COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and Addressing Modes

Lecture_06

Vijay Agarwal sir

01 Expand Opcode Technique

02 Addressing Modes

1. Machine Instruction
2. Instruction Format
3. <u>ISA</u>
   1. Stack Based org
   2. Accumulator Based org
   3. General Register org.

Expand Opcode Technique:

Expand Opcode Technique

Expand Opcode Technique

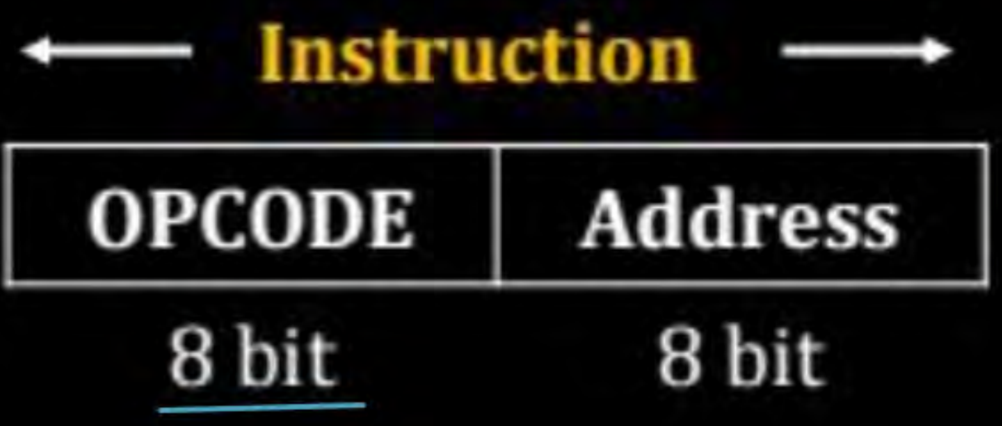
Expand opcode length is required in the fixed length instruction supported CPU Design to implement the various instruction with different formats.

Variable Length Instruction Supported CPU Design

OPCODE = 8 bit

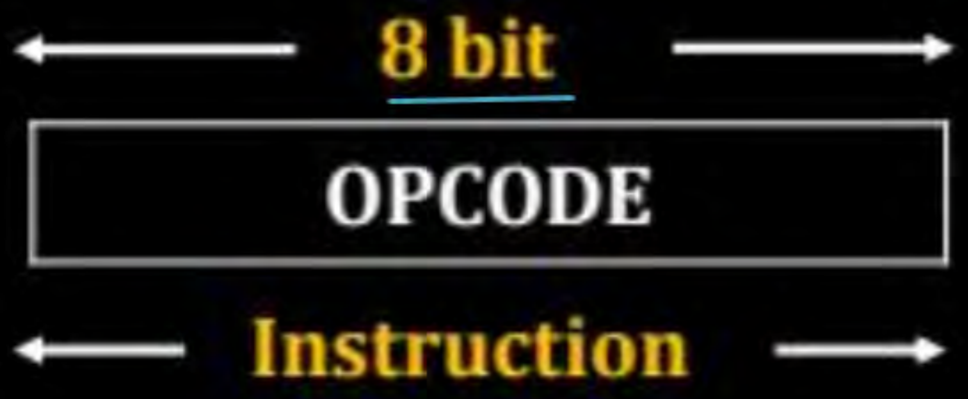
Address field = 8 bit

(i) 1 Address Instruction Design:



= 16 bit

(ii) 0 Address Instruction Design:



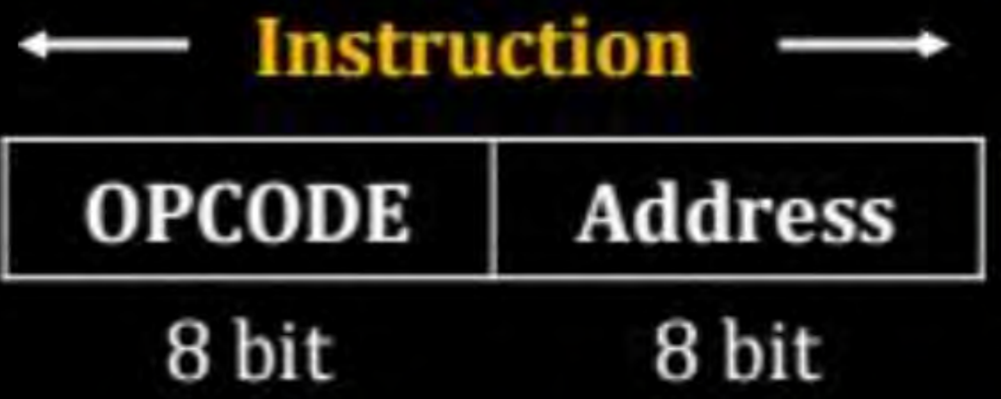
= 8 bit [No need of expand opcode technique]

Fixed Length Instruction Supported CPU Design

OPCODE = 8 bit

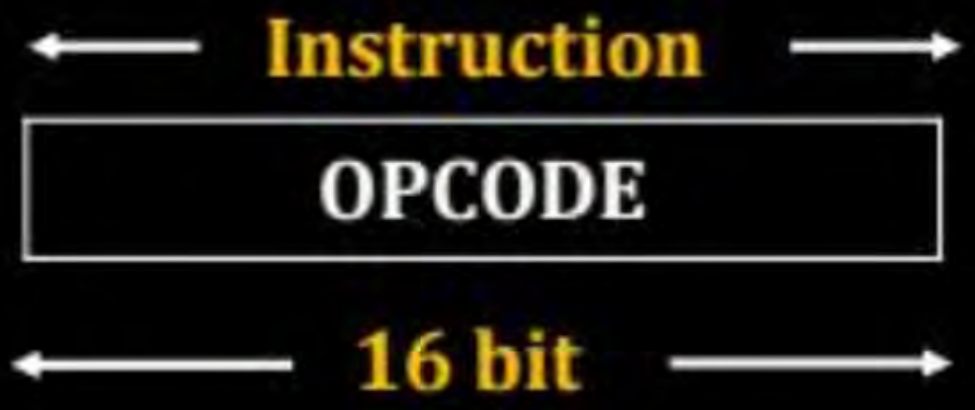
A.F = 8 bit

(i) 1 Address Instruction Design:



= 16 bit

(ii) 0 Address Instruction Design:



= 16 bit [Here expand opcode technique required]

Expand Opcode Technique

- Primitive instruction means smallest opcode instruction.

Step 1: Identify the primitive instruction in the CPU.

