计算机系统结构实验

实验（二）流水线与流水线中的冲突

姓　　名	熊恪峥
学　　号	22920202204622
日　　期	2023年3月23日
学　　院	信息学院
课程名称	计算机系统结构

实验（二）流水线与流水线中的冲突

目录

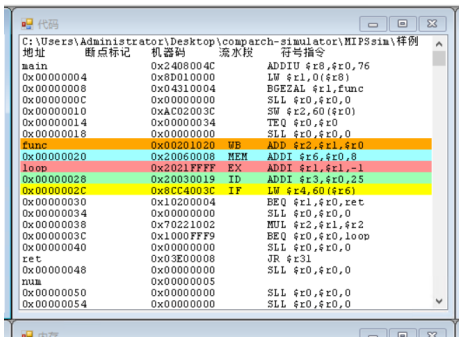
1 实验目的

1. 加深对计算机流水线基本概念的理解；
2. 理解MIPS结构如何用5段流水线来实现，理解各段的功能和基本操作；
3. 加深对数据冲突、结构冲突的理解，理解这两类冲突对CPU性能的影响。
4. 进一步理解解决数据冲突的方法，掌握如何应用定向技术来减少数据冲突引起的停顿。

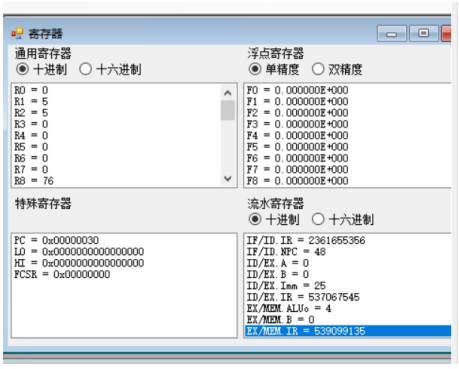
2 实验内容

2.1 观察和分析结构冲突对CPU性能的影响

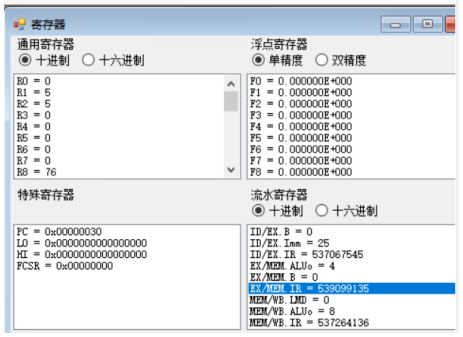
当执行到第13个时钟周期时，各段分别正在处理的指令如图 1a。此时各流水寄存器中的内容如图 1c。时钟周期图如图 1d。



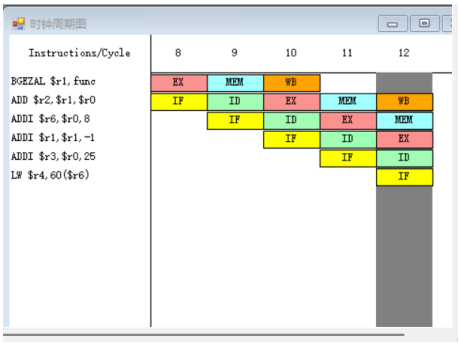
(a) 各段正在处理的指令



(b) 各流水寄存器中的内容(1)



(c) 各流水寄存器中的内容(2)



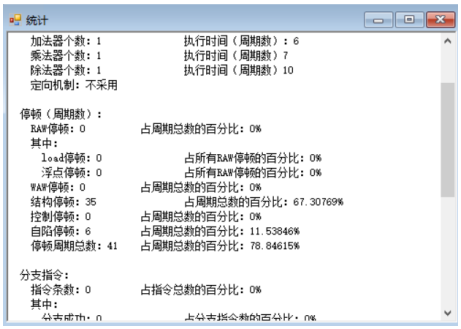
(d) 时钟周期图

加载`structure_hz.s`，执行该程序，总周期数52个，结构停顿周期数41个，占总执行周期数的78.85% 如图 2a。将浮点加法器改为4个，可以发现总周期数变为19个，结构停顿周期数变为8个，占总执行周期数的比例变为42.11%。如图 2b。

