NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

Zong-Ru Li

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

Student's electronic signature: Li Zong-Ru

Course number: 563

Contents

1. Introduction

In this project, I built a simulator that can simulate four types of memory hierarchy: L1 cache only, L1 cache + victim cache, L1 cache + L2 cache, L1 cache + victim cache + L2 cache. This project report analyzed various cache configurations, explained the relation between different block offsets, cache size, associativity, and victim cache. All cache is implemented in the least-recently-used (LRU) replacement policy and write-back + write-allocate (WBWA) write policy.

2. Formulas

2.1 Memory Configurations

32-bit address = {tags, index, block offset}

block offset = log2(BLOCKSIZE)

$$set\ \# = \frac{Cache\ SIZE}{ASSOC * BLOCKSIZE}$$

index = log2(set #)

Tag = address >> (block offset + index)

2.2 Cache Miss Rate

$$swap\ \#\ =\ swaps\ between\ L1\ and\ its\ VC$$

$$\text{Combined L1 + VC miss rate} = \frac{L1\ read\ misses + L1\ write\ misses - swap\ \#}{L1\ reads + L1\ writes}$$

$$\text{L2 miss rate} = \frac{\text{L2 read miss}}{\text{L2 reads}}$$

2.3 Swap Request Rate (SRR)

$$\text{Swap request rate} = \frac{swap\ requests}{L1\ reads + L1\ writes}$$

2.4 Average Access Time (AAT)

HT = hit time
MR = miss rate

$$AAT = HT_{L1} + SRR * HT_{VC} + MR_{L1} * (HT_{L2} + MR_{L2} * \text{Miss Penalty})$$

3. Cache Simulator

The cache simulator is capable of simulating the L1 cache, L2 cache, victim cache.
We can set cache configuration during run time.

$./sim_cache <L1 SIZE> <L1 ASSOC> <VC Entry #> <L2 SIZE> <L2 ASSOC> <trace file>

The format of the trace file should be:

```
r    ffe04540
r    ffe04544
w    0eff2340
r    ffe04548
```

4. Experiments

Various experiments with different cache configurations will run to understand the cache performance affect attributes. All the experiments use GCC benchmark and CACTI tools, such as gcc_trace.txt and cacti_spreadsheet.xls.

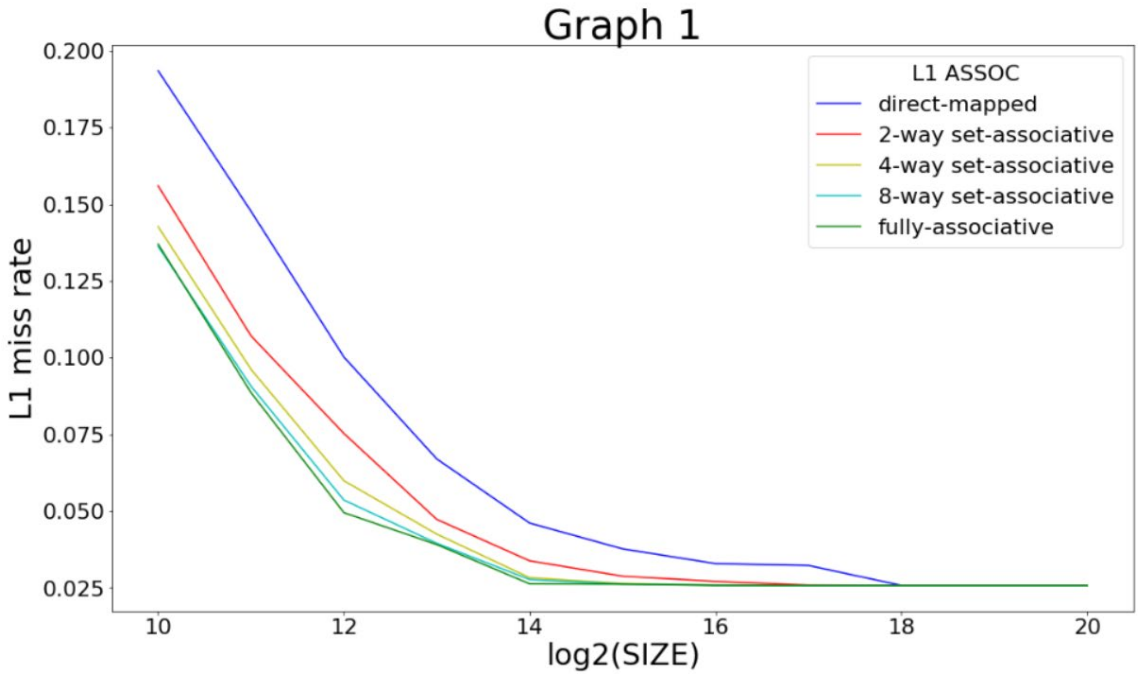
4.1. L1 Cache Exploration: SIZE and ASSOC

