

# Design of RISC Microprocessor Based on MIPS Instructions

ZOU Zhibin, ZHANG Qinghai, MA Ming,

*The First Aeronautic Institute of the Air Force, Xinyang, China*

*e-mail: zouzhibin@foxmail.com*

**Abstract:** The purpose of this dissertation is to design a 32 bits RISC microprocessor based on MIPS instructions and to do FPGA prototype verification. In this paper a 32-bit RISC Microprocessor of 34 MIPS instructions using VHDL is designed. Five stages (viz. instruction fetch stage, instruction decode stage, execution stage, data memory stage and write back stage) pipelining is used to improve the overall CPI (Clock Cycles per Instruction). The data forwarding unit and hazard detection unit are adopted to solve data hazard. And to conquer the question of control hazard which is a result of the branch or jump instructions, comparer and forwarding unit are added in the instruction decode stage to reduce the delay of branches. The validity of this design is verified by soft simulations and FPGA prototype verification.

**Keywords:** RISC microprocessor; pipeline; FPGA; control unit; simulator

## 基于 MIPS 指令集的 RISC 微处理器设计

邹志斌, 张庆海, 马明

空军第一航空学院, 信阳, 中国, 464000

*e-mail: zouzhibin@foxmail.com*

**【摘要】**本文的目的是设计一种基于 MIPS 指令集的 32 位 RISC 微处理器, 并采用 FPGA 进行功能验证。首先根据微处理器要实现的功能选择 MIPS 核心指令中的 34 条作为指令系统。然后将指令执行过程设计为取指令、指令译码、指令执行、存储器访问和数据回写 5 个阶段。对于数据冒险问题, 通过在流水线中设计数据转发单元和冒险检测单元来解决。由分支或跳转语句引发的分支冒险问题, 可以采用缩短分支延迟的方法, 在指令译码阶段增加比较器和数据转发单元并修改相应的地址选择逻辑来解决。软件仿真和 FPGA 硬件验证表明本设计的可行性。

**【关键词】**精简指令集微处理器; 流水线; FPGA; 控制单元; 软件仿真

### 1 引言

随着信息化进程的加快, 越来越多的领域需要使用微处理器。而微处理器又是计算机的核心部件, 是信息产业的基本部件。在当今的移动通信、网络设备、自动控制、信息家电等领域起着重要作用并且越来越被广泛的使用。然而, 由于各种原因我国信息产业中涉及微处理器的许多核心技术与产品仍然全部依赖国外, 不仅经济、技术上受制于人, 而且国家安全也面临威胁。因此, 研究和开发微处理器技术就显得特别有意义。

而另一方面, 随着超大规模集成电路 (Very Large Scale Integration, VLSI) 设计的普及和半导体工艺的迅速发展, 在一个硅片上已经可以实现一个更为复杂的系统, 这就是片上系统 (System on Chip, SoC)<sup>[1]</sup>。现场

可编程门阵列 (FPGA) 提供了一种在单个芯片上实现系统的有效手段, 所以设计微处理器时可以采用标准的硬件描述语言 (Hardware Description Language, HDL) 描述, 软件仿真通过后下载到 FPGA 开发板上验证, 因此这种基于 FPGA 的微处理器的设计方法易于实现而越来越受到人们的关注。

精简指令集 (Reduced Instruction Set Computer, RISC) 微处理器是近几年微处理器设计的主流。RISC 处理器的中心设计思想是尽量降低指令的数量及复杂性, 简化硬件设计。这样做有两个好处: 由于其选用的指令功能简单, 对应的微结构也就变得简单, 因此可以提高处理器的时钟频率; 由于结构简单, 流水线的效率非常容易提高, 从而可以降低每条指令的执行周期数。

