COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and Addressing Modes

Expand Opcode Technique

Lecture_04

Vijay Agarwal sir

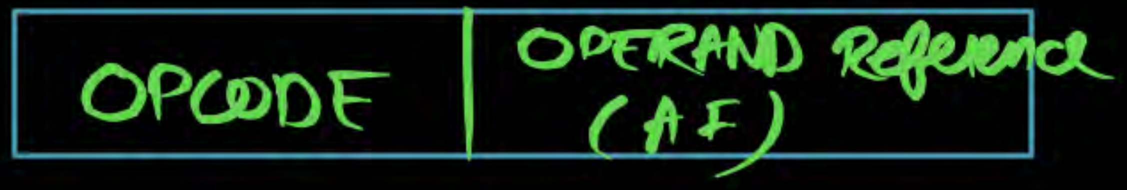
01 Expand Opcode Technique

02 Addressing Modes

Instruction Format

Opcode ⇒ operational Code

$$\begin{tikzcd} \text{Opcode} \arrow[d] \\ \text{Type of operation} \end{tikzcd}$$



$$\begin{tikzcd} & \text{memory} \\ \cdot \arrow[ur] \arrow[dr] & \\ & \text{Register} \end{tikzcd}$$

Based on the ALU operand

① STACK Based org

② Accumlator Based org.

③ General Register Org.

Instruction Set Architecture

CUP Organization is classified into 3 types based on the availability of ALU Operand (Data) (AF: Address field or AI: Address Instruction)

1. Stack-CPU **[0AF]**
2. Accumulator-CPU **[1AF]**
3. General Register organization
   1. Reg-Memory reference CPU **[2AF]**
   2. Reg-Reg reference CPU **[3AF]**

Stack Organization

Q. Consider a 32 bit Hypothetical Processor which use STACK-CPU. Which support 1 Word opcode and 24 bit address following statement is executed on a STACK-CPU (Stack is Initially Empty)

$$\boxed{X = (A + B) \times (C + D)}$$

Stack Organization

$X = (A + B) \times (C + D)$

I_1:	PUSH	A	$TOS \leftarrow A$
I_2:	PUSH	B	$TOS \leftarrow B$
I_3:	ADD		$TOS \leftarrow (A + B)$
I_4:	PUSH	C	$TOS \leftarrow C$
I_5:	PUSH	D	$TOS \leftarrow D$
I_6:	ADD		$TOS \leftarrow (C + D)$
I_7:	MUL		$TOS \leftarrow (C + D) \times (A + B)$
I_8:	POP	X	$M[X] \leftarrow TOS$

8 Machine Instruction Required (Stack-CPU)

Single Accumulator Organization

ALU Operation:	Destination	Source 1	Source 2
	AC	AC	Reg | mem

eg (A+B)

LOAD A; $AC \leftarrow M[A]$

ADD B; $AC \leftarrow AC + M[B]$

Single Accumulator Organization

$$\boxed{X = (A + B) \times (C + D)}$$

I_1:	LOAD	A	$AC \leftarrow M[A]$
I_2:	ADD	B	$AC \leftarrow AC + M[B]$
I_3:	STORE	T	$M[T] \leftarrow AC$
I_4:	LOAD	C	$AC \leftarrow M[C]$
I_5:	ADD	D	$AC \leftarrow AC + M[D]$
I_6:	MUL	T	$AC \leftarrow AC \times M[T]$
I_7:	STORE	X	$M[x] \leftarrow AC$

7 Machine Instruction Required (AC-CPU)

General Register Organization

Reg– Mem Ref.

$X = (A + B) \times (C + D)$

I_1:	MOV	R1, A	$R1 \leftarrow M[A]$
I_2:	ADD	R1, B	$R1 \leftarrow R1 + M[B]$
I_3:	MOV	R2, C	$R2 \leftarrow M[C]$
I_4:	ADD	R2, D	$R2 \leftarrow R2 + M[D]$
I_5:	MUL	R1, R2	$R1 \leftarrow R1 \times R2$
I_6:	MOV	X, R1	$M[X] \leftarrow R1$

6 Machine Instruction Required (Reg-CPU)

RISC Instructions

Reg- Reg Ref.

$X = (A + B) \times (C + D)$

LOAD	R1, A	$R1 \leftarrow M[A]$
LOAD	R2, B	$R2 \leftarrow M[B]$
LOAD	R3, C	$R3 \leftarrow M[C]$
LOAD	R4, D	$R4 \leftarrow M[D]$
ADD	R1, R1, R2	$R1 \leftarrow R1 + R2$
ADD	R3, R3, R2	$R3 \leftarrow R3 + R4$
MUL	R1, R1, R3	$R1 \leftarrow R1 \times R3$
STORE	X, R1	$M[X] \leftarrow R1$

Constant

$\uparrow$

(4) Immediate field.

$\longleftarrow$ Instruction Length (Size) $\longrightarrow$

OPCODE	Mem AF	Reg AF	Immediate field

(1) n bit opcode can perform 2^n operations.

