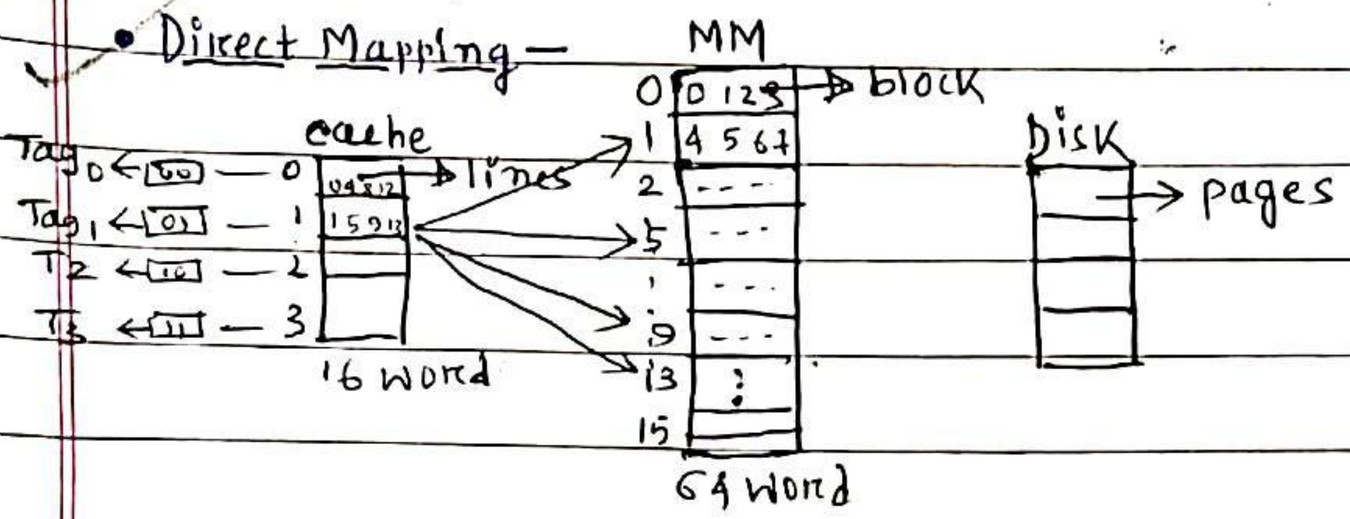
COA - SN (Short Note)

**

- Cache Memory :

W = 4B

- Direct Mapping -



→ No. of Block in MM = MM size / Block size → Physical address = 2^6 → 6 bit

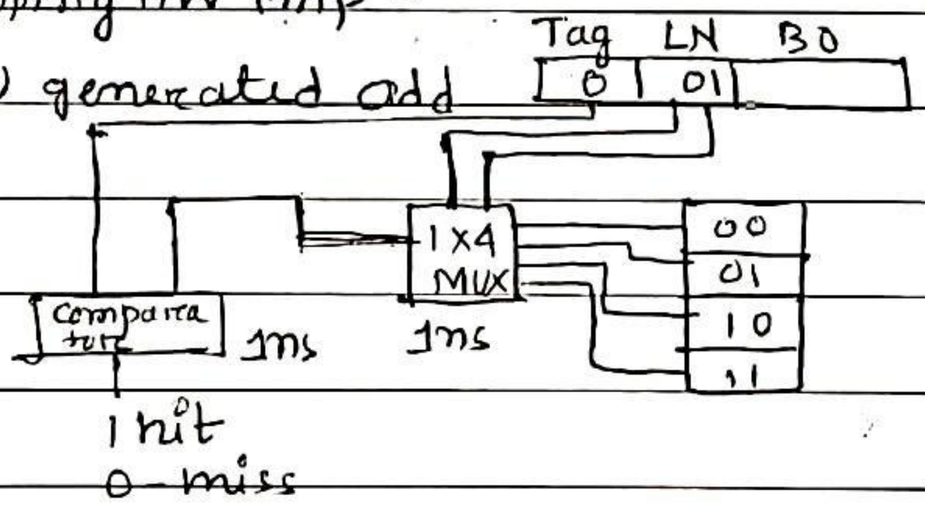
** → P.A = | Tag (2) | Line no (2) | 2 |

Block num (6 bit) | Block offset (Block size)

→ no. of line = Cash size / line size → line size = Block size

→ Tag Directory size = (Tag size * no. of lines)

- Direct Mapping HW Imp -



→ Total time taken = (latency of MUX + latency of comparator)
= (1 + 1) ns = 2ns.

