University of Central Florida

Department of Computer Science

CDA 5106: Fall 2023

Machine Problem 1: Cache Design, Memory Hierarchy Design

by

Tanmay Arora

Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

Student's Electronic Signature: Tanmay Arora

8.1 L1 cache exploration: SIZE and ASSOCIATIVITY

GRAPH #1

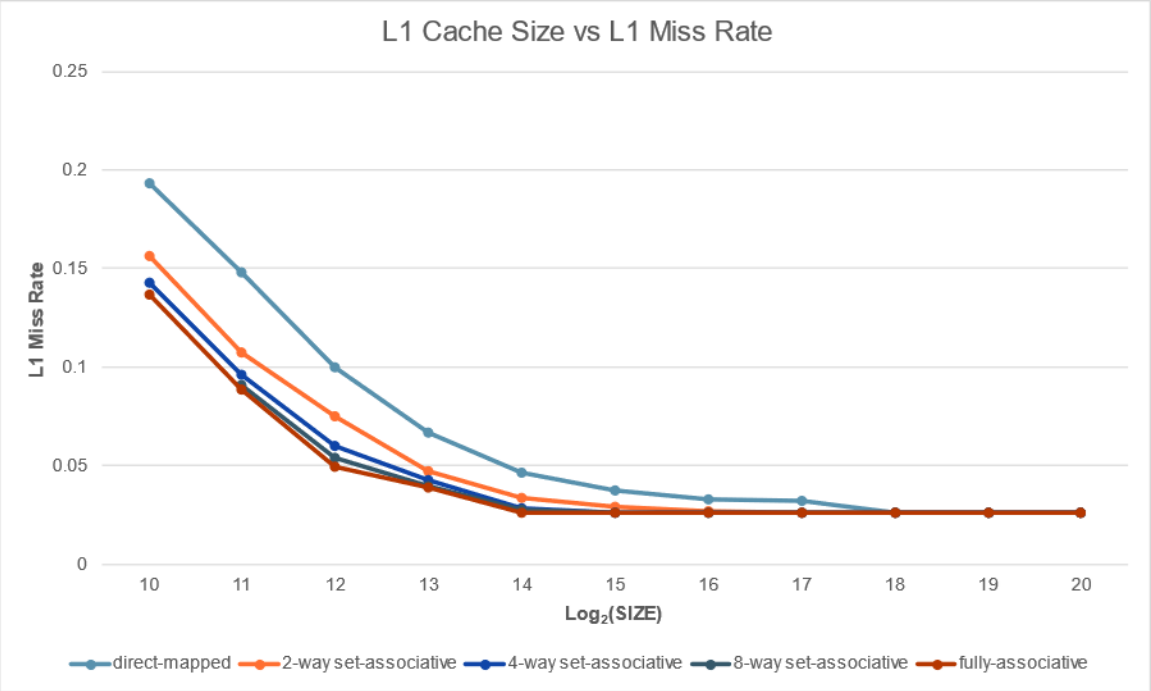


Figure 1

1. **Trends from Graph #1:**
   As we increase the cache size, a consistent downward trend is observed across all the associativity. The miss rate shows a continuous decline with each increment in size, exhibiting diminishing returns overall. Similarly, elevating the associativity also results in a decrease in the miss rate. On Observing closely, the most substantial improvements seem to occur when the L1 cache transitions from a direct-mapped to a 2-way set-associative cache.

   a. **Increasing Cache Size affects miss Rate in this manner:**
   - As the cache size increases, the miss rate decreases for a given associativity.
   - Higher cache sizes result in a reduction in compulsory misses.
   - The effect of increasing associativity is more noticeable for smaller cache sizes. As the cache size increases, the impact of associativity diminishes.

   b. **Effect of increasing Associativity:**
   - For a given cache size, increasing associativity tends to decrease the miss rate.
   - This improvement of miss rate is more significant for smaller cache sizes.