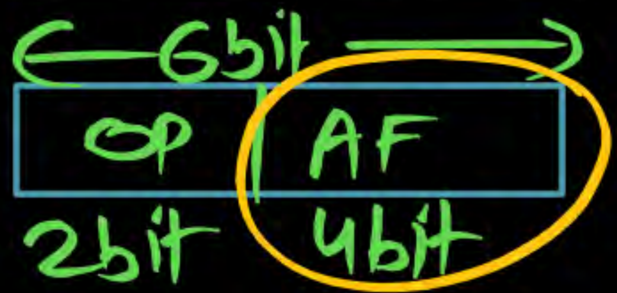
Step 2: Calculate the total number of possible operation.

Step 3: Identify the free opcode after allocating the existed instruction

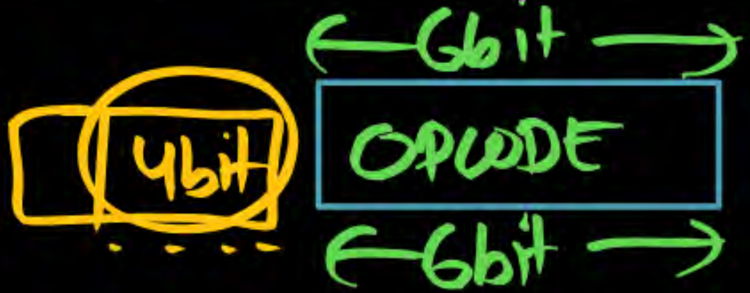
Step 4: Calculate the number of Derived instruction possible by multiply

$$\boxed{\text{Free opcode} \times 2^{\text{Increment bit in opcode}}}$$

Q. Consider a processor which support 6 bit instruction and 4 bit address field. If there exist 2 one address instruction then how many 0 address instruction can be formulated?



Primitive.



Derived.

$\boxed{\begin{matrix}00\\01\end{matrix}}$ USED

10
11

10 - - - -   11 - - - -

&

10 1 1 1 1 $\underbrace{}_{16}$ 11 1 1 1 1 $\underbrace{}_{16}$

Total # operation in 1AF | 1AI $= 2^2 = \boxed{4}$

Given 1AI $= 2$

Free $= 4 - 2 = \boxed{2}$

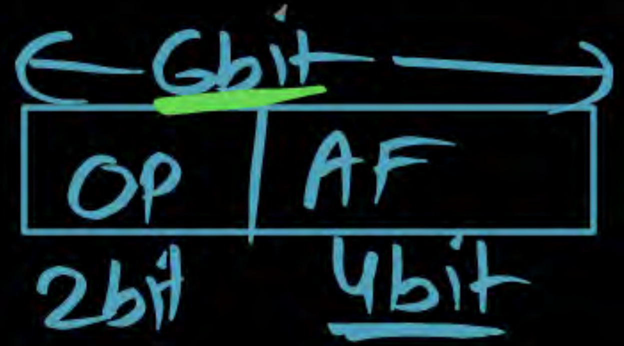
Total # operation in 0AI $=$ Free opcode $\times 2^{\text{Increment bit in opcode}}$

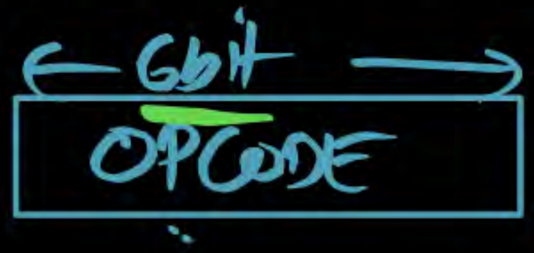
$\Rightarrow 2 \times 2^{6-2} = 2 \times 2^4$

$= \boxed{32}$ Ans

Q.

Consider a processor which support 6 bit instruction and 4 bit address field. If there exist 2 one address instruction then how many 0 address instruction can be formulated?





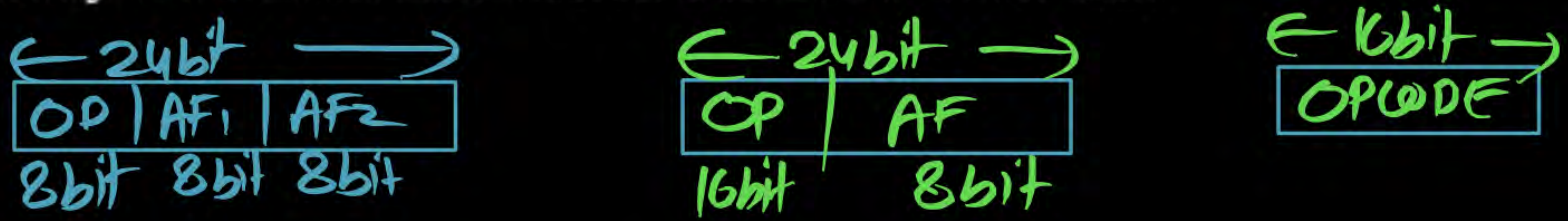
$$2^6 = 2 \times 2^4 + X$$

$$64 = 32 + X$$

$$X = 64 - 32 = \underline{\boxed{32}}$$

Q. ✓

Consider a processor which contain 8 bit word and 256 word memory. It support 3 word instruction. If these exist 254 2-address instruction and 256 1-address instruction then how many 0 address instruction can be formulated?



$$\boxed{2^{24} = 254 \times 2^{8} \times 2^{8} + 256 \times 2^{8} + X}$$

Q. Consider a processor with 16 bit instruction. Processor has 15 registers and support 2 address instruction and 1 address instruction. If processor support 256 1-address instruction then number of 2-address instruction are

(A) 128

(B) 192

[✓ (C) 240]

(D) 248

← 16bit →

OP	AF1	AF2
$\boxed{8}$	4	4

← 16 →

OP	AF1
12bit	$\boxed{4bit}$

$$2^{16} = \underline{N \times 2^4 \times 2^4} + 256 \times 2^4$$

$$2^{16} = N \times 2^8 + 2^8 \times 2^4$$

$$2^{16} = 2^8[N + 16]$$

$$2^8 = N + 16$$

$$256 = N + 16$$

$$N = 256 - 16$$

$$= \boxed{240}$$ Ans

Solution(c): 240

15 register $= 2^4 \Rightarrow$ Register A.F = 4 bit

$\xleftarrow{\quad}$ **16 bit** $\xrightarrow{\quad}$

OPCODE	**Register A.F**	**Register A.F**
8 bit	4 bit	4 bit

OPCDE field $= 16 - (4 + 4) = 8$ bit

So total number of 2 address instruction $= 2^8 = 256$

Let 'x' 2 address instruction used

Number of free opcode $= (2^8 - x)$

1 Address field

OPCODE	A.F
12 bit	4 bit

Total no of 1 address instruction = $(2^8 - x) \times 2^{12-8}$

$[2^8]256 \Rightarrow (2^8 - x) \times 2^4$

$2^4 = 2^8 - x$

$x = 2^8 - 2^4 \Rightarrow 256 - 16$

$= 240$

Q.

