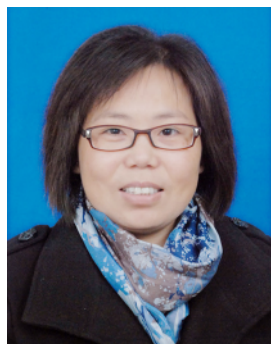


基于 MIPS 指令集的流水线 CPU 设计与实现

刘秋菊¹, 张光照², 王仲英²

(1. 郑州工程技术学院 信息工程学院, 郑州 450000; 2. 河南经贸职业学院 技术科学系, 郑州 450018)



摘 要: 提出了一种 CPU 设计方案, 实现指令集为 MIPS 指令集中选取 15 条指令作为本 CPU 的基本指令, 采用基本 5 步流水线 CPU 设计。分析了流水线 CPU 的逻辑结构与指令的处理过程, 给出了取指阶段 IF、译码阶段 ID、执行阶段 EX、内存访问阶段 MEM、寄存器写回阶段 WB 阶段的设计与实现。对流水线产生的相关性, 采用 Bubble 法和 Forwarding 法相结合的方法来消除相关性, 在 FPGA 平台上进行了测试, 测试结果表明, 该方案符合设计要求。

关键词: 流水线; 中央处理器; 设计; 指令集

中图分类号: TP 332 文献标志码: A

文章编号: 1006-7167(2017)08-0148-05

Design and Implementation of Pipeline CPU Based on MIPS Instruction Set

LIU Qiuju¹, ZHANG Guangzhao², WANG Zhongying²

(1. School of Information Engineering, Zhengzhou Institute of Technology, Zhengzhou 450000, China;

2. Department of Technical, Henan Institute of Economics and Trade, Zhengzhou 450018, China)

Abstract: In this paper, a design method of CPU was proposed, the instruction set of the CPU contained 15 items of MIPS instruction set, and the basic method of five step pipeline CPU design was used. Analyses were conducted on the logic structure of the pipeline CPU and the processing of the instruction. This paper also gave design and realization of stage IF, stage ID, stage EX, stage MEM and stage WB. About the pipeline-related problems, the paper adopted Bubble and Forwarding technologies to eliminate it. The tests on the FPGA platform show that the scheme meets the design requirements.

Key words: pipelines; CPU; design; instruction set

0 引言

在工业设计和机器人研究中, 微处理器作为控制部件的核心是设计的关键, 其性能直接影响整个系统的性能。CPU 作为微处理器的核心其设计得到很多学者的研究, 刘明达在研究中很粗略的介绍了 CPU 各模块的设计, 何克东在研究教学实验平台中提到了如

何设计高性能 CPU, 提到五级流水线 CPU 设计, 介绍了 CPU 设计有单周期和流水线两种设计 CPU 方式^[1-5]。现代微处理器广泛采用流水线 CPU 设计, 本文详细介绍了流水线 CPU 设计的 5 个阶段以及对遇到的 Hazard 数据相关问题提出了解决方法。

1 总体设计

1.1 实现的指令

本次设计从 MIPS 指令集中选取 15 条指令作为本 CPU 的基本指令, 其他指令在译码阶段均解释为未定义指令^[6-8]。指令格式如图 1 所示。

收稿日期: 2016-11-20

基金项目: 河南省科技攻关项目(172102210606); 河南省高等学校重点科研项目(17B520040)

作者简介: 刘秋菊(1970-), 女, 河南孟州人, 教授, 现主要从事计算机应用教学研究。Tel.: 15737152760; E-mail: mzlqj@126.com

