Checkpoint 4

Checkpoint 4 Progress Report

Status:

Bugs with our forwarding code continue to plague that aspect of our processor, but the functionality for checkpoint 4 is completed. Our performance counters work to count our stalls, branches, cache hits/misses, and other aspects of our system. Additionally, these counters can be reset by writing to the specific memory-mapped I/O address corresponding to each counter.

Our eviction write buffer also functions mostly as intended, for what we can tell right now. We have written test code to verify this.

Our static branch prediction is static not-taken, which works in tandem with branch flushing to successfully execute all branch instructions.This feature is completed as well.

Testing:

We wrote test code to test all aspects of our processor, as for this checkpoint, test code is not provided. For checking the performance counters, we first ran a lengthy program, and then loaded R1 and R2 continuously with the various counters, and then zeroed them out, and then fetched the counter values again. This allowed us to check functionality of both the counters and the reset signals for the memory mapped I/O.

We wrote test code to check our branch flushing and static branch prediction as well, by putting instructions that should not be allowed to the MEM stage right after taken branches, to see if those instructions were executed. They were not, and were instead flushed as expected.

We finally wrote test code to check our eviction write buffer by writing test code that evicts from our L2, and then checking to see if the data was correctly written back to memory by loading the evicted data back into registers. It was verified to be working by us.

Work Split:
Rohan added performance counters and support for the eviction write buffer
Tanishq and Tadas continued to work on the forwarding bug

Checkpoint 5 Roadmap

VIctim Cache - Rohan
Tanishq and Tadas - Memory leapfrogging