COA -
Unit 4

MEMORY HIERARCHY.

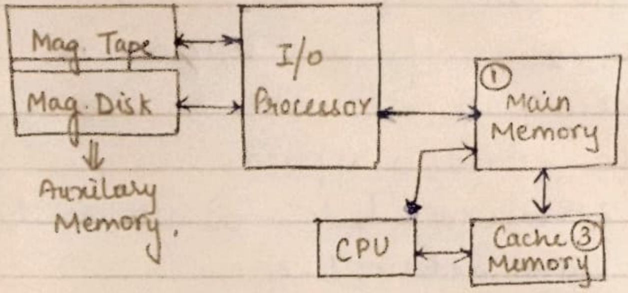
Memory: → essential component

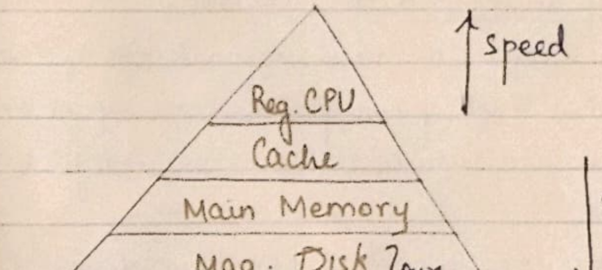
→ holds/stores program instructions, data (info) & operands, and calculat^n

→ CPU: controls info stored in memory

→ Info is stored, fetched, manipulated (under program control) and written into memory for immediate/later use

→ A very small comp. with a limited applicat^n may be able to fulfil its intended task without need of addit^{n}al storage capacity





Memory Hierarchy:

→ consists of all storage devices employed in a computer system from slow but high capacity (Auxiliary) to fast but low capacity memory (cache) ~~(CPU)~~

→ **I/O Processor** → manages data/info transfer b/w aux memory & main memory

→ **Cache Memory** → used by CPU for holding ~~variables~~ data in memory unit that is being accessed again & again

→ acts as a buffer b/w CPU & main memory

→ special & very high speed

→ **Registers** → used for holding variables & temporary results.

→ very small storage but high speed i.e can be accessed immediately

→ **Main Memory** (internal memory) → large & fairly fast

→ stores data & program

→ communicate directly with the CPU.

→ storage location is directly addressed by CPU's load & store instruct^n

→ Access Time is larger because of its large capacity

→ physically seperated from CPU

→ **Secondary / Auxiliary Memory**

→ giant in capacity but slower than all other types of memory.

→ stores large data files, programs & files that are not required by CPU continuously.

→ acts as an overflow memory when the capacity of main memory is exceeded

→ provided by peripheral devices

→ **Overall goal of memory Hierarchy**

→ to ~~op~~ obtain highest possible avg. access speed while minimizing total cost of entire memory system

*Characteristics on which memory devices & techs are compared

① Storage Capacity

→ representative of size of memory

→ Expressed in words / bytes

② Unit of Transfer

→ no of bits read/written in single read/write operatⁿ.

→ Depends on data line.

③ Access Time

→ Time in b/w request made for operatⁿ and the time data is available

→ Depends on (a) physical characteristics & (b) access mode

④ Permanence of Storage

→ loss of info over period of time

→ destructive failure, volatile behavior etc

⑤ Accessing modes

→ accessing info from memory

→ Ways (i) Random (cache) (ii) Direct (Disk / CD-ROM) (iii) Sequential (Tapes) (iv) Associative (content addresablemmro)

⑥ Cycle time

→ minimum time elapsed b/w two consecutive read request.

→ Same as access time but includes refresh cycle time also

⑦ Data Transfer → measured in (bps)

→ Amt of info transferred per unit time.

⑧ Physical Characteristics

Type of memory	Access mode	Permanence of storage	Physical Storage
① Semicond.	Random	Volatile	Electronic
② Mag. Disk	Direct	N. Volatile	Mag
③ Mag Tape	Sequent.	N. Volatile	Mag.
④ CD-ROM	Direct	"	Optical

* Main Memory (Semiconductor)

→ central storage unit of comp. system

→ large & fast

→ stores data & program during comp opot

→ based on semiconductor integrated chips

→ Integrated RAM are available in 2 modes

(i) Static (ii) Dynamic.

