Stock Market Ticker On A Seven Segment Display Status Report

List of Authors (Calvin Mires, Lukas Pape, Reese Preston, Justin Klotz)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: calvinmires@oakland.edu, lukaspape@oakland.edu, reesepreston@oakland.edu
jjklotz@oakland.edu

Abstract - The purpose of this project was to implement a scrolling banner using the eight seven segment displays on the Nexys A7-50T board. This project is designed to display and scroll through four different stock tickers (i.e Apple, Pepsi, Tesla, and Bank of America) and their prices. Which stock is displayed is decided via the switch positions controlled by the user. The entirety of this project can be implemented using concepts and tutorials learned or referenced in the ECE 2700 course. This includes knowledge of components such as multiplexers, decoders, counters, finite state machines (FSM), and shift registers. This project acts as proof that complex systems can be broken down into relatively simple components.

I. Introduction

Our project consisted of creating a banner that displays 4 different company stock prices. The 4 available stock tickers and prices to the user are “APPL 147”, “TSLA 751”,”PEP 179”, AND ”BAC 36”. Displaying these messages is accomplished by utilizing the 8, 7 segment display on an Nexys A7-50T board. Not only will the prices be displayed but they will follow a particular pattern. First, the message scrolls the stock ticker and price all the way to the left and once all of the characters are displayed on screen, it will repeat the pattern until it is reset or a different stock ticker is switched on. The inspiration for this project came from replicating a serializer digital system explained thoroughly in the ECE 2700 course. The serializer implemented a 4-1 multiplexer, 1ms counter, FSM, hex to 7 segment decoder, 2 - 8 decoder, and a 4, 7 segment display. This system gave the user the option to choose between 4 letters (i.e “A”,”B”,”C”, and “D”) and would flash it on the seven segment display. Although the flash is untraceable to the human eye as the flash is occurring very fast every 1ms. With the addition of a 1s counter, 8 - 1 multiplexer, and a 5 bit shift register, the scrolling stock banner was able to be created with the same principle design and functionality as the serializer circuit.

II. Methodology

1. *Counters*

Two counters were utilized in this design. One counter will feed into the 5-bit parallel access shift register at a speed of 1s. The second counter will feed into the FSM at a speed of 1ms. The input of both counters was a standard 100MHZ with a 50% duty cycle clock. The outputs of the counters were the signal’s ‘Z’. The 1s counter designated for the shift register shifts each letter, number, or space down every second as it is hooked up to the clock port of the register. The 1ms counter is designated for the FSM which is hooked up to the enable port. This allows for a state change every 1ms.

2. *Binary to 7 Segment Decoder*

This converts the 5 bit signal from the 8-1 multiplexer into actual numbers and letters to be outputted to the 7 segment displays. This can be accomplished by implementing a directory of all possible numbers, letters, and spaces used for each stock ticker and price corresponding to the correct 7 segment configuration. For example, when a 5-bit input value is equal to "00000”, the binary to seven segment converter will output "1111110” which will show “0” on one of the seven segment displays.

3. *5-bit Parallel Access Shift Register*

The input to the register is the output array of the 4 - 8 multiplexer, which is the data for the entirety of one of the 4 messages. Each letter or number is represented by a 5-bit vector and will have an initial position where the full

message is visible in the correct order on the displays. For each clock rise, each vector will shift one vector to the left (meaning that each character will shift one to the left). The output of the register will be the 8 individual vectors of 5-bits each. For example, say at 0 seconds, the output vector sequence is "AAPL 147" (after being decoded from 5-bit vectors), at the next second, the sequence would be "7AAPL 14".

4. *8-to-1 Multiplexer*

The inputs to the 8-to-1 multiplexer are the characters being outputted by the 5-bit shift register. The select for this multiplexer is controlled by the output of the FSM and sends whichever character that corresponds to the display being enabled by the 3-to-8 decoder output.

5. *4-to-8 Multiplexer*

The beginning of this entire circuit consists of a 4-8 multiplexer which is operated by two switches (4 arrays to 1 array that is 8 vectors). The cases for the multiplexer output are as follows: s_l = "00"-> APPLE, s_l = "01"-> TSLA, s_l = "10"-> PEP, s_l = "11"-> BAC. Once the ticker symbol is selected, the multiplexer will output an array of length 8 where each value of the array holds a vector of 5 bits that represents a character. This array will then be inputted into the 5 bit shift register

6. *3-to-8 Decoder*

The decoder is used to select which display is turned on or off. The decoder acts synchronously with the 8-to-1 multiplexer and converts the output from the FSM into an 8 bit output that is used as the enable for each of the 7 segment displays. Only one display is enabled at a time. Even though only 1 display is on at any given time, it still appears that all of the displays are turned on because this process is happening too quickly for the human brain to process the change (every 10ms).

