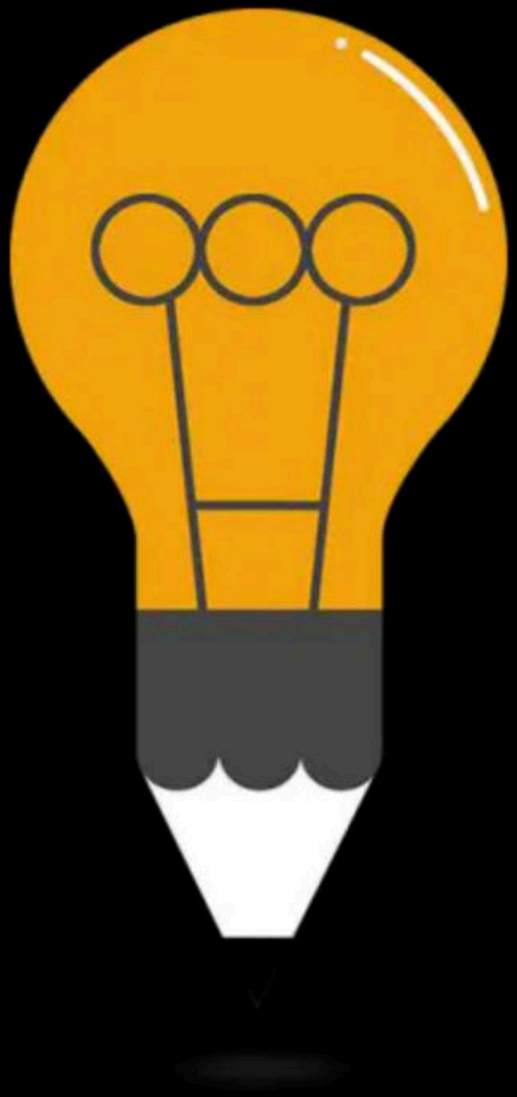
Addressing Modes: Part I

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

Vishvadeep Gothi • Lesson 12 • Feb 27, 2023

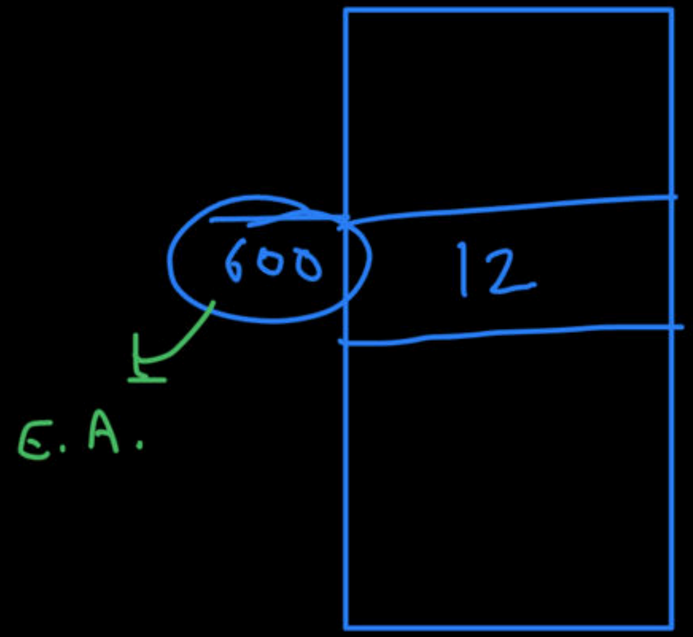


Instruction Cycle & Addressing Modes

By: Vishvadeep Gothi

Effective Address

Address of operand in a computation-type instruction or
The target address in a branch-type instruction.