① Static RAM (SRAM)

→ 4 transistors (T_1, T_2, T_3, T_4) are cross connected

- Logic State 1 $C_1 \rightarrow$ high $C_2 \rightarrow$ low then T_1 & T_2 are off &, T_3 & T_4 are on
- Logic State 0 $C_2 \rightarrow$ high $C_1 \rightarrow$ low T_3, T_2 are off & T_1, T_2, T_4 are on

→ both state are stable as long as dc voltage is applied

→ 2 transistors (T_5, T_6) are used to control address line

② Dynamic RAM (DRAM)

→ uses capacitor to store each bit of data, & the level of charge on each capacitor determines whether that bit is logical 1 or 0

→ Capacitor do not hold their charge indefinitely & ∴ data needs periodic refreshment

→ However these capacitors do not hold charge indefinitely, & ∴ d used in equipment including p.c.s & workstations where it forms the main RAM for computer.

→

SRAM	DRAM
→ stores data & program as long as power is 'ON'.	→ looses stored info even though power supply is 'ON'.
→ made up of transistors & flip flops	→ made up of capacitors & few transistors
→ for single block of memory 6 transistors	→ for single block of memory only 1 transis
→ no charge ~~teda~~ leakage property so does not need to be power refresh	→ has charge leakage property so it need to be refreshed after each read operat^n
→ utilizes less power	→ utilizes more power
→ expensive	→ cheaper than SRAM
→ faster	→ slower " "
→ low density	→ high density
→ ~~avat~~ available in smaller storage ~~ca~~ capacity of few M.B	→ usually available in large storage capacity of few G.B
→ Stores data in form of voltage.	→ stores data in form of Charge
Eg Cache	Eg DDR, DDR$_2$, DDR$_3$ etc

(*) ROM(Memory)

→ type of Semiconductor memory that is designed to hold data that is either permanent or will not change freq.

→ non volatile.

→ TYPES → ROM : Read only Memory
→ PROM : Programmable ROM
→ EPROM : Erasable " "
→ EEPROM : Electrically Erasable PROM
→ Flash EEPROM memory

PROM → one time programmable ROM

→ programed via PROM programmer

→ This devices uses high voltage to permanently destroy or create internal links within a chip.

→ can only be programmed once

EPROM → can be erased when exposed to UV

→ then rewritten ~~when~~ with a process that requires applicat^n of higher voltage than usual

→ can be rewritten many times

EEPROM → f^{n}ally similar to EPROM

→ can be erased whenever needed

→ Erasing is done electrically by applying voltage of appropriate ~~vo~~ polarity & amplitude.

→ common cells are composed of 2 transisto

In EPROM
→ Storage transistor has floating gate whereas in ~~EPROM~~ EEPROM cell is erased when e$^-$s are trapped in floating cell

FLASH Memory Technology → mix of EPROM & EEPROM

→ term 'FLASH' → ∵ large chunk of memory could be ~~sup~~ erased at a time

→ mature technology

→ strong competitor to other non-volatile memory

ROM Memories

→ main memory in a general-purpose comp. is made up RAM integrated circuits chips, but a portion of the memory may be ~~const~~ constructed with ROM Chip

→ RAM → storing bulk of the programs & data that are subject to change

→ is volatile

- ROM → used for storing programs that are permanently resident in the comp & for table of constants that do not change in value once the productⁿ of comp. is completed

→ ROM portion of main memory is needed for storing an initial program called a BOOTSTRAP LOADER.

↳ it is a program whose fⁿ is to start the computer software operating when power is turned on.

↳ Bootstrap program loads a portion of the operating system from disk to main memory and control is then transferred to the operating system, which prepares the computer for general use.

RAM & ROM Chips

→ RAM & ROM chips are connected to a CPU through the data & address bus.