A processor has 16 register (R0, R1, ..., R15) and 64 floating point registers (F0, F1, ... F63). It uses a 2-byte instruction format. There are four categories of instructions: Type-1 Tpye-2 Type-3 and Type-4. Type-1 category consists of four instructions, each with 3 integer register operands (3Rs) Type-2 category consists of eight instructions, each with 2 floating point register operands (2fs). Type-3 category consist of fourteen instructions, each with one integer register operand and one floating point register operand (1R + IF). Type-4 category consists of N instructions, each with a floating point register operand (FR). The maximum value of N is ______.

Integer

[GATE-2018 : 2 Marks]

16 Register $\Rightarrow$ $\textcircled{IR}$ Reg = 4 bit

64 Floating Register. FR = 6 bit

Instr^n = 2 Byte = 2×8 = $\underline{16 \text{ bit}}$

Type1

$\xleftrightarrow{16\text{ bit}}$

OP	IR	IR	IR
4bit	4	4	4

Primitive

Type1 : 4

Type2

$\xleftrightarrow{16\text{bit}}$

OP	FR	FR
4bit	6	6

Type2 : 8
Primitive.

Type3

$\xleftrightarrow{16\text{bit}}$

OP	IR	FR
6bit	4	6

Type3 : 14
Derived

Type4

$\xleftrightarrow{16\text{bit}}$

OP	FR
10bit	6

Type4 : ?
further Derived

① Type1

Total # operation $= 2^4 = \boxed{16}$

Free in type1 $= 16 - 4$

$= \boxed{12}$

② Type2

$4-4$

#operation $= 12 \times 2$

$= 12 \times 2^0$

$= \boxed{12}$

Given = 8

Free = $12 - 8 = \boxed{4}$

③ Type3

$6-4$

#operation $= 4 \times 2$

$\Rightarrow 4 \times 2^2 = 4 \times 4$

$= \boxed{16}$

Given = 14

Free = $16 - 14 = \boxed{2}$

Type4

$10-6$

#operation $= 2 \times 2$

$= 2 \times 2^4 = 2^5$

$= 32$ Ans

Type1

$\xleftarrow{\quad 16\text{bit} \quad}\!\!\!\rightarrow$

OP	IR	IR	IR
$\boxed{4\text{bit}}$	4	4	4

Primitive

Type1 : 4

Type2

$\xleftarrow{\quad 16\text{bit} \quad}\!\!\!\rightarrow$

OP	FR	FR
$\boxed{4\text{bit}}$	6	6

Type2: 8
Primitive.

Type3

$\xleftarrow{\quad 16\text{bit} \quad}\!\!\!\rightarrow$

OP	IR	FR
$\boxed{6\text{bit}}$	4	6

Type3: 14
Derived

Type4

$\xleftarrow{\quad 16\text{bit} \quad}\!\!\!\rightarrow$

OP	FR
$\boxed{10\text{bit}}$	6

Type4 : ?
further Derived

② Type1

$$\#\text{operation} = 8 \times 2^{4-4}$$
$$= 8 \times 2^0 = 8 \times 1$$
$$= \boxed{8}$$

Given = 4

$$\text{Free} = 8 - 4 = \boxed{4}$$

① Type2

$$\#\text{operation} = 2^4 = 16$$

Given = 8

$$\text{Free} = 16 - 8$$
$$= \boxed{8}$$

③ Type3

$$\#\text{operation} = 4 \times 2^{6-4}$$
$$= 4 \times 2^2$$
$$= 16$$

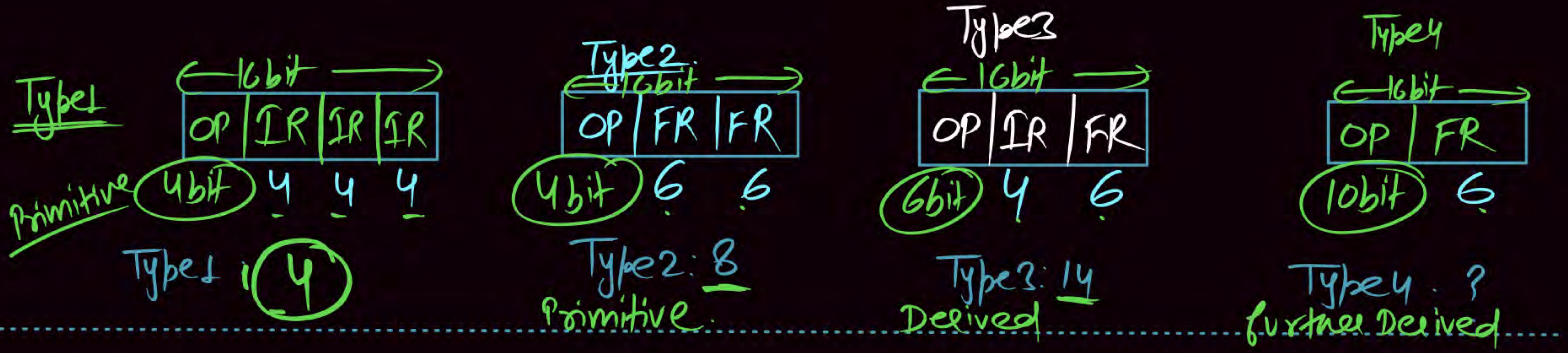
Given = 14

$$\text{Free} = 16 - 14 = \boxed{2}$$

④ Type 4

$$\#\text{operation} = 2 \times 2^{10-6}$$
$$= 2 \times 2^4$$
$$= 2^5$$
$$= 32$$

Ans



$$2^{16} = 4 \times 2^{12} + 8 \times 2^{12} + 14 \times 2^{10} + N \times 2^{6}$$

