CS 222 Computer Organization & Architecture

Lecture 27 [05.04.2019]

Cache Optimization Techniques

John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

Accessing Cache Memory

Average memory access time (AMAT) =
Hit time + (**Miss rate**×**Miss penalty**)

- **Hit Time:** Time to find the block in the cache and return it to processor [indexing, tag comparison, transfer].
- **Miss Rate:** Fraction of cache access result in a miss.
- **Miss Penalty:** Number of cycles required to fetch the block from the next level of memory hierarchy. It is the extra (not total) time (or cycle) for a miss in addition to hit time which is incurred by all accesses.

How to optimize cache ?

- Reduce Average Memory Access Time
- **AMAT= Hit Time + Miss Rate x Miss Penalty**
- Motives
  - **Reducing the miss rate**
  - **Reducing the miss penalty**
  - **Reducing the hit time**

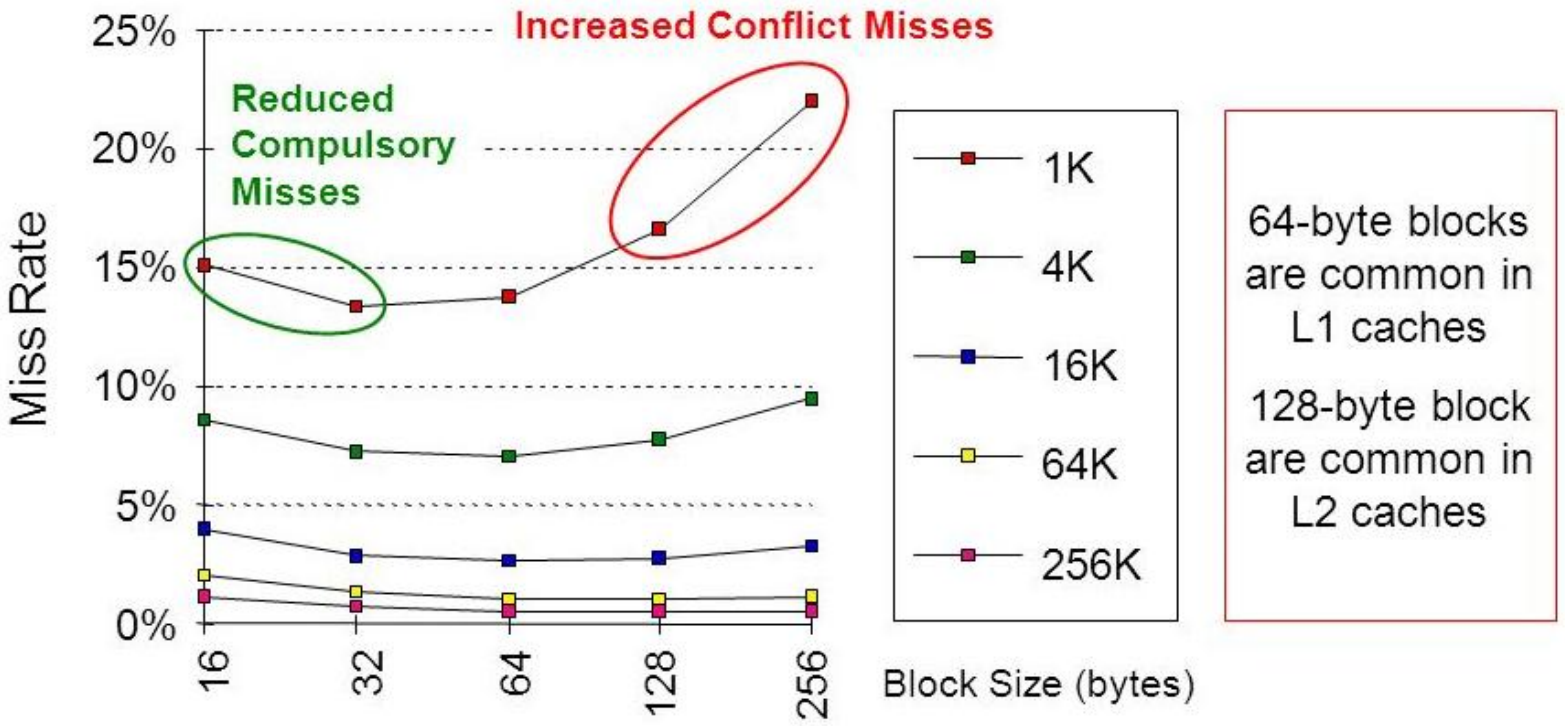
Types of Cache Misses

- **Compulsory**
  - Very first access to a block
  - Will occur even in an infinite cache
- **Capacity**
  - If cache cannot contain all the blocks needed
  - Misses in fully associative cache (due to the capacity)
- **Conflict**
  - If too many blocks map to the same set
  - Occurs in associative or direct mapped cache

Larger Block Size

- **Larger block size to reduce miss rate**
- **Advantages**
  - Utilize spatial locality
  - Reduces compulsory misses
- **Disadvantages**
  - Increases miss penalty
  - More time to fetch a block to the cache [bus width issue]
  - Increases conflict misses
  - More number of blocks mapped to the same location
  - Pollution: Bring useless data and evict useful data

