NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2023 (Prof. Rotenberg)

Project #1: Cache Design, Memory Hierarchy Design

REPORT

by

PADMANABHA NIKHIL BHIMAVARAPU

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

Student's electronic signature: Padmanabha Nikhil B
(sign by typing your name)

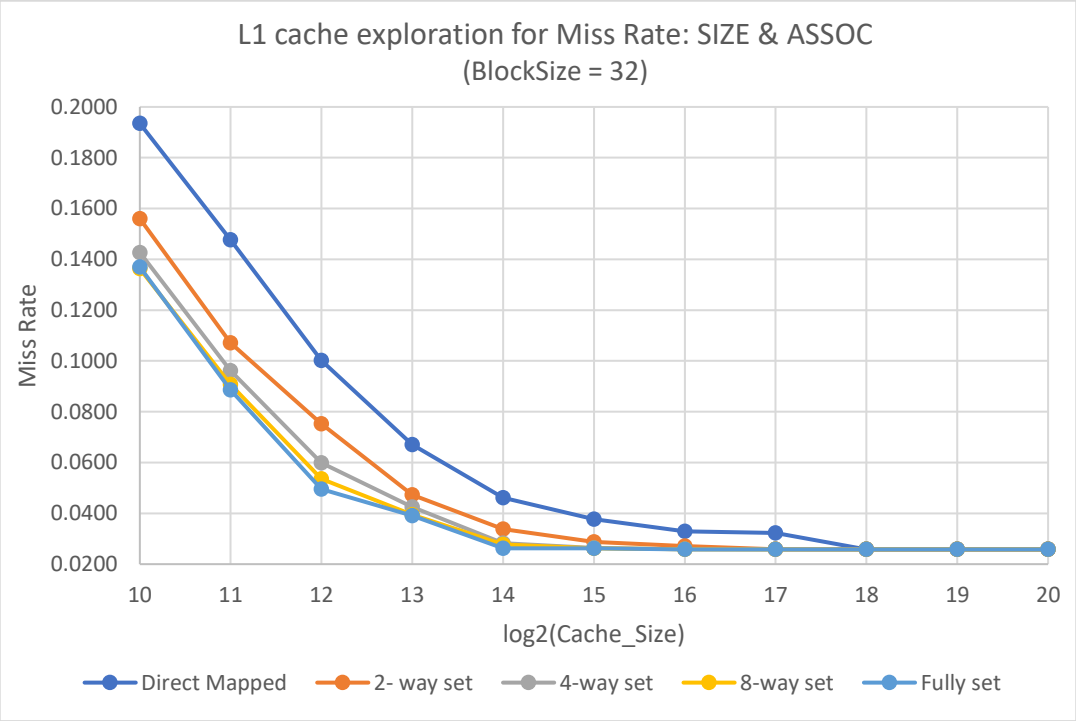
Course number: 563

1. L1 cache exploration: SIZE and ASSOC

GRAPH #1

The graph plotted is for L1 miss rate on Y-axis and $\log_2$(Cache Size) on X-axis for different values of associativity. The Block Size of the cache is set to 32B. We vary the cache size from 1KB to 1MB and the associativity is varied as directedly-mapped, 2-way set, 4-way set, 8-way set and full set.

L1 cache exploration for Miss Rate: SIZE and ASSOC (BlockSize = 32)											
Cache Size	**1024**	**2048**	**4096**	**8192**	**16384**	**32768**	**65536**	**131072**	**262144**	**524288**	**1048576**
log2(Cache Size)	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Direct Mapped	0.1935	0.1477	0.1002	0.0670	0.0461	0.0377	0.0329	0.0323	**0.0258**	**0.0258**	**0.0258**
2- way set	0.1560	0.1071	0.0753	0.0473	0.0338	0.0288	0.0271	0.0259	**0.0258**	**0.0258**	**0.0258**
4-way set	0.1427	0.0962	0.0599	0.0425	0.0283	0.0264	0.0259	**0.0258**	**0.0258**	**0.0258**	**0.0258**
8-way set	0.1363	0.0907	0.0536	0.0395	0.0277	0.0262	0.0259	**0.0258**	**0.0258**	**0.0258**	**0.0258**
Fully set	0.1370	0.0886	0.0495	0.0391	0.0263	0.0262	**0.0258**	**0.0258**	**0.0258**	**0.0258**	**0.0258**



