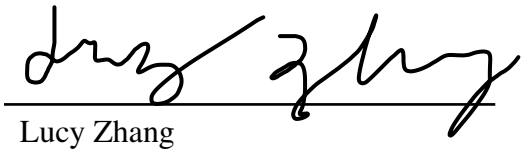
CMPE-160 Digital System Design 1

Laboratory Exercise 9

Design and Simulation of a Moore State Machine

By submitting this report, I attest that its contents are wholly my individual writing about this exercise and that they reflect the submitted code. I further acknowledge that permitted collaboration for this exercise consists only of discussions of concepts with course staff and fellow students. Other than code provided by the instructor for this exercise, all code was developed by me.



Lucy Zhang
Performed 25 October 2022
Submitted 1 November 2022

Lab Section:	1
Instructor:	Purab Sutradhar
TA:	Morgan Kreifels
	Karisha Pradham
	Colin Vo
Lecture Section:	1
Lecture Instructor:	Richard Cliver

Abstract

Given a sequence to detect, a state diagram and state table were created. Based on those, equations were derived and a circuit schematic was made. The circuit was verified through testing in simulation. The results of the simulations were as expected, so the exercise was successful.

Design Methodology

A state machine can be used to detect a sequence. The state machine in this exercise used two data inputs (A and B), a rising-edge triggered clock (CLK), an asynchronous active low reset signal (RST), and one data output (Z). The state machine was detecting the following sequence in the following order: (A,B) = (1,1), (1,0), (0,0). This means that the output Z would output logic 0 until the sequence was detected in which case it would output logic 1. Z is described as staying at logic 1 until the input is changed.

Using the knowledge of the functionality of the state machine, Figure 1 was created.

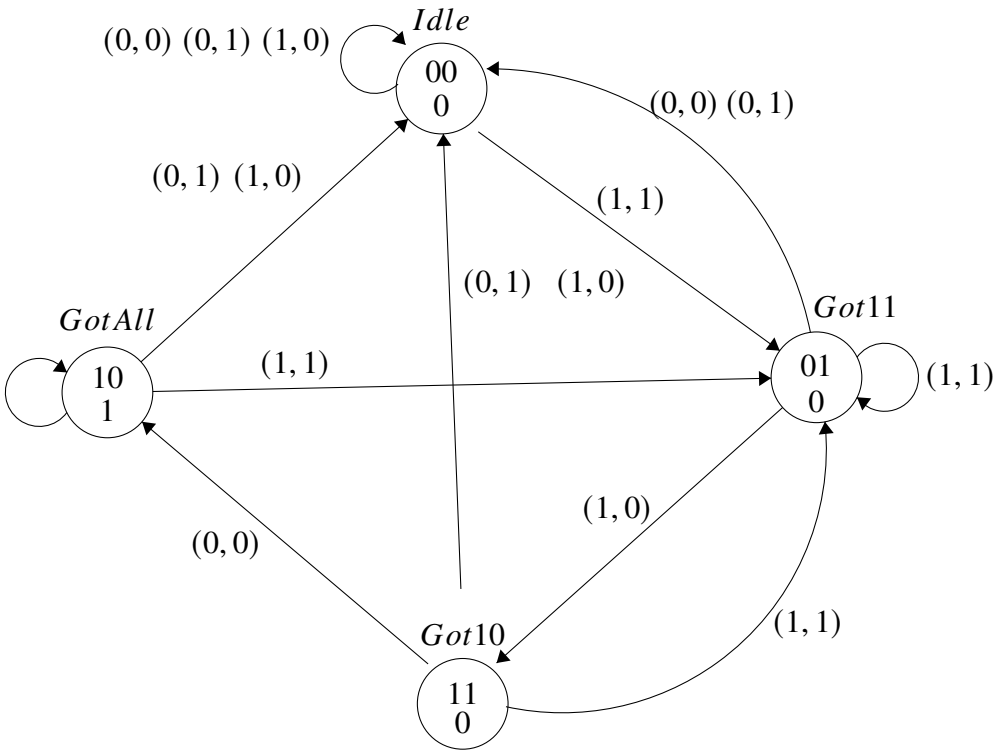


Figure 1: *Moore State Diagram*

Figure 1 shows the state diagram for a Moore state machine described. The state names are Idle, Got11, Got10, and GotAll. Each state corresponds to a state. Inside the circles of Figure 1, the two digits on the top are the state and the bottom digit is the output (Z) value. The data inputs are represented next to each arrow. The arrows point to the next state that each input pair results in.

In addition to the state diagram, Table 1 was created.

