Kyle Powers
Assignment 2
Section 107
2/28/11

Introduction

The purpose of this laboratory assignment was to perform a complete combinational logic design using only NAND and NOR gates. A written description of the problem was given. From there, a combinational logic circuit was designed to realize the function using AND and OR gates and inverters. Then, the design was refined to only use NAND and NOR gates. Lastly, a plan was constructed to properly test the refined design.

Problem Statement

The problem in laboratory assignment 2 was to design a 4-to-1 multiplexer using only NAND and NOR gates.

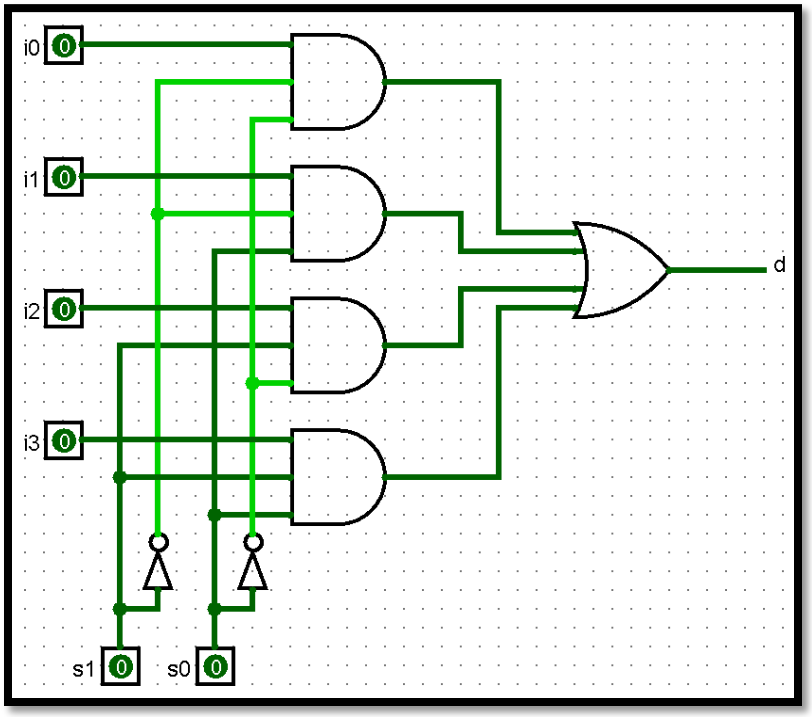
Solution

The first step in solving the problem was to create a function table with an accompanying Boolean equation:

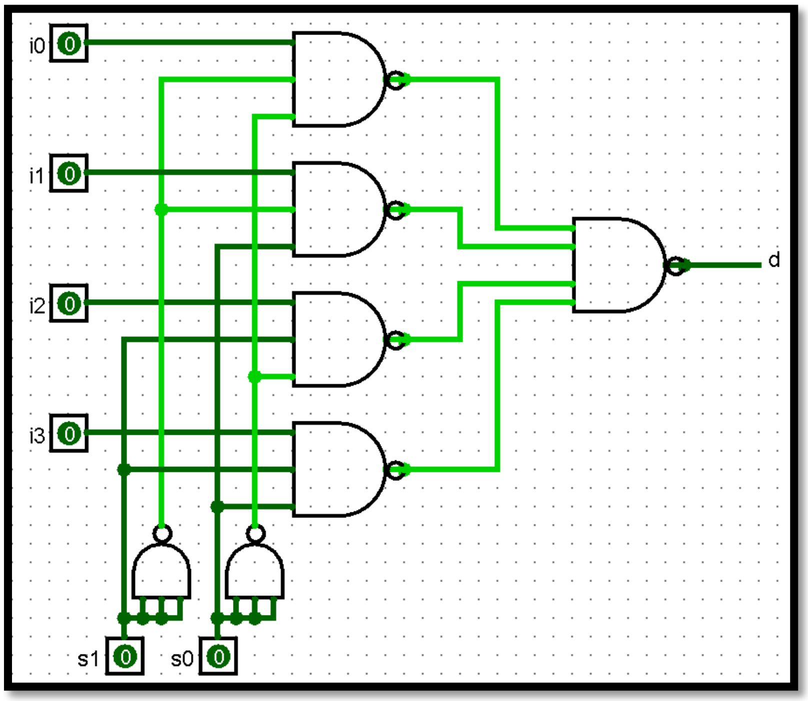
d = (i0·s1'·s0') + (i1·s1'·s0) + (i2·s1·s0') + (i3·s1·s0)

