Scrolling Message on 7-Segment Displays

Grant Mckee, Jason Wegryn, Zachary Kilmer

Project Overview

- Write and implement VHDL code to create the scrolling message “ECE 2700” across multiple seven segment displays on a NEXYS Board.

- Allow the user to adjust the speed of scrolling, orientation of message, and scrolling direction using a series of switches.

Real World Applications

- Alarm clocks
- Kitchen appliances
- Gas Pumps
- Measurement instruments

Preventative Measures

Our project includes a number of safety measures in order to prevent errors and glitches. Included in the controls are a reset button that completely clears the system and sets it back to its original state. There is also an enable switch to stop each of the components from being activated. Finally to avoid error, counters are used to divide and set clock times for each of the components, as well as all of the components being tied into the main clock so that there is no chance of them being out of sync.

Basic Components

Counters- Counters were used as the heart of the code to adjust the timing of each component, one of four counters are used to control the shifting speed of the letters, while the fifth counter is used as the enable for our FSM

Multiplexers- Multiplexers are used in the selection of the speed and direction of the scroll there is also a multiplexer tied to the state machine to coordinate the correct letter with the activated seven segment display.

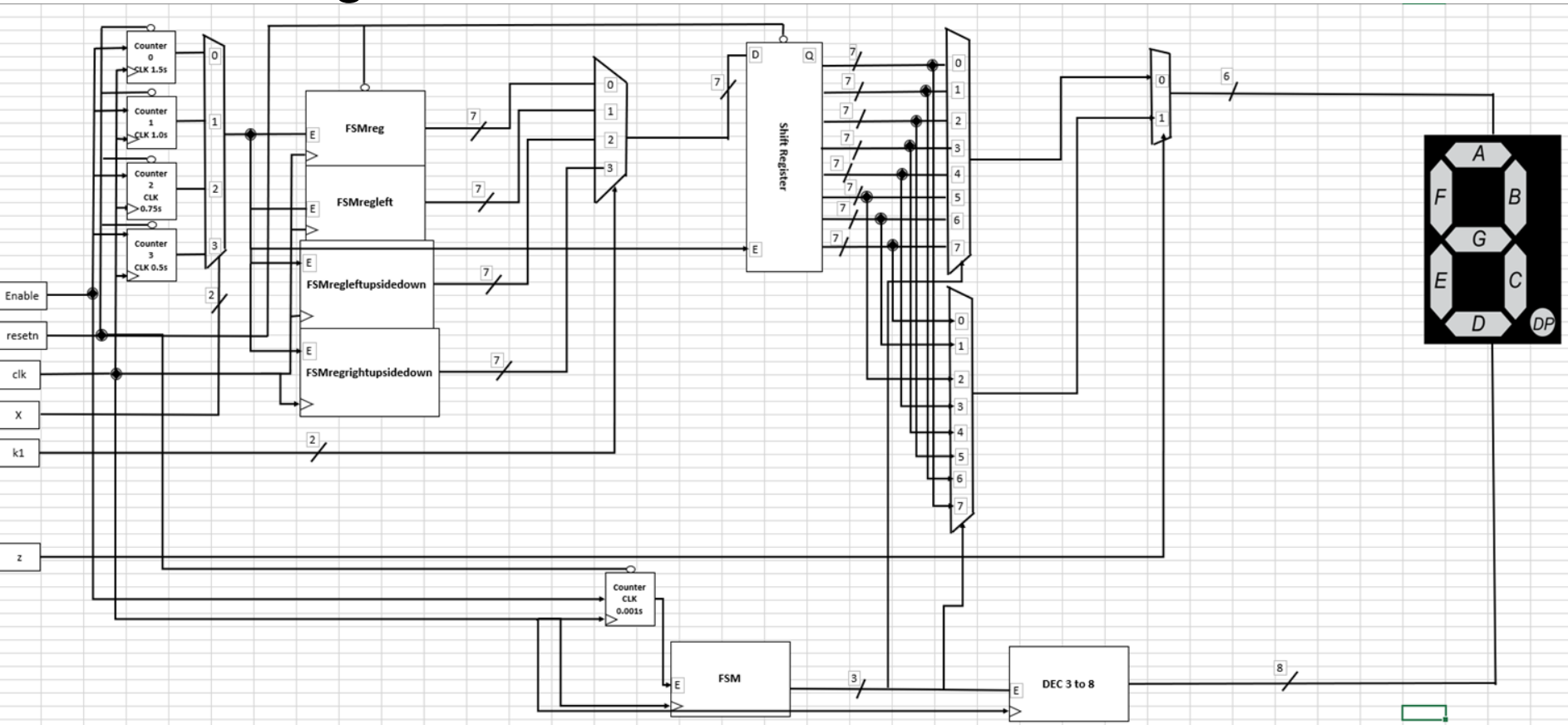
Registers- Registers are set up in a pattern and synced up to the same clock so that on each tick each letter is shifted left or right, and will stay there until the seven segment display has shown each letter of the message. The speed at which these registers shift, is the speed that the message scrolls and is controlled by the 4 counters mentioned before.

