Lab #4

Latches, Flip-Flops, and Registers

Lab Report

Bradley Manzo
916953788

TABLE OF CONTENTS

INTRODUCTION

OBJECTIVES

PART I

- Recreate and learn how the Altera-FPGA RS Gated Latch Circuit works.

PART II

- Create the Gated D Latch using the provided schematic and understand the difference between the RS Gated Latch Circuit

PART III

- Learn how two Gated D Latches can be used to store data in an Edge Triggered D Flip-Flop

PART IV

- Learn how the Edge Triggered D Flip Flops can be implemented as Registers and store 12 bits of data.
- Using what we learned about 7 segment displays in *Lab 2,* create a symbol to convert a 4 bit binary input to a hexadecimal and display the values on the HEX0..6[] outputs.

PROCEDURE

DESIGN AND TEST PROCEDURE

PART I

- Analyze the provided schematic and compile the circuit in Quartus as seen in *Figure 1*.
- Download to the DE-10 Lite Board and demonstrate to the TA.

PART II

- Analyze the provided schematic and compile the circuit in Quartus as seen in *Figure 2.*
- Download to the DE-10 Lite Board and demonstrate to the TA.

PART III

- Analyze the provided schematic and compile the circuit in Quartus as seen in *Figure 3*.
- Download to the DE-10 Lite Board and demonstrate to the TA.

PART IV

- Design a symbol to convert between binary and hexadecimal
  - Establish a truth table to determine which LEDs in the 7 segment display are high or low for every binary input from 0-15 as seen in *Figure 4*.
  - Next, create K-maps for every output X6-X0 and minimize the product of sums in order to find the general equations as seen in *Figure 5*.
  - After finding the simplified general equations, create a Product of Sums circuit in quartus and save it as a symbol file as seen in *Figure 6*.
- Create a 12 bit register circuit using the dff component which emulates the behavior of the D-Latch Circuit from Parts II and III and save as a symbol as seen in *Figure 7*.
- Analyzing the provided schematic, implement both of these custom symbols into a top level circuit as seen in *Figure 8*, and upload the circuit to the DE-10 Lite Board
- Demonstrate the proper behavior to the TA.

ANALYSIS/RESULTS

DATA

PART I:

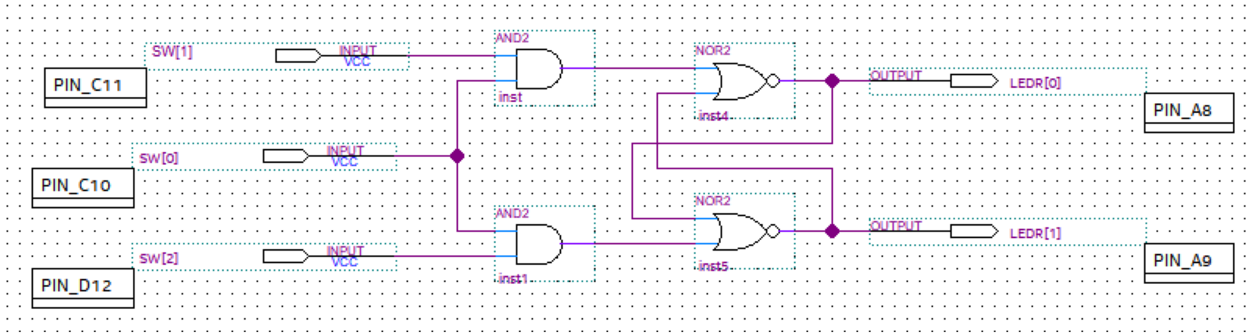


Figure 1: Gated RS Latch Circuit

PART II:

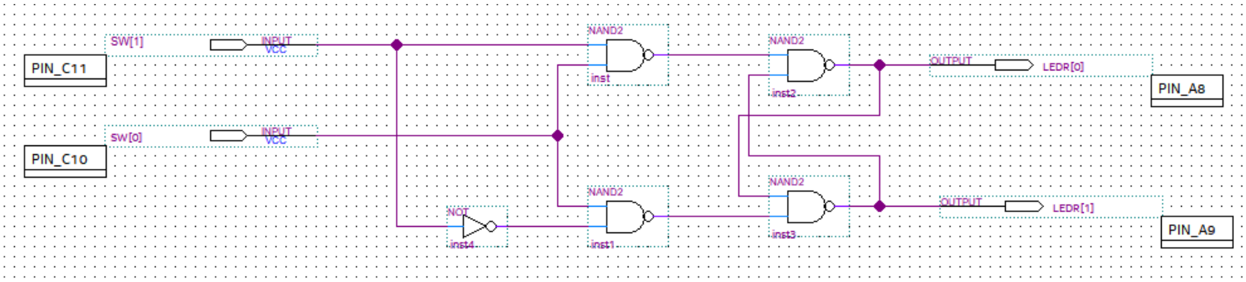


Figure 2: Gated D Latch Circuit

PART III:

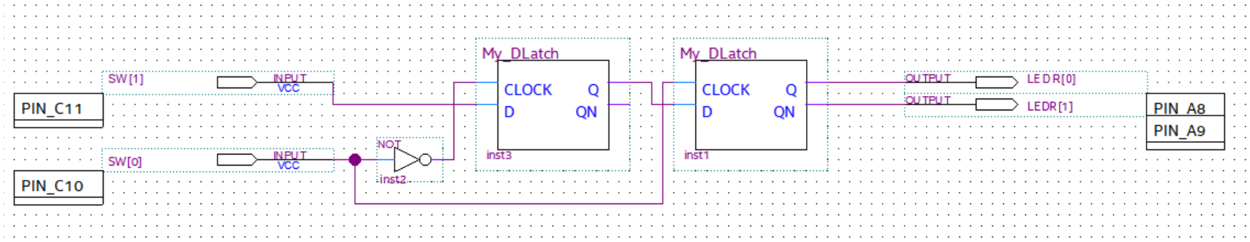
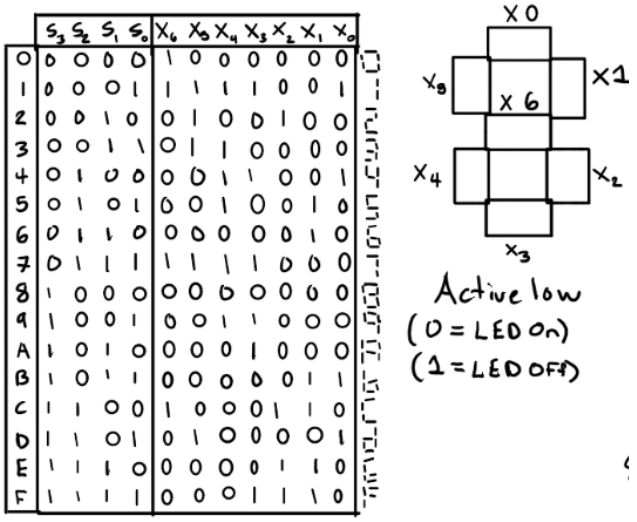


