# NC State University
# Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

## Project #1: Cache Design, Memory Hierarchy Design

by

Prajakta Keshavrao Jadhav
**pkjadhav@ncsu.edu, 200375352**

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

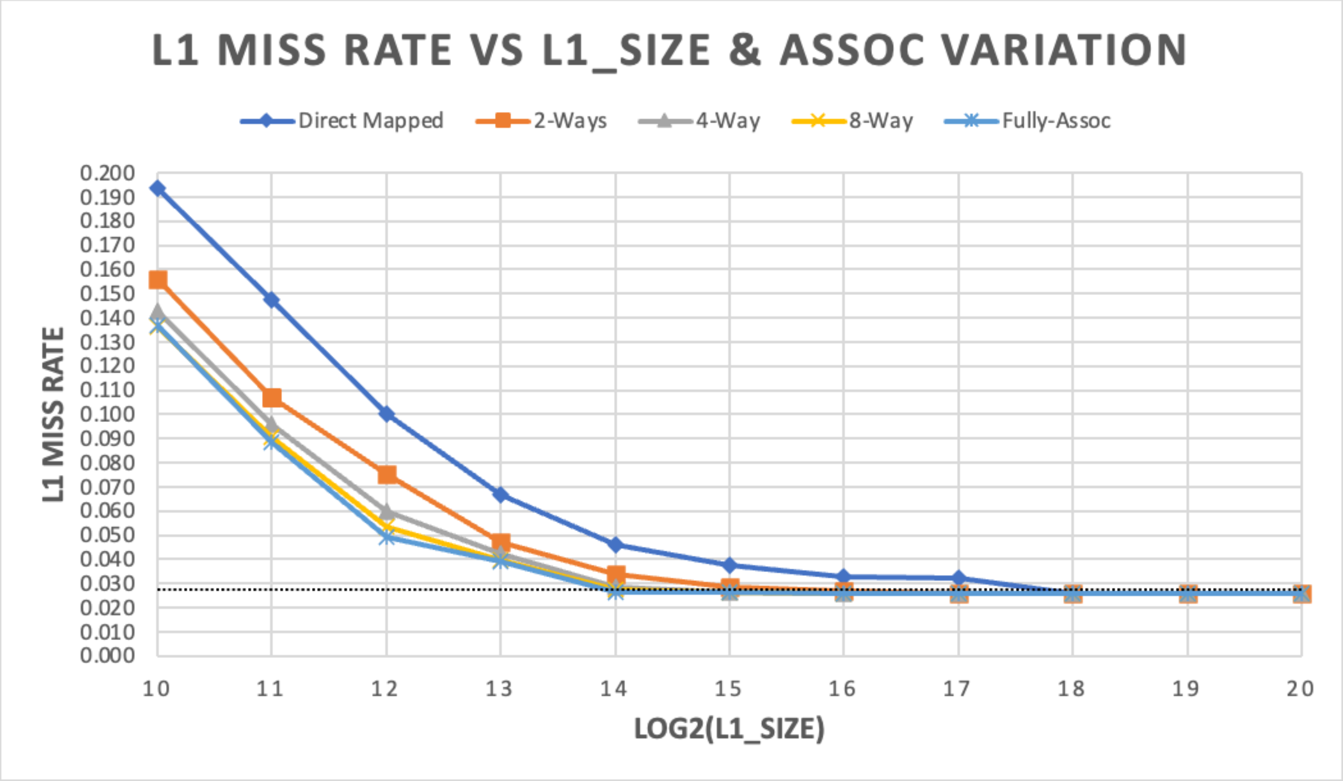
Student’s electronic signature:  Prajakta Keshavrao Jadhav

