DELD

Unit 4

Registers and Counters

Electronics Engineering Department

Registers

❑ An Array of Flip-Flops used to store binary information

❑ No. of Flip-Flops will be equal to the no of bits required to store the information

❑ Data can be entered serially or parallely

❑ It can also be used for Data movement

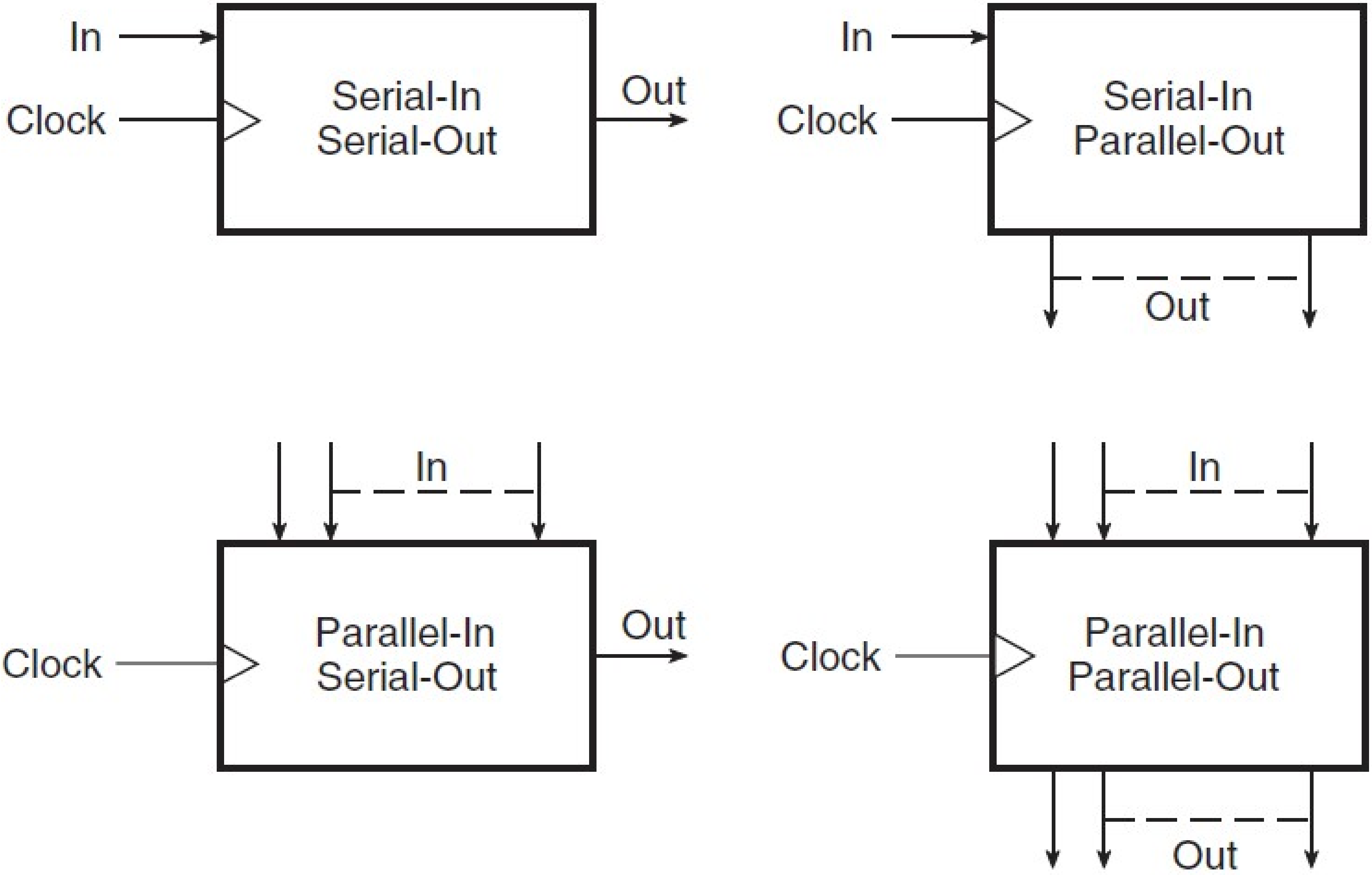
❑ Basically know as shift register, it shifts it output every clock pulse

Types

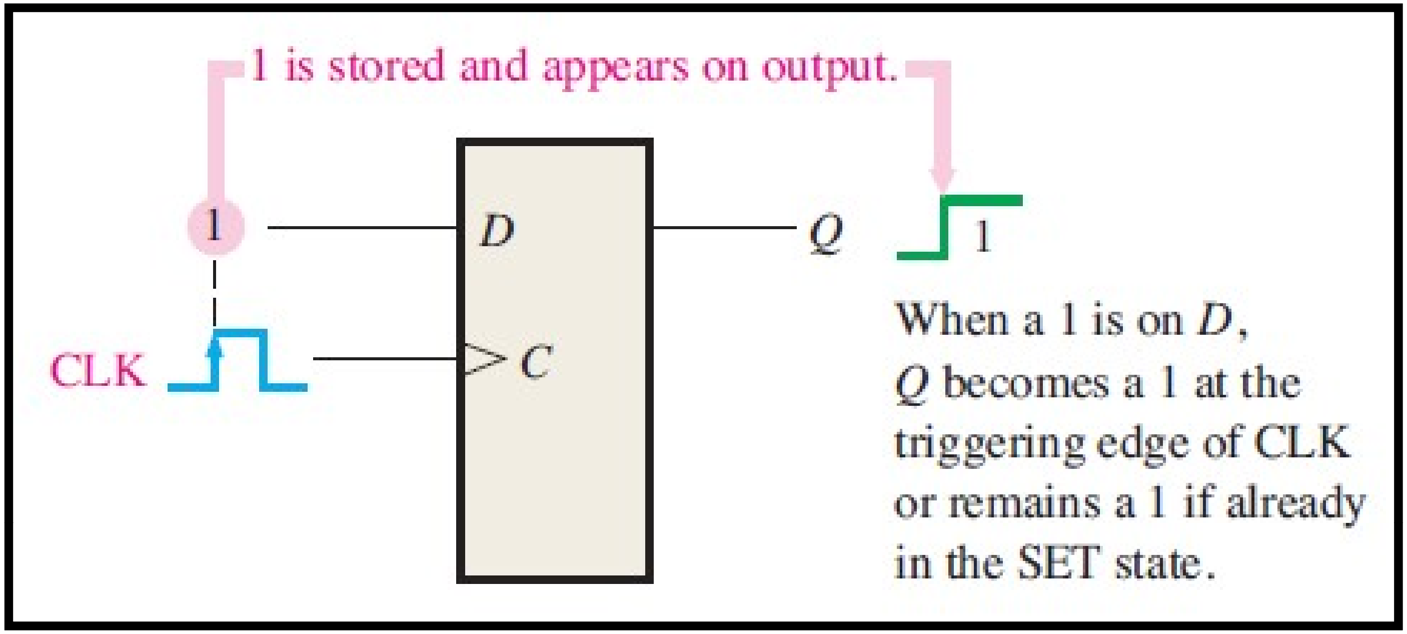
- Serial In Serial Out
- Serial In Parallel Out
- Parallel In Serial Out
- Parallel In Parallel Out

- Shift Left
- Shift Right

Types

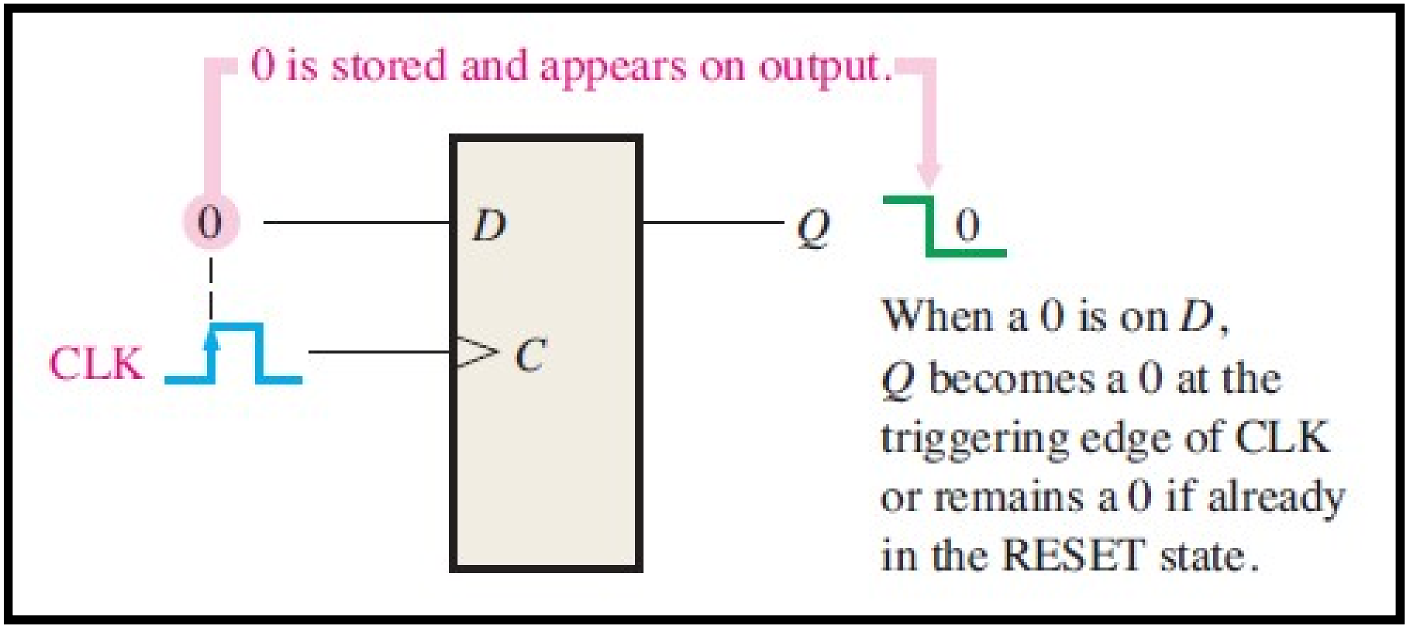

In
Clock
Serial-In
Serial-Out
Out
In
Clock
Serial-In
Parallel-Out
Out
In
Clock
Parallel-In
Serial-Out
Out
In
Clock
Parallel-In
Parallel-Out
Out

Working Concept



1 is stored and appears on output.

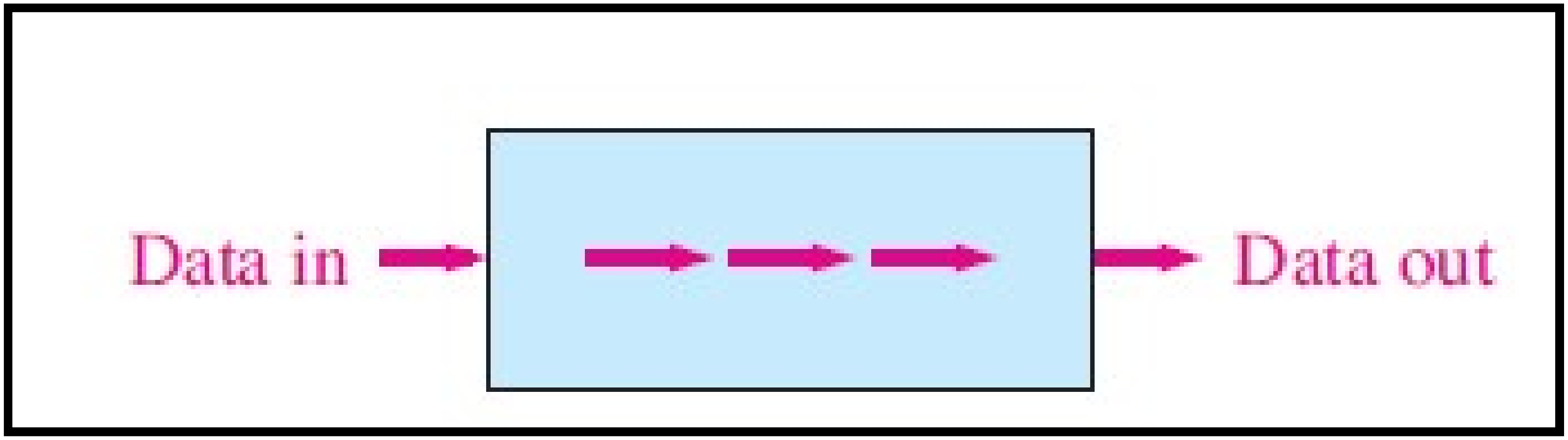
When a 1 is on *D*, *Q* becomes a 1 at the triggering edge of CLK or remains a 1 if already in the SET state.

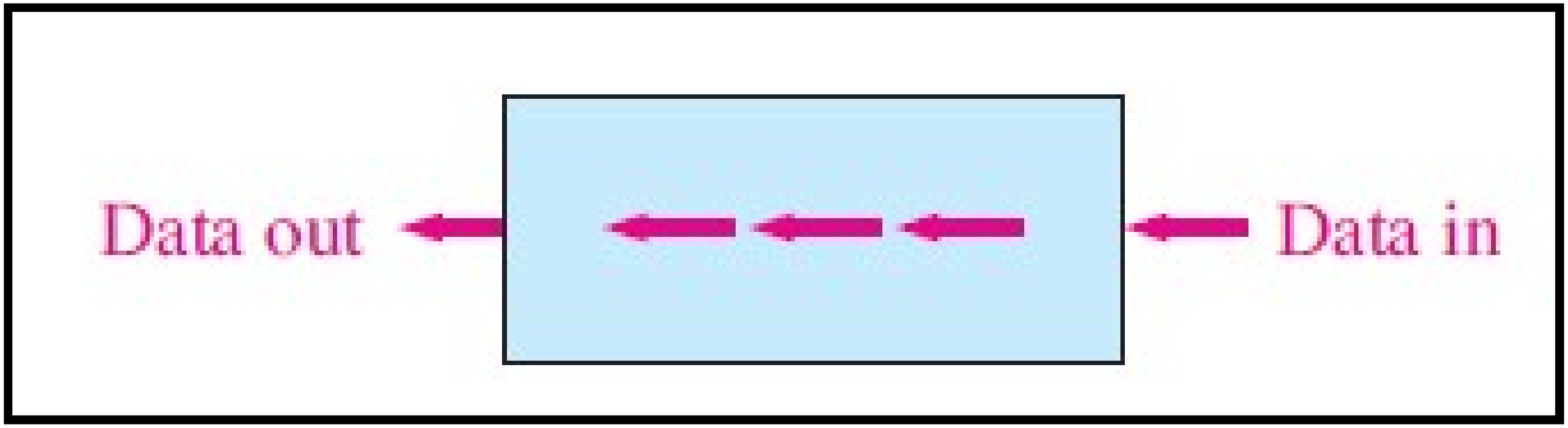


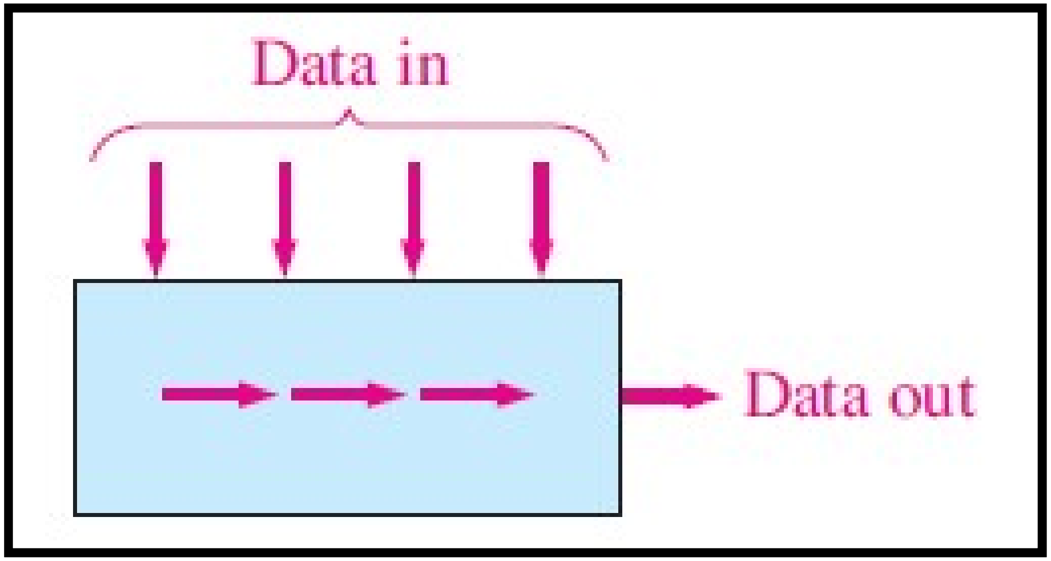
0 is stored and appears on output.

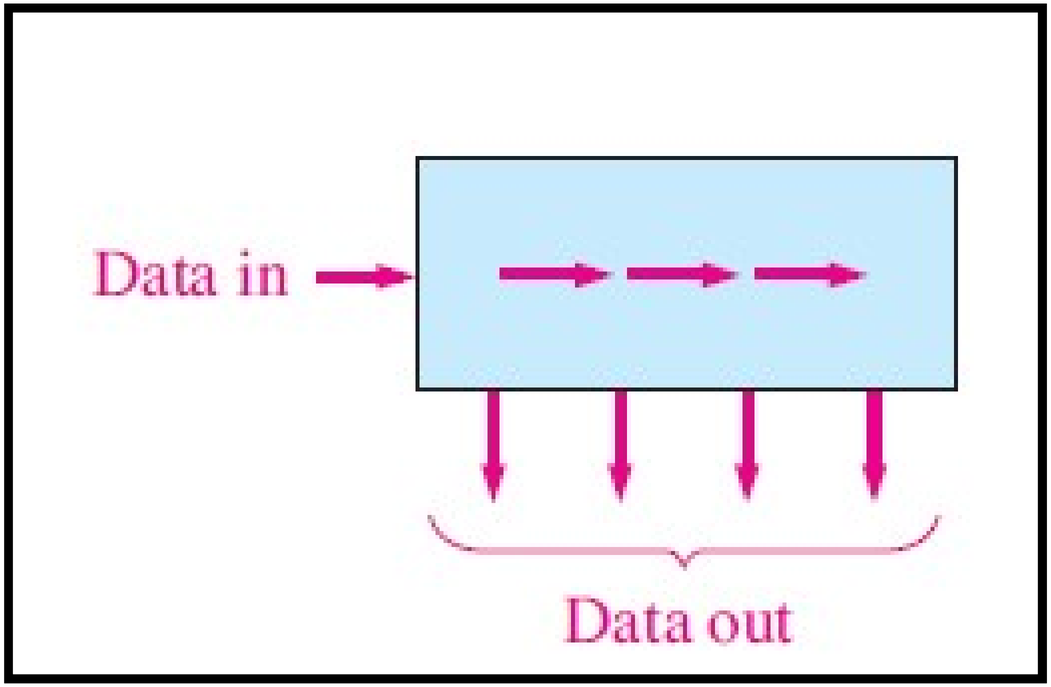
When a 0 is on *D*, *Q* becomes a 0 at the triggering edge of CLK or remains a 0 if already in the RESET state.

