

# 基于FPGA的开放式CPU实验模型设计

康磊, 杨朝辉

(西安石油大学 计算机学院, 陕西 西安 710065)

**摘要:**在计算机组成原理的教学中,CPU设计实验能够使学生深入理解和掌握计算机的运行和控制机制。该文提出了一种基于FPGA的CPU开放式实验平台的设计思想,该平台由上位机和硬件开发平台组成。通过开放式实验平台,学生可以根据自行选取的MIPS指令集修改模型机的数据通路和控制通路并设计控制器,通过实验平台观察运行过程和结果以确定设计的正确性。

**关键词:**CPU实验模型;数据通路;控制通路;MIPS指令集;FPGA

**中国分类号:**TP302 **文献标识码:**A **文章编号:**1009-3044(2017)30-0235-03

**DOI:**10.14004/j.cnki.ckt.2017.3407

## Design of Open CPU Experiment Model Based on FPGA

KANG Lei, YANG Zhao-hui

(School of Computer, Xi'an University of Petroleum, Xi'an 710065, China)

**Abstract:** CPU design experiment in the teaching of computer composing principle can make student understand and master the mechanism of computer operation and control deeply. This paper presents a design idea of an open CPU experiment platform Based on FPGA, which is composed of the host computer and the hardware development platform. Through the open experiment platform, the students can modify the data path and control path of the model machine and design the control unit according to their own MIPS instruction set, and observe the running process and the result to ensure the correctness of the design through the experimental platform.

**Key words:** CPU experimental model; Datapath; Controlpath; The MIPS instruction set; FPGA

由于中央处理器(CPU Central Processing Unit)在现代计算机系统中占有核心功能,对于计算机专业的课程体系,CPU的设计开发是一个重点和难点。完成CPU的设计需要学生学习计算机的组成原理,系统结构等,并以此设计出CPU系统。而在计算机组成原理的CPU设计试验中,目前国内高校普遍使用的是教学系统或者小规模集成电路和手工布线方式,这样只能进行验证性实验,或者实验的规模有限,过于琐碎的操作耗费设计人员的精力,无法提供自住设计和发挥的平台。

本文通过研究和设计基于FPGA的32位开放式CPU试验模型,同时设计开发出相关的验证调试系统,为设计人员提供了一套开放式,可自主设计的CPU设计实验环境。设计人员通过该开放式CPU模型能直观感受CPU的工作原理和各部件的功能,根据自己的设计更新该实验模型的相关部件,并设计自己的数据通路和控制通路来实现自己的设计。通过调试系统设计人员可以监控CPU内部运行情况,对设计的CPU进行运行调试验证,完成CPU的设计和实现验证。

## 1 CPU功能和结构设计

### 1.1. CPU的指令系统

指令系统作为CPU的对上接口,定义了CPU的功能,并直接影响了CPU的实现。MIPS作为精简指令集(RISC Reduced

Instruction Set Computer)的代表,有着指令定长,结构简单的特点。选择MIPS指令集可以减少指令设计难度。本文的指令基于MIPS指令集进行设计,存储系统采用哈佛结构。

MIPS指令分为立即数类型(I类型),跳转类型(J类型),和寄存器类型(R类型)三种。本文设计的模型机选取了如表1所示的10条指令。这10条指令分别属于算术逻辑运算类,访存类和跳转类,可以满足程序设计的基本要求。

