Báo cáo thực hành kiến trúc máy tính

Assignment 1:

Home Assignment 1

This home assignment implements "if-then-else" statement using some fundamental instructions, such as slt, addi, jump and branch.

```
if (i<=j)
        x=x+1;
        z=1;
else
        y=y-1;
        z=2*z;
```

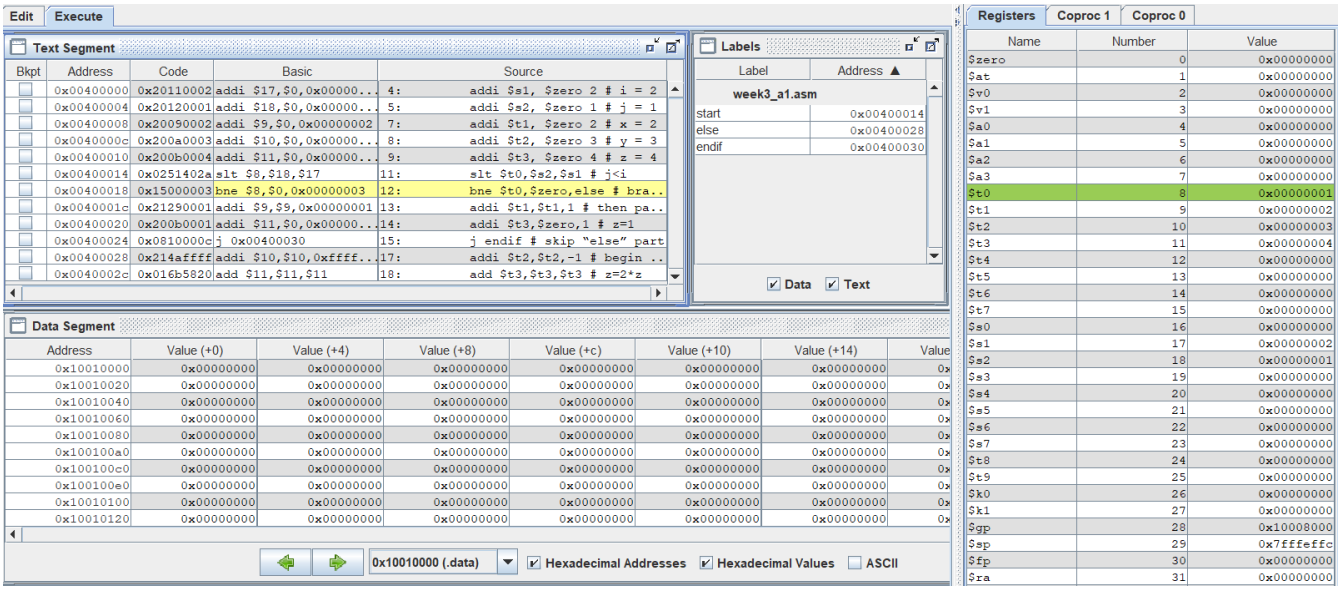
week3_a1.asm

```
1  #Laboratory Exercise 3, Home Assignment 1
2
3  .text
4          addi $s1, $zero 2 # i = 2
5          addi $s2, $zero 1 # j = 1
6
7          addi $t1, $zero 2 # x = 2
8          addi $t2, $zero 3 # y = 3
9          addi $t3, $zero 4 # z = 4
10 start:
11         slt $t0,$s2,$s1 # j<i
12         bne $t0,$zero,else # branch to else if j<i
13         addi $t1,$t1,1 # then part: x=x+1
14         addi $t3,$zero,1 # z=1
15         j endif # skip "else" part
16 else:
17         addi $t2,$t2,-1 # begin else part: y=y-1
18         add $t3,$t3,$t3 # z=2*z
19 endif:
```

