Peter Pettino
CSE 3666
15 April, 2024

Lab 9

Task 1:

In order to get the number of bits in block offset, we must calculate the block size in bytes: $4\ words\ *\ 4\frac{bytes}{word} = 16\ bytes$. The block offset determines the specific word within a cache block. Since the block size is 16 bytes, we need 4 bits to represent any byte within a block as $2^4 = 16$. You determine the number of bits in the cache index with the equation: $2^n = number\ of\ blocks$, where n is the number of bits. Since we have 8 blocks in total, the number of bits needed for the cache index is 3. The tag is calculated by subtracting the number of bits in the block offset and cache index from 32, therefore the tag contains 25 bits.

Address	Cache Index	Tag	Block Offset	Hit/Miss
0x10010000	0	0x00200200	0	Miss
0x10010004	0	0x00200200	4	Hit
0x10010008	0	0x00200200	8	Hit
0x1001000C	0	0x00200200	12	Hit
0x10010010	1	0x00200200	0	Miss
0x10010014	1	0x00200200	4	Hit
0x10010018	1	0x00200200	8	Hit

Explain the cache index, tag, and block offset for the first 2 accesses

For the first access (0x10010000):

- Block Offset: 0, determined by the four least significant bits of the address
- Cache Index: 0, determined by the three bits before the block offset
- Tag: 0x00200200, determined by the 25 most significant bits of the address

For the second access (0x10010004):

- Block Offset: 4. determined by the four least significant bits of the address
- Cache Index: 0, determined by the the three bits before the block offset
- Tag: 0x00200200, determined by the 25 most significant bits of the address

Explain the hit/miss outcomes for each memory access recorded in the table

0x10010000: Miss, this is the first access to this block, so it must be fetched from memory
0x10010004: Hit, the block is already in the cache from the previous access
0x10010008: Hit, still the same block in the cache
0x1001000C: Hit, still the same block in the cache
0x10010010: Miss, a new block is being accessed, so it must be fetched from memory
0x10010014: Hit, the block was fetched on the previous access
0x10010018: Hit, still the same block in the cache

Explain the cache hit rates you observed
Based on the table, there are 2 misses and 5 hits out of the 7 accesses. This means the hit rate for this sample is 71%. However, from what was observed, every miss is followed by 3 hits thus a 3:4 ratio. Therefore, the overall hit rate of the cache is 75%.

Task 2:

Number of Blocks	Cache Size	Hit Rate	Miss Count
8	128	75%	3072
16	256	75%	3072
32	512	75%	3072
64	1024	99%	64
128	2048	99%	64

Explanations for the cache hit rates and miss counts you observed
For cache sizes 128, 256, and 512 bytes, the hit rate is 75%, and the miss count is 3072. However, when the cache size is increased to 1024 bytes or greater, the hit rate increases to 99% with only 64 misses. This may indicate that the program can almost fit entirely within the cache which results in very few misses after the initial cache misses. I was unable to predict the high hit rates of 99% for the larger cache sizes as the initial information was insufficient to determine the memory access patterns of the program.

If the size of warray is doubled
Checking with the simulator:

Number of Blocks	Cache Size	Hit Rate	Miss Count
8	128	75%	3072
16	256	75%	3072

32	512	75%	3072
64	1024	75%	3072
128	2048	99%	128

According to the simulator, if the size of warray were to be doubled, the rows in the table that would have the same data would be those that contain cache sizes 128, 256, and 512 bytes and the hit rate and miss count would remain unchanged. This is because the program may not fit entirely within the cache thus resulting in the same behavior as before. For the cache size of 1024 bytes, the hit rate has decreased from 99% to 75% and the miss count has increased from 64 to 3072 which indicates that doubling warray no longer fits within this cache size leading to more cache misses and a lower hit rate. However, for the cache size of 2048 bytes, the hit rate remains at 99% but the miss count increased from 64 to 128. This increase in miss count is expected because after the doubling the size of warray, there will be more initial misses due to the larger cache size.