Lab #2

Combinational Network Design

Lab Report

Bradley Manzo

916953788

TABLE OF CONTENTS

INTRODUCTION

OBJECTIVES

PART I

- Learn how to develop a truth table and produce both Sum of Products, and Product of Sums, to represent the general equations for outputs X0 through X6.
- Learn how to find the minterms and maxterms using Karnaugh Maps to simplify the general equations and reduce the total number of gates.
- Design SoP and PoS circuits in Quartus using AND->OR and OR->AND configurations respectively.
- Simulate the outputs in Altera-Sim and generate every possible output by forcing the inputs from 000 through 111.
- Confirm that the simulation is correct by uploading the program to the DE10-Lite Board

PART II

- Implement a clock circuit to effectively produce a random value at the push of button KEY[0].

PROCEDURE

DESIGN AND TEST PROCEDURE

PART I

- First I generated a truth table seen in *Figure 1*, by analyzing the problem plotting each output for every input from 000 to 111.
- I then found the minterms and maxterms for each output X0 to X6
- Next I mapped the maxterms of each output to Kmaps and calculated the general equations as seen in *Figure 2* and *Figure 3,* which I will use for the Product of Sums Circuit.

- Starting with Sum of Products in *Figure 5,* after finding the general equations by calculating kmaps using minterms, I decided that there were too many prime implicants and unique equations to consider this a simplification.
- Instead, I created a two level circuit which generates high and low signals from each input, I created a column of 8 AND gates to generate every combination of signals and then I created a row of OR gates for each output which take in the necessary minterm values. I have X5 permanently grounded since it never has to be powered.
- Then for the Product of Sums in *Figure 4,* I followed a similar procedure, however this time with a column of OR gates determined from every unique sum from the general equations. I then create a row of AND gates, one for each output X0-X6 and connect them to the required OR gates. Again, I have X5 permanently grounded.
- Once I assigned the pins for the board, I generated two simulations using the Altera-Sim tool in *Figure 6* and *Figure 7* and I confirmed that my outputs were correct for each input by forcing SW to values 000 through 111.
- Finally I uploaded each of the two programs to my DE10-Lite Board and confirmed that it works successfully.

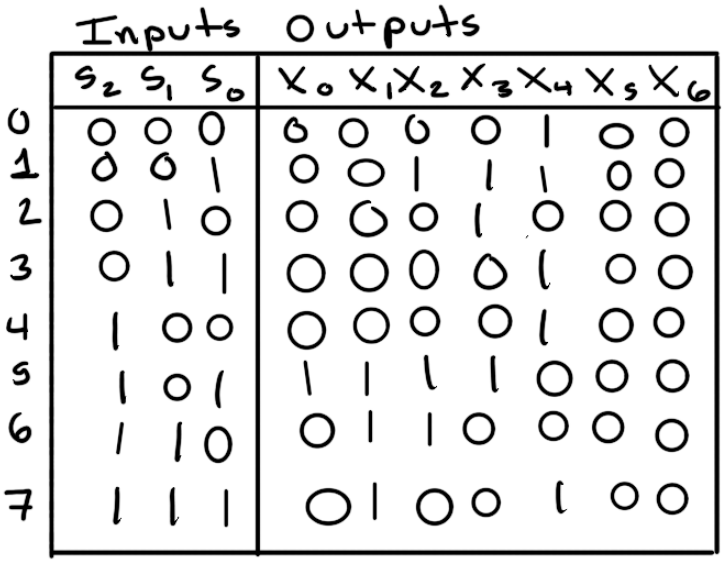
PART II

- For Part II, I implemented the provided counter circuit at the top where my inputs are in *Figure 8*.
- After compiling the program I uploaded it to my board and confirmed that it worked successfully.

