NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2013 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

Naman Muley

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

Student’s electronic signature: ______Naman Muley____________________

(sign by typing your name)

Course number: _______521____________

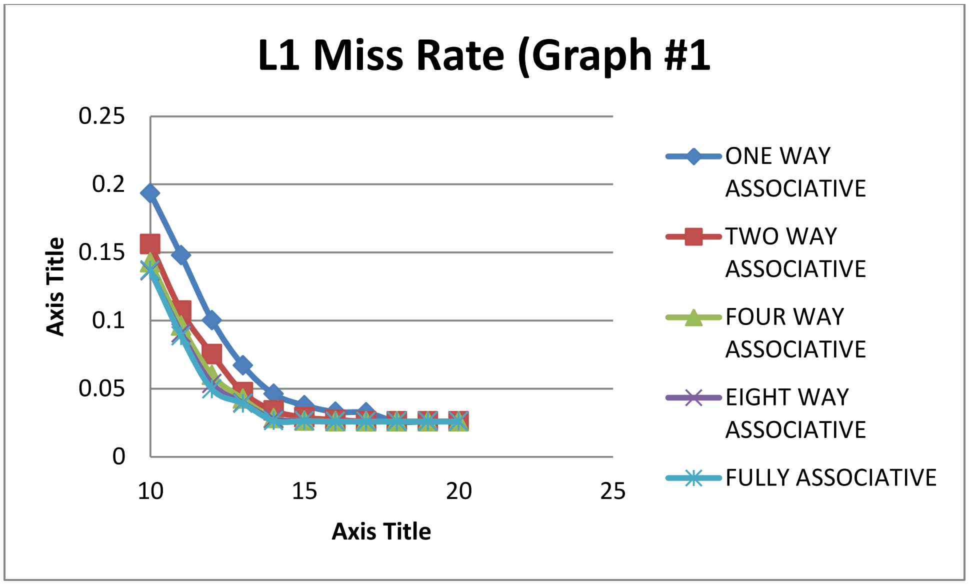
(463 or 521 ?)

Q1. Discussion to include in your report:

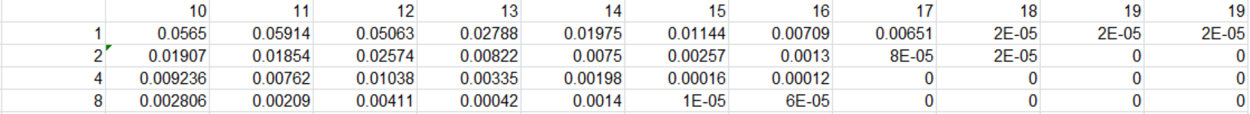
1. **Discuss trends in the graph. For a given associativity, how does increasing cache size affect miss rate? For a given cache size, what is the effect of increasing associativity?**
2. **Estimate the compulsory miss rate from the graph.**

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

Graph #1



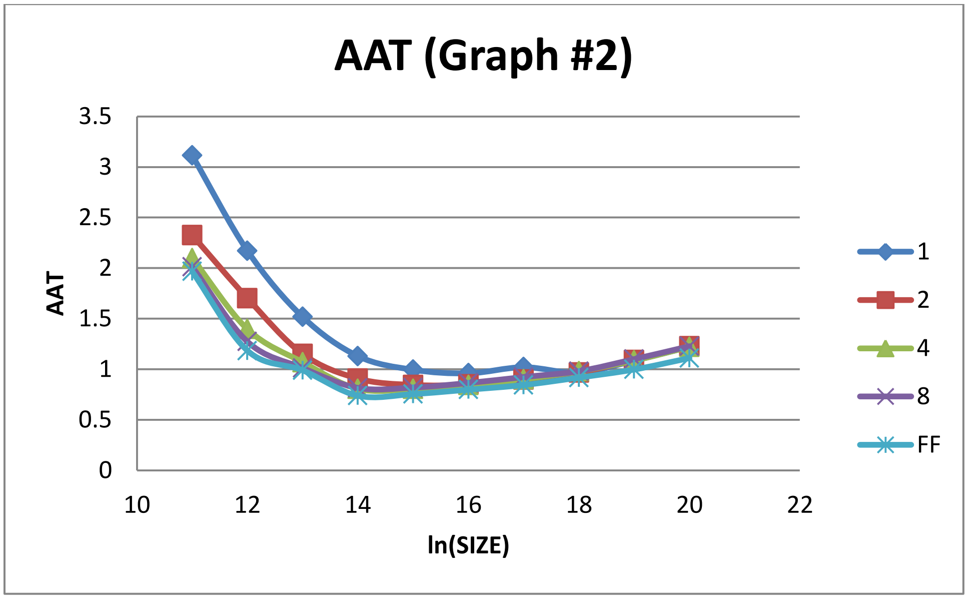
1. As you can see from each of the graphs, the increasing the cache size decreases the miss rate. That is logical as increasing the cache gives more space for keeping blocks and probability of missing decreases.
2. The miss rate generally decreases as the size increases. If the size increases too much, the conflict and capacity misses are eliminated and hence only compulsory misses remain. Here in the graph we can see that the Miss Rate is tending to 0.025. As the cache size increases. Hence, this constant Miss Rate will be the compulsory miss rate.
3. Conflict Miss Rate = Miss Rate for Given Associativity – Miss Rate when Fully Associative



