## 8086 Microprocessor Laboratory Experiments

Experiment 7: Unconditional and Conditional Jumps

Murtadha Hssayeni, Ph.D.

m.hssayeni@uobabylon.edu.iq





#### Outlines

#### o Labels

#### • JUMP instruction

- Unconditional
- Conditional
- Conditional JUMP for signed numbers
- Example
- Procedure

#### Labels

□ To declare a **label**, just type its name and add ":" to the end,

□ label name can be any character combination but it cannot start with a number.

For example: label1: label2: A:

Label can be declared on a separate line or before any other instruction, for example:
 X1:
 MOV AX, 1
 x2: MOV AX, 2

#### JUMP instruction

- **JUMP** is one of the **program execution transfer instructions**
- Controlling the program flow is a very important thing, this is where your program can make decisions according to certain conditions.
- □ There are two main types of this instruction
- 1. Unconditional JUMP instruction: this instruction is executed without any condition. JMP Label
- 1. Conditional JUMP instruction: this instruction is not executed until the condition is true. *Jcondition Statement Label*

## Unconditional JUMP Cont.

□ JMP Label ;Unconditionally transfers control to the target location.

**Example of JMP instruction**: creating a subroutine to add up two numbers:

```
ORG 100H

MOV AX, 5 ; set ax to 5.

MOV BX, 2 ; set bx to 2.

JMP calc ; go to 'calc'.

Back: JMP stop ; go to 'stop'.

Calc:

ADD AX, BX ; add bx to ax.

JMP Back ; go 'back'

Stop:

Ret ; return to operating system.
```

As you can see from this example JMP is able to transfer control both forward and backward.

### Conditional JUMP

□ *Jcondition Label* ;Conditionally transfers control to the target location.

- Condition is one, two, or three letters
- □ These instructions are divided in three groups:
  - First group just test single flag.
  - Second compares numbers as signed.
  - □ Third compares numbers as unsigned.
- □ When a conditional jump is executed, the CPU checks the flags register.
  - □ If the **conditions for the jump are true**, the <u>CPU adjusts the IP register</u> to point to the destination label, so that the instruction at this label will be executed next.
  - □ If **the jump condition is false**, then the <u>IP register is not altered</u>; this means that the next sequential instruction will be executed.

# Conditional JUMP: Jump instructions that test single flag

Instruction	Description	Condition	Opposite Instruction
JZ , JE	Jump if Zero (Equal).	ZF = 1	JNZ, JNE
JC , JB, JNAE	Jump if Carry (Below, Not Above Equal).	CF = 1	JNC, JNB, JAE
JS	Jump if Sign.	SF = 1	JNS
JO	Jump if Overflow.	OF = 1	JNO
JPE, JP	Jump if Parity Even.	PF = 1	ЈРО

- There are some instructions that do that same thing, they even are assembled into the same machine code:
  - when you compile JE instruction, you will get it assembled as: JZ
  - □ JC is assembled the same as JB etc...

#### Conditional JUMP for signed numbers

Instruction	Description	Condition	Opposite Instruction
JE , JZ	Jump if Equal (=). Jump if Zero.	ZF = 1	JNE, JNZ
JNE , JNZ	Jump if Not Equal (⇔). Jump if Not Zero.	ZF = 0	JE, JZ
JG , JNLE	Jump if Greater (>). Jump if Not Less or Equal (not <=).	((SF xor OF) or ZF)=0	JLE, JNG
JLE, JNG	Jump if Less or Equal (<=). Jump if Not Greater or Equal (not >). Jump if Carry.	(SF xor OF) or ZF)=1	JG , JNLE
JGE. JNL	Jump if Greater or Equal (>=). Jump if Not Less (not <).	(SF xor OF)=0	JL, JNGE
JL, JNGE	Jump if Less (<). Jump if Not Greater or Equal (not >=).	(SF xor OF)=0	JGE. JNL

#### Conditional JUMP for signed numbers

Write a simple program in Assembly language to implement the following: If AX>5:

AX=1

Else:

AX=0

The default value of AX is 2.

Answer:ORG 100HMOV AX, OFFFAH; is -6d if it was signedCMP AX, 5dJG AJG A; jump if AX greater than 5dMOV AX, 0JMP STOPA:MOV AX, 1 ;not executed in this exampleSTOP:RET

#### Procedure

- 1. Write a program in 8086 emulator to perform the following tasks:
  - A = X Y \* 3D where X = 20D, Y = 2D
  - □ Reset DL=0 if the result A is less than 10D, else set DL =1.
- 2. Write a program in 8086 emulator to transfer a block of 10 bytes of data using conditional jump instructions.
  - Set DS = A300H, SI=300H is the offset to the source data, and DI=400H is the offset to the destination location.