* → no. of MUX = no. of Tag bits

Direct mapping → Tage size = K bit
MUX = K & comparator = 1 * (K bit compare)

* → Tag = Main Memory size / cache size

- Dis adv: → conflict miss problem.

Block req by CPU 5, 6, 9, 8, 9, 12, 15, 20

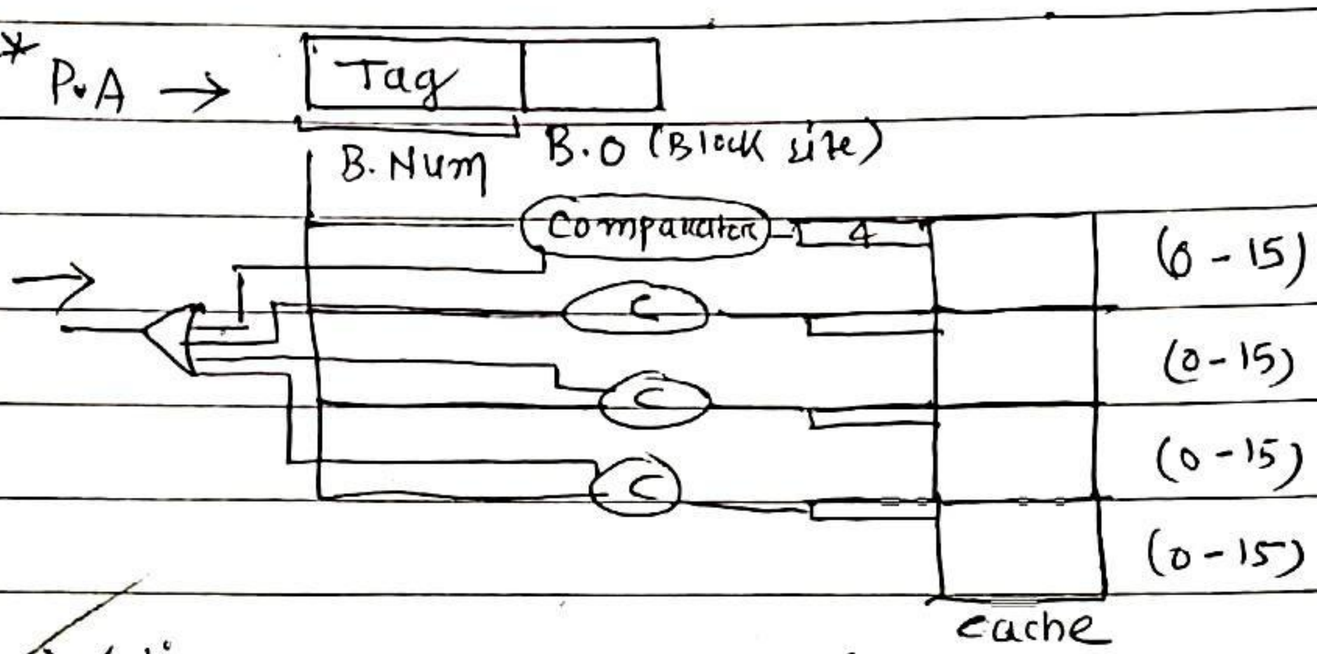
0	A	8 12 20
1	B	9
2	6	
3	15	

(L.N = B.N % NL)
Block num.
→ No. of Line.

Associative Mapping —

* → can solve conflict miss problem.

** P.A →



→ 4 lines req = 4 Comparator, so HW cost increase.

* → no. of comparators = no. of cache lines

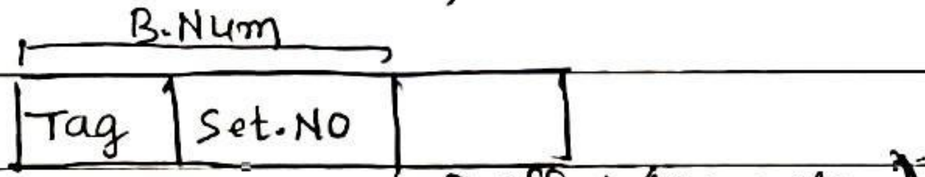
* → hit latency = Pd of comparators + PD of OR gate

Set-Associative Mapping —

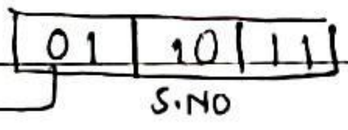
→ no. of comparator reduced.

