Unit 4: Memory

Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2 D memory organization, ROM memories, Cache memories: concept and design issues and performance, address mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks.
Virtual memory: concept implementation.

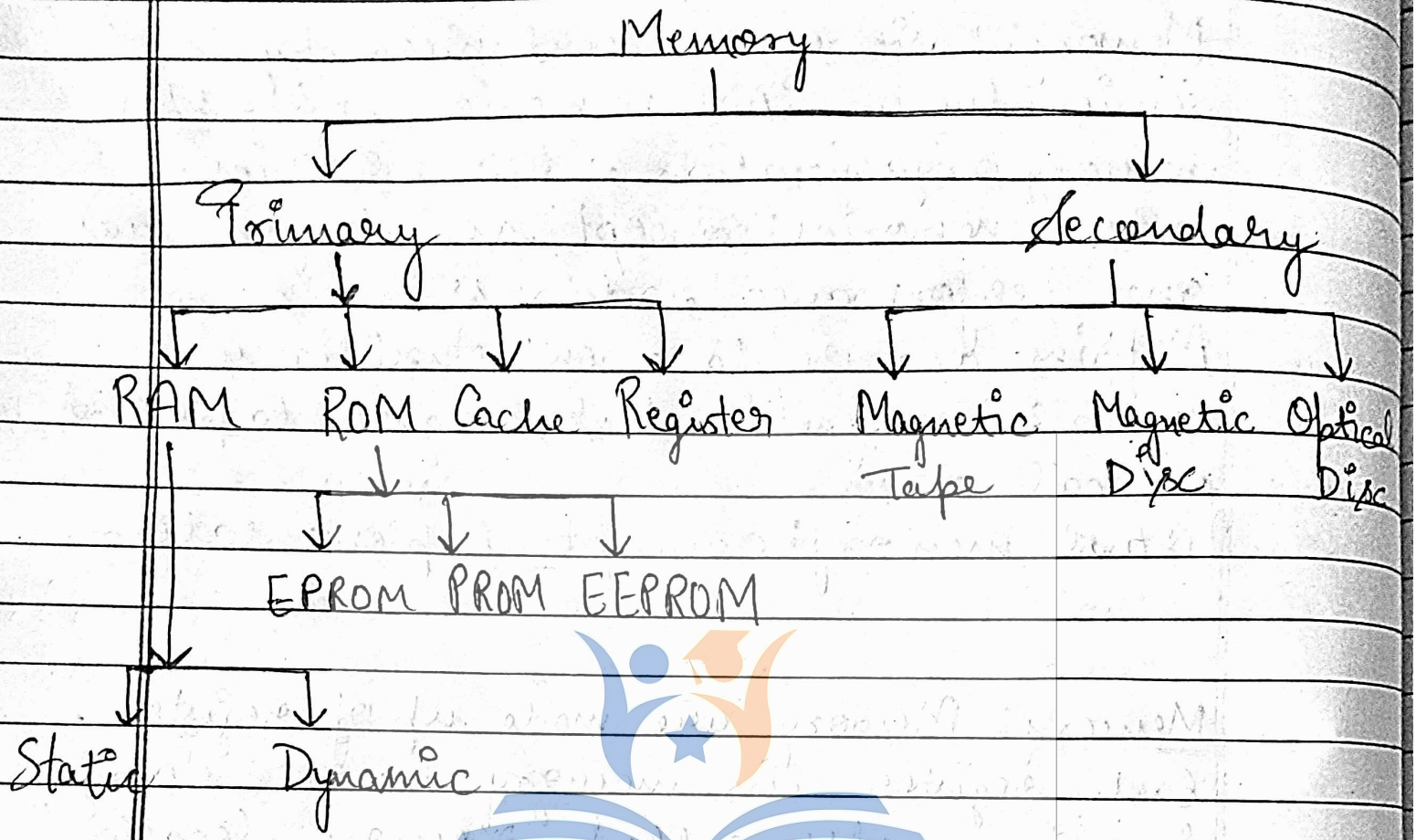
Memory: Memory are made up of register. Each register in memory is one storage location also called "Memory location"

Each memory location is identified by an address the no. of storage location can vary from a few in some memories to hundred of thousand in other.

The total no. of bits that a memory can store is its capacity.

Page No.
Date:

Types of Memory



Primary: It includes ROM and RAM and it is located close to CPU on the computer motherboard enabling the CPU to read data from Primary memory very quickly indeed.

Secondary: It includes physically located with a separate storage device such as a hard disk drive or solid state drive (SSD) which is connected to the computer system either directly or over the ~~network~~ hardware.

Page No.
Date:

RAM or Random Access Memory
or
Volatile memory

⇒ Memory in which any location can be reached in a short fixed amount of time after specifying its address.

Types

(1) SRAM

* It is less memory cell per unit area.
* Access time is less so faster.
* Consists of flip flop register.

↳ Cost is more
↳ Need less power

DRAM

* It is more memory cell per unit area.
* Access time is more than SRAM, so slower.
* Consists of mosfet of capacitor.
* Cost is less.
* More power is required.

Page No.
Date:

ROM (non-volatile) or (Read Only Memory)

Permanent Memory

PROM (Programmable ROM)

* Instruction can be stored only once
* Error in written instruction cannot be changed or erased.
* Once written usable

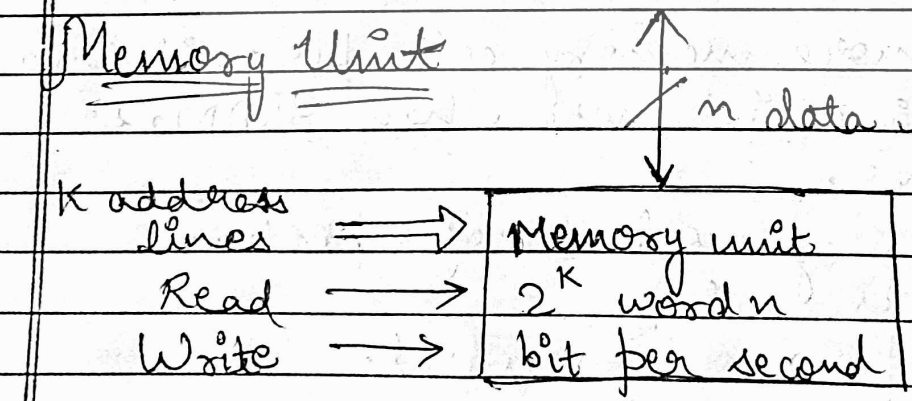
EPROM (Erasable Programmable ROM)

* It can be store multiple times
* written instruction can be changed or erased with help of UV Ray
* Rewriting is possible.

