**Hanoi University of Science and Technology  
School of Information and Communication Technology**



**LABORATORY REPORT**

**Subject: Assembly Language and Computer Architecture Lab**

***Lesson No. 03***

**Instructor: Dr. Do Cong Thuan**

**Student name: Nguyen Thi Minh Chau**

**Student ID: 20184238**

**Class: ICT 01 – K63**

**Hanoi, 2021**

**Exercise 1:**

start:

li $s1, 2

li $s2, 3

slt $t0,$s2,$s1

bne $t0,$zero, else

addi $t1,$t1,1

addi $t3,$zero,1

j endif

else:

addi $t2,$t2,-1

add $t3,$t3,$t3

endif:

Result:

|  |  |
| --- | --- |
| ***Instruction*** | ***Observation & Explanation*** |
| 1 | $s1 register has value 2 |
| 2 | $s2 register has value 3 |
| 3 | Compare value of 2 register $s1 and $s2. Since $s2 is not less than $s1, $t0 value is set to 0 |
| 4 | Compare value of $t0 with $zero. Since $t0 is 0, branch else is not taken, the next instruction is delivered |
| 5 | Addi instruction is delivered: $t1 value is increased by 1 |
| 6 | Addi instruction is delivered: $t3 value is set to be sum of $zero and 1 |
| 7 | Jump unconditionally to label endif. The program stopped here |

**Exercise 2:**

.data

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

N: .word 6

.text

lw $s3, N

la $t1, A

li $s1, -1

li $s4, 1

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 $t0,A($t1) #load value of A[i] in $t0

add $s5,$s5,$t0 #sum=sum+A[i]

bne $s1,$s3,loop #if i != n, goto loop

Result:

* The .text field initialize an array A[3, 7, 6, 9, 8, 2, 5] and N=6 for the loop counter (since A has 7 elements, starts with index 0)

|  |  |
| --- | --- |
| ***Instruction*** | ***Observation & Explanation*** |
| 1 | Load value of N into register $s3 |
| 2 | Load address of A (A[0]) into register $t1 |
| 3 | Set value of $s1 (counter) to -1 (since the loop will add a step to the counter at first) |
| 4 | Set value of $s4 (step) to 1 |
| Loop |  |
| 5 | Increase the counter $s1 by 1 |
| 6 | Double the value of $s1 and store it in $t1 |
| 7 | Double the value of $t1 (which is now 4 times of $s1) |
| 8 | $t1 now store the address of A[counter] |
| 9 | Load value of A[counter] into $t0: since $t1 now have increase offset, and A has the base offset of the array (address of A[0]), the value loaded is at (address of A[0] + increase offset in $t1) |
| 10 | Add value of A[counter] in $t0 into $s5 |
| 11 | Compare $s1 and $s3: If $s1 is not equal to $s3 (index 6, which indicates the last value of the array), then go back to loop |

**Exercise 3:**

Result:

* The .text field initialize test variable to be 1

|  |  |
| --- | --- |
| ***Instruction*** | ***Observation & Explanation*** |
| 1 | Load address of test variable to register $s0 |
| 2 | Load value at address in $s0 and store to $s1 |
| 3+4+5 | Load value 0, 1, 2 into register $t0, $t1, $t2 |
| 6+7+8+9 | If $s1 equal to $t0, go to label case\_0  If $s1 equal to $t1, go to label case\_1  If $s1 equal to $t2, go to label case\_2  Else, jump unconditionally to default label  In this case, case\_1 will be taken |
| case\_1 |  |
| 10+11 | $s2 will be decreased by 1, then jump unconditionally to label continue (end of program, which is equal to break in switch-case loop) |

**Exercise 4:**

To to change the condition of the program, we replace 2 instructions:

slt $t0,$s2,$s1

bne $t0,$zero, else

1. Change to bge $s1, $s2, else (branch to else if i >= j)
2. Change to blt $s1, $s2, else (branch to else if i < j)
3. Change to add $t5, $s1, $s2 #store sum of $s1 and $s2 to $t5  
    bgt $t5, $zero, else #branch to else if i+j > 0
4. Change to li $t5, 1 #store value of m, suppose m=1  
    li $t6, 2 #store value of n, suppose n=2  
    add $t7, $t5, $t6 #store sum of m and n to $t7  
    add $t8, $s1, $s2 #store sum of $s1 and $s2 to $t8   
    ble $t8, $t7, else #branch to else if i+j <= m+n

**Exercise 5:** We change the condition by replacing bne $s1, $s3, loop by

* 1. blt $s1, $s3, loop (branch to loop if i < n)
  2. ble $s1, $s3, loop (branch to loop if i <= n)
  3. bge $s5, $zero, loop (branch to loop if sum >= 0)
  4. beq $t0, $zero, loop (branch to loop if A[i] == 0)

**Exercise 6:** The file is included in attachment

**Quiz:**

- Branch instructions affect pc registers: if the branch is taken, the value in pc register will not increase as usual but with a (significantly) larger value than 0x4, or can decrease significantly. This is because the address of the next instruction to be delivered is not in the same straight flow of program.

- Other instructions set than MIPS may have other register affected, eg. Flag registers of Intel.