Mealy and Moore Type Finite State Machines

Objectives

- There are two basic ways to design clocked sequential circuits. These are using:
  1. Mealy Machine, which we have seen so far.
  2. Moore Machine.

- The objectives of this lesson are:
  1. Study Mealy and Moore machines
  2. Comparison of the two machine types
  3. Timing diagram and state machines

Mealy Machine

- In a Mealy machine, the outputs are a function of the present state and the value of the inputs as shown in Figure 1.

- Accordingly, the outputs may change asynchronously in response to any change in the inputs.

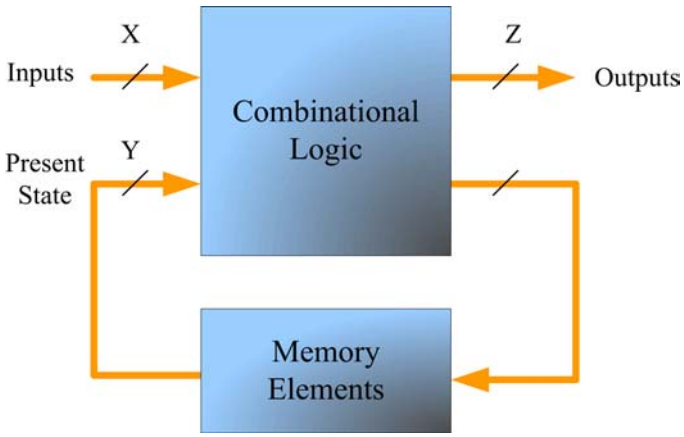

Figure 1: Mealy Type Machine

Mealy Machine

- In a Moore machine the outputs depend only on the present state as shown in Figure 2.

- A combinational logic block maps the inputs and the current state into the necessary flip-flop inputs to store the appropriate next state just like Mealy machine.

- However, the outputs are computed by a combinational logic block whose inputs are only the flip-flops state outputs.

- The outputs change synchronously with the state transition triggered by the active clock edge.

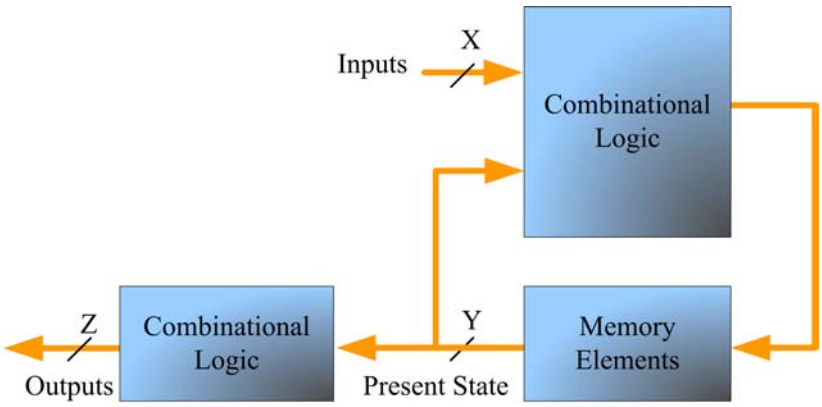

Figure 2: Moore Type Machine

Comparison of the Two Machine Types

- Consider a finite state machine that checks for a pattern of '10' and asserts logic high when it is detected.
- The state diagram representations for the Mealy and Moore machines are shown in Figure 3.
- The state diagram of the Mealy machine lists the inputs with their associated outputs on state transitions arcs.
- The value stated on the arrows for Mealy machine is of the form Zi/Xi where Zi represents input value and Xi represents output value.
- 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 be generated in fewer states using Mealy machine as compared to Moore machine. This was illustrated in the previous example.

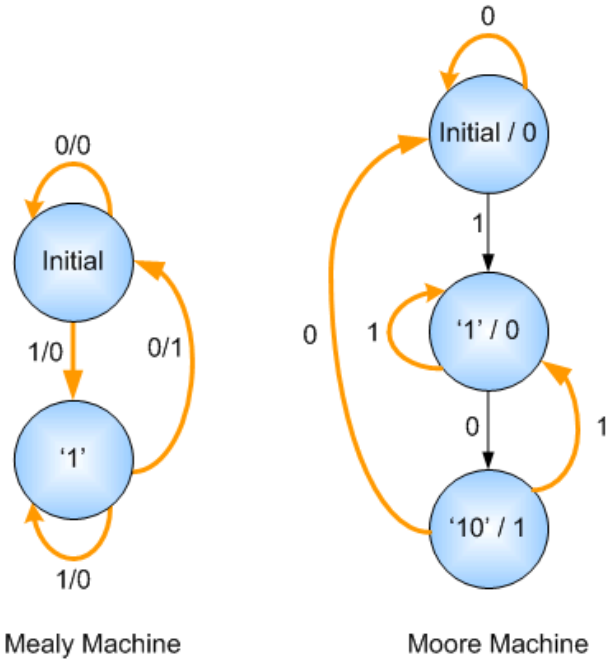

Figure 3: Mealy and Moore State Diagrams for '10' Sequence Detector

Timing Diagrams

- To analyze Mealy and Moore machine timings, consider the following problem. A state-machine outputs '1' if the input is '1' for three consecutive clocks.

