

COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and
Addressing Modes

Addressing Modes-2

Lecture_06



Vijay Agarwal sir



A yellow diamond-shaped sign with a black border and black text, mounted on a silver pole. The sign reads "TOPICS TO BE COVERED". Below the sign is a white and orange striped barrier.

**TOPICS
TO BE
COVERED**



A red diamond-shaped icon with a white border and white text, containing the number "01".

01

Addressing Modes

ADDRESSING MODE



Addressing Modes

Data *Registers*
 ↓
 Memory.

Addressing Modes

Displacement AM

- ① Immediate AM
- ② Memory Direct / Absolute AM
- ③ Memory Indirect AM
- ④ Register Direct AM
- ⑤ Registers Indirect AM
- ⑥ PC Relative AM
- ⑦ Index Reg AM
- ⑧ Based Reg AM
- ⑨ Auto Decrement AM
- ⑩ Auto Increment AM
- ⑪ Implied/Implicit AM

'DATA'
(OPERAND)

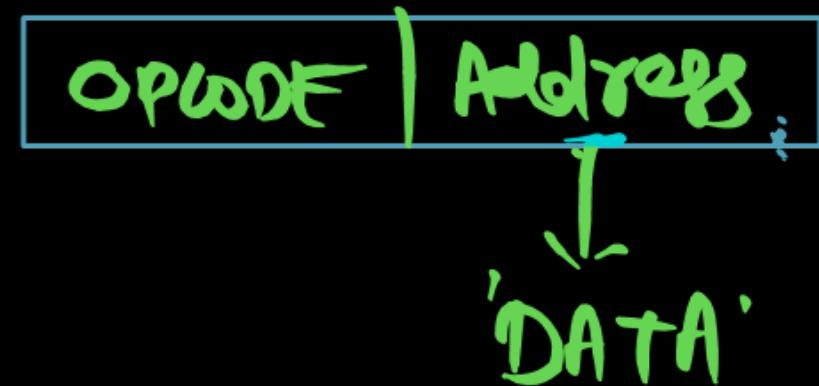
← WHERE it is (operand)
(address)
EA [Effective Address].

Addressing Modes

① Immediate AM : In this AM operand are present in Address field of the Instruction.

OR

operand are present in the Instruction It self.



Note

This AM are used to Access the Constant @ initialize the Register with Constant Value.

Addressing Modes

(eg)



ADD R₁ # 500

R₁ \leftarrow R₁ + 500

Mov R₀ #10

(OR)

MVI R₀ 10

R₀ \leftarrow 10

Addressing Modes

③ Immediate AM

MVI R₁, 4000
 DATA

(OR)

MOV R₁, #4000

R₁ ∈ 4000

MVI R₁, 600

R₁ ∈ 600

(eg)

ADD I R₀ 4000

(@)

ADD R₀ #4000

R₀ ∈ R₀ + 4000

600

2L

600

Memory

ADD F R₀ 600

R₀ ∈ R₀ + 600

4000

Immediate AM has some limitation.

most required
↑ Storage

① Immediate AM Can not be Used as Destination Address.

Bcz Constant does not have Any Storage.

②  dest source
MOV 100 R1

100 \leftarrow R1 X

② The Range of Constant is Limited by the Size of Address field.

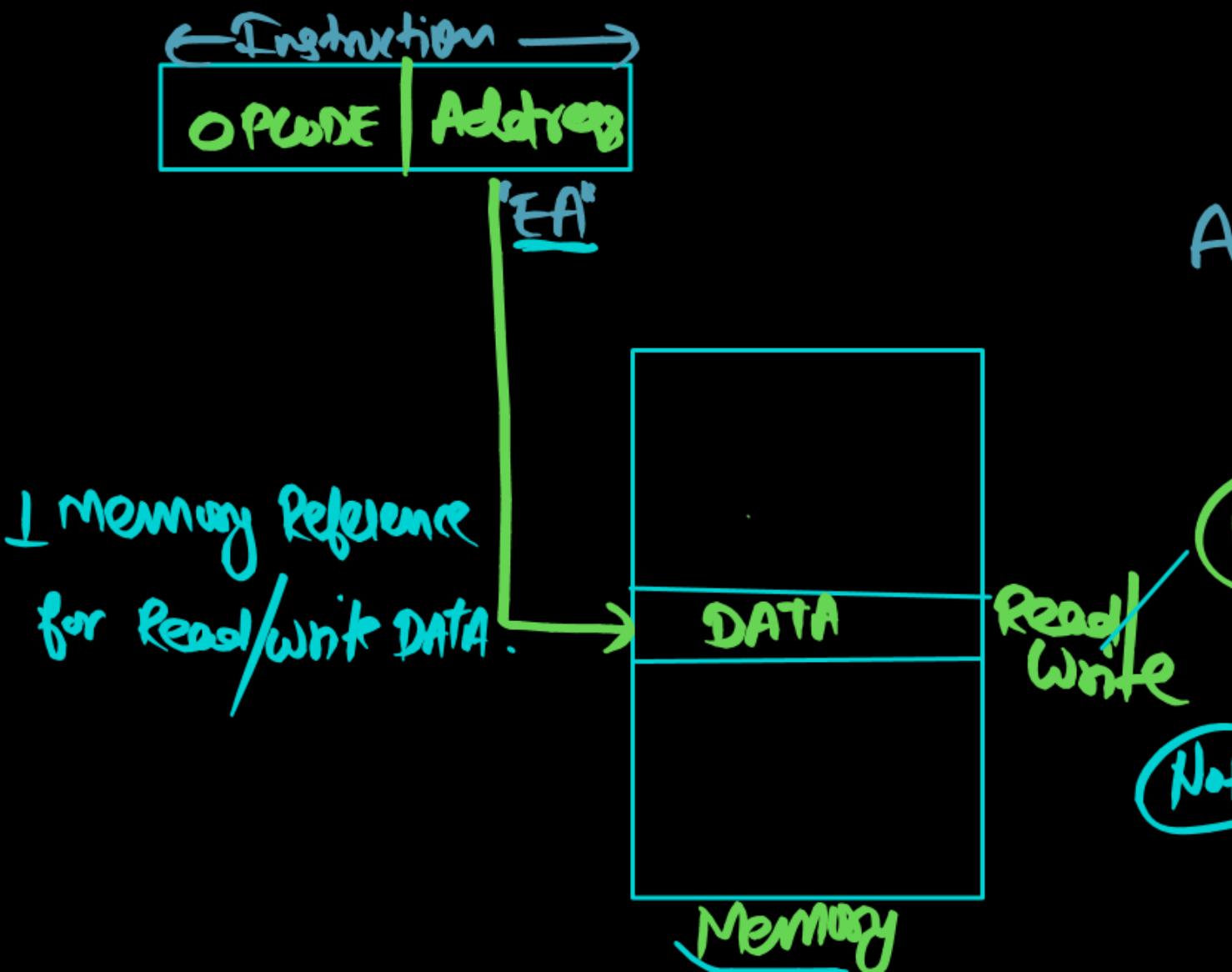
n bit Unsigned Range = 0 to $2^n - 1$

n bit Signed Range : $(2^{n-1})_F + (2^{n-1})_I$.

Addressing Modes

② Memory Direct | Absolute AM.

In this AM operand are Present in the memory & instruction contain the Effective Address.



OR

Address field of the Instruction specify the Effective Address.

Note

This Direct(Memory Direct) | Absolute AM. are Used to Access the Variable.

Note

! Memory Ref. Required for Access Data.

(S) MDV R₁ [1000]

$$R_1 \leftarrow M[1000]$$

(S) ADD R₂ [500]

$$R_2 \leftarrow R_2 + M[500]$$

Addressing Modes

③ Direct AM.