OPCODE $\Rightarrow$ # operation given $= 2^n$

(eg) Instruction set of size = 11

11 Distinct operation/Instⁿ Performed

So opcode = 11 $\Rightarrow 2^n = \boxed{4 \text{ bit}}$

(2) Memory AF (eg) 4MB Memory then 2^{22} B then Mem AF = $\boxed{22 \text{ bit}}$

(3) Register AF (eg) 25 Register $\Rightarrow$ To Represent 25 Reg $\Rightarrow$ Reg AF = $25 = 2^n = \boxed{5 \text{ bit}}$

Immediate field = $\boxed{n \text{ bit}}$

Signed

n bit Signed Range = $-\left(2^{n-1}\right)$ to $+\left(2^{n-1}-1\right)$

4bit Signed Range = $-\left(2^{4-1}\right)$ to $+\left(2^{4-1}-1\right)$

$= \boxed{-8 \text{ to } +7}$ Ans

Un Signed

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

eg 4bit Unsigned Range = 0 to $2^{4}-1$

$= \boxed{0 \text{ to } 15}$ Ans

Note:

Immediate field is n bit

Unsigned Range = $(0 \text{ to } 2^n - 1)$

Signed Range = $-(2^{n-1}) \text{ to } +(2^{n-1} - 1)$

Example

If immediate field is 4 bit

Then unsigned range = $(0 \text{ to } 2^4 - 1) \Rightarrow 0 \text{ to } 15$

Signed Range = $-(2^{4-1}) \text{ to } +(2^{4-1} - 1)$

$= \boxed{-8 \text{ to } +7}$ Ans

Q.

A machine has a 32-bit architecture, with 1-word long instructions. It has 64 registers, each of which is 32 bits long. It needs to support 45 instruction, which have an immediate operand in addition to two register operands. Assuming that the immediate operand is an unsigned integer the maximum value of the immediate operand is _____. [GATE-2014 (Set-1)]

1word = 32bit

64 Reg = 2^n = Reg AF = 6bit

45 Instⁿ ⇒ opcode = 6bit

⟵ Instⁿ size = $\boxed{32 \text{bit}}$ ⟶

OPCODE	Reg1	Reg2	Immediate
6	6	6	

Immediate field = 32 − (6+6+6) = 32 − 18 = $\boxed{14 \text{bit}}$

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

⇒ 0 to $2^{14} - 1$ ⇒ $\boxed{16383}$ Ans

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$ $(9 - 16) = 4$bit

$\boxed{2^5 = 32\ (17 \text{ to } 32) = 5\text{bit}}$

$\boxed{2^6 = 64\ (33 \text{ to } 64) = 6\text{bit}}$

$2^7 = 128$ $(65$ to $128) = 7$bit

$2^8 = 256$ $(129$ to $256) = 8$bit

$2^9 = 512$ $(257$ to $512) = 9$bit

$2^{10} = 1024$ (K)

$2^{10} = 1$K

$2^{20} = 1$M

$2^{30} = 1$G

$2^{40} = 1$T

$2^{50} = 1$ Petta

$2^{60} = 1$ Exa

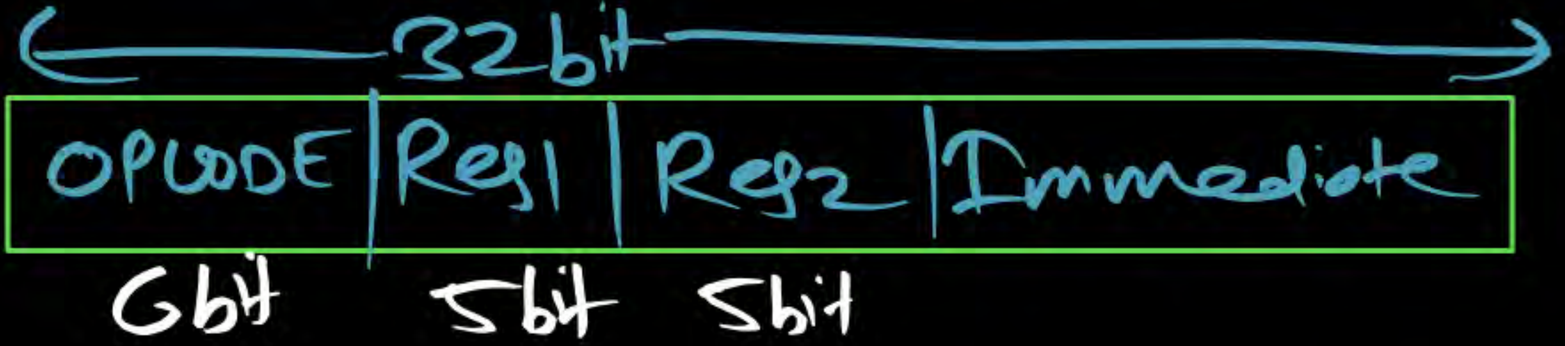
$2^{70} = 1$ Zetta

$2^{80} = 1$ yotta.

Q.

A processor has 40 distinct instructions and 24 general purpose registers. A 32-bit instruction word has an opcode, two register operands and an immediate operand. The number of bits available for the immediate operand field is $\boxed{16bit}$.

[GATE-2016 (Set-2)]



40 Distinct Instr^n/operation $\Rightarrow$ opcode = 6bit

24 Register $\Rightarrow$ Reg AF = 5bit

$$\text{Immediate field} = 32 - (6+5+5)$$
$$= 32 - 16$$
$$= \boxed{16bit} \text{ Ans}$$

Q. Consider a processor with 64 registers and an instruction set of size twelve. Each instruction has five distinct fields, namely, opcode, two source register identifiers, one destination register identifier, and a twelve-bit immediate value. Each instruction must be stored in memory in a byte-aligned fashion. If a program has 100 instructions, the amount of memory (in bytes) consumed by the program text is _____.

