COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and Addressing Modes

Lecture_ 05

Vijay Agarwal sir

TOPICS TO BE COVERED

01 Expand Opcode Technique

02 Addressing Modes

Instruction Concept

Instruction format

Instruction Set Archi.

1. STACK Based org.
2. Accumlator Based org.
3. General Register org.
   - (i) Reg - Mem Ref.
   - (ii) Reg - Reg Ref.

Q) Consider a 16 bit Instruction, & 7bit AF then How many Total number of operation Supported by

2AF [4 Ans] 2AI

←16bit→

OPCODE	AF_1	AF_2
2bit	7bit	7bit

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

1AF [512 Ans] 1AI

←16bit→

OPCODE	AF_1
9bit	7bit

$\# \text{operation} = 2^9 = 512$ Ans

0AF [64k Ans] 0AI

←16bit→

OPCODE
16 bit

$\# \text{operations} = 2^{16} = 64K$ Ans

$2^6 \; 2^{10}$

64K.

3AF|3AI   OP | AF$_1$ | AF$_2$ | AF$_3$ → (It can Support upto 3AF|3AI)
it can support 2AI, 1AF, 0AI)

2AF|2AI   OP | AF1 | AF$_2$ → (It can support upto 2AF|2AI
& 1AI & 0AI)

1AF|1AI   OP | AF1 → (It can support upto 1AF|1AI)
& 0AI)

0AF|0AI   OPCODE (OP) → (It can support 0AI|0AF Inst^n)

Expand opcode Technique :

Fixed length Instruction.

(variable length opcode).