MIPS Instructions									
Bit #	[31..26]	[25..21]	[20..16]	[15..11]	[10..06]	[05..00]	Operations		
R-type	op	rs	rt	rd	sa	func			
add	000000	rs	rt	rd	00000	100000	rd <-- rs + rt;	PC <-- PC + 4	
sub	000000	rs	rt	rd	00000	100010	rd <-- rs - rt;	PC <-- PC + 4	
and	000000	rs	rt	rd	00000	100100	rd <-- rs & rt;	PC <-- PC + 4	
or	000000	rs	rt	rd	00000	100101	rd <-- rs rt;	PC <-- PC + 4	
sll	000000	00000	rt	rd	sa	000000	rd <-- rt << sa;	PC <-- PC + 4	
srl	000000	00000	rt	rd	sa	000010	rd <-- rt >> sa (logical);	PC <-- PC + 4	
sra	000000	00000	rt	rd	sa	000011	rd <-- rt >> sa (arithmetic);	PC <-- PC + 4	
I-type	op	rs	rt	immediate					
addi	001000	rs	rt	immediate			rt <-- rs + (sign_extend)immediate;	PC <-- PC + 4	
andi	001100	rs	rt	immediate			rt <-- rs & (zero_extend)immediate;	PC <-- PC + 4	
ori	001101	rs	rt	immediate			rt <-- rs (zero_extend)immediate;	PC <-- PC + 4	
lw	100011	rs	rt	immediate			rt <-- memory[rs + (sign_extend)immediate];	PC <-- PC + 4	
sw	101011	rs	rt	immediate			memory[rs + (sign_extend)immediate] <-- rt;	PC <-- PC + 4	
beq	000100	rs	rt	immediate			if (rs == rt) PC <-- PC + 4 + (sign_extend)immediate<<2; else	PC <-- PC + 4	
bne	000101	rs	rt	immediate			if (rs != rt) PC <-- PC + 4 + (sign_extend)immediate<<2; else	PC <-- PC + 4	
J-type	op	address							
j	000010	address			PC <-- (PC+4)[31..26],address<<2				

图1 CPU基本指令格式

1.2 五步流水线组成

流水线5个阶段分别是指取指阶段IF、译码阶段ID、执行阶段EX、内存访问阶段MEM、寄存器写回阶段WB(见图2)。

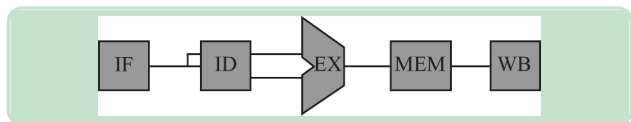


图2 五步流水线组成

1.3 流水线CPU逻辑原理图

基本5步流水线CPU的逻辑原理图如图3所示。

总的来说由5个Stage Register和5个Stage部分组成:在各阶段执行完成后所产生的结果将锁存在Stage Register中,以供下一阶段使用;CPU控制信号在译码阶段由控制单元产生,也将后面阶段才发生作用的控制信号锁存在Stage Register中。执行结果和控制信号就是这样在Stage Register中一级一级传递,以满足各个阶段的执行与控制需求。因此,Stage Register在流水线结构中起着非常重要的作用,它必须是边沿触发的,以保证本条指令的中间数据不会被后续指令干扰。