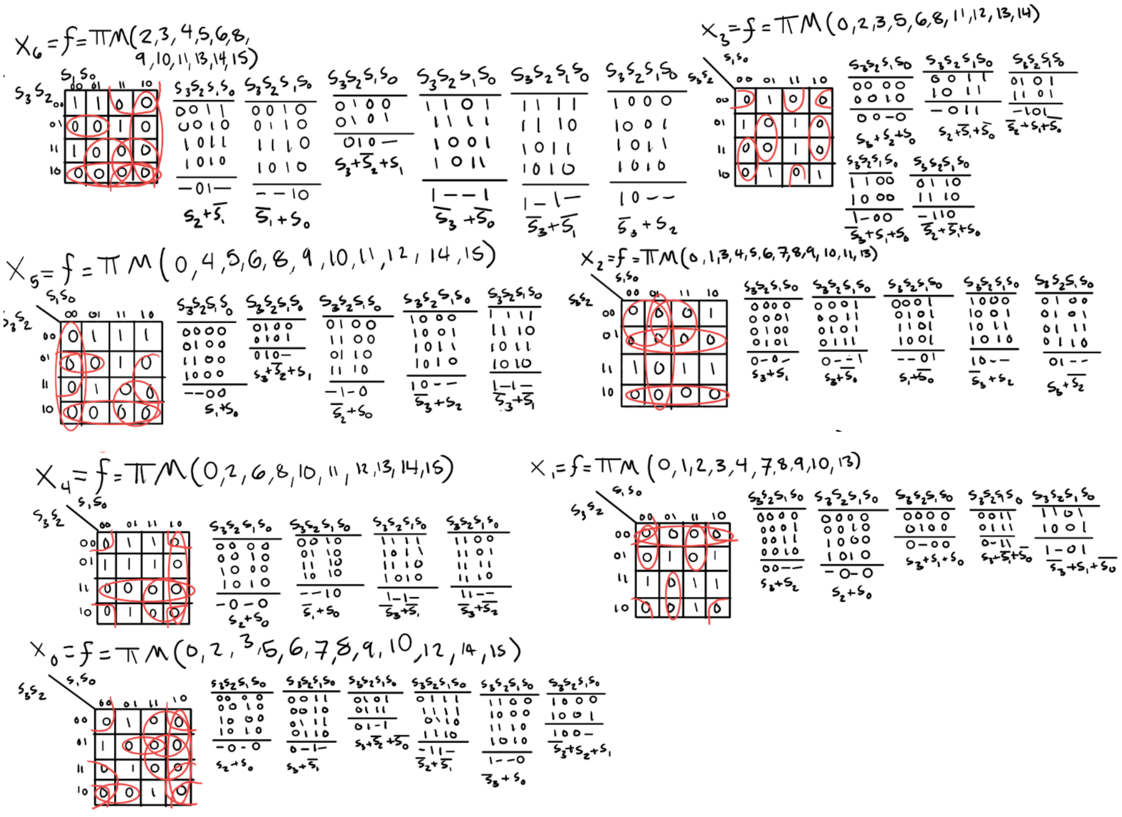
Figure 3: Edge-Triggered D Flip-Flop

PART IV:



	S_3	S_2	S_1	S_0	X_6	X_5	X_4	X_3	X_2	X_1	X_0
0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	0	0	1
2	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0	0	0	0
4	0	1	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	0	1	0	0	1	0
6	0	1	1	0	0	0	0	0	0	1	0
7	0	1	1	1	1	1	1	1	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	1	0	0	0
A	1	0	1	0	0	0	0	1	0	0	0
B	1	0	1	1	0	0	0	0	0	1	1
C	1	1	0	0	1	0	0	0	1	1	0
D	1	1	0	1	0	1	0	0	0	0	1
E	1	1	1	0	0	0	0	0	1	1	0
F	1	1	1	1	0	0	0	1	1	1	0

Figure 4: Bin2Hex Truth Table and 7 Segment Visualization



$X_6 = f = \Pi M(2,3,4,5,6,8,9,10,11,13,14,15)$
S_1S_0
S_3S_2
$S_2+\overline{S_1}$
$\overline{S_1}+S_0$
$S_3+\overline{S_2}+S_1$
$\overline{S_3}+\overline{S_0}$
$\overline{S_3}+\overline{S_1}$
$\overline{S_3}+S_2$
$X_3 = f = \Pi M(0,2,3,5,6,8,11,12,13,14)$
$S_3+S_2+S_0$
$S_2+\overline{S_1}+\overline{S_0}$
$\overline{S_2}+S_1+\overline{S_0}$
$\overline{S_3}+S_1+S_0$
$\overline{S_2}+\overline{S_1}+S_0$
$X_5 = f = \Pi M(0,4,5,6,8,9,10,11,12,14,15)$
S_1+S_0
$S_3+\overline{S_2}+S_1$
$\overline{S_2}+S_0$
$\overline{S_3}+S_2$
$\overline{S_3}+\overline{S_1}$
$X_2 = f = \Pi M(0,1,3,4,5,6,7,8,9,10,11,13)$
S_3+S_1
$S_3+\overline{S_0}$
$S_1+\overline{S_0}$
$\overline{S_3}+S_2$
$S_3+\overline{S_2}$
$X_4 = f = \Pi M(0,2,6,8,10,11,12,13,14,15)$
S_2+S_0
$\overline{S_1}+S_0$
$\overline{S_3}+\overline{S_1}$
$\overline{S_3}+\overline{S_2}$
$X_1 = f = \Pi M(0,1,2,3,4,7,8,9,10,13)$
S_3+S_2
S_2+S_0
$S_3+S_1+S_0$
$S_3+\overline{S_1}+\overline{S_0}$
$\overline{S_3}+S_1+\overline{S_0}$
$X_0 = f = \Pi M(0,2,3,5,6,7,8,9,10,12,14,15)$
S_2+S_0
$S_3+\overline{S_1}$
$S_3+\overline{S_2}+\overline{S_0}$
$\overline{S_2}+\overline{S_1}$
$\overline{S_3}+S_0$
$\overline{S_3}+S_2+S_1$