Course number: ECE563

# 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.



| L1_SIZE | L1 Miss Rate | | | | |
|---|---|---|---|---|---|
| | Direct Mapped | 2-Ways | 4-Way | 8-Way | Fully-Assoc |
| 1 KB | 0.1935 | 0.156 | 0.1427 | 0.1363 | 0.137 |
| 2 KB | 0.1477 | 0.1071 | 0.0962 | 0.0907 | 0.0886 |
| 4 KB | 0.1002 | 0.0753 | 0.0599 | 0.0536 | 0.0495 |
| 8 KB | 0.067 | 0.0473 | 0.0425 | 0.0395 | 0.0391 |
| 16 KB | 0.0461 | 0.0338 | 0.0283 | 0.0277 | 0.0263 |
| 32 KB | 0.0377 | 0.0288 | 0.0264 | 0.0262 | 0.0262 |
| 64 KB | 0.0329 | 0.0271 | 0.0259 | 0.0259 | 0.0258 |
| 128 KB | 0.0323 | 0.0259 | 0.0258 | 0.0258 | 0.0258 |
| 256 KB | 0.0258 | 0.0258 | 0.0258 | 0.0258 | 0.0258 |
| 512 KB | 0.0258 | 0.0258 | 0.0258 | 0.0258 | 0.0258 |
| 1 MB | 0.0258 | 0.0258 | 0.0258 | 0.0258 | 0.0258 |

**Discussion**:

Q1: 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?

→ It is observed that for any of given associativity if cache size is increased the L1 cache miss rate decreases for L1 sizes from 1KB to 256KB.

Similarly, when associativity is increased from 1way set associative to fully associative, miss rate decreases for L1 sizes from 1KB to 16KB. For sizes 32KB & onwards, associativity doesn't impact much on miss rate improvements. Rather, after 128KB fixed miss rate occurs even though associativity is improved.

Q2: Estimate the *compulsory miss rate* from the graph.

→ From graph#1 experiment, as shown in graph, 0.0258 will be the compulsory miss rate of L1 cache.

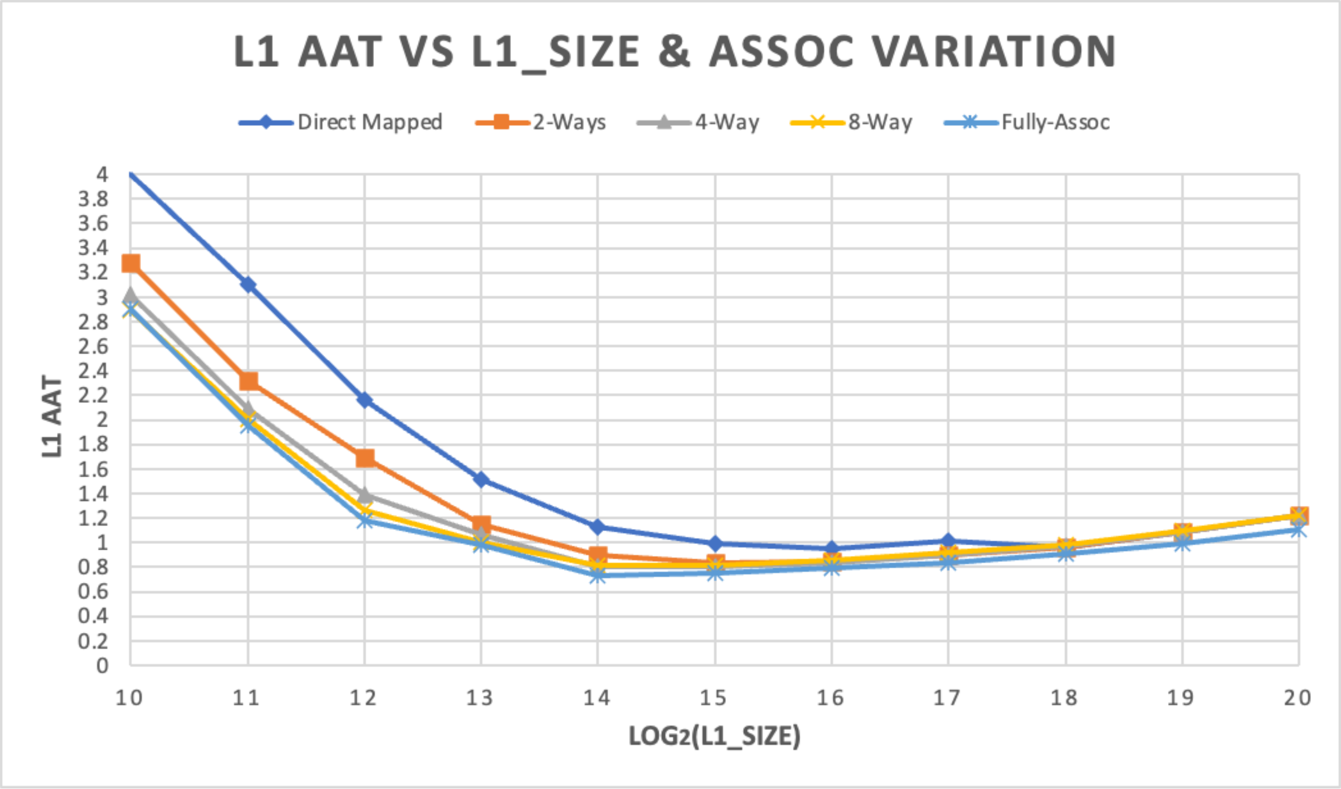
Q3: For each associativity, estimate the *conflict miss rate* from the graph.