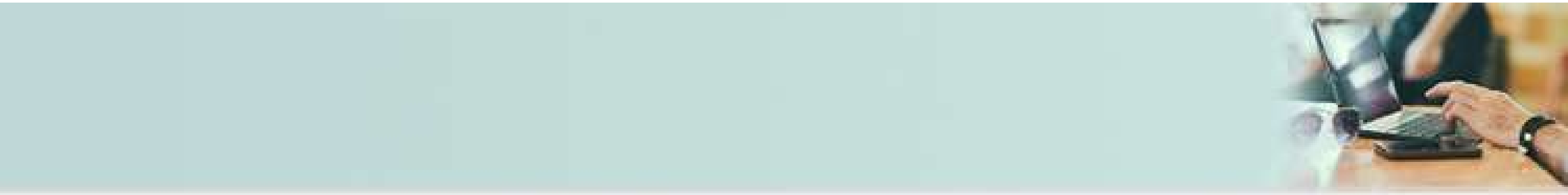
Another Perspective

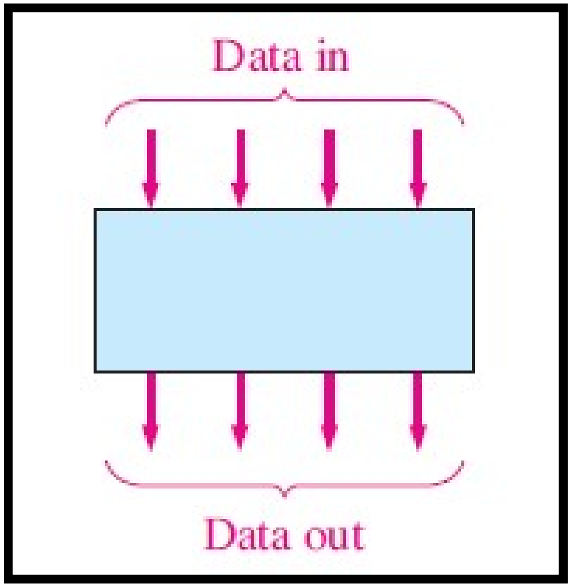


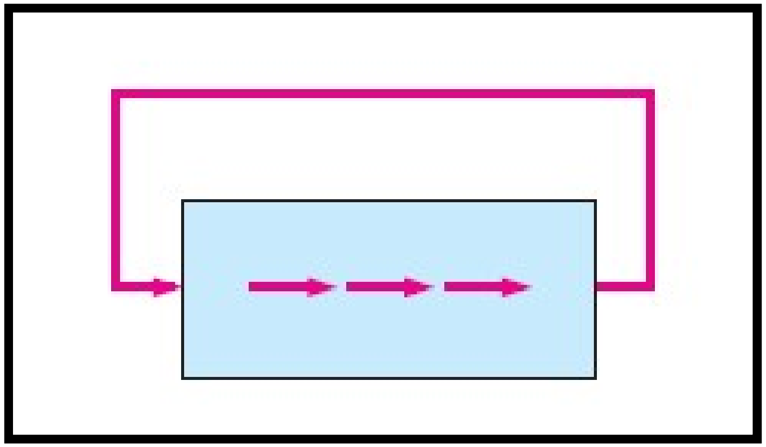


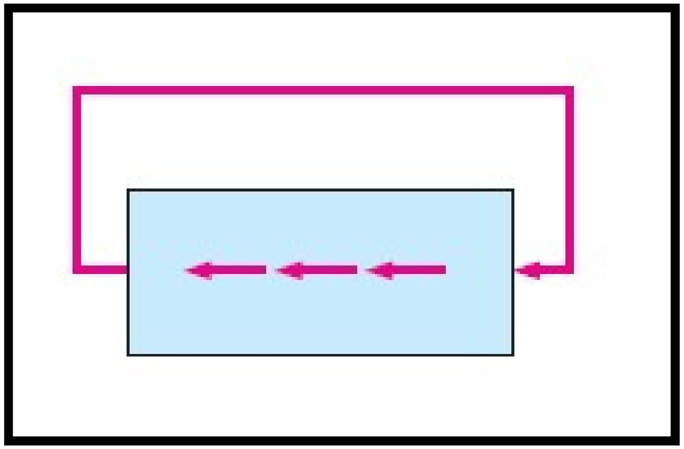

Data in
Data out


Data in
Data out

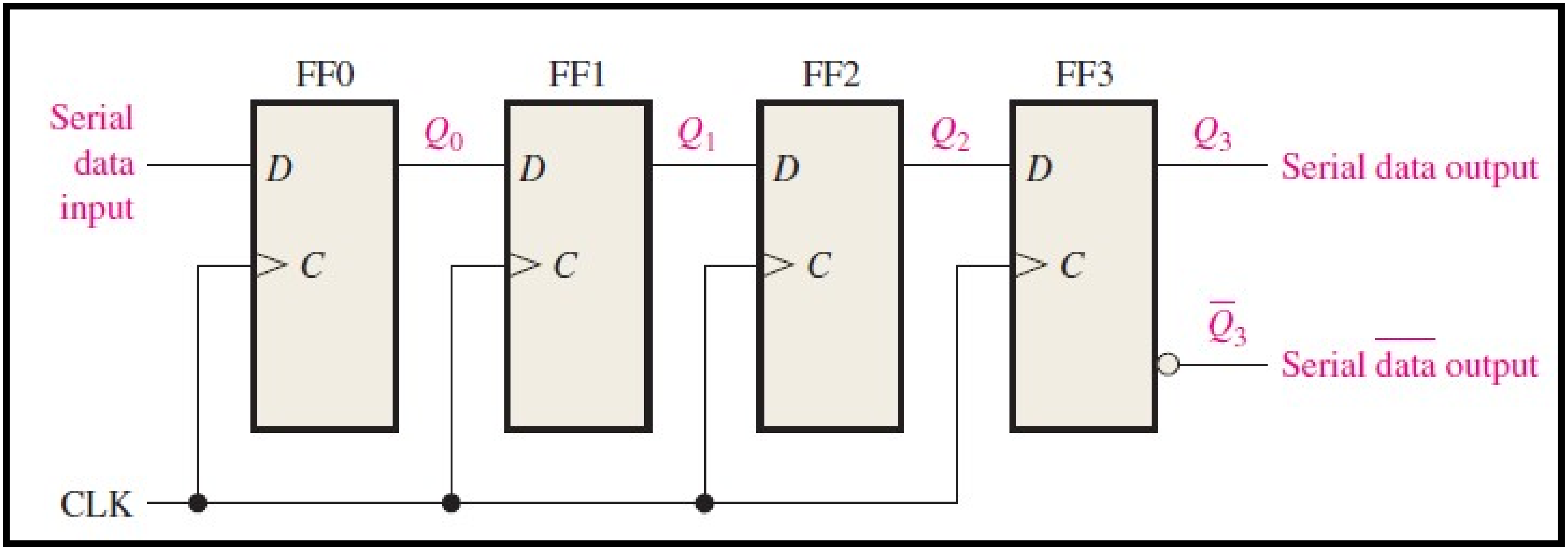



Data in
Data out

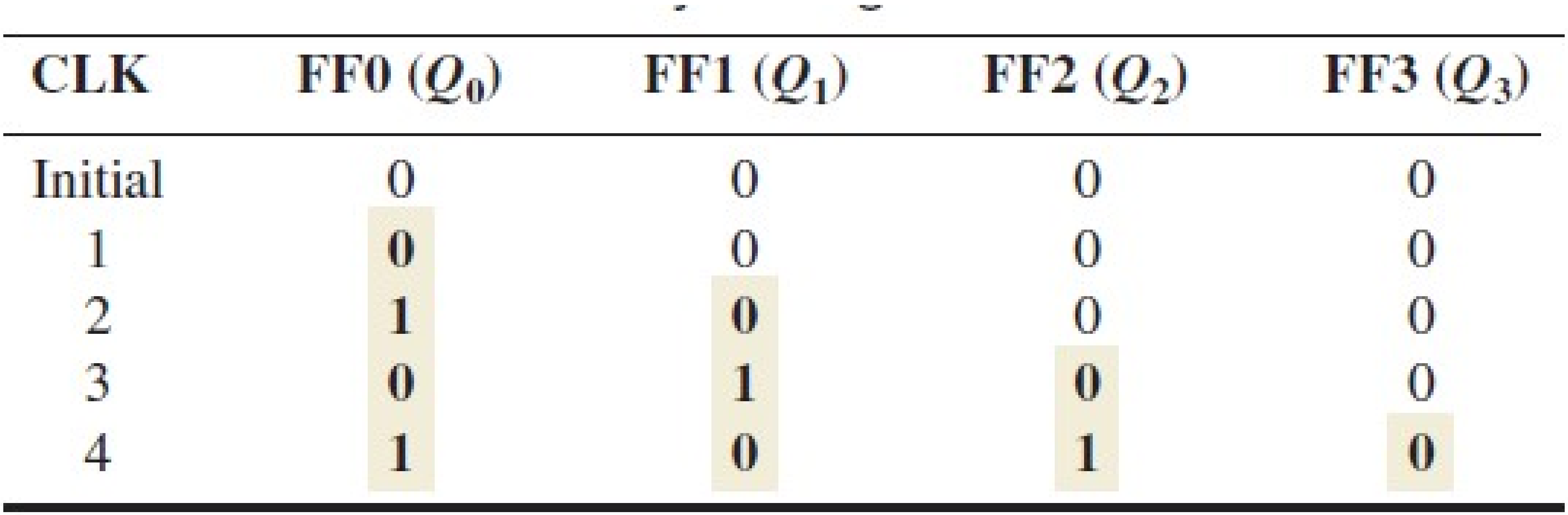




Serial In Serial Out

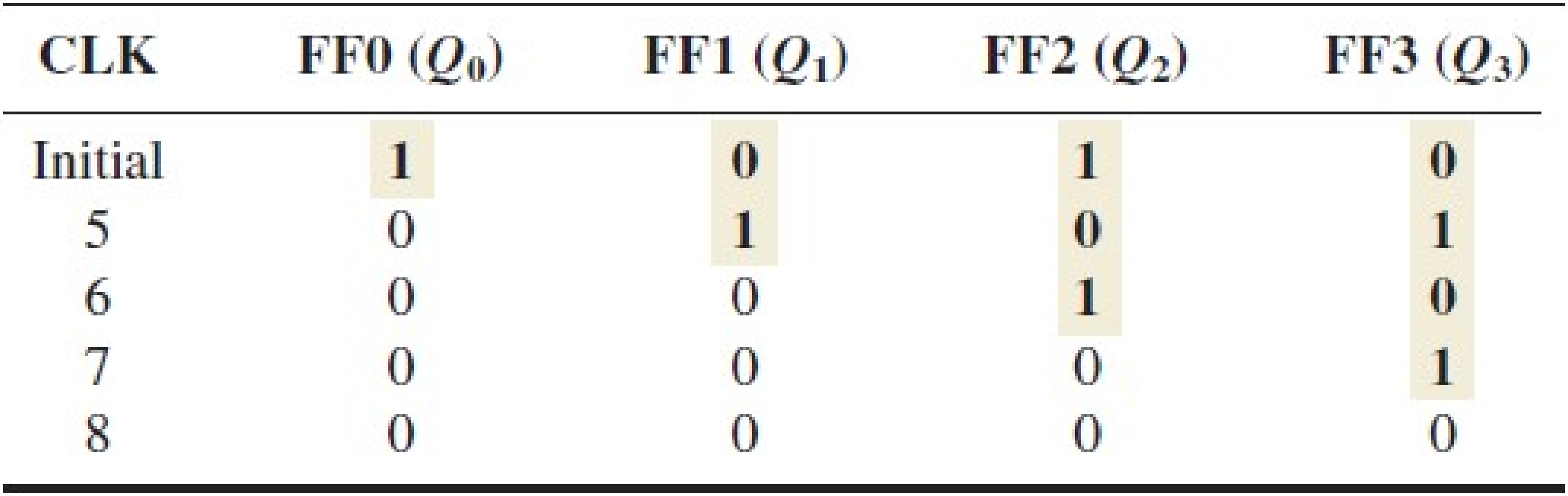


Bit Movement



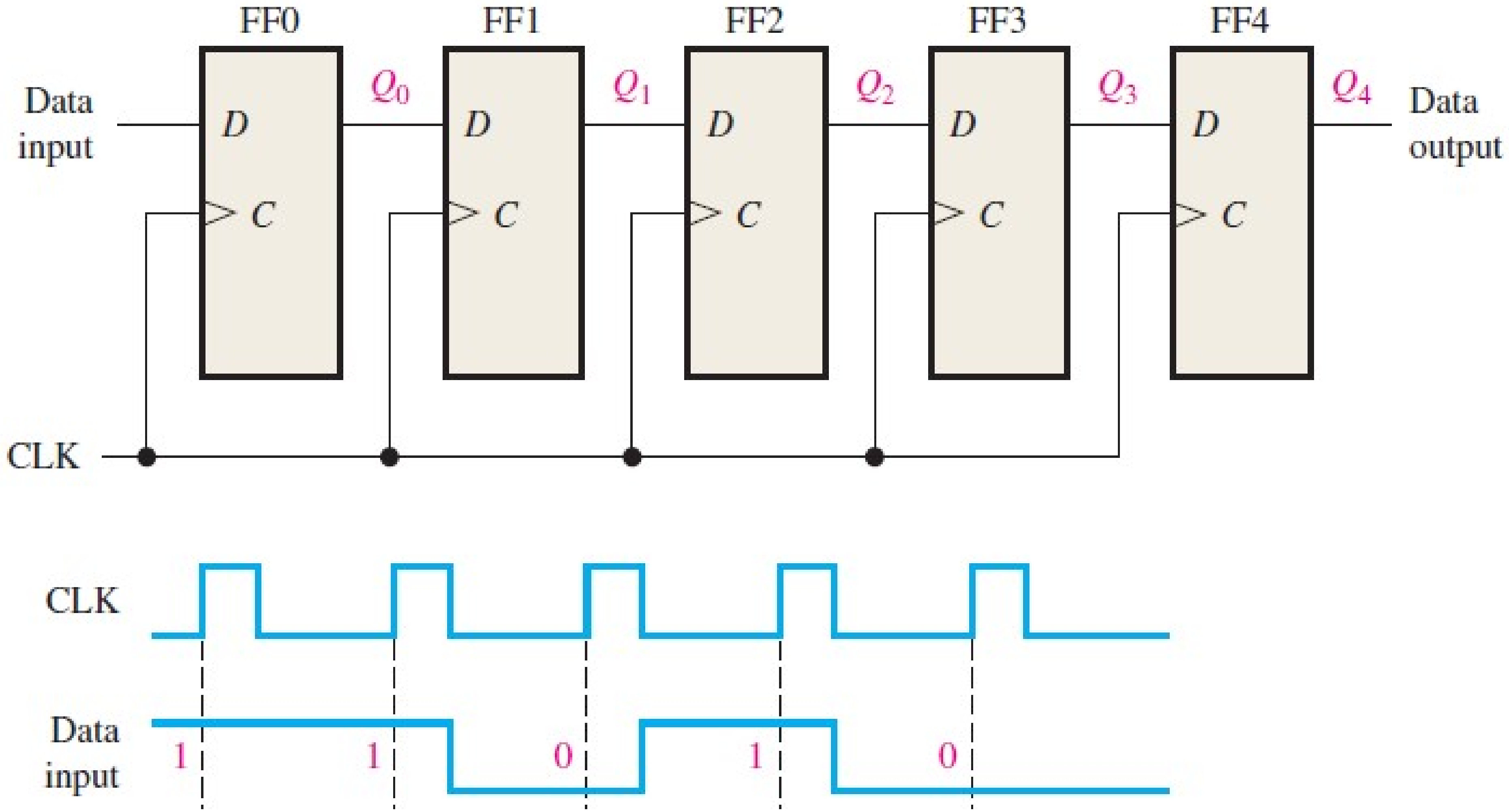
CLK	FF0 (Q_0)	FF1 (Q_1)	FF2 (Q_2)	FF3 (Q_3)
Initial	0	0	0	0
1	**0**	0	0	0
2	**1**	**0**	0	0
3	**0**	**1**	**0**	0
4	**1**	**0**	**1**	**0**

Bit Movement Cont..

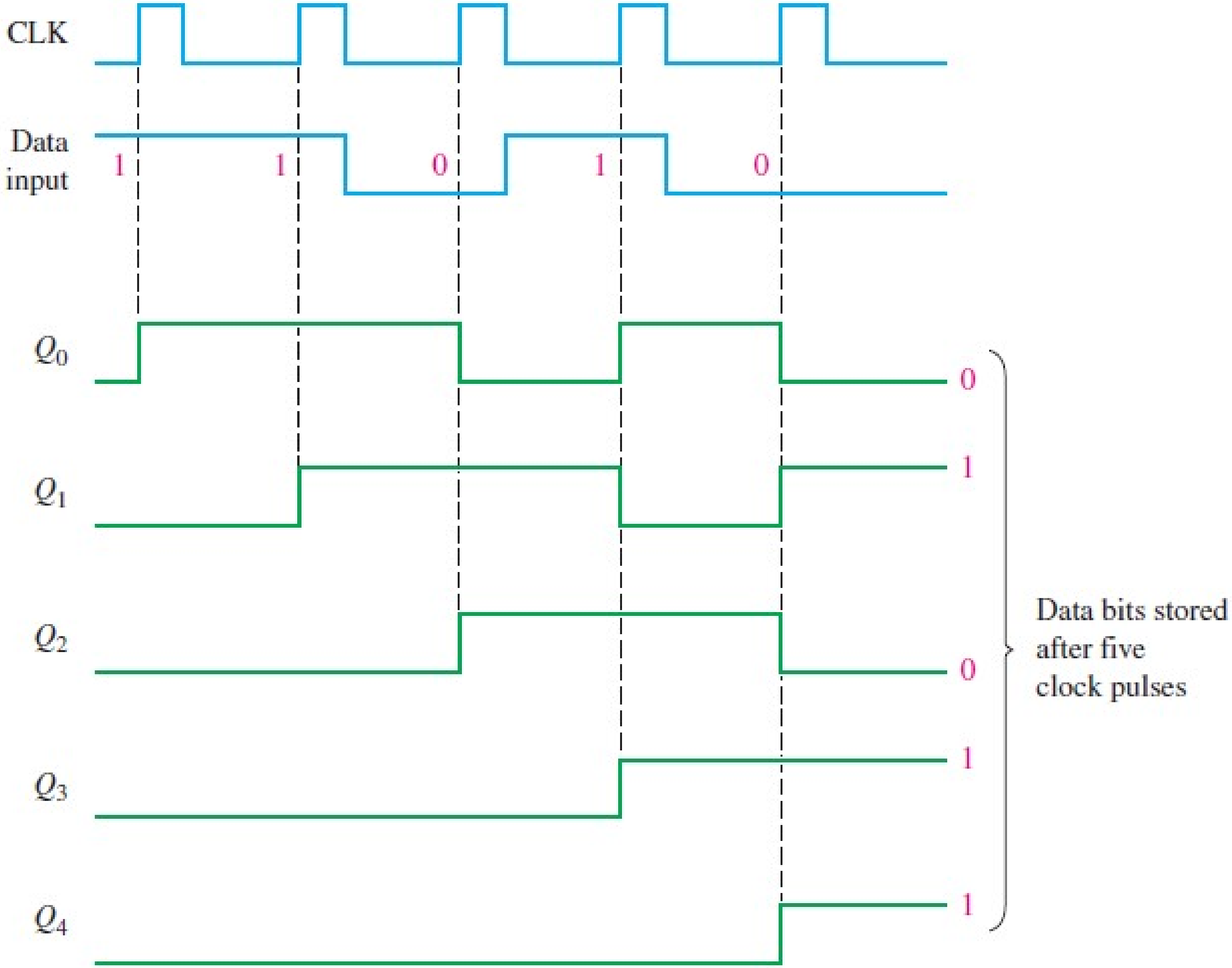


CLK	FF0 (Q_0)	FF1 (Q_1)	FF2 (Q_2)	FF3 (Q_3)
Initial	**1**	**0**	**1**	**0**
5	0	**1**	**0**	**1**
6	0	0	**1**	**0**
7	0	0	0	**1**
8	0	0	0	0

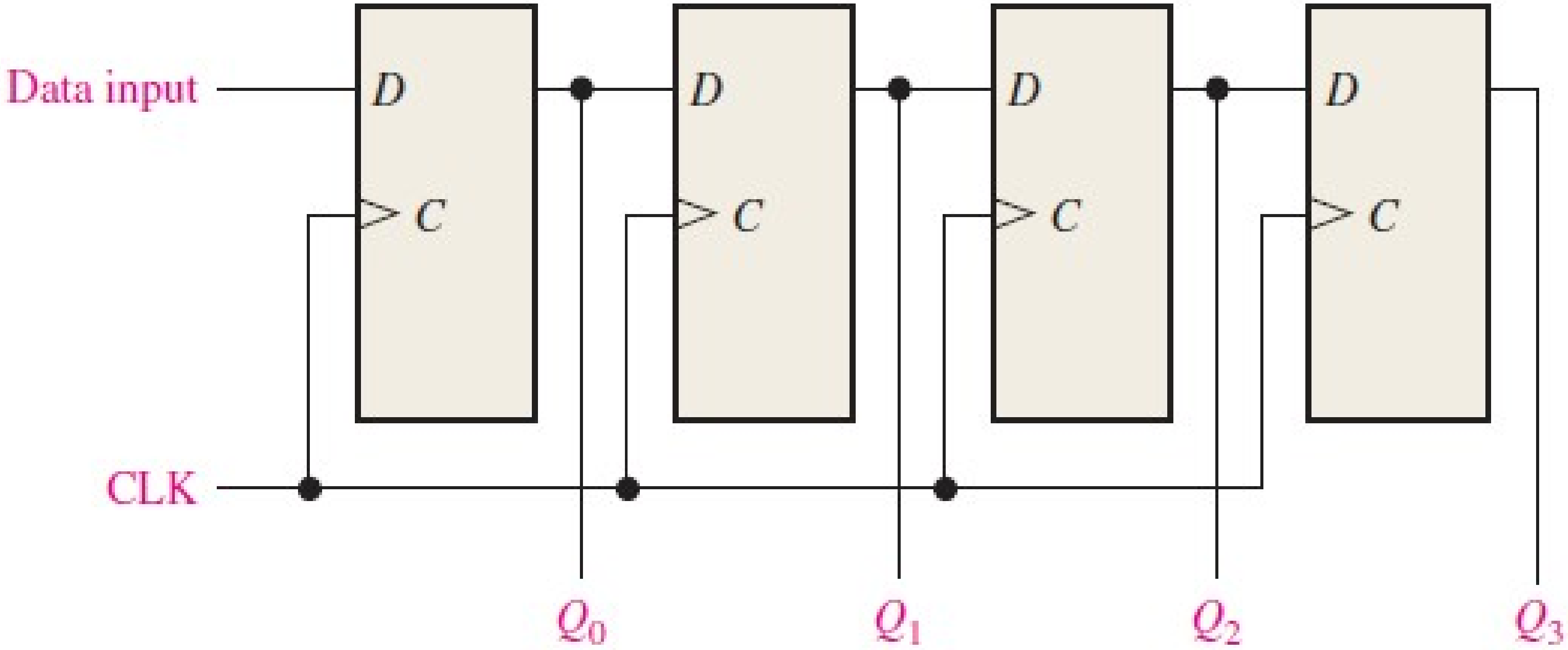
Concept Check



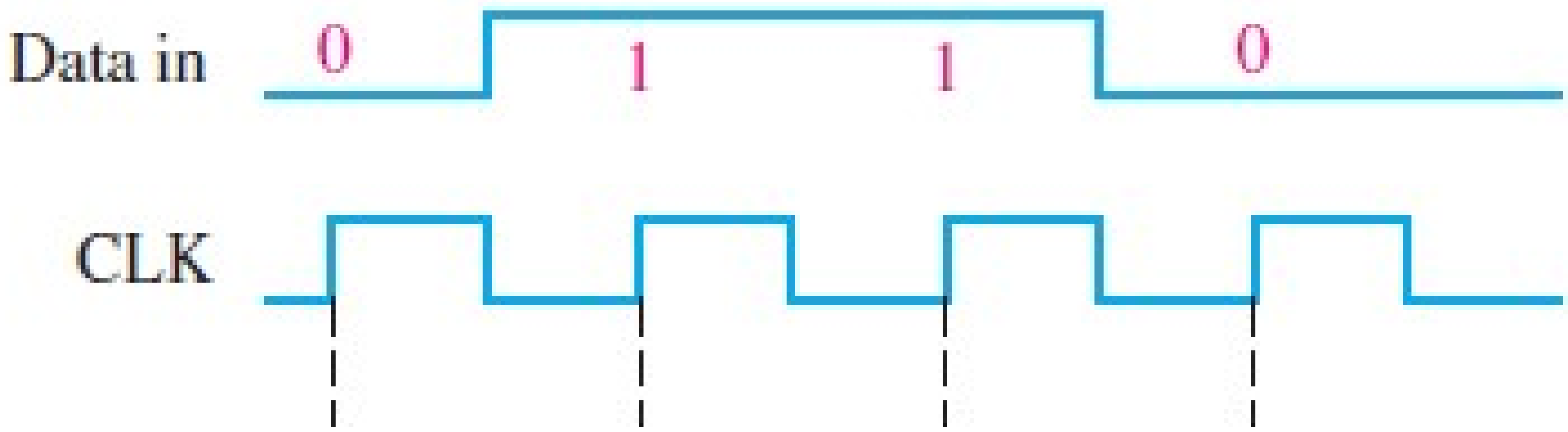
Solution


CLK
Data input
1
1
0
1
0
Q_0
Q_1
Q_2
Q_3
Q_4
0
1
0
1
1
Data bits stored after five clock pulses