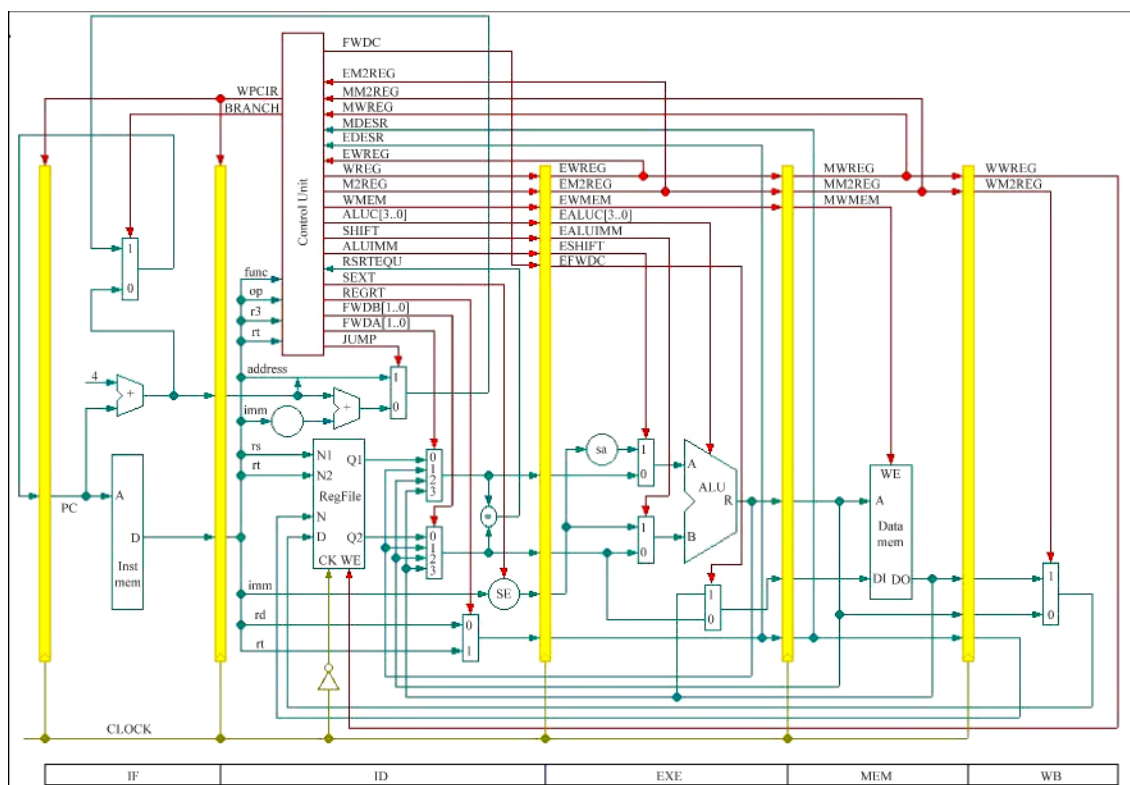


图3 基本5步流水线CPU的逻辑原理图

这样由 5 个 Stage 部分和 5 个 Stage Register 的流水线结构设计非常清晰,各个阶段执行自己的任务,由 Stage Register 作为数据通路,使得具体实现过程中可以对各个阶段单独进行测试,最后再将各阶段串联起来进行集成测试,功能调试与测试将变得简单。

1.4 流水线 CPU 实现原理图

基本 5 步流水线 CPU 的实现原理图如图 4 所示。实现原理图保持了逻辑原理图中 5 个 Stage Register 和

5 个 Stage 的清晰的设计结构:前缀为 REG 的部件是 Stage Register;前缀为 STAGE 的部件为 Stage。整体数据流向为从上往下、自左往右,结构比较清晰,且各个部件均经过单独测试成功后才集成在一起。本实现原理图为基本 5 步流水线结构,还不包括 Cache 和精确中断响应部分。增加 Cache 和精确中断响应部分需要对总体流水线结构做适当扩充,具体扩充后的原理图以及实现在今后继续研究。

