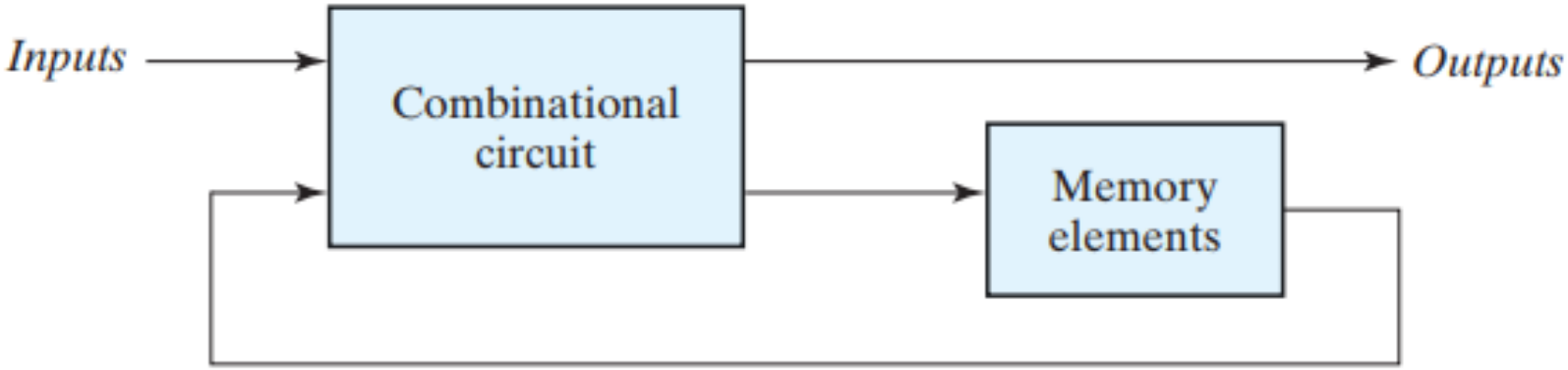
Topic 5: Synchronous Sequential Logic

-- Fahimul Haque (FHE)

* Slides are used as an aiding tool to teach in the classroom. Not every information/details that will be taught in the classes are mentioned on the slides. Hence, you are expected to follow the given textbook(s) for your course.

Sequential Circuit



Block diagram of sequential circuit

A sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

Two main types of sequential circuits:

1. Synchronous Sequential Circuit
2. Asynchronous Sequential Circuit

Storage (Memory) Elements

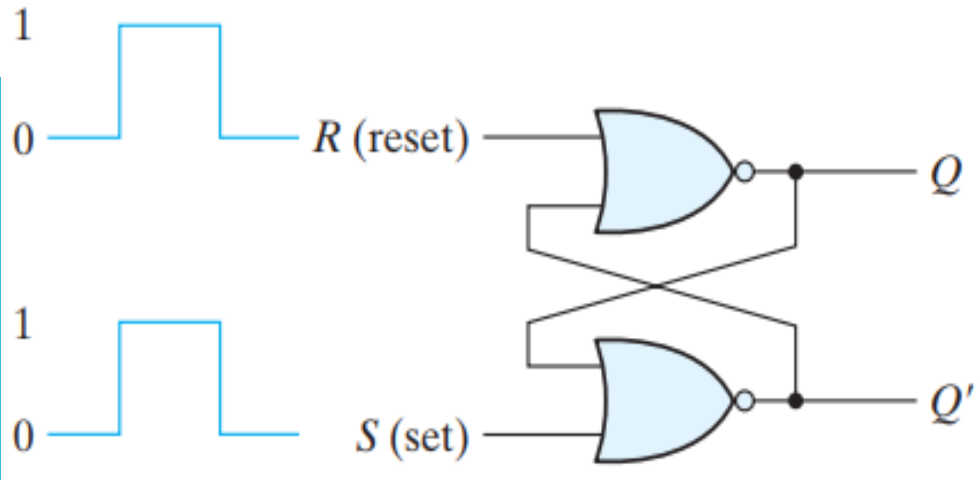
A **storage element** in a digital circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit), until directed by an input signal to switch states.

Storage elements that operate with **signal levels** (rather than signal transitions) are referred to as **latches**; those controlled by a **clock transition** are **flip-flops**. **Latches** are said to be **level sensitive** devices; **flip-flops** are **edge-sensitive** devices.

The two types of storage elements are related because **latches** are the **basic circuits** from which all **flip-flops** are constructed.

Although **latches** are useful for storing binary information and for the design of **asynchronous sequential circuits**, they are not practical for use as storage elements in **synchronous sequential** circuits

SR Latch



(a) Logic diagram

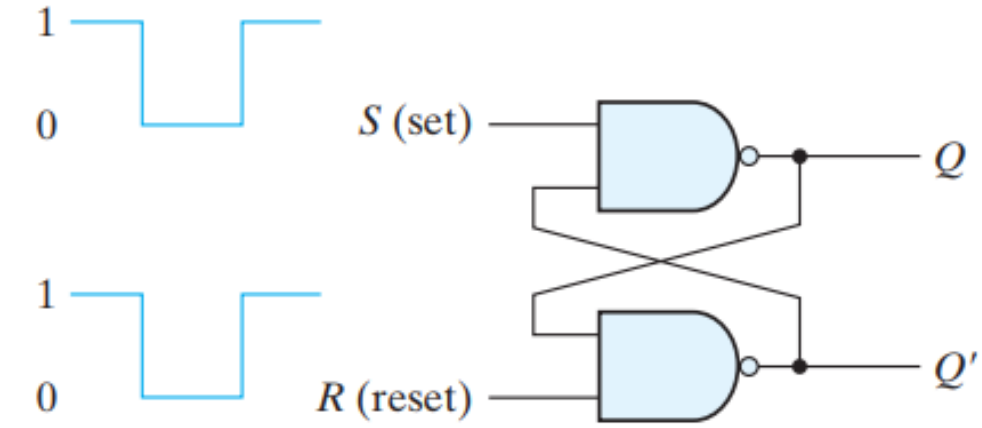
S	R	Q	Q′	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

SR latch with NOR gates

S	R	Next State of Q
0	0	No Change
0	1	Q=0; Reset state
1	0	Q=1; Set state
1	1	Indeterminate

Characteristic Table



(a) Logic diagram

S	R	Q	Q′	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

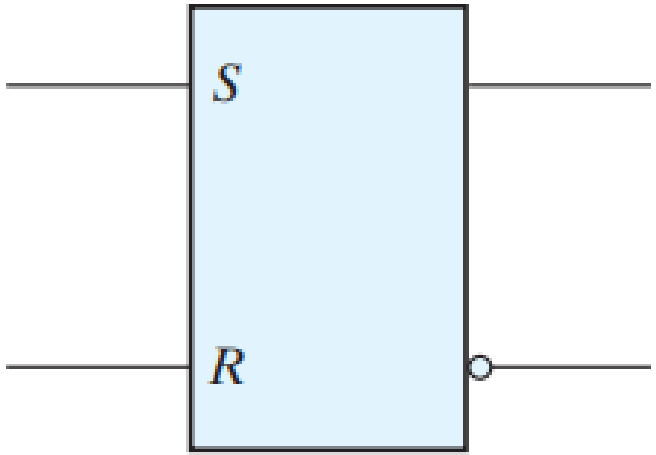
(b) Function table

SR latch (also shown as $S\bar{R}$) with NAND gates

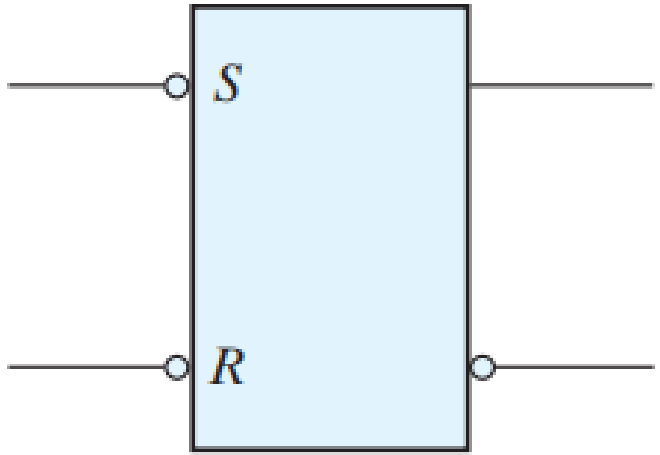
S	R	Next State of Q
0	0	Indeterminate
0	1	Q=1; Set State
1	0	Q=0; Reset State
1	1	No change

