### Lab 1

Name: Dai’zjon Carney TA Signature:

Signature: TA Date:

**GOALS**

Lab 1 was used as a gentle introduction into Vivado. The first goal of the lab was to go through programming the FPGA four different ways. These four ways were using an existing BIT File, using vdhl code for the Mux, using block diagram design with a complete ip core, and using block diagram design with individual gates. The last goal was to learn the Vivado simulator.

**Summary**

The easiest part of the lab came from using the existing bit file. Since the file was already made, there was no real work required besides just simply programming the file to the BASYS3 board. Things began to pick up when it came time to program using VHDL. In this stage a VHDL assignment statement was used to get the MUX wanted. This stage is also where I first learned to see schematics and define constraints, which would need to be replicated again later.

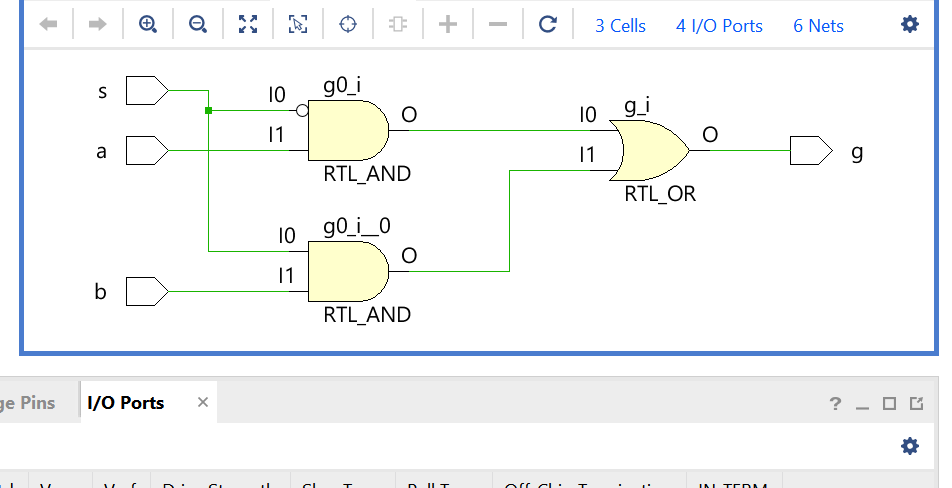
The last couple things to do were to run synthesis and implementation before creating the BIT file. Once the file was created, I could once again begin programming. For both the block design parts, IP cores were introduced. There was one IP core that was a complete MUX as well as smaller IP cores that used gates to come together and create a MUX. Both work the same conceptually and both needed Vivado to translate the block diagrams to VHDL. Afterwards the same steps of viewing schematics and defining constraints were used as well as creating the BIT file and programming.

The simulator part was relatively easy as well. I just had to run a test bench file as a simulation source. This granted the ability to see the simulation waveforms for the project which can help show if the function is correct. Before this lab it was safe to say I had no idea on how to use Vivado. However, now I know how to set up Vivado, program The BASYS3, run VHDL code, run Block design, and run simulations.

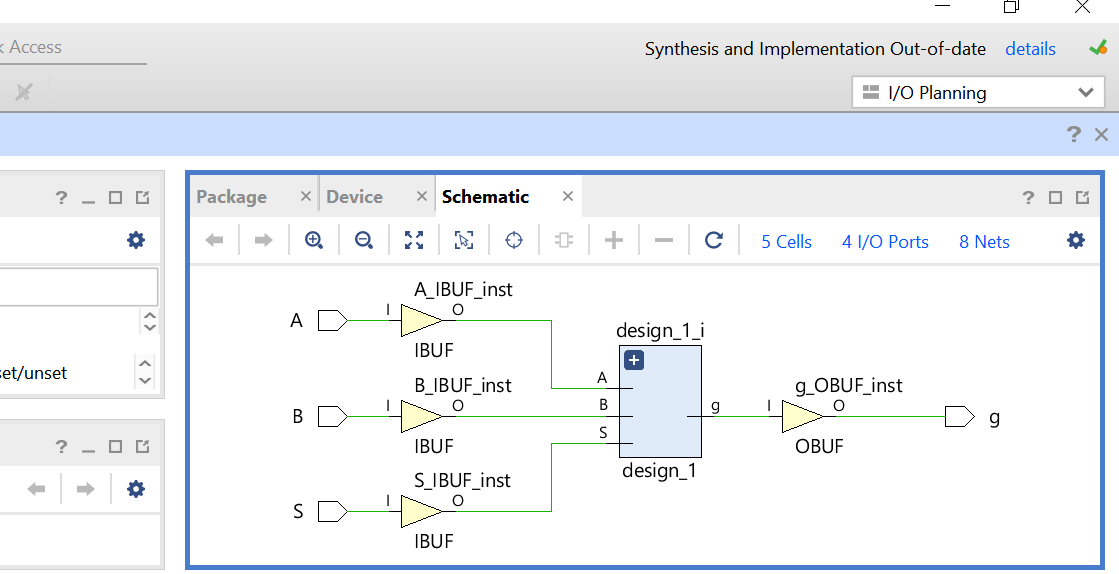
**Appendix**

*Schematics:*

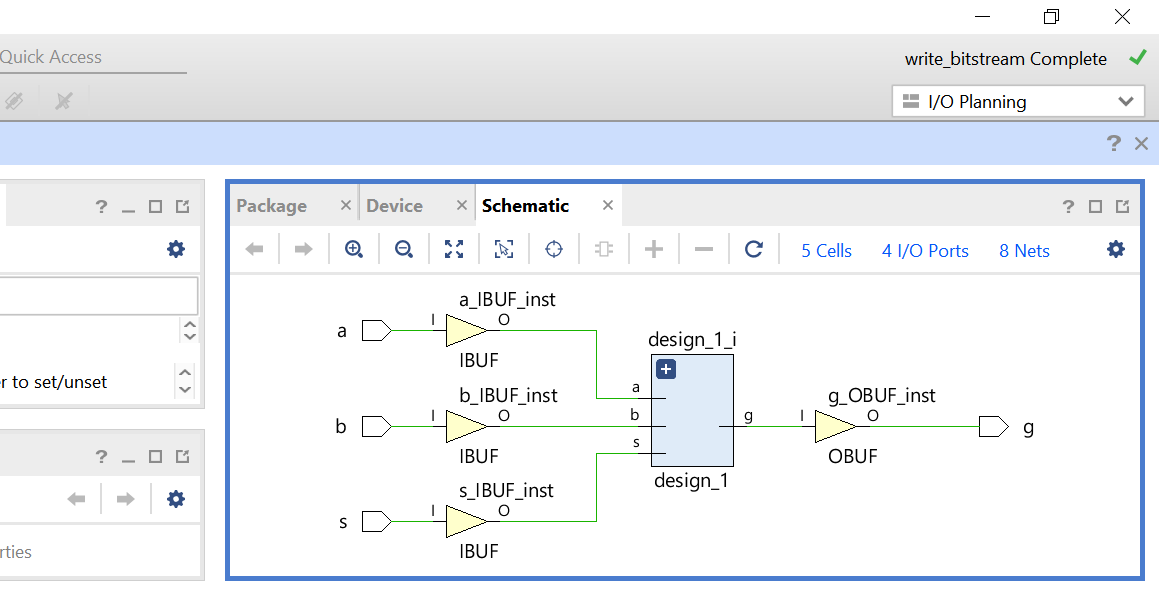
Part 2 (vhdl MUX)



Part 3 (2\_to\_1 MUX)



Part 4 (individual gates MUX)



*Simulation:*