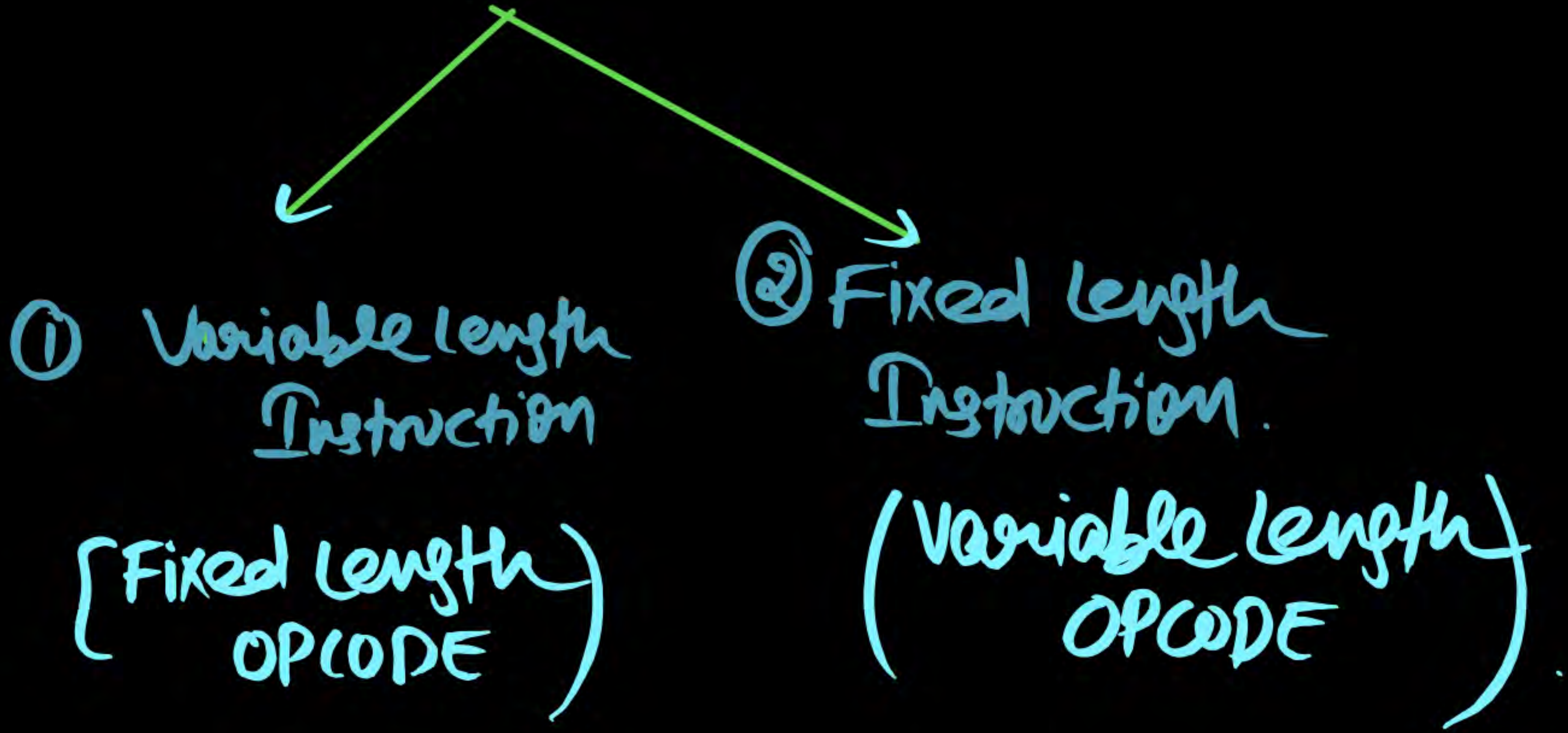
2AF

↓

1AF

↓

0AF



① Variable length Instruction

[Fixed length OPCODE)

② Fixed length Instruction.

(Variable length OPCODE).

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

Variable Length Instⁿ

(Fixed length opcode)

OPCODE = 8bit

AF = 8bit

Instⁿ size.

LAF : | OP | AF1 | = 16 bit

8bit 8bit

OAF: 8bit

| OPCODE | = 8bit

Fixed Length Instruction

Variable length opcode.

OPCODE = 8bit

AF = 8bit

Instⁿ size.

LAF: ← 16bit →

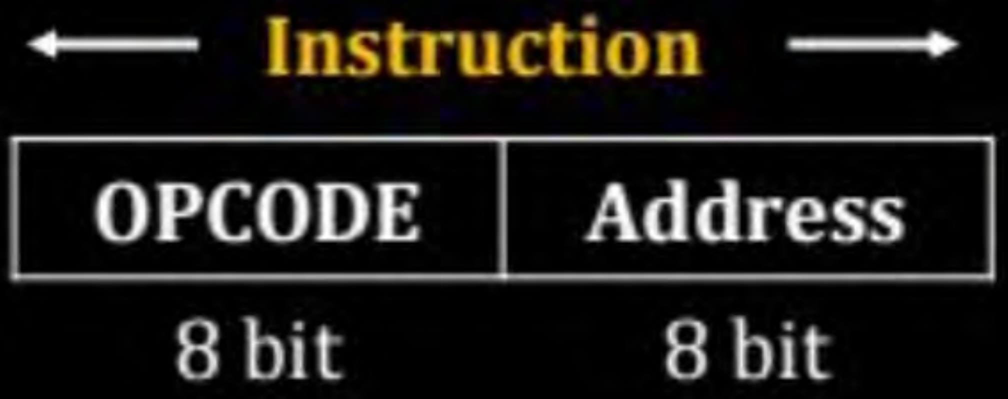
| OP | AF1 | = 16 bit

8bit 8bit

OAF: ← 16bit →

| OPCODE | = 16 bit

(i) 1 Address Instruction Design:

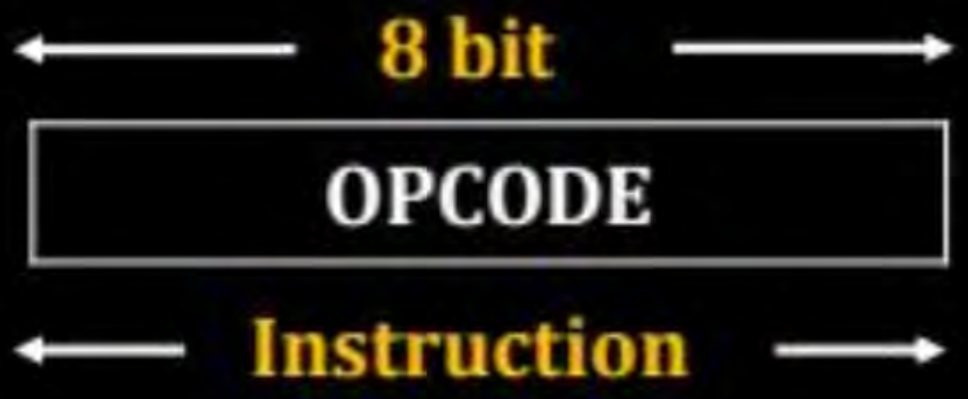


Instruction

OPCODE	Address
8 bit	8 bit

= 16 bit

(ii) 0 Address Instruction Design:



