

COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and
Addressing Modes

Addressing Modes

Lecture_05



Vijay Agarwal sir



**TOPICS
TO BE
COVERED**

- o1 Expand Opcode Technique**
- o2 Addressing Modes**

Instruction Set Arch.

- ① Stack -CPU
- ② AC -CPU
- ③ Reg -CPU.

Expand opcode Technique.

Primitive \Rightarrow lowest opcode bit

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.

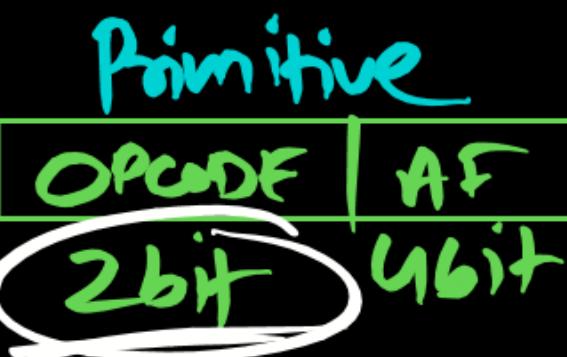
Step 3: Identify the free opcode after allocating the existed instruction

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

Free opcode \times 2 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?

P
W

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

Given in Question = 2

$$\# \text{Free OPCODE} = 4 - 2 = 2$$



Total
 $\# \text{ operation} = \text{free OPCODE} \times 2^{\text{bit in OPCODE}}$

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

$$= 2 \times 2^4 = 32$$

Increment
 2^{bit in OPCODE}

$$= 2 \times 2^4 = 32$$

Q.2

P
W

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

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
- C 240

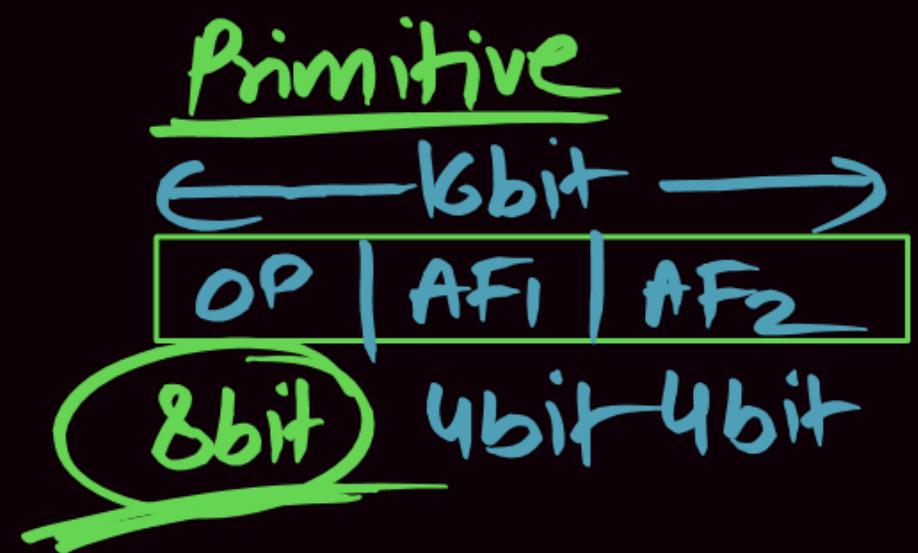
- B 192
- D 248

Instr Size = 16bit

15 Reg. \Rightarrow Reg AF = 4bit

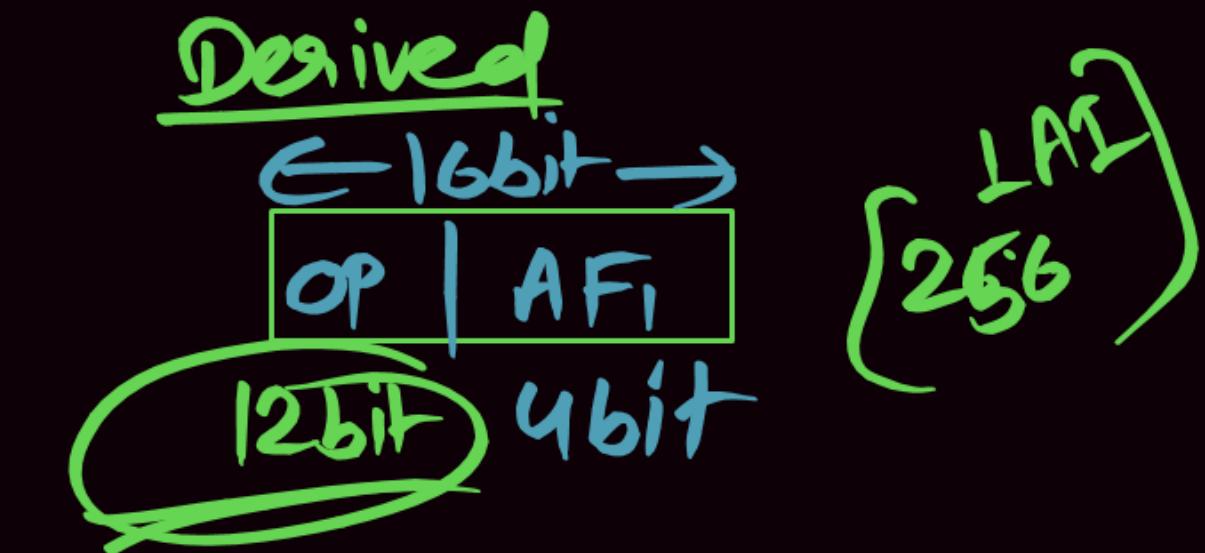
256 1AI & then asking Total # 2AI?