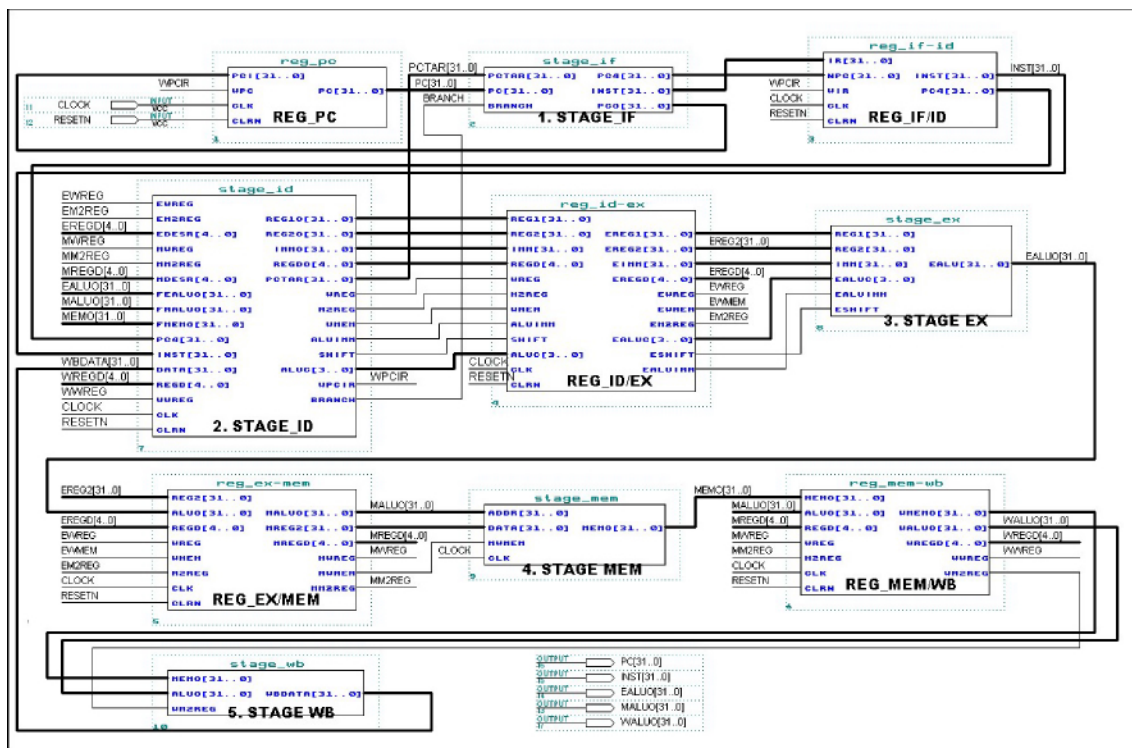


图 4 基本 5 步流水线 CPU 实现原理图

2 各流水阶段具体实现

2.1 IF 阶段实现

IF 阶段要完成的任务是:从指令存储器中取出指令、将当前 PC 值加 4、选择出 PC 的下一个值。实现原理图如图 5 所示,IF 阶段需要判断下一个 PC 值是 PC + 4 还是跳转的目的地址,判断的依据是 ID 阶段所给的 BRANCH 控制信号,因此输入脚必须包括 BRANCH 控制线和跳转目标地址数据线 PCTAR。

2.2 ID 阶段实现

ID 阶段需要完成的任务比较重:指令译码、根据指令类型产生控制信号、从寄存器文件中选择需要的寄存器、判断是否是绝对跳转或分支指令,如果是还要计算出它们的跳转目标地址、检测是否存在可能的 Data Hazard,并选择是否使用 Bubble 法和 Forwarding 法消除 Hazard^[9-12]。实现原理图如图 6 所示,其中需要特别说明的是寄存器文件只实现了 8 个,而不是 MIPS 指令集的 32 个。最重要的部分是控制单元,它是用 Verilog 语言实现的,任务是解释指令、产生控制

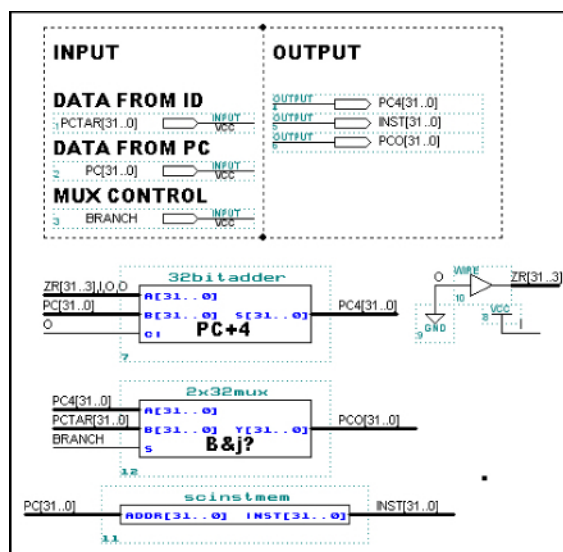


图 5 IF 实现原理图

信号、检测与消除 Data Hazard。

控制单元的输出脚都是控制信号,而输入脚中有 6 个是为了 Data Hazard 的检测与消除。在检测 Data Hazard 时,首先测试 EWREG 和 EM2REG 输入脚:如