ANALYSIS/RESULT

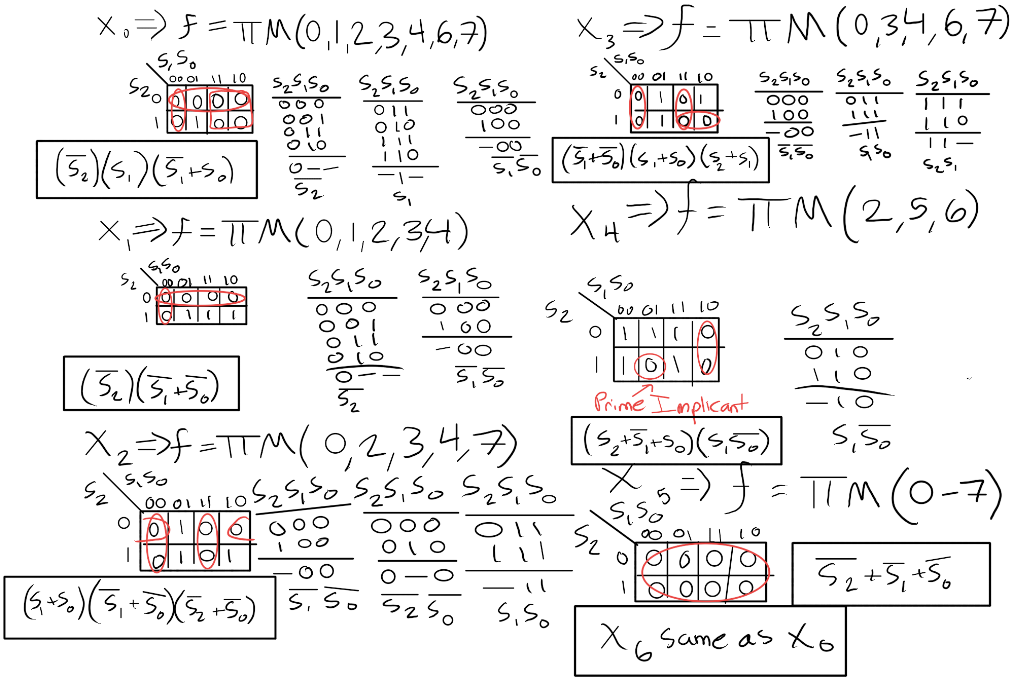
DATA

PART I:



	Inputs			Outputs						
	S_2	S_1	S_0	X_0	X_1	X_2	X_3	X_4	X_5	X_6
0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	0	0	1	1	1	0	0
2	0	1	0	0	0	0	1	0	0	0
3	0	1	1	0	0	0	0	1	0	0
4	1	0	0	0	0	0	0	1	0	0
5	1	0	1	1	1	1	1	0	0	0
6	1	1	0	0	1	1	0	0	0	0
7	1	1	1	0	1	0	0	1	0	0

Figure 1: Problem Truth Table



$X_0 \Rightarrow f = \Pi M(0,1,2,3,4,6,7)$
$(\overline{S_2})(S_1)(\overline{S_1}+S_0)$
$X_1 \Rightarrow f = \Pi M(0,1,2,3,4)$
$(\overline{S_2})(\overline{S_1}+\overline{S_0})$
$X_2 \Rightarrow f = \Pi M(0,2,3,4,7)$
$(S_1+S_0)(\overline{S_1}+\overline{S_0})(\overline{S_2}+\overline{S_0})$
$X_3 \Rightarrow f = \Pi M(0,3,4,6,7)$
$(\overline{S_1}+\overline{S_0})(S_1+S_0)(S_2+S_1)$
$X_4 \Rightarrow f = \Pi M(2,5,6)$
Prime Implicant
$(S_2+\overline{S_1}+S_0)(S_1\overline{S_0})$
$X_5 \Rightarrow f = \Pi M(0-7)$
$\overline{S_2}+\overline{S_1}+\overline{S_0}$
X_6 same as X_0

