LECTURE 1

Intro

RECAP

- Remember the combinational circuit from first year!

❑ Combinational circuit

- Boolean Algebra
- Min-term / Max-term / Standard form
- K-Map simplification
- Basic gates
- Universal gates
- Full adder
- Decoder
- Encoder
- Mux

SEQUENTIAL CIRCUIT

- Remember the combinational circuit from first year!

❑Combinational circuit

  - Output depends only on immediate input
  - No memory

❑Sequential circuit

  - Output depends on current input and past history
  - Storage element
  - Have state

SEQUENTIAL CIRCUIT

- Remember the combinational circuit from first year!

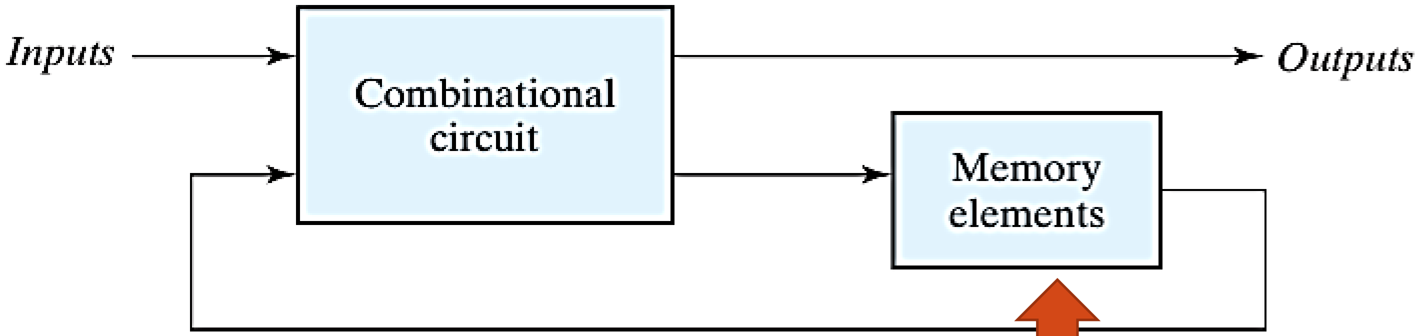


FIGURE 5.1
Block diagram of sequential circuit

Present state

- Sequential circuit is specified by a **time** sequence of **inputs**, **outputs**, and internal **states**

- **State/Storage:** time-delay devices that depends on propagation delay of each circuit/gate

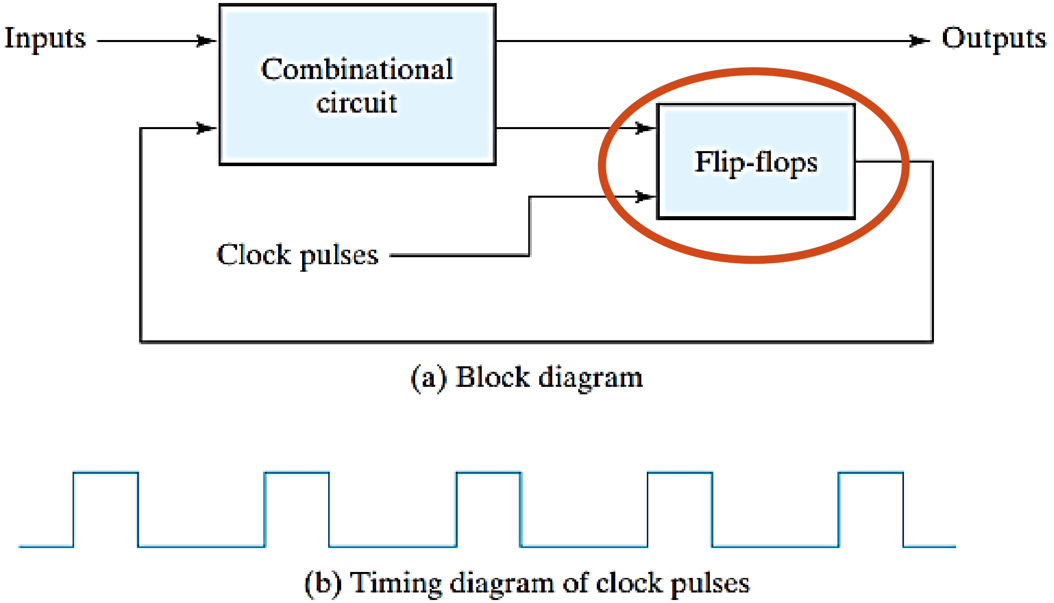
TYPES OF SEQUENTIAL CIRCUIT

- **Synchronous sequential circuit**
  - Behavior can be defined from the knowledge of its signals at discrete instants of time
  - Employs signals that affect the storage elements at only discrete instants of time
  - Synchronization is achieved by clock generator that produces clock pulses

- **Asynchronous sequential circuit**
  - Behavior depends upon the input signals at any instant of time and the order in which the inputs change
  - Mainly composed of combinational circuit + feedback from some storage element

SYNCHRONOUS SEQUENTIAL CIRCUIT

- **aka clocked sequential circuits**
  - clock pulses determines
    - **When** computational activity occurs within the circuit
    - **What** changes will take place in storage elements & outputs



(a) Block diagram

(b) Timing diagram of clock pulses

FIGURE 5.2
Synchronous clocked sequential circuit

STORAGE ELEMENTS

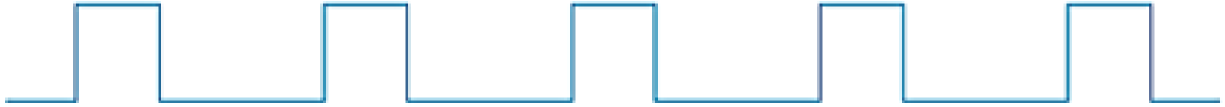
- A storage element in a digital circuit can maintain a binary state indefinitely
  - as long as power is delivered to the circuit

❑ **Latches**

- Storage elements that operate with signal levels (0 / 1. See figure)
- Level sensitive devices
- Not practical for use as storage element in synchronous sequential circuits

❑ **Flip-flops**

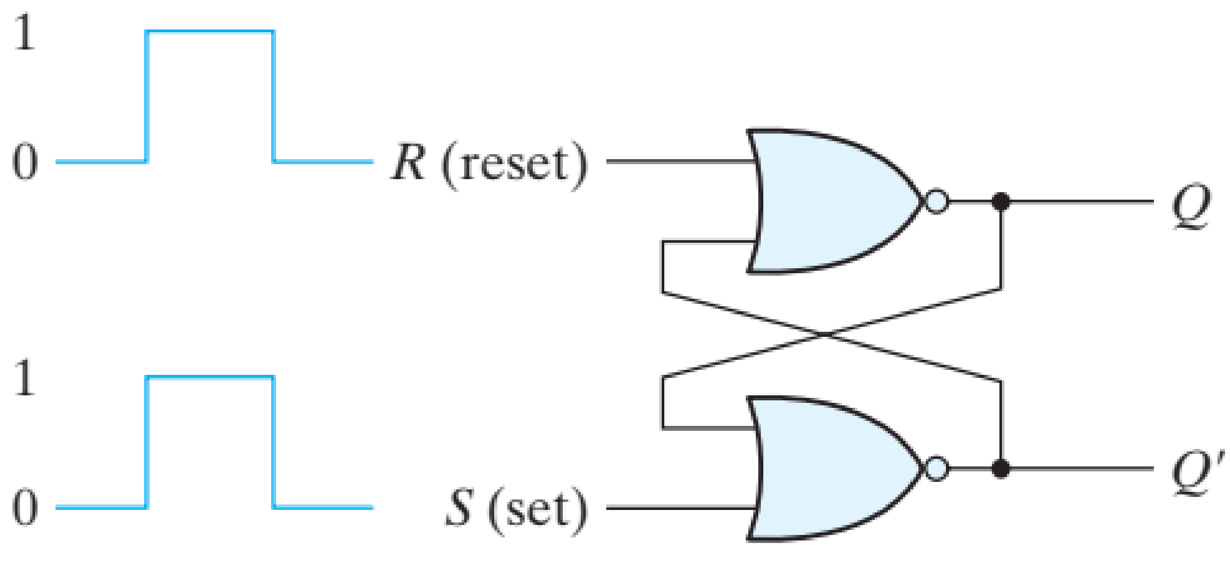
- Storage elements that operate with signal transition (from 1 to 0, and vice verse)
- Edge sensitive devices



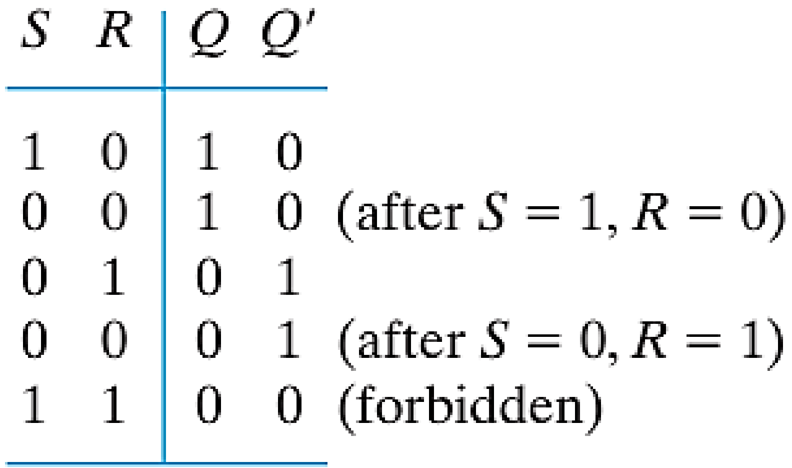
(b) Timing diagram of clock pulses

SR LATCH

- Set/Reset latch
- Two cross-coupled **NOR** gate
- Sensitive to level 1
- States
  - Set state: Q=1, Q`=0
  - Reset state: Q=0, Q`=1
- Normally, Q & Q` are complementary of each other
- Forbidden states S=1, R=1
- What happens if SR=00 after SR=11 🤔
  - Undefined state (meta-stable state)



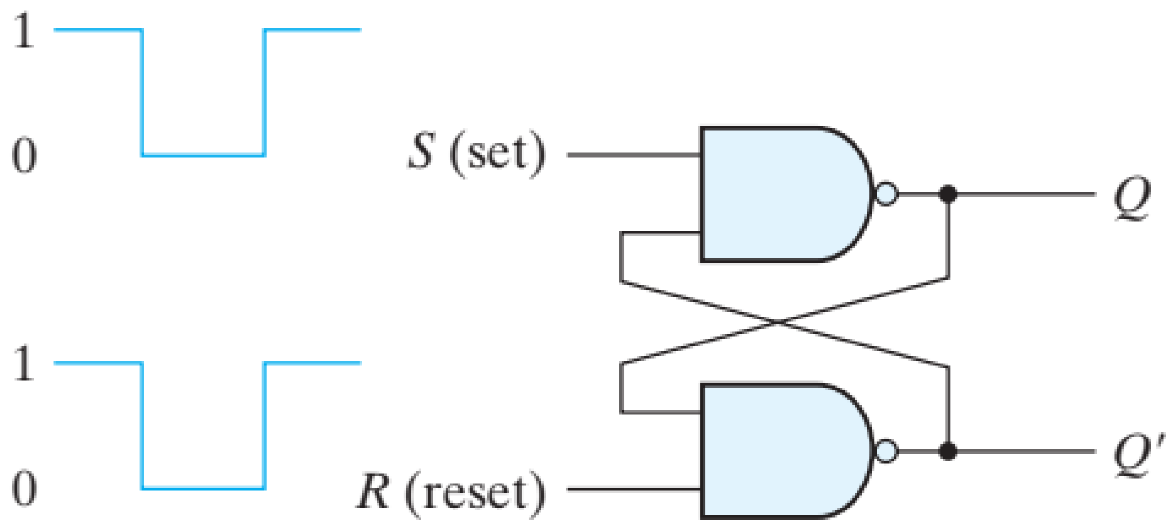
(a) Logic diagram



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)

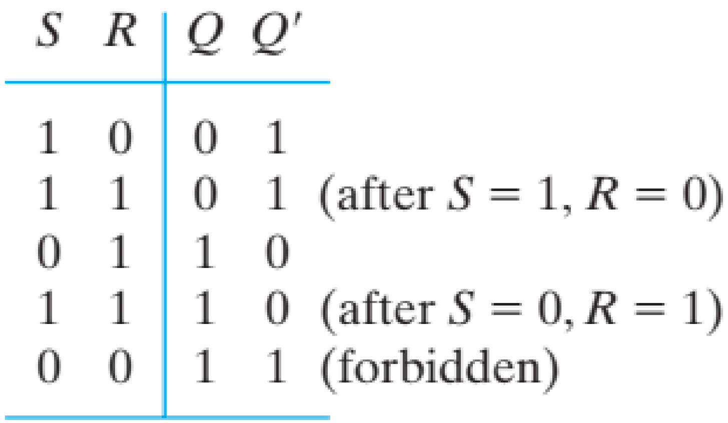
S`R` LATCH

- Set/Reset latch
- Two cross-coupled **NAND** gate
- Sensitive to level 0
- States
  - Set state: Q=1, Q`=0
  - Reset state: Q=0, Q`=1



(a) Logic diagram

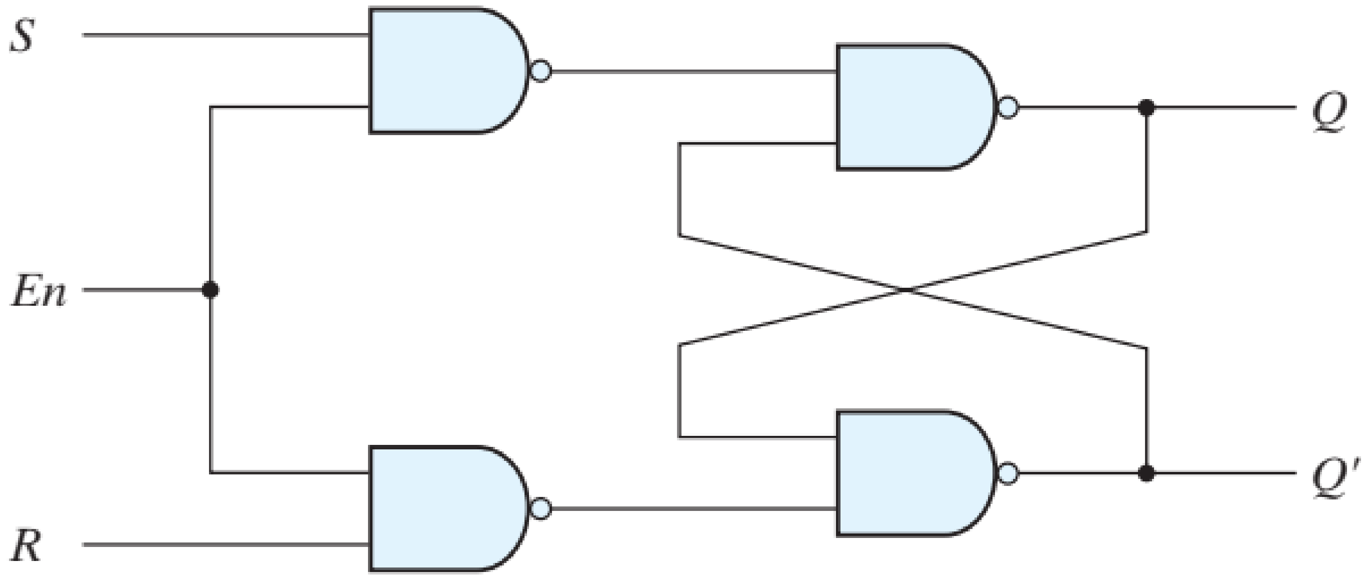
- Normally, Q & Q` are complementary of each other
- Forbidden states S=1, R=1
- What happens if SR=11 after SR=00 🤔
  - Undefined state (meta-stable state)

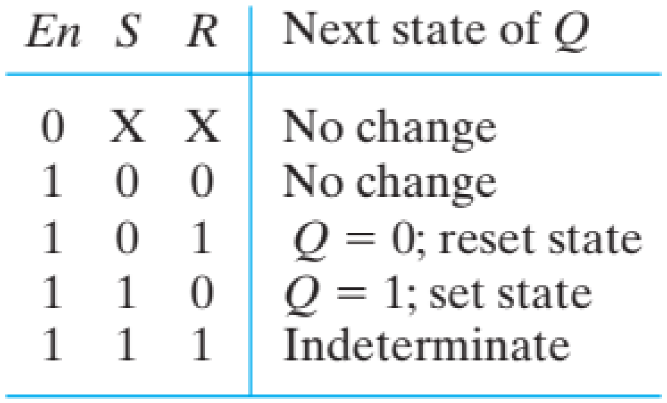


S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

SR LATCH WITH ENABLE SIGNAL

- Same operation of SR latch, but with enable signal
- The circuit is active when E=1
- Two additional NAND gates are added
- Set state occurs when S=1 => Q=1

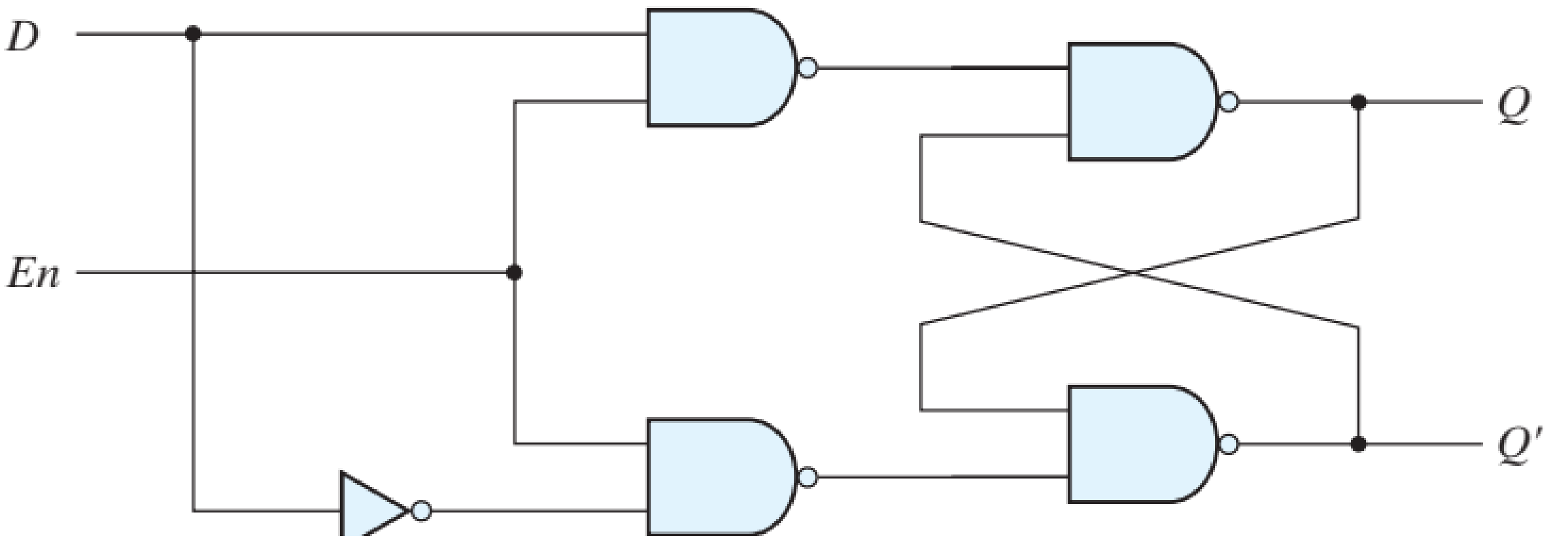




En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

D LATCH (TRANSPARENT LATCH)

- Eliminate the undefined state for SR latch
- Just 2 inputs, **D** (data), and **E** (enable)
- Output Q follows input D when E is enabled (Data transparency)
- Information is retained when E is disabled



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

SUMMARY OF SR LATCHES

- Latches are designated by a rectangular block
- Bubbles at the output represents the complement of the output
- Bubbles at the input represents logic level activation (logic 0)

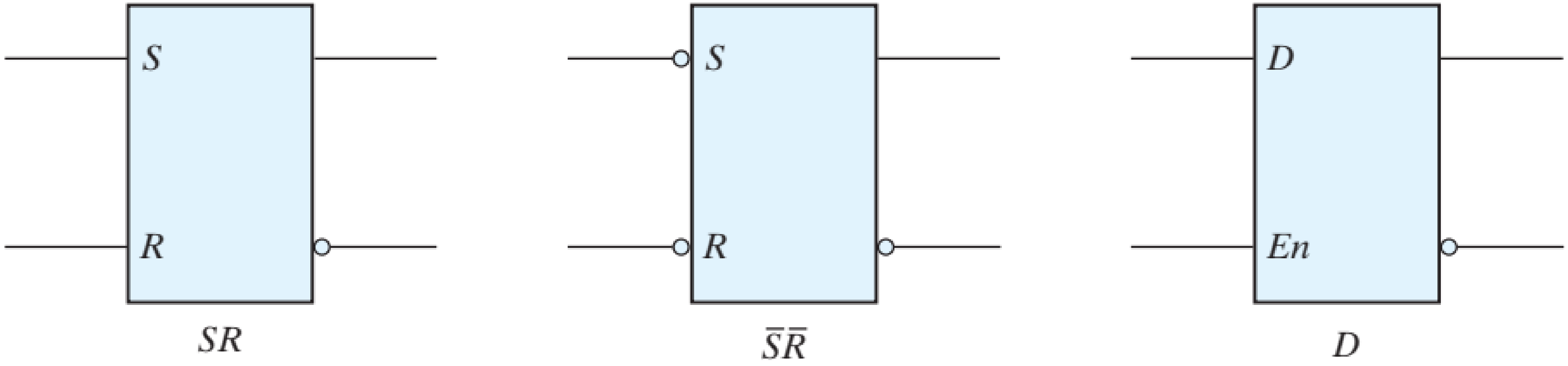


FIGURE 5.7

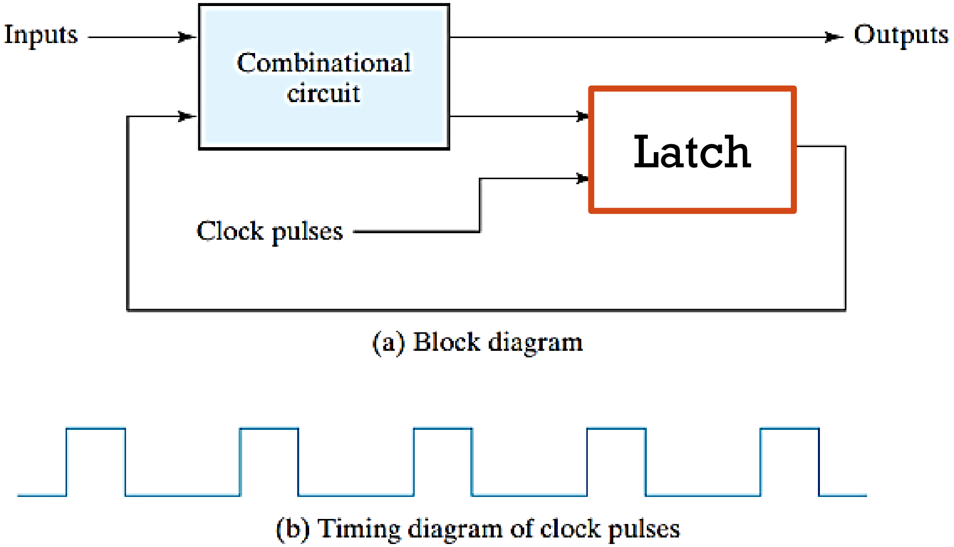
Graphic symbols for latches

LECTURE 2

Flip-Flops

FLIP-FLOP

- Latches can’t be used in synchronous sequential circuits
  - Because it’s level-based triggered (level 1, level 0)
  - Clock level value stay longer, which result in multiple changes in the output of the latch



(a) Block diagram

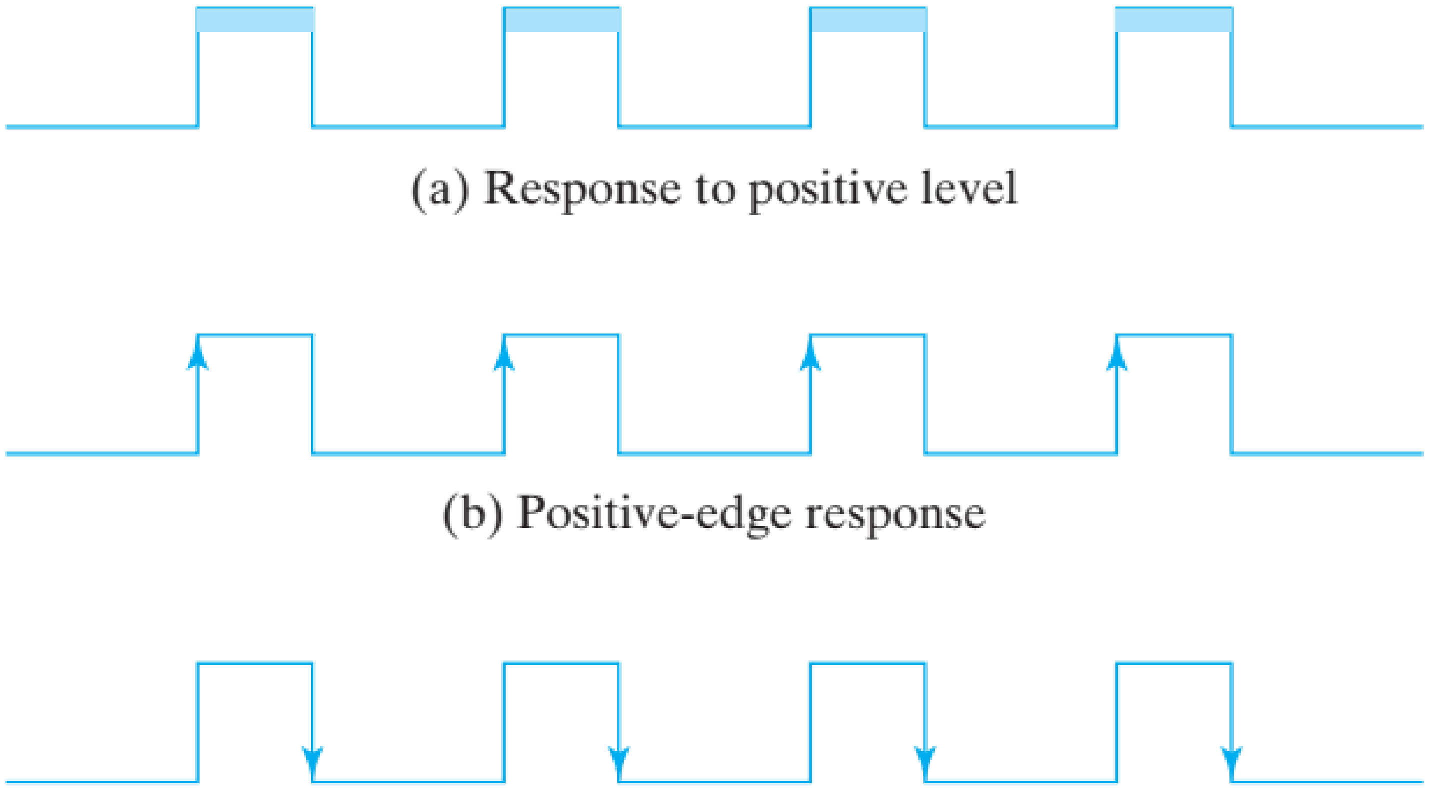
(b) Timing diagram of clock pulses

FIGURE 5.2
Synchronous clocked sequential circuit

- Here comes the flip-flop

FLIP-FLOP

- Flip-flop is edge triggered
  - Positive/negative edge



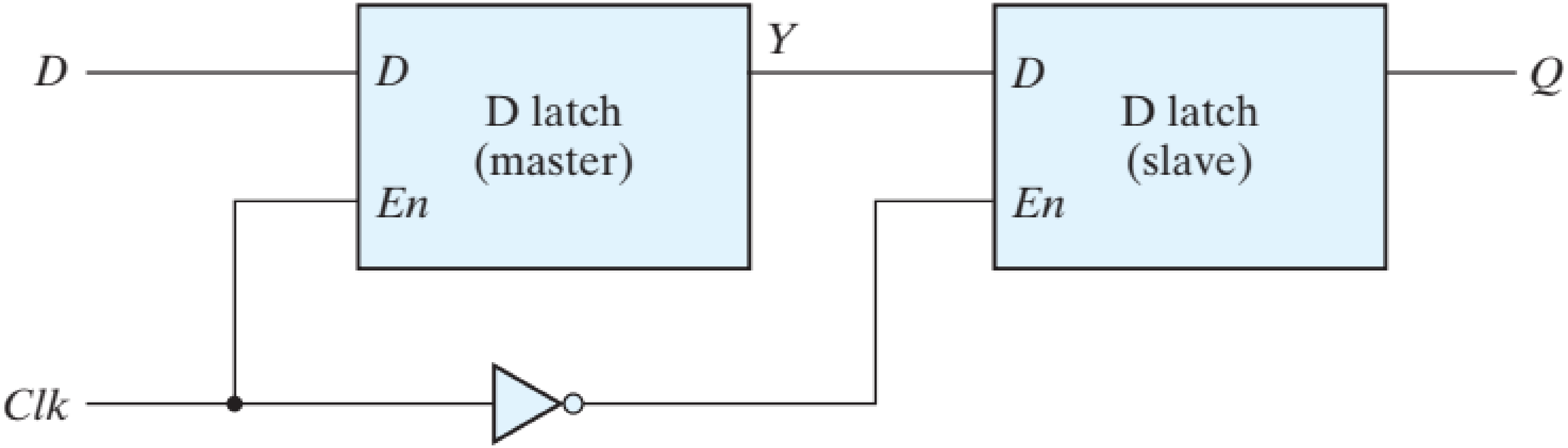
(a) Response to positive level

(b) Positive-edge response

(c) Negative-edge response

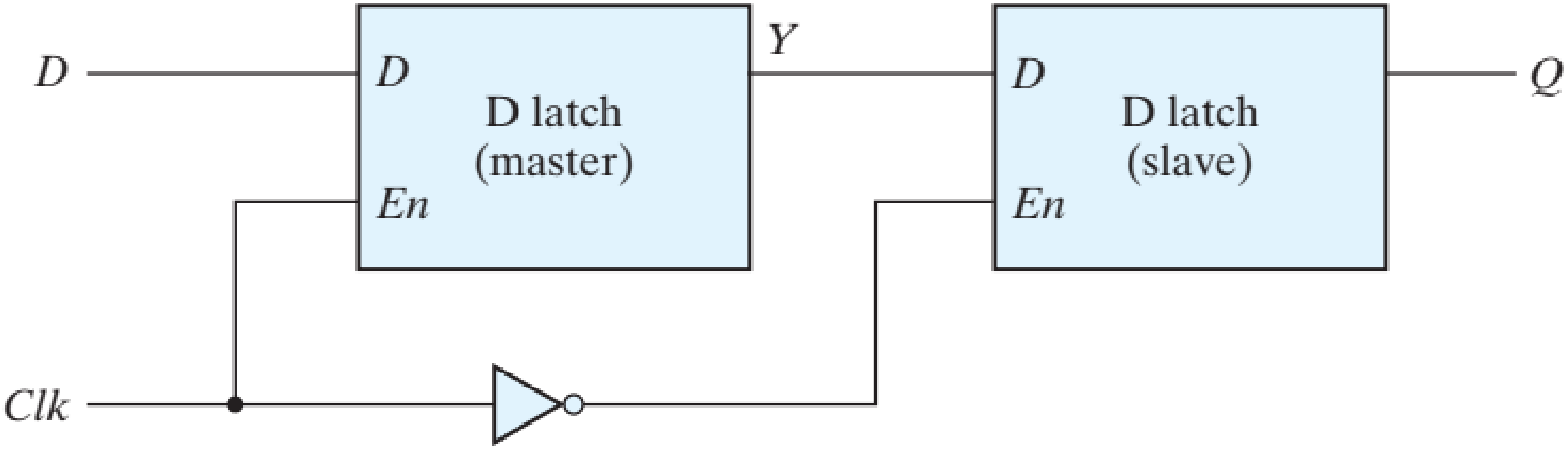
EDGE-TRIGGERED D FLIP-FLOP

- The figure shows NEGATIVE edge D flip flop
  - Master / slave D latch with on input D and one output Q and clock generator
  - Input is sampled at positive edge of the ***clock `clk`***
  - Output is changed at ***negative*** edge of the clock `***clk***`
  - When $clk = 0$, slave is enabled. Changes in master are isolated from slave
  - When $clk = 1$, master is enabled. slave remains fixed and isolated from master
  - Change in the output of the flip-flop can be triggered only by and during the transition of the clock from 1 to 0