MOV R₁, M[4000]

R₁ ∈ M[4000]

R₁ = 600

④ ADD R₀ (4000)

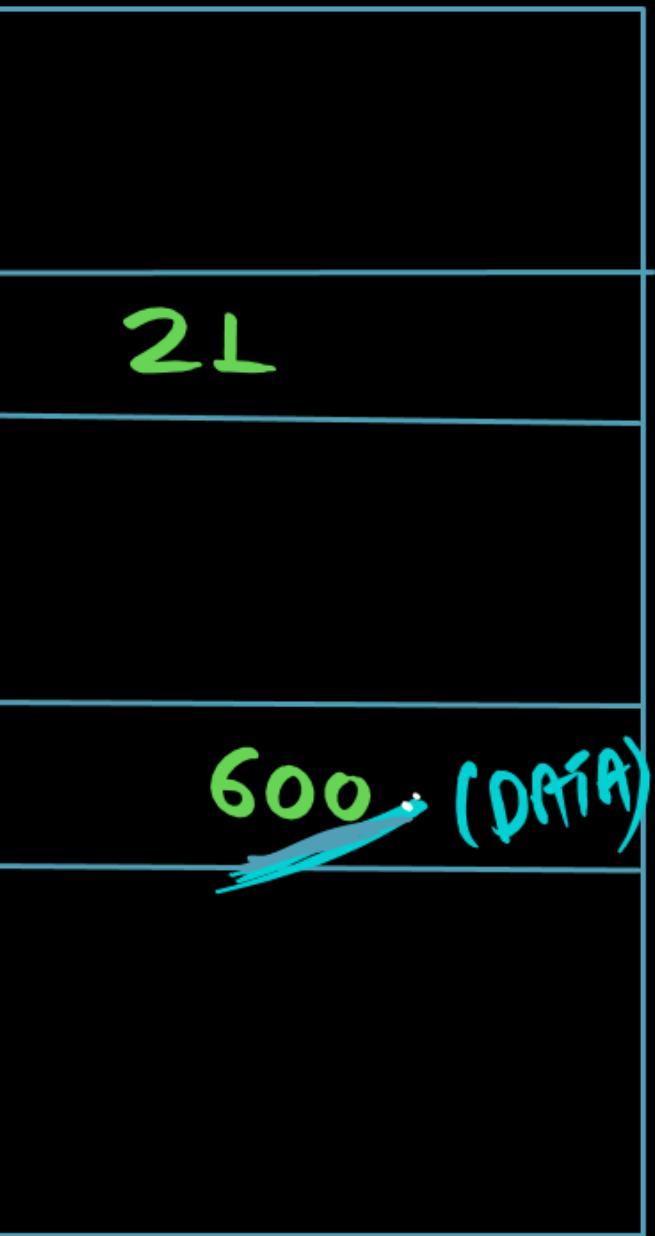
R₀ ∈ R₀ + M[4000]

R₀ ∈ R₀ + 600

R₀ ∈ R₀ + 600

600

4000



Memory

(e) Direct AM:

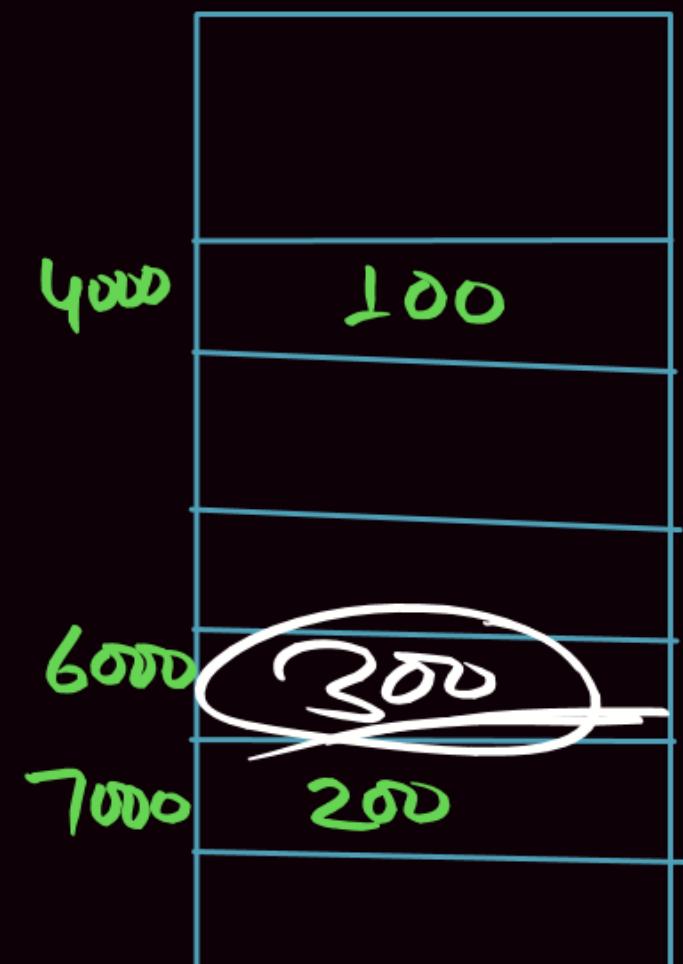
	Destn	Source 1	Source 2
ADD	[6000]	[4000]	[7000]

$$M[6000] \leftarrow M[4000] + M[7000]$$

↑ ↑ ↓
 1 Mem Ref 1 Mem Ref 1 Mem Ref
 for Write DATA for Read DATA for Read DATA

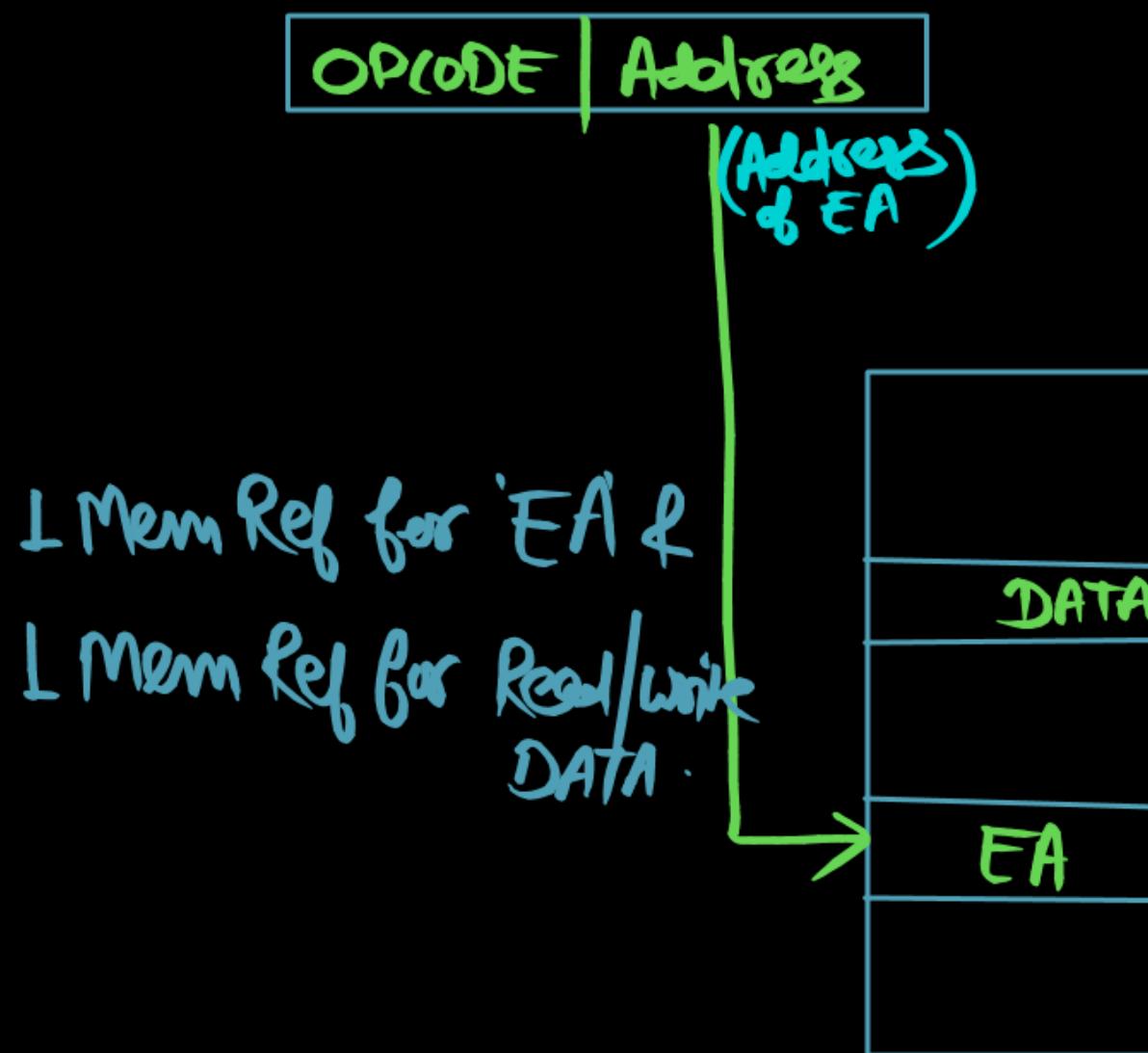
$$M[6000] \leftarrow 100 + 200$$

$M[6000] \leftarrow 300$



Addressing Modes

- ③ Memory Indirect AM: In this AM operand are placed in the memory. Effective Address (EA) is stored in memory. Instruction contain the Addresses of Effective Address.



