

2-way set.

$$\text{main memory} = 16 \text{MB} = 2^{24}$$

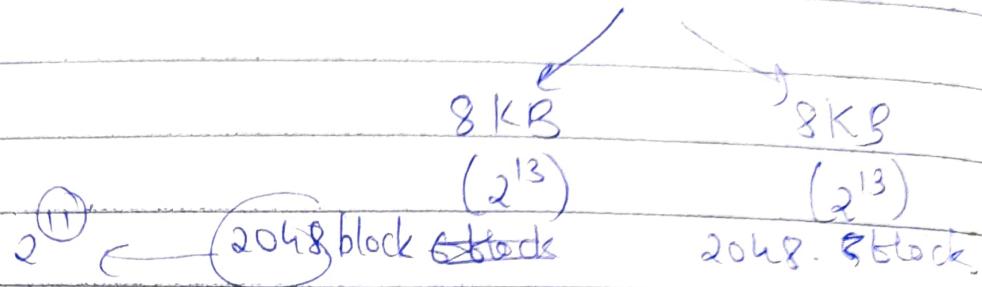
$$\text{cache memory} = 16 \text{KB} = 2^{14}$$

$$\text{block size} = 4 \text{bytes} = 2^2$$

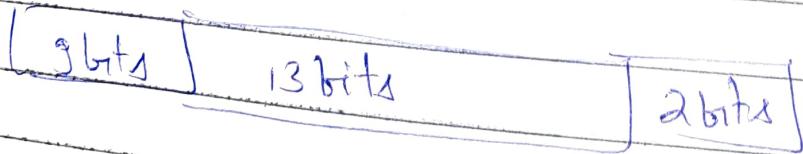
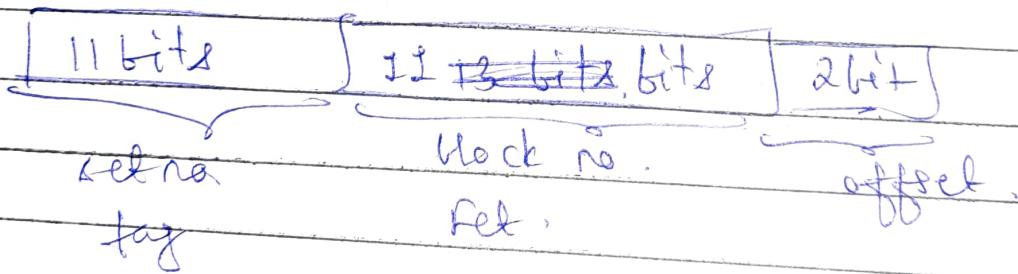
$$\text{no. of cache blocks} = 2^{14}/2^2 = 2^{12}$$

$$\text{no. of sets} = 2$$

c.m (16KB) 4096 blocks → offset



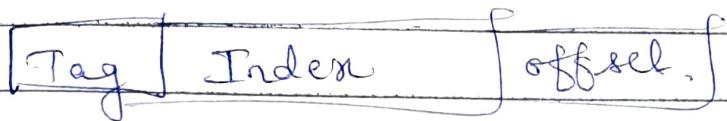
$$\text{no. of sets in main memory} = \frac{\text{total size}}{\text{size of set}} = \frac{2^{24}}{2^{13}} = 2^1$$



32 → 8 way!!

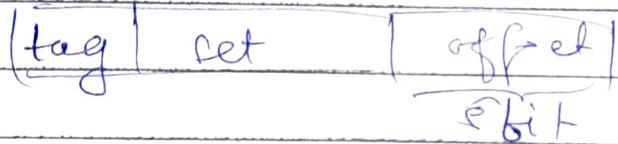


Direct:

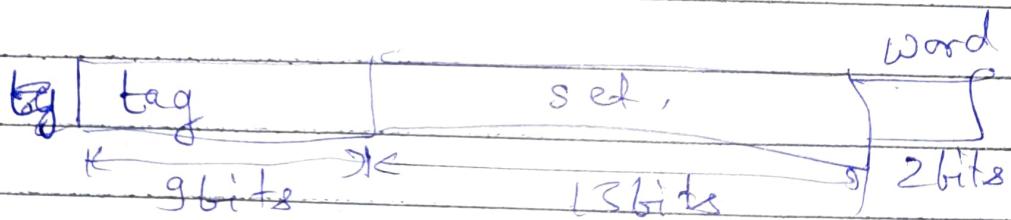


$$\text{Performance} = (\text{hit rate} + \text{miss rate}) \times \text{miss penalty}$$

