

COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and
Addressing Modes

Machine Instruction

Lecture_01



Vijay Agarwal sir

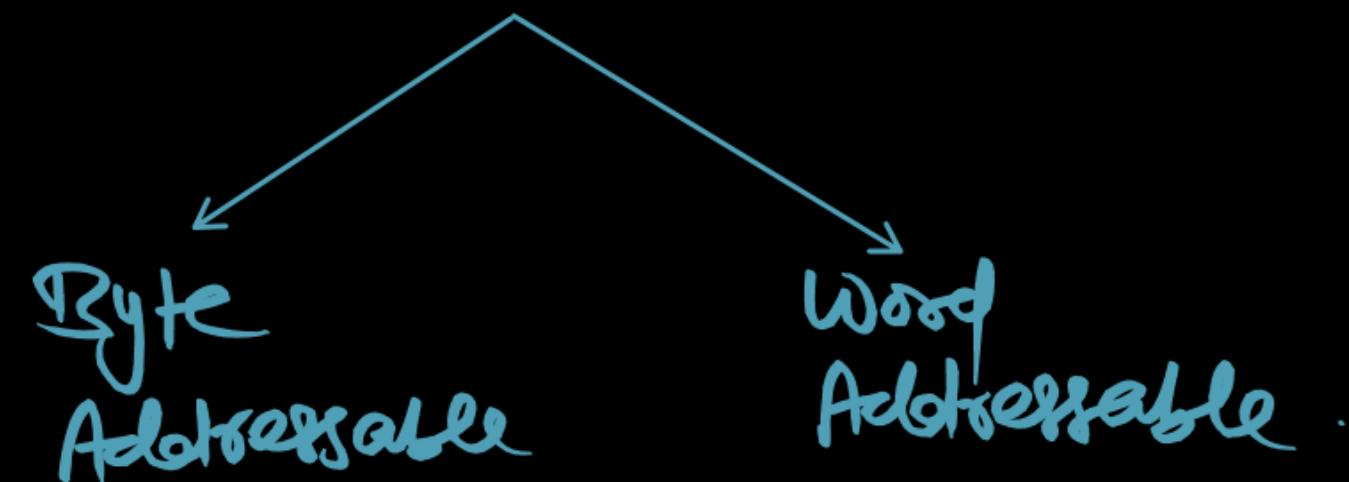


**TOPICS
TO BE
COVERED**

- o1 Machine Instruction**
- o2 Instruction Format**

Memory Concept

↳ Address line
↳ Data line



System Bus

- ① Address Bus → Capacity of the Memory.
- ② Data Bus → 'Word length' of the Processor.
- ③ Control Bus

In 8085

16 bit Address

AD₀ - AD₇, A₈ - A₁₅

Word length = 8 bit

Operation performed
on 8 bit Data format

In 8086

20 bit Address

AD₀ - AD₁₅ & A₁₆ - A₁₉

Word length = 16 bit

operation performed on
16 bit Data format.

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).

1 Byte = 8 bit

Word \Rightarrow Word Length
Size of Processor.

16 bit Processor

1 Word = 16 bit

1 Word = 2 Byte

Memory $\xrightarrow[\text{form}]{\text{Data}} \text{'Byte'}$
CPU (Processor) $\xrightarrow[\text{form}]{\text{Data in 'Word'}}$

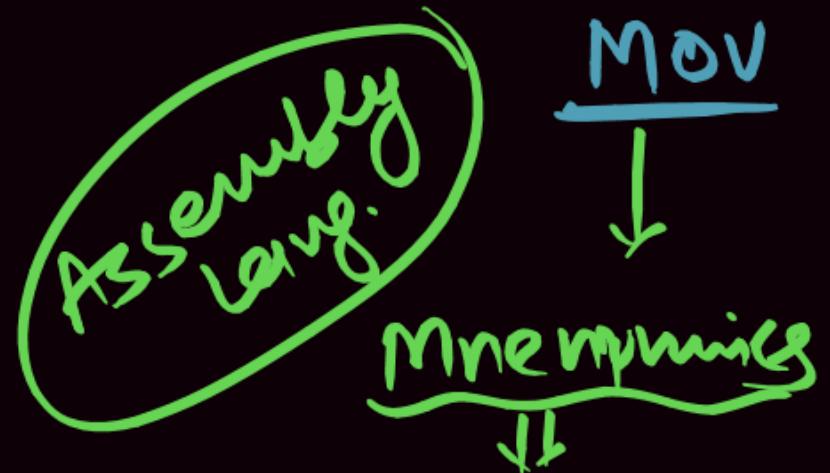
Double Word = 2 * 1 Word
= 2 * 2B
= 4 Byte

