Digital Design and Computer Architecture

Lab 2: Seven-Segment Display

Introduction

In Lab 1, you became familiar with the Quartus schematic editor. In this lab you design a more complicated block of combinational logic. As you are completing the lab, also think about what methods will minimize design time. Use design practices that will make both designing *and* debugging efficient.

Many digital devices use seven-segment displays to represent numbers. Some examples include digital clocks and speedometers. One or more of the seven segments light up to display the desired number. In this lab, you will construct the seven-segment display decoder. It is called a decoder because it takes the four inputs and decodes them into seven outputs. You will also learn more about the schematic editor and ModelSim simulator, including how to draw busses and create formula test vectors.

As always, skim through the entire lab first, and don't forget to refer to the "What to Turn In" section at the end of the lab before you begin. There is also a set of hints on the last page of the lab.

Background

The objective of this lab is to design, simulate, and implement a seven-segment display, as shown in Figure 1. Each of the seven segments is labeled a through g. The numbers 0 through F light up the segments shown in Figure 2. For example, the number 0 lights up all but the middle segment, segment g.

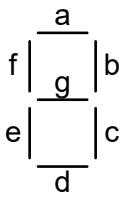

Figure 1. Seven-segment display

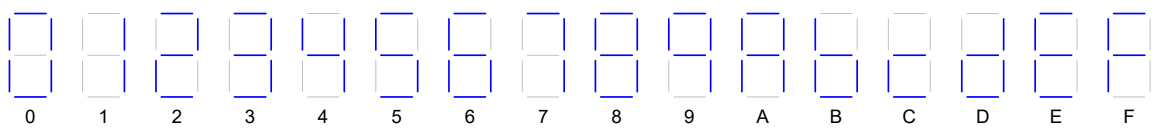

Figure 2. Seven-segment display function

© 2012 David Money Harris and Sarah L. Harris

You will build a seven-segment display decoder, shown in Figure 3. The circuit has four input bits, $D_{3:0}$ (representing a hexadecimal number between 0 and F), and produces seven output bits, $S_{a:g}$, that drive the seven segments to display the number. A segment of the display turns on when it is 0. Such an output is called a *low-asserted output* or *active low output*.

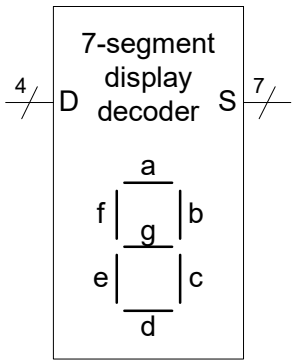

Figure 3. 7-segment display decoder

To design your seven-segment display decoder, you will first write the truth table specifying the output values for each input combination. We have started the truth table for you in Table 1. For example, when the input is $D_{3:0} = 0000$, all of the segments except g should be on. Because the outputs are active low, they must be $S_{g:a} = 1000000$. Complete the truth table for the 7-segment display decoder circuit. You will need to turn in your completed truth table.

Hexadecimal Digit	Inputs				Outputs							(in hex)
	D_3	D_2	D_1	D_0	S_g	S_f	S_e	S_d	S_c	S_b	S_a	
0	0	0	0	0	1	0	0	0	0	0	0	40
1	0	0	0	1								
2	0	0	1	0								
3	0	0	1	1								
4	0	1	0	0								
5	0	1	0	1								
6	0	1	1	0								
7	0	1	1	1								
8	1	0	0	0								
9	1	0	0	1								
A	1	0	1	0								
B	1	0	1	1								
C	1	1	0	0								
D	1	1	0	1								
E	1	1	1	0								
F	1	1	1	1								

Table 1. Truth table for 7-segment display decoder

© 2012 David Money Harris and Sarah L. Harris

After completing the truth table, write equations for each output segment. You should have seven separate equations for S_a through S_g. Next, translate your equations to logic gates and sketch your design. You may use logic gates with any number of inputs. You may choose to optimize for design time or number of gates. Describe your design choice in one paragraph.

Schematic Entry

Now you are ready to enter your design in the schematic editor. Create a new schematic project called lab02_xx as you did in Lab 1.

As in the last lab, choose input and output names so that the design will automatically connect to the switches and LEDs on the DE2 board. Name your inputs SW[3] through SW[0] and your outputs HEX0[6] through HEX0[0]. SW[3] corresponds to D_3 while SW[0] corresponds to D_0. HEX0[6] corresponds to S_g while HEX0[0] corresponds to S_a.

Draw your schematic. Label all of the nodes to make debugging easier. This process is rather tedious as you must zoom in and out. You will soon learn Verilog to make the job much easier.

You may find that you don't have the right sizes of gates available and will have to make your own. For example, you can build a 5-input OR by using a 6-input OR with one input tied to GND, or using a 4-input OR followed by a two-input OR.

Create a Verilog file for your schematic. Watch for warnings in the messages window and fix anything that looks suspicious. Inspect the Verilog file in a text editor such as WordPad. Make sure the design appears to be correct. If you can't tell, it is likely wrong.

Simulation

After you are done drawing your seven-segment decoder logic, your next step is to simulate and debug your design in ModelSim. Follow the directions from Lab 1.

Both input and output bus values can be set to be displayed in either Hexadecimal, Decimal or Binary. Do this by highlighting all of the inputs and outputs. Then right-click and choose **Radix → Binary/Decimal/Hexadecimal.** Choose hex to make your results easy to read.

You can use the force command to drive multi-bit inputs. For example, to test an input of 7, enter

```
force SW 0111
run 100
```

Apply all sixteen possible inputs and check that the outputs agree with your truth table. If they do not, track down and fix your logic errors in the schematic.

When your simulation functions correctly, capture an image of the waveform.

© 2012 David Money Harris and Sarah L. Harris

Hardware Implementation

Download your design to the DE2 board using the steps from Lab 1. Toggle switches 0-3 and check that the display labeled HEX0 cycles through the correct outputs.

What to Turn In

You must submit an electronic copy of the following items via Sakai. These should all be included in a single file (.pdf preferable, but .doc/.docx also acceptable). Be sure to label each section and organize them in the following order. Messy or disorganized labs will lose points.

1. Please indicate how many hours you spent on this lab. This will not affect your grade, but will be helpful for calibrating the workload for next semester's labs.
2. Your completed Table 1.
3. Your output equations for each of the 7 segments.
4. A hand-sketched or computer drawn schematic of your 7-segment display decoder.
5. A paragraph describing your design method and design choice.
6. An image of your Quartus schematic showing your 7-segment display decoder logic (as in Lab 1, File→Export usually works well).
7. An image of your simulation waveforms showing correct operation for all input combinations starting from 0 and going to F (File→Export→Image…). Your waveform should show your inputs on the top and your outputs on the bottom. All values should be displayed in hexadecimal for easy reading.
8. Did your design work correctly on the DE2 board? (this will be evaluated in lab3/colloquium)

Some Altera Notes:

- Make sure you don't have any spaces in any of your folder or file names.
- Some CAD tools are case sensitive and others are case insensitive. Never use two different capitalizations of the same signal because some tools may treat them as different signals while others may treat them as the same signal. The easiest solution is to be consistent in your choice of capitalization.

© 2012 David Money Harris and Sarah L. Harris