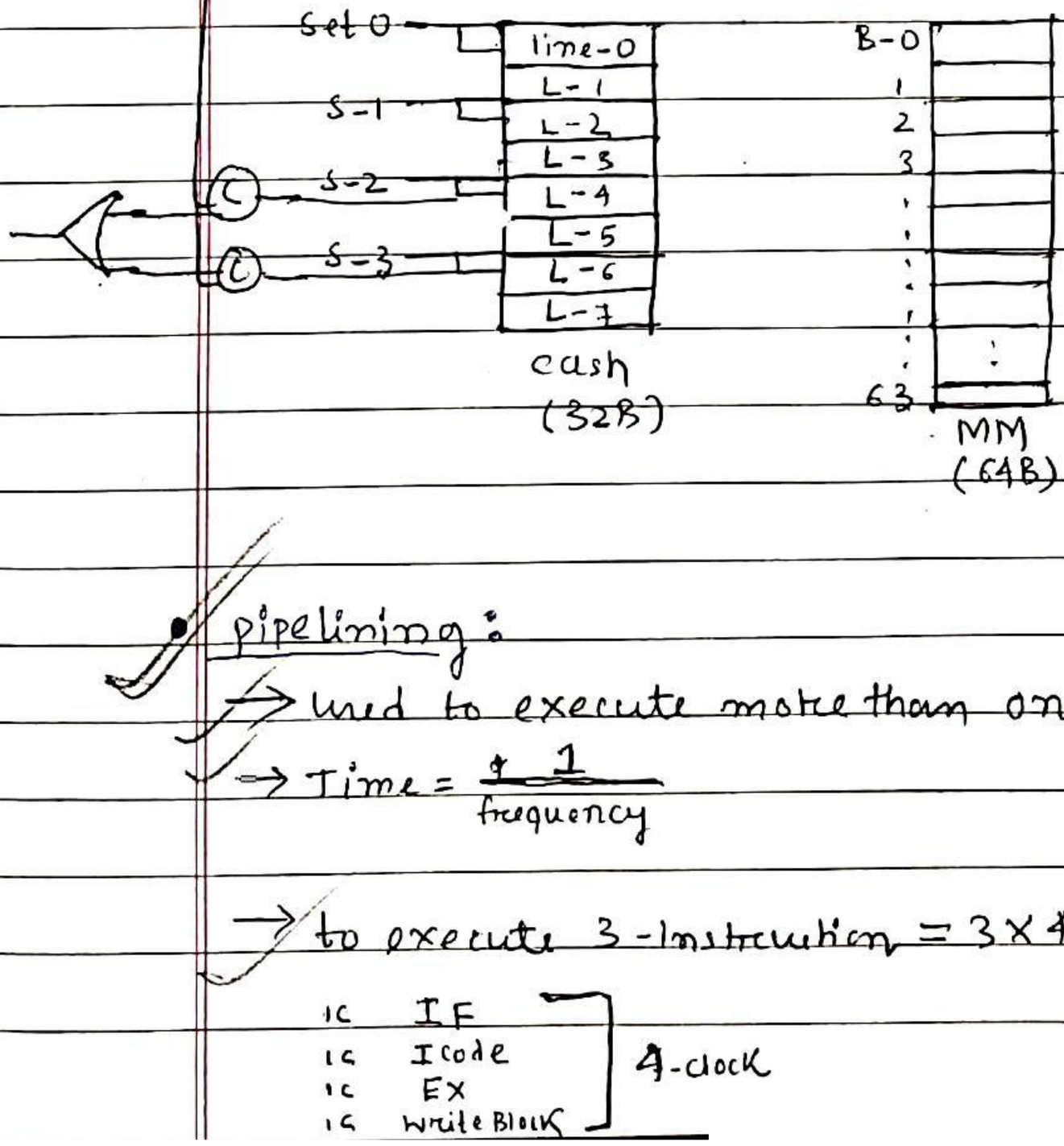
→ K-way set associative req K-Comparator.