Note

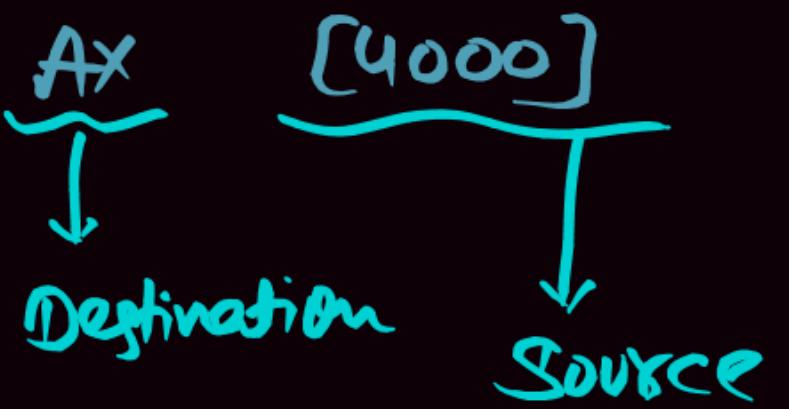
Default memory configuration is Byte Addressable. So in the Memory chip data is always stored Byte wise.

Note

In the Processor operations are always performed on Word Format. So When the Word Length of the Processor is greater than 8 bit (1Byte) then Multiple cell accessing is Required to fetch Data from Memory to CPU parallelly.



Type of
operation



In 8086 Processor

16 bit Processor

MOV Ax [4000]

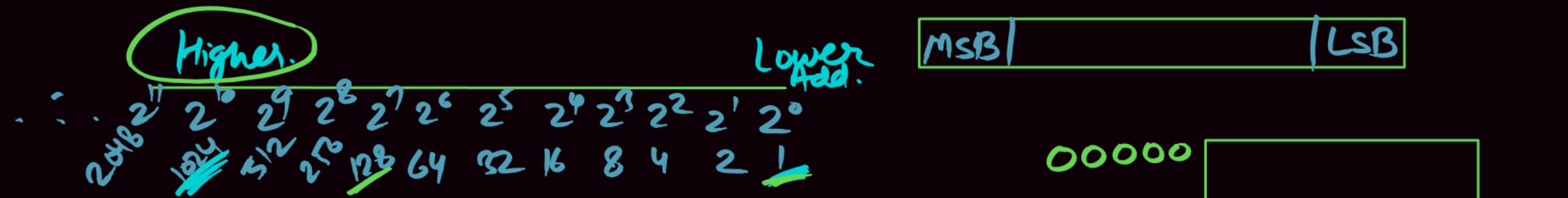
Ax \leftarrow $\begin{bmatrix} m[4000] \\ m[4001] \end{bmatrix}$

M: memory
[4000]: memory location 4000.

32 bit Processor

MOV γ_1 [4000]

$\gamma_1 \leftarrow \begin{bmatrix} m[4000] \\ m[4001] \\ m[4002] \\ m[4003] \end{bmatrix}$



Q.

16 bit Processor.

MOV Ax [4000]

(i) If M[4000] Contain Higher Byte .

<small>Higher</small>	<small>Lower</small>
4000	4001

Ax =

57	69
----	----

Ax = 5769 H Ans

(ii) If M[4000] Contain Lower Byte . OR

Ax =

4001	4000
69	57

Ax = 6957 H Ans

4000	57 H	[0101 0111]	7
4001	69 H	[0110 1001]	6
4002	45 H	[0100 0101]	5
4003	20 H	[0010 0000]	4

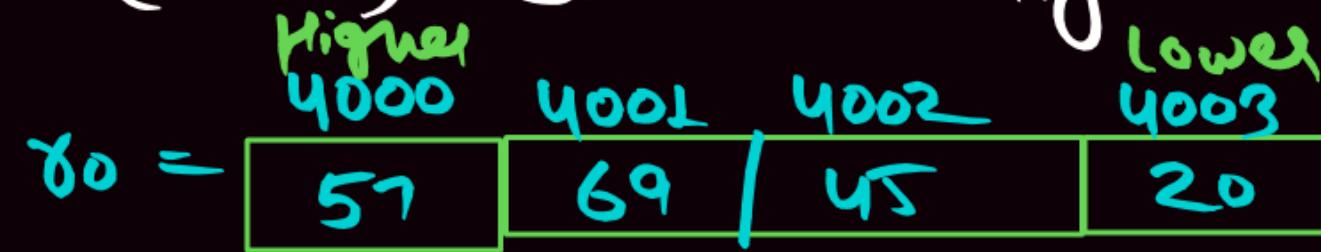
FFFF

	Higher.	Lower Add.	MSB	LSB
$2^{11} \dots 2^0$	$2^9 \dots 2^0$	$2^8 \dots 2^0$	00000	

32 bit Processor.