Larger Block Size

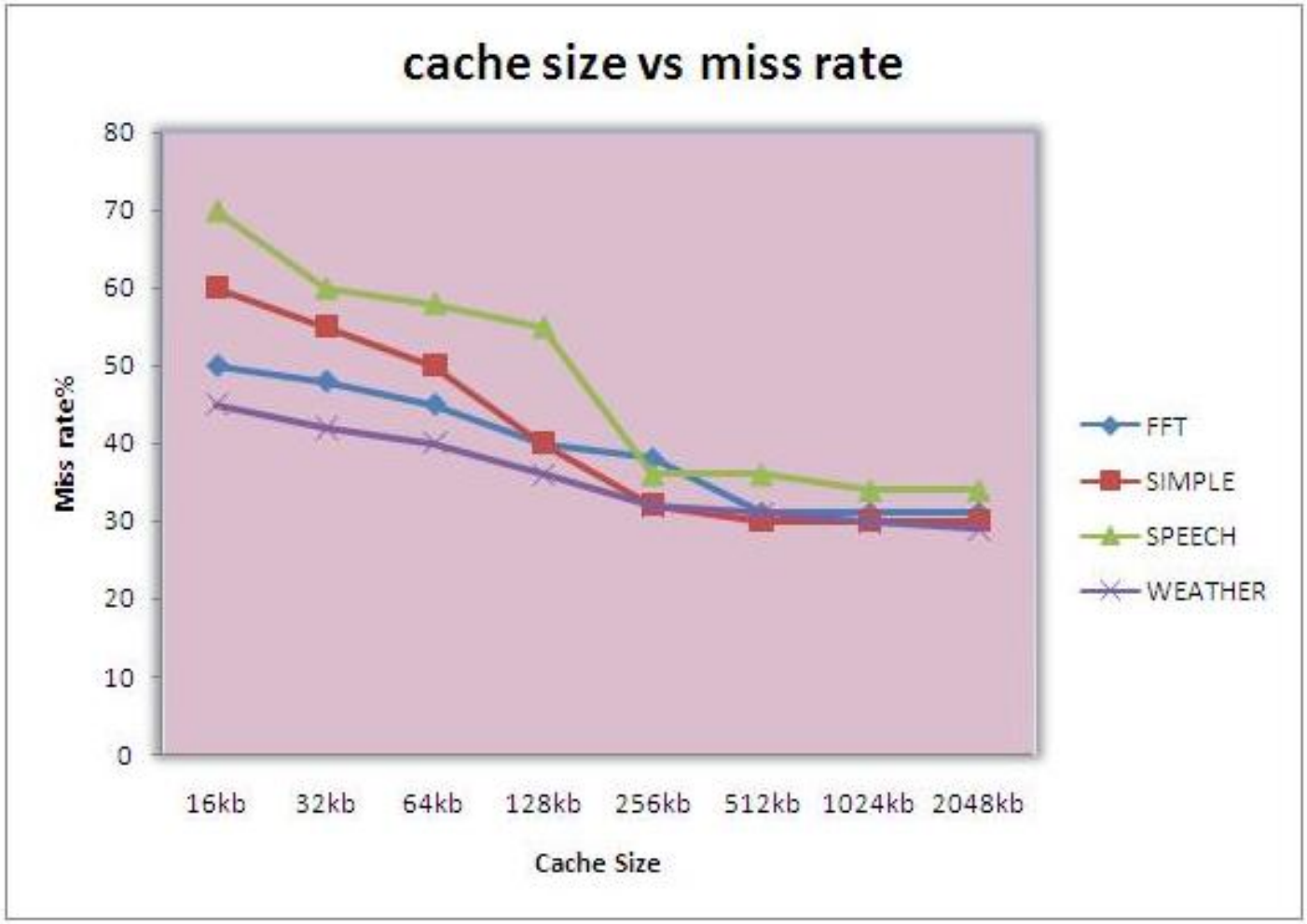


Larger Caches

- **Larger cache to reduce miss rate**
- **Advantages**
  - Reduces capacity misses
  - Can accommodate larger memory footprint
- **Drawbacks**
  - Longer hit time
  - Higher cost, area and power

	Cache size			
Block size	4K	16K	64K	256K
16	8.57%	3.94%	2.04%	1.09%
32	7.24%	2.87%	1.35%	0.70%
64	7.00%	2.64%	1.06%	0.51%
128	7.78%	2.77%	1.02%	0.49%
256	9.51%	3.29%	1.15%	0.49%

