Class:	CPE300L	Semester:	Fall 2021

Points		Document author:	Jerrod Batu
		Author's email:	batuj1@unlv.nevada.edu
		Document topic:	Postlab 1

Instructor's comments:

1. Introduction / Theory of Operation

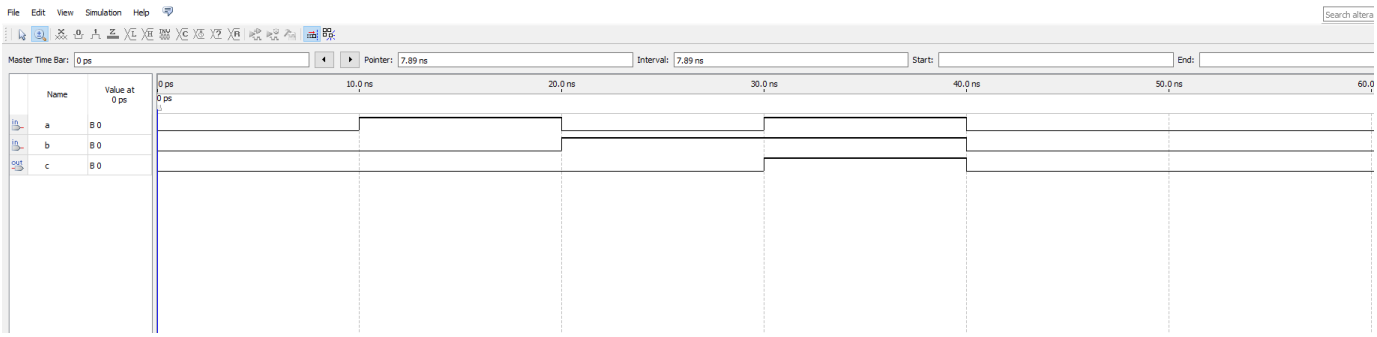
Hardware Description Language (HDL) is a computer language that describes the construction, implementation, and construction of digital electronic circuits. Some common HDL include Verilog and VHDL. Specifically, Verilog is based on C language where VHDL (Verilog Hardware Description Language) is an older programming language composed of Ada and Pascal languages.

