CMPE-160 DSD1

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.

Oliver Vinneras
23 March 2022

Lab Section:	3
Instructor:	Mr. Lange
TA:	Henry Bang Harrison Barnes Ian Chasse Shubhang Mehrotra
Lecture Section:	1
Lecture Instructor:	Mr. Cliver

Abstract

The purpose of this exercise was to observe, design, and simulate a Moore state machine based on the functional description of a sequence detector. This functional description detailed how the two inputs, A and B, reacted to a reset, what the required sequence of inputs was to produce an output of 1, and what the state names were for each input.

Design Methodology

A Moore state machine is a finite state machine with outputs that only depend on the current state. This was implemented by creating a sequence detector. With a given clock and reset, a sequence detector returns true if the specific sequence of states matches the sequence that the model was based off of.

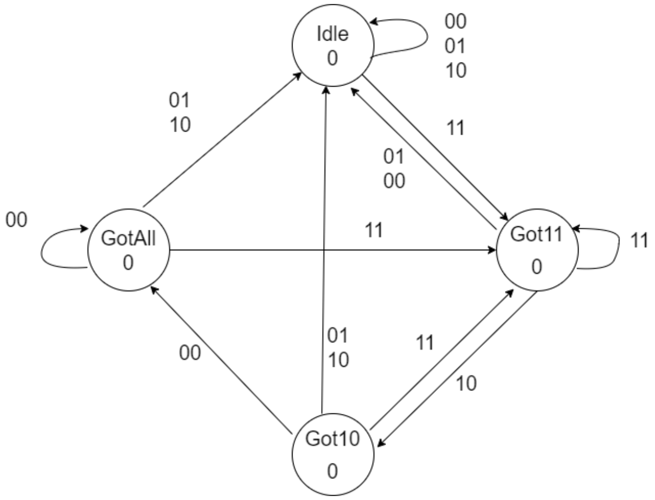

Figure 1: State diagram for a sequence detector

The state diagram shown in Figure 1 shows a representation of how the sequence detector works. After reset, the first value of a sequence starts at Idle, 00. If the inputs A and B are 1 and 1, then the state will move on to Got11, 01. If any other values of A and B are given, then the state will stay at Idle. This is repeated at every state until the sequence of inputs is 11 10 00.

Table 1: State transition table for a sequence detector

State	Q1	Q0	00	01	11	10	Z
Idle	0	0	00	00	01	00	0
Got11	0	1	00	00	01	11	0
Got10	1	1	10	00	01	00	0
GotAll	1	0	10	00	01	00	1

Table 1 shows the state transition table, which describes the sequence in a table. The values of Figure 1 and Table 1 align and both can be used to find the sequence that will return an output on 1.

$$Q0* = AB + Q1'Q0A \tag{1}$$

To find the value of the next Q0, (1) was simplified from Table 1 and used with the current inputs to find the value of Q0.

$$Q1* = Q1A'B' + Q1'Q0AB' \tag{2}$$

Similar to the next Q1 value, (2) was used with the current inputs and the state values to determine the value of Q1. Put together, after a rising edge clock trigger, the current state of Q1 and Q0 were changed to the new values given the inputs and the previous values of Q1 and Q0.

