Joshua Steward

Lab 3

Case 1: There are 70 hits and 30 misses

* In this memory system, the base addressable unit of memory is the Word. So, as the CPU is sequentially accessing this array of 80 words, it looks in the cache first. At A[0], it misses, because that spot is in main memory. It then moves that entire BLOCK into the first line of the cache, which will then include the next 7 words. Therefore, the next 7 words accesses are hits. When it hits the 9th word, it misses again, because it has gotten to the end of the block that was previously transferred. This continues until the end of the array without having to replace any entries, because the cache can hold 15 blocks of memory, and the CPU only needs to search through 10 of those blocks.

Case 2: There are 150 hits and 10 misses

* Since the first run through of the array didn’t fill up the cache, when it gets to the 2nd run through, all the blocks are already stored in the cache. Therefore, we only get the original 10 misses, and the other 150 are hits.

Case 3: There are 70 hits and 10 misses

* In this case, we are using the cache with 5 lines/blocks. We have 10 blocks of memory to go through in the array. So, the first 5 will fill up the cache. Each of those blocks will get 1 miss an 7 hits as it goes through. With the first in, first out replacement policy, we see that when we get to the 6th block in the array, it will replace the first block the CPU put into the cache, and so on and so forth, making the hit and miss ratio the same as in the first pass, just like the first case.

Case 4: There are 140 hits and 20 misses

* This is going to do the same thing as in case 3, as in go through the cache twice, replacing each line and getting 5 misses and 35 hits per filling of the cache. It will do this the same replacement pattern on the second pass, so effectively replacing the entire cache 4 times. So you would have 5 x 4 misses and 35 x 4 hits.

Case 5: The Optimum number of lines/blocks would be 10

* This is because if you have the exact number of blocks that need to be transferred from the array, memory will not be wasted, and you do not have to replace any.

Case 6: The Optimum number of lines/blocks would be 10

* Same as case 5, and you still wouldn’t need to replace any of the blocks because they will already be stored in the cache memory.