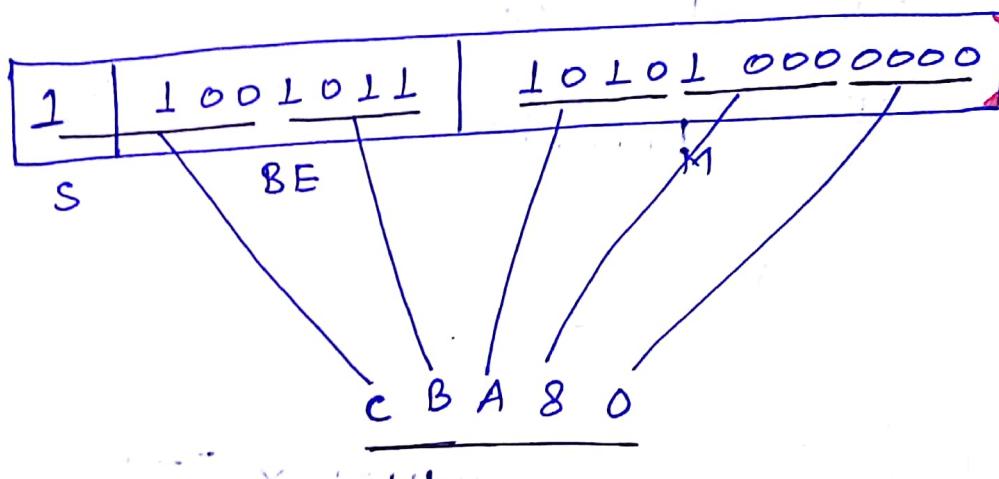
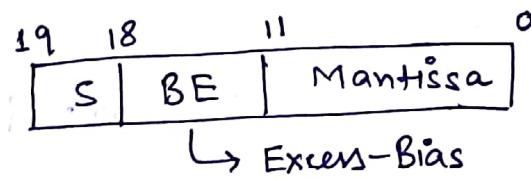


$$\text{so, } BE = AE + \text{bias}$$

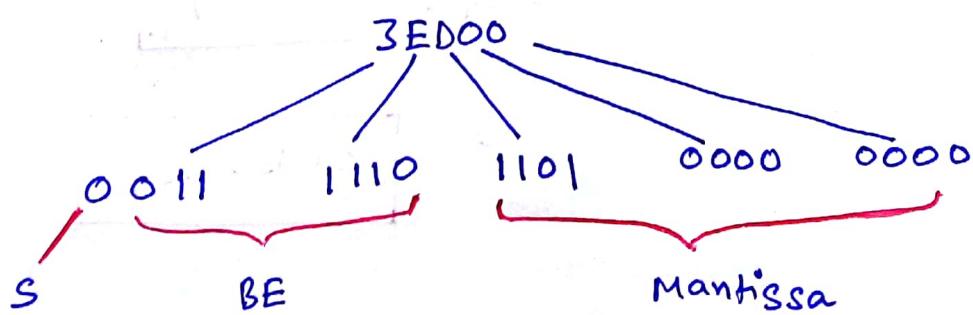
$$\begin{array}{r}
 AE : \quad 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \\
 + \text{ bias} : \quad 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 \hline
 BE : \quad 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1
 \end{array}$$



Q. Data Format



memory Data 0x 3ED00. What is the decimal value associated with the above memory data.



BE is of 7-bits so,

$$\text{Excess Bias} = 2^{n-1} = 2^{7-1} = 2^6 = \underline{\underline{64}} \\ = (1000000)_2$$

so, $\underline{AE} = BE - bias$

BE : 0111110

bias : 1000000

(BE < bias)

AE : 1111110

so, AE = -2

so, Binary value = $0.11101 \times 2^{-2+1}$
= (Prefixed (0.011101))₂

$$= 0.11101 \times \frac{1}{8} + \frac{1}{16} + \frac{1}{64} = \frac{8+4+1}{64} = \frac{13}{64}$$

= $\boxed{(0.011101)}_2$

ans = $\boxed{(0.453125)}_{10}$

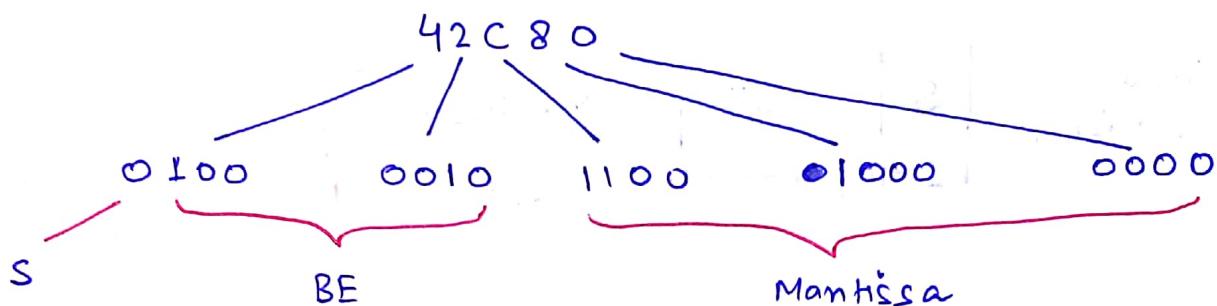
Q:

Format :

S	BE	Mantissa
1	7	12

, memory

data = $(42C80)_H$. what is the Equivalent decimal value ?



$$\text{Normal bias} = +2^{n-1} - 1 = +2^6 - 1 = +63$$

$$\text{BE} = (100\ 001\ 0)_2 = (66)_{10}$$

$$\begin{aligned} \text{so, } AE &= BE - \text{bias} \\ &= 66 - 63 = +3 \end{aligned}$$

$$\text{Mantissa} = (0.11001)_2$$

$$\text{so data} = +0.11001 \times 2^3 (1 \cdot M \times 2^e)$$

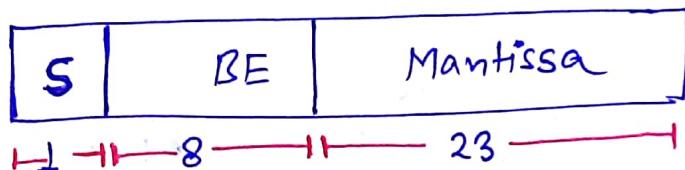
$$= (110.01)_2 \quad \frac{1}{4} = 0.25$$

$$= (14.25)_{10}$$

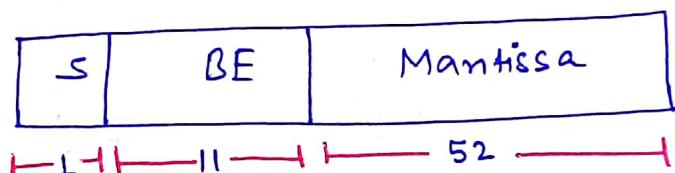
Range of Floating point data :-

- Range always depends on the format of floating point data. Let us consider 32-bits (single-precision) format to analyse its range.

single precision



Double precision



(i) Sign bit - $\begin{cases} 0 & (+ve) \\ 1 & (-ve) \end{cases}$

(ii) Exponent range : depends on BE field

$$\text{bias} = \text{Normal}$$

$$= + (2^{n-1} - 1)$$

$$= +(2^7 - 1) = +127 \quad (32\text{-bit format})$$

$$\text{bias} = + (2^{10} - 1) = +1023 \quad (64\text{-bit format})$$

so, Min BE is All 0's in field i.e

$$BE = 0$$

$$\text{so, } AE = BE - \text{bias}$$

$$\text{so, } (AE_{\min} = 0 - (+127) = -127) \quad (\text{32-bit format})$$

$$(AE_{\min} = 0 - (+1023) = -1023) \quad (\text{64-bit format})$$

Now - Max BE is when all bits are 1's.

