Electronics Lab Final Project Report:
4-Digit Combination Lock using Logic Gates

Duncan Beauch
Will Newberg

Abstract

The goal of this project is to design a proof of concept for a combination lock using digital logic chips without a software interface. The circuit "unlocks" when the correct series of inputs are received by pressing and releasing physical buttons. The circuit's key code is set by the user. Then, after four correct inputs are received the circuit will turn on an LED showing that the code was correct signifying that the circuit has "unlocked".

Introduction

Locking mechanisms are important to personal privacy. While the most common type of passcodes we use today exist as online account logins, we also find mechanical locks such as door locks and keys. However, there are also keypad locks which mix both mechanical components and digital logic to lock or unlock systems on doors or safes.

This project seeks to implement a keypad lock in order to unlock a metaphorical door represented by an LED. This could instead be used to run a motor that moves a bolt or send a signal through electronics to indicate the lock has been unlocked. This project was constructed with strictly hardware components to show the complexity of physical implementation compared to software.

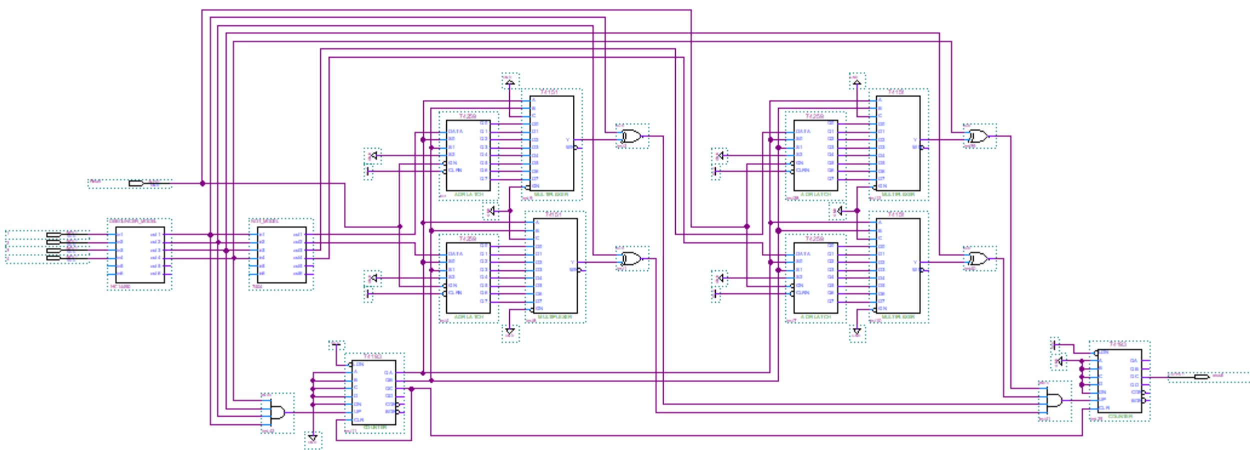
Parts List, Circuit Description, and Procedure

There are many different types of combination locks. Some can take a variable number of inputs and require an "Enter" key or keys might have to be pressed within a certain time frame. We wanted this system to be as user-friendly as possible and made several design decisions to account for this requirement. The following are basic for the lock usage:

1. A lock code consists of a series of four singular inputs
2. "Set" mode allows the user to set a new lock code
3. "Safe" mode allows the user to unlock the circuit once the correct lock code was received
4. The user can switch between "set" and "safe" mode with the flick of a switch
5. No reset switch is required, the lock automatically resets after receiving four inputs
6. No time restriction exists, a user can take as long as desired between inputs

Implementation of these assumptions proved more complex than anticipated, but the described functionality was of primary concern. The full circuit diagram for the project and parts list is below and more detailed portions described later.

