

Bài thực hành số 3

Lớp: 147795 – Học phần: Thực hành kiến trúc máy tính

Họ và Tên: Trần Đình Dũng MSSV: 20215014

Assignment 1:

Tạo một chương trình để thực hiện mã Home Assignment 1

```
mips1.asm
1 #Laboratory Exercise 3, Home Assignment 1
2 .text
3     addi    $s1, $zero, 3    #i
4     addi    $s2, $zero, 4    #j
5 start:
6     slt     $t0,$s2,$s1      # j<i
7     bne    $t0,$zero,else   # branch to else if j<i
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:   addi    $t2,$t2,-1      # begin else part: y=y-1
12     add    $t3,$t3,$t3        # z=2*z
13 endif:
```

Biến i tương ứng với thanh ghi \$s1, j tương ứng với thanh ghi \$s2

Trong chương trình này, ta khởi tạo cho biến i và j với giá trị lần lượt là 3 và 4

Thực hiện chạy chương trình từng bước một, quan sát sự thay đổi của bộ nhớ và nội dung của các thanh ghi sau mỗi bước

The screenshot shows a MIPS assembly debugger interface. The top pane displays the assembly code for 'mips1.asm'. The 'Text Segment' pane shows the assembly instructions with their corresponding binary codes and addresses. The 'Data Segment' pane shows the initial values for variables \$s1, \$s2, \$t1, \$t2, and \$t3. The bottom pane, titled 'Registers', lists the register names, their numbers, and their current values. The registers \$s1 and \$s2 are highlighted in green, indicating they contain the values 3 and 4 respectively.

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	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t8	16	0x00000000
\$s1	17	0x00000003
\$s2	18	0x00000004
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000

Text Segment						Data Segment			
Bkpt	Address	Code	Basic	Source			Name	Number	Value
0x00400000	0x20110003	addi \$17,\$0,0x00000003	3:	addi \$s1,\$zero,3 #i			\$zero	0	0x00000000
0x00400004	0x20120004	addi \$18,\$0,0x00000004	4:	addi \$s2,\$zero,4 #j			\$at	1	0x00000000
0x00400008	0x0251402a	slt \$8,\$18,\$17	6:	slt \$t0,\$s2,\$s1 # j<i			\$v0	2	0x00000000
0x00400009	0x15000003	bne \$8,\$0,0x00000003	7:	bne \$t0,\$zero,else # branch to else if j<i			\$v1	3	0x00000000
0x00400010	0x21290001	addi \$9,\$9,0x00000001	8:	addi \$t1,\$t1,l # then part: x=x+l			\$a0	4	0x00000000
0x00400014	0x200b0001	addi \$11,\$0,0x00000001	9:	addi \$t3,\$zero,1 # z=1			\$a1	5	0x00000000
0x00400018	0x08100009	j 0x00400024	10:	j endif # skip "else" part			\$a2	6	0x00000000
0x0040001c	0x214affff	addi \$10,\$10,0xffff...	11: else:	addi \$t2,\$t2,-1 # begin else part: y=y-1			\$a3	7	0x00000000
0x00400020	0x01eb5820	add \$11,\$11,\$11	12:	add \$t3,\$t3,\$t3 # z=z*z			\$t0	8	0x00000000
							\$t1	9	0x00000000
							\$t2	10	0x00000000
							\$t3	11	0x00000000
							\$t4	12	0x00000000
							\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$s0	16	0x00000000
							\$s1	17	0x00000003
							\$s2	18	0x00000004
							\$s3	19	0x00000000
							\$s4	20	0x00000000
							\$s5	21	0x00000000
							\$s6	22	0x00000000

⇒ Thực hiện dòng lệnh 3, 4: gán giá trị của thanh ghi \$s1 = 3 và \$s2 = 4

Text Segment						Data Segment			
Bkpt	Address	Code	Basic	Source			Name	Number	Value
0x00400000	0x20110003	addi \$17,\$0,0x00000003	3:	addi \$s1,\$zero,3 #i			\$zero	0	0x00000000
0x00400004	0x20120004	addi \$18,\$0,0x00000004	4:	addi \$s2,\$zero,4 #j			\$at	1	0x00000000
0x00400008	0x0251402a	slt \$8,\$18,\$17	6:	slt \$t0,\$s2,\$s1 # j<i			\$v0	2	0x00000000
0x00400009	0x15000003	bne \$8,\$0,0x00000003	7:	bne \$t0,\$zero,else # branch to else if j<i			\$v1	3	0x00000000
0x00400010	0x21290001	addi \$9,\$9,0x00000001	8:	addi \$t1,\$t1,l # then part: x=x+l			\$a0	4	0x00000000
0x00400014	0x200b0001	addi \$11,\$0,0x00000001	9:	addi \$t3,\$zero,1 # z=1			\$a1	5	0x00000000
0x00400018	0x08100009	j 0x00400024	10:	j endif # skip "else" part			\$a2	6	0x00000000
0x0040001c	0x214affff	addi \$10,\$10,0xffff...	11: else:	addi \$t2,\$t2,-1 # begin else part: y=y-1			\$a3	7	0x00000000
0x00400020	0x01eb5820	add \$11,\$11,\$11	12:	add \$t3,\$t3,\$t3 # z=z*z			\$t0	8	0x00000000
							\$t1	9	0x00000000
							\$t2	10	0x00000000
							\$t3	11	0x00000000
							\$t4	12	0x00000000
							\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$s0	16	0x00000000

⇒ Thực hiện dòng lệnh 6: Nếu \$s2 < \$s1 (j < i) thì \$t0 = 1, ngược lại \$t0 = 0

Ta thấy \$s2 >= \$s1 (do j=4 > i=3) nên thanh ghi \$t0 được gán giá trị bằng 0

Text Segment						Data Segment			
Bkpt	Address	Code	Basic	Source			Name	Number	Value
0x00400000	0x20110003	addi \$17,\$0,0x00000003	3:	addi \$s1,\$zero,3 #i			\$zero	0	0x00000000
0x00400004	0x20120004	addi \$18,\$0,0x00000004	4:	addi \$s2,\$zero,4 #j			\$at	1	0x00000000
0x00400008	0x0251402a	slt \$8,\$18,\$17	6:	slt \$t0,\$s2,\$s1 # j<i			\$v0	2	0x00000000
0x00400009	0x15000003	bne \$8,\$0,0x00000003	7:	bne \$t0,\$zero,else # branch to else if j<i			\$v1	3	0x00000000
0x00400010	0x21290001	addi \$9,\$9,0x00000001	8:	addi \$t1,\$t1,l # then part: x=x+l			\$a0	4	0x00000000
0x00400014	0x200b0001	addi \$11,\$0,0x00000001	9:	addi \$t3,\$zero,1 # z=1			\$a1	5	0x00000000
0x00400018	0x08100009	j 0x00400024	10:	j endif # skip "else" part			\$a2	6	0x00000000
0x0040001c	0x214affff	addi \$10,\$10,0xffff...	11: else:	addi \$t2,\$t2,-1 # begin else part: y=y-1			\$a3	7	0x00000000
0x00400020	0x01eb5820	add \$11,\$11,\$11	12:	add \$t3,\$t3,\$t3 # z=z*z			\$t0	8	0x00000000
							\$t1	9	0x00000000
							\$t2	10	0x00000000
							\$t3	11	0x00000000
							\$t4	12	0x00000000
							\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$s0	16	0x00000000

