University of Central Florida

Department of Computer Science

CDA 5106: Fall 2023

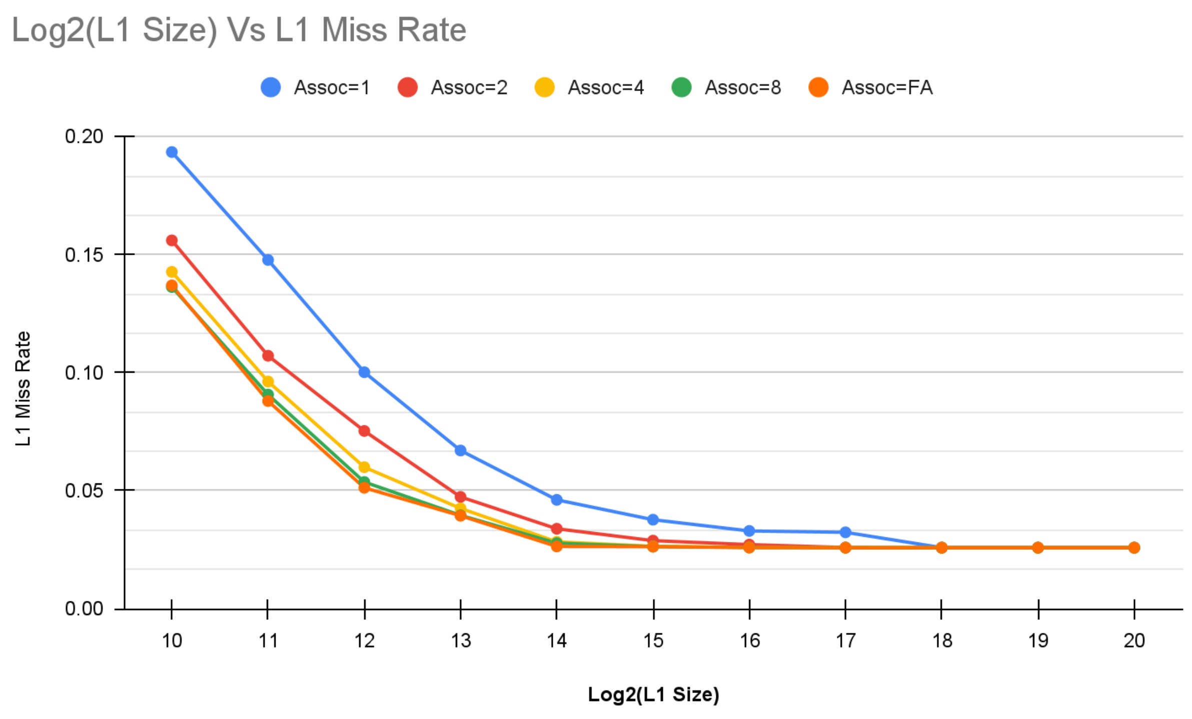
Machine Problem 1: Cache Design, Memory Hierarchy Design

by

<< SAI SRIRAM AMPAPURAPU >>

Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."
Student's electronic signature: ___Sai Sriam Ampapurpau______
(sign by typing your name)

Graph #1



1. Discuss trends in the graph. For a given associativity, how does increasing cache size affect miss rate? For a given cache size, what is the effect of increasing associativity?

a. *Effect of Increasing Cache Size on Miss Rate*: As the cache size increases (moving from left to right in each associativity category), the miss rate generally decreases. This trend is consistent across all associativity levels. Larger cache sizes lead to better spatial locality, reducing the number of cache misses.

b. *Effect of Increasing Associativity for a Given Cache Size*: For a given cache size, increasing associativity generally reduces miss rates. Higher associativity allows for more flexibility in mapping blocks, reducing conflict misses.