Table 1: *Moore State Table*

State Name	State (Q_1, Q_0)	Input (A,B)				Output
		(0,0)	(0,1)	(1,1)	(1,0)	Z
Idle	0 0	0 0	0 0	0 1	0 0	0
Got11	0 1	0 0	0 0	0 1	1 1	0
Got10	1 1	1 0	0 0	0 1	0 0	0
GotAll	1 0	1 0	0 0	0 1	0 0	1

Table 1 shows the state table of the Moore state machine being created. The first two columns show the state names and the encodings for the Moore state machine respectively. The next column shows the four possible input combinations and the resulting state given for each previous state. The last column shows the value of the output Z.

Using Table 1, Karnaugh maps were made to describe Q_1^*, Q_0^*, and Z. Q_1^* and Q_0^* represent the resulting state and Q_1 and Q_0 represent the previous state. A and B are the two data inputs.

Table 2 was created to describe Q_1^*.

Table 2: *K-map for* Q_1^*

Q_1Q_0 \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	0	0	0
10	1	0	0	0

Table 2 is a K-map that shows the value of Q_1^* for the state (Q_1, Q_0) and the input (A, B).

Based off the K-map, equation (1) was derived for Q_1^*.

$$Q_1^* = \bar{A}\bar{B}Q_1 + A\bar{B}\bar{Q}_1Q_0 \tag{1}$$

Equation (1) describes the function of Q_1^*. Equation (1) shows how to find Q_1^* given the input and the previous state.

Table 3 was created to describe Q_0^*.

Table 3: *K-map for* Q_0^*

Q_1Q_0 \ AB	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	0	0	1	0
10	0	0	1	0

Table 3 is a K-map that shows the value of Q_0^* for the state (Q_1, Q_0) and the input (A, B).

Based off the K-map, equation (2) was derived for Q_0^*.

$$Q_0^* = A\bar{Q}_1Q_0 + AB \tag{2}$$

Equation (2) shows how to find Q_0^* given the input and the previous state.

Table 4 was created to describe Z.

Table 4: *K-map for* Z

Q_1 \ Q_0	0	1
0	0	0
1	1	0

Table 4 is a K-map that shows the value of Z based on Q_1 and Q_0.

Based off the K-map, equation (2) was derived for Z.

$$Z = Q_1\bar{Q}_0 \tag{3}$$

Equation (3) describes the output Z.

Based on the equations (1), (2), and (3), Figure 2 was created.

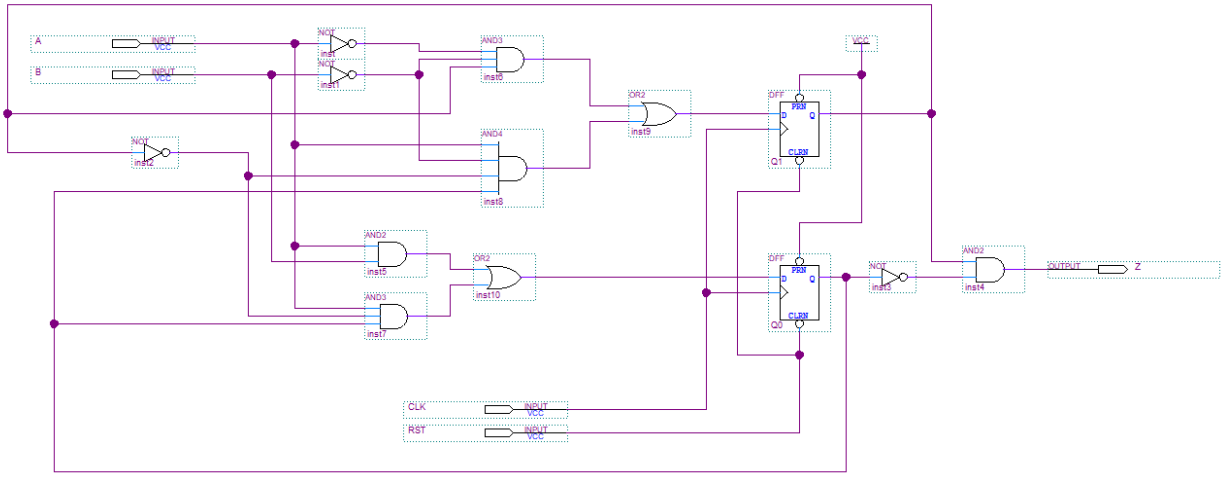


Figure 2: *Circuit Diagram of the Moore State Machine*

Figure 2 shows the circuit diagram of the Moore state machine. It can be seen that the next state is dependent on the previous state. The Q output of each D flip-flop becomes the input of the next state (Q^*). The D flip-flop at the top represents the most significant bit. The D flip-flips are rising edge triggered by the CLK input. The RST sets Q_1^* and Q_0^* to logic 0 asynchronously, meaning it's not dependent on the clock. The other inputs and outputs correspond respectively to the input and outputs described in Table 1.