⇒ Thực hiện dòng lệnh số 7: nếu \$t0 khác \$zero (hay \$t0 == 1) thì rẽ nhánh đến else. Vì trước đó \$t0 == 0 nên chương trình sẽ không nhảy đến else mà thực hiện tiếp dòng lệnh 8

Text Segment						Data Segment			
Bkpt	Address	Code	Basic	Source			Name	Number	Value
0x00400000	0x0251402a	slt \$8,\$18,\$17	6:	slt \$t0,\$s2,\$s1 # j<i			\$a0	4	0x00000000
0x00400001	0x15000003	bne \$8,\$0,0x00000003	7:	bne \$t0,\$zero,else # branch to else if j<i			\$a1	5	0x00000000
0x00400010	0x21290001	addi \$9,\$9,0x00000001	8:	addi \$t1,\$t1,l # then part: x=x+l			\$a2	6	0x00000000
0x00400014	0x200b0001	addi \$11,\$0,0x00000001	9:	addi \$t3,\$zero,1 # z=1			\$a3	7	0x00000000
0x00400018	0x08100009	j 0x00400024	10:	j endif # skip "else" part			\$t0	8	0x00000000
0x0040001c	0x214affff	addi \$10,\$10,0xffff...	11: else:	addi \$t2,\$t2,-1 # begin else part: y=y-1			\$t1	9	0x00000001
0x00400020	0x01eb5820	add \$11,\$11,\$11	12:	add \$t3,\$t3,\$t3 # z=z*z			\$t2	10	0x00000000
							\$t3	11	0x00000000
							\$t4	12	0x00000000
							\$t5	13	0x00000000
							\$t6	14	0x00000000
							\$t7	15	0x00000000
							\$s0	16	0x00000000

0x00400000	0x02514025	sit \$t0,\$t1,\$t1	6:	sit \$t0,\$t2,\$t1 # j< i
0x00400000	0x02514025	bne \$t0,\$t0,0x00000003	7:	bne \$t0,\$tzero,else # branch to else if j< i
0x00400010	0x02150001	addi \$t1,\$t1,1 # then part: x=x+1	8:	addi \$t1,\$t1,1 # then part: x=x+1
0x00400014	0x200b0001	addi \$t0,\$t0,0x00000001	9:	addi \$t3,\$tzero,1 # z=1
0x00400018	0x08010008	j 0x00400024	10:	j endif # skip "else" part
0x0040001c	0x214affff	addi \$t0,\$t0,0xffff...	11: else:	addi \$t2,\$t2,-1 # begin else part: y=y-1
0x00400020	0x01eb5620	add \$t1,\$t1,\$t1	12:	add \$t3,\$t3,\$t3 # z=z*x

\$t0	4	0x00000000
\$t1	5	0x00000000
\$t2	6	0x00000000
\$t3	7	0x00000000
\$t4	8	0x00000000
\$t5	9	0x00000000
\$t6	10	0x00000000
\$t7	11	0x00000001
\$s0	12	0x00000000
\$s1	13	0x00000000
\$s2	14	0x00000000
\$s3	15	0x00000000
\$s4	16	0x00000000

⇒ Thực hiện dòng lệnh 8: \$t1 = \$t1 + 1 (tương ứng với x = x + 1)

⇒ Thực hiện dòng lệnh 9: \$t3 = 0 + 1 (tương ứng với z = 1)

⇒ Thực hiện dòng lệnh 10: endif (bỏ qua phần “else” vì j ≥ i)

Text Segment			Name	Number	Value
Bkpt	Address	Code			
0x00400000	0x20110003	addi \$t1,\$t0,0x00000003	3:	addi \$t1,\$tzero,3 # i	0
0x00400004	0x20120004	addi \$t0,\$t0,0x00000004	4:	addi \$t0,\$tzero,4 #	1
0x00400008	0x02514025	sit \$t0,\$t1,\$t1	6:	sit \$t0,\$t2,\$t1 # j< i	2
0x00400010	0x20150001	addi \$t1,\$t1,1	7:	bne \$t0,\$tzero,else # branch to else if j< i	3
0x00400014	0x200b0001	addi \$t1,\$t1,0x00000001	9:	addi \$t3,\$tzero,1 # z=1	4
0x00400018	0x08010008	j 0x00400024	10:	j endif # skip "else" part	5
0x0040001c	0x214affff	addi \$t0,\$t0,0xffff...	11: else:	addi \$t2,\$t2,-1 # begin else part: y=y-1	6
0x00400020	0x01eb5620	add \$t1,\$t1,\$t1	12:	add \$t3,\$t3,\$t3 # z=z*x	7

Chương trình kết thúc.

Assignment 2:

Tạo một chương trình để thực hiện mã Home Assignment 2

```
mips1.asm
1  #Laboratory 3, Home Assignment 2
2  .data
3      A: .word      1,2,3
4  .text
5      la      $s2,A
6      addi   $s5,$zero,0      #sum=0
7      addi   $s1,$zero,-1      #i=-1
8      addi   $s4,$zero,1      #step=1
9      addi   $s3,$zero,2      #n=2
10 loop:
11     add    $s1,$s1,$s4      #i=i+step
12     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     bne   $s1,$s3,loop      #if i != n, goto loop
```

⇒ Các dòng lệnh từ 5 đến 9 thực hiện gán giá trị cho các thanh ghi:

\$s2 = địa chỉ mảng

$\$s5 = 0$ (sum = 0)

$\$s1 = -1$ (index i)

$\$s4 = 1$ (bước nhảy step)

$\$s3 = 2$ (hàng số so sánh n)

Registers

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	0x00000000
$\$t1$	9	0x00000000
$\$t2$	10	0x00000000
$\$t3$	11	0x00000000
$\$t4$	12	0x00000000
$\$t5$	13	0x00000000
$\$t6$	14	0x00000000
$\$t7$	15	0x00000000
$\$s0$	16	0x00000000
$\$s1$	17	0xffffffff
$\$s2$	18	0x10010000
$\$s3$	19	0x00000002
$\$s4$	20	0x00000001

Text Segment

Blkpt	Address	Code	Basic	Source
0x00400000	0x3c011001	lui \$1,0x00001001	5:	la \$s2,A
0x00400004	0x34320000	ori \$10,\$1,0x00000000		
0x00400008	0x20150000	addi \$21,\$0,0x00000000	6:	addi \$s5,\$zero,0 #sum=0
0x0040000c	0x2011ffff	addi \$17,\$0,0xffffffff	7:	addi \$s1,\$zero,-1 #i=-1
0x00400010	0x20140001	addi \$20,\$0,0x00000001	8:	addi \$s4,\$zero,1 #step=1
0x00400014	0x20130002	addi \$19,\$0,0x00000002	9:	addi \$s3,\$zero,2 #n=2
0x00400018	0x02348820	add \$17,\$17,\$20	11:	add \$s1,\$s1,\$s4 #i+=step
0x0040001c	0x02314820	add \$9,\$17,\$17	12:	add \$t1,\$s1,\$s1 #t1=2*s1
0x00400020	0x01294820	add \$9,\$9,\$9	13:	add \$t1,\$t1,\$t1 #t1=4*s1
0x00400024	0x01324820	add \$9,\$9,\$18	14:	add \$t1,\$t1,\$s2 #t1 store the address of A[i]
0x00400028	0x8d420000	lw \$8,0x00000000(\$9)	15:	lw \$t0,0(\$t1) #load value of A[i] in \$t0
0x0040002c	0x02a8a820	add \$21,\$21,\$8	16:	add \$s5,\$s5,\$t0 #sum=sum+A[i]
0x00400030	0x1e33ffff9	bne \$17,\$19,0xffffffff9	17:	bne \$s1,\$s3,loop #if i != n, goto loop

