Safe Password Controller

Cihan Yildirim
CSCI 210, Fall 2024
Email: cnyildiri@coastal.edu

Abstract

The Safe Password Controller is a digital logic design project that implements a Moore state machine to verify a password sequence. Using tools like truth tables, Karnaugh maps (K-maps), and state transition diagrams, the system was optimized for reliability and efficiency. The circuit successfully displayed password entry progress and lock status on a 7-segment display.

The Safe Password Controller features a keypad interface that allows the user to input a password sequence. The inputs from the keypad are processed by the password logic module, which implements the necessary combinational and sequential logic to validate the entered sequence. The system includes three 7-segment displays: the first display dynamically shows the user's current input, the second indicates the current state of the state machine controlling the safe, and the third provides a status indicator, displaying whether the safe is locked or unlocked.

I. Introduction

The Safe Password Controller is a digital logic design project aimed at developing a secure and efficient system for password validation. This device uses a keypad interface for input, where users can enter a predefined password sequence. The code is 231. The user inputs these numbers as binary number. 0 being 00, 1 being 01, 2 being 10, and 3 being 11. The system processes these inputs through a Moore state machine, designed to determine the correct sequence and manage state transitions. Based on the input sequence, the system can transition between "locked" and "unlocked" states, providing visual feedback on three integrated 7-segment displays.

The first 7-segment display dynamically reflects the current user input, while the second display shows the current state of the system's finite state machine. The third display provides a status indicator, explicitly showing whether the safe is locked or unlocked. This architecture ensures both clarity for the user and a reliable implementation of password verification.

To implement the Safe Password Controller, foundational digital logic techniques were employed, including the creation of truth tables, Karnaugh maps (K-maps), and minimal sum-of-products expressions. The system's Finite State Machine was modeled using a state transition diagram and realized with 2 D flip-flops. The decoder logic for the 7-segment displays was optimized using combinational logic derived from the system's requirements. Finally, after the safe system is unlocked, It stays unlocked until the user inputs a new number.

This document is organized as follows:

- **Background:** A review of fundamental digital logic concepts necessary for understanding the project, including truth tables, K-maps, and state machine design.
- **Design and Implementation:** A detailed walkthrough of the design process, including state tables, transition diagrams, K-map simplifications, and the construction of combinational and sequential logic circuits.
- **Results:** A validation of the system's functionality, demonstrating its ability to accurately process password sequences and provide real-time feedback via the 7-segment displays.
- **Conclusion:** A summary of the findings, evaluation of the system's performance, and suggestions for potential future improvements, such as scalability for longer passwords or enhanced user feedback mechanisms.

tikz

II. Background

Digital logic design forms the foundation of modern computing systems. It involves the systematic design and analysis of circuits that perform logical operations using binary inputs and outputs. This section provides an overview of essential concepts such as truth tables, Karnaugh maps (K-maps), and state machines, with generic examples to illustrate their application.

A. Truth Tables

Truth tables are a fundamental tool in digital logic. They are used to represent the relationship between inputs and outputs of a logic circuit. For instance, consider a simple two-input XOR gate:

A	**B**	**Output (A $\oplus$ B)**
0	0	0
0	1	1
1	0	1
1	1	0

TABLE I
Truth Table for XOR Gate

B. Karnaugh Maps (K-Maps) and Minimal Sum of Products

K-maps are a graphical method used to simplify boolean expressions by minimizing the number of terms in a sum-of-products (SOP) form. Consider the following truth table for a two-input function:

A	B	Output (F)
0	0	0
0	1	1
1	0	1
1	1	0

TABLE II
Truth Table for XOR Function ($F = A \oplus B$)

Using a K-map, the SOP expression for the function can be minimized. After grouping the 1s, the simplified expression becomes:

A\B	0	1
0	0	1
1	1	0

TABLE III
Karnaugh Map for XOR Function ($F = A \oplus B$)

C. *Deriving the Minimal Sum of Products (MSOP)*

From the K-map in Table III, the minterms where $F = 1$ are $A'B$ (row 0, column 1) and AB' (row 1, column 0). The minimized sum-of-products expression for F is:

$$F = A'B + AB' \tag{1}$$

D. *Moore State Machines*

A Moore state machine is a finite state machine where the outputs depend only on the current state, not the inputs. Consider the following example:

1) Example Moore State Machine: This example demonstrates a Moore state machine designed to produce an output based on the current state. The state diagram and the associated truth table are shown below.

2) State Transition Table: The machine has two states: - **a (S0)**: Initial state. - **b (S1)**: Second state.