2. Prelab

No prelab

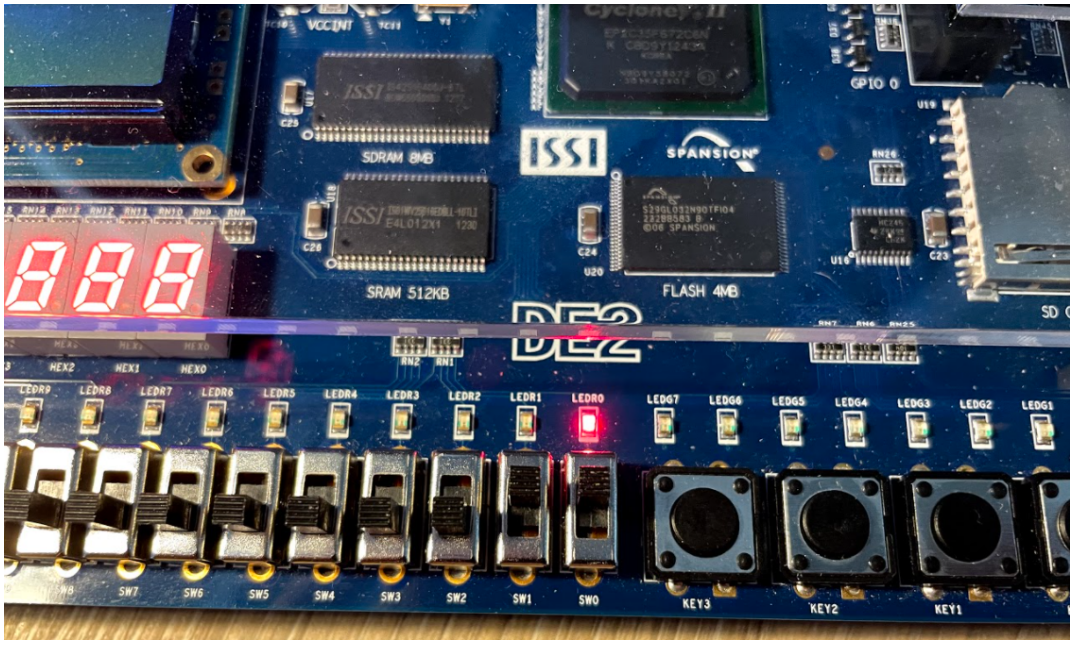
3. Results of Experiments

Experiment 1

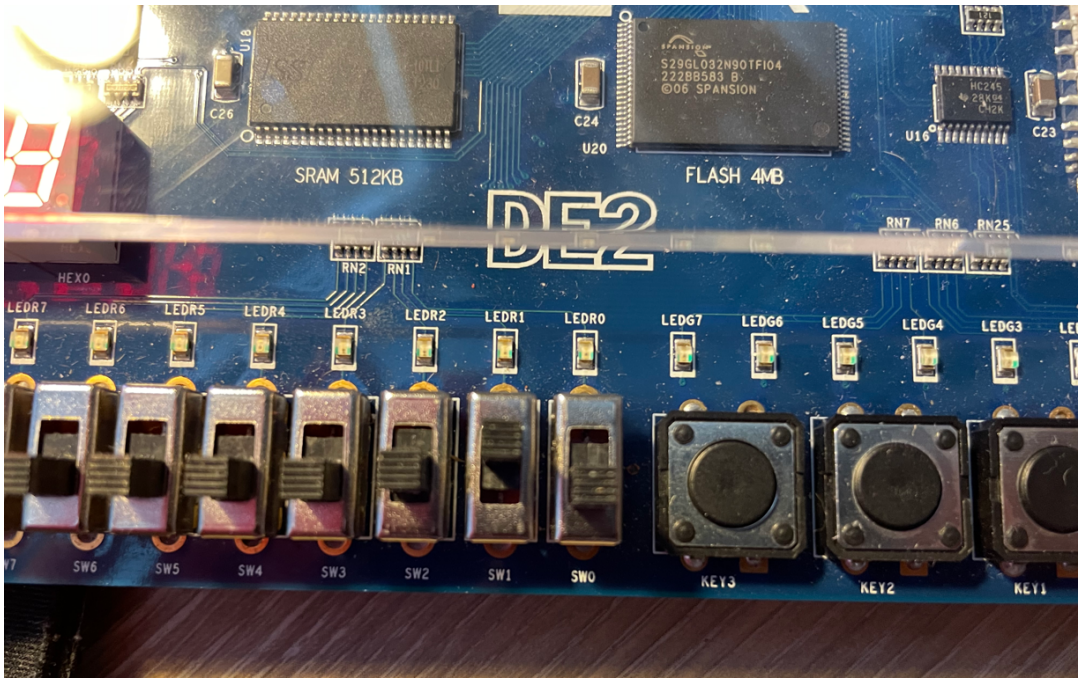


This is a picture of my AND gate waveforms indicating the combinational logic where the output would be 1 or 0.

Experiment 2

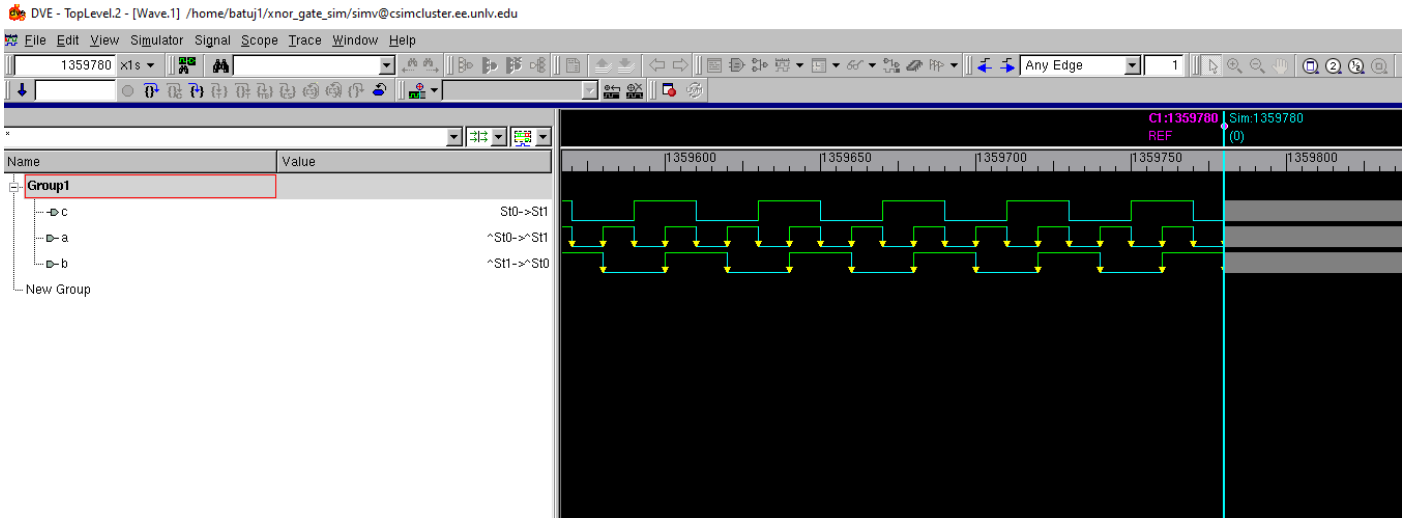


This is a picture of a DE2 board following an AND gate when the inputs are 1 and 1 leading to an output of 1 (on).

