NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

<< FALAK VIJAY RAVANI >>

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

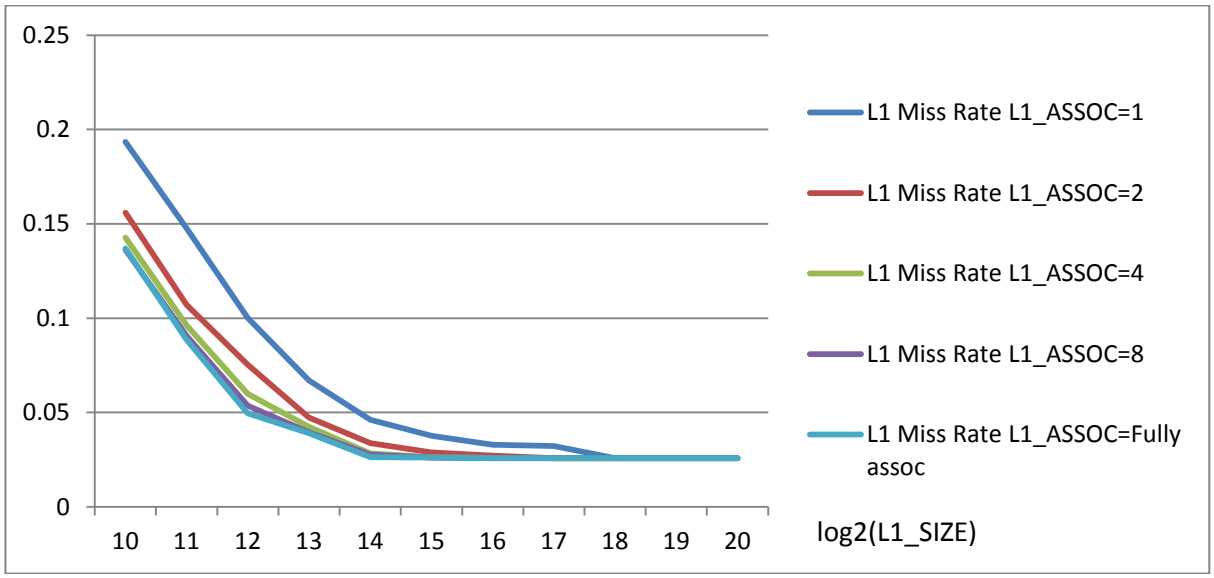
Student's electronic signature: _____Falak V Ravani_____
(sign by typing your name)

Course number: _____521_____
(463 or 521 ?)

Table1

L1_SIZE	log2(L1_SIZE)	L1 Miss Rate				
		L1_ASSOC=1	L1_ASSOC=2	L1_ASSOC=4	L1_ASSOC=8	L1_ASSOC=Fully assoc
1024	10	0.1935	0.156	0.1427	0.1363	0.137
2048	11	0.1477	0.1071	0.0962	0.0907	0.0886
4096	12	0.1002	0.0753	0.0599	0.0537	0.0495
8192	13	0.067	0.0473	0.0425	0.0395	0.0391
16384	14	0.0461	0.0338	0.0283	0.0277	0.0263
32768	15	0.0377	0.0288	0.0264	0.0262	0.0262
65536	16	0.0329	0.0271	0.026	0.0259	0.0258
131072	17	0.0323	0.0259	0.0258	0.0258	0.0258
262144	18	0.0258	0.0258	0.0258	0.0258	0.0258
524288	19	0.0258	0.0258	0.0258	0.0258	0.0258
1048576	20	0.0258	0.0258	0.0258	0.0258	0.0258

GRAPH 1 (L1 miss rate vs log2(L1_SIZE))



1) For a given associativity, increasing cache-size reduces the miss rate considerably for a while. After a certain point ("knee" in the curve), there isn't much reduction in the miss rate which is the area of "diminishing returns". For a given cache size, increasing the associativity reduces the miss rate faster initially and then tends to saturate as the conflict misses are reduced.

2) From the graph, the compulsory miss rate can be estimated to be around half of 0.05. The values in the table prove the same. We know, conflict misses can be reduced by increasing associativity. A full-associative graph has no conflict misses. Also, as we go on increasing the size of cache, we are reducing the capacity misses. Thus, after a certain point, the only misses that are left are compulsory misses which is equal to 0.0258 as per Table1.