2. Estimate the compulsory miss rate from the graph.

The compulsory miss rate (CMR) is associated with first-time accesses, meaning the misses that occur when a particular memory block is accessed for the first time and is not present in the cache. The CMR from the graph is 0.02582

3. For each associativity, estimate the conflict miss rate from the graph.

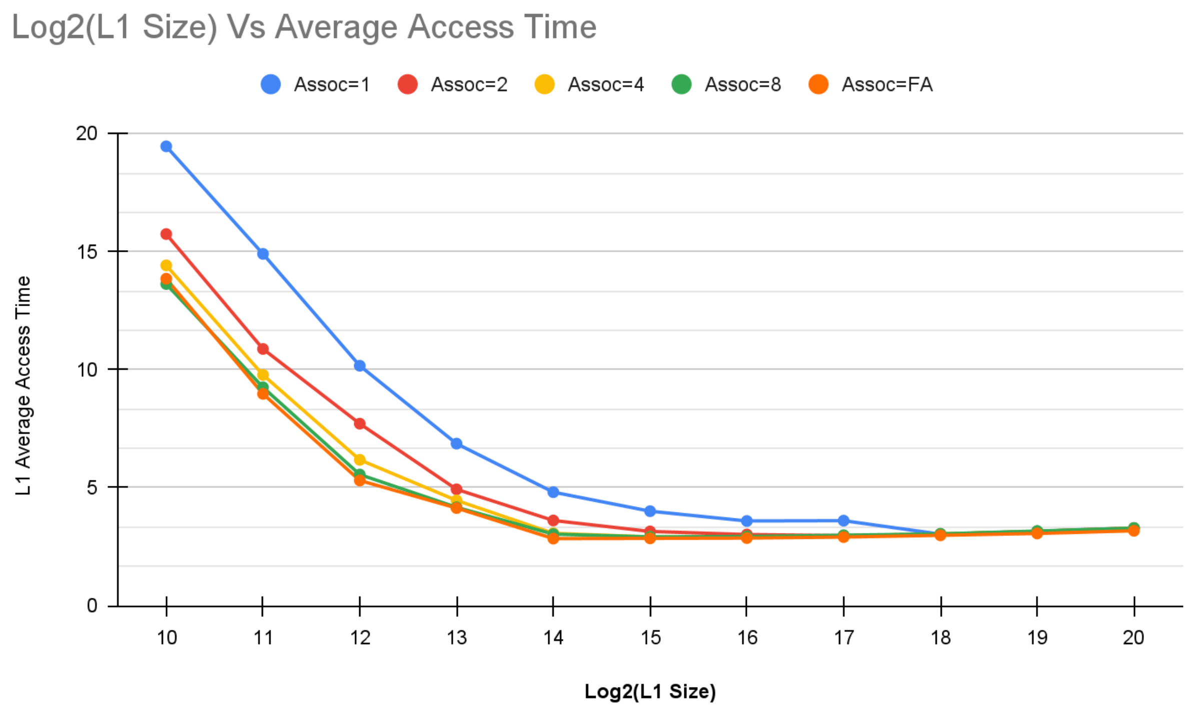
The conflict miss rate, also known as the collision miss rate, occurs in a cache when different memory blocks map to the same set or cache line. In other words, it arises due to conflicts in the cache mapping function, where multiple memory locations contend for the same cache location (set or line). When there are more memory blocks trying to be stored in the cache than there are available cache lines, conflicts occur, leading to cache misses.

Conflict Miss rate = Miss Rate (Assoc=1,2,4,8) - Miss Rate(Assoc=32)

The table below shows the conflict miss rate for each associativity and cache size

Cache_size	1Kb	2Kb	4Kb	8Kb	16Kb	32Kb	64Kb	128Kb	256Kb	512Kb	1Mb
1 way set	0.0565	0.05978	0.04897	0.02764	0.01975	0.01143	0.00706	0.00651	0.00002	0.00002	0.00002
2 way set	0.01907	0.01918	0.02408	0.00798	0.0075	0.00256	0.00127	0.00008	0.00002	0	0
4 way set	0.00574	0.00826	0.00872	0.00311	0.00198	0.00015	0.00009	0	0	0	0
8 way set	-0.00069	0.00273	0.00245	0.00018	0.0014	0	0.00003	0	0	0	0

Graph #2



1. For a memory hierarchy with only an L1 cache and BLOCKSIZE = 32, which configuration yields the best (i.e., lowest) AAT?

The Access Time (AAT) for a memory hierarchy is influenced by the cache's miss rate and the time it takes to access the cache (hit time) or retrieve data from the next level in the hierarchy (miss penalty). The AAT is calculated as:

$$AAT = Hit\ Time + Miss\ Rate \times Miss\ Penalty$$

Considering only an L1 cache and assuming the miss penalty is constant across different configurations, the configuration that yields the lowest AAT is the one with the lowest miss rate.

In order to determine the configuration that yields the lowest Average Access Time (AAT), we need to identify the combination of cache size and associativity that results in the lowest miss rate. The AAT is influenced by both the hit time and miss rate, and a lower miss rate generally leads to a lower AAT.

