NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

<< Mohankumar Nekkarakalaya >>

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

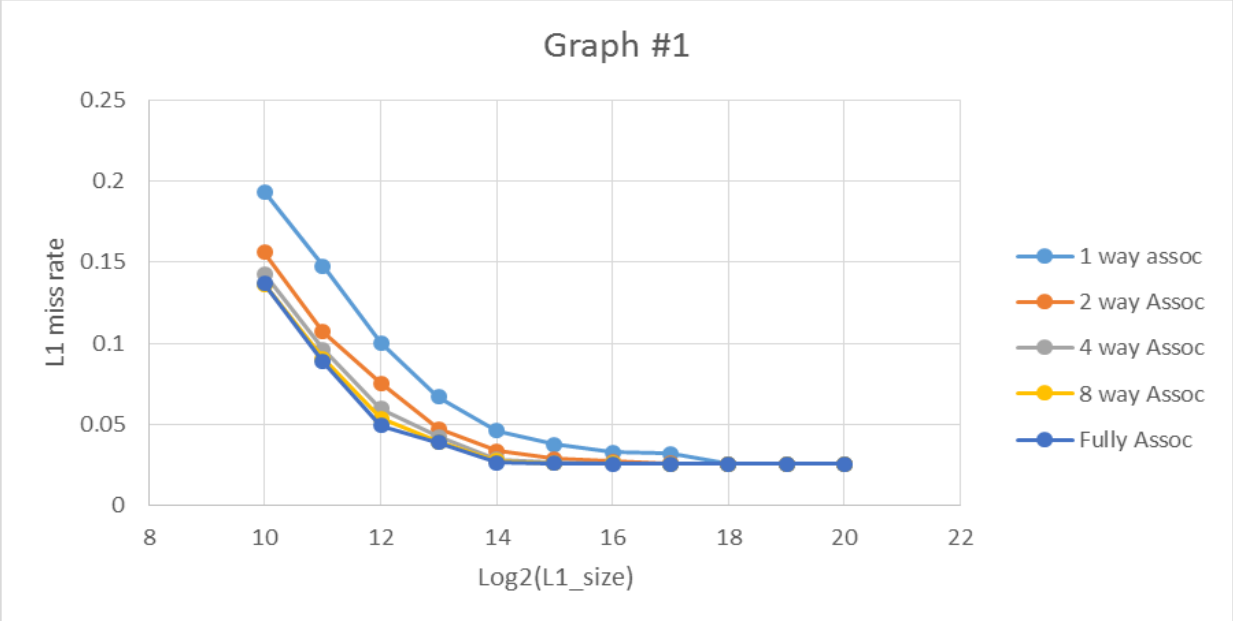
Student's electronic signature: Mohankumar Nekkarakalaya
(sign by typing your name)

Course number: 521
(463 or 521 ?)

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 Miss Rate				
Log2(size)	1 way Assoc	2 way Assoc	4 way Assoc	8 way Assoc	Fully Assoc
10	0.1935	0.156	0.1427	0.1363	0.137
11	0.1477	0.1071	0.0962	0.0907	0.0886
12	0.1002	0.0753	0.0599	0.0536	0.0495
13	0.067	0.0473	0.0425	0.0395	0.0391
14	0.0461	0.0338	0.0283	0.0277	0.0263
15	0.0377	0.0288	0.0264	0.0262	0.0262
16	0.0329	0.0271	0.0259	0.0259	0.0258
17	0.0323	0.0259	0.0258	0.0258	0.0258
18	0.0258	0.0258	0.0258	0.0258	0.0258
19	0.0258	0.0258	0.0258	0.0258	0.0258
20	0.0258	0.0258	0.0258	0.0258	0.0258



1. Increasing cache size for a given associativity reduces the miss rate exponentially upto some point (L1 size = 256 KB) after which cache size doesn't reduce the miss rate. This will be due to the presence of compulsory misses which cannot be removed.