Note Mem Indirect AM Used to Access the Pointer:
In Memory Indirect AM '2 Memory Reference' are Required.
:

Addressing Modes

(3)

MOV R₁ @1000R₁ ∈ M[1000]R₁ ∈ M(100)R₁ ∈ 16

(3)

ADD R₂ (1000)R₂ ← R₂ + M[1000]R₂ ← R₂ + M(100)R₂ ← R₂ + 16

100

16

1000

100

Addressing Modes

③ Memory Indirect

$MOV_R_i @4000$

...

$R_i \in M[4000]$

$R_i \in M[600]$

$R_i \in 2L$

'Pointers'

④ ADD R_o @4000

$R_o \leftarrow R_o + M[4000]$

$R_o \leftarrow R_o + M[600]$

$R_o \leftarrow R_o + 2L$

$R_o \leftarrow R_o + 2L$

4000
Address of
Effective Addr.

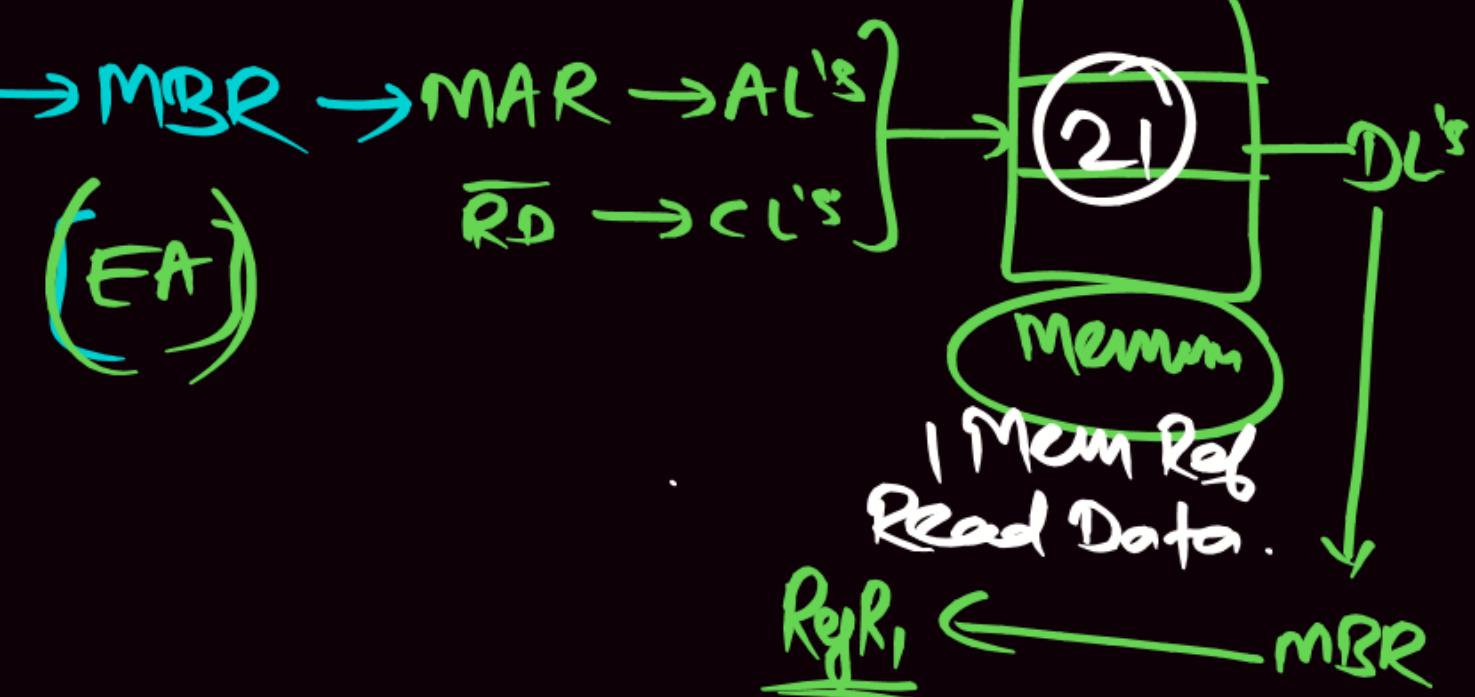
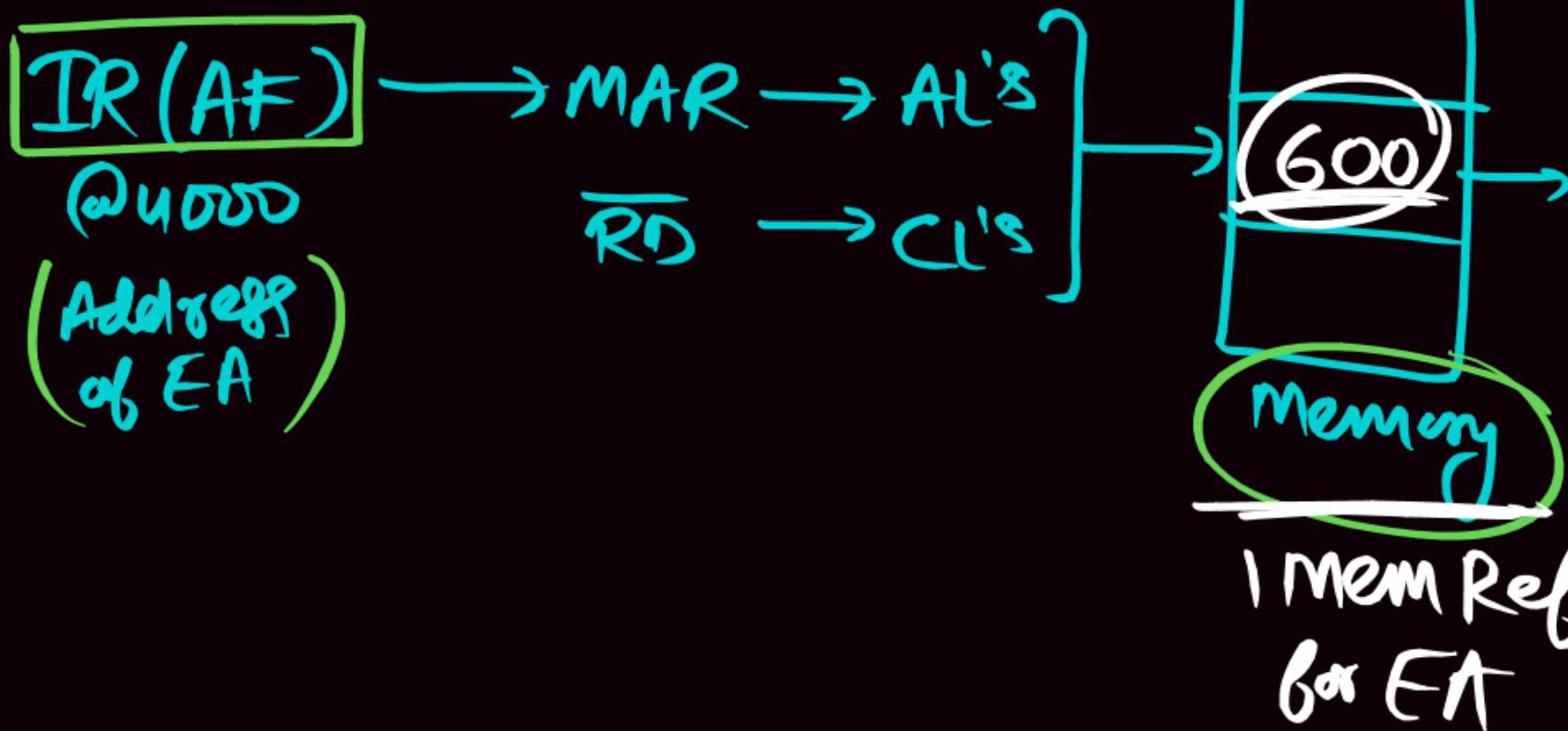


