Course Name: Digital Hardware Design
Course Code: 17B1NEC741

Finite State Machine-2

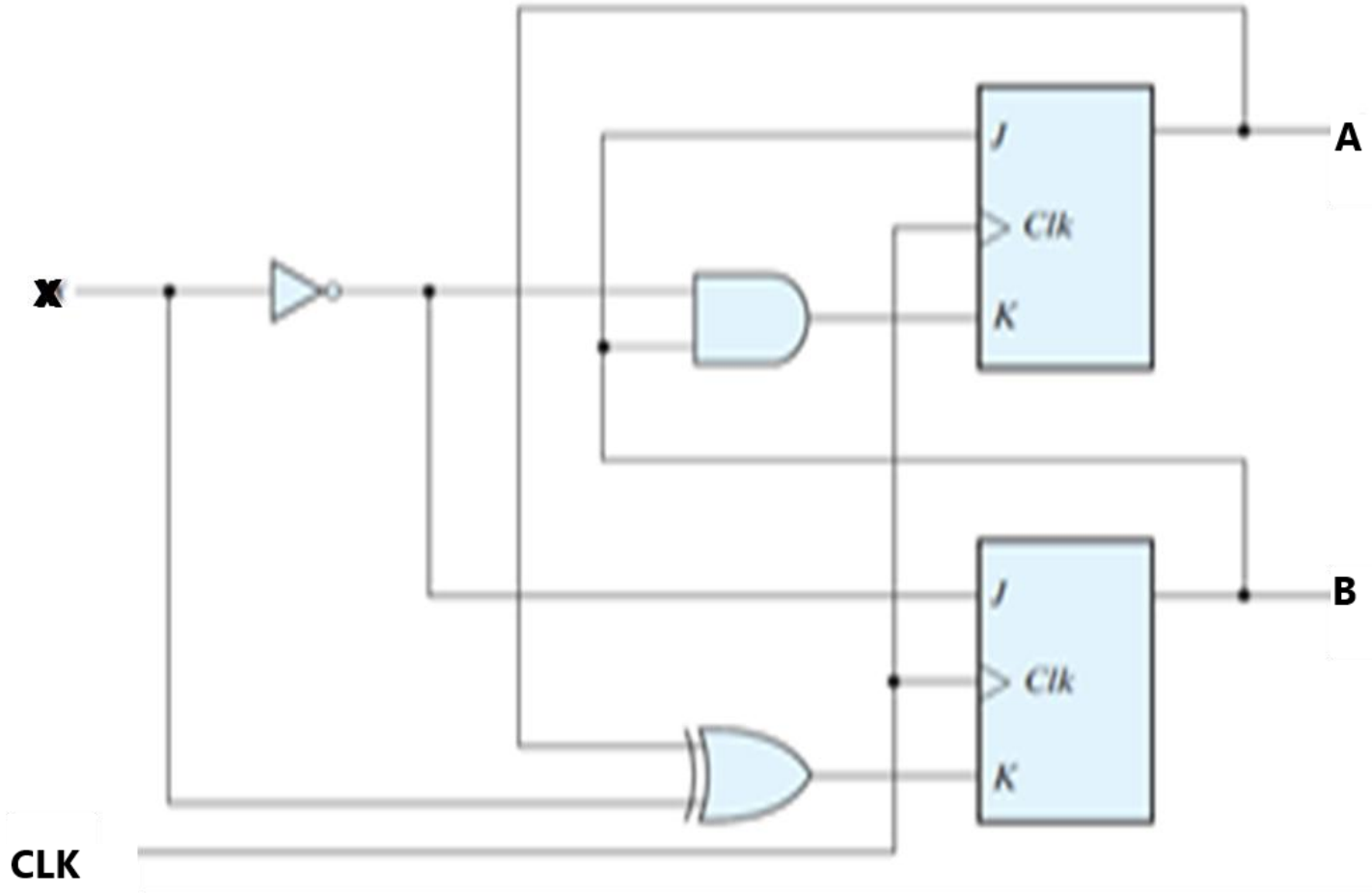
Dr. Arti Noor
Dean, Academic Affairs
Electronics and Communication Engineering,
Jaypee Institute of Information Technology, Noida

Example 3

- Use the characteristic table or characteristic equation of the JK FF, i.e., $Q(t+1)=JQ' + K'Q$ to find the next state.
- Derive the state transition table.
- In this case there is no output.

$J_A=B,\ K_A=X'B$

$J_B=X',\ K_B=X \oplus A$



Example 3

Not part of state table

Current state		Input	FF Inputs				Next state	
$A(t)$	$B(t)$	$x(t)$	J_A	K_A	J_B	K_B	$A(t+1)$	$B(t+1)$
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1

