如下:

地址	机器代码	汇编语言源程序
00	11000000	JMP L1 ;L1->PC
01	00000100	
04	10100000	LDA R2 ;(0E)->R2
05	00001110	
06	10000000	XOR R1,R2 ;(R1)xor(R2)->(R1)
07	00001111	
08	11000001	STO ;(R1)->(0F)
0A	00000000	HLT ; stop
0B	00000000	

0E 00111100

0F 00011000

3.2 微处理器的硬件实现

基于FPGA的RISC微处理器的最终硬件验证在杭州康芯公司生产的GW48EDA系统上进行。前面的仿真结果确认无误后,选用GW48EDA系统的电路模式No. 5,查阅此系统的引脚对照表锁定各引脚,之后需重新编译一次,以便把引脚锁定信息编译进编程下载文件。最后把编译好的top.sof文件对目标器件FPGA下载,得到满足设计要求的芯片。本设计的载体选用Altera公司的Cyclone系列FPGA器件EP1C6Q240C6,硬件验证结果表明,该RISC微处理器时钟频率为23.02MHz,其功能完全达到设计要求。

4 结束语

本文基于FPGA的微处理器具备了RISC微处

理器的基本功能,而且其容易优化升级。该微处理器不仅可作为一个模块用于片上系统的设计,而且也充分展示了使用FPGA和VHDL进行EDA数字系统设计的优越性,具有实用价值。

参考文献:

- [1] 白中英. 计算机组成原理[M]. 第3版. 北京:科学出版社,2002.
- [2] 潘松. EDA技术实用教程[M]. 北京:科学出版社,2002.
- [3] 夏宇闻. Verilog数字系统实际教程[M]. 北京:北京航空航天大学出版社,2003.

(责任编辑:黎贞崇)

(上接第293页)

步骤5:令 $s = b + 1$ 。

步骤6:如果 s 为最后一个交易日,转到步骤9。

步骤7:如果在 s 交易日,在SELLCOND中发现卖出信号,进入步骤9。

步骤8:如果SELLFUN不为空时,用 s 交易日的股票数据执行SELLFUN中的函数,只要有一个函数返回真值,进入步骤9。否则令 $s = s + 1$,然后转到步骤6。

步骤9: b, s 分别代表买入/卖出日期,然后由 b 和 s 获取股票交易的买入卖出价格,将这些代表交易行为的数据作为一条记录添加到trades交易行为数组中。

步骤10: $a = s + 1$ 。

步骤11:转到步骤3。

4 实例分析

运用系统对上述的交易方案进行评估,测试数据选定10支流通量大、声誉较好,并且不易被操纵的股票的数据,时间从2002年1月1日至2004年1月1日,股票分别是深发展、中国石化、四川长虹、宝钢股份、上海汽车、中兴通信、五粮液、万科A、上海医药、隆平高科。买入手续费设为0.35%,卖出手续费设为0.35%。经过系统评估后,系统生成多个可直接观看的分析报告。其中有总体分析报告,个股分析报告,个股对比分析报告,亏损汇总报告,盈利汇总

报告等。下面列出总体分析报告的部分内容:平均利润率为0.40%,最高赢利为60.86%,单笔最高亏损为-11.13%,总交易次数为78,亏损笔数为58,赢利笔数为20。这些数据为投资者的投资决策提供了良好的参考依据。

5 结束语

从应用的情况看,本系统可以达到比较好的效果,整个系统使用方便,规则描述语言易于理解和使用,产生的分析报告准确、详实和实用,可以作为投资分析的良好工具。系统不足之处是分析速度稍慢,分析报告不支持图表方式。下一步的目标就是要克服以上提到的两个不足。

参考文献:

- [1] 波涛. 系统交易方法[M]. 北京:经济管理出版社,1998.
- [2] The MathWorks Inc. Financial Time Series Toolbox User's Guide[K]. The Math Works, Inc, 2000.
- [3] 财富网络公司. 大参考 III 使用手册[K]. 财富网络公司, 2003.
- [4] 世华国际金融信息公司. 世华期货股票工作站 2.19 版使用手册[K]. 世华国际金融信息公司, 2003.
- [5] 王沫然. MATLAB 6.0 与科学计算[M]. 北京:电子工业出版社, 2001.

(责任编辑:韦廷宗)