

CS/EE 3700-- Lab Assignment #2

Board demo due in your lab during the week of February 11th
Verilog and simulation due on Thursday February 7th, 5:00pm
Hand in printed material to the box outside the SoC office

This lab will have you write a circuit description in Verilog, simulate that Verilog code, synthesize that code to a circuit, and map that circuit onto the FPGA board for demonstration. You will use ONLY the XSA-50 board for this lab which is the smaller board with the parallel port and the FPGA chip. You will NOT use the XST board that you used in Lab1. You will not have to wire anything for this lab. Instead you will map the circuit to the FPGA and use the switches and LEDs on the XSA-50 board for demonstration.

There is a tutorial that goes along with the lab that is linked to the class web site.

Specifically, the circuit you will build for the lab is a BCD to 7-segment display circuit. BCD stands for Binary Coded Decimal and it just means that you can represent the 10 decimal digits using four binary digits

BCD Numbers

ABCD (inputs) = output digit (decimal digit)

0000 = 0
0001 = 1
0010 = 2
0011 = 3
0100 = 4
0101 = 5
0110 = 6
0111 = 7
1000 = 8
1001 = 9
1010 = Error
1011 = Error
1100 = Error
1101 = Error
1110 = Error
1111 = Error

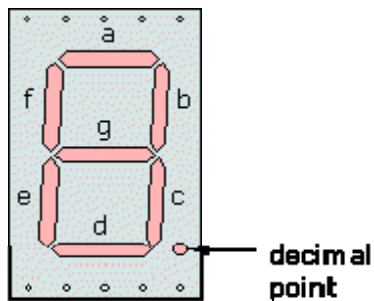
Because there are 16 possible values for four bits, there are 6 combinations that do not correspond to a decimal digit, so those cases should not be used in a pure BCD representation. Note that they can and are used if the inputs are assumed to be hex

values instead of BCD. In the hex case, the last six values are the hex digits A, B, C, D, E and F.

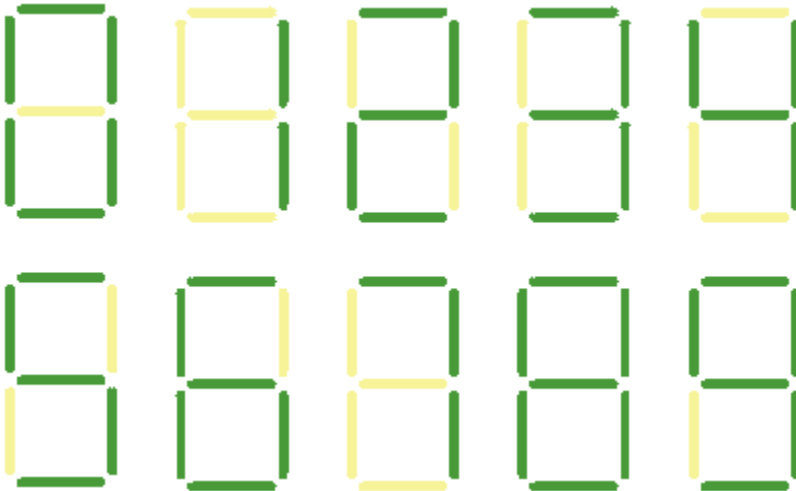
7-Segment LED

A 7-segment LED is a set of seven LEDs in a single package that are arranged in a pattern that lets them represent decimal numbers. I'm sure you've seen these all over the place! The LED segments are organized as shown:

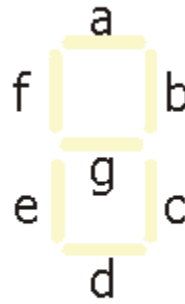
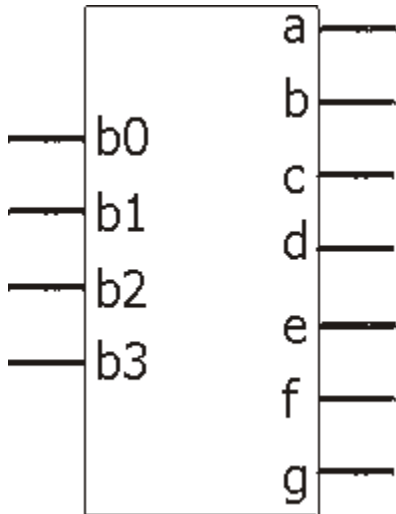
7-segment display



