Unit – 3

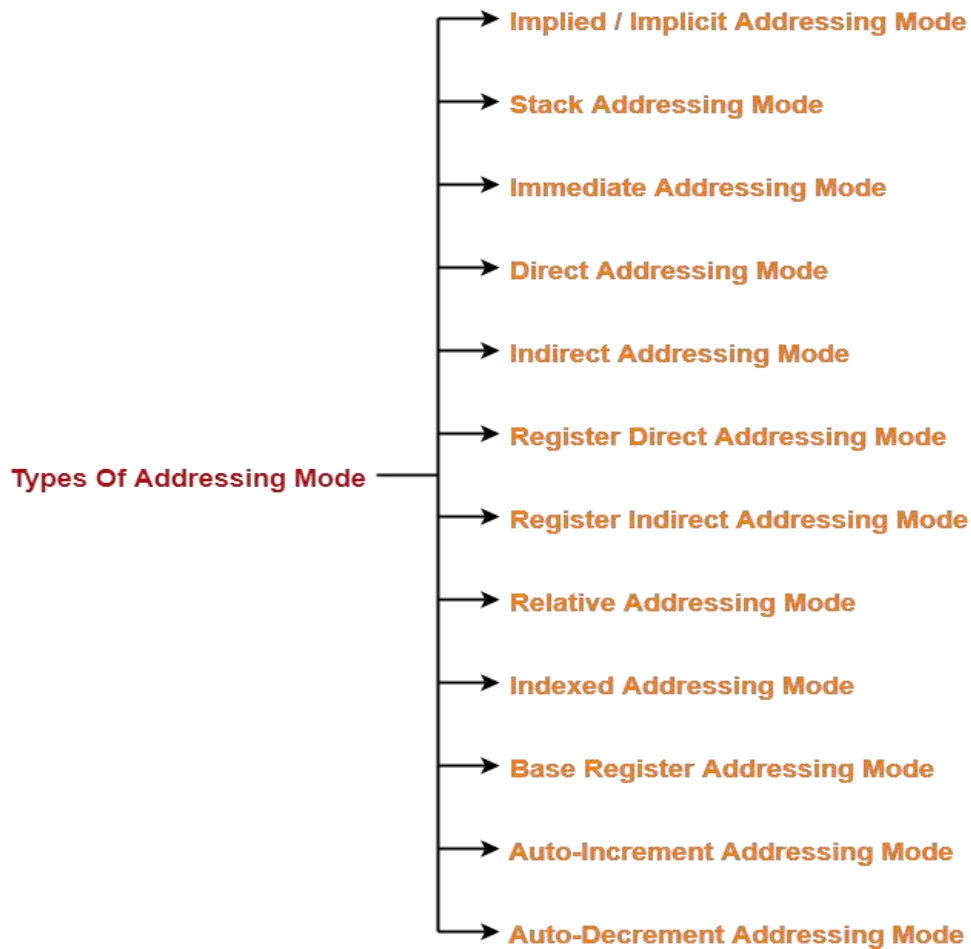## Types of Addressing Modes-

In computer architecture, there are following types of addressing modes-

```
                                    → Implied / Implicit Addressing Mode
                                    → Stack Addressing Mode
                                    → Immediate Addressing Mode
                                    → Direct Addressing Mode
                                    → Indirect Addressing Mode
                                    → Register Direct Addressing Mode
Types Of Addressing Mode ———        → Register Indirect Addressing Mode
                                    → Relative Addressing Mode
                                    → Indexed Addressing Mode
                                    → Base Register Addressing Mode
                                    → Auto-Increment Addressing Mode
                                    → Auto-Decrement Addressing Mode
```

1. Implied / Implicit Addressing Mode
2. Stack Addressing Mode
3. Immediate Addressing Mode
4. Direct Addressing Mode
5. Indirect Addressing Mode
6. Register Direct Addressing Mode
7. Register Indirect Addressing Mode
8. Relative Addressing Mode
9. Indexed Addressing Mode
10. Base Register Addressing Mode
11. Auto-Increment Addressing Mode
12. Auto-Decrement Addressing Mode

1. Implied Addressing Mode-

In this addressing mode,
- The definition of the instruction itself specify the operands implicitly.
- It is also called as implicit addressing mode.

Examples-
- The instruction "Complement Accumulator" is an implied mode instruction.

- In a stack organized computer, Zero Address Instructions are implied mode instructions.

(Since operands are always implied to be present on the top of the stack)

2. Stack Addressing Mode-

In this addressing mode,

- The operand is contained at the top of the stack.

