Memory

SRAM: (Static Random Access Memory)
- Value is stored on a pair of inverting gates.
- Very fast but takes up more space than DRAM (4 to 6 transistors)

DRAM: (Dynamic Random Access Memory)
- Value is stores as a charge on capacitor (must be refreshed)
- Very small but slower than SRAM (factor of 5 to 10)

Memory Hierarchy:
- Different layer/ levels of memory.
- Memory update করা হয় ২টি জিনিসের উপর depend করে:-

1. Temporal Locality (Lacality in Time):
   - Keep most recently accessed data items closer to the processor.
2. Spatial Locality (Locality in Space):
   - Move backs consists of contigeouous words to the upper levels.

General Principles of Memory:

Locality–

Temporal Locality: Referenced memory is likely to be referenced again soon (e.g. code within a loop).

Spatial Locality: Memory close to referenced memory is likely to be referenced soon (e.g. data is In a sequential access array)

Definition–

Upper: Memory closer to processor.

Block: Minimum unit that is present or not present.

Block address: Location of block in memory.

Hit: Data is found in the desired location.

Hit time: Time to access upper level.

Miss Rate: Percentage of time item not found in upper level.

Three types of mapping memory:–

1. Direct Mapping.
2. Set Associative.
3. Full Associative.

1. Direct Mapping:

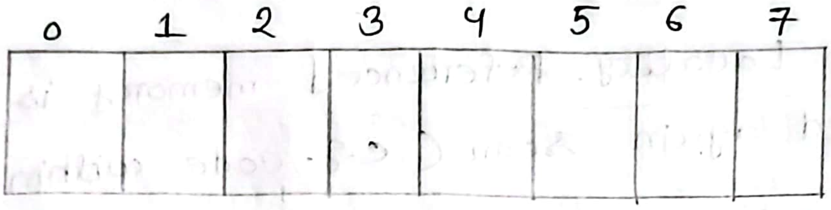
Mapping: Memory mapped to one location in cache.

Process:

1. CPU request → 2. Checking in cache → 3. if hit → processor. 4. else miss

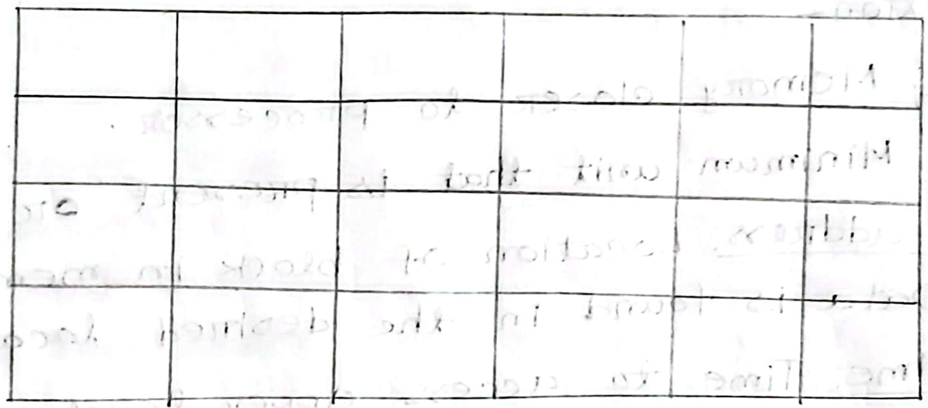
5. Main memory $\boxed{\rightarrow}$ cache → Processor
↳ Mapping functions

Cache Memory:



There are 8 blocks in cache memory.

Main Memory:



Send request to CPU. It first checks the cache memory for the data. If the data is found than data hit, the processor receives it from the cache. If not than data miss. The data is fetched from main memory and placed into the cache, then sent sent to the processor.

Formula:

Direct mapping = (requested address) mod (# of Block in cache)

Example:

CPU requested address:

8, 3, 5, 8, 11, 25.

0	1	2	3	4	5	6	7
MEM [8]	MEM [25]		~~MEM [3]~~ MEM [11]		MEM [5]		

cache memory

For 8 →

8 mod 8 = 0 (miss)

place in 0th index

For 3 →

3 mod 8 = 3 (miss)

place in 3rd index

For 5 →

5 mod 8 = 5 (miss)

place in 5th index

For 8 →

8 mod 8 = 0 (hit)

No need to place it.

