Lab 3 -COVID

CST 231

Objectives

1. Reduce the amount of verbatim instructions given in lab. Ask questions as needed.
2. Setup a multiplexed 7-segment display to display numbers 0-9
3. Write a basic counter
4. Create clock dividers and understand math behind it.
5. Control multiplexing to show two different numbers on the 7-segment display at the same time.
6. Draw a system block diagram
7. Extend design to 4-digit 7-segment

Note: All questions that I require a response for are highlighted in magenta.

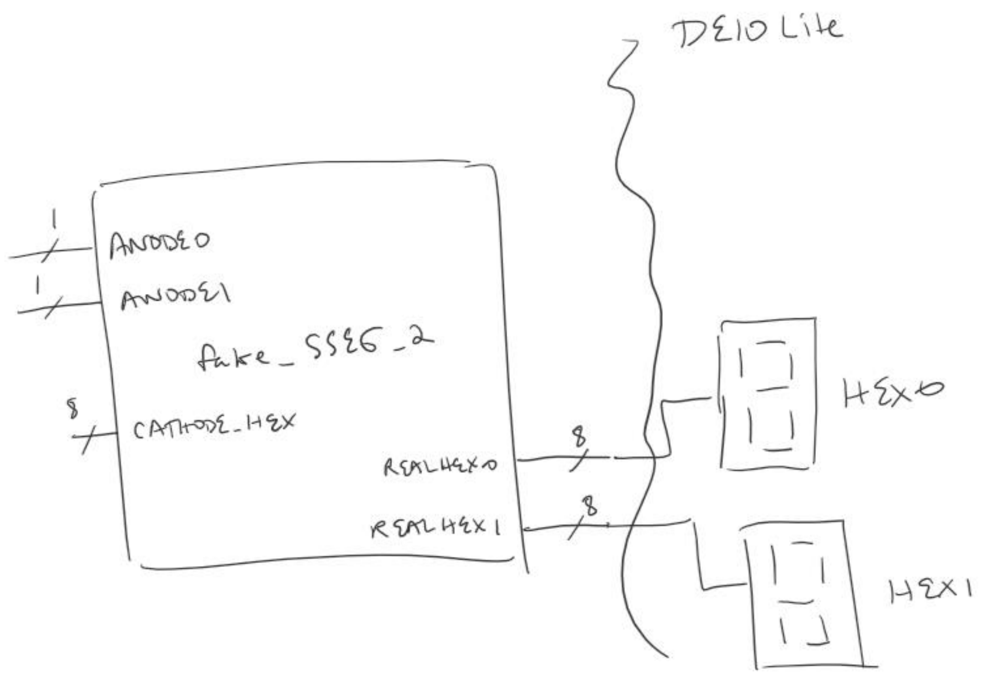
1. Please review the datasheet for the DE10-Lite to see how each single digit hexadecimal display is used.
    a. What is the page number of the information?
    b. Is it configured as common cathode or anode?
    c. Is it active high or low?

2. Initial block diagram. Given that the end goal is to design a system to automatically count up from 0-99 given a 50 MHz input clock and 2-digit 7-segment display, please draw out the block diagram of modules that you require.

3. Copy over a version of the DE10-Lite template file, and add my fake_SSEG_2.v file to the project. The fake_SSEG_2 will mimic the behaviour of a real 2-digit multiplexed 7-segment display.
   a. Instantiate a copy of the fake_SSEG_2 in the top level file.
   b. Wire the fake anodes to individual switches.
   c. Wire the outputs (REAL_HEX*) to the top level file's HEX0 and HEX1.
   d. For CATHODE_HEX, please assign it a real number.

      ```
      .CATHODE_HEX0(8'b11110000)
      ```

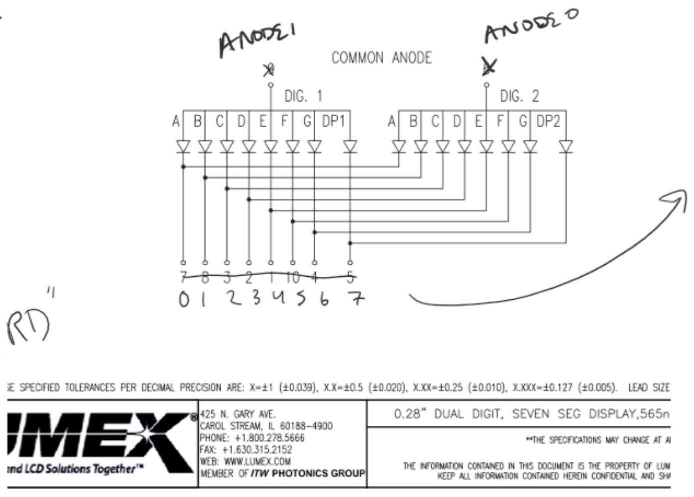
   e. Make sure that you can can display the same number at the same time using only switches.



To illuminate a digit on display 0, set anode0 to logic 1.