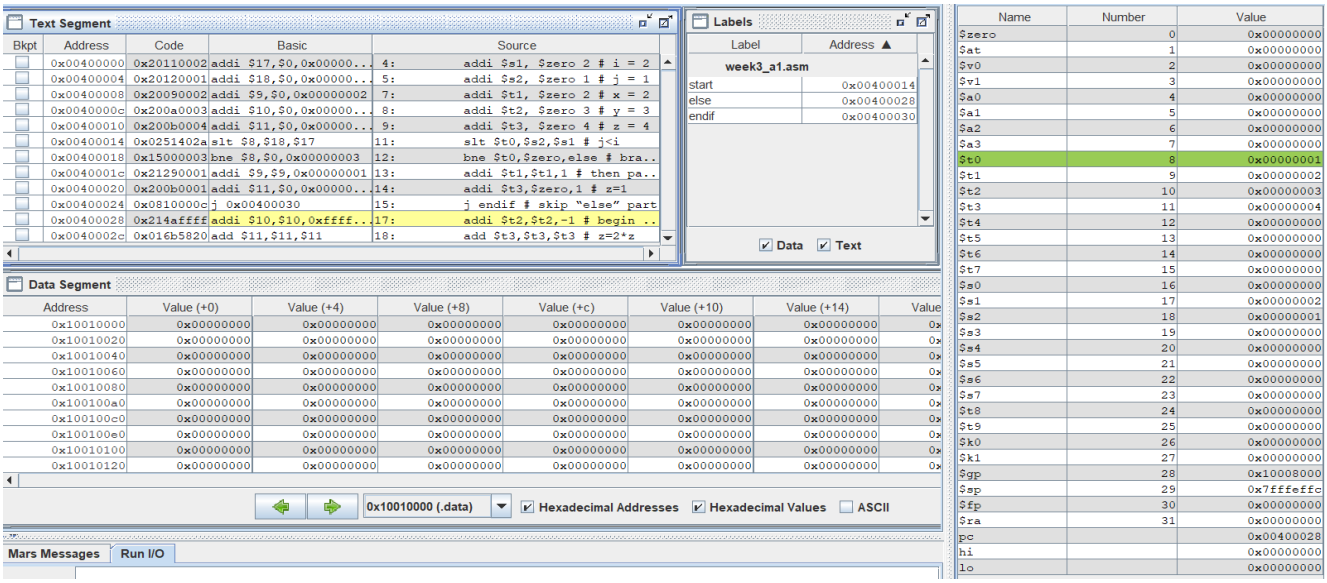
Khởi tạo giá trị i = 2, j = 1, x = 2, y = 3, z = 4

$t1	9	0x00000002
$t2	10	0x00000003
$t3	11	0x00000004
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000001

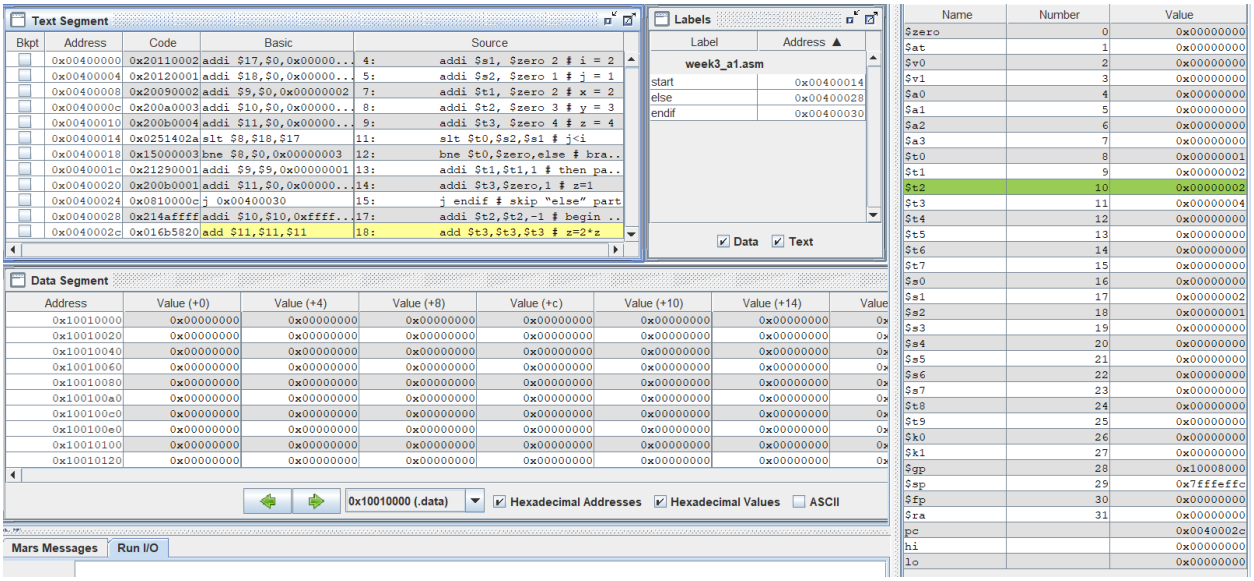
- So sánh j và i (1<2) $t0 = 1



So sánh $t0 = 1, $zero = 0 nhảy đến lệnh else



- Tính y = y -1 (3 – 1 = 2)



