Storing and Moving Text

PURPOSE - In this lab, you will design a circuit capable of storing short messages and displaying them as a scrolling marquee on the four 7-segment LEDs.

----

1. Writing the Verilog description

You will have to design a circuit that can store and display an arbitrary message entered character by character by the user. The characters will be entered using the 8 DIP. You may assume that the total length of the entered message will be no longer than 20 characters for simplicity. To remain consistent, you must use the following mappings for the DIP switches:

SW7 - segA, SW6 - segB, SW5 - segC, SW4 - segD, SW3 - segE, SW2 - segF, SW1 – segG, SW0 – segDP

You will use any three of the remaining switches to determine the state of your machine. All possible states of the machine are given below:

BNT2	BNT1	BNT0	Function
0	0	0	Displaying and moving text to the left.
1	1	0	Users can modify the DIP switches. The pattern is displayed in real-time on the rightmost 7-segment LED, while 'L' is displayed on the leftmost 7-segment LED.
1	1	1	The last character is stored and displayed on the rightmost 7-segment LED, while 'o' is displayed on the leftmost 7-segment LED.
0	1	0	RESET – All previous stored characters are cleared (set the index for stored characters back to 0).
All other possibilities			Same as '000.'

In other words, when (BTN2, BTN1, BTN0) = (110), the user modifies the switches to the letter they want to be loaded and stored. Then, when (BTN2, BTN1) = (11) and BTN0 goes from 0 to 1, the letter is loaded and stored into the proper register. To load and store the next letter, the user only needs to set BTN0 = 0 again, modify the switches, and switch the BTN0 from 0 to 1. These actions occur as long as the user wants to introduce letters or the blank character (the blank character is simply an "empty" character when the 7-segment LED displays nothing). Verify the functionality of the FSM in a testbench. Note that you can plot internal signals of the design to show the cycling text action.

IMPORTANT NOTE: FOR THE FIRST WEEK YOU SHOULD COMPLETE UP TO AT LEAST THE POINT WHERE YOU CAN STORE AND DISPLAY A SINGLE STATIC CHARACTER ENTERED BY THE USER

When the user sets (BTN2, BTN1, BTN0) = (000), the stored text is displayed using the four 7- segment digits in a "moving to the left" fashion. For example, if the input message is "EE4301 CLASS," then the four 7-segment LEDs will display "EE43" during the first second. During the next second, they will display "E430". During the third second, they will display "4301", and so forth until the entire stored message is displayed. **This is repeated indefinitely with one blank between any two repetitions** until the resetting combination is detected. After resetting, the user will have to introduce the wanted text again in a known manner. **Don't forget to check the case where you try to display moving text without any previous input** (in this case, you can display nothing during the 000 state). Also, **test the case for displaying only a single input.** In this case, you should only display one static character. You should also keep track of how many characters have been entered in total only to display those characters **and then repeat without having several blanks at the end** if the message is shorter than 20 characters.

2. Verification

Synthesize and implement the circuit. You will have to write an .xdc file to have correct pin assignments. Download the bit-stream file to the board and try different short, reasonable texts. Show your design to your TA for full credit.

SUMMARY -- In this lab, you designed and verified the "moving text" circuit.

--- \

Lab Notebook deliverables:

- Brief description of what you did in the lab, what steps were followed
- Discussion of results:
    a. What were the test cases you used and discuss why you chose the set of test cases?
    Explain how your test cases verify each of the FSM scenarios.
    b. Attach a snippet of the reports of the tool.
        i. State transition diagrams and state table. (Refer to the description in the notes section)
        ii. Table and Simulation waveforms (**behavioral only**) showing all the test cases used above. All test cases must be visible in the waveforms; make sure the FSM state is visible in the waveform.
        iii. Schematic screenshot of synthesized design.
        iv. Utilization report: synthesis and implementation.
        v. Timing report.
        vi. Power report

    NOTE: Only attach relevant sections, not the entire report.
- Summary or Conclusion:
    a. Were all design goals met?
    b. What were the difficulties encountered?
    c. Any improvement you would like to make time permitting? Etc.…
- Verilog Source code:
(Font: Courier, size 8pt, single line spacing, no paragraph spacing)
- All files must be labeled (ex. Figure 1: …). You should also highlight more interesting parts of your attachments.