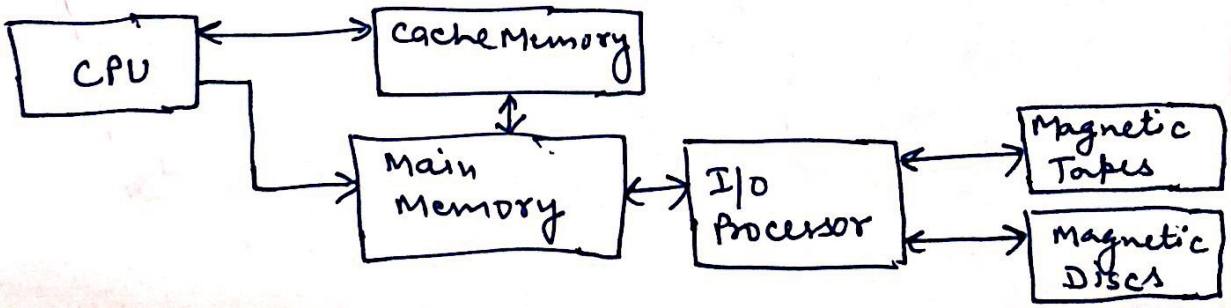
LECTURE

Content : Basic concept and hierarchy of Memory

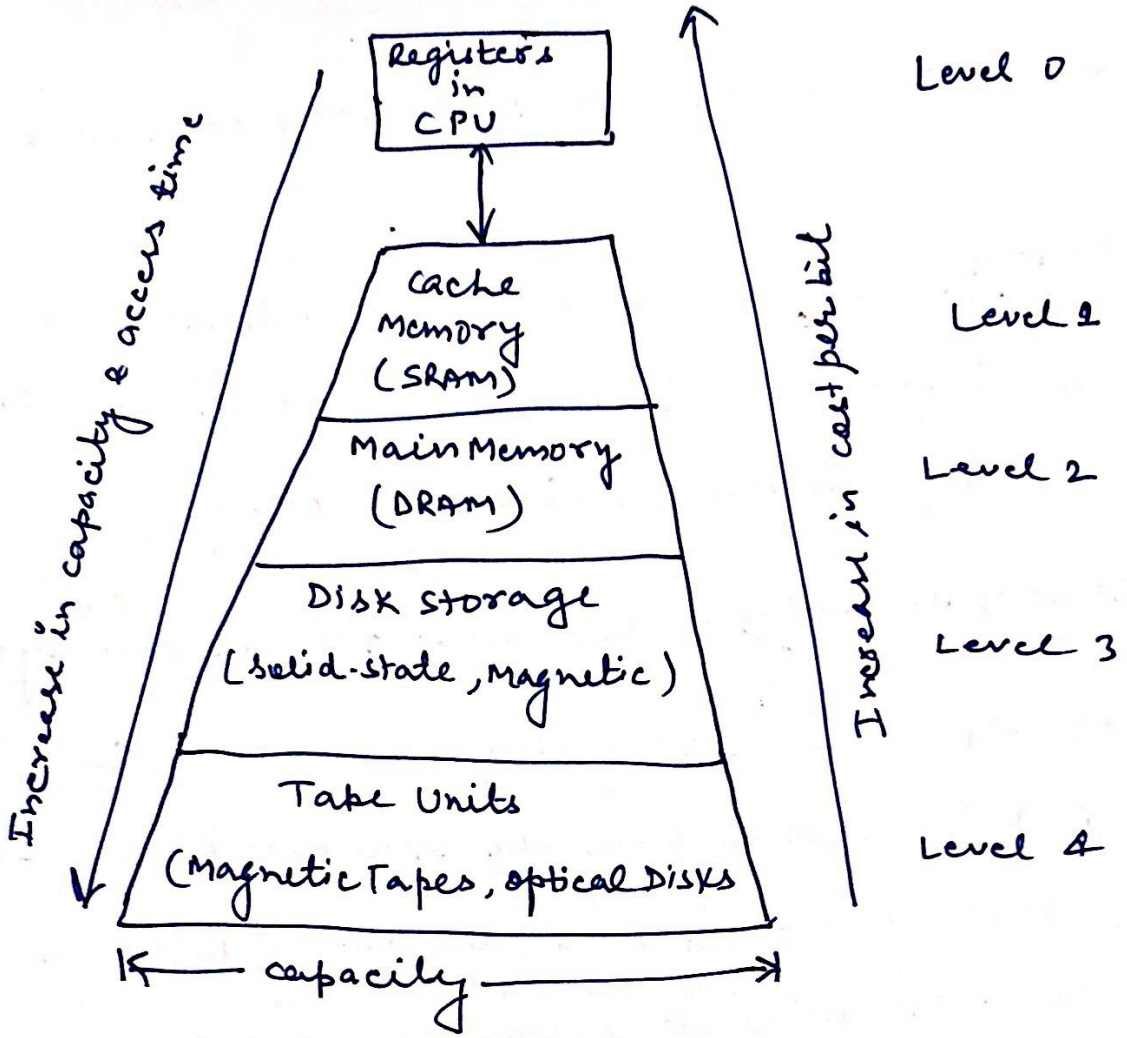
Memory: It is a storage unit which stores the data and information in a computer. It stores data and information in binary (0 or 1).

Memory Hierarchy: The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high capacity auxiliary memory to a relatively faster main memory, to as even smaller and faster cache memory.

The advantage of memory hierarchy system is to increase the storage capacity, increase the processing speed and reduce the storage cost of the computer system.



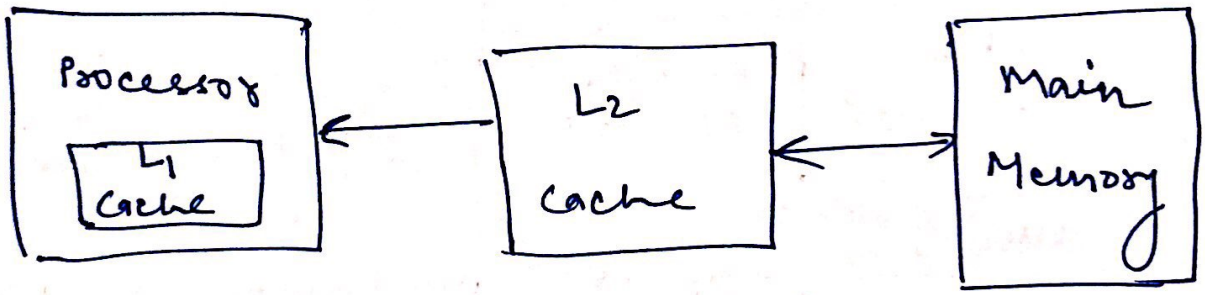
Memory hierarchy in a computer system.



A four level Memory hierarchy system

Cache Memory: Cache memory is defined as a high speed memory. that is placed between the CPU and main memory to compensate the speed difference between the two cache is made up of SRAM.

It is also possible to place an even smaller cache between the cache and the processor thereby creating two level of cache shown in figure below.



