Exploiting Memory Hierarchy

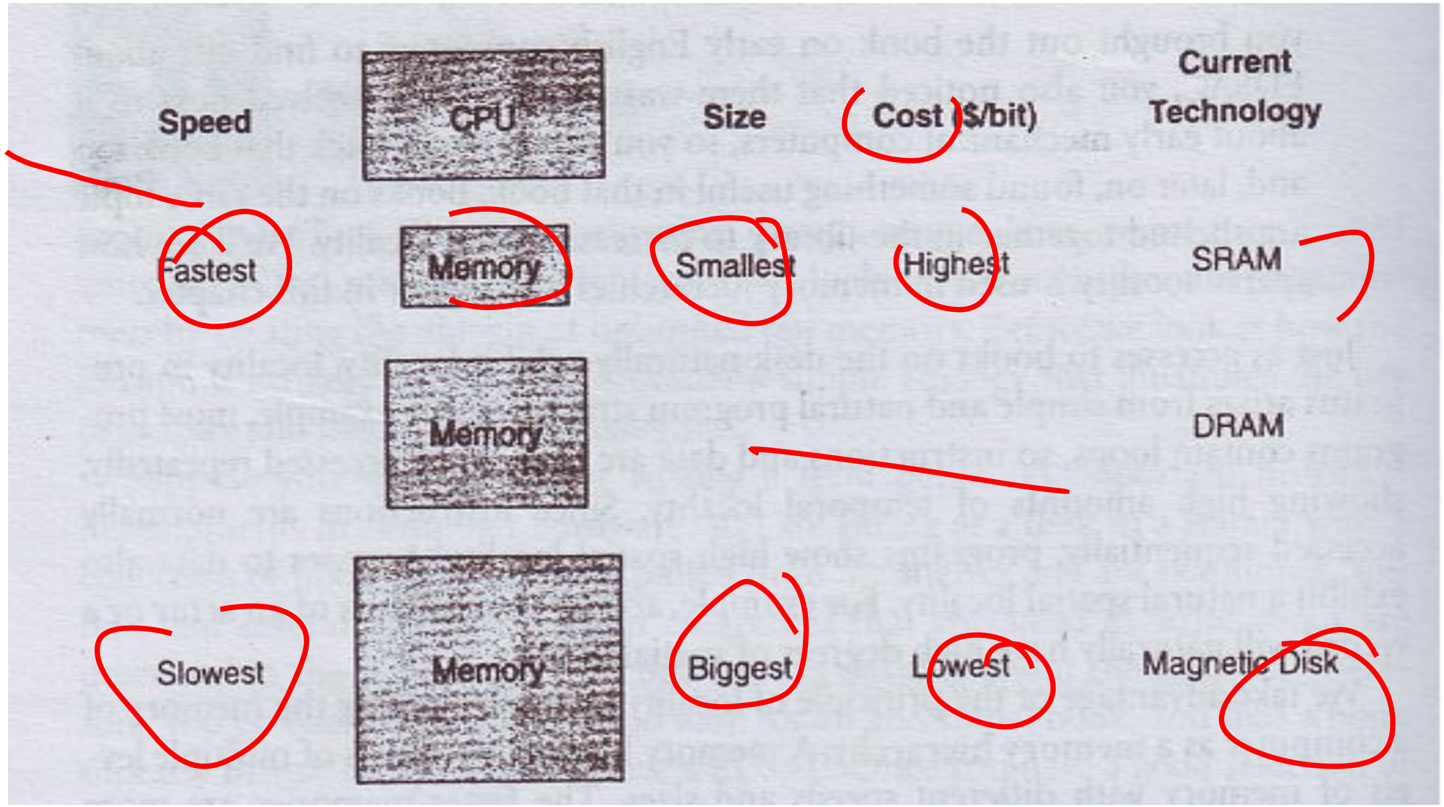
Chapter Five of Book of David A. Patterson

Memory Hierarchy

- ✔ A memory hierarchy consists of multiple levels of memory with different speeds and sizes.
- ✔ Three technologies used in building memory hierarchies:

1. DRAM
2. SRAM
3. Magnetic disk

The Basic Structure of Memory Hierarchy



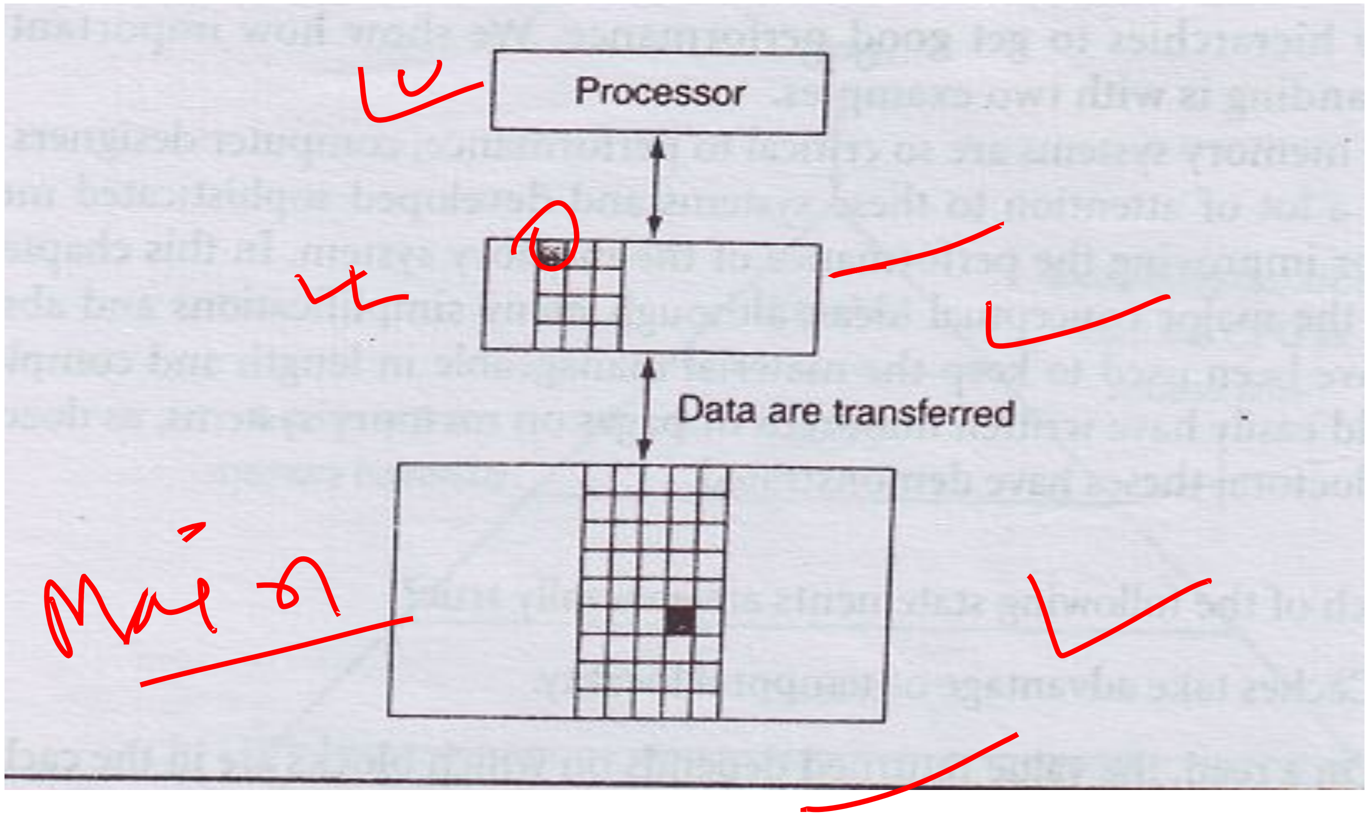
Memory Hierarchy

- ✔ Table:

Memory technology	Typical access time	$ per GiB in 2012
SRAM semiconductor memory	0.5–2.5 ns	$500–$1000
DRAM semiconductor memory	50–70 ns	$10–$20
Flash semiconductor memory	5,000–50,000 ns	$0.75–$1.00
Magnetic disk	5,000,000–20,000,000 ns	$0.05–$0.10

- ✔ The goal is to provide the user with as much memory as is available in the cheapest technology, while providing access at the speed offered by the fastest memory.

Two level Hierarchy



Terminology

- ✓ **Block:**

  The minimum amount of information that can be either present or not present in the two level hierarchy.

- ✓ **Hit:**

  Data requested by the processor appears in some block in the upper level.

- ✓ **Miss:**

  Data requested by the processor is not present in the upper level.

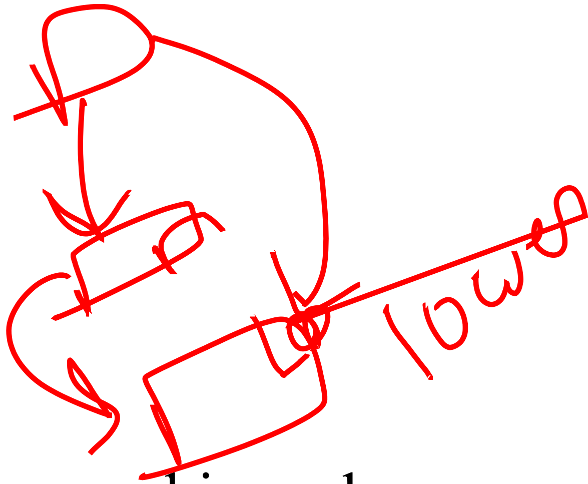
- ✓ **Hit rate:**

  The fraction of memory accesses found in the upper level.

