

# 基于现场可编程门阵列的 RISC 处理器设计

东野长磊

(山东科技大学信息科学与工程学院, 山东 青岛 266510)

**摘 要:** 基于现场可编程门阵列(FPGA)平台, 设计嵌入式精简指令集计算机(RISC)中央处理器(CPU)。参考无内部互锁流水级微处理器(MIPS)指令集制定原则设计 CPU 指令集, 通过分析指令处理过程构建嵌入式 CPU 的 5 级流水线, 结合数据前推技术和软件编译方法解决流水线相关性问题, 并实现 CPU 的算术逻辑单元、控制单元、指令 cache 等关键模块设计。验证结果表明, 该嵌入式 RISC CPU 的速度和稳定性均达到设计要求。

**关键词:** 现场可编程门阵列; 精简指令集计算机处理器; 流水线相关性; 算术逻辑单元

## Design of Reduced Instruction Set Computer Processor Based on Field Programmable Gate Array

DONGYE Chang-lei

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China)

**【Abstract】** This paper designs a embedded Reduced Instruction Set Computer(RISC) Central Processing Unit(CPU) based on Field Programmable Gate Array(FPGA) platform. The instruction set is designed refer to Microprocessor without Interlocked Pipeline Stage(MIPS) instruction set principle. By analyzing the process of each instruction, the 5-stage pipeline of embedded CPU is built. It adopts data forwarding technology and software compiler method to solve pipeline-related problem. The key modules of CPU: Arithmetic Logic Unit(ALU), control unit, instruction cache are designed. Verification results show that the embedded RISC CPU speed and stability meet the design requirements.

**【Key words】** Field Programmable Gate Array(FPGA); Reduced Instruction Set Computer(RISC) processor; pipelining correlation; Arithmetic Logic Unit(ALU)

DOI: 10.3969/j.issn.1000-3428.2011.11.084

### 1 概述

随着集成电路设计和工艺技术的发展, 集成电路的集成度越来越高。将微处理器、数字和模拟 IP 核、存储器和各种外设接口等集成在单一芯片上, 构成片上系统(System on Chip, SoC)。SoC 具有面积小、低功耗、多功能和低成本等优点, 是未来集成电路发展的方向。作为 SoC 核心的微处理器, 其性能直接影响整个系统性能。为提高中央处理器(Central Processing Unit, CPU)的效率和指令执行的并行性, 现代微处理器广泛采用流水线设计, 所以, CPU 流水线设计决定了其性能。无内部互锁流水级微处理器(Microprocessor without Interlocked Pipeline Stage, MIPS)是一种典型的精简指令集计算机(Reduced Instruction Set Computer, RISC)微处理器, 在嵌入式系统领域得到广泛应用。MIPS32™ 指令集开放、指令格式规整、易于流水线设计、大量使用寄存器操作<sup>[1]</sup>。与复杂指令集计算机(Complex Instruction Set Computer, CISC)微处理器相比, RISC 具有设计简单、设计周期短等优点, 并可以应用更多先进技术, 开发更快的下一代处理器。本文搭建嵌入式 RISC 的体系结构。依据 MIPS32 指令集, 给出 CPU 的所实现指令集, 对每条指令进行分析, 构建 5 级流水线结构。对流水线所带来的竞争问题进行分析, 针对 3 种不同类型的竞争, 采用不同的解决方案, 并给出 RISC CPU 关键模块的实现, 得到 CPU 的流水线逻辑结构。

### 2 CPU 体系结构

#### 2.1 指令集的选取

本文设计实现的指令兼容 MIPS 系列 RISC 处理器的指令

集。由于 MIPS32™ 指令集是开放的指令集, 因此指令格式非常简单, 按照指令格式可分为 3 类, 包括寄存器类型(R-type)指令、立即数类型(I-type)指令和跳转类型(J-type)指令。这 3 类指令均为 32 位, 而且指令操作码在固定位置上。这种特点易于将指令代码进行拆分, 易于进行流水线 CPU 设计。

