

北京邮电大学

实验报告



题目： 实验 3 使用 MIPS 指令实现求两个数组的点积

计算机系统结构实验小组成员信息			
班级	姓名	学号	学院
2020211314	王小龙	2020211502	计算机学院
2020211314	闻奔放	2020211505	计算机学院
2020211314	黄洪健	2020211371	计算机学院

注：红色标出的成员为本次实验的完成者

2023 年 4 月 12 日

一、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

二、实验原理

在本次实验中，通过编写汇编程序来完成计算两个向量点积的功能，并且通过在 MIPSsim 实验平台上进行执行来观察程序的运行状况，最后通过对比 MIPSsim 实验平台的统计信息来比较优化前后程序的效率，从而了解静态调度优化的知识。

三、向量点积程序代码清单及注释说明



array.s - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

.text

main:

ADDIU \$r1,\$r0,array1 # 第一个数组的段地址

ADDIU \$r2,\$r0,array2 # 第二个数组的段地址

ADDIU \$r3,\$r0,10 # 向量长度设置为10

ADDIU \$r4,\$r0,0 # 用r4保存点积结果

loop:

LW \$r5,0(\$r1)

LW \$r6,0(\$r2)

MUL \$r7,\$r5,\$r6 # 乘法运算

ADD \$r4,\$r4,\$r7 # 用r4 存放点积结果

ADDI \$r1,\$r1,4 # 获取下一个数据

ADDI \$r2,\$r2,4 # 获取下一个数据

ADDI \$r3,\$r3,-1 # 实现递减

BGTZ \$r3,loop # 进行循环判断

TEQ \$r0,\$r0

.data

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

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

四、优化后的程序代码清单

```
*array_optimum.s - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
.text
main:
ADDIU $r1,$r0,array1 # 第一个数组的段地址
ADDIU $r2,$r0,array2 # 第二个数组的段地址
ADDIU $r3,$r0,10 # 向量长度设置为10
ADDIU $r4,$r0,0 # 用r4保存点积结果
loop:
LW $r5,0($r1)
LW $r6,0($r2)
MUL $r7,$r5,$r6 # 乘法运算
ADDI $r1,$r1,4 # 获取下一个数据
ADDI $r2,$r2,4 # 获取下一个数据
ADDI $r3,$r3,-1 # 实现递减
ADD $r4,$r4,$r7 # 用r4 存放点积结果——优化位置
BGTZ $r3,loop # 进行循环判断
TEQ $r0,$r0
.data
array1: .word 1,2,3,4,5,6,7,8,9
array2: .word 1,2,3,4,5,6,7,8,9
```

五、实验步骤

(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。

向量的点积：假设有两个 n 维向量 a 、 b ，则 a 与 b 的点积为：

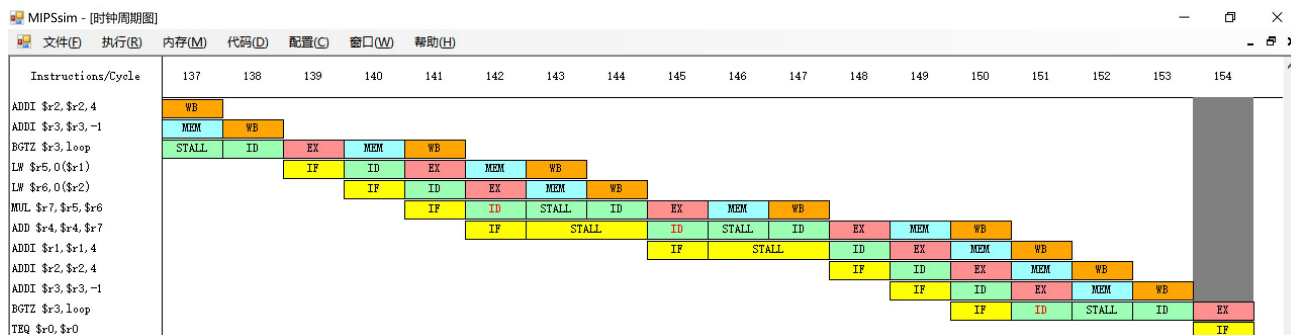
$$a \cdot b = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