$$= 2^{6} \left[4 \times 2^{6} + 8 \times 2^{6} + 14 \times 2^{4} + N \right]$$

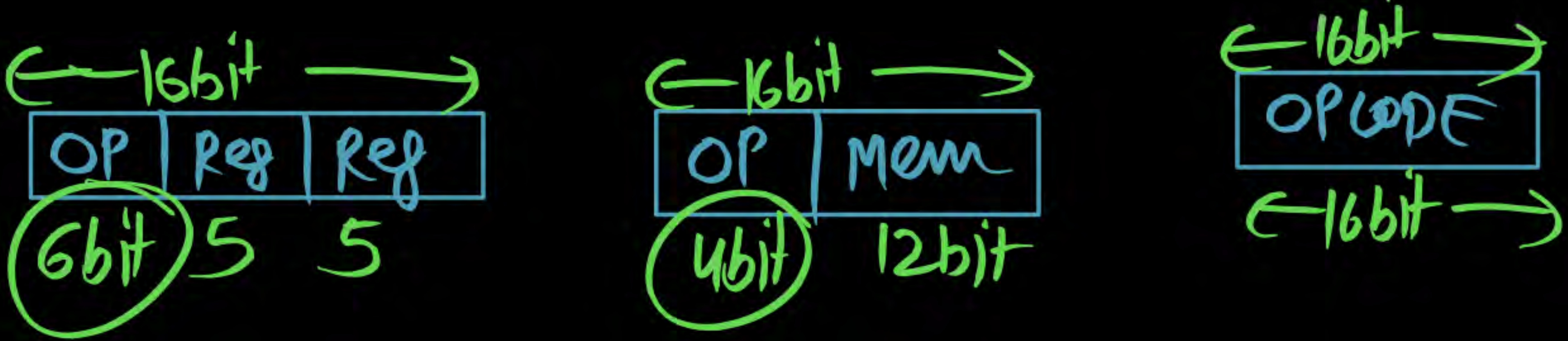
Q.

A processor has 64 registers and uses 16-bit instruction format. It has two types of instructions: I-type and R-type. Each I-type instruction contains an opcode, a register name, and a 4-bit immediate value. Each R-type instruction contains an opcode and two register names. If there are 8 distinct I-type opcodes, then the maximum number of distinct R-type opcodes is 14 Ans

[GATE-2020 : 2 Marks]

Q. Consider a 16 bit hypothetical processor which support 1 word long instruction. Processor has 30 registers and 4KB of memory size. If there exists '11' 2 address register reference instruction and '10' 1 address memory reference instruction. then how many '0' address instruction can be formulated/supported?

✓ $2^{16} = 11 \times 2^5 \times 2^5 + 10 \times 2^{12} + N.$

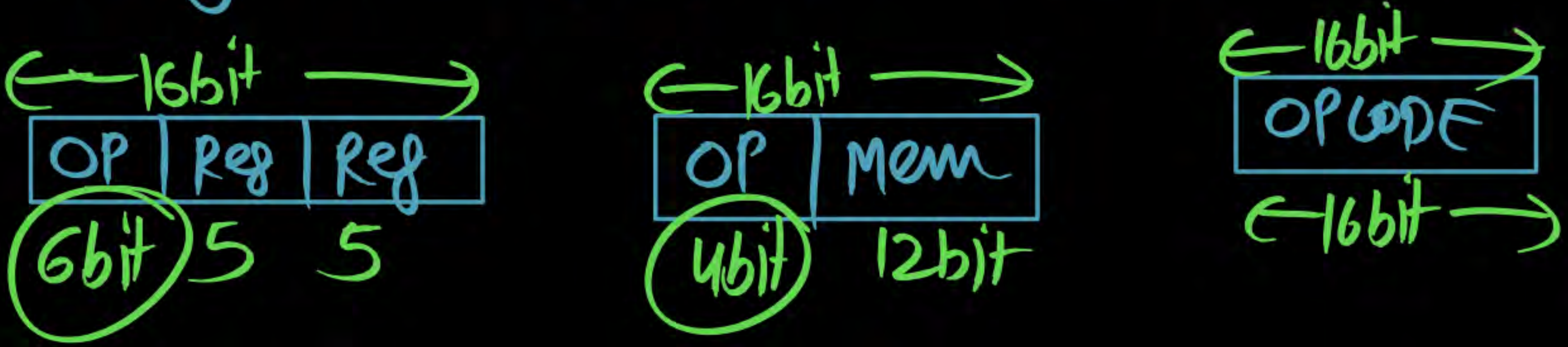


Q. Consider a 16 bit hypothetical processor which support 1 word long instruction. Processor has 30 registers and 4KB of memory size. If there exists '11' 2 address register reference instruction and '10' 1 address memory reference instruction. then how many '0' address instruction can be formulated/supported?

Instruction Size = 1 Word = 16 bit

30 Register ⇒ Reg AF = 5 bit

Memory = 4KB = 2^{12} Byte ⇒ AF = 12 bit



← 16bit →

OP	Reg	Reg
6bit	5bit	5bit

② Derived

← 16bit →

OP	mem
4bit	12bit

① Primitive

← 16bit →

OPCODE
← 16bit →

③ further Derived.

---

① Primitive Total # operation $= 2^4 = 16$

Given mem Ref $= 10$

Free $= 16 - 10 = \boxed{6}$

---

② Derived : Total # operation $= 6 \times 2^{6-4} \Rightarrow 6 \times 2^2 = \boxed{24}$

Given $= 11$

Free opcode $= 24 - 11 = \boxed{13}$

$OAF = 13 \times 2^{16-6} \Rightarrow 13 \times 2^{10} = \boxed{13K}$ Ans

Q. Consider a processor with 11 bit instruction, the size of address fields is 4bits. The computer uses expanding opcode technique and has '5' two(2) address instruction and '32' one(1) address instruction. Then the number of Zero address instruction it can support is__________

[GATE : 2 Marks]

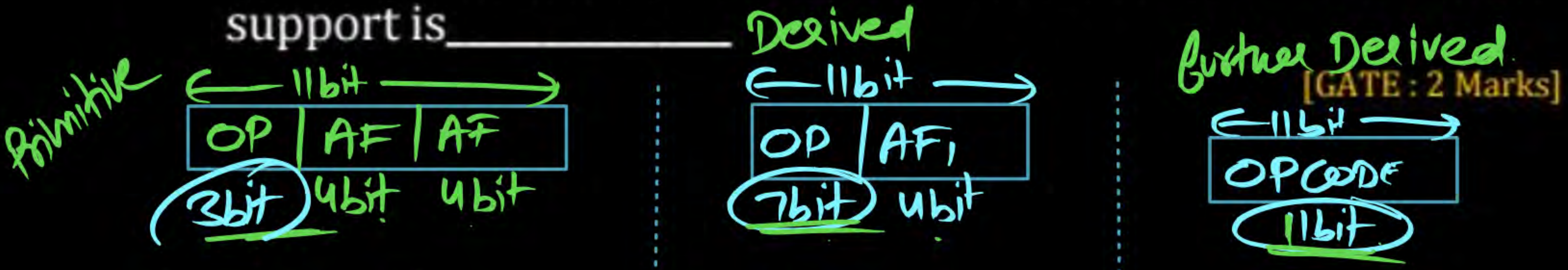
#operation = $2^3 = 8$.
Given = 5
Free = $8 - 5 = 3$.

