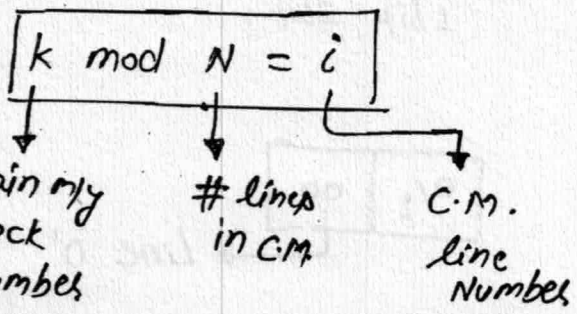
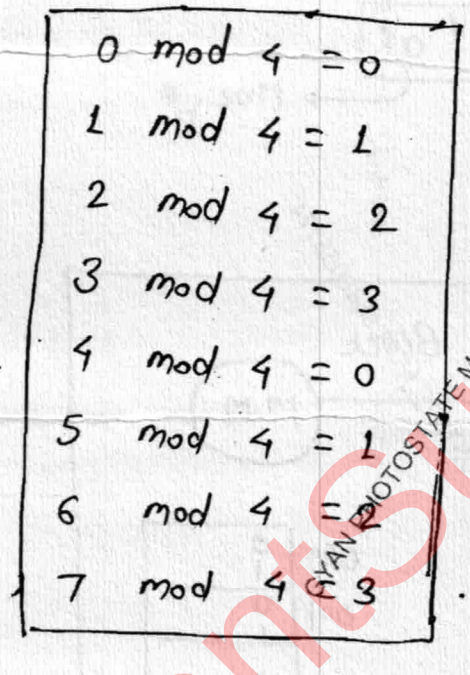
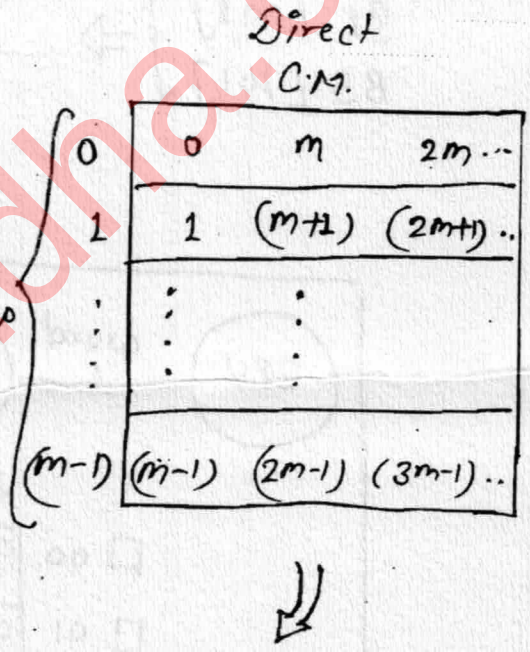
⇒ "Direct mapping" :—

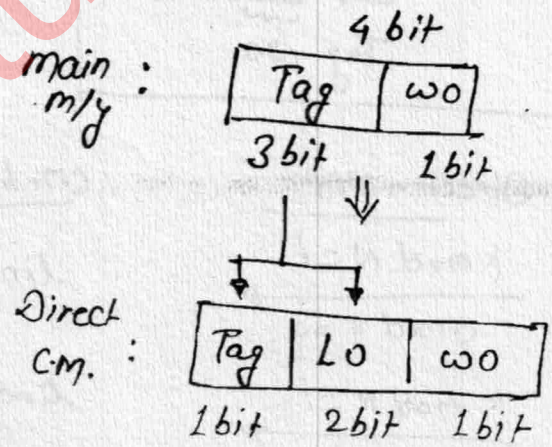
In this technique mod() function is used to map the data that is :







* Above function shows the relationship between main m/y block and cache m/y line therefore Address Binding is :—



