Republic of Iraq
Ministry of Higher Education and Scientific Research
Al-Furat Al-Awsat Technical University
Engineering Technical College/Najaf
Al Najaf Al Ashraf, 31001. Iraq.

8085 Microprocessor
Lecture 10

المدرس ضرغام الخفاف الاسدي

Third Year lecture notes

Avionics Engineering Dept.

Engineering Technical College/ NAJAF 2020-2021

Lecturer: Dhurgham Al-Khaffaf Alasady

Branch Group

This group of instructions changes the normal sequence of the program. The branch instructions include JUMP, CALL and RETURN instructions, which are further sub-divided as unconditional and conditional JUMP, CALL and RETURN instructions.

Unconditional Jump Instructions

***1- JMP address* (label) (Jump to Address)**

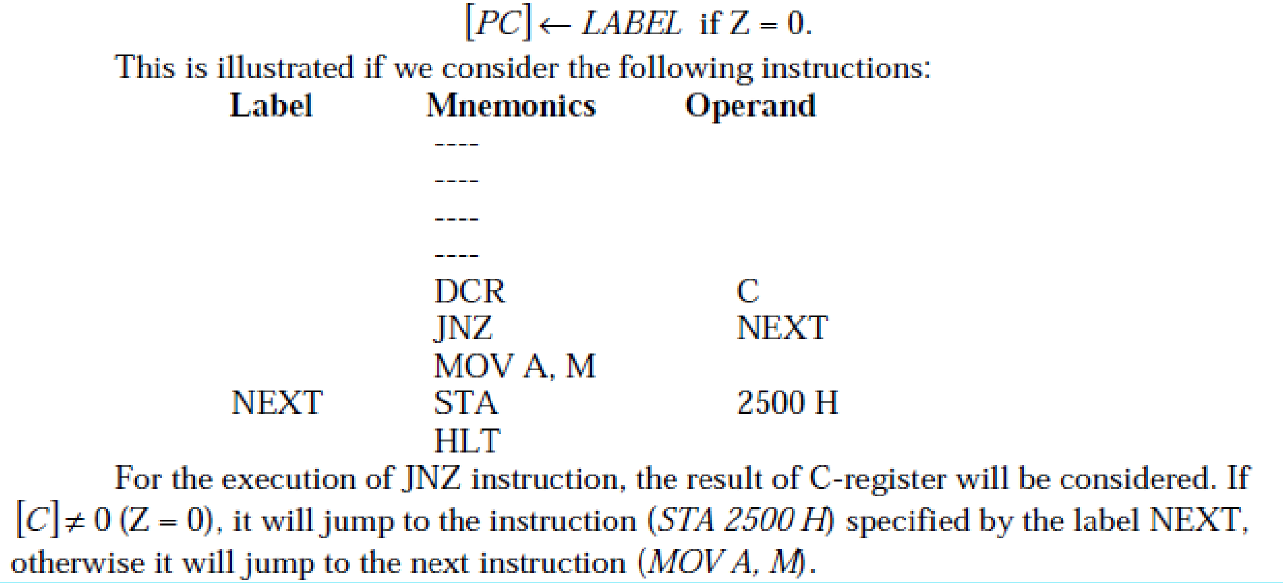
This is an unconditional jump instruction. With the execution of this instruction, the program jump to the address (or label) specified with the instruction. This is a three byte instruction and no flag is affected. In fact during the execution of *JMP address* (label) instruction, the address of the label is copied in the program counter; and whenever the program fetches the next instruction the program counter will send the address of this given label.

$$[PC] \leftarrow [LABEL]$$

Conditional Jump Instructions

In conditional jump instructions the program jumps to the instructions specified by the address (or label) if the given condition is fulfilled. However, if the given condition is not satisfied, the program will not jump to the specified address (or label) rather it will proceed to the normal sequence. In all these conditional jump instructions, if program jumps to the specified address after the given condition is satisfied.

1- JNZ address **(label) (Jump if the result is not zero)** When this instruction is executed, the program jumps to the instruction specified by the address (or label), if the result is not zero; otherwise it will proceed to the next instruction. Here the result of the preceding instruction is considered. In this case the address (or label) is copied in the program counter if zero flag is reset (Z = 0).



$$[PC] \leftarrow LABEL \text{ if } Z = 0.$$

This is illustrated if we consider the following instructions:

```
Label       Mnemonics      Operand
            ----
            ----
            ----
            ----
            DCR            C
            JNZ            NEXT
            MOV A, M
NEXT        STA            2500 H
            HLT
```

For the execution of JNZ instruction, the result of C-register will be considered. If $[C] \neq 0$ (Z = 0), it will jump to the instruction (*STA 2500 H*) specified by the label NEXT, otherwise it will jump to the next instruction (*MOV A, M*).

***2- JZ address* (label) (Jump if the result is zero)** The condition of this instruction is reverse to that of *JNZ address*. In this case the program will jump to the instruction specified by the address (or label), if the result of the preceding instruction is zero (Z = 1) otherwise it will proceed to the next instruction in the normal sequence. The address of the label will be copied in the program counter if Z flag is set (or result is zero).

$$[PC] \leftarrow LABEL \ \text{if } Z = 1.$$

***3- JNC address* (label) (Jump if no carry)** During the execution of this instruction, it will check up the Carry flag modified by the preceding instruction. If there is no carry (CY = 0 or CY flag is reset), the program will jump to the instruction specified by the address (or label) otherwise it will proceed to the next instruction of the normal sequence. In this case the address of the label will be copied in the program counter if CY = 0 or CY is reset.

$$[PC] \leftarrow LABEL\text{, if CY} = 0$$

***4- JC address* (label) (Jump if carry)** The program will jump to the instruction specified by the address (label) if the CY flag is set (CY = 1) which is modified by the preceding instruction. However, if the carry is reset (CY = 0), it will proceed to the next instruction of the normal sequence. The condition of this instruction is opposite to that of the *JNC address*. In this case the address of the label will be copied in the program counter if CY = 1 or CY is set.

$$[PC] \leftarrow LABEL \text{ , if CY} = 1$$

5- *JM address* (label) (Jump if Minus) When this instruction is executed, the program will jump to the instruction specified by the address (label) if the result of the preceding instruction is minus or sign flag is set (S = 1) otherwise it will proceed to the next instruction of the normal sequence.

$$[PC] \leftarrow LABEL \text{ , if S} = 1$$

***6- JP address* (label) (Jump if Positive)** If the result of the preceding instruction is positive or sign flag is reset (S = 0), the program will jump to the instruction specified by the address (label). However, if the condition is not satisfied it will proceed to the next instruction of the normal sequence.

$$[PC] \leftarrow LABEL \text{ , if S = 0}$$

***7- JPO address* (label) (Jump if Parity is Odd)** If the parity is odd or parity flag is reset (P = 0) as a result of the preceding instruction, the program will jump to the instruction specified by the address (label) otherwise next instruction of the normal sequence will be executed.

$$[PC] \leftarrow LABEL \text{ , if P = 0}$$

***8- JPE address* (label) (Jump if Parity is Even)** If the parity is even or parity flag is set (P = 1) as a result of the preceding instruction, the program will jump to the instruction specified by the address (label) otherwise next instruction of the normal sequence will be executed.

$$[PC] \leftarrow LABEL \text{ , if P = 1}$$

Thank you