In this experiment, only L1 cache will be used (L2 cache and victim cache are disabled). The experiment aims to study the relationship between cache size, associativity, and miss rate. The cache configuration is given in Table 4.1. The experiment result is plotted as graphs in Graph 1.

Cache Attributes	Value
L1 BLOCKSIZE	32
L1 SIZE	Variable
L1 ASSOC	Variable
Victim Cache entry #	Disable
L2 BLOCKSIZE	Disable
L2 SIZE	Disable
L2 ASSOC	Disable

Table 4.1: Cache Configuration for log2(L1 Cache Size) vs. L1 miss rate

GRAPH #1



Graph 1: log2(L1 Cache Size) vs. L1 miss rate for different associativity

4.1.1 Graph 1 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?

For a given associativity, increasing cache size can decrease both conflict miss and capacity miss. When cache size increase, L1 miss rate decrease rapidly at the small cache size. L1 miss rate decreases slowly at the middle cache size. Finally, the L1 miss rate stopped dropping at 0.025. Because increasing cache size cannot reduce compulsory miss, the L1 miss rate finally stops at compulsory miss.

For a given cache size, increasing associativity decreases conflict miss. L1 miss rate decrease rapidly from direct-mapped to 2-way associative. However, the L1 miss rate decrease slowly from 2-way set-associative to 8-way set-associative. In this experiment, 8-way associative and fully associative has a very closed L1 miss rate. When associativity is fully associative,

there is no conflict miss. However, there might have capacity misses. This explained why fully associative cache has an L1 miss rate higher than 0.025 when the log2(size) equals 10 and 12.

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

Since the L1 miss rate stops decreasing when log2(size) equals 20, the lowest L1 miss rate is 0.025. The compulsory miss rate should be 0.025.

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

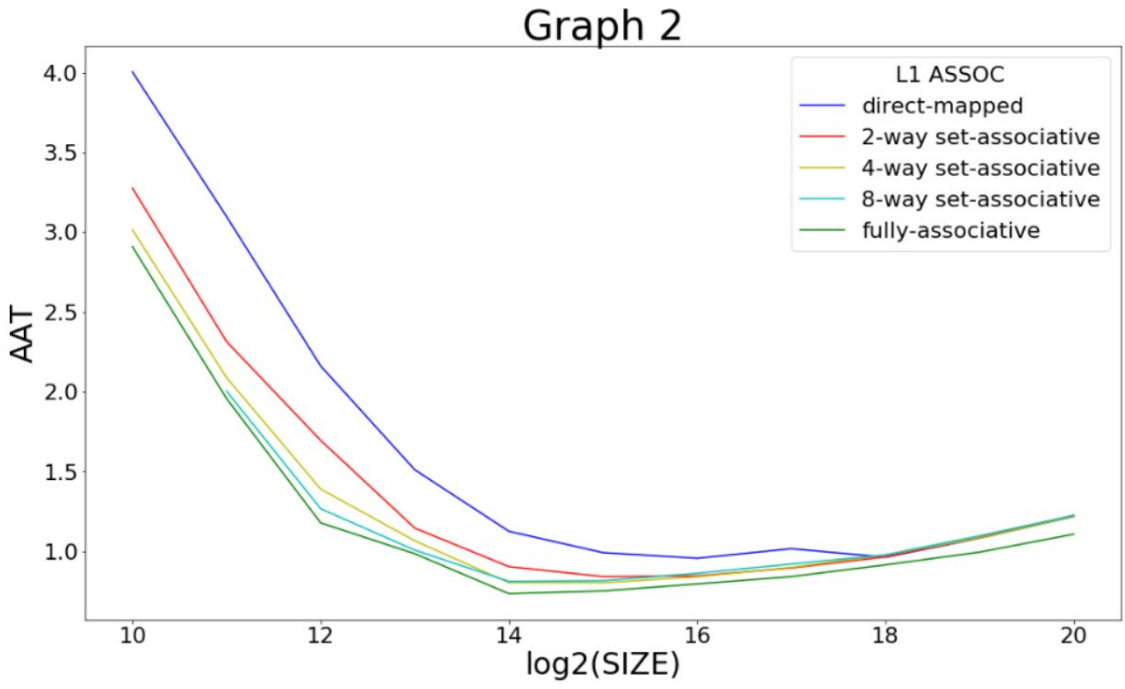
Since fully associate LRU cache only has capacity misses and compulsory misses. Thus, we can calculate conflict miss by this formula:
conflict miss = L1 miss rate of N-way set-associative – L1 miss rate of fully associative.