$$BE = 2^8 = 255 \quad (\text{32-bit format})$$

$$BE_{\max} = 255 \quad AE = 255 - (+127)$$

$$(AE_{\max} = 128) \quad (\text{32-bit format})$$

$$(AE_{\max} = 1024) \quad (\text{64-bit format})$$

so, Expo range is

$$\underline{-127 \text{ to } +128} \quad \text{32-bit format}$$

&

$$\underline{-1023 \text{ to } +1024} \quad \text{64-bit format}$$

(iii) Mantissa range -

$$M_{\min} = 0$$

i.e. 1.0 or 1 ($M_{\max} = 1$)

and $M_{\max} = .111\dots1(23)$

~~$= 2^{24}-1$~~
i.e. $1.M = 1.111\dots1(22)$

so, $111\dots1(24) \times 2^{-23}$

i.e. $(2^{24}-1) \times 2^{-23}$

so, $(M_{\max} = 2 - 2^{-23})$

so, Mantissa range is -

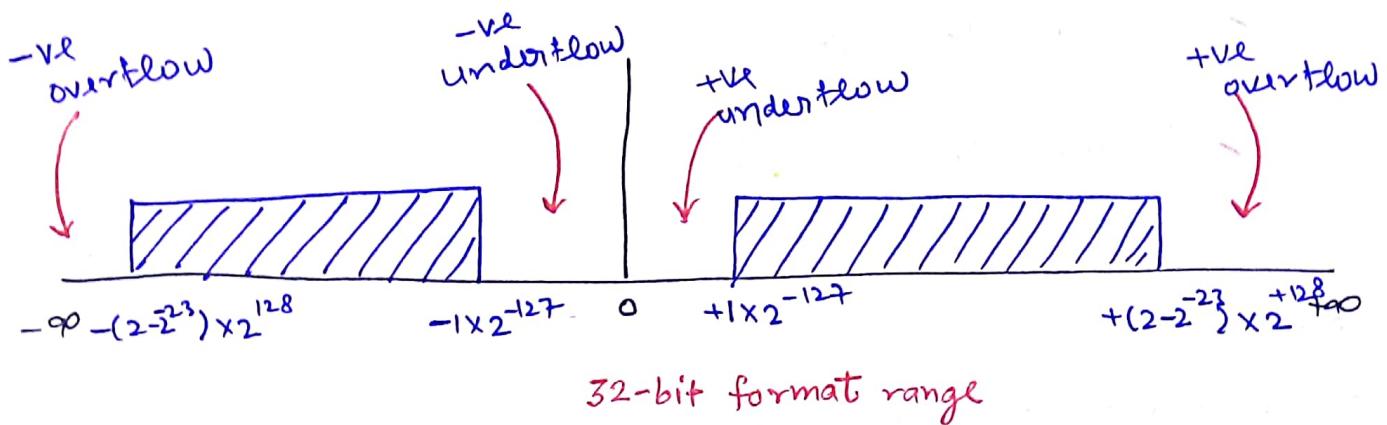
1 to $2 - 2^{-23}$

32-bit format

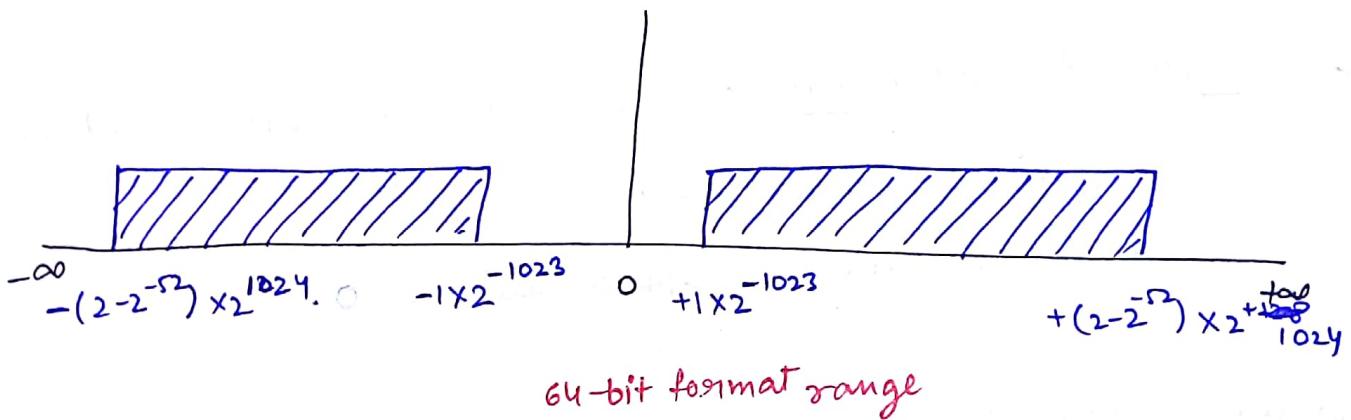
1 to $2 - 2^{-52}$

64-bit format

Now, Range of floating point data :-



32-bit format range



NOTE : In the floating point format special values are defined to handle the overflow & underflow conditions, name as $(0, +\infty, -\infty)$. Two exponents are reserved to define the special values i.e $(e_{\min} (\text{ all 0's in BE})$ and $e_{\max} (\text{ All 1's in BE})$).

Rounding techniques are used to report the overflow & underflow conditions , name as :

- Round to 0
- Round to $+\infty$
- Round to $-\infty$
- Round to nearest

Exponents

- Exponent overflow occurs when exponent exceeding the maximum possible +ve exponent. It becomes real when data approached to $+\infty$ and $-\infty$.
- Exponent underflow occurs when exponent exceeding the maximum possible -ve exponent. It become real when data is approached to 0.

Special value 0 :-

S	BE	Mantissa	value
0/1	All 0's	All 0's	0

to

0	0000000	00000...0(23)
"	"	
0	0000000	11111...1(23)

represent
as 0

$2^{23}-1$ are
used
for +ve
values

1	00000000	00000 - - - 0(23) → represent as 0
"	"	to
+	00000000	111111 - - - .1(23)

so, $\pm (1.0) \times 2^{-127}$



de-normalization



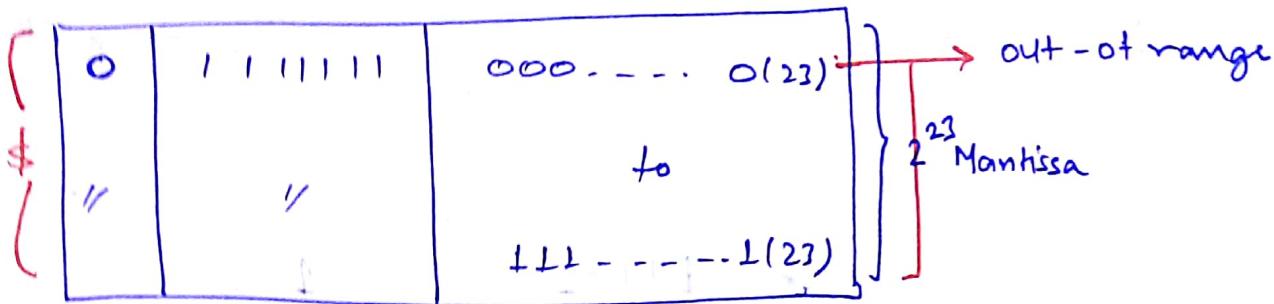
$\boxed{\pm (0.1) \times 2^{-126}}$

is the smallest fraction in (32-bit format)

used to represent 0.

in 32-bit format

S	BE	M	value
0	All 1's	All 0's	$+ \infty$
1	All 1's	All 0's	$- \infty$
0/1	All 1's	$\neq 0$	NAN



as $AE = +128$

and ~~but~~ normal $M = 1 \cdot M$
 $= 1 \cdot 0$

$\pm 1 \cdot 0 * 2^{+128}$

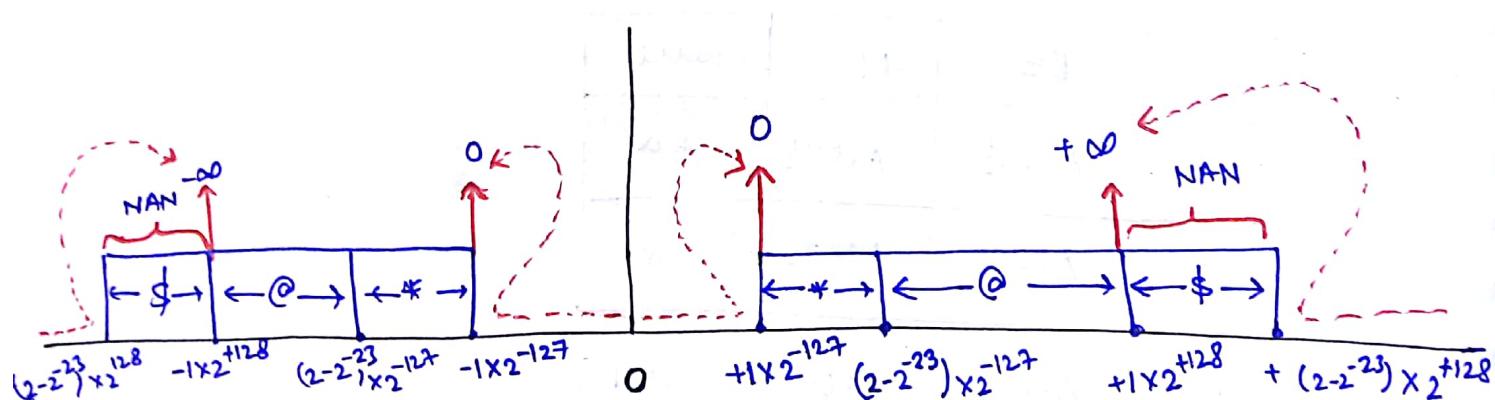


de-normalization



$0 \cdot 1 * 2^{+129}$

, out of range



* Indicates MIN expo data with all possible mantissa

\$ Indicates MAX expo data with all possible mantissa

@ Indicates the data existed b/w the MIN & MAX exponents with all possible mantissa