#operation = $3 \times 2^{7-3}$
$\Rightarrow 3 \times 2^4$
$= 48$.
Given = 32
Free = $48 - 32$
$= \boxed{16}$

#operation = $16 \times 2^{11-7}$
$= 16 \times 2^4$
$= 16 \times 16$
$= 256$. Ans

Q. Consider a processor with 11 bit instruction, the size of address fields is 4bits. The computer uses expanding opcode technique and has '5' two(2) address instruction and '32' one(1) address instruction. Then the number of Zero address instruction it can support is__________

[GATE : 2 Marks]



$$2^{11} = 5 \times 2^{4} \times 2^{4} + 32 \times 2^{4} + 'X'$$

$$2^{11} = \Rightarrow 5 \times 2^{8} + 2^{9} + X$$

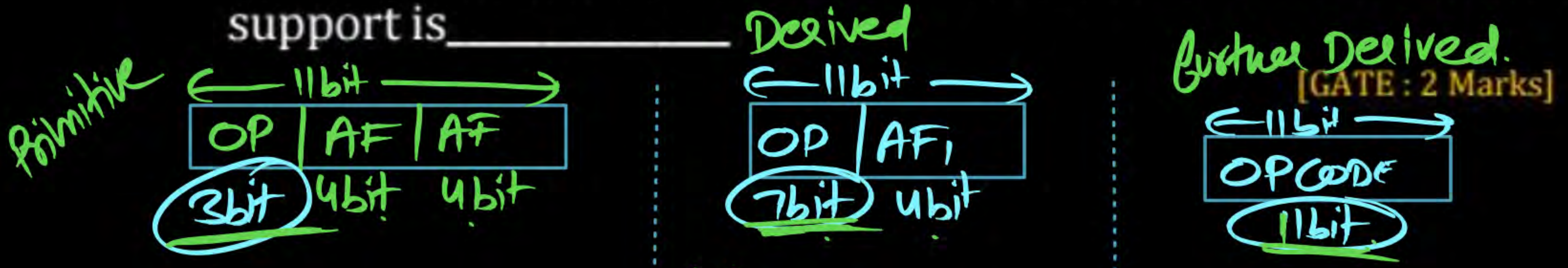
$$2^{11} = 2^{8}[5+2] + X$$

$$\boxed{X = 256}$$ Ans

Q.

Consider a processor with 11 bit instruction, the size of address fields is 4bits. The computer uses expanding opcode technique and has '5' two(2) address instruction and '32' one(1) address instruction. Then the number of Zero address instruction it can support is__________

[GATE : 2 Marks]



$$2^{11} = 5 \times 2^4 \times 2^4 + 32 \times 2^4 + X$$

$$2^{11} = 5 \times 2^8 + 2 \times 2^4 \times 2^4 + X$$

$$2^{11} = 5 \times 2^8 + 2 \times 2^8 + X$$

$$2^{11} = 2^8 [5+2] + X$$

$$X = 2^{11} - 2^8 \times 7 \quad [256 \times 7]$$

$$= 2048 - 1792$$

$$= 256 \text{ Ans}$$

Addressing Modes [AM].

Addressing is a technique used to Calculate the Effective Address [EA]

OR

Addressing Mode Show the Way Where the Required Object is Present

OR

Addressing Mode Show the Way. How to get Operand.

Addressing Modes

Effective Address [EA] is the actual address of the Object.

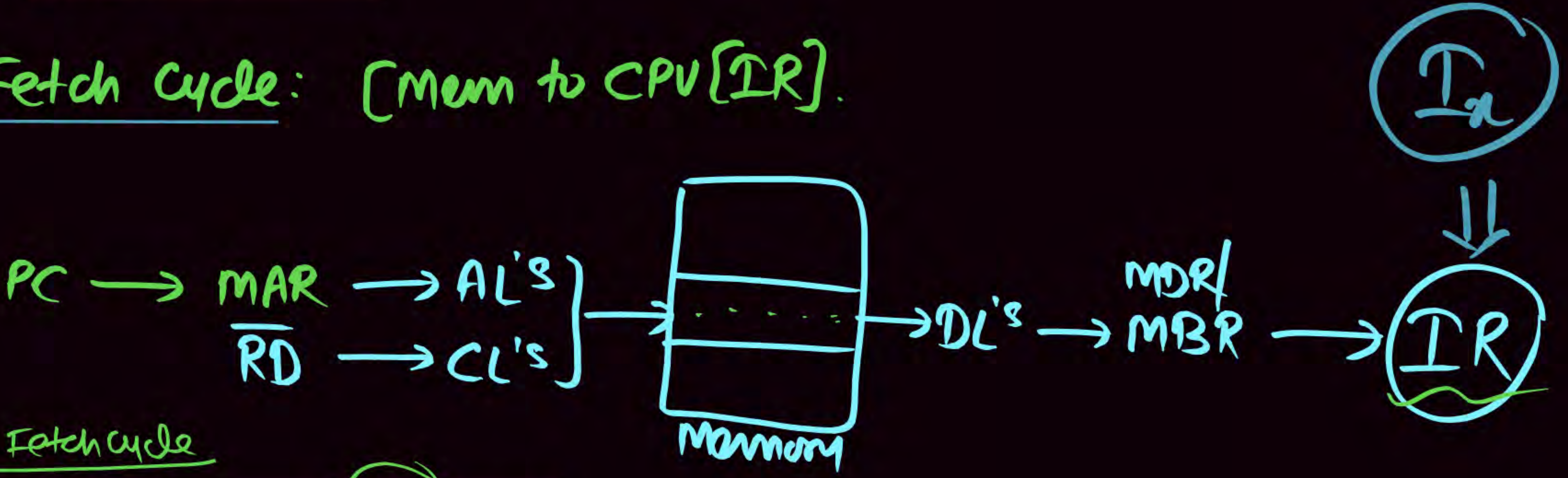
Object May be Instruction (or) DATA.

The output of Addressing Mode is Effective Address.

