NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2018 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

SHREYAS SRINIVASAN

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

Student's electronic signature: Shreyas Srinivasan

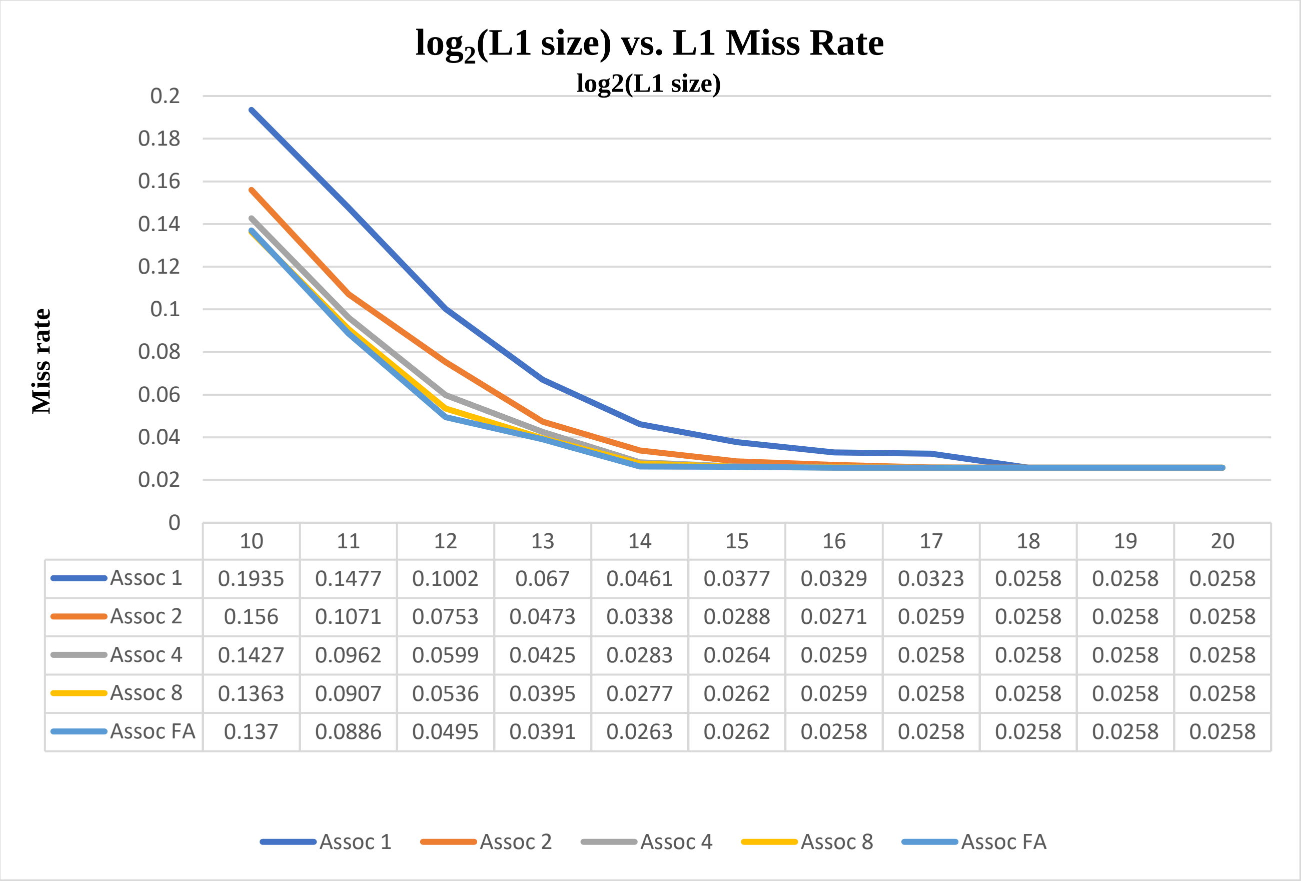
(sign by typing your name)

Course number: ECE 563

(463 or 563 ?)