P
W



Total # operation = $2^8 = 256$.
in the Primitive

Let's Assume 2AF Instⁿ = χ .
#Free opcode = $(256 - \chi)$
After Allocating
2AI/2AF



Total # operation in LAI | Derived Instⁿ = Free opode $\times 2^{12-8}$
Increment bit in opcode

$$256 \Rightarrow (256 - \chi) \times 2^4$$

~~$$256 \Rightarrow (256 - \chi) \times 2^4$$~~

$$2^4 = 256 - \chi$$

$$\chi = 256 - 16 = \underline{\underline{240}}$$

Solution(c): 240

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



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

So total number of 2 address instruction = $2^8 = \underline{\underline{256}}$

Let 'x' 2 address instruction used

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

1 Address field



In Question.

Given LAT = 256

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

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

$$2^4 = 2^8 - x$$

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

$$= 240$$

Q.

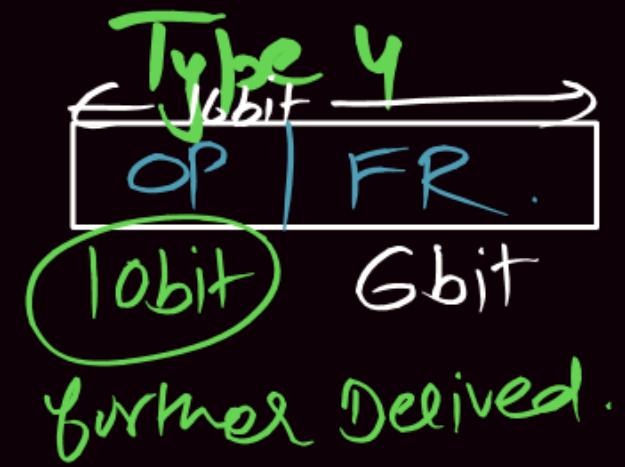
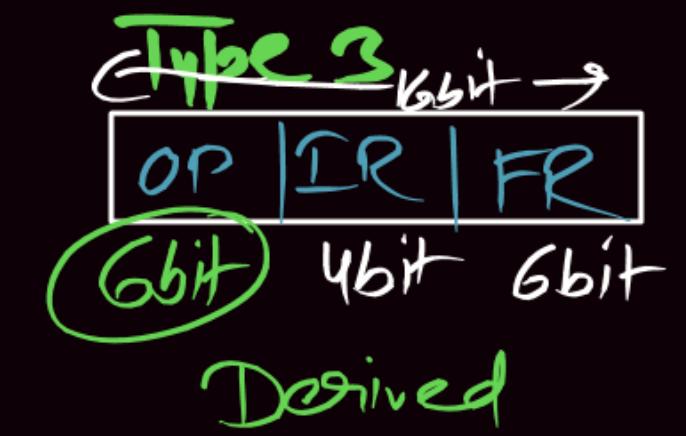
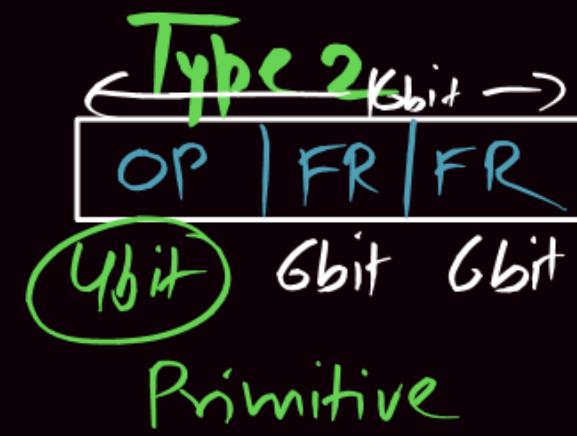
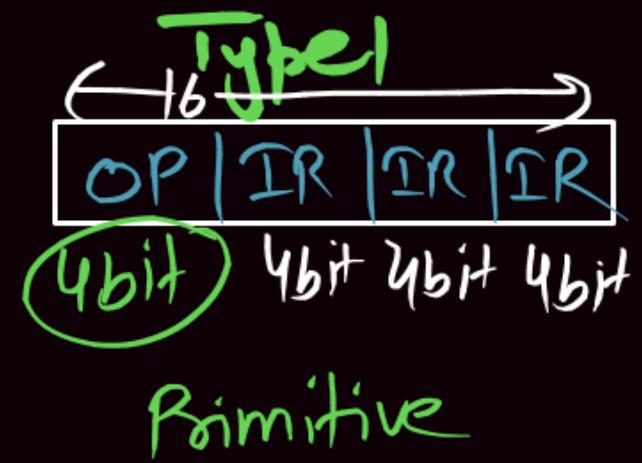
A processor has 16 register (R0, R1, ..., R15) and 64 floating point registers (F0, F1, F63). It uses a 2-byte instruction format. There are four categories of instructions: Type-1 Tpye-2 Type-3 and Type-4. Type-1 category consists of four instructions, each with 3 integer register operands (3Rs) Type-2 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 32 Aug.