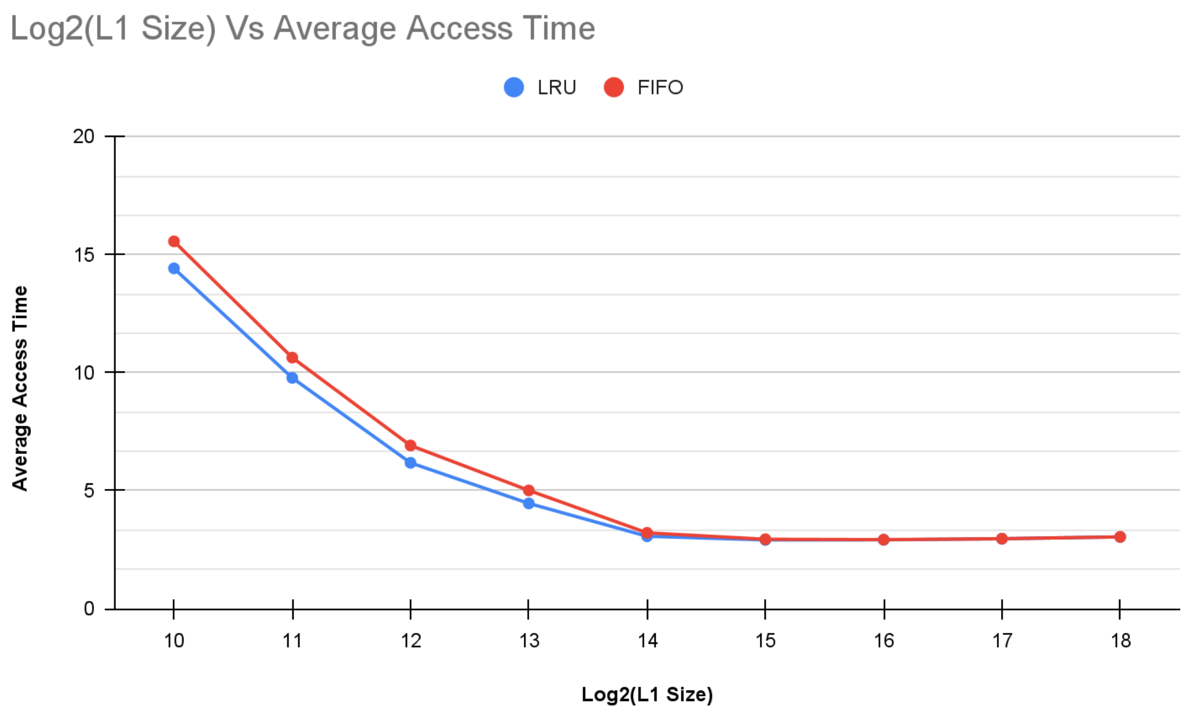
Let's find the configuration with the lowest miss rate for each cache size:

Log_2(L1 Size) = 10:	Assoc=8 has the lowest miss rate (13.627).
Log_2(L1 Size) = 11:	Assoc=32 has the lowest miss rate (8.972515).
Log_2(L1 Size) = 12:	Assoc=32 has the lowest miss rate (5.302948).
Log_2(L1 Size) = 13:	Assoc=32 has the lowest miss rate (4.134581).
Log_2(L1 Size) = 14:	Assoc=32 has the lowest miss rate (2.839608).
Log_2(L1 Size) = 15:	Assoc=32 has the lowest miss rate (2.84974).
Log_2(L1 Size) = 16:	Assoc=32 has the lowest miss rate (2.862281).
Log_2(L1 Size) = 17:	Assoc=32 has the lowest miss rate (2.904486).
Log_2(L1 Size) = 18:	Assoc=32 has the lowest miss rate (2.978009).
Log_2(L1 Size) = 19:	Assoc=32 has the lowest miss rate (3.057728).
Log_2(L1 Size) = 20:	Assoc=32 has the lowest miss rate (3.170474).

The configuration that yields the lowest AAT (based on the lowest miss rate) depends on the cache size. For cache size 1Kb, Assoc=8 provides the lowest miss rate. For cache sizes 2Kb to 1Mb, Assoc=32 provides the lowest miss rate. Therefore, the specific configuration for the lowest AAT would depend on the desired cache size.

