

# ECE 2700 Lab 6

**Due the week of March 23, 2015 at the beginning of your registered lab session (200 points)**

## Objective

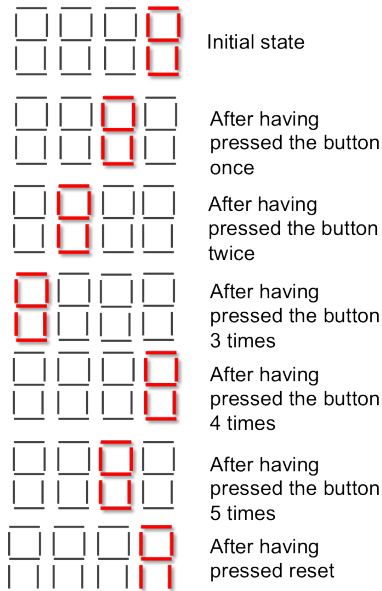
To learn how to build complex sequential circuits in Xilinx ISE and load them on the Basys board.

## 1 Preparation

1. Read this document in its entirety.
2. Draw an FSM for the moving 7-segment display system.
3. Draw FSM(s) for the sophisticated 7-segment display system.
4. NOTE: If you do not have a working 7-segment display design from Lab 4, ask your TA for a copy.

## 2 Moving 7-Segment Display

In this part of the lab, you are to modify the seven-segment display design from Lab 4 so that when a button is pressed, the next digit is displayed. The following figure provides some illustrations. For example, if the board is currently displaying the rightmost digit (whatever the value shown may be), pressing a button (BTN0) will cause the second rightmost digit to light up instead (with the same value shown). Pressing a button when the leftmost digit is displayed will cause the rightmost digit to light up.



Note that we want the system to remember its state so you need to implement this part of the lab as a sequential circuit (otherwise, releasing the button will cause the rightmost digit to be displayed again

regardless of the current state of the system). Use your FSM to guide you. Recall that a sequential circuit needs a clock and a reset. Modify your `SevenSeg.v` file appropriately using the following declaration: `SevenSeg(clk0, rst, button, w, x, y, z, a, b, c, d, e, f, g, n3, n2, n1, n0)`. When the reset button *rst* is pressed, the digit that will be displayed is the rightmost digit (we assume the rightmost digit represents the initial state of the FSM). All operations are synchronous.

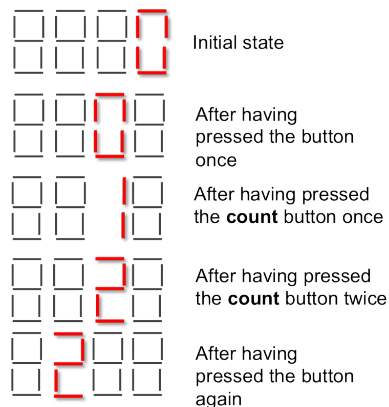
Test your design using simulations before moving on. Once you are convinced your design is correct, switch to implementation mode and create an appropriate `.ucf` file. Select `CLK1` as your clock input, `BTN0` as your *button* input, and `BTN1` as your *rst* input. Before you generate a bit file make sure that the following considerations are taken care of.

First, the Basys board gives you three options for clock frequency: 25Mhz, 50Mhz, and 100Mhz. With no blue jumper, the clock frequency is 50Mhz. If the blue jumper is on the low position, the clock frequency is 25Mhz. Otherwise, it is 100Mhz. For this lab, we will use 50Mhz so there is no jumper needed.

Second, the board clock frequency is way too fast for the human eye. For this reason, you will need to use a clock divider to slow down the frequency (refer to p. 219 of your main textbook for a discussion on this topic). A clock divider (`clockdivider.v`) is provided for your use in Canvas. Download it and use the output of the clock divider as the input *clk* to your FSM. With the clock divider, the output clock now ticks once every second. This means that you may need to hold your button for a little while before changes occur. Don't forget to test your reset button. Show the results to the TA.

### 3 Sophisticated 7-Segment Display

Modify your moving 7-segment display design to increase the displayed value by one every time a count button is pressed. An example is given below. The initial value should be set to 0. If the currently displayed value is F, the next value should be 0. Use `BTN2` as the count button. It may be easier to use two FSMs instead of one to implement this design.



**Challenge (Optional):** Instead of using the count button to increment the displayed value, design your system so that every time *button* is pressed, the display moves **and** increases the displayed value by one.

### 4 TA Checkoff

- (20 points) Prelab.
- (90 points) Moving 7-segment display.
- (90 points) Sophisticated 7-segment display (either the standard or more challenging version).