main m/y

Tag

3bit

Direct C.M

Tag	LO

1bit 2bit

B0 [000]
B4 [100] ⇒

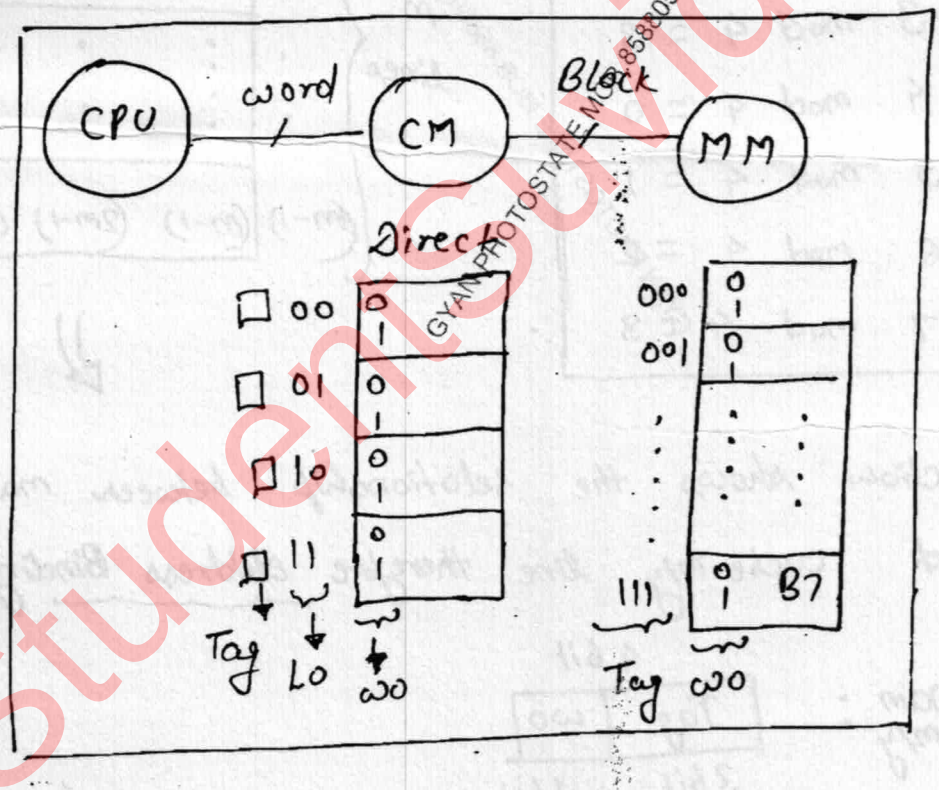
0/1	00

→ Line '0'

B1 [001]
B5 [101] ⇒

0/1	01

→ Line '1'

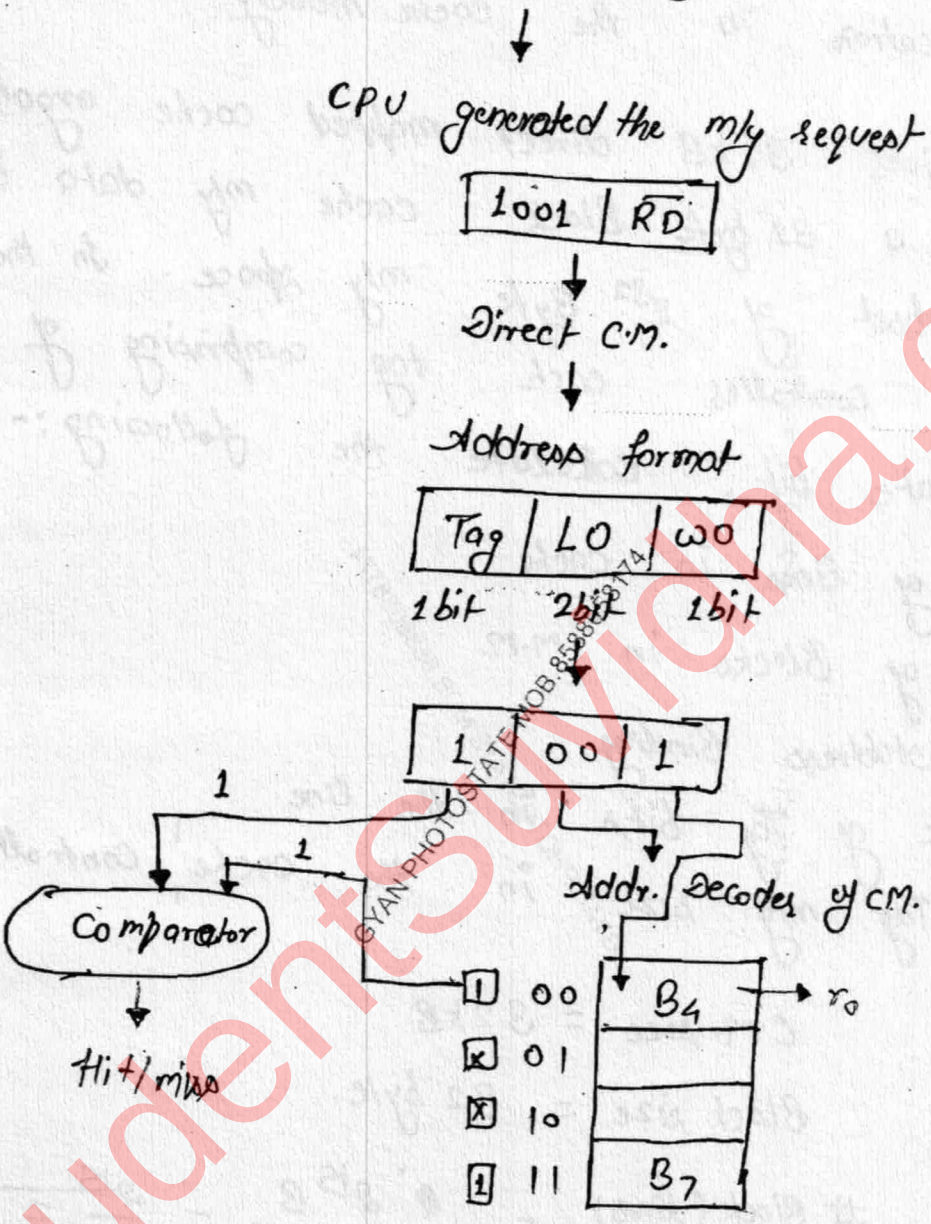


MM Block	Direct-MAP	CM-Line
B0 [000]	k mod N = i 0 mod 4 = 0	line '0'
B7 [111]	k mod N = i 7 mod 4 = 3	line '3'
B4 [100]	k mod N = i	line '0'

Date 21/08/2017

Accessing Sequence :-

machine Instⁿ: mov r0, [1001]



During the mapping process some of the Tag bits are connected to a address logic of cache memory and rest of the bits are stored in the coche controller ap Tag. so Tag m/y size:

$$\text{Tag m/y size} = \#\text{ lines in CM} * \#\text{ tag bits in line}$$

In the Direct cache design, Explicit Replacement policies (FIFO, LRU) are not required to replace the data because each main memory block is having fixed location in the cache memory.

