#Addressing Modes.

These modes basically deals with the way how a digital computer fetches the operands from memory / peripheral / register.

To do any operation, operands can be fetched from

1. Memory 2. Peripheral (Keyboard) 3. Register.

* A General Instruction format

Mendatory	Mendatory	optional.
opcode	mode	Address
Specify operation to be performed	used to locate the operand needed for the operation	if present, ~~tells~~ may specify a memory address or processor register.

Addressing Modes.

1. Implied Mode: Operands are implicitly specified in the definition of the instruction.

e.g. CMA (Compliment Acc. it implies operand is present in Acc.)

CLA (Clear Acc).

In Stack organized system, instruction li

ADD (Addition)

SUB (Subtraction)

lie in the category of Implied Addressing mode. Since the operands are ~~present~~ implied to be on the top of the stack.

2. Immediate Mode: The operand is specified in the instruction.

e.g.

MVI A, #10H ← Given ~~no~~ number is in Hex.

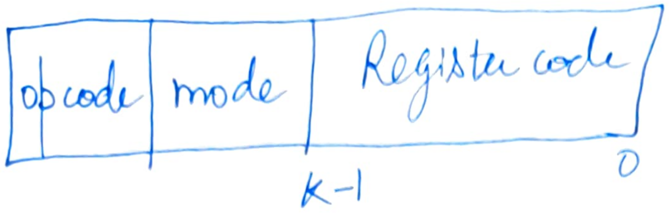
(move Immediately) Acc. operand.

3. Register Mode: Operand is ~~present~~ specified in register.

opcode	mode	Register code
		K-1 0

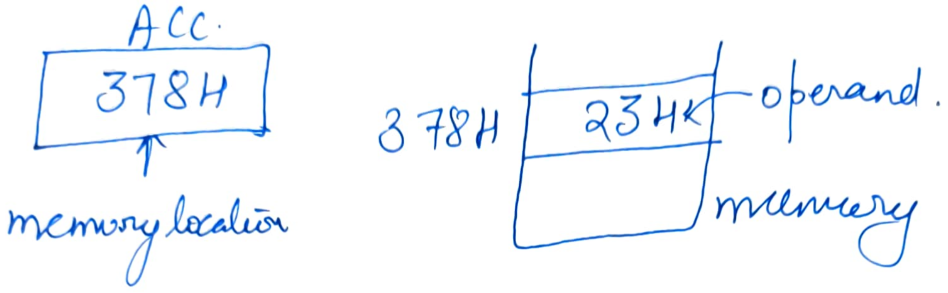
An K-bit field specified one of 2^k registers that hold the operand.

4. Register Indirect Mode: Instruction specify a register that gives the address of the operand in memory.



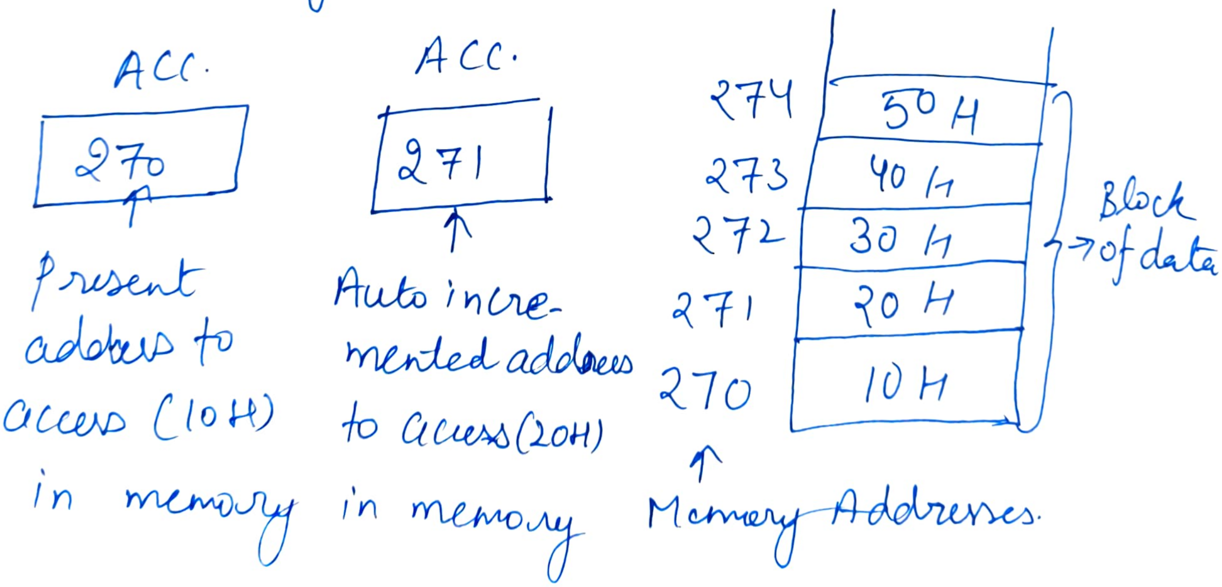
Let K-bit address denote Reg. ACC.
Now acc. will contain add of the operand