Eg:- 32-bit formatted data -

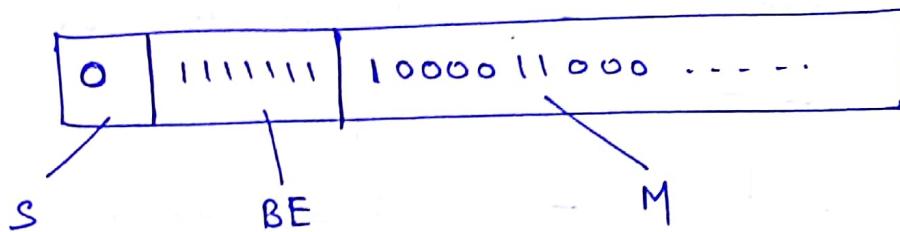
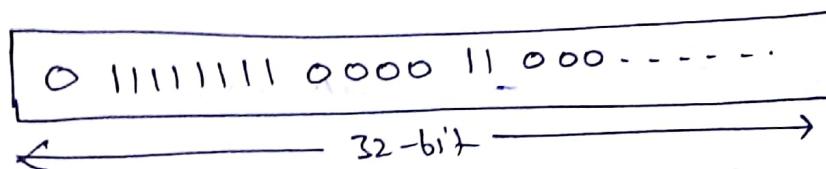
-	$+1.101 \times 2^{+128}$	-	NAN
-	$+1.000 \times 2^{+128}$	-	special value '+∞'
-	$+1.101 \times 2^{+130}$	-	Round to '+∞'
-	$+1.000 \times 2^{-127}$	-	special value 0
-	$+1.101 \times 2^{-129}$	-	Round to 0
-	$-1.000 \times 2^{+128}$	-	Special value '-∞'
-	$-1.1011 \times 2^{+132}$	-	Round to -∞
-	-1.000×2^{-127}	-	special value 0
-	-1.1011×2^{-130}	-	Round to 0
+	1.1011×2^{-127}	-	Round to nearest
+	$1.1011 \times 2^{+120}$	-	Round to nearest

Generalization :

S	BE	M
---	----	---

$$(\text{Data} = (-1)^S (1.M) * 2^{\text{BE-Bias}})$$

Q. Consider the following binary data stored in a 32-bit format. What is the value associated with the code?



$S: 0$
 $BE: \text{All } 1's$
 $M \neq 0$

}

NaN

Q. consider the following hypothetical format used to represent the data:

x	y	z
s	BE	M
		$\xrightarrow{\text{excess - } 512}$ bias

what is the value associated with the above code:

$$\text{Bias} = 2^{n-1} = 512$$

$$\text{so, } n-1 = \log_2 512$$

$$n = \log_2 512 + 1 \quad \text{no. of bits in BE}$$

$$(\text{data value} = (-1)^n (1 \cdot z) 2^{y-512})$$

- Q. Consider the following format used to represent the data

S	BE	M
1	8b	26

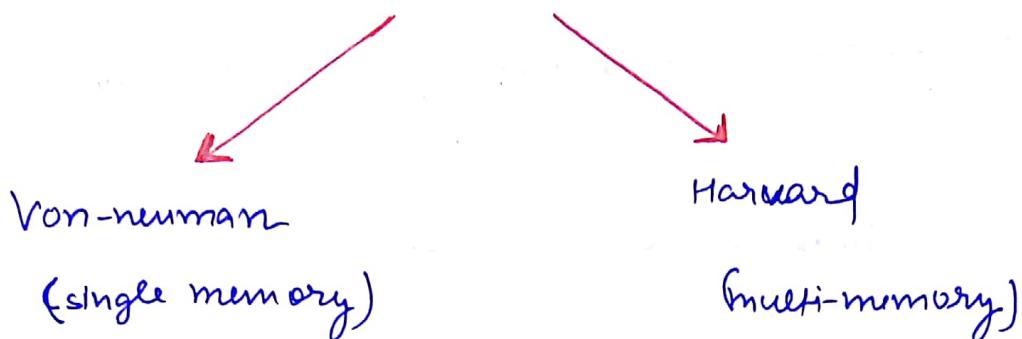
what is the range of a fraction possible in the system?

Range of fraction = Range of mantissa

$$= ((1 \text{ to } 2^{-26}))$$

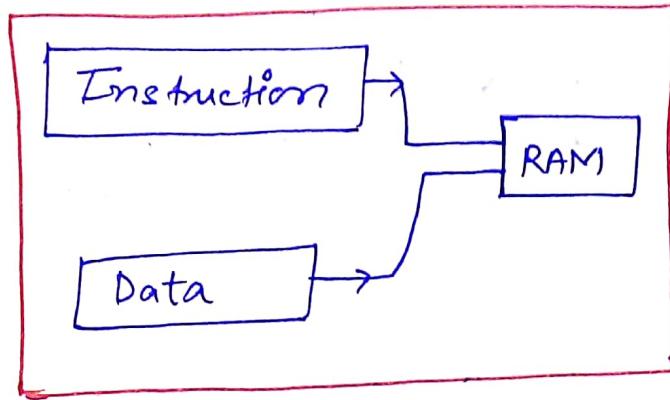
Computer Architecture:-

- Based on the program storage , computer Architecture is of two types.



Von-neuman Architecture

- In this Von-neuman architecture application program is always present in the main-memory for execution , so CPU generate physical address to execute the main memory part of the programs.
- Main Memory is volatile memory so it is flexible to load the different application programs according to the requirement. Therefore in this architecture different programs are executed on the same hardware.
- Both data & instruction are stored on the main memory



- Implemented as a uniprocessor design:

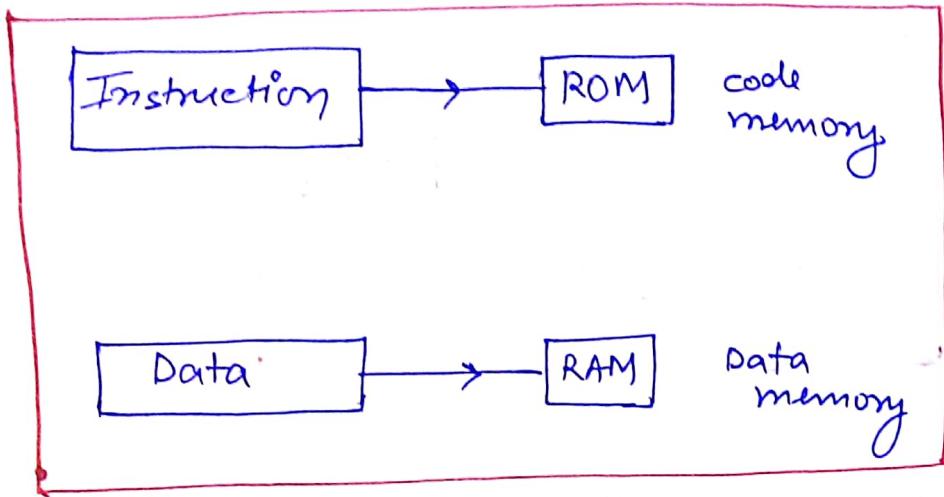
8085 microprocessor → 64 KB RAM

8086 MP → 1 MB RAM

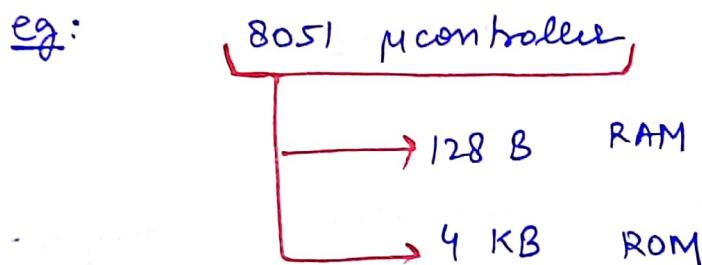
- used in the personal computer (PC) design.

Harvard architecture -

- In this, application programs are stored in the code memory and operational data is stored in data memory.
- Code memory is ROM and Data Memory is RAM.
- CPU always executes the pre-defined program in the code memory on a different data elements.



- Implemented as a μcontroller design



- used in intelligent system design eg. embedded systems.

Note: Computer organization subject describes the implementation of a von-neuman architecture so, CPU always generates the physical address.

