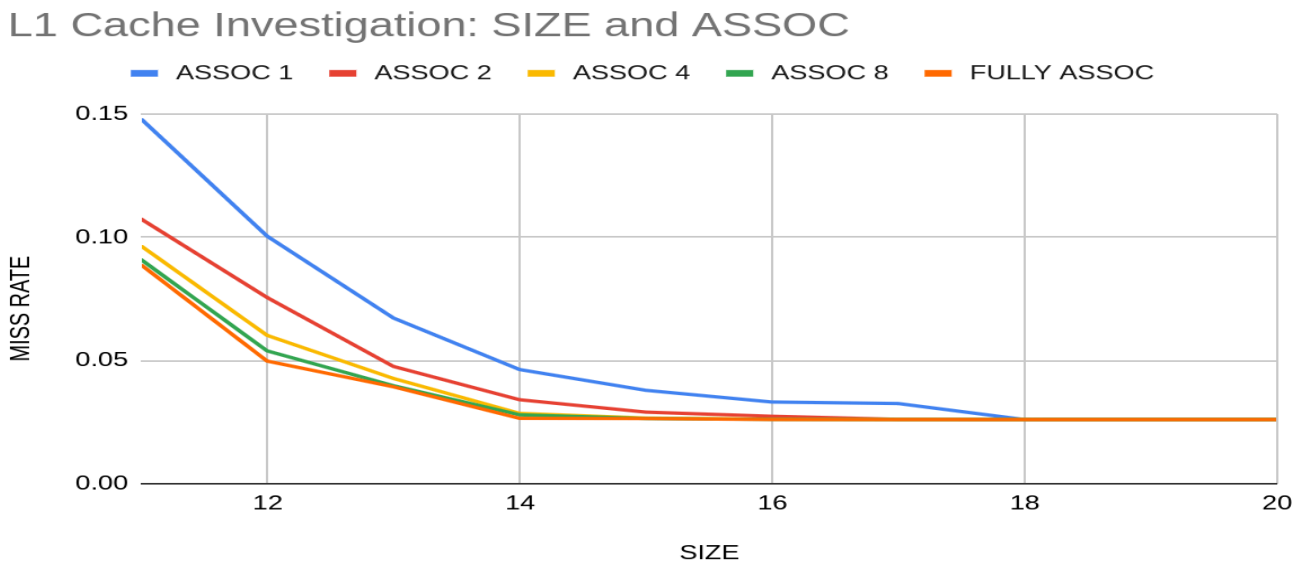
**Cache Report:**
**Name: Kumaran S**
**Roll no:NS24Z265**

**9.1 L1 Cache Investigation: SIZE and ASSOC**
**Plot #1**

For this experiment, use BLOCKSIZE = 32, and assume that there are no Victim and L2 caches.
Plot L1 miss rate on the y-axis versus log2(SIZE) on the x-axis, for ten different cache sizes: SIZE = 2KB, ... , 1MB. The graph should contain five separate curves, one for each of the following ASSOCs: direct-mapped, 2-way set-associative, 4-way set-associative, 8-way set-associative, and fully-associative.

| ASSoc | Missrate | size |
|---|---|---|
| 1 | 0.14774 | 11 |
| | 0.10017 | 12 |
| | 0.067 | 13 |
| | 0.04609 | 14 |
| | 0.03768 | 15 |
| | 0.03292 | 16 |
| | 0.03233 | 17 |
| | 0.02584 | 18 |
| | 0.02584 | 19 |
| | 0.02584 | 20 |
| 2 | 0.10714 | 11 |
| | 0.07528 | 12 |
| | 0.04734 | 13 |
| | 0.03384 | 14 |
| | 0.02881 | 15 |
| | 0.02713 | 16 |
| | 0.0259 | 17 |
| | 0.02584 | 18 |
| | 0.02582 | 19 |
| | 0.02582 | 20 |
| 4 | 0.09622 | 11 |
| | 0.05992 | 12 |
| | 0.04247 | 13 |
| | 0.02832 | 14 |
| | 0.0264 | 15 |

| | | |
|---|---|---|
| | 0.02595 | 16 |
| | 0.02582 | 17 |
| | 0.02582 | 18 |
| | 0.02582 | 19 |
| | 0.02582 | 20 |
| 8 | 0.09069 | 11 |
| | 0.05365 | 12 |
| | 0.03954 | 13 |
| | 0.02774 | 14 |
| | 0.02625 | 15 |
| | 0.02589 | 16 |
| | 0.02582 | 17 |
| | 0.02582 | 18 |
| | 0.02582 | 19 |
| | 0.02582 | 20 |
| full | 0.0886 | 11 |
| | 0.04954 | 12 |
| | 0.03912 | 13 |
| | 0.02632 | 14 |
| | 0.02624 | 15 |
| | 0.02583 | 16 |
| | 0.02582 | 17 |
| | 0.02582 | 18 |
| | 0.02582 | 19 |
| | 0.02582 | 20 |