Figure 5: 7-Segment Display K-maps and General Equations

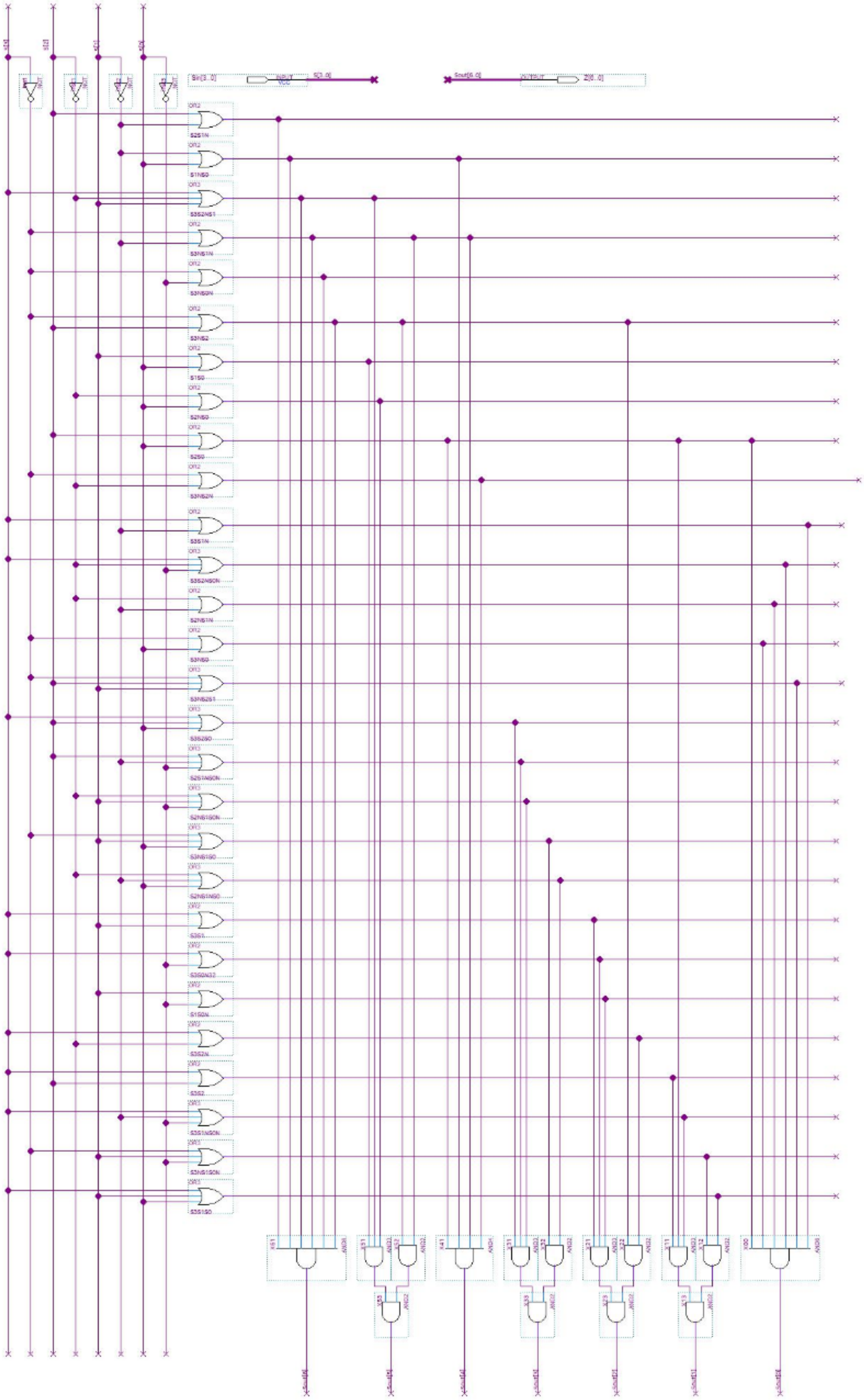


Figure 6: Bin2Hex Product of Sums Circuit

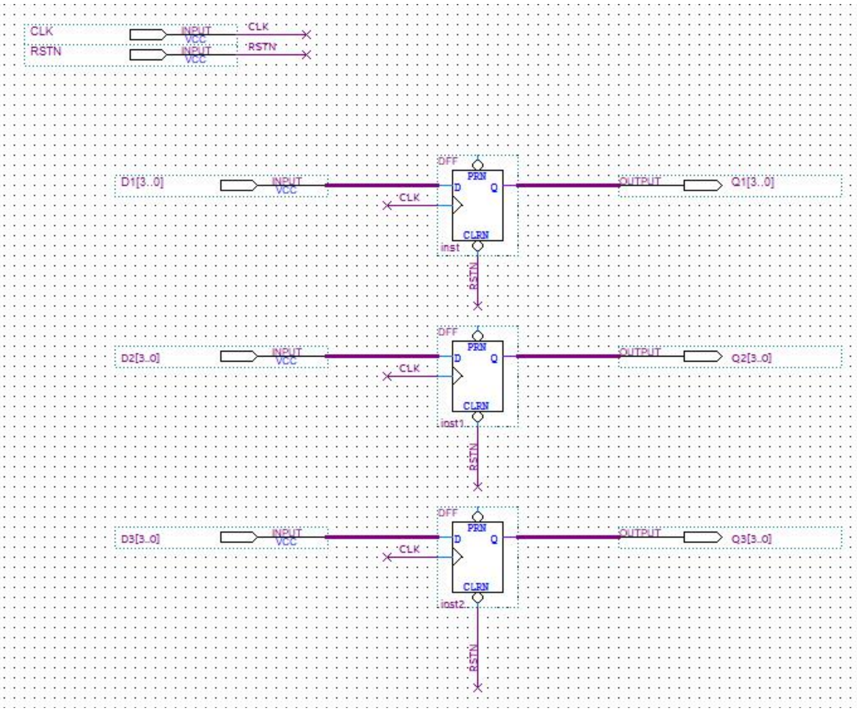


Figure 7: 12 Bit Register Circuit

