SEQUENTIAL LOGIC DESIGN

The FLIP-FLOP is a basic element of sequential logic system. Using FLIP-FLOPs and combinational logic circuits, any sequential logic circuit can be designed.

REGISTERS

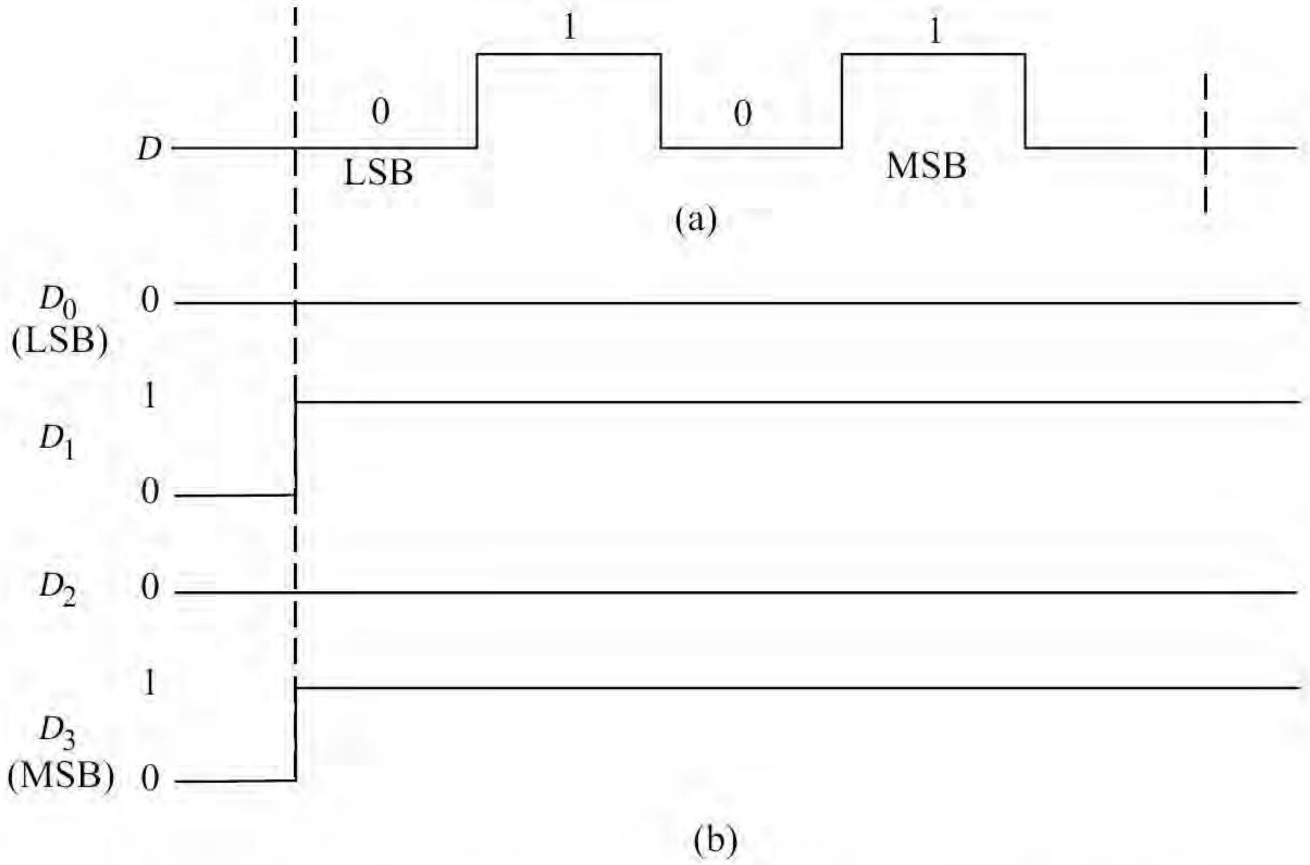


Fig. 8.1 ***Data Representation in (a) Serial Form (b) Parallel Form***

1. Serial-in, serial-out (SISO),
2. Serial-in, parallel-out (SIPO),
3. Parallel-in, serial-out (PISO), and
4. Parallel-in, parallel-out (PIPO).

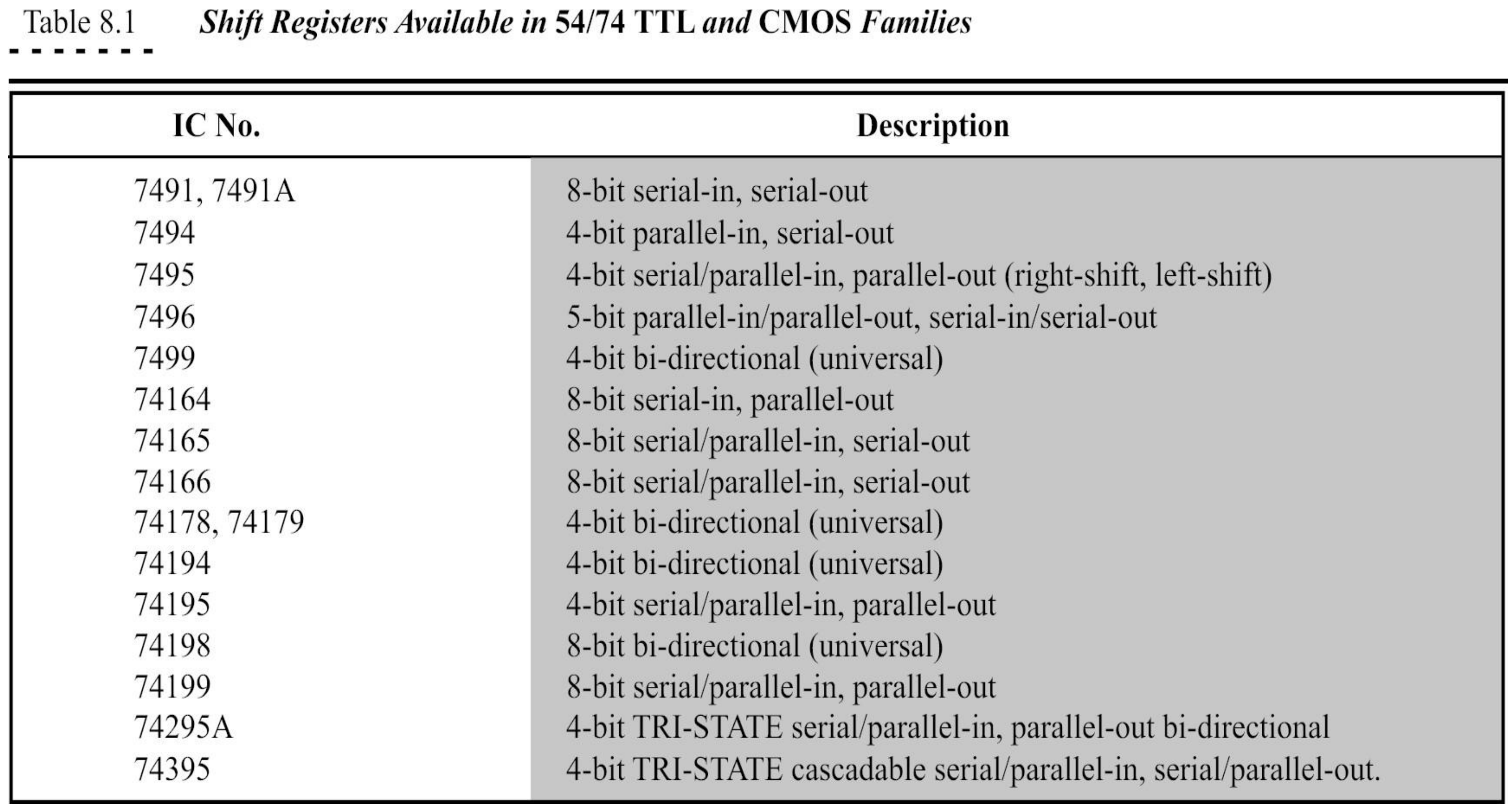


Table 8.1 ***Shift Registers Available in* 54/74 TTL *and* CMOS *Families***

IC No.	Description
7491, 7491A	8-bit serial-in, serial-out
7494	4-bit parallel-in, serial-out
7495	4-bit serial/parallel-in, parallel-out (right-shift, left-shift)
7496	5-bit parallel-in/parallel-out, serial-in/serial-out
7499	4-bit bi-directional (universal)
74164	8-bit serial-in, parallel-out
74165	8-bit serial/parallel-in, serial-out
74166	8-bit serial/parallel-in, serial-out
74178, 74179	4-bit bi-directional (universal)
74194	4-bit bi-directional (universal)
74195	4-bit serial/parallel-in, parallel-out
74198	8-bit bi-directional (universal)
74199	8-bit serial/parallel-in, parallel-out
74295A	4-bit TRI-STATE serial/parallel-in, parallel-out bi-directional
74395	4-bit TRI-STATE cascadable serial/parallel-in, serial/parallel-out.