[GATE-2016 (Set-2): 2Marks]

OPCODE	Reg1	Reg2	Reg3	Immediate field
4bit	6bit	6bit	6bit	12 bit

64 Register = Reg AF = 6bit

Instn set = 12 ⇒ opcode = 4bit

Instruction size = 4+6+6+6+12

= 34 bit

$\frac{34}{8} = \lceil 4.25 \rceil = \boxed{4.25}$ X

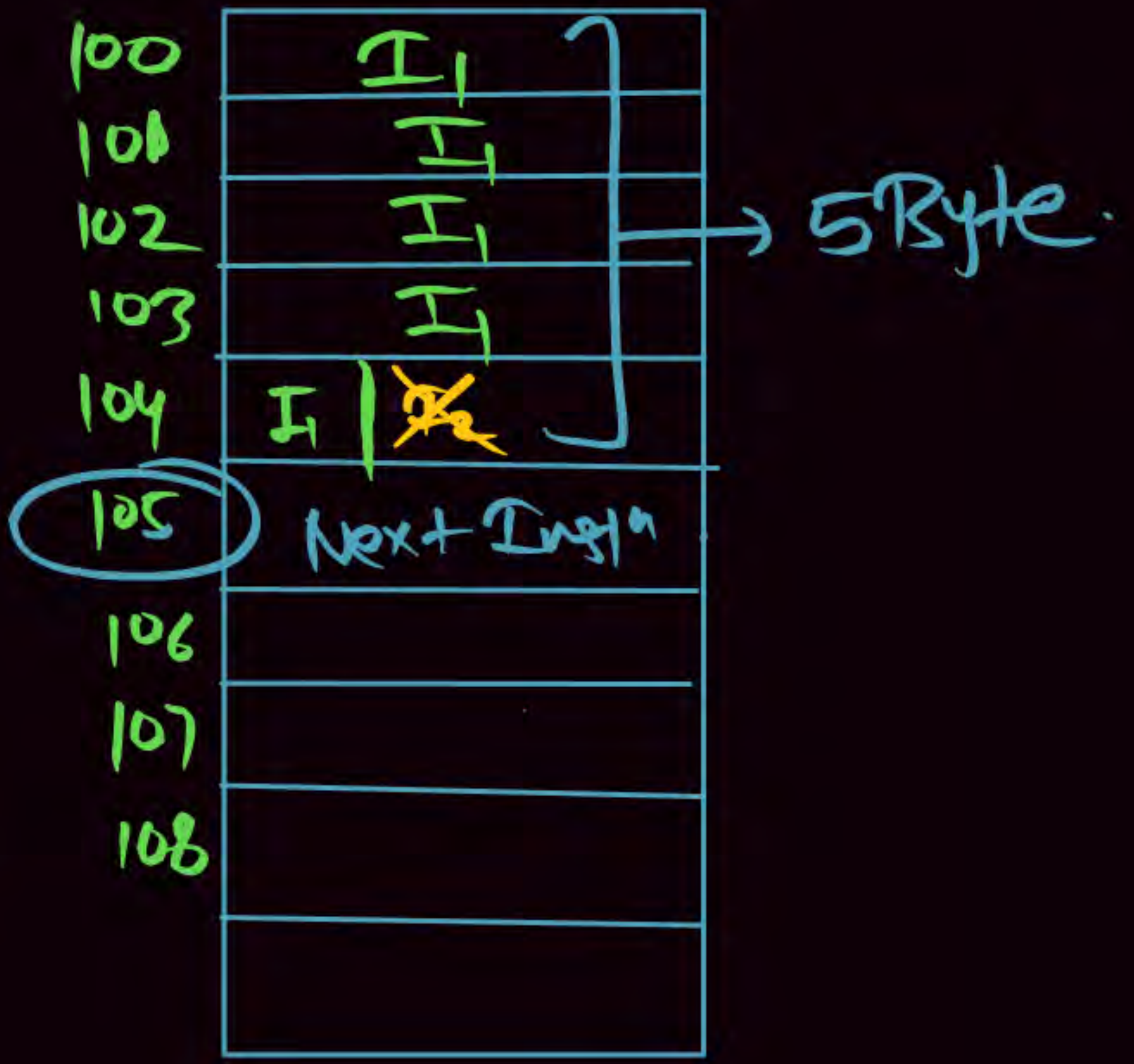
Instruction Size = 34 bits

Instruction Size = $\boxed{5 \text{ Byte}}$

Program has 100 Instruction

Program Size = 100 X 5 Byte

= $\boxed{500 \text{ Byte}}$ Ans

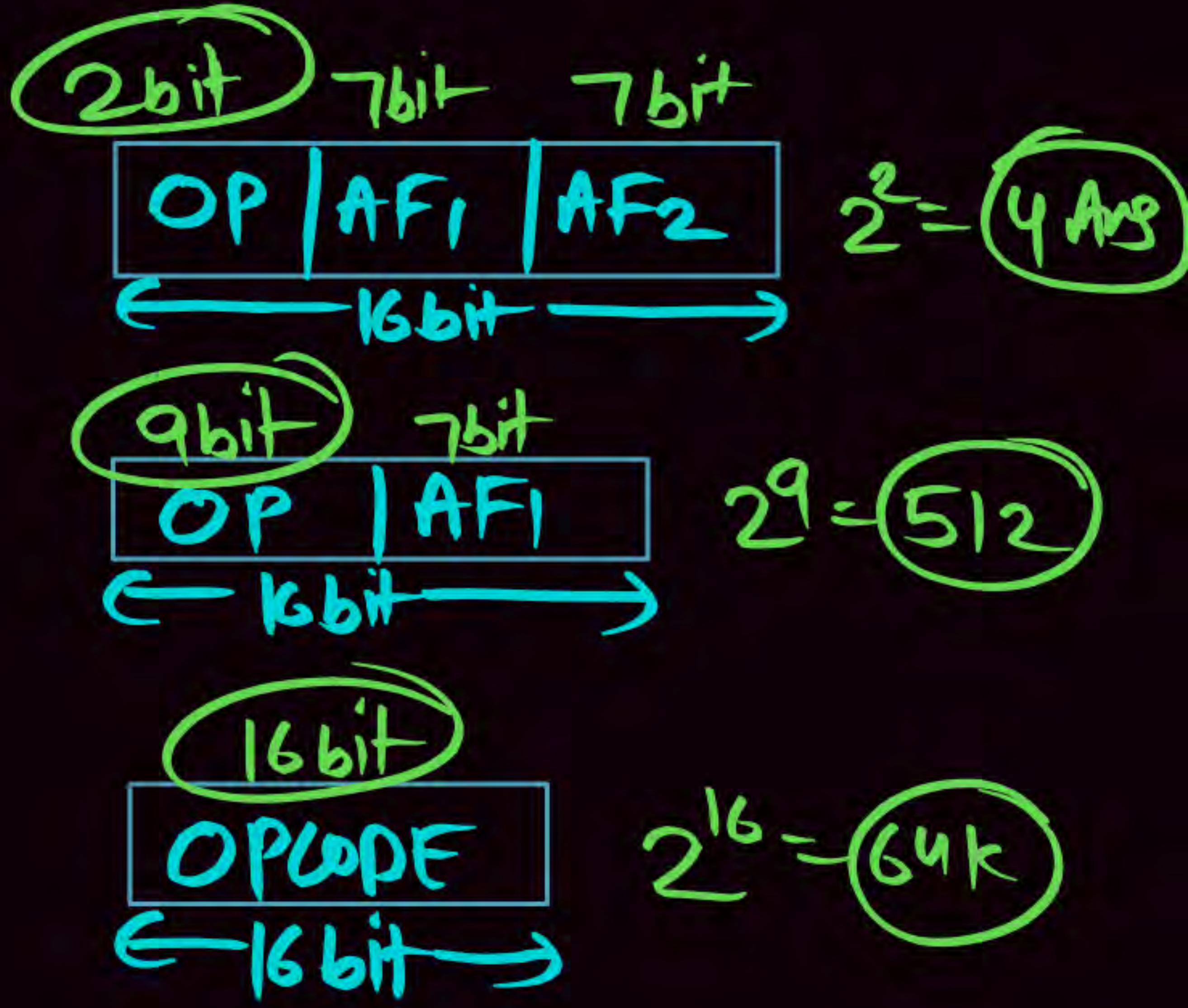


Q Consider a 16bit Instn with AF = 7bit than How Many Instn | operation Can be Supported by

(i) 2AF $\Rightarrow$ 4 Ans

(ii) 1AF = 512 Ans

(iii) 0AF = 64k Ans



→ (eg) MOV R1, R2 (or) LOAD r1, [A]

3AF:

OP	AF1	AF2	AF3

→ Support upto 3AF (it can support 2AF, 1AF & 0AF also)

2AF/2AI:

OP	AF1	AF2

→ Support 2AF (1AF & 0AF also)

1AF/1AI

OP	AF1

→ Support 1AF [0AF also]

0AF/0AI

OPCODE

→ Support 0AF.

Expand opcode Technique.

↳ Fixed length Instruction.

⇓

Variable length opcode.

$$\begin{tikzcd} \underline{2AF} \arrow[d] \\ \underline{1AF} \arrow[d] \\ \underline{0AF} \end{tikzcd}$$

