CECS 201 Lab #1 – Spring 2019

Name _____

General Statement: Implement function **F1 = WY'(X|Z) | X'Z(W|Y) | W'X(Y|Z')** using two-level NAND logic.

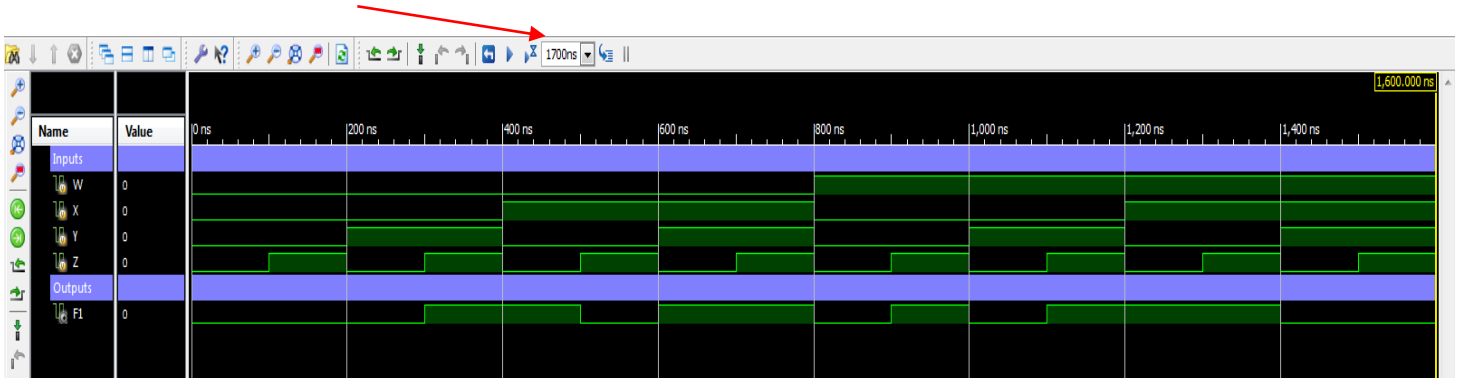
Thus, you are to implement the logic using a "structural" Verilog module, and then simulate it using the **Xilinx ISE Simulator**. You are to get a printout of the Simulation results (i.e. the timing diagram). You must turn in **(1)** this cover sheet, followed by **(2)** a printout of the Verilog module, followed by **(3)** a printout of the simulation waveform via the Xilinx Simulator. You must also implement and download the design into the Spartan FPGA on the Digilent proto-board to verify the correct operation of the circuit.

When creating your ***"New Project"*** in the Project Manager, make sure to select the **"Spartan 6"** family of logic and specify the "**xc3s500e-fg320**" parts (or whatever FPGA part is on your Digilent board) in the ***"New Project"*** dialog box!

After completing the "structural" Verilog module for your logic for **F1**, you must click on the Tools → Check Syntax option. If there are no **errors**, you must then save your Verilog module before going back to the Project Navigator window.

Verification of the circuit is done by checking the output(s) in Xilinx ISE Simulator for each combination of inputs. For example, what is the output when the inputs (**W,X,Y,Z**) are "0 0 0 0?" Is the output **F1** correct or not? What is the output when the inputs are "0 0 0 1?" Is that correct? You must check all combinations in this manner. It is not uncommon for "most" of the combinations to be correct, yet a few are incorrect. This means something is wrong with the original design. In that case, you must go back to your K-map(s) and double check your work. If it seems okay, then go back into the Verilog module and check your work there, etc. etc. **Remember: "Design is iterative."** The method we will use to specify the sequence of input "test patterns" is a verilog "Testbench" file.

Directions for creating a verilog "Testbench" file are found in **Step 3** of the "**Xilinx ISE 14.x Tutorial**" (201 Course CD or Website). Since Lab1 has 4 inputs, you'll specify 16 patterns. To generate the input/output waveform, follow the directions that are in **Step 4** of the **Tutorial**, except the time will be from 0 to 1700 ns. **Note that the Xilinx Simulator only runs for 1000 ns** (default) but we want to simulate 1700 ns. Thus, you must type in the length of simulation you want (e.g. 1700) and "rerun." The output should look similar to what is shown below:



Once your design is verified using the Xilinx simulator, you are ready to "Implement" and "Download" your design into the FPGA. Connect inputs **W,X,Y,Z** to the far right slide switches **SW3**, **SW2**, **SW1** and **SW0**. Connect the **F1** output to LED **LD0**. Directions for implementing and downloading are found in **step 5** and **step 6** of the **Tutorial**.

