Kara Kohutek
CSCI 3441.01
12/07/2021

Lab 8: Final Project

Prompt:

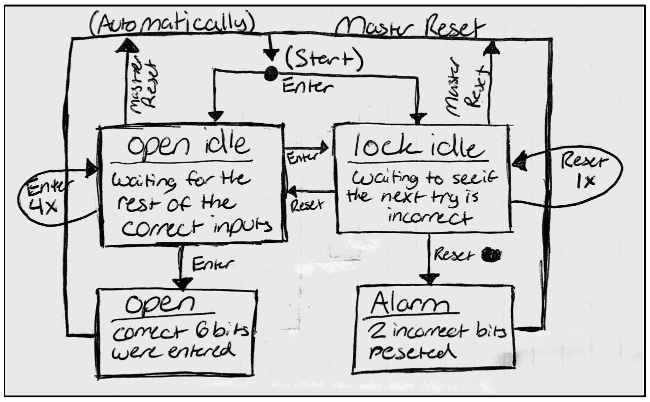
A brief description of the design process you used to obtain the circuits. Be sure to point out why you chose a particular design, along with the steps you took to minimize the amount of hardware required.

Assignment:

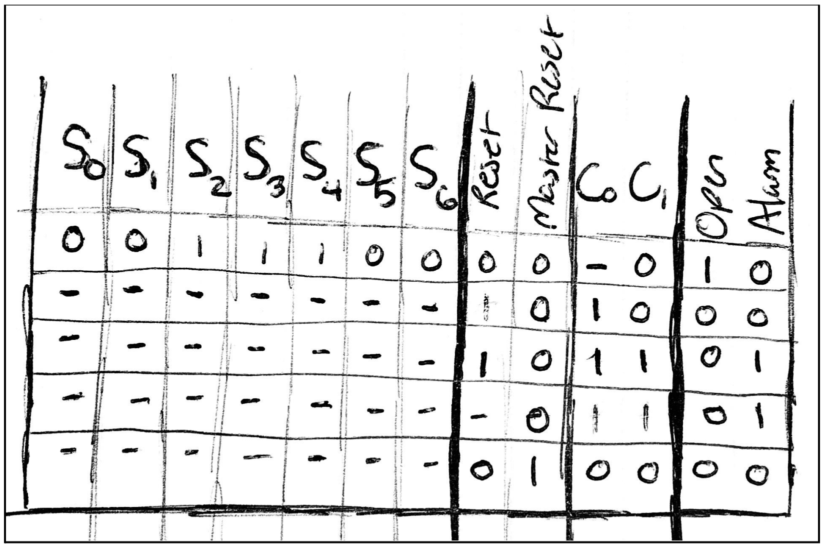
Immediately after reading over the assignment, I knew this is all about timing. Using clocks and different types of flip-flop methods was how this circuit was going to work. Although, trying to find the right type of flip-flop was one of the hardest parts I had. After looking at many examples and getting help from other students, I was able to understand that my D flip-flops can help hold the previous bit, and (like a key in a keyhole,) the bits fit into the last few AND gates how they were supposed to.

The reset button needed two different counts, one right after the other, in order for it to hold on to the bit for whenever the user resets it from an incorrect sequence. Because each count needs to have an on bit, I have an AND gate to combine them into the alarm light. It was also connected to the flip-flops to reset the bits to start a new sequence. Since the master reset button also needs to reset the bits when activated, I have an OR gate to let either one of them do the same thing. But in this case, the master reset button also needs to reset the alarm if it's activated. In order to do this, the master reset button resets the reset counts so that the alarm turns off. Lastly, the alarm light needed to disrupt any inputs that the user might try to put into it. By using a NOT gate and another AND gate at the beginning of the enter button, the button bit would automatically be interrupted by the alarm.

My state diagram below goes over the four states that the circuit would be in and how to get around each state. Creating this was a bit overwhelming at first, but helpful after slowing down and thinking about it.

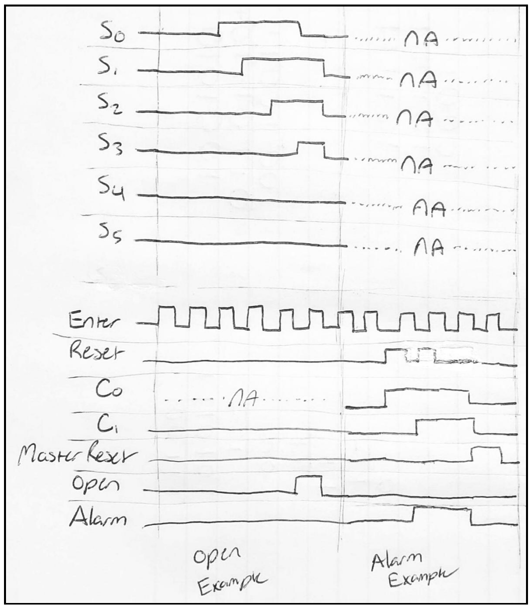


Next, is my simplified truth table. A good portion of this has dashes showing that it doesn't affect the end results if it was on or off. To rescue it as much as possible, I focused on the changes of time: when it's open, when the alarm is on and when it is reseted by both the reset and the master resetter. (I used S for the flip-flops and C for the counters.)

