NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

Prakhar Malhotra

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

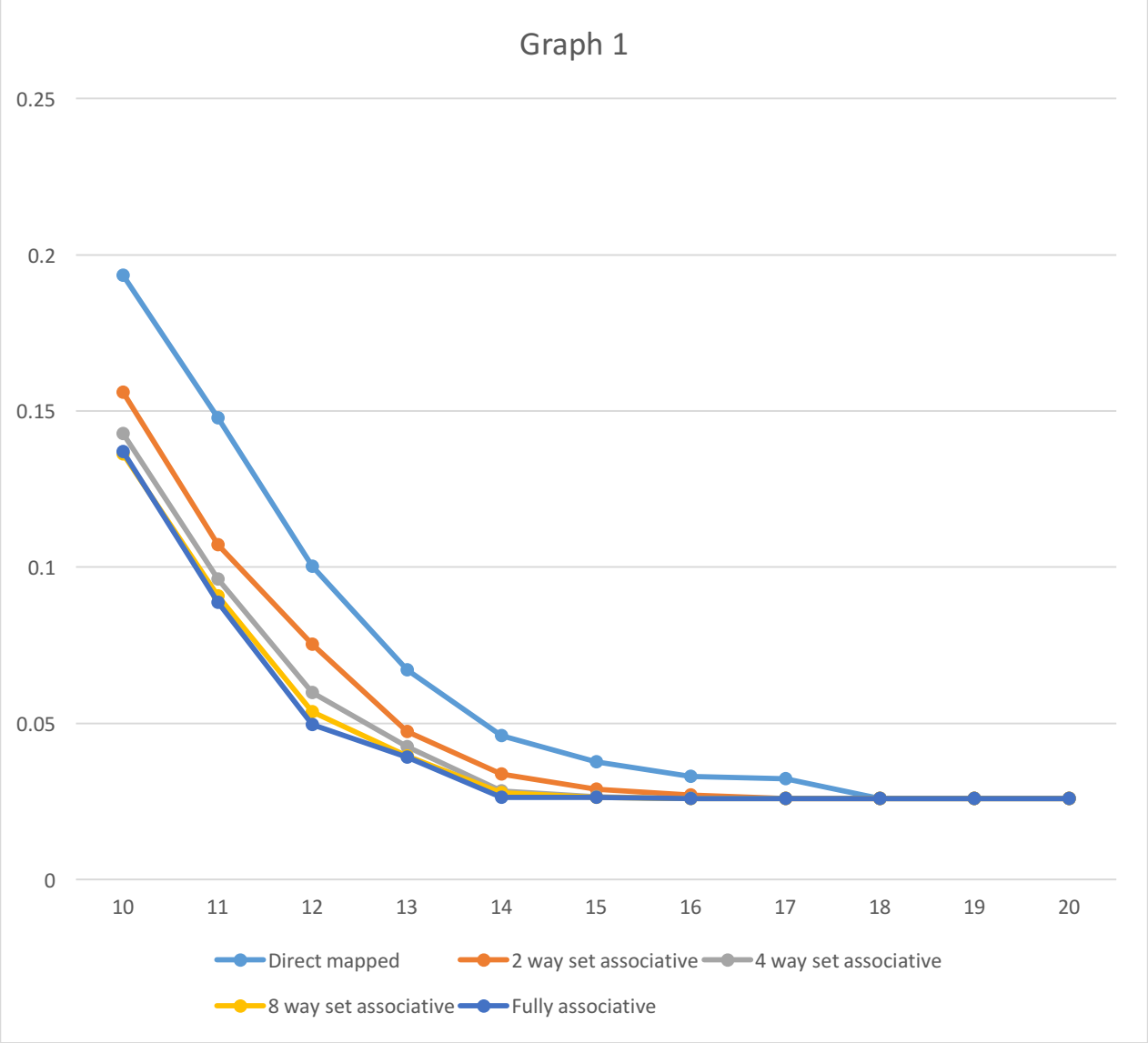
Student's electronic signature: _____Prakhar Malhotra_____

(sign by typing your name)

Course number: _____ECE 521____

(463 or 521 ?)