$$16 \text{TR} \Rightarrow IR = 4 \text{bit}$$

$$64 \text{FR} \Rightarrow FR = 6 \text{bit}$$

Inst size = 2Byte [GATE-2018 : 2 Marks]

= 16bit



Type L: Total # operation in Type L = $2^4 = 16$

Given (Type L) = 4

free opcode After Allocating type L = $16 - 4 = 12$

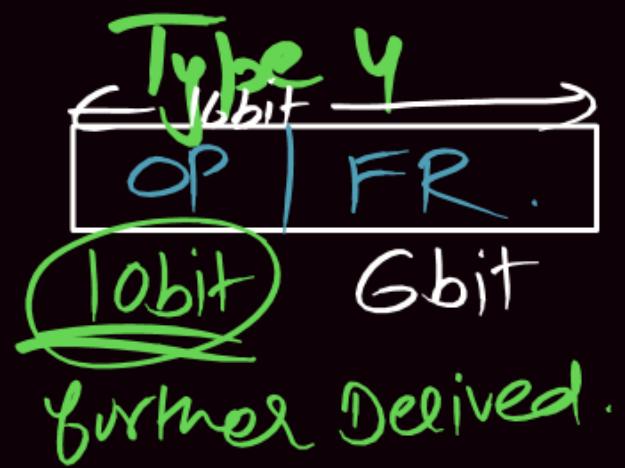
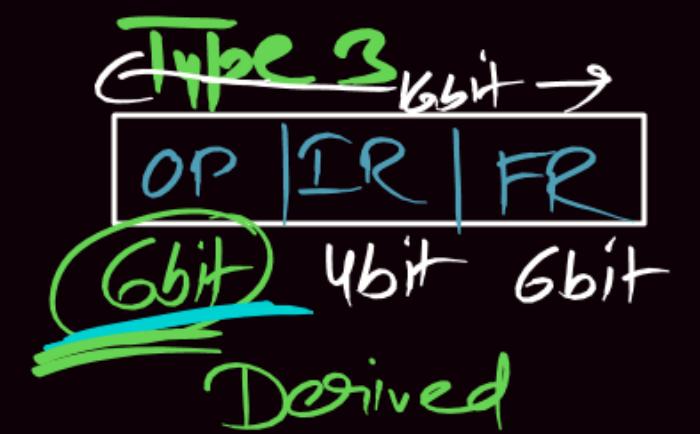
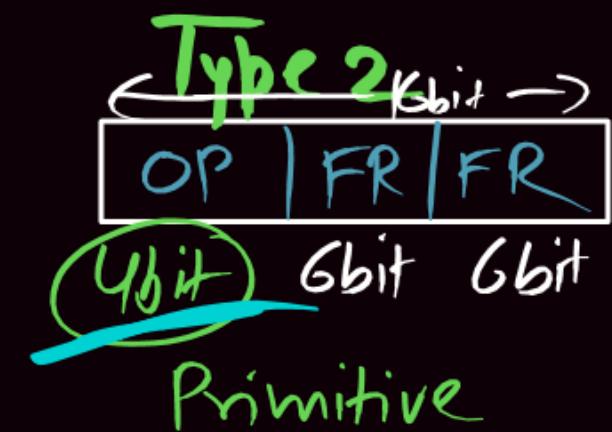
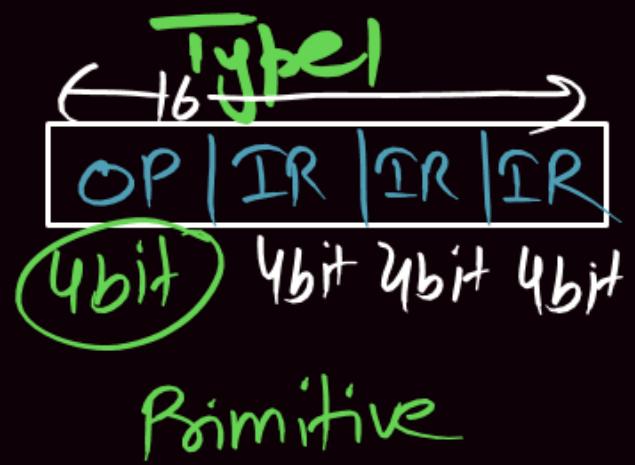
Type 2

