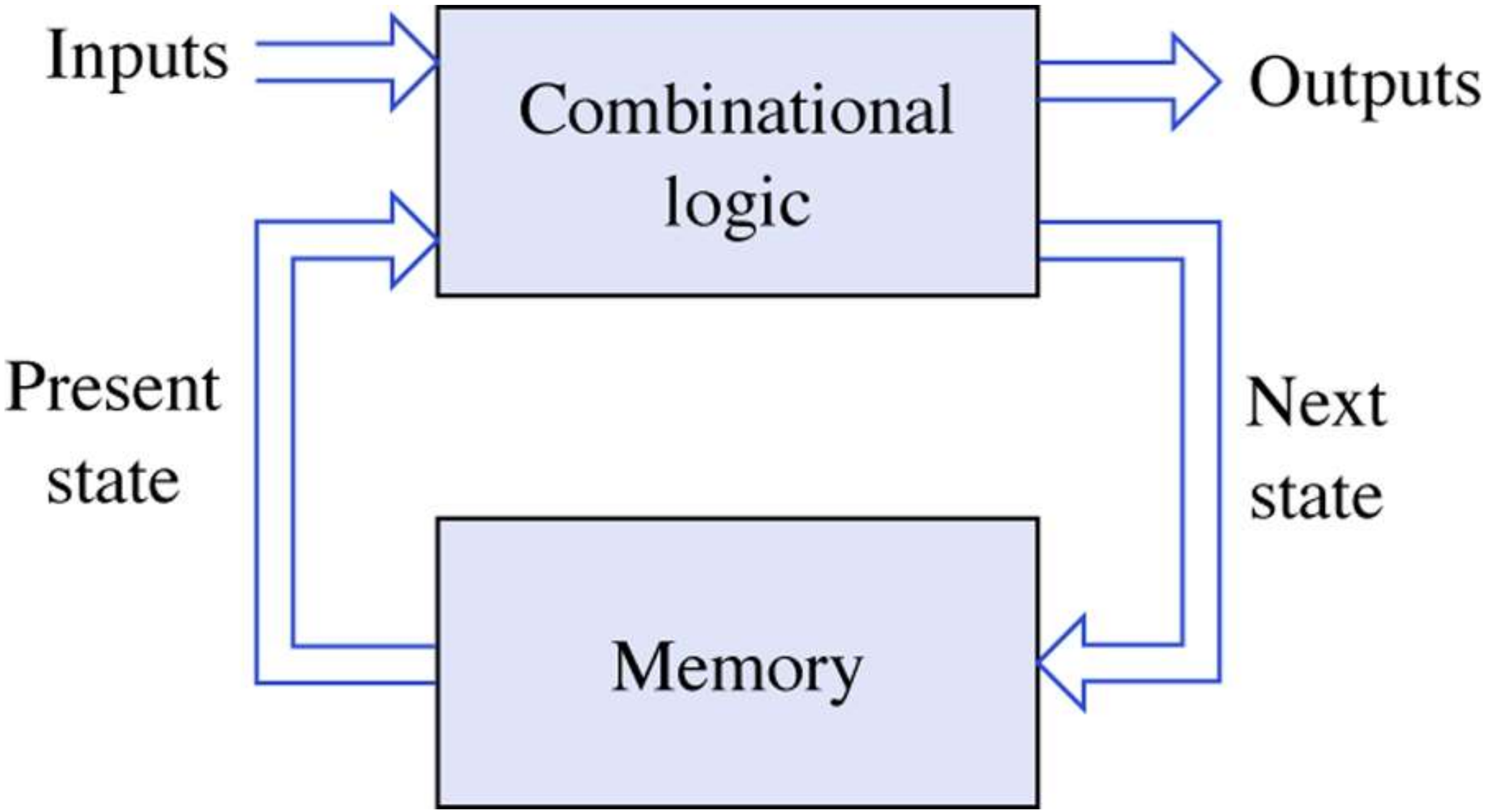
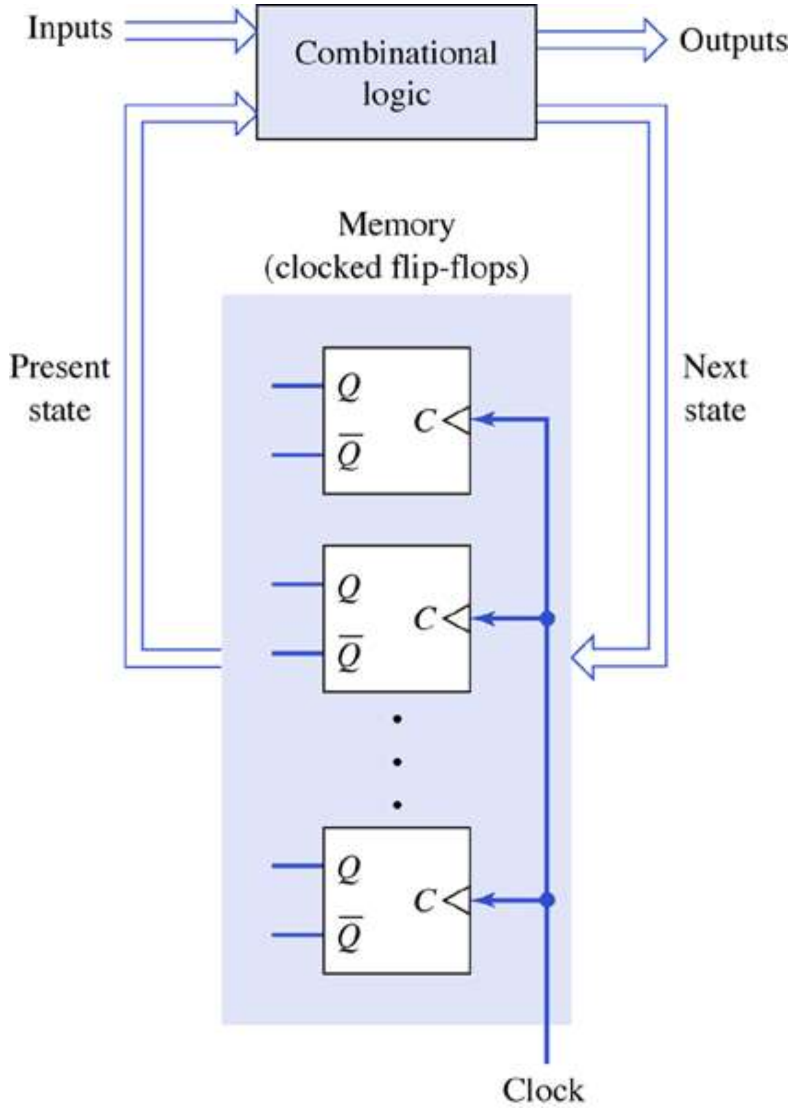
General model of a sequential network.

Figure 7.1



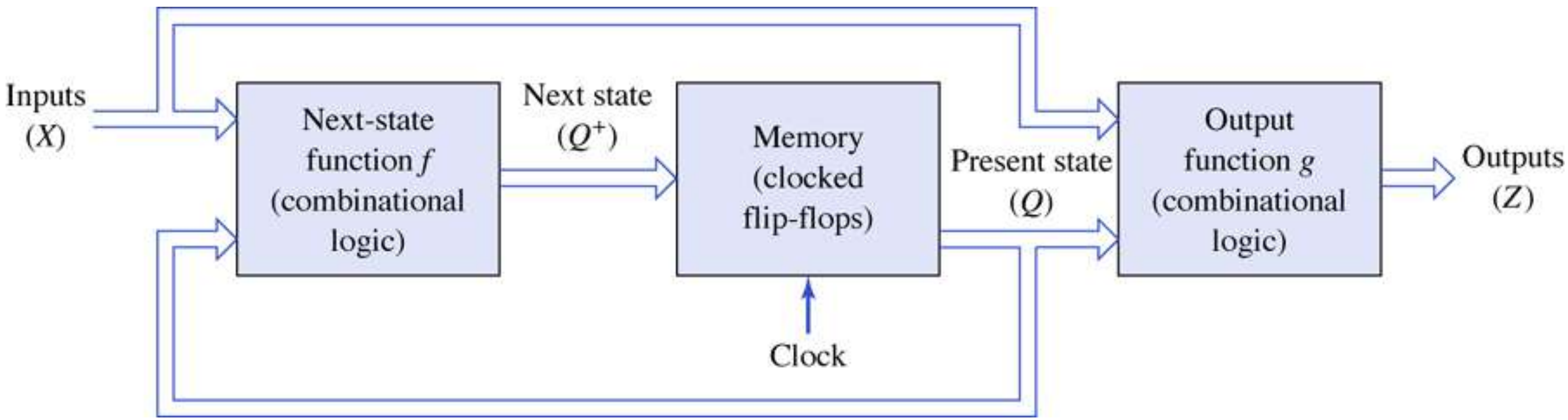
Structure of a clocked synchronous sequential network.

Figure 7.2



Mealy model of a clocked synchronous sequential network.

Figure 7.3



The next state function Q+

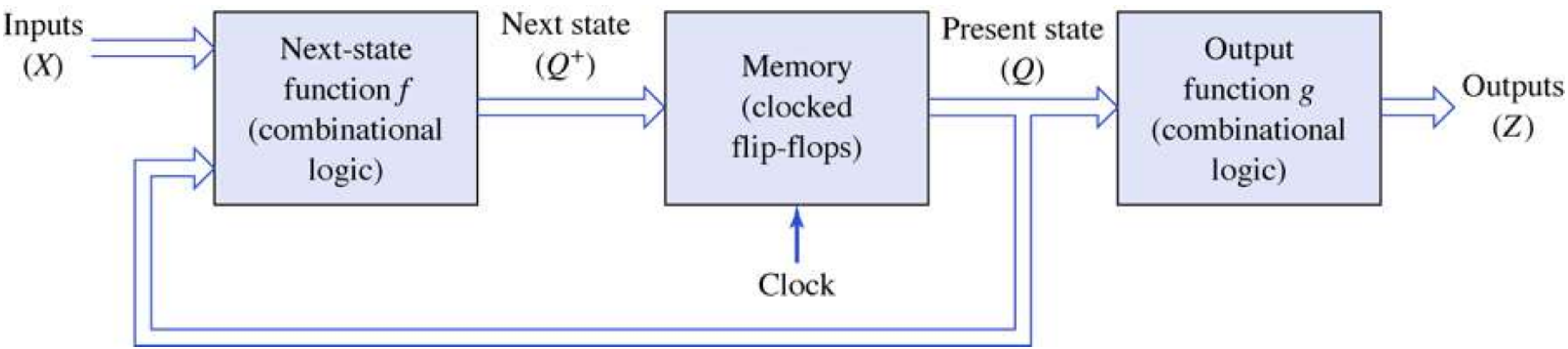
O* = f(X,Q)

Similarly, if Z is regarded as the collective output signals of the network, then under the assumption that the outputs are a function of both the inputs and present state, it immediately follows that

Z= g(X,Q)

Moore model of a clocked synchronous sequential network.

Figure 7.4



The next state function Q+

O* = f(X,Q)

Similarly, if Z is regarded as the collective output signals of the network, then under the assumption that the output is a function of only the present state, it immediately follows that

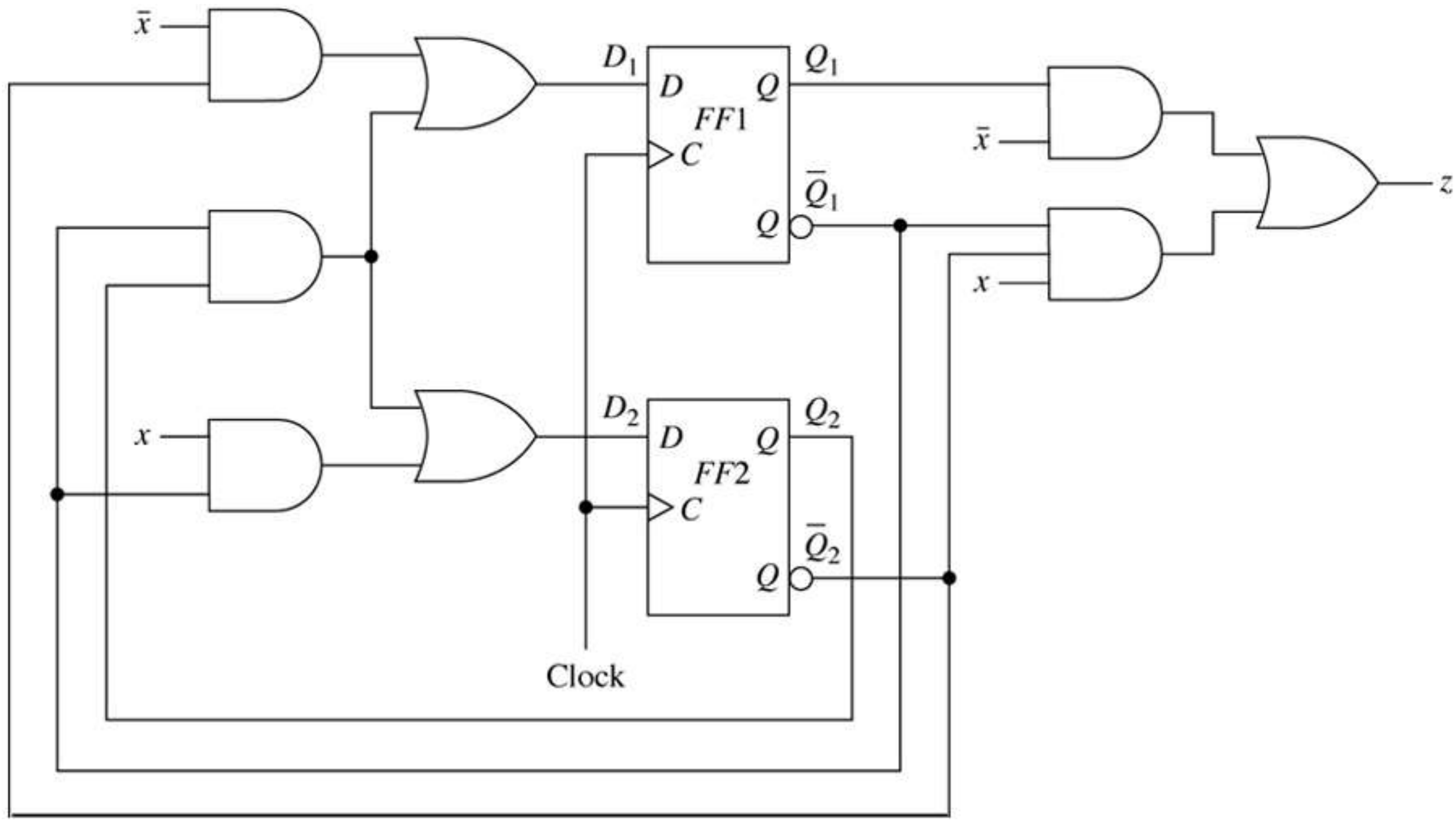
Z= g(Q)

Differences between mealy and moore models

Moore Machine	**Mealy Machine**
Output depends only upon the present state.	Output depends on the present state as well as present input.
Moore machine also places its output on the transition.	Mealy Machine places its output on the transition.
More states are required.	Less number of states are required.
There is less hardware requirement for circuit implementation.	There is more hardware requirement for circuit implementation.
They react slower to inputs(One clock cycle later).	They react faster to inputs.
Synchronous output and state generation.	Asynchronous output generation.
Output is placed on states.	Output is placed on transitions.
Easy to design.	It is difficult to design.
If input changes, output does not change	If input changes, output also changes.
Has more or the same states as that of the Mealy machine.	Has fewer or the same states as that of the Moore machine.

