NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

By

PARTH BHOGATE

(NCSU ID: 200108628)

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

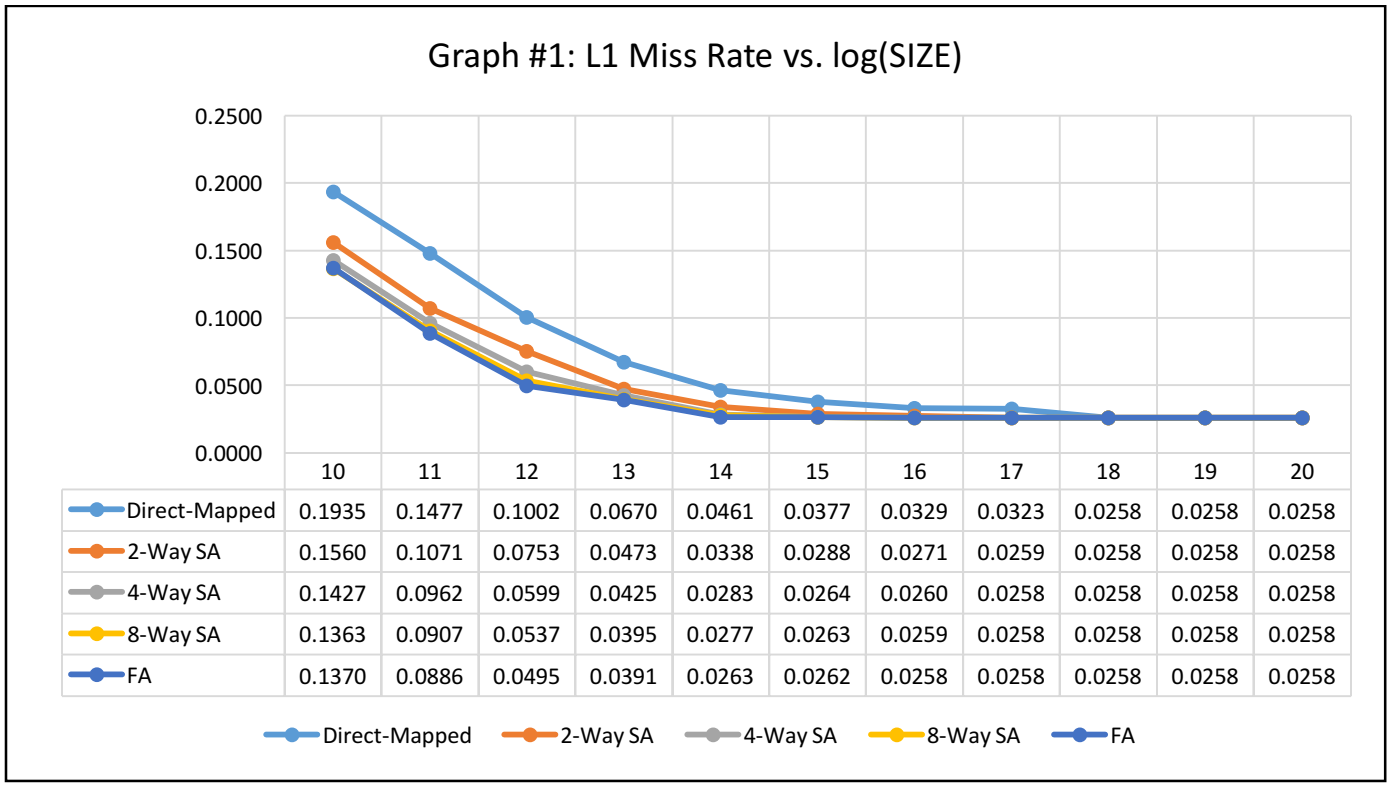
Student's electronic signature: Parth Bhogate
(sign by typing your name)

Course number: ECE 521

Graph #1: L1 Miss Rate vs. $\log_2$(L1 SIZE), without VC and L2

Cache Configuration:
L1 Cache Size = 1KB, 2KB, 4KB, …, 1MB
L1 Cache Associativity: Direct-mapped, 2-way SA, 4-way SA, 8-way SA,
Fully associative
No VC, No L2 Cache

The L1 miss rate is plotted against the cache size. Each line on the plot shows the miss rate for a particular associativity: direct-mapped, 2-way set associative, 4-way set associative, 8-way set associative and fully associative.



