COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and Addressing Modes

Lecture_01

Vijay Agarwal sir

01 Machine Instruction

02 Instruction Format

Introduction of COA

- Computer Generation
- CO & CA
- Component of the Computer
- Register
- Instruction Cycle
  - Fetch cycle
  - Execute cycle

Instruction Cycle With Interrupt

- GATE PYQ.
- Memory
  - A.L
  - D.L
- System Bus
  - A.L
  - D.L
  - C.L
- Byte & Word Addressable Memory
  - Endian M/C ism
    1. Little Endian
    2. Big Endian.
- Pins Type & Register Size
- System Bus & Word Length Concept.
- Type of Arch [Von neumann & Harward Arch.)
- Computer, Program, Instruction, Data

Basic Terms and Notation

The alphabet of computers, more precisely digital computers, consists of 0 and 1.

Each is called a *bit*, which stands for the binary digit.
The term *byte* is used to represent a group of 8 bits.
The term *word* is used to refer to a group of bytes that is processed simultaneously.

The exact number of bytes that constitute a word depends on the system, For example, in the Pentium, a word refers to four bytes or 32 bits. On the other hand, eight bytes are grouped into a word in the Itanium processor.

We use the abbreviation “b” for bits, “B” for bytes, and “W” for words.

Sometimes we also use *doubleword* and *quadword*. A doubleword has twice the number of bits as the word and the quadword has four times the number of bits in a word.

Bits in a word are usually ordered from right to left, as you would write digits in a decimal number. The rightmost bit is called the *least significant bit* (LSB), and the leftmost bit is called the *most significant bit* (MSB).

Byte Ordering

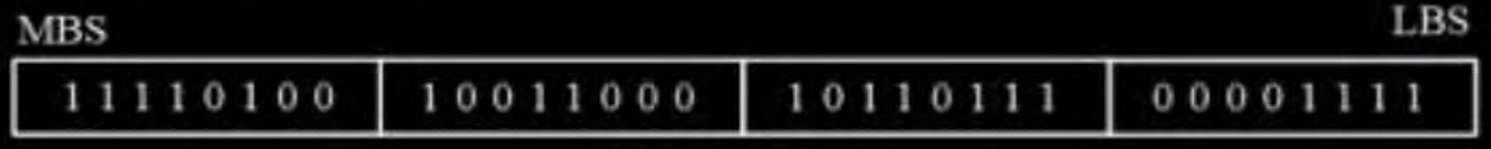
Storing data often requires more than a byte.

Suppose that we want to store these 4-byte data in memory at locations 100 through 103.

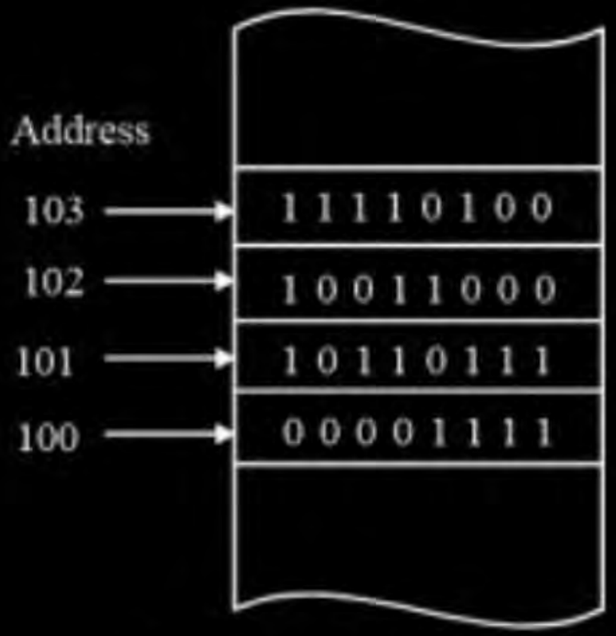
How do we store them?

Figure Shows two possibilities: Least significant byte or Most significant byte is stored at location 100. These two byte ordering schemes are referred to as the little endian and big endian.