MOV X_0 [4000]

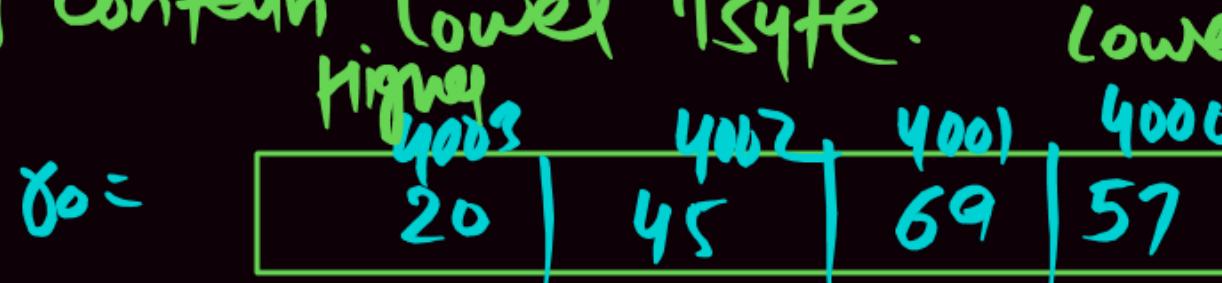
(i) If $M[4000]$ Contain Higher Byte .



$$\text{X}_0 = 57\ 69\ 45\ 20\text{H}$$

Avg

(ii) If $M[4000]$ Contain lower TByte .



OR FFFFF

$$\text{X}_0 = 20\ 45\ 69\ 57\text{H}$$

∴

4000	57 H	[0101 0111]	5	7
4001	69 H	[0110 1001]	6	9
4002	45 H	[0100 0101]		
4003	20 H	[0010 0000]		

Here L Instruction Creates Two(2) outputs.

So there is Problem of Ambiguity.

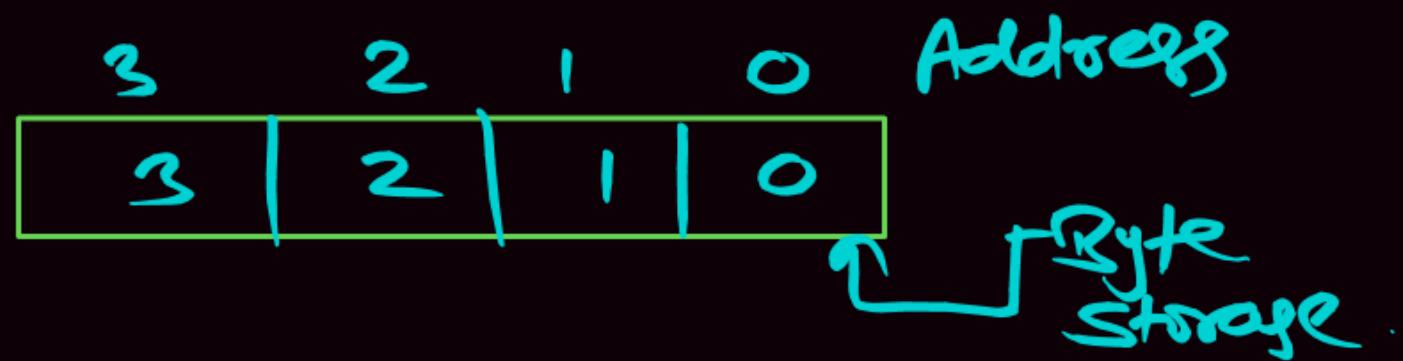
To Handle this Situation there is Need of Memory Address Interpretation Mechanism Called Endian Mechanism.

• Endian Mechanism Show the Order of Data Storage in the Memory.

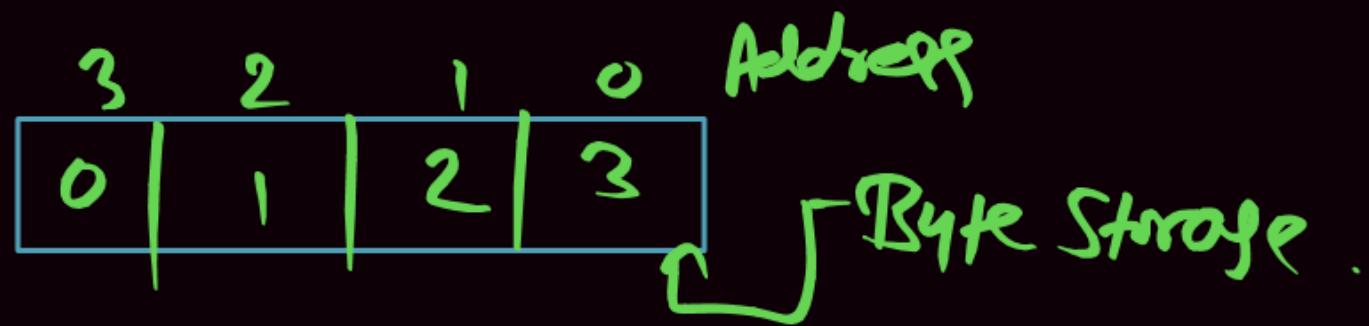
① Little Endian

② Big Endian

① Little Endian : Lower Addresses Contain Lower Byte & Higher Addresses Contain Higher Byte.