	10	11	12	13	14	15	16	17	18	19	19
1	0.0565	0.05914	0.05063	0.02788	0.01975	0.01144	0.00709	0.00651	2E-05	2E-05	2E-05
2	0.01907	0.01854	0.02574	0.00822	0.0075	0.00257	0.0013	8E-05	2E-05	0	0
4	0.009236	0.00762	0.01038	0.00335	0.00198	0.00016	0.00012	0	0	0	0
8	0.002806	0.00209	0.00411	0.00042	0.0014	1E-05	6E-05	0	0	0	0

Graph # 2

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



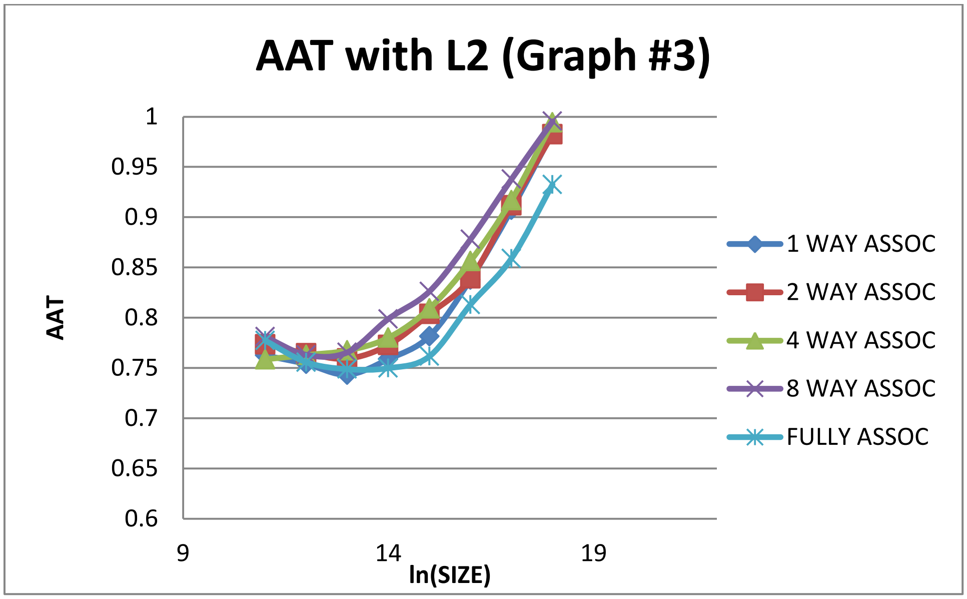
We can see from the graph that the Fully Associative L1 cache and Blocksize=32 has the best AAT for Cache Size of 2^14. The best AAT is 0.73768

Graph #3

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%)?**

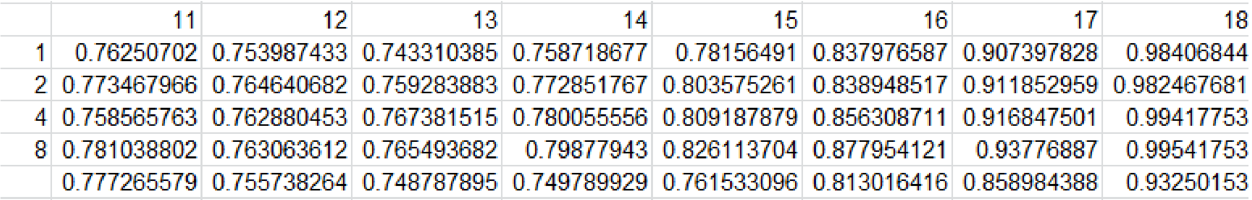
2. **With the L2 cache added to the system, which L1 cache configuration yields the best (i.e. lowest) AAT? How much lower is this optimal AAT compared to the optimal AAT in GRAPH#2?**

