CHAPTER 5
SEQUENTIAL LOGIC

INEL 4205 LOGIC CIRCUITS

SPRING 2008

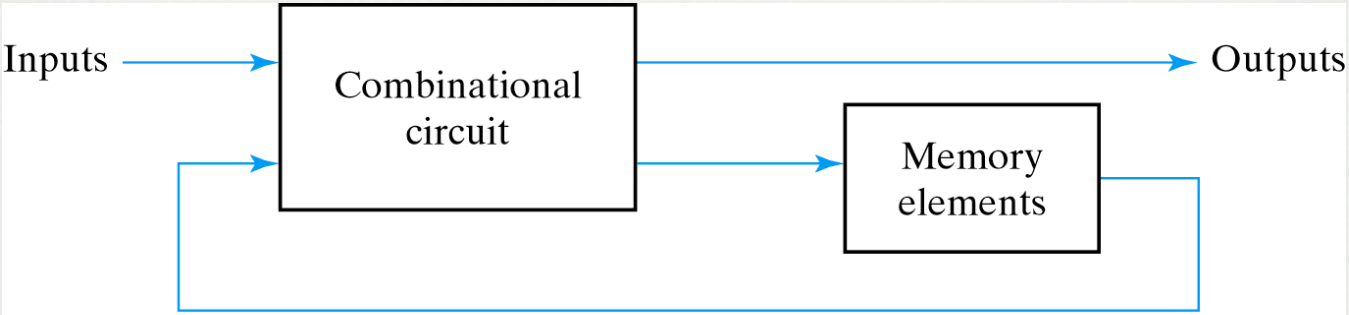
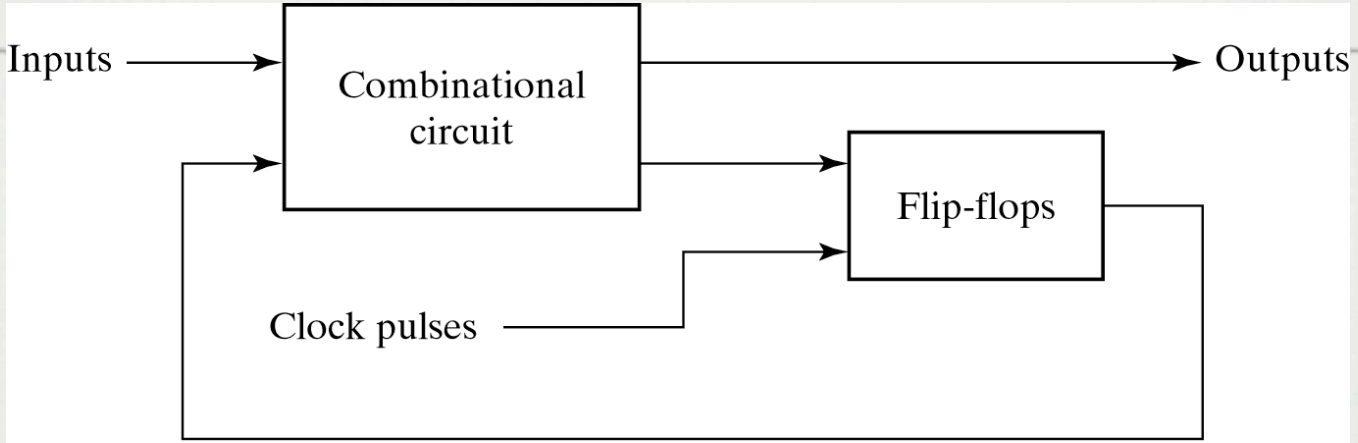
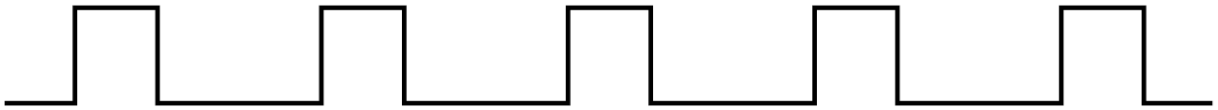