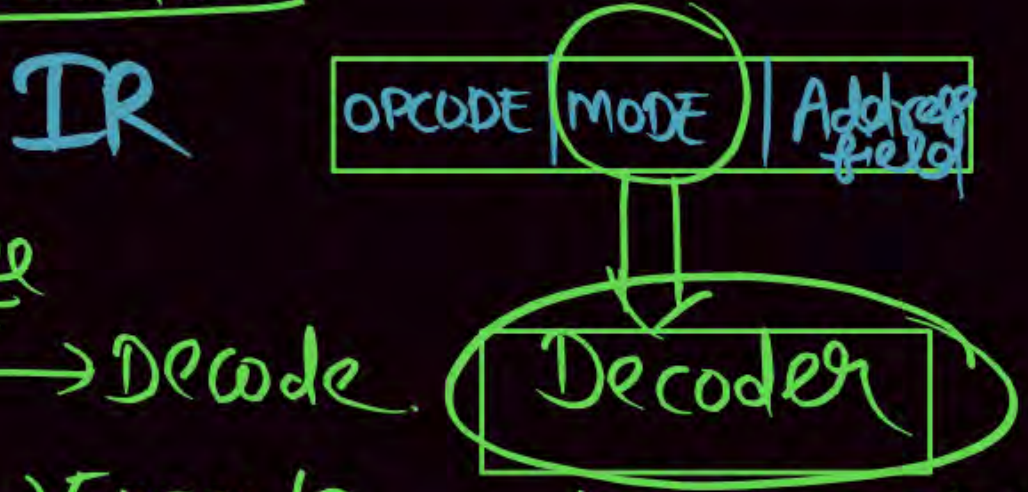
$$\begin{tikzcd}
 & \text{Memory Address} \\
\underline{\text{Address}} : \arrow[ur] \arrow[r] \arrow[dr] & \text{Register Address} \\
 & \text{Immediate field [Constant]}
\end{tikzcd}$$

WHEN AM ?

① Fetch cycle: [Mem to CPU[IR].



Fetch cycle

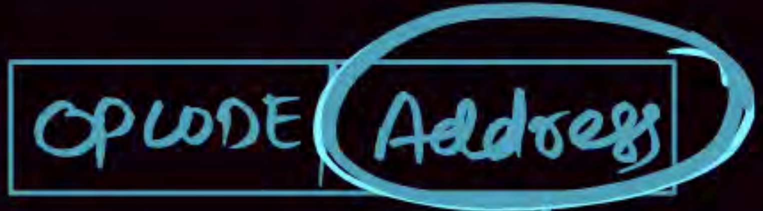


Execute cycle
- → Decode.
- → Execute.

'OAC' : Operand Address Calculation
'OF' : operand Fetch
DP & RS: Data Processing & Result Storage.

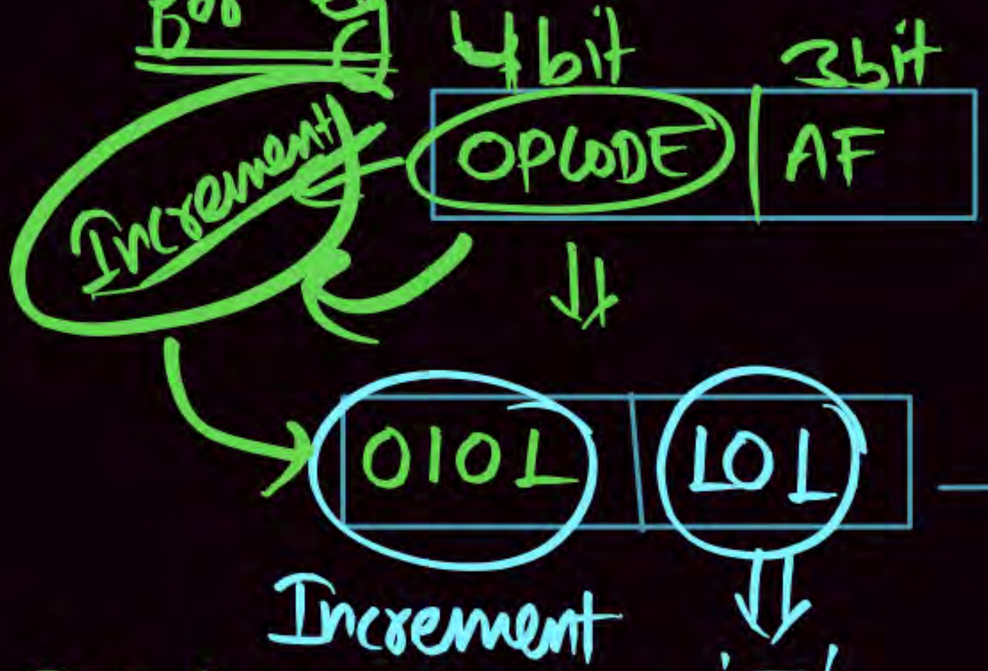
WHY AM ?

How to get operand.



- → Register [Direct or Indirect)
- → memory [Direct or Indirect]
- → Constant [Value]

for eg



- → Constant (Value) = '5'
- → Register (101) ⇒ R_5.
- → memory Direct $M[5]$
- → Memory Indirect $M[[5]]$.

So to Remove the Confusion AM is Used.

Addressing Modes

OPCODE	MODE field	Address field

MODE field →
- Immediate (constant)
- Register (Direct/Indirect)
- Memory (Direct/Indirect)

MODE field : Helps you How to get operand

OR

How to Use this Address field
- Immediate
- Register Address
- Memory Address

Addressing Modes

Addressing Mode Show the way Where the Required Object is Present. OR Location of Required object.

① Data Centric AM [Sequential Control Flow AM]

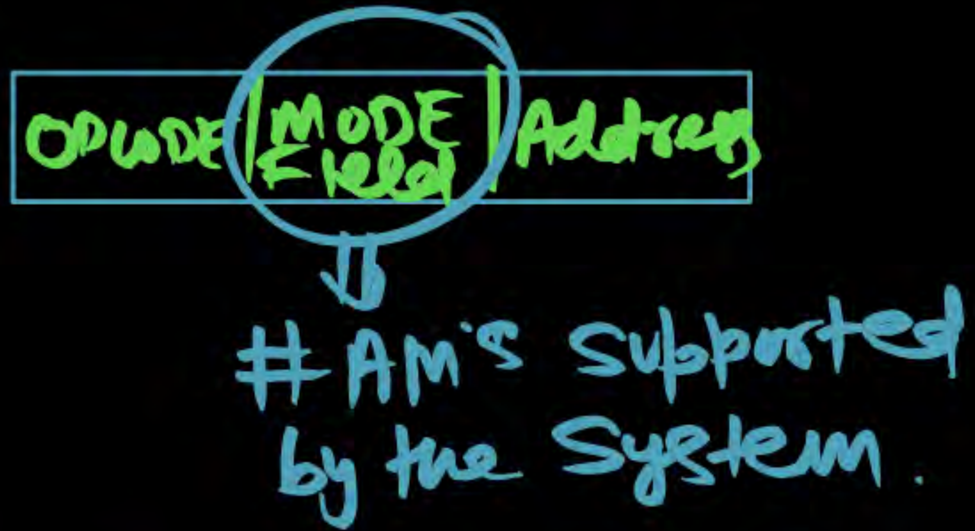
↳ Focus on 'DATA'

② Instruction Centric AM [Transfer of Control Flow AM].