The truth table for the Moore state machine is as follows:

Present State (PS)	Input (X)	Next State (NS)	Output (Z)
0 (a)	0	0 (a)	0
0 (a)	1	1 (b)	1
1 (b)	0	0 (a)	1
1 (b)	1	1 (b)	0

TABLE IV
Truth Table for the Moore State Machine

3) State Transition Diagram: The graphical representation of the state transitions is shown in Figure 1.

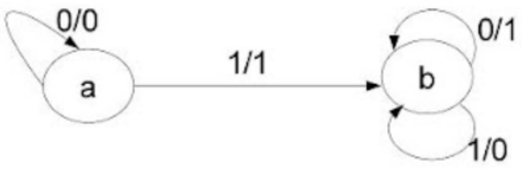

Fig. 1. State Transition Diagram for the Moore State Machine

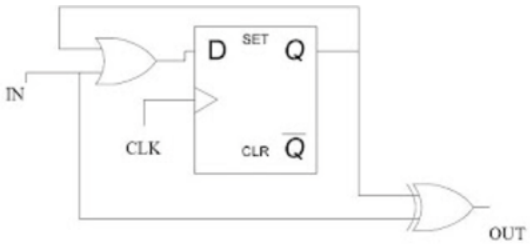

Fig. 2. Logic Diagram for the Moore State Machine

4) Logic Diagram: The Moore state machine is implemented using D flip-flops and combinational logic. The logic diagram for the circuit is shown in Figure 2.

III. Design and Implementation

The Safe Password Controller is designed to validate a password sequence and provide clear visual feedback through 7-segment displays. This section outlines the specific design processes, truth tables, Karnaugh maps (K-maps), state transition diagrams, equations, and the final circuit schematic.

A. *Design Overview*

The system consists of the following main components:

- A password validation logic implemented using state machines.
- 7-segment displays to show the input, current state, and lock status.
- A combinational and sequential logic circuit realized using D flip-flops, K-maps, and minimized equations.

B. *State Transition Diagram*

The password validation system is modeled as a Moore state machine, where the output depends only on the current state. The state transition diagram is shown in Figure 3.

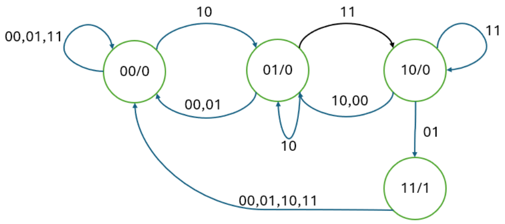

Fig. 3. State Transition Diagram for the Safe Password Controller

C. *Truth Tables and K-Maps*

The design is based on truth tables and K-maps for deriving minimized equations for state transitions and outputs.

1) Truth Table for State Transitions: The truth table is shown in Table V.

2) Karnaugh Maps (K-Maps): K-maps are used to minimize the next-state and output logic equations.

S1	S0	Q1	Q0	Q1Next	Q0Next	Unlock
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	1	1	1
0	1	1	1	0	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	1	1	0
1	1	1	0	1	0	0
1	1	1	1	0	0	0

TABLE V
TRUTH TABLE FOR THE SAFE PASSWORD CONTROLLER STATE TRANSITIONS

D. Karnaugh Maps for Next-State Logic

To derive the next-state logic equations, Karnaugh maps (K-maps) are used. The minimized equations for Q_1^{Next} and Q_0^{Next} are calculated based on the K-map groupings shown below.

1) Q_1^{Next}: The K-map for Q_1^{Next} is shown in Figure 4.

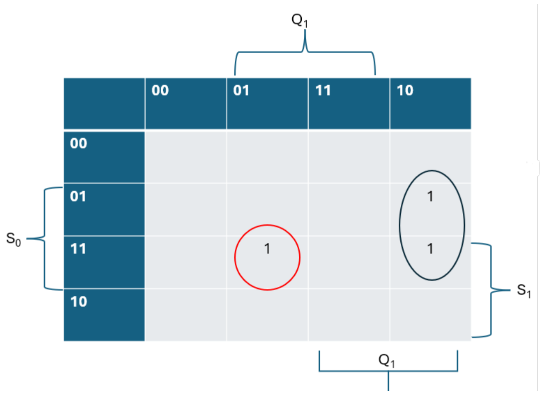

Fig. 4. K-map for Q_1^{Next} with groupings

From the K-map, the minimized Sum of Products (MSOP) equation for Q_1^{Next} is:

$$Q_1^{Next} = \ S_1 \cdot S_0\overline{Q_1} \cdot Q_0 + S_0 \cdot \ Q_1 \cdot \overline{Q_0}$$

2) Q_0^{Next}: The K-map for Q_0^{Next} is shown in Figure 5. From the K-map, the minimized Sum of Products (MSOP) equation for Q_0^{Next} is:

$$Q_0^{Next} = S_1\overline{S_0} \cdot \overline{Q_0} + S_1 \cdot \overline{S_0 Q_1} + \overline{S_1} \cdot Q_1\overline{Q_0}$$

E. 7-Segment Display Decoding

The outputs are displayed on three 7-segment displays:

- **Input Display (SSD0):** Shows the input digit.
- **Current State Display (SSD1):** Indicates the current state of the system.
- **Lock Display (SSD2):** Shows whether the safe is locked or unlocked.