- ✓ **Miss rate:**

  The fraction of memory accesses not found in the upper level. (1-Hit rate)

Terminology



- ✔ **Hit Time:**

  The time needed to access a level of the memory hierarchy, including the time required to determine whether the access is a hit or a miss.

- ✔ **Miss Penalty:**

  The time required to fetch a block into a level of the memory hierarchy from the lower level, including the time to access the block, transmit it from one level to the other and insert it in the level that experienced the miss.

Principle of Locality

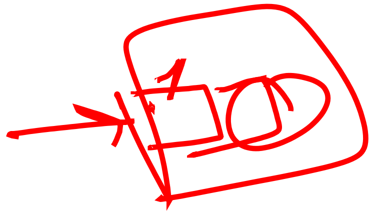
It states that program access a relatively small portion of their address space at any instant of time.

<u>Temporal Locality:</u>

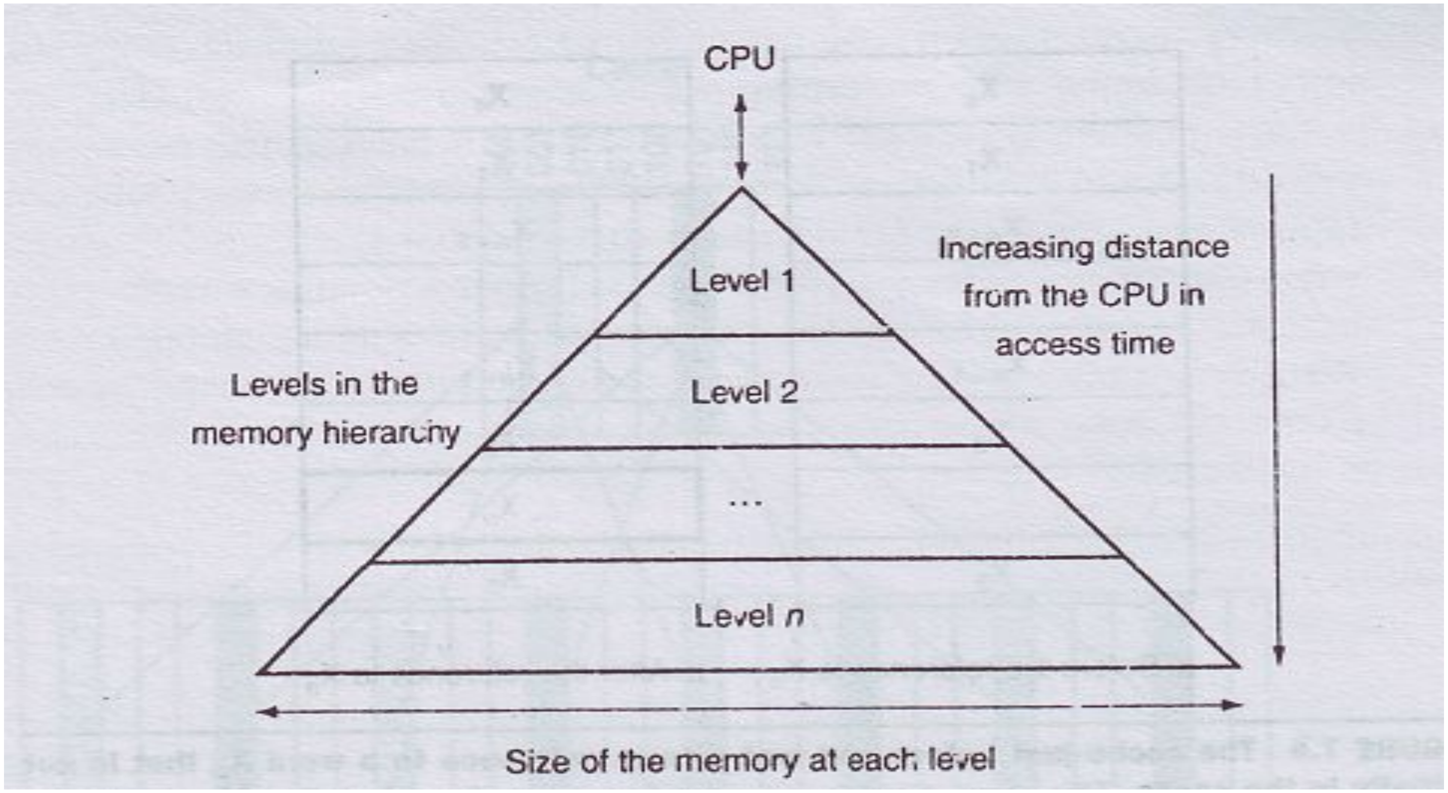
If an item is referenced, it will tend to be referenced again soon.

