Lab 3

Load/Store, Jump & Branch instructions

Goals

- Know how to use load/store, jump and branch instructions.
- Know how to implement high level programming language structures such as **if-then-else**, **for-loop** and **switch-case** statements in assembly.

Preparation

Before doing this lab, you should review your lectures and textbooks in computer architecture.

Inlab Assignments

Sample Program 1

For an ***if-then-else*** statement in C:

```
if (i<=j)
        x=x+1;
        z=1;
else
        y=y-1;
        z=2*z;
```

The following assembly program implements the aforementioned C statement using basic instructions such as **slt**, **addi**, **j** (jump) and **bne** (branch).

```
#Laboratory Exercise 3, Sample Program 1
start:
      slt   $t0,$s2,$s1       # j < i ?
      bne   $t0,$zero,else
      addi  $t1,$t1,1
      addi  $t3,$zero,1
      j     endif
else: addi  $t2,$t2,-1
      add   $t3,$t3,$t3
endif:
```

- Create a new project in MARS to execute the Sample Program 1.
- Initialize the variables *i* and *j*.
- Determine the function of each code line.
- Observe the change in memory and registers.

Sample Program 2

The pseudocode shows how to compute the sum of *n* elements of the array A.

```
loop:  i = i + step;
       sum = sum + A[i];
       if (i != n) goto loop;
```

The following assembly program performs the action of the aforementioned pseudocode.

```
#Laboratory 3, Sample Program 2
.text
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 stores the address of A[i]
      lw    $t0,0($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
```

- Create a new project in MARS to execute the Sample Program 2.
- Assume the *index i*, the *starting address of A*, the *number of elements in A*, the *step*, and the *sum* are stored in $s1, $s2, $s3, $s4 and $s5, respectively.
- Initialize the variables *i*, *n*, *step*, *sum* and the elements of *A*.
- Determine the function of each code line.
- Observe the change memory and registers.

Sample Progam 3

A **switch-case** statement is used to change the control flow of program execution depending on the value of a variable or expression.
In the following pseudocode, the variable *test* can be 0, 1, or 2. For each value of the valuable *test*, the pseudocode performs a different action.

switch(test) {
case 0:
a=a+1; break;
case 1:
a=a-1; break;
case 2:
*b=2*b; break;*
}

The following assembly program performs the action of the aforementioned pseudocode. Assume that **a** and **b** are stored in $s2 and $s3.

```
#Laboratory Exercise 3, Sample Program 3
.data
test: .word 1
.text
            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:
```

- Create a new project in MARS to execute the Sample Program 13.
- Change the value of the variable *test* to check if all the cases of the **switch-case** statement work correctly.
- Determine the function of each code line.
- Observe the change in the memory and registers.

Assignment 1

Write an assembly program to implement an **if-then-else** statement. Assume the test expression is:

a. i <= j
b. i+j >= 0
c. i+j > m+n
d. (i <= j) AND (i+j >= 0)

Assignment 2

Write an assembly program to implement a **while-loop** statement that computes the sum of *n* elements in an array A (the number of elements of A is greater than *n*). Assume the test condition at the end of the loop is:

a. i <= n
b. sum >= 100
c. A[i] % 2 = 0
d. sum >= 100 OR A[i] >= 50

Assignment 3

Write an assembly program to find **three elements** of largest absolute value in an array of *n* elements. Assume $20 \leq n \leq 30$.

Assignment 4

Which registers are affected by jump/branch instructions? Refer to the Sample Programs 1-3, explain how the target addresses of the instructions ***j***, ***bne***, and ***beq*** are calculated.