SANTA CLARA UNIVERSITY	ELEN 127 Fall 2018	Jim Lewis

Laboratory #1: 7-segment Display

For lab section on September 28, 2018

I. OBJECTIVES

Refresh the brain muscles that have atrophied since you took ELEN 21.

PROBLEM STATEMENT

You will be defining your own alphabet of sorts and using the Altera DE2-115 evaluation board (which you used in ELEN 21) to demonstrate. You will pick eight characters that can be effectively rendered with a 7-segment display and associating a unique 3-bit code with each. Then, using groups of 3 switches on the evaluation board, you will drive "words" of different letter combinations on the 7-segment displays on the evaluation board.

II. PRE-LAB

1. Look at the Altera lab that this lab is modeled after. Note the use of a module called "7-segment decoder", which takes a 3-bit input and then generates the correct 7 control signals for a 7-segment display. The first part of your pre-lab is to clearly define the behavior for **your** 7-segment decoder. It's **yours** because **you** will be defining the eight 3-bit encodings and their associated characters, i.e., its behavior. So you're effectively writing a behavioral spec.

   Note that this is different than what is stated in the Altera lab, where they specify four encodings, for the letters H, E, L, and O. You will be defining 8 unique encodings, and you're not limited to the characters you can define. You can even define characters that don't map to English letters, if you so choose. You can also define a "space" (blank) as one of your 8 characters.

   In addition to the behavioral spec, you should also specify the logic to generate the 7 control outputs, given all 8 possible 3-bit input encodings, so that you are ready to start implementing your decoder once you get into the lab. You can specify the logic anyway you want: truth table(s), logic equations, logic gate schematics, etc.

   Submit both your behavioral spec and your logic definitions.

   Also, you should re-acquaint yourself with Quartus, the tool you assumably used when you took ELEN 21. The tutorials and pin reference will be posted in Camino.

LABORATORY PROCEDURE

Since there are not enough eval boards to go around, you will be paired with someone.

1. Implement both of your 7-segment decoders:
   (i) You can do this either with gates, or with Verilog if you're feeling sufficiently comfortable with it. (Starting with the next lab, everything will be done in Verilog.)
   (ii) Simulate both to make sure they appear to be working properly.
   (iii) Create a symbol for your decoders so you can instantiate them at the next higher level. You can call the two decoders whatever you like (decoderA and decoderB, decoder1 and decoder2, Bert and Ernie, …)

2. Implement a 3-bit 4:1 Mux:

   (i) You're going to be instantiating four of your 7-segment decoders, and choosing which of the four 3-bit character inputs get sent to which decoder. So you'll need 4:1 muxes to route the inputs. But each quantity is a 3-bit character, so you need to implement a 3-bit 4:1 mux. Work with your partner to get this implemented.

   (ii) Create a symbol for this mux that you can instantiate at the next higher level.

3. Assigning Inputs and Outputs:

   (i) Your circuit will need to support 4 independent characters at any time. Since each character is specified by a 3-bit code, you'll need to define four groups of 3 input switches as your character inputs.

   (ii) You'll need to define two more switches as the mux selects for the 4:1 muxes.

   (iii) And you're going to drive four of the 7-segment displays, so you'll need to define four groups of 7-segment display outputs. Each group will be driven by an instantiation of your 7-segment decoder.

4. Top-level Design:

   (i) Each of the four 7-segment displays will be driven by an instance of a decoder, so there will be four decoders instantiated. Since you have two different decoders, you will be instantiating two of each type. To make life easier for everyone, use the same decoder design for the left two displays and the other decoder for the right two displays.

   (ii) Supplying the input to each decoder will be one of your muxes, so you'll need to instantiate four of these.

   (iii) The muxes will be taking four 3-bit inputs. These will come from your four 3-bit groupings of switches. And then the select signals will be connected to the two additional switches you have chosen for this purpose.

   (iv) The mux selects should bring about the following behavior. Assuming characters 3210, reading left to right, the 7-segment displays should read (left to right):

Select value	Character output
00	3210
01	2301
10	1032
11	0123

   (v) You might want to some simulation to check this before downloading.

5. Programming the FPGA and testing on the DE2-115 board:

   Download your design and test it operationally. When you are sure that it operates correctly, demonstrate it to the TA.

   If the TA runs a test case that doesn't work properly, you need to debug it, fix it, and have the TA check it again.

   Take a picture of the working FPGA showing the input switches and the 7-segment display.

III. REPORT

Even though you worked as partners in lab, you should turn in individual final reports.

For your lab report, include the behavioral spec for both your decoder and your partner's decoder (for reference), the logic in your decoder (you need not include your partner's logic), and a block diagram of how the two of you ultimately connected the muxes. In addition, include answers to the following questions.

- When you simulated your top-level design, in section 4(v), did you find any problems? Were any of them internal to the decoder or was it a mistake in your connections at the top level? If you found any problems in the decoder, what can you say you learned about your testing of just the decoder, in section 1(ii), even if the problem was in your partner's decode and not yours?
- Similarly, if you or the TA found problems while testing on the eval board, what can you say you learned about the simulation you did in section 4(v)?
- If you never got your design to work, where did you get stuck? List out issues that may have contributed to you getting stuck.