Memory

DR MAR, @4000

MOV R1 @4000

Execution



R1 ← 21

④

	Dest ⁿ	Source ¹	Source ²
ADD	[6000]	@7000	([4000])

$$M[6000] \leftarrow M[7000] + M[4000]$$

↓
Mem Direct

↓
1 Mem Ref. for
write DATA

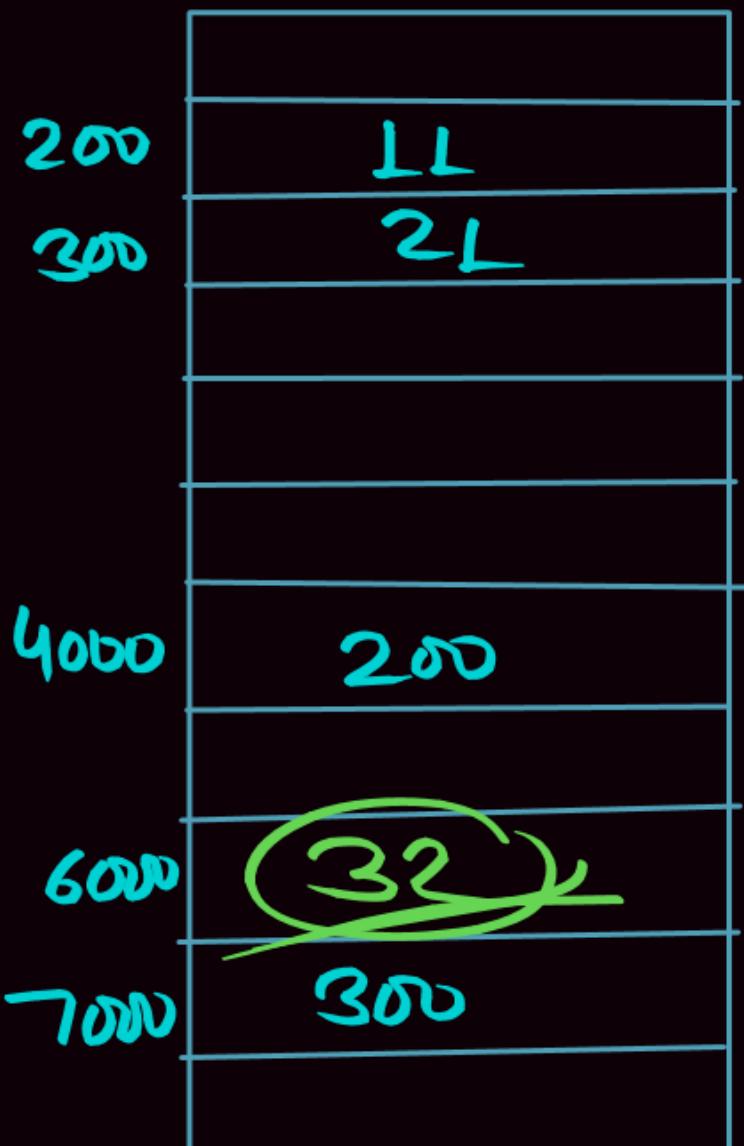
↓
Mem Indirect

↓
1 Mem Ref. for EA
1 Mem Ref. for Read
DATA

↓
Mem Indirect
↓
1 Mem Ref. for EA
1 Mem Ref. for Read
DATA

$$\begin{aligned} M[6000] &\leftarrow M[300] + M[200] \\ &\leftarrow 2! + LL \end{aligned}$$

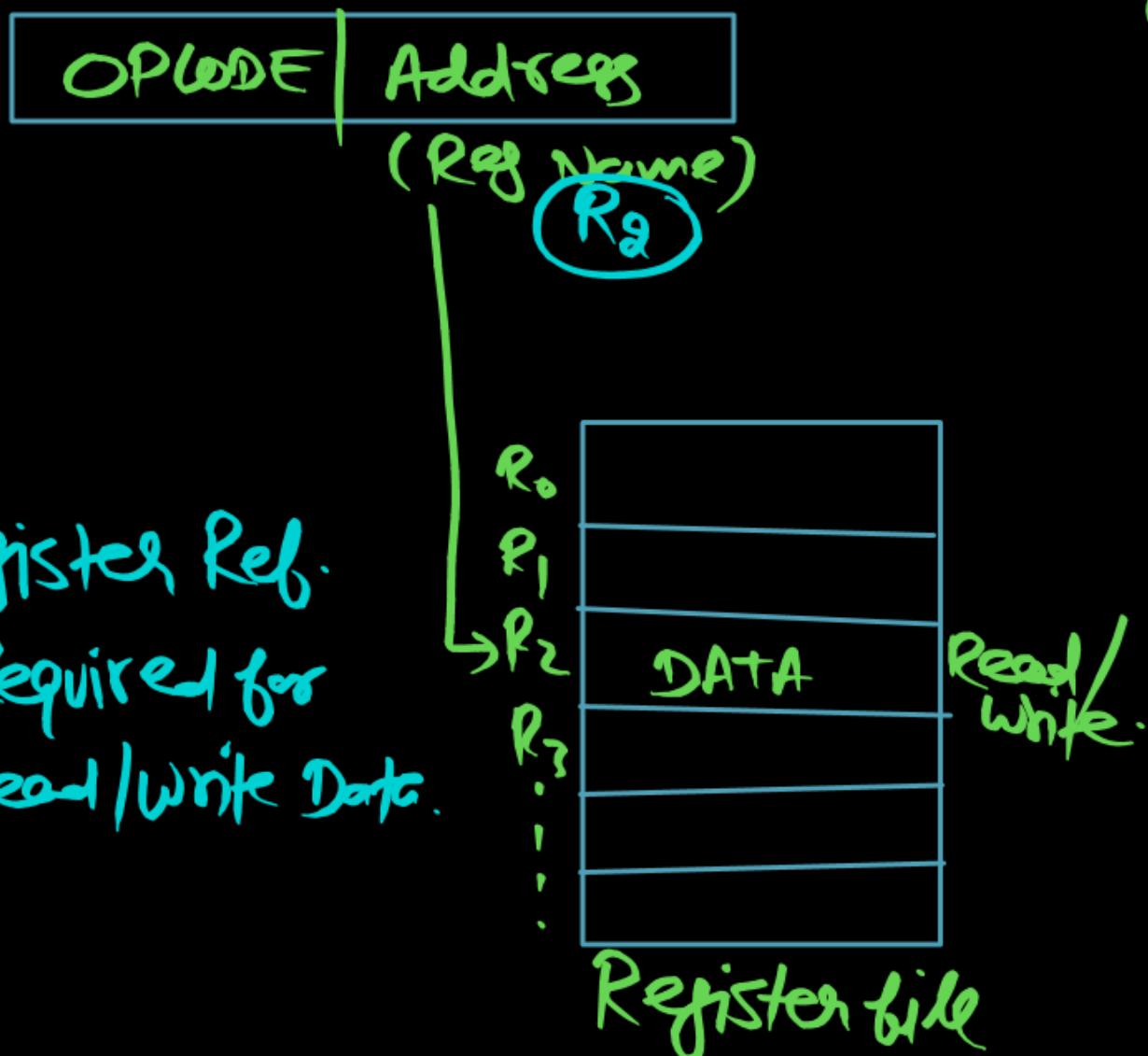
M[6000] ∈ 32



Addressing Modes

④ Register Direct AM

: This AM is same as Memory Direct AM but the Difference Here the operand are present in Register instead of Memory.



In this AM Operand Stored in the Register & that Register Address (Register Name) is Maintain in Address Field of the Instruction.

Addressing Modes

Destⁿ Source
MOV R₁, R₂

R₁ ← R₂

I Reg Ref
for Write

I Reg Ref
for Read

I

ADD

Dest n

[6000]

