

Study of sequential sequence detectors report sample

[Design](#)



Module 6

Study of Sequential Sequence Detectors

ELEC201

Digital Electronics

Partner Name:

Introduction

Circuits whose state transitions are controlled by a clock pulse are called synchronous sequential circuits. Other circuits whose state transitions are not controlled by a clock signal are called asynchronous circuits. This lab aims at equipping students with knowledge on synchronous sequential machines.

These are machines designed to solve a problem, which requires the output, to be some specific value after a given sequence of input values has been observed. This problem led to the development of two types of machines in the mid-1950s. These machines are Mealy machines and Moore machines which are named after their inventors.

In Mealy machines, the output depends on both the present state and the current input. The output in Moore machines depends on the present state only. The present state of the circuit is the state at any given clock pulse. The present state results from past inputs. The signal present at the input terminals is the current input.

These machines are used to implement sequence detectors. Sequence detectors are sequential logic circuits used to check whether a certain pattern of serial data has been received. When the circuit detects the required sequence of bits, its output is set high.

<https://assignbuster.com/study-of-sequential-sequence-detectors-report-sample/>

In the given case, the circuit checks for the input sequence $x = 1111$. There are no strict rules that are defined for drawing state diagrams. Experience is a crucial factor in successfully drawing an economical diagram. The problem presented requires the circuit to check for a sequence of 4 bits. Therefore, an assumption is made that the machine will pass through 4 states. Having fewer states ensures that the resulting circuit is economical.

Define the first state by drawing a circle and label it A. This is the state of the circuit at rest.

Figure 1

Find the next state by following the sequence 1111 (the desired input sequence). When a 1 arrives at the input, the circuit goes into the next state B. The output will be 0. This can be represented as

1/0

Figure 2

The notation 1/0 indicates that the 1 is the input and 0 is the output as the circuit changes from state A to state B.

When a 1 arrives as the next input, the circuit changes to state C.

1/00

Figure 3

The arrival of another 1 at the input propels the circuit to state D.

1/0 1/0 1/0

Figure 4

The arrival of the fourth consecutive 1 results in state E, and the output changes to input 1.

1/0 1/0 1/0 1/1

Figure 5

In constructing the above state diagram, an assumption was made that the input sequence is 1111. The next step is defining the state into which the circuit will go if inputs are invalid.

If the first input is 0, the circuit will remain in state A and the output will be 0.

Figure 6

If the circuit is in state B, and a 0 arrives at the input, the input sequence is now 10, which is invalid. What will be the next state? The circuit could go to another state F, but that is uneconomical. Then, let the circuit go back to state A. This is uneconomical as the circuit will have to go through the first state again. The best option is to make the circuit stay at state B.

Figure 7

If again the input sequence is an invalid 10, the best option is to make it remain at state C.

The same should be applied for input 110. The circuit should remain in state D.

Figure 8

The circuit allows overlapping. Therefore, state E cannot be taken to be the final state. From state E when the input is 1, let the machine go back to state D. This will become the first bit of the next sequence of 1111. If the input is 1110, the circuit should go back to the rest state A as this is an invalid

sequence. The final state diagram is as shown below:

Figure 9

A state table is a representation of a state diagram in tabular form. It helps in the assignment of states.

The table shows the outputs for each transition of the states.

Flip-flops Excitation Tables

There are four states, A to B, B to C, C to D and D to E. Assign each state a binary equivalent code. Since they are four, 2 bits can be used to identify each state. Thus 00, 01, 10, and 11 are assigned to each state.

The table above can be reduced to

One can see that the table above represents the present and the next states of the flip-flops used in designing the machine.

K-map Simplification

$$Y1 = y_1y_2 + xy_1 + xy_2$$

$$Y2 = y_2 + xy_1y_2$$

$$Z = xy_1y_2$$

The K-maps simplification yields the following equations.

$$Y1 = y_1y_2 + xy_1 + xy_2$$

$$Y2 = y_2 + xy_1y_2$$

$$Z = xy_1y_2$$

The resulting circuit diagram is

Figure 10

MultiSim Simulation

Simulate the above circuit using MultiSim.

Load the 7474N D-type flip-flop from the miscellaneous digital parts bin.

Connect all the logic gates to form the circuit shown in Figure 10.

Run the simulation.

It is seen that when the input x, represented by key " Space" is low, the output remains low.

Figure 11

If the input is high, the output Z remains high.

Figure 12

Set the word generator outputs to be as follows.

Figure 13

The word generator is set to produce an output pattern of bits from 0000 to 1111. To do so, it is set as an up-counter with 16 decimal states. However, the word generator produces a parallel pattern of bits while the sequential detector needs a serial input. A parallel to serial circuit is required to convert the input to serial data.

Conclusion and Discussion

The objective of the lab was to design a synchronous sequential detector to detect the sequence 1111. The design was carried out using state diagrams and state tables, and the equations were simplified using Karnaugh maps. The resulting circuit was then simulated using NI MultiSim simulating tool using D-type flip-flops as required. The characteristics of a sequence

detector were observed. When the input sequence was as desired, the output became 1.

The lab was successfully carried out, and the objectives met.

References

Fuhrer, R. M., & Nowick, S. (2001). Sequential optimization of asynchronous and synchronous finite-state machines: Algorithms and tools. Boston: Kluwer Academic Publishers.

Nair, B. S. (2006). Digital electronics and logic design. New Delhi: Prentice-Hall of India.