Cp2 4/8/2020

**Filip Cakulev:** This checkpoint I worked on making my new cache design meet timing and work for a single cycle. This included changing the given array and data_array to work with always_comb and be able to have sme-cycle reads. I also worked on making a working arbiter control based on Robbie's  that was different from the initial design that would work based on data and instruction requests individually. I also attempted to pipeline the cache to no avail. While I believe it was close we decided to hold back on that part of the implementation for this checkpoint. For the next checkpoint I plan on fixing the pipelined aspect and implementing the L2 cache.

**Robbie Ernst:** Instantiated new dut in top test bench. Hooked up shadow memory and parameter memory. Spent majority of time debugging the pipeline cache hookup to the CPU. Eventually had to remove some of the pipeline ability of the cache in order for functionality for this checkpoint. Wrote a control for the arbiter that was eventually adapted by Filip to work for the pipelined cache. Debugged majority of the shadow memory errors. Next week, will attempt to be the first to try and implement forwarding with assistance of Ricky. As well as reimplementing the cache pipeline with the help of Filip.

**Ricky Machado:** For this week, I added the remaining instructions to the control and added two necessary aligner modules in order for loading and storing 4 byte, 2 byte, and byte data transfers. As well as that, I helped robbie debug the processor control to see that our arbiter couldn't handle a miss in our icache and dcache at the same time, which ultimately got resolved. On top of that, I also came up with the paper design for our forwarding unit so we could effectively implement that for next week. I also turned in this paper if that counts.