Que:- Consider 32 kB direct mapped cache organized into a 32 byte Block. cache m/y data is a subset of 2^{32} Byte m/y space. In the cache controller each tag comprising of 1 update bit. Calculate the following :-

a) # of lines in cache
b) # of Blocks in M.M.
c) Address Binding
4.) # of Tag bits in the line
5.) Tag m/y size in the cache controller.

Soln :-

C.M. size = 32 kB

Block size = 32 byte.

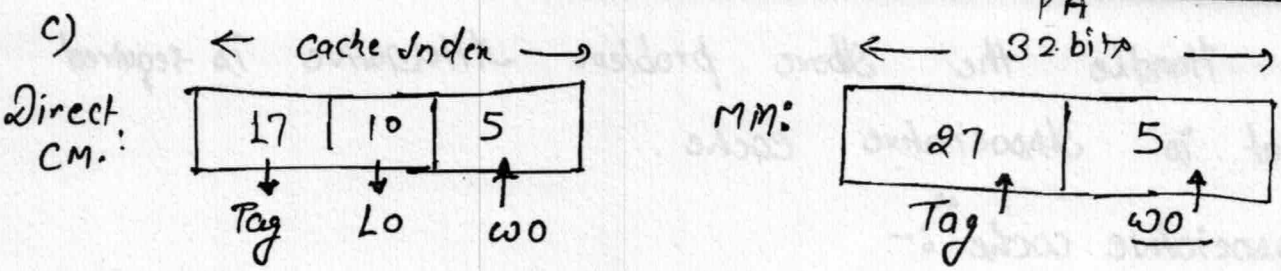
$$\#\text{ Block (lines) in C.M.} = \frac{2^{15}\,B}{32\,B} = \frac{2^{15}}{2^5} = 2^{13}$$

$$= \frac{2^{15}}{2^5} = 2^{10} = 1k$$

$$\#\text{ Blocks in M.M.} = \frac{2^{32}\,B}{32\,B} = \frac{2^{32}}{2^5} = 2^{27}$$

$$= 128\,M\text{ Blocks}$$

c)



d) 17

e) Tag m/y size = # lines in cache * # tag bits in the line

= 1K * (17 bit + 1 update bit)

= 18 K bits

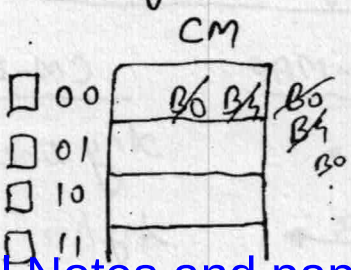
★ Note :-

:- Limitation in the Direct mapping is Each line is capable to hold only one Block (Tag) at a time therefore # of "Conflict misses" will be increased

:- when the CPU frequently refers the multiple Blocks which are mapped into a some cache line then cache Blocks are continuously swapping so Hit ratio falling down this phenomenon is called as "Thrashing" that is described below: CM Contain 4 lines (Empty)

MM Block reference : B0, B4, B0, B4, B0, B4 ....

Execution:



B0 - M: 0 mod 4 = 0 → compulsory miss

B4 - M: 4 mod 4 = 0 → conflict

B0 - M: 0 mod 4 = 0 → "

B4 - M: 4 mod 4 = 0 → "

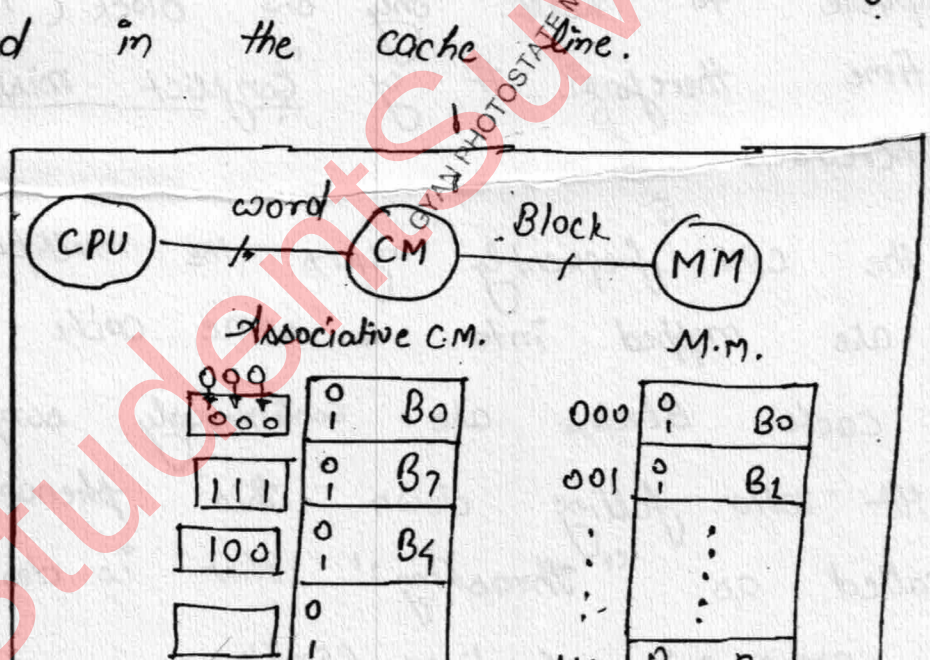
: To Handle the above problem Alternative is required that is Associative cache.