Fig. 5-1 Block Diagram of Sequential Circuit

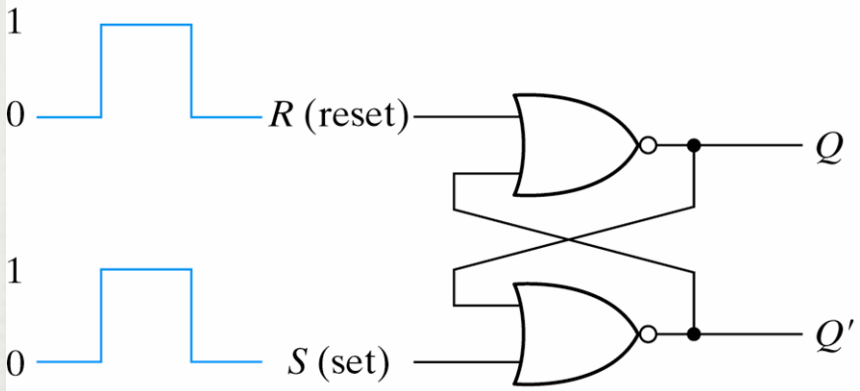


(a) Block diagram



(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

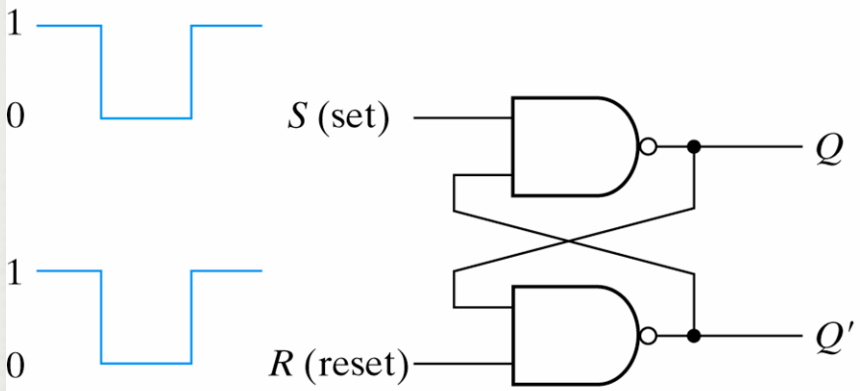


(a) Logic diagram

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	

(b) Function table

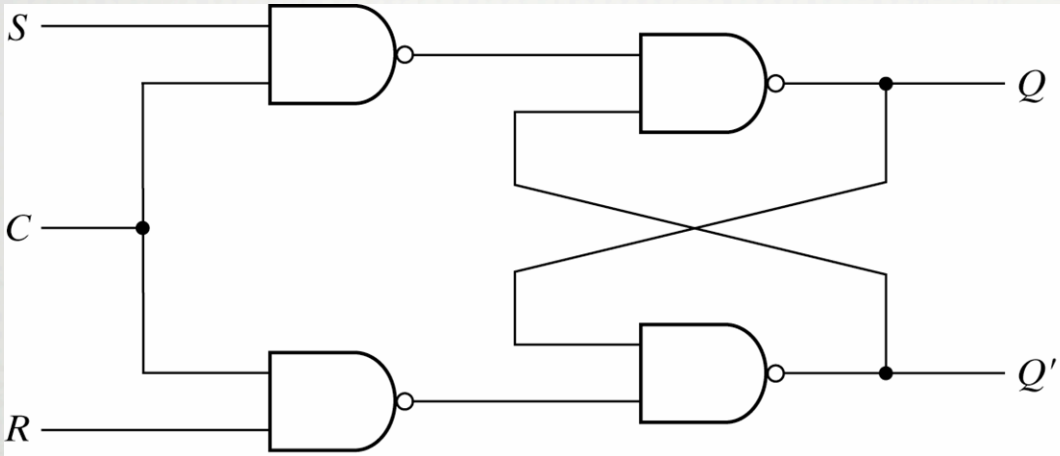
Fig. 5-3 *SR* Latch with NOR Gates



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	

(a) Logic diagram (b) Function table

Fig. 5-4 *SR* Latch with NAND Gates

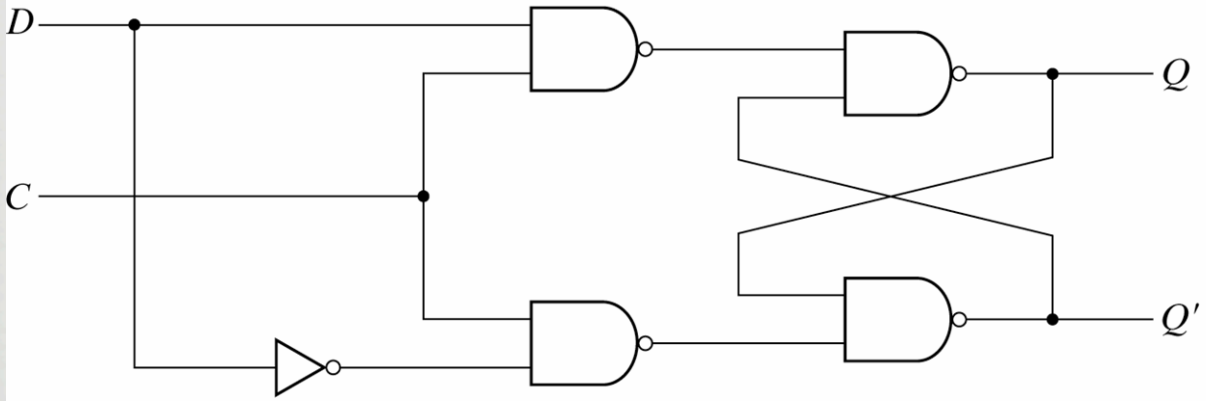


(a) Logic diagram

C	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

Fig. 5-5 SR Latch with Control Input



(a) Logic diagram

C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

(b) Function table

Fig. 5-6 D Latch

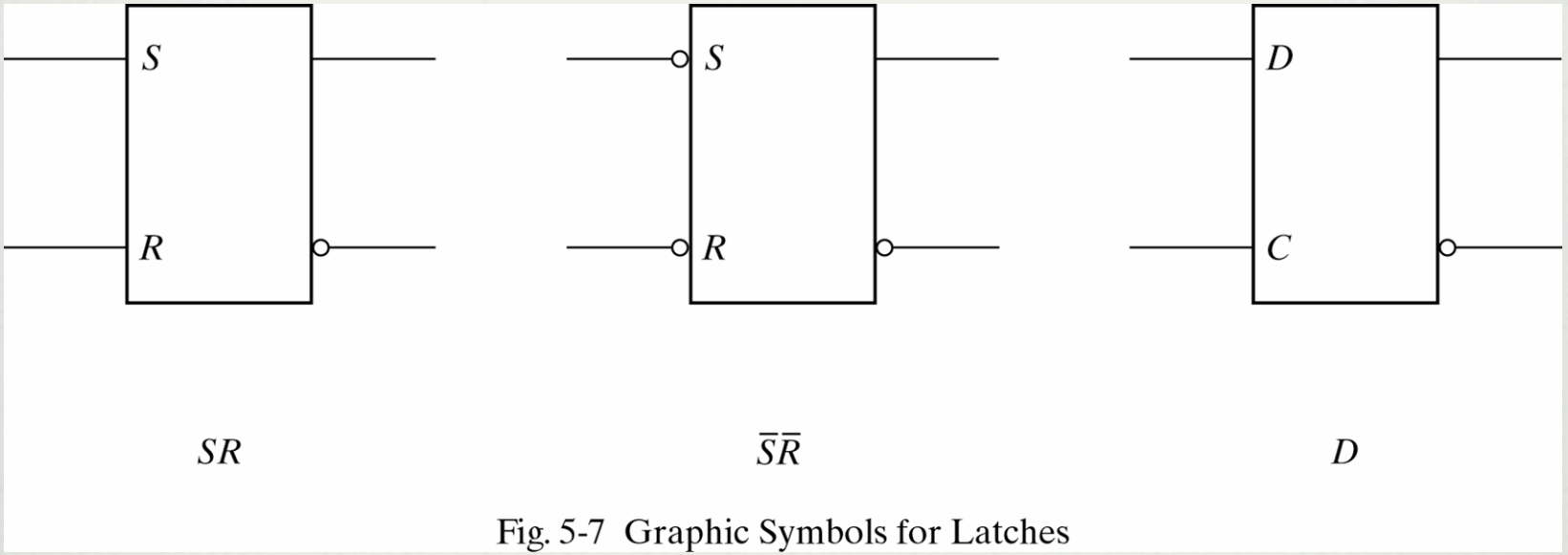


Fig. 5-7 Graphic Symbols for Latches

(a) Response to positive level

(b) Positive-edge response

(c) Negative-edge response

Fig. 5-8 Clock Response in Latch and Flip-Flop

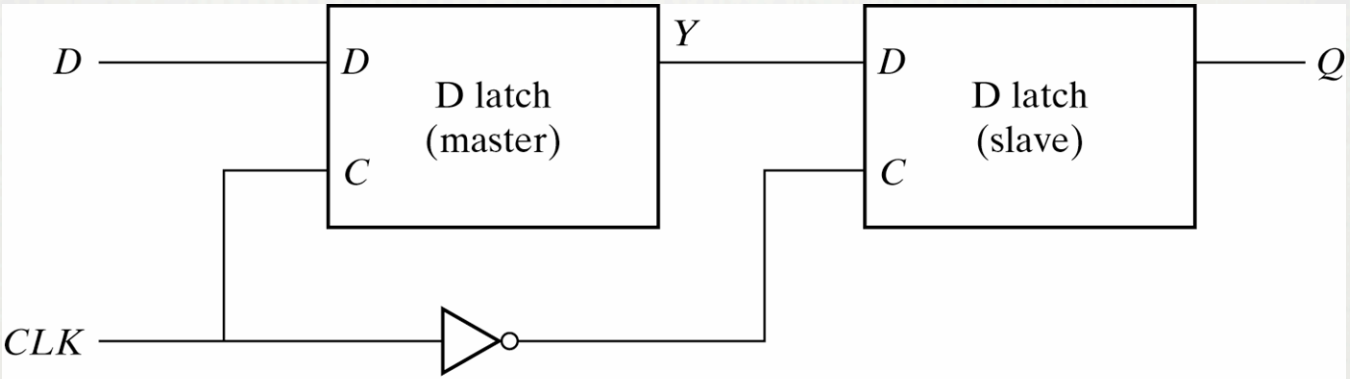
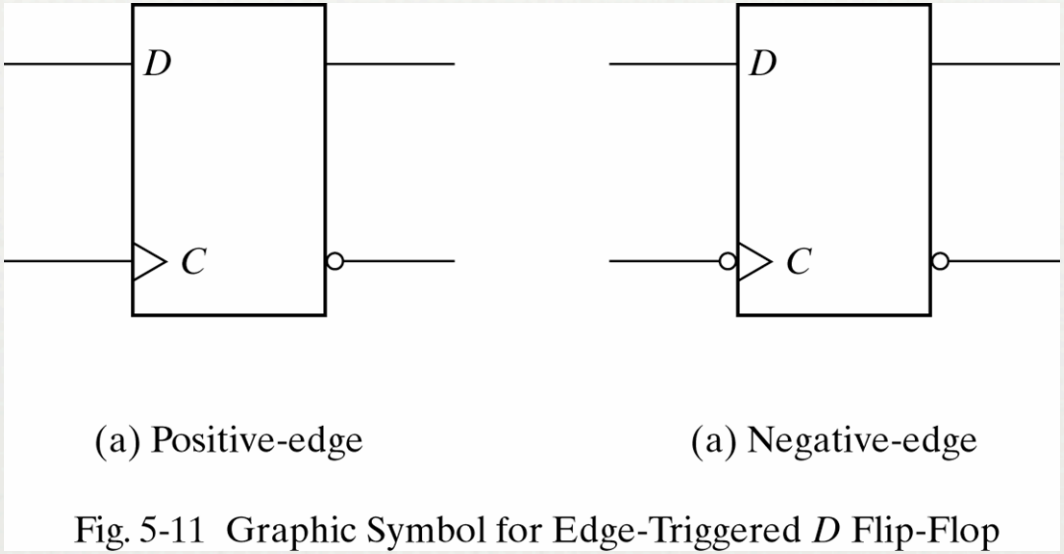
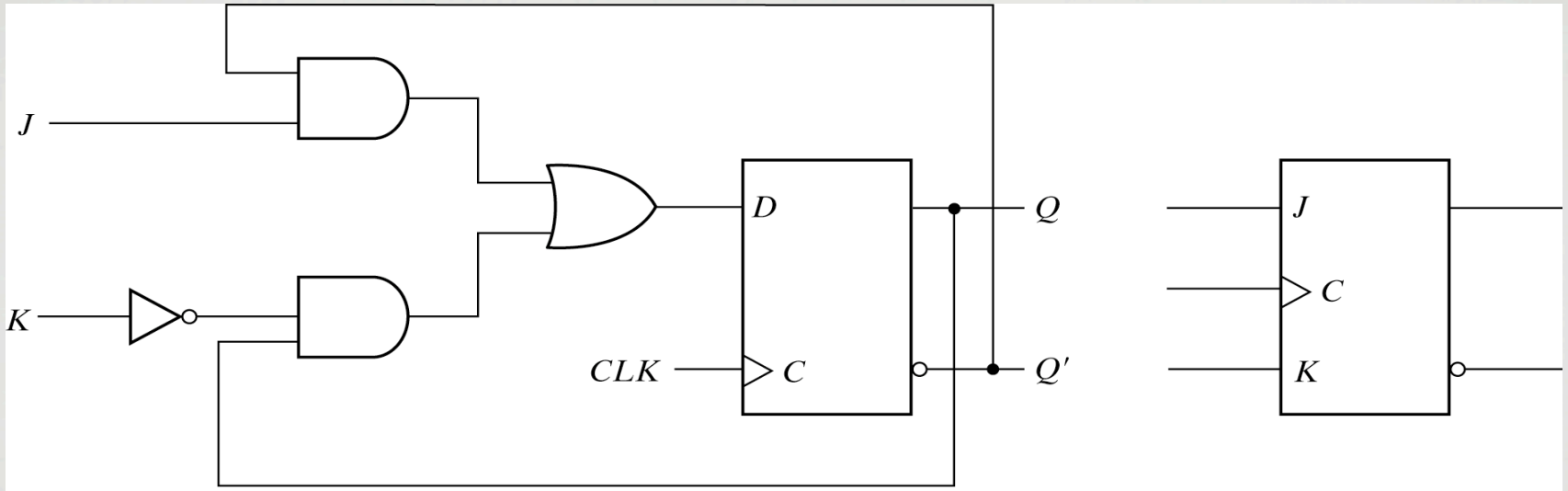