2. **Estimation of Compulsory Miss Rate:**

Compulsory misses in the L1 cache can be estimated from observing the part of the graph where cache misses are inevitable.

Based on the outcomes, we can say the **compulsory miss rate** is around **0.02582**.

3. **Estimation of Conflict Miss Rate:**

Here, we will consider-
Cache size 2 KB or 2048 bytes ($Log_2(2*1024) = 11$) and Block Size is 32 bytes

Conflict Miss Rate is calculated from Difference of Miss rate of given associativity with Miss rate of Fully associative L1 Cache.

- **For Direct-mapped L1 Cache :**

  Conflict Miss Rate = 0.14774 – 0.0886
  = **0.05914**

- **For 2-way set-associative L1 Cache :**

  Conflict Miss Rate = 0.10714 – 0.0886
  = **0.1854**

- **For 4-way set-associative L1 Cache :**

  Conflict Miss Rate = 0.09622 – 0.0886
  = **0.00762**

- **For 8-way set-associative L1 Cache :**

  Conflict Miss Rate = 0.09069 – 0.0886
  = **0.00209**

- **For Fully associative L1 Cache:**

  Conflict Miss Rate = 0.0886 – 0.0886
  = **0**

GRAPH #2

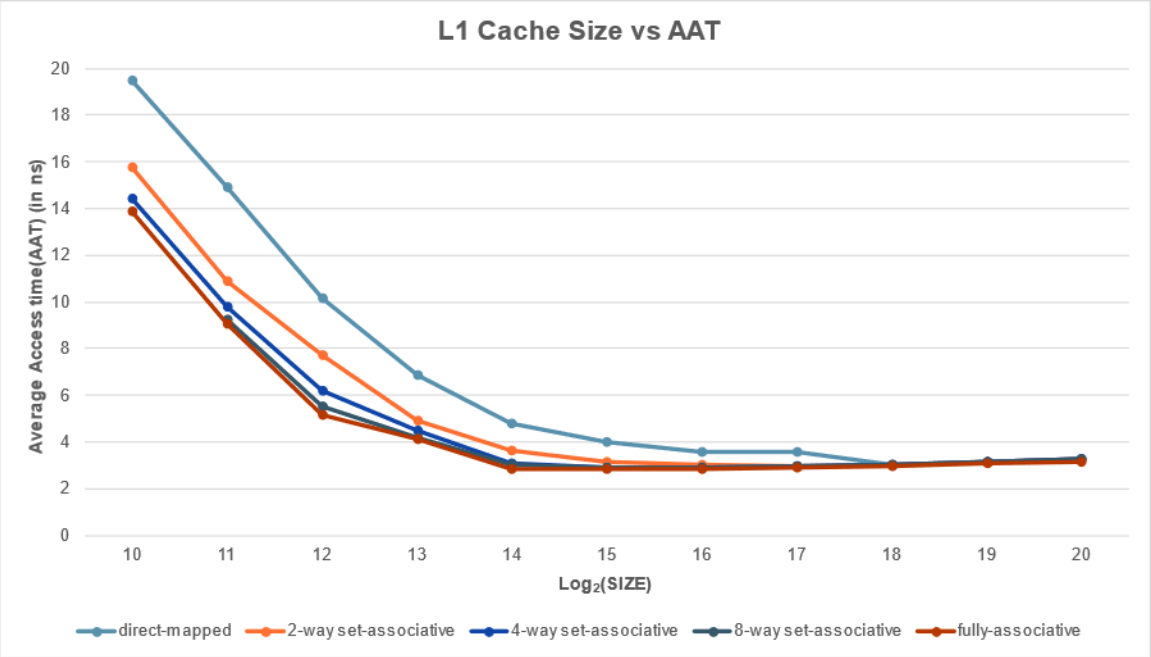
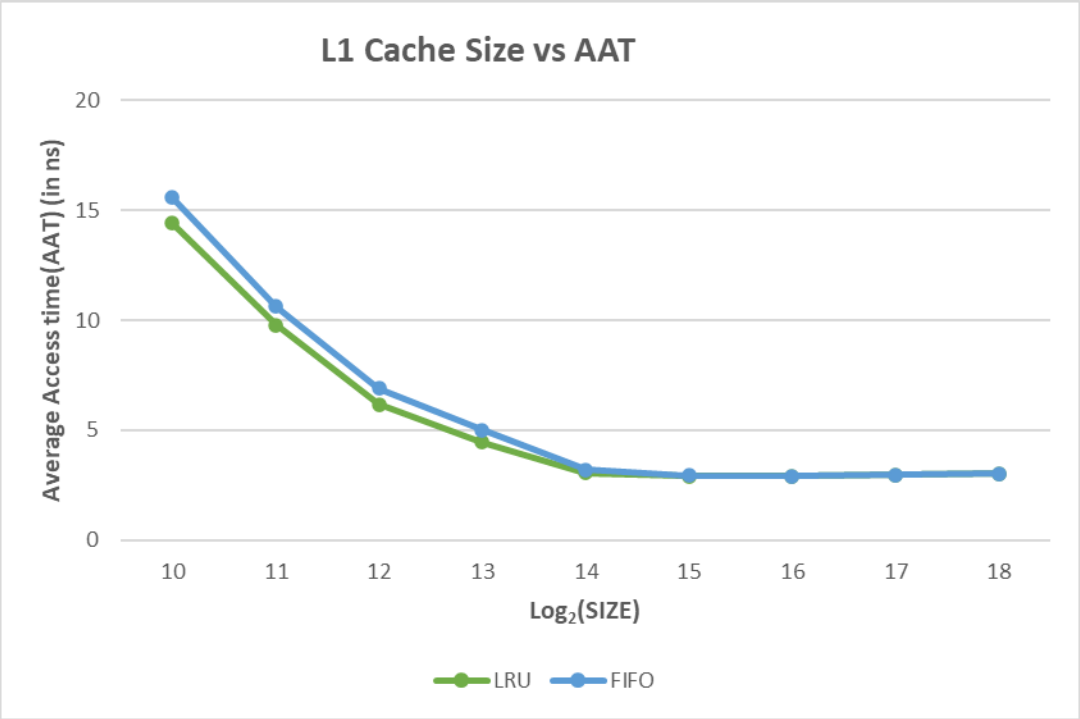


