**Branching instructions in 8085 microprocessor**

* Difficulty Level : [Easy](https://www.geeksforgeeks.org/easy/)
* Last Updated : 29 Jul, 2020

Branching instructions refer to the act of switching execution to a different instruction sequence as a result of executing a branch instruction.

The three types of branching instructions are: 

Attention reader! Don’t stop learning now. Get hold of all the important CS Theory concepts for SDE interviews with the [**CS Theory Course**](https://practice.geeksforgeeks.org/courses/SDE-theory?vC=1) at a student-friendly price and become industry ready.

1. Jump (unconditional and conditional)
2. Call (unconditional and conditional)
3. Return (unconditional and conditional)

**1. Jump Instructions –** The jump instruction transfers the program sequence to the memory address given in the operand based on the specified flag. Jump instructions are 2 types: Unconditional Jump Instructions and Conditional Jump Instructions.

**(a) Unconditional Jump Instructions:** Transfers the program sequence to the described memory address.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| JMP | address | Jumps to the address | JMP 2050 |

**(b) Conditional Jump Instructions:**Transfers the program sequence to the described memory address only if the condition in satisfied.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| JC | address | Jumps to the address if carry flag is 1 | JC 2050 |
| JNC | address | Jumps to the address if carry flag is 0 | JNC 2050 |
| JZ | address | Jumps to the address if zero flag is 1 | JZ 2050 |
| JNZ | address | Jumps to the address if zero flag is 0 | JNZ 2050 |
| JPE | address | Jumps to the address if parity flag is 1 | JPE 2050 |
| JPO | address | Jumps to the address if parity flag is 0 | JPO 2050 |
| JM | address | Jumps to the address if sign flag is 1 | JM 2050 |
| JP | address | Jumps to the address if sign flag 0 | JP 2050 |

**2. Call Instructions –** The call instruction transfers the program sequence to the memory address given in the operand. Before transferring, the address of the next instruction after CALL is pushed onto the stack. Call instructions are 2 types: Unconditional Call Instructions and Conditional Call Instructions.

**(a) Unconditional Call Instructions:** It transfers the program sequence to the memory address given in the operand.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| CALL | address | Unconditionally calls | CALL 2050 |

**(b) Conditional Call Instructions:** Only if the condition is satisfied, the instructions executes.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| CC | address | Call if carry flag is 1 | CC 2050 |
| CNC | address | Call if carry flag is 0 | CNC 2050 |
| CZ | address | Calls if zero flag is 1 | CZ 2050 |
| CNZ | address | Calls if zero flag is 0 | CNZ 2050 |
| CPE | address | Calls if parity flag is 1 | CPE 2050 |
| CPO | address | Calls if parity flag is 0 | CPO 2050 |
| CM | address | Calls if sign flag is 1 | CM 2050 |
| CP | address | Calls if sign flag is 0 | CP 2050 |

**3. Return Instructions –** The return instruction transfers the program sequence from the subroutine to the calling program. Return instructions are 2 types: Unconditional Jump Instructions and Conditional Jump Instructions.

**(a) Unconditional Return Instruction:** The program sequence is transferred unconditionally from the subroutine to the calling program.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| RET | none | Return from the subroutine unconditionally | RET |

**(b) Conditional Return Instruction:** The program sequence is transferred unconditionally from the subroutine to the calling program only is the condition is satisfied.

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
| --- | --- | --- | --- |
| RC | none | Return from the subroutine if carry flag is 1 | RC |
| RNC | none | Return from the subroutine if carry flag is 0 | RNC |
| RZ | none | Return from the subroutine if zero flag is 1 | RZ |
| RNZ | none | Return from the subroutine if zero flag is 0 | RNZ |
| RPE | none | Return from the subroutine if parity flag is 1 | RPE |
| RPO | none | Return from the subroutine if parity flag is 0 | RPO |
| RM | none | Returns from the subroutine if sign flag is 1 | RM |
| RP | none | Returns from the subroutine if sign flag is 0 | RP |