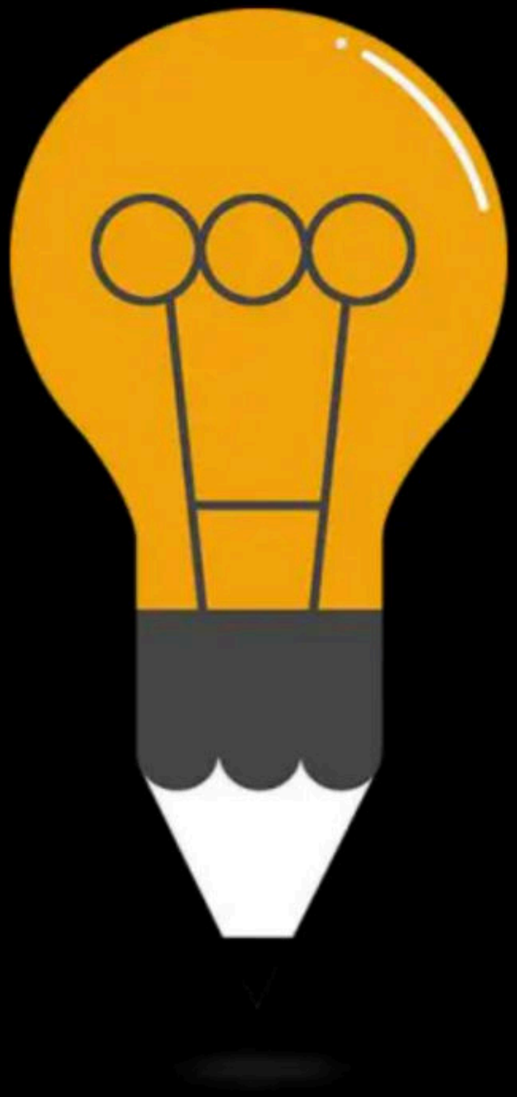
Addressing Modes: Part II

Complete Course on Computer Organization & Architecture for GATE 2024 & 2025

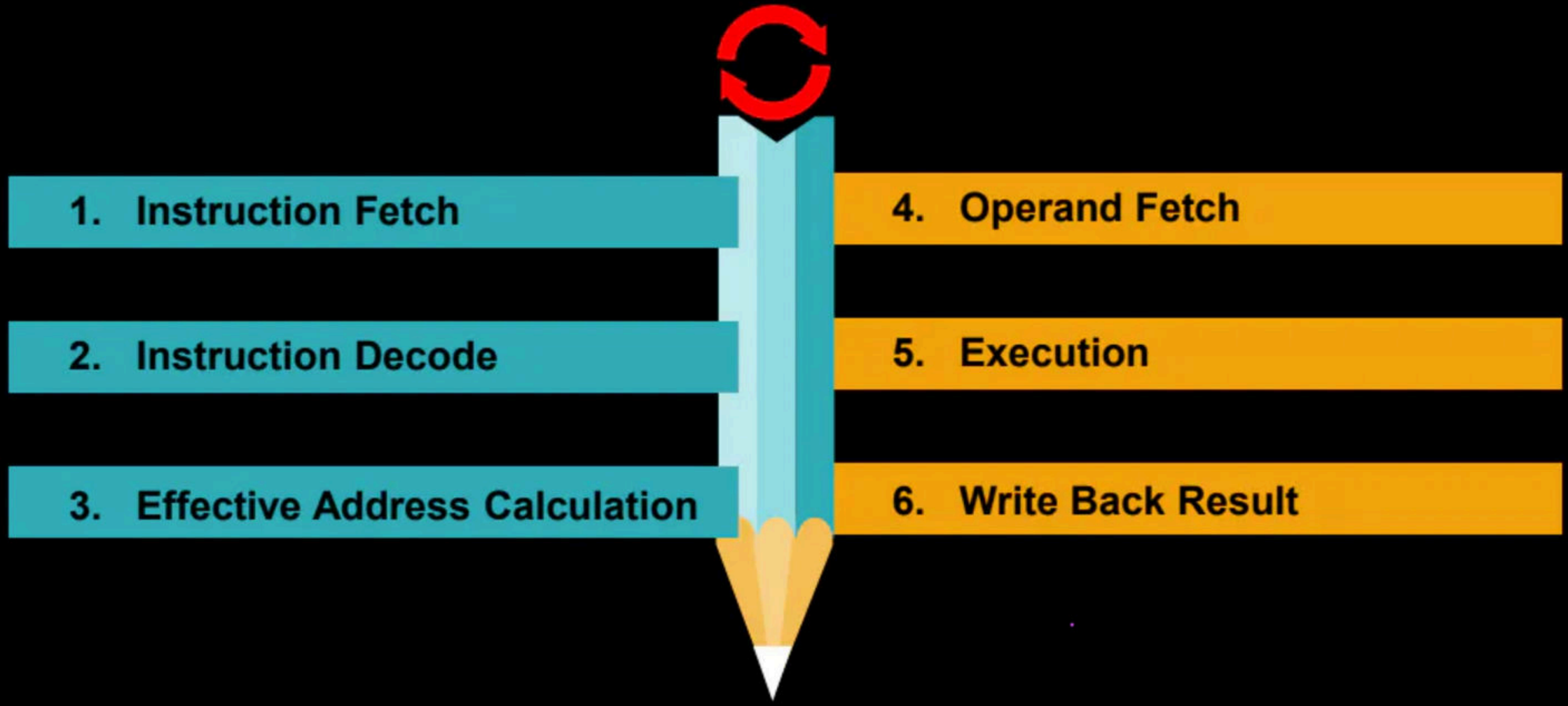
Vishvadeep Gothi • Lesson 13 • Feb 28, 2023