"Associative cache" :-

: Associative cache is designed without Address called as content Addressable memory.

: In this cache design any main-memory Block can be mapped into any cache m/y line in a sequence so that all the lines are effectively used. Replacement will be started only when the cache is full.

: During the mapping process, complete Physical Tag is stored in the cache controller so Tag-size is increased in the cache line.



MM Block	Associative-MAP	CM-line
B0	sequence →	Any line in sequence
[000]		
B7	sequence →	Any line in sequence

B4 [100] → sequence → Any line in sequence.

Address Binding :-

MM: | Tag | wo |
3bit 1bit

Associative CM: | Tag | wo |
3bit 1bit

Tag m/y size = # lines in CM * # Tag bits in line
= 4 * 3 bits
= 12 bits

Accessing sequence :-

machine Instn: mov r0, [1001]

↓

CPU generates the m/y request

| 1001 | RD |

↓

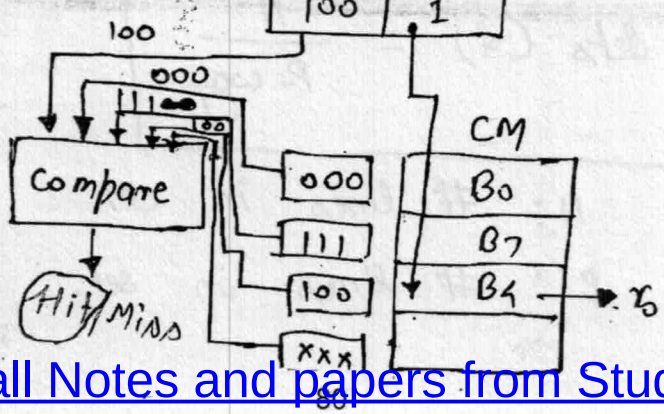
Associative CM.

↓

Address format

| Tag | wo |
3bit 1bit

| 100 | 1 |



★ Limitation in the Associative cache is all the Tags in the cache Controller is compared with a CPU generated Tag in parallel so implimentation of a parallel search comparison circuit become very complex and expensive. search Time is added to a Average Access time Hence Alternative is required name as: 'Set Associative Cache.'

★ In the Associative cache design explicit replacement techniques are used to replace the data when the cache is full (FIFO, LRU).

★

"Set Associative cache" :-

Set Associative Cache is used to compromise the Disadvantages in the Direct and associative cache Designs.

In this organization lines are grouped into a sets to ~~acc~~ accomodate the more than one Block (Tag) in the set at a time.