② Big Endian : Lower Addresses Contain Higher Byte & Higher Addresses Contain Lower Byte.



- Mostly Processors Use Little Endian & Some Processor Use Big Endian

Note

Default Mc is Little Endian Used in Processor design.

69 57

Little
[69 | 57]

(Q1)

AX =

4001	4000
69	57

AX = 69 57 H

Big

[57 69]

(Q2)

X0 =

4003	4002	4001	4000
20	45	69	57

X0 = 20 45 69 57 H

Little Endian : Right to left.

Big Endian : Left to Right.

In My Computer Properties.

- 4GB RAM

32 bit Address line

- 64 bit Processor

Word length = 64 bit

- ✓ 64 bit Processor
- ✓ 4GB RAM

4 GHz Processor

7TB Hard disk

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.

32bit
Data

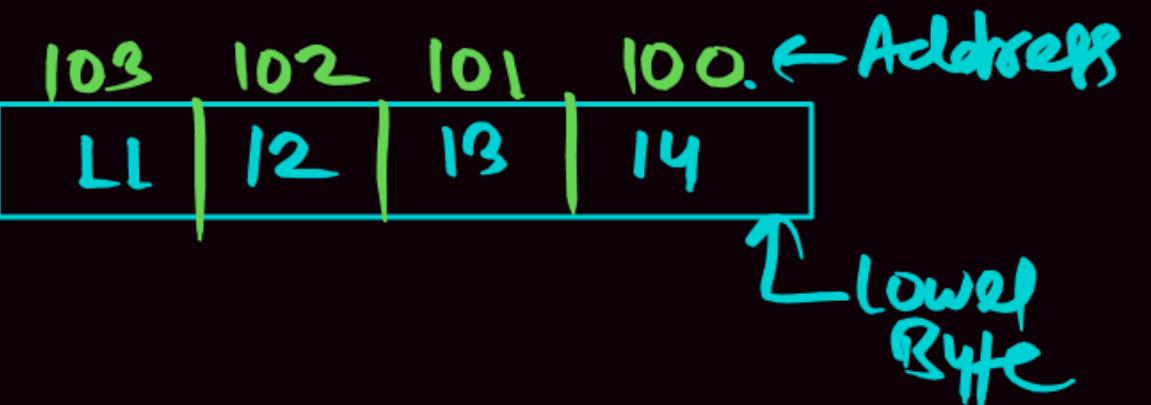
[0001 0001 0001 0010 0001 0011 0001 0100]

⑧ (LL 12 13 14) H.

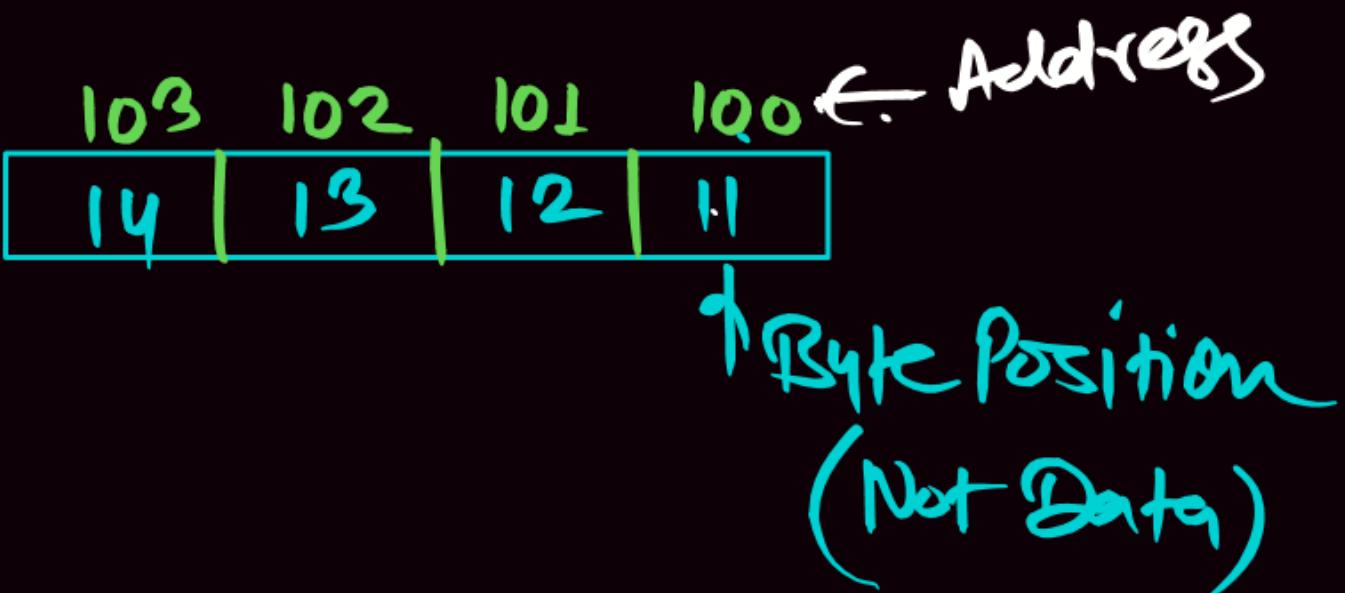
Load in the Memory location 100.
Starting location 100.

