

用MIPS指令实现求两个数组的点积

计算机体系结构实验三



目录

[1． 实验目的 1](#_Toc516526111)

[2． 实验平台 1](#_Toc516526112)

[3． 实验原理 1](#_Toc516526113)

[4． 实验内容和步骤 1](#_Toc516526114)

[5． 实验总结 6](#_Toc516526115)

2017-6-8

[裴子祥 计科七班 学号2015211921]

[指导老师：黄智濒]

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

1. 实验平台

实验平台采用指令级和流水线操作级模拟器 MIPSsim。

1. 实验原理

本次实验通过MIPS语句实现对应的功能，通过代码在指令级和流水线操作级模拟器MIPSsim上执行，根据加载的文档代码中对应操作的通用寄存器，来观察执行的结果是否正确；然后通过观察统计一栏中汇总的执行周期总数，来判断优化和定向前后执行的效率高低，并判断静态调度的情况。

1. 实验内容和步骤

##注意事项

##不使用浮点指令及浮点寄存器！

##使用 TEQ $r0 $r0 结束程序！

（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。

向量的点积：假设有两个 n 维向量 a、b，则 a 与 b 的点积为：

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。

新建”DotProduct.s”汇编程序文件，代码如下：

.text

main**:**

ADDIU **$**r1,**$**r0,a #a是第一个数组

ADDIU **$**r2,**$**r0,b #b是第二个数组