Total # operation in Type 2 = free opcodes × 2 in opcode Increment bit $\neq 12 \times 2$ 4-4 $\Rightarrow 12 \times 2^0 = 12 \times 1$

12 operation.

Given Type 2 = 8

free opcode After Allocating type 2 = $12 - 8 = 4$

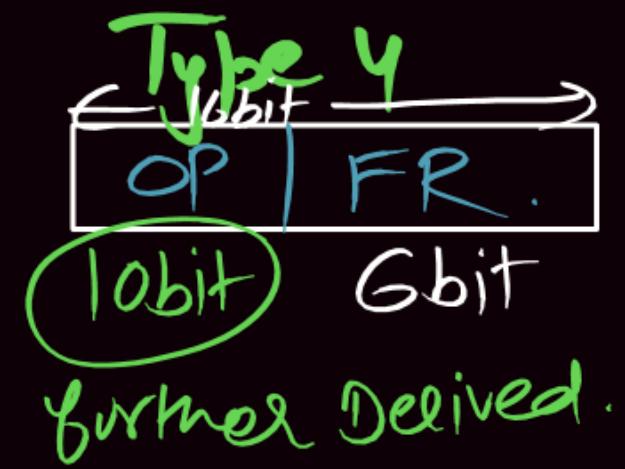
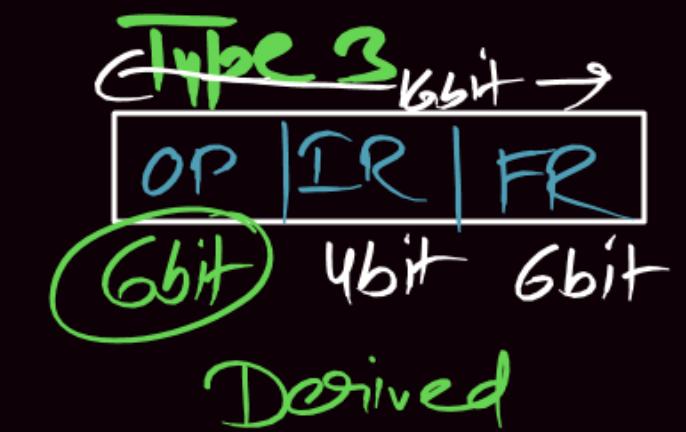
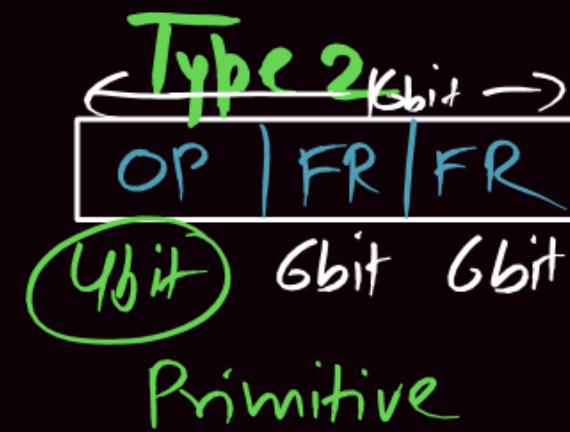
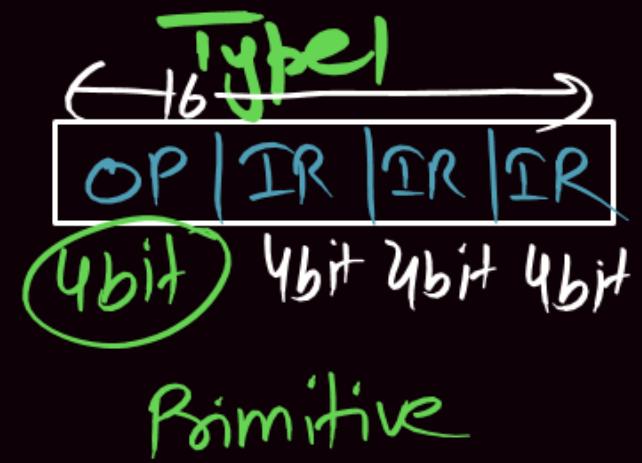


Type 3 Total # operation in Type 3 = $\frac{\text{Free Opcodes}}{4} \times 2^{\text{Incond bit in op code}} = 4 \times 2^4 = 16$ operation.