	10	11	12	13	14	15	16	17	18	19	20
Direct-Mapped	0.1935	0.1477	0.1002	0.0670	0.0461	0.0377	0.0329	0.0323	0.0258	0.0258	0.0258
2-Way SA	0.1560	0.1071	0.0753	0.0473	0.0338	0.0288	0.0271	0.0259	0.0258	0.0258	0.0258
4-Way SA	0.1427	0.0962	0.0599	0.0425	0.0283	0.0264	0.0260	0.0258	0.0258	0.0258	0.0258
8-Way SA	0.1363	0.0907	0.0537	0.0395	0.0277	0.0263	0.0259	0.0258	0.0258	0.0258	0.0258
FA	0.1370	0.0886	0.0495	0.0391	0.0263	0.0262	0.0258	0.0258	0.0258	0.0258	0.0258

1. Trends in the graph:
From the graph, we can observe that the miss rate decreases (almost) exponentially for an increasing cache size. For higher cache sizes, the miss rates for the various cache associativities converge to approximately the same value. With a growing cache size, the conflict and capacity misses are nullified, and only the compulsory miss rate can still be observed in the graph.
For a given cache size, the miss rate decreases with an increase in cache associativity. This decrease is primarily due to the decrease in conflict miss rates.

2. Compulsory Miss Rate Estimation:

Compulsory misses occur when a particular cache block is referenced for the first time. Compulsory misses cannot be reduced by the standard techniques, except for software or hardware prefetching. The miss rate observed in a sufficiently large cache with full associativity can be approximated to be the compulsory miss rate. From the graph, the compulsory miss rate can be estimated to be **0.0258.**

3. Conflict Miss Rate Estimation:

Misses which occur in the cache under test but not in the fully-associative cache being used for comparison are classified as conflict misses. The difference between the miss rate for the fully-associative cache versus that for a N-Way set associative cache is the fraction of the miss rate that can be attributed to conflict misses. From the graph, the following values for conflict miss rate are obtained:

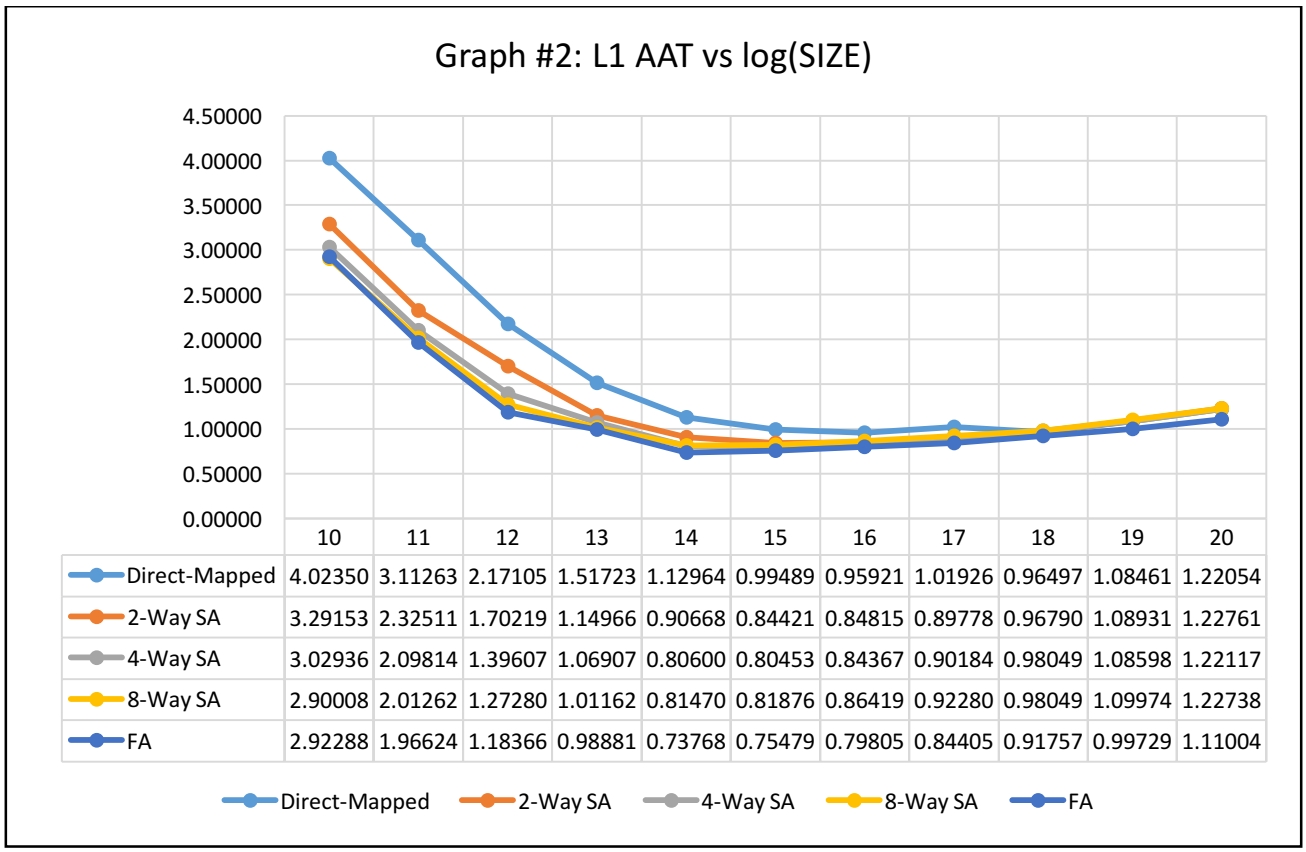
Assoc / Cache Size	1KB	2KB	4KB	8KB	16KB	32KB	64KB	128KB	256KB	512KB	1MB
Direct-Mapped	.056	.059	.051	.027	0.19	.011	.007	0.006	0.00	0.00	0.00
2-Way SA	.019	.018	.026	.008	.007	.003	.001	.0001	0.00	0.00	0.00
4-Way SA	.006	.007	.010	.003	.002	.0002	.0001	0.00	0.00	0.00	0.00
8-Way SA	--	.002	.004	0.00	.001	0.00	0.00	0.00	0.00	0.00	0.00

