Computer Architecture Lab Report Week 3

Full name: Nguyễn Đức Đại Dương

Student ID: 20225616

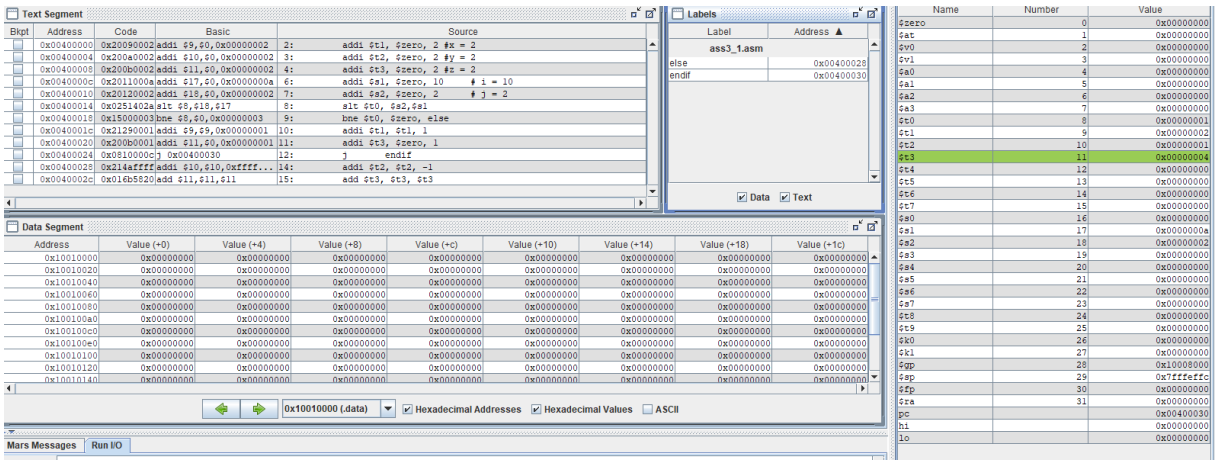
Assignment 1:

Xét x,y,z đều bằng 2

Trường hợp 1: i = 10, j = 2 (i>j)

```
.text
        addi $t1, $zero, 2 #x = 2
        addi $t2, $zero, 2 #y = 2
        addi $t3, $zero, 2 #z = 2

        addi $s1, $zero, 10     # i = 10
        addi $s2, $zero, 2      # j = 2
        slt $t0, $s2,$s1
        bne $t0, $zero, else
        addi $t1, $t1, 1
        addi $t3, $zero, 1
        j       endif
else:
        addi $t2, $t2, -1
        add $t3, $t3, $t3
endif:
```

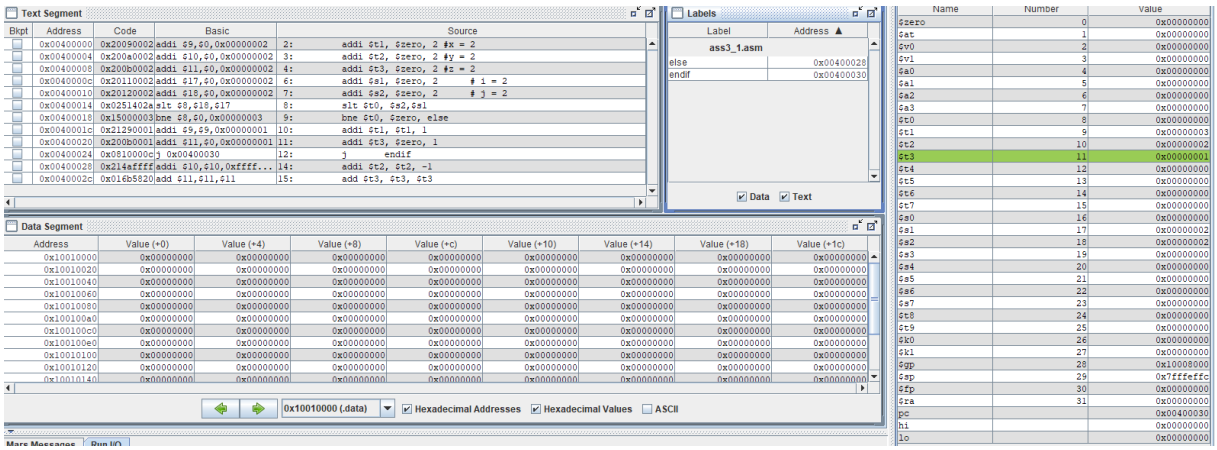


Nhận xét: sau lệnh slt, thanh ghi t0 thay đổi giá trị là 1, thanh ghi chứa x,y,z thay đổi, thanh ghi pc thay đổi

