

Gabriel Emerson

ELEC 2210 – T 11:00

Lab Final

04/13/2021

Introduction

The goal of this lab was to design a circuit that would accept a 4-bit code and determine whether the entry was correct. While waiting for the input, an LED would be blinking. If the code was correct, the waiting LED would turn off, while a fan and a separate LED would turn on. If it was incorrect, the waiting LED would turn off and an alarm would sound. The circuit also had to be able to reset at any time and return to the waiting stage.

Design

I began with a logic level diagram so that I could better visualize the project scheme. I started with the waiting stage, as it was the easiest. I realized that I would need a clock signal to alternate the blinking stage but would also need to be able to sense a correct or incorrect signal to cease the blinking. My solution to this was to use an AND gate with one input being the function generator and the other coming from a NOR gate carrying the correct/incorrect signal. I chose a NOR because if either signal was high, the NOR gate would send a logic low to the AND gate, thus turning off the blinking LED until it was reset.

Next, I decided to do the correct signal. The key to this phase was recognizing the need to keep up with how many bits had been entered. It was imperative to not trigger any signal until the fourth bit was entered. Therefore, I knew I would need a counter. I also had to come up with a way to enter the code bit by bit. The obvious solution to this was using a shift register as it can shift a previously entered bit over on command if you tie its activation to a manual clock pulse. Lastly, I used four XNOR gates to compare the values from the Digital Writer to the input values stored in the shift register. These gates were perfect because they only output a one if both inputs are the same. This signal would then be put into a quad AND gate with the 4th bit output from the counter, which is only a one on the fourth pulse. When all the input code matched the passcode on the fourth pulse, this would send a logic high to the “Correct LED” and turn on the fan.

The last phase to be designed was the incorrect phase. This was tricky because you had to have a frequency with a high enough amplitude to trigger an audible sound. To do this, you had to use another function generator. However, you only want it to make sound when the code is incorrect, so I put the signal into an AND gate. This way, it would only trigger if another signal was there as well. This other signal came from an AND gate with signals coming from the counter and from an inverted output of the quad AND gate. Thus, if the code was incorrect, a logic one would be sent to the first gate, which would send another logic one to satisfy the function generator AND gate. This would meet all the necessary conditions and create an audible sound.

The final thing I did to the design was compare my gates to the parts list. I did not want to have ten or more IC's to wire up so I began counting how many I would need. It was too much, so I began simplifying my circuit design using parts I knew we had in the lab. For example, I initially started with four XORs and four inverters to achieve my design. However, that could have been achieved using four XNORs. I also realized we did not have a quad input AND gate, but we did have a quad NAND. So instead of using the 2 input AND gates, I used one NAND and one inverter. I did things like this and narrowed my design down to seven chips. Then, I created a package level design with those chips in MultiSim to save time in the lab. Once the designs were completed, I reverified their functionality and sent the files to the TA.