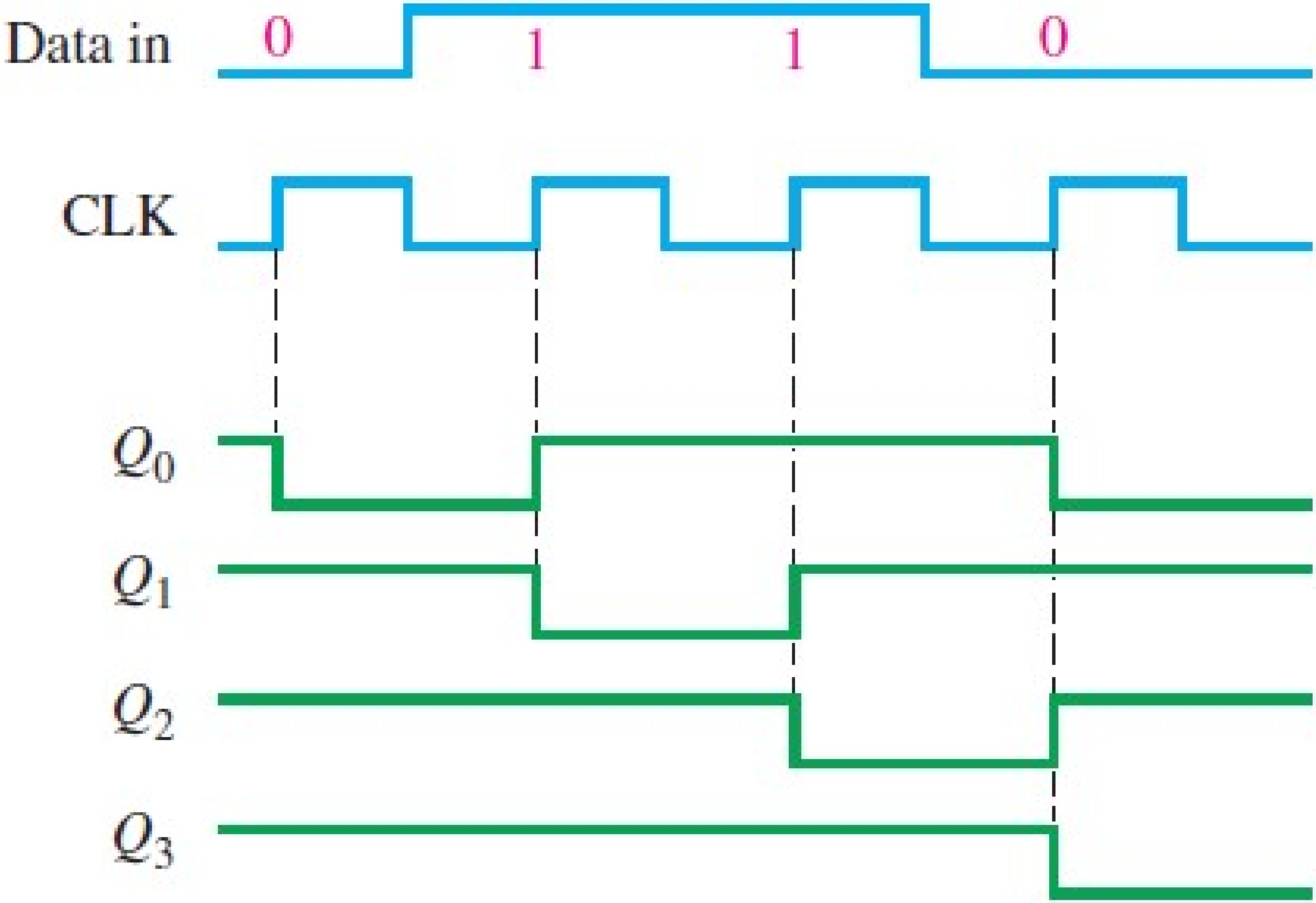
Serial In Parallel Out



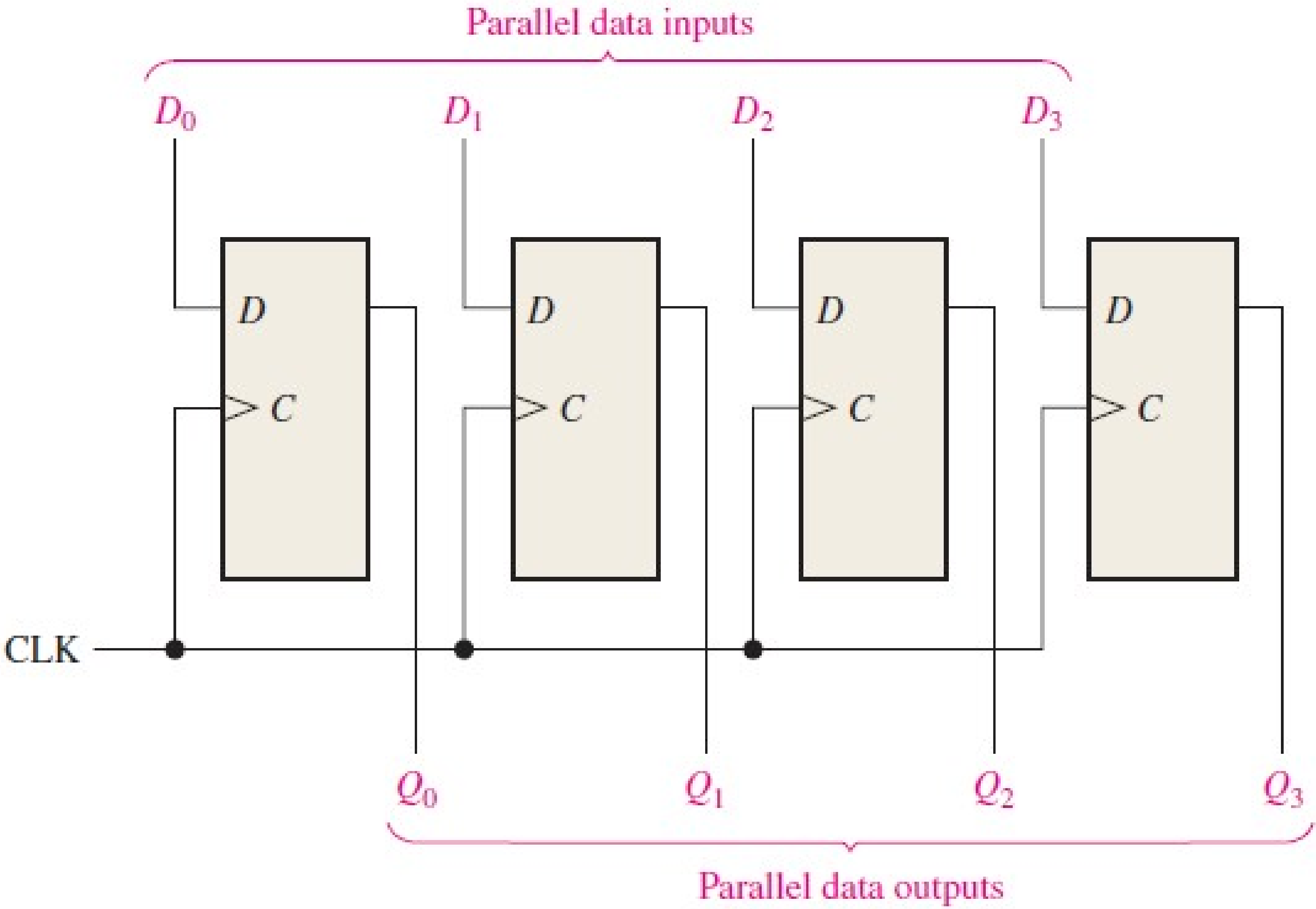
Concept Check



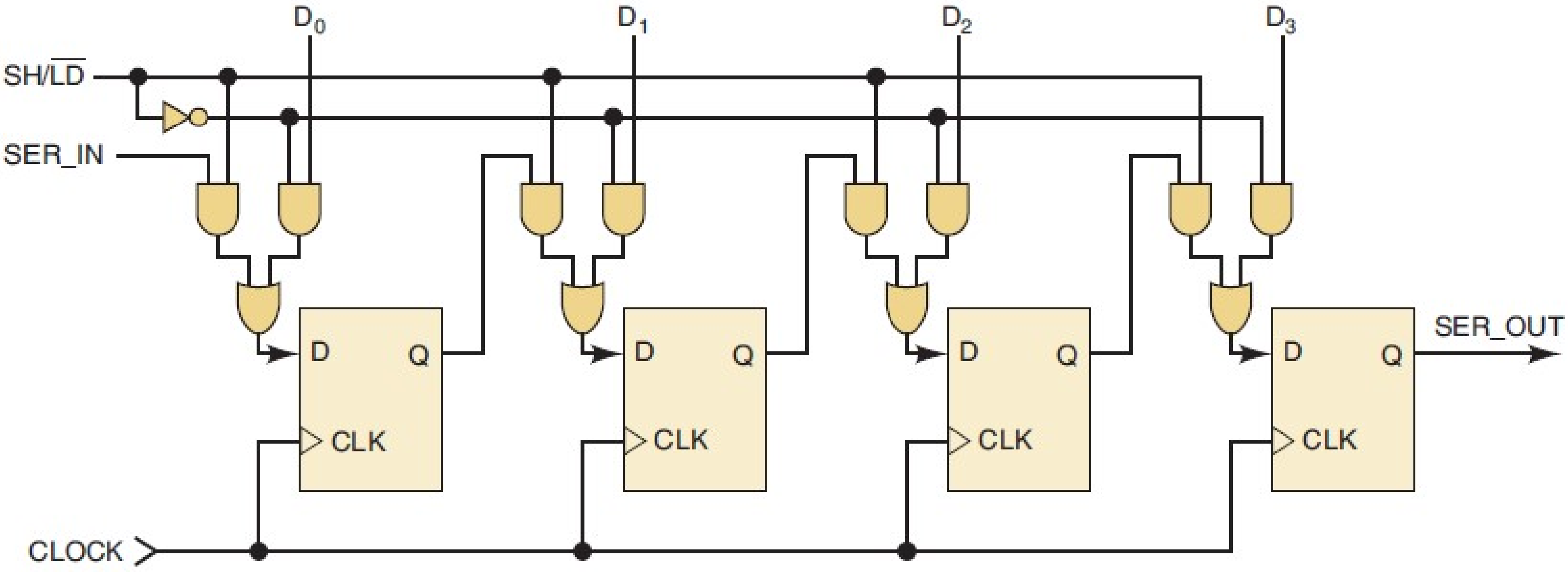
Solution



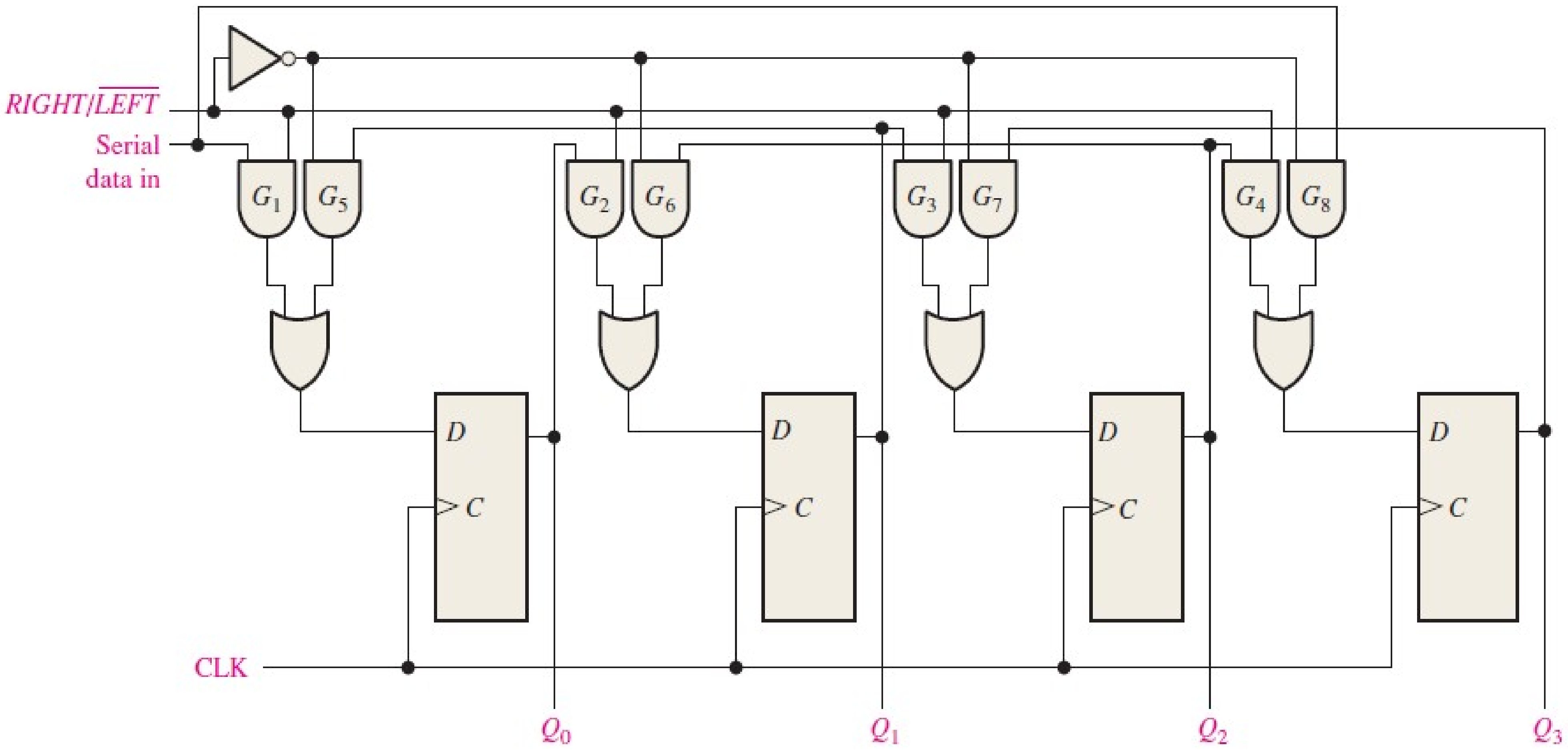
Parallel In Parallel Out



Parallel In Serial Out

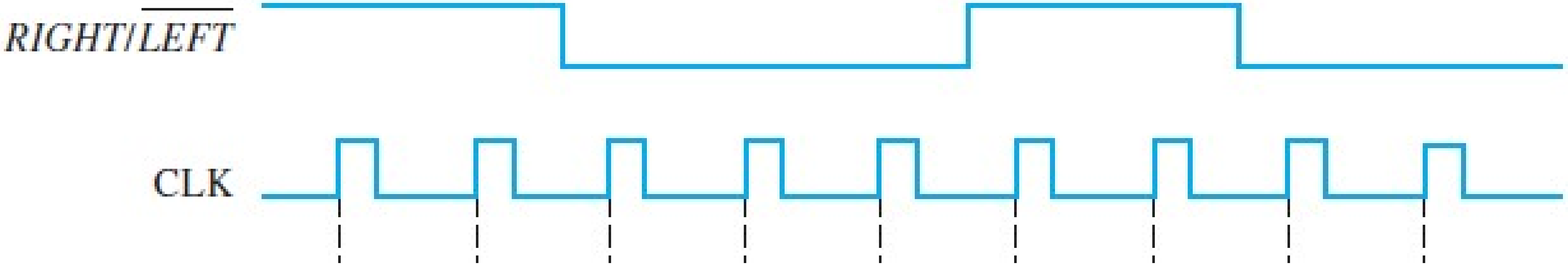


Bidirectional Shift Register

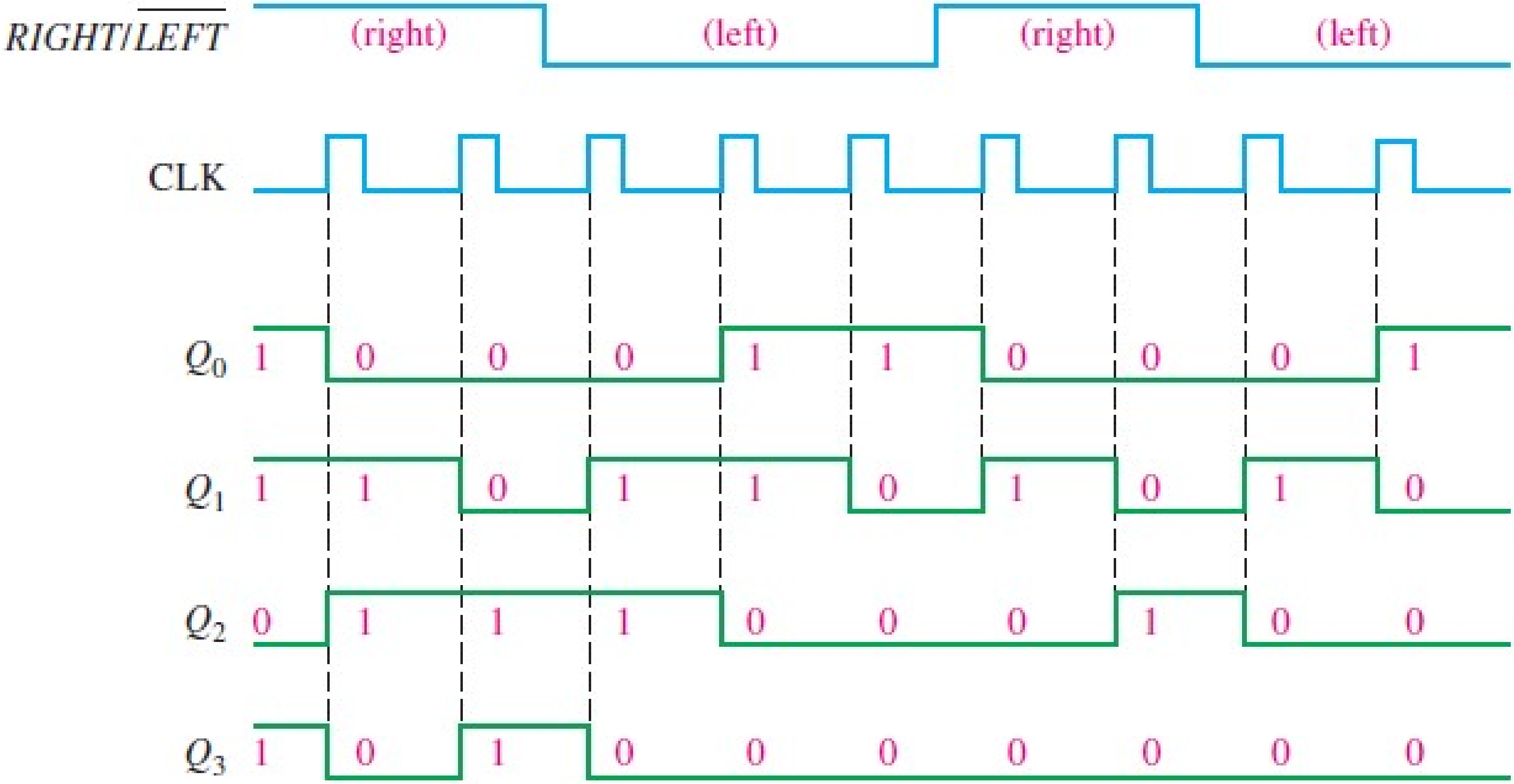


Concept Check

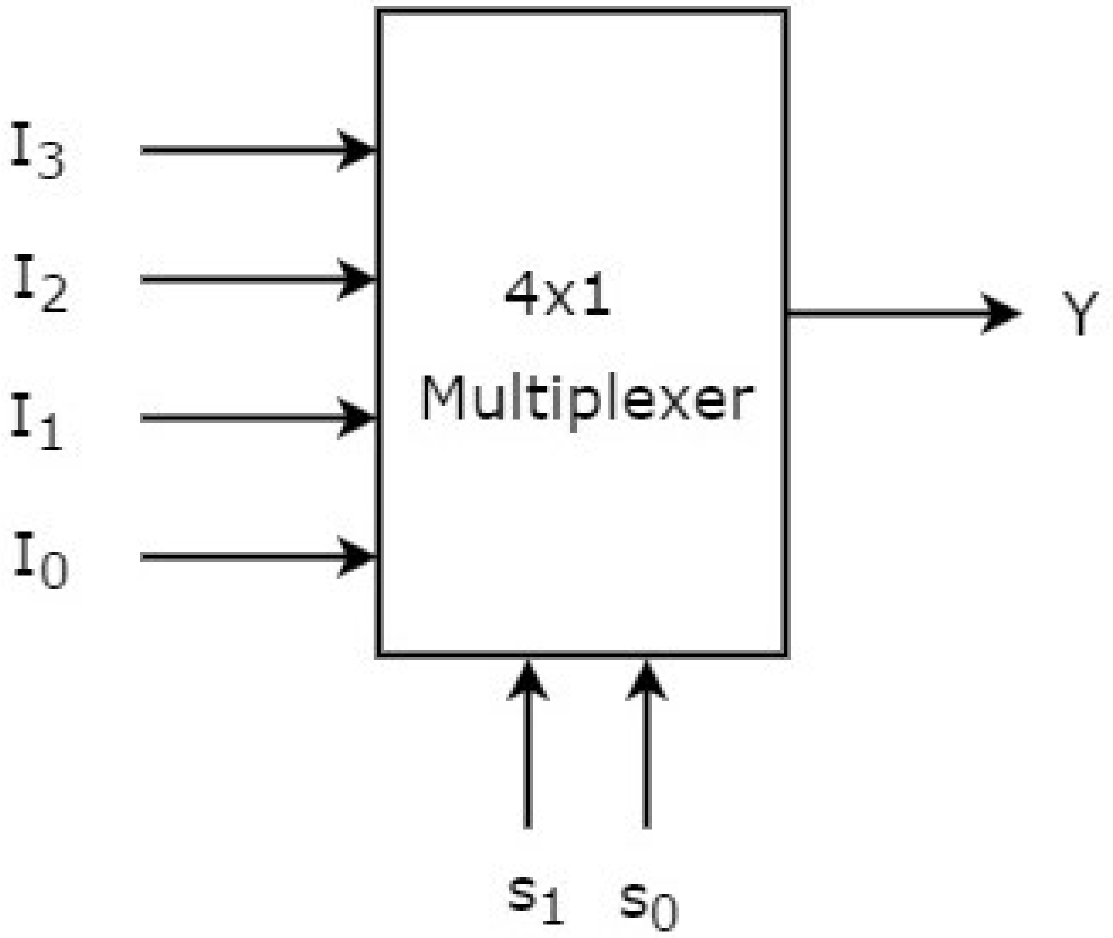
Assume that Q0 = 1, Q1 = 1, Q2 = 0, and Q3 = 1 and that the serial data-input line is LOW. Determine the state of the shift register



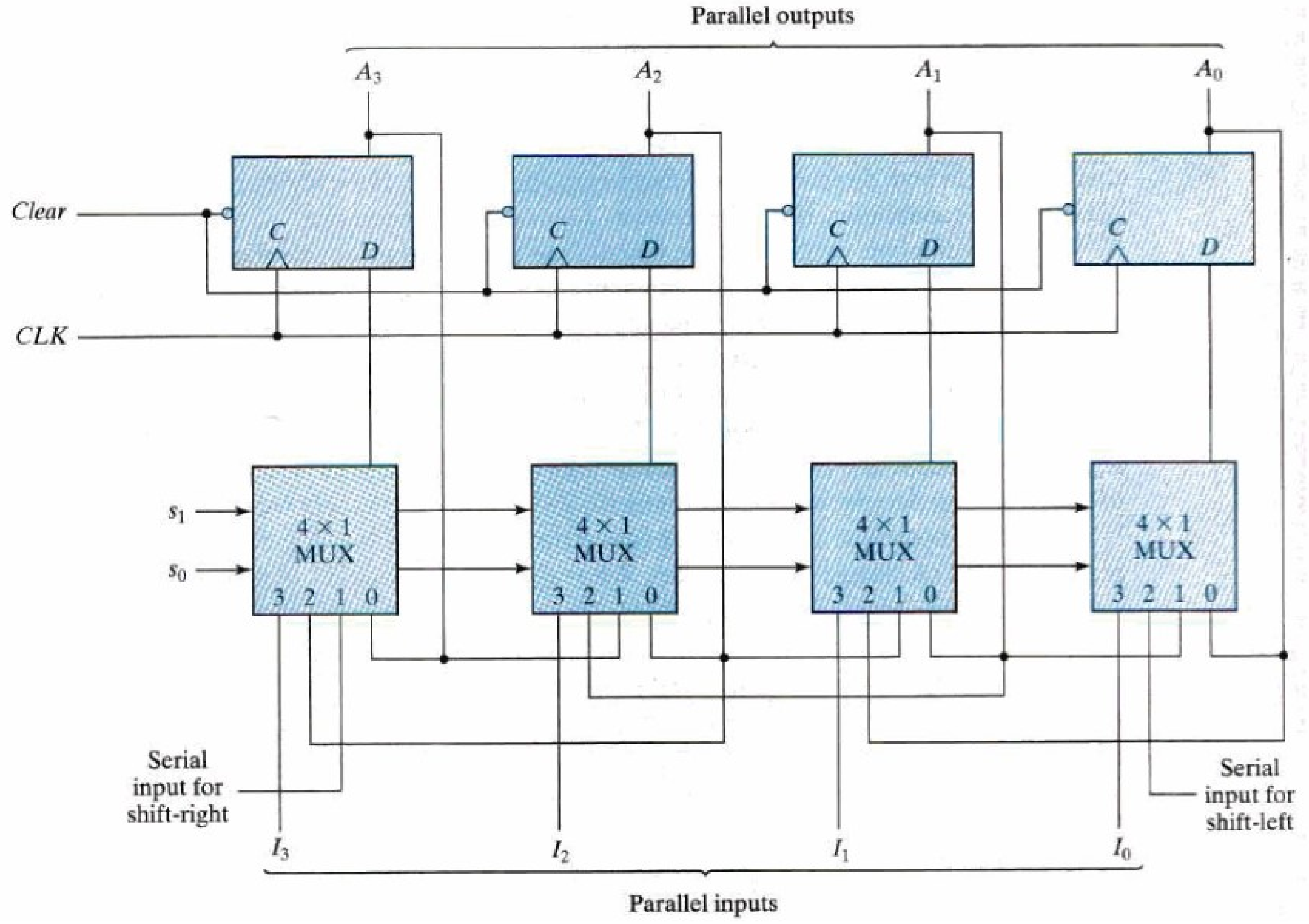
Solution


$RIGHT/\overline{LEFT}$
(right)
(left)
(right)
(left)
CLK
Q_0 1 0 0 0 1 1 0 0 0 1
Q_1 1 1 0 1 1 0 1 0 1 0
Q_2 0 1 1 1 0 0 0 1 0 0
Q_3 1 0 1 0 0 0 0 0 0 0