=>Kết quả thực thi đúng với lí thuyết

Trường hợp 2: i = 2, j = 2 (i=j)

```
1  .text
2          addi $t1, $zero, 2 #x = 2
3          addi $t2, $zero, 2 #y = 2
4          addi $t3, $zero, 2 #z = 2
5
6          addi $s1, $zero, 2      # i = 2
7          addi $s2, $zero, 2      # j = 2
8          slt $t0, $s2,$s1
9          bne $t0, $zero, else
10         addi $t1, $t1, 1
11         addi $t3, $zero, 1
12         j       endif
13 else:
14         addi $t2, $t2, -1
15         add $t3, $t3, $t3
16 endif:
17
```



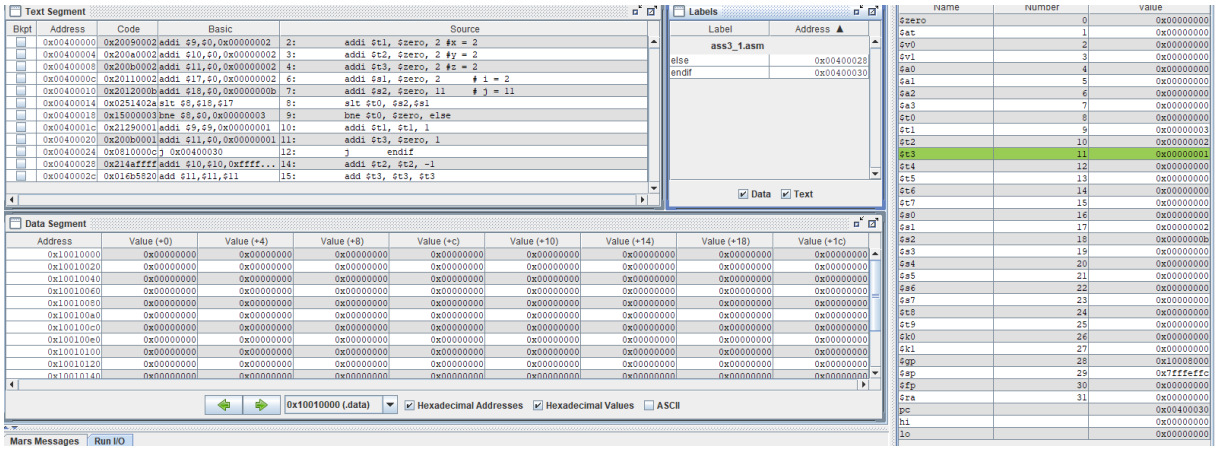
Nhận xét: sau lệnh slt, thanh ghi t0 thay đổi giá trị là 0, thanh ghi chứa x,y,z thay đổi, thanh ghi pc thay đổi

=>Kết quả thực thi đúng với lí thuyết

Trường hợp 3: i = 2, j = 11 (i<j)

```
.text
        addi $t1, $zero, 2 #x = 2
        addi $t2, $zero, 2 #y = 2
        addi $t3, $zero, 2 #z = 2

        addi $s1, $zero, 2      # i = 2
        addi $s2, $zero, 11     # j = 11
        slt $t0, $s2,$s1
        bne $t0, $zero, else
        addi $t1, $t1, 1
        addi $t3, $zero, 1
        j       endif
else:
        addi $t2, $t2, -1
        add $t3, $t3, $t3
endif:
```