For a given cache size increasing associativity reduces the miss rate as the conflict misses reduce. This is more pronounced when the cache size is smaller rather than large. Because in large cache sizes the conflict misses reduce due to blocks mapped to different sets rather than same which is common in lower cache size.

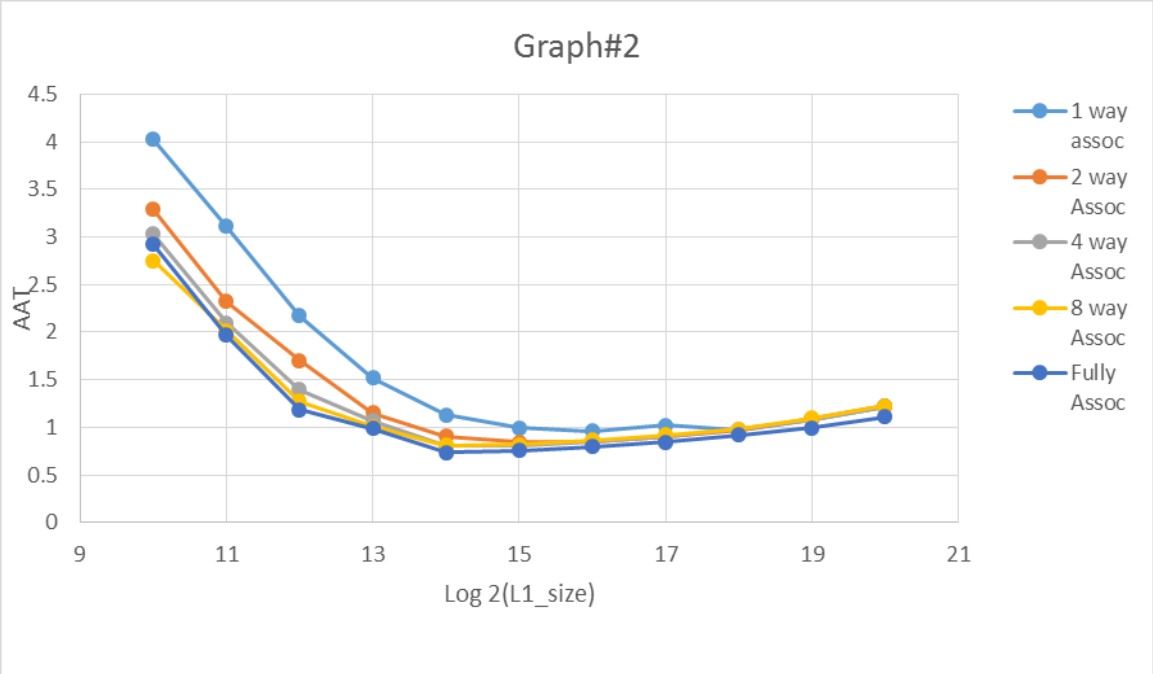
2. According to the fully associative graph which doesn't include conflict misses, the miss rate saturates at **0.0258**, which is nothing but compulsory miss rate.

3. The difference between the miss rates of fully associative cache and the miss rate of a set associative cache gives the conflict miss rate.

	Conflict Miss Rate			
Log2(size)	1 way assoc	2 way Assoc	4 way Assoc	8 way Assoc
10	0.0565	0.019	0.0057	-0.0007
11	0.0591	0.0185	0.0076	0.0021
12	0.0507	0.0258	0.0104	0.0041
13	0.0279	0.0082	0.0034	0.0004
14	0.0198	0.0075	0.002	0.0014
15	0.0115	0.0026	0.0002	0
16	0.0071	0.0013	0.0001	0
17	0.0065	0.0001	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	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.

