Computer Architecture Lab Report Week 3

Full name: On Quang Tung

Student ID: 20226096

Assignment 1

Ta khởi tạo x=10, y=10, z=10, và các biến i, j như sau:

- Trường hợp 1: i=3, j=4 (i<j)

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 3
        j: .word 4

.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=3
lw $s2, 0($t9) # j=4
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10

start:
slt $t0,$s2,$s1 # j<i
bne $t0,$zero,else # branch to else if j<i
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- Thanh ghi $at thay đổi mỗi khi gọi lệnh la
- Thanh ghi $t8, $t9 được ghi địa chỉ của 2 biến i, j
- Thanh ghi $s1, $s2 được ghi giá trị của biến i, j
- Thanh ghi $t1, $t2, $t3 được ghi giá trị của biến x, y, z (không khai báo trong code)
- Thanh ghi $t0 lưu kết quả của slt
- Thanh ghi $t1 tăng 1 đơn vị, $t3 được gán bằng 1
- Thanh ghi pc thay đổi sau mỗi lệnh
  ➔ Kết quả khớp với chương trình

- Trường hợp 2: i=4, j=3 (i>j)

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 4
        j: .word 3

.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=4
lw $s2, 0($t9) # j=3
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10

start:
slt $t0,$s2,$s1 # j<i
bne $t0,$zero,else # branch to else if j<i
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- Thanh ghi $at thay đổi mỗi khi gọi lệnh la
- Thanh ghi $t8, $t9 được ghi địa chỉ của 2 biến i, j
- Thanh ghi $s1, $s2 được ghi giá trị của biến i, j
- Thanh ghi $t1, $t2, $t3 được ghi giá trị của biến x, y, z (không khai báo trong code)
- Thanh ghi $t0 lưu kết quả của slt
- Thanh ghi $t2 giảm 1 đơn vị, $t3 được gán bằng chính nó nhân đôi
- Thanh ghi pc thay đổi sau mỗi lệnh
  ➔ Kết quả khớp với chương trình

- Trường hợp 3: i=4, j=4 (i=j)

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 4
        j: .word 4

.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=4
lw $s2, 0($t9) # j=3
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10

start:
slt $t0,$s2,$s1 # j<i
bne $t0,$zero,else # branch to else if j<i
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- Thanh ghi $at thay đổi mỗi khi gọi lệnh la
- Thanh ghi $t8, $t9 được ghi địa chỉ của 2 biến i, j
- Thanh ghi $s1, $s2 được ghi giá trị của biến i, j
- Thanh ghi $t1, $t2, $t3 được ghi giá trị của biến x, y, z (không khai báo trong code)
- Thanh ghi $t0 lưu kết quả của slt
- Thanh ghi $t1 tăng 1 đơn vị, $t3 được gán bằng 1
- Thanh ghi pc thay đổi sau mỗi lệnh

  ➔ Kết quả khớp với chương trình

Assignment 2

Note: i ~ $s1, A ~ $s2, n ~ $s3, step ~ $s4, sum ~ $s5

Ta khởi tạo i=0, n=5, step=1, sum=0 mảng A={1, 2, 3, 4, 5}

```
#Laboratory 3, Home Assigment 2
.data
A: .word 1 2 3 4 5
.text
addi $s1, $zero, 0 # i = 0
la $s2, A # load address of A into $s2
addi $s3, $zero, 5 # n = 5
addi $s4, $zero, 1 # step = 1
addi $s5, $zero, 0 # sum = 0

loop:
slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0
beq $t2, $zero, endloop
add $t1, $s1, $s1 # $t1 = 2 * $s1
add $t1, $t1, $t1 # $t1 = 4 * $s1
add $t1, $t1, $s2 # $t1 store the address of A[i]
lw $t0, 0($t1) # load value of A[i] in $t0
add $s5, $s5, $t0 # sum = sum + A[i]
add $s1, $s1, $s4 # i = i + step
j loop # goto loop
endloop:
```

- Thanh ghi $s1, $s2, $s3, $s4, $s5 lưu khởi tạo i, địa chỉ A, n, step, sum
- Thanh ghi $t1 lưu địa chỉ của A[i]
  (Ở đây dùng 2 lệnh add để có 4 * index thay vì lệnh mult do dùng lệnh mult sẽ tách ra số lượng lệnh nhiều hơn số lượng lệnh so với lệnh add ở trường hợp tệ nhất. Điều này dẫn đến tăng thời gian chạy)
- Thanh ghi $t0 lưu giá trị của A[i]
- Thanh ghi $s5 tăng giá trị bằng A[i] sau mỗi vòng lặp
- Thanh ghi $s1(i) tăng giá trị 1 lượng bằng $s4 (step)
- Thanh ghi pc thay đổi sau mỗi lệnh
  ➔ Kết quả sum = 15, đúng với chương trình