A fully-associative cache does not suffer from conflict misses; it suffers only compulsory and capacity misses.

Graph #2: AAT vs. log2(L1 SIZE), without VC or L2

Cache Configuration:
Blocksize = 32
L1 Cache Size = 1KB, 2KB, 4KB, …, 1MB
L1 Cache Associativity: Direct-mapped, 2-way SA, 4-way SA, 8-way SA, Fully associative
No VC, No L2 Cache

The graph plots the Average Access Time (AAT) for L1 cache for different cache sizes and cache associativities. L2 Cache or Victim Cache is not included in the memory hierarchy.



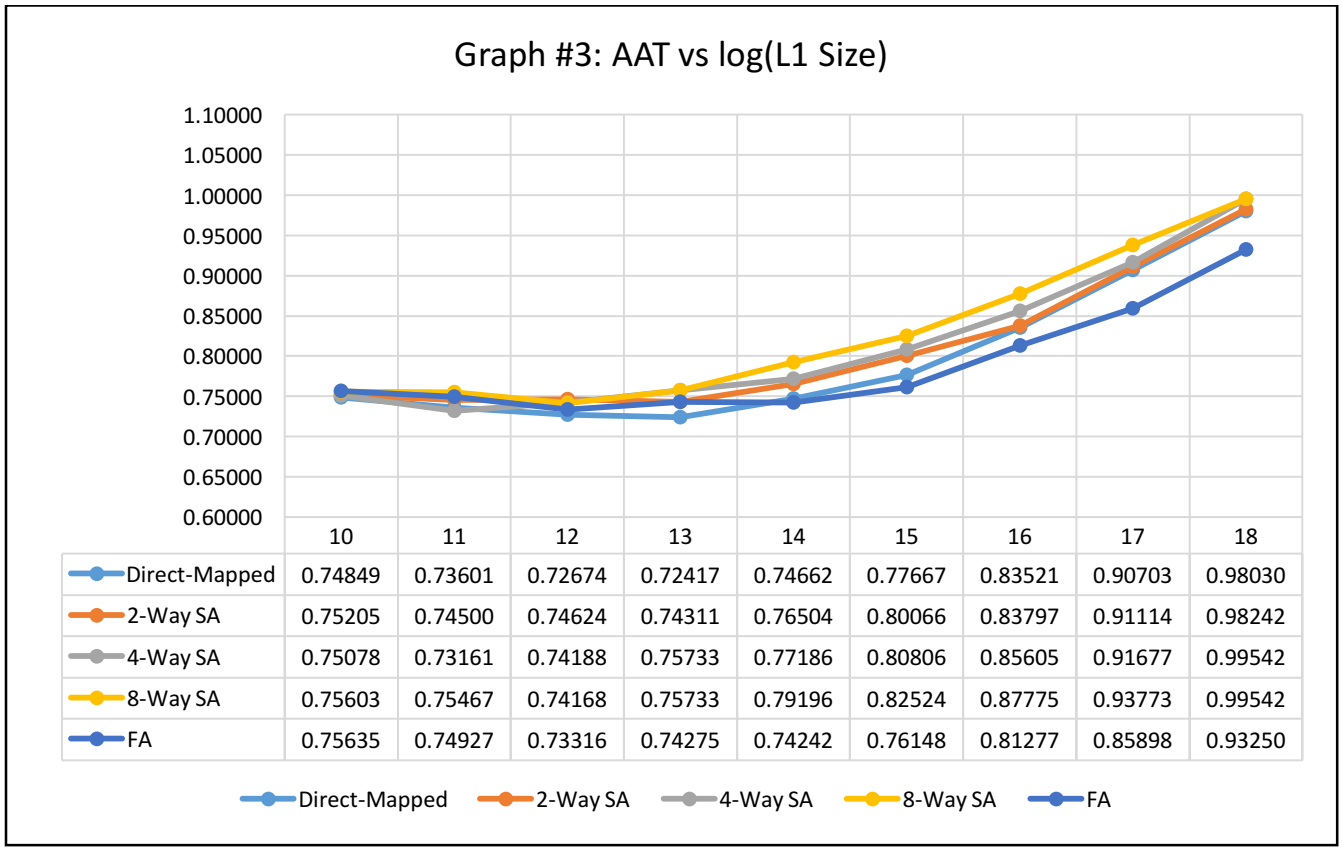
1. Which configuration yields the lowest AAT?
From the graph, the lowest AAT observed is **0.737 ns**, which is for **a fully-associative L1 cache of size 16KB**. AAT depends upon the cache hit time, miss rate and the miss penalty. The miss penalty is the same for all the configurations. Hit time and the miss rate depend both on the associativity and the cache size. Hit time increases with the cache size, and the miss rate decreases with size. It is thus a trade-off between the size and associativity. With a cache of 16KB and full associativity, the trade-off gives the optimal value of the AAT.

