北京航空航天大學
BEIHANG UNIVERSITY

计算机组成原理第四次作业

第五部分 汇编语言

北京航空航天大学

计算机学院

陈麒先

16061160

二〇一七年十一月

郑重声明

关于诚实守信公约：

以下题目标号前注明*号表示本题参考了互联网资料，题目标号为红色表示本题是与同学研究后的结论，其余未注明题目均为翻阅课件、课堂笔记和教材后独立思考的结果。特此声明。

16061160

陈麒先

原创性声明

作业中出现的公式、图片、代码段和图片的文字注释信息，均为作者原创。抄袭行为在任何情况下都被严格禁止(COPY is strictly prohibited under any circumstances)！转载或引用须征得作者本人同意，并注明出处！

16061160

陈麒先

目录

1. (a) 8'b0010_0011

(b) 0x15

(c) signed 8'b1101_1101

(d) 132

(e) Yes.

(f) 0xff88

(g) 00012350 00012344

将以寄存器$a0 中的值为地址的内存中的数据，载入寄存器$t0 中

$a0 ← $a0 + 4

$t1 ← $t0 + 1

若 $t1 的值为 0，则跳转到 loop 标签下的那条指令

2. 00082021

34020004

0000000c

2108ffff

1500fffb

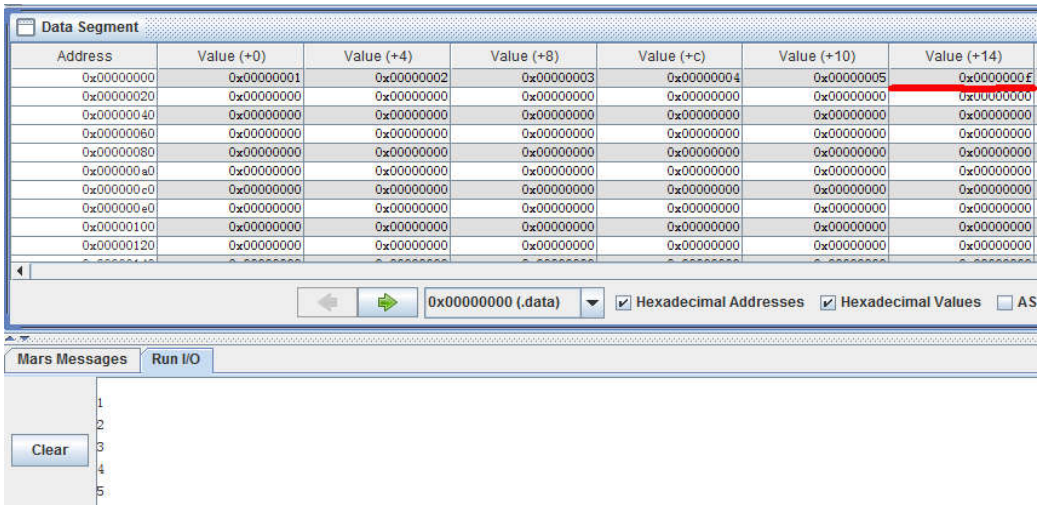
32f0ffc0

01f02025

a2c40004

0017b902

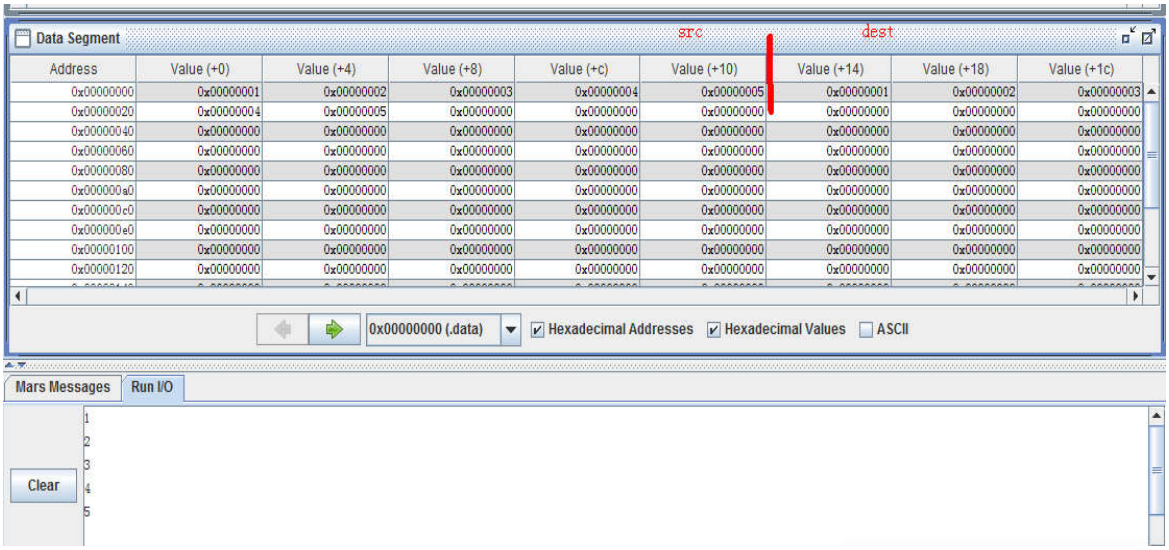
3．代码运行结果（为方便表述，将内存规模缩小至 5 字）



代码片段：

```
.data
        example100: .space 440
.text
        move $t0,$zero                          # $t0 = i
        for_1_begin:                            # store 100 numbers to memory
                beq $t0,5,for_1_end

                li $v0,5
                syscall
                la $t1,example100               # calculate the address
                li $t2,4
                mult $t2,$t0
                mflo $t2
                addu $t2,$t2,$t1
                sw $v0,0($t2)

                addi $t0,$t0,1
                j for_1_begin
        for_1_end:

        li $t0,0                                # $t0 = i
        li $t5,0                                # $t5 = sum
        for_2_begin:
                beq $t0,5,for_2_end

