**BÀI BÁO CÁO TUẦN 3**

Họ và tên: Nguyễn Trọng Khánh Duy

MSSV: 20210284

Graphical user interface, text, application

Description automatically generated***Assignment 1***

Giả sử ta có x = 1, y = 2, z =3 và khởi tạo i = 2, j = 3

**- Sau khi chạy các lệnh:**

addi $s1,$zero,3

Graphical user interface, application, table, Excel

Description automatically generatedaddi $s2,$zero,2

Ta thấy: Các thanh ghi thay đổi giá trị là $s1 = 3, $s2 = 2 (khởi tạo i = 3, j = 2)

**- Sau khi chạy các lệnh:**

lw $t1,x

lw $t2,y

Graphical user interface, application, table, Excel

Description automatically generatedlw $t3,z

Ta thấy: Các thanh ghi $t1 = 1, $t2 = 2, $t3 = 3. VÌ khi thực hiện các câu lệnh lw thì sẽ lấy lần lượt các giá trị của các nhãn x, y ,z gán vào lần lượt các thanh ghi $t1, $t2, $t3

**- Sau khi chạy lệnh:**

Graphical user interface, application, table, Excel

Description automatically generatedslt $t0,$s2,$s1

Ta thấy: Ở câu lệnh này sẽ thực hiện so sánh $s2 < $s1 (j < i). nếu đúng thì thanh ghi $t0 được gán giá trị bằng 1, nếu sai thanh ghi $t0 được gán giá trị bằng 0. Sau khi chạy xong chương trình thì $t0 = 1 do $s2 < $s1 (2< 3) là kết quả đúng.

**- Sau khi chạy lệnh:**

bne $t0,$zero,else

Graphical user interface, application, table, Excel

Description automatically generated

Ta thấy: Câu lệnh so sánh thanh ghi $t0 với thanh ghi $zero, khi không bằng nhau thì sẽ nhảy đến else. Trong trường hợp này vì $t0 $zero nên thanh ghi pc sẽ nhảy đến địa chỉ của else và thực hiện tiếp chương trình

**- Sau khi chạy các lệnh:**

addi $t2,$t2,-1

Graphical user interface, application, table, Excel

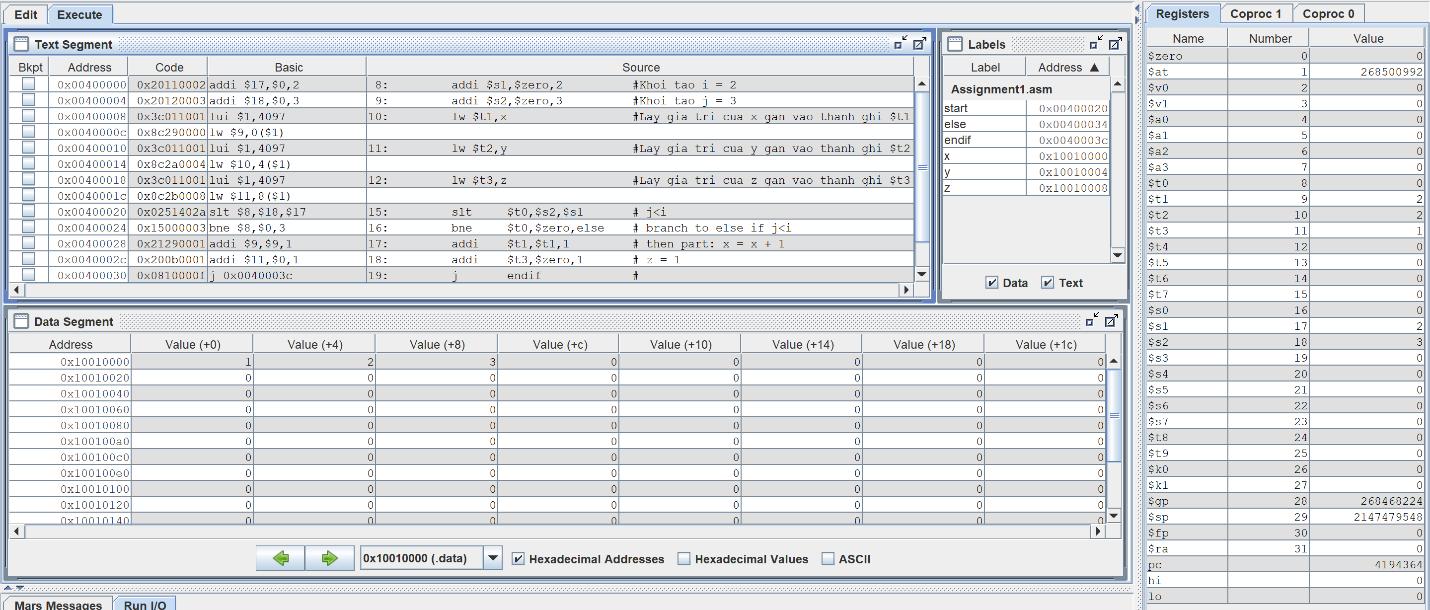
Description automatically generatedadd $t3,$t3,$t3

Ta thấy: Các câu lệnh thực hiện phép toán y = y – 1 và z = 2 \* z ⬄ z = z + z. Lúc này thanh ghi $t2 = 1, $t3 = 6.