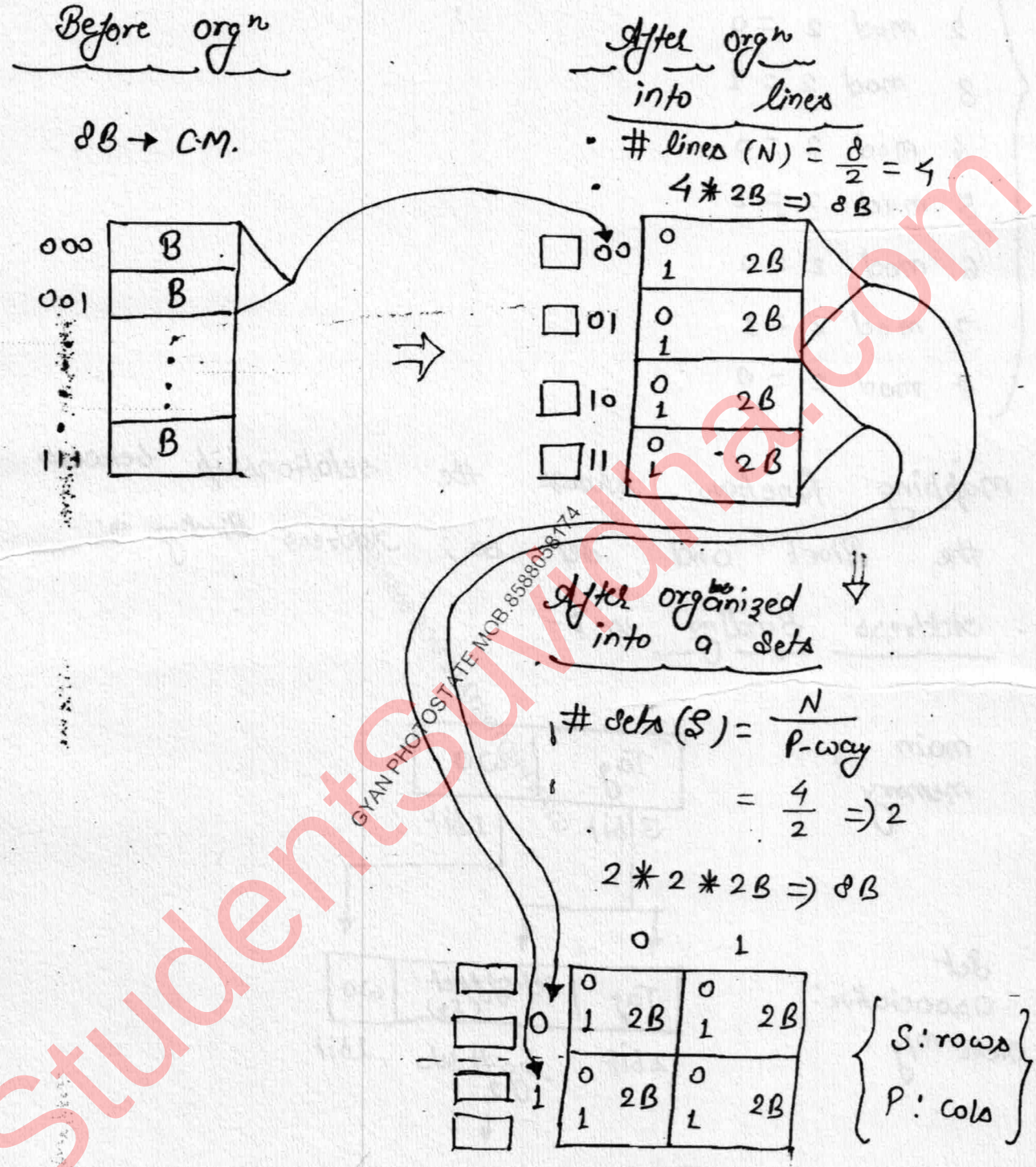
★

$$\#\text{ Sets } (S) = \frac{N}{P\text{-way}}$$

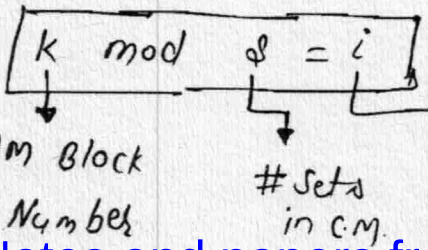
N: # lines in CM.

P: # lines in set.

eg: Consider 2-way set associate cache of 8B with 2B Block :



* In this cache Design mod function is used to map the data that is :

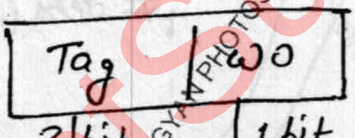


0 mod 2 = 0
1 mod 2 = 1
2 mod 2 = 0
3 mod 2 = 1
4 mod 2 = 0
5 mod 2 = 1
6 mod 2 = 0
7 mod 2 = 1
8 mod 2 = 0

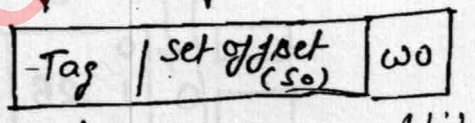
mapping function shows the relationship between the Block and set so; Address Binding is:

Address Binding is :-

main memory :