                li $t1,4                        # calculate address
                la $t2,example100
                mult $t1,$t0
                mflo $t1
                addu $t2,$t2,$t1
                lw $t3,0($t2)
                addu $t5,$t5,$t3                # add

                addi $t0,$t0,1
                j for_2_begin
        for_2_end:

        addi $t2,$t2,4
        sw $t5,0($t2)                           # store num to memory next

        move $a0,$t5
        li $v0,1
        syscall
```

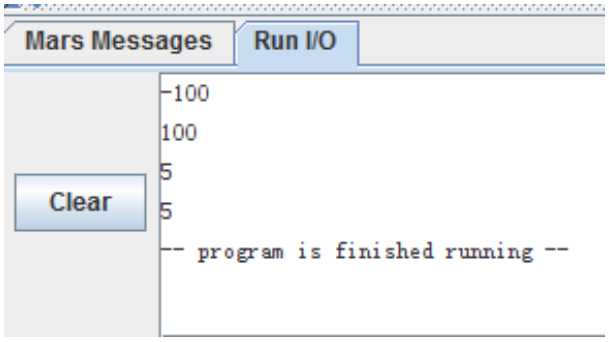
4. 代码运行结果（为方便表述，将内存规模缩小至 5 字）



代码片段：

```
.data
        src: .space 400
        dest: .space 400
.text
        li $t0,0                        # $t0 = i
        for_1_begin:
                beq $t0,100,for_1_end   # initialize memory of src

                li $v0,5
                syscall
                li $t1,4
                la $t2,src
                mult $t0,$t1
                mflo $t1
                addu $t2,$t2,$t1
                sw $v0,0($t2)

                addi $t0,$t0,1
                j for_1_begin
        for_1_end:

    li $t0,0
    for_2_begin:
            beq $t0,100,for_2_end

            li $t1,4                        #calculate address
            la $t2,src                      #transfer contents to new field of memory
            la $t3,dest
            mult $t0,$t1
            mflo $t1
            addu $t2,$t2,$t1
            addu $t3,$t3,$t1
            lw $t4,0($t2)
            sw $t4,0($t3)

            addi $t0,$t0,1
            j for_2_begin
    for_2_end:
```