Fig. 5-9 Master-Slave *D* Flip-Flop



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

Fig. 5-11 Graphic Symbol for Edge-Triggered *D* Flip-Flop



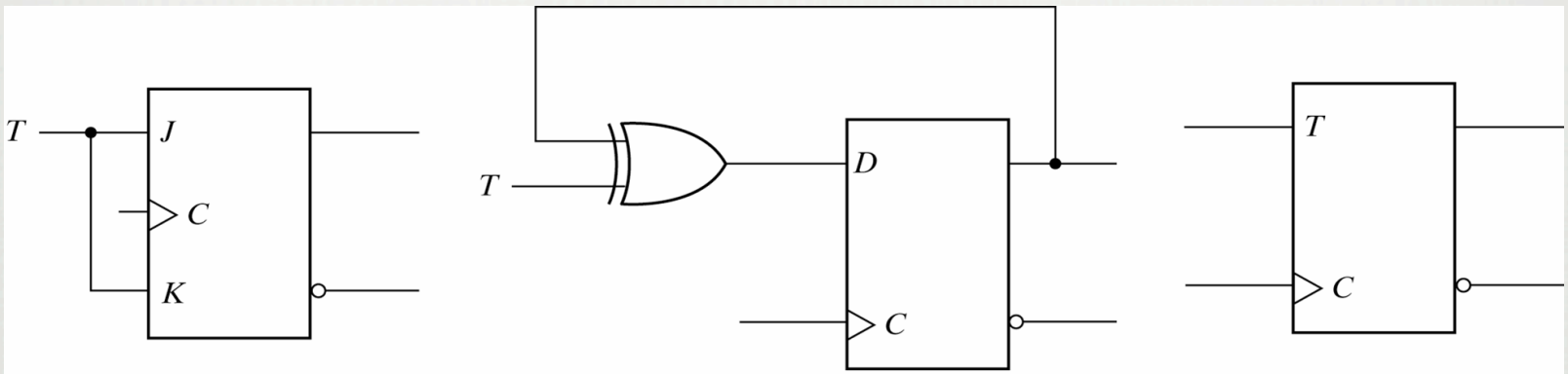
(a) Circuit diagram (b) Graphic symbol

Fig. 5-12 *JK* Flip-Flop

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



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

Fig. 5-13 T Flip-Flop

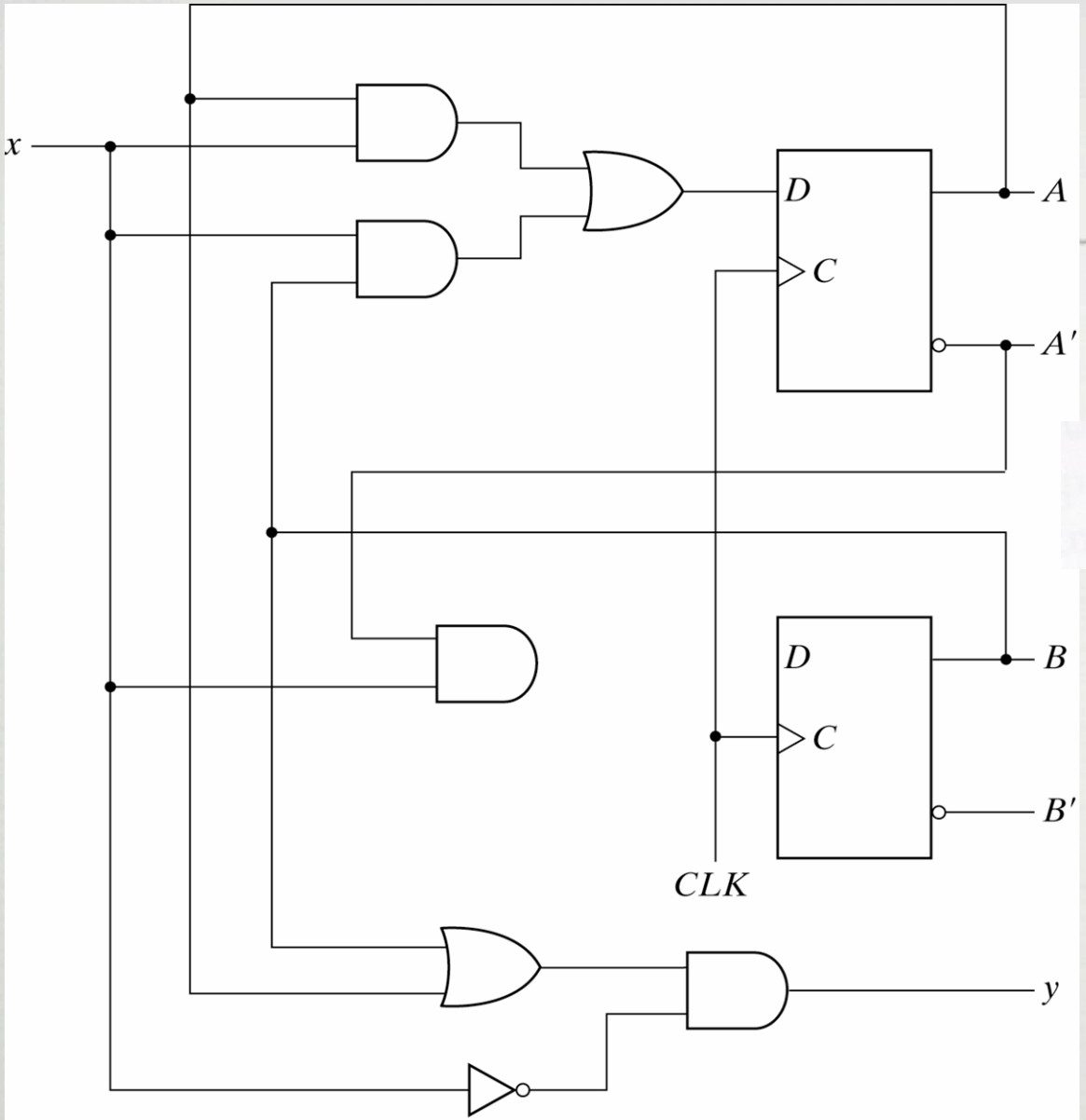


Fig. 5-15 Example of Sequential Circuit

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

$$y = (A + B)x'$$

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

STATE EQUATIONS OR TRANSITION EQUATIONS

$$y = (A + B)x'$$

OUTPUT BOOLEAN EQUATION

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
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

/0 REPRESENTS THE OUTPUT DURING THE PRESENT STATE WITH THE GIVEN INPUT

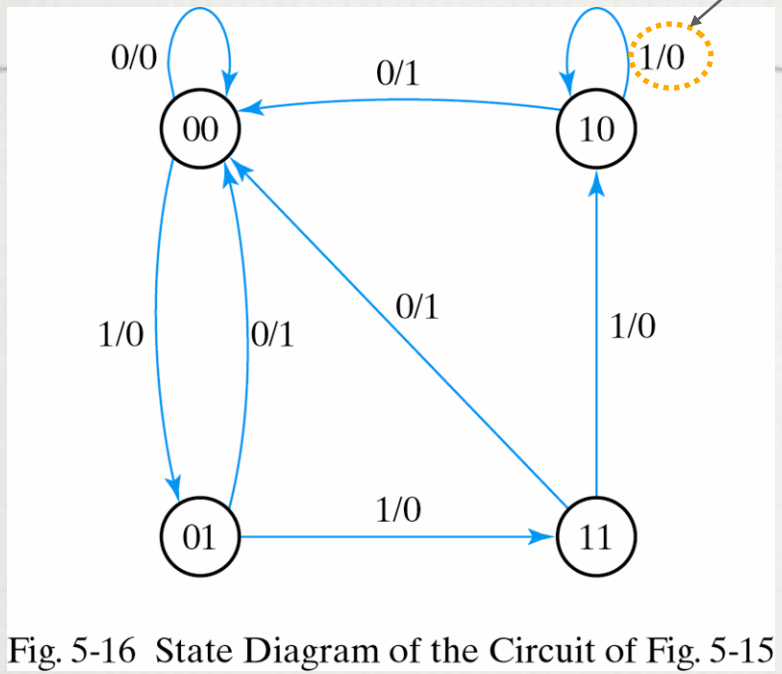
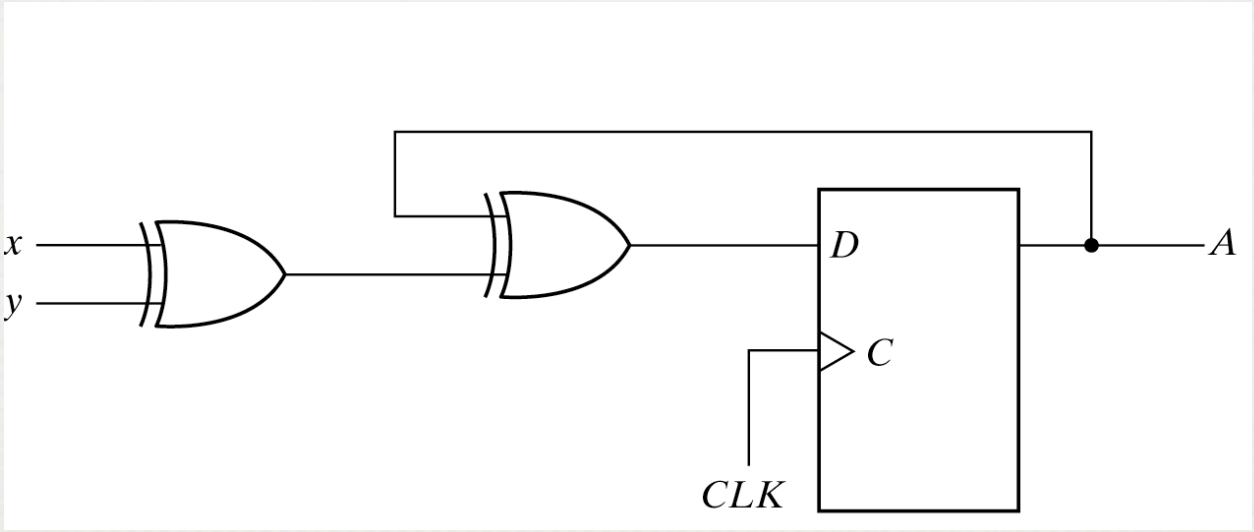


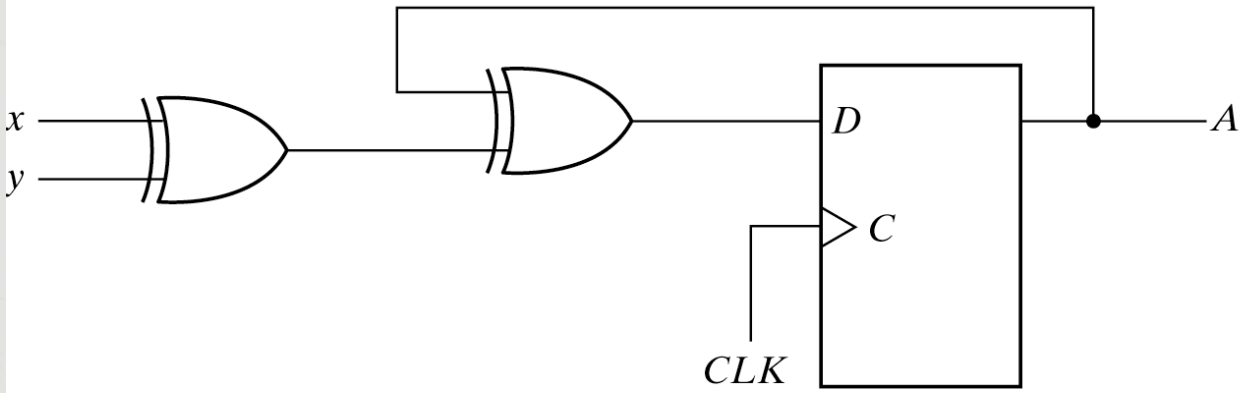
Fig. 5-16 State Diagram of the Circuit of Fig. 5-15

MEALY FINITE STATE MACHINE (FSM) – OUTPUT IS A FUNCTION OF PRESENT STATE AND INPUT

EXAMPLE:



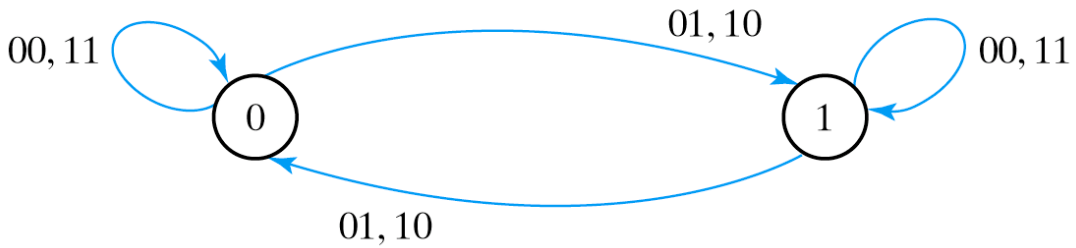
1. FIND THE STATE TABLE
2. DRAW THE STATE DIAGRAM



(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



(c) State diagram

Fig. 5-17 Sequential Circuit with *D* Flip-Flop

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:

$$A(t+1) = x'y + xA$$
$$B(t+1) = x'B + xA$$
$$z = B$$

(a) Draw the logic diagram of the circuit.