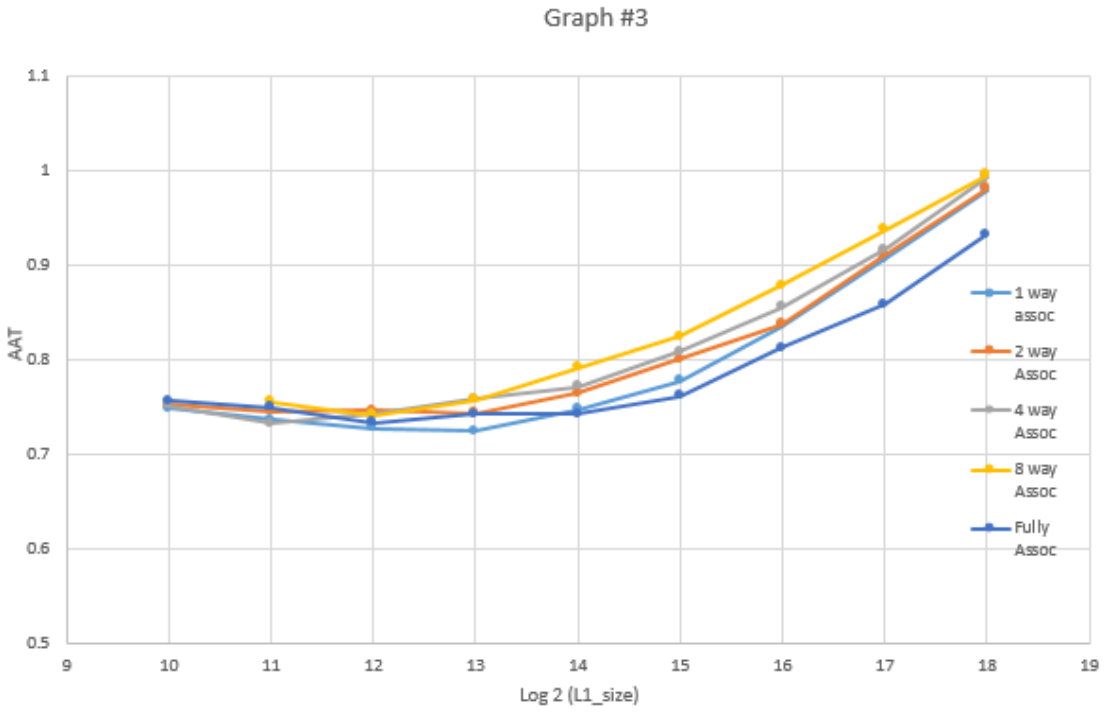
	AAT				
Log2(size)	1 way Assoc	2 way Assoc	4 way Assoc	8 way Assoc	Fully Assoc
10	4.023497	3.291529	3.02936	2.75326	2.922884
11	3.11263	2.325111	2.097736	2.012826	1.966235
12	2.171045	1.702191	1.395665	1.271785	1.182848
13	1.51723	1.149655	1.069673	1.010811	0.988401
14	1.129637	0.906677	0.805596	0.813894	0.736868
15	0.994893	0.844206	0.80453	0.817751	0.75398
16	0.959207	0.848147	0.842661	0.864393	0.797441
17	1.01926	0.897783	0.90144	0.922396	0.843646
18	0.964972	0.967089	0.978845	0.980085	0.917169
19	1.084611	1.088904	1.085578	1.099337	0.996888
20	1.22054	1.227206	1.220767	1.226979	1.109634



A fully associative 16 KB L1 cache gives the lowest AAT of **0.7369ns**

GRAPH #3 (total number of simulations: 45)

	AAT				
Log2(L1_size)	1 way assoc	2 way Assoc	4 way Assoc	8 way Assoc	Fully Assoc
10	0.748486	0.752048	0.750777		0.756349
11	0.73601	0.744998	0.731488	0.754737	0.749266
12	0.726736	0.746391	0.741698	0.741168	0.732716
13	0.724168	0.742651	0.757714	0.756778	0.742469
14	0.74674	0.764405	0.771481	0.791189	0.741608
15	0.776957	0.800471	0.808062	0.82422	0.76066
16	0.834879	0.837375	0.85502	0.877955	0.81215
17	0.90653	0.911136	0.916357	0.937313	0.858563
18	0.979472	0.981589	0.993762	0.995002	0.932086