s1	**s0**	**d**
0	0	i0
0	1	i1
1	0	i2
1	1	i3

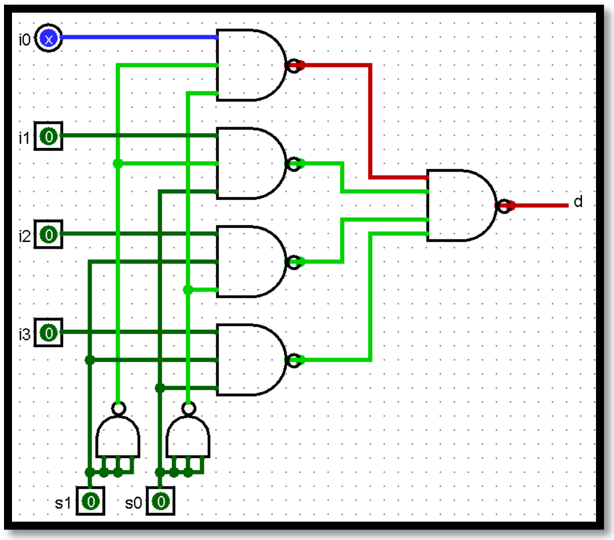
The second step was to design a combinational logic circuit to realize the 6-input 1-output function using only the standard approach that uses AND and OR gates and inverters.

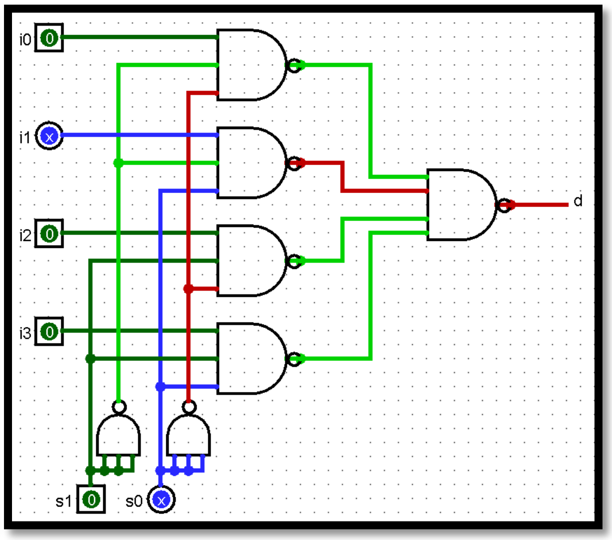


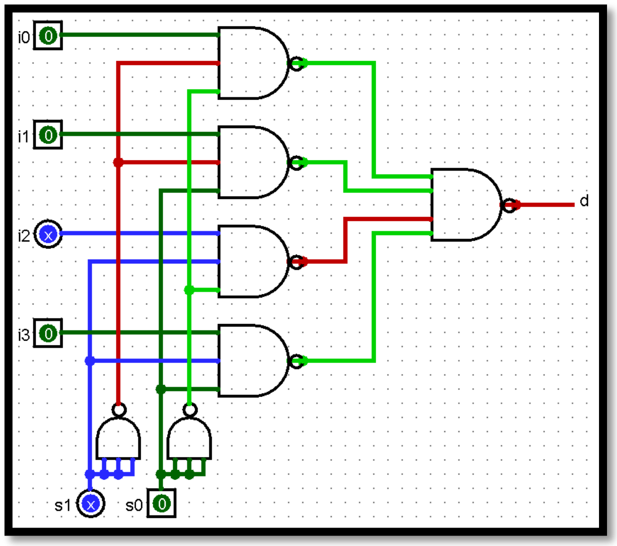
The third step was to refine the design so that it uses NAND and NOR gates only. This was done by double negating the OR gate, turning it into a NAND, turning the AND gates into NAND gates, and replacing the inverters with NAND gates.

