Introduction to Digital Logic

Lecture 19:
State Machines
State Machine Analysis

Characteristic Equations

- With latches and flip-flops we can come up with an equation for the next value of Q (Q*) in terms of the current value of Q and the inputs

S	R	Q*
0	0	Q
1	0	1
0	1	0
1	1	illegal

Function Table
(Q* listed in terms of Q)

S	R	Q	Q*
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Truth Table
(Q* in terms of 0 or 1)

Characteristic Equations

- With latches and flip-flops we can come up with an equation for the next value of Q (Q*) in terms of the current value of Q and the inputs

S	R	Q*
0	0	Q
1	0	1
0	1	0
1	1	illegal

Function Table
(Q* listed in terms of Q)

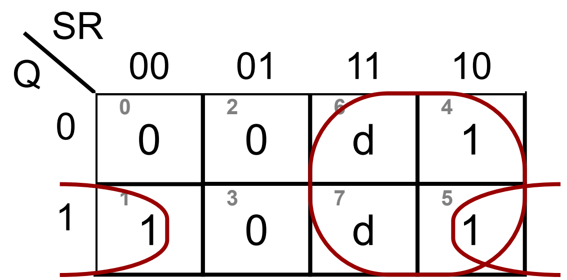
S	R	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	d
1	1	1	d

Truth Table
(Q* in terms of 0 or 1)

Characteristic Equations

- For an SR-Latch make a truth table with S,R, and Q and show the next value of Q*
- The use a K-Map to find an equation for Q*
- This equation indicates what the next value of Q will be

S	R	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	d
1	1	1	d



Q* = S + R'Q

Characteristic Equations

- For a D-Latch make a truth table with D, and Q and show the next value of Q*
- You may use a K-Map but we can eyeball it
- This equation indicates what the next value of Q will be

D	Q	Q*
0	0	0
0	1	0
1	0	1
1	1	1

Q* = D

Characteristic Equation

- For a JK-FF make a truth table with J,K, and Q and show the next value of Q*
- The use a K-Map to find an equation for Q*
- This equation indicates what the next value of Q will be

J	K	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Q \ JK	00	01	11	10
0	(0) 0	(2) 0	(6) 1	(4) 1
1	(1) 1	(3) 0	(7) 0	(5) 1

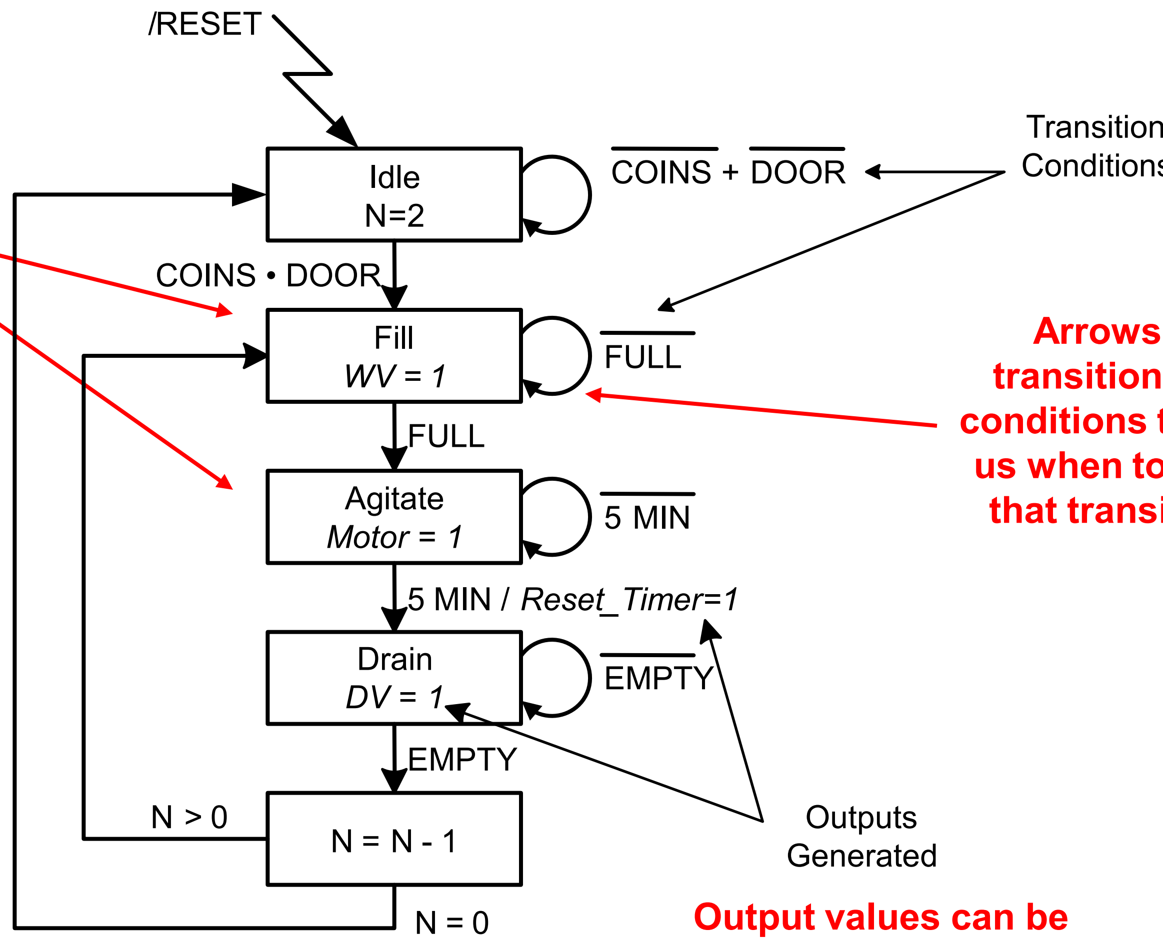
Q* = JQ' + K'Q

State Machines

- Provide the "brains" or control for electronic and electro-mechanical systems
- Implement a set of steps (or algorithm) to control or solve a problem
- **Goal is to generate output values at specific times**
- Combine Sequential and Combinational logic elements
  - Sequential Logic to remember what step (state) we're in
    - Encodes everything that has happened in the past
  - Combinational Logic to produce outputs and find what state to go to next
    - Generates outputs based on what state we're in and the input values
- Use state diagrams (a.k.a. flowcharts) to specify the operation of the corresponding state machine

Washing Machine State Diagram

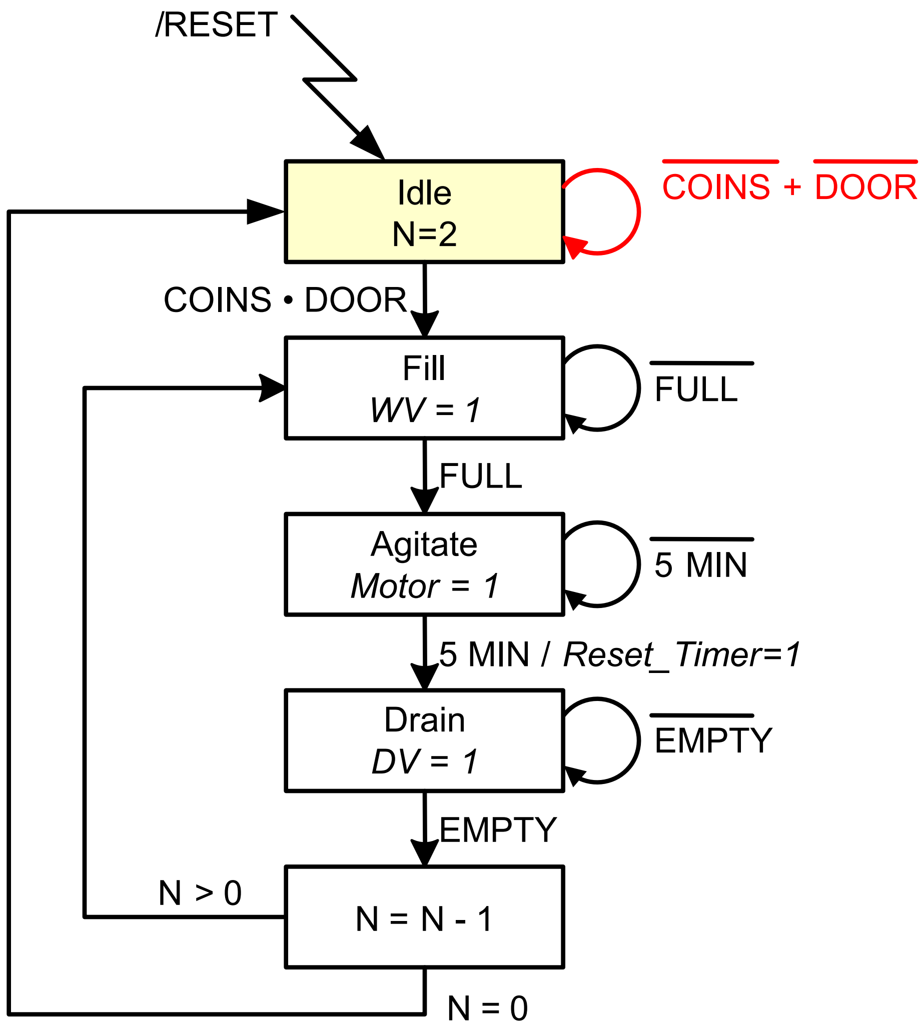
Boxes represent states or steps in the algorithm



