Pre-Lab 11

Carefully read the entirety of Lab 11, then answer the following questions. Attach a separate sheet of paper, if necessary, to show all work and calculations.

1. Carefully read Circuit 1.

   (a) Define all of the necessary states (**and their corresponding outputs**) required to create the specified Moore machine.

   (b) Draw a state diagram.

   (c) Fill out the following state table.

Current State	**Next State**		**Output**
	X=0	**X=1**	

2. Carefully read Circuit 2 in the lab.

   (a) Define all of the necessary states required to create the specified Mealy machine.

   (b) Draw a state diagram.

   (c) Fill out the following state table.

Current State	**Next State**		**Output**	
	X=0	**X=1**	**X=0**	**X=1**

3. Carefully read Circuit 3 in the lab.

(a) Fill out the following state table using the state diagram given in the lab. (Note the use of full Gray code in the state table!)

CS	**Next State**								**Output**							
	XYZ=								**XYZ=**							
	000	001	011	010	110	111	101	100	000	001	011	010	110	111	101	100
S0																
S1																
S2																

(b) Decide what type of flip-flop (D or JK) you want to use to implement this finite state machine.

(c) Create state assignments for each state. State S_0 should be assigned as 00.

(d) Derive an equation (or equations, if using a JK flip-flop) for the next state A^+ flip-flop.

(e) Derive an equation (or equations, if using a JK flip-flop) for the next state B^+ flip-flop.

(f) Derive an equation for the output, T.

(g) Use equation 11.1 to calculate the resistor value required to create a pulse of approximately 4 s duration. Refer to the inventory of parts to select one resistor and capacitor to achieve this within 10% of the desired pulse time.

Lab 11: Sequence Detectors

This lab will focus on sequence detecting finite state machines. Both synchronous (Moore) and asynchronous (Mealy) sequence detectors will be explored. Finally, a more complicated password unlock circuit will be built using a keypad as an input device.

For lab resources and information, go to the following URL or scan the QR code. doctor-pasquale.com/digital-systems-lab-11

11.1 Moore Machines

Moore machines are finite state machines in which the output(s) only depends on the current state of the device, and not on any of the input value(s). The outputs of a Moore machine are therefore said to be synchronous. A Moore machine will have the same output (whose value is defined by the state table) throughout the duration of a state and will not change until the clock ticks and changes the state of the finite state machine.

Given the state diagram shown in figure 11.1, the timing diagram in figure 11.2 will result, given rising-edge triggered flip-flops. Note that the output is HIGH for the entire duration that the finite state machine is in state S2.

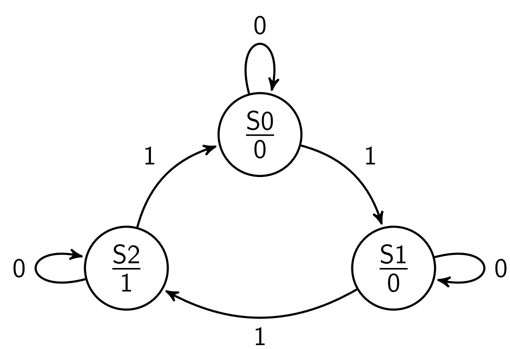


Figure 11.1: Example Moore machine state diagram.

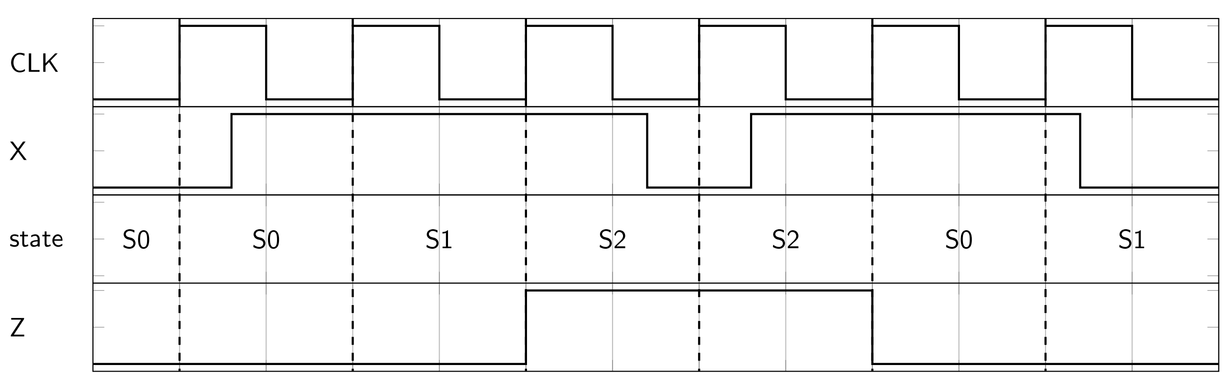


Figure 11.2: Timing diagram corresponding to the example state diagram shown in figure 11.1.

