Assignment 5: Cache Design | Alexander Stradnic - 119377263

"In this lab, we would like to explore the cache performance as we change the cache design parameters. We will do so while executing the provided matrix multiplication code *matrix-vector-multiplication.asm.*

Fill in the table with the achieved ***hit ratio*** using the provided information from the Data Cache Simulator tool (Tools → Data Cache Simulator)

Parameter	Scenario 1	Scenario 2	Scenario 3
Cache block size (words)	2,4,8,16,32,64,128	2,4,8,16,32,64,128	2,4,8,16,32,64
Cache size (bytes)	256, 512, 1024	256, 512, 1024	512
Placement policy	Direct Mapped	Full Associative	n-Way associative
Set size (blocks)	-	-	2,4,8
Block replacement policy	-	LRU, Random	LRU

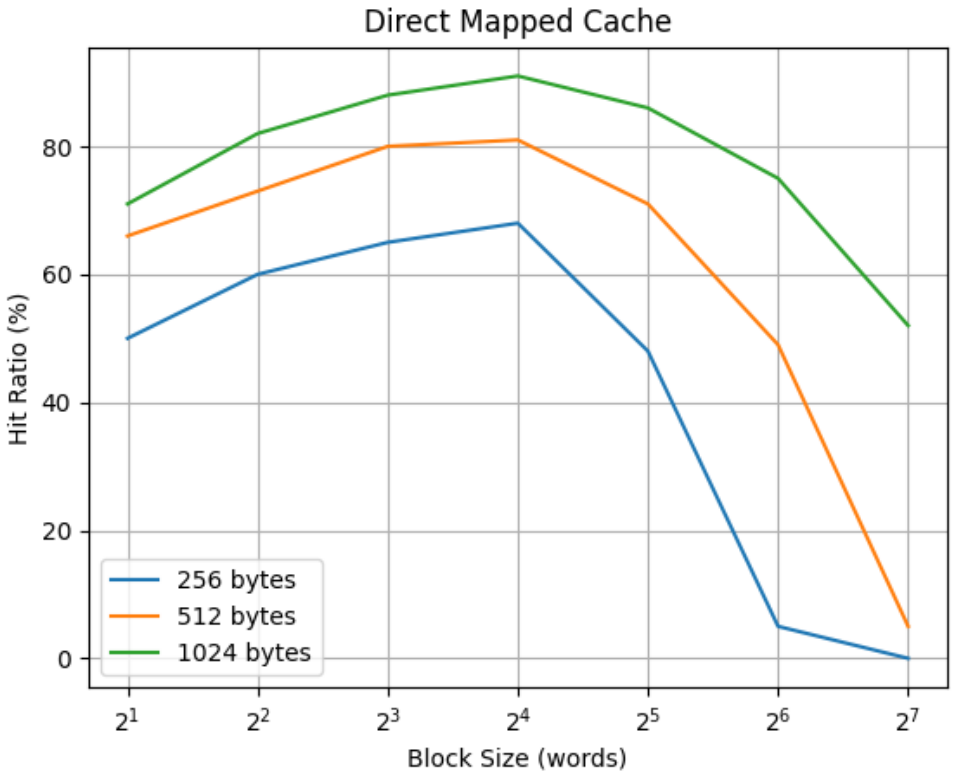
Note that as you change the cache block size, you would need to adjust the number of blocks to maintain a fixed cache size."

Scenario 1: Direct mapped cache

We only test the impact of cache size and block size. Note that in direct-mapped cache you have a rigid placement and replacement strategy (revise lecture slides if not sure)

Block size (words) Cache Size (bytes)	2	4	8	16	32	64	128
256	50%	60%	65%	68%	48%	5%	x
512	66%	73%	80%	81%	71%	49%	5%
1024	71%	82%	88%	91%	86%	75%	52%

