

1 Finite State Machines

In a manner similar to a synchronous counter, a general FSM (Finite State Machine) may be constructed by determining the *next state* from the *current state* plus, optionally, any other additional inputs that the system may have.

There are two different styles of creating state machines. Moore and Mealy.

1.1 Moore State Machines

In a Moore state machine, the state itself is directly mapped to the output signals.

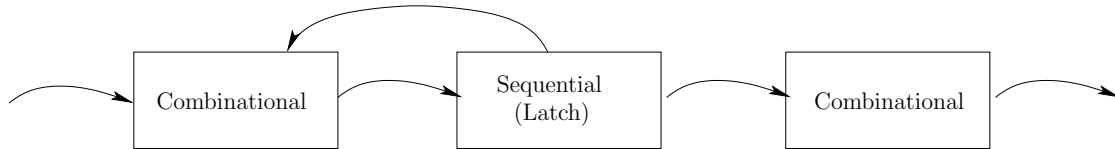


Figure 1: Moore FSM Block Diagram

1.1.1 Synchronous Counter

Consider a synchronous counter with an *enable* input that when high, allows the counter to advance and that when low prevents the counter from advancing.

State table:

Current State		Input E	Next State		Output	
S_1	S_0		N_1	N_0	Q_1	Q_0
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	1
0	1	1	1	0	0	1
1	0	0	1	0	1	0
1	0	1	1	1	1	0
1	1	0	1	1	1	1
1	1	1	0	0	1	1

Boolean functions for the next state:

$$N_1 = (\overline{S_1} \wedge S_0 \wedge E) \vee (S_1 \wedge \overline{S_0} \wedge \overline{E}) \vee (S_1 \wedge \overline{S_0} \wedge E) \vee (S_1 \wedge S_0 \wedge \overline{E}) \quad (1)$$

$$N_0 = (\overline{S_1} \wedge \overline{S_0} \wedge E) \vee (\overline{S_1} \wedge S_0 \wedge \overline{E}) \vee (S_1 \wedge \overline{S_0} \wedge E) \vee (S_1 \wedge S_0 \wedge \overline{E}) \quad (2)$$

$$Q_0 = S_0 \quad (3)$$

$$Q_1 = S_1 \quad (4)$$

The output of this counter is the state itself.

When we draw a Moore style state diagram, the nodes are labeled with the output values from the circuit and the edges include a value representing any inputs.

In Fig 2 the state in the lower-left part of the diagram represents state 11 and shows every possible value of the circuit's inputs (in this case, only E) along the edges that exit from that state. The edge that loops

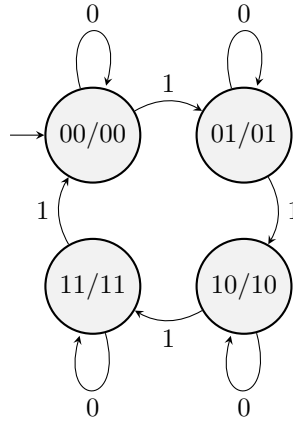


Figure 2: Synchronous Counter With Enable State Diagram

from state 11 back to itself, for example, represents how the circuit will react to the clock signal when E is low. The edge that exits from state 11 to state 00 represents how the circuit will react to the clock signal when E is high.

1.1.2 Message Recognizer

Rather than counting *in order* and providing a mechanism for starting and stopping the counting, a circuit may be created that uses external inputs to change the order in which the counting is done. Additionally, the count value can be used for more than determining the next state.

Consider a circuit designed to recognize a message (011) that arrives over time on signal D as a series of samples taken on the falling edges of a clock signal and generates a high level on signal Q when so recognized.

