

同济大学计算机系

计算机组成原理实验报告



学 号 2152809

姓 名 曾崇然

专 业 计算机科学与技术

授课老师 张冬冬老师

一、实验内容

通过编写几个汇编小程序来熟悉常用的 MIPS 汇编指令，并学习 MARS 模拟器的使用。

小程序具体包括：

- ① Fibonacci 数列：将\$2,\$3 寄存器初始化为 fibonacci 数列的前两个数 0，1；\$4 为数列中所需得到的数字的序号（\$4=4 即表示得到第四个 fibonacci 数）；最后得到的结果存入\$1
- ② 将一串数列输入\$2-\$6，用冒泡排序算法对其进行排序
- ③ 运用布斯乘法算法实现两个数的乘法，结果用两个寄存器表示

二、模块建模

汇编指令代码：

Fibonacci 数列：

```
.text
main:
    li $2,0#Fibonacci 数列的第 1 项
    li $3,1#Fibonacci 数列的第 2 项
    li $4,2#所要计算的 Fibonacci 数列的项
    li $5,2#计数器
    beq $4,1,patch
judge:
    beq $5,$4,end#判断是否到指定项
    bne $5,$4,caculate
caculate:
    add $6,$2,$3
    move $2,$3
    move $3,$6#计算下一项
    addi $5,$5,1#计数器加 1
    j judge
patch:
    li $3,0
end:
    move $1,$3
    li $v0,10
    syscall
```

冒泡排序：

```
.text
main:
    li $2,10
    li $3,11
    li $4,2
```

```

        li $5,6
        li $6,2#5 个待排序的数
        li $7,0#计数器
        li $8,0#存放比较结果
        li $9,0#交换的中间寄存器
loop:
    beq $7,6,end
    addi $7,$7,1
comp_23:
    slt $8,$2,$3
    beq $8,1,comp_34
    move $9,$2
    move $2,$3
    move $3,$9
comp_34:
    slt $8,$3,$4
    beq $8,1,comp_45
    move $9,$3
    move $3,$4
    move $4,$9
comp_45:
    slt $8,$4,$5
    beq $8,1,comp_56
    move $9,$4
    move $4,$5
    move $5,$9
comp_56:
    slt $8,$2,$3
    beq $8,1,loop
    move $9,$5
    move $5,$6
    move $6,$9
    j loop
end:
    li $v0,10
    syscall

```

Booth 乘法:

.text

```

main:
    li $2,5#被乘数
    li $3,5#乘数
    li $4,0#用$4 来表示 R0(前 x 位)
    move $5,$3#用$5 来表示 R1(中间 y 位)

```

```

    li $6,0#辅助位 Z
    li $7,0#计数器
    li $8,0#移位辅助 1
    li $9,0#移位辅助 2
loop:
    beq $7,32,end
    addi $7,$7,1#循环 32 次
    andi $8,$5,0x00000001
    andi $9,$6,0x80000000
    beq $8,0x00000000,choice0
    beq $8,0x00000001,choice1
choice0:
    beq $9,0x00000000,case3
    beq $9,0x80000000,case1
choice1:
    beq $9,0x00000000,case2
    beq $9,0x80000000,case3#检查 R1 的最后 1 位和 Z，进行跳转
case1:
    addu $4,$4,$2
    j move_bit#01 情况
case2:
    subu $4,$4,$2
    j move_bit#10 情况
case3:
    j move_bit#00 或者 11 情况
move_bit:
    srl $6,$6,1
    andi $8,$5,0x00000001
    beq $8,0x00000001,change_Z
    bne $8,0x00000001,set_R1
change_Z:
    addi $6,$6,0x80000000#移位 Z，同时将 R1 最低位赋给 Z
set_R1:
    srl $5,$5,1
    andi $8,$4,0x00000001
    beq $8,0x00000001,change_R1
    bne $8,0x00000001,set_R0
change_R1:
    addi $5,$5,0x80000000#移位 R1,同时将 R0 最低位赋给 R1 最高位
set_R0:
    sra $4,$4,1#算术移位 R0
    j loop
end:
    li $v0,10

```

syscall

三、实验结果

Fibonacci 数列：

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000015
\$v0	2	0x0000000a
\$v1	3	0x00000015
\$a0	4	0x00000009
\$a1	5	0x00000009
\$a2	6	0x00000015
\$a3	7	0x00000000

我输入的计算第九项，可以看到结果被正确计算出后放进了 1 寄存器中
冒泡排序：

\$v0	2	0x0000000a
\$v1	3	0x00000002
\$a0	4	0x00000006
\$a1	5	0x0000000a
\$a2	6	0x0000000b
\$a3	7	0x00000006
\$t0	8	0x00000000
\$t1	9	0x0000000b
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000

我输入的 5 个待排序数是 10, 11, 2, 6, 2. 可以看到在对应的寄存器中已经按照顺序排序
(2 号寄存器因为程序结束的需要所以在排好序之后放入了其它值)

Booth 乘法：

\$zero	0	0x00000000
\$at	1	0x00000020
\$v0	2	0x0000000a
\$v1	3	0x00000005
\$a0	4	0x00000000
\$a1	5	0x00000019
\$a2	6	0x00000005
\$a3	7	0x00000020
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000

我输入的乘数和被乘数位 5 和 5，可以看到计算结果被计算出来存储在了 4 号和 5 号寄存器中