By lighting up different combinations of the 7 LEDs (or 8 if you include the decimal point), you can make them look like all the decimal numbers. The patterns are:



The circuit that you will be designing will take four inputs from switches, and produce seven outputs: one output for each of the seven segments of the 7-segment LED. That is, you will derive seven different Boolean functions. Each function will drive one of the seven segments. The external interface of your circuit will look like:



The b0, b1, b2, and b3 inputs are the BCD inputs (which will come from switches on the XSA_50 board), and the a, b, c, d, e, f, and g outputs are separate Boolean functions for each of the seven LED segments. The LEDs on the XSA_50 board will light if the signal driving them is high, and be off if the signal driving them is low.

For example, if the inputs are 0001, then you want segments b and c to be lit (logic one), and all the other segments to be low. So, for an input of 0001 function a will be 0, function b will be 1, function c will be 1, function d will be 0 and so on. You can easily make a truth table for this:

b3	b2	b1	b0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1

and so on...

For this BCD to 7-segment display circuit, if you enter a number on the input switches higher than 9, the display should show E for error.

Procedure

1. Run through the tutorial on the class web site that demonstrates how to use the ISE tools from a Verilog file rather than from a schematic, how to simulate that Verilog file, how to assign the inputs and outputs of your circuit to specific pins on the FPGA, how to synthesize your Verilog into a circuit for the FPGA, and finally how to load it on the FPGA on the XSA-50 board.

2. Fill in the truth table for the BCD to 7-segment circuit for all 16 rows. For each row you should fill in the values of the seven output functions so that the decimal number is displayed. If the input switches are set to a number higher than 9 (1001), the display should show E for error. Each truth table row will represent one configuration of the entire LED array. Each output column will represent one function that drives one segment of the array.
3. Write a Verilog module that takes four inputs (b3, b2, b1, b0) and produces seven outputs (a, b, c, d, e, f, and g) that implement each of the seven functions in the truth table. Use either the gate-style or continuous assignment (assign) style of writing Verilog as shown in Chapter 2 of your text. This will involve describing a Boolean function for each of the seven outputs. See examples in Chapter 2 of Verilog modules that take multiple inputs and produce multiple outputs (Figure 2.35 for example).
4. Simulate your Verilog code as shown in the tutorial. Generate a Verilog test fixture to drive the simulation, fill in the test code in a way that is self-checking, and run the simulation. Print out the waveform window using File -> Print. I don't know how to print the console window directly so either print a screen capture showing your console window, or copy the contents of the console and paste them into text editor and print it that way.
5. Once it's simulating properly, map your pins to the switches and LEDs on the XSA-50 board (as shown in the tutorial). You should use the four-switch DIP on the XSA_50 board for inputs, and the 7-segment LED as outputs. The pins for these components can be found in the XSA-50 documentation and the XSA-50 pin documentation on the class web site. But, I'll repeat the critical information here too. The pins are:

DIPSW1A: Pin P54, input b0
DIPSW1B: Pin P64, input b1
DINSW1C: Pin P63, input b2
DIPSW1D: Pin P56, input b3

LED-A: pin P49, function a
LED-B: pin P46, function b
LED-C: pin P39, function c
LED-D: pin P67, function d
LED-E: pin P62, function e
LED-F: pin P57, function f
LED-G: pin P60, function g

6. Synthesize your Verilog code, Implement it for the Spartan2 part, and generate the .bit bitstream file (as shown in the tutorial).

7. Using the Xess tools (described in the tutorial), upload the bitstream to the XSA-50 board. You should now be able to change the values on the switches and see the numbers (or E) come up on the 7-segment LED. Demonstrate this circuit to your TA during your lab in the week of February 11th.
8. Print out your Verilog code for the circuit, your testbench output (both waveform and console). Hand them in by Thursday February 7th at 5:00 in the box outside the SoC office.