State table:

Current State		Input	Next State		Output
S_1	S_0	D	N_1	N_0	Q
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	0	1

$$N_1 = (\overline{S_1} \wedge S_0 \wedge D) \vee (S_1 \wedge \overline{S_0} \wedge \overline{D}) \vee (S_1 \wedge \overline{S_0} \wedge D) \quad (5)$$

$$N_0 = (\overline{S_1} \wedge \overline{S_0} \wedge \overline{D}) \vee (\overline{S_1} \wedge S_0 \wedge \overline{D}) \vee (S_1 \wedge \overline{S_0} \wedge D) \vee (S_1 \wedge S_0 \wedge \overline{D}) \quad (6)$$

$$Q = S_1 \wedge S_0 \quad (7)$$

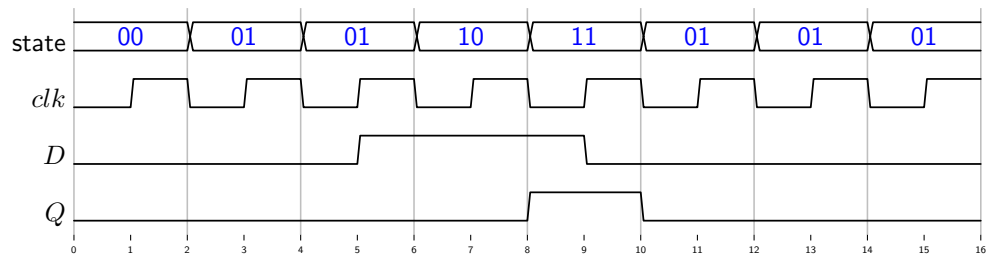


Figure 3: Message 0 1 1 Moore FSM Waveform

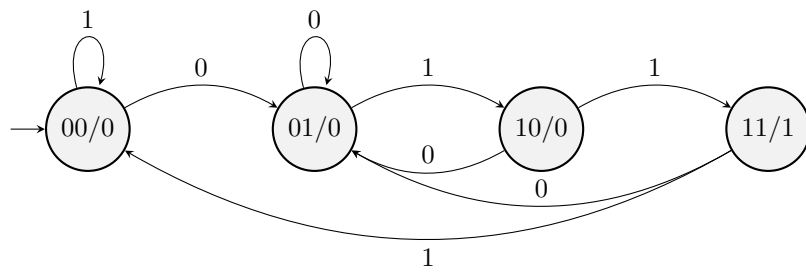


Figure 4: Message 0 1 1 Moore FSM State Diagram

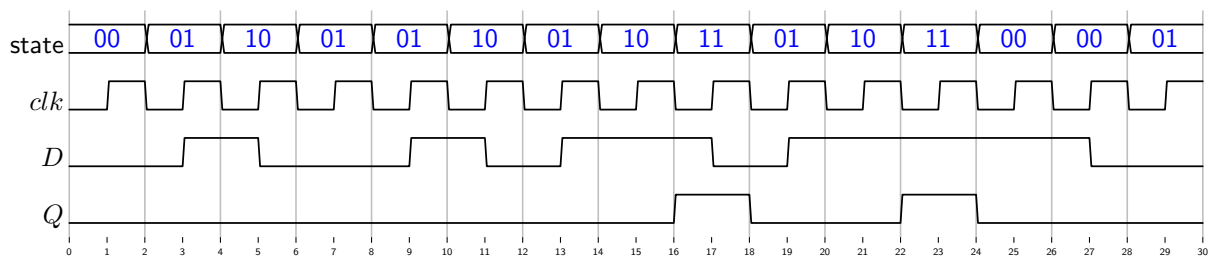


Figure 5: Message 0 1 1 Moore FSM All Possible Transitions

1.2 Mealy State Machines

In contrast to a Moore state machine, a Mealy state machine is allowed to generate output signals from the current state as well as the input signals.

The result is that the outputs from a Mealy state machine can change asynchronously (if/when the inputs change.) Like the ripple counter, this can simplify a circuit as long as the relationship of the inputs and outputs of the circuit is acceptable in the application within which it is used.

One benefit of the Mealy state machine is that it can sometimes require fewer states.

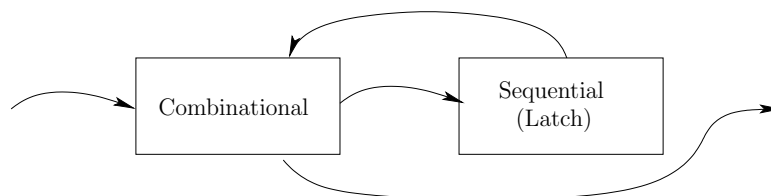


Figure 6: Mealy FSM Block Diagram

1.2.1 Mealy Message Recognizer

Note that the output Q in Fig 5 is high when the state is 11. In Fig 8, Q is high when the state 10 and the input D is high.

