ECE 270 Lab Verification / Evaluation Form

Experiment 10

Evaluation:

IMPORTANT! You must complete this experiment during your scheduled lab period. All work for this experiment must be demonstrated to and verified by your lab instructor *before the end* of your scheduled lab period.

STEP	DESCRIPTION	MAX	SCORE
Pre-lab 1	State transition diagram	4	
Step 1	Original source file functionality verification	1	
Step 2	Scrolling display shift register	4	
Step 3	Character sequence generator	8	
Step 4	Self-starting Johnson counter	4	
Step 5	Johnson counter state decoder	2	
Step 6	Thought questions	2	
	TOTAL	25	

Signature of Evaluator: ____

Academic Honesty Statement:

"In signing this statement, I hereby certify that the work on this experiment is my own and that I have not copied the work of any other student (past or present) while completing this experiment. I understand that if I fail to honor this agreement, I will receive a score of ZERO for this experiment and be subject to possible disciplinary action."

Last Name (Printed): ______________________________ **Lab Div:** _____ **Date:** ____________

E-mail: __ __ __ __ __ __ __ __ @purdue.edu **Signature:** __________________________________

Scrolling 7-Segment LED Display

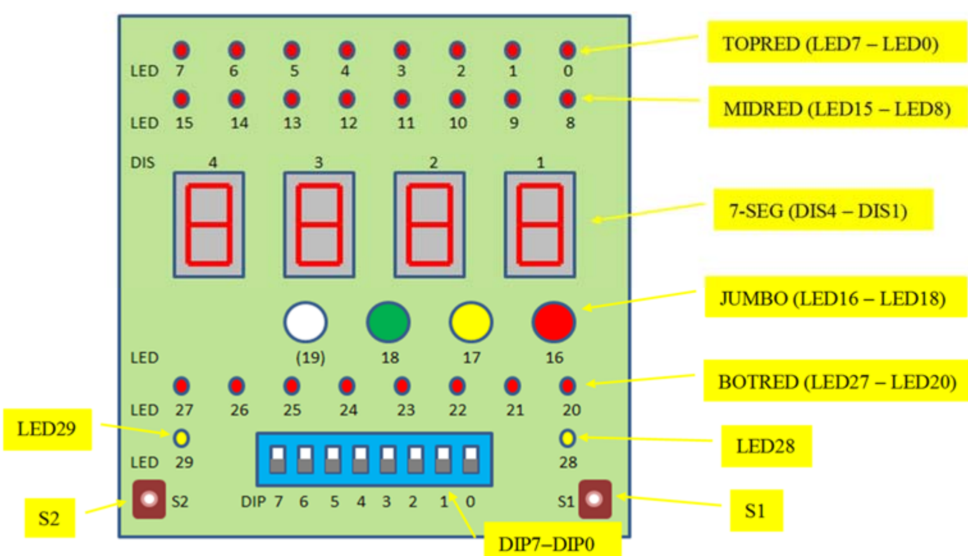
Instructional Objectives:
- To practice creating a sequence generator
- To gain experience writing structured code

You are *highly encouraged* to install a copy of **ispLever Classic (V2.0)** on your own machine in order to make the time spent in lab more efficient and effective – it can be downloaded and licensed by following this link: http://www.latticesemi.com/ispleverclassic)

Prelab Preparation:
- Read this document in its entirety
- Review the referenced Module 3 lecture material
- Write a “rough draft” of all the Verilog code modules before coming to lab; compile them using ispLever to eliminate as many syntax errors as possible

Experiment Description:
In this experiment you will design a scrolling message display (a form of “sequence generator”). Scrolling messages (on the four 7-segment displays) will be utilized in most of the remaining experiments. The scrolling messages to be displayed on the four 7-segment LED displays are indicated in the table below. The selected sequence should scroll across the four 7-segment displays (DIS4-DIS1) from right to left, i.e., the first character of the sequence should initially appear on the right-most display, then *shift left* one position as the next character appears on the right-most display, etc. Two control signals (entered on DIP[1] and DIP[0]) will be used to specify the message to display, as shown. If the message selection entered on the DIP switches is changed while another message is being displayed, *the current message should be completed before the newly selected one starts*. Note that two blank characters are appended to the end of each message so that each message instance will scroll off the left-most display as the next one starts.

