# Laboratory Exercise 3

# Load/Store, Jump & Branch instructions

Trần Khánh Quỳnh – 20225762

#### 1. Assignment 1

*TH1*:  $i \le j$  ( i = 100; j = 200)

-Mã nguồn:

```
addi $s1, $zero, 100 #i
addi $s2, $zero, 200 #j
# $t1: x; $t2: y; $t3: z

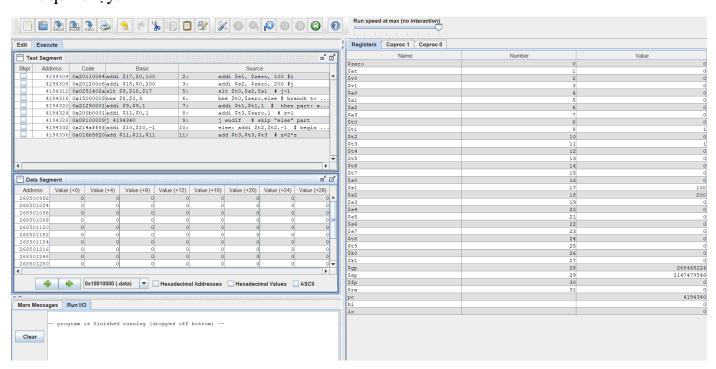
start:

slt $t0, $s2, $s1 #if j < i ($s2 < $s1)--> $t0 = 1; else $t0 = 0
bne $t0, $zero, else # t0 = 1 -> 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 # y = y-1
add $t3, $t3, $t3 #z = 2 * z

endif:
```

-Kết quả chạy:



-Thay đổi của các thanh ghi:

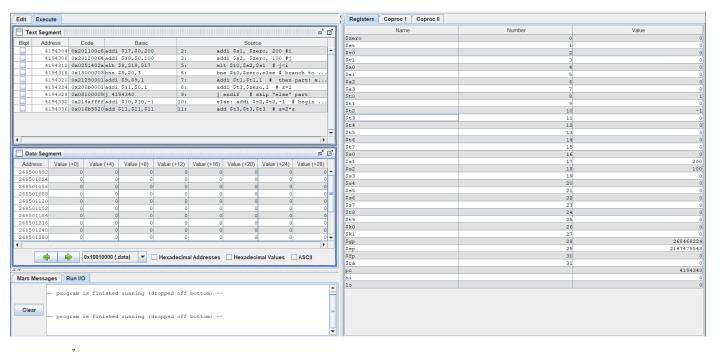
+Gán \$s1 là i có giá trị = 100; \$s2 là j có giá trị = 200

- +Đầu tiên, lệnh sl<br/>t \$t0, \$s2, \$s1 sẽ so sánh \$s1 và \$s2. Vì \$s2 > \$s1 hay j > i nên giá trị \$t0 vẫn là 0
- +Lệnh bne \$t0, \$zero, else sẽ kiểm tra nếu như \$t0 khác 0 thì sẽ nhảy đến nhánh else. Tuy nhiên trong trường hợp này \$t0 có giá trị là 0 nên sẽ thực hiện lệnh kế tiếp
- + addi \$t1, \$t1, 1 để thực hiện lệnh x = x+1, lúc này thanh ghi \$t1 sẽ chuyển thành 1
- + addi \$t3, \$0, 1 để thực hiện lệnh z =1, lúc này thanh ghi \$t3 cũng chuyển thành 1
- +j endif để bỏ qua nhánh else

#### *TH2:* i > j (i = 200; j = 100)

# -Mã nguồn:

```
addi $s1, $zero, 200 #i
 2
             addi $s2, $zero, 100 #j
 3
            # $t1: x; $t2: y; $t3: z
 4
 5
    start:
            slt $t0, $s2, $s1 #if j < i ($s2 < $s1) --> $t0 = 1; else $t0 = 0
 6
            bne $t0, $zero, else # t0 = 1 \rightarrow else
 7
            addi $t1, $t1, 1 #then part x = x+1
 8
 9
            addi $t3, $zero, 1 \#z = 1
            j endif #skip else part
10
11
12
            addi $t2, $t2, -1 \# y = y-1
            add $t3, $t3, $t3 \#z = 2 * z
13
    endif:
14
```