- each locat<sup>n</sup> Multiple blocks are resided
- DM → miss rate is more. & many blocks can reside on
- fm → no. of searches are more. Cache  
line<sup>y</sup>
- SA → miss rate is reduced.
- no. of searches = no. of sets.



no. of cores ↑ then no. of set associativity ↑.



block no. determines which set

(Block no.) modulo (Sets in cache)

→ Search all

Page No.	
Date	

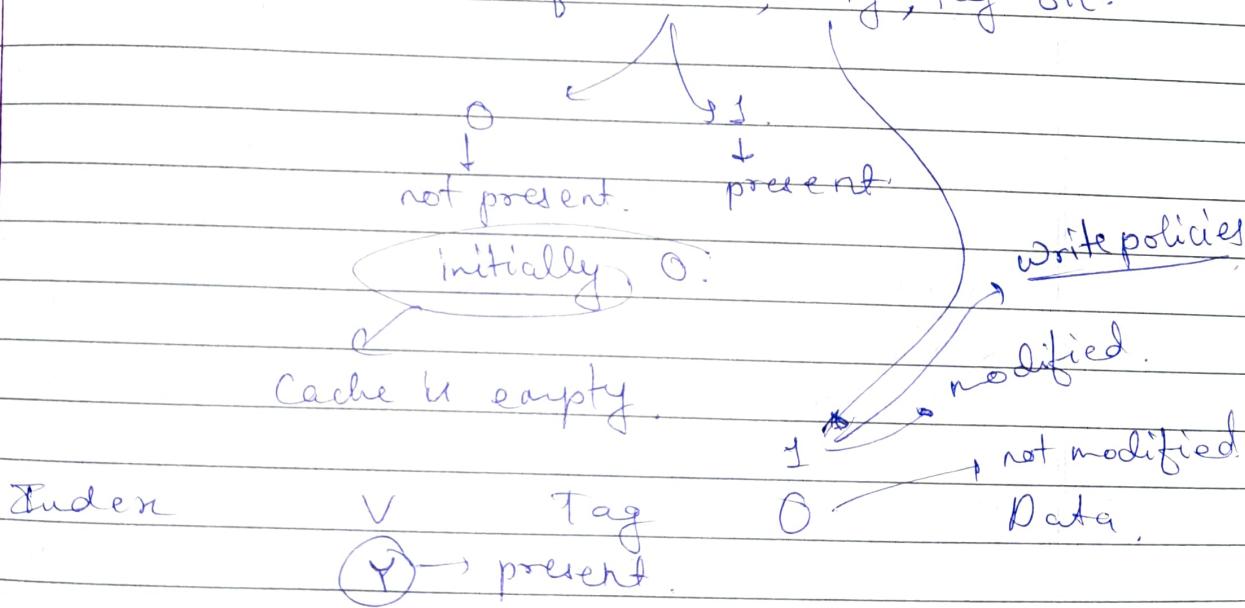
miss

- capacity or cold or compulsory miss } 3C/8
- capacity miss. } p=
- conflict miss.
- coherency miss.

→ block size increased  
Eviction &  
replacement increased  
so misses inc.)

cache to empty

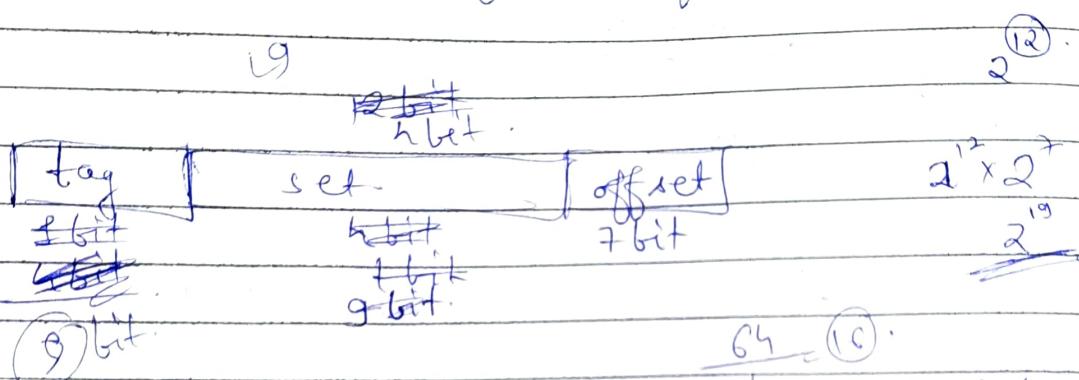
Each block consist of Valid, Dirty, Tag bit.



