NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

by

RAMACHANDRAN SEKANIPURAM SRIKANTHAN

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

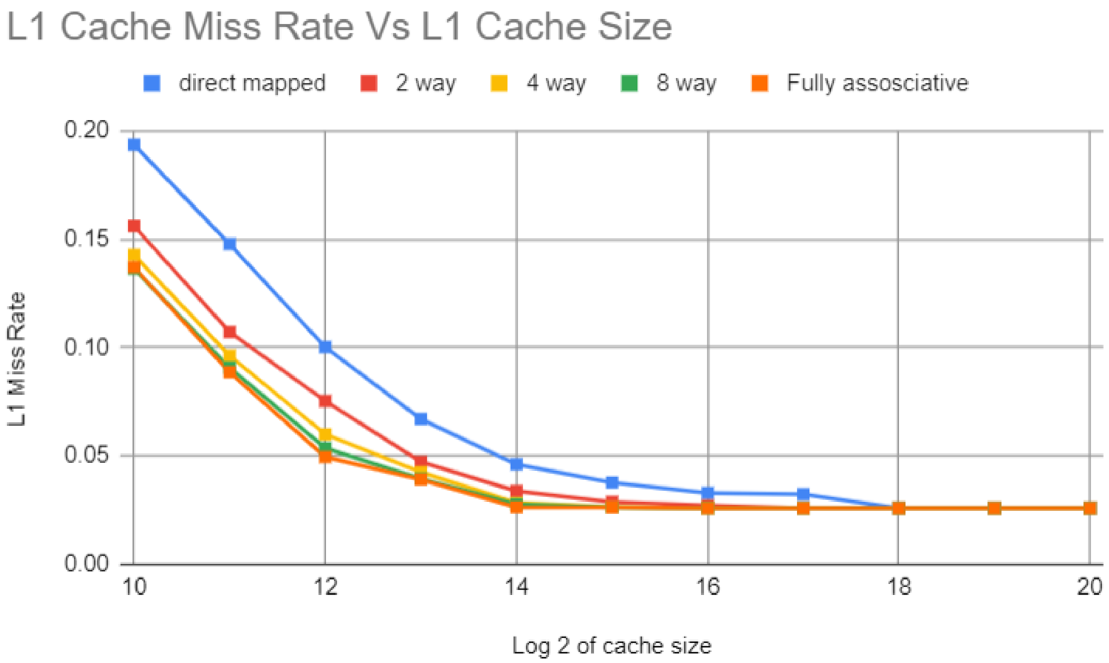
Student’s electronic signature: Ramachandran Sekanipuram Srikanthan

Course number: 563

Graph 1: L1 Cache exploration (L1 miss rate Vs log2(CACHE SIZE))

The Graph is plotted for L1 miss rate along y axis and log2(CACHE SIZE) on x axis for each value of associativity. The L1 block size is fixed to 32 bytes. The cache size is varied from 1KB to 1MB in powers of two and the associativity is varied from direct mapped to fully associative (direct mapped,2 way set assosciative,4 way set assosciative,8 way set associative and fully associative). Here we don't use any Victim cache or L2 cache.

Log 2 of cache size	direct mapped	2 way	4 way	8 way	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**



From the graph we can see that, for a given associativity (say Direct mapped), the miss rate decreases as we increase the size of the cache which denotes that the more space, we have in the cache the lesser is the miss. Also, for a given cache size (say 16KB), the miss rate for direct mapping is greater than the miss rate of fully associative. The reason being that increasing associative increases the space to keep many tags that point to same set unlike direct mapped wherein the tag is being replaced when a new tag pointing to same index occur.

From the graph we can see that, increasing the cache size for a given associativity, at some point the curve asymptotically approaches the compulsory misses. Here the cache size is larger enough to eliminate capacity and conflict misses. In our graph, we have the compulsory miss rate to be **0.0258**.