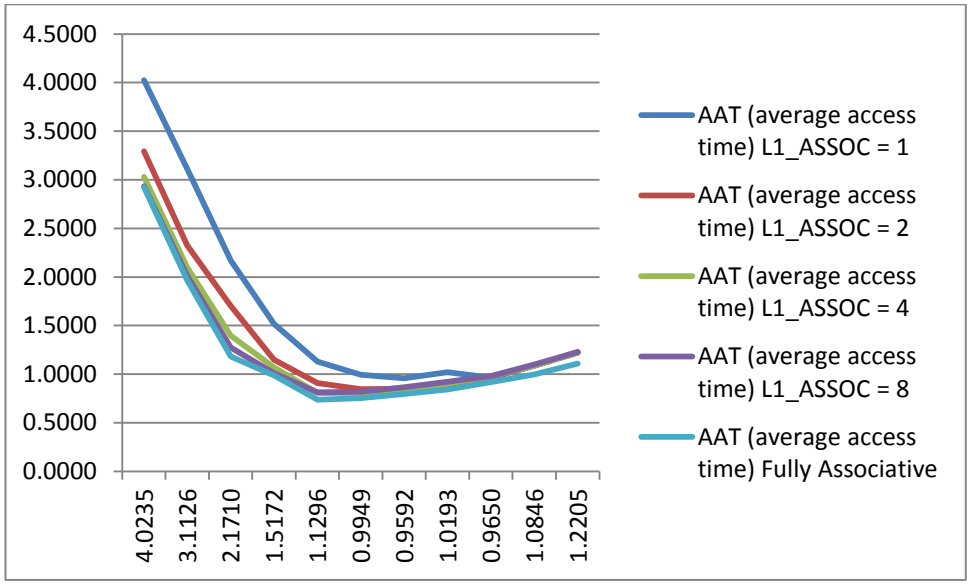
3) For each associativity, we can find the conflict miss rate by comparing the miss rate with the miss rate of the fully-associative cache of the same size.
Consider a cache of size 8kB,

For L1_ASSOC = 1, 0.067 – 0.0391 = 0.0279
For L1_ASSOC = 2, 0.0473 – 0.0391 = 0.0082
For L1_ASSOC = 4, 0.0425 – 0.0391 = 0.0034
For L1_ASSOC = 8, 0.0395 – 0.0391 = 0.0004
For L1_ASSOC = fully-associative, 0.0391 – 0.0391 = 0

Table2

log2(L1 size)	AAT (average access time)				
	L1_ASSOC = 1	L1_ASSOC = 2	L1_ASSOC = 4	L1_ASSOC = 8	Fully Associative
10	4.0235	3.2915	3.0294	2.9339	2.9229
11	3.1126	2.3251	2.0977	2.0128	1.9662
12	2.1710	1.7022	1.3957	1.2738	1.1828
13	1.5172	1.1497	1.0697	1.0108	0.9884
14	1.1296	0.9067	0.8056	0.8139	0.7369
15	0.9949	0.8442	0.8045	0.8178	0.7540
16	0.9592	0.8481	0.8447	0.8644	0.7974
17	1.0193	0.8978	0.9014	0.9224	0.8436
18	0.9650	0.9671	0.9788	0.9801	0.9172
19	1.0846	1.0889	1.0856	1.0993	0.9969
20	1.2205	1.2272	1.2208	1.2270	1.1096