→ In diagram → Memory connectⁿ to CPU using ~~2x4 decoder~~ 4 RAM & 1 ROM

→ Address line 8,9

0 0	→	RAM 1
0 1	→	RAM 2
1 0	→	RAM 3
1 1	→	RAM 4

- RAM Chip

→ suited for communicatⁿ

→ bidirectional data bus
⇓
allows transfer of data

(i) Mem to CPU (read operatⁿ)

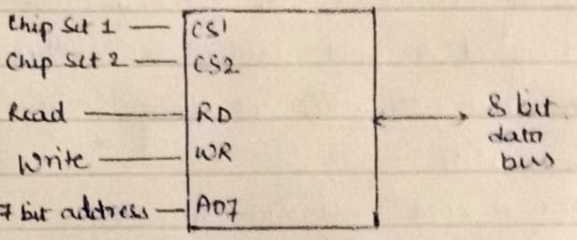
(ii) CPU to Mem (write operatⁿ)

→ A bidirectⁿal bus can be constructed with 3 state buffers

(i) Signal equivalent to logic 1 } Normal Signals
(ii) Signal equivalent to logic 0 } Normal Signals
(iii) High Impedence state / no logic
⇓
- behaves like an open circuit ie. output doesn't carry a signal

→ Capacity of memory

- 128 words of 8 bit /words ie. requires 7 bit address
$2^7 = 128$
- 8 bit bidirectⁿal data bus