	AAT's within 5% of best AAT in graph #2				
Log2(L1_size)	1 way assoc	2 way Assoc	4 way Assoc	8 way Assoc	Fully Assoc
10	TRUE	TRUE	TRUE	FALSE	TRUE
11	TRUE	TRUE	TRUE	TRUE	TRUE
12	TRUE	TRUE	TRUE	TRUE	TRUE
13	TRUE	TRUE	TRUE	TRUE	TRUE
14	TRUE	TRUE	TRUE	FALSE	TRUE
15	FALSE	FALSE	FALSE	FALSE	TRUE
16	FALSE	FALSE	FALSE	FALSE	FALSE
17	FALSE	FALSE	FALSE	FALSE	FALSE
18	FALSE	FALSE	FALSE	FALSE	FALSE

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?

   8 KB Direct mapped L1 cache with 512 KB L2 cache is the lowest AAT (0.72417 ns). This is 0.0127 ns lower than that in graph #2.

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

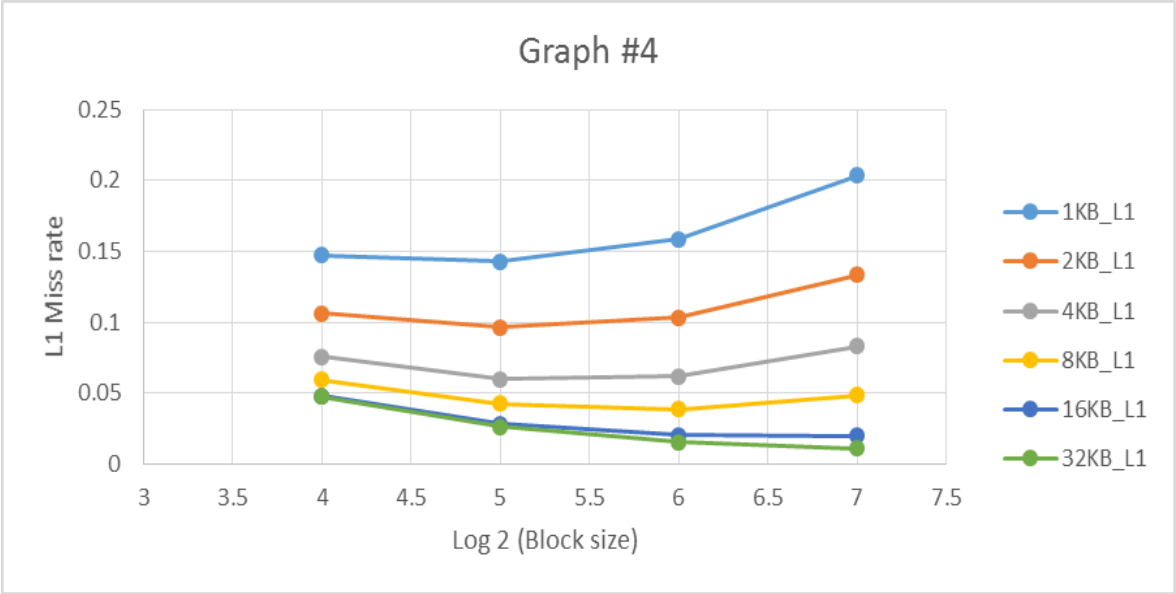
	Total area in graph #2 for optimal AAT's
Log2(L1_size)	Fully Assoc
14	0.063446
15	0.1224915

	Total area in graph #3 for optimal AAT's				
Log2(L1_size)	1 way assoc	2 way Assoc	4 way Assoc	8 way Assoc	Fully Assoc
10	2.650441	2.649614	2.655257		2.644082
11	2.656137	2.659922	2.658804	2.672861	2.647856
12	2.672838	2.6758	2.677783	2.690686	2.656809
13	2.693435	2.723898	2.708576	2.742727	2.673637
14	2.736891				2.703588

So from the table it is clear that adding an L2 cache increases the area significantly compared to L1.

