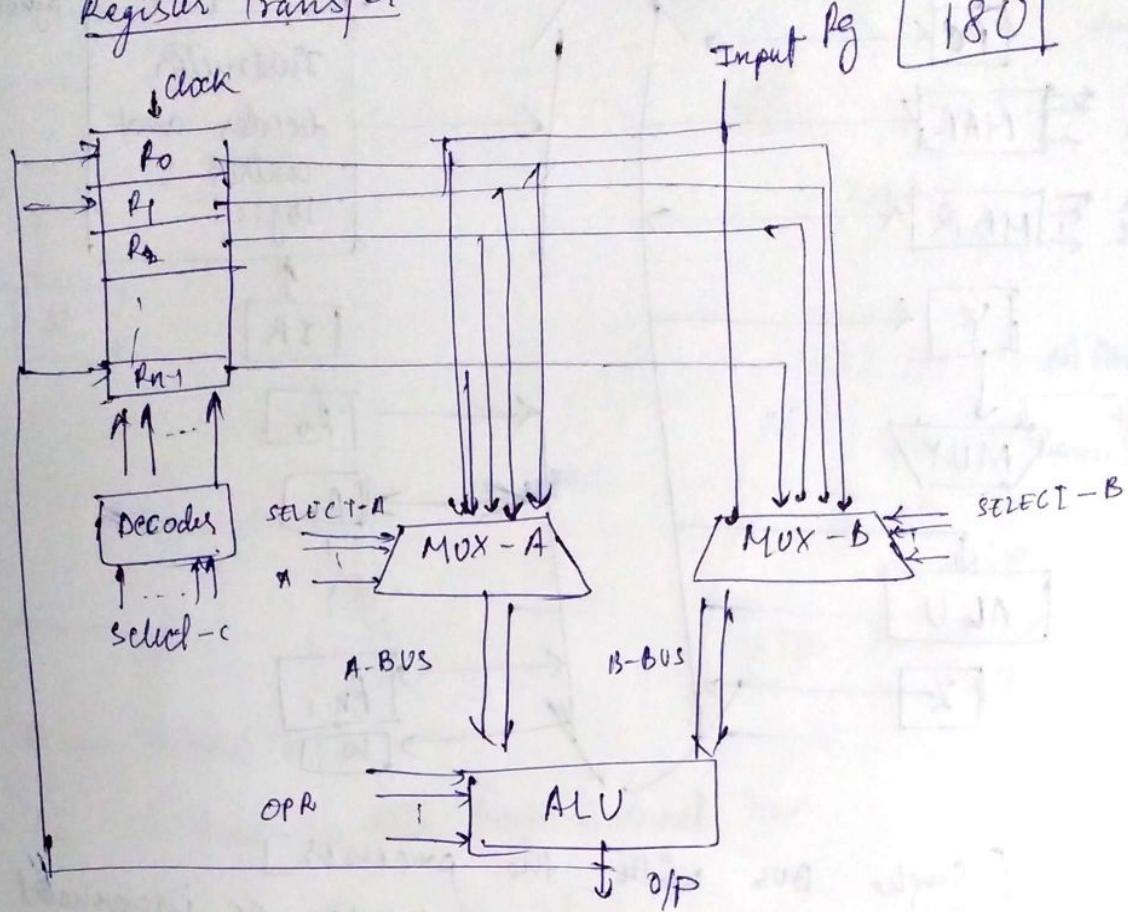


7/3/18

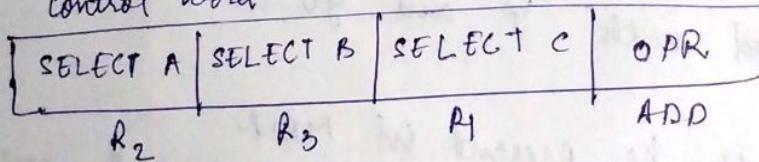
Register Transfer

Hannover
1801



[BLOCK DIAGRAM]

control word



$$A_1 \leftarrow R_2 + R_3$$

$$A_1 \leftarrow R_2 + I/P.$$

MUX-A will select - R_2

MUX-B will select - R_3

ALU will select - ADD

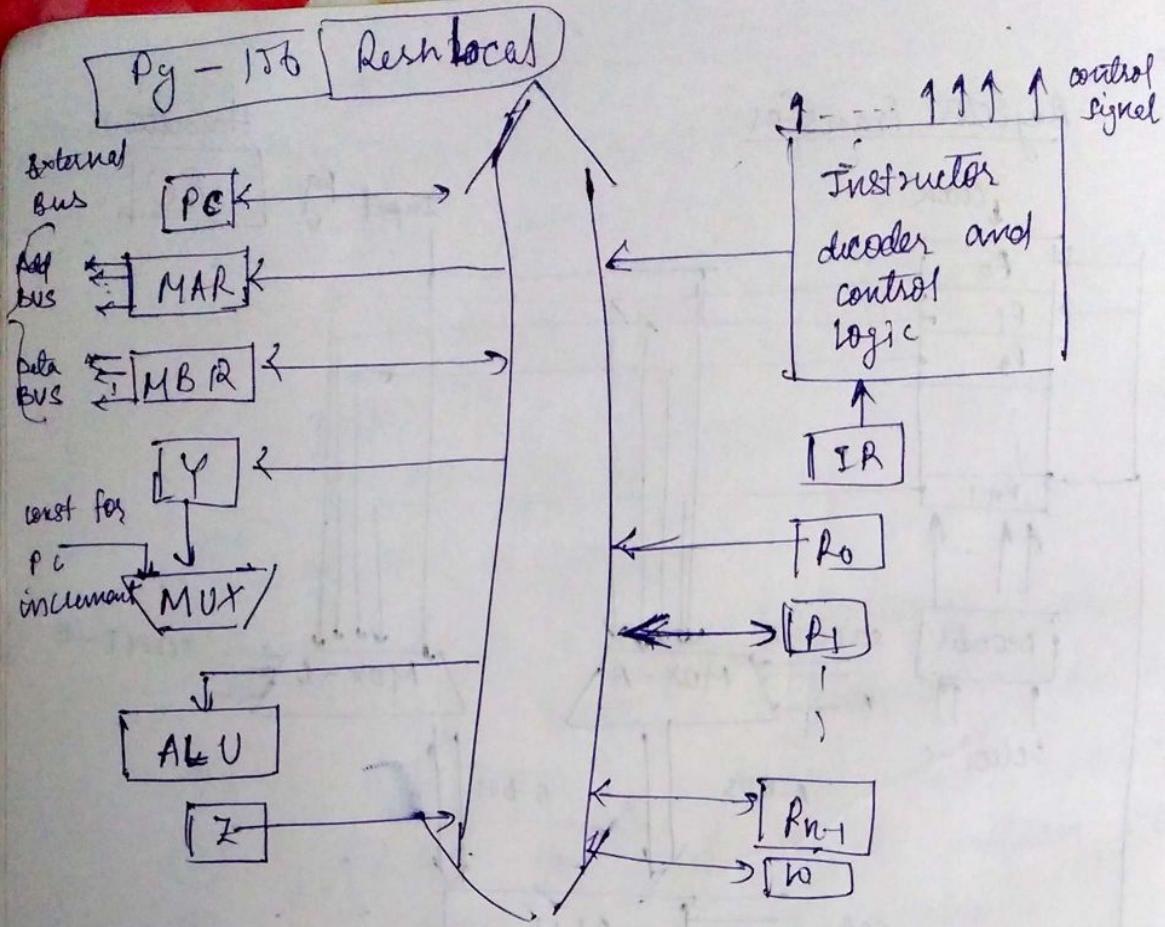
Decoder will select - A_1

④ Accumulator reg always takes part in operation

Internal BUS (in the processor)

1) single bus

2) Multi bus.



[Single Bus with the processor]

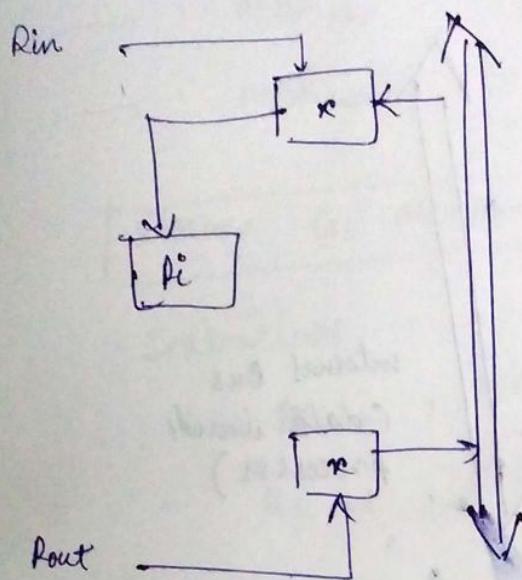
if 4 byte instruction const. for pc increment is 4
added with 4 and go to MUX

Result will be present in **MBR**
MAR gives the location, at that loc result will

store

& all the reg. registers, ALU and interconnected
are collectively known as data path
within the processor.

