Lab 1 Report

In this lab, I Installed modelsim. After Installed modelsim, I worked on implementing a 1-bit full adder on the Xilinx dev board. I used the provided Verilog code for a multi-bit adder and a Testbench for it.I then simulated a 16-bit Adder. As seen in the waveform below, I had objects being Cin, X, Y, Sum, and Cout. The variables here show the carry out process. One particular point shows one-bit addition with carry from the previous stage. A point on a waveform would be some arbitrary t. The output correspond to the addition of the variables. In specific, 1+0 = 1, but the addition here also has 1+1, which will result in a carry. Ultimately, leaving a carry out.

To start off, I walked through model sim. I first added a source file and then added a test bench file. I saved my 2 files and compiled all my files. Afterwards, I simulated my files. I ensured that the objects present were in line with the expected objects. I added all my objects to the wave pane. Before running, I changed the run time to 1us. After I ran my code, I observed my simulation wave.

I then used vivado in order to demonstrate how design code can be used to create working hardware. I started off by creating a new project. I selected to create an RTL project. I then created a source file with Verilog as the target language. I then created a constraint file named nexys4_ddr_constraints. Under the default part, I selected Nexys4 DDR. I then used the provided code for the source file and constraint file to add it in the editor window for each respective file. I found the constraint file code in github. However, I made some edits by removing the '#' symbol infront of selected comments so they can be used as code. I also edited the section for clock signals and file outputs. I programmed the XILINX chip from that point on. To do so, I generated bitstream. I then connected my FPGA board to my pc. I selected "open target" to correctly connect my FPGA board. As a result, I saw LED lights turn on/off.