Comparison Between Mealy and Moore State Models UsingSequence Detector with VHDL Coding Techniques

Mr. TunTun Win

***Abstract*–The automated machine modeling is the essential part for developing proposed model using concept of Finite State Machine (FSM) and also reduces the human resources. FSM VHDL design consists of the modeling issues such as state encoding schemes, VHDL coding style for the sequence detector circuit. This systemconsists of two portions: Mealy state model and Moore state model. The designs of state diagrams, state tables, logic expression and logic diagrams in both modelsare accomplished for sequence detector. The logic circuit diagrams in both types are firstly simulated by using Multisim to predict output timing results of these design models.And then, the process programming for circuit designs of Mealy and Moore state modelsaredeveloped and written in VHDL coding.Then, they are implemented by using Modelsim software to achieve the output timing waveforms.Moore FSM of sequence detector has a simplified design at the cost of requiring more states than Mealy FSMs. This paper describes comparison between Mealy and Moore state models using sequence detector with VHDL coding techniques.**

***Index Terms*— Mealy and Moore, modeling issues, sequential circuit, VHDL coding.**

I. INTRODUCTION

This paper introduces the concept of two types of Finite State Machines (FSMs); Mealy and Moore, and the modeling designs to develop such machines. FSM are sequential circuit used in many digital systems to control the behavior of systems and dataflow paths. Sequential circuits are realized using combinational logic and one or more flip-flops. The general structure of such a circuit is shown in Fig. 1.

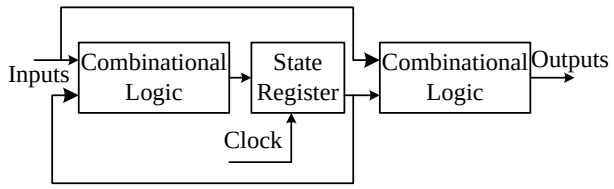

Fig.1. General form of a sequential circuit

The circuit has a set of primary inputs, and produces a set of outputs. The values of the outputs of the flip-flops are referred to as the state of the circuit. Under control of the clock signal, the flip-flop outputs change their state as determined by the combinational logic that feeds the inputs of these flip-flops. Thus the circuit moves from one state to another. To ensure that only one transition from one state to another takes place during one clock cycle, the flip-flops have to be of the edge-triggered type [1]. The sequence detector is considered which output is equal to one when the sequence is 111. The outputis not determined only by the present value of input, so it exist different states in the sequential circuit.

Sequential circuits are also called finite state machines (FSMs) that the functional behavior of the circuits can be represented using a finite number of states [1]. There are two basic ways to design clock sequential circuits. There are using: Moore machine and Mealy machines.

This paper consists of three parts: one for designs of state diagrams, state tables, logic expression and logic diagrams for Mealy and Moore state machines, two for programming of these designs using VHDL codes and other fortesting how to control the processes by using Multisim and Modelsimsoftwares.

II. MEALY STATE MODELING

Fig.2 shows Mealy machine model for the sequential circuit. It consists of theinputs, clock and output for Next state logic, current state register and output logic.So, for Mealy state modeling with VHDL design, itissues to consider the following procedure.State diagram, state table and state assignment models are used to design logic expression.

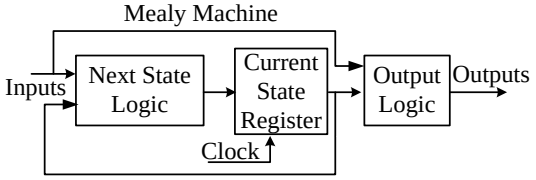

Fig.2. Mealy machine model

A. State Diagram

The first step in designing a finite state machine is to determine how many states are needed and which transitions are possible from one state to another. Fig.3 shows the state diagram of the sequential circuit which depicts states of the circuit as nodes (circles) and transitions between states as directed arcs. States St0, St1 and St2 appear as nodes in the state diagram.

Manuscript received July, 2018.