EEPROM (Electrically erasable programmable ROM)

Instructions already written in EEPROM can be erased with the help of electricity and we can rewrite the instruction.

Memory Unit



The Block diagram of memory unit then the lines provide the information to be stored in memory and the K address lines specify the particular word.

When there are k address lines we can access 2^k memory words.

If $k = 10$

$2^k = 2^{10} = 1024$

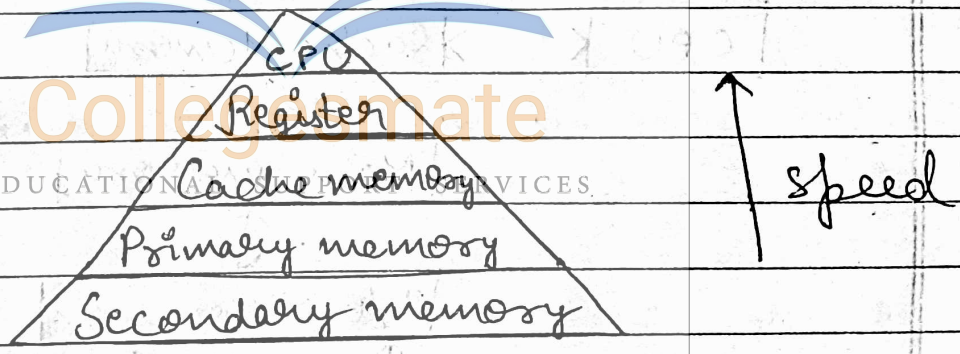
Total memory = 1024

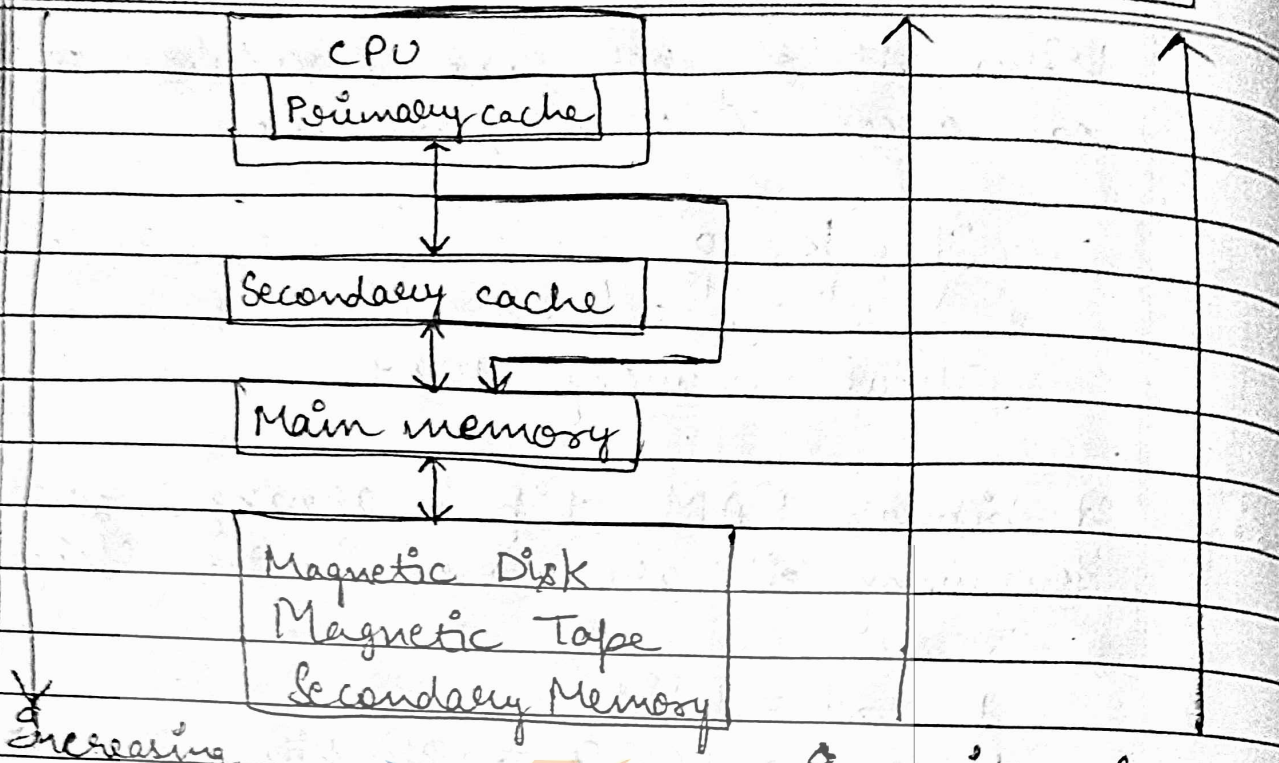
Ques: A bipolar RAM chip is arranged as 16 words. How many bits are stored in the chip.