Addressing Modes

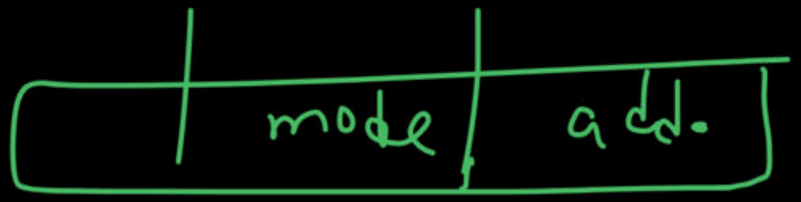
By: Vishvadeep Gothi

Instruction Cycle



Addressing Modes

It specifies how and from where the operands are obtained for an instruction



Implied Mode

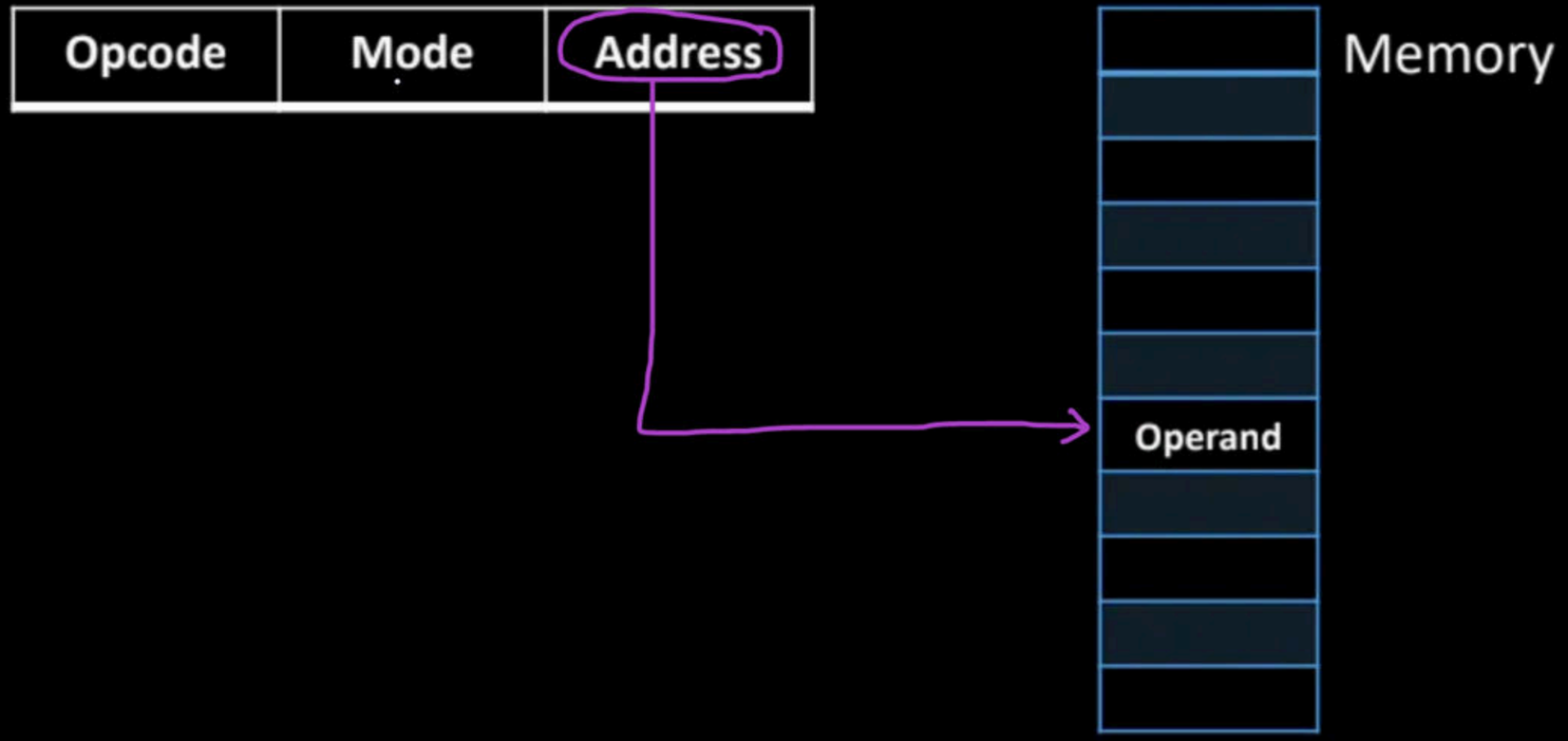
The opcode definition itself defines the operand