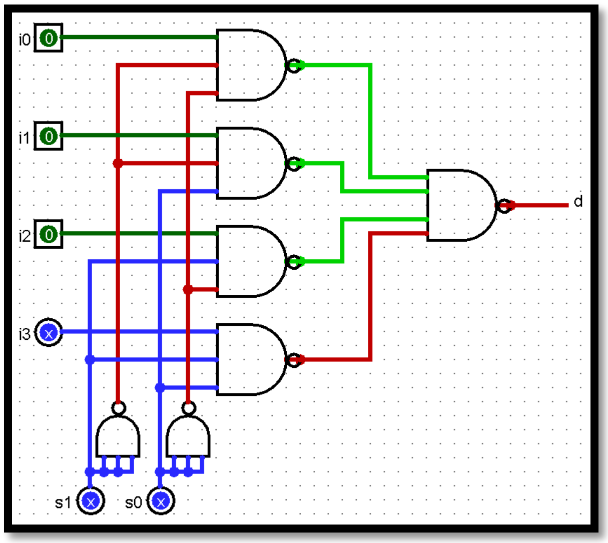


The last step was to design a test plan to demonstrate that the circuit works. This was done by first connecting the output **d** to an LED indicator (red when high, green when low). Then, by switching i0, i1, i2, and i3 on with their respective s1 and s0 inputs (as seen in the aforementioned function table), the circuit was able to be proven successful.







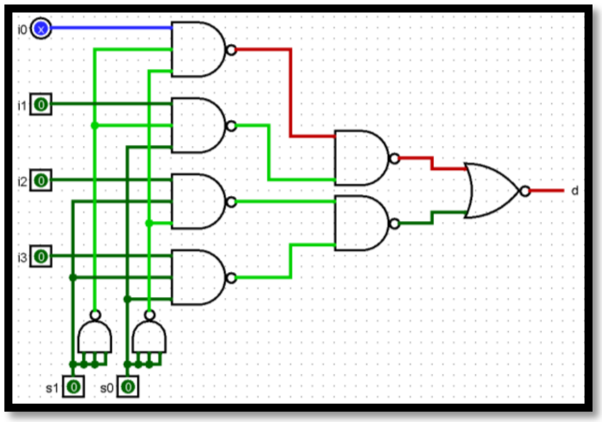


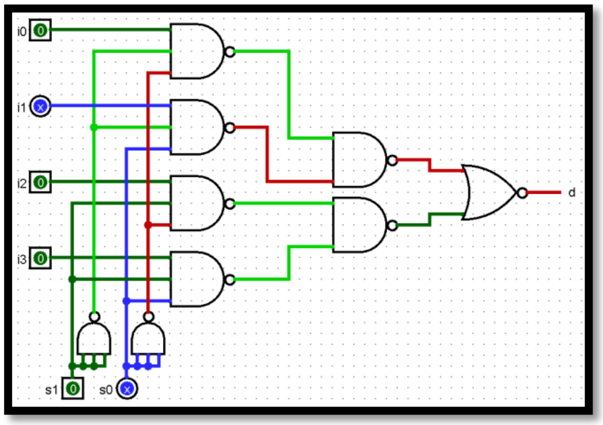
Problems Encountered
The only problem encountered was how complicating the wiring became. This was resolved by starting over, slowly wiring from one point to another, while checking off its completion on my pre-laboratory diagram.

