CSE 201: Digital Logic Design

Analysis of Sequential Circuit

State Reduction and Assignment

Mealy and Moore Model

Design of Sequential Circuit

Design of Counters

Prepared By

Lec Sumaiya Afroz Mila

CSE, MIST

Analysis of Sequential Circuit

Analysis of Clocked Sequential Circuits: Terminology

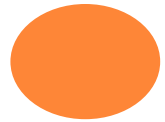
- ***STATE EQUATION*** : A state equation is an algebraic expression that specifies the conditions for a flip-flop state transition.

  Left side of the equation denotes the next state and right side is a boolean function that specifies the present state conditions that make the next state equal to 1.

- ***STATE TABLE*** : A state table consists of 3 sections – Present state, Next state and output(if there is any). [We will need the characteristics table to form state table]

- ***STATE DIAGRAM***: The information in a state table can be represented graphically in a state diagram.

  The state is represented by a circle and the transitions between states are indicated by directed lines connecting the circles.



Characteristics Tables

Flip-Flop Characteristic Tables

JK Flip-Flop

J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

RS Flip-Flop

S	R	$Q(t + 1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Unpredictable

D Flip-Flop

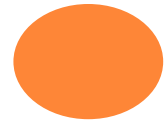
D	Q(t + 1)	
0	0	Reset
1	1	Set

T Flip-Flop

T	Q(t + 1)	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

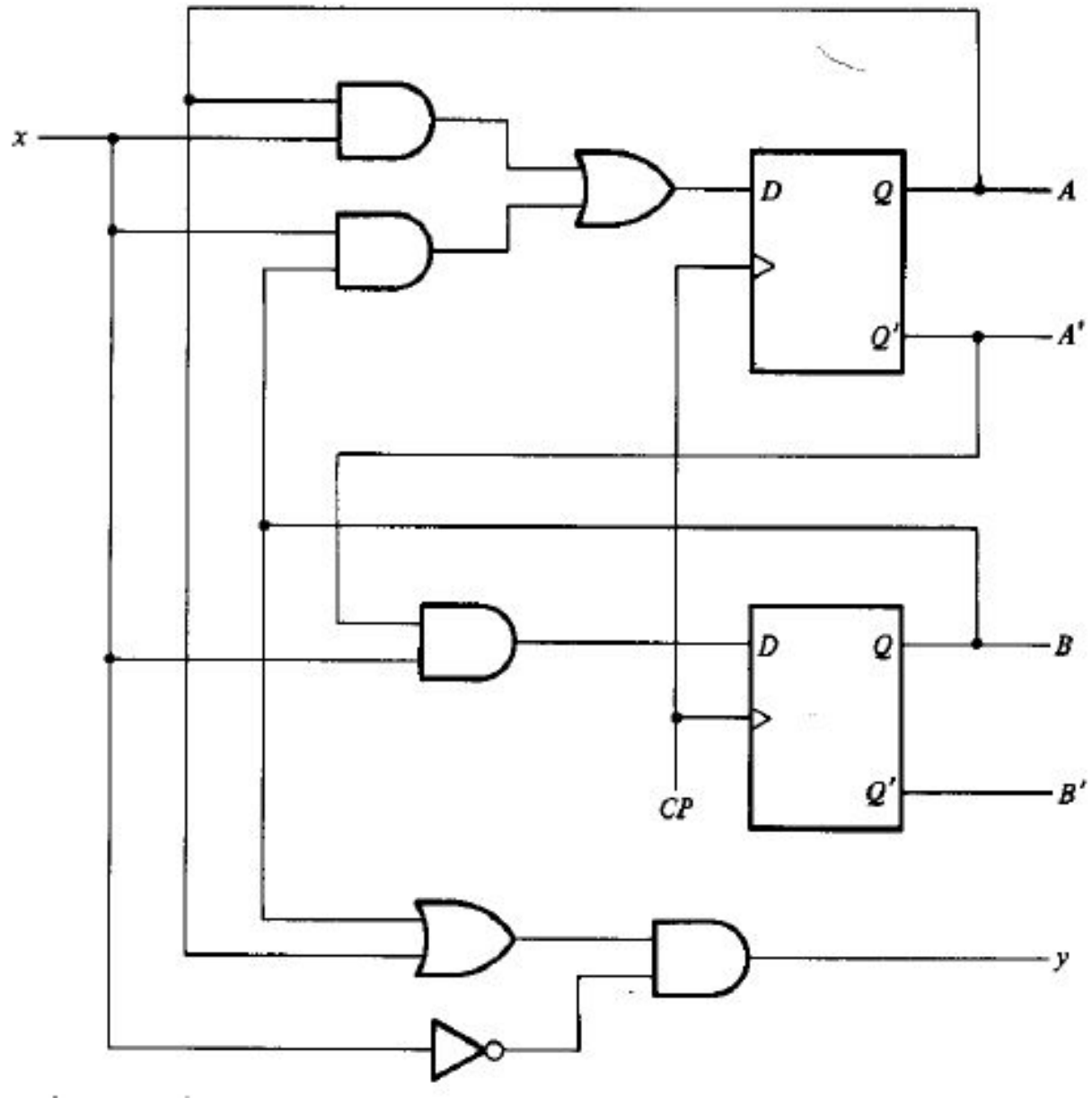
Steps for Analyzing Clocked Sequential Circuits

1. At first, find out the flip-flop specified in the question
2. Derive the input equation of the circuit
3. Generate the state table with the help of input equation and characteristics table of that flip-flop
4. Derive the state equation from the state table
5. Draw the state diagram from the state table



Analysis of Clocked Sequential Circuits: Example 1

You are given a circuit. Analyse it.



Analysis of Clocked Sequential Circuits: Example 1

Input Equation:

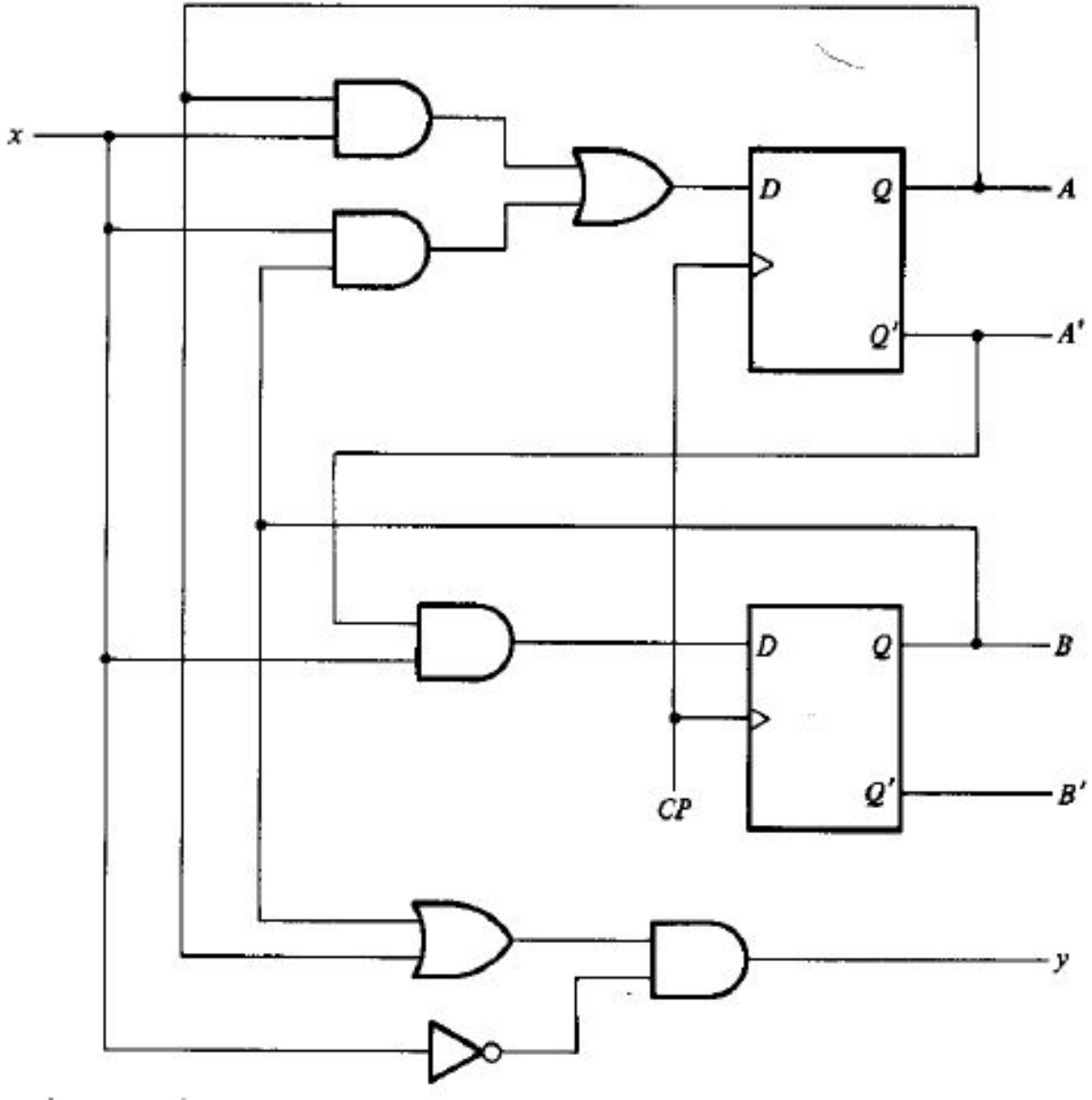
$D_A = A(t)x(t) + B(t)x(t)$

$D_B = A'(t)x(t)$

$Y = [A+B]\,x'$

State Table

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0



Analysis of Clocked Sequential Circuits: Example 1 Cntd.

Input Equation:

$D_A = A(t)x(t) + B(t)x(t)$

$D_B = A'(t)x(t)$

$Y = [A+B]\,x'$

State Table

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0

State Equation

A(t+1):

A \ BX	00	01	11	10
0	0	0	1	0
1	0	1	1	0

A(t+1): Bx + Ax

B(t+1):

A \ BX	00	01	11	10
0	0	1	1	0
1	0	0	0	0

B(t+1): A'x

Analysis of Clocked Sequential Circuits: Example 1 Cntd.

Input Equation:

$D_A = A(t)x(t) + B(t)x(t)$

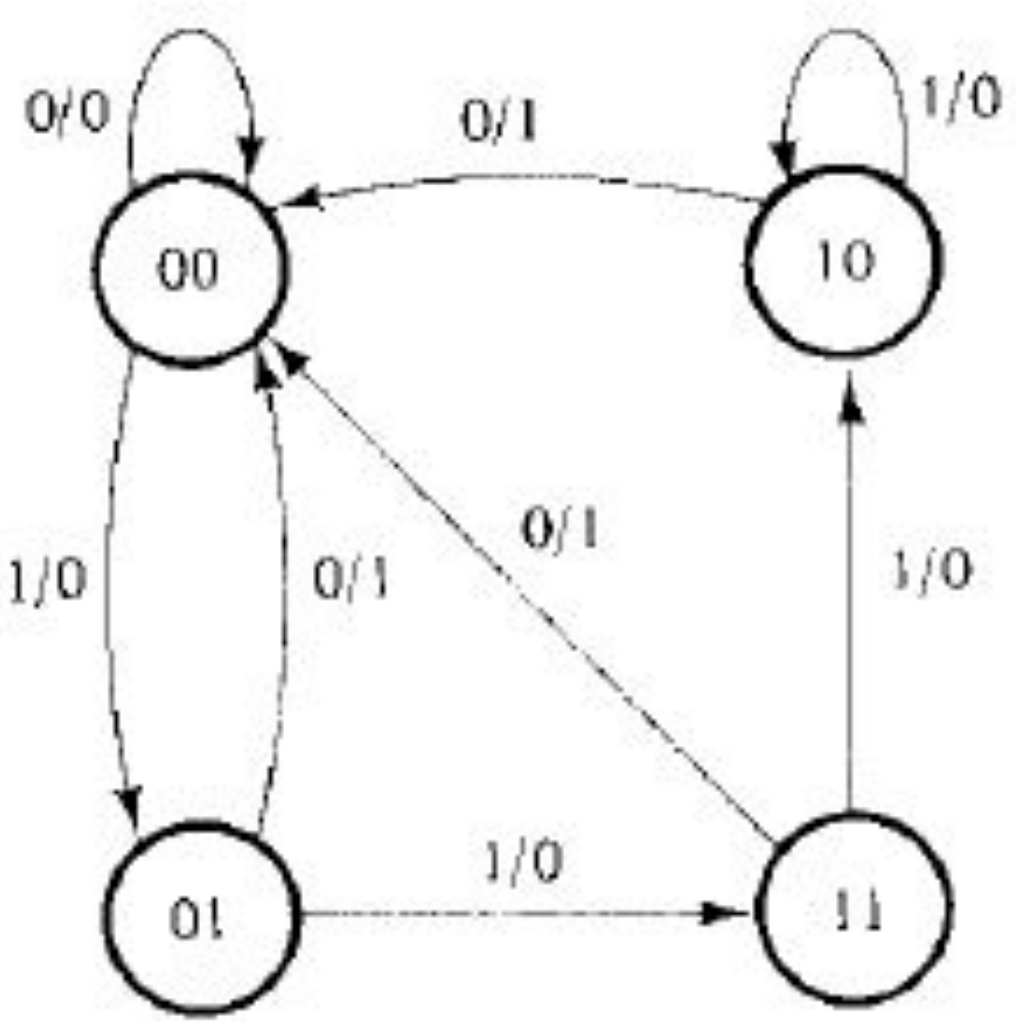
$D_B = A'(t)x(t)$

$Y = [A+B]\,x'$

State Table

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0

State Diagram



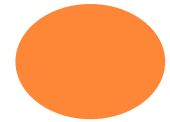
Analysis of Clocked Sequential Circuits: Example 2

Input Equation:

$$JA = B \qquad JB = x'$$

$$KA = Bx' \qquad KB = A'x + Ax' = A \oplus x$$

- You are given input equations of a circuit. Analyse it.