Logic diagram for Example 7.1.

Figure 7.5



The excitations to the flip-flops of the previousnFig. correspond to the logic values that appear at the D input terminals of flip-flops FF 1 and FF2. Algebraically,

$$D_1 = \bar{x}\overline{Q}_2 + \overline{Q}_1 Q_2$$
$$D_2 = x\overline{Q}_1 + \overline{Q}_1 Q_2$$

To complete the algebraic description of a sequential network, it is also necessary to write algebraic expressions for the network outputs. the output is given by

$$z = \bar{x}Q_1 + x\overline{Q}_1\overline{Q}_2$$

For D flip flop

Q+=D, the transition equations for this circuit would be:

Q; =D1

Q2+=D2

Substituting

$$D_1 = \overline{x}\overline{Q}_2 + \overline{Q}_1 Q_2$$
$$D_2 = x\overline{Q}_1 + \overline{Q}_1 Q_2$$

$$Q_1^+ = D_1$$
$$Q_2^+ = D_2$$

Transition table and equations

$$Q_1^+ = \overline{x}\overline{Q}_2 + \overline{Q}_1 Q_2$$
$$Q_2^+ = x\overline{Q}_1 + \overline{Q}_1 Q_2$$

Present state (Q_1Q_2)	Excitation (D_1D_2)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
00	10	01	0	1
01	11	11	0	0
10	10	00	1	0
11	00	00	1	0

Present state (Q_1Q_2)	Next state ($Q_1^+Q_2^+$)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
00	10	01	0	1
01	11	11	0	0
10	10	00	1	0
11	00	00	1	0

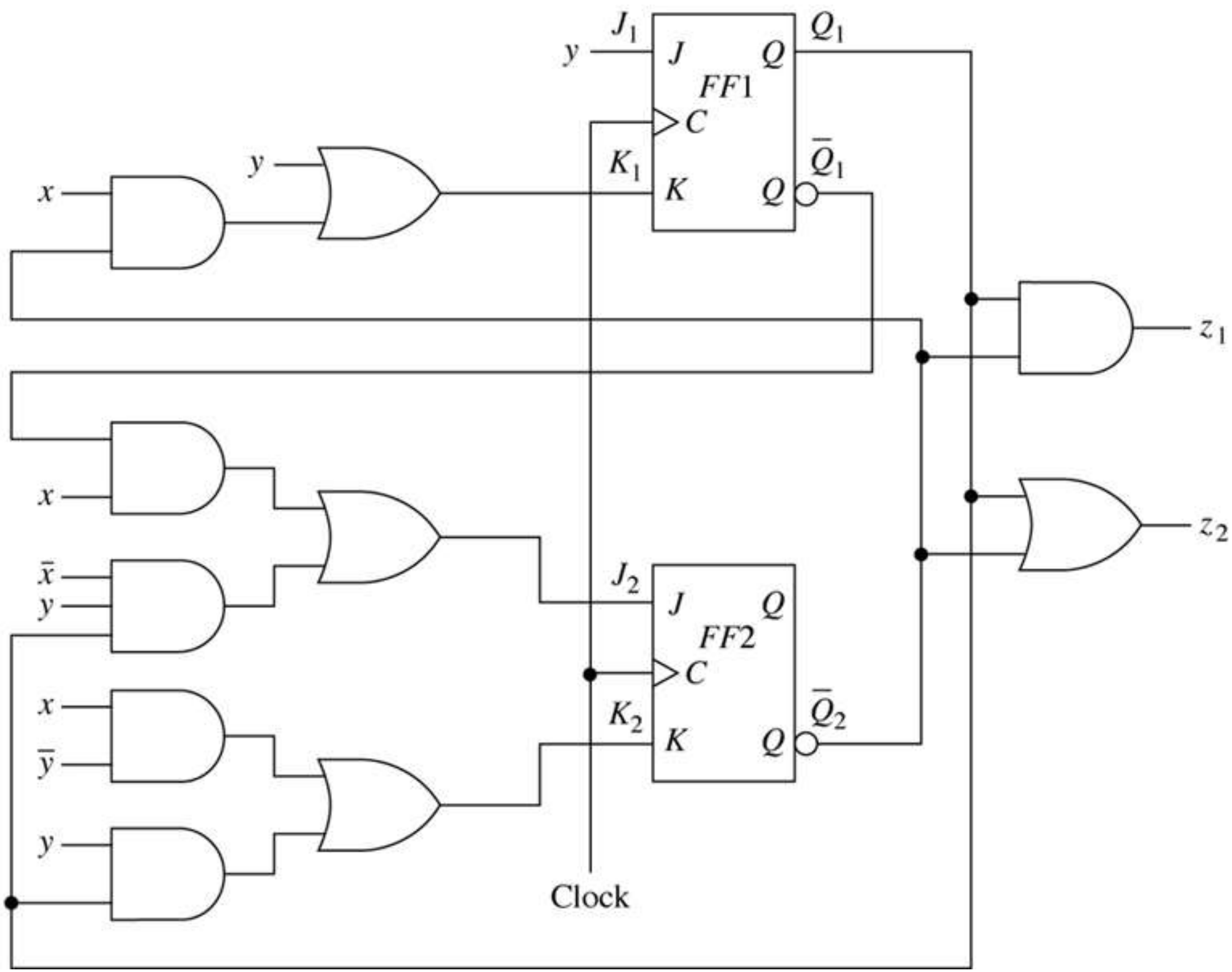
State table

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
$00 \rightarrow A$	C	B	0	1
$01 \rightarrow B$	D	D	0	0
$10 \rightarrow C$	C	A	1	0
$11 \rightarrow D$	A	A	1	0

Present state	Next state, Output (z)	
	Input (x)	
	0	1
A	C, 0	B, 1
B	D, 0	D, 0
C	C, 1	A, 0
D	A, 1	A, 0

Logic diagram for Example 7.2.

Figure 7.6



The excitation equations are as follows:

$$J_1 = y$$
$$K_1 = y + x\overline{Q}_2$$
$$J_2 = x\overline{Q}_1 + \overline{x}yQ_1$$
$$K_2 = x\overline{y} + yQ_1$$

Finally, the outputs of the sequential network are given by

$$z_1 = Q_1\overline{Q}_2$$
$$z_2 = Q_1 + \overline{Q}_2$$

Characteristic equation for a JK flip-flop is

O+=JQ'+K'Q

$$Q_1^+ = J_1\overline{Q}_1 + \overline{K}_1Q_1$$
$$Q_2^+ = J_2\overline{Q}_2 + \overline{K}_2Q_2$$

The transition equations, obtained by substituting Eqs.

$$\begin{aligned}
Q_1^+ &= y\overline{Q}_1 + \overline{(y + x\overline{Q}_2)}Q_1 \\
&= y\overline{Q}_1 + \overline{y}(\overline{x} + Q_2)Q_1 \\
&= y\overline{Q}_1 + \overline{x}\,\overline{y}Q_1 + \overline{y}Q_1Q_2 \\
Q_2^+ &= (x\overline{Q}_1 + \overline{x}yQ_1)\overline{Q}_2 + \overline{(x\overline{y} + yQ_1)}Q_2 \\
&= (x\overline{Q}_1 + \overline{x}yQ_1)\overline{Q}_2 + (\overline{x} + y)(\overline{y} + \overline{Q}_1)Q_2 \\
&= x\overline{Q}_1\overline{Q}_2 + \overline{x}yQ_1\overline{Q}_2 + \overline{x}\,\overline{y}Q_2 + \overline{x}\overline{Q}_1Q_2 + y\overline{Q}_1Q_2
\end{aligned}$$

Table Excitation table for Example

Present state (Q_1Q_2)	Excitation (J_1K_1,J_2K_2)				Output (z_1,z_2)
	Inputs (xy)				
	00	**01**	**10**	**11**	
00	00,00	11,00	01,11	11,10	01
01	00,00	11,00	00,11	11,10	00
10	00,00	11,11	01,01	11,01	11
11	00,00	11,11	00,01	11,01	01

Table Transition table for Example

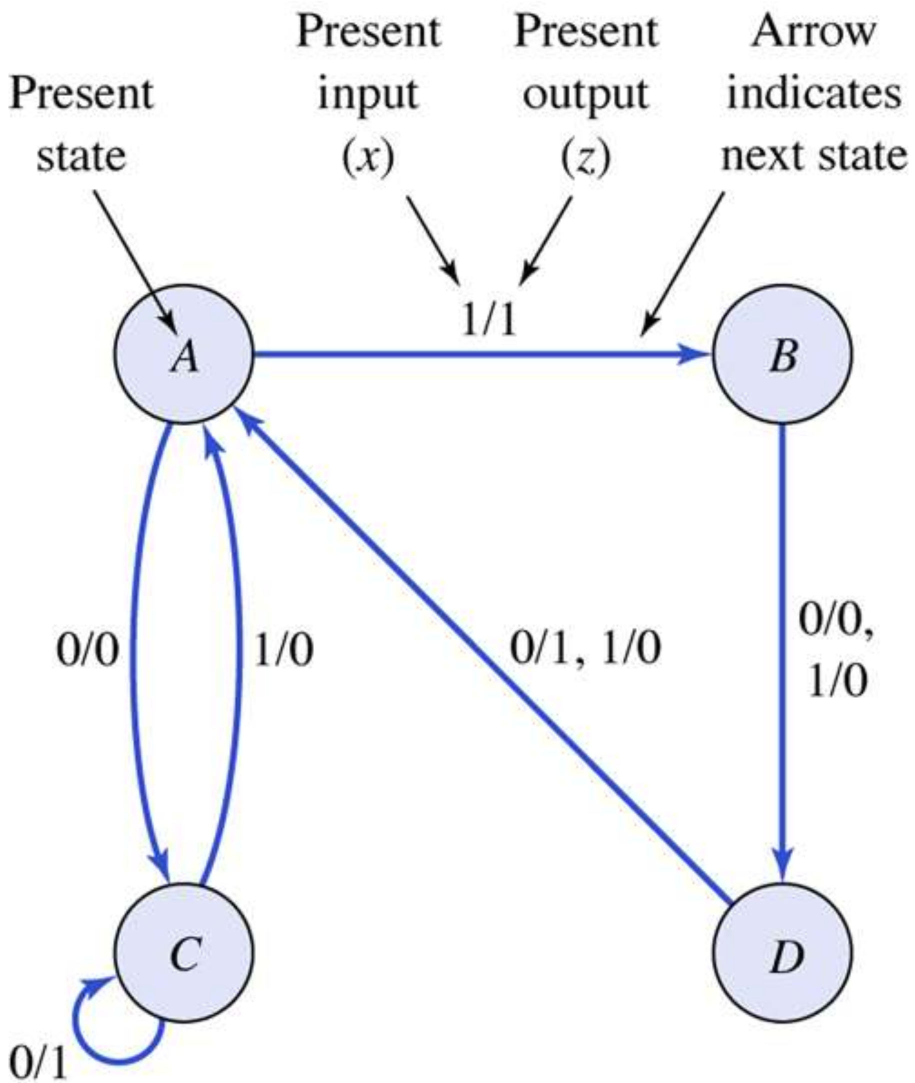
Present state (Q_1Q_2)	Next state ($Q_1^+Q_2^+$) Inputs (xy) 00	01	10	11	Output (z_1z_2)
00	00	10	01	11	01
01	01	11	00	11	00
10	10	01	00	00	11
11	11	00	10	00	01

State table

Present state	Next state				Output (z_1z_2)
	Inputs (xy)				
	00	01	10	11	
$00 \rightarrow A$	A	C	B	D	01
$01 \rightarrow B$	B	D	A	D	00
$10 \rightarrow C$	C	B	A	A	11
$11 \rightarrow D$	D	A	C	A	01