GRAPH 2

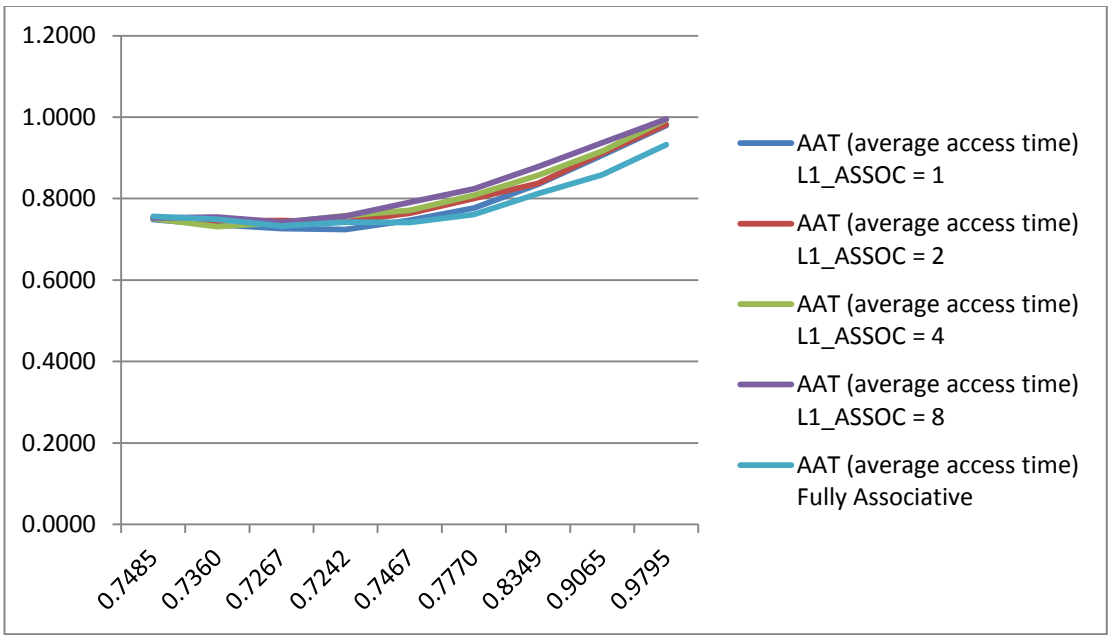


1) The configuration that yields the best AAT is –
A fully-associative, 16kB L1 cache with an AAT of 0.7369ns

Table3

	log2(L1 size)	AAT (average access time)				
		L1_ASSOC = 1	L1_ASSOC = 2	L1_ASSOC = 4	L1_ASSOC = 8	Fully Associative
1024	10	0.7485	0.7520	0.7508	0.7520	0.7563
2048	11	0.7360	0.7450	0.7315	0.7547	0.7493
4096	12	0.7267	0.7464	0.7417	0.7422	0.7327
8192	13	0.7242	0.7427	0.7577	0.7568	0.7425
16384	14	0.7467	0.7644	0.7715	0.7912	0.7416
32768	15	0.7770	0.8005	0.8081	0.8242	0.7607
65536	16	0.8349	0.8374	0.8571	0.8780	0.8121
131072	17	0.9065	0.9111	0.9164	0.9373	0.8586
262144	18	0.9795	0.9816	0.9938	0.9950	0.9321

GRAPH 3



1) The cache configurations that are close to the best AAT observed in graph2 are –
   - 1kB with L1_ASSOC = 1, 2, 4, 8 and fully-associative.
   - 2kB with L1_ASSOC = 1, 2, 4, 8 and fully-associative.
   - 4kB with L1_ASSOC = 1, 2, 4, 8 and fully-associative.
   - 8kB with L1_ASSOC = 1, 2, 4, 8 and fully-associative.
   - 16kB with L1_ASSOC = 1, 2, 4 and fully-associative.
   - 32kB with L1_ASSOC = fully-associative.

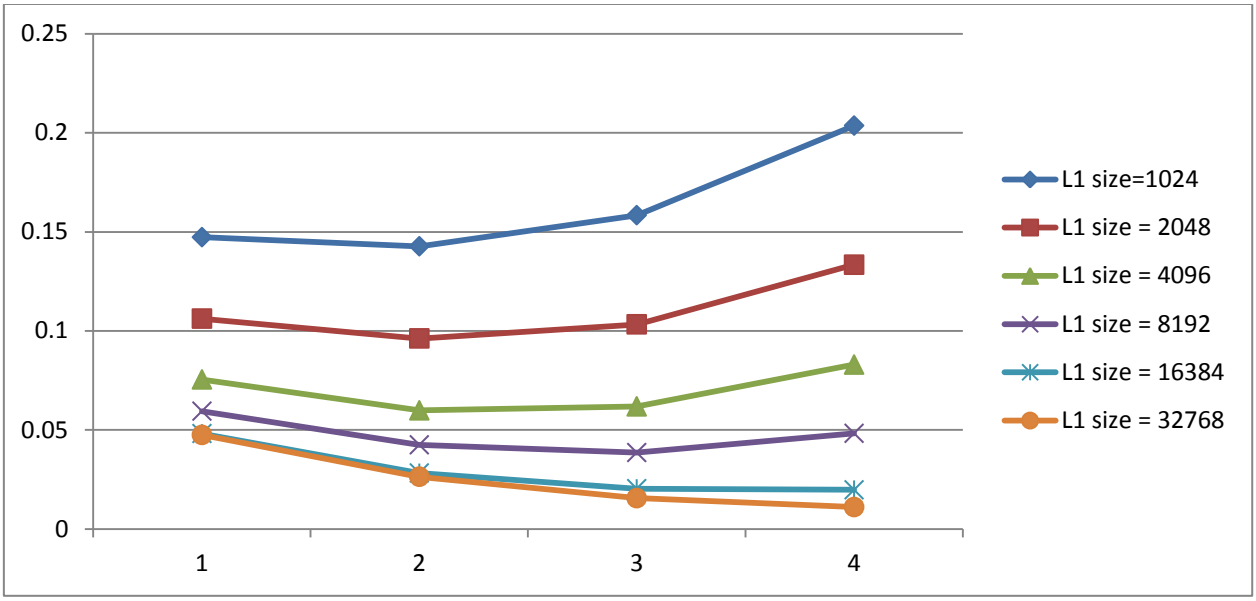
2) An 8kB, direct-mapped cache has the lowest AAT of 0.7242ns which is 1.71% lower than optimal AAT in Graph2.