***Mr. TunTun Win**, Lecturer, Department of Electronic Engineering, Technological University (Mawlamyine), Mawlamyine, Myanmar, Mobile No : +959261542990.*

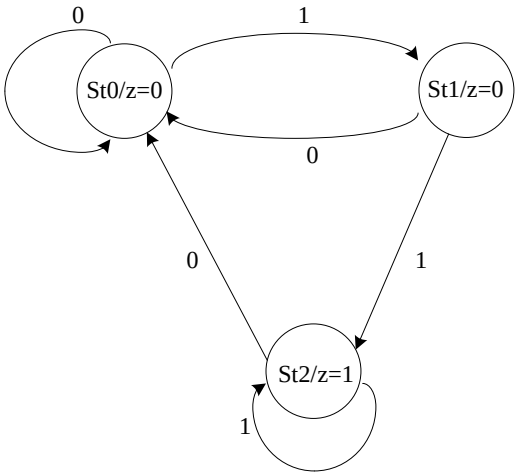

Fig.3.State diagram of Mealy type sequence detector

B. State Table

The description of the state table is easy to understand, to proceed with the implementation of the sequential circuit.Table I indicates all transitions from each present state to the next state for different values of the input signal with three conditions.

Table I. State table for Mealy circuit

Present State	Next State		Output
	x = 0	x =1	z
St0	St0	St1	0
St1	St0	St2	0
St2	St0	St2	1

Table II. State assigned table for Mealy circuit

Present State	Next State		Output
AB	x = 0	x =1	z
00	00	01	0
01	00	10	0
10	00	11	1
11	dddd		d

Table II shows state assigned table for Mealy circuit.Implementing the logic circuit, each state is represented by particular valuation of state variables. Each state variable is implemented in the form of a flip-flop. So, two state variables are used to sufficient for three states. These variables are A and B.The signals A and B are also fed back to the combinational circuit that determines the next state of the FSM. This circuit also uses the primary input signal x and outputs are two signals, D_A and D_B, which are used to set the state of the flip-flops. The flip-flops change their state to the values.

C. Logic Expressions and Logic Diagram

From the state-assigned table, the logic expressions would be derived for the next-state and output functions. D-type flip-flops are used in this circuit, because the values of D_A and D_B are simply clocked into the flip-flops to become thenew values of A and B. The inputs to the flip-flops are called D_1 and D_2, then these signals are D_A and D_B.

The logic circuit is derived by using Karnaugh maps to make it easy to verify the validity of the expressions:

$$D_A = x(A+B) \quad (1)$$
$$D_B = x\overline{A}\overline{B} \quad (2)$$
$$z = A \quad (3)$$

Since $D_1 = D_A$ and $D_2 = D_B$, the logic circuit is implemented as shown in Fig.4 that corresponds to the preceding expressions.

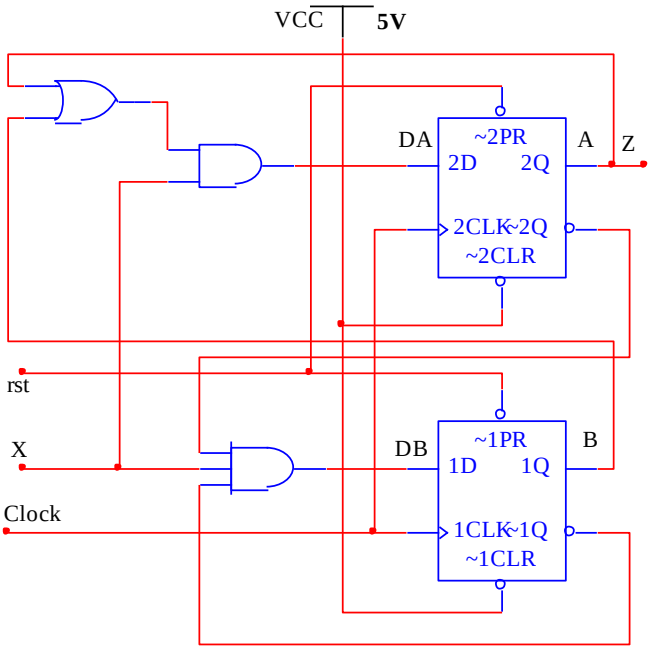

Fig.4. Logic diagram of Mealy sequence detector

There is no reset condition in the state machine that employs two flip-flops. This means that the state machine can enter its unused state '11' on start up.The positive edge triggered flip-flops are used.