In caches with full-associativity, the AAT increases with further increase in the cache size on account of increase in the hit time.

Graph #3: AAT vs. log2(L1 SIZE), with L2 Cache included

Cache Configuration:
Blocksize = 32
L1 Cache Size = 1KB, 2KB, 4KB, …, 1MB
L1 Cache Associativity: Direct-mapped, 2-way SA, 4-way SA, 8-way SA,
Fully associative
L2 Cache Size = 512KB
L2 Associativity: 8-way set associative
No VC

The graph plots the AAT versus the L1 cache size for different L1 cache sizes and associativities. L2 cache of fixed size and associativity is added to the memory hierarchy. Victim cache is not present.



	10	11	12	13	14	15	16	17	18
Direct-Mapped	0.74849	0.73601	0.72674	0.72417	0.74662	0.77667	0.83521	0.90703	0.98030
2-Way SA	0.75205	0.74500	0.74624	0.74311	0.76504	0.80066	0.83797	0.91114	0.98242
4-Way SA	0.75078	0.73161	0.74188	0.75733	0.77186	0.80806	0.85605	0.91677	0.99542
8-Way SA	0.75603	0.75467	0.74168	0.75733	0.79196	0.82524	0.87775	0.93773	0.99542
FA	0.75635	0.74927	0.73316	0.74275	0.74242	0.76148	0.81277	0.85898	0.93250

1. With L2 added, which L1 cache configurations result in AATs close to the best AAT observed without L2 in the memory hierarchy?

The lowest AAT observed in Graph #2 was 0.737 ns. Looking at the AAT values in Graph #3 which are lesser than 0.774 ns, we observe that the AATs for the following configurations are within 5% of the lowest AAT in Graph #2:
(Yes implies that AAT is within 5% of the best AAT from Graph #2; No implies the AAT is not within a 5% range.)

Assoc/Cache Size	1KB	2KB	4KB	8KB	16KB	32KB	64KB	128KB	256KB
Direct-Mapped	Yes	Yes	Yes	Yes	Yes	No	No	No	No
2-Way SA	Yes	Yes	Yes	Yes	Yes	No	No	No	No
4-Way SA	Yes	Yes	Yes	Yes	Yes	No	No	No	No
8-Way SA	Yes	Yes	Yes	Yes	No	No	No	No	No
Fully-Associative	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No

2. With L2 added to the memory hierarchy, which L1 cache configuration yields the lowest AAT?

The lowest AAT as seen in the graph is 0.724 ns, for a **direct-mapped L1 cache of 8KB** size. In Graph #2 which did not include a L2 cache, the lowest observed AAT was 0.737 ns; thus, the memory hierarchy with L2 included has a AAT which is **0.013 ns lower, which is 1.76%** lower than the one without L2. The presence of L2 reduces the miss penalty in the hierarchy and thus leads to a decrease in AAT.

3. Comparison of total areas required for optimal-AAT configuration with and without L2 cache

From Graph #2, the total area for optimal AAT configuration is as follows:
Area of 16KB fully-associative L1 cache without L2 cache:
Area of L1 Cache = **0.063446019 mm^2**.

From Graph #3, the total area for optimal AAT configuration is:
Area of 8KB direct-mapped L1 cache with 512KB 8-way associative L2 cache -
Area of L1 Cache + Area of L2 Cache
= 0.053293238 + 2.640142073
Total memory area = **2.693435311 mm^2**.

