BHARAT ACHARYA EDUCATION
Cell: 98204 08217
BharatSir@hotmail.com

Branch Group

Conditions

Condition	Description	True if:
NZ	No Zero	Z=0
Z	Zero	**Z=1**
NC	No Carry	C=0
C	Carry	**C=1**
PO	Parity Odd	P=0
PE	Parity Even	**P=1**
P	Plus	S=0
M	Minus	**S=1**

```
Finite Loop with JNZ
 MVI B, 05H
Back: -
 -
 -
 DEC B
 JNZ Back
```

```
Infinite Loop
Back:
 -
 -
 JMP Back
```

1) JMP 16-bit address (Unconditional Jump)

Loads PC with the 16-bit address specified in the instruction.

Eg: JMP 2500 ; PC ← 2500

Addr. Mode	Flags Affected	Cycles	T-States
Immidiate	**None**	3	**10**

2) J Condition **16-bit address (Conditional Jump)**

It is the same as UnConditional JUMP except that the action takes place ONLY if the condition is true.

Eg: JZ 2500 ; PC ← 2500 if Z =1

Addr. Mode	Flags Affected	Cycles	T-States
Immidiate	**None**	2/3	**7/10**

```
Jc, Cf=1
JNC Cf=0
Jz ZF=1
JNZ ZF=0
JPE PE=1
JPO PE=0
Jm SF=1
Jp SF=0
```

3) CALL 16-bit address (Unconditional Call) --- ***Very Imp.***

Loads PC with the 16-bit address specified in the instruction.
Before doing so, it also Pushes the Current PC into the Stack.

Eg: JMP 2500 ; SP ← SP – 1
[SP] ← PC_H
SP ← SP - 1
[SP] ← PC_L
PC ← 2500

#Please refer Bharat Sir's Lecture Notes for this ...

Addr. Mode	Flags Affected	Cycles	T-States
Immidiate	**None**	5	**18**

4) C Condition **16-bit address (Conditional Call)**

It is the same as UnConditional Call except that the action takes place ONLY if the condition is true.

Eg: CNZ 2500 ; IF Z = 0 then
SP ← SP – 1
[SP] ← PC_H
SP ← SP - 1
[SP] ← PC_L
PC ← 2500

#Please refer Bharat Sir's Lecture Notes for this ...

Addr. Mode	Flags Affected	Cycles	T-States
Immidiate	**None**	2/5	**9/18**

BHARAT ACHARYA EDUCATION
Cell: 98204 08217
BharatSir@hotmail.com

5) RET (Unconditional Return)

This instruction is written at the end of the sub-routine and enables the control to go back to the main program.

We enter a subroutine using CALL instruction in which we push the return address into the stack.

In RET instruction we do the opposite i.e. we POP the return address from the stack into PC.

Eg: RET ; $PC_L \leftarrow$ [SP]
SP $\leftarrow$ SP + 1
$PC_H \leftarrow$ [SP]
SP $\leftarrow$ SP + 1

#Please refer Bharat Sir's Lecture Notes for this ...

Addr. Mode	Flags Affected	Cycles	T-States
Indirect	**None**	3	**10**

6) R Condition **(Conditional Return)**

It is the same as UnConditional Return except that the action takes place ONLY if the condition is true.

Eg: RC ; IF C = 1 then
$PC_L \leftarrow$ [SP]
SP $\leftarrow$ SP + 1
$PC_H \leftarrow$ [SP]
SP $\leftarrow$ SP + 1

#Please refer Bharat Sir's Lecture Notes for this ...

Addr. Mode	Flags Affected	Cycles	T-States
Indirect	**None**	1/3	**6/12**

7) RSTn (Restart n)

This instruction is very similar to the CALL instruction.

Here the branch address, instead of being specified directly in the instruction, is calculated as (n × 8).

The current PC is Pushed into the stack.

The new value of PC is (n × 8).

The value of n = 0,1,2 … 7.

These instructions are called as Software Interrupts.

Operationally it is thus simillar to CALL except that it is a **1 byte instruction**.

Eg: RST1 ; SP $\leftarrow$ SP – 1
[SP] $\leftarrow PC_H$
SP $\leftarrow$ SP - 1
[SP] $\leftarrow PC_L$
PC $\leftarrow$ 0008 ($\because$ 1 × 8 = 0008)

#Please refer Bharat Sir's Lecture Notes for this ...

Addr. Mode	Flags Affected	Cycles	T-States
Indirect	**None**	3	**12**

***Please Note**: PCHL also causes a branch in the program flow (to the location pointed by the HL Pair), but is already included in the data transfer group. It is a **very important** instruction and during programming, you should remember that it can also cause a branch.*