III. Moore State Modeling

Fig.5 shows Moore machine model for the sequential circuit. It consists of theinputs, clock and output for Next state logic, current state register and output logic. So, for Moore state modeling with VHDL design, itissues to consider the following procedure. State diagram, state table and state assignment models are used to design logic expression.

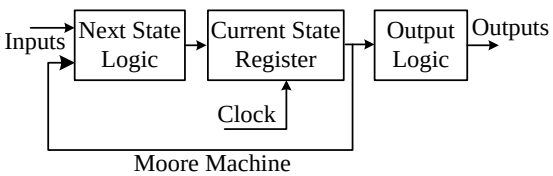

Fig.5. Moore machine model

A. State Diagram

The Moore machine state diagram for '111' sequence detector is shown in Fig.6. States St0, St1, St2 and St3 appear as nodes in the state diagram.

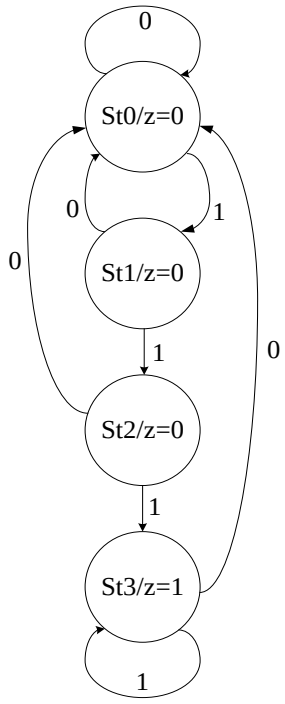

Fig.6.State diagram of Moore sequence detector

B. Stage Table

The state diagram is converted into its equivalent state table as shown in Table III. The states are next encoded with binary values and achieve a state transition table as Table IV.

Table III. State table for Moore circuit

Present State	Next State		Output
	x = 0	x =1	z
St0	St0	St1	0
St1	St0	St2	0
St2	St0	St3	0
St3	St0	St3	1

Table IV. State assigned table for Moore circuit

Present State	Next State		Output
AB	x = 0	x =1	z
00	00	01	0
01	00	10	0
10	00	11	0
11	00	11	1

C. Logic Expressions and Logic Diagram

From the state-assigned table, the logic expressions would be derived for the next-state and output functions. D-type flip-flops are used in this circuit, because the values of D_A and D_B are simply clocked into the flip-flops to become thenew values of A and B. The inputs to the flip-flops are called D_1 and D_2, then these signals are D_A and D_B.

Simplifying Table IV using maps, the following equations are:

$D_A = x(A+B)$

(4)$D_B = Ax+B'x$

(5)z = A B

(6)

The output is a function of present state values only. The circuit diagram for Moore machine circuit implementation is shown in Fig.7. There is no false output in a Moore model, since the output depends only on the state of the flip flops, which are synchronized with clock. The outputs remain valid throughout the logic state in Moore model.

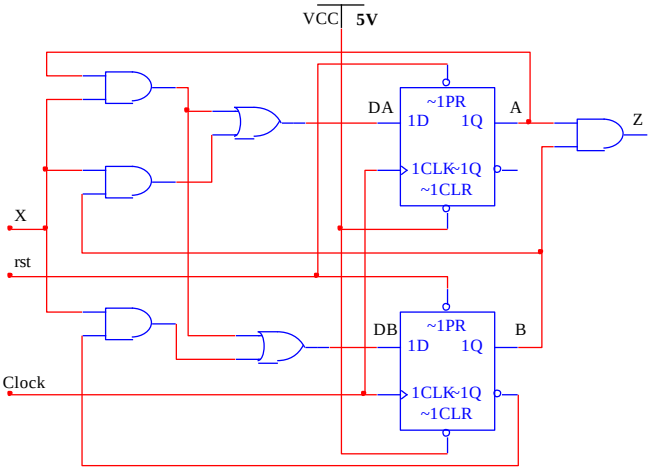

Fig.7. Logic diagram of Moore sequence detector

IV. VHDL CODING FOR MEALY AND MOORE

VHDL is a complex, sophisticated language and is usedfor documentation language for describing the structure of the complex digital circuits and simulation. In this, it is discussed to write VHDL codes for Mealy and Moore sequence detector.