DIP[1]	DIP[0]	*Scrolling Message Displayed*
0	0	*blank* → g → o → *blank* → P → U → r → d → U → E → *blank* → *blank* → …
0	1	*blank* → n → o → I → S → E → *blank* → *blank* → …
1	0	*blank* → b → o → I → L → E → r → *blank* → U → P → *blank* → *blank* → …
1	1	*blank* → I → U → *blank* → L → o → S → E → r → S → *blank* → *blank* → …



Pre-lab Step (1):

In the space below, draw a state transition diagram that specifies the behavior of the scrolling display – note that a Moore model should be utilized.

Step (1):
Carefully examine the `lab10_top_template.v` file provided for this experiment on the course website. Rename this file *myuserid*_`lab10.v` (where *myuserid* is your 8-character login name) and insert your identifying information where indicated. Next, open ispLever and create a new Verilog project named `Lab10`, select `LC4256ZE-5TN144C` as the device to use, and import your *myuserid*_`lab10.v` source file. Compile the program and load the JEDEC file produced by ispLever into the CPLD. Verify that the two bounceless switches function as expected, and that the Jumbo LEDs blink at the expected rates.

Step (2):
Write a Verilog module that realizes a 7-bit wide, 4-word "left shift" register to facilitate scrolling messages on the four 7-segment LEDs. Route each "display code word" of the shift register's outputs to the corresponding 7-segment display (DIS1-DIS4). Route DIP[6]-DIP[0] to the inputs of the shift register's right-most word. For debugging purposes, clock your shift register using the right pushbutton (S1BC); use the left pushbutton (S2BC) to asynchronously clear your shift register. Change the DIP switch values as the display is scrolling to verify correct operation. Once you have verified that your scrolling display is working as expected, change the clocking source for the shift register to `tim_div4`.

Step (3):
Write a Verilog module that realizes a state machine that "looks up" the next character to display based on the message selected on DIP[1] and DIP[0]. If the display mode is changed mid-sequence, *the message in progress should complete before the newly selected one starts*. Use the left pushbutton (S2BC) to asynchronously clear your state machine. Use a new instance of the module created for Step 2 to scroll the message strings. Use `tim_div4` as the clocking source for your scrolling display.

Checkpoint: Demonstrate Step 3 to your Lab Instructor.

Step (4):
Write a Verilog module that realizes a 4-bit self-starting Johnson counter (as illustrated on page 28 of the *Lecture Summary* notes). Route the output of your Johnson counter to the four right-most MIDRED LEDs. Use the right pushbutton (S1BC) to clock the Johnson counter, and the left pushbutton (S2BC) to provide it with an asynchronous reset.

Step (5):
Write a Verilog module that decodes the states of your Johnson counter (as illustrated on page 29 of the *Lecture Summary* notes). The 8 decoded states should be displayed on the TOPRED row of LEDs, where state 0000 of the Johnson counter (i.e. its initial state) is indicated by the right-most LED of TOPRED. After verifying that your Johnson counter and state decoder modules work correctly, change the clocking source from S1BC to `tim_div`.

Checkpoint: Demonstrate Step 5 to your Lab Instructor.

Step (6): Write your answers to the following Thought Questions in the space provided**.**

1. Examine the fitter report generated by ispLever and determine the total number of flip-flops utilized by your design as well as the total number of P-terms and macrocells.

   Number of flip-flops: ______________________________

   Number of P-terms: ______________________________

   Number of macrocells: ______________________________

2. Describe how you would approach this design differently if any of the character sequences were longer (than specified) or there were more character strings (than four).

   ____

   ____

   ____