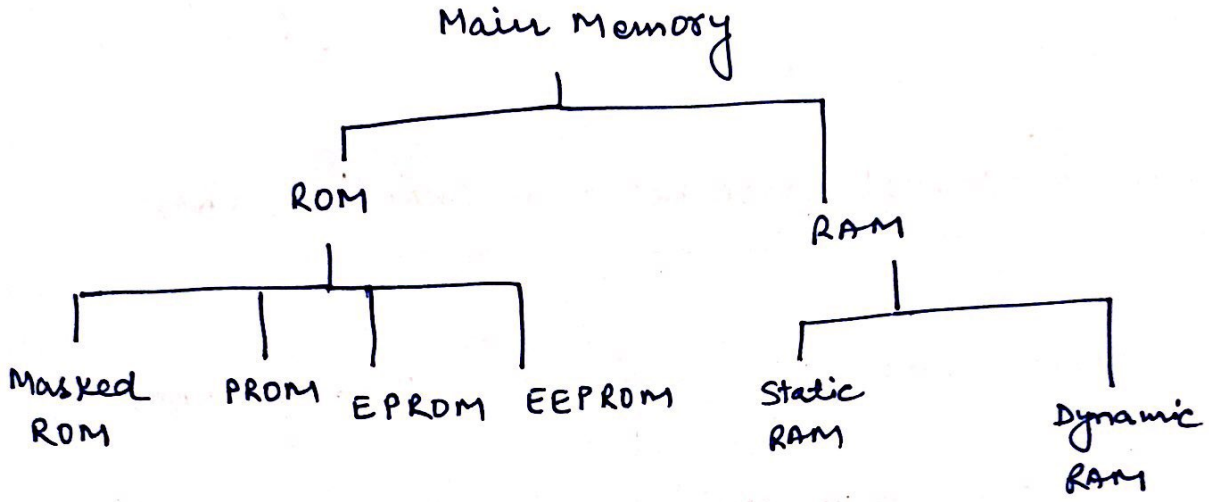
L1 & L2 Cache Memory

Memory level characteristics	Level 0 CPU register	Level 1 cache	Level 2 Main Memory	Level 3 Disk Storage	Level 4 Tape storage
Device Technology	ECL	256 K-bit SRAM	4M-bit DRAM	1-G byte Magnetic disk unit	5-G byte magnetic tape unit
Access time, t_i	10 ns	25-40 ns	60-100 ns	12-20 ms	2-20 min
capacity, s_i in bytes	512 bytes	128 Kbytes	512 Mbytes	60-220 G bytes	512 G bytes - 2 T bytes
cost, c_i (in cents/KB)	18000	72	5-6	0.23	0.01
Bandwidth, b_i (in MB/s)	400 - 800	251 - 400	80 - 133	3 - 5	0.18 - 0.23
Unit of transfer	4-8 bytes per word	32 bytes per block	0.5 - 1 Kbytes per page	5-512 Kbytes per file	Back up storage
Allocation Management	Compiler assignment	Hardware control	operating system	operating system/ user	operating system/ user

LECTURE

Content: Semi. conductor RAM Memories

Main Memory classification: The main memory is classified as:



Random Access Memory (RAM):

It is the main memory of a computer. It is used to store data and application that are currently in use. Both read and write operations are performed in RAM. In RAM data is accessed Randomely (some time to access any word in the RAM at any location). RAM is volatile (retains data & information written in it till power supply is provided.

The two main classification of RAM are:

SRAM (Static RAM): It is made up of flip flops. so it holds data without external refresh as long as power is supplied to the circuit.

DRAM (Dynamic RAM): It is made up of capacitors. so external refreshing is done many hundreds of times per second.

Comparison/Difference between SRAM & DRAM

Static RAM	Dynamic RAM
1- It contains less memory cells per unit area	1. It contains more memory cell as compared to SRAM per unit area.
2. It requires less access time hence faster memory	2. Its access time is greater than SRAM.
3. It is made up of flipflops.	3. It is made up of capacitor + MOSFET.
4. Refreshing circuit is not required	4. Refreshing circuitry is required to maintain the charge on the capacitors after every few miliseconds
5. cost is more	5. cost is less.

Note: DRAM is the main component of the main memory because it has more memory cells per unit area as compared to SRAM.

Read Only Memory (ROM) :

ROM is a non-volatile memory which retains the data even when power is removed from this memory. It is also random access. It is used for storing programs that are permanently resident in the computer. ROM is used to store an initial program called a bootstrap loader. The function of this program is to start the computer operating system, when the power is turned on.

When power is turned on, the hardware of the computer set the program counter to the first address of the bootstrap loader. The bootstrap program loads a portion of the operating system from disk to main memory and control it then transferred to the operating system, which prepares the computer for general use –

The various kinds of ROM are –

Masked ROM :

1- Mask Programming : It is done by the company during the fabrication process of the unit. The procedure for fabricating a ROM requires that the customer fills out the truth table he wishes the ~~store~~ ROM to satisfy.

~~2. PROMs: PROM contain all~~

(1) Masked ROM : The very first ROM's were hard-wired devices that contained a pre-programmed set of data or instructions. These kind of ROMs are known as masked ROM. which are inexpensive

(2) PROM (Programmable ROM) : PROM is read only memory that can be modified only once by a user. The user buys a blank ROM and enters the desired contents using a PROM program.

(3) EPROM (Erasable and Programmable ROM) : EPROM can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes

(4) EEPROM (Electrically Erasable & Programmable ROM)

EEPROM is programmed and erased electrically or can be erased and programmed about ten thousand times. Both erasing and programming take about 40 to 10 ms.

LECTURE

Content: 2D & $2\frac{1}{2}$D Memory organization.

RAM Organizations:

To perform read and write operation in RAM memory can be organized by the following three methods:

1. One dimensional RAM (1-D RAM)
2. TWO dimensional RAM (2-D RAM)
3. $2\frac{1}{2}$ dimensional RAM ($2\frac{1}{2}$ D RAM)

One dimensional (1-D) RAM:

This cell structure is of a one dimensional length. It has large number of addressable storage locations (2^m). each address of which stores n bit word. Therefore the capacity of RAM is specified by —