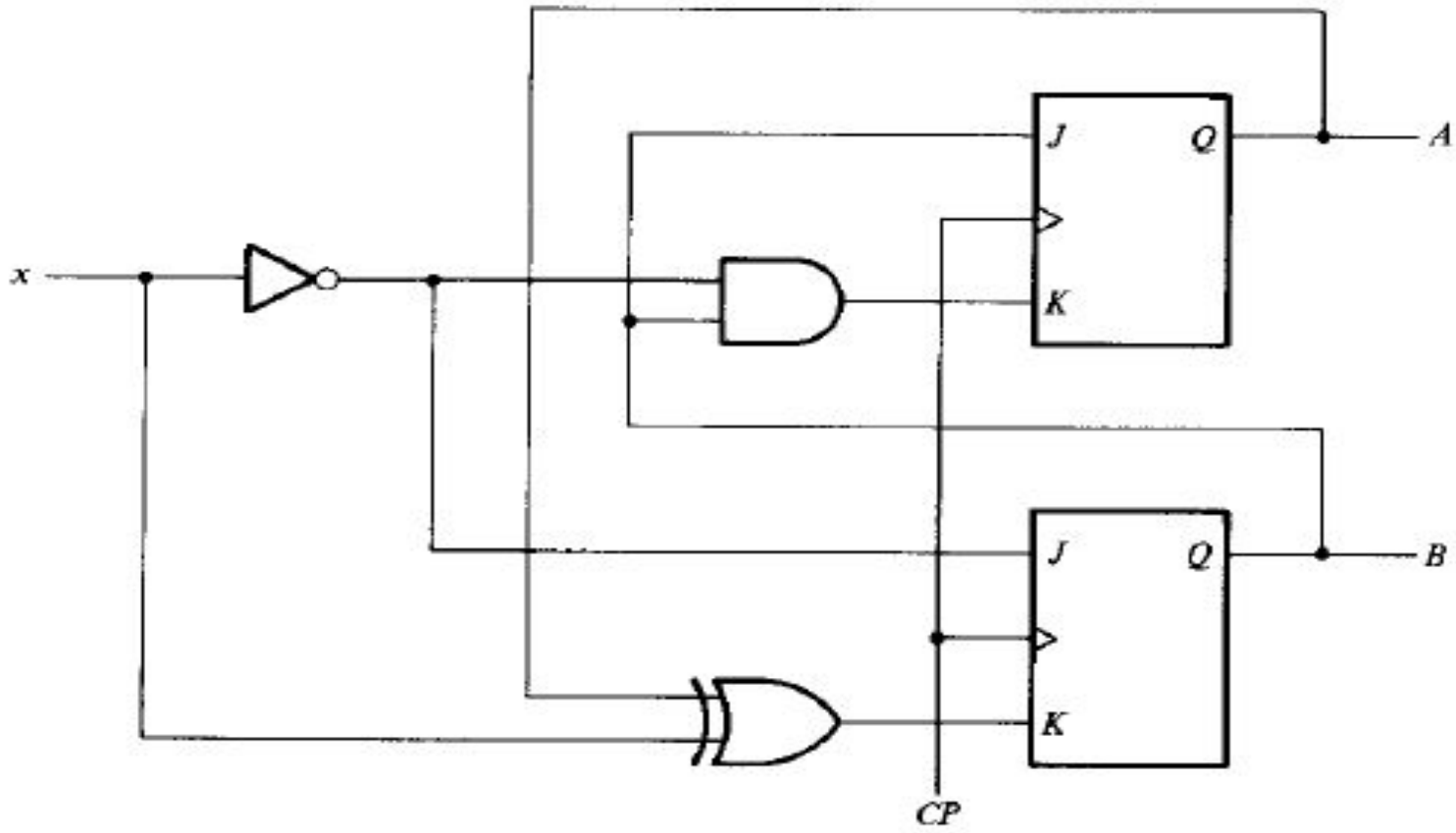


Analysis of Clocked Sequential Circuits: Example 2 Cntd.

Input Equation:

$JA = B \qquad JB = x'$

$KA = Bx' \qquad KB = A'x + Ax' = A \oplus x$



Analysis of Clocked Sequential Circuits: Example 2 Cntd.

Input Equation:

$$JA = B \qquad JB = x'$$

$$KA = Bx' \qquad KB = A'x + Ax' = A \oplus x$$

State Table

Present State		Input	Next State		Flip Flop Inputs			
A	B	x	A_{t+1}	B_{t+1}	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

A(t+1):

A \ BX	00	01	11	10
0	0	0	1	1
1	1	1	1	0

A(t+1): A'B + Ax + AB'

B(t+1):

A \ BX	00	01	11	10
0	1	0	0	1
1	1	0	1	0

B(t+1): B'x' + A'x' + ABx

Analysis of Clocked Sequential Circuits: Example 2 Cntd.

Input Equation:

$JA = B$ $\quad JB = x'$

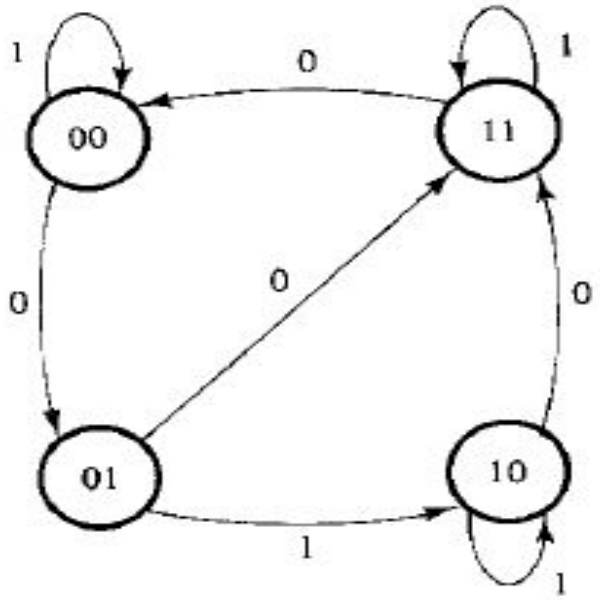
$KA = Bx'$ $\quad KB = A'x + Ax' = A \oplus x$

State Table

Present State		Input	Next State		Flip Flop Inputs			
A	B	x	A_{t+1}	B_{t+1}	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

JK Flip-Flop

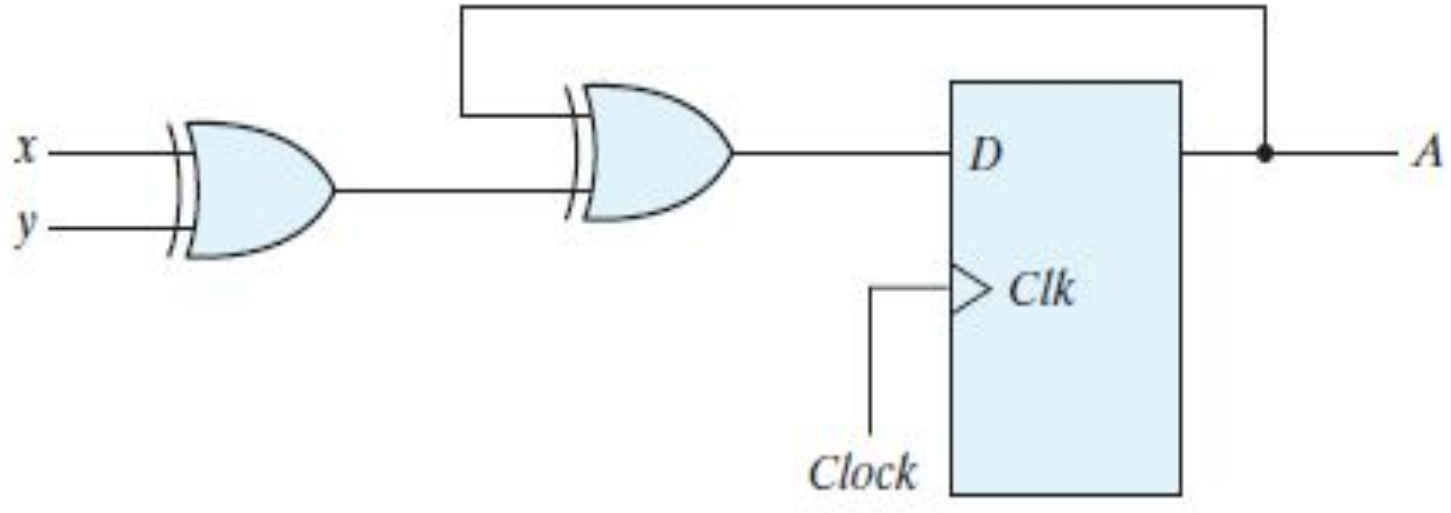
J	K	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement



Analysis of Clocked Sequential Circuits: Example 3

Input Equation:

$$D_A = A \oplus x \oplus y$$



State Table

Present State	Input		Next State	Flip Flop Inputs
A	x	y	A_{t+1}	D_A
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

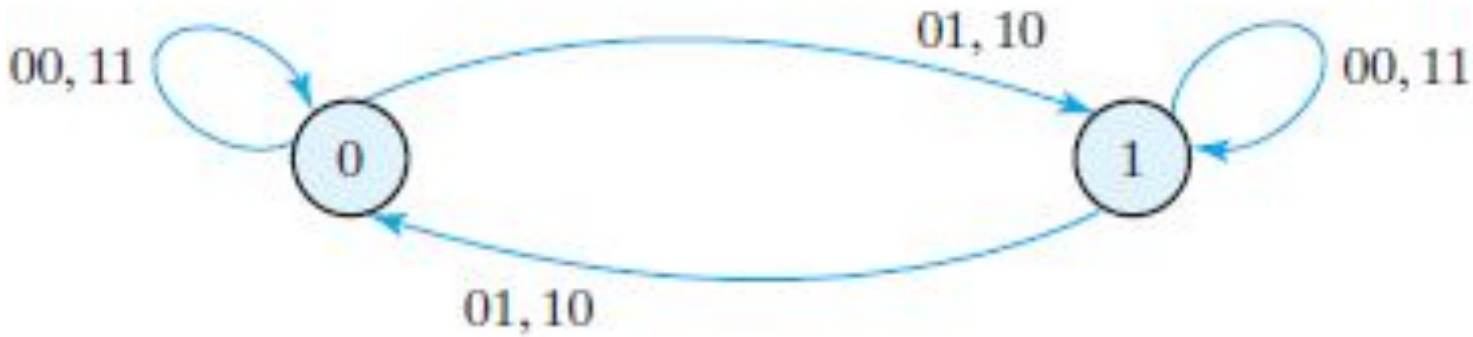
Analysis of Clocked Sequential Circuits: Example 3 Cntd.