Figure 2: PoS Maxterm Calculations and General Equations

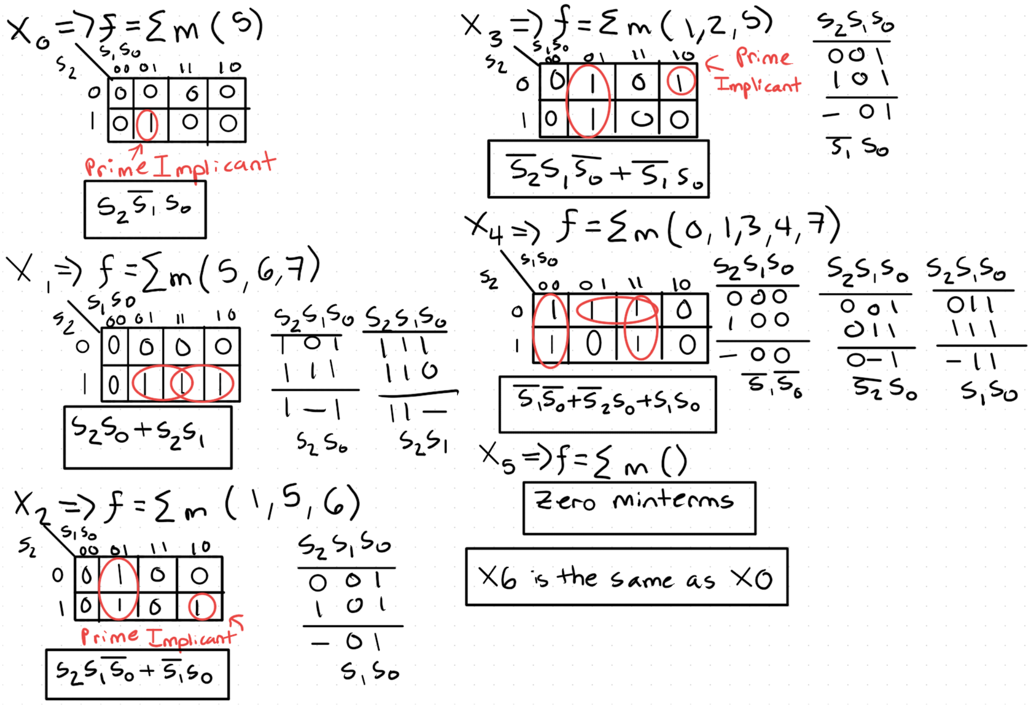


Figure 3: SoP Minterm Calculations and General Equations

