Class:	CPE-300L	Semester:	Spring 2020

Points		Document author:	Brysen Kokubun
		Author's email:	Brysenkokubun96@gmail.com
		Document topic:	Prelab 2

Instructor's comments:

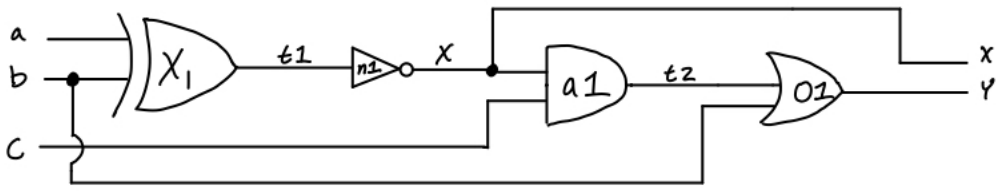
Introduction / Theory of operation

For this lab we will familiarize ourselves with Quartus 7.1, ModelSim and review the Verilog HDL. Verilog HDL was originally developed as a tool for simulation and testing for digital systems but later evolved and used for synthesizing digital designs. Verilog HDL is similar to C language but instead of being procedural operating sequential algorithms, Verilog is for modeling hardware and operates concurrently.

Within Verilog HDL there are continuous assignment statements which allows Boolean equations describing relationships between inputs and outputs to become an assignment statement. There are also primitives, which have built in gate logic: or, xor, nand, and, nor, etc. Verilog also has the ability to use propagation delays for gates and signals. Within this lab we will learn about Verilog value types such as logic 1/0, Outputs, Nets, Regs etc, also within the lab we will learn how to properly use vectors which consists of a set number of bits that sends a signal. The last outcome from this lab was to give us the knowledge of Verilog's operators and their functionalities within Quartus.

Prelab Deliveries:

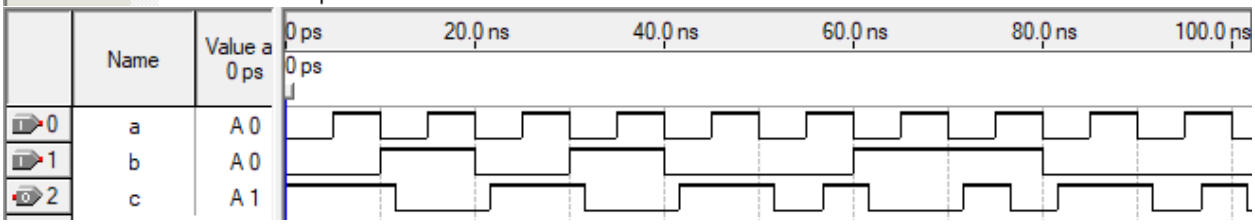
1. **Not Required**
2. **Schematic of Circuit from code**



3. Verilog Code for XNOR gate

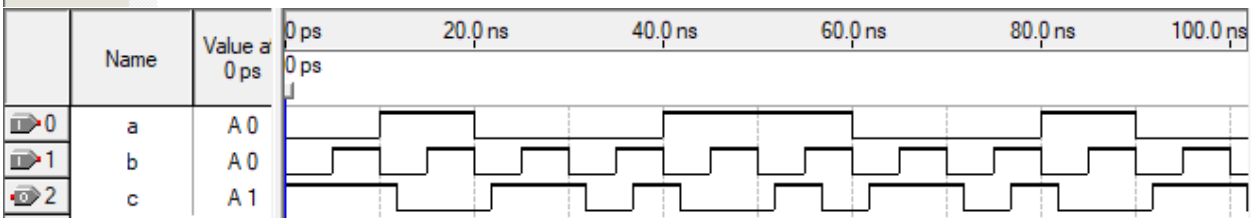
a. Normal Verilog Style

```
1  module preLabXnor(c, b, a); //Normal Verilog Style
2      output c;
3      input b, a;
4      assign c = a~^b;
5  endmodule
```



b. ANSI-C Style

```
1  module preLabXnor(output c, input b,a); //ANSI-C style
2      assign c = a~^b;
3  endmodule
```



4. Verilog Code for 4 input AND Gate

```
1  module PrelabAND4(x, a, b, c, d);
2      output x;
3      input a, b, c, d;
4      assign x = a&b&c&d;
5  endmodule
```