→ The given experiment shows as associativity is varied; total miss rate improves but with fixed compulsory+capacity miss rate. Hence, each configuration will have different conflict miss rate = total miss rate – 0.0258

| L1 | L1 Miss Rate Comparison | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DM | | 2-Ways | | 4-Way | | 8-Way | | Fully-Assoc | |
| | Total | Conflict | Total | Conflict | Total | Conflict | Total | Conflict | Total | Conflict |
| 1 KB | 0.194 | 0.168 | 0.156 | 0.13 | 0.143 | 0.117 | 0.136 | 0.111 | 0.137 | 0.111 |
| 2 KB | 0.148 | 0.122 | 0.107 | 0.081 | 0.096 | 0.07 | 0.091 | 0.065 | 0.089 | 0.063 |
| 4 KB | 0.1 | 0.074 | 0.075 | 0.05 | 0.06 | 0.034 | 0.054 | 0.028 | 0.05 | 0.024 |
| 8 KB | 0.067 | 0.041 | 0.047 | 0.022 | 0.043 | 0.017 | 0.04 | 0.014 | 0.039 | 0.013 |
| 16 KB | 0.046 | 0.02 | 0.034 | 0.008 | 0.028 | 0.003 | 0.028 | 0.002 | 0.026 | 5E-04 |
| 32 KB | 0.038 | 0.012 | 0.029 | 0.003 | 0.026 | 6E-04 | 0.026 | 4E-04 | 0.026 | 4E-04 |
| 64 KB | 0.033 | 0.007 | 0.027 | 0.001 | 0.026 | 1E-04 | 0.026 | 1E-04 | 0.026 | 0 |
| 128 KB | 0.032 | 0.007 | 0.026 | 1E-04 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 |
| 256 KB | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 |
| 512 KB | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 |
| 1 MB | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 | 0.026 | 0 |

## GRAPH #2

Same as GRAPH #1, but the y-axis should be AAT instead of L1 miss rate.