Immediate Mode

- The address field of instruction specifies the operand value

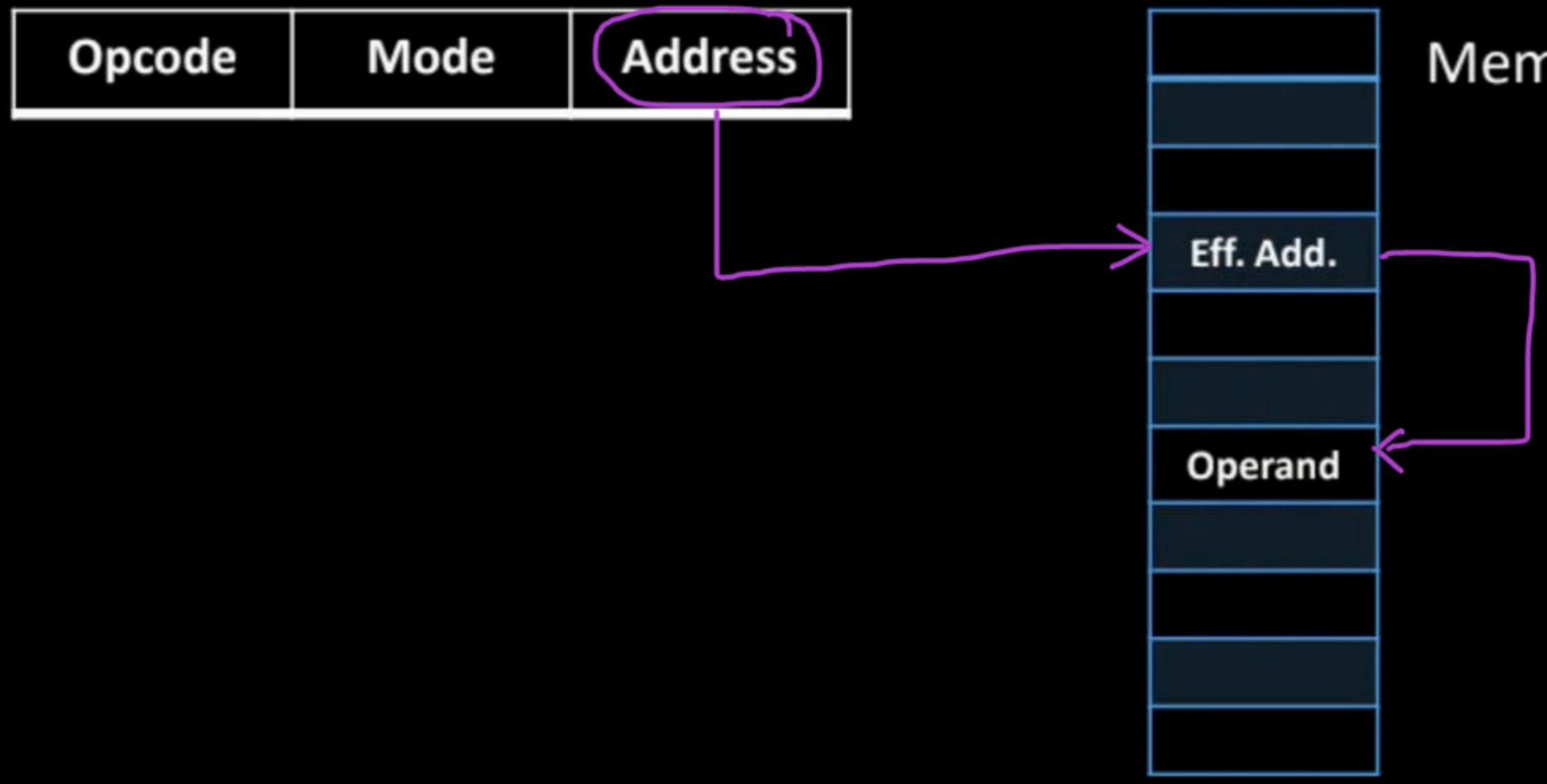
Direct Mode

- The address field of instruction specifies the effective address



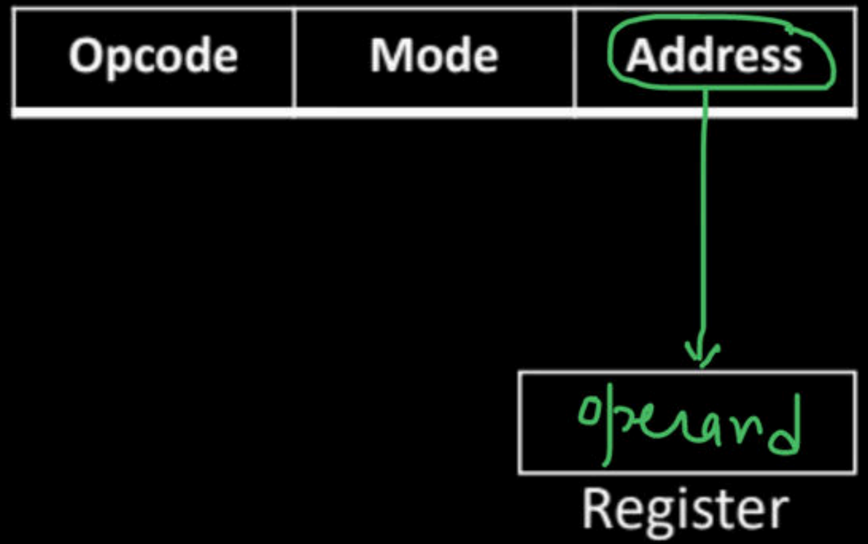
