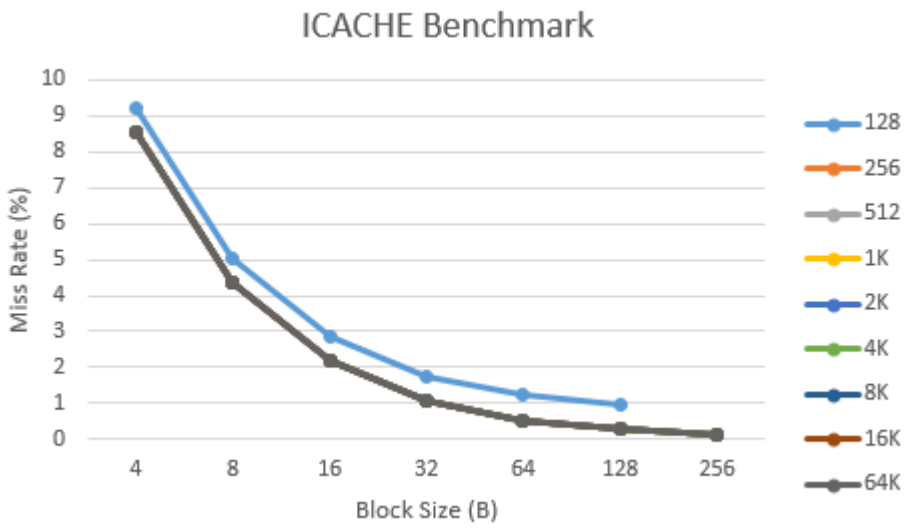
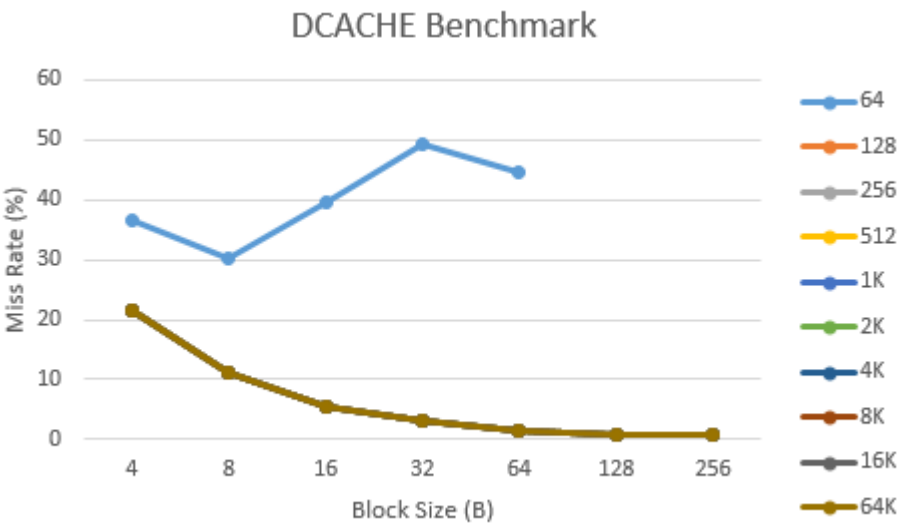
Computer Organization

0610746 0610862

Basic Problem





In both diagram, we can see that generally larger block size decrease the miss rate. This is because of spatial locality.

For cache size, the miss rate decreases as the cache size become bigger since we reduce the capacity misses. In other word, we have more blocks to store data, so we have less misses.