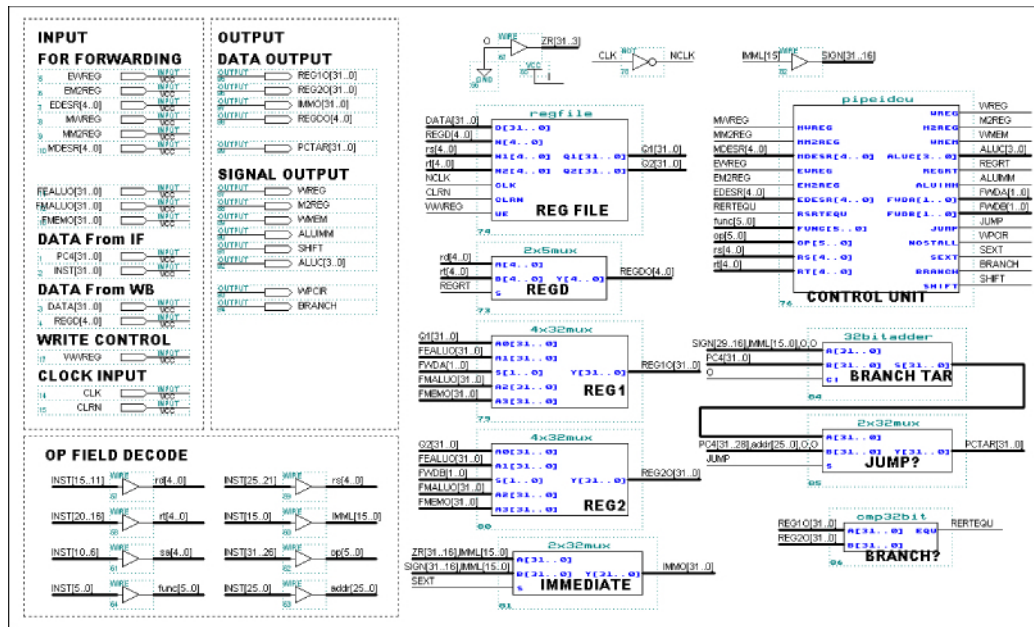


图6 ID实现原理图

果 EWREG 为 1,那么本条指令的前一条指令是回写寄存器指令;如果 EM2REG 为 1,那么前一条指令是 Load 指令。再用本条指令的 RS 和 RT 字段(即本条指令可能引用的寄存器地址)与 EDESRL 比较就可以知道是否存在 Data Hazard 数据相关性。同样通过测试 MWREG、MM2REG 及 MDESRL 即可知道本条指令与前条指令是否存在 Data Hazard。

如果存在 Hazard,则根据不同情况发出不同的信号控制 Hazard 的消除。NOSTALL 输出信号用于 Bubble 技术的控制,而 FWDA、FWDB 及 FWDC 输出