Sau đó thanh ghi pc sẽ trỏ đến địa chỉ của câu nhãn *endif* và kết thúc chương trình tại đây

Tương tự khi ta thay đổi i, j sao cho j > i (ví dụ i = 2, j = 3) thì chương trình vẫn chạy như trên đến lúc gặp câu lệnh *slt $t0,$s2,$s1* thì điều kiện lúc này không đúng. Vì vậy thanh ghi pc sẽ nhảy đến những câu lệnh ngay sau đó để thực hiện và khi đến câu lệnh

*J endif* thì thanh ghi pc lại trỏ đến địa chỉ của câu nhãn *endif* và kết thúc chương trình tại đây.



***Assignment 2***

Graphical user interface, text, application

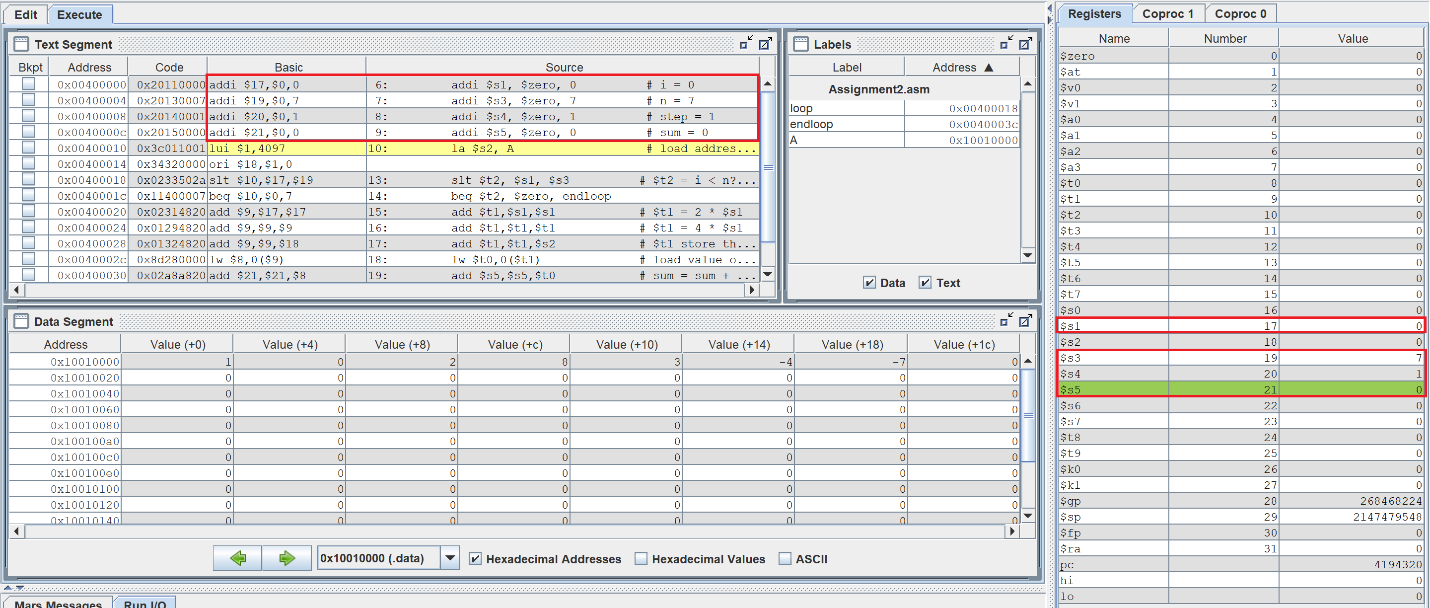
Description automatically generated

Khởi tạo mảng A gồm n = 7 phần tử {1, 0, 2, 8, 3, -4, -7} và i = 0; step = 1; sum = 0