Now the conflict miss rate is found out by finding out the difference between the miss rate for a particular cache and the corresponding size of fully associative cache. We compare the miss rate for direct mapped, 2-way, 4-way and 8-way associative with the miss rate of the corresponding fully associative cache.

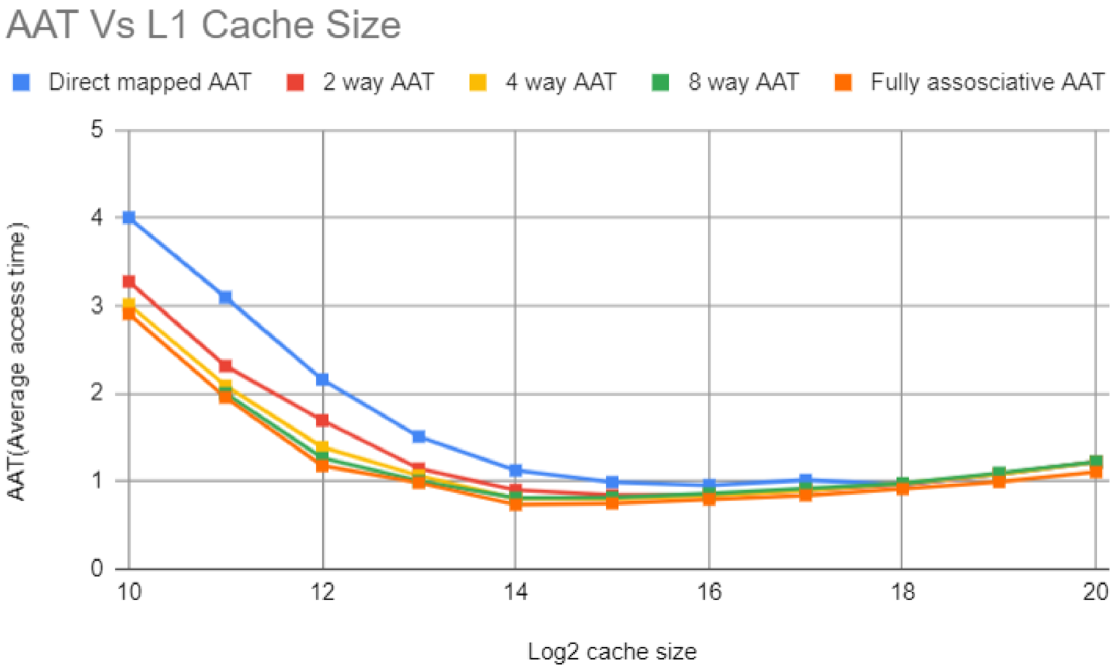
Conflict miss Rate

Log 2 of cache size	direct mapped	2-way	4-way	8-way
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	1E-04	1E-04
17	0.0065	1E-04	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0

Graph 2: L1 Cache exploration (AAT Vs log2(CACHE SIZE))

The Graph is plotted for AAT (Average Access time) along y axis and log2(CACHE SIZE) on x axis for each value of associativity. The L1 block size is fixed to 32 bytes. The cache size is varied from 1KB to 1MB in powers of two and the associativity is varied from direct mapped to fully associative (direct mapped,2 way set assosciative,4 way set assosciative,8 way set associative and fully associative). Here we don't use any Victim cache or L2 cache.

Log2 cache size	Direct mapped AAT (ns)	2-way AAT (ns)	4-way AAT (ns)	8-way AAT (ns)	Fully associative AAT (ns)
10	4.004147	3.275929	3.01509	NaN	2.909184
11	3.09786	2.314401	2.088116	2.003756	1.957375
12	2.161025	1.694661	1.389675	1.266425	1.177898
13	1.51053	1.144925	1.065423	1.006861	0.984491
14	1.125027	0.903297	0.802766	0.811124	**0.734238**
15	0.991123	0.841326	0.80189	0.815131	0.75136
16	0.955917	0.845437	0.840071	0.861803	0.794861
17	1.01603	0.895193	0.89886	0.919816	0.841066
18	0.962392	0.964509	0.976265	0.977505	0.914589
19	1.082031	1.086324	1.082998	1.096757	0.994308
20	1.21796	1.224626	1.218187	1.224399	1.107054