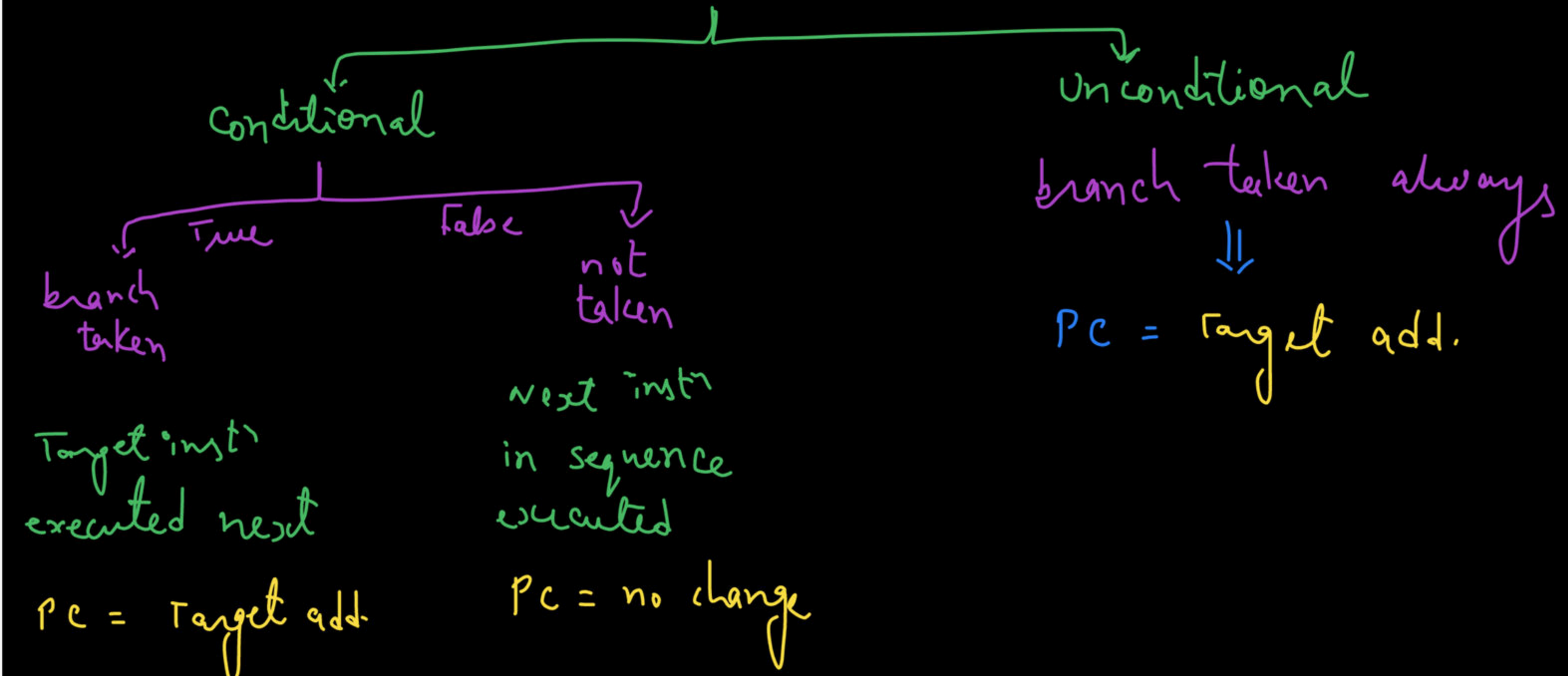


PC = target add.