Table 1.1: Direct Mapped Cache

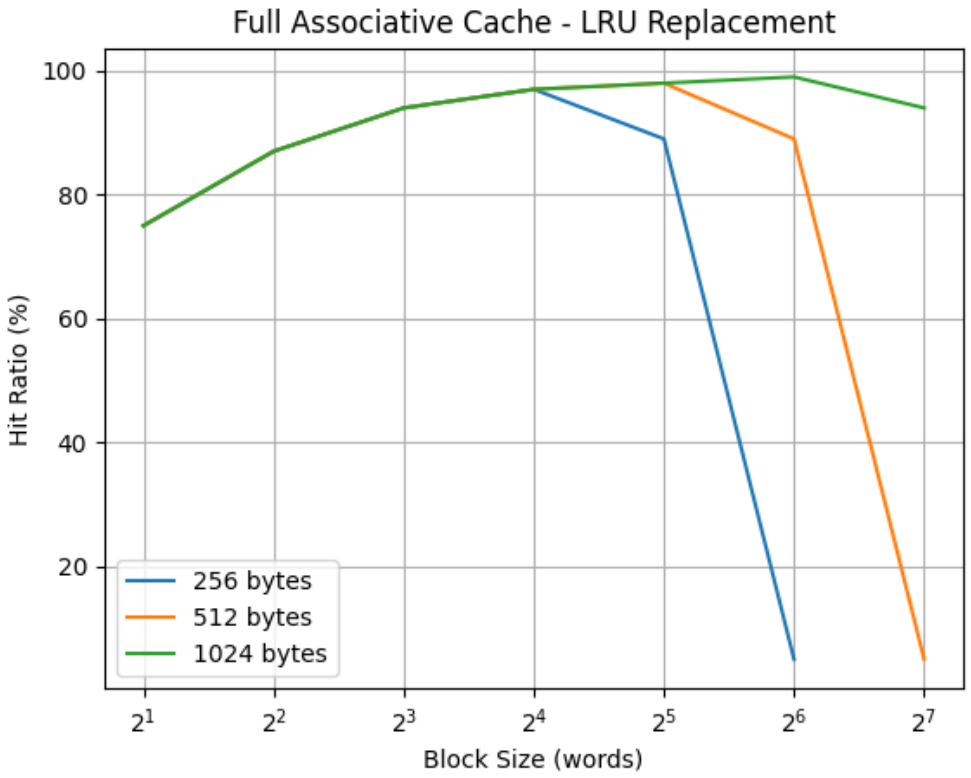


Scenario 2: Full Associative Cache

LRU replacement

Block size (words) Cache Size (bytes)	2	4	8	16	32	64	128
256	75%	87%	94%	97%	89%	5%	x
512	75%	87%	94%	97%	98%	89%	5%
1024	75%	87%	94%	97%	98%	99%	94%

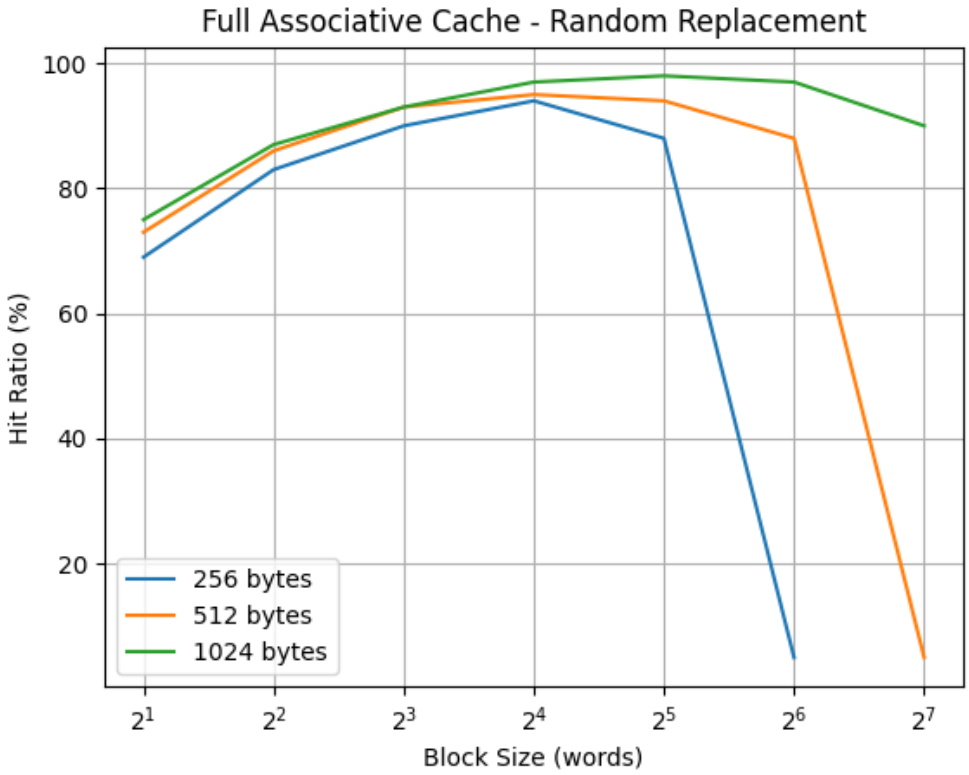
Table 2.1: Full Associative Cache with LRU replacement



