Cache Testing Report

Eli Pardo

4/20/2024

**Introduction**

In Project 7, Cache, it was necessary to build a cache to make the speed of the Processor faster. This is done by implementing two separate caches, one is called the Instruction Cache and is “closer” to the Processor so accessing it takes 10 clock cycles. The next cache is called Level 2 Cache and is 4 times the size of the Instruction Cache, but also a little “farther away” from the Processor. Because of this fact, the Level 2 Cache takes 20 clock cycles to access and another 20 clock cycles to fill the Instruction Cache with the data from the Level 2 Cache. This means that a data access from Level 2 Cache takes 50 clock cycles. If information is not stored in the Instruction Cache or the Level 2 Cache, it must be pulled from Main Memory. Such an instance is called a cache miss and takes 350 clock cycles. By recycling what is and isn’t in the cache, one could speed up their program instead of using just the Main Memory which would take 300 clock cycles every time it gets accessed. For math, doing operations like comparisons, addition, subtraction, and bit logic takes 2 clock cycles, while multiplication takes 10 clock cycles.

**Results**

To test the cache, there were three tests to be done with both the cache and the Main Memory. These tests were to create an array of 20 integers and then take their sum by adding up their values and iterating both forward and backward through the array. The result of this was that using the cache took 8,168 clock cycles. On the other hand, when this test was performed with the Main Memory only, it took 80,768 clock cycles. Obviously using the cache was much faster than using the Main Memory in this case. The story is surprisingly the same when creating the same array and then iterating over it backwards. This is because the information was still stored in the cache the same way, however, it is interesting that the results were the same. The results were that using the cache took 8,168 clock cycles while using the Main Memory took 80,768 clock cycles so the same data as with adding the array forwards. The next test was to create a linked list in the memory and add this up. Creating the linked list was done by putting the integer value into an address in memory and then storing the address of the next integer (in this case, 20 words in memory ahead) in the next address in memory. Using the cache to add the integers in the linked list took 27,198 clock cycles while using the Main Memory took 111,188 clock cycles. Clearly for this test, the Main Memory was also optimal.

**Conclusion**

The conclusion of these tests is that using the cache reduces the number of clock cycles per test significantly. For the first two tests, using the cache reduced the number of clock cycles by almost 90%. Additionally, for the last test, using the cache reduced the number of clock cycles by just over 75.5%. Both massive factors proving that using a cache is extremely useful in computer architecture.