CP3 Advanced Features Design

iCPU - Prannav Gupta, Hemank Kohli, Dom Quigley

Advanced Features

- Eviction Write Buffer - The purpose of the eviction write buffer is to avoid the delay incurred by writing back dirty cachelines to memory. We can do this by adding an extra buffer to our cache datapath and alter cache control to send cachelines to the buffer rather than waiting on memory if the dirty bit is set. We also need to keep track of the memory address associated with each block in the buffer. Clean blocks do not need to be written back with this implementation and can simply be replaced. We think we should be able to add control logic as well to allow data in the buffer to be written to memory in parallel while newly fetched data is being sent to the CPU.

- Pipelined L1 Cache - Implementing a small pipeline in our cache will require registers to separate the two stages from each other so that the second stage isn't using the memory address/data from the second request, which should be isolated to the first stage.

- L2 Cache - Adding another cache level is relatively straightforward as it does not impact the implementation of L1 cache very much. We can implement a second intermediate cache between our current cache and main memory to decrease the amount of memory reads/writes.

- Basic Hardware Prefetching - We think the most efficient way to implement prefetching would be to first bring in the desired cacheline and send it back to the processor, then in a second take fetch the block that follows. This makes fetching from memory faster and also saves time in the case where the block to be prefetched is already in cache.

- RISCV-M Extension - The implementation of the M instructions will require us to add proper control assignment in the Decode stage, adding any stall logic, and then to add the logic for the actual multiplication/division operations into our ALU.

- Local Branch History Table - This can be implemented using a simple table, like the data arrays, indexed by 32 bit memory addresses and associated to a 2 bit number that represents how frequently the branch has been taken (11 being high probability). This can then be used to determine the next address for PC to fetch instruction from for prefetching, and the 2 bit number can be incremented/decremented after the correct code executes based on the correctness of the prediction.