RAM Chip Block Diagram

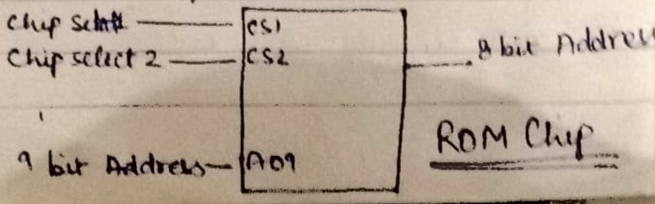
- ROM Chip

→ Only for Read operatⁿ

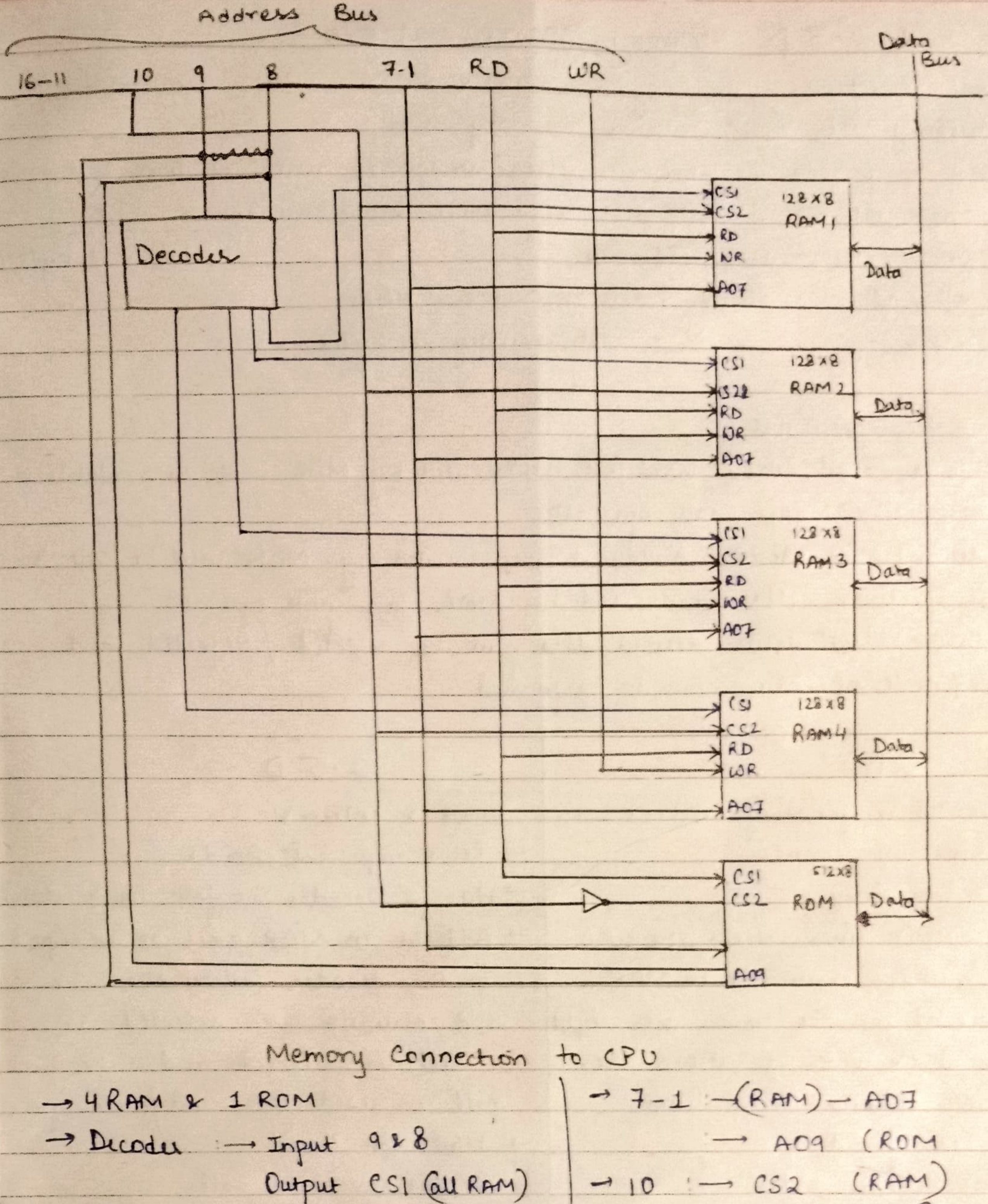
→ Unidirectⁿal ⇒ Mem to CPU

→ Capacity of Memory
- 128 words of 8 bit/w
- 9 bit unidirectⁿal data bus

→ can only read, ∴ data bus can only be in output mode



ROM Chip



Memory Connection to CPU

→ 4 RAM & 1 ROM

→ Decoder → Input 9 & 8
Output CS1 (all RAM)

→ WR → WR all RAM
→ Not conn (ROM)

→ RD → RD all RAM
→ CS1 (ROM)

→ 7-1 → (RAM) – A07
→ A09 (ROM

→ 10 :→ CS2 (RAM)
:→ CS2 Do (ROM)

- Select^{n} b/w RAM & ROM is achieved through bus line 10
1 → ROM & 0 → RAM is selected.

⊛ 2D and 2.5D Memory Organization

→ Internal structure of memory has RAM & ROM which are made up of memory cell that contains memory bit

→ Memory is present in form of multidimensional array of rows & columns where each cell stores a bit and a complete row contains a word.

→ Memory can be represented as

$\boxed{8 \text{ bits} = 1 \text{ byte}}$

$\underbrace{2^n}_{\text{words}} = N$ where n → no of address lines

N → Total memory in bytes

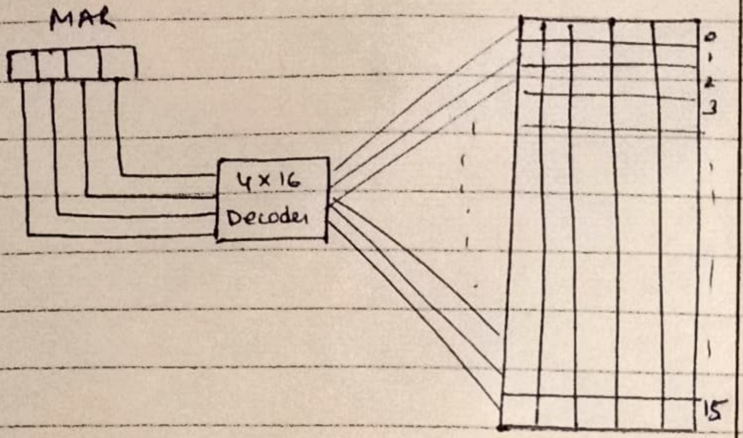
- **Read & Write Operations**

① Select^n lines (Read Mode) : Word/bit represented by MAR will be available to data lines for read operat^n