8 bit

OPCODE

Instruction

= 8 bit

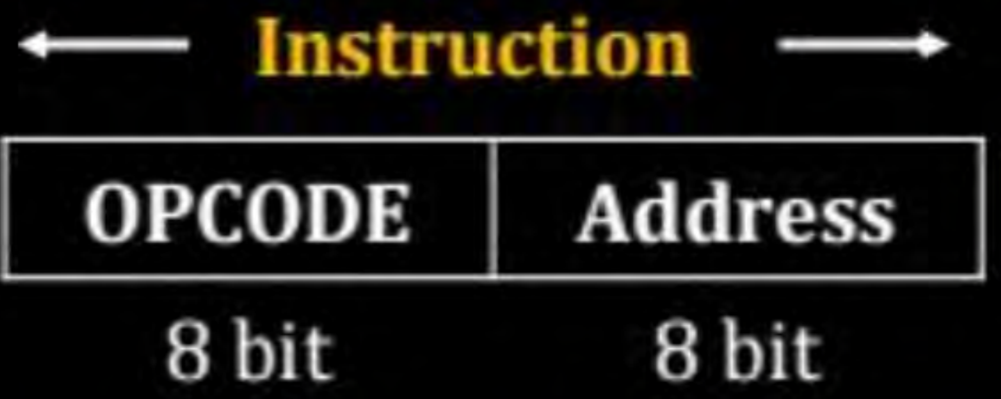
[No need of expand opcode technique]

Fixed Length Instruction Supported CPU Design

OPCODE = 8 bit

A.F = 8 bit

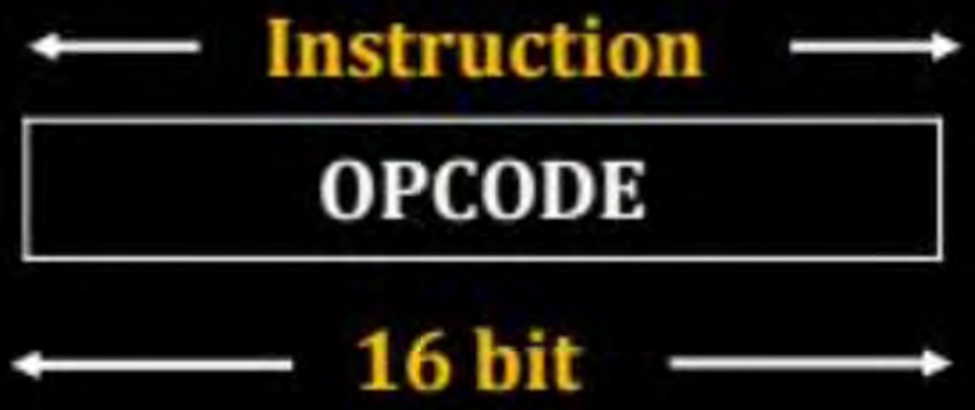
(i) 1 Address Instruction Design:


Instruction

OPCODE	Address
8 bit	8 bit

= 16 bit

(ii) 0 Address Instruction Design:


Instruction

OPCODE

16 bit

= 16 bit [Here expand opcode technique required]

(Fixed length Instruction].

Expand opcode Technique:

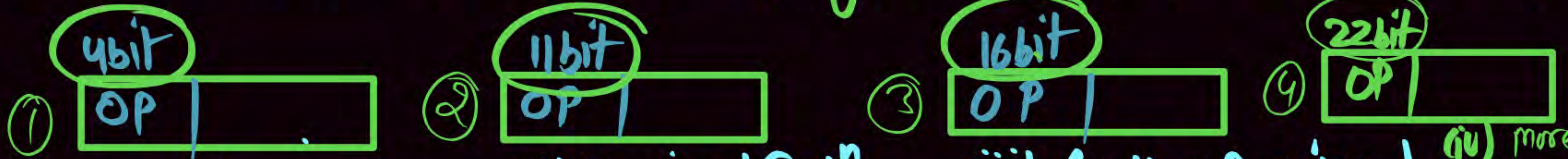
we starts from the Primitive Instn.

