National University of Computer and Emerging Sciences



Laboratory Manual

for

Computer Organization and Assembly Language Programming

(EL 213)

|  |  |
| --- | --- |
| Course Instructor | Ms. Tazeem Haider |
| Lab Instructor(s) | Mr. Gullsher Ali Chaudhary  Ms. Nimra Abbas |
| Section | CS-3L |
| Semester | Fall 2022 |

Department of Computer Science

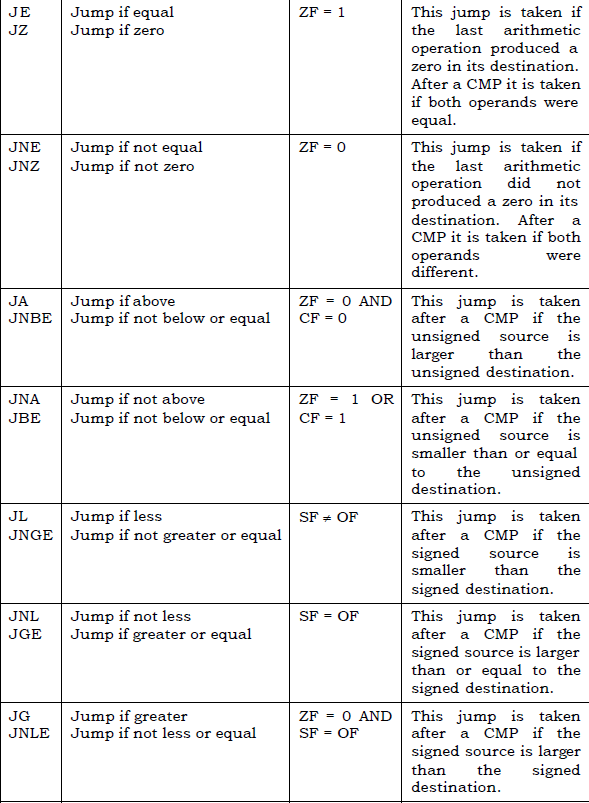
FAST-NU, Lahore, Pakistan

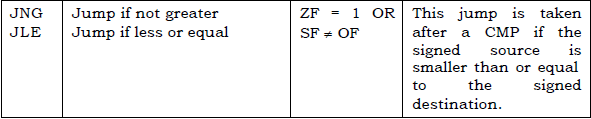
## Objectives

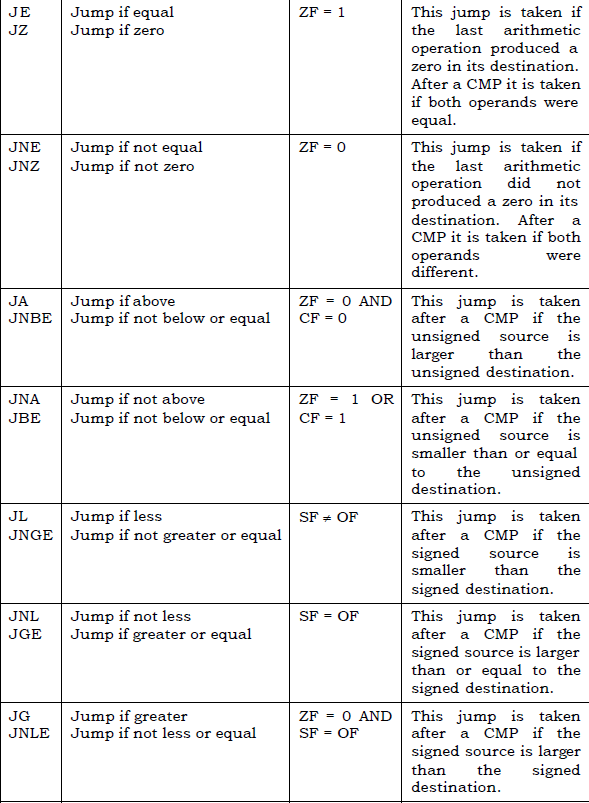
After performing this lab, students shall be able to:

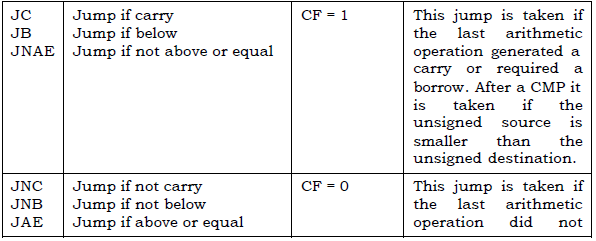
* Understand and learn different conditional jumps.
* Understand and learn unconditional jumps.
* Rationally code jumps based on the requirements.