信号用于 FORWARDING 技术的控制。

2.3 EX 阶段实现

EX 阶段执行运算任务,本 CPU 可以执行加减、逻辑及移位三类运算。EX 阶段最主要的任务是进行运算,运算单元 ALU 实现原理图如图 7 所示。

2.4 MEM 阶段实现

MEM 阶段执行访问数据存储器任务,本次设计中将指令与数据存储器分开,且存储容量都是 32 个字。实现原理图如图 8 所示。

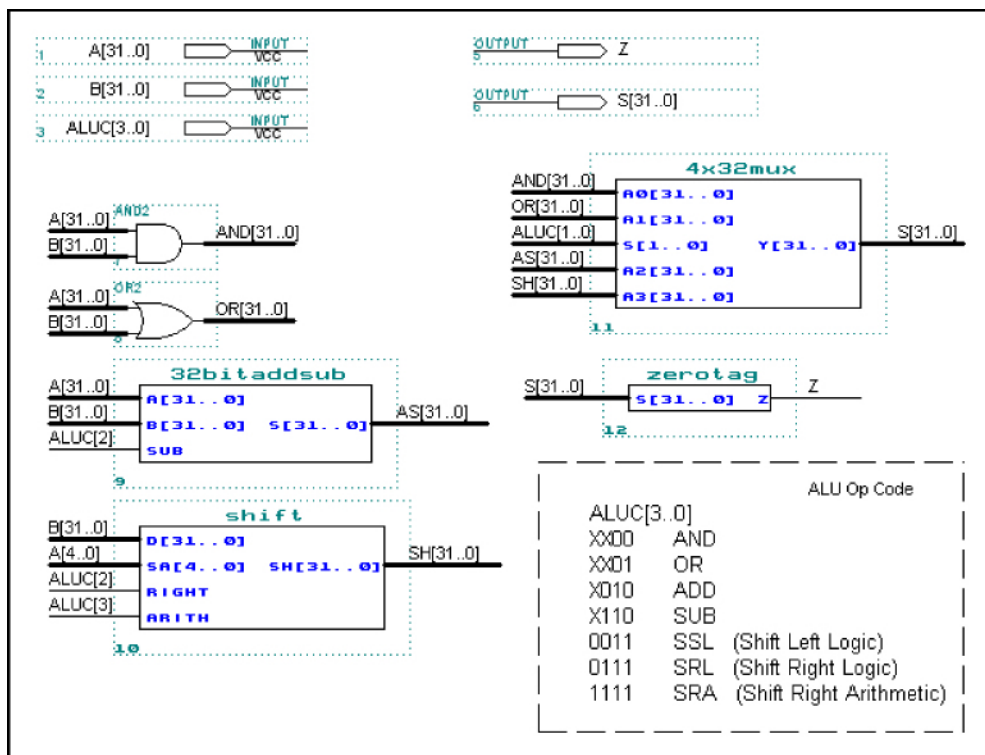


图7 ALU实现原理图

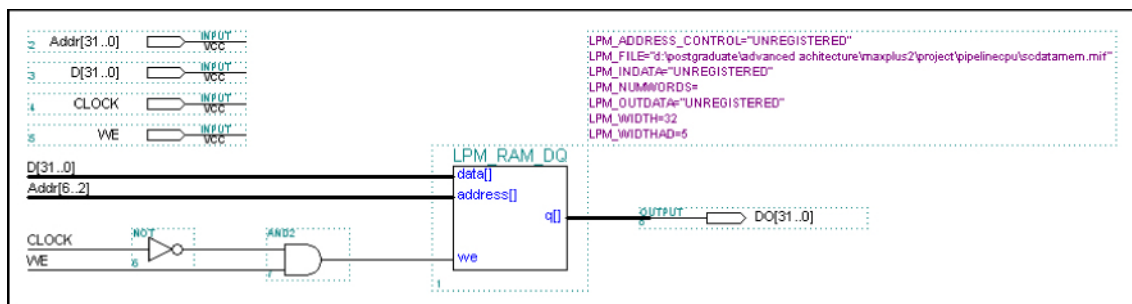


图 8 MEM 阶段实现原理图

2.5 WB 阶段实现

WB 阶段执行将指令执行结果写回寄存器的任务,而执行结果有两种可能:一种是 ALU 运算结果;另一种是 MEM 取数据结果。实现原理图如图 9 所示,

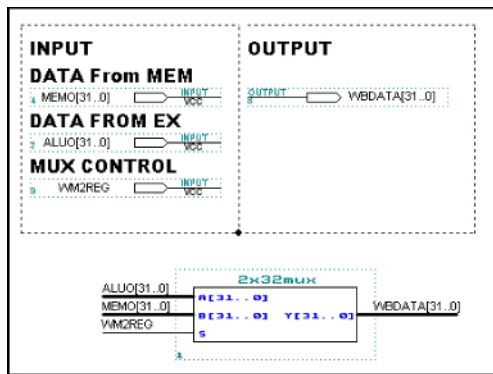


图 9 WB 阶段实现原理图

根据 WM2REG 控制信号(即判断指令是否为 Load 指令)来选择最终的回写数据。实际上从这可以看出本设计中所用的都是开关式控制方式,它的特点是不管指令属于何种类型,按其所有可能的形式执行操作,让所有的部件都运转起来,最后可能产生多种结果数据,再由控制信号来控制开关以选择所需的结果。