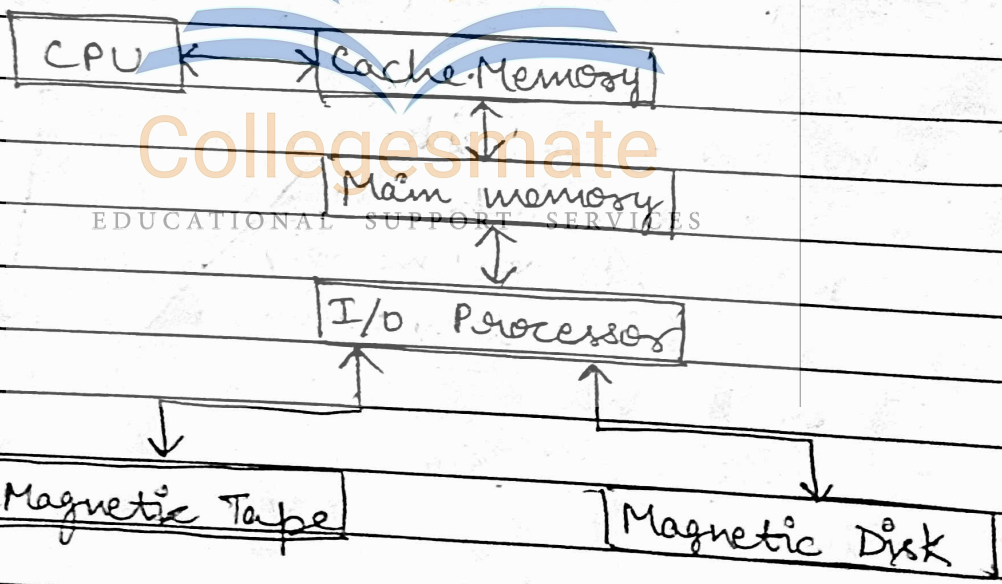
⇒ 1 word = 8 bits

16 word = $16 \times 8 = 128$ bits

Memory hierarchy







Computer memory should be fast, large and inexpensive.

Increased size, speed are achieved as increase cost.

Page No.
Date:

Very fast memory system can be achieved if SRAM (static RAM chips are used).

Processor fetch decode and data from the main memory to execute the program. The DRAM which form the main memory are slower device.

The program which is to be executed is loaded in the main memory.

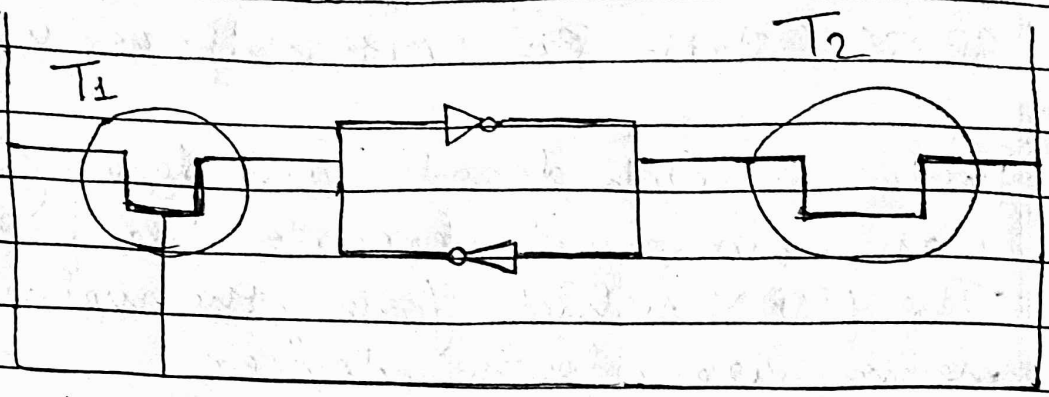
The cache memory just discussed is called secondary cache, resend processor have the built in cache memory called primary cache.

Semiconductor RAM

Memory that consist of circuit capable retaining these state as long as power is applied are known as static memory. These are of Random Access Memory so called static RAM.

Page No.
Date:

Static RAM cell



This figure shows the implementation of SRAM cell. It consists of 2 cross clip and 2 transistor T_1 and T_2 which act as switch.

The latch is connected two bits line transister T_1 and T_2.

When the word line is zero level, the transistor are off and the latch is stable state in Read operation.

Read op^{n}: If $b=1$, $b'=0$ then T_1 transistor is on and T_2 is also on and word line off.

Write op^{n}: Only word line is activated and transistor T_2 is activated and $b=0$, $b'=1$ and T_1 is off.

Page No.
Date:

Dynamic RAM

Dynamic RAM store the data as a charge on the capacitor it contain 1000 of such memory cells. When column and row line are high the mosfet corder and charges the capacitor.

When the column and row lines go low the mosfet open and the capacitor withtens its charge in this way it store one bit.