Example-

ADD

- This instruction simply pops out two symbols contained at the top of the stack.
- The addition of those two operands is performed.
- The result so obtained after addition is pushed again at the top of the stack.

3. Immediate Addressing Mode-

In this addressing mode,

- The operand is specified in the instruction explicitly.
- Instead of address field, an operand field is present that contains the operand.

Examples-

- ADD 10 will increment the value stored in the accumulator by 10.
- MOV R #20 initializes register R to a constant value 20.

4. Direct Addressing Mode-

In this addressing mode,

- The address field of the instruction contains the effective address of the operand.
- Only one reference to memory is required to fetch the operand.
- It is also called as **absolute addressing mode**.
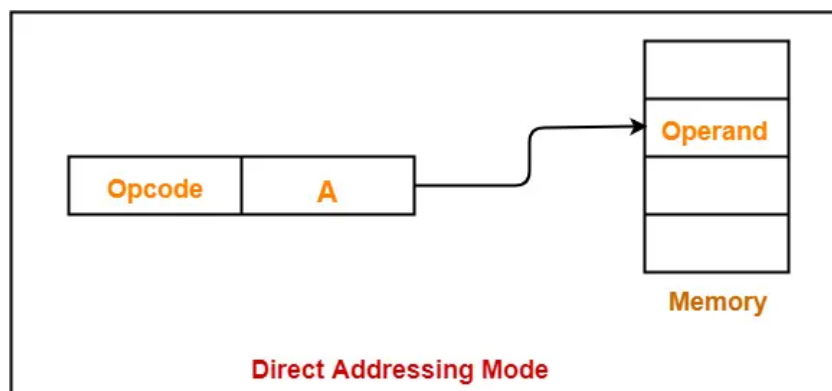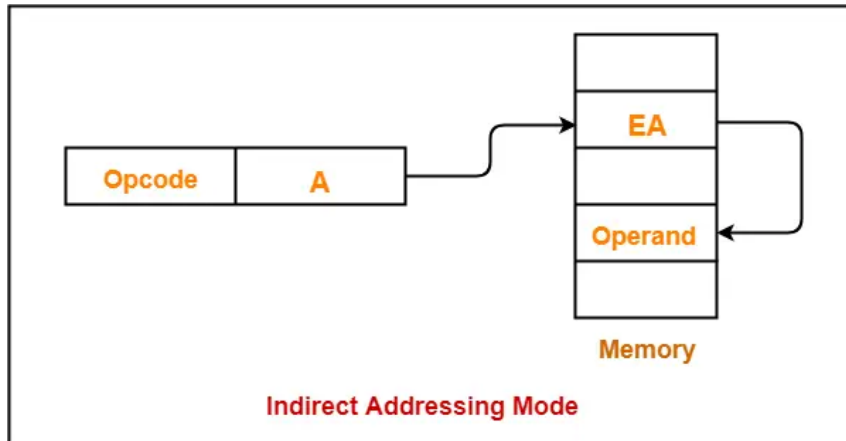

Example-

- ADD X will increment the value stored in the accumulator by the value stored at memory location X.

$$AC \leftarrow AC + [X]$$

5. Indirect Addressing Mode-

In this addressing mode,
- The address field of the instruction specifies the address of memory location that contains the effective address of the operand.
- Two references to memory are required to fetch the operand.

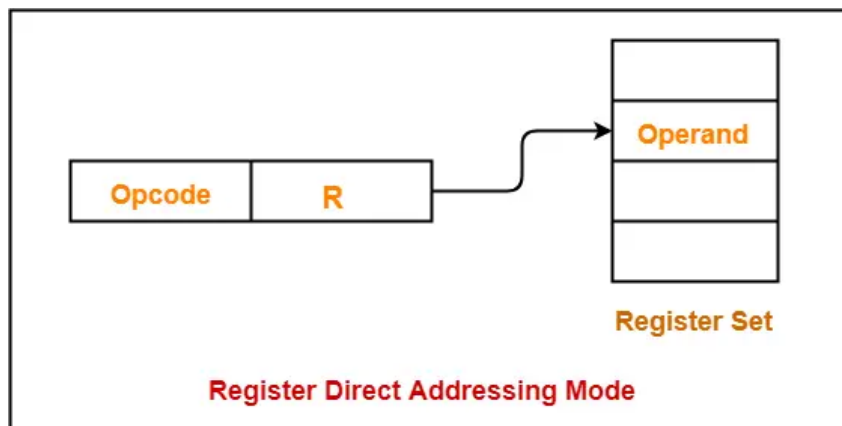**Indirect Addressing Mode**

Example-
- ADD X will increment the value stored in the accumulator by the value stored at memory location specified by X.

$$AC \leftarrow AC + [[X]]$$

6. Register Direct Addressing Mode-

In this addressing mode,

- The operand is contained in a register set.
- The address field of the instruction refers to a CPU register that contains the operand.
- No reference to memory is required to fetch the operand.