EDGE-TRIGGERED D FLIP-FLOP

1. The output may change only once
2. A change in the output is triggered by the negative edge of the clock
3. The change may occur only during the clock's negative level
4. The value that is produced at the output of the flip-flop is the value that was stored in the master stage immediately before the negative edge occurred



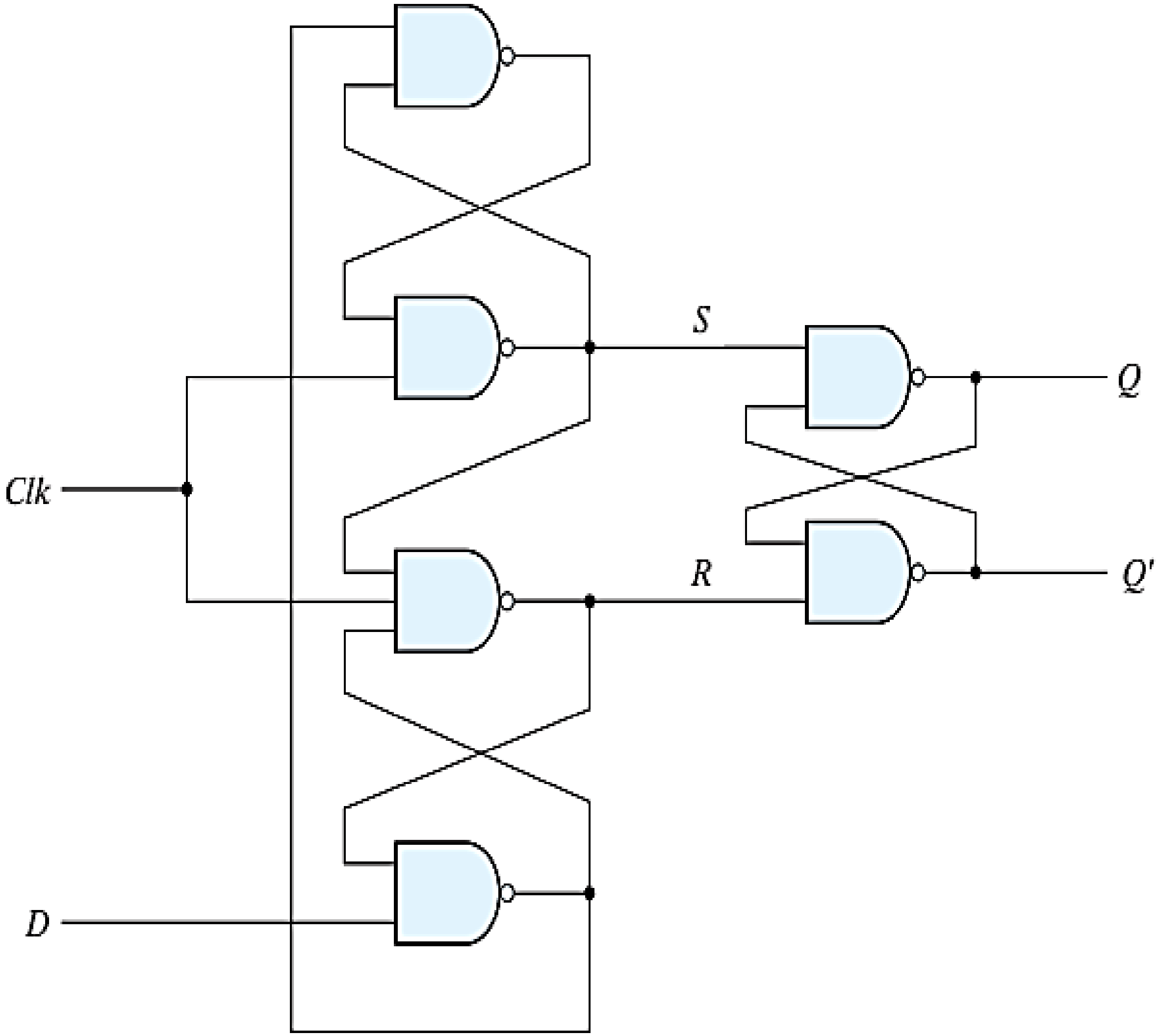
EDGE-TRIGGERED D FLIP-FLOP

- The figure shows POSITIVE edge D flip flop

1. When $clk = 0 \Rightarrow$ SR = 11 $\Rightarrow$ no change
2. When $clk = 1$ & $D = 0 \Rightarrow$ SR = 10 $\Rightarrow$ Q = 0
3. If there is a change in input, output will remain constant because of $Q = 0$

----

1. When $clk = 0 \Rightarrow$ SR = 11 $\Rightarrow$ no change
2. When $clk = 1$ & $D = 1 \Rightarrow$ SR = 01 $\Rightarrow$ Q = 1
3. If there is a change in input, output will remain constant because of $Q` = 0$



❑ Observations

- Input is propagated to output on positive clock edge
- Output remains fixed as long as ***clk*** is active

EDGE-TRIGGERED D FLIP-FLOP

- The timing of the response of a flip-flop to input data and to the clock must be taken into consideration when one is using edge-triggered flip-flops

❑ Definitions

- **Setup time:** The minimum time D input must be maintained at a constant value prior to the occurrence of the clock transition
- **Hold time:** The minimum time D input must not change after the application of the positive transition of the clock
- **Propagation time:** The interval between the trigger edge and the stabilization of the output to a new state

EDGE-TRIGGERED D FLIP-FLOP

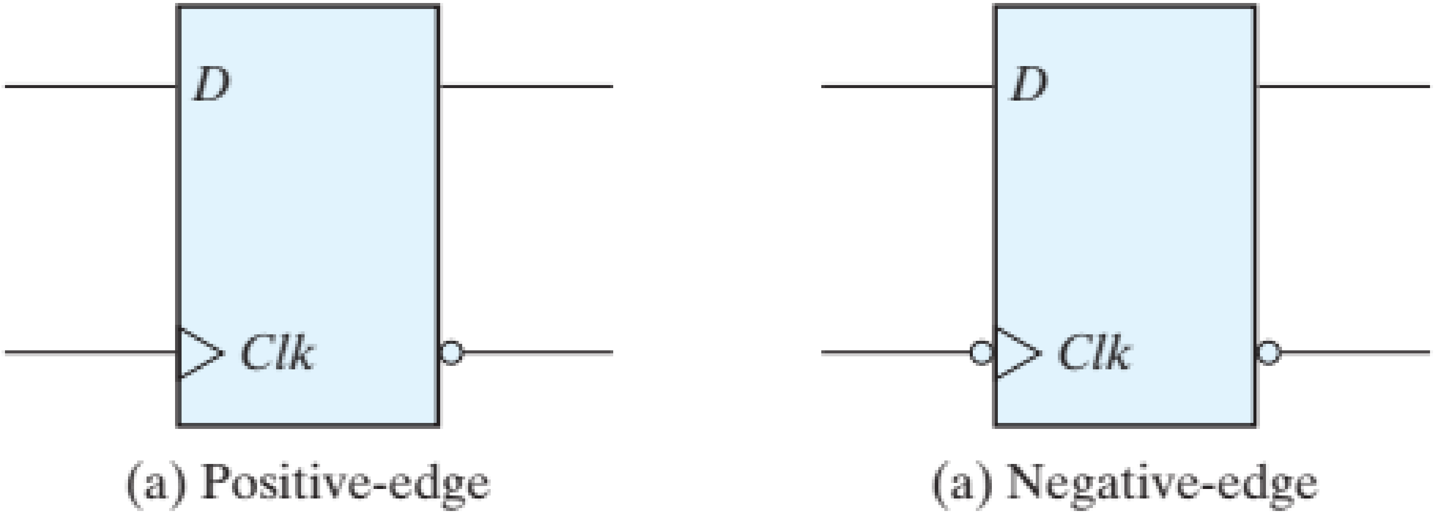


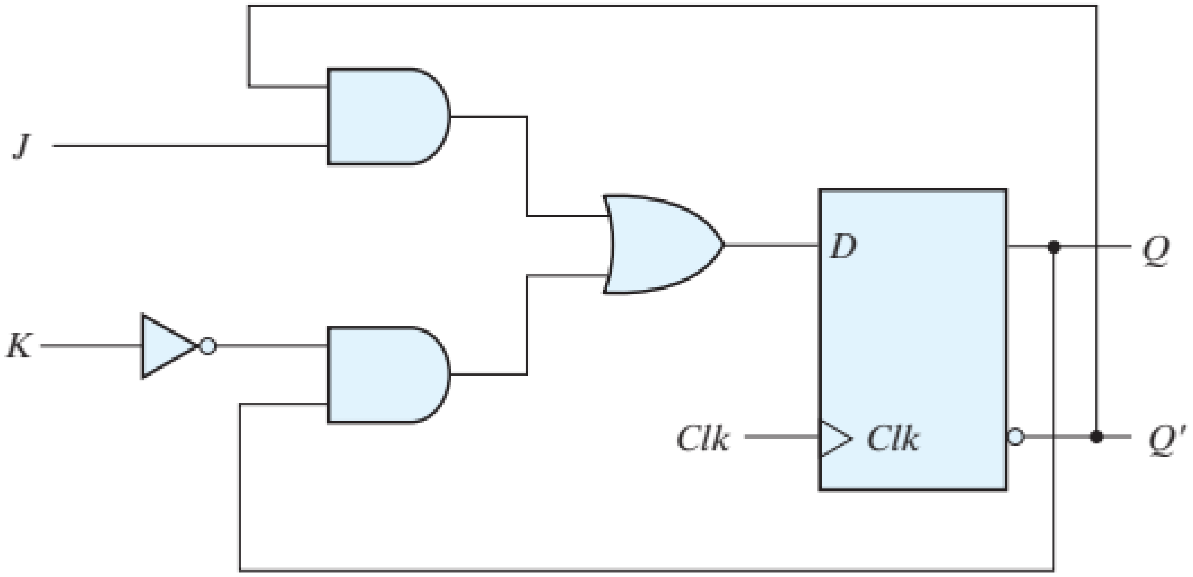
FIGURE 5.11
Graphic symbol for edge-triggered *D* flip-flop

J-K FLIP-FLOP

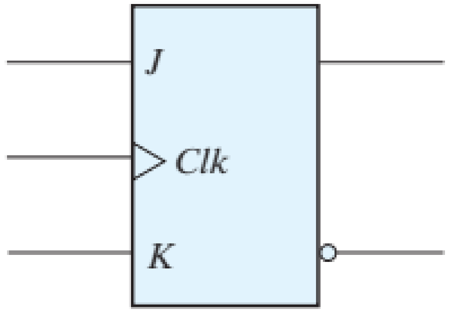
- Another type of flip-flops

Operations

- **Set:** $J = 1$
- **Reset:** $K = 1$
- **Complement output:** $J = 1, K = 1$



(a) Circuit diagram



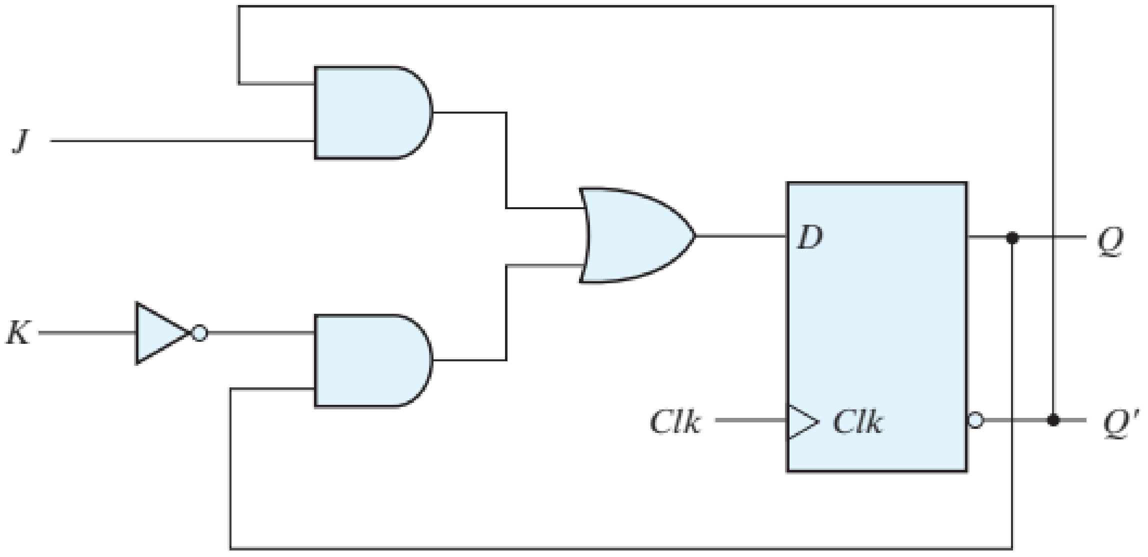
(b) Graphic symbol

J-K FLIP-FLOP

- Equations

$$D = JQ' + K'Q$$

  - $J = 1 \,\&\, k = 0 \Rightarrow\ D = 1 \Rightarrow\ Q(t+1)\ =\ 1 \Rightarrow Set$
  - $J = 0 \,\&\, k = 1 \Rightarrow\ D = 0 \Rightarrow\ Q(t+1)\ = 0 \Rightarrow Reset$
  - $J = 1 \,\&\, k = 1 \Rightarrow\ D = Q` \Rightarrow\ Q(t+1)\ = Q` \Rightarrow Complement$
  - $J = 0 \,\&\, k = 0 \Rightarrow\ D = Q \Rightarrow\ Q(t+1)\ = Q \Rightarrow No\ change$



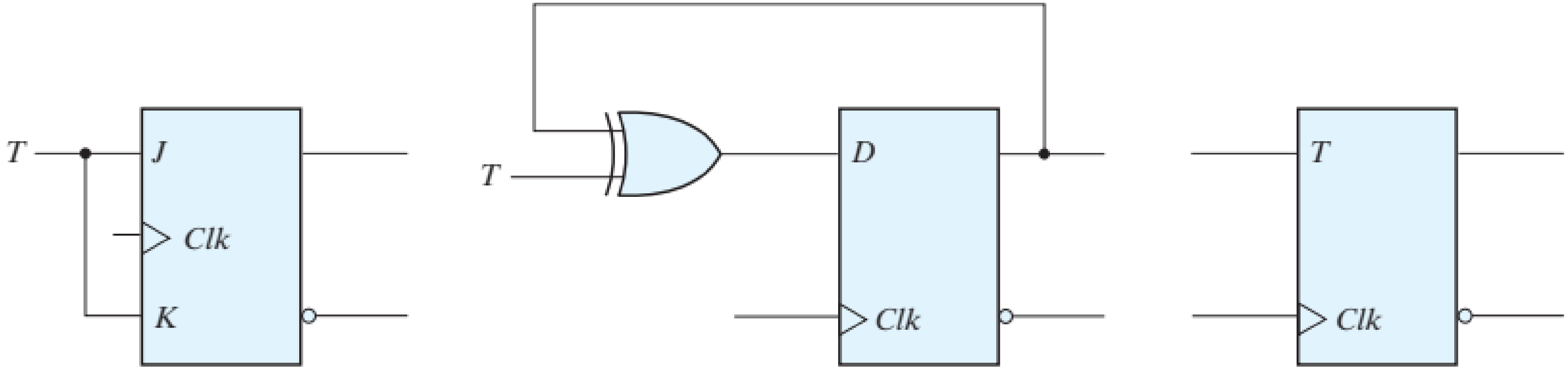
(a) Circuit diagram

T FLIP-FLOP

- T (toggle) flip-flop
- Very useful in binary counter

Operations

- **No change:** $T = 0$
- **Complement:** $T = 1$

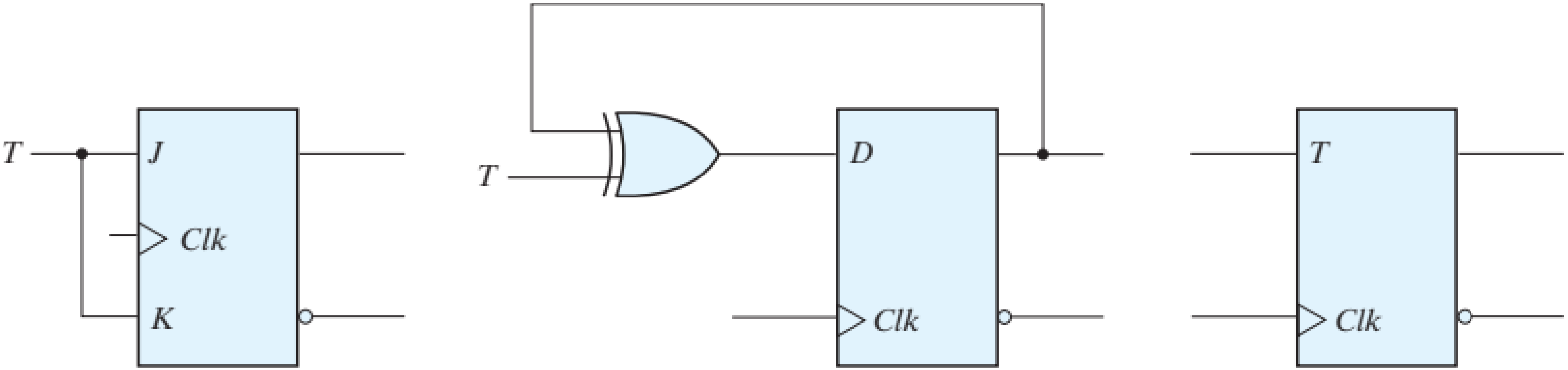


(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

T FLIP-FLOP

- Could be constructed by tying up J & K inputs together
- $\boldsymbol{T = 1 \Rightarrow J = 1 \;\&\; k = 1 \Rightarrow\; D = Q` \Rightarrow\; Q(t+1) \;= Q` \Rightarrow \mathit{Complement}}$
- $\boldsymbol{T = 0 \Rightarrow J = 0 \;\&\; k = 0 \Rightarrow\; D = Q \Rightarrow\; Q(t+1) \;= Q \Rightarrow \mathit{No\ change}}$
- Could be constructed from D flip-flop with XOR

$$D = T \oplus Q = TQ' + T'Q$$



(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

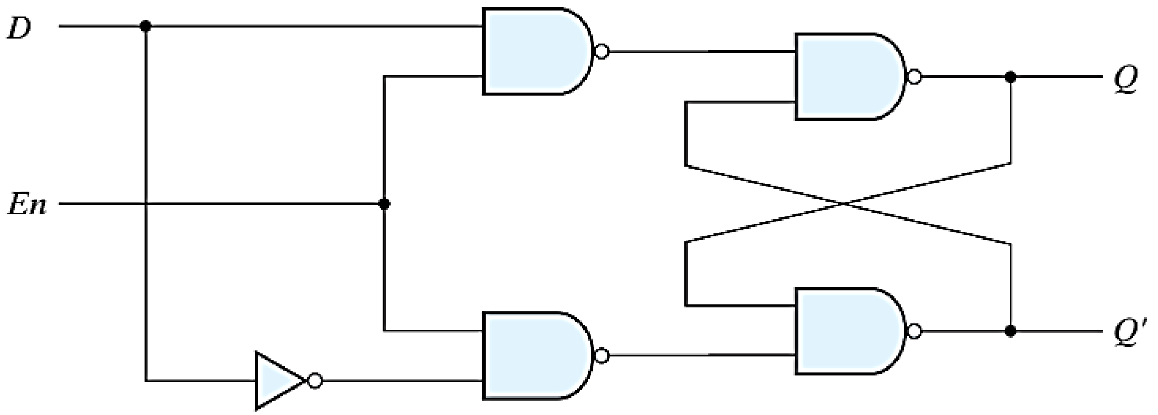
PROBLEMS

5.1 The *D* latch of Fig. 5.6 is constructed with four NAND gates and an inverter. Consider the following three other ways for obtaining a *D* latch. In each case, draw the logic diagram and verify the circuit operation.

(a) Use NOR gates for the *SR* latch part and AND gates for the other two. An inverter may be needed.

(b) Use NOR gates for all four gates. Inverters may be needed.

(c) Use four NAND gates only (without an inverter). This can be done by connecting the output of the upper gate in Fig. 5.6 (the gate that goes to the *SR* latch) to the input of the lower gate (instead of the inverter output).

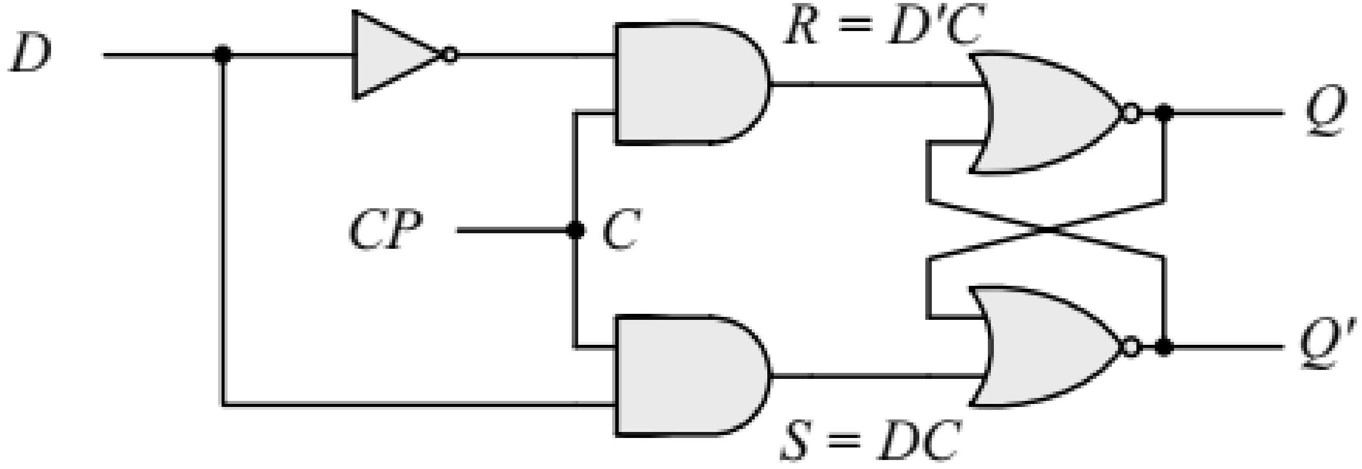


(a) Logic diagram

FIGURE 5.6

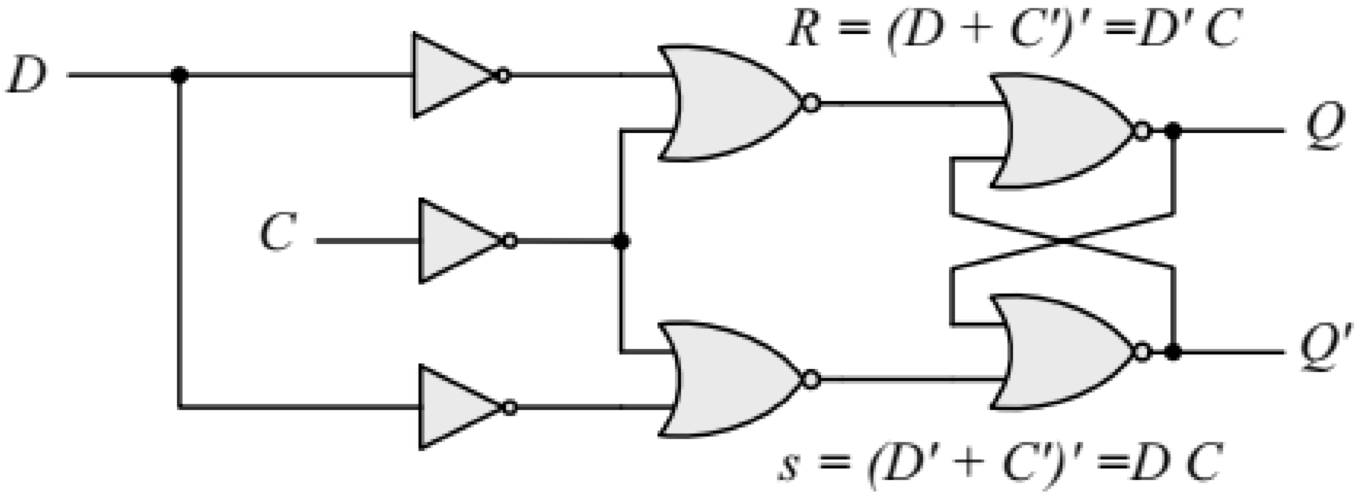
PROBLEMS

(a)



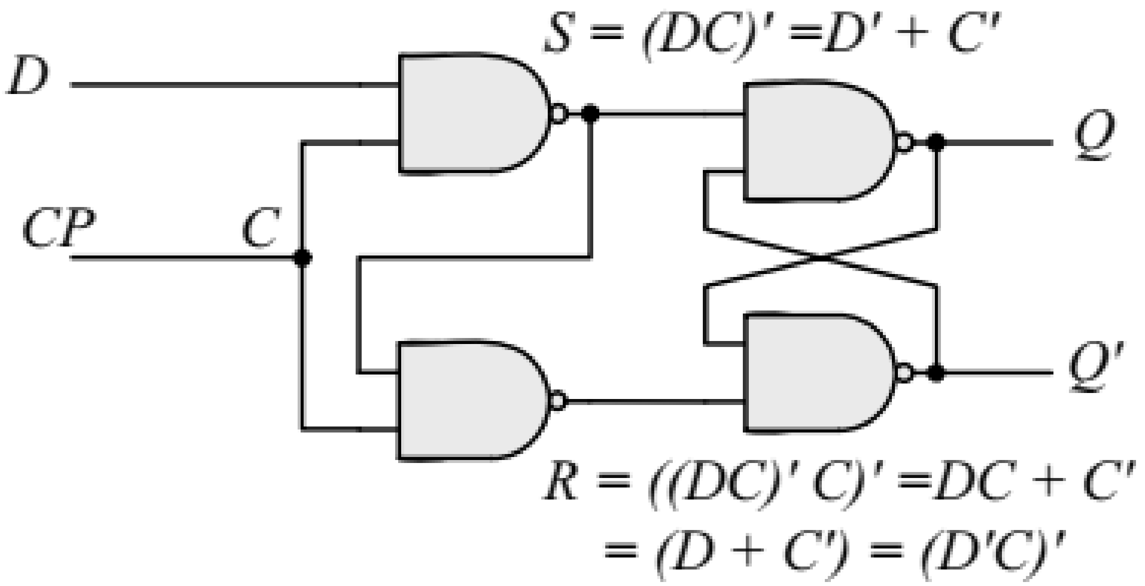
PROBLEMS

(b)



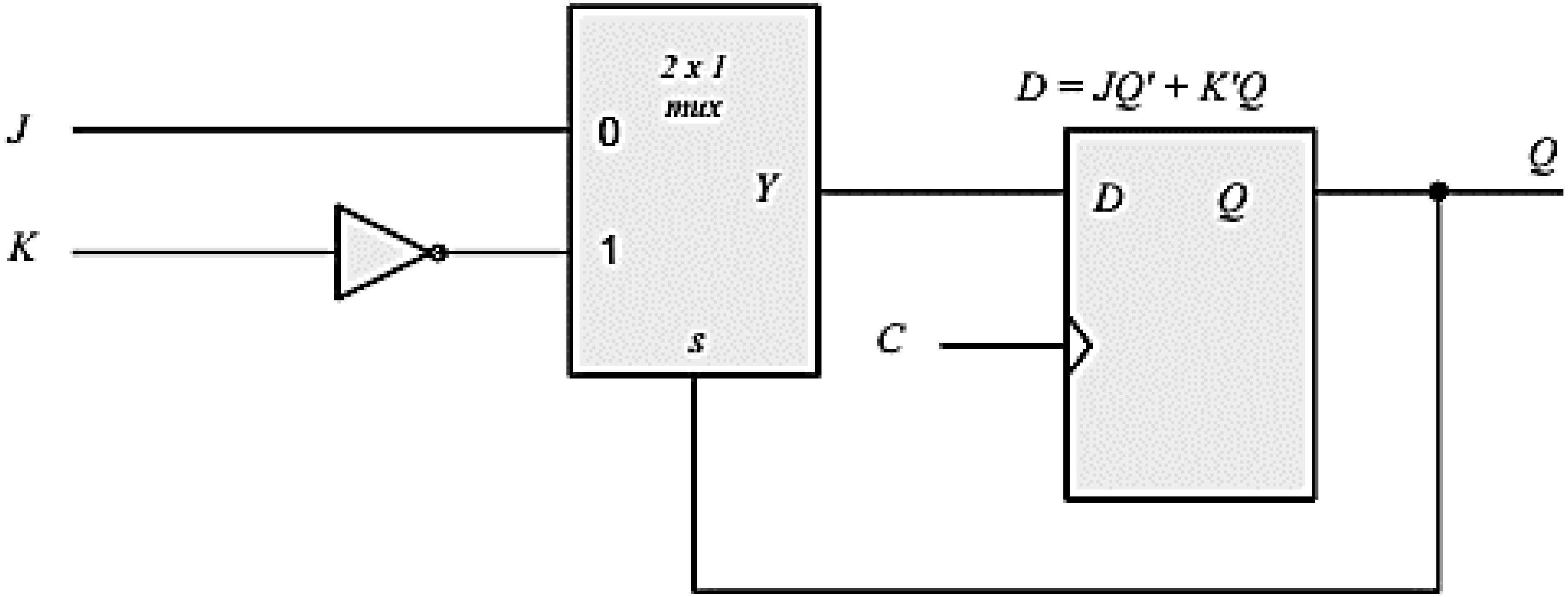
PROBLEMS

(c)



PROBLEMS

5.2 Construct a *JK* flip-flop using a *D* flip-flop, a two-to-one-line multiplexer, and an inverter. (HDL—see Problem 5.34.)