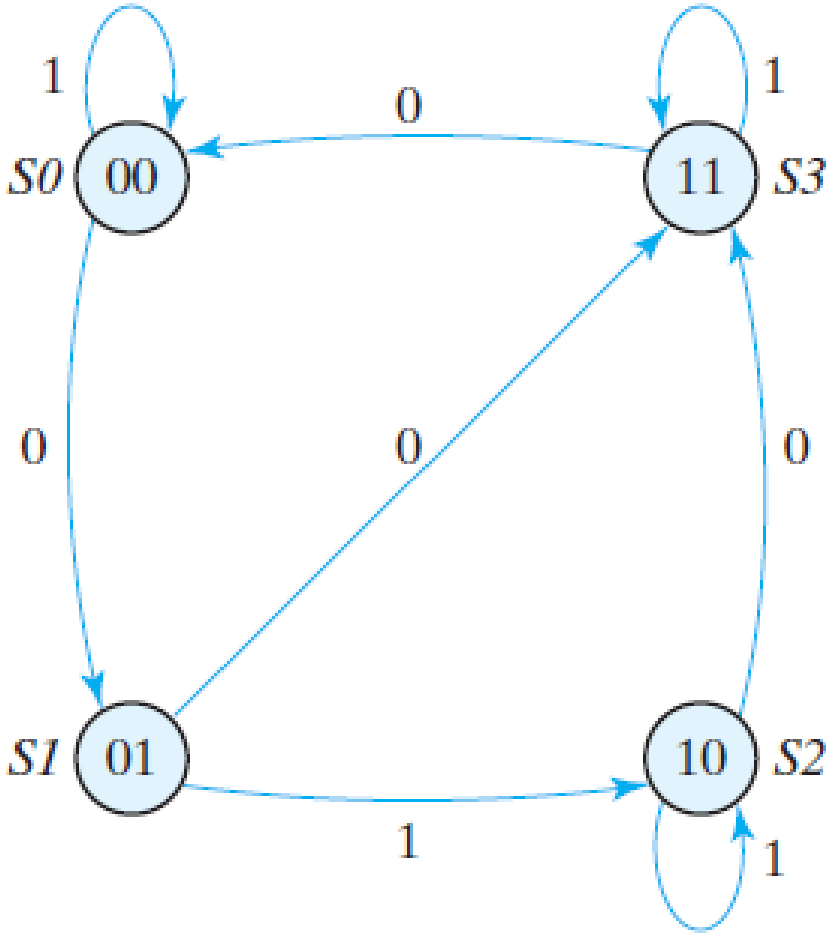
Present State	Next State	Inputs	
Q	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$A(t+1)=J_A\ A' + K'_A\ A$

$B(t+1)=J_B\ B' + K'_B\ B$

Example 3

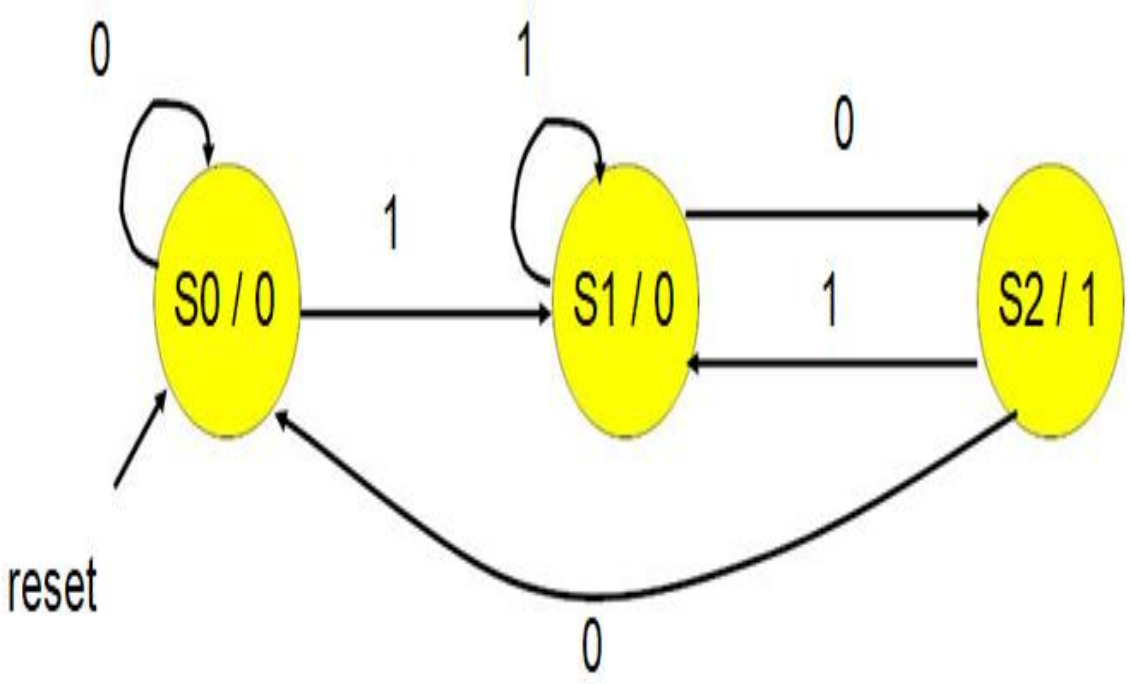
Present State		Input	Next State	
A	B	x	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1