Register Direct Addressing Mode

Example-
- ADD R will increment the value stored in the accumulator by the content of register R.

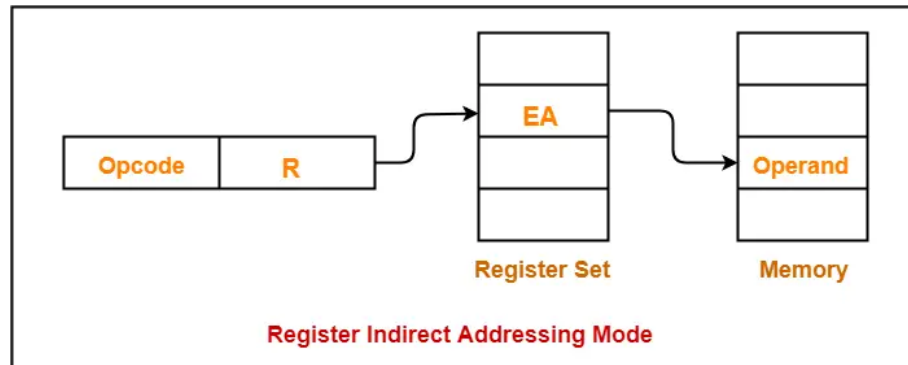$$AC \leftarrow AC + [R]$$

NOTE-

It is interesting to note-
- This addressing mode is similar to direct addressing mode.
- The only difference is address field of the instruction refers to a CPU register instead of main memory.

7. Register Indirect Addressing Mode-

In this addressing mode,

- The address field of the instruction refers to a CPU register that contains the effective address of the operand.
- Only one reference to memory is required to fetch the operand.

Register Indirect Addressing Mode

Example-
- ADD R will increment the value stored in the accumulator by the content of memory location specified in register R.

$$AC \leftarrow AC + [[R]]$$

NOTE-
It is interesting to note-
- This addressing mode is similar to indirect addressing mode.
- The only difference is address field of the instruction refers to a CPU register.

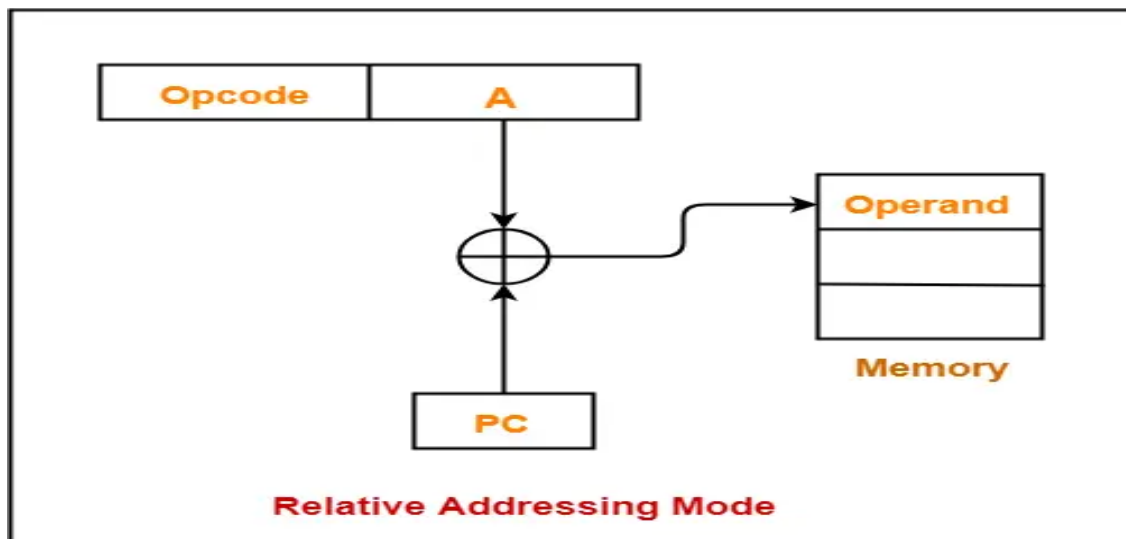## 8. Relative Addressing Mode-

In this addressing mode,

- Effective address of the operand is obtained by adding the content of program counter with the address part of the instruction.

