Objective

1. Implementing IF_ELSE Statement In Assembly Language
2. Discuss more Instruction Sets with more interesting programs.
Background

In previous lab, you have been introduced to some basic instructions of Intel Instruction Set and few small programs on how to use those commands. In today’s lab, we will explore more Instruction Set with some simple tricky problems.

Jump instructions are used to alter the execution path of instruction in your program. In 8088, the code segment register and instruction pointer are used to keep track the next instruction to be fetched for execution. Therefore, jump instruction will alter these two registers.

You have seen JMP instructions last week. JMP means jump and it has no condition for performing the jump. This is known as unconditional jump. As opposite to this unconditional jump, there is conditional jump, which is characterized by instructions such as JNZ, JNBE and many more. Compare these two instructions:

- **JMP 0FFF** means jump to address 0FFF (h)
- **JNZ 0FFF** JUMP NOT ZERO means if the previous instruction result was not zero, then jump to address 0FFF (h). Else if the result is zero (condition is not met) then proceed with next line.

[Note: There are 31 conditional jump instructions provided in 8088 microprocessor]

Also in previous lab, you have seen the LOOP instruction. The LOOP checks CX. If CX is 0, the processor proceeds to the next instruction. Otherwise, it decreases CX by 1, and then jumps to the address specified in the LOOP instruction. The same process can be also simulated using the JNZ instruction.

For example:

```
Mov cx, 10
** ------
Dec cx
Jnz **
```

This is equivalent to:

```
Mov cx, 10
** ------
Loop **
```

New Assembly Language Instructions

- **CMP AX, DX** – Subtracts DX from AX (AX-DX), but does not store the result. Instead it sets some flags.
- **JNBE 200** – "JUMP NOT BELOW NOR EQUAL"
  If the previous instruction (for example, CMP AX, DX) resulted in a positive value, then jump to the instructions at address 200.

**IF –ELSE**

Since CMP is a subtraction operation, we can follow CMP with a conditional jump instruction to implement the IF-ELSE statement. We also can implement WHILE – DO loop programs as we will see later.
Step 1: LOOP and JNZ

In ‘Background’ above find the programs, which shows you about JNZ and replace the ------ with ‘inc bx’. Rewrite those programs and explain the flow of those programs. (You are not required to run the program). If the ‘inc ax’ is to repeated with ‘inc cx’, what do you think will happen?

Step 2: CMP and JNBE

```assembly
Mov ax, value1
Mov bx, value2
Mov cx,1
Cmp ax, bx
Jnbe NEXT
Dec cx
NEXT inc cx
```

[Note: You can count the address NEXT. The jump instruction takes 2 memories, and DEC takes 1]

Test the above program for
1. ax > bx
2. ax < bx.

In the above program, AX is compared to BX. If JNBE evaluates to true, only INC CX will be executed, and the value of CX will be 2. If JNBE is false, both DEC CX and INC CX will be executed, and the value of CX will be 1.

3. Now with the values of ax < bx, reverse the CMP instruction. What you observe?

Explain the results for Step 2. Which part of the program resembles IF – ELSE statement?
Step 3: Arithmetic sum
By using instruction LOOP; write a program that computes the sum of the numbers from 1 to 10h. Draw a flow chart before you program.

Step 4
Write a program that counts the number of times a certain number appears in a range of memory. (Hint: to compare two numbers, subtract the two numbers. If the numbers are equal, the subtraction result will be zero. Then you can use JNZ.)
Questions:

1. What are flags? How many flags are available in 8088?
2. In CMP instructions, which flags are effected?
3. How can we implement While-Do loop by using CMP and jump instructions?
4. Describe the difference between a jump and call instructions.
5. The following program is known as delay loop. How many times does the JNZ DLY instruction get executed?

```
Mov cx,100H
DLY:     Dec  cx
        JNZ  DLY
Next:    . . . . .
```