CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Instruction & Addressing Modes

Lecture No.- 06

By- Vishvadeep Gothi sir

Recap of Previous Lecture

- Topic: Instruction Cycle
- Topic: Fetch & Execution Cycle
- Topic: Branch Instruction
- Topic: Addressing Modes

Topics to be Covered

- **Topic** Addressing Modes
- **Topic** PC Relative Mode
- **Topic** Questions on Addressing Modes

Topic : Implied Mode

The opcode definition itself defines the operand

Opcode	Mode	Address

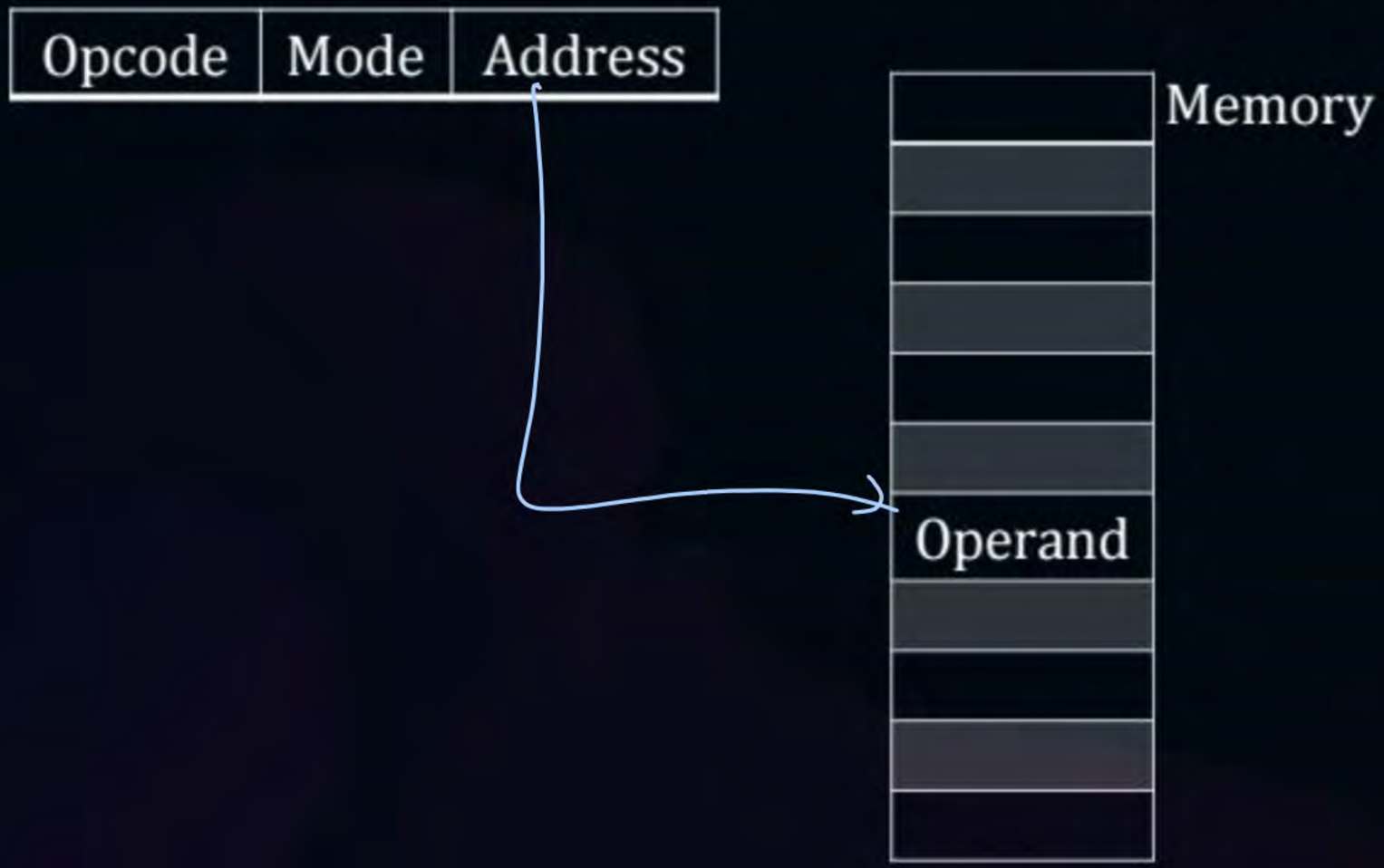
Topic : Immediate Mode

The address field of instruction specifies the operand value

Opcode	Mode	Address

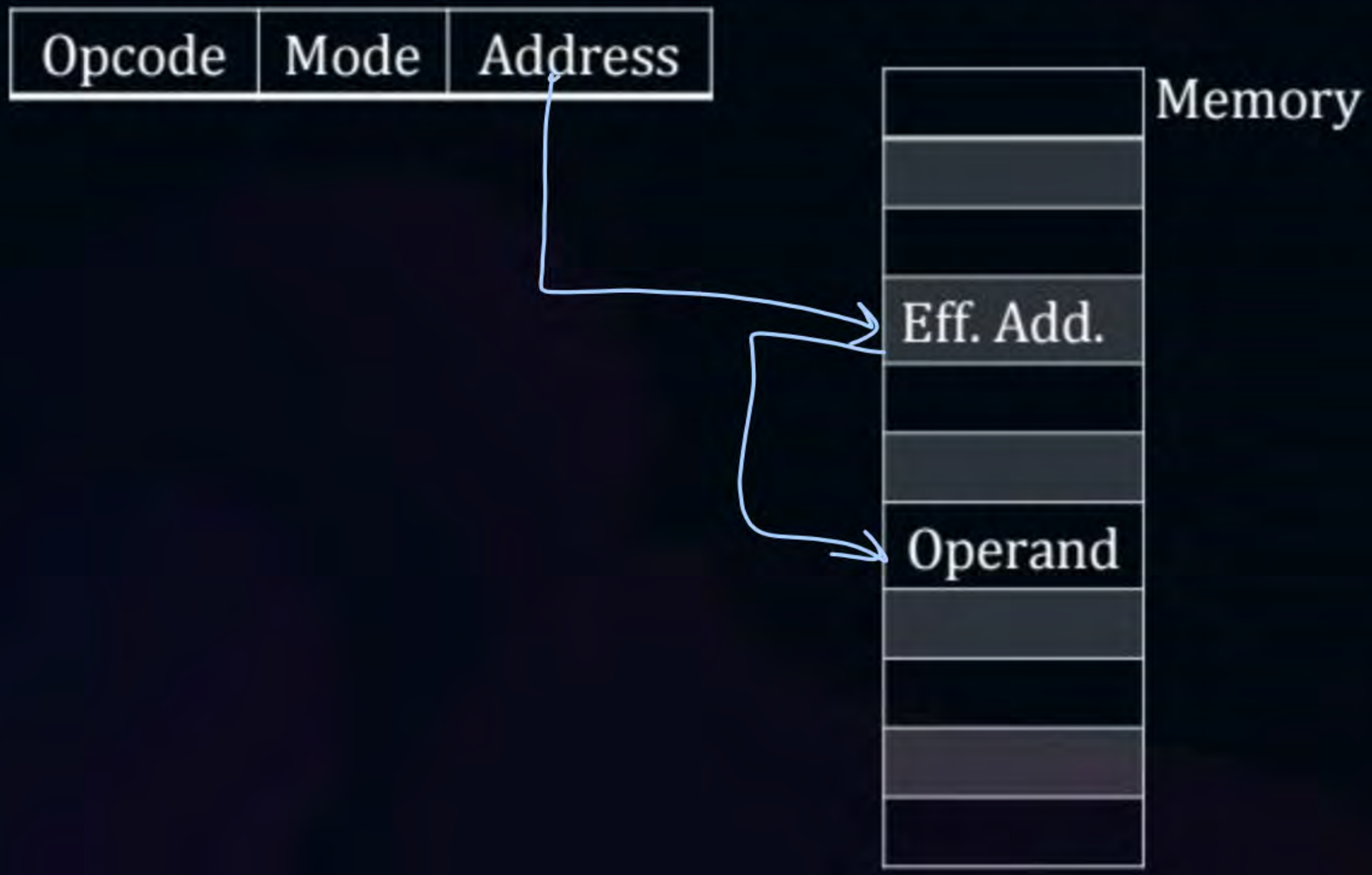
Topic : Direct Mode

The address field of instruction specifies the effective address



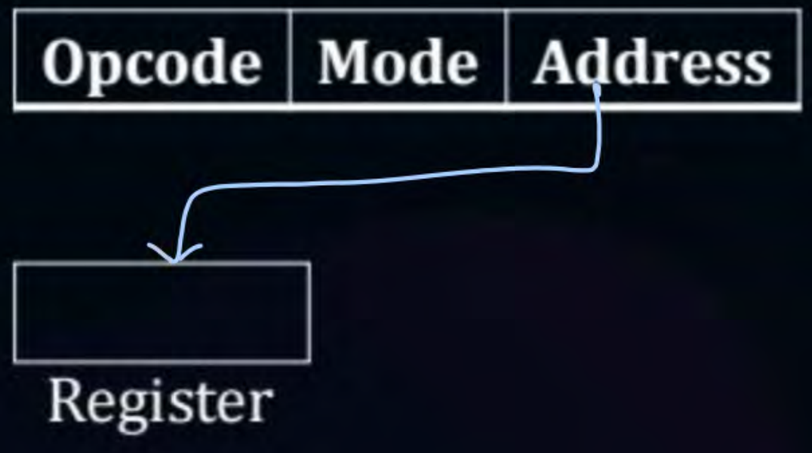
Topic : Indirect Mode