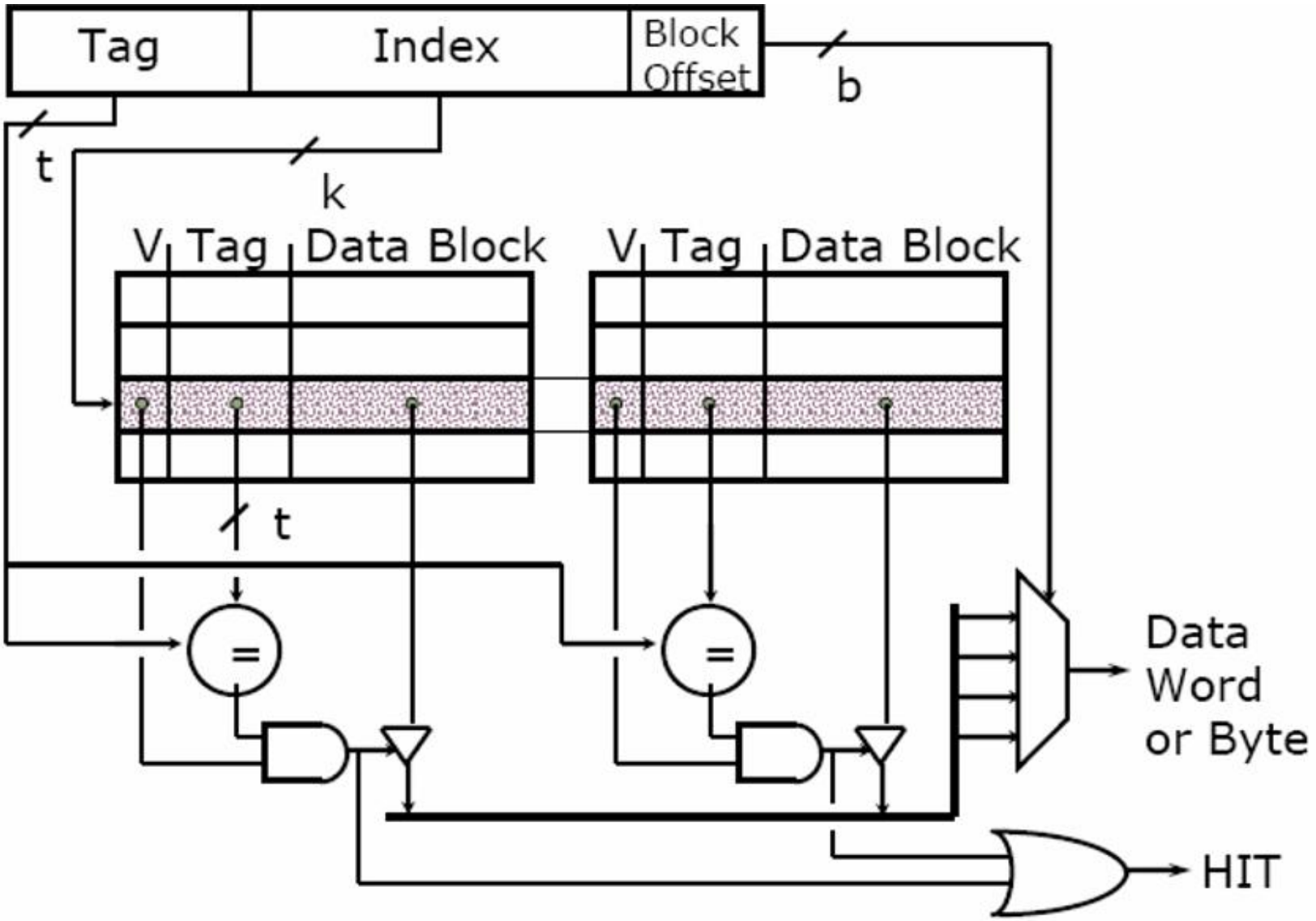
Larger Caches



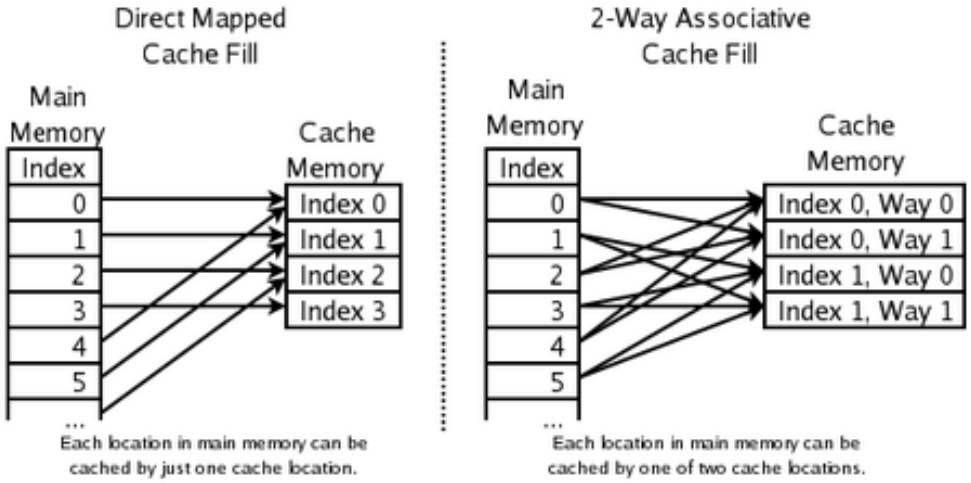
Higher Associativity

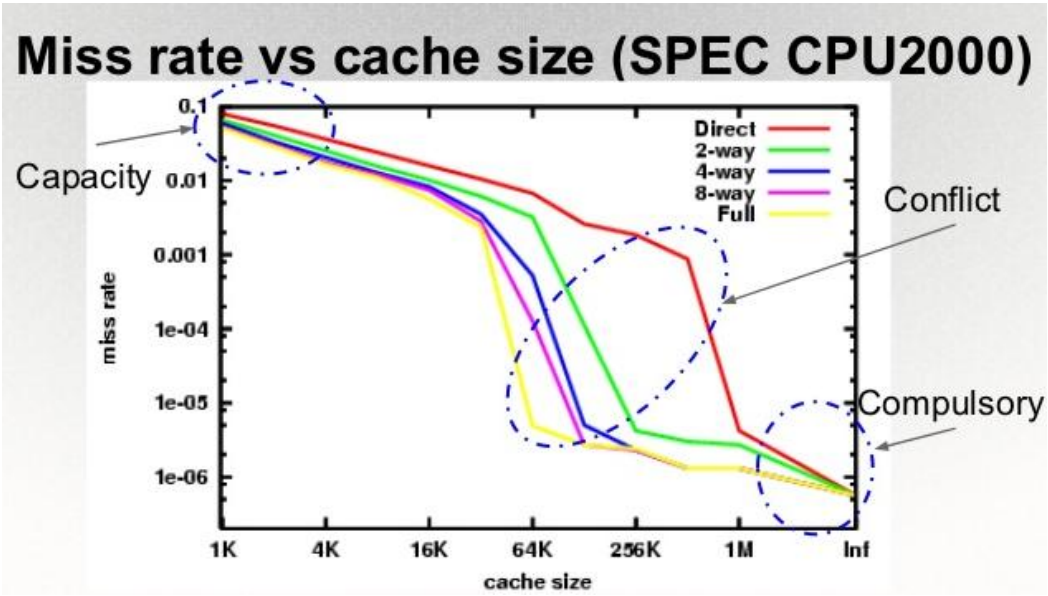
- **Higher associativity to reduce miss rate**
  - Fully associative caches are the best; high hit time.
  - So increase the associativity to the possible level
- **Advantages**
  - Reduce conflict miss
  - Reduce miss rate and eviction rate