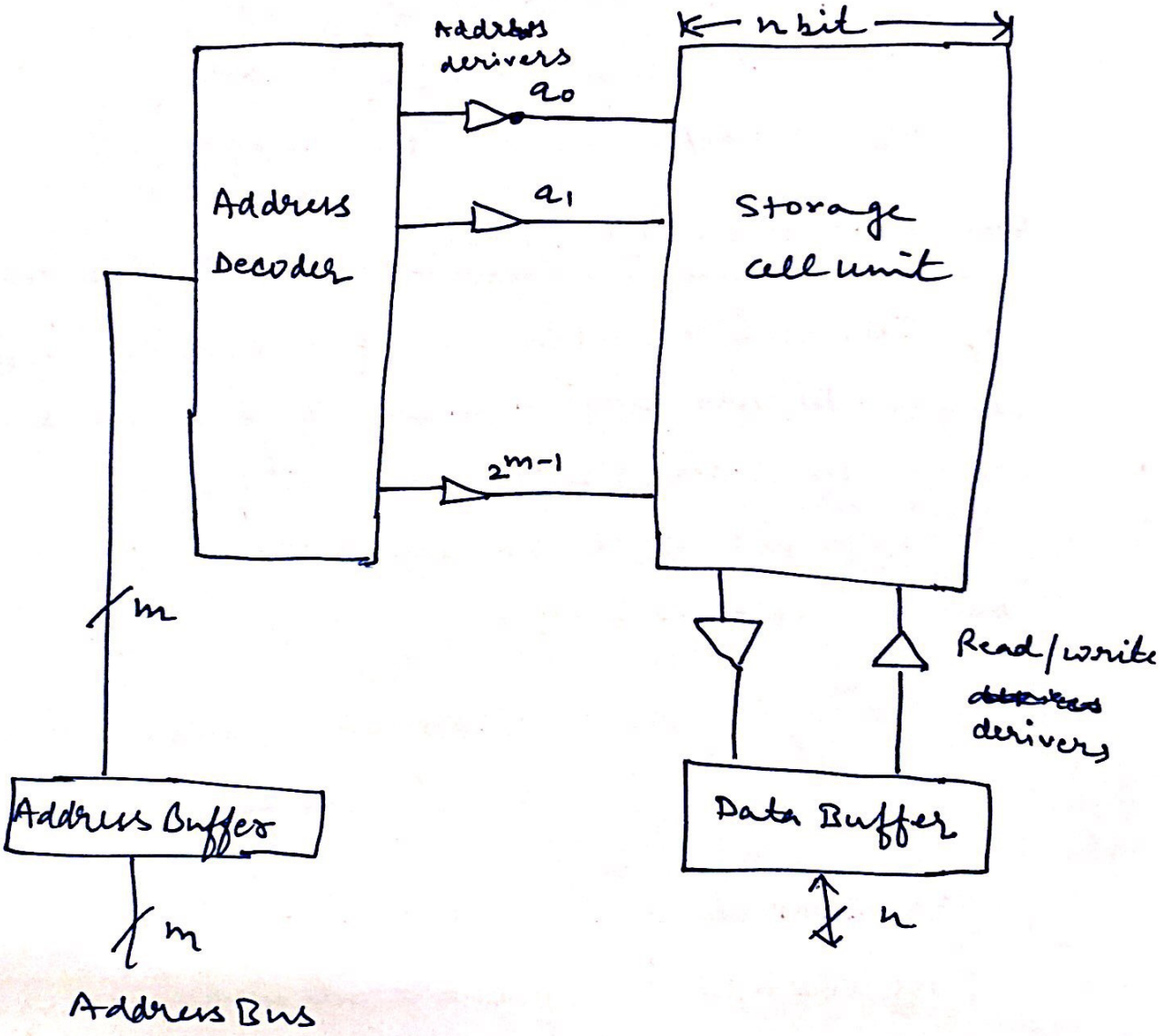
$2^m \times n$ where ($2^m \rightarrow$ no of address lines, $n \rightarrow$ no of bits per address)

The RAM operates as follows —

First the address of the word to be accessed in RAM is transferred to address buffer & of the RAM. This address is

the contents of the location (of the storage cell, RAM) is transferred to data buffer and from there to data bus. If the operation is write, the data from data bus is transferred to location specified in the storage cell, RAM.

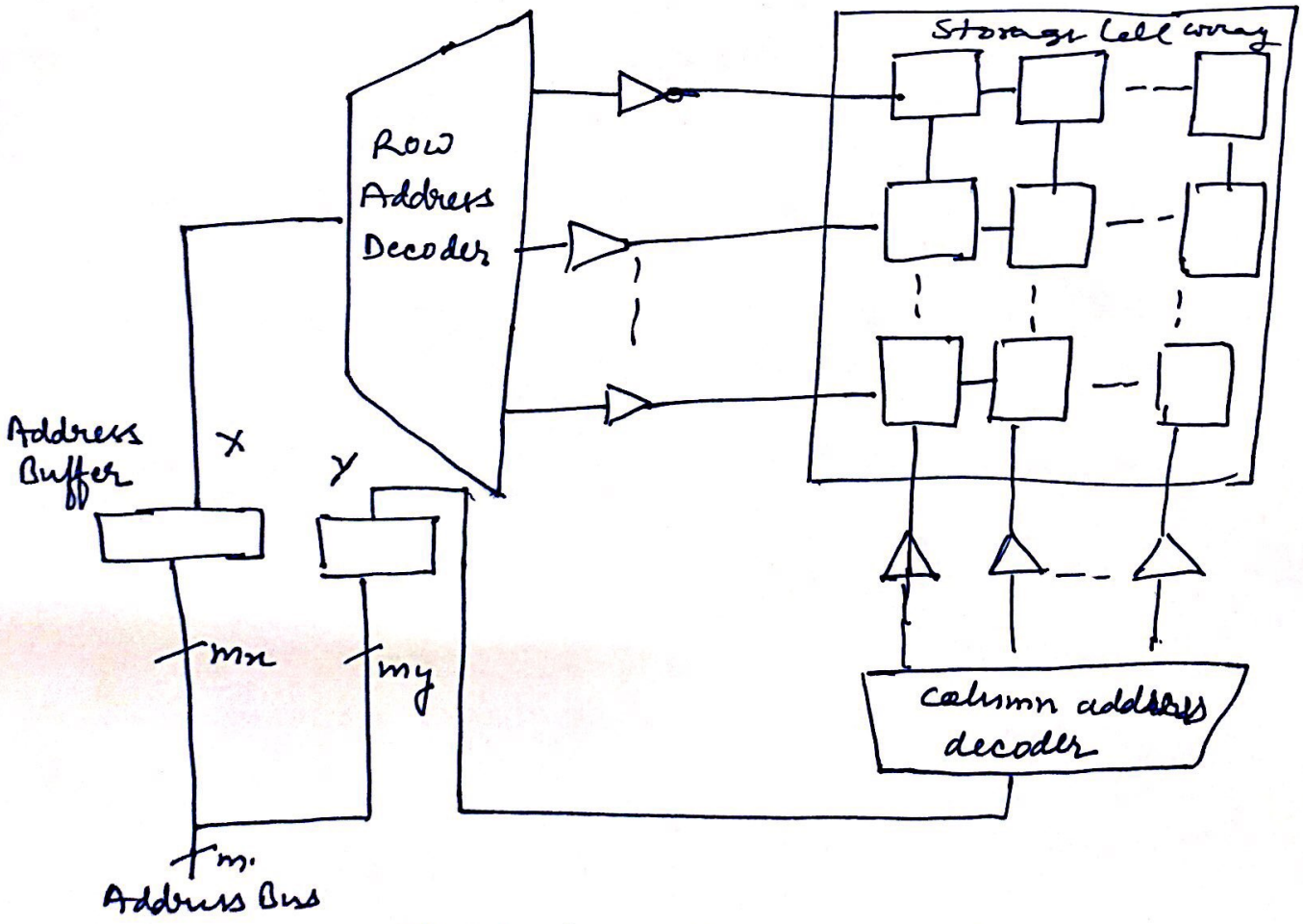
It is used for low memory capacity organization.



One Dimensional (1-D) RAM

Two Dimensional (2-D) RAM: In large memory capacity the chips are often arranged as row multiplexed column. Here the m-bit address word is divided into two parts x and y for address selection and column selection respectively. The cells are arranged in a rectangular array of N_x rows and N_y columns. (Where m_x, m_y are the number of bits for address and column line respectively as $N_x \leq 2^{m_x}$, $N_y \leq 2^{m_y}$). The total number of cell in 2D memory is $N_x \cdot N_y$.

- It is used for large memory organization.