指令类型参考 MIPS 处理器的指令集设计原则。所有指令的运算都在寄存器中进行, 当需要和内存交换数据时, 通过内存访问指令完成内存和寄存器的数据交换。设计实现程序中经常使用的 34 条指令, 实现指令集按照功能分成 5 种类型, 包括算术运算类指令、逻辑运算类指令、数据传送指令、条件转移和无条件跳转类指令、特殊指令等。

#### 2.2 流水线设计

在基本 MIPS 处理器中有 5 个流水级<sup>[2]</sup>, 其中, 各流水级定义与主要功能为 IF 为计算下一条指令的地址 PC, 并从指令存储器读取指令; ID 对指令进行译码, 从寄存器堆中取出源操作数; EX 为当指令是运算类指令时执行运算, 当指令是转移类指令时进行有效地址计算; MEM 为从数据存储器读写数据; WB 为将数据写回到寄存器堆。按照这个流水线结构, 本文设计和实现一种较通用的 MIPS CPU, 通过超高速集成电路硬件描述语言(Very High Speed Integrated Hardware

**基金项目:** 国家“863”计划基金资助重点项目(2009AA062701); 山东科技大学“群星计划”基金资助项目(qx104011)

**作者简介:** 东野长磊(1978—), 男, 讲师、博士研究生, 主研方向: 嵌入式系统

**收稿日期:** 2010-12-30 **E-mail:** dycl.cn@126.com

Description Language, VHDL)语言实现,各模块之间的关系如图1所示。

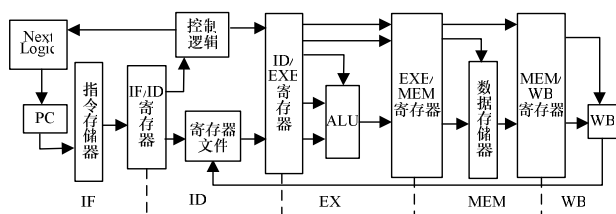


图1 流水线 CPU 结构

### 3 流水线中的相关性问题

由于指令以流水线形式并行处理,因此会产生指令相关性问题,一般存在3种相关:结构相关,数据相关和控制相关<sup>[3]</sup>,引起流水线竞争。结构相关问题是指导由于硬件资源不足导致流水线不畅通,例如只有一个存储器模块时,不能对存储器同时取指令和数据。数据相关问题是指导一条指令的后续指令要使用该条指令的结果。控制相关问题是指导转移指令从取指到转向目标地址要花费几个时钟周期,但流水线CPU在每个周期都取指令。

本文通过尽量增加硬件电路资源解决结构相关问题,采用哈佛架构,使用指令存储器和数据存储器避免结构竞争。对于寄存器组存在的结构竞争,采用由D触发器构建的寄存器避免这种状况,当写入地址和读出地址相同时,直接用写入数据驱动读出总线。数据相关问题可以用数据前推技术得到缓解。数据前推技术对于算术逻辑单元(Arithmetic Logic Unit, ALU)计算指令非常有效,但对于存储器读数据指令(load 指令),如果下面指令希望立即使用该结果,则必须暂停流水线一个周期。对于控制相关,可以使用指令重组优化及延迟转移技术等软件编译方法以解决。

## 4 流水线关键模块的实现

### 4.1 ALU 的实现

ALU 是数据通路中的重要部件,负责完成各种运算功能。根据CPU实现的指令集,确定ALU的操作控制信号数据宽度为5位,运算的数据位数为32位。操作控制信号(ALU\_OP)和ALU的源数据选择信号根据不同指令的译码由控制逻辑产生。