<u>Spatial Locality:</u>

If an item is referenced, items whose addresses are close by will tend to be referenced soon.

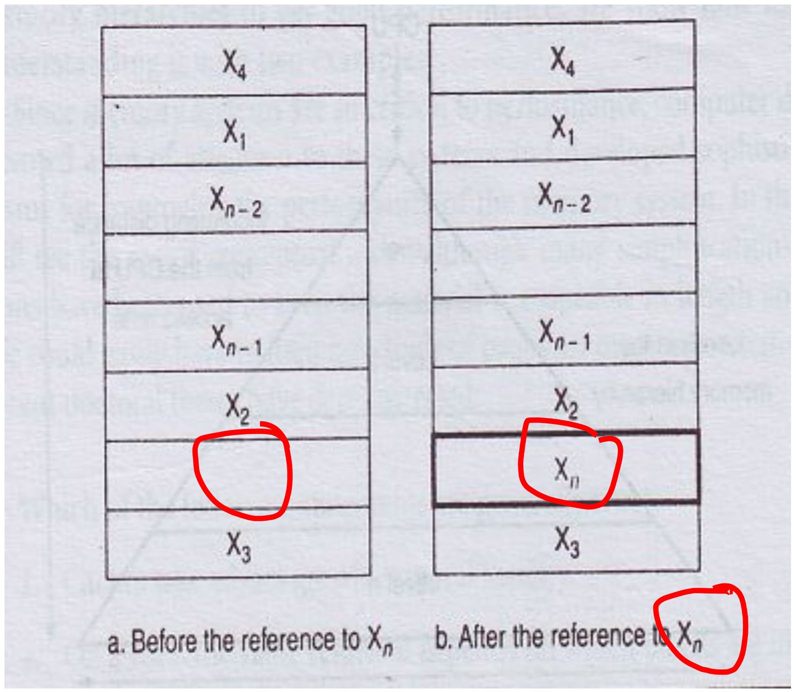


Memory Hierarchy


CPU
Level 1
Level 2
...
Level n
Increasing distance from the CPU in access time
Levels in the memory hierarchy
Size of the memory at each level

Cache

- ✔ It refers to the level of memory hierarchy between the processor and main memory.



a. Before the reference to X_n b. After the reference to X_n

12%, 8

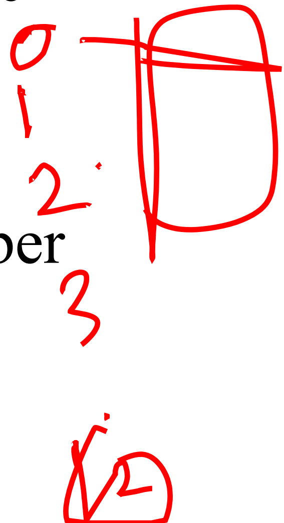
Direct Mapped Cache

- ✔ A cache structure in which each memory location is mapped to exactly one location in the cache.
- ✔ Assign cache location based on the address of the word in the memory.
- ✔ Mapping:

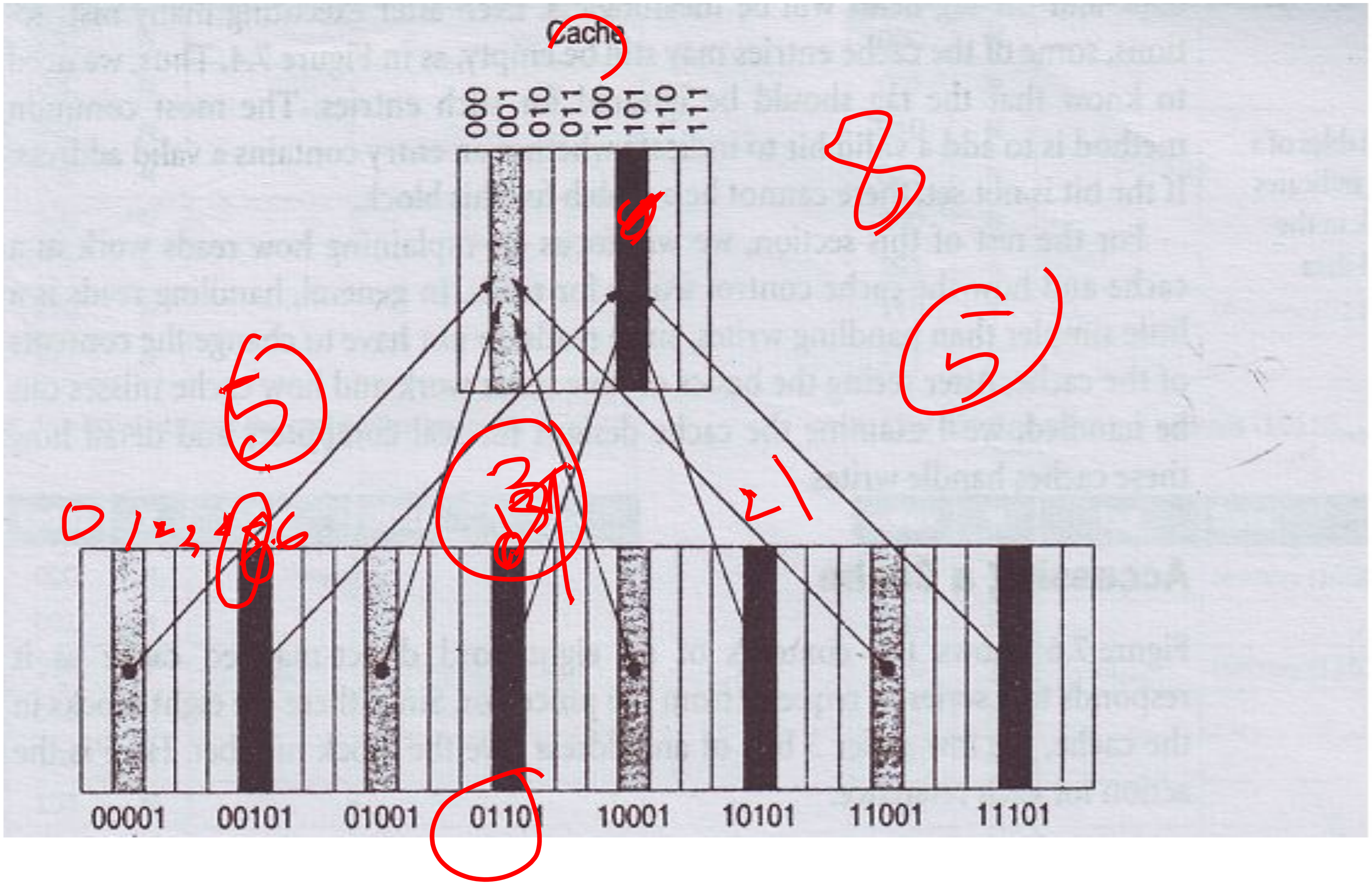
  (Block address) modulo (Number of cache blocks in the cache).

7 word

- ✔ Can accessed directly with the lower order bits.
- ✔ Each cache location can contain the contents of a number of different memory locations.



