Digital Electronics

(Second Year B. Tech program in Computer Engineering)

Shift Register

Register

It is the array of FLIP-FLOPs to store binary information.

Theses are used in various applications of digital system. Data in register can be entered (write) and retrieved (read) in serial form or parallel form.

Registers are classified according to the way in which data are read or write. There are 4 possible mode of operation:

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

Shift Register

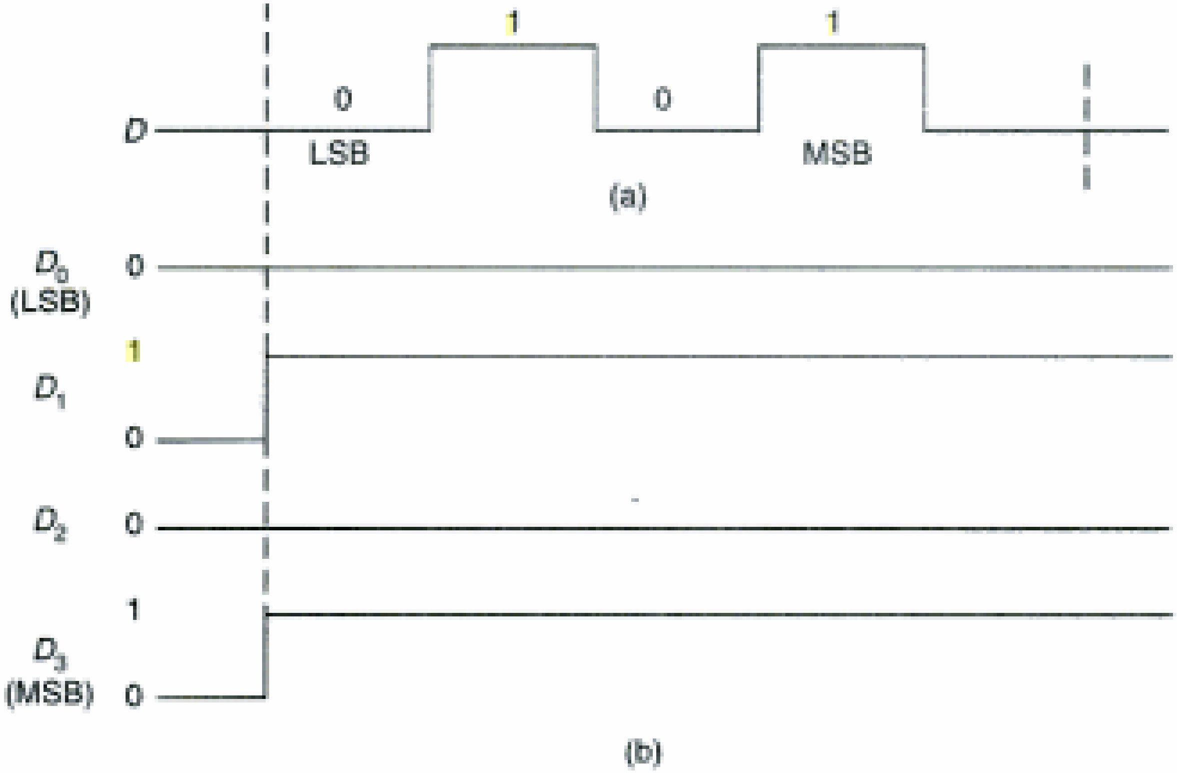


Fig. 8.1

Data representation in (*a*) Serial form (*b*) Parallel form.

Shift Register

Shift Register

Register in which data can be entered (write) and retrieved (read) in serial form are called as shift register.

1. Left Shift Register

Data can be shifted in left direction.

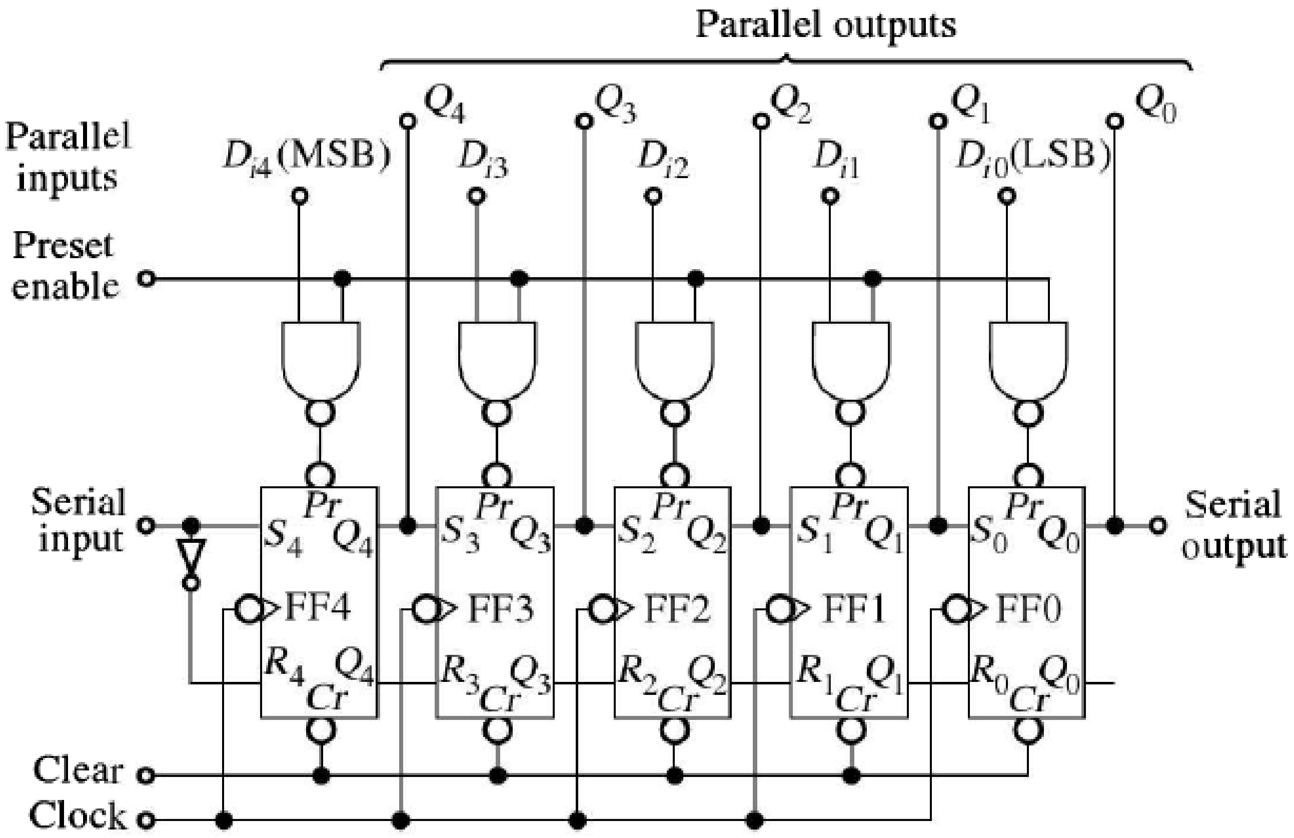
2. Right Shift Register

Data can be shifted in right direction.

3. Bi-directional Register

Data can be shifted from left to right direction as well as in reverse direction using mode control.

Shift Register



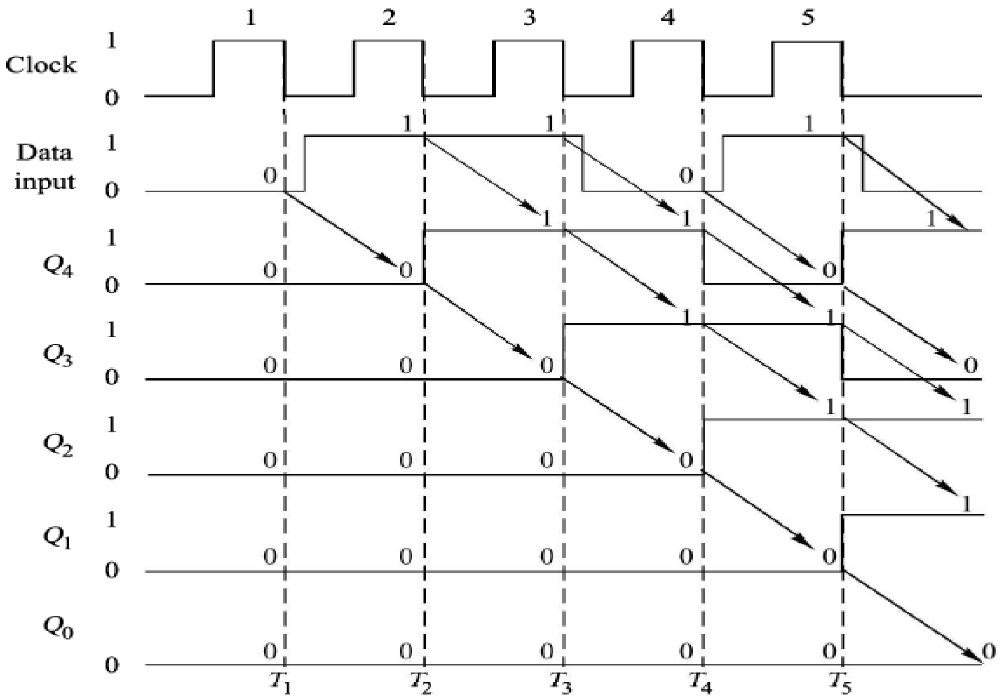
Assume 5 bit data 10110

5 bit Shift Register (7496)

Data word in serial form is applied at serial input. The preset enable signal is to be held at 0 so **Pr** for every FLIP-FLOP is 1.

Input & Output waveforms are shown in next fig.

Assume 5 bit data
 10110



Waveforms of Shift Register for Serial Input

Process of entering digital word starts with data i/p corresponding to LSB (0) at the serial i/p & 1st CLK pulse. At the falling edge of 1st CLK pulse T1, the o/p of FF4 will be 0 & o/p of all other F/F are o since their i/p are 0.

Next data i/p corresponding to next bit (1) is applied at the serial i/p & 2nd CLK pulse. At the falling edge of 2nd CLK pulse T2, the o/p of F/F will be Q4=1 & Q3,Q2,Q1,Q0=0

Similarly data i/p corresponding to each bit is applied at the serial i/p till the MSB & bit goes on shifting from left to right, At the end of 5th CLK pulse, the o/p of F/F will be Q4=1, Q3=0,Q2=1,Q1=1 and Q1=0 which is same as that of number to be stored

The data stored can be retrieved (read) in 2 ways:

Serial Out- Data in serial form is available at Q_0 when CLK pulse is applied. Here the number of CLK pulses required are same as that of number of bits.

In serial output, once data is retrieved the register is empty

Parallel Out-Data in serial form is available at Q_4 Q_3 Q_2 Q_1 Q_0 and CLK pulses is not required for reading.

In parallel output, data is retrieved any number of times until new data is stored in register

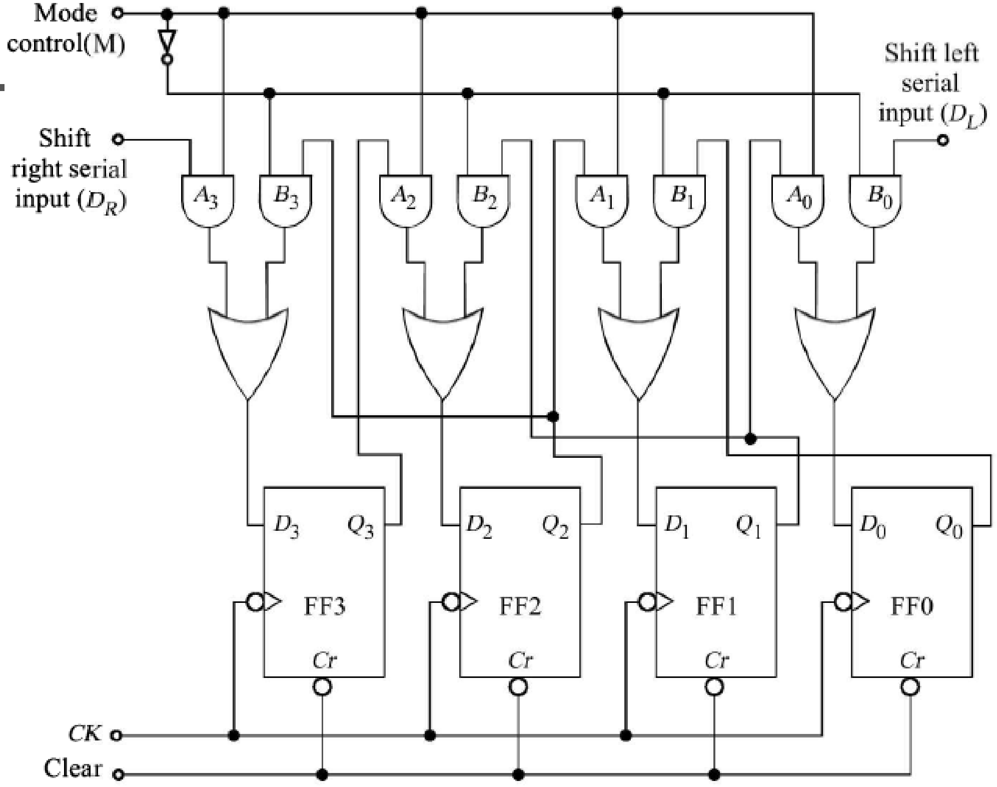
Parallel Input –

Data can be entered in parallel form making use of preset i/p. If the data lines are connected to the parallel i/p (D_{i4} D_{i3} D_{i2} D_{i2} D_{i0}) & 1 is applied at preset i/p. the data are written into the register. This is called as asynchronous loading.

Data is also loaded when CLK pulse is applied. This is called as synchronous loading.

Bi-directional Register–

Data can be shifted in left or right direction as per the requirement of application.



A 4-bit Bi-directional Shift Register

When M (mode control)=1, all the A AND gates are enable & data at D_R is shifted to right. On the other hand, When M (mode control)=0, all the B AND gates are enable & data at D_L is shifted to left.