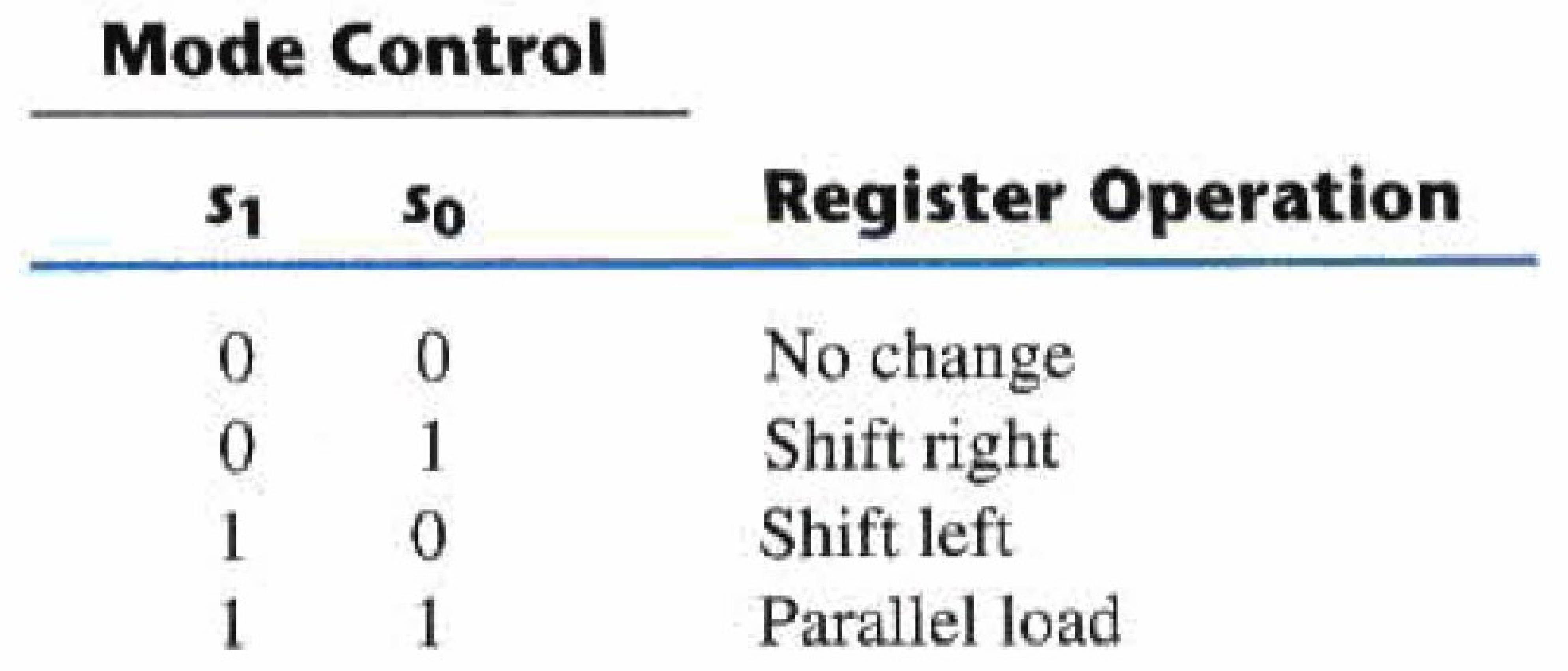
Multiplexer



Universal Shift Register

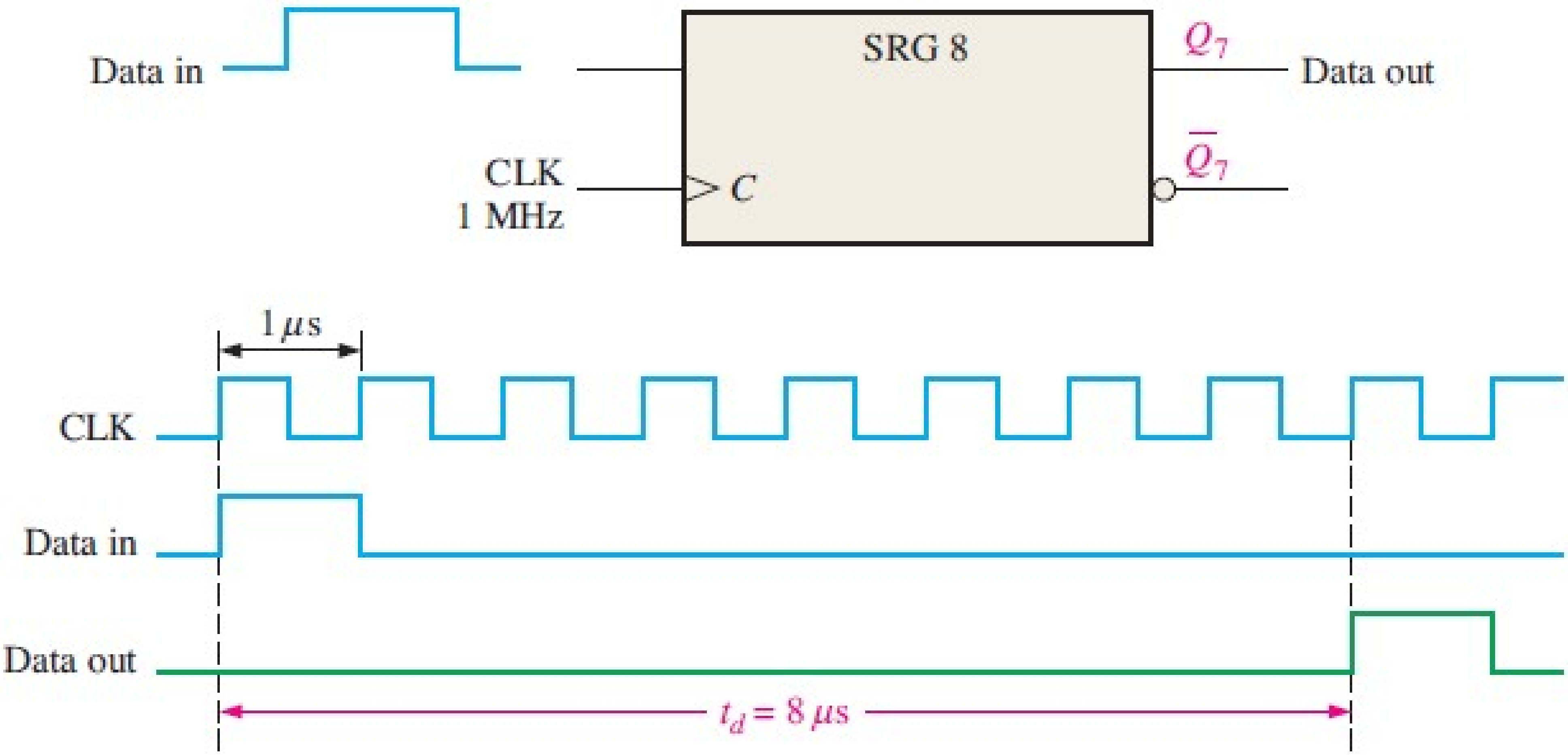


Modes (Universal Shift Register)

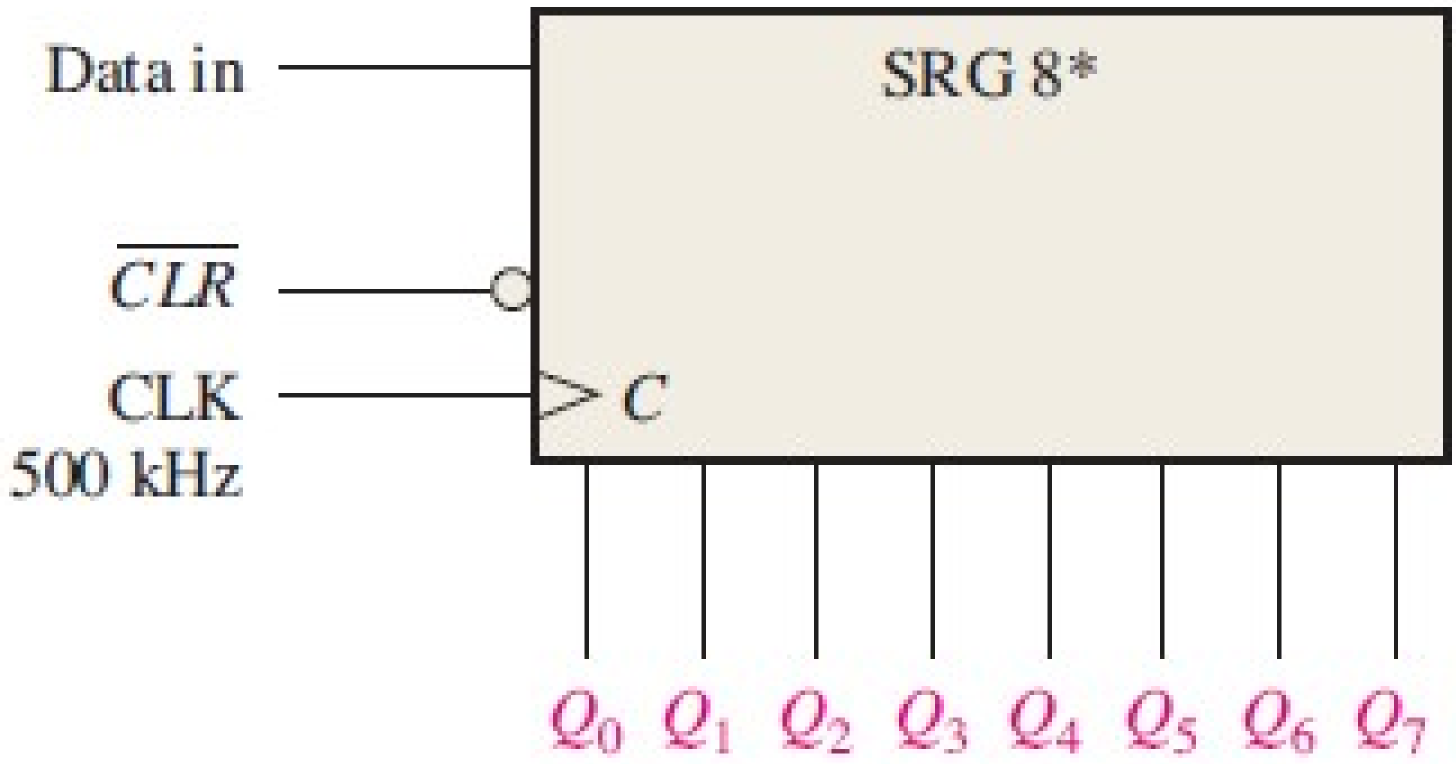


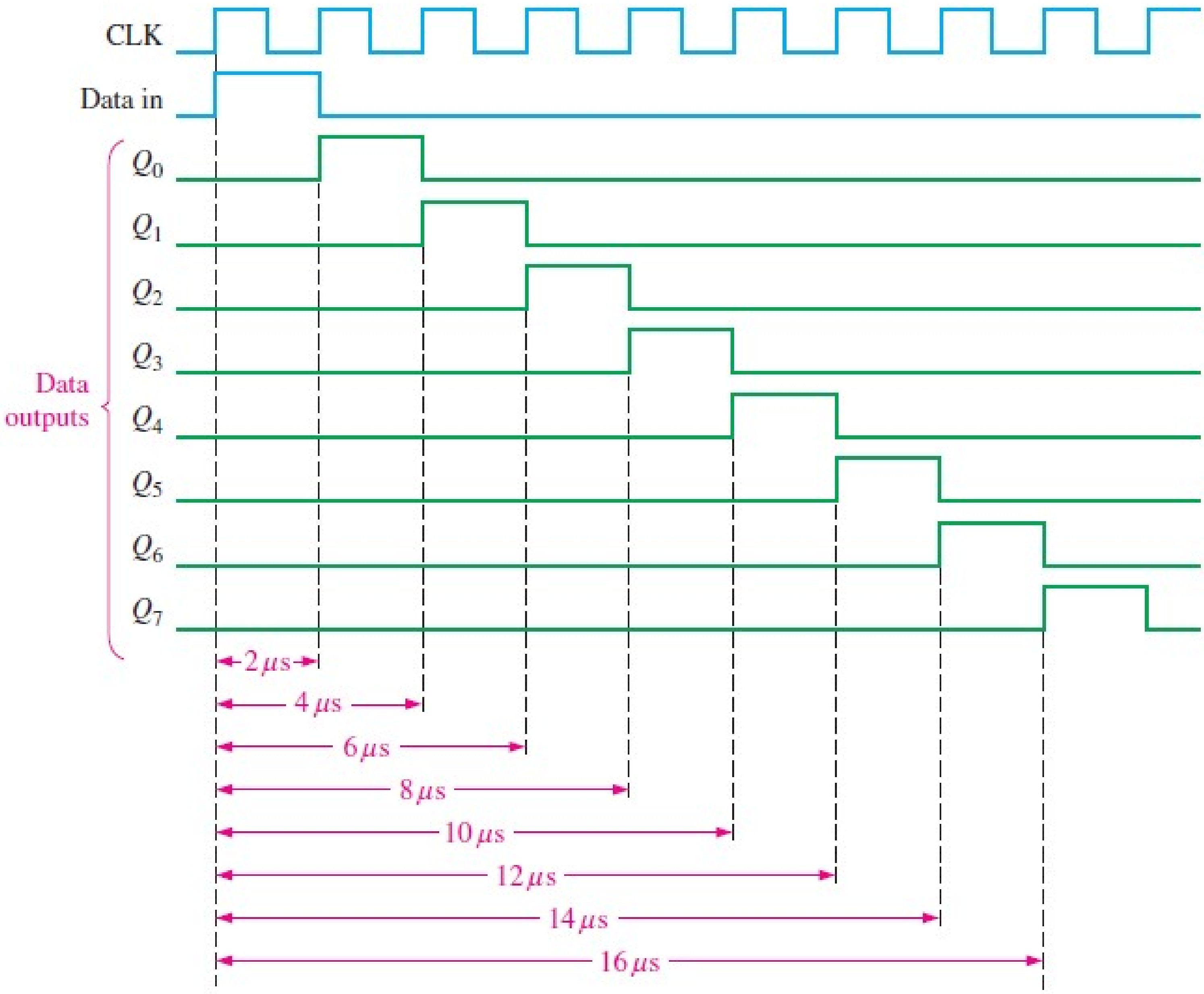
Mode Control		
s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Register For Time Delay



Concept Check




CLK
Data in
Q_0
Q_1
Q_2
Q_3
Data outputs
Q_4
Q_5
Q_6
Q_7
2 μs
4 μs
6 μs
8 μs
10 μs
12 μs
14 μs
16 μs

Counters

Ripple/Asynchronous/Serial Counters

❑ Cascaded arrangement of Flip-Flops where the output of one Flip-Flop drives the input of the following Flip-Flop.

❑ The clock input to any subsequent flip-flop comes from the output of its immediately preceding flip-flop. For instance, the output of the first flip-flop acts as the clock input to the second flip-flop, the output of the second flip-flop feeds the clock input of the third flip-flop and so on.

❑ The modulus (MOD number) of a counter is the number of unique states it goes through before it comes back to the initial state to repeat the count sequence.

❑ An n-bit counter that counts through all its natural states and does not skip any of the states has a modulus of 2n.

❑ We can see that such counters have a modulus that is an integral power of 2, that is, 2, 4, 8, 16 and so on.

❑ These can be modified with the help of additional combinational logic to get a modulus of less than 2n.

Modulus

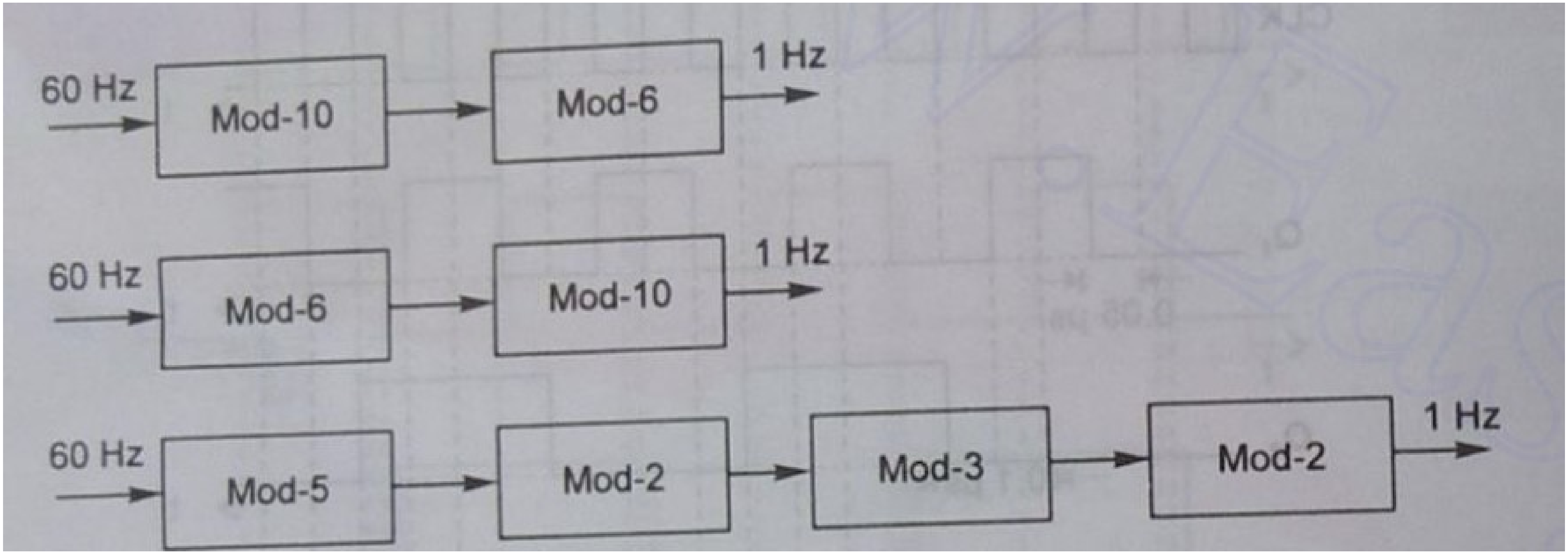
- To determine the number of flip-flops required to build a counter having a given modulus, identify the smallest integer m that is either equal to or greater than the desired modulus and is also equal to an integral power of 2. For instance, if the desired modulus is 10, which is the case in a decade counter, the smallest integer greater than or equal to 10 and which is also an integral power of 2 is 16. The number of flip-flops in this case would be 4, as 16 = 24.

$$modulus \leq 2^N$$

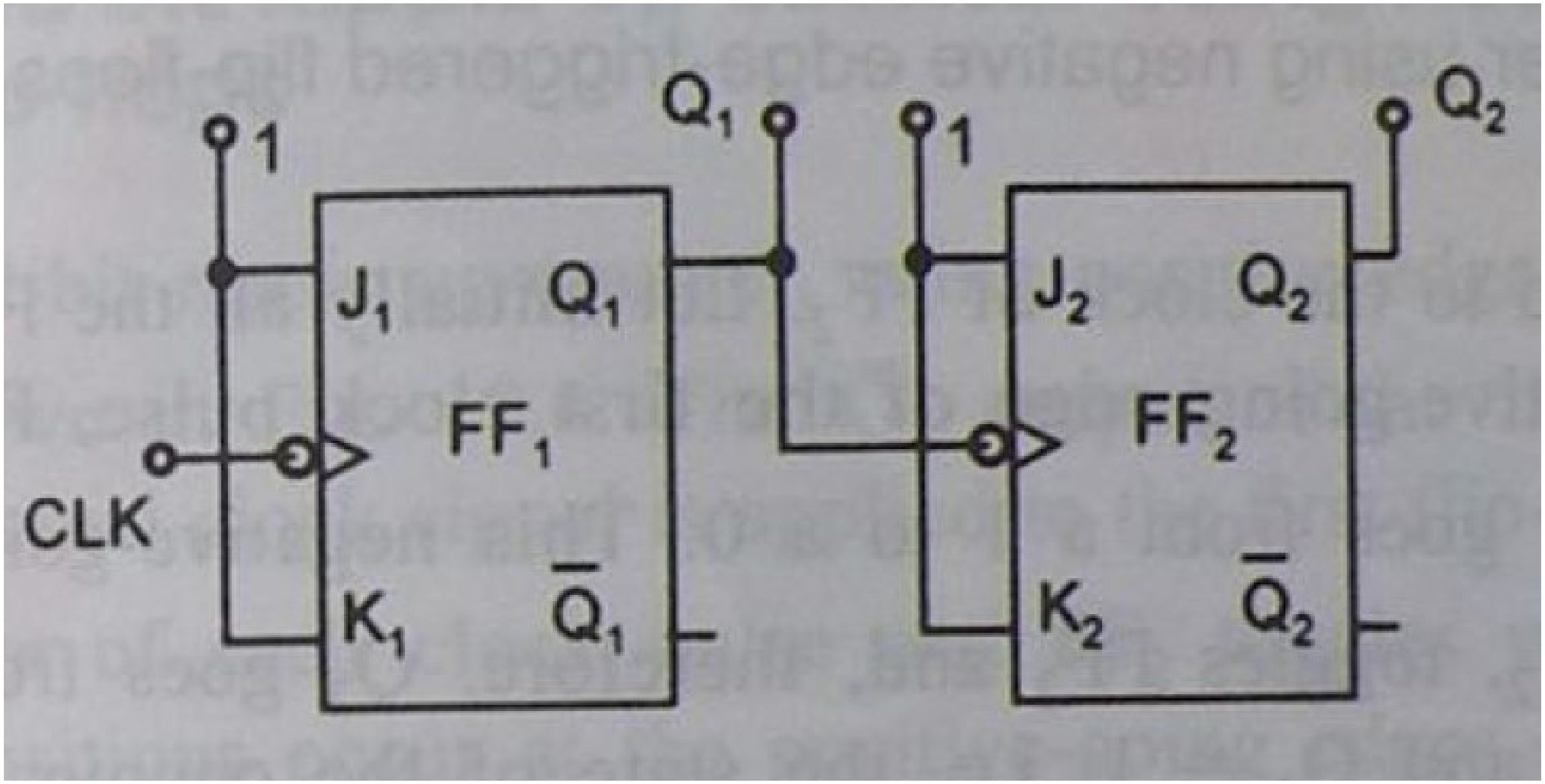
Concept Check

It is desired to design a binary ripple counter that is capable of counting the number of items passing on a conveyor belt. Each time an item passes a given point, a pulse is generated that can be used as a clock input. If the maximum number of items to be counted is 6000, determine the number of flip-flops required.