e.g.



6. Autoincrement or Autodecrement mode:

This mode is similar to Register Indirect mode except. that the register is incremented or decremented after its value is used to access memory.

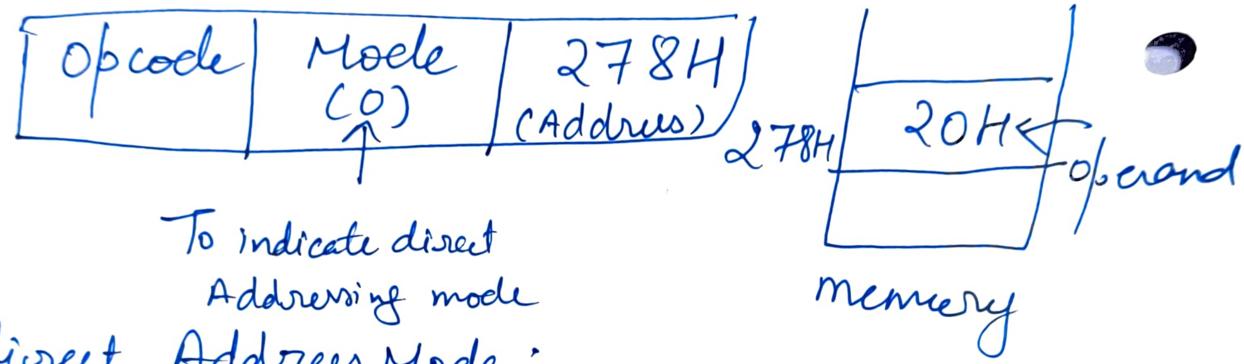

ACC.
270
Present address to access (10H) in memory
ACC.
271
Auto incremented address to access (20H) in memory
274
273
272
271
270
50 H
40 H
30 H
20 H
10 H
Block of data
Memory Addresses.