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110002	addi $17,$0,0x00000...	4: addi $s1, $zero 2 # i = 2
	0x00400004	0x20120001	addi $18,$0,0x00000...	5: addi $s2, $zero 1 # j = 1
	0x00400008	0x20090002	addi $9,$0,0x00000002	7: addi $t1, $zero 2 # x = 2
	0x0040000c	0x200a0003	addi $10,$0,0x00000...	8: addi $t2, $zero 3 # y = 3
	0x00400010	0x200b0004	addi $11,$0,0x00000...	9: addi $t3, $zero 4 # z = 4
	0x00400014	0x0251402a	slt $8,$18,$17	11: slt $t0,$s2,$s1 # j<i
	0x00400018	0x15000003	bne $8,$0,0x00000003	12: bne $t0,$zero,else # bra..
	0x0040001c	0x21290001	addi $9,$9,0x00000001	13: addi $t1,$t1,1 # then pa..
	0x00400020	0x200b0001	addi $11,$0,0x00000...	14: addi $t3,$zero,1 # z=1
	0x00400024	0x0810000c	j 0x00400030	15: j endif # skip "else" part
	0x00400028	0x214affff	addi $10,$10,0xffff...	17: addi $t2,$t2,-1 # begin ..
	0x0040002c	0x016b5820	add $11,$11,$11	18: add $t3,$t3,$t3 # z=2*z

Labels

Label	Address ▲
week3_a1.asm	
start	0x00400014
else	0x00400028
endif	0x00400030

Data Segment

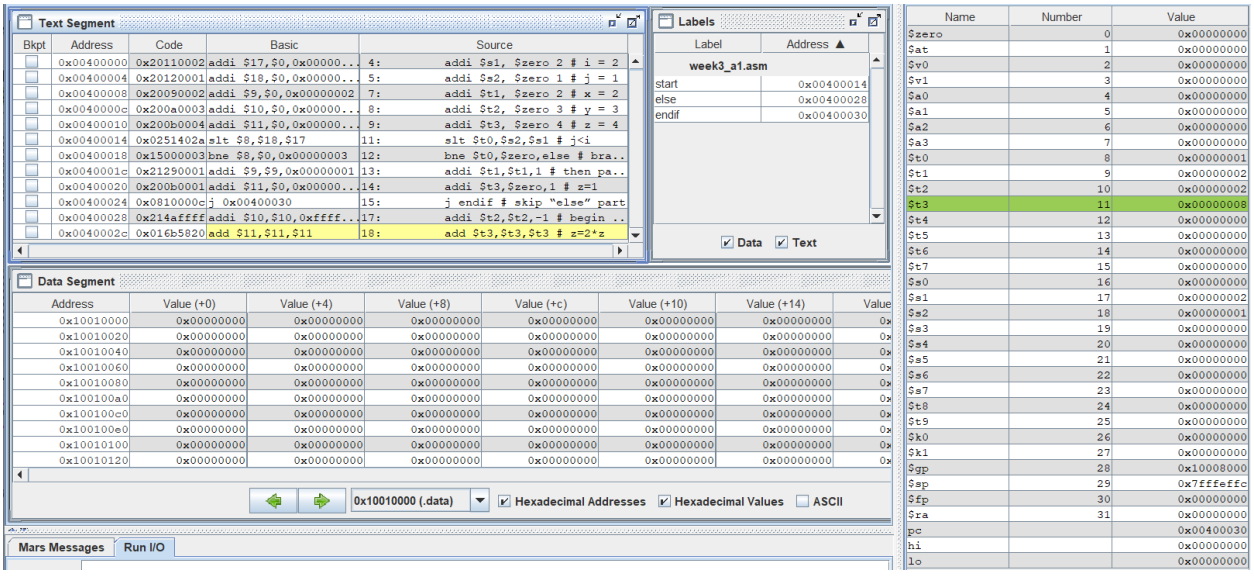
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Mars Messages | Run I/O

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000001
$t1	9	0x00000002
$t2	10	0x00000002
$t3	11	0x00000004
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000001
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$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		0x0040002c
hi		0x00000000
lo		0x00000000

- Tính z = z*2 (2*4=8)