Free opcode After Type 3 = $16 - 14 = 2$

Type 4 Total # operation in Type 4 = $\frac{\text{Free Opcodes}}{2} \times 2^{\text{Incond bit in op code}}$

$$\Rightarrow 2 \times 2^{10-6} \Rightarrow 2 \times 2^4 \times 2 \times 16 = 32 \text{ Avg}$$



Type 2

Total # operation in Type 2 = $2^4 = 16$ operation

Given Type 2 = 8

#Free opcode After type 2 = $16 - 8 = 8$.

Type 1: Total # operation in Type 1

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

$$8 \times 2^{4-4} = 8 \times 2^0 = 8 \times 1 = 8$$

= 8 operation

Type 1 Given = 4

#Free opcode After type 1 = $8 - 4 = 4$

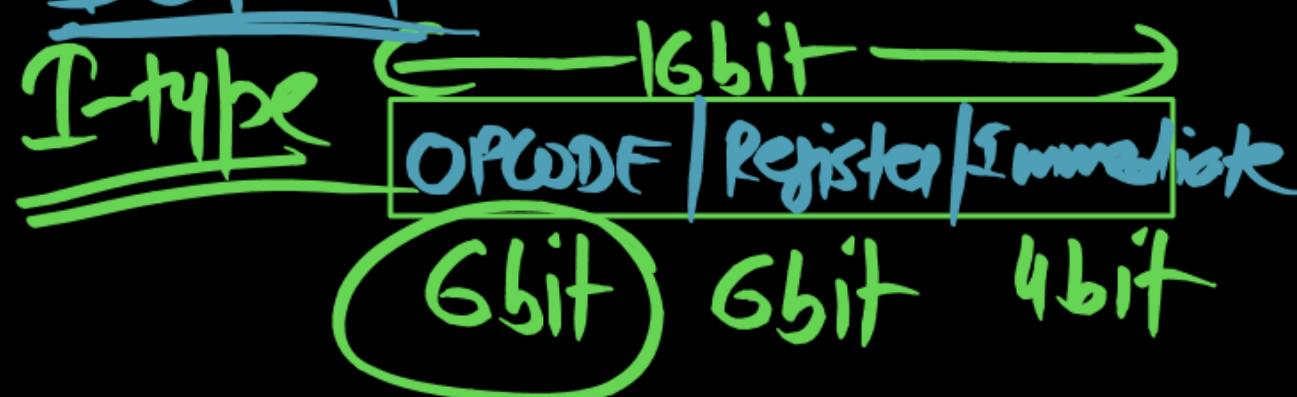
Q.

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.

64 Registers \Rightarrow Reg AF = 6bit

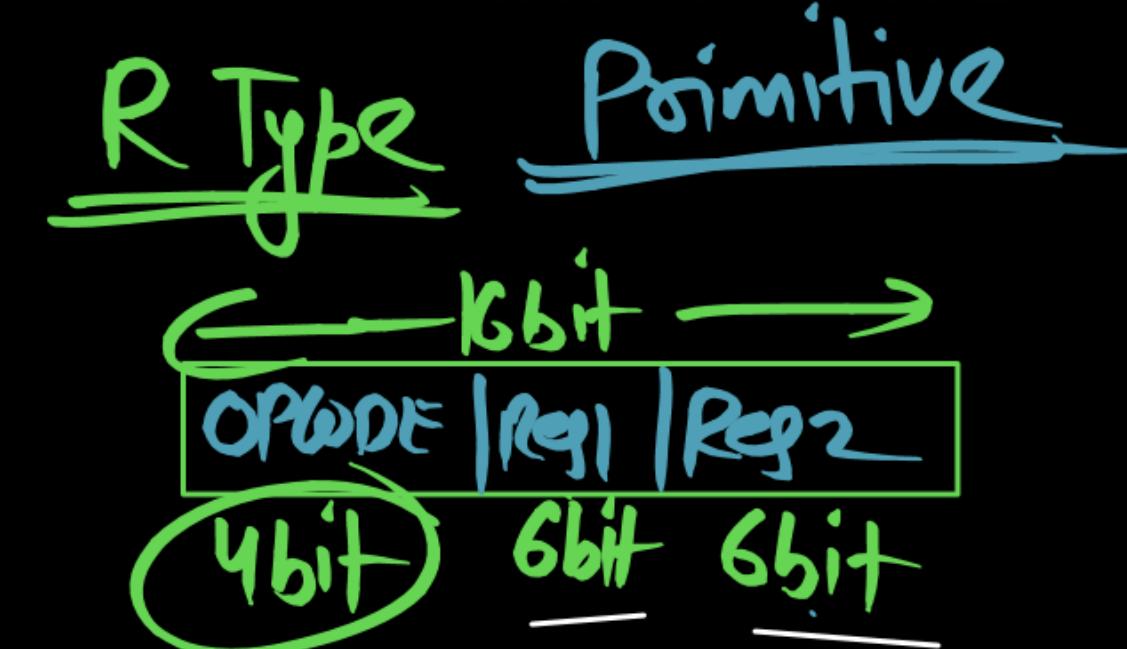
Instn Size = 16bit

Derived



Ans(14).