**- Sau khi chạy các lệnh:**

addi $s1, $zero, 0

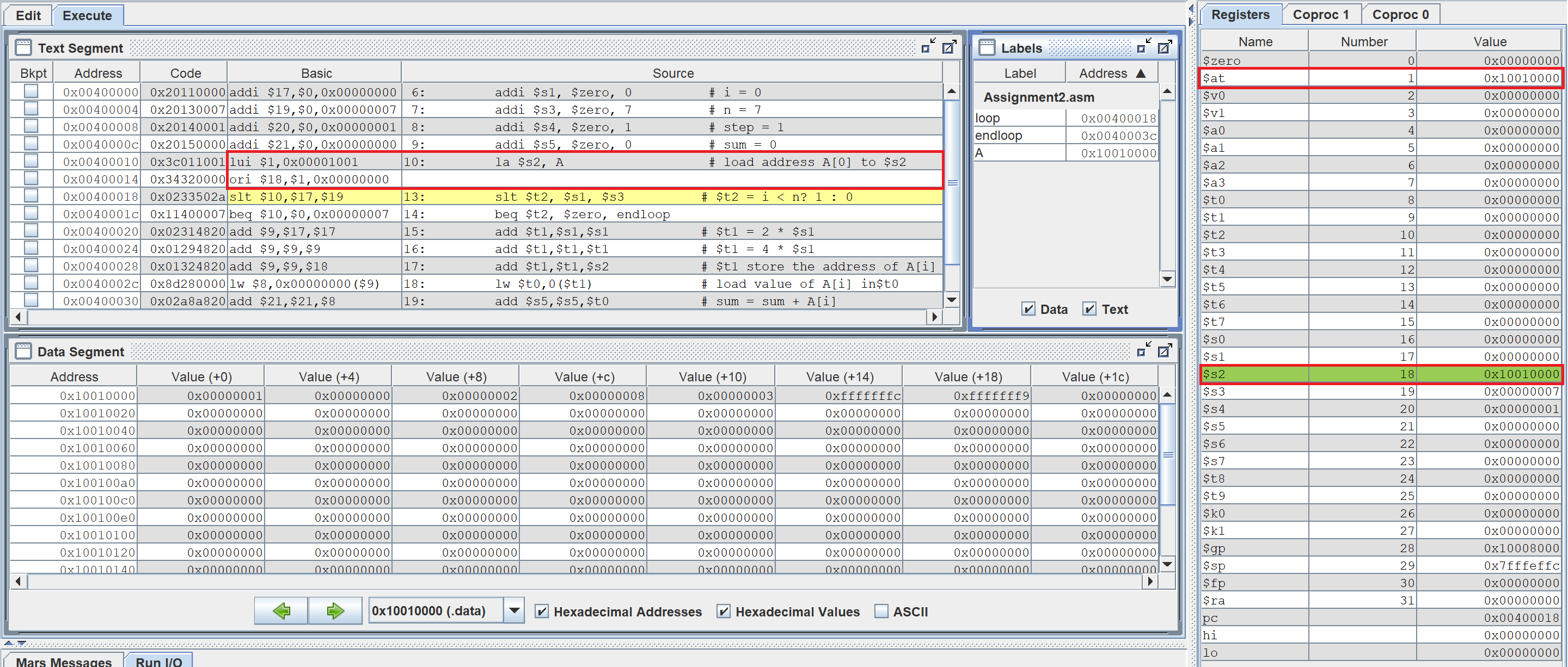
addi $s3, $zero, 7

addi $s4, $zero, 1

addi $s5, $zero, 0

Ta thấy: Các thanh ghi thay đổi giá trị là $s1 = 0, $s3 = 7, $s4 = 1, $s5 = 0 (khởi tạo i = 0, n = 7, step = 1, sum = 5)

**- Sau khi chạy các lệnh:**

la $s2, A

Ta thấy: Chương trình thực hiện 2 câu lệnh. Lúc đầu thực hiện lệnh *lui $s1, 0x00001001* thì thanh ghi $at được nạp vào địa chỉ mảng A. Sau đó thực hiện lệnh *ori $18, $1, 0x00000000* thì lúc này địa chỉ của mảng A sẽ được lưu vào thanh ghi $s2. Và đây cũng chính là địa chỉ cơ sở A[0].

**- Sau khi chạy lệnh:**

Graphical user interface, application, table, Excel

Description automatically generatedslt $t2, $s1, $s3

Ta thấy: Thanh ghi $t2 = 1 vì câu lệnh kiểm tra i < n không. Và trong trường hợp này câu lệnh đúng (do 0 < 7) nên thanh ghi $t2 = 1

**- Sau khi chạy lệnh:**

beq $t2, $zero, endloop

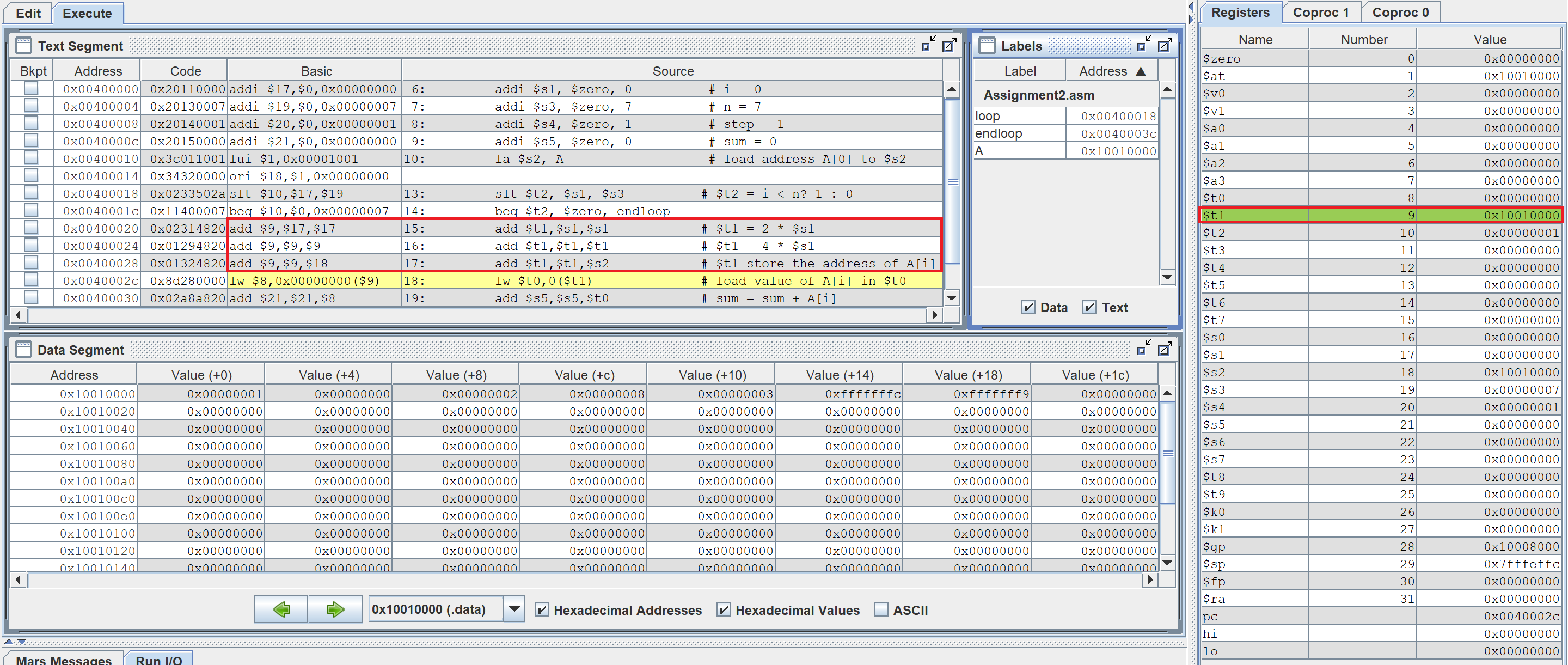
Vì $t2 $zero nên thanh ghi pc trỏ đến địa chỉ của câu lệnh tiếp theo mà không nhảy đến nhãn *endloop*. Vì thế chương trình sẽ chạy tiếp đến câu lệnh add $t1, $s1, $s1

