National University of Computer and Emerging Sciences

Lab Manual 04

Computer Organization and Assembly Language

Lab 04

Instructor	Rida Mahmood, Amna Sehar
Class	BCS-3A
Sections	A1, A2
Semester	Fall 2023

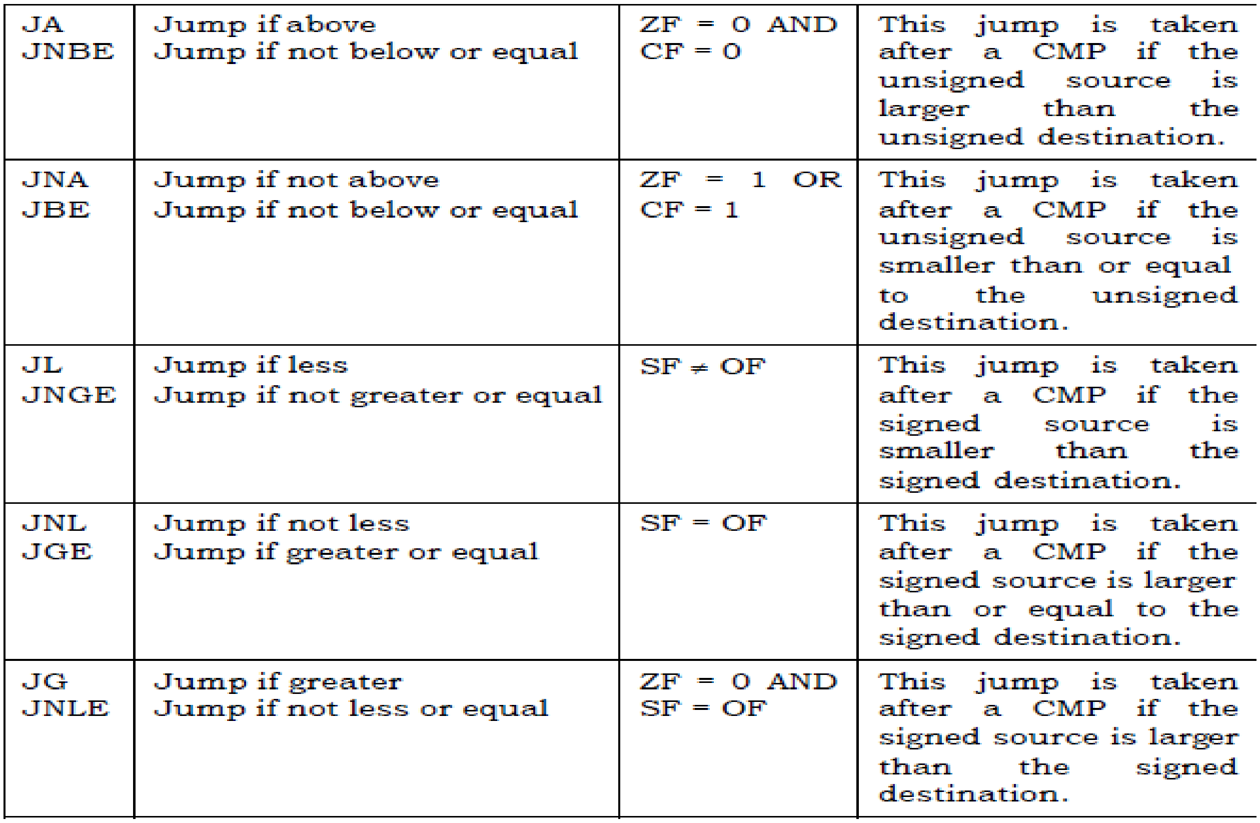
Fast School of Computing

FAST-NU, Lahore, Pakistan

Objectives

- How to interpret the different types of jumps
- How to use the different types of registers and how to manipulate them in assembly language
- How to perform arithmetic operations with registers and conditional jumps
- How to use the debugger for viewing the available registers and their function

Conditional Jumps



JA JNBE	Jump if above Jump if not below or equal	ZF = 0 AND CF = 0	This jump is taken after a CMP if the unsigned source is larger than the unsigned destination.
JNA JBE	Jump if not above Jump if not below or equal	ZF = 1 OR CF = 1	This jump is taken after a CMP if the unsigned source is smaller than or equal to the unsigned destination.
JL JNGE	Jump if less Jump if not greater or equal	SF ≠ OF	This jump is taken after a CMP if the signed source is smaller than the signed destination.
JNL JGE	Jump if not less Jump if greater or equal	SF = OF	This jump is taken after a CMP if the signed source is larger than or equal to the signed destination.
JG JNLE	Jump if greater Jump if not less or equal	ZF = 0 AND SF = OF	This jump is taken after a CMP if the signed source is larger than the signed destination.