(b) List the state table for the sequential circuit.

(c) Draw the corresponding state diagram.

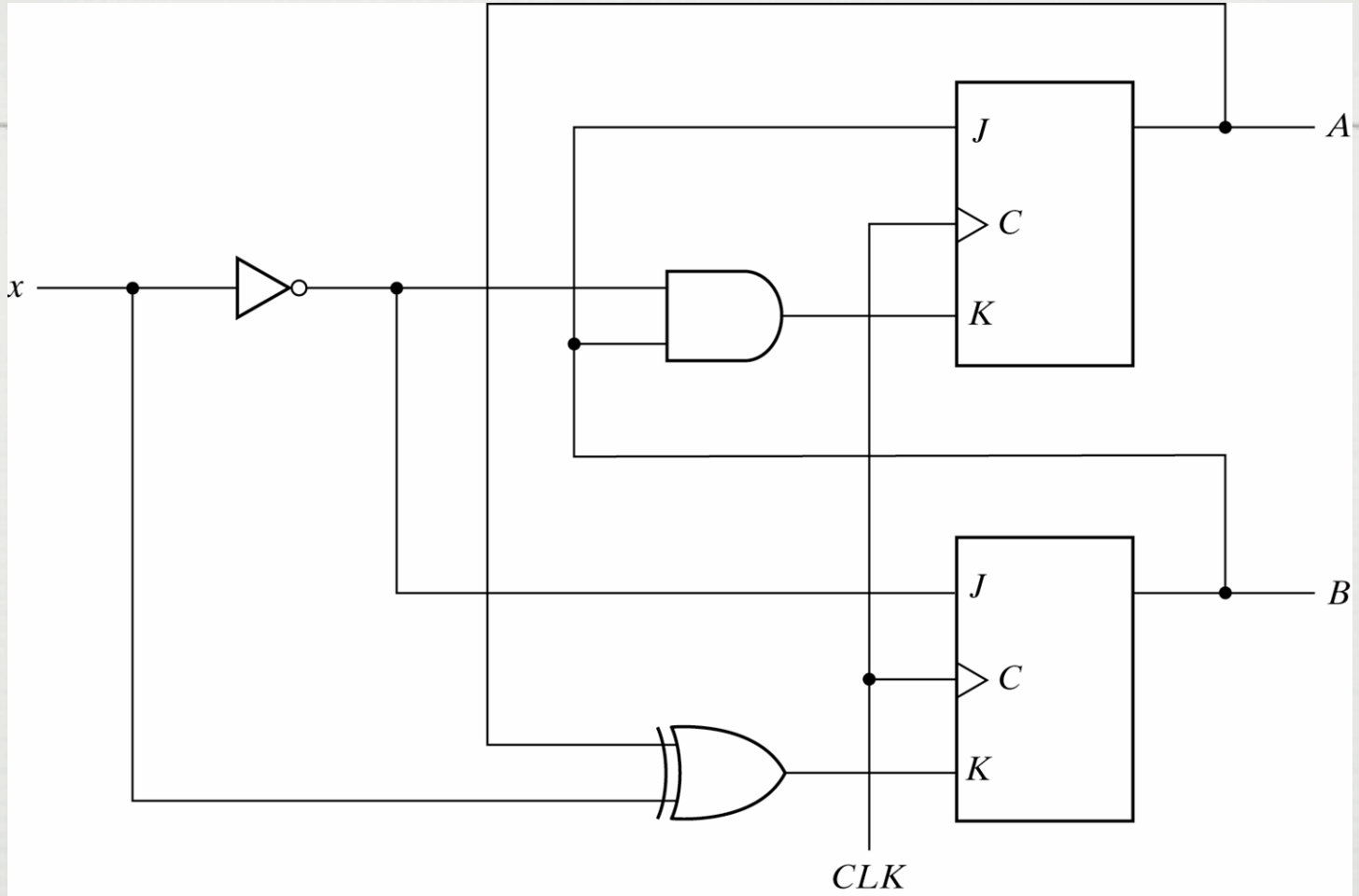


Fig. 5-18 Sequential Circuit with *JK* Flip-Flop

flip-flop input equations

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

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

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

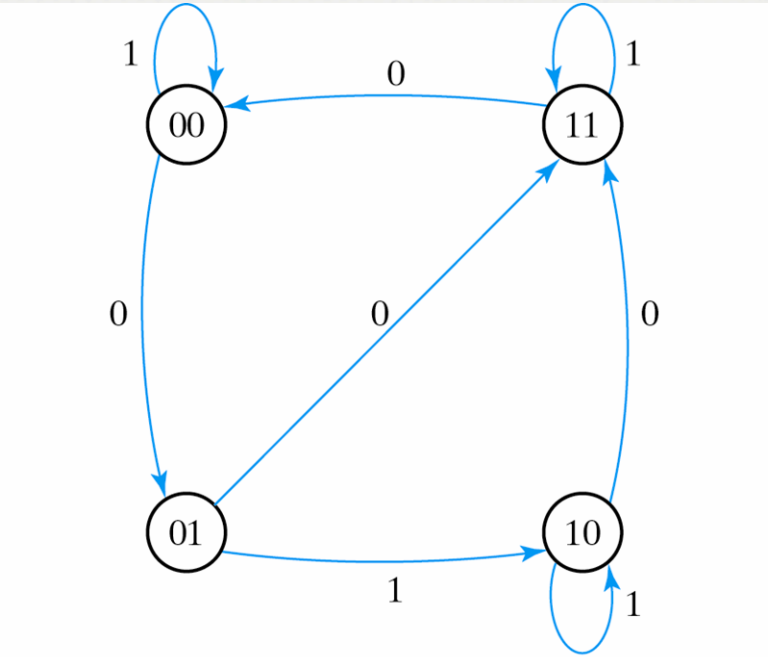
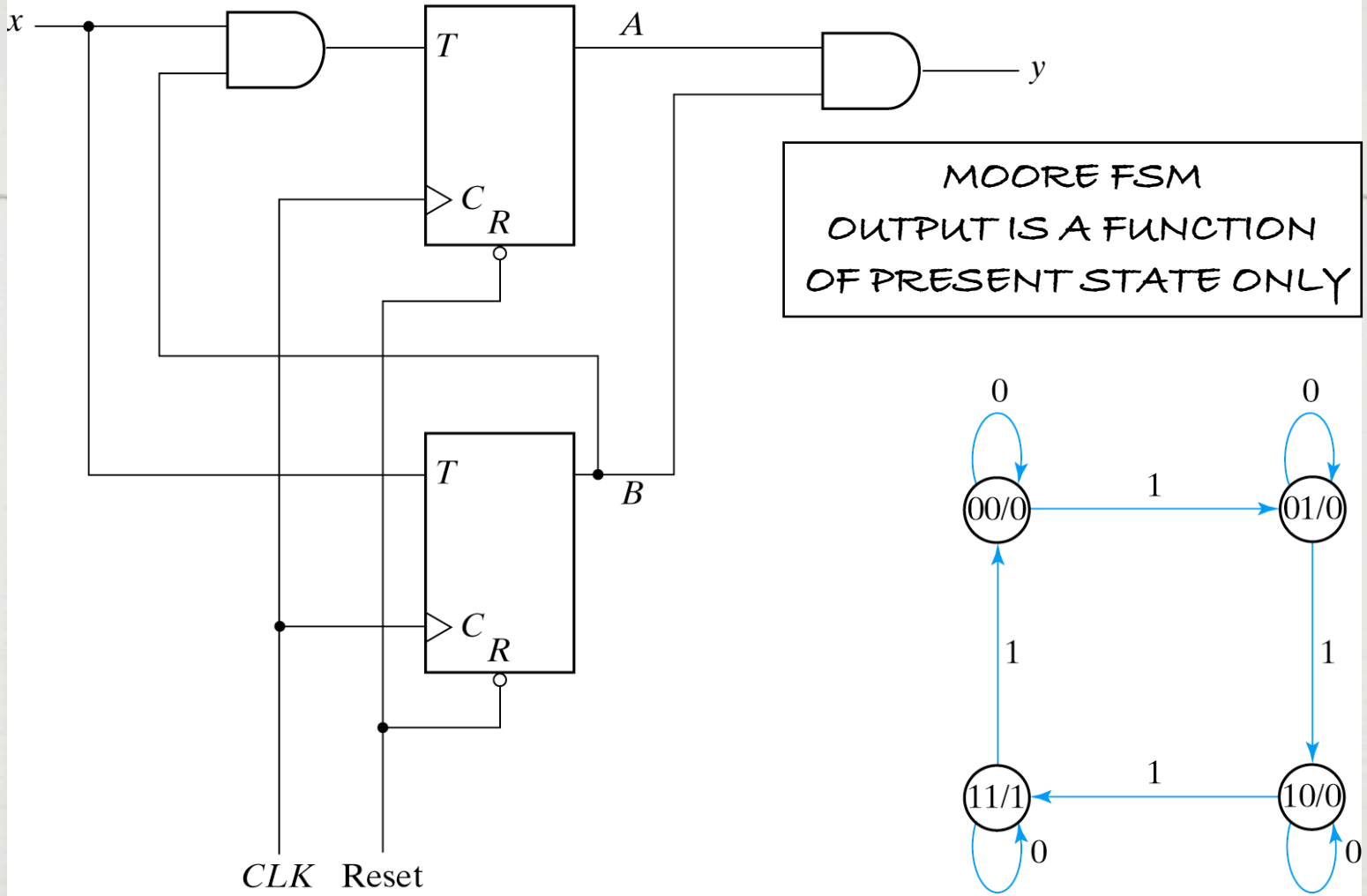


Fig. 5-19 State Diagram of the Circuit of Fig. 5-18



(a) Circuit diagram

(b) State diagram

Fig. 5-20 Sequential Circuit with *T* Flip-Flops

5-10 A sequential circuit has two JK flip-flops A and B, two inputs x and y, and one output z. The flip-flop input equations and circuit output equation are

$$J_A = Bx + B'y' \qquad K_A = B'xy'$$
$$J_B = A'x \qquad K_B = A + xy'$$
$$z = Ax'y' + Bx'y'$$

(a) Draw the logic diagram of the circuit.

(b) Tabulate the state table.

(c) Derive the state equations for A and B.

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.