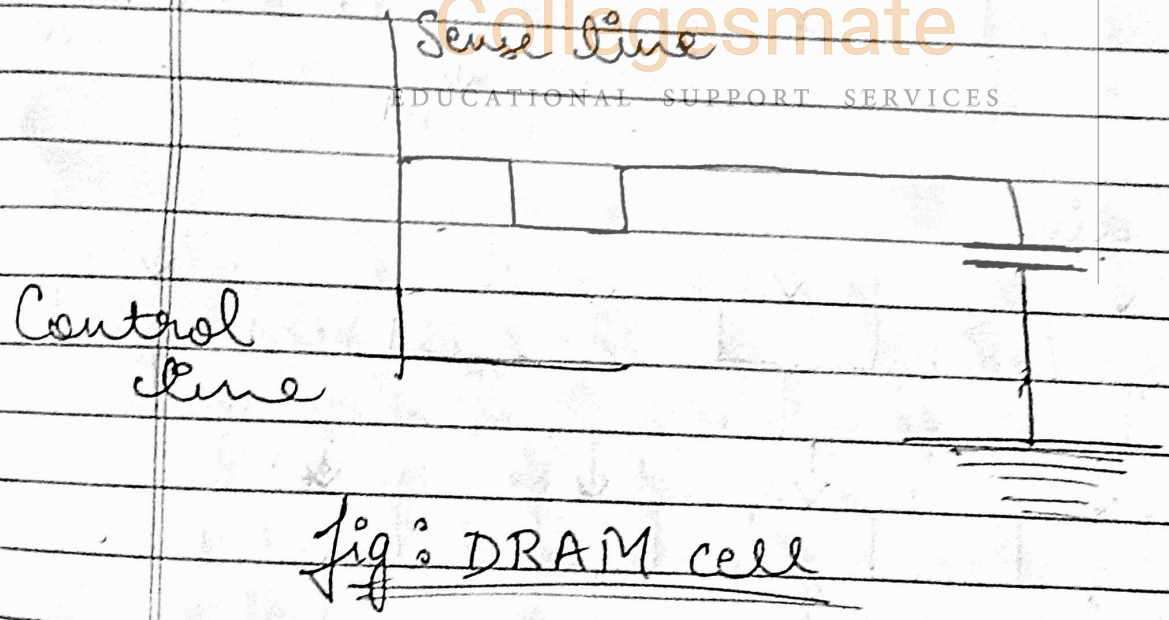
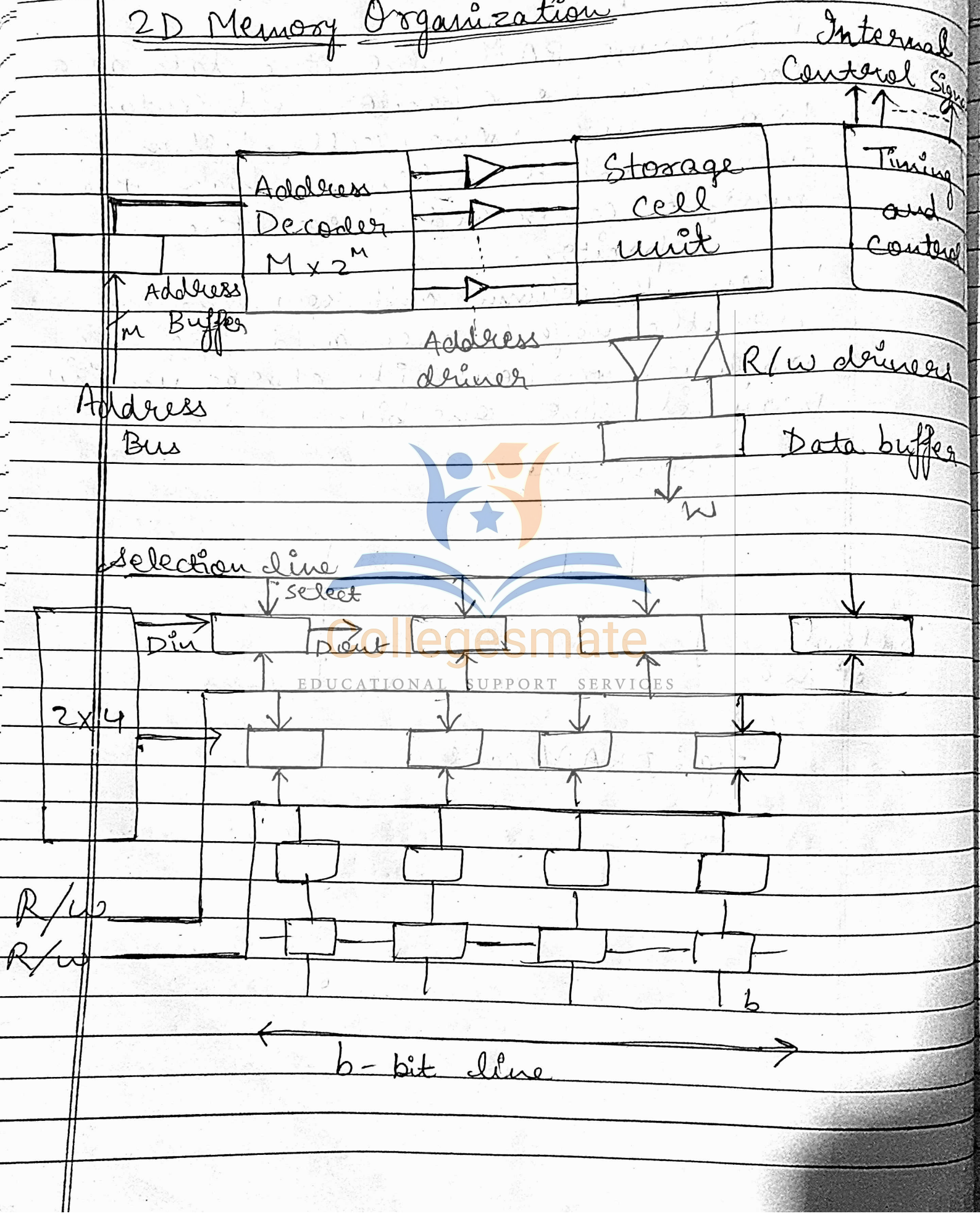


fig: DRAM cell

2D Memory Organization



Page No.
Date:

This is a simplest organization as shown in fig the cells are organized in the form of a 2D array with row and column.

Each row refers to word line for b-bit per word memory b no. of cells are interconnected to a word line. Each column in the array refers to a bit line.