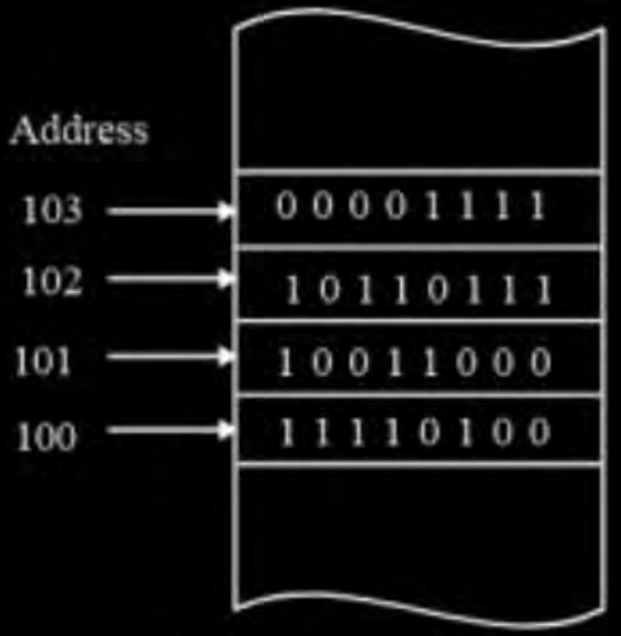
Two Important Memory Design Issues



(a) 32- bit data



(b) Little-endian byte ordering



(c) Big endian byte ordering

Two byte ordering schemes commonly used by computer systems.

Byte Addressable

Q.1

Memory = 4 GByte.

$\Rightarrow 2^2 . 2^{30}$ Byte

$\Rightarrow 2^{32}$ Byte

Address = 32 bit.

Q.2 Address = 25 bit

$\Rightarrow 2^{25}$ Byte $\Rightarrow 2^5 . 2^{20}$ Byte.

= 32 MByte

Word Addressable

(Q.1) Address = 27 bit

then Memory = 2^{27} words

= 128 M Words.

(Q.2) Memory = 512 M words.

$\Rightarrow 2^{9} \cdot 2^{20}$ words

$\Rightarrow 2^{29}$ Words.

$\Rightarrow$ 512 M Words.

Instruction Format

ADD : Addition
MUL : Multiplication

OPCODE ⇒ operational Code

→ Type of operation.

OPERAND : DATA

(OR)

Address of operand.

Instruction Format

ADD : Addition
MUL : Multiplication.

OPCODE ⇒ operational Code.

↳ Type of operation.

2 bit opcode Can Perform $2^2 = 4$ operation.

Assume

00 → ADD
01 → OR
10 → XOR
11 → MUL.

Note 3 bit opcode can perform $2^3 \Rightarrow$ 8 operation.

n bit opcode can perform 2^n operation.

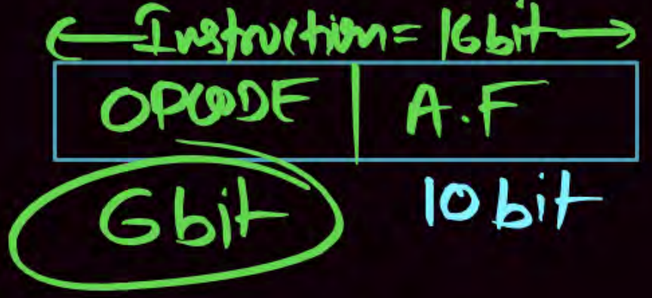
Note If Memory Size is given then we can Calculate Address field Size.

eg if Memory = 1MByte $\Rightarrow$ 20 bit Ans

Address [AF].

Q) If Instruction Size is 16 bits & Memory is 1K Byte then How Many Number of Operation Supported by the System?

Soln



Memory = 1K Byte

$\Rightarrow 2^{10}$ Byte

A.F = 10 bit

If opcode = 6 bit

then Total # operation = 2^6 = 64 operation.