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Bắt đầu vòng lặp thứ 1:

Dòng lệnh 11: $\$s1 = \$s1 + \$s4 = -1 + 1 = 0$ ($i = i + step$)

⇒ Thanh ghi $\$s1$ nhận giá trị 0

Registers

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	0x00000000
$\$t1$	9	0x00000000
$\$t2$	10	0x00000000
$\$t3$	11	0x00000000
$\$t4$	12	0x00000000
$\$t5$	13	0x00000000
$\$t6$	14	0x00000000
$\$t7$	15	0x00000000
$\$s0$	16	0x00000000
$\$s1$	17	0x00000000
$\$s2$	18	0x10010000

Text Segment

Blkpt	Address	Code	Basic	Source
0x00400000	0x3c011001	lui \$1,0x00001001	5:	la \$s2,A
0x00400004	0x34320000	ori \$10,\$1,0x00000000		
0x00400008	0x20150000	addi \$21,\$0,0x00000000	6:	addi \$s5,\$zero,0 #sum=0
0x0040000c	0x2011ffff	addi \$17,\$0,0xffffffff	7:	addi \$s1,\$zero,-1 #i=-1
0x00400010	0x20140001	addi \$20,\$0,0x00000001	8:	addi \$s4,\$zero,1 #step=1
0x00400014	0x20130002	addi \$19,\$0,0x00000002	9:	addi \$s3,\$zero,2 #n=2
0x00400018	0x02348820	add \$17,\$17,\$20	11:	add \$s1,\$s1,\$s4 #i+=step
0x0040001c	0x02314820	add \$9,\$17,\$17	12:	add \$t1,\$s1,\$s1 #t1=2*s1
0x00400020	0x01294820	add \$9,\$9,\$9	13:	add \$t1,\$t1,\$t1 #t1=4*s1
0x00400024	0x01324820	add \$9,\$9,\$18	14:	add \$t1,\$t1,\$s2 #t1 store the address of A[i]
0x00400028	0x8d420000	lw \$8,0x00000000(\$9)	15:	lw \$t0,0(\$t1) #load value of A[i] in \$t0
0x0040002c	0x02a8a820	add \$21,\$21,\$8	16:	add \$s5,\$s5,\$t0 #sum=sum+A[i]
0x00400030	0x1e33ffff9	bne \$17,\$19,0xffffffff9	17:	bne \$s1,\$s3,loop #if i != n, goto loop

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Dòng lệnh 12, 13:

gán $\$t1 = \$s1 + \$s1$, sau đó gán $\$t1 = \$t1 + \$t1 = 4 * \$s1 = 0$

Registers

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	0x00000000
$\$t1$	9	0x00000000
$\$t2$	10	0x00000000
$\$t3$	11	0x00000000
$\$t4$	12	0x00000000
$\$t5$	13	0x00000000
$\$t6$	14	0x00000000
$\$t7$	15	0x00000000
$\$s0$	16	0x00000000
$\$s1$	17	0x00000000
$\$s2$	18	0x10010000

Text Segment

Blkpt	Address	Code	Basic	Source
0x00400000	0x3c011001	lui \$1,0x00001001	5:	la \$s2,A
0x00400004	0x34320000	ori \$10,\$1,0x00000000		
0x00400008	0x20150000	addi \$21,\$0,0x00000000	6:	addi \$s5,\$zero,0 #sum=0
0x0040000c	0x2011ffff	addi \$17,\$0,0xffffffff	7:	addi \$s1,\$zero,-1 #i=-1
0x00400010	0x20140001	addi \$20,\$0,0x00000001	8:	addi \$s4,\$zero,1 #step=1
0x00400014	0x20130002	addi \$19,\$0,0x00000002	9:	addi \$s3,\$zero,2 #n=2
0x00400018	0x02348820	add \$17,\$17,\$20	11:	add \$s1,\$s1,\$s4 #i+=step
0x0040001c	0x02314820	add \$9,\$17,\$17	12:	add \$t1,\$s1,\$s1 #t1=2*s1
0x00400020	0x01294820	add \$9,\$9,\$9	13:	add \$t1,\$t1,\$t1 #t1=4*s1
0x00400024	0x01324820	add \$9,\$9,\$18	14:	add \$t1,\$t1,\$s2 #t1 store the address of A[i]
0x00400028	0x8d420000	lw \$8,0x00000000(\$9)	15:	lw \$t0,0(\$t1) #load value of A[i] in \$t0
0x0040002c	0x02a8a820	add \$21,\$21,\$8	16:	add \$s5,\$s5,\$t0 #sum=sum+A[i]
0x00400030	0x1e33ffff9	bne \$17,\$19,0xffffffff9	17:	bne \$s1,\$s3,loop #if i != n, goto loop

Dòng lệnh 14: thanh ghi $\$t1$ lưu địa chỉ của $A[i]$ ($i=0$)

$\$t1 = \$t1 + \$s2 = 0x00000000 + 0x10010000 = 0x10010000$

Registers

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	0x00000000
\$t1	9	0x10010000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c011001	lui	\$1,0x00000001	
0x00400004	0x34320000	ori	\$18,\$1,0x00000000	
0x00400008	0x20150000	addi	\$21,\$0,0x00000000	6: addi \$s5,\$zero,0 #sum=0
0x0040000c	0x2011ffff	addi	\$17,\$0,0xffffffff	7: addi \$s1,\$zero,-1 #i=-1
0x00400010	0x20140001	addi	\$20,\$0,0x00000001	8: addi \$s4,\$zero,1 #step=1
0x00400014	0x20130002	addi	\$19,\$0,0x00000002	9: addi \$s3,\$zero,2 #n=2
0x00400018	0x20134882	addi	\$17,\$17,\$20	11: add \$s1,\$s1,\$s4 #i+=step
0x0040001c	0x20134820	add	\$9,\$17,\$17	12: add \$t1,\$s1,\$s1 #i=2*s1
0x00400020	0x20129482	add	\$9,\$9,\$9	13: add \$t1,\$t1,\$t1 #i=4*s1
0x00400024	0x20132482	add	\$9,\$9,\$18	14: add \$t1,\$t1,\$s2 #t1 store the address of A[i]
0x00400028	0x8d280000	lw	\$8,0x00000000(\$9)	15: lw \$t0,0(\$t1) #load value of A[i] in \$t0
0x0040002c	0x202a8a20	addi	\$21,\$21,\$8	16: add \$s5,\$s5,\$t0 #sum=sum+A[i]
0x00400030	0x1e33ffff	bne	\$17,\$19,0xffffffff	17: bne \$s1,\$s3,loop #if i != n, goto loop