-Thay đổi của các thanh ghi

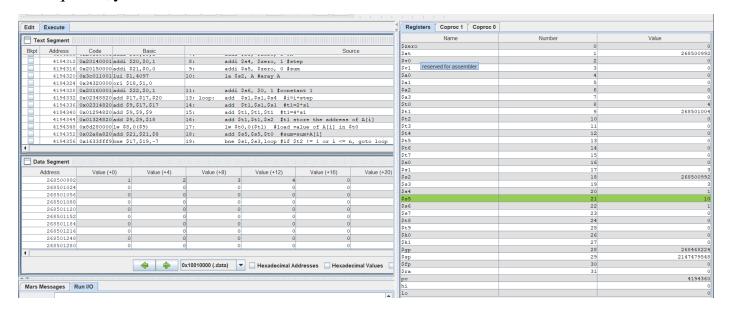
- +Gán \$s1 là i có giá trị = 200, \$s2 là j có giá trị là  $100 \rightarrow$  thanh ghi \$s1 và \$s2 thay đổi, \$s1 = 200 và \$s2 = 100
- +Lệnh slt so sánh \$s1 và \$s2, vì \$s2 < \$s1 hay j  $\le$  i nên \$t0 thay đổi thành 1
- +Lệnh bne \$t0, \$zero, else sẽ kiểm tra nếu như \$t0 khác 0 thì sẽ nhảy đến nhánh else.
- → Thực hiện các lệnh y = y-1 và z = 2 \* z
- (t2 hay y ban dau có giá trị bằng 0; gán y = y-1 nên <math>t2 chuyển thành -1;
- \$t3 hay z ban đầu có giá trị bằng 0, khi nhân 2 lên thì sẽ vẫn là 0)

#### 2. Assignment 2

TH1: Khởi tạo i = -1. Mã nguồn:

```
.data
            A: .word 1,2,3,4
 2
 3
   .text
            addi $s1, $zero, -1 #i
 4
            addi $s3, $zero, 3 #n
 5
            addi $s4, $zero, 1 #step
 6
 7
            addi $s5, $zero, 0 #sum
            la $s2, A #array A
 9
    loop:
            add $s1,$s1,$s4 #i=i+step
10
            add $t1,$s1,$s1 #t1=2*s1
11
12
            add $t1,$t1,$t1 #t1=4*s1
            add $t1,$t1,$s2 #t1 store the address of A[i]
13
            lw $t0,0($t1) #load value of A[i] in $t0
14
            add $s5,$s5,$t0 #sum=sum+A[i]
15
            bne $s1,$s3,loop #if i != n, goto loop
16
17
```

Biến i lưu ở \$s1, địa chỉ cơ sở mảng A lưu ở \$s2, n lưu ở \$s3, step lưu ở \$s4 và sum lưu ở \$s5.



-Sự thay đổi của thanh ghi:

+Đầu tiên các thanh ghi lưu các giá trị được gán vào:

Thanh ghi \$s1 = -1; thanh ghi \$s3 = 3; thanh ghi \$s4 = 1, thanh ghi \$s5 = 0

+Lệnh la \$s2, A lưu địa chỉ cơ sở của mảng A vào \$s2 → \$s2 = 268500992

Bkpt	Address	Code	Basic		Source
	0x00400000	0x2011ffff	addi \$17,\$0,-1	4:	addi \$s1, \$zero, -1 #i
	0x00400004	0x20130003	addi \$19,\$0,3	5:	addi \$s3, \$zero, 3 #n
	0x00400008	0x20140001	addi \$20,\$0,1	6:	addi \$s4, \$zero, 1 #step
	0x0040000c	0x20150000	addi \$21,\$0,0	7:	addi \$s5, \$zero, 0 #sum
	0x00400010	0x3c011001	lui \$1,4097	8:	la \$s2, A #array A
	0x00400014	0x34320000	ori \$18,\$1,0		
	0x00400018	0x02348820	add \$17,\$17,\$20	10:	add \$s1,\$s1,\$s4 #i=i+step
	0x0040001c	0x02314820	add \$9,\$17,\$17	11:	add \$t1,\$s1,\$s1 #t1=2*s1
	0x00400020	0x01294820	add \$9,\$9,\$9	12:	add \$t1,\$t1,\$t1  #t1=4*s1
	0x00400024	0x01324820	add \$9,\$9,\$18	13:	add \$t1,\$t1,\$s2 #t1 store the address of A
	0x00400028	0x8d280000	lw \$8,0(\$9)	14:	<pre>lw \$t0,0(\$t1) #load value of A[i] in \$t0</pre>
	0x0040002c	0x02a8a820	add \$21,\$21,\$8	15:	add \$s5,\$s5,\$t0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$∀0	2	0
\$∀1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	-1
\$s2	18	268500992
\$s3	19	3