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110002	addi $17,$0,0x00000...	4: addi $s1, $zero 2 # i = 2
	0x00400004	0x20120001	addi $18,$0,0x00000...	5: addi $s2, $zero 1 # j = 1
	0x00400008	0x20090002	addi $9,$0,0x00000002	7: addi $t1, $zero 2 # x = 2
	0x0040000c	0x200a0003	addi $10,$0,0x00000...	8: addi $t2, $zero 3 # y = 3
	0x00400010	0x200b0004	addi $11,$0,0x00000...	9: addi $t3, $zero 4 # z = 4
	0x00400014	0x0251402a	slt $8,$18,$17	11: slt $t0,$s2,$s1 # j<i
	0x00400018	0x15000003	bne $8,$0,0x00000003	12: bne $t0,$zero,else # bra..
	0x0040001c	0x21290001	addi $9,$9,0x00000001	13: addi $t1,$t1,1 # then pa..
	0x00400020	0x200b0001	addi $11,$0,0x00000...	14: addi $t3,$zero,1 # z=1
	0x00400024	0x0810000c	j 0x00400030	15: j endif # skip "else" part
	0x00400028	0x214affff	addi $10,$10,0xffff...	17: addi $t2,$t2,-1 # begin ..
	0x0040002c	0x016b5820	add $11,$11,$11	18: add $t3,$t3,$t3 # z=2*z

Labels

Label	Address ▲
week3_a1.asm	
start	0x00400014
else	0x00400028
endif	0x00400030

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Mars Messages | Run I/O

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000001
$t1	9	0x00000002
$t2	10	0x00000002
$t3	11	0x00000008
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000001
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$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		0x00400030
hi		0x00000000
lo		0x00000000

Assignment 2:

Home Assignment 2

The following example demonstrates how to implement loop statement. This program computes the sum of elements of array A.

With C language:

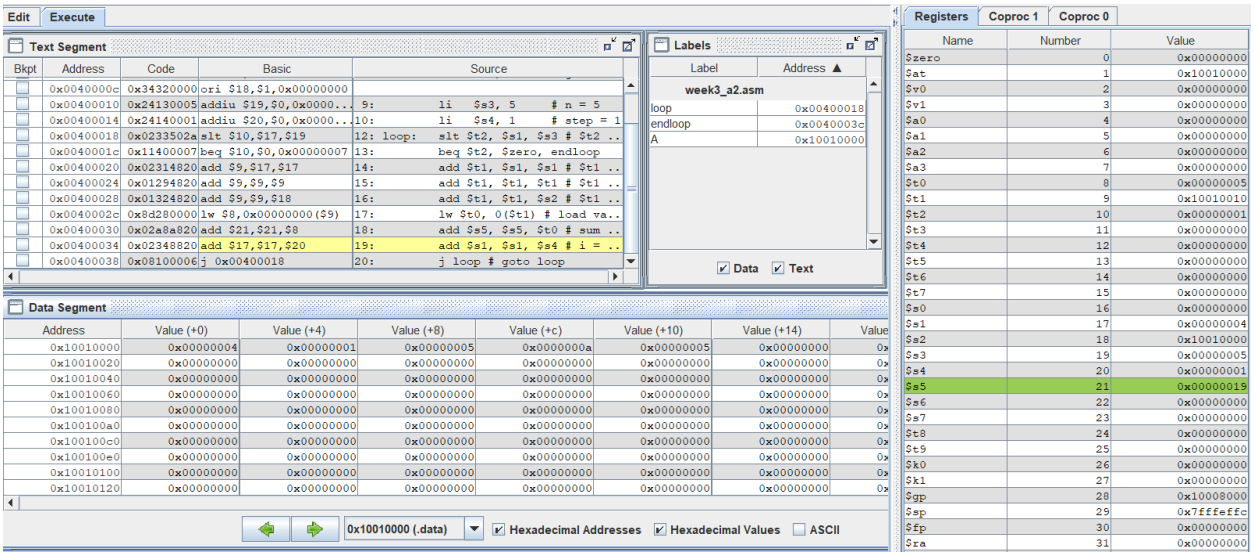
```
sum = 0;
for (int i = 0; i < n; i += step)
    sum += A[i];
```