Dòng lệnh 15: tải giá trị của A[i] vào \$t0 (i=0)

$$\$t0 = A[0] = 1 \text{ (hay } 0x00000001 \text{ ở hexa)}$$

Registers

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	0x10010000
\$t1	9	0x10010000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c011001	lui	\$1,0x00000001	
0x00400004	0x34320000	ori	\$18,\$1,0x00000000	
0x00400008	0x20150000	addi	\$21,\$0,0x00000000	6: addi \$s5,\$zero,0 #sum=0
0x0040000c	0x2011ffff	addi	\$17,\$0,0xffffffff	7: addi \$s1,\$zero,-1 #i=-1
0x00400010	0x20140001	addi	\$20,\$0,0x00000001	8: addi \$s4,\$zero,1 #step=1
0x00400014	0x20130002	addi	\$19,\$0,0x00000002	9: addi \$s3,\$zero,2 #n=2
0x00400018	0x20134882	addi	\$17,\$17,\$20	11: add \$s1,\$s1,\$s4 #i+=step
0x0040001c	0x20134820	add	\$9,\$17,\$17	12: add \$t1,\$s1,\$s1 #i=2*s1
0x00400020	0x20129482	add	\$9,\$9,\$9	13: add \$t1,\$t1,\$t1 #i=4*s1
0x00400024	0x20132482	add	\$9,\$9,\$18	14: add \$t1,\$t1,\$s2 #t1 store the address of A[i]
0x00400028	0x8d280000	lw	\$8,0x00000000(\$9)	15: lw \$t0,0(\$t1) #load value of A[i] in \$t0
0x0040002c	0x202a8a20	addi	\$21,\$21,\$8	16: add \$s5,\$s5,\$t0 #sum=sum+A[i]
0x00400030	0x1e33ffff	bne	\$17,\$19,0xffffffff	17: bne \$s1,\$s3,loop #if i != n, goto loop

Dòng lệnh 16: Cập nhật giá trị thanh ghi \$s5

$$\$s5 = \$s5 + \$t0 = 0x00000000 + 0x00000001 = 0x00000001 \text{ (hay } 1 \text{ ở hệ thập phân)}$$

Registers

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	0x00000001
\$t1	9	0x10010000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x10010000
\$s3	19	0x00000000
\$s4	20	0x00000001
\$s5	21	0x00000001
\$s6	22	0x00000000

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c011001	lui	\$1,0x00000001	
0x00400004	0x34320000	ori	\$18,\$1,0x00000000	
0x00400008	0x20150000	addi	\$21,\$0,0x00000000	6: addi \$s5,\$zero,0 #sum=0
0x0040000c	0x2011ffff	addi	\$17,\$0,0xffffffff	7: addi \$s1,\$zero,-1 #i=-1
0x00400010	0x20140001	addi	\$20,\$0,0x00000001	8: addi \$s4,\$zero,1 #step=1
0x00400014	0x20130002	addi	\$19,\$0,0x00000002	9: addi \$s3,\$zero,2 #n=2
0x00400018	0x20134882	addi	\$17,\$17,\$20	11: add \$s1,\$s1,\$s4 #i+=step
0x0040001c	0x20134820	add	\$9,\$17,\$17	12: add \$t1,\$s1,\$s1 #i=2*s1
0x00400020	0x20129482	add	\$9,\$9,\$9	13: add \$t1,\$t1,\$t1 #i=4*s1
0x00400024	0x20132482	add	\$9,\$9,\$18	14: add \$t1,\$t1,\$s2 #t1 store the address of A[i]
0x00400028	0x8d280000	lw	\$8,0x00000000(\$9)	15: lw \$t0,0(\$t1) #load value of A[i] in \$t0
0x0040002c	0x202a8a20	addi	\$21,\$21,\$8	16: add \$s5,\$s5,\$t0 #sum=sum+A[i]
0x00400030	0x1e33ffff	bne	\$17,\$19,0xffffffff	17: bne \$s1,\$s3,loop #if i != n, goto loop

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Dòng lệnh 17: So sánh giá trị thanh ghi \$s1 và \$s3, nếu \$s1 khác \$s3 thì tiếp tục lặp, ở đây ta có giá trị \$s1 = 0x00000000 khác \$s3 = 0x00000002 nên vòng lặp tiếp tục với điểm bắt đầu là dòng lệnh 11

Text Segment		Registers					
Bkpt	Address	Code	Basic	Source	Name	Number	Value
0x00400000	0x3c010101	lui \$1,0x00001001	5:	la \$s2,\$a	\$zero	0	0x00000000
0x00400004	0x34320000	ori \$18,\$1,0x00000000			\$at	1	0x10010000
0x00400008	0x20150000	addi \$t1,\$0,0x00000000	6:	addi \$s5,\$zero,0 #sum=0	\$v0	2	0x00000000
0x0040000c	0x2011ffff	addi \$t1,\$0,0xffffffff	7:	addi \$s1,\$zero,-1 #i=-1	\$v1	3	0x00000000
0x00400010	0x20140001	addi \$t0,\$0,0x00000001	8:	addi \$s4,\$zero,1 #step1	\$a0	4	0x00000000
0x00400014	0x20130002	addi \$t1,\$0,0x00000002	9:	addi \$s3,\$zero,2 #n=2	\$a1	5	0x00000000
0x00400018	0x02348800	addi \$t1,\$17,\$20	11:	addi \$s1,\$s1,\$s4 #i+=step	\$a2	6	0x00000000
0x0040001c	0x02124020	add \$t1,\$s1,\$t1	12:	add \$t1,\$s1,\$s1 #t1+=s1	\$a3	7	0x00000000
0x00400020	0x01254020	add \$t1,\$s1,\$s1	13:	add \$t1,\$t1,\$t1 #t1+=s1	\$t0	8	0x00000001
0x00400024	0x01324020	add \$t1,\$s1,\$s1	14:	add \$t1,\$t1,\$t1 #t1 stores the address of A[i]	\$t1	9	0x00000000
0x00400028	0x08d20000	lw \$t0,0(\$t1)	15:	lw \$t0,0(\$t1) #load value of A[i] in t0	\$t2	10	0x00000000
0x0040002c	0x02a8e200	add \$t0,\$s1,\$t0	16:	add \$s5,\$s5,\$t0 #sum+=A[i]	\$t3	11	0x00000000
0x00400030	0x1e33ffff	bne \$t1,\$19,0xffffffff	17:	bne \$s1,\$s3,loop #if i != n, goto loop	\$t4	12	0x00000000
					\$t5	13	0x00000000
					\$t6	14	0x00000000
					\$t7	15	0x00000000
					\$s0	16	0x00000000
					\$s1	17	0x00000000
					\$s2	18	0x10010000
					\$s3	19	0x00000002
					\$s4	20	0x00000001
					\$s5	21	0x00000001

Vòng lặp thứ 2 và 3 tương tự như vòng lặp 1

Sau vòng lặp thứ 3, giá trị các thanh ghi như sau:

$\$s1 = 2$ (hay 0x00000002 ở hệ hexa)

$\$t1 = 0x10010008$

$\$t0 = 3$ (hay 0x00000003 ở hệ hexa)

$\$s5 = 6$ (hay 0x00000006 ở hệ hexa)

Lúc này giá trị thanh ghi $\$s1 = 2$ và bằng với $\$s3$ nên chương trình thoát khỏi vòng lặp.

