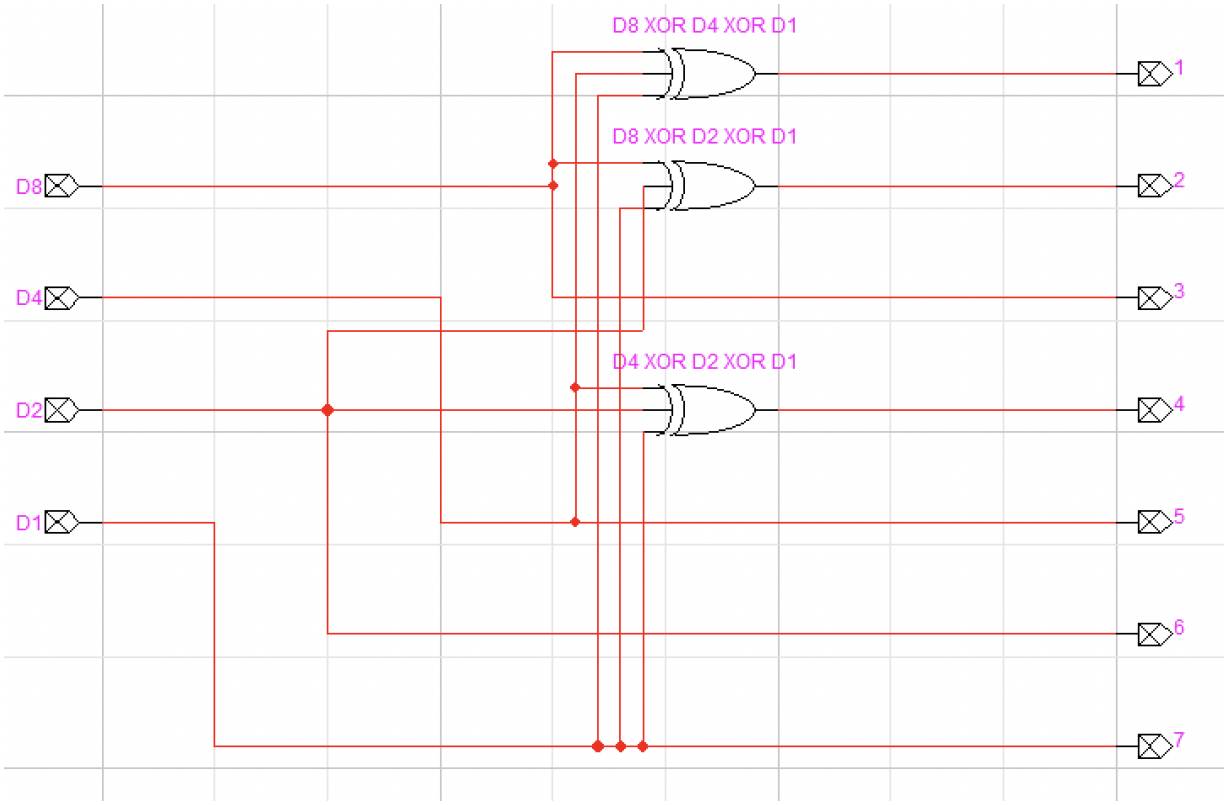