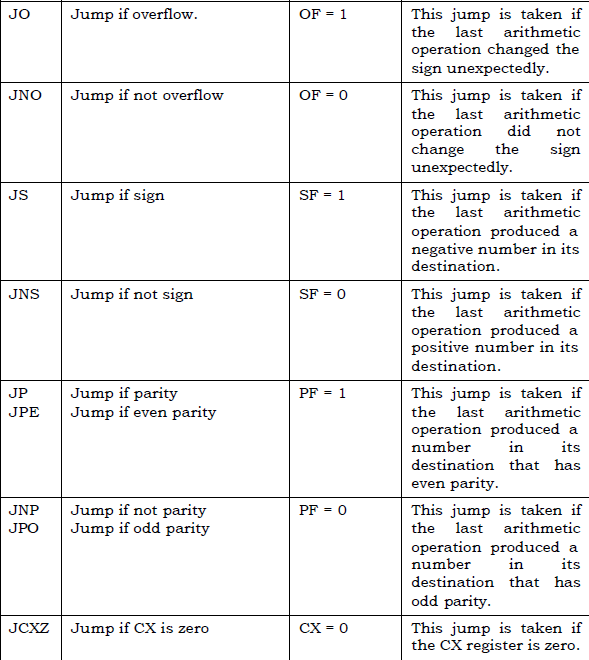
# Conditional Jumps

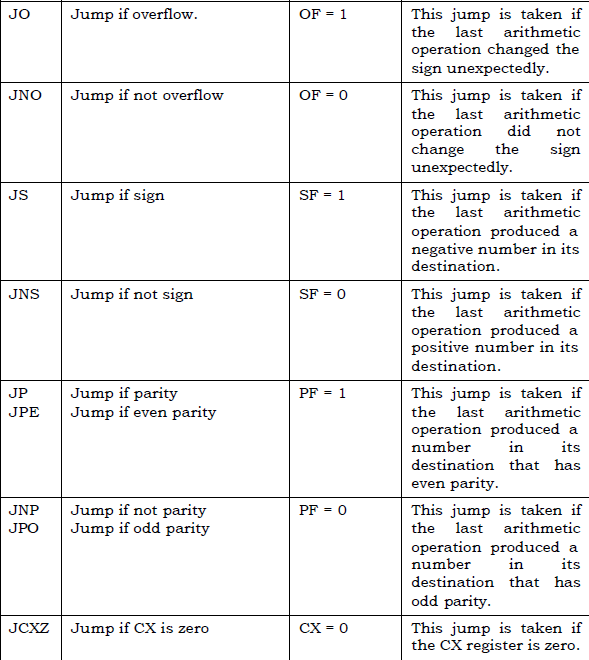












**Exercise 1:** Write an assembly language code that computes square of ‘num’ and saves it in ‘square’. num and square are two memory variables of word length. For any number n, your program should add n, n times to produce its square. Note how 5^2 = 5+5+5+5+5.

**For Example:**

num=6

square=0x24

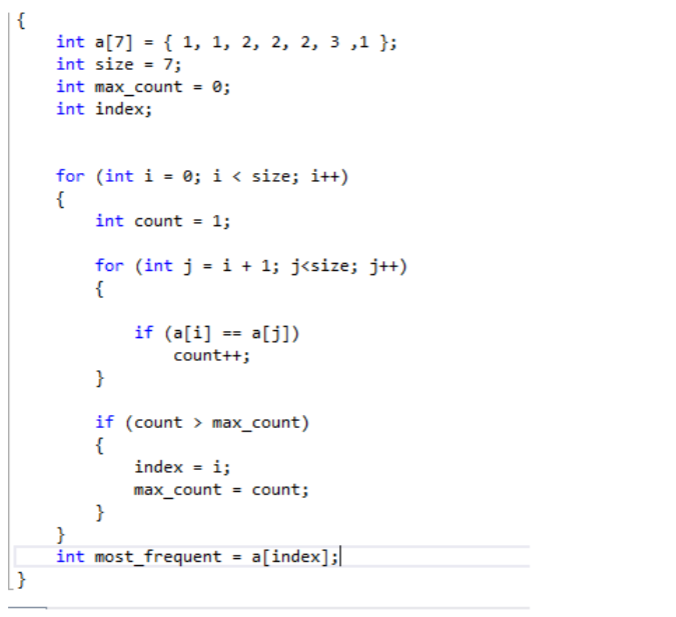
**Exercise 2:** Write an assembly program that which finds if an array 'digits' is duplicate or not. The array ends on a -1. If the number in the array is duplicate, set bx to 1 and 0 otherwise.

**For example:**

|  |  |
| --- | --- |
| If  digits: 1,1,2,1,1, -1  set bx=1 | If  digits: 1,2,3,4,-1  set bx=0 |

**Exercise 3:** Write a program in assembly code that generate the first twelve Fibonacci number sequence (1, 1, 2, 3, 5, 8,…) in an array named as your name using indirect addressing mode and loop. The rule to generate sequence is Fn= Fn-1+Fn-2.

**Exercise 4:** Following is code to find most frequent element of an array. Convert this code to assembly language



**(Self-Exercise) Exercise 5:** You are given an array of positive numbers, the total number of elements in array and a key to be searched for in the array (all values are to be initialized in memory). Write a program that finds an element (Key) in given array. If you find that element then replace that element with sum of array elements up to the key (excluding key element). Otherwise append the key at the end of array (after the last element of the array in the memory).

**Example:**

Array: db 3,5,9,10,11,4,1,3,15,8

Key: db 1

Size: dw 10

Array (after program execution) = 3, 5, 9, 10, 11, 4, 42 (3+5+9+10+11+4), 3, 15, 8