Text Segment		Registers					
Bkpt	Address	Code	Basic	Source	Name	Number	Value
0x00400000	0x3c010101	lui \$1,0x00001001	5:	la \$s2,\$a	\$zero	0	0x00000000
0x00400004	0x34320000	ori \$18,\$1,0x00000000			\$at	1	0x10010000
0x00400008	0x20150000	addi \$t1,\$0,0x00000000	6:	addi \$s5,\$zero,0 #sum=0	\$v0	2	0x00000000
0x0040000c	0x2011ffff	addi \$t1,\$0,0xffffffff	7:	addi \$s1,\$zero,-1 #i=-1	\$v1	3	0x00000000
0x00400010	0x20140001	addi \$t0,\$0,0x00000001	8:	addi \$s4,\$zero,1 #step1	\$a0	4	0x00000000
0x00400014	0x20130002	addi \$t1,\$0,0x00000002	9:	addi \$s3,\$zero,2 #n=2	\$a1	5	0x00000000
0x00400018	0x02348800	addi \$t1,\$17,\$20	11:	addi \$s1,\$s1,\$s4 #i+=step	\$a2	6	0x00000000
0x0040001c	0x02124020	add \$t1,\$s1,\$t1	12:	add \$t1,\$s1,\$s1 #t1+=s1	\$t0	8	0x00000003
0x00400020	0x01254020	add \$t1,\$s1,\$s1	13:	add \$t1,\$t1,\$t1 #t1+=s1	\$t1	9	0x10010000
0x00400024	0x01324020	add \$t1,\$s1,\$s1	14:	add \$t1,\$t1,\$t1 #t1 stores the address of A[i]	\$t2	10	0x00000000
0x00400028	0x08d20000	lw \$t0,0(\$t1)	15:	lw \$t0,0(\$t1) #load value of A[i] in t0	\$t3	11	0x00000000
0x0040002c	0x02a8e200	add \$t0,\$s1,\$t0	16:	add \$s5,\$s5,\$t0 #sum+=A[i]	\$t4	12	0x00000000
0x00400030	0x1e33ffff	bne \$t1,\$19,0xffffffff	17:	bne \$s1,\$s3,loop #if i != n, goto loop	\$t5	13	0x00000000
					\$t6	14	0x00000000
					\$t7	15	0x00000000
					\$s0	16	0x00000000
					\$s1	17	0x00000002
					\$s2	18	0x10010000
					\$s3	19	0x00000002
					\$s4	20	0x00000001
					\$s5	21	0x00000001

Chương trình kết thúc

Assignment 3:

Tạo một chương trình để thực hiện mã Home Assignment 2

```

mips1.asm mips2.asm
1 #Laboratory Exercise 3, Home Assignment 3
2 .data
3     test: .word 1
4     addi $s2,$zero,1
5     addi $s3,$zero,2
6 .text
7     la $s0, test      #load the address of test variable
8     lw $s1, 0($s0)    #load the value of test to register $t1
9     li $t0, 0          #load value for test case
10    li $t1, 1
11    li $t2, 2
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:

```

Ta gán giá trị cho 2 thanh ghi \$s2, \$s3 lần lượt là 1 và 2

Dòng lệnh 7: gán giá trị thanh ghi \$s0 bằng địa chỉ test

Edit Execute			
Text Segment			
Bkpt	Address	Code	Basic
	0x00400000	0x3c011001	lui \$1,0x00001001
7:	0x00400004	0x34300000	ori \$16,\$1,0x00000000
	0x00400008	0x8e110000	lw \$s1,0(\$s0) #load the value of test to register \$t1
	0x0040000c	0x24000000	addiu \$t0,\$0,0x00000000
	0x00400010	0x24050001	addiu \$s5,\$0,0x00000001
	0x00400014	0x24040002	addiu \$t1,\$0,0x0000... 11:
	0x00400018	0x12280003	beq \$s1,\$t0,case_0
	0x0040001c	0x12290004	beq \$s1,\$t1,case_1
	0x00400020	0x122a0005	beq \$s1,\$t2,case_2
	0x00400024	0x00100010	j default
	0x00400028	0x22520001	addi \$s2,\$s2,1 #a=a+1

Registers Coproc 1 Coproc 0			
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	0x00000000	
\$t1	9	0x00000000	
\$t2	10	0x00000000	
\$t3	11	0x00000000	
\$s-a	12	nnnnnnnn	

Dòng lệnh 8: ghi giá trị thanh ghi \$s1 bằng giá trị test =1

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	0x00000000
\$t1	9	0x00000001
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x10010000
\$s1	17	0x00000001
\$s2	18	0x00000000

Dòng lệnh 9-11: lần lượt gán các giá trị vào các thanh ghi:

$\$t0 = 0$

$\$t1 = 1$

$\$t2 = 2$

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	0x00000000
\$t1	9	0x00000001
\$t2	10	0x00000002
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x10010000
\$s1	17	0x00000001
\$s2	18	0x00000000

Dòng lệnh 12: thực hiện so sánh 2 thanh ghi $\$s1$ và $\$t0$

⇒ Nếu $\$s1 == \$t0$ thì rẽ nhánh đến case_0

Tương tự với dòng lệnh 13, 14 sẽ rẽ nhánh đến case_1 và case_2

ở đây vì giá trị thanh $\$s1 = \$t1 = 1$ nên chương trình rẽ nhánh tại dòng lệnh 13 tới case_1, sau đó thực hiện ghi giá trị mới cho thanh ghi $\$s2 = \$s2 - \$t1 = 0x00000000 - 0x00000001 = 0xffffffff$

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	0x00000000
\$t1	9	0x00000001
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x10010000
\$s1	17	0x00000001
\$s2	18	0xffffffff
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000

thực hiện dòng lệnh thứ 21: rẽ nhánh đến continue và kết thúc chương trình.

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	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x10010000
\$s1	17	0x00000001
\$s2	18	0xffffffff
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000

Assignment 4:

a) $i < j$

để đổi điều kiện kiểm tra thành $i < j$, ta thay đổi dòng lệnh ‘slt \$t0, \$s2, \$s1’ thành: sgt \$t0, \$s2, \$s1

```

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

```

Khi thực hiện chương trình, do $j=4 > i=3$ nên thanh ghi \$t0 có giá trị bằng 1

⇒ chương trình sẽ rẽ nhánh đến else

chương trình thực hiện dòng lệnh 13:

gán giá trị thanh ghi \$t2 = \$t2 - 1 = 0x00000000 - 0x00000001 = 0xffffffff

thực hiện dòng lệnh 14:

$$\$t3 = \$t3 + \$t3 = 0x00000000 + 0x00000000 = 0x00000000$$

Text Segment						Registers			
Bkpt	Address	Code	Basic	Source		Name	Number	Value	
	0x00400000	0x20110003	addi \$t1,\$zero,3	4:	addi \$t1,\$zero,3	#i	tzero	0	0x00000000
	0x00400004	0x20120004	addi \$t1,\$zero,4	5:	addi \$t1,\$zero,4	#j	tz0	1	0x00000000
	0x00400008	0x0232402a	slt \$t0,\$t1,\$t1	7:	slt \$t0,\$t1,\$t1	# kiem tra i>=j <=> kiem tra j > i	tz1	2	0x00000000
	0x00400010	0x21500003	beq \$t0,\$zero,else	8:	beq \$t0,\$zero,else	# branch to else if i >= j	tz0	3	0x00000000
	0x00400014	0x200b0001	addi \$t1,\$t1,1	9:	addi \$t1,\$t1,1	# then part: x=x+1	ta0	4	0x00000000
	0x00400018	0x08100005	addi \$t3,\$zero,1	10:	addi \$t3,\$zero,1	# z=1	ta1	5	0x00000000
	0x0040001c	0x144fffff	addi \$t2,\$t1,-1	11:	j endif	# skip "else" part	ta2	6	0x00000000
	0x00400020	0x01eb5820	add \$t1,\$t1,\$t1	13:	add \$t1,\$t1,\$t1	# begin else part: y=y-1	ta3	7	0x00000000
				14:	add \$t3,\$t3,\$t3	# z=2*z	ta4	8	0x00000000
							ta5	9	0x00000000
							ta6	10	0xffffffff
							ta7	11	0x00000000
							ta8	12	0x00000000
							ta9	13	0x00000000
							ta10	14	0x00000000
							ta11	15	0x00000000
							ta12	16	0x00000000

b) $i \geq j$

ta có kiểm tra $i \geq j$ tương đương với kiểm tra $j < i$

để đổi điều kiện kiểm tra thành $i \geq j$, t cập nhật lại chương trình như sau:

mips1.asm	mips2.asm
<pre> 1 #Laboratory Exercise 3, Home Assignment 1 2 3 .text 4 addi \$s1, \$zero, 3 #i 5 addi \$s2, \$zero, 4 #j 6 start: 7 slt \$t0, \$s1, \$s2 # kiem tra i>=j <=> kiem tra j > i 8 beq \$t0, \$zero, else # branch to else if i >= j 9 addi \$t1,\$t1,1 # then part: x=x+1 10 addi \$t3,\$zero,1 # z=1 11 j endif # skip "else" part 12 else: 13 addi \$t2,\$t2,-1 # begin else part: y=y-1 14 add \$t3,\$t3,\$t3 # z=2*z 15 endif: </pre>	<pre> 1 #Laboratory Exercise 3, Home Assignment 1 2 3 .text 4 addi \$s1, \$zero, 3 #i 5 addi \$s2, \$zero, 4 #j 6 start: 7 sge \$t0, \$s1, \$s2 # kiem tra i>=j <=> kiem tra j > i 8 beq \$t0, \$zero, else # branch to else if i >= j 9 addi \$t1,\$t1,1 # then part: x=x+1 10 addi \$t3,\$zero,1 # z=1 11 j endif # skip "else" part 12 else: 13 addi \$t2,\$t2,-1 # begin else part: y=y-1 14 add \$t3,\$t3,\$t3 # z=2*z 15 endif: </pre>

Thực hiện chương trình:

Do khởi tạo $i=3 < j=4$ nên thanh ghi $\$t0$ có giá trị = 1

⇒ không rẽ nhánh

chương trình tiếp tục thực hiện dòng lệnh 9, 10:

$$\$t1 = \$t1 + 1 = 0x00000000 + 0x00000001 = 0x00000001$$

$$\$t3 = 0 + 1 = 0x00000000 + 0x00000001 = 0x00000001$$

Registers				Coproc 1	Coproc 0
Name	Number	Value			
tzzero	0	0x00000000			
tzt0	1	0x00000000			
tzt1	2	0x00000000			
tzta0	3	0x00000000			
tzta1	4	0x00000000			
tzta2	5	0x00000000			
tzta3	6	0x00000000			
tzta4	7	0x00000000			
tzta5	8	0x00000000			
tzta6	9	0x00000000			
tzta7	10	0x00000000			
tzta8	11	0x00000000			
tzta9	12	0x00000000			
tzta10	13	0x00000000			
tzta11	14	0x00000000			
tzta12	15	0x00000000			
tzta13	16	0x00000000			
tzta14	17	0x00000000			
tzta15	18	0x00000000			

Thực hiện dòng lệnh 11: bỏ qua phần 'else' và kết thúc chương trình

c) $i+j \leq 0$

để đổi điều kiện kiểm tra thành $i+j \leq 0$, ta cập nhật chương trình như sau

```

1 #Laboratory Exercise 3, Home Assignment 1
2
3 .text
4     addi $s1, $zero, 3      #i
5     addi $s2, $zero, 4      #j
6 start:
7     add $s5, $s1, $s2      # luu gia tri i + j vao $s5
8     slt $t0, $zero, $s5    # tra ve 1 neu 0 < $s5, nguoc lai tra ve 0
9     bne $t0, $zero, else   # branch to else if i + j <= 0
10    addi $t1,$t1,1          # then part: x=x+1
11    addi $t3,$zero,1        # z=1
12    j endif                # skip "else" part
13 else:
14    addi $t2,$t2,-1         # begin else part: y=y-1
15    add $t3,$t3,$t3          # z=2*z
16 endif:

```

Gán giá trị thanh ghi $\$s5 = \$s1 + \$s2$, nếu giá trị $\$s5 > 0$ thì giá trị $\$t0 = 1$, ngược lại thì bằng 0. Thực hiện dòng lệnh 9 nếu $\$t0$ khác 0 thì rẽ nhánh đến else

ở đây do $\$s5 = \$s1 + \$s2 = 3 + 4 = 7$ nên giá trị thanh ghi $\$t0 = 1$

⇒ chương trình rẽ nhánh đến else

thực hiện dòng lệnh 14:

$$\$t2 = \$t2 - 1 = 0x00000000 - 0x00000001 = 0xffffffff$$

thực hiện dòng lệnh 15:

$$\$t3 = \$t3 + \$t3 = 0x00000000 + 0x00000000 = 0x00000000$$

Registers	Toproc t	Toproc u
\$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	0x00000000
\$t2	10	0xffffffff
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000

d) $i+j > m+n$

Ta cập nhật lại chương trình:

```

mips1.asm mips2.asm
1 #Laboratory Exercise 3, Home Assignment 1
2
3 .text
4     addi $s1, $zero, 3      #i
5     addi $s2, $zero, 4      #j
6 start:
7     li $t5, 5                # m = 5
8     li $t6, 6                # n = 6
9     add $s6, $t5, $t6        # m + n, store in $s6
10    add $s5, $s1, $s2        # i + j, store in $s5
11    slt $t0, $s6, $s5        # tra ve 1 m + n < i + j, nguoc lai tra ve 0
12    beq $t0, $zero, else      # branch to else if m + n ≥ i + j
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:

```

thực hiện chương trình:

gán giá trị cho thanh ghi \$t5 và \$t6 với giá trị là 5 và 6 (tương ứng với m, n)

gán giá trị cho thanh ghi \$s6 = \$t5 + \$t6 (hay m + n)

gán giá trị cho thanh ghi \$s5 = \$s1 + \$s2 (hay i + j)

câu lệnh 11 thực hiện so sánh \$s6 (m+n) và \$s5 (i+j), nếu \$s6 < \$s5 thì gán giá trị \$t0 bằng 1, ngược lại gán bằng 0

ở đây do $m+n = 5+6 = 11 > i+j = 3+4 = 7$

nên giá trị \$t0 = 0

câu lệnh 12: do \$t0 = 0 nên chương trình thực hiện rẽ nhánh đến else

thực hiện dòng lệnh 17:

$\$t2 = \$t2 - 1 = 0x00000000 - 0x00000001 = 0xffffffff$

thực hiện dòng lệnh 18:

$\$t3 = \$t3 + \$t3 = 0x00000000 + 0x00000000 = 0x00000000$

Registers			
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	0x00000000	
\$t1	9	0x00000000	
\$t2	10	0xffffffff	
\$t3	11	0x00000000	
\$t4	12	0x00000000	
\$t5	13	0x00000005	
\$t6	14	0x00000006	
\$t7	15	0x00000000	
\$s0	16	0x00000000	

Assignment 5:

a) $i < n$

để đổi điều kiện vòng lặp thành $i < n$, ta thay đổi dòng lệnh ban đầu:

bne \$s1, \$s3, loop, trở thành

blt \$s1, \$s3, loop

```
10    loop:  
11        add    $s1,$s1,$s4      #i=i+step  
12        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        blt   $s1, $s3, loop  # if i < n, goto loop
```

khi điều chỉnh điều kiện lặp là $i < n$ thì kết quả sẽ giống hệt với kết quả của chương trình ban đầu với điều kiện lặp $i \neq n$

b) $i \leq n$

để đổi điều kiện vòng lặp thành $i < n$, ta thay đổi dòng lệnh 17 ban đầu thành: ble \$s1, \$s3, loop

```
10    loop:  
11        add    $s1,$s1,$s4      #i=i+step  
12        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        ble   $s1, $s3, loop  # if i \leq n, goto loop
```

Khi thực hiện chương trình với mảng A [] = {1,2,3,4} và n = 3

Ta được kết quả với tổng mới là \$s5 = 0x0000000a (hay 10 ở hệ thập phân)

Text Segment						Registers					
Bkpt	Address	Code	Basic	Source			Name	Number	Value		
	0x00400000	0x3c010001	lui \$1,0x00001001	S:	la	\$s2,A	\$zero	0	0x00000000		
	0x00400004	0x34320000	ori \$18,\$1,0x00000000				\$at	1	0x00000001		
	0x00400008	0x20150000	addi \$21,\$0,0x00000000	E:	addi	\$s5,\$zero,0 #sum=0	\$v0	2	0x00000000		
	0x0040000c	0x2011ffff	addi \$17,\$0,0xfffffff7	7:	addi	\$s1,\$zero,-1 #i=-1	\$v1	3	0x00000000		
	0x00400010	0x20140001	addi \$20,\$0,0x00000001	8:	addi	\$s4,\$zero,1 #step=1	\$a0	4	0x00000000		
	0x00400014	0x20130003	addi \$19,\$0,0x00000003	9:	addi	\$s3,\$zero,3 #n=3	\$a1	5	0x00000000		
	0x00400018	0x02340000	add \$17,\$17,\$20	11:	add	\$s1,\$s1,\$s4 #i+=step	\$a2	6	0x00000000		
	0x00400020	0x02314000	add \$5,\$17,\$17	12:	add	\$t1,\$s1,\$s1 #i=2*s1	\$a3	7	0x00000000		
	0x00400024	0x01294000	add \$5,\$9,\$9	13:	add	\$t1,\$t1,\$t1 #i=4*s1	\$t0	8	0x00000000		
	0x00400028	0x01324000	add \$5,\$9,\$10	14:	add	\$t1,\$t1,\$s2 #t1 store the address of A[i]	\$t1	9	0x10010010		
	0x00400032	0x02a00000	lw \$0,0x00000000(\$t1)	15:	lw	\$t0,0(\$t1) #load value of A[i] in \$t0	\$t2	10	0x00000000		
	0x00400036	0x02a0a000	add \$21,\$21,\$0	16:	add	\$s5,\$s5,\$t0 #sum=sum+A[i]	\$t3	11	0x00000000		
	0x00400040	0x02710820	ble \$s1,\$19,\$17	17:	bne	\$s1,\$s3,loop # if i <= n, goto loop	\$t4	12	0x00000000		
	0x00400044	0x10000000	lw \$1,0x00000000(\$t1)				\$t5	13	0x00000000		
	0x00400048	0x10000000	add \$1,0x00000000				\$t6	14	0x00000000		
	0x00400052	0x10000000	add \$1,0x00000000				\$t7	15	0x00000000		
	0x00400056	0x10000000	add \$1,0x00000000				\$s0	16	0x00000000		
	0x00400060	0x10000000	add \$1,0x00000000				\$s1	17	0x00000004		
	0x00400064	0x10000000	add \$1,0x00000000				\$s2	18	0x10010000		
	0x00400068	0x10000000	add \$1,0x00000000				\$s3	19	0x00000003		
	0x00400072	0x10000000	add \$1,0x00000000				\$s4	20	0x00000001		
	0x00400076	0x10000000	add \$1,0x00000000				\$s5	21	0x0000000a		
	0x00400080	0x10000000	add \$1,0x00000000				\$s6	22	0x00000000		

c) sum ≥ 0

để đổi điều kiện vòng lặp thành sum ≥ 0 , ta thay đổi dòng lệnh 17 ban đầu thành: bge \$s5, \$zero, loop

```

10    loop:
11        add    $s1,$s1,$s4      #i=i+step
12        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        bge   $s5,$zero,loop   # if sum >= 0, goto loop

```

Khi thực hiện chương trình với mảng A [] = {1, 2, -3, -4} và n = 3

Ta được kết quả với tổng mới là \$s5 = 0xfffffffffc hay -4 ở thời điểm dừng

Text Segment						Registers					
Bkpt	Address	Code	Basic	Source			Name	Number	Value		
	0x00400000	0x3c010001	lui \$1,0x00001001	S:	la	\$s2,A	\$zero	0	0x00000000		
	0x00400004	0x34320000	ori \$18,\$1,0x00000000				\$at	1	0x00000001		
	0x00400008	0x20150000	addi \$21,\$0,0x00000000	E:	addi	\$s5,\$zero,0 #sum=0	\$v0	2	0x00000000		
	0x0040000c	0x2011ffff	addi \$17,\$0,0xfffffff7	7:	addi	\$s1,\$zero,-1 #i=-1	\$v1	3	0x00000000		
	0x00400010	0x20140001	addi \$20,\$0,0x00000001	8:	addi	\$s4,\$zero,1 #step=1	\$a0	4	0x00000000		
	0x00400014	0x20130003	addi \$19,\$0,0x00000003	9:	addi	\$s3,\$zero,3 #n=3	\$a1	5	0x00000000		
	0x00400018	0x02340000	add \$17,\$17,\$20	11:	add	\$s1,\$s1,\$s4 #i+=step	\$a2	6	0x00000000		
	0x00400020	0x02314000	add \$5,\$17,\$17	12:	add	\$t1,\$s1,\$s1 #i=2*s1	\$t0	8	0xfffffffffc		
	0x00400024	0x01294000	add \$5,\$9,\$9	13:	add	\$t1,\$t1,\$t1 #i=4*s1	\$t1	9	0x10010010		
	0x00400028	0x01324000	add \$5,\$9,\$10	14:	add	\$t1,\$t1,\$s2 #t1 store the address of A[i]	\$t2	10	0x00000000		
	0x00400032	0x02a00000	lw \$0,0x00000000(\$t1)	15:	lw	\$t0,0(\$t1) #load value of A[i] in \$t0	\$t3	11	0x00000000		
	0x00400036	0x02a0a000	add \$21,\$21,\$0	16:	add	\$s5,\$s5,\$t0 #sum=sum+A[i]	\$t4	12	0x00000000		
	0x00400040	0x02710820	ble \$s1,\$19,\$17	17:	bne	\$s1,\$s3,loop # if i <= n, goto loop	\$t5	13	0x00000000		
	0x00400044	0x10000000	lw \$1,0x00000000(\$t1)				\$t6	14	0x00000000		
	0x00400048	0x10000000	add \$1,0x00000000				\$t7	15	0x00000000		
	0x00400052	0x10000000	add \$1,0x00000000				\$s0	16	0x00000000		
	0x00400056	0x10000000	add \$1,0x00000000				\$s1	17	0x00000003		
	0x00400060	0x10000000	add \$1,0x00000000				\$s2	18	0x10010000		
	0x00400064	0x10000000	add \$1,0x00000000				\$s3	19	0x00000003		
	0x00400068	0x10000000	add \$1,0x00000000				\$s4	20	0x00000001		
	0x00400072	0x10000000	add \$1,0x00000000				\$s5	21	0xfffffffffc		
	0x00400076	0x10000000	add \$1,0x00000000				\$s6	22	0x00000000		