STATE REDUCTION

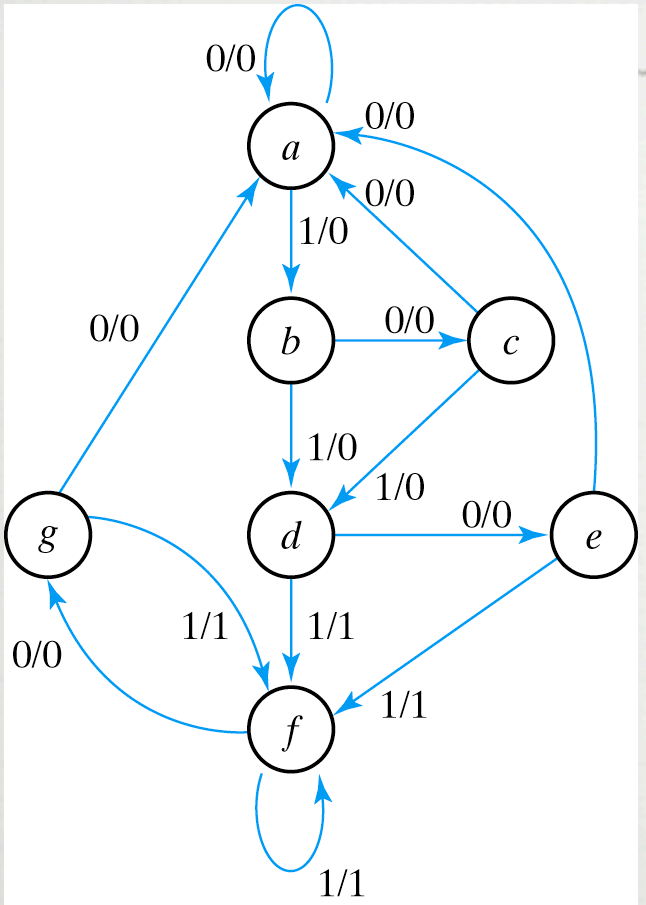


Fig. 5-22 State Diagram

Table 5-6
State Table

Present State	Next State $x = 0$	Next State $x = 1$	Output $x = 0$	Output $x = 1$
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

STATES ARE APPLICATION-DEPENDANT.
THE NAMES GIVEN HERE (A,B,C,D,...)
ARE ARBITRARY.
IT IS ASSUMED THAT ONLY THE OUTPUT
RESPONSE TO A GIVEN SEQUENCE OF
INPUTS IS IMPORTANT.

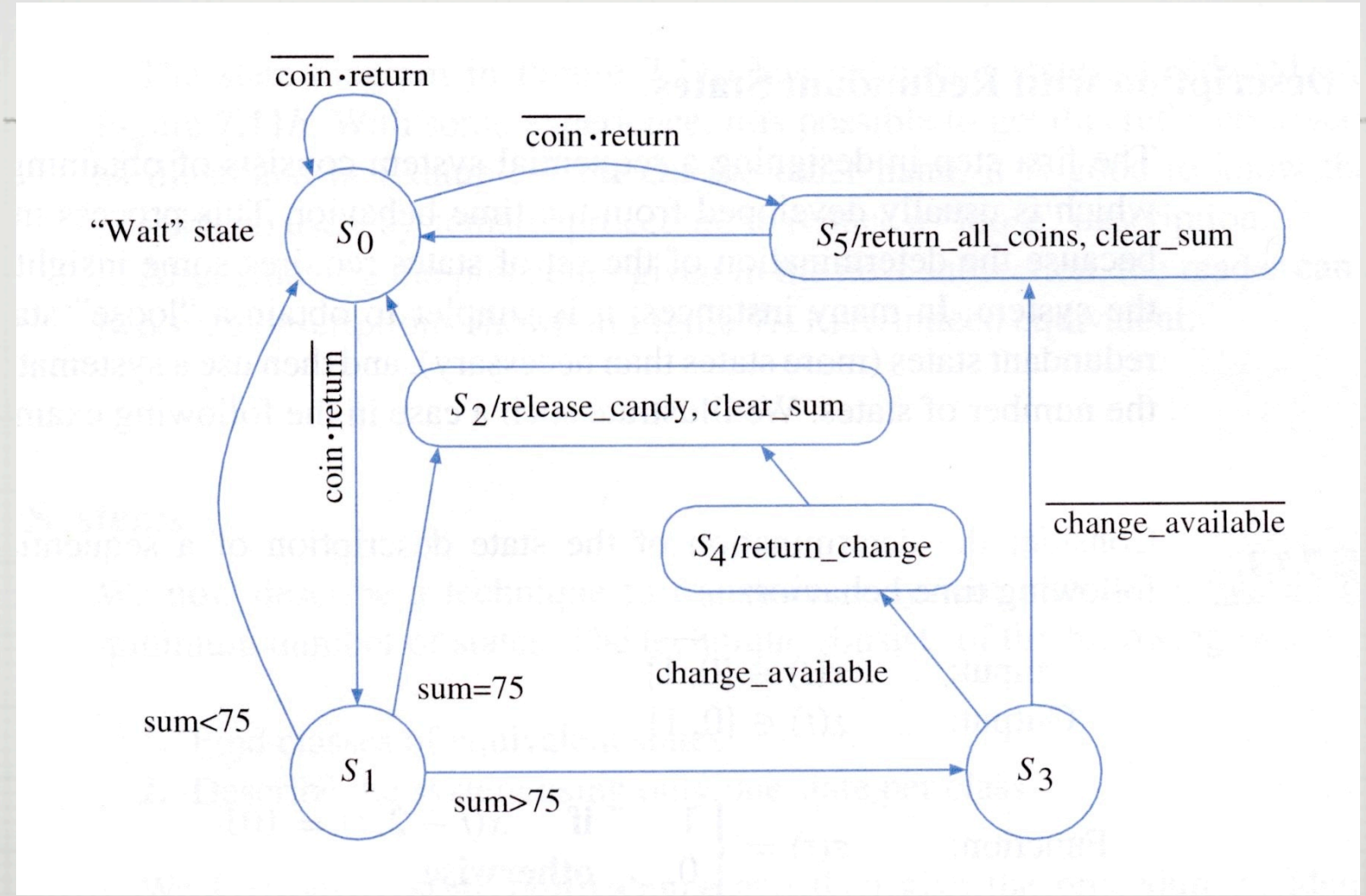

$\overline{\text{coin}} \cdot \overline{\text{return}}$
$\overline{\text{coin}} \cdot \text{return}$
"Wait" state
S_0
S_5/return_all_coins, clear_sum
$\overline{\text{coin} \cdot \text{return}}$
S_2/release_candy, clear_sum
S_4/return_change
$\overline{\text{change\_available}}$
change_available
sum=75
sum<75
S_1
S_3
sum>75

Table 5-6
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	g	f	0	1
g	a	f	0	1

An algorithm for the state reduction of a completely specified state table is given here without proof: "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." When two states are equivalent, one of them can be removed without altering the input–output relationships.

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

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

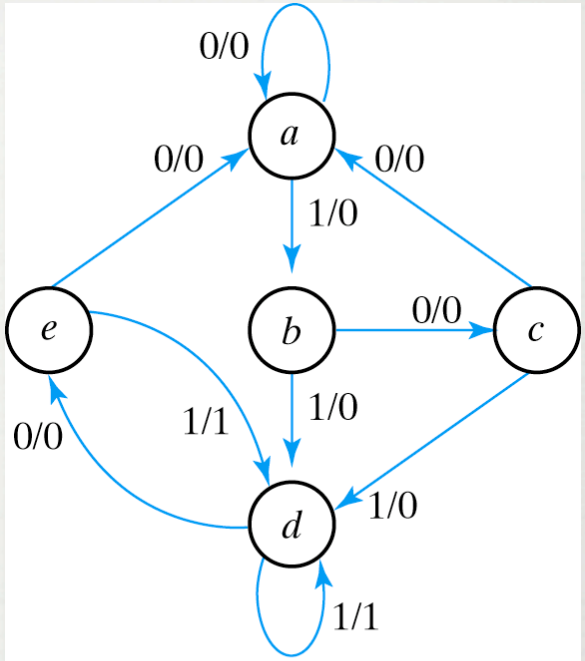


Fig. 5-23 Reduced State Diagram

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

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

SEQUENCE DETECTOR: CIRCUIT THAT DETECTS 3 CONSECUTIVE 1'S IN A STRING OF BITS COMING THROUGH THE INPUT LINE

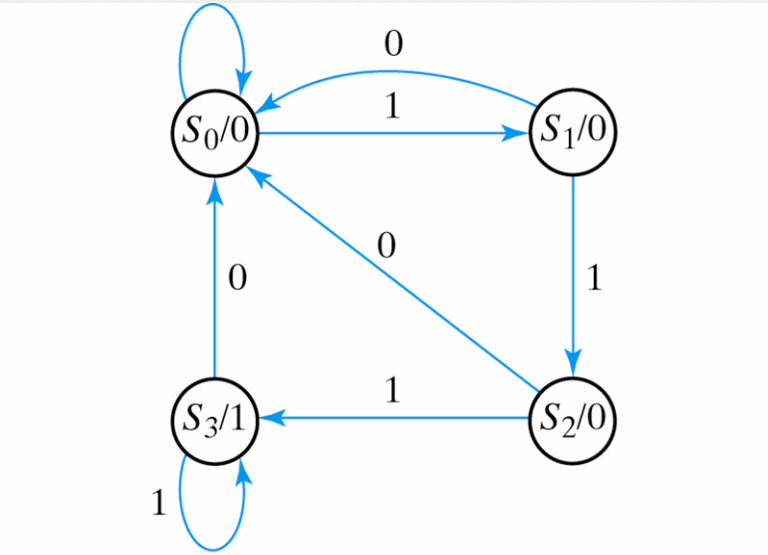
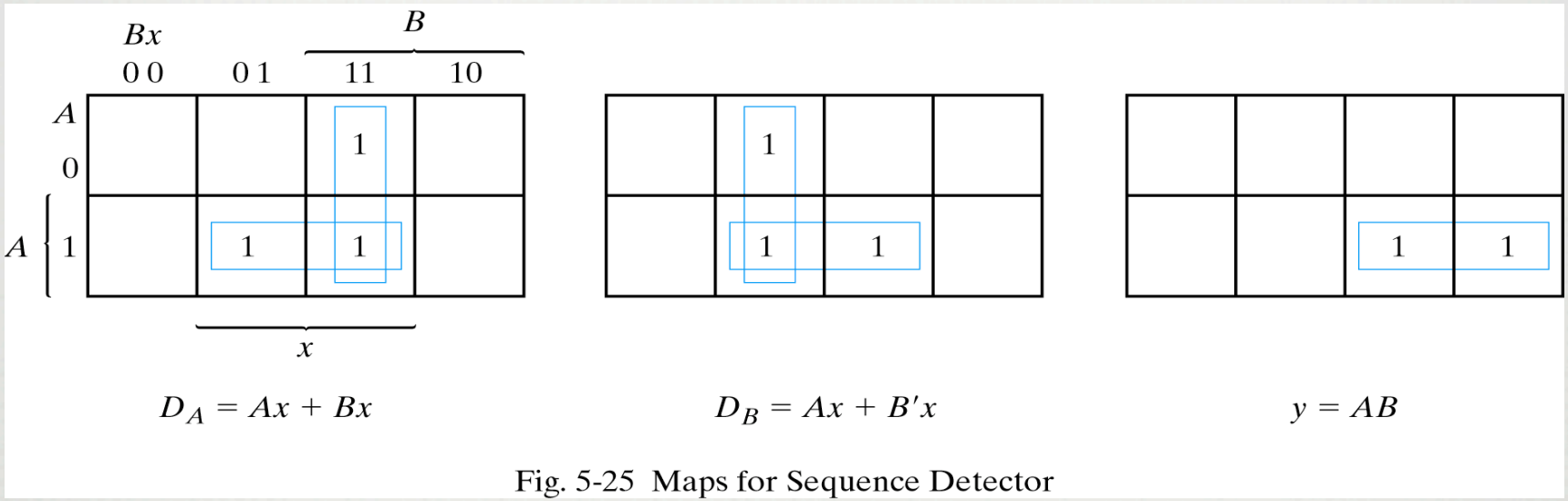