Register operation.



tristated buffer

- D_f

enable

if $e = 1$

then

if $i/p = 1, o/p = 1$

$i/p = 0, o/p = 0$

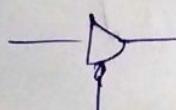
if
enable = 0
no o/p
high impedance state

r — tristated buffer

R_{in} — Read the data from internal bus

$R_{in} = \text{when } R_{in} = 1$

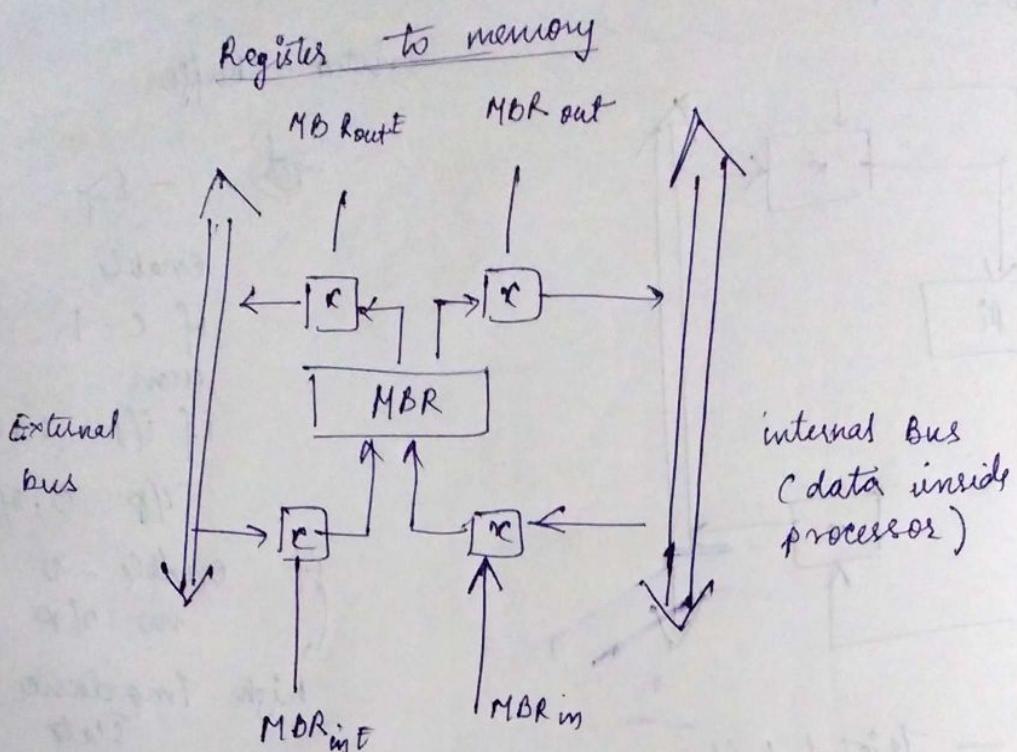
$R_{out} = \text{when } R_{out} = 1$.



enable = 0 (output)
= 1 (no O/P).

- $R_3 \leftarrow R_1 + R_2$
- R_1 out, Y_{in}
- R_2 out, SELECT Y
- ADD, Z_{in}
- Z_{out} , R_3 in

↑
Processor
Memory
Periphery



Memory to ALU → MBR_{in E} ← MBR - MB_{out}

ALU to memory -

MBR_{in} — MBR — MBR_{out E} — Memory

Processor needs only 1 clock pulse

MFC (Memory Function complete)

processor wait for the signal WMFC

WMFC = 0, Waiting

WMFC = 1, MFC completed

Reading a word from memory

Instruction - MOV A, [R₂]

add given by R₂ - [R₂]
from that add data
transfers to P₁

- R_2 out, MAR in, Read
- MBR_{intE}, WMFC
- MBR_{out}, R₁ in

Storing in Memory

Instruction - MOV [R₁] | R₂

- R₂ out, MAR in
- R₂ out, MBR in, write
- MBR_{outE}, WMFC

before reading
writing memory
row should be 1st
selected

Instruction - ADD A₁, [R₂]
 (A) (B)

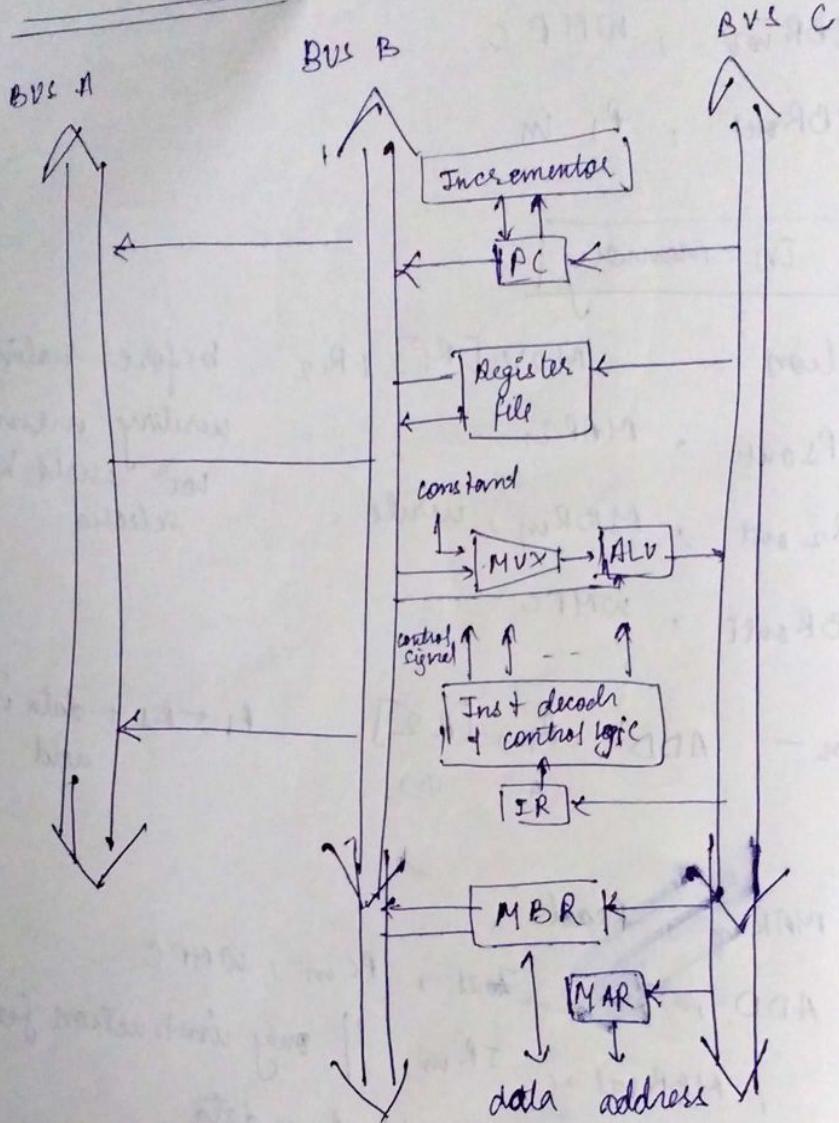
A₁ + R₁ + data in
add. of R₂

- PC_{out}, MAR in, Read
- Select, ADD, Z_m, Z_{out}, PC in, WMFC
- MBR_{intE}, MBR_{out}, SR in || only instruction fetched
- R₂ out, MAR in, Read || R₂ - data
- R₂ out, Y_{in}, WMFC
- MBR_{intE}, MBR_{out}, Select Y, ADD Z_m
- Z_{out}, R₁ in

Disadvantages of single BUS :-

- no of sequence enhanced
- no of clock pulse to complete operation enhanced
- therefore time enhanced

MULTIBUS



constant - required for multi byte instruction

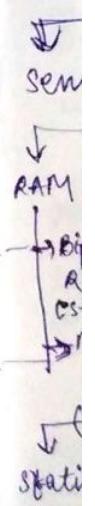
Instruction

ADD R₁, R₂, R₃

1. PC read, MAR on, Read Inst PC
2. W/MFC 11 wastage of 1 clock pulse
3. MBR write, MBR read, IR in
4. R₁ read, R₂ read, Selected A, ADD, Rm