Graph 1:



Plot of L1 miss rate (*on the y axis*) vs log (L1size) (*on the x axis).*

Plot values:

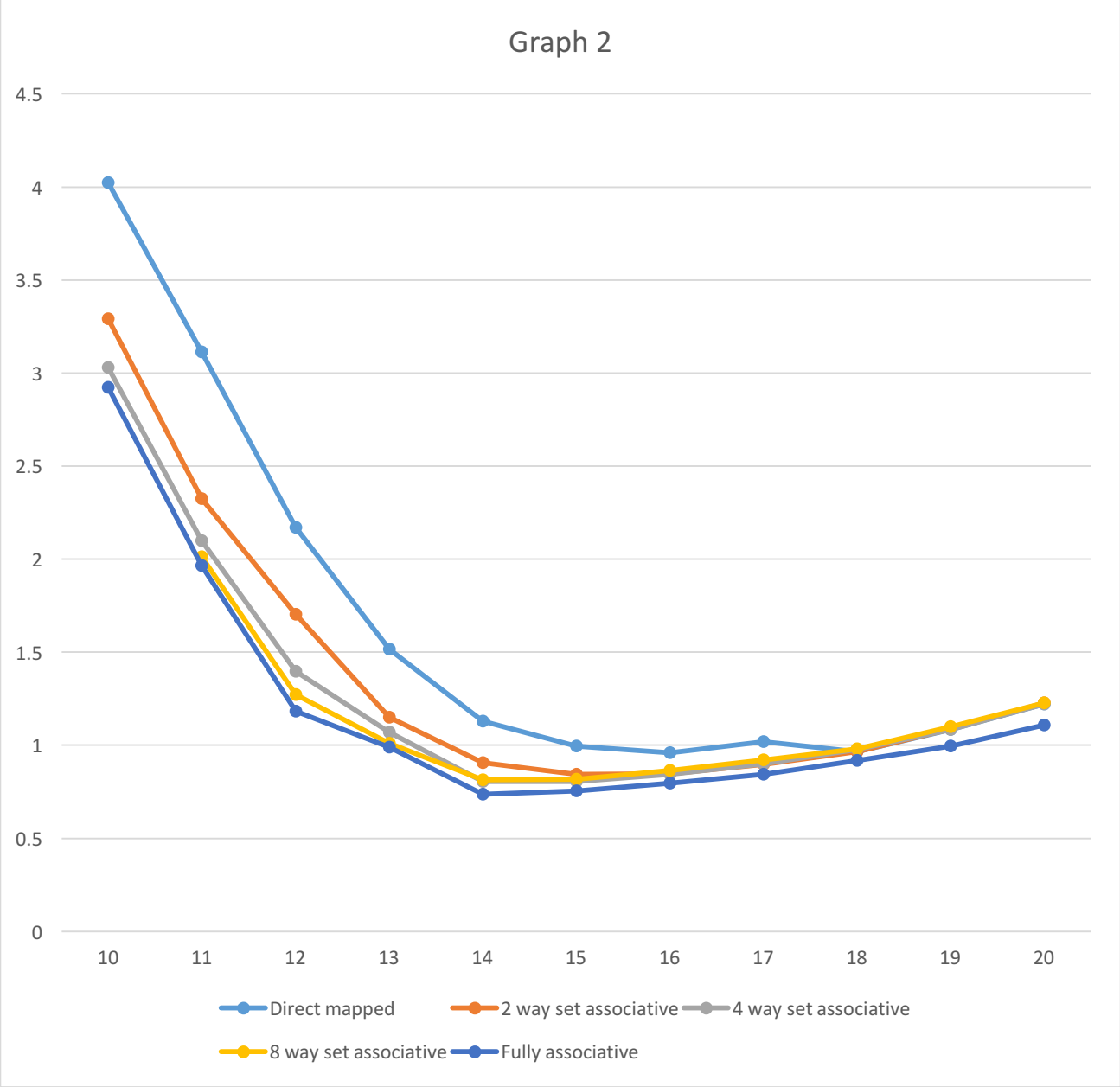
log(size)	Direct mapped	2 way set associative	4 way set associative	8 way set associative	Fully associative
10	0.1935	0.156	0.1427	0.1363	0.137
11	0.1477	0.1071	0.0962	0.0907	0.0886
12	0.1002	0.0753	0.0599	0.0536	0.0495
13	0.067	0.0473	0.0425	0.0395	0.0391
14	0.0461	0.0338	0.0283	0.0277	0.0263
15	0.0377	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