GRAPH #4 (total number of simulations: 24)

	L1 Miss Rate			
Log 2 (block size)	1KB_L1	2KB_L1	4KB_L1	8KB_L1
4	0.1473	0.1062	0.0755	0.0595
5	0.1427	0.0962	0.0599	0.0425
6	0.1584	0.1033	0.0619	0.0386
7	0.2036	0.1334	0.083	0.0483



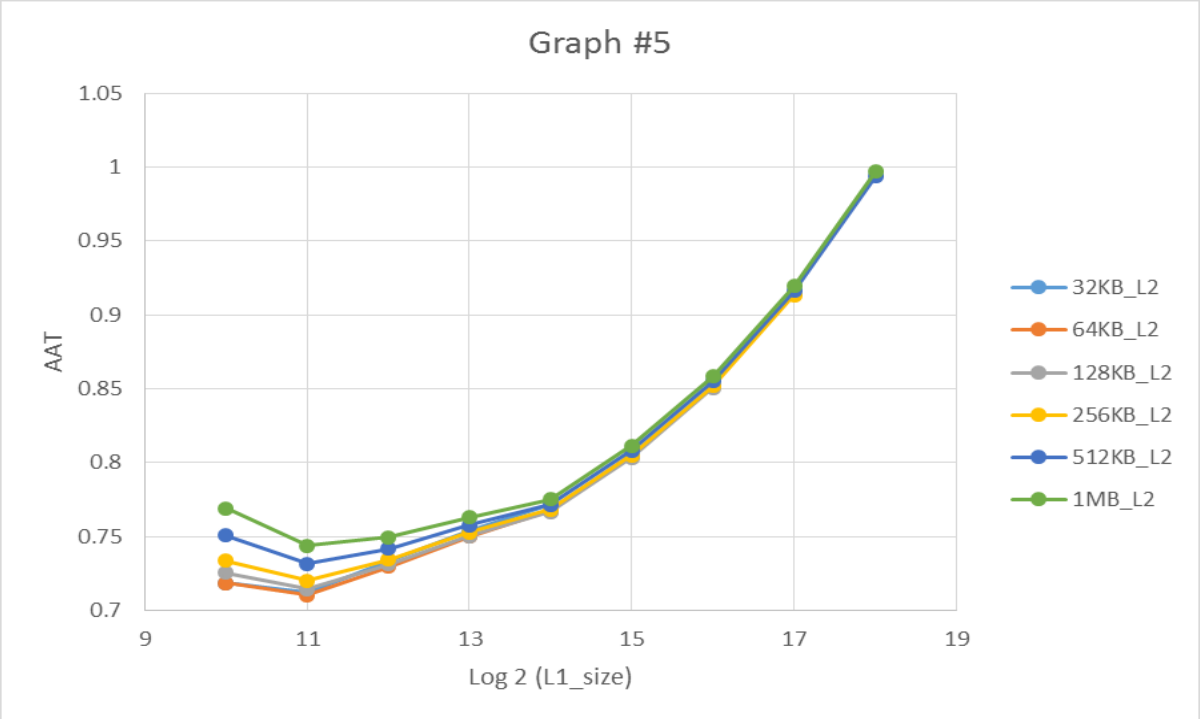
Smaller caches **do not** prefer larger block sizes. Because since the cache is small having a larger block size will lead to fewer blocks occupying the cache. Cache pollution will happen as the block size increases. This will need to frequent eviction due to capacity misses/conflict misses.

Larger cache sizes prefer larger block sizes. This is because larger Cache sizes inherently have majorly compulsory misses which will reduce when we increase the block size.