Shift Register

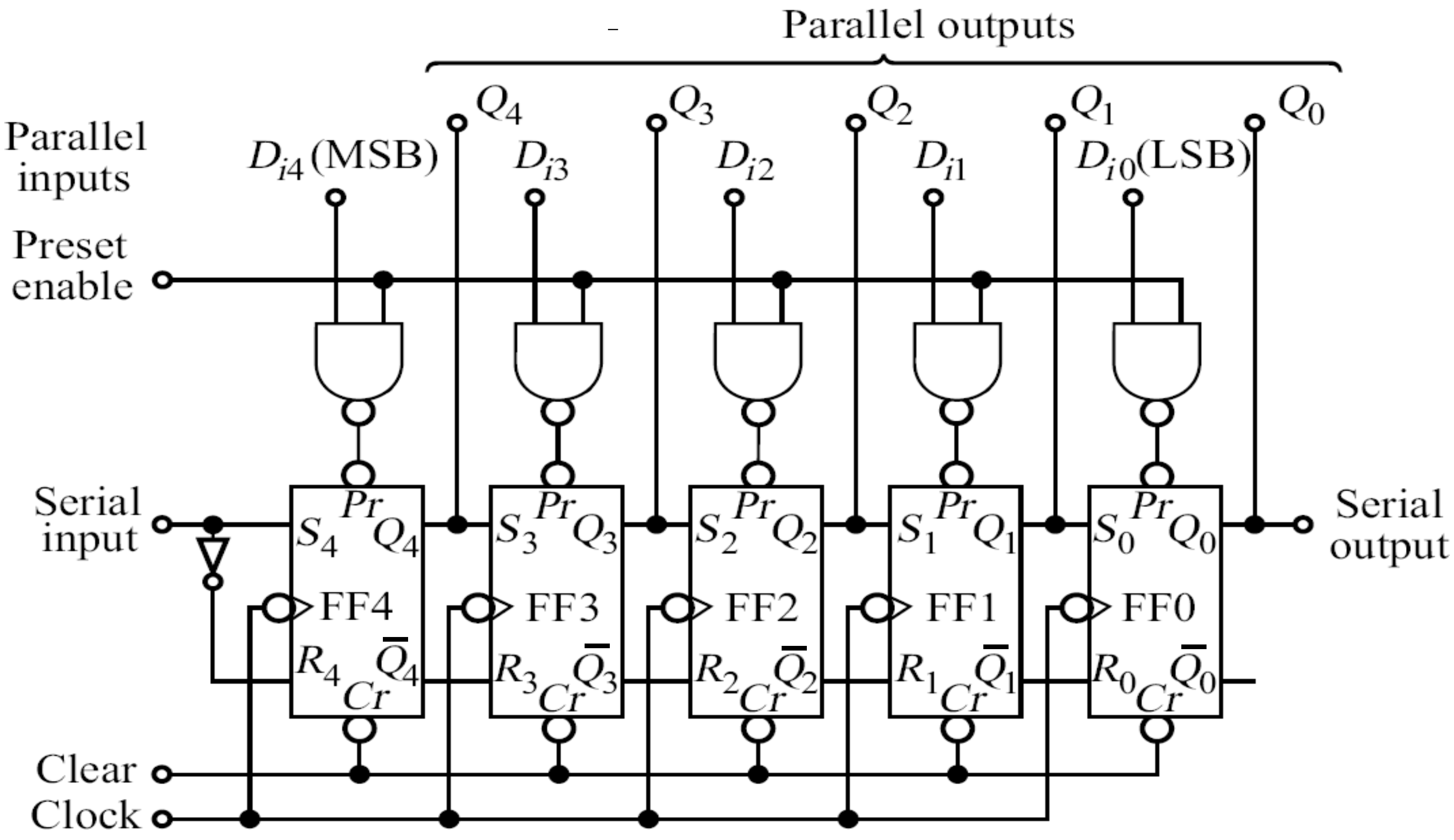
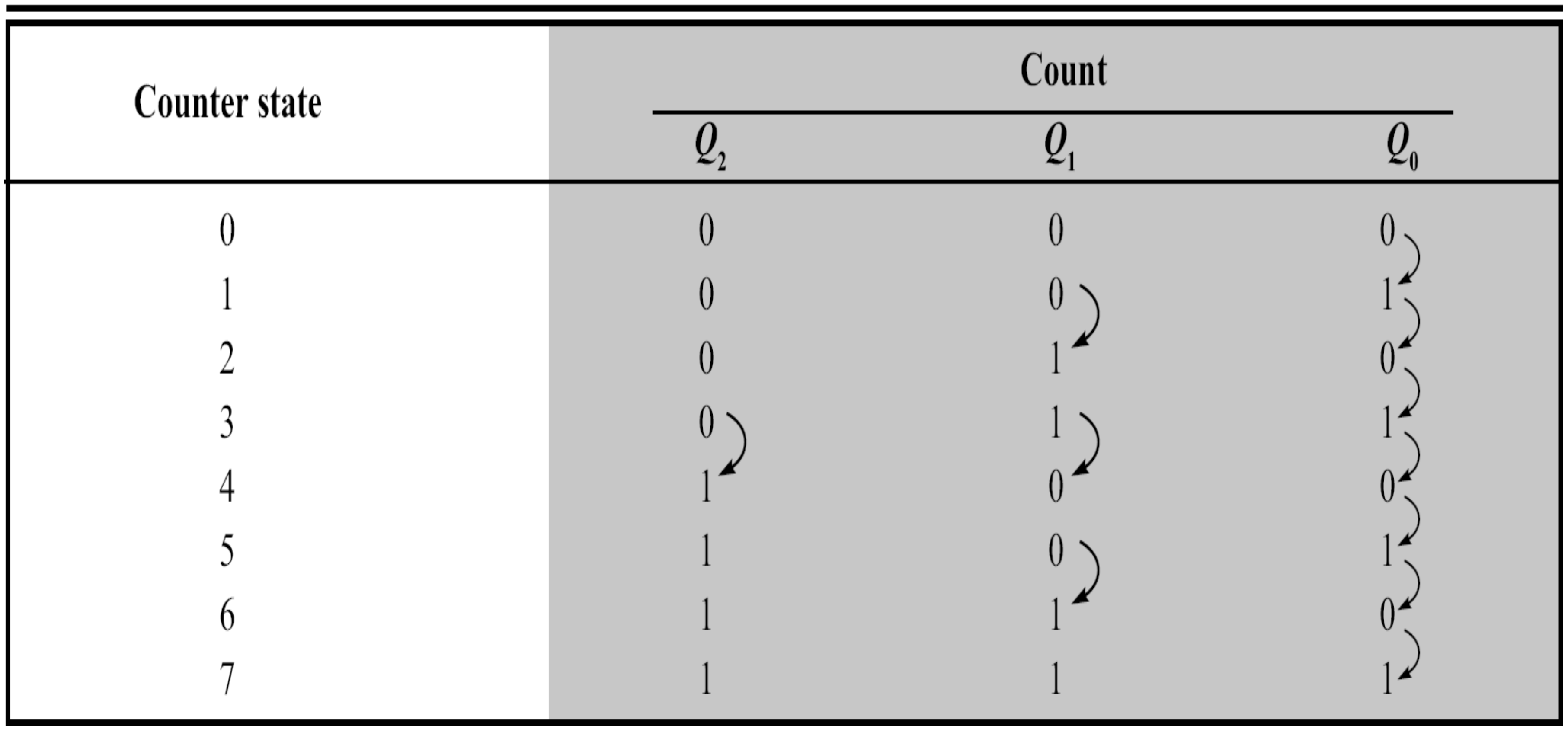