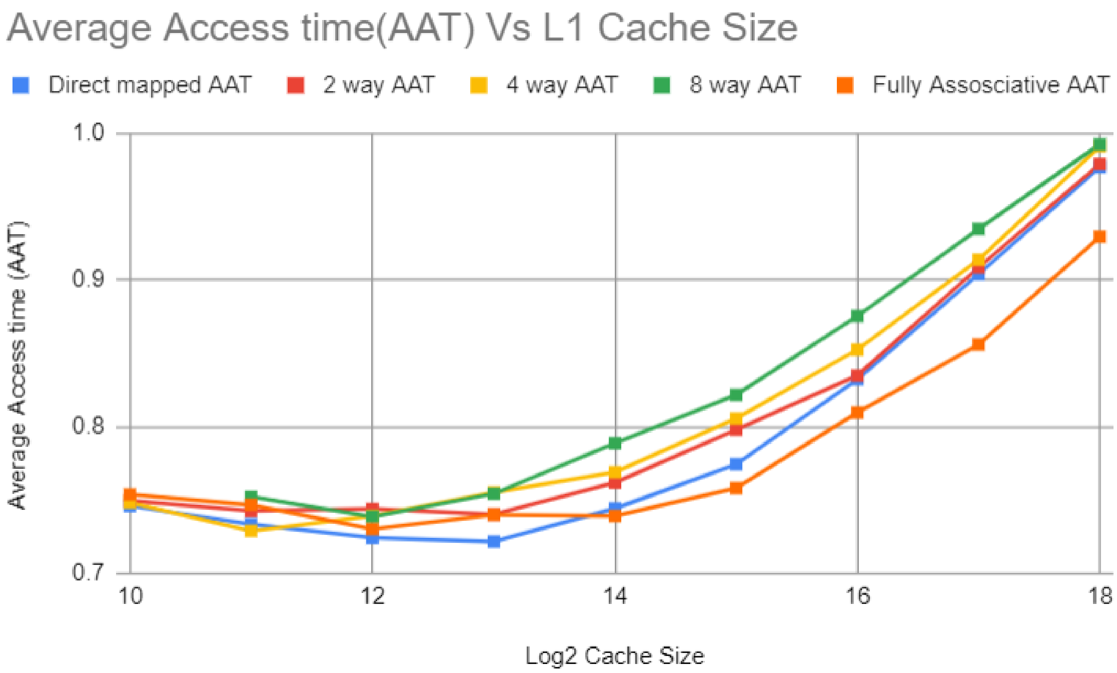


From the graph we can see that, the Fully assosciative cache configuration yields lowest AAT when compared to other configurations. The AAT is higher whenever the cache size is small due to the occurrences of misses. But at a point of cache size it becomes lower due to the reduction in misses. At this point the fully assosciative cache has an AAT much lower than direct mapped since in direct mapped only one tag for an index is stored casuing the significant misses. But after this point the AAT starts to increase as the hit time increases with larger cache sizes.

Graph 3: L1 Cache exploration (Average Access time (AAT) Vs log2(CACHE SIZE))

The Graph is plotted for Average Access time (AAT) along y axis and log2(L1 CACHE SIZE) on x axis for each value of associativity. The L1 and L2 block size is fixed to 32 bytes. The cache size is varied from 1KB to 256KB in powers of two and the associativity is varied from direct mapped to fully associative (direct mapped,2 way set assosciative,4 way set assosciative,8 way set associative and fully associative). Here we don’t use any Victim cache. The Cache Size of L2 is set to 512KB and its associativity is set to 8.