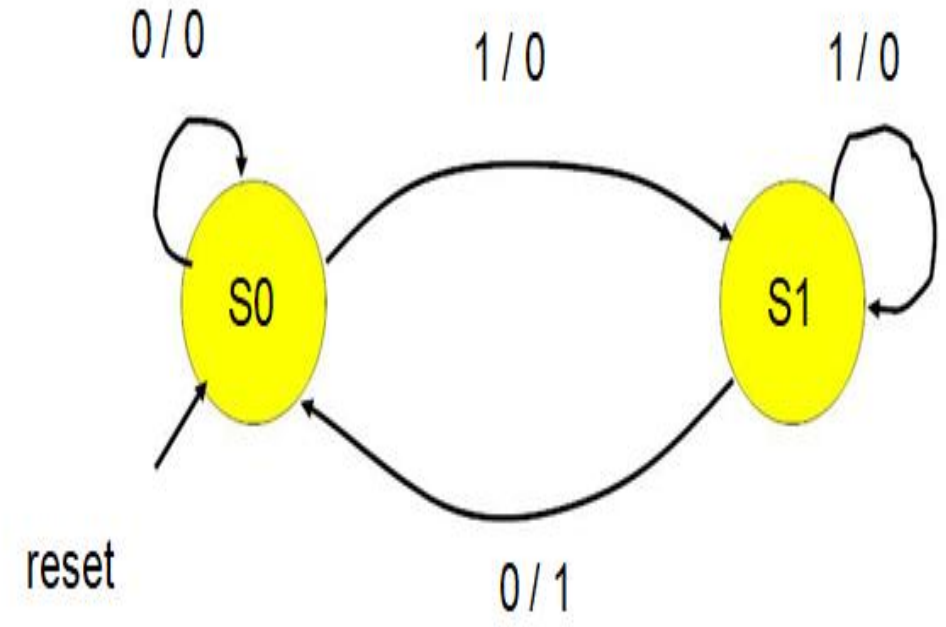
State diagram

Difference between Mealy and Moore

Example: State Machine ’10’ Sequence Detector



Moore Machine



Mealy Machine

State Machine Sequence Detector

Sequence detector is of two types:
1. Overlapping
2. Non-Overlapping

Overlapping: **In this type, the last bit of one sequence becomes the first bit of the next.**

Non-Overlapping: **In this type, the last bit of one sequence does not become the first bit of the next sequence.**

Example: 101 Mealy sequence detector

For non-overlapping case
Input :0110101011001
Output:0000100010000

For overlapping case
Input :0110101011001
Output:0000101010000

FSM Design (Moore)

Problem Statement:

Design a FSM that detects a sequence of three or more consecutive ones on an input bit stream.

The FSM should output a 1 when the sequence is detected, and a 0 otherwise.

Input: 0 1 1 1 0 1 0 1 1 0 1 1 1 0 1 …

Output: 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0

…

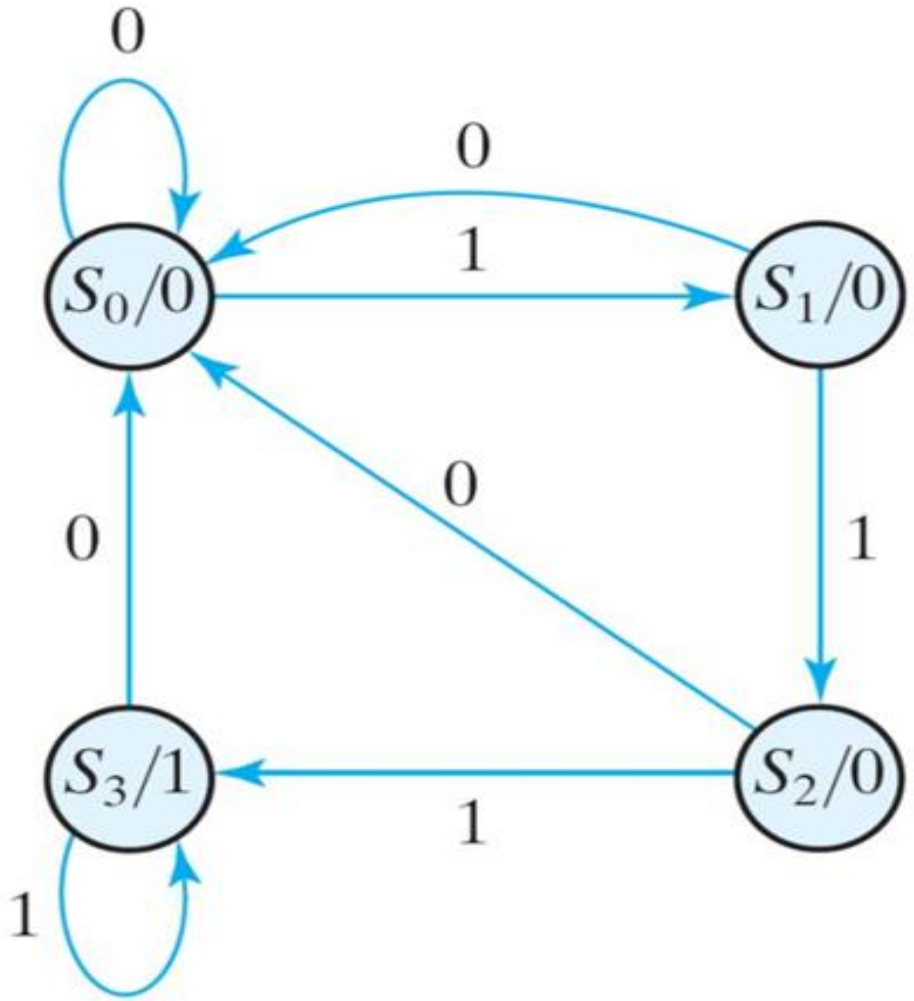
Non-Overlapping Case

FSM Design (Moore)