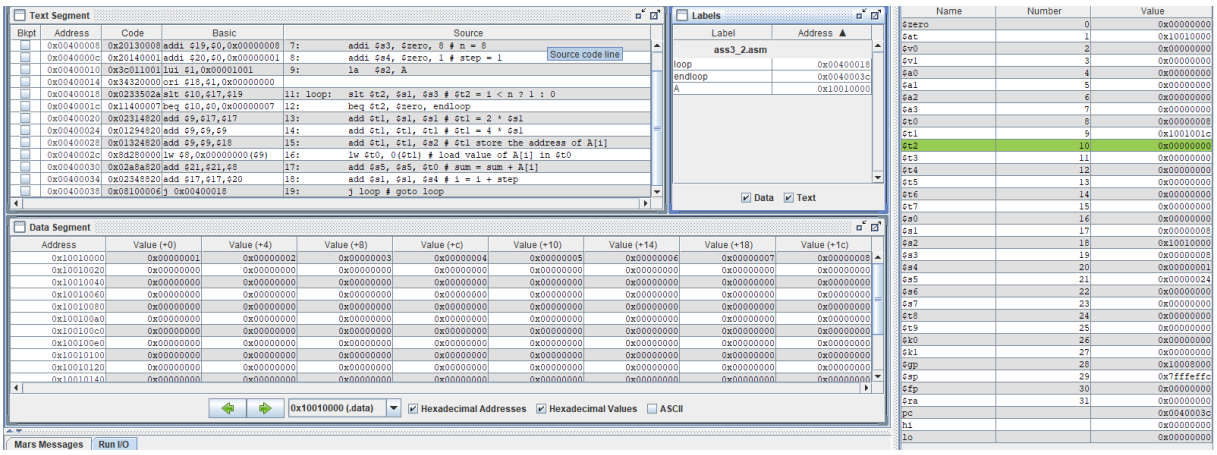
Nhận xét: sau lệnh slt, thanh ghi t0 thay đổi giá trị là 0, thanh ghi chứa x,y,z thay đổi, thanh ghi pc thay đổi

=>Kết quả thực thi đúng với lí thuyết

Assignment 2

```
.data
A: .word        1,2,3,4,5,6,7,8
 #the index i, the starting address of A, the comparison constant n, step and sum are found in registers $s1, $s2, $s3, $s4 and $s5
.text
        addi $s5, $zero, 0 # sum = 0
        addi $s1, $zero, 0 # i = 0
        addi $s3, $zero, 8 # n = 8
        addi $s4, $zero, 1 # step = 1
        la   $s2, A

loop:   slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
        beq $t2, $zero, endloop
        add $t1, $s1, $s1 # $t1 = 2 * $s1
        add $t1, $t1, $t1 # $t1 = 4 * $s1
        add $t1, $t1, $s2 # $t1 store the address of A[i]
        lw $t0, 0($t1) # load value of A[i] in $t0
        add $s5, $s5, $t0 # sum = sum + A[i]
        add $s1, $s1, $s4 # i = i + step
        j loop # goto loop
endloop:
```



Tổng được lưu ở thanh ghi $s5

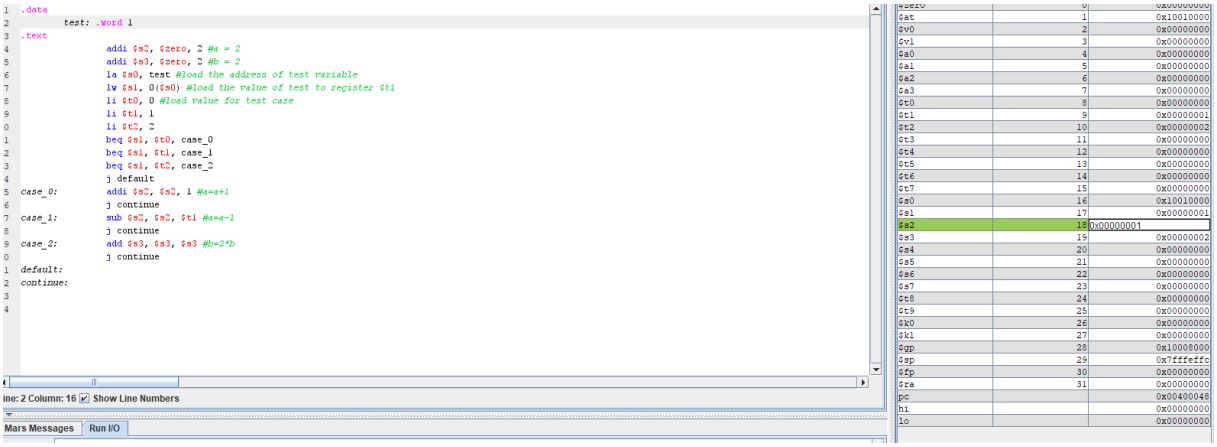
Khi kết thúc thực thi thì giá trị của thanh ghi $s5 là 0x00000024