- **Drawbacks**
  - Increase in the hit time
  - Complex design than direct mapped
  - More time to search in the set (tag comparison time)

Indexing, Tag comparison, Transfer



AMAT vs cache associativity

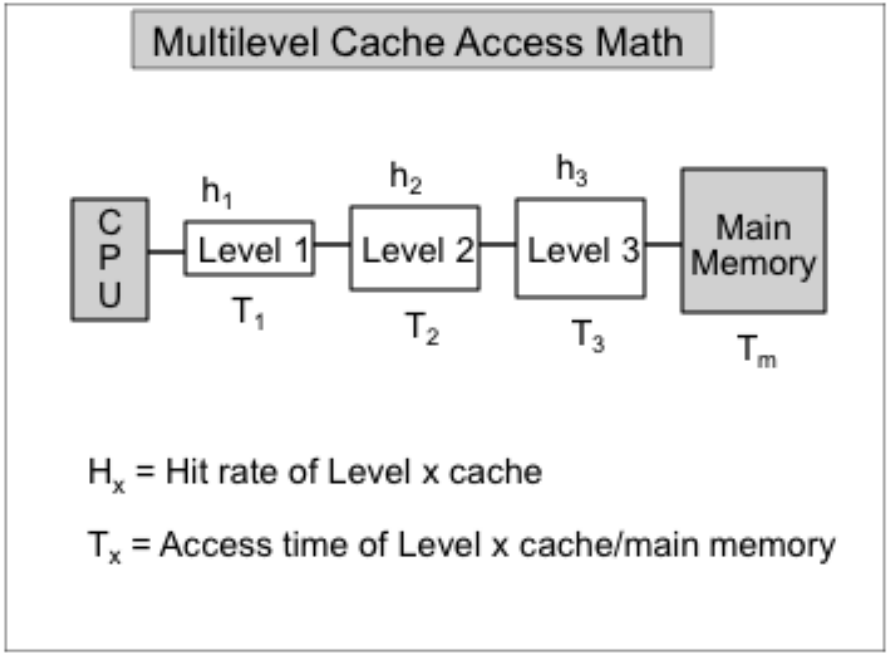




Multilevel caches

- **Multilevel caches to reduce miss penalty**
- **Caches should be faster** to keep pace with the speed of processors, **AND cache should be larger** to overcome the widening gap between the processor and main memory
- Multiple levels of cache between processor and memory.
- The L1 cache should be small enough to match the clock cycle time of the fast processor. [Low hit time]
- The L2 cache should be large enough to capture many accesses that would go to main memory, thereby lessening the effective miss penalty. [Low miss rate]