(using)
BJT

(using)
MOS

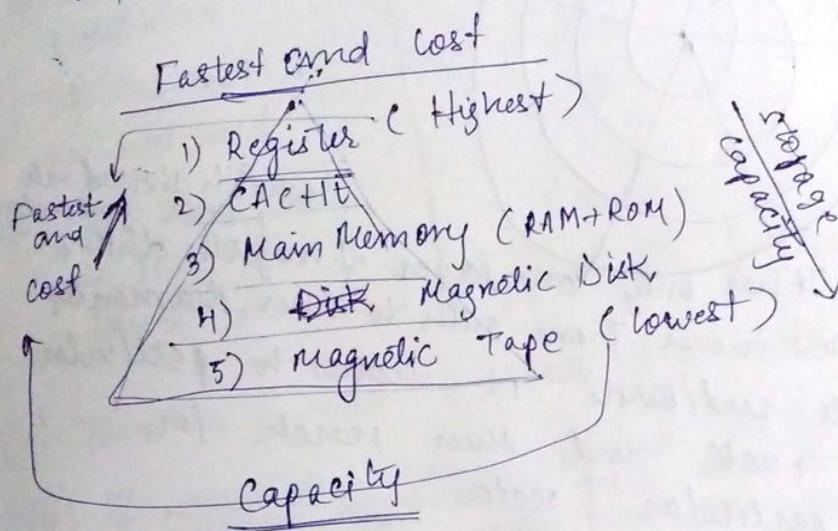


16/03/18

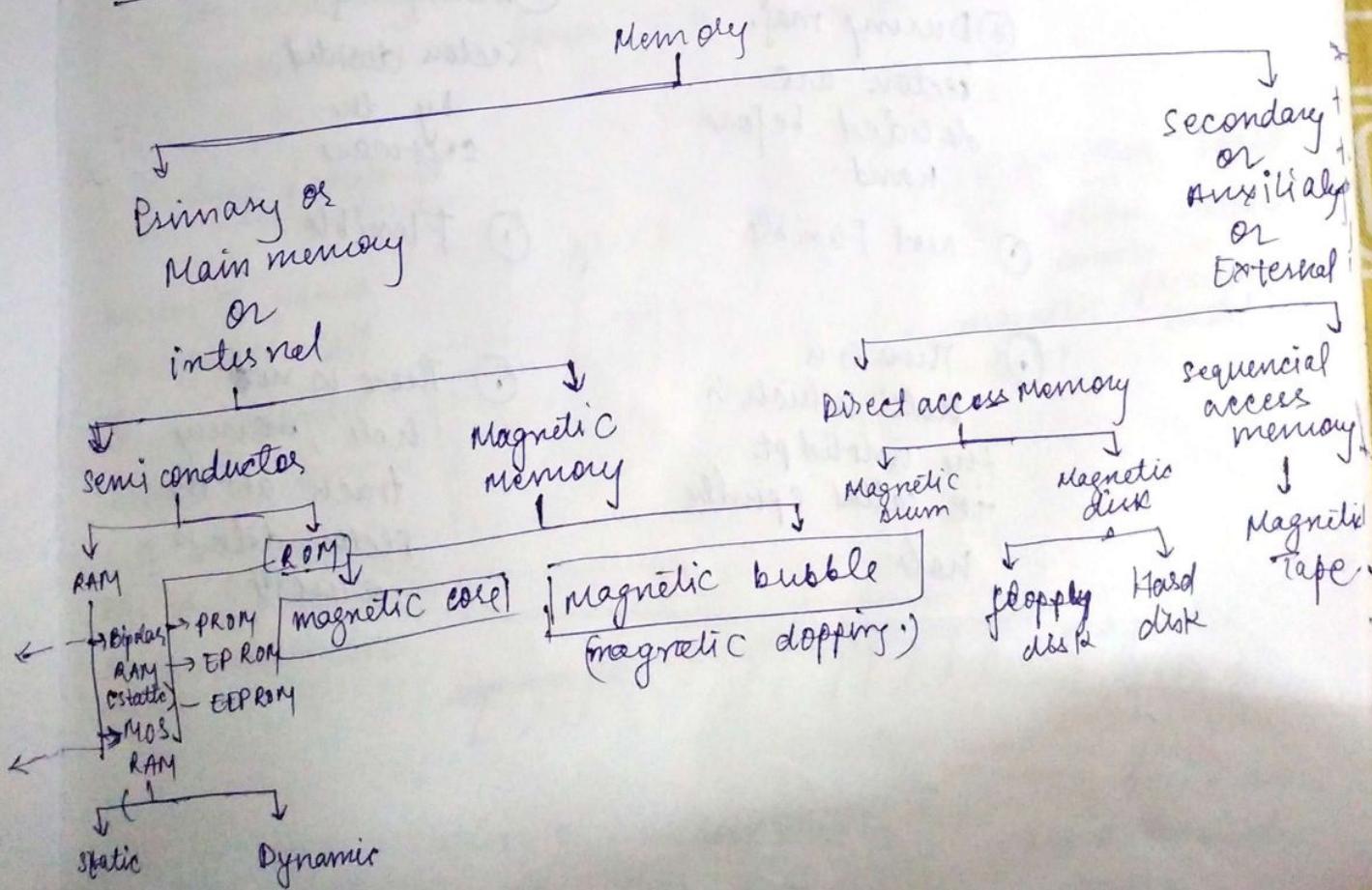
Memory

Module 3

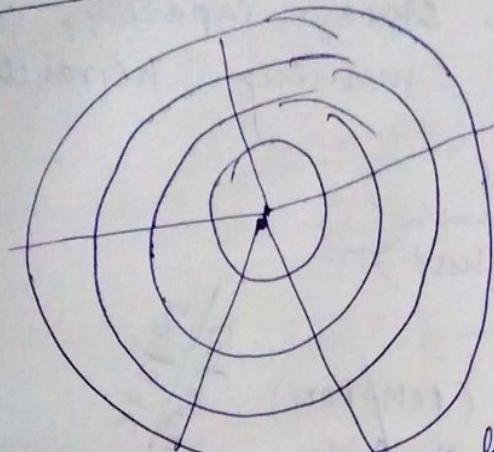
Depending on cost, storage capacity, fast the memory is, there is a memory hierarchy.



Classification of memory



FLOPPY DISK



- ④ It has only one layer of magnetic disk n which is divided into tracks and sectors
- ⑤ Head moves from outer to inner diameter to read/write it moves to particular track and then search for particular sector.

Hard sectoing

- ⑥ During maf. sectors are decided before hand
- ⑦ Not Flexible.

- ⑧ There is a hole which is the initial pt. i.e. called spindle hole

soft sectoing

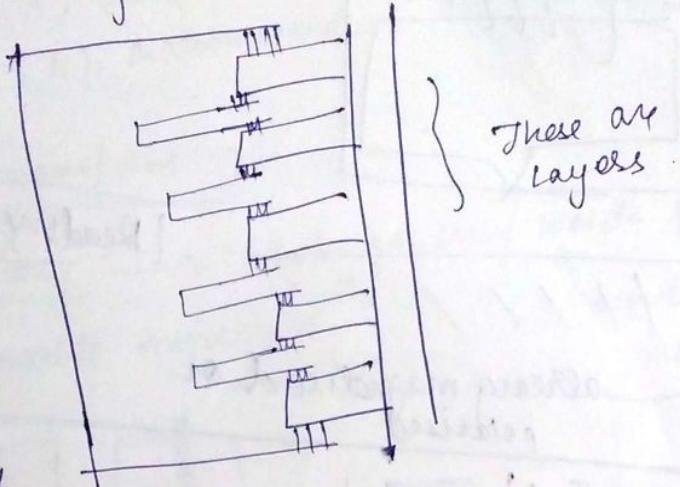
- ⑨ During maf. sectors decided by the software

- ⑩ Flexible.

- ⑪ There is no hole, memory track abtly sector, data is present

Hard disk

- Many magnetic layers.
- If vertically cut it looks like



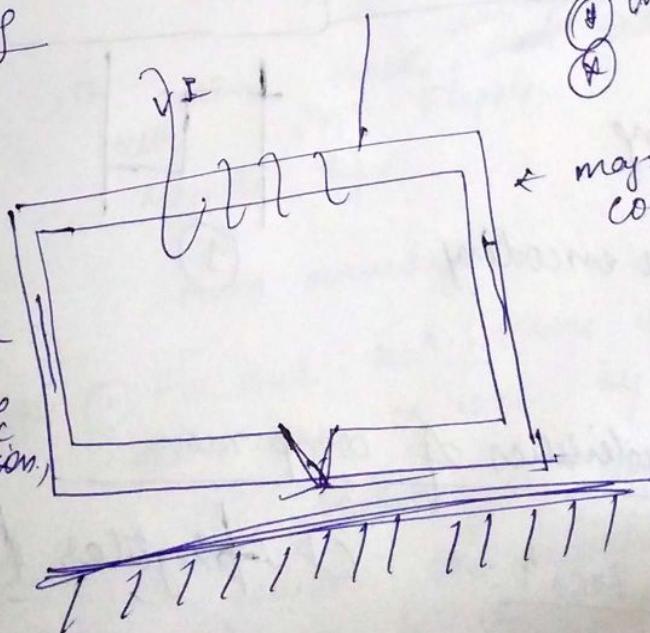
- (*) In some Hard disk Top and last are not used
- (*) In some only 1 side of a layer is used

(*) There is a head to read the data

- (*) If ~~searchable~~ for particular track and if it is there, it will first have many heads for each track
- (*) If ~~not movable~~, there is a head point in each layer there is a track, if we collect all the tracks then it forms a cylinder.

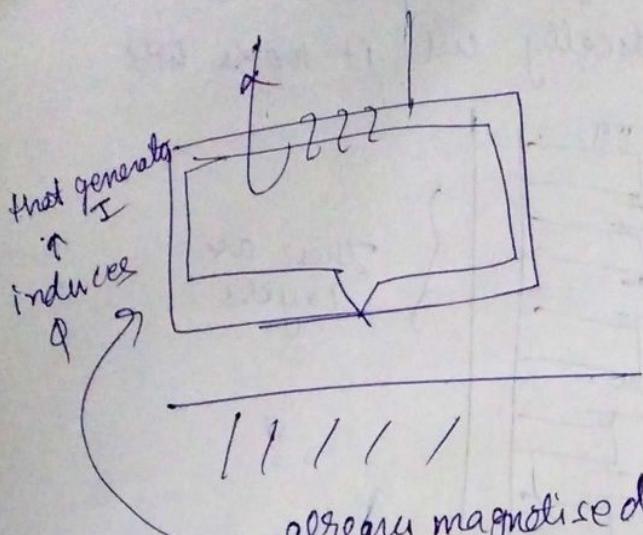
Reading & writing

when I pass through the coil B is generated which induces a Φ and depending on the Φ , the magnetic field magnetise the disk in

