MP4.2 Report

Progress Report

For checkpoint 2, we distributed the checkpoint objectives as follows…

Neo, Anchit - Forwarding / Hazards

Implemented Functionalities - Implemented basic forwarding through the stages of the pipeline; Forward and branch stalling for branch commands and changing instructions.

Testing Strategies - Tested against provided checkpoint 2 code, factorial.s, and other test programs created during MP2. Modified given code to test back-to-back branching as well as branching on multiple successful branches. This portion of code was tested with magic memory before progression onto the arbiter.

Naveen - Arbiter

Implemented Functionalities - Implemented basic arbiter functionalities including switching to ICACHE or DCACHE depending on input changes.

Testing Strategies - Since forwarding and hazards were tested extensively before work was done on the arbiter, we were able to narrow down any bugs during arbiter development to cache-related problems. These bugs were narrowed down using the provided checkpoint 2 tests.

Roadmap - Advanced Features

For the next checkpoint, we have decided to split up the following tasks among us three.

Perceptron Branch Prediction (8)

Assignment: Neo
Explanation: Branch prediction to preload instructions and operations based on patterns in current program instructions.
Performance Counter: Amount of correctly predicted branches / total branches
More Information: https://www.cs.utexas.edu/users/lin/papers/hpca01.ps

4-Way Set Associative BTB (3)

Assignment: Neo
Explanation: BTB is needed in order to implement the branch predictor. The 4-way set associative BTB is a faster version to optimize our operations.
Performance Counter: Runtime comparison against non-BTB code

8-Way Set Associative L2 Cache (5)

Assignment: Naveen
Explanation: Faster cache shared between ICACHE and DCACHE to improve storage speeds and hit accuracy.
Performance Counter: Hit-to-eviction ratio

BRAM Arrays (2)

Assignment: Anchit
Explanation: Improvement on f_max by improving speed of arrays.
Performance Counter: Runtime comparison against non-BRAM array code

Explanation of Inclusive/Exclusive (1)

Assignment: Anchit
Explanation: Explanation of design choices between inclusive L2 cache vs exclusive L2 cache.

Hardware Prefetching (4)

Assignment: Anchit
Explanation: Using hardware optimizations to speed up prefetching of instructions.
Performance Counter: Amount of correctly predicted instructions / total instructions & comparison to ensure that there is a positive improvement in performance.