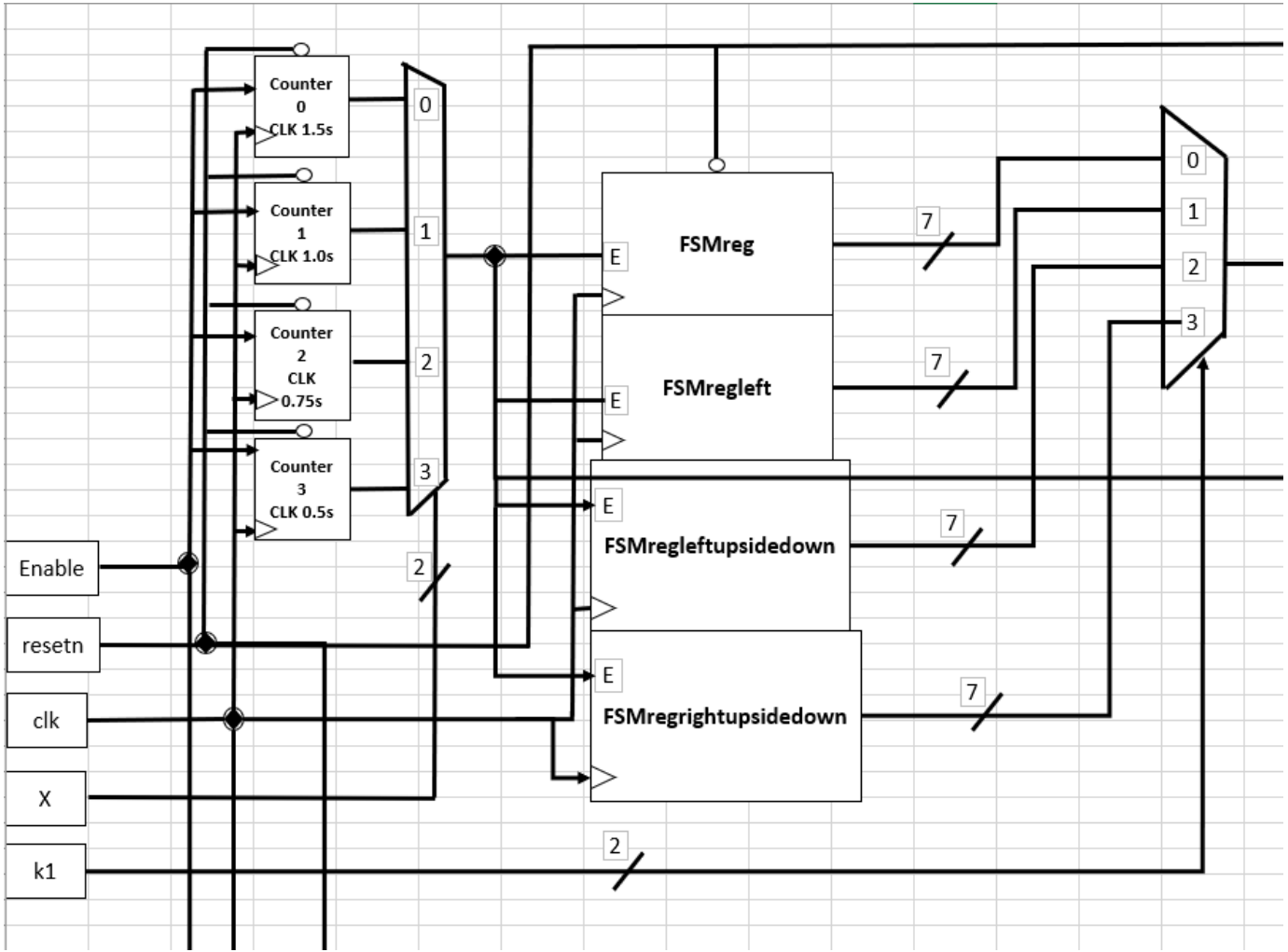
Basic Components Cont.