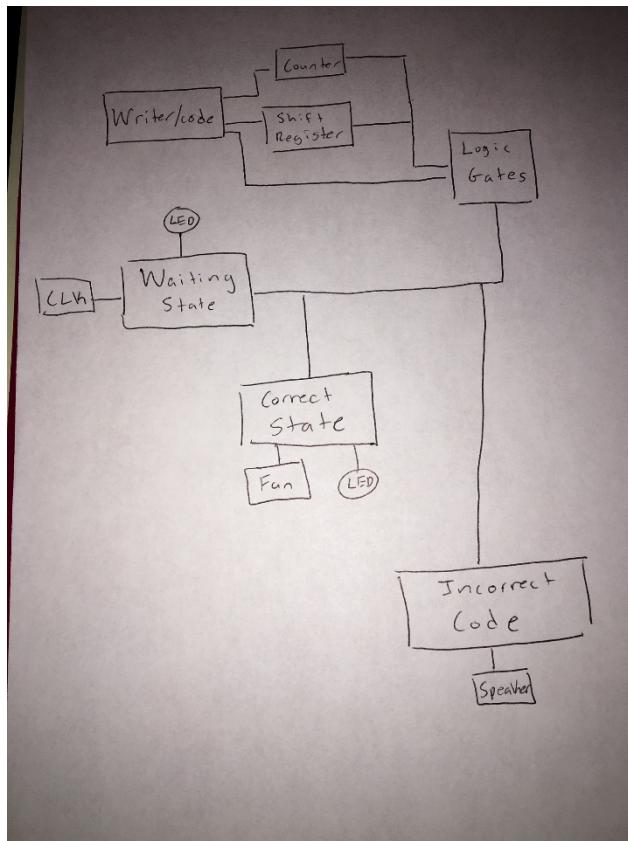


Figure 1: Block Diagram

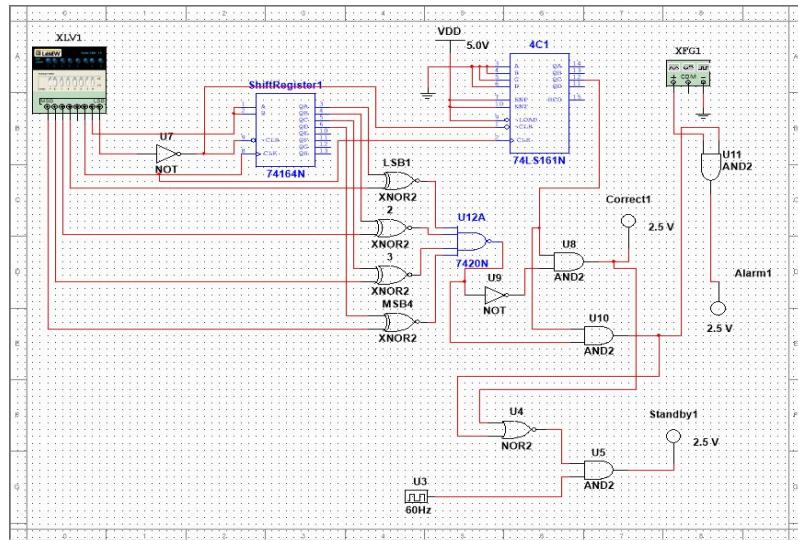


Figure 2: Logic Gate Level Design

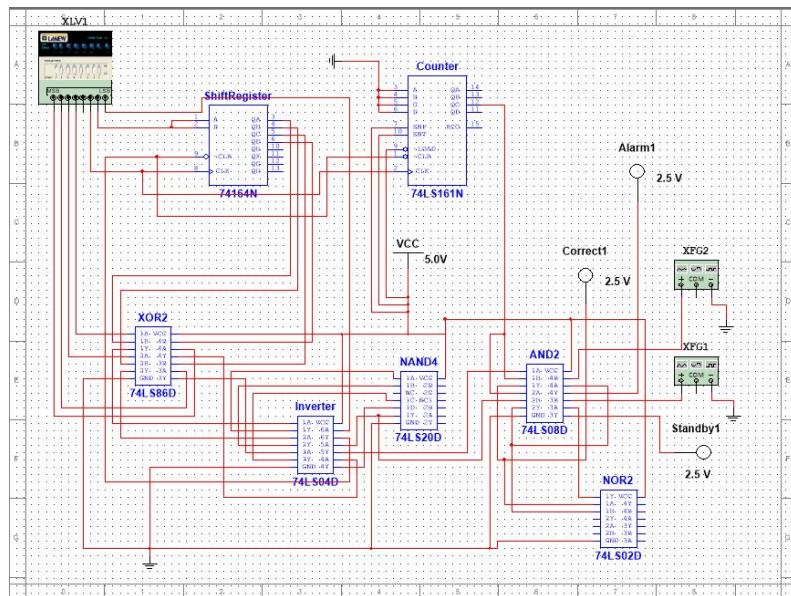


Figure 3: Package Level Design

Conclusion:

After about 3 hours of lab time, I was unable to complete my circuit. I had picked it up and moved a few components, and changed a few parts, but was unable to debug the system in time. I had gotten closer by swapping a few parts, the voltages were switching like they were supposed to, but not to the level at which the logic required. I believe the error was likely due to either a short on the board, or a broken part. I included below my final circuit before my 3 hours had ended.