	direct-mapped	2-way set-associative	4-way set-associative	8-way set-associative	Fully-associative
1KB	0.0565	0.019	0.0057	0	0
2KB	0.0591	0.0185	0.0076	0.0021	0
4KB	0.0507	0.0258	0.0104	0.0041	0
Average	0.0554	0.021	0.078	0.0018	0

Table 4.2: experiment data for conflict miss of different L1 cache sizes and associativity.

In the next experiment, all the cache configurations will remain the same. The experiment aims to study the relationship between cache size and associativity, and average access time. The cache configuration is given in Table 4.1. The experiment result is plotted as graphs in Graph 2.

GRAPH #2



Graph 2: log2(L1 Cache Size) vs. average access time for different associativity

4.1.2 Graph 2 Discussion:

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

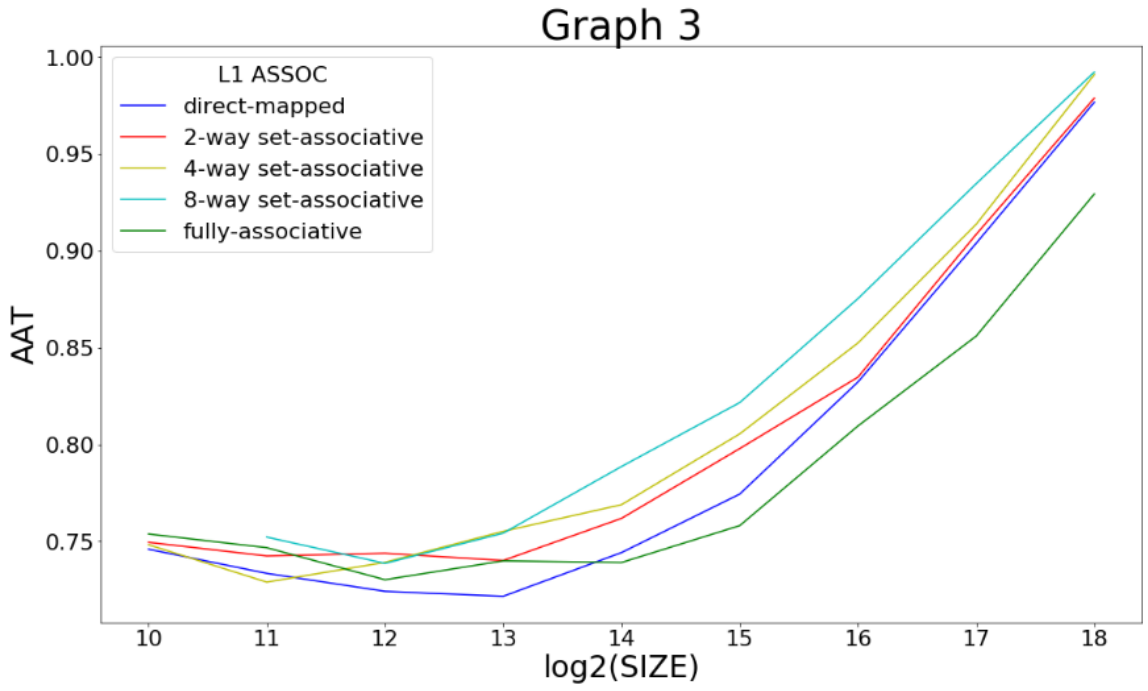
When associativity is fully associative, and L1 cache size is 4KB, cache yields the lowest AAT, 0.7342.