The address field of instruction specifies the address of effective address



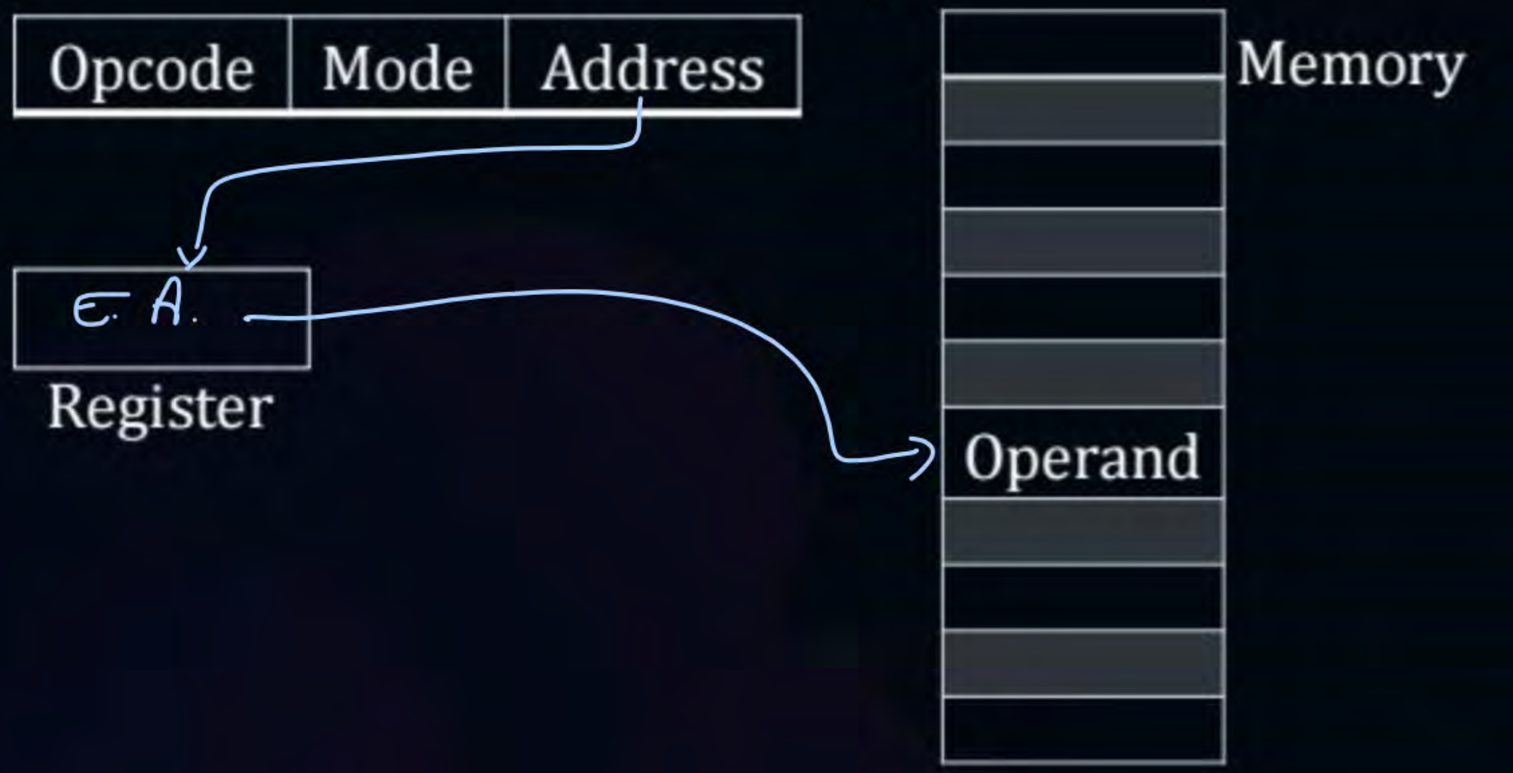
Topic : Register Mode

The address field of instruction specifies a register which holds operand



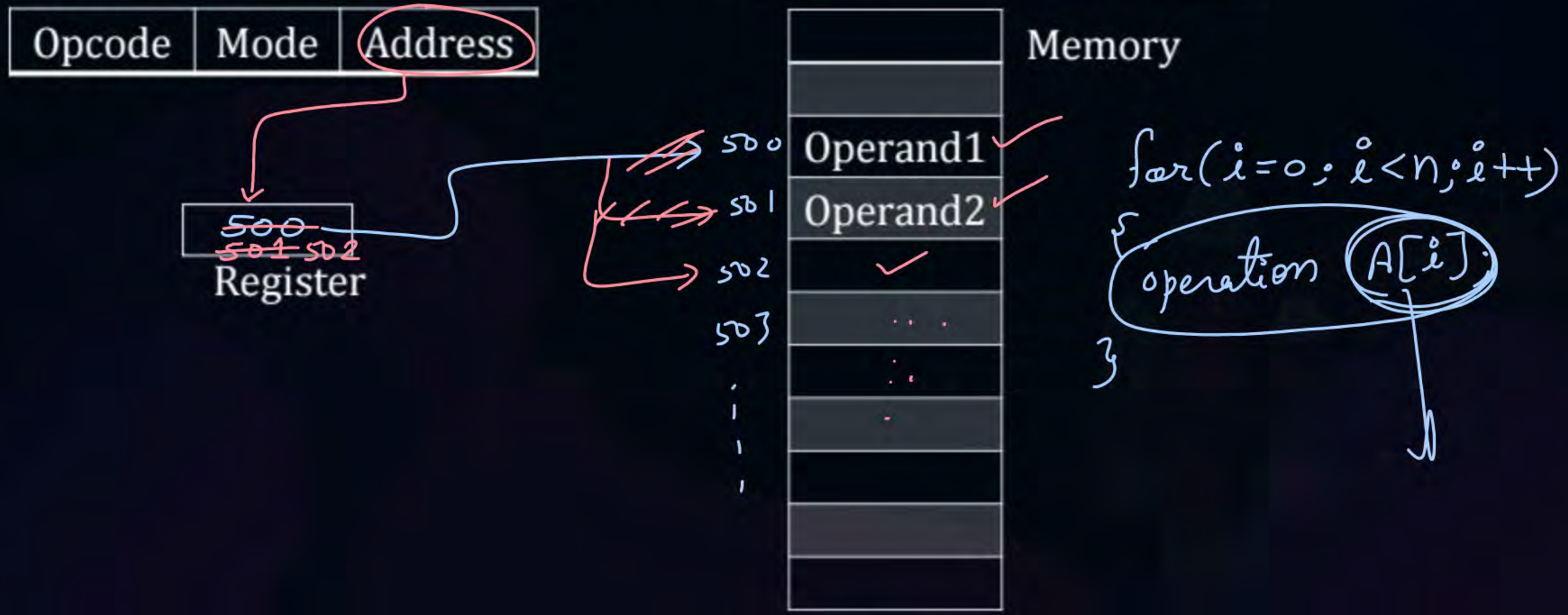
Topic : Register Indirect Mode

The address field of instruction specifies a register which holds effective address



Topic : Autoincrement/Autodecrement Mode

