

同济大学计算机系

计算机组成原理实验报告



学 号 2351753

姓 名 黄保翔

专 业 计算机科学与技术

授课老师 张冬冬

一、实验内容

通过编写汇编程序熟悉 54 条 cpu 指令。

二、模块建模

1、斐波那契数列

使用汇编求解斐波那契数列，利用寄存器，通过滚动数组的方式实现求解
代码:

```
.text
#初始化$2$3$4
xor $2,$2,$2
addiu $2,$2,0
xor $3,$3,$3
addiu $3,$3,1
xor $4,$4,$4
addiu $4,$4,10

loop:
beq $4,$0,main
addiu $4,$4,-1
addu $6,$2,$3
addiu $2,$3,0
addiu $3,$6,0
j loop

main:
#将结果存入$1
xor $1,$1,$1
addiu $1,$2,0
```

2、冒泡排序

使用汇编进行冒泡排序，设置相邻的寄存器，循环次数设为所需排序的元素个数，每次循环比较相邻的两个数字，根据条件进行交换

```
#使用 汇编进行冒泡排序
#进行数组初始化
xor $1,$1,$1#作为 0
xor $2,$2,$2
xor $3,$3,$3
xor $4,$4,$4
xor $5,$5,$5
xor $6,$6,$6
xor $7,$7,$7
```

```
xor $8,$8,$8#作为大小标记
xor $9,$9,$9#作为临时变量
xor $10,$10,$10#计数器
#进行数组赋值
addiu $2,$2,5
addiu $3,$3,6
addiu $4,$4,4
addiu $5,$5,2
addiu $6,$6,3
addiu $7,$7,1
addiu $10,$10,6

loop:
beq $10,$1,end
addiu $10,$10,-1
#进行冒泡排序

comp23:
slt $8,$2,$3
beq $8,$1,comp34#小于的话直接跳过比较下一个
#交换
xor $9,$9,$9
addu $9,$9,$2
xor $2,$2,$2
addu $2,$2,$3
xor $3,$3,$3
addu $3,$3,$9
#继续比较

comp34:
slt $8,$3,$4
beq $8,$1,comp45
#交换
xor $9,$9,$9
addu $9,$9,$3
xor $3,$3,$3
addu $3,$3,$4
xor $4,$4,$4
addu $4,$4,$9
#继续比较

comp45:
slt $8,$4,$5
beq $8,$1,comp56
```

```

#交换
xor $9,$9,$9
addu $9,$9,$4
xor $4,$4,$4
addu $4,$4,$5
xor $5,$5,$5
addu $5,$5,$9
#继续比较

comp56:
slt $8,$5,$6
beq $8,$1,comp67
#交换
xor $9,$9,$9
addu $9,$9,$5
xor $5,$5,$5
addu $5,$5,$6
xor $6,$6,$6
addu $6,$6,$9
#继续比较

comp67:
slt $8,$6,$7
beq $8,$1,loop
#交换
xor $9,$9,$9
addu $9,$9,$6
xor $6,$6,$6
addu $6,$6,$7
xor $7,$7,$7
addu $7,$7,$9

j loop
end:

```

2、booth 乘法

使用汇编手动模拟 booth 乘法，每次取结尾两位，根据条件进行相应位置的跳转，结尾进行右移操作，由于需要跨越寄存器进行右移，因此首先判断临时积的结尾是否是 1，若为 1，则手动对低位积的最高位赋 1，否则跳过这一步骤，直接进行右移

```

.text
#初始化寄存器
xor $1,$1,$1#0 计数器
xor $2,$2,$2#乘数
xor $3,$3,$3#被乘数

```

```

xor $4,$4,$4#结果低 32 位
xor $5,$5,$5#结果高 32 位
xor $6,$6,$6#循环计数器
xor $7,$7,$7#存结尾 2 位
xor $8,$8,$8#辅助移位器
xor $10,$10,$10#临时积
xor $11,$11,$11#存 00
xor $12,$12,$12#存 01
xor $13,$13,$13#存 10
xor $14,$14,$14#存 11
xor $15,$15,$15#存乘数
main:
    #初始化
    addiu $2,$2,3
    addiu $3,$3,2#计算 2*3=6
    addiu $6,$6,32#循环 32 次
    addiu $11,$11,0x00000000#存 00
    addiu $12,$12,0x00000001#存 01
    addiu $13,$13,0x00000002#存 10
    addiu $14,$14,0x00000003#存 11
    addu $15,$2,$15#存乘数
    sll $8,$2,1#初始结尾补 0
loop:
    beq $6,$0,end#循环结束
    addi $6,$6,-1#计数器减 1
    andi $7,$8,0x00000003#取最后两位
    beq $7,$11,case00#00
    beq $7,$12,case01#01
    beq $7,$13,case10#10
    beq $7,$14,case11#11

case00:
    j shift
case01:
    #加 x
    addu $10,$10,$3
    j shift

case11:
    j shift
case10:
    #减 x
    subu $10,$10,$3

```

```

        j shift
shift:
    andi $8,$10,0x00000001#取最低位
    srl $4,$4,1#右移位 1 位
    bne $8,0x00000001,set0#跳过，最高位设置为 0
    addiu $4,$4,0x80000000#设置最高位为 1
set0:
    sra $10,$10,1#右移位 1 位

    xor $8,$8,$8#清空辅助移位器
    addu $8,$8,$15
    sra $15,$15,1#右移位 1 位
    j loop#进入下一轮循环
end:
    addiu $5,$5,$10#高 32 位存结果

```

三、实验结果

1、斐波那契数列

Registers	Coproc 1	Coproc 0	
Name	Number	Value	
\$zero	0	0x00000000	
\$at	1	0x00000037	
\$v0	2	0x00000037	
\$v1	3	0x00000059	
\$a0	4	0x00000000	
\$a1	5	0x00000000	
\$a2	6	0x00000059	
\$a3	7	0x00000000	
\$t0	8	0x00000000	

测试斐波那契数列第 10 项，结果存入\$1，为 0X37 等于十进制 55

2、冒泡排序

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000005
\$v1	3	0x00000006
\$a0	4	0x00000004
\$a1	5	0x00000002
\$a2	6	0x00000003
\$a3	7	0x00000001

排序前

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000006
\$v1	3	0x00000005
\$a0	4	0x00000004
\$a1	5	0x00000003
\$a2	6	0x00000002
\$a3	7	0x00000001
\$t0	8	0x00000000

排序后

3、Booth 乘法

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000001
\$v0	2	0x00000002
\$v1	3	0x00000003
\$a0	4	0x00000006
\$a1	5	0x00000000
\$a2	6	0x00000000

测试 2*3，低位存入\$4,高位存入\$5