Input Equation:

$$D_A = A \oplus x \oplus y$$

State Table

Present State	Input		Next State	Flip Flop Inputs
A	x	y	A_{t+1}	D_A
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



A \ XY	00	01	11	10
0	0	1	0	1
1	1	0	1	0

A(t+1): A'x'y + A'xy'+Ax'y'+ Axy

= A' (x'y+xy') + A (x'y'+xy)

= A' (x xor y) + A (x xnor y)

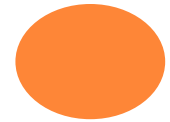
= A xor x xor y

Analysis of Clocked Sequential Circuits: Example 4(self study)

Input Equation:

$$T_A = Bx$$
$$T_B = x$$
$$y = AB$$

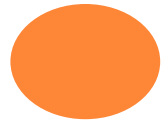
Try analyzing a circuit containing the above input equation



State Reduction & Assignment

State Reduction

- State reduction is required to minimize the cost of the final circuit
- State reduction algorithms are concerned with procedures for reducing the number of states in a state table without affecting the input-output sequence
- **Two circuits are said to be equivalent:**
  - **If identical outputs occur for all input sequences**
  - **Number of states is not important**
- **An algorithm for state reduction:**
  - Two states are said to be equivalent
    - For each member of the set of inputs, they give exactly the same output and send the circuit to the same state or to an equivalent state.
  - One of them can be removed without altering the input output relations

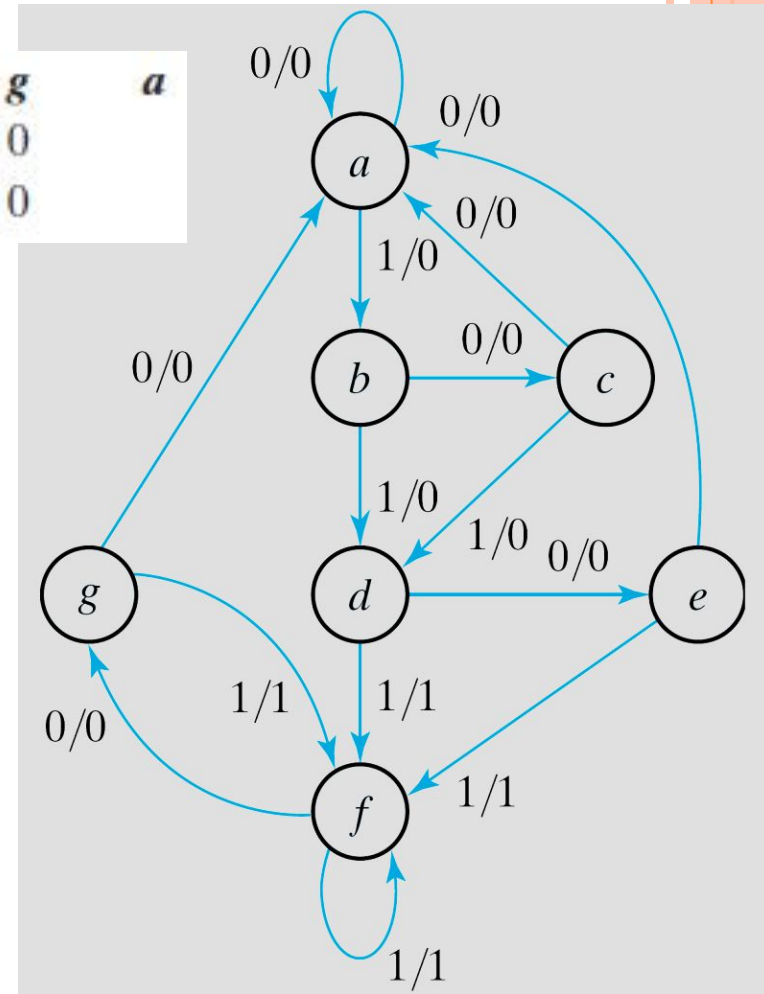


State Reduction Example 1

- Total states : 7
- Consider an Input sequence: 0 1 0 1 0 1 1 0 1 0 0, starting with a, The complete sequence will be:

state	*a*	*a*	*b*	*c*	*d*	*e*	*f*	*f*	*g*	*f*	*g*	*a*
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

- If we can find a circuit whose state diagram has less than 7 states, and identical outputs occur for the above input sequence and all other possible sequences, then the two circuits are said to be equivalent



State Reduction Example 1

- We will apply the algorithm to this state table
- Look for two present states that go to the same next state and have the same output for both input combinations
  - 1. *e* = *g* (remove *g*);
  - 2. Replace all g by e

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	*a*	*b*	0	0
b	*c*	*d*	0	0
c	*a*	*d*	0	0
d	*e*	*f*	0	1
e	*a*	*f*	0	1
f	*g*	*f*	0	1
g	*a*	*f*	0	1

State Reduction Example 1

- $d = f$ (remove f);

Table 5.7
Reducing the State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

State Reduction Example 1

- Reduced state table

Table 5.8
Reduced State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

New output and state sequence for the previous input sequence: 0 1 0 1 0 1 1 0 1 0 0

state	***a***	***a***	***b***	***c***	***d***	***e***	***d***	***d***	***e***	***d***	***e***	***a***
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

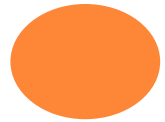
State Reduction Example 1

- Previous state Sequence

state	*a*	*a*	*b*	*c*	*d*	*e*	*f*	*f*	*g*	*f*	*g*	*a*
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

- New state Sequence

state	*a*	*a*	*b*	*c*	*d*	*e*	*d*	*d*	*e*	*d*	*e*	*a*
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	



State Reduction Example 1

- Reduced state Diagram

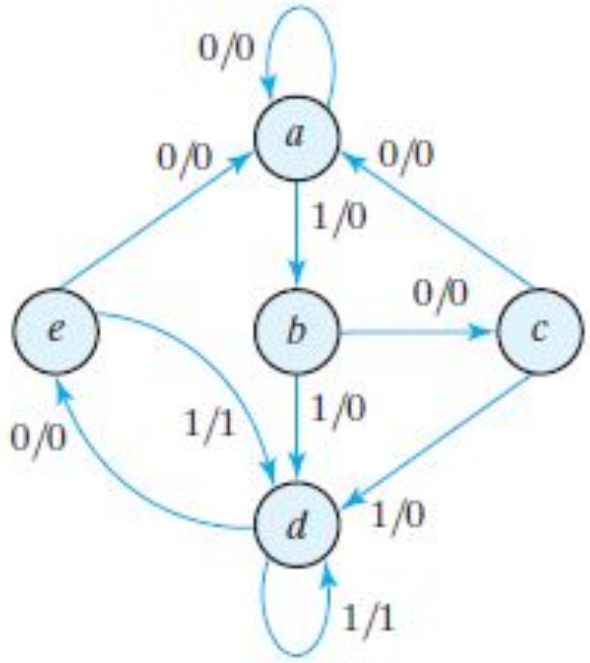
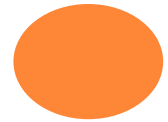


FIGURE 5.26
Reduced state diagram

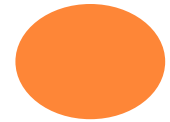
State Reduction Example 2

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
A	F	B	0	0
B	D	C	0	0
C	F	E	0	0
D	G	A	1	0
E	D	C	0	0
F	F	B	1	1
G	G	D	0	1



State Assignment

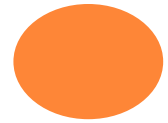
- Assign coded binary values to the states for physical implementation
- For a circuit with m states, the codes must contain n bits where $2^n >= m$
- Unused states are treated as don't care conditions during the design
  - Don't cares can help to obtain a simpler circuit
- There are many possible state assignments
  - Have large impacts on the final circuit size



Popular State Assignment

Table 5.9
Three Possible Binary State Assignments

State	Assignment 1, Binary	Assignment 2, Gray Code	Assignment 3, One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000



State Assignment

- Any binary number assignment is satisfactory as long as each state is assigned a unique number
- Use binary assignment 1

Table 5.10
Reduced State Table with Binary Assignment 1

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

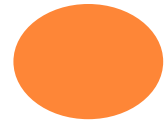
Mealy and Moore Model

Mealy and Moore Model

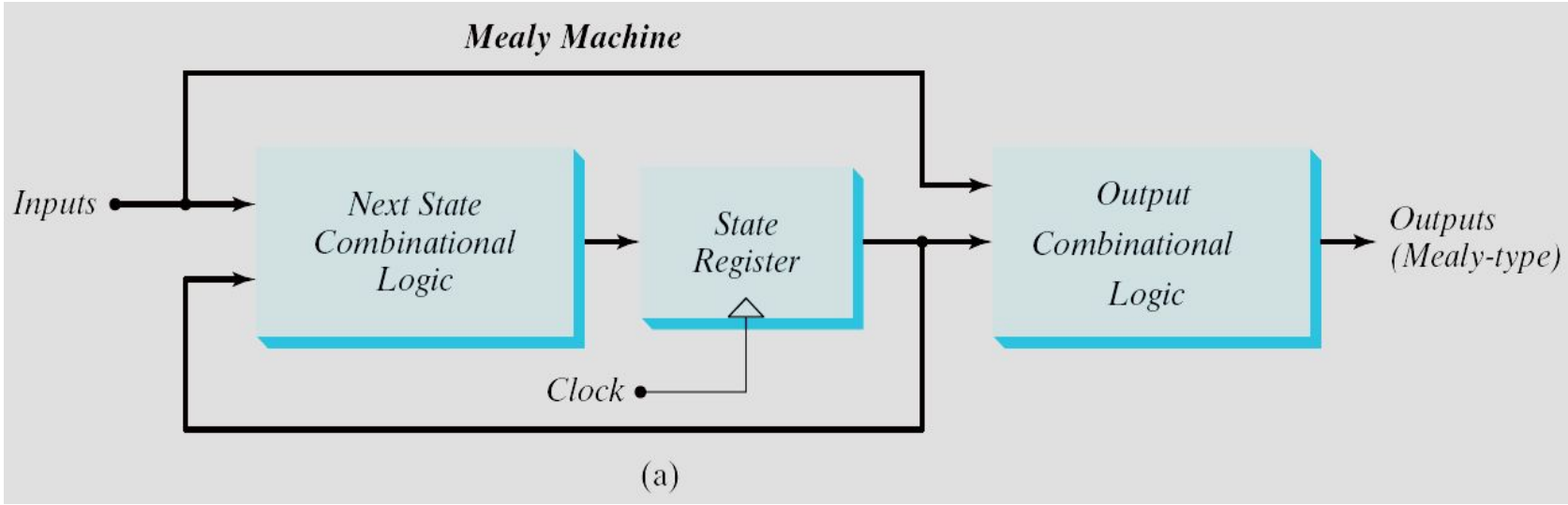
- All the sequential circuits have input, output and internal states
- Sequential circuits can be divided into ***two*** types based on their characteristics:
  - Mealy Model of sequential circuits

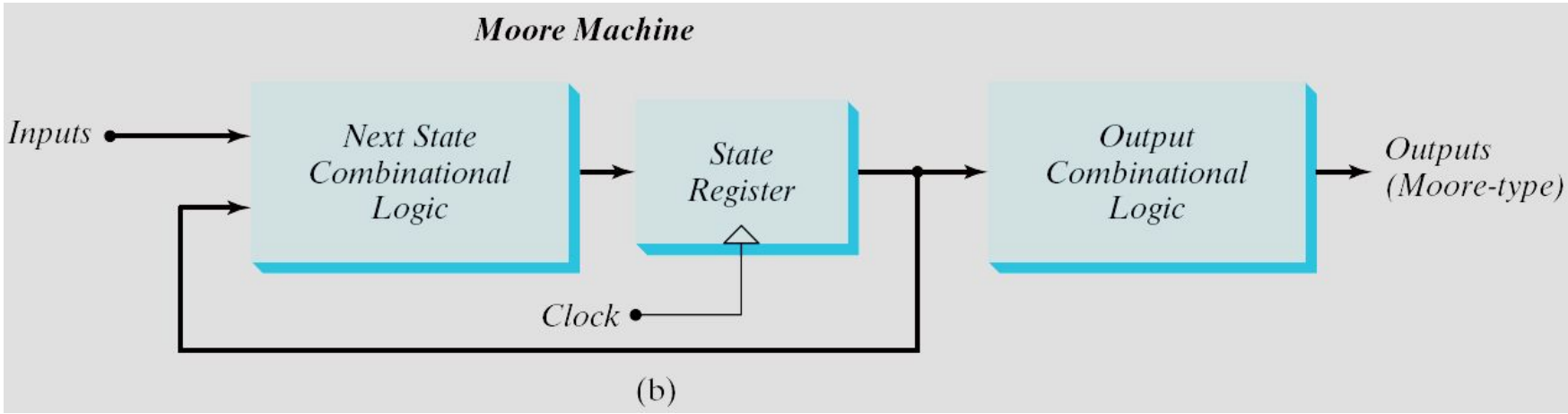
    &

  - Moore Model of sequential circuits
- They differ only in the way their output is generated
  - **Mealy Model:** The output is a function of **both** the present state and the input
  - **Moore Model:** The output is a function of **only** the present state