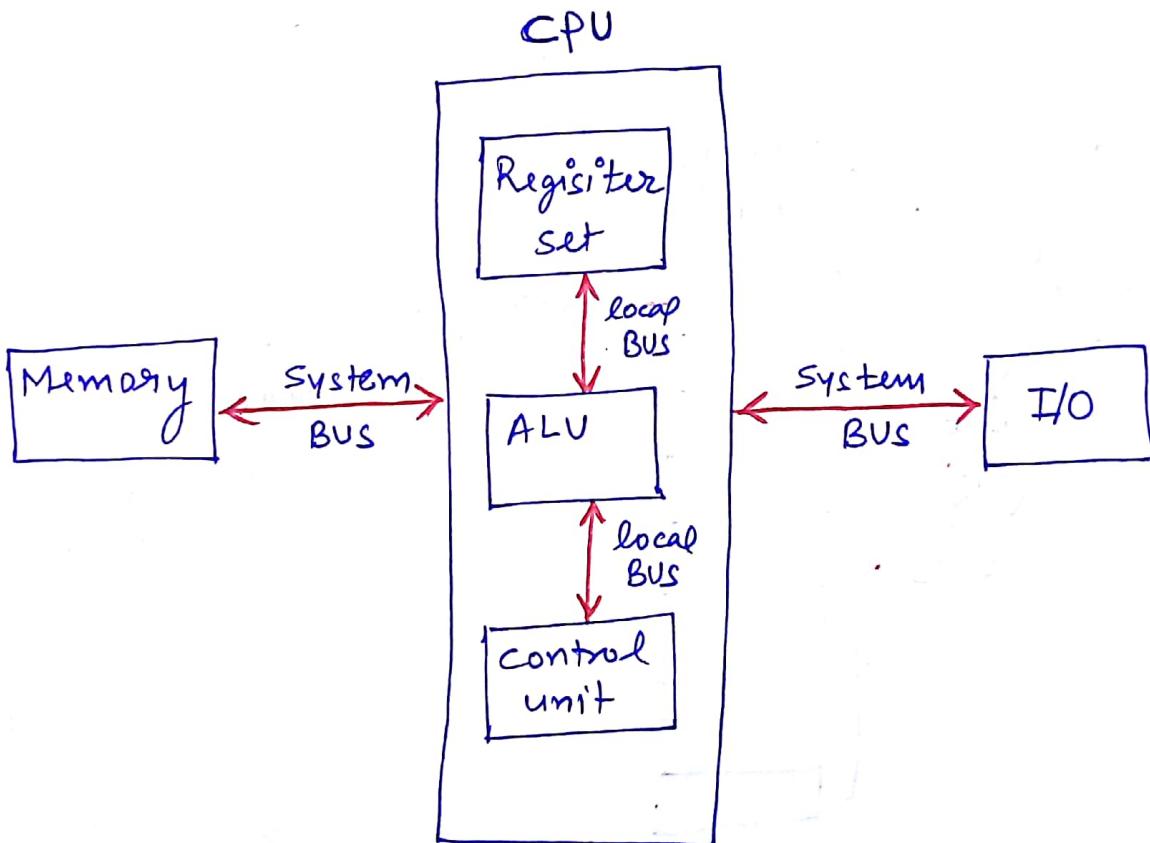
Block Diagram of a Computer System :-

- computer system contains three fundamental components :

(i) CPU (processing unit)

(ii) Memory (storage unit)

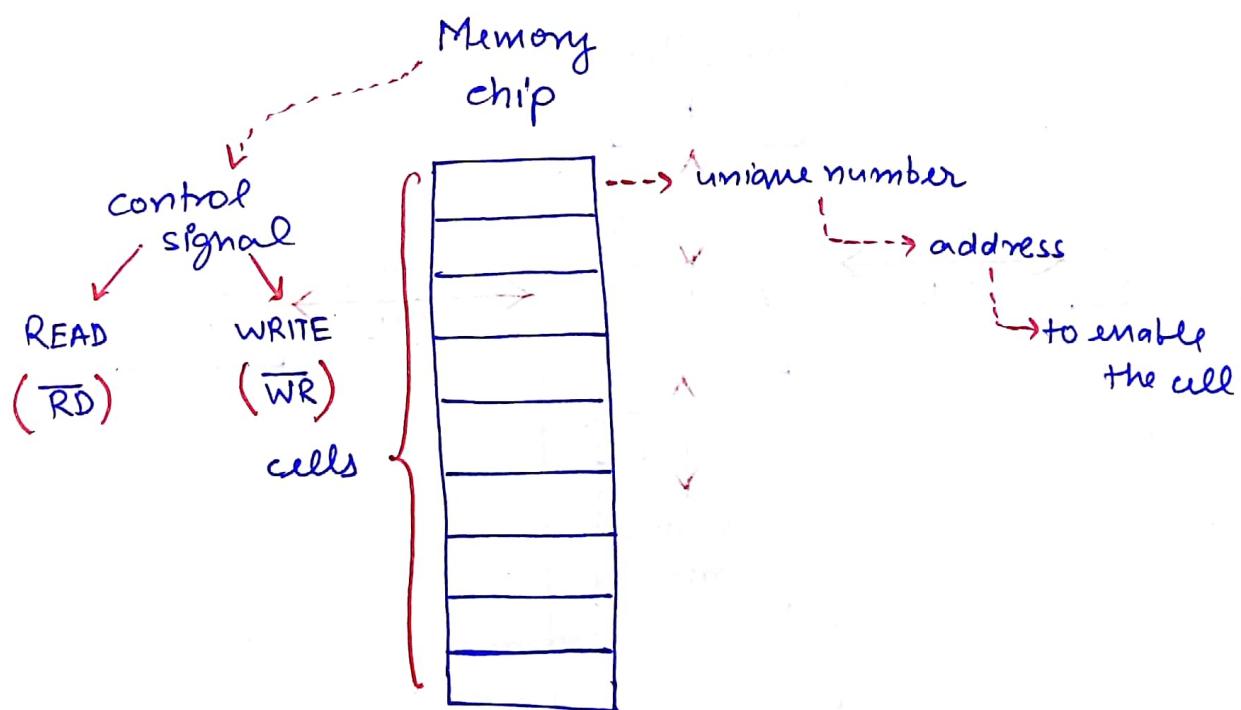
(iii) I/O (input-output devices) (External communication unit)



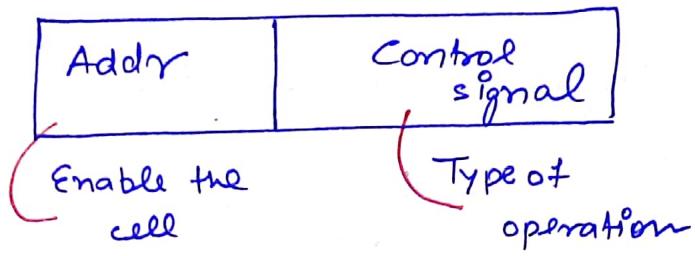
Memory Organization -

- It is a storage component
- Memory chips is organized into cells.
- Memory cells holds the data.
- Memory cells are identified with a unique number called as Address.
- Memory chips recognizes the control signals to indicate the type of operations.

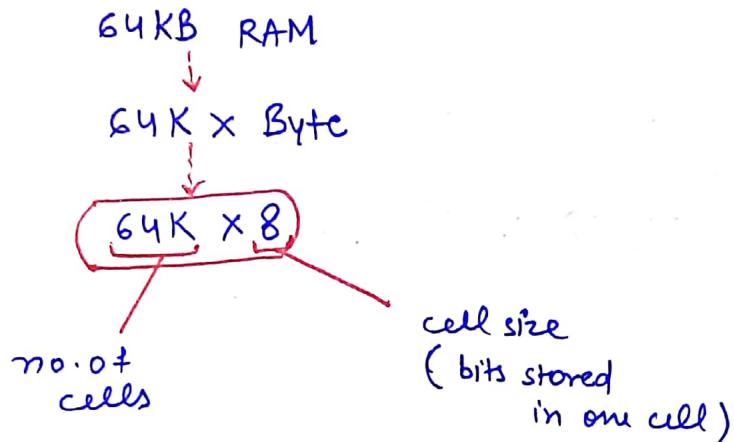
Q1:



- CPU generates the memory request, to access the program from the memory chip i.e



- Memory chip configuration is represented as -



$$\text{so, no. of cells} = 64K$$

$$\text{so, no. of address bits} = \log_2 64K = 16\text{-bit}$$

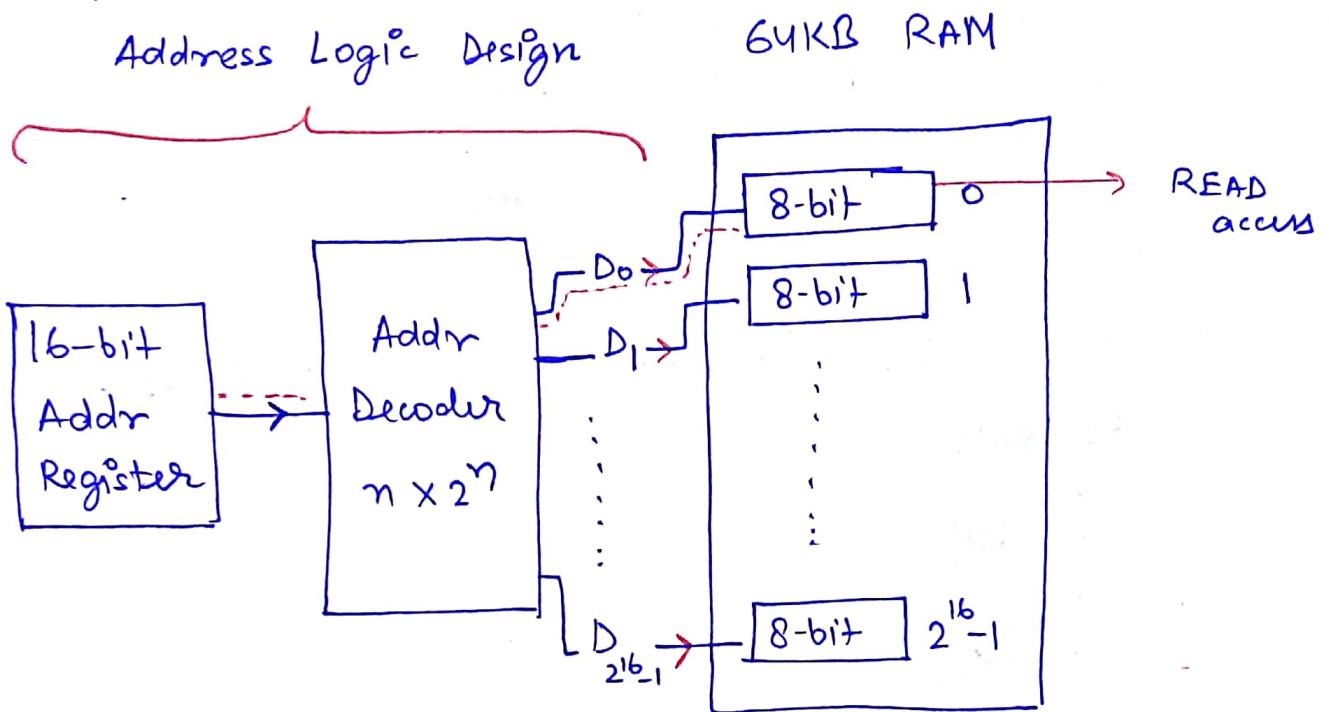
so, Address space of a chip is:

$$\begin{pmatrix} 0000 & 0000 & 0000 & 0000 \\ 1111 & 1111 & 1111 & 1111 \end{pmatrix}_2 \Leftrightarrow \begin{pmatrix} 0000 \\ FFFF \end{pmatrix}_{16}$$

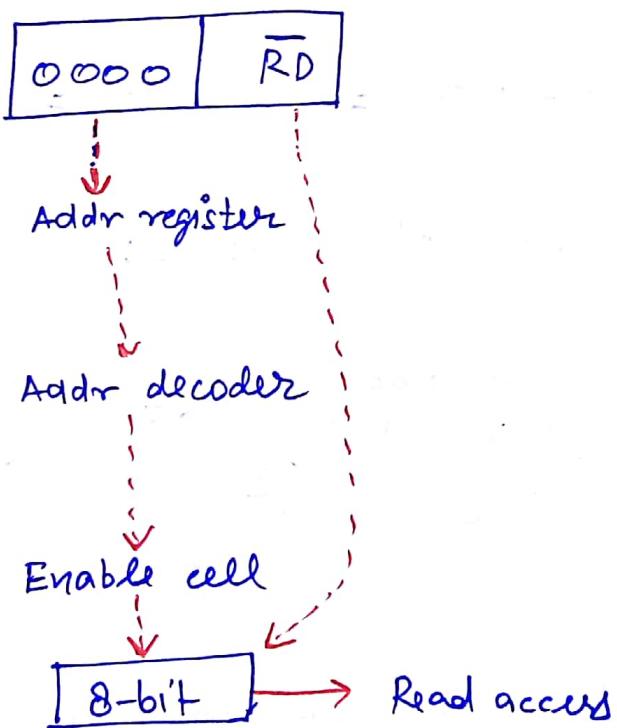
Conclusion:-

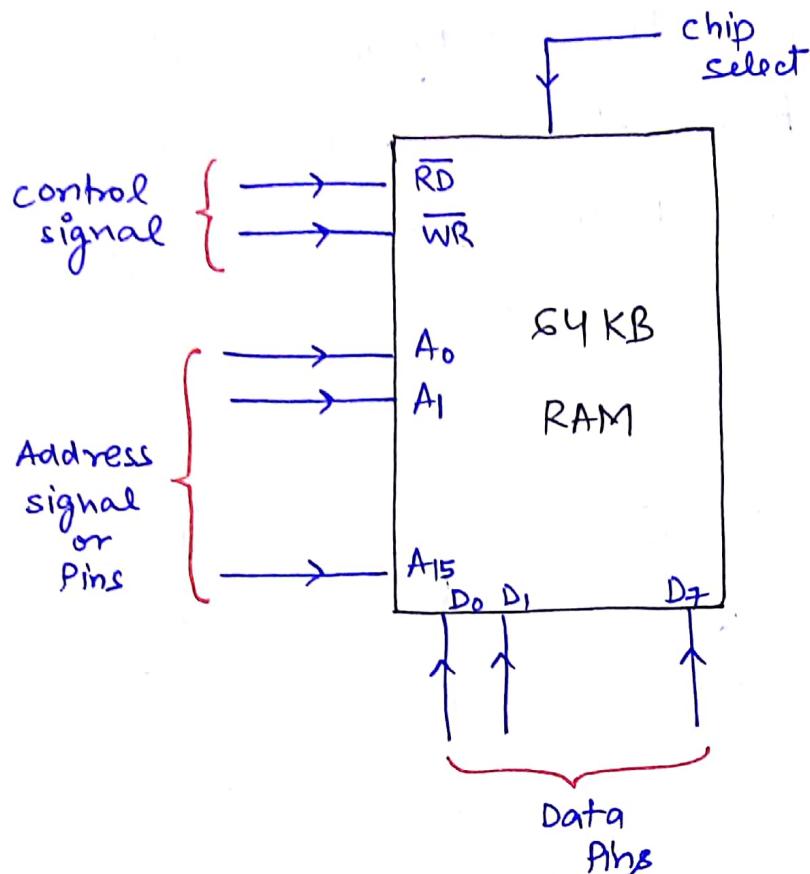
- 64K cells of mem chip contains . 8-bit cell space.
- 16-bit addressed cell contains 8-bit Data space.

Address Logic Design



Eg:- CPU generates the memory Request:





Hypothetical chips

$$2^n \times y$$

$$M \times n$$

[where M cells of
a memory chip contains
N-bit cell space]

[log M bit Addressed
cells contain N-bit
Data space.]

Notations:

Bit - switch (ON & OFF)

Nibble - 4-bits

Byte - 8-bits

Word - bits processed in a CPU

$$1K = 2^{10}$$

$$1M = 2^{20}$$

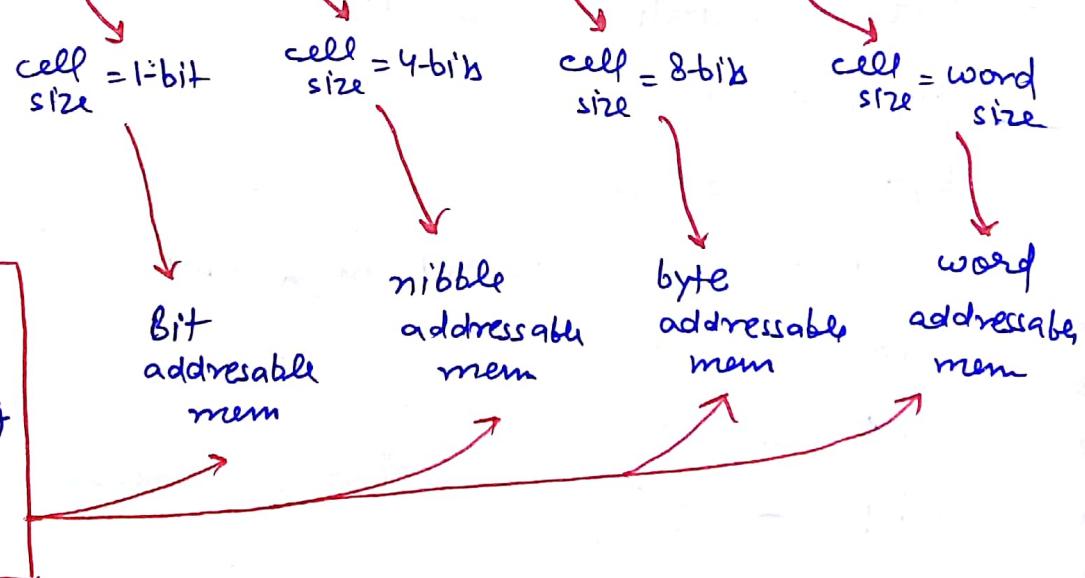
$$1G = 2^{30}$$

$$1T = 2^{40}$$

$$1P = 2^{50}$$

NOTE: In the computer system, data is always represented in a form of binary. Following referential units are used to refer the binary data.

CPU	Bit	Nibble	Byte (B)	word (w)
8-bit	1	4	8	8
16-bit	1	4	8	16
32-bit	1	4	8	32
n-bit	1	4	8	n



NOTE: In the CPU, operations are always performed

on a word formats. In the memory, data is always stored in a byte-wise sequence i.e byte addressable configuration.

- Byte addressable memory means address cell-size is 8-bits, whereas address size depends upon no. of cells in a chip.
- Word addressable memory means address size is depends on the no. of cells in the chip but the cell size is equal to word-size i.e processor word length.