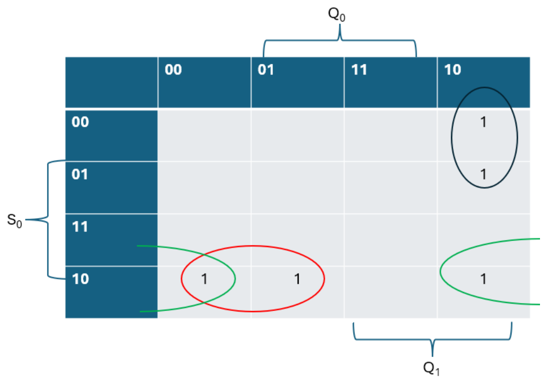

Fig. 5. K-map for Q_0^{Next} with groupings

F. SSD Decoder 0 - Input

The SSD Decoder 0 is designed to take the binary inputs S_1 and S_0 and generate control signals for the segments (a, b, c, d, e, f, g) of a 7-segment display. The decoder maps the binary state inputs to their respective decimal outputs.

1) Truth Table for SSD Decoder 0: The truth table for SSD Decoder 0 is shown in Table VI.

S_1	S_0	Decimal	a	b	c	d	e	f	g
0	0	0	1	1	1	1	1	1	0
0	1	1	0	1	1	0	0	0	0
1	0	2	1	1	0	1	1	0	1
1	1	3	1	1	1	1	0	0	1

TABLE VI
TRUTH TABLE FOR SSD DECODER 0

2) Karnaugh Maps for SSD Decoder 0: The K-maps for the control signals are shown below. The minimized equations derived from these maps are used to drive the segments of the 7-segment display.

The minimized equations for the segments are as follows:

- $a = S_1 + \overline{S_0}$
- $b = 1$
- $c = \overline{S_1} + S_0$
- $d = S_1 + \overline{S_0}$
- $e = \overline{S_0}$
- $f = \overline{S_1 S_0}$
- $g = S_1$

G. SSD Decoder 1 - State Display

SSD Decoder 1 is identical to SSD Decoder 0, except that its inputs are Q_1 and Q_0, which represent the current state. The decoder maps the binary state inputs to their corresponding decimal and segment outputs.

1) Truth Table for SSD Decoder 1: The truth table for SSD Decoder 1 is identical in structure but replaces S_1 and S_0 with Q_1 and Q_0, as shown in Table VII.

2) Karnaugh Maps for SSD Decoder 1: The K-maps for SSD Decoder 1 are identical to SSD Decoder 0 but use Q_1 and Q_0 instead of S_1 and S_0. The minimized equations for the segments of SSD Decoder 1 are as follows:

Q_1	Q_0	**Decimal**	a	b	c	d	e	f	g
0	0	0	1	1	1	1	1	1	0
0	1	1	0	1	1	0	0	0	0
1	0	2	1	1	0	1	1	0	1
1	1	3	1	1	1	1	0	0	1

TABLE VII
TRUTH TABLE FOR SSD DECODER 1

- $a = Q_1 + \overline{Q_0}$
- $b = 1$
- $c = \overline{Q_1} + Q_0$
- $d = Q_1 + \overline{Q_0}$
- $e = \overline{Q_0}$
- $f = \overline{Q_1 Q_0}$
- $g = Q_1$

IV. Full Circuit Design

The full circuit design for the Safe Password Controller is shown in Figure 6. This circuit integrates all the components, including state machines, SSD decoders, and combinational logic, to achieve the desired functionality.

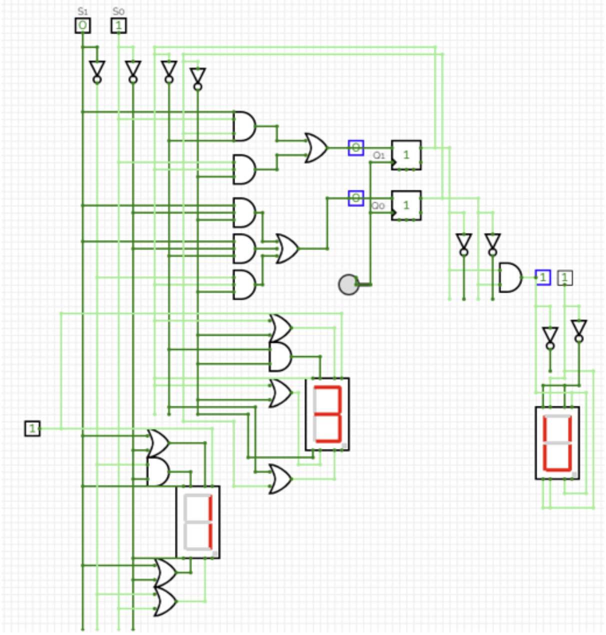

Fig. 6. Full Circuit Design for the Safe Password Controller

V. Results and Conclusions

The Safe Password Controller was thoroughly tested to ensure that it functions as intended. This section documents the test cases, provides examples of the password entry process, and demonstrates the progression of the system's state and outputs on the 7-segment displays.