Variant of register indirect mode, in which content of register is automatically incremented or decremented to access a table of content sequentially.



Autoincrement ⇒ Post increment
Autodecrement ⇒ Pre decrement

} old architecture conventions

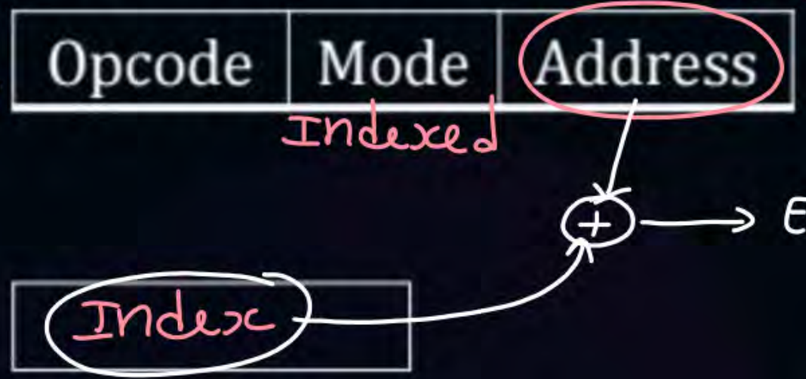
Amount of increment, decrement depends on size of data item accessed.

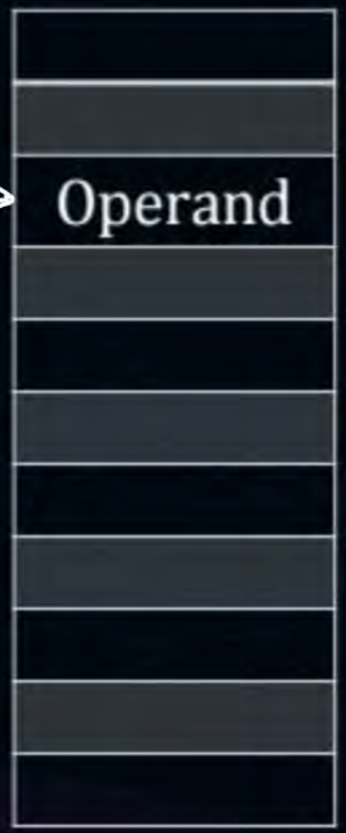
Topic : Indexed Mode

→ Used for accessing an element of array.