Characteristic Table

Graphic Symbols (Block) for *SR* Latches

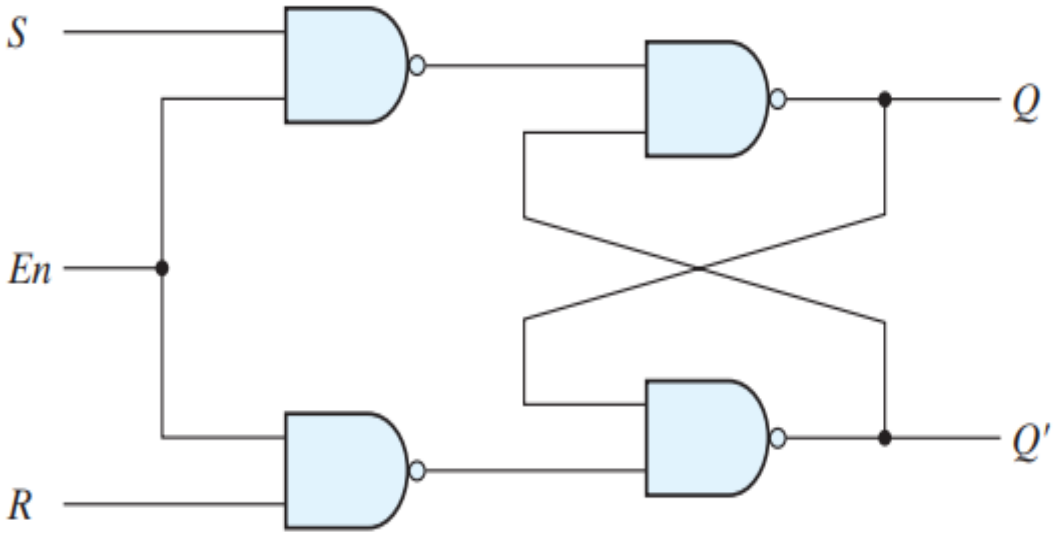


SR



$\overline{S}\overline{R}$

SR Latch with Control (Enable) Input

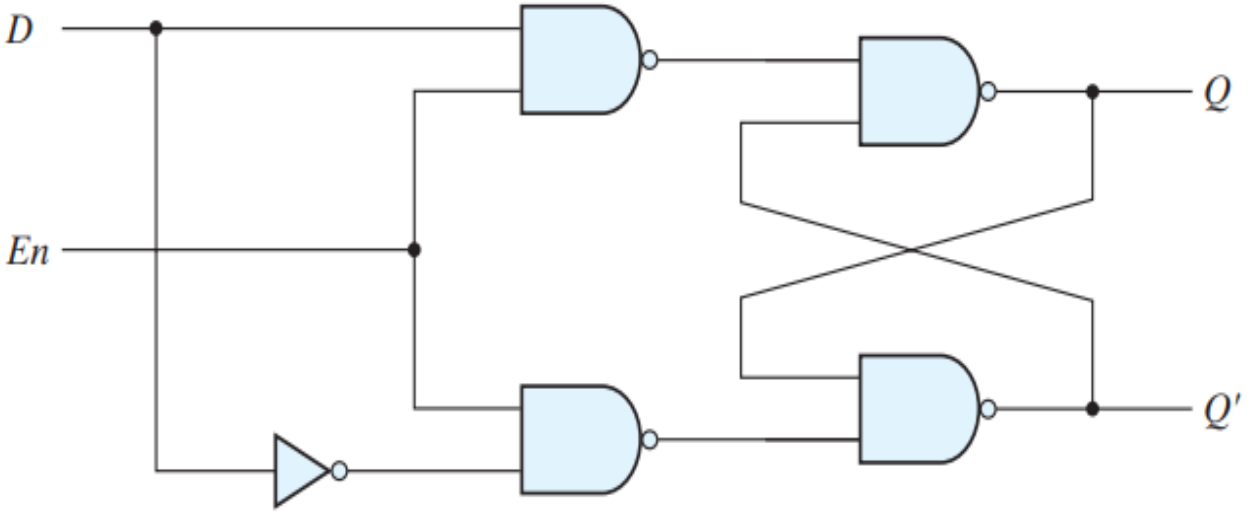


(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

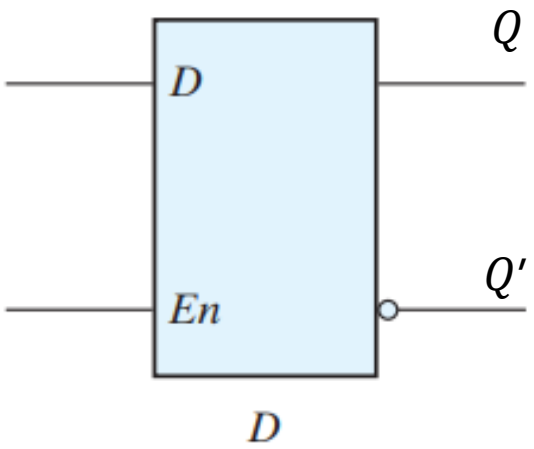
D Latch (Transparent Latch)



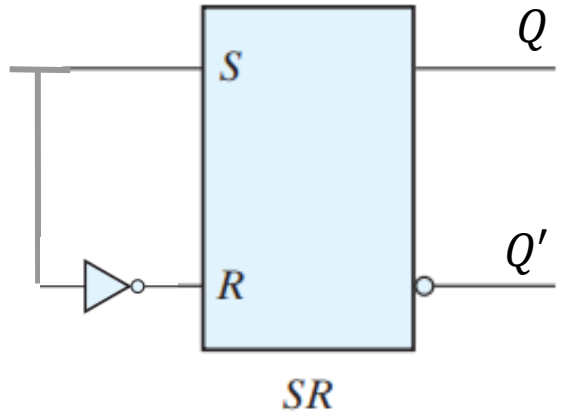
(a) Logic diagram

En	D	Next state of Q
0	X	No change
1	0	Q = 0; reset state
1	1	Q = 1; set state

(b) Function table

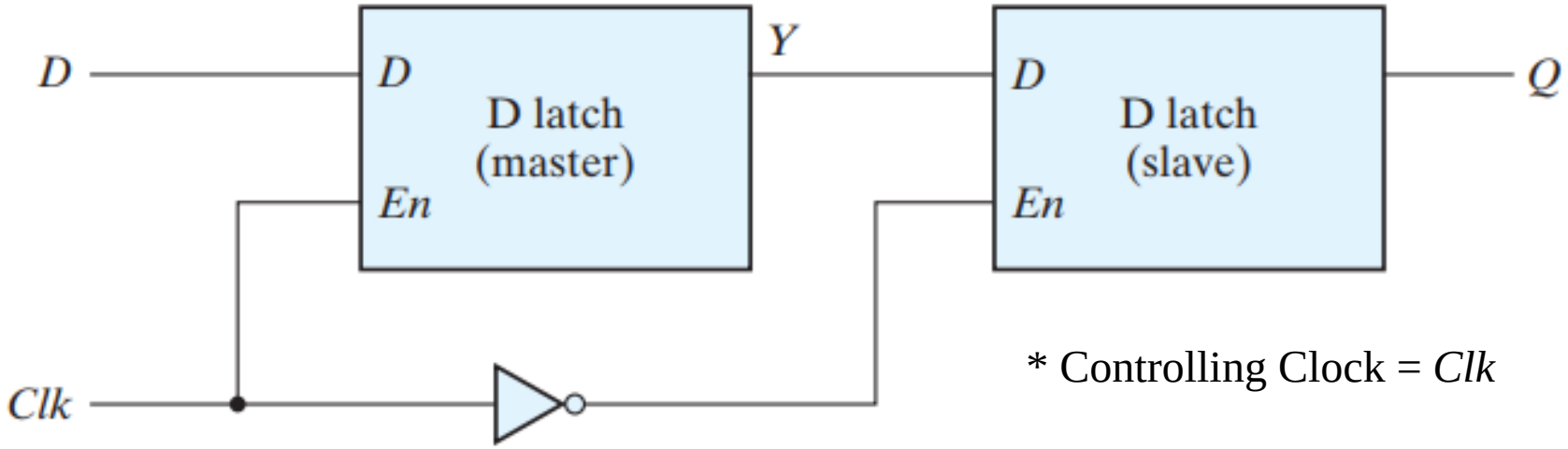


Graphic symbol (Block) for D Latch



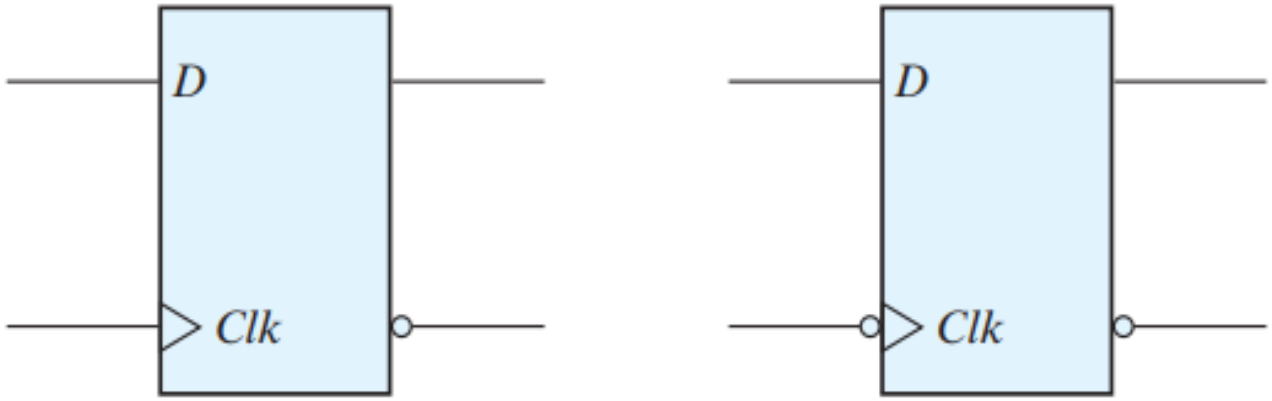
D Latch implementation using SR Latch

D Flip-Flop (FF)



* Controlling Clock = *Clk*

D-type negative-edge FF



(a) Positive-edge (a) Negative-edge

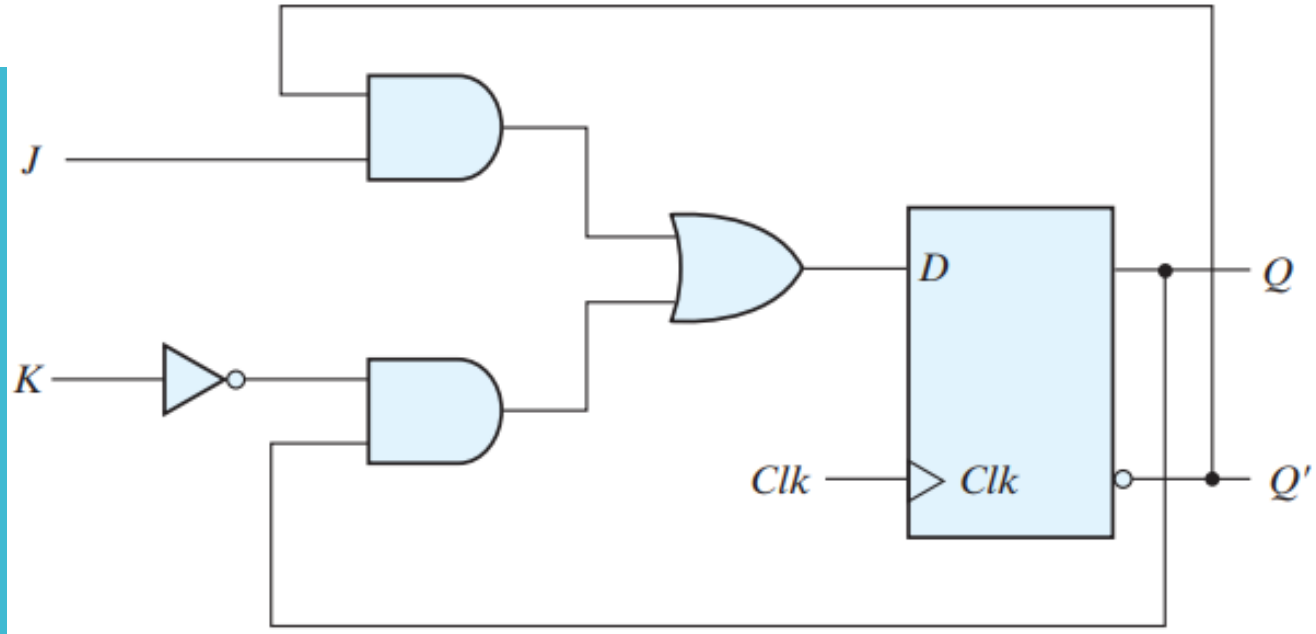
Graphic symbol for edge-triggered *D*-type FF

Characteristic Table

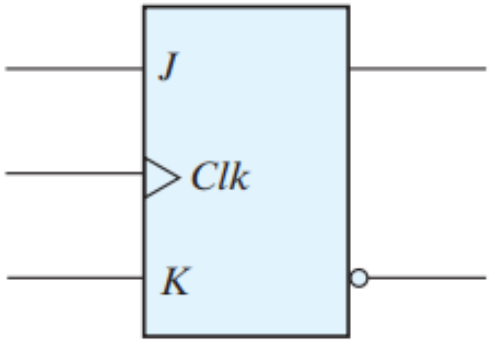
D Flip-Flop

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

JK Flip-Flop (FF)



