Intro to Digital Logic, Lab 4
High-Level Verilog

Lab Objectives

Implementing designs directly in schematics or structural (gate-level) Verilog can give you the best control, and often the smallest designs. But, sometimes it can be a real pain to optimize all the way down at that level. An alternative is high-level (RTL) Verilog, where you tell the CAD tools what you what the output to look like, and it automatically does the Boolean Algebra for you!

Assigned Task – Seven-segment display

In the class notes we presented a seven-segment display driver. RTL code for that seven-segment display is given below. Create a copy of your lab #3 project, then enter the code below into a new System Verilog file. Then, create a new module that instantiates the seg7 code twice – one taking an input from SW3 – SW0 and displaying on Hex0 of the board, another taking input from SW7 – SW4 and displaying on Hex1 of the board.

```
module seg7 (bcd, leds);
   input  logic  [3:0] bcd;
   output logic  [6:0] leds;

   always_comb begin
      case (bcd)
         //          Light: 6543210
         4'b0000: leds = 7'b0111111; // 0
         4'b0001: leds = 7'b0000110; // 1
         4'b0010: leds = 7'b1011011; // 2
         4'b0011: leds = 7'b1001111; // 3
         4'b0100: leds = 7'b1100110; // 4
         4'b0101: leds = 7'b1101101; // 5
         4'b0110: leds = 7'b1111101; // 6
         4'b0111: leds = 7'b0000111; // 7
         4'b1000: leds = 7'b1111111; // 8
         4'b1001: leds = 7'b1101111; // 9
         default: leds = 7'bX;
      endcase
   end
endmodule
```

Note that the 7-segment display on the DE-1 is ACTIVE LOW. That means putting a FALSE on the wire makes it light up, while a TRUE means that light is off. You will have to adjust your design accordingly.

Design Problem – UPC code to display

Review the Verilog Tutorial on the website up to (but not including) Sequential Logic.

In lab #3 we built a system that took in a UPC code and output whether the item was on sale and whether it was stolen. A neighboring store, Fred's House of Useful Stuff, wants a similar system, but has found that people are changing the UPC sticker on their stuff. To combat that, they'd like you to add a display on Hex5 – Hex0 that describes each product when it is entered – if the description is different from the product actually entered, they know someone is trying to cheat! Note that, since Fred pays his employees embarrassingly

little, they tend to be not the brightest folks, so the Hex display should be as simple as possible.

Since Fred doesn't want to pay for a redesign of the circuit for Lab #3, we will use the same UPC codes, On Sale values, and Expensive markings as before BUT, Fred is willing to sell whatever products you will like him to (Hint: pick products that make good Hex displays!), though the expensive ones must use UPC codes designated "expensive", and the inexpensive ones must use UPC codes designated "inexpensive". BE CREATIVE! And you MUST sell 6 items.

Determine your items and Hex displays. You can use any or all of the lights on the hex display you choose, upper and lower case, pictograms, whatever. So, for example you could put in "Dress Shoes" as an item, and have the hex display show "ShOE". Note that Fred was beaten with a pair of loafers as a kid and so is unwilling to sell shoes (i.e. no copying from me! And all your items must be DIFFERENT than lab #3's).

Once you have figured out what you'll display, create a high-level design for the circuit. It will be similar to the seg7 module, with 3 inputs (U, P, and C), but up to 6 7-bit outputs (for each of the 7seg displays). It will be written in RTL. Simulate it in ModelSim, then hook it to the switches and lights of your board to make sure it works. Finally, create a new module that instantiates one copy of your new display code, and one copy of your lab #3. It should hook them both together, so the system simultaneously computes the HEX display, and the Sale and Stolen lights. Test and debug with ModelSim, then load onto your board.

ModelSim Tip: When you put your entire design together and simulate in ModelSim, you'll probably need some help organizing all your signals. A couple tips:

1.) ModelSim can have signals from multiple modules displayed at the same time. Simply select in the left pane the module whose signals you want to display, then drag the signals you want from the middle pane to the waveform window.
2.) To order the signals, you can click and drag names in the waveform window.
3.) To organize signals for each module, highlight all of the signal names related to one module in the waveform window, right-click on one of the signal names, and select "Group". Give it a memorable name, then hit okay. You can now move the signals as a group, and hide/expose them easily.
4.) Once you have organized signals the way you like, remember to save the formatting into the <modulename>_wave.do file.

You will be graded 100 points on correctness, style, testing, test coverage, etc. Your bonus is quality of results. In this lab "quality of results" means how nice/creative your display is. Results that are particularly well done, pretty, amusing, or otherwise makes your TA laugh or applaud will get more points.

Assigned Task – Don't Cares

Your design has outputs for only 6 of the 8 possible UPC codes. For the other two cases, a line such as "default: LEDs = 7'bx;" tells Quartus II that it can treat these cases as a Don't Care condition (if you didn't do this, go back and correct it to do so). Test your design on the circuit board, and record the pattern it shows for these Don't Care conditions.

Lab Demonstration/Turn-In Requirements

A TA needs to "Check You Off" for each of the tasks listed below.

- Demonstrate both the 2x7seg and the Fred's House of Useful Stuff circuits in ModelSim to the TA.
- Demonstrate both the 2x7seg and the Fred's House of Useful Stuff circuits on the DE1 to the TA.
- Turn in a new table, based on the one from Lab #3, that shows what items you are selling, the UPC code, on sale, and expensive.
- Turn in a printout of your Verilog for the design for Fred's House of Useful Stuff.
- Turn in a paragraph explaining why you assigned certain products to "expensive" vs. others to "inexpensive".
- Turn in a drawing of what the circuit does when given each of the unused UPC codes (from Assigned Task – Don't Cares).
- Tell the TA how many hours (estimated) it took to complete this lab, including reading, planning, design, coding, debugging, testing, etc. Everything related to the lab (in total).