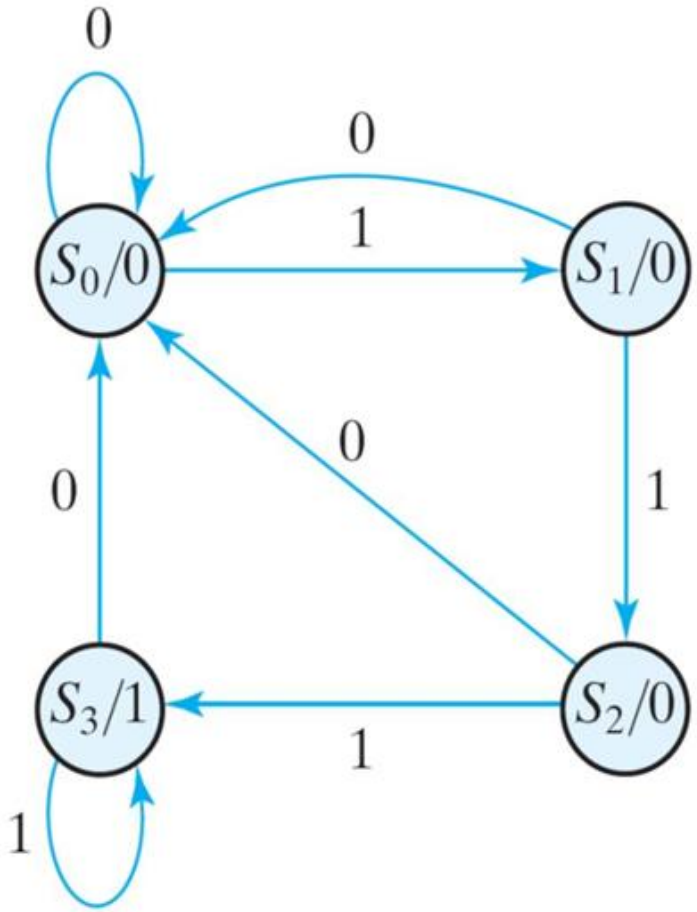
Input: 0 1 1 1 0 1 0 1 1 0 1 1 1 0 1 …

Output: 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0

…



FSM Design (Moore)



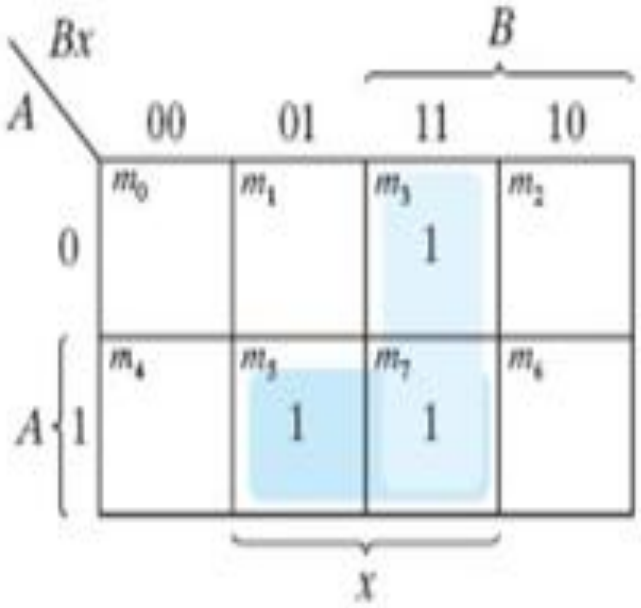
(S0=00, S1=01, S2=10, S3=11)

Present State		Input	Next State		Output
Q_A	Q_B	x	Q_A^+	Q_B^+	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

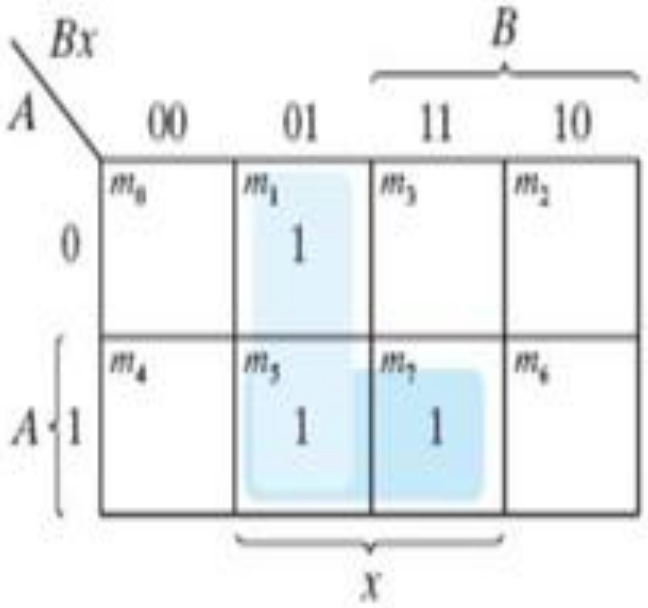
FSM Design (Moore)
Synthesis using D Flip Flop

Present State		Input	Next State		Flip-Flop Input	
Q_A	Q_B	x	Q_A^+	Q_B^+	D_A	D_B
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	0	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	1	1	1	1	1