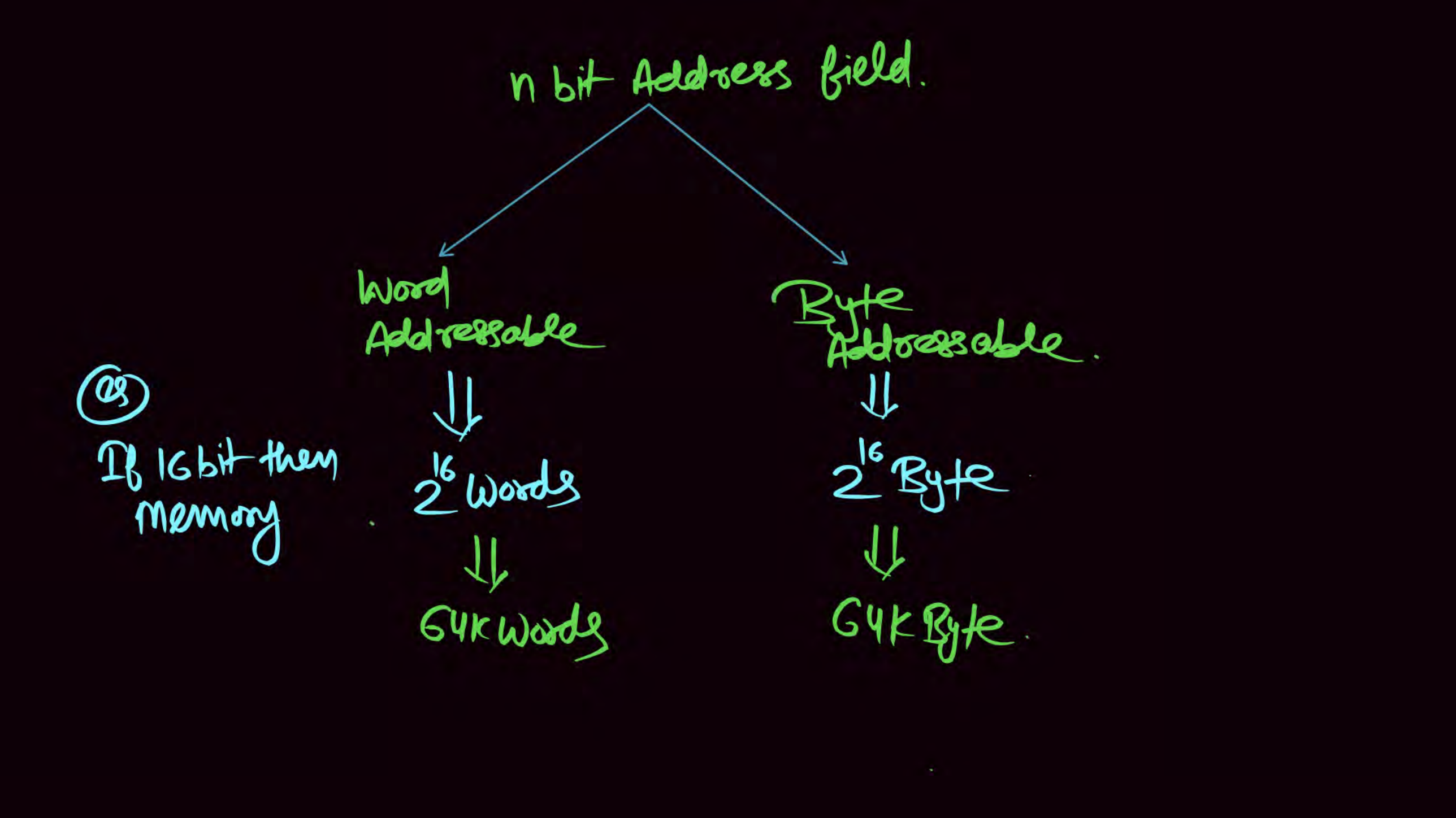
n bit Address field.

Word Addressable

2^n Words

Byte Addressable.

2^n Byte.


n bit Address field.
Word Addressable
Byte Addressable.
2^{16} Words
2^{16} Byte
64K Words
64K Byte
Q
If 16 bit then memory

Instruction Format

Important points

OPCODE	Address field

B: Byte
W: Word

① If n bit opcode can perform 2^n operation.

② If N operations then opcode = $\lceil \log_2 N \rceil$ bits $[2^n]$

① If n bit Address line can Represent Memory of Capacity 2^n [B/w]

② If Memory Capacity of M B/w then Address field = $\lceil \log_2 M \rceil$ bits.

Instruction Format

Q.1 If opcode is 6 bits then Total # operations ?

Soln 1 opcode = 6bit

Total # operation = 2^6 = 64 Ans

Q.2 If 100 operation supported by the system then size of opcode ?

Soln 2 100 operation $\leq 2^n$

opcode = 7bit

Q.1 If memory is 256kB then Size of Address field ?

Soln 1 256 kB $\Rightarrow 2^{18}$ Byte

Address = 18bit Ans

Q.2 If AF is 29 bits then Size of Memory ?

Soln AF = 29bit

Memory = 2^{29} Byte $\Rightarrow$ 512 MByte

Instruction Representation

- Within the computer each instruction is represented by a sequence of bits
- The instruction is divided into fields, corresponding to the constituent elements of the instruction

Instruction Format

AI: Address Instruction

AF: Address field.