Parallel outputs
Q_4
Q_3
Q_2
Q_1
Q_0
Parallel inputs
D_{i4} (MSB)
D_{i3}
D_{i2}
D_{i1}
D_{i0}(LSB)
Preset enable
Serial input
S_4 Pr Q_4 FF4 R_4 Cr $\overline{Q}_4$
S_3 Pr Q_3 FF3 R_3 Cr $\overline{Q}_3$
S_2 Pr Q_2 FF2 R_2 Cr $\overline{Q}_2$
S_1 Pr Q_1 FF1 R_1 Cr $\overline{Q}_1$
S_0 Pr Q_0 FF0 R_0 Cr $\overline{Q}_0$
Serial output
Clear
Clock

Fig. 8.2 ***A 5-bit Shift Register* (7496)**

RIPPLE OR ASYNCHRONOUS COUNTERS

Table 8.4 *Counting Sequence of a 3-bit Binary Counter*



Counter state	Count		
	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

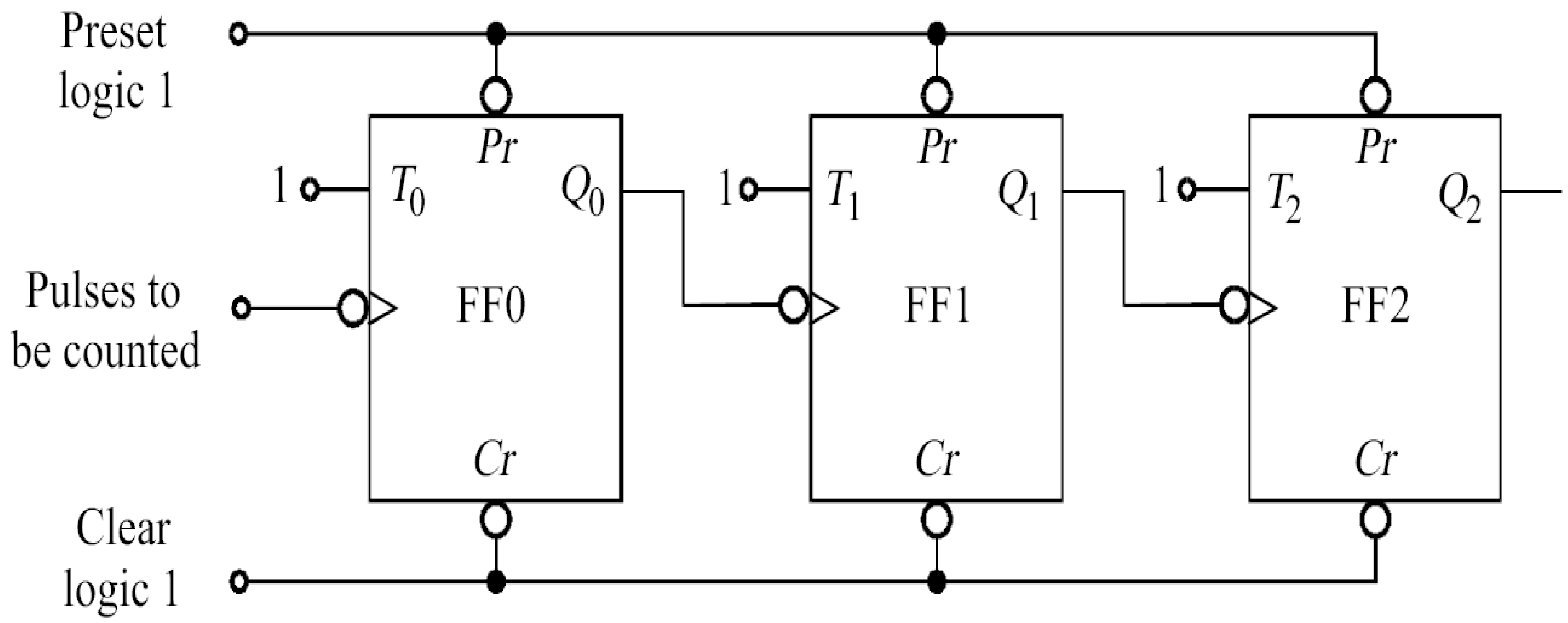


Fig. 8.10 ***A 3-bit Binary Counter***

SYNCHRONOUS COUNTERS

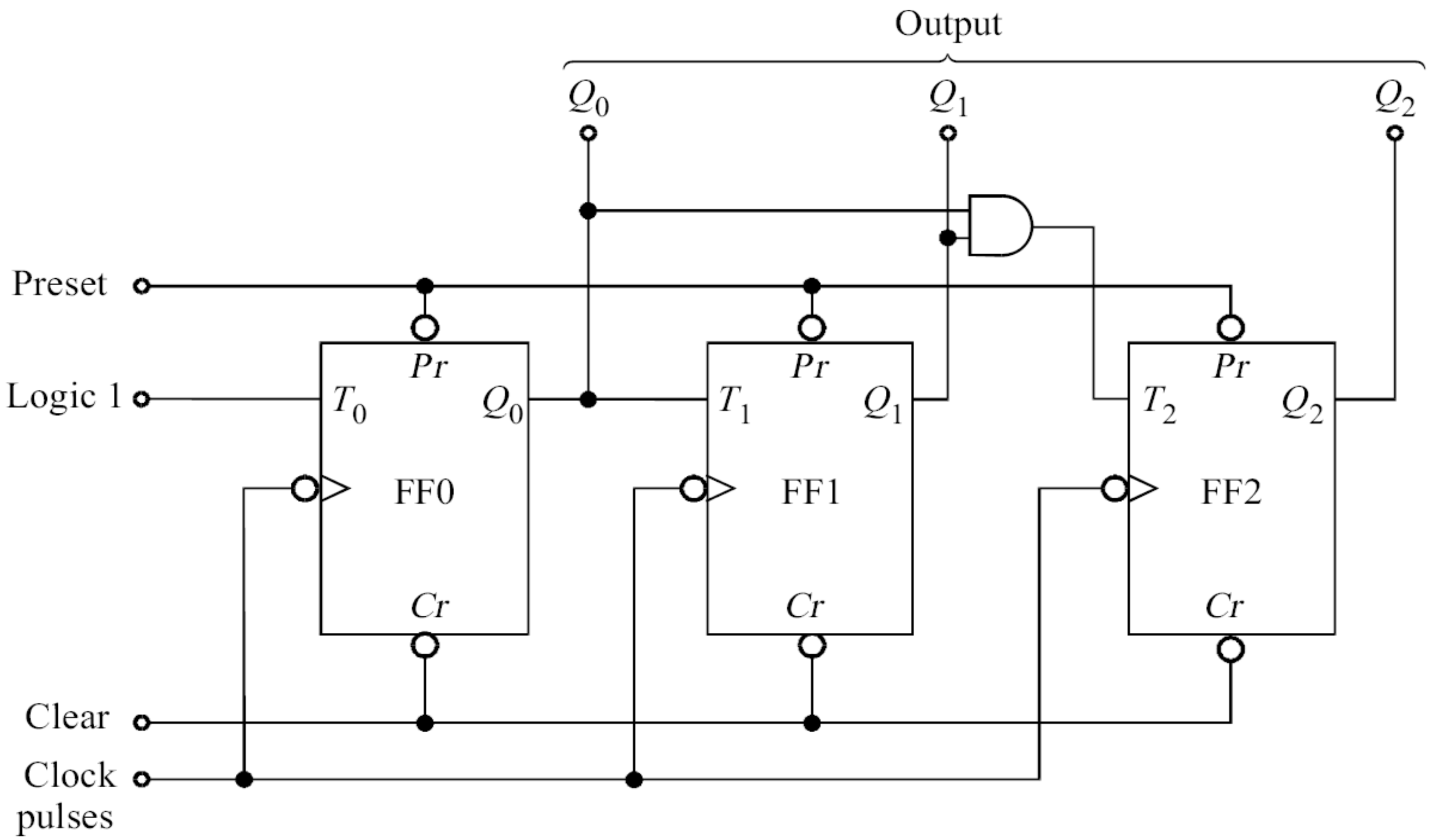


Fig. 8.22 ***A 3-bit Synchronous Counter***

Synchronous Counter Design

Synchronous counters for any given count sequence and modulus can be designed in the following way:

1. Find the number of FLIP-FLOPs required using Eq. (8.5).

2. Write the count sequence in the tabular form similar to Table 8.4.

3. Determine the FLIP-FLOP inputs which must be present for the desired next state from the present state using the excitation table of the FLIP-FLOPs (Table 7.6).

