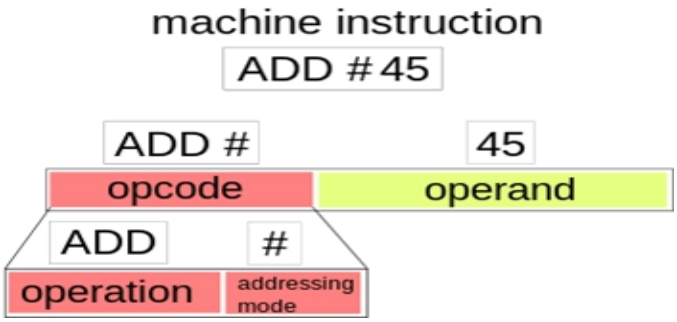
Microprocessors and Micro-controllers

Class 5

Instruction Set Classification

- An instruction is a command given to a computer to perform a specified operation on a given data. It is a binary pattern designed inside a microprocessor to perform a specific function.
- Each Instruction consists of two parts:
    1. An Opcode (Operation Code) and
    2. An Operand



- The first part of an instruction, known as opcode specifies the operation to be performed and the second part, called operand is the data on which computer performs the specified operation.
- The entire group of instructions, called the **instruction set**, determines what functions the microprocessor can perform. The Instruction set of 8085 microprocessor consists of **74 operation codes** (e.g. MOV) and **246 Instructions** (e.g. MOV A,B).
- These instructions can be classified into the following five functional categories:

    1. Data transfer (copy) operations
    2. Arithmetic operations
    3. Logical operations
    4. Branching operations and
    5. Machine-control operations.

1. Data Transfer (Copy) Operations

This group of instructions copy data from a location called a source to another location called a destination, without modifying the contents of the source. The various types of data transfer are listed below together with examples of each type:

Types	Examples
1. Between Registers.	1. Copy the contents of the register B into register D.
2. Specific data byte to a register or a memory location.	2. Load register B with the data byte 32H.
3. Between a memory location and a register.	3. From a memory location 2000H to register B.
4. Between an I/O device and the accumulator.	4. From an input keyboard to the accumulator.

Examples are: MOV, MVI, LXI, LDA, STA etc

2. Arithmetic Operations

These instructions perform arithmetic operations such as addition, subtraction, increment, and decrement.

Addition - Any 8-bit number, or the contents of a register or the contents of a memory location can be added to the contents of the accumulator and the sum is stored in the accumulator. No two other 8-bit registers can be added directly (e.g., the contents of register B cannot be added directly to the contents of the register C). The instruction DAD is an exception; it adds 16-bit data directly in register pairs.

Subtraction - Any 8-bit number, or the contents of a register, or the contents of a memory location can be subtracted from the contents of the accumulator and the results stored in the accumulator. The subtraction is performed in 2's compliment, and the results if negative, are expressed in 2's complement. No two other registers can be subtracted directly.

Increment/Decrement - The 8-bit contents of a register or a memory location can be incremented or decrement by 1. Similarly, the 16-bit contents of a register pair (such as BC) can be incremented or decrement by 1. These increment and decrement operations differ from addition and subtraction in an important way; i.e., they can be performed in any one of the registers or in a memory location.

Examples are: ADD, SUB, INK, and DAD etc.

3. Logical Operations

These instructions perform various logical operations with the contents of the accumulator.

AND, OR Exclusive-OR - Any 8-bit number, or the contents of a register, or of a memory location can be logically ANDed, Ored, or Exclusive-ORed with the contents of the accumulator. The results are stored in the accumulator.

Rotate- Each bit in the accumulator can be shifted either left or right to the next position.

Compare- Any 8-bit number, or the contents of a register, or a memory location can be compared for equality, greater than, or less than, with the contents of the accumulator.

Complement - The contents of the accumulator can be complemented. All 0s are replaced by 1s and all 1s are replaced by 0s.

Examples are: ANA, XRA, ORA, CMP, and RAL etc.

4. Branching Operations

This group of instructions alters the sequence of program execution either conditionally or unconditionally.

Jump - Conditional jumps are an important aspect of the decision-making process in the programming. These instructions test for a certain conditions (e.g., Zero or Carry flag) and alter the program sequence when the condition is met. In addition, the instruction set includes an instruction called *unconditional jump.*

Call, Return, and Restart - These instructions change the sequence of a program either by calling a subroutine or returning from a subroutine. The conditional Call and Return instructions also can test condition flags.

Examples are: JMP, JC, JZ, CALL, CZ, RST etc.

5. Machine Control Operations

These instructions control machine functions such as Halt, Interrupt, or do nothing.

Examples are: IN, OUT. PUSH, POP, HLT etc.

Classification of Instruction Formats

Microprocessor Instructions can be classified, based on Instruction word Size and based on the number of operand address, the instruction contain.

Based on Instruction word size

- The binary codes for all instructions are not of the same length.
- These instructions and data are fed to the computer in binary format (0 and 1), known as machine language, but they are written in hexadecimal form, for the convenience of the user.
- Instructions are classified into the following three types according to their word length (i.e. length of the binary code).

(i) Single Byte instruction (Binary code of instruction in one byte)

- Here the single byte contains the instruction (opcode) as well as the operand

opcode	operand	Binary Code	Hex Code	Task
MOV	C, A	0100 1111	4FH	Copy the contents of the accumulator in the register C
ADD	B	1000 0000	80H	Adds the contents of the register B to the contents of the Accumulator
HLT		0111 0110		Terminates the program

(ii) Two Byte instruction (Machine code is of two bytes)

- Here the first byte contains the instruction (opcode) and the second byte specifies the operand

opcode	operand	Binary Code	Hex Code	Task
MVI	A, 7FH	0011 1110	3E	MVI - Move Immediate 8-bit, to register(R) or memory (M)
		0111 1111	7FH	
ADI	0FH	1100 0110	C6	ADI - Add immediate 8-bit to ACC and res stored back to ACC
		0000 0000	0F	

(iii) Three Byte instruction

- Here the first byte contains the instruction (opcode) and the second and third byte specifies the 16-bit address of the operand.
- Here the second byte is the lower-order address and the third byte is the higher-order address.

opcode	operand	Binary Code	Hex Code	Task
LXI	H, 2500H	0010 0001	21	Loads H-L pair with 16-bit data 2500H Note that the lower bits are in 2nd byte and the higher bits in 3rd byte.
		0000 0000	00H	
		0010 0101	25H	
LDA	0FH	0011 1010	3A	Loads accumulator with the contents of the memory location 2400H
		0000 0000	0F	
		0010 0100	24H	