**- Sau khi chạy các lệnh:**

add $t1, $s1, $s1

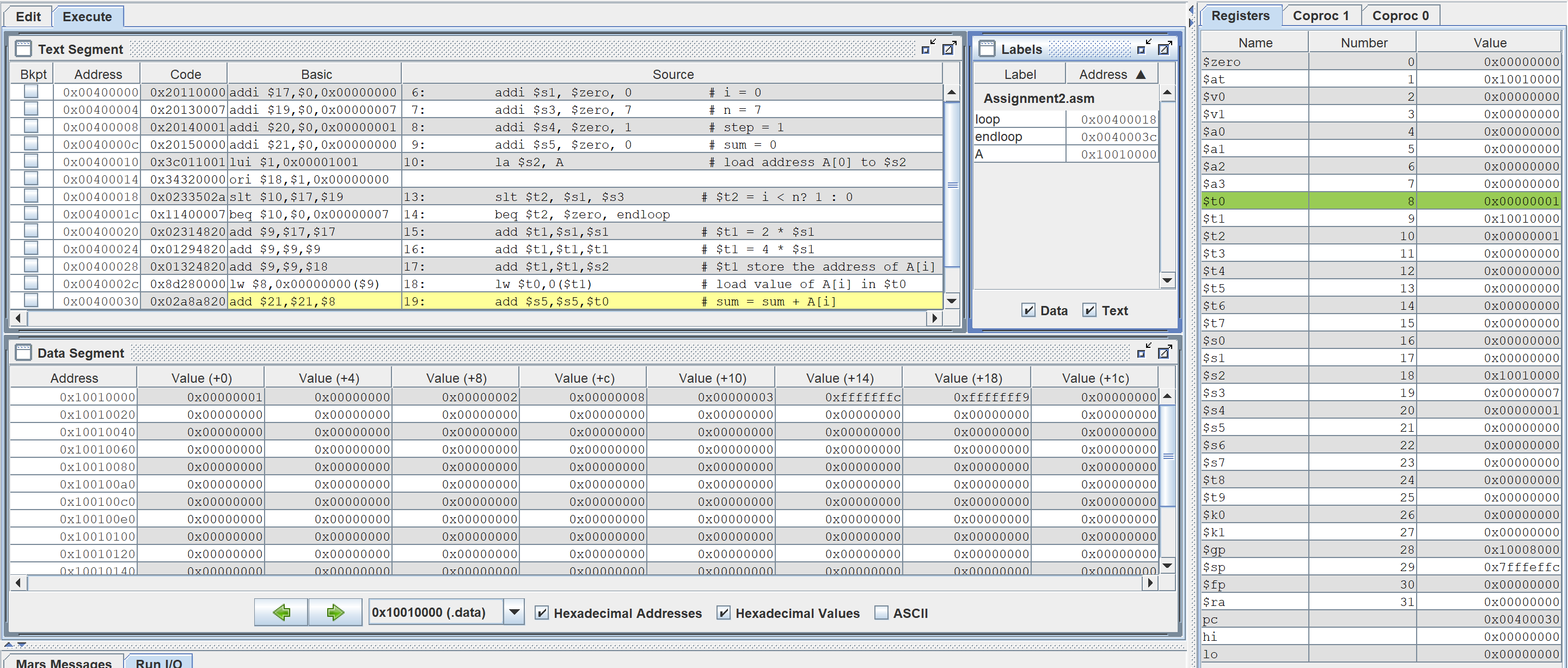
add $t1, $t1, $t1

add $t1, $t1, $s2



Ta thấy : Lúc này thanh ghi $t1 đang lưu địa chỉ của A[i]

