

北京邮电大学 计算机学院
《计算机系统结构》实验报告

姓名 王睿嘉

学号 2015211906

班级 2015211307

实验五 指令调度与延迟分支

一、实验目的和环境描述

1. 实验目的

- 1) 加深对指令调度技术的理解；
- 2) 加深对延迟分支技术的理解；
- 3) 熟悉用指令调度技术解决流水线数据冲突的方法；
- 4) 进一步理解指令调度技术对 CPU 性能的改进；
- 5) 进一步理解延迟分支技术对 CPU 性能的改进。

2. 实验平台

指令级和流水线操作级模拟器 MIPSsim。

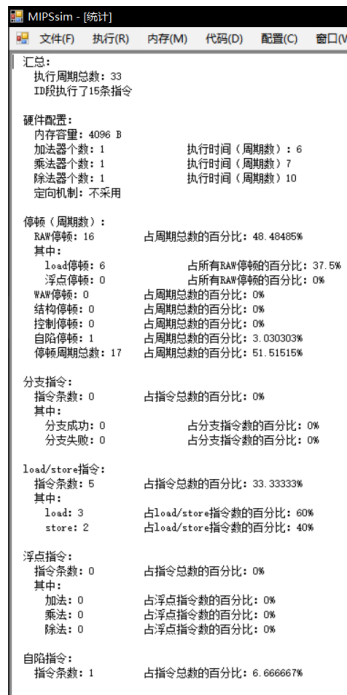
二、实验内容和步骤描述

- 1) 启动 MIPSsim；
- 2) 根据实验二相关知识中关于流水线各段操作的描述，进一步理解流水线窗口中各段的功能，掌握各流水线寄存器的含义；
- 3) 选择“配置”→“流水方式”选项，使模拟器工作在流水方式下；
- 4) 用指令调度技术解决流水线中的结构与数据冲突：
 - ① 启动 MIPSsim；
 - ② 使用 MIPSsim 的“文件”→“载入程序”选项加载 schedule.s；
 - ③ 通过“配置”→“定向”选项，关闭定向功能；
 - ④ 执行所载入的程序。

时钟周期图如下：



统计情况如下：



由上可知，RAW 停顿周期数为 16。其中，load 停顿周期数为 6。

发生冲突的指令组合为：

```
ADDIU    $r1,$r0,56
LW       $r2,0($r1)      #与上条指令发生 RAW 冲突
ADD      $r4,$r0,$r2     #与上条指令发生 RAW 冲突
SW       $r4,0($r1)      #与上条指令发生 RAW 冲突
LW       $r6,4($r1)
ADD      $r8,$r6,$r1     #与上条指令发生 RAW 冲突
MUL      $r12,$r10,$r1
ADD      $r16,$r12,$r1   #与上条指令发生 RAW 冲突
ADD      $r18,$r16,$r1   #与上条指令发生 RAW 冲突
SW       $r18,16($r1)    #与上条指令发生 RAW 冲突
LW       $r20,8($r1)
MUL      $r22,$r20,$r14  #与上条指令发生 RAW 冲突
MUL      $r24,$r26,$r14
```

程序执行的总时钟周期数为 33。

- ⑤ 采用调度技术对程序进行指令调度，消除冲突。将修改后的程序重命名为 after_schedule.s。

代码如下：

```
.text
main:
ADDIU    $r1,$r0,A
MUL      $r22,$r20,$r14
LW       $r2,0($r1)
MUL      $r24,$r26,$r14
ADD      $r4,$r0,$r2
LW       $r6,4($r1)
```

```

SW      $r4,0($r1)
ADD     $r8,$r6,$r1
MUL     $r12,$r10,$r1
ADD     $r18,$r16,$r1
ADD     $r16,$r12,$r1
SW      $r18,16($r1)
LW      $r20,8($r1)
TEQ     $r0,$r0