# + Vòng lặp

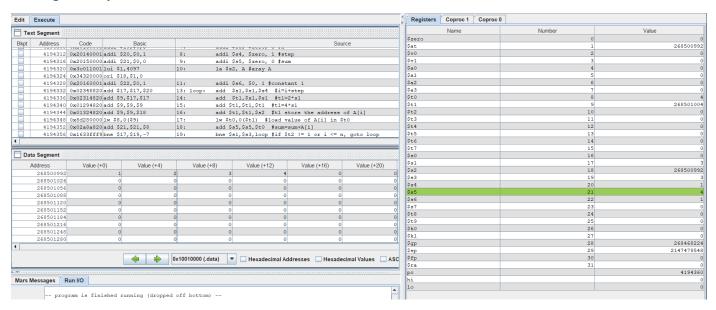
- Lệnh add \$s1, \$s1, \$s4 để tăng i dần lên, thực hiện vòng lặp. Với vòng lặp đầu tiên, \$s1 tăng lên 1, vì vậy \$s1 = 0
- 2 lệnh add tiếp theo để gán \$t1 thành độ dịch địa chỉ của mảng. Với vòng lặp đầu thì \$s1=0\*\$s1=0
- Lệnh add \$t1, \$t1, \$s2 để khiến \$t1 sẽ lưu địa chỉ của A[i] ta cần xác định. Với i = 0, tức là vẫn ở nguyên địa chỉ cơ sở nên lúc này \$t1 = 268500992
- Lệnh lw \$t0, 0(\$t1) nhằm lưu giá trị của A[i] vào \$t0. Vì thế nên \$t0 lúc này chuyển thành 1 (Mảng A =  $\{1,2,3,4\} \rightarrow A[0] = 1 \rightarrow $t0 = 1$ )
- Lệnh add sau đó để thực hiện sum = sum + A[i]; thanh ghi \$s5 chuyển thành

$$(0 + A[0] = 0 + 1 = 1 \rightarrow \$s5 = 1)$$

 Do \$s1 có giá trị = 0; \$s3 có giá trị bằng 3, hai giá trị không bằng nhau cho nên sẽ thực hiện lặp lại vòng lặp cho đến khi \$s1 = \$s3

# TH2: Thay đổi giá trị i, i = 2.

-Kết quả chạy:



- +Lệnh add đầu tiên khiến i thành  $i = 3 \Rightarrow \$s1 = 3$
- +Đối với lệnh add thứ 4, địa chỉ \$t1 trỏ đến chuyển thành 268501004, tức đã tăng 12 đơn vị so với địa chỉ của A[0]. Nguyên nhân là do lúc này \$t1 đang lưu địa chỉ của A[3]
- +Lênh lw để lưu giá tri mà \$t1 trỏ tới vào \$t0. → \$t1 trỏ đến A[3] nên \$t0 = 4
- +\$s5 hay sum khởi tạo có giá trị  $= 0 \rightarrow$  Thanh ghi \$s5 chuyển giá trị thành 4. \$s1 và \$s3 lúc này đã bằng nhau nên chương trình kết thúc, không còn lặp lại.

### 3. Assignment 3

TH1: test: .word 1

- -Lệnh la \$s0, test: Lưu địa chỉ của test vào \$s0 → \$s0 = 268500992
- -Lệnh lw \$s1, 0(\$s0): Lưu giá trị của địa chỉ lưu trong \$s0 vào  $\$s1 \rightarrow \$s1 = 1$
- -Các lệnh li \$t0, 0 li \$t1,1 và li \$t2, 2  $\rightarrow$  \$t0 = 0; \$t1 = 1; \$t2 = 2
- -Lệnh beq \$s1, \$t0, case\_0: Kiểm tra xem giá trị \$s1 có bằng với giá trị \$t0 không. Tuy nhiên ở trường hợp này \$s1 = 1 → chương trình tiếp tục chạy
- -Lệnh beq \$s1, \$t1, case\_1. Do đúng nên chương trình thực hiện case\_1
- -Lệnh sub \$s2, \$s2, \$t1 để thực hiện phép gán a-1 vào a  $\rightarrow$  \$s2 = -1