A. Results

The circuit was designed to validate a predefined password sequence (e.g., `231`). Testing was performed by entering various sequences through the keypad and observing the behavior of the state machine and the output displays.

1) Password Entry Process: The following is an example of entering the correct password (`231`) into the system:

1) **Initial State:** The system starts in state $S0$ with all displays showing their default values:
   - **Input Display:** ‘0‘ (no input yet)
   - **State Display:** ‘S0‘
   - **Lock Display:** ‘Locked‘
2) **First Input (`2`):** The user presses `2`, transitioning the system to $S1$. Displays update as follows:
   - **Input Display:** ‘2‘
   - **State Display:** ‘S1‘
   - **Lock Display:** ‘Locked‘
3) **Second Input (`3`):** The user presses `3`, transitioning the system to $S2$. Displays update as follows:
   - **Input Display:** ‘3‘
   - **State Display:** ‘S2‘
   - **Lock Display:** ‘Locked‘
4) **Third Input (`1`):** The user presses `1`, transitioning the system to $S3$, the unlocked state. Displays update as follows:
   - **Input Display:** ‘1‘
   - **State Display:** ‘S3‘
   - **Lock Display:** ‘Unlocked‘

2) Test Case Observations: Table VIII summarizes the test cases and their outcomes.

Input Sequence	Final State	Output (Lock Status)	Result
231	S3	Unlocked	Pass
230	S0	Locked	Pass
233	S2	Locked	Pass
213	S1	Locked	Pass

TABLE VIII
SUMMARY OF TEST CASES FOR THE SAFE PASSWORD CONTROLLER

B. Conclusions

The Safe Password Controller successfully met its design specifications, validating password sequences and providing accurate visual feedback via the 7-segment displays. Key observations include:

- The state machine transitioned between states as expected based on the input sequence.
- The 7-segment displays updated dynamically to reflect the current input, state, and lock status.
- The circuit reliably unlocked the safe upon entry of the correct password and maintained a locked state for incorrect sequences.

1) Future Improvements: While the system performed as intended, potential enhancements could include:

- Expanding the safe system to handle longer passwords.
- Adding a reset button to restart the process at any point.
- Being able to input 0 - 9.

This project demonstrated the practical application of digital logic design principles and provided a robust, user-friendly solution for password validation.

REFERENCES

[1] M. M. Mano and M. D. Ciletti, *Digital Design*, 6th ed., Pearson, 2017.

[2] C. H. Roth and L. L. Kinney, *Fundamentals of Logic Design*, 8th ed., Cengage Learning, 2020.

[3] J. F. Wakerly, *Digital Design: Principles and Practices*, 5th ed., Pearson, 2018.

[4] E. F. Moore, ”Gedanken-Experiments on Sequential Machines,” in *Automata Studies*, vol. 34, pp. 129–153, 1956.

[5] ”Karnaugh Map Tutorial,” *Karnaugh Map*, [Online]. Available: https://www.karnaughmap.com. [Accessed: Nov. 28, 2024].

[6] ”CircuitVerse: Digital Logic Circuit Simulator,” [Online]. Available: https://circuitverse.org. [Accessed: Nov. 28, 2024].

[7] T. Tantau, ”TikZ and PGF Manual,” [Online]. Available: https://texample.net/tikz/. [Accessed: Nov. 28, 2024].

[8] ”7-Segment Decoder Circuit,” *Electronics Tutorials*, [Online]. Available: https://www.electronics-tutorials.ws. [Accessed: Nov. 28, 2024].

[9] ”Password Validation with Digital Logic Circuits,” *ResearchGate*, [Online]. Available: https://www.researchgate.net. [Accessed: Nov. 28, 2024].

[10] ”FSM Questions and Answers,” *VLSI Universe*, [Online]. Available: https://www.vlsiuniverse.com/fsm-finite-state-machine-questions/. [Accessed: Nov. 28, 2024].

[11] ”FSM Questions and Answers,” VLSI Universe. [Online]. Available: https://www.vlsiuniverse.com/fsm-finite-state-machine-qustions/. Accessed: Nov. 28, 2024.