ADDIU **$**r3,**$**r0,10 #r3存放循环变量，循环10次，向量维度是10

ADDIU **$**r4,**$**r0,0 #r4存放最终结果

lp**:**

LW **$**r5,0**($**r1**)**

LW **$**r6,0**($**r2**)**

MUL **$**r7,**$**r5,**$**r6

ADD **$**r4,**$**r4,**$**r7

ADDI **$**r1,**$**r1,4 #a数组的下一个元素

ADDI **$**r2,**$**r2,4 #b数组的下一个元素

ADDI **$**r3,**$**r3,**-**1 #循环变量减一

BGTZ **$**r3,lp

TEQ **$**r0,**$**r0 #结束程序

.data

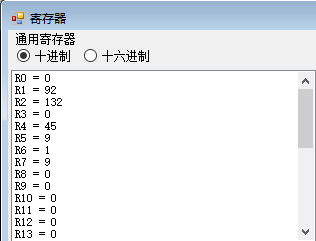
a**:** .word 0,1,2,3,4,5,6,7,8,9

b**:** .word 1,1,1,1,1,1,1,1,1,1

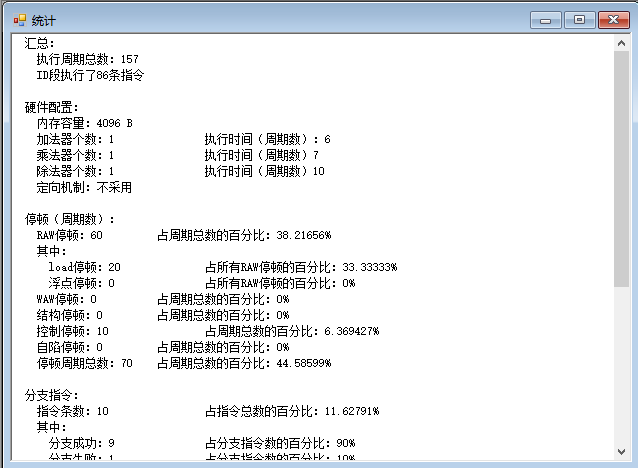
（2）启动 MIPSsim。

（3）载入自己编写的程序，观察流水线输出结果。

汇编程序运行结果如下：

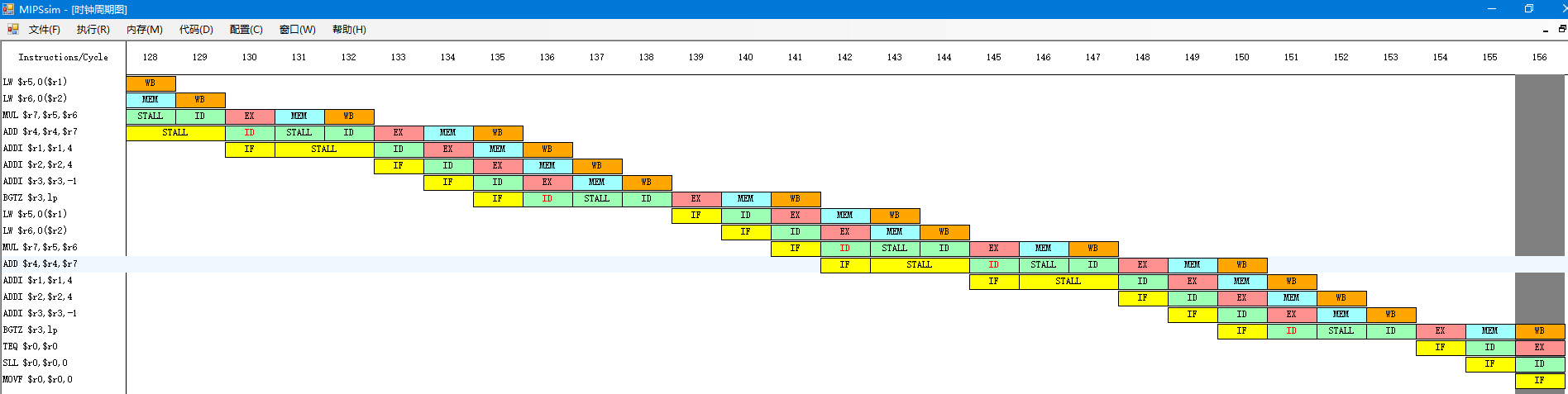


R4寄存器中保存正确结果45，知程序运行结果正确。



执行周期总数157；RAW停顿60；

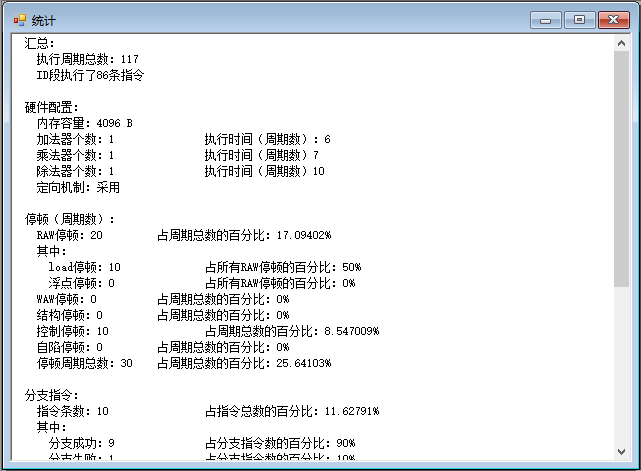
时钟周期图如下所示：



分析，可知，MUL $r7,$r5,$r6; ADD $r4,$r4,$r7; 指令之间产生较长STALL。还有ADDI $r3,$r3,-1; BGTZ $r3,lp; 指令之间产生STALL。，这是我们可以优化的地方。