In this experiment, L1 and L2 cache will be used (victim cache is disabled). The purpose of the experiment is to study the relationship between cache size, associativity, and average access time. The cache configuration is given in Table 4.3. The experiment result is plotted as graphs in Graph 3.

Cache Attributes	Value
L1 BLOCKSIZE	32
L1 SIZE	Variable
L1 ASSOC	Variable
Victim Cache entry #	Disable
L2 BLOCKSIZE	32
L2 SIZE	512 KB
L2 ASSOC	8

Table 4.3: Cache Configuration for log2(L1 Cache Size) vs. average access time

GRAPH #3



Graph 3: log2(L1 Cache Size) vs. average access time for different associativity

4.1.3 Graph 3 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%)?

When associativity is direct-mapped, and L1 SIZE is 2KB, we get the AAT equal to 0.7334. 0.7334 is close to the best AAT in GRAPH #2.

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?

When associativity is direct-mapped and L1 SIZE is 8KB, we get the lowest AAT of 0.7215.

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

Regarding optimal-AAT with L2 cache, the L1 cache area is 0.053293238(mm*mm). L2 cache area is 2.640142073. The total cache area is 2.693435311(mm*mm).

Regarding optimal-AAT without L2 cache, the total cache area is equal to the L1 cache area of 0.016666697(mm*mm).

The total area of optimal-AAT with L2 cache is 161.6 times greater than that of optimal-AAT without L2 cache.

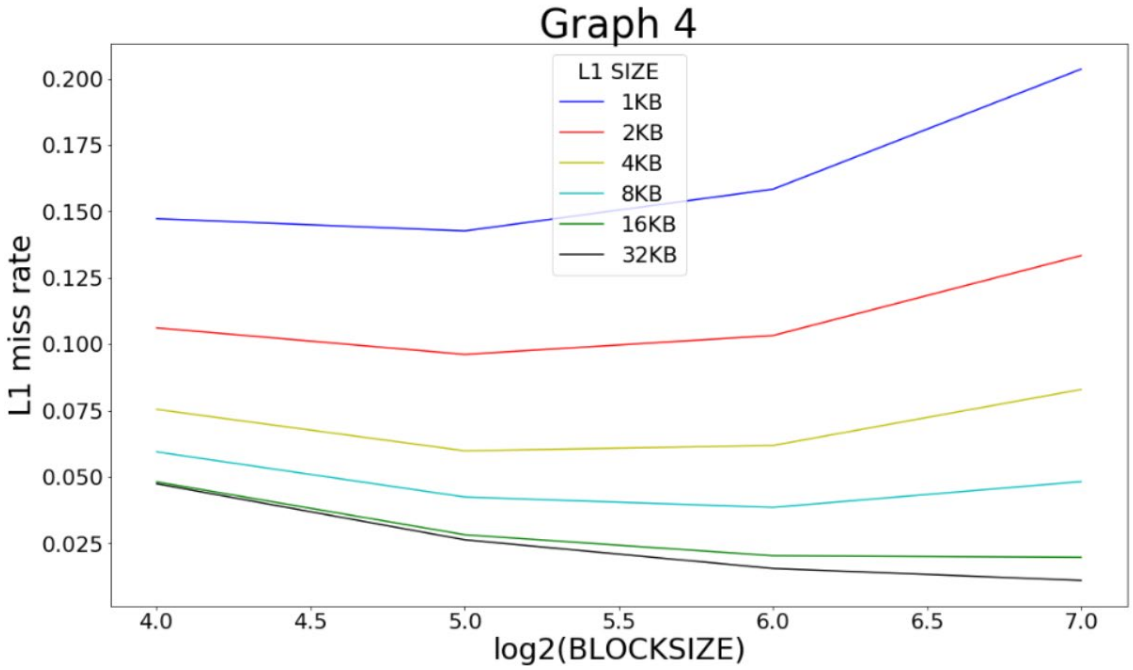
4.2. L1 Cache Exploration: SIZE, and BLOCKSIZE

In this experiment, only L1 cache will be used (L2 cache and victim cache is disabled). The purpose of the experiment is to study the relationship between cache size and block size, and cache miss rate. The cache configuration is given in Table 4.4. The experiment result is plotted as graphs in Graph 4.

