



# University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

## Computer Organization and Assembly Language

### Lab 05

Lab 05	
Topic	<ul style="list-style-type: none"><li>• JMP instructions</li><li>• Conditional/Unconditional JMP instructions (if-else)</li><li>• Memory addressing modes, Data movement, Array, addition and subtraction instructions.</li></ul>



## PART 1

### JUMP INSTRUCTIONS

#### Two main types of jump instructions

##### a) Unconditional jump:

To start executing instructions unconditionally

JMP SKIP ; UNCONDITIONAL JMP INSTRUCTION

MOV AX, 2;

ADD AX, 1;

SKIP: ; SKIP IS A LABEL

MOV AX,7

MOV AX,0X4C00

INT 21H

VAR1: DB 5

VAR2: DW 77



# University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

## b) Conditional jump:

To start executing instructions based on some condition.

Some conditional jumps are as follows:

JC	Jump if carry flag set
JNC	Jump if not carry
JZ	Jump if zero flag set
JE	Jump if zero flag set
JNZ	Jump if zero flag not set
JNE	Jump if zero flag not set
JS	Jump if sign flag is set
JNS	Jump if sign flag not set
JO	Jump if overflow
JNO	Jump if not overflow

## 1. Conditional jumps after *signed operand comparison*

JG	Jump if greater
JNG	Jump if not greater
JGE	Jump if greater or equal
JNGE	Jump if not greater or equal
JL	Jump if less
JNL	jump if not less
JLE	Jump if less or equal
JNLE	jump if not less or equal



# University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

## 2. Conditional jumps after *unsigned operand* comparison

JA	Jump if above
JNA	Jump if not above
JAЕ	Jump if above or equal
JNAЕ	Jump if not above or equal
JB	Jump if below
JNB	Jump if not below
JBE	Jump if below or equal
JNBE	jump if not below or equal

- **Compare Instruction**

`cmp operand1,operand2`

It subtracts operand2 from operand1 and updates the flags only *without updating the value of the operands.*

### Example: Signed number comparison

In this example it compares two numbers and stores **1** in ax if *al* is greater than *bl* else **0**.

```
[org 0x100]
```

```
MOV AL, 5
```

```
MOV BL, 0xFF
```

```
CMP AL, BL
```

```
JG L1 ; signed statement; jump if greater ...
```

```
MOV AX, 0
```

```
JMP Exit
```

```
L1:
```

```
MOV AX, 1
```



# University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Exit:

```
mov ax,0x4c00
```

```
int 21h
```

## Example: Unsigned number comparison

```
[org 0x100]
```

```
MOV AL, 5
```

```
MOV BL, 0xFF
```

```
CMP AL, BL
```

```
JA L1 ; unsigned statement; jump if above ...
```

```
MOV AX, 0
```

```
JMP Exit
```

```
L1:
```

```
MOV AX, 1
```

Exit:

```
mov ax,0x4c00
```

```
int 21h
```

**Note:**

			Considering 8-bits
Unsigned	5	<	255(0xFF)
Signed	5	>	-1(0xFF)



# University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

- **ArrayName: db 1,2,3,4,5,6,7,8,9,10;** *This will reserve 10 byte sized values in consecutive memory locations.*
- **ArrayName: dw 1,2,3,4,5,6,7,8,9,10;** *This will reserve 10 word sized values 20 bytes in consecutive memory locations.*
- **var: times 10 db 7;** *Initializing array var with total 10 elements with value 7.*

## Sample Code 1: (Declaration and Access from arrays....)

```
[org 0x100]

MOV AL, [ARRAY1]; THIS WILL LOAD FIRST ELEMENT FROM ARRAY1

MOV BX, [ARRAY2]; THIS WILL LOAD FIRST ELEMENT FROM ARRAY2

mov ax,0x4c00
int 21h

ARRAY1 DB 1,2,3,4,5;

ARRAY2 DW 0xA, 0xB, 0xC, 0xD, 0xE;
```

## Sample Code 2: (Access data from an array....)

**TO ACCESS NEXT ELEMENTS WITHIN AN ARRAY ADD OFFSETS TO ARRAY NAME DEPENDING UPON ARRAY SIZE TYPE.**

```
[org 0x100]

MOV AL, [ARRAY1+1]; THIS WILL LOAD SECOND ELEMENT FROM ARRAY1

MOV BX, [ARRAY2+2]; THIS WILL LOAD SECOND ELEMENT FROM ARRAY2

mov ax,0x4c00
int 21h

ARRAY1 DB 1,2,3,4,5;

ARRAY2 DW 0xA, 0xB, 0xC, 0xD, 0xE;
```



# University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

**FACULTY OF INFORMATION TECHNOLOGY**

## **Sample Code 3: Access data from an array (Indexed register Indirect mode)**

```
[org 0x100]
mov SI, VEC1 ;Loads the effective address of array VEC1 in SI
mov DI, VEC2 ;Loads the effective address of array VEC2 in DI

L1:
MOV AL, [SI]
MOV [DI], AL

INC SI; ;adds '1' to the destination operand.
INC DI;
DEC byte [COUNT]; Subtracts '1' from the destination operand.
JNZ L1; Repeats executing instructions from label L1 until last arithmetic operation
;produces 0.

mov ax,0x4c00
int 21h

VEC1: DB 1, 2, 5, 6,8
VEC2: DB 0,0,0,0,0
COUNT: DB 5
```