-Lệnh j continue: nhảy xuống continue → kết thúc chương trình

#### TH2: test: .word 2

- -Các lệnh từ la đến hết lệnh beq \$s1, \$t0, case\_0 thực hiện tương tự như trường hợp 1
- -Đối với lệnh beq \$s1, \$t1, case\_1, vì không đúng do lúc này \$s1 = 2 → chương trình tiếp tục chạy
- -Lệnh beq \$s1, \$t2, case\_2 → đúng → nhảy xuống thực hiện case\_2
- -Lệnh add \$s3, \$s3, \$s3 thực hiện phép gán 2 \* b vào b. Tuy nhiên do \$s3 ban đầu bằng  $0 \rightarrow $s3 = 0$
- -j continue → chương trình kết thúc

#### TH3: test: .word 0

- -Các lệnh từ la \$s0, test cho đến li \$t2, 2 thực hiện tương tự như trường hợp 1
- -Đối với lệnh beq \$s1, \$t0, case\_0 → trường hợp này \$s1 = 0, bằng với \$t0
- → chương trình thực hiện case\_0
- -addi \$s2,\$s2,1 #a=a+1 → Giá trị của \$s2 = 1
- j continue → Xuống continue và kết thúc chương trình

#### TH4: test: .word 3

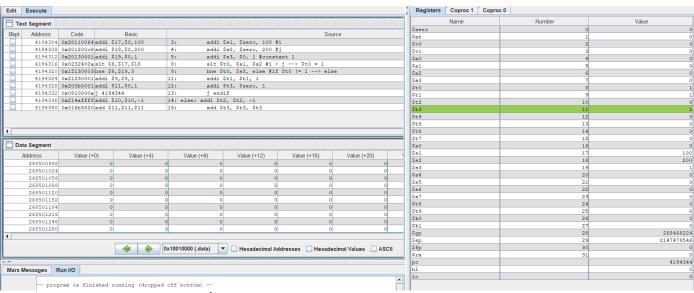
- -Các lệnh từ la \$s0, test cho đến li \$t2, 2 thực hiện tương tự như trường hợp 1
- -Lệnh beq \$s1, \$t0, case\_0 → không đúng, chương trình tiếp tục
- -Lệnh beq \$s1, \$t1, case\_1 → không đúng, chương trình tiếp tục
- -Lệnh beq \$s1, \$t2, case 2 → Không đúng, chương trình tiếp tục
- -j default → nhảy xuống trường hợp default và kết thúc chương trình

# 4. Assignment 4

a. 
$$i < j$$
 ( $i = 100$ ;  $j = 200$ )

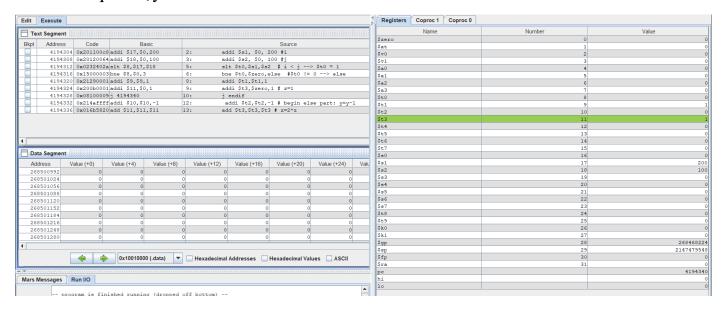
-Mã nguồn:

```
# i < j
 2
    .text
            addi $s1, $zero, 100 #i
 3
            addi $s2, $zero, 200 #j
 4
            addi $s3, $0, 1 #constant 1
 5
 6
 7
    start:
            slt $t0, $s1, $s2 \#i < j --> $t0 = 1
 8
            bne $t0, $s3, else #if $t0 != 1 --> else
 9
            #i < j
10
            addi $t1, $t1, 1
11
            addi $t3, $zero, 1
12
            j endif
13
    else: addi $t2, $t2, -1
14
            add $t3, $t3, $t3
16
    endif:
17
```



Vì i < j thỏa mãn → chạy tiếp chương trình → \$t1 = 1\$ và \$t3 = 1\$

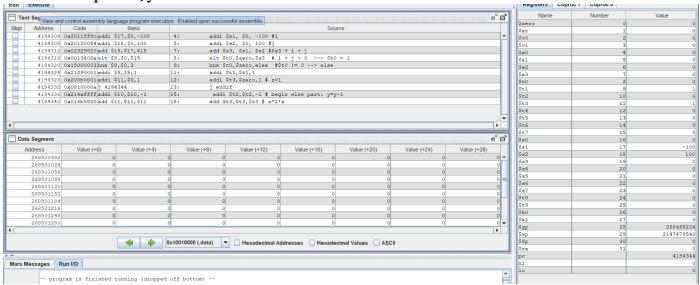
```
mips1.asm
     text
            addi $s1, $0, 200 #i
 2
            addi $s2, $0, 100 #j
 3
 4
    start:
            slt $t0,$s1,$s2 # i < j --> $t0 = 1
 5
            bne $t0,$zero,else #$t0 != 0 --> else
 6
            #i >= j
 7
            addi $t1,$t1,1
 8
            addi $t3,$zero,1 # z=1
 9
10
            j endif
11
    else: #i < j
             addi $t2,$t2,-1 # begin else part: y=y-1
12
13
            add $t3,$t3,$t3 # z=2*z
14
    endif:
```



