

Computing Systems Assignment 4 - Finite State Machines

Daniel Hannon (19484286)

February 2020



Figure 1: The Lock on this trolley is An Example of A Finite State Machine. Source: Twitter @Wagerlkunst

1 Introduction

Finite State Automata or Finite State Machines are one of the key components of a computer. In order for a computer to exist be it conceptual or physical, it must have some ability to deal with changes in inputs and alter outputs as such.

Within the contents of this report I hope to achieve the following:

1. Give a general overview and history of Finite State Machines(FSM)
2. Distinguish between the Mealy and Moore types of Finite State Machines.
3. Give a fully conceptualized example of a Finite State Machine.

2 What is an FSM?

2.1 Overview

A Finite State Machine (or FSM for short) is a tool which is often used to model the behaviour of a sequential system such as an Elevator. They consist of numerous states. The next state in which the machine will go to is determined by:

- The Inputs going into the machine.
- The Current state of the machine

2.2 History of the Finite State Machine

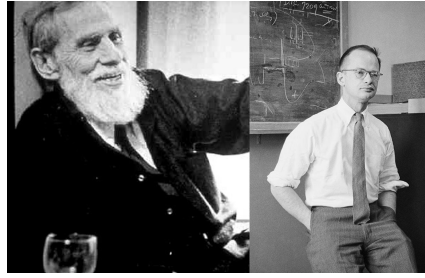


Figure 2: McCulloch (left) and Pitts (right)

Finite State Machines are not wholly restricted to computer science. In fact the groundwork for the theory of finite state automata was written by academics from numerous fields, including but not limited to: Biology, Psychology, Mathematics, Engineering, and of course Computer Science.

All of these academics were motivated by one common goal: to accurately model the Human Thought Process. One of the first defining papers for Finite State Machines as we know them today were set out in a 1943 paper titled "A Logical Calculus Immanent in Nervous Activity" which was written by two Neurophysiologists called Warren McCulloch and Walter Pitts. This paper made extremely large contributions to the study of Neural Networks, Automata, and Cybernetics. It was not until the 1950's when G.H. Mealey and E.F. Moore, in separate papers applied the work to far more powerful machines that it truly made an impact in the field. There are a few notable differences between what Mealy and Moore defined as a Finite State Machine, and we will explore these in Section 2 of this report.

2.3 What are Finite State Machines Used For?

Finite State Machines are used for many things in everyday life, everything from a trolley lock to an elevator is are Finite State Machines. some very good examples of Finite State Machines are as follows:

- Vending Machines - They have the number pad and cash inserted as an input and food as an output
- Automated Teller Machines - These work similarly to a vending machine but they allow you to perform menial/mundane tasks with your bank account such as, checking your balance, changing your pin, depositing money, and withdrawing money. they are often employed as a means of both accelerating banking operations and of reducing the required workforce to perform basic teller duties.
- Security Systems - such as fire alarms, burglar alarms, automatic locks, and Emergency Lighting systems.
- Assistive Technologies - Power assisted doors, chair lifts, and alarms.

2.4 Basic Example of a Finite State Machine

In order to explain a Finite State Machine we will return to Figure1 where there is an image depicting an out of commission shopping trolley. The lock on the shopping trolley is a great example of a very basic Finite State Machine. O1 refers to a state change while O2 refers to a looping action

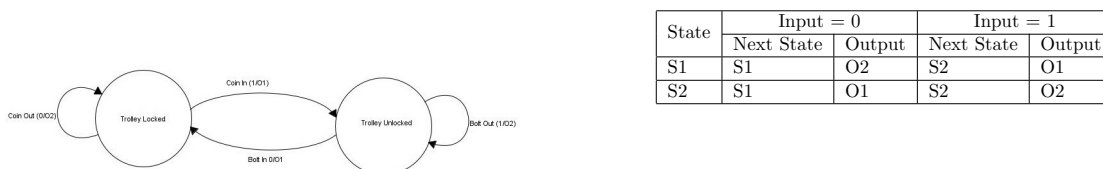


Figure 3: Trolley Finite State Machine and State Table

Based on the information above, hopefully I have made it clear that a Finite State Machine changes states depending on change in input.

3 Mealy versus Moore

As Mentioned before, there are two main types of Finite State Machine which we look at when it comes to computational theory. Mealy Machines, and Moore Machines.