Index – or – Reg. mode

Address part of instruction (base address) is added to index register value to get the effective address

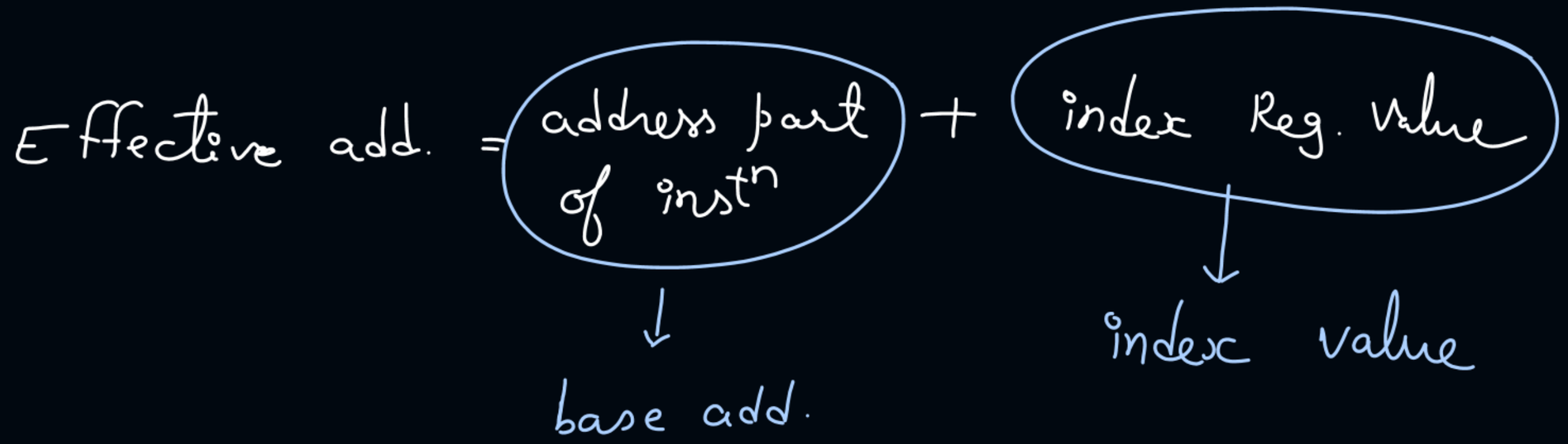




Memory

No. of memory references to get operand = 1

1. index Reg. ← i * size of element
2. inst^n which used index mode


Effective add. =
address part of instⁿ
+
index Reg. value
base add.
index value

address of A[i]

= Base + (Size of element * i)

index value

add. of A[3] = 200 + 1 * 3

= 203

base add.

index value

Implementation of Indexed mode

Special purpose Reg. for Index Reg.

opcode	mode	add.

Index Reg.

no any special purpose index Reg.

↓

one of general purpose Reg. used as index Reg.

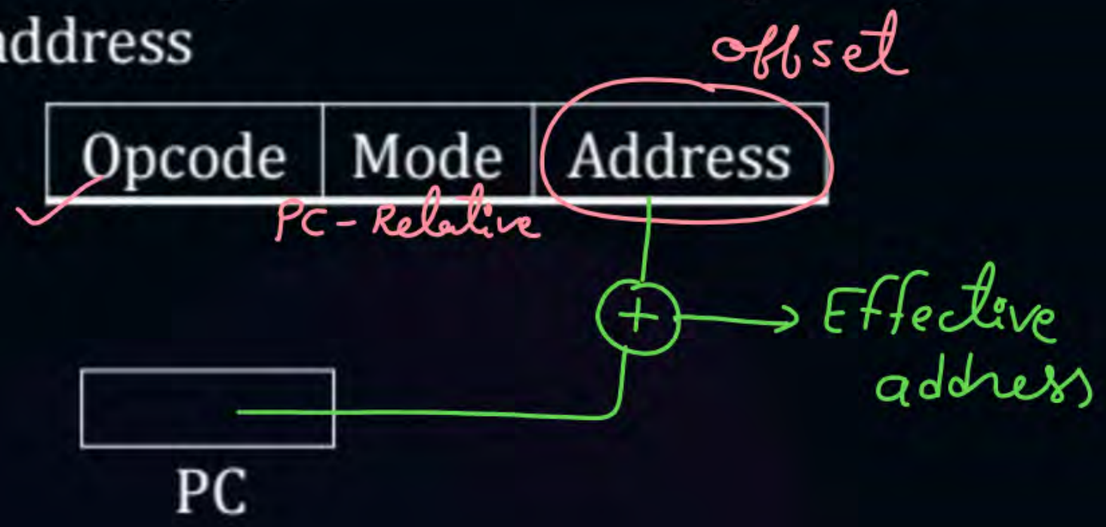
opcode	mode	Index Reg. Identifier	Address

Topic : PC-Relative Mode

Position Independent Mode :-

Used for branching (intra-segment)

Address part of instruction (offset) is added to PC register value to get the effective address



Effective add. = PC-value + address part of inst^n

offset or relative locations

⇒ called as position independent mode because the inst^n remains same even program location is whatever.