4. Prepare *K*-map for each FLIP-FLOP input in terms of FLIP-FLOP outputs as the input variables. Simplify the *K*-maps and obtain the minimized expressions.

5. Connect the circuit using FLIP-FLOPs and other gates corresponding to the minimized expressions.

The above design steps can be clearly understood from the following examples.

Example 8.9

Design a 3-bit synchronous counter using *J-K* FLIP–FLOPs.

Solution

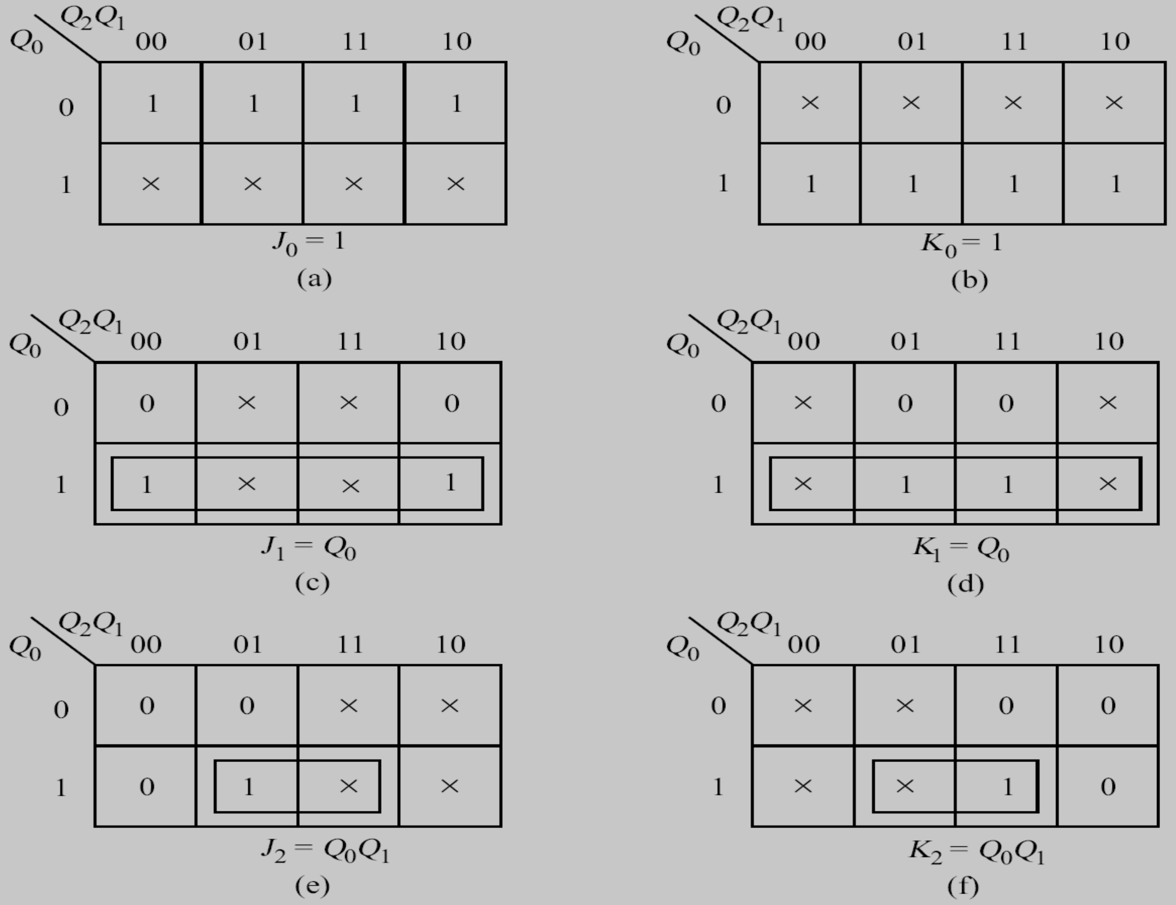
The number of FLIP–FLOPs required is 3. Let the FLIP–FLOPs be FF0, FF1, and FF2 and their inputs and outputs are given below:

FLIP–FLOP	Inputs	Output
FF0	J_0, K_0	Q_0
FF1	J_1, K_1	Q_1
FF2	J_2, K_2	Q_2

Table 8.10

Counter state			FLIP–FLOP inputs					
			FF0		FF1		FF2	
Q_2	Q_1	Q_0	J_0	K_0	J_1	K_1	J_2	K_2
0	0	0	1	×	0	×	0	×
0	0	1	×	1	1	×	0	×
0	1	0	1	×	×	0	0	×
0	1	1	×	1	×	1	1	×
1	0	0	1	×	0	×	×	0
1	0	1	×	1	1	×	×	0
1	1	0	1	×	×	0	×	0
1	1	1	×	1	×	1	×	1
0	0	0						

The count sequence and the required inputs of FLIP–FLOPs are given in Table 8.10. The inputs to the FLIP–FLOPs are determined in the following manner:



$Q_0 \backslash Q_2Q_1$	00	01	11	10
0	1	1	1	1
1	×	×	×	×

$J_0 = 1$

(a)

$Q_0 \backslash Q_2Q_1$	00	01	11	10
0	×	×	×	×
1	1	1	1	1

$K_0 = 1$

(b)

$Q_0 \backslash Q_2Q_1$	00	01	11	10
0	0	×	×	0
1	1	×	×	1

$J_1 = Q_0$

(c)

$Q_0 \backslash Q_2Q_1$	00	01	11	10
0	×	0	0	×
1	×	1	1	×

$K_1 = Q_0$

(d)

$Q_0 \backslash Q_2Q_1$	00	01	11	10
0	0	0	×	×
1	0	1	×	×

$J_2 = Q_0Q_1$

(e)

$Q_0 \backslash Q_2Q_1$	00	01	11	10
0	×	×	0	0
1	×	×	1	0

$K_2 = Q_0Q_1$

(f)

Fig. 8.23 ***K-Maps of Ex. 8.9***

Consider one column of the counter state at a time and start from the first row, for example, consider Q_0. Before the first pulse is applied, $Q_0 = 0$ and it is required to be 1 at the end of the first clock pulse. Therefore, to achieve this condition, the values of J_0 and K_0 are 1 and × respectively (from the excitation Table 7.6). These are entered in the table in the row corresponding to 0 pulse. When the second clock pulse is applied Q_0 is to change from 1 to 0, therefore, the required inputs are

$$J_0 = \times, K_0 = 1$$

In a similar manner inputs of each `FLIP-FLOP` are determined.

Now, we prepare the *K*-maps (Fig. 8.23) with Q_2, Q_1, and Q_0 as input variables and `FLIP-FLOP` inputs as output variables. We then minimize the *K*-maps and the resulting minimized expressions are:

$$\begin{aligned} J_0 &= 1, & K_0 &= 1 \\ J_1 &= Q_0, & K_1 &= Q_0 \\ J_2 &= Q_0 Q_1, & K_2 &= Q_0 Q_1 \end{aligned}$$

The resulting counter circuit is same as the circuit of Fig. 8.22.

Example 8.27

Design a sequence detector circuit to detect a serial input sequence of 1010. It should produce an output 1 when the input pattern has been detected.

Solution

Let the input string be 10101010, the desired output string can be determined, which is given below

Input X	1	0	1	0	1	0	1	0
Output Y	0	0	0	1	0	1	0	1

A state diagram can be constructed for the required input-output relationship.

Let us start with the initial state A. If the input is 0, the detection cycle must not start and therefore, the state remains the same and output is 0. When it is 1, it should go to the next state B with output as 0. In the present state B, if the input is 0 the next state will be C and output 0, while for an input 1, it is to be counted as the first correct bit in the string (since it is the second 1 bit from the start) and the state of the circuit should remain unchanged. In the present state C, if an input bit 0 occurs, the circuit should go back to the initial state (since it is second consecutive 0), while an input 1 causes state transition to next state D. When the circuit is in the state D, a 0 input will detect the correct sequence and will produce an output of 1. If the input is 1, it is a second consecutive 1 which must take the circuit to the state B. The complete state diagram is shown in Fig. 8.61. Its state table is constructed as given in Table 8.26.

Contd.