LECTURE 3

Timing analysis

CHARACTERISTIC TABLES

- defines the logical properties of a flip-flop
  - Using tabular form
  - Shows next state as function of input and present state

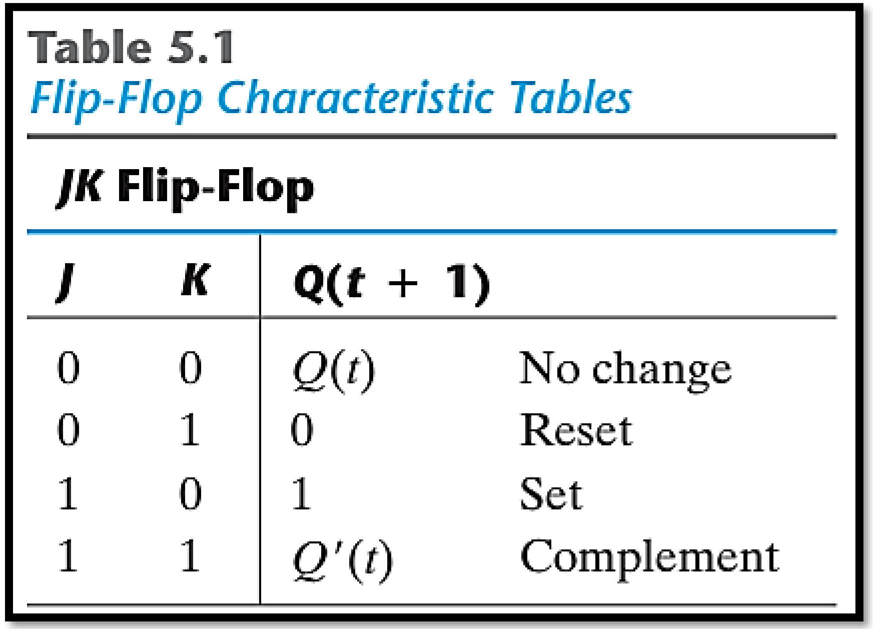
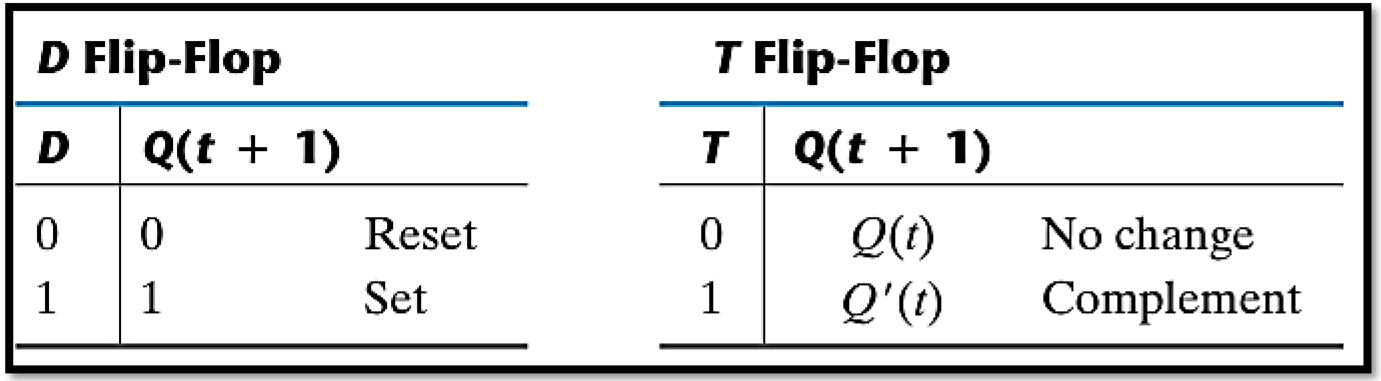


Table 5.1
Flip-Flop Characteristic Tables

***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 EQUATIONS

- Extracted from the characteristics table
  - For **D** flip-flop the next state of the output will be equal to the value of input **D**

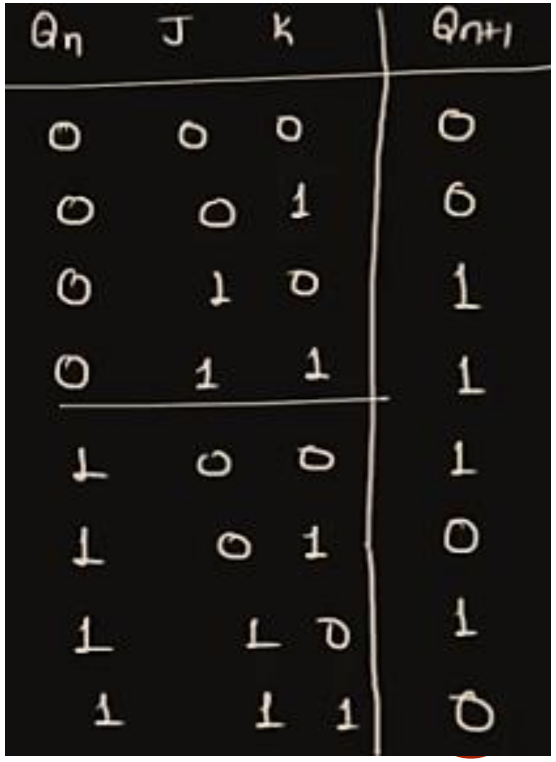
$$Q(t+1) = D$$

  - For **J-K** flip-flop, the equation could be deduced from truth table

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

  - **T** flip-flop

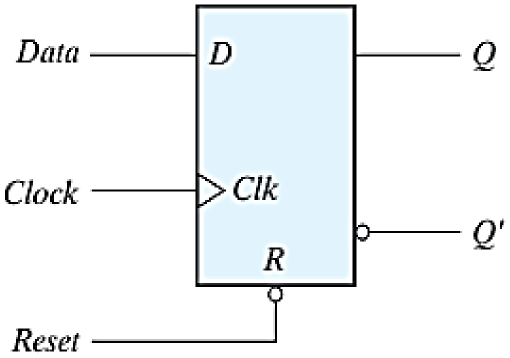
$$Q(t+1) = T \oplus Q = TQ' + T'Q$$



Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

ASYNCHRONOUS INPUT

- aka direct input
  - Non-governed by clock signal
  - Mainly used when power is turned on
  - Can set/preset the flip-flop to 1
  - Can clear/reset the flip-flop to 0

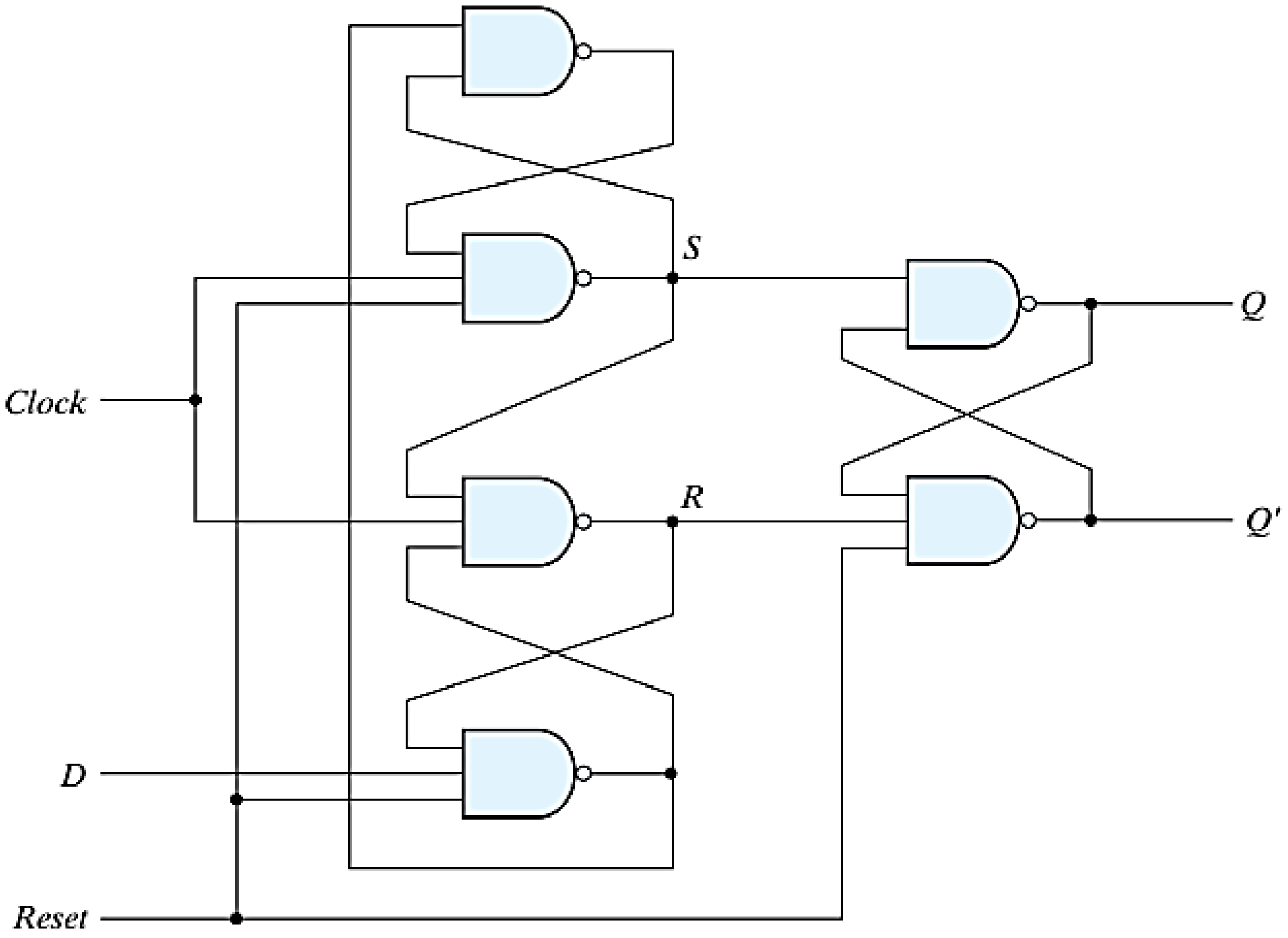


(b) Graphic symbol

R	Clk	D	Q	Q'
0	X	X	0	1
1 ~~0~~	↑	0	0	1
1 ~~0~~	↑	1	1	0

(b) Function table

FIGURE 5.14
D flip-flop with asynchronous reset



STATE EQUATIONS

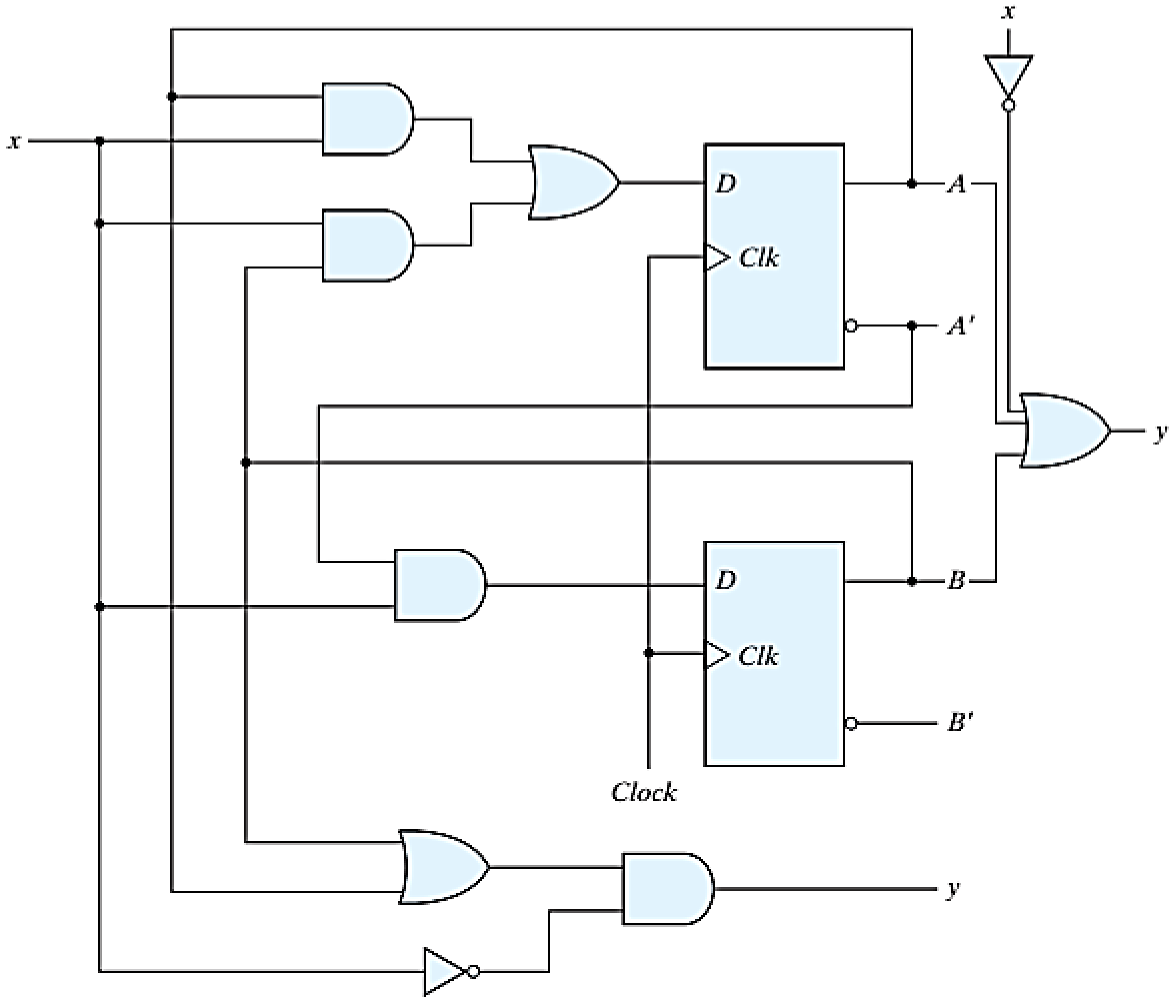
- aka transition equation
  - Describes the behavior of a clocked sequential circuit
  - specifies the next state as a function

    of the present state and inputs

$$A(t+1) = A(t)x(t) + B(t)x(t)$$

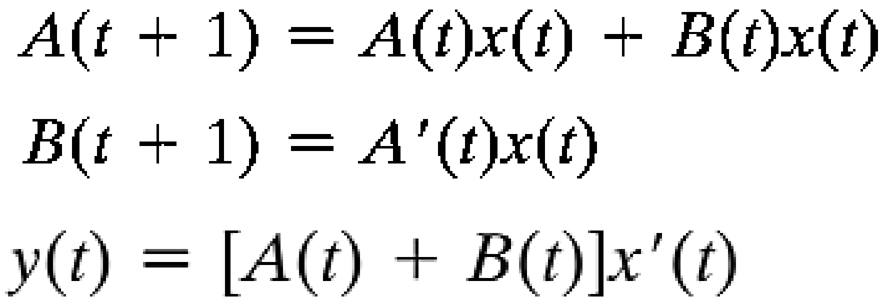
$$B(t+1) = A'(t)x(t)$$

$$y(t) = [A(t) + B(t)]x'(t)$$



STATE TABLES

- aka transition table
  - a sequential circuit with ***m*** flip-flops and ***n*** inputs needs 2^{m+n} rows in the state table
  - The next-state section has ***m*** columns, one for each flip-flop



$$A(t+1) = A(t)x(t) + B(t)x(t)$$

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

$$y(t) = [A(t) + B(t)]x'(t)$$

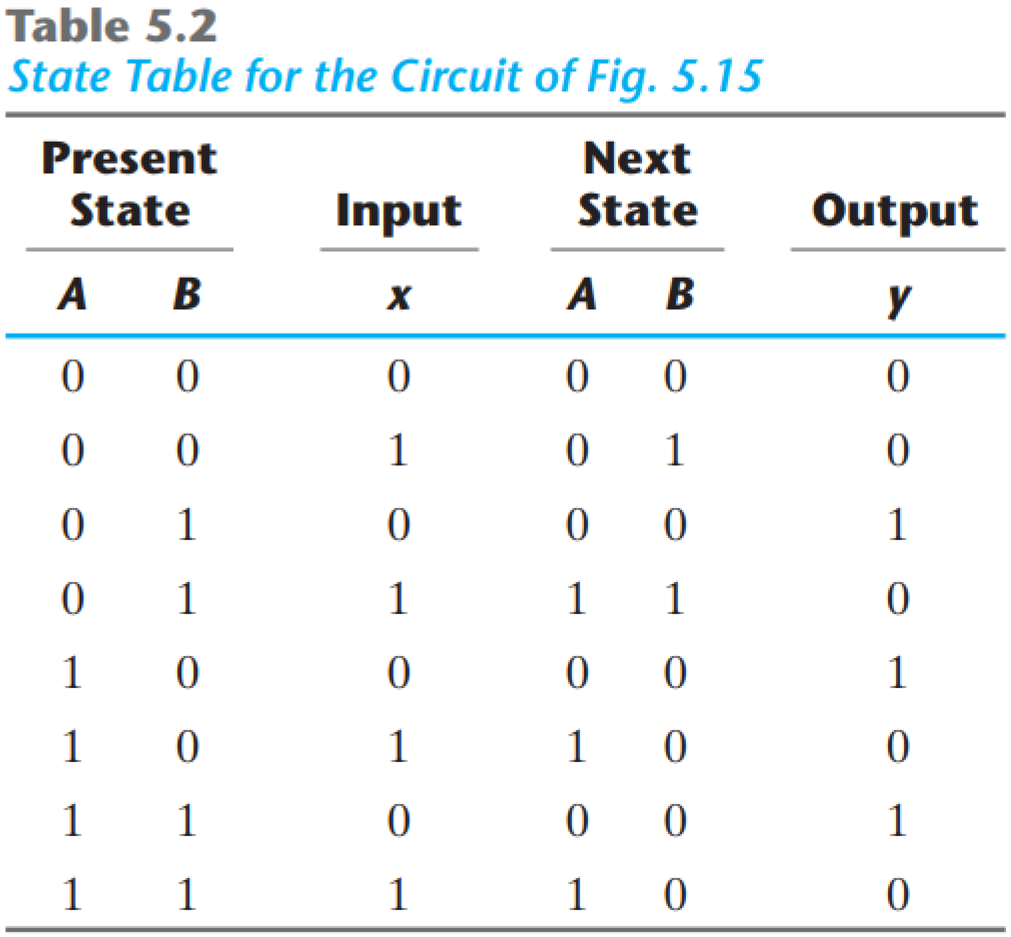


Table 5.2
State Table for the Circuit of Fig. 5.15

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 TABLES

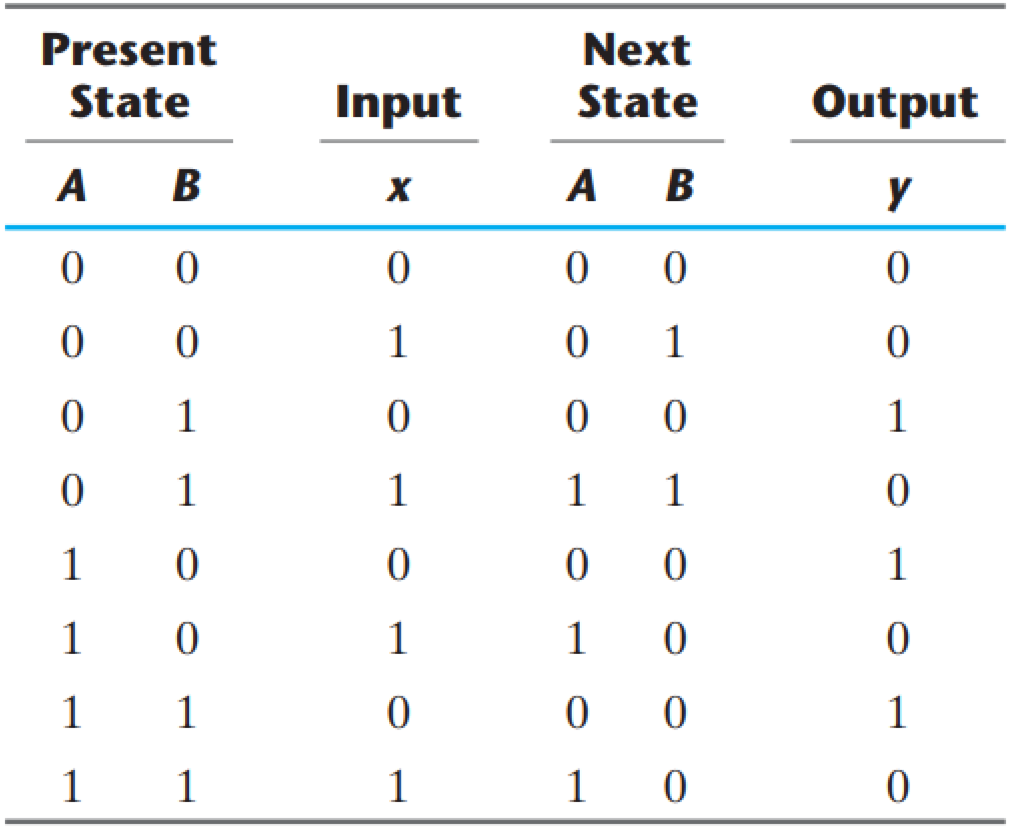
- aka transition table

$$A(t+1) = A(t)x(t) + B(t)x(t)$$

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

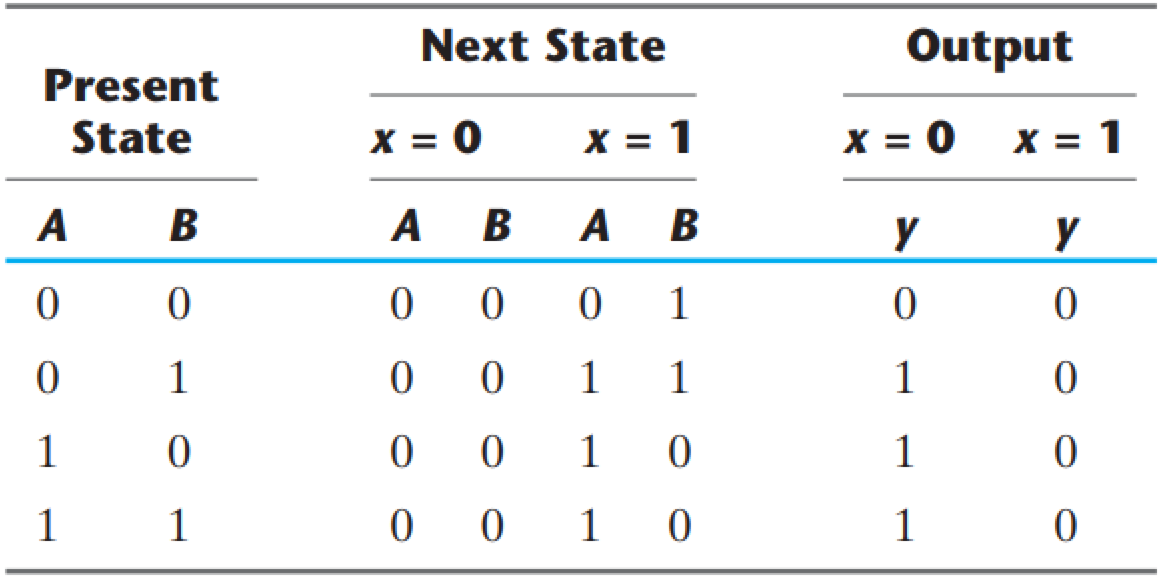
$$y(t) = [A(t) + B(t)]x'(t)$$

Table 5.2
State Table for the Circuit of Fig. 5.15



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

Table 5.3
Second Form of the State Table

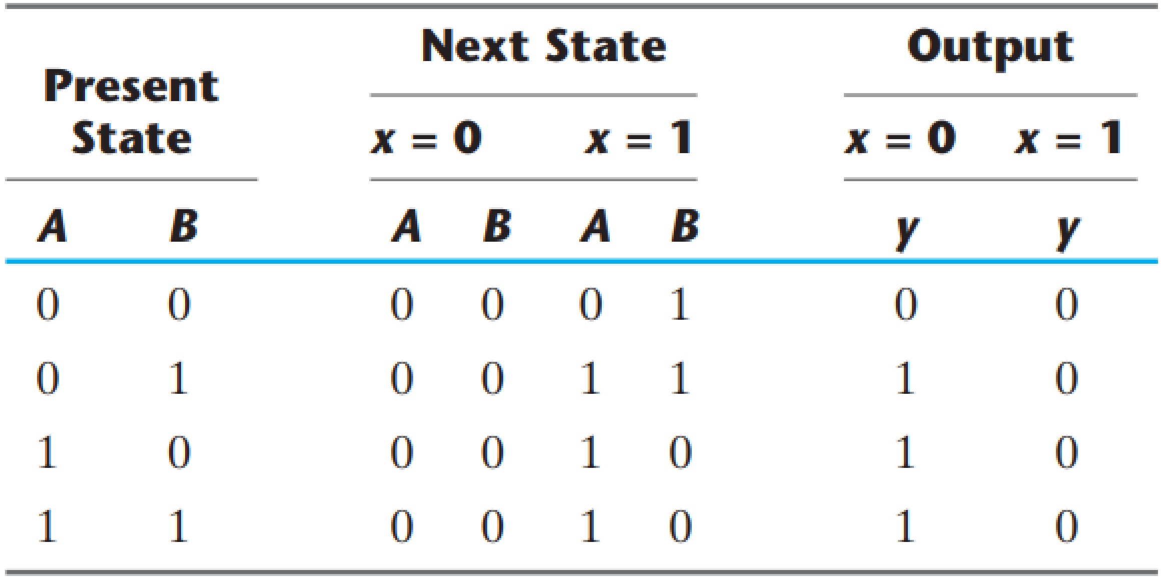


Present State		Next State				Output	
		x = 0		*x* = 1		*x* = 0	*x* = 1
A	*B*	*A*	*B*	*A*	*B*	*y*	*y*
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

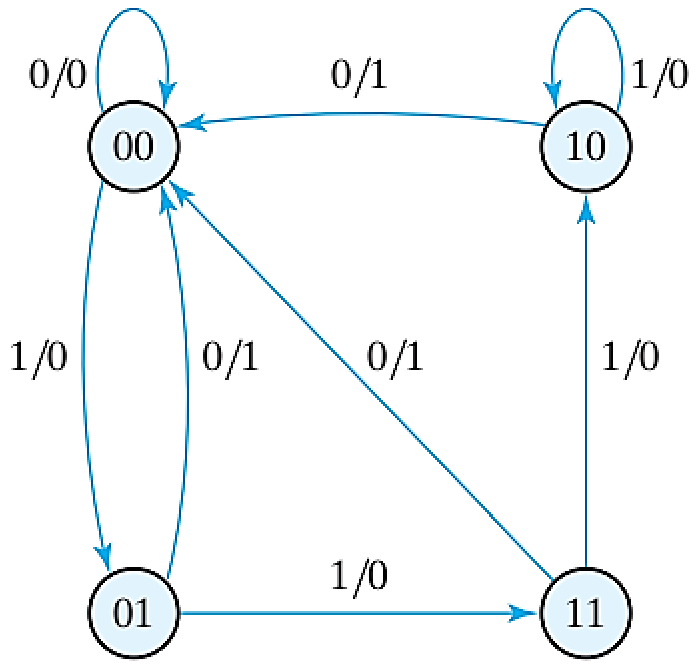
STATE DIAGRAM

- State is represented as circle
- Transition is represented as arcs/lines
- binary number inside each circle identifies the state of the flip-flops
- The directed lines are labeled with two binary numbers (input/output)

Table 5.3
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 ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

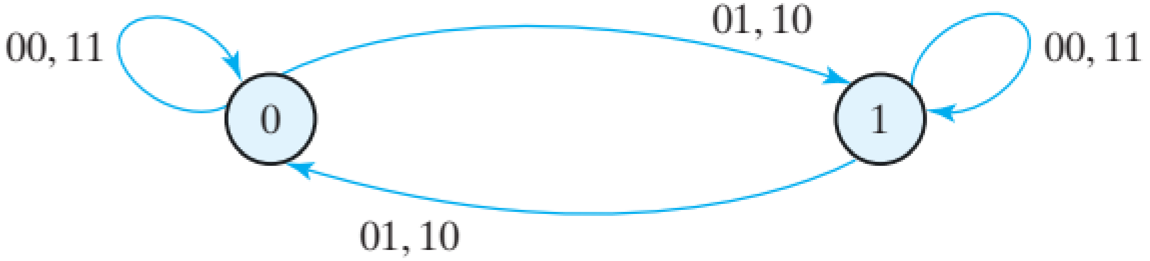
$$D_A = A \oplus x \oplus y$$

Input equation	$D_A = A \oplus x \oplus y$
Output equation	$D_A = A \oplus x \oplus y$
State equation	$A(t + 1) = A \oplus x \oplus y$
State table	
State diagram	

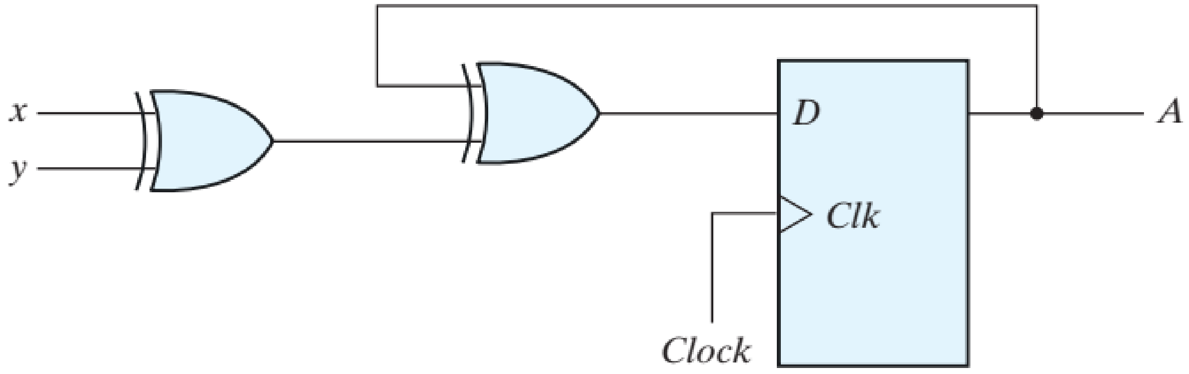
STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams
- State table
- State diagram

