NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

Zachary Murray

NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this project."

Student's electronic signature: Zachary Murray

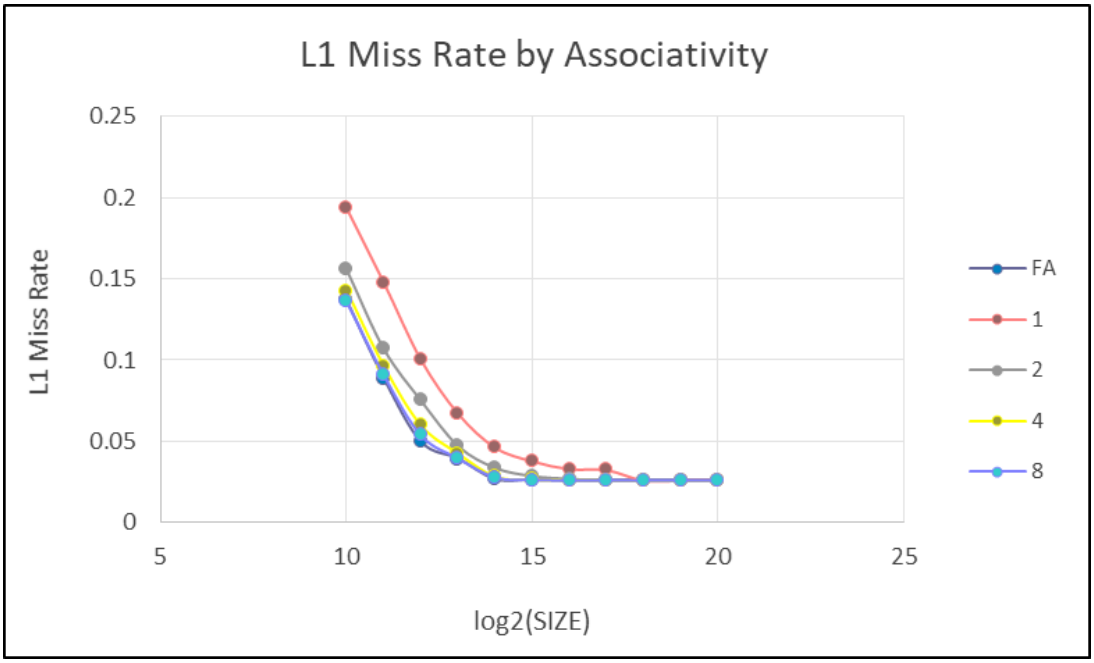
Course number: ECE 563

9.1. L1 cache exploration: SIZE and ASSOC

GRAPH #1 (*total number of simulations: 55*)

For this experiment:
• L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
• Victim Cache: None.
• L2 cache: None.

Plot L1 miss rate on the y-axis versus $\log_2$(SIZE) on the x-axis, for eleven different cache sizes: SIZE = 1KB, 2KB, … , 1MB, in powers-of-two. (That is, $\log_2$(SIZE) = 10, 11, …, 20.) The graph should contain five separate curves (*i.e.*, lines connecting points), one for each of the following associativities: direct-mapped, 2-way set-associative, 4-way set-associative, 8-way set-associative, and fully-associative. All points for direct-mapped caches should be connected with a line, all points for 2-way set-associative caches should be connected with a line, *etc*.



Discussion:
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?

Regardless of associativity, each curve approaches an asymptote at an L1 miss rate of roughly 0.025. The greater the associativity of the cache, the more benefit increasing the size has in terms of the L1 miss rate.