Eg:-

4×8 → 2-bits Add

64×8 → 6-bits Add

256×8 → 8-bits Add

$1K \times 8$ → 10-bits Add

$2^n \times 8$ → n-bits Add

cell-size

Byte addressable
i.e 8-bits cell
space

$4 \times w$ → 2-bits Add

$64 \times w$ → 6-bits Add

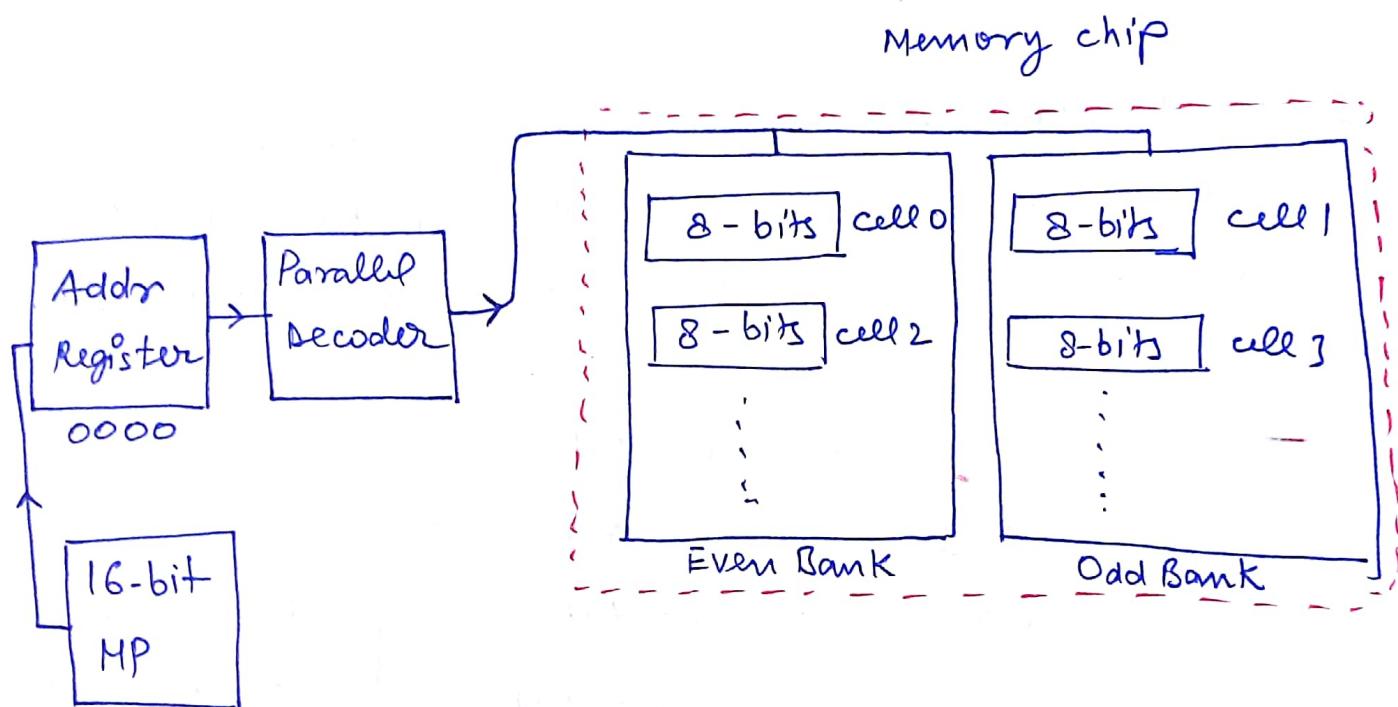
$1K \times w$ → 10-bits Add

$2^n \times w$ → n-bits Add

Word addressable
i.e cell size
= word size

Eg:- Consider 16-bit CPU which supports byte addressable memory then accessing sequence is:

So, 2 cells interfacing required to access the data in a word wise - sequence.



* So, when ~~16-bit~~ address came to decoder then decoder activates 1 cell from each bank or units.

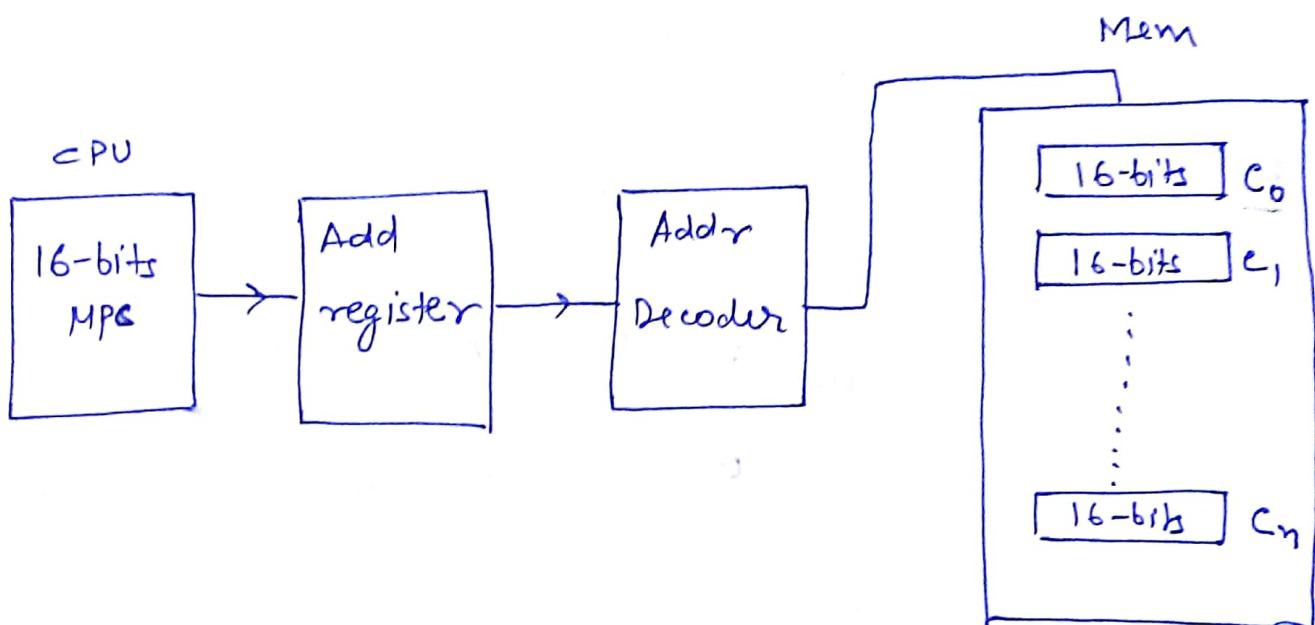
[storage sequence : Byte-wise]
[accessing sequence : Word-wise]

Eg:- Consider 16-bit CPU which supports word addressable memory then accessing sequence is :

CPU word size = 16-bits

Mem cell size = word size = 16-bits

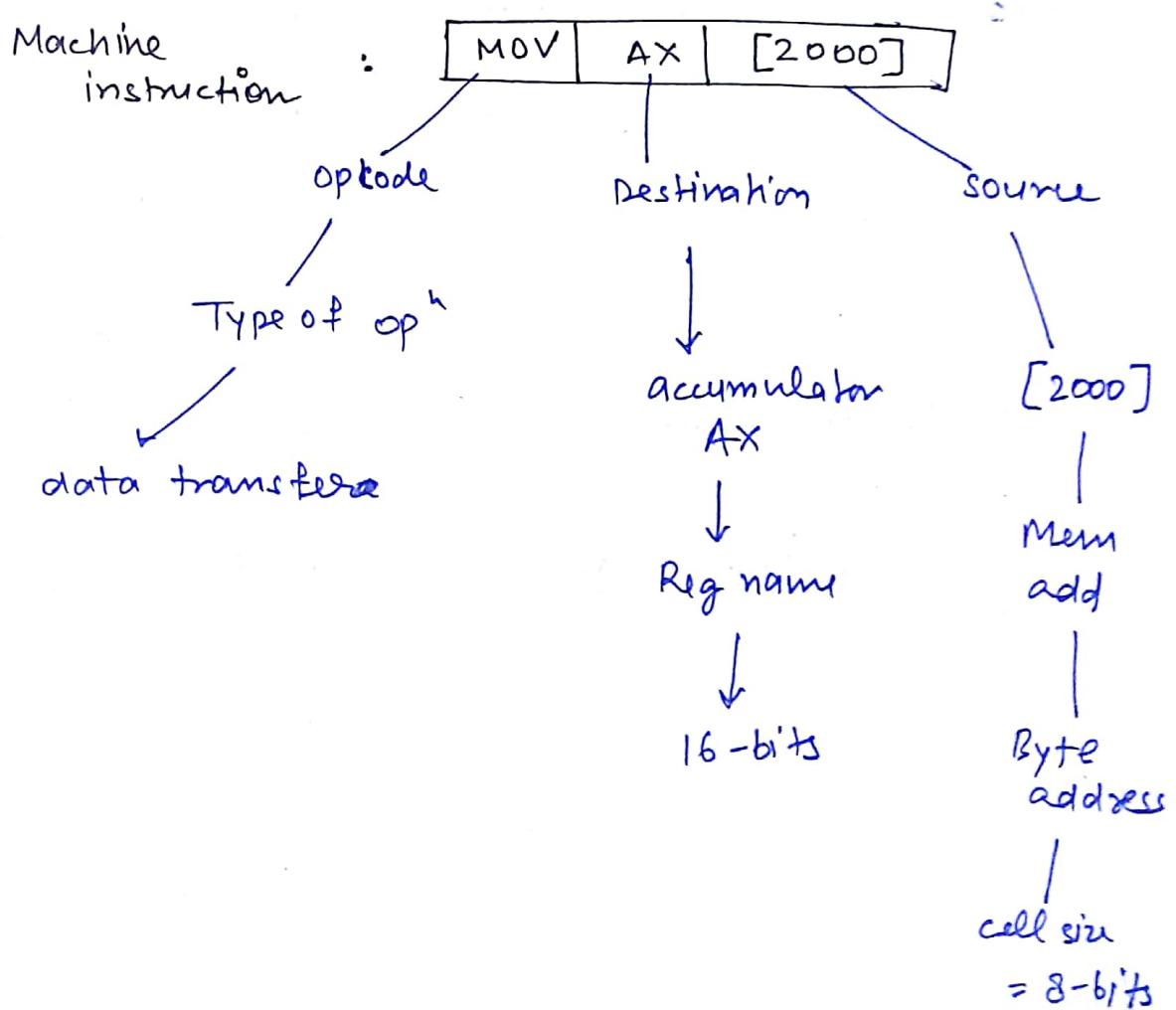
So, 1 cell interacting is required to access the data in a word wise sequence.

