# Báo cáo thực hành KTMT tuần 3 Lê Quốc Đảng 20225801

#### Exercise 1

Đặt x, y, z = 10

i=5, j=6

```
ex1.asm ex2.asm ex3.asm temp.asm
           i: .word 5
          j: .word 6
           la $t5, i
    lw $s1, ($t5)
           la $t6, j
 9
           lw $s2, ($t6)
           li $t1, 10
10
          li $t2, 10
11
          li $t3, 10
12
13 start:
14
           slt $t0,$s2,$s1 # j<i
           bne $t0,$zero,else # branch to else if j<i
15
           addi $t1,$t1,1 # then part: x=x+1
16
17
           addi $t3,$zero,1 # z=1
18
           j endif # skip "else" part
19 else:
           addi $t2,$t2,-1 # begin else part: y=y-1
20
21
           add $t3,$t3,$t3 # z=2*z
22 endif:
```

Thanh ghi \$t0 chứ giá trị của câu lệnh slt có giá trị bằng 0.

Các thanh ghi \$t1 thay đổi từ 0x0000000a sang 0x0000000b

\$t2 thay đổi từ 0x0000000a sang 0x00000001

Thanh ghi pc tăng 0x00000004 sau mỗi lệnh

⇒ Kết quả thực thi đúng với lí thuyết.

#### **Exercise 2**

Mång A =  $\{1,2,3,4,5,6,7,8,9,10\}$ 

TH1: n=5

```
ex1.asm ex2.asm* ex3.asm temp.asm
           A: .word 1,2,3,4,5,6,7,8,9,10
           addi $s1,$zero, 0 #i=0
           la $s2,A #truy cap vao mang
          addi $s3,$zero,5 #n
6
          addi $s4,$zero,1 #step=1
7
          addi $s5,$zero,0 #sum=0
8
9 100p:
           add $s1,$s1,$s4 #i=i+step
10
          add $t1,$s1,$s1 #t1=2*s1
11
12
          add $t1,$t1,$t1 #t1=4*s1
13
          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]
15
          bne $s1,$s3,loop #if i != n, goto loop
16
```

Trong vòng lặp đầu tiên:

Thanh ghi \$s1 thay đổi là 0x00000001

Thanh ghi \$t1 lấy địa chỉ của A[1] là 0x10010004

Thanh ghi \$t0 chứa giá trị A[1] là 0x00000002

Thanh ghi \$s5 thay đổi là 0x00000002

Các vòng lặp sau tương tự. Sau 5 vòng lặp, \$s1 có giá trị là 5, bằng với n nên kết thúc vòng lặp

Sau vòng lặp, thanh ghi \$s5 bằng 0x00000014

⇒ Kết quả đúng

## TH2: n=7

```
ex1.asm ex2.asm* ex3.asm temp.asm
           A: .word 1,2,3,4,5,6,7,8,9,10
           addi $s1,$zero, 0 #i=0
5
           la $s2, A #truy cap vao mang
       addi $s3,$zero,7 #n
          addi $s4,$zero,1 #step=1
8
           addi $s5,$zero,0 #sum=0
           add $s1,$s1,$s4 #i=i+step
10
           add $t1,$s1,$s1 #t1=2*s1
11
12
13
14
          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]
15
          bne $s1,$s3,loop #if i != n, goto loop
```

Tương tự như n=5, chương trình thực hiện 7 vòng lặp

Sau vòng lặp \$s5 có giá trị là 0x00000023

⇒ Kết quả đúng

### Exercise 3

Đặt a=5, b=6

TH1: test=0, sau chương trình a=0x00000006

```
ex1.asm ex2.asm ex3.asm* temp.asm
 1 #Laboratory Exercise 3, Home Assignment 3
         test: .word 0
3
4 .text
5
          li $s2, 5
          li $s3, 6
6
          la $s0, test #load the address of test variable
7
          lw $s1, 0($s0) #load the value of test to register $t1
8
9
          li $t0, 0 #load value for test case
10
          li $t1, 1
          li $t2, 2
11
          beq $s1, $t0, case_0
12
          beq $s1, $t1, case_1
13
14
15
          beq $s1, $t2, case_2
          j default
16 case_0: addi $s2, $s2, 1 #a=a+1
17 j continue
18 case_1: sub $s2, $s2, $t1 #a=a-1
19 j continue
20 case_2: add $s3, $s3, $s3 #b=2*b
21 j continue
22 default:
23 continue:
```

## TH2: test=1, sau chương trình a=0x00000004

```
ex1.asm ex2.asm ex3.asm temp.asm
1 #Laboratory Exercise 3, Home Assignment 3
2 .data
          test: .word 1
4 .text
5
          li $s2, 5
          li $s3, 6
6
         la $s0, test #load the address of test variable
8
          lw $s1, 0($s0) #load the value of test to register $t1
          li $t0, 0 #load value for test case
    li $t1, 1
10
          li $t2, 2
11
12
          beq $s1, $t0, case_0
13
          beq $s1, $t1, case_1
         beq $s1, $t2, case_2
14
15
          j default
16 case_0: addi $s2, $s2, 1 #a=a+1
17 j continue
18 case_1: sub $s2, $s2, $t1 #a=a-1
19 j continue
20 case_2: add $s3, $s3, $s3 #b=2*b
21 j continue
22 default:
23 continue:
```