d) A[i] == 0

để đổi điều kiện vòng lặp thành sum A[i] == 0, ta thay đổi chương trình ban đầu thành như sau:

```
mips1.asm
1 .data
2     A: .word      1,2,3,4,0,5,6,7
3 .text
4     la    $s2,A
5     addi   $s5,$zero,0      #sum=0
6     addi   $s1,$zero,-1     #i=-1
7     addi   $s4,$zero,1      #step=1
8     addi   $s3,$zero,7      #n=7
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    beq   $t0,$zero,exit_loop  # if A[i] == 0, exit loop
16    add    $s5,$s5,$t0      #sum=sum+A[i]
17    j     loop
18 exit_loop:
```

Khi thực hiện chương trình với mảng A [] = {1,2,3,4,0,5,6,7} và n = 7

Ta được kết quả với tổng mới là $\$s5 = 0x0000000a$ hay 10 hệ thập phân

Text Segment		Basic		Source				Name		Number		Value	
Bkpt	Address	Code											
	0x00000000	0x3e011001	lui	\$1,0x00000100	5:	la	\$s2,A						
	0x00000004	0x34320000	ori	\$10,\$1,0x00000000									
	0x00000008	0x20150000	addi	\$21,\$0,0x00000000	6:	addi	\$s3,\$zero,0	#sum=0					
	0x0000000c	0x2011ffff	addi	\$17,\$0,0xffffffff	7:	addi	\$s1,\$zero,-1	#i=-1					
	0x00000010	0x20140001	addi	\$20,\$0,0x00000001	8:	addi	\$s4,\$zero,1	#step1					
	0x00000014	0x20130007	addi	\$19,\$0,0x00000007	9:	addi	\$s5,\$zero,7	#n=7					
	0x00000018	0x20348020	add	\$17,\$17,\$20	11:	add	\$t1,\$s1,\$s4	#i1+step					
	0x0000001c	0x02314820	add	\$5,\$17,\$17	12:	add	\$t1,\$s1,\$t1	#t1+=2*i					
	0x00000020	0x02156480	add	\$5,\$9,\$9	13:	add	\$t1,\$t1,\$t1	#t1+=4*i					
	0x00000024	0x02134820	add	\$5,\$9,\$18	14:	add	\$t1,\$t1,\$s2	#t1 store the address of A[i]					
	0x00000028	0x08d20000	lw	\$0,\$t0(\$t1)	15:	lw	\$t0,0(\$t1)	#load value of A[i] in t0					
	0x0000002c	0x21100000	breq	\$0,0x00000002	16:	breq	\$t0,\$zero,exit	loop # if A[i] == 0, exit loop					
	0x00000030	0x02a8a000	add	\$21,\$21,\$8	17:	add	\$s5,\$s5,\$t0	#sum+=A[i]					
	0x00000034	0x00010000	l	0x00000018	18:	l	loop						

Data Segment													
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)		Name	Number		
0x10010000	0x00000001	0x00000000	0x00000003	0x00000004	0x00000000	0x00000005	0x00000006	0x00000007		zero	0	0x00000000	
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fat	1	0x00100000	
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fv0	2	0x00000000	
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fvi	3	0x00000000	
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fa0	4	0x00000000	
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fa1	5	0x00000000	
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fa2	6	0x00000000	
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fa3	7	0x00000000	
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft0	8	0x00000000	
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft1	9	0x10010010	
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft2	10	0x00000000	
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft3	11	0x00000000	
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft4	12	0x00000000	
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft5	13	0x00000000	
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft6	14	0x00000000	
0x100101e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		ft7	15	0x00000000	
0x10010200	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs0	16	0x00000000	
0x10010220	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs1	17	0x00000000	
0x10010240	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs2	18	0x10010000	
0x10010260	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs3	19	0x00000000	
0x10010280	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs4	20	0x00000000	
0x100102a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs5	21	0x00000000	
0x100102c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000		fs6	22	0x00000000	

Assignment 6:

```

mips1.asm mips2.asm
1 .data |
2     A: .word 3, -5, 8, -12, 6
3 .text
4     la $s0, A                      # Load A to $s0
5     addi $s1, $zero, 0               # i = 0
6     addi $s2, $zero, 0               # Max = 0
7     addi $s3, $zero, 5               # n = 5
8 loop:
9      slt $t2, $s1, $s3             # $s2 = $s1 < $s3 ? 1 : 0
10     beq $t2, $zero, endloop
11     add $t1, $s1, $s1
12     add $t1, $t1, $t1             # $t1 = 4 * $s1
13     add $t1, $t1, $s0
14     lw $t0, 0($t1)                # $t0 = A[i]
15 if_left:
16     slt $t2, $t0, $zero           # $t2 = $t0 < $zero ? 1 : 0
17     beq $t2, $zero, if_else
18     sub $t0, $zero, $t0            # $t0 = -$t0
19     j if_else

20 if_else:
21     slt $t2, $t0, $s2             # $t2 = $t0 < $s2 ? 1 : 0
22     bne $t2, $zero, next
23     add $s2, $t0, $zero           # max = A[i]
24 next:
25     addi $s1, $s1, 1              # i += 1
26     j loop
27 endloop:

```

chương trình thực hiện lặp qua các phần tử của mảng để so sánh với giá trị lớn nhất hiện tại

thực hiện lấy ra các phần tử và kiểm tra giá trị có là số dương không

- Là số dương: thực hiện so sánh với giá trị lớn nhất hiện tại
- Là số âm: thực hiện chuyển đổi thành số dương bằng cách lấy 0 trừ đi giá trị của mảng được lấy ra. Thực hiện so sánh với giá trị lớn nhất hiện tại

Sau khi lặp qua các phần tử ta thu được giá trị lớn nhất trong mảng và lưu vào thanh ghi \$s2

Trong chương trình này, ta khởi tạo mảng A [] = {3, -5, 8, -12, 6}

Dễ thấy số có giá trị tuyệt đối lớn nhất trong dãy là -12

⇒ Thanh ghi \$s2 có giá trị bằng 0x0000000c hay 12 ở hệ thập phân