no change in PC

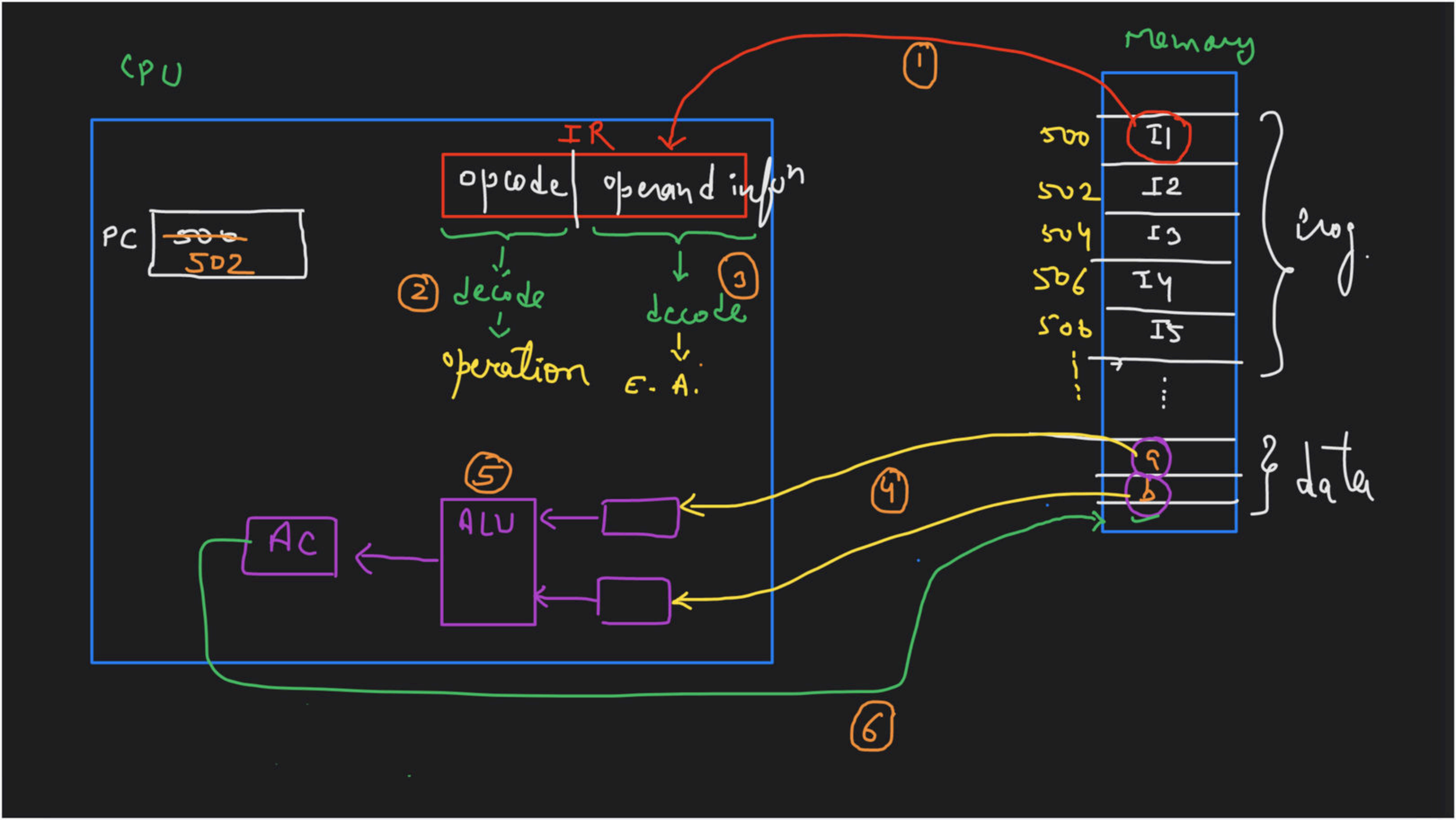
Branch Instruction

Type of Branch Instruction

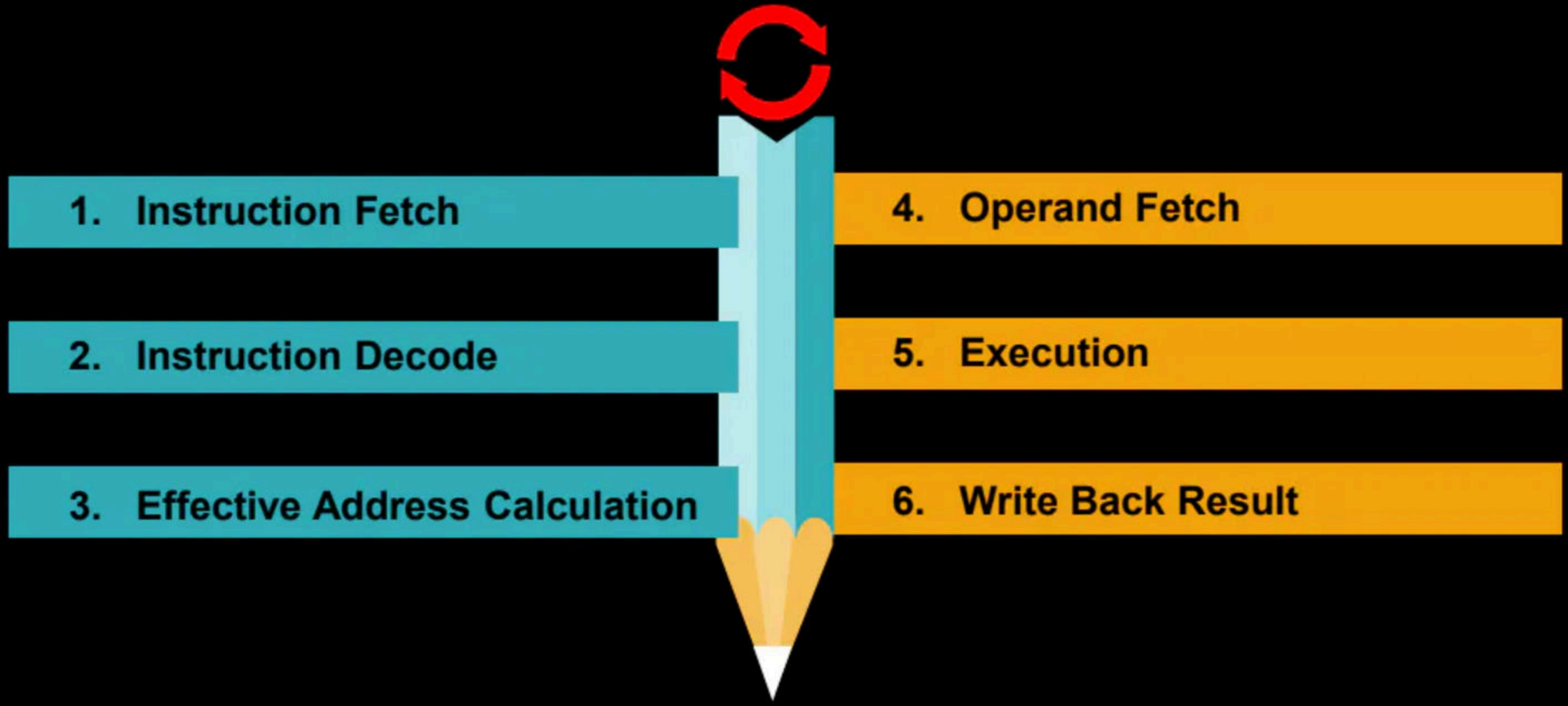


Instruction Cycle

Inst^ executed in 6 phases


CPU
Memory
IR
opcode
operand info
PC
500
502
1
2 decode
operation
3
decode
E. A.
5
ALU
AC
4
6
500
502
504
506
508
I1
I2
I3
I4
I5
prog.
9
6
data

Instruction Cycle



Fetch Cycle & Execution Cycle

⇓

Instⁿ fetch

⇓

decode
to
write back result

Computation vs Branch Type Instruction

	Computation	Branch
Instⁿ fetch	fetch of instⁿ & PC increment	fetch of instⁿ & PC inc.
decode	opcode decode ⇒ operatⁿ	opcode decode ⇒ operatⁿ
E.A. calculatⁿ	operand decode ⇒ E.A.	calculate Target add.
operand fetch	fetch operands till ALU	—
Executⁿ	computation performed	condⁿ check & PC update
write back	Result copied back to destinatⁿ	—

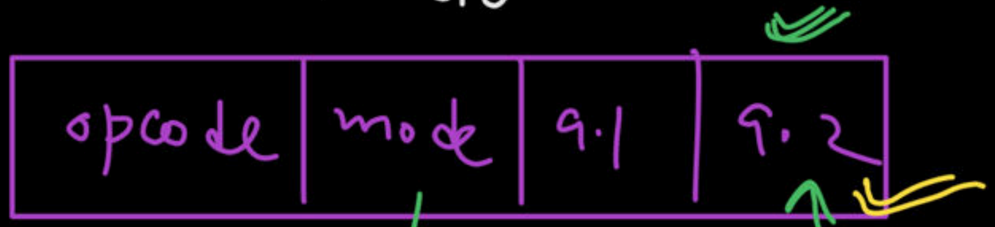
Why Addressing Modes

Reg. mem based arch.

'Addition

Reg.

Reg or mem



0 ⇒ Reg.