6 Direct Address Mode:

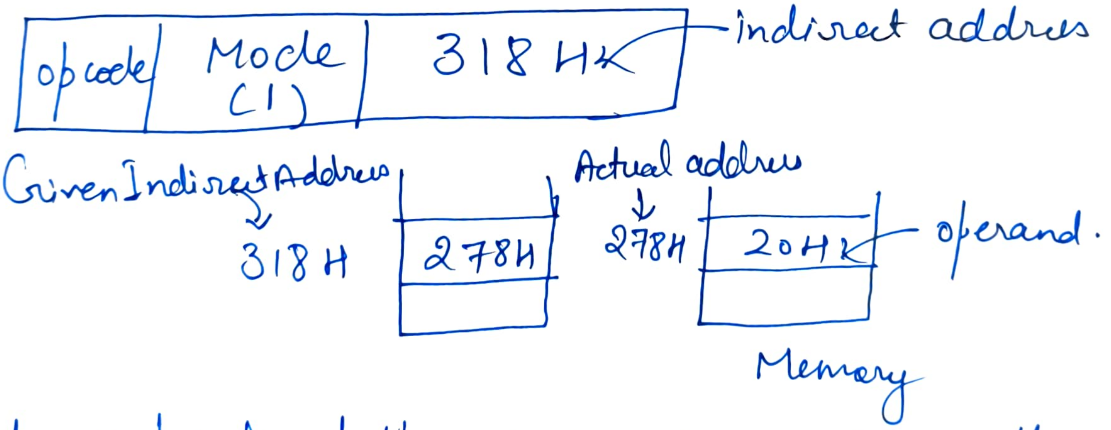
opcode	Mode (0)	(Effective Add) ADDRESS

Here effective ~~ADD~~ address of the Operand is specified in the instruction in address part. ~~from~~ Using this address the operand is fetched from memory.

e.g



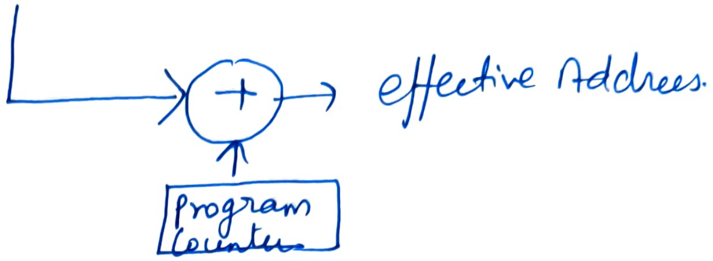
7. Indirect Address Mode:



Address part of the instruction contains the indirect address of the operand.

lative Address Mode:

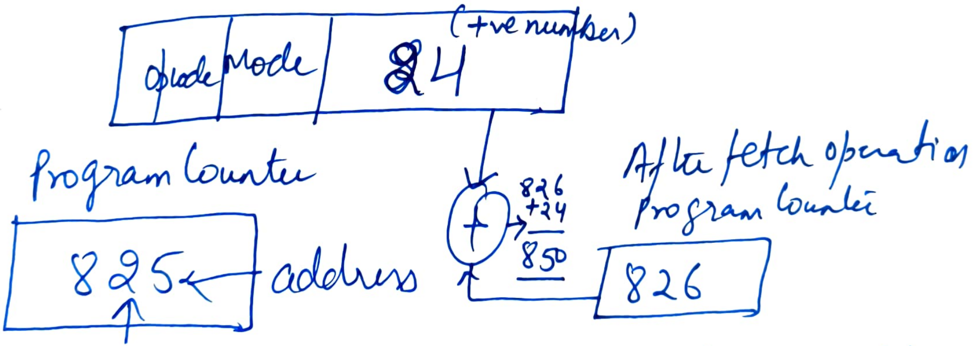
opcode	mode	Address (a signed number +ve or -ve) e.g. 2,3,4 or -1,-2,-3



The content of the program Counter are added ~~with~~ to the addres part of the instruction to get effective addres.

The address part of the Instruction contains a signed number -ve or +ve this number is added to Program counter.