2. Estimate the *compulsory miss rate* from the graph.

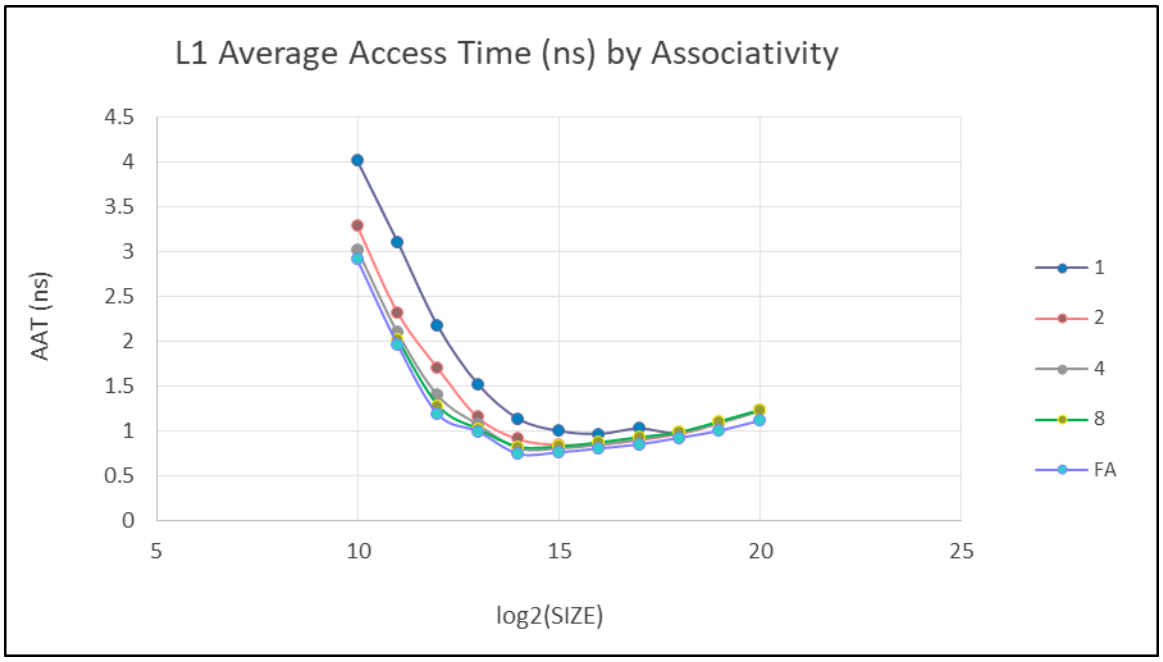
As noted above, the curves approach an asymptote at roughly 0.025. This asymptote can be used as an approximation of the compulsory miss rate, as an increase in both associativity and size doesn't lower the miss rate any further.

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

For the fully associative (FA) curve, there can be no conflict misses due to how the term is defined. In that sense, I estimate the conflict miss rates as the difference between the fully associative curve and the remaining curves.

Direct Mapped: Conflict miss rate is approximately 0.05.
2 way set-associative: Conflict miss rate is approximately 0.025.
4 way set-associative: Minimal conflict misses; conflict miss rate is approximately 0.01.
8 way set-associative: Minimal conflict misses; conflict miss rate is approximately 0.