（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。

开启定向功能：



执行周期总数117；RAW停顿20；比之前没开启定向，少了40个RAW停顿周期数。

定向后执行效率为定向前的 157/117 = 134.188%

开启定向，时钟周期图如下：



（5）采用静态调度方法重排指令序列，减少相关，优化程序

采用静态调度方法重排指令序列

根据之前分析，将ADD **$**r4,**$**r4,**$**r7 后移，使MUL与之STALL消减。

ADDI **$**r3,**$**r3,**-**1 前移，以更快执行BGTZ。

“BetterDotProduct.s”汇编程序文件，优化后代码：

.text

main**:**

ADDIU **$**r1,**$**r0,a

ADDIU **$**r2,**$**r0,b

ADDIU **$**r3,**$**r0,10

ADDIU **$**r4,**$**r0,0

lp**:**

LW **$**r5,0**($**r1**)**

LW **$**r6,0**($**r2**)**

MUL **$**r7,**$**r5,**$**r6

ADDI **$**r3,**$**r3,**-**1 #该步操作从后前移

ADDI **$**r1,**$**r1,4

ADDI **$**r2,**$**r2,4

ADD **$**r4,**$**r4,**$**r7 #该步操作从前后移

BGTZ **$**r3,lp

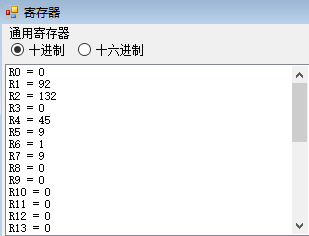
TEQ **$**r0,**$**r0

.data