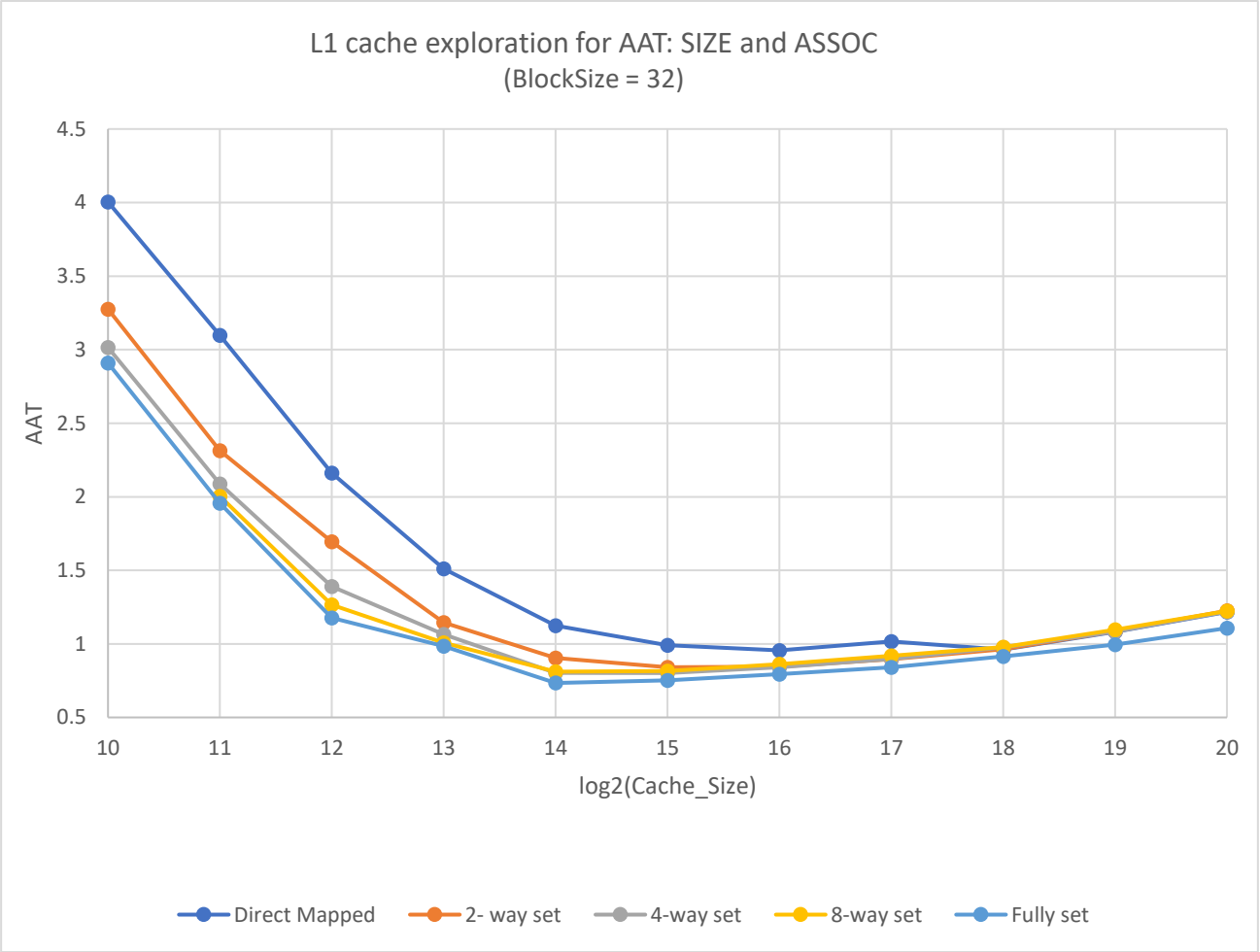
Answer the following questions:

1) For a given associativity, how does increasing cache size affect miss rate?
   For a given Associativity, as Cache Size increases the miss rate decreases.
2) For a given cache size, how does increasing associativity affect miss rate?
   For a given Cache Size, as Associativity increases, the miss rate decreases.
3) Estimate the compulsory miss rate from the graph and briefly explain how you arrived at this estimate.
   Compulsory Miss Rate = **0.0258**
   After a certain Cache Size, even as the Cache size increases, we start seeing the miss rate become constant. That constant miss rate we see is the Compulsory miss rate. The cache size becomes huge that it can store all the address bytes after the first reference and there will be no cache miss an address is reference for first time. The constant miss rate we achieve is bolded in the table above.