Graph 1:

- L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
- Victim Cache: None.
- L2 cache: None
- Measuring L1 Miss rate against L1 size



	10	11	12	13	14	15	16	17	18	19	20
Assoc 1	0.1935	0.1477	0.1002	0.067	0.0461	0.0377	0.0329	0.0323	0.0258	0.0258	0.0258
Assoc 2	0.156	0.1071	0.0753	0.0473	0.0338	0.0288	0.0271	0.0259	0.0258	0.0258	0.0258
Assoc 4	0.1427	0.0962	0.0599	0.0425	0.0283	0.0264	0.0259	0.0258	0.0258	0.0258	0.0258
Assoc 8	0.1363	0.0907	0.0536	0.0395	0.0277	0.0262	0.0259	0.0258	0.0258	0.0258	0.0258
Assoc FA	0.137	0.0886	0.0495	0.0391	0.0263	0.0262	0.0258	0.0258	0.0258	0.0258	0.0258

Discussions:

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?

A: For a given associativity, as the cache size increases, miss rate decreases since more blocks can be accommodated and reaches a point beyond which it remains constant.

For a given cache size, as the associativity increases, miss rate decreases since more blocks can be accommodated within a set reducing the conflict miss rate. The miss rate then reaches a point beyond which it remains constant.

2. Estimate the compulsory miss rate from the graph.

A: Compulsory miss rate = 0.0258 (Miss rate value which remains constant beyond a certain point which cannot be reduced)

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

A: Conflict miss rate can be found by subtracting the miss rate achieved for a particular associativity with the miss rate of a fully associative cache. A fully-associative cache has no conflict misses. It has only compulsory & capacity misses.

Eg. For L1 size: 1kb, Block size = 32, Assoc = 1 => Miss rate = 0.1935

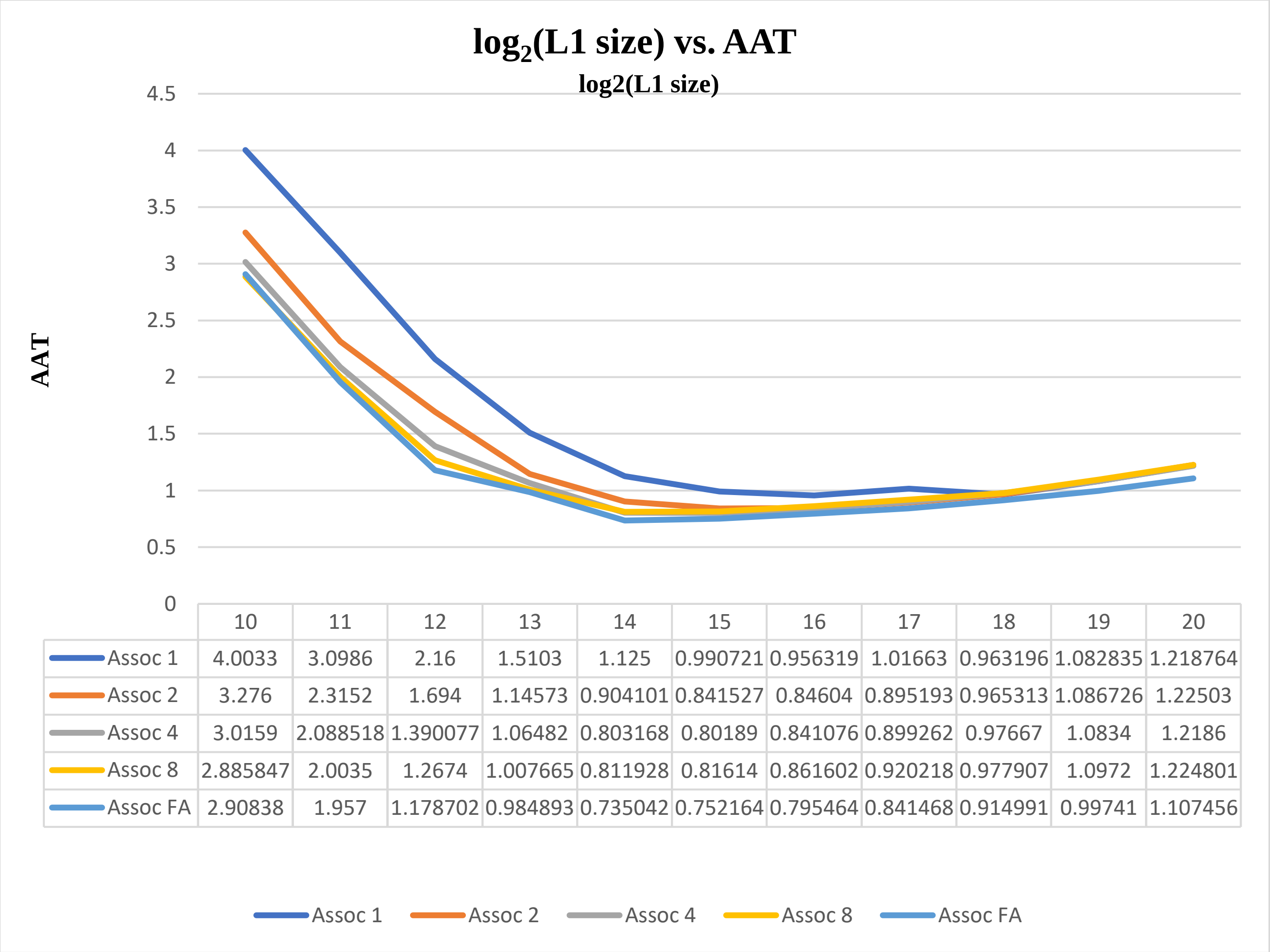
Whereas, for the same configuration with a fully-associative cache,

