Lecture - 15

16 February 2024 13:00

Q:- How is a block identified if present in the cache?

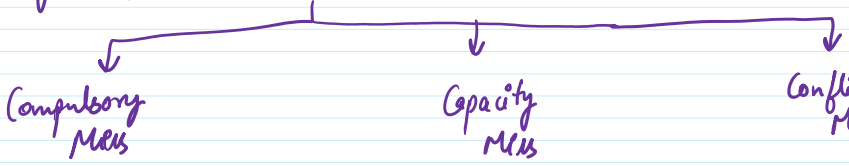
Caches include a TAG associated with each block.

- The TAGs of every cache block where the block being requested may be present needs to be compared with the TAG field of Main Memory Address.

How many comparisons?

1) Direct Mapping :- One Comparison
2) Associative Mapping :- Full associative search over all TAGs of cache blocks.
3) Set associative Mapping :- Limited associated search over TAGs (only the selected SET).

Types of Cache Misses :-



Compulsory Miss

→ Cold start miss / first referenced miss

→ on the first access of a block.

Capacity Miss

Cache cannot hold all the blocks needed.

blocks may be replaced.

Conflict Miss

In direct mapping / set mapping several blocks mapped to same block/ set in cache

Results in block replacement even though all the cache blocks may not be occupied.

Q: Which block should be replaced on a cache miss?

Direct Mapping :- trivial, no choice

Associative & Set Associative Mapping :-

there can be several blocks to choose from for replacement when a miss occurs.

- Random
- LRU (Least Recently Used)

Replace the block which has not been used for the longest period of time.

Uses temporal locality :-

"If recently used blocks are likely to be used again, then best candidate for replacement is the LRU block."

LRU Cache block replacement policy :-

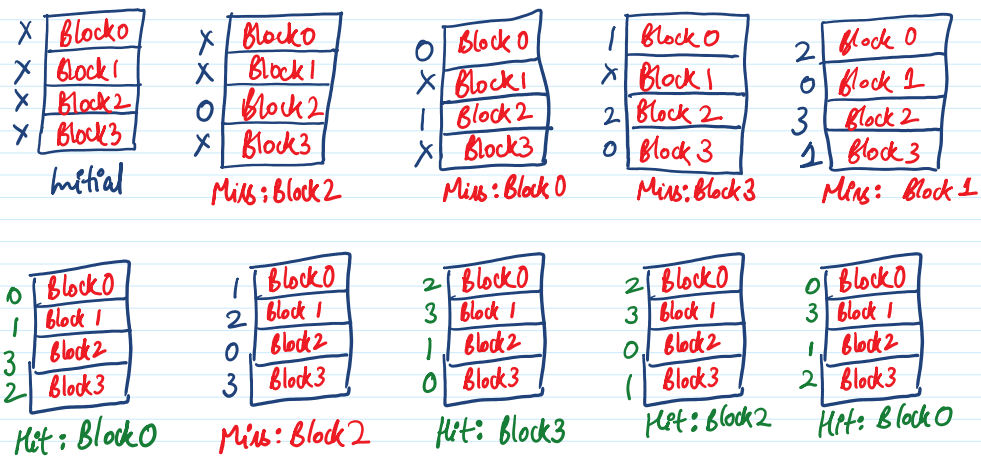
⇒ The cache controller tracks the LRU block using a counter.

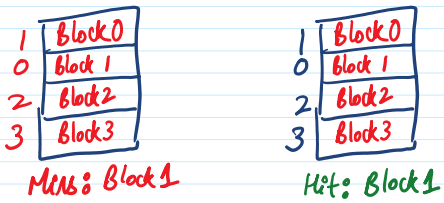
4-way set associative cache

⇒ 2-bit counter

- hit
  - Counter of referenced block is reset to 0.
  - Counters with values originally lower than referenced one are incremented by 1.
- miss
  - full
    - Counter associated with new block is set to 0
    - all other counters incremented by 1
  - not full
    - block with counter value 3 is removed
    - new block is placed in its place with counter 0
    - Other three counters are incremented by 1

Example :-





Improving Cache Performance :-

We know, for a two-level memory hierarchy, average memory access time,

$$T_{avg} = H_1 \times T_1 + (1 - H_1) \times T_2$$

- $H_1 \times T_1$ → Hit Time
- $(1 - H_1)$ → Miss Ratio
- T_2 → miss Penalty.

How can we improve the performance of cache memory?

Hit Time ↓ → avoiding the address translation when indexing the cache.

Miss Ratio ↓ → using larger block size, larger cache size, and higher associativity.

Miss Penalty ↓ → using multi-level caches.