Simon Says

List of Authors (Victor Lee, Jason Dickey, Nick Ottoy, Billy Jaynes)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: lee7@oakland.edu, jasondickey@oakland.edu, nottoy@oakland.edu, jaynesjr@oakland.edu

***Abstract*—Simon Says is a game that tests the user skill and memory. With the use of FSMs, registers, counters, and multiplexors, this program will have the user match LEDs with its corresponding switch below it. The user can ultimately win or lose when the Seven Segment Displays display "victory" or "you lose" respectively.**

I. Introduction

Using Vivado, a memory game influenced by Simon Says, will be built programmed onto a Nexys A7 board. This game will have the player memorize a set of lit LEDs of which will deactivate after a period of three seconds. After the LEDs deactivate, the user must then use their skill and memory to flip the corresponding switches within ten seconds to win. If the user passes, the seven segment displays will display "yea yeet", otherwise it will display "you lose". The user must complete ten of these cycles without failing once in order to finally be prompted with "victory". The goal of this project is to successfully program an autonomous sequence of which cannot be interrupted until the user is prompted to provide in input in the form of flipping switches or clicking buttons. The greatest challenges to the project were the implementation of a complex state machine with the incorporation of several counters. Ultimately, we learned that many states were needed within the finite state machine in order display the correct sequence of segments from the seven-segment display and display them for a set time.

II. Methodology

Using a Nexys A7 FPGA Board, Simon Says, a memory game, was be built in Vivado. Our project used a different set of LEDs in our FSM for each level. With these values already set, we used comparators, and counters to check the user input and regulate the time allowed to the user. The program will also consist of components such as registers, clock dividers, multiplexors, and seven segment displays. After using a test bench to test the program, the code will be programmed onto the board and await user inputs.

To design our project, we first spent time deciding what type of components would be useful. We used a few components that we created in previous labs and also needed to custom make several components of our own. Our FSM was the most difficult component to configure with each of the other components. To better regulate the LED display time as well as the user input time, we created one, three, and ten second counters.

III. Experimental Setup

Using testbenches and the Nexys A7 Board, components were tested prior to creating the top file. Specifically, the comparator, clock divider, FSM, and seven segment displays were tested using the methods mentioned previously. The comparator will be expected to output a '1' during specific and set inputs. The clock divider was expected to generate a longer pulse than the 100MHz clock. The seven segment displays were expected displayed at least two different images simultaneously. The FSM was expected to generate the correct outputs during certain states and inputs. Finally, after the implementation of the top file, the game was tested by programming the FPGA.

IV. Results

After thought and research, a block diagram was created as a baseline for the project. The diagram consisted of the various components that will be used as well as a basic state machine design. After creating and testing each component, they were eventually completed. The top file was created and only consisted of port mapping each component. After programming the code with a constraint file onto the board, it was tested and mostly debugged.

