Lab 2 – Finite State Machine and Digital Design – Prelab Report

Sunday, February 2nd, 2025

Joey McIntyre (400520473), Devin Gao (400508489), Minhazur Rakin (400511143)

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is our own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario.

1. **Starting with a truth table, design a 4-bit combinational digital lock that opens only when the parallel inputs are 1010. State the Boolean expression and draw the schematic for this combinational circuit. When the digital combinational lock is open, the output of the circuit should be a logic 1 to turn on an LED. The LED should remain off until the lock is opened. Flowcharts are useful to describe procedural steps in a simple system, such as this one. While not required for this question, consider how a flowchart would be written for your parallel digital lock. [10]**

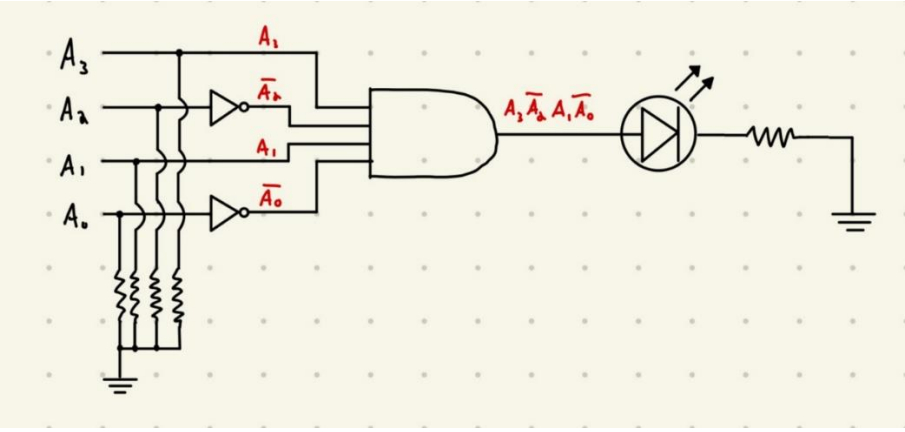
Truth table:

In - A3	In - A2	In - A1	In - A0	Output (X)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

(Only outputs high when the correct combination is inputted)

Boolean expression: $X = A3 * A2' * A1 * A0'$ - This Boolean expression indicates that the output is only high when A3 = 1. A2 = 0, A1 = 1, and A0 = 0.

Schematic:

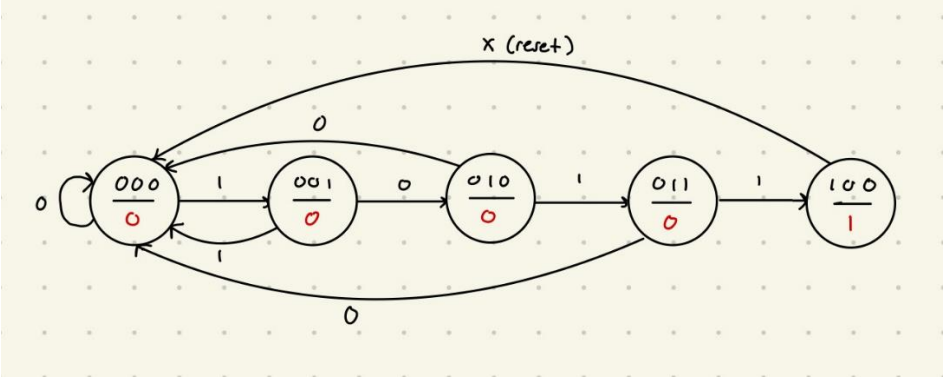


The schematic consists of 2 NOT gates, one 4 input AND gate, a LED to display output results, and resistors.

2. A better way to describe a more complex system is to use a finite state machine (FSM). Using the FSM approach, show the design of a 4-bit digital lock that opens only when the 1-bit serial sequential input is entered as 1,0,1,1. When the digital lock is open, the output of the circuit should be a logic 1 to turn on an LED. When the input serial sequence fails, the design should reset to the initial state. The LED should remain off until the lock is opened. This question is only asking for the FSM - do not design the sequential circuit. Do not worry about microcontroller settings for this question. For simplicity you may assume a Moore implementation. [10]

Moore Finite State Machine: This is the FSM for a 4-bit digital logic that turns on an LED (outputs HIGH) for the combination 1,0,1,1. When the sequence fails, the design returns to state 000 (starting state). I assumed that after the correct combination is entered, the next input automatically resets the sequence, returning to state 0, regardless of what it is.

FSM State Diagram:



State table:

Current State	Input	Next State	Output
000	0	000	0
000	1	001	0
001	0	010	0
001	1	000	0
010	0	000	0
010	1	011	0
011	0	000	0
011	1	100	0
100	X	000	1

The state diagram and table demonstrate the output is only changed to HIGH when the code is entered in in the correct order.