Effective Address

= Content of Program Counter + Address part of the instruction

NOTE-
- **Program counter** (PC) always contains the address of the next instruction to be executed.
- After fetching the address of the instruction, the value of program counter immediately increases.
- The value increases irrespective of whether the fetched instruction has completely executed or not.
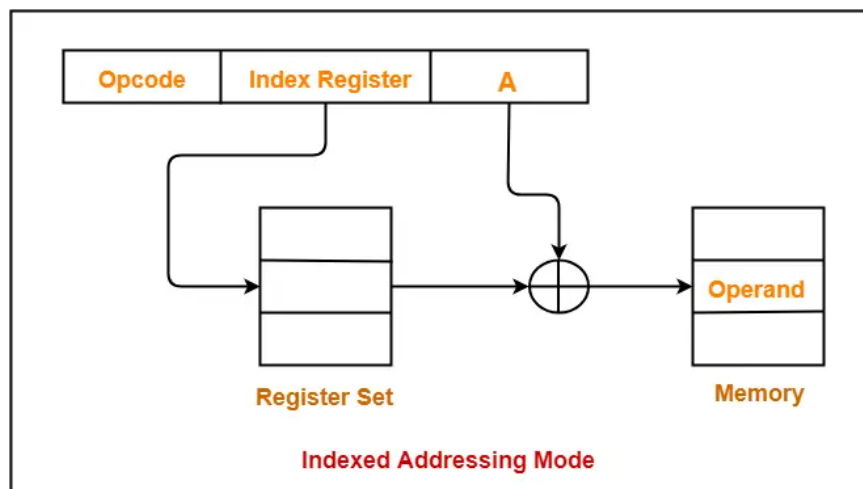
Relative Addressing Mode

9. Indexed Addressing Mode-

In this addressing mode,

- Effective address of the operand is obtained by adding the content of index register with the address part of the instruction.

Effective Address

= Content of Index Register + Address part of the instruction
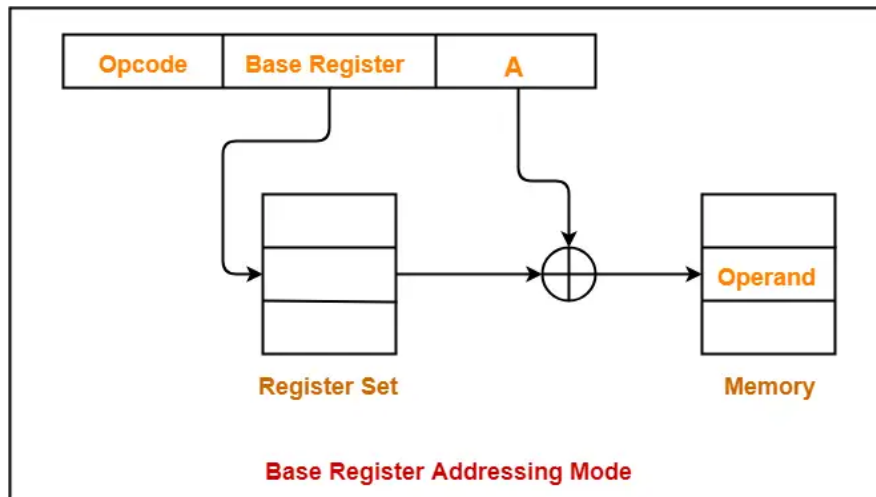

Indexed Addressing Mode

10. Base Register Addressing Mode-

In this addressing mode,

- Effective address of the operand is obtained by adding the content of base register with the address part of the instruction.

Effective Address

= Content of Base Register + Address part of the instruction
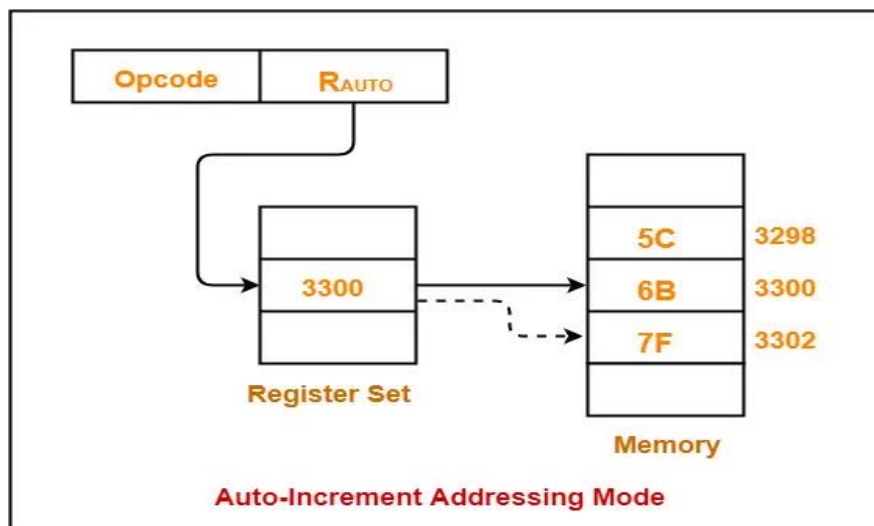
Base Register Addressing Mode