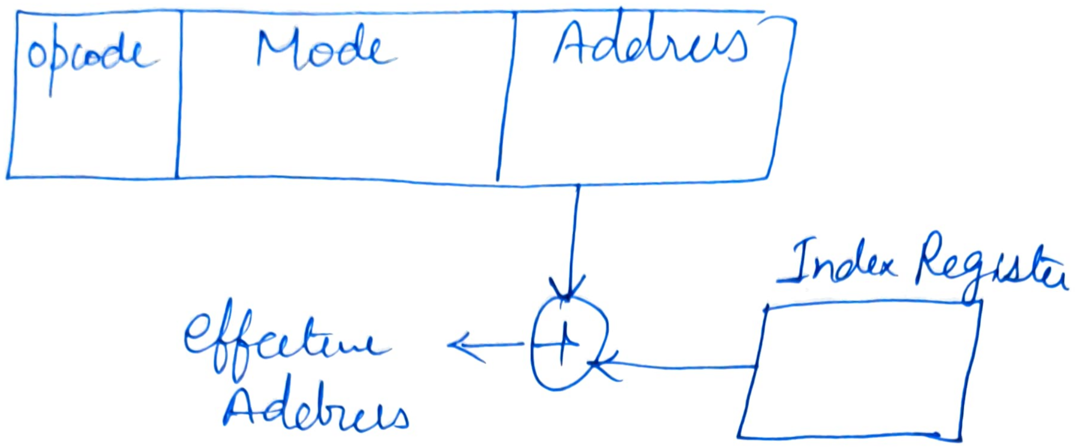
e.g.



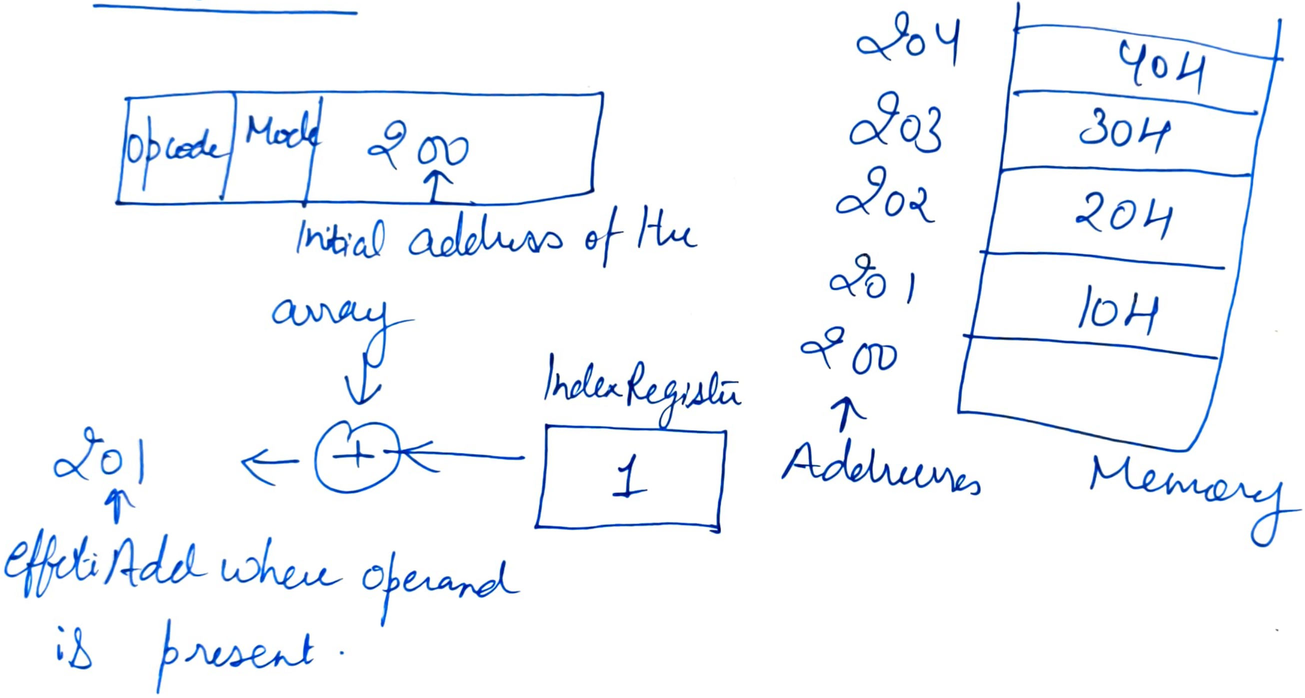
Instruction at the given addres is fetched. Now the Program Counter will contain next address in seq. that will be added to Given signed number in Instruction (24)

826 + 24 → 850 (effective address).

