Lab 19: Building an 8-bit Counter with 4-Digit Hexadecimal Readout which stops at a user selected value.

In this lab you will build upon your previous experience with the Digilab board.

Submission of the lab consists of:

1) (0 pts) a cover sheet with you name and section,
2) (110 pts total) a print out of the 2 schematic diagrams (from the schematic editor),
   35 pts part A
   35 pts part B
   40 pts part C
3) (30 pts total) a print of the pin assignment file (*.ucf) [click on "edit constraints (text)" which will bring up the *.ucf file in a text editor from which it can be printed]
   10 pts part A
   10 pts part B
   10 pts part C
4) (10 pts total) Answer to the two questions in part A.
5) Demonstrate the working project (part C only) on the Digilab board to your instructor.

Failure to demonstrate your project will cause your grade for the above to be discounted by 50%. A circuit that does not meet the required specifications will result in your lab being discounted by a minimum of 10%.

Your instructor has a working model you can inspect.

New in this lab are:

- counters

Use the LF_clock_source to drive the counter with the 10 Hz output. The counter can be slowed more by selecting 25 MHz or 50 MHz for that each value displayed by the counter can be observed.

Part A) Build the 8-bit binary counter using toggle flip-flops. Create a new source schematic in your project "eight_bit_counter". Chain eight toggle flip-flops (TFF) together with the output of one serving as the clock input to the next. Hint: use the 'ftc' FF in the 'flip flop' symbols library of the schematic capture CAD utility. Connect the Toggle inputs together and assign it to a terminal called "enable". The "ftc" FF also has a clear input. Connect the clear inputs together and assign it to an input terminal "clear". Connect the clock input of the first TFF in the chain to an input terminal labeled "clk". The outputs of each TFF form one binary digit of the output. Connect each of these to an output terminal. Because the first TFF in the chain toggles the most rapidly it is the LSB. The last TFF in the chain toggles the least rapidly hence is the MSB. Label these outputs in order from the LSB to the MSB, b0, b1 b2, …b7, respectively. (After initial testing in this part, you will create a schematic symbol for "eight_bit_counter" and use it in part B.)

Now test the counter. Assign the "clk" input terminal to the LF_clock_source 10 Hz output. Assign the enable input to one of the slide switches (for example, SW0). Assign b7, b6, b5, … b0 to LD7 through LD0 respectively. Test the counter on the Digilab board.

As described, does this make an up counter or a down counter?

How could you change an up counter to a down counter and vice versa?

You will need an up counter. Make it so.

Once it works as it should, create the schematic symbol.

Part B) Build the 8-bit up counter with hexadecimal readout using the right two digits of the 7-segement LED and with reset (zero), start, and have it stop when it gets to FF. Create a new source schematic in your project "counter_LED". You will use the three modules "eight_bit_counter", LF_clock_source, and the 4-digit 7-segment LED decoder you made in a previous lab. Follow the directions in previous labs on how to use modules. Use BTN0 for start and BTN3 for reset.

Part C) Build the 8-bit up counter with hexadecimal readout using the right two digits of the 7-segement LED to display the output of the counter and with reset, start, and have it stop when it gets to a user input value. The stop value is entered by the switches (SW0–SW7) and displays as a 2-digit hexadecimal value on the left two digits. Create a new source schematic in your project "counter_LED2". You will three modules "eight_bit_counter", LF_clock_source, and the 4-digit 7-segment LED decoder you made in a previous lab. Follow the directions in previous labs on how to use modules.

Here are the specifications your circuit (part C) must meet.

- The counter will read out in hexadecimal using the right 2-digits of the 7-segment display. The 2-digits should be in the correct order to form a two digit hexadecimal number, the least significant bits (b0-b3) on the right most digit and the most significant bits (b4-b7) on the second digit
- The counter is an up counter.
- The counter will ONLY start when BTN0 is pressed and then continue to run until it reaches the stop value.
- When your circuit is initially powered, the counter will be stopped.
- The counter will count at approximately 2.5 Hz.
- Every value in the sequence 00 through FF will be displayed (visible), when the stop value = FF.
- The value entered by the switches (SW0-SW7, SW0 is lsb, SW7 is msb) is displayed on the left 2-digits as a hexadecimal number.
- The counter shall stop when it reaches the hex value displayed on the left two digits. When the counter stops the left two 2-digit hexadecimal numbers displayed will be equal.
- Pressing BTN3 will reset the counter to zero, but will NOT cause the counter to start.

Hints on controlling the start/stop behavior of the clock: The *enable input* controls whether the clock is running or is stopped. **Part B:** Find the logic which will stop the clock at zero when BTN0 is not pressed (0); will start if BTN0 is pressed (1) and the clock is zero (part B) or the user defined value (part C); and will also run if the clock is not zero (part B) or the user defined value (part C). **Part C:** you will need to modify your circuit in part B using a flip-flop to get the start/stop behavior you need.