32bit Processor

(i) Little Endian



(ii) Big Endian

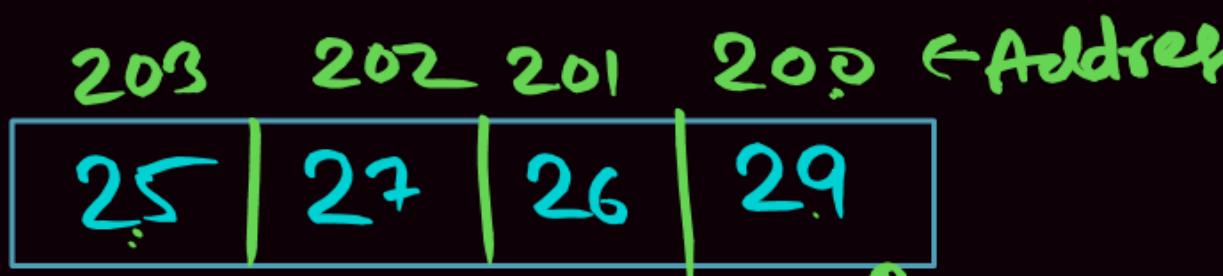


0010 0101 0010 0111 0010 0110 0010 1001L

eg (25 27 26 29)H
Higher Byte Position
Lower Byte Position

32 bit Processor Starting Address 200.