A Direct Mapped Cache

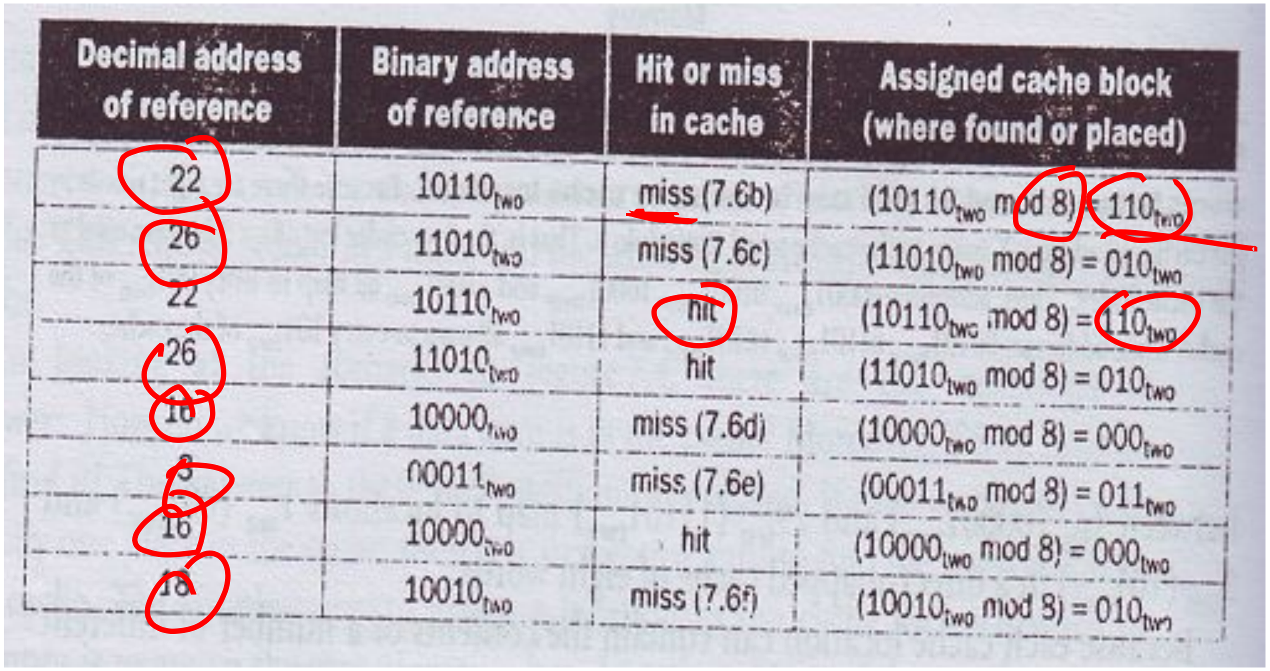


Tag and Valid Bit

- ✔ A field contains the address information required to identify whether a word in the cache corresponds to the requested word.
- ✔ It indicates that the associated block contains valid data.

Accessing A Cache

22 26



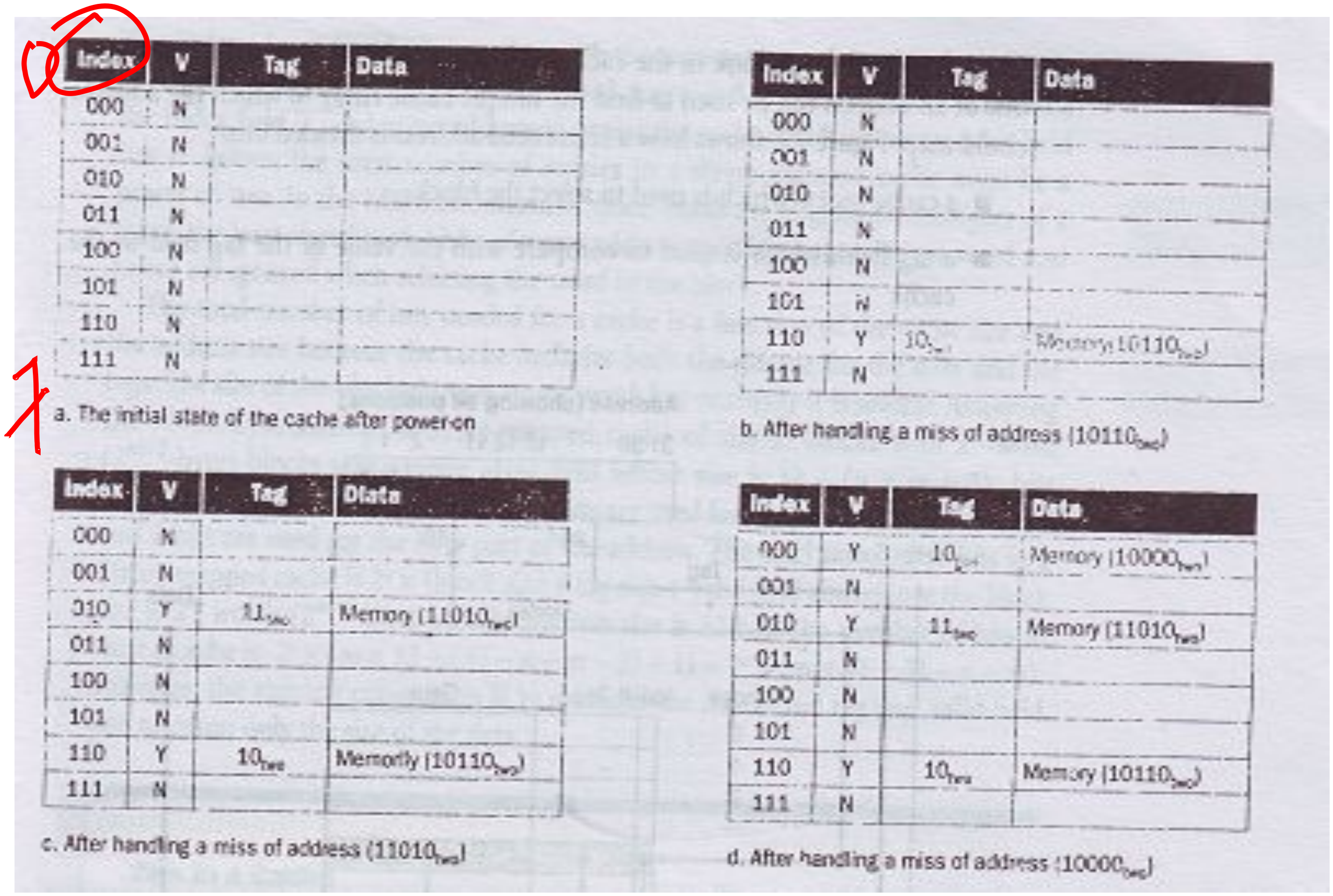
Decimal address of reference	Binary address of reference	Hit or miss in cache	Assigned cache block (where found or placed)
22	10110_{two}	miss (7.6b)	$(10110_{two} \bmod 8) = 110_{two}$
26	11010_{two}	miss (7.6c)	$(11010_{two} \bmod 8) = 010_{two}$
22	10110_{two}	hit	$(10110_{two} \bmod 8) = 110_{two}$
26	11010_{two}	hit	$(11010_{two} \bmod 8) = 010_{two}$
16	10000_{two}	miss (7.6d)	$(10000_{two} \bmod 8) = 000_{two}$
3	00011_{two}	miss (7.6e)	$(00011_{two} \bmod 8) = 011_{two}$
16	10000_{two}	hit	$(10000_{two} \bmod 8) = 000_{two}$
18	10010_{two}	miss (7.6f)	$(10010_{two} \bmod 8) = 010_{two}$