Application of Shift Registers

1. Delay Line
2. Parallel-to-Serial Converter
3. Serial-to-parallel Converter
4. Ring Counter
5. Twisted Counter
6. Sequence Generator

Application of Shift Registers

1. Delay Line

A SISO shift register may be used to introduce time delay Δt in digital signal & is given by

$$\Delta t = N \times \frac{1}{fc}$$

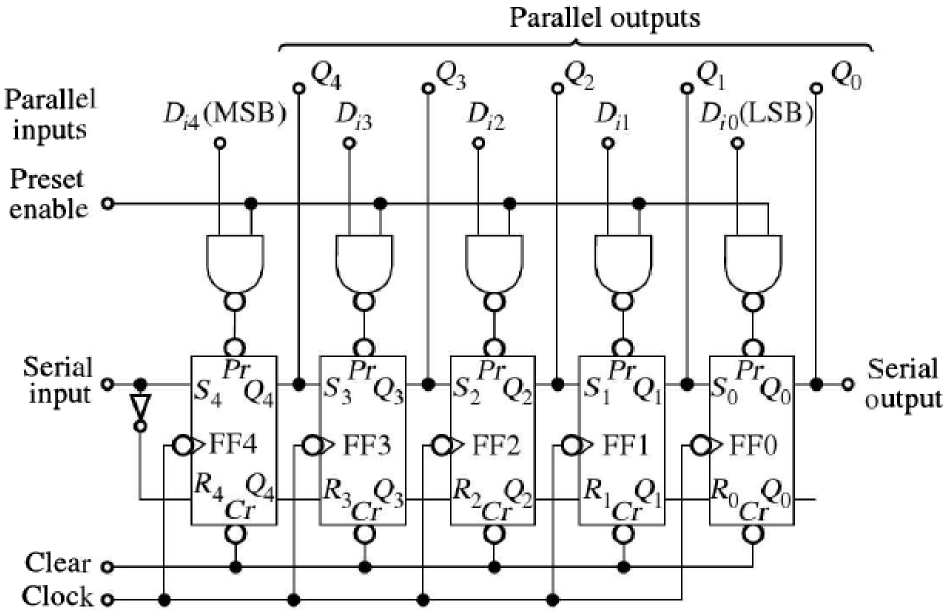
2. Serial-to-parallel Converter

Data in serial form can be converted into parallel form by using SIPO shift register

3. Parallel-to-Serial Converter

Data in parallel form can be converted into serial form by using PISO shift register

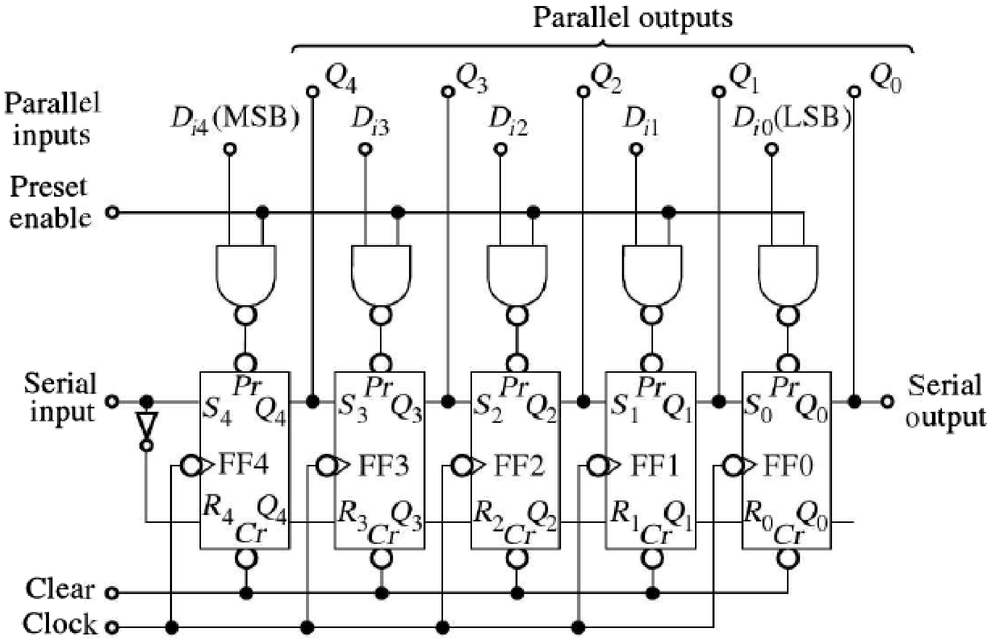
Application of Shift Registers


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
Serial output
Pr
S_4 Q_4 FF4 R_4 $\bar{Q}_4$ Cr
S_3 Q_3 FF3 R_3 $\bar{Q}_3$ Cr
S_2 Q_2 FF2 R_2 $\bar{Q}_2$ Cr
S_1 Q_1 FF1 R_1 $\bar{Q}_1$ Cr
S_0 Q_0 FF0 R_0 $\bar{Q}_0$ Cr
Clear
Clock

4 . Ring Counter

- If the serial output Q_0 of the shift register is connected back to the serial input, then the injected pulse will keep circulating and the circuit is then called as Ring counter.
- Ring counter sequence is used for control-state counters, stepper motors
- This circuit is used for counting number of CLK pulses
- No decoding circuitry is required here
- Since number of stages in an N ring counter is N, so this circuit is also called as modulo N or divide-by N counter or N:1 scaler

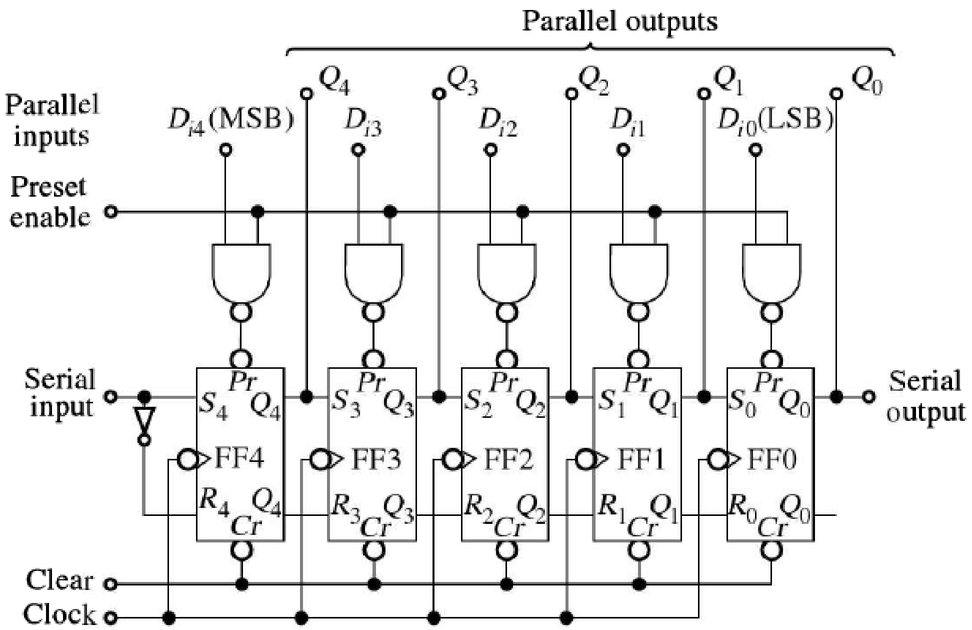
Applications of Shift Registers

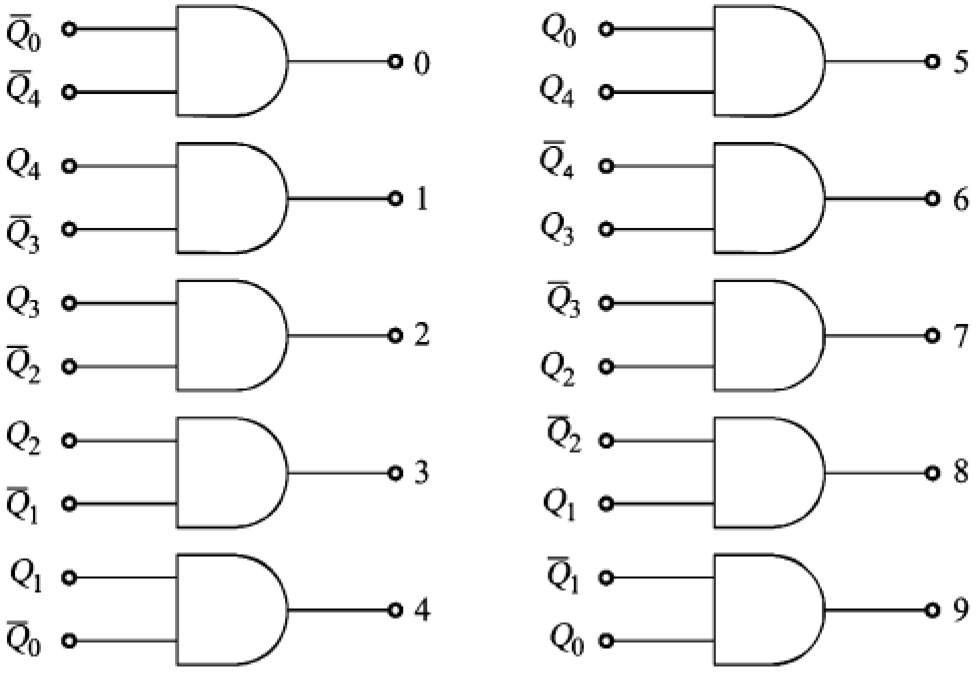


5. Twisted Counter

If the serial output Q_0 of the shift register is connected back to the serial input, then resulting circuit is then called as Twisted counter / Johnson or Moebius counter

- Moebius counter sequence is used for control-state counters, stepper motors
- Decoding circuitry is required here
- Since number of stages in an N ring counter is 2N, so this circuit is also called as modulo 2N or divide-by 2N counter or N:2 scaler





The Decoding Logic for a 5-Stage Twisted-Ring Counter

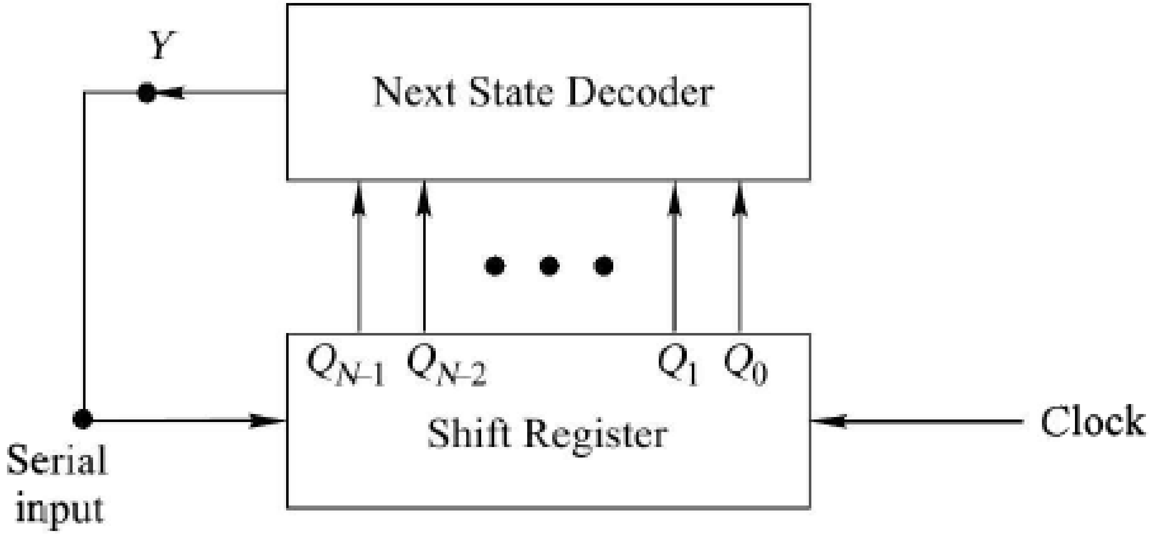
Application of Shift Registers

6. Sequence Generator

A Circuit which generates given sequence of bit, in synchronism with a CLK, is referred to as a sequence generator. Such generators are used as

- Counters
- Random bit generators
- Prescribed period & sequence generators
- Code generators

Application of Shift Registers



Basic Structure of Sequence Generator

Here output Y of next state decoder is a function of Q_{N-1}, $Q_{N-2\ldots}$ $Q_{1,}$ Q_0

Generate Sequence of 10110

No. of Flip Flops required can be identified by

$$L <= 2^n - 1$$

Where, L = Length of Sequence

n = No. of Flip Flops

$5 <= 2^n - 1$

$5 + 1 <= 2^n$

$6 <= 2^n$

$n = 3$

A	B	C
1	0	1
0	1	0
1	0	1
1	1	0
0	1	1

Generate Sequence of 10110

No. of Flip Flops required can be identified by

$$L <= 2^n - 1$$

Where, L = Length of Sequence

n = No. of Flip Flops

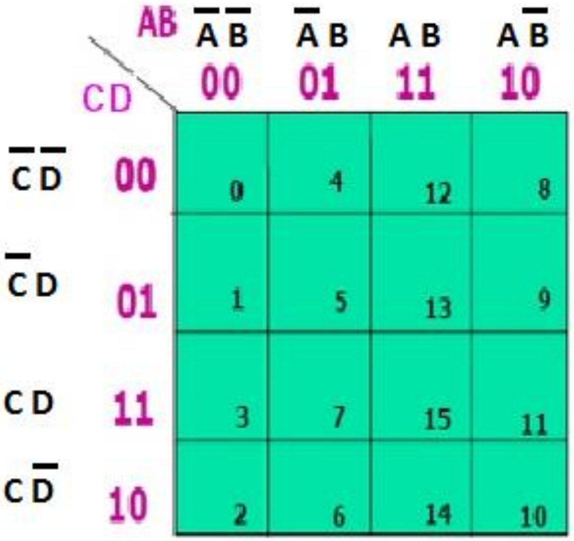
$5 <= 2^n - 1$

$5 + 1 <= 2^n$

$6 <= 2^n$

$n = 3$

A	B	C	D	F
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
0	1	1	0	1



CD \ AB		$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}\bar{D}$	00	0	4	12	8
$\bar{C}D$	01	1	5	13	9
CD	11	3	7	15	11
$C\bar{D}$	10	2	6	14	10

Generate Sequence of 10110

No. of Flip Flops required can be identified by

$$L <= 2^n - 1$$

Where, L = Length of Sequence

n = No. of Flip Flops

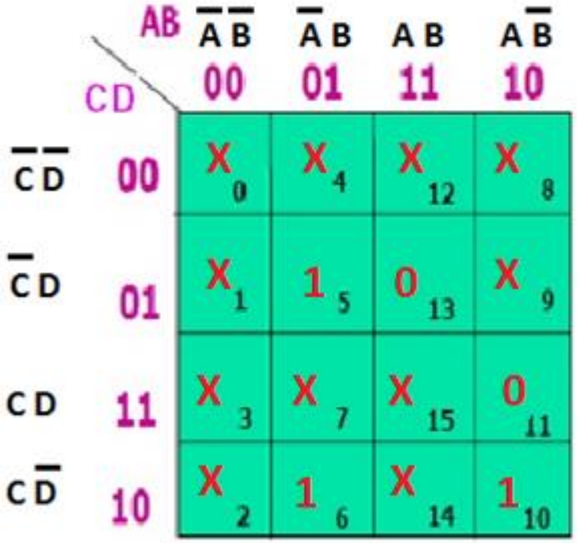
$5 <= 2^n - 1$

$5 + 1 <= 2^n$

$6 <= 2^n$

$n = 3$

A	B	C	D	F
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
0	1	1	0	1



CD \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}\bar{D}$ 00	X_{0}	X_{4}	X_{12}	X_{8}
$\bar{C}D$ 01	X_{1}	1_{5}	0_{13}	X_{9}
CD 11	X_{3}	X_{7}	X_{15}	0_{11}
$C\bar{D}$ 10	X_{2}	1_{6}	X_{14}	1_{10}

