32 位 MIPS 架构的流水线 CPU 设计

赵春蕾[1]，乔东海[2]
(1. 苏州大学电子信息学院，江苏苏州 215006；2. 中国科学院声学研究所，北京 100190)

摘要：微处理器有很多种体系结构，本设计采用 MIPS(无内部互锁流水级的微处理器)体系结构，该结构是 RISC(精简指令集微处理器)的典型代表，在嵌入式和高端领域具有相当高的占有率，因此成为本次研究的选择。本次设计使用 Verilog HDL 硬件描述语言设计出一个 32 位 5 级流水线 MIPS 体系结构的 CPU，设计包含 R-type, I-type 和 J-type 三个类型大约 20 种基本指令。同时通过内部前推技术和延迟转移技术解决冲突，并加入了中断和异常处理模块用于响应外部中断和处理内部异常。最后，用 Verilog HDL 硬件描述语言设计出完整的 MIPS CPU 代码，并且在 Quartus II 软件上完成了仿真验证。

关键词：MIPS CPU；流水线；指令系统；Verilog HDL
中图分类号：TN402 文献标识码：A

Design of MIPS 32-bit pipelined CPU

ZHAO Chun-lei[1], QIAO Dong-hai[2]
(1. *School of Electronic and Information Engineering, Soochow University, Suzhou* 215006, *Jiangsu, China*;
2. *Institute of Acoustics, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: There are many kinds of microprocessor architectures and the MIPS (Microprocessor without Interlocked Piped Stages) architecture, which is the typical representative of the RISC (Reduced Instruction Set Computer), is a considerable occupation in the embedded and high-end applications. Thus, MIPS architecture becomes the microprocessor design of this study. This MIPS processor system design uses the Verilog as the HDL language to describe a 5 level 32-bit pipelined MIPS CPU, containing R-type, I-type and J-type three types of about 20 kinds of basic instructions. Meanwhile, the interrupt and exception module are added to resolve the interruption and error problems. Finally, the overall 32-bit MIPS system design uses Altera Quartus II software to complete the simulation verification.

Key words: MIPS CPU; pipeline; instruction set; Verilog HDL

0 引 言

CPU 设计是整个计算机系统设计中的核心设计，其性能的好坏是决定计算机性能的关键因素。基于指令系统的不同结构，处理器系统被分为复杂指令系统(CISC)和精简指令系统(RISC)[1]。CISC 是对那些具有复杂指令系统的 CPU 的总称。在 RISC 概念提出的同时，MIPS 概念也被提出，MIPS 是 Microprocessor without Interlocked Piped Stages 的缩写，意为无内部互锁流水级处理器。本文根据 MIPS 架构对 MIPS CPU 进行了原理分析与代码设计，包括中断与异常处理的模块，并通过内部前推和延迟转移技术解决流水线冲突，完成了 5 级流水线的 CPU 设计。

收稿日期: 2017-06-19; **修回日期**: 2017-06-26
基金项目：国家重大专项项目(2011ZX05008-005-02-02)
作者简介：赵春蕾(1994—)，女，安徽六安人，硕士研究生，研究方向为集成电路设计。
通讯作者：赵春蕾，E-mail: zhaochunlei0564@163.com

1 MIPS CPU 架构设计

MIPS 系统的指令集共包含三种类型，分别是 R-Type（寄存器类型）、I-Type（立即数类型）和 J-Type(转移跳转类型)，其格式如图 1 所示。

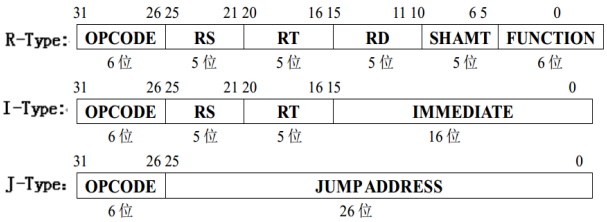


图 1 MIPS 指令集类型
Fig.1 MIPS instruction formats

图 1 中，OPCODE 为指令的操作码，位数固定为 6 位；RS 为源寄存器号，RT 表示源/目的寄存器号、RD 为目的寄存器号；SHAMT 是位移指令中移位的位数；指令扩展码用 FUNCTION 表示；IMMEDIATE 是 16 位立即数；JUMP ADDRESS 是