↓
Mem Direct

↓
1 Mem Ref
for Write DATA

Source 1

R1

↓
Reg Direct

↓
1 Reg Ref for
Read DATA

Source 2

@ 7000

↓
Mem Indirect

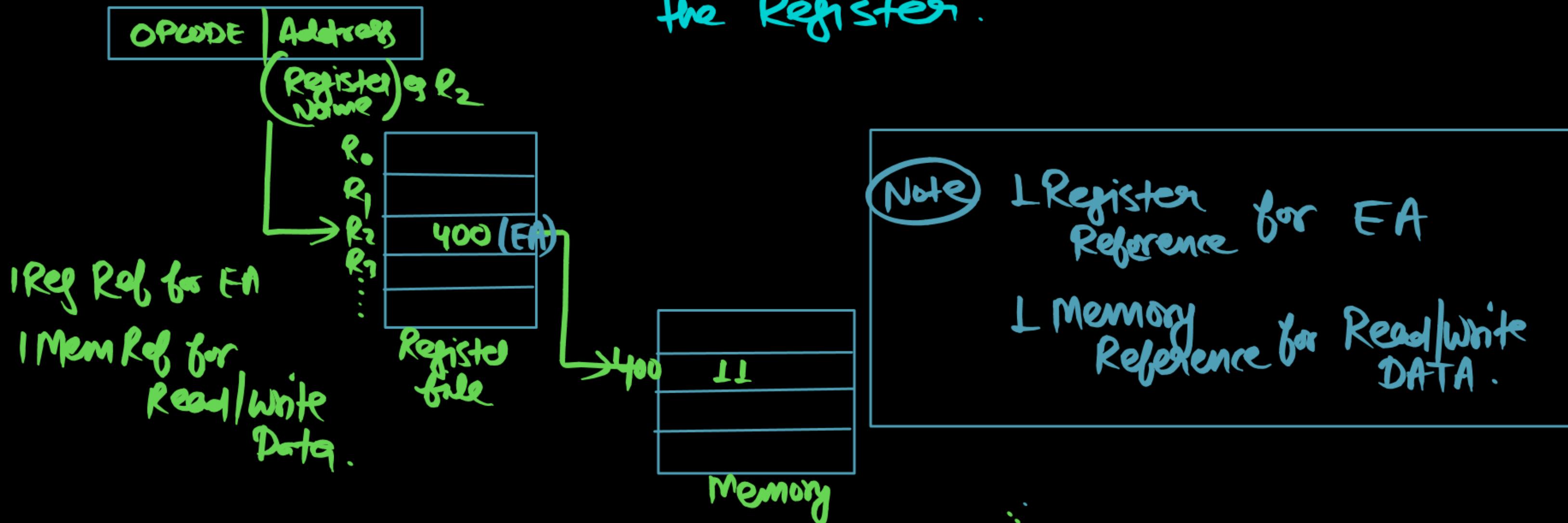
↓
1 Mem Ref for EA
1 Mem Ref for
Read DATA

3 Mem Ref.
1 Reg Ref.

∴ 3 Memory Ref
1 Register Ref.

Addressing Modes

⑤ Register Indirect : In this AM OPERAND are placed in the Memory, & Effective Address Present in the Register.



Addressing Modes

③ $\text{MOV } R_1 \text{, } @R_2$

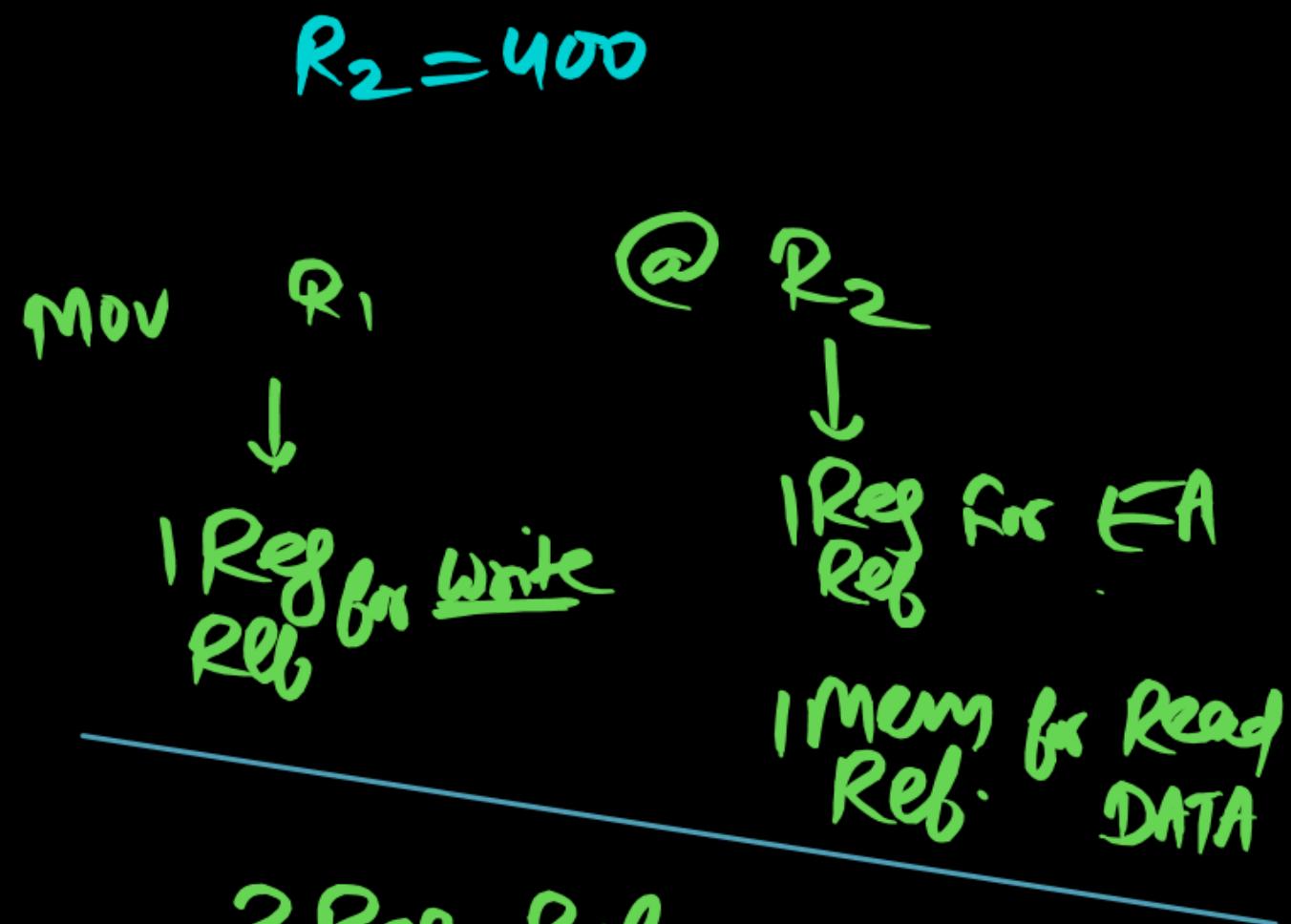
Destn Source

$$R_1 \leftarrow @R_2$$

$$R_1 \leftarrow M[400]$$

$R_1 \leftarrow I$

$$@R_2 \equiv M[R_2]$$



2 Reg. Ref.

1 Mem Ref.

Addressing Modes

ADD

[6000]

@7000

@RL

@ RL

M[RI]

Memory
Direct

|
Memory Ref for
Write DATA

4 Mem Ref
1 Register Ref.

↓
Mem.
Indirect

| Mem Ref for EA
| Mem Ref for
Read DATA

↑
Registered
Indirect

↓
| Reg Ref for EA
| Mem Ref for Read DATA .

Addressing Modes

Q: WHY Register Indirect AM Used Sometimes instead of Memory Indirect ?

Soln (i) To Shorten the Instruction length .

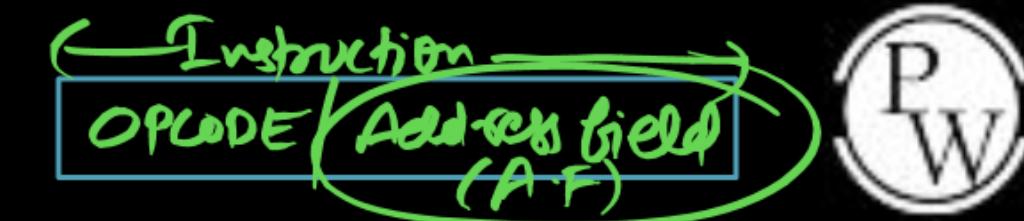
(ii) Accessing the Register very fast compare to Memory .

Q) Memory of LGGB than 34 bit Memory address But in the Process we have 32, 64 Register So it Require 5@6 bit to Access the Register .

So Two Advantage
Majorally

- ① Instr Length become Short .
- ② Fast Accessing .

Addressing Modes



Displacement AM.

① Relative AM

Pc - Relative AM
[Relative AM]

$$EA = \underbrace{Pc}_{\text{value}} + \underbrace{\text{Address field}}_{\text{OFFSET}} + \underbrace{A.F}_{\text{OFFSET}}$$

(OR)

$$EA = \underbrace{Pc}_{\text{value}} + \underbrace{\text{Relative value.}}_{\text{OFFSET}}$$

Index Reg AM

② Index-Register AM

$$EA = \underbrace{\text{Index Reg value}}_{\text{Base}} + \underbrace{A.F}_{\text{OFFSET}}$$

Base-Reg AM.

③ Base Register AM.

$$EA = \underbrace{\text{Base Reg value}}_{\text{Base}} + \underbrace{A.F}_{\text{OFFSET}}$$

↓
Array Implementation

$$\boxed{EA = \underset{\text{Value}}{\underset{\uparrow}{\text{XR}}} + AF}$$

Index Reg.

| Register Ref for Index Value .