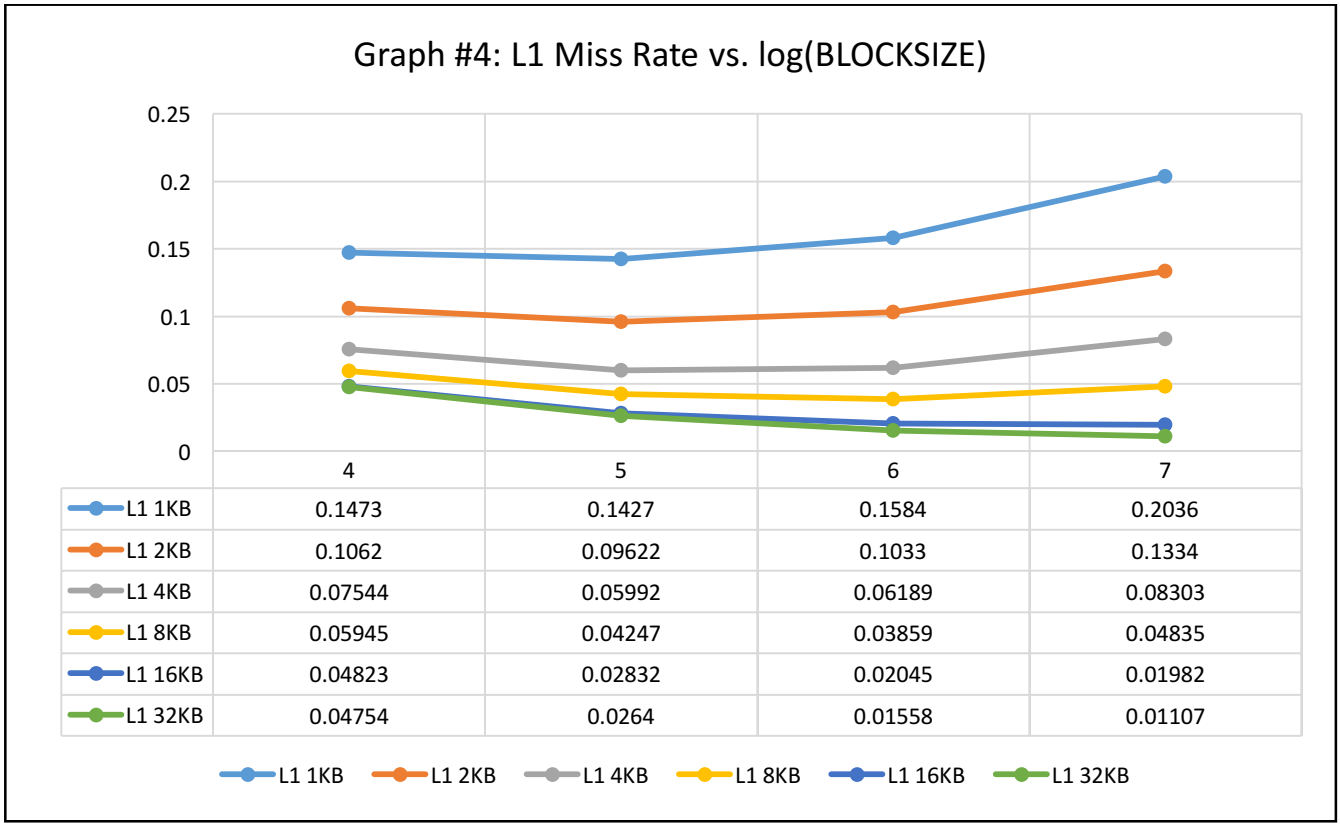
The increase in area when L2 cache is included in the memory hierarchy is:
2.693435311 - 0.063446019 = **2.629989292 mm^2**
Percentage increase is 2.629989292/0.063446019 * 100 = **4145.24%.**

Thus, for optimal AAT, the total area for L1-only cache configuration is much lower compared to a hierarchy with L2 included. For a small improvement in the best AAT, there is huge increase in the total memory hierarchy area. This clearly shows the trade-off between achieving an improvement in the AAT and the memory hierarchy area.

Graph #4: L1 Miss Rate vs. log2(BLOCKSIZE), for fixed L1 Associativity

Cache Configuration:
Blocksize = 32
L1 Cache Size = 1KB, 2KB, 4KB, …, 1MB
L1 Cache Associativity: 4-way set associative
No VC, No L2 Cache

The graph plots the L1 miss rate versus the L1 blocksize. The associativity of the cache is fixed, and the miss rate is plotted versus different blocksizes so as to study the effect of varying the cache block size. Each line in the graph depicts the miss rate for a particular L1 cache size.



	4	5	6	7
L1 1KB	0.1473	0.1427	0.1584	0.2036
L1 2KB	0.1062	0.09622	0.1033	0.1334
L1 4KB	0.07544	0.05992	0.06189	0.08303
L1 8KB	0.05945	0.04247	0.03859	0.04835
L1 16KB	0.04823	0.02832	0.02045	0.01982
L1 32KB	0.04754	0.0264	0.01558	0.01107

1. Trends observed in the graph:
The cache blocksize influences the amount of spatial locality present in the cache. If the address trace has good spatial locality, a larger blocksize will help to reduce the miss rate. On the other hand, a large blocksize leads to fewer unique blocks being present in the cache. Thus, a larger blocksize leads to lesser variety of blocks in the cache. This might lead to an increase in miss rate.