GRAPH #2

The graph plotted is for AAT for L1 Cache on Y-axis and $\log_2$(Cache Size) on X-axis for different values of associativity. The Block Size of the cache is set to 32B. We vary the cache size from 1KB to 1MB and the associativity is varied as directedly-mapped, 2-way set, 4-way set, 8-way set and full set.

L1 cache exploration for AAT: SIZE and ASSOC (BlockSize = 32)											
Cache Size	**1024**	**2048**	**4096**	**8192**	**16384**	**32768**	**65536**	**131072**	**262144**	**524288**	**1048576**
log2(Cache Size)	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Direct Mapped	4.004147	3.09786	2.161025	1.51053	1.125027	0.991123	0.955917	1.01603	0.962392	1.082031	1.21796
2- way set	3.275929	2.314401	1.694661	1.144925	0.903297	0.841326	0.845437	0.895193	0.964509	1.086324	1.224626
4-way set	3.01509	2.088116	1.389675	1.065423	0.802766	0.80189	0.840071	0.89886	0.976265	1.082998	1.218187
8-way set		2.003756	1.266425	1.006861	0.811124	0.815131	0.861803	0.919816	0.977505	1.096757	1.224399
Fully set	2.909184	1.957375	1.177898	0.984491	0.734238	0.75136	0.794861	0.841066	0.914589	0.994308	1.107054



Answer the following question:

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

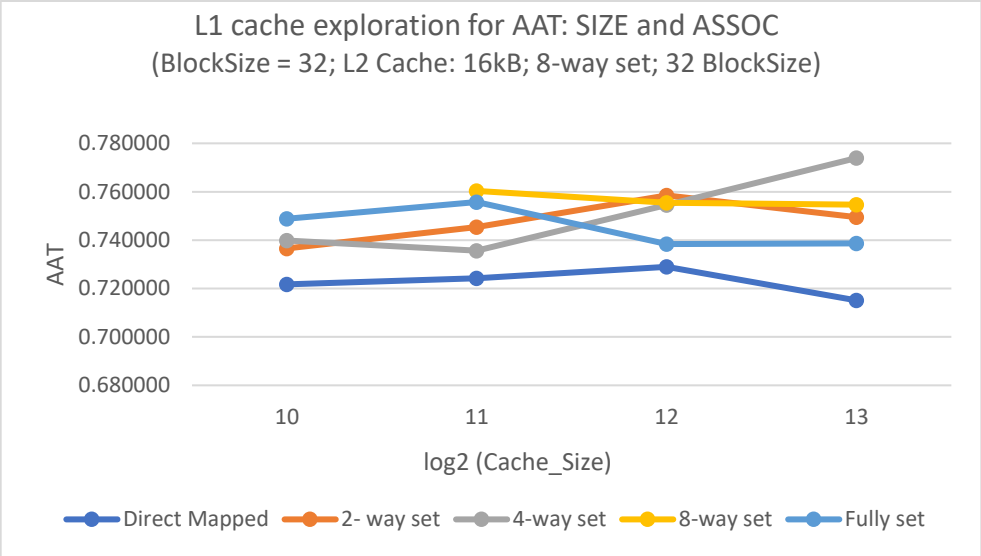
The Lowest AAT we achieve is **0.734238 ns** for the cache is with **Cache Size = 16384 and Associativity of 512 (In this case it is a Fully Associative cache)**. Among all the different set associative caches, the fully set cache gives the lowest AAT,

As we increase the Cache Size, the AAT also goes increasing.

GRAPH #3

The graph plotted is for AAT for Cache with L2 on Y-axis and $\log_2$(Cache Size) on X-axis for different values of associativity along with a L2 Cache. The L2 Cache is of 16KB and 8-way set associative. The Block Size of the cache is set to 32B. We vary the L1 cache size from 1KB to 8KB and the associativity is varied as directedly-mapped, 2-way set, 4-way set, 8-way set and full set.

