一种带 Cache 的嵌入式 CPU 的设计与实现*

东野长磊，戚 梅
（山东科技大学 信息科学与工程学院，山东 青岛 266510）

摘 要：基于 FPGA 平台实现了嵌入式 RISC CPU 的设计。根据项目要求，实现指令集为 MIPS CPU 指令集的一个子集，分析指令处理过程，构建了嵌入式 CPU 的 5 级数据通路。分析了流水线产生的相关性问题，采用数据前推技术和软件编译结合的解决方案。给出了控制单元、运算单元、指令 Cache 的实现与设计。在 FPGA 平台上实现并验证了 CPU 的设计。

关键词：嵌入式 CPU；流水线；数据相关；指令 Cache

中图分类号：TP368.1 **文献标识码**：A **文章编号**：1674-7720(2010)14-0017-03

The design and implementation of embedded CPU with Cache

DONGYE Chang Lei, QI Mei
(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China)

Abstract: In this paper, an embedded RISC CPU was designed based on FPGA platform. According to the project requirements, the instruction set of the cpu was a subset of MIPS instruction set. By analysising of instruction processing, the paper built five stages data path of pipeline CPU. By analysising of the pipeline-related problems, the paper adopted data forwarding technology and software compiler method. The paper gave the implementation and design of the control unit, arithmetic unit, instruction Cache. And the CPU was realized and verified on the FPGA platform.

Key words: embedded CPU; pipeline; data hazard; instruction Cache

随着集成电路设计和工艺技术的发展，嵌入式系统已经在 PDA、机顶盒、手机等信息终端中被广泛应用。嵌入式系统具有电路尺寸小、成本低廉、可靠性高、功耗低等优点，是未来集成电路发展的方向。而作为嵌入式系统核心的微处理器，其性能直接影响整个系统的性能。为了提高 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 流水线的设计成为决定其性能的关键。

MIPS(Microprocessor without Interlocked Pipeline Stages)是一种典型的 RISC (Reduced Instruction Set Computer)微处理器，在嵌入式系统领域中得到广泛的应用。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。与 CISC(Complex Instruction Set Computer)微处理器相比，RISC 具有设计更简单、设计周期更短等优点，并可以应用更多先进的技术，开发更快的下一代处理器。

1 基于 MIPS 指令集的 CPU 流水线结构

1.1 指令集的选取

设计实现了指令兼容 MIPS 系列 RISC 处理器的指令集。由于 MIPS32™ 指令集是开放的指令集，指令格式非常简单，按照指令格式可分为寄存器类型(R-type)指令、立即数类型(I-type)指令和跳转类型(J-type)指令。这三类指令均为 32 bit，而且指令操作码在固定的位置上。这种特点易于指令代码的拆分，易于流水线 CPU 的设计。

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，当需要与内存交换数据时，通过内存访问指令进行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：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。

* 基金项目：国家 863 课题重点项目(2009AA0627018)；山东科技大学“春蕾计划”(2008BZC016)

1.2 流水线的设计

在基本的 MIPS 处理器中有 5 个流水级，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 为对指令进行译码，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。按照这一流水线结构，本文设计实现一种较为通用的 MIPS CPU，通过 VHDL 语言实现，各模块之的关系如图 1 所示。

2 嵌入式 CPU 流水线中的相关性

由于指令以流水线形式并行处理，必产生指令相关性问题，一般存在三种相关：结构相关、数据相关和控制相关，引起流水线竞争。

结构相关问题是指由于硬件资源不足而导致流水线不畅通，例如只有一个存储器模块时，就不能对存储器同时取指令和数据。数据相关问题是指一条指令的后续指令要使用该条指令的结果。而控制相关问题是指转移指令从取指到转向目标地址要花几个时钟周期，但流水线 CPU 在每个周期都取指令。

解决结构相关问题的方法是尽量增加硬件电路资源，本设计采用哈佛架构，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-FF 构建寄存器予以避免，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 ALU 计算指令非常有效，但对于存储器读数据指令，如果下面的指令想立即使用其结果，则必须暂停流水线一个周期。至于控制相关，可以使用指令重组优化及延迟转移技术等软件编译方法解决。

3 关键模块的实现

3.1 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 要实现的指令集，确定出 ALU 的操作控制信号数据宽度为 5 bit，运算的数据位数为 32 bit。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。