[GATE-2020 : 2 Marks]



Primitive - R Type

R type:

OP	Rg1	Rg2
4bit	6	6

Total # operation in R Type = $2^4 = 16$ operation.

Let's Assume R Type = ' x '.

Free opcode After Allocating R type = $16 - x$.

I-type Derived

Freeopcode $\times 2^{6-4}$ Increment bit in Opcode.

$$\text{Total # operation in I-type} = (16-x) \times 2^{6-4}$$

$$16-x=2 \quad \swarrow 8 = (16-x) \times 4$$

$$x=16-2=14$$

$$8 = 64 - 4x$$

$$4x = 56$$

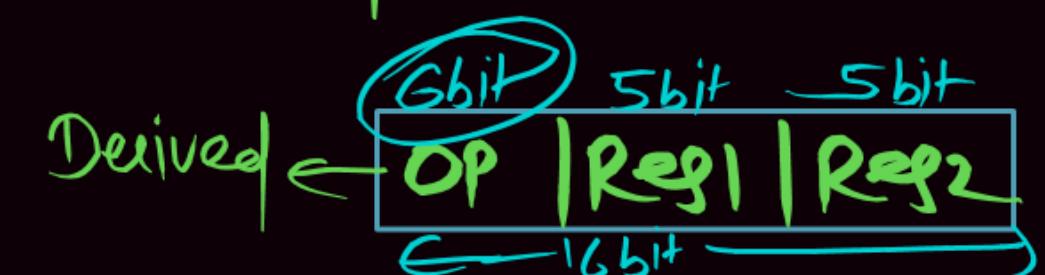
$$x = \frac{56}{4} = 14 \text{ Ans}$$

OP	Rg1	Rmm
6bit	6	4

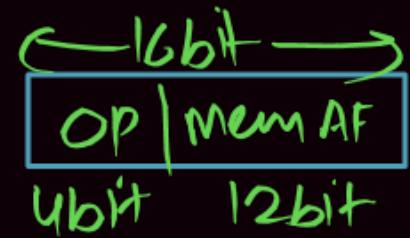
Q) Consider a ~~16bit~~ Hypoth. Processor which Support 1 word Long Instruction, CPU contain 30 Register. UKB Memory size.
 If there exists LL 2 Address Register Reference Instruction & 10 1 Address Memory Reference Instruction then How Many 0 Address Instn Can be formulated | Supported ?

Soln 1Word Instn \Rightarrow Instn Size = 16bit

(2) 30Reg \Rightarrow Reg AF = 5bit
 4KKB Memory MemAF = 12bit



Primitive



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

Given (10)

$$\#\text{Free opcode} = 16 - 10 = 6$$

Derived

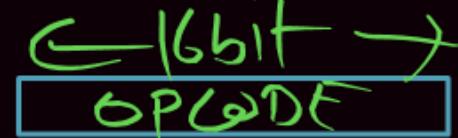


$$\begin{aligned} \text{Total \# operation} &= 6 \times 2^5 \Rightarrow 6 \times 2^6 \Rightarrow 6 \times 4 \\ &= 24 \end{aligned}$$

Given (2AF) = 11

$$\#\text{Free opcode After 2AF} = 24 - 11 = 13$$

Further (More)
Derived



$$\begin{aligned} \text{Total \# operation} &= 13 \times 2^{16-6} \\ &= 13 \times 2^{10} \end{aligned}$$