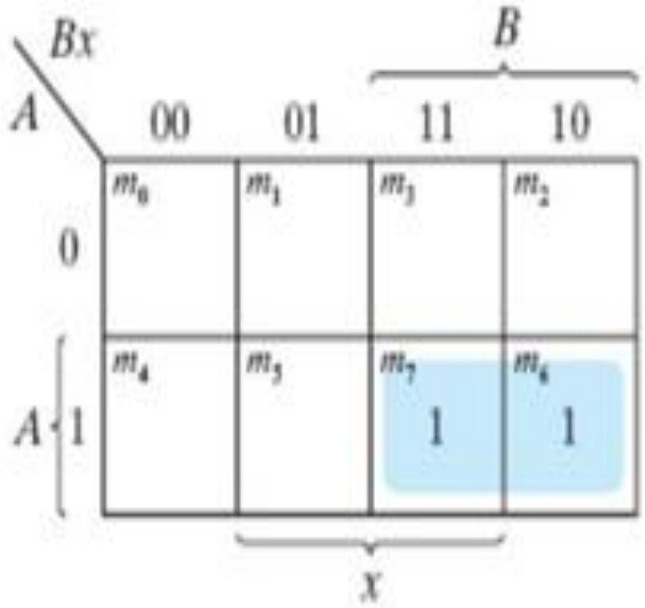
FSM Design (Moore)
Synthesis using D Flip Flop