Generate Sequence of 10110

No. of Flip Flops required can be identified by

$$L <= 2^n - 1$$

Where, L = Length of Sequence

n = No. of Flip Flops

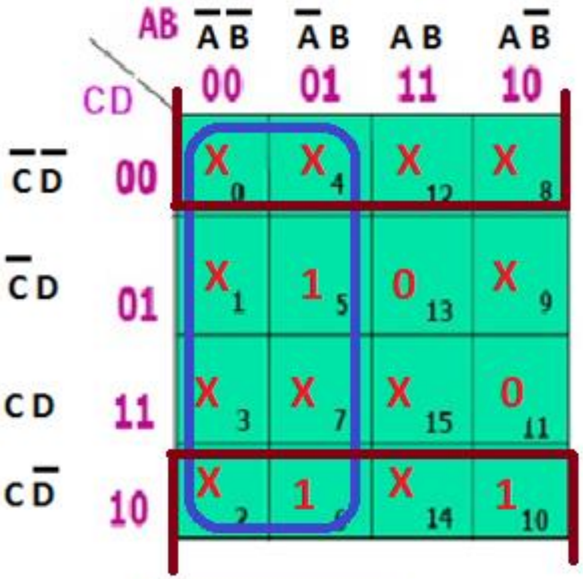
$5 <= 2^n - 1$

$5 + 1 <= 2^n$

$6 <= 2^n$

$n = 3$

A	B	C	D	F
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
0	1	1	0	1



CD \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}\bar{D}$ 00	X $_0$	X $_4$	X $_{12}$	X $_8$
$\bar{C}D$ 01	X $_1$	1 $_5$	0 $_{13}$	X $_9$
CD 11	X $_3$	X $_7$	X $_{15}$	0 $_{11}$
$C\bar{D}$ 10	X $_2$	1 $_6$	X $_{14}$	1 $_{10}$

$$\boxed{F = \bar{A} + \bar{D}}$$

Generate Sequence of 10110

No. of Flip Flops required can be identified by

$$L <= 2^n - 1$$

Where, L = Length of Sequence

n = No. of Flip Flops

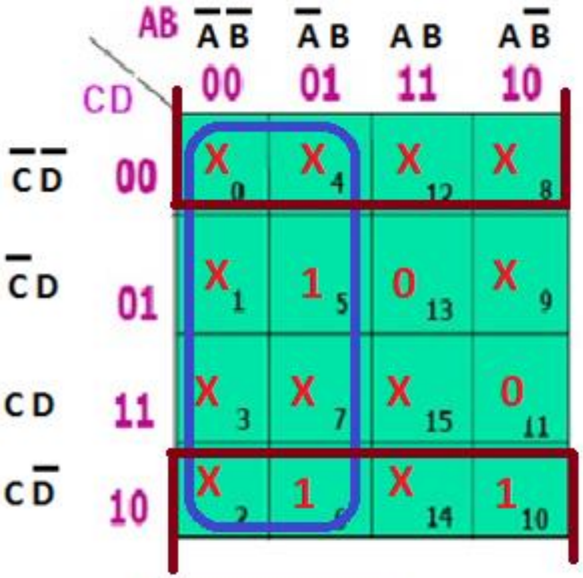
$5 <= 2^n - 1$

$5 + 1 <= 2^n$

$6 <= 2$

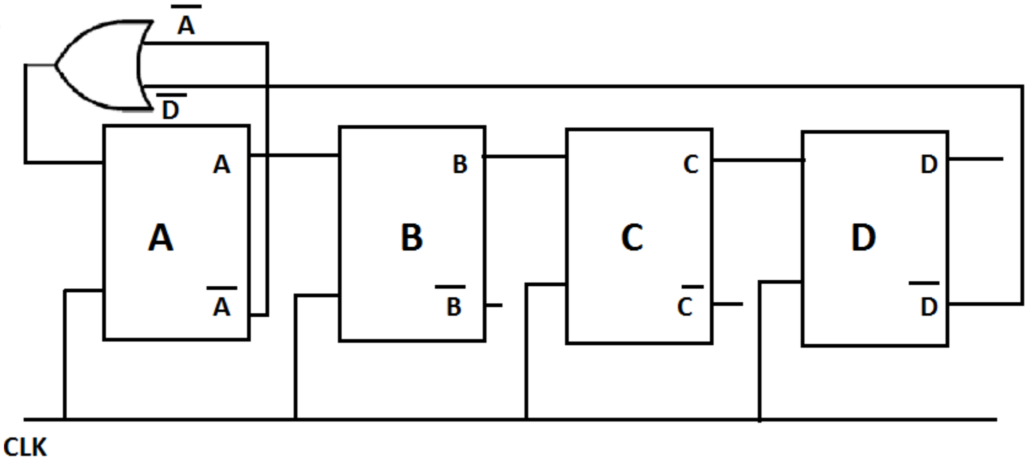
$n = 3$

A	B	C	D	F
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
0	1	1	0	1

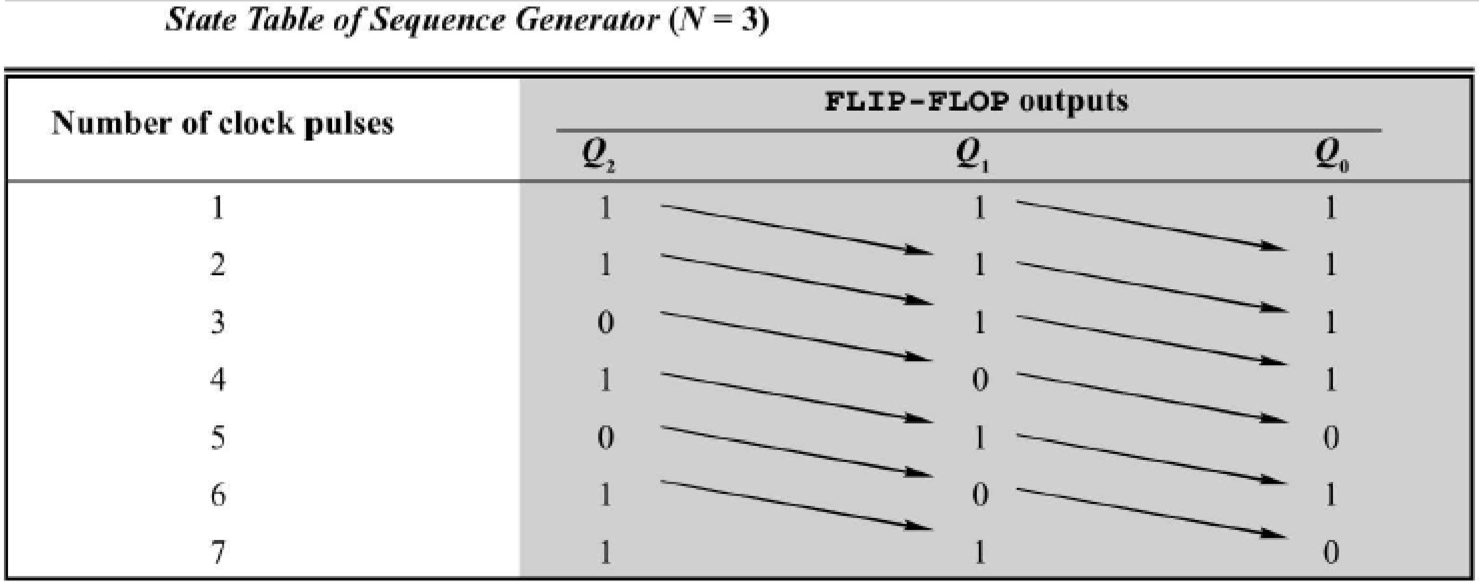


CD \ AB		$\overline{A}\,\overline{B}$ 00	$\overline{A}B$ 01	AB 11	$A\overline{B}$ 10
$\overline{C}\,\overline{D}$	00	X 0	X 4	X 12	X 8
$\overline{C}D$	01	X 1	1 5	0 13	X 9
CD	11	X 3	X 7	X 15	0 11
$C\overline{D}$	10	X 2	1 6	X 14	1 10

$$\boxed{F = \overline{A} + \overline{D}}$$



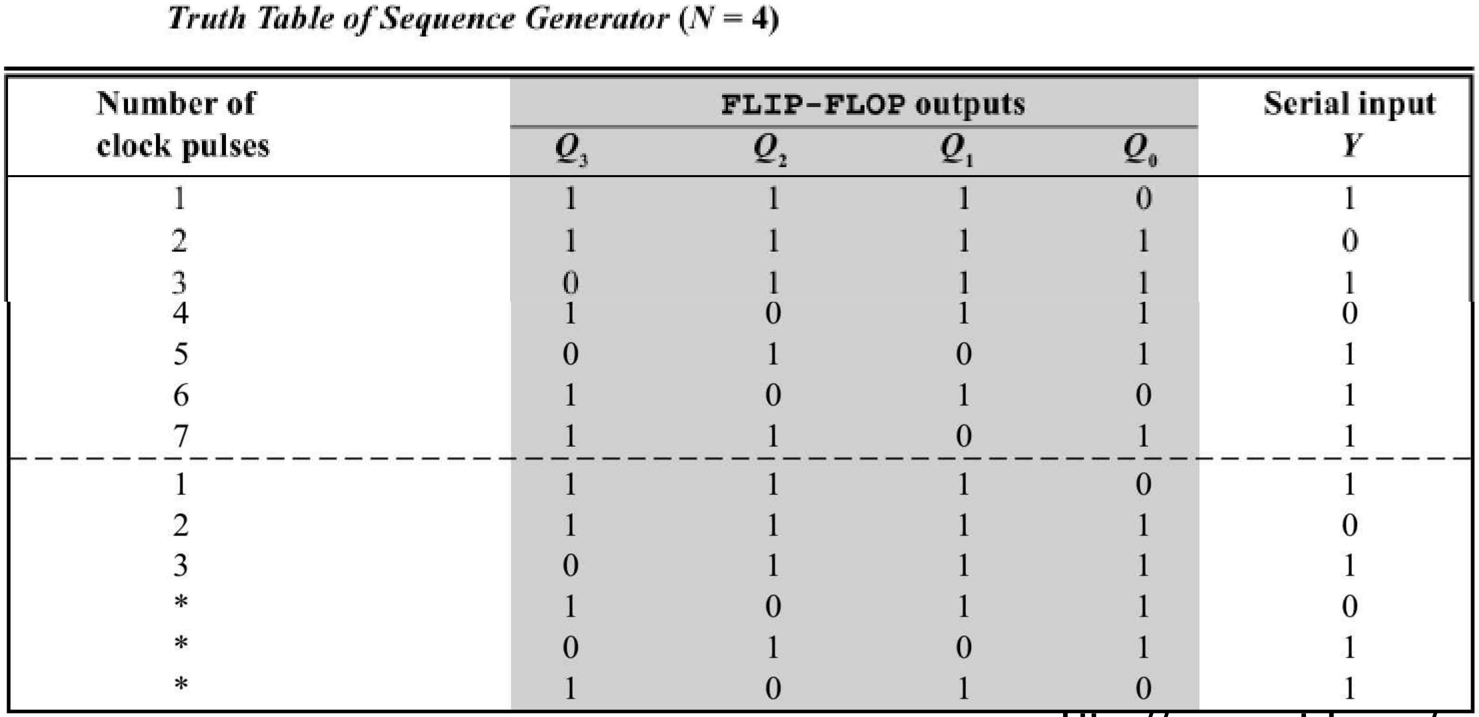
Design a sequence generator to generate the sequence … 1101011…



State Table of Sequence Generator (*N* = 3)

Number of clock pulses	FLIP-FLOP outputs		
	Q_2	Q_1	Q_0
1	1	1	1
2	1	1	1
3	0	1	1
4	1	0	1
5	0	1	0
6	1	0	1
7	1	1	0

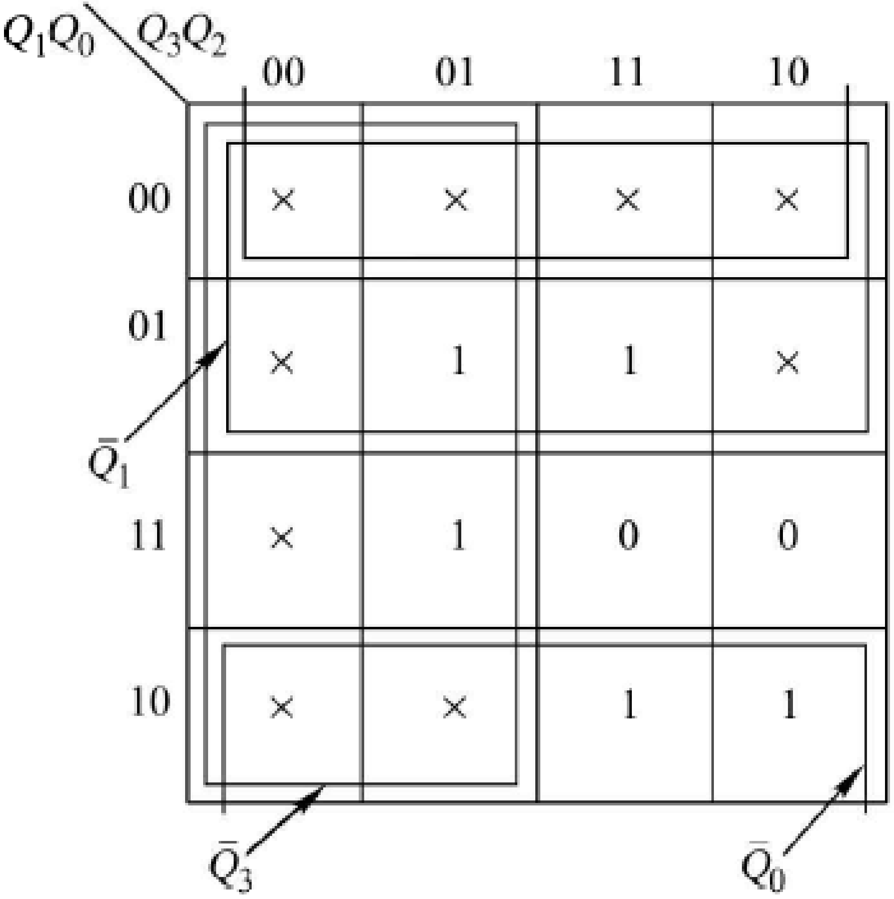
1101011



Truth Table of Sequence Generator (*N* = 4)

Number of clock pulses	FLIP-FLOP outputs				Serial input
	Q_3	Q_2	Q_1	Q_0	Y
1	1	1	1	0	1
2	1	1	1	1	0
3	0	1	1	1	1
4	1	0	1	1	0
5	0	1	0	1	1
6	1	0	1	0	1
7	1	1	0	1	1
1	1	1	1	0	1
2	1	1	1	1	0
3	0	1	1	1	1
*	1	0	1	1	0
*	0	1	0	1	1
*	1	0	1	0	1

https://www.youtube.com/watch?v=hWvqg6mmooQ



Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	×	×	×	×
01	×	1	1	×
11	×	1	0	0
10	×	×	1	1

$$Y = \overline{Q}_3 + \overline{Q}_1 + \overline{Q}_0$$

Counters

The difference between **asynchronous and synchronous counters**.

In Asynchronous counter, all the flip-flops are not clocked simultaneously whereas in Synchronous counter, all the flip-flops are clocked simultaneously.

Asynchronous counters allow outputs of some flip-flop to be used as a source of clock for other flip-flops whereas Synchronous counters apply the same clock to all flip-flops.

With a synchronous circuit, all the bits in the count changes synchronously with the assertion of the clock. With an asynchronous circuit, all the bits in the count do not change at the same time.

An example of an asynchronous counter is a ripple counter.
An example of an synchronous counter is a ring counter & twisted ring counter

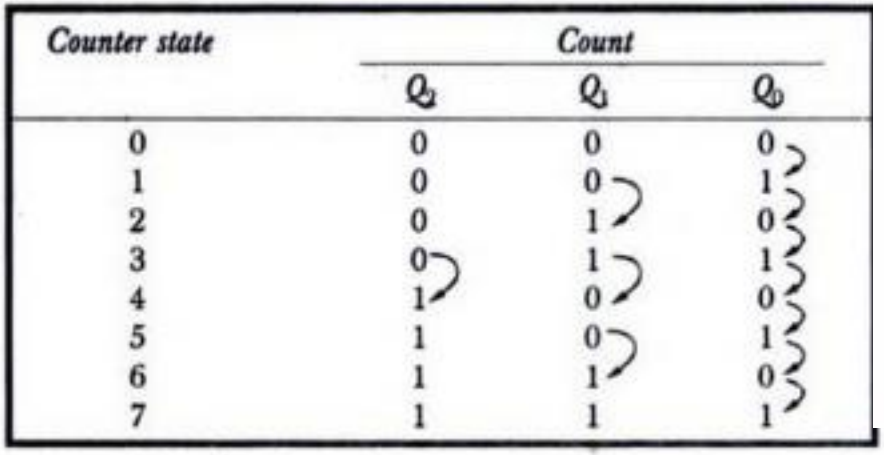
An asynchronous counter is one in which the flip-flop within the counter do not change states at exactly the same time because they do not have a common clock pulse.

The main characteristic of an asynchronous counter is each flip-flop derives its own clock from other flip-flops and is therefore independent of the input clock. Consequently, the output of each flip-flop may change at different time, hence the term asynchronous. From the asynchronous counter diagram above, we observed that the output of the first flip-flop becomes the clock input for the second flip-flop, and the output of the second flip-flop becomes the clock input for the third flip-flop etc.