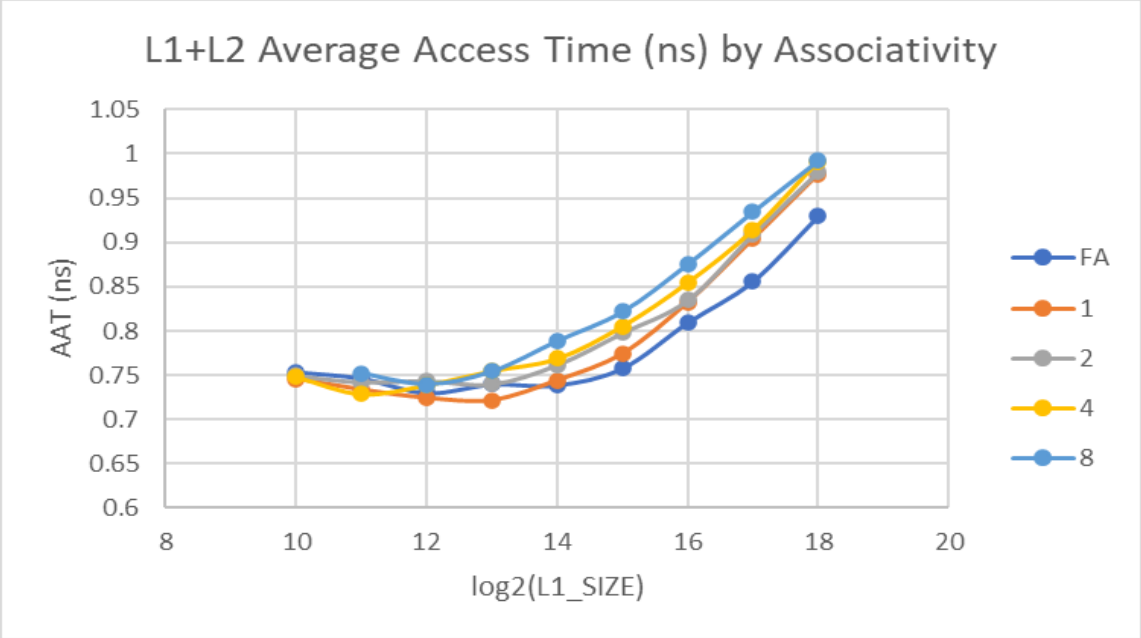
GRAPH #2 (*no additional simulations with respect to GRAPH #1*)
Same as GRAPH #1, but the y-axis should be AAT instead of L1 miss rate.



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

A fully associative L1 cache with cache size 16 KB yields the minimum Average Access Time (AAT). Based on the simulation results, the minimum access time using this configuration is 0.735042 ns.

GRAPH #3 (*total number of simulations: 45*)
Same as GRAPH #2, except make the following changes:
• Add the following L2 cache to the memory hierarchy: 512KB, 8-way set-associative, same block size as L1 cache.
• Vary the L1 cache size only between 1KB and 256KB (since L2 cache is 512KB).



Discussion:
1. With the L2 cache added to the system, which L1 cache configurations result in AATs close to the best AAT observed in GRAPH #2 (*e.g.*, within 5%)?

For all L1 cache configurations using a size of 8 KB or less, the AAT was within 5% of the minimum from Graph #2. For L1 configurations using a size of 16 KB, all having an associativity less than 8 way were within 5% of the minimum. The only other configuration that met this requirement was the 32 KB fully associative L1 cache configuration.

2. With the L2 cache added to the system, which L1 cache configuration yields the best (*i.e.*, lowest) AAT? How much lower is this optimal AAT compared to the optimal AAT in GRAPH #2?

The minimum AAT is generated by the direct-mapped L1 cache of size 8 KB. The minimum AAT is 0.721586. This value is 0.924% lower than the minimum AAT from Graph #2.

3. Compare the *total area* required for the optimal-AAT configurations with L2 cache (GRAPH #3) versus without L2 cache (GRAPH #2).

The configuration providing the minimum AAT with an L2 cache has a total area of 2.693 mm^2, and without an L2 cache the total area is 0.0634 mm^2. Although access time is decreased, the area is significantly increased.

9.2. L1 cache exploration: SIZE and BLOCKSIZE

GRAPH #4 (*total number of simulations: 24*)

For this experiment:

• L1 cache: SIZE is varied, BLOCKSIZE is varied, ASSOC = 4.
• Victim Cache: None.
• L2 cache: None.

Plot L1 miss rate on the y-axis versus $\log_2$(BLOCKSIZE) on the x-axis, for four different block sizes: BLOCKSIZE = 16, 32, 64, and 128. (That is, $\log_2$(BLOCKSIZE) = 4, 5, 6, and 7.) The graph should contain six separate curves (*i.e.*, lines connecting points), one for each of the following L1 cache sizes: SIZE = 1KB, 2KB, …, 32KB, in powers-of-two. All points for SIZE = 1KB should be connected with a line, all points for SIZE = 2KB should be connected with a line, *etc*.

