18/11/2021

MICROPROCESSOR

CONDITIONAL JUMP INSTRUCTION

This instruction cause a jump to an address given in the instruction if the desired condition occurs in the program before execution of the conditional jump instruction.

Type of conditional jump :-

① Jump on carry.

Syntax : JC 16-bit memory address

This instruction will jump to address given if CY = 1

For example :

```
        MVI A, 01H.
        MVI B, FFH
Next :  ADD B
        JC Next
        HLT.
```

A = 01H
B = FFH
A ← A+B
A = 00H ; CY = 1

A ← A+B
A = FF ; CY = 0

```
  0000 0001
+ 1111 1111
 [1] 0000 0000
  ↓     A
  CY
```

Next → Label used in places of address.

② Jump on no Carry.

Syntax : JNC 16-bit memory address.

Example : JNC 3000H.

This instruction will jump to 3000H if CY=0 when this instruction is executed else it will execute the next instruction.

③ Jump on Zero

Syntax : JZ 16-bit memory address.

Example : JZ 2020H.

This instruction will jump to 2020H if $\boxed{Z=1}$ else it will execute next instruction.

```
AGAIN : MVI A, 42H.
        MVI B, 20H.
        SUB B
        JZ AGAIN
        ADI 02H
        HLT.
```

A = 42H
B = 20H
A ← A-B } Z=0
A ← 22H

A ← A+02H.
A = 24H.

(4) Jump on no zero

Syntax: JNZ 16-bit address.

Example: JNZ 4000H.

This instruction will jump to 4000H when Z=0 else it will execute the next instruction.

(5) Jump on positive.

Syntax: JP 16-bit address.

Example: JP 2000H.

This instruction will jump to 2000H if Sign flag (S=0) else it will execute the next instruction.

(6) Jump on minus.

Syntax: JM 16-bit address.

Example: JM 2000H.

This instruction will jump to 2000H if Sign flag (S=1) else it will execute next instruction.

⑦ Jump on even parity.

Syntax : JPE 16-bit address.

Example : JPE 3000H.

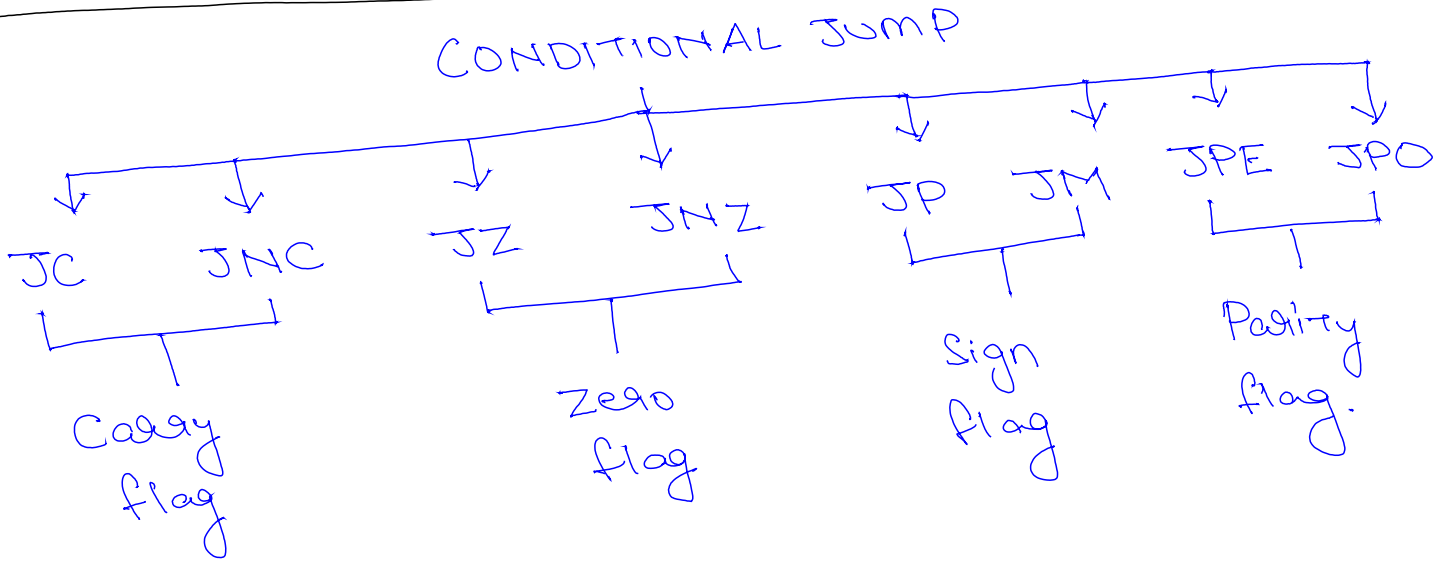
This instruction will jump to 3000H. if parity flag (P=1) else it will execute next instruction.

⑧ Jump on odd parity.

Syntax : JPO 16-bit address.

Example : JPO 3000H.

This instruction will jump to 3000H if parity flag (P=0) else next instruction is executed.



→ Multiply two numbers stored in memory 2000H and 2001H. Store the result at 2002H. Assume 2000H | 02H |

2001H | 03H.

Soln. → Multiplication is repeated addition.