Block Diagram of Mealy and Moore Machine

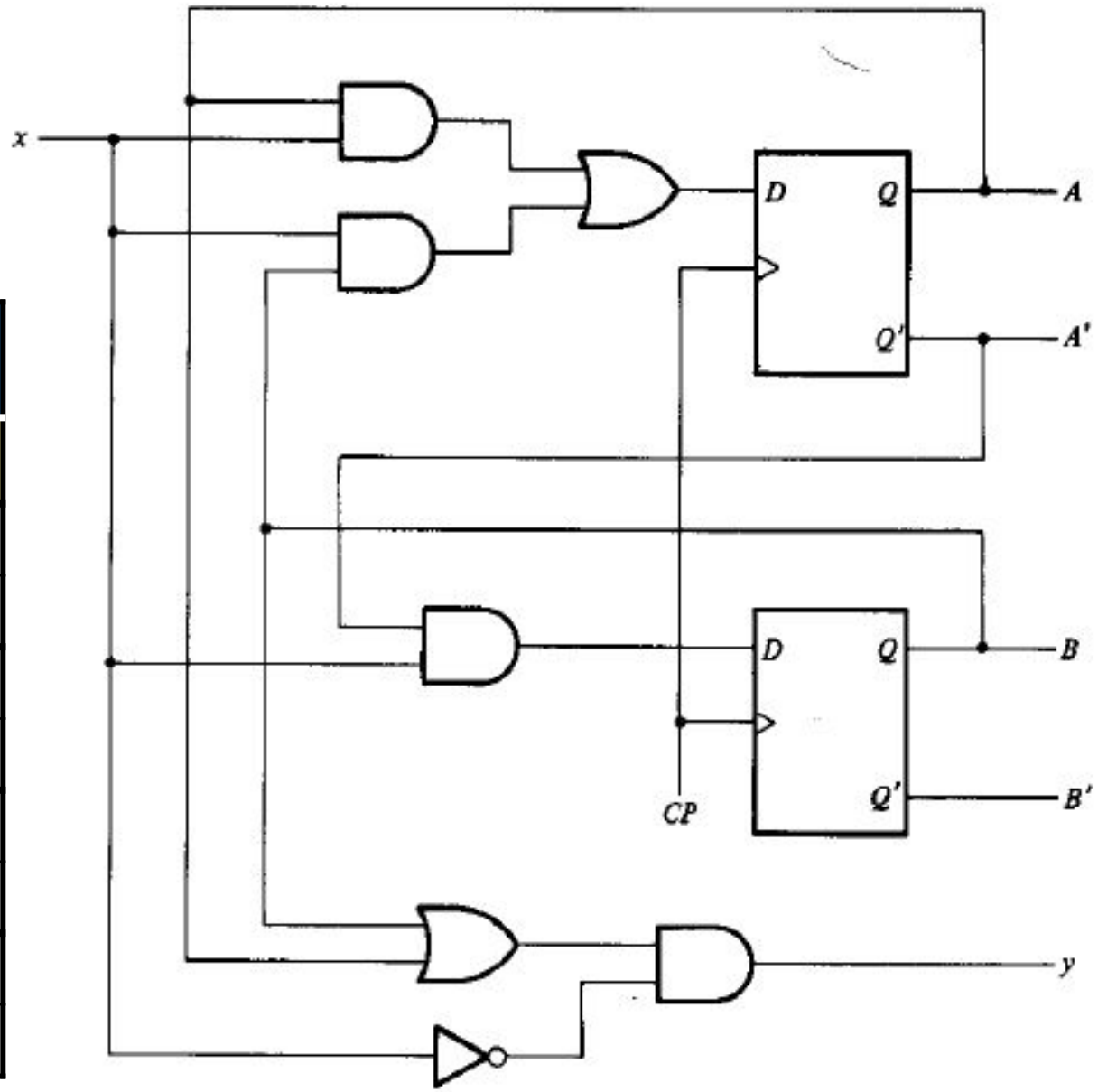




Example of Mealy Machine

- This circuit is an example of Mealy Machine

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0



Example of Mealy Machine

- The output ***y*** is a function of both present state ***AB*** and input ***x.***
- For present state ***01***
  - when input ***x=0;*** output is 1
  - when input ***x=1;*** output is 0

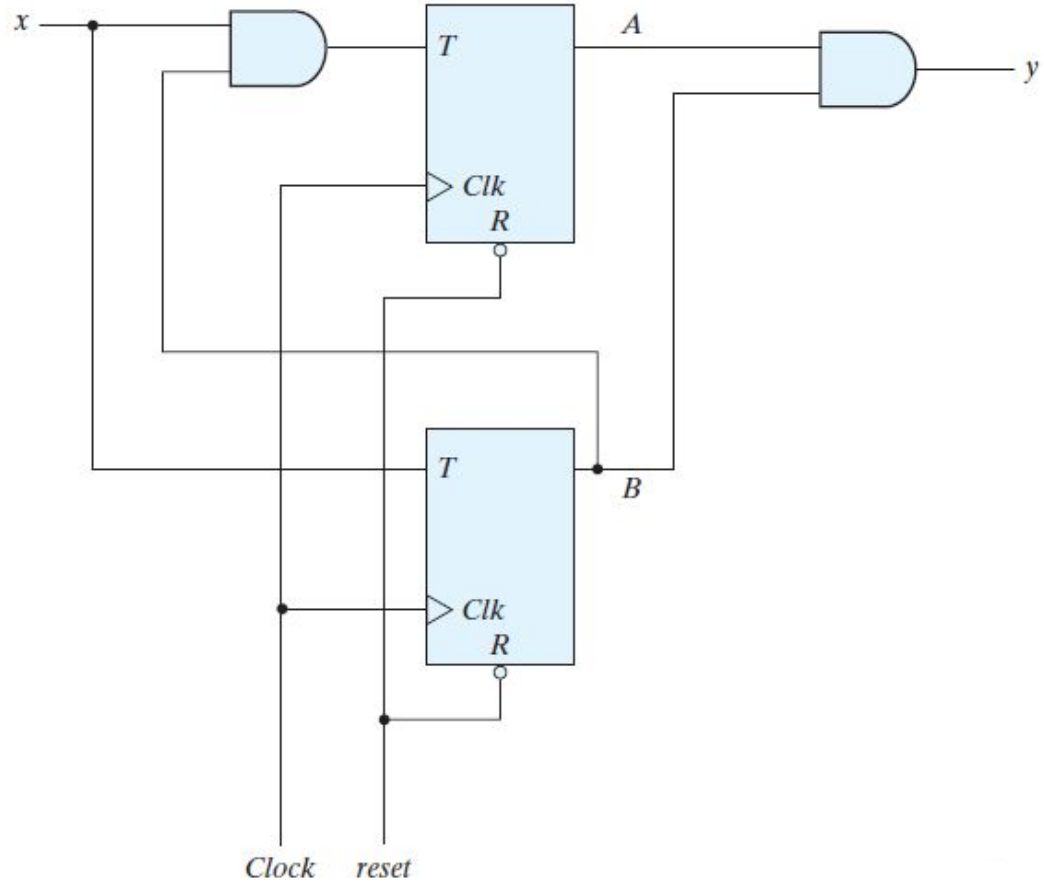
- ***For the same state, the output changes with the input***

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0

Example of Moore Machine

- This circuit is an example of Moore Machine

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	T_A	T_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	1	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1



Example of Mealy Machine

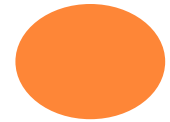
- The output ***y*** is a function of only present state ***AB***
- For present state ***01***
  - when input ***x=0;*** output is 0
  - when input ***x=1;*** output is 0
- This is same for all the combinations
- ***For the same state, the output does not change with the input***

Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	T_A	T_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	1	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1

Design of Sequential Circuit

Steps of Design Procedure

- Derive a state diagram for the circuit
- Reduce the number of states if necessary
- Assign binary values to the states
- Obtain the binary coded state table
- Choose the type of flip flop to be used
- Derive the simplified flip flop input equation and output equation
- Draw the logic diagram



Excitation Tables

TABLE 6-10
Flip-Flop Excitation Tables

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(a) *RS*

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

(b) *JK*

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

(c) *D*

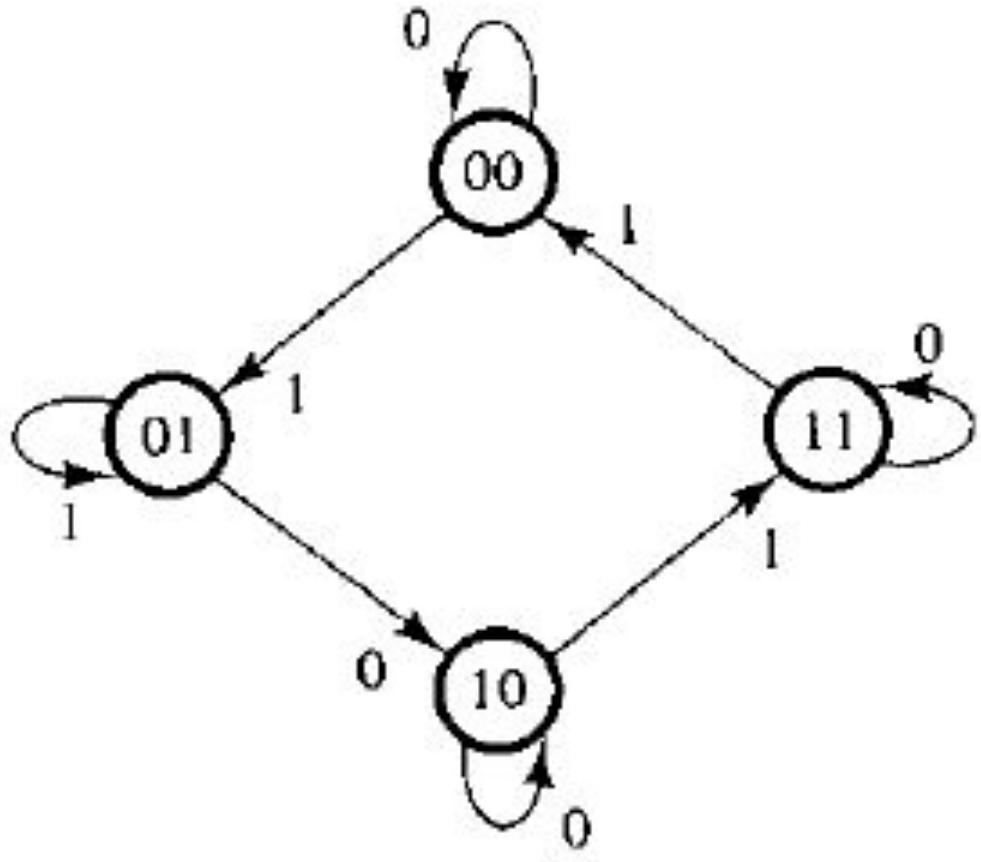
$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) *T*

Example 1: State Diagram For Design

State Table for Circuit Design

Present State		Input	Next State	
A	B	x	A_{t+1}	B_{t+1}
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



Example 1: State Diagram For Design

- Let's design the circuit with JK flip flop first

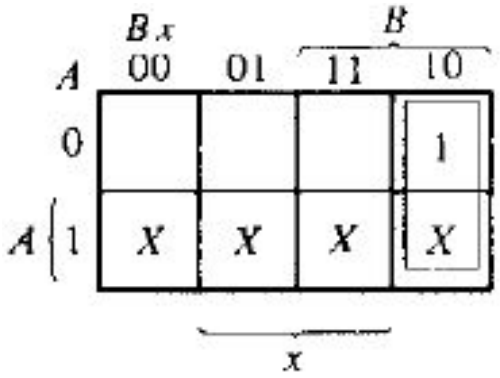
Present State		Input	Next State		Flip Flop Inputs			
A	B	x	A_{t+1}	B_{t+1}	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