表1 选取的10条指令

类别	指令助记符	描述
运算类	addu rd, rs, rt	$GPR[rd] \leftarrow GPR[rs] + GPR[rt]$
	subu rd, rs, rt	$GPR[rd] \leftarrow GPR[rs] - GPR[rt]$
	and rd, rs, rt	$GPR[rd] \leftarrow GPR[rs] \text{ AND } GPR[rt]$
	ori rt, rs, imm	$GPR[rt] \leftarrow GPR[rs] \text{ OR } imm$
访存类	lw rt, offset(base)	$GPR[rt] \leftarrow mem[GPR[base] + offset]$
	sw rt, offset(base)	$mem[GPR[base] + offset] \leftarrow GPR[rt]$
跳转类	beq rs, rt, offset	if( $GPR[rs] == GPR[rt]$ ) then 转移
	bne rs, rt, offset	if( $GPR[rs] \neq GPR[rt]$ ) then 转移
	jr rs	$PC \leftarrow GPR[rs]$
	jal target	$PC \leftarrow PC31..28    instr\_index    0^2$ $GPR[31] \leftarrow PC + 4$

收稿日期:2017-09-10

作者简介:康磊(1968—),女,陕西省西安市人,副教授,硕士,主要研究方向为计算机体系结构、嵌入式系统;杨朝辉(1991—),男,硕士研究生,主要研究方向为计算机系统结构。

本栏目责任编辑:梁书

计算机工程应用技术 235

## 1.2. 数据通路设计

在一条指令周期中,一条指令的执行大致可以分为取指,分析,执行几个步骤。这些步骤是由一个个模块的工作实现的。而数据以及各个模块的控制信号通过在模块之间的流动完成了指令的执行,所形成的数据传送的路径称为数据通路(DP Datapath)。

对于不同的指令经过相同步骤所进行的操作也可能不同。需要对每一条指令的功能和进行的操作进行分析。以指令无符号加addu的执行为例,addu指令所执行的操作: $GPR[rd] \leftarrow GPR[rs] + GPR[rt]$ ,指令编码为:

31..26	25..21	20..16	15..11	10..6	5..0
Special	rs	rt	rd	0	addu
001001				00000	100001

根据addu要完成的操作,分析其执行需要经过以下步骤:  
取指令步骤:

$IR \leftarrow IM[PC]; PC \leftarrow PC + 4;$

接着译码步骤:

$rs \leftarrow IR[25:21]; rt \leftarrow IR[20:16]; rd \leftarrow IR[15:11];$

执行步骤:

$A \leftarrow R[rs]; B \leftarrow R[rt]; C = A + B; R[rd] \leftarrow C;$

根据以上分析,可以得出addu的数据通路如图1所示:

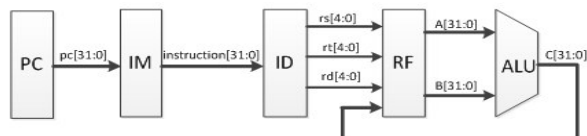


图1 addu指令的数据通路

以此类推,分析出其他指令的执行步骤和数据通路。因为不同指令执行时需要进行不同的操作,模块在复用时也需要提供正确的数据源。因此我们增加多路选择器来为模块选择正确的数据源。同时增加控制器(CU Control Unit)来控制各个模块执行的操作。10条指令的数据通路如图2所示。

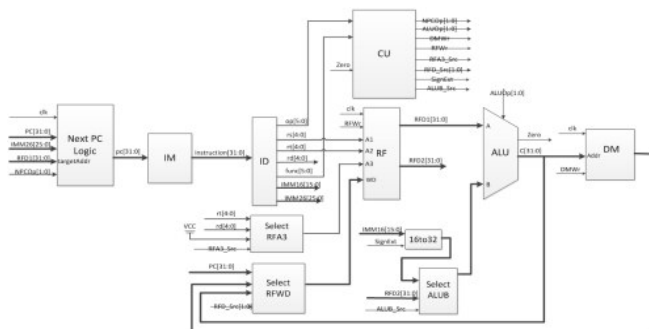


图2 CPU模型机的数据通路

由于增加了指令,使用FPGA实现时,需要更新原有的模块来满足10指令的需求。同时增加控制器,多路选择器等模块,并根据图2建立对应的数据通路,以及控制通路完成10条指令的CPU模型实现。