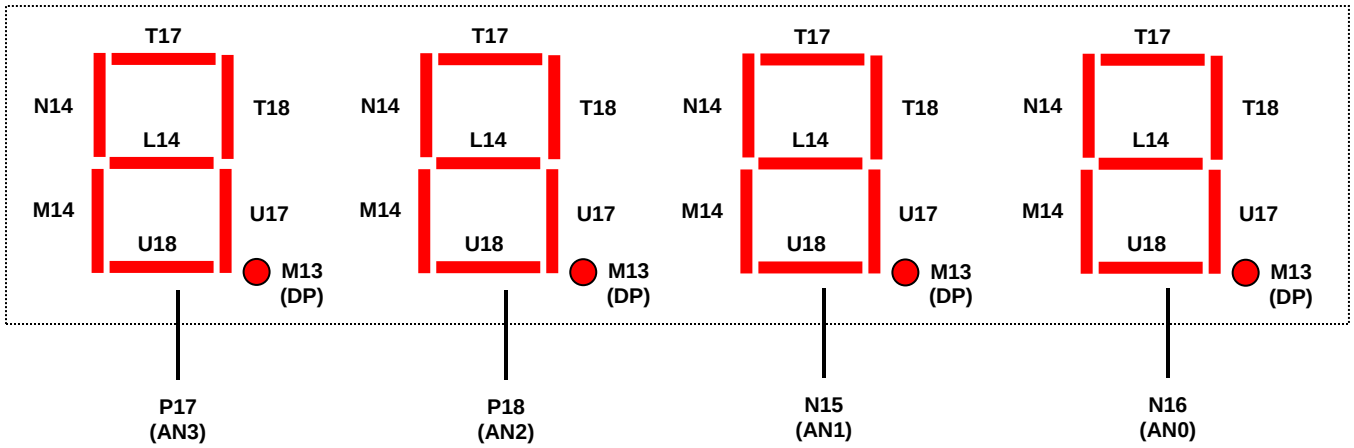
Deliverables: you are to turn in **(1)** a clean copy of this cover sheet (p. 1), **(2)** a printout of the actual Verilog module (with appropriate "**header block**") and **(3)** a printout of the simulation waveforms with you **name printed** on it. Be sure to print the simulation waveform results using "landscape" mode.

Due Date: **Monday, February 25, 2019** {Mon. of week 6}

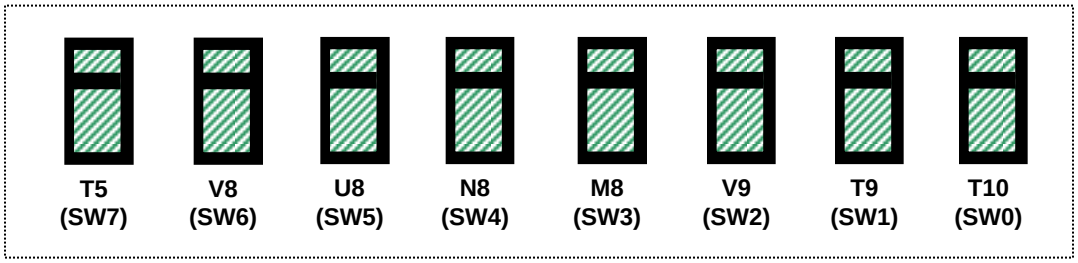
Digilent Nexys 3 Board – Basic I/O Pin Assignments