Arrows = transitions w/ conditions telling us when to take that transition

Output values can be associated with a state and transition or just with a state

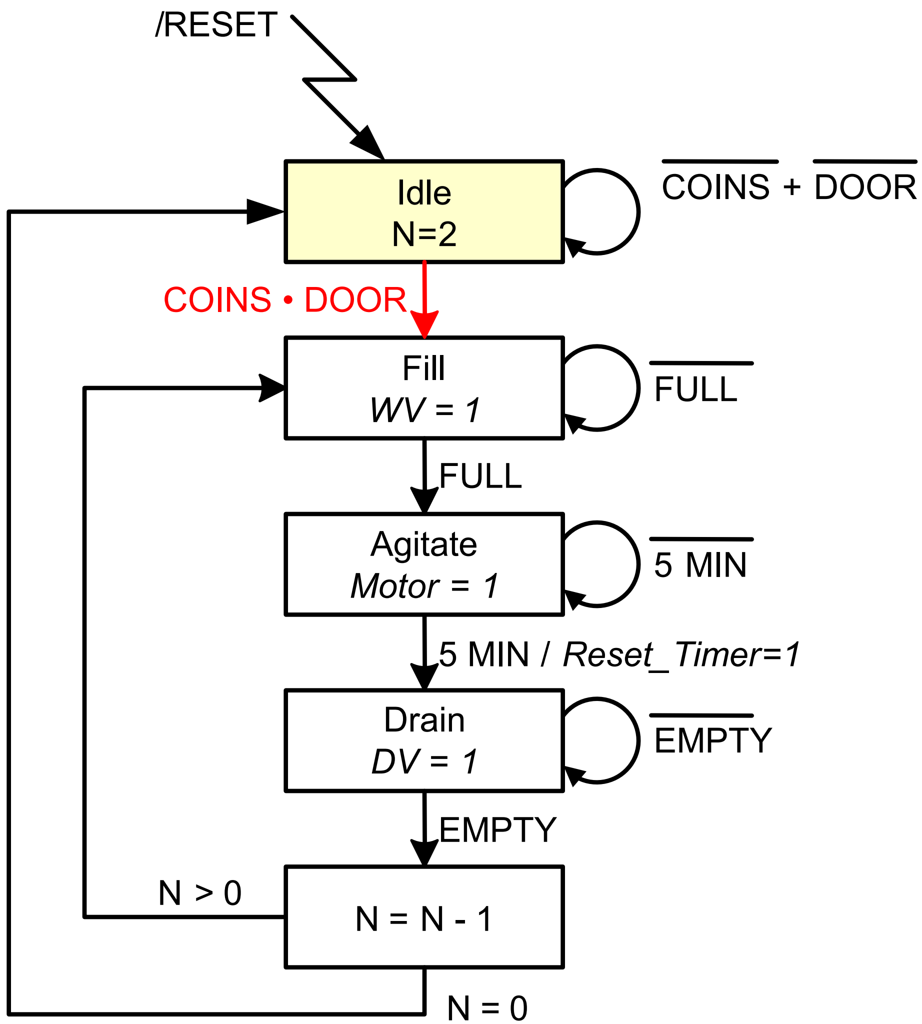
Washing Machine State Diagram



Stay in the initial state until there is enough money (coins) and the door is closed

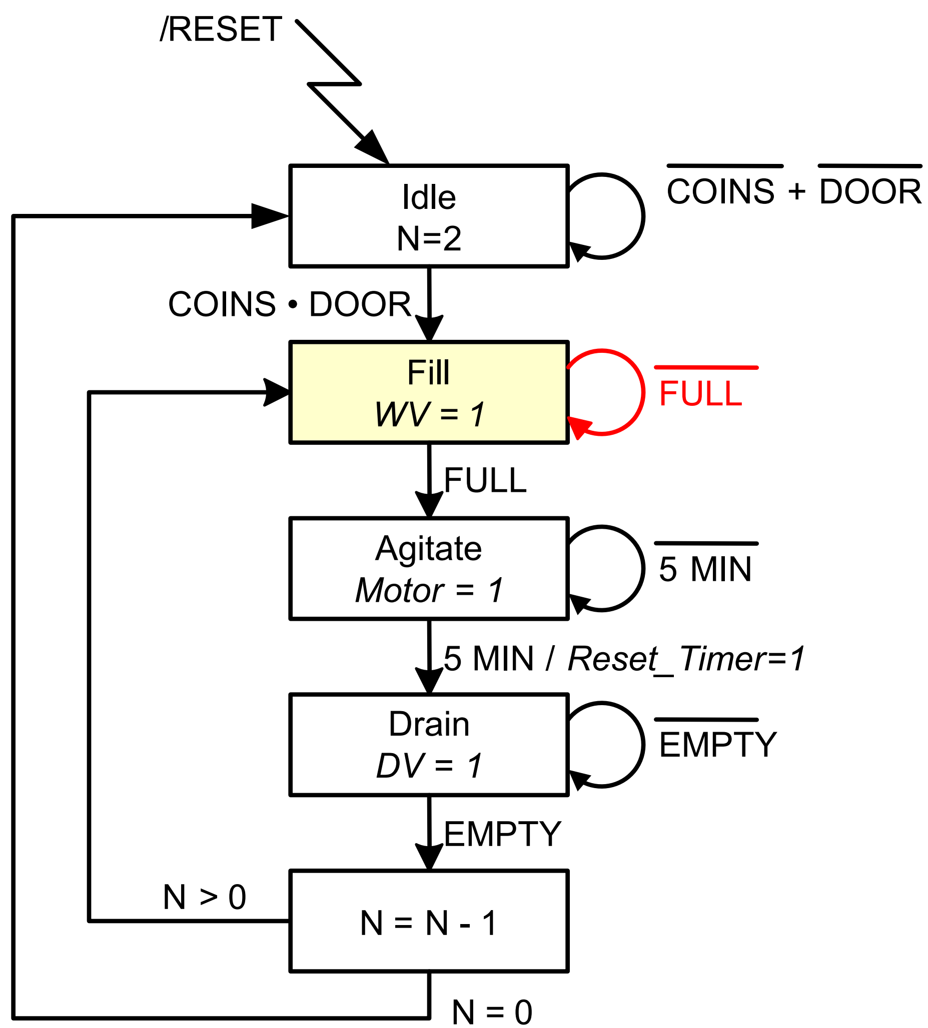
We move through the states based on the conditions. Outputs get asserted when the machine is in that state and the transition is true.