If NAT
= 13×1024
= 13312 Ans

Ans
13K

- MCQ
- (A) 10K
 - (B) 12K
 - (C) 13K
 - (D) 15K

Addressing Modes

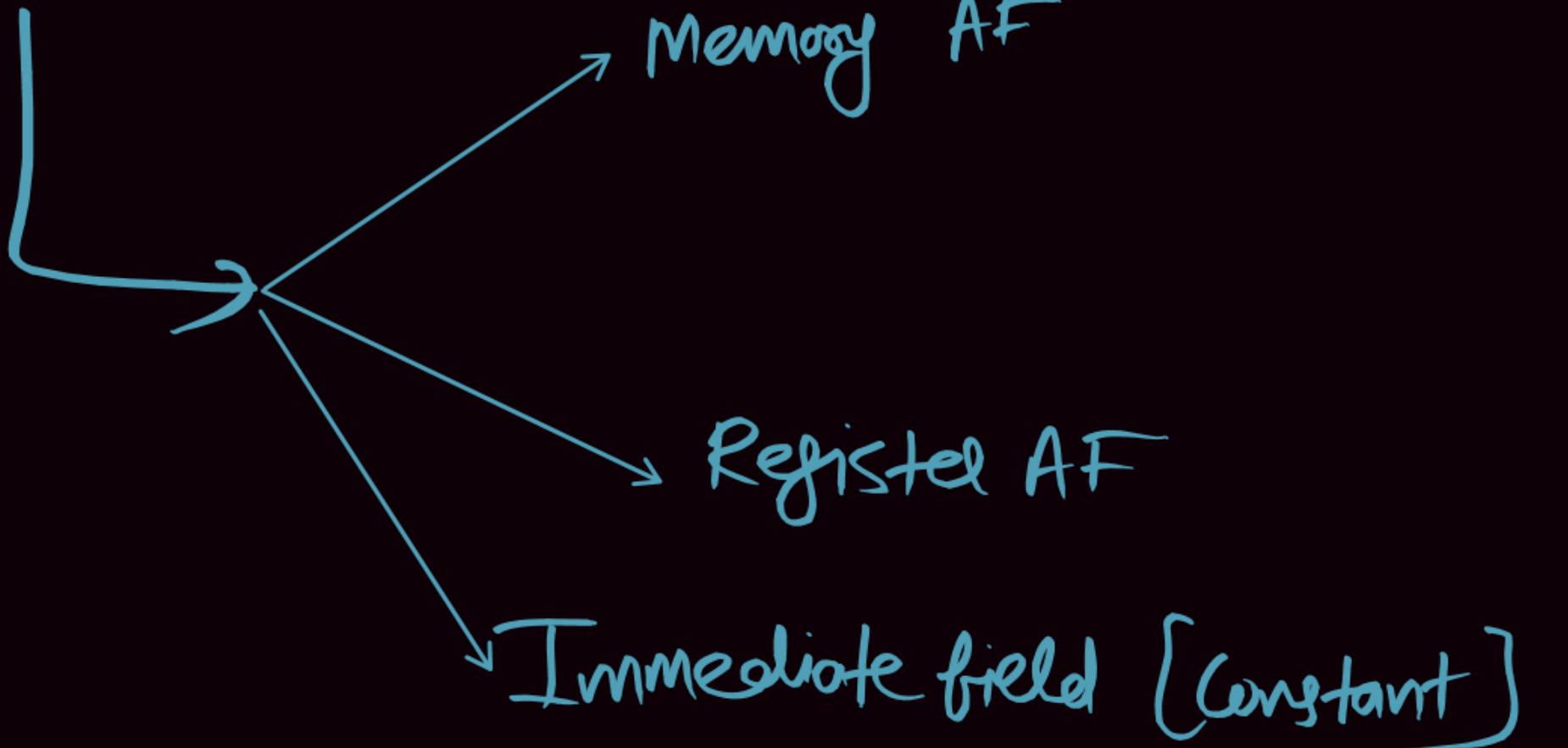
- Addressing Mode is a technique Used to Calculate the Effective Addresses.
(OR)
- Addressing Mode Show the Way where the Required Object is Present
- Address Mode Show the Way How to get operand
(OR)

Addressing Modes

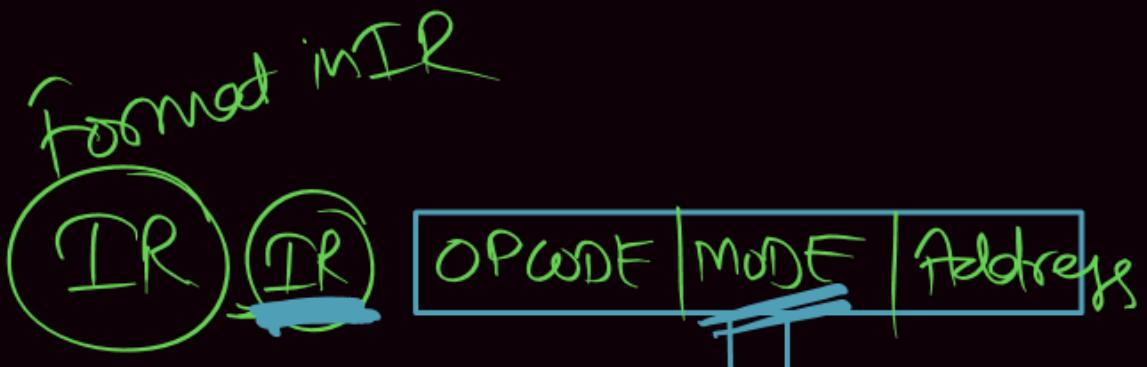
Effective Address is a Address where the Required Object is Present.

