U10M11007 试点班实验报告

指令分析报告/XX 设计报告

姓 名: 王 嘉 利
学 号: 2018302278
班 号: 10011801

CS 11007 计算机组成与体系结构
(春季, 2020)

西北工业大学
计算机学院
ERCESI

2020 年 5 月 10 日

摘 要

这次实验主要整理和分析了 SMIPS 指令，将 MIPS 指令整理在 EXCEL 中，并且简单分析了数据通路特征

版 权 声 明

目 录

1 概述 **5**

2 指令译码 **6**

2.1 运算指令 6

2.1.1 算术运算 逻辑运算 6

2.1.2 移位运算 7

2.2 分支指令 7

2.2.1 无条件跳转 7

2.2.2 有条件跳转 7

2.3 访存指令 7

2.4 数据移动指令 8

2.5 特权指令和自陷指令 8

3 数据通路特征 **8**

3.1 运算指令 8

3.2 分支跳转 9

3.2.1 有条件跳转 9

3.2.2 无条件跳转 9

3.3 访存指令 9

3.3.1 load 9

3.3.2 store 9

3.4 数据移动指令 9

3.5 关于 PC 10

3.6 关于 RF 10

3.7 关于存储器 10

4 实验过程记录 **10**

4.1 问题与解决方案 10

1 概述

SMIPS 指令可以大概分为运算指令（逻辑运算、算术运算、移位运算）、分支跳转指令、访存指令、数据移动指令、和自陷指令。因为指令的数量是比较少的，所以可以从指令的编码规律来考虑译码。再按不同指令类别考虑数据通路。

2 指令译码

对于 R-type 指令，区分指令依赖于 func 段的编码，而对于 I-type 和 J-type 依赖 opcode 段。不难发现，可以将 func 段和 opcode 段分成前三位和后三位来译码。

2.1 运算指令

算术运算指令由 R-type 和 I-type 两类指令组成

2.1.1 算术运算 逻辑运算

R-type

可以将 R-type 的算术运算和逻辑运算分为

```
class1 func[5:3] 100 : ADD ADDU SUB SUBU   AND NOR OR XOR
class2 func[5:3] 101 : SLT SLTU
class3 func[5:3] 011 : DIV DIVU MULT MULTU
```

对于第一类，`func[2]`可以将其分为算术运算和逻辑运算，对于算术运算 `func[1]`区分加减法,`func[0]`区分有无符号。

对于其他两类都是算术运算,`SLT SLTU`将比较结果写回`rt`寄存器,其`func[2:0]`与相应的`SUB SUBU`相同；对于乘法和除法指令再数据通路上与 class1，class2 有差异，是将计算的结果写回到`HI LO`寄存器

另外，区分`ADD(U) SUB(U) SLT(U) DIV(U) MULT(U)`的有无符号都可以用 `func[0]`

I-type

可以发现下面这些指令的`opcode[5:3]`都是`001`

```
ADDI ADDIU SLTI SLTIU
ANDI LUI   ORI  XORI
```

另外可以发现除了`LUI`, 其他指令的`opcode[2:0]`可 R-type 中相同功能的指令的`func[2:0]`是一样的

LUI 指令 与其他指令一些差异，写回rt寄存器的值是 $imm||0_{16}$，因此在数据通路可能需要一个单元完成数值的拼接。

2.1.2 移位运算

移位运算指令是SLLV SLL SRAV SRA SRLV SRL,可以发现他们的opcode[5:3]都是 0，opcode[2]区分使用sa还是R[rs]作为移位量，opcode[1]区分左移和右移，opcode[0]区分逻辑移位还是算术移位

2.2 分支指令

2.2.1 无条件跳转

有四个无条件跳转指令JR JALR J JAL

对于JR JALR是 R-type 指令,func[5:3]都是001；对于J JAL是 J-type 指令，opcode[5:3]都是 0

另外func[0]（或者opcode[0]）区分了是否要写rt寄存器

2.2.2 有条件跳转

BEQ BNE BGEZ BQTZ
BLEZ BLTZ BGEZAL BLTZAL

可以发现这些指令的opcode[5:3]都是 0（与J JAL在这一点上相似）

需要注意的是只有BEQ BNE是需要读rs rt寄存器的，其他指令只需要读 rs, 而且rt段时给定的。另外对于BGEZ BLTZ BGEZAL BLTZAL 这四个指令的op段都是000 001，要靠rt段来区分