Results and Analysis

The circuit was tested and verified through simulation. The following sequence was used to test the state machine: (A,B) = (1,1), (1,0), (0,0), (1,0), (0,0), (1,1), (1,0), (0,0), (0,0). Whenever the circuit encounters the sequence (A,B) = (1,1), (1,0), (0,0), Z should output logic 1.

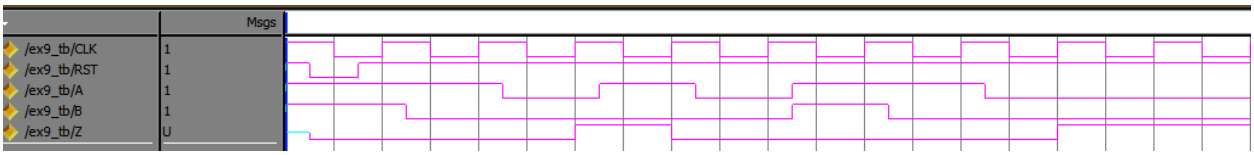


Figure 3: *Waveform of the First Sequence Test*

Figure 3 shows the waveforms from the simulation. The sequence used to test the circuit has two times where the desired sequence is detected, so the Z should output high twice. As seen in Figure 3, the Z is high twice. The waveform of Z is blue at the beginning because before the first rising edge of the clock or a reset, the value of Z is undefined. The waveform results are as expected. The waveforms match the expected results in Table 1. This verifies the circuit is working accurately.

The circuit is tested again using a different input sequence. The following sequence was used for the second test: (A,B) = (1,1), (1,0), (0,0), (0,0), (1,1), (1,0), (1,1), (0,1), (1,1). Whenever the circuit encounters the sequence (A,B) = (1,1), (1,0), (0,0), Z should output logic 1.

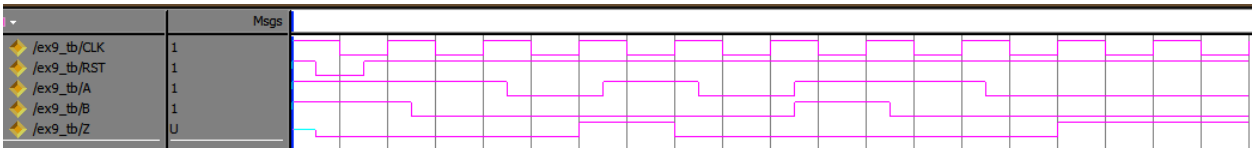


Figure 4: *Waveform of the Second Sequence Test Expressions*

Figure 4 shows the waveforms from the second simulation test. The sequence used to test the circuit has one time where the desired sequence is detected, so the Z should output high once. Immediately after the sequence occurs, the input does not change, so Z is still high. The waveform of Z is blue at the beginning because before the first rising edge of the clock or a reset, the value of Z is undefined. The waveform results are as expected. The waveforms match the expected results in Table 1. This also verifies the circuit is working accurately.

The timing of a state machine is important for their operation. T_{min} is the minimum clock period that the circuit can operate correctly at. The maximum operating speed of a circuit can then be determined from the T_{min} of a circuit.

The T_{min} of a circuit can be determined with Equation (4).

$$T_{min} = t_{co} + t_{pd} + t_s \tag{4}$$

Equation (4) describes how the minimum clock period that the circuit could operate at correctly. t_{co} is the clock to output time, t_s is the setup time, and t_{pd} is the propagation delay through the logic. For the circuit in Figure 2, the t_{co} was 6.114 ns, the t_s was 2.239 ns, and the t_{pd} was 0 ns. The T_{min} was calculated using (4) and was 8.353 ns. T_{min} was converted from seconds to 119.717 MHz. This means the maximum frequency the circuit can operate at is about 120 MHz.

Conclusion

A state diagram and state table were created of a Moore state machine. Equations were derived from the state table and used to create a circuit. The circuit was tested and verified through simulation. The results were as expected and the exercise was a success. Understanding how one state machine works can help engineers understand how to implement them in a more complex circuit or in a larger scale. A state machine can be used to detect sequences. The timing of a state machine is important to consider to prevent hazards and accurate data flow.