- Object May be INSTR or DATA.
- The output of AN is the Effective Address.

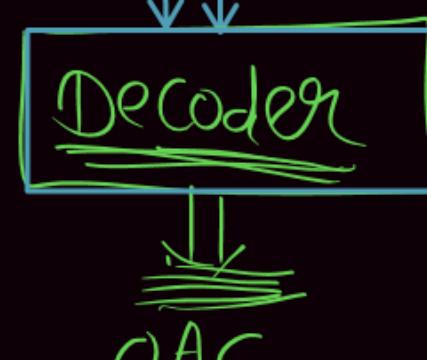
Address



Instruction Cycle



Execution
Cycle
(Decode)



OAC
OF
D.P & RS.

Addressing Modes

WHY AM ??



So How to get the operand.

(g) (g)



110L

101
5

Incsemment

So to Remove the
Confusion AM is Required.

Value (Constant)

OR
Register (Direct & Indirect)

OR
Memory Address (Direct, Indirect)

Register = (101) $\Rightarrow R_5 \Rightarrow$ Increment R_5

OR
Immediate = '5' Constant

OR
Memory Direct $M[5]$

OR
Memory Indirect $M[R_5]$.

Addressing Modes



→ Immediate (value)
→ (Register)
→ Memory AF

MODE bit (MODE Field) : Helps you How to get the operand.
OR

How to use this {
Addressess Part } {
of Immediate
Register AF
Memory AF etc }

Addressing Modes

Addressers Mode Show the way where the Required object Present.

① Location of the Required Object

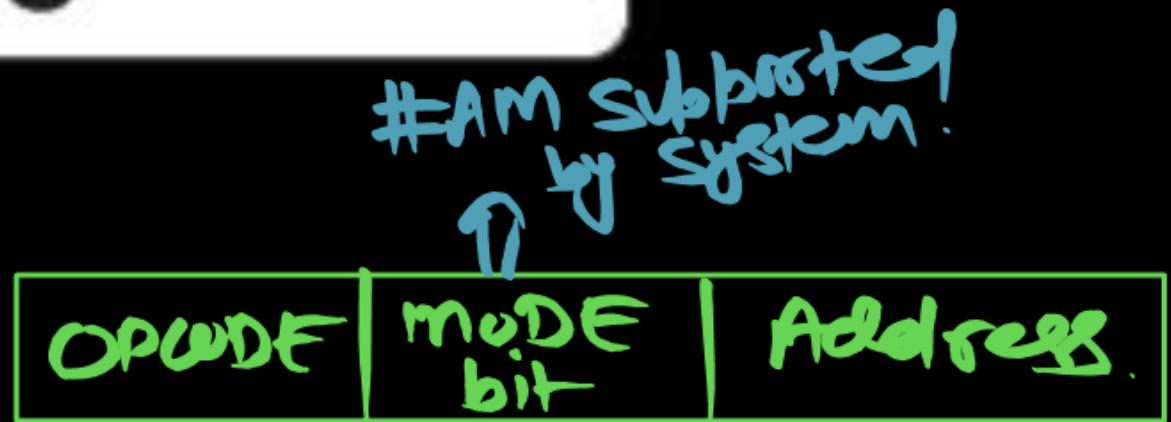
~~①~~ Data Centric AM [sequential Control Flow AM]
↳ 'FOCUS ON DATA'

② Instruction Centric AM [Transfer of Control Flow AM]

↳ FOCUS ON 'INSTRUCTION'.

AM in the Instruction is implemented using Mode field.

Addressing Modes



Note

In the Computer Data is Present either

in Memory OR in Registers.

- Based on that there are Various Type of AM.

Addressing Modes (AM).

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

Addressing Modes

Symbols for AM.

Immediate AM - # | I

Memory Direct AM - []

Memory Indirect AM - @ | ()

Register AM - Reg Name.

Index Reg AM - Index Reg Name.

Addressing Modes

- Immediate AM
- Direct | Absolute AM | memory Direct AM.
memory
- Indirect AM
- Register AM
- Register Indirect AM
- Displacement AM
- Stack AM.

OPERAND = 'DATA'

Addressing Modes

- ① Immediate AM : In this AM operand are present (placed) in the Instruction Itself.



↓

'DATA'

(OR)

Operands are present in the Address field of the Instruction.

MVI R₁, 1000

R₁ ← 1000

③ ADD R₂, #500.

ADDI R₂, 500;

R₂ ← R₂ + 500

**THANK
YOU!**