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

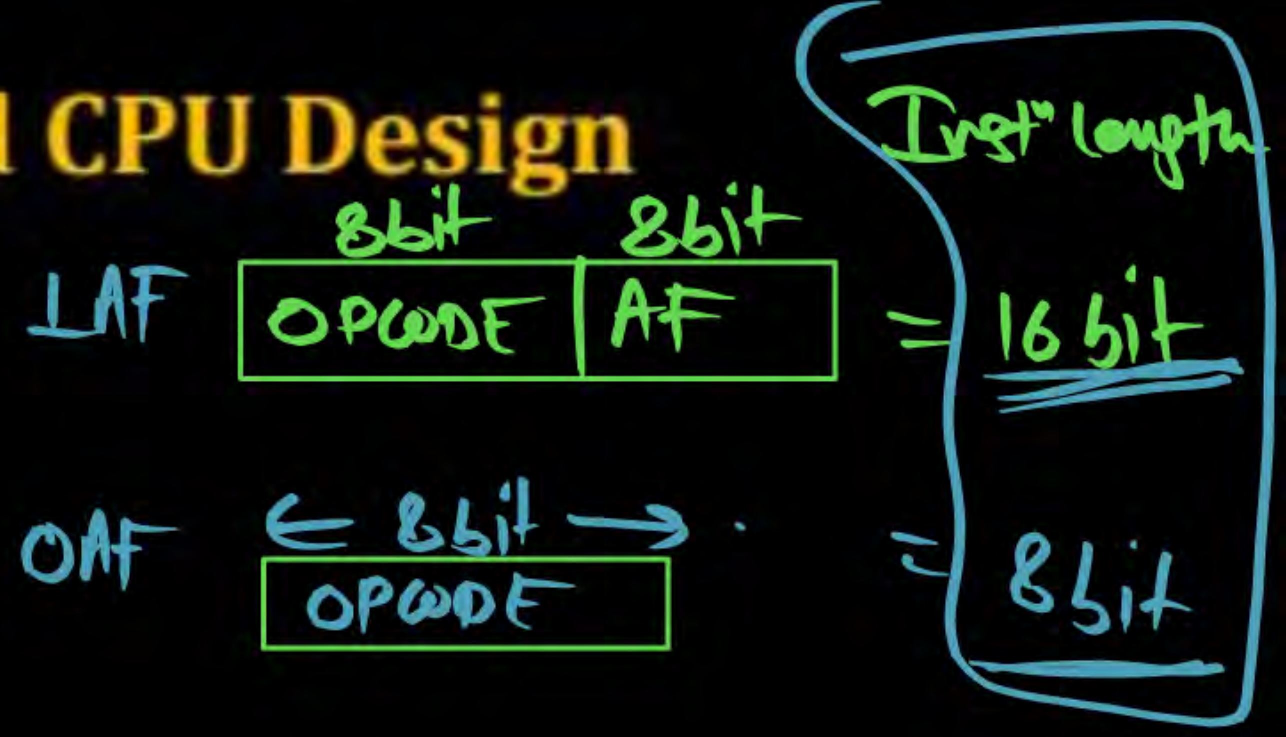
OPCODE = 8 bit

Address field = 8 bit

Fixed Length opcode.

OPCODE = 8bit

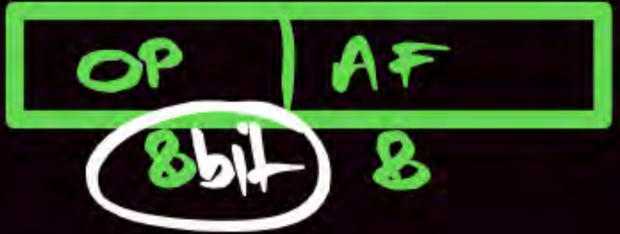
AF = 8bit



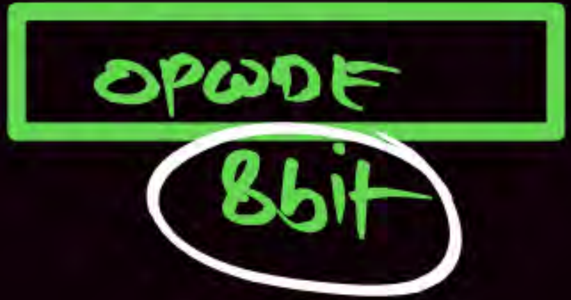
Variable length (Fixed length Opcode)

OPCODE = 8bit
AF = 8bit

Instⁿ size

LAF:


= 16 bit

OAF


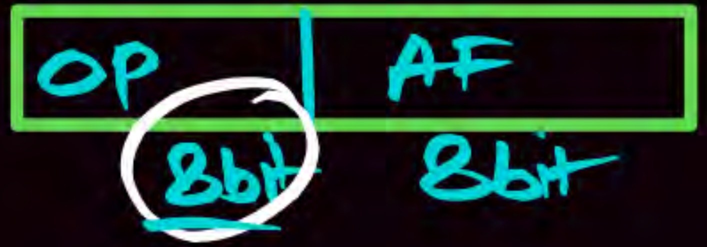
= 8bit

Fixed length Instⁿ

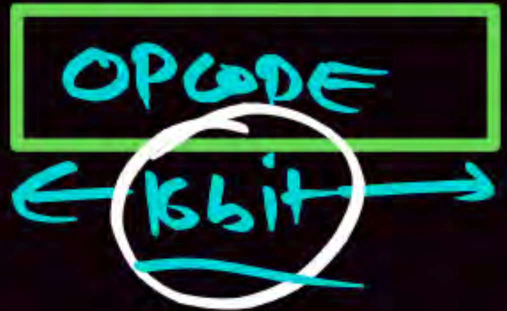
→ Opcode is variable length.

OPCODE = 8bit
AF = 8bit

Instⁿ size

LAF


= 16bit

OAF:


= 16bit

(i) 1 Address Instruction Design:

← Instruction →

OPCODE	Address
8 bit	8 bit

= 16 bit

(ii) 0 Address Instruction Design:

← 8 bit →

OPCODE

← Instruction →

= 8 bit [No need of expand opcode technique]

2AI

op	AF₁	AF₂
8	8	8

Instⁿ = 24 bit

op	AF1	AF2	AF₃
8	8	8	8

Instⁿ = 32 bit

Fixed Length Instruction Supported CPU Design

$\Downarrow$

Instruction length fixed

$\Downarrow$

Opcode is variable length.