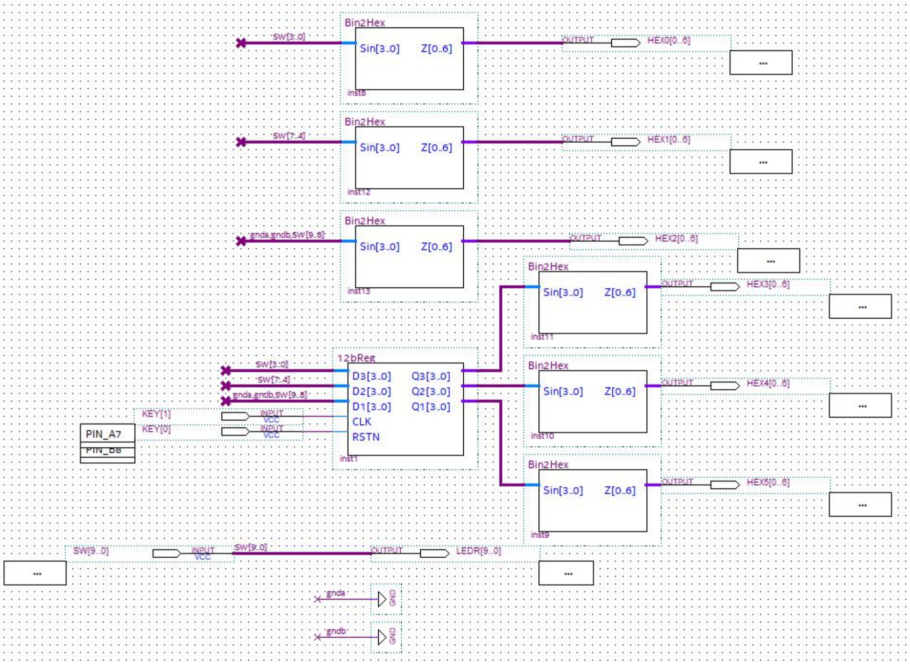


Figure 8: Top Level Schematic with Custom Symbols

ANALYZING RESULTS

PART I

- The Gated RS Latch worked as expected, with SW[0] acting as the enabler, SW[1] as the set, and SW[2] as the reset.