↳ Focus on 'Instruction'.

Addressing Modes

AM in the Instruction is Implemented with the Help of 'MODE Field'



Note In the Computer Data Present in either Register or Memory. Based on that there are various type of Addressing Mode.

Addressing Modes

1. Immediate AM.
2. Direct/Abuslute AM.
3. Memory Indirect AM.
4. Register Direct AM
5. Registe Indirect AM
6. Pc-Relative AM.
7. Based Registe AM.
8. Indexed Registe AM.
9. Implied | Implict AM.
10. Auto Decrment AM.
11. Auto Increment AM.

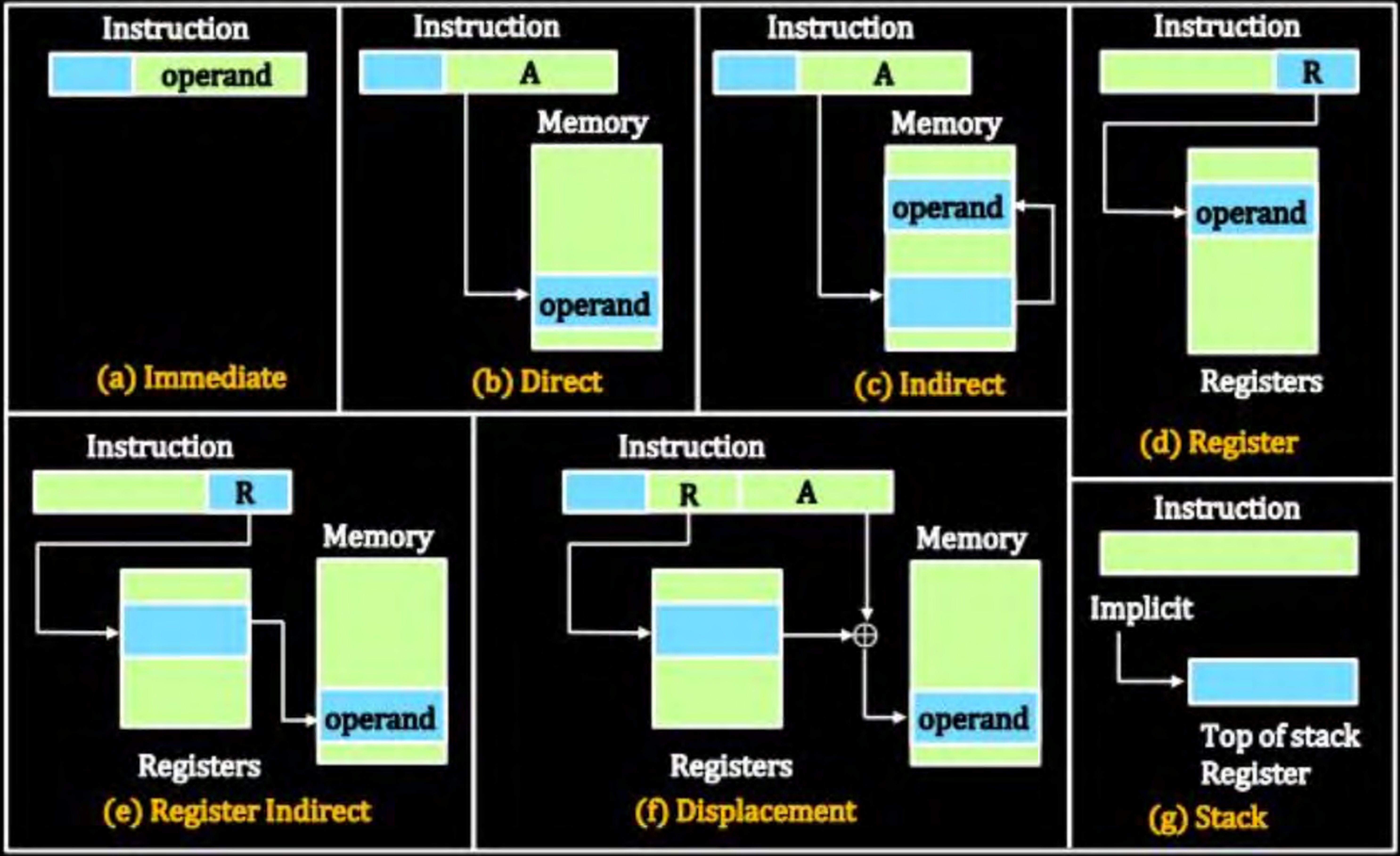
Addressing Modes

Symbol's of AM.

Symbol's	AM.
I or #	Immediate AM
[ ]	Direct AM.
[( )] or @	Indirect AM.
Reg Name	Register AM.
Index Reg Name	Indexed Reg. AM.