Fig. 5-24 State Diagram for Sequence Detector



$D_A = Ax + Bx$

$D_B = Ax + B'x$

$y = AB$

Fig. 5-25 Maps for Sequence Detector

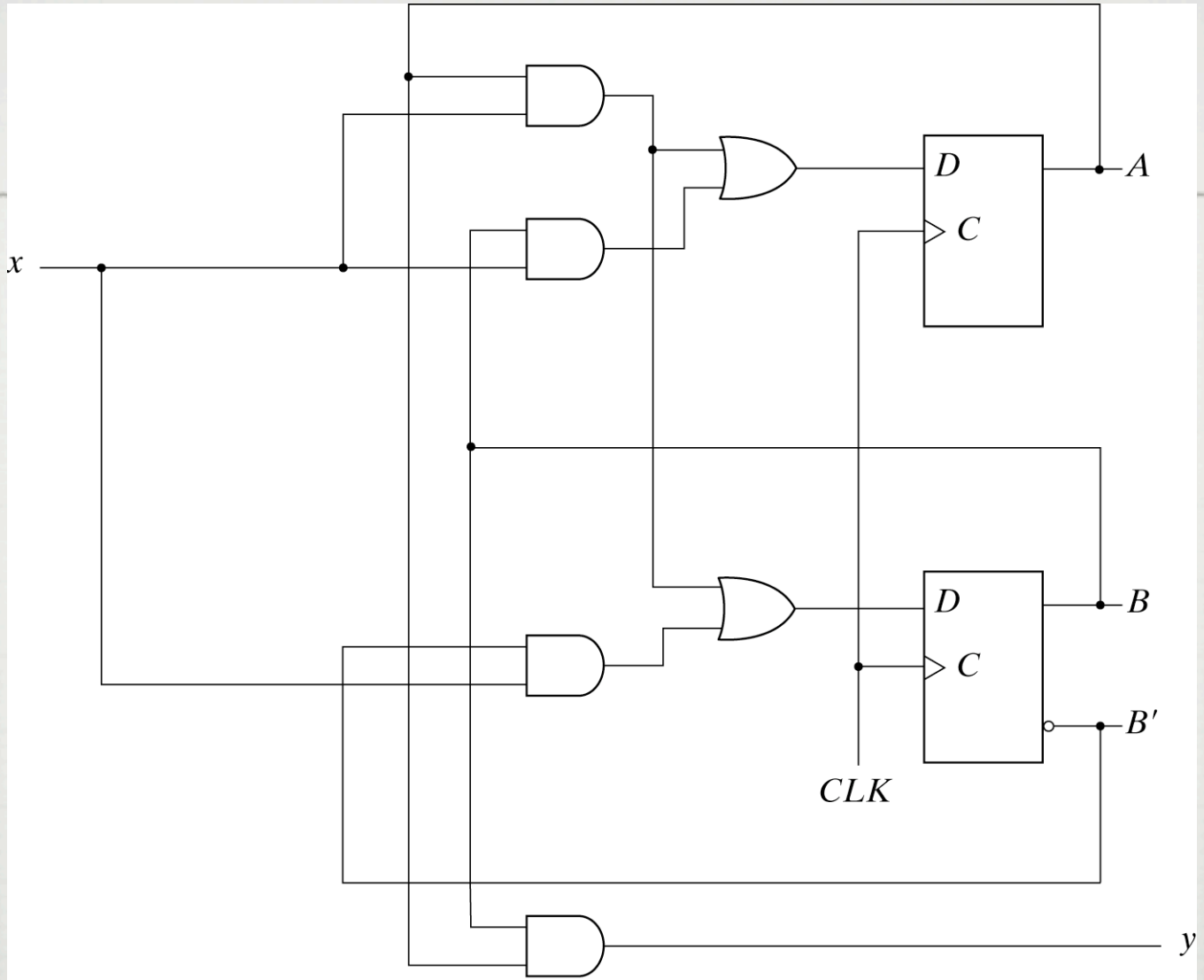


Fig. 5-26 Logic Diagram of Sequence Detector

USING JK OR T FLIP-FLOPS

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

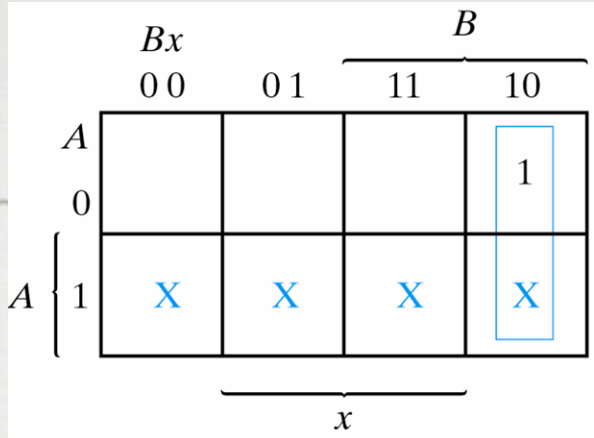
(a) JK

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

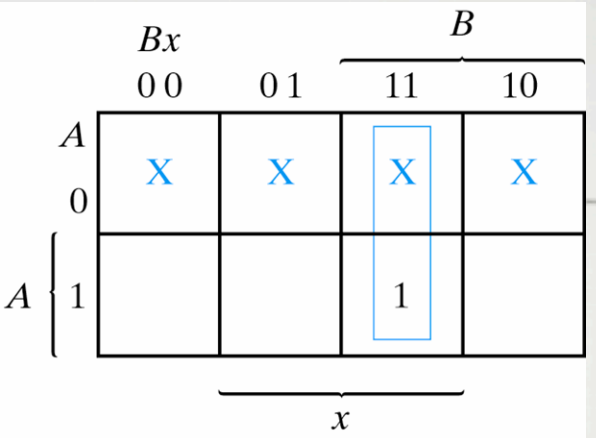
(b) T

Table 5-13
State Table and JK Flip-Flop Inputs

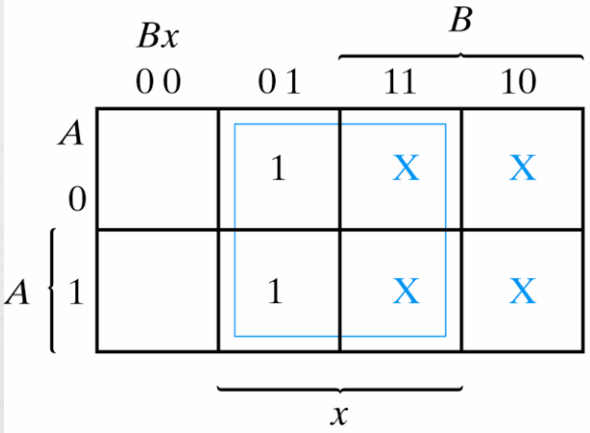
Present State		Input	Next State		Flip-Flop Inputs			
A	*B*	*x*	*A*	*B*	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