A Moore machine is therefore slightly easier to troubleshoot than a Mealy machine. Because the outputs are synchronous, the timing of the output signal is steadier and simpler to confirm using the troubleshooting methods outlined below. However, because Moore machines have outputs defined by their states, a Moore machine will have more states than an otherwise equivalent Mealy machine. This can lead to more difficulty in the initial design and setup of the circuit, and can lead to more complicated expressions for each flip-flop input.

11.2 Mealy Machines

Mealy machines are finite state machines in which the output(s) depends on **both** the current state of the device and on the input value(s). The outputs of a Mealy machine are therefore said to be asynchronous. A Mealy machine output will change as soon as the input changes to a value that would lead to a changing output based on the state table.

Given the state diagram shown in figure 11.3, the timing diagram in figure 11.4 will result, given rising-edge triggered flip-flops. This finite state machine has a very similar design to the Moore machine shown in figure 11.1, and the timing diagram has the same clock and input values as the Moore machine shown in figure 11.2. Note that the output is HIGH only when the input changes asynchronously to a HIGH value in state S2, and becomes LOW again immediately upon the next clock tick.

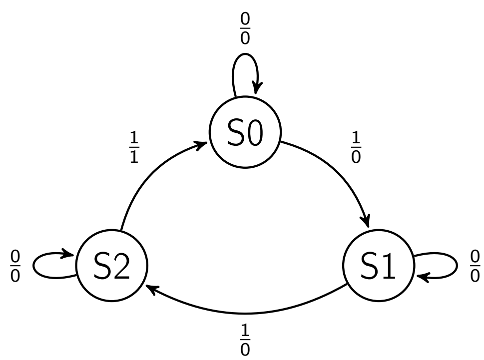


Figure 11.3: Example Mealy machine state diagram.

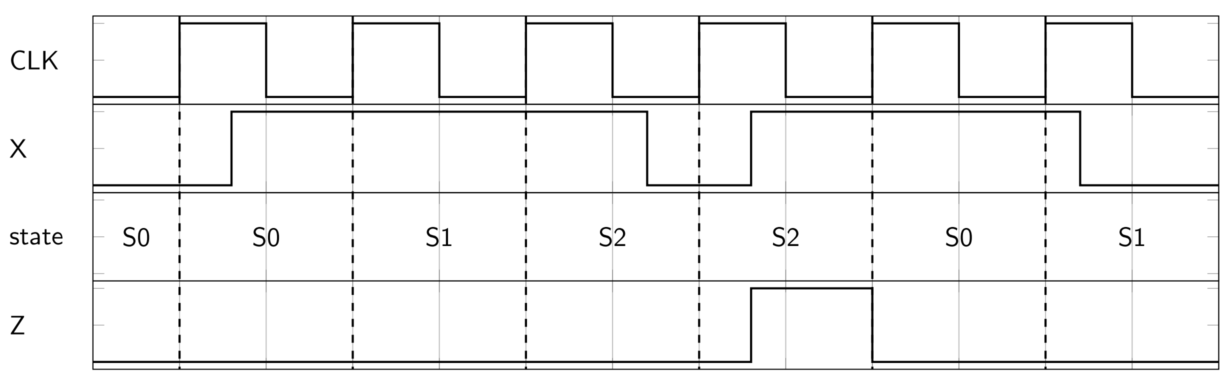


Figure 11.4: Timing diagram corresponding to the example state diagram shown in figure 11.3.

Mealy machines are therefore slightly more difficult to troubleshoot than Moore machines. The timing of the input is important to consider when building and debugging a Mealy machine. Because the output will change asynchronously with the input, it is important to note that the output may change only for a very short period of time, especially if the clock is about to tick and cause the state to change and cause the output to change back to its initial value. Output values can appear as a glitch in these cases. Having a solid understanding of the timing details is important.

11.3 Troubleshooting a Finite State Machine

Troubleshooting a synchronous circuit is generally more complicated than troubleshooting a combinational circuit. Because the circuit relies on feedback, and states are constantly changing, it can be difficult to obtain a steady state of input and output signals to verify using a logic probe. Following are a few troubleshooting tips that you can use with synchronous circuits.

- Use a slow clock signal to observe flip-flop and output values. (In this lab, a clock frequency of 1 Hz will be used specifically because it is slow enough for humans to meaningfully interact with and view the inputs and outputs.)
- Place debug LEDs on the output of each flip-flop. They should cycle through values based on the state table and state assignments every time the clock ticks.