To make display 0 show the number 1, set HEX[7:0] = 8'b11111001;

fake_SSEG_2 "equivalent" internal circuit



SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE

LUMEX
and LCD Solutions Together™

425 N. GARY AVE.
CAROL STREAM, IL 60188-4900
PHONE: +1.800.278.5666
FAX: +1.630.315.2152
WEB: WWW.LUMEX.COM
MEMBER OF ITW PHOTONICS GROUP

0.28" DUAL DIGIT, SEVEN SEG DISPLAY,565n

**THE SPECIFICATIONS MAY CHANGE AT A

THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF LUM
KEEP ALL INFORMATION CONTAINED HEREIN CONFIDENTIAL AND SH

"LDD-2802 RD"

CATHODE_HEX[0] =A
CATHODE_HEX[1] =B
CATHODE_HEX[2] =C
CATHODE_HEX[3] =D
CATHODE_HEX[4] =E
CATHODE_HEX[5] =F
CATHODE_HEX[6] =G
CATHODE_HEX[7] =DP

4. Implement or recycle your binary to 7-segment decoder from CST 133.
    a. Make sure that given 4-bit input values of 0 to 9, you are able to get it to show 0 to 9 on the 7-segment display. Tie the ANODE0 and ANODE1 both to logic 1 for this part.
    b. Make sure that you can show 0 to 9 on both 7-segment displays at the same time using 4-bit input (switches) and just holding ANODE0 and ANODE1 to 1.
5. Now we need to implement a module that will allow us to show two different values on the 7-segment display at the same time.
    a. Both displays can not be driven at once because they share the same fake cathodes. Use persistence of vision and drive each display one at a time. Review lecture notes.
    b. You'll need a clock divider for this part. Note that your system clock is 48 MHz, 50 MHz, or 100 MHz, so you will most likely need to divide the clock down to the KHz range (Revisit the multiplexing slides for more information, or just ask me.)
        i. Please show your math for this.
        ii. What is your desired output frequency and duty cycle? Why?
        iii. What is your input clock frequency?
        iv. What is the ratio, and what is your count up value? Please justify this.
        v. What is the clock period of your input and output clocks?
    c. Recycle components from previous parts.
    d. Implement the module using the pseudo-code located on the next page, or your class notes.

Pseudo-code (sseg is essentially HEX0 and the anodes are actually ANODE_0 and ANODE_1 in our design.

A. Counter reaches display 1 turn.
B. Drive display 1 by setting anodes = 2'b10;
C. Output desired value to the 7-segment pins. eg. sseg = 7'b01010011;
D. Counter reaches display 2 turn.
E. Drive display 2 by setting anodes = 2'b01;
F. Output desired value to the 7-segment pins. eg. sseg = 7'b01010011;
G. Keep switching between display 1 and display 2 when the counter reaches display 1/2 turn.

6. After done with this, show that you can display two different numbers on the 2-digit display at once.

7. Implement a counter module that can count from 0 to 99 on the 7-segment display. Have one from a previous class? Just recycle or expand on it.
   a. Consider testing your modules individually. For this counter- maybe don't hook it into the 7-segment right away. Just pop the outputs to LEDs.
   b. You can use a 7-bit register to count from 0 to 99. You will also most likely need the outputs to not show the binary value 99, but rather 99 on the 7-segment. Your code should account for this either in the counter itself, or an additional module. This was covered in CST 133, feel free to use your notes (Did you even take notes?)

8. If you were to use the 50 MHz clock or the multiplex clock for your counter module above, you would not be able to actually see anything. Please implement a clock divider specifically for the counter module above and write the specifications below.
   i. Please show your math for this.
   ii. What is your desired output frequency and duty cycle? Why?
   iii. What is your input clock frequency?
   iv. What is the ratio, and what is your count up value? Please justify this.
   v. What is the clock period of your input and output clocks?

9. Now hook up the clock divider to your counter module. You don't have to, but consider testing this part of the system in isolation by popping the output of your counter (driven by your slower clock divider) to LEDs. Doing this is helpful to pinpoint which modules might be at fault…
10. After all individual modules are complete, you are ready to wire the system together. Wire the system together and show instructor the following functionality.
    a. You must show the instructor the code and device programmed during demo.
    b. The demo must show that using the fake 2-digit 7-segment, clock dividers, and other modules you have created you are able to count 0-99 glitch free and roll over back to 0 when 99 is reached (no pausing or glitching).
11. After you are done with the above, please draw out a block diagram of your system by hand or using a computer program.
12. Please go to Tools > Netlist Viewer > RTL Viewer. Take a screenshot of the expanded design (Ask me if you are not sure what to take screenshot of).
13. Please ask a colleague for their design and compare your RTL viewer. See any differences in how they implemented their design?

14. Last part- take my fake_SSEG_2 module. Copy it to a new file and expand it to 4 displays. Show me four displays counting. (Note- Do not change your counter. Just show that the two new displays can duplicate content of old displays

    a. What is the new clock frequency you used to multiplex this four display module? Why is this most likely higher than the previous clock frequency?
    b. What is the bare minimum clock frequency to maintain persistence of vision on four displays assuming 60 Hz is the persistence of vision threshold. Please show calculations relative to clock period. (Winter 2021- I sent an email with my calculations with example frequencies).