100 MHz Clock
V10

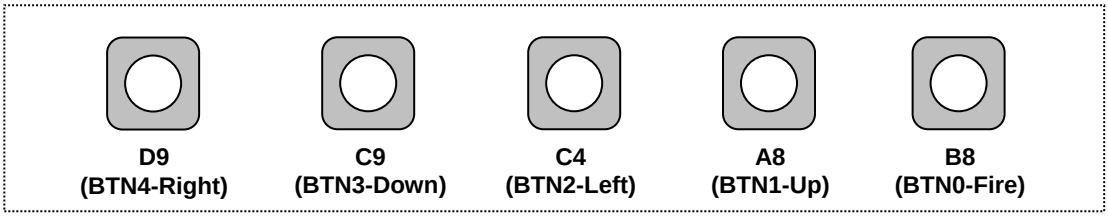
7-segment displays



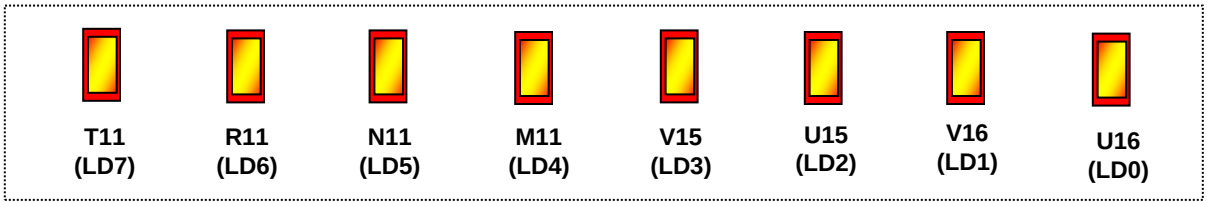
Slide Switches



Push Button Switches



LED's



Others

RS-232

(RxD)	N17
(TxD)	N18

VGA

(Red2)	N7	(Grn2)	V6				
(Red1)	V7	(Grn1)	T6	(Blu2)	T7	(HSynch)	T4
(Red0)	U7	(Grn0)	N8	(Blu1)	R7	(VSynch)	P7

PS2 KeyBoard

(PS2Data)	J13
(PS2Clk)	L12

Verilog Testbench for Lab 1

```
`timescale 1ns / 100ps
module CECS201_Lab1_TestBench();

   // ********************************************************************************
   // Designer:
   // Date:
   // Email:
   //
   // Description:
   //
   // Version:
   // ********************************************************************************

   // Inputs
      reg W;
      reg X;
      reg Y;
      reg Z;

   // Output
      wire F1;

   // Instantiate the Unit Under Test
       Lab1 uut (
            .W(W),
            .X(X),
            .Y(Y),
            .Z(Z),
            .F1(F1);

   // Initial Block for the sequence of "test vectors," applied every 100 ns

      initial begin

        {W,X,Y,Z} = 4'b0000;  #100;  //t =  000 to  100
        {W,X,Y,Z} = 4'b0001;  #100;  //t =  100 to  200
        {W,X,Y,Z} = 4'b0010;  #100;  //t =  200 to  300
        {W,X,Y,Z} = 4'b0011;  #100;  //t =  300 to  400
        {W,X,Y,Z} = 4'b0100;  #100;  //t =  400 to  500
        {W,X,Y,Z} = 4'b0101;  #100;  //t =  500 to  600
        {W,X,Y,Z} = 4'b0110;  #100;  //t =  600 to  700
        {W,X,Y,Z} = 4'b0111;  #100;  //t =  700 to  800
        {W,X,Y,Z} = 4'b1000;  #100;  //t =  800 to  900
        {W,X,Y,Z} = 4'b1001;  #100;  //t =  900 to 1000
        {W,X,Y,Z} = 4'b1010;  #100;  //t = 1000 to 1100
        {W,X,Y,Z} = 4'b1011;  #100;  //t = 1100 to 1200
        {W,X,Y,Z} = 4'b1100;  #100;  //t = 1200 to 1300
        {W,X,Y,Z} = 4'b1101;  #100;  //t = 1300 to 1400
        {W,X,Y,Z} = 4'b1110;  #100;  //t = 1400 to 1500
        {W,X,Y,Z} = 4'b1111;  #100;  //t = 1500 to 1600
        {W,X,Y,Z} = 4'b0000;  #100;  //t = 1600 to 1700
        $finish;

      end  // end of initial block

endmodule
```

Note: the "name" of the instantiated module (i.e. Lab1) and the port names MUST BE the name specified in the Verilog module that you created. Thus, change this appropriately.