Washing Machine State Diagram



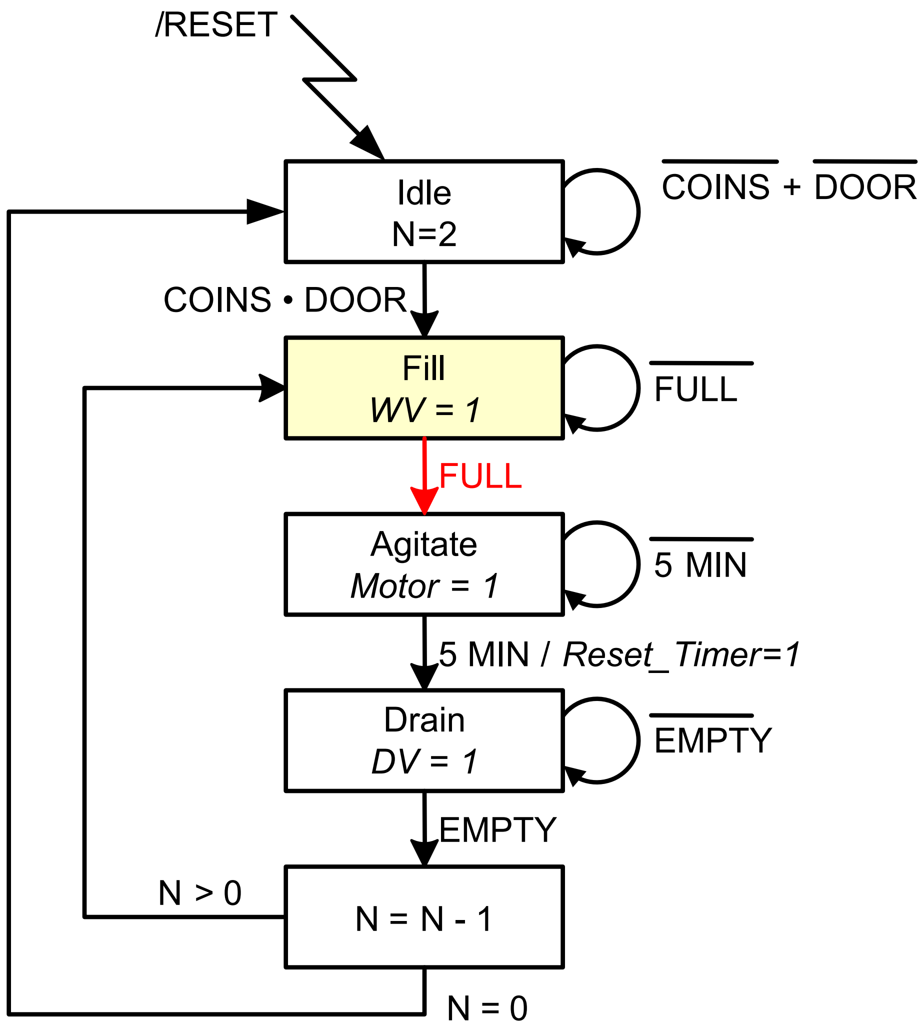
Move to the Fill state when there is enough money (coins) and the door is closed

Washing Machine State Diagram



Stay in the Fill state until it is full…also set the Water Valve Open output to be true

Washing Machine State Diagram



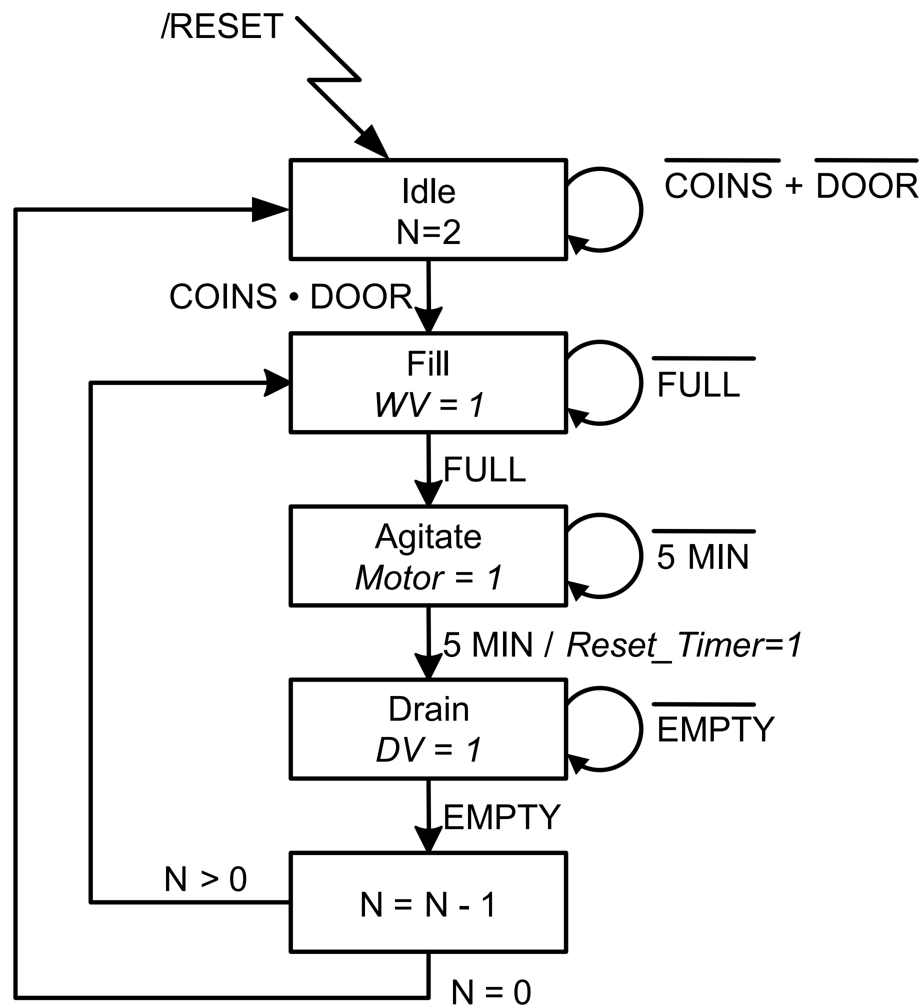
Move to the Agitate state after it is full

State Machines

- Use sequential and combinational logic to implement a set of steps (i.e. an algorithm)
- Goal is to produce outputs at specific points of time
  - Combinational logic alone cannot do that because the outputs will change as soon as the inputs change (no notion of time)

State Diagrams

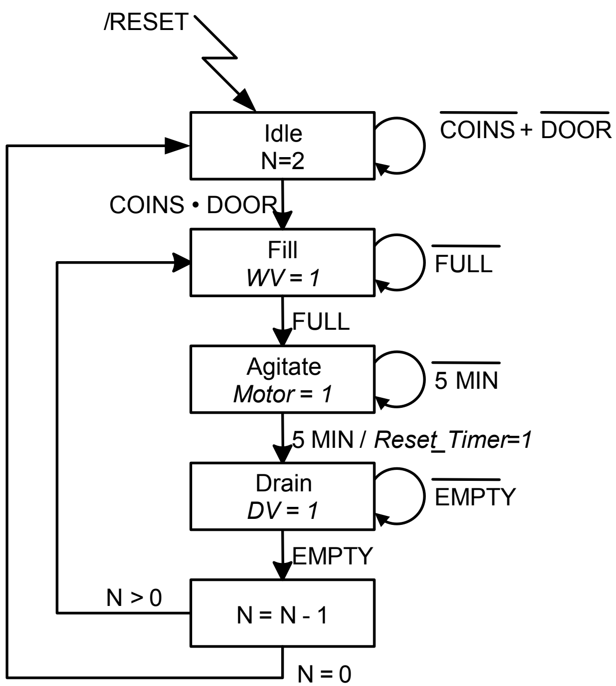
- Used to show operation or function of a state machine
- Like a flowchart but called a state diagram
- 3 parts
  - States
  - Transitions
  - Outputs



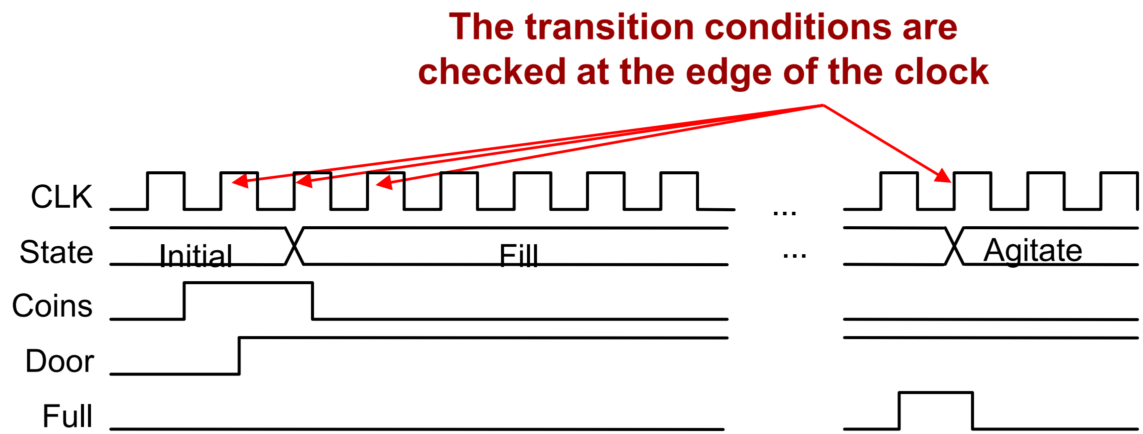
State Diagram for a Washing Machine

State Diagrams

- One transition is made at each clock edge
  - Based on the current state and the conditions associated w/ the transitions

