

Laboratory Exercise 3

Load/Store, Jump & Branch instructions

Trần Khánh Quỳnh – 20225762

1. Assignment 1

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

-Mã nguồn:

```
.text
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:

The screenshot shows the Mars MIPS simulator interface. The **Text Segment** window displays the assembly code with addresses from 4194304 to 4194336. The **Data Segment** window shows memory addresses from 268500992 to 268501280, all containing 0. The **Registers** window shows the state of registers: \$s1 is 100, \$s2 is 200, and \$t1 is 1. The **Console** window shows the message "-- program is finished running (dropped off bottom) --".

-Thay đổi của các thanh ghi:

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

+Đầu tiên, lệnh `slt $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:

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

-Kết quả chạy:

The screenshot displays the Mars MIPS simulator interface. The 'Text Segment' window shows the assembly code being executed, with line numbers 1 to 14. The 'Data Segment' window shows memory addresses from 268500992 to 268501280. The 'Registers' window shows the state of registers \$zero through \$lo, with \$s1 at 200 and \$s2 at 100. The 'Mars Messages' window shows the program finished running.

-Thay đổi của các thanh ghi

+Gán \$s1 là i có giá trị = 200, \$s2 là j có giá trị là 100 → thanh ghi \$s1 và \$s2 thay đổi, \$s1 = 200 và \$s2 = 100

+Lệnh slt so sánh \$s1 và \$s2, vì \$s2 < \$s1 hay j < 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 đầu có giá trị bằng 0; gán y = y-1 nên \$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:

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

-Kết quả chạy:

Registers					
Coproces		Coproces			
Name		Number		Value	
\$zero		0		0	
\$at		1		268500992	
\$v0		2		0	
\$v1		3		0	
\$a0		4		0	
\$a1		5		0	
\$a2		6		0	
\$a3		7		0	
\$t0		8		0	
\$t1		9		268501004	
\$t2		10		0	
\$t3		11		0	
\$t4		12		0	
\$t5		13		0	
\$t6		14		0	
\$t7		15		0	
\$s0		16		0	
\$s1		17		268500992	
\$s2		18		268500992	
\$s3		19		3	
\$s4		20		1	
\$s5		21		10	
\$s6		22		1	
\$s7		23		0	
\$s8		24		0	
\$s9		25		0	
\$k0		26		0	
\$k1		27		0	
\$gp		28		268468224	
\$sp		29		2147475548	
\$fp		30		0	
\$ra		31		0	
\$pc				4194360	
\$hi				0	
\$lo				0	

Data Segment						
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)
268500992	1	2	3	4	0	0
268501024	0	0	0	0	0	0
268501056	0	0	0	0	0	0
268501088	0	0	0	0	0	0
268501120	0	0	0	0	0	0
268501152	0	0	0	0	0	0
268501184	0	0	0	0	0	0
268501216	0	0	0	0	0	0
268501248	0	0	0	0	0	0
268501280	0	0	0	0	0	0

-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: lw \$t0,0(\$t1) #load value of A[i] in \$t0
	0x0040002c	0x02a8a820	add \$21,\$21,\$8	15: add \$s5,\$s5,\$t0 #sum=sum+A[i]

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	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} → A[0] = 1 → \$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:

The screenshot shows a MIPS simulator interface. The 'Text Segment' tab is active, displaying assembly code. The 'Data Segment' tab shows memory addresses and values. The 'Registers' tab shows the state of MIPS registers.

Bkpt	Address	Code	Basic	Source
4194312	0x20140001	addi \$20,\$0,1		
4194316	0x20150000	addi \$21,\$0,0		
4194320	0x3c011001	lui \$1,4097		
4194324	0x34320000	ori \$18,\$1,0		
4194328	0x20160001	addi \$22,\$0,1		
4194332	0x02340820	add \$17,\$17,\$20		
4194336	0x02314820	add \$9,\$17,\$17		
4194340	0x01294820	add \$9,\$9,\$9		
4194344	0x01324820	add \$9,\$9,\$10		
4194348	0x8d200000	lw \$t0,0(\$t1)		
4194352	0x02a8a820	add \$21,\$21,\$8		
4194356	0x1633ffff	bne \$17,\$19,-7		

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)
268500992	1	2	3	4	0	0
268501024	0	0	0	0	0	0
268501056	0	0	0	0	0	0
268501088	0	0	0	0	0	0
268501120	0	0	0	0	0	0
268501152	0	0	0	0	0	0
268501184	0	0	0	0	0	0
268501216	0	0	0	0	0	0
268501248	0	0	0	0	0	0
268501280	0	0	0	0	0	0

Register	Name	Number	Value
\$zero		0	0
\$at		1	268500992
\$v0		2	0
\$v1		3	0
\$a0		4	0
\$a1		5	0
\$a2		6	0
\$a3		7	0
\$t0		8	4
\$t1		9	268501004
\$t2		10	0
\$t3		11	0
\$t4		12	0
\$t5		13	0
\$t6		14	0
\$t7		15	0
\$a0		16	0
\$a1		17	3
\$a2		18	268500992
\$a3		19	3
\$a4		20	1
\$a5		21	4
\$a6		22	1
\$a7		23	0
\$t8		24	0
\$t9		25	0
\$t0		26	0
\$t1		27	0
\$gp		28	268468224
\$fp		29	2147479548
\$sp		30	0
\$ra		31	0
\$pc			4194360
\$hi			0
\$lo			0

+Lệnh add đầu tiên khiến i thành $i = 3 \rightarrow \$s1 = 3$

+Đối với lệnh add thứ 4, địa chỉ $\$t1$ trở nên 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 $A[3]$

+Lệnh lw để lưu giá trị mà $\$t1$ trở tới vào $\$t0$. $\rightarrow \$t1$ trở 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 $\$s0 \rightarrow \$s0 = 268500992$

-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 $\$s1 = 1 \rightarrow$ 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 → $s3 = 0$

-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:

```

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

```

-Kết quả chạy:

Execute

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	4194304	0x20110064	addi \$17,\$0,100	3: addi \$s1, \$zero, 100 #i
<input type="checkbox"/>	4194308	0x201200c8	addi \$18,\$0,200	4: addi \$s2, \$zero, 200 #j
<input type="checkbox"/>	4194312	0x20130001	addi \$19,\$0,1	5: addi \$s3, \$0, 1 #constant 1
<input type="checkbox"/>	4194316	0x2023402a	slt \$0,\$17,\$18	8: slt \$t0, \$s1, \$s2 #i < j --> \$t0 = 1
<input type="checkbox"/>	4194320	0x15130003	bne \$8,\$19,3	9: bne \$t0, \$s3, else #if \$t0 != 1 --> else
<input type="checkbox"/>	4194324	0x21290001	addi \$9,\$9,1	11: addi \$t1, \$t1, 1
<input type="checkbox"/>	4194328	0x200b0001	addi \$11,\$0,1	12: addi \$t3, \$zero, 1
<input type="checkbox"/>	4194332	0x0810000a	j 4194344	13: j endif
<input type="checkbox"/>	4194336	0x214affff	addi \$10,\$10,-1	14: else: addi \$t2, \$t2, -1
<input type="checkbox"/>	4194340	0x01b5820	add \$11,\$11,\$11	15: add \$t3, \$t3, \$t3

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)
268500992	0	0	0	0	0	0
268501024	0	0	0	0	0	0
268501056	0	0	0	0	0	0
268501088	0	0	0	0	0	0
268501120	0	0	0	0	0	0
268501152	0	0	0	0	0	0
268501184	0	0	0	0	0	0
268501216	0	0	0	0	0	0
268501248	0	0	0	0	0	0
268501280	0	0	0	0	0	0

0x10010000 (.data)

Hexadecimal Addresses

Hexadecimal Values

ASCII

Registers

Coproc 1

Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	1
\$t1	9	1
\$t2	10	0
\$t3	11	1
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	100
\$a2	18	200
\$a3	19	1
\$a4	20	0
\$a5	21	0
\$a6	22	0
\$a7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194344
hi		0
lo		0

Mars Messages

Run I/O

-- program is finished running (dropped off bottom) --

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

b. $i \geq j$ ($i = 200$; $j = 100$)

-Mã nguồn:

```

mips1.asm

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

```

-Kết quả chạy:

Edit

Execute

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	4194304	0x201100e8	addi \$17,\$0,200	2: addi \$s1, \$0, 200 #i
<input type="checkbox"/>	4194308	0x201200e4	addi \$18,\$0,100	3: addi \$s2, \$0, 100 #j
<input type="checkbox"/>	4194312	0x0232402a	slt \$8,\$11,\$18	5: slt \$t0,\$s1,\$s2 # i < j --> \$t0 = 1
<input type="checkbox"/>	4194316	0x15000003	bne \$9,\$0,3	6: bne \$t0,\$zero,else # \$t0 != 0 --> else
<input type="checkbox"/>	4194320	0x21290001	addi \$9,\$9,1	8: addi \$t1,\$t1,1
<input type="checkbox"/>	4194324	0x200b0001	addi \$11,\$0,1	9: addi \$t3,\$zero,1 # z=1
<input type="checkbox"/>	4194328	0x08100009	j 4194340	10: j endif
<input type="checkbox"/>	4194332	0x214affff	addi \$10,\$10,-1	12: addi \$t2,\$t2,-1 # begin else part: y=y-1
<input type="checkbox"/>	4194336	0x016b5820	add \$11,\$11,\$11	13: add \$t3,\$t3,\$t3 # z=2*z

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Val
268500992	0	0	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0

0x10010000 (.data)


```

#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:

```

-Kết quả chạy

The screenshot displays the Mars MIPS simulator interface. The main window shows the assembly code being executed, with the following instructions highlighted:

```

4194304 0x2011fff9c addi $17,$0,-100 4: addi $s1, $0, -100 #i
4194308 0x20120064 addi $18,$0,100 5: addi $s2, $0, 100 #j
4194312 0x02329820 add $19,$17,$18 7: add $s3, $s1, $s2 #s3 = i + j
4194316 0x0013402a slt $t0,$s3 # i + j > 0 --> $t0 = 1
4194320 0x15000003 bne $t0,$s3 # $t0 != 0 --> else
4194324 0x21250001 addi $t1,$t1,1 11: addi $t1,$t1,1
4194328 0x200b0001 addi $t3,$zero,1 # z=1 12: addi $t3,$zero,1 # z=1
4194332 0x0010000a j endif 13: j endif
4194336 0x214fffff addi $t2,$t2,-1 15: addi $t2,$t2,-1 # begin else part: y=y-1
4194340 0x016b5820 add $t3,$t3,$t3 # z=2*z 16: add $t3,$t3,$t3 # z=2*z

```

The Data Segment window shows memory addresses and their corresponding values. The registers window on the right shows the current values of the registers. The registers \$s1, \$s2, and \$s3 contain the values -100, 100, and 0 respectively. The registers \$t1 and \$t3 contain the values 1 and 1 respectively. The registers \$t2 and \$t0 contain the values 0 and 0 respectively.

Vì $i + j = 0$ tức $i + j \leq 0$ thỏa mãn \rightarrow tiếp tục chạy tiếp chương trình mà không chuyển sang nhánh else \rightarrow $\$t1 = 1$ và $\$t3 = 1$

TH2: $i + j > 0$ ($i = 100; j = 100$)

-Kết quả chạy:

Text Segment								
Bkpt	Address	Code	Basic	Source		Name	Number	Value
	4194304	0x20110064	addi \$t7,\$0,100	4:	addi \$s1, \$0, 100 #i	\$zero	0	0
	4194308	0x20120064	addi \$t8,\$0,100	5:	addi \$s2, \$0, 100 #j	\$at	1	0
	4194312	0x00329820	add \$t9,\$t7,\$t8	7:	add \$s3, \$s1, \$s2 #s3 = i + j	\$v0	2	0
	4194316	0x00134020	slt \$t0,\$0,\$t9	8:	slt \$t0,\$zero,\$s3 # i + j > 0 --> \$t0 = 1	\$v1	3	0
	4194320	0x15000003	bne \$t0,\$0,3	9:	bne \$t0,\$zero,else # \$t0 != 0 --> else	\$a0	4	0
	4194324	0x21290001	addi \$t1,\$0,1	11:	addi \$t1,\$t1,1	\$a1	5	0
	4194328	0x200b0001	addi \$t1,\$0,1	12:	addi \$t3,\$zero,1 # z=1	\$a2	6	0
	4194332	0x0810000a	j 4194344	13:	j endif	\$a3	7	0
	4194336	0x214affff	addi \$t0,\$t0,-1	15:	addi \$t2,\$t2,-1 # begin else part: y=y-1	\$t0	8	1
	4194340	0x016b5820	add \$t1,\$t1,\$t1	16:	add \$t3,\$t3,\$t3 # z=2*z	\$t1	9	0
						\$t2	10	-1
						\$t3	11	0
						\$t4	12	0
						\$t5	13	0
						\$t6	14	0
						\$t7	15	0
						\$s0	16	0
						\$s1	17	100
						\$s2	18	100
						\$s3	19	200
						\$s4	20	0
						\$s5	21	0
						\$s6	22	0
						\$s7	23	0
						\$t8	24	0
						\$t9	25	0
						\$k0	26	0
						\$k1	27	0
						\$SP	28	268468284
						\$GP	29	2147479540
						\$FP	30	0
						\$ra	31	0
						pc		4194344
						hi		0
						lo		0

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500952	0	0	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0

Vì không thỏa mãn $i + j \leq 100 \rightarrow$ chuyển sang nhánh else $\rightarrow \$t2 = -1; \$t3 = 0$

d. $i + j > m + n$

TH1: $i + j > m + n$