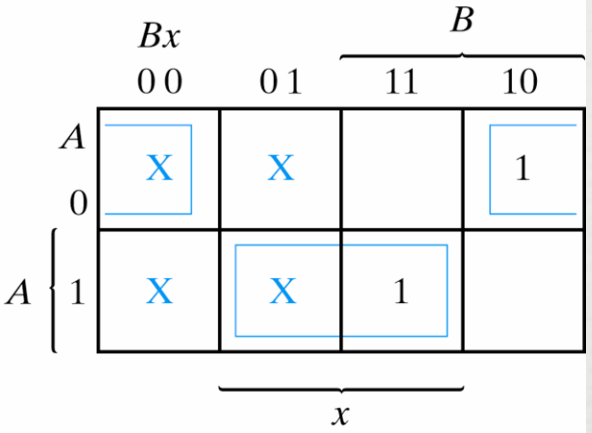
$J_A = Bx'$



$K_A = Bx$



$J_B = x$



$K_B = (A \oplus x)'$

Fig. 5-27 Maps for J and K Input Equations

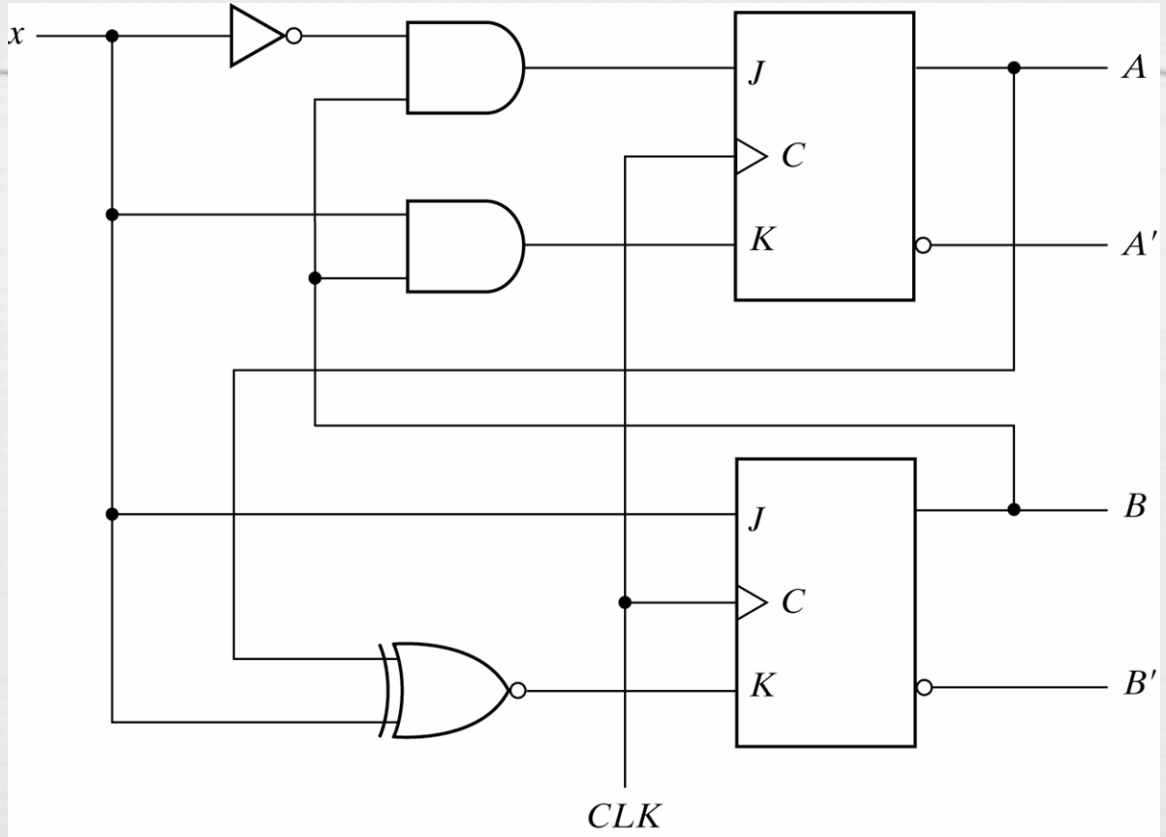
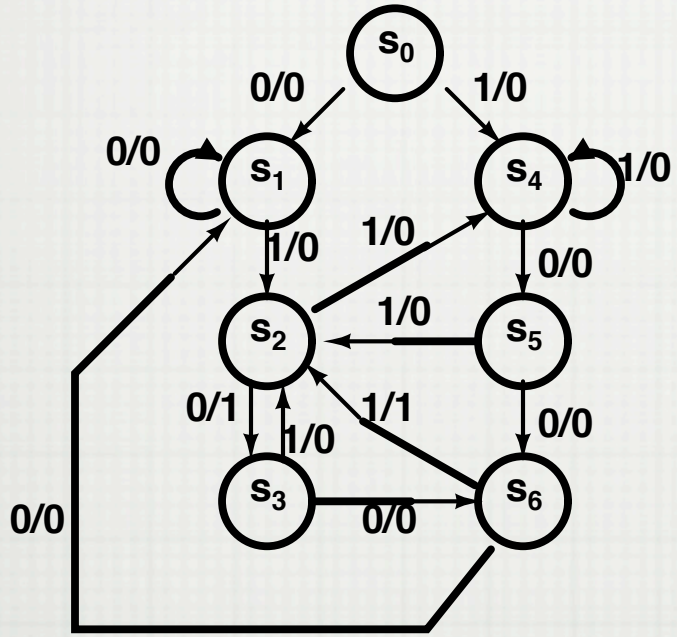


Fig. 5-28 Logic Diagram for Sequential Circuit with *JK* Flip-Flops

STATE ASSIGNMENT GUIDELINES

- ASSIGN NEIGHBORING CODES IF STATES HAVE THE SAME
  - NEXT STATE (G1)
  - PREVIOUS STATE (G2)
  - OUTPUTS (G3)
- PRIORITIZE STATE COMBINATIONS FOR WHICH G1, G2, G3 APPLY MORE THAN ONCE

SEQUENCE DETECTOR FOR 010 OR 1001



present	next x=0	next x1	output x=0	output x=1
s_0	s_1	s_4	0	0
s_1	s_1	s_2	0	0
s_2	s_3	s_4	1	0
s_3	s_6	s_2	0	0
s_4	s_5	s_4	0	0
s_5	s_6	s_2	0	0
s_6	s_1	s_2	0	1

S_3 & S_5 ARE EQUIVALENT

SEQUENCE DETECTOR FOR 010 OR 1001 (CONT)

present	next x=0	next x1	output x=0	output x=1
s_0	s_1	s_4	0	0
s_1	s_1	s_2	0	0
s_2	s_3	s_4	1	0
s_3	s_6	s_2	0	0
s_4	s_3	s_4	0	0
s_6	s_1	s_2	0	1

- (S0,S1,S6), (S2, S4), (S0,S2,S4), (S1,S3,S6) G1
- (S1,S2), (S3,S4) G2 X 2
- (S0, S1, S3, S4) G3

	00	01	11	10
0	s0	s1	s6	X
1	s4	s2	X	s3

ONE POSSIBILITY

SEQUENCE DETECTOR FOR 010 OR 1001 (CONT)

present		Next		output	
		x=0	x=1	x=0	x=1
s0	000	001	100	0	0
s1	001	001	101	0	0
s2	101	110	100	1	0
s3	110	011	101	0	0
s4	100	110	100	0	0
s6	011	001	101	0	1

5-19 A sequential circuit has three flip-flops A, B, C; one input x; and one output y. 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.

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

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

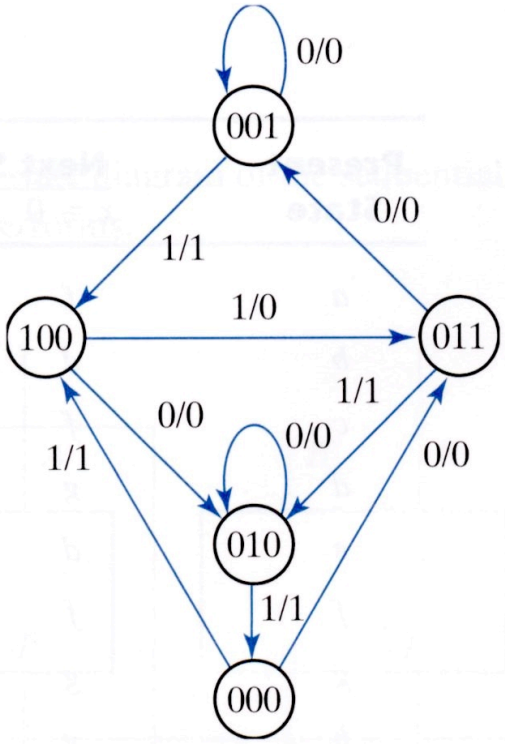
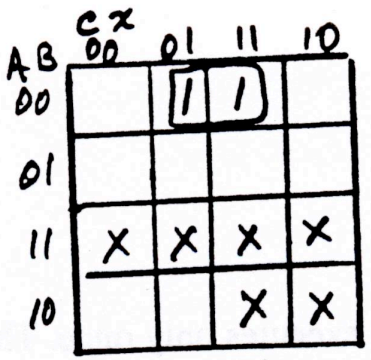


FIGURE P5-19

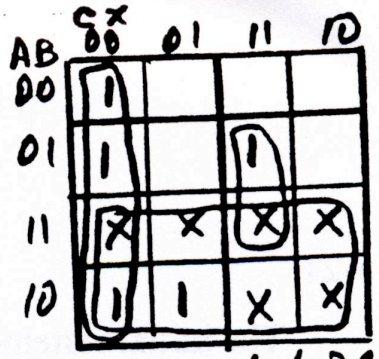
5-19 (a)

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	0

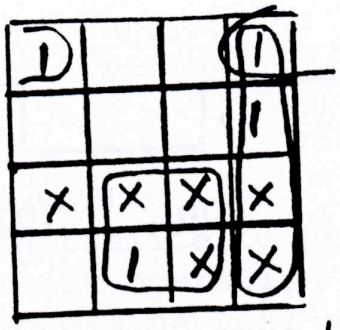
$d(A,B,C,x) = \Sigma(10,11,12,13,14,15)$



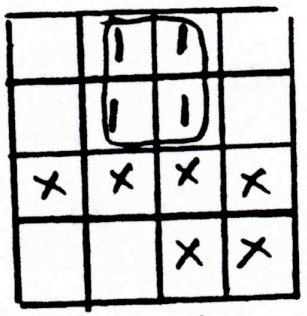
$DA = A'B'x$



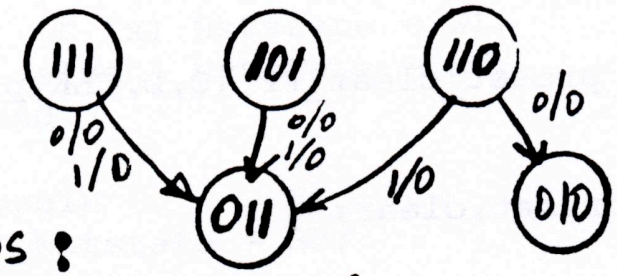
$DB = A + C'x' + BCx$



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



$y = A'x$



self-correcting

(b) Use JK flip flops:
same state lable as in part(a).

Flip-flop inputs

JA	KA	JB	KB	JC	KC
0	X	1	X	1	X
1	X	0	X	0	X
0	X	0	X	X	0
1	X	0	X	X	1
0	X	X	0	0	X
0	X	X	1	0	X
0	X	X	1	X	0
0	X	X	0	X	1
X	1	1	X	0	X
X	1	1	X	1	X