1 ⇒ mem.

0101 | 101 | 110

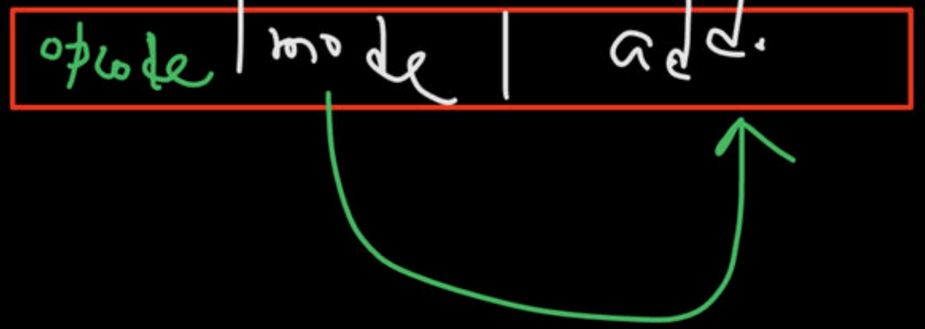
addition

Reg. R5

Reg. R6 or memory operand on add. 6

Addressing Modes

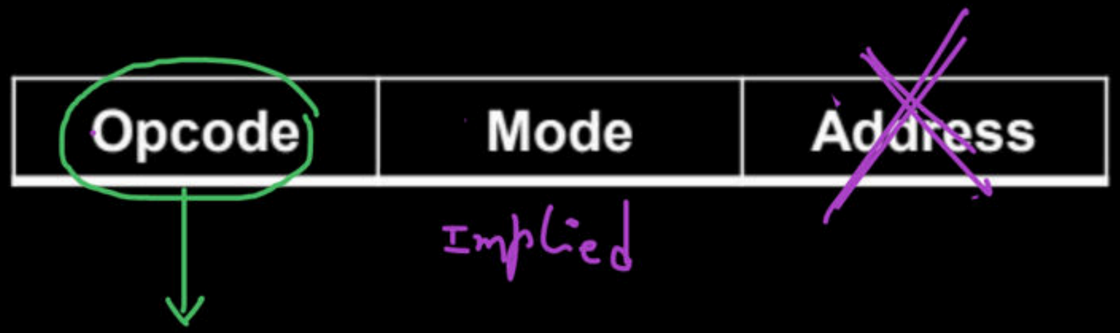
It specifies how and from where the operands are obtained for an instruction



→ CPU uses addr. modes in E.A. calculatn phase

Implied Mode

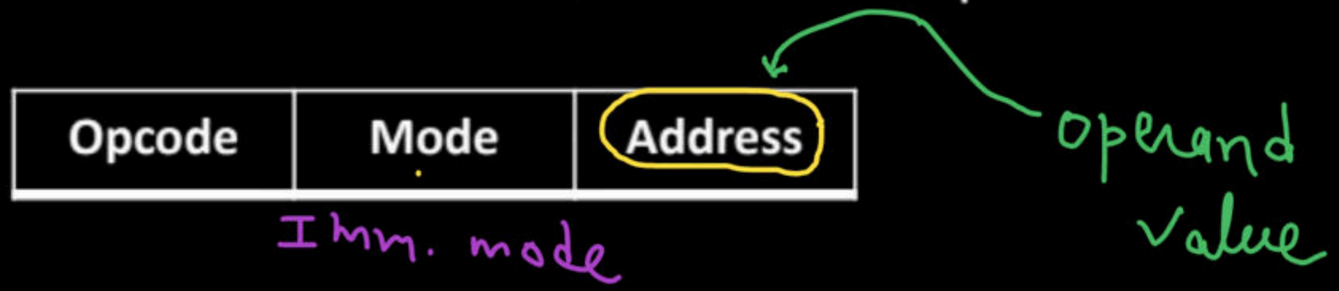
The opcode definition itself defines the operand