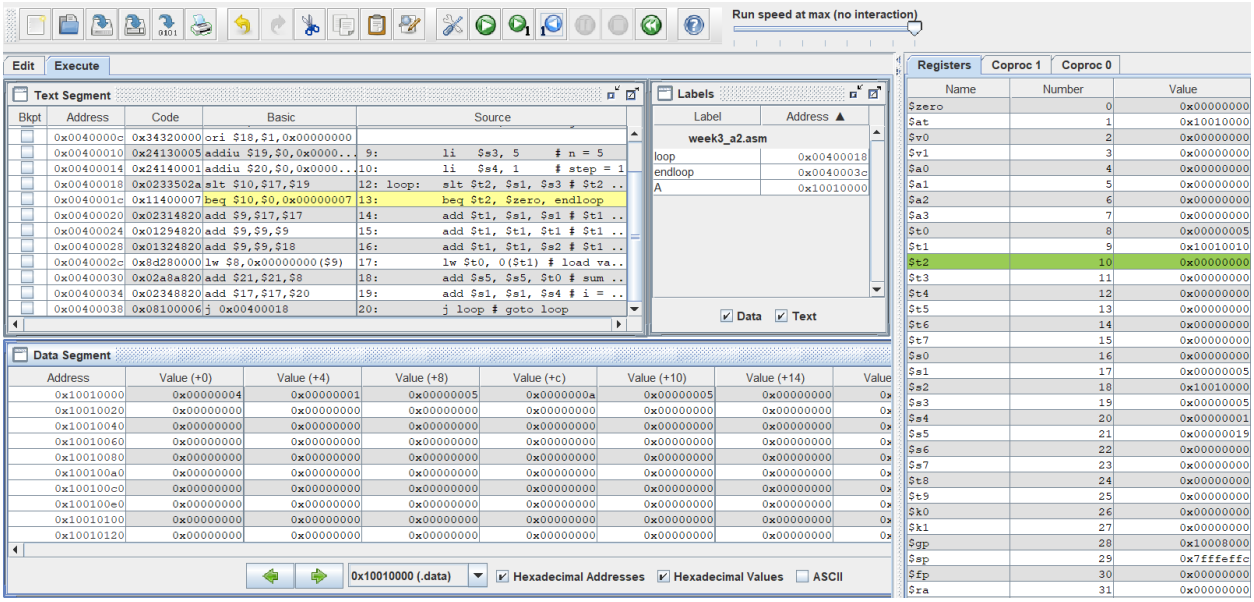
With assembly language:

```
        sum = 0
        i = 0
loop:   if (i >= n) goto endloop
        sum = sum + A[i]
        i = i + step
        goto loop
endloop:
```

week3_a2.asm

```
1  #Laboratory 3, Home Assigment 2
2  .data
3  A: .word  4,1,5,10,5
4  .text
5          addi $s5, $zero, 0 # sum = 0
6          addi $s1, $zero, 0 # i = 0
7
8          la   $s2, A     # gan dia chi dau mang A cho $s2
9          li   $s3, 5     # n = 5
10         li   $s4, 1     # step = 1
11
12 loop:    slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
13         beq $t2, $zero, endloop
14         add $t1, $s1, $s1 # $t1 = 2 * $s1
15         add $t1, $t1, $t1 # $t1 = 4 * $s1
16         add $t1, $t1, $s2 # $t1 store the address of A[i]
17         lw $t0, 0($t1) # load value of A[i] in $t0
18         add $s5, $s5, $t0 # sum = sum + A[i]
19         add $s1, $s1, $s4 # i = i + step
20         j loop # goto loop
21 endloop:
```





Kết quả cuối 0x0000 0019 = 25 (= 4 + 1 + 5 + 10 +5)

Assignment 3:

Home Assignment 3

A switch/case statement allows multiway branching based on the value of an integer variable. In the following example, the switch variable test can assume one of the three values in [0, 2] and a different action is specified for each case.

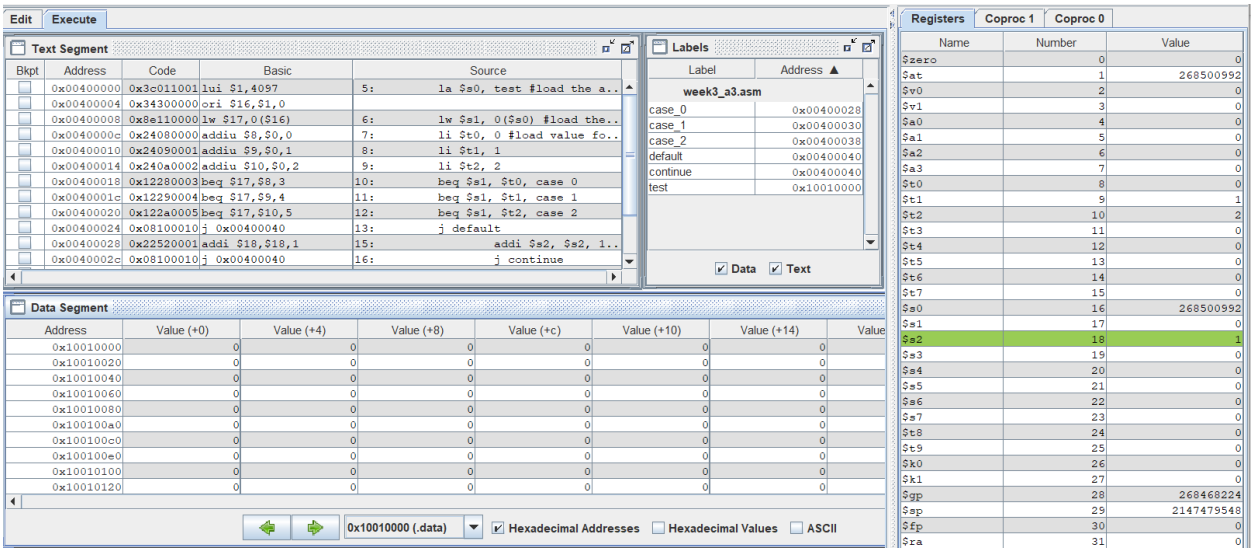
```
switch(test) {
      case 0:
            a=a+1; break;
      case 1:
            a=a-1; break;
      case 2:
            b=2*b; break;
      }
```