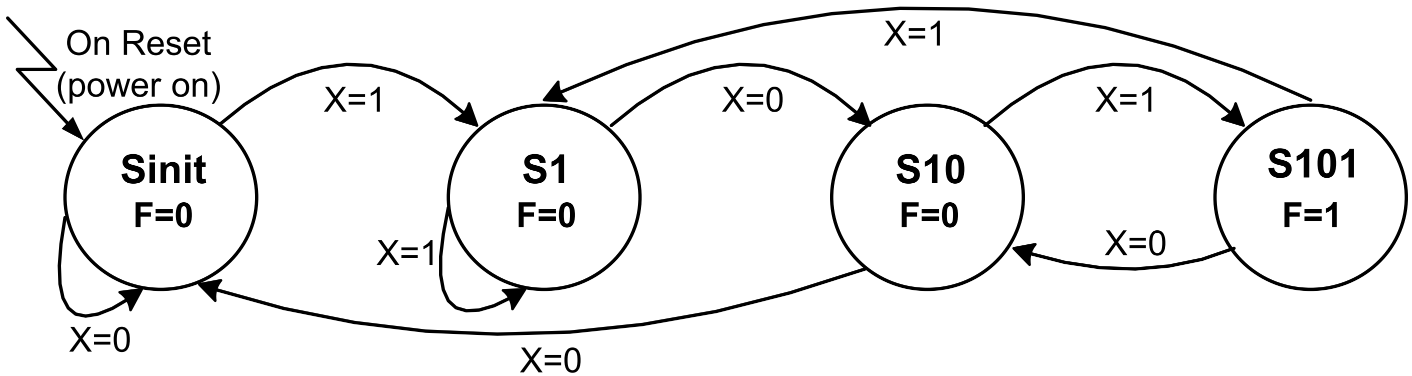


State Diagram for a Washing Machine



Another State Diagram Example

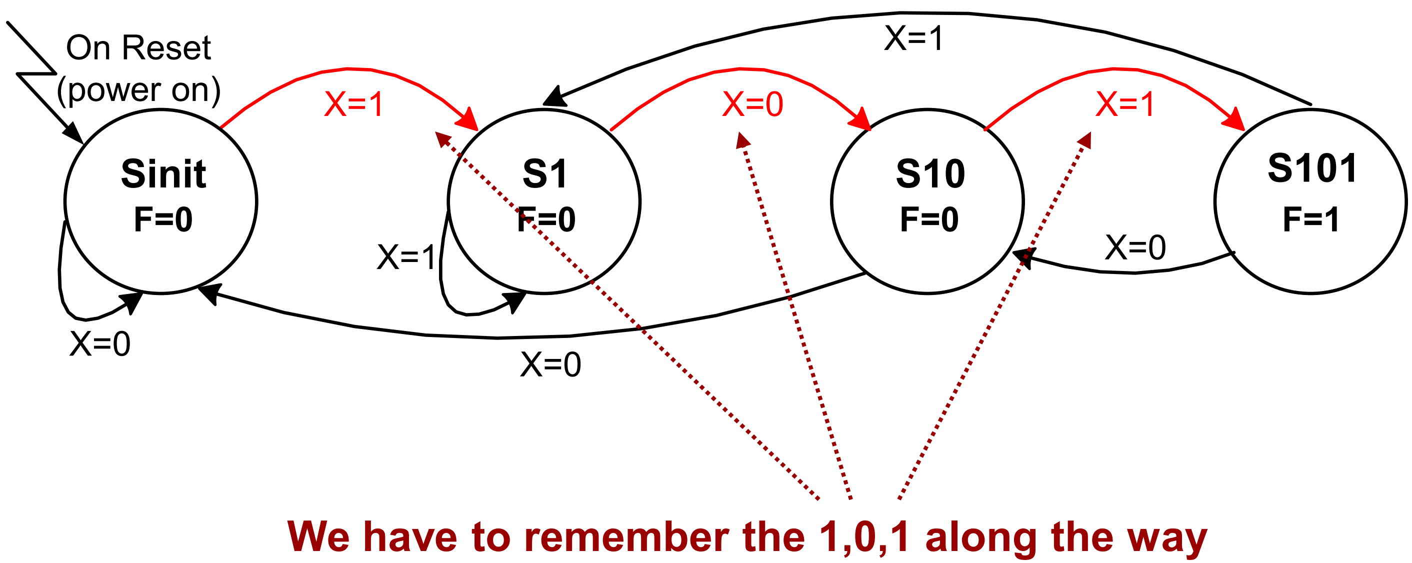
- "101" Sequence Detector should output F=1 when the sequence 101 is found in consecutive order



State Diagram for "101"
Sequence Detector

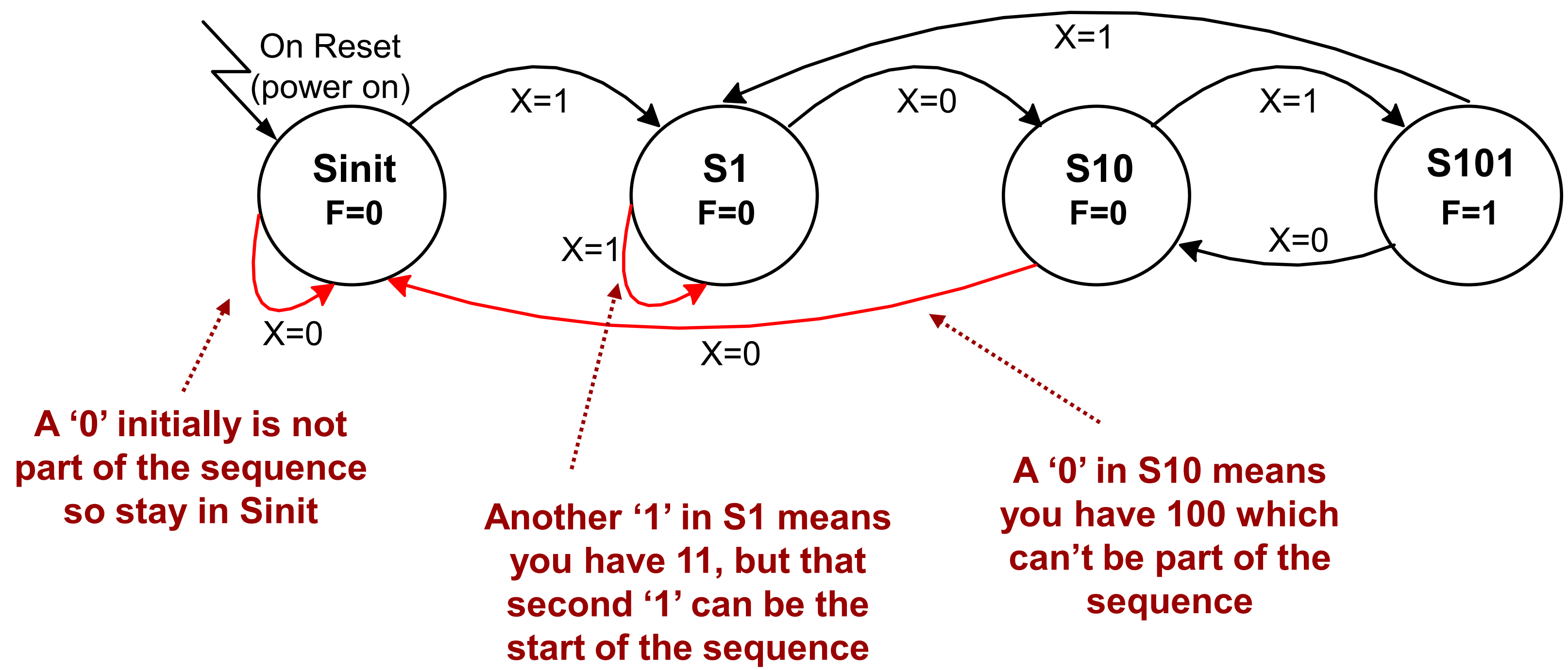
Another State Diagram Example

- "101" Sequence Detector should output F=1 when the sequence 101 is found in consecutive order



