《计算机系统结构》实验报告

实验名称		<u>实验 3 使用 MIPS 指令实现求两个数组的点积</u>
班	级	2015211307
学	号	2017526019
姓	名	刘禾子

实验3 使用MIPS指令实现两个数组的点积

1. 实验目的

- (1) 通过实验熟悉实验1和实验2的内容
- (2) 增强汇编语言编程能力
- (3) 学会使用模拟器中的定向功能进行优化
- (4) 了解对代码进行优化的方法

2. 实验平台

指令级和流水线操作级模拟器MIPSsim。

3. 实验内容和步骤

(1) 自行编写一个计算两个向量点积的汇编程序,该程序要求可以实现求两个向量点积计算后的结果。

两个向量元素使用数组进行数据存储,要求向量的维度不得小于10

- (2) 启动 MIPSsim。
- (3) 载入自己编写的程序,观察流水线输出结果。
- (4)使用定向功能再次执行代码,与刚才执行结果进行比较,观察执行效率的不同。
 - (5) 采用静态调度方法重排指令序列,减少相关,优化程序
- (6) 对优化后的程序使用定向功能执行,与刚才执行结果进行比较,观察执行效率的不同。

(1) 具体代码如下:

```
.text
main:
ADDIU $r1,$r0,array1 #r1存放矩阵1內容
ADDIU $r2,$r0,array2 #r2存放矩阵2內容
ADDIU $r3,$r0,10 #循环变量r3 10次
ADDIU $r7,$r0,0 #r7存放结果
loop:
LW $r4,0($r1) #取数
LW $r5,0($r2)
MUL $r6,$r4,$r5 #分别计算点积
ADDD $r7,$r7,$r6
ADDI $r1,$r1,4 #取下一个数
```

```
ADDI $r2, $r2, 4

ADDI $r3, $r3, -1 #循环变量-1

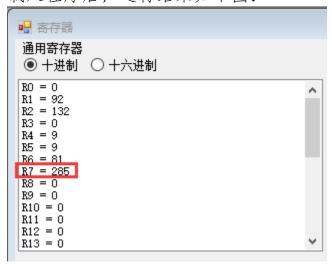
BGTZ $r3, loop

TEQ $r0, $r0
. data

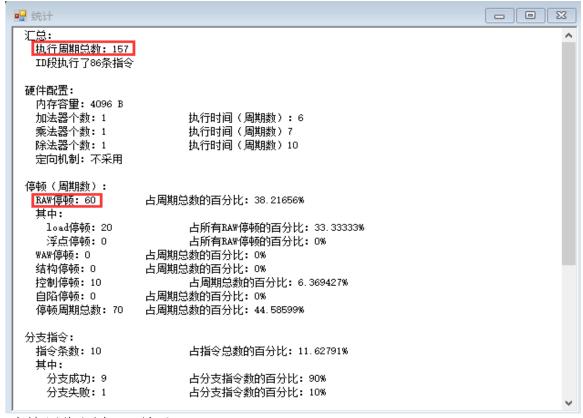
array1:. word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

array2:. word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

载入程序后,运行结果如下图:
```



汇总如下图所示:



时钟周期图如下所示:

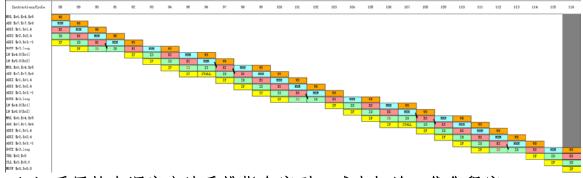


(2) 采用定向功能再次执行结果如下所示:



由上图可知,定向后的执行效率为定向前的(157/117) *100%=134.188034%,减少的周期数为RAW停顿周期数。

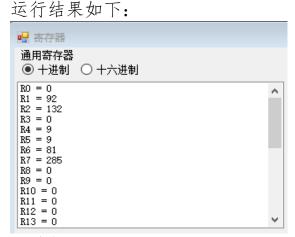
时钟周期图如下所示:



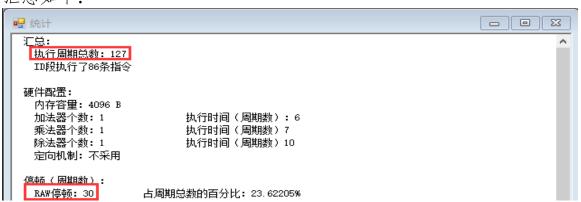
(3) 采用静态调度方法重排指令序列,减少相关,优化程序

分析源代码发现指令MUL \$r6,\$r4,\$r5与ADD \$r7,\$r7,\$r6指令存在数据相关先写后读即(RAW),分析其下面的指令均与其无关故将ADD \$r7,\$r7,\$r6指令下移至BGTZ \$r3,100p指令上方,从而消除其相关性。代码如下:

```
.text
main:
ADDIU $r1, $r0, array1 #r1存放矩阵1内容
ADDIU $r2, $r0, array2 #r2存放矩阵2内容
                     #循环变量r3 10次
ADDIU $r3, $r0, 10
ADDIU $r7, $r0, 0
                      #r7存放结果
loop:
                      #取数
LW $r4, 0($r1)
LW $r5, 0($r2)
MUL $r6, $r4, $r5
                      #分别计算点积
                      #取下一个数
ADDI $r1, $r1, 4
ADDI $r2, $r2, 4
ADDI $r3, $r3, -1
                      #循环变量-1
ADD $r7, $r7, $r6
                      #此处优化
BGTZ $r3, loop
TEQ $r0, $r0
.data
array1:.word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
array2:. word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```



汇总如下:



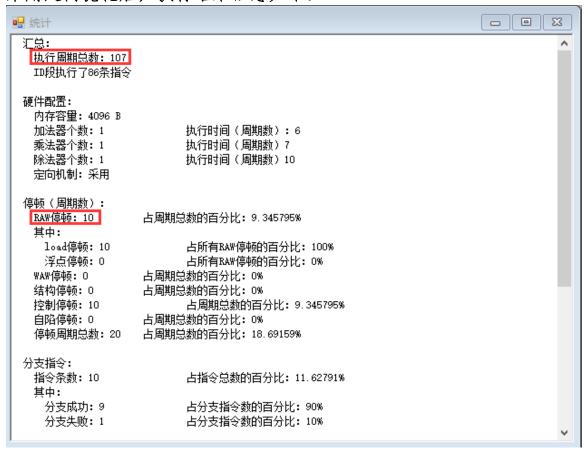
时钟周期如下:



由上可知,与未进行优化相比,执行效率为原来的 157/127=123.622%,减少的周期数为 RAW 停顿周期数。

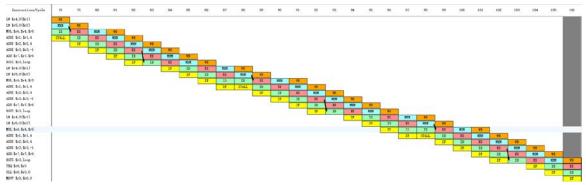
(4) 对优化后的程序使用定向功能执行,与刚才执行结果进行比较,观察执行效率的不同。

采用定向优化后, 执行结果汇总如下:



由上图可知,定向后的执行效率为定向前的 127/107=118.692%,减少的周期数为 RAW 停顿周期数。

时钟周期如下:



(5) 比较分析

采用重排并且定向功能的代码的性能较初始代码性能提高很大。初始代码需要 157 个时钟周期,优化后需要 107 个时钟周期,停顿周期从70 个时钟周期减少到了 10 个时钟周期,流水线冲突(数据相关和控制相关)都有所减少。

4. 实验心得

通过此次实验,我了解了代码优化的相关方法(定向、重排等)及其重要性,优化代码时需要理解上下文,基于其相关性进行优化,从而提高执行效率,减少多余的资源开销。