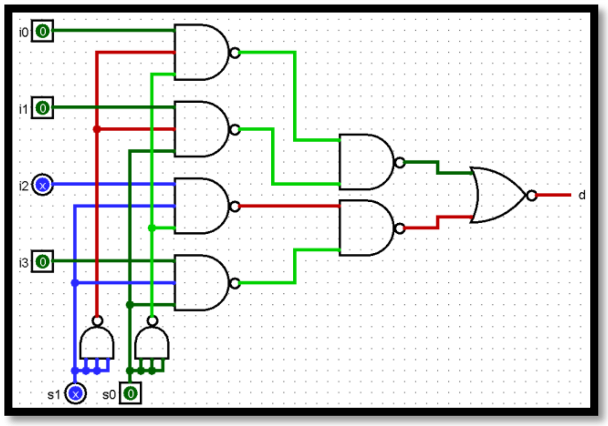
Conclusion
In the end, the 6-input 1-output function proved to be a successful 4-to-1 multiplexer. While the lab took about an hour, due to the complications of wiring and attempting to use as little NAND packages (7410) as possible, the use of Logisim to test the revised circuit diagram prior to lab proved to be an effective pre-laboratory decision. Laboratory assignment 2 ultimately enhanced my understanding of multiplexers.

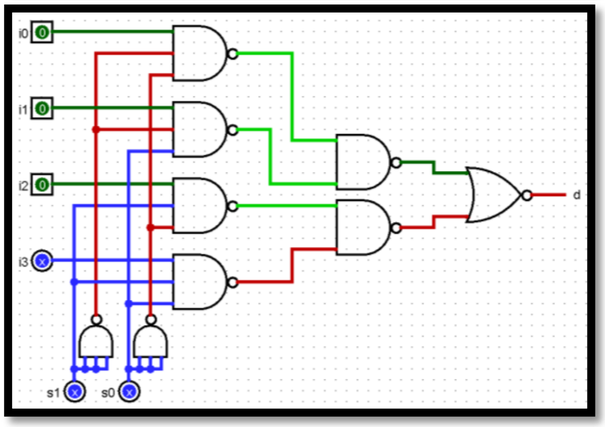
Questions

1) If your design used the 4-input NAND gate, how would your design change if this 4-input gate were not available? Be specific with your design changes and demonstrate that your new design works (using algebra to derive the output and show it's equivalence to the desired function).
A replacement for a 4-input NAND gate would be 2 2-input NAND gates NOR'd together. One of the NAND gates would output i0, i1, i2, or i3, while the other NAND gate outputs a 0. Then, the NOR gate – receiving an input of 1 and 0 – outputs the 1 (which of course is i0, i1, i2, or i3). Starting with the initial Boolean equation: **d = (i0·s1'·s0') + (i1·s1'·s0) + (i2·s1·s0') + (i3·s1·s0)**, the equation would then become **d = [(i0·s1'·s0') + (i1·s1'·s0)][(i2·s1·s0') + (i3·s1·s0)]**.



