

Branch Group Instructions:

The branch instructions allow the microprocessor to change the sequence of program execution.

The branch group instructions is classified into three categories:

- 1- *Jump instructions.*
- 2- *Call instructions.*
- 3- *Return instructions.*

The branching instruction alter the normal sequential flow. These instructions alter either (**unconditionally or conditionally**)

1. Jumping instructions

- **Unconditional:** there is one instruction

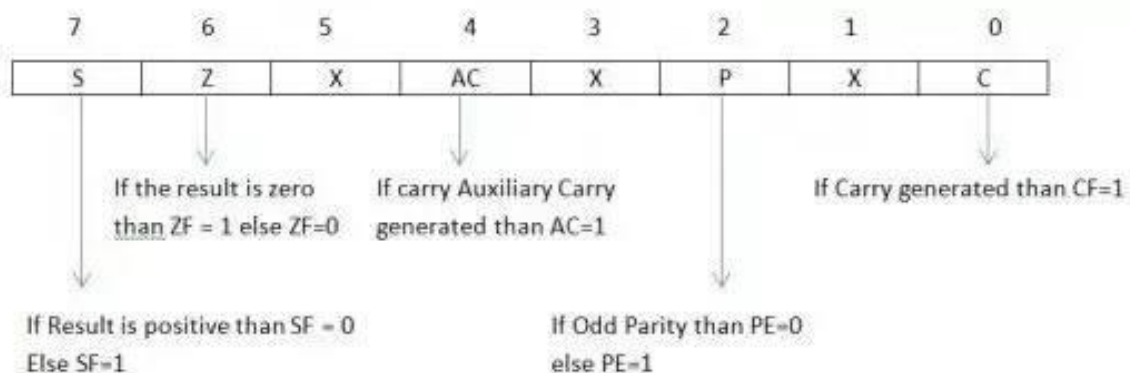
JMP operand [three bytes instruction]

The program sequence is transferred to the memory address given in the operand.

Exp:

JMP 2300 changing the execution of program starting from address (2300).

- **Conditional:** The jump instruction transfers the program sequence to the memory address given in the operand based on the specified flag.



1. **JC operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present carry flag value is 1. If carry flag value is 0, program flow continues sequentially

Exp:

JC 2300 changing the execution of program starting from address (2300) if (CY=1).

2. **JNC operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present carry flag value is 0. If carry flag value is 1, program flow continues sequentially. Operand is 16-bit number.

Exp:

JNC 2300 changing the execution of program starting from address (2300) if (CY=0).

3. **JPO operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present parity flag value is 0(odd). If parity flag value is 1, program flow continues sequentially. Operand is 16-bit number.

Exp:

JPO 2300 changing the execution of program starting from address (2300) if (P=0).

4. **JPE operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present parity flag value is 1(even). If parity flag value is 0, program flow continues sequentially. Operand is 16-bit number. Operand is 16-bit number.

Exp:

JPE 2300 changing the execution of program starting from address (2300) if (P=1).

5. **JZ operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present zero flag value is 1. If zero flag value is 0, program flow continues sequentially. Operand is 16-bit number.

Exp:

JZ 2300 changing the execution of program starting from address (2300) if (Z=1).

6. **JNZ operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present zero flag value is 0. If zero flag value is 1, program flow continues sequentially. Operand is 16-bit number.

Exp:

JNZ 2300 changing the execution of program starting from address (2300) if (Z=0).

7. **JP operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present sign flag value is 0 (positive). If sign flag value is 1, program flow continues sequentially. Operand is 16-bit number. Operand is 16-bit number.

Exp:

JP 2300 changing the execution of program starting from address (2300) if (S=0).

8. **JM operand** [three bytes instruction]

This instruction is used to jump to the address a16 as provided in the instruction. But as it is a conditional jump so it will happen if and only if the present sign flag value is 1 (minus). If sign flag value is 0, program flow continues sequentially. Operand is 16-bit number.

Exp:

JM 2300 changing the execution of program starting from address (2300) if (S=1).

Conditional Jumps

Instruction Code	Description	Condition For Jump
JC	Jump on carry	CY=1
JNC	Jump on not carry	CY=0
JP	Jump on positive	S=0
JM	Jump on minus	S=1
JPE	Jump on parity even	P=1
JPO	Jump on parity odd	P=0
JZ	Jump on zero	Z=1
JNZ	Jump on not zero	Z=0

Examples:

1. Design a counter to display the numbers from 0 to 9 and display them on the output port 7 segment

START: MVI A 3Fh

OUT 03h

MVI A 06h

OUT 03h

MVI A 5Bh

OUT 03h

MVI A 4Fh

OUT 03h

MVI A 66h

OUT 03h

MVI A 6Dh

OUT 03h

MVI A 7Dh

OUT 03h

MVI A 07h

OUT 03h

MVI A 7Fh

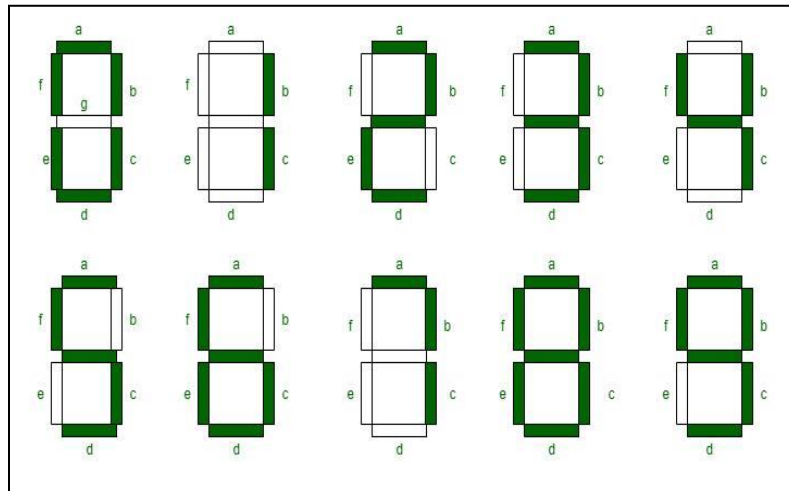
OUT 03h

MVI A 6Fh

OUT 03h

JMP START

Number	g f e d c b a	Hexadecima
0	0 1 1 1 1 1 1	3F
1	0 0 0 0 1 1 0	06
2	1 0 1 1 0 1 1	5B
3	1 0 0 1 1 1 1	4F
4	1 1 0 0 1 1 0	66
5	1 1 0 1 1 0 1	6D
6	1 1 1 1 1 0 1	7D
7	0 0 0 0 1 1 1	07
8	1 1 1 1 1 1 1	7F
9	1 1 0 1 1 1 1	6F



2. Find the operation of multiplying 20 by 5 and store the result in register D?

MVI B 20h

MVI C 05h

MVI A 00

LOOP: ADD B

DCR C

JNZ LOOP

MOV D A

3. Write a program to sum the numbers in memory locations from [1000]h to [100C] and save the result of the operation in memory location [100D]h

MVI A 00h

LXI H 1000h

MVI C 0Ch

START: ADD M

INX H

DCR C

JNZ START

INX H

MOV M A