9. Indexed Addressing Mode



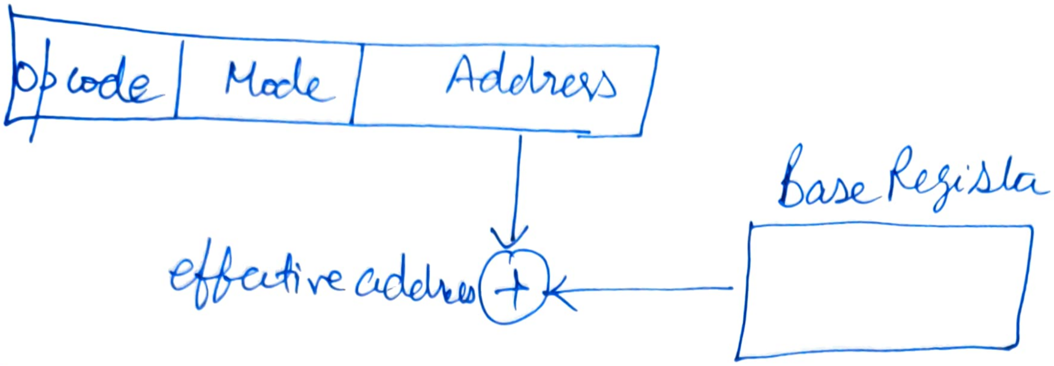
illustration



Address part of the instruction contains the intial address of the array (200 in this example). And Index register contains the diff. b/w inital address and address of the operand in memory (1 in this example)

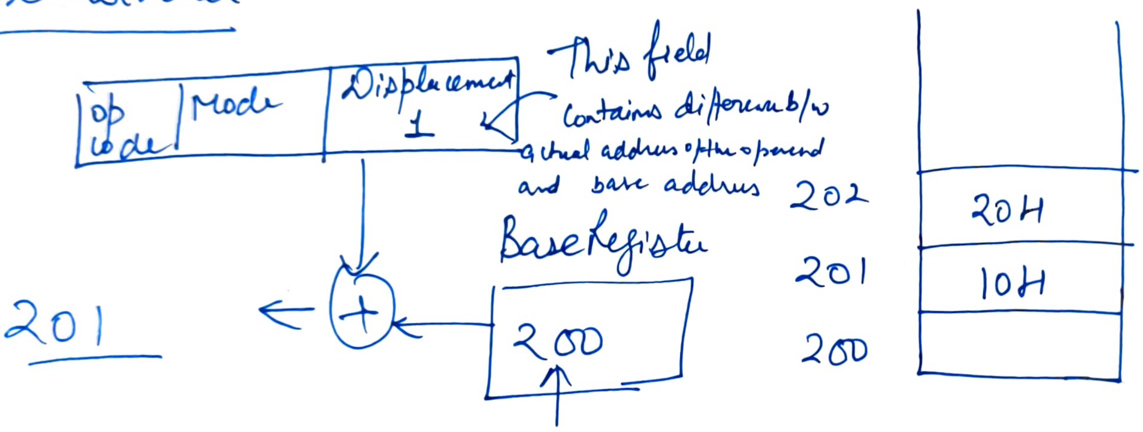
* Note: Index register can be incremented to have access to consective operand

Register addressing Mode



Similar to Indexed addressing mode. except Index register is now replaced with base register.