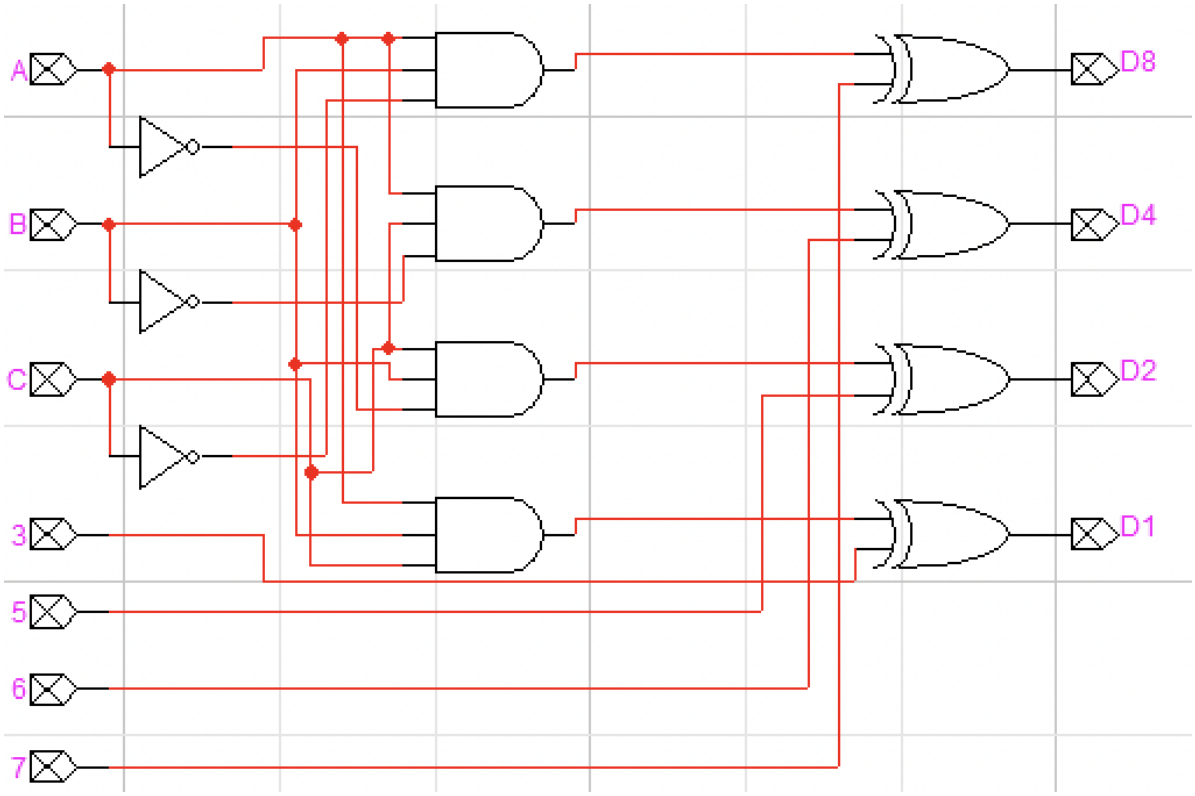
Benny Chen
3/21/2022
Lab 6