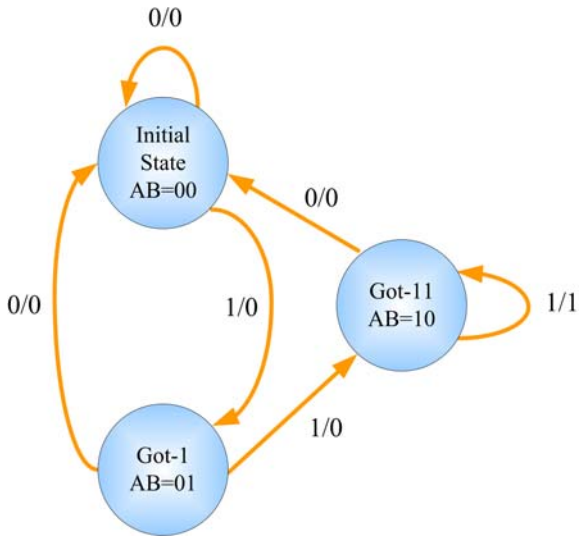

Figure 4: Mealy State Machine for '111' Sequence Detector

Mealy State Machine

- The Mealy machine state diagram is shown in Figure 4.
- Note that 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.
- To make sure that machine gets resetted to a valid state, we use a 'Reset' signal.
- The logic diagram for this state machine is shown in Figure 5. Note that negative edge triggered flip-flops are used.

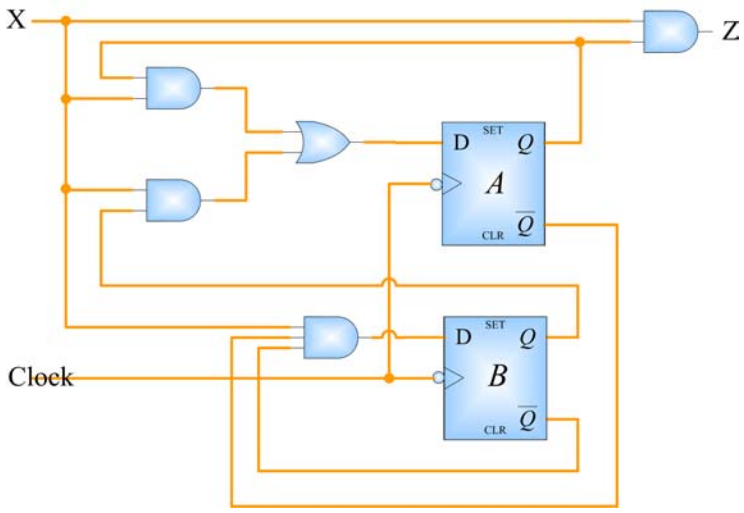

Figure 5: Mealy State Machine Circuit Implementation

- Timing Diagram for the circuit is shown in Figure 6.
- 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.
- Consider the present case where input 'x' remains high for sometime after state 'AB = 10' is reached. This results in 'False Output', also known as 'Output Glitch'.

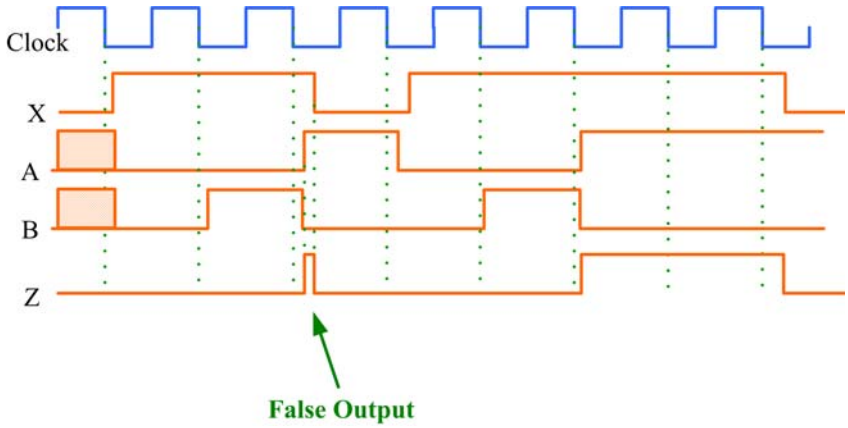

Figure 6: Timing Diagram for Mealy Model Sequence Detector

Moore State Machine

- The Moore machine state diagram for '111' sequence detector is shown in Figure 7.
- The state diagram is converted into its equivalent state table (See Table 1).
- The states are next encoded with binary values and we achieve a state transition table (See Table 2).