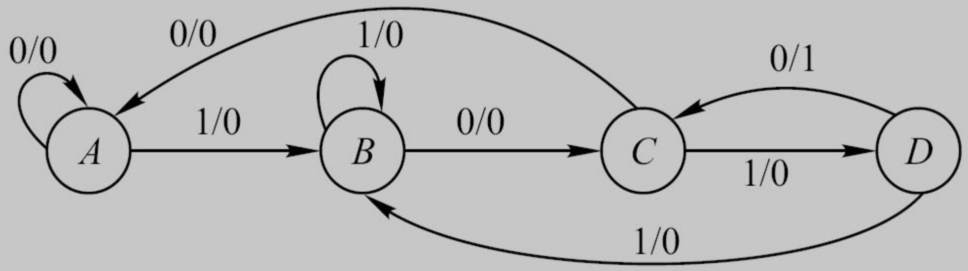


Fig. 8.61 ***State Diagram of Sequence Detector Circuit***

There are four states in this circuit and no equivalent states are present. Therefore, two *D*-type `FLIP-FLOPs` can be used for the implementation of this circuit.

Table 8.26 ***State Table of Sequence Detector***

Present state	**Next state, Output**	
	$\boldsymbol{X = 0}$	$\boldsymbol{X = 1}$
A	A, 0	B, 0
B	C, 0	B, 0
C	A, 0	D, 0
D	C, 1	B, 0

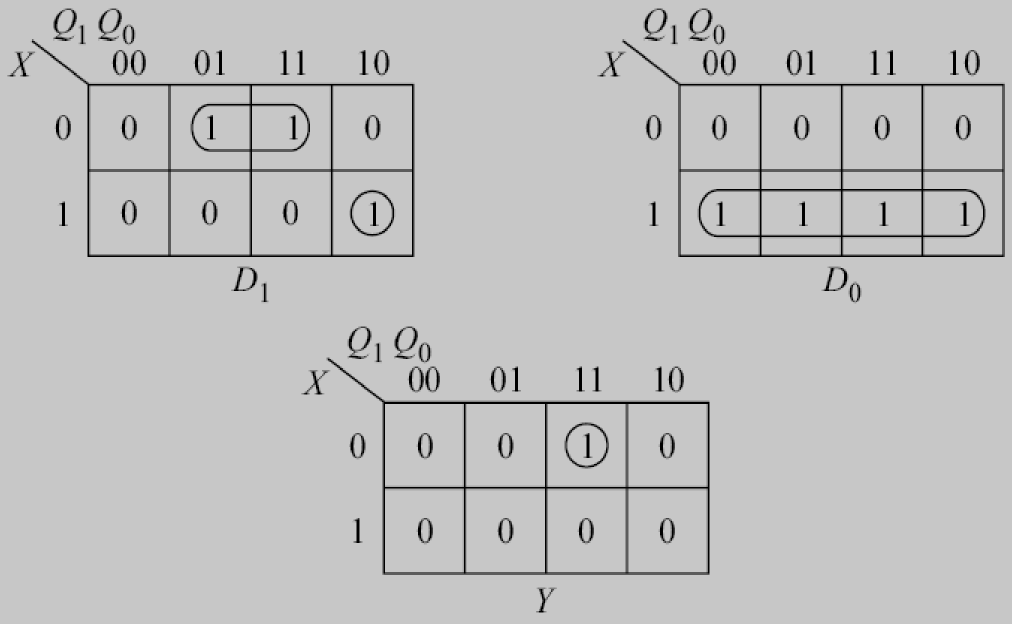
The two states B and D must be assigned adjacent codes, since the next state is same from these states for $X = 0$ and $X = 1$. Therefore, the following state assignment is made.

$$\begin{aligned} A &\to 00 \\ B &\to 01 \\ C &\to 10 \\ D &\to 11 \end{aligned}$$

Table 8.27 gives state transition and output table. From this K-maps are constructed for the FLIP-FLOP inputs D_1 and D_0; and output Y. These are given in Fig. 8.62.

Table 8.27 ***State Transition and Output Table***

Present state		Input	Next state		Output
Q_1	Q_0	X	Q_1^*	Q_0^*	Y
0	0	0	0	0	0
0	1	0	1	0	0
1	0	0	0	0	0
1	1	0	1	0	1
0	0	1	0	1	0
0	1	1	0	1	0
1	0	1	1	1	0
1	1	1	0	1	0



$X \backslash Q_1Q_0$	00	01	11	10
0	0	1	1	0
1	0	0	0	1

D_1

$X \backslash Q_1Q_0$	00	01	11	10
0	0	0	0	0
1	1	1	1	1

D_0

$X \backslash Q_1Q_0$	00	01	11	10
0	0	0	1	0
1	0	0	0	0

Y

Fig. 8.62 ***K-Maps of Ex. 8.27***

- - - - - - - -

The minimised logic expressions are:

$$D_1 = Q_0\overline{X} + Q_1\overline{Q_0}X$$

$$D_0 = X$$

$$Y = Q_1Q_0\overline{X}$$

Its circuit diagram is given in Fig. 8.63.