Asynchronous Counters

3 Bit Binary Counter

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

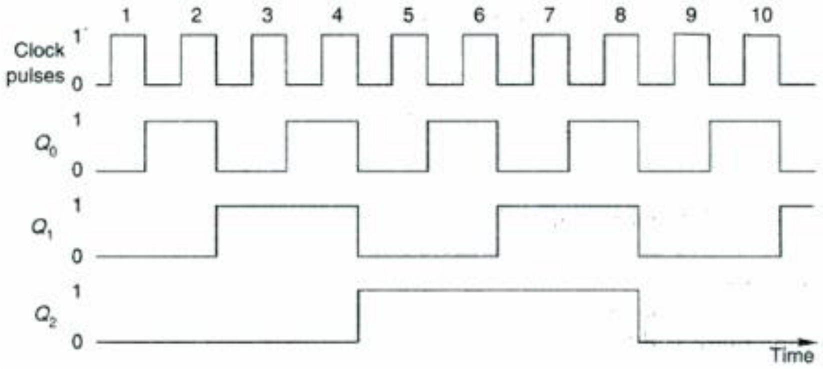
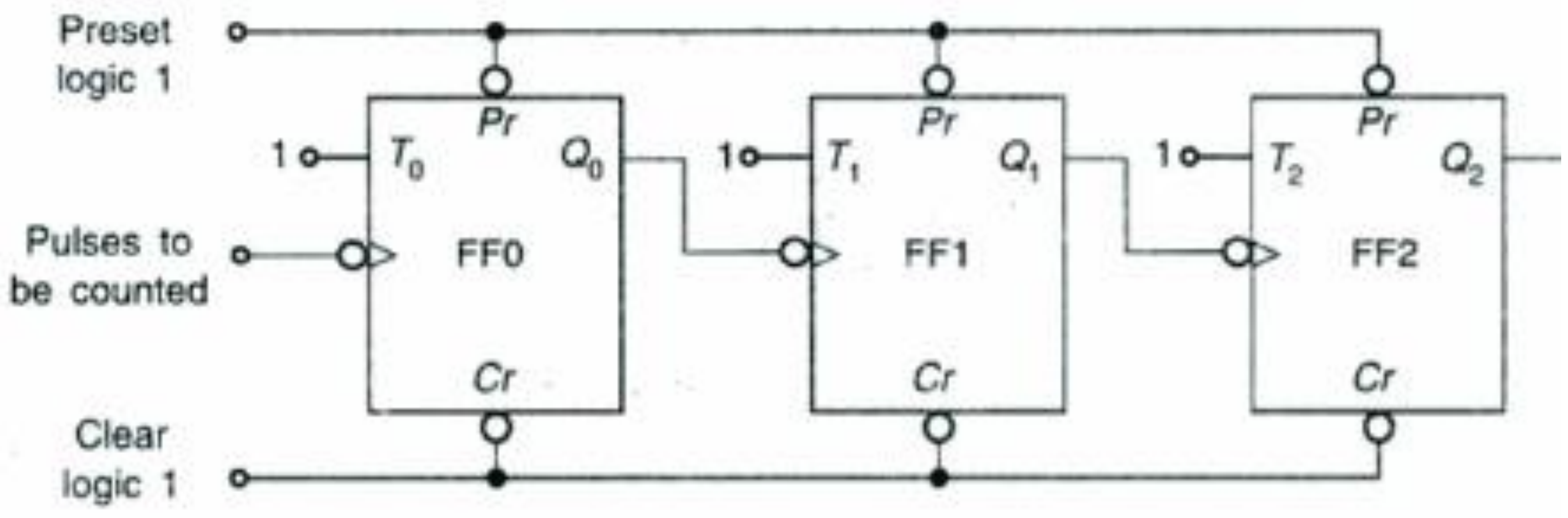


Fig. 8.11

Output waveforms of counter of Fig. 8.10.



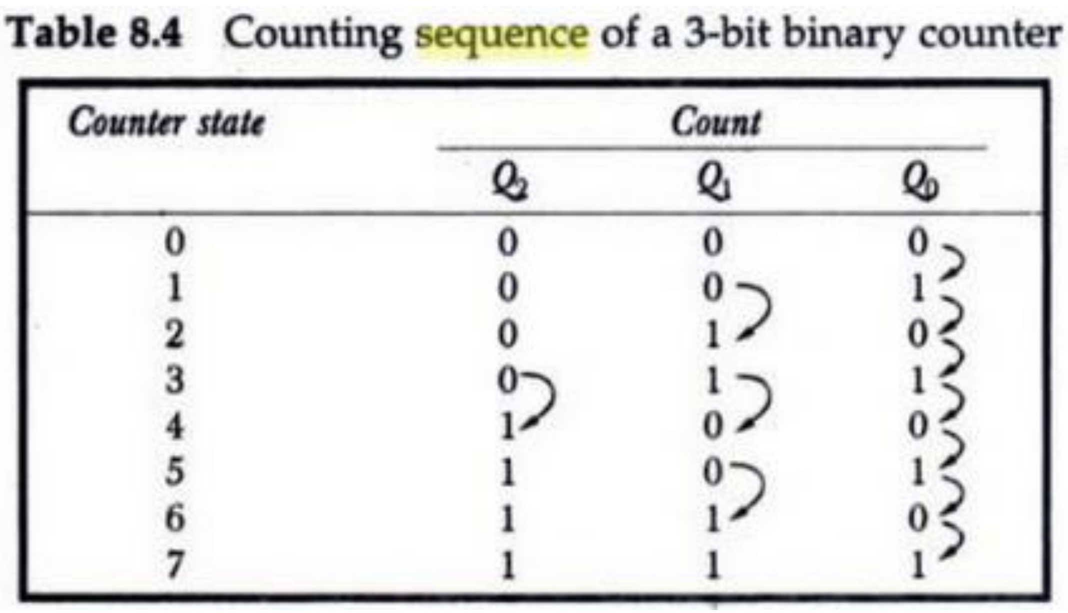
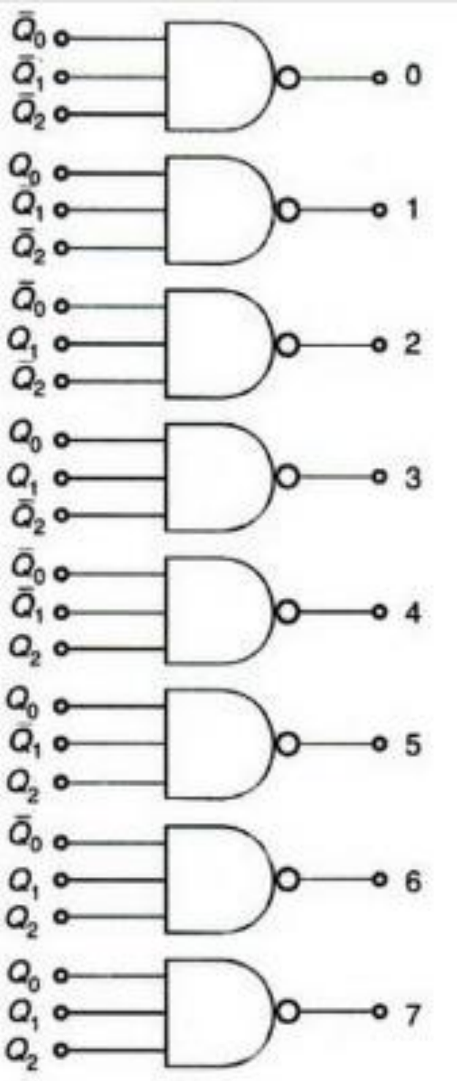


Table 8.4 Counting sequence of a 3-bit binary counter

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

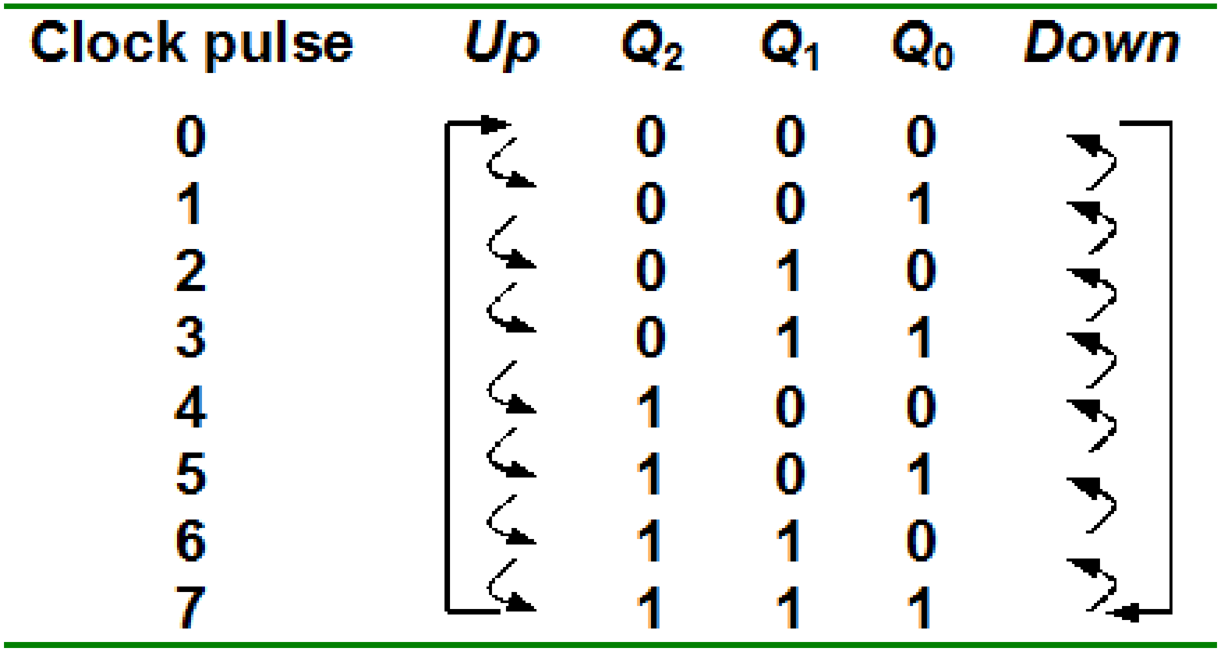


Counting In above table, output Q_0 of Least significant flip-flop changes for every CLK pulse by using T flip flop with T_0=1.

The output Q_1 makes a transition from 0 to 1 or 1 to 0 whenever Q_0 changes from 1 to 0.

Similarly The output Q_2 makes a transition from 0 to 1 or 1 to 0 whenever Q_1 changes from 1 to 0.

Up / Down Counter



Clock pulse	*Up*	Q_2	Q_1	Q_0	*Down*
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	

Counting In above table, output Q_0 of Least significant flip-flop changes for every CLK pulse by using T flip flop with T_0=1.

The output Q_1 makes a transition from 0 to 1 or 1 to 0 whenever Q_0 changes from 1 to 0.

Similarly The output Q_2 makes a transition from 0 to 1 or 1 to 0 whenever Q_1 changes from 1 to 0.

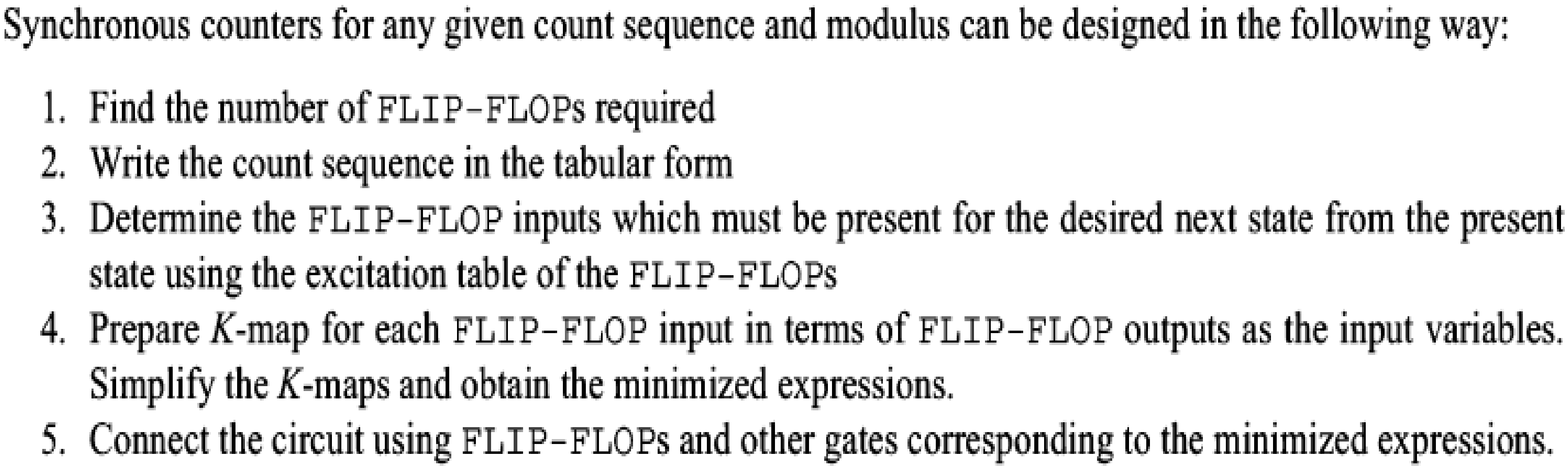
Synchronous Counter

Ripple/Asynchronous counters have the advantages of simplicity but their speed is low because of ripple action.

The maximum time required when output changes from 111….111 to 0000…..000 & this limits the frequency of operation of ripple counters.

The speed of operation improves significantly if all the flip flops are clocked simultaneously. The resulting circuit is known as synchronous counter.

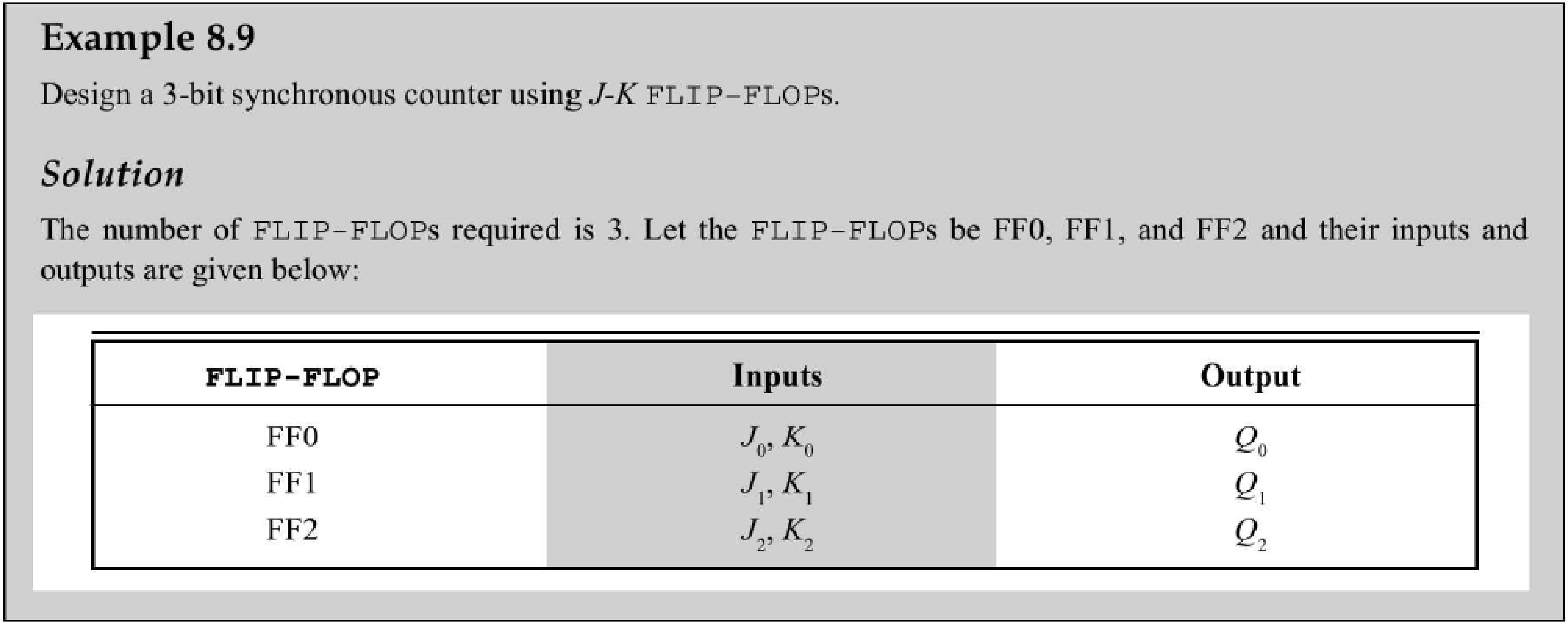
Synchronous Counter Design



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

1. Find the number of FLIP-FLOPs required
2. Write the count sequence in the tabular form
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
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.