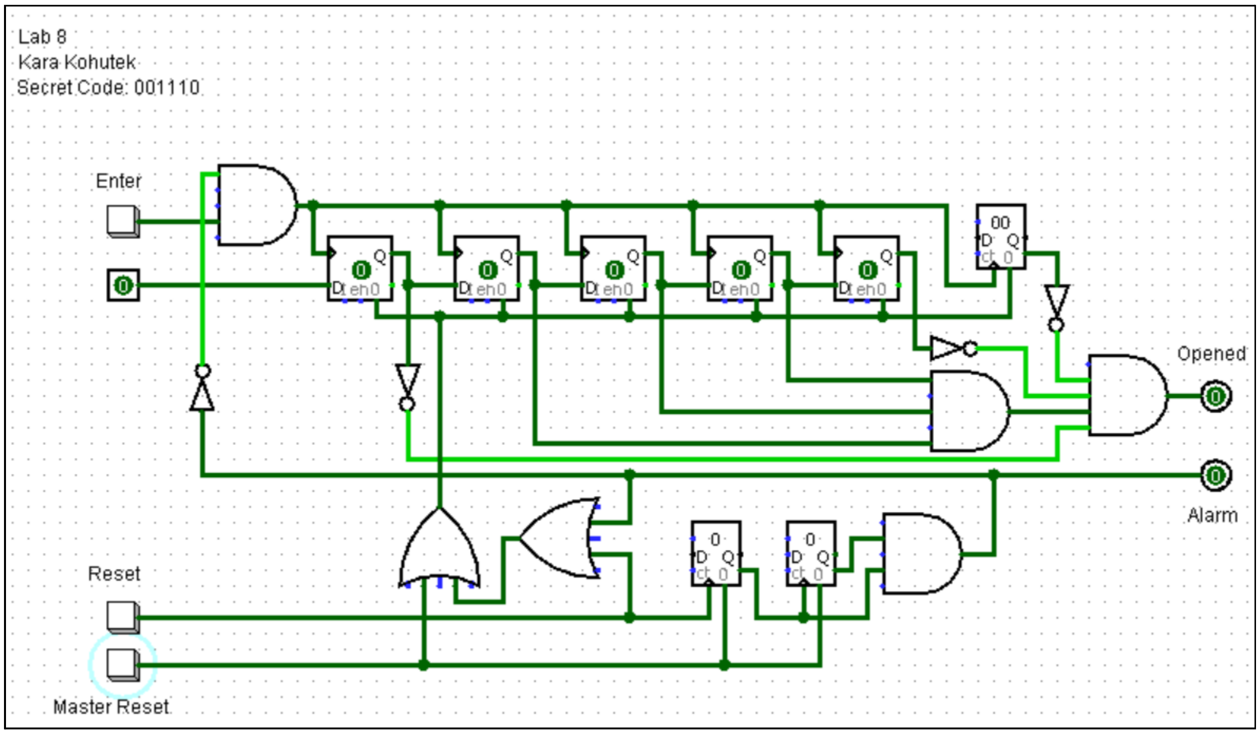


S_0	S_1	S_2	S_3	S_4	S_5	S_6	Reset	Master Reset	C_0	C_1	Open	Alarm
0	0	1	1	1	0	0	0	0	-	0	1	0
-	-	-	-	-	-	-	[illegible]	0	1	0	0	0
-	-	-	-	-	-	-	1	0	1	1	0	1
-	-	-	-	-	-	-	-	0	1	1	0	1
-	-	-	-	-	-	-	0	1	0	0	0	0

Lastly, I have my time sequence for the major events: when it's open, when the alarm is on, and when the alarm is turned off. Each S is dependent on the S before it from the last enter clock up motion. Both Cs are dependent on the resets, while the alarm is dependent on the Cs and the master reset button. (I used S for the flip-flops and C for the counters.)



Overall, this is the remaining image of my circuit that this information is supported by:


Lab 8
Kara Kohutek
Secret Code: 001110
Enter
Reset
Master Reset
Opened
Alarm