$V = 1 \& \text{tag matched}$  if & only if then hit happen.

$$\cancel{h^2} \cancel{(2)} x^2 = \cancel{h^2} \cancel{(2)} x^2$$

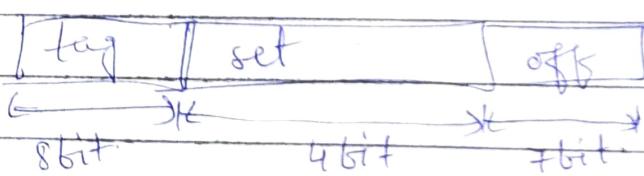
Ques. → The cache divided into 16 sets of 4 lines each. → 4 way set ass.  
A set assoc. cache consists of 64 lines slots, divided into a lines set  
main memory contains 4K blocks of 128 words each. Show the format of main memory addr.



64 x 128

$$2^2 \times 2^7$$

$2^{13}$  + 8 KB - cache.



$$2^0 \rightarrow 256$$

	L1	0 12P
SJ		
	L2	0 12P

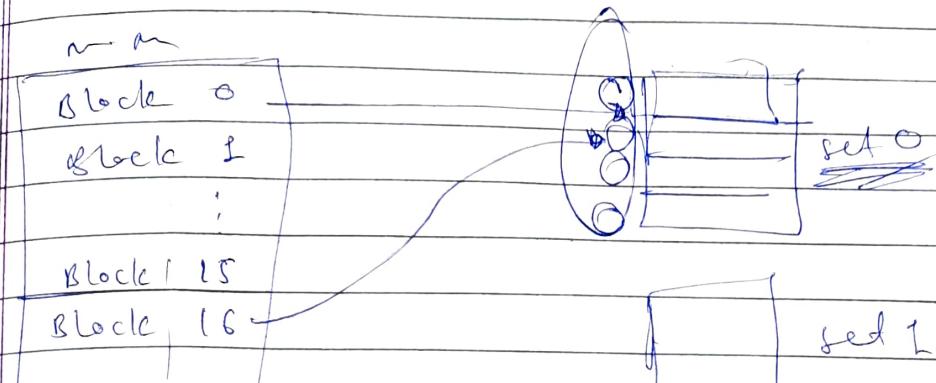
$$\textcircled{1} + 4 + 2$$

2048  
2096  
Page No.  
Date

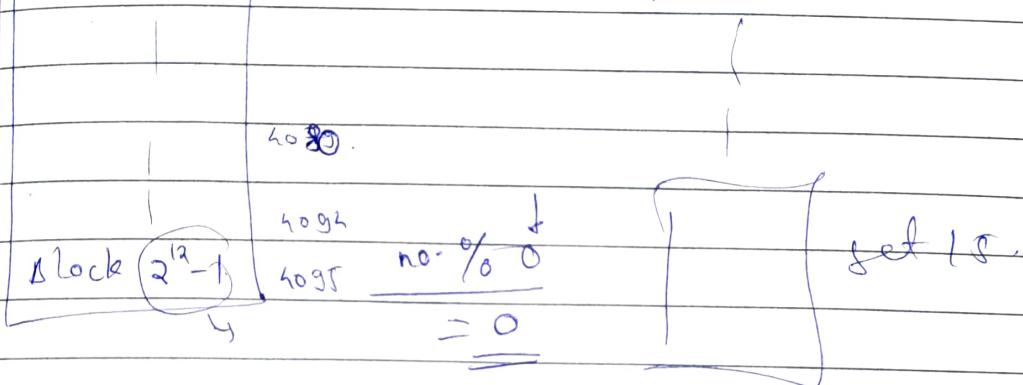
$$2^{12} \times 2^7 \rightarrow 2^{19}$$

(8) hit.

$$\frac{2^{12}}{2^4} = 2^8 \cdot \frac{2^{12}}{2^8} = 2^8 \cdot \frac{\text{total m.m. blocks}}{\text{total sets}}$$