**1. Discuss trends in the graph.**

To increase the cache performance ,one of the ways is by reducing the miss rate .The miss rate can be reduced by increasing the cache size . .In this plot we keep the Blocksize as constant as 32 and analyse the miss rate for different associativity by increasing the cache size .

Larger in size in lesser the miss rate.Across all levels of associativity, as the cache size increases , the miss rate decreases. This is expected because larger caches can hold more data, reducing the number of cache misses.

For smaller cache sizes, higher associativity (e.g., 2-way, 4-way, 8-way, and fully-associative) shows a significant reduction in miss rates compared to direct-mapped (ASSOC 1). This is because higher associativity allows for more flexible block placement, reducing conflict misses.As cache size increases, the difference in miss rates between different associativities becomes smaller, especially around log2(SIZE) = 16 and beyond. This indicates that for large cache sizes, even a direct-mapped cache performs almost as well as a fully-associative one, as the cache is large enough to avoid many conflict misses.

In the direct cache the miss rate declined at a good rate until some extent ,while increasing the cache size.Upto cache size 128KB the miss rate has been reduced ,after which the miss rate is constant  neither increased nor decreased.Similarly  we analyse for other associativity it's quite the same.For higher associativity (e.g., 4-way, 8-way, fully-associative), the reduction in miss rates becomes minimal as cache size grows. This suggests that after a certain point, increasing associativity yields diminishing returns, and additional complexity from increased associativity might not be justified for large caches.
Overall, the graph shows that both increasing cache size and associativity reduce miss rates, but beyond a certain size, the performance gains from higher associativity become marginal.

**2. Estimate the compulsory miss rate from the graph.**

Compulsory misses occur when a cache block is accessed for the first time and is not found in the cache. This is typical when the cache is empty or when a program accesses a new data block that has never been loaded into the cache.
From the graph, the compulsory miss rate can be estimated by looking at the point where all curves converge and stop decreasing significantly. All associativities converge to a similar miss rate of about 0.0258. This suggests that 0.0258 (or 2.58%) is a reasonable estimate for the compulsory miss rate as it represents the point where conflict and capacity misses have been reduced to very minimal, leaving only compulsory misses.

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

Conflict Miss Rate=Total Miss Rate−total miss rate (as same as full associative)

For direct-Mapped Cache the difference between the direct-mapped cache's miss rate and the fully associative cache's miss rate at the same cache size gives an estimate of the conflict miss rate.The direct-mapped miss rate is 0.10017, while the fully associative miss rate is 0.04954, giving a conflict miss rate estimate of approximately 0.05063 .

For 2-Way Set Associative Cache : The miss rate difference between 2-way and fully associative can be used to estimate the conflict miss rate for 2-way associativity. The 2-way miss rate is 0.07528, while the fully associative miss rate is 0.04954, giving a conflict miss rate estimate of about 0.02574 .

4-Way Set Associative Cache (ASSOC 4): For 4-way associativity, the conflict miss rate is 0.01038 .

8-Way Set Associative Cache (ASSOC 8): For 8-way, the conflict miss rate is 0.00411 .