ex:- INC A

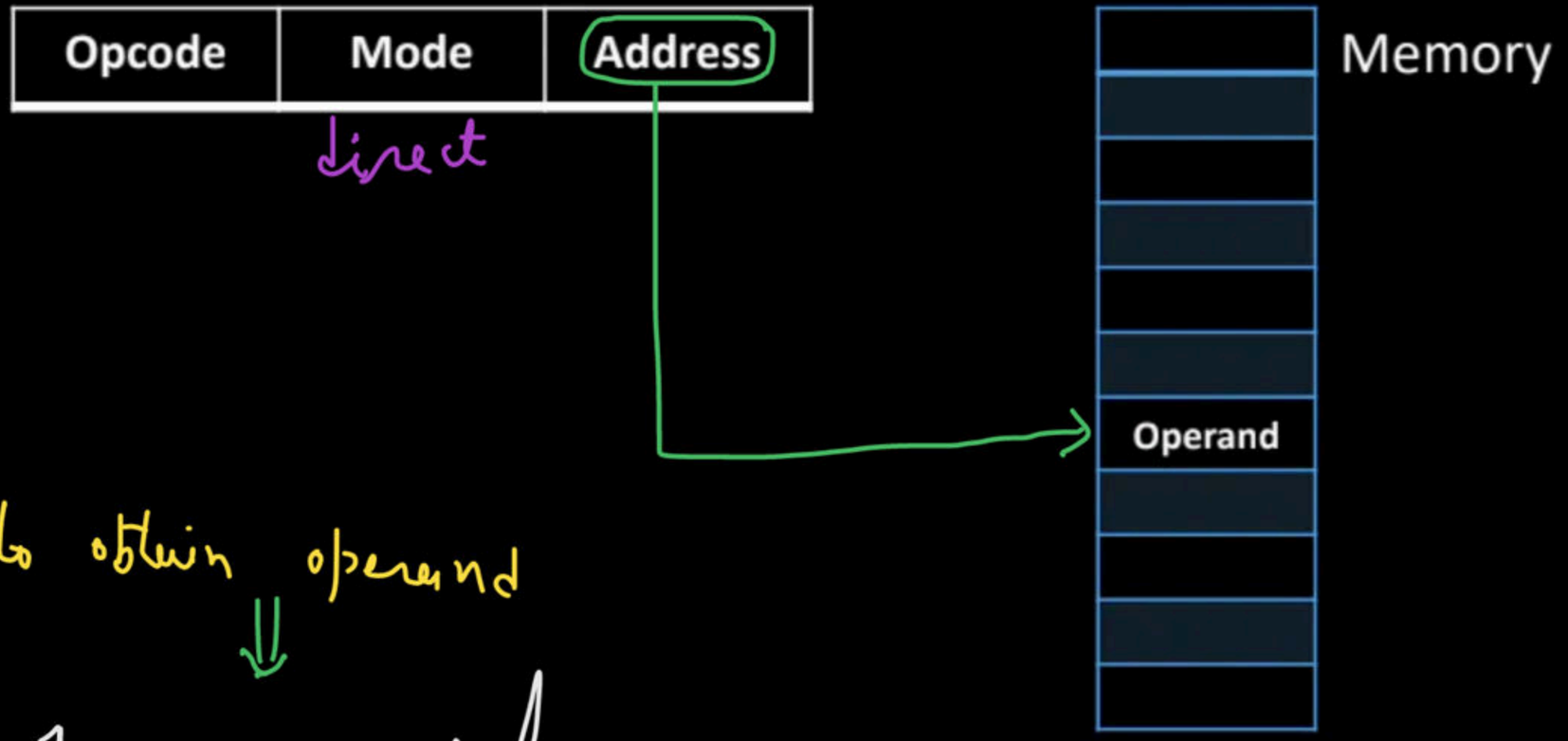
Immediate Mode

- The address field of instruction specifies the operand value



Direct Mode / Absolute mode

- The address field of instruction specifies the effective address



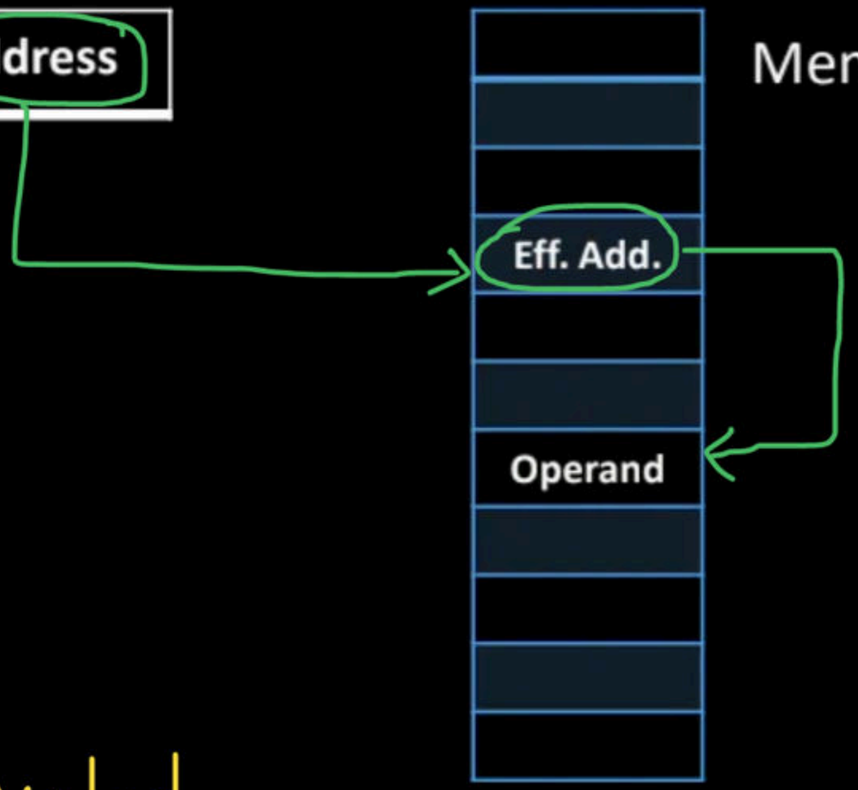
to obtain operand