$D_A = Ax + Bx$



$D_B = Ax + B'x$



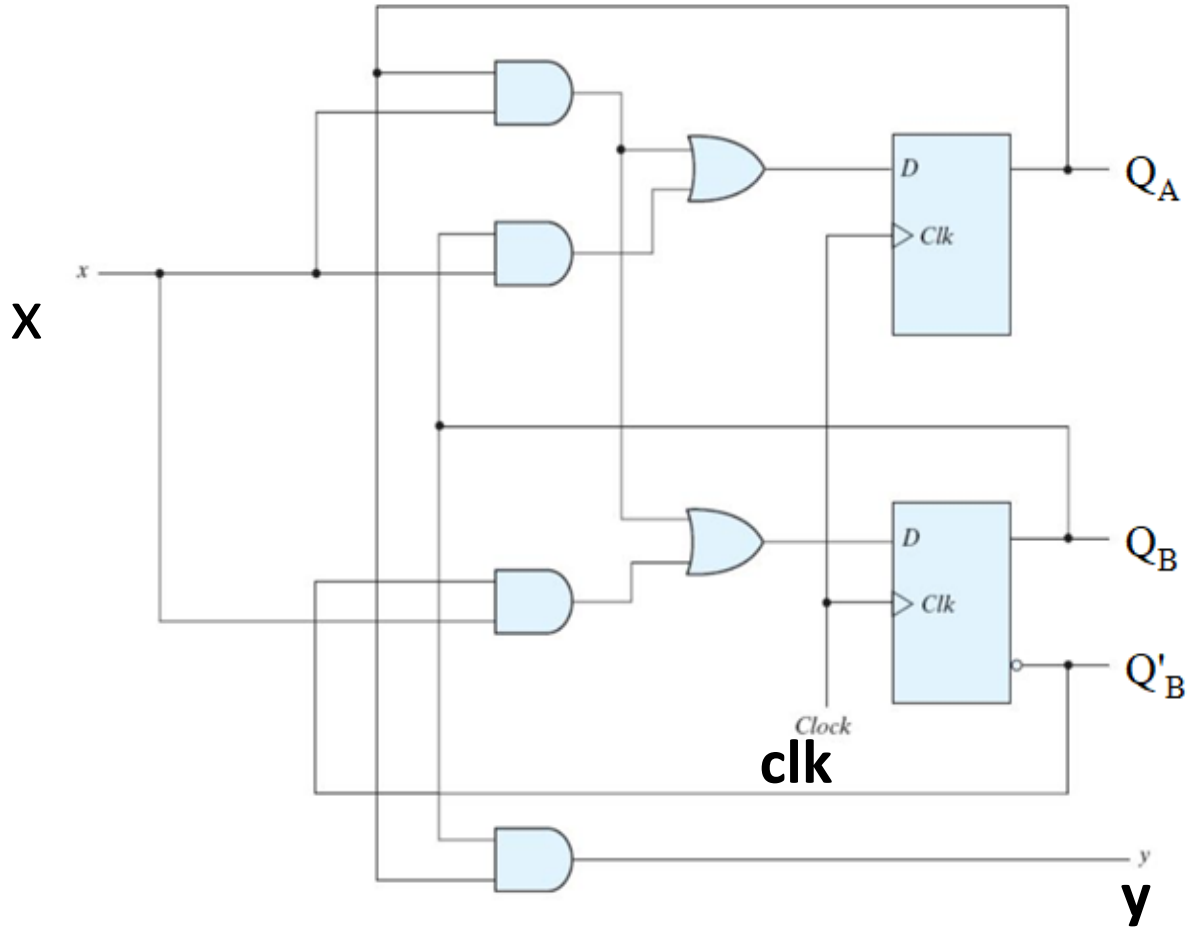
$y = AB$

FSM Design (Moore)
Synthesis using D Flip Flop

$D_A=Ax+Bx \quad D_B=Ax+B'x$

$y=AB$

$A=Q_A$, $B=Q_B$

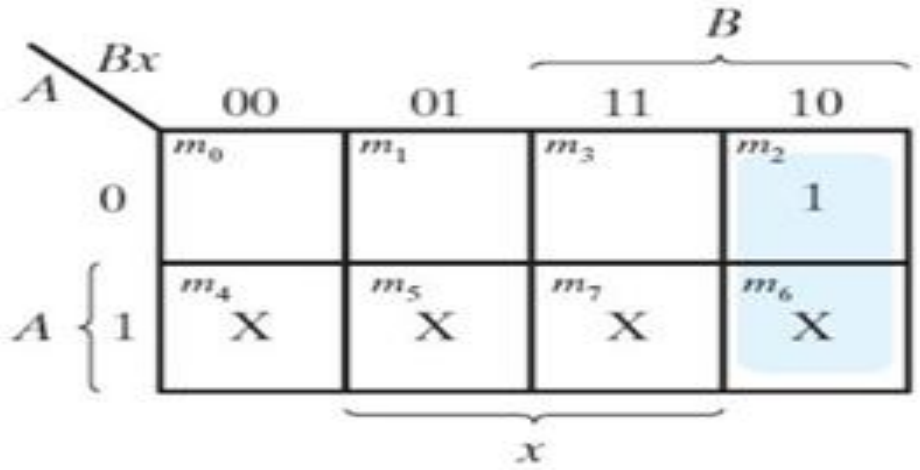


FSM Design (Moore)
Synthesis using JK Flip Flop

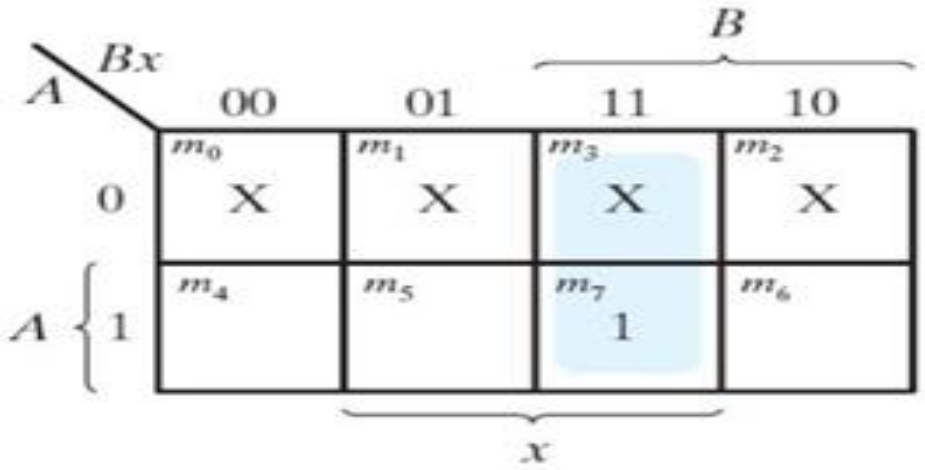
Present State	Next State	Inputs	
Q	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present State		Input	Next State		Flip-Flop Inputs			
Q_A	Q_B	x	Q_A^+	Q_B^+	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

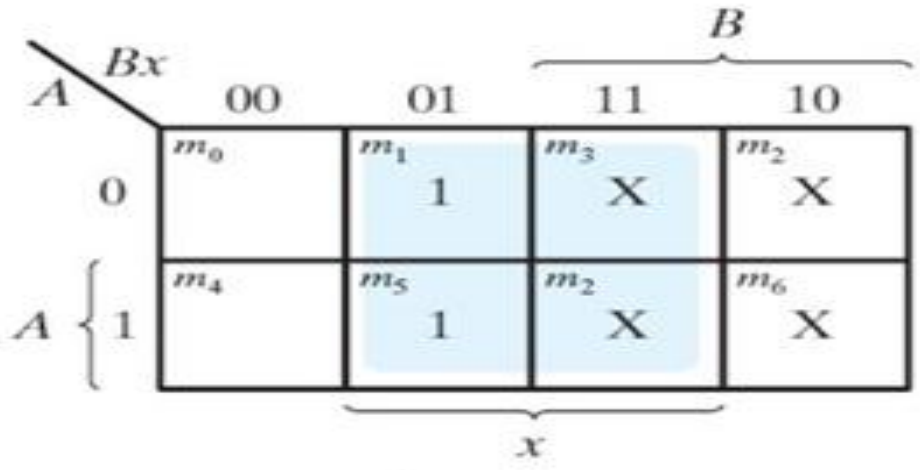
FSM Design (Moore)
Synthesis using JK Flip Flop