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



(c) State diagram



(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

LECTURE 4

Finite state machine

STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

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

- Input equation
- Output equation
- State equation
- State table
- State diagram

STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

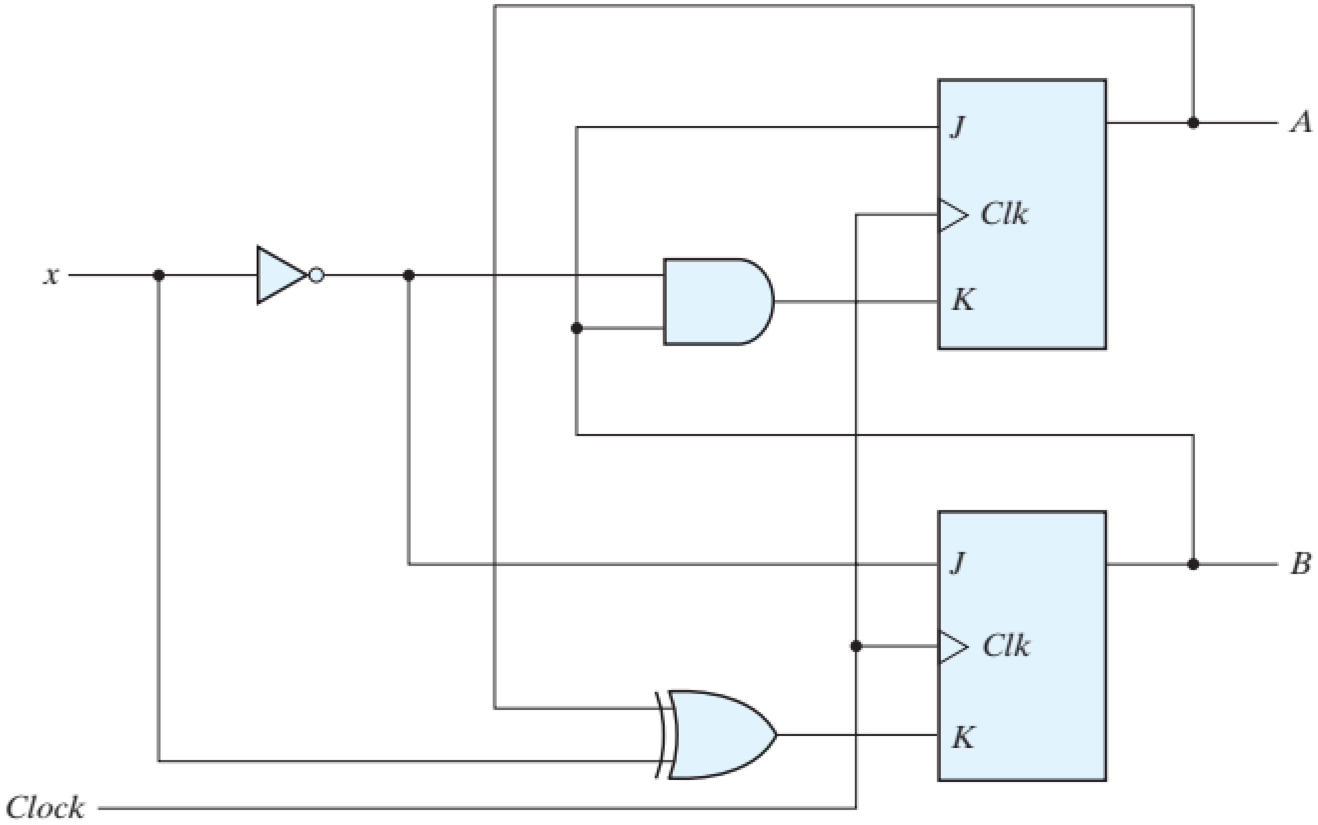


FIGURE 5.18
Sequential circuit with *JK* flip-flop

STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

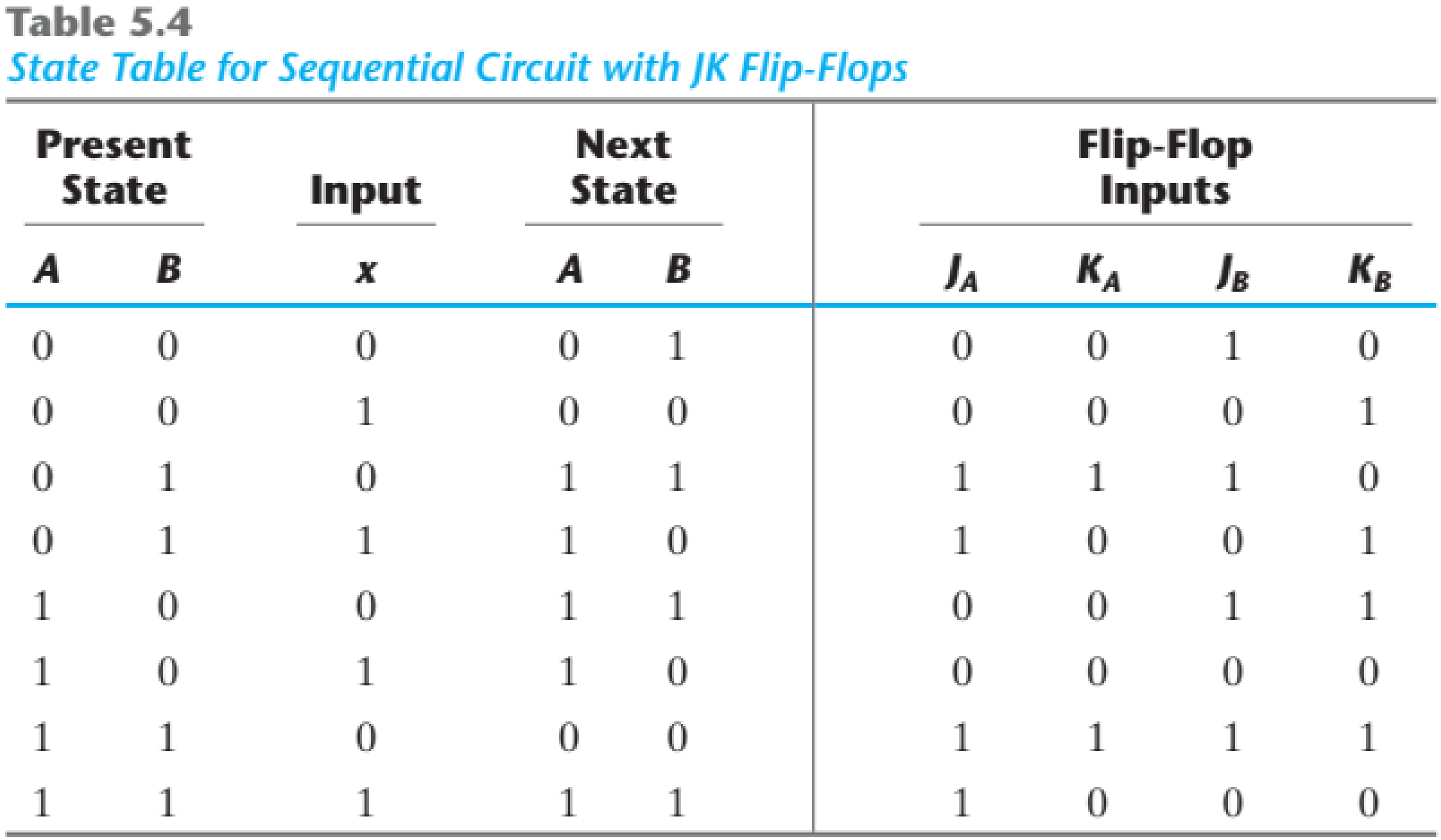


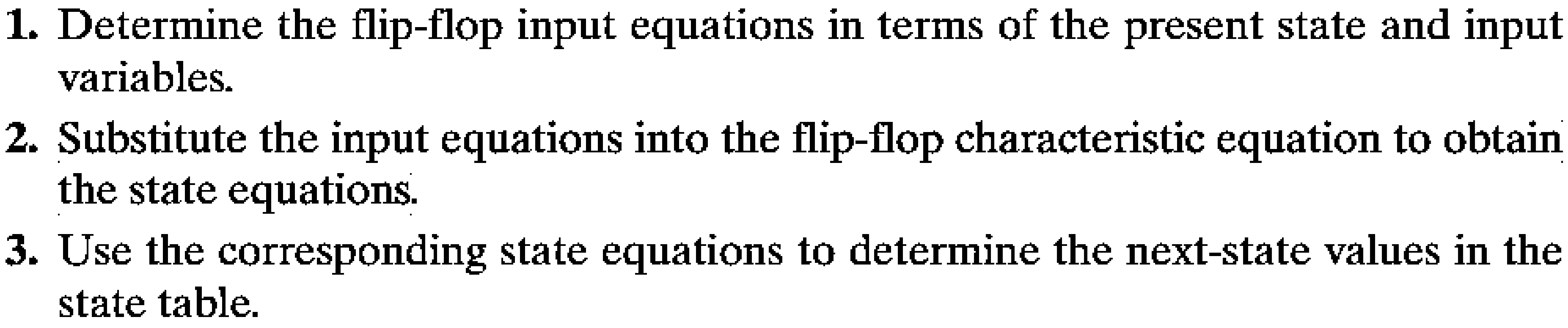
Table 5.4

State Table for Sequential Circuit with JK Flip-Flops

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	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

STATE ANALYSIS

- The next-state values can also be obtained by evaluating the state equations from the characteristic equation



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.

$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'x + Ax' = A \oplus x$$

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

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

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

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

STATE ANALYSIS

- State diagram

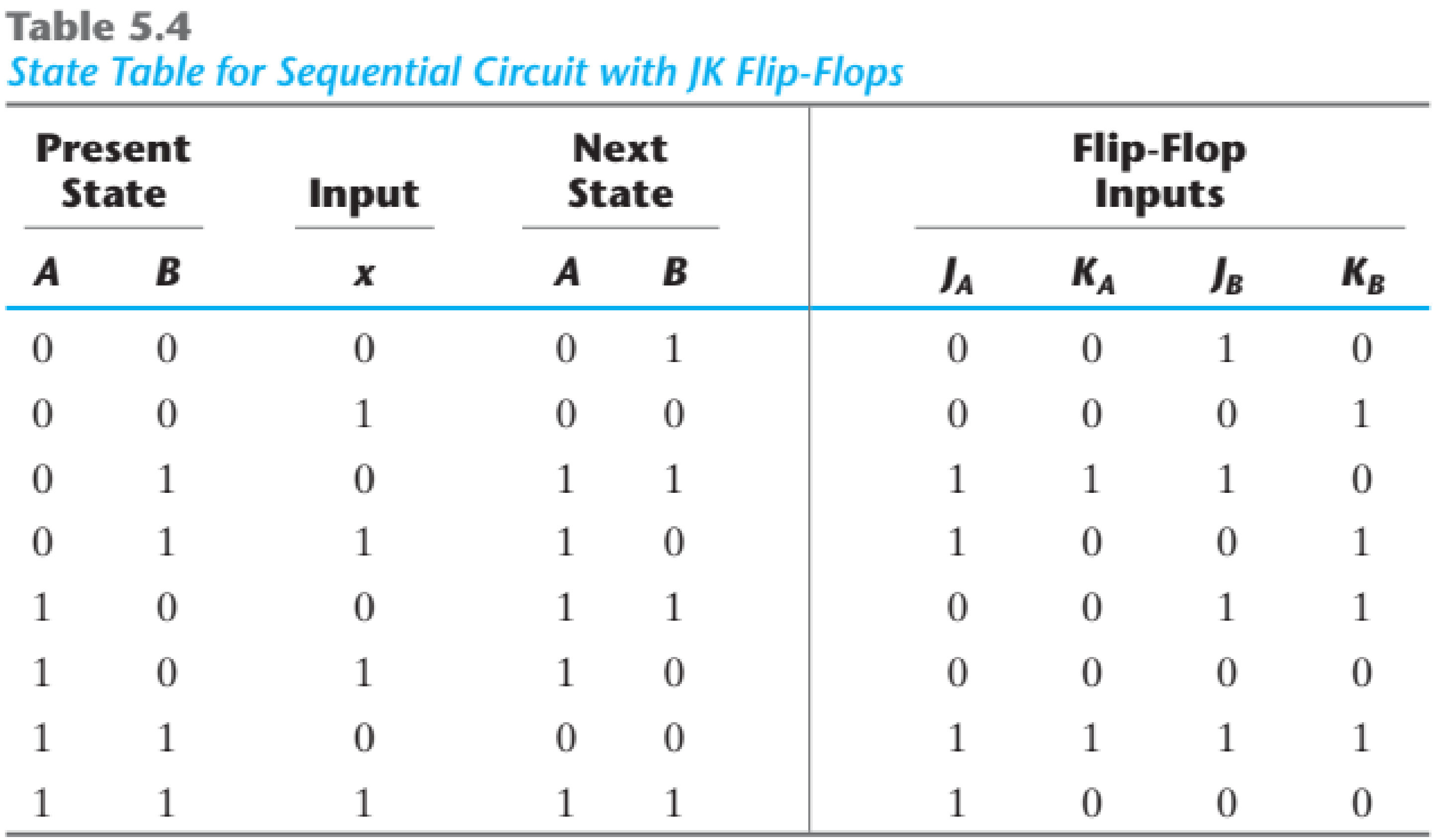
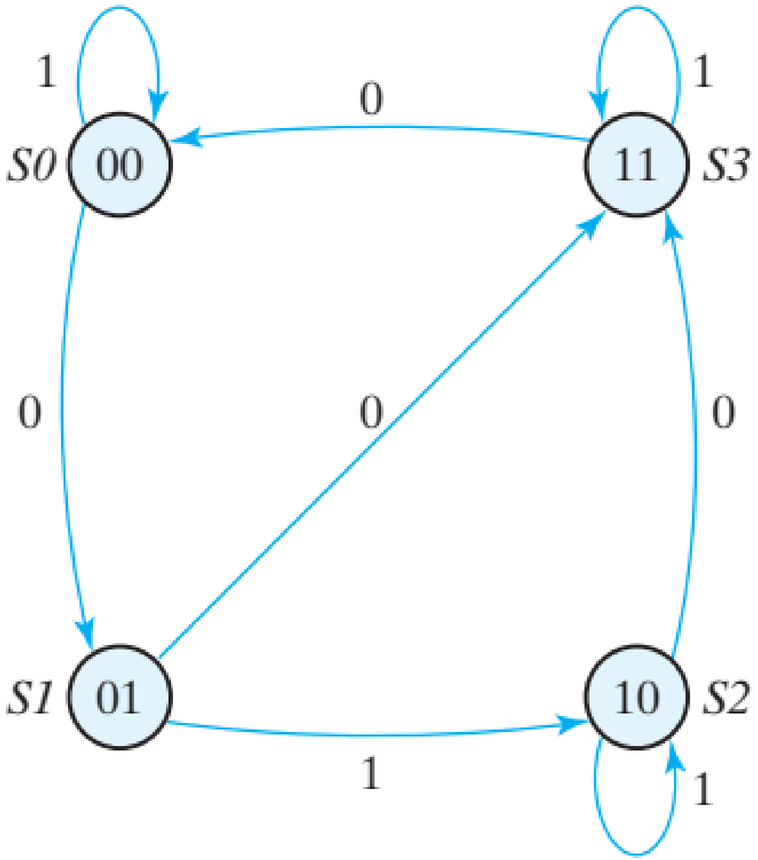


Table 5.4
State Table for Sequential Circuit with JK Flip-Flops

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	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



STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

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

- Input equation
- Output equation
- State equation
- State table
- State diagram

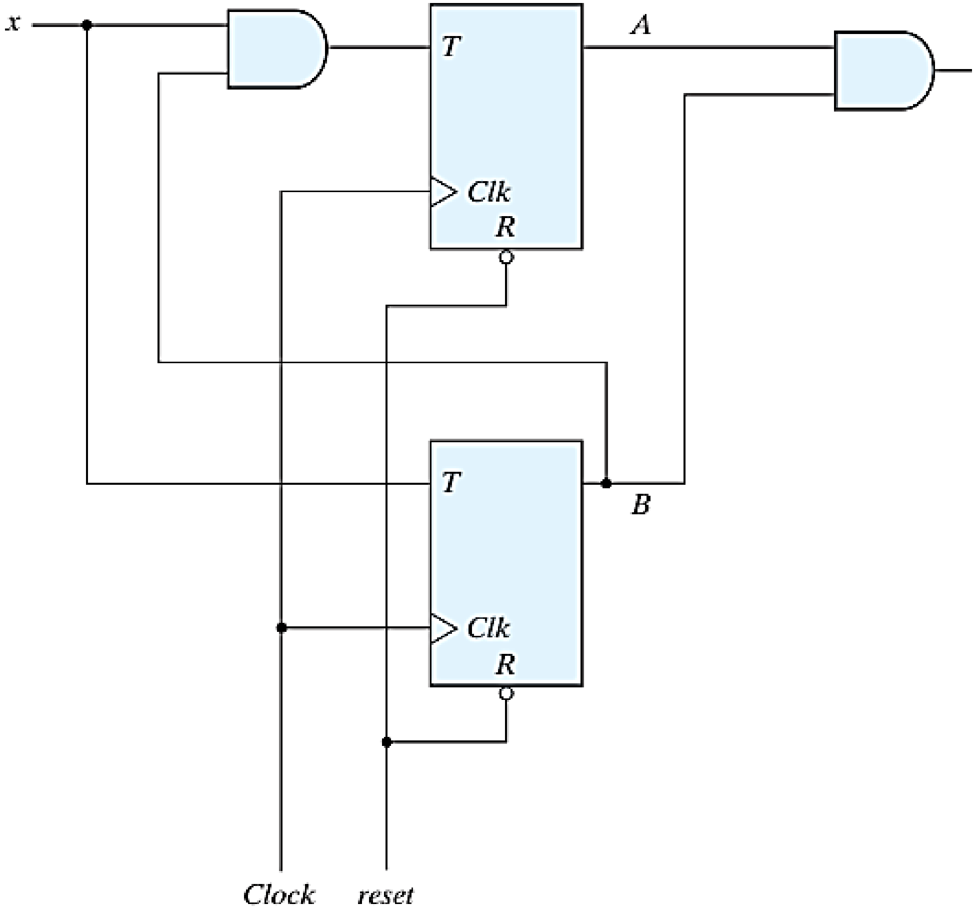
STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$



(a) Circuit diagram

FIGURE 5.20
Sequential circuit with *T* flip-flops (Binary Counter)

- State equation

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

$$\Downarrow$$

$$A(t+1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$$

$$B(t+1) = x \oplus B$$

STATE ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

$$T_A = Bx$$

$$T_B = x$$

$$y = AB$$

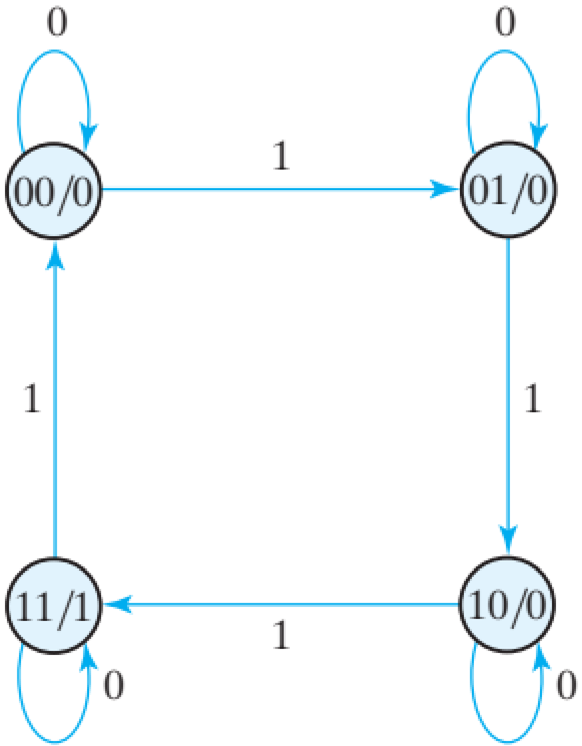


Table 5.5
State Table for Sequential Circuit with T Flip-Flops

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 ANALYSIS

- Analyze the following sequential circuit equation using state diagrams

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

- Observations
  - $\boldsymbol{X = 1} \Rightarrow Binary\ counter$
  - $\boldsymbol{X = 0} \Rightarrow Retain\ value$

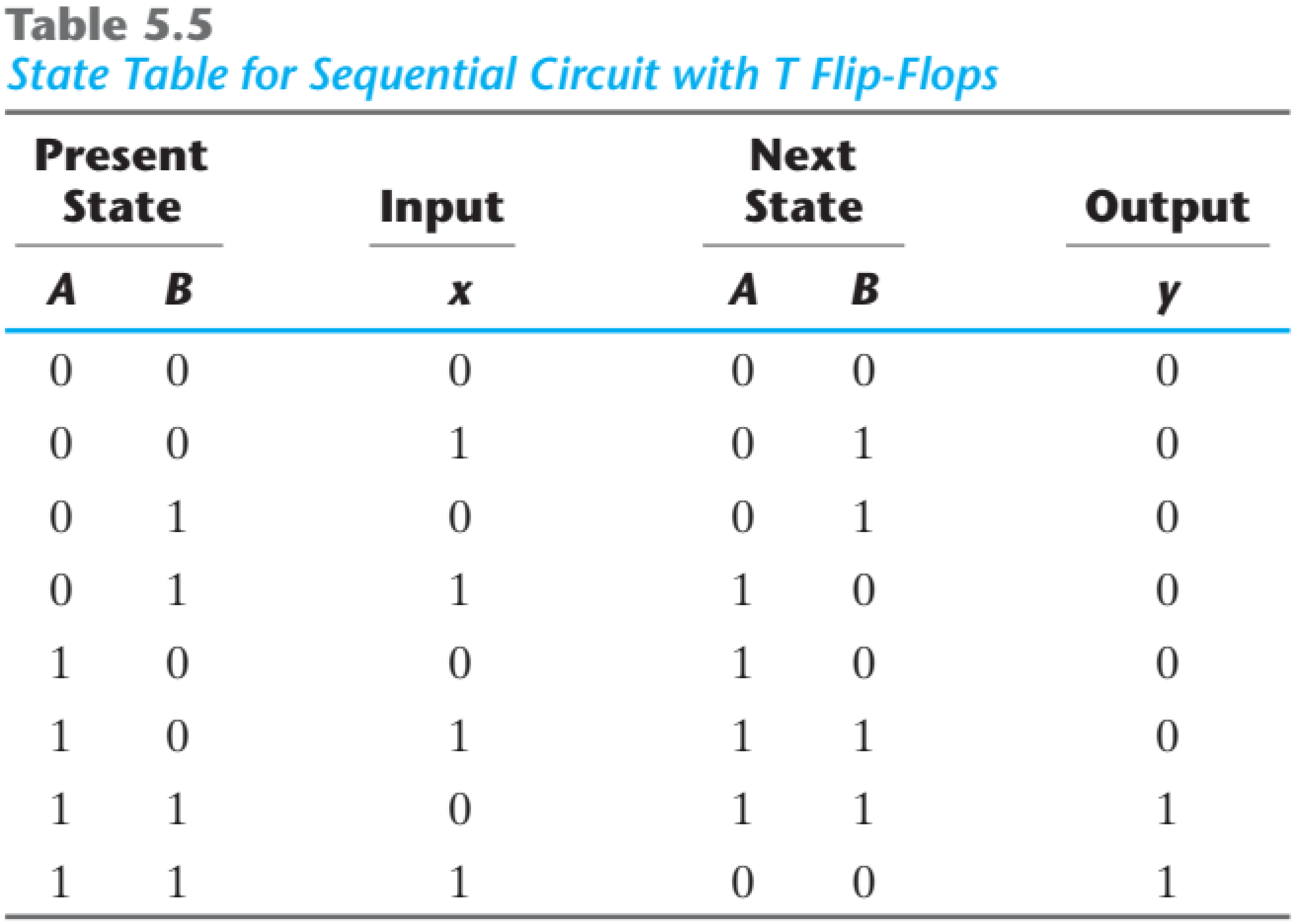


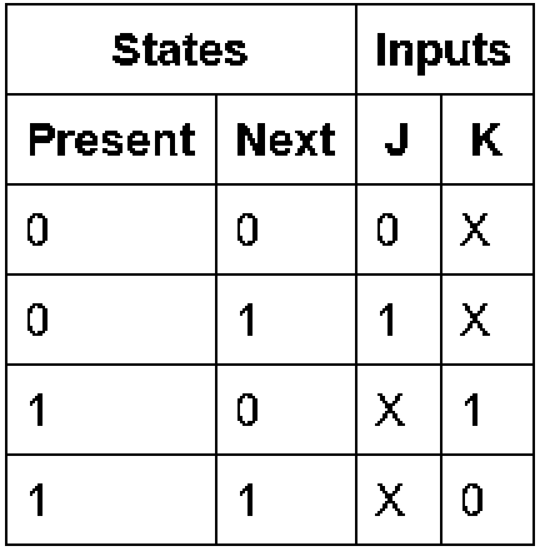
Table 5.5
State Table for Sequential Circuit with T Flip-Flops

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

EXCITATION TABLE

- Describes flip-flop input equation required to excite flip-flop to next state
  - shows the minimum inputs necessary to generate a particular next state when the current state is known
  - Current state and next state are next to each other on the left-hand side of the table
  - inputs needed to make that state change happen are shown on the right side of the table

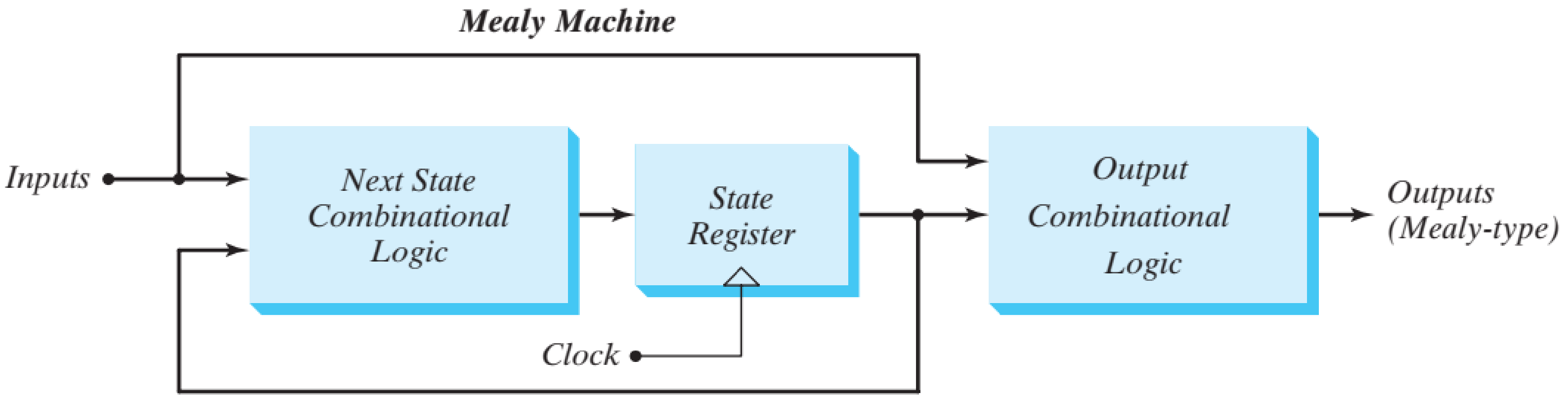
- Question
  - Find excitation table for JK flip-flop

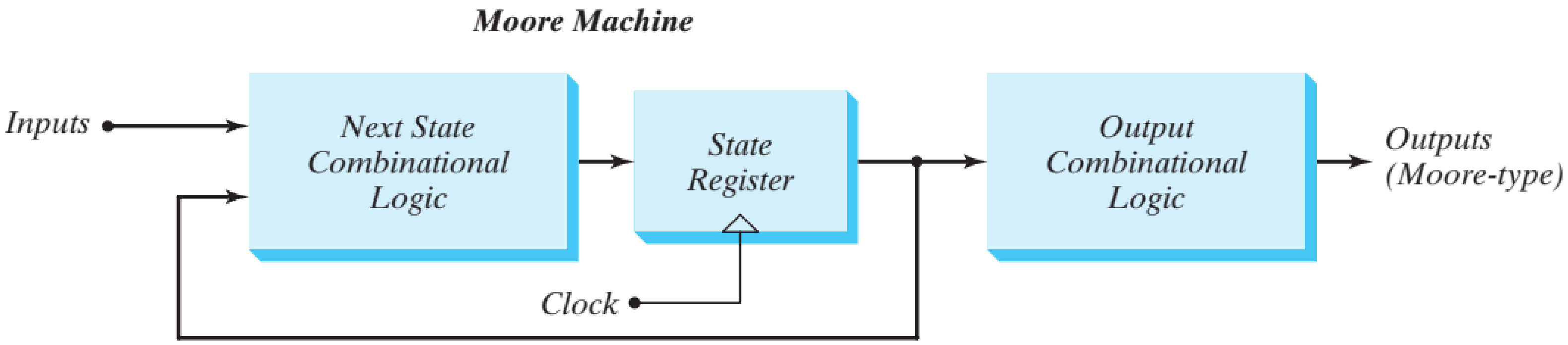


States		Inputs	
Present	Next	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