In this case L1 size 8Kb and Fully Associative provides the lowest miss rate of 2.839608

Graph #3

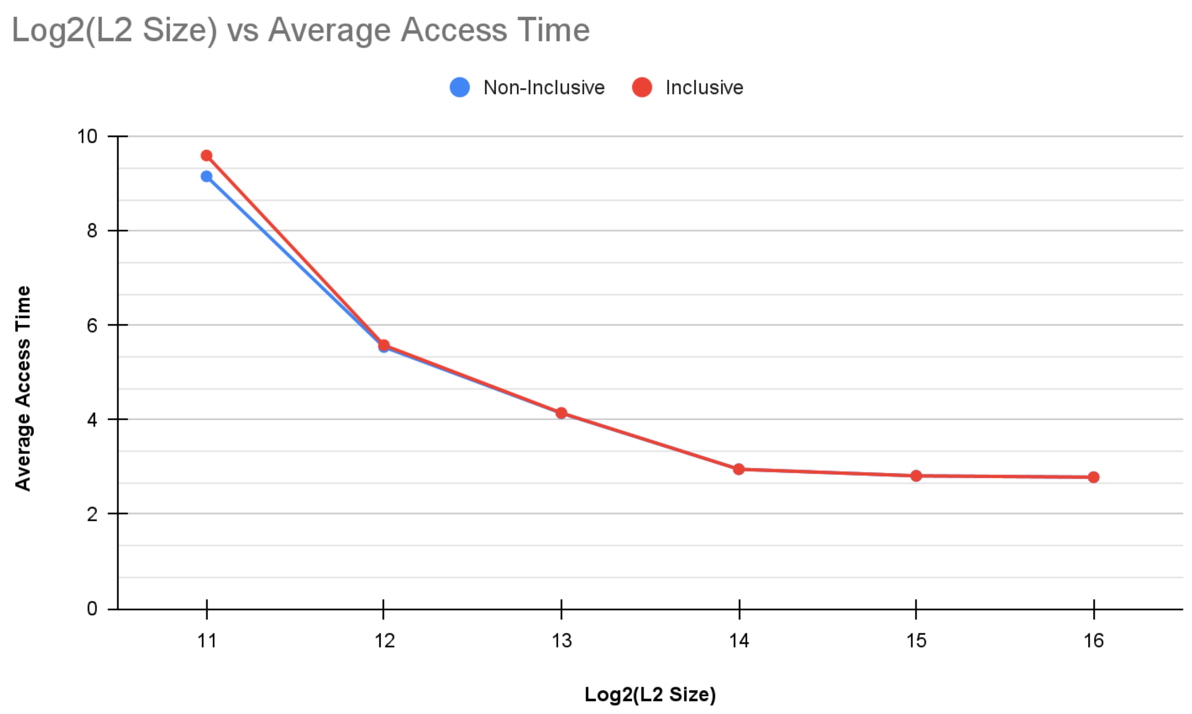


1. Discuss trends in the graph. Which replacement policy yields the best (i.e., lowest) AAT?

As the L1 cache size increases, both LRU AAT and FIFO AAT generally decrease. This is expected, as larger cache sizes typically lead to better cache hit rates, resulting in a lower AAT. In most cases, LRU AAT is slightly lower than FIFO AAT. This suggests that the Least Recently Used (LRU) replacement policy tends to perform better than First-In-First-Out (FIFO) in terms of Average Access Time (AAT) for this dataset.

The Least Recently Used (LRU) replacement policy generally yields the best (lowest) AAT across various L1 cache sizes. This conclusion is drawn from comparing the LRU AAT and FIFO AAT values for each cache size. It's important to note that specific performance may vary based on the characteristics of the workload and access patterns.

Graph #4



1. Discuss trends in the graph. Which inclusion property yields a better (i.e., lower) AAT?

As the L2 cache size increases, both AAT (non-inclusive) and AAT (inclusive) generally decrease. This is expected, as larger cache sizes tend to result in better cache hit rates, reducing the average access time.

In all cases, AAT (inclusive) is slightly higher than AAT (non-inclusive). This suggests that the non-inclusive property tends to yield a better (lower) Average Access Time (AAT) compared to the inclusive property.

The non-inclusive inclusion property generally yields a better (lower) Average Access Time (AAT) across various L2 cache sizes. This conclusion is drawn from comparing the AAT (non-inclusive) and AAT (inclusive) values for each cache size.