3.1 Mealy Machine

3.1.1 History of a Mealy Machine

In 1955 George H. Mealy, published a paper titled "A Method for Synthesizing Sequential Circuits" in which he outlined the Mealy Machine as we know it today.

3.1.2 Overview of a Mealy Machine

A Mealy Machine is defined as a machine whose output values are determined by both its current state and its input values. The Machine described in Figure1 and Figure3 is of course a Mealy machine as the state of the trolley lock mechanism was entirely dependent on the inputs that were present as well as the current state of the trolley. Below is a diagram for a sequence detector for 01 or 10 With an accompanying table.

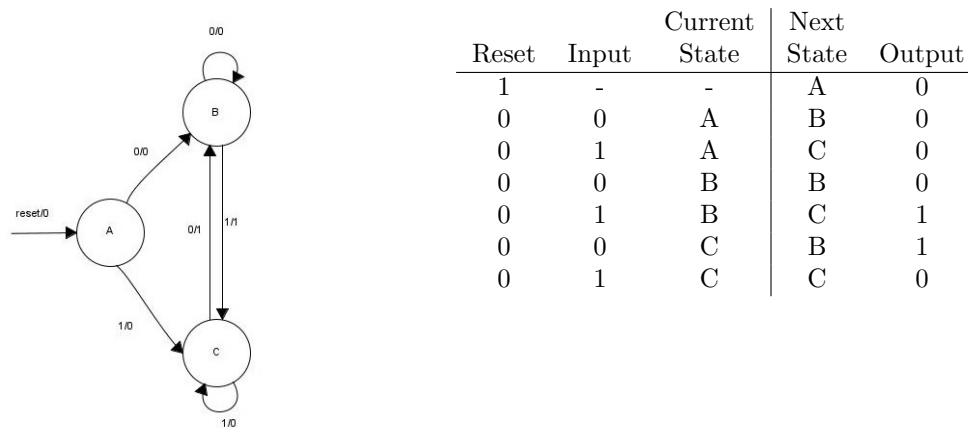


Figure 4: Mealy Machine Diagram and State Table

3.2 Moore Machine

3.2.1 History of the Moore Machine

In 1956, Edward F. Moore presented his take on the Finite State Machine in a paper titled "Gedanken-experiments on Sequential Machines"

3.2.2 Overview of the Moore Machine

Contrary to the Mealy Machine, a moore machines outputs are determined entirely by it's current state and I will provide a diagram of a sequence detector for 01 or 10 along with an accompanying table.

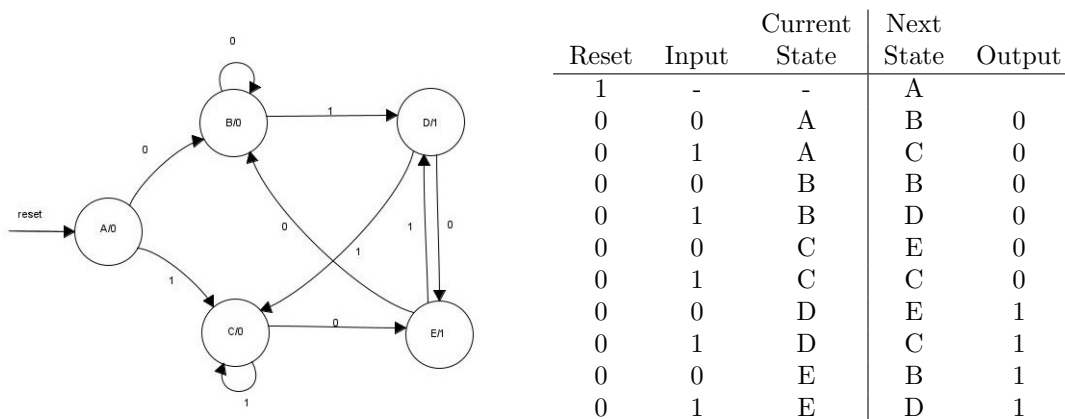


Figure 5: Moore Machine and State Table

4 Worked Example of a somewhat complex Finite State Machine

4.1 Overview of the Finite State Machine of choice

For my Finite State Machine I have selected a 4-digit combination lock. Due to the fact that the lock would in reality would have 100,000,000 combinations as the passcode mechanism would consist of 4 sets of mod 10 counters as one would be required to keep track of the position of the pin in the barrel and another would be required to keep track of the digit displayed to the end user.