Indirect Mode

- The address field of instruction specifies the address of effective address



Register Mode

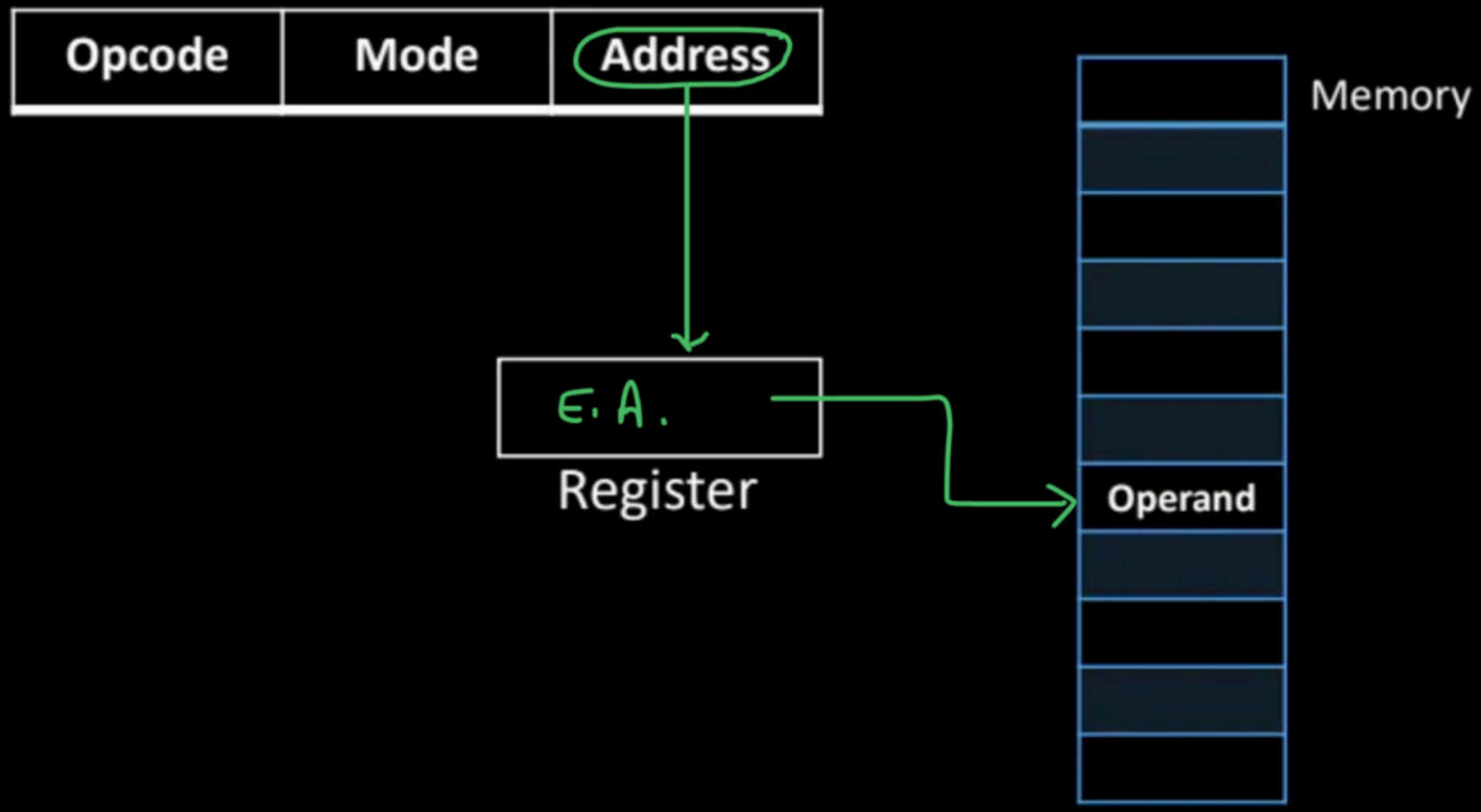
- The address field of instruction specifies a register which holds operand



