3. Add a PDF file REPORT.pdf about your pipeline. They must include at least the following:

a. Describe your design in terms of organization and operation.

We designed a superscalar two-way 5 stage pipelined design with forwarding enabled as well as stalling. We changed our branch prediction, forwarding, and stalling logic to work for a superscalar two-way implementation. We used a two bit hysteresis counter for our branch history tables. Majority if not all of the registers were doubled for the superscalar two-way implementation. We also added a second decoder, alu, and multiplier.

b. Describe the critical path of your synthesized implementation. (Is it different than what

you expected?) How does the clock frequency compare to your Lab 3 implementation?

Our critical path happens when we forward data back to the alu which is then used for bcond, branch prediction, and next pc logic.

d. Any special tricks you played that may not be obvious to the course staff?

To check for no\_op instructions without causing a weird compiler error, we checked whenever the pc value of the instruction was less than 1 as opposed to checking if it was 0 directly.

e. Estimate the number of hours spent on: 49 hours

i. planning the design in your mind or on paper: 2 hour

ii. capturing the design (that is, actually hacking in Verilog): 15 hours

iii. testing the design (not including debug time): 10 hours

iv. debugging the design (actually fixing the bugs once found) 20 hours

v. analyzing the performance of the design (next item) 2 hours