2.3 访存指令

LB LBU LH LHU LW
SB SH SW

对于 lowd 指令opcode[5:3]都是100，opcode[2]区分了有符号扩展和无符号扩展

对于 store 指令opcode[5:3]则是101

2.4 数据移动指令

MFHI MFLO MTHI MTLO

这四个指令是 R-type 指令，func[5:3]=010，func[1]区分 Hi、LO，func[0]区分数据移动的方向

2.5 特权指令和自陷指令

特权指令	ERET MFC0 MTC0	opcode=010 000
自陷指令	BREAK SYSCALL	func[5:3]=001

3 数据通路特征

3.1 运算指令

R-type

对于 R-type 类型的运算指令的数据通路的共同特征可以总结为，从 RF 中取出rt rs的值，func段指定 ALU 的运算规则，将结果写回 rd（乘法和除法除外，乘除的结果写回 HI LO 寄存器，而且其他指令的计算结果是 32bit）

I-type

对于 I-type 类型的运算指令有两类，一类是算术运算，另一类是逻辑运算。对于算术运算Imm扩展都是符号扩展，对于逻辑运算Imm扩展都是无符号扩展。在指令译码部分已经提到了这两类的opcode[5:3]=001,而opcode[1:0]进一步功能划分，可以发现opcode[2]是可以作为符号扩展还是无符号扩展的标志

I-type 类型的数据通路特征则是，从 RF 中取出rs，将立即数扩展至 32bit，将这两个数送入 ALU 得到一个输出，将运算结果写回 rt（对于 LUI 比较特殊，可能可以认为是从 $0 取出 0，和 Imm 运算，因为对于其他情况都要扩展立即数，所以，可能还可以认为 LUI 也是将 Imm 进行扩展后送入 ALU，这样就可以简化设计）

3.2 分支跳转

3.2.1 有条件跳转

这类指令需要从 RF 中读出 rs 的值，对于BEQ BNE还需要读出 rt 的值，对于其他六个指令不需要。另外，在译码阶段还需要把 offset 进行符号扩展。在执行阶段算出是是否要跳转，产生一个选择信号。对于BGEZAL BLTZAL还需要在写回阶段将pc+8写回 $31

3.2.2 无条件跳转

对于J JAL不需要在译码阶段读寄存器,但是需要将target和 pc 拼接成 32bit; 对于JR JALR在译码阶段读寄存器 rs 得到跳转目标。对于JAL JALR需要在写回阶段将pc+8写回 $31

3.3 访存指令

3.3.1 load

在译码阶段需要从读出 base 寄存的值，并对 offset 进行有符号扩展，在执行阶段得到地址，在访存阶段得到要写到 rt 寄存器的数据，在写回阶段将读出的数据写回寄存器。因为读出的数据可以一个字节、半字、字，也就是有LB LH LW等这三种指令，对读出的数据要进行符号扩展，那么在数据通路中就应该再有一个扩展单元，把从内存读到的数扩展。

3.3.2 store

在译码阶段、执行阶段和 load 类似，在访存阶段需要把数据 rt 寄存器的一个字节、半字或字写入到存储器。

3.4 数据移动指令

MFHI MFLO需要在写回阶段把 HI 或 LO 寄存器中的值写回到 rd 寄存器

MTHI MTLO在译码阶段读出 rs 的值，在写回阶段写回到 rd 寄存器

3.5 关于 PC

每拍需要从跳转指令的跳转目标和 PC+4 中选择一个（要考虑 ERET）

3.6 关于 RF

通过上述的分析可以确定 RF 需要一个写端口和两个读端口，写入的数据可能来自译码阶段的立即数扩展（LUI）、ALU 计算的结果、存储器中读出的值、来自 HI，LO 寄存器或者来自协处理器。

3.7 关于存储器

要提供一个地址接口，输出接口，写入数据端口（还有 WRITE_EN）。地址来自于对 offset 扩展。另外还需要考虑`LB LH LW`，以及`SB SH SW`，他们在读出、和写入的位宽不一样。

4 实验过程记录

将 SMIPS 的指令按 R-type，I-type 和 J-type 整理，以及按功能整理

4.1 问题与解决方案

问题 对于 MIPS 怎么处理异常、协处理器怎么控制 CPU，不是特别了解

解决方案 查看 MIPS 手册了解了大概

参考文献

MIPS Architecture For Programmers Volume I
MIPS Architecture For Programmers Volume II
MIPS Architecture For Programmers Volume III
SEE MIPS RUN