Cache Attributes	Value
L1 BLOCKSIZE	Variable
L1 SIZE	Variable
L1 ASSOC	4
Victim Cache entry #	Disable
L2 BLOCKSIZE	Disable
L2 SIZE	Disable
L2 ASSOC	Disable

Table 4.4: Cache Configuration for log2(L1 Cache Size) vs. L1 miss rate

GRAPH #4



Graph 4: log2(L1 Cache Size) vs L1 miss rate for different L1 BLOCKSIZE

4.2.1 Graph 4 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?

Observed GRAPH #4, smaller L1 cache size has lower L1 miss rate at smaller block size. Also, a larger L1 cache size has a lower L1 miss rate at a larger block size. Therefore, smaller caches prefer smaller block sizes, and larger caches prefer larger block sizes.

Yes, when the block size increases from 16 to 128, the L1 cache with 1 KB size increases the L1 miss rate. Apparently, in this case, increasing block size increases more cache pollution and does not exploit more spatial locality.

Yes, the balance between these two factors shifts with different cache sizes. Regarding the 1 KB L1 cache, we find the lowest L1 miss rate when the block size equals 16. However, 32 KB L1 cache has the lowest L1 miss rate when block size equals 128. The balance of tradeoff shifts.

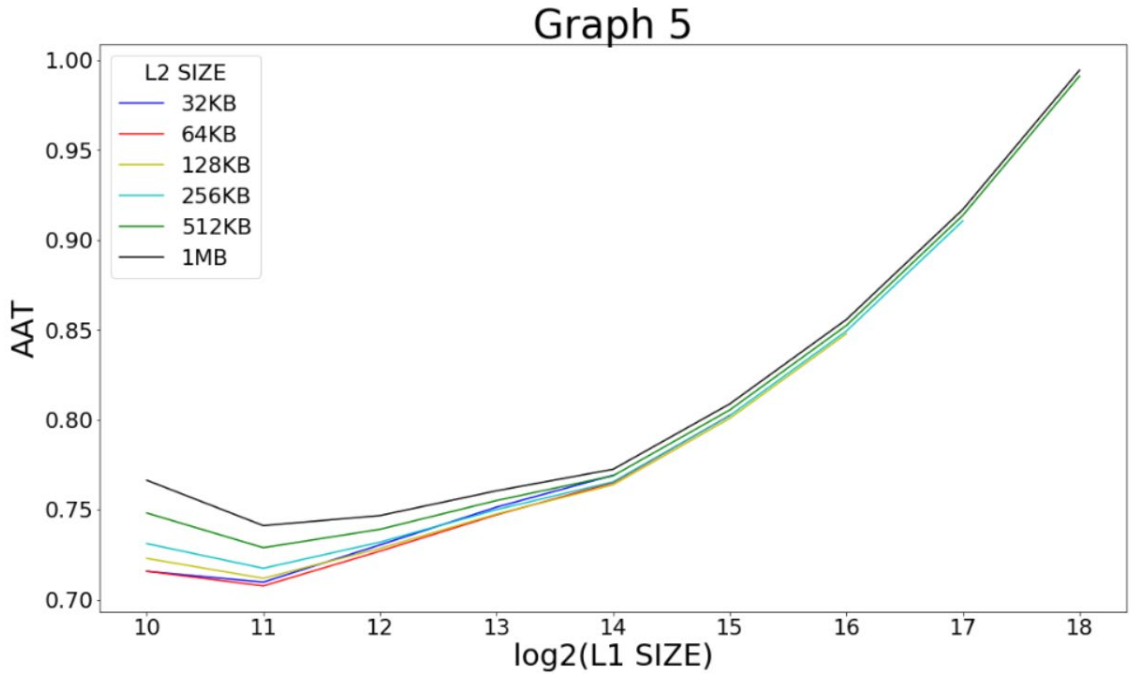
4.3. L1 + L2 Cache Co-Exploration

In this experiment, L1 and L2 cache will be used (victim cache is disabled). The experiment aims to study the relationship between cache size and block size, and cache miss rate. The cache configuration is given in Table 4.5. The experiment result is plotted as graphs in Graph 5.