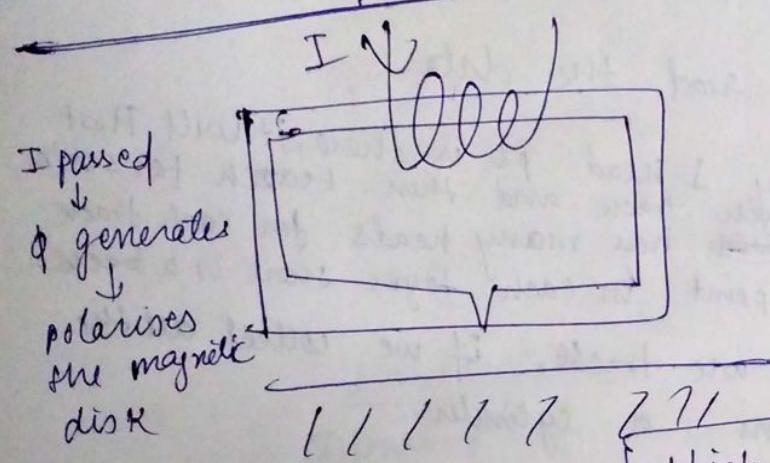


if anticlockwise = 1 then clockwise = 0
else clockwise = 1 then anti = 1

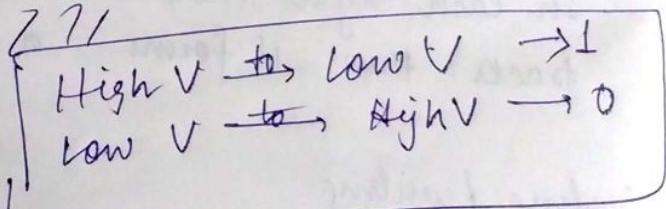
magnetic surface or disk where we want to read, write



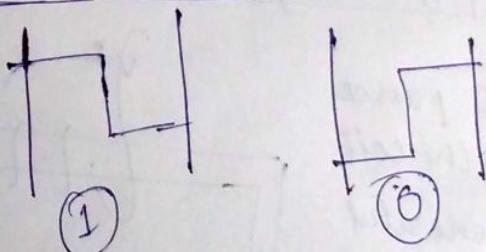
| Reading |



| writing |



This
is
phase coding
or
manchester encoding.



Characteristics of comp mem :-

①

Loc'n :-

Internal

External

CPU → Register → ~~comp mem~~

→ main memory

auxiliary

(a) Capacity :-

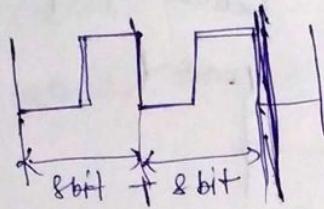
- 1) Bit
- 2) Byte
- 3) word (no. of bits accessed at a time)
- 4) Million words

6bit / 8 bit (16 bit)
6bit / 8 bit (16 bit)

(b) Unit of transform :-

Main memory → bytes per sec / words per sec.
byte addressable memory

words addressable
memory



For 16 bit 2 clocks are required.

Block per sec

Secondary storage :-

(c)

Access Methods :-

direct access

sequential access

random access

then sequential search

direct :-

1st goes to sector track, e.g.: Floppy

magnetic tape

sequential

Main memory

random :-

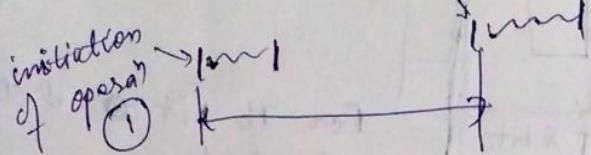
address addressable ① For each locn there is add. ~~selection~~
addressing is done by processor,
the reading or writing is done
at predefined where R/W
is to be done.

Associative access :- ① Happens only in associative memory
content addressable memory

⑤ Performance :-

Access time: is the time of initiation of memory till the completion of read or write operation is called access time.

Cycle time:
initiation of successive memory operation.



Transfer rate: rate at which data is transferred

⑥

Physical characteristics:

- volatile / non volatile (do it by urself)
- destructive / non destructive
- ① seek time (time to reach particular track)
② latency time (time within track to reach particular sector)
↓
Access time

21/3/18

2/11

Seek Time: ① Time required to position the head over proper track

② It is only for movable head system

③

① A

Late

A

R

- ③ For Fixed head System lat is zero.
- ④ Avg seek dist = $\frac{n}{3}$ where n = no. of cylinders.
for hard drive
 n = no. of track for floppy

Latency time :-

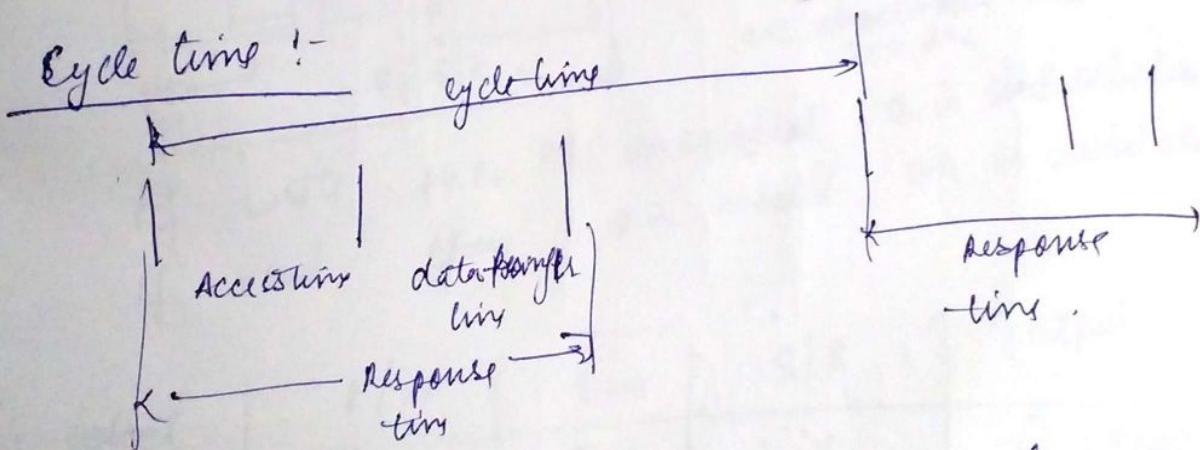
After reaching track, time require to position head
on prop or ~~set~~, sector.

avg latency time = half of rotation time
 $\frac{1}{2}$ of rpm.

Access time = seek time + latency time

Response time = Data transfer time + Access time

Cycle time :-



Q11 A disk system fetch 16 data recording surfaces with 1024 tracks per surface. There are 16 sectors per track and each containing 1024 bytes. Diameter of inner cylinder is 6 inch and

outer cylinder is 10 inch. Find

- ① Capacity of disk
- ② Transfer rate if rotational speed is 3000 rpm
- ③ Latency and avg. latency time
- ④ What is track density.
- ⑤ What is max. bit density = ~~1000~~
calculate linear and areal density.
per inch square (cm²).
- ⑥

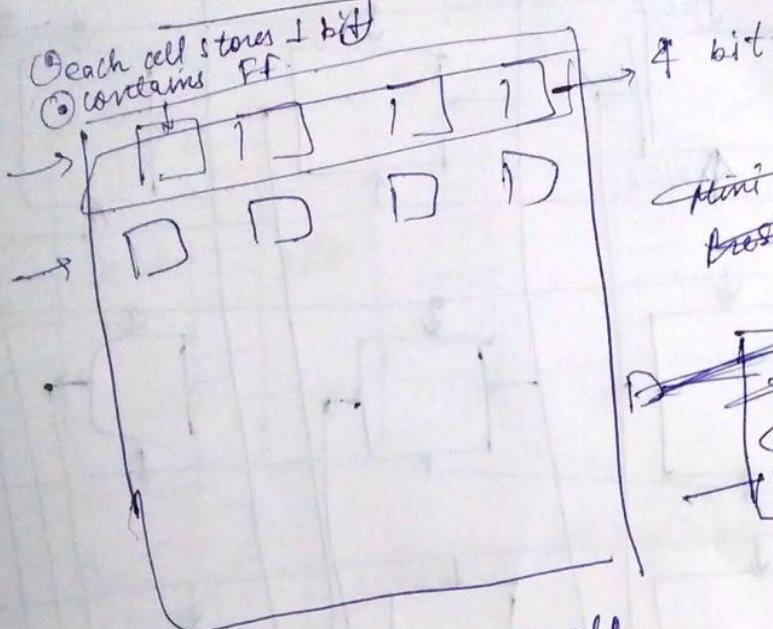
Soln

16 data recording surface

1 surface has 1024 tracks

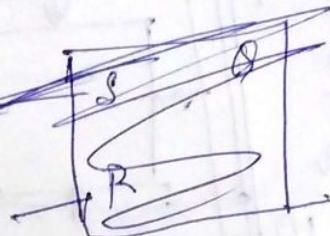
1 track = 16 sectors.