L ALU (Arithmetic) Ref for EA Calculation

L Mem Ref for Read / Write DATA .

..

Addressing Modes

(9 & 10)

Auto Decrement & Increment AM:

in which Register Value Decrement & Increment.

Decrement : Pre decrement [first Decrement the Ref Value then Access the DATA].

Increment : Post Increment

(9) Job Salary Negotiation
Promotion (After 1 year)

Addressing Modes

⑪ Implied / Implicit AM. : In this AM Operand (DATA Info) are present in the opcode itself.

OPCODE

Type of operation

DATA Information

③

STC

$CY=1$

↳ Set Carry

④

CLC

$CY=0$

↳ Clear Carry

③ . ADD

TOS (POP)
TOS (POP)
 \oplus
TOS (PUSH)

④ INCA

Stack Based org Used Implied | Implicit AM.

Addressing Modes

V. Imb

Constant — Immediate AM.

Variable — Direct AM.

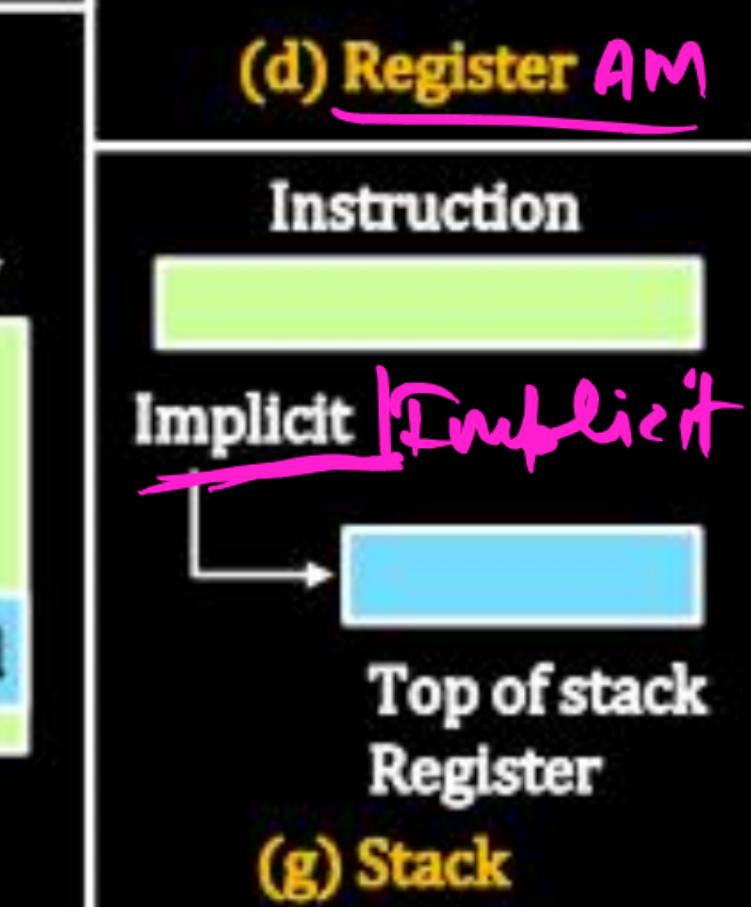
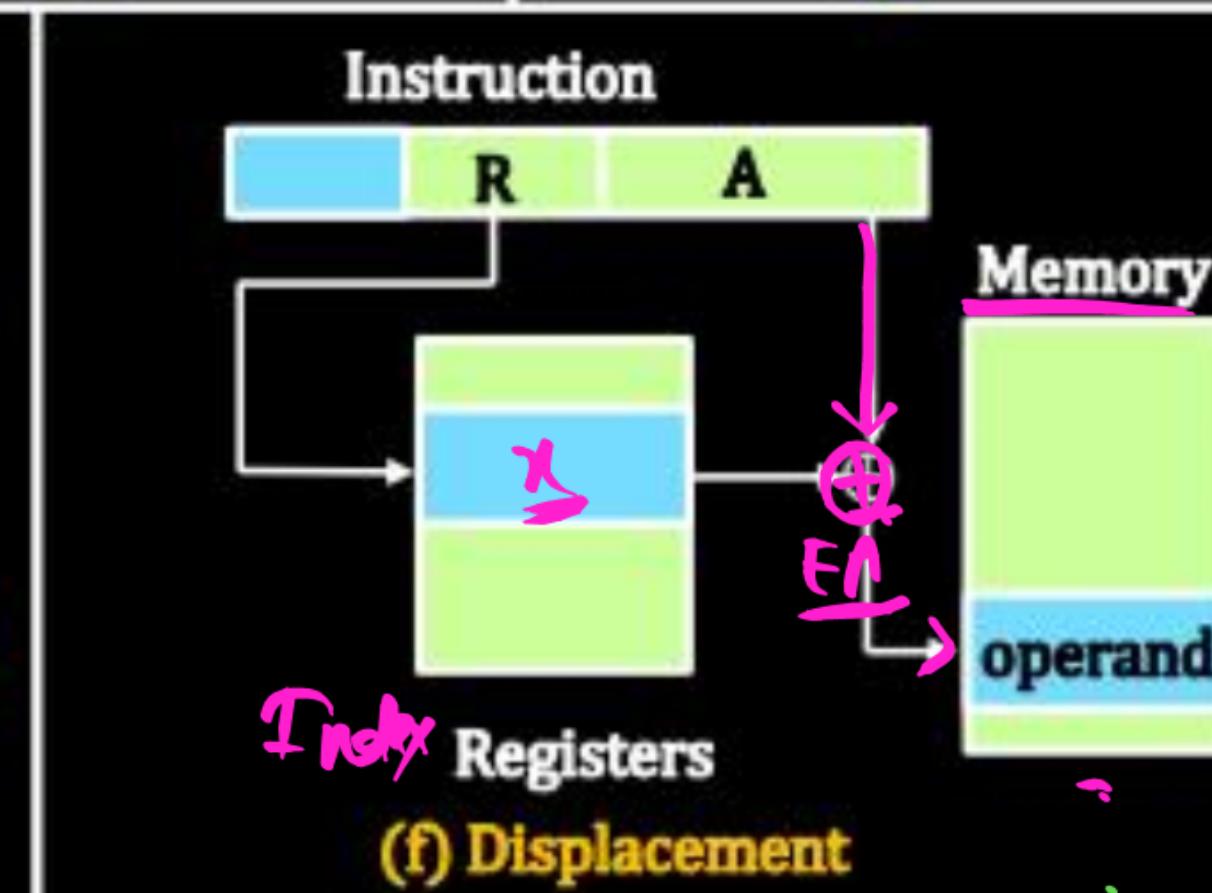
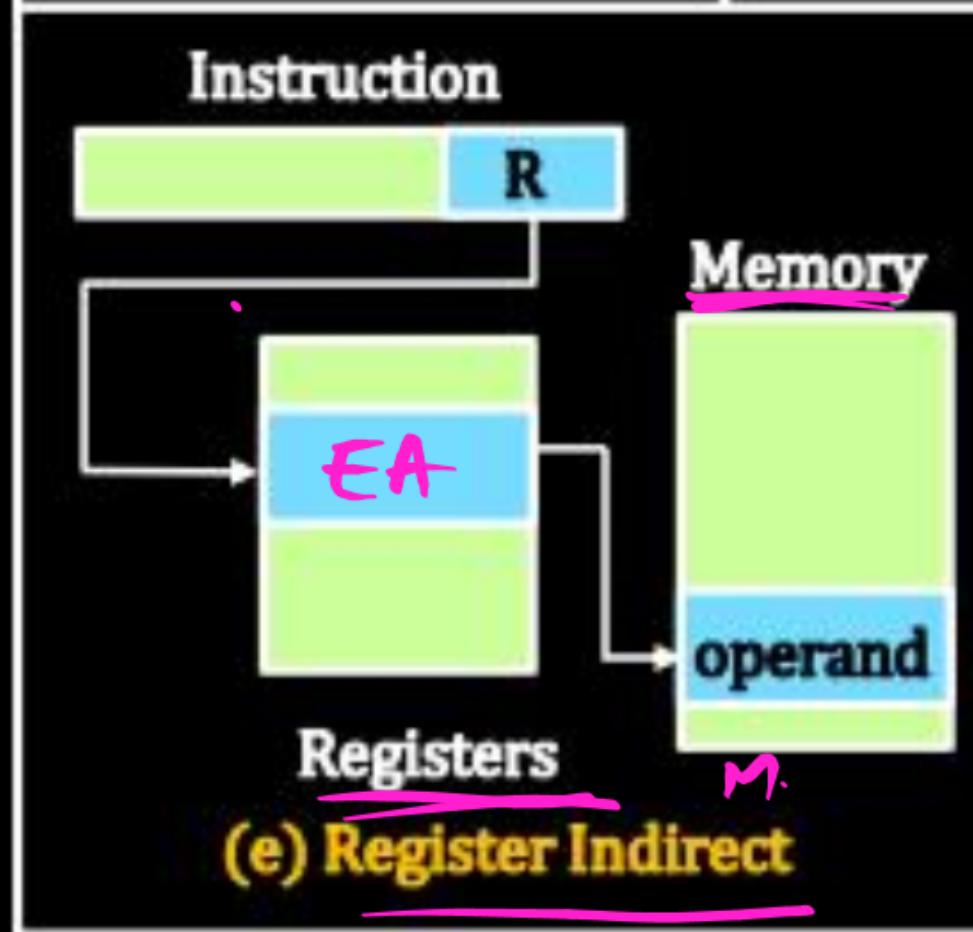
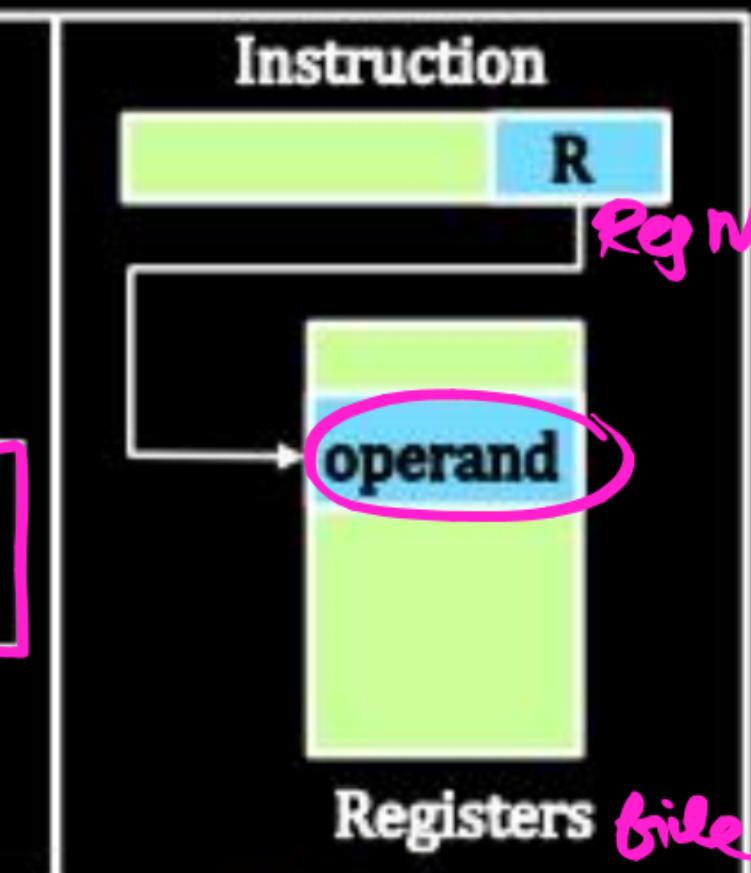
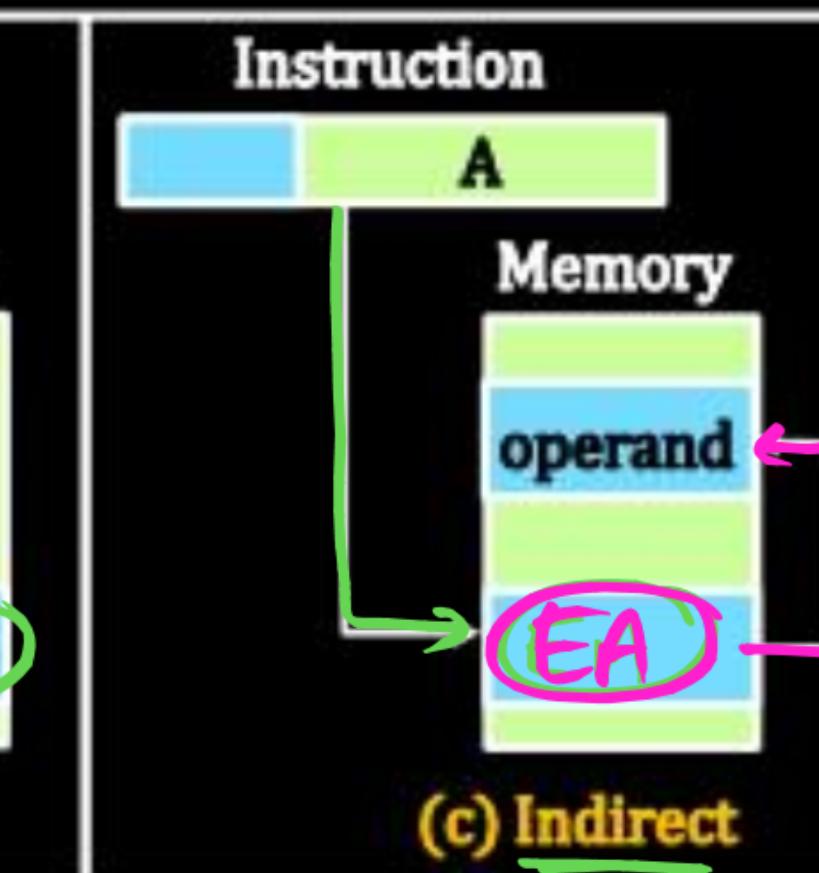
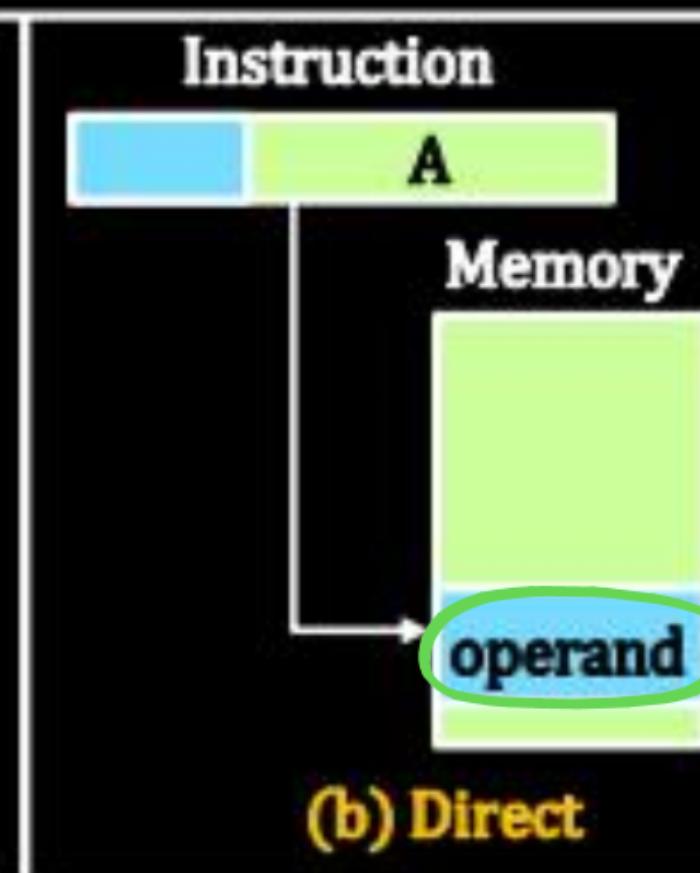
Pointers — Mem. Indirect AM.

• Array — Index Reg. AM.

• Reallocation — Base Reg. AM.

Addressing Modes

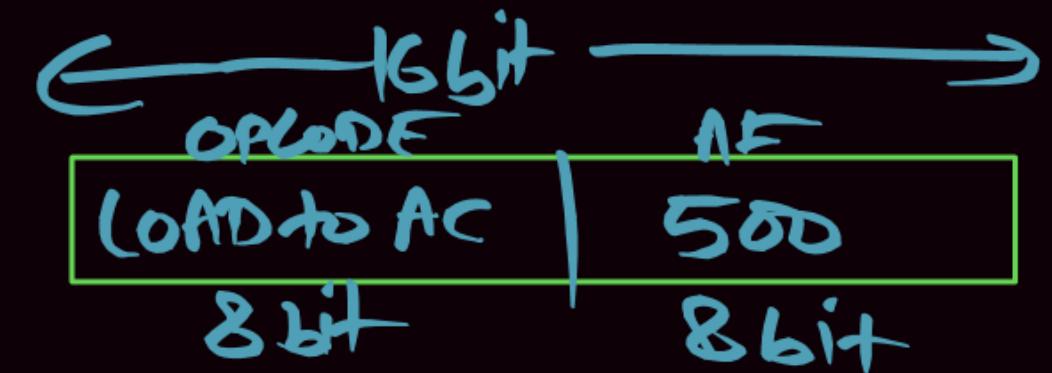
- Immediate
- Direct
- Indirect
- Register
- Register Indirect
- Displacement
- Stack



Q Consider a 16 bit Instruction Load to AC with Address 500

Starting from memory location 200 onwards.