+i=200;  $j=100 \Rightarrow i>= j$  thỏa mãn nên tiếp tục chạy các lệnh tiếp theo mà không nhảy đến nhánh else

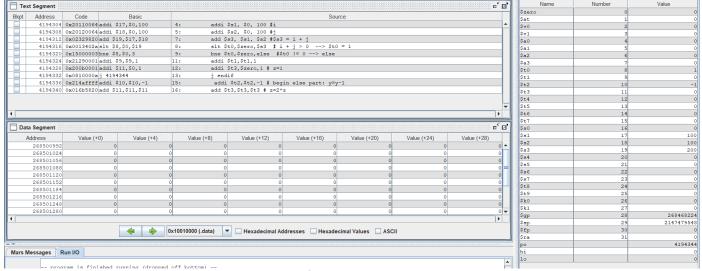
c. 
$$i + j \le 0$$
  
 $TH1: i + j = 0 \ (i = -100; j = 100)$   
-Mã nguồn:

```
#i + j <= 0
.text
       addi $s1, $0, -100 #i
       addi $s2, $0, 100 #j
start:
        add $s3, $s1, $s2 \#$s3 = i + j
        slt $t0,$zero,$s3 # i + j > 0 --> $t0 = 1
       bne $t0,$zero,else #$t0 != 0 --> else
        #i + j <= 0
        addi $t1,$t1,1
        addi $t3,$zero,1 # z=1
       j endif
else: \# i + j > 0
         addi $t2,$t2,-1 # begin else part: y=y-1
        add $t3,$t3,$t3 # z=2*z
endif:
```



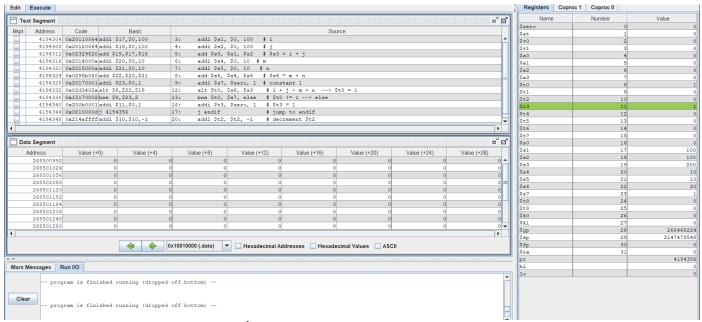
Vì i + j = 0 tức i + j <= 0 thỏa mãn → tiếp tục chạy tiếp chương trình mà không chuyển sang nhánh else → t = 1 và t = 1 và t = 1

 $TH2: i + j > 0 \ (i = 100; j = 100)$ -Kết quả chạy:



Vì không thỏa mãn  $i + j \le 100$  → chuyển sang nhánh else → \$t2 = -1; \$t3 = 0 d. i + j > m + n TH1: i + j > m + n

```
1
     \# i + j > m + n
 2
    .text
         addi $s1, $0, 100
 3
         addi $s2, $0, 100
 4
                            # j
         add $s3, $s1, $s2
                              \# \$s3 = i + j
 5
         addi $s4, $0, 10 # m
 6
         addi $s5, $0, 10
 7
 8
         add $s6, $s4, $s5
                             \# \$s6 = m + n
         addi $s7, $zero, 1 # constant 1
 9
10
11
     start:
         slt $t0, $s6, $s3
                              \# i + j > m + n --> $t0 = 1
12
13
         bne $t0, $s7, else
                               # $t0 != 1 --> else
14
         \# i + j > m + n
15
         addi $t3, $zero, 1
                               # $t3 = 1
16
         j endif
                               # jump to endif
17
18
     else: \# i + j \le m + n
19
         addi $t2, $t2, -1
20
                               # decrement $t2
         add $t3, $t3, $t3
                               # $t3 = 2 * $t3
21
22
23 endif:
```



Vì i + j > m + n nên thực hiện tiếp đoạn chương trình ở sau mà không sang nhánh else  $\rightarrow$  \$t1 = 1; \$t3 = 1