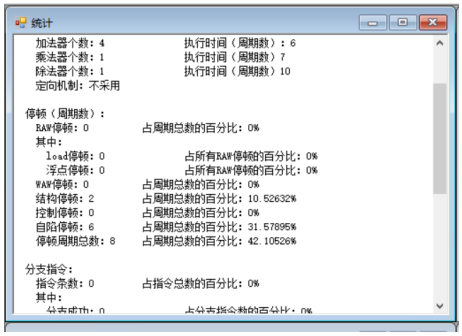
2.1.1 实验结论

由以上实验结果可知，发生冲突时，流水线会出现停顿从而降低CPU的性能，增加时钟周期数。结构停顿周期数的减少，可以提高CPU的性能。

为了解决这一问题，可以在流水线处理机中设置足够多的相互独立的指令寄存器和数据寄存器，或者改变程序的设计，尽量减少停顿的发生。



(a) 结构停顿周期数为41个



(b) 结构停顿周期数为8个

2.2 观察数据冲突并用定向技术来减少停顿

关闭定向技术，执行`data_hz.s`程序，执行结果如图 3。

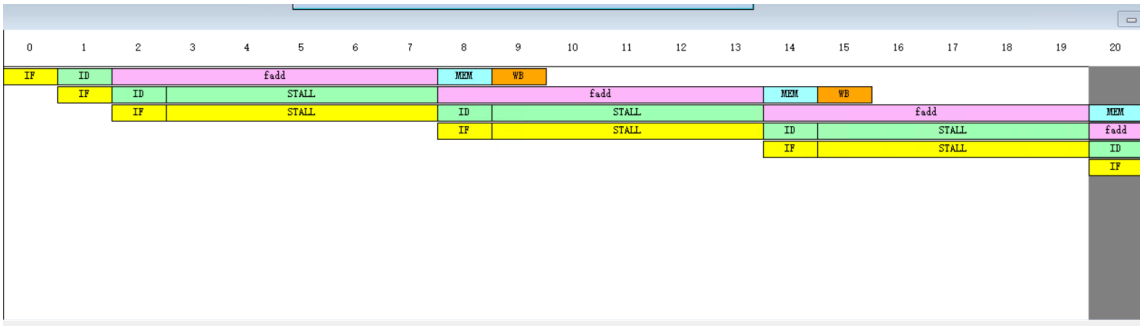


图 3: 关闭定向技术

可以观察到在第4，6，7，9，10，13，14，17，18，20，21，25，26，28，29，32，33，36，37，39，40，44，45，47，48，51，52，55，56，58，59 时钟周期发生RAW冲突。

启动定向技术，执行`data_hz.s`程序，执行结果如图 4。 在第 5,10,13,18,22,25,30,34,37 时钟周期发生

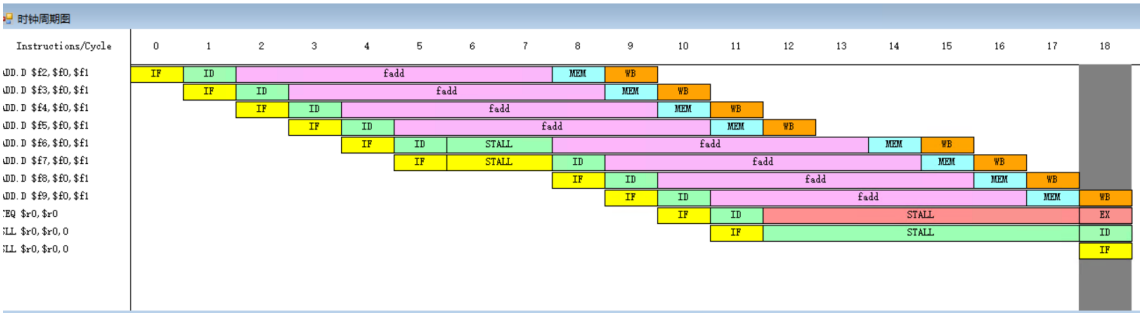
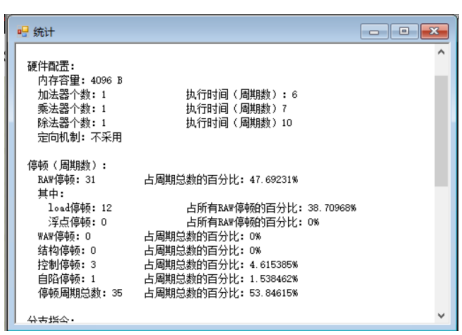


图 4: 启动定向技术

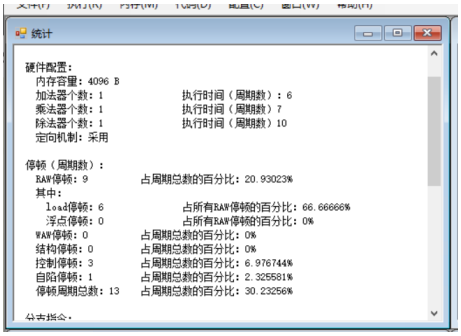
了RAW冲突，相比之下大大减少

对比是否启用定向技术，可以发现启用定向技术后，结构停顿周期数减少了很多，如图 ??。

可以发现数据冲突引起的停顿周期数为9，程序的总时钟周期数为43，停顿时钟周期数占总执行周期数的百分比为20.93023%，采用定向技术后的性能是原来的1.51倍。