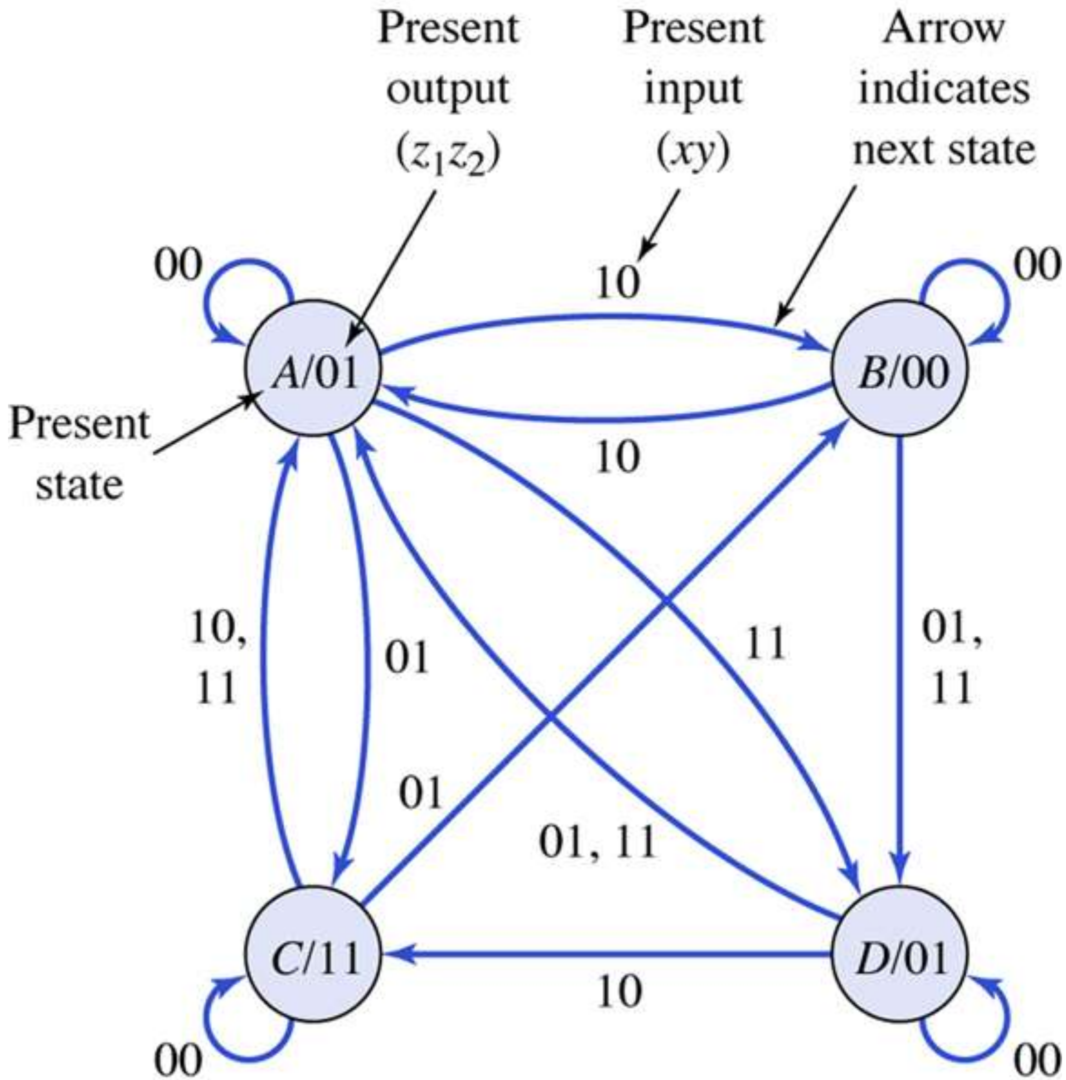
State diagram for Example 7.1.

Figure 7.7



State diagram for Example 7.2.

Figure 7.8



Assume the flip-flops are both in their 0-states, which corresponds

to state A, and the input sequence x = 0011011101 1s applied to the network.* From the general discussion on the operation of clocked

synchronous sequential networks, the inputs are assumed to be applied prior to the triggering time of the clock signal that can affect a state transition, and the effects of the inputs have propagated through the combinational logic so that final values appear at the network outputs and flip-flop inputs.

Therefore, according to Tables it is seen that when the first x = 0 input is applied to the network when in state A, the network produces a z = 0 output.

Furthermore, upon receipt of the positive edge of the clock signal, the memory portion of the network goes to state C. Next, another x = 0 is applied. Since the network is now in state C, a current output of | is produced and the network remains in state C upon receipt of the next positive edge of the clock signal. It can readily be checked that the input sequence x = 0011011101 applied to the network of Example 7.1 when initially in state A produces the following state andoutput sequences:

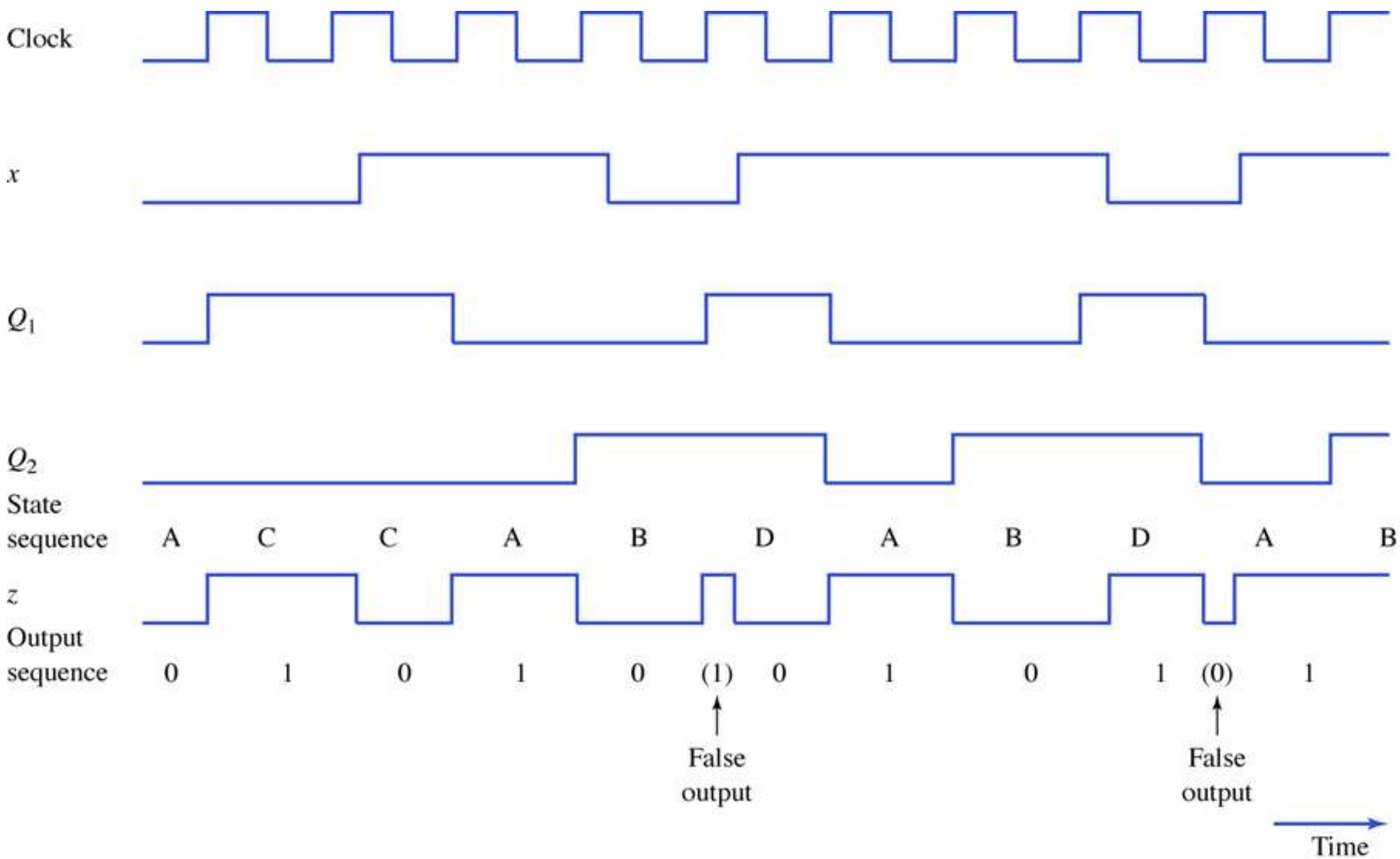
Input sequencex = 0011011101

State sequence = ACCABDABDAB

Output sequencez =O 107001011

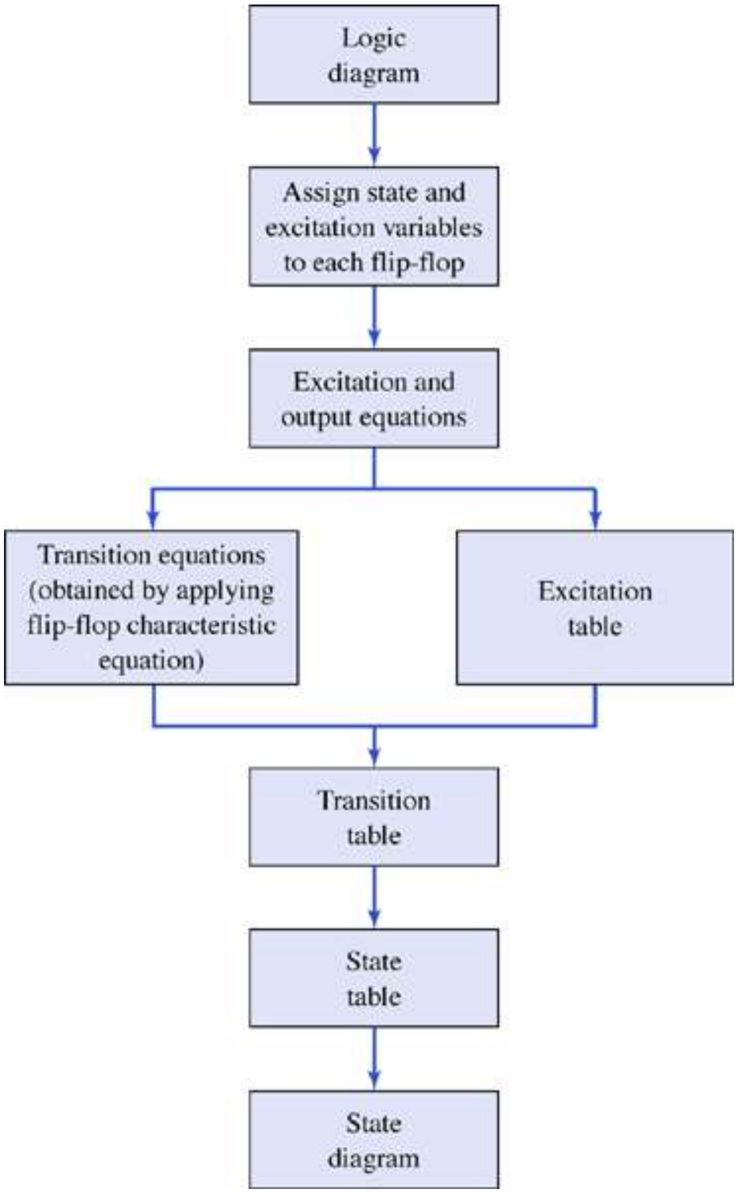
Timing diagram for Example 7.1.

Figure 7.9



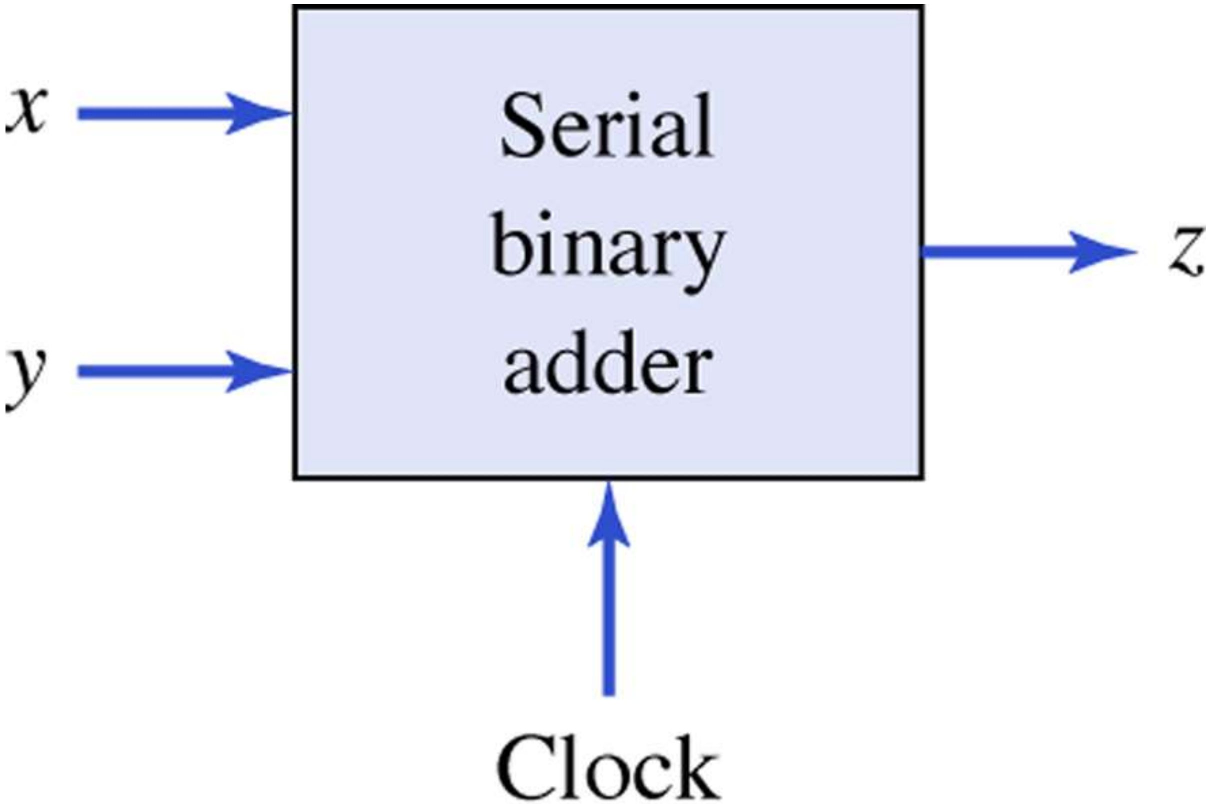
The analysis procedure.