State table:

Current State		Input D	Next State		Output Q
S_1	S_0		N_1	N_0	
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1

$$N_1 = (\overline{S_1} \wedge S_0 \wedge D) \quad (8)$$

$$N_0 = (\overline{S_1} \wedge \overline{S_0} \wedge \overline{D}) \vee (\overline{S_1} \wedge S_0 \wedge \overline{D}) \vee (S_1 \wedge \overline{S_0} \wedge \overline{D}) \quad (9)$$

$$Q = S_1 \wedge \overline{S_0} \wedge D \quad (10)$$

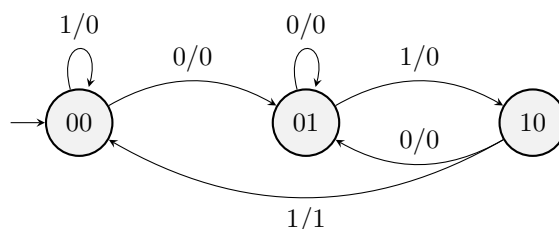


Figure 7: Message 0 1 1 Mealy FSM State Diagram

The output of a Mealy state machine can deliver its output sooner than the Moore state machine. However, the Mealy state machine can also glitch as shown in Fig 9 in ways that the Moore machine will not.

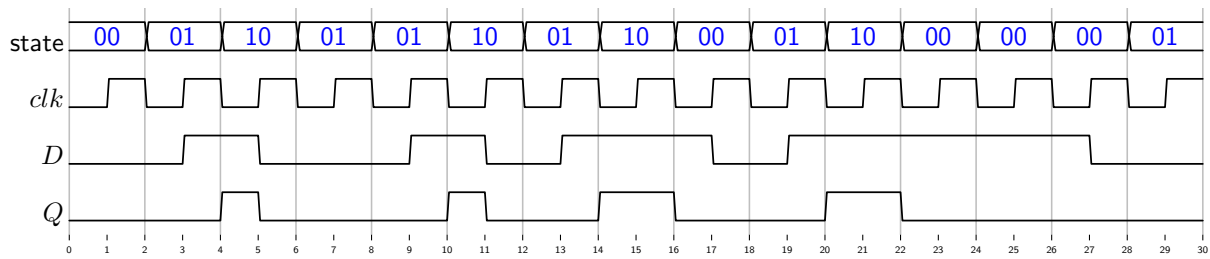


Figure 8: Message 0 1 1 Mealy FSM All Possible Transitions

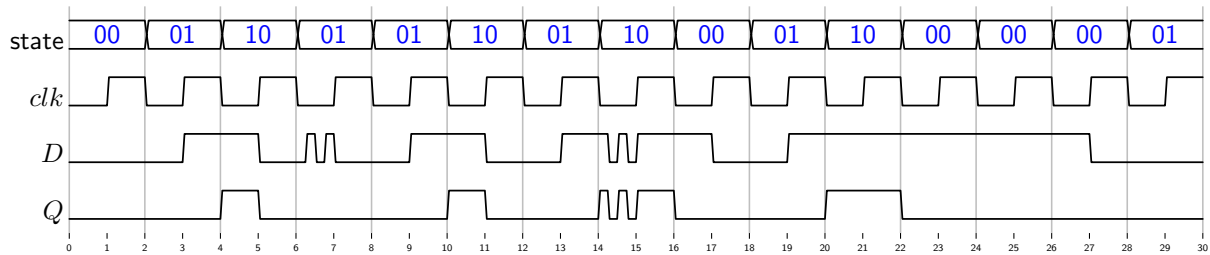


Figure 9: Message 0 1 1 Mealy FSM With Glitches

2 Observations

- All state machines contain:
 1. Latches to hold the current state.
 2. A combinational circuit to determines the next state.
- In some cases a Mealy machine can have fewer states than the equivalent Moore machine.
- Moore machine outputs are driven from the latch signals only.
- Moore machine outputs only change synchronously with the clock signal edge that cause the latches to load.
- Mealy machine outputs are driven from the latch signals and the input signals.
- Mealy machine outputs can change with the clock signal or when an input signal changes.