3 功能模拟测试

功能模拟测试需要设计典型的代码序列来测试流水线的运行状况及 Hazard 问题的解决^[13-16]。图 10 所示为测试代码与测试波形图。

测试结果表明,该设计是成功的,同时该设计的应用成果在电子设计大赛中也取得了很好的成绩。

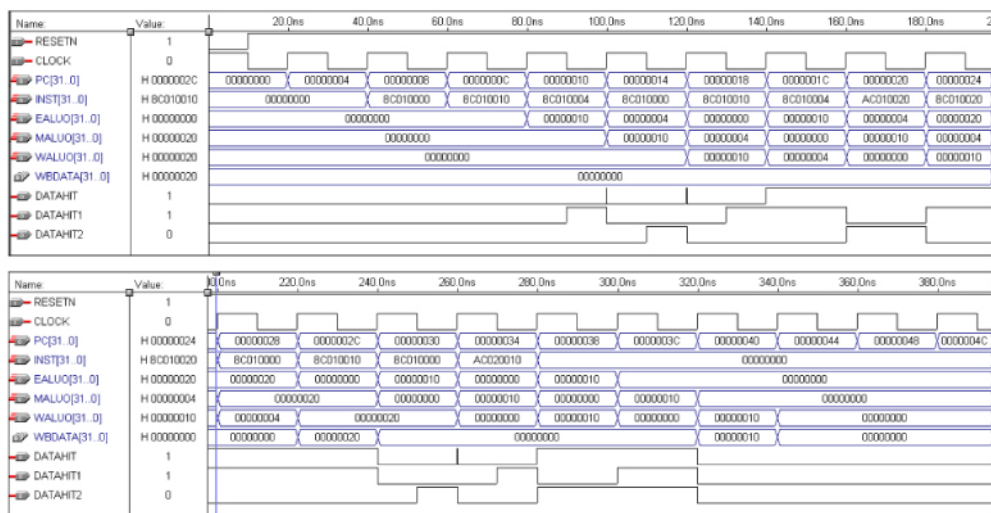


图 10 测试代码与测试波形图

4 结 语

提出了一种流水线 CPU 的设计方法。介绍了 5 步流水线 CPU 设计的步骤,并针对逻辑图介绍了每个阶段的详细实现,CPU 设计中还要考虑数据相关性问题,本次设计中在 ID 阶段通过不同的控制信号来解决不同的数据相关性问题,最后进行了功能测试,测试结果符合设计要求。CPU 设计中会遇到很多问题,例如缓冲、中断等,这也是下一步要继续研究的问题。

参考文献(References):

- [1] 朱子玉,李亚民. CPU 芯片逻辑设计技术[M]. 北京:清华大学出版社,2005.31-110.
- [2] John L. Computer architecture: A quantitative approach, fourth edition [M]. 3rd ed. Boston: AP ProtwaiaonL, 2006. 2009: 88-172.
- [3] 刘明达. 基于 VerilogHDL 的简单 CPU 设计[J]. 计算机光盘软件与应用,2013,16(13):302-304.

(下转第 172 页)

文化氛围更有利于碰撞出新思维的火花。实验中心从教学文化、管理文化和创新文化三个方面营造开放共享、宽松自由、具有竞争力的文化氛围,培养学生的批判性思维和创新能 力,使学生的知识、能力和素质协调发展,具有到国内外一流大学继续深造的实力。为商界特别是“互联网+”及其应用领域,培养兼备信息技术专业素质和经济管理才能的复合型未来领军人才。我们的初步探索也希望能对其他高校经管类实验室文化建设提供有益的参考。

参考文献(References):

- [1] 柯红岩,钱大益,刘云,等. 高校实验室文化建设探索[J]. 实验技术与管理,2012,29(3):328-330.
- [2] 高峰. 高校实验室文化建设问题的探讨[J]. 实验室科学,2012,15(2):128-129.
- [3] 周守喜,胡毅,石东平. 论高校实验室文化建设及应注意的几个问题[J]. 重庆文理学院学报(自然科学版),2006,5(4):95-97.
- [4] 朱绪海. 借力“互联网+”助推教育现代化[J]. 学周刊,2016(5):212.
- [5] 刘国瑜,张英. 试析国家重点实验室的文化管理[J]. 实验室研究与探索,2007,26(9):107-110.
- [6] 徐建东,王海燕,楼松林. 高校开放实验管理理念创新[J]. 实验技术与管理,2008,25(9):127-130.
- [7] 武芳. 高校实验室文化建设创新探究[J]. 实验室研究与探索,2014,33(8):244-247.
- [8] 王松武,王伞. 开放实验室中创建实验室文化[J]. 实验室研究与探索,2010,29(10):154-156.
- [9] 张原,李鑫,杜兴号. 高校实验室文化的内涵及建设途径[J]. 实验技术与管理,2011,28(3):15-19.
- [10] 李东荣. 大数据时代的金融人才培养[J]. 中国金融,2013(24):9-10.
- [11] 汪连新. 互联网金融和大数据时代金融学课程改革及人才培养的思考[J]. 教育教学论坛,2015(32):189-191.
- [12] 夏晖,孔晓李,李平. 管理与技术相融合的大学生创新创业模式[J]. 实验室研究与探索,2016,35(9):196-199.
- [13] 夏晖,马永开. 加强经管类专业实验教学的思考[J]. 实验技术与管理,2007,24(4):117-119.
- [14] 孟昭霞. 高校实验室管理:制度管理与文化管理的融合[J]. 实验技术与管理,2016,33(4):12-15.
- [15] 夏石头,萧浪涛,匡勇,等. “以人为本和谐发展”的实验室文化建设[J]. 实验室研究与探索,2009,28(2):5-7.
- [16] 陶行知. 育才学校手册[J]. 重庆陶研文史,2012(3):11-25.
- [17] 胡小平. 专业实验室文化建设探究[J]. 实验室研究与探索,2011,30(9):328-330.

