CPE 133
Lab 5 - Behavioral Design

The “SystemVerilog Coding Guidelines” posted in Canvas must be followed.

Problem 1 – N-Bit Comparator

Create a behavioral SystemVerilog module to compare two unsigned values to determine three outputs. The scalar outputs are GT, LT, EQ to represent Greater Than, Less Than, or Equal. The width of the input values must be parameterized with a default width of 2.

Instantiate and simulate a 4-bit comparator using your parameterized 2-bit comparator to verify it behaves correctly. *Simulate your design to verify that it behaves correctly. You do not need to simulate every possible combination of inputs. You only need enough cases to ensure all operations of your circuit work properly, so set up your simulation wisely*.

Problem 2 - Seven Segment Decoder

Create a behavioral SystemVerilog module to implement a binary to single digit hex seven-segment decoder circuit for the 7 segment display on the Basys3. By behavioral, I mean that instead of using K-maps and other logic reduction techniques to determine gate level equations for each output, describe the circuit using case, if, and else statements in an always_comb block.
Your decoder should display the decimal digits 0-9 for bit patterns 0000-1001, and A-F for bit patterns 1010-1111 (you will need to get a little bit creative to show all the hex digits – you can use lower case letters as well).

Refer to chapter 8 “Basic I/O” of the “*Basys3 Reference Manual*” posted in Canvas for information on how the number display works. Section 1 “Seven Segment Display” is of particular interest.

After you have verified the circuit through simulation, download it to the Basys3 board using four slide switches as inputs (B_3 B_2 B_1 B_0) to enter the binary number. Only display the number on the rightmost digits of the 7-segment display.

B_3: Switch 3 B_2: Switch 2 B_1: Switch 1 B_0: Switch 0

You do not need to demo problem 2. Program the FPGA and test the display before moving on to part 3.

Problem 3 – Big or small

Create a top-level design that uses your modules from problem 1 and problem 2. You may also use your comparator or other modules from previous labs. The design takes in two 4 bit numbers on switches 7:4 and 3:0, along with 3 modes set by the left most switches 15:13.

Display the number that matches the condition on the 1st digit of the 7-Segment display. The modes set the condition as shown in the table below. Turn the display off if no switches are set, or if there is more than one switch set.

Inputs

A_3: SW3	B_3: SW7	GT: SW15
A_2: SW2	B_2: SW6	LT: SW14
A_1: SW1	B_1: SW5	EQ: SW13
A_0: SW0	B_0: SW4	

Start by creating a modular block diagram for this problem that shows the connections between all the modules in the design and the widths of each signal. A modular block diagram helps in a similar fashion to pseudo-code, flowcharts, or UML diagrams (you don't need to know what these are) in software design planning.

You do not need to simulate every possible combination of inputs. You only need enough cases to ensure all operations of your circuit work properly.

Provide a demonstration of the working design to the instructor.