11. Auto-Increment Addressing Mode-

- This addressing mode is a special case of Register Indirect Addressing Mode where-

Effective Address of the Operand

= Content of Register

In this addressing mode,

- After accessing the operand, the content of the register is automatically incremented by step size 'd'.
- Step size 'd' depends on the size of operand accessed.
- Only one reference to memory is required to fetch the operand.

Auto-Increment Addressing Mode

Example-

Assume operand size = 2 bytes.
Here,

- After fetching the operand 6B, the instruction register R_{AUTO} will be automatically incremented by 2.
- Then, updated value of R_{AUTO} will be $3300 + 2 = 3302$.
- At memory address 3302, the next operand will be found.

In auto-increment addressing mode,

- First, the operand value is fetched.
- Then, the instruction register R_{AUTO} value is incremented by step size 'd'.
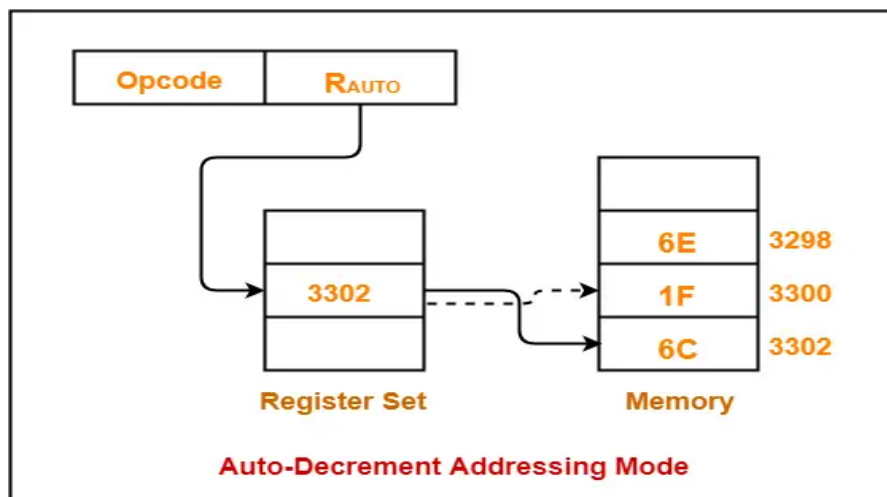
12. Auto-Decrement Addressing Mode-

- This addressing mode is again a special case of Register Indirect Addressing Mode where-

---

Effective Address of the Operand

= Content of Register – Step Size

In this addressing mode,

- First, the content of the register is decremented by step size 'd'.
- Step size 'd' depends on the size of operand accessed.
- After decrementing, the operand is read.
- Only one reference to memory is required to fetch the operand.

Example-

Auto-Decrement Addressing Mode

Assume operand size = 2 bytes.

Here,

- First, the instruction register R_{AUTO} will be decremented by 2.
- Then, updated value of R_{AUTO} will be $3302 - 2 = 3300$.
- At memory address 3300, the operand will be found.

NOTE-

In auto-decrement addressing mode,

- First, the instruction register R_{AUTO} value is decremented by step size 'd'.
- Then, the operand value is fetched.

Applications of Addressing Modes-

Addressing Modes	Applications
Immediate Addressing Mode	• To initialize registers to a constant value
Direct Addressing Mode **and** **Register Direct Addressing Mode**	• To access static data • To implement variables
Indirect Addressing Mode **and** **Register Indirect Addressing Mode**	• To implement pointers because pointers are memory locations that store the address of another variable • To pass array as a parameter because array name is the base address and pointer is needed to point the address
Relative Addressing Mode	• For program relocation at run time i.e. for position independent code • To change the normal sequence of execution of instructions • For branch type instructions since it directly updates the program counter
Index Addressing Mode	• For array implementation or array addressing • For records implementation
Base Register Addressing Mode	• For writing relocatable code i.e. for relocation of program in memory even at run time • For handling recursive procedures
Auto-increment Addressing Mode **and** **Auto-decrement Addressing Mode**	• For implementing loops • For stepping through arrays in a loop • For implementing a stack as push and pop