COUNTER WITH DISPLAY

Project 2

CS 200
Spring 2022
Date Submitted: February 9, 2022

Sam Hemmelgarn
sph227@nau.edu
6016903

Project Overview

The purpose of this project was to create a binary-coded decimal (BCD) to 7 segment display counter. This particular circuit uses JK flip flops to represent 1 bit of binary and we can then string together 4 of these to represent a 4-bit binary number in 1 circuit. We can program the logic to those circuits to light up a segment of a 7-segment display, combining 7 circuits (1 for each segment) to light them up showing that particular binary digit in decimal. Another aspect of the project is that we want our counter to be modulo-10 instead of modulo-16 so that when the counter gets to the number 9, it resets back down to 0.

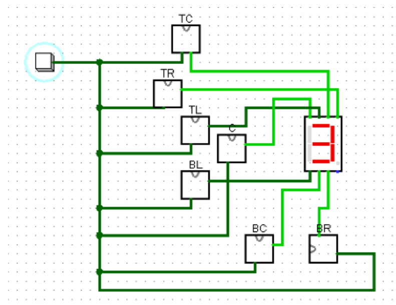
When I first started on the project, reading the overview, I really wasn't sure where to start. I read through the overview several times and played with the provided Logisim circuit to figure out how it was supposed to work. Because I had a hard time determining where to start, I started with the truth table. A truth table was created with 4 inputs, 1 input to represent each bit of binary in the digit we are trying to show. There are 7 outputs, 1 output for each segment of the display, indicating whether it should be 'on' for that particular binary digit or 'off'. Once the truth table was created, I used it to create the Karnaugh Maps for each segment of the display. Once I saw how the Karnaugh Maps worked out and how they translated to the provided 4-bit counter, it became more clear to me how to proceed. In the provided circuit, the lowest-order binary digit was on top and the highest-order binary digit was at the bottom. As you click the button (the 'clock') it clocks the circuit and causes it to count in binary from 0-15 and then back to 0000 once it reaches 1111. From the functions created, AND gates were used for each term in the function that turned 'on' or 'off' a particular segment in the display for that decimal digit, all connected to an OR gate connected to the display. 7 circuits were created based off the 7 Karnaugh Maps and functions and all circuits were wired together in the main circuit to the same clock button. The outputs of these 7 circuits were then connected to the input of the display that they were to correspond to, creating a counter that displayed decimal digits 1 through 9 and then reset to 0.

Results

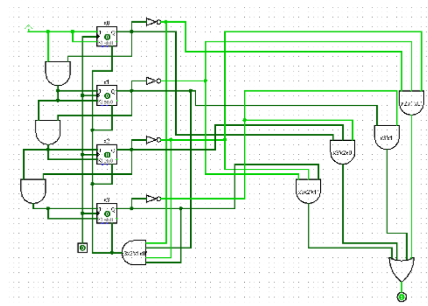
In the truth table, I used abbreviations to represent the different segments of the display, such as 'TC' for 'top center' and 'BL' for 'bottom left'. I used this because it allowed me to easily identify which segment I was working with, without having to remember or look up what arbitrary identifiers such as 'a', 'b', or 'c' were used for particular segments. In my truth table, I decided not to use the 'don't cares' as although it seemed as if the functions would be more simple, I erred on the side of making sure that my circuit was not true for digits 10-15. In the truth tables and KMaps, all not true's are left blank. During wiring of the individual circuits, I would wire them to a 7-segment display in Logisim, wire all of the gates, and clock the circuit after I was done wiring to ensure that the correct display segment lit up for the intended decimal digit. This method proved to be very helpful for me, as if the segment did not light up when it was supposed to, I was able to troubleshoot and diagnose what was causing the issue. There were times when a gate was wired to x0' instead of x0 and I was able to look at the KMaps to verify the functions were correct and re-wire the gate. Another aspect of the project that needed to be solved was how to get the counter to reset after 9 and go back to 0. What I did was connect an AND gate to each circuit that would only be true for the binary digit 1010. This gate was then wired to the 'clear' on all 4 flip-flops to reset each input back to 0, once the circuit counted up to 1010. See below for screenshots of Logisim circuit, truth table, KMaps, and functions.