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

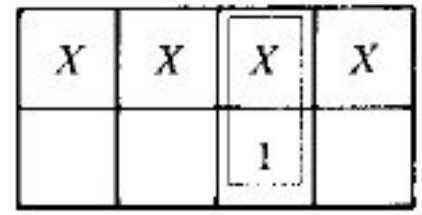
(b) *JK*

Example 1: State Diagram For Design

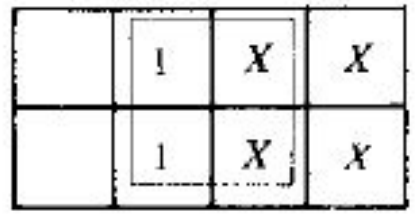
Present State		Input	Next State		Flip Flop Inputs			
A	B	x	A_{t+1}	B_{t+1}	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



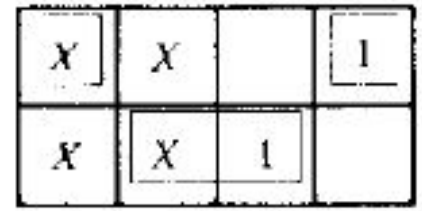
$JA = Bx'$



$KA = Bx$

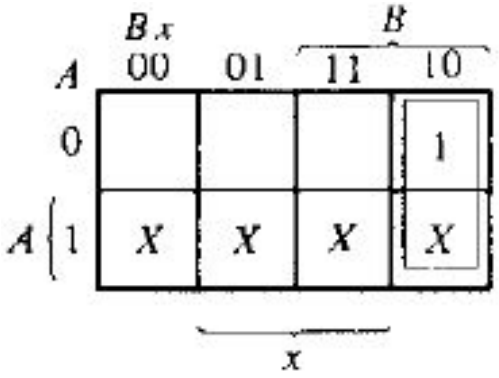


$JB = x$

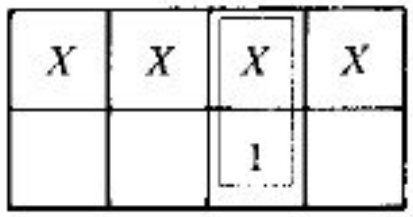


$KB = (A \oplus x)'$

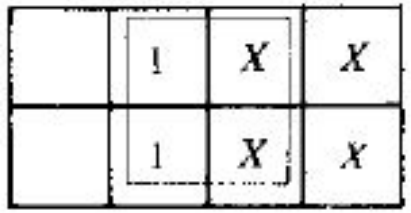
Example 1: State Diagram For Design



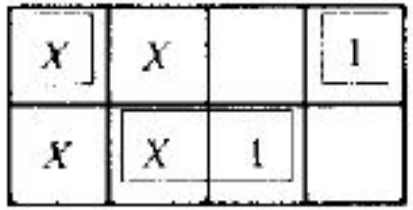
$JA = Bx'$



$KA = Bx$



$JB = x$



$KB = (A \oplus x)'$

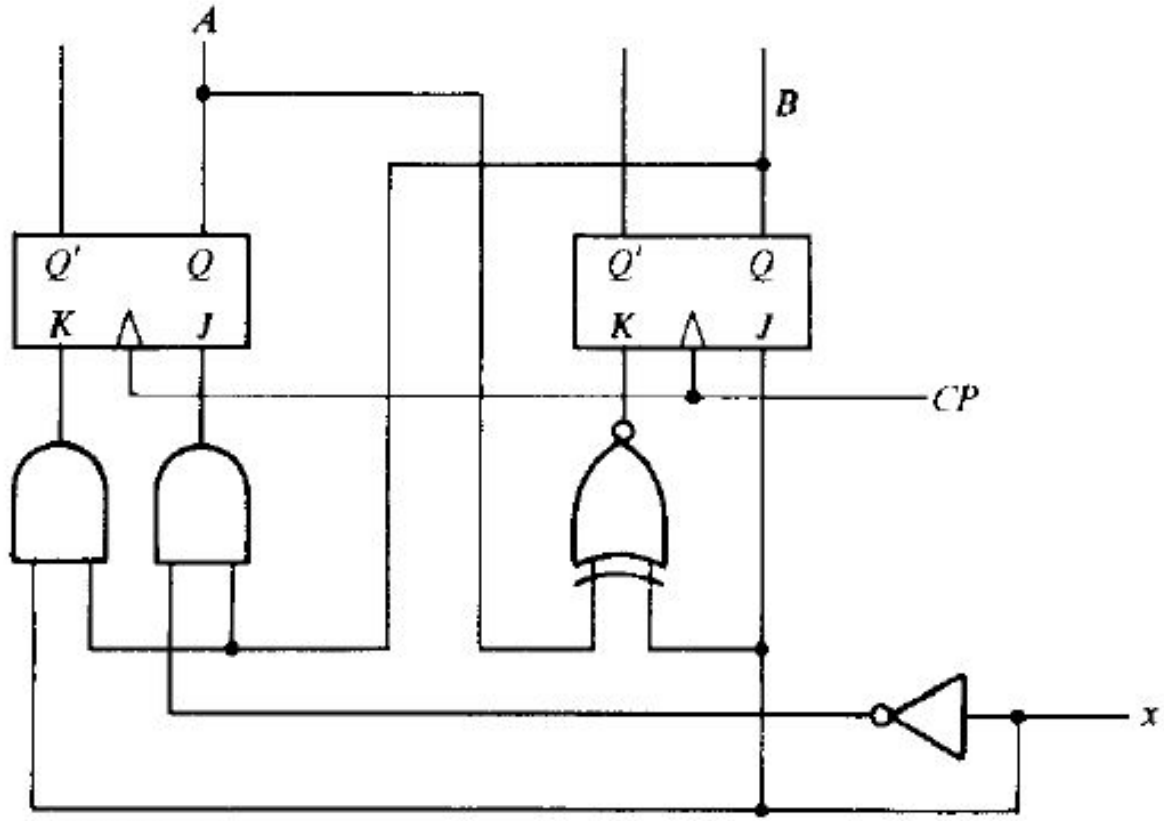
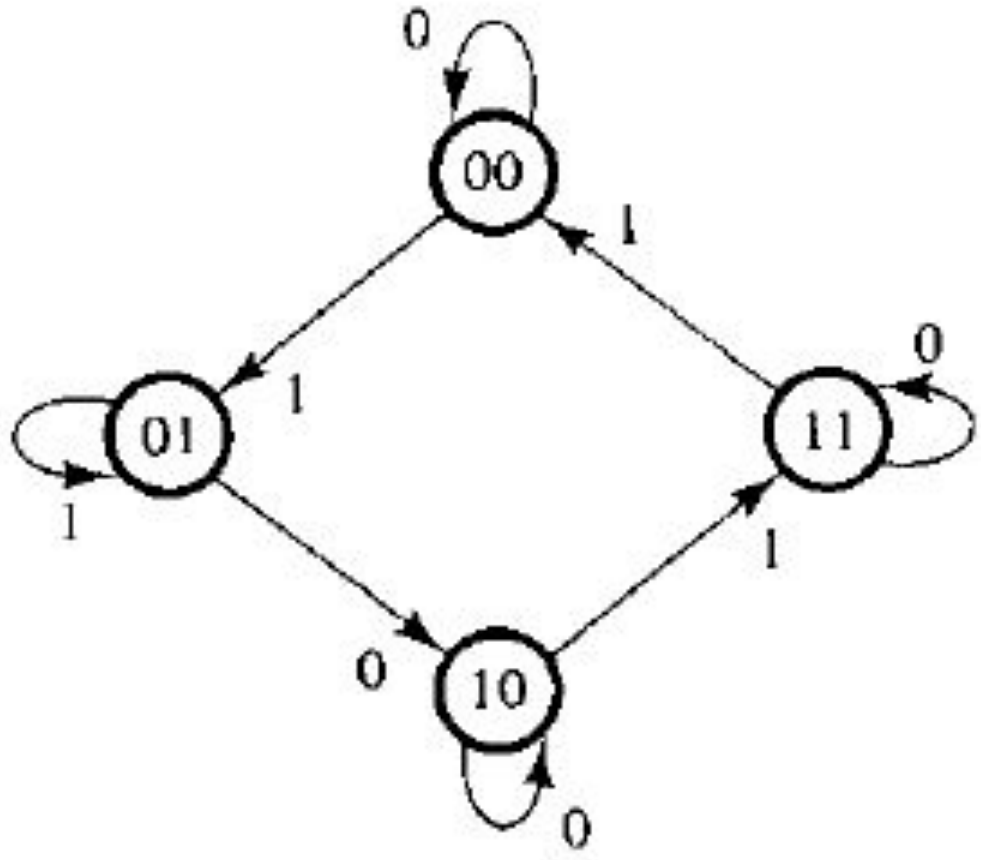


FIGURE 6-26
Logic diagram of sequential circuit

Example 2: State Diagram For Design

State Table for Circuit Design

Present State		Input	Next State	
A	B	x	A_{t+1}	B_{t+1}
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



Example: 2 State Diagram For Design

- Let's design the circuit with D flip flop first

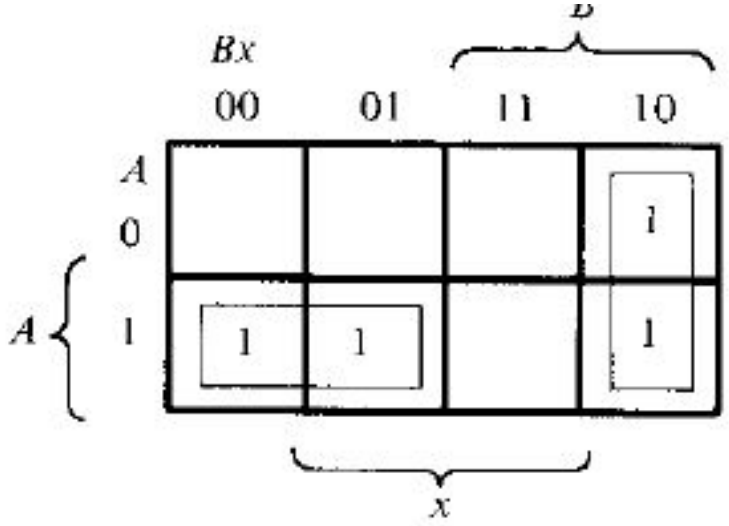
Present State		Input	Next State		Flip Flop Inputs	
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	0	0	0	0

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

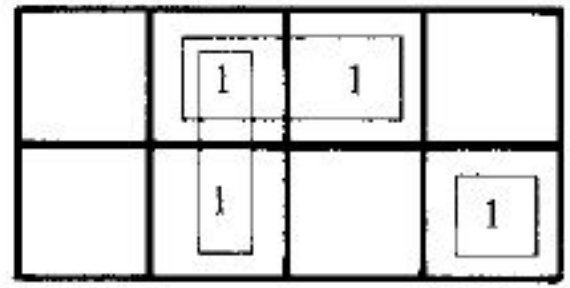
(c) D

Example 2: State Diagram For Design

Present State		Input	Next State		Flip Flop Inputs	
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	0	0	0	0

