Benny Chen
1/25/2022
Lab 3a

Theory

In this lab we built a simple switch to LED that will turn on that corresponding LED that is connected to that switch number. Though the concept is switching a LED on or off, we also use other components like a 7405 HEX inverter chip to help us. A 7405 HEX inverter has 6 outputs which work by inverting or “flipping” every input then outputting that.

The reason why we invert the input is due to how the switch itself behaves. A switch behaves like a “bridge” of sorts for connecting a power source to ground, when it is “off” the switch or bridge disconnects from the loop making it incomplete and the opposite when it's “on”. In our circuit, we connect our wires leading to the HEX inverter, from this when the switch is “off” it technically is “on” for the HEX inverter. This is due the switch completing the circuit by finding the path to the HEX inverter as the switch loop is disconnected, making only one path for the current to go through. The opposite happens if the switch is “on”, the switch path from the power source to ground is completed which is also the shortest path it needs to travel so it ignores the path to the HEX inverter making it “off” for the HEX inverter.

This circuit is the baseline in creating future circuits for this class as right now we just have a simple input from the switch to direct output, however later down the line we would put something in the middle of that so it would be input, logic, then output. This HEX inverter is different from its counterpart 7404 due to it being an open-collector system which we prefer. For a 7404 Inverter, which is a push-pull, the output would just be a simple invert of the input, 0 to 1 or 1 to 0, which is sent to the LED. However in this circuit, the 7405 chip has a different reaction when sent with those signals. When the 7405 chip is sent a low input, there is a capacitor in the chip that generates a high impedance that blocks the current from passing that pin, essentially making it like if the pin was disconnected from the chip and is not there. For a high input the opposite happens and a low impedance is generated, allowing the current to flow through the LED’s. This is essential as our LED’s “drive an external load” or have their own power source and in the future we will have multiple devices and chips that are all intertwined, trying to make their line high thus decreasing interference a lot.

According to Ohm’s Law, Voltage = Current * Resistance. Through this we have a 330Ω resistor in this circuit and the resistance from the LED’s which have a voltage drop of 1.7 V-2 V. The assuming power source is 5V and each LED is working independently as its own circuit. The total resistance, 330, and voltage = 5-1.7=4.3. With Ohm’s law, 4.3=I*330 making the current traveling through each LED 13.0303 mA.

Deliverables

POSTNET VWXYZ to XS3 DCBA table:

Deci	**P**	**O**	**S**	**T**	**NET**	**->**		**X**	**S**	**3**
	V	**W**	**X**	**Y**	**Z**	**->**	**D**	**C**	**B**	**A**
0	1	1	0	0	0	->	0	0	1	1
1	0	0	0	1	1	->	0	1	0	0
2	0	0	1	0	1	->	0	1	0	1
3	0	0	1	1	0	->	0	1	1	0
4	0	1	0	0	1	->	0	1	1	1
5	0	1	0	1	0	->	1	0	0	0
6	0	1	1	0	0	->	1	0	0	1
7	1	0	0	0	1	->	1	0	1	0
8	1	0	0	1	0	->	1	0	1	1
9	1	0	1	0	0	->	1	1	0	0

Karnaugh Maps:

D: V.W’ + W.X + W.Y

V	**YZ**					**V’**	**YZ**				
WX		**00**	**01**	**11**	**10**	**WX**		**00**	**01**	**11**	**10**
	00	X	1	X	1		**00**	X	X	0	X
	01	1	X	X	X		**01**	X	0	X	0
	11	X	X	X	X		**11**	1	X	X	X
	10	0	X	X	X		**10**	X	0	X	1

C: V’.Z + W’.X

V	YZ					V'	YZ				
WX		**00**	**01**	**11**	**10**	**WX**		**00**	**01**	**11**	**10**
	00	X	0	X	0		**00**	X	X	1	X
	01	1	X	X	X		**01**	X	1	X	1
	11	X	X	X	X		**11**	0	X	X	X
	10	0	X	X	X		**10**	X	1	X	0