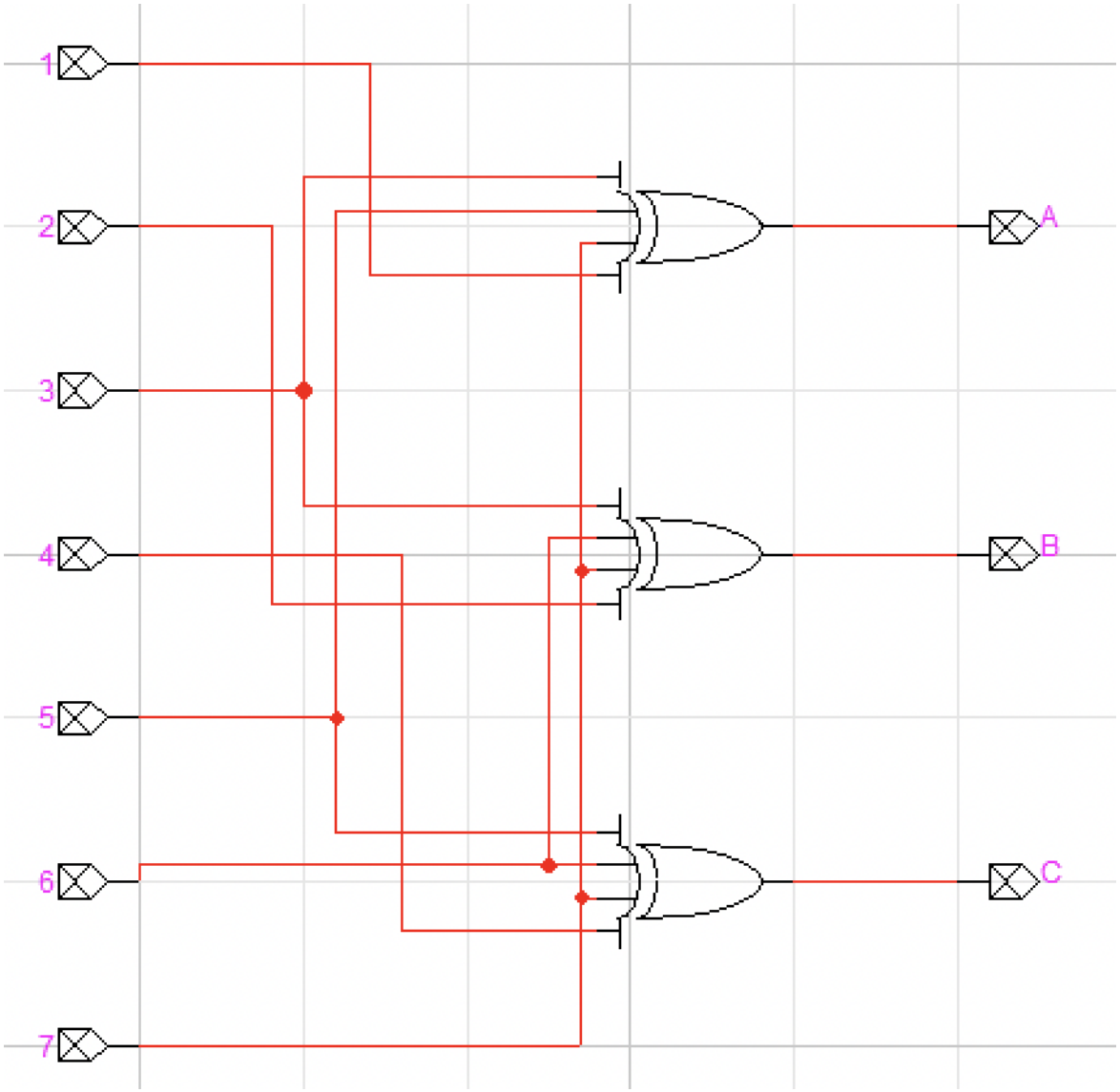
# Theory

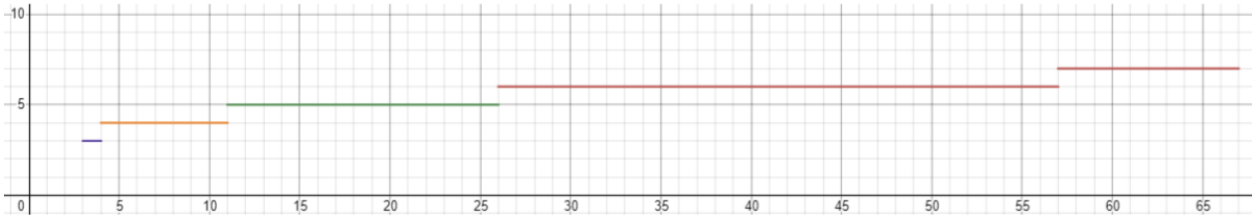
In this lab we learned more about and implemented hamming codes. We created a parity bit generator first for all 7 bits of a hamming code. We then created an error locator and corrector for a hamming code. We created a parity generator by XORing the given bits with the given parity bit formula. For the error locator we then use the given check code formulas to XOR the given bits. For the error corrector, we check for the error position given the locator then XOR it with the other given bits to correct that position. Hamming distance is the number of positions, comparing 2 hamming codes, that are different. For example for 1111111 and 1111110, the hamming distance would be 1 as the only difference is just one bit. For this lab we only are able to detect one error, but to detect two we just simply need to add another parity bit that encompasses the whole hamming code to cover another error. With this the correction logic and everything else would be the same.