⇓

1 memory reference

Indirect Mode

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



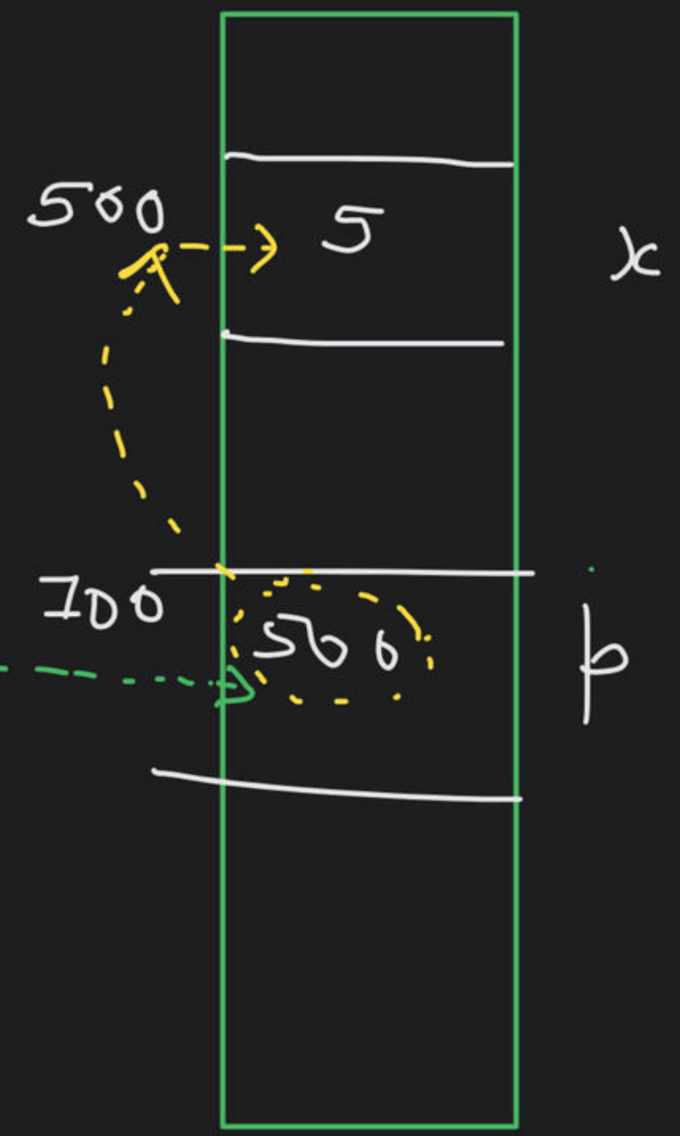
Used to implement pointer.

to obtain operand

⇓

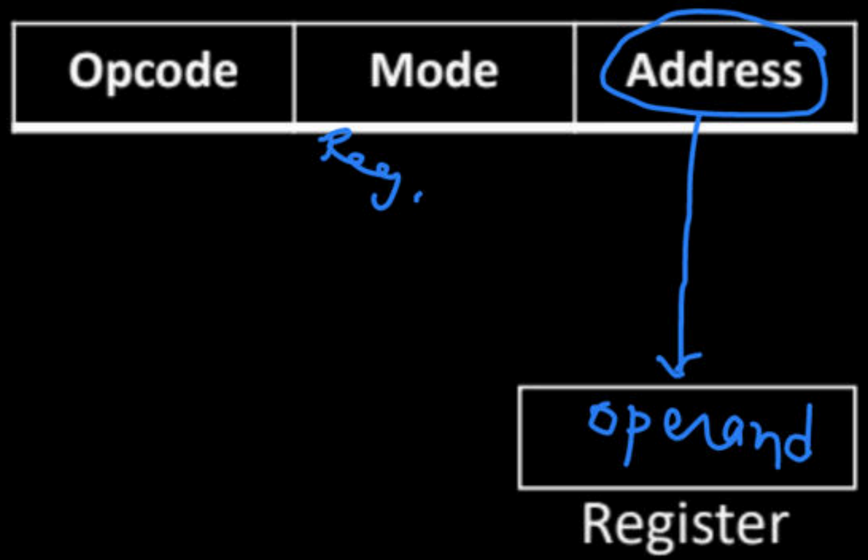
2 mem. references needed

```
int x = 5;
int *p;

p = &x;

*p
```