(a) Circuit diagram



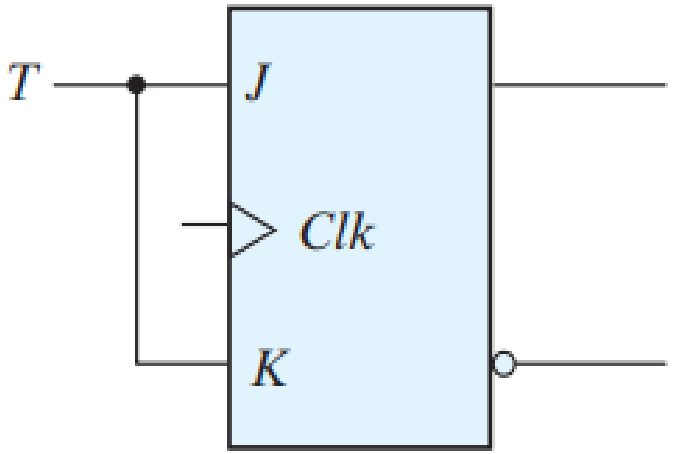
(b) Graphic symbol

Flip-Flop Characteristic Tables

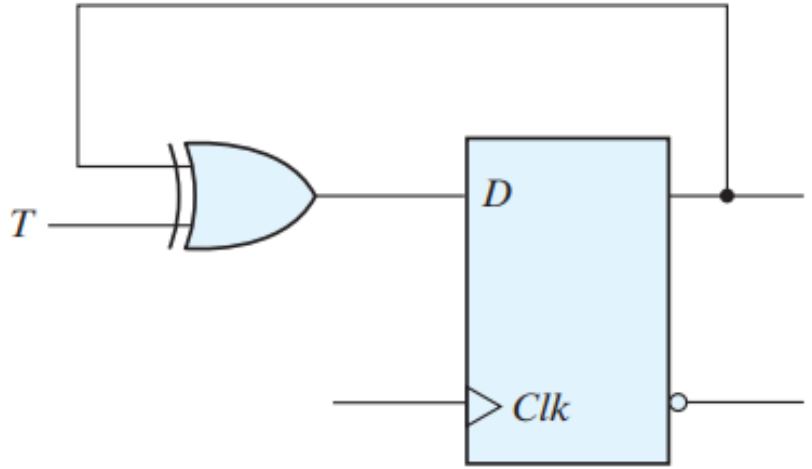
JK Flip-Flop

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

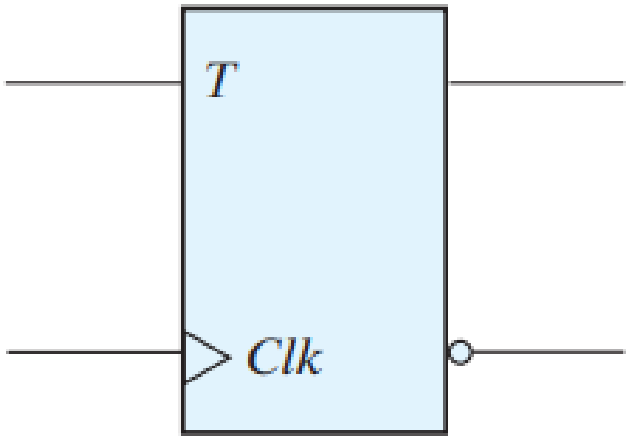
T Flip-Flop (FF)



T FF from *JK* FF



T FF from *D* FF



Graphic Symbol of *T* FF

Characteristic Table

T Flip-Flop

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

Characteristic Table

- A **characteristic table** defines the **logical properties** of a flip-flop by describing its operation in **tabular form**.

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

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

Characteristic Equation

- The **logical properties** of a flip-flop, as described in the characteristic table, can be expressed **algebraically** with a **characteristic equation**.

For *D* Flip-Flop (FF):

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

For *JK* FF (using characteristic table)**:**

$$Q(t+1) = JQ' + K'Q$$

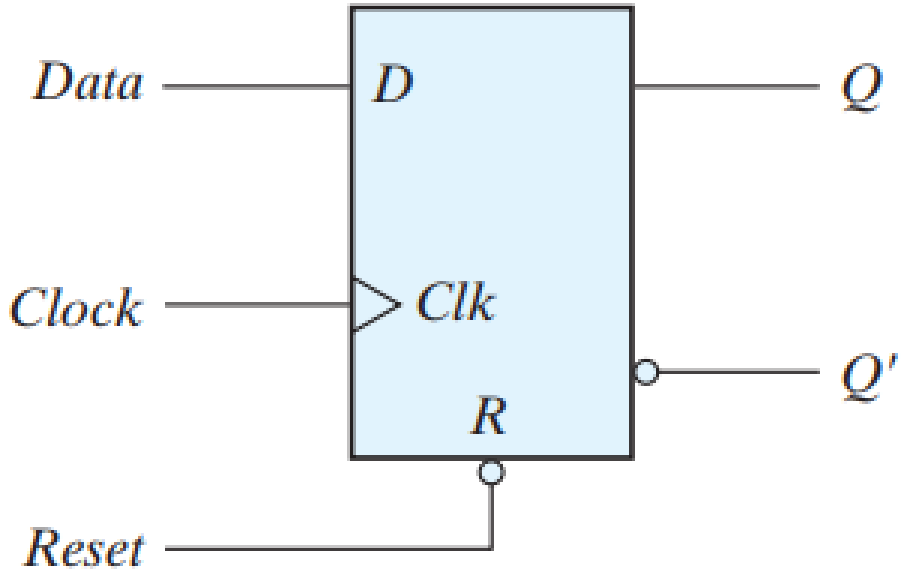
For *T* FF (From diagram using D FF)**:**

$$Q(t+1) = T \oplus Q = TQ' + T'Q$$

* $(t + 1)$ denotes the next state of the flip-flop one clock edge later

Direct Inputs

- Some flip-flops have asynchronous inputs that are used to **force the flip-flop** to a particular state **independently of the clock**. The input that sets the flip-flop to 1 is called ***preset*** or ***direct set***. The input that clears the flip-flop to 0 is called ***clear*** or ***direct reset***.
- When power is turned on in a digital system, the state of the flip-flops is unknown. The direct inputs are **useful** for bringing all flip-flops in the system to a **known starting** state prior to the clocked operation.



Graphic Symbol (block) of a *D* Flip-Flop with direct reset or clear input

State Equations (transition equation)

- A ***state equation*** (also called a *transition equation*) specifies the **next state** as a **function** of the **present state** and **inputs**.
- In other words, a ***state equation*** is an algebraic expression that specifies the condition for a flip-flop state transition.
- The **behavior** of a clocked sequential circuit can be described algebraically by means of state equations.

Flip-Flop Input and Output Equations

- The part of the combinational circuit that generates external outputs is described algebraically by a set of Boolean functions called ***output equations***.
- The part of the circuit that generates the inputs to flip-flops is described algebraically by a set of Boolean functions called flip-flop ***input equations*** (or, sometimes, ***excitation equations***).

- Circuit shown on Slide 16 consists of two *D* flip-flops *A* and *B*, an input *x*, and an output *y*. The logic diagram of the circuit can be expressed algebraically with two flip-flop input equations and an output equation:

$$D_A = Ax + Bx$$
$$D_B = A'x$$
$$y = (A + B)x'$$

Input Equations

Output Equations

State Equations

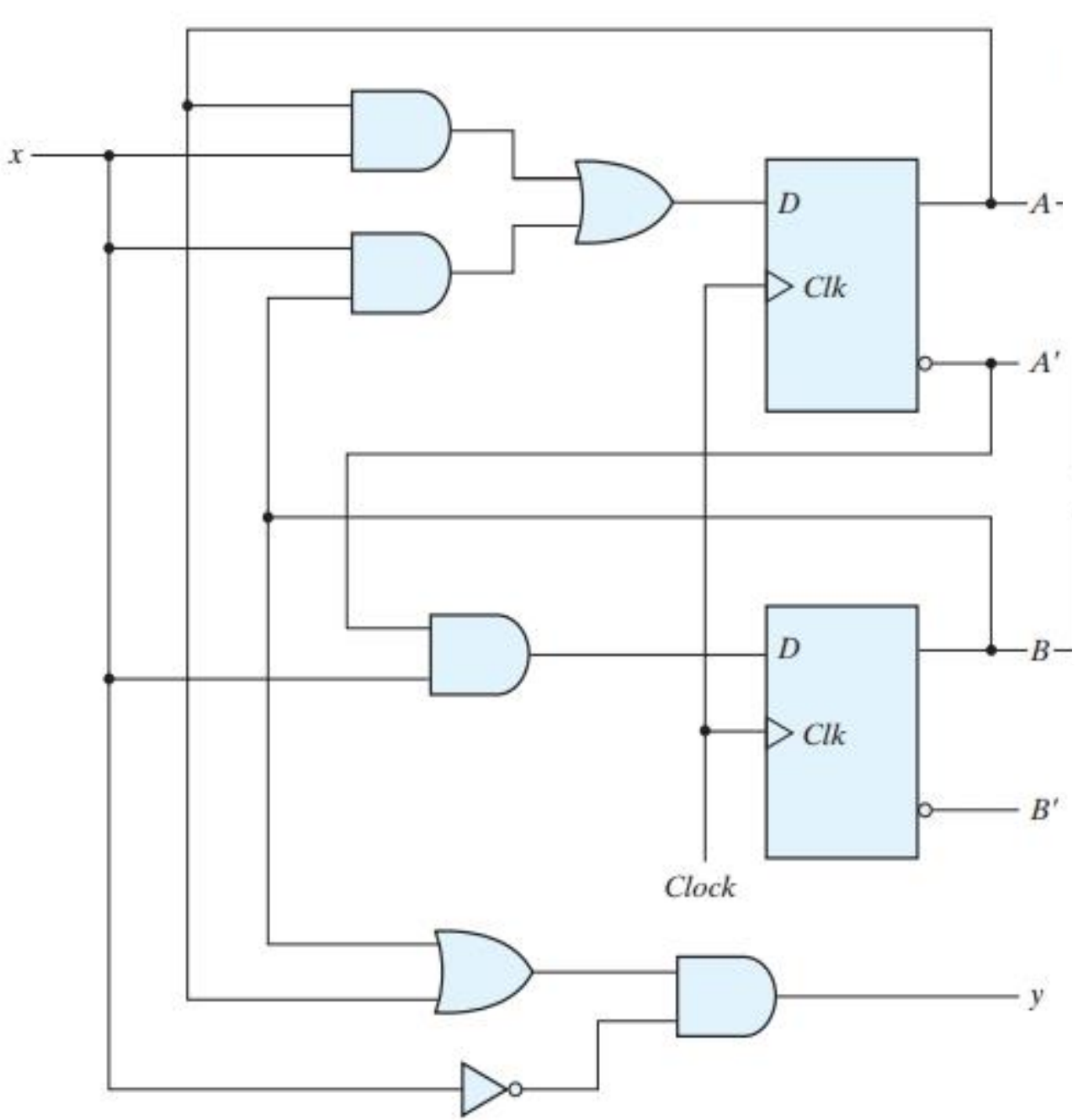


Fig: S16

Consider the given circuit,

$$A(t+1) = A(t)x(t) + B(t)x(t)$$
$$B(t+1) = A'(t)x(t)$$

* $A(t+1)$ or $B(t+1)$denotes the next state of the flip-flops.

* Right side specifies the present state and input that makes next state equal to 1.

For convenience we can omit (t) from the right side, since every variable on the rights side is a function of present time.

$$A(t+1) = Ax + Bx$$
$$B(t+1) = A'x$$

Output Equation: $y = (A + B)x'$

State Table

- The time sequence of inputs, outputs, and flip-flop states can be enumerated in a ***state table*** (sometimes called a *transition table*).
- The table consists of **four sections** labeled *present state*, *input*, *next state*, and *output*.
- The derivation of a state table requires listing all possible binary combinations of present states and inputs.
- The next-state values are then determined from the logic diagram or from the state equations.

