## Progress Report (CP2):

In this checkpoint, we implemented our data forwarding as well as handling control hazards. We also implemented the arbiter that connects to our caches, which replaces the use of magic memory and now uses the cache system. We all worked on the forwarding, stalling, control hazards, and arbiter. Aayush and Yash focused on working on the data forwarding and stalling, while Vishal focused on working on the Arbiter and connecting it in our datapath. However, after the basic code was written for both of these parts, debugging the errors and the warnings that we got from Quartus/Modelsim was done together. While trying to verify our design with our datapath, we noticed that our conditions of forwarding were not the most accurate, and so after modifying this, we noticed that we were off by one cycle. Another error that we saw was when we had a load instruction, we were actually forwarding the address from the ALU output, instead of actually forwarding the data from that unit. In order to fix this, we added a condition that if the instruction is a load instruction, we forward the rdata instead of the address. We also connected the RVFI monitor and Shadow Memory, and noticed an error of being off by one cycle. We went to office hours to try to fix this error, and a couple days later they released the fix which fixed our warnings. To test our code, we ran the given test program for CP2 and looked at the values in each register and signals for each instruction to make sure they were grabbing the correct values. We also ran the CP1 checkpoint code to make sure that our cache was able to respond correctly and still run the instructions. In terms of the design for CP3, we all worked on diagrams.net again, with each of us focusing on one of the features. Yash worked on the tournament branch predictor diagrams and the branch target buffer, Vishal worked on the Basic Hardware Prefetching and Multiplier Description, and Aayush worked on the Eviction Write Buffer and Cache Diagrams. We all verified each other's diagrams and made some minor adjustments as needed.

## Roadmap for CP3:

For this checkpoint, we will need to implement our advanced features. As of now, we are planning on implementing a tournament branch predictor using a local branch history table and a global branch history table, a branch target buffer, an L2 cache as well as a 4-way set associative cache, and the RISC-V M Extension. We also plan on parameterizing the cache to some extent. This is more than 20 points as of now, but we were going to measure the feasibility of these advanced features and adjust accordingly. We will all take the lead of the parts that we made the diagram for, but as usual we will use Microsoft Visual Studio Code Live Share in order to all work on the parts of code and to get an understanding of how everything is connected together. We will have to verify these parts of the design one at a time and measure if they make a valid decrease in our timing and efficiency. We will proceed with our typical verification process of monitoring particular signals and making sure they have the correct values at a given time. We will test the design without the advanced feature and then with that feature added so that we can compare the improvement in performance.