JNG JLE	Jump if not greater Jump if less or equal	ZF = 1 OR SF ≠ OF	This jump is taken after a CMP if the signed source is smaller than or equal to the signed destination.
JE JZ	Jump if equal Jump if zero	ZF = 1	This jump is taken if the last arithmetic operation produced a zero in its destination. After a CMP it is taken if both operands were equal.
JNE JNZ	Jump if not equal Jump if not zero	ZF = 0	This jump is taken if the last arithmetic operation did not produced a zero in its destination. After a CMP it is taken if both operands were different.
JC JB JNAE	Jump if carry Jump if below Jump if not above or equal	CF = 1	This jump is taken if the last arithmetic operation generated a carry or required a borrow. After a CMP it is taken if the unsigned source is smaller than the unsigned destination.
JNC JNB JAE	Jump if not carry Jump if not below Jump if above or equal	CF = 0	This jump is taken if the last arithmetic operation did not

JO	Jump if overflow.	OF = 1	This jump is taken if the last arithmetic operation changed the sign unexpectedly.
JNO	Jump if not overflow	OF = 0	This jump is taken if the last arithmetic operation did not change the sign unexpectedly.
JS	Jump if sign	SF = 1	This jump is taken if the last arithmetic operation produced a negative number in its destination.
JNS	Jump if not sign	SF = 0	This jump is taken if the last arithmetic operation produced a positive number in its destination.
JP JPE	Jump if parity Jump if even parity	PF = 1	This jump is taken if the last arithmetic operation produced a number in its destination that has even parity.
JNP JPO	Jump if not parity Jump if odd parity	PF = 0	This jump is taken if the last arithmetic operation produced a number in its destination that has odd parity.
JCXZ	Jump if CX is zero	CX = 0	This jump is taken if the CX register is zero.

Lab Exercises – Nested Loops + Conditional Jumps

Activity 1[Nested Loop]: Write a program that computes factorial of a number without using multiplication instruction.

Activity 2: Write a program that compresses an ordered array (having multiple occurrences of one integer). You are not allowed to use any extra array, just modify the input array.

Sample Run:

Arr: 2,2,2,3,4,4,5,5,5,6
After Compression, **Arr:** 2,3,4,5,6,0,0,0,0,0

Activity 3: We did following Sorting Example in class.

i. Run this code (without any modification) on signed data and verify output.

ii. Modify this code to sort **signed data** and test it.

```
; sorting a list of ten numbers using bubble sort
; a program to add ten numbers without a separate counter
[org 0x0100]
    jmp start ; unconditionally jump over data
num1: dw 5,4,3,2,1

start:
mov si, 0

outerloop:
mov di, si
add di, 2

innerloop:
mov ax, [num1+si]
cmp ax, [num1+di]
jb noswap
mov dx, [num1+di]
mov [num1+si], dx
mov [num1+di], ax

noswap:
add di, 2
cmp di, 10
jb innerloop

add si,2
cmp si, 8
jb outerloop
```

```
mov ax, 0x4c00 ; terminate program
int 0x21
```

Homework (Marked Evaluations)

1- Find min and max elements from an array of 10 integers. Write two functions for

i. Unsigned Numbers

ii. Signed Numbers

2- Write a program that finds union of two sets (ordered integer arrays). Assume there will be at max 10 elements in union.

Sample Run: (Zero indicates end of array)

```
Set1: 1,4,6,0
Set2: 1,3,5,8,0
Union: 1, 3, 4, 5, 6, 8, 0
```

3- Write a program that finds intersection of two sets (ordered integer arrays). Assume there will be at max 10 elements in union.

Sample Run: (Zero indicates end of array)

```
Set1: 1,4,6,0
Set2: 1,3,6,8,0
Intersection: 1, 6, 0
```

4- Write a program that finds staring index of a subset from a set (smaller array from a larger array).

Sample Run:

```
Set:1,2,4,1,1,2,3,1,2,5
Subset: 1,2,3
Index: 4 (After program. -1 if subset not found.)
```

Practice [Conditional Jump]: Find frequency of an integer from an array of 10 integers.
For example frequency of 2 in array 2, 4, 2, 8, 2 is 3.