Cache Attributes	Value
L1 BLOCKSIZE	32
L1 SIZE	Variable
L1 ASSOC	4
Victim Cache entry #	Disable
L2 BLOCKSIZE	32
L2 SIZE	Variable
L2 ASSOC	8

Table 4.5: Cache Configuration for log2(L1 Cache Size) vs average access time

GRAPH #5



Graph 5: log2(L1 Cache Size) vs average access time for different L1 SIZE

4.3.1 Graph 5 Discussion:

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

When L1 cache size is equal to 2 KB, and L2 cache size equals 64KB, we get the lowest AAT at 0.7077.

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

When the L1 cache size is 1 KB and the L2 cache size is 32 KB, we get the smallest total area which is 0.257285583. The AAT is 0.7158.

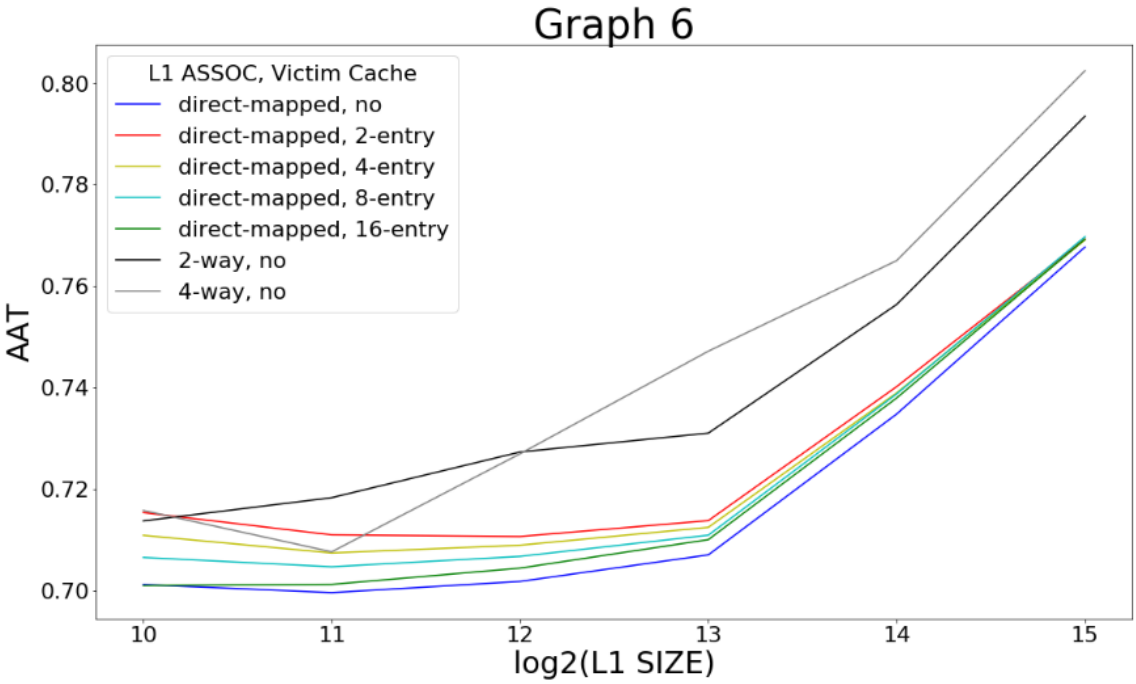
4.4. Victim Cache Study

In this experiment, L1 cache, L2 cache, and victim cache will be used. The purpose of the experiment is to study the relationship between cache size and set associativity, and victim cache. The cache configuration is given in Table 4.6. The experiment result is plotted as graphs in Graph 6.