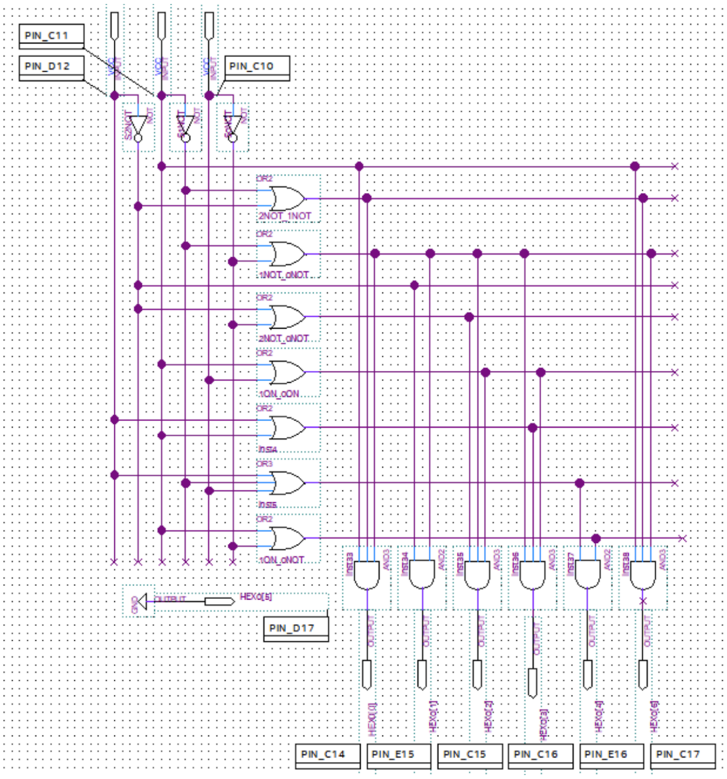


Figure 4: Product of Sums Quartus Circuit using Reduced General Equations

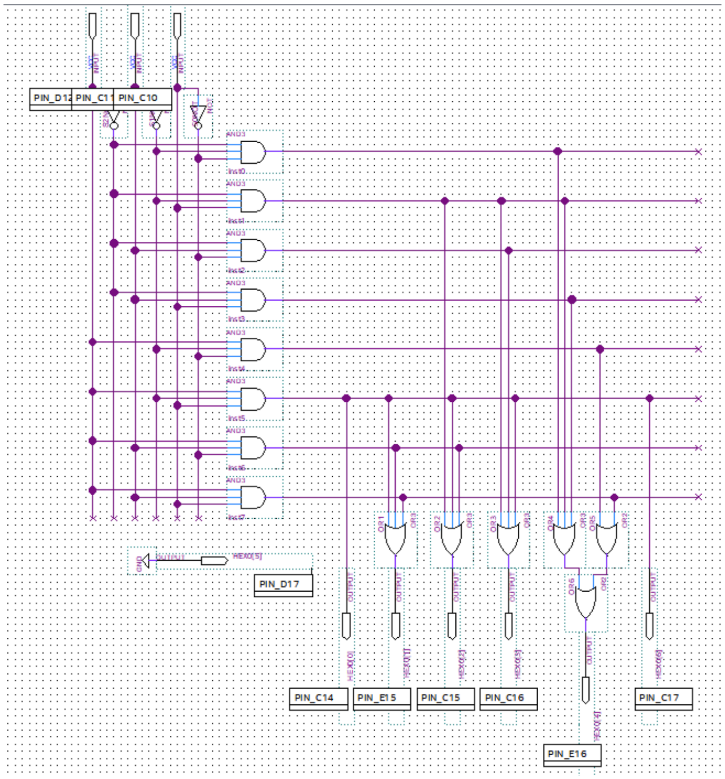
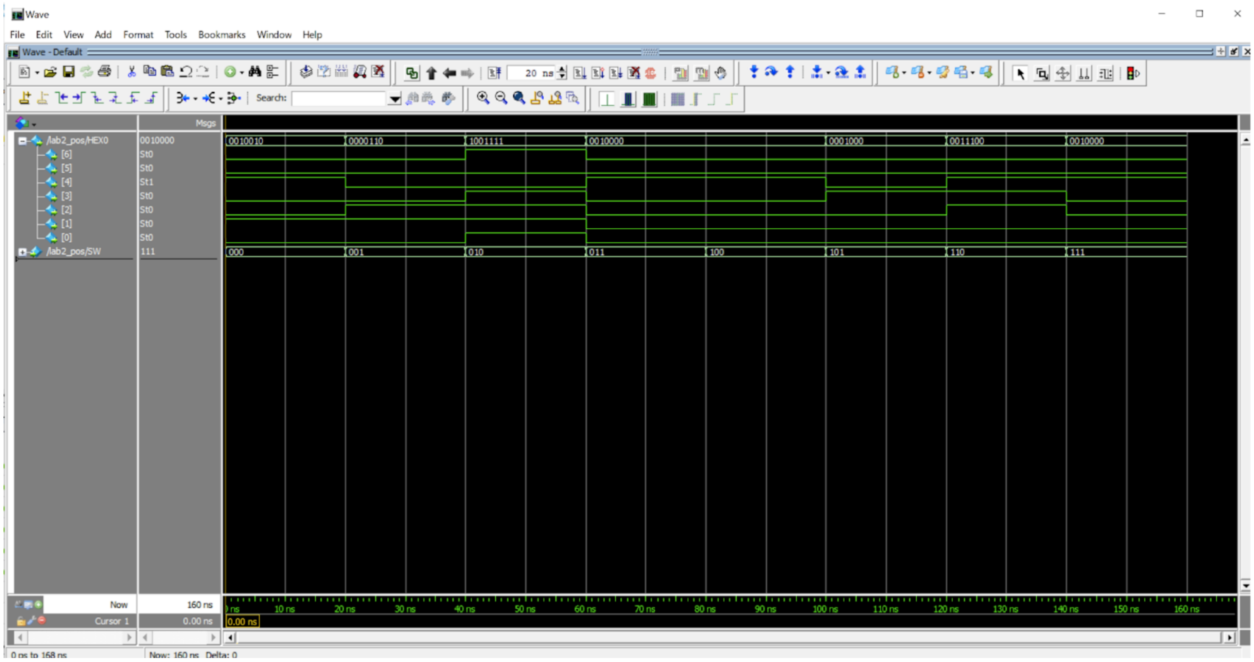