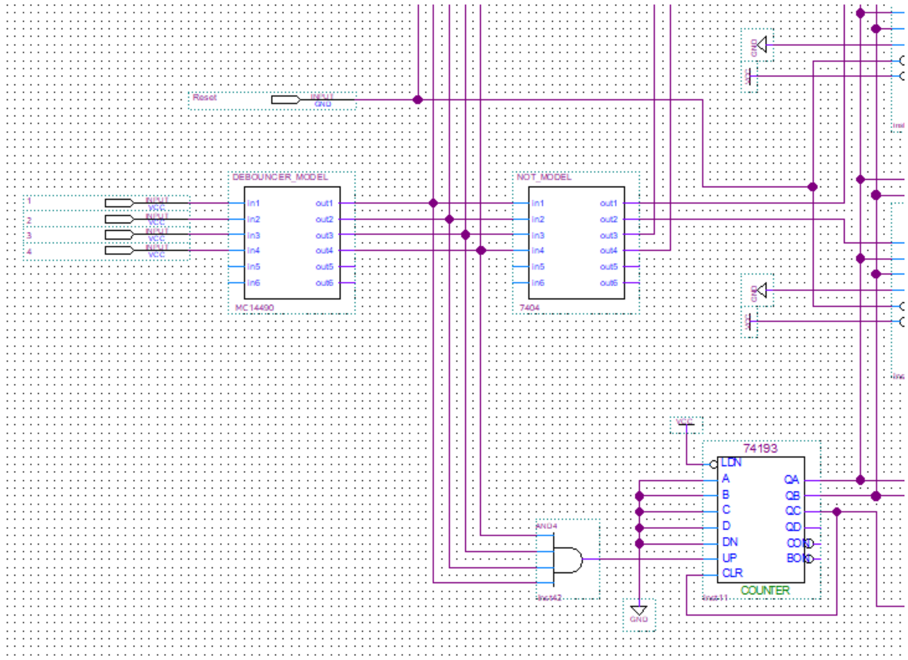
Full Circuit Diagram



Parts List:

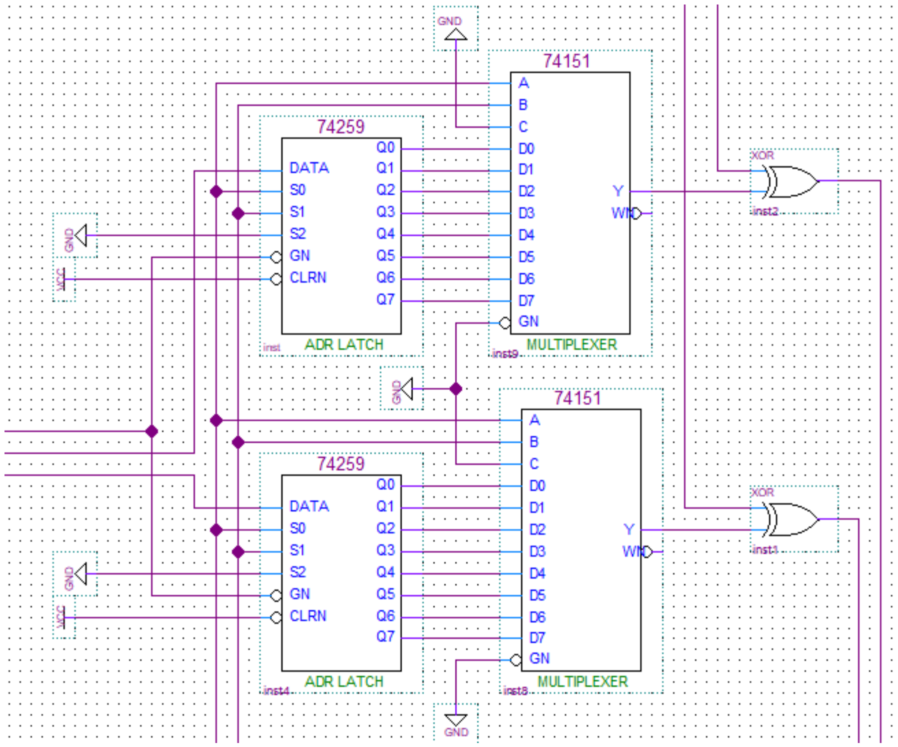
- Breadboard
- Miniature Push Buttons (x4)
- Toggle Switch
- MC14490 Debouncer Chip
- 5 nF Capacitor
- 1kΩ Resistors (x4)
- 74LS08 AND Chip (x2)
- 74193 Counter Chip (x2)
- IC 74151 Multiplexer (MUX) Chip (x4)
- 74259 Addressable Latch Chip (x4)
- 7486 XOR Chip
- SN7404 NOT Chip
- Inductors (x4)
- Jumper wires

