1. Opcode and Operands

Opcode (Operation Code)

Definition: The part of a machine instruction that tells the CPU what operation to perform.

Example: MOV, ADD, SUB, INT

Role: It defines the action (e.g., move data, add numbers, call interrupt).

Operands

Definition: The values or data being manipulated by the opcode.

Types of Operands:

- Immediate: Constant value (e.g., MOV AX, 5)
- Register: CPU register (e.g., MOV AX, BX)
- Memory Address: Accessing data from memory (e.g., MOV AX, [1234H])
- Indirect Addressing: Through pointers (e.g., MOV AX, [SI])

Example Instruction Breakdown

MOV AX, 1234h

Opcode: MOV -> Move data

Operands: AX (destination), 1234h (source)

2. Routines and Subroutines

Routine

A general term for a reusable block of code.

Used to perform a specific task and often called multiple times.

Subroutine (Function Call)

A named block (like a function) that you can call using a special instruction.

Helps to modularize code.

Call and Return in Assembly

```
CALL MySubroutine   ; Jump to subroutine
...
RET                 ; Return from subroutine
```

Label

A label is a name assigned to a line or block of code (acts like a function name).

```
MySubroutine:
  ; some instructions
  RET
```

3. Interrupts

Definition:

An interrupt is a signal to the CPU to pause current execution and run a special Interrupt Service Routine (ISR).

After handling, the CPU returns to the previous code.

Types of Interrupts:

- Hardware Interrupts: Triggered by devices (keyboard, mouse, etc.)

- Software Interrupts: Triggered by instructions like INT

DOS Interrupt Example

INT 21h ; Calls DOS services like print, read, etc.

Structure:

The interrupt number (e.g., INT 21h) tells the CPU which handler to run.

Often used to call system routines for input/output or file operations.