=>Kết quả đúng vơi lý thuyết

Assignment 3:

Xét a=b=2

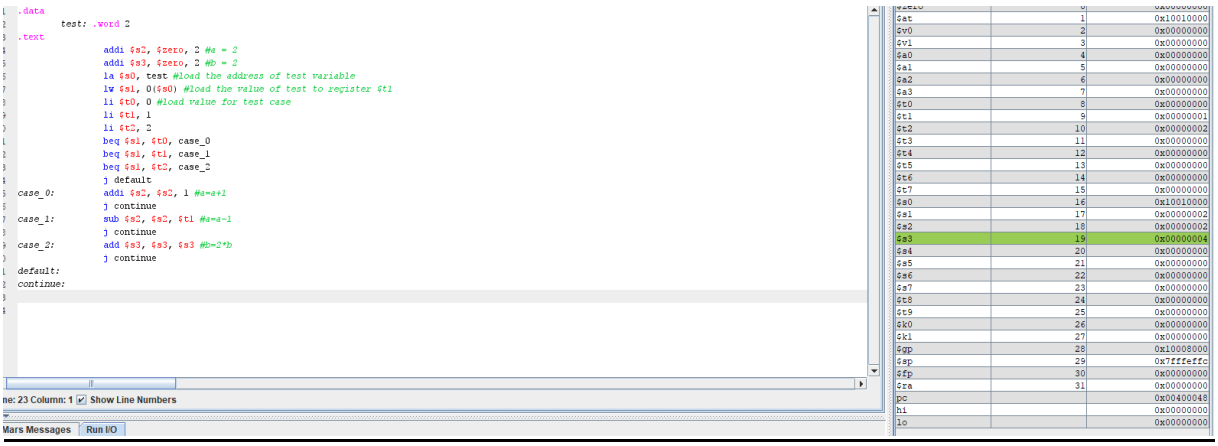
- Trường hợp 1: test = 1



Nhận xét: a = 0x00000001, b không thay đổi

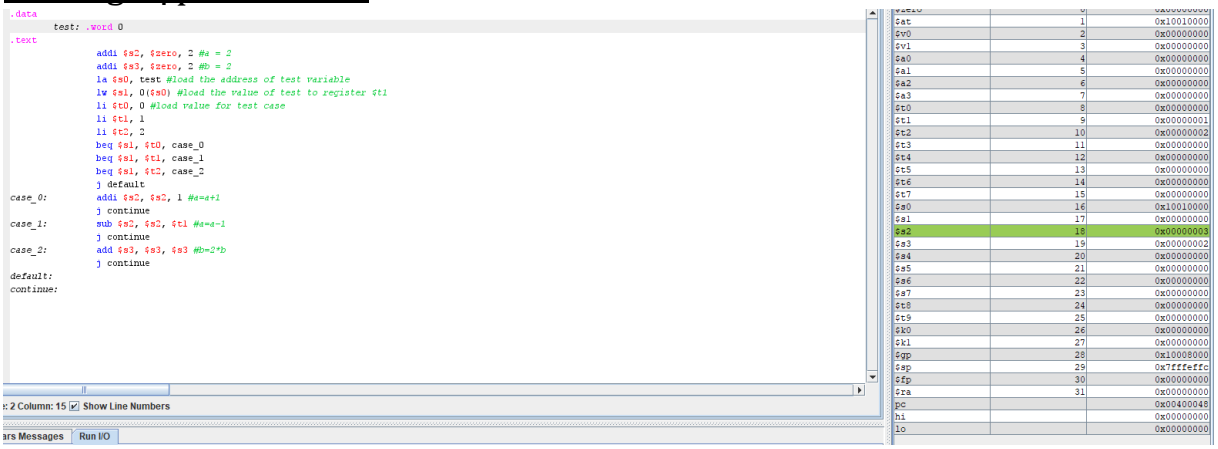
Kết quả đúng với lí thuyết

- Trường hợp 2: test = 2



Nhận xét: b= 0x00000004, a không thay đổi
Kết quả đúng với lí thuyết

- <u>Trường hợp 3: test = 0</u>



Nhận xét: a = 0x00000003, b không thay đổi

=>Kết quả đúng với lí thuyết

<u>Assignment 4</u>

a) i<j
Code:

```
.text
        addi $t1, $zero, 2 #x = 2
        addi $t2, $zero, 2 #y = 2
        addi $t3, $zero, 2 #z = 2

        addi $s1, $zero, 2  # i = 2
        addi $s2, $zero, 11         # j = 11
        slt $t0, $s1,$s2        #so sánh i < j ? 1:0
        beq $t0, $zero, else   # nếu t0 = 0 thì nhảy xuống else
```

```
        addi $t1, $t1, 1  # x = x+1
        addi $t3, $zero, 1  #z = 1
        j       endif
    else:
        addi $t2, $t2, -1  #y = y-1
        add $t3, $t3, $t3  # z = 2*z
    endif:
```

b) i>= j

Code:

```
    .text
        addi   $t1, $zero, 2 #x = 2
        addi $t2, $zero, 2 #y = 2
        addi $t3, $zero, 2 #z = 2

        addi $s1, $zero, 11        # i = 11
        addi $s2, $zero, 5  # j = 5
        slt $t0, $s1,$s2
        bne $t0, $zero, else
        addi $t1, $t1, 1
        addi $t3, $zero, 1
        j       endif
    else:
        addi $t2, $t2, -1
        add $t3, $t3, $t3
    endif:
```

c) i+j <= 0

Code:

```
    .text
        addi $t1, $zero, 2 #x = 2
        addi $t2, $zero, 2 #y = 2
        addi $t3, $zero, 2 #z = 2

        addi $s1, $zero, 11        # i = 11
        addi $s2, $zero, 5  # j = 5

        add $t4, $s1, $s2   #t4 = s1+s2 (tức là i+j)
        slt $t0, $zero,$t4
