ECE411 Progress

PCB

Prateek Tenkale, Chris Shannon, Brendan Callas

Checkpoint 3

Who Worked on What

Chris: 8-way L2 Cache with bit-pLRU and Eviction Write Buffer

Brendan: M Extension

Prateek: OBL Prefetching

Functionalities

For this checkpoint, we implemented advanced features for our RISC-V pipelined processor. As mentioned above, we implemented an 8-way set associative L2 cache shared by the L1 Data and Instruction Caches. The L2 cache is a 2-cycle cache so that BRAM can be used for competition code and to limit the performance impact for the Fmax of our processor. A bit-pLRU algorithm was used for cache evictions due to its efficiency and scalability for the cache. An eviction write buffer was also implemented to hide latency when cache misses occurred, and slightly increases the cache hit rate for recently-evicted cache lines. We also added the RISC-V M-extension to support multiplication and division operations using a shift-add multiplier and similar divider to speed up multiplication operations. Finally, we implemented an OLB prefetcher for the instruction cache to fetch the next cache line on a cache miss without stalling the cache's hits.

Testing Strategy

Like previous checkpoints, we used ModelSim to test our features and for debugging purposes. We can compare the code running on MP3 to the pipelined processor to ensure proper register states, and analyze the timing of each program to determine how effective each advanced feature is at improving performance. For each advanced feature, we used separate branches in Github to ensure no conflicts arise and that work can be saved and tested without impacting others' work. We also wrote our own assembly code to specifically test each advanced feature as it was implemented to ensure full functionality and correctness, useful in unit testing specifics, edge cases, and for debugging purposes.

Features for Next Checkpoint

Next checkpoint is the competition code, so we plan to make a few improvements to our processor to increase performance and lower power. For the L2 cache we will implement BRAM

to significantly reduce the power and latency cost of the L2 cache. If it remains stable, we can also scale the L2 cache to be 16-way to further reduce costly memory operations and limit misses as much as possible. We will also look for ways to reduce our critical path in our design to increase Fmax. For OBL Prefetching, we will evaluate Fmax and compare overall execution times with and without this feature to determine whether it is worth implementing

Work Plan For Next Checkpoint

Chris: Implementing BRAM for L2 cache, testing the ability to increase the L2 cache to be 16-way set associative.

Brendan and Prateek: Reduce critical paths as much as possible and test CPU with and without advanced features

Performance Metrics

L2 Cache

- Mp4-cp3
  - Cache hits: 990
  - Cache Misses: 1334
  - Writebacks: 640
- Comp1
  - Cache hits: 1
  - Cache Misses: 35
  - Writebacks: 0
- Comp2_i
  - Cache hits: 4772
  - Cache Misses: 1131
  - Writebacks: 15
- Comp3
  - Cache hits: 5341
  - Cache Misses: 1841
  - Writebacks: 1

Eviction Write Buffer

- Mp4-cp3
  - Hits: 24
  - Misses: 2300
  - Writebacks: 640
  - Cycles Saved: 30680
- Comp1
  - Hits: 0
  - Misses: 36

Writebacks: 0
Cycles Saved: 0

Comp2_i

Hits: 0
Misses: 5903
Writebacks: 15
Cycles Saved: 705

Comp3

Hits: 0
Misses: 7182
Writebacks: 1
Cycles Saved: 57

OBL Prefetching

Mp4-cp3

Cycles wasted prefetching: 20371
Instruction hits while prefetching: 155502
Instruction misses: 719

Comp1

Cycles wasted prefetching: 2040
Instruction hits while prefetching: 53386
Instruction misses: 266

Comp2_i

Cycles wasted prefetching: 4657
Instruction hits while prefetching: 127565
Instruction misses: 2877

Comp3

Cycles wasted prefetching: 28368
Instruction hits while prefetching: 69622
Instruction misses: 1012