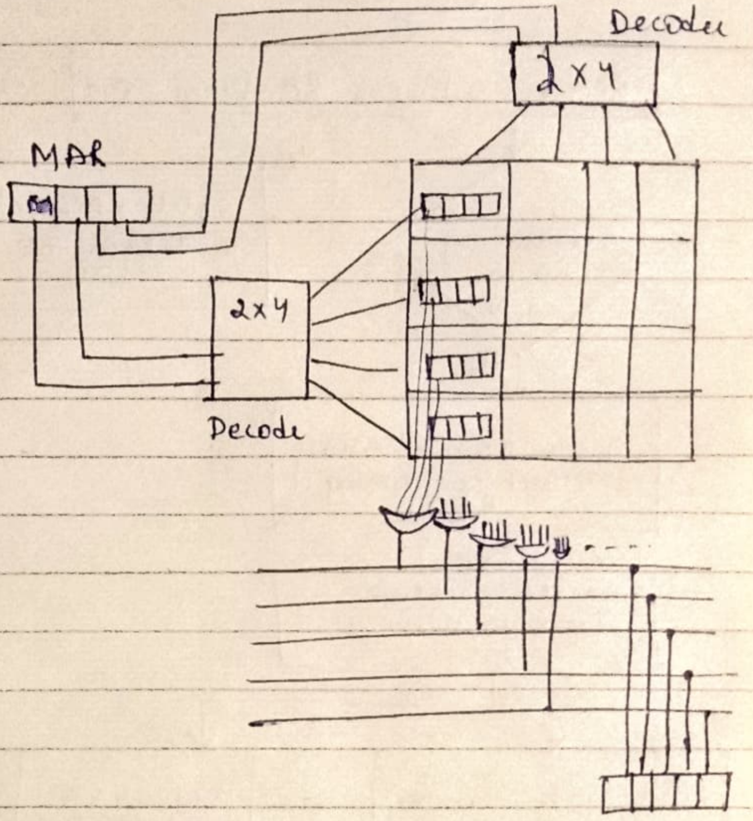
② Select^n lines (Write Mode) : Word/bit represented by MDR will be sent to respective cells addressed by MAR for write operat^n.

③ With these select^n lines, desired data can be selected/rejected and read/write operation can be performed.

2D	2.5D
→ Represented in form of Matrix i.e rows & colums	
→ Rows represent words	→ Rows represent words
→ Has 1 Decoder	→ Has 2 Decoder i.e for row & column
→ One of output lines selects the row using address contained in MAR, represented by the row, gets selected and read or write operations are performed through data lines	→ Address in MAR goes as an input in the decoder, then rows and columns gets selected i.e a cell is selected, and data from cell is used for read/write operat^n.
→ H/w is fixed	→ H/w not fixed
→ Requires more gates	→ Requires less gates
→ More complex	→ less complex
→ Error correct^n is difficult/impossible	→ Error correct^n is easy
→ More difficult to fabricate	→ less difficult to fabricate.



2D

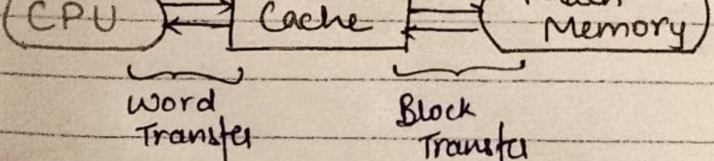


2·5 D

CACHE MEMORY

→ If a active portion of program/data are placed in fast small memory, the avg memory access time can be reduced also the total execution time of a prog. gets reduced. This fast small memory is referred to as CACHE MEMORY.

→ placed b/w CPU & Main Memory



→ Fundamental Idea → keeping the most freq. accessed instructions & data in the fast cache memory, and the mem. access time & total execut^n time reduces