RISC 体系结构使用长度相同、格式固定的指令系统，大多数指令都能够在一个机器周期内完成，便于流水线操作执行。RISC 体系结构只允许取数 (LOAD) /存数 (STORE) 指令访问主存，其它指令都是面向寄存器的操作。采用分离的指令和数据存储的哈佛结构，这样每周期内能同时提供一条指令和一个数据 [2][3]。

本文主要介绍了一种设计五级流水的 RISC 微处理器的方法，这个工作是为了评估用硬件描述语言 VHDL 快速设计 RISC 微处理器原型的可行性。

## 2 指令集设计

### 2.1 MIPS 指令集

在 RISC 指令体系中，有 ARM 指令集、MIPS 指令集、SPARC 指令集等等。由于 MIPS32 指令集中的大部分指令专利已经到期，并且 MIPS 公司提供了详细的指令编码文档，而且基于 MIPS 指令的操作系统、编译器和应用软件等资源非常丰富，为了方便以后应用的扩展，设计就采用了 MIPS32 指令集中一个子集作为设计微处理器的指令集。

MIPS32 指令集是规整的 32 位结构，每条指令的长度都是 32 位。所有指令共分为三类：立即数类型 (R 类型) 指令、寄存器类型 (I 类型) 指令和跳转类型 (J 类型) 指令 [4]，指令格式如图 1 所示。

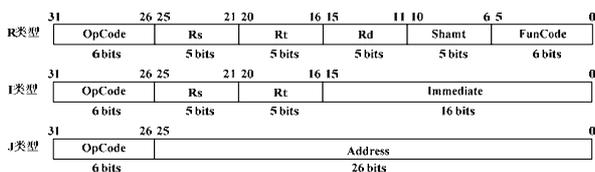


Figure 1. MIPS 32-bit instruction formats

图 1. MIPS 32 指令类型

### 2.2 待实现指令集

指令集的设计是根据微处理器要完成的特定功能来确定的。本设计的指令集包括整数的基本算术运算指令、逻辑运算指令、移位指令、条件测试指令、分支和跳转指令以及数据传送指令。经过在 MIPS32 核心指令集中选择、整理后，确定了待实现指令共 34 条，R 类型指令：nop、add、addu、sub、subu、and、or、xor、nor、slt、sltu、sll、srl、sra、sllv、srlv、srav、jr；I 类型指令：bltz、bgez、blez、bgtz、beq、bne、

addi、addiu、andi、ori、xori、slti、sltiu、lw、sw；J 类型指令：j。

## 3 RISC 微处理器设计

### 3.1 五级流水线设计

流水线 (pipeline) 是一种可以将多条指令的执行过程相互重叠的技术，目前它是提高微处理器运算速度的一个关键技术。当然在微处理器设计中并不是采用的流水线越多速度就越快。因为如果 CPU 执行的程序是一条跳转语句，在没有进行指令分析前，CPU 是不会知道它是否要跳转，那么在分析指令的时候，取指令功能模块已经把下一条指令取了进来。由于跳转指令的存在，这条后取的指令实际上是不应该被执行的，所以要从流水线上清除掉，才能从跳转指令要跳转的地方重新开始，这就造成了整个流水线的等待。于是流水线越长，速度越快，进入流水线中的指令也就越多，流水线的等待也就越明显，整个 CPU 的利用率也就越低 [5]。