Concept of Frequency Division

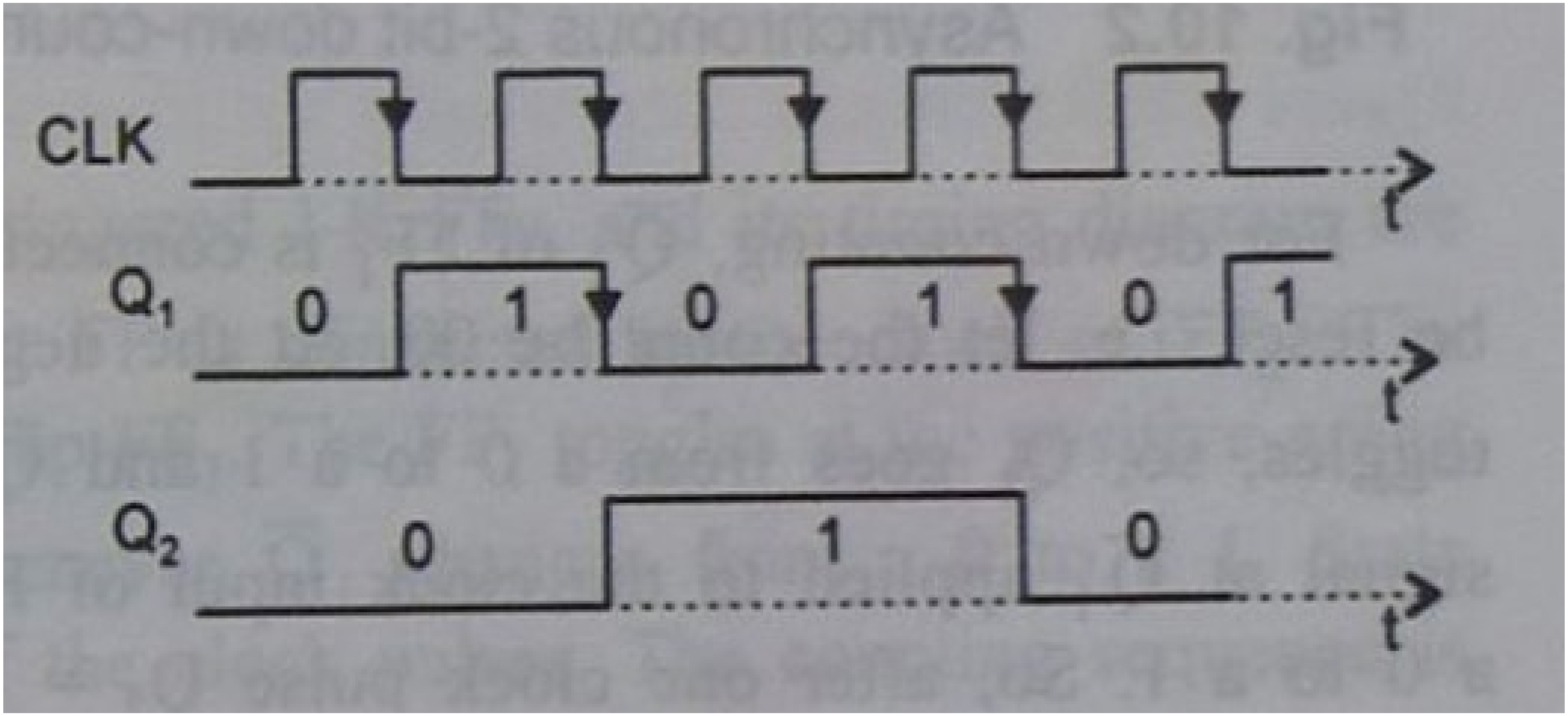


2 Bit NET Up Counter



2 Bit NET Up Counter

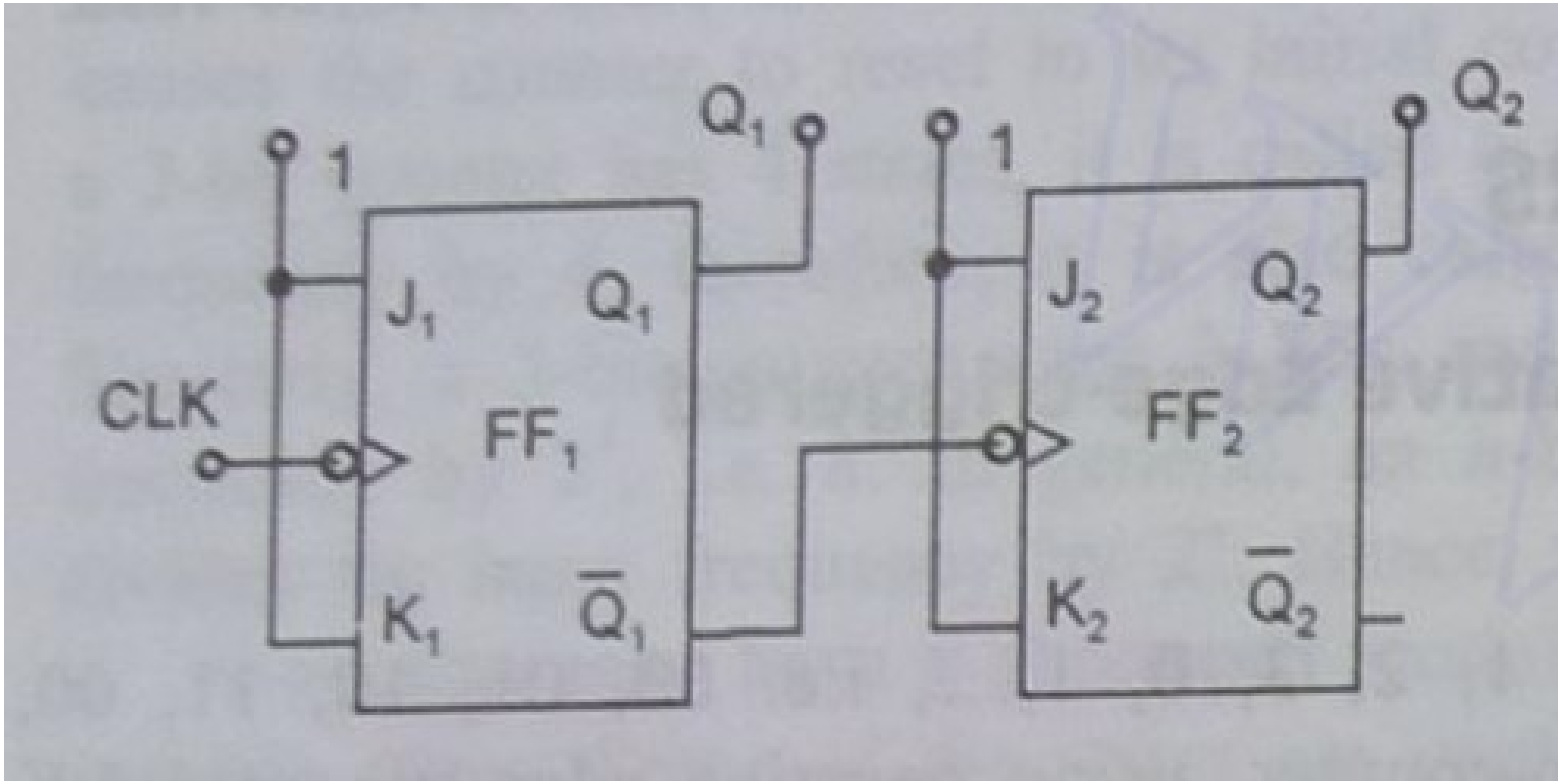
Clock Pulse	Q2	Q1
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0


CLK
t
Q_1
0
1
0
1
0
1
t
Q_2
0
1
0
t

3 – Bit NET Up Counter

Clock Pulse No.	Q3	Q2	Q1
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
8	0	0	0

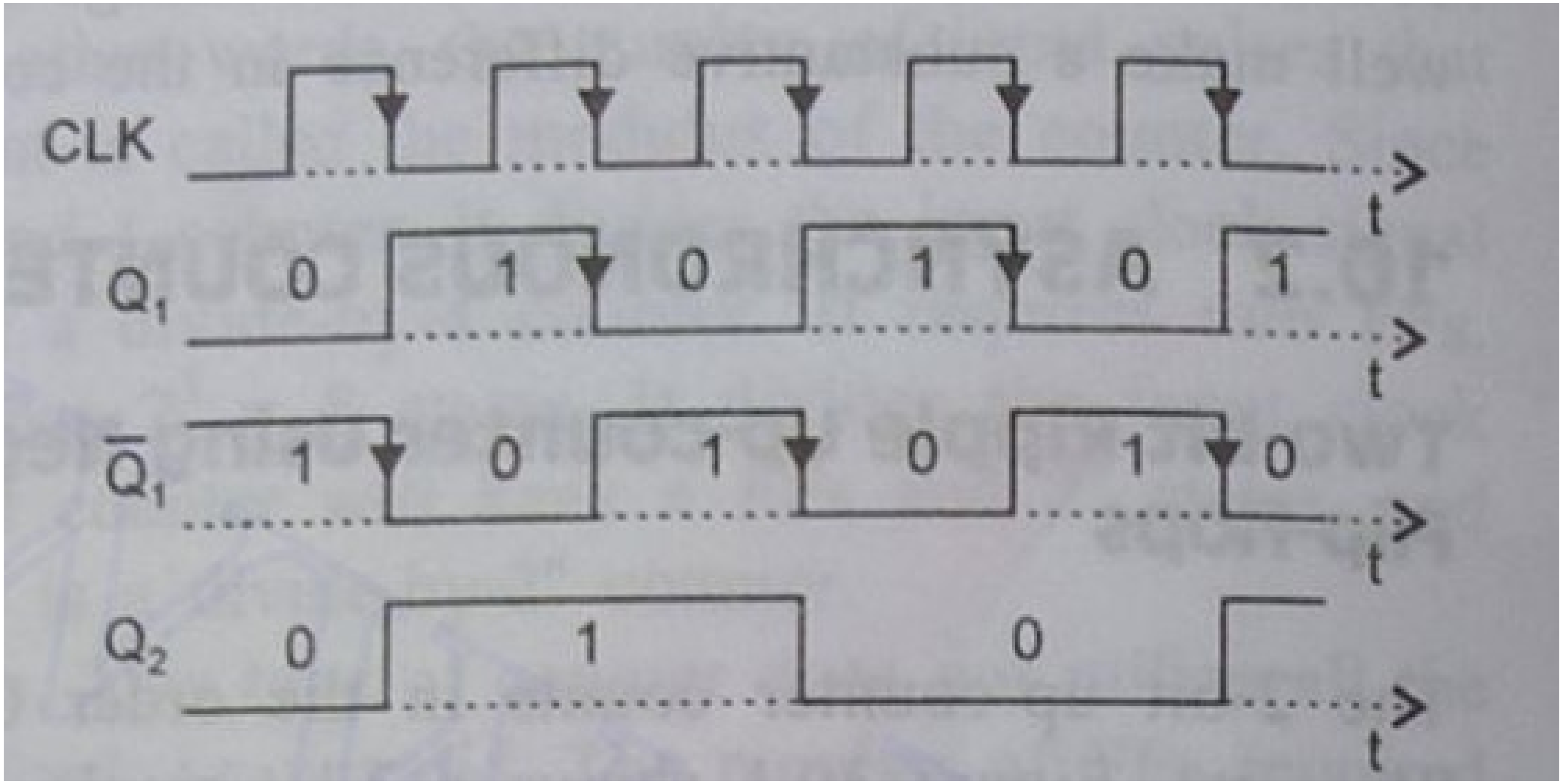
2 – Bit NET Down Counter



2 – Bit NET Down Counter

Clock Pulse	Q2	Q1	Q1’
0	0	0	1
1	1	1	0
2	1	0	1
3	0	1	0
4	0	0	1

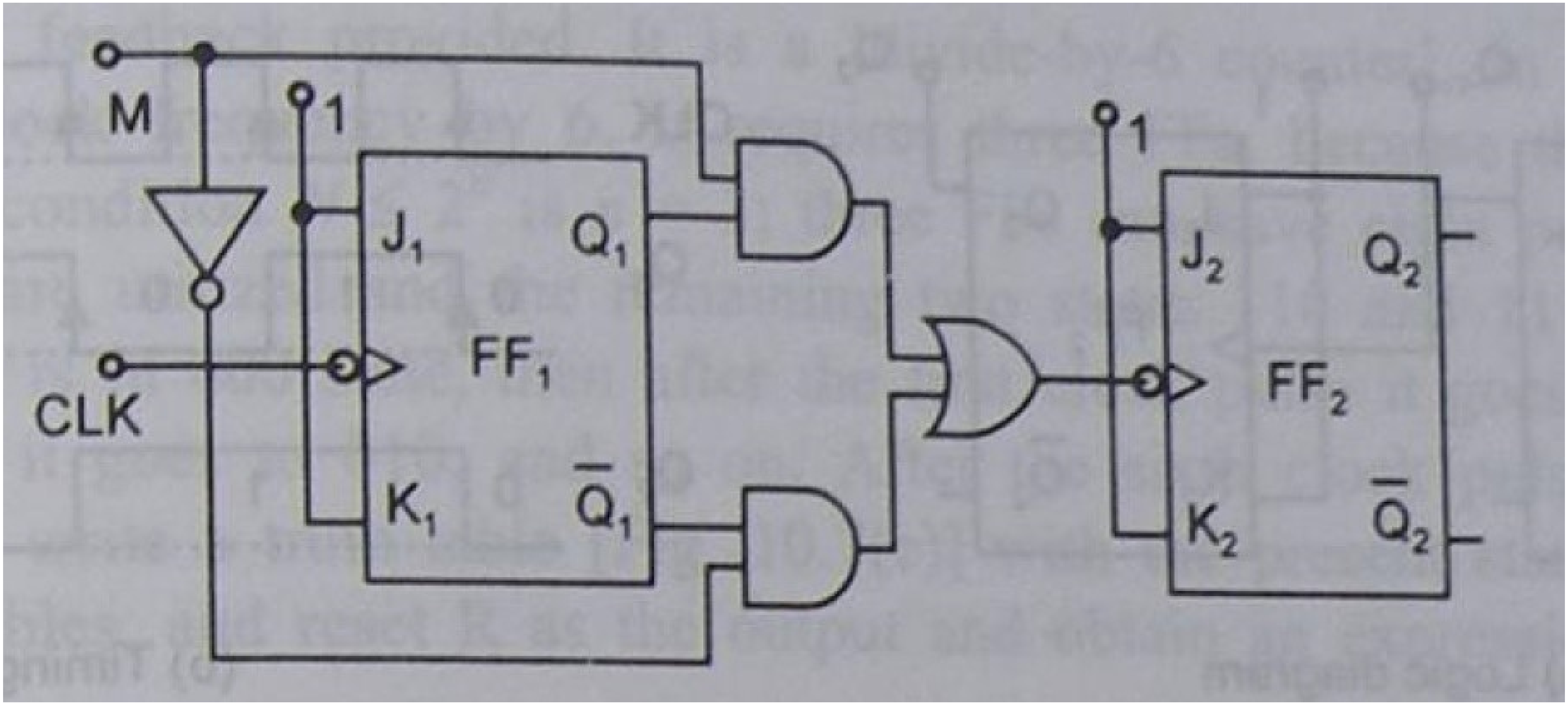
2 – Bit NET Down Counter



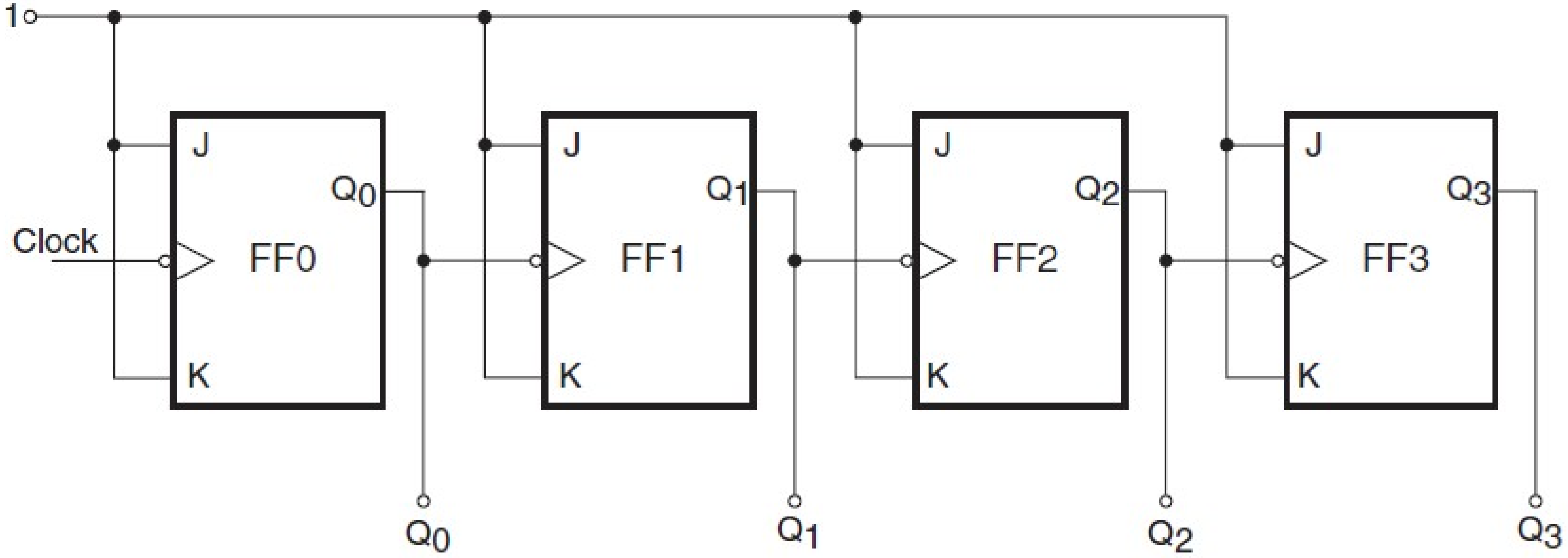
3 – Bit NET Down Counter

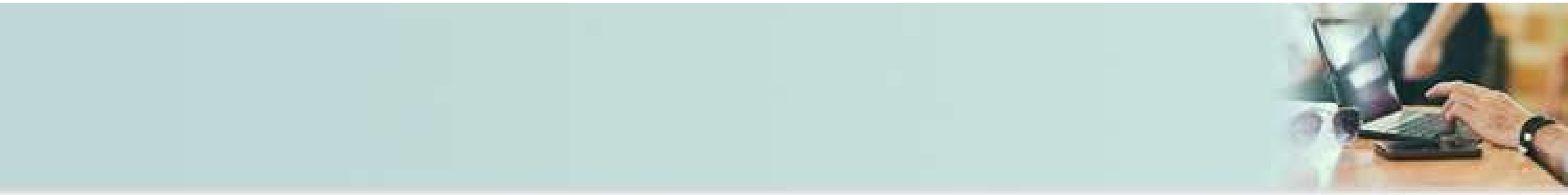
Clock Pulse No.	Q3	Q2	Q2′	Q1	Q1′
0	0	0	1	0	1
1	1	1	0	1	0
2	1	1	0	0	1
3	1	0	1	1	0
4	1	0	1	0	1
5	0	1	0	1	0
6	0	1	0	0	1
7	0	0	1	1	0
8	0	0	1	0	1

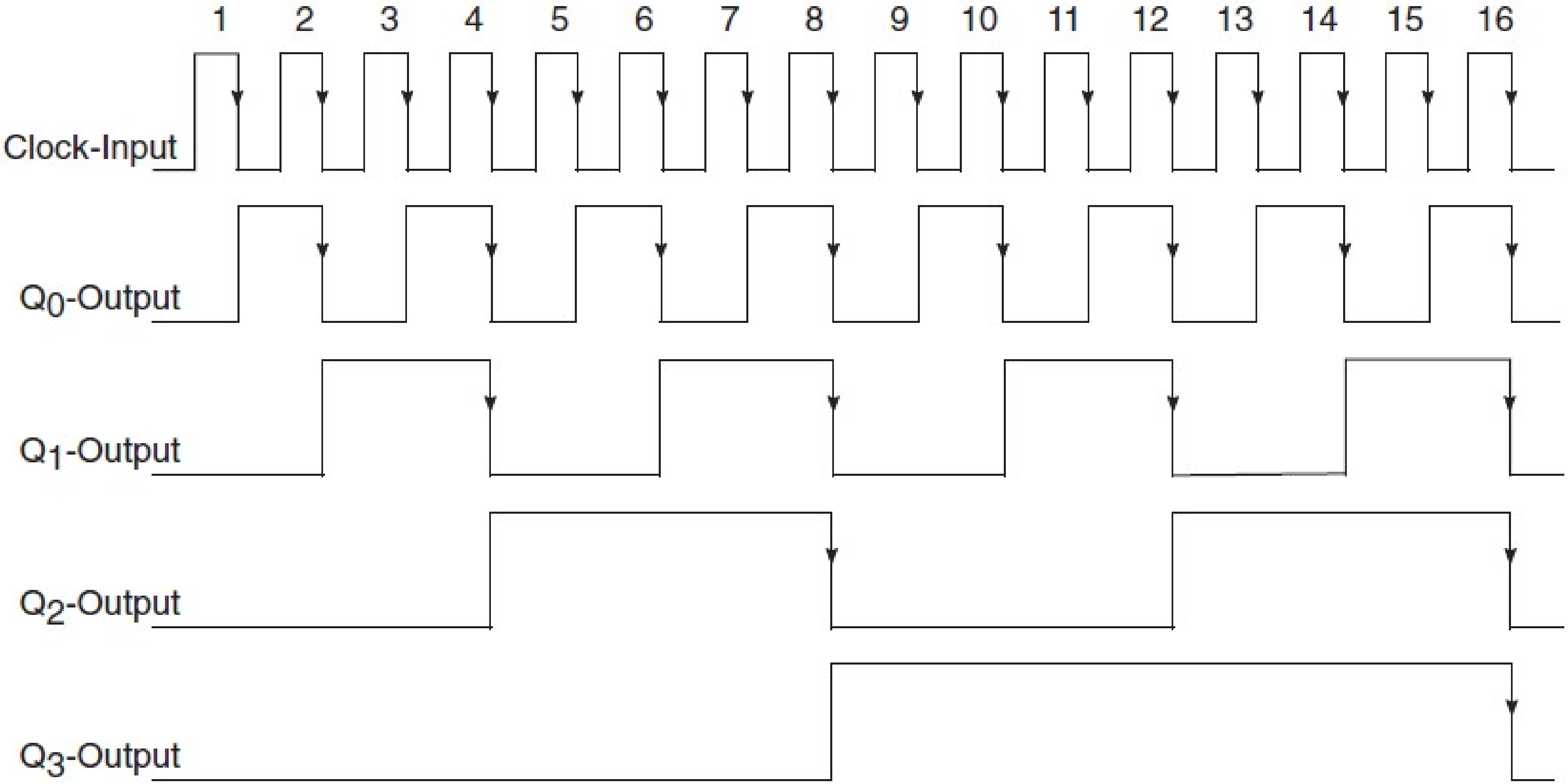
2-Bit NET UP/Down Counter



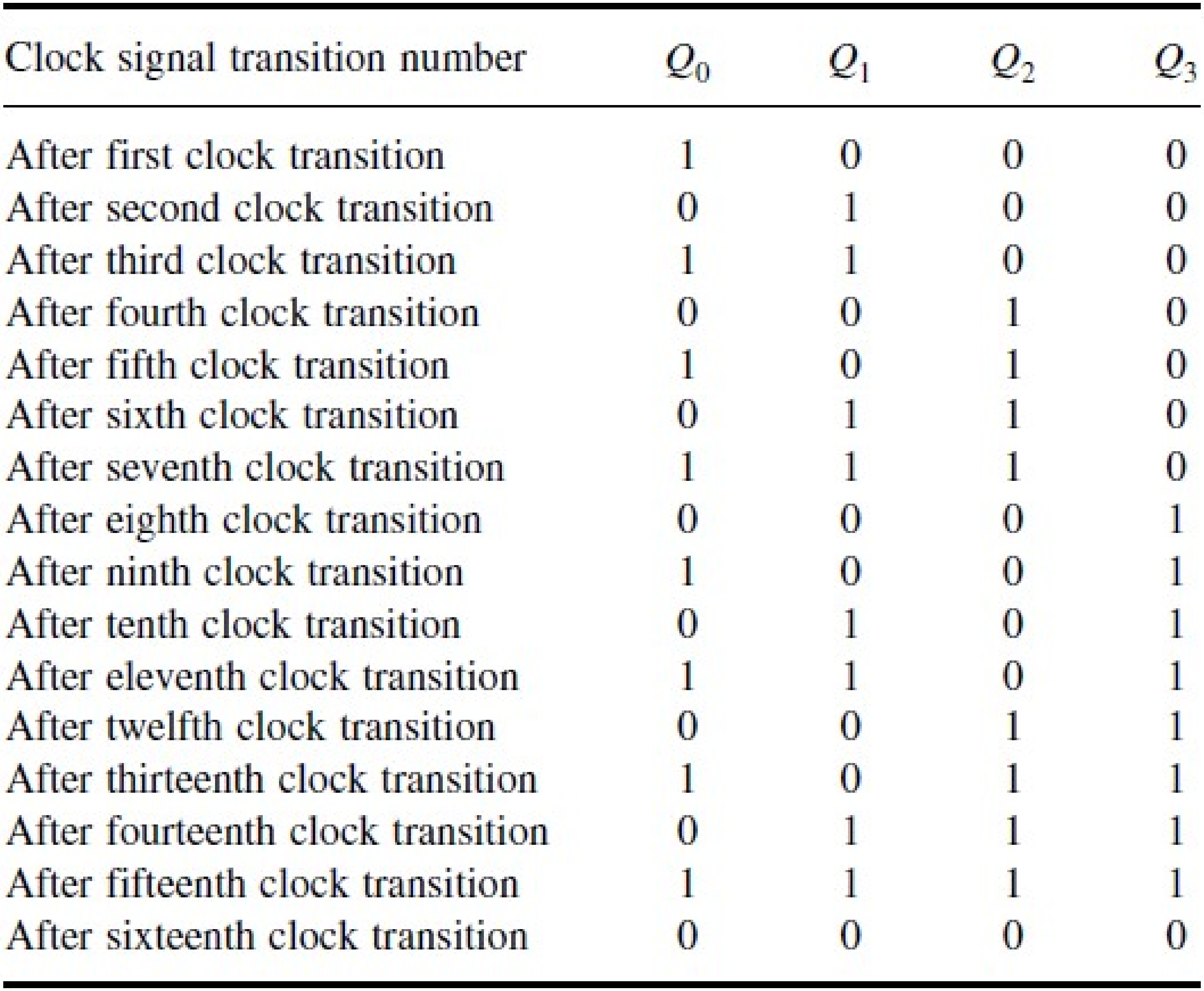
Predict the Output States





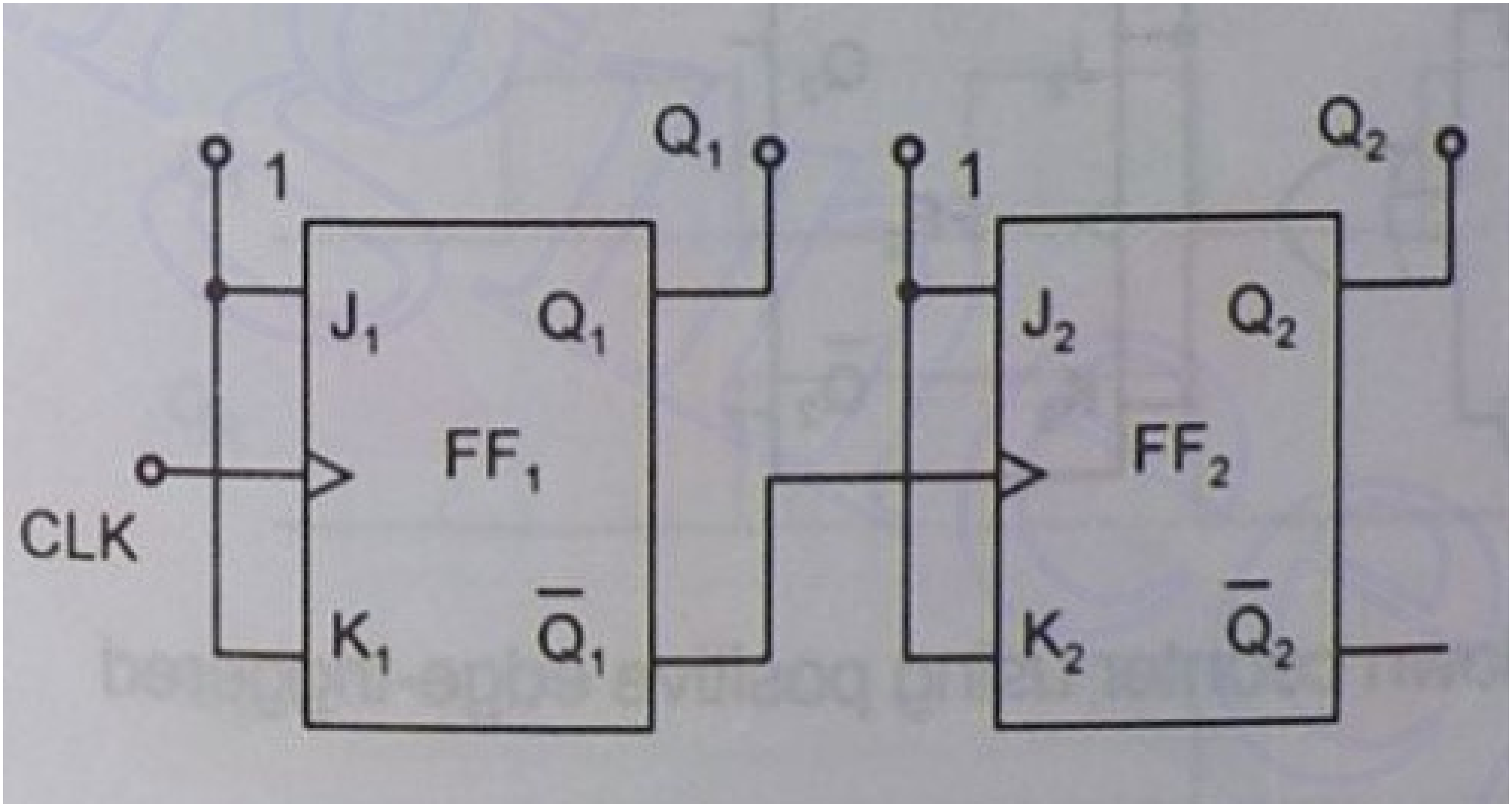

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Clock-Input
Q0-Output
Q1-Output
Q2-Output
Q3-Output