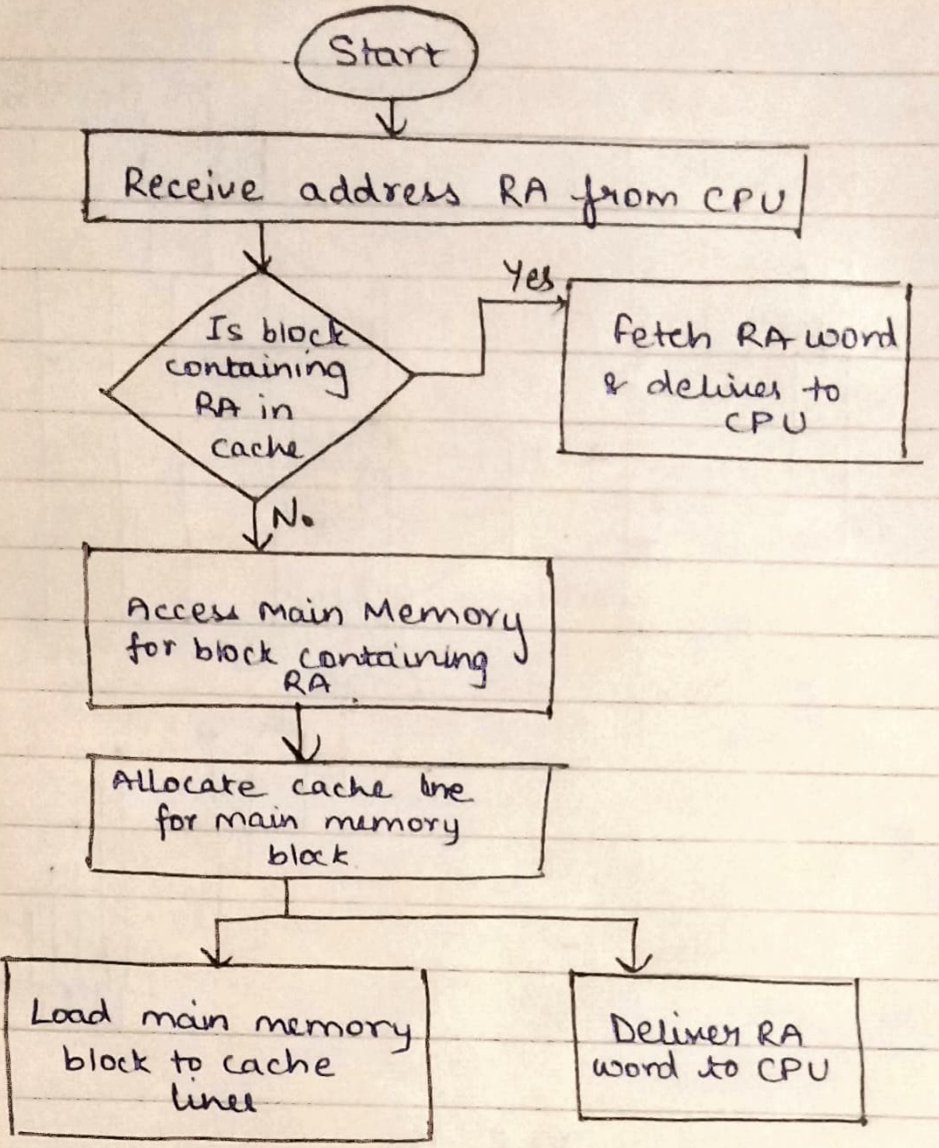
→ Basic Operation → When CPU needs to access memory, the cache is examined.

→ If word is found in cache : it's read.

→ If not found : a block of main memory containing the word is read in cache and the word is delivered to processor

→ F^n is hidden from program & user

→ high speed volatile memory


Start
Receive address RA from CPU
Is block containing RA in cache
Yes
fetch RA word & delives to CPU
No.
Access Main Memory for block containing RA
Allocate cache line for main memory block
Load main memory block to cache lines
Deliver RA word to CPU

Mapping Function

→ ∵ there are fewers cache lines than main memory blocks, ∴ an algo is needed for mapping main memory block into cache lines.

→ Types ① Direct Mapping
② Set Associative Mapping
③ Associative Mapping

⊛ Auxiliary Memory | Secondary

- Physical Properties : complex
- Logical Properties characterised on
  - (i) access mode
  - (ii) access time
  - (iii) transfer rate
  - (iv) Capacity
  - (v) Cost

• Magnetic Disk

→ circular plate (plastic/metal coated with magnetised material)

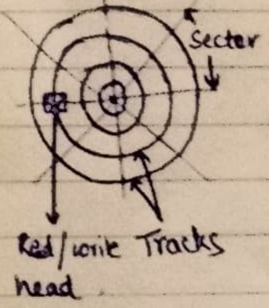
→ One/Both sides are used with read/write

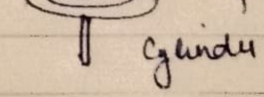
→ All disks rotate together at high uniform speed.