TH test =0

week3_a3.asm

```
1  #Laboratory Exercise 3, Home Assignment 3
2  .data
3  test: .word 0
4  .text
5          la $s0, test #load the address of test variable
6          lw $s1, 0($s0) #load the value of test to register $t1
7          li $t0, 0 #load value for test case
8          li $t1, 1
9          li $t2, 2
10         beq $s1, $t0, case_0
11         beq $s1, $t1, case_1
12         beq $s1, $t2, case_2
13         j default
14         case_0:
15                 addi $s2, $s2, 1 #a=a+1
16                 j continue
17         case_1:
18                 sub $s2, $s2, $t1 #a=a-1
19                 j continue
20         case_2:
21                 add $s3, $s3, $s3 #b=2*b
22                 j continue
23 default:
24 continue:
```

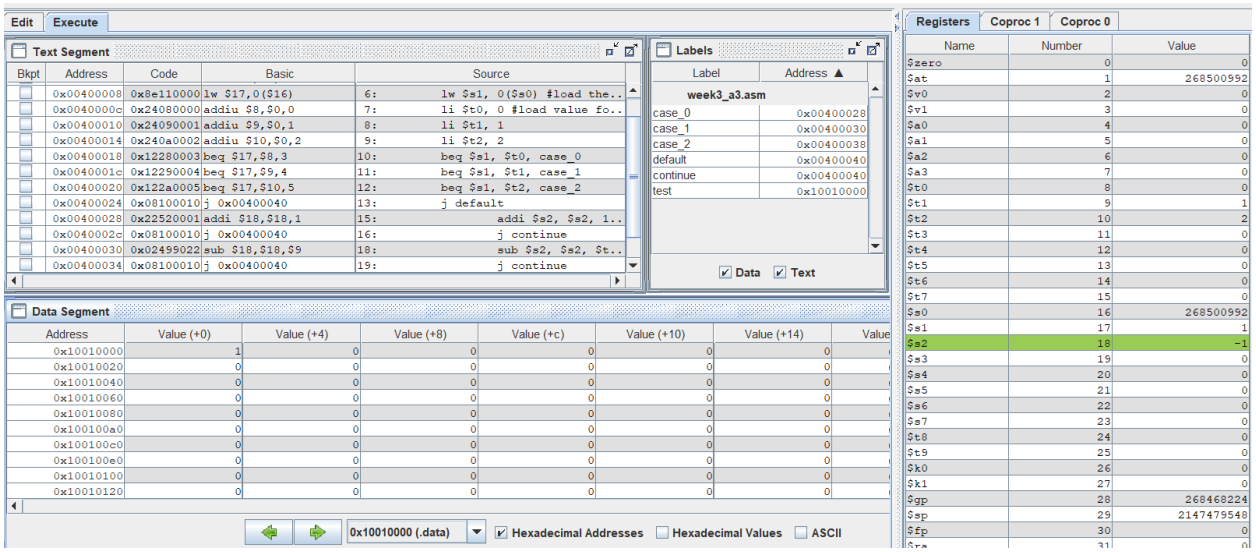
- **Test = 0 => $s1 =0 chạy vào case 0: a = a +1 (a=0+1=1)**



TH1 test = 1

```
#Laboratory Exercise 3, Home Assignment 3
.data
test: .word 1
.text
        la $s0, test #load the address of test variable
        lw $s1, 0($s0) #load the value of test to register $t1
        li $t0, 0 #load value for test case
        li $t1, 1
        li $t2, 2
        beq $s1, $t0, case_0
        beq $s1, $t1, case_1
        beq $s1, $t2, case_2
        j default
        case_0:
                addi $s2, $s2, 1 #a=a+1
                j continue
        case_1:
                sub $s2, $s2, $t1 #a=a-1
                j continue
        case_2:
                add $s3, $s3, $s3 #b=2*b
                j continue
default:
continue:
```