TH3: test=2, sau chương trình b=0x0000000c

```
ex1.asm ex2.asm ex3.asm temp.asm
 1 #Laboratory Exercise 3, Home Assignment 3
          test: .word 2
3
4 .text
5
           li $s2, 5
          li $s3, 6
6
          la $s0, test #load the address of test variable
7
          lw $s1, 0($s0) #load the value of test to register $t1
8
9
          li $t0, 0 #load value for test case
10
          li $t1, 1
          li $t2, 2
11
          beq $s1, $t0, case_0
12
13
          beq $s1, $t1, case_1
14
          beq $s1, $t2, case_2
          j default
15
16 case_0: addi $s2, $s2, 1 #a=a+1
17 j continue
18 case_1: sub $s2, $s2, $t1 #a=a-1
19 j continue
20 case_2: add $s3, $s3, $s3 #b=2*b
21 j continue
22 default:
23 continue:
```

⇒ Kết quả đúng

#### **Exercise 4**

a. i<j

```
mips2.asm*
1 .data
           i: .word 5
           j: .word 6
3
4 .text
           la $t5, i
5
          lw $s1, ($t5)
7
           la $t6, j
           lw $s2, ($t6)
8
          li $t1, 10
9
          li $t2, 10
10
          li $t3, 10
12 start:
           slt $t0,$s1,$s2 # i<j
13
           beq t0,\zero,else # branch to else if i \ge j
14
15
           addi $t1,$t1,1 # then part: x=x+1
16
           addi $t3,$zero,1 # z=1
          j endif # skip "else" part
17
18 else:
           addi $t2,$t2,-1  # begin else part: y=y-1
19
20
           add $t3,$t3,$t3 # z=2*z
21 endif:
22
```

b. i >= j

```
mips2.asm
1 .data
           i: .word 5
2
          j: .word 6
3
4 .text
5
           la $t5, i
           lw $s1, ($t5)
6
          la $t6, j
7
          lw $s2, ($t6)
9
          li $t1, 10
          li $t2, 10
10
          li $t3, 10
11
12 start:
13
           slt $t0,$s1,$s2 # i<j
           bne $t0,$zero,else # branch to else if i<j
14
          addi $t1,$t1,1 # then part: x=x+1
15
           addi $t3,$zero,1 # z=1
16
17
          j endif # skip "else" part
18 else:
           addi $t2,$t2,-1 # begin else part: y=y-1
19
           add $t3,$t3,$t3 # z=2*z
20
21 endif:
22
```

c.  $i+j \le 0$ 

```
mips2.asm
1 .data
           i: .word 5
2
           j: .word 6
4 .text
           la $t5, i
5
6
           lw $s1, ($t5)
           la $t6, j
           lw $s2, ($t6)
8
9
           li $t1, 10
           li $t2, 10
10
           li $t3, 10
12
           add $t7, $s1, $s2
13 start:
           sgt $t0,$t7,$zero # i+j>0
14
           bne $t0,$zero,else # branch to else if <math>i+j>0
15
16
           addi $t1,$t1,1 # then part: x=x+1
           addi $t3,$zero,1 # z=1
17
           j endif # skip "else" part
18
19 else:
           addi $t2,$t2,-1 # begin else part: y=y-1
20
21
           add $t3,$t3,$t3 # z=2*z
22 endif:
23
```

d. i+j>m+n

```
mips2.asm*
 1 .data
 2
           i: .word 5
           j: .word 6
 4 .text
 5
           la $t5, i
           lw $s1, ($t5)
 6
           la $t6, j
 7
 8
           lw $s2, ($t6)
           li $t1, 10
 9
           li $t2, 10
10
           li $t3, 10
11
12
           add $t7, $s1, $s2
13
           addi $t8, $zero, 7 #m
           addi $t9, $zero, 8 #n
14
           add $t4, $t8, $t9 #m+n
15
16 start:
17
           sgt $t0,$t7,$t4 # i+j>m+n
18
           beq $t0,$zero,else # branch to else if i+j<=m+n
           addi $t1,$t1,1 # then part: x=x+1
19
20
           addi $t3,$zero,1 # z=1
21
22 else:
           j endif # skip "else" part
23
           addi $t2,$t2,-1 # begin else part: y=y-1
24
           add $t3,$t3,$t3 # z=2*z
25 endif:
26
4
```

