Final Report

What did I learn from this project?
Just because you can simulate something in Modelsim, doesn't mean that you can synthesize it. This mistake could be attributed to my bad VHDL coding habits. For example, I frequently use the “process” keyword to code asynchronous elements. So when I coded the asynchronous handshakes, the fact that I used a process statement didn't tip me off to the fact that I may have been storing values in a latch or register. It was only until later in the semester that I talked to Dr. Agrawal that I realized that there may be memory elements in my circuit, and started to worry about how they would be synthesized.

Another thing that I learnt was the difficulty in keeping within a set of constraints when designing something. Many of the constraints seemed arbitrary, but I suppose that is how it is in the real world. When a customer wants something, he makes the demands and it is up to the engineers to fulfill them. There may be some wiggle room for negotiation but in the end, there are some things that the customer might want and will not compromise on.

I did learn a lot about asynchronous circuits which was tough since there is not a lot of information available online.

What would you do differently next time?
Scale down the ambition. I grossly overestimated the amount of free time that I had. What I based my CPU off was the PicoBlaze, a fully functional soft core CPU made by Xilinx. I then decided to make it asynchronous as well, because why not?

What advice would I give someone who is working a similar project?
Take it easy and slow, no need to be fancy. Also a multicyle CPU seems to be the easiest and has the greatest success rate despite the fact that it's control unit is more complex. And start with the model for the complete CPU first and keep synthesizing it every step of the way.