Miss rate = 0.137

Hence, the conflict miss rate for that original cache = 0.1935 – 0.137 = 0.0565

Graph 2:

- L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
- Victim Cache: None.
- L2 cache: None
- Measuring AAT against L1 size



	10	11	12	13	14	15	16	17	18	19	20
Assoc 1	4.0033	3.0986	2.16	1.5103	1.125	0.990721	0.956319	1.01663	0.963196	1.082835	1.218764
Assoc 2	3.276	2.3152	1.694	1.14573	0.904101	0.841527	0.84604	0.895193	0.965313	1.086726	1.22503
Assoc 4	3.0159	2.088518	1.390077	1.06482	0.803168	0.80189	0.841076	0.899262	0.97667	1.0834	1.2186
Assoc 8	2.885847	2.0035	1.2674	1.007665	0.811928	0.81614	0.861602	0.920218	0.977907	1.0972	1.224801
Assoc FA	2.90838	1.957	1.178702	0.984893	0.735042	0.752164	0.795464	0.841468	0.914991	0.99741	1.107456

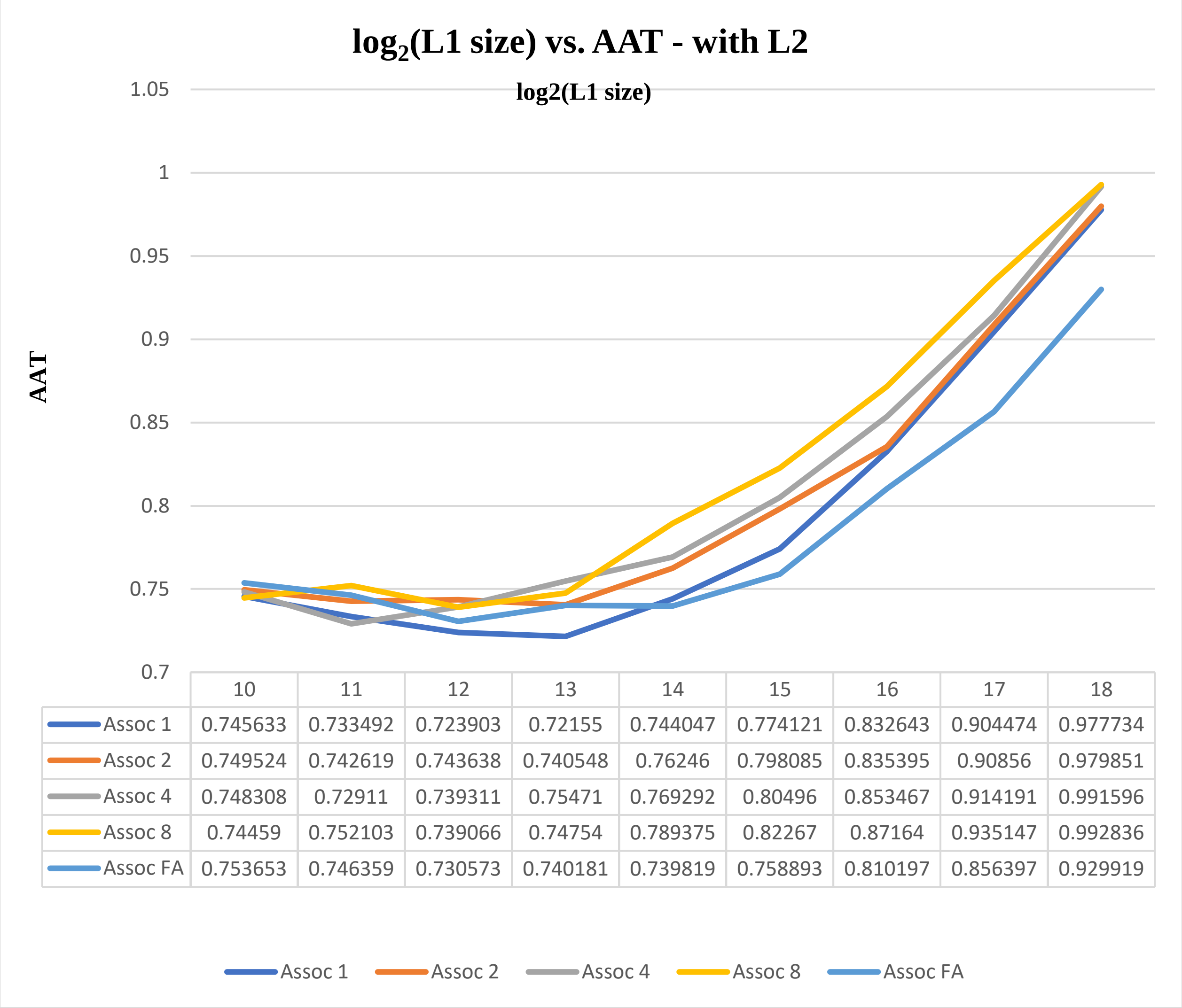
Discussions:

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

A: The cache configuration: L1 size = 16kb, Block size = 32, Fully Associative yields the best(lowest) AAT.

Graph 3:

- L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
- Victim Cache: None.
- L2 cache: 512kb 8-way set associative
- Varying L1 cache size only between 1kb and 256kb
- Measuring AAT against L1 size



	10	11	12	13	14	15	16	17	18
Assoc 1	0.745633	0.733492	0.723903	0.72155	0.744047	0.774121	0.832643	0.904474	0.977734
Assoc 2	0.749524	0.742619	0.743638	0.740548	0.76246	0.798085	0.835395	0.90856	0.979851
Assoc 4	0.748308	0.72911	0.739311	0.75471	0.769292	0.80496	0.853467	0.914191	0.991596
Assoc 8	0.74459	0.752103	0.739066	0.74754	0.789375	0.82267	0.87164	0.935147	0.992836
Assoc FA	0.753653	0.746359	0.730573	0.740181	0.739819	0.758893	0.810197	0.856397	0.929919

Discussions:

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%)?

A: The lowest AAT value found in Graph 2 is 0.735042.