Advanced Problem

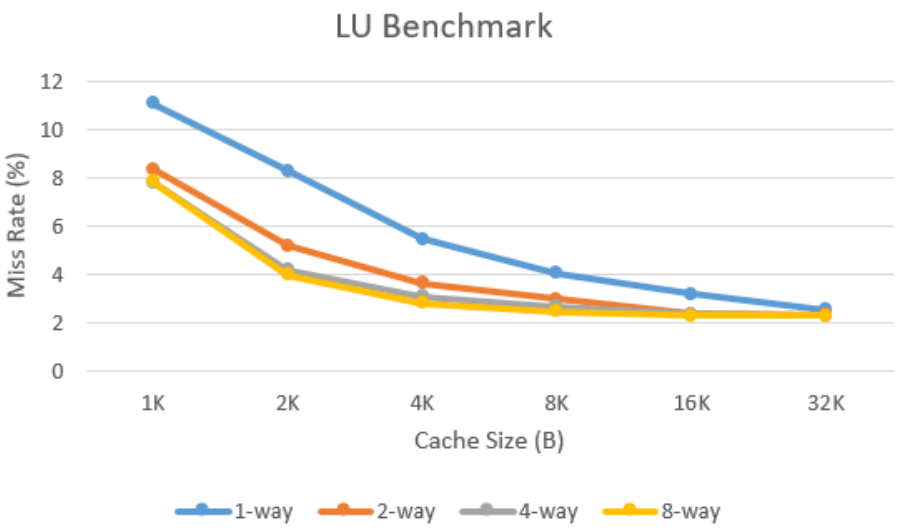
Total bits needed for the one cache line = 64*8 + 1 + (32 - index_bit - offset_bit)

Associativity / Cache Size	1-way	2-way	4-way	8-way
1K	535	536	537	538
2K	534	535	536	537
4K	533	534	535	536
8K	532	533	534	535
16K	531	532	533	534
32K	530	531	532	533

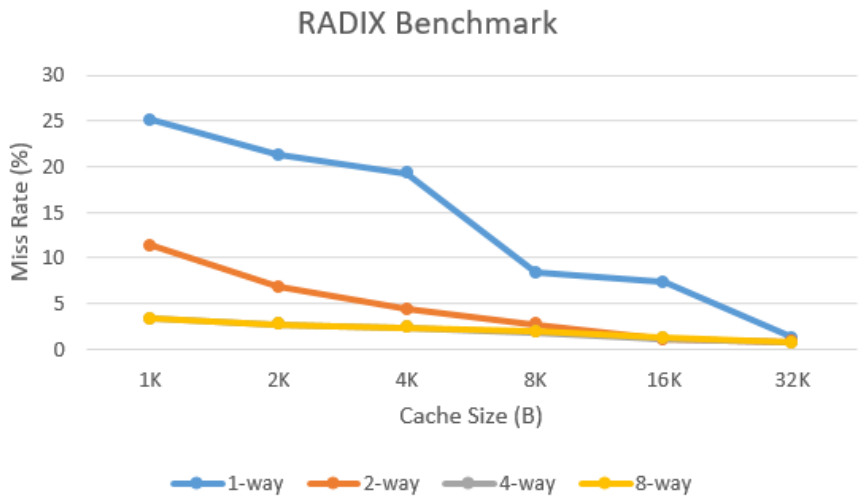
In the instruction PDF, I'm not sure if the TA require us to include the data bits or not for the calculation. For safety, I included it. To get the total bits without data bit, just minus 512bits from the answer.

The only variable in our calculation is the tag bit because we already fix the block size. For cache with 1 increase in associativity, we need 1 more bit for tag because now the block that we are able to use for each memory location is now doubled. The other way we can see this is as associativity increase, index-bit decrease, hence, tag bit increase.
For larger cache size, we need more index bit for each of the cache line, therefore, tag bit is lesser.

Next we are going to see the benchmark (miss rate) of set associative cache.



In LU benchmark, we see that the miss rate decrease as the cache size and associativity increase. This is because as the associativity increase, the program is able to utilise cache more efficiently as it does not need to be replaced as often. Other than that, cache size increase also allowed more memory to be cached and in turns the cache does not need to be replaced that often, thus decreasing the miss rate.



In RADIX benchmark, we see a similar trend compared to LU benchmark. The miss rate decrease as the cache size and associativity increase. Note that for 8-way set associative cache, the decrease in miss rate as the cache size increase is insignificant. We conclude that as n (n-way) increase, the increase in cache size is not important, at least for this particular program.