**- Sau khi chạy lệnh:**

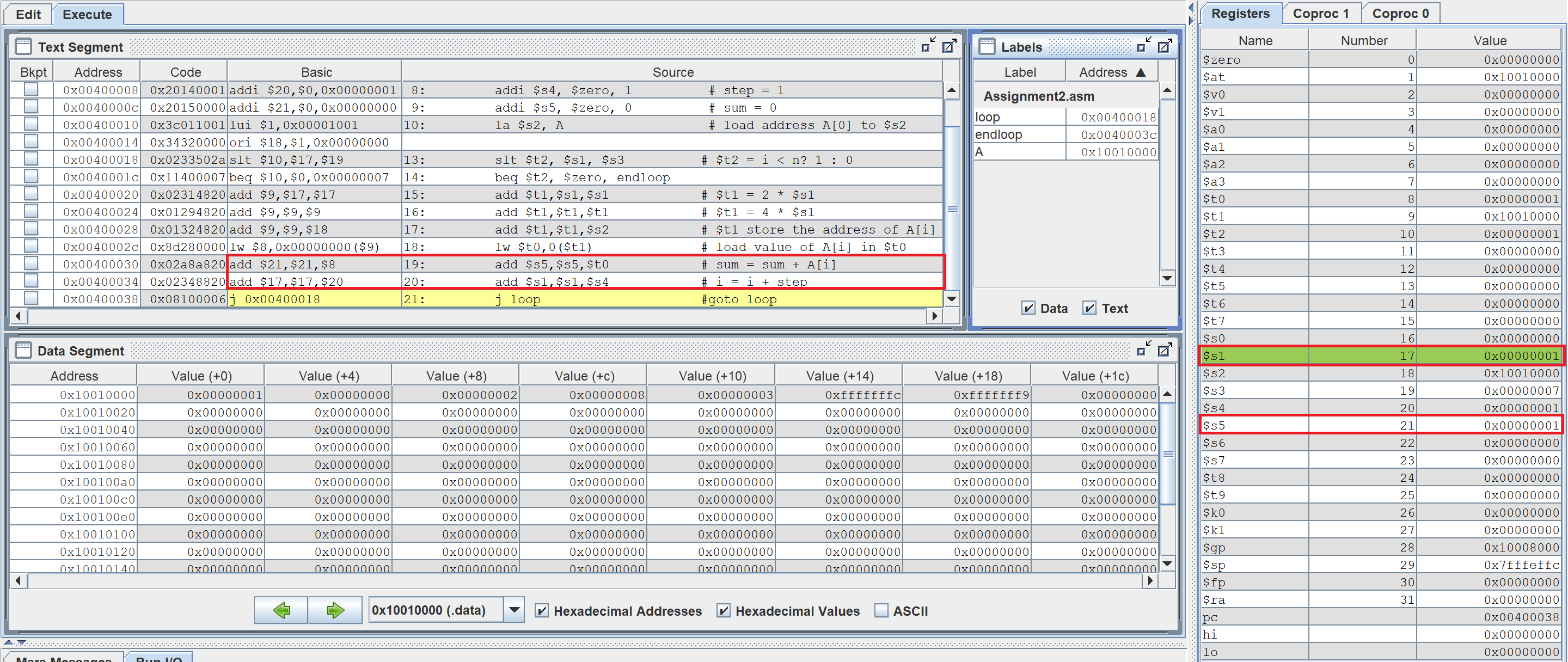
lw $t0, 0($t1)

Ta thấy: Thanh ghi $t0 được nạp vào giá trị của ô nhớ có địa chỉ $t1. Và lúc này thanh ghi $t0 có giá trị = 1 vì A[0] = 1

**- Sau khi chạy các lệnh:**

add $s5, $s5, $t0

add $s1, $s1, $s4



Ta thấy: Các thanh ghi lần lượt được cập nhật giá $s5 = 1, $s1 = 1 vì thực hiện lần lượt 2 câu lệnh *sum = sum + A[i]* và *i = i + step*.