For smaller cache sizes, increasing the block size initially decreases the miss rate, but the cache pollution effect takes over quickly since the low cache size does not permit bringing in a wide variety of blocks. Thus, a **small cache size works better with a smaller blocksize**.

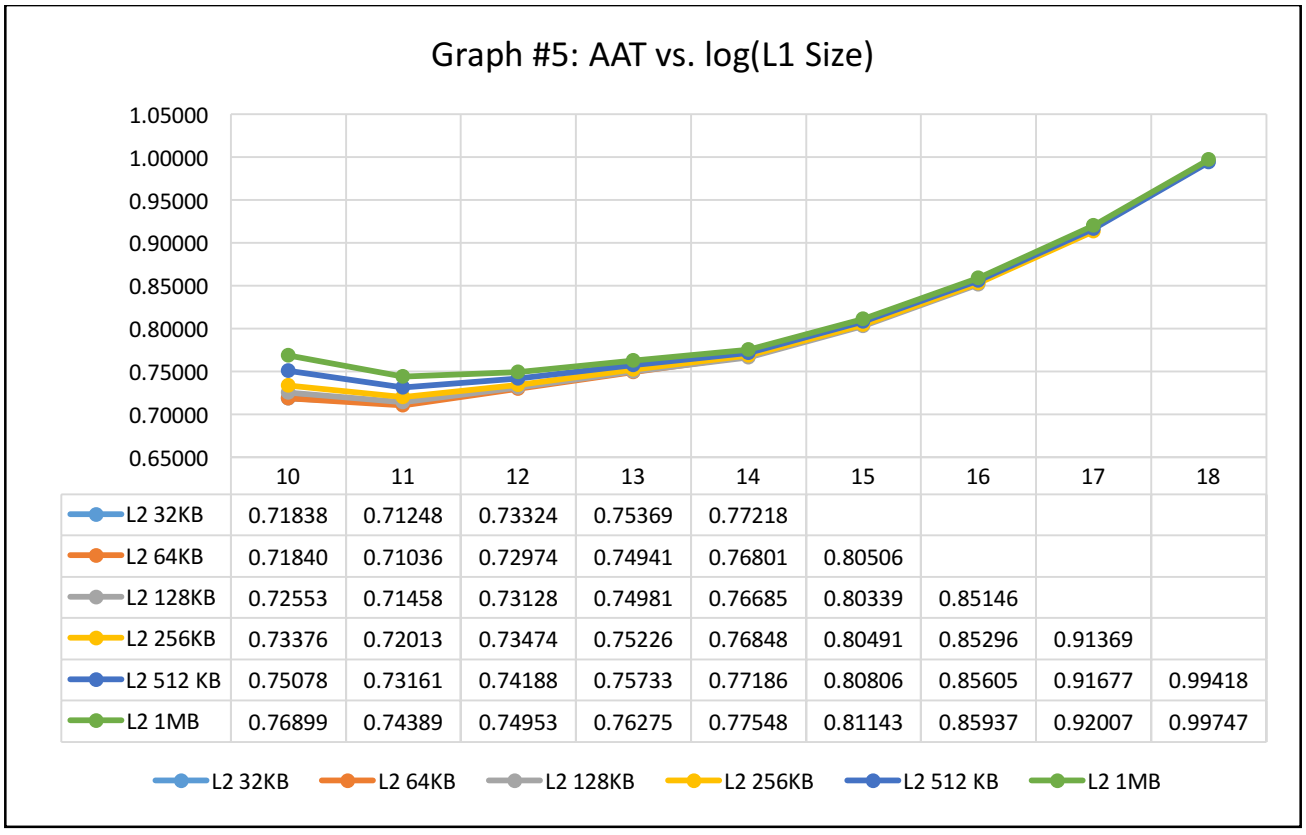
In the case of a larger cache size, the miss rate decreases with an increase in blocksize because of exploiting the spatial locality. Since the large cache can bring in a variety of different blocks along with sequential blocks, the miss rate goes on decreasing with an increasing blocksize. The cache pollution effect does not kick in (atleast until the blocksize is made a significant percent of the total cache size) for the case of a large cache size. Thus, **a large cache size works better with a larger blocksize**.

The effect on miss rate of an increase in blocksize depends upon the trade-off between the positive impact of the increase in spatial locality and the negative impact of possible cache pollution because of bringing in lesser variety of memory blocks. The overall effect of varying blocksize depends upon the total cache size. For a smaller cache size, the cache pollution effect begins to dominate, whereas for a larger cache size the spatial locality provides a greater positive impact than the negative effect of cache pollution.

Graph #5: AAT vs. log2(L1 SIZE), with different L2 Cache Sizes

Cache Configuration:
Blocksize = 32
L1 Cache Size = 1KB, 2KB, 4KB, …, 1MB
L1 Cache Associativity: 4-way set associative
L2 Cache Size = 32KB, 64KB, 128KB, 256KB, 512Kb, 1MB
L2 Cache Associativity: 8-way set associative
No VC

The graphs looks at the AAT variation with L1 cache sizes for different L2 cache sizes included in the memory hierarchy. The associativity for both L1 and L2 caches is fixed. Victim cache is not included with L1 cache.