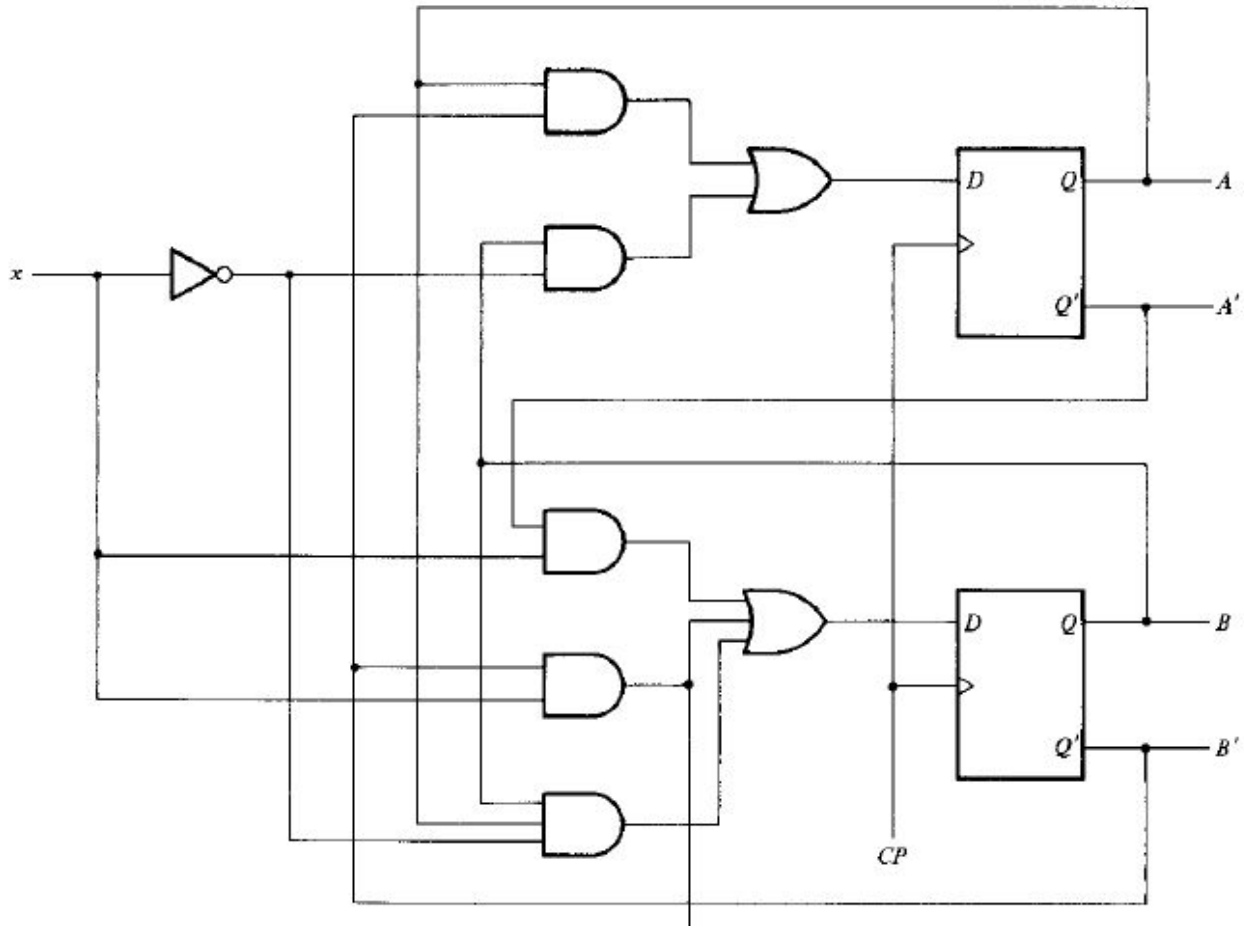


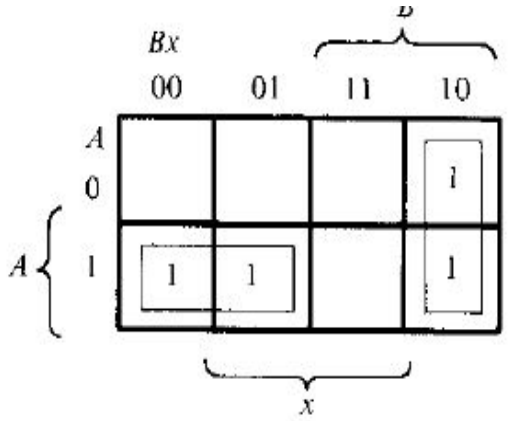
$DA = AB' + Bx'$



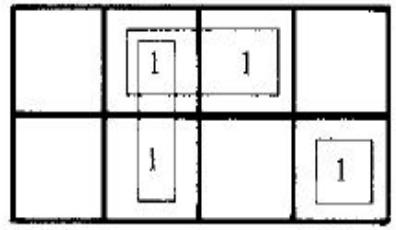
$DB = A'x + B'x + ABx'$

Example 2: State Diagram For Design





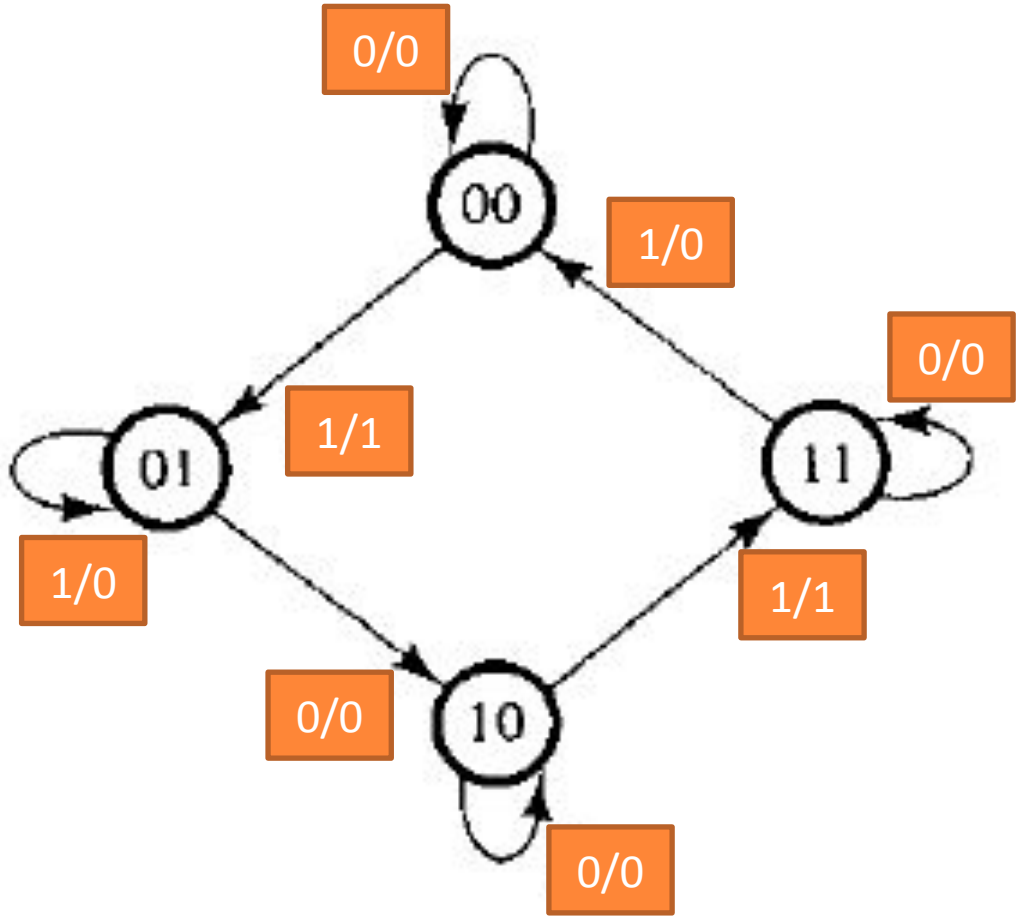
$DA = AB' + Bx'$



$DB = A'x + B'x + ABx'$

Example 3: State Diagram For Design

Present State		Input	Next State		Output
A	*B*	*x*	A_{t+1}	B_{t+1}	*Y*
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0



Example 3: State Diagram For Design

- Let's design the circuit with D flip flop first

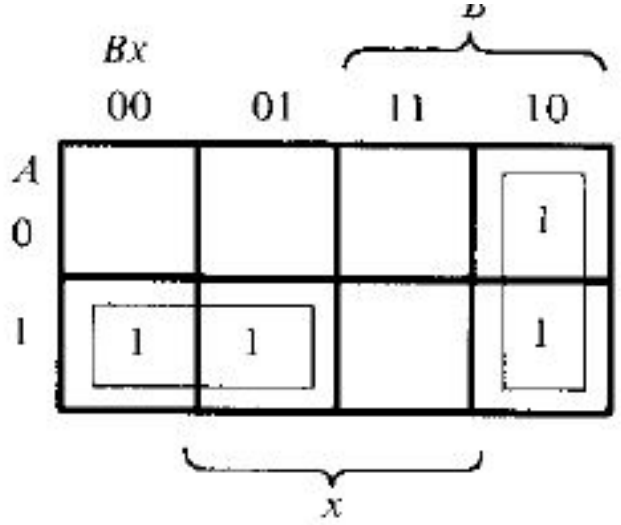
Present State		Input	Next State		Flip Flop Inputs		Output
A	B	x	A_{t+1}	B_{t+1}	D_A	D_B	Y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	0

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

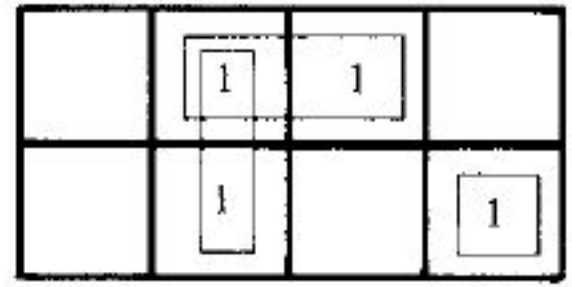
(c) D

Example 3: State Diagram For Design

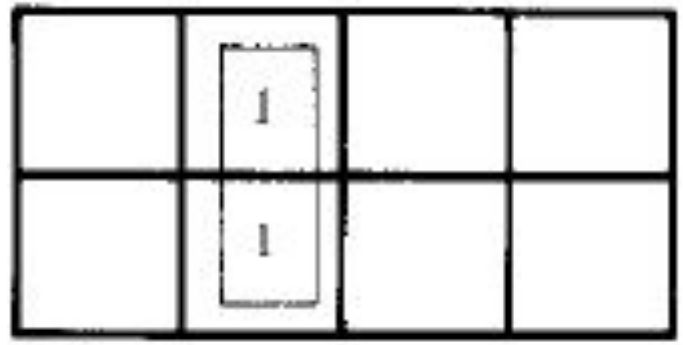
Present State		Input	Next State		Flip Flop Inputs		Output
A	*B*	*x*	A_{t+1}	B_{t+1}	D_A	D_B	*Y*
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	0