Segment One: Buttons and Timing Signal



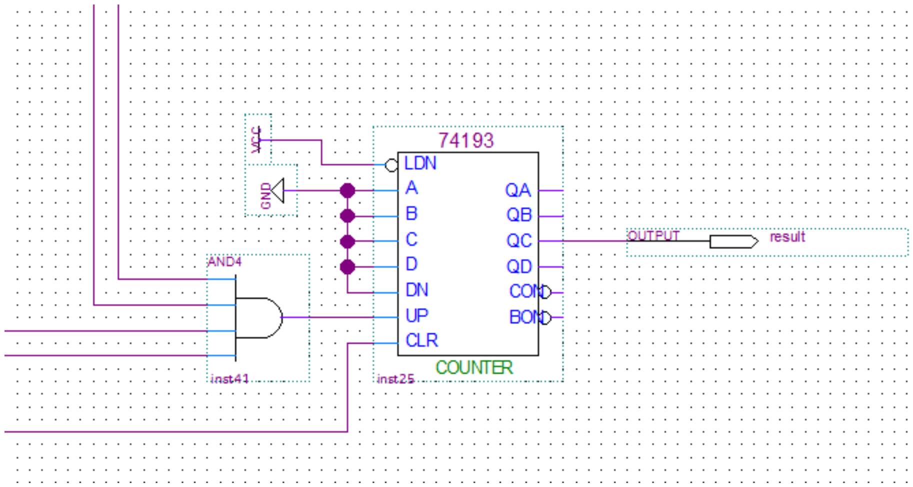
The circuit begins with the buttons labeled 1-4 and “set” switch on the left of segment one. The buttons are wired to 1V DC and when pressed open the flow of current which is fed to the debouncer chip to clean up the signal. Whenever a button is released, a rising edge signal is sent to the counter chip on the bottom right of segment one. This counter functions as a timing signal to keep track of where in the series of inputs the circuit currently is and resets every four rising edges. For example, if the counter is currently outputting 0 it means the first input is being read and when it receives a rising edge from a button press, it increments to 1 to designate the second input is currently active. The pin on the counter corresponding to the third binary digit Qc is wired to the clear pin which makes the counter reset every four button presses. The next section of the circuit to analyze is the code memory system implemented with four addressable latches and multiplexers shown in segment two below.