Questions

1. Repeat the synthesis process for the given state encoding style:
   Table 5 was created based off the given encoding style.

Table 5: *State Table*

State Name	State	Input (A,B)				Output
	(Q_1, Q_2)	(0,0)	(0,1)	(1,1)	(1,0)	Z
Idle	0 0	0 0	0 0	0 0	0 1	0
Got11	0 1	0 0	0 0	0 1	0 1	0
Got10	1 0	1 1	0 0	0 0	0 1	0
GotAll	1 1	1 1	0 0	0 0	0 1	1

Table 5 describes the functionality of the Moore state machine.
Using Table 5, Equations (5), (6), and (7) were created.

$$Q_1^* = \bar{A}\bar{B}Q_1 + A\bar{B}\bar{Q}_1Q_0 \tag{5}$$

$$Q_0^* = \bar{A}\bar{B}Q_1 + AB \tag{6}$$

$$Z = Q_1Q_0 \tag{7}$$

Equations (5) and (6) describe the output and and (7) describes the next state.
Figure 5 was created based from the equations.

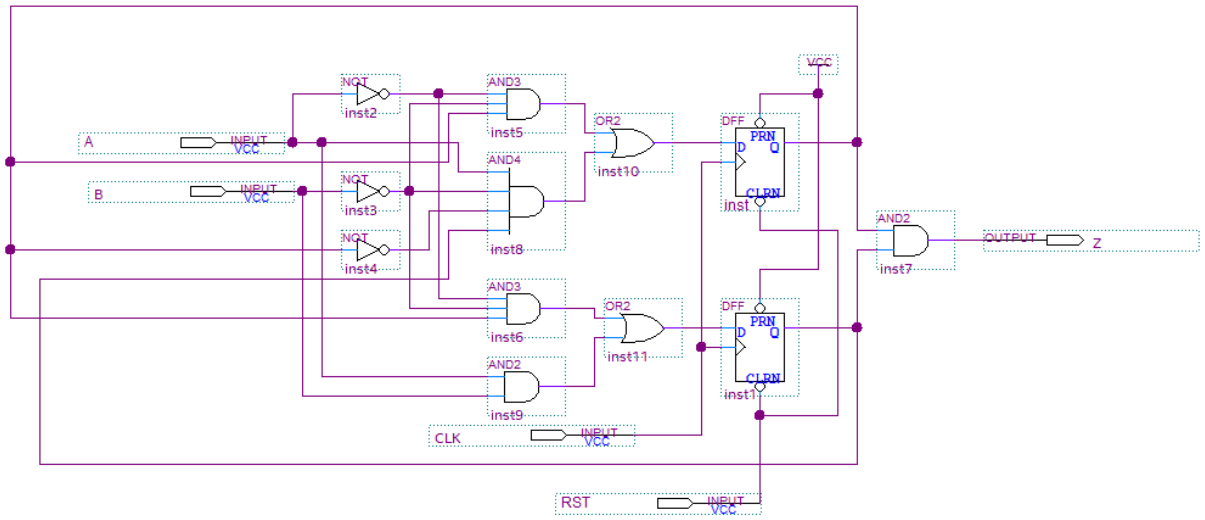


Figure 5: *Circuit for State Machine*

Figure 5 shows the resulting circuit made from the alternative states and encodings for the Moore state machine.

2. How many different ways are there to encode a state machine with four states that is implemented using two D flip-flops?
   There are four different ways to encode a state machine with four states that is implemented using two D flip-flops.
3. What is ”one-hot” encoding style?
   Only one bit is changed in a ”one-hot” encoding style.

Exercise 9: Design and Simulation of a Moore State Machine

Student's Name:_____ Section:_____

Prelab		Point Value	Points Earned	Comments
Prelab	State diagram	5	5	CV 10/27
	State table	5	5	
	Equations	5	5	
	Circuit diagram	5	5	

Demo		Point Value	Points Earned	Date
Demo	Part 1 simulation	15	15	CV 10/27
	Part 2 simulation	15	15	CV 10/27
	Timing analysis	10	10	CV 10/27

To receive any grading credit students must earn points for both the demonstration and the report.

Exercise 9: Design and Simulation of a Moore State Machine

Report		Point Value	Points Earned	Comments
Abstract		4		
Design Methodology	State diagram	3		
	State transition table	3		
	Equations	3		
	Circuit diagram	3		
Results and Analysis	Test sequence	3		
	Simulation results	3		
	Timing results	2		
Questions	Q1	4		
	Q2	2		
	Q3	2		
Conclusion		4		
Writing Composition		4		
Total for prelab, demo, and report		100		