26 位转移地址。

MIPS CPU 执行指令时，在一个时钟周期内，有 5 条指令并行执行，其具体的结构框图见图 2。框图中为每一级的主要模块，其中 IM 表示指令存储器，RF 为寄存器堆，ALU 是算术逻辑单元，DM 表示数据寄存器。IF 意为取指令，利用程序计数器 PC 的值到存储器中取出响应指令；ID 为译码，对上一级取出的指令进行译码；EXE 是执行，对取出的操作数进行运算。MEM 是访问存储器，从存储器中读出或者写入数据。WB 为写回，将得到的结果写回寄存器堆中。

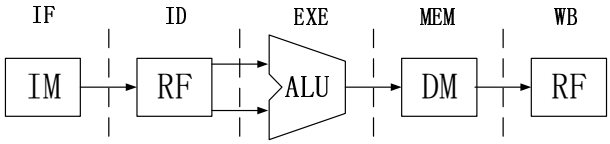


图 2　五级流水线结构框图
Fig.2　Block diagram of five stages pipeline

其中 EXE 级是 CPU 指令执行的核心阶段，完成最重要的运算步骤，本论文完成的运算有加减法、逻辑运算、设置高位和移位。其中最主要的 ALU 的逻辑图如下图 3 所示。由图可得 ALU 在运行时根据 aluc 的信号决定做何种运算，运算完成后再根据 aluc 决定输出何种结果。

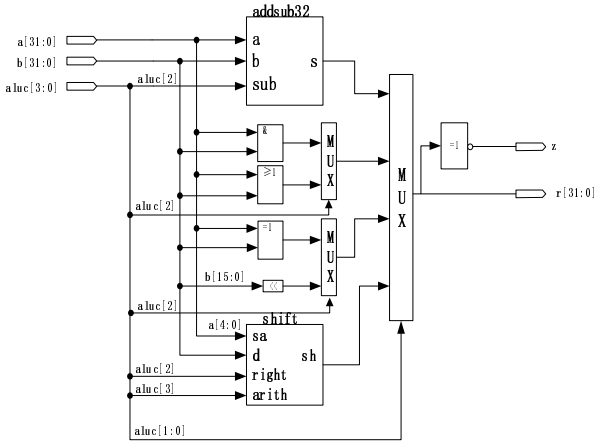


图 3　ALU 逻辑电路图
Fig.3　ALU logic circuit

流水线 CPU 设计还存在数据相关和控制相关问题。当流水线 CPU 在一条指令还未执行完成时，下一条指令便需要使用它的结果，这样指令之间就出现了数据相关问题，解决方法是使用内部前推技术将 EXE 级计算出来的最新结果送至下一条指令的 ID 级；CPU 在执行跳转指令时会存在控制冲突，本次设计采用延迟转移技术来解决控制冲突，转移指令执行完成后，不论是否转移，在该转移指令的后续一条指令都要执行，这一位置通常被称作“延迟槽”。通过以上对流水线每一级的设计分析，再结合最后流水线相关问题的解决对策，可以设计出 MIPS CPU 的整体架构如图 4 所示。