7. *FSM*

The FSM is the "brain" or logic behind this circuit. The FSM monitors the state of the switches used to toggle between different stock tickers. The FSM outputs the 3 bit select line number for the 8-1 multiplexer and the 3-8 decoder. The FSM has 8 states (1 for each seven segment display) where the state changes when the output from the 1ms counter is 1. The FSM then and output's the 3 - bit select number corresponding to the state (i.e S1 = "000", S2 = "001" etc.).

III. Experimental Setup

The experimental setup of this project consisted of using the NEXYS A7-50T board and Vivado - the same components as seen in the lab. One of the strenuous parts of this project was writing the test bench for the 7-segment display. For our usage, displaying stock tickers, the configuration of CA-CG was difficult to map to the tickers for characters not commonly displayed on a 7 segment, such as t.

IV. Results

Upon completion of our code and a synthesis without critical errors, we were able to generate a bitstream and program the FPGA. Our initial test was not functional. We eventually figured out that the input clock of our shift register was incorrect and was shifting every 10 ns instead of every second. This was because for testing the circuit with our testbench, it was necessary for this process to be quicker than the 1s it is supposed to be as the simulation would have taken a significant amount of time to process. After this correction we were able to get the circuit working as planned. The switch functioned as intended to select which stock and corresponding price were to be displayed and the reset button was used to send the selection to the seven segment displays. The datapath circuit controlled which stocks were selected and what the output to the seven segment displays should be based off that selection. The FSM controlled the timing of the serializer, allowing us to display multiple different characters despite all the seven segment displays sharing the same data-in line.

Conclusions

The main take away points from this project are that we learned a lot about 7 segment displays outside of what was taught in the notes or lecture. Learning new methods of displaying and translating messages across all the 7 segment displays. Issues that remain to be solved and cannot be solved is not being able to display some tickers on the 7 segment display. Tickers with letters such as Z, X, K, W etc cannot be displayed on a 7 segment display. An improvement that could be implemented is to automatically display the next selected stock

ticker when the switches are flipped. The reset button must currently be pressed in order to display the next selected ticker. Another finite state machine could achieve this goal.

References

[1] Llamocca, Daniel. “DIGITAL SYSTEM DESIGN VHDL Coding for FPGAs].” RECRLab, Electrical and Computer Engineering Department, Oakland University www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html.

[2] “Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics.” Xilinx.com, Xilinx, Inc., 13 Apr. 2017, www.xilinx.com/support/documentation/data_sheets/ds181_Artix_7_ Data_ Sheet.pdf

[3] VHDL Coding for FPGAs. [Online]. Available: http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html. [Accessed: 14-Apr-2021].

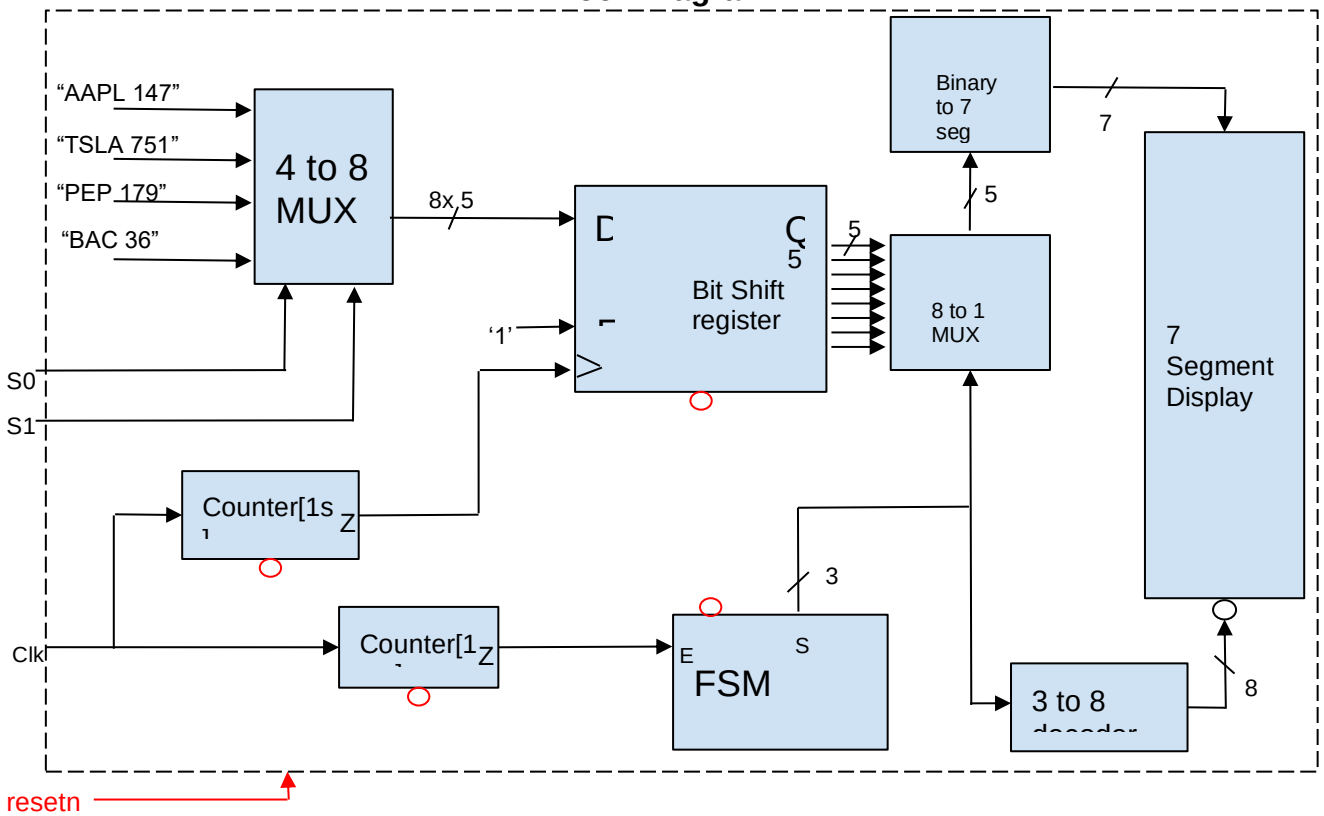
[4] 7-segment serializer (four displays) [Online]. Available: http://www.secs.oakland.edu/~llamocca/Tutorials/VHDLFPGA/Viva do/Unit_7/serializer.zip [Accessed: 14-Apr-2021].