Memory Unit

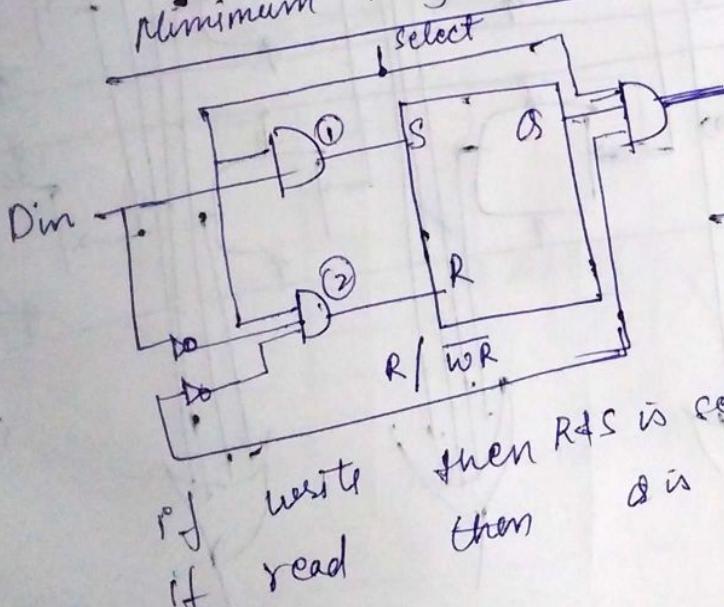


F.F = flip flop

Only min things present in a cell



Minimum thing in a cell



$WK = 0$ then and(1)
and and(2) get input
as 1 (ie activated)
 $Q = 0$ (deactivated.)

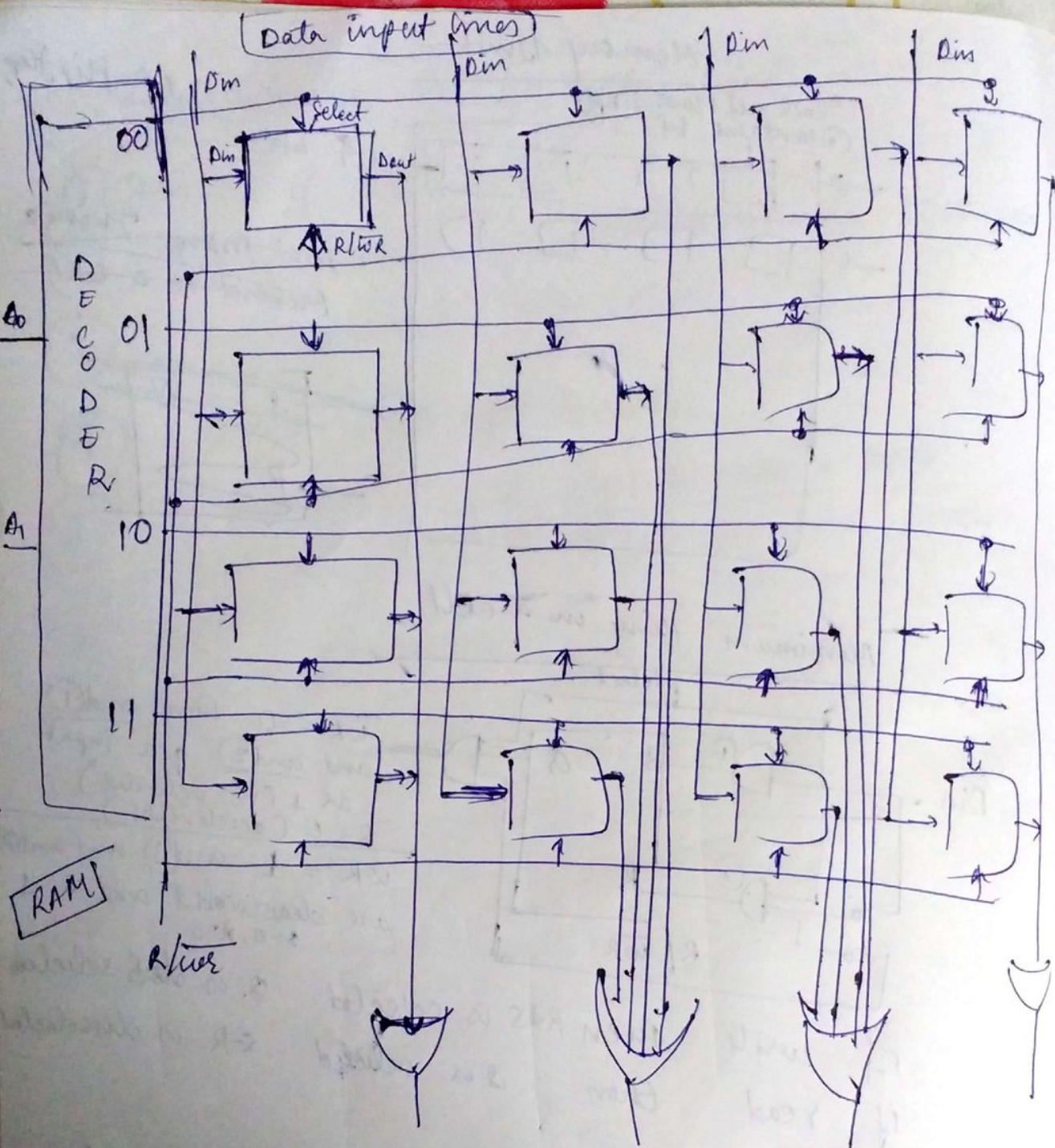
$WR = 1$ and(1) and and(2)
are deactivated and $Q = 1$
 $S = 0, R = 0$

Q is off selected
 $S-R$ is disselected

when R/S is selected
 Q is selected

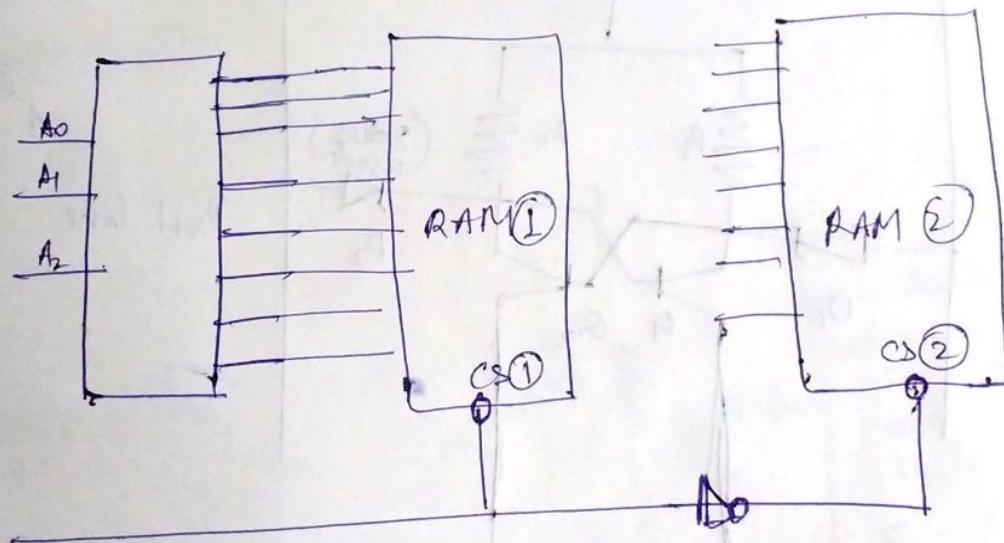
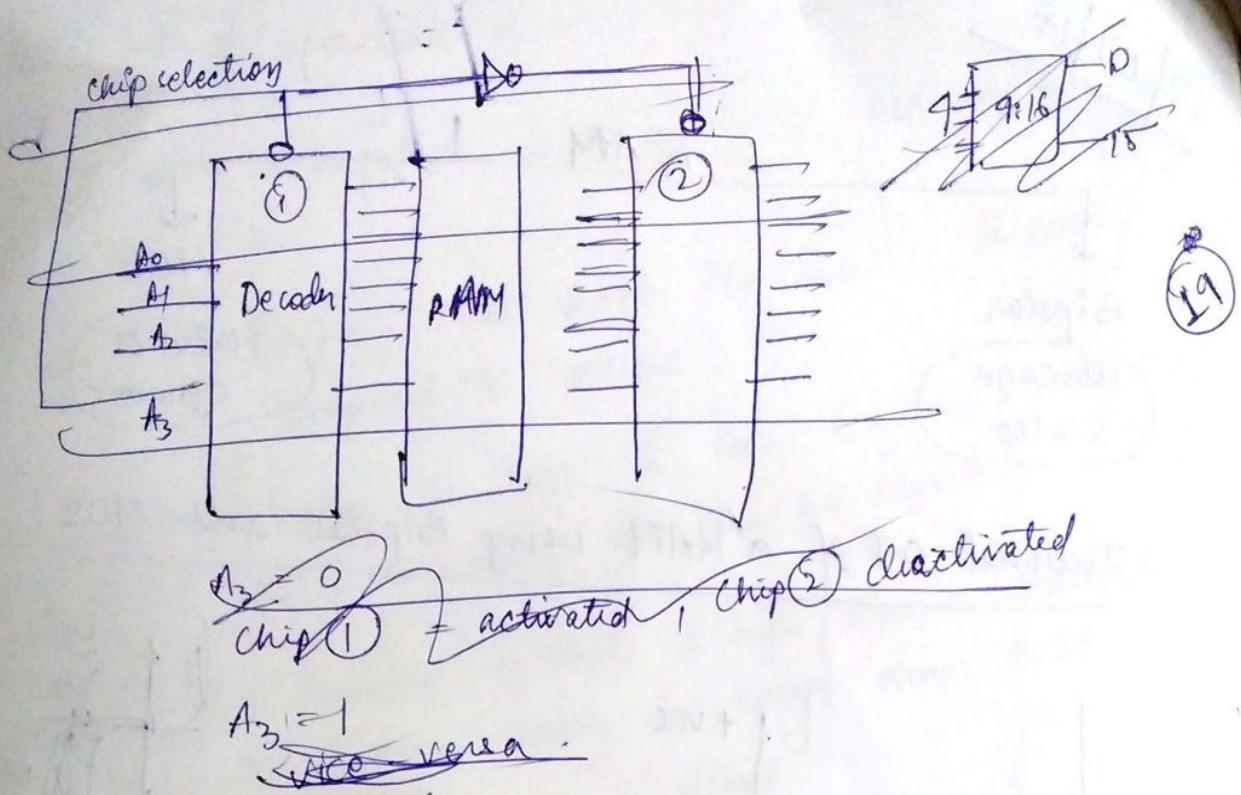
if write
if read

select	R/W	Dim	S/R	Output
1	1	Dim	X	Q (Read)
0	1	X	X	X (Write)
1	0	Dim	Dim	-
0	0	X	X	Y



address bit

A ₀	A ₁	Row① selected	Row②	Row③ selected	Row④ selected	Data in	t _{we} = 0 (write)	t _{rs} = 1 (read) B.P.
0	0							
0	1							
1	0							
1	1							



$A_3 = 0$, $CS(1)$ selected
 $A_3 = 1$, $CS(2)$ selected.

