Nathan Beauchamp, Rishi Thakkar, Melissa Jin
Team name: Silver Pipelining
ECE 411 MP3, Checkpoint 4

Progress Report:

For Checkpoint 4, our group implemented three core features: an eviction write buffer, static branch prediction, and performance counters. First, the eviction write buffer is an optimization meant to improve the performance of our write-back caches. The eviction write buffer is meant to hold data being written back to the next memory hierarchy level (e.g. physical memory for the L2 cache). This enables the cache to respond to read requests while the write is taking place. This is very useful as the cache can potentially service multiple read hits while writing back a dirty line. Next, for branch prediction, we decided to implement a static not-taken approach. Essentially, when the processor encounters a branch instruction, the following instructions are loaded into the pipeline as normal (i.e. we're predicting not taken). Whether or not the branch is actually taken is determined in the writeback stage- if the branch is indeed not taken, then pipeline operation continues. Else, all instructions currently in the pipeline are flushed, and the next instruction is loaded from the calculated PC address. Unfortunately, the pipeline still has to stall for one cycle to account for the case of an STR instruction appearing immediately after a BR. Finally, we implemented performance counters to track the following metrics: number of cache hits and misses (for all three caches), number of correct and incorrect branch predictions, and number of pipeline stall cycles. We built a memory-mapped I/O structure so that the performance counters are accessible from software. A programmer can read the registers by reading from memory addresses FFE0-FFF0; they can clear the registers by writing to the same addresses.

For this checkpoint, work was divided in an interesting but effective way. Nathan implemented static branch prediction and performance counters while Rishi and Melissa implemented the write eviction buffer. However, Rishi and Melissa also got a head start on advanced features for the final design, implementing a victim cache and a basic version of hardware prefetching. They also handled the testing and debugging of such features.