Hence the caches which must be within 5% of the lowest AAT value must have values in the range (0.6983 – 0.7717)

The cache configurations which satisfy the condition are:

L1 size: 1kb Assoc = 1,2,4,8, FA Block size = 32

L1 size: 2kb Assoc = 1,2,4,8, FA Block size = 32

L1 size: 4kb Assoc = 1,2,4,8, FA Block size = 32

L1 size: 8kb Assoc = 1,2,4,8, FA Block size = 32

L1 size: 16kb Assoc = 1,2,4, FA Block size = 32

L1 size: 32kb Assoc = FA Block size = 32

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?

A: The cache configuration:

L1 size: 8kb Assoc = 1 Block size = 32 achieves the lowest AAT with a value of 0.72155 which is 0.013492 lower than that achieved in Graph 2 without L2 cache.

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

A: Total area for optimal-AAT configuration with L2 = 0.053293238 + 2.640142073 = 2.6934353 mm^2

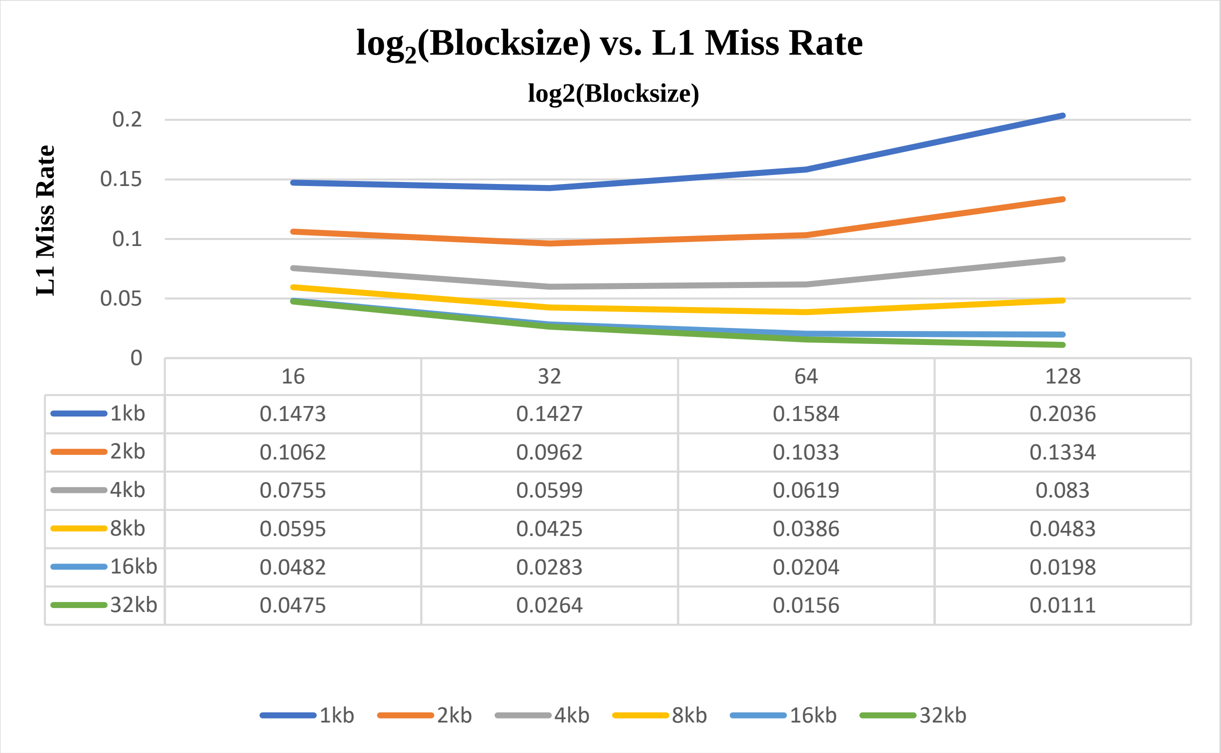
Total area for optimal-AAT configuration without L2 = 0.063446019 mm^2

Hence, the difference in area for lowering AAT, with L2 cache is

2.62998928 mm^2

Graph 4:

- L1 cache: SIZE is varied, BLOCKSIZE is varied, ASSOC = 4.
- Victim Cache: None.
- L2 cache: None.
- Measuring L1 Miss Rate against Blocksize



	16	32	64	128
1kb	0.1473	0.1427	0.1584	0.2036
2kb	0.1062	0.0962	0.1033	0.1334
4kb	0.0755	0.0599	0.0619	0.083
8kb	0.0595	0.0425	0.0386	0.0483
16kb	0.0482	0.0283	0.0204	0.0198
32kb	0.0475	0.0264	0.0156	0.0111

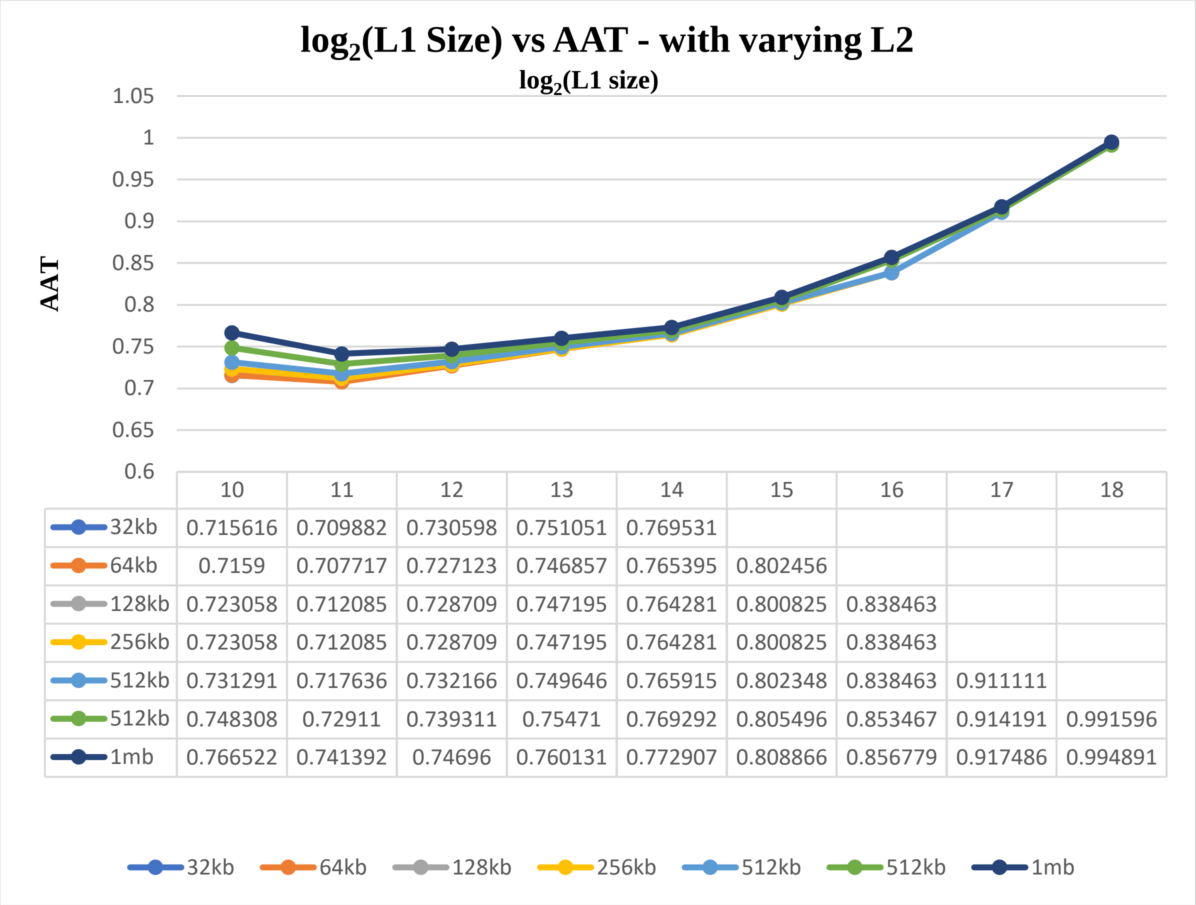
Discussions:

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 trade-off 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?