Assignment 3

Note: a ~ $s2, b ~ $s3

Ta khởi tạo a=3 và b=3

```
#Laboratory Exercise 3, Home Assignment 3
.data
test: .word 1

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

- Thanh ghi $s2, $s3 lưu giá trị khởi tạo của a, b
- Thanh ghi $s0 lưu địa chỉ của biến test
- Thanh ghi $s1 lưu giá trị của biến test
- Thanh ghi $t0, $t1, $t2 lưu các case để biến test có thể thực hiện so sánh
- Nếu:
    - Test = 0: $s2(a) tăng 1 đơn vị
    - Test = 1: $s2(a) giảm 1 đơn vị
    - Test = 2: $s3(b) gấp đôi ban đầu

  ➔ Kết quả chạy đúng với chương trình

Assignment 4

a. i < j

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 4
        j: .word 3
.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=4
lw $s2, 0($t9) # j=3
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10

start:
slt $t0,$s1,$s2 # i<j
bne $t0,$zero,else # branch to else if i >= j
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- TH1: i = 4, j = 3
  ➔ x($t1) = 11, y($t2) = 10 z($t3) = 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	0x0000000b
$t2	10	0x0000000a
$t3	11	0x00000001
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000004
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

- TH2: i = 2, j = 3
  ➔ x($t1) = 10, y($t2) = 9, z($t3) = 20(hex:14)

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	0x00000001
$t1	9	0x0000000a
$t2	10	0x00000009
$t3	11	0x00000014
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

b. i >=j

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 4
        j: .word 3

.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=4
lw $s2, 0($t9) # j=3
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10

start:
slt $t0,$s2,$s1 # j<i
bne $t0,$zero,else # branch to else if j >= i
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- TH1: i = 4, j = 3
  ➔ x($t1) = 10, y($t2) = 9, z($t3) = 20(hex:14)

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	0x00000001
$t1	9	0x0000000a
$t2	10	0x00000009
$t3	11	0x00000014
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000004
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

- TH2: i = 2, j = 3
  ➔ x($t1) = 11, y($t2) = 10 z($t3) = 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	0x0000000b
$t2	10	0x0000000a
$t3	11	0x00000001
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000002
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

c. i+j <= 0

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 4
        j: .word 3

.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=4
lw $s2, 0($t9) # j=3
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10
add $t7, $s1, $s2 # tổng của i và j

start:
slt $t0,$0,$t7 # 0 < i+j
bne $t0,$zero,else # branch to else if 0 >= i+j
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1,$t1,1 # then part: x=x+1
addi $t3,$zero,1 # z=1
j endif # skip “else” part
else: addi $t2,$t2,-1 # begin else part: y=y-1
add $t3,$t3,$t3 # z=2*z
endif:
```

- TH1: i = 4, j = 3
  ➔ x($t1) = 10, y($t2) = 9, z($t3) = 20(hex:14)

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	0x00000001
$t1	9	0x0000000a
$t2	10	0x00000009
$t3	11	0x00000014
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000007
$s0	16	0x00000000
$s1	17	0x00000004
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400044
hi		0x00000000
lo		0x00000000

- TH2: i = -4, j = -3
  ➔ x($t1) = 11, y($t2) = 10, z($t3) = 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	0x0000000b
$t2	10	0x0000000a
$t3	11	0x00000001
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0xfffffff9
$s0	16	0x00000000
$s1	17	0xfffffffc
$s2	18	0xfffffffd
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400044
hi		0x00000000
lo		0x00000000

d. i+j > m+n

```
#Laboratory Exercise 3, Home Assignment 1
.data
        i: .word 4
        j: .word 3

.text
la $t8, i
la $t9, j
lw $s1, 0($t8) # i=4
lw $s2, 0($t9) # j=3
addi $t1, $0, 10 # x=10
addi $t2, $0, 10 # y=10
addi $t3, $0, 10 # z=10
addi $t4, $0, 5 # m = 5
addi $t5, $0, 6 # n = 6
add $t6, $t4, $t5 # m+n = 5+6
add $t7, $s1, $s2 # tổng của i và j

start:
slt $t0, $t6, $t7 # 0 m+n< i+j
bne $t0, $zero, else # branch to else if m+n >= i+j
# kiểm tra xem j($s2) < i($s1), nếu không gán $t0=0
# nếu $t0 khác 0 rẽ sang else
addi $t1, $t1, 1 # then part: x=x+1
addi $t3, $zero, 1 # z=1
j endif # skip “else” part
else: addi $t2, $t2,-1 # begin else part: y=y-1
add $t3, $t3, $t3 # z=2*z
endif:
```