按照参考文献 [2] 中的 DLX 体系结构，我们将 RISC 微处理器的流水线分为五级。

第 1 级：IF (Instructions Fetch) 取指令。

第 2 级：ID (Instructions Decode) 指令译码。

第 3 级：EX (Execute) 执行指令或地址计算。

第 4 级：MEM (Data Memory Access) 数据存储器访问。

第 5 级：WB (Write Back) 结果写回寄存器。

五级流水框图如图 2 所示。

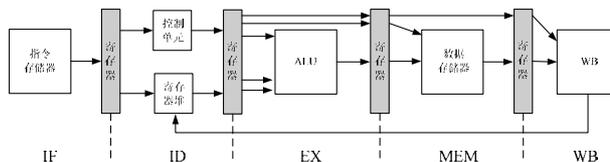


Figure 2. Block diagram of five stages pipeline

图 2. 五级流水框图

### 3.2 数据冒险

通过分析微处理器执行指令过程，我们发现如果前面的指令要修改某个寄存器的值，而后面两条指令的 Rs 或 Rt 寄存器恰好是要修改的那个寄存器，这个时候就会出现数据冒险。解决的方法就是将新的寄存器值直接转发到后面指令的 EX 阶段。由于 EX 阶段原先 ALU 的输入是从寄存器堆中读取的 Rs 和 Rt 寄存器的值，现在又有可能是前两条指令的计算结果在

MEM 段和 WB 段的保存，所以这里需要两个 3 选 1 的多路选择器帮助选择正确的操作数，多路选择器的控制输入信号也就是转发单元的输出信号，设计转发单元如图 3 所示。

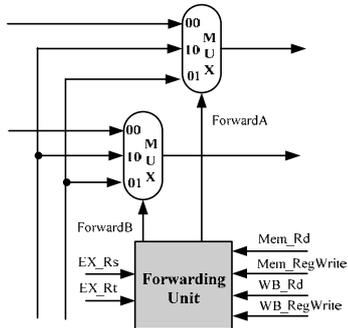


Figure 3. Block diagram of forwarding unit  
图 3. 转发单元设计框图

### 3.3 分支冒险

流水线的设计就意味着每个时钟周期内都要取指令，这样才能够保证流水线的正常运转。但是对于分支指令的执行，如果按照前面的数据通路设计，就只能在 MEM 流水阶段才能够确定是否发生分支。而此时分支指令之后的三条指令已经进入流水线，如果执行分支，这三条指令将都不能执行。像这种为了确定预取正确的指令而导致的延迟就叫做分支冒险。

这里将讨论通过缩短分支的延迟来解决分支冒险的设计方案。通过对要实现的 6 条分支指令的观察，发现分支指令的结果仅仅依赖于一些简单的比较测试操作。比如：是否相等（指令 BEQ、BNE）、符号的正负（指令 BGEZ、BGTZ、BLEZ、BLTZ），这些操作并不需要完整的 ALU 操作，完全可以设计个比较器对这些指令进行测试。而如果要提前确定分支的决策，就必须提前完成两个操作：计算分支的目的地址和判断分支指令的跳转条件。对于目的地址的计算需要有 PC 值和 IF\_ID 流水寄存器的指令立即数，那么只需要将计算用的加法器从 EX 阶段移到 ID 阶段；而对于分支的跳转条件判读，可以增加一个比较器，比较从寄存器堆读出的 Rs 和 Rt 寄存器的值。当然如果前两条指令的写寄存器与 Rs 或 Rt 寄存器相同，这就会存在数据冒险问题，那么也要引入数据转发单元来解决。所以可以将分支指令的判断放在 ID 阶段，如果发生分支转移，只需要将 IF 阶段的指令清除掉，这样就只清除了一条指令，流水线也只阻塞一个周期，与原设计相比改善了系统的性能，设计如图 4 所示。

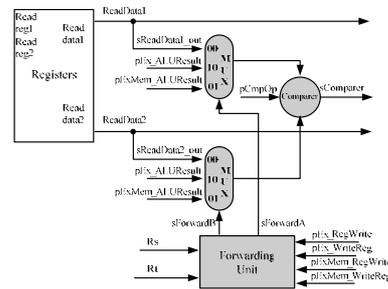


Figure 4. Block diagram of solving branch hazard  
图 4. 分支冒险解决方案设计框图

### 3.4 最终设计框图

最终完成的能够正确执行设计的 34 条 MIPS32 指令，采用五级流水方式运行的 RISC 微处理器的逻辑设计详图如图 5 所示。

## 4 仿真与验证

嵌入式 RISC 微处理器在完成了指令集设计、整体结构设计及各功能模块的设计之后，通过硬件描述语言 VHDL 对设计进行实现，最后并对这一实现进行模块级和系统级的软件仿真及 FPGA 验证<sup>[6]</sup>。