Two dimension (2-D) RAM

$2\frac{1}{2}$ - Dimension ($2\frac{1}{2}$ - D) RAM :

$2\frac{1}{2}$- is similar to 2-D RAM but- in this the column lines and row lines are split into unequal size. (In 2-D the column lines and row lines are equal)

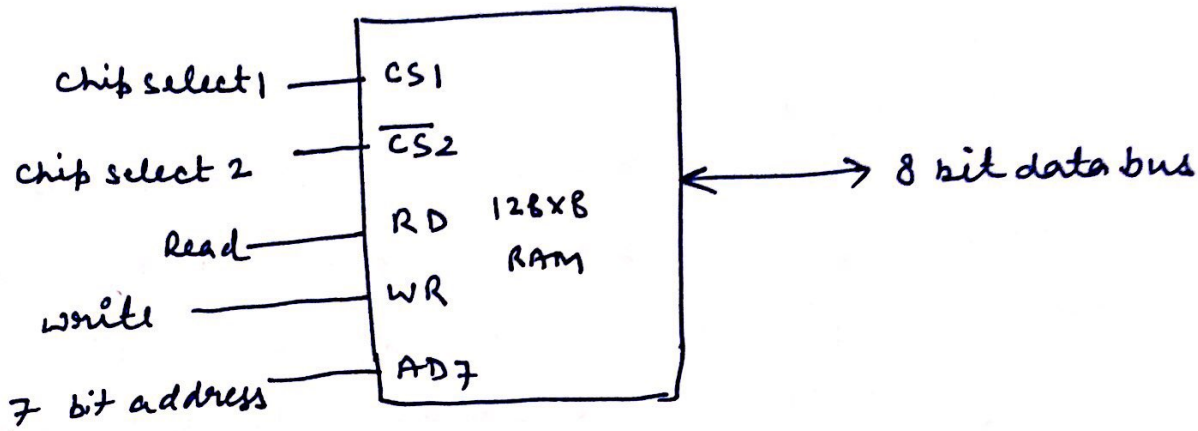
LECTURE

RAM and ROM Chips:

RAM and ROM chips are available in a variety of size. If the memory needed for the computer is larger than the capacity of one chip. It is necessary to combine a number of chips to form the required memory size.

RAM Chip: The figure shows a RAM chip. It has one or more control inputs that select the chip only when needed.

- It operates only when $CS1 = 1$ and $\overline{CS2} = 0$.
- If the chip select inputs are not enabled, or if they are enabled but the read or write inputs are not enabled, the memory is inhibited and its data bus is in a high impedance state.
- RAM chip has bidirectional data bus that allows the transfer of data either from memory to CPU (Read operation), or from CPU to memory (write operation).
- RAM chip has 7 No. of address lines which are used to select a specific word from the RAM chip of 128 bytes ($128 = 2^7$)



Typical RAM chip

ROM chip :

The figure shows a ROM chip. It has one or more control inputs that select the chip only when needed.

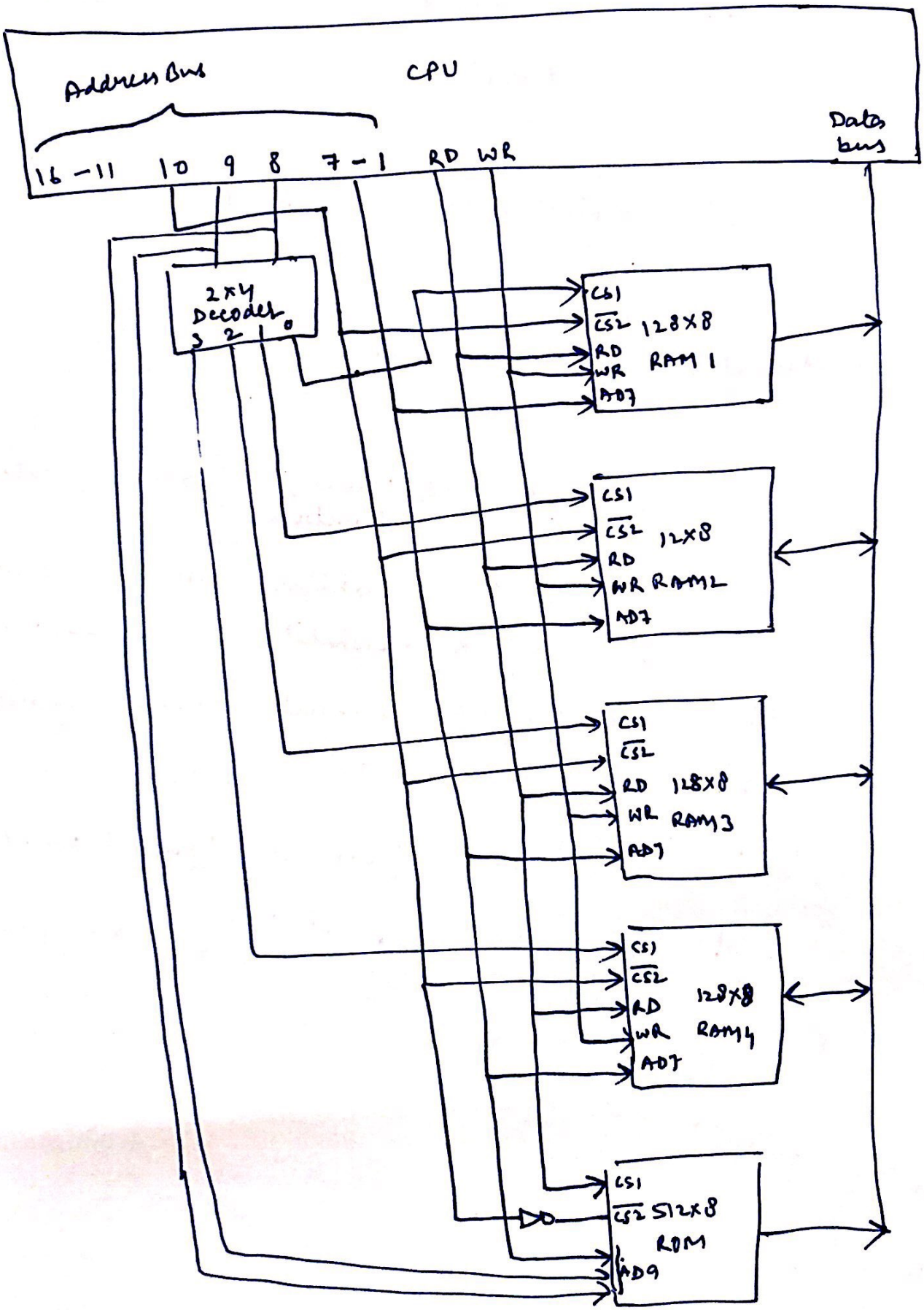
- It operates only when CS1 = 1 and $\overline{CS2}$ = 0
- It does not required read or write control because the unit can only read. Thus when the chip is enabled by the two select inputs, the data of the location selected by the address lines appears on the data bus.
- For the same size of RAM or ROM chip, it is possible to have more word in ROM, because the internal binary cells in ROM occupies less space than in RAM. For this reason, the diagram

Typical ROM - chip - Block diagram

Functional Table For RAM chip

CS1	$\overline{CS2}$	RD	WR	Memory Function	state of data bus
0	0	X	X	Inhibit	High Impedence
0	1	X	X	Inhibit	High Impedence
1	0	0	0	Inhibit	High Impedence
1	0	0	1	write	Input data to RAM
1	0	1	0	read	Output data from RAM
1	1	X	X	Inhibit	High Impedence

Memory connection to the CPU :



Memory Address Map

Component	Hexadecimal address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM1	0000 - 007F	0	0	0	x	x	x	x	x	x	x
RAM2	0080 - 00FF	0	0	1	x	x	x	x	x	x	x
RAM3	0100 - 017F	0	1	0	x	x	x	x	x	x	x
RAM4	0180 - 01FF	0	1	1	x	x	x	x	x	x	x
ROM	0200 - 03FF	1	x	x	x	x	x	x	x	x	x

memory capacity of 2048 bytes?

(b). How many lines of the address bus must be used to access 2048 bytes of memory? How many of these lines will be common to all chip.