22/03/18

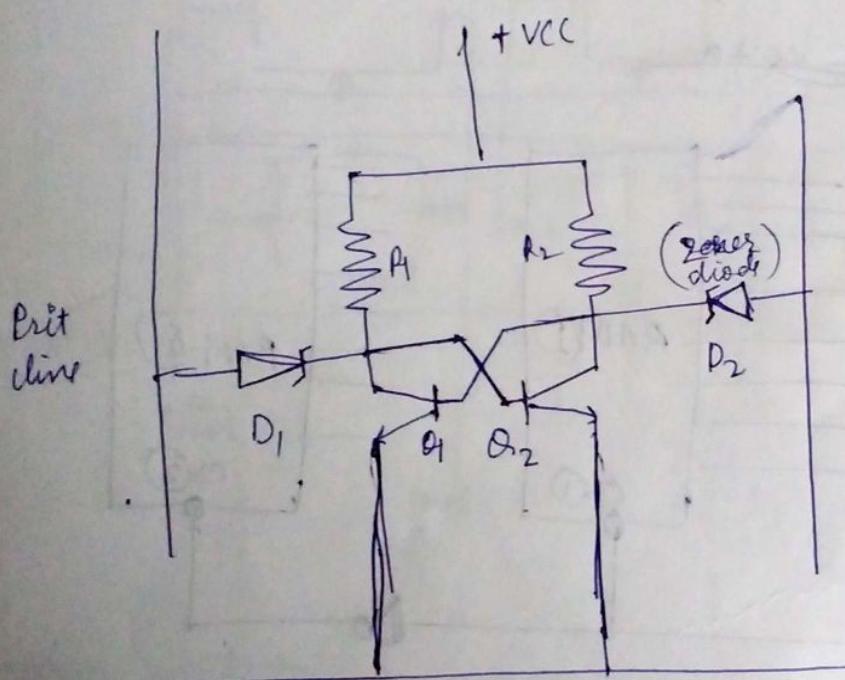
RAM

Bipolar
always
static

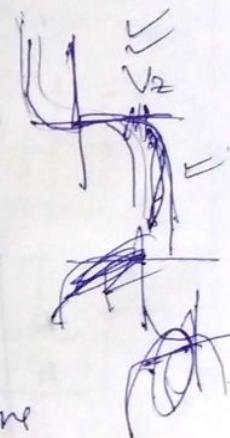
MOS

static or
dynamic

Internal cell of a RAM using Bipolar and MOS



Bit line



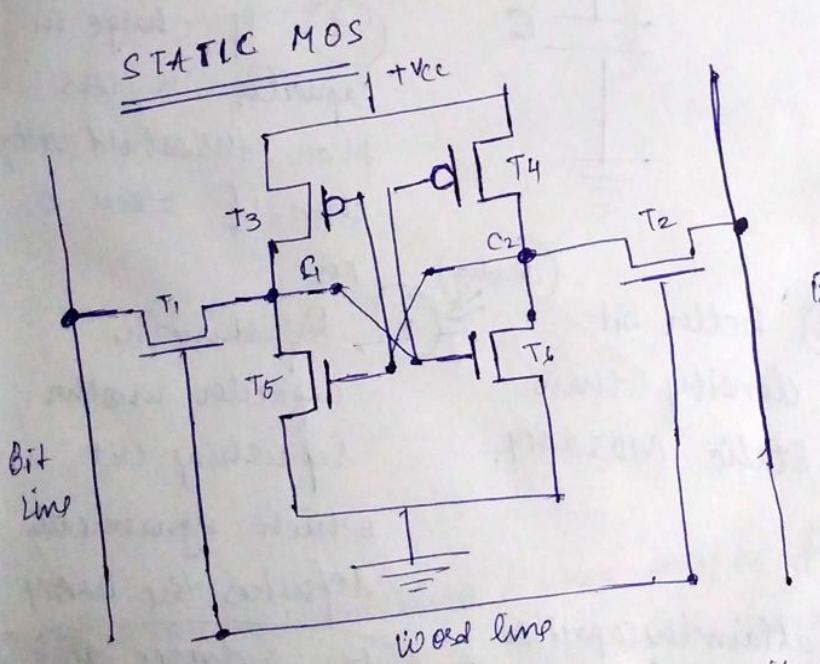
$Q_1 = ON$	$Q_2 = OFF$	\downarrow is stored
$Q_1 = OFF$	$Q_2 = ON$	\circ is stored

Bipolar static RAM

This is a flip-flop using
Bipolar junction
Transistor (BJT)

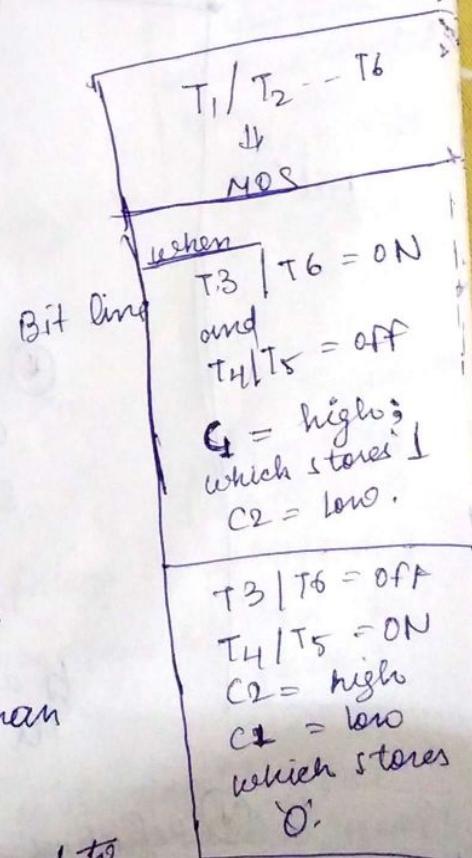
Bit line

- Speed of operation of BJT RAM is higher than other
- Power dissipation is high, \therefore burning probability is high
- To design this RAM for $\pm \text{bit}$, is costly as compared to others.
- The bit density is less (For 1 cm^2 or 1 mm^2 , 1 nm^2 , how many bits can be stored.)
- Bit density of MOS is high / power dissipation is less / MOS designing is easier than BJT. This is why MOS is preferred.



Disadv: It is slower than BJT static RAM.

Though cost is low as compared to BJT still its not affordable.

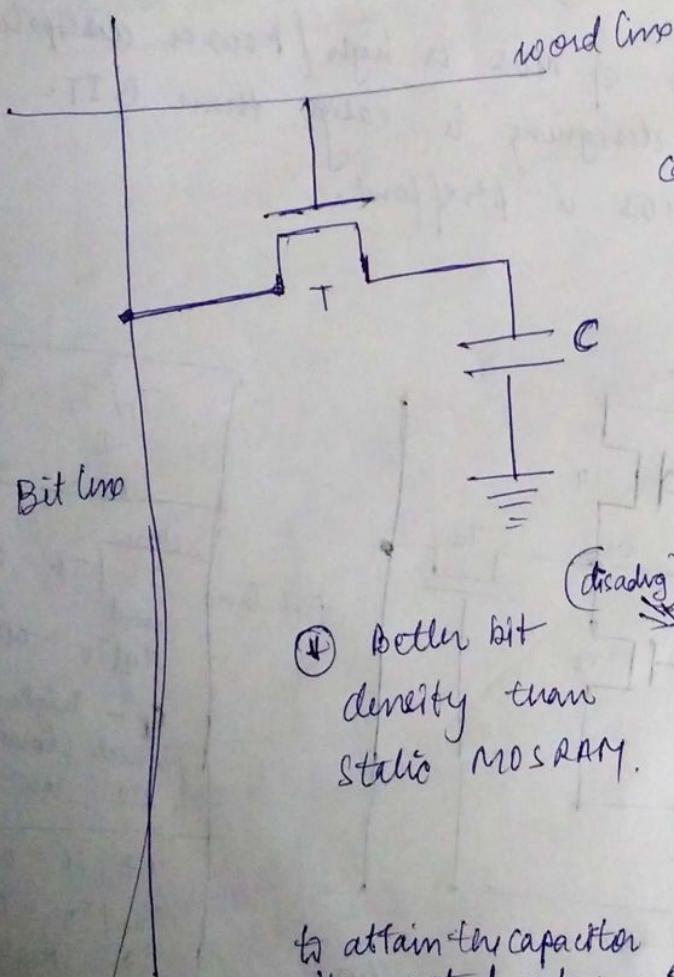


* lower cost RAM is DRAM (Dynamic RAM) using MOS:

DRAM

When T is off
then C starts discharging
from it, stores 0.