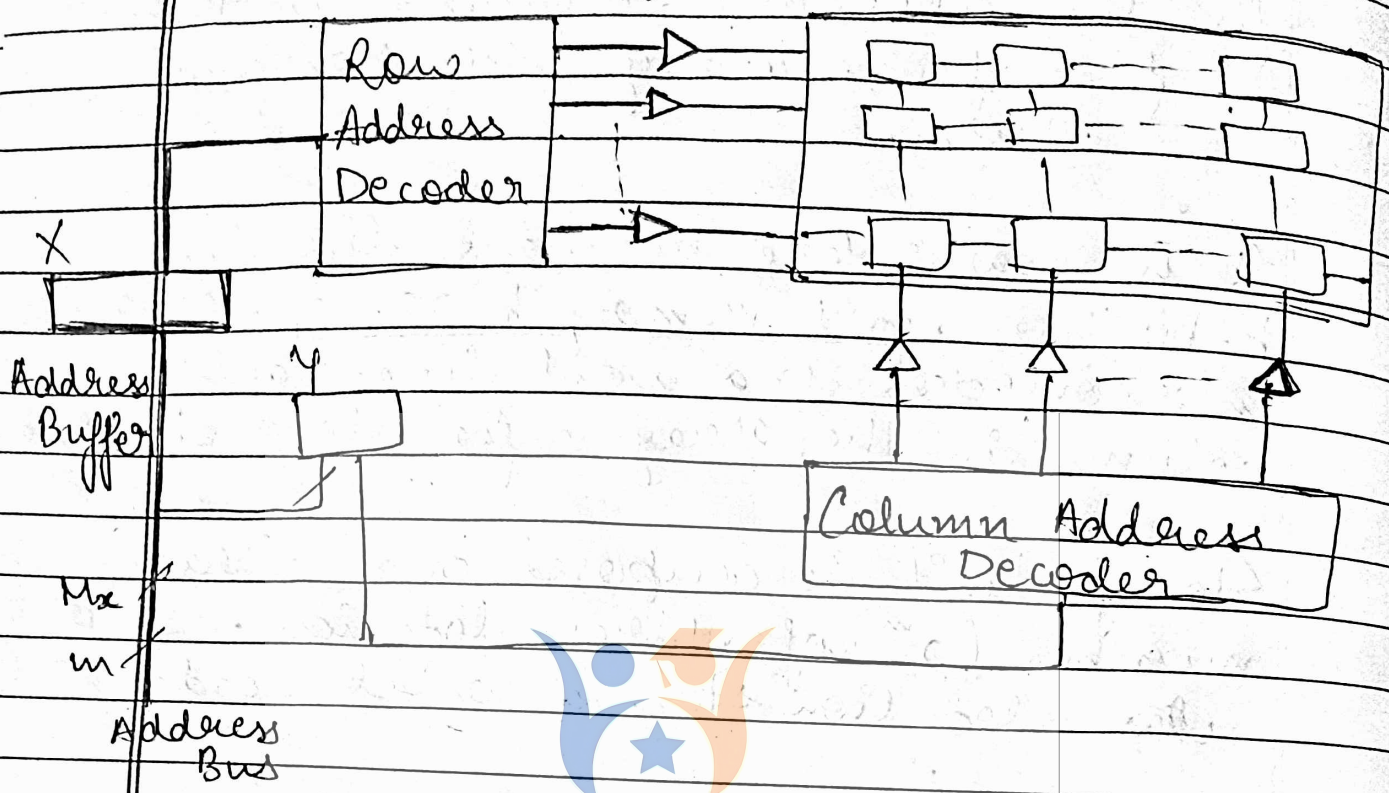
Storage unit is composed of a large number (2^m) of address location. Each location store a w bit word.

An m bit address is generated by the processor. It is store into an address buffer. Output of address buffer is include to the address decoder.

The address decoder select on of the address decoder in the memory the of address decoder passes through try state buffer.

Page No.
Date:

2 ½ D Memory



In 2.5D organization the scenario is the same but we have 2 different decoders one is column decoder and another is row decoder. Column decoder is used to select the column and row decoder is used to select the row. The address from the MAR will go in decoder's input.

Decoder will select the respective cell through the bit outline, the data from the location will be written at that memory location.

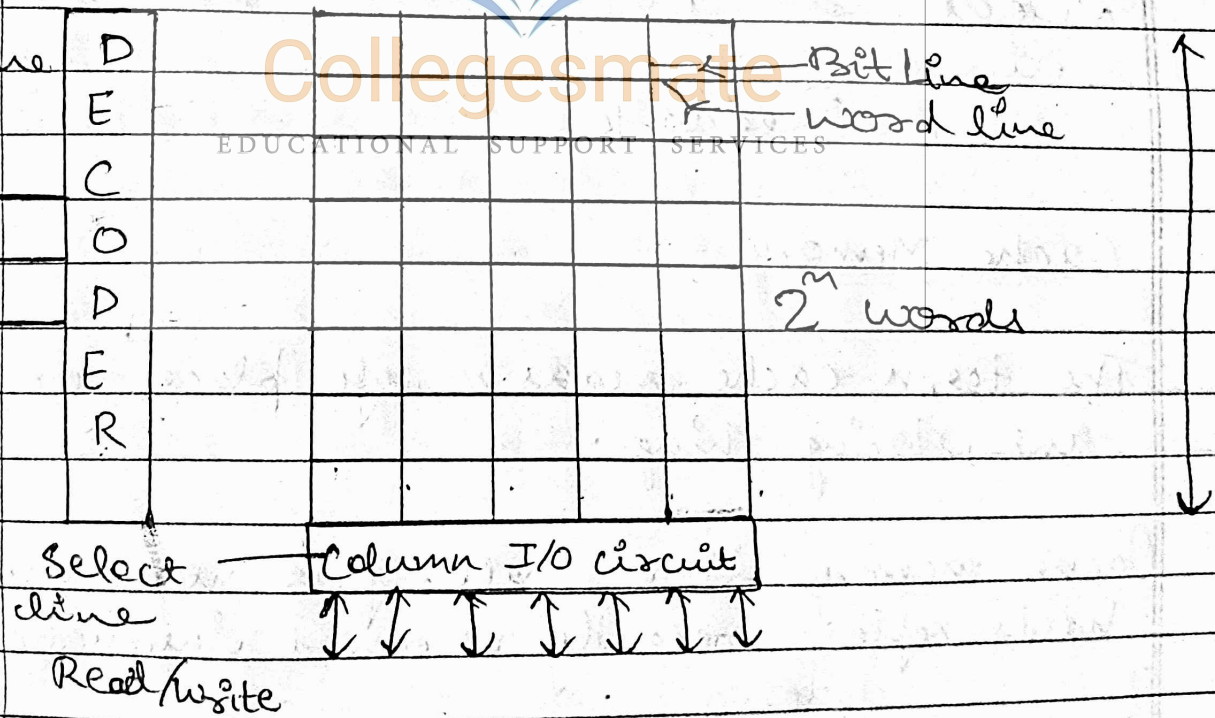
Page No.
Date:

Read and Write op^n

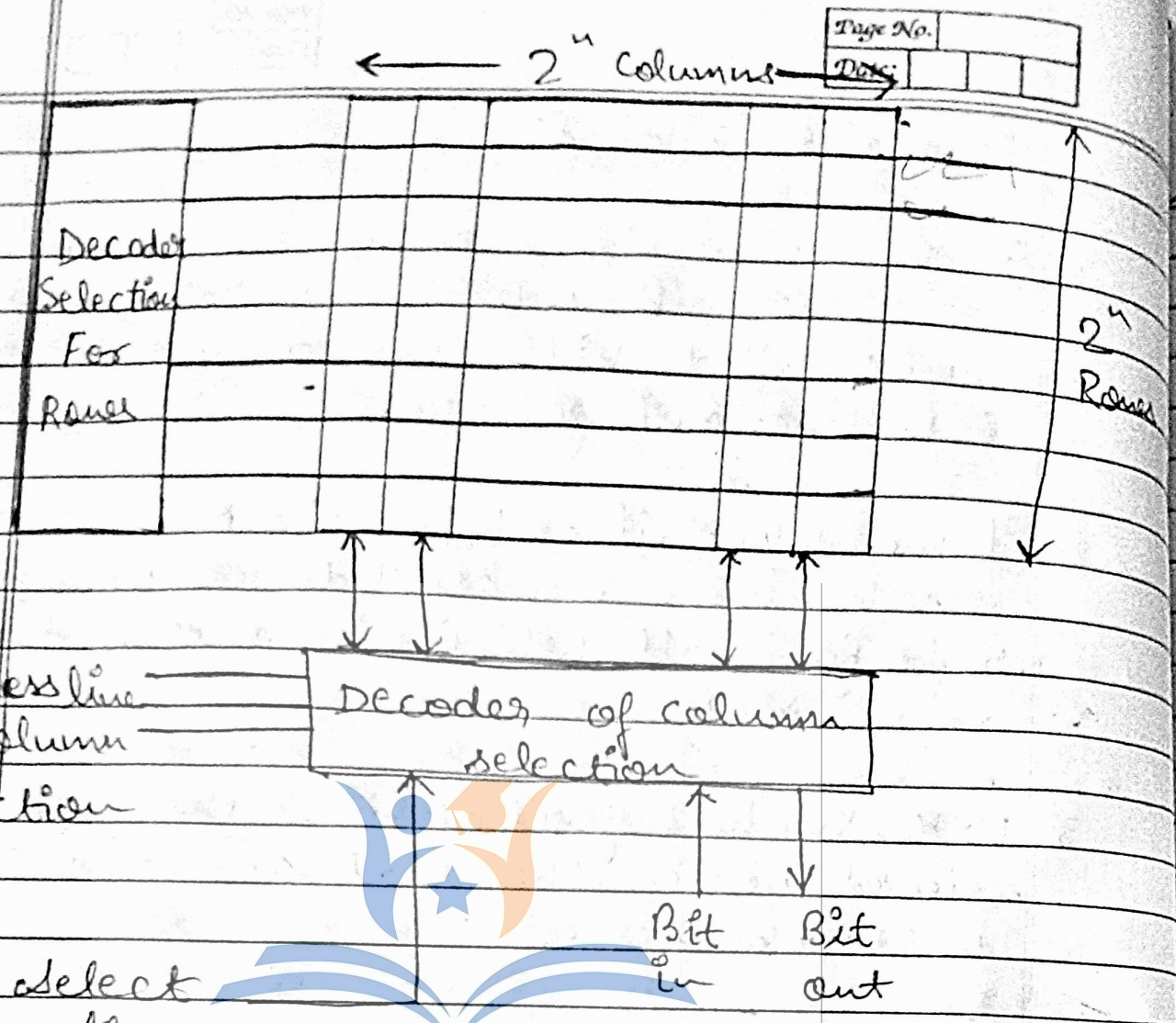
1. If the select line is in Read mode then the word/bit which is represented by the MAR that will be coming out to the Data lines and get read.

2. If the select line is in write Mode then the data from memory data Reg will go to the respective cell which is address by the MAR.

3. With the help of select line the data will get selected where the read and write op^n will take place.



2D Memory Organization



Cache Memory

The term cache means a safe place for hiding and storing things.

Cache memory is a small, fast memory which holds copies recently accessed instruction and data.

When the processor makes a request for memory reference the request is first sought in the cache.

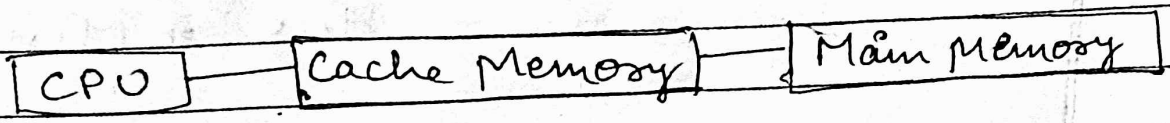
If we get that memory reference which is requested we call it 'cache hit' otherwise 'cache miss'.