| Cache Size in KB | | $AAT = HT_{L1} + MR_{L1} \cdot MP_{L1}$ | | | | |
|---|---|---|---|---|---|---|
| | log2(L1_SIZE) | Direct Mapped | 2-Ways | 4-Way | 8-Way | Fully-Assoc |
| 1 | 10 | 4.004147 | 3.275929 | 3.01509 | 2.88953 | 2.909184 |
| 2 | 11 | 3.09786 | 2.314401 | 2.088116 | 2.003756 | 1.957375 |
| 4 | 12 | 2.161025 | 1.694661 | 1.389675 | 1.266425 | 1.177898 |
| 8 | 13 | 1.51053 | 1.144925 | 1.065423 | 1.006861 | 0.984491 |
| 16 | 14 | 1.125027 | 0.903297 | 0.802766 | 0.811124 | 0.734238 |
| 32 | 15 | 0.991123 | 0.841326 | 0.80189 | 0.815131 | 0.75136 |
| 64 | 16 | 0.955917 | 0.845437 | 0.840071 | 0.861803 | 0.794861 |
| 128 | 17 | 1.01603 | 0.895193 | 0.89886 | 0.919816 | 0.841066 |
| 256 | 18 | 0.962392 | 0.964509 | 0.976265 | 0.977505 | 0.914589 |
| 512 | 19 | 1.082031 | 1.086324 | 1.082998 | 1.096757 | 0.994308 |
| 1024(1MB) | 20 | 1.21796 | 1.224626 | 1.218187 | 1.224399 | 1.107054 |

**Discussion**:

Q1. For a memory hierarchy with only an L1 cache and BLOCKSIZE = 32, which configuration yields the best (*i.e.*, lowest) AAT?

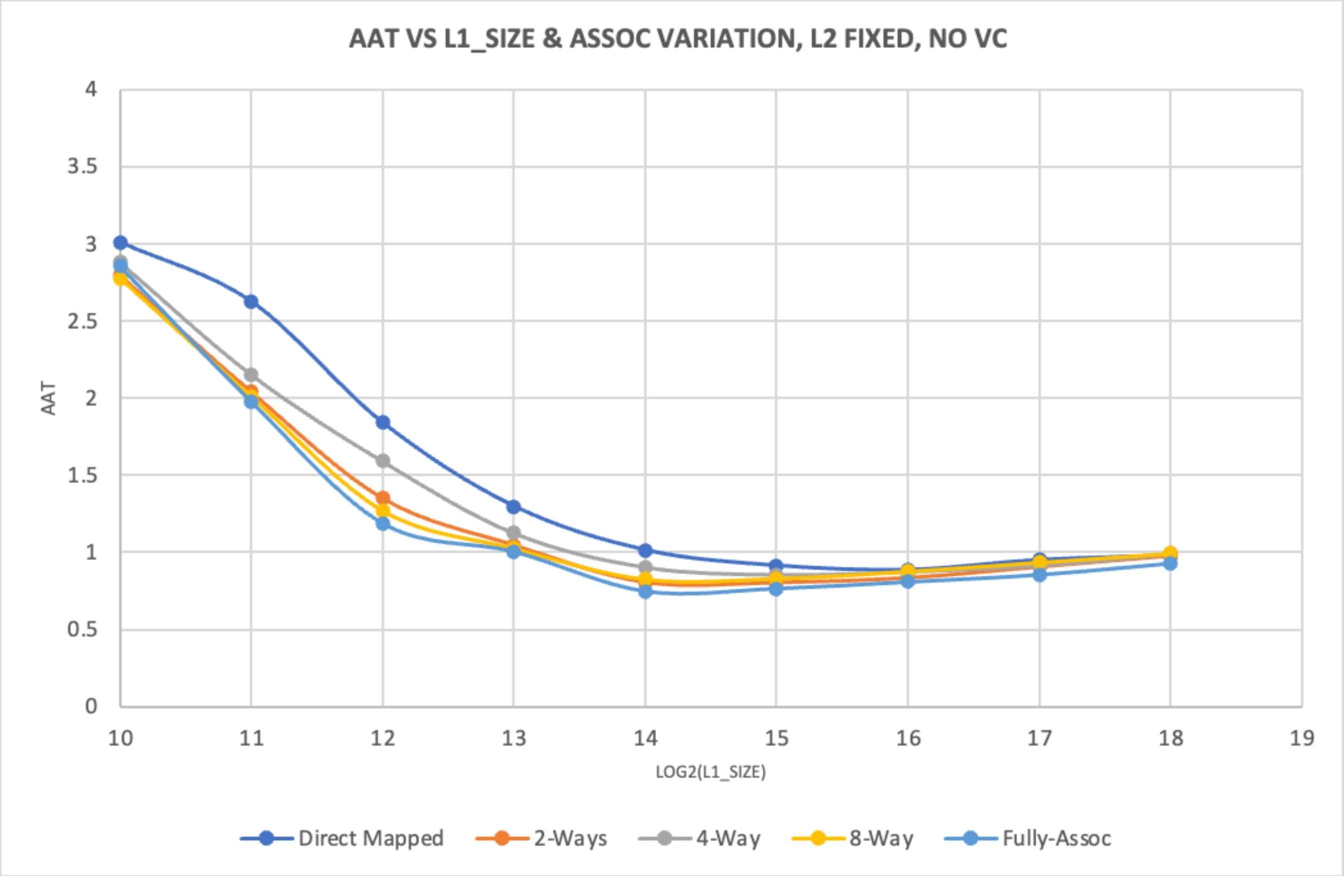
→ 16KB Fully Associative L1 Cache gives lowest AAT i.e. 0.734238 for fixed block size of 32bytes