### 4.2 控制单元的设计

控制单元根据输入的指令码产生一系列控制信号,用于控制数据通路上的多路选择器和各功能部件,保证每条指令

都能正确执行。

控制单元的输入信号必须设计为32位的指令码,输出信号则根据需要进行设计<sup>[4]</sup>。在IF阶段,控制单元需要根据指令的译码情况,决定PC的更新值,如果是顺序执行的指令,则PC自动加4,对于分支和跳转指令,需要发出跳转指令信号和分支指令信号,从而决定PC的更新值。在ID阶段,控制单元对指令进行译码,根据指令的操作码和功能部分,发出相应控制信号;根据指令中的操作数字段,控制单元给出寄存器号,从寄存器堆中读出操作数送入ID与EXE之间流水线寄存器。如果发生数据相关,则数据前置逻辑产生前置控制信号。在EXE阶段,控制单元给出ALU的数据来源的选择信号,以及ALU的运算选择信号,在MEM阶段,控制单元需要给出数据存储器的读写信号、片选信号等。存储器需要向控制单元返回响应信号。在WB阶段,控制单元主要控制数据流向,给出多路选择器的选择信号,选择将存储器读出数据还是将ALU的运算结果写回寄存器组。

### 4.3 数据前推技术的设计

对于数据竞争的检测,通过比较连续3条指令的寄存器标号,把本条指令的rs、rt和前面2条指令的操作数结果寄存器进行分别比较,比较器的输出信号传递到EXE阶段用于选择ALU操作数的来源。对于load指令发生的数据相关,由于要等到MEM阶段完成后才能得到有效数据,因此发生数据相关的下一条指令,只能通过延迟一个周期才能利用数据前置技术,如果配合MIPS编译器,通过使用延迟槽技术,则可以解决load类型的数据相关<sup>[5]</sup>。

### 4.4 指令 cache 的实现

系统实现了一个容量为2KB指令cache,每个cache行大小为16Byte数据,这样可以利用存储器的16Byte的突发式传送<sup>[6]</sup>。本文采用2路组相联方式,支持通写模式,由同步静态随机存取存储器(Static Random Access Memory, SRAM)实现。数据cache由控制模块、命中与缺失比较模块、访问内存模块、替换模块、输出模块组成。其中,控制模块是整个cache的主控部件,它控制存储器和cache协调工作:当执行单元有取指请求时,以指令的物理地址作为索引看是否命中,如果不命中,那么控制逻辑启动访存逻辑到内存中去取指,当指令取回时控制逻辑启动替换逻辑对指令cache进行替换并将指令输出;如果命中,则将指令输出。