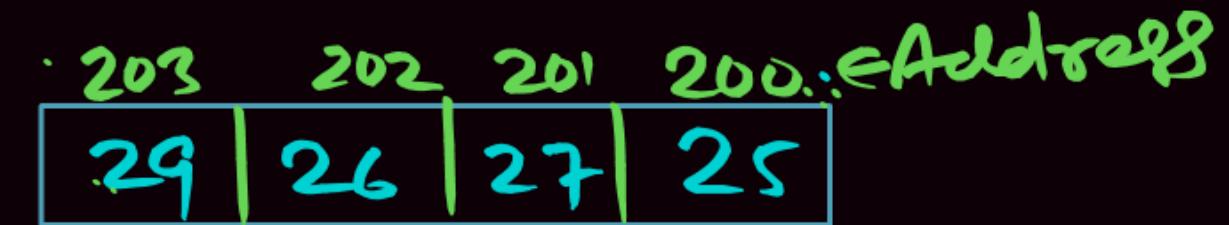
(i) Little Endian



Byte Position

200	29H
201	26H
202	27H
203	25H

(ii) Big Endian

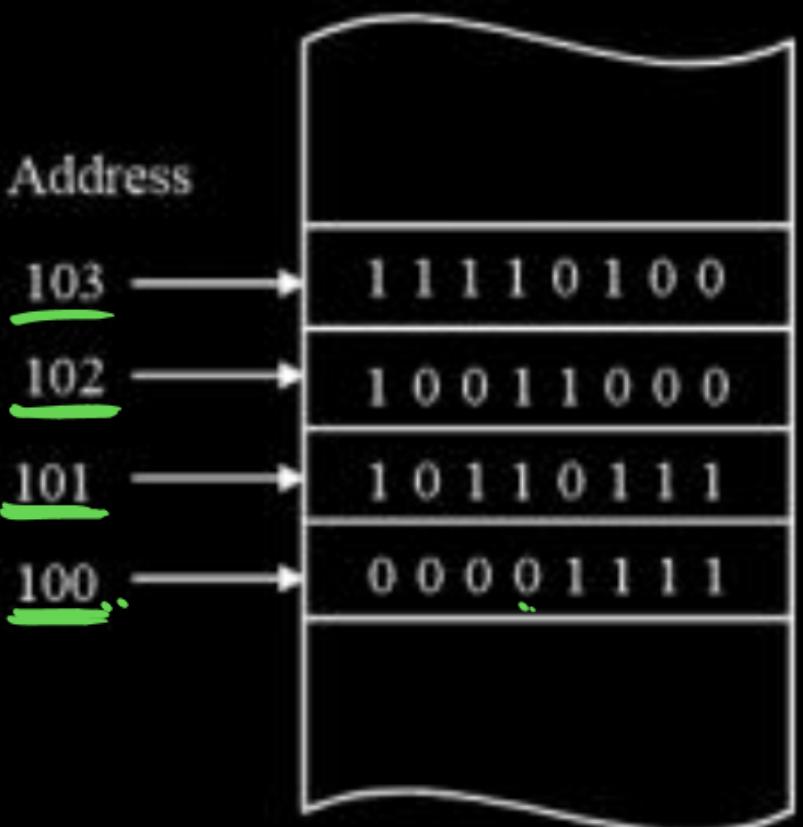


200	25H
201	27H
202	26H
203	29H

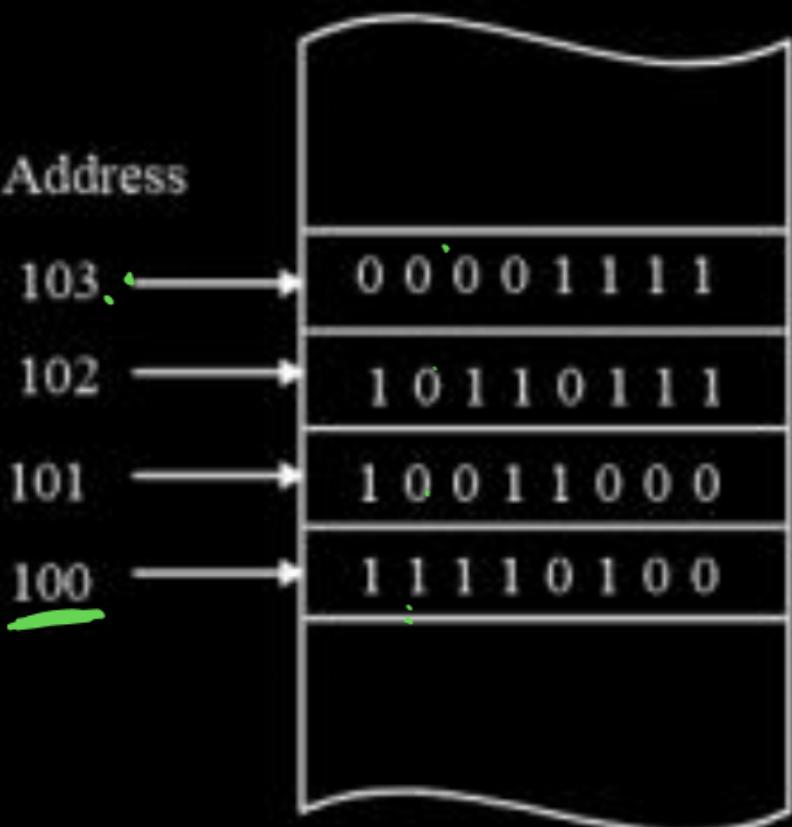
Two Important Memory Design Issues



(a) 32-bit data



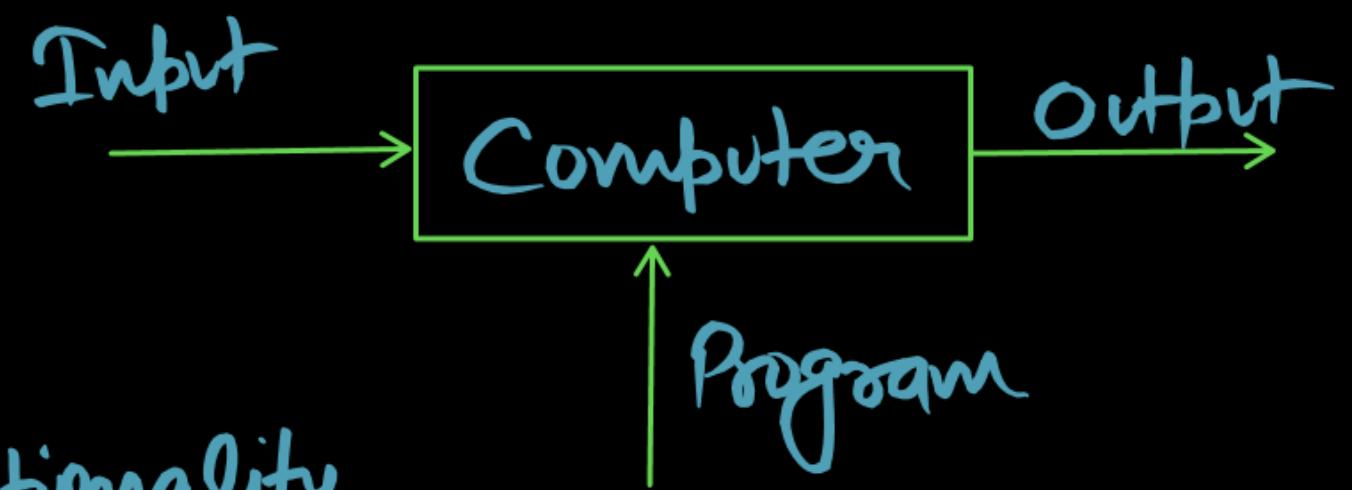
(b) Little-endian byte ordering



(c) Big endian byte ordering

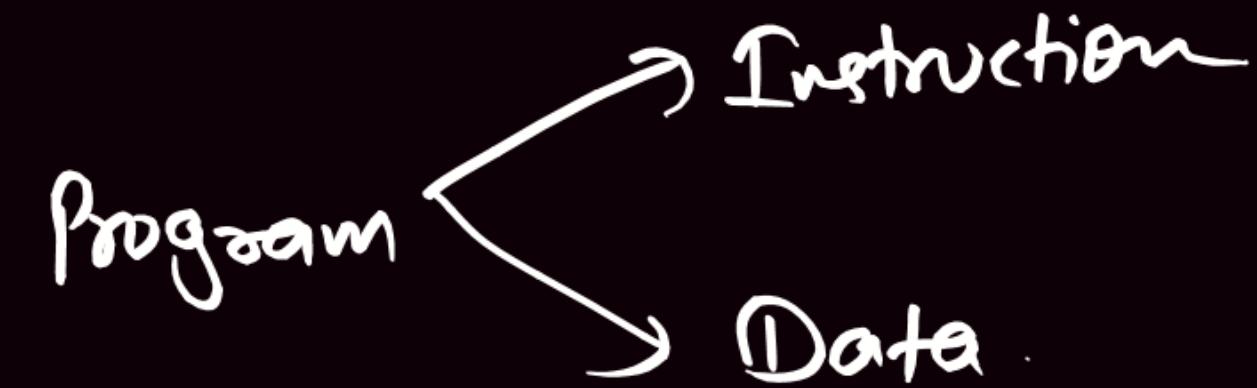
Two byte ordering schemes commonly used by computer systems.

Computer : Computer is a Computational Devices Used to Process the Data Under the Control of Program.



Computer functionality
is program execution.

Program: Program is sequence of Instruction along with the Data.



① Instruction: It is a binary Sequence (Code) which is designed inside the Processor to perform the operation.
② → ADD

OR

Binary bits - Bind with - operation.