Set associative: cache m/y

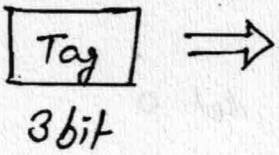


$\log_2$ # sets

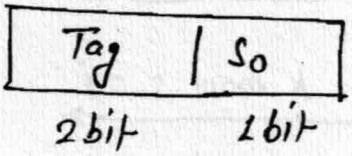
$\log_2 2$

1 bit

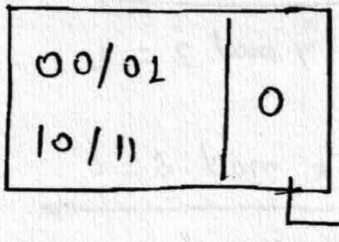
MM



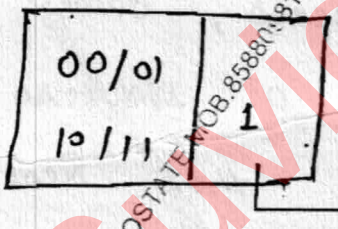
Set Associative m/y

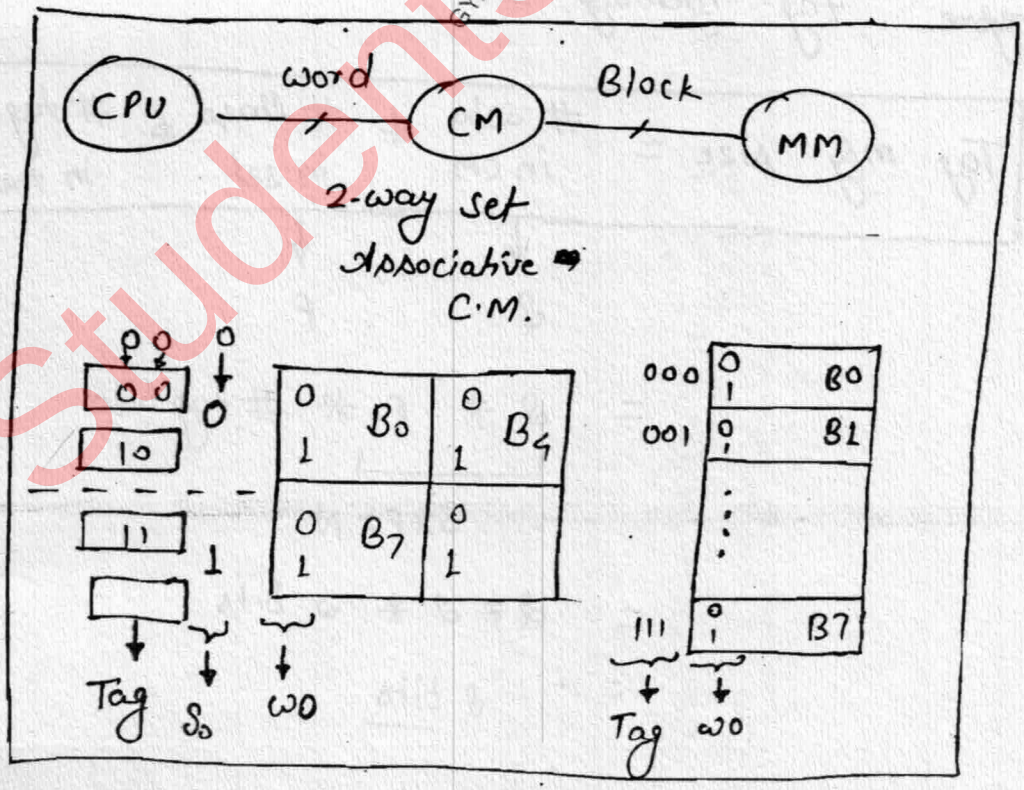


B_0 [000]
B_2 [010]
B_4 [100]
B_6 [110]



B_1 [001]
B_3 [011]
B_5 [101]
B_7 [111]





MM Block	Set Associative ~~cache~~ ~~mfg~~ map	C.M. Set
B_0 [000]	k mod s = i 0 mod 2 = 0	set '0'
B_7 [111]	k mod s = i 7 mod 2 = 1	set '1'
B_4 [100]	k mod s = i 4 mod 2 = 0	set '0'

In the mapping Process, some of the Tag bits are connected to a Address Logic of the Cache - memory and the Rest of the bits are stored in the Cache controller as a Tag therefore tag memory size;

$$\text{Tag m/y size} = \underset{\text{in CM}}{\#\text{sets}} * \underset{\text{in set}}{\#\text{lines}} * \underset{\text{in the line}}{\#\text{tag bits}}$$

$$= s * p * \#\text{tag bits}$$

$$S * P = N$$

$$= 2 * 2 * 2 \text{ bits}$$

$$= 8 \text{ bits}$$

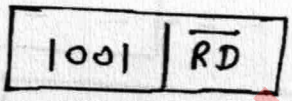
* Set Associative cache is designed with a explicit replacement policies, used to replace the data when the set is full.

Accessing Sequence :-

machine Instn: Mov r_0, [1001]

↓

CPU generates the m/y Request

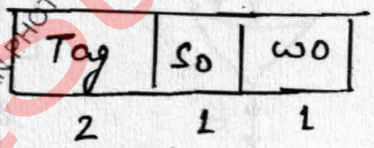


↓

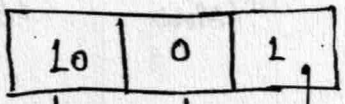
ee 2-way Set Associative C.M."

