Executive Summary – Spencer Cheney

Contributions

Libero

- Removed LVDS UART modules that were not working
- Configured UART module for Full-Duplex
- Added LVDS module
- Configured Clock Generation module for LVDS

SoftConsole

- Developed UART communication test
- Set up UI for UART test
- Documented UART code

Testing

- Programed Full-Duplex UART communication on Raspberry Pi to test the UART communication on the FPGA
- Edited SPI and I2C test to work with usable values

Documentation

- Executive Summary
- Final Report
- Presentation
- Poster
- Video showing how to turn off Sophos Anti-virus software, begin synthesizing Libero, and communicate with the FPGA through Putty
- Packed up Project Materials and Testing Equipment

Future Suggestions

The hardest part about this project was communication (I'm not talking about UART even though that's the majority of what I did). A big part of this is probably due to COVID-19 but I believe we could have gotten a lot done if there was a lot more communication in the beginning.

The first issue we had with communication was with the BYUI I.T. department, they were all friendly and easy to work with, but it took a long time to get everything set up and get the I.T. staff at the help desk on the same page (I got into an argument with I.T. staff this last week (week 13) about whether the computer was from the STC or if it was I.T.'s and whether or not we could use it). To avoid this next time I would make sure that the supervisor in charge of the computer knew exactly what we needed and we got all the admin access we need right away because every couple days our admin access would be removed because something wasn't set up right and make sure we can communicate with him to understand what we need to tell the Help Desk staff to get set up with want we need.

Next thing we could have done better is communication within the team. A big issue with this project is the fact that only one computer can run Libero and the synthesis project takes forever (almost 2 hours). It would have been a lot easier to have everyone on the team making changes to the Libero project and get all the changes done at one time, so you only synthesize about once a week. With Libero and SoftConsole being new software to everyone there was a huge learning curve. I think it would've been much more beneficially to run through the Libero synthesis project and how to set up SoftConsole code together at the very beginning of the semester so everyone on the team knew how to work on project right away.

A couple other things I would recommend outside of communication would be to install Libero on a personal computer as opposed to a school's computer. This would solve all the issues with I.T., Sophos, and we would have almost constant access to it. Plus, the school's computer does not run Libero well. There were many time that we ran into errors synthesizing Libero because there wasn't enough RAM installed on the computer, for at least the "Generate FPGA Array Data" step you can't have Chrome or any other intense programs running during it. Also, it would have been a lot more helpful to have everyone here in Rexburg for the project. Don't get me wrong Kyle helped out plenty and did what he could but it was hard enough having to meet in the MC during the specified times to work on the project but it was hard to find time to test Kyle's code to find out what it was suppose to do wait to hear back from him then make specific changes to it.

As you can see from my table it took 5 – 6 weeks to even turn on the FPGA with the basic program from the tutorial. An additional 3 - 4 weeks (about week 10) to figure out how SoftConsole and Libero work together to program the FPGA. I honestly feel like I have done all my work for the project in the last 3 weeks and that wasn't enough time to get everything done I wanted to and I was hesitant to understand what aspects the other members of the team were working on and to help them with the. Half of this project is teaching you how to program the FPGA so without immediate access to the FPGA and Libero as well as plenty of documentation and training on how to operate all of the software I don't think you could complete a significant amount of this project (over 33.3%) in a single semester.

Timetable

Week	Time	Work Done
Week 1	12	• Completed Leadership Material
Week 2	10.5	• Learned VHDL • Review UART communication • Got familiar with SoftConsole and Libero • Started developing UART communication for FPGA
Week 3	10	• Reviewed UART, I2C, and SPI communication • Learned about Bus Protocols for other processor set ups on FPGA • Review SoftConsole code from previous semester
Week 4	13	• Set up SoftConsole Code for UART test • Got set up (including admin access) on computer in the MC
Week 5 – 6	23	• Work on UART SoftConsole test • Was able to Communicate with FPGA through Putty • Learned and worked through Libero
Week 7	13	• Got UART SoftConsole code ready for testing

		• Loaded the current working solution with SoftConsole Code to FPGA • Researched Pin Mapping

Week 8	13	• Debugged UART test
		  ○ Reviewed pin map documentation
		  ○ Review Libero Code

Week 9	11	• Finished UART test • Review Libero UART modules
Week 10	23	• Rewrote UART test to transmit and receive properly • Stripped LVDS UART module to remove differential signal functionality to ensure we can transmit regular UART correctly • Increased Baud rate to allow for future LVDS functionality • Researched device to test LVDS UART • Wrote UART test for Raspberry Pi to test FPGA UART functionality at the high baud rate • Tested and Debugged FPGA UART communication with Raspberry Pi
Week 11	9.5	• Tested Full-Duplex UART • Generated Differential Signal
Week 12	15.5	• Tried to set up LVDS signal so high was around 1.25 v and low is 0.75 v • Tried testing SPI and I2C
Week 13	12	• Final Report • Poster • Presentation
Total	**165.5/161**	