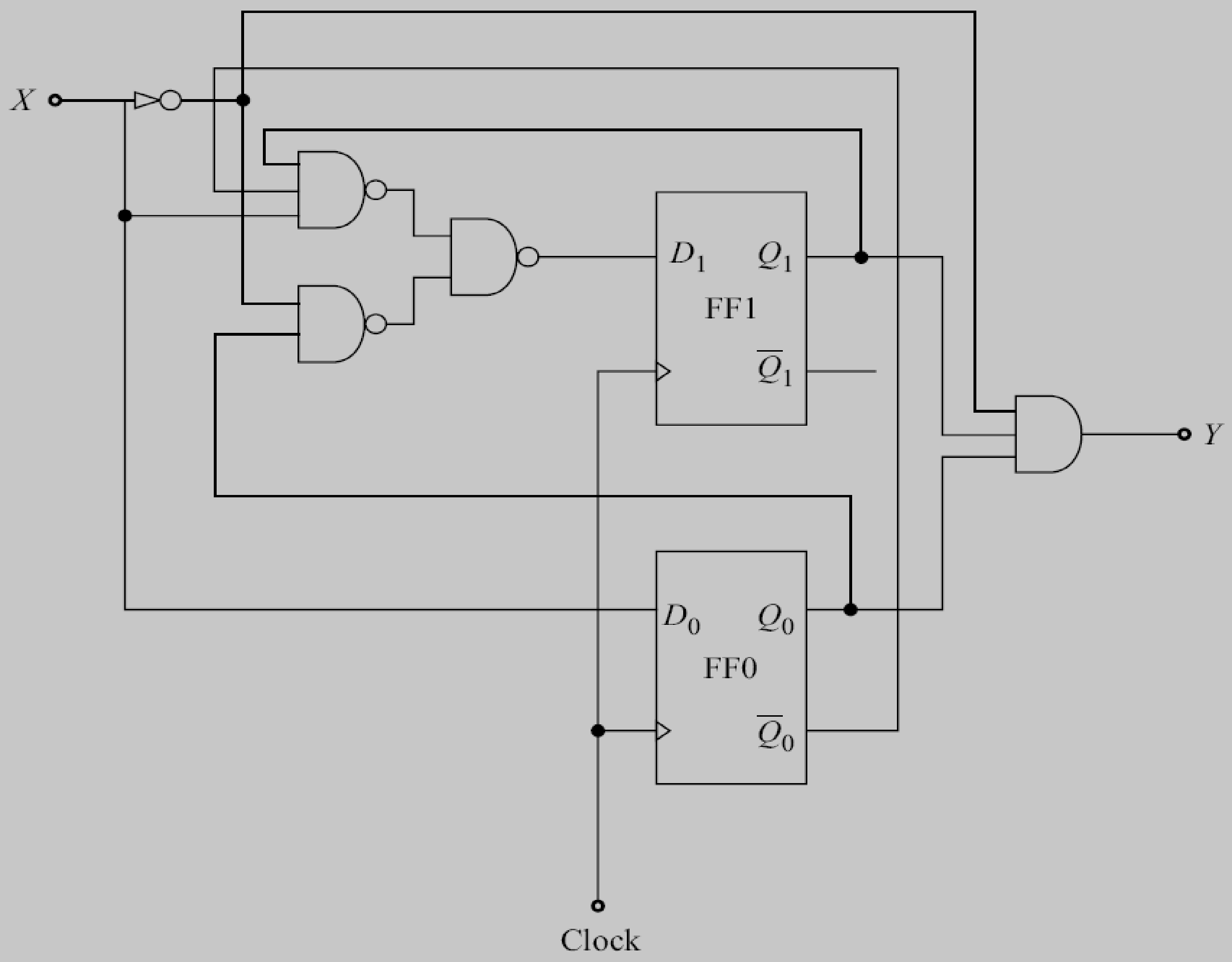


Fig. 8.63 ***Circuit of Sequences Detector***

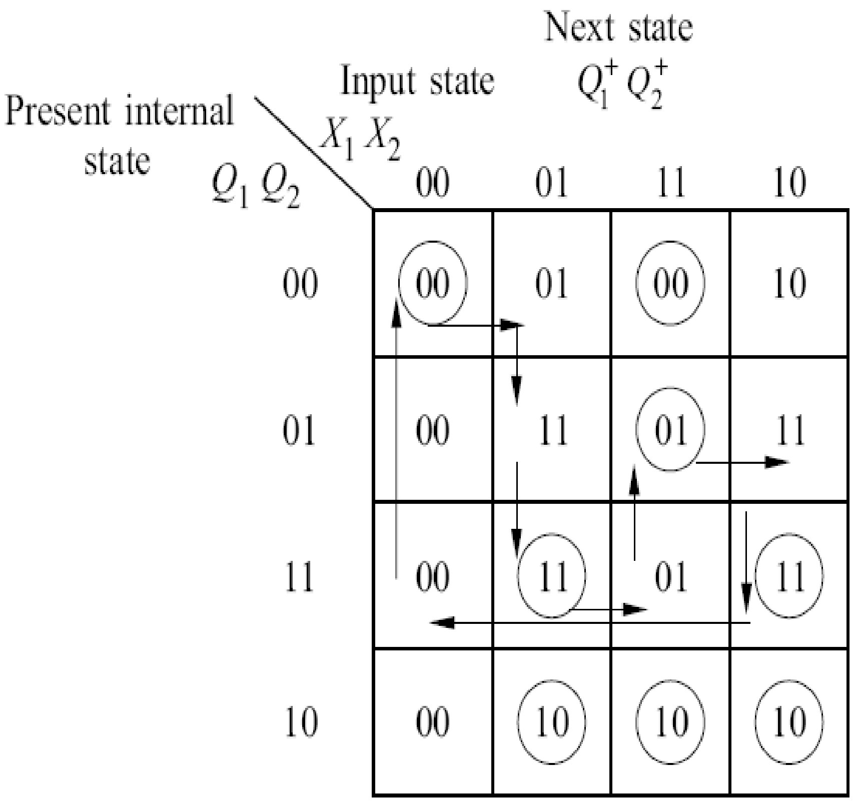
Transition Table

A state table can be represented in another form known as *transition table*.

Table 8.28 ***State Table***

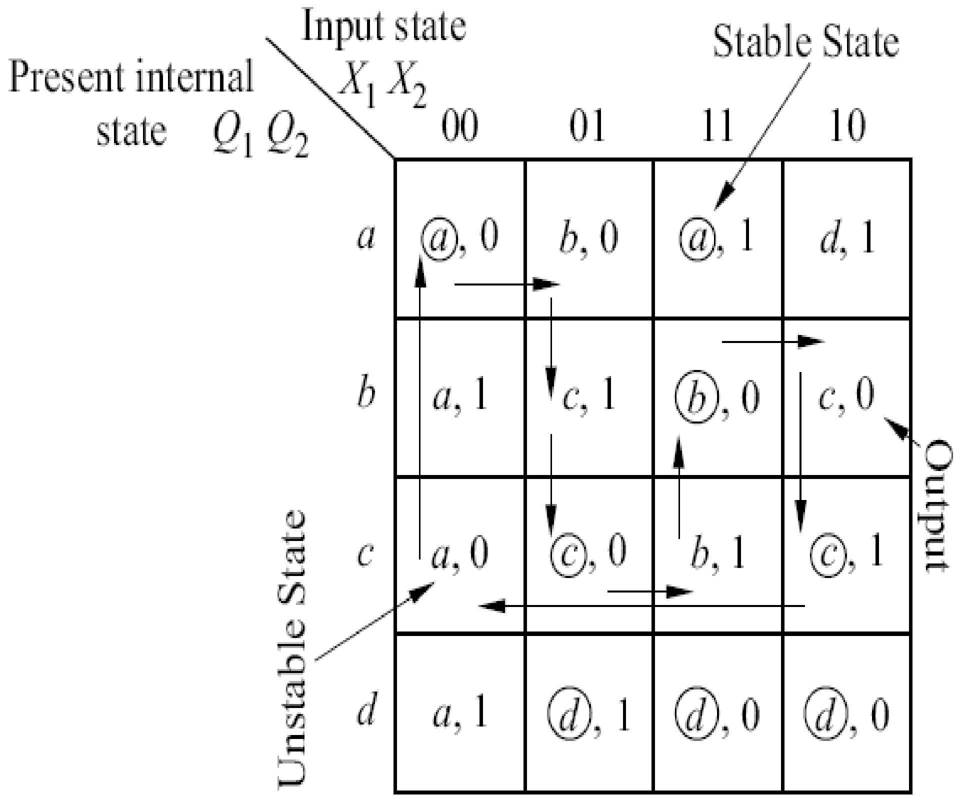
Present total state				Next total state				Stable total state	Output
Q_1	Q_2	X_1	X_2	Q_1^+	Q_2^+	X_1	X_2	Yes/No	Y
0	0	0	0	0	0	0	0	Yes	0
0	0	0	1	0	1	0	1	No	0
0	0	1	1	0	0	1	1	Yes	1
0	0	1	0	1	0	1	0	No	1
0	1	0	0	0	0	0	0	No	1
0	1	0	1	1	1	0	1	No	1
0	1	1	1	0	1	1	1	Yes	0
0	1	1	0	1	1	1	0	No	0
1	1	0	0	0	0	0	0	No	0
1	1	0	1	1	1	0	1	Yes	0
1	1	1	1	0	1	1	1	No	1
1	1	1	0	1	1	1	0	Yes	1
1	0	0	0	0	0	0	0	No	1
1	0	0	1	1	0	0	1	Yes	1
1	0	1	1	1	0	1	1	Yes	0
1	0	1	0	1	0	1	0	Yes	0

Flow table



$Q_1 Q_2$ \ $X_1 X_2$	00	01	11	10
00	$\textcircled{00}$	01	$\textcircled{00}$	10
01	00	11	$\textcircled{01}$	11
11	00	$\textcircled{11}$	01	$\textcircled{11}$
10	00	$\textcircled{10}$	$\textcircled{10}$	$\textcircled{10}$

Fig. 8.66 ***Transition Table for Table 8.28***



$Q_1 Q_2$ \ $X_1 X_2$	00	01	11	10
a	$\textcircled{a}$, 0	b, 0	$\textcircled{a}$, 1	d, 1
b	a, 1	c, 1	$\textcircled{b}$, 0	c, 0
c	a, 0	$\textcircled{c}$, 0	b, 1	$\textcircled{c}$, 1
d	a, 1	$\textcircled{d}$, 1	$\textcircled{d}$, 0	$\textcircled{d}$, 0

Fig. 8.67 ***Flow Table***

Circuits with Latches

For the circuit of Fig 8.68*a*, the next-state equation is

$$\begin{aligned} Q^+ &= \overline{\overline{S}\cdot\overline{Q}} = \overline{\overline{S}\cdot\overline{(Q\overline{R})}} \\ &= S + \overline{R}Q \end{aligned}$$

Similarly, for the circuit of Fig. 8.68*b*, the next-state equation is

$$\begin{aligned} Q^+ &= \overline{R+\overline{Q}} = \overline{R+\overline{(S+Q)}} \\ &= \overline{R}\cdot(S+Q) \\ &= S\overline{R} + \overline{R}Q \end{aligned}$$