L1 cache exploration for AAT: SIZE and ASSOC (BlockSize = 32) L2 Cache: 16kB, 8-way set, 32 BlockSize				
Cache Size	**1024**	**2048**	**4096**	**8192**
log2(Cache Size)	**10**	**11**	**12**	**13**
Direct Mapped	0.721747	0.724193	0.728965	0.715108
2- way set	0.736577	0.745408	0.758474	0.749473
4-way set	0.739848	0.735654	0.754515	0.773914
8-way set	NaN	0.760339	0.755492	0.754666
Fully set	0.748781	0.755748	0.738383	0.738701



Answer the following questions:

1) With the L2 cache added to the system, which L1 cache configuration yields the best (i.e., lowest) AAT and what is that AAT?
   The lowest AAT we receive is 0.715108 ns for the Directly Mapped 8KB size L1 Cache.
2) How does the lowest AAT with L2 cache (GRAPH #3) compare with the lowest AAT without L2 cache (GRAPH #2)?
   The lowest AAT with L2 cache is 0.019130392 ns less than the lowest AAT without L2 cache.
3) Compare the total area required for the lowest-AAT configurations with L2 cache (GRAPH #3) versus without L2 cache (GRAPH #2).
   Graph #2:-
   L1 Cache: Cache Size = 16KB; Block Size = 32; Associativity = *Fully Associative* (512)
   Graph #3:-
   L1 Cache: Cache Size = 8KB; Block Size = 32; Associativity = 1
   L1 Cache: Cache Size = 16KB; Block Size = 32; Associativity = 8

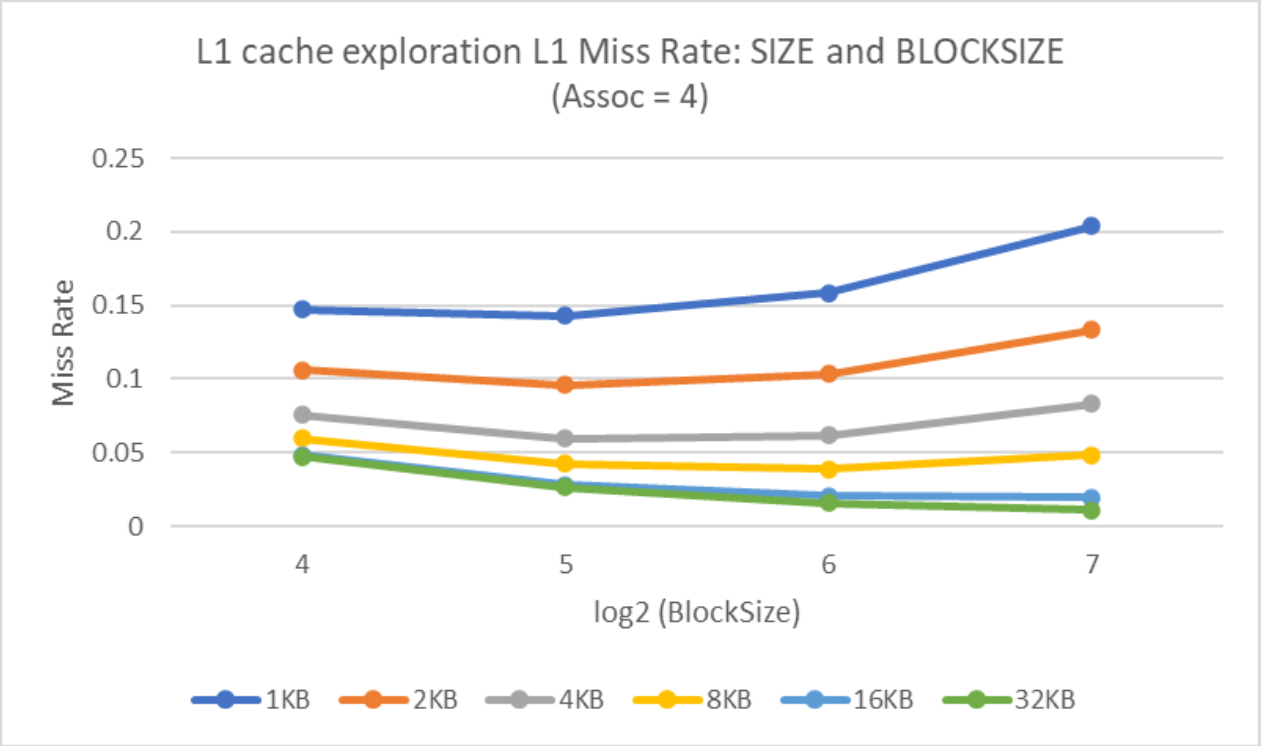
   Total Area for lowest AAT with L2 Cache = (L1 Area) + (L2 Area)
   = 0.053293238 mm2 + 0.130444675 mm2
   = 0.183737913 mm^2
   Total Area for lowest AAT without L2 cache = 0.063446019 mm^2
   The total Area for lowest AAT with L2 cache is 0.120291894 mm^2 i.e., 189.5972291 % greater than the AAT of cache without L2.