(c). How many lines must be decoded for chip select? specify the size of the decoders.

Ans: (a) Number of RAM chips needed = $\frac{2048 \times 8}{128 \times 8}$

= 16 chips of 128×8 are needed.

(b) No. of address lines required (N)

$= 2048 = 2^{11}$

$\boxed{N = 11}$

No of lines common to all chips (m) $= 128 = 2^7$

$\boxed{M = 7}$

(c). No. of lines for chip selection = 4 as there are 16 chips size of decoder 4×16.

Q. A Computer uses RAM chips of 1024×1 capacity.

(a) · How many chips are needed and how should their address lines be connected to provide a memory capacity of 1024 bytes.

(b). How many chips are needed to provide a memory capacity of 16K bytes? Explain in words how the chips are to be connected to the address bus.

Ans. (a) No. of chips required = $\frac{1024 \times 8}{1024 \times 1}$ = 8 chips.

∴ 8 chips of 1024×1 are required in parallel to provide a memory capacity of 1024 bytes.

(b). No. of chips required = $\frac{16 \times 1024 \times 8}{1024 \times 1}$

= 128 chips are required.

Total No of lines required 16K = 2^{14} ∴.

14 address lines are required

10 lines are used to specify chip address.

4 lines are used to specify chip number.

Q. Extend the memory system of Fig(12-4) (512×8 ROM, 128×8 RAM chips are available) to 4096 bytes of RAM and 4096 bytes of ROM. list the memory address map and indicate what size decoders are needed.

Sol: No of RAM chips needed = $\frac{4096 \times 8}{128 \times 8}$ = 32 RAM chips

No of ROM chips needed = $\frac{4096 \times 8}{512 \times 8}$ = 8 ROM chips

Total Memory = 4096 + 4096 = 8192 = 2^{13}

∴ Total 13 Address lines are required

- 1 is used to select the RAM or ROM.
- 12 lines of RAM are used as

  7 lines are common to all RAM.
  5 lines are used to select the RAM chip
  + (5×32) decoder is used.

- 12 lines are used for ROM as

  9 lines are common to all ROM.
  3 lines are used to select the ROM chip + (3×8) decoder is used.

Memory address map:

Component	Add. in Hexadecimal	Address lines 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
RAM	0000 - 0FFF	- - - 0 5×32 decoder x x xxxxx
ROM	1000 - 1FFF	3×8 decoder x xxxxxxxx

Q. A computer employs RAM chips of 256×8 and ROM chips of 1024×8. The computer system needs 2K bytes of RAM. 4K bytes of ROM and 4 interface units. each with 4 registers. A memory mapped I/O configuration is used. The two highest order bits of the address bus are assigned 00 for RAM, 01 for ROM and 10 for interface registers.

(a) How many RAM and ROM chips are needed

(b) Draw a memory-address map for the system

(c). Give the address range in Hexadecimal for RAM, ROM and interface.

(a) No. of RAM chips needed = $\frac{2 \times 1024 \times 8}{256 \times 8}$ = 8 RAM chips of 256 × 8 are required.

No. of ROM chips needed = $\frac{4 \times 1024 \times 8}{1024 \times 8}$ = 4 ROM chips of 1024 × 8 are required.

(b) No of address lines required for

RAM = 2048 = 2^{11}

→ 8 lines are common to all chips.

→ 3 lines are used for chip selection + (3×8) decoder is used

ROM = 4096 = 2^{12}

→ 10 lines are common to all chips

→ 2 lines are used for chip selection + (2×4) decoder is used

Interface = 4 × 4 = 16 = 2^4

Component	Hexadecimal Address	Address Bus 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
RAM	0000 - 07FF	0 0 0 0 0 (3×8 decoder) x x x x x x x x
ROM	4000 - 4FFF	0 1 0 0 (2×4 decoder) x x x x x x x x x x
Interface	8000 - 800F	1 0 0 0 0 0 0 0 0 0 0 0 x x x x

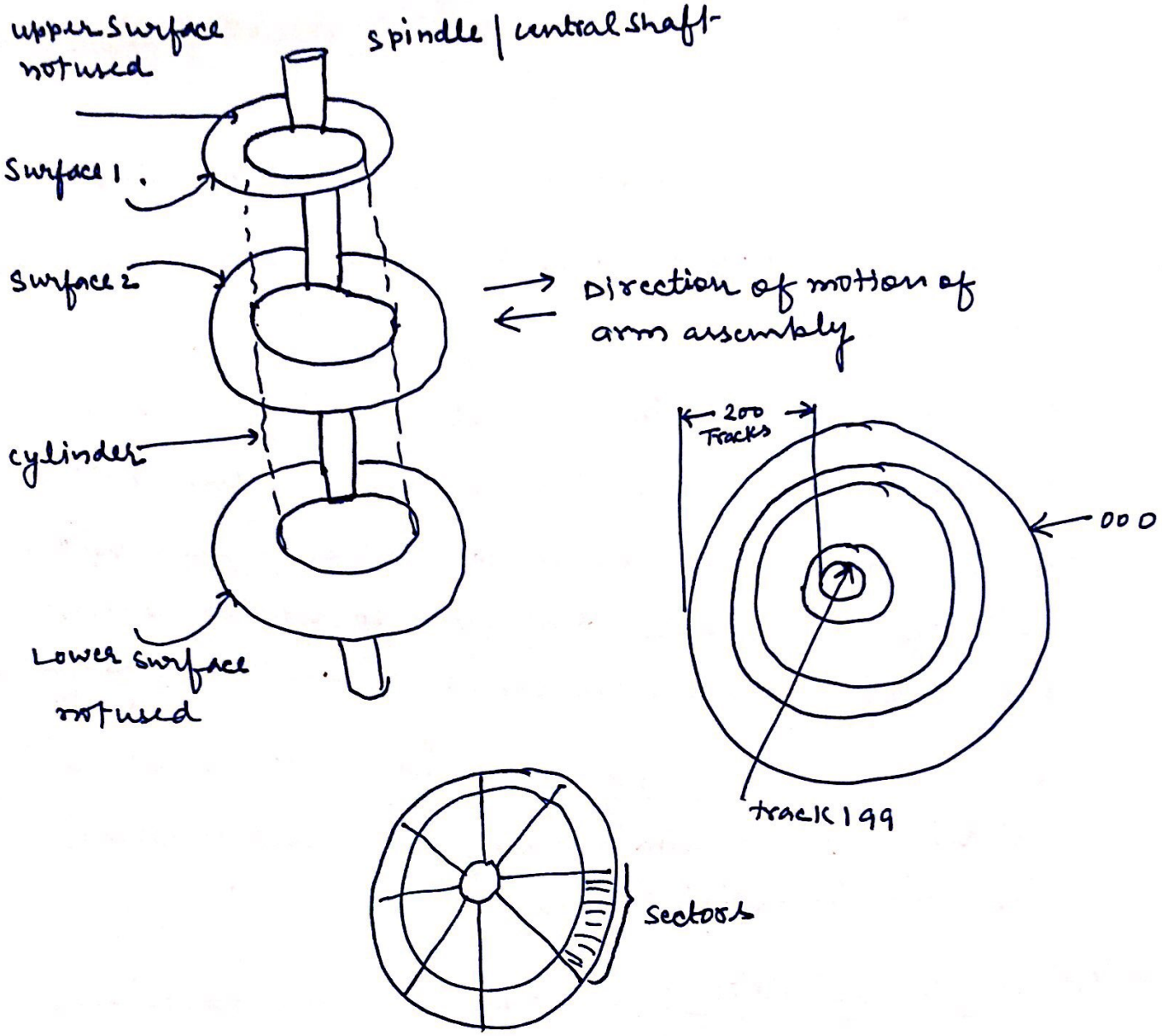
LECTURE

Magnetic Hard Disk : Magnetic disks are thin, circular metal plates coated on both the sides with a thin film of magnetic material.