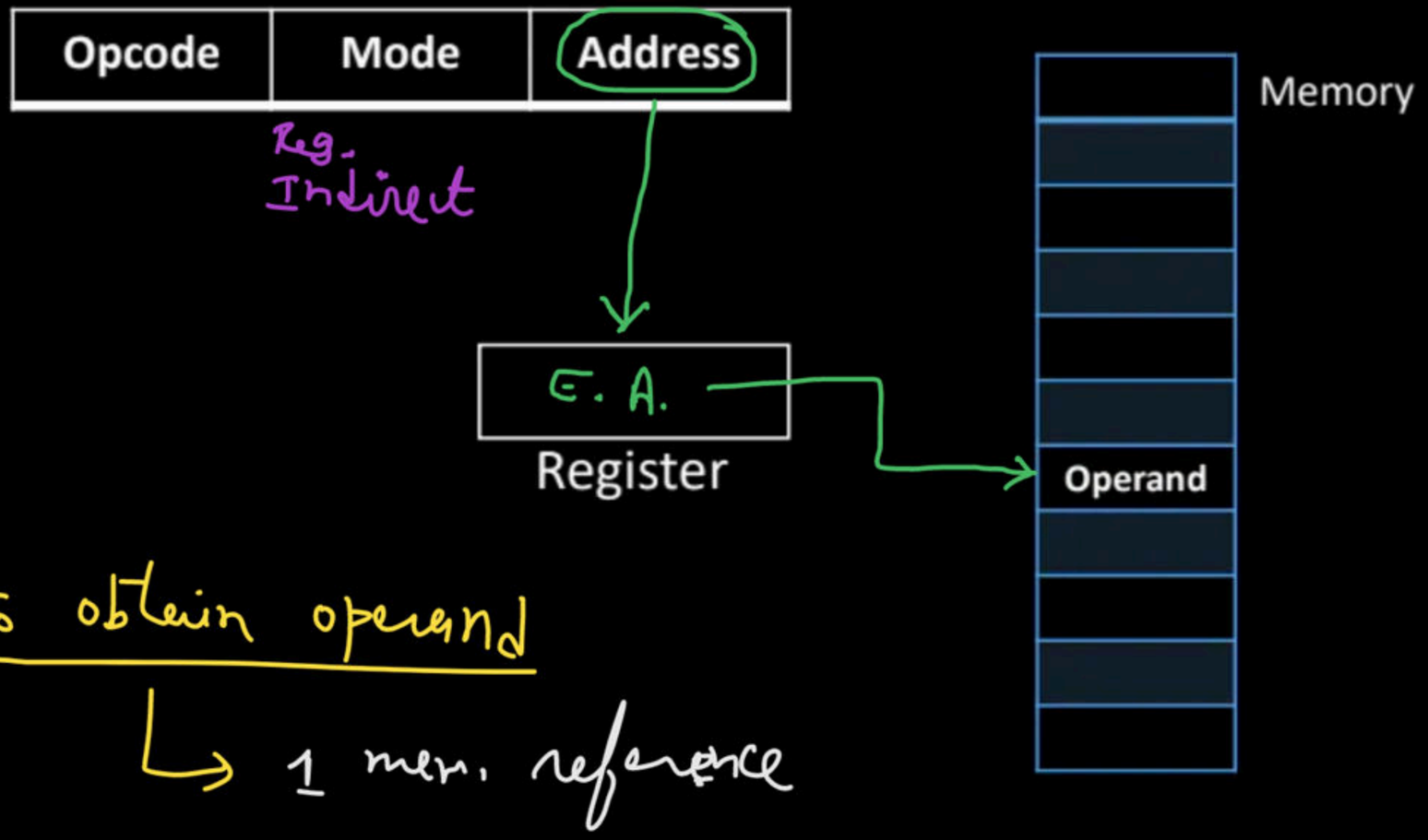
Register Mode

- 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



used to shorten instn length

To obtain operand

↳ 1 mem. reference

Assume CPU with 64 Reg. ⇒ Reg. reference = 6 bits

memory add. => 32 bits

Direct

(x + y + 32) bits

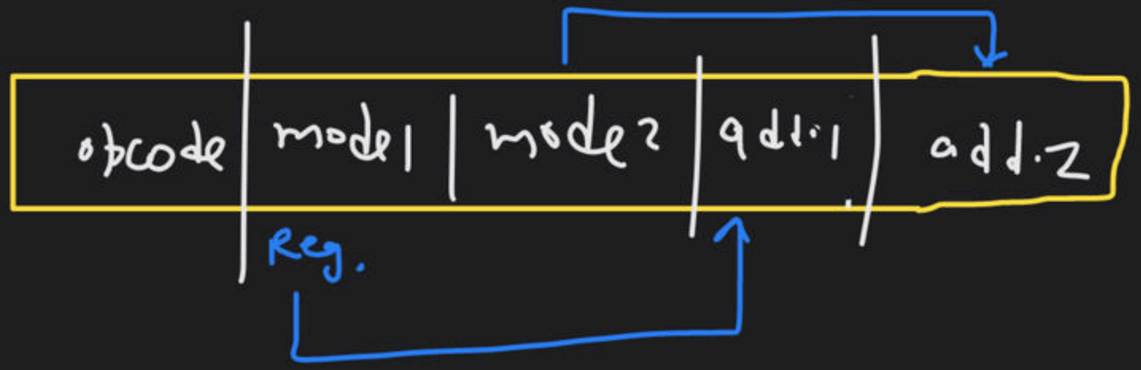
Time to get operand

= 1 mem access

Reg. Indirect

(x + y + 6) bits

= 1 Reg. access + 1 mem. access



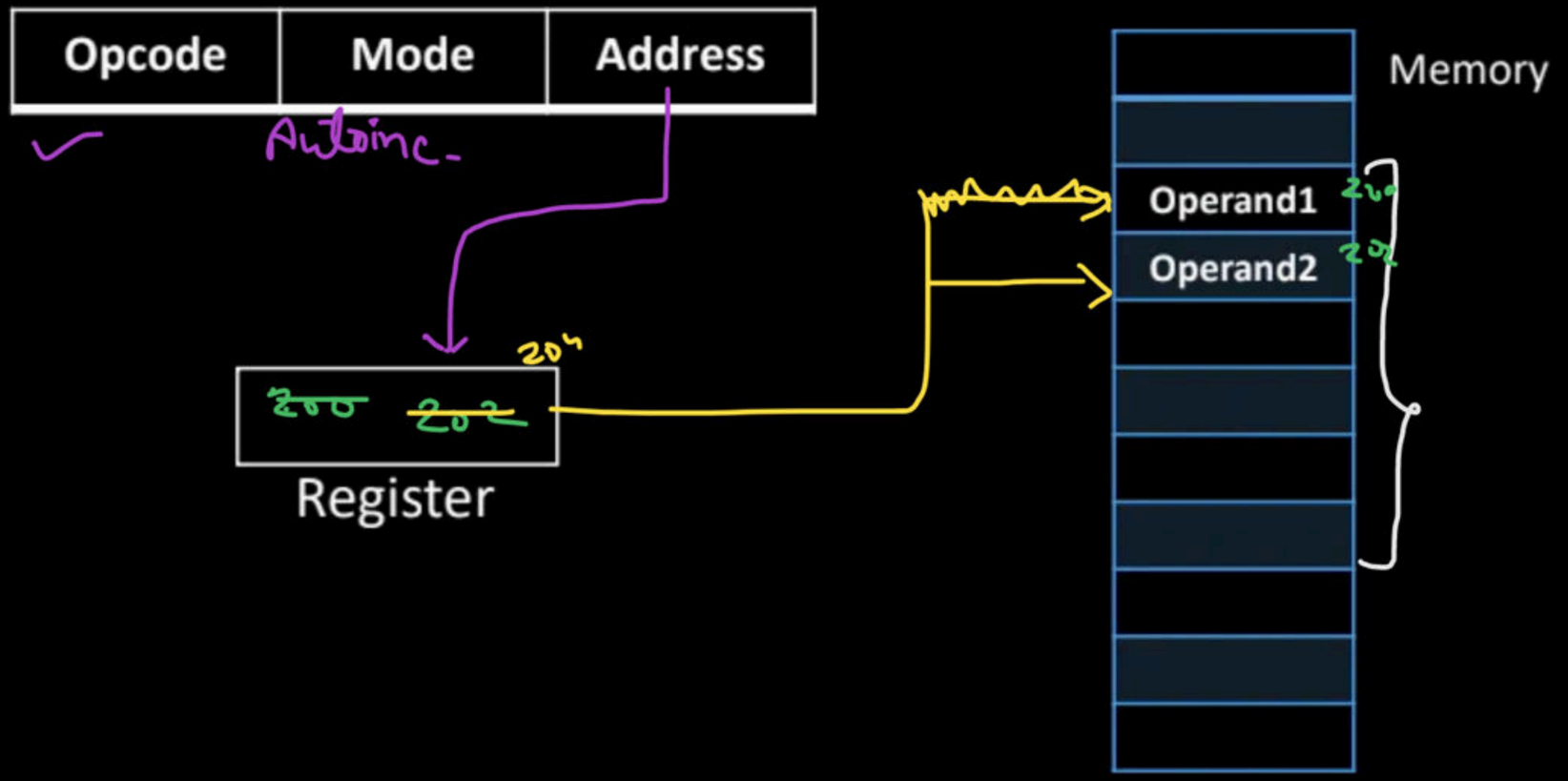
move Reg. 001 011

R1 <—

mode2	
Imm. mode	R1 ← #3
direct mode	R1 ← M[3]
Reg. mode	R1 ← R3
Reg. Indirect	R1 ← M[R3]
Indirect	R1 ← M[M[3]]

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.




```
for(i = 0; i < 50; i++)
{
  operat^n A[i];
}
```

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

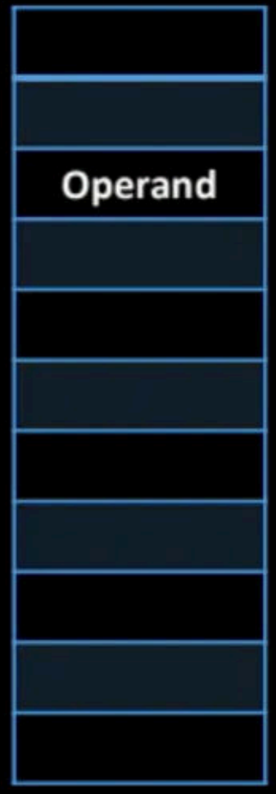