Circuit 1: Using a clock set to a frequency of approximately 1 Hz, and using D or JK flip-flops, create a Moore machine to detect the binary sequence **001**. This should be a sliding window detector type circuit. Display the output on an LED. Include a definition of all states and their binary state assignments, a transition table, and a circuit diagram. Finally, wire up the circuit and verify its functionality. Then, demonstrate it to your instructor to receive a stamp.

State Assignments:

Transition Table:

Current State	Next State		Flip-Flop		Output
	X=0	X=1	X=0	X=1	

Expressions:

Flip-Flop A equation(s): ____________________

Flip-Flop B equation(s): ____________________

Z = ____________________

Instructor Stamp: _____

Circuit 2: Using a clock set to a frequency of approximately 1 Hz, and using D or JK flip-flops, create a Mealy machine to detect the binary sequence **001**. This should be a sliding window detector type circuit. Display the output on an LED. Include a definition of all states and their binary values, a state table, a state diagram, and a circuit diagram. Finally, wire up the circuit and demonstrate its functionality to your instructor to receive a stamp.

State Assignments:

Transition Table:

Current State	Next State		Flip-Flop		Output	
	X=0	X=1	X=0	X=1	X=0	X=1

Expressions:

Flip-Flop A equation(s): ______________________

Flip-Flop B equation(s): ______________________

Z = ______________________

Instructor Stamp: ____

11.4 Keypad

A keypad is a device with several pushbuttons (usually 12 or 16 of them) that correspond to different values. Used with an encoder, the output of a keypad circuit will correspond to the binary value of the button that was pressed. A side-by-side comparison of a keypad (used with an encoder) and a DIP switch is presented in table 11.1.

Keypad with Encoder	**DIP Switch**
No knowledge of binary is needed	Requires knowledge of binary
The output values all change at once	The output value only changes one bit at a time as the user toggles each switch
A keypad requires an encoder to scan each row and column for a button press	Does not require an encoder because each switch contains connections to Vcc and GND
The output value persists until the next button press	The output value persists until a switch is toggled
The DATA pin on the encoder creates an asynchronous signal to alert that an input has been selected	External circuitry or a pushbutton would be required to alert that an input has been selected

Table 11.1: Side-by-side comparison of a keypad with encoder and a DIP switch.

The keypad used in this lab has 12 buttons corresponding to values 0–9, as well as symbols that output binary values corresponding to decimal values of 10 and 11. Each keypad has been wired up on a PCB with the 74922 keypad encoder used in asynchronous data entry mode. Schematics of the 74922 encoder, the 12-character keypad, and the keypad PCB are all provided in Appendix A.

The keypad encoder has an output pin (DATA) that causes a temporary HIGH signal any time a keypad button is pressed. This pin is the perfect candidate to act as a clock source for a finite state machine requiring an input value from a keypad. Rather than synching user input to a clock signal (as was done in the previous two circuits), the state can change asynchronously based on the user button press. This creates a much more user-friendly interface, and does not require any extra circuitry that would cause the finite state machine to go into a WAIT or IDLE mode while waiting for the user to input a new value.

11.5 Monostable Timer

Previously, a 555 timer had been used in astable mode to create a signal that continuously oscillates between LOW and HIGH values. This was used as a clock signal for flip-flops.

In this lab, a 555 timer will be used in **monostable** mode to create a single HIGH pulse (sometimes called a one-shot) upon a triggering event. When the trigger pin on the timer (in monostable mode) is set to logic LOW, the timer will generate a HIGH output signal for the time period described by equation 11.1, otherwise the signal output will be LOW.

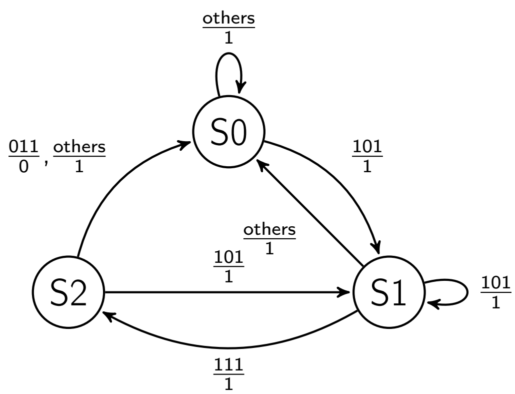
$$t \approx 1.1RC \qquad (11.1)$$

A monostable timer is the ideal solution for a circuit that requires a finite interval of time to elapse between events. Rather than relying on a clock signal to synchronize the timing of states for that duration, a one-shot pulse can be used instead. Because the monostable timer requires a LOW signal to generate a pulse, that active-LOW property will need to be taken into account in the design of a circuit.