```

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

```

-Kết quả chạy:


```

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

```

-Kết quả chạy

Text Segment					Name			Number	Value
Bkpt	Address	Code	Basic	Source					
	4194304	0x2011ffff	addi \$17,\$0,-1	4: addi \$s1, \$zero, -1 #i	\$zero			0	0
	4194308	0x20130002	addi \$19,\$0,2	5: addi \$s3, \$zero, 2 #n	\$at			1	268500592
	4194312	0x20140001	addi \$20,\$0,1	6: addi \$s4, \$zero, 1 #step	\$v0			2	0
	4194316	0x20150000	addi \$21,\$0,0	7: addi \$s5, \$zero, 0 #sum	\$v1			3	0
	4194320	0x3c011001	lui \$1,4097	8: la \$s2, A #array A	\$a0			4	0
	4194324	0x34320000	ori \$18,\$1,0		\$a1			5	0
	4194328	0x02348820	add \$17,\$17,\$20	10: loop: add \$s1,\$s1,\$s4 #i=i+step	\$a2			6	0
	4194332	0x02314820	add \$9,\$17,\$17	11: add \$t1,\$s1,\$s1 #t1=2*s1	\$a3			7	0
	4194336	0x01294820	add \$9,\$9,\$9	12: add \$t1,\$t1,\$t1 #t1=4*s1	\$t0			8	3
	4194340	0x01324820	add \$9,\$9,\$18	13: add \$t1,\$t1,\$s2 #t1 store the address of A[i]	\$t1			9	268501000
	4194344	0x8d280000	lw \$8,0(\$9)	14: lw \$t0,0(\$t1) #load value of A[i] in \$t0	\$t2			10	0
	4194348	0x02a8a820	add \$21,\$21,\$8	15: add \$s5,\$s5,\$t0 #sum=sum+A[i]	\$t3			11	0
					\$t4			12	0
					\$t5			13	0
					\$t6			14	0
					\$t7			15	0
					\$t8			16	0
					\$t9			17	2
					\$a2			18	268500592
					\$a3			19	2
					\$a4			20	1
					\$a5			21	6
					\$a6			22	0
					\$a7			23	0
					\$t8			24	0
					\$t9			25	0
					\$k0			26	0
					\$k1			27	0
					\$9p			28	268468224
					\$9p			29	2147475543
					\$fp			30	0
					\$ra			31	0
					pc				4194360
					hi				0
					lo				0

Data Segment							
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	V
268500592	1	2	3	4	0	0	
268501024	0	0	0	0	0	0	
268501056	0	0	0	0	0	0	
268501088	0	0	0	0	0	0	
268501120	0	0	0	0	0	0	
268501152	0	0	0	0	0	0	
268501184	0	0	0	0	0	0	
268501216	0	0	0	0	0	0	
268501248	0	0	0	0	0	0	
268501280	0	0	0	0	0	0	

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

b. $i \leq n$

-Mã nguồn