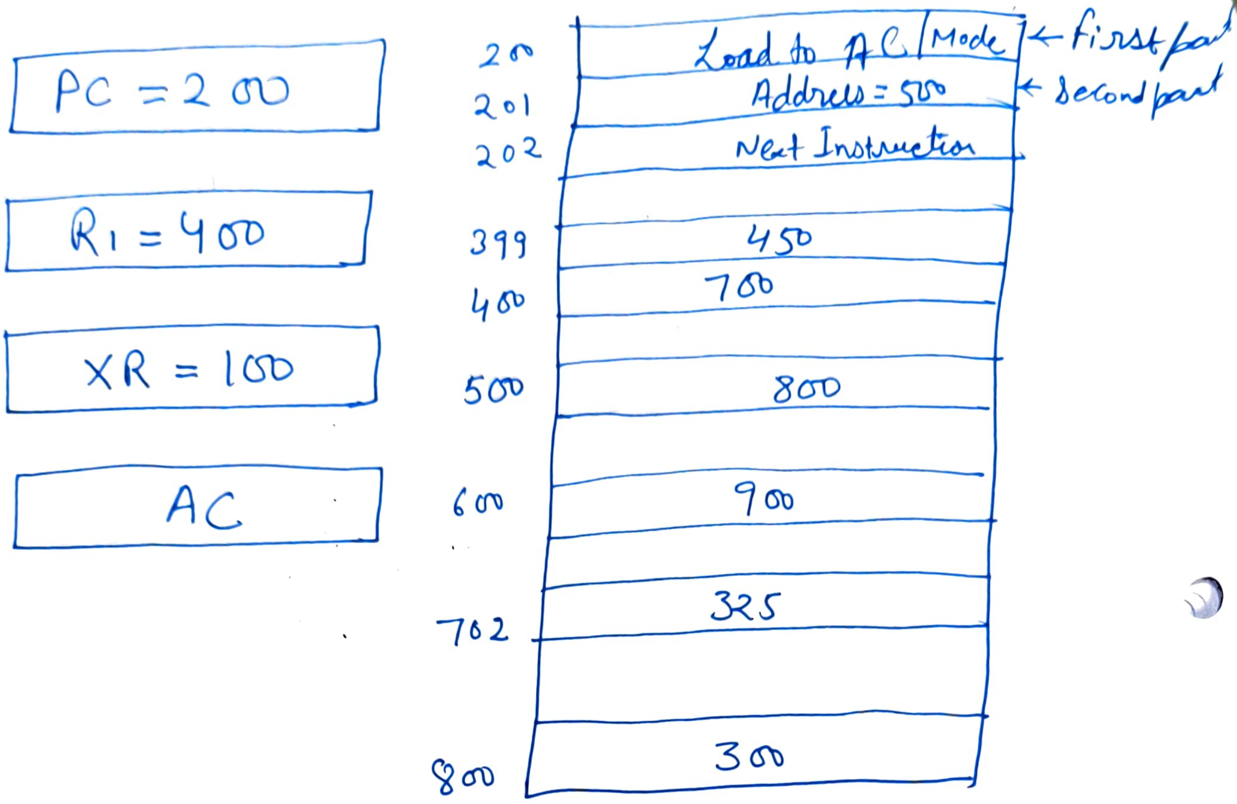
Illustration



This field contains difference b/w actual address of the operand and base address

Here Initial address IS Contained in base Register

Numerical Example



Instruction : Load to AC Mode Address = 500

Its two word

Ist word : Load to AC Mode [200]

2^{nd} word : Address = 500 [201]

Mode field will specify one of the various addressing modes. for every mode, effective address will be calculated in different manner. However, ~~after every~~ irrespective of the addressing mode given operand must be loaded into the AC.

1. Direct Address mode.

Effective Address (EA) = 500

Operand at 500 = 800

2. Immediate mode

EA = 201

Operand at 201 = 500

3. Indirect mode.

Given add = 500

Eff. Add = 800

operand at 800 = 300

4. In relative mode

EA = PC + Given add.

(after fetch of Ist Inst.)

202 + 500 = 702

Operand at 702 = 325

5. In Index mode

EA = 500 + XR = 600 [XR=100]

operand at 600 = 900

6. Register mode

Operand in R_1 = 400

7. Register Indirect mode.

EA in R1 = 400

operand at 400 = 700