ALU operation

OPCODE	Destination	Source 1	Source 2
ADD	~~~	-	-

Addition

ADD ∈ OPCODE

Instruction Format

AI: Address Instruction
AF: Address field.

Next Instn address

already PC

4AI|4AF:

OPCODE Destn S1 S2

3AI|3AF

ADD R_1 R_2 R_3; $R_1 \in R_2 + R_3$

2AI|2AF

ADD R_1 R_2; $R_1 \in R_1 + R_2$.

1AI|1AF

ADD R_1; $AC \in AC + R_1$

0AI|0AF

ADD.

Machine Instruction Characteristics

- The operation of the processor is determined by the instructions it executes, referred to as machine instructions or computer instructions
- The collection of different instruction that the processor can execute is referred to as the processor's instruction set **(ISA)**

Elements of a Machine Instruction

Operation Code (opcode)

- Specifies the operation to be performed. The operation is specified by a binary code, known as the operation code, or opcode.

Source Operand Reference

- The operation may involve one or more source operands, that is, operands that are inputs for the operation

Result Operand Reference

- The operation may produce a result

Next Instruction Reference

- This tells the processor where to fetch the next instruction after the execution of this instruction is complete

Instruction Representation

- Within the computer each instruction is represented by a sequence of bits
- The instruction is divided into fields, corresponding to the constituent elements of the instruction

Instruction Representation

- Opcodes are represented by abbreviations called mnemonics
- Examples includes:

Mnemonics	
ADD	Add
SUB	Subtract
MUL	Multiply
DIV	Divide
LOAD [LD]	Load data from memory
STORE [ST]	Store data to memory

- Operands are also represented symbolically
- Each symbolic opcode has a fixed binary representation

Instruction Set Architecture Classification

- Single accumulator organization.
- General register organization.
- Stack organization.

Q.1

Consider a Hypothetical Processor which support 128 byte memory and instruction length is 16 bit.

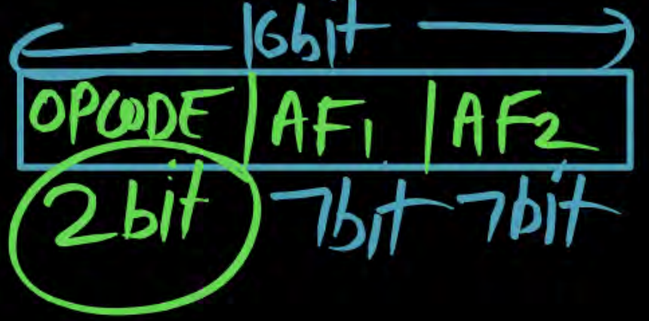
(i) If 2AF(2AI(Address Instruction) same size) is used then How many total number of operation supported (formulated)?

(ii) If 1AF (Address field) is used then how many total number of operation supported formulated?

Soln

(i) $128\ \text{Byte} \Rightarrow 2^7\ \text{Byte} \Rightarrow AF = 7\text{bit}$

Instruction Length = 16bit



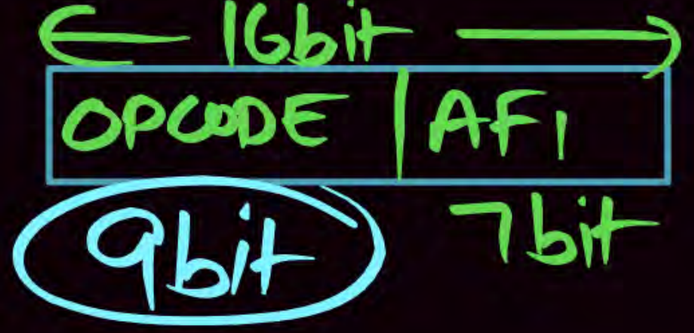
If opcode = 2bit

Total # operation $= 2^2$

$= 4$ operation.

Ans

(ii) LAF | LAI :



Total # operation = 2^9

= 512 operations. Ans

Q.2 A Hypothetical Processor support 100 different operation and 3 address memory field (same size). Instruction is stored in 1 MB memory. Then what is the length of the instruction?

Soln 2

Memory = 1MB (2^{20}B) = AF = 20bit

100 operation ⇒ opcode = 7bit

$\lceil \log_2 100 \rceil$ $2^n = 100$.

OPCODE	AF_1	AF_2	AF_3
7bit	20bit	20bit	20bit

Instn length = 7+20+20+20 = 67 bits Ans

THANK YOU!