2. L1 cache exploration: SIZE and BLOCKSIZE

GRAPH #4

The graph plotted is for Miss Rate of L1 Cache on Y-axis and $\log_2$(Block Size) on X-axis for different values of Cache Size. The L2 Cache is not present. The Associativity of the cache is set to 4. We vary the block size as 16, 32, 64, 128 and the cache size is varied from 1KB to 32KB.

L1 cache exploration: SIZE and BLOCKSIZE (Assoc = 4)						
Cache Size	**1024**	**2048**	**4096**	**8192**	**16384**	**32768**
log2(Block Size)	**1KB**	**2KB**	**4KB**	**8KB**	**16KB**	**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



Answer the following questions:

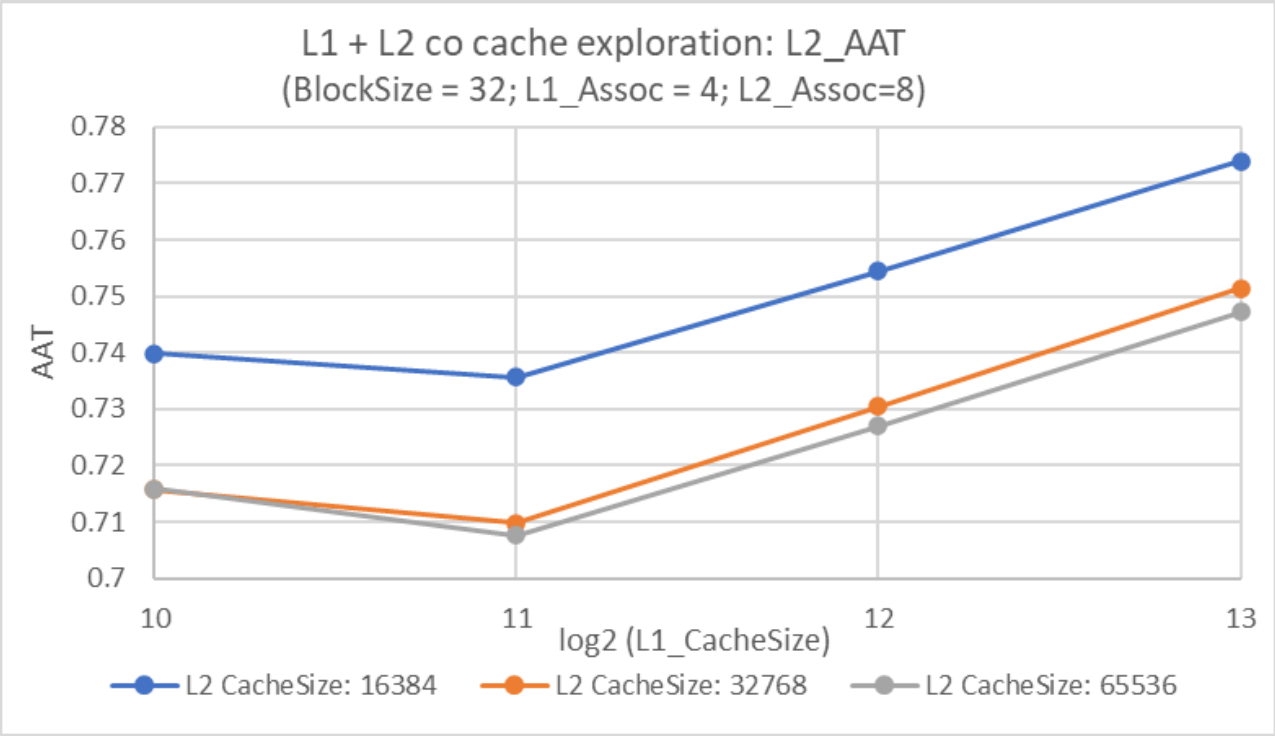
1) Do smaller caches prefer smaller or larger block sizes?
   Smaller caches prefer smaller block sizes. For example, the smallest cache considered in Graph #4 (1KB) achieves its lowest miss rate with a block size of 32 B.
2) Do larger caches prefer smaller or larger block sizes?
   Larger caches prefer larger block sizes. For example, the largest cache considered in Graph #4 (32KB) achieves its lowest miss rate with a block size of 128 B.
3) As block size is increased from 16 to 128, is the tension between exploiting more spatial locality and cache pollution evident in the graph? Explain.
   For example, consider the smallest (1KB) cache in Graph #4. Increasing block size from 16 B to 32 B is helpful (reduces miss rate) due to exploiting more spatial locality. But then increasing block size further, from 32 B to 64 B, is not helpful (increases miss rate) due to cache pollution having greater effect.