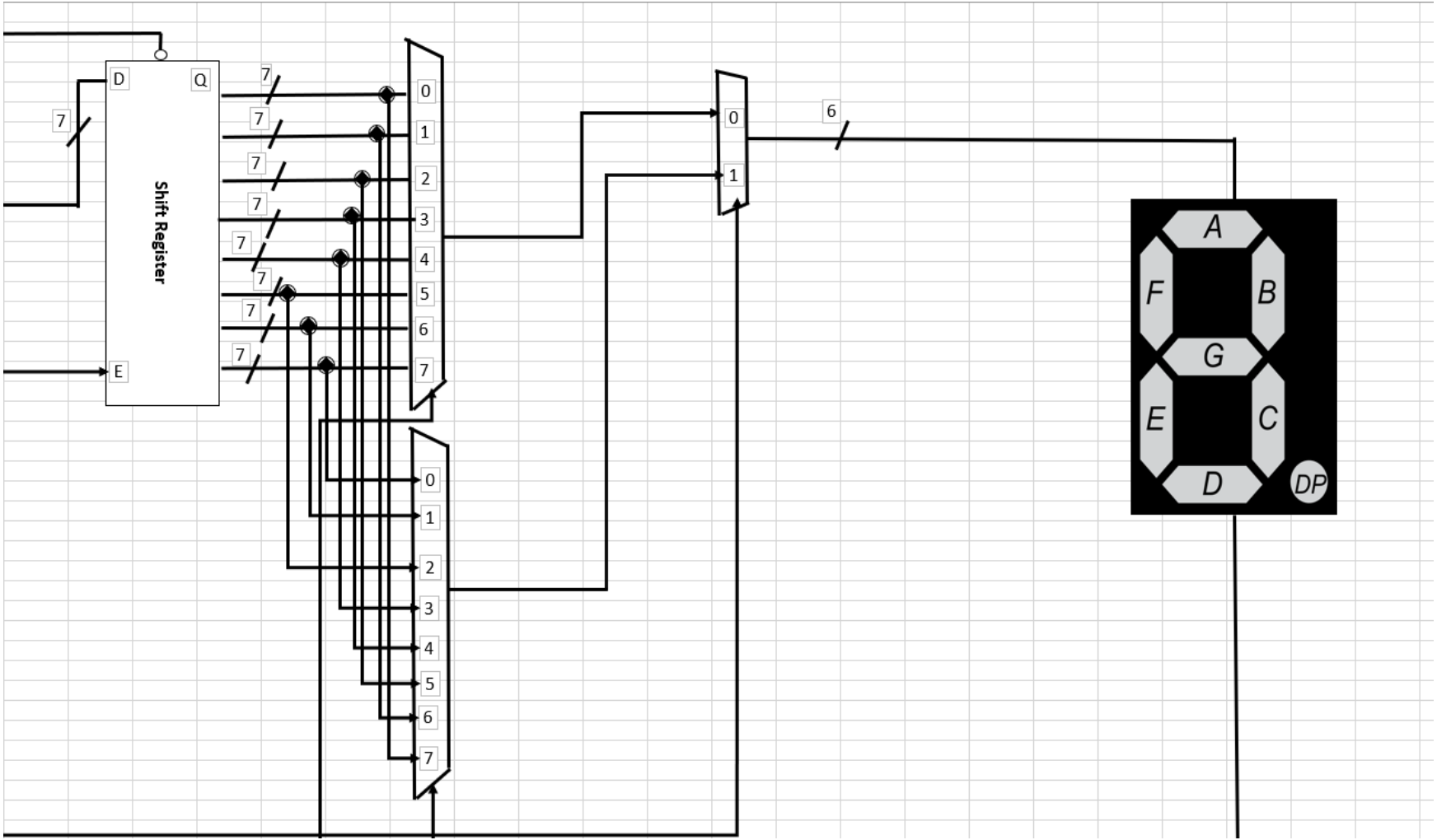
Decoder- There is a decoder tied to one of the final state machines, this decoder decides which of the seven segment displays is lit up in each of the eight states.