A. VHDL Coding Process for MealySequence Detector

Table V describes VHDL code for Mealy type sequence detector. Firstly, the program introduces the TYPE keyword, which is a feature of VHDL. The TYPE keyword allows us to create a user-defined signal type. The new signal type is named State_type, and the code specifies that a signal of this type can have three possible values: St0, St1, or St2. The zsignal is used in the architecture body to represent the outputs of the flip-flops that implement the states in the FSM.

The PROCESS statement describes the finite state machine as a sequential circuit. The signals used by the process are Clock, x, rst and z. The input signals that can cause the process to change areClock and Rst; hence these signals appear in the sensitivity list.

In other words, the ELSIF condition implies that zcould be implemented as the output of one or more flip-flops. Each

WHEN clause in the CASE statement represents one state of the machine.

Table V. VHDL code for Mealy type sequence detector

Main body	Test bench

architecture Behavioral of mealy is
type state is (st0, st1, st2);
signal present_state, next_state : state;
begin
syncronous_process : process (clk)
begin
if rising_edge(clk) then
if (rst = '0') then
present_state<= st0;
else
present_state<= next_state;
end if;
end if;
end process;
next_state_and_output_decoder : process(present_state, din)
begin
z<= '0'; case (present_state) is
when st0 =>
if (x = '1') then
next_state<= st1;
z<= '0';
else
next_state<= st0;
z<= '0'; end if; when St1 =>
if (x= '1') then
next_state<= st2;
z<= '0';
else
next_state<= st0;
z<= '0'; end if; when St2 =>
if (x = '1') then
next_state<= st2;
z<= '1';
else
next_state<= st0;
z<= '0'; end if; when others =>
next_state<= st0;
z<= '0';
end case;
end process;
end Behavioral;
	Clock period definitions
	constant clk_period : time := 20 ns;
	BEGIN
	-- Instantiate the Unit Under Test (UUT)
	uut: mealy PORT MAP (
	clk =>clk,
	x =>x,
	rst =>rst,
	z =>z,
	);
	-- Clock process definitions
	clk_process :process
	begin
	clk<= '0';
	wait for clk_period/2;
	clk<= '1';
	wait for clk_period/2;
	end process;
	-- Stimulus process
	stim_proc: process
	begin
	rst<= '0';
	wait for 20 ns;
	rst<= '1';
	x<= '0';
	wait for 20 ns;
	x<= '1';
	wait for 20 ns;
	x<= '0';
	wait for 20 ns;
	x<= '1';
	wait for 120 ns;
	x<= '0';
	wait for 20 ns;
	x<= '1';
	wait for 20 ns;
	x<= '0';
	wait for 20 ns;
	x<= '1';
	end process;
	END;

B. VHDL Coding Process for Moore Sequence Detector

There arefour possible values in sequence detector withmooreVHDL code: St0, St1, St2 and St3. The zsignal is used in the architecture body to represent the outputs of the flip-flops that implement the states in the FSM.Table VI describes VHDL code for Moore type sequence detector.