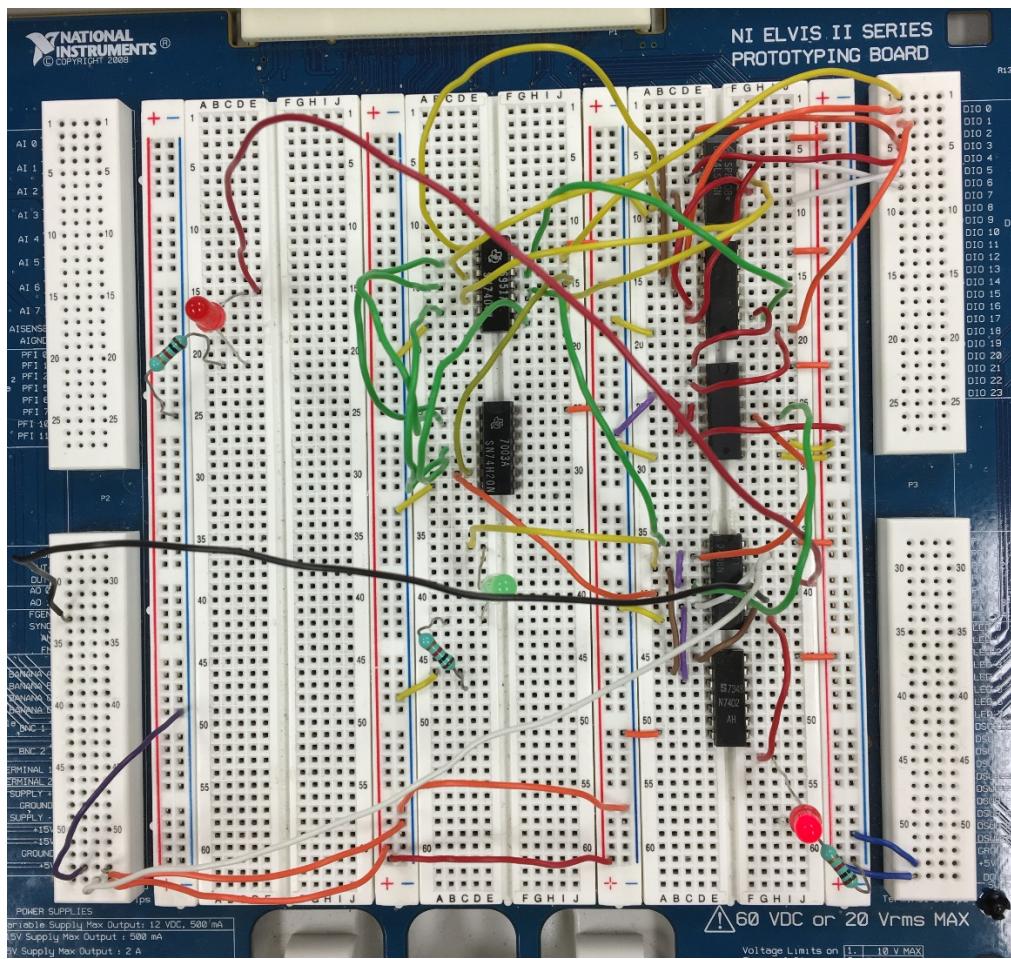


Figure 4: Final Circuit

ELEC 2210 Lab Checklist

Student Name Gabriel Emerson
Meeting Date & Time T 11:00 GTA Name Paul Atilola
Section # B01 Station # _____
Meeting # & Title Lab Final

Student Instructions: Fill in the items to be checked off by the GTA. When you are ready for checking off, notify the GTA. Include this sheet in your lab report.

GTA Instructions: Initial the student activities as requested in the experiment. Include comments as appropriate.

Part 1 Final GTA Initials [Incomplete]

Comments (GTA / Student):

Part 2 _____ GTA Initials _____

Comments (GTA / Student):

Part 3 _____ GTA Initials _____

Comments (GTA / Student):

Part 4 _____ GTA Initials _____

Comments (GTA / Student):

Cleanup Inspection GTA Initials P.A >