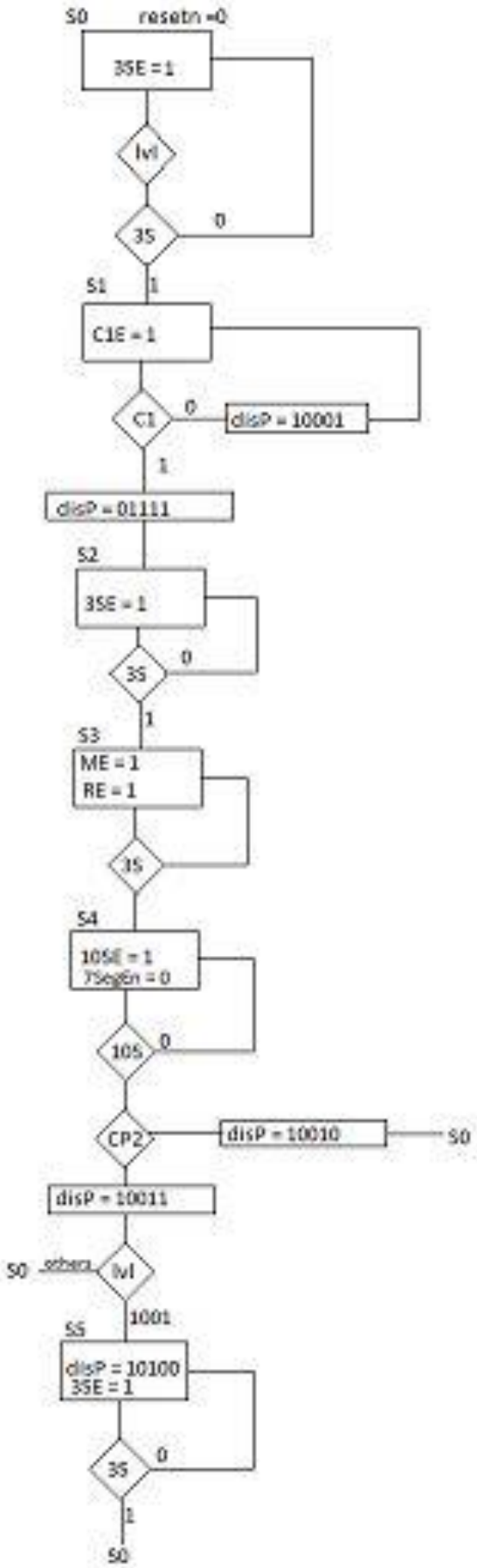
Conclusions

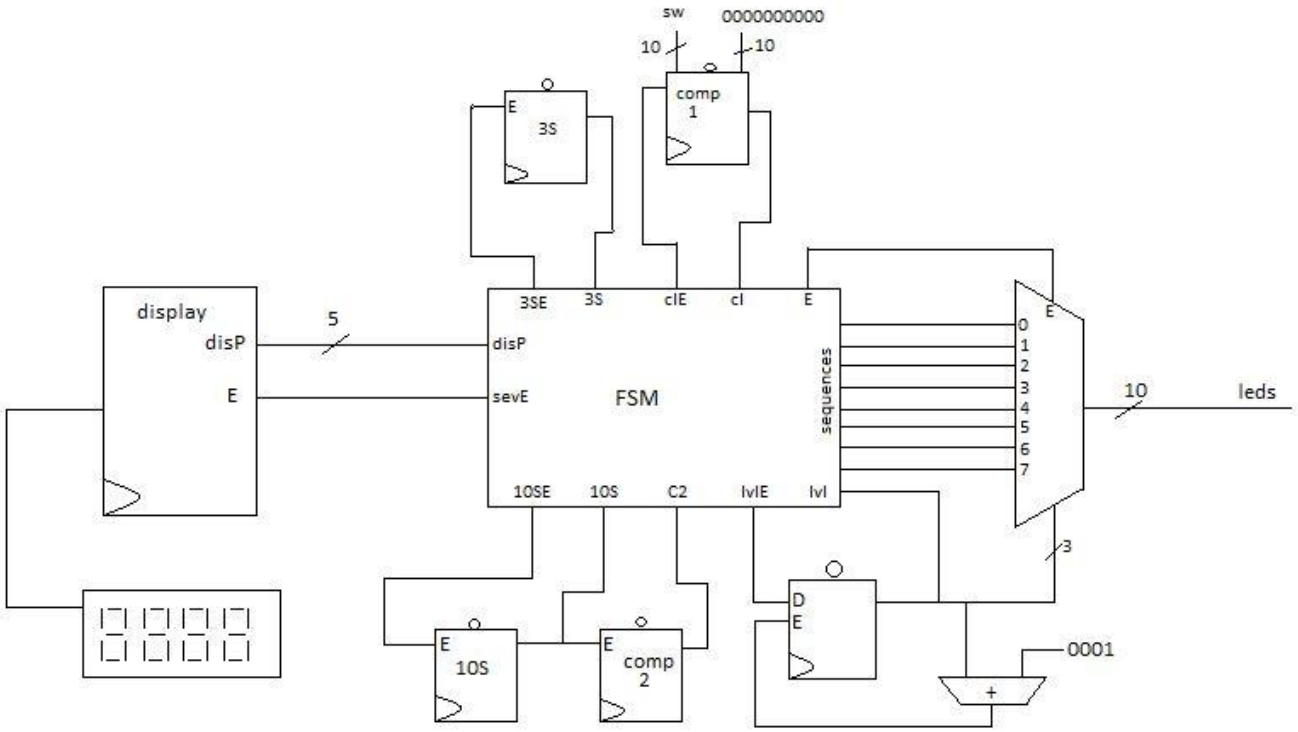
The main take-away of this project, is the importance of testing a small component before developing a greater component. Although not as grand, the process of state machines was explored and became essential to the project. Implementing the counters with the FSM remained as the greatest challenge to the project and with time was resolved.

References

[1] Tsuker, D. (2017, October 12). 'Simon' CPE 133 Final Project. Retrieved March 25, 2019 from https://www.instructables.com/id/Simon-CPE-Final-Project/

[2] [FPGA Tutorial] Seven-Segment LED Display on Basys 3 FPGA. (n.d.). Retrieved April 7, 2019, from https://www.fpga4student.com/2017/09/seven-segment-led-display-controller-basys3-fpga.html


S0
resetn =0
3SE = 1
lvl
3S
0
1
S1
C1E = 1
C1
0
disP = 10001
1
disP = 01111
S2
3SE = 1
3S
0
1
S3
ME = 1
RE = 1
3S
S4
10SE = 1
7SegEn = 0
10S
0
CP2
disP = 10010
S0
disP = 10011
S0
others
lvl
1001
S5
disP = 10100
3SE = 1
3S
0
1
S0


sw
0000000000
10
10
comp
1
E
3S
display
disP
E
5
3SE
3S
clE
cl
E
disP
sevE
FSM
sequences
0
1
2
3
4
5
6
7
E
10
leds
10SE
10S
C2
lvlE
lvl
3
D
E
E
10S
E
comp
2
0001
+