⇒ autoincrement mode ⇒ post increment

autodecrement mode ⇒ pre decrement

Indexed Mode

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

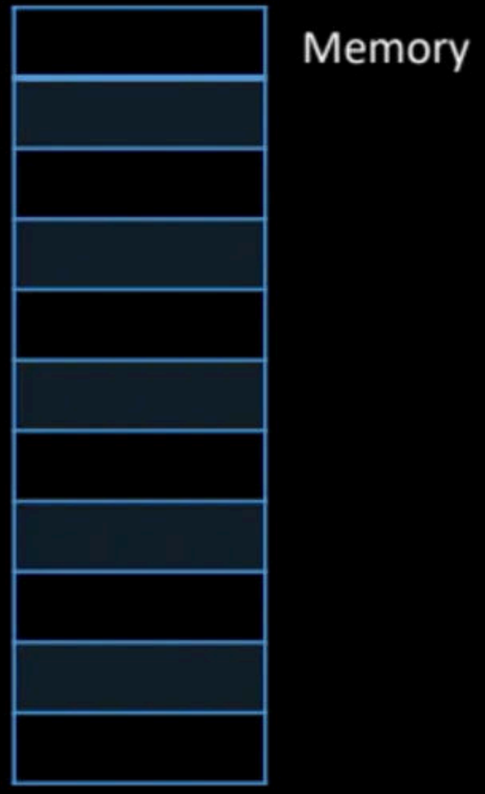
Index Register



Memory

PC-Relative Mode

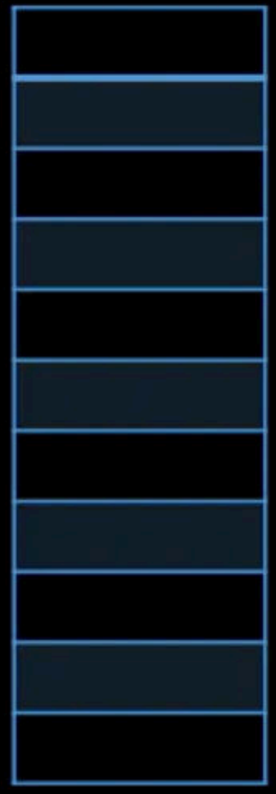
- Address part of instruction (offset) is added to PC register value to get the effective address



Base Register Mode

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

Base Register



Memory

Example

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

PC =200

R500 = 400

XR = 100

AC

Mode	Effective Address	Operand
1. Immediate Mode		
2. Direct Mode		
3. Indirect Mode		
4. Register Mode		
5. Register Indirect Mode		
6. Autodecrement Mode		
7. Indexed Mode		
8. PC- Relative Mode		

Happy Learning.!