3.2 控制单元的设计

控制单元要根据输入的指令码产生一系列的控制信号，用于控制数据通路上的多路选择器和各功能部件，保证每一条指令都能够正确执行。

控制单元的输入信号必须设计为 32 bit 的指令码，而输出信号则要根据需要进行设计。

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，发出相应的控制信号；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 之间的流水线寄存器。如果发生数据相关，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 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的流向，给出多路选择器的选择信号，选择将存储器读出的数据或 ALU 的运算结果写回寄存器组。

3.3 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 和 rt 及前面 2 条指令的操作数结果寄存器分别进行比较，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。

而对于 LOAD 指令发生的数据相关，必须等到 MEM 阶段完成之后才能得到有效的数据，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可以解决 LOAD 类型的数据相关。

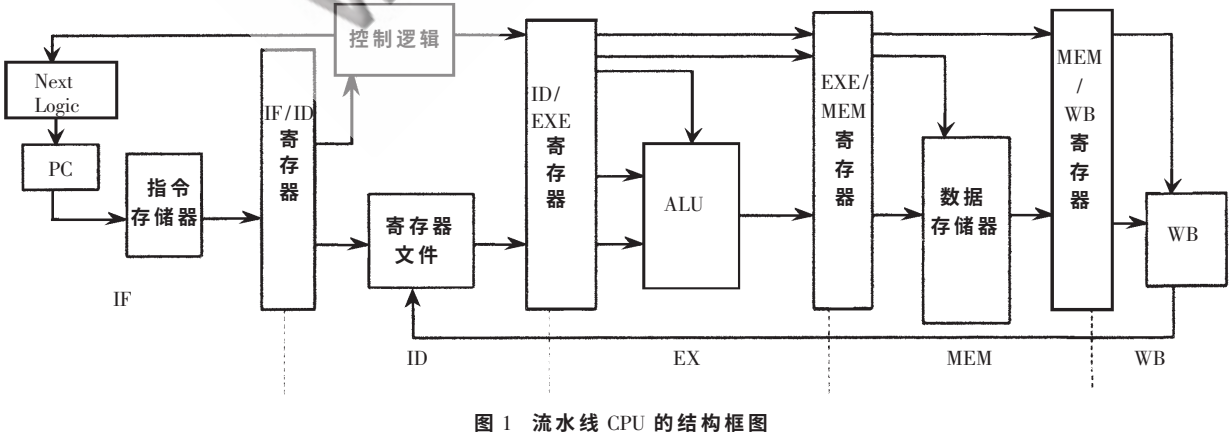

图 1 流水线 CPU 的结构框图

3.4 指令 cache 的实现

系统实现了一个容量为 2 KB 指令 Cache，每个 Cache 行为 16 B 数据，这样可以利用存储器的 16 B 的突发式传送。采用 2 路组相联方式，支持通写(Write Through)模式。由同步 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，如果不命中则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。

使用 VHDL 来设计和实现上述各关键模块。综合后的接口信号图如图 2 所示。

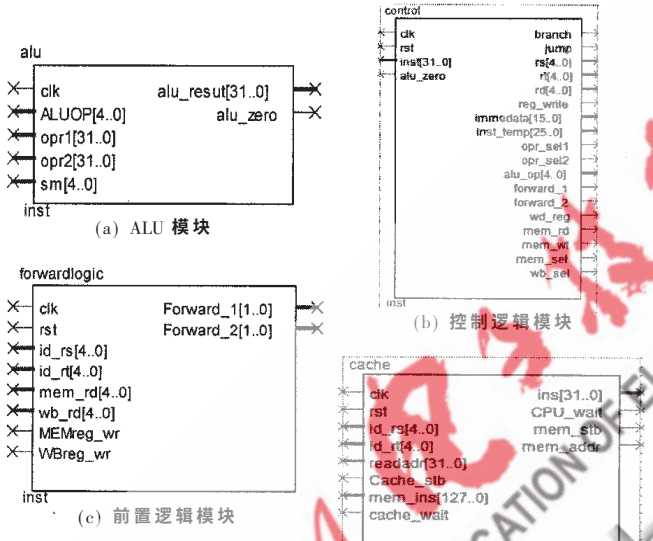

图 2 关键模块的接口信号图

对关键模块和其他模块进行融合，最后得到的 CPU 流水线结构图如 3 所示。

4 系统的仿真与验证

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 4 所示。

每个时钟周期为 10 ns，第一个时钟周期 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，第二条指令 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可以看出，数据前推成功。通过分析仿真波形图中各个输出信号的波形，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。

本文给出了流水线 CPU 的关键模块的 VHDL 实现，经过逻辑综合和仿真，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 的时钟频率下稳定的运行。

参考文献

[1] STALLINGS W. Reduced instruction set computer architecture[J]. Proceedings of the IEEE, 1998,76(6).