Another State Diagram Example

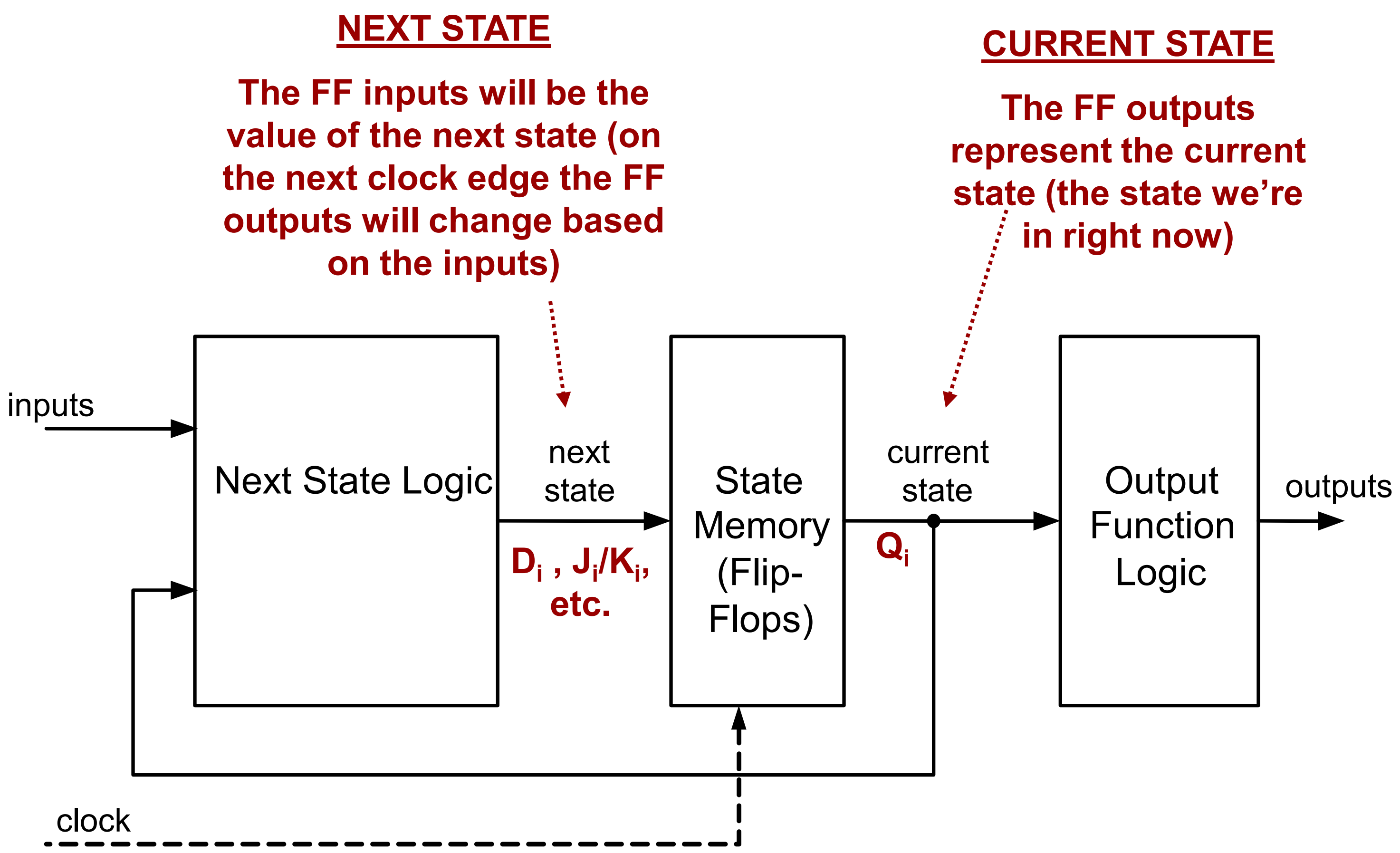
- "101" Sequence Detector should output F=1 when the sequence 101 is found in consecutive order



State Machines

- State Machines can be broken into 3 sections of logic
  - State Memory (SM)
    - Just FF's to remember the *current state*
  - Next State Logic (NSL)
    - Combo logic to determine the next state
    - Essentially implements the transition conditions
  - Output Function Logic (OFL)
    - Combo logic to produce the outputs

State Machine

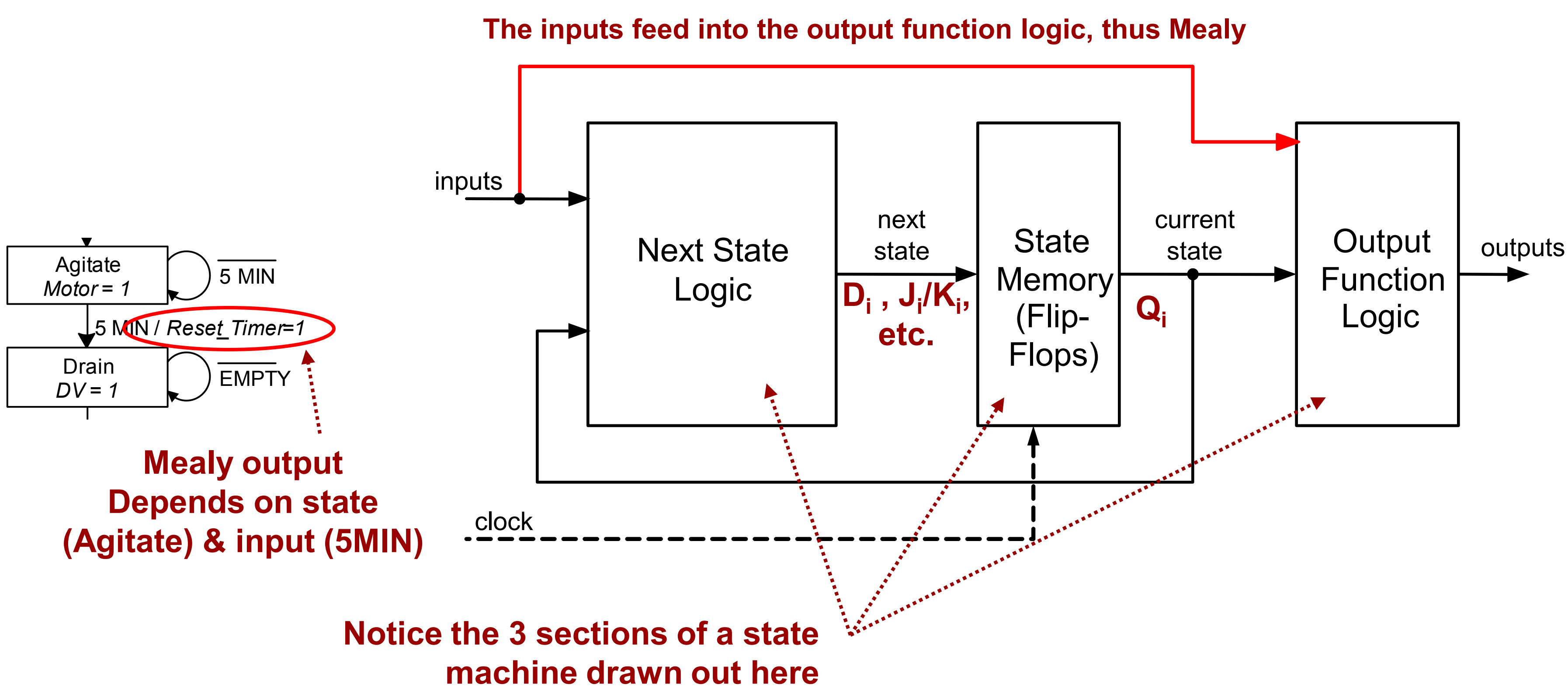


State Machines

- State Machines can be classified according to how the outputs are produced
  - If *Outputs = f(current state, external inputs)*… MEALY Machine
  - If *Outputs = f(current state)*… MOORE Machine

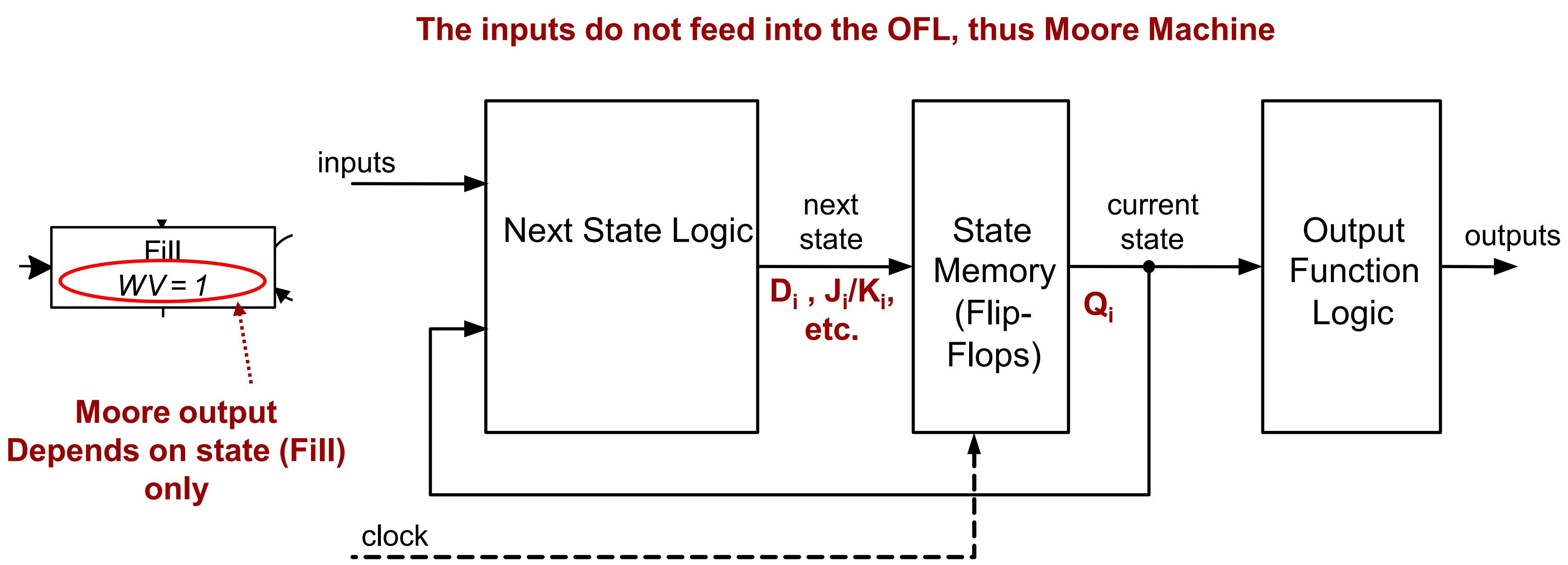
Mealy Machine

- In a Mealy Machine the outputs depend not only on the current state but the external inputs