A way to reduce it to 10,002 states would be by making the pin have a fixed location relative to the display. Due to the fact that 100,000,002 and 10,002 are both very difficult to neatly represent, I have completely Omitted the mechanism determining the position of the pin, reducing it to a single bit for each barrel.

Due to this omission it has become a significantly easier to represent, in fact it only requires 4 bits to determine state, three of which for current state, and a forth to determine whether or not the pin is in place or the bolt of the lock is open or not. Here is the now condensed state of the combination lock in table and graphic format

Current State	U	Next State	States represented in binary format
000	0	000	
000	1	001	Locked = 000
001	0	000	Row 1 = 001
001	1	010	Row 2 = 010
010	0	000	Row 3 = 011
010	1	011	Row 4 = 100
011	0	000	Unlocked = 101
011	1	100	
100	0	000	
100	1	101	
101	0	000	
101	1	101	

What "U" Represents

Locked = Button

Unlocked = Bolt

Rows 1-4 = if pin from mod 10 is in place.

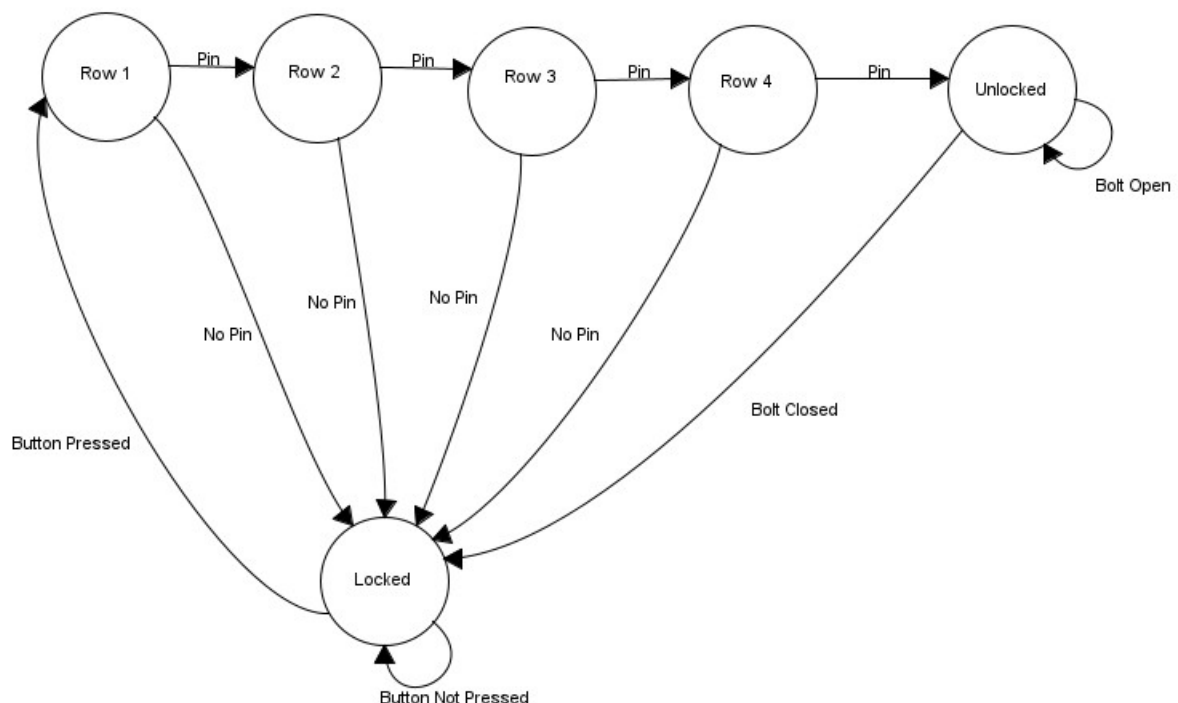


Figure 6: Simplified State Diagram for a Combination Lock.

4.2 Next state Logic of the FSM

Here are the karnaugh maps which I used to create the combinational logic featured at the very bottom. the groupings have been marked in red.