Truth Table / Output States



Clock signal transition number	Q_0	Q_1	Q_2	Q_3
After first clock transition	1	0	0	0
After second clock transition	0	1	0	0
After third clock transition	1	1	0	0
After fourth clock transition	0	0	1	0
After fifth clock transition	1	0	1	0
After sixth clock transition	0	1	1	0
After seventh clock transition	1	1	1	0
After eighth clock transition	0	0	0	1
After ninth clock transition	1	0	0	1
After tenth clock transition	0	1	0	1
After eleventh clock transition	1	1	0	1
After twelfth clock transition	0	0	1	1
After thirteenth clock transition	1	0	1	1
After fourteenth clock transition	0	1	1	1
After fifteenth clock transition	1	1	1	1
After sixteenth clock transition	0	0	0	0

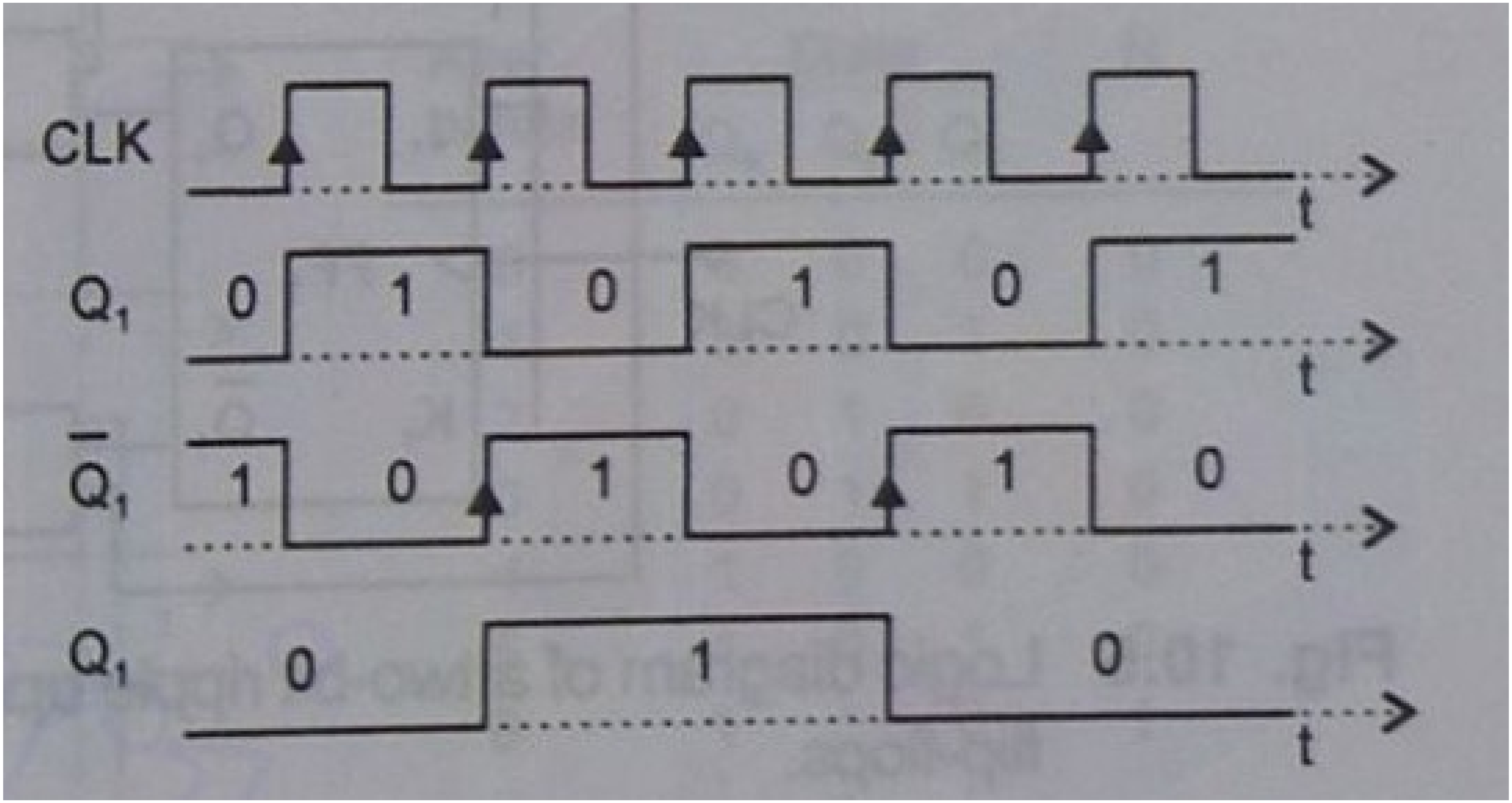
2 – Bit PET Up Counter



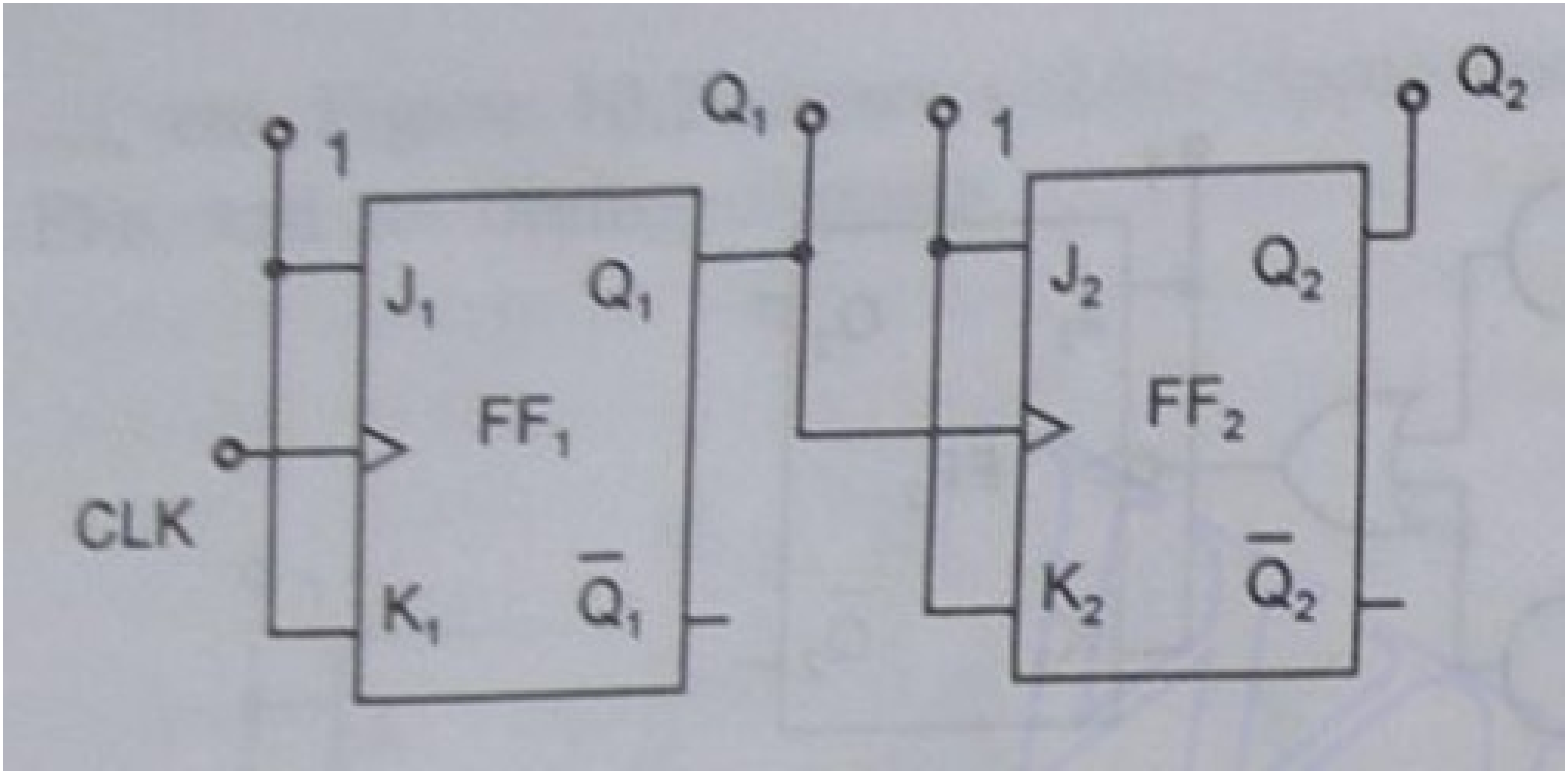
2 – Bit PET Up Counter

Clock Pulse	Q2	Q1	Q1’
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0
4	0	0	1