$JA = B'x$
$JB = A + c'x'$
$JC = Ax + AB'x'$
$y = A'x$

$KA = 1$
$KB = c'x + cx'$
$KC = x$

Self-corecting because $KA = 1$

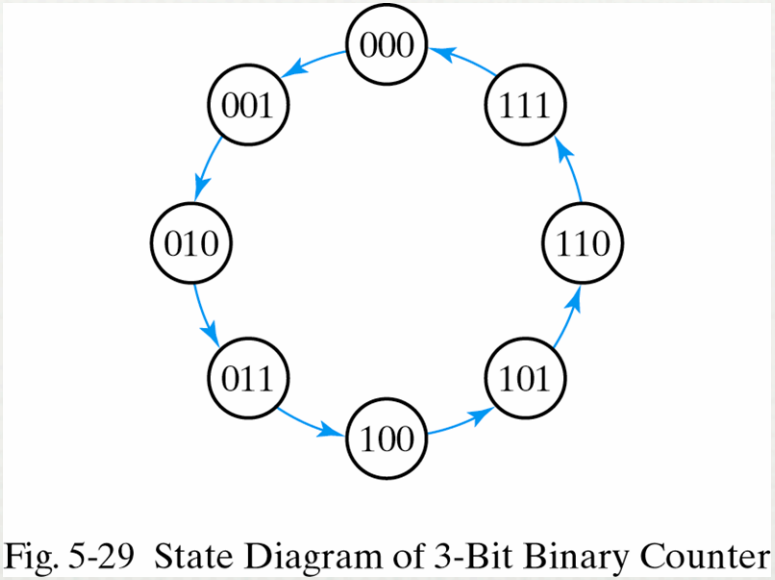


Fig. 5-29 State Diagram of 3-Bit Binary Counter

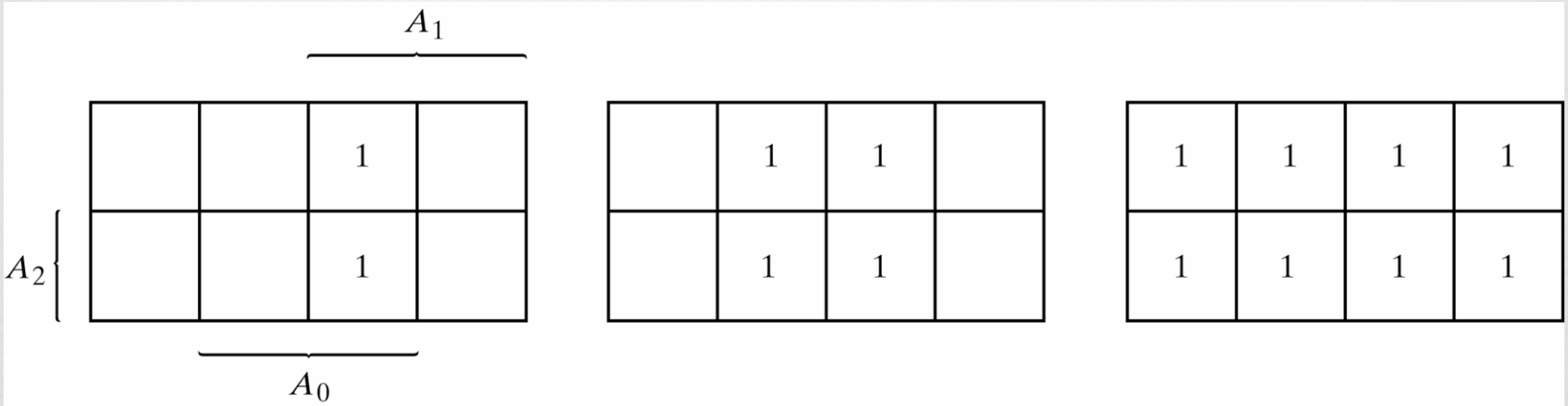


Fig. 5-30 Maps for 3-Bit Binary Counter

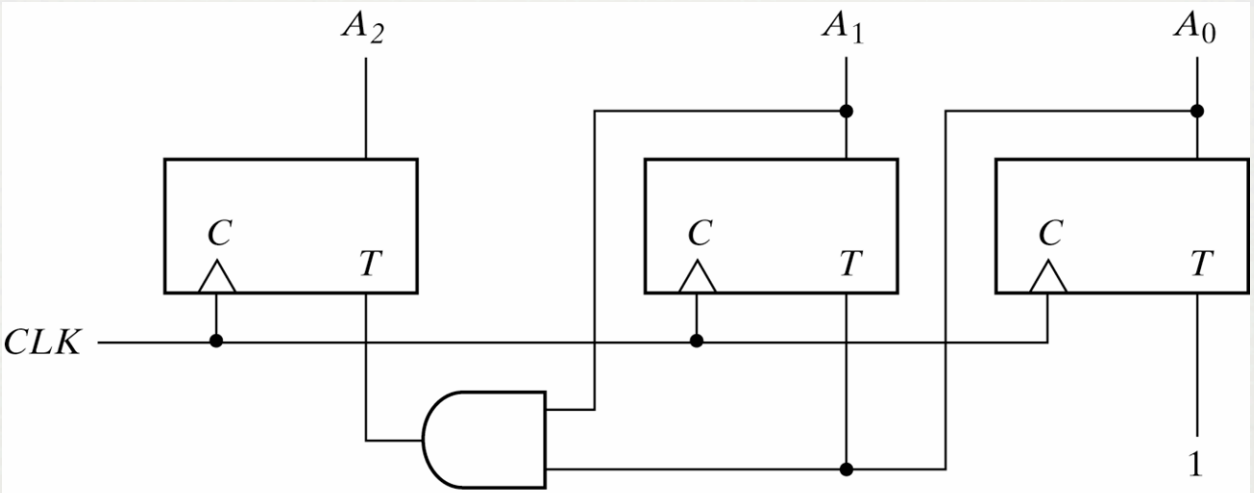


Fig. 5-31 Logic Diagram of 3-Bit Binary Counter

EXERCISE

- DRAW THE STATE DIAGRAM FOR A CIRCUIT THAT DETECTS THE SEQUENCE "0101" (LEFT-TO-RIGHT) USING
  - A MOORE FINITE STATE MACHINE (FSM)
  - A MEALY FSM

EXERCISE

- FOR A CLOCKED SYNCHRONOUS STATE MACHINE WITH TWO INPUTS, X AND Y, AND ONE OUTPUT, Z, THE OUTPUT SHOULD BE 1 IF THE NUMBER OF 1 INPUTS ON X AND Y SINCE RESET IS A MULTIPLE OF 4, AND 0 OTHERWISE. DRAW THE STATE DIAGRAM FOR A
  - MOORE MACHINE
  - MEALY MACHINE

EXERCISE

- DESIGN A CIRCUIT TO DETECT THE SEQUENCE $D_0D_1D_2D_3D_4$=01101, WHERE D_0 IS THE FIRST BIT TO ARRIVE AT INPUT "X". THE OUTPUT "Y" SHOULD BE A LOGIC-1 FOR A FULL CLOCK CYCLE FOLLOWING DETECTION OF THE SEQUENCE.
  - DRAW A STATE DIAGRAM
  - ASSIGN BINARY STATES
  - WRITE A STATE TABLE
  - FIND THE COMBINATIONAL CIRCUIT'S LOGIC EXPRESSIONS IF D, JK AND T FLIP FLOPS WILL BE USED FOR THE 1ST, 2ND AND 3RD STATE BITS, RESPECTIVELY

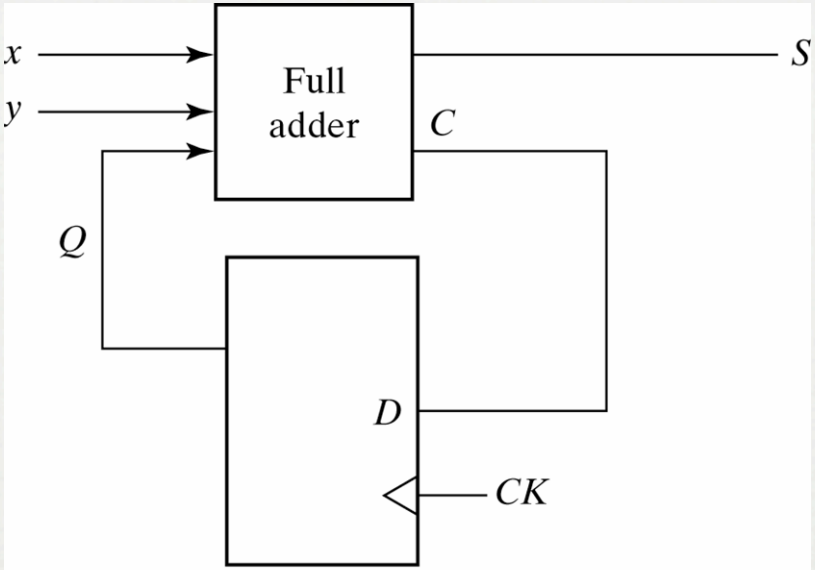


Fig. P5-7

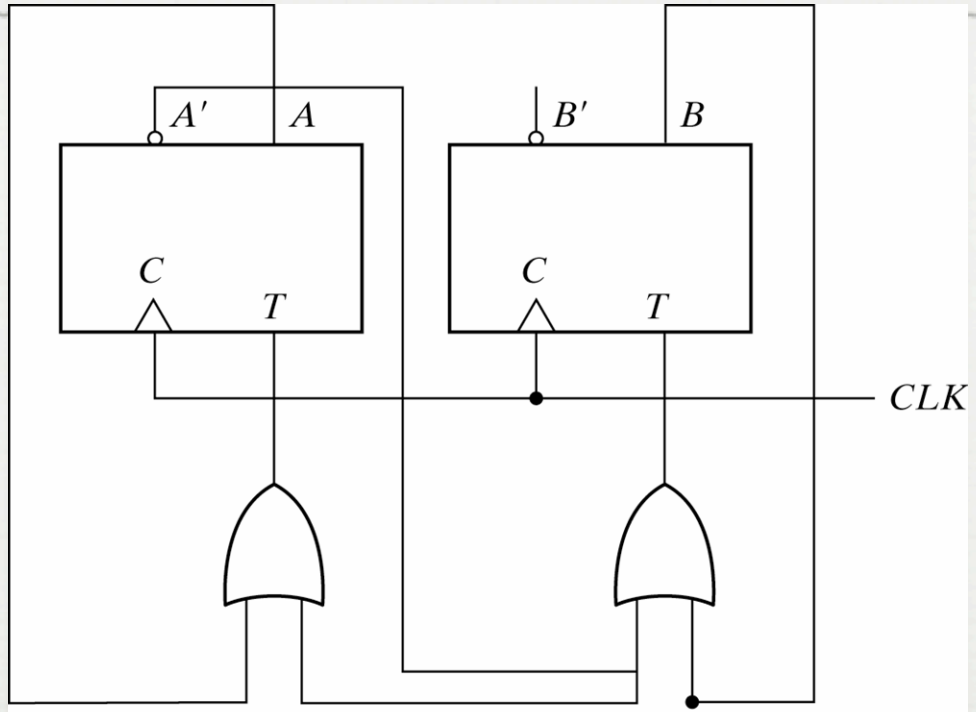


Fig. P5-8

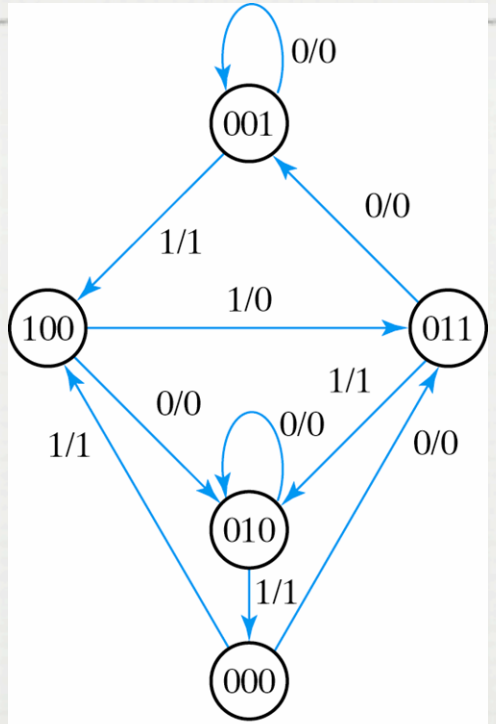


Fig. P5-19