CS2610: Computer Organization and Architecture Lab
Assignment #4

Due Date: 28th Feb

Objective: **To identify the cache block size and the associativity of the L1 cache.**

Team Size: 2

Problem Statement: Reverse Engineering of L1 Cache Memory.

We know that a cache hit takes less access latency as compared to a cache miss. By exploiting this fact, you need to identify the cache block size and the associativity of the L1 cache present in your laptop/desktop. Consider the L1 cache is initially empty and initiate a LOAD request to some address X, which brings a block of data into the L1 cache. Now keep on generating a sequence of LOAD requests to neighboring addresses of X and measure the access latency for each request. If the access latencies are more or less equal, the corresponding addresses are from the same cache block. If the access latency for a LOAD request to some address Y is very high as compared to that of other addresses, it indicates that the load request to address Y is a miss in the cache. For example, consider an L1 cache that contains a 32B cache block at address 0. This cache can give hits for any load request for address X, where 0<=X<=31. A load request to any address outside this address range will incur a cache miss. By repeating this process multiple times, and by observing variation in the access latency, we can infer the cache block size.

Having found the block size, let us find the associativity of the cache. All we need is to generate LOAD requests with addresses such that all these addresses bring cache blocks that belong to the same set. If the cache is having 4 sets, and each cache block is 32B, we know that the cache blocks that are brought for LOAD requests to addresses 0, 128, 256, 384, 512, etc., go to the same cache set. Every time, when we bring a new cache block to a cache set, we initiate access to all the blocks that are already brought in earlier to the cache set and observe the access latency. If the access latencies are almost the same, indicating that the cache blocks are present in the cache set, we try to bring in a new cache block to the same cache set and repeat the process. At any time, if we observe high access latency when we access a cache block, indicating that the cache block is evicted by a newly brought-in cache block. In such a case, the number of blocks brought into the cache set is one more than the associativity of the cache, so we can infer the cache associativity from this observation. For example, assume that we bring-in two cache blocks, A and B. If the associativity is 2, when we access the two cache blocks, we observe the same latency. Now if we bring-in one more cache block, let us say C, either A or B must be evicted. Now, if we initiate an access request to A and B, we observe high access latency for the block that is not present in the cache.

One-page writeup explaining the procedure in detail along with the source files in proper format, must be uploaded on course moodle page.