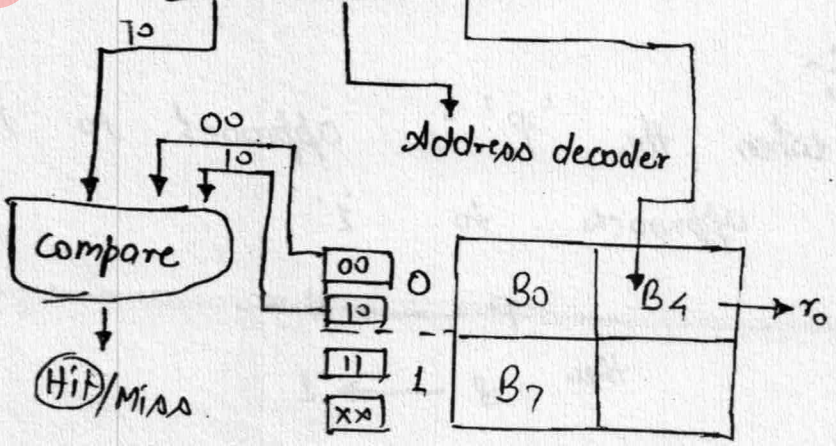
↓

Address format



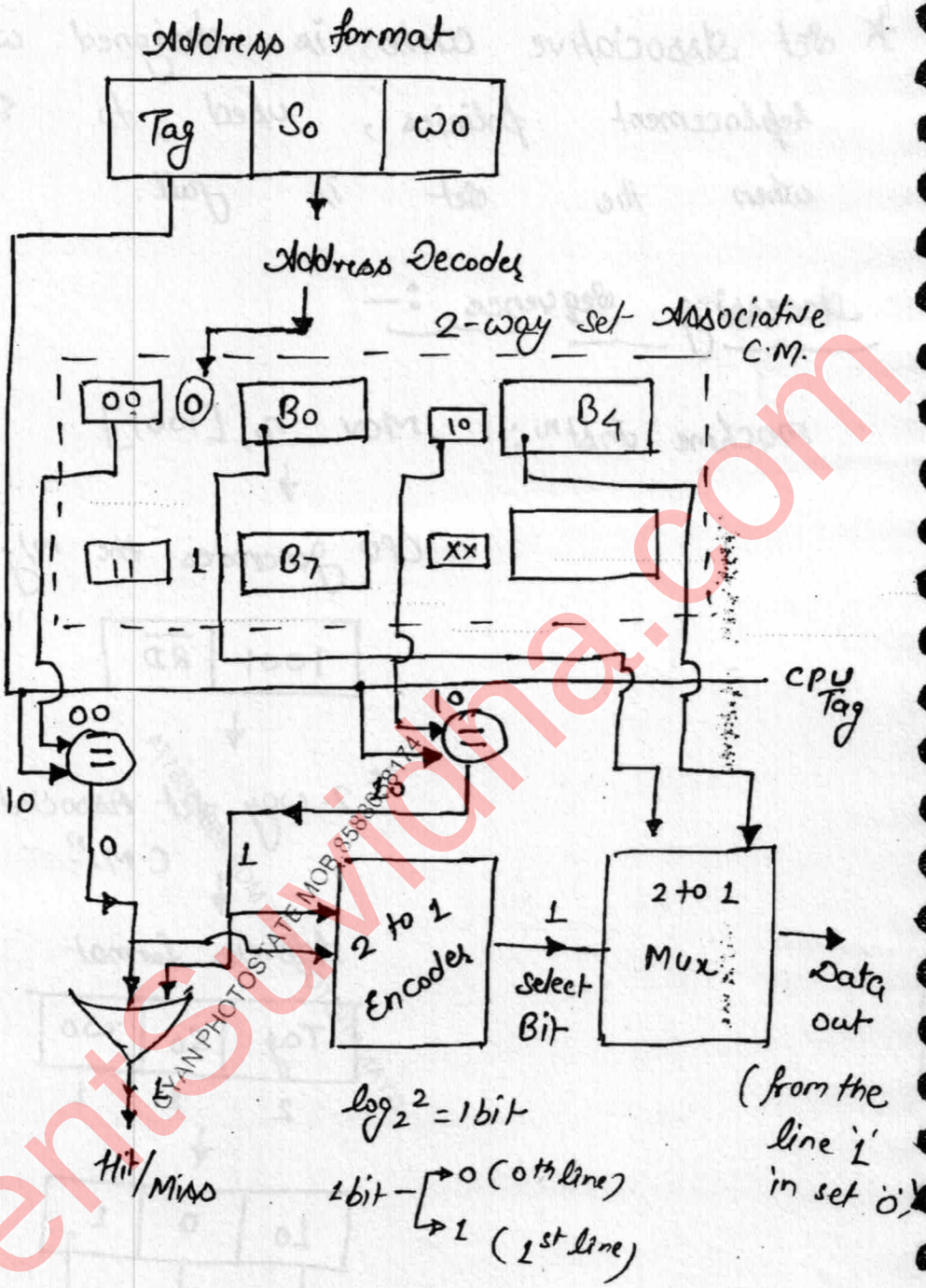
↓





Comparison :-

Ckt



Note :-

When the 'P' is approach to N then 's' is approach to '1'

P ⟶ N

then s ⟶ 1

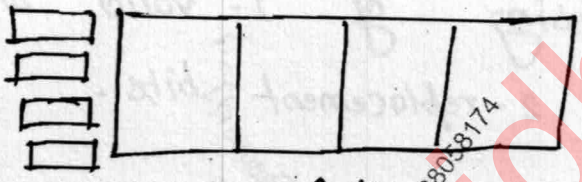
then, cache m/y behaves as a fully associative.

eg: 4 way set Associative Cache of 8B with 2B Block

* # lines (N) = $\frac{8}{2}$ = 4

* # sets (S) = $\frac{N}{P\text{-way}}$

= $\frac{4}{4}$ = 1



Fully Associative.

2.) when P is approach to 1 then S is approach to N, then cache behaves as a Direct.

P → 1

then S → N

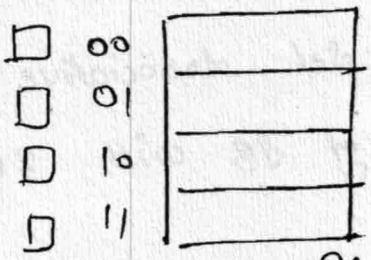
then set Asso. C.M. → Direct

eg:

1- way set Associative cache of 8B with 2B Block.