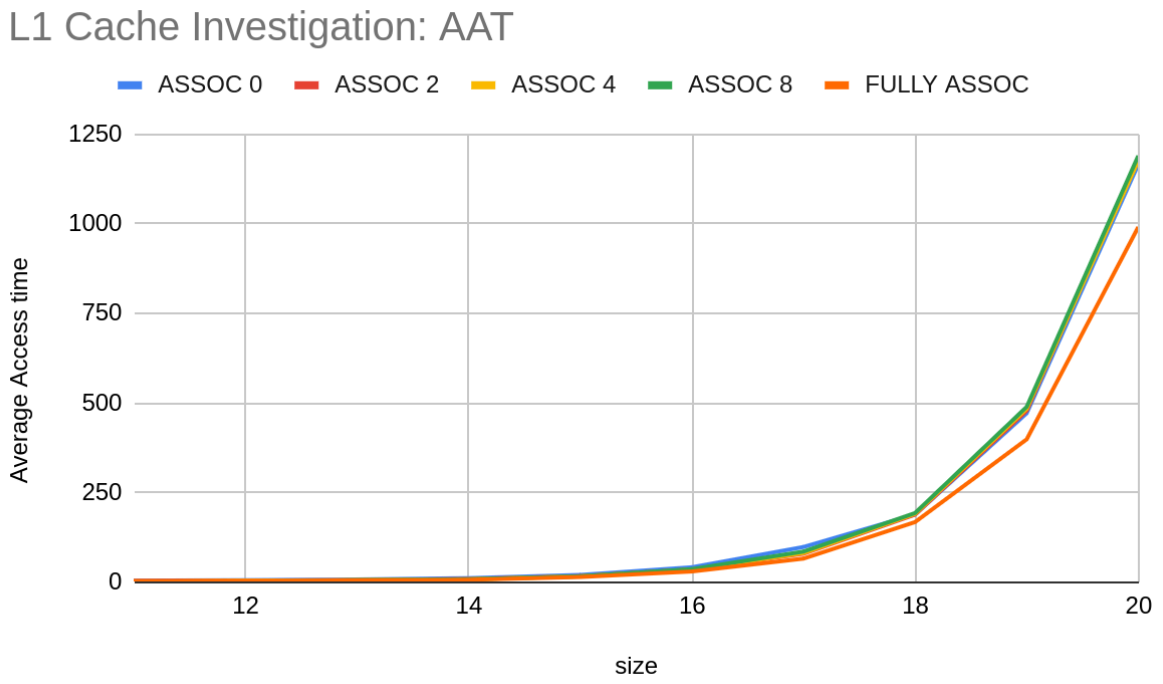
For the fully associative cache the conflict miss is always 0.

**Plot #2:**

**Similar plot #1, but plot AAT on the y-axis instead of L1 miss rate.**
**Discussion to include in your report:**
**1. For a memory hierarchy with only an L1 cache and BLOCKSIZE = 32, which configuration yields the best AAT?**

| size | ASSOC 0 | ASSOC 2 | ASSOC 4 | ASSOC 8 | FULLY ASSOC |
|---|---|---|---|---|---|
| 11 | 3.175026563 | 2.885510765 | 0 | 0 | 2.308062605 |
| 12 | 4.523478463 | 3.939102183 | 3.421976793 | 0 | 2.49316946 |
| 13 | 6.542527488 | 5.353071429 | 4.893213837 | 5.102023023 | 4.021284537 |
| 14 | 10.13575957 | 8.255671373 | 7.192983773 | 7.291602915 | 5.625642275 |

| | | | | | |
|---|---|---|---|---|---|
| 15 | 19.12461121 | 16.3930647 | 15.37138775 | 15.65318832 | 13.44203176 |
| 16 | 40.50680035 | 34.83074893 | 34.31170865 | 35.57262109 | 28.69582713 |
| 17 | 96.97326185 | 78.73621994 | 80.31926673 | 84.49018439 | 64.63279121 |
| 18 | 187.7213328 | 187.8027176 | 189.7259839 | 192.018667 | 166.4675125 |
| 19 | 470.0887903 | 477.8259275 | 482.9536999 | 488.1779831 | 397.3096576 |
| 20 | 1164.544745 | 1184.6848 | 1174.647351 | 1188.691317 | 989.5469627 |




The Fully Associative Cache shows consistently lower AAT values at smaller sizes (11 and 12).As the cache size increases (sizes 13 and above), the differences in AAT values become more pronounced, but the Fully Associative Cache retains a competitive edge until size 15.
At larger sizes (16 to 20), all configurations experience significantly higher AATs, indicating increased latency as cache sizes grow.

Higher associativity configurations begin to show diminishing returns in terms of AAT, suggesting that while they reduce conflict misses, the complexity may lead to increased access times as well.The Fully Associative Cache yields the best AAT for smaller cache sizes (11 and 12), benefiting from its ability to eliminate conflict misses.L1 cache hierarchy with BLOCKSIZE = 32, the Fully Associative Cache configuration generally provides the best average access time.