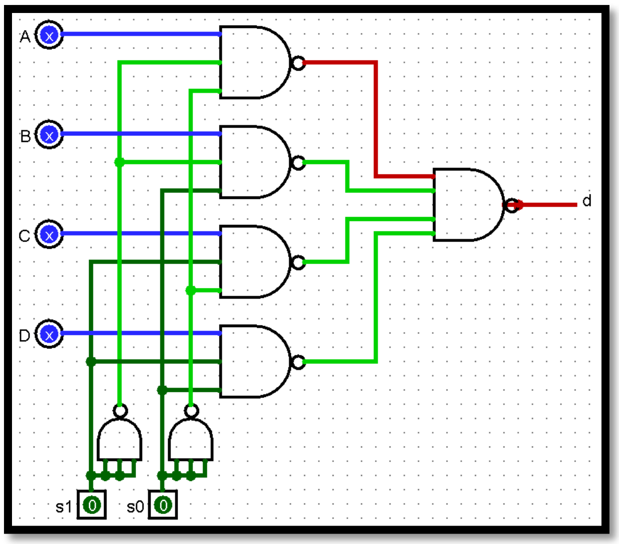


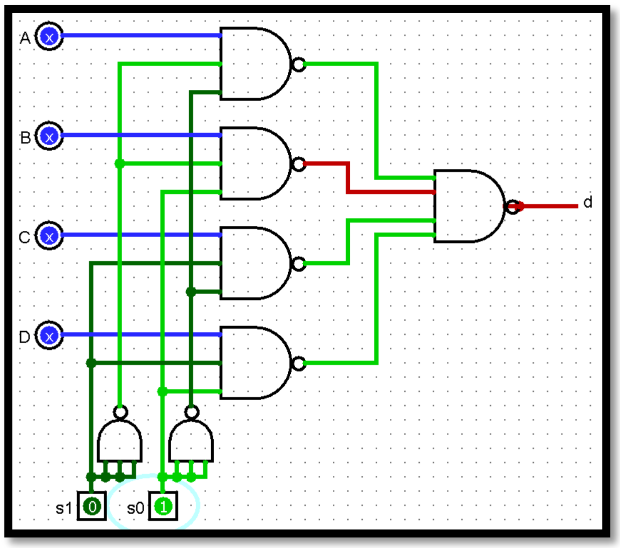


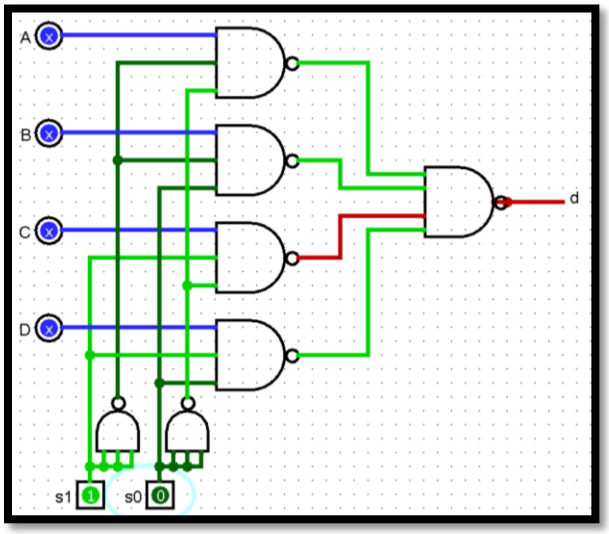


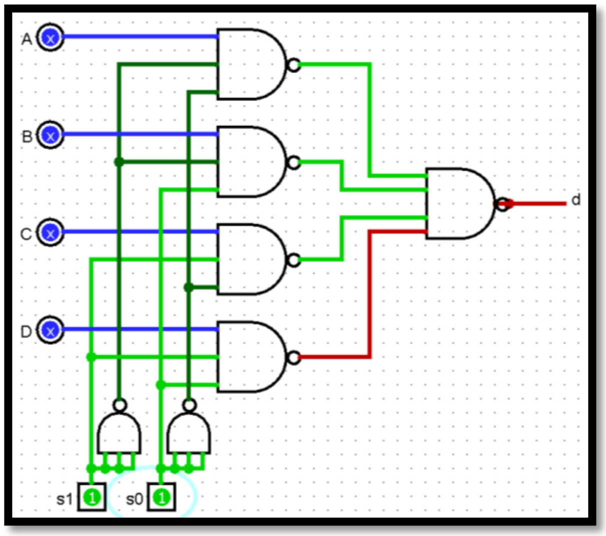
3) Show how to use your 4-to-1 multiplexer to implement the function you implemented in lab 1. You do not have to build it, simply draw the logic circuit with clearly labeled inputs and outputs. Make sure that the order of the input variables is clearly specified. Demonstrate that your circuit works (using algebra to derive the output and show it's equivalence to the desired function).

Replacing **d = (i0·s1'·s0') + (i1·s1'·s0) + (i2·s1·s0') + (i3·s1·s0)** with **d = (A·s1'·s0') + (B·s1'·s0) + (C·s1·s0') + (D·s1·s0)** yields the following results:









On my honor as a student of the University of Virginia, I have neither given nor received aid on this assignment.

Kyle Powers

Kyle Powers