## GRAPH #3 (*total number of simulations: 45*)

Same as GRAPH #2, except made the following changes:

Added the following L2 cache to the memory hierarchy: 512KB, 8-way set-associative,same block size as L1 cache.

Varied the L1 cache size only between 1KB and 256KB (since L2 cache is 512KB).



| BlockSize:32, vary L1 Size & Assoc, L2:512KB, 8WAY, VC Absent, | | | | | | |
|---|---|---|---|---|---|---|
| L1_SIZE(KB) | | AAT = HTL1 + MRL1(HTL2 +MRL2 *Miss_Penalty) | | | | |
| | log2(L1_SIZE) | Direct Mapped | 2-Ways | 4-Way | 8-Way | Fully-Assoc |
| 1 | 10 | 3.0079 | 2.7971 | 2.8806 | 2.7777 | 2.8551 |
| 2 | 11 | 2.6236 | 2.0403 | 2.1524 | 2.0106 | 1.9787 |
| 4 | 12 | 1.8393 | 1.3509 | 1.5903 | 1.2705 | 1.1898 |
| 8 | 13 | 1.2976 | 1.0479 | 1.125 | 1.0247 | 1.0055 |
| 16 | 14 | 1.0133 | 0.8073 | 0.9045 | 0.8254 | 0.7492 |
| 32 | 15 | 0.913 | 0.806 | 0.8565 | 0.8303 | 0.7665 |
| 64 | 16 | 0.8855 | 0.8363 | 0.8758 | 0.8768 | 0.8098 |
| 128 | 17 | 0.9498 | 0.9081 | 0.9158 | 0.9347 | 0.856 |
| 256 | 18 | 0.9773 | 0.9794 | 0.9912 | 0.9924 | 0.9295 |

**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%)?
   → From given experiment, its observed that by adding L2 cache(512KB 8-way) in hierarchy, following L1 cache configurations' in AATs are close to AATs(within 5%) found in #Graph2 configurations:
   - Direct mapped 1KB, 2KB, 4KB, and 16KB L1 cache
   - 2)Two-way L1 cache (1KB, 2KB, 4KB, 16KB)
   - 4 -way L1 cache (1KB, 2KB, 4KB, 16KB)
   - 8-way L1 cache (1 KB, 2 KB, 4KB)
   - Fully associative L1 cache (1 KB, 2 KB, 4 KB, 16 KB, 32 KB)

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?

→ We achieve the most optimal AAT for 16KB Fully Assoc L1 cache which gives optimal AAT of 0.7492. As compared to the #Graph2's optimal AAT: 0.734238, the #Graph3's AAT is 0.014962 more.