Log2 L1 Cache Size	Direct mapped AAT (ns)	2-way AAT (ns)	4-way AAT (ns)	8-way AAT (ns)	Fully Associative AAT (ns)
10	0.745902	0.749466	0.748196	NaN	0.7537667
11	0.733428	0.742417	0.728907	0.752155	0.74668409
12	0.724153	0.743809	0.739117	0.738589	0.7301357
13	**0.721586**	0.740071	0.75513	0.754198	0.73988832
14	0.744158	0.761826	0.768901	0.788611	0.73903004
15	**0.774374**	0.79789	0.80548	0.821643	0.75808232
16	0.832299	0.834796	0.852443	0.875372	0.80957053
17	0.90395	0.908554	0.913777	0.934733	0.85598297
18	0.976894	0.979011	0.991182	0.992422	0.92950597



Here we can see that with L2 being there in the hierarchy the AAT is better for lower L1 cache sizes. But if L1 cache size increased while trying to getting better performance, the AAT increases as now we have both L1 and L2 hit times coming into picture and so increasing the L1 cache size further causes icrease in L1 hit time therby increasing AAT.

From the graph, we can see that adding L2 to the same L1 configuration as previous graph will produce an AAT of **0.774374** in L1 direct mapped configuration which is in 5% tolerance range of the best value of AAT produced in the Graph 2 of *0.734238* for the L1 fully assosciative configuration of L1 cache size 16KB.

With L2 addded to provious configruation, the lowest AAT of **0.721586** is obtained for Direct mapped cache for L1 cache size of 8KB.

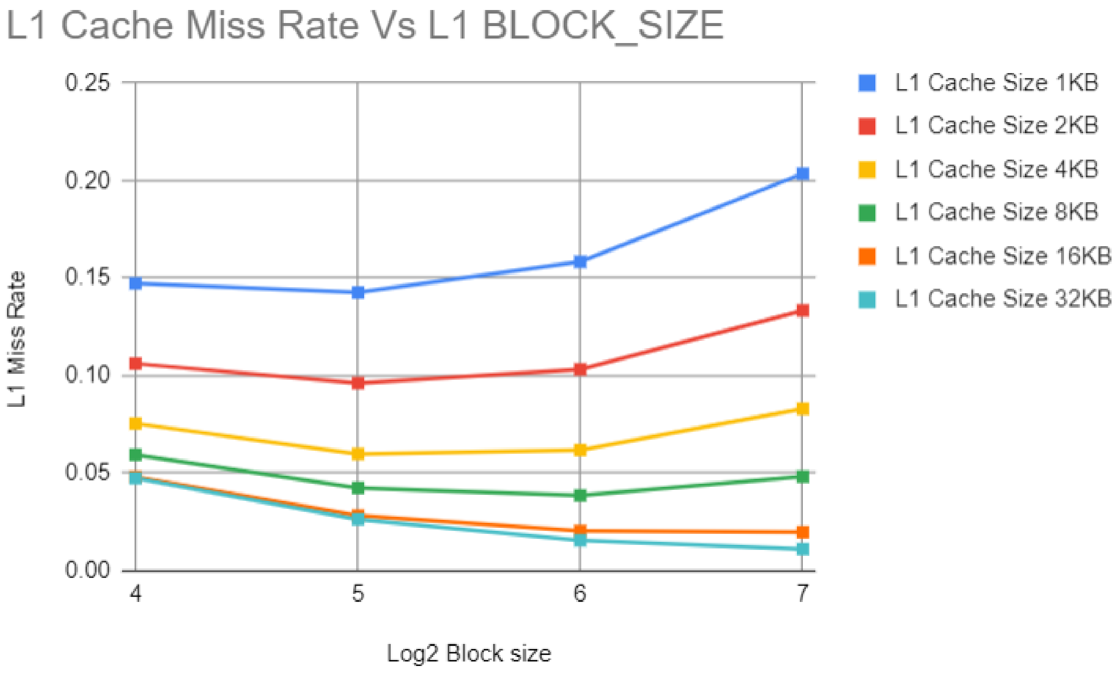
Total area required for optimal AAT configuration with L2 cache:
Area = Area of L1 + Area of VC + Area of L2
Area = 0.053293238 + 0 + 2.640142073 = **2.693435311 mm^2**