Discussion:

1. Discuss trends in the graph. Do smaller caches prefer smaller or larger block sizes? Do larger caches prefer smaller or larger block sizes? Why? As block size is increased from 16 to 128, is the tradeoff between *exploiting more spatial locality* versus *increasing cache pollution* evident in the graph, and does the balance between these two factors shift with different cache sizes?

Smaller caches appear to favor smaller block sizes. Larger caches do appear to prefer larger block sizes, at least to the extent shown on this graph. The referenced tradeoff is definitely visible, as the 1 KB through 4 KB distinctly show a decline and then rise in miss rate, due to initially exploiting spatial locality and then eventually suffering from cache pollution. This

does shift as size increases, and if we continued to increase block size the 32 KB curve would eventually begin to rise as well due to cache pollution.

9.3. L1 + L2 co-exploration

GRAPH #5 (*total number of simulations: 44*)

For this experiment:

• L1 cache: SIZE is varied, BLOCKSIZE = 32, ASSOC = 4.
• Victim Cache: None.
• L2 cache: SIZE is varied, BLOCKSIZE = 32, ASSOC = 8.

Plot AAT on the y-axis versus $\log_2$(L1 SIZE) on the x-axis, for nine different L1 cache sizes: L1 SIZE = 1KB, 2KB, … , 256KB, in powers-of-two. (That is, $\log_2$(L1 SIZE) = 10, 11, …, 18.) The graph should contain six separate curves (*i.e.*, lines connecting points), one for each of the following L2 cache sizes: 32KB, 64KB, 128KB, 256KB, 512KB, 1MB. All points for the 32KB L2 cache should be connected with a line, all points for the 64KB L2 cache should be connected with a line, etc. *For a given curve / L2 cache size, only plot points for which the L1 cache is smaller than the L2 cache. For example, the curve for L2 SIZE = 32KB should have only 5 points total, corresponding to L1 SIZE = 1KB through 16KB. In contrast, the curve for L2 SIZE = 512KB (and 1MB) will have 9 points total, corresponding to L1 SIZE = 1KB through 256KB.*
There should be a total of 44 points (5+6+7+8+9+9, for the six curves / L2 cache sizes, respectively).

Discussion:

1. Which memory hierarchy configuration yields the best (*i.e.*, lowest) AAT?

A 2 KB L1 paired with a 64 KB L2 yields the lowest AAT, which is 0.7077 ns.

2. Which memory hierarchy configuration has the smallest total area, that yields an AAT within 5% of the best AAT?

A L1 cache size of 1 KB paired with a 32 KB L2 provides an AAT of 0.7158 ns, which is within 5% of the best AAT. This configuration provides the lowest area among the configurations within 5% of the lowest AAT, with an area of 0.2573 mm^2.

9.4. Victim cache study (ECE 563 students only)

GRAPH #6 (*total number of simulations: 42*)