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

State Table for the circuit on the slide 16 (**Fig: S16**)

2nd Form of State Table

1st form of state table

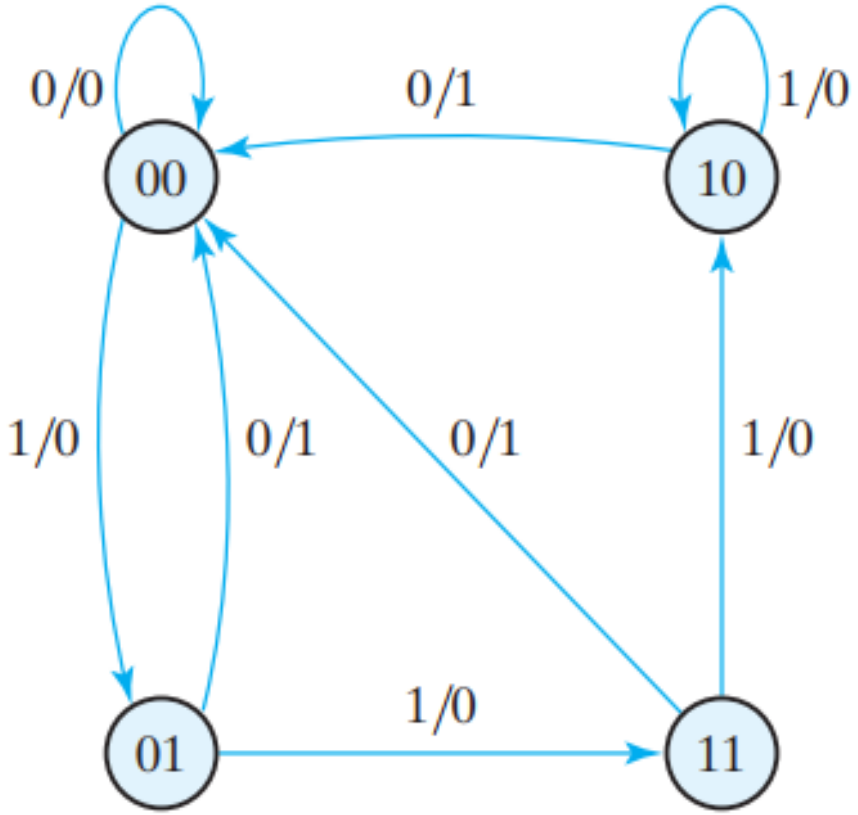
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Second Form of the State Table

Present State		Next State				Output	
		x = 0		*x* = 1		*x* = 0	*x* = 1
A	*B*	*A*	*B*	*A*	*B*	*y*	*y*
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

State Diagram

- The information available in a state table can be represented graphically in the form of a state diagram.
- In this type of diagram, a state is represented by a circle, and the (clock-triggered) transitions between states are indicated by directed lines connecting the circles.



State diagram for the circuit on the slide 16 (**Fig: S16**)

Analysis of a Flip-Flop Circuit

During analysis, our goal is to come up with a state table and state diagram for a given flip-flop circuit or flip-flop input equation(s).

1st Method:

1. Determine the flip-flop input equations in terms of the present state and input variables.

2. List the binary values of each input equation.

3. Use the corresponding flip-flop characteristic table to determine the next-state values in the state table.

2nd Method:

1. Determine the flip-flop input equations in terms of the present state and input variables.

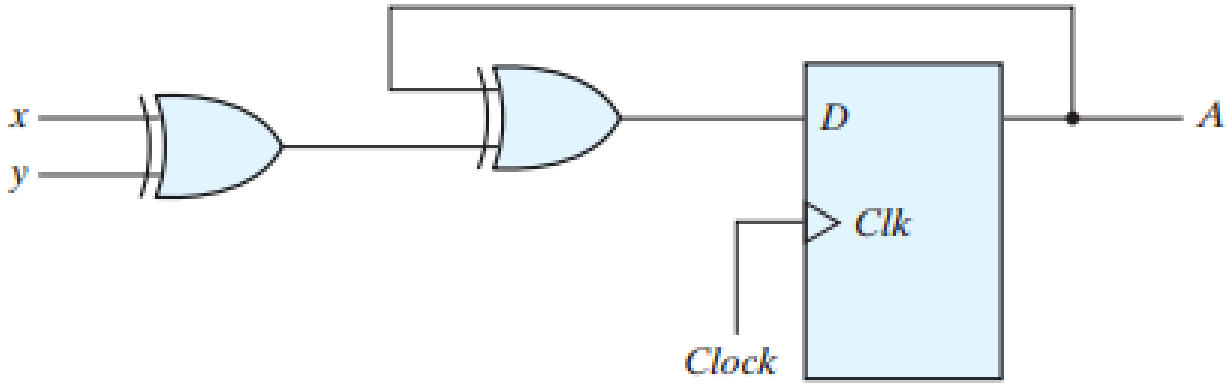
2. Substitute the input equations into the flip-flop characteristic equation to obtain the state equations.

3. Use the corresponding state equations to determine the next-state values in the state table.

Analysis with D Flip-Flops

Consider the input equation: $D_A = A \oplus x \oplus y$

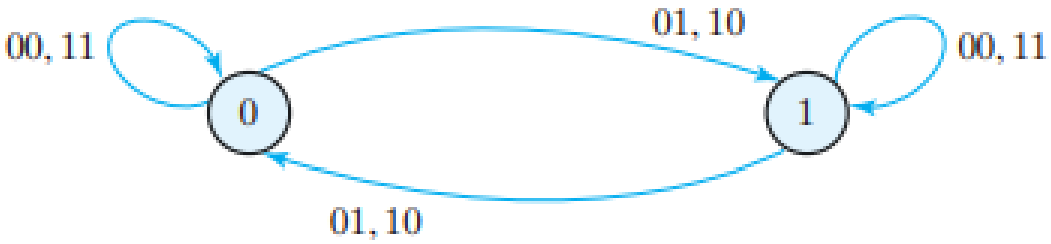
Determine the logic diagram, state table and state diagram of the following circuit.



(a) Circuit diagram

Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

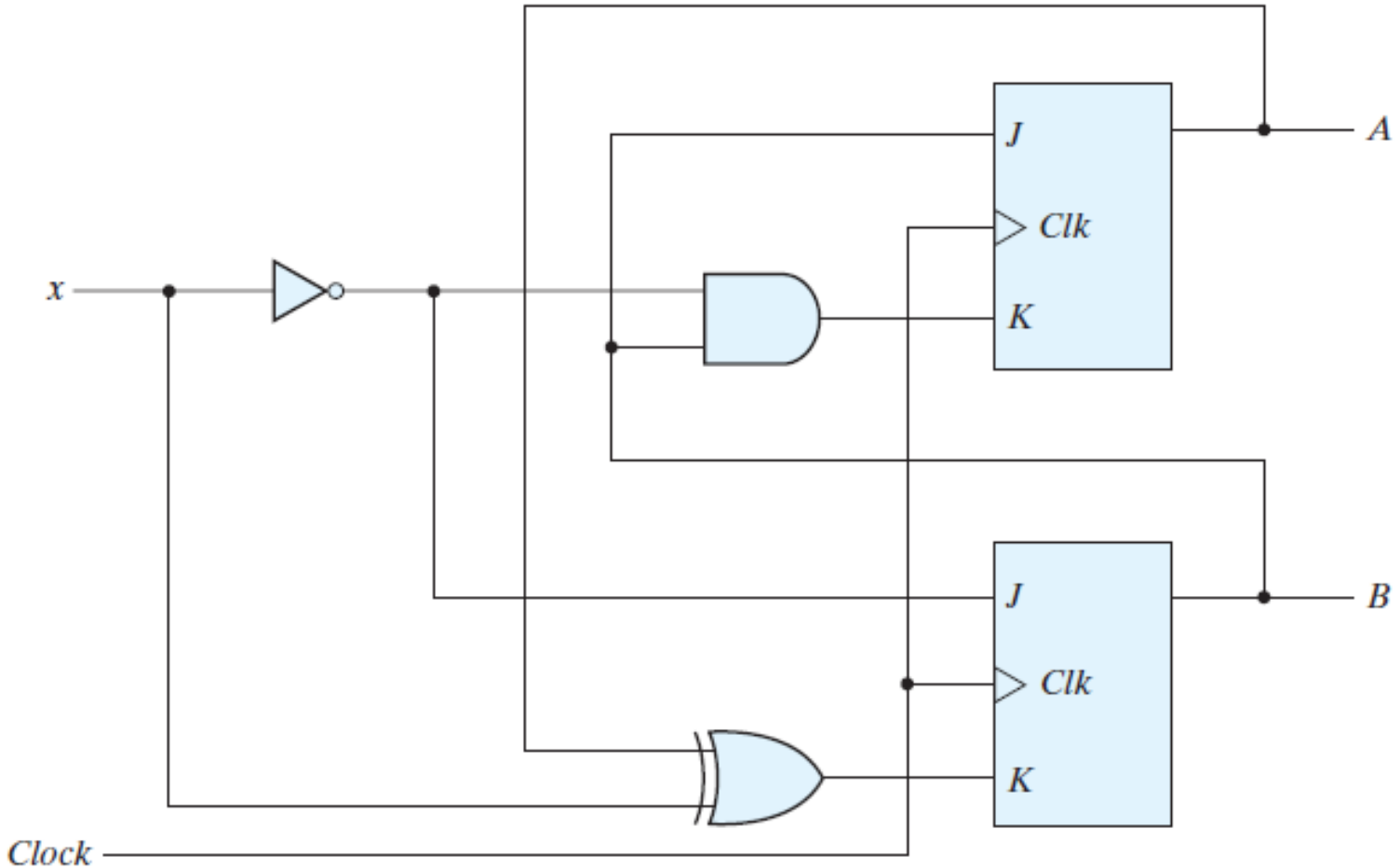
(b) State table



(c) State diagram

Analysis Using J-K flip-flop (FF)

For the given sequential circuit, determine the input equations, state table and state diagram of the following circuit:



Input equations for the circuit:

$$J_A = B \quad K_A = Bx'$$
$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

Analysis Using J-K flip-flop (FF)

Input equations for the circuit:

$$J_A = B \quad K_A = Bx'$$
$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

Characteristic Equations:

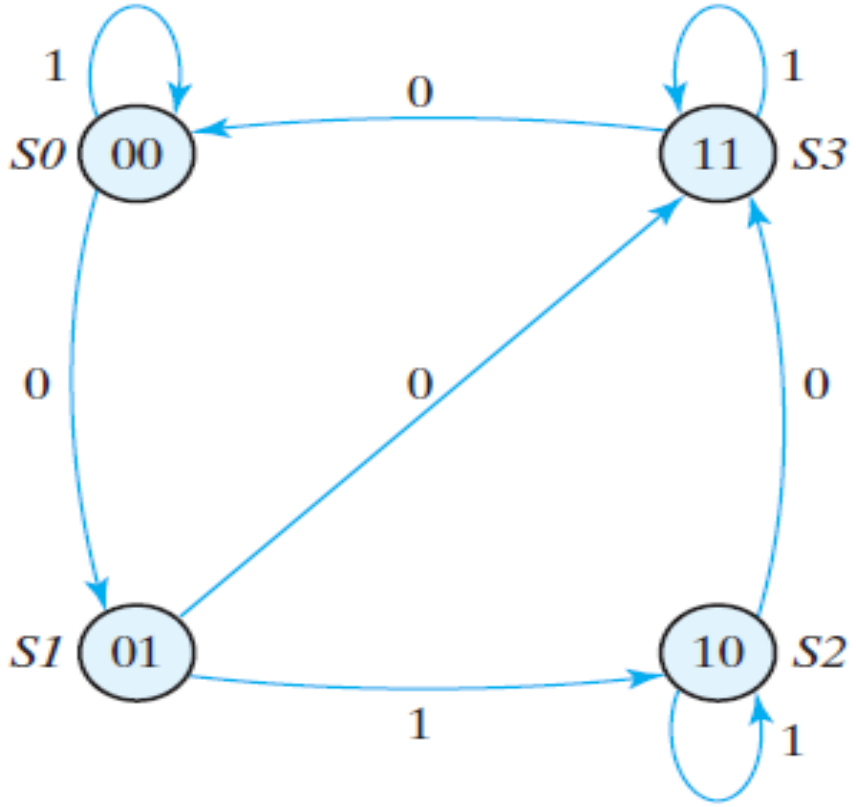
$$A(t+1) = JA' + K'A$$
$$B(t+1) = JB' + K'B$$

