Comp Org
Hw5
Seth Pratt
Fidele Akakpo

1)

a. We don't access memory until line 16, inside a loop. The initial pass will be a miss, and the line + 15 more bytes will be loaded into the cache. The next 3 iterations will increment by 4, which will already be loaded into the cache. Our next miss will occur once we extend our line. So 1 miss per 4 lines. 75% hit rate.
b. The cache hit rate matches our prediction. As we stepped through the program, we observed a cache miss every 4^{th} iteration of the loop.
c. The hit rate should remain the same we still have the same size cache and same size block lines.
d. After we figured out the hit rate of fully associative, we predicted it would not change, and it did not. Direct mapping vs fully associative hit rates should remain the same, they just determine how memory in the cache is organized.
e. We predict the hit rate will still not change. We are not missing because we don't have enough blocks, we are missing when our block lines run out of space.
f. There isn't a difference in hit rate, because we have not changed the amount of data we are bringing into our cache. We could increase the number of blocks as much as we want and will still have the same hit rate until we increase the line size.
g. Our first 8 misses will be compulsory, because the cache is simply empty. Since we have more data than cache size, the remaining conflicts with be capacity misses.
h. If we change our constant 'stride' from 4 to 1, we will now only miss 1 in every 16 cache attempts. Since we are loading 4 words, and accessing each index 1 by 1, instead of skipping by 4, our hit rate will be 15/16 = 93.74%
i. Our predictions held up, as we stepped through the mars program we observed 1 miss every 16 passes.
j. Spatial locality. Because, we are referencing items memory that are stored closely and we are accessing them close in time.

2) Cache.2.s
   a. We predict the hit rate will be 75%. We will have one miss at the beginning of each block, and then 3 hits.
   b. After running the simulation, we stepped through our program and observed we had a miss at the beginning of each cache block, followed by 3 hits.
   c. The first 8 misses will be compulsory misses, since we have no data loaded into the cache, there is nothing to hit. After that, the misses will be capacity misses because our cache will be full, and needs to remove the least used.
   d. We could lb, load the data, in both locations in one loop. That way, we can access the same block in the cache, for each variable. This would mean less cache misses.

3) Cahce.3.s
    a. So we have 8 words per line, and our stride is 16, so we can fit 2 strides per line. We are going to miss on the initial cache check, because the new line has not been loaded. Then we will hit on the second stride, and so on and so forth. Hit rate = 50%
    b. The experiment lined up with our predictions exactly. We could observe the initial miss of each line, followed by the hit.
    c. The first 4 misses will be compulsory since we do not have any data loaded into the cache. After that as our cache fills up, we will have conflict misses each time we need to reuse a block.