→ Bits are stored in magnetized surface in spots along concentric circles called TRACKS

→ Tracks are divided into sectns called SECTOR

→ The min. qty of info that can be transfered is a sector.





• Magnetic Tape

→ consists of electrical, mechanical mag, & electronic components

→ The tape itself is a strip of plastic coated with mag recording medium

→ Bits are recorded as magnetic spots on tape along several tracks uniformly

→ 7-9 bits are recorded simultaneously

→ Read/Write head are mounted on each track so that data can be recorded & read as a sequence of Characters

→ Low cost & for backup storage purposes

• Optical Disk

→ The disk contains a single spiral track beginning near the centre & spiraling out to the outer edge of the disk.

→ Sector near the outer edge of disk are same length as those near inside

→ Info is packed evenly across the disk in segments of same size & these are scanned at the same rate by rotating the disk at variable speed

→ Storage medium from which data is read and written is by LASER.

→ Can store upto 6 gbs

PAGE REPLACEMENT

Ref. String

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0.

→ seq. of Pages.

(*) **Page Replacement**

1) FIFO

2) Optimal Page Rep.

3) Least Recently used (LRU)

- **Page fault** → absence of page in main memory

f3			1	1	1	~~1~~	0	0	~~0~~	3	3	3	~~3~~	2	2
f2		0	0	0	0	3	3	~~3~~	2	2	2	~~2~~	1	1	1
f1	7	7	~~7~~	2	2	2	~~2~~	4	4	~~4~~	0	0	0	0	0
	*	*	*	*	Hit	*	*	*	*	*	*	Hit	*	X.	Hit

⇒ Page Hit = 3

⇒ Page Fault = 12 / miss

- Hit Ratio = $\frac{\text{Hit}}{\text{Total}}$ = $\frac{3}{15}$ = $\frac{1}{5}$

- Miss Ratio = $\frac{12}{15}$ =

				4	4	4	4	4	~~4~~	4	4	4	4
f3			3	3	3	3	3	3	~~2~~	3	3	3	3
f2		2	2	2	2	2	~~2~~	1	1	1	1	1	1
f1	1	1	1	1	1	~~1~~	5	5	5	5	5	5	5

* * * * . H . H * * * * Hit Hit

1 2 3 4 1 2 5 1 2 3 4 5

Hit = 4

f4				4	4	4	4	4	4	3	3	3
f3			3	3	3	3	3	~~3~~	2	2	2	2
f2		2	2	2	2	2	~~2~~	1	1	1	~~1~~	5
f1	1	1	1	1	1	~~1~~	5	5	5	~~5~~	4	4
					Hit	Hit						

2 Hit

f3			3	3	~~3~~	2	2	2	2	~~2~~	4	4
f2		2	2	~~2~~	1	1	1	1	~~1~~	3	3	3
f1	1	1	~~1~~	4	4	~~4~~	5	5	5	5	5	5
								H	H			Hit

Hit = 3

Fault

~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~1~~ ~~2~~ ~~5~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

* Optimal Page Replacement

↓

Replace page which is not used in longest Dimension of time in future

f_3				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
f_2			1	1	1	1	~~1~~	4	4	4	4	4	~~4~~	1	1	1	1	1	~~1~~	1
f_2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_1	7	7	7	7	~~7~~	3	3	3	3	3	3	3	3	3	3	3	3	7	7	7
					Hit		Hit		Hit	Hit	Hit	Hit	Hit		Hit	Hit	Hit		Hit	Hit

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

2, 1, 2,0, 1, 7, 0,1

↕

Hit = 12

fault = 8

$$\text{Hit} = \frac{\cancel{12}^{6}}{\cancel{20}} \times 10\cancel{0} = 60\%$$

$$\text{fault} = \frac{\cancel{8}^{4}}{\cancel{20}} \times 10\cancel{0} = 40\%$$

(*) Least Recently Used → least ~~recent~~ recently used page in past.

f_4				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
f_3			1	1	1	1	~~1~~	4	4	4	4	4	~~4~~	1	1	1	1	1	1	1
f_2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_1	7	7	7	7	~~7~~	3	3	3	3	3	3	3	3	3	3	3	~~3~~	7	7	7
					Hit		Hit		Hit	Hit	Hid	H	H		H	H	H		H	H

12

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