(上接第128页)

- [3] 顿爱波. 远程教学用虚拟电子实验室系统研究[D]. 大连:大连理工大学,2005.
- [4] 李世红. 基于LabVIEW和单片机的温度监测系统[J]. 湖北农业科学,2015(19):4836-4839.
- [5] 艾建生. 基于Web自控远程实验室的研究[D]. 重庆:重庆大学,2008.
- [6] 米西峰. 基于Web和Labview的虚拟仪器实验平台设计[J]. 微计算机信息,2008(4):155-157.
- [7] 唐立军,宾峰,邓敏,等. 远程实验操作平台设计与实现[J]. 实验室研究与探索,2015(12):49-52.
- [8] 江贤志,刘华章,张教育,等. 基于C8051F020单片机的RS485串行通信设计[J]. 电子测量技术,2014(3):121-124.
- [9] 李楠. 基于单片机技术的多机通信系统设计[J]. 现代电子技术,2016(12):164-166.
- [10] 金彧,王彩莲,卢军平,等. 一种自由进出JTAG菊花链的硬件设计方法[J]. 电脑开发与应用,2012(8):47-49.
- [11] 刘文怡,李进武. 基于RS485总线多机通信系统可靠性的研究[J]. 火箭与制导学报,2005(3):102-104.
- [12] 刘俊伏,刘金娥. PC机与单片机的多机通信技术及抗干扰措施[J]. 河北师范大学学报(自然科学版),2009(1):28-32.
- [13] 张茜,雷勇. 基于虚拟仪器技术的网络化远程实验室系统[J]. 实验室研究与探索,2013(9):90-93.
- [14] 孙道让,刘蕴红. 基于LabVIEW和IAP的单片机远程实验系统[J]. 电子设计工程,2011(20):177-180.
- [15] 陶方华,张艳荣. 基于DataSocket技术的远程实验系统设计[J]. 四川兵工学报,2011(5):83-84.
- [16] 罗建,雷勇,黄昊,等. 基于LabVIEW的远程电机实验台的构建[J]. 实验技术与管理,2014(3):122-125.

(上接第152页)

- [4] 何克东,王恒才. 高性能CPU设计实验平台及其应用[J]. 实验室研究与探索,2007(12):212-214.
- [5] 陈国强. 基于AD9954的信号发生器设计[J]. 实验室研究与探索,2010,29(8):222-225.
- [6] 刘明,蔡启先,余祖峰. 改进MIPS指令集模拟方法的研究[J]. 广西工学院学报,2009(12):54-57.
- [7] 薛勃,周玉洁. MIPS32指令集兼容的CPU模拟器设计[J]. 计算机工程,2009,35(1):263-265.
- [8] 袁婷,刘怡俊. 自主设计精简指令集的流水线CPU[J]. 微电子学与计算机,2015,32(2):124-128.
- [9] 李山山,刘敬晗. 利用 Tomasulo 算法处理数据相关的流水线CPU设计[J]. 实验室研究与探索,2014,33(12):90-95.
- [10] 刘秋菊,李飞,刘书伦. 一种流水线CPU设计中的Hazard消除法[J]. 微电子学与计算机,2012,29(8):33-36.
- [11] 王艳秋. 基于FPGA的CPU设计与实现[D]. 天津:河北工业大学,2008.
- [12] 赖兆馨,潘明,张辉. 基于FPGA流水线CPU控制器的设计与实现[J]. 微计算机信息,2008(20):233-235.
- [13] 曹学飞,张盛兵,张骏. 32位CISC微处理器流水线的设计[J]. 微电子学与计算机,2007,24(4):186-189.
- [14] 刘秋菊,李飞,刘书伦. 带Cache和精确中断响应的CPU设计[J]. 实验室研究与探索,2012,31(3):68-74.
- [15] 曹学飞,张盛兵. “LongtiumC2”微处理器流水线设计[J]. 微处理机,2010(1):9-12.
- [16] 东野长磊,戚梅. 一种带Cache的嵌入式CPU的设计与实现[J]. 微型机与应用,2010,29(14):17-19.