By replacing input equations into characteristic equations, we obtain state equations:

$$A(t+1) = BA' + (Bx')'A = A'B + AB' + Ax$$
$$B(t+1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$$

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

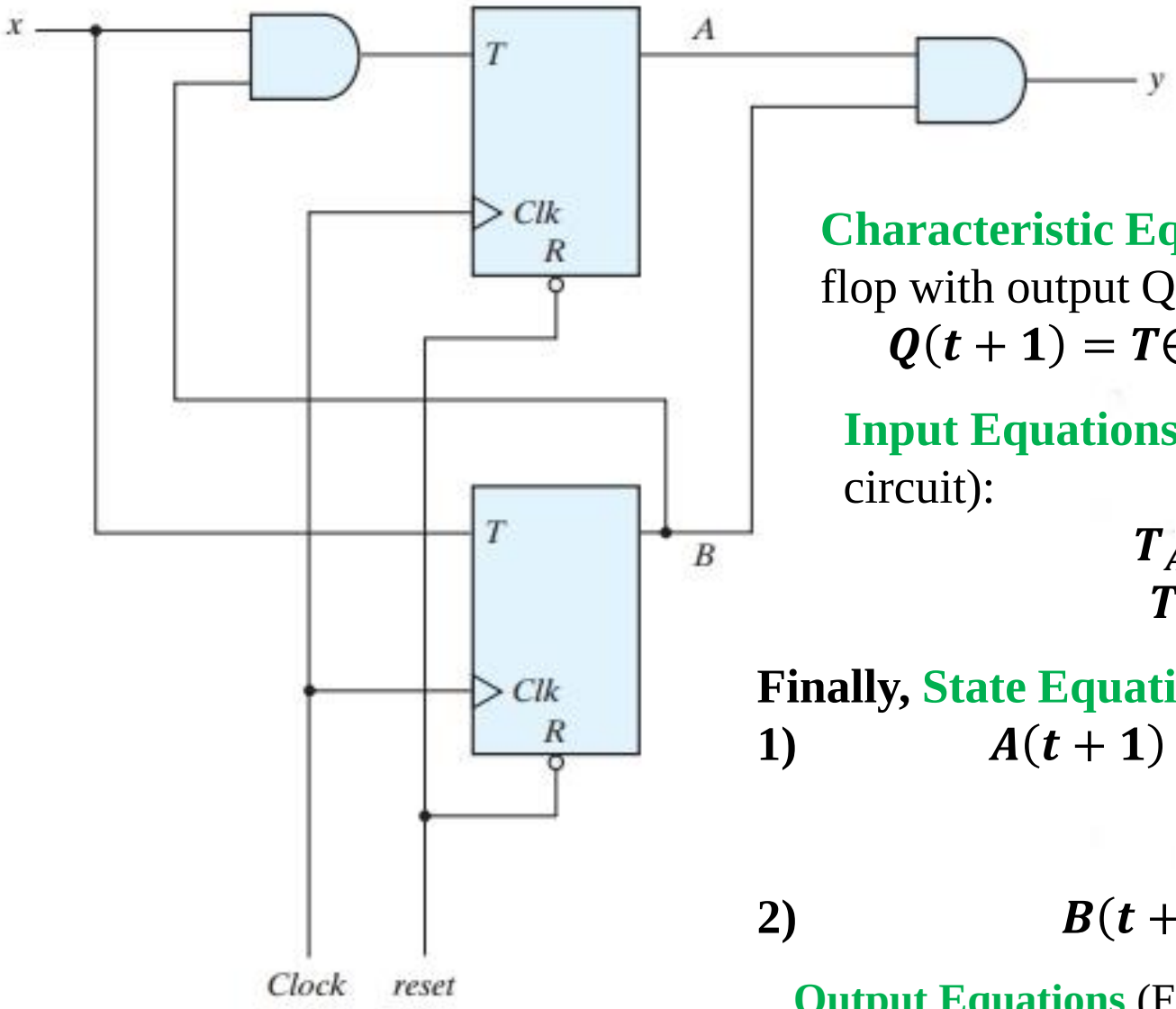
Present State		Input	Next State	
A	B	x	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1



State Table and State Diagram for the J-K FF Circuit on Slide 22

Analysis Using T flip-flop (FF)

For the given sequential circuit (2-bit Binary counter), determine the input equations, state table and state diagram of the following circuit:



Characteristic Equation (For a *T* flip-flop with output Q):

$$\boldsymbol{Q(t+1) = T \oplus Q = T'Q + TQ'}$$

Input Equations (For the given circuit):

$$\boldsymbol{T_A = Bx}$$
$$\boldsymbol{T_B = x}$$

Finally, State Equation (For 2 *T* flip-flops):

1) $$\boldsymbol{A(t+1) = (Bx)'A + (Bx)A'}$$
$$\boldsymbol{= AB' + Ax' + A'Bx}$$

2) $$\boldsymbol{B(t+1) = x \oplus B}$$

Output Equations (For the given circuit):

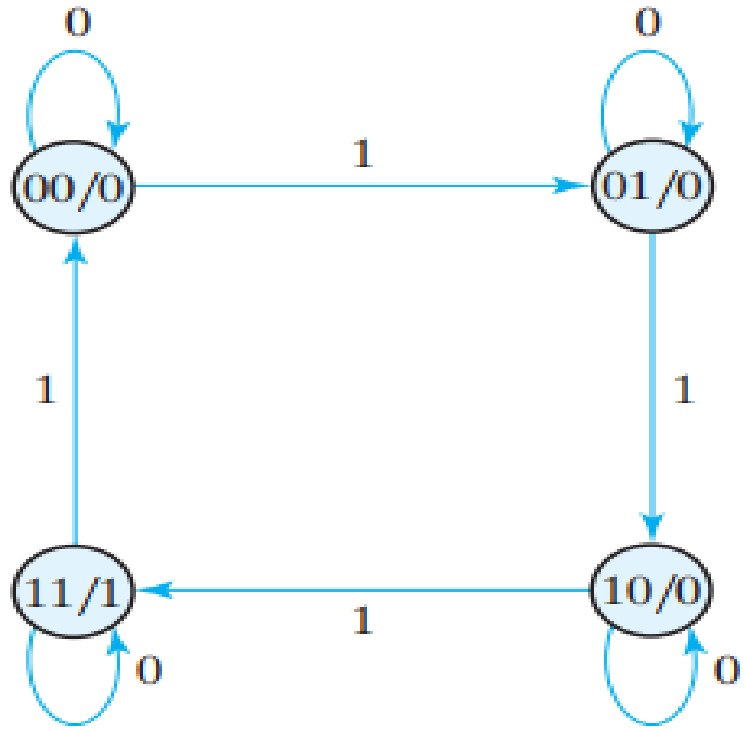
$$\boldsymbol{y = AB}$$

Analysis Using T flip-flop (FF): State table & State diagram

State Table for Sequential Circuit with T Flip-Flops

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	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	1	1	1
1	1	1	0	0	1

State Diagram:



Designing Synchronous Sequential Circuit

- Design procedures or methodologies specify hardware that will implement a desired behavior.

Design Procedure/Steps

- From the word description and specifications of the desired operation, derive a state diagram for the circuit.
- Assign binary values to the states.
- Obtain the binary-coded state table.
- Choose the type of flip-flops to be used.
- Derive the simplified flip-flop input equations and output equations.
- Draw the logic diagram.

Excitation Tables

- The design of a sequential circuit with flip-flops other than the *D* type is complicated by the fact that the input equations for the circuit must be derived indirectly from the state table.
- During the design process, we usually know the transition from the present state to the next state and wish to find the flip-flop input conditions that will cause the required transition. For this reason, we need a table that lists the required inputs for a given change of state. Such a table is called an *excitation table*.
- Characteristic tables are used in analysis.
- *Excitation tables* are used in design.

Excitation Tables

Flip-Flop Excitation Tables

Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

JK Flip-flop

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

T Flip-flop

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

D Flip-flop

Here, **Q^+ =Q(t+1), that is, output of a flip-flop at next state**

Design of a Sequence Detector Using *D* Flip-Flop

- Suppose we wish to design a circuit that detects a sequence of three or more consecutive 1's in a string of bits coming through an input line (i.e., the input is a *serial bit stream*). The state diagram for this type of circuit is shown below:

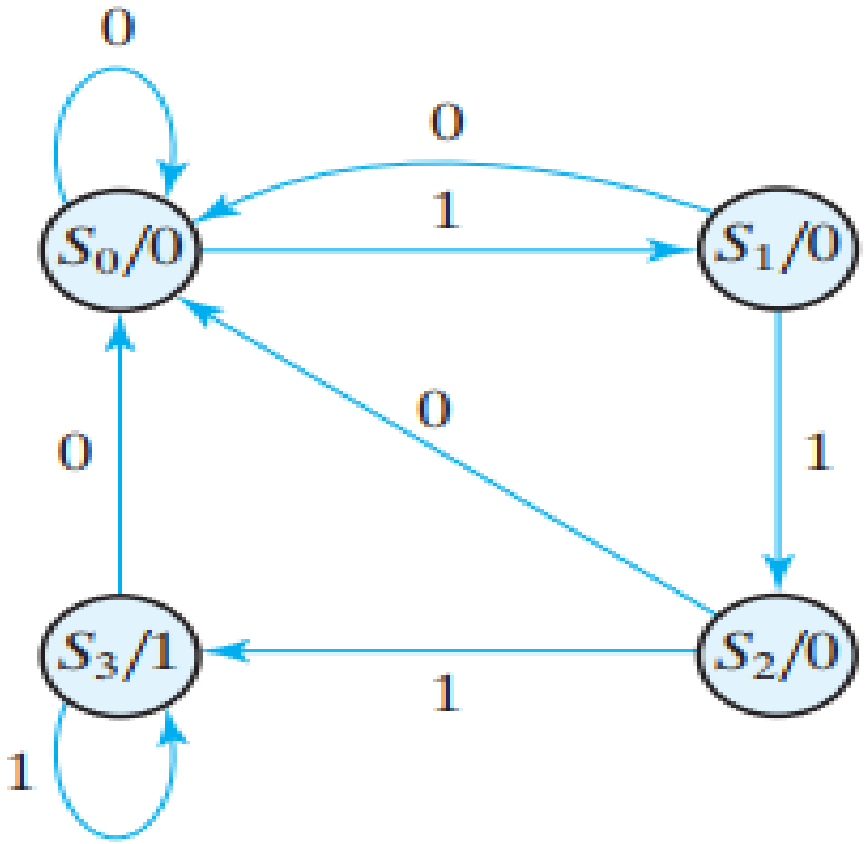


Fig.: Sequence Detector

Here,

S_0 represents state **00**

S_1 represents state **01**

S_2 represents state **10**

S_3 represents state **11**

Design/Synthesis of a Sequence Detector Using *D* flip-flop

State Table for Sequence Detector

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*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