Register Indirect Mode

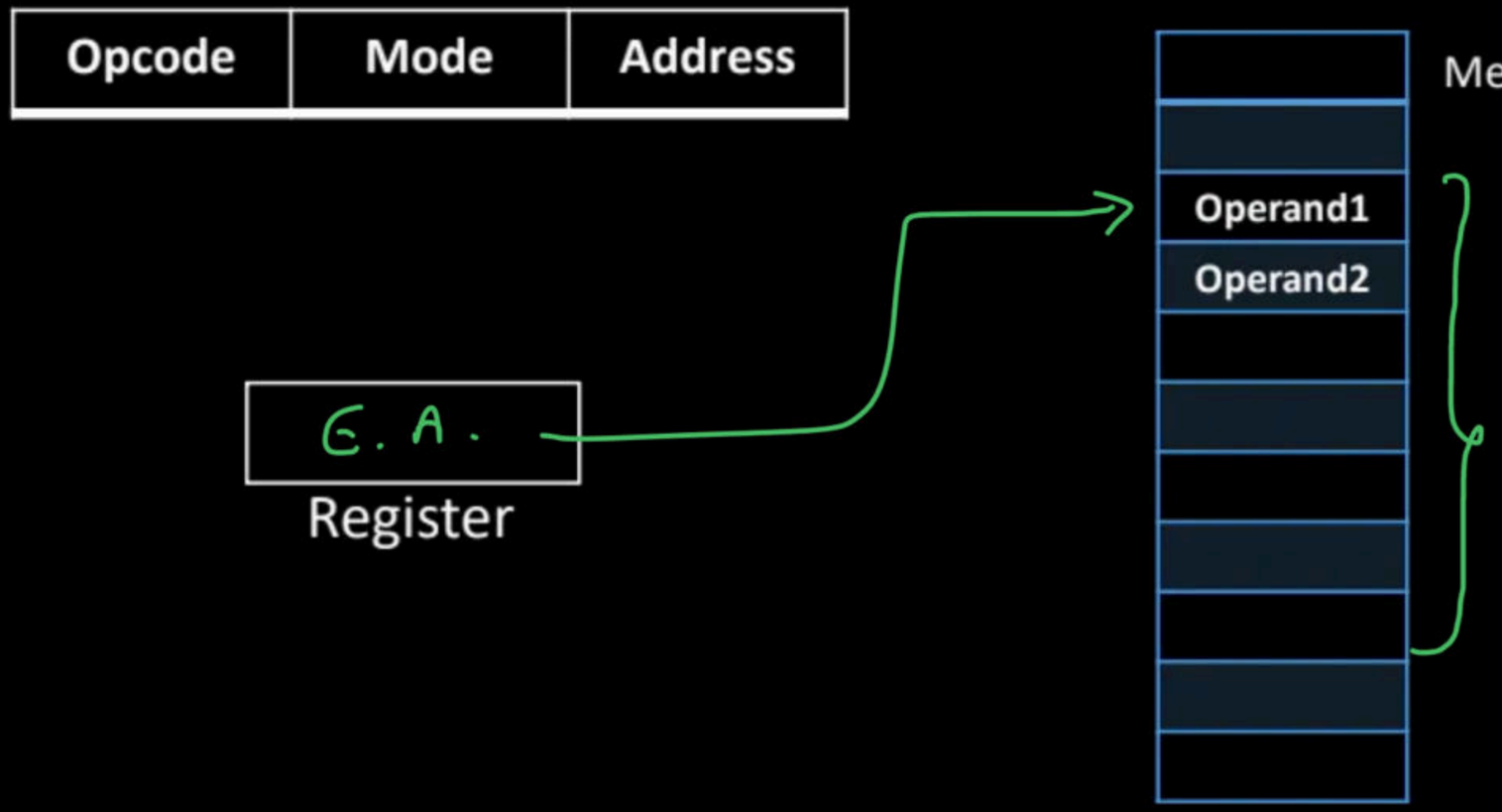
- The address field of instruction specifies a register which holds operand

E. A.



Autoincrement/Autodecrement Mode

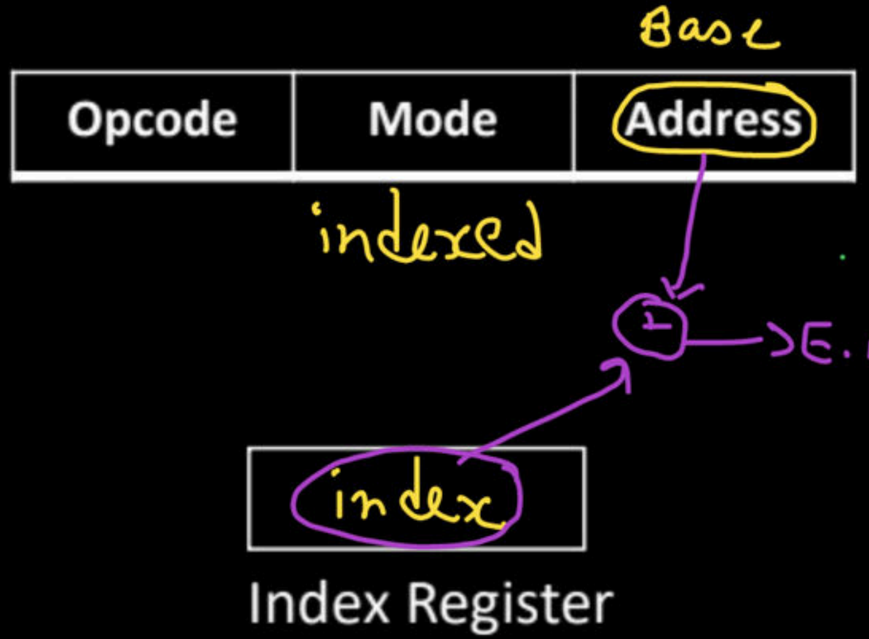
- Variant of register indirect mode, in which content of register is automatically incremented or decremented to access a table of content sequentially.