Accessing A Cache

32

8L

32



Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

7

a. The initial state of the cache after power-on

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

b. After handling a miss of address (10110_{two})

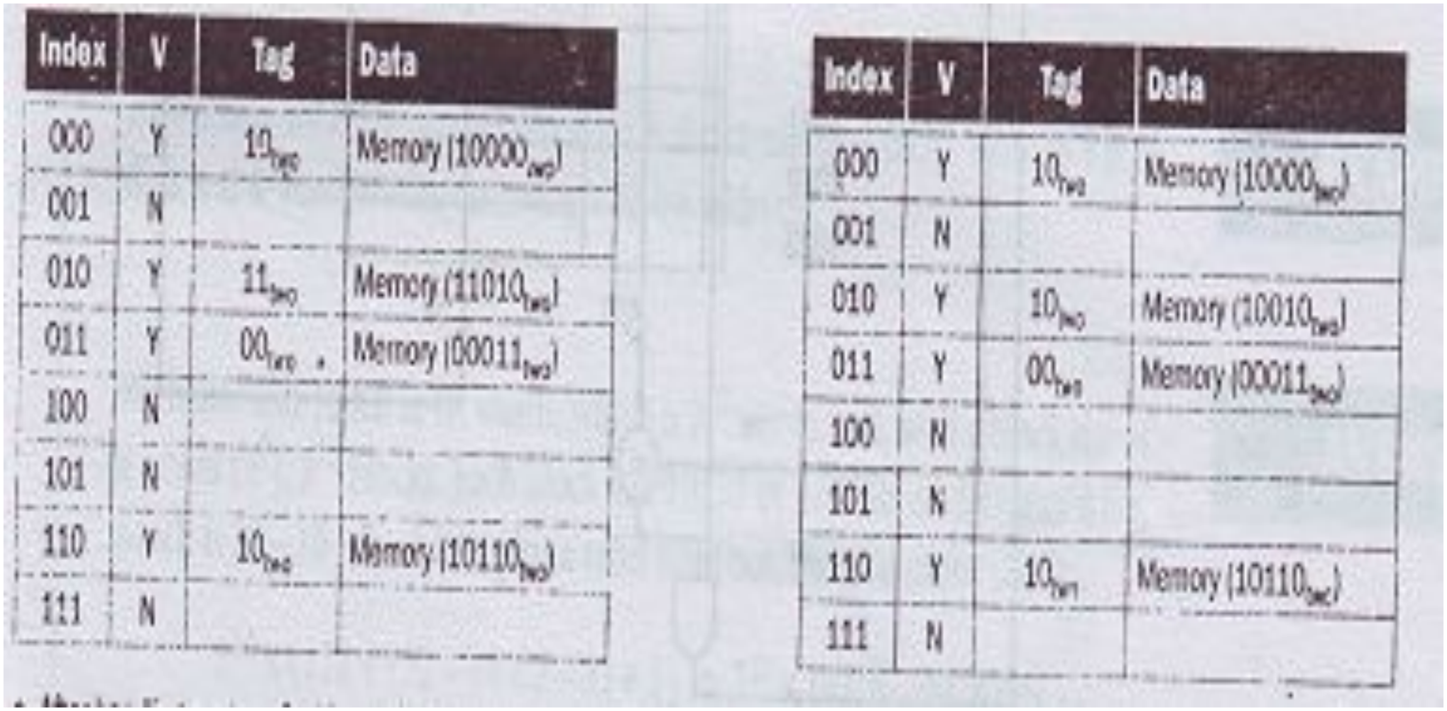
Index	V	Tag	Data
000	N		
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memorly (10110_{two})
111	N		

c. After handling a miss of address (11010_{two})

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

d. After handling a miss of address (10000_{two})