- **Test = 1 => $s1 = 1 chạy vào case 1: a= a -1 = 0 – 1= -1**

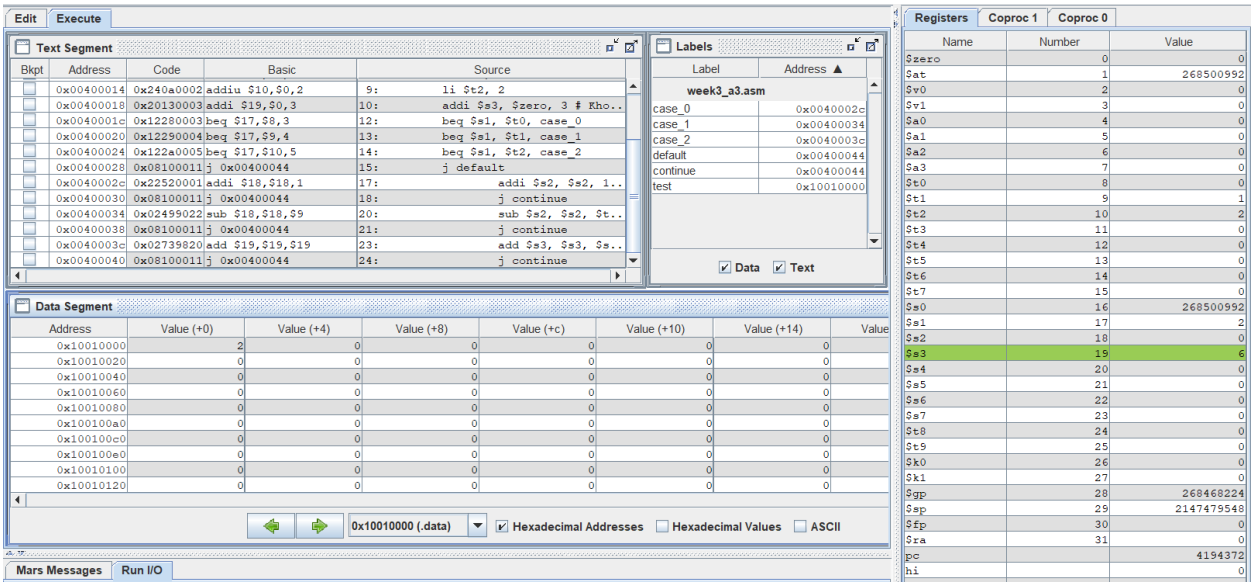


TH3: test = 2

week3_a3.asm*

```
1  #Laboratory Exercise 3, Home Assignment 3
2  .data
3  test: .word 2
4  .text
5          la $s0, test    #load the address of test variable
6          lw $s1, 0($s0)  #load the value of test to register $t1
7          li $t0, 0       #load value for test case
8          li $t1, 1
9          li $t2, 2
10         addi $s3, $zero, 3 # Khoi tao b = 3
11
12         beq $s1, $t0, case_0
13         beq $s1, $t1, case_1
14         beq $s1, $t2, case_2
15         j default
16         case_0:
17                 addi $s2, $s2, 1 #a=a+1
18                 j continue
19         case_1:
20                 sub $s2, $s2, $t1 #a=a-1
21                 j continue
22         case_2:
23                 add $s3, $s3, $s3 #b=2*b
24                 j continue
25 default:
26 continue:
```

- **Test = 2 => $s1 = 2 chạy vào case 1: b=2*b = 2*3 = 6**



Assignment 4:

a) **i < j**

week3_a4a.asm

```
1   #Laboratory Exercise 3, Home Assignment 1
2
3   .text
4           addi $s1, $zero 2 # i = 1
5           addi $s2, $zero 2 # j = 1
6
7           addi $t1, $zero 2 # x = 2
8           addi $t2, $zero 3 # y = 3
9           addi $t3, $zero 4 # z = 4
10  start:
11          bge  $s1,$s2,else # i >= j
12          addi $t1,$t1,1 # then part: x=x+1
13          addi $t3,$zero,1 # z=1
14          j endif # skip "else" part
15  else:
16          addi $t2,$t2,-1 # begin else part: y=y-1
17          add $t3,$t3,$t3 # z=2*z
18  endif:
19
```

b) **i>=j**

```
#Laboratory Exercise 3, Home Assignment 1

.text
        addi $s1, $zero 1 # i = 1
        addi $s2, $zero 5 # j = 2

        addi $t1, $zero 2 # x = 2
        addi $t2, $zero 3 # y = 3
        addi $t3, $zero 4 # z = 4
start:
        slt $t0,$s1,$s2 # i<j
        bne $t0,$zero,else # branch to else if j<i
        addi $t1,$t1,1 # then part: x=x+1
        addi $t3,$zero,1 # z=1
        j endif # skip "else" part
else:
        addi $t2,$t2,-1 # begin else part: y=y-1
        add $t3,$t3,$t3 # z=2*z
endif:
```