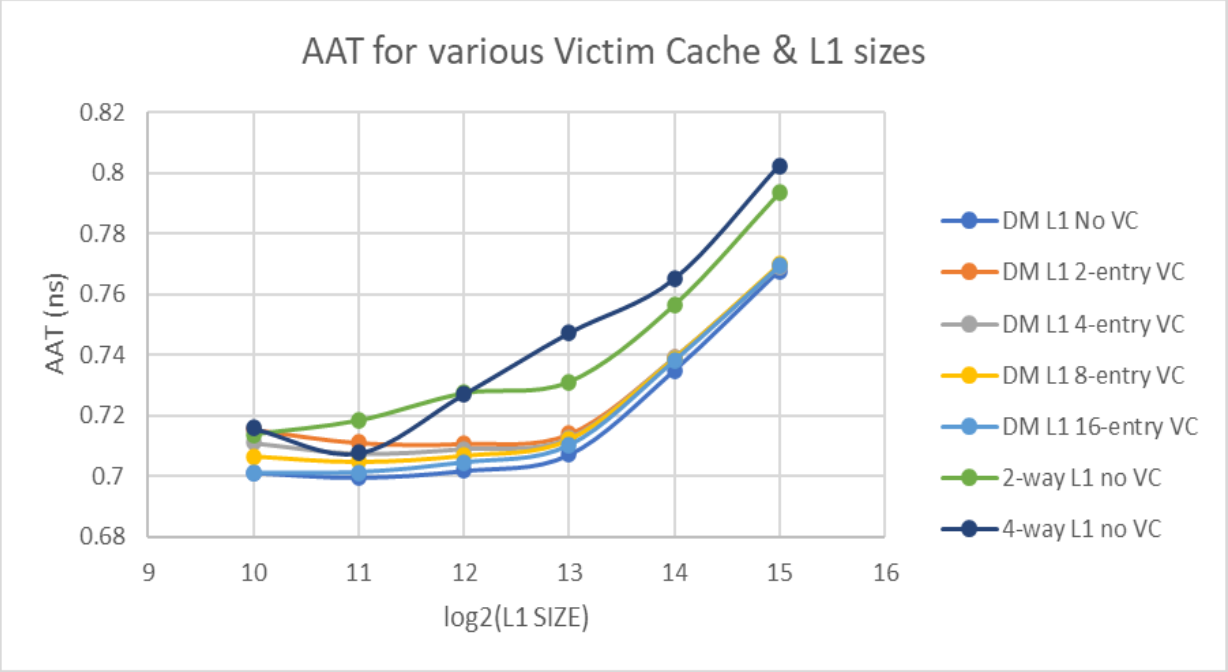
For this experiment:

• L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
• Victim Cache: # entries is varied.
• L2 cache: SIZE = 64KB, ASSOC = 8, BLOCKSIZE = 32.

Plot AAT on the y-axis versus $\log_2$(L1 SIZE) on the x-axis, for six different L1 cache sizes: L1 SIZE = 1KB, 2KB, … , 32KB, in powers-of-two. (That is, $\log_2$(L1 SIZE) = 10, 11, …, 15.) The graph should contain seven separate curves (*i.e.*, lines connecting points), one for each of the following scenarios:

• Direct-mapped L1 cache with no Victim Cache.
• Direct-mapped L1 cache with 2-entry Victim Cache.
• Direct-mapped L1 cache with 4-entry Victim Cache.
• Direct-mapped L1 cache with 8-entry Victim Cache.
• Direct-mapped L1 cache with 16-entry Victim Cache.
• 2-way set-associative L1 cache with no Victim Cache.
• 4-way set-associative L1 cache with no Victim Cache.

Discussion:



1. Discuss trends in the graph. Does adding a Victim Cache to a direct-mapped L1 cache yield performance comparable to a 2-way set-associative L1 cache of the same size? …for which L1 cache sizes? …for how many Victim Cache entries?

Adding a victim cache (all shown are considered) does increase the AAT of the cache, but much less so than adding more set associativity to the L1 cache without adding a victim cache. As shown in Graph #6, this is true across almost all of the configurations, with the one exception being that a 4-way set associative L1 cache of size 2 KB has a lower access time than the direct-mapped version with a 2-entry victim cache. As size increases, the AAT increases dramatically for each configuration. However, configurations utilizing a victim cache still performed better than the 2 and 4-way set associative L1 configurations with no victim cache.

2. Which memory hierarchy configuration yields the best (*i.e.*, lowest) AAT?

The direct mapped 2 KB L1 cache provided the lowest AAT of 0.6996 ns.

3. Which memory hierarchy configuration has the smallest total area, that yields an AAT within 5% of the best AAT?

The 2-way set associative 1 KB L1 cache provided the minimum total area and was within 5% of the lowest AAT. This configuration had an AAT of 0.7138 ns and a total area of 0.0095 mm^2.