Figure 7.10



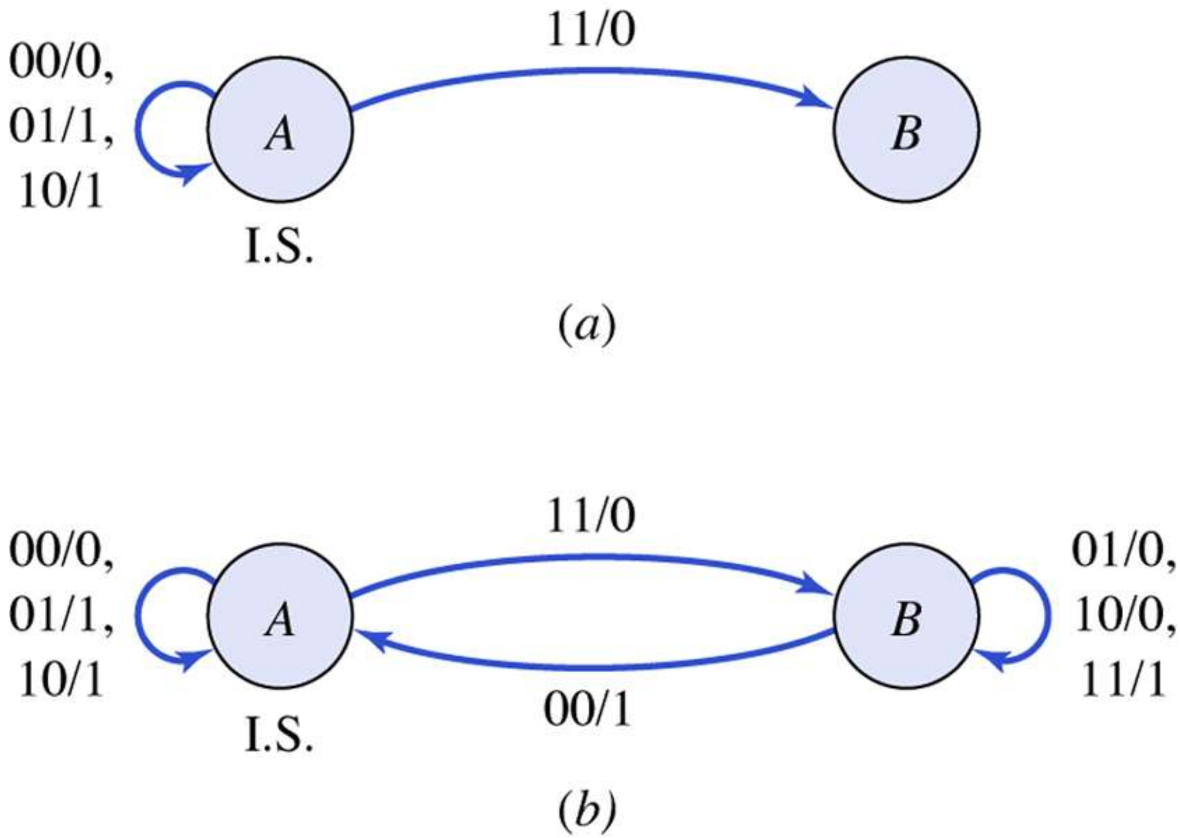
The serial binary adder.

Figure 7.11



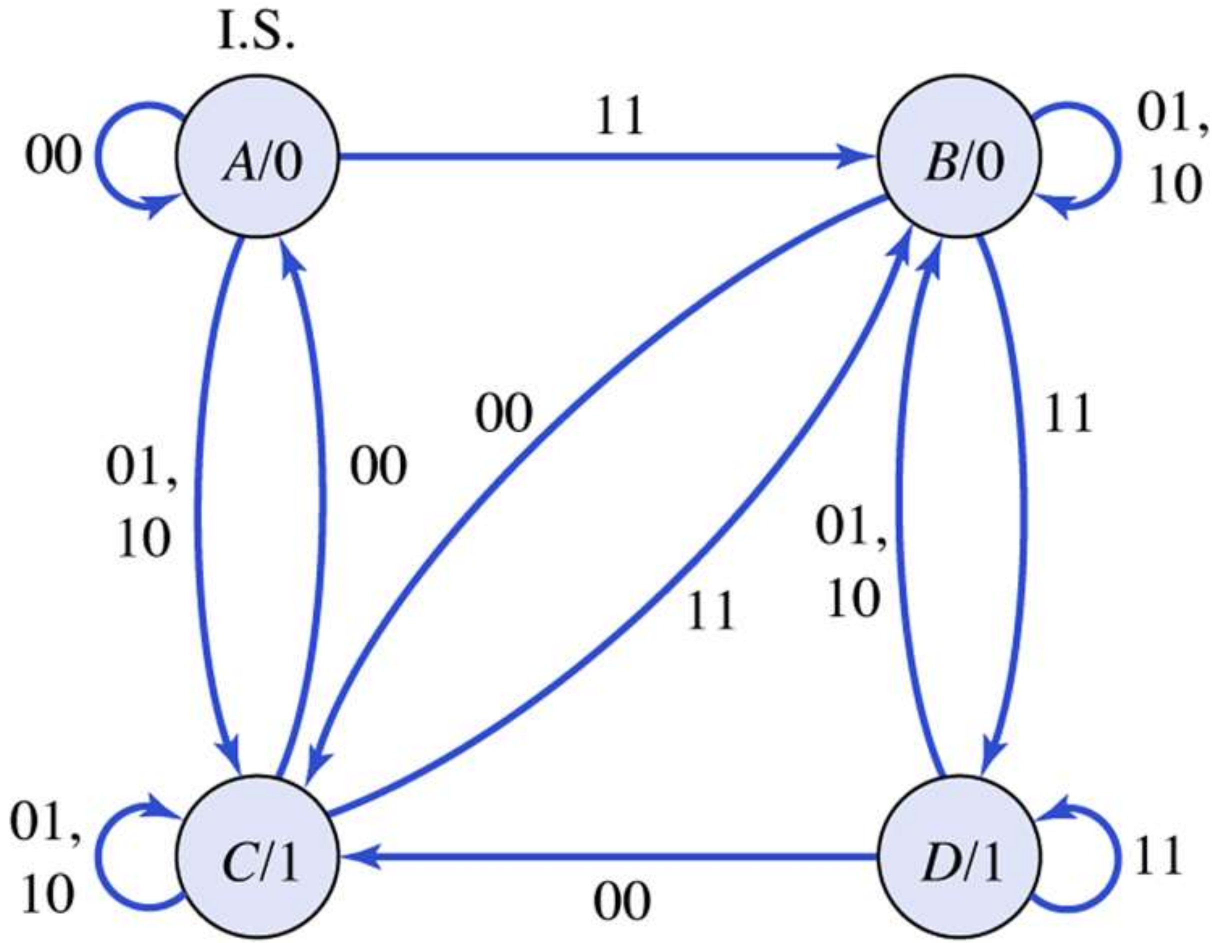
Obtaining the state diagram for a Mealy serial binary adder. (*a*) Partial state diagram. (*b*) Completed state diagram.

Figure 7.12



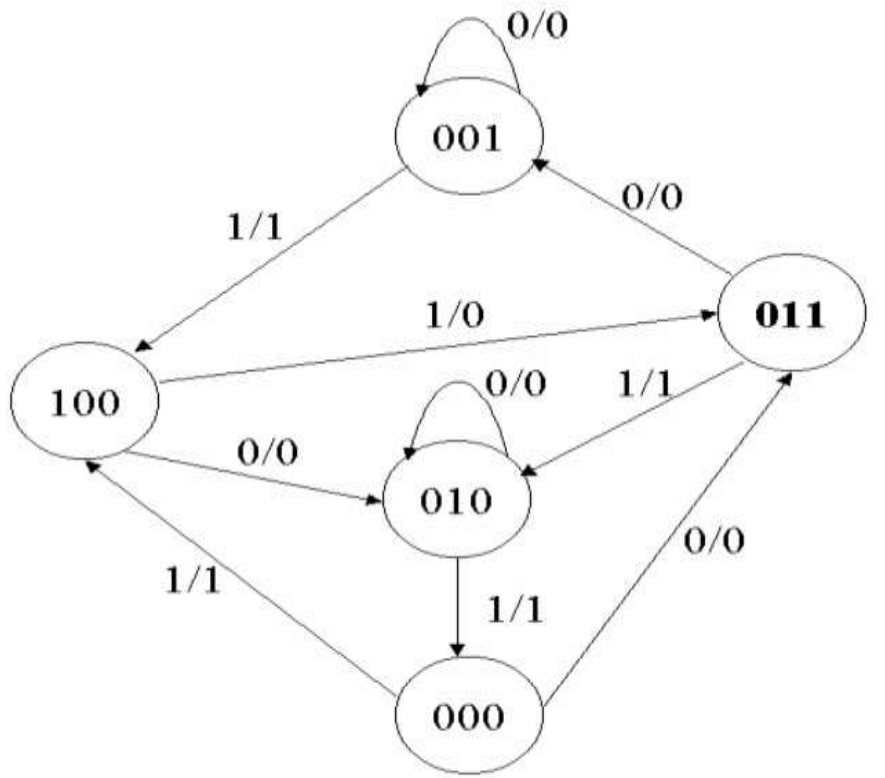
State diagram for a Moore serial binary adder.

Figure 7.13



A sequential circuit has three flip-flops A, B, C; one input x; and one output, y. The state diagram is shown in Fig.below. 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.

i) Use D flipflops ii) Use JK flipflops



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

C

AB\Cx	00	01	11	10
00		1	1	
01				
11	x	x	x	x
10			x	x

A

$D_A = A'B'X$

C

AB\Cx	00	01	11	10
00	1			
01	1		1	
11	x	x	x	x
10	1	1	x	x

A

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

C

AB\Cx	00	01	11	10
00	1			1
01				1
11	x	x	x	x
10		1	x	x

A

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

C

AB\Cx	00	01	11	10
00		1	1	
01		1	1	
11	x	x	x	x
10			x	x

A

$D_D = A'x$

b) Use JK flip-flops:

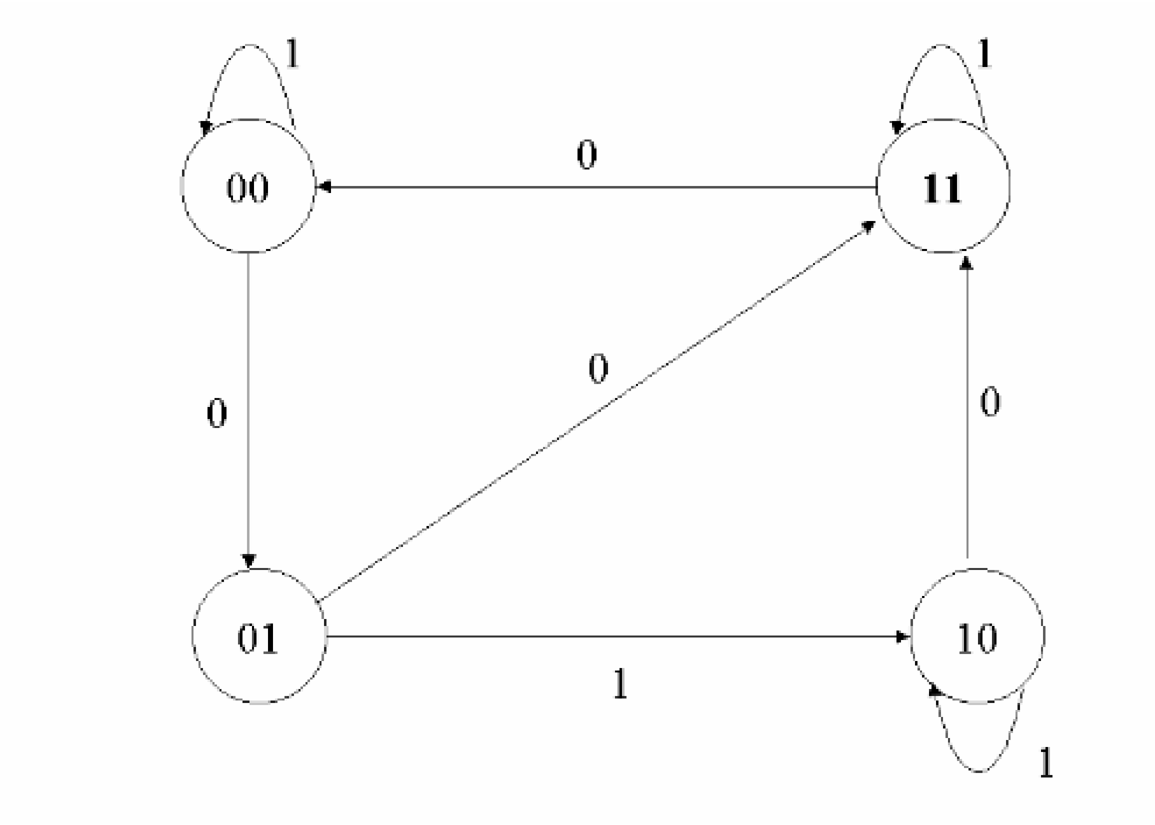
J_A	K_A	J_B	K_B	J_C	K_C
0	X	1	X	1	X
1	X	0	X	0	X
0	X	0	X	X	0
1	X	0	X	X	1
0	X	X	0	0	X
0	X	X	1	0	X
0	X	X	1	X	0
0	X	X	0	X	1
X	1	1	X	0	X
X	1	1	X	1	X

$J_A = B'x$ $\quad J_B = A + C'x'$ $\quad J_C = Ax + A'B'x'$

$K_A = 1$ $\quad K_B = C'x + Cx'$ $\quad K_C = x$

Self-correction because $K_A = 1$

Design the sequential circuit specified by the state diagram for the figure below usingT flip-flops



Draw the state table

TA (A, B, x) = $\sum$ (2, 3, 6)

TB (A, B, x) = $\sum$ (0, 3, 4, 6)