Multilevel caches



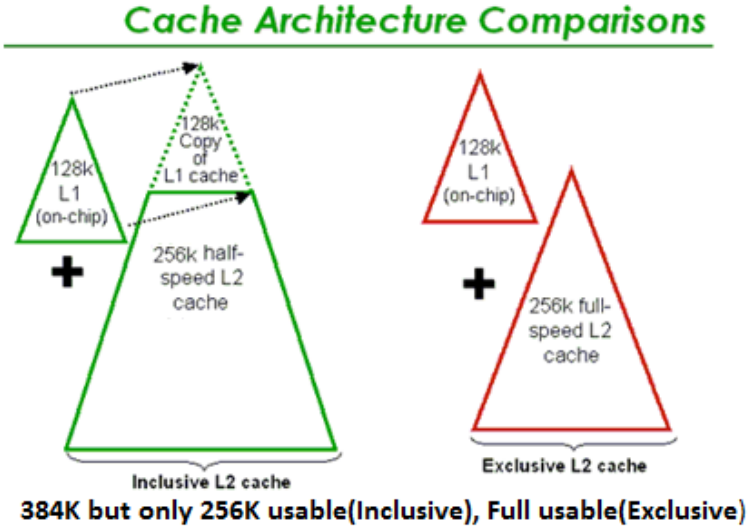
$$\text{Average memory access time} = \text{Hit time}_{L1} + \text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}$$

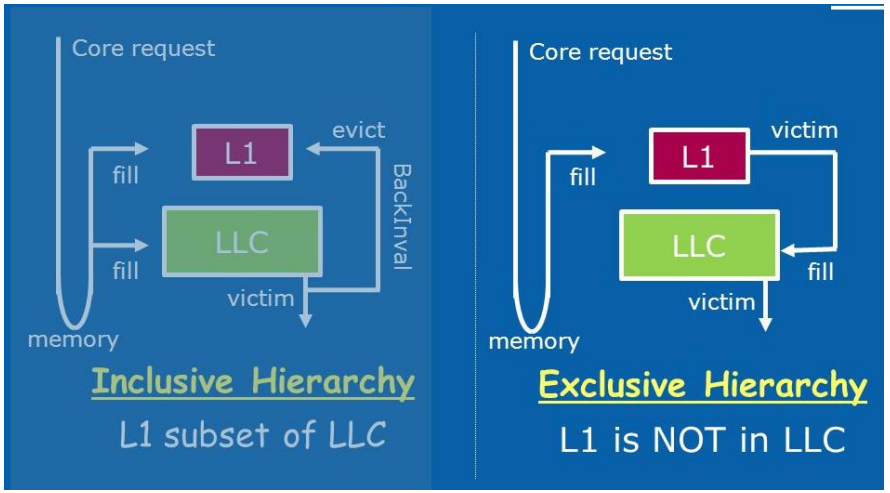
$$\text{Miss penalty}_{L1} = \text{Hit time}_{L2} + \text{Miss rate}_{L2} \times \text{Miss penalty}_{L2}$$

$$\text{Average memory access time} = \text{Hit time}_{L1} + \text{Miss rate}_{L1} \times (\text{Hit time}_{L2} + \text{Miss rate}_{L2} \times \text{Miss penalty}_{L2})$$

Multilevel caches

- **Multilevel caches to reduce miss penalty**
- **Local miss rate:** Number of misses in a cache level divided by number of memory access to this level.
- **Global miss rate:** Number of misses in a cache level divided by number of memory access generated by the CPU.
- **Inclusive and Exclusive caches**





johnjose@iitg.ac.in
http://www.iitg.ac.in/johnjose/