3) Total area required for optimal AAT in Graph2 is 0.0634mm^2 whereas the total area required for optimal AAT in Graph3 is 0.0532mm^2 + 2.6401mm^2 which is equal to 2.6933mm^2. The area for Graph3 is 42.48 times that of Graph2.

Table4

log2(blocksize)	L1 miss rate					
	L1 size=1024	L1 size = 2048	L1 size = 4096	L1 size = 8192	L1 size = 16384	L1 size = 32768
4	0.1473	0.1062	0.0755	0.0595	0.0482	0.0475
5	0.1427	0.0962	0.0599	0.0425	0.0283	0.0264
6	0.1584	0.1033	0.0619	0.0386	0.0204	0.0156
7	0.2036	0.1334	0.083	0.0483	0.0198	0.0111

Graph4

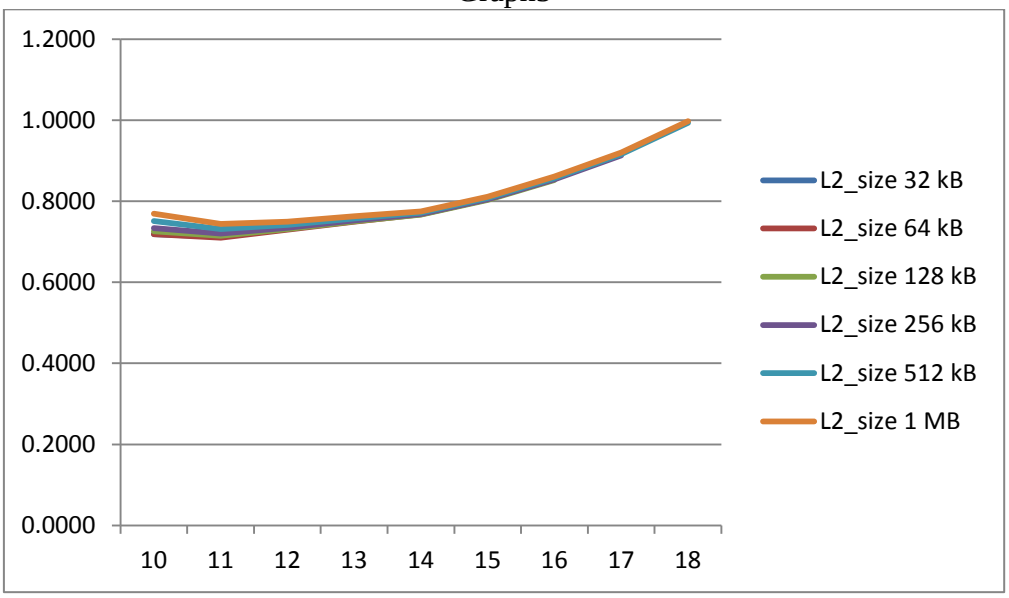


From Graph4, for a cache of size 1kB, the miss rate goes on increasing as we increase the blocksize. This is due to cache pollution. Thus, smaller caches prefer smaller blocksizes. For 32kB cache, the miss rate reduces as we increase the blocksize. As we increase blocksize, the miss rate reduces initially as we are exploiting the spatial locality. Later, the miss rate increases due to cache pollution which is clearly evident from Graph4 (for cache size 2kB, 4kB and 8kB). The lowest point i.e. the balance between spatial locality and cache pollution does shift with different cache sizes.

Table5

log2(L1 size)	AAT (average access time)					
	L2_size 32 kB	L2_size 64 kB	L2_size 128 kB	L2_size 256 kB	L2_size 512 kB	L2_size 1 MB
10	0.7184	0.7184	0.7255	0.7338	0.7508	0.7690
11	0.7124	0.7102	0.7145	0.7200	0.7315	0.7438
12	0.7331	0.7296	0.7311	0.7346	0.7417	0.7493
13	0.7541	0.7498	0.7502	0.7526	0.7577	0.7631
14	0.7718	0.7676	0.7665	0.7681	0.7715	0.7751
15		0.8051	0.8034	0.8049	0.8081	0.8114
16			0.8525	0.8540	0.8571	0.8604
17				0.9133	0.9164	0.9197
18					0.9938	0.9971

Graph5



1) The configuration that yields the lowest AAT is a L1 cache of size 2kB with a L2 cache of size 64kB i.e. AAT of 0.7102ns

2) The memory configuration that has the smallest total area of 0.2572mm^2 with an AAT of within 5% of best AAT is - L1 cache of 1kB and L2 cache of 32kB.