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

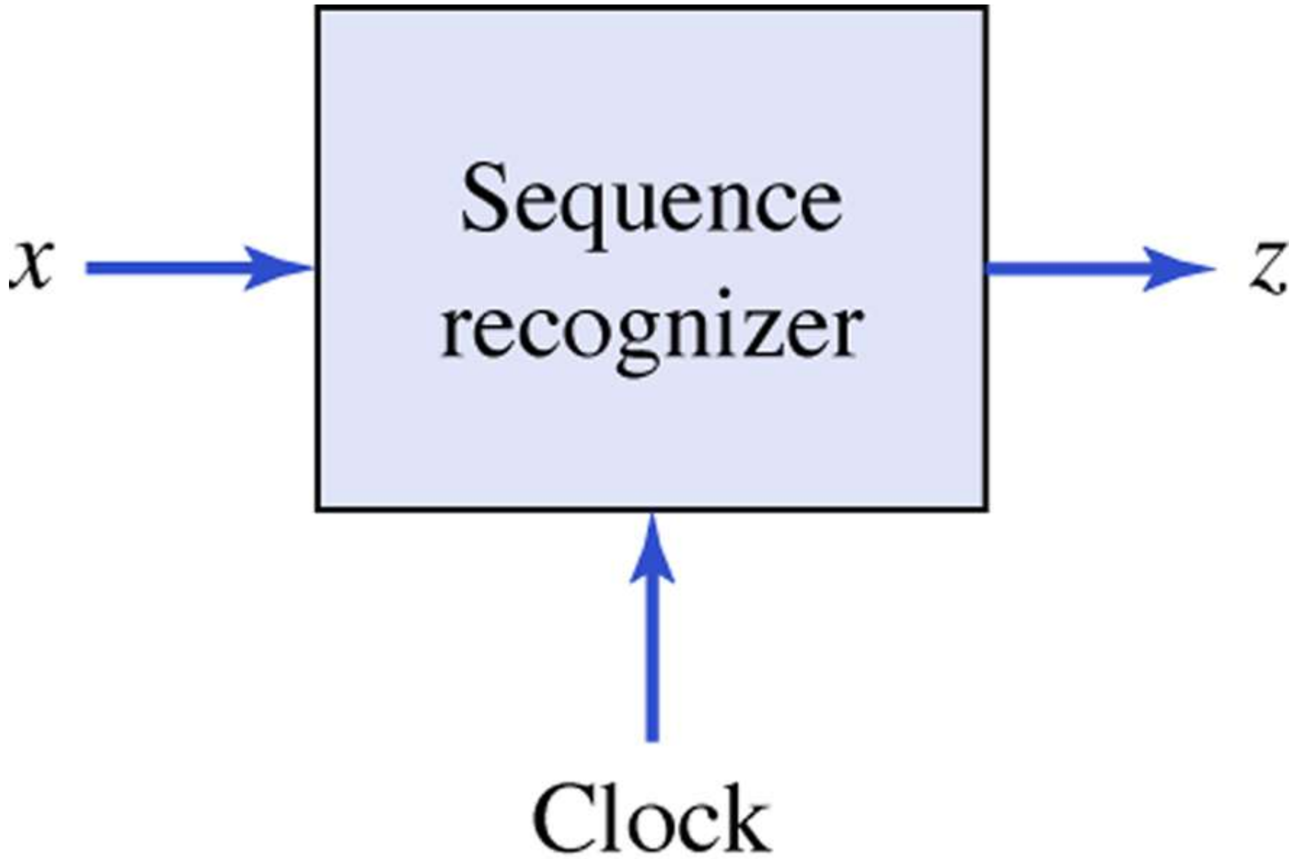
$T_B = A'B + Bx'$

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

$T_A = Ax' + B'x' + A'Bx$

A sequence recognizer.

Figure 7.14

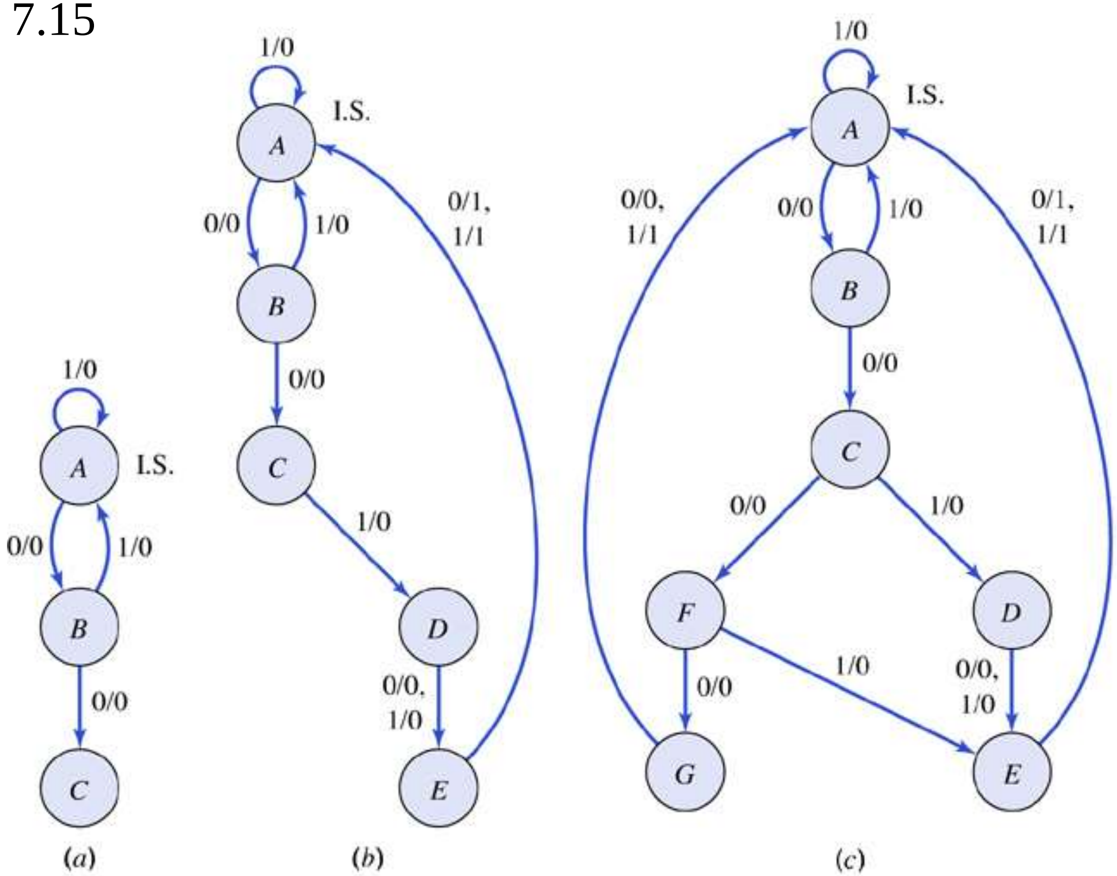


x=0100010010010010000000011'

z=0000001000000100000000001

State diagram for a sequence recognizer. (*a*) Detection of two consecutive 0’s. (*b*) Partial analysis of the three-symbol sequence. (*c*) Completed state diagram. (*d*) Definition of states.

Figure 7.15



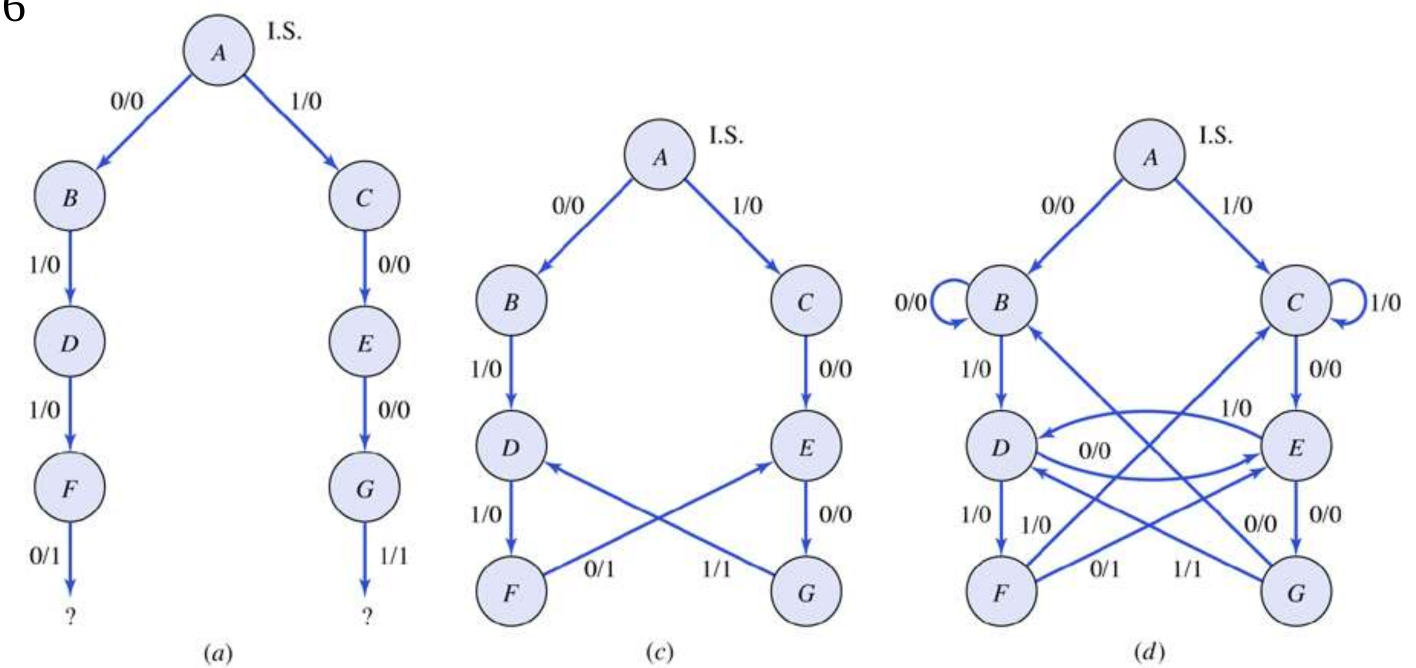
A: Waiting to detect two consecutive 0’s.
B: First 0 detected.
C: Two consecutive 0’s detected.
D: A 1 detected in three-symbol sequence, but two more inputs must still be applied.
E: At least one 1 detected in three-symbol sequence, but one more input must still be applied.
F: First input after two consecutive 0’s was 0.
G: First two inputs after two consecutive 0’s were 00.

(*d*)

$$x = 0\,0\,1\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,0\,1$$
$$z = 0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1\,0\,0$$

A 0110/1001 sequence recognizer. (*a*) Beginning the detection of the sequences 0110 or 1001. (*b*) Definition of states. (*c*) Completing the detection of the two sequences 0110 or 1001. (*d*) Completed state diagram.

Figure 7.16

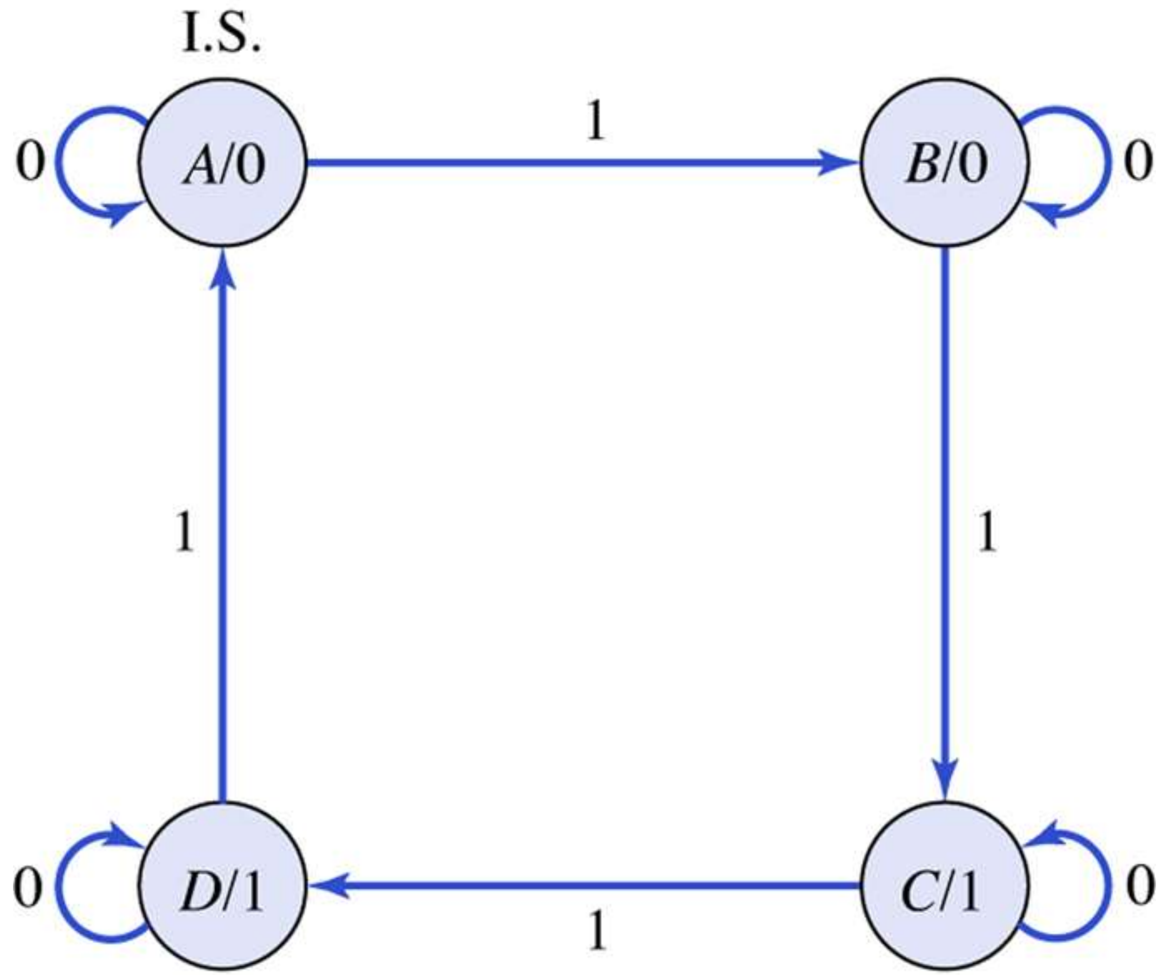


A: No inputs received (initial state).
B: Last input received was 0.
C: Last input received was 1.
D: Last two inputs received were 01.
E: Last two inputs received were 10.
F: Last three inputs received were 011.
G: Last three inputs received were 100.