# Deliverables




A
B
C
3
5
6
7
D8
D4
D2
D1



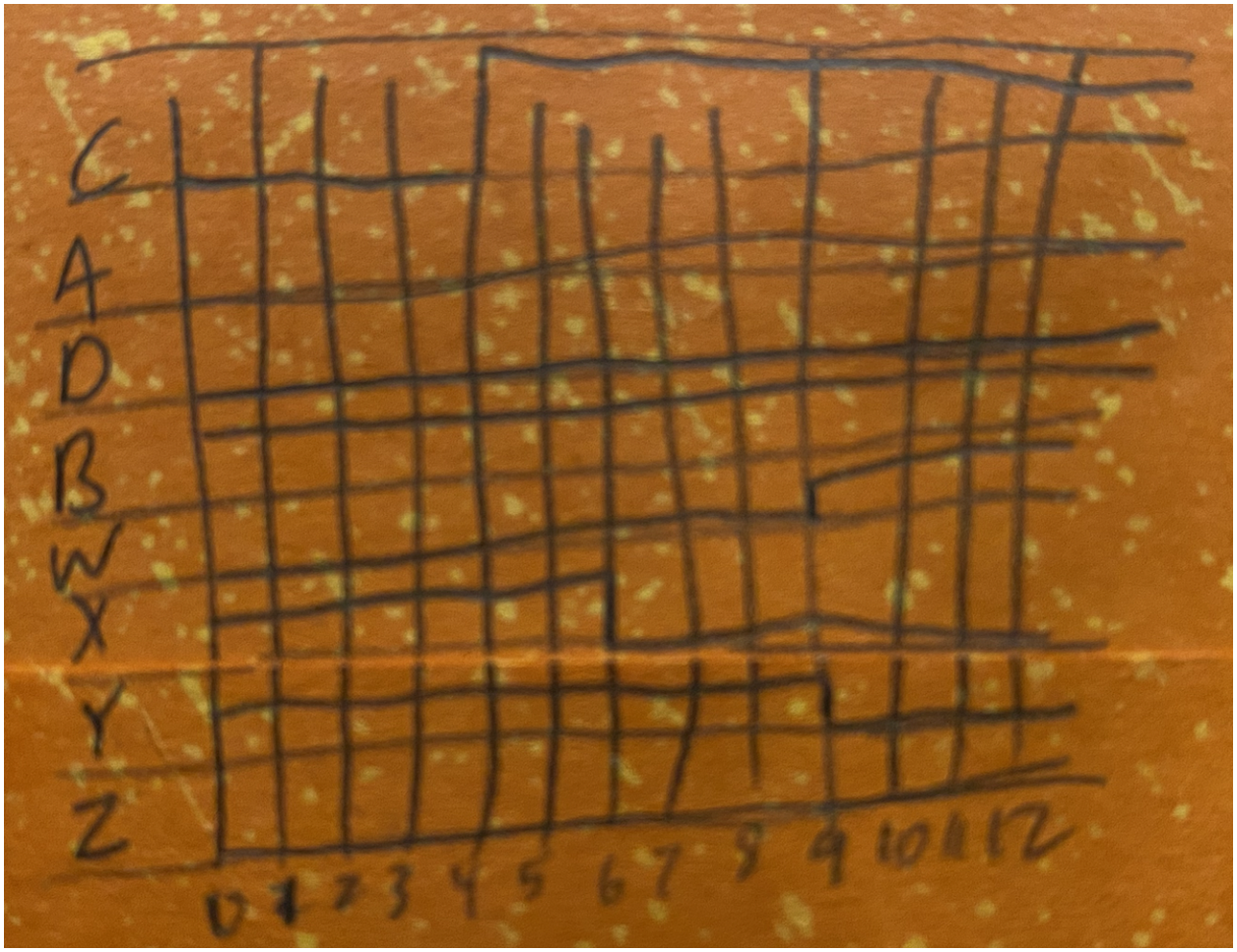

Y-axis = Number of parity bits required
X-axis = Number of data bits
Equation = $2^y \geq y+x+1$

# Discussion

In this lab we implemented and corrected hamming codes. This lab was pretty straightforward, the only challenging part was figuring out the equation for the graph and finding the bits from 4 - 67. Overall this was a fun lab and reinforced how to create symbols in LogicWorks.

# Questions

1. **The circuit below consists of gates in which each INV/NOT has a delay of 3 ns. The OR and AND gates have 5 ns delays. Given an initial condition of A=0, B=1, C=0, and D=0, complete a timing diagram where C becomes 1 at 4 ns. Assume transport delay, and that the initial W, X, Y, and Z values are based on initial A, B, C, and D conditions.**




**Yes I did this on a parking ticket**

2. **Complete the Hamming code below for the given 4-bit hex values. Insert the EVEN parity bits into positions 1, 2, and 4. For this problem, P means POSITION, not parity.**

| **HEX:** | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** |
|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| **A** | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

3. **Write a logic expression that generates an ODD parity bit from four data bits A, B, C, and D. Optimal solutions only require XOR and NOT gates.**

(NOT (A XOR B XOR C XOR D) )

4. **There is an error in the following 7-bit Hamming code with EVEN parity bits**

| P1 | P2 | P3 | P4 | P5 | P6 | P7 |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |

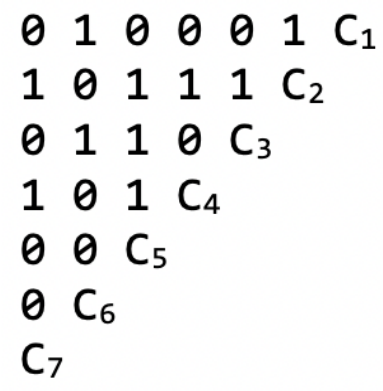
   a) **What is the position of the erroneous bit?**

**P3**

   b) **What is the correct code?**

**0111100**

5. **Find the values for all the check/parity bits in the following ODD triangular code.**




**C1 = 1**
**C2 = 0**
**C3 = 0**
**C4 = 0**
**C5 = 0**
**C6 = 1**
**C7 = 1**