	10	11	12	13	14	15	16	17	18
L2 32KB	0.71838	0.71248	0.73324	0.75369	0.77218				
L2 64KB	0.71840	0.71036	0.72974	0.74941	0.76801	0.80506			
L2 128KB	0.72553	0.71458	0.73128	0.74981	0.76685	0.80339	0.85146		
L2 256KB	0.73376	0.72013	0.73474	0.75226	0.76848	0.80491	0.85296	0.91369	
L2 512 KB	0.75078	0.73161	0.74188	0.75733	0.77186	0.80806	0.85605	0.91677	0.99418
L2 1MB	0.76899	0.74389	0.74953	0.76275	0.77548	0.81143	0.85937	0.92007	0.99747

1. Which memory hierarchy configuration yields the lowest AAT?
The lowest AAT as obtained from the graph is for a **2KB 4-way set associative L1 cache along with a 64KB 8-way set associative L2 cache**. The lowest AAT value is found to be **0.710 ns**. The AAT in this hierarchy involves a complex trade-off between the L1 hit time

(which depends on L1 cache size), the L1 miss rate (depends onL1 size and associativity), the L2 hit time (depends on L2 size), and the L2 miss rate (again depends on L2 size and associativity). Moving towards bigger cache sizes or associativities leads to a decrease in miss rate but an increase in hit time. The optimal AAT is the one which has the balance between size and associativity for both the caches.

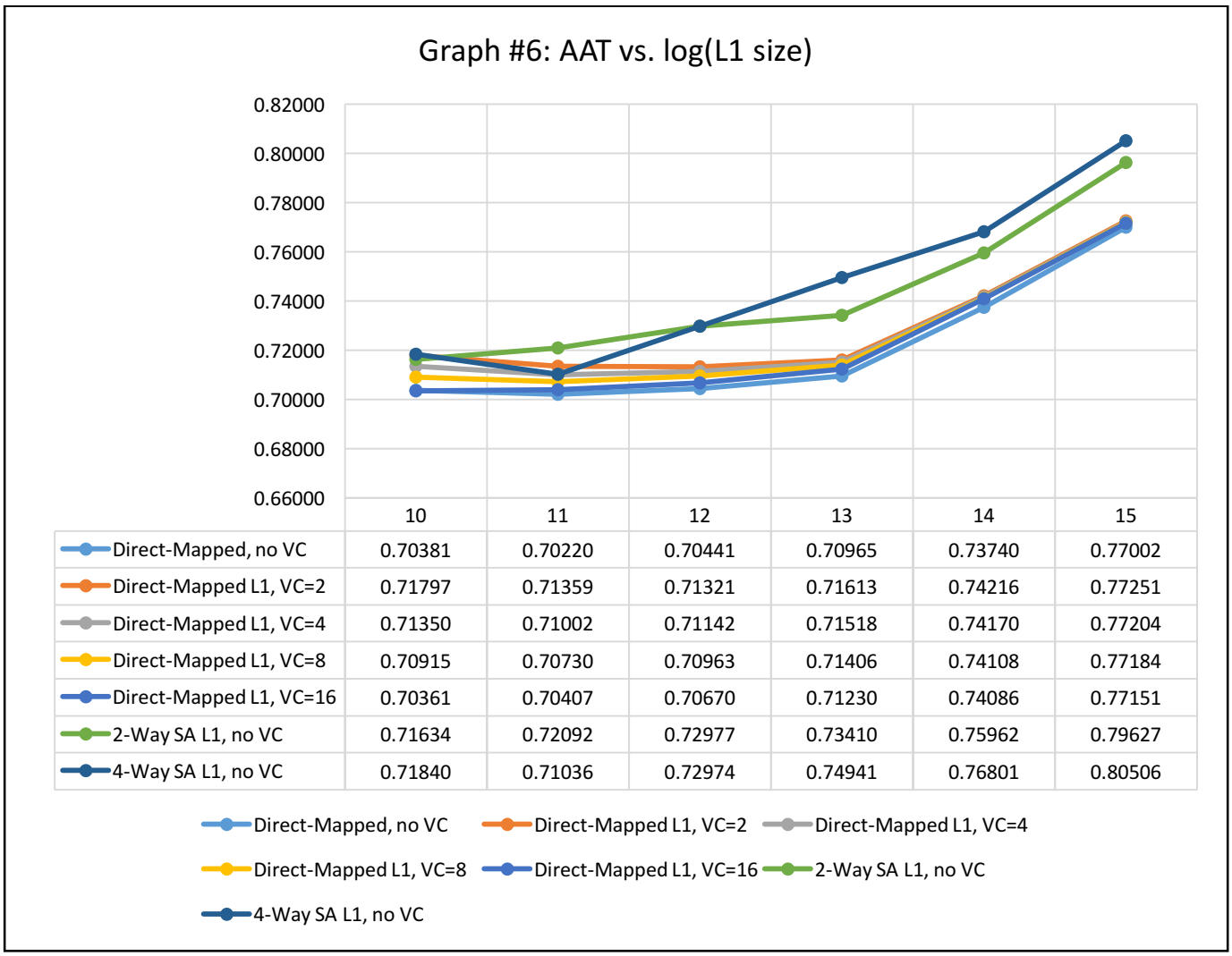
2. Which memory hierarchy configuration has the smallest total area, that yields an AAT within 5% of the best AAT?