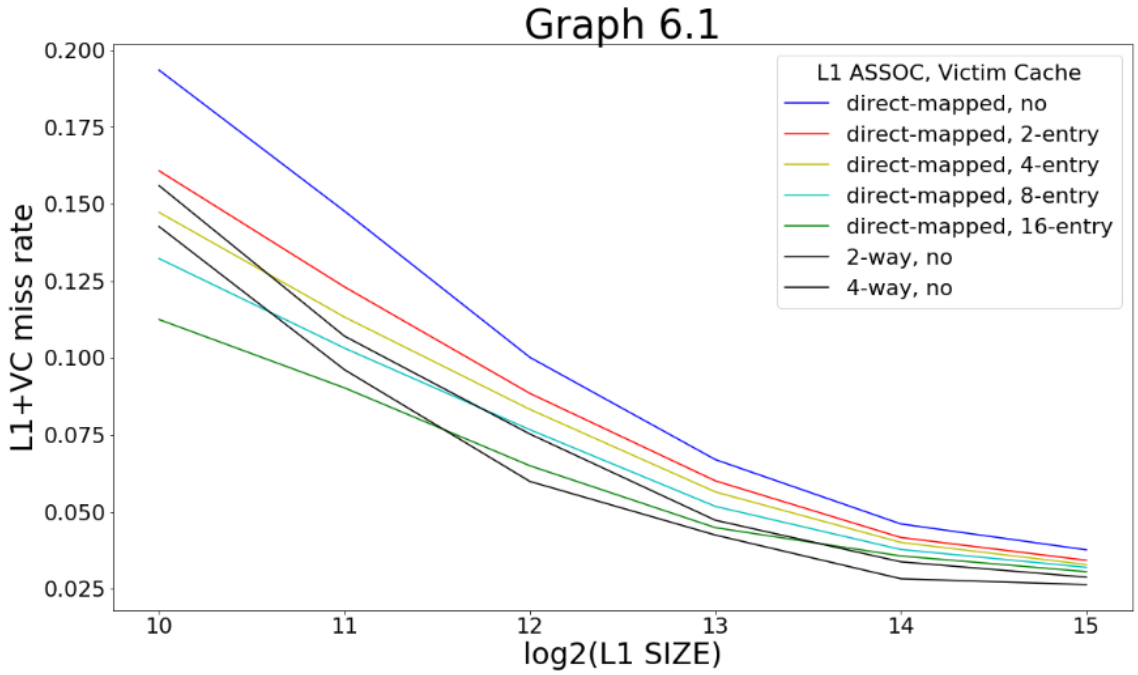
Cache Attributes	Value
L1 BLOCKSIZE	32
L1 SIZE	Variable
L1 ASSOC	Variable
Victim Cache entry #	Variable
L2 BLOCKSIZE	32
L2 SIZE	64 KB
L2 ASSOC	8