基于此开放式CPU实验模型的设计步骤,设计人员在进行CPU设计实验时,需要选择自己的指令系统,更新此CPU模型上相应模块,以满足指令需求。并根据各条指令的执行步骤给出各个模块所需要的正确的控制信号,同时建立满足要求的数据通路,完成自己的设计。而设计是否正确需要观察每条指令执行时对应模块内的寄存器的数据是否正确变化来确定。基

于此,本文设计开发了相应的调试系统,对指令执行时的关键数据进行展示,使设计人员能直观地看到诸如PC,IR,GPR等寄存器的变化。

## 2 CPU 调试系统设计

CPU 调试系统支持两种运行模式,连续运行和单步调试。连续运行时实验CPU根据连续的时钟信号自动进行取指执行。单步调试时,调试者在上位机手动控制实验CPU单步执行。在这个过程中上位机展示实验CPU运行过程中寄存器的值,让设计人员对CPU的运行过程一目了然。因此调试系统需要读出CPU中关键寄存器的数据,并能和上位机通信,接收上位机发送的指令并且向上位机发送当前CPU的状态数据。图3是本次设计的CPU模型和调试系统的框架图。

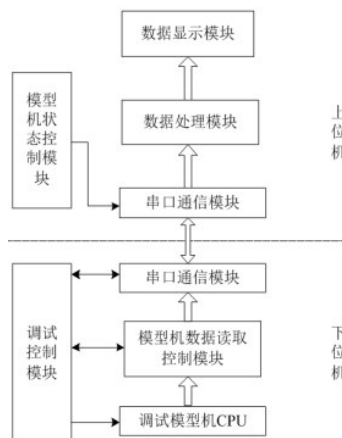


图3 开放式CPU的调试环境构成

### 2.1 上位机部分

调试系统上位机的功能分为控制模型机运行,串口通信,数据处理和数据显示。模型机状态控制模块控制下位机模型机的运行,通过串口通信模块向下发送调试控制命令。串口通信模块负责串口的管理,与FPGA和上层进行通讯。数据处理模块对提供的数据进行解析处理,分析出指令和数据,并将机器码反编译为对应的汇编码,组织好数据模型交给数据显示模块进行显示。数据显示模块提供了可视化的界面。本文用java语言对以上设计进行了实现。

### 2.2 下位机部分

调试系统的下位机的功能有:与上位机进行通信,根据上位机的调试控制命令控制实验CPU的执行和读取实验CPU的数据。图3中下位机的串口通信模块实现了串口通信,模型机数据读取控制模块负责从实验CPU中读取数据。调试控制模块控制串口通信完成通信功能,并根据上位机发送的调试控制命令控制实验CPU的运行。下位机全部用FPGA实现。

## 3 调试与验证

设计人员完成CPU编译仿真完后就可以和调试系统的下位机部分结合起来,并下载到FPGA开发板进行运行验证。本文使用Altera公司的DE2-70开发板进行验证。

本文用实验CPU的指令编写了一段简单的程序:实现 $1+3+5+7$ 的求和,并将和存取到数据存储器来验证。汇编程序设计如下:

ori \$1,\$0,0;设置\$1存储累加和,初始值为0

ori \$2,\$0,1;设置\$2为增长的加数,初始值为1  
ori \$3,\$0,2;设置\$3为加数增长的步长,步长是2  
ori \$4,\$0,9;设置\$4为累加终止条件,当加数增长到9时  
停止

loop:  
addu \$1,\$1,\$2;进行累加求和  
addu \$2,\$2,\$3;加数\$2增长  
bne \$2,\$4,loop;判断累加未结束跳至loop继续累加,否则  
执行下一条指令