TH2:  $i + j \le m + n$ Đặt i + j = m + n = 300-Kết quả chay:

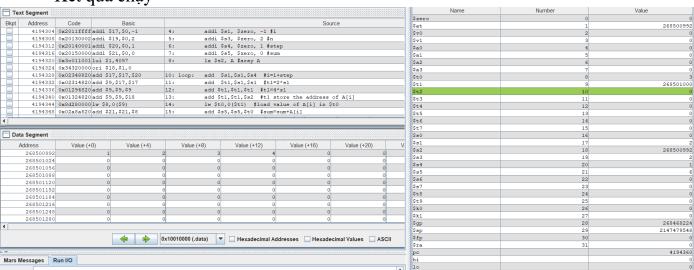
Text Segment								o c		Number	Value
kpt Address	Code	Basic		Source					Szero Sat	0	
4104300	0x20120064 addi \$	10.00.100		o. man vol, vo, 100 , 1					▲ Sv0	2	
	0x20120064 add1 \$			4: addi \$s2, \$0, 100 # j					Sv1	2	
	0x20140064 addi \$								\$a0	4	
	0x20140064 addi \$								\$a1	5	
	0x0295b020 add \$2								Sa2	6	
	0x20170001 addi \$			s7, \$zero, 1 # cons					\$a3	7	
	0x02d3402aslt \$8			0, \$s6, \$s3 # i +		0 = 1			St0	8	
	0x15170002 bne \$8			0, Ss7, else # \$t0		0 - 1			St1	9	
	0x200b0001 addi \$			t3, \$zero, 1 # \$t3					\$t2	10	
	0x0810000d j 4194		17: j endi:						St3	11	
	0x214affff addi \$			t2, \$t2, -1 # dec					St4	12	
	0x016b5820 add \$1			3, \$t3, \$t3 # \$t3					▼ \$t5	13	
1171000	011020000000000000000000000000000000000	,,	D21 000 70	0, 100, 100 1 100						14	
									St7	15	
Data Segment								o <sup>r</sup>	☑ Sa0	16	
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)	Şs1	17	
268500992		Value (+4)	value (+o)	value (+12)	value (+10)	value (+20)	Value (+24)		\$s2	18	
266500992			0	0	0	U					
0.000.000.004									\$=3	19	
268501024			0	0 0	0	0	C		\$53 \$54	19	
268501056	0		0	0 0	0	0	0	) (	\$s4 \$s5	20 21	
268501056 268501088	0		0	0 0 0	0	0	0	) ()	\$84	20	
268501056 268501088 268501120	0 0		0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0 0 0	0 0 0		\$s4 \$s5	20 21	
268501056 268501088 268501120 268501152	0 0 0		0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0 0	C C		\$s4 \$s5 \$s6	20 21 22	
268501056 268501088 268501120 268501152 268501184	0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0		\$s4 \$s5 \$s6 \$s7 \$t8 \$t9	20 21 22 22 23	
268501056 268501088 268501120 268501152 268501184 268501216	0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )		\$s4 \$s5 \$s6 \$s7 \$t8	20 21 22 23 24	
268501056 268501088 268501120 268501152 268501184 268501216 268501248	0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	C C C C C C C C C C C C C C C C C C C		\$ \$ \$ 4 \$ \$ 5 5 \$ 5 6 \$ 5 7 \$ 5 8 \$ 5 9 \$ 5 0 \$ 5 1 \$ 5 1 \$ 5 1 \$ 5 1 \$ 1 \$ 1 \$ 1 \$ 1	20 21 22 23 24 25	
268501056 268501088 268501120 268501152 268501184 268501216	0 0 0 0 0		0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	C C C C C C C C C C C C C C C C C C C		\$54 \$55 \$56 \$57 \$58 \$58 \$57 \$58 \$59 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50	20 21 22 23 24 25 26	
268501056 268501088 268501120 268501152 268501184 268501216 268501248	0 0 0 0 0		0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	C C C C C C C C C C C C C C C C C C C		\$54 \$55 \$56 \$57 \$58 \$58 \$57 \$58 \$59 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50	20 21 22 23 24 25 26 27	268468
268501056 268501088 268501120 268501152 268501184 268501216 268501248	0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		\$34 \$35 \$36 \$37 \$t8 \$t9 \$k0 \$k1 \$gp	20 21 22 23 24 25 26 27 28	268468
268501056 268501088 268501120 268501152 268501184 268501216 268501248	0 0 0 0 0	• •	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	C C C C C C C C C C C C C C C C C C C		\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	20 21 22 23 24 25 26 27 28 29	268468
268501056 268501088 268501120 268501152 268501184 268501216 268501248	0 0 0 0 0	• •	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 ddresses Hexadec	0 0 0 0 0 0 0 0 0 0			Ss4 Ss5 Ss6 Ss6 Ss7 St8 St9 Sk0 Sk1 Sgp Ssp Ssp Ssp	20 21 22 23 24 25 26 27 28 29	268468 2147475
268501056 268501088 268501120 268501152 268501184 268501216 268501218 268501248	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<b>→</b> •	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0			584 535 536 537 548 549 540 540 541 541 542 542 543 544 544 544 544 544 544 544 544 544	20 21 22 23 24 25 26 27 28 29	
268501056 268501088 268501120 268501152 268501184 268501216 268501248	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• •	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	o o o o o o o o o o o o o o o o o o o	C C C C C C C C C C C C C C C C C C C		594 595 596 597 5t8 5t9 5t9 5t9 5t1 5gp 5gp 5ra 9cp	20 21 22 23 24 25 26 27 28 29	268468 2147479