B: V.X' + W.Z +X.Y

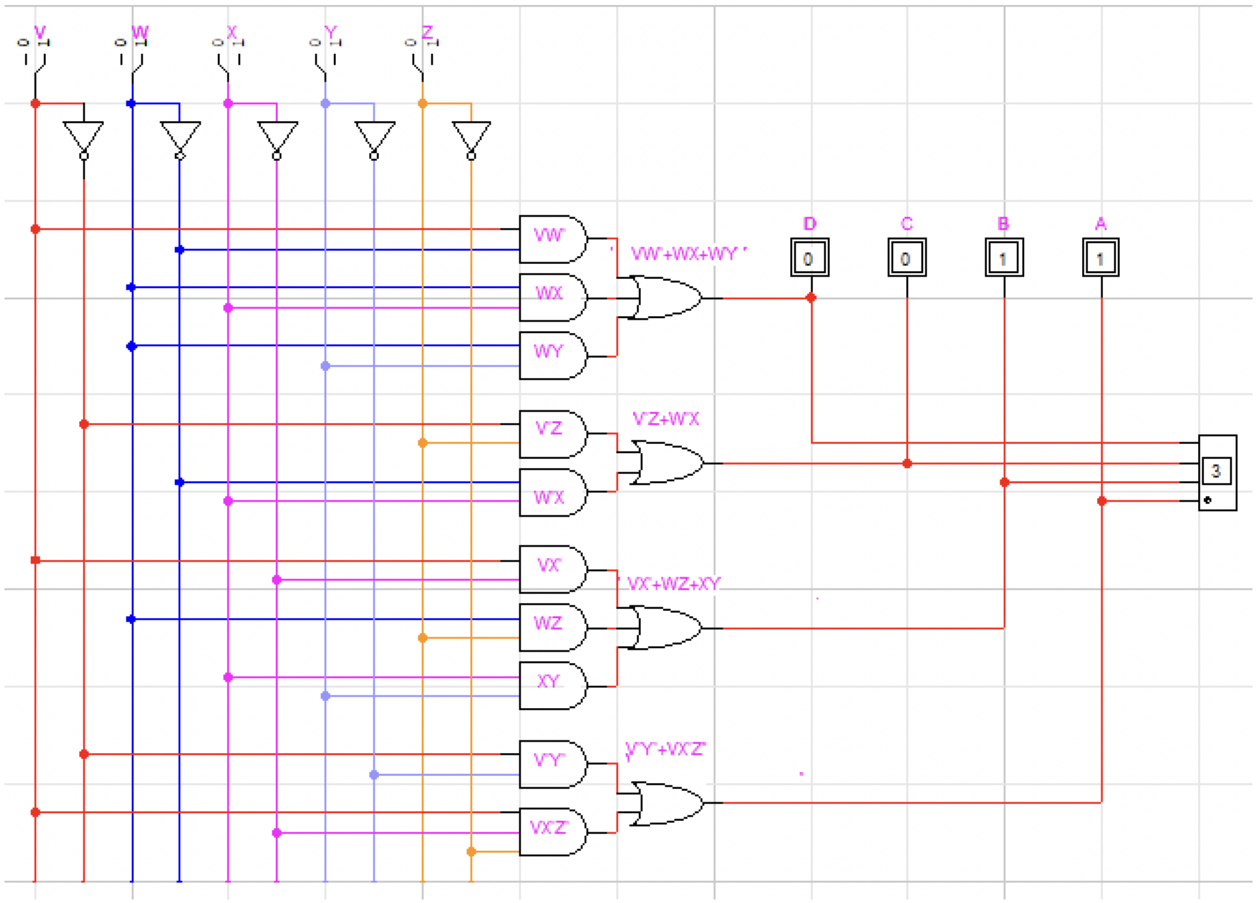
V	YZ					V'	YZ				
WX		**00**	**01**	**11**	**10**	**WX**		**00**	**01**	**11**	**10**
	00	X	1	X	1		**00**	X	X	0	X
	01	0	X	X	X		**01**	X	0	X	1
	11	X	X	X	X		**11**	0	X	X	X
	10	1	X	X	X		**10**	X	1	X	0

A: V'.Y' + V.X'.Z'

V	YZ					V'	YZ				
WX		**00**	**01**	**11**	**10**	**WX**		**00**	**01**	**11**	**10**
	00	X	0	X	1		**00**	X	X	0	X
	01	0	X	X	X		**01**	X	1	X	0
	11	X	X	X	X		**11**	1	X	X	X
	10	1	X	X	X		**10**	X	1	X	0

Equations:
D: V.W' + W.X + W.Y
C: V'.Z + W'.X
B: V.X' + W.Z +X.Y
A: V'.Y' + V.X'.Z'

Through the karnaugh maps, I'm able to group together like groups, with 1 and don’t care values, to create a much simpler equation than each instance having a unique one. Each color in the k-maps represents a grouping with overlapping ones having the bit highlighted.



Discussion

In this lab I learned more about simplifying equations using karnaugh maps, especially with more than 4 variables, and about how POSTNET works. I also found out a more efficient way of organizing my logicworks circuit by having the binary switches upside down. I also learned about why we invert the switch inputs, due to how the actual switch operates and why we even use an open-collector for our circuit. I found the actual reasoning why the switch is being inverted very interesting as the current is trying to find a path to ground.

Questions

1. **Create a 4-variable Karnaugh map for $\sum m$ (0,2,3,4,7,8,10,13,15) and reduce it as much as possible. Suboptimal solutions will receive partial credit.**

		BA			
	m	**00**	**01**	**11**	**10**
DC	**00**	**1**	**0**	**1**	**1**
	01	**1**	**0**	**1**	**0**
	11	**0**	**1**	**1**	**0**
	10	**1**	**0**	**0**	**1**

Equation:

D'B'A' + D'BA + DCA + C'A'

2. **Convert the number −59 to IEEE-754 floating point. Include only the first 6 bits of the mantissa. Show *all of your work*.**

Step 1: -59 = negative = signed bit = 1

Step 2: Convert 59 to binary

32 + 16 + 8 + 4 + 2 + 1 = 59

1/2 =.5 .5 * 2 = 1	**3/2 = 1.5 .5 * 2 = 1**	**7/2 = 3.5 .5 * 2 = 1**	**14/2 = 7.0 0 * 2 = 0**	**29/2 = 14.5 .5 * 2 = 1**	**59/2 = 29.5 .5 * 2 = 1**
32	**16**	**8**	**4**	**2**	**1**
1	**1**	**1**	**0**	**1**	**1**

111011 or 111011.0

Step 3: Convert to floating point

111011.0

move decimal 5 places to left
1.11011 x 2^5

Step 4: Convert exponent sign to unbiased then convert to binary

127 + 5 = 132

132 = 10000100

Step 5: Remove leftmost 1/after decimal point from binary representation of 59

1.110110
to
110110

Step 6: Combine sign bit, exponent, and mantissa

Signed bit	**Exponent**	**Mantissa (only 6 bits)**
1	**10000100**	**110110**

-59 in IEEE-754 32 bit floating point: 1 10000100 110110