(*b*)

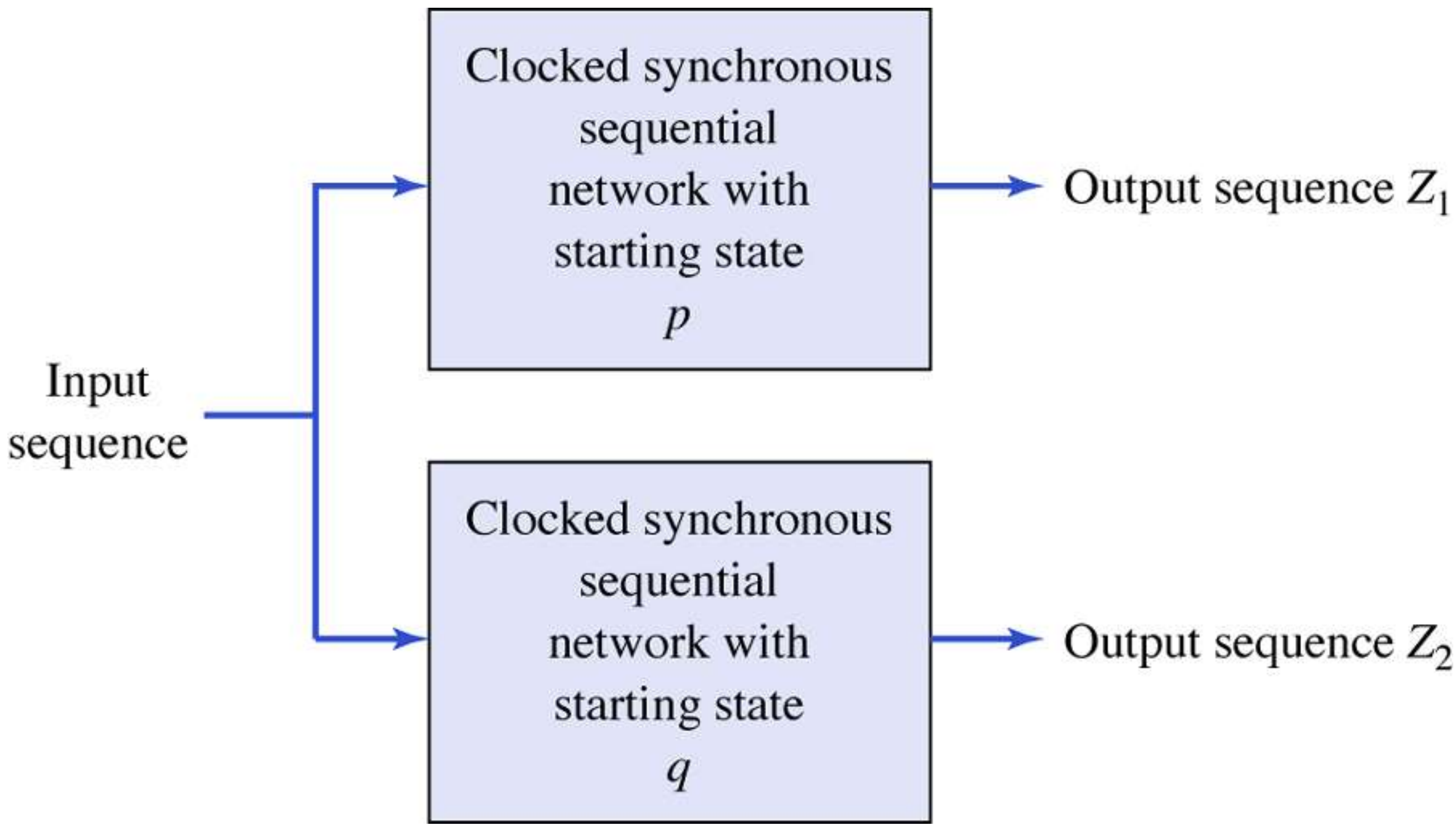
State diagram for the final example.

Figure 7.17



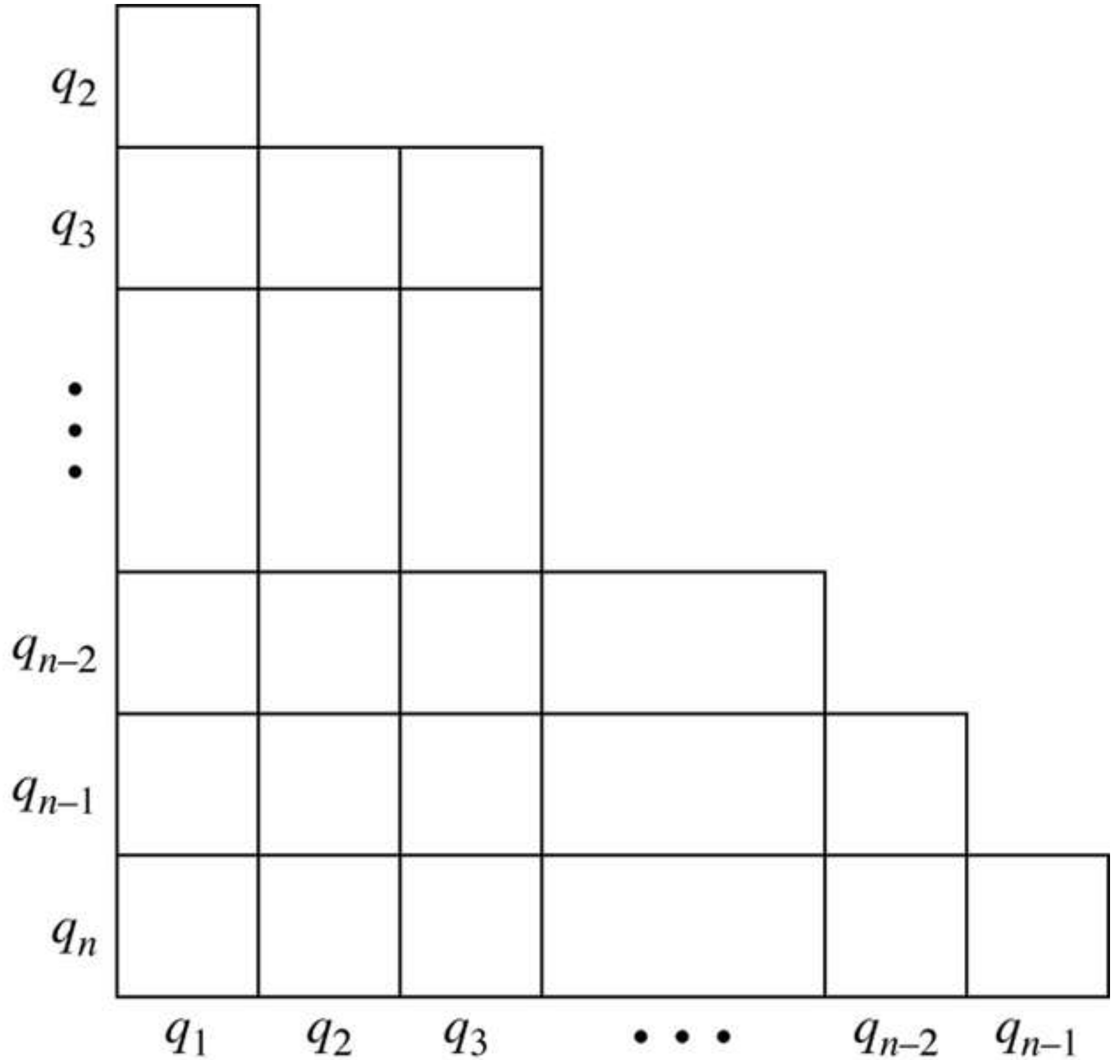
Experiment for determining equivalent pairs of states.

Figure 7.18



The structure of an implication table.

Figure 7.19

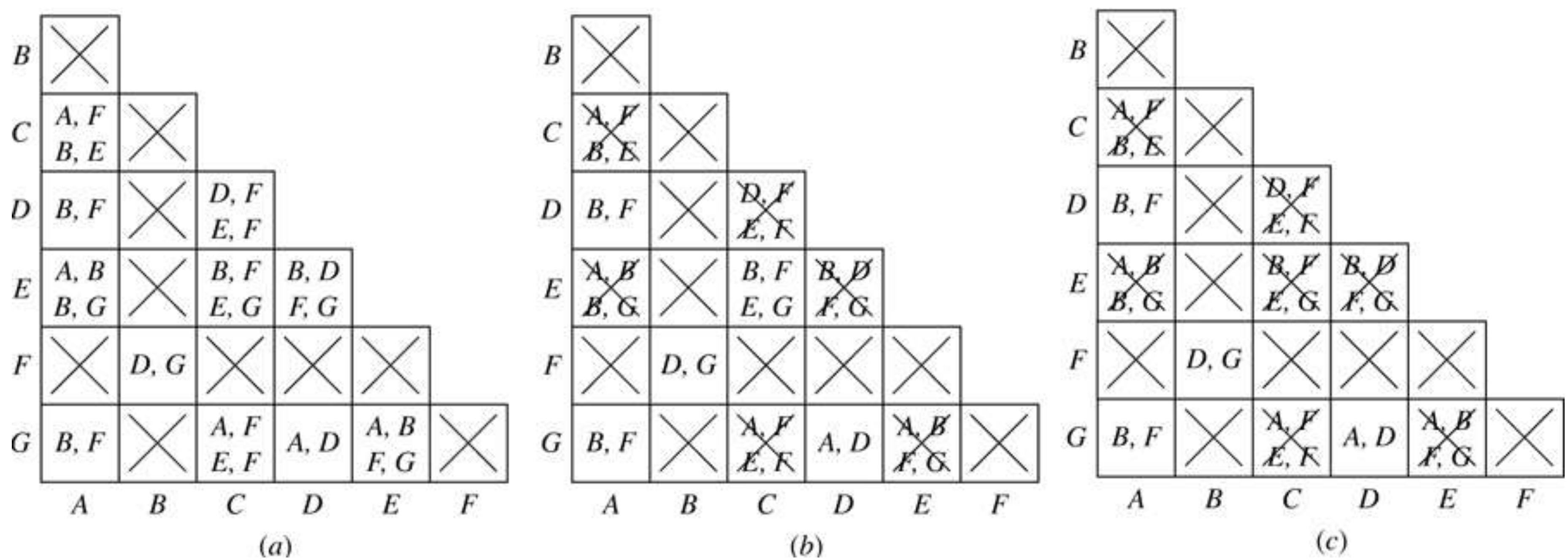


Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*A	B	A	0	0
B	C	A	0	0
C	F	D	0	0
D	E	E	0	0
E	A	A	1	1
F	G	E	0	0
G	A	A	0	1

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*A	A	B	0	0
B	D	C	0	1
C	F	E	0	0
D	D	F	0	0
E	B	G	0	0
F	G	C	0	1
G	A	F	0	0

Implication table for determining the equivalent states of Table 7.13. (*a*) The initial table. (*b*) After the first pass. (*c*) After the second pass.

Figure 7.20



Implication table for determining equivalent states of the 0110/1001 sequence recognizer. (*a*) Initial table. (*b*) Final table.

Figure 7.21

	a	b	c	d	e	f	g	h	i	j	k	l	m	n
b	b, d c, e													
c	b, f c, g	d, f e, g												
d	b, h c, i	d, h e, i	f, h g, i											
e	b, j c, k	d, j e, k	f, j g, k	h, j i, k										
f	b, l c, m	d, l e, m	f, l g, m	h, l i, m	j, l k, m									
g	b, n c, o	d, n e, o	f, n g, o	h, n i, o	j, n k, o	l, n m, o								
h	b, h c, i	d, h e, i	f, h g, i	✓	h, j i, k	h, l i, m	h, n i, o							
i	b, j c, k	d, j e, k	f, j g, k	h, j i, k	✓	j, l k, m	j, n k, o	h, j i, k						
j	b, l c, m	d, l e, m	f, l g, m	h, l i, m	j, l k, m	✓	l, n m, o	h, l i, m	j, l k, m					
k	X	X	X	X	X	X	X	X	X	X				
l	X	X	X	X	X	X	X	X	X	X	X			
m	b, j c, k	d, j e, k	f, j g, k	h, j i, k	✓	j, l k, m	j, n k, o	h, j i, k	✓	j, l k, m	X	X		
n	b, l c, m	d, l e, m	f, l g, m	h, l i, m	j, l k, m	✓	l, n m, o	h, l i, m	j, l k, m	✓	X	X	j, l k, m	
o	b, n c, o	d, n e, o	f, n g, o	h, n i, o	j, n k, o	l, n m, o	✓	h, n i, o	j, n k, o	l, n m, o	X	X	j, n k, o	l, n m, o

(*a*)

	a	b	c	d	e	f	g	h	i	j	k	l	m	n
b	~~b, d c, e~~													
c	~~b, f c, g~~	~~d, f e, g~~												
d	~~b, h c, i~~	d, h e, i	~~f, h g, i~~											
e	~~b, j c, k~~	~~d, j e, k~~	~~f, j g, k~~	~~h, j i, k~~										
f	~~b, l c, m~~	~~d, l e, m~~	~~f, l g, m~~	~~h, l i, m~~	~~j, l k, m~~									
g	~~b, n c, o~~	~~d, n e, o~~	f, n g, o	~~h, n i, o~~	~~j, n k, o~~	~~l, n m, o~~								
h	~~b, h c, i~~	d, h e, i	~~f, h g, i~~	✓	~~h, j i, k~~	~~h, l i, m~~	~~h, n i, o~~							
i	~~b, j c, k~~	~~d, j e, k~~	~~f, j g, k~~	~~h, j i, k~~	✓	~~j, l k, m~~	~~j, n k, o~~	~~h, j i, k~~						
j	~~b, l c, m~~	~~d, l e, m~~	~~f, l g, m~~	~~h, l i, m~~	~~j, l k, m~~	✓	~~l, n m, o~~	~~h, l i, m~~	~~j, l k, m~~					
k	X	X	X	X	X	X	X	X	X	X				
l	X	X	X	X	X	X	X	X	X	X	X			
m	~~b, j c, k~~	~~d, j e, k~~	~~f, j g, k~~	~~h, j i, k~~	✓	~~j, l k, m~~	~~j, n k, o~~	~~h, j i, k~~	✓	~~j, l k, m~~	X	X		
n	~~b, l c, m~~	~~d, l e, m~~	~~f, l g, m~~	~~h, l i, m~~	~~j, l k, m~~	✓	~~l, n m, o~~	~~h, l i, m~~	~~j, l k, m~~	✓	X	X	~~j, l k, m~~	
o	~~b, n c, o~~	~~d, n e, o~~	f, n g, o	~~h, n i, o~~	~~j, n k, o~~	~~l, n m, o~~	✓	~~h, n i, o~~	~~j, n k, o~~	~~l, n m, o~~	X	X	~~j, n k, o~~	~~l, n m, o~~

(*b*)

State assignment rules/guidelines

Define two states as being adjacent if their binary codes differ in exactly one bit.