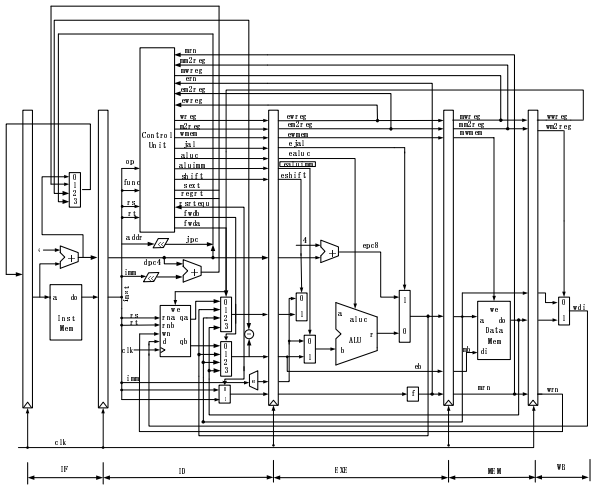


图 4　五级流水线结构框图
Fig.4　Block diagram of five stages pipeline

2　异常和中断处理

异常(Exception)和中断(Interrupt)是在 CPU 运行的过程中可能会出现的各种不可预知的事件，这些事件会干扰程序的正常运行。通常情况下，异常来自于 CPU 的内部，比如除数为 0，计算结果溢出等；而中断来源于 CPU 的外部，比如键盘中断鼠标中断等。当异常和中断出现时，CPU 一般会停止执行当前程序并保存中断异常现场，转去执行异常和中断的处理程序，即异常或中断响应。。

本论文设计的 CPU 共处理 3 个异常：结果溢出、未实现的指令和系统调用；采用查询中断方式，一次只处理一个外部中断请求。CPU 在响应中断和异常时需要暂停当前正在执行的程序，进入中断处理，完成处理程序后再返回。当 CPU 在处理异常和中断时需要使用三个寄存器，分别是 Cause 寄存器、Status 寄存器和 EPC 寄存器，如图 5 所示。以上 3 个寄存器的宽度均为 32 位。Cause 寄存器中的 BD 通常情况下置 0，但当引起异常的指令是转移跳转指令时，BD 置 1；ExcCode 用来表示中断和异常的种类来源。

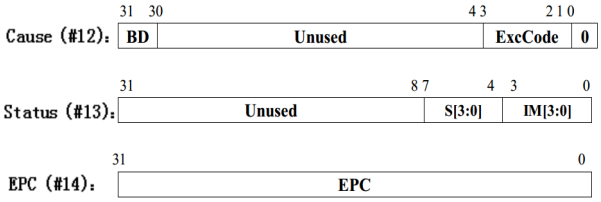


图 5 异常和中断相关的 3 个寄存器

Fig.5 3 registers associated with exception and interrupt

CPU 在响应中断时需要暂停当前正在执行的程序，进入中断处理，完成处理程序后再返回。中断请求到来时，不论是处于延迟槽中的指令还是正在执行转移指令，都立即响应；CPU 在出现异常时，与出现中断相同，将引起异常的指令地址写入 EPC 寄存器；若处在延迟槽中则将转移跳转指令的目标地址写入 EPC 寄存器，同时将 Cause 寄存器的 BD 置 1。实现异常和中断处理的电路逻辑框图如图 6 所示。

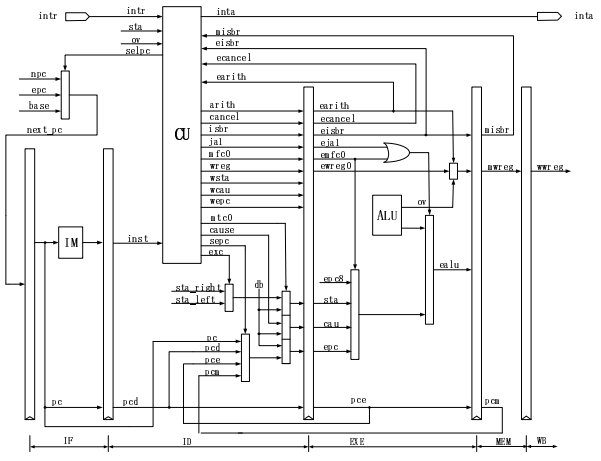


图 6 异常和中断处理架构

Fig.6 Exception and interrupt's architecture

上图电路在原有 CPU 的基础上添加的部分主要围绕 Status、Cause 和 EPC 三个寄存器进行设计。添加了以上三个寄存器的使能端，并通过四选一多路选择器先选出写入 EPC 寄存器的数据，其中 mtc0 和 mfc0 指令用来读写上述三个寄存器。至此，MIPS CPU 对异常和中断的处理电路部分的设计已经完成。