① Primitive Instruction. [lowest Opcode bit or lowest bit in OPCODE)

② Derived Instruction (Higher opcode bit)

③ further Derived Instruction (Highest opcode bit wallah).

eg



Primitive Instn

(ii) Derived Instn

(iii) further Derived.

(iv) more further Derived Instruction.

Primitive, Derived, further Derived - . . -

eg Type 1, Type 2, Type 3, Type 4.

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.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?

Instrn Length = 6bit
AF = 4 bit

Step1

← 6bit →

OPCODE	AF1
2bit	4bit

Primitive Instrn

← 6bit →

OPCODE
6bit

Derived Instrn.

Step2: Total Number of Operation|Instrn in 1AF = 2^2 = 4

Given 1AF Instrn = 2

Step3 Number of Free opcode After Allocating 1AF. = 4 − 2 = 2 Free

OAF:

$\boxed{\text{OPCODE}}$

6bit

Increment bit in OPCODE.

Step 4

Total # operation in OAF = Free opcode × 2

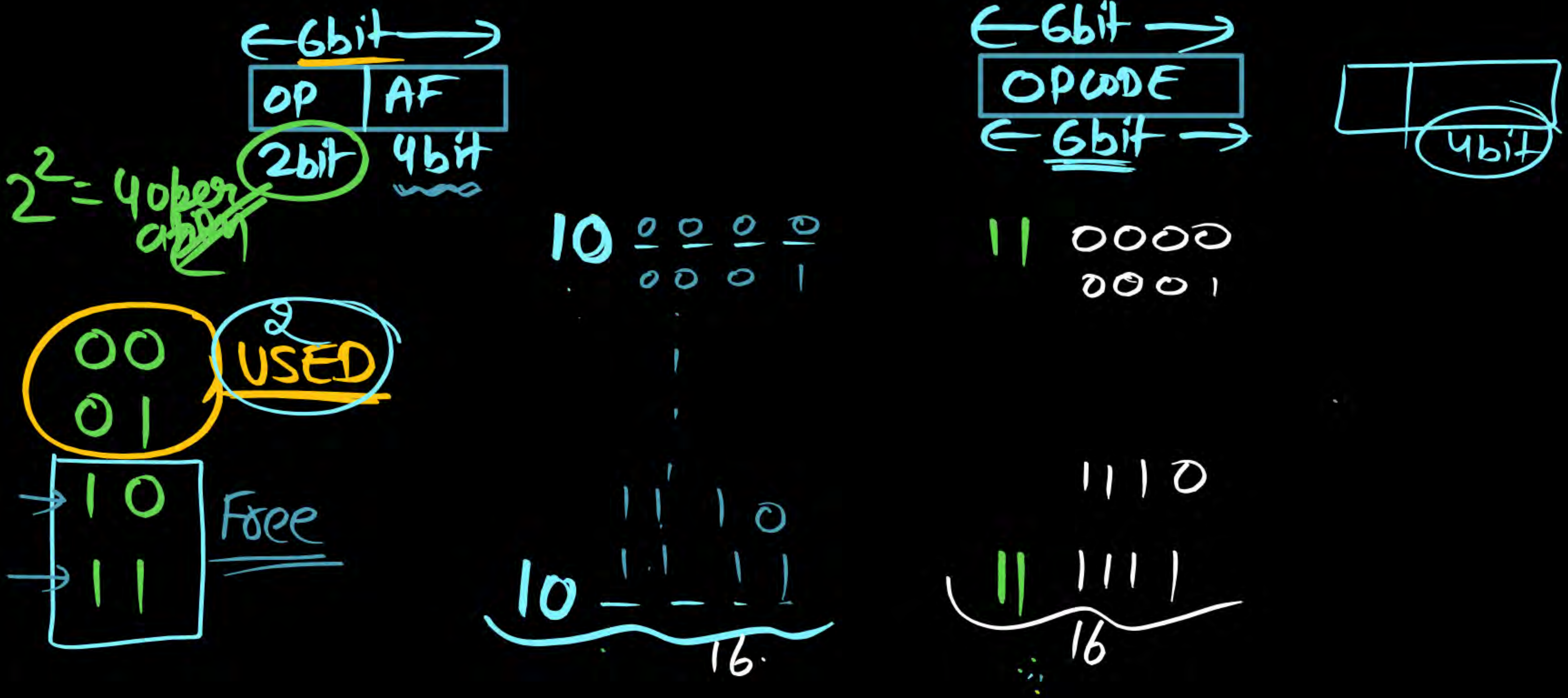
$$\Rightarrow 2 \times 2^{6-2}$$

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

$$= 32 \text{ Ans}$$

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?



Q.2

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?

2AI
1AI
0AI

1word = 8bit = 1Byte.

Memory = 256 Word = 256 × 1Byte = 256 Byte

Memory = 256Byte $[2^8 B] \Rightarrow AF = \boxed{8\,bit}$

Instruction Size = 3Word ⇒ 3 × 8bit = $\boxed{24\,bit}$

① ← 24bit →

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

Primitive Instn

② ← 24bit →

OP	AF1
$\boxed{16bit}$	8

Derived Instn

③ ← 24bit →

OPCODE
$\boxed{24\,bit}$

further Instn.

I Primitive

OP	AF1	AF2
8bit	8	8

Total Number of operation in 2AF[Primitive Inst^n] = 2^8 = 256 oper^n/Inst^n

Given 2AF/2AI = 254

#Free opCode [operational Code) After Allocating 2AF = 256-254 = 2

II Derived Inst^n

←24bit→

OPCODE	AF1
16bit	8bit

Total # operation in 1AF = Free opcode × $2^{\text{Increment bit in opcode}}$

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