Flip Flop Inputs

D_A	D_B
0	0
0	1
0	0
1	0
0	0
1	1
0	0
1	1

Sum of minterms for Inputs & Output variables:

$$D_A(A, B, x) = \Sigma(3, 5, 7)$$

$$D_B(A, B, x) = \Sigma(1, 5, 7)$$

$$y(A, B, x) = \Sigma(6, 7)$$

Excitation Table

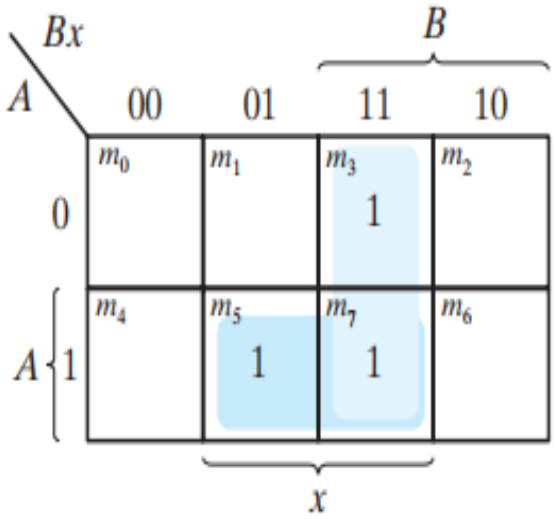
D Flip-flop

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

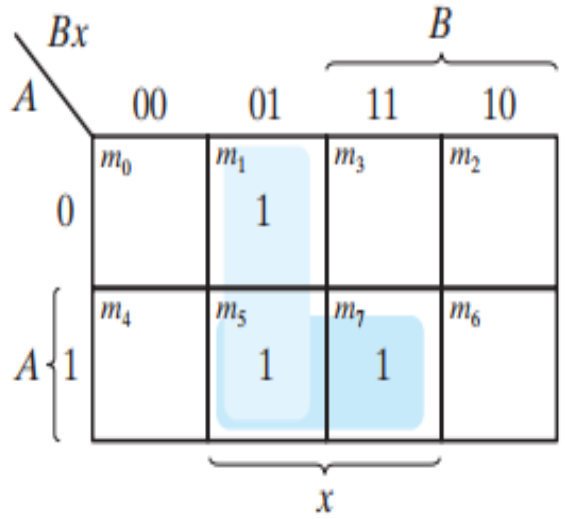
* The advantage of designing with *D* flip-flops is that the Boolean equations describing the inputs to the flip-flops can be obtained directly from the state table.

Design of a Sequence Detector Using *D* flip-flop

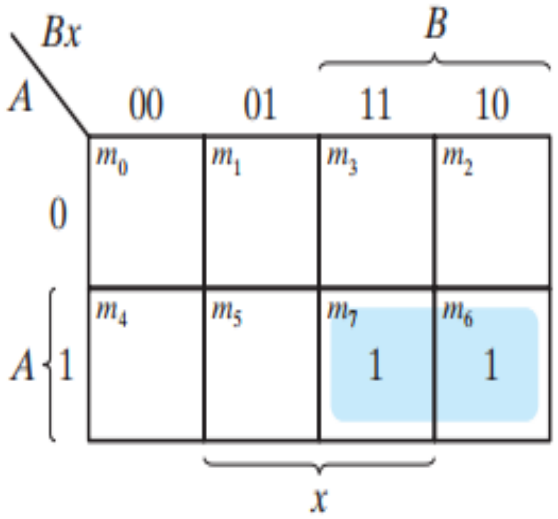
Using K-map to simplify,



$D_A = Ax + Bx$



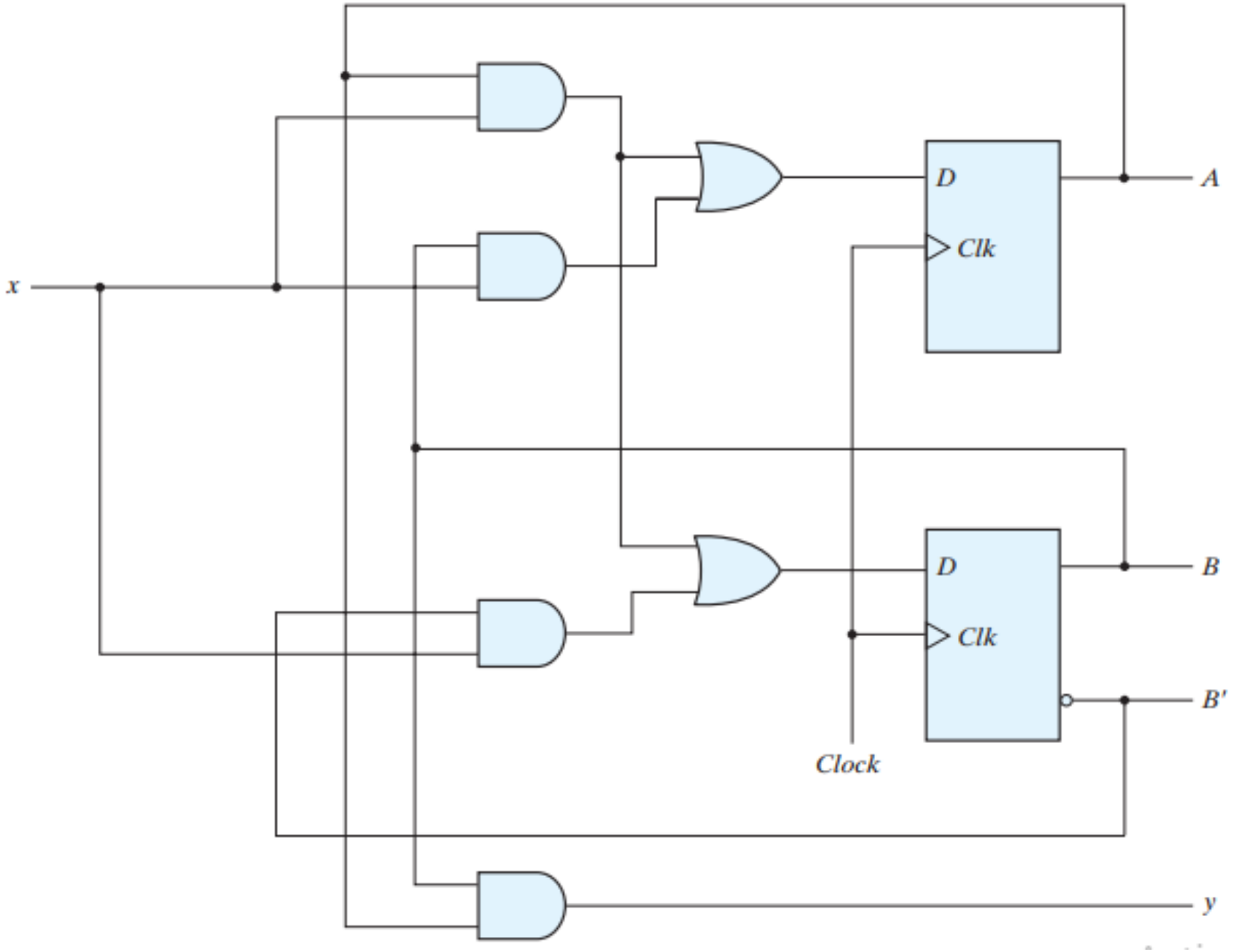
$D_B = Ax + B'x$



$y = AB$

$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = AB$$

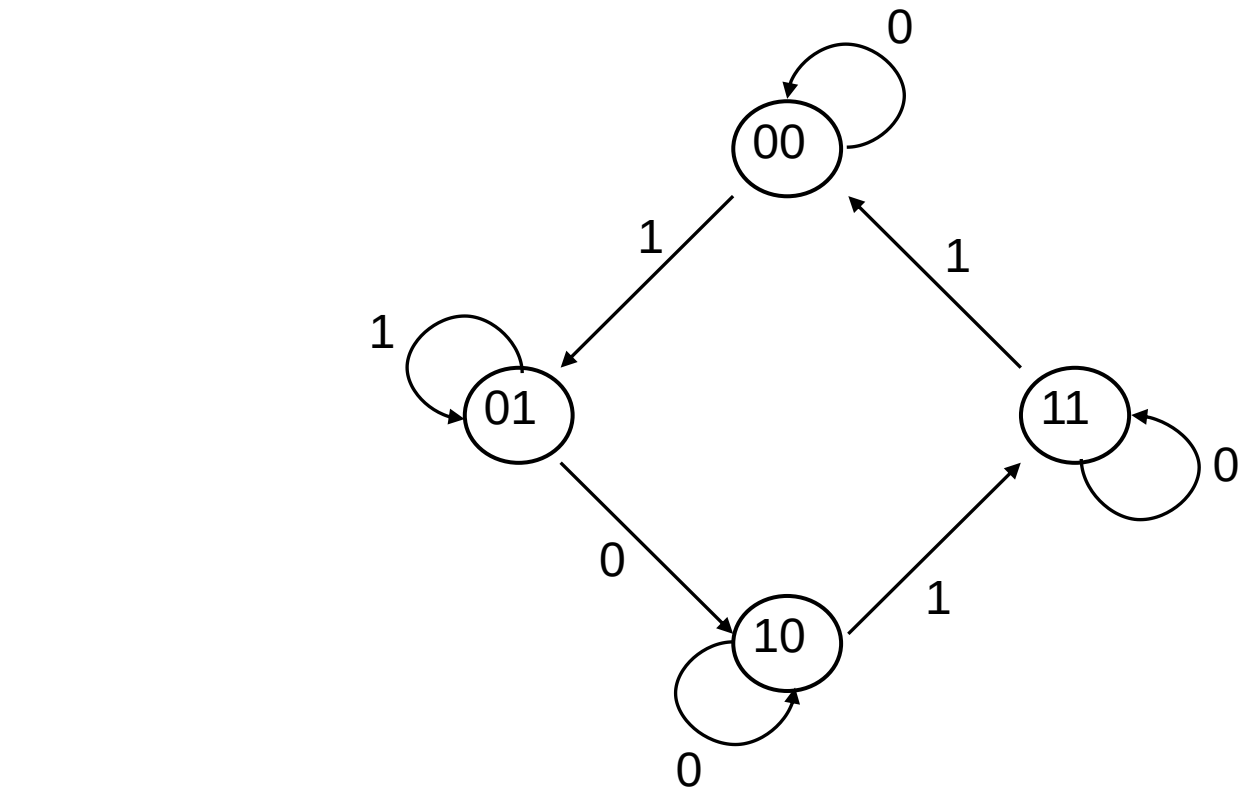
Design of a Sequence Detector Using *D* flip-flop