Discussion:

1. For a given associativity, the miss rate reduces non-linearly as the cache size is increased up-to a certain limiting value of miss rate which is constant across various values of associativity. The miss rate does not decrease any further even if the cache size is increased. For a given cache size, as the associativity is increased, the miss rate reduces (this too follows a non-linear trend).
2. Firstly, for a fully associative cache, there are no conflict misses. Secondly, as the cache size is increased for the fully associative cache, capacity misses are reduced but compulsory misses are bound to happen irrespective of cache size. Following this train of thought, we can estimate the compulsory miss rate from the graph as the value the miss rate takes when we keep increasing the cache size, but there is no effect on the miss rate. We can safely assume that nearly all the capacity misses have been addresses and only the compulsory misses happen. The value of the compulsory miss rate is 0.0258.
3. In order to estimate the conflict miss rate, we can simply compare the curve for a given set associativity to the miss rate we get for a fully associative cache of the same size. A fully associative cache does not have conflict misses. Therefore, we can subtract the miss rate of a fully associative cache from the miss rate of the set associative cache to get the conflict miss rate of the given set associative cache. For a cache size of 1 KB:
    a. Estimated conflict rate for direct mapped cache = 0.1935 – 0.137 = 0.0565
    b. Estimated conflict rate for 2 way set associative cache = 0.156 – 0.137 = 0.019
    c. 4 way set associative cache = 0.1427 – 0.137 = 0.0057
    d. 8 way set associative cache = 0.1363 – 0.137 = 0. An 8 way set associative is nearly fully associative in that it has no conflict misses. In fact, for the given trace, 8 way set associative cache performs better than a fully associative cache.
    e. Conflict miss rate for a fully associative cache is 0.

Graph 2:



Plot of average access time (*on y axis*) vs log (L1size) (*on x axis*)

Plot values (in nanoseconds):