```

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

```

-Kết quả chạy

Run one step at a time

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	4194304	0x2011ffff	addi \$17,\$0,-1	5: addi \$s1, \$zero, -1 #i
<input type="checkbox"/>	4194308	0x20130002	addi \$19,\$0,2	6: addi \$s3, \$zero, 2 #n
<input type="checkbox"/>	4194312	0x20140001	addi \$20,\$0,1	7: addi \$s4, \$zero, 1 #step
<input type="checkbox"/>	4194316	0x20150000	addi \$21,\$0,0	8: addi \$s5, \$zero, 0 #sum
<input type="checkbox"/>	4194320	0x3c011001	lui \$1,4097	9: la \$s2, A #array A
<input type="checkbox"/>	4194324	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	4194328	0x20160001	addi \$22,\$0,1	10: addi \$s6, \$0, 1 #constant 1
<input type="checkbox"/>	4194332	0x02340820	add \$17,\$17,\$20	12: loop: add \$s1,\$s1,\$s4 #i=i+step
<input type="checkbox"/>	4194336	0x02314820	add \$9,\$17,\$17	13: add \$t1,\$s1,\$s1 #t1=2*s1
<input type="checkbox"/>	4194340	0x01294820	add \$9,\$9,\$9	14: add \$t1,\$t1,\$t1 #t1=4*s1
<input type="checkbox"/>	4194344	0x01324820	add \$9,\$9,\$18	15: add \$t1,\$t1,\$s2 #t1 store the address of A[i]
<input type="checkbox"/>	4194348	0x8d280000	lw \$8,0(\$9)	16: lw \$t0,0(\$t1) #load value of A[i] in \$t0

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	V
268500992	1	2	3	4	0	0	
268501024	0	0	0	0	0	0	
268501056	0	0	0	0	0	0	
268501088	0	0	0	0	0	0	
268501120	0	0	0	0	0	0	
268501152	0	0	0	0	0	0	
268501184	0	0	0	0	0	0	
268501216	0	0	0	0	0	0	
268501248	0	0	0	0	0	0	
268501280	0	0	0	0	0	0	

0x10010000 (.data)

☐ Hexadecimal Addresses☐ Hexadecimal Values☐ ASCII

Mars Messages

Run I/O

Registers

Coproc 1

Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	4
\$t1	9	268501004
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$w0	16	0
\$s1	17	3
\$s2	18	268500992
\$s3	19	2
\$s4	20	1
\$s5	21	10
\$s6	22	1
\$s7	23	0
\$s8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194364
hi		0
lo		0

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

c. sum \geq 0

-Mã nguồn

```

.text
    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

Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x201ffff	addi \$17,\$0,-1	6: addi \$s1, \$zero, -1 #i
	4194308	0x20130003	addi \$19,\$0,3	7: addi \$s3, \$zero, 3 #n
	4194312	0x20140001	addi \$20,\$0,1	8: addi \$s4, \$zero, 1 #step
	4194316	0x20150000	addi \$21,\$0,0	9: addi \$s5, \$zero, 0 #sum
	4194320	0x3e011001	lui \$1,4097	10: la \$s2, A #array A
	4194324	0x34320000	ori \$18,\$1,0	
	4194328	0x20160001	addi \$22,\$0,1	11: addi \$s6, \$0, 1 #constant 1
	4194332	0x02348820	add \$17,\$17,\$20	13: loop: add \$s1,\$s1,\$s4 #i=i+step
	4194336	0x02314820	add \$9,\$17,\$17	14: add \$t1,\$s1,\$s1 #t1=2*s1
	4194340	0x01294820	add \$9,\$9,\$9	15: add \$t1,\$t1,\$t1 #t1=4*s1
	4194344	0x01324820	add \$9,\$9,\$18	16: add \$t1,\$t1,\$s2 #t1 store the address of A[i]
	4194348	0x8d280000	lw \$8,0(\$9)	17: lw \$t0,0(\$t1) #load value of A[i] in \$t0