### 4.1 软件仿真

在软件 ModelSim 下编写相应的 test bench，对设计进行软件仿真，结果如图 6 所示。

通过观察仿真波形图中各个输出信号的波形，分析程序运行的过程，信号波形完全正确。因此可以说，RISC 微处理器流水线工作正常，控制单元、ALU 控制器、数据转发单元、冒险检测单元、分支冒险解决方案等模块设计正确。

### 4.2 FPGA 验证

FPGA 验证是在 Xilinx 的 Spartan 3AN 开发板上进行的。在 ISE9.2 集成开发环境下，导入设计的 VHDL 文件，用 CoreGen 生成指令存储器（ROM）和数据存储器（RAM），设置引脚约束，然后由 ISE 自带的综合软件 XST 将其转换成所选 FPGA 的网表，经过设计实现、布局、布线，最后构造出 FPGA 的配置文件，通过软件 iMPACT 和 JTAG 接口电缆对 FPGA 进行编程，将配置文件下载到 FPGA 芯片或开发板上的存储芯片中，进行硬件验证。本设计最大工作频率为 38.4MHz，占用资源数量如表 1 所示。

## 5 总结

本文设计并实现了一个能够支持 34 条 MIPS 指令的 32 位嵌入式 RISC 微处理器。因为现在的 FPGA 芯

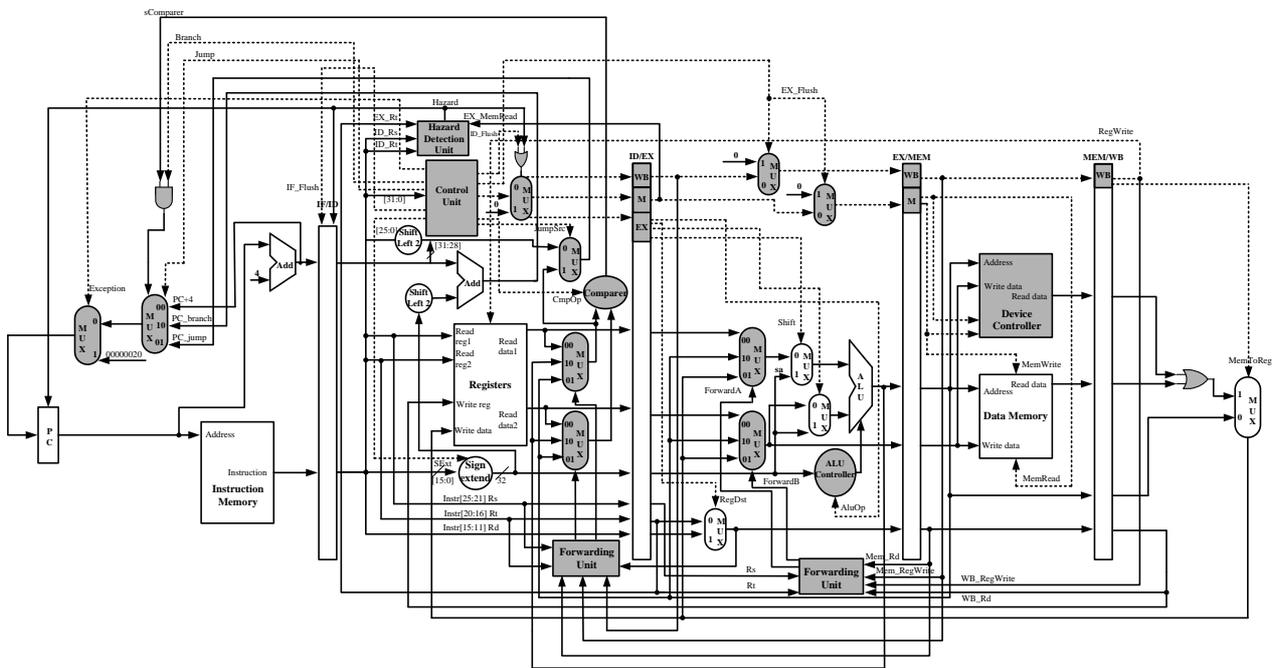
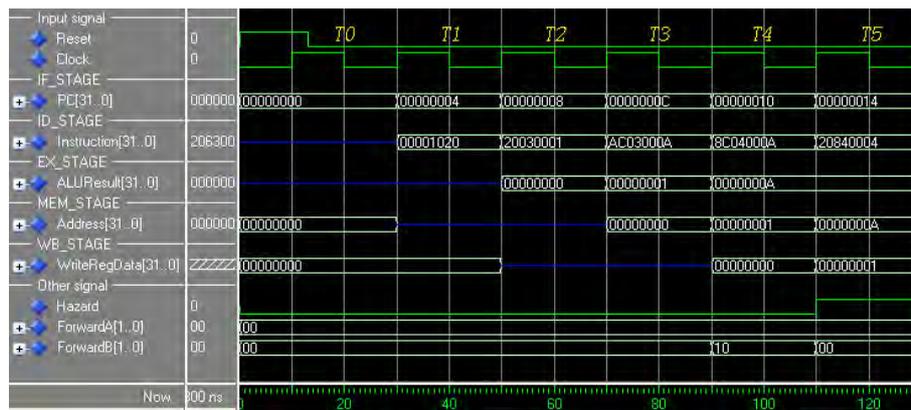
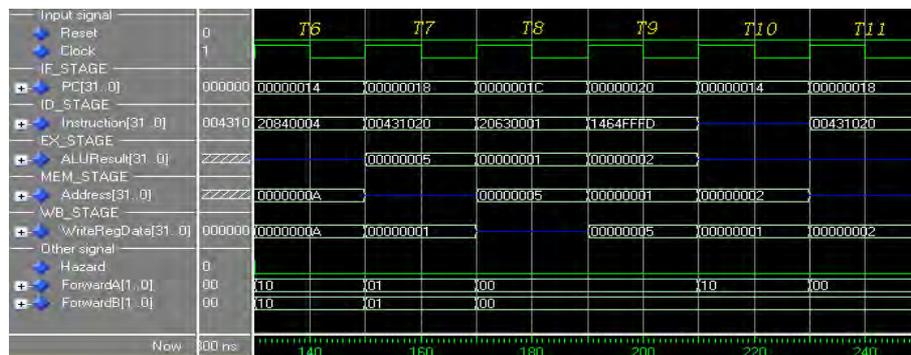