Since the lowest AAT is 0.710 ns, thus we consider configurations whose AAT values are lesser than 0.7455 ns. Comparing the total cache areas for these memory configurations, the minimum area is found to be for a 4-way set associative **1KB L1 cache along with a 32KB 8-way set associative L2 cache**. The area for this configuration is **0.257 mm^2**. Also, the AAT is **0.718 ns**, which is within 1.1% of the lowest AAT. Here, the trade-off is between greater area needed for a bigger cache and the decrease in miss rate observed on account of the greater size. The optimal-area configuration gives a reasonably good AAT for the least chip area.

Graph #6: AAT vs. $\log_2$(L1 SIZE), with Victim Cache

Cache Configuration:
Blocksize = 32
L1 Cache Size = 1KB, 2KB, 4KB, …, 32KB
L1 Cache Associativity: Direct-mapped, 4-way, 8-way, 16-way set associative
L2 Cache Size = 64KB
L2 Cache Associativity: 8-way set associative
Victim Cache: No. of entries is varied – 0, 2, 4, 8, 16

The graph shows the effect of adding a Victim Cache to augment the L1 cache. By varying the size and associativity of L1 cache and the number of memory blocks in the fully-associative victim cache, the effect on the Average Access Time (AAT) is studied in detail. A fixed L2 cache of size 64KB with 8-way set associativity is included in the memory hierarchy.



	10	11	12	13	14	15
Direct-Mapped, no VC	0.70381	0.70220	0.70441	0.70965	0.73740	0.77002
Direct-Mapped L1, VC=2	0.71797	0.71359	0.71321	0.71613	0.74216	0.77251
Direct-Mapped L1, VC=4	0.71350	0.71002	0.71142	0.71518	0.74170	0.77204
Direct-Mapped L1, VC=8	0.70915	0.70730	0.70963	0.71406	0.74108	0.77184
Direct-Mapped L1, VC=16	0.70361	0.70407	0.70670	0.71230	0.74086	0.77151
2-Way SA L1, no VC	0.71634	0.72092	0.72977	0.73410	0.75962	0.79627
4-Way SA L1, no VC	0.71840	0.71036	0.72974	0.74941	0.76801	0.80506

1. Trends in the graph:
Comparing the miss rates from the address trace data and the AAT's from the graph, we observe that the performance of L1 cache augmented with a Victim Cache is better than that of a set-associative L1. With a smaller cache size, the AAT for cache with Victim Cache is comparable to that of a set-associative cache. However, as the size of the cache is increased, the Victim Cache configuration outperforms the set-associative configuration.

The performance of a **1KB 2-way set associative L1 cache** is thus comparable to a 1KB L1 cache with a 2-entry or a 4-entry Victim Cache. As the size of the L1 cache increases, the Victim Cache configuration performs better than the set associative cache.

2. Which memory hierarchy configuration yields the best AAT?
The lowest AAT as seen from the graph is **0.702 ns**, for **a direct-mapped 2KB L1 cache with no Victim Cache**. Although the miss rate for this configuration is not the least among the various configurations, the overall AAT is lower because of no additional delay on account of VC access. Other comparable low AAT values are obtained with a 1KB, 2KB and 4KB L1 cache with 8 VC blocks. These configurations have a sufficiently low miss rate to offset the extra hit time due to presence of Victim Cache.

3. Which memory hierarchy has the smallest total area, that yields an AAT within 5% of the best AAT?
The best observed AAT is **0.702 ns**. Looking at total cache areas for configuration with an AAT within 5%, *i.e.* AAT upto 0.737 ns, the smallest total area is found to be for a **1KB 2-way set associative L1 cache with no Victim Cache, with 64KB 8-way set associative L2 Cache**. The area is found to be **0.369 mm^2**. The AAT for this configuration is **0.716 ns**, which is within 2% of the lowest AAT. The trade-off here is between the extra area cost added to the memory hierarchy due to the presence of Victim Cache, and the decrease in the miss rate because of the presence of Victim Cache. The lowest area is found to be for a direct-mapped cache with no VC, which has lesser area since Victim Cache is absent and also a reasonably low AAT at the same time.