[5] Counter modulo-N (generic pulse generator) with enable and synchronous clear [Online] http://www.secs.oakland.edu/~llamocca/Tutorials/VHDLFPGA/Viva do/Unit_5/my_genpulse_sclr.vhdl [Accessed: 14-Apr-2021].

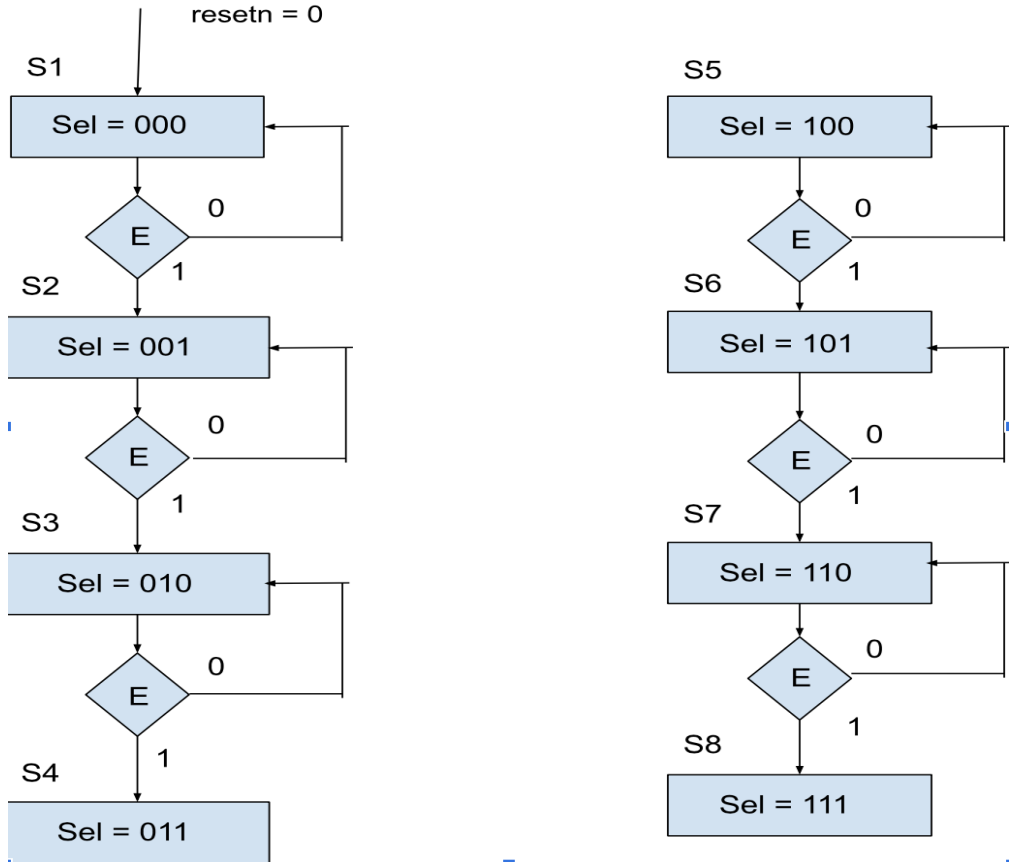
[6] D. Llamocca, “Laboratory 5”. Oakland University, Rochester Michigan, 2022.

[7] D. Llamocca, “Notes – Unit 7”. Oakland University, Rochester Michigan, 2022.

Block Diagram



FSM ASM Diagram



Top File Timing Diagram