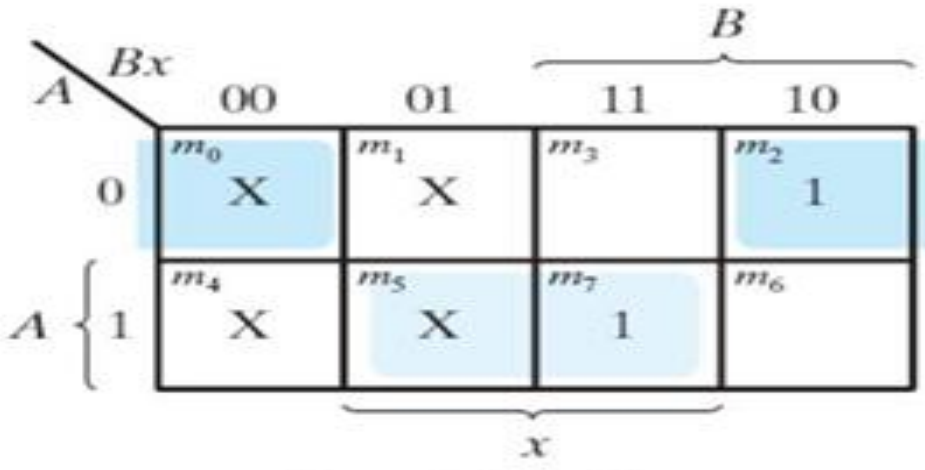
$$J_A = Bx'$$



$$K_A = Bx$$



$$J_B = x$$



$$K_B = (A \oplus x)'$$

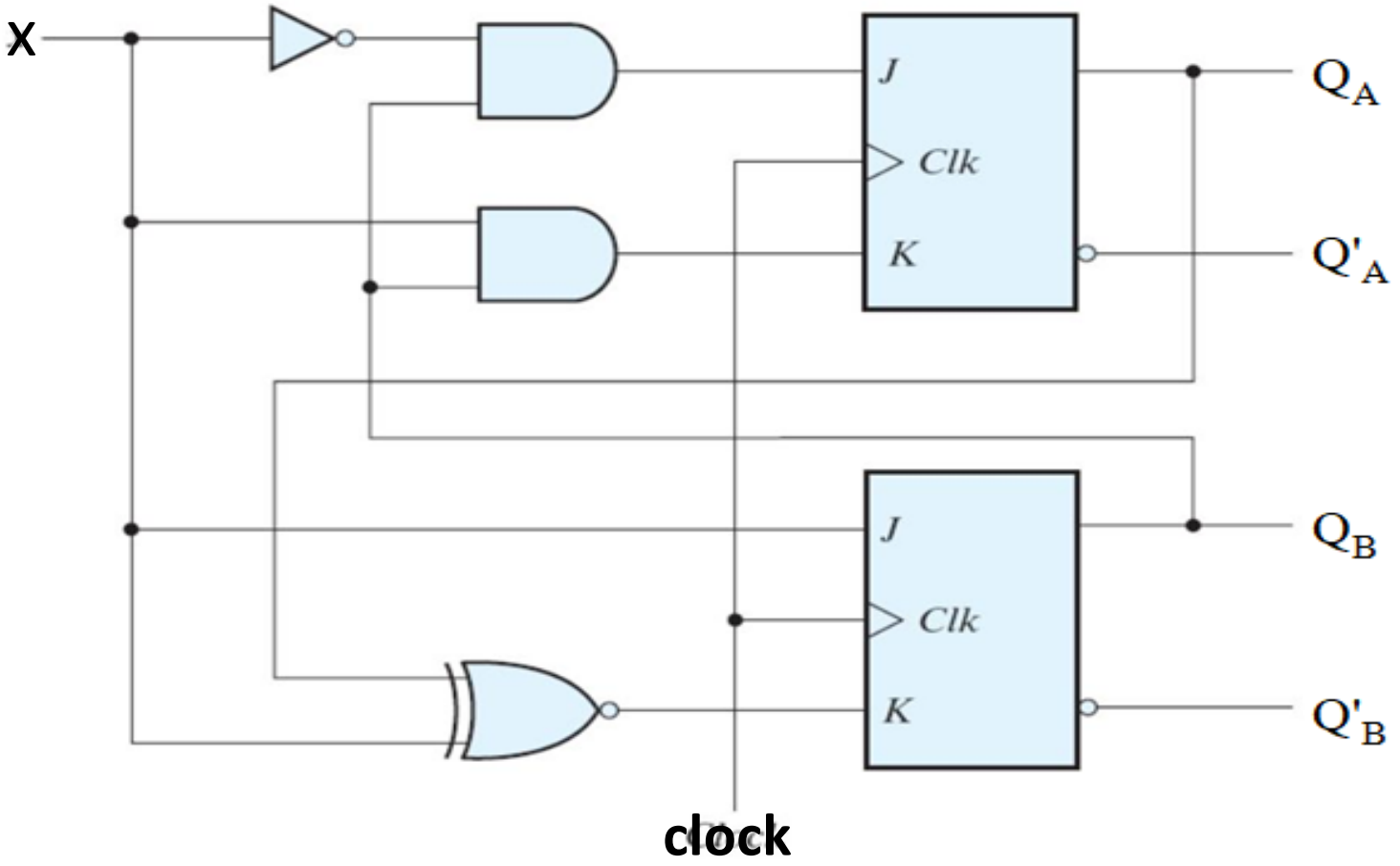
FSM Design (Moore)
Synthesis using JK Flip Flop

$J_A = Bx'$

$K_A = Bx$

$J_B = x$

$K_B = (A \text{ XOR } x)'$



FSM Design Example 3 (Moore)

Design a Finite State Machine (FSM) that meets the following specifications:

1. The circuit has one input, w, and one output, z.

2. All changes in the circuit occur on the positive edge of the clock.

3. The output z is equal to 1 if the pattern 101 is detected on the input w. Otherwise, the value of z is equal to 0. Overlapping sequences **should** be detected.

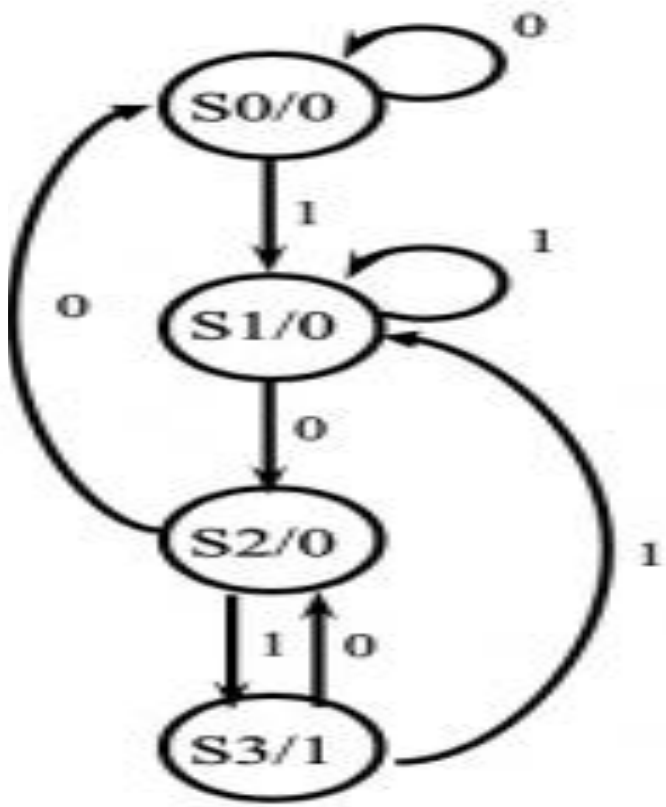
Input (w): 0 0 0 1 0 1 0 1 1 0 1 1 0 1 1 ...

