Daniel Williamson

Project #1

2/12/16

CS 200

Project Overview

Purpose:

The purpose of this project was to simply create a circuit that could add two 3-but numbers together. Which is also known as a 3-bit adder. In the 3 bit-adder we can only use AND, OR and NOT gates.

Approach:

In the beginning of our approach we made a truth table to depict our needed values. Once we completed the truth table, we used it to create our Karnaugh Maps. We had two separate K-maps, one for the Sum and one for the Carry OUT. With those two K-maps we created two Boolean functions. Once we had our two functions it was just a matter of creating them in Logisim with circuits.

Results

These are my results:

This is a depiction of my Truth Table:

Without both outputs (Sum, Carry OUT) we would not be able to calculate the carry.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Inputs** | | | **Outputs** | |
| X | Y | Carry IN | Sum | Carry OUT |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

This is a depiction of my K-map for the Sum:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sum** | | | | | |
| **XY** | | | | | |
| Carry IN |  | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

Unfortunately we cannot make any pairs so we end up with a Boolean function looking like this:

Fsum = X’Y’Cin + X’YCin’ + XYCin + XY’Cin’

This is a depiction of my K-map for the Carry OUT

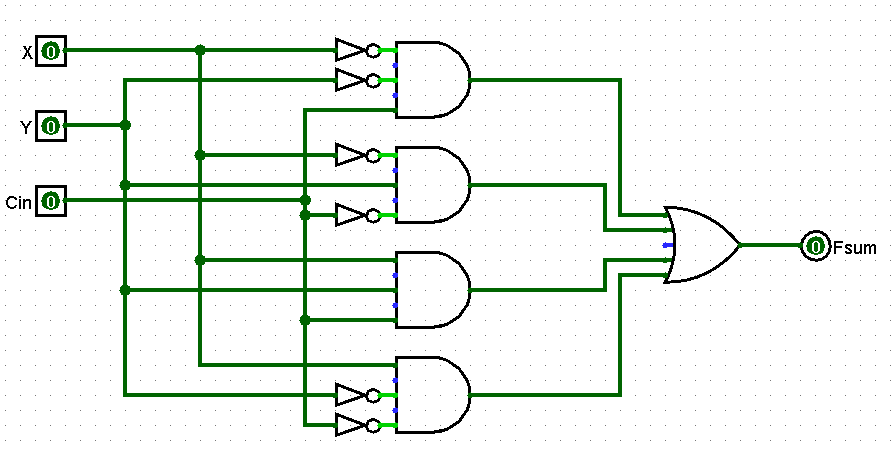
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Carry OUT** | | | | | |
| **XY** | | | | | |
| Carry IN |  | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

This time I was able to create three pairs reducing the Boolean function to look like this:

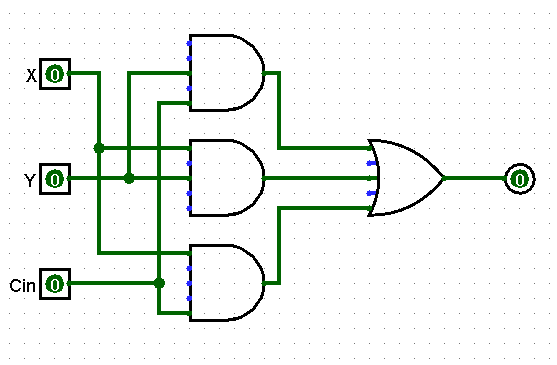
FCout = YCin + XY + XCin

Once I had both reduced Boolean functions I created circuits from both.

This is a depiction of the completed circuit for Fsum:

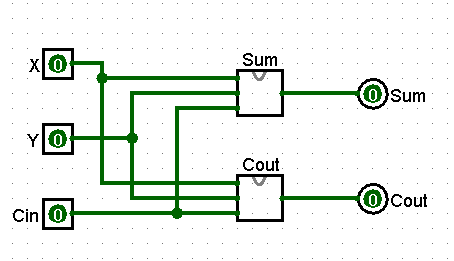


This is a depiction of the completed circuit for FCout:



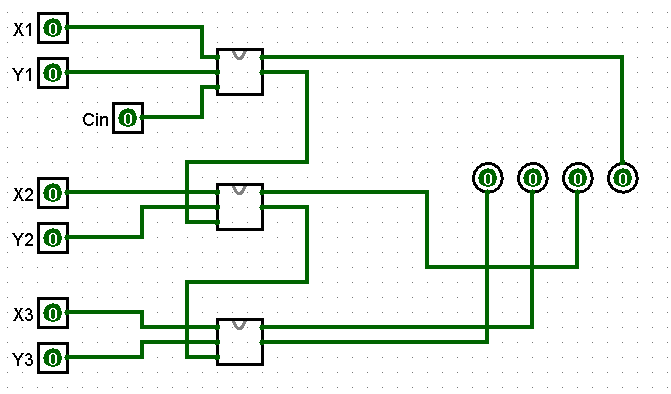
After completing both the Sum circuit and Carry OUT circuit I combined them both to make our Full Adder.

This is a depiction of the completed circuit for the Full Adder:



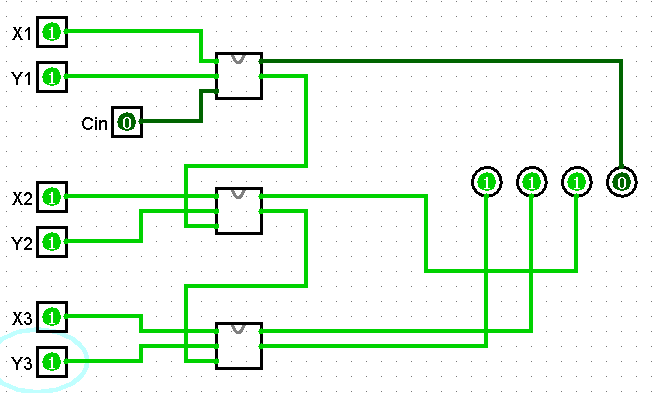
And finally once my Full Adder was complete I was able to simply copy it 3 times to make a 3 bit counter.

This is a depiction of the completed 3 bit counter:



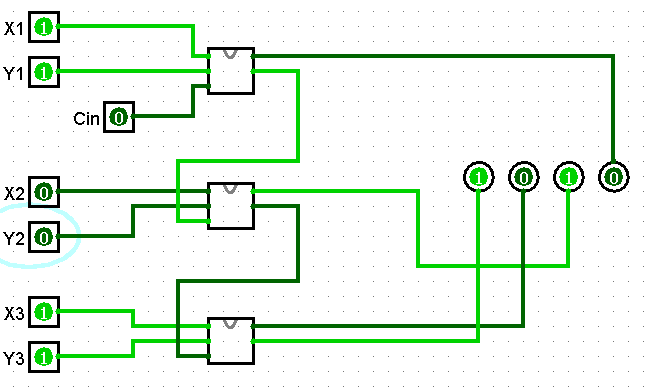
After completion of my 3-bit counter I put it through some testing. Knowing that if I were to add 111+111 it should return to me a binary value of 1110

Which it did:



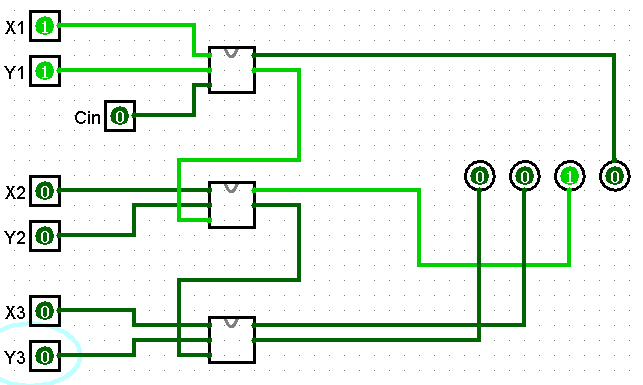
Knowing that 101+101 should return the binary value of 1010.

Which it did:



And lastly, knowing that 001+001 should return the binary value of 0010.

Which it did:



Conclusion

In conclusion this was an extremely interesting and insightful project. This project taught me how adding binary numbers work using AND, OR and NOT gates. It is pretty amazing how a simple circuit like the one I created can be so educational. There were a few problems I encountered in the beginning. For instance, I was not combining my Sum and Carry OUT circuits to make the actual Full Adder, instead I was attempting to have them in the final design next to each other. Making the project very messy with tons of inputs. A pier pointed out combining them.