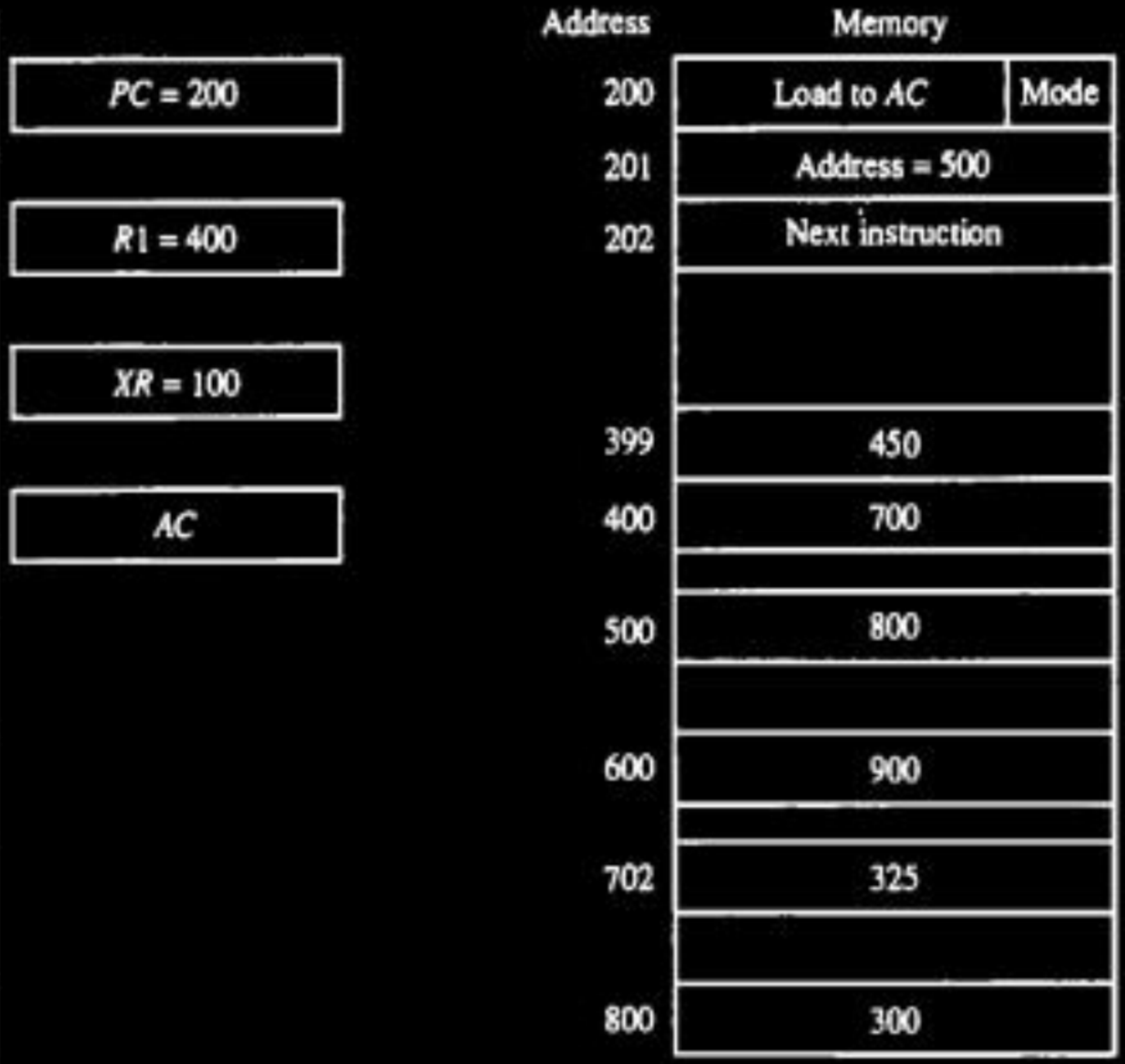
Addressing Modes

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



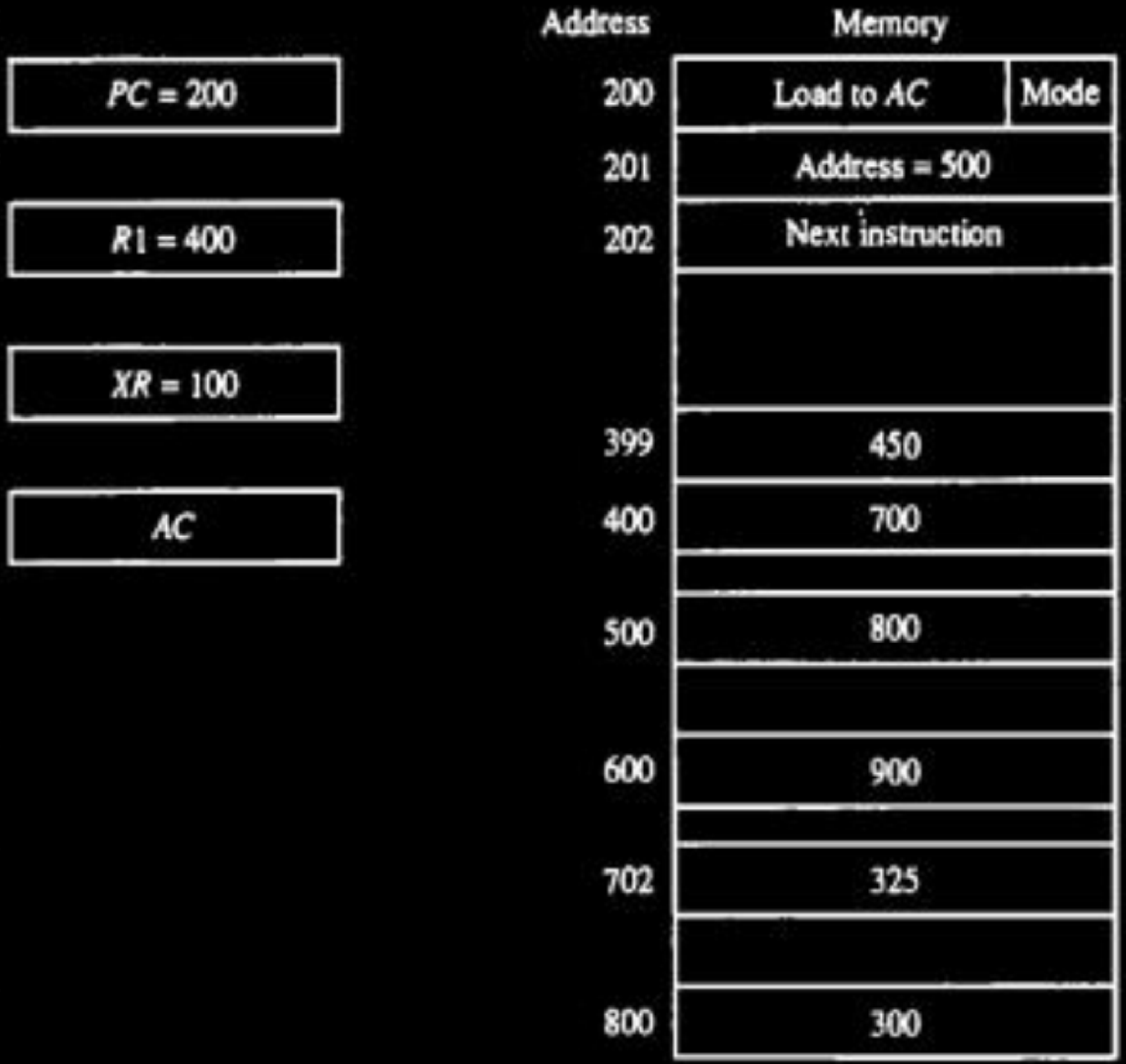
(a) Immediate (b) Direct (c) Indirect (d) Register (e) Register Indirect (f) Displacement (g) Stack

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



Numerical example for addressing modes.

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



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

THANK YOU!