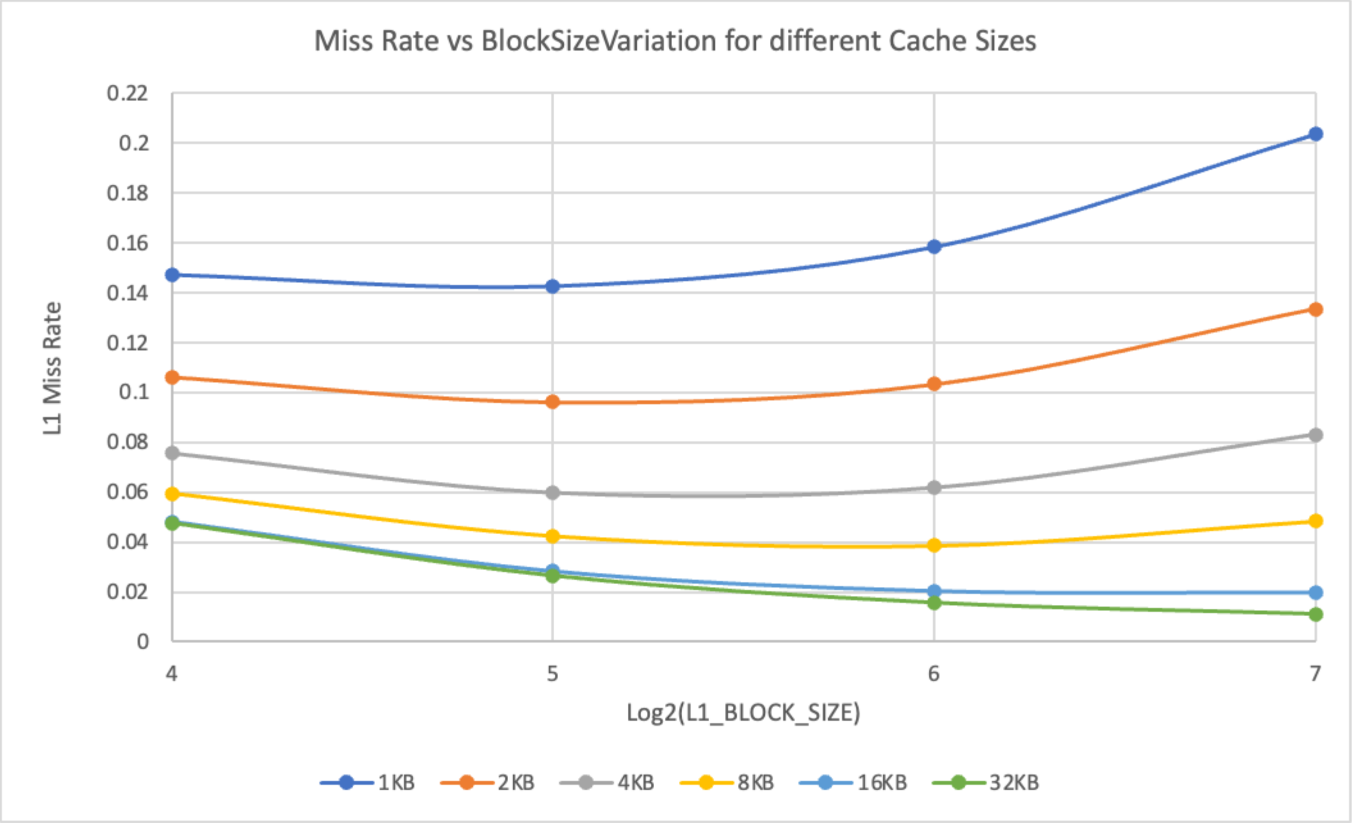
3. Compare the total area required for the optimal-AAT configurations with L2 cache (GRAPH #3) versus without L2 cache (GRAPH #2).