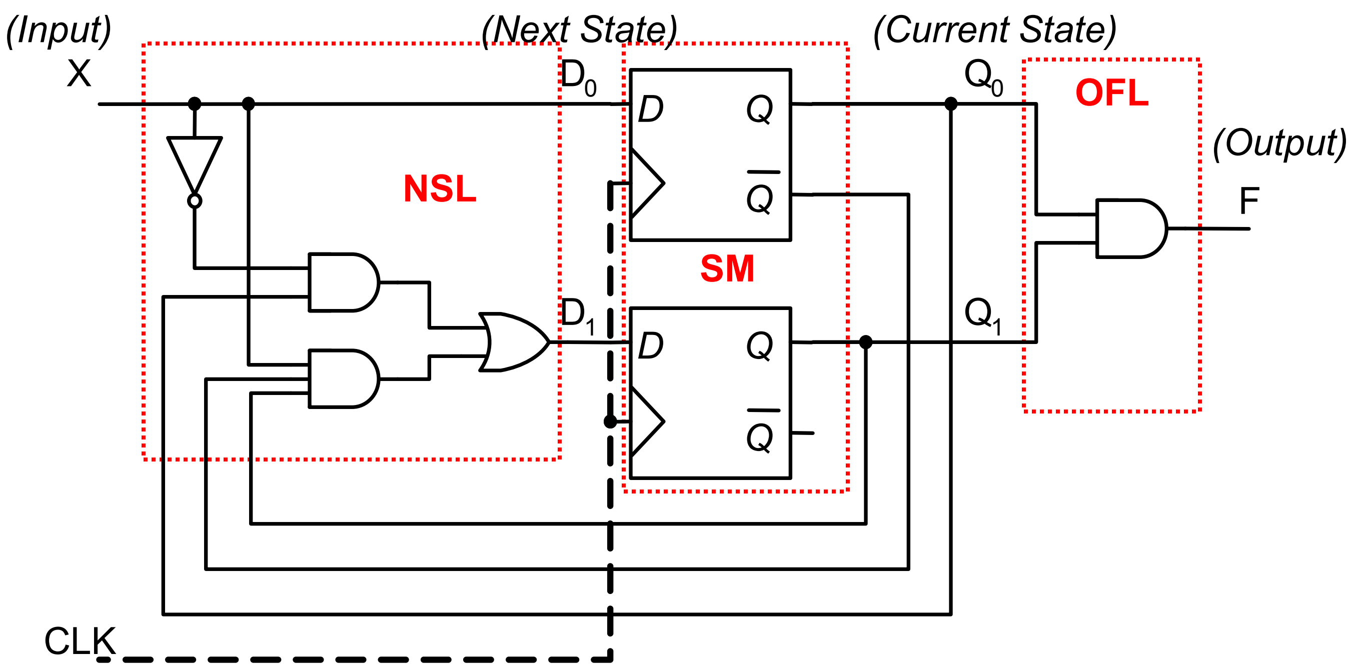
Moore Machine

- In a Moore Machine the outputs only depend on the current state



State Machines

- Below is a circuit implementing a state machine, notice how it breaks into the 3 sections

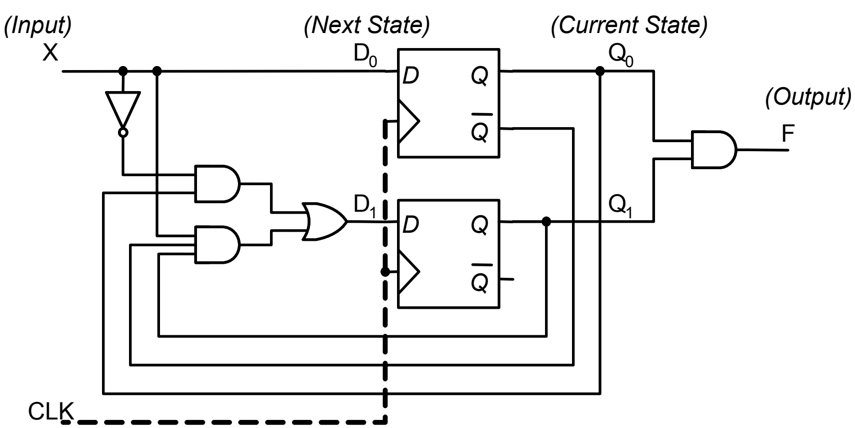


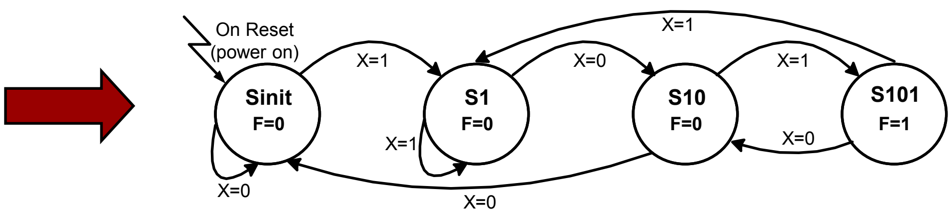
State Machine Analysis

- In state machine analysis our goal is to take a circuit and figure out the state diagram that it implements

Convert…

Circuit **to** **State Diagram**





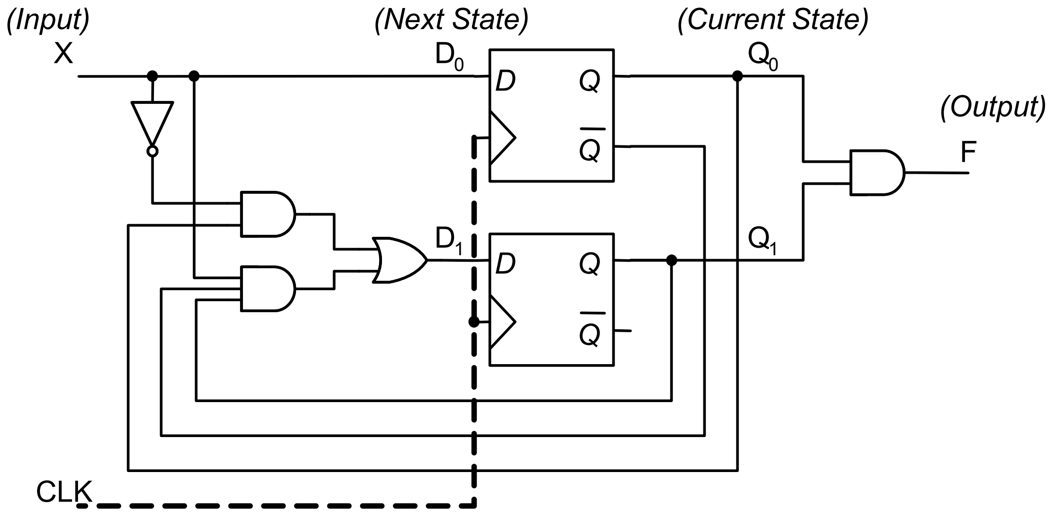
State Machine Analysis

- 6 Steps to analyze
  - Excitation Equations
    - (eqn's for FF inputs)
  - Transition Equations
    - (use characteristic equation of FF and substitute excitation equations for the FF inputs)
  - Output Equations
  - Transition/Output Table
    - Make a table showing all combinations of current state and external inputs and then what each of the next state and output values will be for each of those combinations
  - State Name Assignment
    - (Symbolic names replace binary codes)
  - Draw the State Diagram