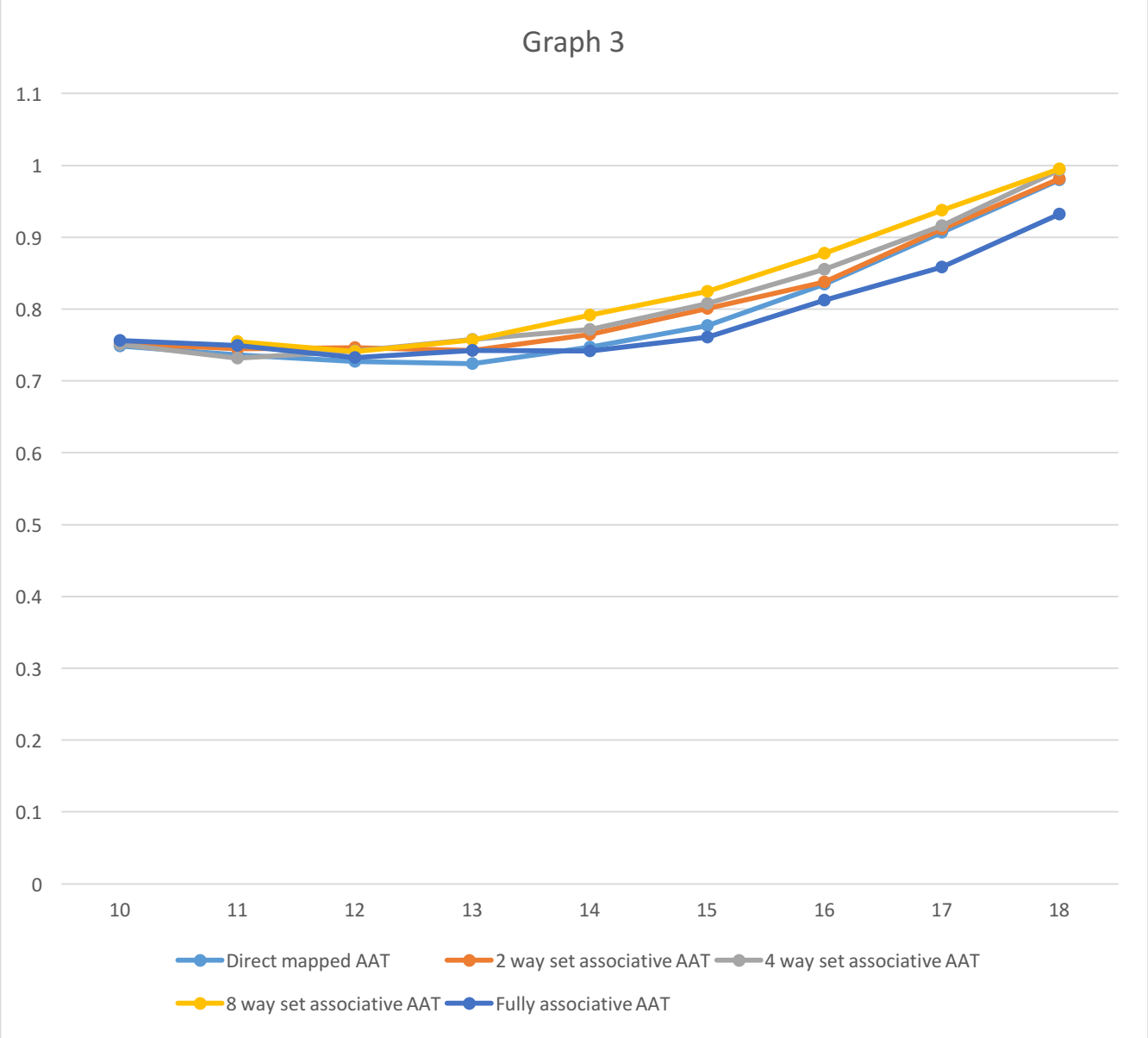
log(size)	Direct mapped	2 way set associative	4 way set associative	8 way set associative	Fully associative
10	4.023497	3.291529	3.02936		2.922884
11	3.11263	2.325111	2.097736	2.012826	1.966235
12	2.171045	1.702191	1.395665	1.271785	1.182848
13	1.51723	1.149655	1.069673	1.010811	0.988401

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

Discussion:

1. A 16 KB fully associative cache has the lowest average access time 0.7368 ns for the given trace.

Graph 3:



Plot of average access time (*on y axis*) vs log (L1size) (*on x axis*)

Plot values:

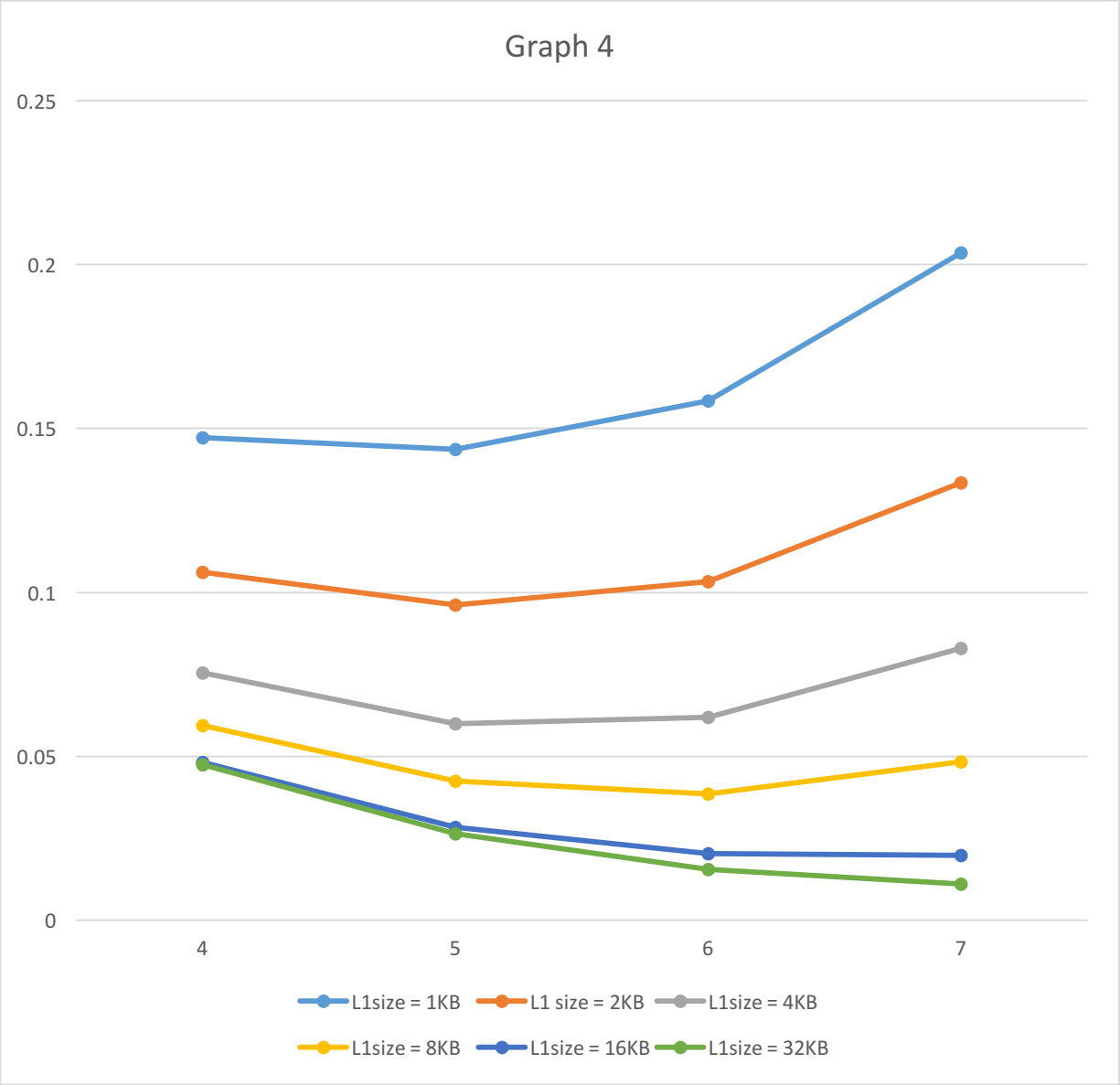
log(size)	Direct mapped AAT	2 way set associative AAT	4 way set associative AAT	8 way set associative AAT	Fully associative AAT
10	0.7484857	0.752048212	0.750777344		0.756349149
11	0.736009535	0.744997977	0.731487919	0.754736912	0.74926589
12	0.726735847	0.746391308	0.741698184	0.741168423	0.732715642
13	0.724168219	0.742650656	0.757713523	0.756777692	0.742468921

14	0.746740404	0.764405263	0.771480831	0.791189335	0.741608233
15	0.776957481	0.80047081	0.808061713	0.824219701	0.760660397
16	0.834879117	0.837375211	0.855019884	0.877955198	0.812149503
17	0.906529673	0.911135926	0.916356967	0.937312967	0.858562967
18	0.979472039	0.981589039	0.993761967	0.995001967	0.932085967

Discussion:

1. A 2 KB direct mapped L1 cache provides the AAT closest (0.7360 ns) to that achieved from a single L1 cache with the lowest AAT (0.7368 ns) in the previous graph.
2. The lowest average access time is achieved with an 8KB direct mapped L1 cache and the value is 0.7241 ns. This value is 1.72 % lower than the previously achieved lowest value of average access time.
3. Total area required for the optimal AAT configuration in graph 2 is 0.0634 millimeter squared. Total area required for the optimal AAT configuration in graph 3 is 0.0532 + 2.6401 = 2.6933 millimeter squared or 42.48 times the area required for optimal position in graph 2.