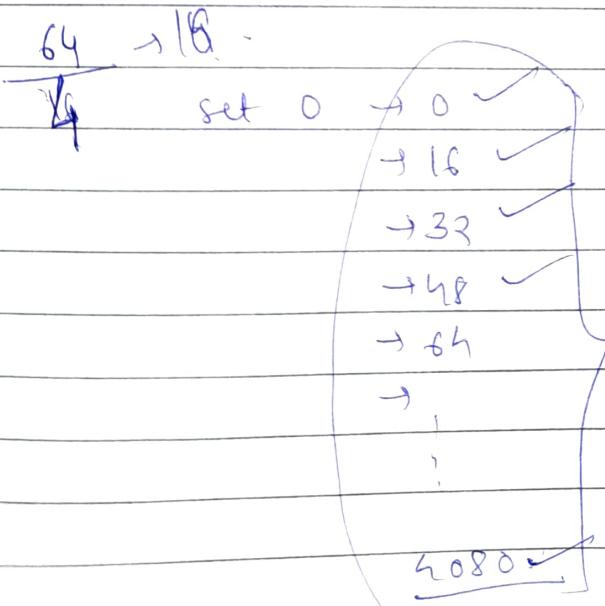


815  
log<sub>2</sub>  
16  
9079



$$\frac{2^5}{2^2} = 3 \text{ bit.}$$

[ 5 bit ] + 6 bit



tag | holes off

tag set off

$$\frac{2080+1}{16} = 125 + 1$$

$$= 256$$

$$= 2^8$$

Page No.	
Data	

Direct

$$m \cdot m = 128 \text{ bytes} \rightarrow 2^7$$

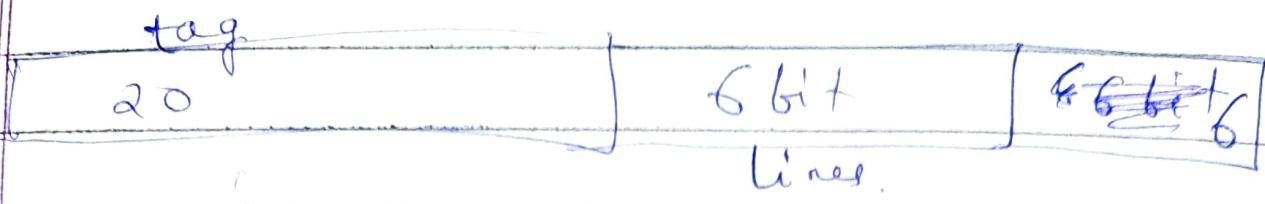
$$c \cdot m = 16 \text{ bytes} = 2^4$$

$$\text{block si} = 32 \text{ bit} = 4 \text{ bytes} = 2^2$$

$$\text{no. of } m \cdot m \text{ blocks} = 2^5$$

$$\text{no. of } c \cdot m \text{ blocks} = 2^5 / 2^2 = 2^2$$

block size  $\rightarrow$  64 byte  $\rightarrow 2^6$

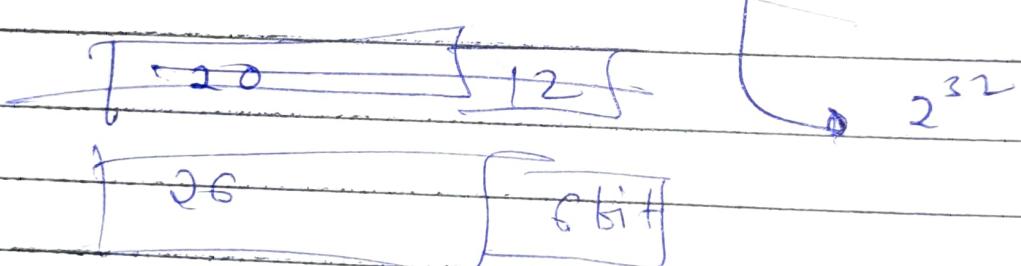


$\rightarrow 2^6 \rightarrow$  cache line  $\rightarrow 64$

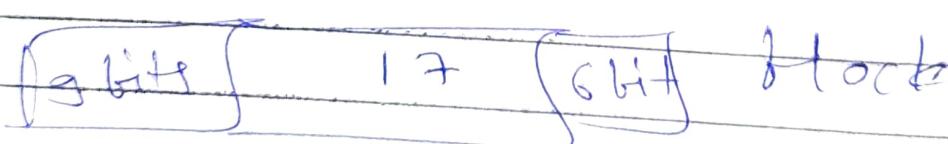
tag - 20 bit.

$\frac{2^{32}}{2^6} \rightarrow 2^{26}$  - no. of block in m.m.