$DA = AB' + Bx'$

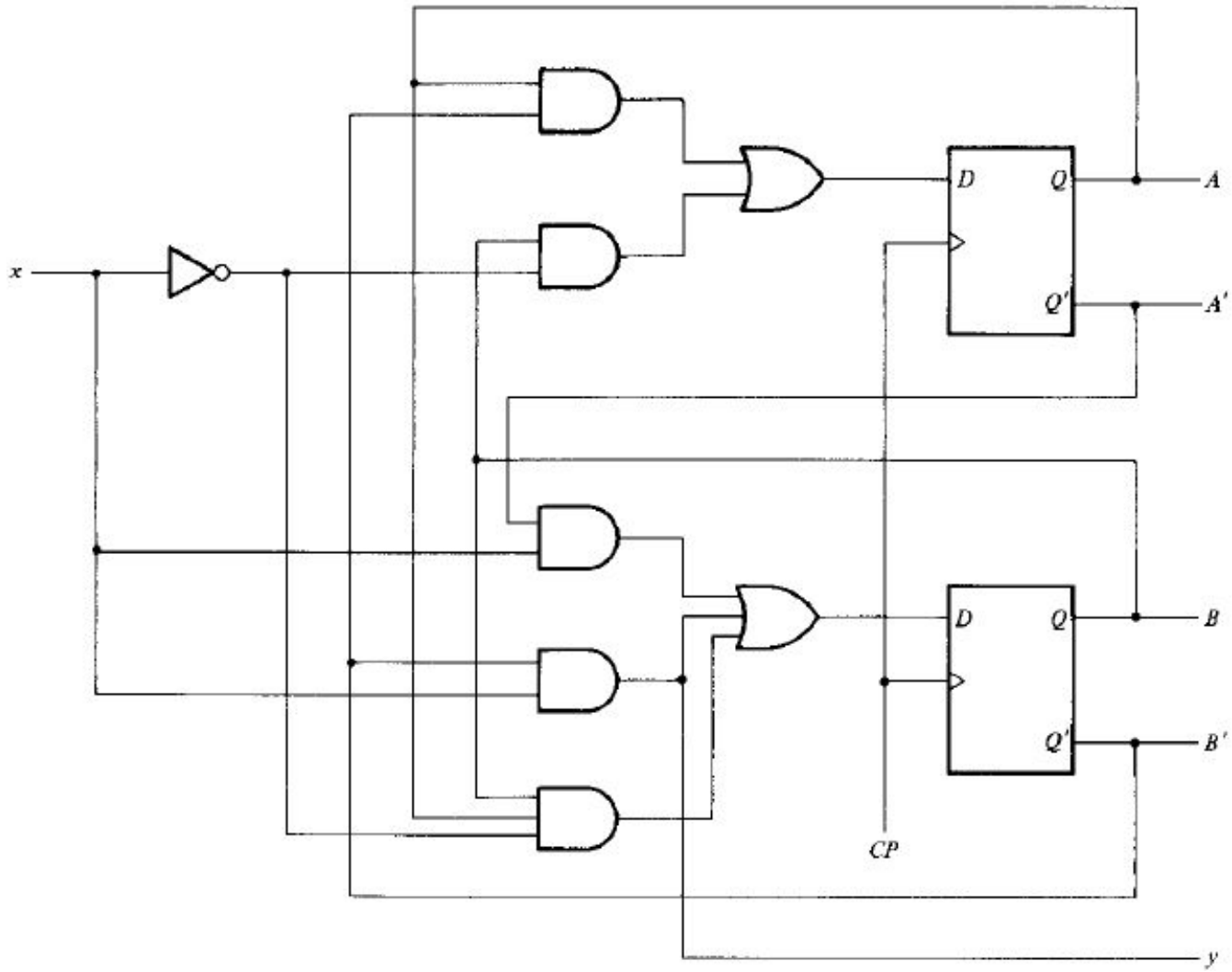


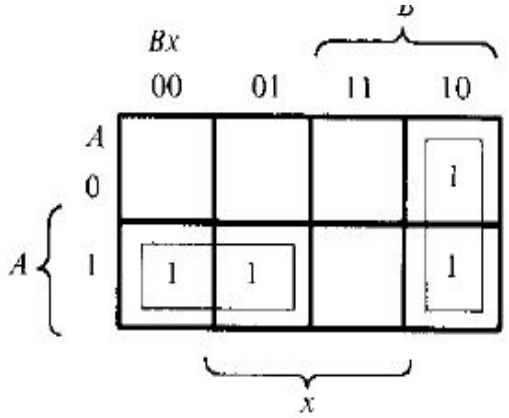
$DB = A'x + B'x + ABx'$



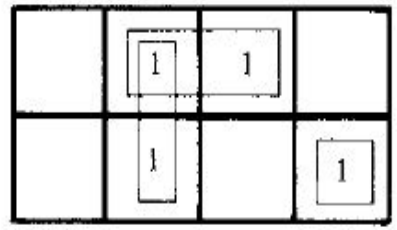
$y = B'x$

Example 3: State Diagram For Design

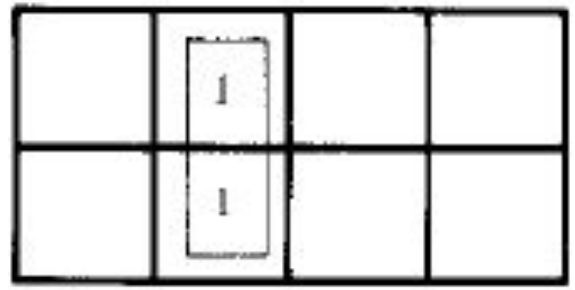




$DA = AB' + Bx'$



$DB = A'x + B'x + ABx'$

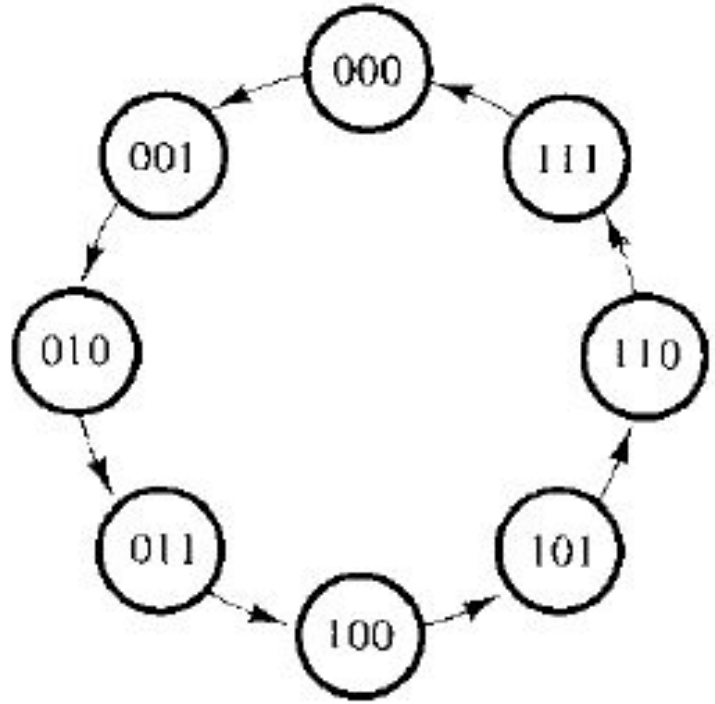


$y = B'x$

Design of Counters

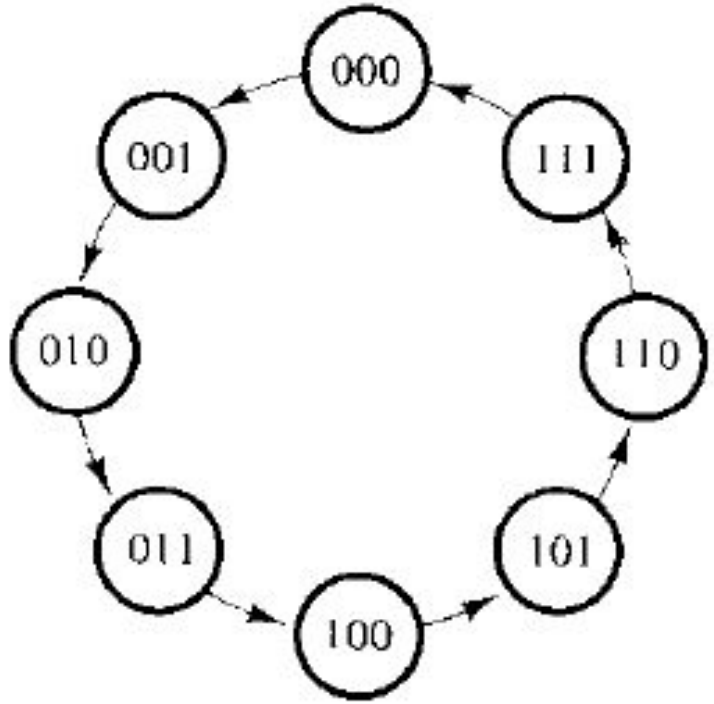
Counters

- An **n-bit binary** counter consists of **n flip-flops**
- Can count in binary from **0 to 2^n-1**
- Here is a state diagram of 3-bit binary counter
- Next state of the counter depends entirely on its present state
- State transition occurs every time a pulse occurs



Example 1: Design of Counters

- Design the counter with the following state diagram with T flip-flop/
- Design a 3-bit binary counter with T flip flop

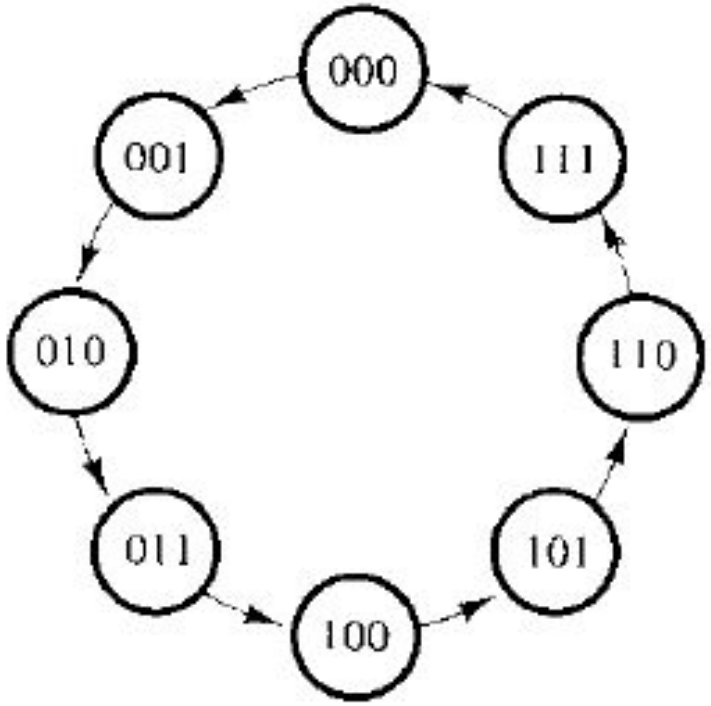


Example 1: Design of Counters

- Design the counter with T flip-flop

State Table for Three-Bit Counter

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1



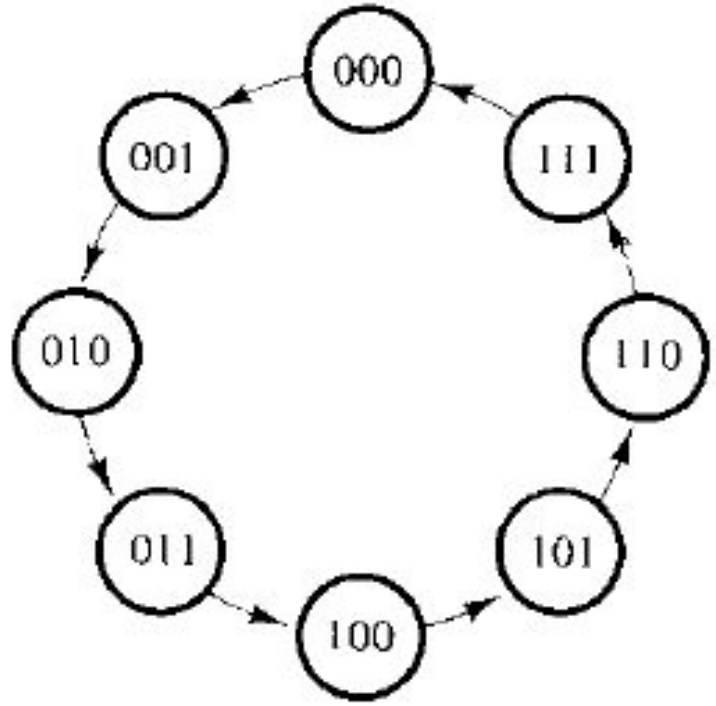
$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

