

## Assignment - 5

Q.1. Describe different mappings func<sup>n</sup> used in cache memory organisation, such as direct mapping, associative mapping & set-associative mapping. Discuss the advantages & disadvantages of each mapping.

Ans. Different types of cache mapping are:-

1. Direct Mapping
  - a particular block of main memory can map only to a particular line of the cache.
  - the new incoming block will always replace the existing block (if any) in that particular line.

Advantages :-

1. Simplicity -
2. Low Hardware complexity.
3. Deterministic access time.
4. Cost effective.

Disadvantages

1. Limited associativity.
2. Higher conflict misses.
3. Uneven cache utilization.
4. Inflexibility.



## 2. Associative mapping.

- A block of main memory can map to any line of the cache that is freely available at that moment.
- If all the cache lines are occupied, then one of the existing blocks will have to be replaced.

### \* Advantages

1. Flexibility.
2. Low conflict misses.
3. Efficient cache utilization.

### \* Disadvantages

1. Higher hardware complexity.
2. Longer access time.
3. Higher cost.

## 3. Set-associative mapping

- Cache lines are grouped into sets where each set contains  $k$  no. of lines.
- A particular block of main memory can map to only one particular set of cache.
- However, within that set, the memory block can map any cache line that is freely available.

### \* Advantages

1. Increased flexibility.



2. Reduced conflict misses
3. Improved cache hit rate
4. Disadvantages
  1. Increased hardware complexity.
  2. Higher access time
  3. Limited Scalability.

Q-2 Explain the concept of cache replacement algorithms. Compare & contrast different replacement algorithms, such as FIFO, LRU & LFU in terms of <sup>their</sup> efficiency & complexity.

Ans. If a page is required for processing & that is not in main memory & the main memory is full, in that case page replacement is required.

1. FIFO (First In First Out)

- Efficiency - Simple to Implement
  - only requires tracking the order in which blocks are brought into cache.
  - not efficient in terms of cache hit.
- Complexity - low computational complexity
  - does not require any additional bookkeeping or tracking of access pattern



## 2. LRU (Least Recently Used):

- Efficiency - LRU aims to replace the least recently used block in the cache, assuming that recently used blocks are more likely to be accessed again.
- can be effective in capturing temporal locality & improving cache hit rates.
- Complexity - higher computational complexity
  - complexity increases with the cache size & efficiency of the implementation.

## 3. LFU (Least Frequently Used):

- Efficiency - replaces the least frequently used block in the cache.
  - aims to capture the frequency of block accesses.
  - can be effective in scenarios with varying access patterns & can lead to improve cache hit rates.
- Complexity - higher computational complexity compared to FIFO.
  - requires tracking the frequency of block accesses & updating the frequency counters.