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)
268500992	-1	-2	-3	-4	0	0
268501024	0	0	0	0	0	0
268501056	0	0	0	0	0	0
268501088	0	0	0	0	0	0
268501120	0	0	0	0	0	0
268501152	0	0	0	0	0	0
268501184	0	0	0	0	0	0
268501216	0	0	0	0	0	0
268501248	0	0	0	0	0	0
268501280	0	0	0	0	0	0

0x10010000 (.data)

Hexadecimal Addresses

Hexadecimal Values

ASCII

Mars Messages Run I/O

-- program is finished running (dropped off bottom) --

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	-1
\$t1	9	268500992
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$s1	17	0
\$s2	18	268500992
\$s3	19	3
\$s4	20	1
\$s5	21	-1
\$s6	22	1
\$s7	23	0
\$s8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$fp	29	2147475548
\$fp	30	0
\$ra	31	0
pc		4194364
hi		0
lo		0

Khi $i = 0 \rightarrow \text{sum} = -1$; không thỏa mãn điều kiện $\text{sum} \geq 0$ 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 #array A

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

```

-Kết quả chạy

Edit Execute					Registers Coproc 1 Coproc 0		
Text Segment					Name	Number	Value
Bkpt	Address	Code	Basic	Source	\$zero	0	0
	4194308	0x20130003	addi \$19,\$0,3	8: addi \$s3, \$zero, 3 #n	\$at	1	268500992
	4194312	0x20140001	addi \$20,\$0,1	9: addi \$s4, \$zero, 1 #step	\$v0	2	0
	4194316	0x20150000	addi \$21,\$0,0	10: addi \$s5, \$zero, 0 #sum	\$v1	3	0
	4194320	0x3c011001	lui \$1,4097	11: la \$s2, A #array A	\$a0	4	0
	4194324	0x34320000	ori \$18,\$1,0		\$a1	5	0
	4194328	0x02348820	add \$17,\$17,\$20	13: loop: add \$s1,\$s1,\$s4 #i=i+step	\$a2	6	0
	4194332	0x02314820	add \$9,\$17,\$17	14: add \$t1,\$s1,\$s1 #t1=2*s1	\$a3	7	0
	4194336	0x01294820	add \$9,\$9,\$9	15: add \$t1,\$s1,\$s1 #t1=4*s1	\$a0	8	268501000
	4194340	0x01324820	add \$9,\$9,\$18	16: add \$t1,\$t1,\$s2 #t1 store the address of A[i]	\$t1	9	0
	4194344	0x8d280000	lw \$8,0(\$9)	17: lw \$t0,0(\$t1) #load value of A[i] in \$t0	\$t2	10	0
	4194348	0x02a8a820	add \$21,\$21,\$8	18: add \$s5,\$s5,\$t0 #sum=sum+A[i]	\$t3	11	0
	4194352	0x1100ff9	beq \$8,\$0,-7	19: beq \$t0, \$zero, loop #if A[i] == 0 then loop	\$t4	12	0
					\$t5	13	0
					\$t6	14	0
					\$t7	15	0
					\$a0	16	0
					\$a1	17	2
					\$a2	18	268500992
					\$a3	19	3
					\$a4	20	1
					\$a5	21	2
					\$a6	22	0
					\$a7	23	0
					\$a8	24	0
					\$a9	25	0
					\$a0	26	0
					\$a1	27	0
					\$gp	28	268468224
					\$op	29	2147479548
					\$fp	30	0
					\$ra	31	0
					pc		4194356
					hi		0
					lo		0

Data Segment						
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)
268500992	0	0	2	3	0	0
268501024	0	0	0	0	0	0
268501056	0	0	0	0	0	0
268501088	0	0	0	0	0	0
268501120	0	0	0	0	0	0
268501152	0	0	0	0	0	0
268501184	0	0	0	0	0	0
268501216	0	0	0	0	0	0
268501248	0	0	0	0	0	0
268501280	0	0	0	0	0	0

Mars Messages Run I/O

program is finished running (dropped off bottom) --

$A = \{ 0, 0, 2, 3 \}$. 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
    A:      .word    -10, 7, 16, -17, -123

.text

    la $s0, A           # Load A to $s0
    addi $s1, $zero, 0   # i
    addi $s2, $zero, 0   # max
    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
    sub $t0, $zero, $t0    # $t0 = 0 - $t0

ifgreat:
    slt $t2, $t0, $s2     # $t2 = A[i] < max ? 1 : 0
    bne $t2, $zero, endif
    add $s2, $t0, $zero    # max = A[i]

endif:

    addi $s1, $s1, 1      # i++
    j loop

end:

```

-Kết quả chương trình:

Text Segment						Name			
Bkpt	Address	Code	Basic	Source			Number	Value	
	4194304	0x3e011001	lui \$1,4097	4: la \$s0, A # Load A to \$s0		\$zero	0	0	
	4194308	0x34300000	ori \$16,\$1,0			\$at	1	1	
	4194312	0x20110000	addi \$17,\$0,0	5: addi \$s1, \$zero, 0 # i		\$v0	2	0	
	4194316	0x20120000	addi \$10,\$0,0	6: addi \$s2, \$zero, 0 # max		\$v1	3	0	
	4194320	0x20130005	addi \$19,\$0,5	7: addi \$s3, \$zero, 5 # n - num of elements		\$a0	4	0	
	4194324	0x0233502a	slt \$10,\$17,\$19	9: slt \$t2, \$s1, \$s3 #if i < n --> \$t2 = 1; else \$t2 = 0		\$a1	5	0	
	4194328	0x1140000e	beq \$10,\$0,\$14	10: beq \$t2, \$zero, end #if i = n or \$t2 = 0 --> end		\$a2	6	0	
	4194332	0x02314820	add \$9,\$17,\$17	11: add \$t1, \$s1, \$s1		\$a3	7	0	
	4194336	0x01294820	add \$9,\$9,\$9	12: add \$t1, \$t1, \$t1		\$a0	8	123	
	4194340	0x01304820	add \$9,\$9,\$16	13: add \$t1, \$t1, \$s0		\$t0	9	268501008	
	4194344	0x8d280000	lw \$8,0(\$9)	14: lw \$t0, 0(\$t1) # \$t0 = A[i]		\$t2	10	0	
	4194348	0x0100502a	slt \$10,\$8,\$0	16: sle \$t2, \$zero, \$t0		\$t3	11	0	
						\$t4	12	0	
						\$t5	13	0	
						\$t6	14	0	
						\$t7	15	0	
						\$a0	16	268500992	
						\$a1	17	5	
						\$a2	18	123	
						\$a3	19	5	
						\$a4	20	0	
						\$a5	21	0	
						\$a6	22	0	
						\$a7	23	0	
						\$t8	24	0	
						\$t9	25	0	
						\$k0	26	0	
						\$k1	27	0	
						\$gp	28	268468224	
						\$gp	29	2147479540	
						\$fp	30	0	
						\$ra	31	0	
						\$c		4194388	
						\$hi		0	
						\$lo		0	

Data Segment							
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)
268500992	-10	7	16	-17	-123	0	
268501024	0	0	0	0	0	0	
268501056	0	0	0	0	0	0	
268501088	0	0	0	0	0	0	
268501120	0	0	0	0	0	0	
268501152	0	0	0	0	0	0	
268501184	0	0	0	0	0	0	
268501216	0	0	0	0	0	0	
268501248	0	0	0	0	0	0	
268501280	0	0	0	0	0	0	

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