- TH1: i = 4, j = 3
  ➔ x($t1) = 11, y($t2) = 10, z($t3) = 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	0x0000000b
$t2	10	0x0000000a
$t3	11	0x00000001
$t4	12	0x00000005
$t5	13	0x00000006
$t6	14	0x0000000b
$t7	15	0x00000007
$s0	16	0x00000000
$s1	17	0x00000004
$s2	18	0x00000003
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400050
hi		0x00000000
lo		0x00000000

- TH2: i = 6, j = 7
  ➔ x($t1) = 10, y($t2) = 9, z($t3) = 20(hex:14)

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	0x00000001
$t1	9	0x0000000a
$t2	10	0x00000009
$t3	11	0x00000014
$t4	12	0x00000005
$t5	13	0x00000006
$t6	14	0x0000000b
$t7	15	0x0000000d
$s0	16	0x00000000
$s1	17	0x00000006
$s2	18	0x00000007
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x10010000
$t9	25	0x10010004
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400050
hi		0x00000000
lo		0x00000000

Assignment 5

a. i <= n

```
#Laboratory 3, Home Assigment 2
.data
A: .word 1, 2, 3, 4, 5

.text
addi $s1, $zero, 0 # i = 0
la $s2, A # load address of A into $s2
addi $s3, $zero, 5 # n = 5
addi $s4, $zero, 1 # step = 1
addi $s5, $zero, 0 # sum = 0

loop:
bge $s1, $s3, endloop #if i > n then endloop
add $t1, $s1, $s1 # $t1 = 2 * $s1
add $t1, $t1, $t1 # $t1 = 4 * $s1
add $t1, $t1, $s2 # $t1 store the address of A[i]
lw $t0, 0($t1) # load value of A[i] in $t0
add $s5, $s5, $t0 # sum = sum + A[i]
add $s1, $s1, $s4 # i = i + step
j loop # goto loop
endloop:
```

- TH1: A = {1, 2, 3, 4, 5}
  ➔ sum($s5) = 15

Registers | Coproc 1 | Coproc 0

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
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x0000000f
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

- TH2: A = {1, 2, 3, 4}
  ➔ sum($s5) = 10

Registers | Coproc 1 | Coproc 0

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	0x00000004
$t1	9	0x1001000c
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000004
$s2	18	0x10010000
$s3	19	0x00000004
$s4	20	0x00000001
$s5	21	0x0000000a
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040003c
hi		0x00000000
lo		0x00000000

b. sum >= 0

```
#Laboratory 3, Home Assigment 2
.data
A: .word -1, -2, -3, 4, 5

.text
addi $s1, $zero, 0 # i = 0
la $s2, A # load address of A into $s2
addi $s3, $zero, 5 # n = 5
addi $s4, $zero, 1 # step = 1
addi $s5, $zero, 0 # sum = 0

loop:
blt $s5, $0, endloop
add $t1, $s1, $s1 # $t1 = 2 * $s1
add $t1, $t1, $t1 # $t1 = 4 * $s1
add $t1, $t1, $s2 # $t1 store the address of A[i]
lw $t0, 0($t1) # load value of A[i] in $t0
add $s5, $s5, $t0 # sum = sum + A[i]
add $s1, $s1, $s4 # i = i + step
j loop # goto loop
endloop:
```

- TH1: A = {-1, -2, -3, 4, 5}
  ➔ sum($s5) = -1

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000001
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0xffffffff
$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	0x00000001
$s2	18	0x10010000
$s3	19	0x00000005
$s4	20	0x00000001
$s5	21	0xffffffff
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040003c
hi		0x00000000
lo		0x00000000