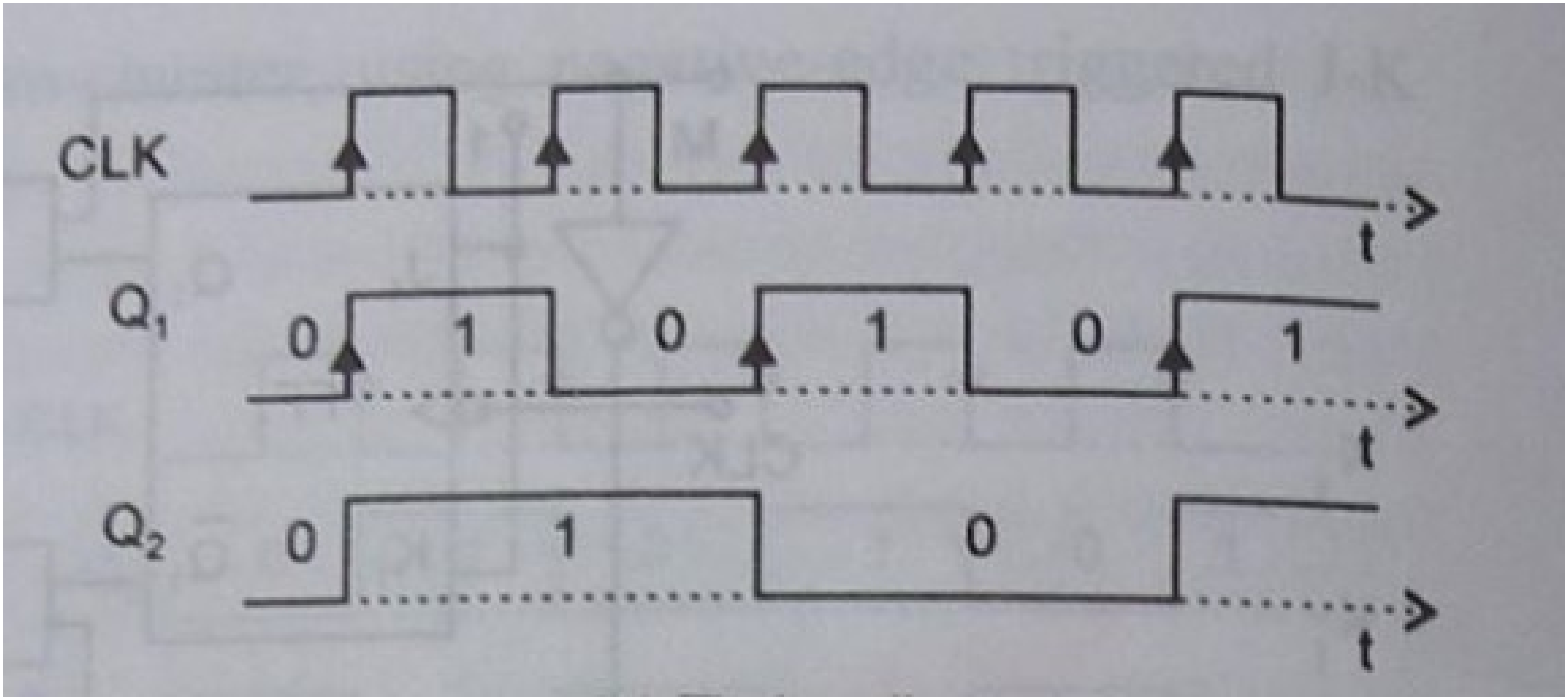
2 – Bit PET Up Counter



2 – Bit PET Down Counter



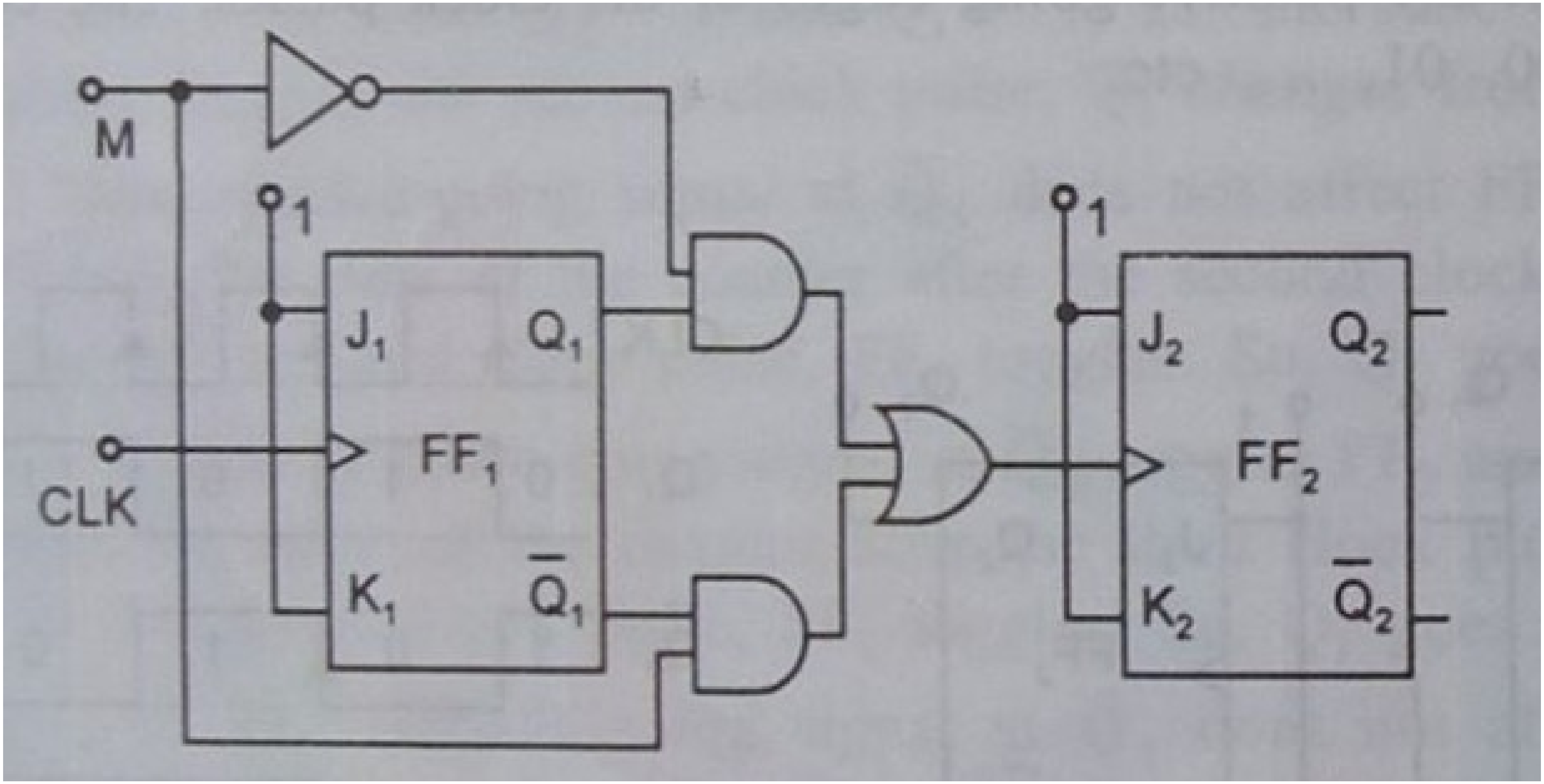
2 – Bit PET Down Counter



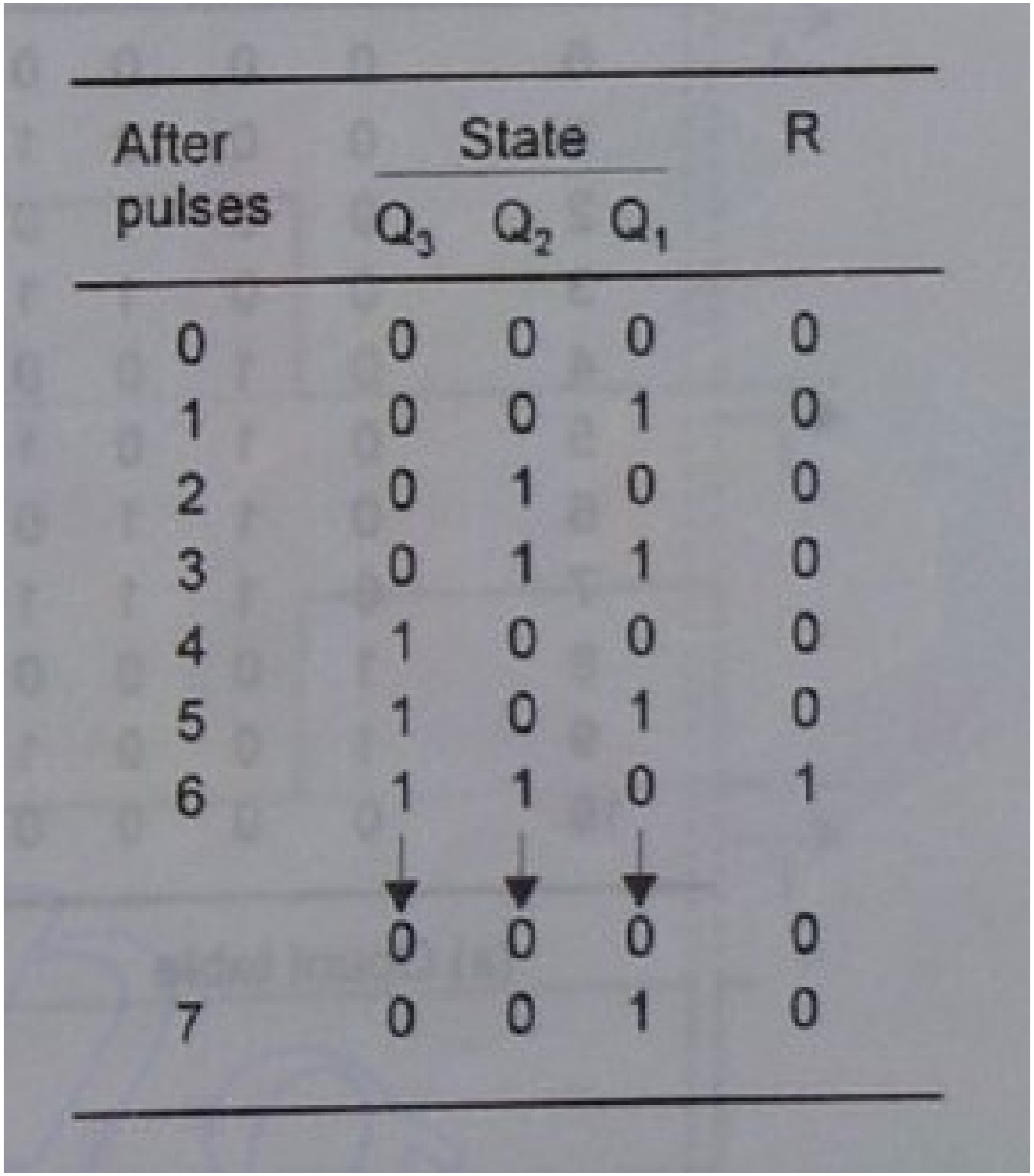
2 – Bit PET Down Counter

Clock Pulse	Q2	Q1
0	0	0
1	1	1
2	1	0
3	0	1
4	0	0

2 – Bit PET Up/Down Counter

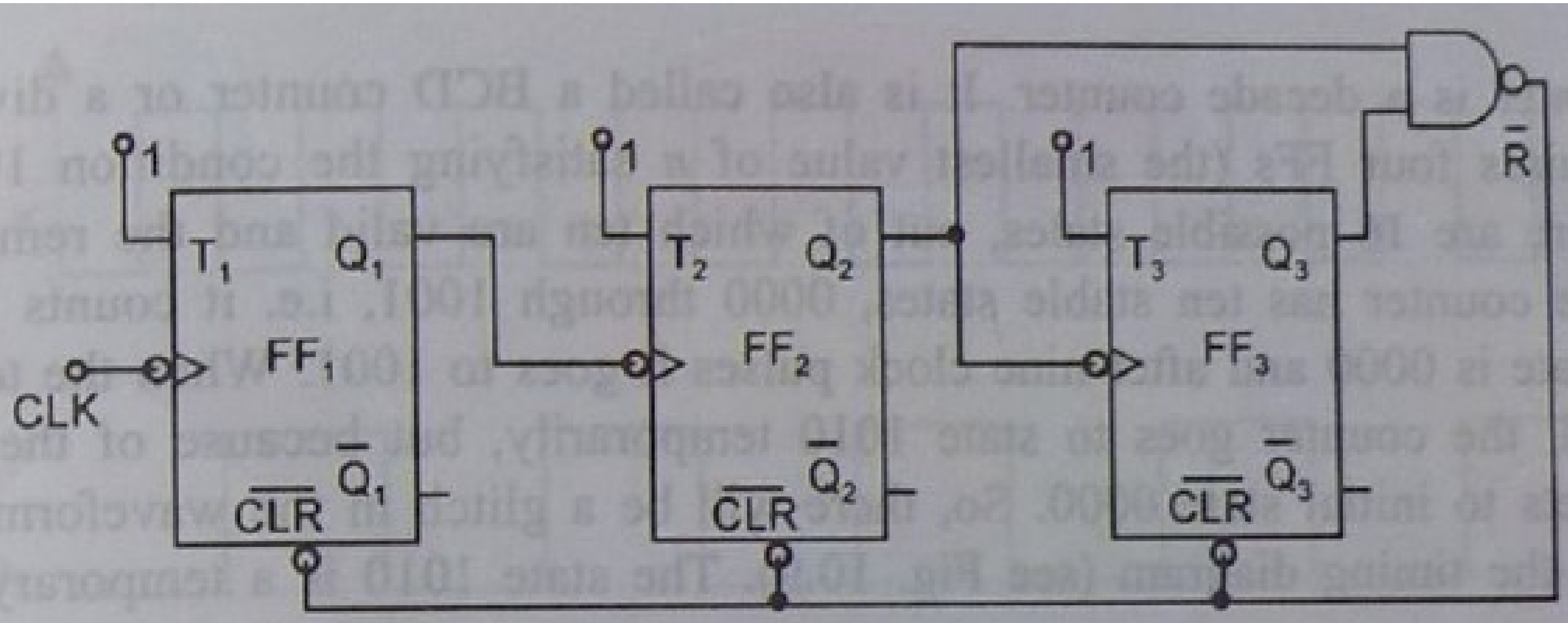


Mod – 6 Counter Design



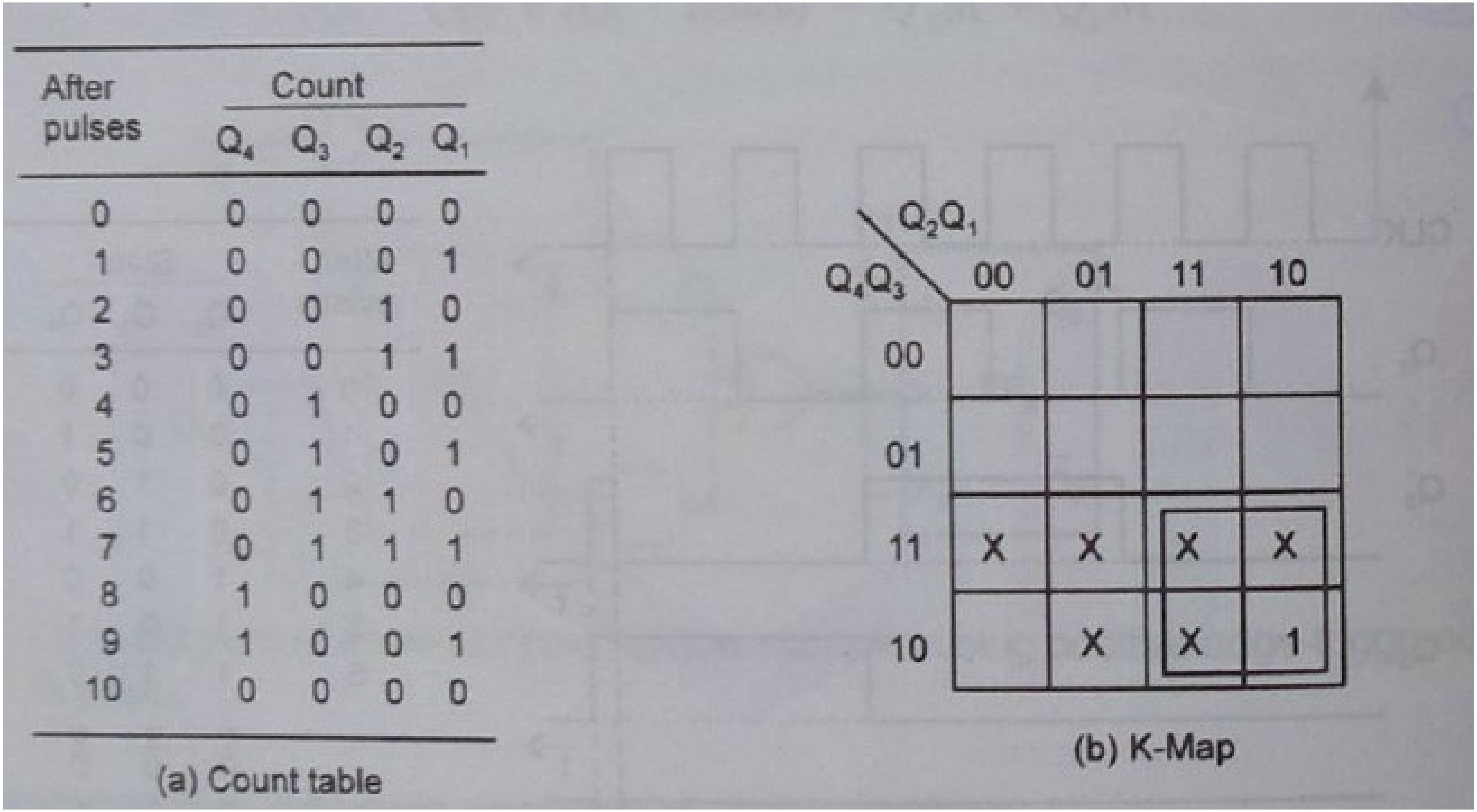
After pulses	State			R
	Q_3	Q_2	Q_1	
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
	↓	↓	↓	
	0	0	0	0
7	0	0	1	0

Mod – 6 Counter



(a) Logic diagram

Mod – 10 Asynchronous Counter



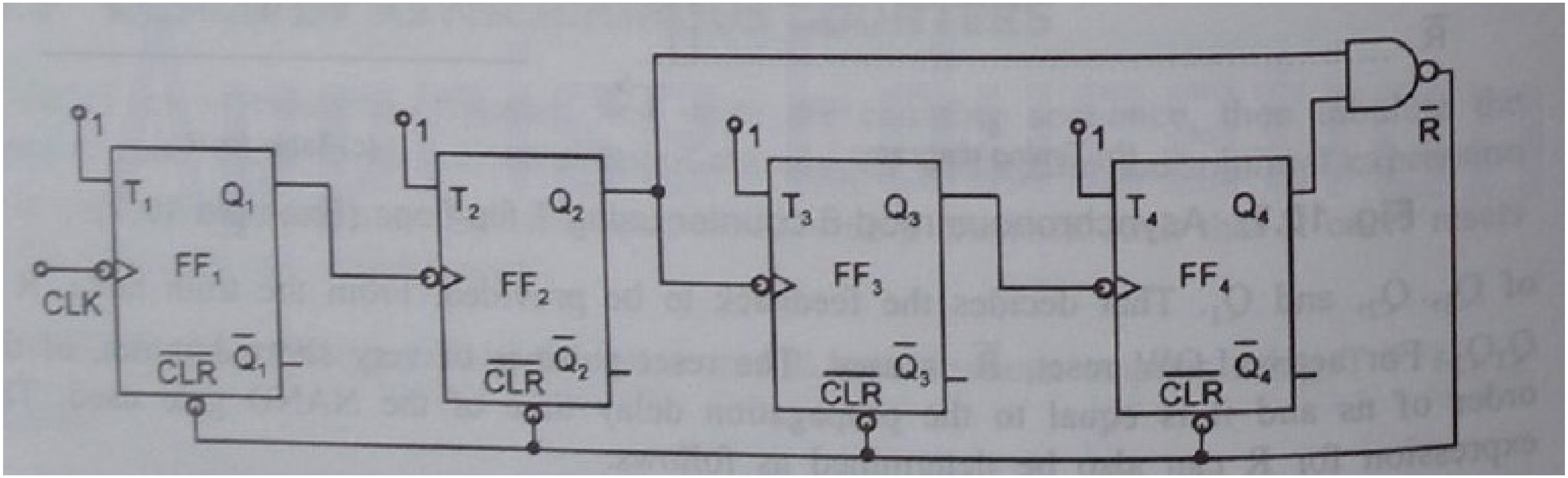
After pulses	Count			
	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

(a) Count table

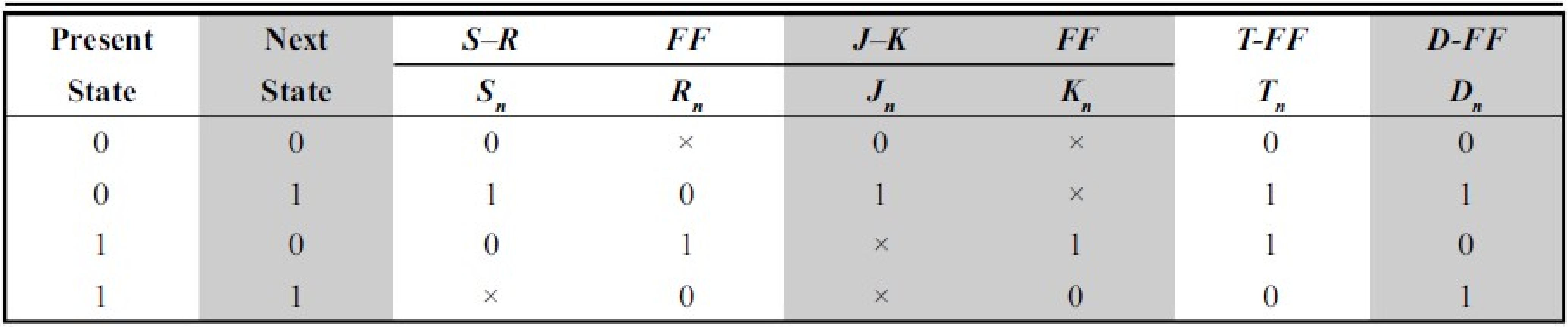
Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01				
11	X	X	X	X
10		X	X	1

(b) K-Map

Mod – 10 Counter

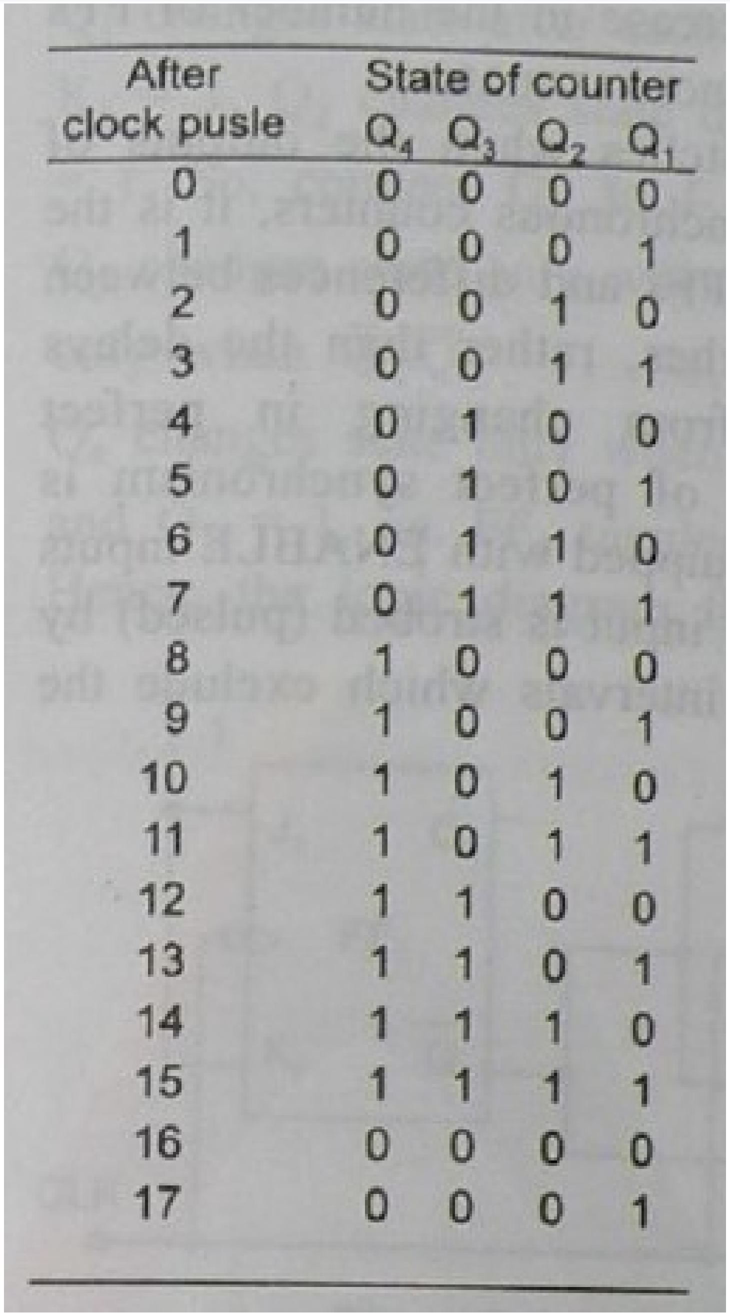


Excitation Tables



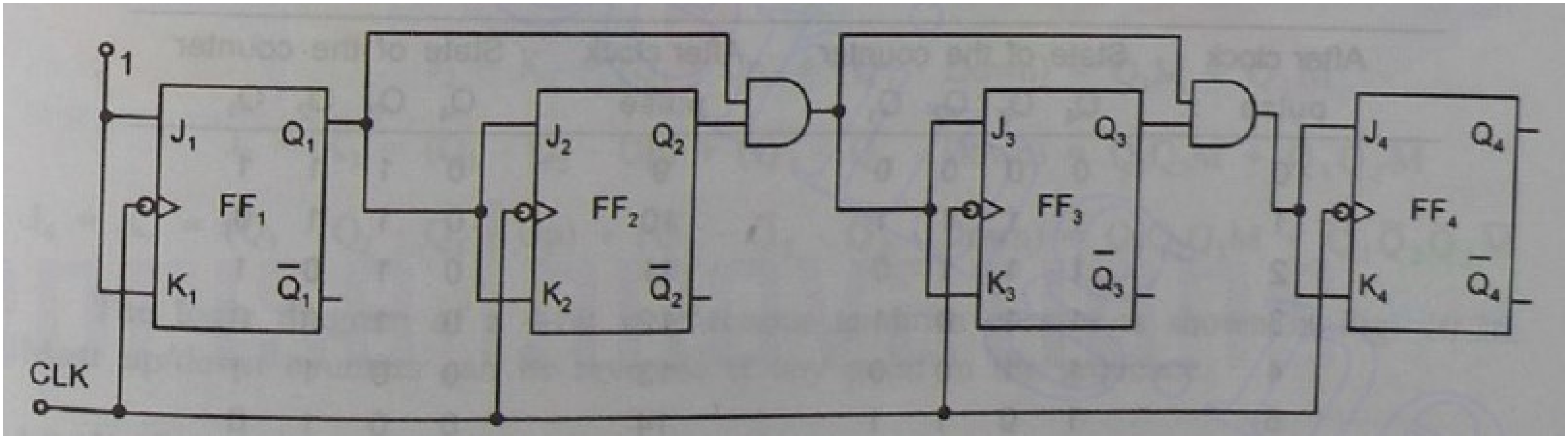
Present	Next	*S–R*	*FF*	*J–K*	*FF*	*T-FF*	*D-FF*
State	State	S_n	R_n	J_n	K_n	T_n	D_n
0	0	0	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1

4 – Bit Up Counter

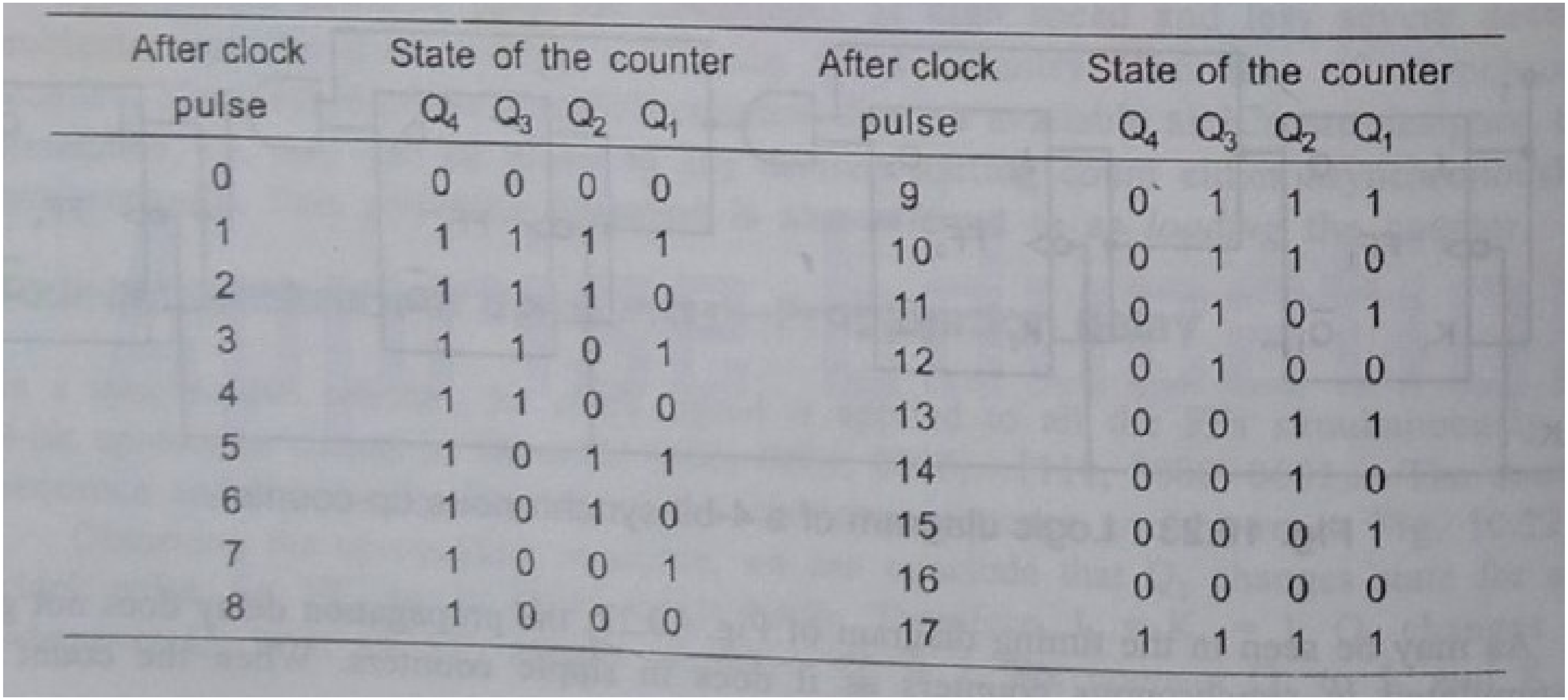


After clock pusle	State of counter Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0
17	0	0	0	1

4 – Bit Up Counter

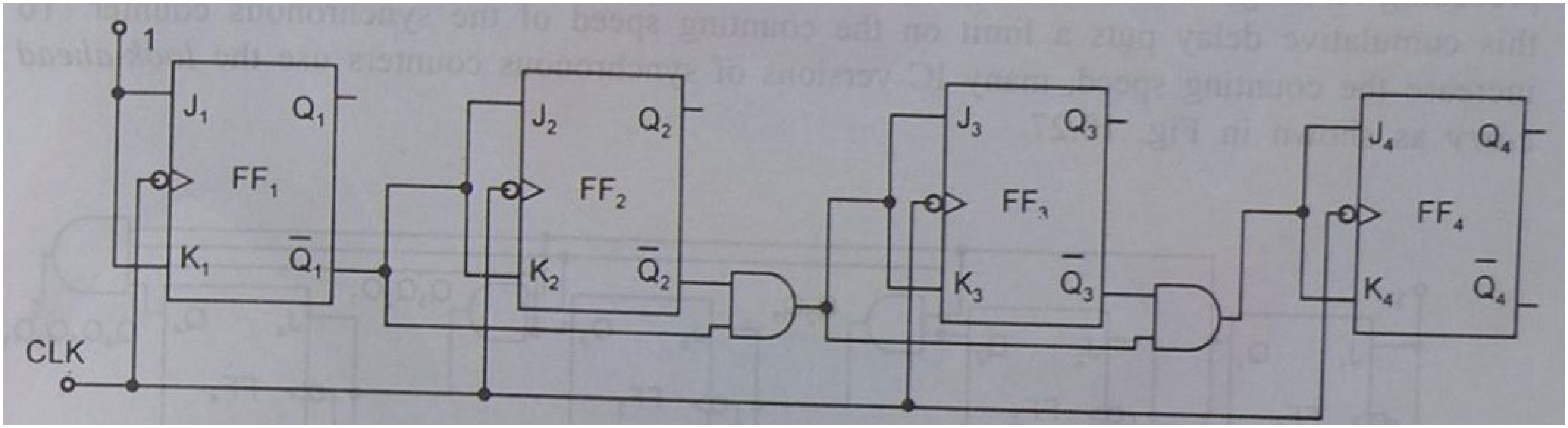


4 – Bit Down Counter

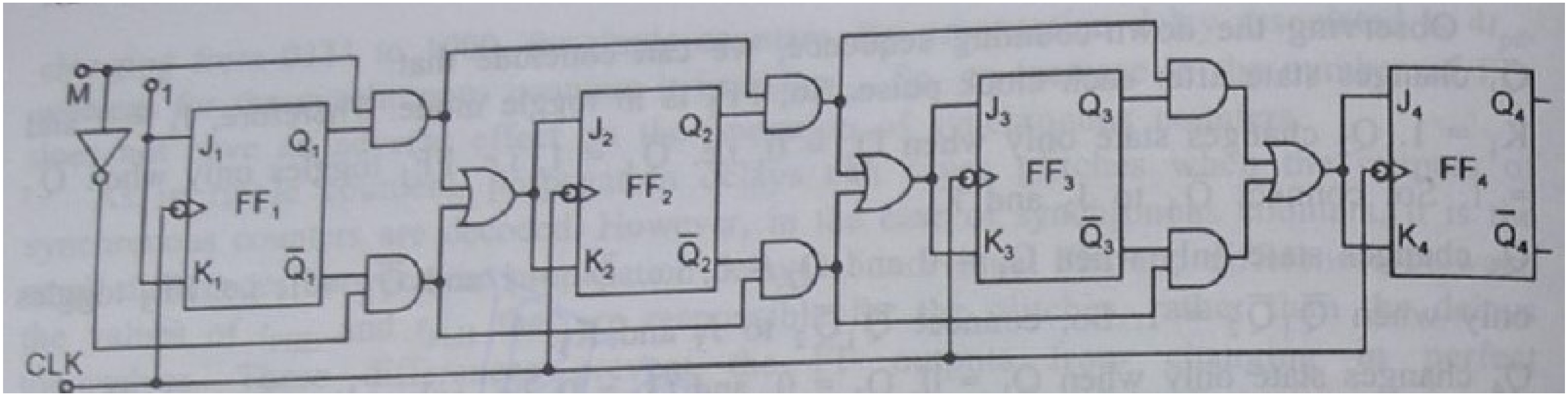


After clock pulse	State of the counter				After clock pulse	State of the counter			
	Q_4	Q_3	Q_2	Q_1		Q_4	Q_3	Q_2	Q_1
0	0	0	0	0	9	0`	1	1	1
1	1	1	1	1	10	0	1	1	0
2	1	1	1	0	11	0	1	0	1
3	1	1	0	1	12	0	1	0	0
4	1	1	0	0	13	0	0	1	1
5	1	0	1	1	14	0	0	1	0
6	1	0	1	0	15	0	0	0	1
7	1	0	0	1	16	0	0	0	0
8	1	0	0	0	17	1	1	1	1