ori \$5,\$0,8;设置\$5为8,作为存储器基址  
sw \$1,0(\$5);将\$1的值写入存储器\$5+0的地址中  
jal a;跳转到a处执行。  
a: lw \$2,0(\$5);将存储器\$5+0地址中的值写入\$2寄存器  
jr \$31;返回到Jal调用处  
译为机器码并写入到实验CPU中的IM模块中,IM中相关  
verilog代码如下:

```
initial begin
    imem[0]= 32'h34010000;//ori $1,$0,0
    imem[1]= 32'h34020001;//ori $2,$0,1
    imem[2]= 32'h34030002;//ori $3,$0,2
    imem[3]= 32'h34040009;//ori $4,$0,9
    imem[4]=32'h00220821;//loop: addu $1,$1,$2
    imem[5]= 32'h00431021;//addu $2,$2,$3
    imem[6]= 32'h1444fffd;//bne $2,$4,loop
    imem[7]= 32'h34050008;//ori $5,$0,8
    imem[8]= 32'haca10000;//sw $1,0($5)
    imem[9] = 32'h0c10000a; //jal a
    imem[10] = 32'h8ca20000; //lw $2,0($5)
    imem[11] = 32'h03e00008; //jr $31
end
```

使用调试系统进行调试后的结果为图4。

图4指令查看部分中红色标注的指令为待执行指令,存储器和寄存器的值在下一个时钟沿到来时更新。指令查看部分当前显示的是测试程序执行完时寄存器的情况,累加的和为0x10,将\$1中的和存入存储器再在跳转程序中取出到\$2后\$2更新为0x10,指令执行正确。实验CPU的指令符合单周期CPU的指令运行特点,测试程序能正确执行。

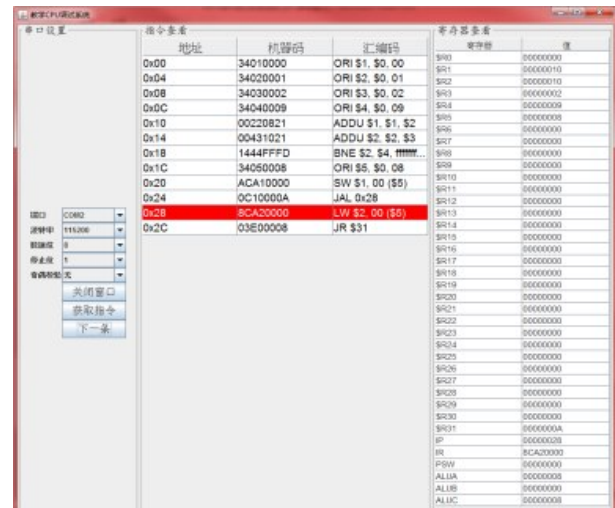


图4 上位机调试界面

#### 4 总结

通过设计和实现基于FPGA的开放式CPU实验模型,并通过调试系统进行调试验证,为设计人员提供了一个可以自主设计,更加直观开放式CPU实验平台。设计人员可以将主要精力放在CPU的设计和实现时的控制部分,调试时可以方便的调试,而忽视其他无关部分。提高了设计人员的创新性和动手能力。

#### 参考文献:

- [1] 康磊,宋彩利,李润洲. 数字电路设计及 Verilog HDL 实现[M]. 西安: 西安电子科技大学出版社, 2010.
- [2] 李山山,汤志忠,周继群. 基于FPGA的开放式教学CPU的设计与测试系统[J]. 计算机工程与应用, 2005, 41(14):98-100.
- [3] 戴维 A. 帕特森, 约翰 L. 亨尼斯, David A. Patterson, 等. 计算机组成与设计: 硬件/软件接口[M]. 机械工业出版社, 2011.
- [4] J.L.Hennessy. 计算机系统结构: 量化研究方法[M]. 电子工业出版社, 2004.
- [5] 唐翔飞. 计算机组成原理[M]. 2版. 北京: 高等教育出版社, 2008.
- [6] 雷思磊. 自己动手写CPU[M]. 电子工业出版社, 2014.
- [7] 夏宇闻. Verilog 数字系统设计教程. 第3版[M]. 北京航空航天大学出版社, 2013.