Similarly, two input combinations are considered adjacent if they differ in exactly one bit. The following rules suggest which states should be made adjacent in an attempt to obtain simple Boolean expressions for the next-state and output functions.

Rule I: Two or more present states that have the same next state for a given input combination should be made adjacent.

Rule II: For any present state and two adjacent input combinations, the two next states should be made adjacent.

Rule III: Two or more present states that produce the same output symbol, i.e., 0 or 1, for a given input combination should be made adjacent. This rule need only be applied to one of the output symbols.

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(a)

It is noted that state B is the next state for both present states B and G when the input is x = 0. Rule I suggests that states B and G should have adjacent codes. Similarly, for the same input, states C and F should be coded as ad- jacent states since their next states are both state D, and states D and H should be coded as adjacent states since their next states are both state F. Referring to the column for x = 1, it is concluded that the pairs of states (A,E), (B,G), and (D,H) should each be coded as adjacent states.

To summarize, Rule I proposes the following adja-cency conditions should be attempted:

Rule I: (B,G)(2X), (C,F), (D,H)(2X), (A,E£)

The (2) following the pairs (B,G) and (D,A) indicates that the recommended adjacency conditions appear twice and should be given higher priority than those that appear only once.

Next consider Rule II. Since x = 0 and x = | are adjacent input

combinations, the next-state pair for each present state should be made adjacent according to Rule II. Thus, Rule II recommends the following state adjacencies:

Rule II: (A,B), (B,C)(3X), (D,E), (F,G)(2X), (D,A)

Finally, in an attempt to group the | entries in the output map, Rule III proposes the

following pairs of states should be adjacent:

Rule III: (D,F), (E,G)

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignme based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(a)

Present state ($Q_1Q_2Q_3$)	**Next state** ($Q_1^+Q_2^+Q_3^+$)		**Output** (z)	
	Input (x)		**Input** (x)	
	0	**1**	**0**	**1**
$^*A \rightarrow 000$	000	001	0	0
$B \rightarrow 001$	001	010	0	0
$C \rightarrow 010$	011	100	0	0
$D \rightarrow 011$	101	110	1	0
$E \rightarrow 100$	010	001	0	1
$F \rightarrow 101$	011	111	1	0
$G \rightarrow 110$	001	010	0	1
$H \rightarrow 111$	101	110	0	0

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$)		Output (z)	
	Input (x)		Input (x)	
	0	**1**	**0**	**1**
*$A \rightarrow 000$	000	001	0	0
$B \rightarrow 001$	001	010	0	0
$C \rightarrow 010$	011	100	0	0
$D \rightarrow 011$	101	110	1	0
$E \rightarrow 100$	010	001	0	1
$F \rightarrow 101$	011	111	1	0
$G \rightarrow 110$	001	010	0	1
$H \rightarrow 111$	101	110	0	0

$$Q_1^+ = Q_2Q_3 + \overline{Q}_1Q_2x + Q_1Q_3x$$

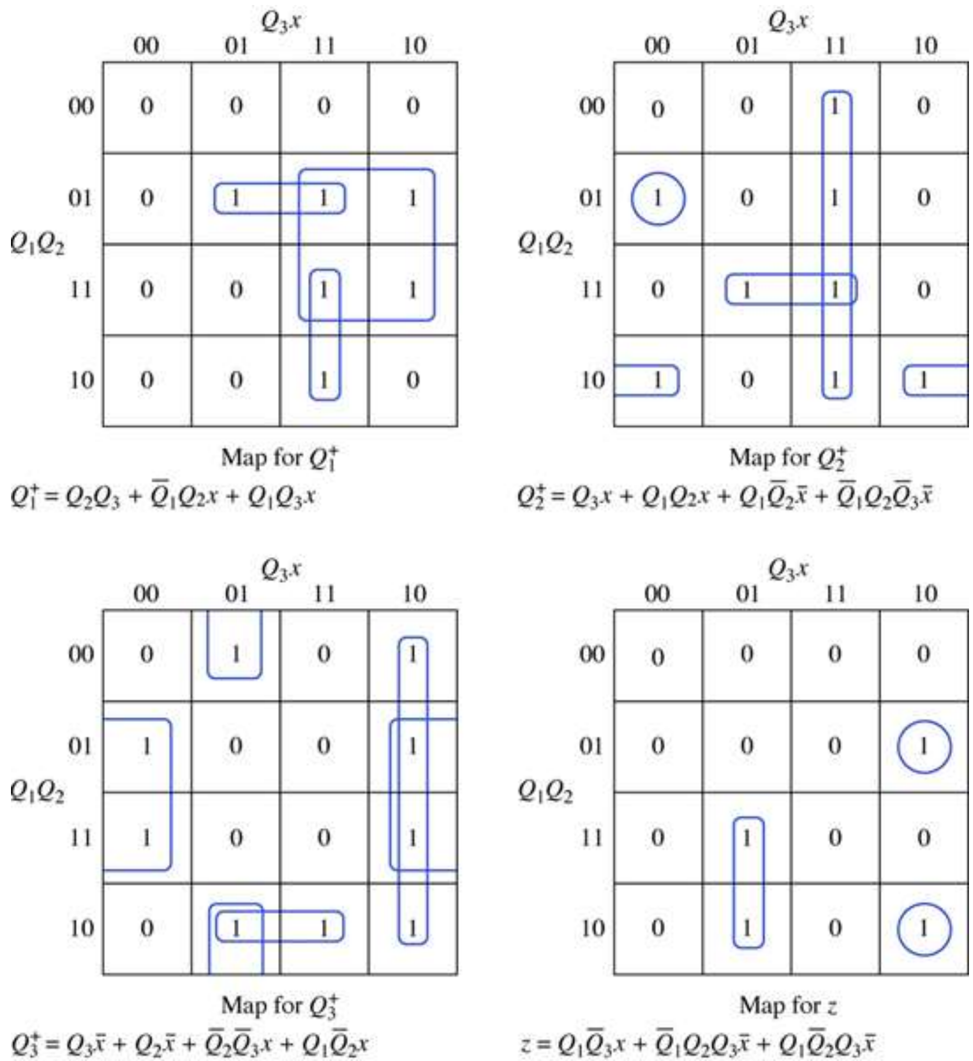
$$Q_2^+ = Q_3x + Q_1Q_2x + Q_1\overline{Q}_2\overline{x} + \overline{Q}_1Q_2\overline{Q}_3\overline{x}$$

$$Q_3^+ = Q_3\overline{x} + Q_2\overline{x} + \overline{Q}_2\overline{Q}_3x + Q_1\overline{Q}_2x$$

$$z = Q_1\overline{Q}_3x + \overline{Q}_1Q_2Q_3\overline{x} + Q_1\overline{Q}_2Q_3\overline{x}$$

Next-state and output Karnaugh maps for the transition table of Table 7.17*b*.

Figure 7.22



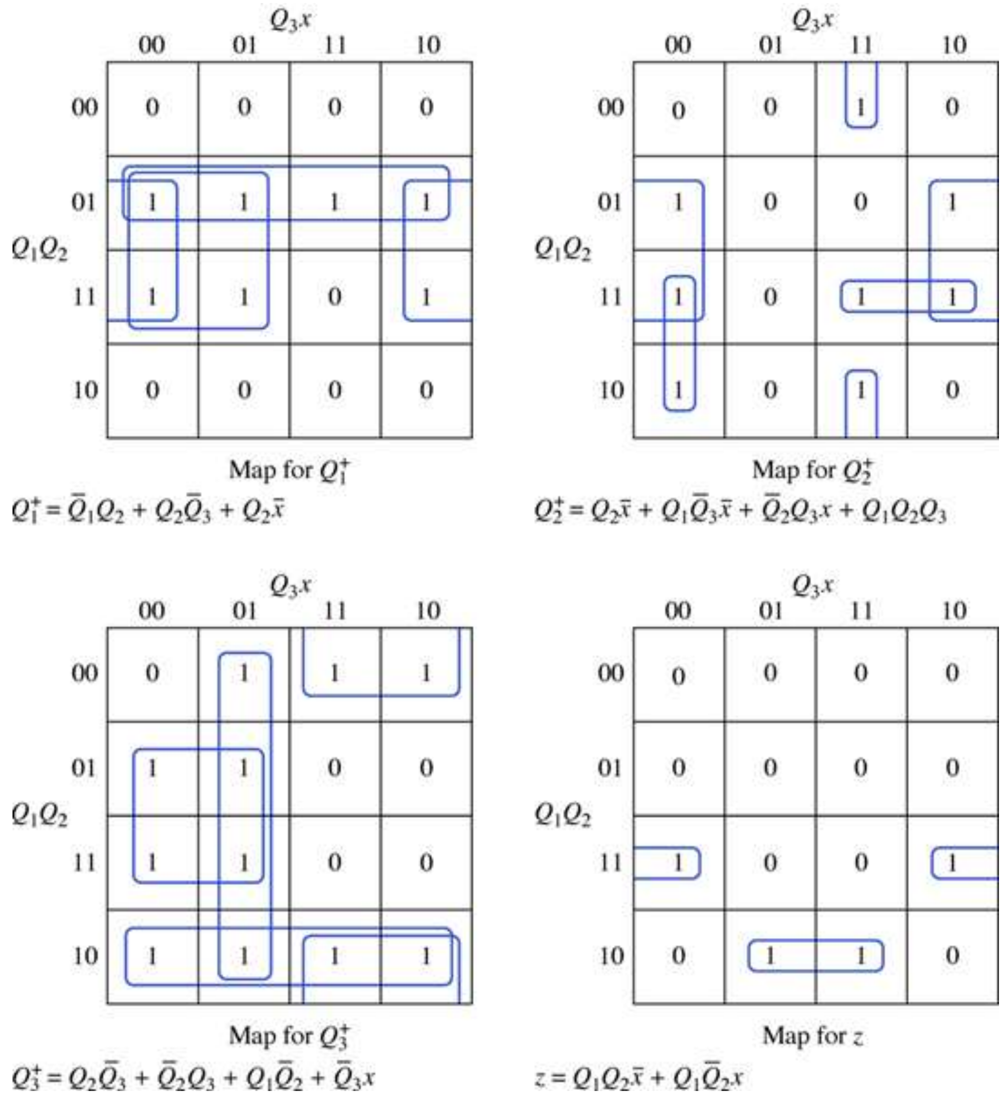
A state-assignment map for the state table of Table 7.17*a.*

Figure 7.23

Q_1 \ Q_2Q_3	00	01	11	10
0	A	B	C	H
1	E	G	F	D

Next-state and output Karnaugh maps for the transition table of Table 7.17*c.*

Figure 7.24



Two approaches to handling unused states. (*a*) State table. (*b*) Transition table with don't-cares for unused states. (*c*) Next-state maps, output map, and expressions for table of Fig. 7.25*b*.

Figure 7.25

Present state	Next state, Input (x) 0	Next state, Input (x) 1	Output (z), Input (x) 0	Output (z), Input (x) 1
*A	A	B	0	0
B	C	D	1	0
C	A	D	0	1
D	E	A	1	1
E	C	B	0	0

(*a*)

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$), Input (x) 0	Next state ($Q_1^+Q_2^+Q_3^+$), Input (x) 1	Output (z), Input (x) 0	Output (z), Input (x) 1
$A \rightarrow$ 000	000	001	0	0
$B \rightarrow$ 001	010	011	1	0
$C \rightarrow$ 010	000	011	0	1
$D \rightarrow$ 011	100	000	1	1
$E \rightarrow$ 100	010	001	0	0
101	–	–	–	–
110	–	–	–	–
111	–	–	–	–

(*b*)

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	–	–	–	–
10	0	0	–	–

Map for Q_1^+

$Q_1^+ = Q_2Q_3\bar{x}$

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	–	–	–	–
10	1	0	–	–

Map for Q_2^+

$Q_2^+ = \bar{Q}_2Q_3 + Q_1\bar{x} + Q_2\bar{Q}_3x$

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	1	1	0
01	0	1	0	0
11	–	–	–	–
10	0	1	–	–

Map for Q_3^+

$Q_3^+ = \bar{Q}_3x + \bar{Q}_2x$

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	0	0	1
01	0	1	1	1
11	–	–	–	–
10	0	0	–	–

Map for z

$z = Q_2x + Q_3\bar{x}$

(*c*)

(*d*) Transition table when unused states cause the network to go to state *A*. (*e*) Next-state maps, output map, and expressions for table of Fig. 7.25*d*.

Figure 7.25 cont.