First word line is selected
and bit line is given
voltage, if T is ON then
 C starts charging
and stores 1.



* Better bit density than static MOS RAM.

To attain the capacitor its actual value

* Refreshing CLK is inbuilt in the RAM. This type of RAM is called pseudostatic because Refreshing CLK is inbuilt not visible from outside.

(+) if charge in capacitor is more than threshold voltage then it stores 1 or reads 1

(+) if charge in capacitor is less than threshold voltage then it stores 0.

(+) For Refreshing the capacitor another refreshing CLK which dynamically refreshes the DRAM

(+) space of capacitor is more than BJT / mos

During
① \$
② \$
This me and

Types

E

F

Assi

RON

Bit

Types of DRAM

EDRAM (Enhanced Dynamic RAM)

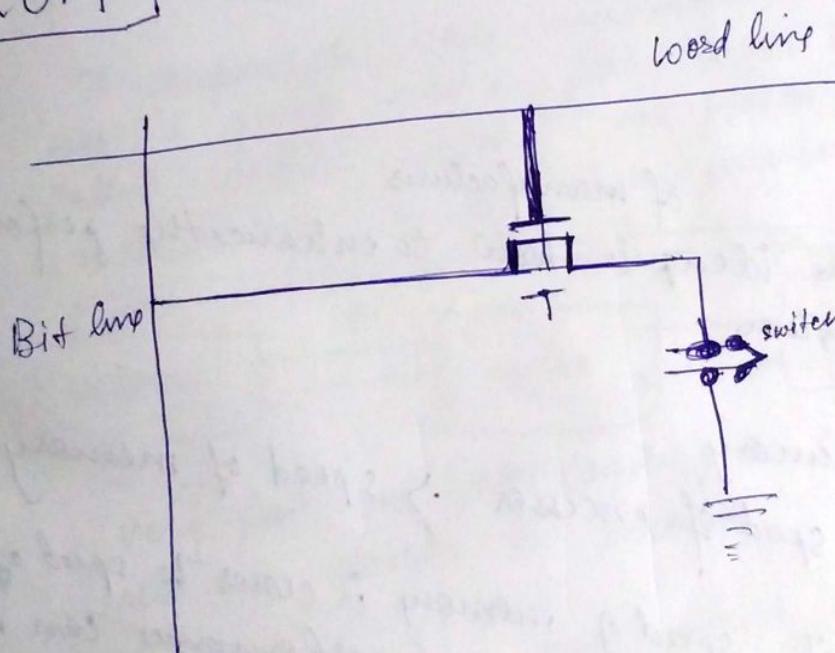
CDRAM (Cache)

SDRAM (Synchronous)

RDRAM (Rambus)

Assignment

ROM



switch
when connected
then no
voltage diff
because
grounded
∴ it is
0.
if disconnected
then there
will be a
voltage diff
then it
stores 1.

During communication, there may be failure

① Hard Failure \Rightarrow Permanently damage (hard disk)

② Soft Failure \Rightarrow

This may be caused due to harsh environmental effect
or sometimes manufacturing defect or due to wear
and tear.

soft failure / soft error :- you may find a data but this may be errors and that is due to power fluctuation (can be avoided using UPS) or due to α -particle i.e. a radioactive particle which has higher penetration power which may change the value from 1 to 0 or 0 to 1.

Assignment

(2) ROM design (2-3 problems)

23/03/18

* Main idea is how to enhance the performance of system.

* memory
speed of processor \rightarrow speed of memory.

(*) if speed of memory is closer to speed of processor then performance can be enhanced adopting

(+) This can be done by some high performance memory techniques:-

1) CACHÉ

2) Virtual memory

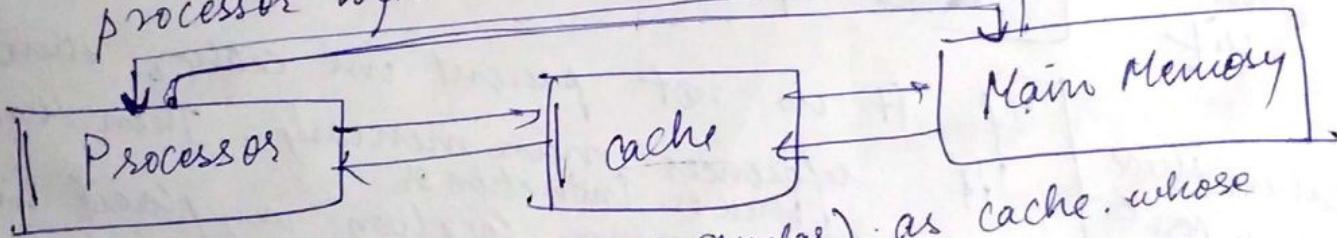
3) Associative memory

4) Memory interleaving

If memory remain active for most of the time then the performance is enhanced.

Cache Technique:-

- generally speed of process is $>$ speed of memory (DRAM)
- we generally use DRAM not SRAM because we need huge size of memory. cost of DRAM is also less and bit density of DRAM is higher etc.
- To compensate this mismatch of speed, we use a small called cache which is placed between processor logic and main memory.

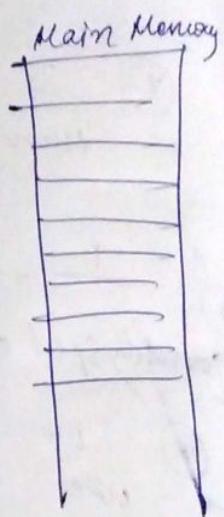


- very fast is SRAM (bipolar) as cache whose speed is nearer to processor.
- There are some prot. which are very frequently used by processor and some data which are frequently needed, and if we are able to make it access it in very less time by keeping it in cache memory which will make the processing faster, enhanced this is called locality of reference.

Two type of Locality of reference

1) Temporal:

- (i) If a data is to be read/written from main memory, simultaneously it is also placed in cache.
- (ii) If such referencing of an address which is accessed in some time which can be referenced in future, this is for future use, ~~use~~ is placed in cache - This is temporal locality



Read Technique

- (i) If any data is required, first it checks for in cache, then read operation takes place

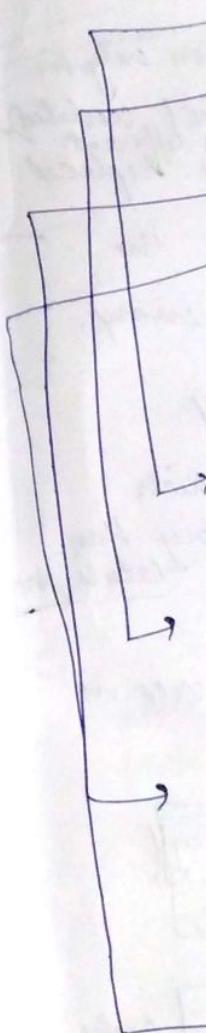
This is
Hit

- when there is in MIS
i.e. not in cache.
 - If it is not present in cache, then it references block or instruction or particular memory location is placed in Cache.

$$\text{performance} = \frac{\text{Hit}}{\text{Hit} + \text{Miss}}$$

important.

Total no of references



④ The cache may become full,
∴ if Full (if it is imp. to replace a block
for the fresh new block of
inst to be replaced in cache).
Various Techniques / Algo are :-

- 1) Random (First come first served).
- 2) FCFS
- 3) Least Recently Used.
- 4) Least Frequently Used

→ No logical / random block replaced
→ The block which 1st comes is 1st replaced
That block which is not being referenced
for a longer period of time, that block
is replaced

→ Block that is referenced for less no. of
time, that block is replaced.
write policy (updating cache)

- (i) write through Technique
- (ii) write back Technique

Write Through Cache

Memory

1. Cache write
then
memory
write

Cache

Process

④ During every update of cache, main memory is also updated.

⑤ Disadvantage: every time writing in both takes more time.

Write back

this means
update flag = 1

The processor continues the update only in cache (main memory values are not updated) when that block is going to be replaced

i.e. update flag bit = 1 then the value is updated in main memory.

Hit conditions

⑥ Errorious data may be produced

⑦ Disadvantage: In case of multiprocessor which takes value from cache and main memory then errorious data is given.

Write Through

First read then write

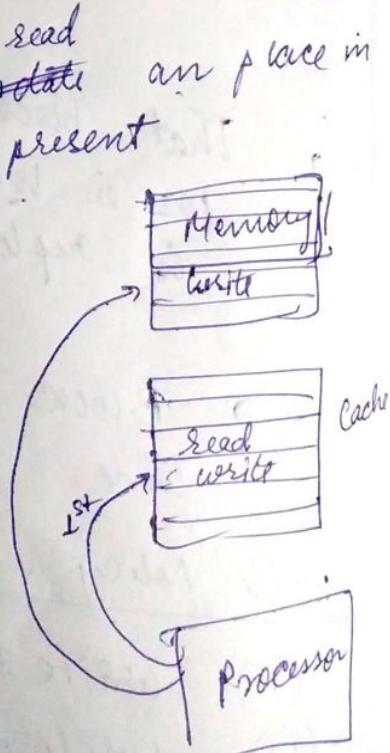
Hit condⁿ

If processor wants to update an place in cache then that must be present in main memory.

Hit case

Now if processor wants to write at some place in cache then it simultaneously writes in the same lock in main memory

Level
Level
Level



Miss Case

If the location is not present in cache then processor directly writes in main memory and simultaneously written in cache.

