11/09/22

Progress

The pipeline cpu has been integrated with the given direct mapped cache. Our arbiter correctly prioritizes the d-cache in the case where both d-cache and i-cache requests data from memory. The stalling unit also works by properly stalling the fetch state in the case of an i-cache miss and stalling fetch, decode, execute, and memory access in the case of a d-cache miss. Our static not-taken branch predictor correctly detects mispredicts and sets the previous 2 instructions to nops. The forwarding unit allows for WB->EXE, MEM->EXE, and WB->MEM data forwarding. We have also changed our reg file to be transparent.

Distribution of Work

Jing: Stalling unit and branch predictor
Albert: Forwarding unit
Justin: Arbiter

Testing Strategy

After integrating the i-cache and d-cache, we ran the given test code for cp1 to ensure that it still produces the correct value.

Afterwards, we used a fully working mp2 cpu to run cp2 test codes and compared the data in the reg file. This allowed us to identify bugs in our forwarding unit and stalling unit.

Timing and Area

The timing report gives a slack (MET) of 4.62, indicating that the design is synthesizable. The area report gives a combinational area of 27869 and non-combinational area of 24012.

Roadmap

There are still issues with changing our mp3 cache to a single-cycle hit so we will continue to debug it. Alternatively, we can leave it as a 2 cycle hit and change it to be pipelined. Furthermore, there are bugs when running our cpu with the competition test files so that will have to be addressed.

Advanced features we currently plan to implement are:

- Convert our cache to 4-way set associative, which requires changing the LRU to 3 bits instead of 1 bit.
- Convert our cache to be pipelined
- Local branch history predictor
- Basic hardware prefetching

If time allows for it, we will attempt to implement the RISC-V M Extension.