本文使用VHDL设计和实现上述各关键模块。对关键模块和其他模块进行融合,最后得到的CPU流水线结构见图2。

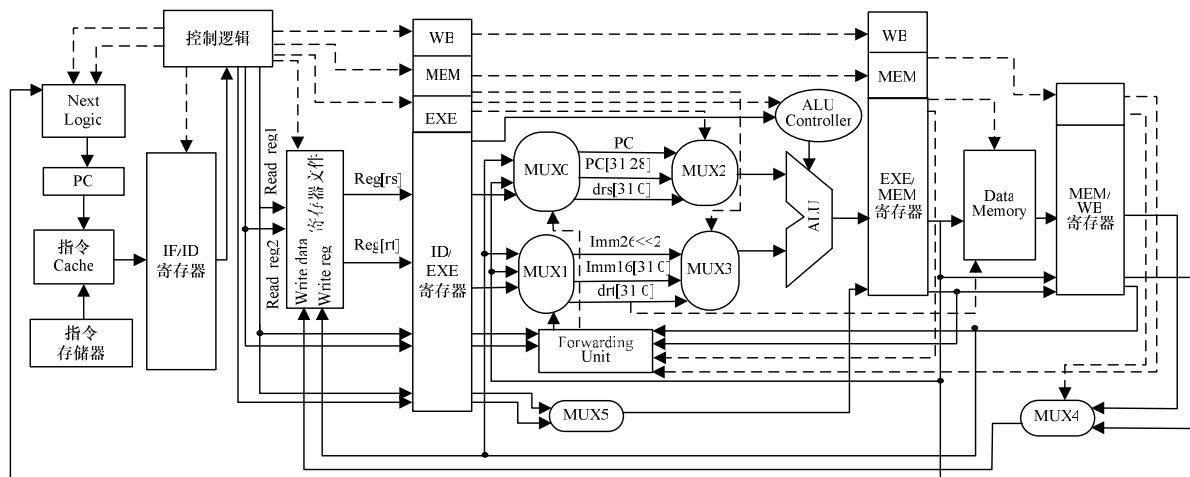


图2 流水线 CPU 的逻辑结构



## 5 系统仿真与验证

本文使用 VHDL 实现对各功能模块的设计, 并完成功能仿真后, 将设计的控制单元和数据通路的各模块进行合并, 形成一个完整的嵌入式 RISC CPU 核, 进行系统级仿真。基于系统实现的指令集编写了一个简单的测试程序。

```
add    $5,$0,$0
addi   $7,$0,1
sw     $7,10($5)
lw     $8,10($5)
```

将指令码写入指令存储器的仿真文件, 测试程序运行得到的仿真波形图如图 3 所示。

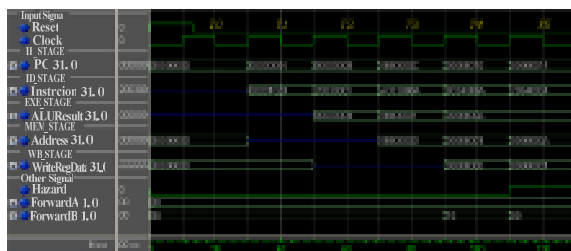


图 3 测试程序运行时的仿真波形图

每个时钟周期为 10 ns, 第 1 个时钟周期 T1 从 10 ns 处开始, 根据仿真波形可以看出, 在 T5 周期, 指令 sw \$7, 10(\$5) 处于 EXE 阶段, 第 2 条指令 addi \$7, \$0, 1 处于 MEM 阶段, 需要进行数据前推, Forward\_2 的值为“10”, 通过对测试结果分析, 数据前推成功。通过分析仿真波形图中各输出信号的波形, 根据程序的运行过程, 可以判断信号波形正确, 达到设计要求。

(上接第 241 页)

析图 1 所示的区域泵站优化调度模型性能, 并给出该模型的可达图。在图 1 所示初始状态下, 仿真运行 150 步(变迁次数)后得 19 个实时水位值, 结果如图 2 所示, 泵站被控制在目标水位 7 m 附近稳定运行, 控制效果较好。

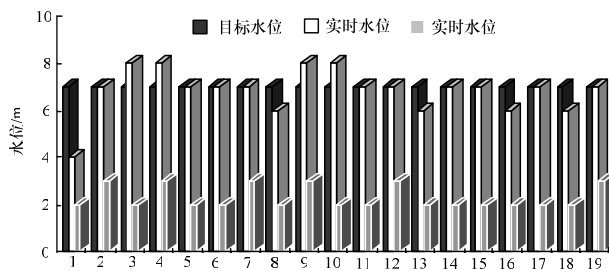


图 2 实时水位仿真结果

用 CPN Tools 工具对该 CPN 模型作状态空间分析, 报告摘录如下:

Liveness Properties: Dead Markings—None||Dead Transition Instances—None||Live Transition Instances—All  
Fairness Properties: Fuzzy\_control'Defuzzier—Impartial||Fuzzy\_control'Fuzzier—Impartial||Fuzzy\_control'Fuzzier  
\_Reasoning—Impartial||Fuzzy\_control'Order—Impartial||TOP'Adj ust\_h—Impartial||TOP'Data\_Store—Impartial||  
TOP'Execute—Impartial||TOP'Get\_Qin—Impartial