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



1. The best AAT from Graph #2 is 0.73768. A 5% of variation will mean values between 0.774564 and 0.700796.

According to the values shown below, 1 way associative cache with size of 2^11 gives the closest AAT



	11	12	13	14	15	16	17	18
1	0.76250702	0.753987433	0.743310385	0.758718677	0.78156491	0.837976587	0.907397828	0.98406844
2	0.773467966	0.764640682	0.759283883	0.772851767	0.803575261	0.838948517	0.911852959	0.982467681
4	0.758565763	0.762880453	0.767381515	0.780055556	0.809187879	0.856308711	0.916847501	0.99417753
8	0.781038802	0.763063612	0.765493682	0.79877943	0.826113704	0.877954121	0.93776887	0.99541753
	0.777265579	0.755738264	0.748787895	0.749789929	0.761533096	0.813016416	0.858984388	0.93250153

2. It can be seen from the graph, that the most optimal (Lowest AAT) is for direct mapped cache (1 way set associative) at the cache size of 2^13. The value is 0.7433

3. From graph 2 and its answers, For a fully associative 16 KB block the Area given in the CACTI spread sheet is 0.063446019 mm^2

   Area of graph#3 = Area of L1+ Area of L2

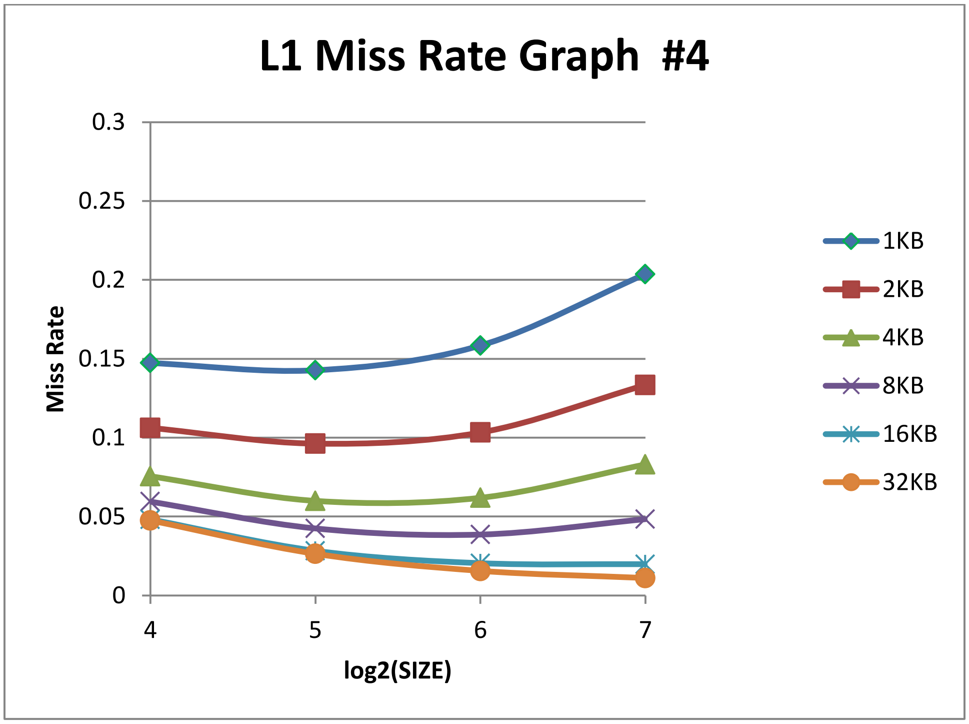
Graph#3 $= 0.053293238\ mm^2 + 2.640142073\ mm^2$

$= \underline{2.693435311\ mm^2}$

Thus we can see that graph#3 is much bigger than Graph#2 's area

Graph #4

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?