Screenshots

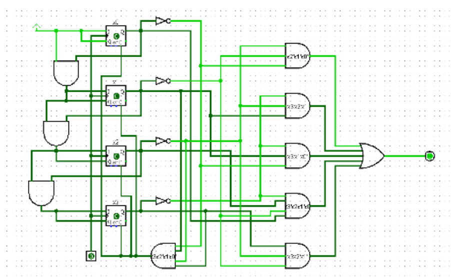
Main Circuit



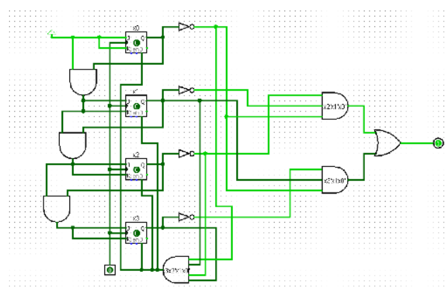
Top Center Circuit



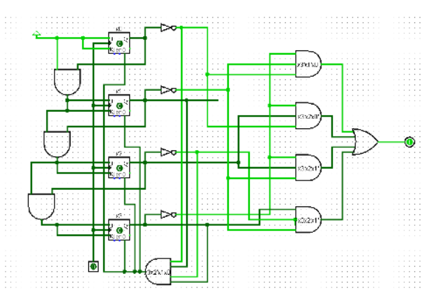
Bottom Center Circuit



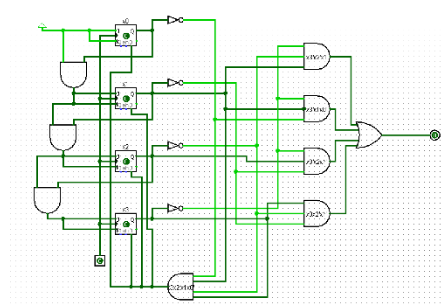
Bottom Left Circuit



Top Left Circuit



Center Circuit



Truth Table

A	B	C	D	TC	TR	BR	BC	BL	TL	C
0	0	0	0	1	1	1	1	1	1	
0	0	0	1		1	1				
0	0	1	0	1	1		1	1		1
0	0	1	1	1	1	1	1			1
0	1	0	0		1	1			1	1
0	1	0	1	1		1	1		1	1
0	1	1	0	1		1	1	1	1	1
0	1	1	1	1	1	1				
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1		1	1
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

Karnaugh Maps

Top Center

x1x0

x3x2	**00**	**01**	**11**	**10**
00	1		1	1
01		1	1	1
11				
10	1	1		

Top Right

x1x0

x3x2	**00**	**01**	**11**	**10**
00	1	1	1	1
01	1		1	
11				
10	1	1		

Bottom Right

x1x0

x3x2	**00**	**01**	**11**	**10**
00	1	1	1	
01	1	1	1	1
11				
10	1	1		

Bottom Center

x1x0

x3x2	**00**	**01**	**11**	**10**
00	1		1	1
01		1		1
11				
10	1	1		

Bottom Left

x3x2 \ x1x0	00	01	11	10
00	1			1
01				1
11				
10	1			

Top Left

x3x2 \ x1x0	00	01	11	10
00	1			
01	1	1		1
11				
10	1	1		

Center

x3x2 \ x1x0	00	01	11	10
00			1	1
01	1	1		1
11				
10	1	1		

Functions

Top Center = x2’x1’x0’ + x3’x1 + x3’x2x0 + x3x2’x1’
Top Right = x3’x2’ + x1’x0’x3’ + x1x0x3’ + x2’x1’
Bottom Right = x1’x3’x2’ + x1x0x3’ + x3’x2 + x3x2’x1’
Bottom Center = x2’x1’x0’ + x3’x2’x1 + x3’x1x0’ + x3’x2x1’x0 + x3x2’x1’
Bottom Left = x2’x1’x0’ + x3’x1x0’
Top Left = x3’x1’x0’ + x3’x2x0’ + x3’x2x1’ + x3x2’x1’
Center = x3’x2’x1 + x3’x1x0’ + x3’x2x1’ + x3x2’x1’

Conclusion

This project helped me to understand more about how truth tables and KMaps translate into real life circuits. I felt pretty lost at the beginning, not quite sure where to start on the project. My approach was developed as a trial and error, but once I started with the basic steps of doing the truth table and creating the Karnaugh Maps, it was much easier to see how to proceed with building the actual circuit. Once I started building and testing my first circuit, I was able to see how each circuit was supposed to work and it made much more sense how the circuit works and how to proceed.

I still feel like my understanding of JK flip flops is a little shaky and something I hope to shore up before the exam next week. Overall, I thought this was a fun project and taught me how to problem solve when it comes to building circuits. I think if I start with a truth table and then translate it into a KMap, I feel pretty confident I would be able to build and wire up the circuit correctly. It's also very interesting to see how this translates into a real life scenario and the circuits that make up the calculators that we use on a daily basis.