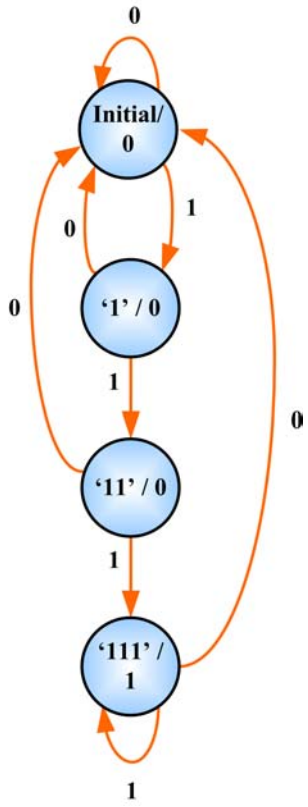

Figure 7: Moore Machine State Diagram

Table 1: State Table

Present State	Next State		Output Z
	x = 0	x = 1	
Initial	*Initial*	*Got*-1	0
Got-1	*Initial*	*Got*-11	0
Got-11	*Initial*	*Got*-111	0
Got-111	*Initial*	*Got*-111	1

Table 2: State Transition Table and Output Table

Present State	Next State		Output Z
	x = 0	x = 1	
Initial	*Initial*	*Got*-1	0
Got-1	*Initial*	*Got*-11	0
Got-11	*Initial*	*Got*-111	0
Got-111	*Initial*	*Got*-111	1

- We will use JK and D flip-flops for the Moore circuit implementation. The excitation tables for JK and D flip-flops (Table 3 & 4) are referenced to tabulate excitation table (See Table 5).

Table 3: Excitation Table for JK flip-flop

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 4: Excitation Table for D flip-flop

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

Table 5: Excitation Table for the Moore Implementation

Inputs of Comb.Circuits			Next State		Outputs of Comb.Circuit			Output
Present State		Input			Flip-flop Inputs			
A	**B**	**X**	**A**	**B**	J_A	K_A	D_B	**Z**
0	0	0	0	0	0	X	0	0
0	0	1	0	1	0	X	1	0
0	1	0	0	0	0	X	0	0
0	1	1	1	0	1	X	0	0
1	0	0	0	0	X	1	0	0
1	0	1	1	1	X	0	1	0
1	1	0	0	0	X	1	0	1
1	1	1	1	1	X	0	1	1

- Simplifying Table 5 using maps, we get the following equations:
  - $J_A = X.B$
  - $K_A = X'$
  - $D_B = X(A + B)$
  - $Z = A \,.\, B$

- Note that the output is a function of present state values only.

- The circuit diagram for Moore machine circuit implementation is shown in Figure 8.

- The timing diagram for Moore machine model is also shown in Figure 9.

- 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.

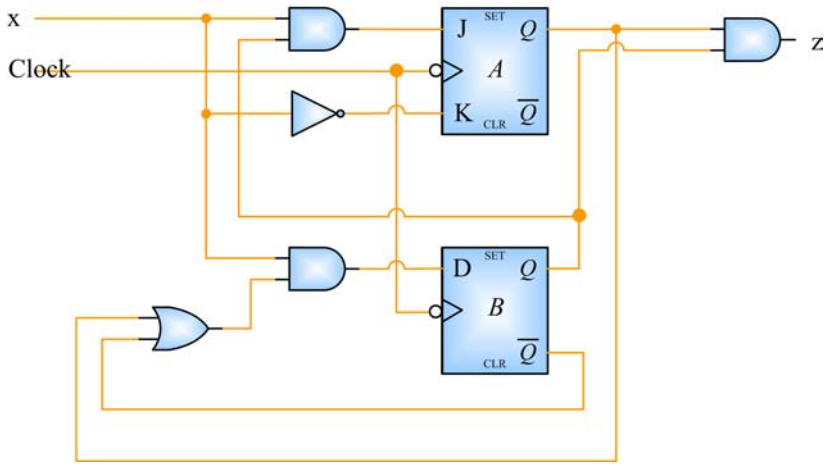

Figure 8: Moore Machine Circuit Implementation for Sequence Detector.

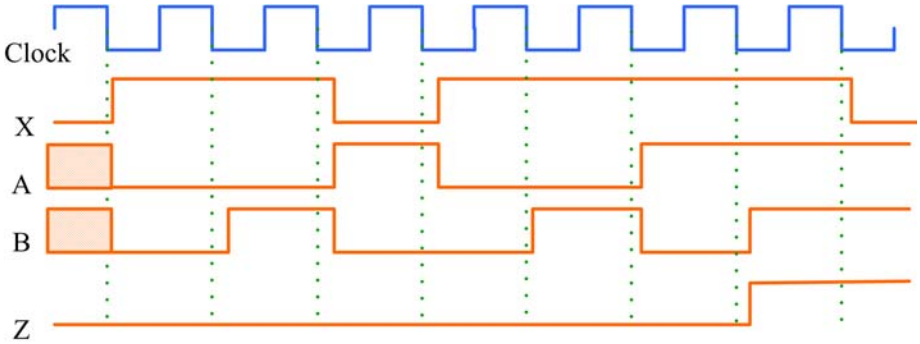

Figure 9: Timing Diagram for Moore Model Sequence Detector.