Graph 4:



Plot of L1 miss rate (*on y axis*) vs log (block size) (*on x axis*)

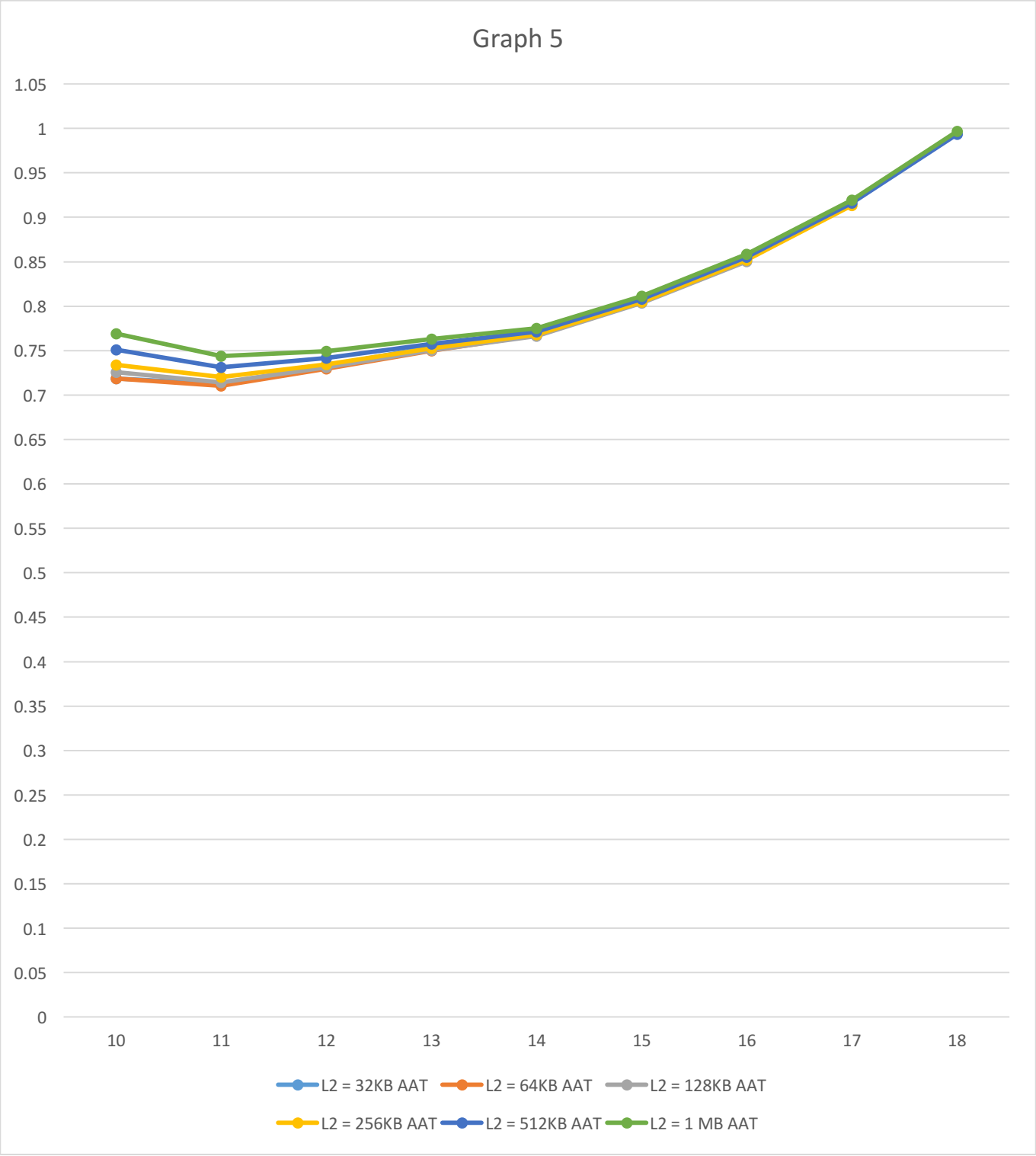
Plot values:

log(BLOCKSIZE)	L1size = 1KB	L1 size = 2KB	L1size = 4KB	L1size = 8KB	L1size = 16KB	L1size = 32KB
4	0.1473	0.1062	0.0755	0.0595	0.0482	0.0475
5	0.1437	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