For 11 →

11 mod 8 = 3 (miss)

~~Repla~~ As, there is already an address in 3rd index, replace it with the new one.

For 25 →

25 mod 8 = 1 (miss)

place it in 1st index

Direct Mapping in cache memory done.

2. Set associative: (N-way set associative)

2-way associative:

Formula:

1. (# of Block in cache) ÷ (# of way)
2. (Requested address) mod (# of sets)

Example:

Requested address:

8, 3, 5, 8, 11, 3, 25.

Sets = (# of block in cache) ÷ (# of way)

= 8 ÷ 2

= 4 sets.

Here,
# of block in cache = 8.
# of way = 2

0		1		2		3	
0	1	0	1	0	1	0	1
MEM [8]		MEM ~~[3]~~ [5]	MEM [25]			MEM [3]	MEM [11]

(Requested address) mod (# of ways)

for 8 →

8 mod ~~2~~ 4 = 0 (miss)

place in 0th ~~index~~ set's in any index.

~~5~~ for 3 →

3 mod ~~2~~ 4 = ~~1~~ 3 (miss)

place in ~~4th~~ 3rd set's in any index

for 5 →

5 mod ~~2~~ 4 = 1 (miss)

place in 1st set's in any index

for 8 →

8 mod 4 = 0 (hit) as 8 ~~e~~ is already in the memory.

No ~~ned~~ need to place it.

for 11 →

11 mod 4 = 3 (miss)

~~place~~ as there ~~are~~ is another option in 3rd set.

place it in 3rd set's index.

for 3 →

3 mod 4 = 3 (hit)

No need to place it.

for 25 →

25 mod 4 = 1 (miss)

place it in 1st ~~is~~ set.

set associative (2-ways) mapping done.

Note:

set এর option সব fill-up হলে আরেকটি আসলে প্রথমে যেটি যাবে সেটিকে replace করতে হবে।

8

Same example for 4-ways:

8 ÷ 4 = 2 sets

0				1			
0	1	2	3	0	1	2	3
MEM [8]				MEM [3]	MEM [5]	MEM [11]	MEM [25]

8 mod 2 = 0 (miss)

3 mod 2 = 1 (miss)

5 mod 2 = 1 (miss)

8 mod 2 = 0 (hit)

11 mod 2 = 1 (miss)

3 mod 2 = 1 (hit)

25 mod 2 = 1 (miss)

Example:

CPU requested the following Block addresses (x+3), (x+5), (x+2) and (x+3). There are 16 one-word blocks in cache. Design and show the memory mapping for the following cache configurations.

(i) Direct mapped.

(ii) 4-way, 8-way and 16-way set associative mapped. (use LRU replacement policy).

Where X = last two digits of your ID. (50);

Ans:

~~Requ~~

CPU requested addresses = 50+3 = 53, 50+5 = 55, 50+2 = 52, 50+3 = 53.

53, 55, 52, 53.

~~Dire~~

(i) Direct mapped :-

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				MEM [52]	MEM [53]		MEM [55]								

No. of blocks in cache = 16

53 mod 16 = 5 (miss)

55 mod 16 = 7 (miss).

52 mod 16 = 4 miss

53 mod 16 = 5 (hit)

(11) Set associative :

<u>2-ways :</u>

No of blocks in cache = 16

No. of ways = 2

So, 16 ÷ 2 = 8 sets.

0		1		2		3		4		5		6		7	
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
								MEM [52]		MEM [53]				MEM [55]	

53 mod 8 = 5 (miss)

55 mod 8 = 7 (miss)

52 mod 8 = 4 (miss)

53 mod 8 = 5 (hit)

<u>4-ways:</u>

16 ÷ 4 = 4 sets.