FINITE STATE MACHINE

- Two types of FSM
  - Mealy model
  - Moore model

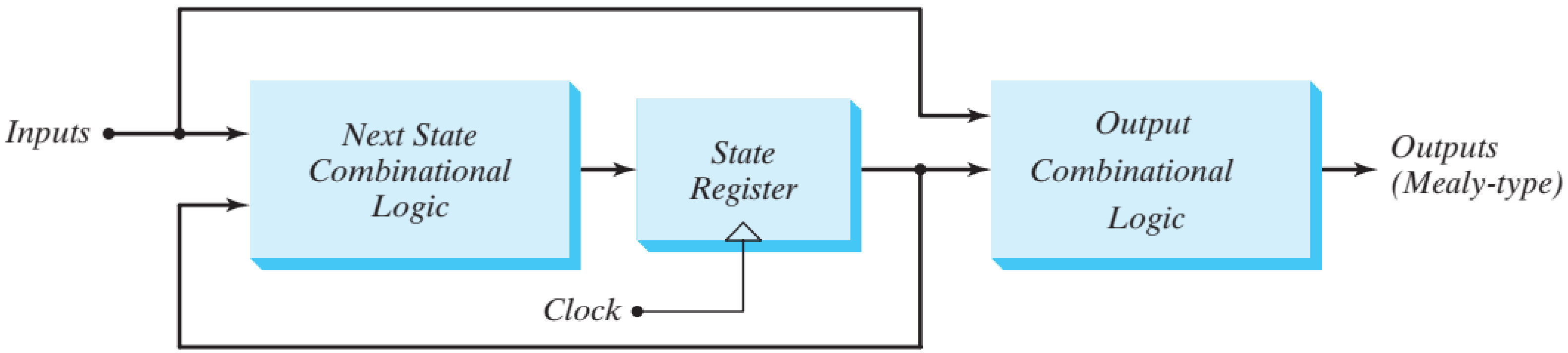




FINITE STATE MACHINE

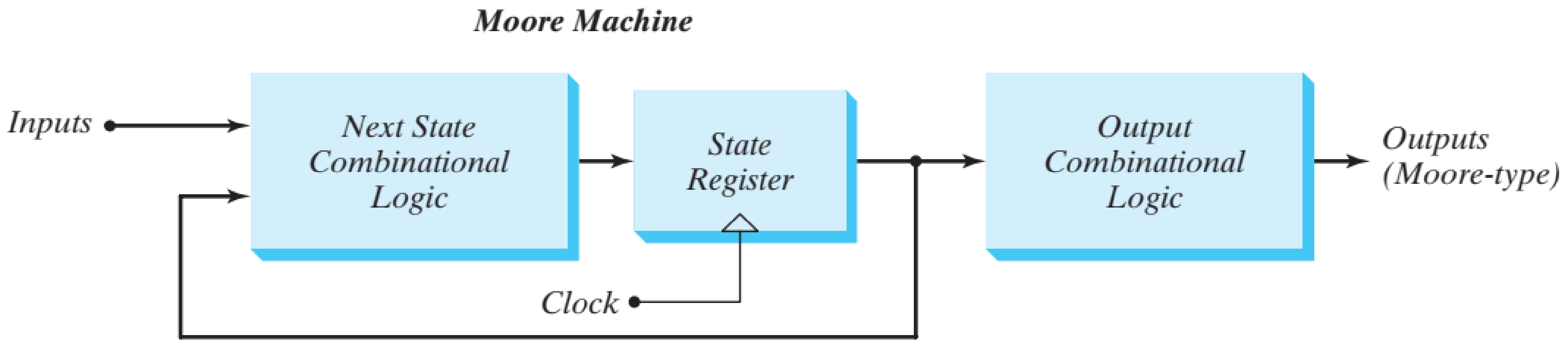
- Mealy model
  - Output is function of input & present state
  - Input is not synchronized with clock
  - Output may change according to input
  - Output may have false values until input propagates and flip-flop changes

Mealy Machine

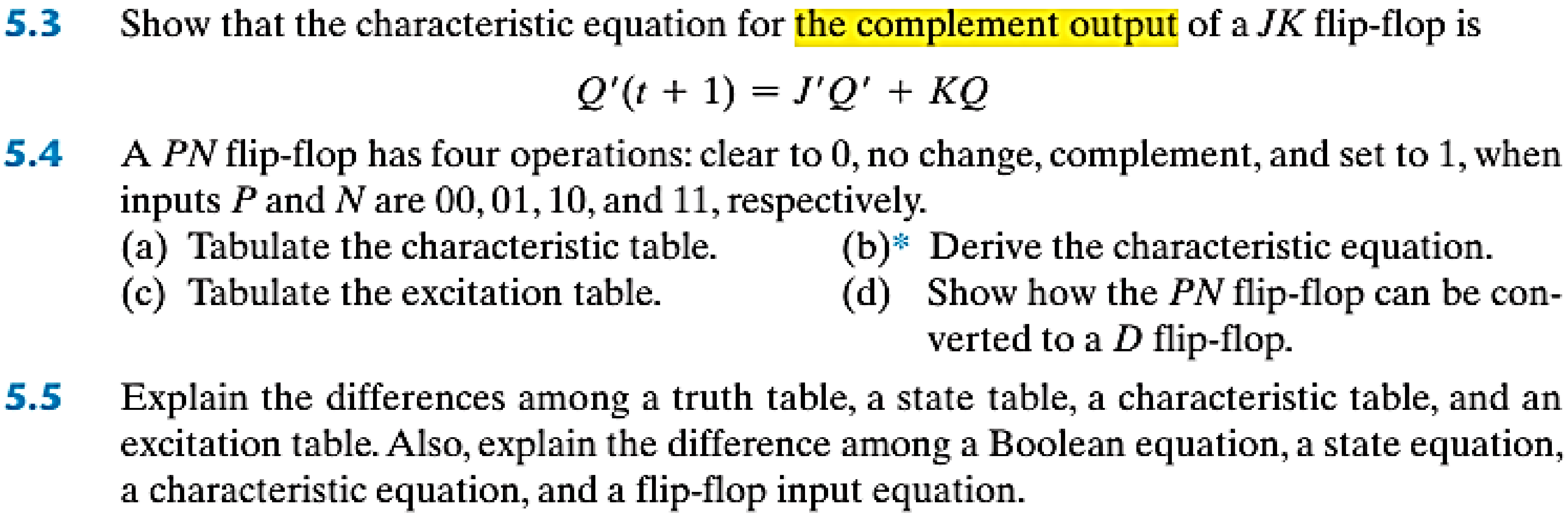


FINITE STATE MACHINE

- Moore model
  - Output is function of only present state
  - Output is synchronized with the clock
  - Output may have false values until input propagates and flip-flop changes



PROBLEMS



5.3 Show that the characteristic equation for the complement output of a *JK* flip-flop is

$$Q'(t + 1) = J'Q' + KQ$$

5.4 A *PN* flip-flop has four operations: clear to 0, no change, complement, and set to 1, when inputs *P* and *N* are 00, 01, 10, and 11, respectively.

(a) Tabulate the characteristic table.

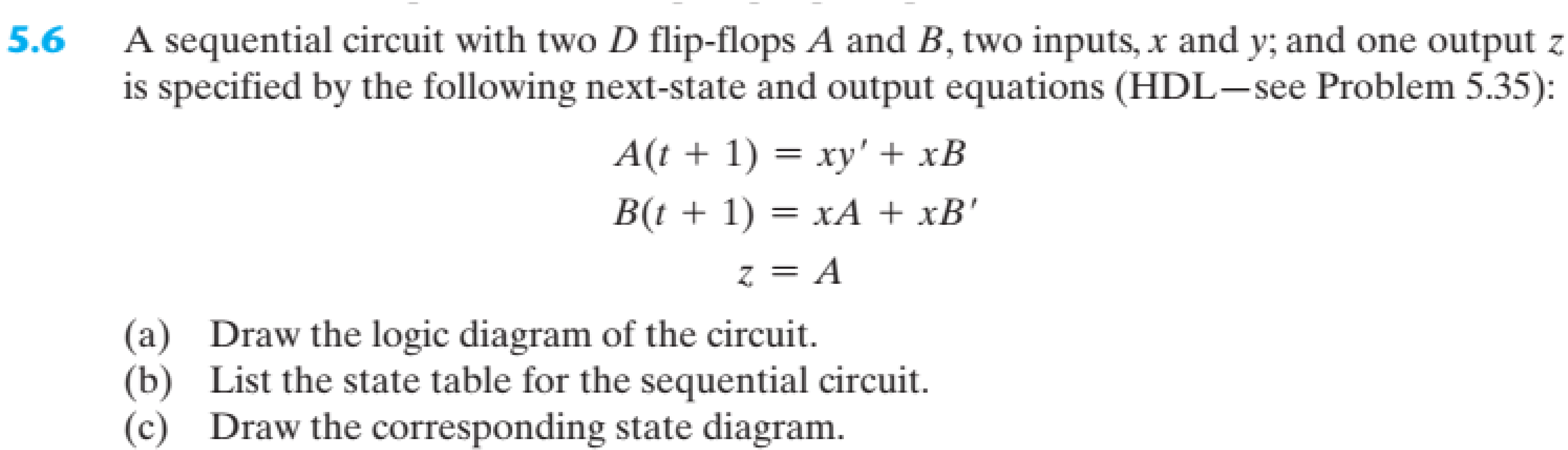
(b)* Derive the characteristic equation.

(c) Tabulate the excitation table.

(d) Show how the *PN* flip-flop can be converted to a *D* flip-flop.

5.5 Explain the differences among a truth table, a state table, a characteristic table, and an excitation table. Also, explain the difference among a Boolean equation, a state equation, a characteristic equation, and a flip-flop input equation.

PROBLEMS



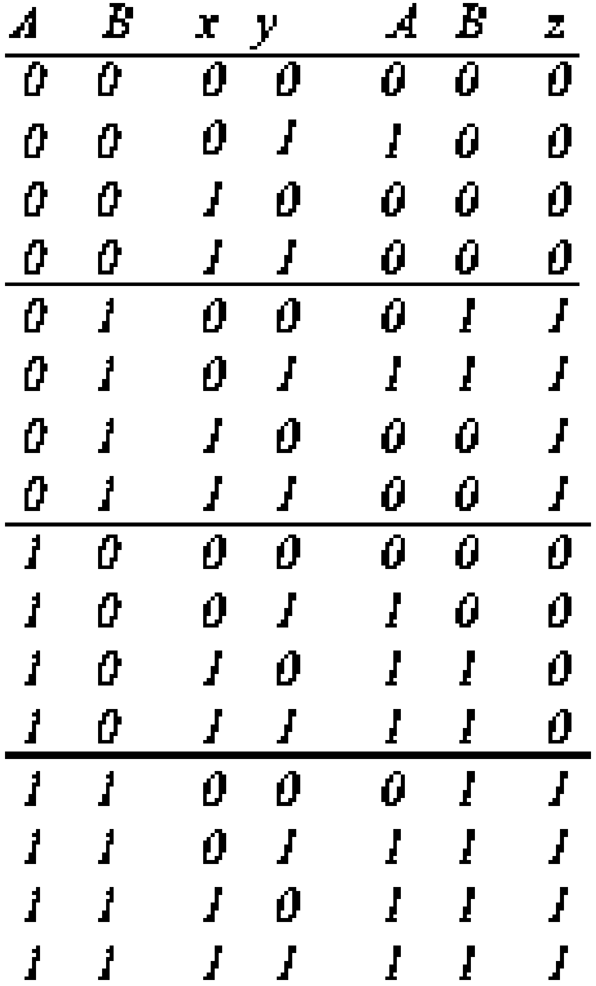
5.6 A sequential circuit with two *D* flip-flops *A* and *B*, two inputs, *x* and *y*; and one output *z* is specified by the following next-state and output equations (HDL—see Problem 5.35):

$$A(t+1) = xy' + xB$$

$$B(t+1) = xA + xB'$$

$$z = A$$

(a) Draw the logic diagram of the circuit.
(b) List the state table for the sequential circuit.
(c) Draw the corresponding state diagram.



A	B	x	y	A	B	z
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

PROBLEMS

5.7* A sequential circuit has one flip-flop Q, two inputs x and y, and one output S. It consists of a full-adder circuit connected to a D flip-flop, as shown in Fig. P5.7. Derive the state table and state diagram of the sequential circuit.

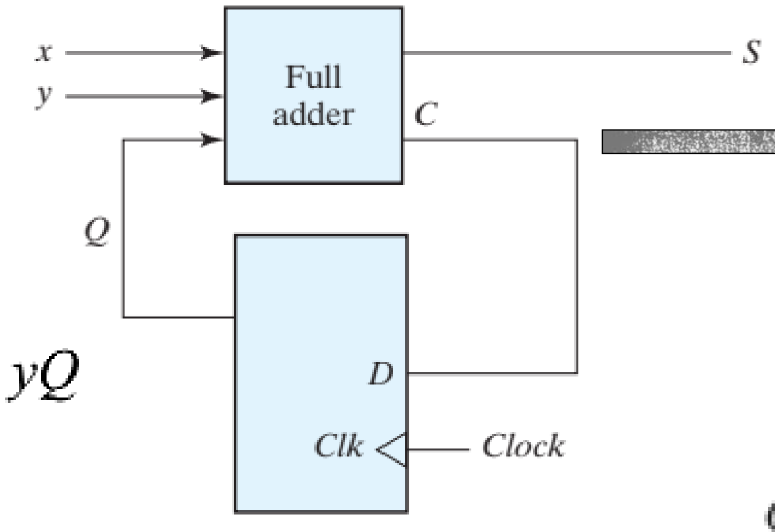


FIGURE P5.7

$$x \oplus y \oplus Q$$

$$xy + xQ + yQ$$

$$Q(t + 1) = xy + xQ + yQ$$

Q	x	y	Q	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

PROBLEMS

5.8* Derive the state table and the state diagram of the sequential circuit shown in Fig. P5.8. Explain the function that the circuit performs. (HDL—see Problem 5.36.)

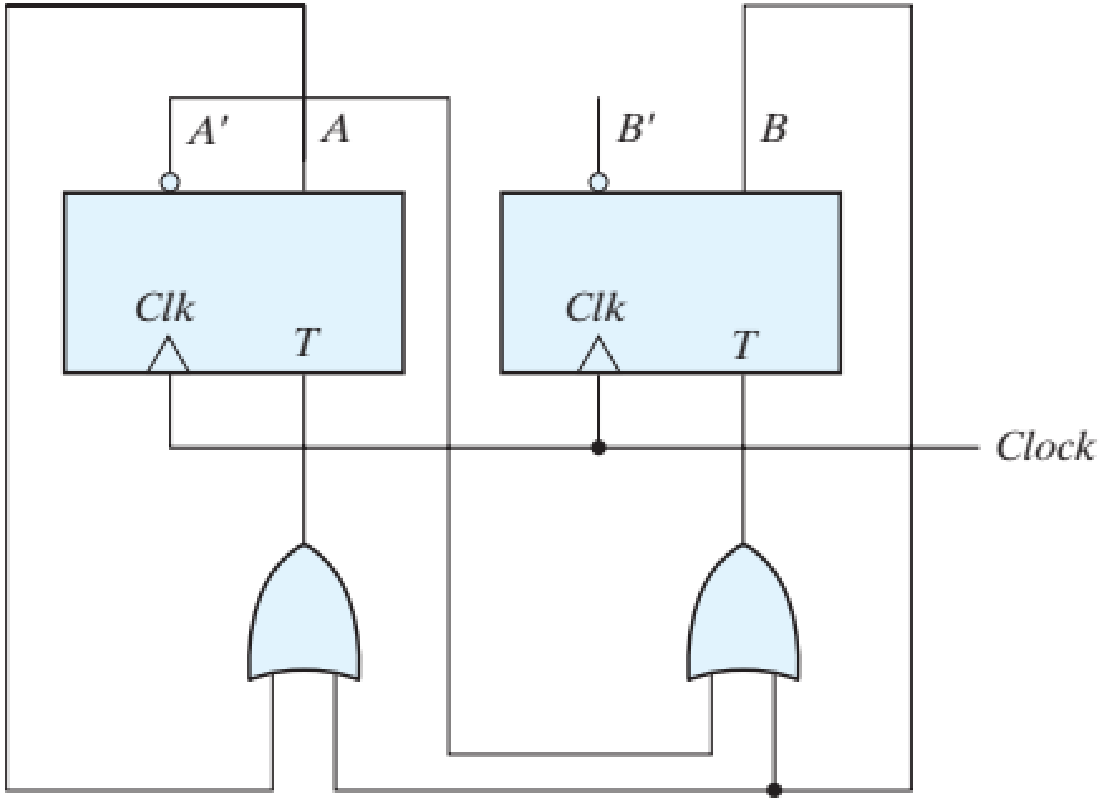


FIGURE P5.8

LECTURE 5

Sequential circuit design

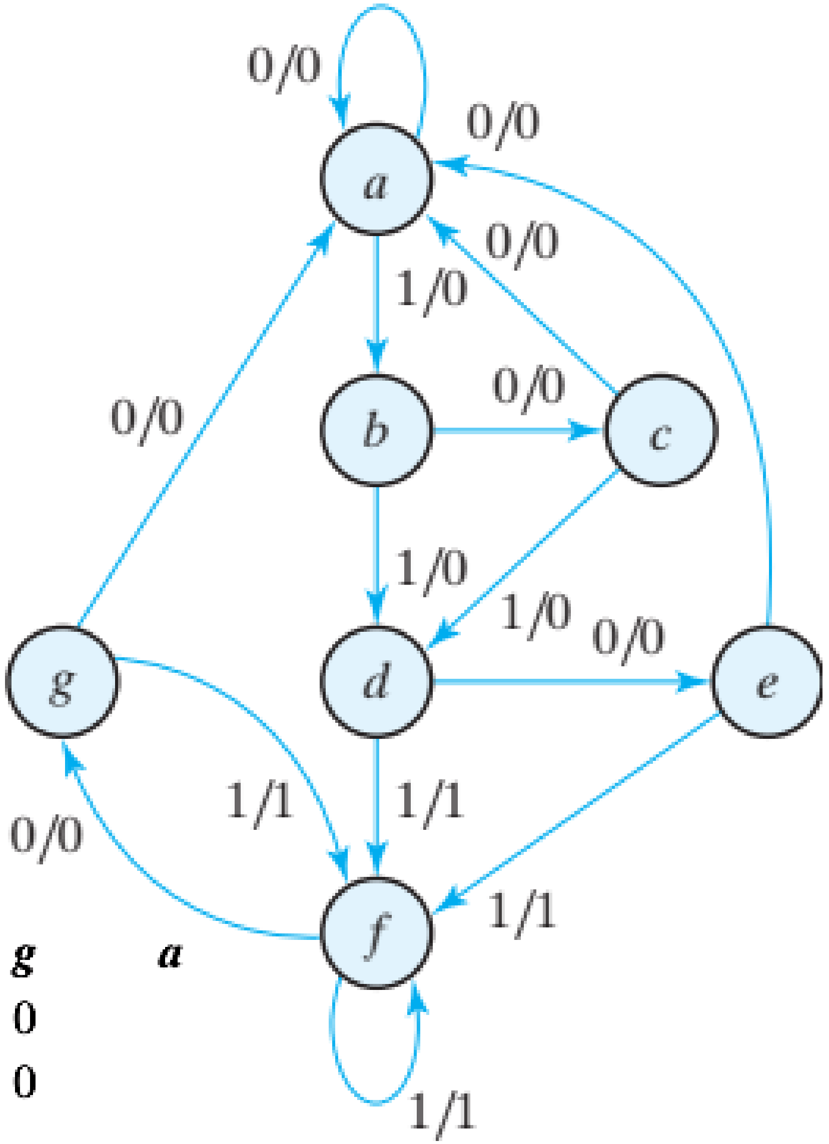
STATE REDUCTION PROBLEM

- Pertaining to the reduction of states/flip-flops in sequential circuit
- procedures for reducing the number of states in a state table, while keeping the external input–output requirements unchanged
- Produces minimal gate design
- the equivalent circuit (with fewer flip-flops) may require more combinational gates to realize its next state and output logic

STATE REDUCTION PROBLEM

- Reduce the states in the following state diagram
- Consider the input sequence 01010110100

- **Notes:**
  - States are of secondary importance
  - we are interested only in output sequences caused by input sequences

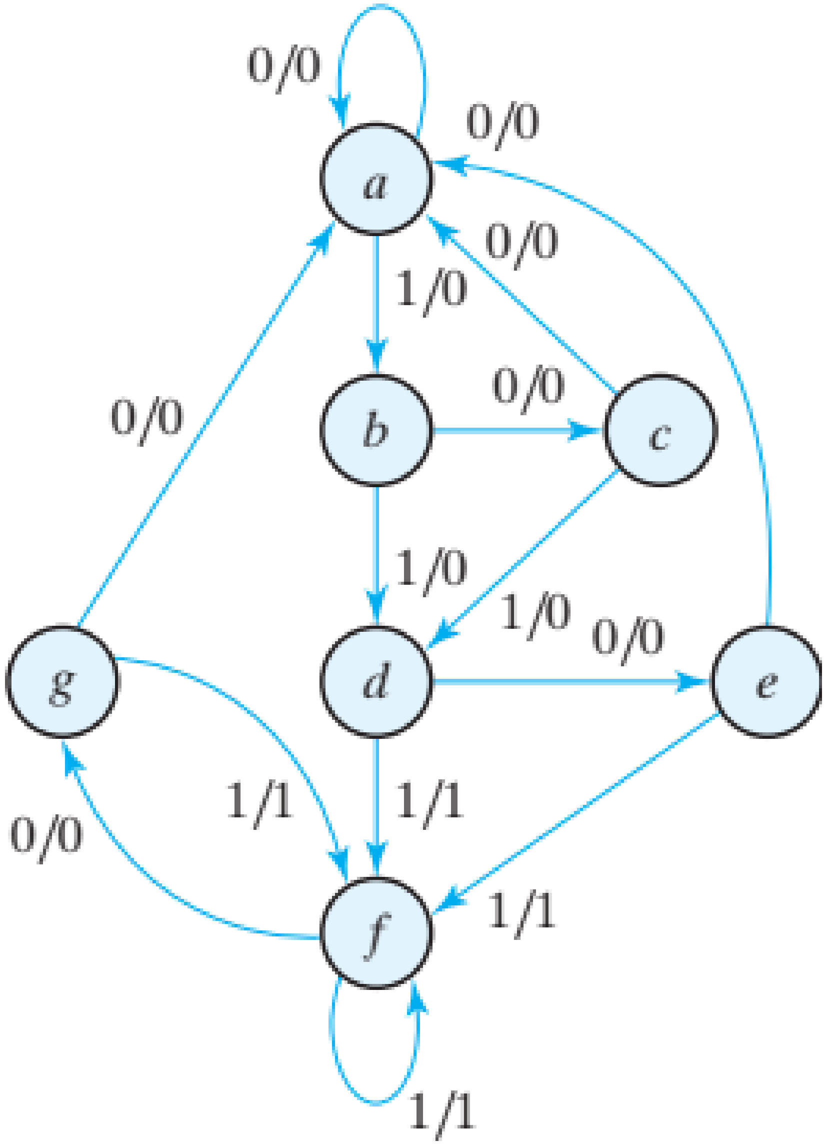


state	*a*	*a*	*b*	*c*	*d*	*e*	*f*	*f*	*g*	*f*	*g*	*a*
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

STATE REDUCTION PROBLEM

- **Equivalence of sequential circuit**
  - If identical input sequences are applied to the two circuits and identical outputs occur for all input sequences, then the two circuits are said to be equivalent

- **Equivalence of states**
  - Two states are said to be equivalent if, for each member of the set of inputs, they give exactly the same output and send the circuit either to the same state or to an equivalent state



STATE REDUCTION PROBLEM

- **Equivalence of states**
  - Two states are said to be equivalent if, for each member of the set of inputs, they give exactly the same output and send the circuit either to the same state or to an equivalent state

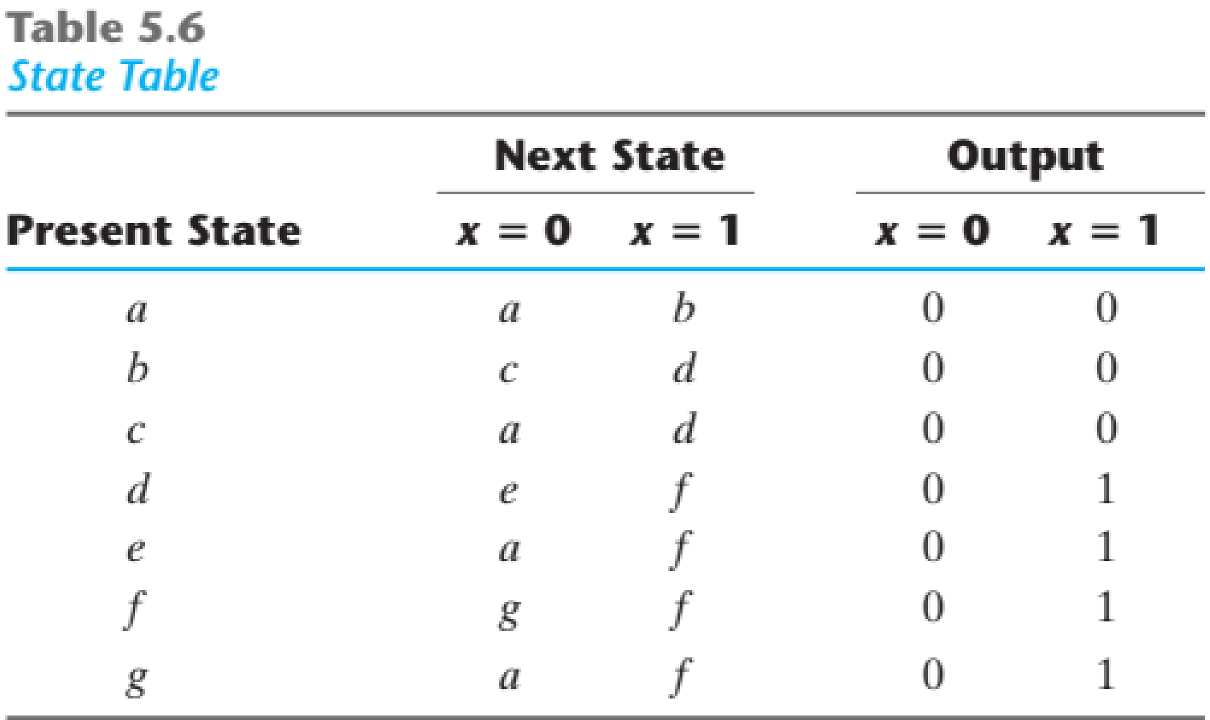
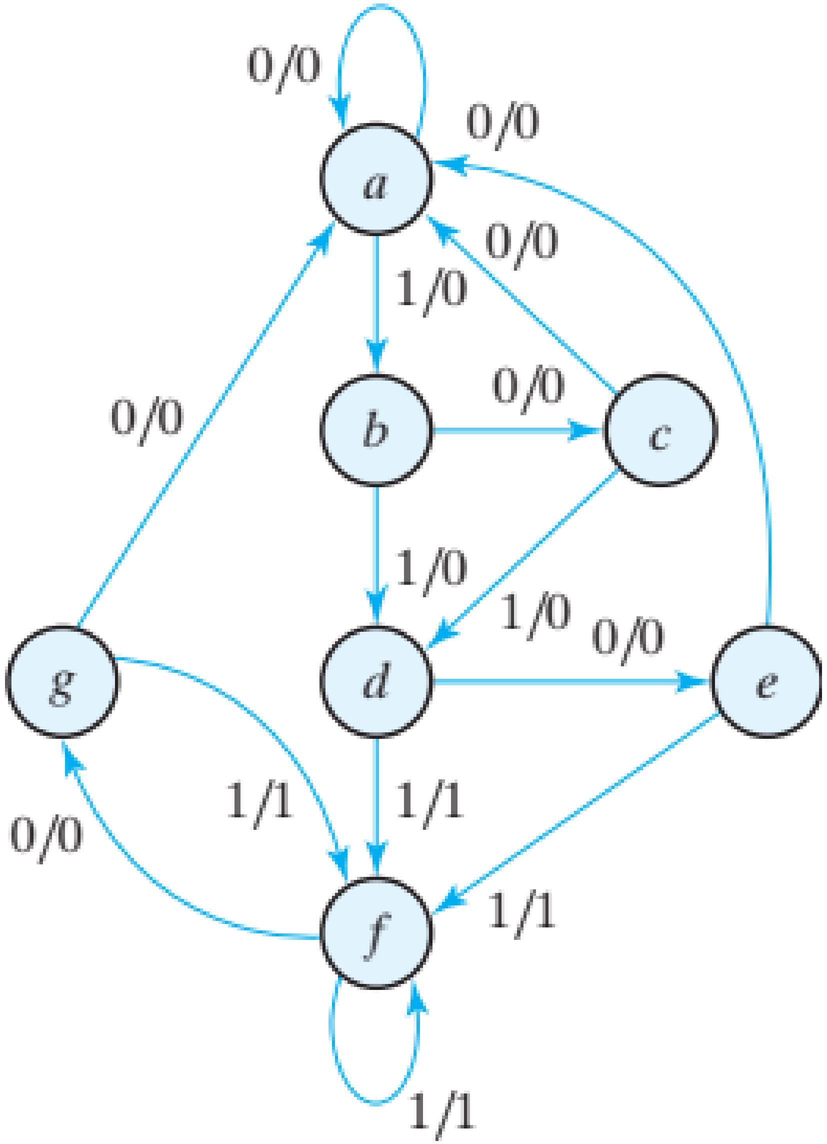


Table 5.6
State Table

	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	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1



STATE REDUCTION

- **Equivalence of states**
  - State ***e*** & ***g*** are equivalent
  - They both go to states a and f and have outputs of $0\ and\ 1\ for\ x\ =\ 0\ and\ x\ =\ 1$
  - Replace ***g by e***, and remove the last row

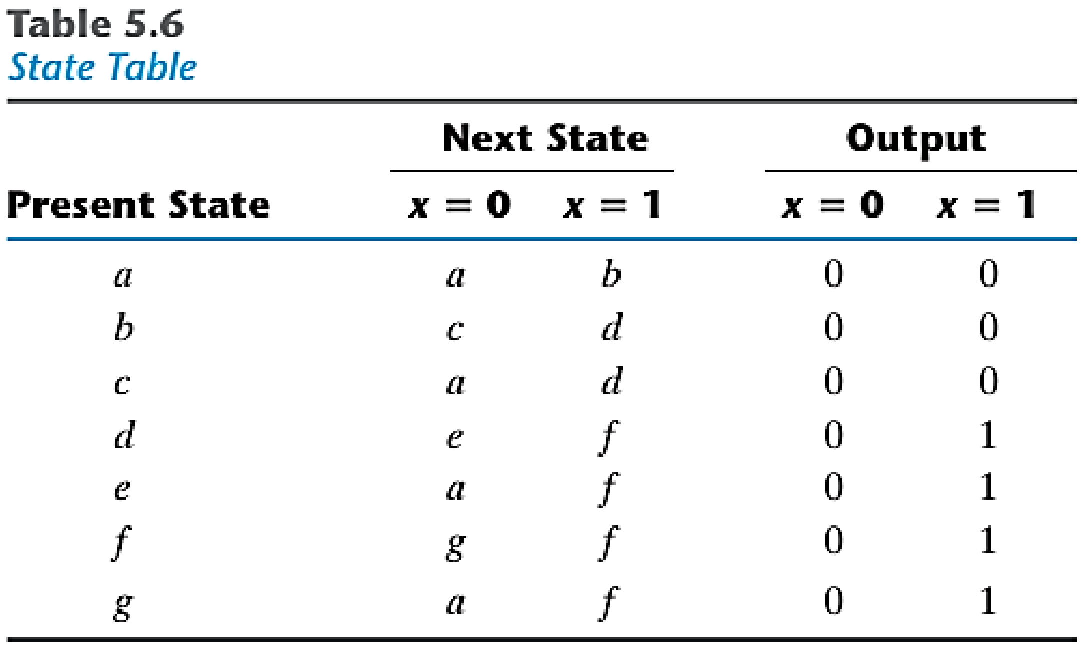


Table 5.6
State Table

	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	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

