Title : Reverse Engineering of L1 Cache memory

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

Explanation:

Firstly,we are given to compute the cache block size in our laptop.We implemented the idea that cache hit takes less access latency when compared to a cache miss.We created a array cacher of size 8192 and array of size 8192.We initialised cache with the array cacher we created.We used CPUID for serialising the out of order execution.We used rtdsc to get the processor’s time stamp counter into eax register.We initiate load request to some address with array.We generated a sequence of load requests neighbouring to previous one.We measured access latency for each.

Then printed the access latency.We kept the values of time in a file.Then plotted a graph with this file using python.There will be small variancies with the cache hit locations.We have to ignore them.We’ll see a jump in the execution time once that we loaded doesnot fit into the cache.In this way,we can determine cache block size.

Secondly,we are given to compute the associativity of cache in our laptop/desktop.We created arrays cacher of size 8192 and array of size 16385.We used CPUID for serialising the out of order execution.We used rtdsc to get the processor’s time stamp counter into eax register.We initiate a load request to some address.We generate a sequence of load requests from the same set.We measure the access latency while loading each value.

Then printed the access latency.We can see it giving peak values in access latency at intervals of 8 giving cache associativity 8.We can see getting peak values in access latency when loading nineth value.In this way,we can determine associativity of cache in our laptop.