CPU is executing I2.

PC = 204

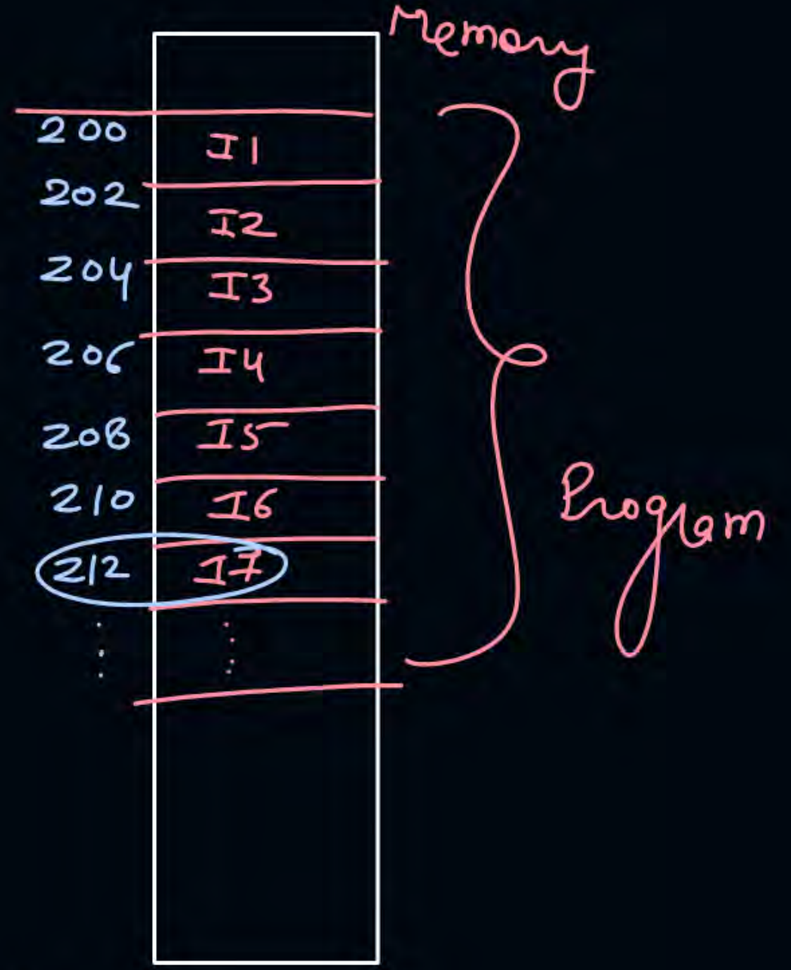
Assume CPU decodes I2 as branch instⁿ and I2 gives infoⁿ ⇒ Target is 8 location away for I7 as target instⁿ

Target add. = PC + 8

(Effective add.) = 204 + 8

= 212

8 → relative add. or offset or relative location to skip

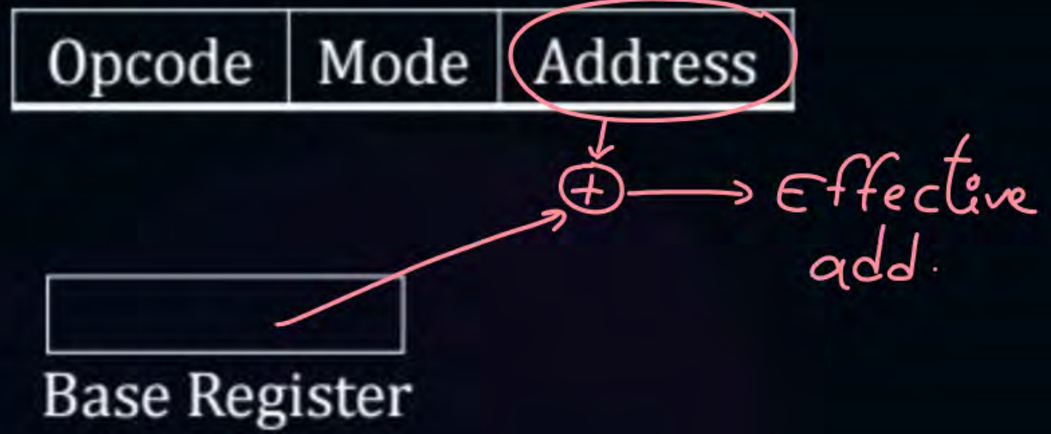


	offset
for forward jumping	+ve
for backward jumping	−ve

Topic : Base Register Mode → used for branching (inter-segment)

Address part of instruction (offset) is added to Base register value to get the effective address

offset



Effective add. = Base Register + (Address part of inst^n)

↓

offset

1. Implied mode
2. Immediate mode
3. Direct
4. Indirect
5. Reg. mode
6. Reg. Indirect
7. Auto inc./Autodec.
8. Indexed mode

→ used for computation type instns

⇓

provide either operand or effective add. of operand

9. PC – Relative
10. Base Reg. mode

used for branch type instns ⇒ Provide target address for branch instn

Modes

Non-Computable

No any computation needed for eff. add.