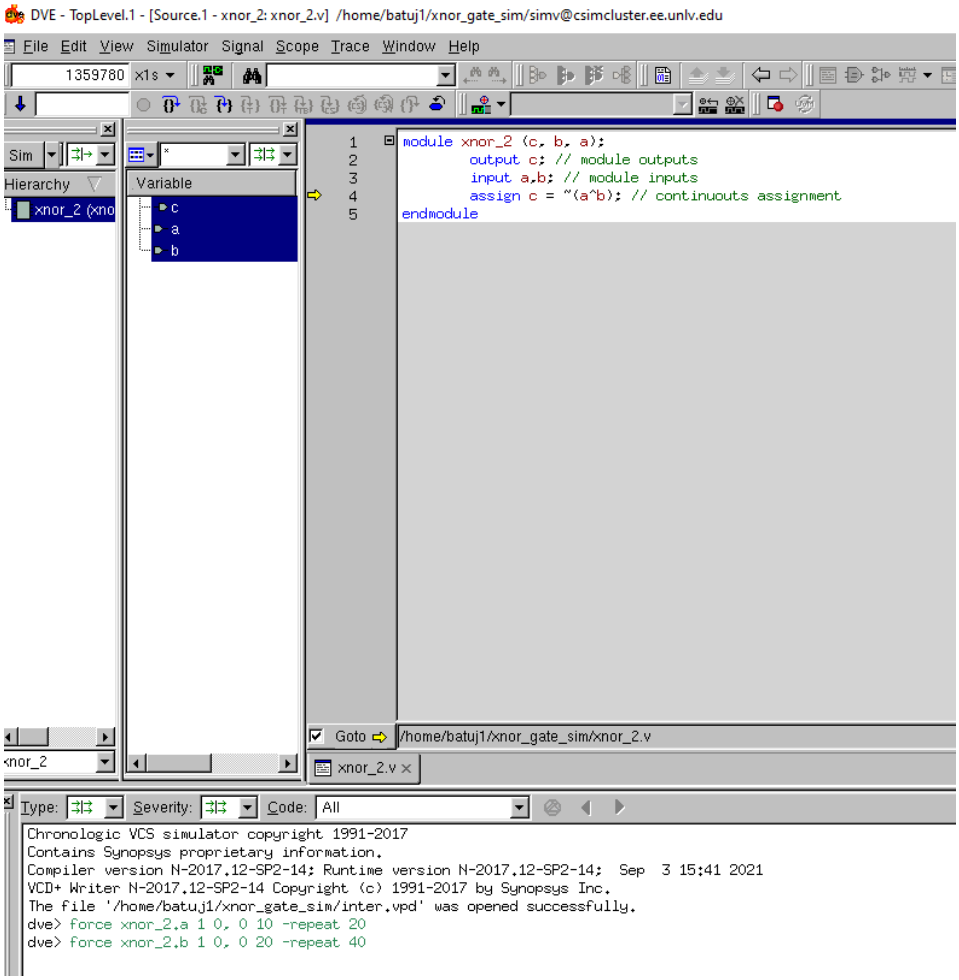


This is a picture of a DE2 board following an AND gate when the inputs are 1 and 0 leading to an output of 0 (off).