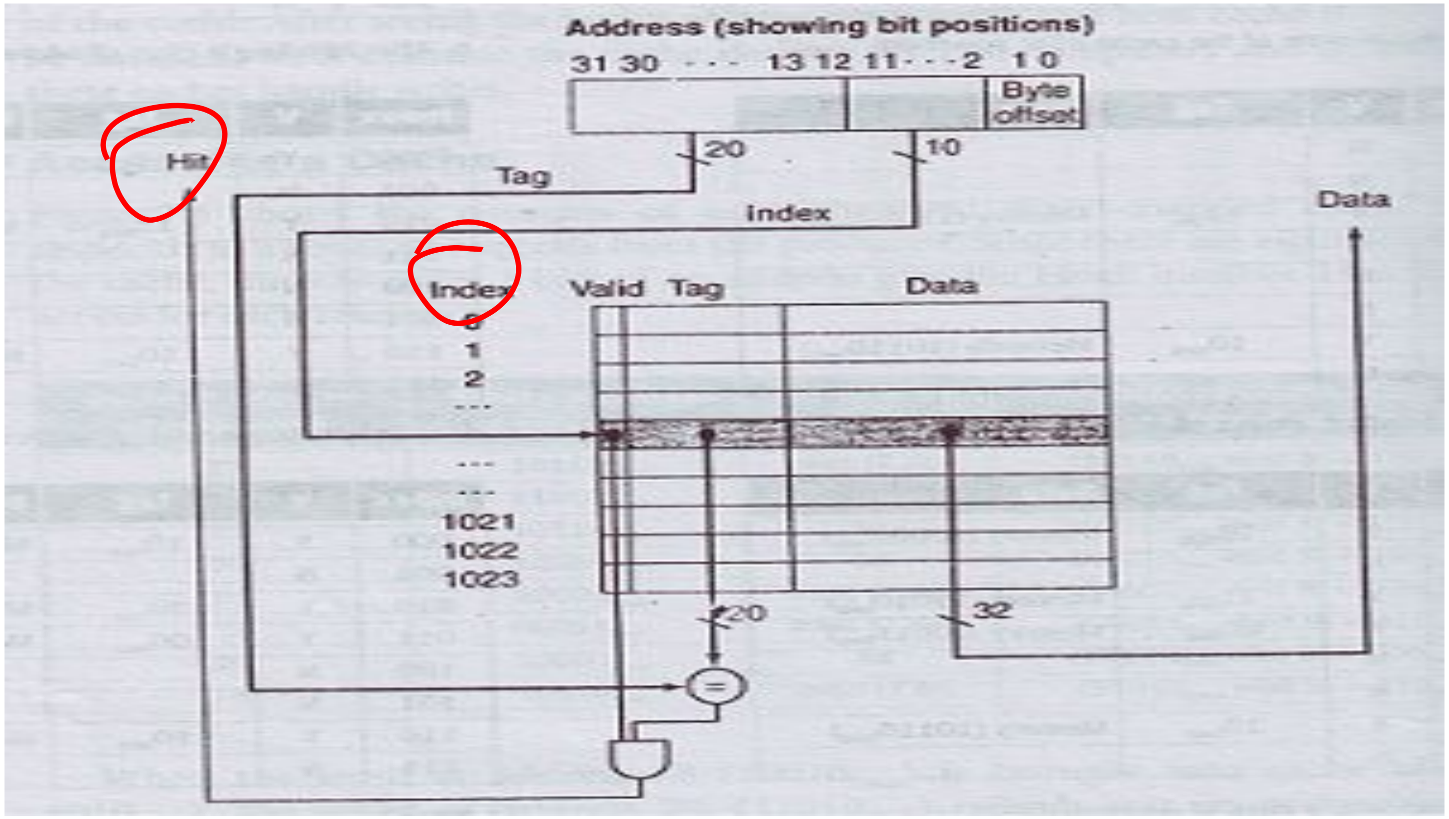
Accessing A Cache



Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	N		
010	Y	10_{two}	Memory (10010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	Y	10_{two}	Memory (10110_{two})
111	N		

Referencing a Cache Block



Cache Size

- ✔ The total number of bits needed for a cache is a function of the cache size and the address size.
- ✔ Let address = 32 bits

  Cache size = 2^n blocks with 2^m words.

  Tag size = 32 – (n+m+2)

10 = m

1024

~~1000 word~~

32 bi

2 wo

10

Bits in a Cache

16KB

D

4

✔ How many total bits are required for a direct-mapped cache with 16 KB of data and 4 word blocks, assuming a 32 bit address?

Data size = 16 KB = 4K words = 2^{12} words.

Block size = 4 words (2^2).

Cache Entries = 2^{10} blocks

Block size = 4 × 32 = 128 bits.

Tag = 32-10-2-2 = 18 bits.

Valid bit = 1 bit

Total Cache size = 2^{10} × (128+18+1) = 147Kbits = 18.4 KB

10 bit

4

32