**9.2 L1 Cache Investigation: SIZE and BLOCKSIZE**

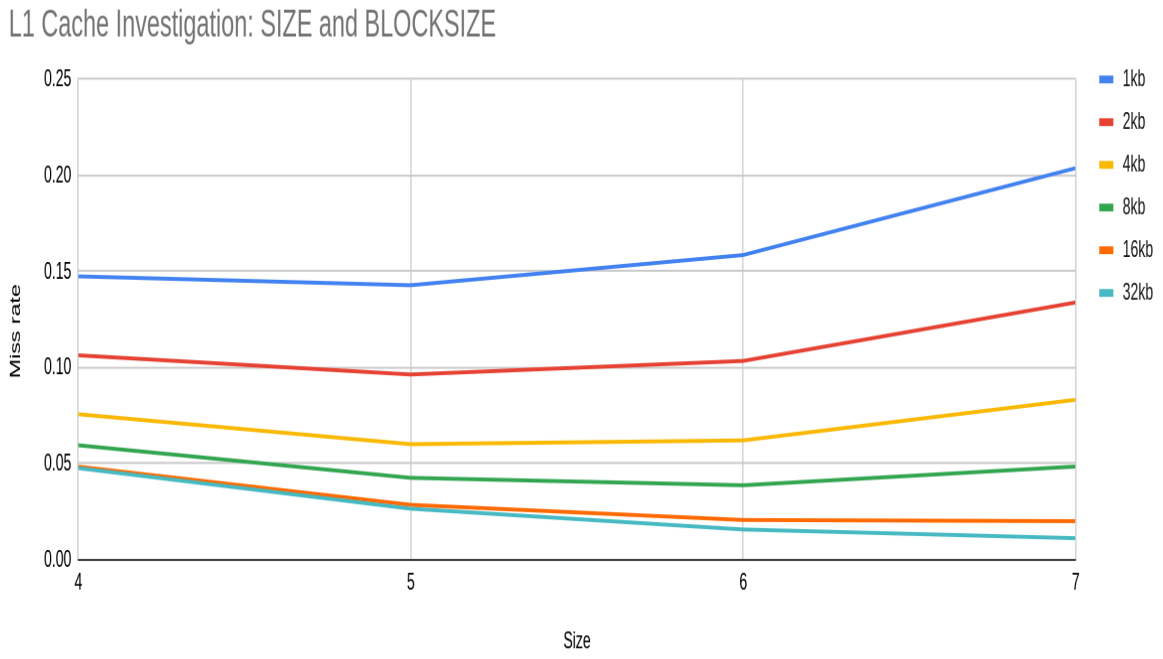
**Plot #4:**

For this experiment, use ASSOC = 4 and vary both the SIZE and the BLOCKSIZE. Assume that there is no L2 cache.

Plot L1 miss rate on the y-axis versus log2(BLOCKSIZE) on the x-axis, for four different block sizes: BLOCKSIZE = 16, 32, 64, and 128. The plot should contain six separate curves, one for each of the following L1 cache sizes: SIZE = 1KB, 2KB, …, 32KB.

Discussions to include in your report: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? Does the spatial locality-cache pollution trade-off depend on the cache size?

| size | 1kb | 2kb | 4kb | 8kb | 16kb | 32kb |
|---|---|---|---|---|---|---|
| 4 | 0.14731 | 0.10621 | 0.07554 | 0.05945 | 0.04823 | 0.04754 |
| 5 | 0.1427 | 0.09622 | 0.05992 | 0.04247 | 0.02832 | 0.0264 |
| 6 | 0.15839 | 0.10326 | 0.06189 | 0.03859 | 0.02045 | 0.01558 |
| 7 | 0.20363 | 0.1337 | 0.08303 | 0.04835 | 0.01982 | 0.01107 |




Trade-off Between Spatial Locality and Cache Pollution

Spatial Locality: As the block size increases from 16 to 128 bytes, the amount of contiguous data fetched with each cache miss increases. This enhances the likelihood of subsequent accesses hitting in the cache, thus improving the hit rate.

Cache Pollution: However, larger block sizes can lead to cache pollution, where less frequently accessed data evicts more useful data. In smaller cache sizes (like 1KB and 2KB), this effect is particularly pronounced as the cache is filled quickly, leading to higher miss rates when block sizes are increased.

For smaller caches, the miss rate may initially decrease with larger block sizes due to better spatial locality but starts to rise again as block sizes increase significantly, indicating cache pollution.

Larger caches (like 16KB and 32KB) show a more favourable trend, as they can store more data fetched in larger blocks without leading to significant increases in miss rates.These configurations benefit from larger block sizes, leading to consistently lower miss rates. They can accommodate more data fetched with larger blocks, thereby mitigating the cache pollution issue.