3. L1 + L2 co-exploration

GRAPH #5

The graph plotted is for AAT of Cache on Y-axis and $\log_2$L1 Cache Size) on X-axis for different values of L2 Cache Size. The L2 Cache is set to Associativity 8. The Block Size of the cache is set to 32. We vary the L2 Cache Size as 16KB, 32KB and 64KB and the L1 cache size is varied from 1KB to 8KB.

L1 + L2 co cache exploration: AAT (BlockSize = 32; L1_Assoc = 4; L2_Assoc=8)				
L1 Cache Size	**1024**	**2048**	**4096**	**8192**
log2(L1 Cache Size)	**10**	**11**	**12**	**13**
L2 CacheSize: 16384	0.739847523	0.735654053	0.754515407	0.77391397
L2 CacheSize: 32768	0.7157522	0.709742294	0.730434828	0.751446643
L2 CacheSize: 65536	0.715815273	0.707657833	0.726969733	0.747194178



Answer the following questions:

Which memory hierarchy configuration in Graph #5 yields the best (i.e., lowest) AAT and what is that AAT?

The Lowest AAT we achieve is 0.707657833 for the cache with L1 Cache Size of 2KB and L2 cache size of 64KB.

4. Stream buffers study (ECE 563 students only)

TABLE #1 (***total number of simulations: 5***)

For this experiment:

- Microbenchmark: stream_trace.txt
- L1 cache: SIZE = 1KB, ASSOC = 1, BLOCKSIZE = 16.
- L2 cache: None.
- PREF_N (number of stream buffers): 0 (pref. disabled), 1, 2, 3, 4
- PREF_M (number of blocks in each stream buffer): 4

The trace "stream_trace.txt" was generated from the loads and stores in the loop of interest of the following microbenchmark:

```
#define SIZE 1000

uint32_t a[SIZE];
uint32_t b[SIZE];
uint32_t c[SIZE];

int main(int argc, char *argv[]) {
…
   // LOOP OF INTEREST
   for (int i = 0; i < SIZE; i++)
      c[i] = a[i] + b[i];   // per iteration: 2 loads (a[i], b[i]) and 1 store (c[i] = …)
…
}
```

Fill in the following table and answer the following questions:

N, PREF_M	rate
f. disabled)	

1. For this streaming microbenchmark, with prefetching disabled, do L1 cache size and/or associativity affect the L1 miss rate (feel free to simulate L1 configurations besides the one used for the table)? Why or why not?

With prefetching disabled, L1 cache size and/or associativity << do / do not >> affect L1 miss rate (for this streaming microbenchmark).

The reason: _____.

2. For this streaming microbenchmark, what is the L1 miss rate with prefetching disabled? Why is it that value, *i.e.*, what is causing it to be that value? Hint: each element of arrays a, b, and c, is 4 bytes (uint32_t).

The L1 miss rate with prefetching disabled is _____, because _____.

3. For this streaming microbenchmark, with prefetching disabled, what would the L1 miss rate be if you doubled the block size from 16B to 32B? (hypothesize what it will be and then check your hypothesis with a simulation)

The L1 miss rate with prefetching disabled and a block size of 32B is _____, because _____.

4. With prefetching enabled, what is the minimum number of stream buffers required to have any effect on L1 miss rate? What is the effect on L1 miss rate when this many stream buffers are used: specifically, is it a modest effect or huge effect? Why are this many stream buffers required? Why is using fewer stream buffers futile? Why is using more stream buffers wasteful?

Minimum number of stream buffers needed to have any effect on L1 miss rate: _____

With this many stream buffers, the effect on L1 miss rate is << modest / huge >>. Specifically, the L1 miss rate is nearly _____. We only miss on the first elements of _____ (hence a total of _____ misses).

This many stream buffers are required because _____.

Using fewer stream buffers is futile because _____.

Using more stream buffers is wasteful because _____.