Miss condition of write back

Suppose processor wants to update a block in main memory but that block is not present in cache memory then firstly that block is transferred to cache and the update by processor starts i.e. update flag = 1 and after updating it remains in cache.

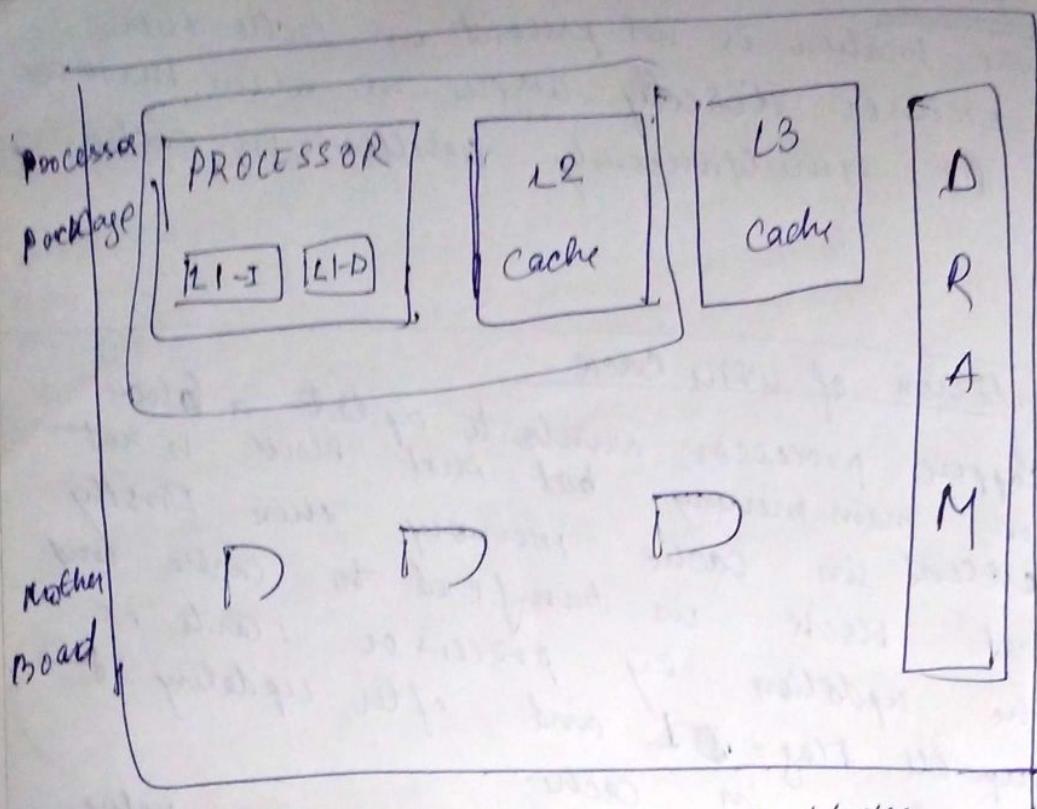
Only during the replacement, then update flag = 1. if then written backs in the main memory

Level 1 cache
Level 2 cache
Level 3 cache

Types of Cache :

Cache

Processor



L1 is splitted into 2 blocks

- ① one for instruction (left)
- ② other for data (right)

size of L1 Cache = 16 KB

③ L2 Cache = 1 MB

④ L3 Cache = some MB

4/4/18
to

⑤ Data/inst in = Data/instⁿ L1 + Data/instⁿ some

L2 + L3 + Extra

⑥ Data/inst in L3 = L1 + L2 + Extra
speed L1 > speed L2 > L3 > DRAM.

⑦ speed of L1 is same as processor.
⑧ speed of L2 is near to L1.

(i)
m
to

④ using this same how parallelism is achieved

⑤ It avoids data hazard.

⑥ If two processor wants to access same instruction and data place in a memory location then 1 processor will have to wait.

∴ This slows down the processor.

if instruction and data is present in diff memory location then no processor will have to wait.

Therefore this data hazard.

Assignment

4/4/18 ① What is cache coherency?

② Mapping : Transformation of data from main memory to cache. There are different techniques

(i) Associative mapping

(ii) Direct

(iii) Set associative

(i) Association : memory used here is associating memory size of main memory. For $1M = 2^{20}$ bits required to address memory. $2^{10} \times 2^{10} = 2^{20}$ bits required.

i.e. 20 bits required.

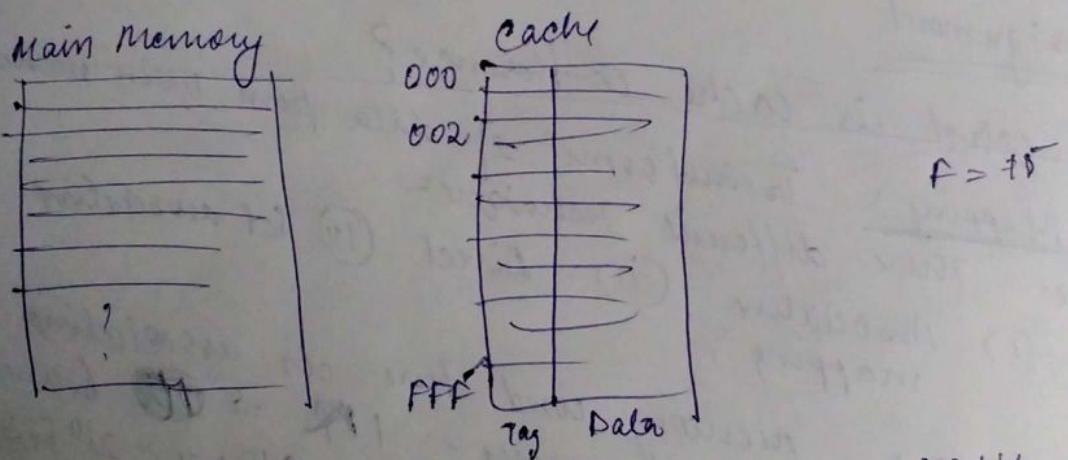
Address	Argument register
Data	

In this technique, when data transferred from memory location to cache then simultaneously add field my data field.

When CPU wants to read, then 1st it searches in cache. During addressing of the data, that address is placed in argument register. If it matches with the ~~associative~~ memory then data is retrieved.

If no memory matches in cache, then the memory location is searched in main memory then that memory location will be placed in cache memory.

$$\begin{array}{l} \text{Main Memory size} = 1M \\ \text{Cache size} = 4K \end{array} = \begin{array}{l} 20 \text{ bits required to} \\ \text{address main memory} \\ 12 \text{ bit required to} \\ \text{address cache} \\ = 2^2 \times 2^{10} = 2^{12} \end{array}$$



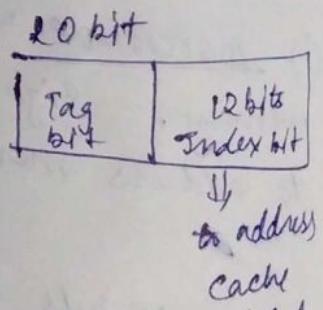
$$f = 15$$

20 bit required to address Main memory further divided into 2 bits tag bit + Index bit

CPU wants to read, then in that address, ~~has 100~~

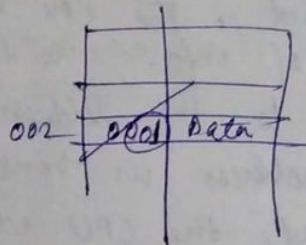
Ex 01002

value of has a decimal value address
tag in cache

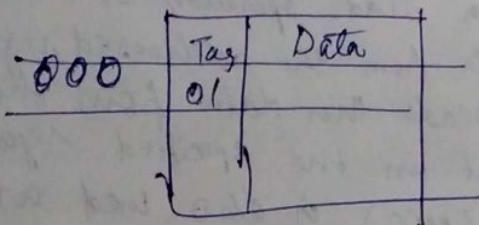


If it's a match, hit then that value is read
 If it's a miss, then value is first read from main mem
 and then it is the the value of tag is replaced

Q1. Q02



Ex:



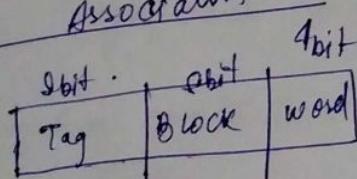
Q1 & Q00
 Hit condition

Q0, Q00
 miss because of
 diff. tag.
 ! replace 1st
 tag = 00 and data

Disadvantage :-

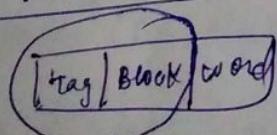
if data field is same but
 diff. tag then each time
 replacement causes page i.e. more in no. Q1 000
 miss and replace.
 which reduces efficiency.

set Association :-



when block transfer is required, then 00
 First block is matched then the tag is searched
 then word is matched.

In case association :-



will be together

searched in
 cache for the
 block

that address
 will contain the block coop.

memory to the CPU. The CPU then invokes the operand fetch cycle. address

The opcode specifies the ~~system~~ of memory info is stored. The CPU transfers the address of the required word of info. to the address lines of the memory bus. is connected to the address bus.

So, the address is transferred to the memory. In this regard, the CPU activates the read signal of the memory to indicate that a read operation is needed. Then result, memory copies data from the addressed register in data bus. The CPU then reads this data from the data register and loads it in the specified register.

Memory function completed (MFC) is also used as a control signal for this memory transfer. Memory sets this signal to + to indicate that the contents of the specified location have been read and are available on data bus.