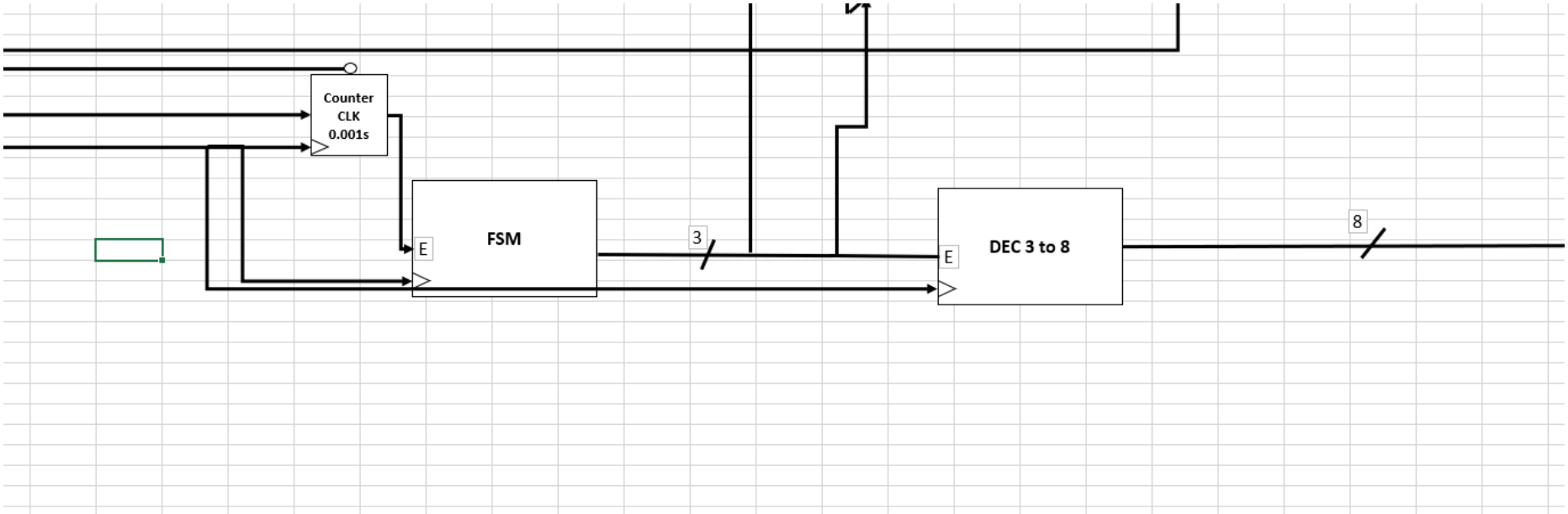
Final State Machines- There are 5 FSM in the design of the project. Four of these FSM's will control the message that is being sent out depending which one is picked by the MUX. The fifth FSM will control the select bits for the MUX 8-to-1 as well as give 3 bits to the decoder to tell the system which 7-segment displays need to be on.