#### **Exercise 5**

a. i<n

```
ex2.asm* ex3.asm temp.asm
 ex1.asm
1 .data
           A: .word 1,2,3,4,5,6
2
3 .text
4
           addi $s1,$zero, 0 \#i=0
5
           la $s2, A #truy cap vao mang
           addi $s3,$zero,4 #n
6
           addi $s4,$zero,1 #step=1
7
           addi $s5,$zero,0 #sum=0
9 100p:
           add $s1,$s1,$s4 #i=i+step
10
           add $t1,$s1,$s1 #t1=2*s1
11
           add $t1,$t1,$t1 #t1=4*s1
12
13
           add $t1,$t1,$s2 #t1 store the address of A[i]
14
           lw \$t0,0(\$t1) #load value of A[i] in \$t0
          add $s5,$s5,$t0 #sum=sum+A[i]
15
           slt $t2,$s1,$s3 #i<n
16
17
           bne t_2, e_n, loop #if i < n, goto loop
18
```

b.  $i \le n$ 

```
ex2.asm* ex3.asm temp.asm*
 ex1.asm
1 .data
          A: .word 1,2,3,4,5,6
3 .text
4
          addi $s1,$zero, 0 #i=0
          la $s2, A #truy cap vao mang
5
          addi $s3,$zero,4 #n
6
          addi $s4,$zero,1 #step=1
7
          9 loop:
          add $s1,$s1,$s4 #i=i+step
10
          add $t1,$s1,$s1 #t1=2*s1
11
12
          add $t1,$t1,$t1 #t1=4*s1
13
          add $t1,$t1,$s2 #t1 store the address of A[i]
          lw $t0,0($t1) #load value of A[i] in $t0
14
          add $s5,$s5,$t0 #sum=sum+A[i]
15
        sgt $t2,$s1,$s3 #i>n
17
          beq $t2,$zero,loop #if i<=n, goto loop
18
```

c. sum>=0, Khởi tạo mảng  $A = \{1,2,3,-6,8\}$ 

```
ex1.asm ex2.asm* ex3.asm temp.asm
1 .data
2
3 .text
           A: .word 1,2,3,-6,8
           addi $s1,$zero, 0 #i=0
           la $s2,A #truy cap vao mang
           addi $s3,$zero,4 #n
6
7
           addi $s4,$zero,1 #step=1
8
           addi $s5,$zero,0 #sum=0
           add $s1,$s1,$s4 #i=i+step
10
           add $t1,$s1,$s1 #t1=2*s1
11
           add $t1,$t1,$t1 #t1=4*s1
12
13
           add $t1,$t1,$s2 #t1 store the address of A[i]
           lw $t0,0($t1) #load value of A[i] in $t0
14
15
           add $s5,$s5,$t0 #sum=sum+A[i]
           slt $t2,$s5,$zero #sum<0
16
17
           beq $t2,$zero,loop #if sum>=0, goto loop
```

d. A[i]==0, Khởi tạo mảng  $A = \{1,0,2,3,4\}$ 

```
ex1.asm ex2.asm* ex3.asm temp.asm*
1 .data
          A: .word 1,0,2,3,4
2
3 .text
4
          addi $s1,$zero, 0 #i=0
5
          la $s2, A #truy cap vao mang
6
          addi $s3.$zero.4 #n
          addi $s4,$zero,1 #step=1
          9 loop:
          add $s1,$s1,$s4 #i=i+step
10
          add $t1.$s1.$s1 #t1=2*s1
11
12
          add $t1,$t1,$t1 #t1=4*s1
          add $t1,$t1,$s2 #t1 store the address of A[i]
          lw $t0,0($t1) #load value of A[i] in $t0
14
          add Ss5.Ss5.St0 #sum=sum+A[i]
15
          beq $t0,$zero,loop #if A[i]==0, goto loop
17
```

#### Exercise 6

Khởi tạo mảng A={1,-2,-5,3}

```
ex2.asm ex3.asm temp.asm*
 ex1.asm
 1 .data
           A: .word 1,-2,-5,3
           addi $s1,$zero, 0 #i=0
           la $s2,A #truy cap vao mang
5
           addi $s3,$zero,4 #so phan tu cua mang
 6
           addi $s4,$zero,1 #step=1
           addi $s5,$zero,0 \#max=0
           addi $s6,$zero,0 #index max
10 loop:
           add $t1,$s1,$s1 #t1=2*s1
11
12
           add $t1,$t1,$t1 #t1=4*s1
13
           add $t1,$t1,$s2 #t1 store the address of A[i]
           lw $t0,0($t1) #load value of A[i] in $t0
14
           slt $t2, $t0, $zero #A[i]<0
15
          bne $t2, $zero, duong
           beq $t2, $zero, check
18 duong: #lay tri tuyet doi
          sub $t0,$zero,$t0 #A[i]=|A[i]|
19
20
           j check
21 check: #kiem tra xem max<|A[i]|
          slt $t3,$s5,$t0 #max<|A[i]|
22
23
           bne $t3,$zero,max
          beq $t3,$zero,cont
24
25 max: #thay doi gia tri max
26
           add $s5,$t0,$zero
27
           add $s6,$s1,$zero #index_max=i
```

```
25 max: #thay doi gia tri max
26 add $s5,$t0,$zero
27 add $s6,$s1,$zero #index_max=i
28 j cont
29 cont:
30 add $s1,$s1,$s4 #i=i+step
31 slt $t4,$s1,$s3 #i<n
bne $t4,$zero, loop
33 end:
34
```

Kết quả thu được là max = 5,  $index_max = 2$