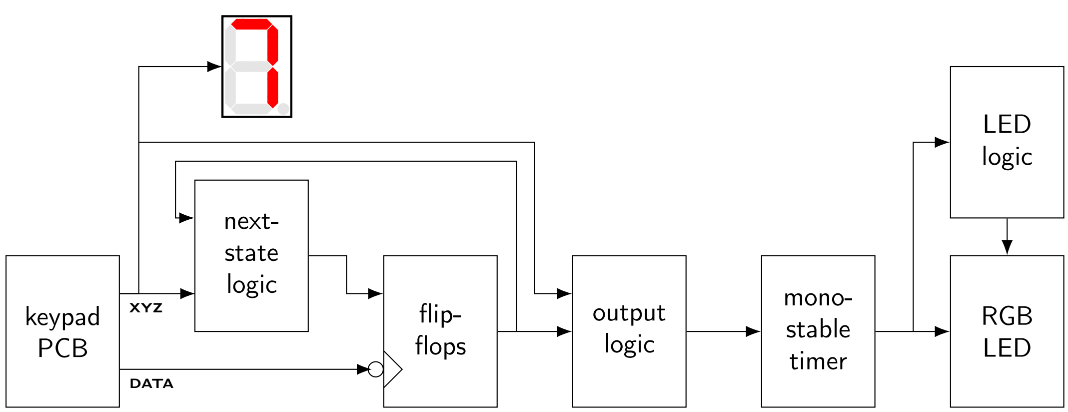
Circuit 3: Design a three digit Mealy Machine sequence detector with input coming from a keypad wired up to a keypad encoder chip. Use either D or JK flip-flops. A an RGB status LED will be red when the system is locked and green when it is unlocked. The system will unlock for 4 seconds when the sequence **5–7–3** is entered into the keypad. The keypad numeral will also be displayed simultaneously on a 7-segment display. The clock signal will come from pin 12 of the keypad encoder (labeled `DATA`). (Note that, to avoid setup issues, the flip-flops will require a falling-edge trigger.) Wire up the circuit and verify its functionality. Demonstrate it to your instructor to receive a stamp. A high-level schematic of this circuit is provided below.

State Definitions and Diagram:

- S_0 – Reset state (no numbers toward the sequence have been detected, or, the sequence has been successfully completed)
- S_1 – 5 (the numeral 5 has been detected toward completing the sequence)
- S_2 – 5–7 (the numeral 7 has been detected after the numeral 5)



High-Level Schematic:



Expressions:

Flip-Flop A equation(s): ____________________

Flip-Flop B equation(s): ____________________

T = ____________________

Instructor Stamp: ______

Lab 11 Homework

Carefully read each question before answering. Show all work or justify your answers to receive credit. Attach a separate sheet of paper, if necessary, to show all work and calculations.

1. Design a **non-overlapping sliding window** Moore machine detector that has an output $Z = 1$ if the sequence **1101** is detected, and otherwise the output $Z = 0$.

   (a) Define all of the states required to implement this.

   (b) Draw a state diagram

   (c) Fill out the following state table

Current State	**Next State**		**Output**
	X=0	**X=1**	

(d) Define each of the state assignments. Ensure that the reset state is properly assigned.

(e) Fill out the following transition table

Current State	**Next State**		**Output**
	X=0	**X=1**	
000			
001			
011			
010			
100			
101			
111			
110			

(f) Derive an equation for A^+ using a D flip-flop.

(g) Derive an equation for B^+ using a D flip-flop.

(h) Derive an equation for C^+ using a D flip-flop.

(i) Derive an equation for Z.

2. Design an **overlapping sliding window** Mealy machine detector that has an output $Z = 1$ if either the sequence **001** or **1100** is detected. Otherwise the output $Z = 0$.

   (a) Define all of the states required to implement this.

   (b) Draw a state diagram

   (c) Fill out the following state table

Current State	Next State		Output	
	X=0	X=1	X=0	X=1

(d) Define each of the state assignments. Ensure that the reset state is properly assigned.

(e) Fill out the following transition table

Current State	**Next State**		**Output**	
	X=0	**X=1**	**X=0**	**X=1**
000				
001				
011				
010				
100				
101				
111				
110				

(f) Derive an equation for A^+ using a D flip-flop.

(g) Derive an equation for B^+ using a D flip-flop.

(h) Derive an equation for C^+ using a D flip-flop.

(i) Derive an equation for Z.