Total area required for optimal AAT configuration without L2 cache.
Area = Area of L1 + Area of VC + Area of L2 = **0.063446019 mm^2**

Graph 4: L1 Cache exploration (L1 Miss Rate Vs log2(BLOCK_SIZE))

The Graph is plotted for L1 miss rate along y axis and log2(BLOCK SIZE) on x axis for each value of L1 Cache Sizes. The Block size is varied as in 16, 32, 64 and 128 bytes. The L1 Cache Size is varied from 1KB to 32KB in powers of 2. The L1 associativity is fixed to 4. Victim Cache and L2 Cache is not being used here.

Log2 Block size	L1 Cache Size 1KB	L1 Cache Size 2KB	L1 Cache Size 4KB	L1 Cache Size 8KB	L1 Cache Size 16KB	L1 Cache Size 32KB
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



From the graph, we can see that for the fixed Cache Size the miss rate may first decrease upto a point by exploiting more spatail locatlity as block size being increased. But after this point, as the block size gets larger for this fixed Cache size, causing total number of blocks becoming while the block size of a block is getting greater. This causes Cache Pollution as increasing cache size now takes cache space away from useful bytes in other blocks.

We can also see that, in smaller Caches when we increase the block size the miss rate starts to increase beyond a point due to Cache Pollution. Thus it is better to have small block sizes for smaller caches as they don’t take away the space in adjacent blocks where other data can be stored.

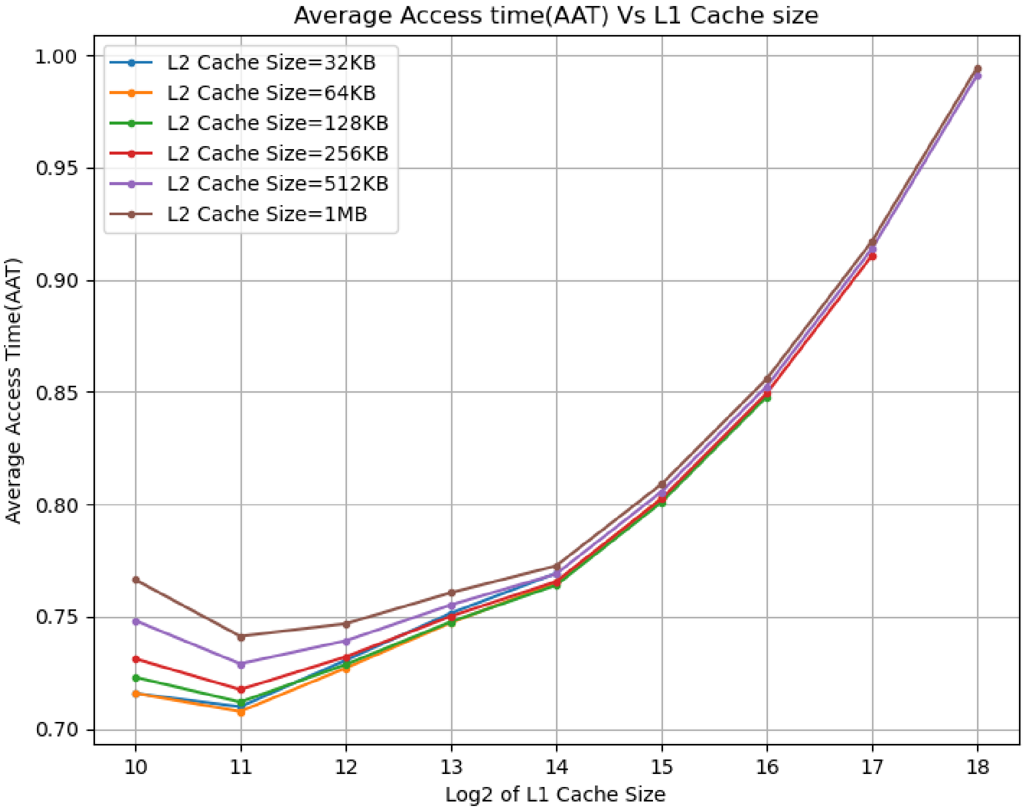
On the other hand, in the case of larger Caches, having a larger block size is advantageous. We can see from the graph that, for the Cache Size of 32KB increasing the block size to 128 bytes gives lower miss rate than the block size of 16 bytes. Since the Cache size is larger, it takes larger block size to pollute the cache.