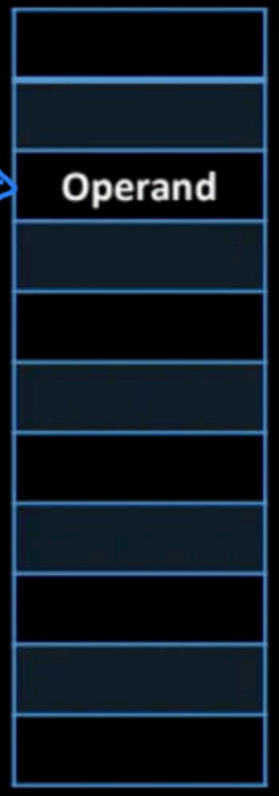


Indexed Mode / Index Reg. Mode

↳ used to access an array element

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





Memory

To obtain operand

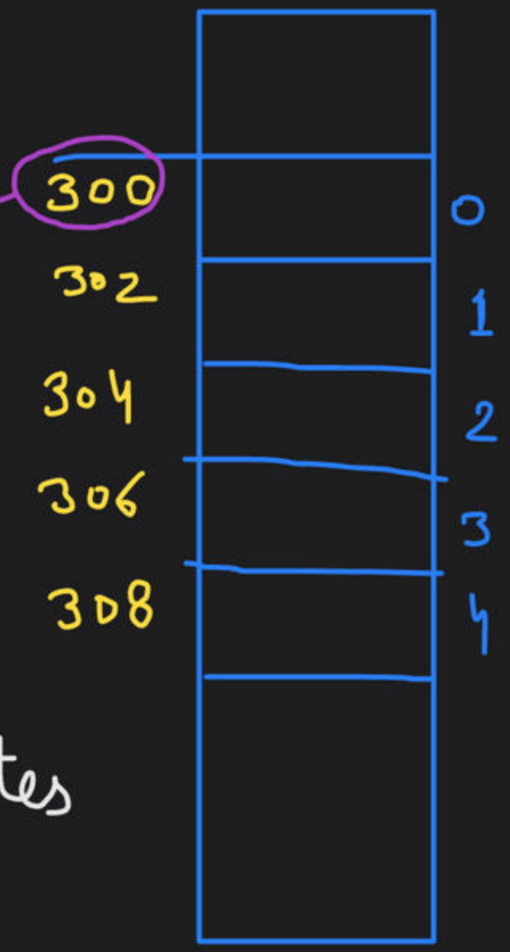
↓

→ 1 addit^{n} operat^{n} (for E.A. calculat^{n})

→ 1 mem. access

E.A. = Add. part of inst^{n} + index Reg. value

int A[5]



base add. of array

each int = 2 bytes size

Access A[3]

E.A. of A[3] ⇒ 306

add. of A[i] = base + size * i

= 300 + 2 * 3

= 306

---

i^{nst^h} index ⇐ 2 * i

Reg.

$operat^n$ on operand stored at add.

E.A. = base + index value

base + index value

If array data relocated then base add. of array should be updated in address part of instruction.

Updation of inst^{n} is a costly operation.

Indexed mode does not support relocation

Index Reg. Mode

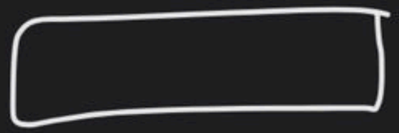
Special purpose

opcode	mode	add.

Any GPR used as index Reg.

opcode	mode	Index reg.	add.

designatn

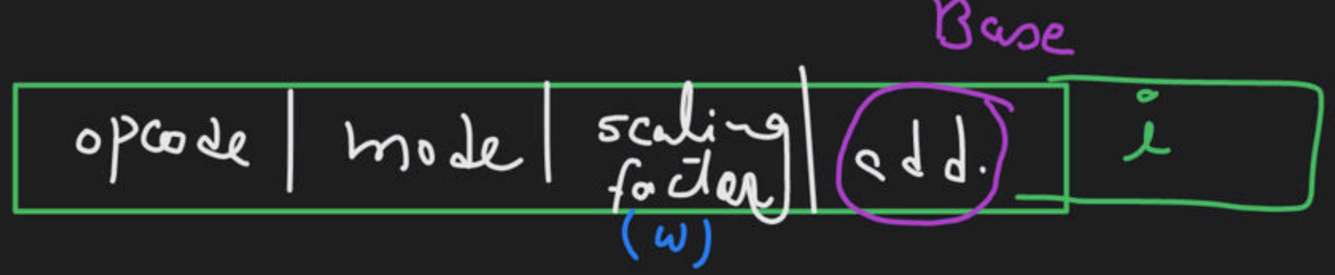


Index Reg.