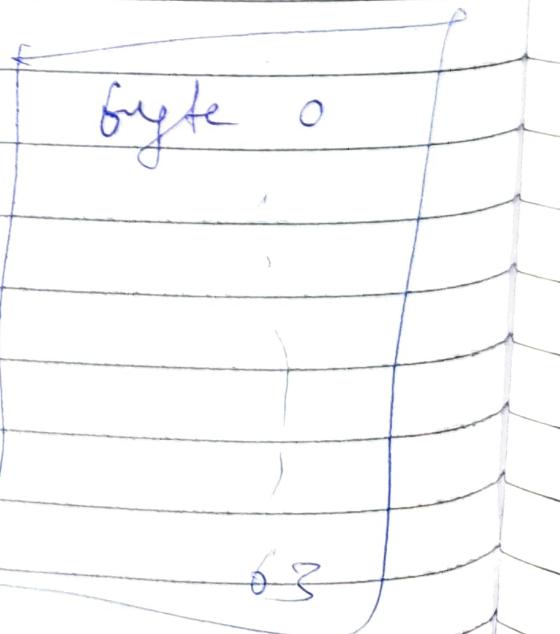
no. of addressable units  $= 2^{32}$ .



4 way.



$2^7 \times 4$  way  
lines in cache  $\rightarrow 2^{19}$



+ , \* stands for hit!!

Page No.	2 <sup>0</sup>
Date	2 <sup>2</sup>

$$1 \text{ MB} \rightarrow 2^{20} \rightarrow \text{m.m}$$

$$\text{block size} \rightarrow 2^4$$

$$\text{cache size} \rightarrow 2^{16}$$

$$\frac{2^{16}}{2^4} = 2^{12}$$

no. of blocks  
in cache

F 0 0 1 0

↓ line. offset.

tag	1111	0000 0000 0001	0000	✓
tag	0000	0001 0010 0011	0100	✓
tag	0100	1010 1011 1011	1110	✓

$$2^{12}$$

$$2^{11} \quad 2^{11}$$

5 bit tag.

tag F 0 0 1 off → 0

tag 0 1 2 3 off → 4

tag C A B R off → E

0000	0000 0000 0000	0000
------	----------------	------

$$= \frac{2^{16}}{2^{11}} =$$

0000	0000 0000 0000	0000
1100	0000 0000 0000	0000

3 bit 11 bit 4 bit  
tag set off

$$0\% (2) = 0$$

$$12\% 12 = 0$$

word offset → 0111

line → 1111 1111 1111

tag → 0000

addr. → DFFFF

word offset → 0001,

line → 1111 1111 1111

tag → 0011

addr. = 3 FFFF1.

random, pseudo random, LRU, LFU, FIFO

Page No.		
Date		

## FOOT CABC

tag	line	offset
5	5	6

128 + 24  
144  
2176 +.  
2180

~~0 = 1 01 0 02~~  
~~hit ratio = 1/3~~

128 ~~Tag off +~~  
144  
2176 + 6128  
2180

n n m m

-128 → (00000) (00010) (000000) miss  
144 → (00000) (00010) (010000) hit  
~~2176~~ → (00001) (00010) (000000) miss  
2180 → (00001) (00010) (000100) hit.  
128 → (00000) → miss  
2176 → (00001) → miss

(P) → (S) → (M)

$\alpha$   
T.L.D.S.

## MSI & MESI

MMU (Memory management Unit)

Writing / modifying needs permission from cache controller (sheff)  
 recent generated " use (1) invalid based cache coherency protocol  
 (2) update based  
 ↳ create traffic  
 issue &  
 energy  
 consumption".  
 ↳ No need to seek  
 permission from sheff again while writing in modified state.

P# : <op> <address> [Value]

P.O.BO : (I, 120, 00 01)

↑ tag      ↳ data words

MSI

a) [0] <S.2>

P0: read 120

P.O.BO : (S, 120, 00 20)

b) [10] <S.2>

P0 : write 120 ← 80

P.O.BO : (M, 120, 00 20)

c) [10] <S.2>

P3 : write 120 ← 80

(b)

P.O.BO

P.O.BO : (M, 120, 00 80)

P3.BO : (I, 120, 00 20)

(c) P3. B0 : (M, 120, 00 80)

P1: read 110

M changed to \$!!

P0: write 108 ← 48

P3. B2 : (S, 110, 00 30)

P0: write 130 ← 78

P3: write 130 ← 78.