A: Smaller caches prefer larger block sizes while larger caches prefer smaller block sizes. As, block size is increased from 16 to 128, the trade-off between exploiting more spatial locality versus increasing cache pollution is evident in the graph. For larger cache size, the configuration with lowest block size yields the lowest miss rate but is the opposite in the case of a smaller cache size. This is due to cache pollution where large data is loaded into cache, making the useful data to be evicted.

Graph 5:

- L1 cache: SIZE is varied, BLOCKSIZE = 32, ASSOC = 4.
- Victim Cache: None.
- L2 cache: SIZE is varied, BLOCKSIZE = 32, ASSOC = 8.
- Measuring AAT against L1 size



	10	11	12	13	14	15	16	17	18
32kb	0.715616	0.709882	0.730598	0.751051	0.769531				
64kb	0.7159	0.707717	0.727123	0.746857	0.765395	0.802456			
128kb	0.723058	0.712085	0.728709	0.747195	0.764281	0.800825	0.838463		
256kb	0.723058	0.712085	0.728709	0.747195	0.764281	0.800825	0.838463		
512kb	0.731291	0.717636	0.732166	0.749646	0.765915	0.802348	0.838463	0.911111	
512kb	0.748308	0.72911	0.739311	0.75471	0.769292	0.805496	0.853467	0.914191	0.991596
1mb	0.766522	0.741392	0.74696	0.760131	0.772907	0.808866	0.856779	0.917486	0.994891

Discussions:

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

A: The cache configuration: L1 size = 2kb Assoc = 4 Block size = 32

L2 size = 64kb Assoc = 8 Block size = 32

yields the best(lowest) AAT.

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

A: The lowest AAT is 0.707717.

The range of values within 5% of lowest AAT is (0.74310285 – 0.67233115)

The cache configuration:

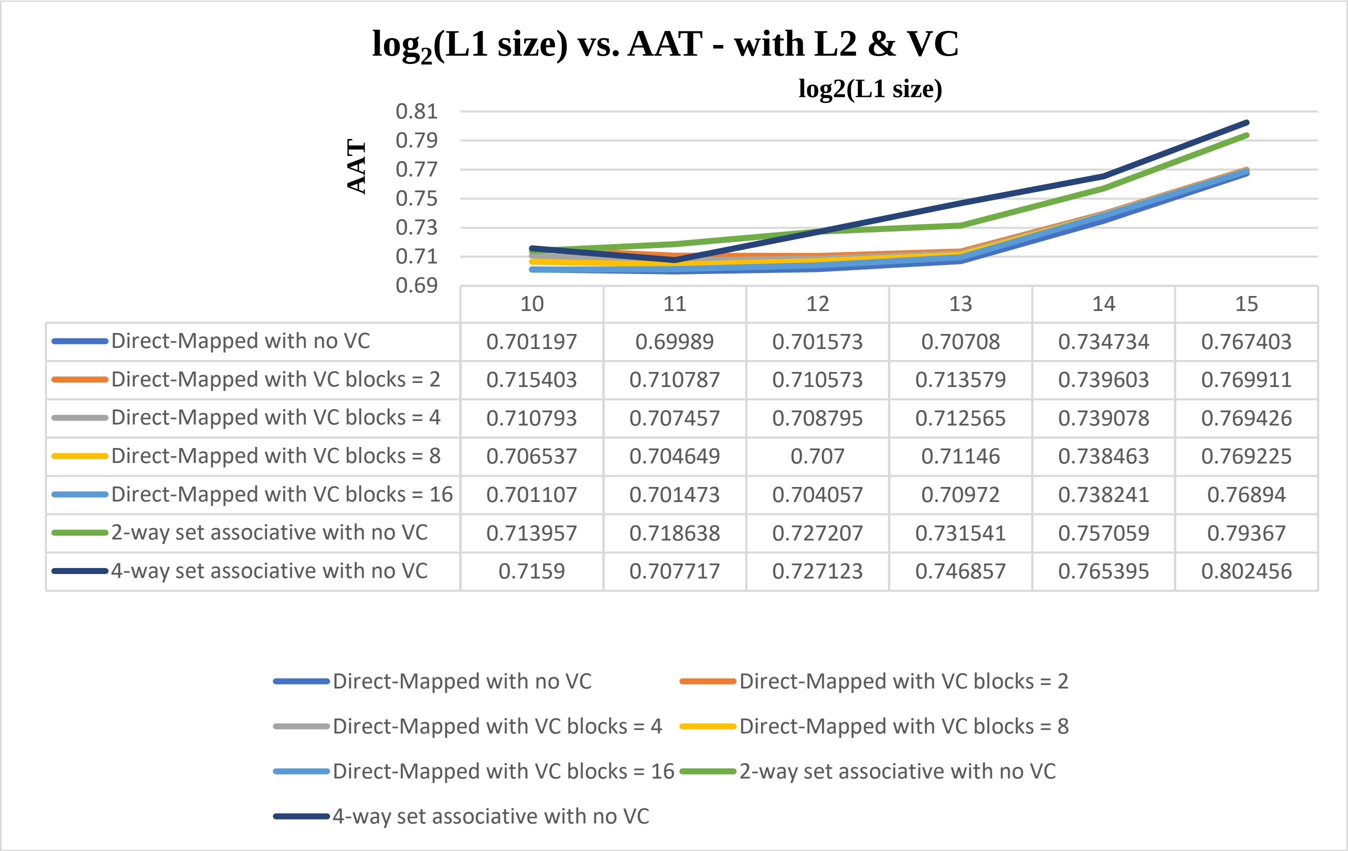
L1 size = 1kb Assoc = 4 Block size = 32

L2 size = 32kb Assoc = 8 Block size = 32

yields the lowest area of 0.257285583 mm^2 which falls within the 5% of the best AAT.

Graph 6:

- L1 cache: SIZE is varied, ASSOC is varied, BLOCKSIZE = 32.
- Victim Cache: # entries is varied.
- L2 cache: SIZE = 64KB, ASSOC = 8, BLOCKSIZE = 32.
- Measuring AAT against L1 size



	10	11	12	13	14	15
Direct-Mapped with no VC	0.701197	0.69989	0.701573	0.70708	0.734734	0.767403
Direct-Mapped with VC blocks = 2	0.715403	0.710787	0.710573	0.713579	0.739603	0.769911
Direct-Mapped with VC blocks = 4	0.710793	0.707457	0.708795	0.712565	0.739078	0.769426
Direct-Mapped with VC blocks = 8	0.706537	0.704649	0.707	0.71146	0.738463	0.769225
Direct-Mapped with VC blocks = 16	0.701107	0.701473	0.704057	0.70972	0.738241	0.76894
2-way set associative with no VC	0.713957	0.718638	0.727207	0.731541	0.757059	0.79367
4-way set associative with no VC	0.7159	0.707717	0.727123	0.746857	0.765395	0.802456

Discussions:

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?

A: AAT increases if a victim cache is added with L1 cache since we need to search both L1 and VC. Also, as the number of entries in VC is increased, the AAT decreases, as the number of L2 searches decreases in case of VC hit. Thus, direct mapped cache with Victim cache provides better performance than a 2-way set-associative cache without VC.

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

A: The cache configuration: Block size = 32 L1 size = 2kb Direct-mapped

Block size = 32 L2 size = 64kb Assoc = 8

No Victim Cache

yields the best(lowest) AAT.

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

A: The lowest AAT value is 0.69989

The range of values within 5% of lowest AAT is (0.7348445 – 0.6648955)

The cache configuration:

L1 size = 1kb Assoc = 1(Direct-mapped) with no VC yields the lowest area within 5% of the best AAT.