Since, $S = R = 1$ is not allowed, which means $SR = 0$, therefore,

$$S\overline{R} = S\overline{R} + SR = S(\overline{R} + R) = S$$

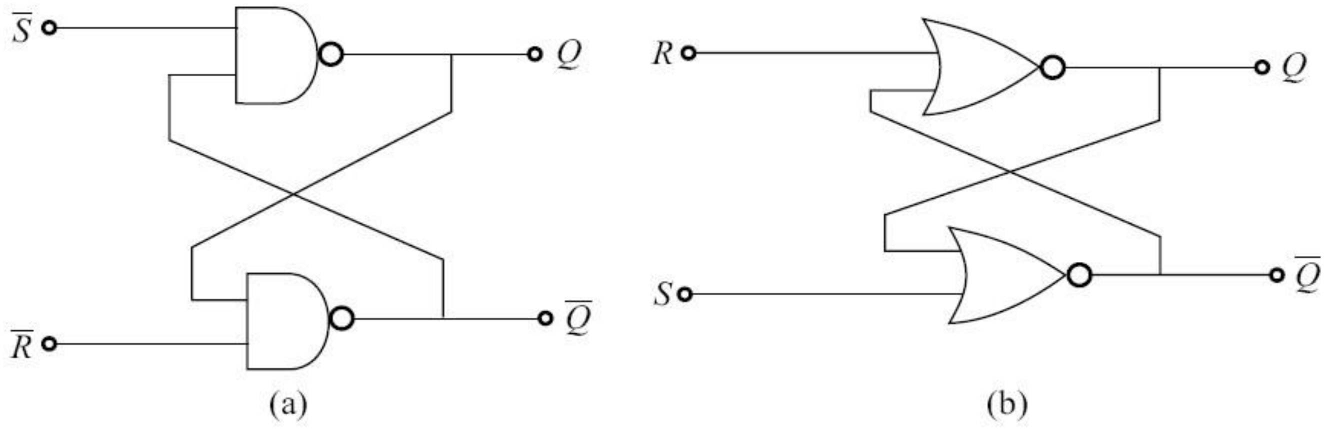


Fig. 8.68 ***(a)*** $\overline{S}$ **-** $\overline{R}$ ***Latch Using*** **NAND** ***Gates (b) S-R Latch Using*** **NOR** ***Gates***

which gives,

$$Q^{+} = S + \overline{R}Q$$

It is same as Eq. (8.9)
The transition table of *S*-*R* latch is shown in Fig. 8.69.

$Q \backslash SR$	00	01	11	10
0	$\textcircled{0}$	$\textcircled{0}$	$\textcircled{0}$	1
1	$\textcircled{1}$	0	0	$\textcircled{1}$

$$\begin{aligned} Q^{+} &= S\overline{R} + \overline{R}Q \\ &= S + \overline{R}Q \end{aligned}$$