→ Total 2.703588092 (0.063446019 + 2.640142073) mm*mm is the area required for optimal AAT with L1+L2 Cache (GRAPH #3) hierarchy whereas 0.063446mm*mm is the total area required when only L1 cache present with optimal AAT (GRAPH #2). The area difference between these two hierarchies is 2.640142073.

## 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.



| Graph 4: Vary BlockSize, L1 Size & Assoc, VC & L2 Absent, | | | | | | | |
|---|---|---|---|---|---|---|---|
| L1_BLOCK_SIZE | Log2(L1_BlockSIZE) | Miss Rate | | | | | |
| | | 1KB | 2KB | 4KB | 8KB | 16KB | 32KB |
| 16Bytes | 4 | 0.1473 | 0.1062 | 0.0755 | 0.0595 | 0.0482 | 0.0475 |
| 32Bytes | 5 | 0.1427 | 0.0962 | 0.0599 | 0.0425 | 0.0283 | 0.0264 |
| 64Bytes | 6 | 0.1584 | 0.1033 | 0.0619 | 0.0386 | 0.0204 | 0.0156 |
| 128Bytes | 7 | 0.2036 | 0.1334 | 0.083 | 0.0483 | 0.0198 | 0.0111 |

**Discussion**

Q. 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 *increasingcache pollution* evident in the graph, and does the balance between these two factors shiftwith different cache sizes?

a)→ As shown in this graph, when the L1 size is small, we favor smaller block sizes, which results in a lower miss rate. We will be able to fit more blocks into the cache due to the tiny L1 size and smaller block size, lowering the miss rate and boosting the number of hits.

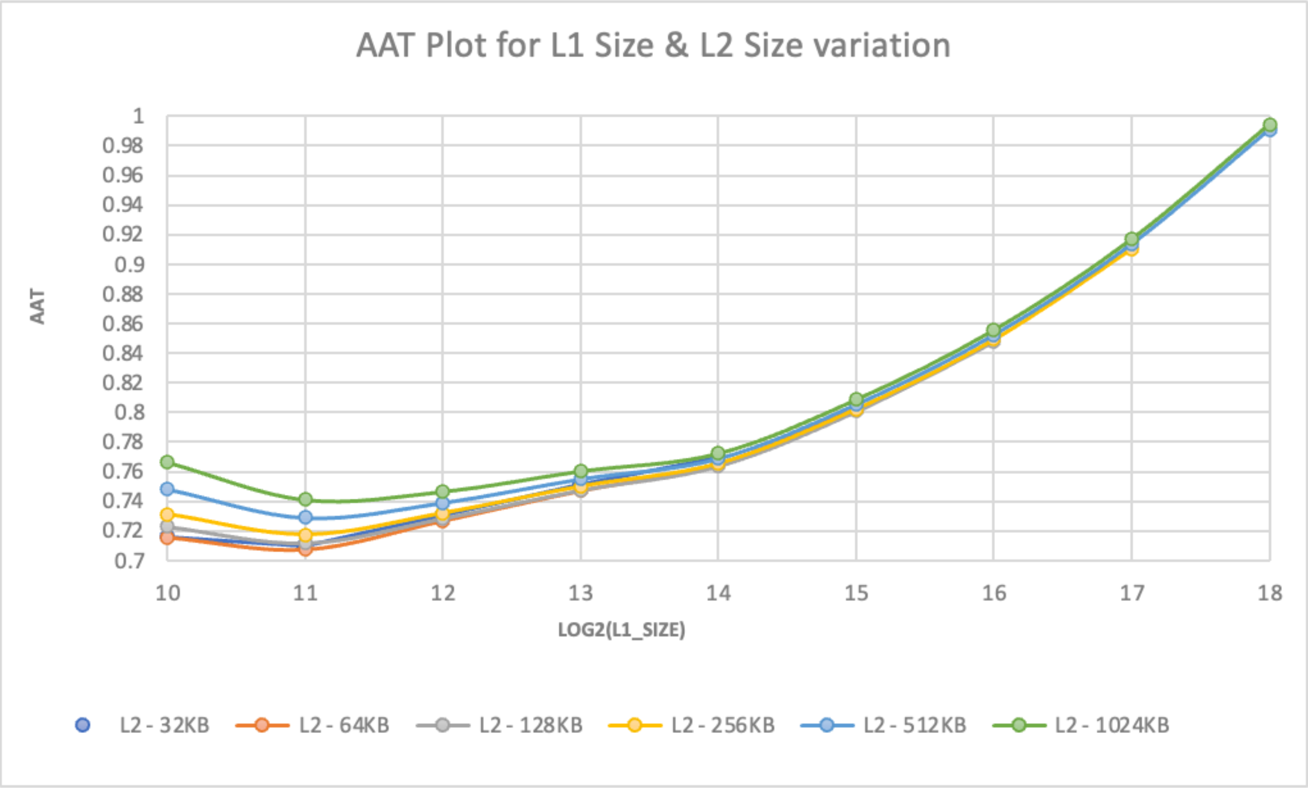
b)→ We may deduce from the graph that for large L1 sizes, we would choose larger block sizes. The miss rate tends to reduce. Because we have a big L1 size, we will be able to take advantage of spatial locality, resulting in a higher number of hits and a lower miss rate.