Segment Two: Latch/MUX Memory



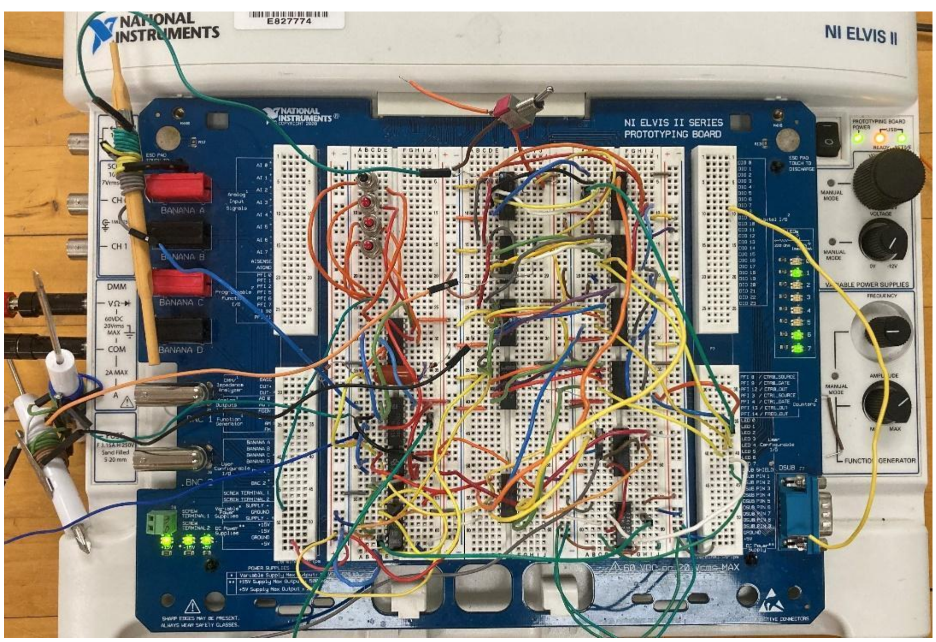
This snippet depicts two of the four addressable latch and multiplexer pairs. Each pair is identical. This system only functions when the circuit is in set mode as the latch's enable pin is wired to the set switch. Input to the data pin on the latch is received from the buttons where each latch/MUX pair corresponds to a single button, thus there are four pairs for four buttons. The latches are given the address of which location to write the data to based on the timing signal from the counter system. This sets the data received from the buttons to one of addressable latch outputs which connect to the MUX's input pins. The MUX selects its data based on the timing signal from the first counter the same way as the latch. When in safe mode, the addressable latch outputs will be constant and the MUXs will select which output to compare the signal from each respective button. Therefore, when the timing signal is at 0, the MUX outputs the first stored signal. This output signal from the MUX is compared with the NOTed signal from the buttons through an XOR. Thus, all four pairs output true only when both the stored signal and the button signal match. These four signals meet at the snippet shown below.

Segment Three: Final Counter and Output



The four compared signals from the latch/MUX pairs meet at an AND gate shown above. This AND gate then outputs true only when the signal from all four latch/MUX pairs matches the signal from the buttons. This AND gate then sends a rising edge to our final counter which has its clear pin wired to the same clear pin from the first counter. Therefore, all counters reset every four button presses. This final counter tallies how many consecutive button states matched the stored states in the MUXes. When the correct key code is input to the circuit via the buttons, the final counter will have counted to four at which point its third binary digit Qc will become true which is our final output to show that the circuit has been unlocked.

Implemented Circuit



Results and Conclusion

This project proved to be a successful prototype for a 4-digit combination lock following the designed circuit diagram. The implementation is displayed above where the final output is wired to LED 7 on the ELVIS breadboard and in the photo the correct key code was input which lit the LED. The most notable addition from the theoretical diagram to the implemented circuit is the use of inductors. Inductors were necessary to fix a logic race that occurred when trying to set the circuit. Before, when a button was released, ground would reach the addressable latch data input before a rising edge would hit the counter and the new timing signals could reach the addressable latch. This meant that each addressable latch always stored a zero, even if you tried to set that data location to a one. To solve this, small inductors and resistors were placed directly before the data input to the addressable latches. Since inductors oppose change, they held the signal for a couple nanoseconds longer, enough for the new timing signals to reach the addressable latch and have the proper data be stored. The inductors were made using wires wrapped around screwdrivers and adjusted until the delay was long enough to fix the issue.

While in theory the circuit is capable of receiving multiple button signals as one of the digit inputs, in practice the buttons prove ineffective and no more than two buttons can be pressed at once to both set and test a key code. If all four buttons are to be seen as binary digits, this lock would be capable of storing 65,536 (16^4) combinations. However, with its limited physical capacity it can only consistently function with 256 (4^4) combinations which still proves a powerful lock for a simple design. The circuit of course has several security concerns and could not be used as is for a realistic lock. A housing would need to be crafted to store the electronics and only the four buttons and output available to a user. The set switch would need to be reimagined so that only the owner of the lock would be able to set its stored key code.

Since this was a proof of concept rather than a working prototype the circuit output became more anticlimactic. Lighting up an LED simply showed that an output signal can be received and serve a purpose. Alternatively, the output could be wired to a motor which extends a bolt to lock a safe door. Further research and testing could be done to allow all four inputs to be used at once and a physical housing with functional output would improve upon this project.