Excitation Equations

Write equations for all FF inputs

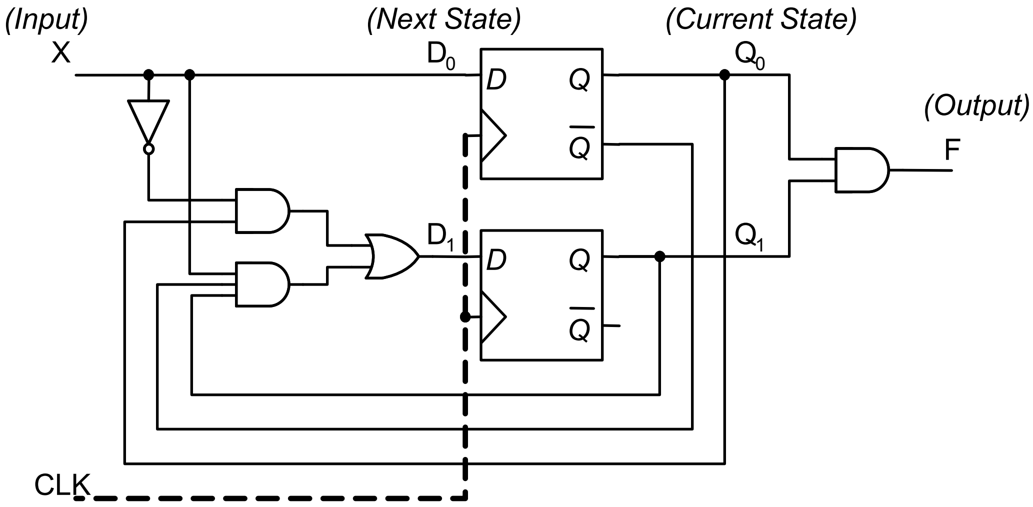
- $D_0 = X$
- $D_1 = X'Q_0 + XQ_1Q_0'$



Transition Equations

Come up with equations for the next value of the Q's (use characteristic equation Q* = D)

- $Q_0^* = D_0 = X$
- $Q_1^* = D_1 = X'Q_0 + XQ_1Q_0'$

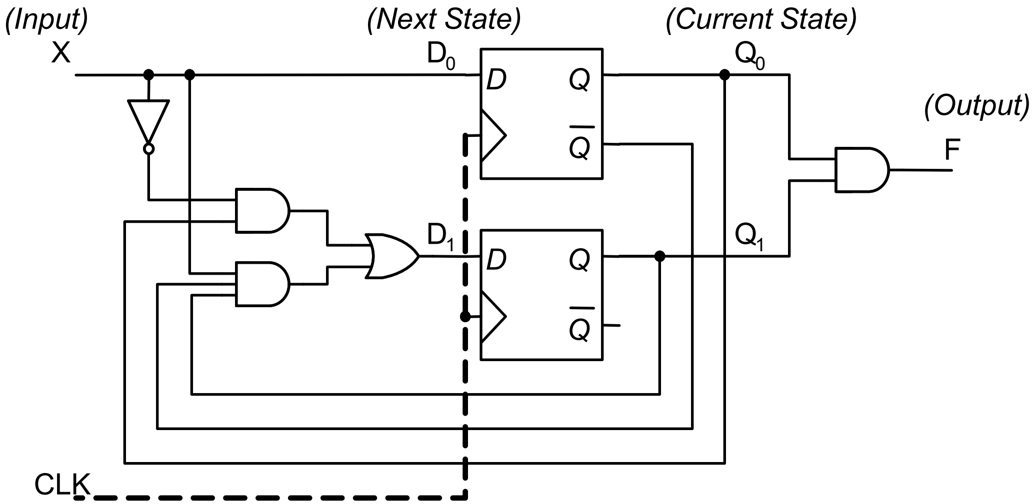


Output Equations

Equations for all outputs

- $F = Q_1Q_0$

Notice that this is a Moore Machine since the outputs depend on only the current state



Transition/Output Table

- Make a table of the current state, next state, and outputs
- Use the previous equations to fill out the transition output table

Current State		Next State				Outputs
		X = 0		X = 1		
Q1	Q0	Q1*	Q0*	Q1*	Q0*	F
0	0					
0	1					
1	0					
1	1					

$Q_0^* = X$

$Q_1^* = X'Q_0 + XQ_1Q_0'$

$F = Q_1Q_0$

Because this is a Moore Machine, we can make a separate column for F apart from the values of X

Table Format

Current State		Next State				Outputs
		X = 0		X = 1		
Q1	Q0	Q1*	Q0*	Q1*	Q0*	F
0	0					
0	1					
1	0					
1	1					

→

X	Q1	Q0	Q1*	Q0*
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Transition/Output Table

- Make a table of the current state, next state, and outputs
- Use the previous equations to fill out the transition output table

Current State		Next State				Outputs
		X = 0		X = 1		
Q1	Q0	Q1*	Q0*	Q1*	Q0*	F
0	0		0		1	
0	1		0		1	
1	0		0		1	
1	1		0		1	

$Q_0^* = X$

$Q_1^* = X'Q_0 + XQ_1Q_0'$

$F = Q_1Q_0$

Transition/Output Table

- Make a table of the current state, next state, and outputs
- Use the previous equations to fill out the transition output table

Current State		Next State				Outputs
		X = 0		X = 1		
Q1	Q0	Q1*	Q0*	Q1*	Q0*	F
0	0	0	0	0	1	
0	1	1	0	0	1	
1	0	0	0	1	1	
1	1	1	0	0	1	

$Q_0^* = X$

$Q_1^* = X'Q_0 + XQ_1Q_0'$

$F = Q_1Q_0$

Transition/Output Table

- Make a table of the current state, next state, and outputs
- Use the previous equations to fill out the transition output table

Current State		Next State				Outputs
		X = 0		X = 1		
Q1	Q0	Q1*	Q0*	Q1*	Q0*	F
0	0	0	0	0	1	0
0	1	1	0	0	1	0
1	0	0	0	1	1	0
1	1	1	0	0	1	1

$Q_0^* = X$

$Q_1^* = X'Q_0 + XQ_1Q_0'$

$F = Q_1Q_0$

Transition/Output Table

- Make a table of the current state, next state, and outputs
- Use the previous equations to fill out the transition output table

Current State		Next State				Outputs
		X = 0		X = 1		
Q1	Q0	Q1*	Q0*	Q1*	Q0*	F
0	0	0	0	0	1	0
0	1	1	0	0	1	0
1	0	0	0	1	1	0
1	1	1	0	0	1	1

If current state (Q_1Q_0) = 00 and X = 0 then we'll stay in state 00

If current state (Q_1Q_0) = 00 and X = 1 then we'll go to state 01

State Name Assignment

- Just give a symbolic name to each state (i.e. 00 = SA, 01 = SB, etc.)
- In this case use the following names…

Current State			Next State						Output
			X = 0			X = 1			
Q_1	Q_0	State	Q_1^*	Q_0^*	State	Q_1^*	Q_0^*	State	F
0	0	SInit	0	0	SInit	0	1	S1	0
0	1	S1	1	0	S10	0	1	S1	0
1	0	S10	0	0	Sinit	1	1	S101	0
1	1	S101	1	0	S10	0	1	S1	1