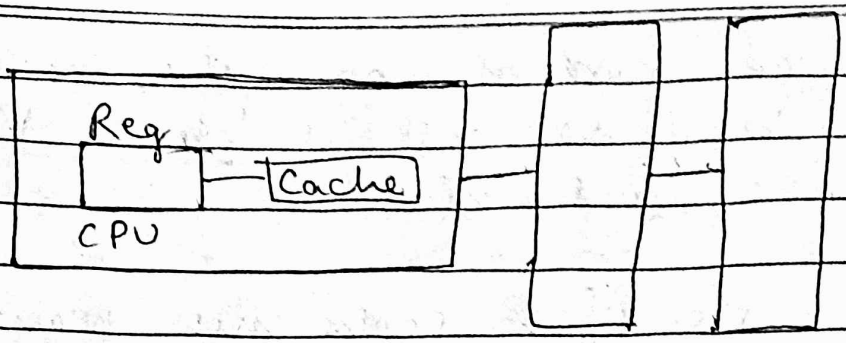
In the case of a cache miss requested element is brought from a subsequent memory level from memory hierarchy and placed in cache.

A block of elements are transferred from main memory to cache memory by expecting that the next requested element will be residing in the neighboring locality of current requested element (spatial locality) and this has to happen under one main memory Access time.

* Cache memory is organized not in bytes, but as blocks of cache lines with each lines containing some no. of bytes (16-64).

* Cache lines do not have fixed addresses which enables the cache system to populate each cache line with a unique (non-contiguous) address.





Cache Memory Performance

It is measured in terms of hit ratio.

Cache hit: If the required word is found in cache is called cache hit.

Cache Miss: If the required word is not found in cache is called cache miss.

$$\text{Hit ratio (H)} = \frac{\text{Hits}}{\text{Hits + Miss}}$$

OR

$$= \frac{\text{no. of hits}}{\text{Total no. of CPU references}}$$

$$\text{Miss Ratio} = \frac{\text{No. of Miss}}{\text{Hits + Miss}}$$

or

$$= \frac{\text{No. of Miss}}{\text{Total no. of CPU references}}$$

Page No.
Date:

Cache Access time

If cache hit when access the cache that time is called cache access time it is also called (cache hit time).
Time required to access word from the cache.

Miss Penalty

Cache Miss Time Penalty: It is the time required to fetch the required Block from main memory.
(cache access + MM Access time)

Average Access time of CPU

= hit ratio x Cache Access time +
(1 - hit ratio) x Miss Penalty

= h x Tc + (1-h) x Tm //

Page No.
Date:

Ques: The access time Cache Memory is 100 ns and the of main memory 1000 ns. It is estimated that 80% of memory request are for read and the remaining 20% for write. The hit ratio for read access only is 0.9. A write through procedure is used.

(a) What is the avg access time of the system considering only memory read cycle
(b) What is the avg access time of the system for both read and write cycle?
(c) What is the hit ratio taking into consideration time the write cycles.

a) Total avg access time = $0.9 \times 100 + (1-0.9) \times (100 + 1000)$

$= 0.9 \times 100 + (1-0.9) \times 1100$

$= 200$ ns //

Total write $= 1 \times \text{Max}(100, 1000)$ ← MM

$= 1000$ ns

b) Avg access time for Booth read and write
= 80% are read request + 20% write Request
$= 0.8 \times 200 + 0.2 \times 1000$
$= 160 + 200$
$= 360$ ns

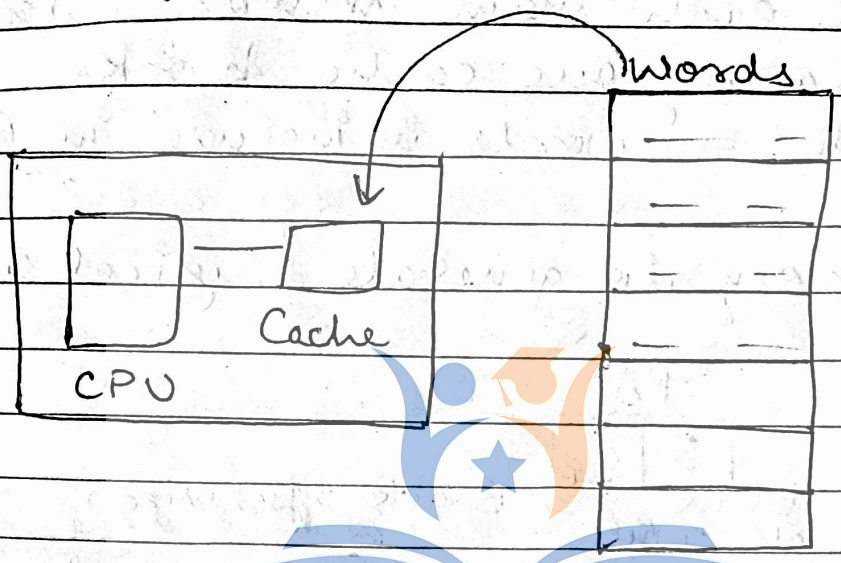
Page No.
Date:

Cache Mapping and its types

Direct Mapping

Fully Associative

Set Associative



which line contains in the cache memory. So this is called a cache mapping and How to fetch the Block or reference of cache data by CPU.

Direct Mapping

	Cache		
L_0	B_0	B_4	B_{28}
L_1	B_1	B_5	B_{29}
L_2	B_2	B_6	B_{30}
L_3	B_3	B_7	B_{31}

16 Words

Line size = Block size

$\frac{16}{4} = 4$ //

W_0 W_1 W_2 W_3	B_0
W_4 W_5 W_6 W_7	B_1
W_8 W_9 W_{10} W_{11}	B_2
W_{12} W_{13} W_{14} W_{15}	B_3
W_{16} W_{17} W_{18} W_{19}	B_4
⋮	⋮
W_{124} W_{125} W_{126} W_{127}	B_{31}

128 Words

$\frac{128}{4} = 32$ //

(K mod n formula used)

↓ 0 mod 4 = 0

where K is Block number

n is no. of lines

A direct-mapped cache is the simplest approach each main memory address maps to exactly one cache block.