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$) Input (x) 0	1	Output (z) Input (x) 0	1
$A \rightarrow$ 000	000	001	0	0
$B \rightarrow$ 001	010	011	1	0
$C \rightarrow$ 010	000	011	0	1
$D \rightarrow$ 011	100	000	1	1
$E \rightarrow$ 100	010	001	0	0
101	000	000	0	0
110	000	000	0	0
111	000	000	0	0

(*d*)

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	0	0	0
10	0	0	0	0

Map for Q_1^+

$Q_1^+ = \overline{Q}_1Q_2Q_3\overline{x}$

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	0	0	0	0
10	1	0	0	0

Map for Q_2^+

$Q_2^+ = \overline{Q}_1\overline{Q}_2Q_3 + \overline{Q}_1Q_2\overline{Q}_3x + Q_1\overline{Q}_2\overline{Q}_3\overline{x}$

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	1	1	0
01	0	1	0	0
11	0	0	0	0
10	0	1	0	0

Map for Q_3^+

$Q_3^+ = \overline{Q}_1\overline{Q}_2x + \overline{Q}_1\overline{Q}_3x + \overline{Q}_2\overline{Q}_3x$

Q_1Q_2 \ Q_3x	00	01	11	10
00	0	0	0	1
01	0	1	1	1
11	0	0	0	0
10	0	0	0	0

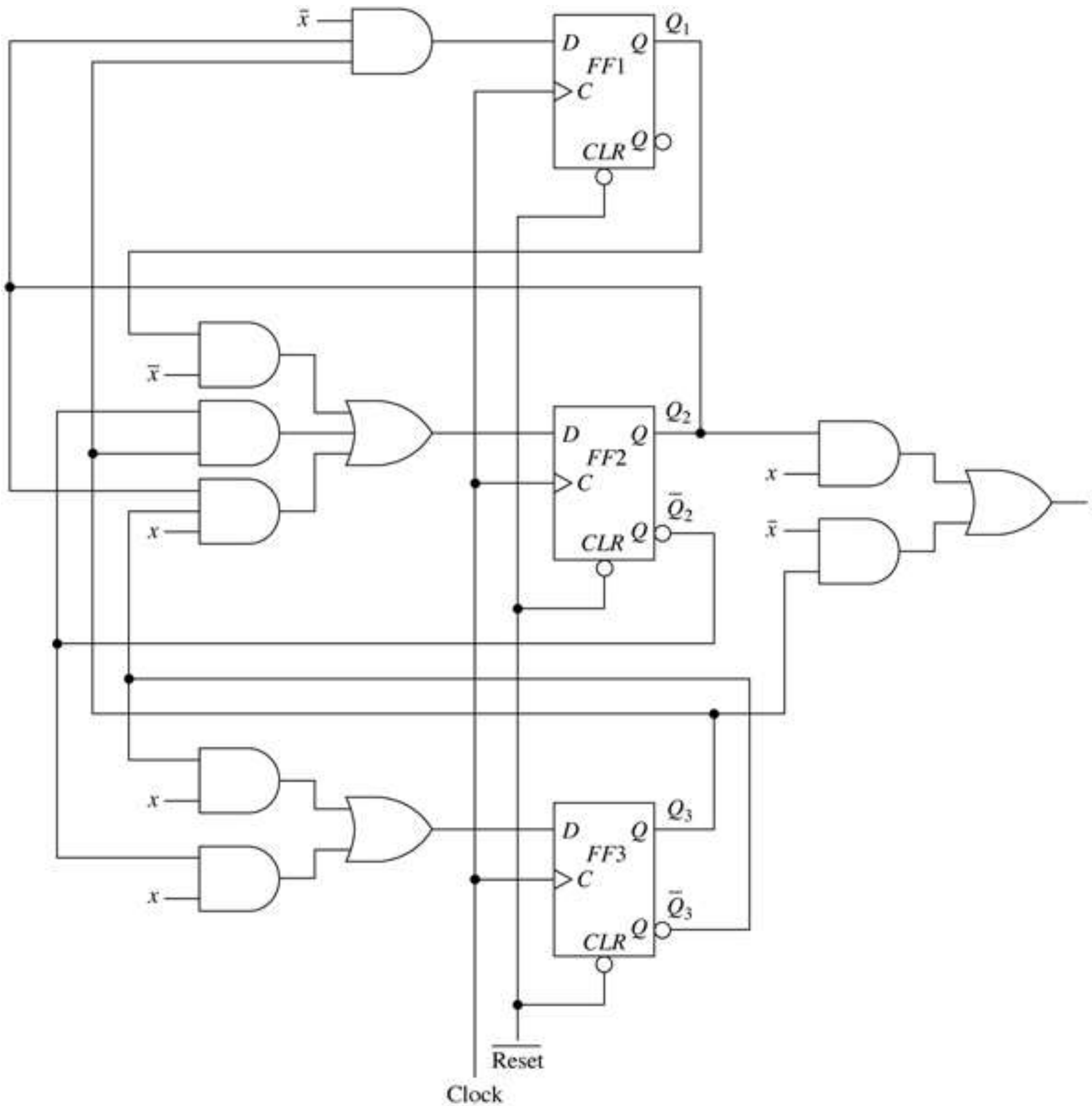
Map for z

$z = \overline{Q}_1Q_2x + \overline{Q}_1Q_3\overline{x}$

(*e*)

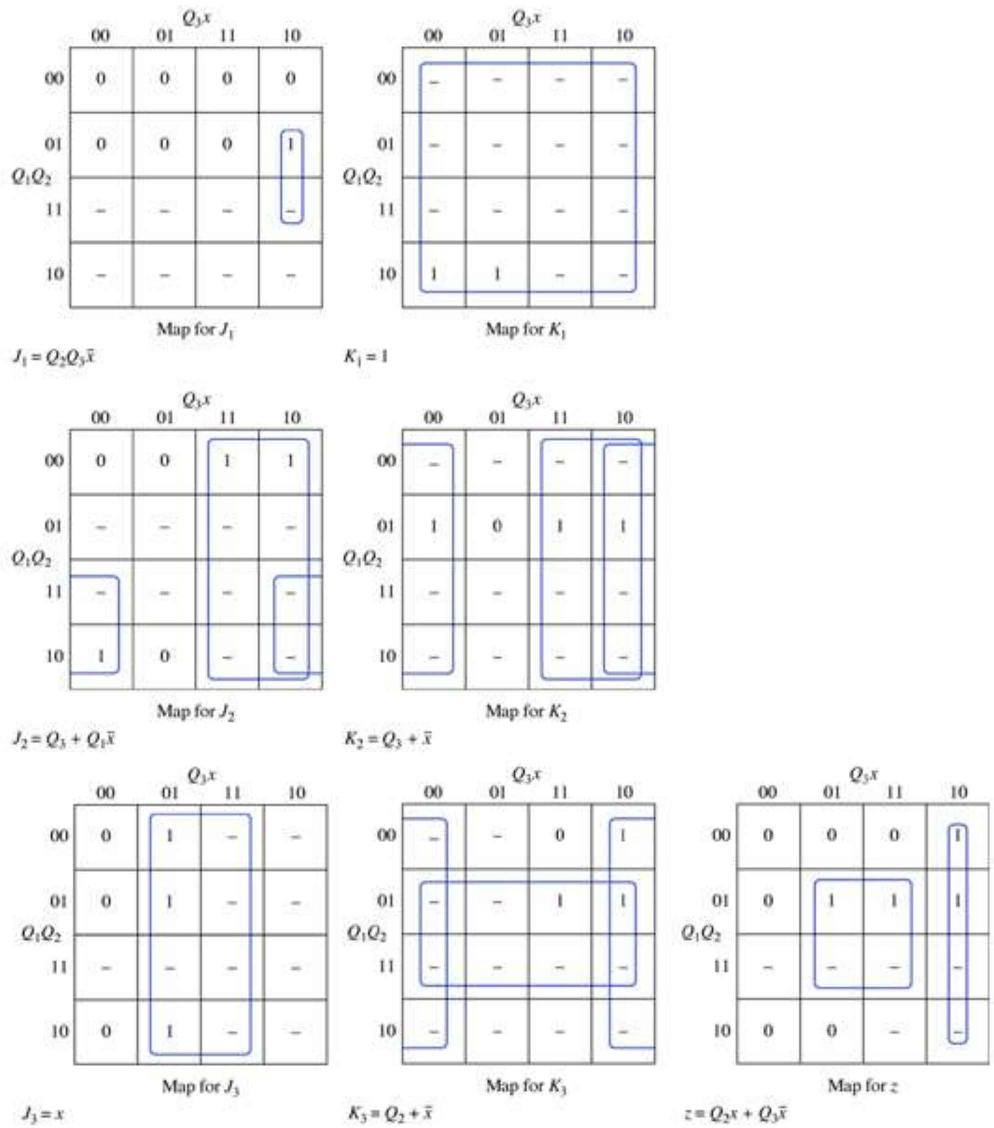
Logic diagram for the excitation table of Table 7.19.

Figure 7.26



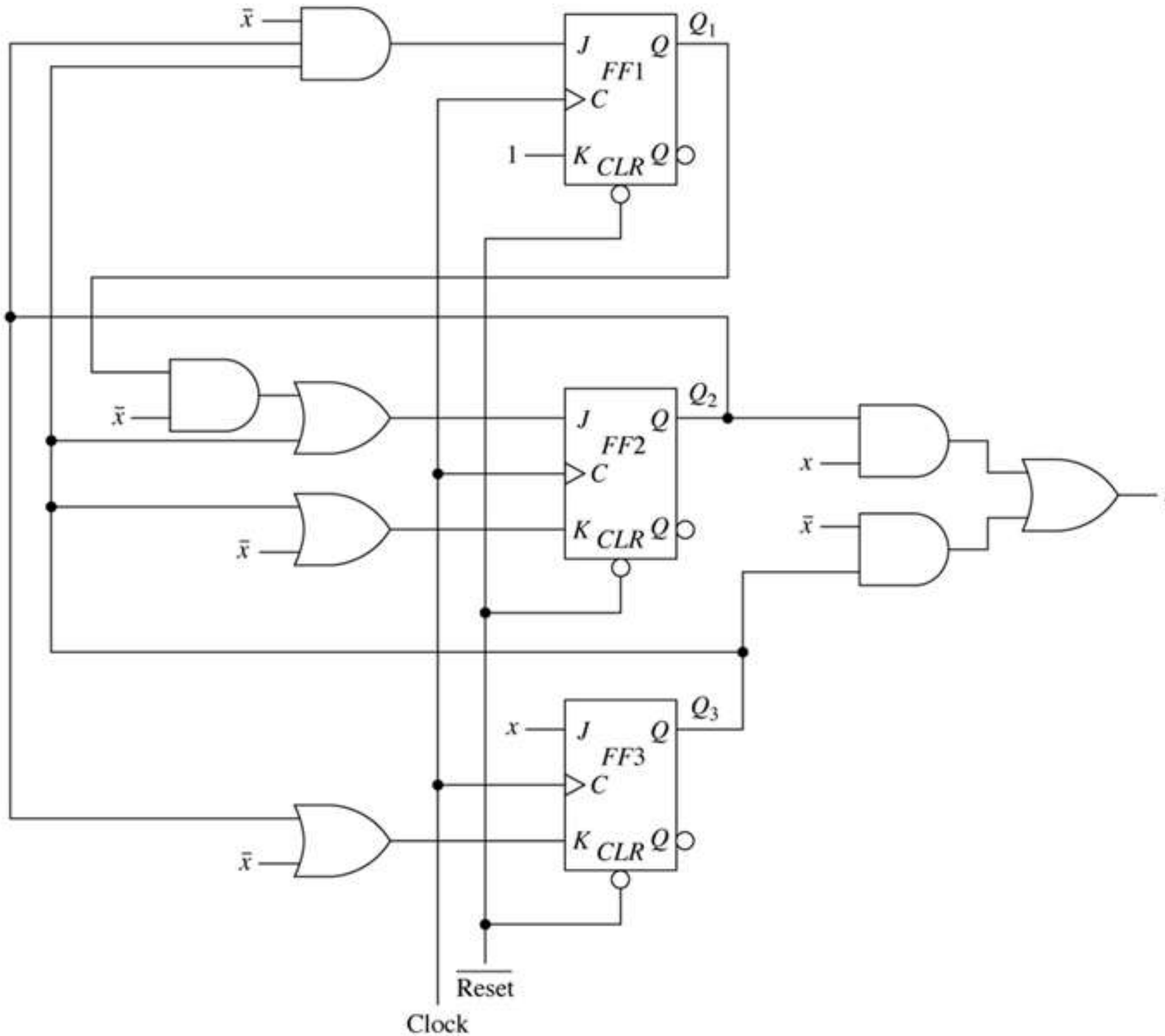
Excitation and output maps for the excitation table of Table 7.20.

Figure 7.27



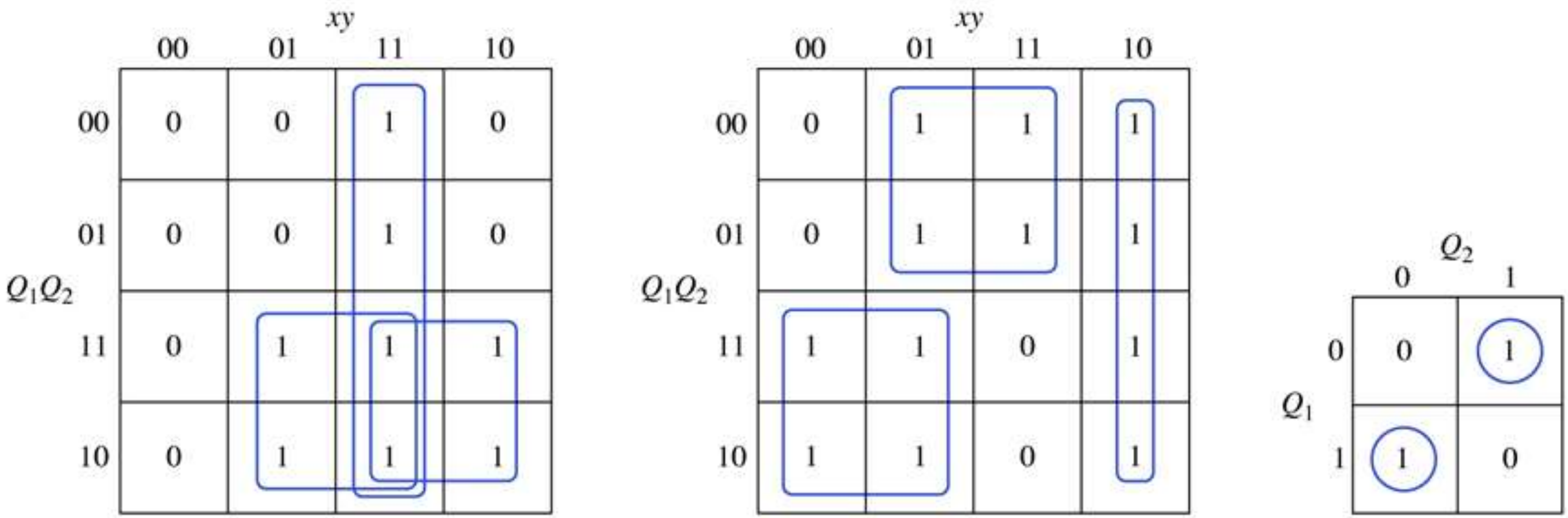
Logic diagram for the excitation table of Table 7.20.

Figure 7.28



Excitation and output maps for the Moore serial binary adder.

Figure 7.29



Map for $Q_1^+ = D_1$

$D_1 = Q_1^+ = xy + Q_1x + Q_1y$

Map for $Q_2^+ = D_2$

$D_2 = Q_2^+ + x\bar{y} + Q_1\bar{x} + \bar{Q}_1y$

Map for z

$z = Q_1\bar{Q}_2 + \bar{Q}_1Q_2 = Q_1 \oplus Q_2$

Logic diagram for the Moore serial binary adder.

Figure 7.30