S2S1\S0U = N0					S2S1\S0U = N1				
	00	01	10	11		00	01	10	11
00	0	1	0	0	00	0	0	0	1
01	0	1	0	0	01	0	1	0	0
10	0	1	0	1	10	0	0	0	0
11	X	X	X	X	11	X	X	X	X

S2S1\S0U = N2				
	00	01	10	11
00	0	0	0	0
01	0	0	0	1
10	0	1	0	1
11	X	X	X	X

Figure 7: Karnaugh Maps

4.3 Combinational Logic of FSM

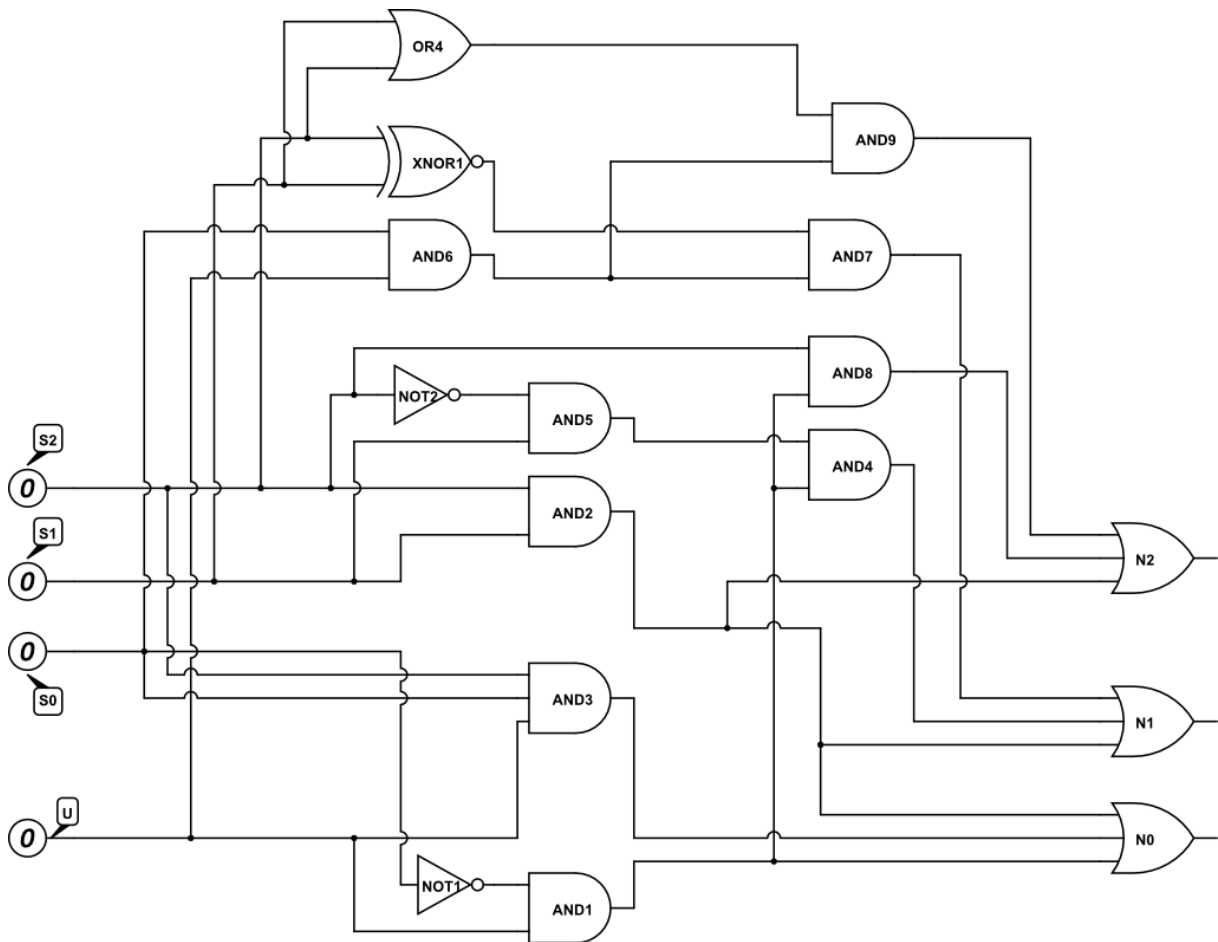


Figure 8: Combinational Logic of Combination Lock

5 References

FSM Design Tool
fizzim.com

History of Finite State Machines/Mealey Moore Machines

<https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/automata-theory/basics.html>

<https://www.geeksforgeeks.org/difference-between-mealy-machine-and-moore-machine/>

Examples of Mealy and Moore Machines

<http://www-inst.eecs.berkeley.edu/cs150/fa05/Lectures/07-SeqLogicIIIx2.pdf>

McCulloch and Pitts picture

<https://infonintelli.blogspot.com/2017/11/psychon-and-mcculloch-pitts-neurons.html>