Yes in small cache sizes we can see the miss rate reduces initially from block size of 16 B to 32 B then miss rate starts increasing. This is because cache pollution exceeds the gain due to increased spatial locality. However that is not the case in larger cache sizes, as cache pollution will not be a factor as they already have a large cache size to accommodate data and they need spatial locality to reduce the miss rate.

GRAPH #5 (total number of simulations: 44)

	AAT					
Log2(L1_size)	32KB_L2	64KB_L2	128KB_L2	256KB_L2	512KB_L2	1MB_L2
10	0.718378	0.718404	0.725528	0.73376	0.750777	0.768992
11	0.712367	0.710247	0.714466	0.720016	0.731488	0.743767
12	0.733059	0.729561	0.731099	0.734555	0.741698	0.749344
13	0.754074	0.749789	0.750194	0.752645	0.757714	0.763138
14	0.771801	0.767633	0.766473	0.768106	0.771481	0.775093
15		0.805059	0.80339	0.804913	0.808062	0.811431
16			0.850437	0.851931	0.85502	0.858326
17				0.91328	0.916357	0.91965
18					0.993762	0.997055



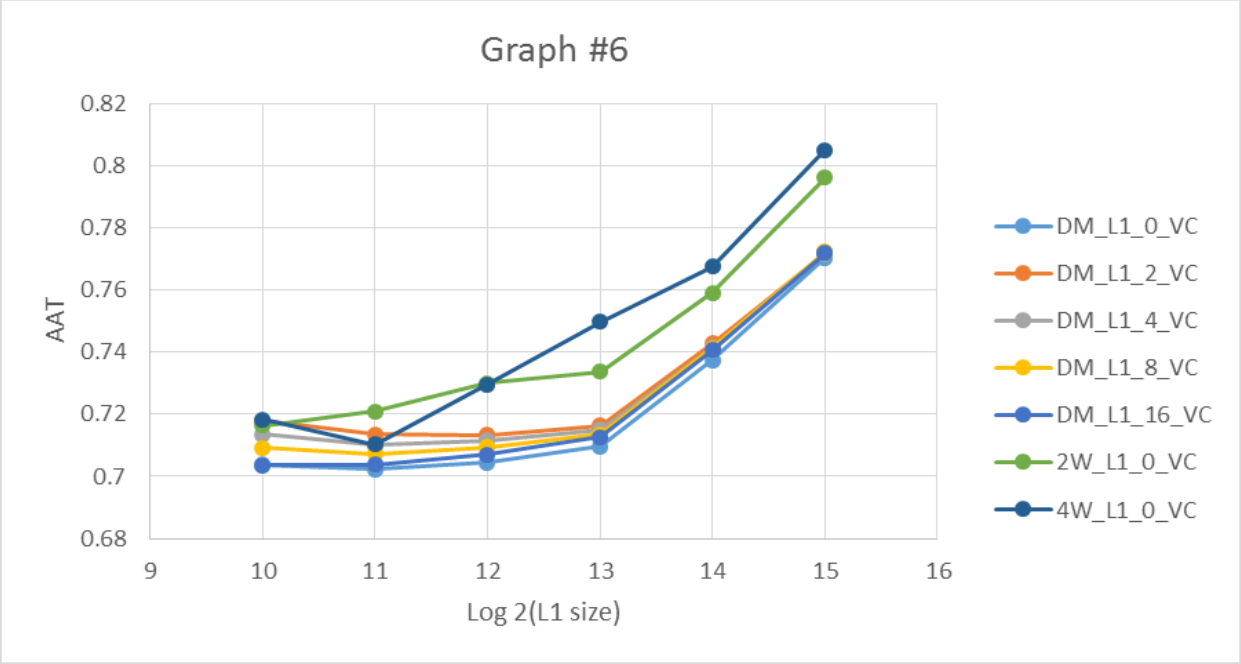
Discussion to include in your report:

1. Which memory hierarchy configuration yields the best (i.e., lowest) AAT?
   64 KB L2 cache with 2 KB L1 cache yields the best AAT (**0.71024 ns**)

2. Which memory hierarchy configuration has the smallest total area, that yields an AAT within 5% of the best AAT?
   32 KB L2 cache with 1 KB L1 cache
   Area = 0.25729 mm^2

GRAPH #6 (total number of simulations: 42)

	AAT						
Log 2 (l1 size)	DM_L1_0_VC	DM_L1_2_VC	DM_L1_4_VC	DM_L1_8_VC	DM_L1_16_VC	2W_L1_0_VC	4W_L1_0_VC
10.00000	0.70381	0.71797	0.71350	0.70915	0.70361	0.71634	0.71840
11.00000	0.70220	0.71359	0.71002	0.70730	0.70383	0.72092	0.71025
12.00000	0.70441	0.71327	0.71155	0.70935	0.70704	0.72992	0.72956
13.00000	0.70965	0.71640	0.71508	0.71353	0.71266	0.73364	0.74979
14.00000	0.73740	0.74281	0.74157	0.74137	0.74057	0.75899	0.76763
15.00000	0.77031	0.77189	0.77172	0.77234	0.77186	0.79609	0.80506



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?

   The graphs indicate that for a given direct mapped L1 cache adding a victim cache is beneficial in reducing the AAT as the cache size decreases. The 16 vc blocks is the best AAT at 1 KB L1 cache size. Also in large cache sizes the number of VC blocks don't matter as the conflict and capacity misses disappear. Also the AAT of direct mapped L1 cache is better than 2W and 4W set associative L1 cache in large cache sizes.

   According to AAT, for cache sizes equal to 1 KB and 2 KB, then direct mapped l1 cache with 2 & 4 VC blocks is comparable to 2 way set associative performance. In higher cache sizes the AAT of direct mapped cache is better than 2W and 4W set associative L1 cache.

According to miss rate the 2W set associative cache performs better than 2VC blocks.

	L1 Miss rate						
Log 2 (l1 size)	DM_L1_0_VC	DM_L1_2_VC	DM_L1_4_VC	DM_L1_8_VC	DM_L1_16_VC	2W_L1_0_VC	4W_L1_0_VC
10.00000	0.1935	0.1608	0.1473	0.1323	0.1125	0.156	0.1427
11.00000	0.1477	0.1231	0.1133	0.1032	0.0903	0.1071	0.0962
12.00000	0.1002	0.0885	0.0833	0.0767	0.065	0.0753	0.0599
13.00000	0.067	0.0601	0.0565	0.0518	0.0449	0.0473	0.0425
14.00000	0.0461	0.0417	0.0401	0.0378	0.0357	0.0338	0.0283
15.00000	0.0377	0.0343	0.0329	0.032	0.0306	0.0288	0.0264

2. Direct Mapped L1 Cache of size 2KB without VC yields the best AAT. (0.70220 ns)

3. 2 way set associative L1 cache of 1 KB with no victim cache yields the least area.

   Area = 0.36979 mm^2.

	Total Area						
Log 2 (l1 size)	DM_L1_0_VC	DM_L1_2_VC	DM_L1_4_VC	DM_L1_8_VC	DM_L1_16_VC	2W_L1_0_VC	4W_L1_0_VC
10.00000	0.37062	0.37096	0.37131	0.37194	0.37310	0.36979	0.37543
11.00000	0.37631	0.37666	0.37701	0.37763	0.37880	0.38010	0.37898
12.00000	0.39301	0.39336	0.39371	0.39433	0.39550	0.39598	0.39796
13.00000	0.41361	0.41396	0.41431	0.41493	0.41610	0.44407	0.42875
14.00000	0.45707	0.45741	0.45776	0.45839	0.45955	0.49042	0.46626
15.00000	0.57086	0.57121	0.57156	0.57218	0.57335	0.56587	0.59697