## Types of Instructions in 8085

**1.** Arithmetic instructions in 8085 microprocessors

Arithmetic Instructions are the instructions which perform basic arithmetic operations such as addition, subtraction and a few more. In 8085 microprocessor, the destination operand is generally the accumulator. In 8085 microprocessor, the destination operand is generally the accumulator.

Following is the table showing the list of arithmetic instructions:

In the table,
R stands for register
M stands for memory
Mc stands for memory contents
r.p. stands for register pair

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
|---|---|---|---|
| ADD | R | A = A + R | ADD B |
| ADD | M | A = A + Mc | ADD 2050 |
| ADI | 8-bit data | A = A + 8-bit data | ADI 50 |
| ADC | R | A = A + R + prev. carry | ADC B |
| ADC | M | A = A + Mc + prev. carry | ADC 2050 |
| ACI | 8-bit data | A = A + 8-bit data + prev. carry | ACI 50 |
| SUB | R | A = A – R | SUB B |
| SUB | M | A = A – Mc | SUB 2050 |
| SUI | 8-bit data | A = A – 8-bit data | SUI 50 |

| **OPCODE** | **OPERAND** | **EXPLANATION** | **EXAMPLE** |
|---|---|---|---|
| SBB | R | A = A – R – prev. carry | SBB B |
| SBB | M | A = A – Mc -prev. carry | SBB 2050 |
| SBI | 8-bit data | A = A – 8-bit data – prev. carry | SBI 50 |
| INR | R | R = R + 1 | INR B |
| INR | M | M = Mc + 1 | INR 2050 |
| INX | r.p. | r.p. = r.p. + 1 | INX H |
| DCR | R | R = R – 1 | DCR B |
| DCR | M | M = Mc – 1 | DCR 2050 |
| DCX | r.p. | r.p. = r.p. – 1 | DCX H |
| DAD | r.p. | HL = HL + r.p. | DAD H |

**2.** Logical instructions in 8085 microprocessor

Logical instructions are the instructions which perform basic logical operations such as AND, OR, etc. In 8085 microprocessor, the destination operand is always the accumulator. Here logical operation works on a bitwise level.

Following is the table showing the list of logical instructions:

| OPCODE | OPERAND | DESTINATION | EXAMPLE |
|---|---|---|---|
| ANA | R | A = A AND R | ANA B |
| ANA | M | A = A AND Mc | ANA 2050 |
| ANI | 8-bit data | A = A AND 8-bit data | ANI 50 |
| ORA | R | A = A OR R | ORA B |
| ORA | M | A = A OR Mc | ORA 2050 |
| ORI | 8-bit data | A = A OR 8-bit data | ORI 50 |
| XRA | R | A = A XOR R | XRA B |
| XRA | M | A = A XOR Mc | XRA 2050 |
| XRI | 8-bit data | A = A XOR 8-bit data | XRI 50 |
| CMA | none | A = 1’s compliment of A | CMA |
| CMP | R | Compares R with A and triggers the flag register | CMP B |

| OPCODE | OPERAND | DESTINATION | EXAMPLE |
|---|---|---|---|
| CMP | M | Compares Mc with A and triggers the flag register | CMP 2050 |
| CPI | 8-bit data | Compares 8-bit data with A and triggers the flag register | CPI 50 |
| RRC | none | Rotate accumulator right without carry | RRC |
| RLC | none | Rotate accumulator left without carry | RLC |
| RAR | none | Rotate accumulator right with carry | RAR |
| RAL | none | Rotate accumulator left with carry | RAR |
| CMC | none | Compliments the carry flag | CMC |
| STC | none | Sets the carry flag | STC |

In the table,
R stands for register
M stands for memory
Mc stands for memory contents

**3.** Data transfer instructions in 8085 microprocessor

Data tranfer instructions are the instructions which transfers data in the microprocessor. They are also called copy instructions.

Following is the table showing the list of logical instructions:

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
|---|---|---|---|
| MOV | Rd, Rs | Rd = Rs | MOV A, B |
| MOV | Rd, M | Rd = Mc | MOV A, 2050 |
| MOV | M, Rs | M = Rs | MOV 2050, A |
| MVI | Rd, 8-bit data | Rd = 8-bit data | MVI A, 50 |
| MVI | M, 8-bit data | M = 8-bit data | MVI 2050, 50 |
| LDA | 16-bit address | A = contents at address | LDA 2050 |
| STA | 16-bit address | contents at address = A | STA 2050 |
| LHLD | 16-bit address | directly loads at H & L registers | LHLD 2050 |
| SHLD | 16-bit address | directly stores from H & L registers | SHLD 2050 |

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
|---|---|---|---|
| LXI | r.p., 16-bit data | loads the specified register pair with data | LXI H, 3050 |
| LDAX | r.p. | indirectly loads at the accumulator A | LDAX H |
| STAX | 16-bit address | indirectly stores from the accumulator A | STAX 2050 |
| XCHG | none | exchanges H with D, and L with E | XCHG |
| PUSH | r.p. | pushes r.p. to the stack | PUSH H |
| POP | r.p. | pops the stack to r.p. | POP H |
| IN | 8-bit port address | inputs contents of the specified port to A | IN 15 |
| OUT | 8-bit port address | outputs contents of A to the specified port | OUT 15 |
|  |  |  |  |

In the table,
R stands for register
M stands for memory
r.p. stands for register pair

Instruction Execution in 8085

**Branching instructions in 8085 microprocessor**

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:

1. **Jump (unconditional and conditional)**
   **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 |

2. **(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 |

| OPCODE | OPERAND | EXPLANATION | EXAMPLE |
|---|---|---|---|
| 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 |

Addressing modes in 8085 microprocessor

**Types of addressing modes –**

In 8085 microprocessor there are 5 types of addressing modes:

1. **Immediate Addressing Mode –**
   In immediate addressing mode the source operand is always data. If the data is 8-bit, then the instruction will be of 2 bytes, if the data is of 16-bit then the instruction will be of 3 bytes.
   **Examples:**
   MVI B 45 (move the data 45H immediately to register B)
   LXI H 3050 (load the H-L pair with the operand 3050H immediately)
   JMP address (jump to the operand address immediately)

2. **Register Addressing Mode –**
   In register addressing mode, the data to be operated is available inside the register(s) and register(s) is(are) operands. Therefore, the operation is performed within various registers of the microprocessor.
   **Examples:**
   MOV A, B (move the contents of register B to register A)
   ADD B (add contents of registers A and B and store the result in register A)
   INR A (increment the contents of register A by one)

3. **Direct Addressing Mode –**
   In direct addressing mode, the data to be operated is available inside a memory location and that memory location is directly specified as an operand. The operand is directly available in the instruction itself.

**Examples:**
LDA 2050 (load the contents of memory location into accumulator A)
LHLD address (load contents of 16-bit memory location into H-L register pair)
IN 35 (read the data from port whose address is 01)

4. **Register Indirect Addressing Mode –**
   In register indirect addressing mode, the data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair.

**Examples:**
MOV A, M (move the contents of the memory location pointed by the H-L pair to the accumulator)
LDAX B (move contents of B-C register to the accumulator)
LXIH 9570 (load immediate the H-L pair with the address of the location 9570)

5. **Implied/Implicit Addressing Mode –**
   In implied/implicit addressing mode the operand is hidden and the data to be operated is available in the instruction itself.

**Examples:**
CMA (finds and stores the 1's complement of the contains of accumulator A in A)
RRC (rotate accumulator A right by one bit)
RLC (rotate accumulator A left by one bit)