- TH2: A = {1, 2, 3, -4, -5}
  ➔ sum($s5) = -3

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000001
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0xfffffffb
$t1	9	0x10010010
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000005
$s2	18	0x10010000
$s3	19	0x00000005
$s4	20	0x00000001
$s5	21	0xfffffffd
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040003c
hi		0x00000000
lo		0x00000000

c. A[i] != 0

```
#Laboratory 3, Home Assigment 2
.data
A: .word 1 2 3 0 5

.text
addi $s1, $zero, 0 # i = 0
la $s2, A # load address of A into $s2
addi $s3, $zero, 5 # n = 5
addi $s4, $zero, 1 # step = 1
addi $s5, $zero, 0 # sum = 0

loop:
add $t1, $s1, $s1 # $t1 = 2 * $s1
add $t1, $t1, $t1 # $t1 = 4 * $s1
add $t1, $t1, $s2 # $t1 store the address of A[i]
lw $t0, 0($t1) # load value of A[i] in $t0
beq $t0, $0, endloop
add $s5, $s5, $t0 # sum = sum + A[i]
add $s1, $s1, $s4 # i = i + step
j loop # goto loop
endloop:
```

- TH1: A = {1, 2, 3, 0, 5}
  ➔ sum($s5) = 6

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	0x1001000c
$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	0x10010000
$s3	19	0x00000005
$s4	20	0x00000001
$s5	21	0x00000006
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400038
hi		0x00000000
lo		0x00000000

- TH2: A = {1, 0, 3, 0, 5}
  ➔ sum($s5) = 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	0x10010004
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000001
$s2	18	0x10010000
$s3	19	0x00000005
$s4	20	0x00000001
$s5	21	0x00000001
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400038
hi		0x00000000
lo		0x00000000

Assignment 6

```
.data
 A: .word -1, 2, -4, 5, -6, -7, 3

.text
  li $s1, 0 # i = 0
  la $s2, A # load address of A into $s2
  li $s3, 8 # n = 8
  li $s4, 1 # step = 1
  li $s5, 0 # max = 0
  li $s6, 0 # pos_max = 0

  loop:
  bge $s1, $s3, endloop
  add $t1, $s1, $s1 # $t1 = 2*i
  add $t1, $t1, $t1 # $t1 = 4*i
  add $t1, $t1, $s2 # $t1 store the address of A[i]
  lw $t0, 0($t1) # load the value of A[i]
  slt $t8, $0, $t0 # $t8 = 0 < A[i] ? 1 : 0
  beq $t8, $0, aBs

  positive:
  slt $t9, $s5, $t0 # $t9 = max < A[i] ? 1 : 0
  beq $t9, $0, continue
  add $s5, $t0, $0 # max = A[i]
  add $s6, $s1, $0 #pos_max = i

  aBs:
  sub $t0, $0, $t0 # A[i] = 0 - A[i]
  j, positive

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

  endloop:
```

- Thanh ghi $s5 lưu giá trị của số trị tuyệt đối lớn nhất
- Thanh ghi $s6 lưu index của số có trị tuyệt đối lớn nhất
- Các test case:
  - TH1: A={-1, 2, -4, 5, -6, -7, 3}
    ➔ Đáp án của bài toán với trường hợp này là A[5] = 7

Registers | Coproc 1 | Coproc 0

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	0x1001001c
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000008
$s2	18	0x10010000
$s3	19	0x00000008
$s4	20	0x00000001
$s5	21	0x00000007
$s6	22	0x00000005
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040005c
hi		0x00000000
lo		0x00000000

- TH2: A={2, 4, -1, 5, -3}
  ➔ Đáp án của bài toán với trường hợp này là A[3] = 5

Registers | Coproc 1 | Coproc 0

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	0x00000003
$t1	9	0x10010010
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000005
$s2	18	0x10010000
$s3	19	0x00000005
$s4	20	0x00000001
$s5	21	0x00000005
$s6	22	0x00000003
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040005c
hi		0x00000000
lo		0x00000000

- TH3: A={5, 4, -1, -3, 2, -6, -7, 8, 10, -9}
  ➔ Đáp án của bài toán với trường hợp này là A[8] = 10

Registers | Coproc 1 | Coproc 0

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	0x00000009
$t1	9	0x10010024
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x0000000a
$s2	18	0x10010000
$s3	19	0x0000000a
$s4	20	0x00000001
$s5	21	0x0000000a
$s6	22	0x00000008
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040005c
hi		0x00000000
lo		0x00000000

⇨ Chương trình chạy đúng với mong đợi