Figure 2

1. To find the configuration that yields the best (lowest) AAT (Average Access Time) for a memory hierarchy with only an L1 cache and BLOCKSIZE = 32, we need to examine the values for different cache configurations and associativities from the above graph.
   It is evident from the Graph #2 that increasing cache size reduces the AAT for all types of cache configuration. So we can say that the **fully associative cache configuration consistently has the lowest AAT** values across different cache sizes and associativities.
   Therefore, for a memory hierarchy with only an L1 cache and BLOCKSIZE = 32, the **fully associative configuration** is likely the best choice in terms of AAT.

8.2 Replacement Policy Study:

GRAPH #3



1. In analyzing the graph representing the Average Access Time (AAT) for different cache sizes and replacement policies, we observe the following trends:

   • **Decreasing AAT with Cache Size:**
   As the cache size increases, there is a noticeable decrease in the AAT for both LRU and FIFO replacement policies. Larger cache sizes tend to result in lower access times.

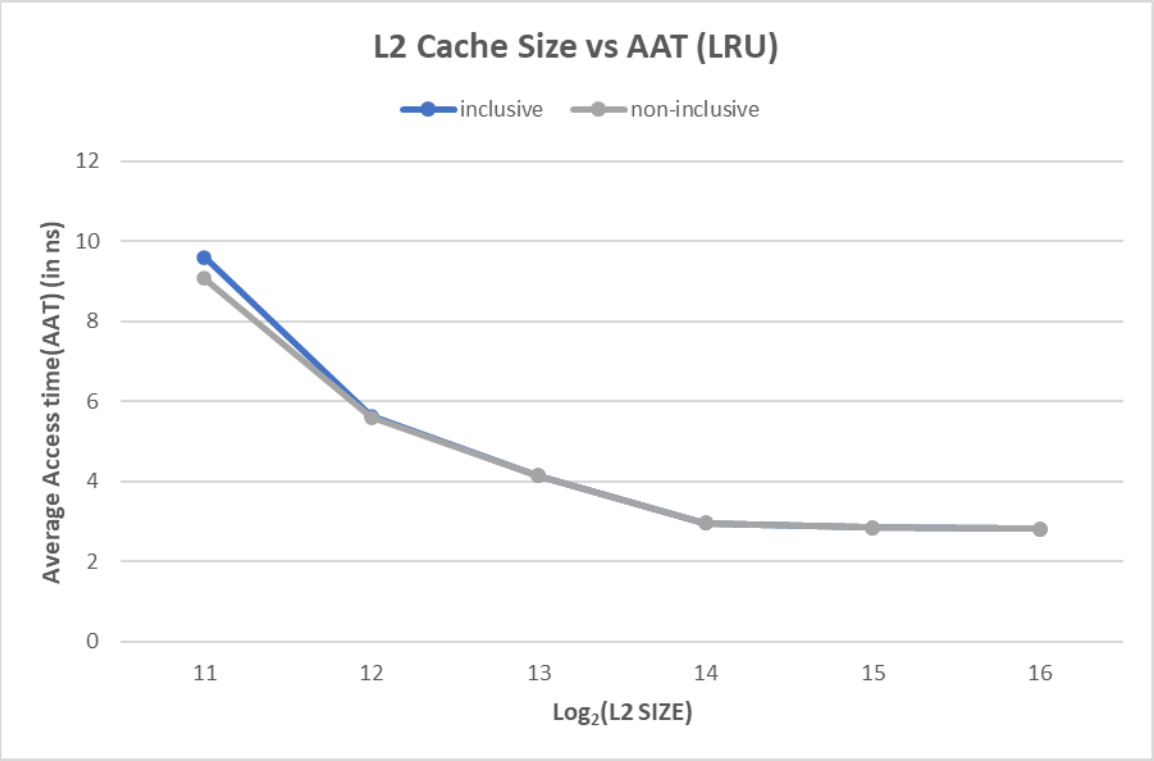
   • **Comparison between LRU and FIFO:**
   Across all the cache sizes, the LRU replacement policy consistently outperformed the FIFO replacement policy in terms of AAT. LRU exhibited lower AAT values compared to FIFO.

   • **Optimal Cache Size for LRU:**
   The optimal cache size for LRU, where the AAT is minimized, appears to be around 16KB based on the graph. After this point, there is diminishing returns in AAT reduction.

   In conclusion, for the given configuration, **the LRU replacement policy consistently results in lower Average Access Times(AAT) compared to FIFO.** The trends in the graph highlight the importance of choosing an appropriate replacement policy, and in this scenario, **LRU proves to be more efficient in optimizing cache performance.**

8.3 Inclusion Property Study:

GRAPH #4



1. **Trends from Graph #4:**

   In this graph of Inclusion property study, we aimed to contrast Average access time between inclusive and non-inclusive L2 cache policies using the LRU cache replacement policy. The L1 cache was configured with a size of 1 KB (1024 bytes), 4-way associativity, and a block size of 32. Meanwhile, the L2 cache was characterized by 9 varying sizes (2KB, 4KB, 8KB, ... 64KB), 8-way associativity, and a block size of 32.

   As indicated by the outcomes depicted in the graph, **the non-inclusive cache policy exhibits a slightly Better(lower) AAT compared to the inclusive cache policy.** Specifically, a marginal difference of 0.4325 ns was observed when the cache size was 2 KB. However, as the L2 cache sizes increased, the disparity diminished with each incremental size. By the time the cache size reached 16 KB, there was virtually no practical distinction in AAT between the two cache policies.