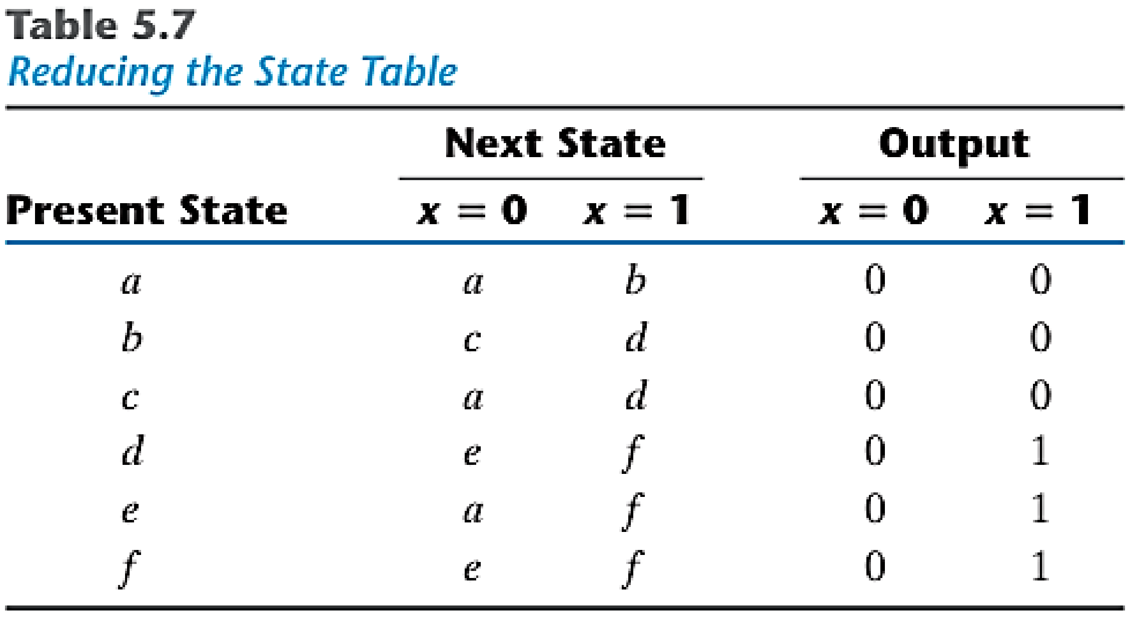


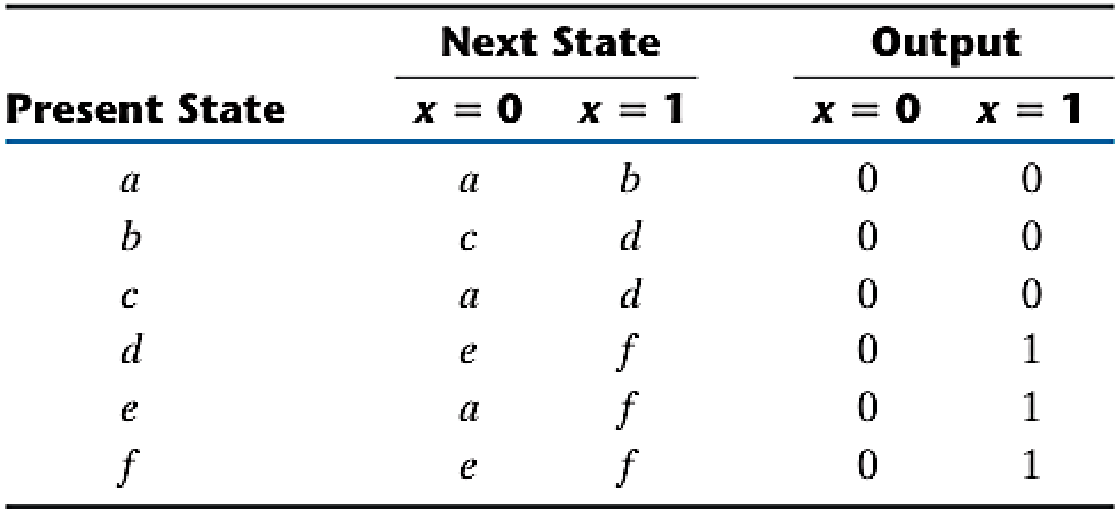
Table 5.7
Reducing the State Table

	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	f	0	1
e	a	f	0	1
f	e	f	0	1

STATE REDUCTION

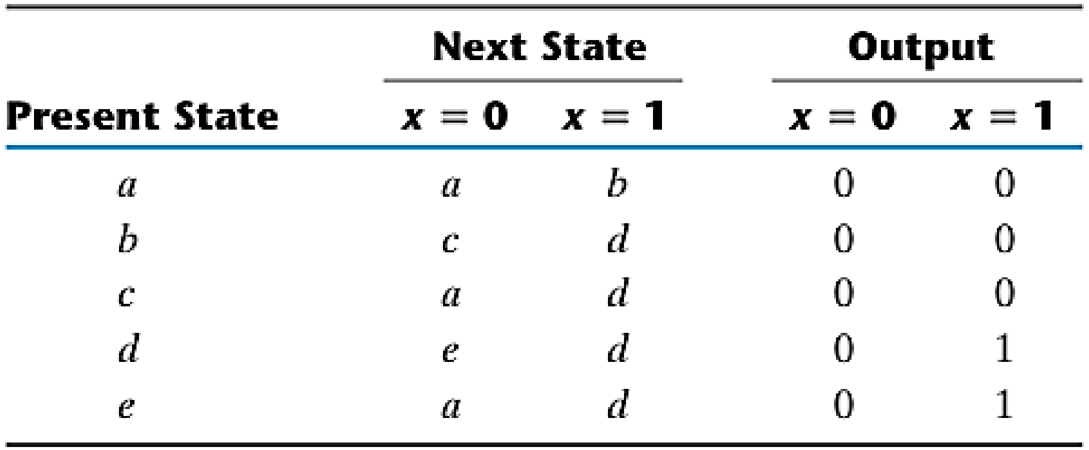
- **Equivalence of states**
  - State $\boldsymbol{d}$ & $\boldsymbol{f}$ are equivalent
  - They both go to states e and f and have outputs of 0 and 1 $for\ x\ =\ 0\ and\ x\ =\ 1$
  - Replace $\boldsymbol{f\ by\ d}$, and remove the last row

Table 5.7
Reducing the 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*	*f*	0	1
e	*a*	*f*	0	1
f	*e*	*f*	0	1

Table 5.8
Reduced State Table

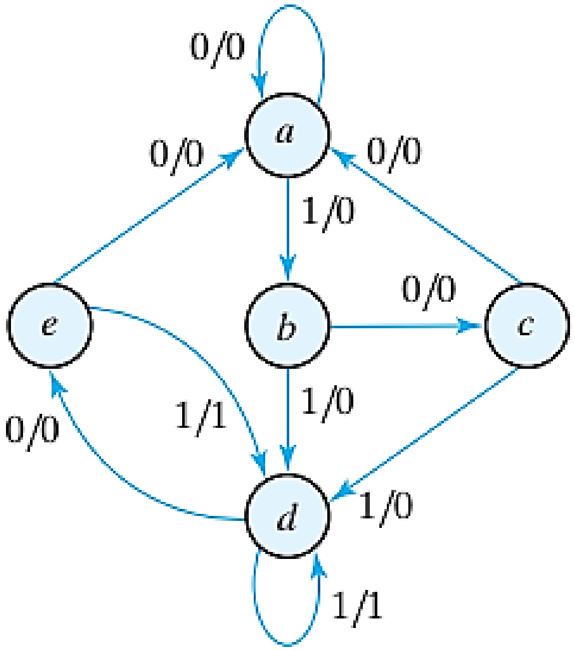


	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	*a*	*b*	0	0
b	*c*	*d*	0	0
c	*a*	*d*	0	0
d	*e*	*d*	0	1
e	*a*	*d*	0	1

STATE REDUCTION

Final result

- Reduction of states from 7 to 5
- Satisfies the original input-output specification
- Produces the same output sequence for the given input sequence
- Doesn’t guarantee reduction of flip-flops



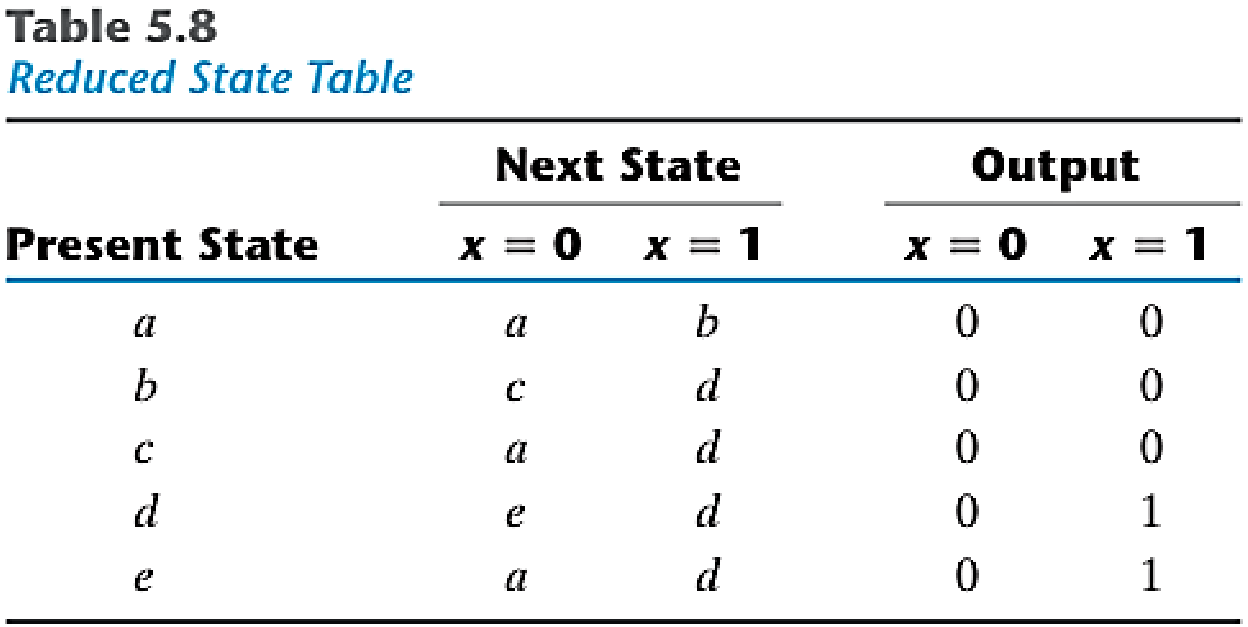


Table 5.8
Reduced State Table

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*	*d*	0	1
e	*a*	*d*	0	1

STATE ASSIGNMENT

- State should be numbered in binary format
- **m** states requires **n** bits, where $\mathbf{2^n \geq m}$
- Unused states are treated as don’t-care conditions during the design

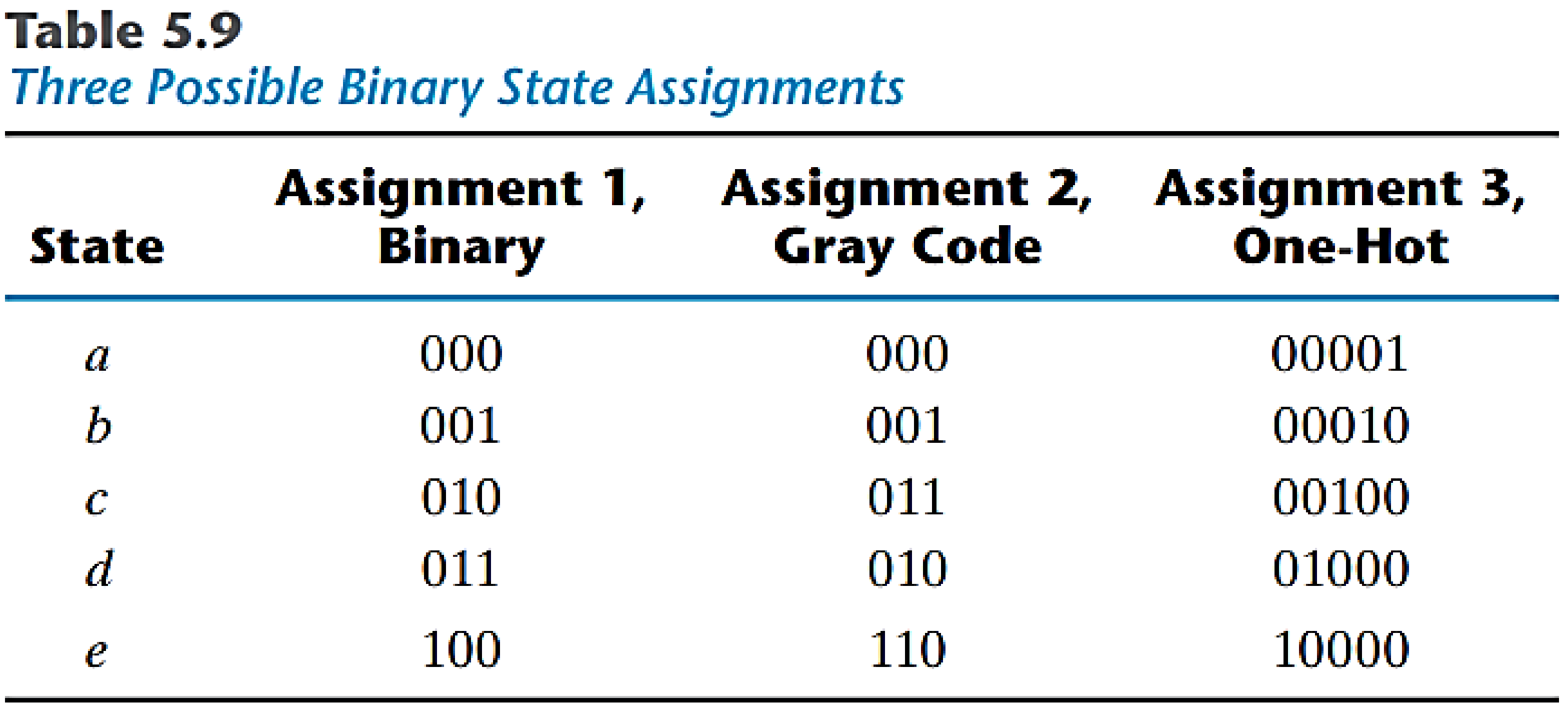


Table 5.9

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

- Binary assignment code (aka transition table)

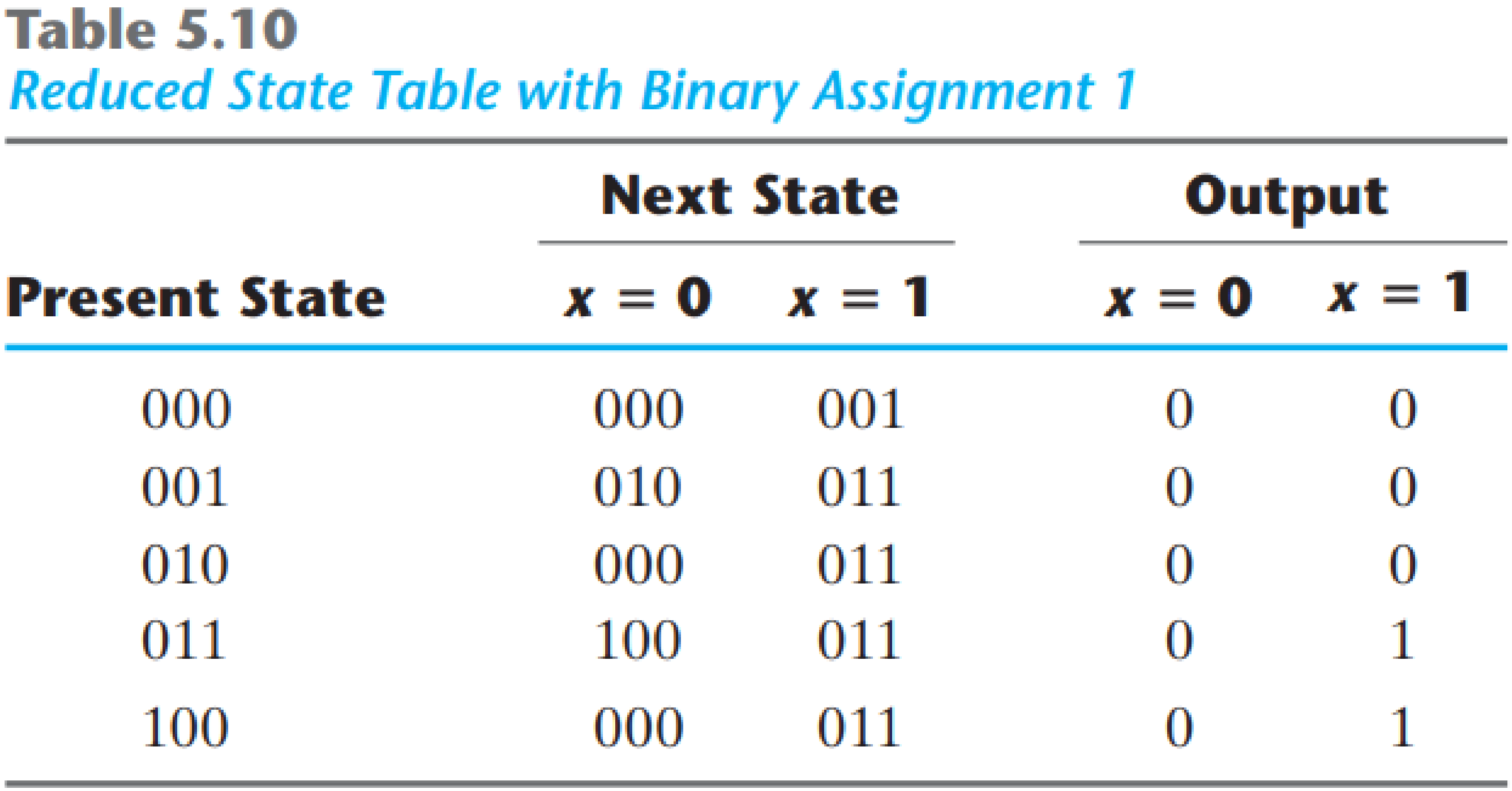


Table 5.10
Reduced State Table with Binary Assignment 1

	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

SEQUENTIAL CIRCUIT DESIGN

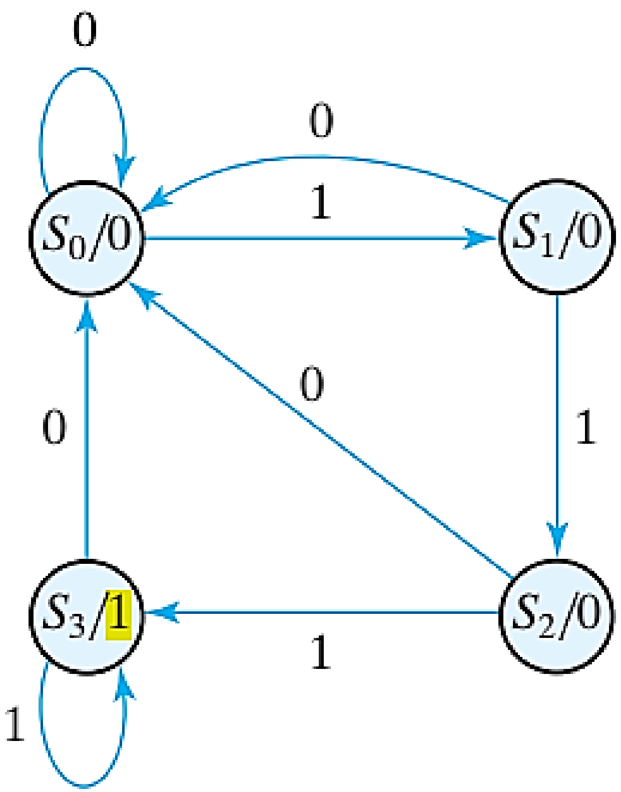
- **Procedure**

1. From the word description and specifications of the desired operation, derive a state diagram for the circuit
2. Reduce the number of states if necessary
3. Assign binary values to the states
4. Obtain the binary-coded state table
5. Choose the type of flip-flops to be used
6. Derive the simplified flip-flop input equations and output equations
7. Draw the logic diagram

SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - a circuit that detects a sequence of three or more consecutive 1’s in a string of bits coming through an input line

- **Specs**
  - Start with state $\mathbf{S_0}$
  - Move to next state on **1**
  - Move to $\mathbf{S_0}$ on **0**
  - Mealy or Moore?



SEQUENTIAL CIRCUIT DESIGN

- **Synthesis**
  - With **D** flip-flop for simplicity because $Q(t + 1) = D_Q$

- **Steps**
  - Assign binary codes to states
  - Obtain state table
  - Choose the type of flip-flops to be used

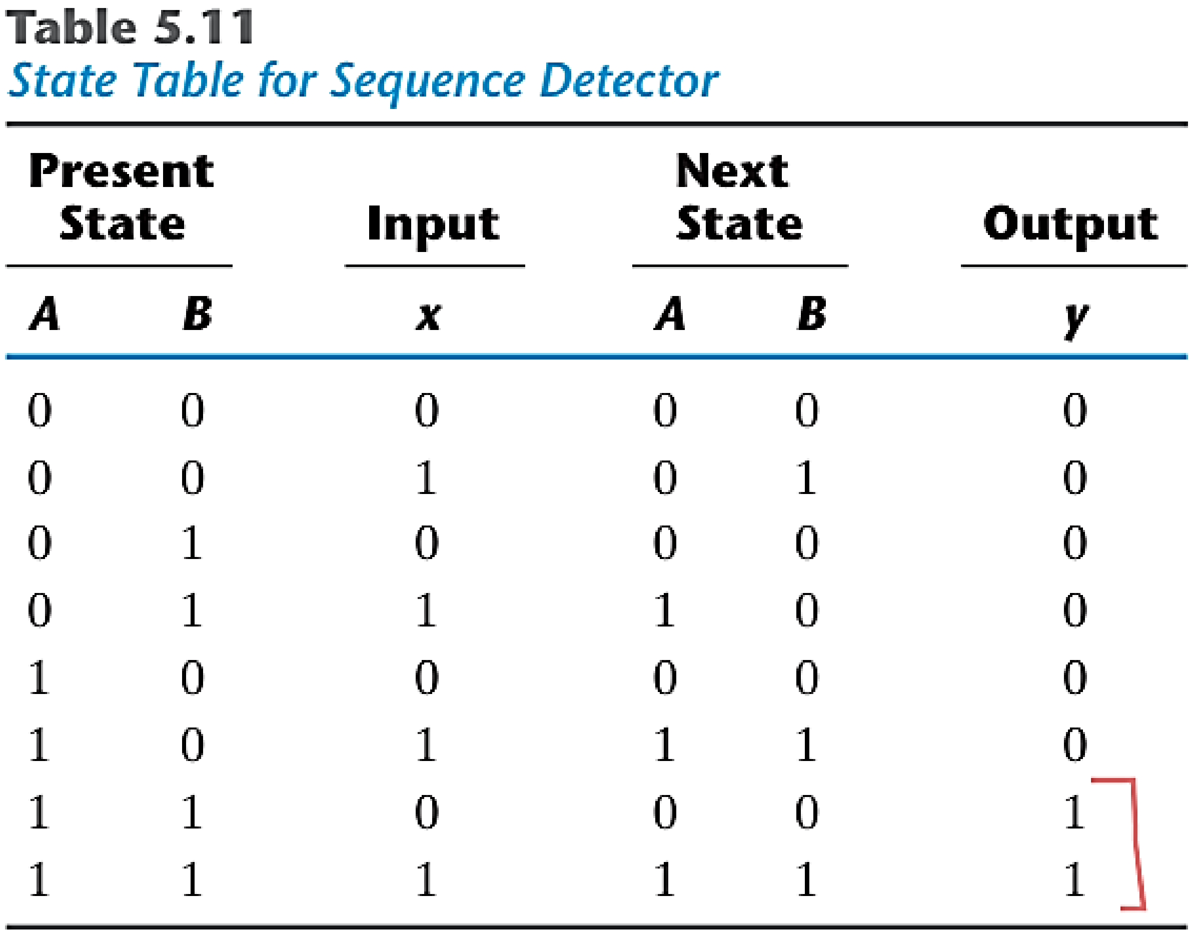


Table 5.11
State Table for Sequence Detector

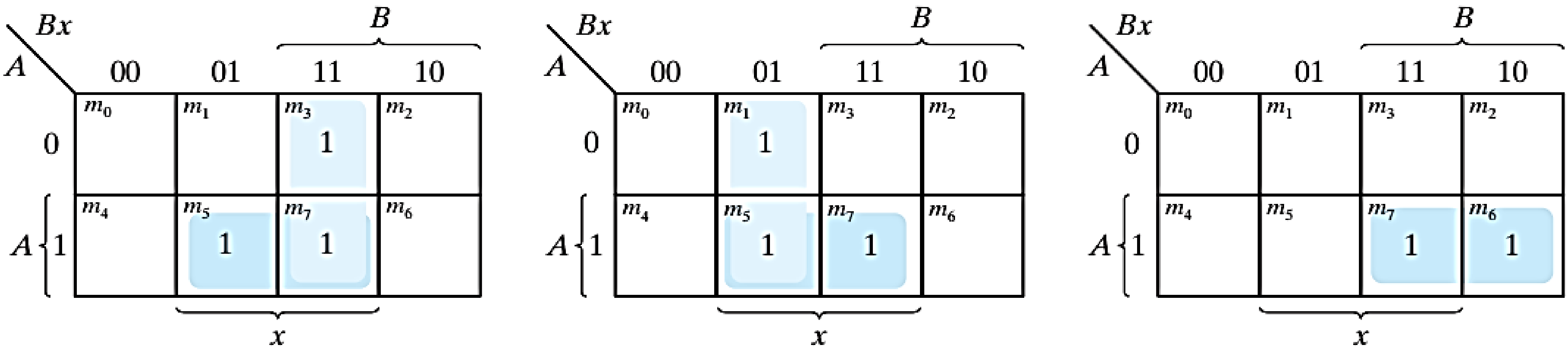
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

SEQUENTIAL CIRCUIT DESIGN

- Synthesis
- Steps
  - Derive the simplified flip-flop input equations and output equations

Table 5.11
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



$D_A = Ax + Bx$

$D_B = Ax + B'x$

$y = AB$

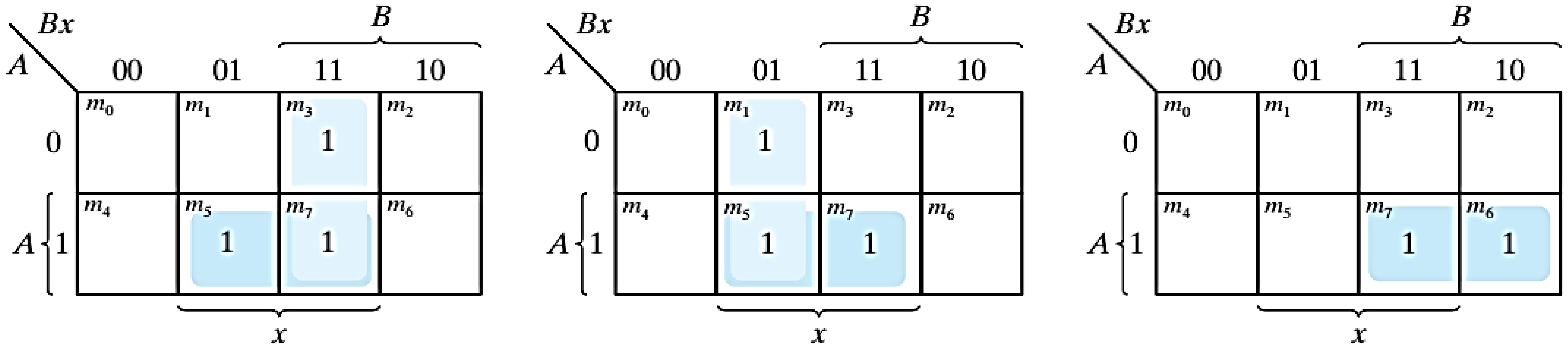
SEQUENTIAL CIRCUIT DESIGN

- **Synthesis**
- **Steps**
  - Derive the simplified flip-flop input equations and output equations

$$A(t+1) = D_A(A, B, x) = \Sigma(3, 5, 7)$$

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

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



$D_A = Ax + Bx$

$D_B = Ax + B'x$

$y = AB$

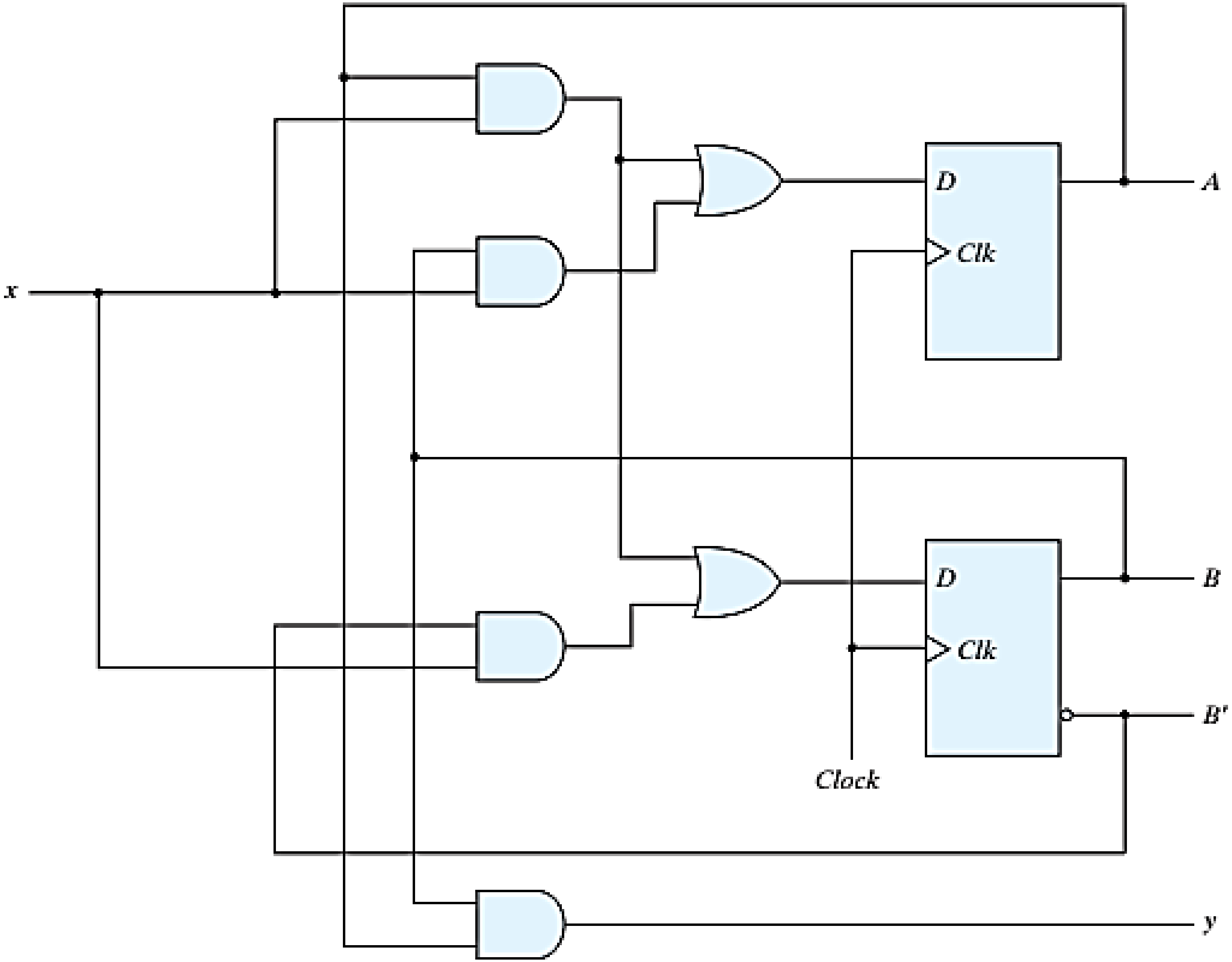
SEQUENTIAL CIRCUIT DESIGN

- **Synthesis**
- **Steps**
  - Draw logic diagram

$$D_A = Ax + Bx$$

$$D_B = Ax + B'x$$

$$y = AB$$



SEQUENTIAL CIRCUIT DESIGN

- **Synthesis**
  - Straight forward with D flip-flop, **BUT**
  - Complicated with other types of flips-flops
    - SR
    - JK
    - T
    - PN
  - Because input equations for the circuit must be derived **indirectly** from the state table
  - As such, requires the use of **excitation table**

