Yash Anand’s Journal

Milestone 5

10/28/23
Converted the control state diagram to table form and added a new control signal called JumpOut to select the value we would add to PC if Branch is 1.

10/31/23
Finished implementing the control and started writing the test benches. When the opcode is 0 or 1, the func can be 0, 1, 2, or 3, so the ALUop will change for each of those values. Each control signal needs to be updated for every opcode.

11/1/23
Finished test benches for control and fixed errors in control. The tests run through every func when opcode is 0 or 1, then stop checking func after opcode = 2 since ALUop should be fixed regardless of func. Both branches have the same control signals so they are grouped in the control implementation but separate in the tests to make sure they both work.

Milestone 4

10/23/23
Started the state diagram and implementing the verilog for the control. The I-type and R-type control are similar but will split for the execute and rejoin for the writeback.

10/24/23
Completed the state diagram for control. The branch and jump type currently have the same states, so we need some way of knowing which one we are doing. One idea we may implement to the datapath is an ‘or’ gate after the ‘and’ gate for the branch and have another control signal to ‘or’ with.

Milestone 3

10/18/23
Completed milestone 3. I did the control signal descriptions and updated the components list. For implementing pipelining, we have added all the necessary register blocks. We have also decided to not do BDS, but branch prediction instead because it

will likely be faster for the programs we need to execute. Thus, we have tentatively added a Branch Prediction & Hazard Unit that we will implement in M4. Next milestone, I will do all the control input stuff.

Milestone 2

10/10/23
Completed instructions for I-type RTL, making progress on milestone 2. Should be done with milestone 2 by tomorrow. We should add more stuff to distance from just RISC-V soon, maybe by milestone 3 meaning we will have to meet and discuss sometime this week.

10/11/23
Finished milestone 2. We are adding pipelining to add more uniqueness to our project. Next week, I will handle the control signal descriptions and contribute to the datapath design for milestone 3.

Milestone 1

9/27/23
We met in class to discuss which implementation we wanted to do. We have decided on load/store but might incorporate an accumulator later on to gain more optimization. We decided to have registers specific to the problems our machine will have to solve: Euclid’s algorithm and basic code operations.

10/4/23
Meeting with Robert, we need to add something more to distinguish it from just RISC-V. Ideas include looking at what we don’t like about RISC-V such as order, procedure calls, and understandability.