- A set of such magnetic disks/plates are fixed to a spindle one below the other to make up a disk pack.
- The disk pack is sealed and mounted on a disk drive.
- The disk drive consists of a motor to rotate the disk pack about its axis at a speed of about 3600 rev/min.
- The drive also has a set of magnetic heads mounted on arms. The arm assembly is capable of moving in & out in radial direction.

Storage of Information : Information is recorded on both the surfaces of each disk plate, as it rotates about axis. However, the upper surface of the top plate & lower surface of the bottom plate are not used.

- Each disk consists of
- A set of corresponding tracks in all surfaces of a disk pack is called a cylinder.



Surfaces ∴ It has 18 tracks | cylinder

Each track is divided into sectors.

The total no. of bytes that can be stored in a disk pack is = no. of cylinder x tracks | cylinder x sectors | track x Bytes per sector.

Eg: Disk pack has 10 disk plates each having 200 tracks.

⇩

18 recording surfaces ⇒ 18 tracks | cylinder

Suppose, there are 40 sectors/track & each sector can store 256 bytes.

200 tracks/plate ⇒ 200 cylinders

∴ capacity of disk pack = 200 × 18 × 40 × 256

= 31864000 bytes

= 36.864 M bytes

- A set of disk drives are connected to a disk controller. The disk controller accepts commands from the computer & positions the read-write heads of the specified disk for reading or writing.

- In order to read from or write on a disk, the computer must specify the drive no., cylinder no., surface no. & the sector no.. The drive no. should be specified because a controller normally controls more than one drive.

When a read-write command is received by the disk controller, the controller first positions the arm assembly so that the read-write head reaches the specified cylinder. The time taken to reach the specified cylinder is known as the seek-time (Ts). The seek time, the rate at which information is read from the disk

$\therefore$ transfer rate $= 10260 \times \frac{3600}{60}$ bytes/sec.

$= 615600$ bytes/sec.

Time to read one sector is $\frac{256}{615600} = 4.158$ msec.

$\therefore$ Avg time to access a section is $30 + 8.3$ sec.

Compact Disk - Read only Memory (CD-ROM)

Also known as a Laser disk. It is a shiny metal like disk whole diameter is 5.25". It can store around 650 megabytes (equivalent to 2,50,000 pages) info. In CD-ROM is written by creating pits on the disk surface by shinning a laser beam.

As the disk rotates the laser beam traces out a continuous spiral. The sharply focused beam creates a circular pit of around 5×10^{-4} mm wherever a 1 is to be written & no pit (also called a land) if a zero is to be written.

LECTURE

Content : Cache performance Issue..

Cache performance Issue :

Cache Hit : When CPU needs to read data/instruction from memory, first cache is examined, If the word is found in cache, it is called cache Hit.

Cache Hit Time : In case of a cache hit, the required word is read from the cache, This time to access data from cache is cache hit time.

Cache Miss : If the requested data is not found in cache memory, then a cache miss is said to occur.

Cache miss Time Penalty : In case of cache Miss, the time required to fetch the required block from main memory (to deliver to CPU) is called cache miss time penalty

Hit Ratio (h):

$$= \frac{\text{No. of Hits}}{\text{No. of Cache Hits} + \text{No. of Misses}}$$

Miss Ratio (1-h):

$$= \frac{\text{No. of cache Misses}}{\text{No. of Cache Hits} + \text{No. of cache Misses}}$$

Average Memory Access Time:

$$t_A = h\,t_c + (1-h)(t_c + t_m)$$

- t_A → Average Memory Access Time
- h → Hit Ratio
- t_c → Cache Memory Access Time
- $(1-h)$ → Miss Ratio
- $(t_c + t_m)$ → Cache Memory access time + Main Memory access time

Suppose, the ratio of main memory access time to cache access time is γ, then effeciency (η) of a system that employs a cache is

$$\eta = \frac{t_c}{t_A}$$

$$\eta = \frac{t_c}{h t_c + (1-h)(t_c + t_m)}$$

$$= \frac{t_c}{t_c\left(h + (1-h)\left(1 + \frac{t_m}{t_c}\right)\right)}$$

$$= \frac{\cancel{t_c}}{\cancel{t_c}\left(h + (1-h)(1+\gamma)\right)}$$

$$= \frac{1}{h + (1-h)(1+\gamma)}$$

$$= \frac{1}{\cancel{h} + 1 - \cancel{h} + \gamma(1-h)}$$

$$\boxed{\eta = \frac{1}{1 + \gamma(1-h)}}$$

η is max when $h = 1$. i.e all reference are from a cache.

Q. Calculate t_A (average access time), γ (ratio of main memory access time to cache memory access time) and η (effeciency) of a memory system whose parameters are as indicated:

$t_c = 160$ ns

$t_m = 960$ ns

$h = 0.90$

Sol:

$$t_A = h t_c + (1-h)(t_c + t_m)$$

$$= 0.90 \times 160 + (1 - 0.90) \times (160 + 960)$$

$$= 144 + (0.1) \times 1120$$

$$= 144 + 112$$

$$\boxed{t_A = 256 \text{ ns}}$$

$$\gamma = \frac{t_m}{t_c} = \frac{960}{160} = 6, \text{ so } \boxed{\gamma = 6}$$

$$\eta = \frac{1}{1 + \gamma(1-h)}$$

$$= \frac{1}{1 + 6(1 - 0.90)}$$

$$= \frac{1}{1.6} = 0.625$$

$$\boxed{\eta = 0.625}$$

8. The access time of a cache memory is 50 n sec and that of the main memory is 500 ns. It is estimated that 80% of the main memory requests are for read operation and the remaining are for write. The hit ratio for read access only is 0.9 and a write through policy is used.

(a). What is the average access time of the system considering only memory read cycles?

(b). What is the average access time of the system for both read and write requests?

(c) What is the hit ratio taking into consideration the write cycle?

Sol:

(a) average access time = $t_A = h \cdot t_c + (1-h) * (t_c + t_m)$

$= 0.9 \times 50 + (1 - 0.9) * (50 + 500)$

$= 45 + 55$

$\boxed{t_A = 100 \text{ ns}}$

$P_R \Rightarrow$ % of read operation

(b) Average access time = P_R * Avg read time + $(1-P_R)$ * t_m

$= P_R * [\quad] + (1 - P_R) \times 500$

h_r → hit ratio for read cycle
h_w → hit ratio for write cycle

© Hit Ratio = $P_R \times h_r + (1 - P_R) * h_w$

$= 0.8 \times 0.9 + (1 - 0.8) * 0$

$= 0.72$

Virtual Memory : In a memory hierarchy system, programs and data are first stored in auxiliary memory. Portion of a program or data are brought into main memory as they are needed by the CPU.

Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large memory space were available, equal to the totality of auxiliary memory. Each address that is referenced by the CPU goes through an address mapping from the so-called virtual address to a physical address in main memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small memory.