* # lines (N) = $\frac{8}{2}$ = 4

sets (S) = $\frac{N}{P\text{-way}}$ = $\frac{4}{1}$ = 4



Direct C.M.

Que:- Consider a 16-way set-associative cache of 64kB organized into a 32 Byte Block. CPU generates 32 bit address to access the data. In the cache controller each Tag comprising of 1-valid bit, 1 modified bit and 2-replacement bits.

a) Show the Address Binding.

b) # tag bits in the line

c) Tag m/y size.

P = 16

CM_{size} = 64kB = 16 bit

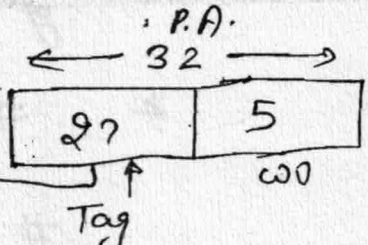
$m/m_{size} = 2^{32}B$

$Block_{size}$ = 32 Bytes

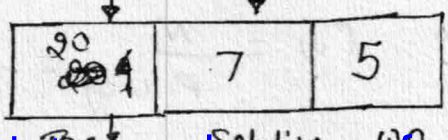
$$\# \text{ lines } (N) = \frac{2^{16}B}{2^5B} = 2^{11} = 2k$$

$$\# \text{ sets } (S) = \frac{2^{11}}{2^4} = 2^7$$

a) Address Binding :-



Set Ass. :-



b) # Tag bits in line = 20

c). Tag m/y size = $S * P *$ # tag bits

$= 2^7 * 16 *$ (20 bit + 1 valid bit + 1 mod bit + 2 replaced bit)

$= 2^7 * 2^4 *$ 24 bits

$= 48k$ bits

"Replacement Algorithms":-

:- Replacement Algorithms are used in the cache design, To Replace the data when the cache is full they are:

1) Random
2.) FIFO
3) LRU

In FIFO, replace the Block (cache block) which is having the Longest time-stamp.

In LRU, replace the cache block which is present in the cache longest time without reference.

Que:- Consider 4-Block cache (initially empty) with the following main m/y Block references 4, 5, 7, 12, 4, 5, 13, 4, 5, 7. identify the Hit ratio using:

a) ~~LRU~~ FIFO

b) LRU

c) Direct mapped cache

d) 2-way set associative with LRU.

a.)

4	5	7	12	4	5	13	4	5	7
4	4	4	4	4	4	13	13	13	13
	5	5	5	5	5	5	4	4	4
		7	7	7	7	7	7	5	5
			12	12	12	12	12	12	7
F	F	F	F			F	F	F	F

Hit ratio = $\frac{2}{10}$ = 20% = 0.2

b)

4	5	7	12	4	5	13	4	5	7
4	4	4	4	4	4	4	4	4	4
	5	5	5	5	5	5	5	5	5
		7	7	7	7	13	13	13	13
			12	12	12	12	12	12	7
F	F	F	F			F			F

Hit ratio = $\frac{6}{10}$ = 60% = 0.6

c.) Direct mapped :-

Cache

4	5
0	1

CM.

0	4 4 12
1	5 5 13
2	
3	7 ✓

k mod N = i

4

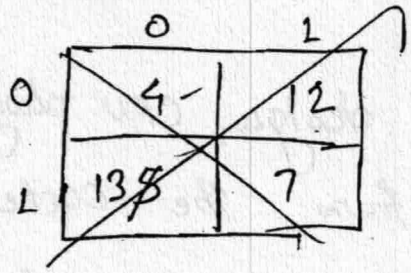
C	C	C	C ✓	
4	5	7	12	4 5
M	M	M	M	M H
C		✓		
13	4	5	7	
M	H	M	H	

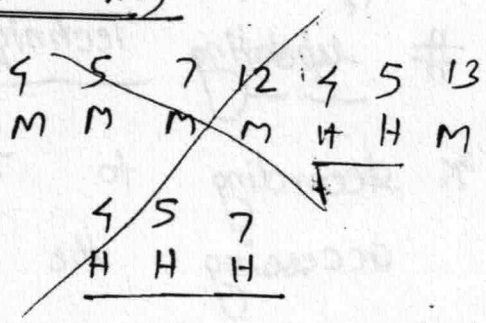
5 compulsory miss

2 conflict miss

$\frac{3}{10}$ = 0.3

d). 2- way set Associative (with LRU)



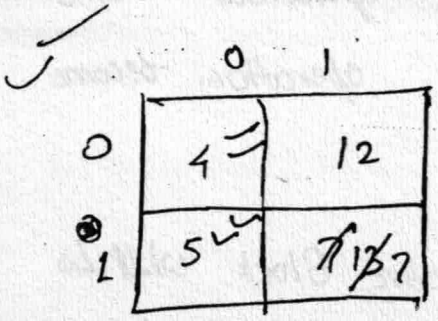


k mod s = i

↓

mod 2

4 Hit = 4/10 = 0.4



4 mod 2 = 0
5 mod 2 = 1
7 mod 2 = 1
12 mod 2 = 0
13 mod 2 = 1

4 - M
5 - M
7 - M

12 - M
4 - H
5 - Hit
13 - m
4 - Hit

5 Hit, 7 miss
7 mod 2 = 1

4/10 0.4