= 512 operation.

Given 1AI(AF) = 256.

Number of Free opcode After Allocating 1AF = 512 - 256 = 256.

III further Derived

OPCODE

← 24 bit →

Total # operation in OAF

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

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

$$\Rightarrow 256 \times 2^{8} \quad \underline{\text{Ans}}$$

$$\simeq 2^{16} \ \underline{\text{Ans}} \ \text{or} \ 64K$$

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

15 Register

⇓

Reg AF = 4bit

Instn Size = 16bit

2AF

←16bit→

OP	AF$_1$	AF$_2$
8bit	4bit	4bit

Primitive Instn

←16bit→

OP	AF$_1$
12bit	4bit

1AF

Derived Instn

← 16 bit →

OP	AF	AF
8bit	4	4

Total # operation in 2AF = 2^8

Assume 2AI/2AF Inst^n/operation = X

Free opcode After Allocating 2AI = $[2^8 - X]$

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

← 16bit →

OP	AF
12bit	4bit

#operation in 1AF | 1AI = $[2^8 - X] \times 2^{12-8}$

$[2^8]$ 256

$$2^8 \Rightarrow [2^8 - X]\,2^4 \Rightarrow \frac{2^8}{2^4} = 2^8 - X$$

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

$$x = 256 - 16 = \boxed{240}$$ Ans

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 ___ 14 Ans Ans

[GATE-2020 : 2 Marks]

64 Register ⇒ Reg AF = 6 bit

Inst^n Size = 16 bit

I type & R type.

I type: ⟵ 16 bit ⟶

OP	Reg AF	Immediate field
6 bit	6 bit	4 bit

16 - (6+4) = 6 bit

R Type: ⟵ 16 bit ⟶

Op	Reg AF	Reg AF
4 bit	6 bit	6 bit

16 - (6+6) = 4 bit

R type:
(Primitive)

←16bit→

OP	Reg1	Reg2.
4bit	6bit	6bit

I type:
(Derived)

←16bit→

OP	Reg	Immediate
6bit	6bit	4bit

Total # operation in R type $= 2^4$

Assume R type Insln = X

Number of Free opcode After Allocating R type $= 2^4 - X$

I-type

←16bit→

OP	Reg	Imm.
6bit	6	4

Total # operation in I type $= (2^4 - X) \times 2^{6-4} \Rightarrow 8 = (2^4 - X) \times 2^2$

$$8 = (2^4 - X) \times 4$$

$$2 = 16 - x$$

$$x = 16 - 2 = \boxed{x = 14}$$ Ans

THANK YOU!