OPCODE = 8 bit

A.F = 8 bit

(i) 1 Address Instruction Design:

← Instruction →

OPCODE	Address
8 bit	8 bit

= 16 bit Ans

Instn length

(ii) 0 Address Instruction Design:

← Instruction →

OPCODE
← 16 bit →

= 16 bit [Here expand opcode technique required]

Expand opcode Tech.

Expand opcode Technique. : We Start from Primitive Instⁿ.

[Smallest opcode bit]

I. Primitive Instⁿ : Lowest opcode bit wallah.

II. Derived Instⁿ : Higher opcode bit

III. More Derived Instⁿ = Highest opcode bit.

eg Type1 Type2, Type3

Expand opcode Technique

The Technique Used to generate Low Order Instⁿ After Allocation Higher Order Instruction.

Expand Opcode Technique

- Primitive instruction means smallest opcode instruction.

✓ **Step 1:** Identify the primitive instruction in the CPU. According to the Question.

✓ **Step 2:** Calculate the total number of possible operation.

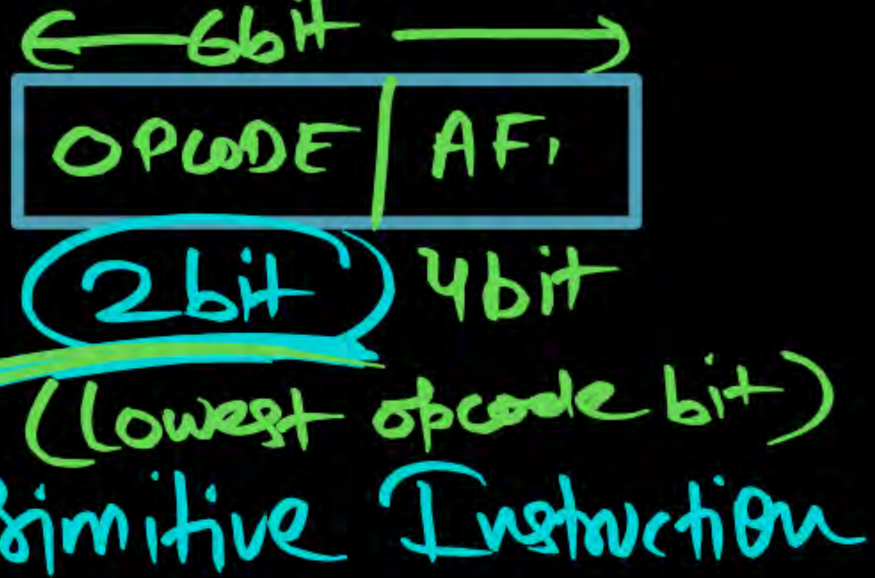
✓ **Step 3:** Identify the free opcode after allocating the existed instruction.

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

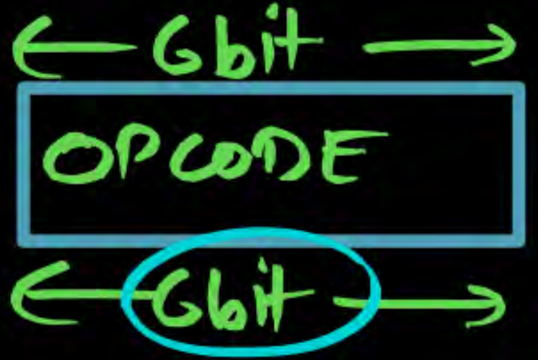
$$\boxed{\text{Free opcode} \times 2^{\text{Increment bit in opcode}}}$$

Q.1

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?

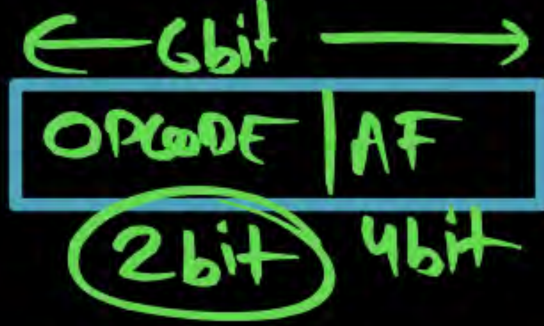


Primitive Instruction



Derived Instruction.

Step1:



Step2: opcode = 2bit then Total # operation in 1AF $= 2^2 =$ 4 operation

In the Question Given we Used '2' 1Address Instn.
Using.

