**­­­­Computer Architecture Lab Report Week 3**

**Full name: Nguyễn Hồng Phúc**

**Student ID: 20225659**

Assignment 1Create a new project to implement the code in Home Assignment 1. Initialize for  
i and j variable. Compile and upload to the simulator. Run this program step by  
step, observe the changing of memory and the content of registers at each step.

Đặt x =4, y=3, z=2.

**Trường hợp 1: I < J** (theo lý thuyết sau khi chạy xong lệnh, x=5, y=3, z=1)

.data

I: .word 2

J: .word 3

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4 #gán x=4

addi $t2, $zero, 3 #gán y=3

addi $t3, $zero, 2 #gán z=2

start:

slt $t0, $s2, $s1 # So sánh nếu j<i, thanh ghi $s2 chứa giá trị j, thanh ghi $s1 chứa giá trị i

bne $t0,$zero,else # branch to else if j<i nếu $t0=1

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else”

else:

addi $t2,$t2,-1 # y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

=> Các thanh ghi như slt thay đổi có giá trị là 0, các thanh ghi chứa  
giá trị x, y, z thay đổi, thanh ghi pc thay đổi.  
=> Sau khi slt so sánh j<i thì $t0 có giá trị là 0, sau lệnh bne so sánh $t0 với $zero máy tính sẽ thực hiện dòng lệnh ngay sau đấy mà không nhảy đến nhãn else, x=x+1=5, z=1 sau đấy lệnh jump nhảy đến endif để skip part “else”. Kết quả thực thi cho ra kết quả x, y, z thay đổi đúng như lý thuyết.

**Trường hợp 2: I>J** (theo lý thuyết x=4, y=2, z=4)

.data

I: .word 3

J: .word 2

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4 #gán x=4

addi $t2, $zero, 3 #gán y=3

addi $t3, $zero, 2 #gán z=2

start:

slt $t0, $s2, $s1 # So sánh nếu j<i, thanh ghi $s2 chứa giá trị j, thanh ghi $s1 chứa giá trị i

bne $t0,$zero,else # branch to else if j<i

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else”

else:

addi $t2,$t2,-1 # y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

=> Các thanh ghi như slt thay đổi có giá trị là 0, các thanh ghi chứa  
giá trị x, y, z thay đổi, thanh ghi pc thay đổi.  
=>Sau khi slt so sánh j<i thì $t0 có giá trị là 1, sau lệnh bne so sánh $t0 với $zero máy tính sẽ nhảy đến nhãn else, ở đây sau khi máy tính thực hiện các dòng lệnh y=y-1=3 và z=2\*z=4. Kết quả thực thi cho ra kết quả x, y, z thay đổi đúng như lý thuyết.

**Trường hợp 3: I =J** (theo lý thuyết sau khi chạy xong lệnh, x=5, y=3, z=1)

.data

I: .word 3

J: .word 3

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4 #gán x=4

addi $t2, $zero, 3 #gán y=3

addi $t3, $zero, 2 #gán z=2

start:

slt $t0, $s2, $s1 # So sánh nếu j<i, thanh ghi $s2 chứa giá trị j, thanh ghi $s1 chứa giá trị i

bne $t0,$zero,else # branch to else if j<i nếu $t0=1

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else”

else:

addi $t2,$t2,-1 # y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

=> Các thanh ghi như slt thay đổi có giá trị là 0, các thanh ghi chứa  
giá trị x, y, z thay đổi, thanh ghi pc thay đổi.  
=> Sau khi slt so sánh j<i thì $t0 có giá trị là 0, sau lệnh bne so sánh $t0 với $zero máy tính sẽ thực hiện dòng lệnh ngay sau đấy mà không nhảy đến nhãn else, x=x+1=5, z=1 sau đấy lệnh jump nhảy đến endif để skip part “else”. Kết quả thực thi cho ra kết quả x, y, z thay đổi đúng như lý thuyết.

Assignment 2Create a new project implementing the code in Home Assignment 2. Initialize for  
i, n, step, sum variables and array A. Compile and upload to the simulator. Run  
this program step by step, observe the changing of memory and the content of  
registers by each step. Try to test with some more cases (change the value of  
variables).

.data

A: .word 1,2,3,4,5,6,7,8,9

.text

addi $s3, $zero, 9 #tao so phan tu cua mang

addi $s4, $zero, 1 #tao buoc nhay

la $s2, A #Truy nhap thanh ghi $s2 vao dia chi A

addi $s6, $zero, 0 #sum = 0

addi $s1, $zero, 0 #i = 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 $s6, $s6, $t0 # sum = sum + A[i]

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

j loop # goto loop

endloop:

* Tổng được lưu ở thanh ghi $s6
* Khi thực thi xong, giá trị của thanh ghi $s6 là 0x0000002d(45)
* Kết quả trên đúng với lý thuyết

Assignment 3Create a new project implementing the code in Home Assignment 3. Compile and  
upload to the simulator. Run this program step by step; observe the changing of  
memory and the content of registers by each step. Change the value of test variable  
and run this program some times to check all cases.

Đặt a = b = 6. Theo lý thuyết:  
Nếu test = 0, a = 0x00000007  
Nếu test = 1, a = 0x00000005  
Nếu test = 2, b = 0x0000000c  
Code:

**Trường hợp test =1**  
.data  
test: .word 1  
.text  
addi $s2,$zero,6  
addi $s3,$zero,6  
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:

=> Kết quả đúng với lý thuyết a=0x00000005.

=>Lần lượt thay test=0 và test=2 cũng cho ra các giá trị của a,b đúng với lý thuyết.