Thus as block size is increased from 16, 32, 64 to 128 bytes, the smaller sized caches suffer from cache pollution at 64 bytes block sizes and the larger sized graphs didn’t suffer from cache pollution even at 128 bytes of block size. Thus the balance between exploiting more spatial locality versus increasing cache pollution shifts towards the higher block sizes for larger Caches.

Graph 5: L1 + L2 co exploration (Average Access time (AAT) Vs log2(L1_CACHE_SIZE))

The graph is plotted by taking Average Access time (AAT) on Y axis versus Log2(L1 Cache size) in x axis. The L2 cache sizes are varied from 32KB, 64KB, 128KB, 256KB, 512KB to 1MB. For each value of L2 cache size, the L1 cache size is varied from 1KB to 256KB in powers of two. Also at any time the size of L1 cache does not go beyond the size of L2. Here we use L1 and L2 Block size of 32 bytes along with L1 assosciativity of 4 and L2 associativity of 8. Here there is no Victim Cache.

	L2 Log2 Cache Size					
Log2 L1 cache size	32 KB	64KB	128KB	256KB	512KB	1MB
10	0.715752	0.715815	0.722946	0.731179	0.748196	0.76641
11	0.709742	**0.707658**	0.711885	0.717435	0.728907	0.741186
12	0.730435	0.72697	0.728518	0.731974	0.739117	0.746763
13	0.751447	0.747194	0.74761	0.750061	0.75513	0.760554
14	0.769179	0.765039	0.763893	0.765526	0.768901	0.772513
15	NA	0.802461	0.800809	0.802332	0.80548	0.80885
16	NA	NA	0.84786	0.849354	0.852443	0.855749
17	NA	NA	NA	0.9107	0.913777	0.91707
18	NA	NA	NA	NA	0.991182	0.994475



The lowest AAT is attained by using the configuration of L1 Cache Size of 2KB and L2 Cache size of 64KB with an AAT of **0.707658**.

The total area of is given by the following table. The red and green highlighted ones are the Cache sizes which fall under the margin of 5%.

	L2 Log2 Cache Size					
Log2 L1 cache size	32 KB	64KB	128KB	256KB	512KB	1MB
10	0.25728558	0.37543256	0.57504828	1.30865548	2.65525702	4.88931635
11	0.26083299	**0.37897997**	0.57859569	1.3122029	2.65880443	4.89286376
12	0.27981119	0.39795816	0.59757389	1.33118109	2.67778263	4.91184196
13	0.31060479	0.42875177	0.62836749	1.36197469	2.70857623	4.94263556
14	0.34811233	0.4662593	0.66587503	1.39948223	2.74608377	4.9801431
15	NA	0.59696529	0.79658101	1.53018822	2.87678975	5.11084909
16	NA	NA	0.8622227	1.59582991	2.94243144	5.17649077
17	NA	NA	NA	1.9605585	3.30716004	5.54121937
18	NA	NA	NA	NA	3.78143688	6.01549621

The configuration of L2 Cache Size of 32KB and L1 Cache Size of 1KB (highlighted in green) yields the smallest total area of 0.25728558 mm^2 which falls in the margin of 5% if the best AAT(highlighted in bold).