        bne $t0, $zero, else
```

```
        addi $t1, $t1, 1
        addi $t3, $zero, 1
        j       endif
else:
        addi $t2, $t2, -1
        add $t3, $t3, $t3
endif:
```

d) i+j>m+n

Code:

```
.text
        addi $t1, $zero, 2          #x = 2
        addi $t2, $zero, 2          #y = 2
        addi $t3, $zero, 2          #z = 2

        addi $s1, $zero, 11         # i = 11
        addi $s2, $zero, 5          # j = 5
        addi $s3, $zero, 3          # m = 3
        addi $s4, $zero, 2          # n = 2

        add $t4, $s1, $s2           #t4 = s1+s2 (tức là i+j)
        add $t5, $s3, $s4           # t5 = s3+s4 ( tức là m+n)

        slt $t0, $t5,$t4
        beq $t0, $zero, else
        addi $t1, $t1, 1
        addi $t3, $zero, 1
        j       endif
else:
        addi $t2, $t2, -1
        add $t3, $t3, $t3
endif:
```

Assignment 5:

a) i<=n

Code:

```
.data
A: .word 1,2,3,4,5,6,7,8,9,10
 #the index i, the starting address of A, the comparison constant n,
step and sum are found in registers $s1, $s2, $s3, $s4 and $s5
```

```
.text
    addi $s5, $zero, 0          # sum = 0
    addi $s1, $zero, 0          # i = 0
    addi $s3, $zero, 8          # n = 8
    addi $s4, $zero, 1          # step = 1
    la   $s2, A

loop:
    slt $t2, $s3, $s1           # $t2 = n < i ? 1 : 0
    bne $t2, $zero, endloop
    add $t1, $s1, $s1           # $t1 = 2 * $s1
    add $t1, $t1, $t1           # $t1 = 4 * $s1
    add $t1, $t1, $s2           # $t1 store the address of A[i]
    lw $t0, 0($t1)              # load value of A[i] in $t0
    add $s5, $s5, $t0           # sum = sum + A[i]
    add $s1, $s1, $s4           # i = i + step
    j loop                      # goto loop
endloop:
```