5. 代码运行结果：

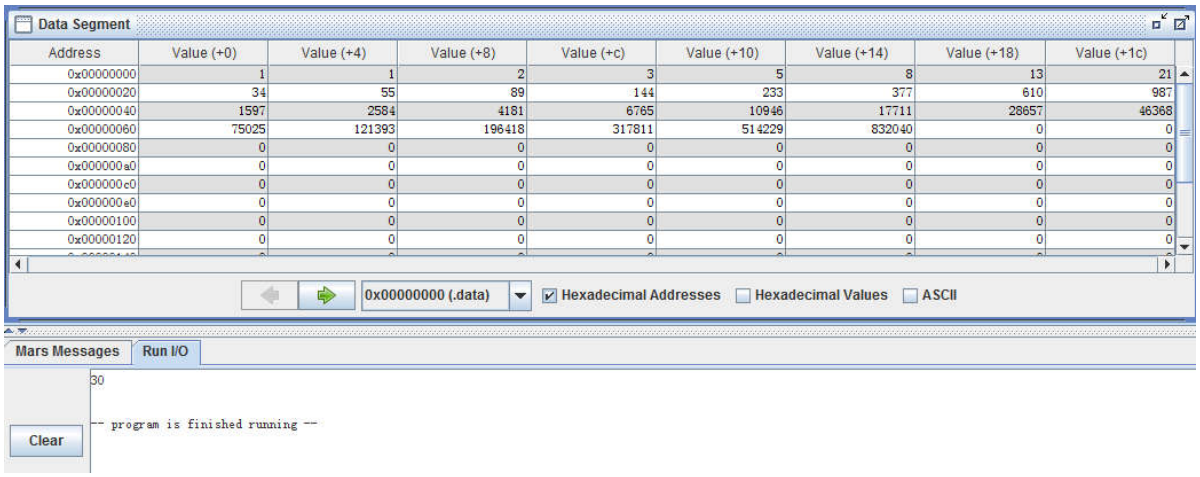


代码片段：

```
4  main:
5          li $v0,5
6          syscall
7          move $a0,$v0
8
9          jal ABS
10         move $a0,$v0
11         li $v0,1
12         syscall
13
14         la $a0,space              #completely unnecessarily
15         li $v0,4                  #for beauty only
16         syscall
17
18         li $v0,5                  # call ABS a second time
19         syscall
20         move $a0,$v0
21
22         jal ABS
23         move $a0,$v0
24         li $v0,1
25         syscall
26
27         li $v0,10                 #to end the programme successfully
28         syscall
```

```
29
30 ABS:
31         move $t0,$a0
32
33         blt $t0,$zero,if_1_else   # if $t0 > 0
34                 move $v0,$t0      # then
35                 jr $ra
36         if_1_else:                #else
37                 sub $v0,$zero,$t0
38                 jr $ra
```

6. 代码运行结果：（由于题目表述不完备，此处数组缺省为 4000 Byte，测试 N=30）



Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x00000000	1	1	2	3	5	8	13	21
0x00000020	34	55	89	144	233	377	610	987
0x00000040	1597	2584	4181	6765	10946	17711	28657	46368
0x00000060	75025	121393	196418	317811	514229	832040	0	0
0x00000080	0	0	0	0	0	0	0	0
0x000000a0	0	0	0	0	0	0	0	0
0x000000c0	0	0	0	0	0	0	0	0
0x000000e0	0	0	0	0	0	0	0	0
0x00000100	0	0	0	0	0	0	0	0
0x00000120	0	0	0	0	0	0	0	0