R7 ← 2 * i

Scaled addressing mode

Advance version of indexed mode.



E.A. = add + i * w

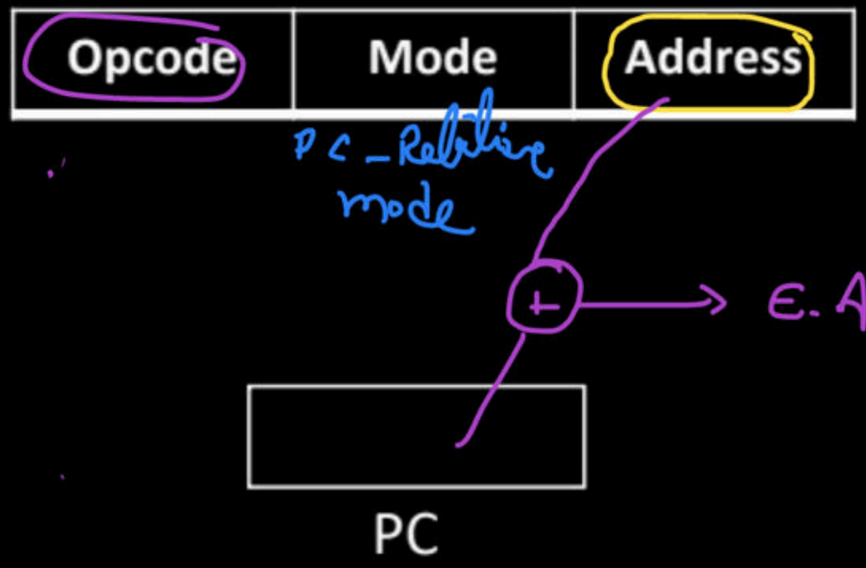
used for branch type of instn (intra-segment)