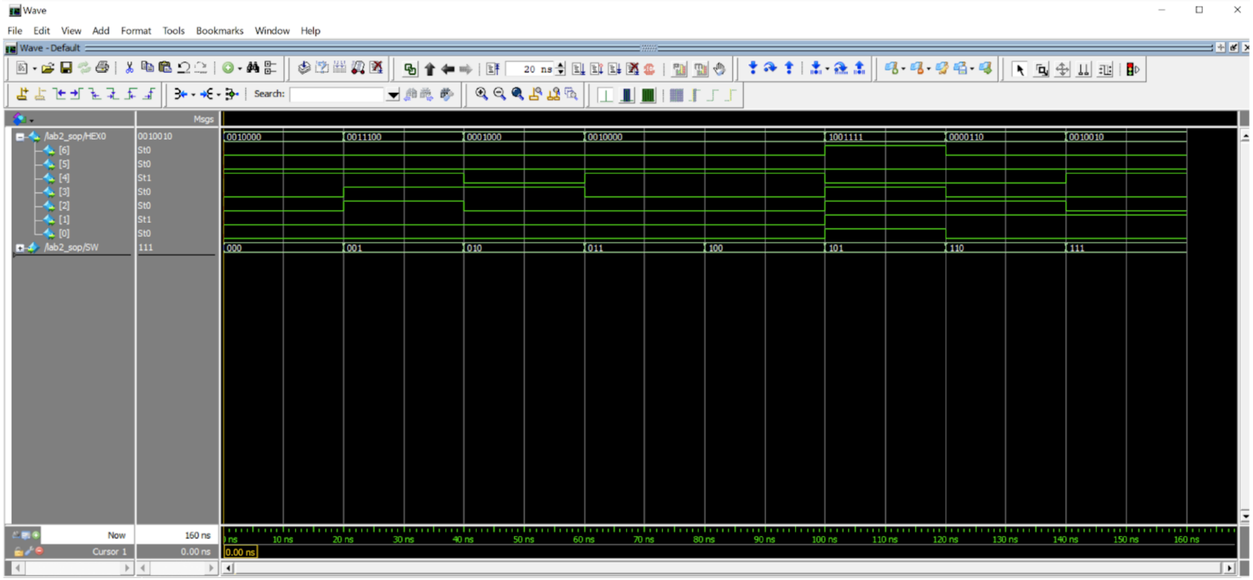
Figure 5: Sum of Products Quartus Circuit using Minterms



```
/lab2_pos/HEX0   0010000   0010010  0000110  1001111  0010000  0001000  0011100  0010000
  [6]            St0
  [5]            St0
  [4]            St1
  [3]            St0
  [2]            St0
  [1]            St0
  [0]            St0
/lab2_pos/SW     111       000  001  010  011  100  101  110  111

Now      160 ns    0 ns 10 ns 20 ns 30 ns 40 ns 50 ns 60 ns 70 ns 80 ns 90 ns 100 ns 110 ns 120 ns 130 ns 140 ns 150 ns 160 ns
Cursor 1 0.00 ns   0.00 ns
0 ns to 168 ns     Now: 160 ns  Delta: 0
```

Figure 6: Product of Sums Simulation



```
Wave
File  Edit  View  Add  Format  Tools  Bookmarks  Window  Help
Wave - Default

                    Msgs
/lab2_sop/HEX0      0010010   0010000  0011100  0001000  0010000  1001111  0000110  0010010
  [6]               St0
  [5]               St0
  [4]               St1
  [3]               St0
  [2]               St0
  [1]               St1
  [0]               St0
/lab2_sop/SW        111       000  001  010  011  100  101  110  111

Now       160 ns
Cursor 1  0.00 ns    0.00 ns
```

Figure 7: Sum of Products Simulation

PART II:

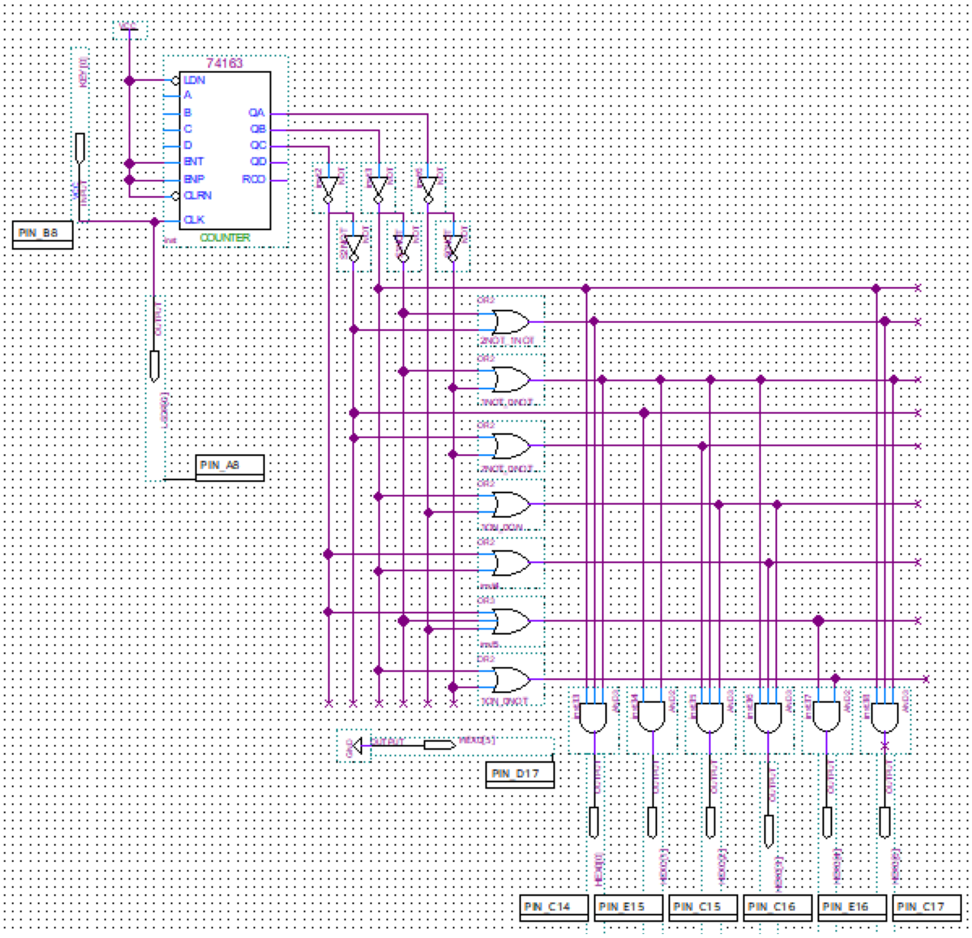


Figure 8: PoS Circuit with Counter Circuit Input Generation

ANALYZING RESULTS

PART I

- After completing the first part of the lab, I've learned a great deal about how Karnaugh maps can make the implementation of certain outputs much simpler in SoP and PoS two level circuits. Originally, I tried doing both without minimizing general equations and just connecting every minterm and maxterm with a column of 8 gates for each value 000 through 111. However I found that this was very inefficient for my PoS circuit as some outputs required 7 maxterms. Reducing the general equations with kmaps greatly reduced my need for complex OR gates and redundancy within my circuit. However I found the opposite to be true with my SoP Circuit, as the amount of Prime implicants in the kmaps would increase the amount of exclusive OR gates, and wouldn't yield as great of a reduction. Out of my own curiosity, I decided to create a a circuit using just the minterms and a column of 8 AND gates and I got the desired result with more or less the same amount of gates as the reduced general equations would have provided me. Despite both of these circuits displaying the correct outputs and in the right order, my PoS circuit displays the letters of GO AGGIES in reverse. This behavior on my waveform also exists when uploading to my DE10-Lite Board. Although my TA verified both my waveform and my Board behavior, I am still unsure of what this means and how I would go about resolving it. I have verified my kmaps are in the correct order, but it might be an issue with how I translated my general equations to my circuit design, however the similar design worked as expected for my SoP circuit, in both the waveform and the DE10-Lite Board behavior.

PART II

- This part went as expected and after implementing the provided counter circuit in my PoS circuit in Figure 8 and uploading it to the DE10-Lite board, I was able to effectively generate one of the 7 outputs at random with the push of KEY0.

CONCLUSION

- After completing this lab, I learned the value of reducing complex problems to their general equations, and while not impossible to achieve directly through minterms and maxterms, will greatly reduce the amount of gates and therefore reduce the costs of developing a circuit in the real world. I gained a greater understanding of the design process through creating the truth table to establish the general equations to finding a two level circuit that is easy to understand and modify. I also feel like I gained a greater understanding concerning the role of inputs and outputs in a circuit, as the problem required our input values to be a combination of SW inputs, and our output values to be a combination of LED outputs. I have also solidified my understanding of how the system clock can be used to generate random values from a set of provided values. I am still unsure of what exactly went wrong with my PoS circuit that yielded the reverse order of the GO AGGIES output, however I would narrow it down to the design of my circuit and possibly how I designed my OR gates in said circuit. Besides this, everything else in my lab went as expected. I believed my design and testing procedures to be quite thorough, and besides practicing my kmaps more, I wouldn't change much about my design process. Additionally, I quite like the design of my two level circuit, so I may reuse this format in future assignments or labs.