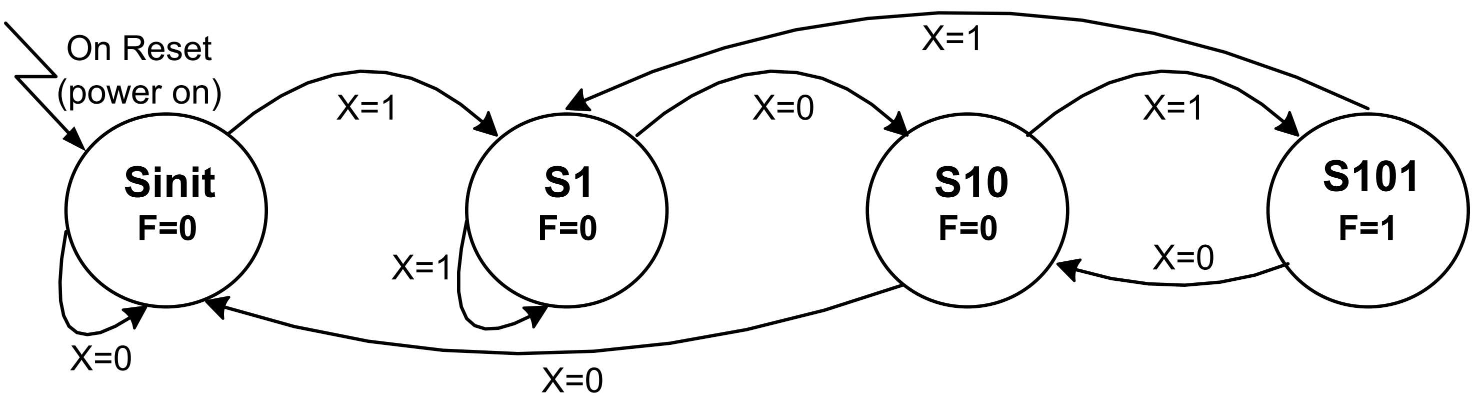
We call state 01 = S1

Replace 01 with S1

State Diagram

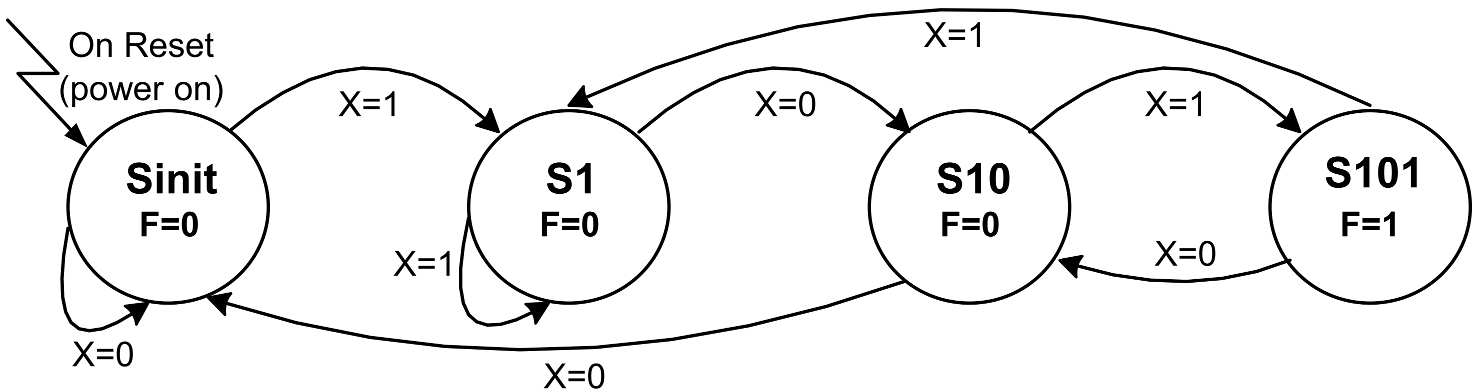
- Translate table to State Diagram

Current State			Next State						Output
			X = 0			X = 1			
Q_1	Q_0	State	Q_1^*	Q_0^*	State	Q_1^*	Q_0^*	State	F
0	0	SInit	0	0	SInit	0	1	S1	0
0	1	S1	1	0	S10	0	1	S1	0
1	0	S10	0	0	Sinit	1	1	S101	0
1	1	S101	1	0	S10	0	1	S1	1



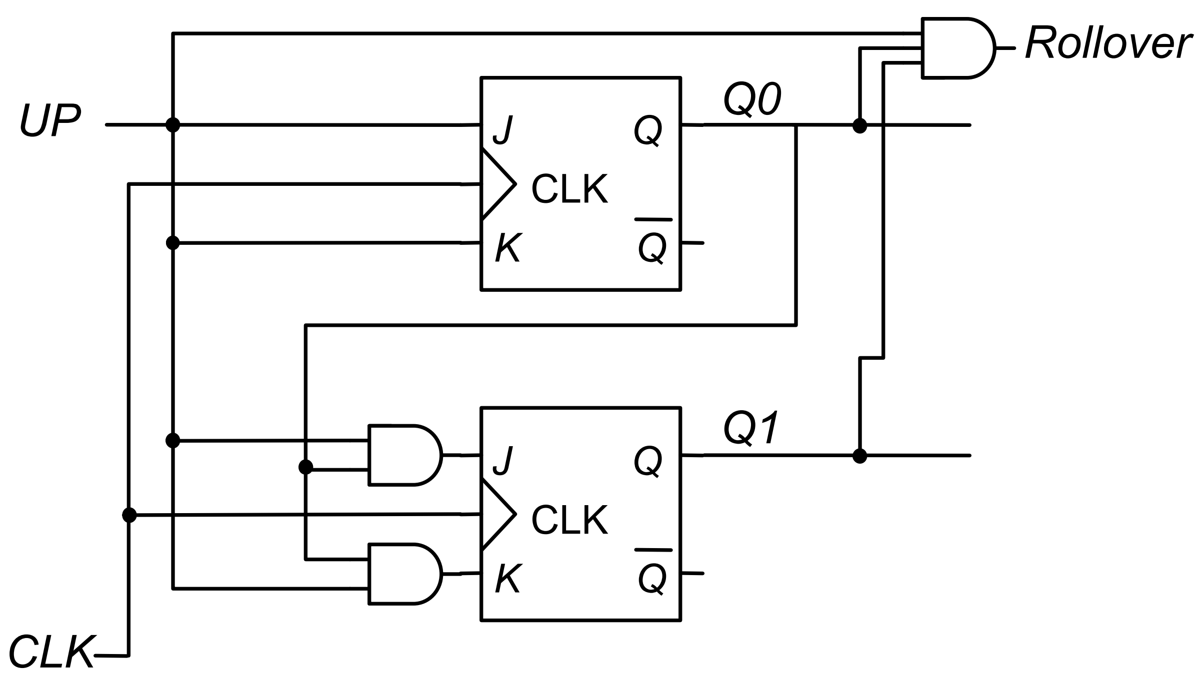
State Diagram

- This state diagram implements the "101" sequence detector



State Machine Analysis

- Consider the following circuit
- We now use JK FF's and a Mealy output



State Machine Analysis

Excitation Equations

- $J0 = K0 = Up$
- $J1 = K1 = Up \cdot Q_0$

Transition Equations

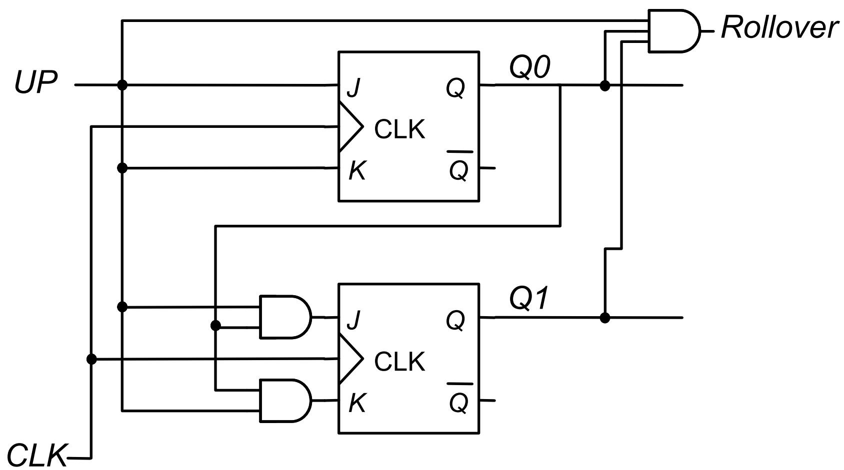
(char. eqn for JK FF: Q* = JQ' + K'Q)

- $Q_0^* = Up \cdot Q_0' + Up' \cdot Q_0$
- $Q_1^* = Up \cdot Q_0 \cdot Q_1' + \overline{(Up \cdot Q_0)} \cdot Q_1$

  $= Up \cdot Q_1' \cdot Q_0 + Up' \cdot Q_1 + \cdot Q_1 Q_0'$

Output Equations

- $R = Up \cdot Q_1 \cdot Q_0$ *(Mealy output)*



Transition Output Table

Current State			Next State							
			Up = 0				Up = 1			
Q1	Q0	State	Q1*	Q0*	State	R	Q1*	Q0*	State	R
0	0	S0	0	0	S0	0	0	1	S1	0
0	1	S1	0	1	S1	0	1	0	S2	0
1	0	S2	1	0	S2	0	1	1	S3	0
1	1	S3	1	1	S3	0	0	0	S0	1

Mealy outputs must be shown for each sub column of the inputs

Q0* = Up•Q0’ + Up’•Q0

Q1* = Up•Q1’•Q0 + Up’•Q1 + •Q1Q0’

R = Up•Q1•Q0

Transition Output Table

Current State			Next State							
			Up = 0				Up = 1			
Q1	Q0	State	Q1*	Q0*	State	R	Q1*	Q0*	State	R
0	0	S0	0	0	S0	0	0	1	S1	0
0	1	S1	0	1	S1	0	1	0	S2	0
1	0	S2	1	0	S2	0	1	1	S3	0
1	1	S3	1	1	S3	0	0	0	S0	1

Notice that…
- When Up=1, we count up
- When Up=0 we stay in the current state
- Rollover = 1 only when we're about to rollover from S3 to S0