(a) 关闭定向技术



(b) 启动定向技术

3 补充实验

3.1 实验题目

编写程序。在流水线 CPU 中，立方和公式的两种计算方法应该如何安排指令使得结构停顿占周期总数的百分比最小？两种方法的性能之比？立方和的两种公式如(1)

$$\begin{aligned} &a^3+b^3 \\ &(a+b)(a^2-b^2+ab) \end{aligned} \tag{1}$$

其中a、b的初始值存放在R1 R2寄存器中，计算结果存放在R3寄存器中。从R4开始使用寄存器，数量不限。

3.2 代码实现

为了最大限度地减少代码中的相关，我按照以下原则来实现代码：

- 前后两条指令读写的寄存器尽量不相同；
- 访存的指令尽量相邻；
- 尽量减少计算中的依赖关系；

然后根据结果再实际调整代码，使得停顿占周期总数的百分比最小。

3.3 a^3+b^3

计算a^3+b^3，首先从内存中加载操作数，然后计算a^2，然后计算b^2，最后分别计算a^3和b^3再求和，最后将a^3+b^3的结果存放在R3寄存器中。这样做总的停顿次数为5次。实现如代码 1

3.4 $(a+b)(a^2-b^2+ab)$

计算$(a+b)(a^2-b^2+ab)$，首先从内存中加载操作数，然后分别计算 a^2、b^2和ab，然后计算a^2-b^2+ab，最后计算$(a+b)(a^2-b^2+ab)$，最后将$(a+b)(a^2-b^2+ab)$的结果存放在R3寄存器中。这样做总的停顿次数为6次。实现如代码 2

3.5 运行结果和计算

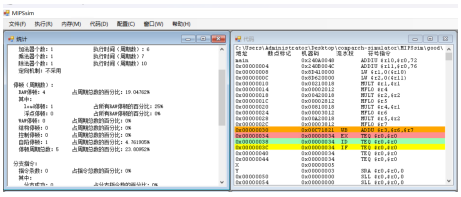
两段程序的运行结果如图 6。前者的停顿周期数为5，后者的停顿周期数为6。运行效率之比为：

$$\frac{CPI_1}{CPI_2} \approx 1.14 \tag{2}$$

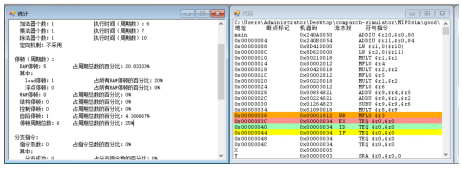
代码 1 计算a^3+b^3

```
.data
X: .word 5
Y: .word 3
.text
main:
ADDIU $r10, $r0, X
ADDIU $r11, $r0, Y
LW $r1, 0($r10)
LW $r2, 0($r11)
MULT $r1,$r1
MFLO $r4
MULT $r2,$r2
MFLO $r5
MULT $r4,$r1
MFLO $r6
MULT $r5,$r2
MFLO $r7
ADDU $r3,$r6,$r7
TEQ $r0,$r0
TEQ $r0,$r0
TEQ $r0,$r0
TEQ $r0,$r0
TEQ $r0,$r0
```

图 6: 计算a^3+b^3和$(a+b)(a^2-b^2+ab)$的运行结果



(a) 计算a^3+b^3



(b) 计算$(a+b)(a^2-b^2+ab)$

代码 2 计算$(a+b)(a^2-b^2+ab)$

```
.data
X: .word 5
Y: .word 3
.text
main:
ADDIU $r10, $r0, X
ADDIU $r11, $r0, Y
LW $r1, 0($r10)
LW $r2, 0($r11)
MULT $r1,$r1
MFLO $r4
MULT $r2,$r2
MFLO $r5
MULT $r1,$r2
MFLO $r6
ADDU $r9,$r4,$r5
ADDU $r8,$r1,$r2
SUBU $r9,$r9,$r6
MULT $r8,$r9
MFLO $r3
TEQ $r0,$r0
TEQ $r0,$r0
TEQ $r0,$r0
TEQ $r0,$r0
TEQ $r0,$r0
```

4 实验总结

在此次实验中，我加深了对计算机流水线基本概念的理解，理解了MIPS结构如何用5段流水线来实现，理解了各段的功能和基本操作，加深了对数据冲突、结构冲突的理解，理解了这两类冲突对CPU性能的影响。并且进一步理解了解决数据冲突的方法，掌握了如何应用定向技术来减少数据冲突引起的停顿。并且实际编写了计算立方和的程序来对比不同计算方法产生的误差。