b) sum>=0

Code:

```
.data
A: .word 1,2,-4,3,4,5,6,7,8,9,10
 #the index i, the starting address of A, the comparison constant n,
step and sum are found in registers $s1, $s2, $s3, $s4 and $s5
.text
    addi $s5, $zero, 0 # sum = 0
    addi $s1, $zero, 0 # i = 0
    addi $s3, $zero, 8 # n = 8
    addi $s4, $zero, 1 # step = 1
    la   $s2, A

loop:     slt $t2, $s5, $zero # $t2 = sum < 0 ? 1 : 0
    bne $t2, $zero, endloop          # nếu sum<0 thì endloop
    add $t1, $s1, $s1 # $t1 = 2 * $s1
    add $t1, $t1, $t1 # $t1 = 4 * $s1
    add $t1, $t1, $s2 # $t1 store the address of A[i]
    lw $t0, 0($t1) # load value of A[i] in $t0
    add $s5, $s5, $t0 # sum = sum + A[i]
    add $s1, $s1, $s4 # i = i + step
```

```
    j loop # goto loop
endloop:
```

c) A[i] ! 0

Code:

```
.data
A: .word 1,2,0,3,4,5,6,7,8,9,10
 #the index i, the starting address of A, the comparison constant n,
step and sum are found in registers $s1, $s2, $s3, $s4 and $s5
.text
    addi $s5, $zero, 0 # sum = 0
    addi $s1, $zero, 0 # i = 0
    addi $s3, $zero, 8 # n = 8
    addi $s4, $zero, 1 # step = 1
    la   $s2, A

loop:
    add $t1, $s1, $s1 # $t1 = 2 * $s1
    add $t1, $t1, $t1 # $t1 = 4 * $s1
    add $t1, $t1, $s2 # $t1 store the address of A[i]
    lw $t0, 0($t1) # load value of A[i] in $t0
    beq $t0, $zero, endloop          # nếu A[i] = 0 thì endloop
    add $s5, $s5, $t0 # sum = sum + A[i]
    add $s1, $s1, $s4 # i = i + step
    j loop # goto loop
endloop:
```

Assignment 6:

```
.data
A: .word      1,2,0,-3,4,5,-10,7,-8,9
 #$s5 là max là phần tử có giá trị tuyệt đối lớn nhất
 #$s6 là vị trí của phần tử có trị tuyệt đối max
.text
        addi $s5, $zero, 0 #tamp = 0
        addi $s1, $zero, 0 # i = 0
        addi $s3, $zero, 10 # n = 10
```

```
        addi $s4, $zero, 1 # step = 1
        la   $s2, A

loop:
        slt $t2, $s1, $s3           # $t2 = i < n ? 1 : 0
        beq $t2, $zero, endloop
        add $t1, $s1, $s1           # $t1 = 2 * $s1
        add $t1, $t1, $t1           # $t1 = 4 * $s1
        add $t1, $t1, $s2           # $t1 store the address of A[i]
        lw $t0, 0($t1)                  # load value of A[i] in $t0

        slt $t4, $t0, $zero         # A[i] < 0 ? 1: 0
        beq $t4, $zero,duong
        sub $t0, $zero, $t0         #khi a[i] là số âm thì thực hiện lấy trị tuyệt đối
duong:
        slt $t4, $s5, $t0           # so sánh a[i] với max
        beq $t4, $zero, sai         # nếu a[i] < max thi nhay qua sai
        add $s5 , $zero, $t0            #max v= a[i]
        add $s6, $zero, $s1             #cập nhật vị trí mới
        j       cont
sai:
        addi $s5, $s5, 0
        addi $s6, $s6, 0
        j       cont
cont:
        add $s1, $s1, $s4           # i = i + step
        j loop # goto loop
endloop:
```

Mảng: A= {1,2,0,-3,4,5,-10,7,-8,9}

Kết quả: Vị trí có giá trị tuyệt đối lớn nhất là $s6 là 6

Giá trị có trị tuyệt đối lớn nhất là $s5 là 0xa

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000009
$t1	9	0x10010024
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x0000000a
$s2	18	0x10010000
$s3	19	0x0000000a
$s4	20	0x00000001
$s5	21	0x0000000a
$s6	22	0x00000006
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400064
hi		0x00000000
lo		0x00000000

=>Kết quả thực thi đúng với lí thuyết

Mảng: A= {1,2,0,-3,4,-15,10,7,-8,-14}

Kết quả: Vị trí có giá trị tuyệt đối lớn nhất là $s6 là 5

Giá trị có trị tuyệt đối lớn nhất là $s5 là 0xf

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x0000000e
$t1	9	0x10010024
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x0000000a
$s2	18	0x10010000
$s3	19	0x0000000a
$s4	20	0x00000001
$s5	21	0x0000000f
$s6	22	0x00000005
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400064
hi		0x00000000
lo		0x00000000