ECE 3829: Advanced Digital System Design with FPGAs - A Term 2020

Lab 1: Combinational Logic Design

Demo and signoff due during first week of labs (week of August 31st)

Lab exercise 1: (worth 5% of total lab grade)

This lab is to remind you how to use the Xilinx Vivado Design Suite tools and the Basys3 board to implement a simple combinational logic design.

Background Review

Complete the Verilog Decoder tutorial - you may find it helpful to also look at the Verilog Counter tutorial showing how to instantiate a lower level module.

Description

Part 1:

- Create a new module called 'seven_seg' in a separate file. This should allow for four digits (each one 0 to F) to be displayed on the four seven-segment displays. This module should have the following ports:
  - Four 4-bit inputs (A, B, C, and D) and one 2-bit input (called 'SEL')
  - A 7-bit output to drive the seven-segment cathodes and four 1-bit anodes to drive the four anodes
- Connect 4 slider switches to each of the A, B, C and D inputs of the seven_seg module.
- Connect the 'SEL' input to two push-button switches
- When no push-buttons are pressed, display the value of the lower 4-bits of the slider switches (A) on seven-segment display #1
- When one of the push-button is pressed, display the value of the next lower 4-bits of the slider switches (B) on seven-segment display #2
- Do the same with other push-button combinations to allow the remaining slider switches to be displayed on seven_segment display #3 and #4.
- Test this out on the board before proceeding with part 2

Part 2:

- Create a new top_level module called 'lab1_top' in a second file that has inputs to the slider switches and push buttons, and outputs to the seven segment display.
  - Instantiate (using *named* association) the seven_seg module from part 1 into the top level module

- Add additional Verilog statements to the **top level** module to add the following functionality:
    - Use 4 LEDs to display the value of input ‘A’ added to input ‘B’ (ignore any carry).
    - Use 8 LEDs to display the value of input ‘A’ multiplied by input ‘B’.
    - Use 4 LEDs to display the last digit of your WPI ID number.

Note: This is a combinational logic design, no clocks are required.

No lab report is required for this exercise but you need to submit a copy of your TWO Verilog files (make sure you include a header with your name, date, and description – and add comments) to Canvas, and demo your working lab (either in-person during a lab session, or through zoom) during the first week of class (deadline is Sunday 4pm).

You do not need to submit a copy of your XDC constraints file.

This lab is to be completed individually.

Reference Material

Complete the decoder tutorial.

Review the counter tutorial to see how to instantiate a lower-level module:

```
// instantiate copy of display module using named association
// counter_10 signal is connected to the digit port
// seg signal is connected to the seven_seg port
display disp1 (.digit(counter_10), .seven_seg(seg));
```

Read the *Seven Segment* section in the Basys3 Reference Manual.

Note: If you want to eliminate the two warning messages during the implementation phase add the following to your XDC file (also shown in the counter tutorial XDC file):

```
# Configuration bank voltage select (CFGBVS) must be set to VCCO or GND, and CONFIG_VOLTAGE must be set to the correct configuration voltage,
# in order to determine the I/O voltage support for the pins in bank 0.
set_property CFGBVS VCCO [current_design]
#where value1 is either VCCO or GND
set_property CONFIG_VOLTAGE 3.3 [current_design]
#where value2 is the voltage provided to configuration bank 0
```