Thus, we decrease the area requirements of the Cache to 32.11% by having an AAT within the tolerance of 5% of the best AAT.

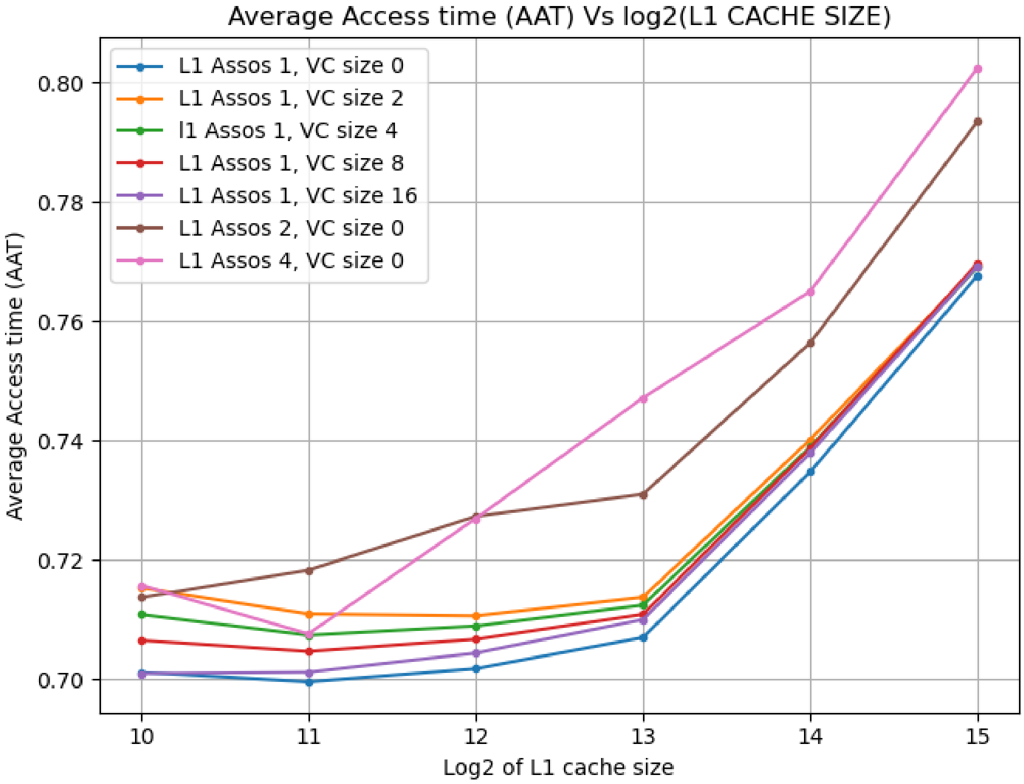
Graph 6: Victim Cache Study (Average Access time (AAT) Vs log2(L1_CACHE_SIZE))

The graph is plotted having Average Access time (AAT) on y-axis versus log2(L1 Cache size) on the x axis. The L1 Cache size is varied from 1KB to 32KB in powers of 2 for each of the following combinations as given below.

- Direct-mapped L1 cache with no Victim Cache.
- Direct-mapped L1 cache with 2-entry Victim Cache.
- Direct-mapped L1 cache with 4-entry Victim Cache.
- Direct-mapped L1 cache with 8-entry Victim Cache.
- Direct-mapped L1 cache with 16-entry Victim Cache.
- 2-way set-associative L1 cache with no Victim Cache.
- 4-way set-associative L1 cache with no Victim Cache.

Here we fix the Blocksize of L1 and L2 to 32 bytes .The Size of L2 is set to 64KB and assosciativity is set to 8.