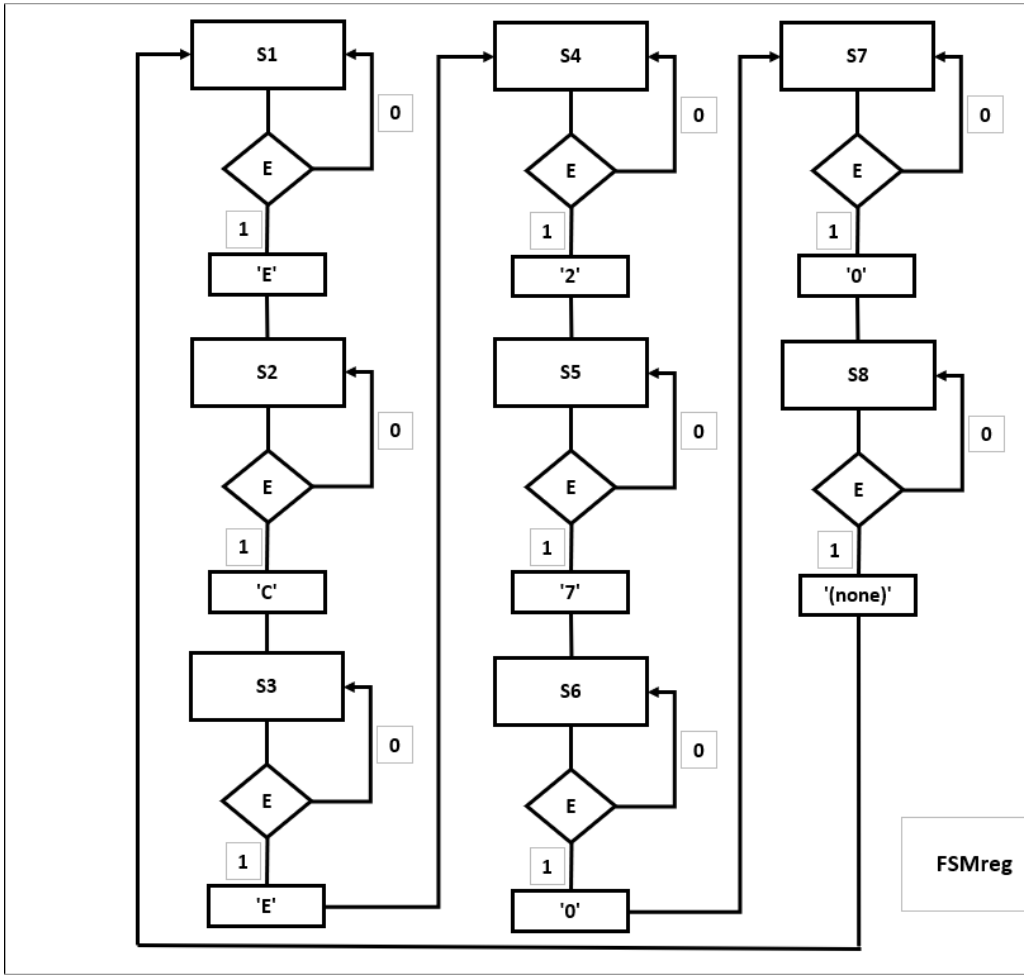
Block Diagram

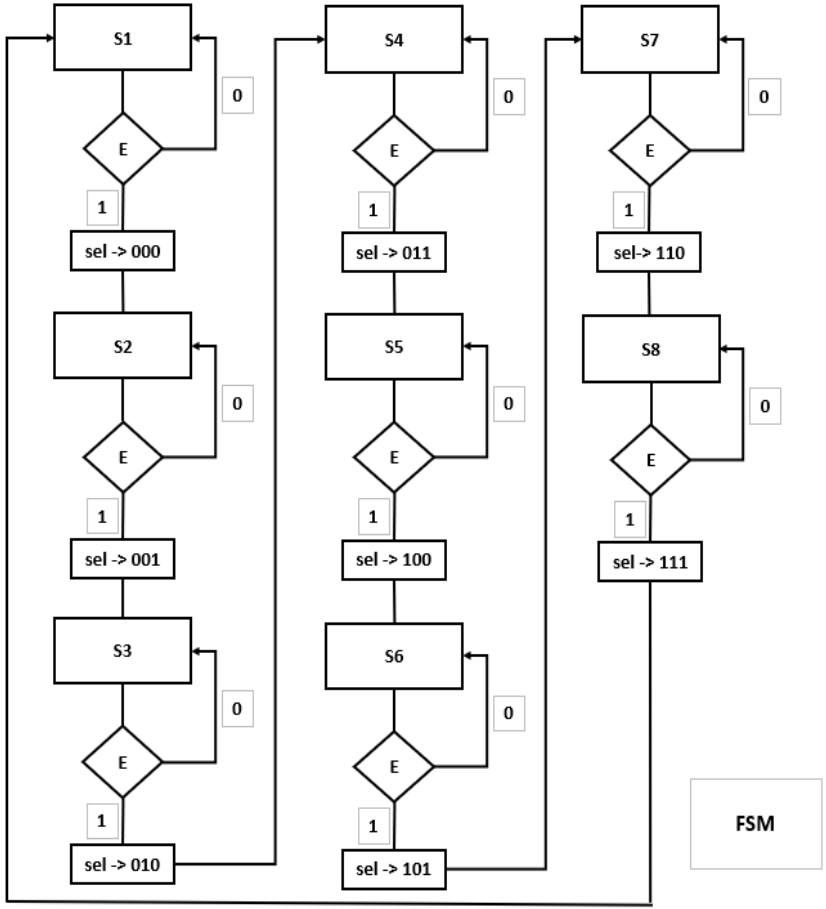


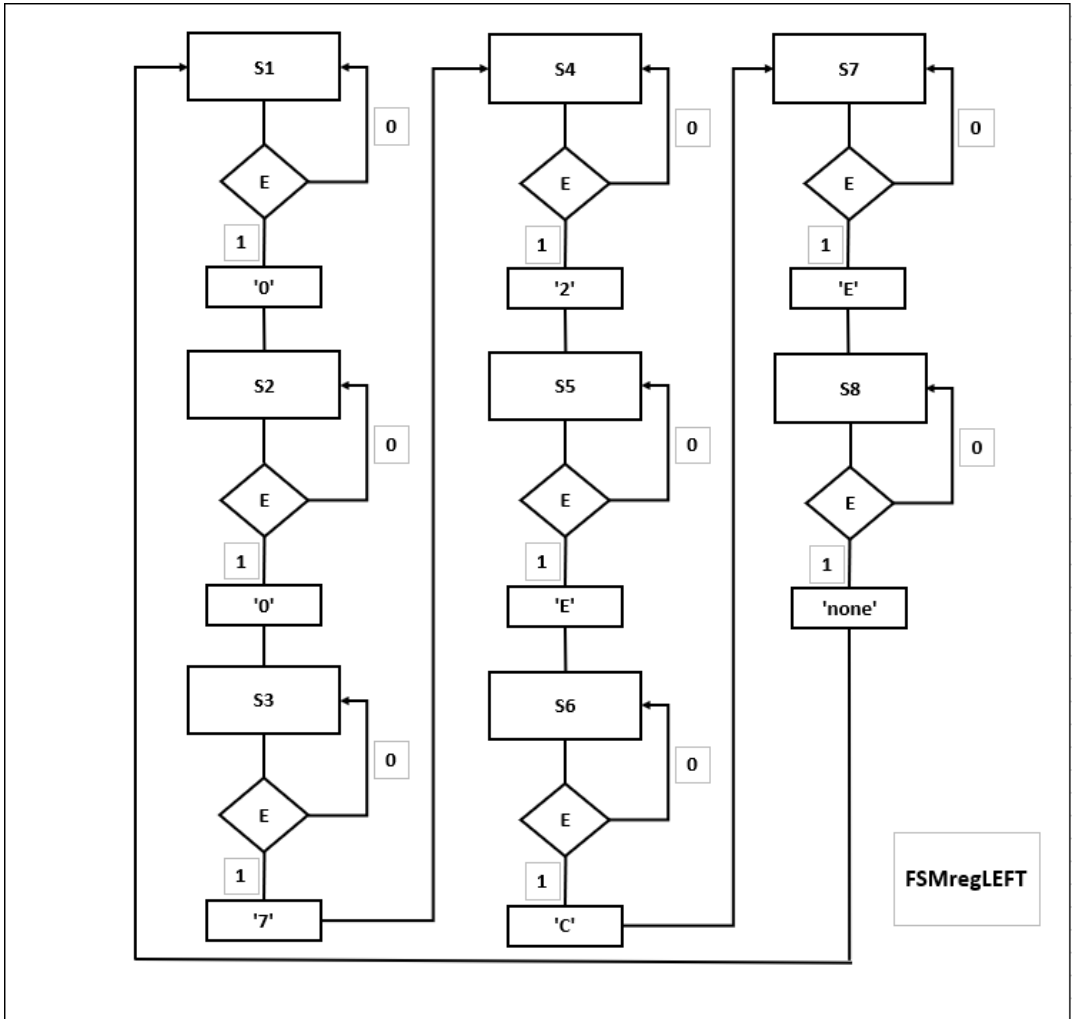

Counter 0 CLK 1.5s
Counter 1 CLK 1.0s
Counter 2 CLK 0.75s
Counter 3 CLK 0.5s
0
1
2
3
Enable
resetn
clk
X
k1
2
E
FSMreg
FSMregleft
FSMregleftupsidedown
FSMregrightupsidedown
7
0
1
2
3


D
Q
E
Shift Register
7
0
1
2
3
4
5
6
7
6
A
B
C
D
E
F
G
DP


Counter
CLK
0.001s
E
FSM
3
E