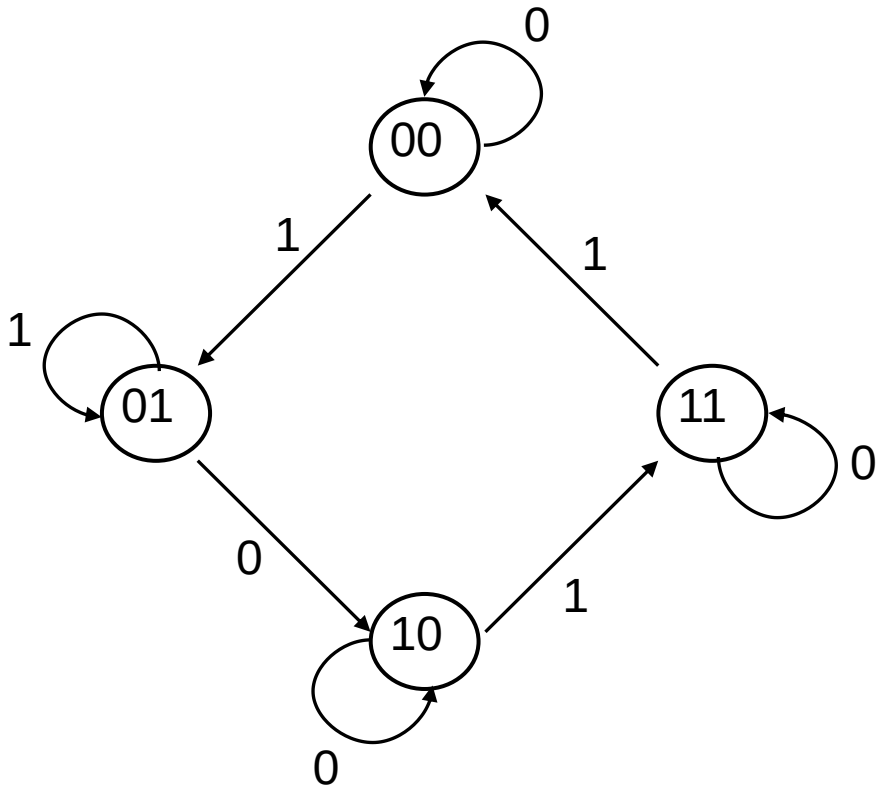


Circuit Diagram of Sequence Detector Using a *D* Flip-flop

Design a Sequential Circuit Using *JK* flip-flop

- Given the following state diagram, design the sequential circuit using JK flip-flops.





Excitation Table

Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

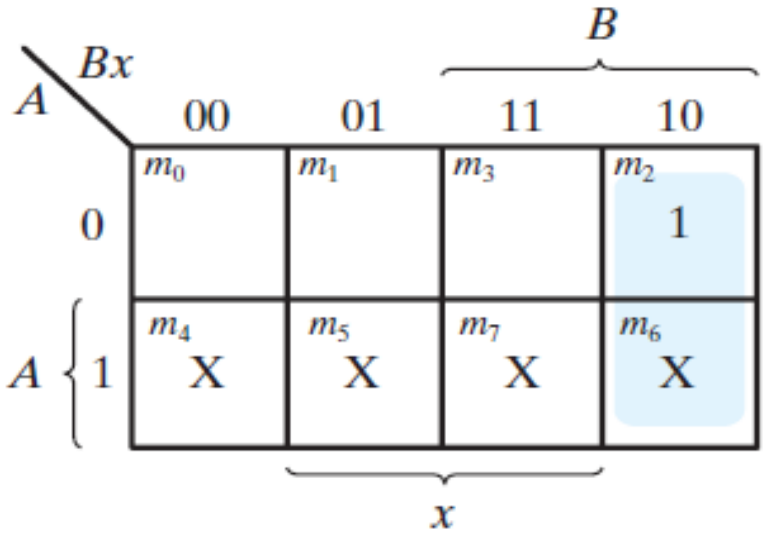
JK Flip-flop

State Table and JK Flip-Flop Inputs

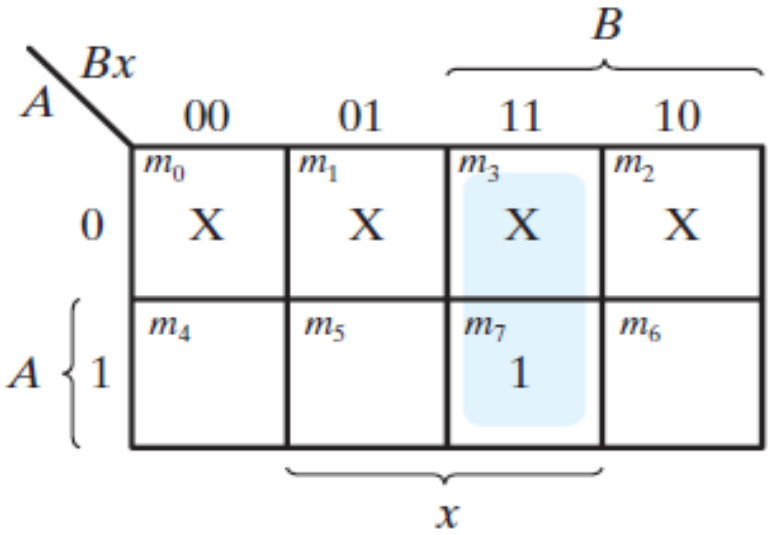
Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Design Sequential Circuit Using *JK* flip-flop: Deriving State Table

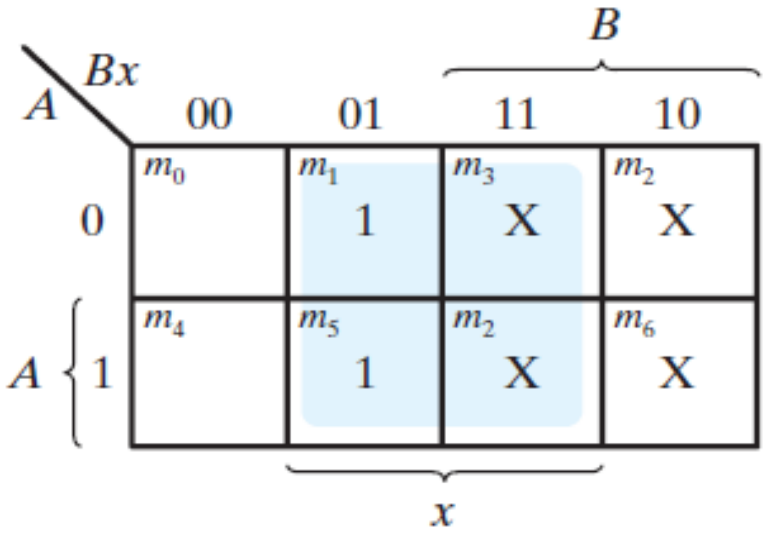
Design/Synthesis Using a *JK* flip-flop: Applying K-map