Instructions are sequence of bits to instruct the Computer to perform the operation.

② Data : Data is a binary sequence bind with value.

(Data format eg BCD, Decimal)
Hexa Decimal.

Binary Sequence — Bind with — Value.

② \Rightarrow Decim: 2

Hex \Rightarrow : 0010

Instruction



↓

(e)

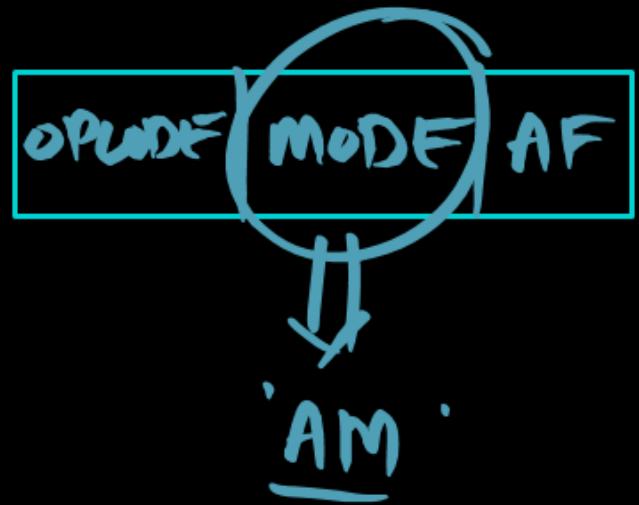
Addition : ADD

Multiplication: MUL OPCODE \Rightarrow operational Code

↳ Type of operation.

OPERAND : 'DATA'

AF: Address of the
operand (DATA).





00 → ADD
01 → Increment
10 → SUB
11 → MUL

4 operation

If 2 bit opcode. \Rightarrow Total Number of operation = $2^2 = 4$ operation/
Instn

Note

n bit Opcode Can Perform 2^n operation

Note

If Memory size is given then we can calculate Address field

Q) Memory = LMB \Rightarrow Address = 20bit Avg

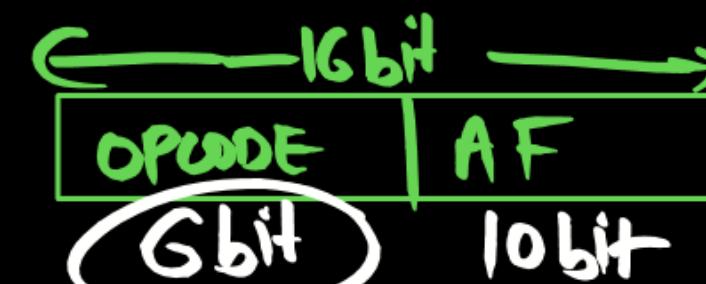
Q



Instruction Size is 16 bit & Address field is 10 bit then

How Many Number of operations Supported by the System?