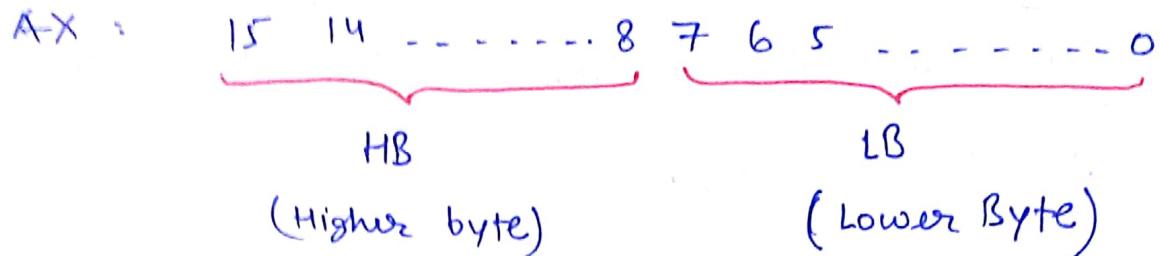
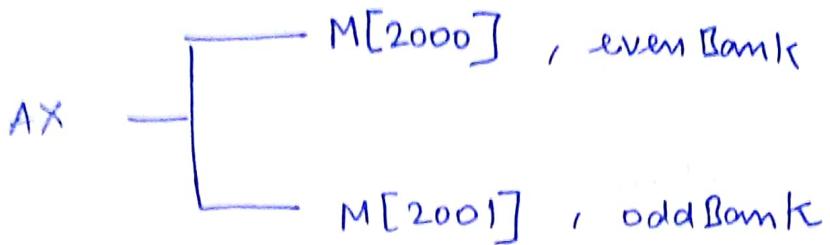


[storage sequence : word size]
[Accessing sequence : word wise]

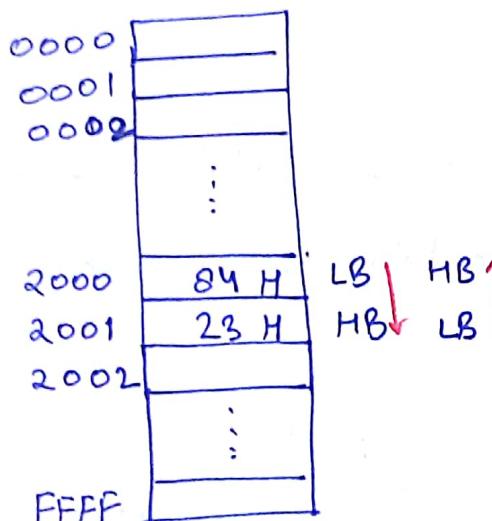
NOTE :- Default memory configuration is Byte addressable so data stored in memory is in Byte-wise sequence, whereas default CPU data type is word so operations are always performed on a word-wise data format so memory interfacing is adjusted by the designer based on the word-length of the CPU to access the data from the memory.

Eg:- 16-bit op (8086 CPU)





while execution of above instruction, two possible outcomes are generated due to lack of order of data storage in the memory.



Two outcomes are -

- 23 84 H
- 84 23 H

To handle the above ambiguity problem, memory address interpretation mechanisms (Endian-ness mechanism) are used in the memory design

(i) Little - Endian

(ii) Big - Endian

- In Little - Endian, lower address contains lower byte and higher address contains higher byte (ascending order).

(eg: 2384 H) as in

2000 and 2001
|
LB HB

6	5	4	3	2	1	0
6	5	4	3	2	1	0

: Addr

: Byte-position

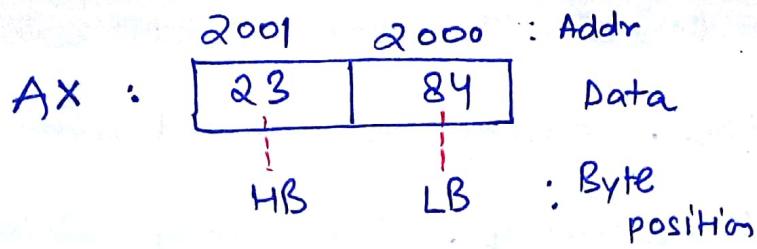
- In Big - Endian, lower address contains Higher byte and lower address contains higher byte.

6	5	4	3	2	1	0
0	1	2	3	4	5	6

: Addr

: Byte-position

NOTE: Default address interpretation mechanism is Little Endian.



$$\boxed{AX = 2384 \text{ H}}$$

Q. Consider 32-bit hypothetical CPU which supports 256KB memory space. System is enhanced with a word addressable memory. How many address pins are saved in the Enhanced CPU.

$$\text{Data} = 256 \text{ KB} \quad \text{and word size} = 32\text{-bits}$$

$$\begin{aligned}
 \text{old memory cell size} &= 256 \text{ K} \times 8 \text{ bits} \\
 &= 64 \text{ K} \times 32 \text{ bits}
 \end{aligned}$$

$$\begin{aligned}
 \text{So, Address size} &= \log_2 64\text{K} = 16 \text{ bits} \\
 &\text{in enhanced } \cancel{\text{mem}}
 \end{aligned}$$

$$\begin{aligned}
 \text{Address size} &= \log_2 256\text{K} = 18\text{-bits} \\
 \text{in old } \cancel{\text{mem}} &
 \end{aligned}$$

$$\begin{aligned}
 \text{So, no. of pins saved} &= 18\text{-bits} - 16 \text{ bits} \\
 &= \underline{\underline{2}}
 \end{aligned}$$

Q. Consider 64-bit hypothetical CPU which supports 64KW memory. System is enhanced with a byte-addressable memory. How many address pins are required in the enhanced system.

~~data size~~ mem size = 64 K W

$$= 64K \times 64$$

$$= \frac{64 \times 8 \times k}{2^{6+3+10}} \times \underline{8}$$

$$= 2^{19} \times 8$$

(Address pins = 19)

(cell size = 8)

Q. Consider 16-bit hypothetical CPU which supports 32KB memory space. System is enhanced with 64 bit word length. What are the changes in ~~enhanced~~ enhanced CPU.

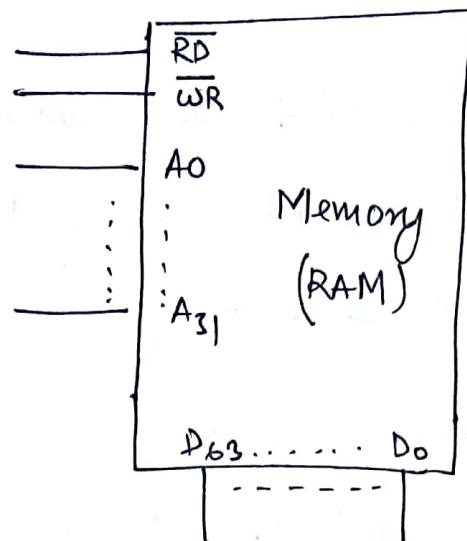
Mem size = 32 KB

$$= 32K \times 8$$

$$= 4K \times 64$$

$$= 2^{12} \times 64$$

Q. Consider the following Memory chip configuration



What is the memory capacity in bytes?

Address pins = 32

$$\text{so, no. of address} = 2^{32}$$

Data pins = size of each cell

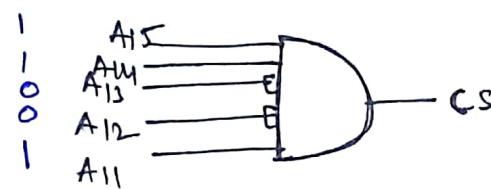
$$= 64\text{-bits}$$

$$\begin{aligned}\text{so, memory size} &= 2^{32} \times 64 \\ &= 2^2 \times 2^{30} \times 2^6 \cancel{8} \times 8 \\ &= 2^2 \times 9 \times 2^3 \times 8 \\ &= 2^5 9 \times 8 \\ &= \boxed{3298}\end{aligned}$$

Q. Consider the following chip-select logic to enable the DRAM chip.

A_0 to A_{10} pins are used to address the memory chips and remaining pins are used for chip select and 16-bit address format.