Output (z): 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 ...

FSM Design Example 3 (Moore)

Input (w): 0 0 0 1 0 1 0 1 1 0 1 1 0 1 1 ...

Output (z): 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 ...



FSM Design Example 4 (Moore)

Design a Finite State Machine (FSM) that meets the following specifications:

1. The circuit has one input, w, and one output, z.

2. All changes in the circuit occur on the positive edge of the clock.

3. The output z is equal to 1 if the pattern 110 or the pattern 010 is detected on the input w. Otherwise, the value of z is equal to 0. Overlapping sequences **should** be detected.

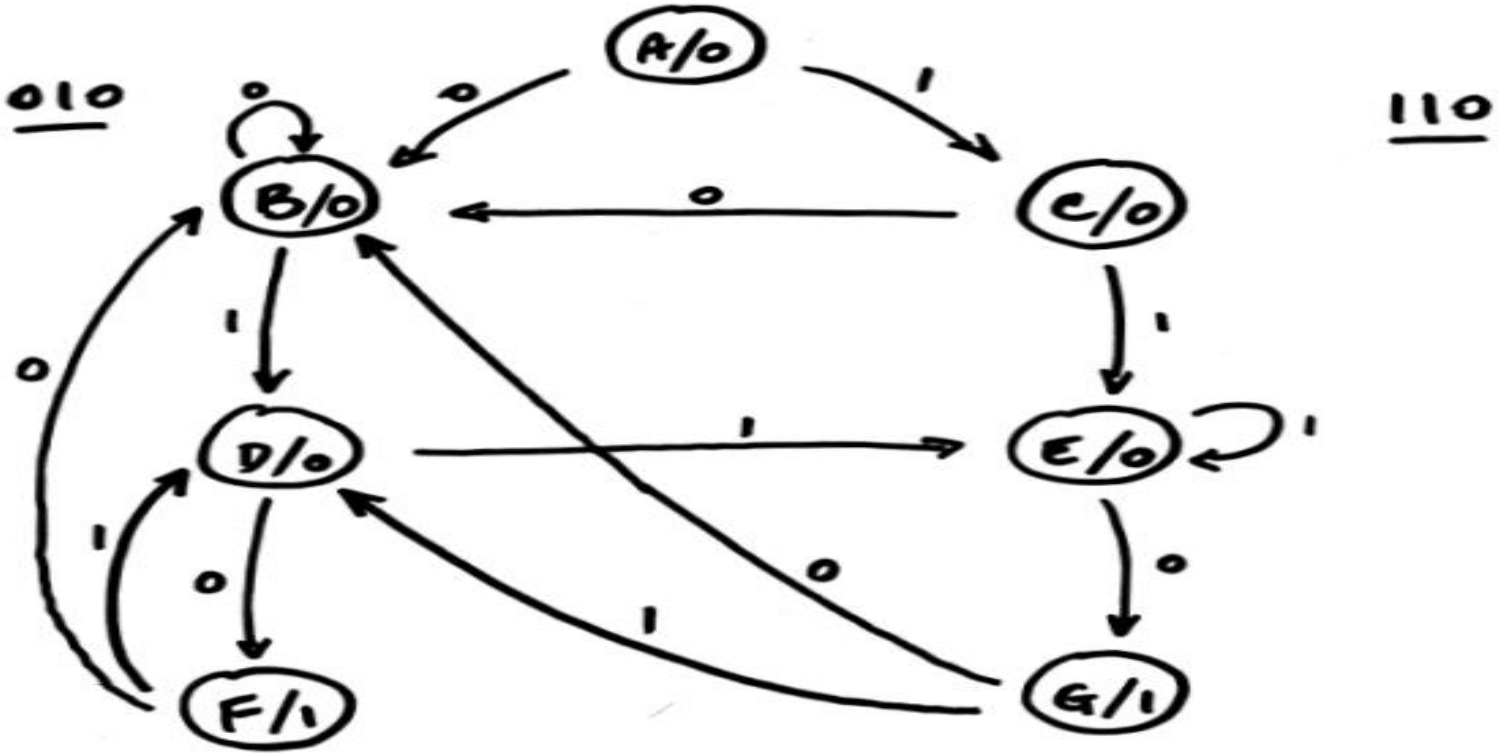
Input (w): 0 1 0 0 1 1 0 1 0 1 1 1 0 1 1 ...

Output (z): 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 ...

FSM Design Example 4 (Moore)

Input (w): 0 1 0 0 1 1 0 1 0 1 1 1 0 1 1 …

Output (z): 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 …



State Diagram