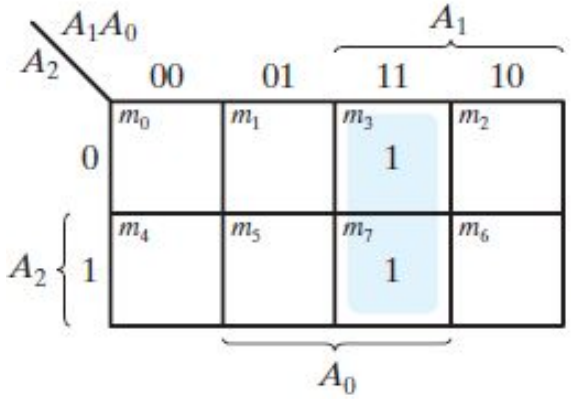
(b) T

Example 1: Design of Counters

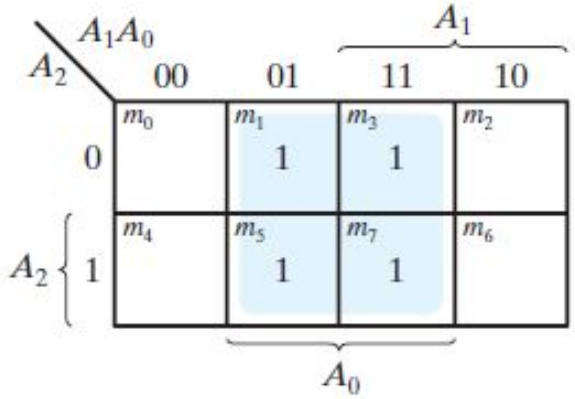
State Table for Three-Bit Counter

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1

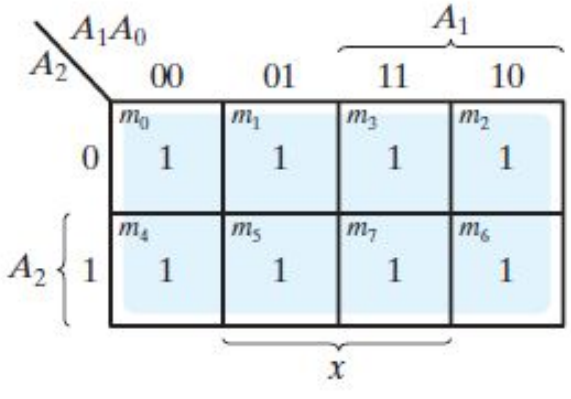




$T_{A2} = A_1A_0$

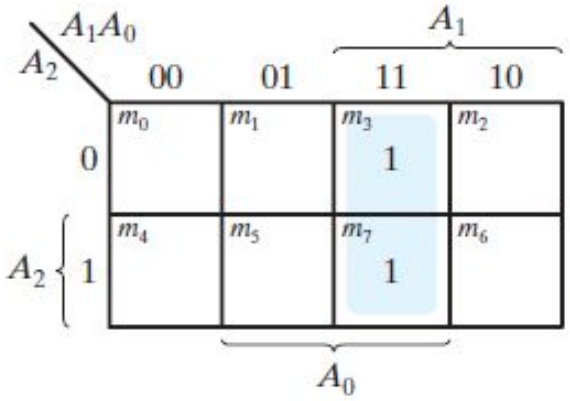


$T_{A1} = A_0$

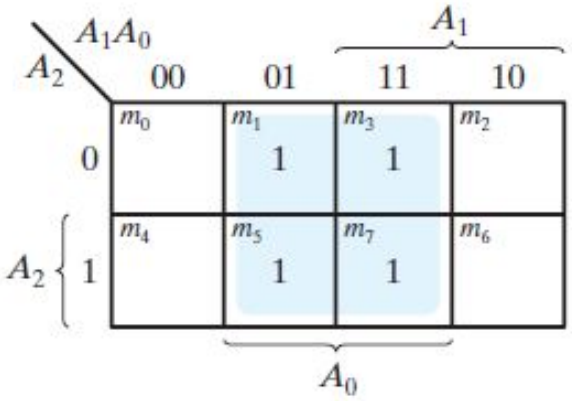


$T_{A0} = 1$

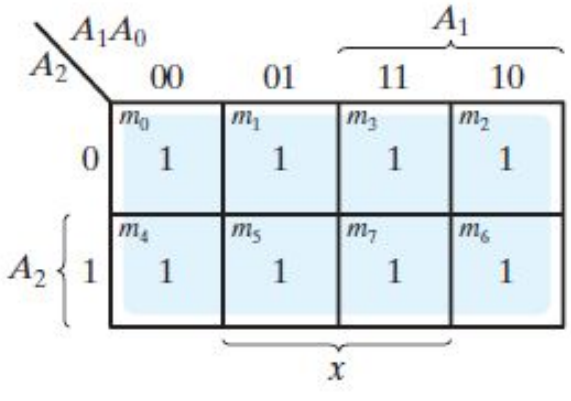
Example 1: Design of Counters



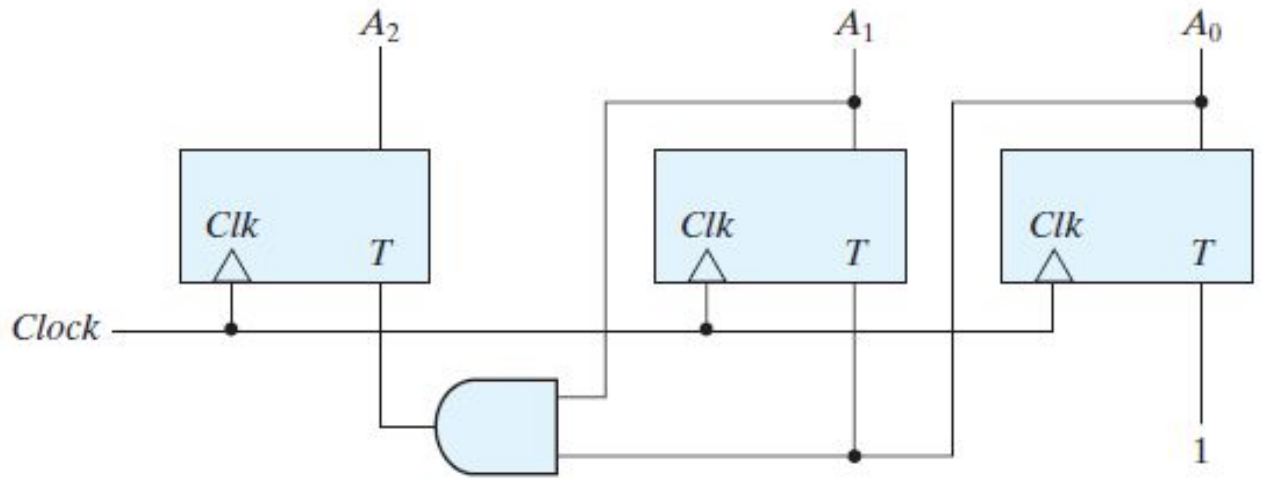
$T_{A2} = A_1A_0$



$T_{A1} = A_0$

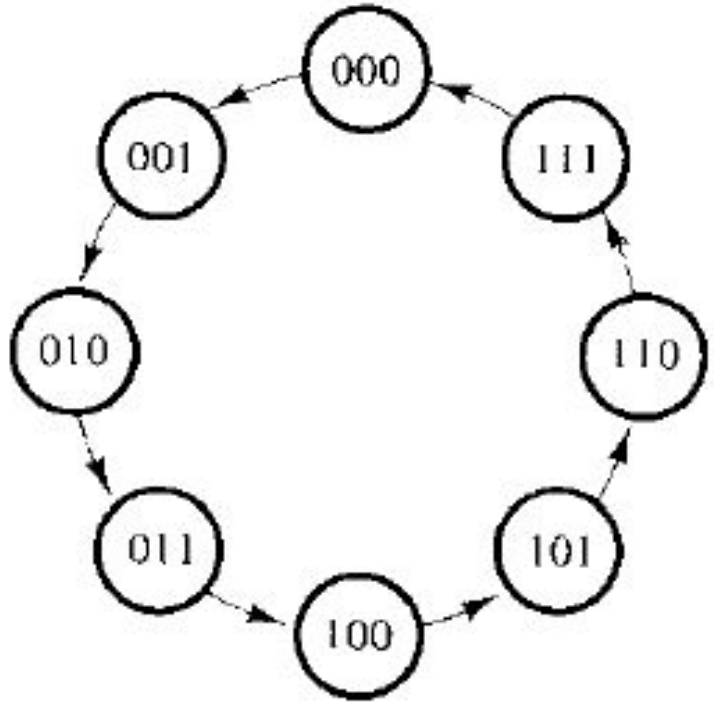


$T_{A0} = 1$



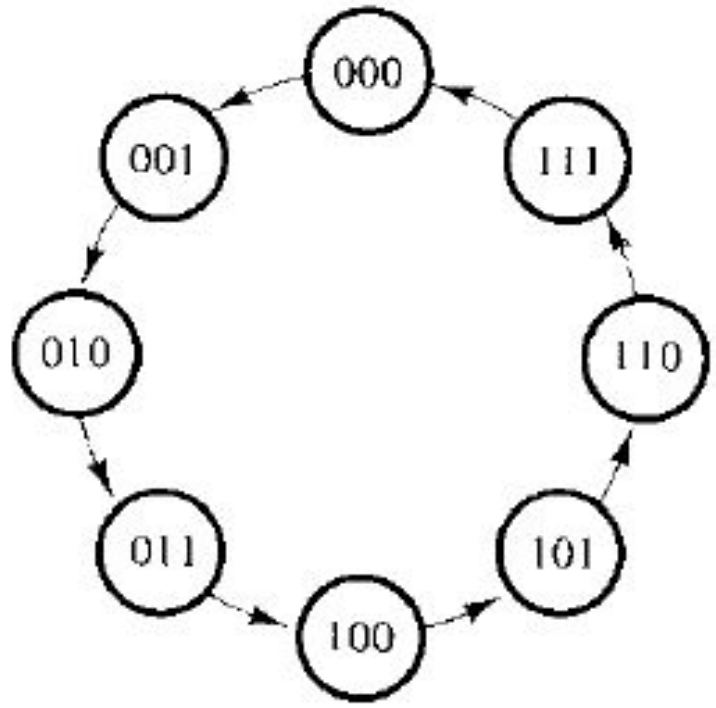
Example 2: Design of Counters

- Design the counter with the following state diagram with JK flip-flop/
- Design a 3-bit binary counter with JK flip-flop



Example 2: Design of Counters

Present State			Next State			Flip Flop Inputs					
A_2	A_1	A_0	A_2	A_1	A_0	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

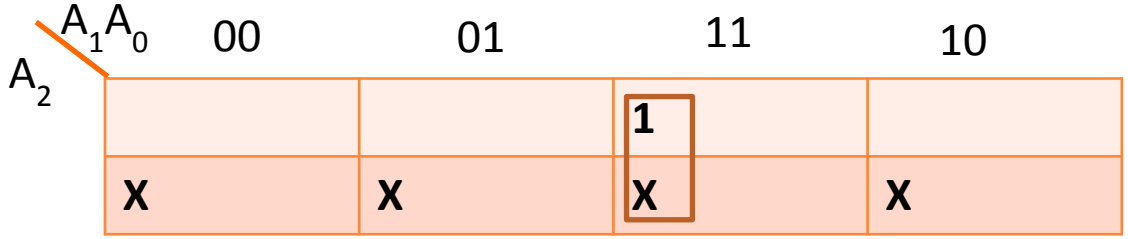


$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

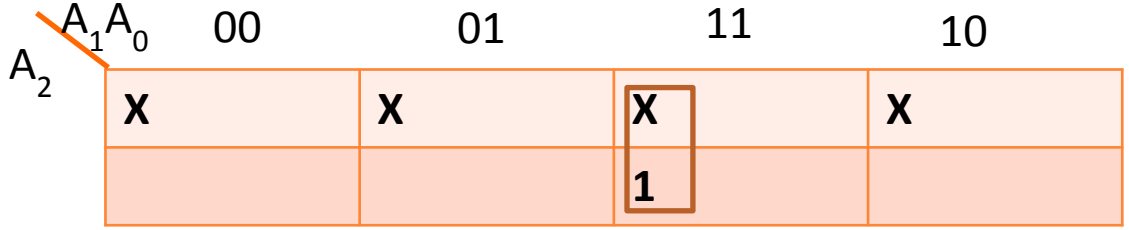
(b) JK

Example 2: Design of Counters

Present State			Next State			Flip Flop Inputs					
A_2	A_1	A_0	A_2	A_1	A_0	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1



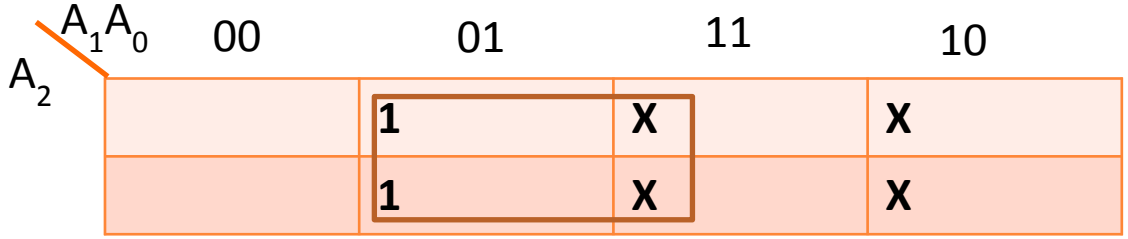
$JA = A_1A_0$



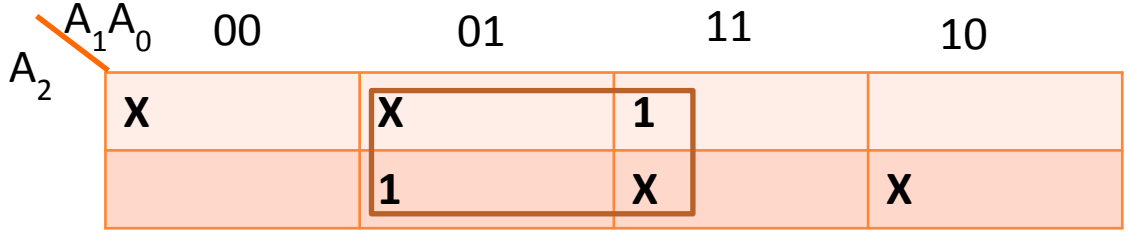
$KA = A_1A_0$

Example 2: Design of Counters

Present State			Next State			Flip Flop Inputs					
A_2	A_1	A_0	A_2	A_1	A_0	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1



JB= A_0



KB = A_0

Example 2: Design of Counters

Present State			Next State			Flip Flop Inputs					
A_2	A_1	A_0	A_2	A_1	A_0	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

A_2 \ A_1A_0	00	01	11	10
	1	X	X	1
	1	X	X	1

JC= 1

A_2 \ A_1A_0	00	01	11	10
	X	1	1	X
	X	1	1	X

KC = 1

Example 2: Design of Counters

- Functions for 3-bit binary counter with JK flip flop

$JA = A_1A_0$ $JB = A_0$ $JC = 1$

$KA = A_1A_0$ $KB = A_0$ $KC = 1$