DEC 3 to 8
8


S1
0
E
1
'E'
S2
0
E
1
'C'
S3
0
E
1
'E'
S4
0
E
1
'2'
S5
0
E
1
'7'
S6
0
E
1
'0'
S7
0
E
1
'0'
S8
0
E
1
'(none)'
FSMreg


S1
0
E
1
sel -> 000
S2
0
E
1
sel -> 001
S3
0
E
1
sel -> 010
S4
0
E
1
sel -> 011
S5
0
E
1
sel -> 100
S6
0
E
1
sel -> 101
S7
0
E
1
sel-> 110
S8
0
E
1
sel -> 111
FSM


S1
0
E
1
'0'
S2
0
E
1
'0'
S3
0
E
1
'7'
S4
0
E
1
'2'
S5
0
E
1
'E'
S6
0
E
1
'C'
S7
0
E
1
'E'
S8
0
E
1
'none'
FSMregLEFT

Video of FPGA board

Problems We Ran Into and How We Solved Them

- Scrolling the Message
- Changing the Orientation
- Changing the Speed

What Could Be Improved?

- Code could be simplified
- More displays
  - Longer messages
- Use liquid crystal display in order to use all possible letters
  - Increases possible messages
- Use ASCII table for more possible characters

THIS IS OUR ECE 2700 FINAL PROJECT

ANY FINAL QUESTION?