** P.A =



ex: PA =





- MM size = 64B
- cache S = 32B
- Block size = 4B
- Set size = 2 Blocks (lines)
  ↓
  2-way set associative
- Sets = $\frac{\text{Lines (no. of)}}{\text{2-Way set (set size)}} = \frac{8}{2} = 4$

pipelining :

→ used to execute more than one Instruction at same time.

→ Time = $\frac{1}{\text{frequency}}$

→ to execute 3-Instruction = 3 × 4 = 12 clock pulse req (Without pipelining)

IC IF
IC Icode
IC EX
IC Write Block
} 4-clock

** → Phase time Diagram -

	1	2	3	4	5	6	7	8	9	10	11	12
IF	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9			
ID		I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9		
EX			I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	
WB				I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9

now to execute 3-instruction = 6 clock req (with pipelining)

→ Utilization or efficiency = $\frac{\text{Total no. of box used}}{\text{Total no. of box}}$

$= \frac{36}{4 \times 12} = 3/4$

→ ~~Seed~~ Speed up = $\frac{\text{Time taken without pipeling}}{\text{Time taken with pipeling.}} = \frac{T_{WP}}{T_P} = \frac{nm}{n+(m-1)}$

Speed up = m X efficiency

↳ no. of phases, here $m = 4$.

→ $\boxed{T_p = 1 \times m + (n-1) \times 1}$ → with pipeling.

n → no. of instruction.

$\boxed{T_{WP} = n \times m}$ → without pipeling.

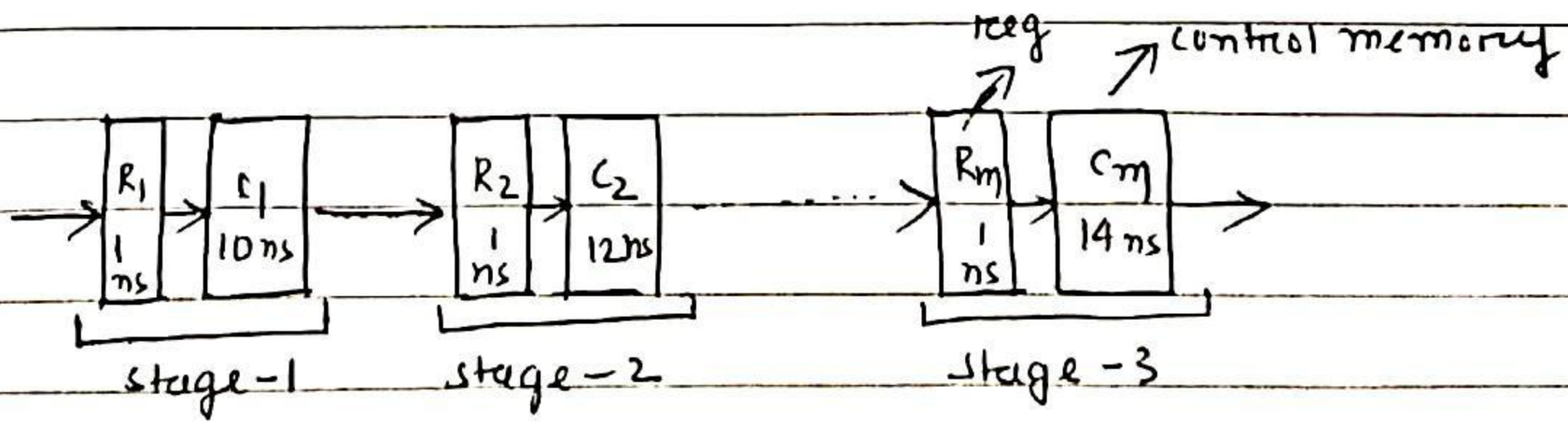
→ Q.

	S_1	S_2	S_3	S_4
I_1	2	1	1	1
I_2	1	3	2	2
I_3	2	1	1	3
I_4	1	2	2	2

=>

CLK -	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	S_1	S_1	S_2	S_3	S_4										
I_2	X	X	S_1	S_2	S_2	S_2	S_3	S_3	S_4	S_4					
I_3		X	S_1	S_1	X	S_2	X	S_3	X	S_4	S_4	S_4			
I_4			X	X	S_1	X	S_2	S_2	S_3	S_3	X	X	S_4	S_4	

• Pipeline ex -



CPU CPI = 1

T = 14 ns + 1 ns = 15 ns

$f = \frac{1}{15\ ns}$