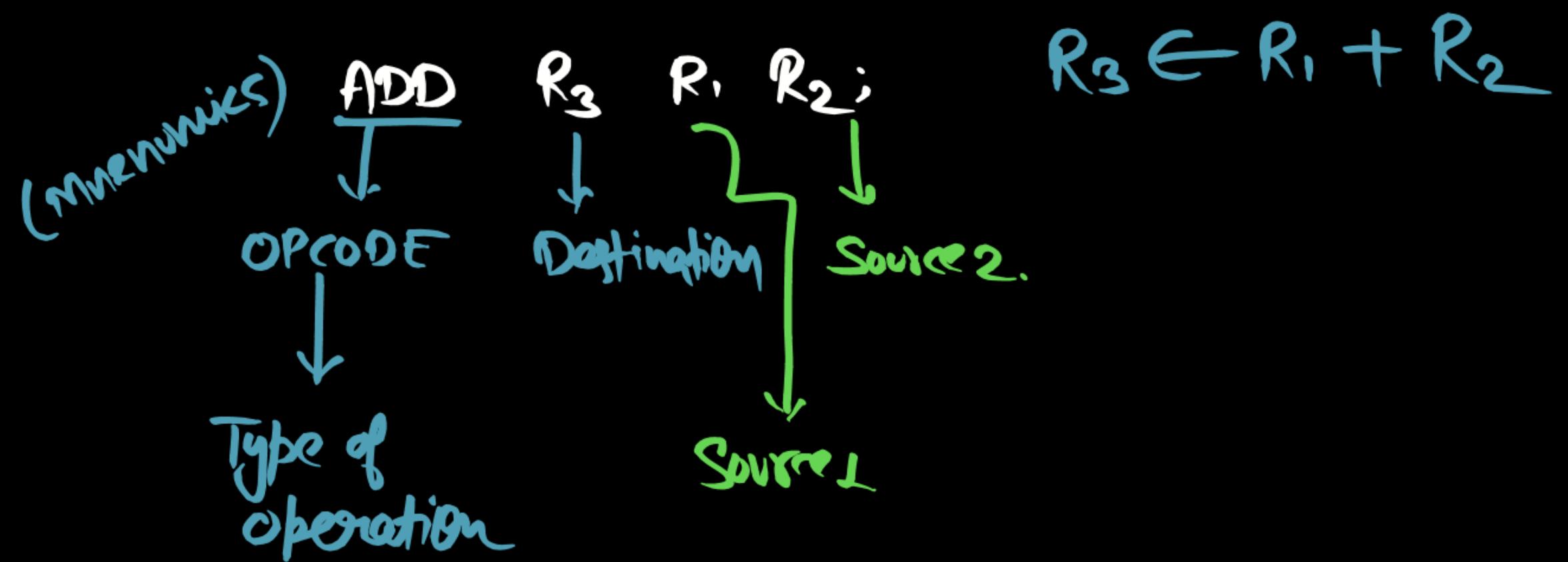
Soln



$$\text{Opcode} = 16 - 10 = 6 \text{ bit}$$

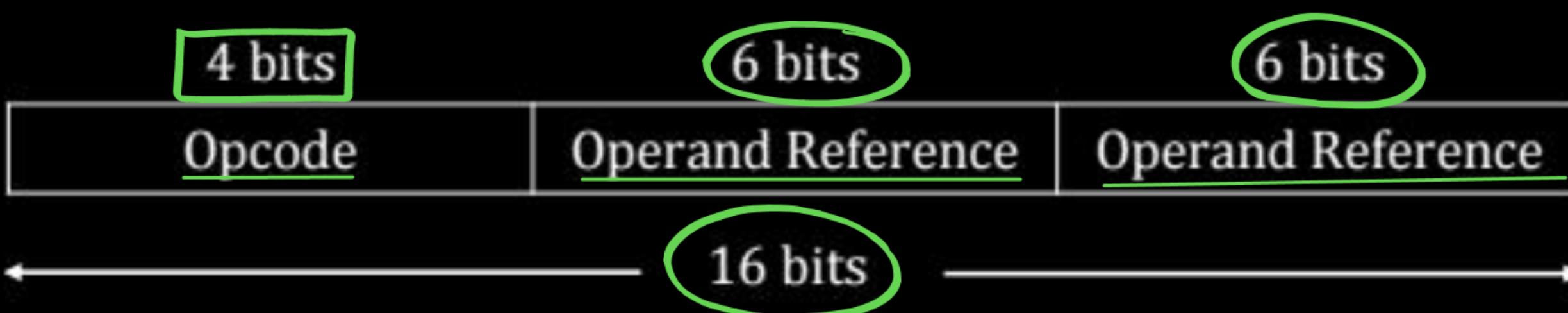
Total # operation = $2^6 = 64$ operation.

OPCODE	Dest	Source 1	Source 2
--------	------	----------	----------



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



**THANK
YOU!**