⇓

- ⟶ Implied
- ⟶ Imm. mode
- ⟶ Direct
- ⟷ Indirect
- ⟶ Reg. mode
- ⟶ Reg. Indirect

Computable

computation needed to calculate eff. add.

⇓

- ⟶ Autoinc. / Autodec.
- ⟶ Indexed mode
- ⟶ PC - Relative
- ⟶ Base Reg. mode

→ PC relation and base Reg. mode supports runtime relocation of program.

Topic : Example

Inst^n | opcode | mode | add. = 500

add. of memory

	Memory	
200	*Opcode*	Mode
201	Address = 500	
202	Next Instruction	
399	450	
400	700	
500	800	
600	900	
702	----------	
800	300	

202

PC = ~~200~~

Register R500

R500 = ~~400~~ 399

Index Reg.

XR = 100

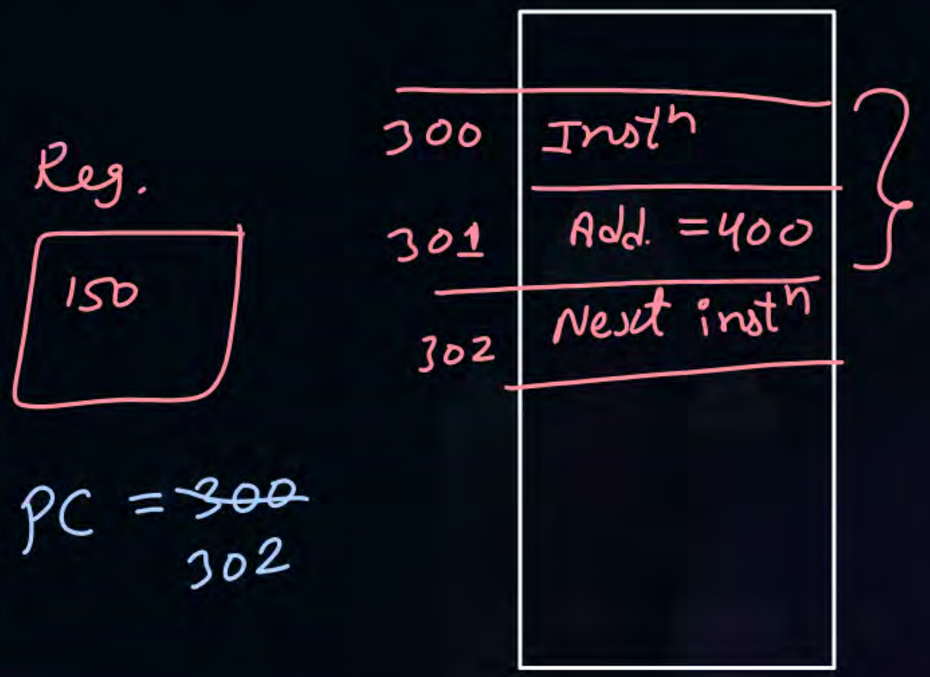
AC

Mode	Effective Address	Operand
1. Immediate Mode	201	500
2. Direct Mode	500	800
3. Indirect Mode	800	300
4. Register Mode	—	400
5. Register Indirect Mode	400	700
6. Autodecrement Mode (Bre)	399	450
7. Indexed Mode	500 + 100 = 600	900
8. PC- Relative Mode	202 + 500 = 702	—

[MCQ]

#Q. An instruction is stored at Location 300 with its address field at location 301. The address field has the value 400. A processor register contains the number 150. Evaluate the effective address, if addressing mode is:

1. Direct ⇒ 400
2. Immediate ⇒ 301
3. Relative ⇒ 302 + 400 = 702
4. Register Indirect ⇒ 150



[MCQ]

#Q. In case the code is position independent, the most suitable addressing mode is

A Direct mode

B Indirect mode

C Relative mode

D Indexed mode

[MSQ]

#Q. The addressing mode that permits relocation, without any change whatsoever in the code, is

A Indirect addressing

B Base register addressing

C Indexed addressing

D PC relative addressing

~~[MCQ]~~ [NATs]

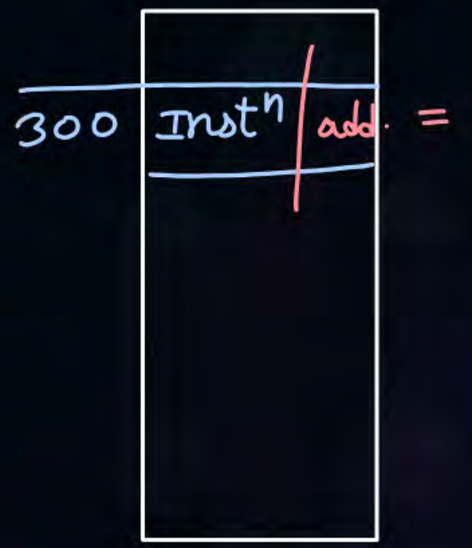
#Q. A relative branch mode type instruction is stored in memory at address 300. The branch is made to an address 450.