```

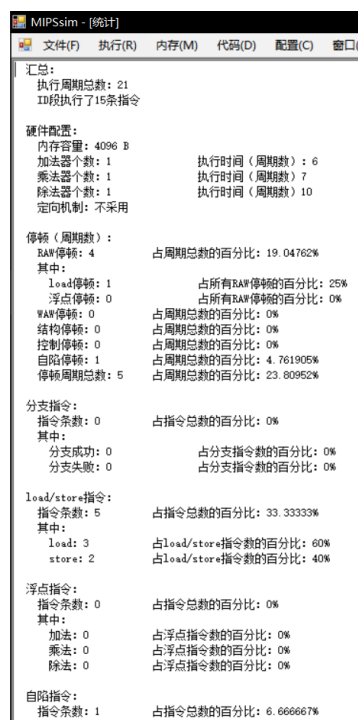
```
.data
```

```
A:
```

```
.word 4,6,8
```

⑥ 载入 after_schedule.s。

统计情况如下：



程序执行的总时钟周期为 21。

⑦ 比较调度前后的性能。

调度前的总时钟周期为 33，调度后为 21。重新组织指令顺序，可以消除部分数据冲突。

指令调度的优劣直接影响 CPU 性能的好坏，好的指令调度可以让停顿周期大幅减少。

5) 使用延迟分支技术减少分支指令对性能的影响：

① 在 MIPSsim 中载入 branch.s 样例程序；

② 通过在“配置”→“延迟槽”选项，关闭延迟分支功能；

③ 执行该程序。

分支延迟的时刻为第 13 和 28 周期，程序执行的总时钟周期数为 38。

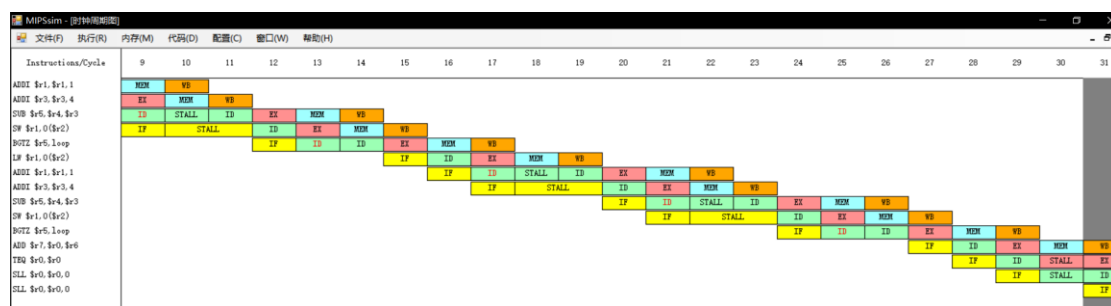
④ 假设延迟槽为一个，对 branch.s 程序进行指令调度。将修改后的程序重命名为 delayed_branch.s；

代码如下：

```
.text
main:
ADDI $r2,$r0,1024
ADD $r3,$r0,$r0
ADDI $r4,$r0,8
loop:
LW $r1,0($r2)
ADDI $r1,$r1,1
ADDI $r3,$r3,4
SUB $r5,$r4,$r3
SW $r1,0($r2)
BGTZ $r5,loop
ADD $r7,$r0,$r6
TEQ $r0,$r0
```

- ⑤ 载入 delayed_branch.s，执行该程序。

时钟周期图如下：



程序执行的总时钟周期数为 32。

- ⑥ 比较延迟分支前后的性能。

延迟分支前的总时钟周期为 38，之后为 32。在使用延迟槽后，指令在运行到跳转指令时，不会出现延迟等待，从而提高了 CPU 性能。

三、实验结果

使用指令调度和延迟分支技术，均能使总时钟周期数减少，从而提高执行效率。

四、实验心得

实践出真知，本次实验是对课堂和书本所学知识的补充。通过数据的比较，强烈感受到了指令调度和延迟分支技术对于提高 CPU 利用率的显著作用。平时应注重分析指令间的相关性，从而得到较优的代码。同时，通过自己动手、亲力亲为编写 MIPS 语句，加深了对汇编语言的理解和记忆，收获颇丰。