# Project 1: Verilog Introduction

***I am submitting entirely my own work in this exercise, and I am aware of the penalties for cheating if I submit work that is not my own.***

Name: _____ Date: _____

Demonstration

| # | Demonstration | Pts | Score |
|---|---|---|---|
| 4 | Demo Circuit Design | 10 | |

Printouts

| # | Submit | Pts | Score |
|---|---|---|---|
| 1 | Simulation | 5 | |
| 2 | Simulation | 5 | |
| 3 | Simulation | 5 | |

Online Submission

| # | Submit | Pts | Score |
|---|---|---|---|
| 1 | Verilog Module Source | 5 | |
| 2 | Verilog Module Source | 5 | |
| 3 | Verilog Module Source | 5 | |
| 4 | Verilog Module Source | 5 | |

**Overview**

This project will introduce Xilinx ISE and Verilog by requiring you to write, simulate, and build digital circuits using Verilog. To learn how to use Xilinx ISE, there are 2 tutorials on Moodle:

1. ***Xilinx Vivado Verilog Tutorial***
   overviews how to create a project, design a digital part with Verilog, and create a .bit file to load on the Basys3 board.

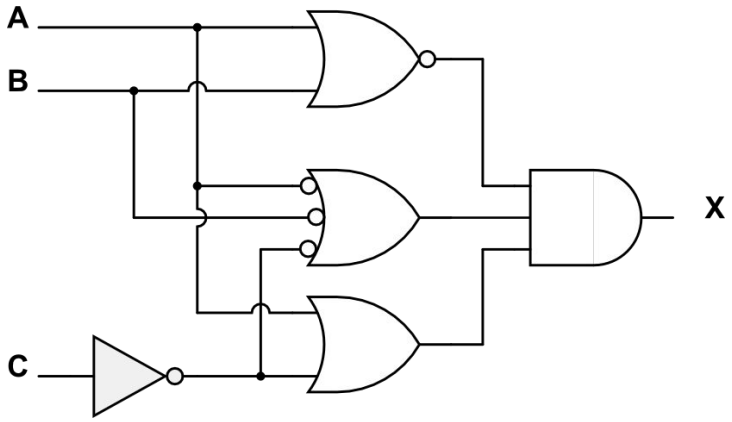
2. ***Xilinx Vivado ISim Tutorial***
   overviews how to simulate a design on Vivado with ISim simulator

**Problem 1.** Create a structural Verilog module for a 4-input, 2-output logic system that behaves according to the two logic equations shown. Simulate your design to verify it behaves correctly. Print and submit your simulation plot. Submit your Verilog module file online.

$$X = \overline{A} \cdot C + A \cdot \overline{B} \cdot \overline{D} + A \cdot C \cdot D + A \cdot \overline{B} + B \cdot D$$

$$Y = C \cdot D + A \cdot B + A \cdot \overline{C} + \overline{A} \cdot \overline{B}$$

**Problem 2.** Create a structural Verilog module for the logic circuit shown below. Simulate your design to verify it behaves correctly. Print and submit your simulation plot. Submit your Verilog module file online.




**Problem 3.** Create a structural Verilog module for a three-input, two-output circuit that behaves according to the truth table shown. Simulate your design to verify it behaves correctly. Print and submit your simulation plot. Submit your Verilog module file online.

| A | B | C | X | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |

**Problem 4.** Create a structural Verilog module that can detect all prime numbered terms in a five input truth table. (FYI, 1 is not considered prime) Assume that the five inputs to your circuit (A B C D E) are used to form a five-bit input truth table with 32 possible input combinations. Each Minterm and Maxterm will be numbered in order 0 to 31 as shown below

| Term | ABCDE | Minterm | Maxterm |
|---|---|---|---|
| 0 | 00000 | $\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E}$ | $A + B + C + D + E$ |
| 1 | 00001 | $\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot E$ | $A + B + C + D + \overline{E}$ |
| 2 | 00010 | $\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D \cdot \overline{E}$ | $A + B + C + \overline{D} + E$ |
| 3 | 00011 | $\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D \cdot E$ | $A + B + C + \overline{D} + \overline{E}$ |
| 4 | 00100 | $\overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} \cdot \overline{E}$ | $A + B + \overline{C} + D + E$ |
| 5 | 00101 | $\overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} \cdot E$ | $A + B + \overline{C} + D + \overline{E}$ |
| 6 | 00110 | $\overline{A} \cdot \overline{B} \cdot C \cdot D \cdot \overline{E}$ | $A + B + \overline{C} + \overline{D} + E$ |
| 7 | 00111 | $\overline{A} \cdot \overline{B} \cdot C \cdot D \cdot E$ | $A + B + \overline{C} + \overline{D} + \overline{E}$ |

. . .

Your circuit should illuminate an LED whenever the input combination forms a prime number term. After you have verified the circuit through simulation, download it to the Basys3 board, using five slide switches as inputs, and a single LED as output. Submit your Verilog module file online and demonstrate your Basys3 board in class.

For programming the Basys3 board use the following switches and LEDs

A: Switch 4
B: Switch 3
C: Switch 2
D: Switch 1
E: Switch 0

Output: Led 0

*When creating a .xdc file, (refer to the tutorial) you will need to know the pin numbers for the different components (switches, push buttons, LEDs, 7 segments) on the board which can be found in the reference manual for the Basys3 board.*