Step3 : Number of Free opcode After Allocating 1AF = 4-2 = $\boxed{2}$ Free

Step4 Derived Instn

OAF

OPCODE

$\xleftarrow{\quad 6bit \quad}$

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

Total # operation (in Derived Instn in OAF =

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

$\Rightarrow 2 \times 2^{4}$

= 32 Opern/Instn Ans

Analysis

Primitive

← 6bit →

OPCODE	AF
2bit	4bit

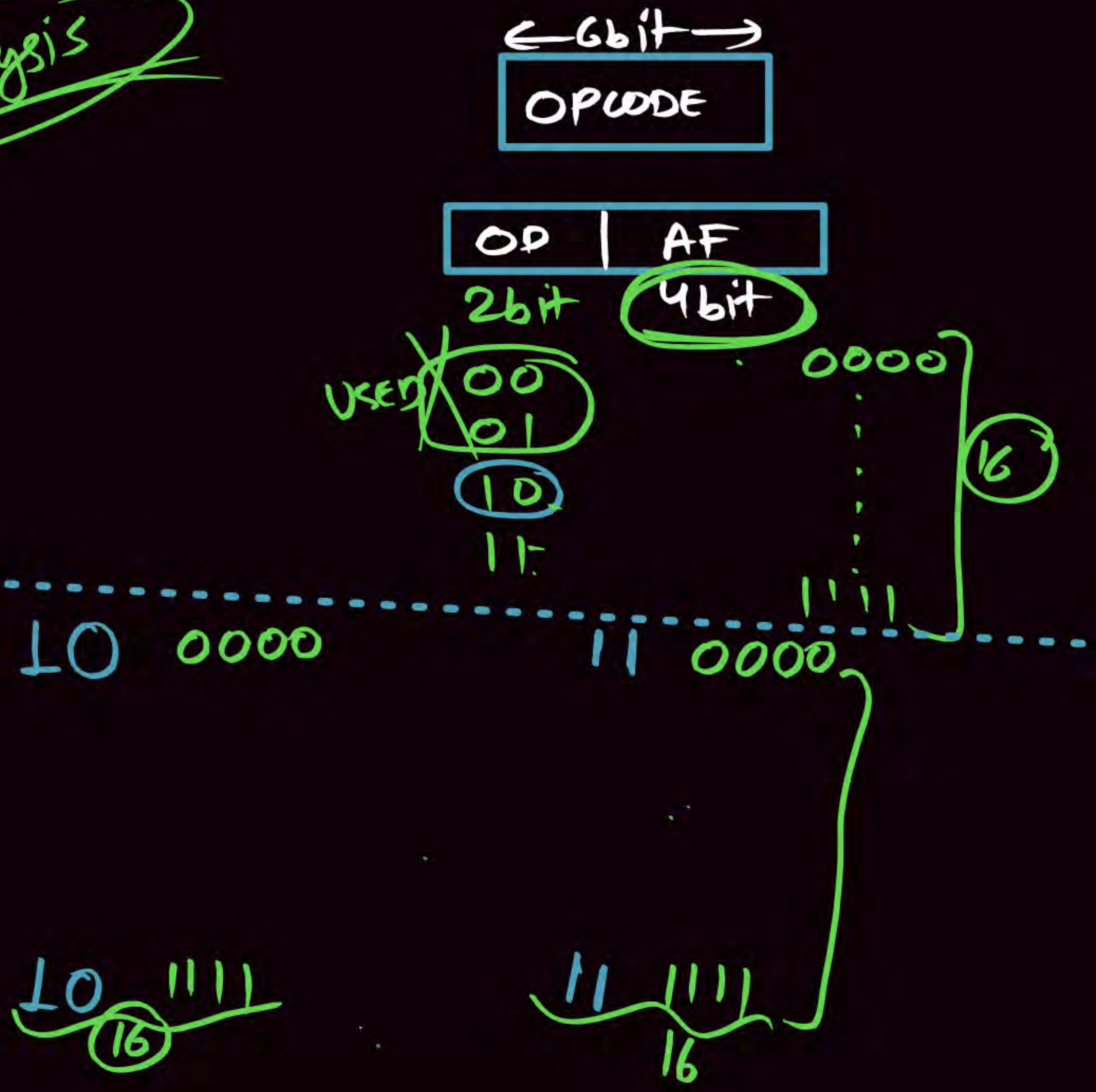
Analysis

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

Any op ← 00 · ← 01 · USED

10 · 11 · Free

$\underbrace{\qquad}$ · 4 operation.



3bit obcode $= 2^3 = 8$ obcy

000 $\longrightarrow$	Addition
001 $\longrightarrow$	Subtr
010 $\longrightarrow$	MUL
011 $\longrightarrow$	AND
100 $\longrightarrow$	OR
101 $\longrightarrow$	XOR
110 $\longrightarrow$	DIV
111 $\longrightarrow$	LOAD

Q.

Word Size = 1 Byte

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?

8bit Word

1 Word = 8bit

$\boxed{1\text{Word} = 1\text{Byte}}$

3 Word Instn ⇒

Instn size = 3 Word ⇒ 3 × 8bit
= 24bit

256 Word Memory

256 × 1B = 256B Memory $= 2^8$ Byte

$\boxed{AF = 8bit}$

Q.

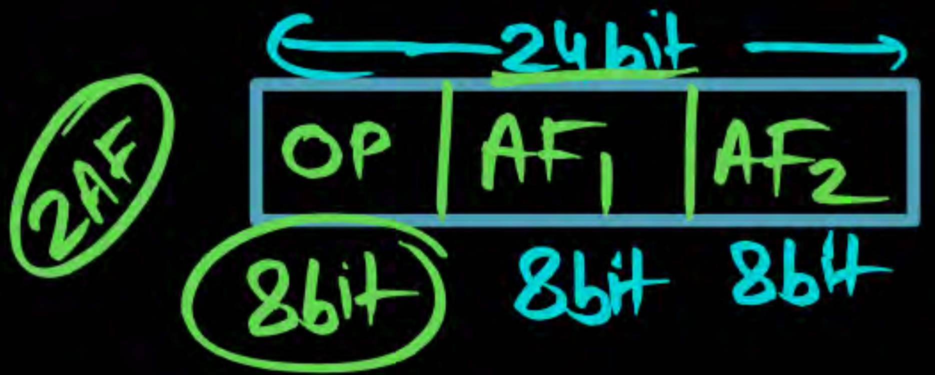
Word Size = 1 Byte

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?