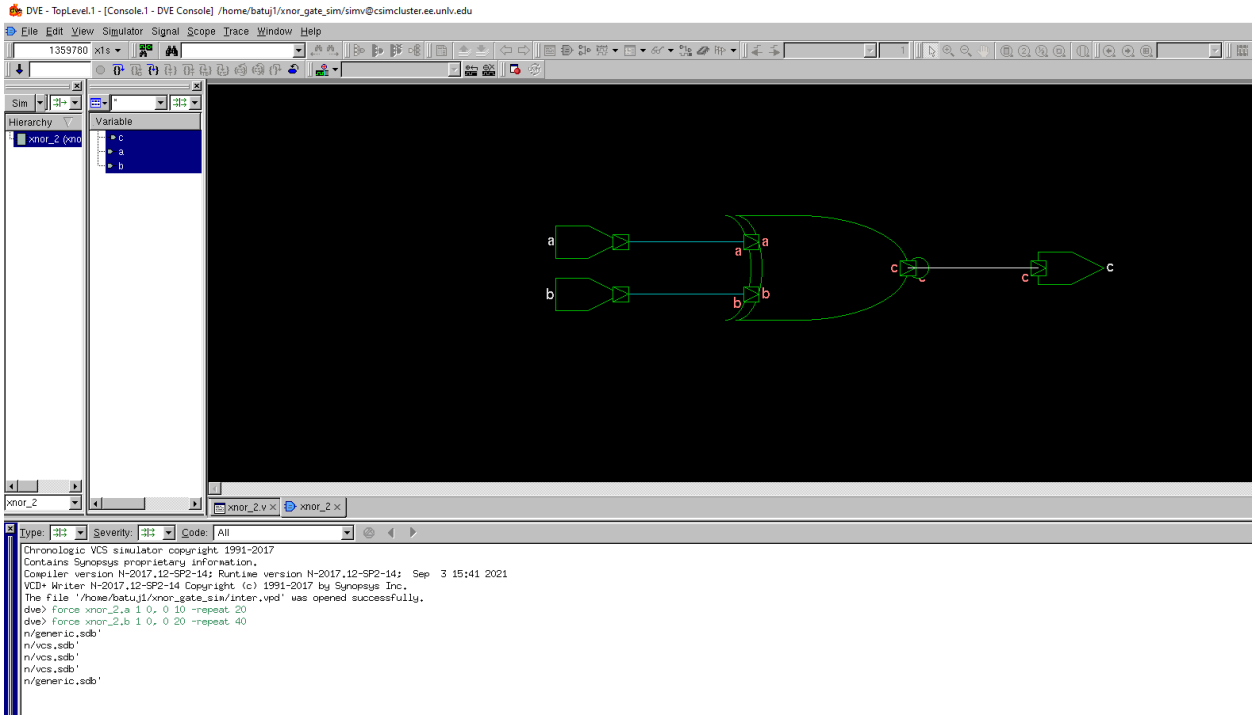
Experiment 3



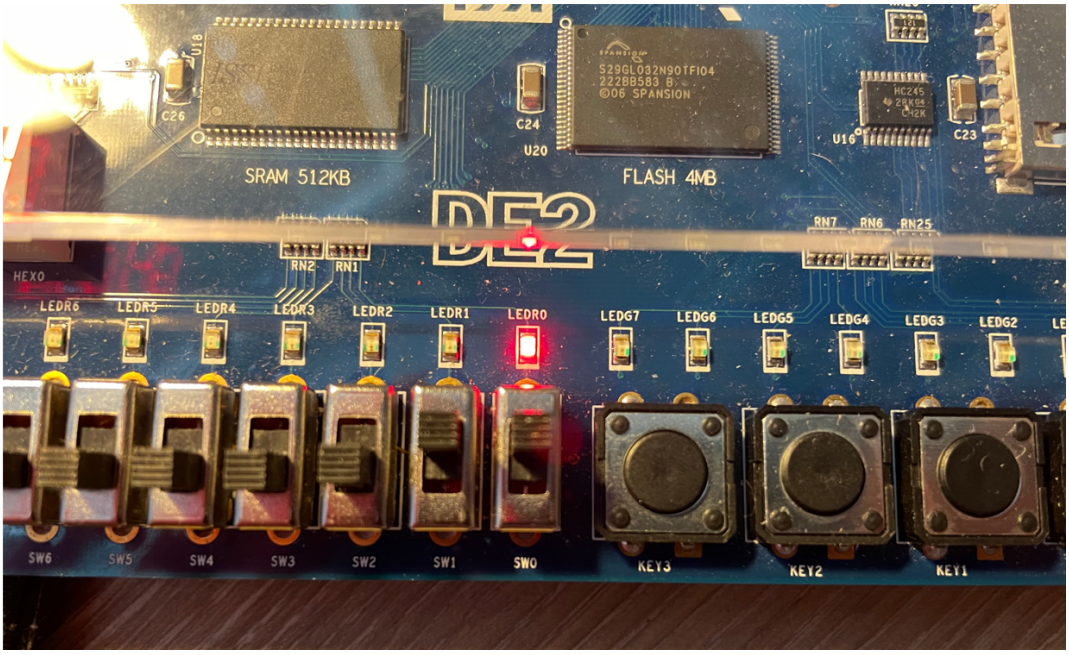
This is a picture of my XNOR waveforms in Synopsys indicating the combinational logic where the output would be 1 or 0.