0				1				2				3			
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
MEM [52]				MEM [53]								MEM [55]			

553 mod 4 = $\overset{1}{\cancel{3}}$ (miss)

55 mod 4 = 3 (miss)

552 mod 4 = 0 (miss)

53 mod 4 = 1 (hit)

8-ways:

16 ÷ 8 = 2 sets

0								1							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
MEM [52]				~~ME~~				MEM [53]	MEM [55]						

53 mod 2 = 1 (miss)
55 mod 2 = 1 (miss)
52 mod 2 = 0 (miss)
53 mod 2 = 1 (hit)

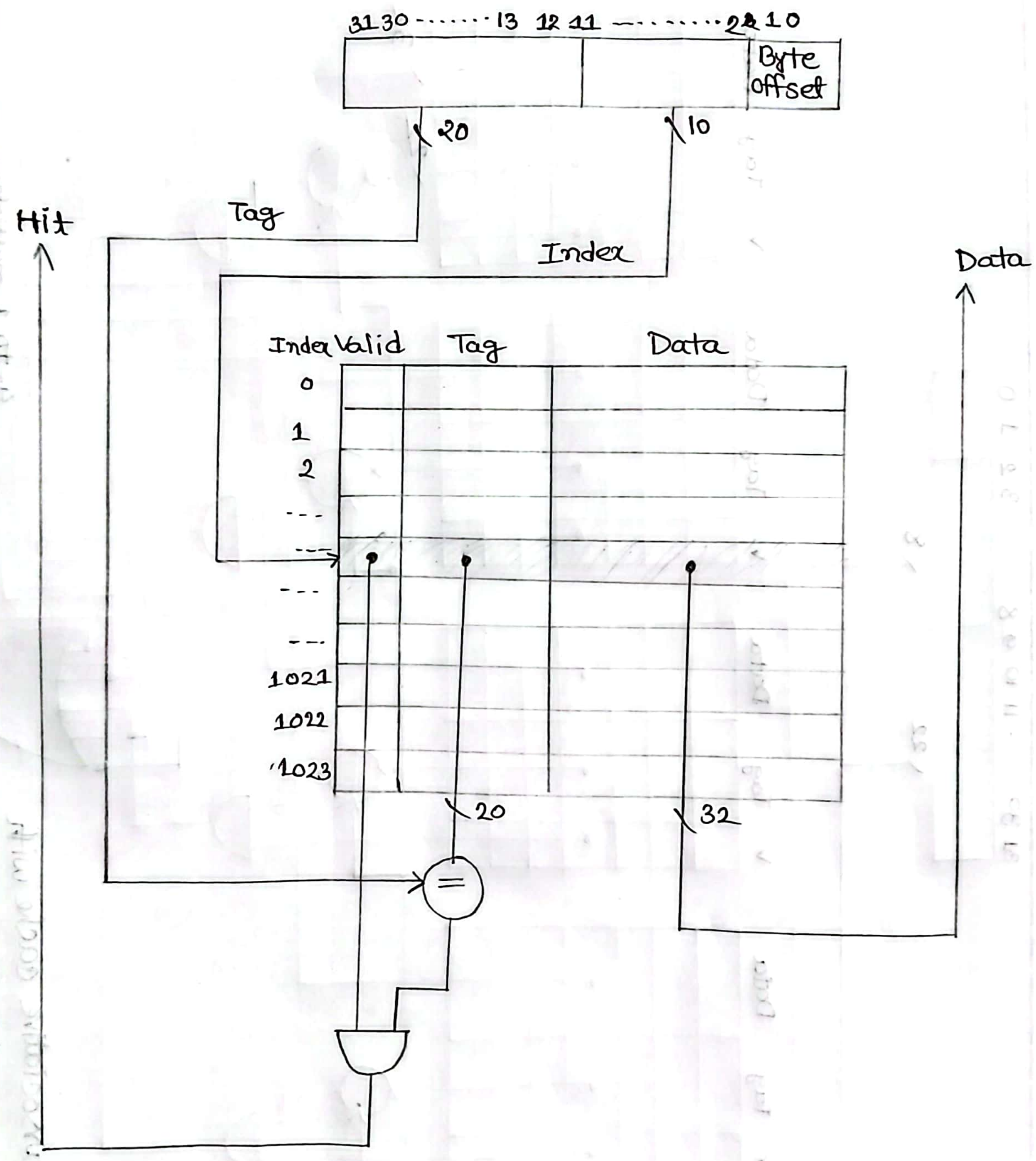
16-ways:

16 ÷ 16 = 1 sets

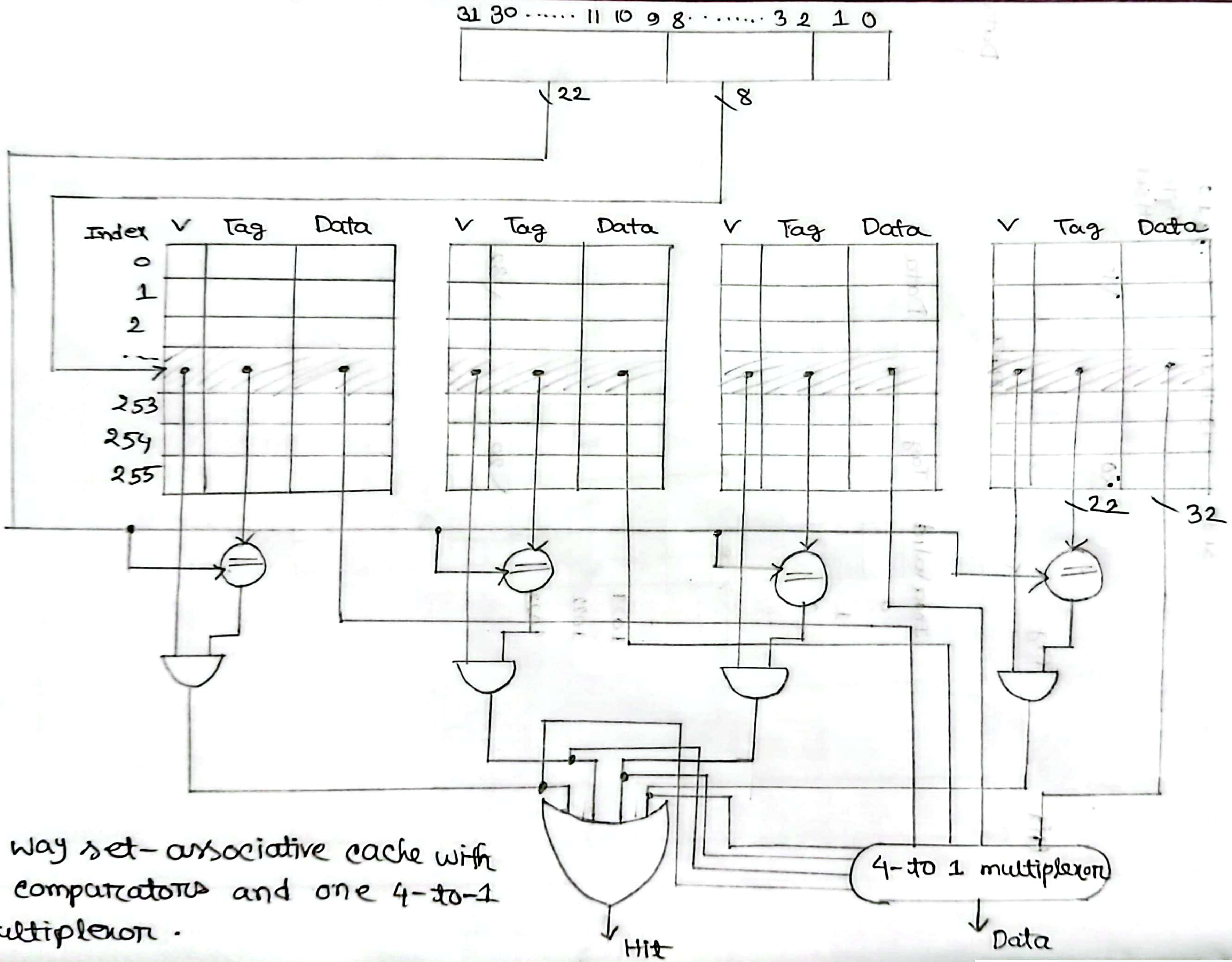
0															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MEM [53]	MEM [55]	MEM [52]													

53 mod 1 = 0 (miss)
55 mod 1 = 0 (miss)
52 mod 1 = 0 (miss)
53 mod 1 = 0 (hit)

Direct Mapped Cache (diagram):



Set Associative cache (diagram):



4 way set-associative cache with 4 comparators and one 4-to-1 multiplexor.