(d) P0. B1 : (M, 108, 00 48)

P3. B1 : (I, 108, 00 08)

(e) P0. B2 : (M, 130, 00 78)

P0. B2 : (M, 130, 00 ~~30~~ 78)  
M ← 110

of most recently modified

(g) P3. B2 (M, 130, 00 78)

avg. access time

avg. memory access time (AMAT)

AMAT = Hit time + miss rate × miss penalty.

ex: CPU with 1ns clock, hit time = 1 cycle,  
miss penalty = 20 cycles, L-cache miss rate 5%

$$\rightarrow = 1 + 0.05 \times 20$$

$$AMAT = \underline{\underline{2}}$$

$A_{MAT} = R + H_{L1} + m_i \cdot c_{rate_{L1}} \times (H + H_{L2})$   
 $+ m_i \cdot r_{rate_{L2}} \times m_i \cdot p_{penalty_{L2}}$ .

### \* DRAM Refreshing

3 types =

i) RAS

only  
refresh

ii) Hidden  
refresh

iii) Distributed  
refresh

↓

↓

• refresh the

cells after once if we try

to read particular

location then RAS is reading or

writing

• while refreshing

time period

• All cells in a

row get refreshed

side by side is not disturbed

refreshed

at a time

• AT the same time

after some

time then

Also called as

other rows are

burst refresh

refreshed

it will wait

for same time

so again flag

refresh next

line

• while waiting

period is very

read or write

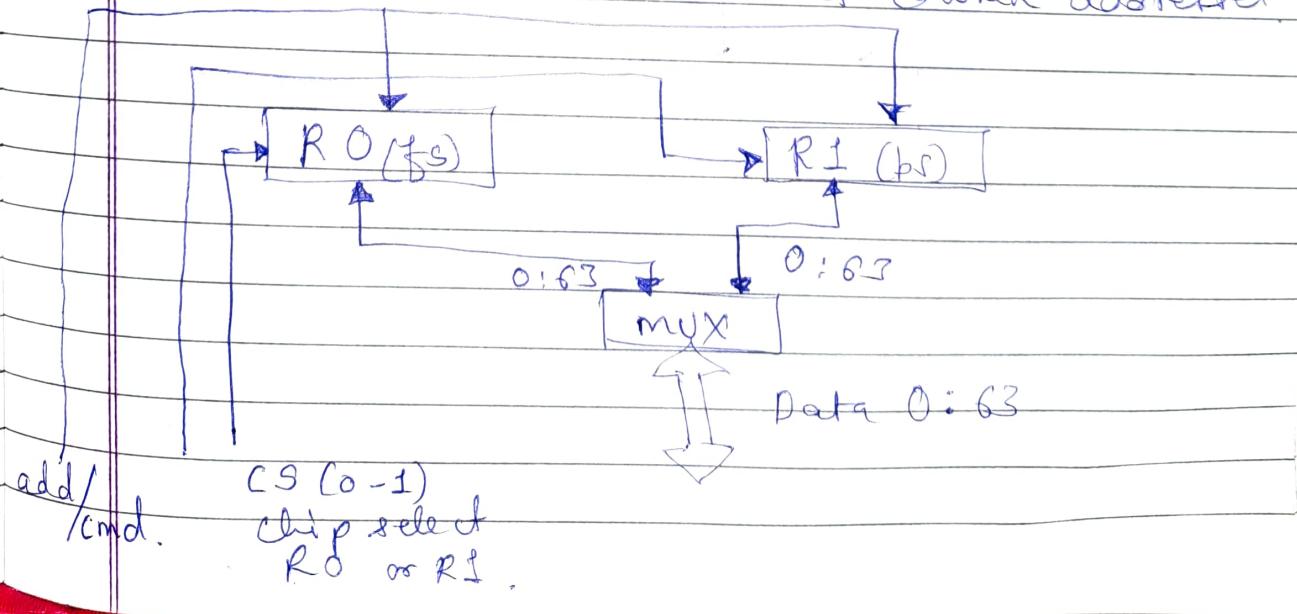
action can take

place.

- A chip present on DDR, one side is called Rank 0 while other side is called Rank 1.
- On each rank there are 8 chips simultaneously working.
- ~~Dimm~~ DIMM, dual inline memory module

- 1) Channel -
- 2) DIMM
- 3) Rank
- 4) Chip -
- 5) Bank
- 6) Row/Column

- 1) Is a socket where DDR gets inserted.
  - 2) actual packet or module
  - 3) Rank 0 & Rank 1 on either of the sides, that are 2 surfaces or sides on which chips are present
  - 4) On each side 8 chips are there (per Rank)
- 
- 5) Each chip consists of 8 banks.
  - 6) In bank there are row & column addressed



8 chips on a rank.