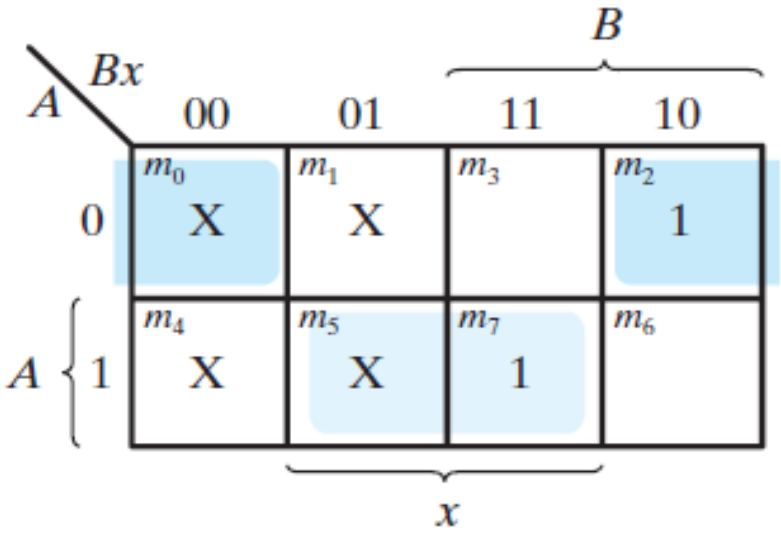
$J_A = Bx'$



$K_A = Bx$

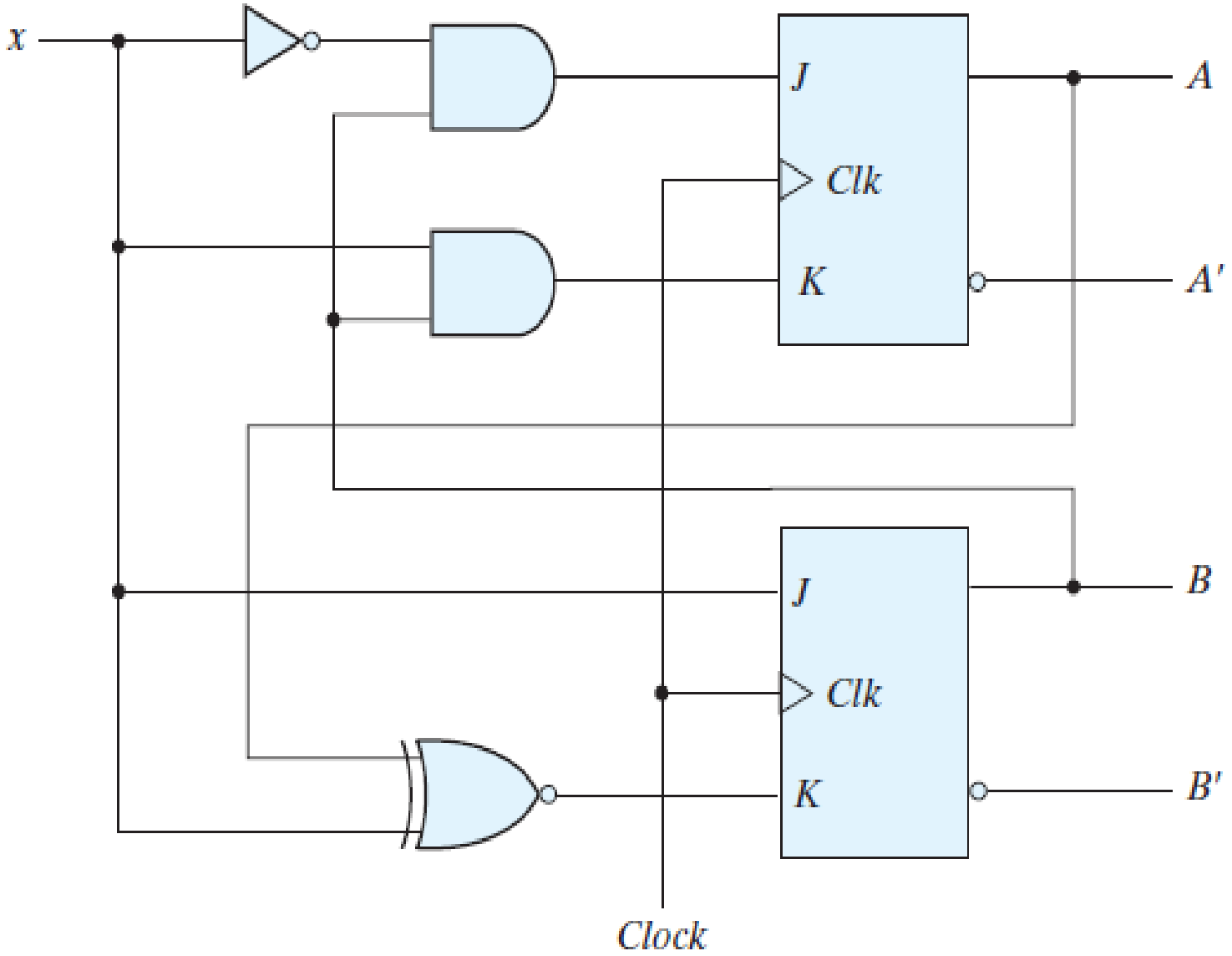


$J_B = x$



$K_B = (A \oplus x)'$

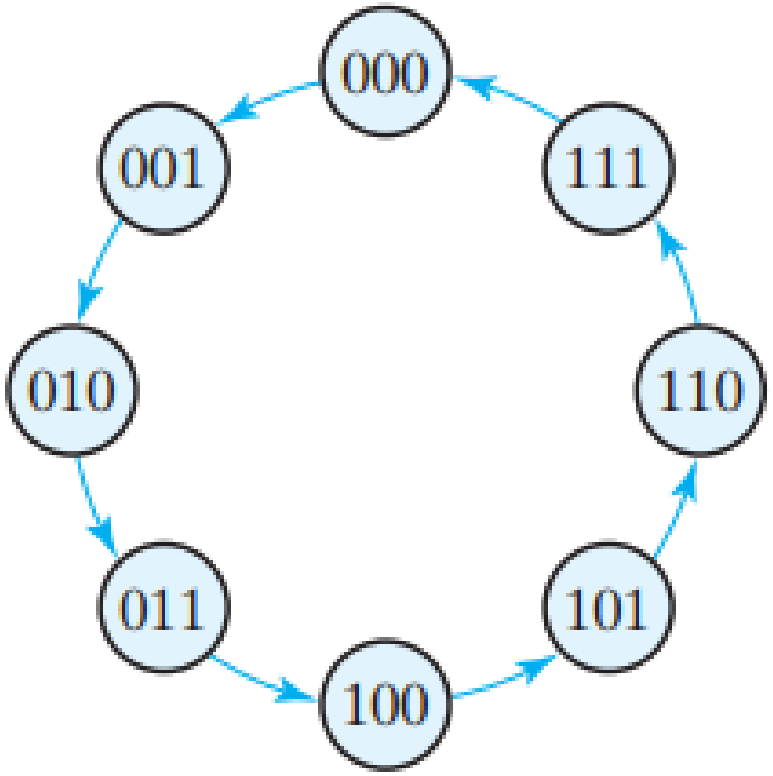
Design/Synthesis Using a *JK* flip-flop: Implementing Logic Circuit



Logic diagram for sequential circuit with *JK* flip-flops

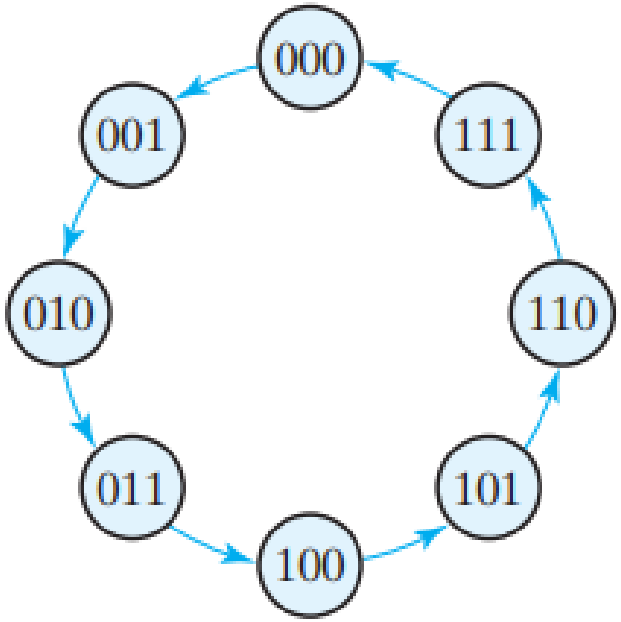
Design a Binary Counter Using a *T* flip-flop

Design Statement: Design (state diagram, state table, K-map, logic diagram) a 3-bit binary counter using *T* flip-flop.



State diagram of 3-bit binary counter

Design a Binary Counter Using a *T* flip-flop: State Table

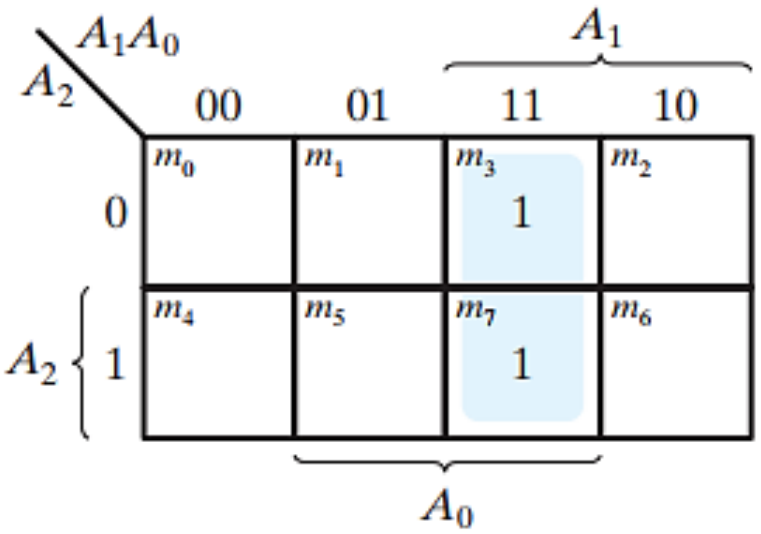


State Diagram

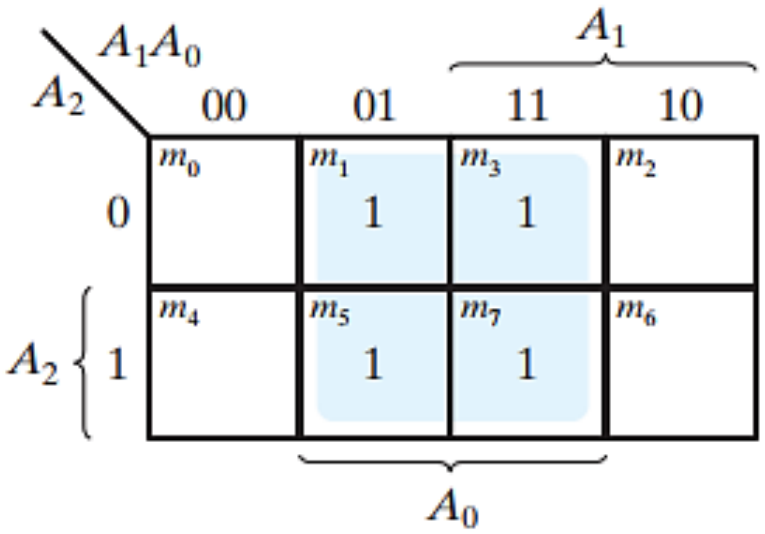
State Table

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

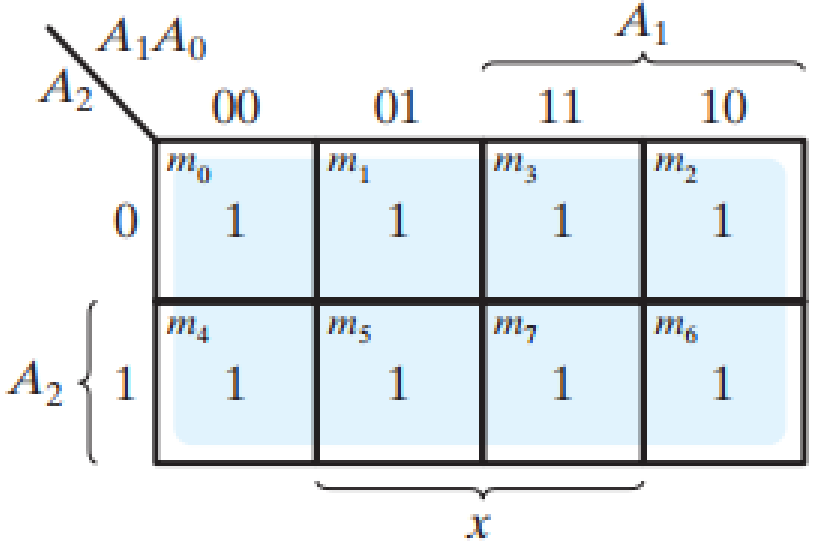
Design a Binary Counter Using a *T* flip-flop: K-map



$$T_{A2} = A_1A_0$$

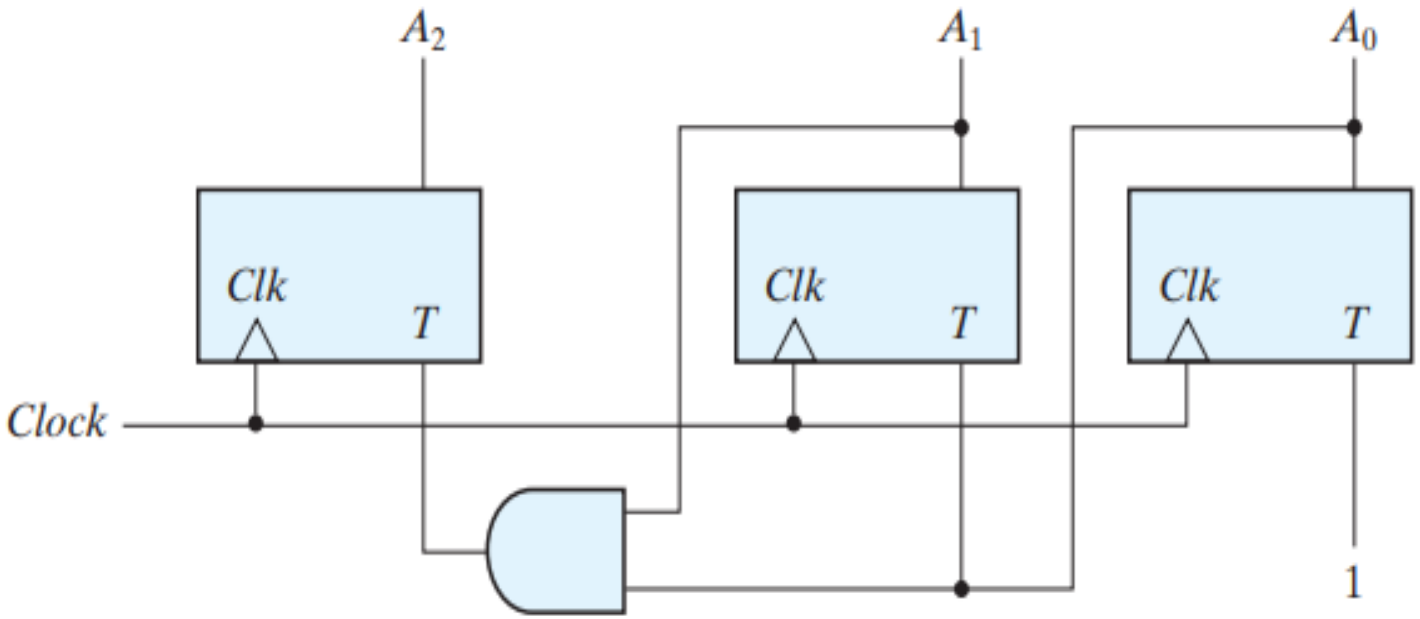


$$T_{A1} = A_0$$



$$T_{A0} = 1$$

Design a Binary Counter Using a *T* flip-flop: Logic Diagram



Logic diagram of three-bit binary counter using *T* flip-flop