For example: words is include in Block.

Main memory to generate physical address.

PA

| 5 | 2 |

Block No. 7 bits → Block offset (size)

0 - 127 = 7

0 - 3 = 2 bits (How many bits to be used to represent a word?)

Address 0001010 = 10th Word

5 bit (Block no.) → 2

10 (Block offset) ⇒ 2 (word)

P(A) = 7

| 3 | 2 | 2 |

Tag | Line no. | (Block offset)

No. of lines = 4 = (2)

Ex: Cache memory available to represent a

Tag

| 010 | 00 |

00 = 32
01 = 33
10 = 34
11 = 35

Page No.
Date:

~~Tag = 0010~~ Tag = 001 01

010	L_0 ↓ B_8
001	L_1 ↓ B_5

00	20
01	21
10	22
11	23

Ques: Consider a direct mapped cache of size 32 kB with Block size 32 bytes. The CPU generates 32 bit address. The no. of bits required for cache indexing and tag bits respectively.

?

PA = 32 bit

17	10	5

(Tag) Block no. (Line no.) Block offset (Block size)

0	
1023	

Line size = Block size

$K = 2^{10}$

32 kB

~~32 B~~

Page No.
Date:

Fully associative

L_0	B_1
L_1	B_0
L_2	B_2
L_3	B_3

16 words

W_0, W_1, W_2, W_3	B_0
W_4, W_5, W_6, W_7	B_1
W_8, W_9, W_{10}, W_{11}	B_2
$W_{12}, W_{13}, W_{14}, W_{15}$	B_3
⋮	⋮
$W_{124}, W_{125}, W_{126}, W_{127}$	B_{31}

128 words

Line size = Block size

← 7 bits →

5	2

Block no. (Block offset)

Cache

← 7 →

Tag (5)	2

Set Associative: It is combination of Direct and fully Associative.

L_0	B_0	S_0
L_1	B_0	
L_2	B_1	S_1
L_3	B_1	

16 words Cache

$\frac{16}{4} = 4$

W_0 W_1 W_2 W_3	B_0
W_4 W_5 W_6 W_7	B_1
W_8 W_9 W_{10} W_{11}	B_2
⋮	⋮
W_{124} W_{125} W_{126} W_{127}	B_{31}

128 words

Page No.
Date:

K way set associative are given means no of set to calculate.

2 way
4 way
8 way

$= \frac{\text{no. of lines}}{K} = \frac{4}{2} = 2$ //

Direct Mapping : K mod n where n number of set

$= 0 \bmod 2$
$= (\text{Associative})$

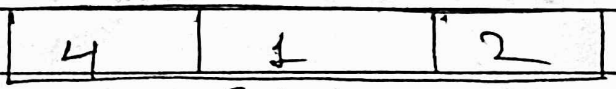
$B_1 = 1 \bmod 2$
$= (1)$

set 1

PA

(Block no.) Block offset

Cache memory



Tag Set No Block offset

Page No.
Date:

Eg: A 4 way set associative cache memory with a capacity of 16 kB is built using a Block size of 8 words. The word length is 32 bit. The size of PA space is 4 GB. The no. of bits for the tag field is

4 GB $= 2^{30}$

$= 2^2 \, 2^{30}$

$=$ 32 bit

PA = 4 GB

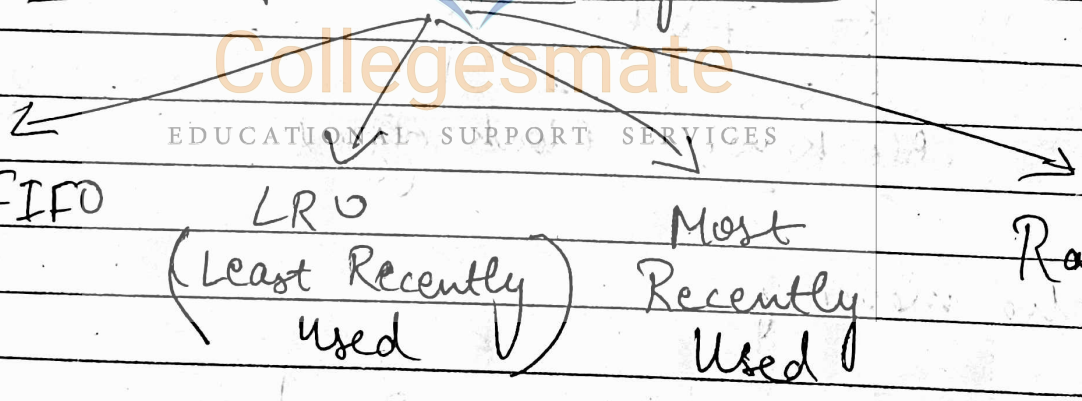
← 32 bit →

20	7	5
Tag	Set No.	Block offset

8 words

2^3

Cache Replacement Algorithm



Ques: Consider a 4 way set associative cache with total 16 cache blocks main memory Block. Request are

0, 255, 1, 4, 3, 8, 133, 159, 216, 129, 63, 8, 48, 32, 73, 92, 155

S_0	0	set 1
S_1	1	set 2
S_2	2	set 3
S_3	3	set 4
	4	
	255	

16 lines
4 set
no. of set = 4

16/4

k mod n
0 mod 4
= 0

Ques: Consider a fully associative cache with 8 cache blocks (0–7) and the following sequence of memory Block requests.

4, 3, 25, 8, 19, 6, 25, 8, 16, 35, 45, 22, 8, 3, 16, 25, 7

If LRU replacement Policy is used, which cache Block will have memory block 7?

⇒

0	~~4~~ 45
1	3 22
2	25
3	8
4	19
5	6
6	16
7	35

cache.

Page No.
Date:

Virtual Memory: A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer RAM.