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

b**:** .word 1,1,1,1,1,1,1,1,1,1

优化后程序运行结果：



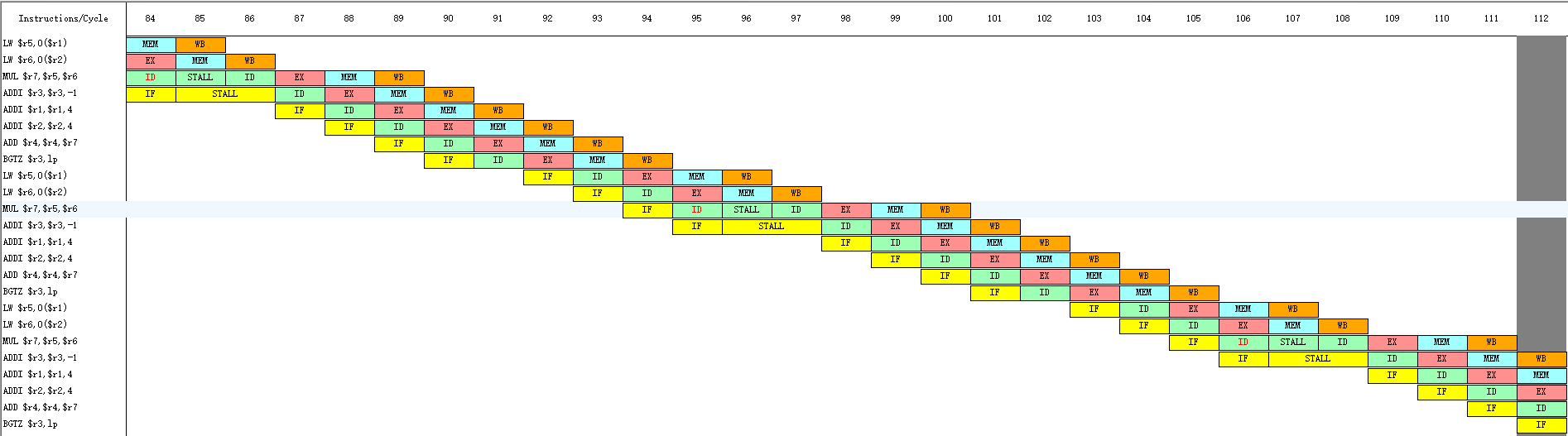
R4寄存器中保存正确结果45，知程序运行结果正确。



优化后的执行周期总数为117；RAW停顿20；

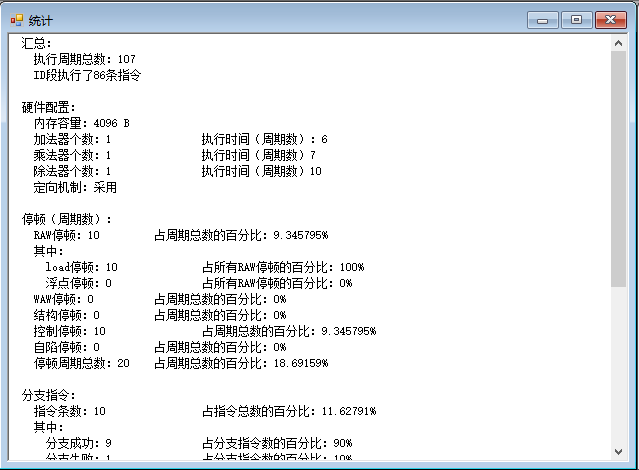
优化后执行效率为优化前的 157/117 = 134.188%

时钟周期图如下：



（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。

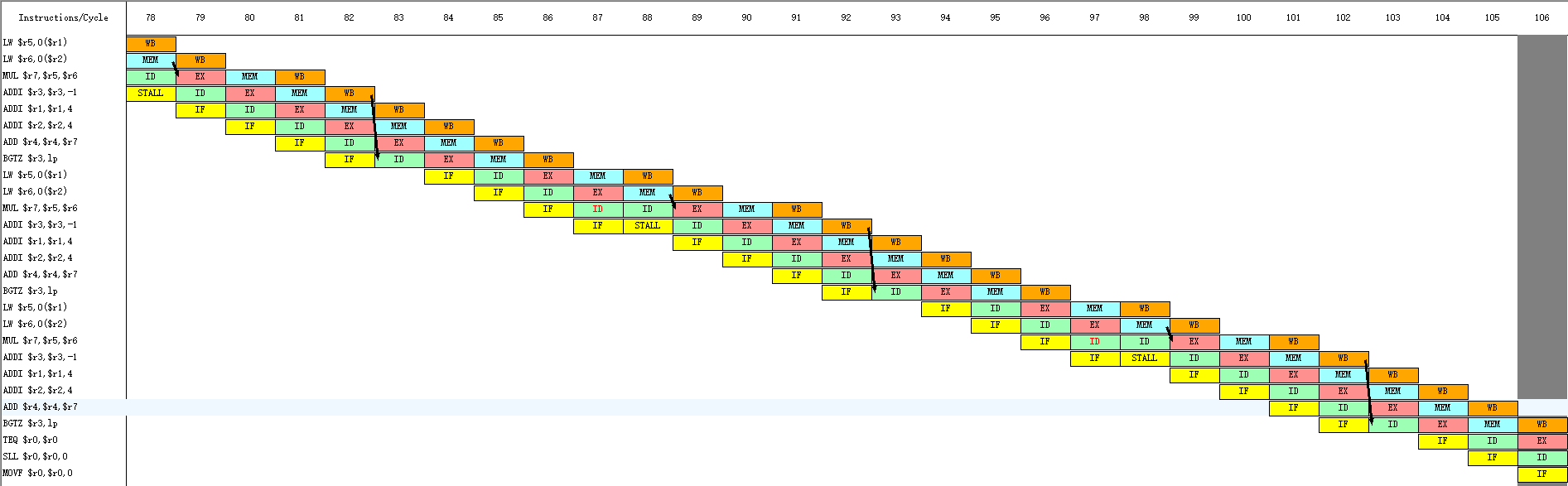
优化后代码，开启定向功能：



执行周期总数107；RAW停顿10；比之前没开启定向，少了10个RAW停顿周期数。

定向后执行效率为定向前的 117/107 = 109.3458%

优化后代码，开启定向，时钟周期图如下：



1. 实验总结

此次实验，通过MIPSsim上运行自身编写的数组点积汇编程序，成功地验证了，代码优化后的性能比代码优化前的性能有很大的提高。通过对原始代码结构，与时钟周期图，能够很快地找出，代码中可改进的代码处。主要在于两条汇编语句之间使用共同的寄存器时会导致冲突而形成STALL，这是影响执行性能的关键之处。采用静态调度方法重排指令序列以达到优化效果，保证其原有逻辑不变。还有定向功能，对程序运行效率也能起到极为关键的作用，定向技术将某条指令产生的计算结果从其产生的地方直接送到其它指令需要的地方，这样避免了停顿。

通过此次实验，收获很大，主要在于定向技术与代码优化，加深了对指令流水线的理解，加强了动手实践能力。