Table VI. VHDL code for Moore sequence detector

Main body	Test bench

architecture Behavioral of moore is
type state is (st0, st1, st2, st3);
signal present_state, next_state : state;
begin
synchronous_process:
process (clk)
begin
if rising_edge(clk) then
if (rst = '0') then
present_state<= st0;
else
present_state<= next_state;
end if;
end if;
end process;
output_decoder : process(present_state, din)
begin
next_state<= st0; case (present_state) is when st0 =>
if (x = '1') then
next_state<= st1;
else
next_state<= st0; end if;
when st1 =>
if (x = '1') then
next_state<= st2;
else
next_state<= st0; end if;
when st2 =>
if (x = '1') then
next_state<= st3;
else
next_state<= st0; end if;
when st3 =>
if (x = '1') then
next_state<= st3;
else
next_state<= st0; end if;
when others =>
next_state<= st0; end case;
end process;
next_state_decoder : process(present_state) begin
case (present_state) is when st0 =>
z<= '0'; when st1 =>
z<= '0'; when st2 =>
z<= '0'; when st3 =>
z<= '1'; when others =>
z<= '0';
end case;
end process;
end Behavioral;
	COMPONENT moore
	PORT(
	clk : IN std_logic;
	x : IN std_logic;
	rst : IN std_logic;
	z : OUT std_logic
	);
	END COMPONENT;
	--Inputs
	signal clk : std_logic := '0';
	signal x : std_logic := '0';
	signal rst : std_logic := '0';
	--Outputs
	signal z : std_logic;
	Clock period definitions
	constant clk_period : time := 20 ns;
	BEGIN
	-- Instantiate the Unit Under Test (UUT)
	uut: moore PORT MAP (
	clk =>clk,
	x =>x,
	rst =>rst,
	z =>z,
	);
	-- Clock process definitions
	clk_process :process
	begin
	clk<= '0';
	wait for clk_period/2;
	clk<= '1';
	wait for clk_period/2;
	end process;
	-- Stimulus process
	stim_proc: process
	begin
	rst<= '0';
	wait for 20 ns;
	rst<= '1';
	x<= '0';
	wait for 20 ns;
	x<= '1';
	wait for 20 ns;
	x<= '0';
	wait for 20 ns;
	x<= '1';
	wait for 120 ns;
	x<= '0';
	wait for 20 ns;
	x<= '1';
	wait for 20 ns;
	x<= '0';
	wait for 20 ns;
	x<= '1';
	end process;
	END;

V. RESULTS

The simulation results describeto compare timing waveforms of Moore and Mealy FSMs programming VHDL code. Firstly, there are simulating the design circuits of Mealy and Moore sequence detector by using

Multisimsoftware. These outputtingtiming diagrams are to provethe same with outputting timing waveforms using VHDL code.

After programming VHDL code for design circuits of sequence detector usingMealy and Moore machines, there are implemented to achieve the simulation timing results by using Modelsim Software. VHDL is the powerful and flexible language code, and then Modelsim software is used to compile and simulate its design. The procedure is certainly seen in each figure of result analysis. Each figure shows different execution of steps. It is little bit lengthy as it was simulated but gives the desired output.

A. Simulation of Mealy Sequence Detector

Fig. 8 shows the output simulation waveform of mealy sequence detector circuit using Multisim. Fig. 9 shows the output simulation waveform of mealy sequence detector using Modelsim.

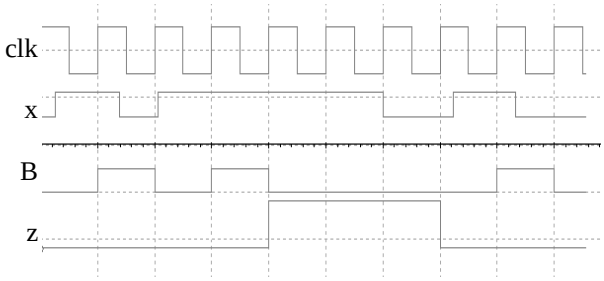

Fig. 8.Timing diagram of Mealy sequence detector circuit using Multisim

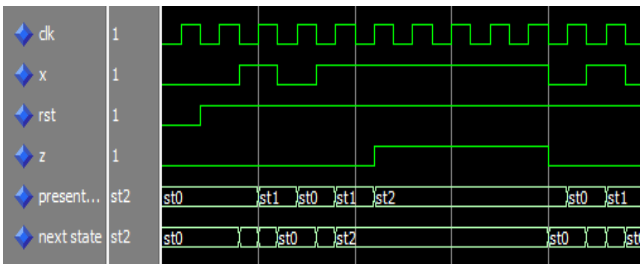

Fig. 9. Timing diagram of Mealy sequence detector using Modelsim

Outputs of the Mealy sequence detector are based on the current state and the current inputs. Each next state is a function of the current state and the current inputs. Observations of the system are that outputs change a*synchronously* to the clock edge because they are affected by changes to the inputs (as well as by changes to the state).Since the output in Mealy model is a combination of present state and input values, an unsynchronized input with triggering clock may result in invalid output, as in the present case.

B. Simulation of Moore Sequence Detector

Timing diagram of Moore sequence detector using Multisim is shown in Fig. 10. The timing diagram for Moore machine model is also shown in Fig.11.