	Average Access time (AAT) ns						
Log2 L1 Cache size	Associativity = 1; No Victim Cache	Associativity = 1; 2 VC blocks	Associativity = 1; 4 VC blocks	Associativity = 1; 8 VC blocks	Associativity = 1; 16 VC blocks	Associativity = 2; No Victim Cache	Associativity = 4; No Victim Cache
10	0.701216746	0.715383926	0.710909004	0.706557357	0.701022141	0.713754268	0.715815273
11	0.699615664	0.71100311	0.707434303	0.704713746	0.70123871	0.718329648	0.707657833
12	0.701818713	0.710679925	0.708963304	0.706757709	0.704444198	0.727327306	0.726969733
13	0.707056151	0.713807656	0.712487232	0.710938997	0.710059702	0.731049196	0.747194178
14	0.734809078	0.740208997	0.738971616	0.738772048	0.737972443	0.756398583	0.765038789
15	0.767714044	0.769295478	0.76912732	0.769744213	0.769266187	0.793496486	0.802460847



The AAT comparison of L1 direct mapped cache for various victim block sizes with the corresponding sized L1 cache with 2 way set assosciativity is given below. The highlighted ones indicates the lower AAT time using vicitm cache for a direct mapped L1 cache in comparison with the corresponding sized 2-way set assosciative cache.

	Average Access time (AAT) ns						
Log2 L1 Cache size	Associativity = 1; No Victim Cache	Associativity = 1; 2 VC blocks	Associativity = 1; 4 VC blocks	Associativity = 1; 8 VC blocks	Associativity = 1; 16 VC blocks	Associativity = 2; No Victim Cache	Associativity = 4; No Victim Cache
10	0.701216746	0.715383926	0.710909004	0.706557357	0.701022141	0.713754268	0.715815273
11	0.701615664	0.71100311	0.707434303	0.704713746	0.70123871	0.718329648	0.707657833
12	0.701818713	0.710679925	0.708963304	0.706757709	0.704444198	0.727327306	0.726969733
13	0.707056151	0.713807656	0.712487232	0.710938997	0.710059702	0.731049196	0.747194178
14	0.734809078	0.740208997	0.738971616	0.738772048	0.737972443	0.756398583	0.765038789
15	0.767714044	0.769295478	0.76912732	0.769744213	0.769266187	0.793496486	0.802460847

Thus we can see that adding Vicitim cache blocks to direct mapped cache increases the performance by decreasing the AAT, in comparison with the similar sized cache with an assosciativity of 2 without victim cache.

The memory hierachy configuration having L1 cache size of 1KB having a victim cache blocks of 16 and L2 Cache Size of 64KB provides the least AAT of 0.701022141 ns.

The total area for each of the configuration is given in the below table. The red and green highlighted ones indicate the tolerance of 5% from the best AAT value.

	Area (mm^2)						
Log2 L1 Cache size	Associativity = 1; No Victim Cache	Associativity = 1; 2 VC blocks	Associativity = 1; 4 VC blocks	Associativity = 1; 8 VC blocks	Associativity = 1; 16 VC blocks	Associativity = 2; No Victim Cache	Associativity = 4; No Victim Cache
10	0.370616	0.371312	0.371312	0.371937	**0.373101**	0.369789	0.375433
11	0.376312	0.377008	0.377008	0.377633	0.378798	0.380098	0.37898
12	0.393014	0.393709	0.393709	0.394334	0.395499	0.395976	0.397958
13	0.413611	0.414306	0.414306	0.414931	0.416096	0.444074	0.428752
14	0.457067	0.457762	0.457762	0.458387	0.459552	0.490425	0.466259
15	0.570861	0.571557	0.571557	0.572182	0.573347	0.565872	0.596965

The configuration of L1 Cache Size of 1KB, direct mapped with Victim blocks of 16 and having L2 Cache size of 64KB gives the least area of **0.369789** mm^2 with the AAT tolerance of 5% from the best AAT.