Assignment 4Modify the Assignment 1, so that the condition tested is  
a. i < j  
b. i >= j  
c. i+j <= 0  
d. i+j > m+n (m and n stored in other registers)

**a. i < j**

.data

I: .word 3

J: .word 5

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4 #x=4

addi $t2, $zero, 3 #y=3

addi $t3, $zero, 2 #z=2

start:

slt $t0,$s2,$s1 # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i

bne $t0,$zero,else # branch to else if j<i

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else”

else:

addi $t2,$t2,-1 # y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

**b. i >= j**

.data

I: .word 5

J: .word 3

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4 #x=4

addi $t2, $zero, 3 #y=3

addi $t3, $zero, 2 #z=2

start:

slt $t0,$s2,$s1 # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i

bne $t0,$zero,else # branch to else if j<i

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else”

else:

addi $t2,$t2,-1 # y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

**c. i+j<=0**

**Trường hợp i+j<=0**

.data

I: .word -3

J: .word -1

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4

addi $t2, $zero, 4

addi $t3, $zero, 4

addu $t7, $s1, $s2 #Thanh $t7 la tong cua I va J

start:

slt $t0,$zero,$t7 # So sanh 0<I+J ? 1 : 0

bne $t0,$zero,else # branch to else if

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:

=>Kết quả cho ra $t1=0x00000005 và $t3=0x00000001 đúng với lý thuyết

**Trường hợp i+j>0**

.data

I: .word 3

J: .word 1

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4

addi $t2, $zero, 4

addi $t3, $zero, 4

addu $t7, $s1, $s2 #Thanh $t7 la tong cua I va J

start:

slt $t0,$zero,$t7 # So sanh 0<I+J ? 1 : 0

bne $t0,$zero,else # branch to else if

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:

=>Kết quả cho ra $t2=0x00000003 và $t3=0x00000008 đúng với lý thuyết

**d. i+j>m+n**

.data

I: .word 3

J: .word 3

.text

la $t8, I

la $t9, J

lw $s1, 0($t8)

lw $s2, 0($t9)

addi $t1, $zero, 4

addi $t2, $zero, 4

addi $t3, $zero, 4

addi $t4, $zero, 8 #m

addi $t5,$zero,7 #n

add $t6,$t4,$t5 #m+n

add $t7,$s1,$s2 #i+j

start:

slt $t0,$t6,$t7

addi $s4,$zero,1

bne $t0,$s4,else # branch to else if i+j<=m+n

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:

Assignment 5Modify the Assignment 2, so that the condition tested to keep the loop running is  
a. i <= n  
b. sum >= 0  
c. A[i] != 0

a. i <= n

.data

A: .word 1,2,3,4,5,6,7,8,9

.text

addi $s3, $zero, 9 #tao so phan tu cua mang

addi $s4, $zero, 1 #tao buoc nhay

la $s2, A #Truy nhap thanh ghi $s2 vao dia chi A

addi $s6, $zero, 0 #sum = 0

addi $s1, $zero, 0 #i = 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 $s6, $s6, $t0 # sum = sum + A[i]

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

j loop # goto loop

endloop:

b. sum >= 0

.data

A: .word 1,2,3,4,5,-6,7,8,-9,10

.text

addi $s3, $zero, 10

addi $s4, $zero, 1

la $s2,A

addi $s5, $zero, 0 #sum = 0

addi $s1, $zero, 0 #i = 0

loop:

bltz $s5, endloop # endloop if sum < 0

add $t1,$s1,$s1 #$t1=2\*$s1add $t1,$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:

c. A[i] != 0

.data  
A: .word 1,2,3,0,5,6,7,8,9,10  
.text  
addi $s3, $zero, 10  
addi $s4, $zero, 1  
la $s2,A  
addi $s5, $zero, 0 #sum = 0  
addi $s1, $zero, 0 #i = 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  
beqz $t0, endloop # endloop if A[i] == 0  
add $s5,$s5,$t0 #sum=sum+A[i]  
add $s1,$s1,$s4 #i=i+step  
j loop #goto loop  
endloop:

Assignment 6Using all of above instructions and statements, create a new project to implement  
this function: find the element with the largest absolute value in a list of integers.  
Assuming that this list is store in an integer array and we know the number of  
elements in it

Code:

.data

A: .word 1,8,3,-4,5,6,7,-9,-5,10

.text

addi $s3, $zero, 10 #tao so phan tu cua mang

addi $s4, $zero, 1 #tao buoc nhay

la $s2, A #Truy nhap thanh ghi $s2 vao dia chi A

addi $s5, $zero, 0 #gt = 0

addi $s6, $zero, 0 #dia chi = 0

addi $s1, $zero, 0 #i = 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

slt $t4, $t0, $zero #Kiem tra tri tuyet doi

beq $t4, $zero, duong

sub $t0,$zero,$t0 #Thuc hien khi A[i] la so am

duong:

slt $t5, $s5, $t0 #$s5 chua gia tri max

beq $t5, $zero, sai #Neu |a[i]| <= max thi nhay qua nhan “sai”

add $s5,$zero,$t0 #max = |a[i||

add $s6,$zero,$s1 #Cap nhat vi tri moi

j cont

sai:

addi $s5, $s5, 0

addi $s6, $s6, 0

j cont

cont:

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

j loop #goto loop

endloop:

TH1: A: .word 1,8,3,-4,5,6,7,-9,-5,10

A screenshot of a computer

Description automatically generated

TH2: A: .word 1,-11,3,-4,5,6,7,-9,-5,10

A screenshot of a computer

Description automatically generated