**CS 200**

**Project 1: 3-bit Adder**

Brandon Horner

**February 08, 2013**

**Project Overview**

**Purpose**

The purpose of this project is to create a 3-bit adder, which will add two 3-bit Boolean inputs.

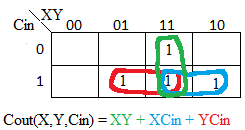
**Approach**

The approach started with drawing a truth table for the inputs X, Y and Cin and outputs Sum and Cout.

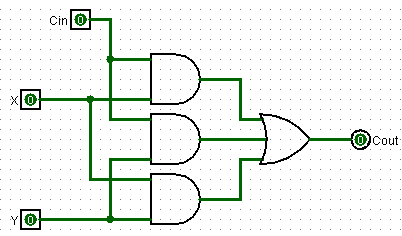
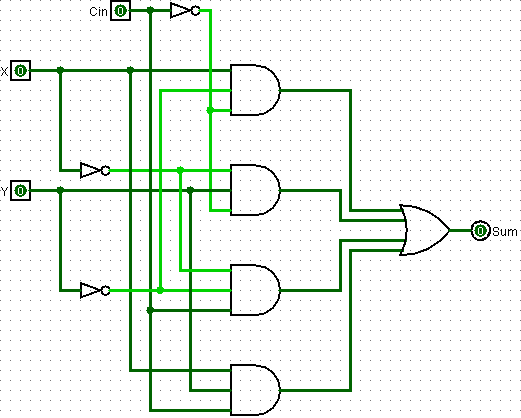
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | | | Output | |
| X | Y | Cin | Sum | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

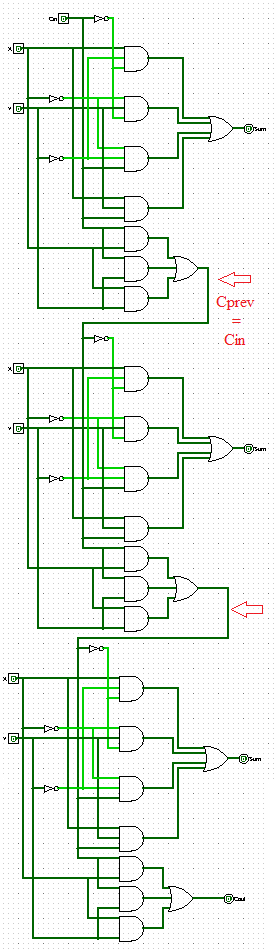
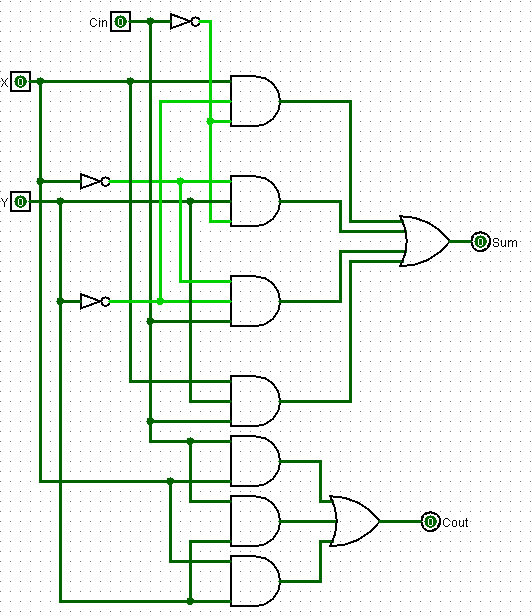
From the truth table I made the following Boolean expression:

Cout(X,Y,Cin) = (!XYCin) + (X!YCin) + (XY!Cin) + (XYCin)

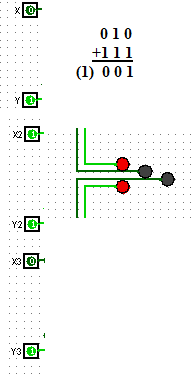
Using a Karnaugh map I simplified the Cout expression:

Sum(X,Y,Cin) = (X!Y!Cin) + (!XY!Cin) + (!X!YCin) + (XYCin)

I then created a circuit for each function:   
Cout(X,Y,Cin) = Sum(X,Y,Cin) =

Once I made the two circuits, I combined them to produce two outputs (left):

I proceeded to attempt transforming this circuit from a 1-bit adder to a 3-bit adder (on right). I decided the only way was to force the Carry in for the next circuit to be the Carry out of the previous. (Cout[next] = Cin[prev])

Finally, I tested the circuit with LED lights by connecting them to the outputs. The LED lights are in order from Sum1 to Cout.

The input,

x = 0

x2 = 1

x3 = 0

y = 1

y2 = 1

y3 = 1

Gives the output,

Sum1 = 1

Sum2 = 0

Sum3 = 0

Cout = 1

**Conclusion**

I learned a lot from this project. I did not play around with Logisim before attempting this, so it was a little hard to pick up right away. Learning AND/OR/NOT gates with simple expressions was easy, but once I had to create a full-adder with carries, I was lost. I stuck it out and attempted to create the 3-bit full-adder, which (as far as I can tell) turned out to work. Once I finished the project by testing it with LED lights, I finally comprehended what I was actually doing with these bits, and how the carries functioned. Overall, I feel like I can now grasp the concepts in this project with confidence.