c)→ The tradeoff between utilizing spatial locality and cache pollution becomes clear when the block size increases from 16 to 128. For varied cache sizes, the balance is shifted for L1 cache size. The tradeoff is particularly obvious for L1 cache sizes of 1KB, 2KB, and 4KB, as there is less capacity in the cache and higher block size leads to cache pollution, raising the miss rate, and so breaking the balance. The tradeoff is fairly balanced for L1 cache sizes of 16KB and 32KB.

# 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.



| Blocksize:32, L1-1KB-256KB, 4Way, L2-32KB-1MB, 8Way, No VC | | | | | | | |
|---|---|---|---|---|---|---|---|
| L1_SIZE (KB) | | AAT = HTL1 + MRL1(HTL2 +MRL2 *Miss_Penalty) | | | | | |
| | Log2(L1_SIZE) | L2 - 32KB | L2 -64KB | L2 -128KB | L2 -256KB | L2 -512KB | L2 - 1024KB |
| 1 | 10 | 0.716 | 0.716 | 0.723 | 0.731 | 0.748 | 0.766 |
| 2 | 11 | 0.71 | 0.708 | 0.712 | 0.717 | 0.729 | 0.741 |
| 4 | 12 | 0.73 | 0.727 | 0.729 | 0.732 | 0.739 | 0.747 |
| 8 | 13 | 0.751 | 0.747 | 0.748 | 0.75 | 0.755 | 0.761 |
| 16 | 14 | 0.769 | 0.765 | 0.764 | 0.766 | 0.769 | 0.773 |
| 32 | 15 | | 0.802 | 0.801 | 0.802 | 0.805 | 0.809 |
| 64 | 16 | | | 0.848 | 0.849 | 0.852 | 0.856 |
| 128 | 17 | | | | 0.911 | 0.914 | 0.917 |
| 256 | 18 | | | | | 0.991 | 0.994 |

**Discussion**

Q1. Which memory hierarchy configuration yields the best (*i.e.*, lowest) AAT?

→ We discovered that L1 size of 2kb with 4-way associativity and L2 size of 64KB with 8-way associativity give the best AAT based on the data used to plot the graph.

Q2. Which memory hierarchy configuration has the smallest total area, that yields an AATwithin 5% of the best AAT?

→ The smallest area in the range of 5% of the best AAT is found in the L1 cache size of 1KB with 4-way associativity and the L2 cache size of 32kb with 8-way associativity.

END

*Sample cover page for Project #1.*