Synchronous Counter Design Example 1



Example 8.9

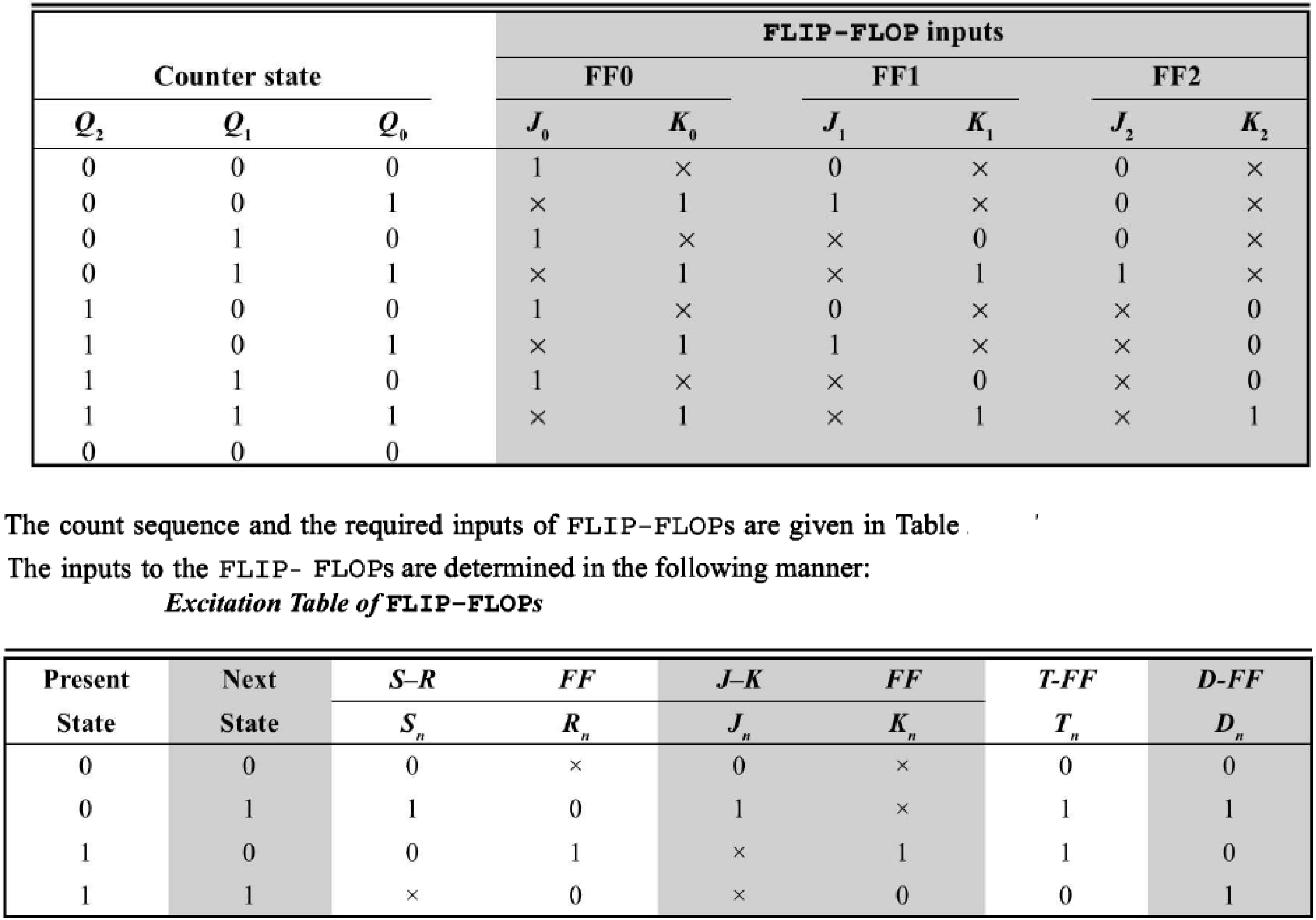
Design a 3-bit synchronous counter using *J-K* FLIP-FLOPs.

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

Synchronous Counter Design Example



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						

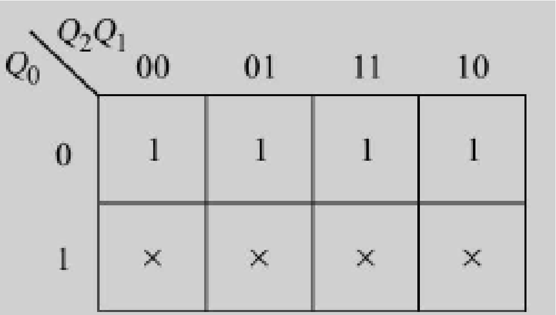
The count sequence and the required inputs of FLIP-FLOPs are given in Table .

The inputs to the FLIP- FLOPs are determined in the following manner:

Excitation Table of **FLIP-FLOPs**

Present	Next	*S–R*	*FF*	*J–K*	*FF*	*T-FF*	*D-FF*
State	State	S_n	R_n	J_n	K_n	T_n	D_n
0	0	0	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1

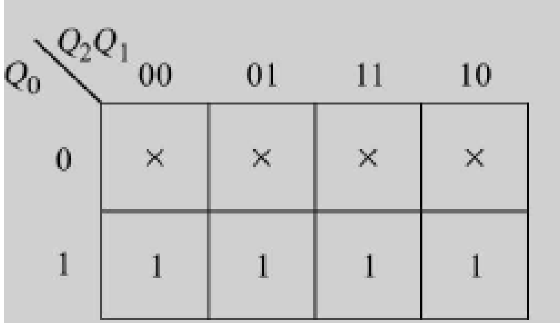
Synchronous Counter Design Example



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

$J_0 = 1$

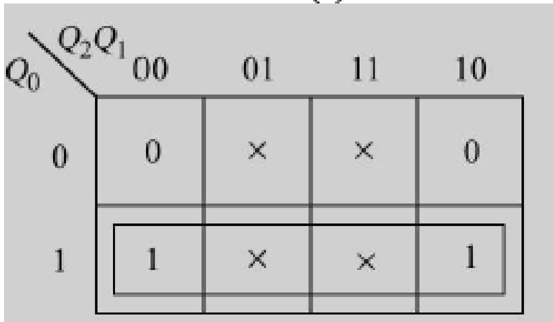
(a)



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

$K_0 = 1$

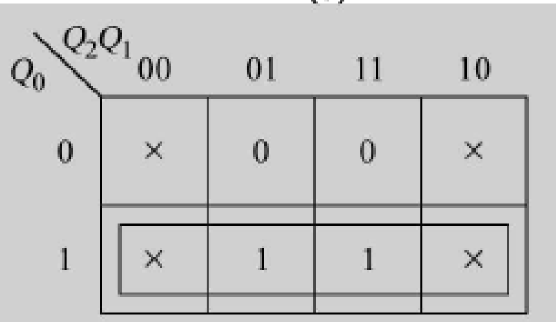
(b)



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

$J_1 = Q_0$

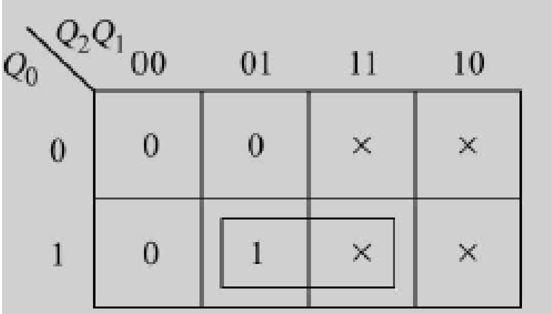
(c)



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

$K_1 = Q_0$

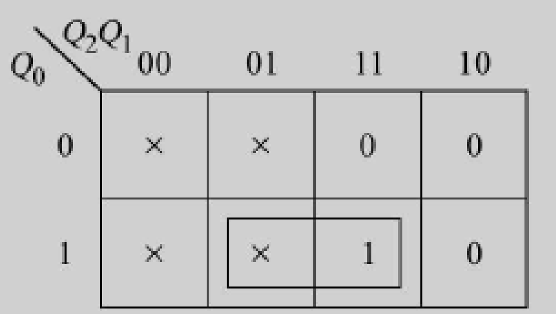
(d)



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

$J_2 = Q_0Q_1$

(e)

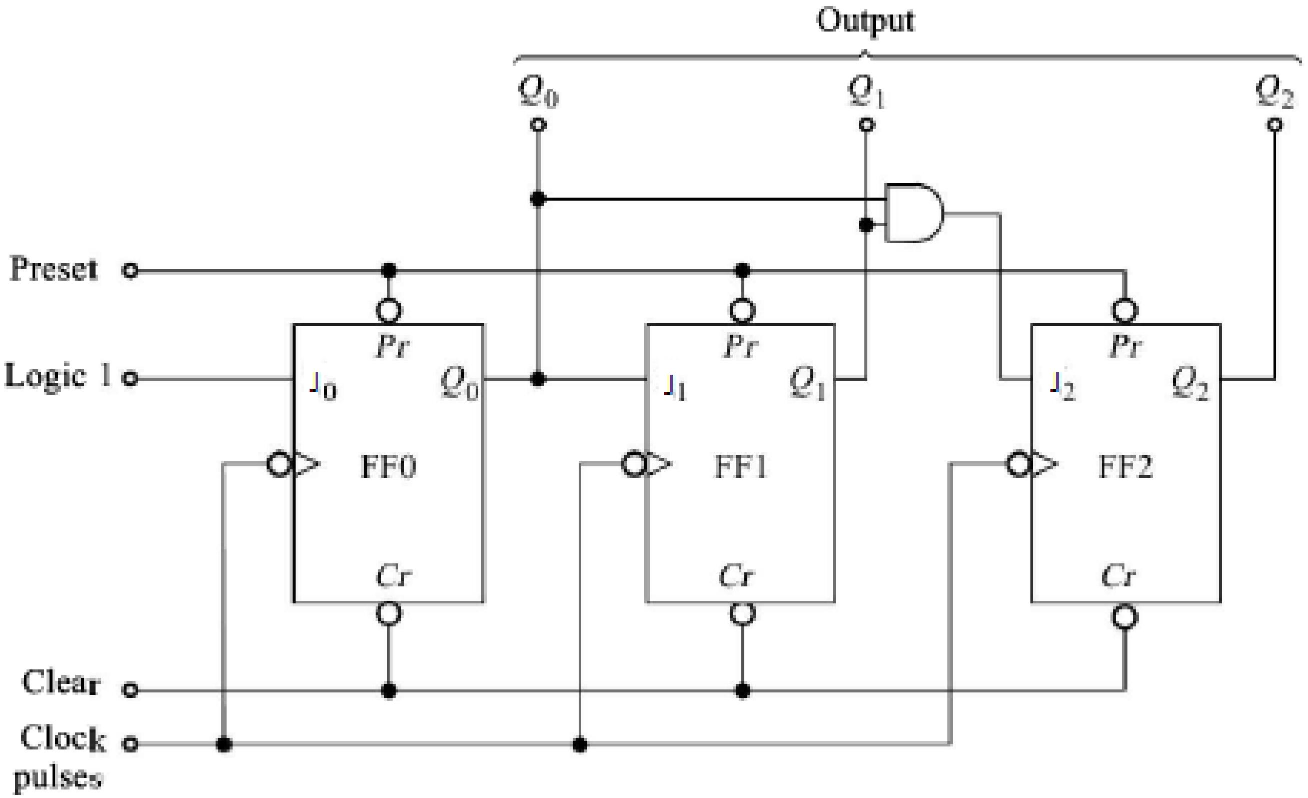


Q_0 \ Q_2Q_1	00	01	11	10
0	×	×	0	0
1	×	×	1	0

$K_2 = Q_0Q_1$

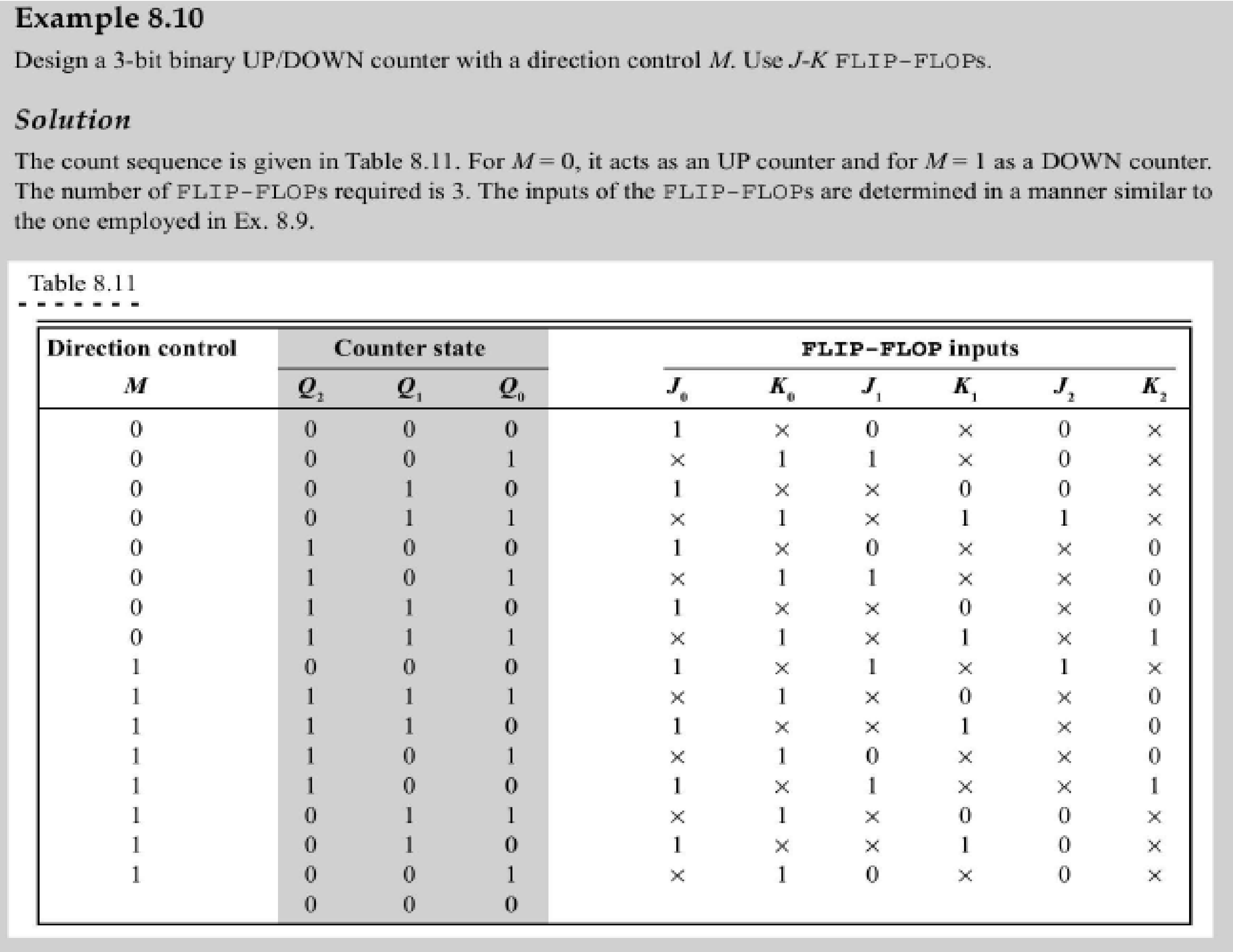
(f)

Synchronous Counter Design Example



A 3-bit Synchronous Counter

Synchronous Counter Design Example 2



Example 8.10

Design a 3-bit binary UP/DOWN counter with a direction control *M*. Use *J-K* FLIP-FLOPs.