Discussion:

1. Smaller caches seem to prefer smaller block sizes whereas larger caches prefer larger block size. For a cache of a given size and fixed associativity, as the block size is increased to exploit more spatial locality, the number of sets reduces. Thus there is a tradeoff between exploiting more spatial locality and the number of sets. For a cache as small as 1 KB, a block size of 32 bytes gives an optimum miss rate as it achieves the optimum miss rate(for number of sets equal to 32). A block size more than that leads to less number of sets (16) which may cause more conflict misses. Also, we could be bringing in blocks which are never referenced assuming a more aggressive spatial locality than what actually exists. This phenomenon is referred to as *cache pollution.* This tradeoff is evident from the graph as the miss rate initially decreases for each cache size as the block size is increased and then starts increasing as more and more *cache pollution* is introduced. Also there is shift in the lowest point of the trough from left to right as the cache size is increased or the *balance between the two factors shifts*.

Graph 5:



Plot of average access time(*on y axis*) vs log (L1size) (*on x axis*)

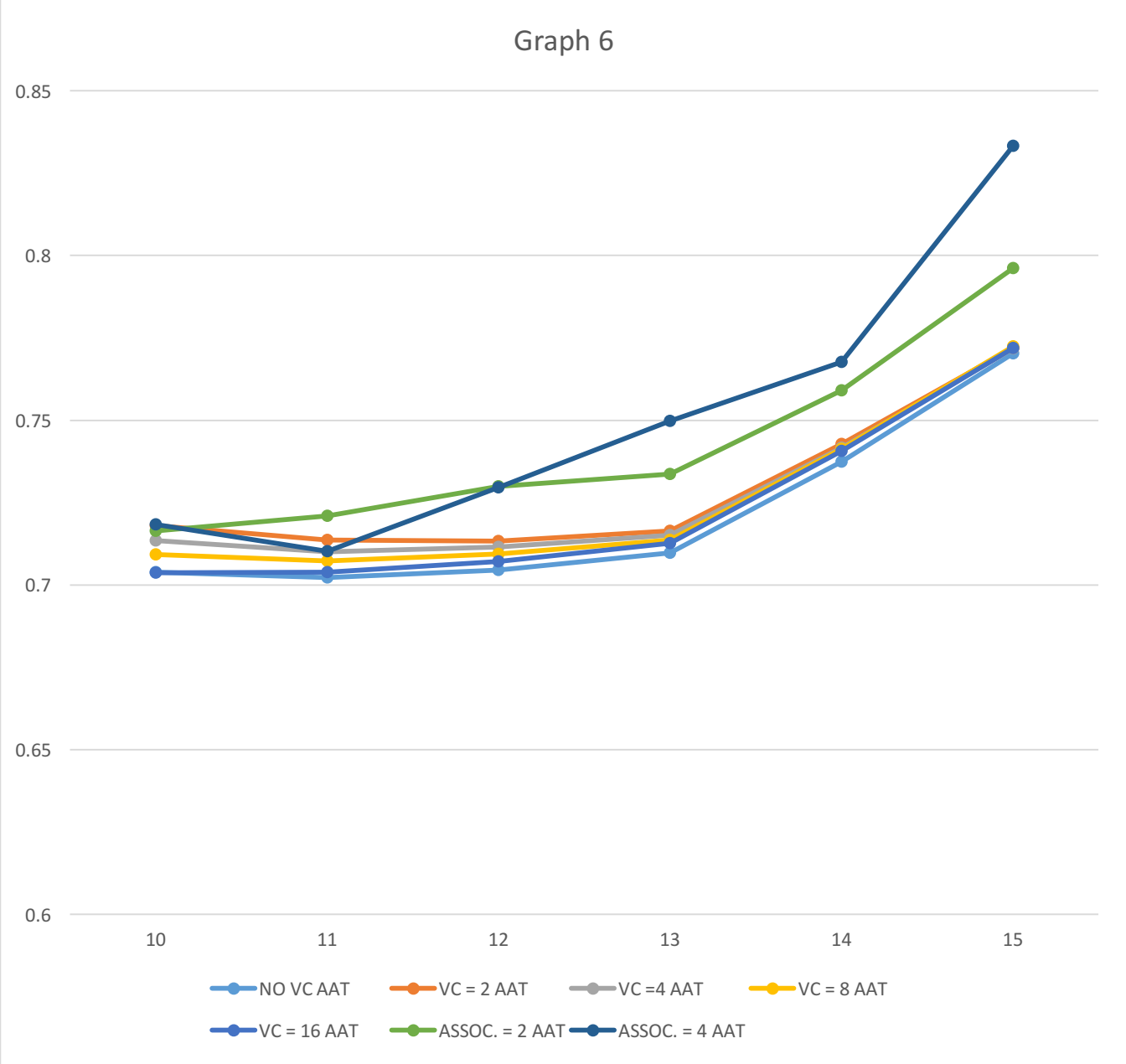
Plot values:

log(size)	L2 = 32KB AAT	L2 = 64KB AAT	L2 = 128KB AAT	L2 = 256KB AAT	L2 = 512KB AAT	L2 = 1 MB AAT
10	0.71837788	0.718403851	0.725527863	0.733760084	0.750777344	0.768991857
11	0.71236663	0.710246575	0.714466195	0.720015877	0.731487919	0.74376708
12	0.733059047	0.729561007	0.731099418	0.73455499	0.741698184	0.74934394
13	0.754073568	0.749788803	0.75019353	0.752645313	0.757713523	0.763138308
14	0.771801017	0.76763305	0.766473401	0.768106	0.771480831	0.7750931
15		0.805058871	0.80339047	0.80491346	0.808061713	0.811431462
16			0.850437112	0.851931258	0.855019884	0.858325812
17				0.913280265	0.916356967	0.91965013
18					0.993761967	0.99705513

Discussion:

1. A 2KB L1 cache and a 64 KB L2 cache give the lowest average access time of 0.7102 ns for the given block size and associativity values.
2. A 1KB L1 cache and a 32 KB L2 cache have the smallest total area of 0.2572 millimeter squares and give an AAT value of 0.7183 which falls within 5% of the optimal AAT.

Graph 6:



Plot average access time (*on y axis*) vs log (L1size) (*on x axis*)

Plot values:

log(L1size)	NO VC AAT	VC = 2 AAT	VC =4 AAT	VC = 8 AAT	VC = 16 AAT	ASSOC. = 2 AAT	ASSOC. = 4 AAT
10	0.70380578	0.717973	0.713498538	0.709146468	0.703610766	0.716342308	0.718403851
11	0.70220337	0.713593	0.710023208	0.707303034	0.703827611	0.720917184	0.710246575

12	0.70440888	0.713269	0.711553101	0.709345567	0.707037048	0.729916873	0.729561007
13	0.70964503	0.7164	0.715078887	0.713529515	0.712655371	0.733639817	0.749788803
14	0.73739944	0.742807	0.741566086	0.74136853	0.740567119	0.758990367	0.76763305
15	0.77030856	0.771885	0.771718853	0.772339733	0.771860761	0.796089638	0.805058871

Discussion:

1. Adding a victim cache to a direct mapped L1 cache works better on the whole than a two way set associative cache across the range of L1 sizes observed here. The performances are comparable for a 1KB cache size but other than that, two way set associative caches have more access time than an L1 cache with victim cache(irrespective of the number of blocks in the VC).
2. A 2KB direct mapped L1 cache gives the lowest AAT 0.7022 ns.
3. A 1 KB 2 way set associative L1 cache + given L2 cache gives an AAT within 5% of the optimal AAT 0.7022 ns and the smallest area 0.3697 millimeter squared.