3 MIPS CPU 的仿真与验证

3.1 MIPS CPU 仿真结果

对 MIPS CPU 的仿真验证是通过用一组指令来验证的，这组指令包含对整体流水线各个指令的执行情况和各类冲突的解决状态的验证。

使用 Quartus II 工具对设计代码进行仿真，使用该工具提供的 LPM 的 ram 模块 lpm_ram 实现数据存储器；用 rom 模块 lpm_rom 来实现指令存储器。通过对 RAM 和 ROM 进行初始化来设定所要执行的指令和所需要的数据。指令存储器和数据存储器的初始化文件见下图。

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	3C010000	34240050	0C00001B	20050004	AC820000	8C890000	01244022	20050003
8	20A5FFFF	34A8FFFF	39085555	2009FFFF	312AFFFF	01493025	01494026	01463824
16	10A00003	00000000	08000008	00000000	2005FFFF	000543C0	00084400	00084403
24	000843C2	08000019	00000000	00004020	8C890000	01094020	20A5FFFF	14A0FFFC
32	20840004	03E00008	00081000	00000000	00000000	00000000	00000000	00000000
40	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
48	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
56	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

图 7 指令存储器的初始化文件

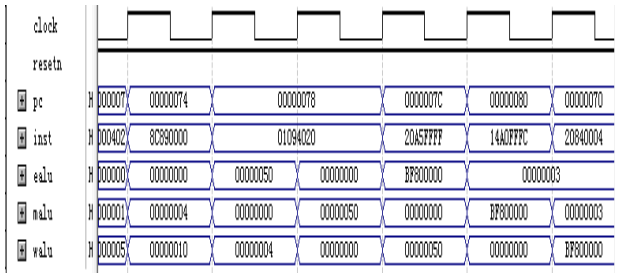
Fig.7 The initialization file for instruction memory

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
08	00000030	0000003C	00000054	00000068	00000000	00000000	00000000	00000000
10	00000000	00000000	00000002	7FFFFFFF	000000A3	00000027	00000079	00000115
18	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

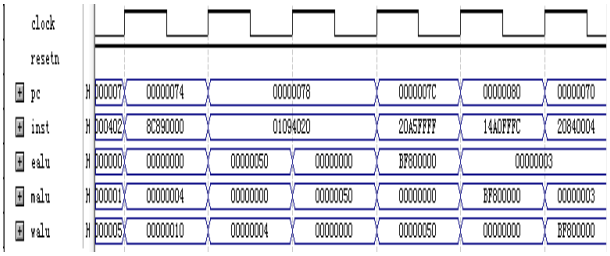
图 8 数据存储器的初始化文件

Fig.8 The initialization file for data memory

图 9 是 MIPS CPU 在执行以上初始化文件时，使用 Quartus II 工具仿真的输出结果。图中的时钟 clock 设定为 50MHz，pc 表示程序计数器值，同时也代表第一级取指令 IF 级的执行情况；inst、ealu、malu、walu 分别表示译码 ID、执行 EXE、访问存储器 MEM 和写回 WB 这接下来的四个流水级的执行状况。由图 9(a)可得流水线 CPU 的指令按顺序依次执行，在跳转指令到来时，PC 值由 0x0000000C 跳转至 0x0000006C；同时，当 PC 值为 0x00000078 时，由于出现了数据冲突，流水线暂停一个周期，如图 9(b) 所示。



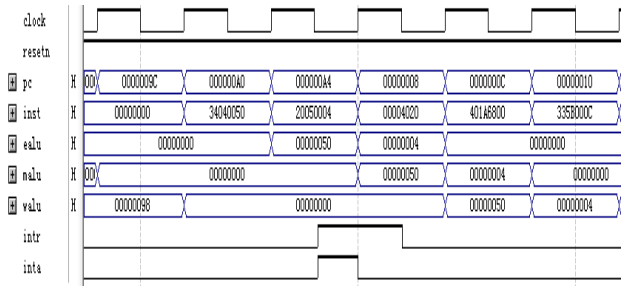
(a)