SEQUENTIAL CIRCUIT DESIGN

- **Excitation table (revisited)**

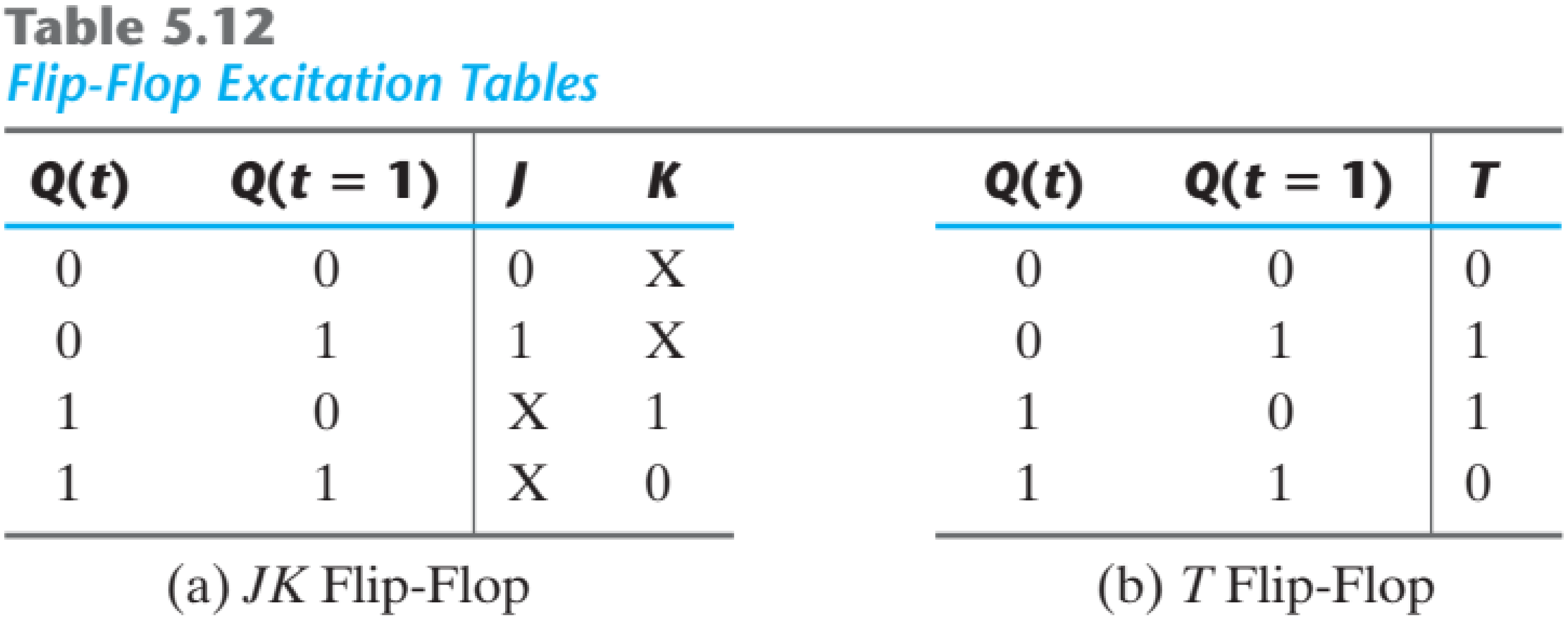


Table 5.12

Flip-Flop Excitation Tables

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

(a) *JK* Flip-Flop

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

(b) *T* Flip-Flop

SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - a sequential circuit from the following state table using JK

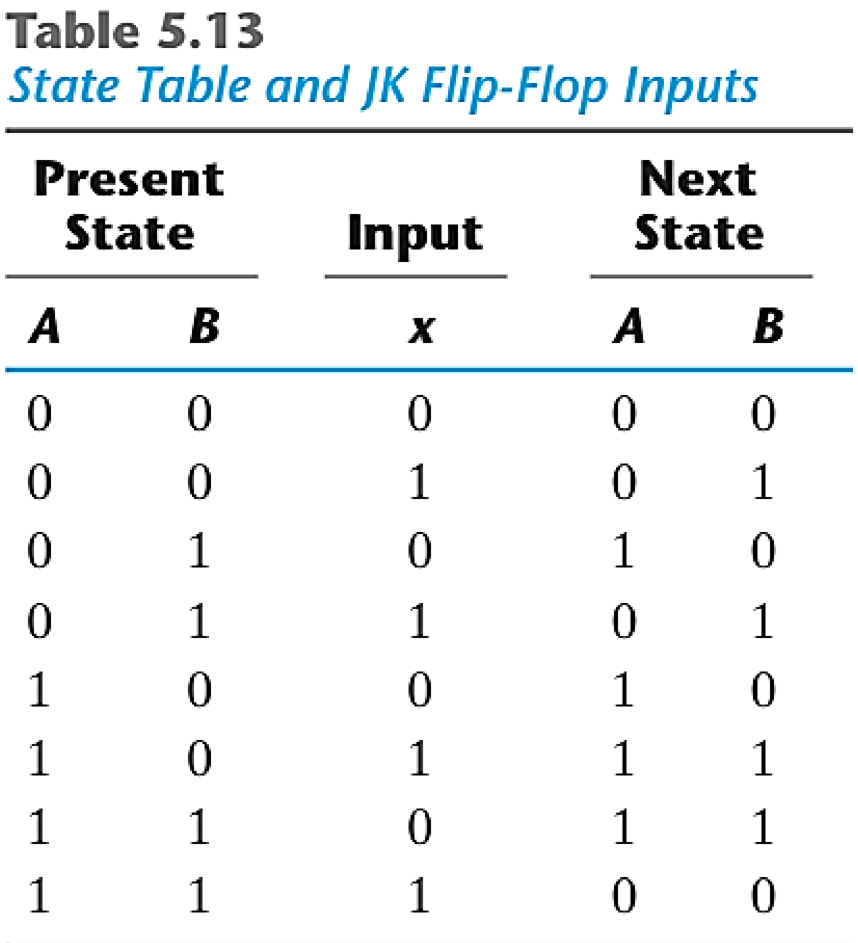
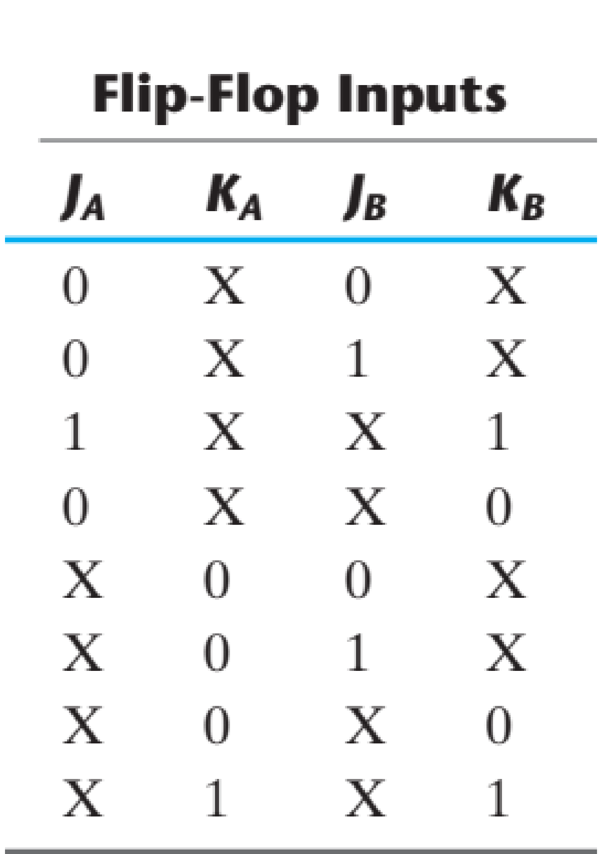


Table 5.13
State Table and JK Flip-Flop Inputs

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

Excitation table

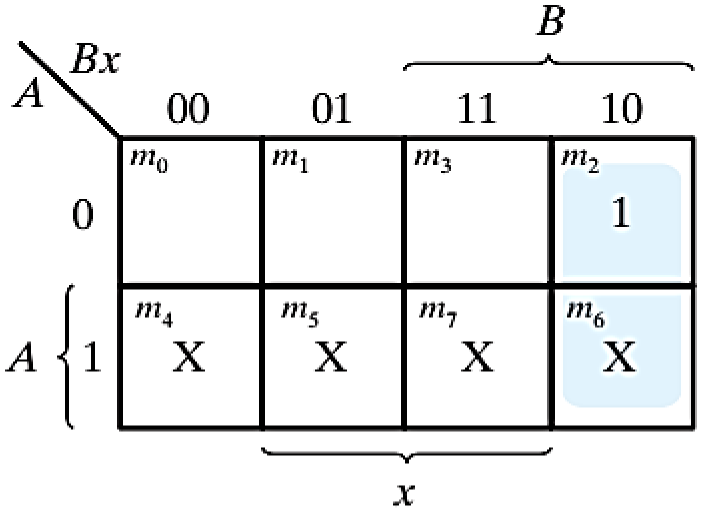


Flip-Flop Inputs			
J_A	K_A	J_B	K_B
0	X	0	X
0	X	1	X
1	X	X	1
0	X	X	0
X	0	0	X
X	0	1	X
X	0	X	0
X	1	X	1

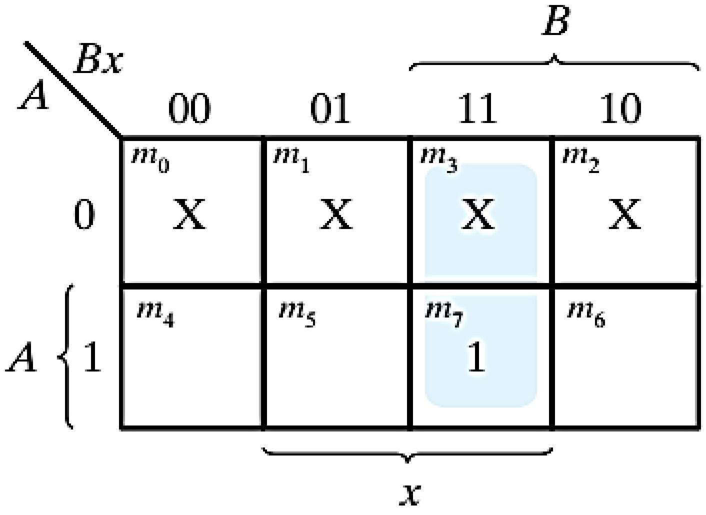
SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - Input equation

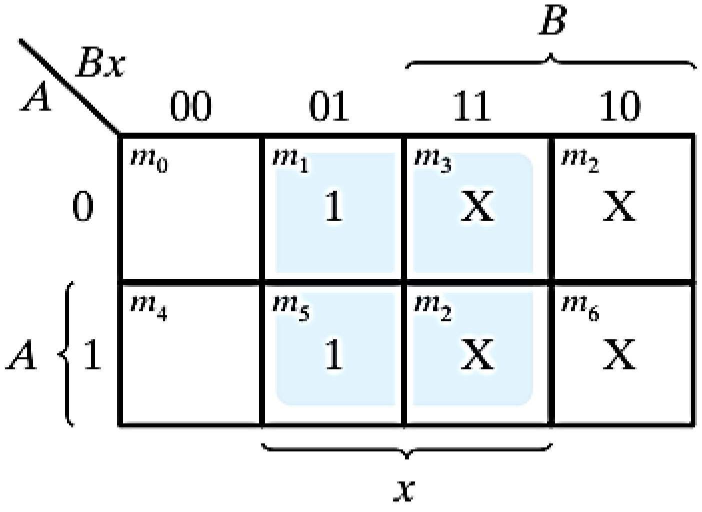
Flip-Flop Inputs			
J_A	K_A	J_B	K_B
0	X	0	X
0	X	1	X
1	X	X	1
0	X	X	0
X	0	0	X
X	0	1	X
X	0	X	0
X	1	X	1



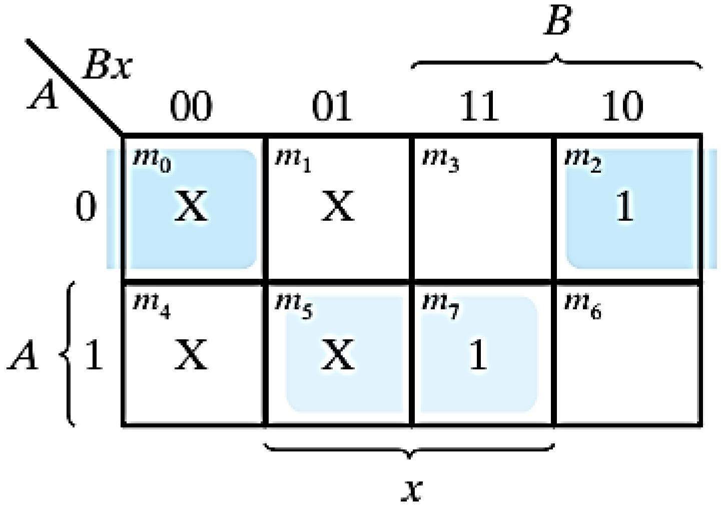
$J_A = Bx'$



$K_A = Bx$



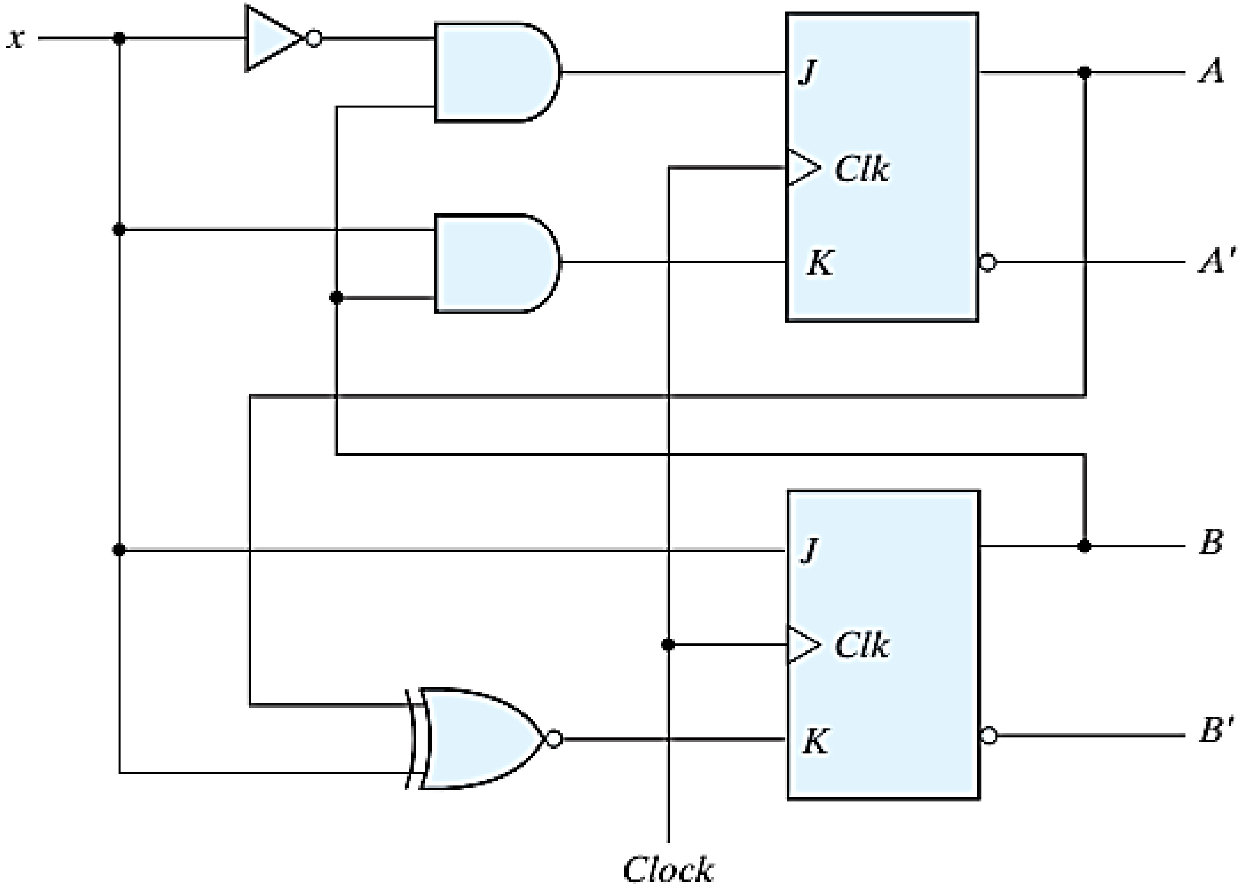
$J_B = x$



$K_B = (A \oplus x)'$

SEQUENTIAL CIRCUIT DESIGN

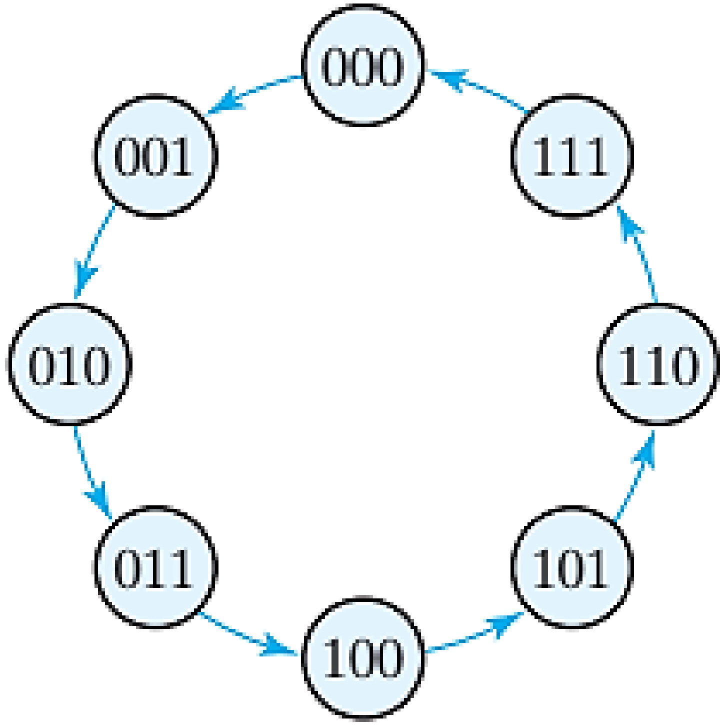
- **Advantage of JK**
  - So many Don’t care condition
    - Combinational circuits are likely to be simpler



SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - a binary counter (0 → 7) using T flip-flop
  - How many bits?

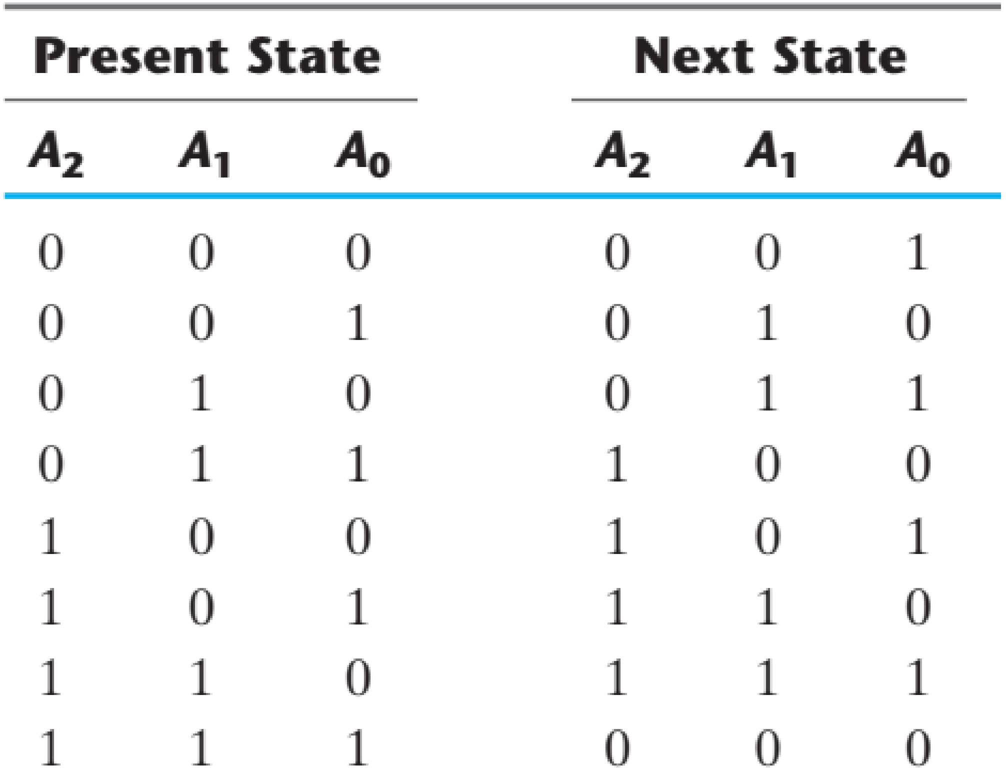
- **Specs**
  - The only input to the circuit is the clock
  - Counter moves based on clock, not input
  - Next state depends entirely on present state
  - Moore machine



SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - State table + excitation table

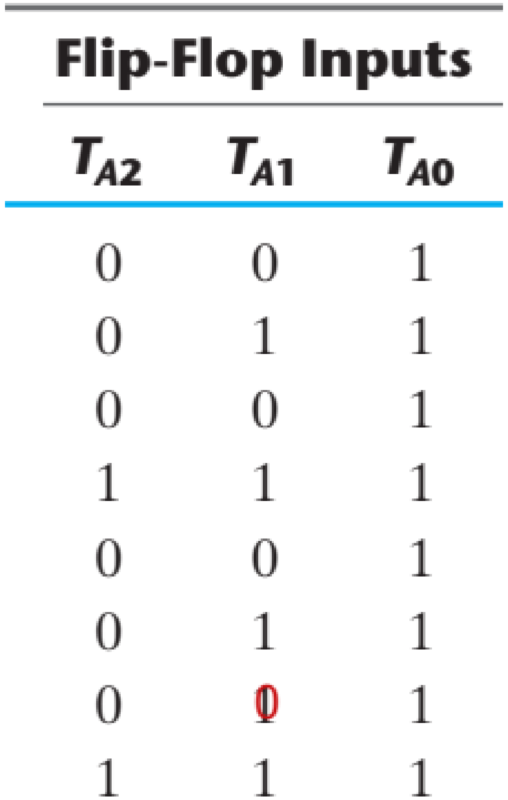
Table 5.14
State Table for Three-Bit Counter



Present State			Next State		
A_2	A_1	A_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

+

Excitation table

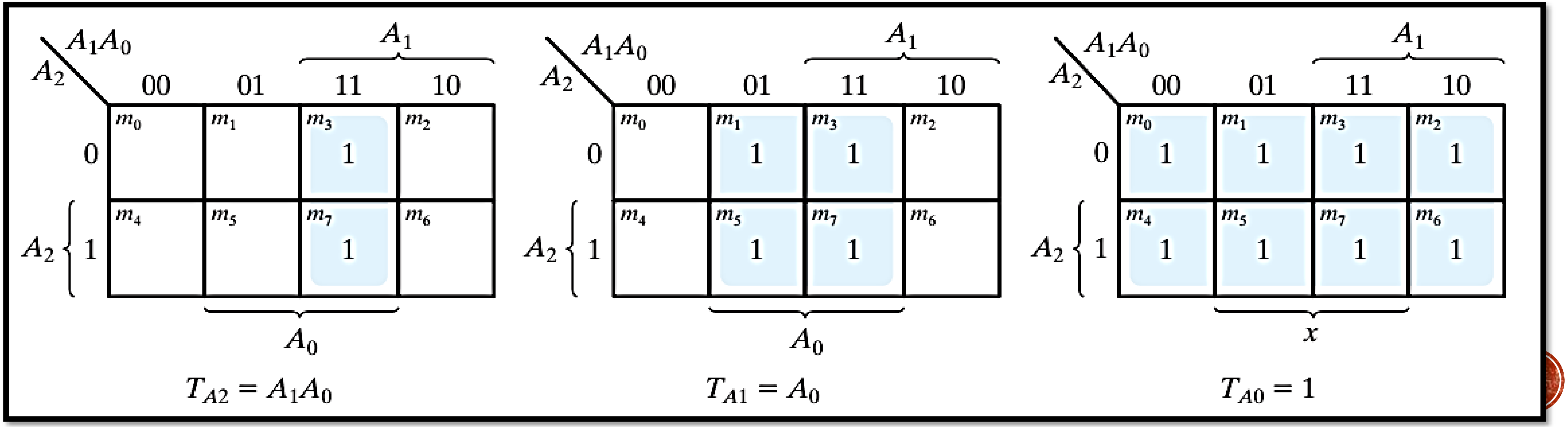


Flip-Flop Inputs		
T_{A2}	T_{A1}	T_{A0}
0	0	1
0	1	1
0	0	1
1	1	1
0	0	1
0	1	1
0	0	1
1	1	1

SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - Input equations

Flip-Flop Inputs		
T_{A2}	T_{A1}	T_{A0}
0	0	1
0	1	1
0	0	1
1	1	1
0	0	1
0	1	1
0	~~1~~ 0	1
1	1	1