8bit OPCODE



XR: Index
Register

Q) Consider a 16bit Instn 8bit opcode LOAD to AC



Starting 200 now

Addressing Mode	Effective Address	Content Of AC
Direct address		
Immediate Operand		
Indirect Address		
<u>Relative address</u>		
<u>Indexed address</u>		
Register		
Register Indirect		
Autoincrement		
Autodecrement		

PC = 200
R1 = 400
<i>Index</i> XR = 100
AC

Address	Memory
200	Load to AC Mode
201	Address = 500
202	Next instruction
399	450
400	700
500	800
600	900
702	325
800	300

Numerical example for addressing modes.

Addressing Mode	Effective Address	Content Of AC	
			$PC = 200$
Direct address	500	800	$R1 = 400$
Immediate Operand	201	500	$XR = 100$
Indirect Address	800	300	AC
Relative address	702	325	
Indexed address	600	900	
Register	--	400	
Register Indirect	400	700	
Autoincrement	400	700	
Autodecrement	399	450	

Address	Memory	
200	Load to AC	Mode
201	Address = 500	
202	Next instruction	
399	450	
400	700	
500	800	
600	900	
702	325	
800	300	

Numerical example for addressing modes.

Eight addressing modes for the load instruction

Mode	Assembly Convention	Register Transfer
Direct address	LD ADR	$AC \leftarrow M[ADR]$
Indirect address	LD @ADR	$AC \leftarrow M[M[ADR]]$
Relative address	LD \$ADR	$AC \leftarrow M[PC + ADR]$
Immediate operand	LD#NBR	$AC \leftarrow NBR$
Index addressing	LD ADR(X)	$AC \leftarrow M[ADR + XR]$
Register	LD R1	$AC \leftarrow R1$
Register indirect	LD (R1)	$AC \leftarrow M[R1]$
Autoincrement	LD (R1)+	$AC \leftarrow M[R1], R1 \leftarrow R1 + 1$

Q.

In which of the following addressing modes, operand is NOT
A part of instruction?

[MSQ]

A Immediate

B Direct

C Indirect

D Register

P
W

Q. Consider a 16 bit hypothetical processor which support 1 Address Instruction Design with various addressing mode. It contain 8 bit OPCODE. It supports 1 word Instruction stored in the Memory with a starting address of $(745)_{10}$ (Decimal) onwards. Address field value of the instruction is 222. Register r_1 contain 111 memory content of [222] is 155. Which of the following is/are correct about effective address of various Addressing Mode (AM)? (all values are in decimal)

[MSQ]

- (I) In the Immediate AM Effective Address is 745.
- (II) In the Immediate AM Effective Address is 746.
- (III) In the Memory Indirect AM Effective Address is 155.
- (IV) In the Index AM effective Address is 333
(r_1 as index register)

Q.1

The most appropriate matching for the following pairs

X. Indirect addressing

1. Loops

Y. Immediate addressing

2. Pointers

Z. Auto decrement addressing

3. Constants

[GATE - 2000: 1 Mark]

A X - 3 Y - 2 Z - 1

B X - 1 Y - 3 Z - 2

C X - 2 Y - 3 Z - 1

D X - 3 Y - 1 Z - 2

Q.3

If we use internal data forwarding to speed up the performance of a CPU (R1, R2 and R3 are registers and M[100] is a memory reference), then the sequence of operations.

$$R1 \rightarrow M[100]$$

$$M[100] \rightarrow R2$$

$M[100] \rightarrow R3$ can be replaced by

[GATE - 2004: 2 Mark]

- A** $R1 \rightarrow R3$
 $R2 \rightarrow M[100]$

- B** $M[100] \rightarrow R2$
 $R1 \rightarrow R2$
 $R1 \rightarrow R3$

- C** $R1 \rightarrow M[100]$
 $R2 \rightarrow R3$

- D** $R1 \rightarrow R2$
 $R1 \rightarrow R3$
 $R1 \rightarrow M[100]$

Q.4

Match List-I with List-II and select the correct answer using the codes given below the lists:

[GATE - 2005: 2 Mark]

List-I

- A. $A[I] = B[J];$
- B. `while[*A++];`
- C. `int temp = *X;`

List-II

- 1. Indirect addressing
- 2. Indexed addressing
- 3. Auto increment

Codes:

	A	B	C
A	3	2	1
B	1	3	2
C	2	3	1
D	1	2	3

Q.5

The memory locations 1000, 1001 and 1020 have data values 18, 1 and 16 respectively before the following program is executed.

[GATE - 2006: 2 Mark]

MOVI	Rs, 1	Move immediate
LOAD	Rd, 1000(Rs)	Load from memory
ADD I	Rd, 1000	Add immediate
STOREI	<u>0(Rd), 20</u>	Store immediate

$$Rd \leftarrow M(1000 + Rs)$$

$$Rd \leftarrow M(1000 + Rs)$$

$$Add\ immediate$$

$$M(0 + Rd) \leftarrow 20$$

$$M(0 + Rd) \leftarrow 20$$

Which of the statements below is TRUE after the program is executed?

- A** Memory location 1000 has value 20
- B** Memory location 1020 has value 20
- C** Memory location 1021 has value 20
- D** Memory location 1001 has value 20

Q.7

The absolute addressing mode

[GATE - 2002: 1 Mark]

P
W

- A** The operand is inside the instruction
- B** The address of the operand is inside the instruction
- C** The register containing the address of the operand is specified inside the instruction.
- D** The location of the operand is implicit.

Q. 8

A CPU has 24-bit instructions. A program starts at address 300

(in decimal). Which one of the following is a legal program counter
(all values in decimal)?

[GATE-1 Marks]

- (a) 400
- (b) 500
- (c) 600
- (d) 700

COMMON DATA QUESTION (9 -10)

Consider the following program segment, Here R1, R2 and R3 are the general purpose register.

Instruction	Operation	Instruction size (no. of words)
MOV R1, (3000)	R1 \leftarrow M[3000]	2
LOOP; $R = 10$		
MOV R2, M[R3]	R2 \leftarrow M[R3]	1
ADD R2, R1	R2 \leftarrow R1 + R2	1
MOV (R3), R2	M[R3] \leftarrow R2	1
INC R3	R3 \leftarrow R3+1	1
DEC R1	R1 \leftarrow R1-1	1
BNZ LOOP	Branch on not zero	2
HALT		Stop

Assume that the content of memory location 3000 is 10 and the content of the Register R3 is 2000. The content of each of the memory locations from 2000 to 2010 is 100. The program is loaded from the memory location 1000. All the Numbers are in decimal.

Q.9

Assume that the memory is word addressable. The number of memory references for accessing the data in executing the program completely is

[2 marks]

- (a) 10
- (b) 11
- (c) 20
- (d) 21

COMMON DATA QUESTION (9 - 10)

Consider the following program segment, Here R1, R2 and R3 are the general purpose register.

Instruction	Operation	Instruction size (no. of words)
MOV R1, (3000)	$R1 \leftarrow M[3000]$	2
LOOP:		
MOV R2, M[R3]	$R2 \leftarrow M[R3]$	1
ADD R2, R1	$R2 \leftarrow R1 + R2$	1
MOV (R3), R2	$M[R3] \leftarrow R2$	1
INC R3	$R3 \leftarrow R3 + 1$	1
DEC R1	$R1 \leftarrow R1 - 1$	1
BNZ LOOP	Branch on not zero	2
HALT		Stop

Assume that the content of memory location 3000 is 10 and the content of the Register R3 is 2000. The content of each of the memory locations from 2000 to 2010 is 100. The program is loaded from the memory location 1000. All the Numbers are in decimal.

Q.10 Assume that the memory is word addressable. After the execution of this program, the content of memory location 2010 is

[2 marks]

- (a) 100
- (b) 101
- (c) 102
- (d) 110

**THANK
YOU!**