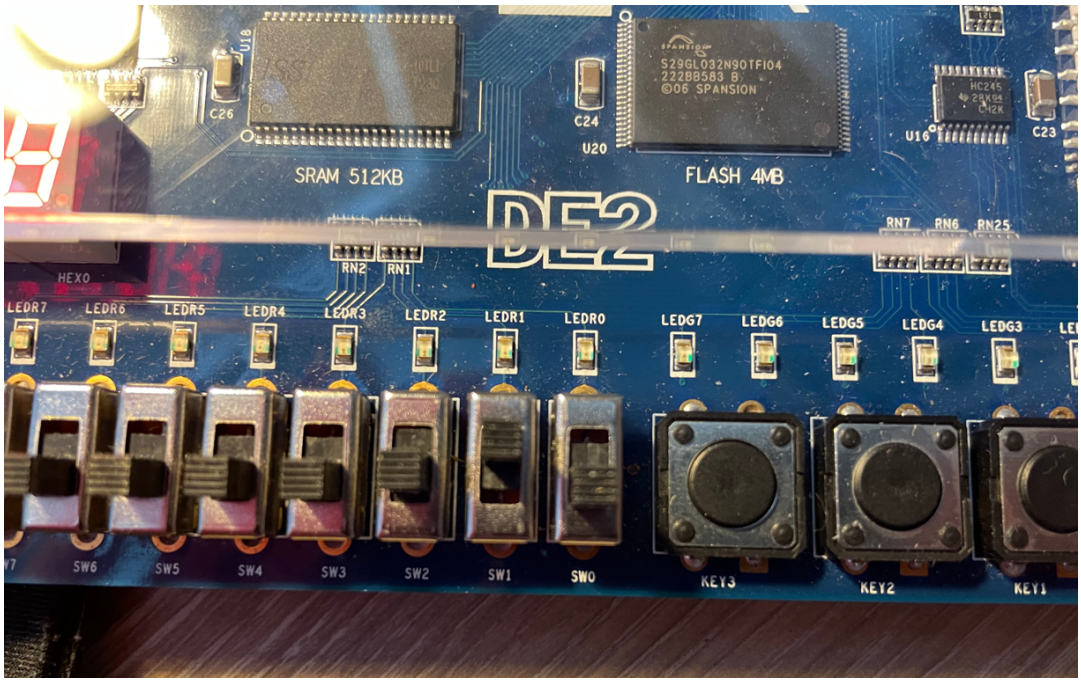
This is a picture of my SystemVerilog code for an XNOR gate with 2 inputs and 1 output.



This is a picture of the schematic for my XNOR gate with 2 inputs and 1 output.



This is a picture of a DE2 board following an XNOR gate when the inputs are 1 and 1 leading to an output of 1 (on).



This is a picture of a DE2 board following an XNOR gate when the inputs are 1 and 0 leading to an output of 0 (off).

4. Answers to questions

1. Question 1:

Register Transfer Level (RTL) is an abstraction that describes a circuit at a lower level than behavioral level. RTL code may include combinational logic, flip-flops, etc. It replicates how a synchronous circuit would transfer data between logical functions and hardware registers. RTL focuses on the signals/combinational logic between registers.

2. Question 2:

HDL and general purpose languages, like C++ or C, vary in several ways. General purpose languages involve procedural programming (series of computations to be carried out) that dictate the steps on how a code will be carried out. Such examples include, while loops, recursive functions, classes, structs, etc. The general purpose language is to be written how instructions would be to carry out to complete a task. In comparison, HDL involves concurrent programming (several computations executed during a given time period rather than in a sequential order) that replicates multiple parallel instructions. Such examples include, D flip-flops, ALUs, adders, subtracters, etc. HDL is written how these components would act between each other by knowing the RTL of how the combinational logic would be carried out.

3. Question 3:

A module is a segment of verilog code that indicates how a circuit is implemented by certain functionalities, and it describes how inputs, outputs, wires, registers, etc. are delivered across the verilog code. Modules are the building blocks of your design that allows you to design a hierarchy of multiple modules.

5. Conclusions & Summary

Overall, this lab was very great for review material on System Verilog HDL. I was able to clearly understand and review my knowledge for future labs to come. There were absolutely no problems with downloading the DE2 board onto my computer nor completing the XNOR gate or the AND gate. I hope to extend my understanding of HDL implementation onto hardware throughout this course.