Vì i+j>m+n không thỏa mãn nên chuyển sang nhánh else  $\rightarrow$  \$t2 = -1; \$t3 = 0

### 5. Assignment 5

a. i < n
-Mã nguồn

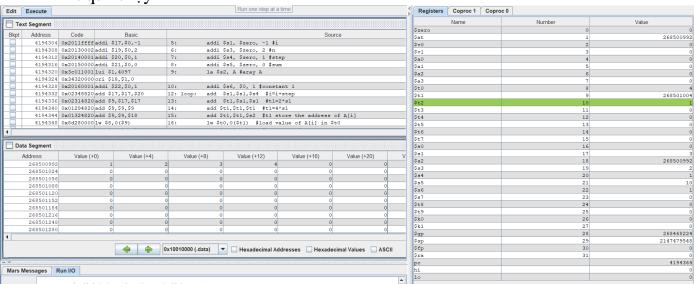
```
.data
2
           A: .word 1, 2, 3, 4
3
    .text
4
           addi $s1, $zero, -1 #i
            addi $s3, $zero, 2 #n
5
6
            addi $s4, $zero, 1 #step
           addi $s5, $zero, 0 #sum
7
           la $s2, A #aray A
8
9
           add $s1,$s1,$s4 #i=i+step
10
   loop:
           add $t1,$s1,$s1 #t1=2*s1
11
           add $t1,$t1,$t1 #t1=4*s1
12
           add $t1,$t1,$s2 #t1 store the address of A[i]
13
           lw $t0,0($t1) #load value of A[i] in $t0
14
           add $s5,$s5,$t0 #sum=sum+A[i]
15
16
           slt t^2, s^3 #compare i with n. If i < n --> t^2 = 1
           bne t2, zero, loop #if t2 = 1 or i < n, goto loop
17
18
```



Vì i < n thì sẽ tiếp tục vòng lặp → i = n sẽ kết thúc vòng lặp. Với n = 2→ sum = = 6 → \$s5 = 6

b. i <= n -Mã nguồn

```
1
    #i <= n
    .data
 2
            A: .word 1, 2, 3, 4
 3
 4
    .text
            addi $s1, $zero, -1 #i
 5
            addi $s3, $zero, 2 #n
 6
 7
            addi $s4, $zero, 1 #step
            addi $s5, $zero, 0 #sum
 8
 9
            la $s2, A #aray A
            addi $s6, $0, 1 #constant 1
10
11
            add $s1,$s1,$s4 #i=i+step
12
    loop:
            add $t1,$s1,$s1 #t1=2*s1
13
            add $t1,$t1,$t1 #t1=4*s1
14
            add $t1,$t1,$s2 #t1 store the address of A[i]
15
            lw $t0,0($t1) #load value of A[i] in $t0
16
            add $s5,$s5,$t0 #sum=sum+A[i]
17
            slt $t2, $s3, $s1 #compare i with n. If i > n--> $t2 = 1
18
            bne $t2,$s6,loop #if $t2 != 1 or i <= n, goto loop
19
20
```