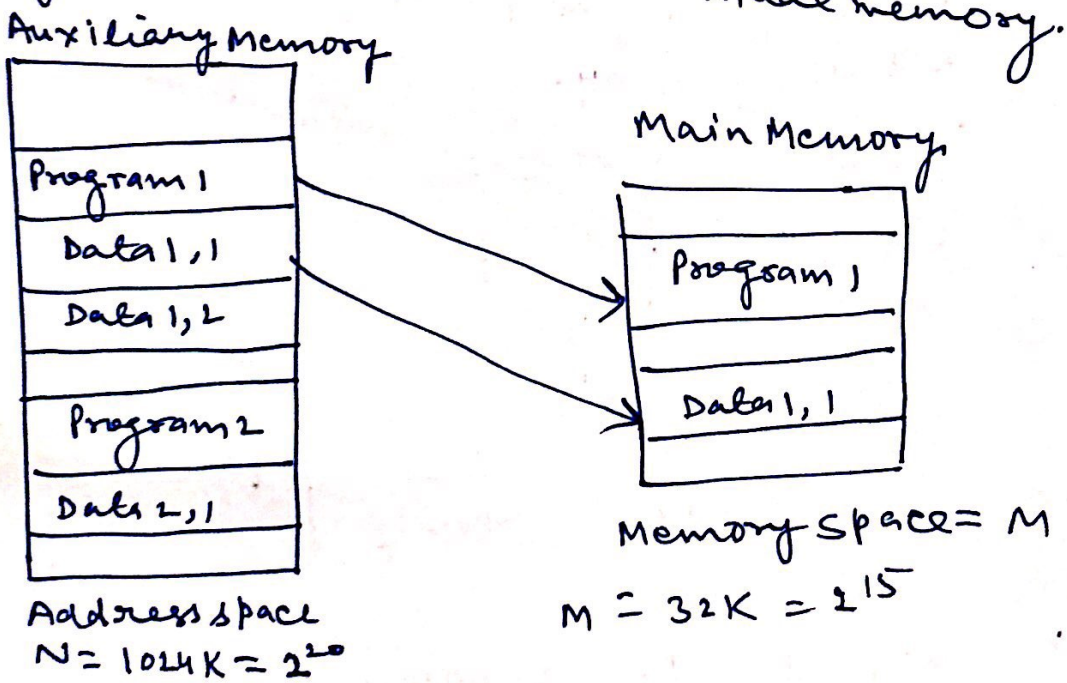
A virtual memory system provides a mechanism for translating program-generated addresses into correct main memory locations. This is done dynamically, while programs are being executed in the CPU. The translation or mapping is handled automatically by the hardware by means of mapping Table.

Address Space and Memory Space :

An address used by a programmer will be called a virtual address, and the set of such addresses the address space.

An address in main memory is called a location or physical address. The set of such locations is called the memory space.

Thus the address space is the set of addresses generated by programs as they reference instructions and data; the memory space consist of the actual main memory locations directly addressable for processing. In most computers, the address and memory spaces are identical. The address space is allowed to be larger than the memory space in computers with virtual memory.

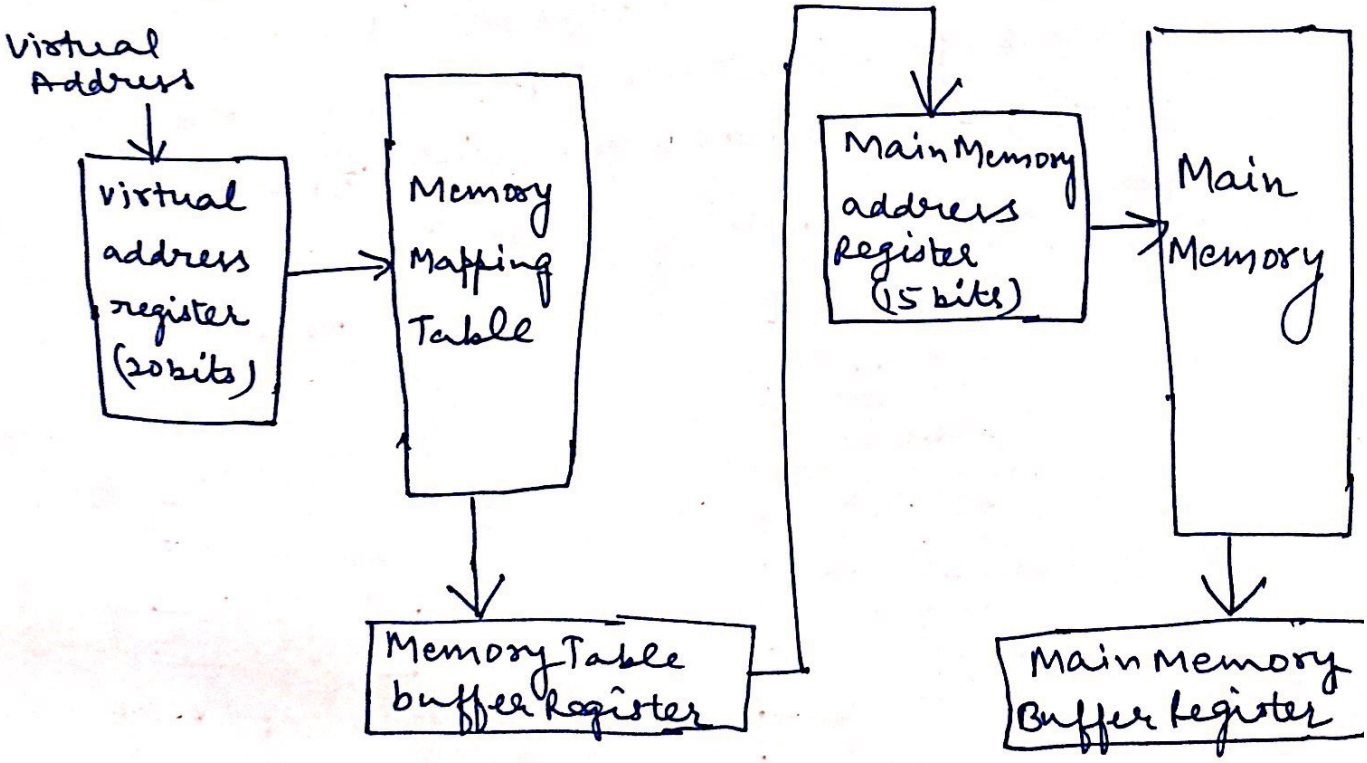


Relation b/w address and Memory space in a virtual Memory System

Mapping Table: A table is needed, as shown in figure, to map a virtual address of 20 bits to a physical address of 15 bits. The mapping is ~~dynamically~~ dynamic operation, which means that every address is translated immediately as a word is referenced by CPU.

The mapping table may be stored in a separate memory. In the first case, an additional memory unit is required as well as one extra memory access time.

In second case, the table takes space from main memory and two accesses to memory are required with the program running at half speed.



Memory table for Mapping a virtual Address

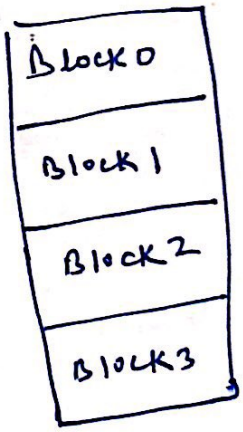
Address Mapping using Pages

The table implementation of the address mapping is simplified if the information in the address space and the memory space are each divided into groups of fixed size. The physical memory is broken down into groups of equal size called blocks.

Consider a computer with an address space of 8K and a memory space of 4K. if we split each into group of 1K words we obtain eight pages and four blocks.