c) i+j<=0

Edit | Execute

week3_a4a.asm

```
1   #Laboratory Exercise 3, Home Assignment 1
2
3   .text
4           addi $s1, $zero 1 # i = 1
5           addi $s2, $zero 5 # j = 2
6
7           addi $t1, $zero 2 # x = 2
8           addi $t2, $zero 3 # y = 3
9           addi $t3, $zero 4 # z = 4
10  start:
11          add $t0, $s1, $s2 # $t0 = i +j
12          bgtz $t0, else    # $t0 > 0 jumb to else
13          addi $t1,$t1,1 #then part: x=x+1
14          addi $t3,$zero,1 # z=1
15          j endif # skip "else" part
16  else:
17          addi $t2,$t2,-1 # begin else part: y=y-1
18          add $t3,$t3,$t3 # z=2*z
19  endif:
20
```

d) i+j>m+n

```
#Laboratory Exercise 3, Home Assignment 1

.text
        addi $s1, $zero 1 # i = 1
        addi $s2, $zero 5 # j = 5
        addi $s3, $zero 1 # m = 1
        addi $s4, $zero -1 # n = -1

        addi $t1, $zero 2 # x = 2
        addi $t2, $zero 3 # y = 3
        addi $t3, $zero 4 # z = 4
start:
        add $t0, $s1, $s2 # $t0 = i +j
        add $t4, $s3, $s4 # $t4 = m +n
        ble $t0, $t4, else        # $t0 > 0 jumb to else
        addi $t1,$t1,1 #then part: x=x+1
        addi $t3,$zero,1 # z=1
        j endif # skip "else" part
else:
        addi $t2,$t2,-1 # begin else part: y=y-1
        add $t3,$t3,$t3 # z=2*z
endif:
```

Assignment 5:

a) **i < n:**

```
#Laboratory 3, Home Assigment 2
.data
A: .word  4,1,5,10,5
.text
        addi $s5, $zero, 0 # sum = 0
        addi $s1, $zero, 0 # i = 0

        la   $s2, A     # gan dia chi dau mang A cho $s2
        li   $s3, 5     # n = 5
        li   $s4, 1     # step = 1

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:
```

b) **I <= n:**

week3_a5a.asm

```
#Laboratory 3, Home Assigment 2
.data
A: .word  4,1,5,10,5,5
.text
        addi $s5, $zero, 0 # sum = 0
        addi $s1, $zero, 0 # i = 0

        la   $s2, A     # gan dia chi dau mang A cho $s2
        li   $s3, 5     # n = 5
        li   $s4, 1     # step = 1

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:

```

c) **sum >=0**

```
#Laboratory 3, Home Assigment 2
.data
A: .word  4,1,5,-15,5,5
.text
        addi $s5, $zero, 0 # sum = 0
        addi $s1, $zero, 0 # i = 0

        la   $s2, A     # gan dia chi dau mang A cho $s2
        li   $s3, 5     # n = 5
        li   $s4, 1     # step = 1

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

        bltz $s5, endloop # sum < 0 branch endloop
        j loop # goto loop
endloop:
```

d) A[i] == 0

```
#Laboratory 3, Home Assigment 2
.data
A: .word  0,0,0,3,5
.text
        addi $s5, $zero, 0 # sum = 0
        addi $s1, $zero, 0 # i = 0
        la   $s2, A     # gan dia chi dau mang A cho $s2
        li   $s3, 5     # n = 5
        li   $s4, 1     # step = 1

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

        beq $t0, $zero, endloop # A[i] == 0 branch endloop
        add $s5, $s5, $t0 # sum = sum + A[i]
        add $s1, $s1, $s4 # i = i + step
        j loop # goto loop
endloop:
```

Assignment 6:

```
A: .word  1,0,0,-25,5
.text
        addi $s5, $zero, 0 # max = 0
        addi $s1, $zero, 0 # i = 0
        la   $s2, A     # gan dia chi dau mang A cho $s2
        li   $s3, 5     # n = 5
        li   $s4, 1     # step = 1

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
        bgtz $t0, if
        sub $t0, $zero, $t0

if:
        slt $t6,$s5,$t0  # $t6 = max < n? 1 : 0
        beq $t6, $zero, endif # $t6 = 0 branch endif
        add $s5, $t0, $zero # max = A[i]
endif:
        add $s1, $s1, $s4 # i = i + step
        j loop # goto loop
endloop:
```