Random replacement

Block size (words) Cache Size (bytes)	**2**	**4**	**8**	**16**	**32**	**64**	**128**
256	69%	83%	90%	94%	88%	5%	x
512	73%	86%	93%	95%	94%	88%	5%
1024	75%	87%	93%	97%	98%	97%	90%

Table 1.2: Full Associative Cache with random replacement

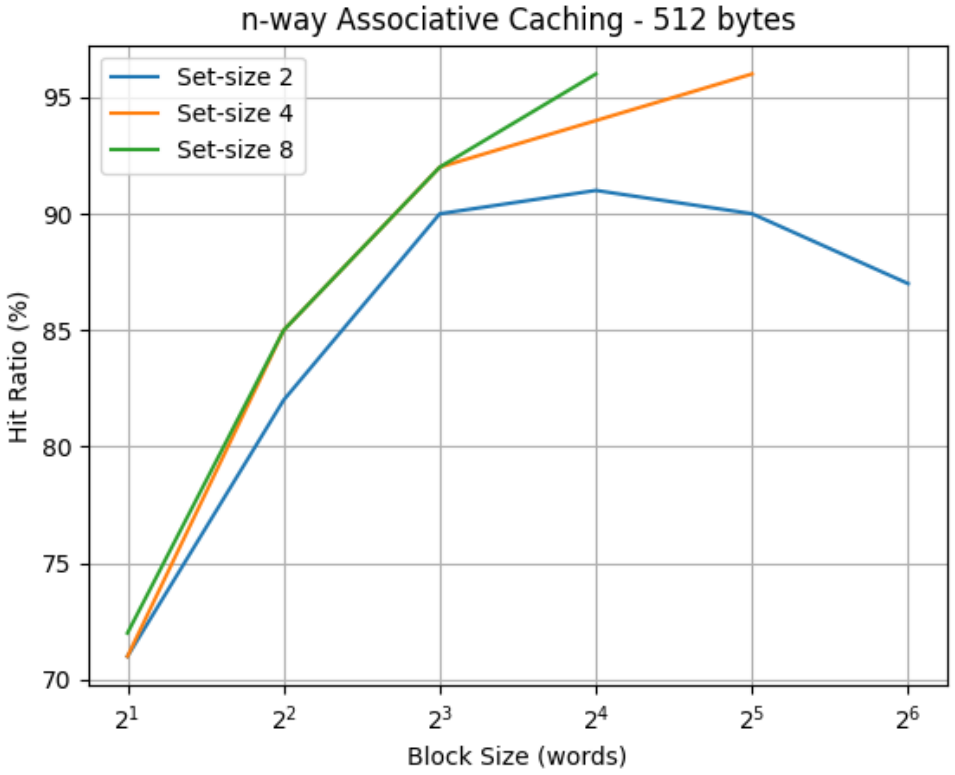


Scenario 3: n-way Associative Caching

Cache Size 512 Bytes

Block size (words) set-size	2	4	8	16	32	64	128
2	71%	82%	90%	91%	90%	87%	x
4	71%	85%	92%	94%	96%	x	x
8	72%	85%	92%	96%	x	x	x