0x00000000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

30

-- program is finished running --

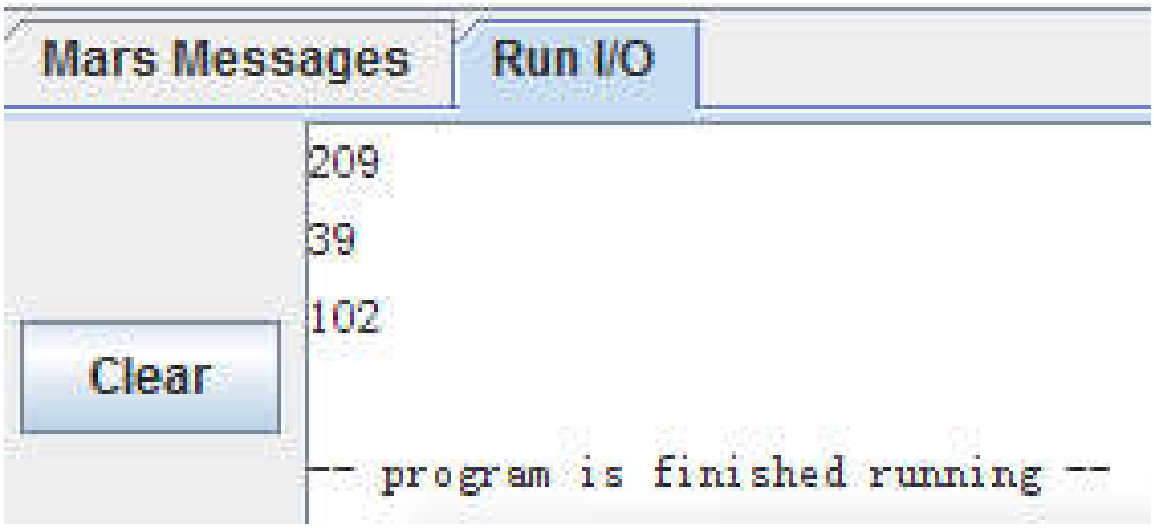
Clear

代码片段：

```
1   .data
2           array: .space 4000
3   .text
4   main:
5           li $v0,5            # read $a0 = N
6           syscall
7           move $a0,$v0
8           la $a1,array        # get address
9
10          jal FIB
11
12          li $v0,10           # end programme
13          syscall
14
```

```
15  FIB:
16          move $t0,$a0          # $t0 = N
17          move $t1,$a1          # $t1 = array[0]
18          li $t2,2              # $t2 = i
19          li $t3,4
20          li $t4,1              # $t4 = current numS
21          sw $t4,0($t1)
22          addu $t1,$t1,$t3
23          sw $t4,0($t1)
24          for_1_begin:
25                  beq $t2,$t0,for_1_end
26
27                  addu $t1,$t1,$t3
28                  lw $t5,-8($t1)
29                  lw $t6,-4($t1)
30                  add $t4,$t5,$t6
31                  sw $t4,0($t1)
32
33                  addi $t2,$t2,1
34                  j for_1_begin
35          for_1_end:
36          jr $ra
```

7. 代码运行结果：



$v0	2	10
$v1	3	0
$a0	4	39
$a1	5	102
$a2	6	209
$a3	7	0

代码片段：

```
main:
        li $v0,5
        syscall
        move $a0,$v0
        li $v0,5
        syscall
        move $a1,$v0
        li $v0,5
        syscall
        move $a2,$v0            # input $a0,$a1,$a2

        jal sort

        li $v0,10
        syscall

```

```
sort:
        move $t0,$a0
        move $t1,$a1
        move $t2,$a2
        if_1_begin:
                bgt $t0,$t1,if_1_else        # if $a0 > $a1
                j if_2_begin
        if_1_else:
                move $t4,$t0                 # swap $a0,$a1
                move $t0,$t1
                move $t1,$t4
        if_2_begin:
                bgt $t0,$t2,if_2_else        # if $a0 > $a2
                j if_3_begin
        if_2_else:
                move $t4,$t0                 # swap $a0,$a1
                move $t0,$t2
                move $t2,$t4
        if_3_begin:
                bgt $t1,$t2,if_3_else        # if $a1 > $a2
                j end
        if_3_else:
                move $t4,$t1                 # swap $a0,$a1
                move $t1,$t2
                move $t2,$t4
        end:
                move $a0,$t0
                move $a1,$t1
                move $a2,$t2      # save contents back to $a0~$a2
                jr $ra                       # return
```

***后记：所有编程题源代码已打包至文件夹，请老师注意查收，谢谢！**