[2] BALCH M. Complete digital design: A comprehensive guide to digital electronics and computer system architecture[M]. 北京：清华大学出版社, 2004.

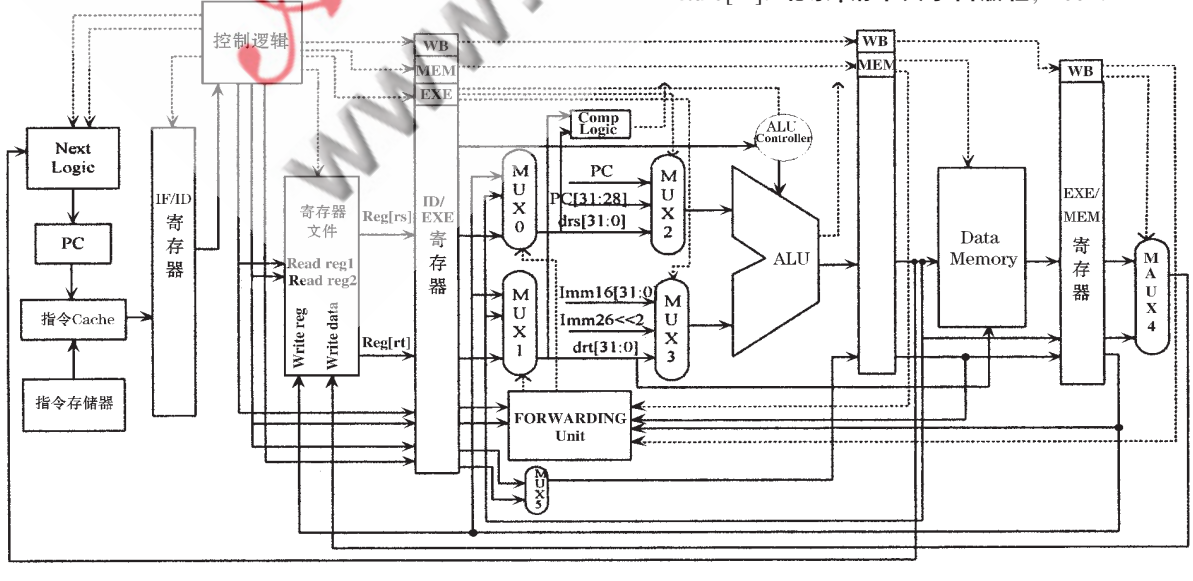

图 3 流水线 CPU 的逻辑结构图

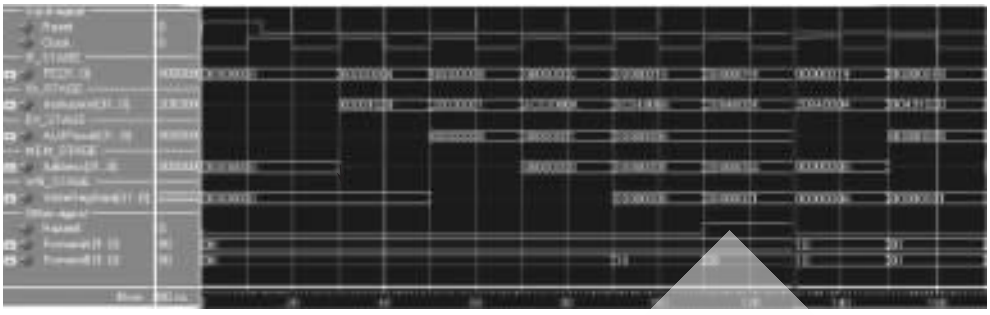

图 4 仿真波形图

[3] SWEETMAN D. See mips run(the second edition)[M]. 北京:机械工业出版社,2007.

[4] 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.

[5] 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.

[6] 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

[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.

(收稿日期:2010-04-13)

作者简介:

东野长磊,男,1978 年生,博士生,讲师,主要研究方向:嵌入式系统及应用。

戚梅,女,1973 年生,硕士,实验师,主要研究方向:计算机体系结构的研究与开发。