**- Sau khi chạy lệnh:**

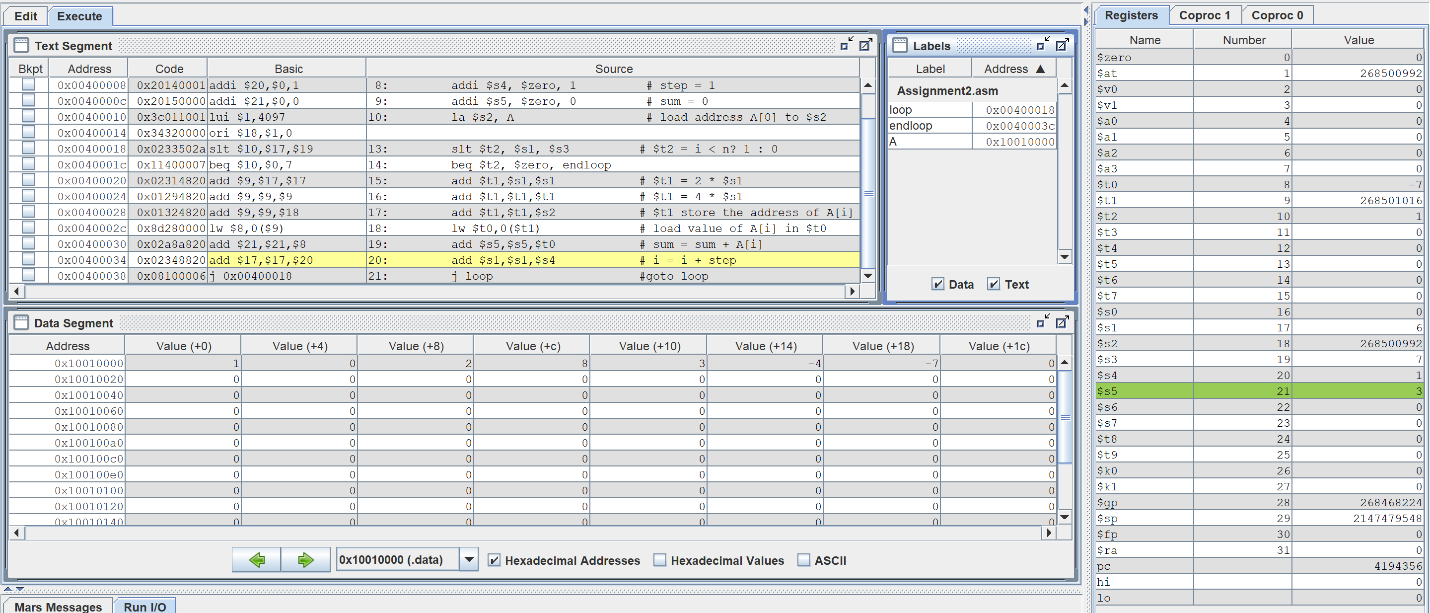
j loop

Graphical user interface, application, table, Excel

Description automatically generated

Câu lệnh nhảy lên nhãn *loop* và thực hiện các câu lệnh bên trong nhãn *loop* và cứ lặp đi lặp lại các bước trên đến lúc thanh ghi $s1 = 5 (i = 5). Khi thực hiện câu lệnh *beq $t2, $zero, endloop* thì *$t2 = $zero* nên câu lệnh sẽ nhảy đến nhãn *endloop*  và kết thúc chương trình.

Kết quả cuối cùng trả về sum = $s5 = 3



***Assignment 3***

Graphical user interface, application

Description automatically generated

Dữ liệu của *test* được nạp vào thanh ghi $s1. Sau đó ta so sánh giá trị thanh ghi $s1 lần lượt với các giá trị thanh ghi $t0, $t1, $t2

$s1 = $t0 => nhảy tới *case\_0*

$s1 = $t1 => nhảy tới case\_1

$s1 = $t2 => nhảy tới *case\_2*

Khi cả ba đều không bằng$s1thì lúc chạy đến lệnh *j default* thì lệnh sẽ nhảy đến nhãn *default* và kết thúc chương trình tại đây.

Trong trường hợp này $s1 = $t1 = 1 nên sẽ thực hiện lệnh trong *case\_1*

=> Kết quả thu được: $s2 = $s2 – 1 = 3 – 1 = 2

***Assignment 4***

**a) i < j (i = 2; j = 3)**

Graphical user interface, text, application, email

Description automatically generated

Code:

#Laboratory Exercise 3, Home Assignment 4

#i < j

.data

x: .word 1

y: .word 2

z: .word 3

.text

addi $s1,$zero,2 #Khoi tao i = 2

addi $s2,$zero,3 #Khoi tao j = 3

lw $t1,x #Lay gia tri cua x gan vao thanh ghi $t1

lw $t2,y #Lay gia tri cua y gan vao thanh ghi $t2

lw $t3,z #Lay gia tri cua z gan vao thanh ghi $t3

start:

slt $t0,$s1, $s2 # i < j

beq $t0,$zero,else # branch to else if i >= j

addi $t1,$t1,1 # then part: x = x + 1

addi $t3,$zero,1 # z = 1

j endif

else:

addi $t2,$t2,-1 # begin else part: y = y - 1

add $t3,$t3,$t3 # z = 2 \* z

endif:

**b) i > = j (i = 3; j = 3)**

Graphical user interface, text, application

Description automatically generated

Code:

#Laboratory Exercise 3, Home Assignment 4

#i >= j

.data

x: .word 1

y: .word 2

z: .word 3

.text

addi $s1,$zero,3 #Khoi tao i = 3

addi $s2,$zero,3 #Khoi tao j = 3

lw $t1,x #Lay gia tri cua x gan vao thanh ghi $t1

lw $t2,y #Lay gia tri cua y gan vao thanh ghi $t2

lw $t3,z #Lay gia tri cua z gan vao thanh ghi $t3

start:

slt $t0,$s1, $s2 # i < j

bne $t0,$zero,else # branch to else if i < j

addi $t1,$t1,1 # then part: x = x + 1

addi $t3,$zero,1 # z = 1

j endif

else:

addi $t2,$t2,-1 # begin else part: y = y - 1

add $t3,$t3,$t3 # z = 2 \* z

endif:

**c) i + j < = 0 (i = 3; j = -3)**

Graphical user interface, text, application

Description automatically generated

Code:

#Laboratory Exercise 3, Home Assignment 4

#i + j <= 0

.data

x: .word 1

y: .word 2

z: .word 3

.text

addi $s1,$zero,3 #Khoi tao i = 3

addi $s2,$zero,-5 #Khoi tao j = -5

lw $t1,x #Lay gia tri cua x gan vao thanh ghi $t1

lw $t2,y #Lay gia tri cua y gan vao thanh ghi $t2

lw $t3,z #Lay gia tri cua z gan vao thanh ghi $t3

start:

add $t4, $s1, $s2 # $t4 = i + j

sgt $t0,$t2, 0 # i + j > 0

beq $t0,$zero,else # branch to else if i + j > 0

addi $t1,$t1,1 # then part: x = x + 1

addi $t3,$zero,1 # z = 1

j endif

else:

addi $t2,$t2,-1 # begin else part: y = y - 1

add $t3,$t3,$t3 # z = 2 \* z

endif:

**d) i + j > m + n**

Graphical user interface, text, application

Description automatically generated

Code:

#Laboratory Exercise 3, Assignment 4

# i + j > m + n

.data

x: .word 1

y: .word 2

z: .word 3

.text

addi $s1,$zero,3 #Khoi tao i = 3

addi $s2,$zero,4 #Khoi tao j = 4

addi $s3,$zero,3 #Khoi tao m = 3

addi $s4,$zero,2 #Khoi tao n = 2

lw $t1,x #Lay gia tri cua x gan vao thanh ghi $t1

lw $t2,y #Lay gia tri cua y gan vao thanh ghi $t2

lw $t3,z #Lay gia tri cua z gan vao thanh ghi $t3

start:

add $t4, $s1, $s2 # $t4 = i + j

add $t5, $s3, $s4 # $t5 = m + n

sgt $t0,$t4,$t5 # i + j > m + n

beq $t0,$zero,else # branch to else if i + j <= m + n

addi $t1,$t1,1 # then part: x = x + 1

addi $t3,$zero,1 # z = 1

j endif

else:

addi $t2,$t2,-1 # begin else part: y = y - 1

add $t3,$t3,$t3 # z = 2 \* z

endif:

***Assignment 5***

Graphical user interface, text, application, email

Description automatically generateda) i < n

Code:

#Laboratory 3, Assigment 5

# i < n

.data

A: .word 1, 0, 2, 8, 3, -4, -7

.text

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 7

addi $s4, $zero, 1 # step = 1

addi $s5, $zero, 0 # sum = 0

la $s2, A # load address A[0] to $s2

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:

Kết quả Sum = $s5 = 3

b) i < = n

Graphical user interface, text, application

Description automatically generated

Code:

#Laboratory 3, Assigment 5

# i <= n

.data

A: .word 1, 0, 2, 8, 3, -4, -7

.text

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 7

addi $s4, $zero, 1 # step = 1

addi $s5, $zero, 0 # sum = 0

la $s2, A # load address A[0] to $s2

loop:

sgt $t2, $s1, $s3 # $t2 = i > n? 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:

Kết quả Sum = $s5 = 3

c) sum >= 0

Graphical user interface, text, application

Description automatically generated

Code:

#Laboratory 3, Assigment 5

# sum >= 0

.data

A: .word 1, 0, 2, 8, -8, -4, -7

.text

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 7

addi $s4, $zero, 1 # step = 1

addi $s5, $zero, 0 # sum = 0

la $s2, A # load address A[0] to $s2

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

add $s5,$s5,$t0 # sum = sum + A[i]

slt $t2, $s5, $zero # $t2 = sum < 0? 1 : 0

bne $t2, $zero, endloop

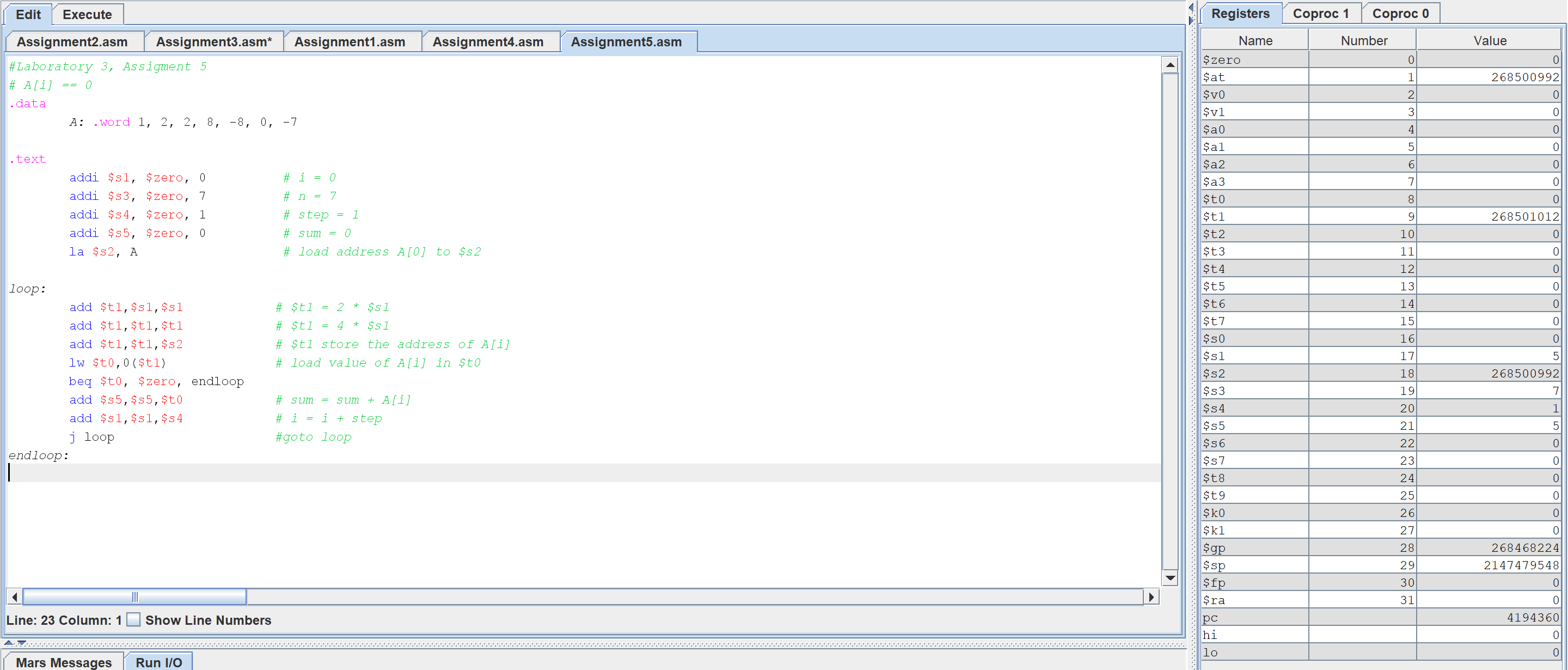
add $s1,$s1,$s4 # i = i + step

j loop #goto loop

endloop:

Kết quả Sum = $s5 = -1

d) A[i] == 0



Code:

#Laboratory 3, Assigment 5

# A[i] == 0

.data

A: .word 1, 2, 2, 8, -8, 0, -7

.text

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 7

addi $s4, $zero, 1 # step = 1

addi $s5, $zero, 0 # sum = 0

la $s2, A # load address A[0] to $s2

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

add $s5,$s5,$t0 # sum = sum + A[i]

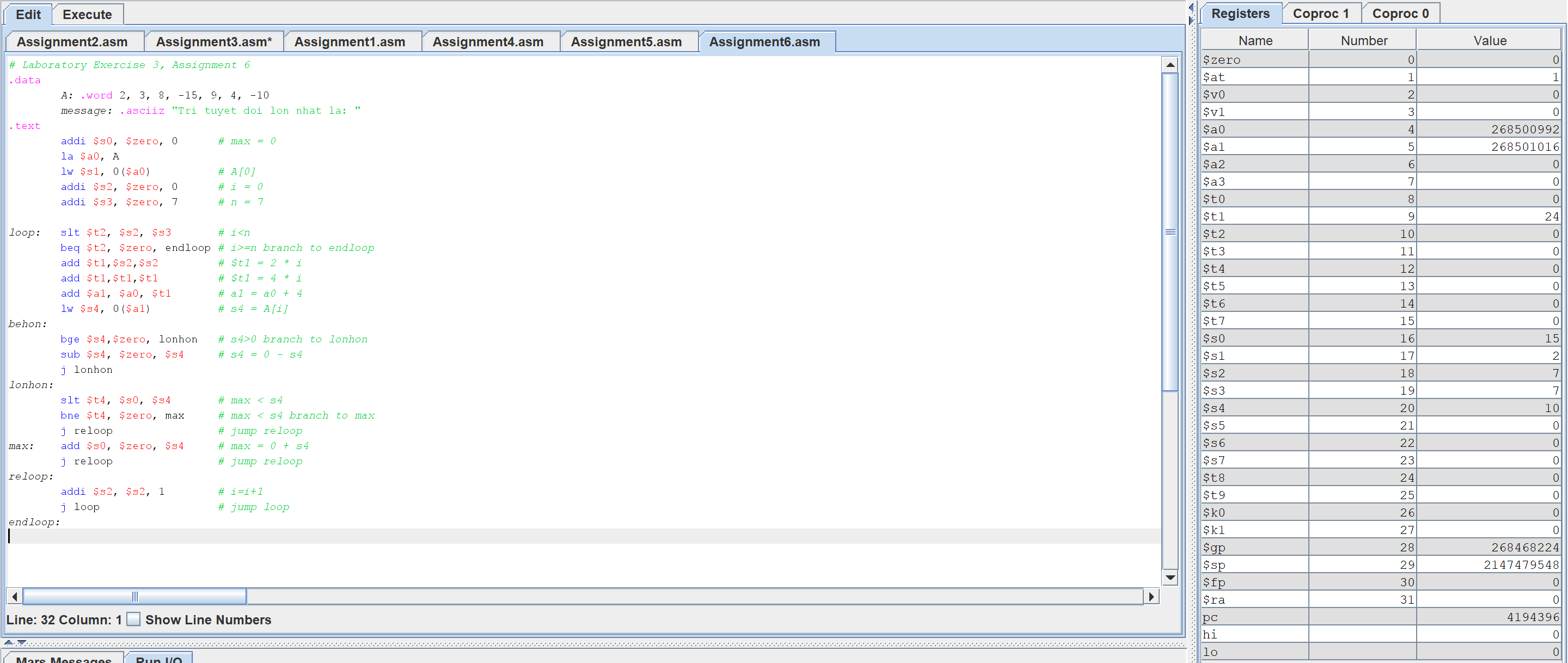
add $s1,$s1,$s4 # i = i + step

j loop #goto loop

endloop:

Kết quả Sum = $s5 = 5

***Assignment 6***



Code:

# Laboratory Exercise 3, Assignment 6

.data

A: .word 2, 3, 8, -15, 9, 4, -10

message: .asciiz "Tri tuyet doi lon nhat la: "

.text

addi $s0, $zero, 0 # max = 0

la $a0, A

lw $s1, 0($a0) # A[0]

addi $s2, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 7

loop: slt $t2, $s2, $s3 # i<n

beq $t2, $zero, endloop # i>=n branch to endloop

add $t1,$s2,$s2 # $t1 = 2 \* i

add $t1,$t1,$t1 # $t1 = 4 \* i

add $a1, $a0, $t1 # a1 = a0 + 4

lw $s4, 0($a1) # s4 = A[i]

behon:

bge $s4,$zero, lonhon # s4>0 branch to lonhon

sub $s4, $zero, $s4 # s4 = 0 - s4

j lonhon

lonhon:

slt $t4, $s0, $s4 # max < s4

bne $t4, $zero, max # max < s4 branch to max

j reloop # jump reloop

max: add $s0, $zero, $s4 # max = 0 + s4

j reloop # jump reloop

reloop:

addi $s2, $s2, 1 # i=i+1

j loop # jump loop

endloop:

Phần tử có giá trị tuyệt đối lớn nhất là $s0 = 15