Address space

$N = 8K = 2^{13}$

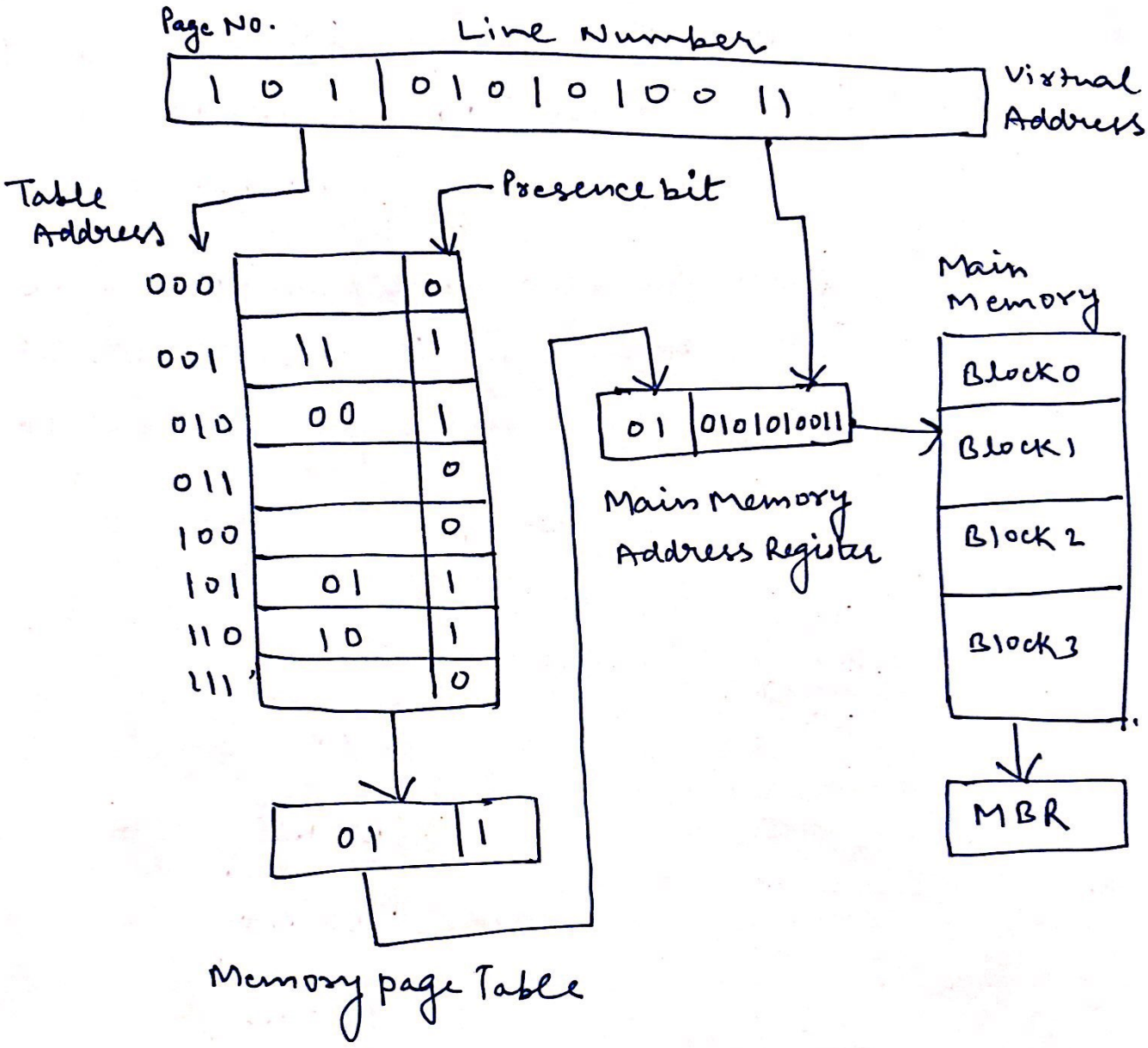


Memory space

$M = 4K = 2^{12}$

Address space and Memory space split into groups of 1K words

The organization of memory mapping table in a paged system is shown in figure



Memory table in a paged systems

Page Replacement :

The virtual memory system is a combination of hardware and software techniques. The memory management software system handles all the software operations for the effcient utilization of memory space.

It must decide :

1- which page in main memory ought to be removed to make room for a new page.

2- when a new page is to be transferred from auxiliary memory to main memory.

3- where the page is to be placed in main memory.

The hardware mapping mechanism and the memory management software together constitute the architecture of a virtual memory.

Page fault :

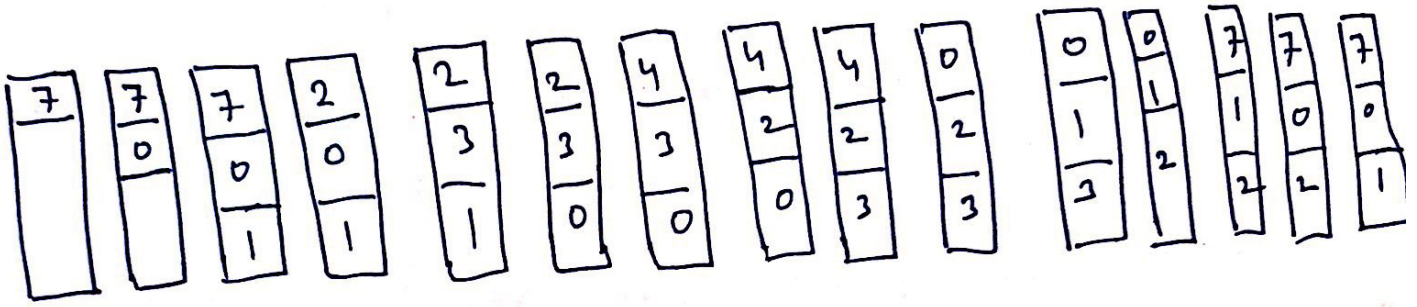
When a program starts execution, one or more pages are transferred into main memory and the page table is set to indicate their position. The program is executed from main memory until it attempts to reference a page that is still in auxiliary memory. This condition is called page fault.

When page fault occurs, the execution of the present program is suspended untill the required page is brought into main memory.

There are two replacement algorithms used:

FIFO (First-In First-Out):

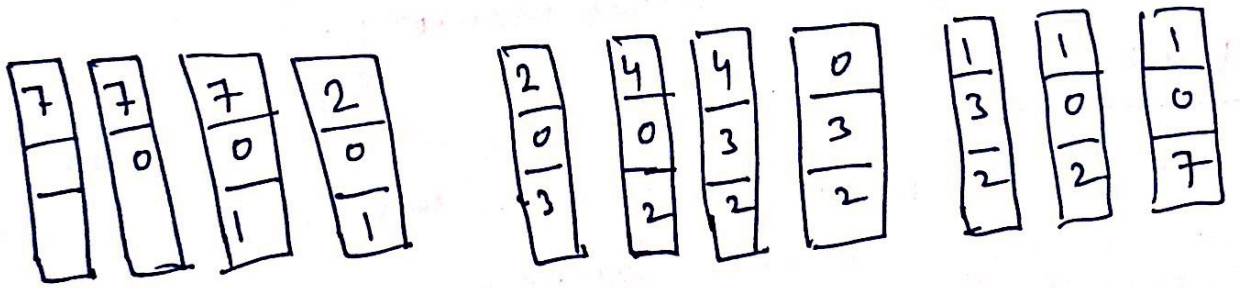
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



FIFO Page Replacement Algorithm

LRU (Least-Recently-used):

7 0 ~~1~~ 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



LRU page Replacement algorithm