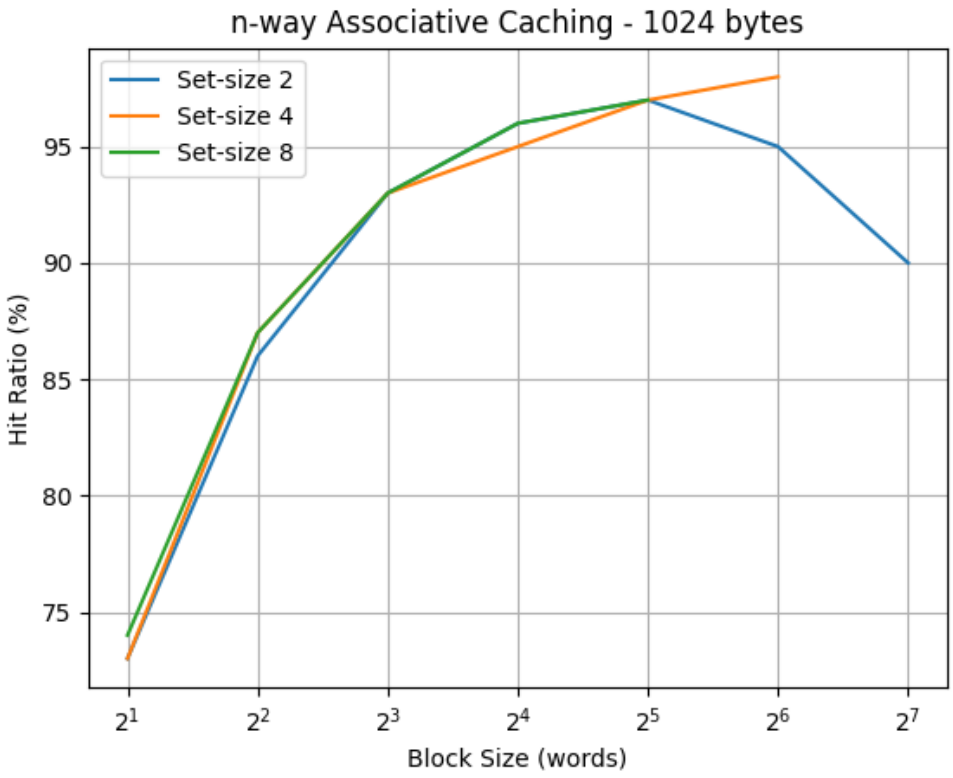
Table 3.1: n-Way associative (512 Kbyte)



Cache Size 1024 Bytes

Block size (words) set-size	2	4	8	16	32	64	128
2	73%	86%	93%	96%	97%	95%	90%
4	73%	87%	93%	95%	97%	98%	x
8	74%	87%	93%	96%	97%	x	x

Table 3.1: n-Way associative (512 Kbyte)



Discussion

1. What is the impact of cache size on the cache performance? What is the downside of using larger cache?

   Having a larger cache size increases the cache hit ratio to an extent but it is also important to divide the cache into an appropriate amount of blocks for it to be effective, otherwise it has a worse miss penalty.

2. Discuss the impact of block size of cache performance illustrating
   a) Why we have a small hit ratio for small block sizes?

      This is because smaller blocks increase the miss rate due to spatial locality, however they have a smaller miss penalty as compared to larger blocks.

   b) Why we have a small hit ratio for large block sizes?

      This is because the block size is too large for the cache size and so it cannot store as many blocks, so the hit ratio is reduced.

c) What is the optimal block size for various cache configurations?

The optimal block size depends on the size of the cache as well type of caching being used. From my results, a cache block size of between 16-32 words was adequate for the cache sizes tested.

3. How Associative caching perform in comparison to direct mapped caching? Why? What is the downside of using Associative cache?

It has a better hit ratio compared to direct mapping caching because of its more flexible block location. The downside is that you have to search the whole cache so it takes longer to actually hit the cache.

4. Discuss the performance of LRU and Random replacement policies for various cache configurations.

The hit ratio performance of Random Replacement suffers compared to LRU Replacement, especially at lower cache sizes but it is approaches LRU hit rates at higher cache sizes.

5. Discuss the impact of set size in n-way associative cache. (Hint: consider a specific cache size)

For 512 bytes, the hit ratio was impacted quite substantially for set-size 2, as compared to the set-size 4 and 8. Set-size 4 was also behind set-size 8 but to a lesser degree.

In contrast, for cache size 1024 bytes, set-size 2 was still the worst but not nearly as much as in 512-byte size cache.

6. Do you believe these conclusions would change if the array size if different? Why? (You may conduct more simulations)

I think that the results would be consistent for other array sizes in that increasing the set-size will always improve hit rate, but this comes at a computer resource cost.