Figure 5. The final data path and control unit

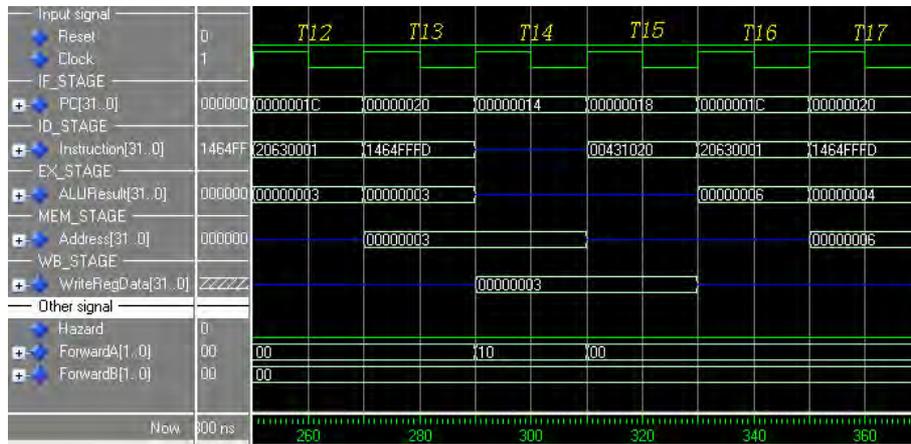
图 5. 最终设计的数据通路和控制单元



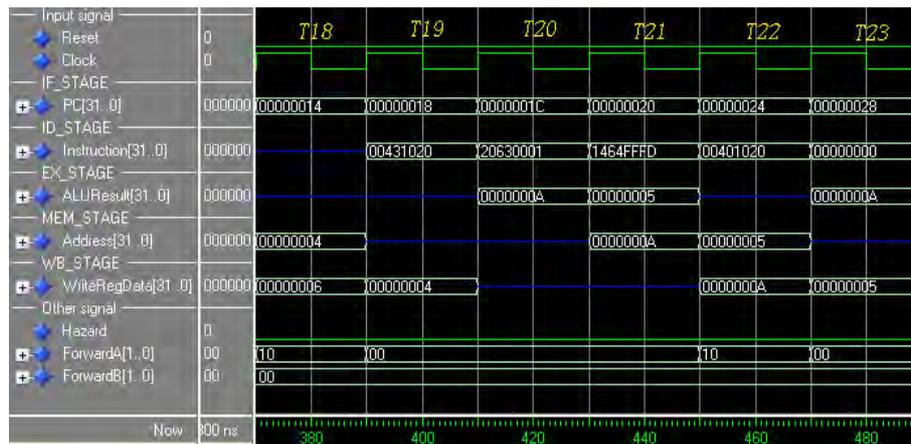
(a)



(b)



(c)



(d)

Figure 6. Simulation result of designed RISC microprocessor

图 6. 软件仿真波形图

Table 1. Resource usage

表 1. 资源统计表

资源	使用数	可用数	占有率
Slices	2644	5888	44%
Flip Flops	1507	11776	12%
4 input LUTs	4304	11776	36%
IOBs	11	372	2%
BRAMs	2	20	10%
GCLKs	2	24	8%

片已经可以为开发者提供足够的资源去设计微处理器，所以研究如何基于 FPGA 芯片去设计微处理器是一项有必要的工作。我们在这个工作中也总结了一些设计原则和经验，希望对以后的基于 FPGA 的实际应用设计能起到指导的作用。

References (参考文献)

[1] XU Tao LIANG Li-ping WANG Dong-sheng. Embedded SOC

system design[J], Computer Engineering and Design, 2006, 27(11), P205-207.  
 徐涛, 梁利平, 嵌入式 SOC 系统设计[J], 计算机工程与设计, 2006, 27(11), P205-207.  
 [2] John L. Hennessy, D. A. Patterson. Computer Architecture:A Quantitative Approach[M]. Fourth Edition. New York: Morgan Kaufmann Publishers, 2007.66-185.  
 [3] David A. Patterson, John L. Hennessy. Computer Organization and Design: The Hardware/Software Interface[M]. Third Edition. New York:Morgan Kaufmann Publishers, 2007.284-454.  
 [4] MIPS Corporation. MIPS32 Architecture for Programmers Volume I:Introduction to the MIPS32 Architecture, 2005, 7-15.  
 [5] Pravin S.Mane, Indra Gupta, M.K. Vasantha, Implementation of RISC Processor on FPGA[J]. in: Proceedings of IEEE International Conference on Industrial Technology, ICIT 2006. Washington, DC: IEEE Computer Society, 2006. P2096-2100.  
 [6] XU Ke, YANG Xue-fei, ZHU Ke-jia. Hardware Verification of a 32-Bit Embedded RISC Microprocessor[J], MICROELECTRONICS, 2003, 33(6), P502-505.  
 徐科, 杨雪飞, 朱柯嘉. 32 位嵌入式 RISC 处理器的硬件验证[J], 微电子学, 2003, 33(6), P502-505.