活性<sup>[5]</sup>分析表明该模型不存在死标识状态和死变迁, 所有变迁都是活的, 进一步表明了模型的安全性; 从公平性上看, 所有变迁都是公平的(变迁 Fuzzy\_control 也公平, 因为 Fuzzy\_control 层的 4 个变迁均公平), 说明该区域泵站优化调

## 6 结束语

嵌入式 CPU 是 Soc 平台的重要组成部分<sup>[7]</sup>, 本文根据 CPU 核的体系结构和所实现的指令集, 设计一个 5 级流水线 CPU, 分析流水线带来的 3 种相关性问题, 给出相应的解决方案, 并在 CPU 和指令存储器之间构建一个指令 cache, 仿真结果证明了 cache 工作的有效性。本文仅给出流水线 CPU 的关键模块的 VHDL 实现, 经过逻辑综合和仿真结果表明在时序上设计的嵌入式 CPU 很好地满足了流水线的要求, 生成的位流数据文件对 FPGA 进行器件编程, FPGA 芯片可以在 50 MHz 的时钟频率下稳定运行。

### 参考文献

- [1] Stallings W. Reduced Instruction Set Computer Architecture[J]. Proceedings of the IEEE, 1998, 76(1): 38-55.
- [2] Balch M. Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture[M]. [S. l.]: Hill Professional Press, 2004.
- [3] Sweetman D. See Mips Run[M]. 2nd ed. San Francisco, USA: Morgan Kaufmann Publishers, 2007.
- [4] 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, 2004.
- [5] 薛 勃. 32 位 MIPS 微处理器研究及其软硬件建模[D]. 上海: 上海交通大学, 2007.
- [6] 马 鹏, 卢景芬, 龚令侃. 32 位嵌入式 CPU 的微体系结构设计[J]. 计算机工程, 2008, 34(9): 136-138.
- [7] 赖铭强, 聂新义, 段国东. 高性能嵌入式处理器技术[J]. 计算机工程, 2009, 35(14): 280-282.

编辑 陆燕菲

度 CPN 模型系统中不存在任何独占资源部分。

仿真 CPN 模型能清晰显示调度状态参数(即包含有各库所托肯的标识)随变迁发生而变化的动态过程。该功能为区域泵站优化调度动态运行过程参数可视化提供了新的途径。

## 4 结束语

城镇污水合理调度排放事关节能和环境改善两大主题。本文从排水调度可视化的角度探讨了“区域泵站调度过程”的参数有效显示问题。依据泵站流量进/出平衡防止污水溢出的控制原则, 设计了一种区域排水泵站优化调度模糊控制策略, 并用基于 CPN 理论建立相应的区域泵站优化调度模型做 CPN Tools 软件仿真。实验证明, 该策略控制的泵站能稳定运行在目标水位附近, 该模型为区域排水泵站调度过程可实现参数可视化。这也为下一步研究排水系统的调度规律、建立优化调度新模型提供了方便。

### 参考文献

- [1] 汪雄海, 石宏宇. 污水管网泵站系统的控制策略研究[J]. 电工技术杂志, 2002, (8): 32-34.
- [2] He Zhongjie, Wang Xionghai. Adaptive Parameter Estimation-Based Predictive Multi-model Switching Control of Drainage Systems[C]//Proceedings of the 6th World Congress on Intelligent Control and Automation. Dalian, China: [s. n.], 2006: 6540-6543.
- [3] 杨 飞, 于永海, 徐 辉. 国内梯级泵站调水工程运行调度综述[J]. 水利水电科技进展, 2006, 26(4): 84-86.
- [4] 袁崇义. Petri 网原理与理论[M]. 北京: 电子工业出版社, 2004.
- [5] 王鹏伟, 吴哲辉. 几种满足汇合性质的 Petri 网子类[J]. 计算机工程, 2009, 35(4): 44-47.

编辑 金胡考