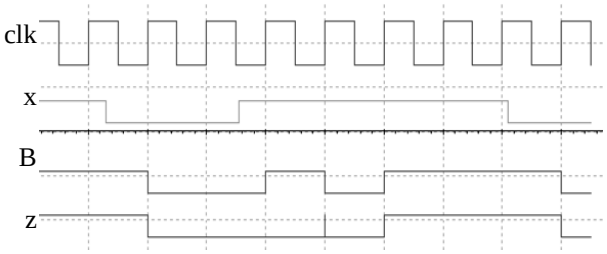

Fig.10.Timing diagram of Moore sequence detector usingMultisim

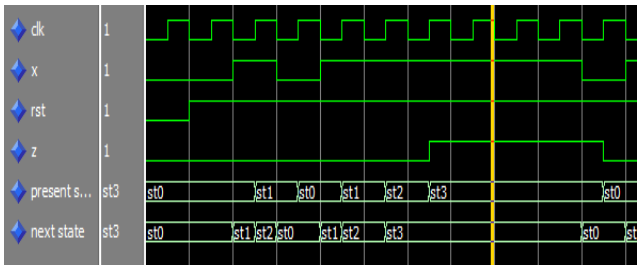

Fig.11.Timing diagram of Moore sequence detector usingModelsim

Outputs of the Moore FSM are determined based on the current state. Each next state is a function of the current state and the current inputs. Moore FSM Observations are which outputs change *synchronously*with the clock edge.There is no false output in a Moore model, since the output depends only on the state of the flop flops, which are synchronized with clock. The outputs remain valid throughout the logic state in Moore model.

VI. Conclusion

The paper describes about comparison between the Mealy and Moore machine types of sequence detector design with VHDL coding techniques. It presents the relationship between FSM and VHDL code with models.A Moore machine produces a unique output for every state irrespective of inputs. Accordingly the state diagram of the Moore machine associates the output with the state in the form state-notation/output-value. The state transition arrows of Moore machine are labeled with the input value that triggers such transition. Since a Mealy machine associates outputs with transitions, an output sequence can begenerated in fewer states using Mealy machine.

By comparing Moore and Mealy FSMs of sequence detector, a Moore FSM has a simplified design at the cost of requiring more states. The Moore FSM has outputs that change synchronously with the clock. Often a Mealy FSM requires fewer states. That can reduce the number of bits requires to store the state number. It is possible to design either a Moore or a Mealy FSM to exhibit identical functionality.

In further extension, the circuit can be designed to control more complex digital counter circuits. The circuit tends to the circuit how might be assigned to the pins in IC package.

ACKNOWLEDGMENT

The author would like to express the heart-felt gratitude to Dr. SeintSeintHtwe, Professor and Head, Electronic Engineering Department, Technological University (Mawlamyine) for her kind advices and encouragements in the route of carrying out this paper. The author wishes to convey his heartily thanks to all his colleagues and to all teachers who had helped towards the successful completion of this paper.

REFERENCES

[1] Stephen Brown and ZvonkoVranesic, "Fundamental of Digital Logic with VHDL Design", Third Edition, The McGraw-Hill Companies, Inc., 2009.

[2] T. R. Padmanabhan and B. Bala Tripura Sundari, "Design Through Verilog HDL", A JOHN WILEY & SONS, INC., 2004.

[3] Volnei and A .Pedroni, "Circuit Design with VHDL", 1MIT Press Cambridge, Massachusetts, London, England, ISBN 0-262-16224-5, pp. 159-186, 2004.

[4] RitikaKalihari, "Comparison Between Mealy and Moore Using Automated Machine", ISSN 2455-5061, Vol.-2, Page.49-55, 2017.

[5] Iuliana CHIUCHISAN, Alin Dan POTORAC and Adrian GRAUR, "Finite State Machine Design and VHDL Coding Techniques", tenth International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, 2010.

[6] VHDL:Programmingby Example, Fourth Edition, Douglas L. Perry, The McGraw-Hill Companies, Inc, Copyright © 2002.

I am TunTun Win, Lecturer, from Technological University (Mawlamyine), Myanmar. I got the degree of Bachelor of Engineering (Electronics) from Mandalay Technological University since 2002. And then I got the degree of Master of Engineering (Electronics) from Yangon Technological University in 2005. I also served as teaching staff at Technological University since 2006. I am very interested in digital Electronics, Microelectronics and Controller programming field etc.