What is the range of addresses in hexadecimal of the memory system, that can be enabled by the chip select signal.

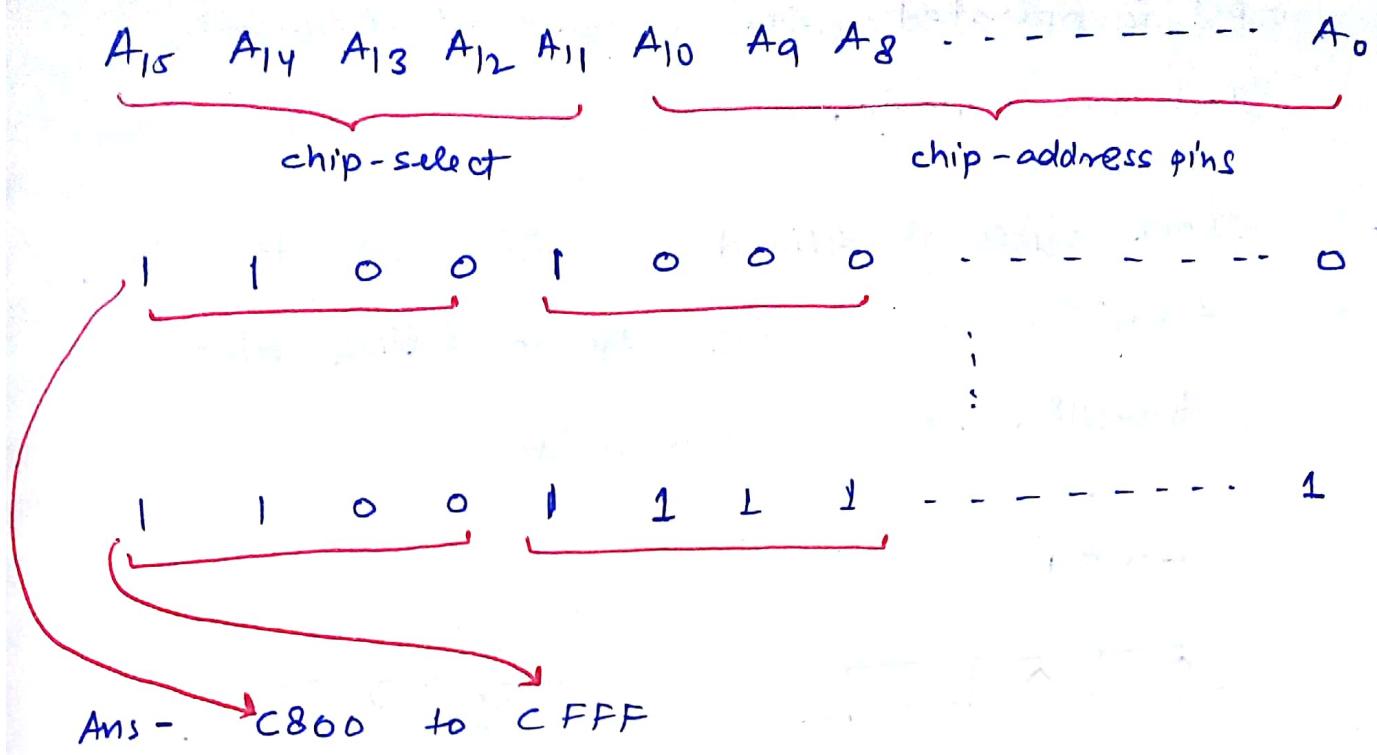


No. of chips = 2^5

$$\text{No. of address in each chips} = 2^{11} = 2K$$

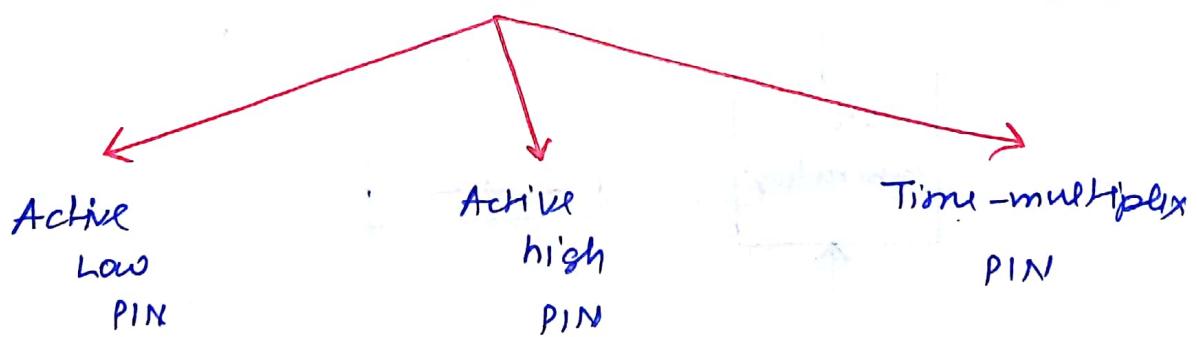
so, for CS to be active we need 11001 as input,

then,



CPU Pin structure -

- CPU contains hardware pins used to perform the operations on externally connected components (MEM & I/O).
 - Hardware pins are categorized into three groups.

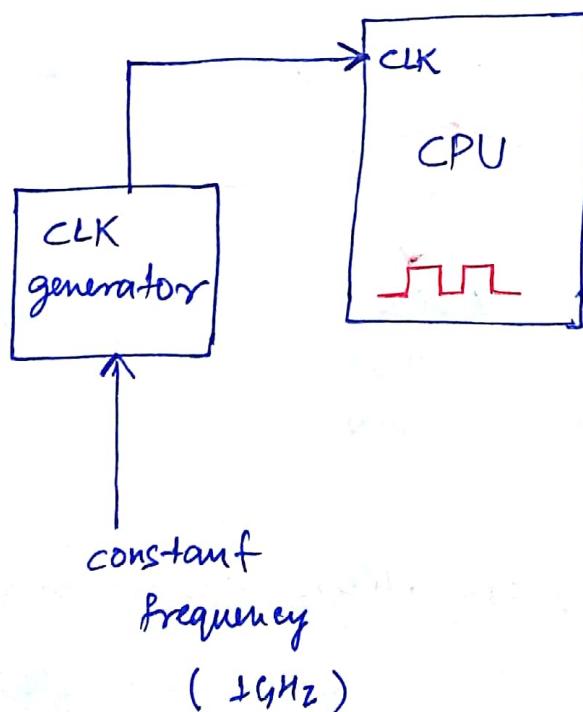


- In the CPU design, operations are controlled by the CLK signals.

- CPU is operated with a constant frequency clock oscillator.
- Clock cycle is defined as raising edge to raising transition or falling edge to falling edge transition of a clock pulse.



- Cycle time is defined as time required to transfer the clock pulse either from raising -to-raising edge or falling -to-falling edge, depends on CLK frequency.



$$\text{cycle time} \propto \frac{1}{\text{clk frequency}}$$

$$\text{cycle time} = \frac{1}{\text{clk frequency}}$$

$$= \frac{1}{1 \text{ GHz}}$$

$$= 10^{-9} = \underline{\underline{1 \text{ ns}}}$$

Active Low PIN :- This pin is enabled when the clock pulse is in low state. Denoted as "Pinname"

e.g., \overline{RD} , \overline{WR} , \overline{INTA}

Active High PIN :- This pin is enabled when the clock pulse is in high state. Denoted as "Pinname"

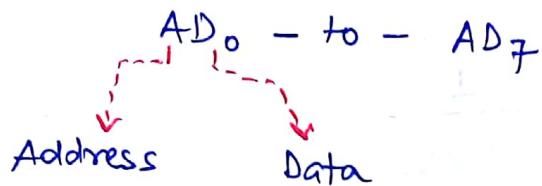
e.g., ALE, INTR, HOLD, HLDA

address
Latch
enabled

Interrupt
request

Time-multiplexed PIN :- This pin is used to carry the address and data but not at the same time.

Eg:- In 8085 MP



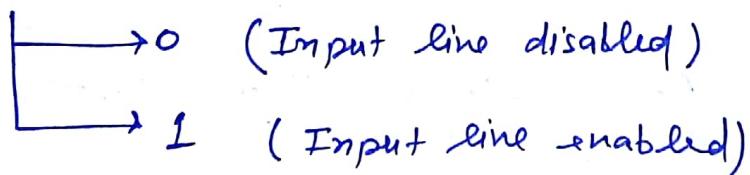
When ALE is 1 then address is on bus is enabled

and data is disable i.e on High clock signal

Address is taken and at low clock signal data is taken.

- Address Latch is used to Lock the address. It contains 2 control pins (STB and \overline{OE}) to control the input and output lines of a Latch respectively.

STB (strobe pins)



\overline{OE} (output enable pin)