(02+02+02)

```
MVI A, 00H.        ──→ A = 00H.
LXI H, 2000H.      ──→ B ← 2000H. counter.
MOV B, M.
INX H
AGAIN: ADD M       ←   A + M = A
                       00 + 03 = 03
       DCR B       ←   B = B-1 ; B = 2-1 = 1
                       Z = 0
       JNZ AGAIN   ←   check zero flag.
       STA 2002H.      A+M = A
                       03+03 = 06
       HLT.            B = B-1 ; B = 1-1 = 0
                       Z = 1
```

2002 | 06 |

18/11/2021

MICROPROCESSOR

Multiply two 8-bit numbers

Q Multiply two 8-bit numbers stored in memory locations 2200H and 2201H. Store the result in memory 2300H and 2301H.

Sample Problem:

* Multiplication of two 8-bit numbers will give 16-bit product.

2200H | 03H

2201H | B2H

Result = B2 + B2 + B2

= 216H.

2300H | 16H (lower byte)

2301H | 02H. (higher byte.)

```
    LDA 2200H
    MOV E, A      ← E = 03
    MVI D, 00H    ← D = 00
    LDA 2201H.
    MOV C, A      ← C = B2H ; C will work as Counter
    LXI H, 0000H  ← HL = 00H
AG: DAD D         ← HL ← HL + DE ; 0000H + 0003
    DCR C         ← C = B2 - 1 = B1 ; Z = 0
    JNZ AG
    SHLD 2300H
    HLT.
```

HL ← HL + DE ; 0003 + 0003

C = B1 − 1 = B0 ; Z = 0

03 + 03 + 03 + 03 + ... + 03

(B2 times.)

COMPARE INSTRUCTION

Syntax : ① CMP R ② CMP M ③ CPI 8-bit data

This instruction compares the value of Accumulator with the value of R, M or 8-bit data.

If the values are equal : Z = 1

If the value of A < Second number : CY = 1

Example : ① MVI A, 20H
MVI B, 20H.
CMP B ← Compare values of A and B

Since A = B ; after execution Z = 1
CY = 0

→ How Comparison is done by 8085 µp?.

8085 performs Subtraction. internally

② MVI A, 02H.
CPI 03H

Z = 0 CY = 1

③ LXI H, 2000H
MVI A, 10H.
CMP M.

2000H | 06H

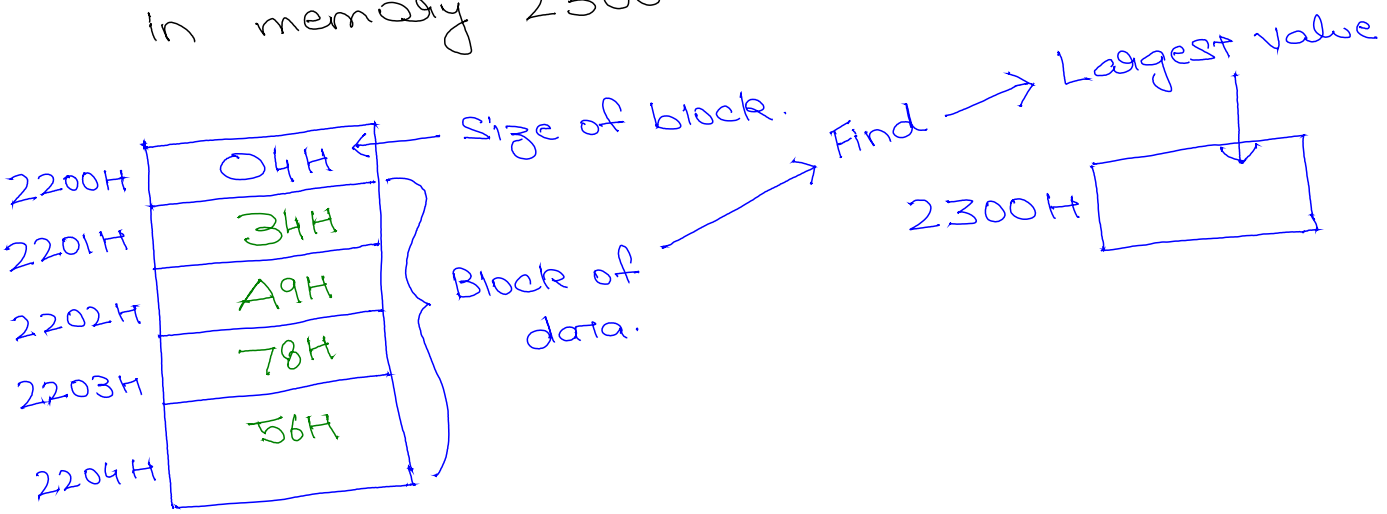
Since (A) > (M)

Z = 0 CY = 0

Find the largest of given numbers

00 → 0
01 → 1
10 → 1
11 → 0

Q Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself start from memory location 2201H. Store the maximum number in memory 2300H.




```
LDA   2200H
MOV   C,A      ← Initialized a counter (Size of block)
XRA   A        ← clear the accumulator A=0
LXI   H,2201H  ← point to first data of block.
BACK: CMP   M
      JNC   SKIP
      MOV   A,M
SKIP: INX   H
      DCR   C
      JNZ   BACK
      STA   2300H
      HLT.
```

1	2	3	4
A=0 ; M=34 CY=1 JNC X A=34 HL=2202H. C=04-1=3 Z=0 JNZ ✓	A=34 ; M=A9 CY=1 JNC X	A=A9 M=78 CY=0 JNC ✓ HL=2204 C=2-1=1 Z=0 JNZ ✓	A=A9 ; M=56 CY=0 JNC ✓ HL=2205H. C=1-1=0 Z=1 JNZ X 2003 A9H. Stop.
A=A9 HL=2203 C=3-1=2 Z=0 JNZ ✓			

Q1. WAP to SORT given 10 numbers from memory 2200H in ascending order.

Q2. WAP to SORT given 10 numbers from memory 2200H in descending order.