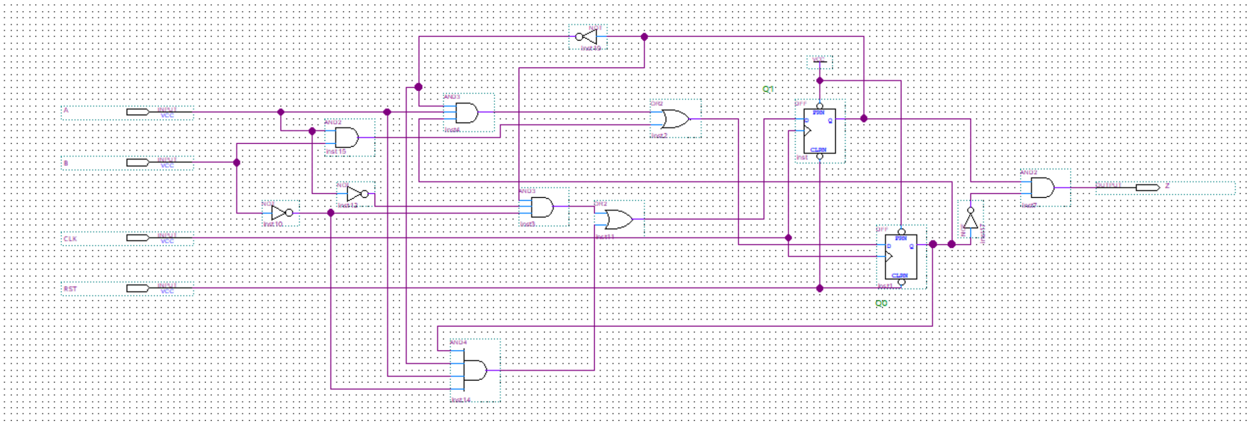

Figure 2: Circuit diagram of the sequence detector

Figure 2 shows the model of the sequence detector. As the output Z is GotAll, the value of Q1 needs to be 1 and the value of Q0 needs to be 0. The results of this circuit will give the same outputs as Table 1 and Figure 1.

Results and Analysis

The results of the simulation of the sequence detector aligned with both the transition table shown in Table 1 and the state diagram shown in Figure 1.

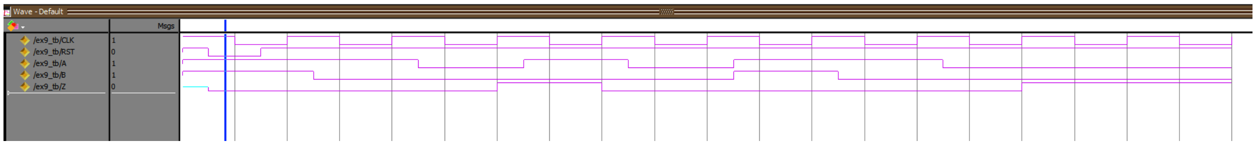

Figure 3: Waveforms for the test bench

The inputs of (A,B) for the test bench that led to the waveforms shown in Figure 3 were (1,1), (1,0), (0,0), (1,0), (0,0), (1,1), (1,0), (0,0), (0,0). These values of AB followed the sequence detector as when the sequence (1,1), (1,0), (0,0), was detected, then Z went to 1 until AB was no longer 00.

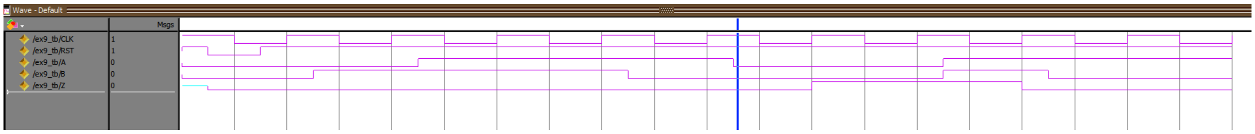

Figure 4: Waveforms for the test sequence

When the sequence detector was checked with a different set of values, (0,0), (0,1), (1,1), (1,1), (1,0), (0,0), (0,0), (1,1), (1,0), then the simulation shown in Figure 4 behaved the same way that the waveform did in Figure 3. If the correct sequence, (1,1), (1,0), (0,0), was found then Z became 1.

$$\begin{gathered} Tmin = tco + tpd + ts \\ 8.353ns = 6.114ns + 0ns + 2.239ns \qquad (3) \\ 1/8.353ns = 0.1197x10^3 = 119.7MHz \end{gathered}$$

Shown in (3), the minimum clock delay of the sequence detector circuit was 8.353ns, and the maximum clock frequency for the sequence detector circuit was 119.7MHz. Tmin represents the minimum clock period that the circuit can operate correctly at, tco is the Clock to Output time, tpd is the propagation delay, which is 0ns, and ts is the setup time.

Conclusion

The functionality of a Moore state machine was implemented by modeling and simulating a sequence detector. This implementation was successful as the expected results aligned with the results found. This was true for two different sequences of inputs.

Questions

Q1. This would change the encoding from grey to binary and change the equations, state diagram and the circuit. The results would be the same.

Q2. Using binary, there are 16 different ways to encode a state machine with four states.

Q3. One-hot encoding uses only one bit that shifts to the left 1 bit each time a value of 1 is added to it.

Exercise 9: Design and Simulation of a Moore State Machine

Student's Name: Oliver Vinperas Section: 3

Prelab		Point Value	Points Earned	Comments
Prelab	State diagram	5	5	HB 03/16/2022
	State table	5	5	
	Equations	5	5	
	Circuit diagram	5	5	HB 03/16/2022

Demo		Point Value	Points Earned	Date
Demo	Part 1 simulation	15	15	HB 03/16/2022
	Part 2 simulation	15	15	HB 03/16/2022
	Timing analysis	10	10	HB 03/16/2022

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		