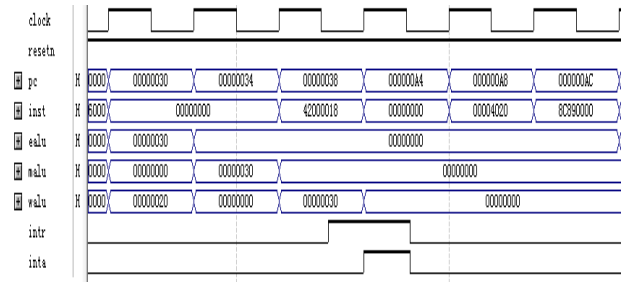
(b)

图 9　MIPS CPU 仿真结果
Fig.9　Simulation result of MIPS CPU

异常和中断处理的仿真波形过长，所以此处仅对通常情况下的中断进行分析，其仿真波形如图 10 所示。时钟 clock 同为 50MHz，pc、 inst、ealu、malu、walu 分别表示取指令 IF、译码 ID、执行 EXE、访问存储器 MEM 和写回 WB 五个流水级的执行情况。



(a)



(b)

图 10　异常和中断处理仿真结果
Fig.10　Simulation result of exception and interrupt

图 10(a)中断发生时处于 ID 级的指令为 0x000000A0，所以接下来的指令 0x000000A4 就被废弃，转而进入中断异常处理程序的入口 0x00000008，图 10(b)中，在中断处理程序执行完毕后，PC 值返回到 0x000000A4。由此验证异常和中断处理程序的正确性。

3.2 资源占用情况

本设计采用 ALTERA 公司的 EP4CE22F17C6 将整个 MIPS CPU 下载到可编程芯片内部，验证了绝大部分指令功能。MIPS CPU 的资源占用情况如表 1 所示。

表 1　资源统计表
Table 1　Resource usage

Resource	Usage
Total logic elements	30417
Total combinational functions	9957
Dedicated logic registers	26108
Total registers	26108
Total pins	68
Total memory bits	261888

由该表可得本文设计的 MIPS CPU 耗尽了 30417 个 LE，使用寄存器共 26108 个，相对其他 CPU 而言，MIPS CPU 使用更多的寄存器，但也具有更高的速度。

4　结 论

本文设计了一个 32 位的流水线 MIPS CPU，主要工作有：从 MIPS 经典指令系统中选出了具有代表性的指令组成了本次设计的指令集系统，且系统主要完成的是对于整型数据的处理；将 MIPS CPU 的指令执行过程划分为 5 个流水级；采用内部前推和延迟转移技术解决了数据相关和控制相关问题；设计了精确中断和异常事件处理模块用来处理外部的中断和内部的异常情况。最后，用 Verilog HDL 硬件描述语言设计出了完整的 MIPS CPU 代码，并且在 Quartus II 软件上完成了仿真验证，给出了仿真波形和相应的分析。

参　考　文　献

[1] 李亚民. 计算机原理与设计[M]. 北京：清华大学出版社, 2011.
[2] David A. Patteroson, John L. Hennessy. Computer organization and Design: The Hardware/Software Interface[M]. Fifth Edition China Machine Press, 2015.
[3] Tyson, Romas A L, Siti Intan P, et al. A Pipelined Double-Issue MIPS Based Processor Architecture[C]// Intelligent Signal Processing and Communication Systems, 2009. ISPACS 2009. International Symposium. 2009:583-586.
[4] D. Harris, M. Harris. Digital design and computer architecture [M]. Beijing: Chian Machine Press, 2014.
[5] SungMo (Steve) Kang, Yusuf Leblebici, Chul Woo Kim. CMOS Digital Integrated Circuits Analysis & Design[M]. Digital integrated circuit design, 北京：人民邮电出版社, 2010: 86-87.