Billy Koech and Nicolas Weinger

CS 141: Computing Hardware

Lab 5 : Caches

Brief Questions

1. Why do we use the least significant bits when building a DMC’s address to set mapping rather than the most significant bits?
   1. **Using the least significant bits takes advantage of spatial locality when mapping blocks to sets in the cache because adjacent memory blocks will be mapped to adjacent sets. If we used the most significant bits then memory blocks in close proximity would more likely map to the same set because it would take large increments in the addresses for the blocks to be mapped to different sets; Using MSBs would thus increase the miss rates.**
2. t19.test accesses memory blocks with indexes separated by multiples of 16. It only accesses 16 distinct memory blocks.
   1. Which of the three caches you built is worst at this test and why is it the worst?
      1. **DMC is the worst. This is because the number of sets for the dmc is as defined by DIRECT\_MAPPED\_NUM\_SETS is 16 therefore the address of every memory access maps to the same set which likely results in a miss. The same set is therefore more like to be overwritten every time data is accessed in the cache.**
   2. Which of the three caches you built is best at this test and why is it the best?
      1. **FAC is the best. The LRU eviction policy means that data can stay in the cache provided it’s being used frequency regardless of the data’s address in memory. Therefore, the index separation in the t19.test has little impact on the miss rate.**
3. t20.test repeats through the same 17 memory blocks.
   1. Which of the three caches you built is worst at this test and why is it the worst?
      1. **FAC is the worst. This is because FAC has a cache limit of 16 ways as defined by FULLY\_ASSOCIATIVE\_NUM\_WAYS. Therefore, by the time a memory block is repeated, the LRU policy has already evicted it due to the fact that it has not been used for the past ~16 memory accesses.**
   2. Which of the three caches you built is best at this test and why is it the best?
      1. **SAC is the best. This is because it maps the 17 blocks into different sets as defined by SET\_ASSOCIATIVE\_NUM\_SETS and each set has its own LRU eviction policy therefore by the time a memory block is repeated it’s likely that there is still space in a set for it to be stored/loaded thus decreasing the likelihood eviction.**
4. t21.test accesses memory blocks with indexes separated by multiples of 8. It repeats through the same three memory blocks.
   1. Which of the three caches you built is worst at this test and why is it the worst?
      1. **SAC is the worst. This is because the set limit as define by SET\_ASSOCIATIVE\_NUM\_SETS is 8 and the way limit as defined by SET\_ASSOCIATIVE\_NUM\_WAYS is 2 therefore the same three memory blocks get mapped to the same set and can’t all fit because the limit is 3 ways. Therefore by the time one of the three blocks is repeated, the LRU policy has already evicted it due to the fact that it has not been used for the past 2 memory accesses.**
   2. Which of the three caches you built is best at this test and why is it the best?
      1. **FAC is the best. This is because FAC has a cache limit of 16 ways as defined by FULLY\_ASSOCIATIVE\_NUM\_WAYS therefore each of the three blocks can be stored in a way and don’t have to be evicted because the cache never gets full.**