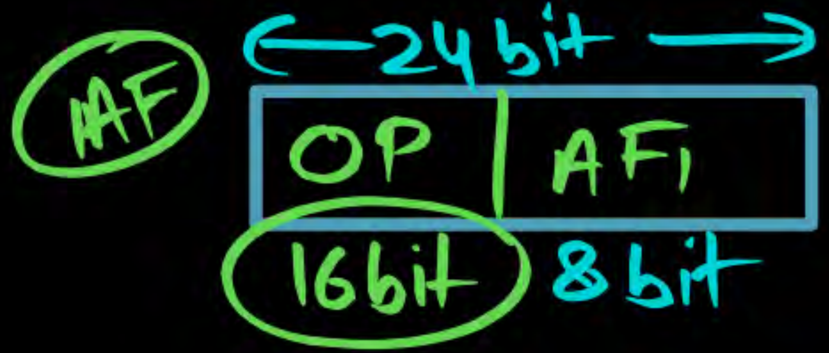
Instn size = 24 bit

AF = 8 bit

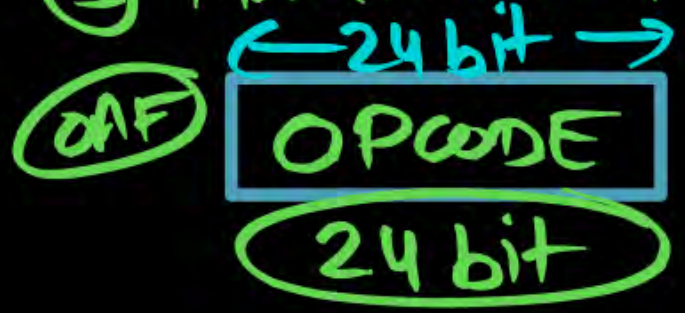
① Primitive



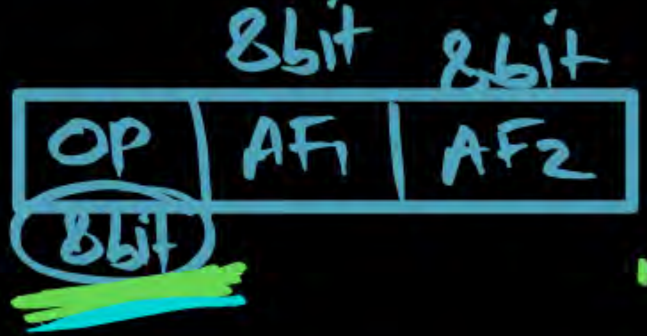
② Derived



③ More (Further) Derived.



Step 1 : Identify the Primitive Instn



Step 2 Total # operation in 2AF = 2^8 = 256 operation

Given 254 2AI USED.

Step3: Number of Free opcode After Allocating 2AI $= 256 - \overset{\text{USED}}{254} = \underset{\text{Free}}{\boxed{2}}$

Step4

OPCODE	AFI
16bit	8bit

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

$2 \times 2^{16-8} \Rightarrow 2 \times 2^{8} \Rightarrow 2^{9} = \underset{\text{operation}}{\boxed{512}}$

$\boxed{\text{Total \# operation in LAF (Derived Inst}^{n}\text{)} = 512.}$

Given in the Question LAF = 256 operation | Instⁿ USED.

Free opcode After Allocating LAF $= 512 - 256 = \underset{\text{Free}}{\boxed{256}}$

Free opcode in 1AF (Derived) $= \boxed{256 \text{ Free}}$

Further Derived

OAI $\boxed{\text{OPCODE}}$ $\boxed{24 \text{ bit}}$

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

Total # operation in OAI/OAF $= 256 \times 2^{24-16}$

$\Rightarrow 256 \times 2^8$

Total # operation in OAI $= \boxed{256 \times 2^8}$ Ans

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?

Instrn size = 24 bit
AF = 8 bit

① Primitive

2AF

← 24 bit →

OP	AF_1	AF_2
8bit	8bit	8bit

Total # operation $= 2^8 = 256$

Given(2AF) = 254

#Free opcode $= 256 - 254 = 2$

② Derived

1AF

← 24 bit →

OP	AF_1
16bit	8 bit

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

$= 2 \times 2^{16-8} \Rightarrow 2 \times 2^8$

Total # operation $= 2^9 = 512$

Given(1AF) = 256.

#Free opcode $= 512 - 256 = 256$

③ More(Further) Derived.

0AF

← 24 bit →

OPCODE
24 bit

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

$= 256 \times 2^{24-16}$

$= \boxed{256 \times 2^8}$ Ans

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

Solution(c): 240

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

$\longleftarrow$ **16 bit** $\longrightarrow$

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 ______.

[GATE-2018 : 2 Marks]

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 _____.

[GATE-2020 : 2 Marks]

THANK YOU!