Table 4.6: Cache Configuration for log2(L1 Cache Size) vs. average access time

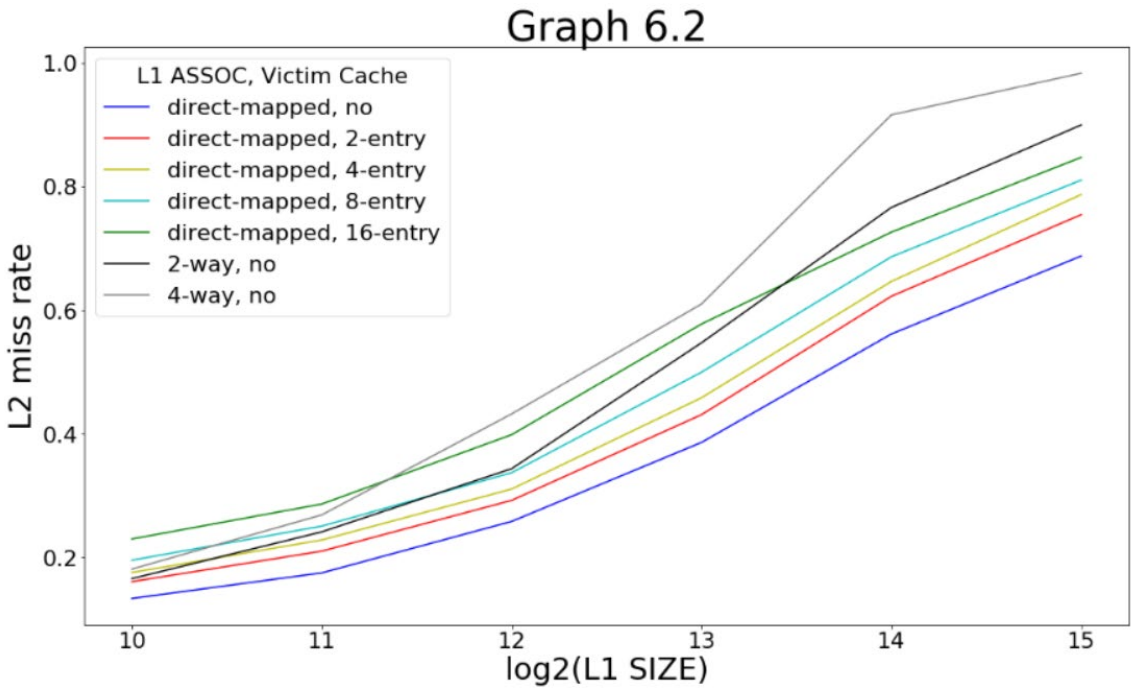
GRAPH #6



Graph 6: log2(L1 Cache Size) vs. average access time for different L1 SIZE



Graph 6.1: log2(L1 Cache Size) vs. L1 + victim cache miss rate for different L1 SIZE



Graph 6.2: log2(L1 Cache Size) vs. L2 cache miss rate for different L1 SIZE

4.4.1 Graph 6 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?

Comparing three cases: direct-mapped L1 cache without victim cache, 2-way set-associative L1 cache, and 4-way set-associative cache, we find that 2-way and 4-way set-associative L1 cache has higher average access time. Even 2-way and 4-way set-associative L1 caches have lower L1 miss rates (refer to GRAPH #6.1). Their average access time is still higher than the direct-mapped L1 cache because their cache access time is 1.3 times greater than the direct-mapped L1 cache.

We are comparing five cases: direct-mapped L1 cache with 2,4,8,16 victim cache and without victim cache. We find that direct-mapped L1 caches with victim cache have a higher average access time than the same size direct-mapped L1 cache without victim cache. Because swap request rate is around 20%, and victim cache access time is approximately 0.13, which is even higher than direct-mapped regular cache's access time. These overhead increases more than 20% access time to the original direct-mapped L1 cache. Even victim cache can decrease L1+VC miss rate a lot; the overhead of victim cache access time is still too high.

Observed GRAPH #6, we can find there are two particular points. The first point occurs at L1 cache size equals 1 KB. Direct-mapped L1 cache with 2-entry victim cache has an average access time at 0.7154, which is very close to 2-way associative L1 cache's average access time, which is around 0.7158. The second point happened at the L1 cache equal to 2 KB. Direct-mapped L1 cache with 4-entry victim cache has average access time around 0.7074, and 2-way associative L1 cache's average access time is 0.7077. these two cache configurations have very close performance in average access time.

Regarding L1 + VC miss rate, adding a victim cache to a direct-mapped L1 cache has similar performance improvement as a 2-way set-associative L1 cache of the same size. However, both two cache configurations have overheads that increase average access time.
For victim cache, there is an extra victim cache hit time for the L1 cache to access the victim cache. According to the CACTI spreadsheet, for 2-way and 4-way set-associative cache, they have higher cache access time than direct-mapped cache in the same size.

Finally, when I observed the L2 miss rate, I found that when we decrease the L1 miss rate by increasing associativity or increasing the entry number of victim cache, the L2 miss rate will increase. Because when we catch more spatial locality at the L1 cache, there will be less spatial locality remain to the next level cache hierarchy. Therefore, the L2 miss rate increased.

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

L1 cache size equals 2 KB, and associativity is direct-mapped. AAT is equal to 0.6996, which is the lowest AAT in GRAPH #6.

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

When L1 cache size is 2 KB, and associativity is 2-way set associative, the total cache area is 0.369789343(mm*mm). AAT is equal to 0.7138.

5. Summary

Affected Cache Attribute	Advantages	Disadvantages
Block size	Miss rate may decrease, up to a point, due to exploiting more spatial locality	Miss rate may increase after a point, due to cache pollution
Cache size	Decrease miss rate	Increase in hit time and area
Set associativity	Decrease conflict misses Increase hit time	Increase in hit time and energy per access
Victim cache	Reduce L1 miss rate and miss penalty	Adding extra victim cache access time, increase L2 miss rate