PART II

- The D Latch works similarly to the RS Latch, however it apparently only requires one input to set, while the RS needs two(The set and the Enabler).

PART III

- The edge triggered Flip-Flop works as expected and effectively stores one bit of memory when the set input is switched on.
- I had a bit of trouble demonstrating to the TA but upon revisiting my symbol and correcting the circuit, I completed the check off.

PART IV

- The designing phase of the 7 segment binary to hexadecimal converter was quite easy after learning how to effectively generate the kmaps and general equations in Lab 2. Additionally, the 12 bit Register was just a series of edge triggered flip-flops like in Part III. I had a bit of trouble adapting to using buses instead of simple wire connections, but overall, it made the designing process much less cluttered! I had very little trouble debugging the program and I demonstrated the behavior to the TA as it worked as expected.

ANSWERS TO HIGHLIGHTED QUESTIONS

PART I

- When the CLK (SW[0]) is high, does the output change when R and S are changed?
  - When the enabler switch CLK is high, SW[1] turns on LEDR1, and SW[2] turns on LEDR0.
- When the CLK is low, does the output change when R and S are changed?
  - When the enabler switch CLK is low, LEDR0 is high and LEDR1 is low and neither of the switches affect the states.
- Is there any case when Q = QN? (i.e. when LEDR[1] and LEDR[0] are either both ON or both OFF). Describe this case. Since Q and QN should

always be complements, this case would be an illegal state for an RS latch.

  - The only case where Q = QN is when all three switches are high. This is an illegal state.
- When R=S=0, what happens to the output when the CLK is toggled?
  - When R and S are =0, CLK doesn’t change the LEDs
- When R=S=1 and CLK goes from 1 to 0, what happens to the output? Explain why this happens. Is it what you expect?
  - While it isnt what I would initially expect, this happens due to the fact that CLK being low disables the other switches, making the previous illegal state null, and displays the default.

PART II

- When CLK is high, does changing the D input affect the output?
  - Yes, when the CLK switch is high, SW[1] powers LEDR1 when low and LEDR0 when high respectively.
- When CLK is low, does changing the D input affect the output?
  - When CLK is low, changing the D input does not affect the output.
- Is there any case when Q = QN? (i.e. when LEDR[1] and LEDR[0] are either both ON or both OFF). Describe this case. Since Q and QN should always be complements, this case would be an illegal state for a D latch.
  - No there is no combination of inputs that would yield an ilegal state like in part I.

PART III

- When CLK is high, does changing the D input affect the output? If so, give examples.
  - When CLK is high, changing the D input does not affect the output.
- When CLK is low, does changing the D input affect the output?
  - When CLK is lowm changing the d input does not affect the output.

- When does the output change? How is this different from a gated D latch? The output only changes when the CLK switch is toggled after toggling the D latch, this is different from the gated D latch since the CLK switch actively changes the output rather than passively disabling it.

CONCLUSION

- After completing Lab 4, I feel that I have gained a greater understanding of how flip flops and latches can work alone and in larger circuits, and how they can effectively store bits of memory depending on how many flip flops are included. I've learned that latches have the advantage over flip flops, as latches are level triggered and don't require the control signal to change outputs like the flip flop does, rather the output will change immediately with the inputs. I feel I've learned a good deal about quartus too, as I've seen how effective buses are used to organize large quantities of connections. I will definitely be using this in future labs to make my circuits less confusing. Additionally, I feel I have gained even more experience with designing a 7 segment Product of Sums circuit, as it was much easier than in Lab 2, and I finished a similar homework question in even less time than the lab! I feel that my results accurately reflect the effort I have put into completing this lab and understanding the material.