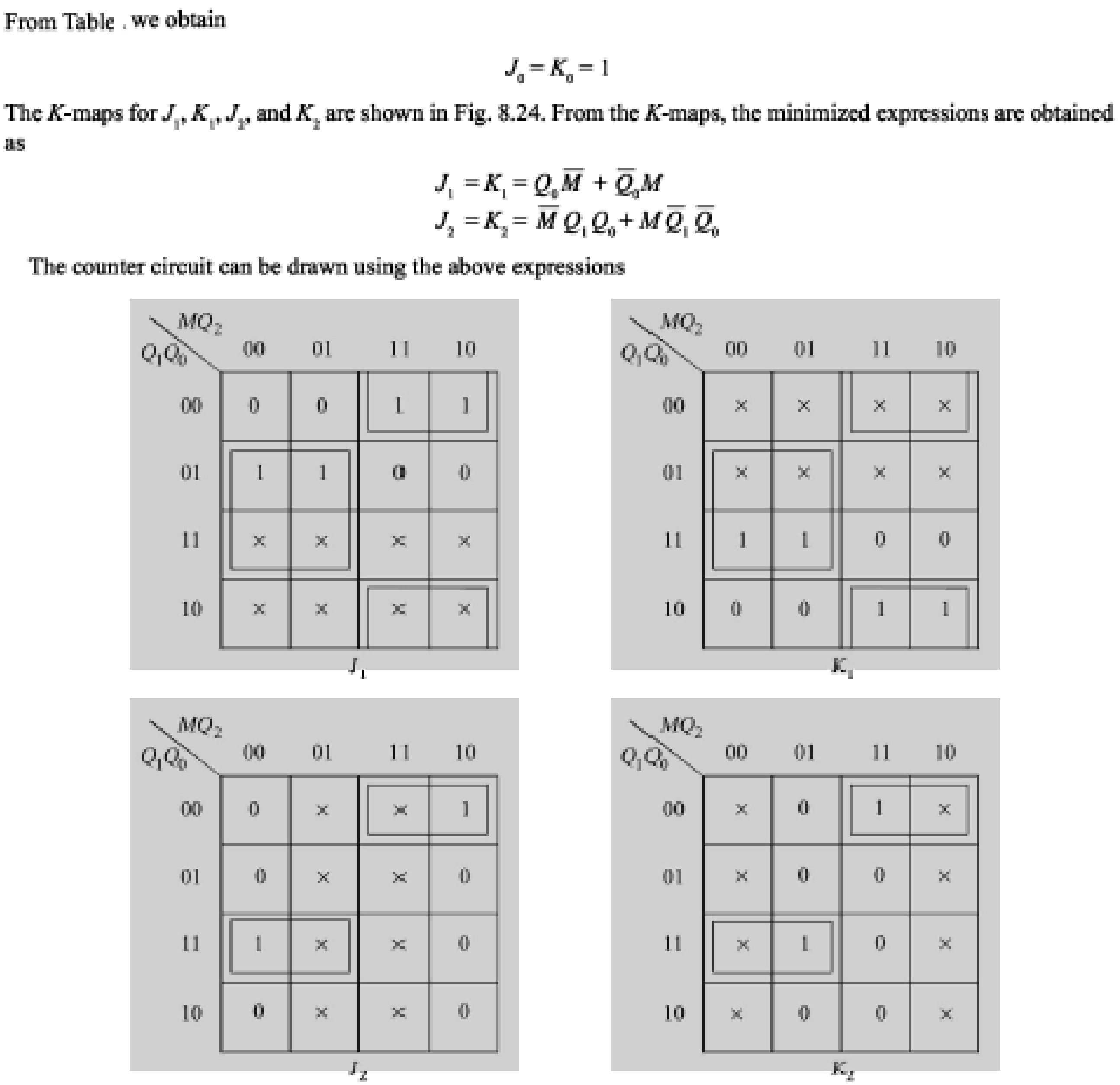
Solution

The count sequence is given in Table 8.11. For $M = 0$, it acts as an UP counter and for $M = 1$ as a DOWN counter. The number of FLIP-FLOPs required is 3. The inputs of the FLIP-FLOPs are determined in a manner similar to the one employed in Ex. 8.9.

Table 8.11

Direction control	Counter state			FLIP-FLOP inputs					
M	Q_2	Q_1	Q_0	J_0	K_0	J_1	K_1	J_2	K_2
0	0	0	0	1	×	0	×	0	×
0	0	0	1	×	1	1	×	0	×
0	0	1	0	1	×	×	0	0	×
0	0	1	1	×	1	×	1	1	×
0	1	0	0	1	×	0	×	×	0
0	1	0	1	×	1	1	×	×	0
0	1	1	0	1	×	×	0	×	0
0	1	1	1	×	1	×	1	×	1
1	0	0	0	1	×	1	×	1	×
1	1	1	1	×	1	×	0	×	0
1	1	1	0	1	×	×	1	×	0
1	1	0	1	×	1	0	×	×	0
1	1	0	0	1	×	1	×	×	1
1	0	1	1	×	1	×	0	0	×
1	0	1	0	1	×	×	1	0	×
1	0	0	1	×	1	0	×	0	×
	0	0	0						

Synchronous Counter Design Example



From Table . we obtain

$$J_0 = K_0 = 1$$

The K-maps for J_1, K_1, J_2, and K_2 are shown in Fig. 8.24. From the K-maps, the minimized expressions are obtained as

$$J_1 = K_1 = Q_0\overline{M} + \overline{Q}_0 M$$
$$J_2 = K_2 = \overline{M}\, Q_1 Q_0 + M\,\overline{Q}_1\,\overline{Q}_0$$

The counter circuit can be drawn using the above expressions

Q_1Q_0 \ MQ_2	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	×	×	×	×
10	×	×	×	×

J_1

Q_1Q_0 \ MQ_2	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	1	1	0	0
10	0	0	1	1

K_1

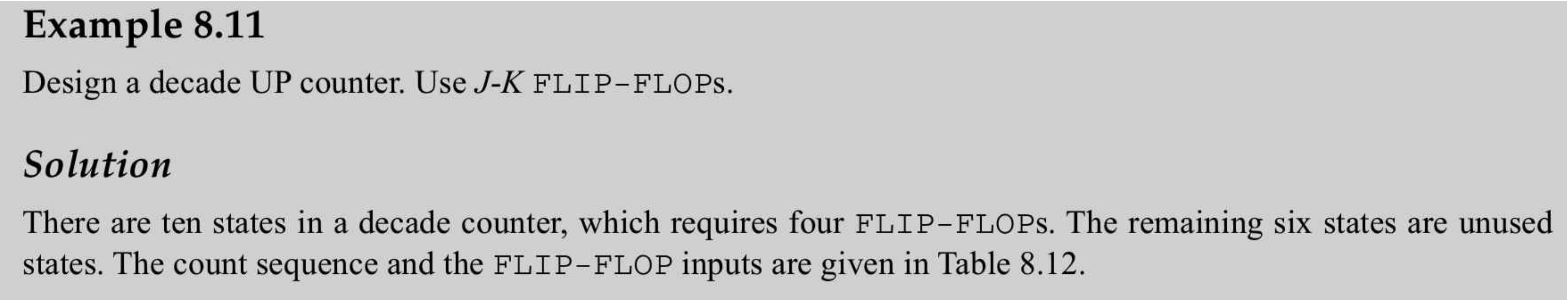
Q_1Q_0 \ MQ_2	00	01	11	10
00	0	×	×	1
01	0	×	×	0
11	1	×	×	0
10	0	×	×	0

J_2

Q_1Q_0 \ MQ_2	00	01	11	10
00	×	0	1	×
01	×	0	0	×
11	×	1	0	×
10	×	0	0	×

K_2

Synchronous Counter Design Example 3



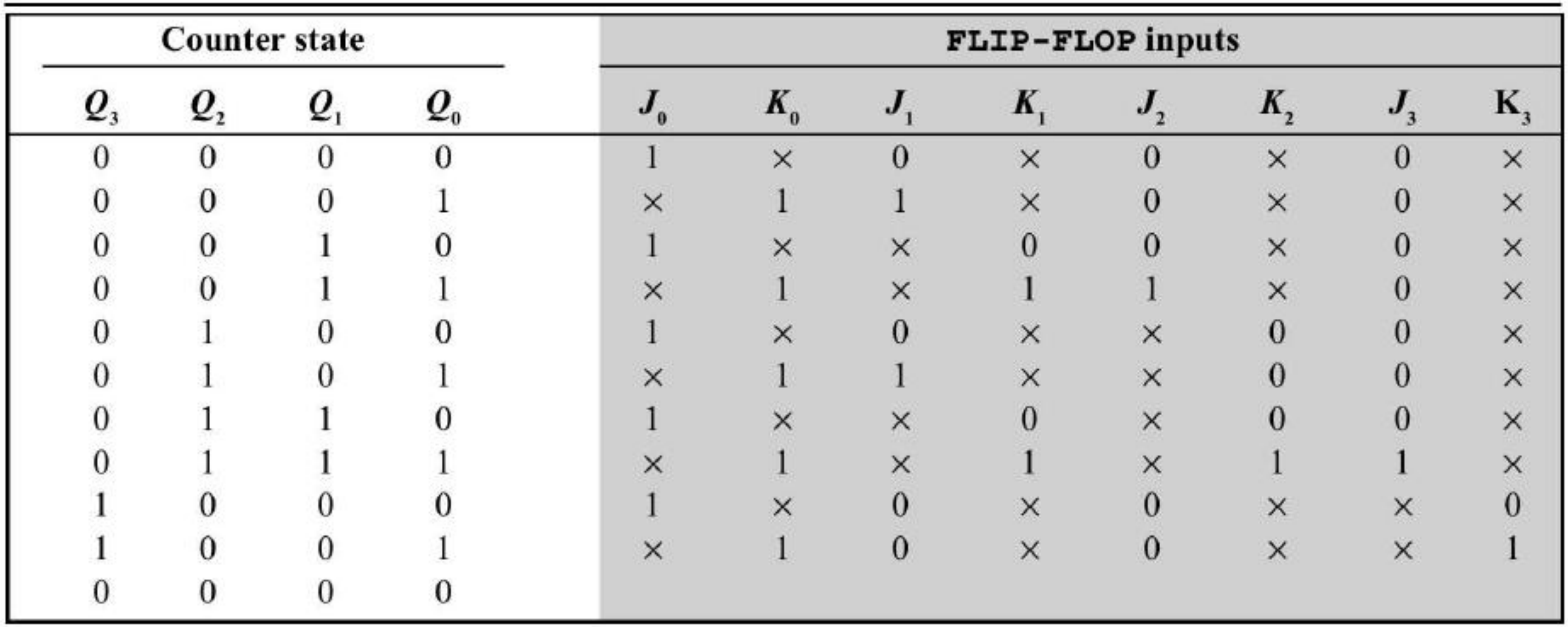
Example 8.11

Design a decade UP counter. Use *J-K* FLIP-FLOPs.

Solution

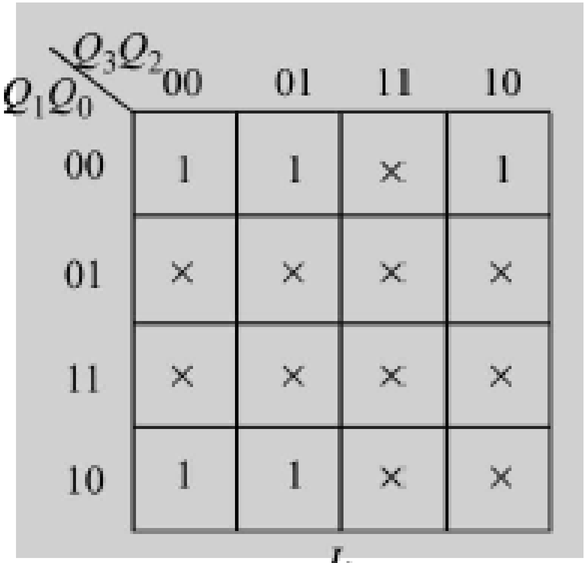
There are ten states in a decade counter, which requires four FLIP-FLOPs. The remaining six states are unused states. The count sequence and the FLIP-FLOP inputs are given in Table 8.12.

Table --------



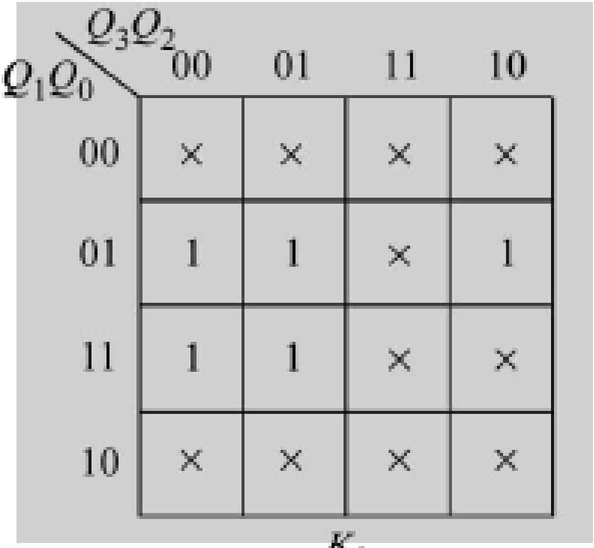
Counter state				FLIP-FLOP inputs							
Q_3	Q_2	Q_1	Q_0	J_0	K_0	J_1	K_1	J_2	K_2	J_3	K_3
0	0	0	0	1	×	0	×	0	×	0	×
0	0	0	1	×	1	1	×	0	×	0	×
0	0	1	0	1	×	×	0	0	×	0	×
0	0	1	1	×	1	×	1	1	×	0	×
0	1	0	0	1	×	0	×	×	0	0	×
0	1	0	1	×	1	1	×	×	0	0	×
0	1	1	0	1	×	×	0	×	0	0	×
0	1	1	1	×	1	×	1	×	1	1	×
1	0	0	0	1	×	0	×	0	×	×	0
1	0	0	1	×	1	0	×	0	×	×	1
0	0	0	0								

Synchronous Counter Design Example 3



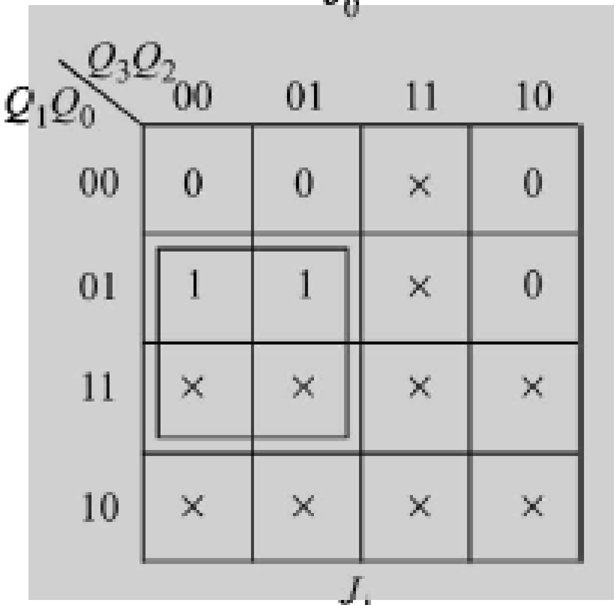
$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	1	1	×	1
01	×	×	×	×
11	×	×	×	×
10	1	1	×	×