Fig. 8.69 ***Transition Table of S-R Latch***

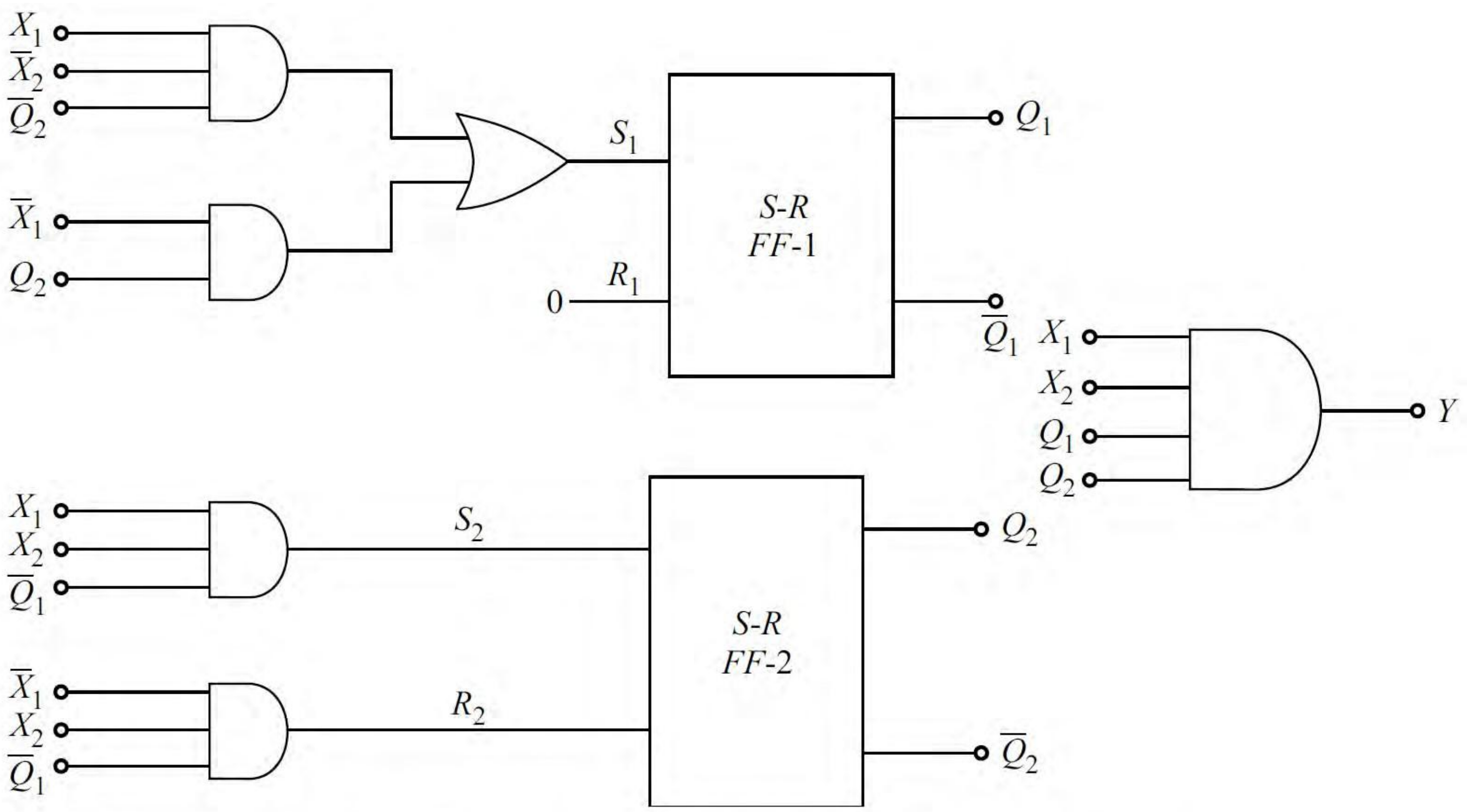


Fig. 8.70 ***Asynchronous Sequential Circuit with Latches***

$$Y = X_1\, X_2\, Q_1\, Q_2$$

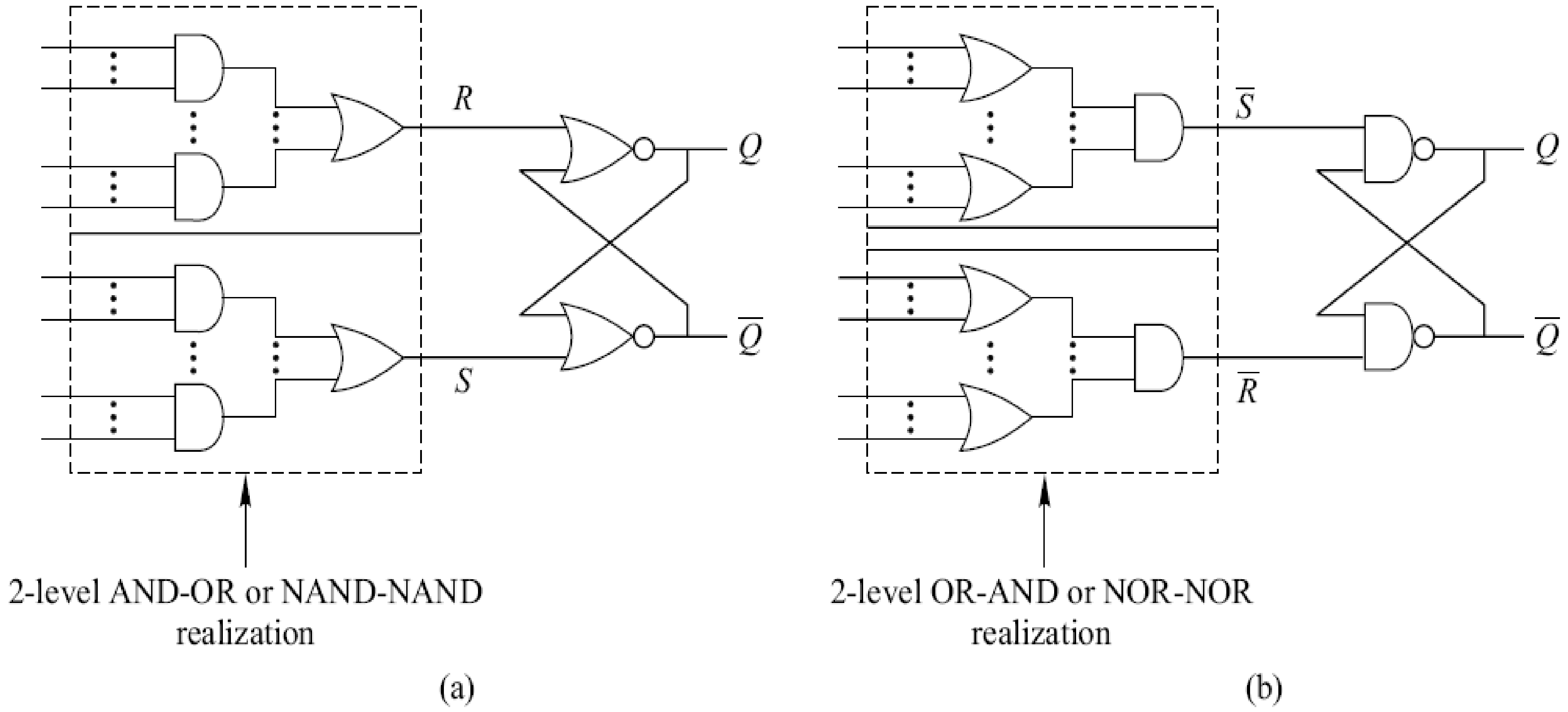
$Q_1\, Q_2$ \ $X_1\, X_2$	$Q_1^+\, Q_2^+$			
	00	01	11	10
00	$\textcircled{00}$	$\textcircled{00}$	10	10
01	11	11	$\textcircled{01}$	$\textcircled{01}$
11	$\textcircled{11}$	10	$\textcircled{11}$	$\textcircled{11}$
10	$\textcircled{10}$	$\textcircled{10}$	$\textcircled{10}$	$\textcircled{10}$

Fig. 8.71 ***Transition Table for the Circuit of Fig. 8.70***

$Q_1 Q_2$ \ $X_1 X_2$	$Q_1^+ Q_2^+$			
	00	01	11	10
a	ⓐ, 0	ⓐ, 0	b, 0	d, 0
b	c, 0	c, 0	ⓑ, 0	ⓑ, 0
c	ⓒ, 0	d, 0	ⓒ, 1	ⓒ, 0
d	ⓓ, 0	ⓓ, 0	ⓓ, 0	ⓓ, 0

Fig. 8.72 ***Flow Table for the Circuit of Fig. 8.70***

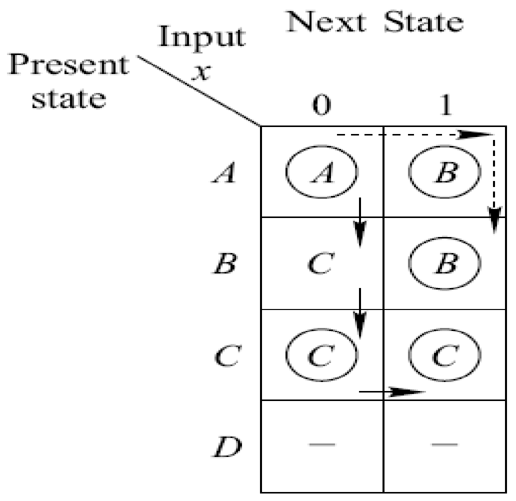
Essential-Hazards



R
Q
$\overline{Q}$
S
2-level AND-OR or NAND-NAND realization
(a)
$\overline{S}$
Q
$\overline{Q}$
$\overline{R}$
2-level OR-AND or NOR-NOR realization
(b)

Fig. 8.89 ***Hazard Free Asynchronous Circuits Using Latches***

The essential hazard can only be eliminated by introducing additional delay in the feedback path so that incorrect transition does not take place.
Essential hazard in an asynchronous sequential circuit can be detected by observing its flow table.



Present state \ Input x	0	1
A	$\textcircled{A}$	$\textcircled{B}$
B	C	$\textcircled{B}$
C	$\textcircled{C}$	$\textcircled{C}$
D	—	—

Fig. 8.90 ***Flow Table***

Example 8.39

Determine whether essential hazard exists in the flow tables of Fig. 8.91. Assume initial stable state $\textcircled{A}$.

Solution

In Fig. 8.91*a* when *x* changes from 0 to 1, the circuit may go to *B* and then to $\textcircled{B}$ or it may go to *A* (second row, first column) and eventually to $\textcircled{B}$. This shows that the circuit is free of essential hazard.

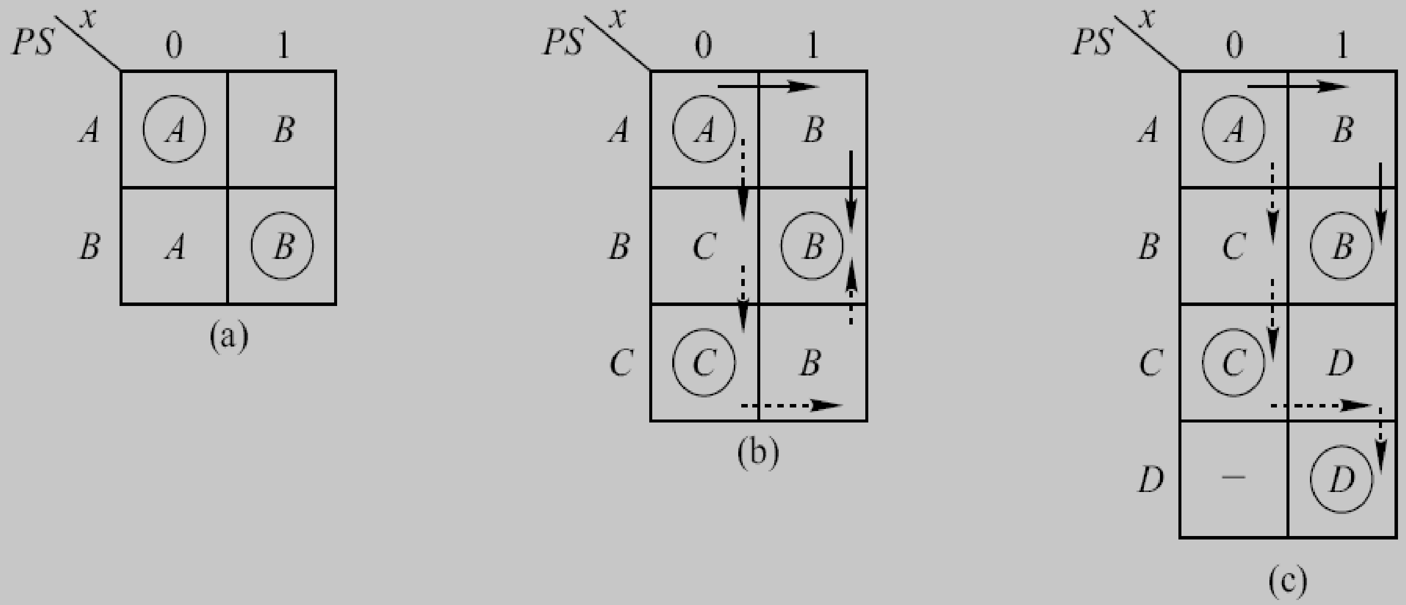


Fig. 8.91 ***Flow Tables of Ex. 8.39***

In Fig. 8.91*b*. The circuit goes to stable state $\textcircled{B}$ when x changes from 0 to 1. It may follow either the path shown by solid arrows or the path shown by the dotted arrows.

In Fig. 8.91*c*. The circuit goes to stable state $\textcircled{B}$ (see the solid arrows) or to the stable state D following the path shown by the dotted arrows. Therefore, unequal delays in the different feedback paths will cause transition to stable state $\textcircled{D}$ rather than $\textcircled{B}$ which shows the existence of essential-hazard.

Essential hazard can be eliminated by introducing additional delay in the feedback path with lesser delay.