1. What should be the value of relative address field of the instruction? ⇒ 149
2. Determine the value of PC before instruction fetch, after the fetch and after execution phase?

Target (eff.) add. = 450 = PC + offset

450 = 301 + offset

offset = 149



	PC
Before fetch	300
After fetch	301
After execution	450

Ques) Consider a prog. which has 200 instns I1 to I200. Branch instn I4 has target instn I37. Each instn takes 4 locations in memory. The branch instn I4 uses pc-relative mode. The offset ____ ?

Ans :-

offset = 128

no. of instns skipped = 36 - 5 + 1
= 32

no. of locations to skipped = 32 * 4
= 128

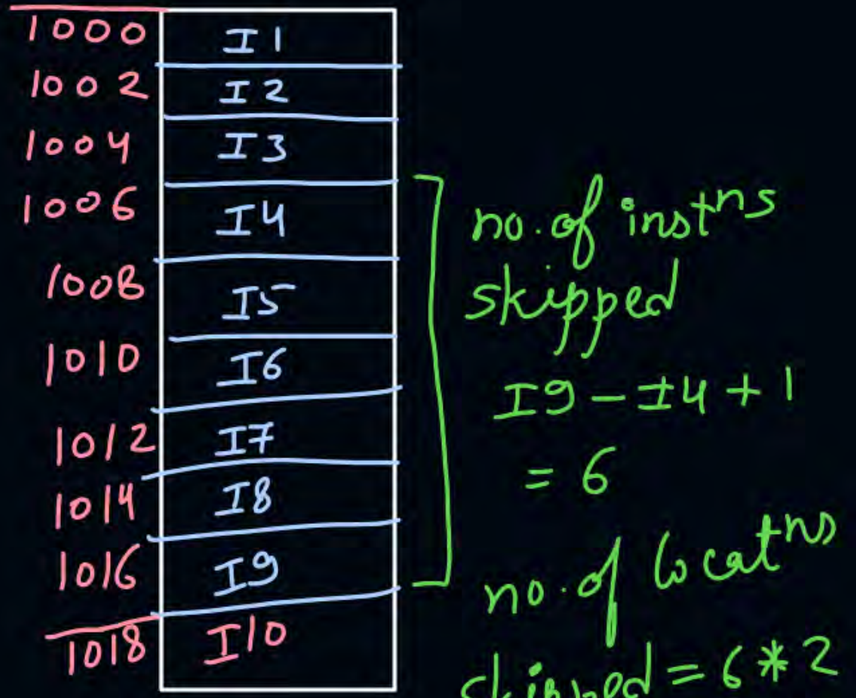
Ques) PC relative mode instⁿ I9 jumps to I4.
Each instⁿ size => 2 locations in memory.

offset = ——?

Ans.

1006 = 1018 + offset

offset = −12



[MCQ]

#Q. Consider a hypothetical processor with an instruction of type LW R1, 20(R2); which during execution reads a 32-bit word from memory and stores it in a 32-bit register R1. The effective address of the memory location is obtained by the addition of a constant 20 and the contents of register R2. Which of the following best reflects the addressing mode implemented by this instruction for operand in memory?

A Immediate Addressing

B Register Addressing

C Register indirect scaled addressing

D Base indexed addressing

[NAT]

#Q. Consider a three-word machine instruction

ADD A[R0], @B

The first operand (destination) "A[R0]" uses indexed addressing mode with R0 as the index register. The second operand (Source) "@B" uses indirect addressing mode. A and B are memory addresses residing at the second and the third words, respectively. The first word of the instruction specifies the opcode, the index register designation and the source and destination addressing modes. During execution of ADD instruction, the two operands are added and stored in the destination (first operand).

The number of memory cycles needed during the execution cycle of the instruction is_____.

[NAT]

#Q. Consider a 6-words instruction, which is of the following type:

Opcode	Mode1	Mode2	Address1	Address2

The first operand (destination) uses register indirect mode and second operand uses indirect mode. Assume each operand is of size 2 words, each address is of 2 words and main memory takes 50ns for 1 byte access. Further assume that the opcode denotes addition operation which copies result of addition of 2 operands. One word size is 4 bytes. Total time required in:

1. Fetch cycle of instruction
2. Execution cycle of instruction
3. Instruction cycle of instruction

2 mins Summary

Topic Addressing Modes

Topic PC Relative Mode

Topic Questions on Addressing Modes

Happy Learning

THANK - YOU