$T_{A2} = A_1A_0$

$T_{A1} = A_0$

$T_{A0} = 1$

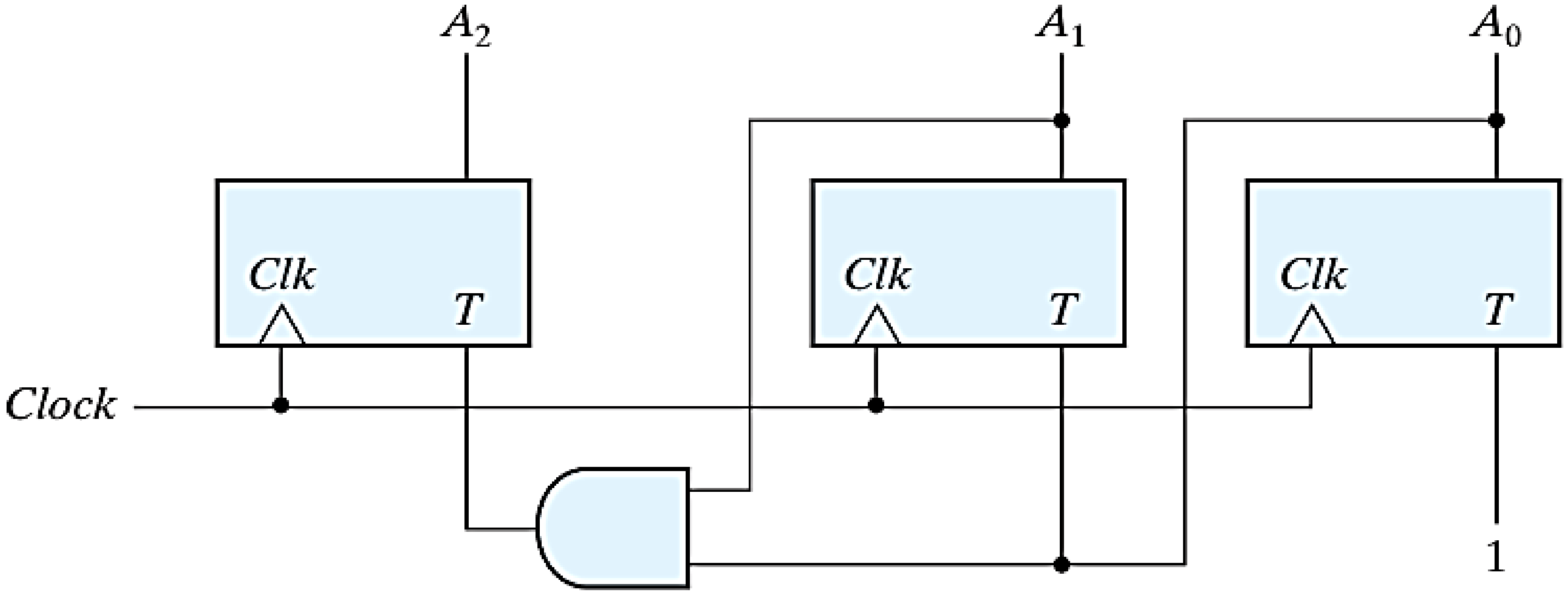
SEQUENTIAL CIRCUIT DESIGN

- **Design**
  - Logic diagram

$$T_{A2} = A_1A_0$$

$$T_{A1} = A_0$$

$$T_{A0} = 1$$



LECTURE 6

Problems

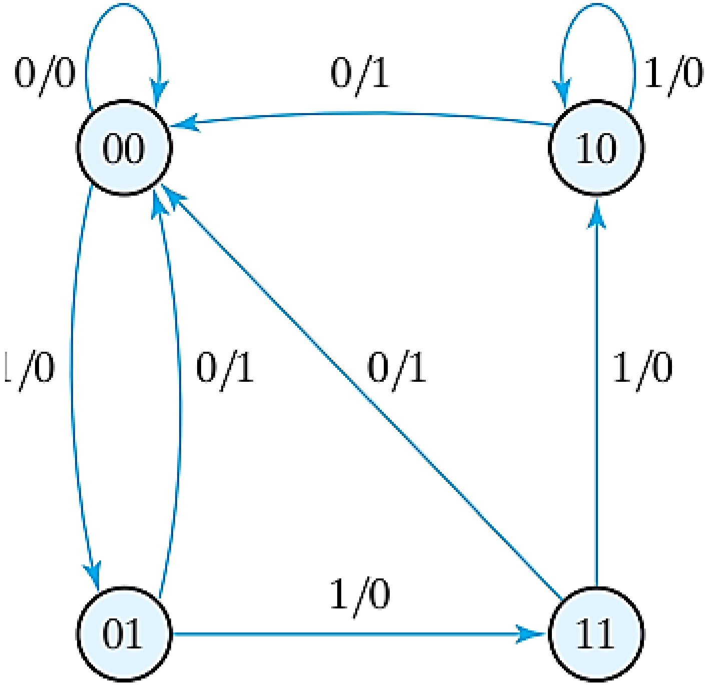
PROBLEMS

5.11 For the circuit described by the state diagram of Fig. 5.16,

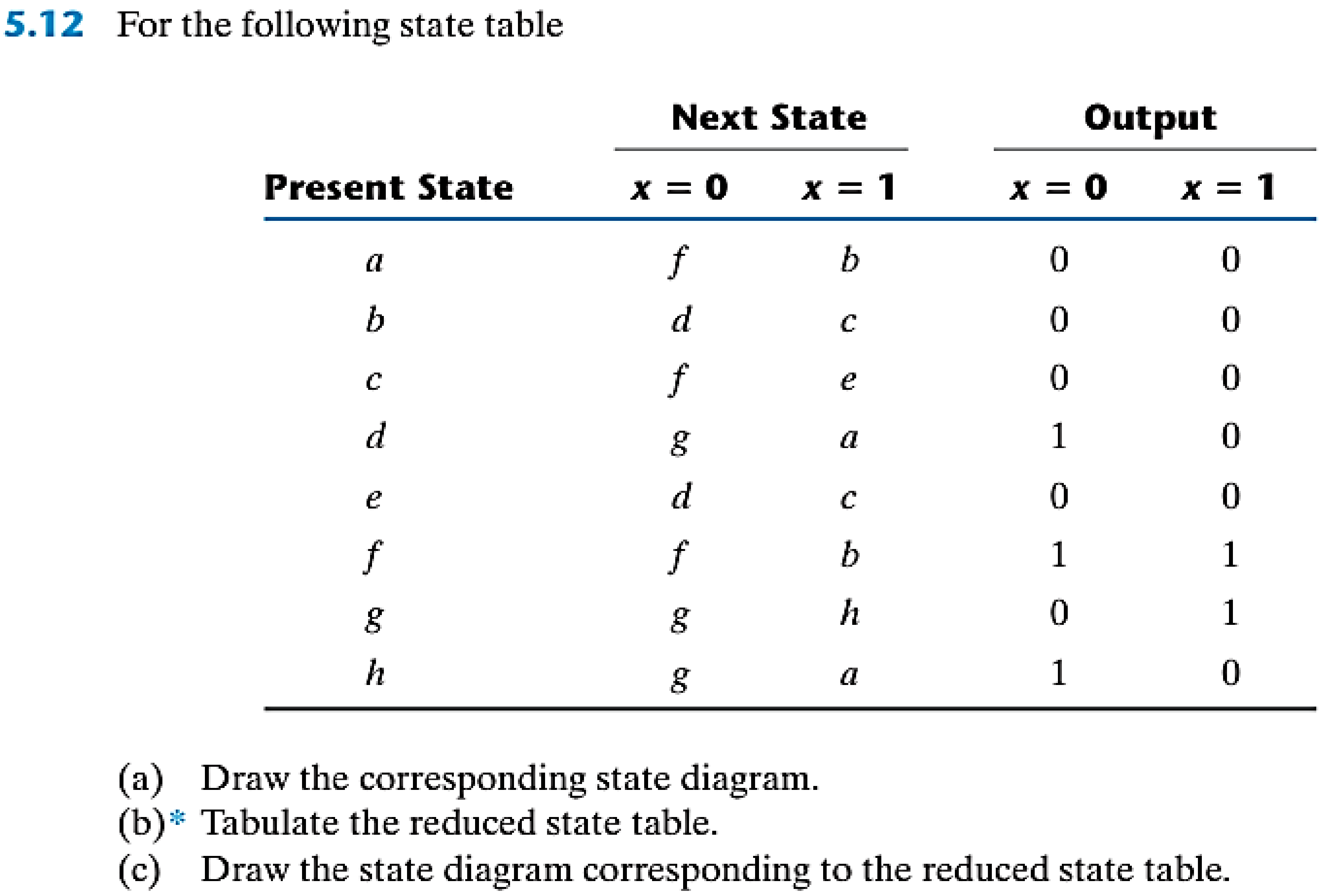
(a)* Determine the state transitions and output sequence that will be generated when an input sequence of 010110111011110 is applied to the circuit and it is initially in the state 00.

(b) Find all of the equivalent states in Fig. 5.16 and draw a simpler, but equivalent, state diagram.

(c) Using *D* flip-flops, design the equivalent machine (including its logic diagram) described by the state diagram in (b).



PROBLEMS

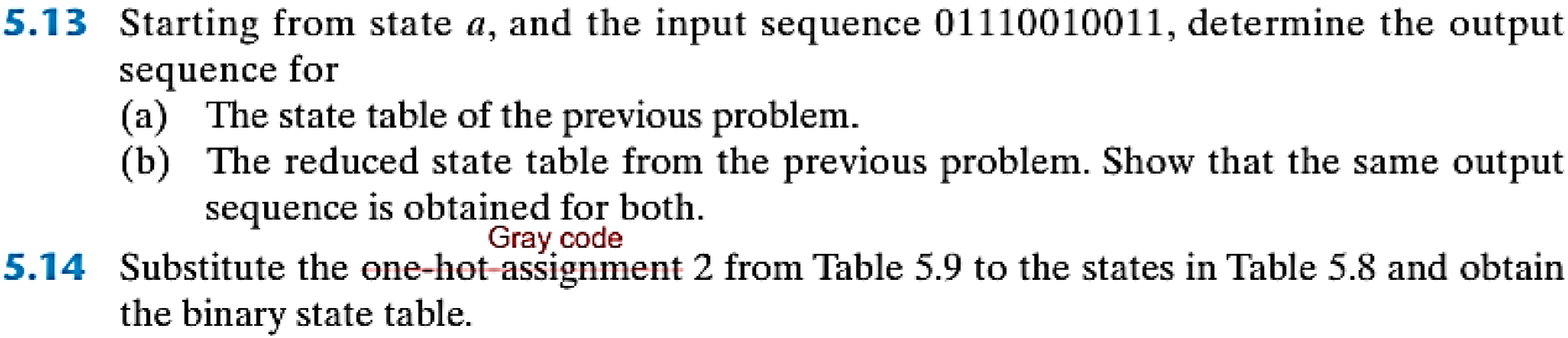


5.12 For the following state table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
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	h	0	1
h	g	a	1	0

(a) Draw the corresponding state diagram.
(b)* Tabulate the reduced state table.
(c) Draw the state diagram corresponding to the reduced state table.

PROBLEMS



5.13 Starting from state *a*, and the input sequence 01110010011, determine the output sequence for

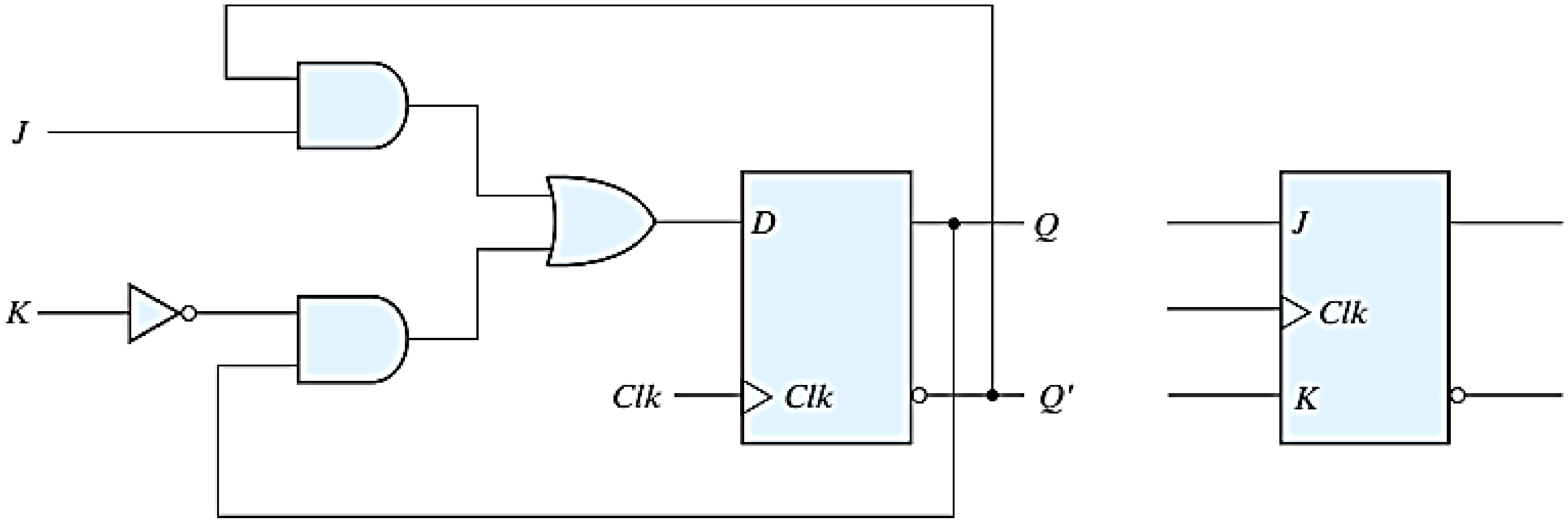
(a) The state table of the previous problem.

(b) The reduced state table from the previous problem. Show that the same output sequence is obtained for both.

5.14 Substitute the ~~one-hot assignment~~ Gray code 2 from Table 5.9 to the states in Table 5.8 and obtain the binary state table.

PROBLEMS

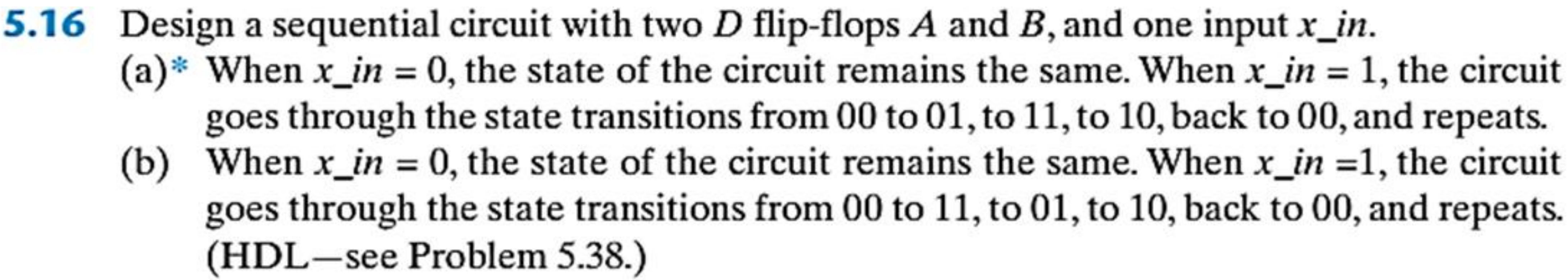
5.15 List a state table for the *JK* flip-flop using *Q* as the present and next state and *J* and *K* as inputs. Design the sequential circuit specified by the state table and show that it is equivalent to Fig. 5.12(a).



(a) Circuit diagram

(b) Graphic symbol

PROBLEMS

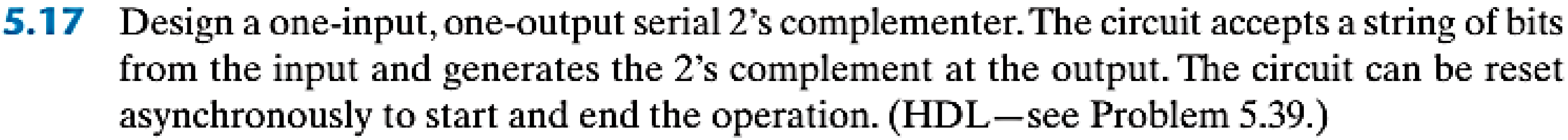


5.16 Design a sequential circuit with two *D* flip-flops *A* and *B*, and one input *x_in*.

(a)* When *x_in* = 0, the state of the circuit remains the same. When *x_in* = 1, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.

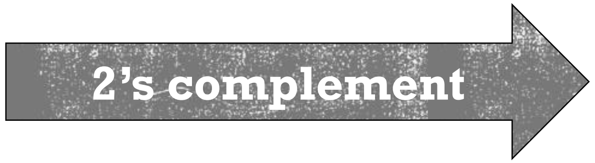
(b) When *x_in* = 0, the state of the circuit remains the same. When *x_in* =1, the circuit goes through the state transitions from 00 to 11, to 01, to 10, back to 00, and repeats. (HDL—see Problem 5.38.)

PROBLEMS



5.17 Design a one-input, one-output serial 2's complementer. The circuit accepts a string of bits from the input and generates the 2's complement at the output. The circuit can be reset asynchronously to start and end the operation. (HDL—see Problem 5.39.)

- **Consider the input sequence**
  - 1 0 1 0 0

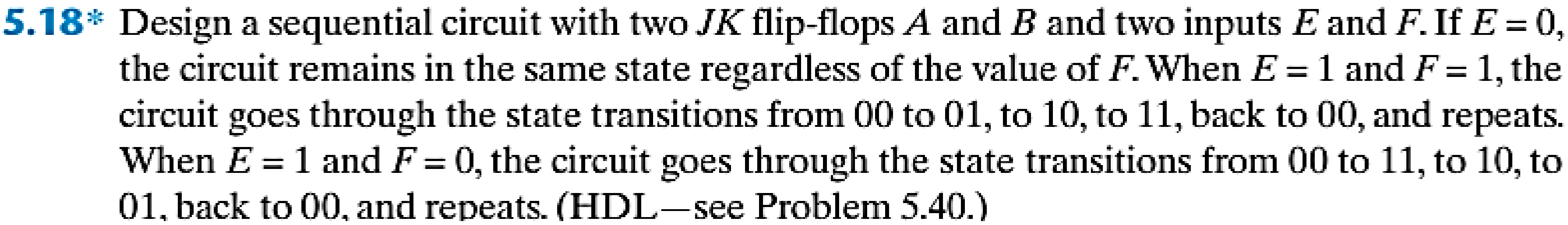


2's complement

0 1 1 0 0 0

- **Observations**
  - The output is 0 for all 0 inputs until the first 1 occurs, at which time the output is 1, after which, the output is the complement of the input.
  - The state diagram has two states. In state 0: **output = input**; in state 1: **output = complement(input)**

PROBLEMS



5.18* Design a sequential circuit with two *JK* flip-flops *A* and *B* and two inputs *E* and *F*. If *E* = 0, the circuit remains in the same state regardless of the value of *F*. When *E* = 1 and *F* = 1, the circuit goes through the state transitions from 00 to 01, to 10, to 11, back to 00, and repeats. When *E* = 1 and *F* = 0, the circuit goes through the state transitions from 00 to 11, to 10, to 01, back to 00, and repeats. (HDL—see Problem 5.40.)

PROBLEMS

5.19 A sequential circuit has three flip-flops *A*, *B*, *C*; one input *x_in*; and one output *y_out*. The state diagram is shown in Fig. P5.19. The circuit is to be designed by treating the unused states as don't-care conditions. Analyze the circuit obtained from the design to determine the effect of the unused states. (HDL—see Problem 5.41.)

(a)* Use *D* flip-flops in the design.
(b) Use *JK* flip-flops in the design.

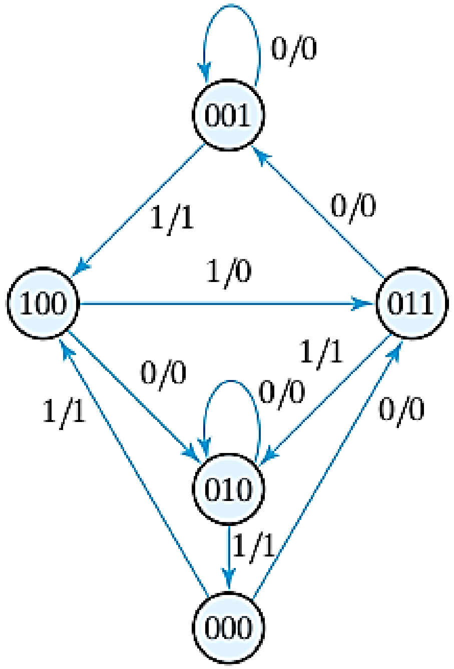
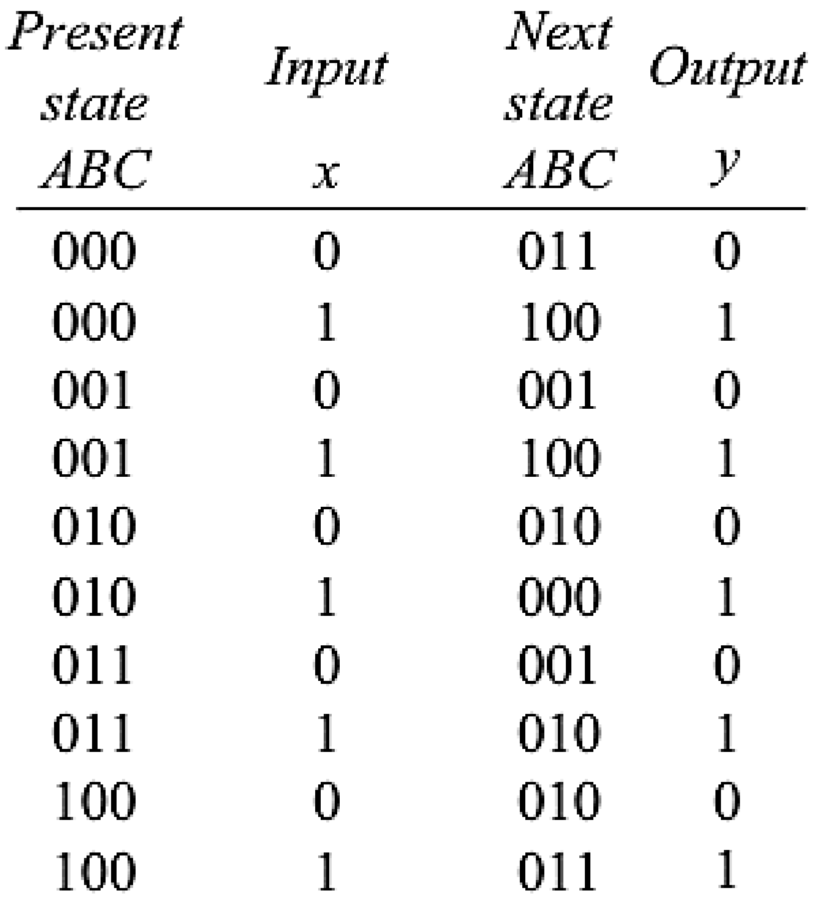


FIGURE P5.19

PROBLEMS

Unused states (see Fig. P5.19): 101, 110, 111.



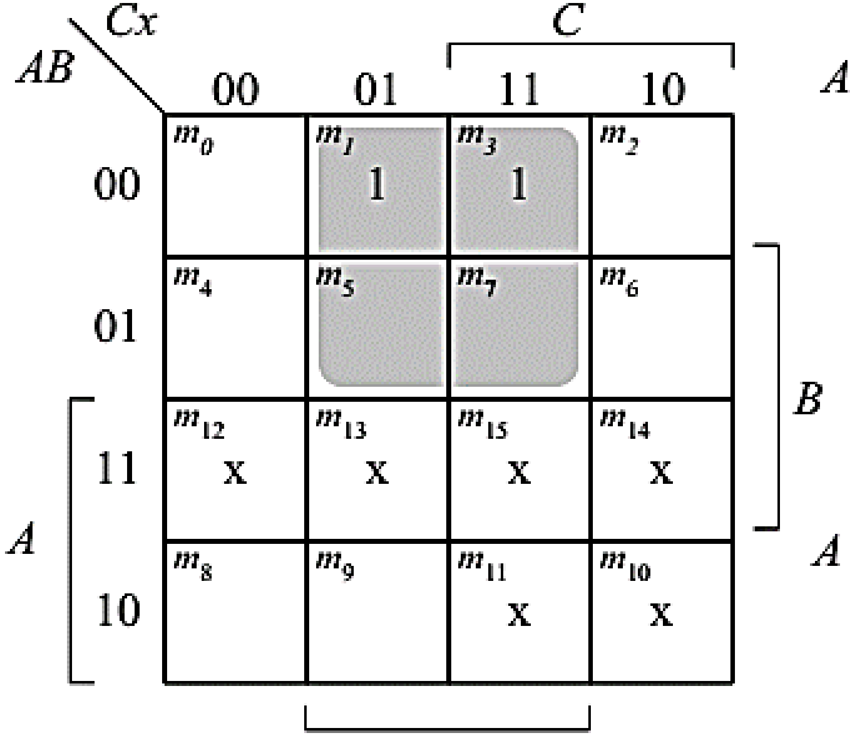
Present *state* *ABC*	*Input* *x*	*Next* *state* *ABC*	*Output* *y*
000	0	011	0
000	1	100	1
001	0	001	0
001	1	100	1
010	0	010	0
010	1	000	1
011	0	001	0
011	1	010	1
100	0	010	0
100	1	011	1

Don’t care

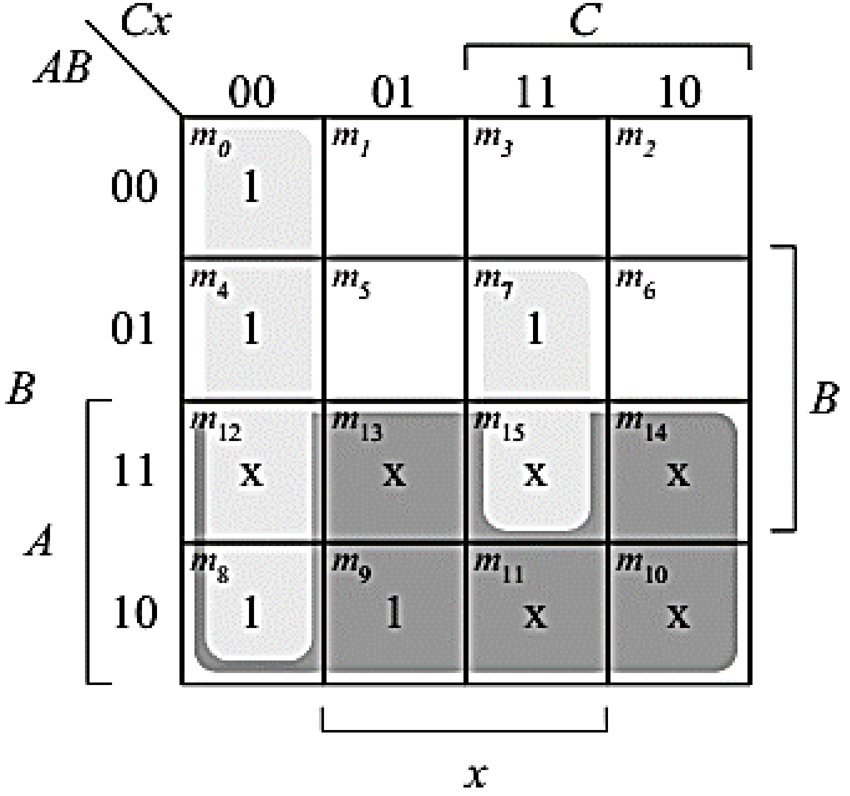
$$d(A, B, C, x) = \Sigma\,(10, 11, 12, 13, 14, 15)$$

PROBLEMS

Present state	*Input*	*Next state*	*Output*
ABC	*x*	*ABC*	*y*
000	0	011	0
000	1	100	1
001	0	001	0
001	1	100	1
010	0	010	0
010	1	000	1
011	0	001	0
011	1	010	1
100	0	010	0
100	1	011	1



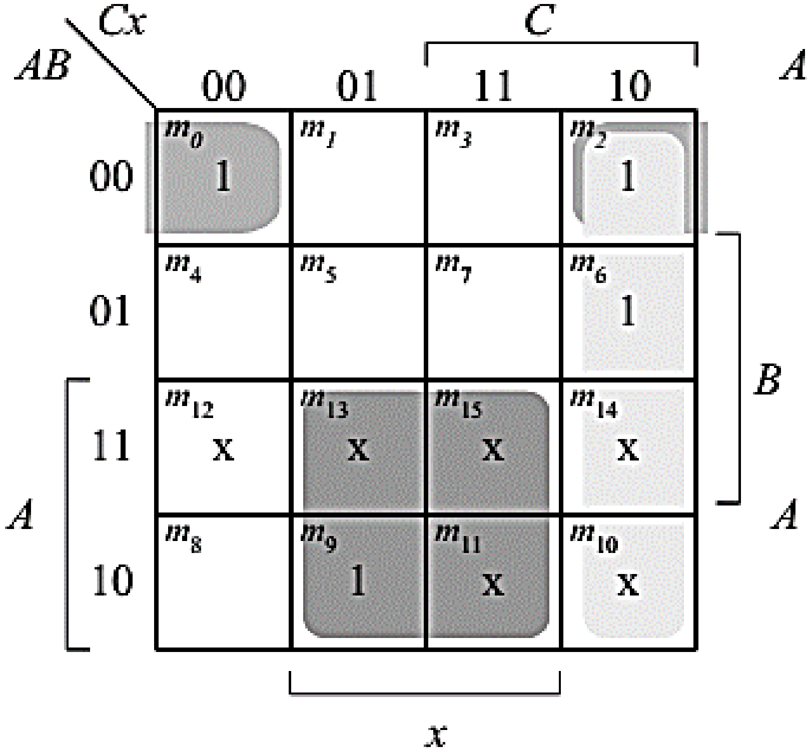
$D_A = A'B'x$



$D_B = A + C'x' + BCx$

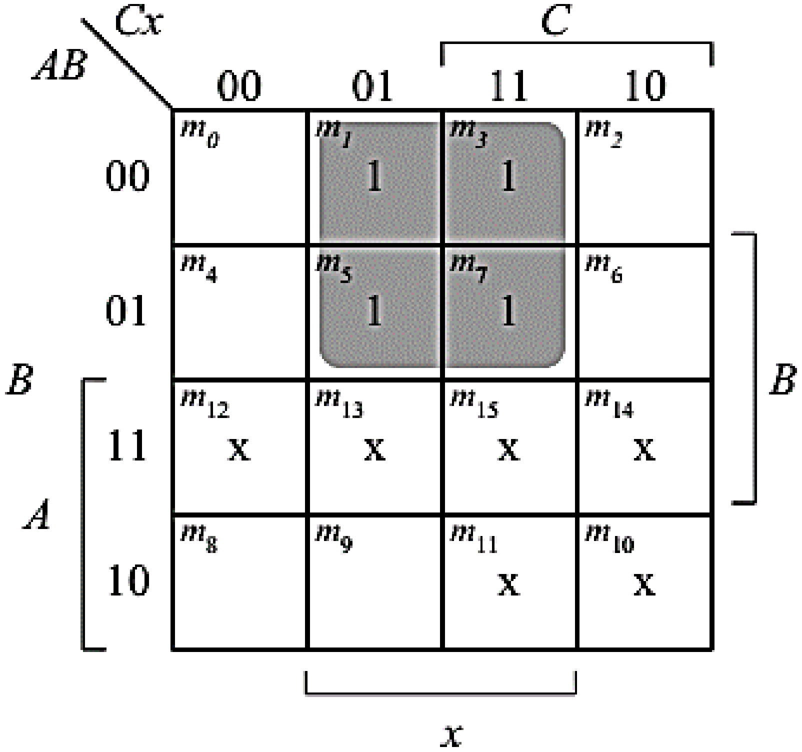
PROBLEMS

Present state *ABC*	*Input* *x*	*Next state* *ABC*	*Output* *y*
000	0	011	0
000	1	100	1
001	0	001	0
001	1	100	1
010	0	010	0
010	1	000	1
011	0	001	0
011	1	010	1
100	0	010	0
100	1	011	1



AB \ Cx	00	01	11	10
00	m_0 1	m_1	m_3	m_2 1
01	m_4	m_5	m_7	m_6 1
11	m_{12} x	m_{13} x	m_{15} x	m_{14} x
10	m_8	m_9 1	m_{11} x	m_{10} x

$D_C = Cx' + Ax + A'B'x'$



AB \ Cx	00	01	11	10
00	m_0	m_1 1	m_3 1	m_2
01	m_4	m_5 1	m_7 1	m_6
11	m_{12} x	m_{13} x	m_{15} x	m_{14} x
10	m_8	m_9	m_{11} x	m_{10} x

$y = A'x$

PROBLEMS

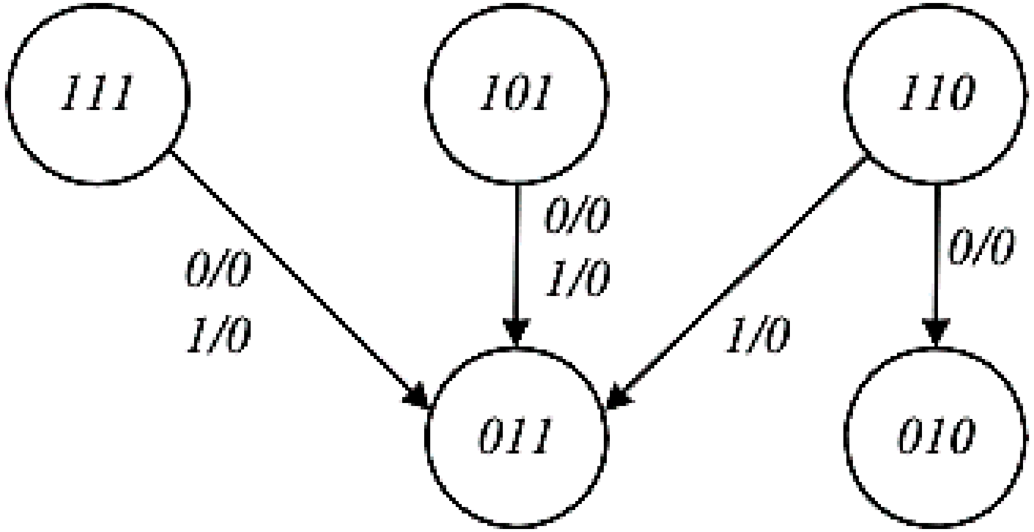
$D_A = A'B'x$

$D_B = A + C'x' + BCx$

$D_C = Cx' + Ax + A'B'x'$

$y = A'x$

The machine is self-correcting, i.e., the unused states transition to known states.



111
101
110
0/0
1/0
0/0
1/0
1/0
0/0
011
010