PC-Relative Mode

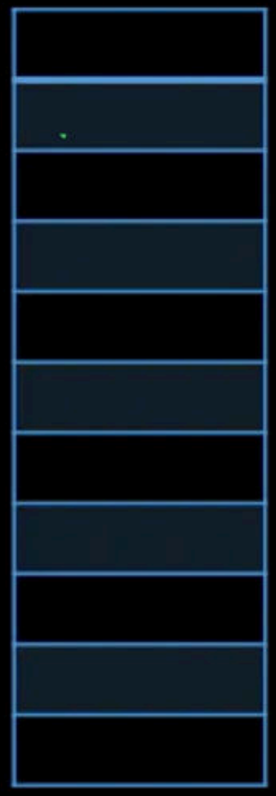
Position independent mode

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

relative locn's or offset or relative add. field



to skip



	offset
forw. jumping	+ve
back. jumping	−ve

CPU fetched I2,

PC = 504

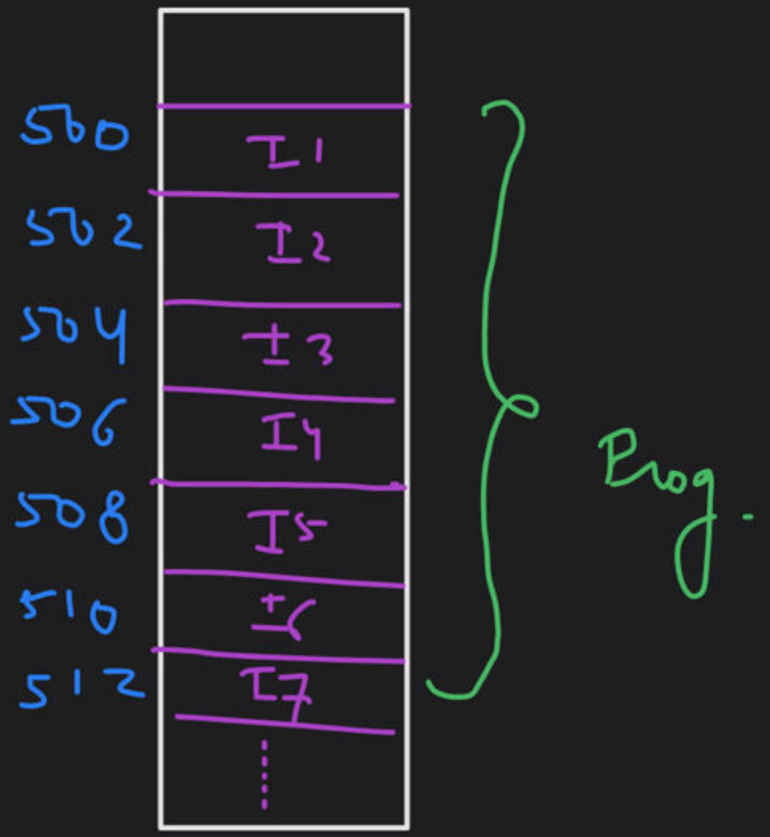
CPU decoded I2 as branch type instn

Target instn = I7

Target add. = PC + 8

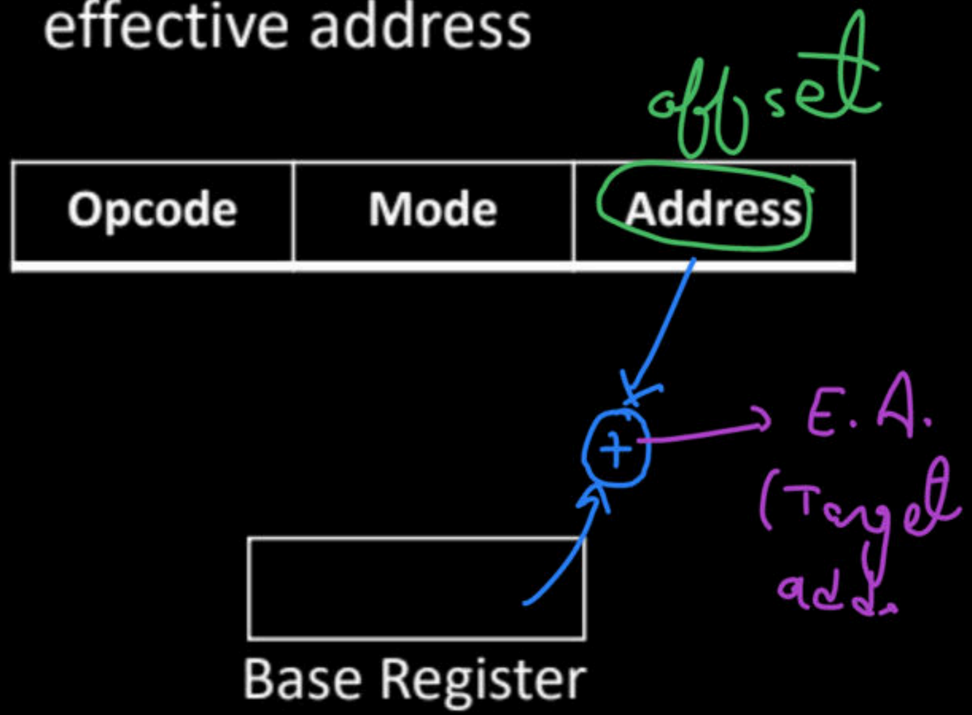
= 504 + 8

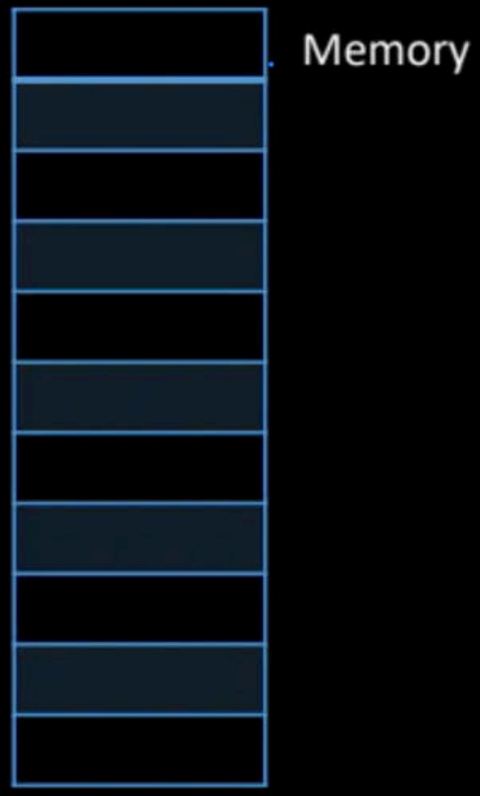
= 512



Base Register Mode → inter-segment branching

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





→ Implied
→ Immediate
→ Direct
→ Indirect
→ Reg. mode
→ Reg. Indirect
→ Indexed
→ Autoinc/Autodec.
→ Scaled

} data/operand related addressing mode

→ PC relative
→ Base Reg. mode

} branching

category

computable add. mode
for E.A. calculatn some
computation needed

→ Indexed
→ Auto inc. / Auto dec.
→ Scaled
→ PC - Relative
→ Based - Reg. mode

non-computable add. modes

→ Implied
→ Imm. mode
→ Direct
→ Indirect
→ Reg.
→ Reg. Indirect

Example

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

Regs

PC = ~~200~~ 202

R500 = ~~400~~ 399

XR = 100

Index Reg.

AC

mem. add.

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	399	450
7. Indexed Mode	500 + 100 = 600	900
8. PC- Relative Mode	202 + 500 = 702	–

Question Morris Mano

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
2. Immediate
3. Relative
4. Register Indirect

Question

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

Question

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

Question Morris Mano

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?
2. Determine the value of PC before instruction fetch, after the fetch and after execution phase?

Question GATE-2011

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

Question GATE-2005

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

Happy Learning.!