J_0



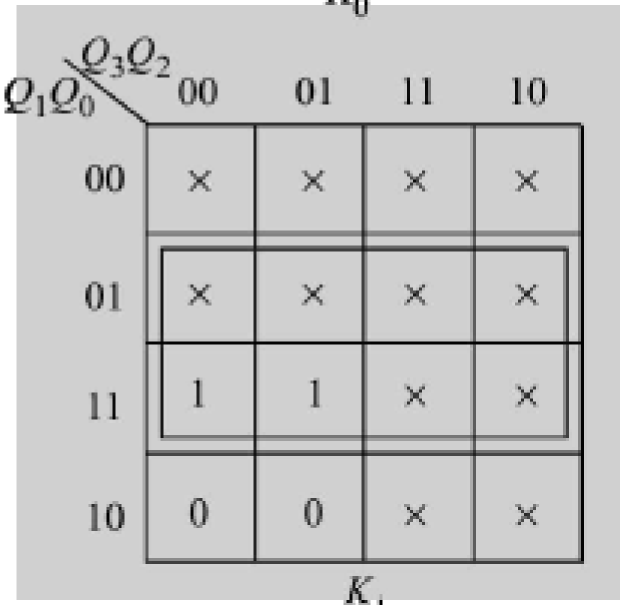
$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	×	×	×	×
01	1	1	×	1
11	1	1	×	×
10	×	×	×	×

K_0



$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	0	0	×	0
01	1	1	×	0
11	×	×	×	×
10	×	×	×	×

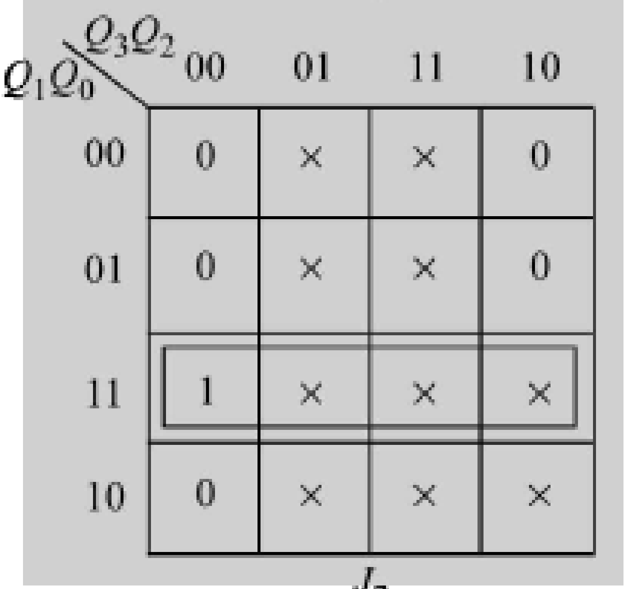
J_1



$Q_1Q_0 \backslash Q_3Q_2$	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	1	1	×	×
10	0	0	×	×

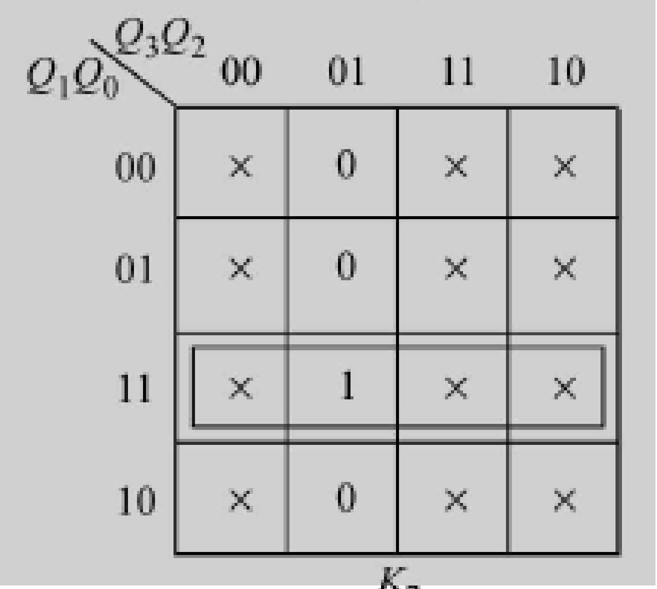
K_1

Synchronous Counter Design Example 3



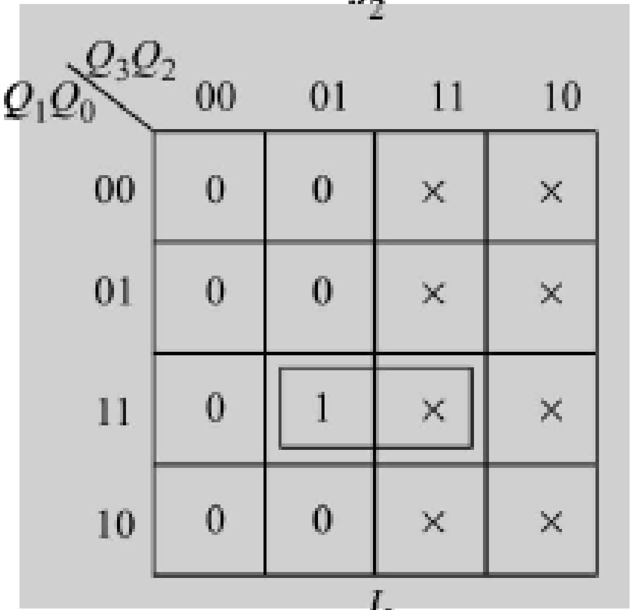
Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	0	×	×	0
01	0	×	×	0
11	1	×	×	×
10	0	×	×	×

J_2



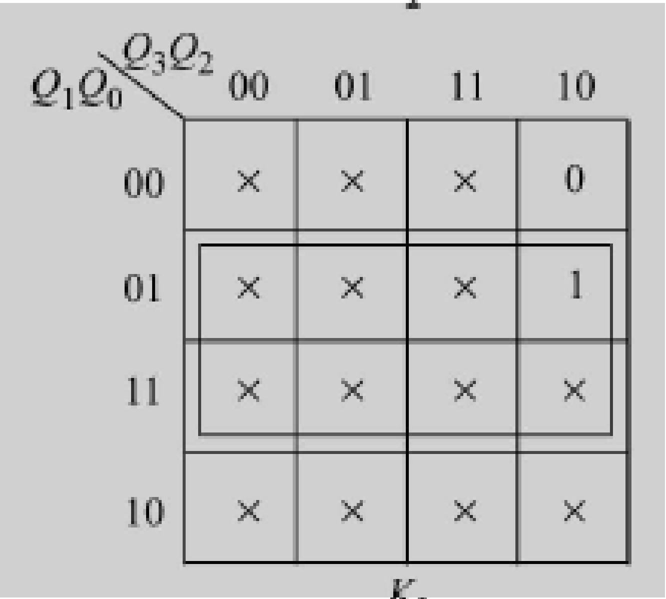
Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	×	0	×	×
01	×	0	×	×
11	×	1	×	×
10	×	0	×	×

K_2



Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	0	0	×	×
01	0	0	×	×
11	0	1	×	×
10	0	0	×	×

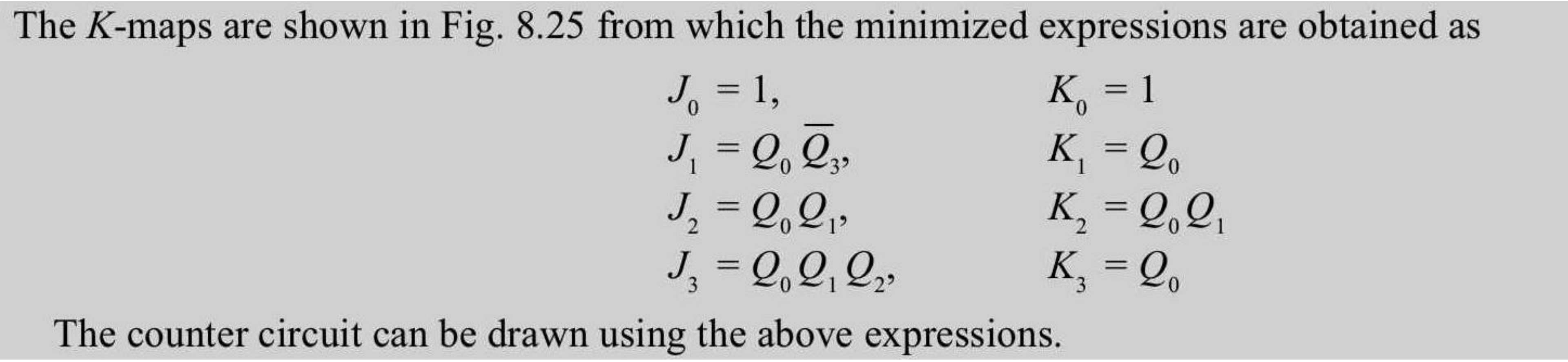
J_3



Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	×	×	×	0
01	×	×	×	1
11	×	×	×	×
10	×	×	×	×

K_3

Synchronous Counter Design Example 3

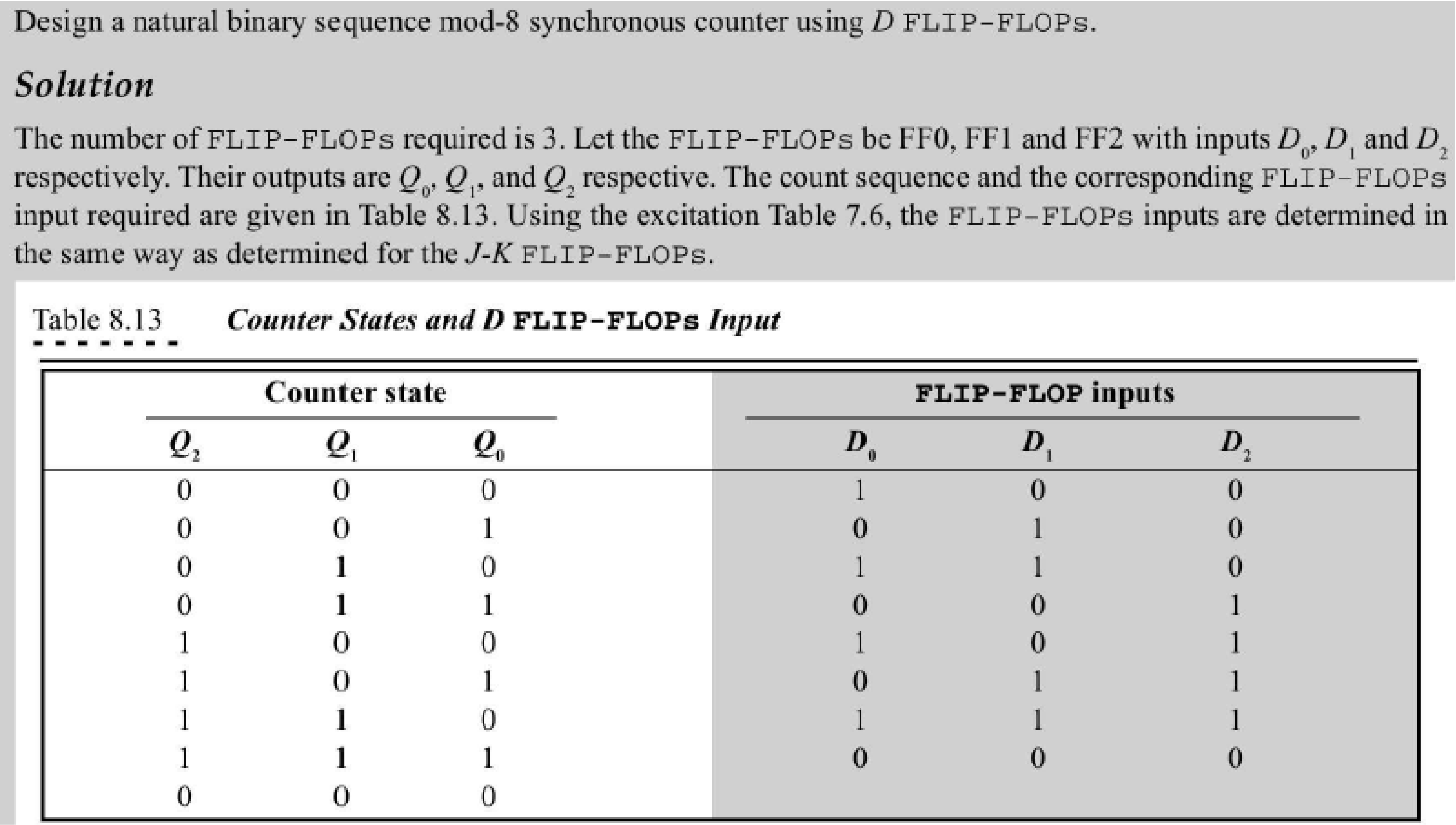


The *K*-maps are shown in Fig. 8.25 from which the minimized expressions are obtained as

$$\begin{aligned} J_0 &= 1, & K_0 &= 1 \\ J_1 &= Q_0\,\overline{Q}_3, & K_1 &= Q_0 \\ J_2 &= Q_0\,Q_1, & K_2 &= Q_0\,Q_1 \\ J_3 &= Q_0\,Q_1\,Q_2, & K_3 &= Q_0 \end{aligned}$$

The counter circuit can be drawn using the above expressions.

Synchronous Counter Design Example 4



Design a natural binary sequence mod-8 synchronous counter using *D* FLIP-FLOPs.

Solution

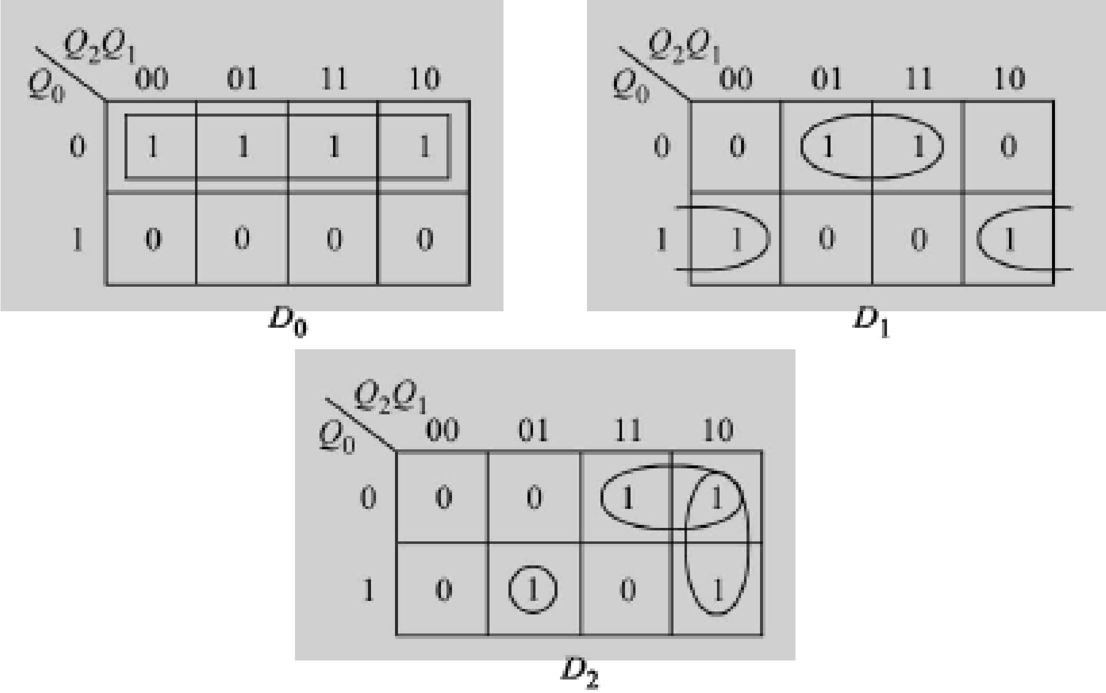
The number of FLIP-FLOPs required is 3. Let the FLIP-FLOPs be FF0, FF1 and FF2 with inputs D_0, D_1 and D_2 respectively. Their outputs are Q_0, Q_1, and Q_2 respective. The count sequence and the corresponding FLIP-FLOPs input required are given in Table 8.13. Using the excitation Table 7.6, the FLIP-FLOPs inputs are determined in the same way as determined for the *J-K* FLIP-FLOPs.

Table 8.13 ***Counter States and D* FLIP-FLOPs *Input***

Counter state			FLIP-FLOP inputs		
Q_2	Q_1	Q_0	D_0	D_1	D_2
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	0	0	0
0	0	0			

Synchronous Counter Design Example 4

The *K*-maps for D_0, D_1, and D_2 are given in Fig. 8.26.