Do điều kiện là  $i \le n$  nên cho đến khi i > n thì vòng lặp mới kết thúc  $\rightarrow$  sum = 10  $\rightarrow$  \$s5 = 10

c. sum >= 0-Mã nguồn

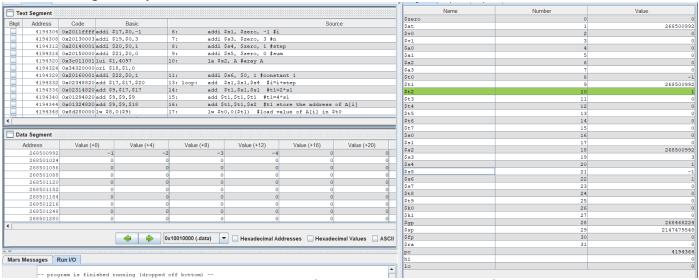
```
addi $s1, $zero, 100 #i
    addi $s2, $zero, 200 #j
    # $t1: x; $t2: y; $t3: z

start:

    slt $t0, $s2, $s1 #if j < i ($s2 < $s1)--> $t0 = 1; else $t0 = 0
    bne $t0, $zero, else # t0 = 1 -> 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 # y = y-1
    add $t3, $t3, $t3 #z = 2 * z

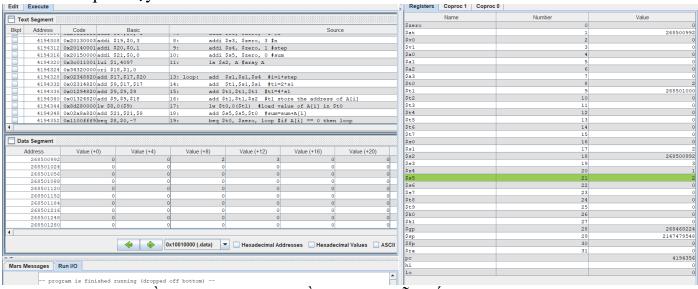
endif:
```



Khi  $i = 0 \rightarrow sum = -1$ ; không thỏa mãn điều kiện sum >= 0 nên kết thúc vòng lặp

d. A[i] == 0-Mã nguồn

```
#sum >= 0
.data
       A: .word 0, 0, 2, 3
text
       addi $s1, $zero, -1 #i
       addi $s3, $zero, 3 #n
       addi $s4, $zero, 1 #step
       addi $s5, $zero, 0 #sum
       la $s2, A #aray A
       add $s1,$s1,$s4 #i=i+step
100p:
       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]
       beq $t0, $zero, loop #if A[i] == 0 then loop
```



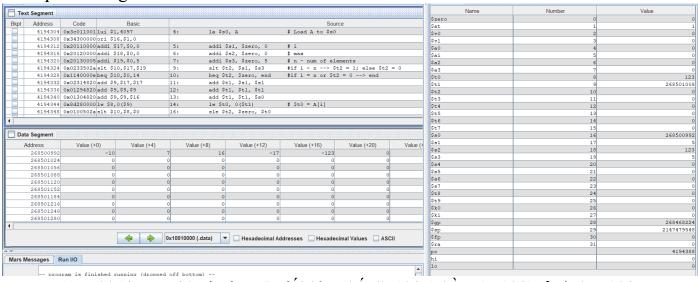
 $A = \{0, 0, 2, 3\}$ . Phần tử A[0] và A[1] đều là 0 nên vẫn tiếp tục thực hiện vòng lặp. Chương trình dừng lại khi gặp phần tử  $A[2] = 2 \rightarrow$  thoát khỏi vòng lặp

# 6. Assignment 6

-Mã nguồn:

```
.data
              .word -10, 7, 16, -17, -123
text
       la $s0, A
                              # Load A to $s0
       addi $s1, $zero, O
                              # i
       addi $s2, $zero, O
       addi $s3, $zero, 5
                              # n - num of elements
loop:
       slt $t2, $s1, $s3
                              #if i < n --> $t2 = 1; else $t2 = 0
       beq $t2, $zero, end
                              #if i = n or $t2 = 0 --> end
       add $t1, $s1, $s1
       add $t1, $t1, $t1
       add $t1, $t1, $s0
       lw $t0, 0($t1)
                              # $t0 = A[i]
ifsmaller:
       sle $t2, $zero, $t0
       bne $t2, $zero, ifgreat
                              \# $t0 = 0 - $t0
       sub $t0, $zero, $t0
ifgreat:
       slt $t2, $t0, $s2
                              bne $t2, $zero, endif
       add $s2, $t0, $zero
                                      \# \max = A[i]
endif:
       addi $s1, $s1, 1
                                      # 1++
       j loop
end:
```

### -Kết quả chương trình:



Trong mảng khởi tạo, giá trị trị tuyệt đối lớn nhất là 123 (phần tử -123) → \$s2 = 123