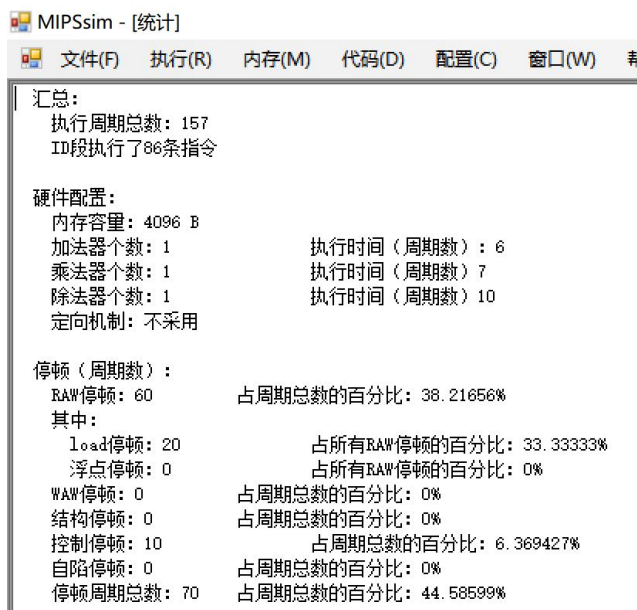
两个向量元素使用数组进行数据存储，要求向量的维度不得小于 10。

编写代码如上。

(2) 启动 MIPSsim。

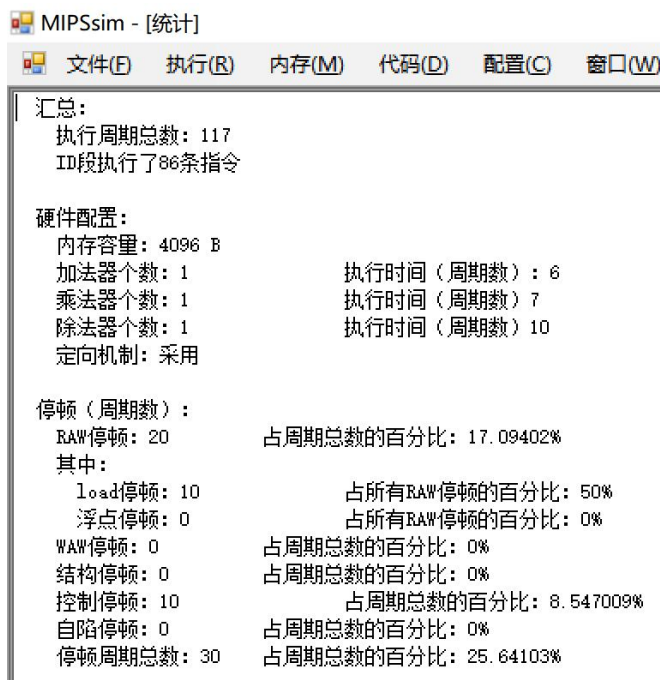
(3) 载入自己编写的程序，观察流水线输出结果如下：





从上图可以看出，程序执行周期总数为 157 个，RAW 停顿有 60 个周期，占比为 38.21656%

(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。



从上图可以看到，程序执行周期总数为 117 个，RAW 停顿有 20 个周期，占比为 17.09402%
 执行效率变为 $157/117=1.342$ 倍

(5) 采用静态调度方法重排指令序列，减少相关，优化程序：

由于红框标出的指令存在读后写相关，故根据静态调度方法，将这几条指令用 ADDI 指令隔开

```
array_optimum.s - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
.text
main:
ADDIU $r1,$r0,array1 # 第一个数组的段地址
ADDIU $r2,$r0,array2 # 第二个数组的段地址
ADDIU $r3,$r0,10 # 向量长度设置为10
ADDIU $r4,$r0,0 # 用r4保存点积结果
loop:
LW $r5,0($r1)
LW $r6,0($r2)
ADDI $r1,$r1,4 # 获取下一个数据
MUL $r7,$r5,$r6 # 乘法运算——优化位置
ADDI $r2,$r2,4 # 获取下一个数据
ADDI $r3,$r3,-1 # 实现递减
ADD $r4,$r4,$r7 # 用r4 存放点积结果——优化位置
BGTZ $r3,loop # 进行循环判断
TEQ $r0,$r0
.data
array1: .word 1,2,3,4,5,6,7,8,9
array2: .word 1,2,3,4,5,6,7,8,9
```

(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。

MIPSSim - [统计]

文件(F) 执行(R) 内存(M) 代码(D) 配置(C) 窗口(W) 帮助(H)

汇总:

执行周期总数: 97
ID段执行了86条指令

硬件配置:

内存容量: 4096 B
加法器个数: 1
乘法器个数: 1
除法器个数: 1
定向机制: 采用

执行时间 (周期数): 6
执行时间 (周期数) 7
执行时间 (周期数) 10

停顿 (周期数):

RAW停顿: 0 占周期总数的百分比: 0%

其中:

load停顿: 0 占所有RAW停顿的百分比: 0%
浮点停顿: 0 占所有RAW停顿的百分比: 0%
WAW停顿: 0 占周期总数的百分比: 0%
结构停顿: 0 占周期总数的百分比: 0%
控制停顿: 10 占周期总数的百分比: 10.30928%
自陷停顿: 0 占周期总数的百分比: 0%
停顿周期总数: 10 占周期总数的百分比: 10.30928%

从上图可以看到，优化后的程序使用定向功能执行后，执行周期总数为 97，RAW 停顿为 0，效率变为原来的 $117/97=1.2062$ 倍

五、总结体会

通过本次实验，增强了自己的汇编语言编程能力，使自己进一步了解了代码进行优化的方法，在实验过程中认识到了，代码优化和定向技术对代码效率提高的重要性，对流水线有了更深的理解，总之，收获很多。