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

D_0

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

D_1

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

D_2

Fig. 8.26 ***K-Maps of Ex. 8.12***

The minimised expressions for D_0, D_1, and D_2 are:

$$D_0 = \bar{Q}_0 \qquad D_1 = Q_1\bar{Q}_0 + \bar{Q}_1 Q_0$$

$$\begin{aligned} D_2 &= Q_2\bar{Q}_0 + Q_2\bar{Q}_1 + \bar{Q}_2 Q_1 Q_0 \\ &= Q_2(\bar{Q}_0 + \bar{Q}_1) + \bar{Q}_2 Q_1 Q_0 \\ &= Q_2(\overline{Q_0 \cdot Q_1}) + \bar{Q}_2(Q_1 Q_0) \\ &= Q_2 \oplus Q_1 \cdot Q_0 \end{aligned}$$

Synchronous Counter Design Example 4

The complete circuit of the synchronous counter using positive edge triggered D FLIP-FLOPs is shown in Fig. 8.27.

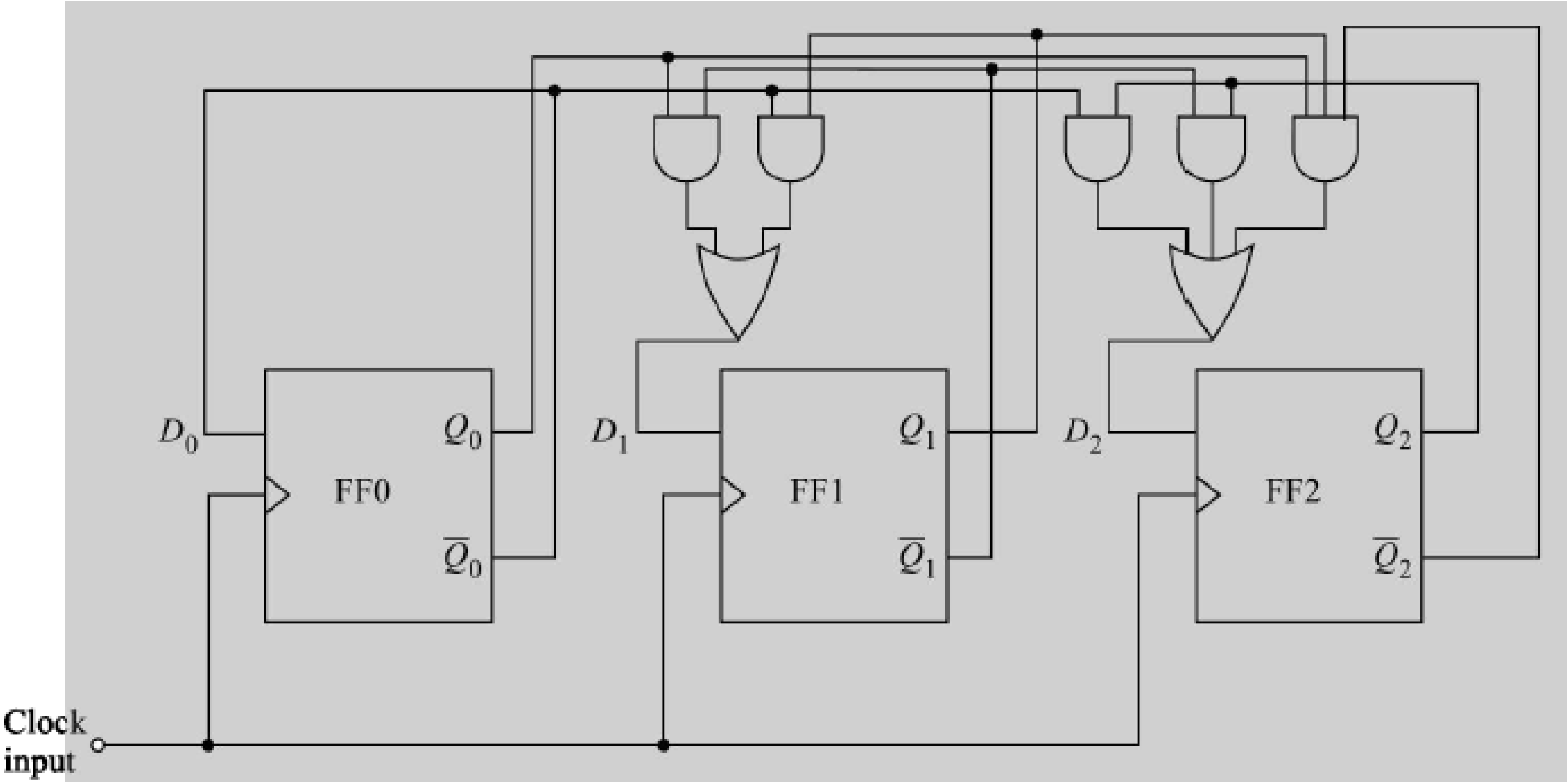


Fig. 8.27 ***Synchronous Counter Circuit of Ex. 8.12***

Asynchronous Counter ICs

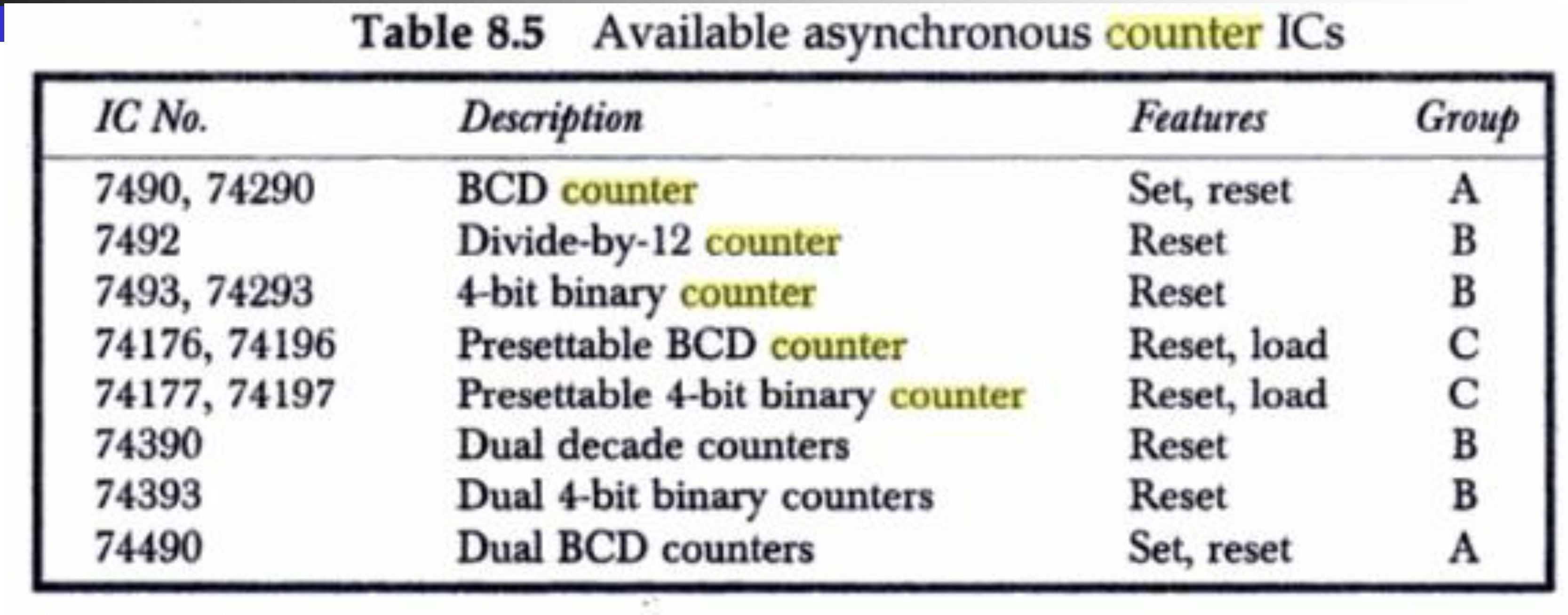


Table 8.5 Available asynchronous counter ICs

IC No.	*Description*	*Features*	*Group*
7490, 74290	BCD counter	Set, reset	A
7492	Divide-by-12 counter	Reset	B
7493, 74293	4-bit binary counter	Reset	B
74176, 74196	Presettable BCD counter	Reset, load	C
74177, 74197	Presettable 4-bit binary counter	Reset, load	C
74390	Dual decade counters	Reset	B
74393	Dual 4-bit binary counters	Reset	B
74490	Dual BCD counters	Set, reset	A

- **Group A Asynchronous Counter ICs**
- **Group B Asynchronous Counter ICs**
- **Group C Asynchronous Counter ICs**

Synchronous Counter ICs

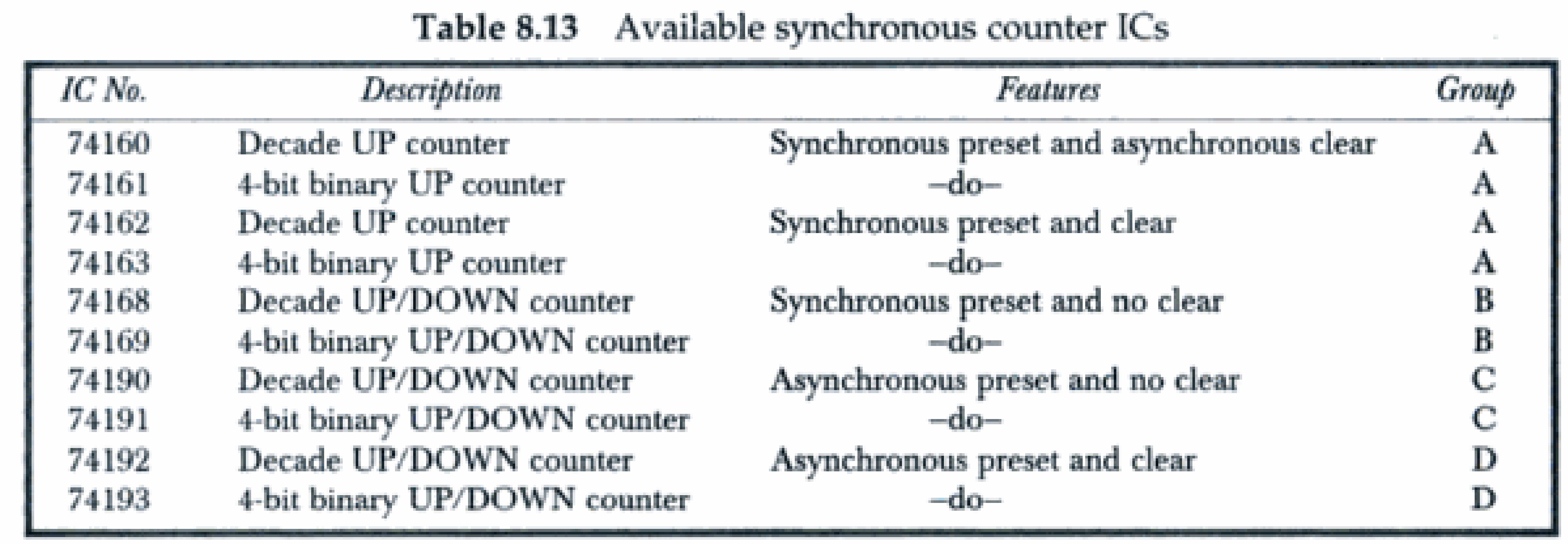


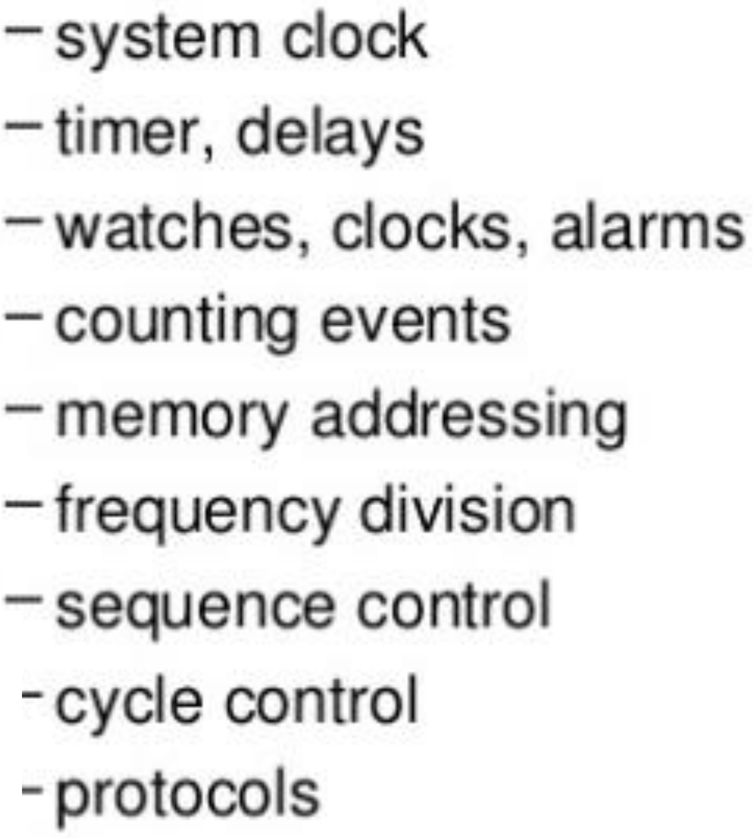
Table 8.13 Available synchronous counter ICs

IC No.	*Description*	*Features*	*Group*
74160	Decade UP counter	Synchronous preset and asynchronous clear	A
74161	4-bit binary UP counter	–do–	A
74162	Decade UP counter	Synchronous preset and clear	A
74163	4-bit binary UP counter	–do–	A
74168	Decade UP/DOWN counter	Synchronous preset and no clear	B
74169	4-bit binary UP/DOWN counter	–do–	B
74190	Decade UP/DOWN counter	Asynchronous preset and no clear	C
74191	4-bit binary UP/DOWN counter	–do–	C
74192	Decade UP/DOWN counter	Asynchronous preset and clear	D
74193	4-bit binary UP/DOWN counter	–do–	D

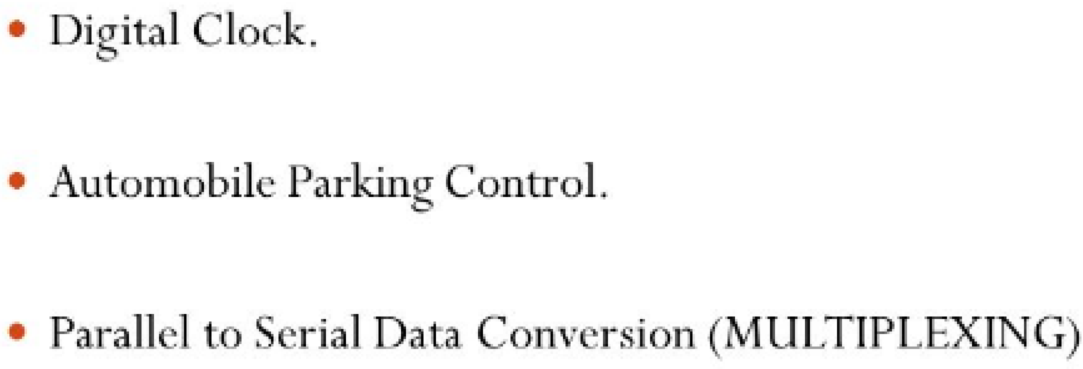
- **Group A Synchronous Counter ICs**
- **Group B Synchronous Counter ICs**
- **Group C Synchronous Counter ICs**

Applications of Counter

- Frequency counters
- Digital clock
- Time measurement
- A to D converter
- Frequency divider circuits



- system clock
- timer, delays
- watches, clocks, alarms
- counting events
- memory addressing
- frequency division
- sequence control
- cycle control
- protocols



- Digital Clock.
- Automobile Parking Control.
- Parallel to Serial Data Conversion (MULTIPLEXING)