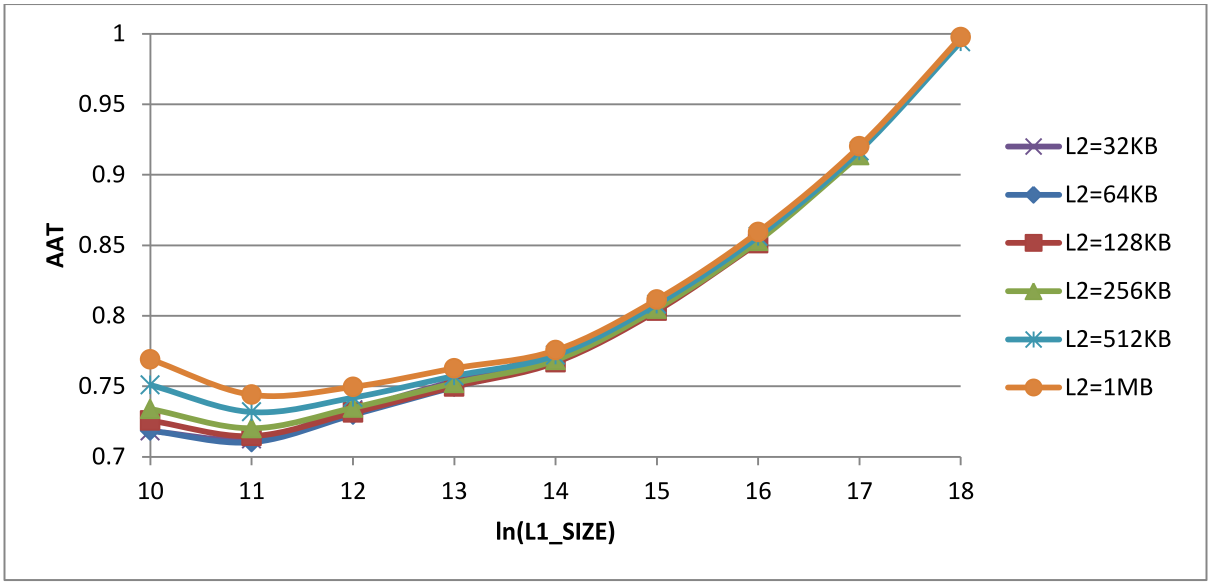


From the graph we can see that the blue curve, 1KB, smallest cache size has smaller miss rate for smaller block sizes. Also, we can see that larger caches prefer larger block sizes – The orange curve

This can be explained by cache pollution and spatial locality. Imagine if smaller cache sizes had larger block sizes, then most of the blocks would have caused cache pollution rather than helping in spatial locality. The larger cache sizes compensates for the large block sizes. Vice versa for the larger cache sizes and smaller block sizes will not be able to utilize the spatial locality offered by a cache.

The balance can be clearly seen shifting in the graph.

Graph #5



1. Which memory hierarchy configuration yields the best (i.e., lowest) AAT?
2.Which memory hierarchy configuration hasthe smallest total area, that yields an AAT within 5% of the best AAT?

1. We can see from the graph that L2 cache size = 64 KB (the deep blue color curve) and the L1 cache size= 2^11 yields the best AAT (lowest AAT).
2. The smallest cache area is of Cache L1 at 1KB and L2 at 32 KB
   The area caclulations:
   Total Area = Area of L1 + Area of L2

   = 0.015114948+0.242170635

   = 0.257285583 mm^2

Stream Buffer Study

1. Which combination of L1 SIZE, BLOCKSIZE, and L1 prefetcher (# stream buffers and stream buffer depth) yields the best (i.e., lowest) AAT?

We noted earlier that smaller cache sizes have lower AATs. We try out cache size of 1KB, 2KB.. and we find that 4KB, we find that lowest AAT comes for block size of 4KB.

We also notice with multiple passes of simulator that as stream buffer size increases, AAT decreases, so we take the maximum stream buffer size of 4x4.

The best combination thus comes out to be:

```
./sim_cache 64 4096 4 4 4 65536 8 0 0 gcc_trace.txt
```

2. **Now find two cheaper configurations – simpler L1 and/or simpler prefetchers (fewer stream buffers, smaller stream buffers) – that yield an AAT within 5% of the best AAT. The two cheaper configurations are up to you, but you need to provide an explanation of why you think these configurations are the most prudent ones. Answers may vary … just as different companies design different products based on judgment calls.**

```
./sim_cache 64 8192 2 4 4 65536 8 0 0 gcc_trace.txt
```

A simpler L1 cache – a two way set associative cache gives a very close AAT to optimal.

```
./sim_cache 64 4096 4 2 2 65536 8 0 0 gcc_trace.txt
```

A smaller stream buffer unit with only 2 stream buffers, each with 2 blocks along with an 8 way set associative cache also yields a very close AAT.