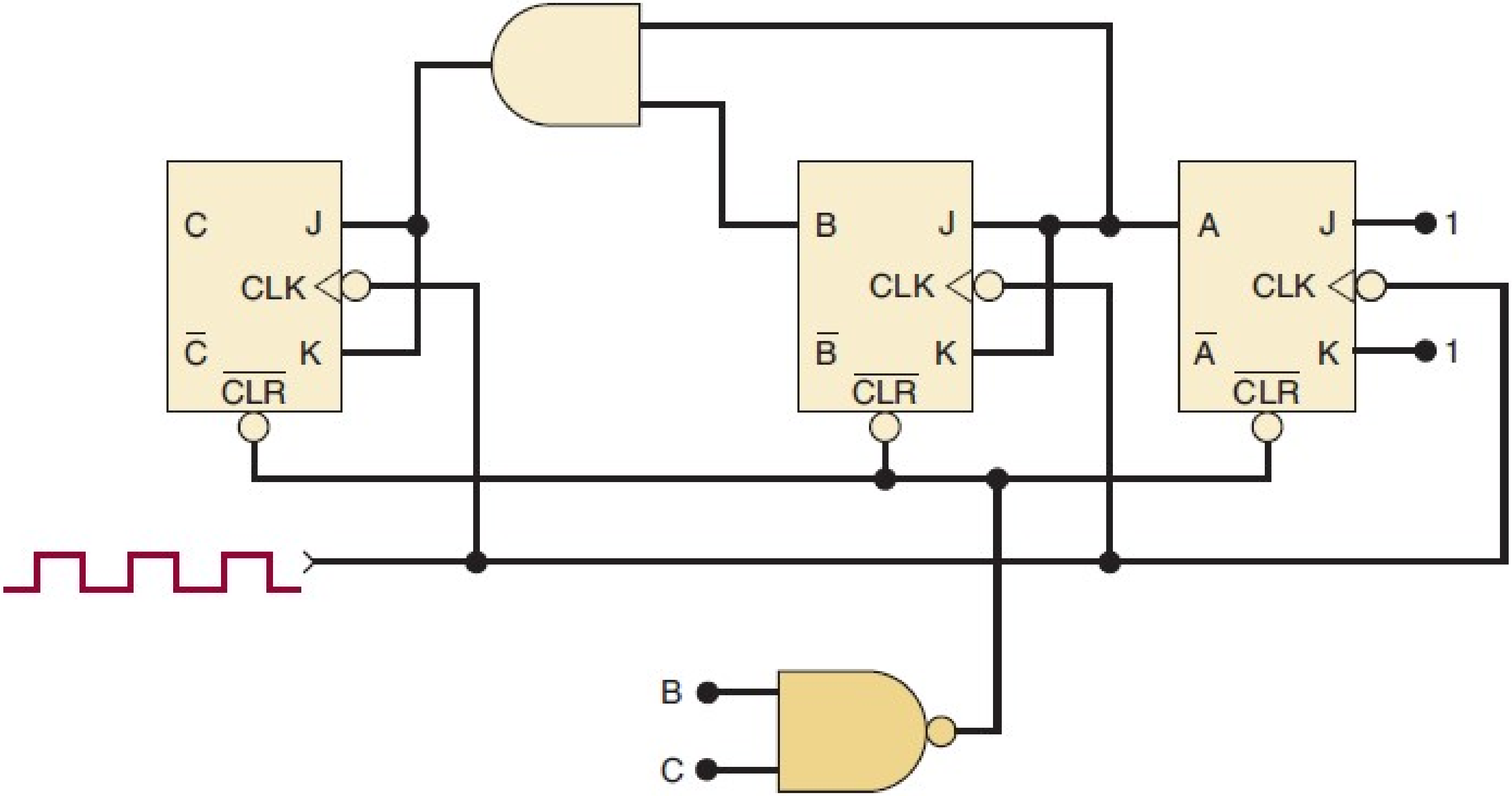
4 – Bit Down Counter



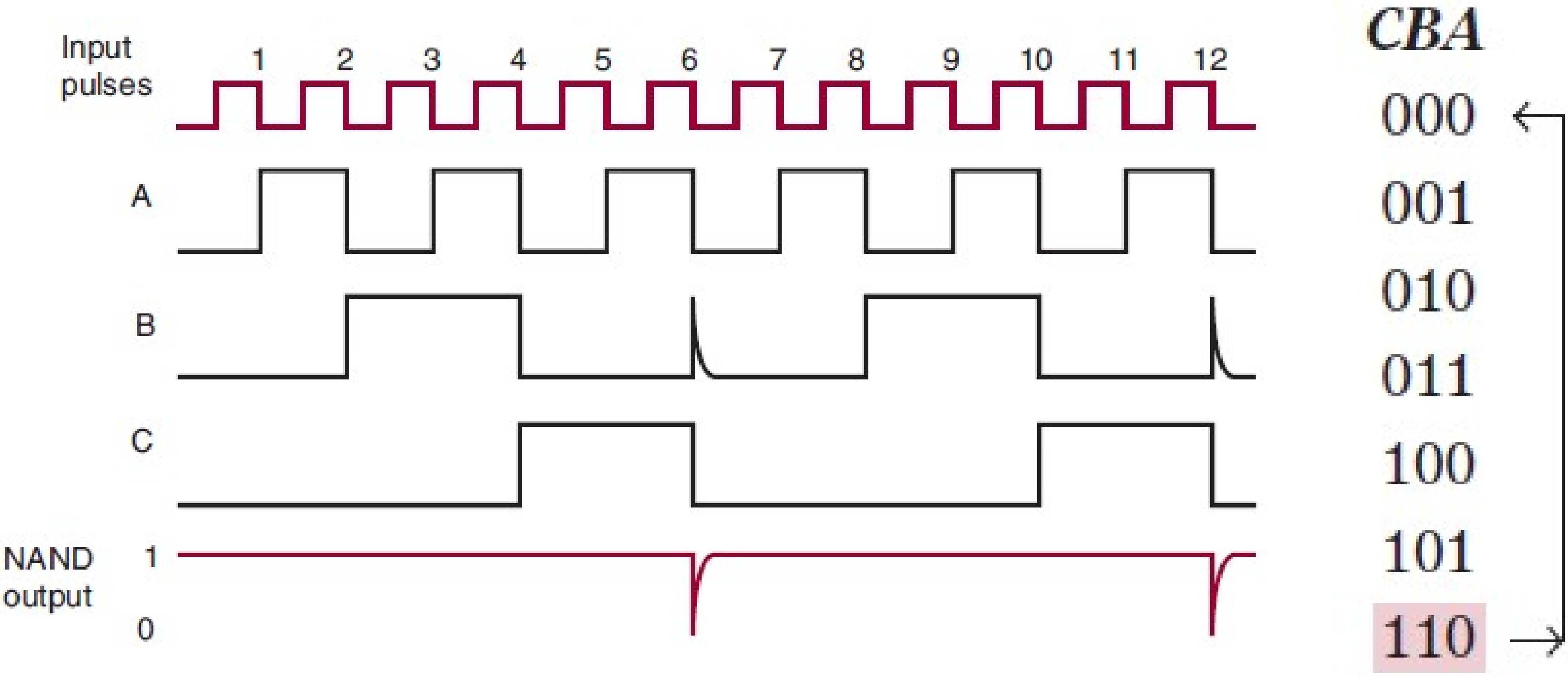
4 – Bit UP/DOWN Counter



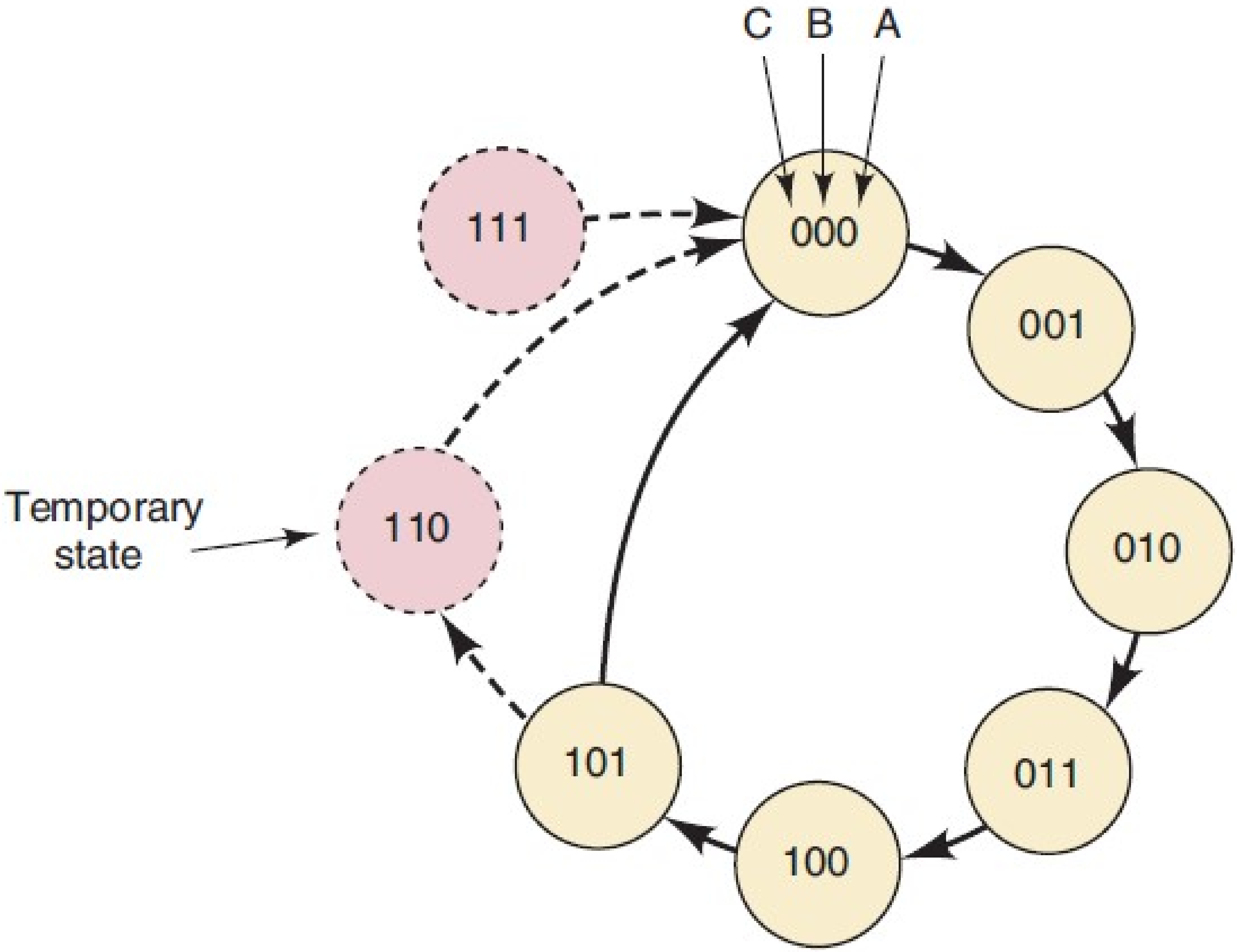
Mod < 2^n



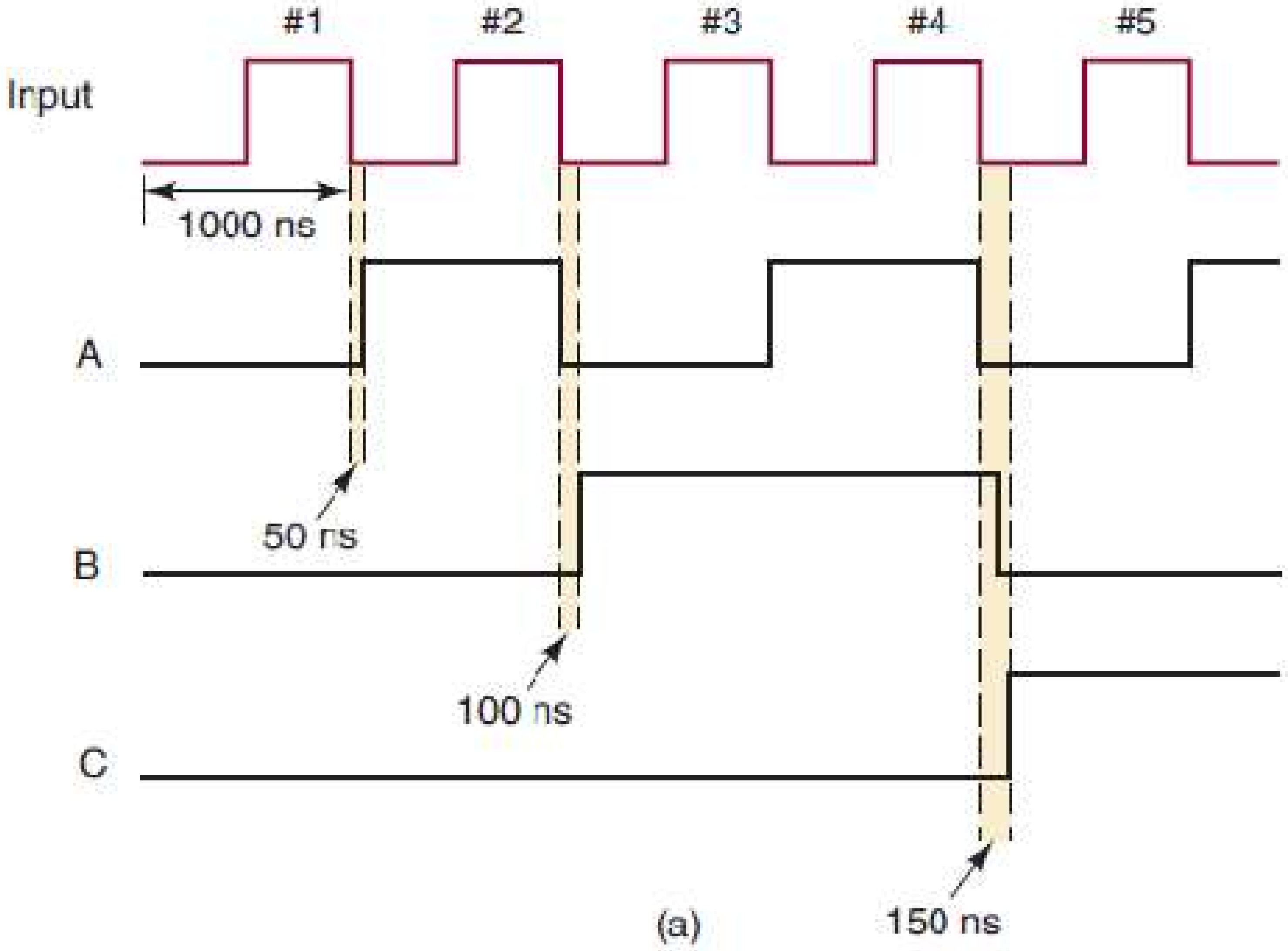
Mod – 6 SC


Input
pulses
1
2
3
4
5
6
7
8
9
10
11
12
A
B
C
NAND
output
1
0
CBA
000
001
010
011
100
101
110

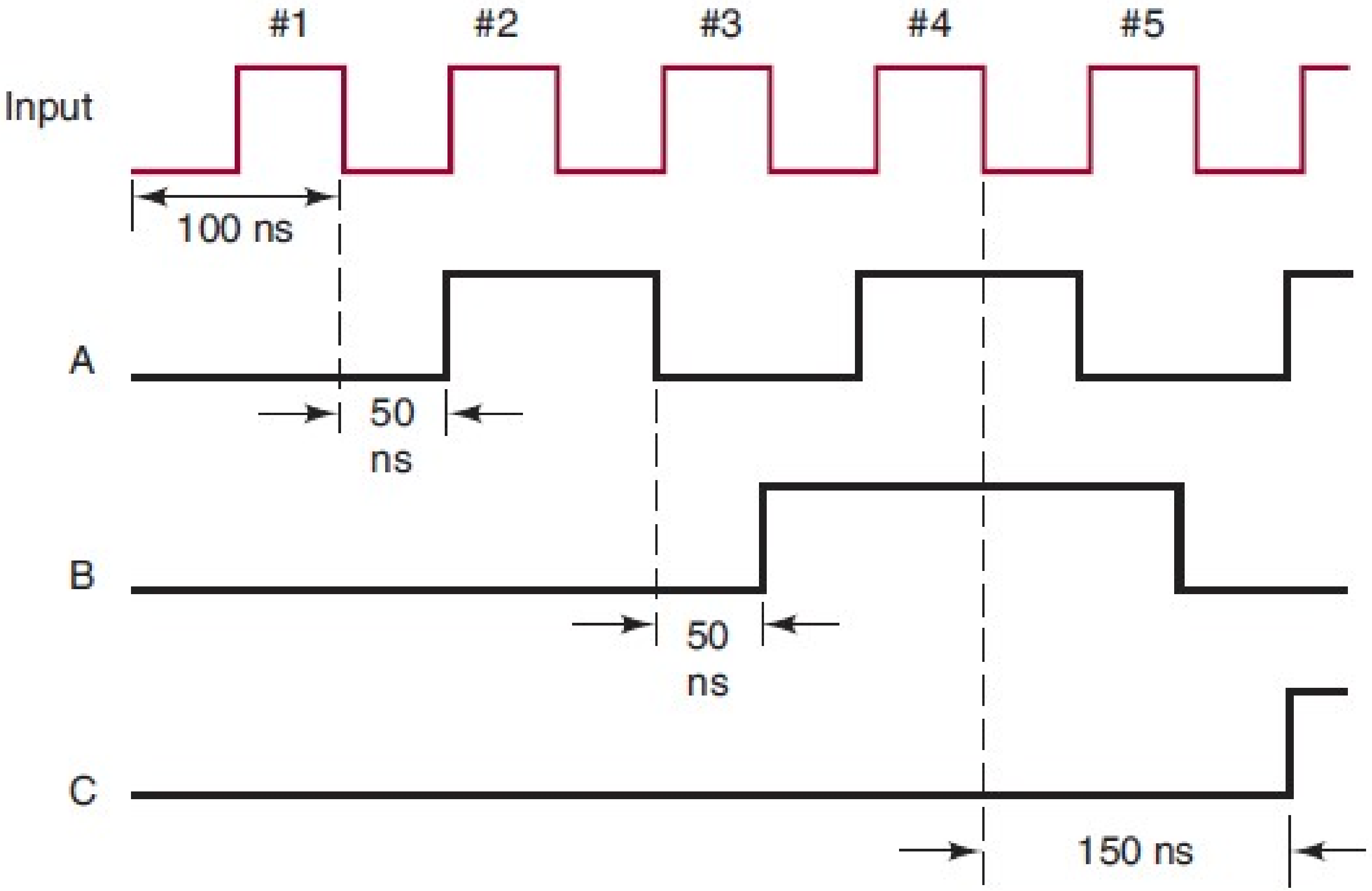
State Transition Diagram



Propagation Delay



(a)



The 100 condition does *not* occur.

(b)

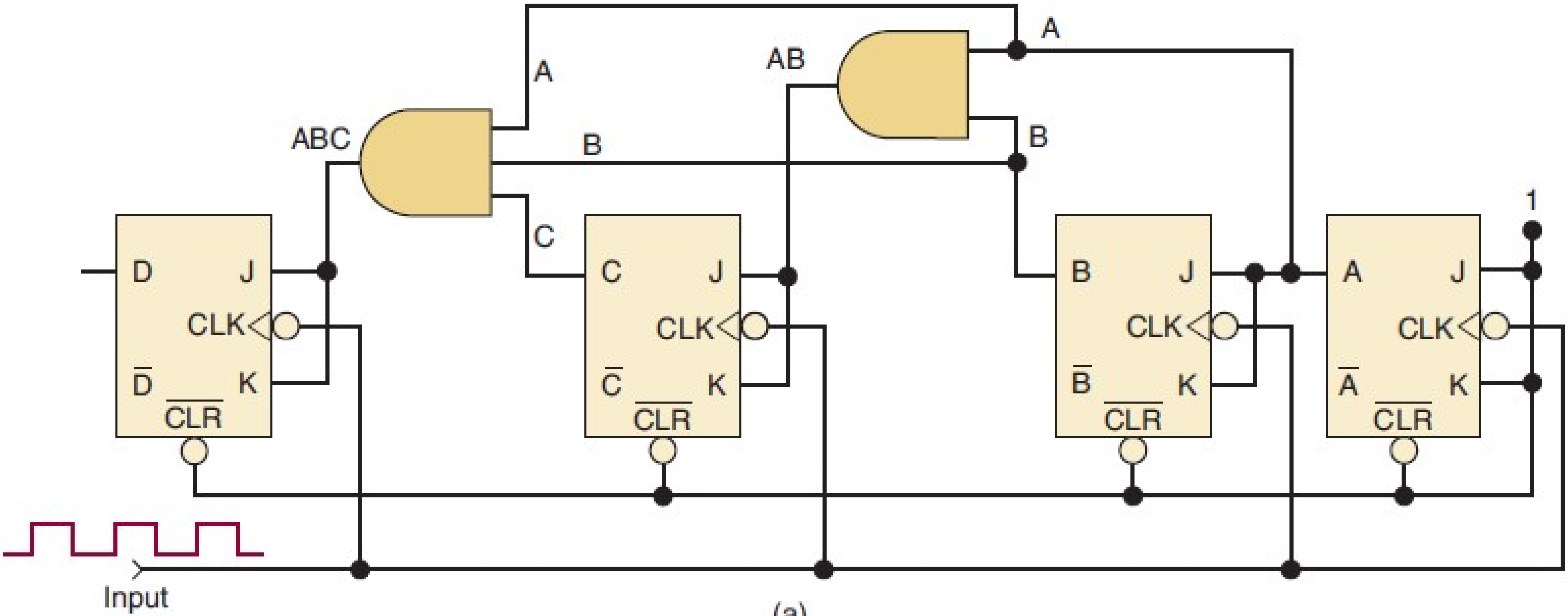
$$T_{\mathrm{clock}} \geq N \times \, t_{\mathrm{pd}}$$

$$f_{\max} = \frac{1}{N \times \, t_{\mathrm{pd}}}$$

Concept Check

A certain J-K flip-flop has tpd = 12 ns. What is the largest MOD counter that can be constructed from these FFs and still operate up to 10 MHz?

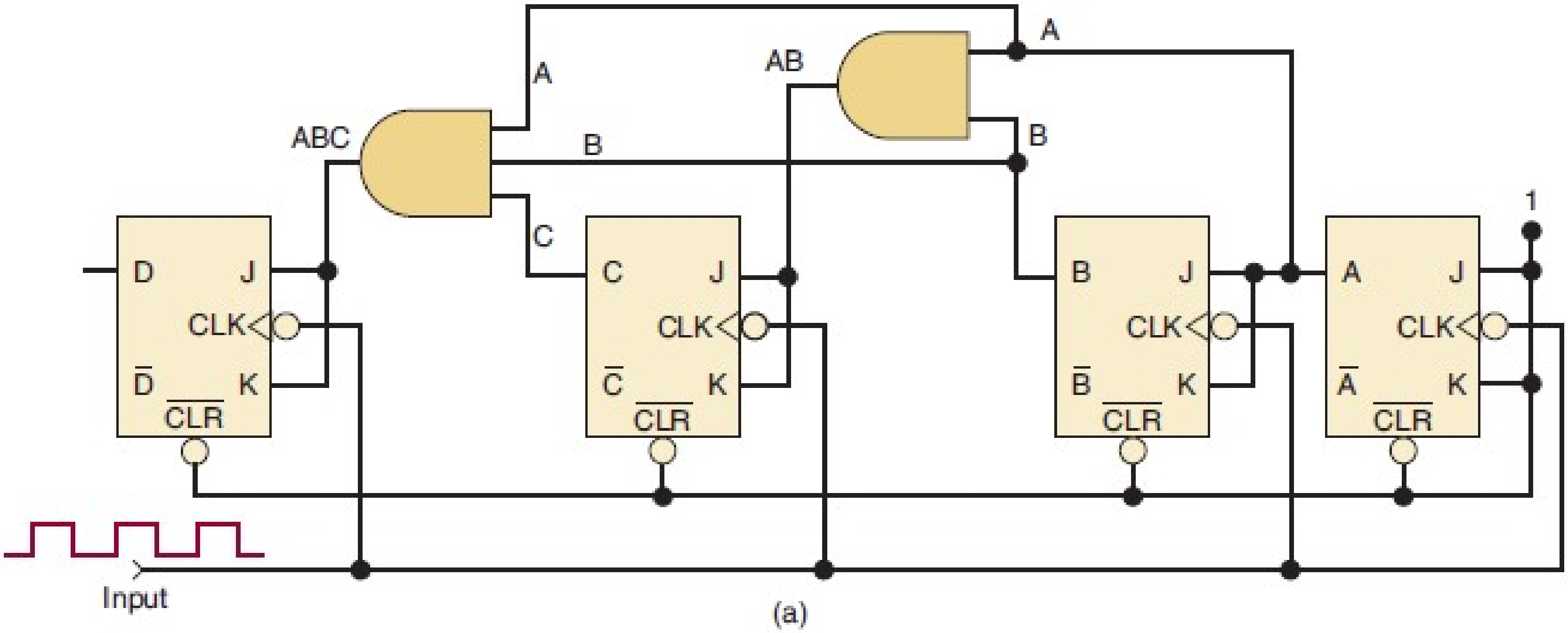
Advantage of SC over ASC



(a)

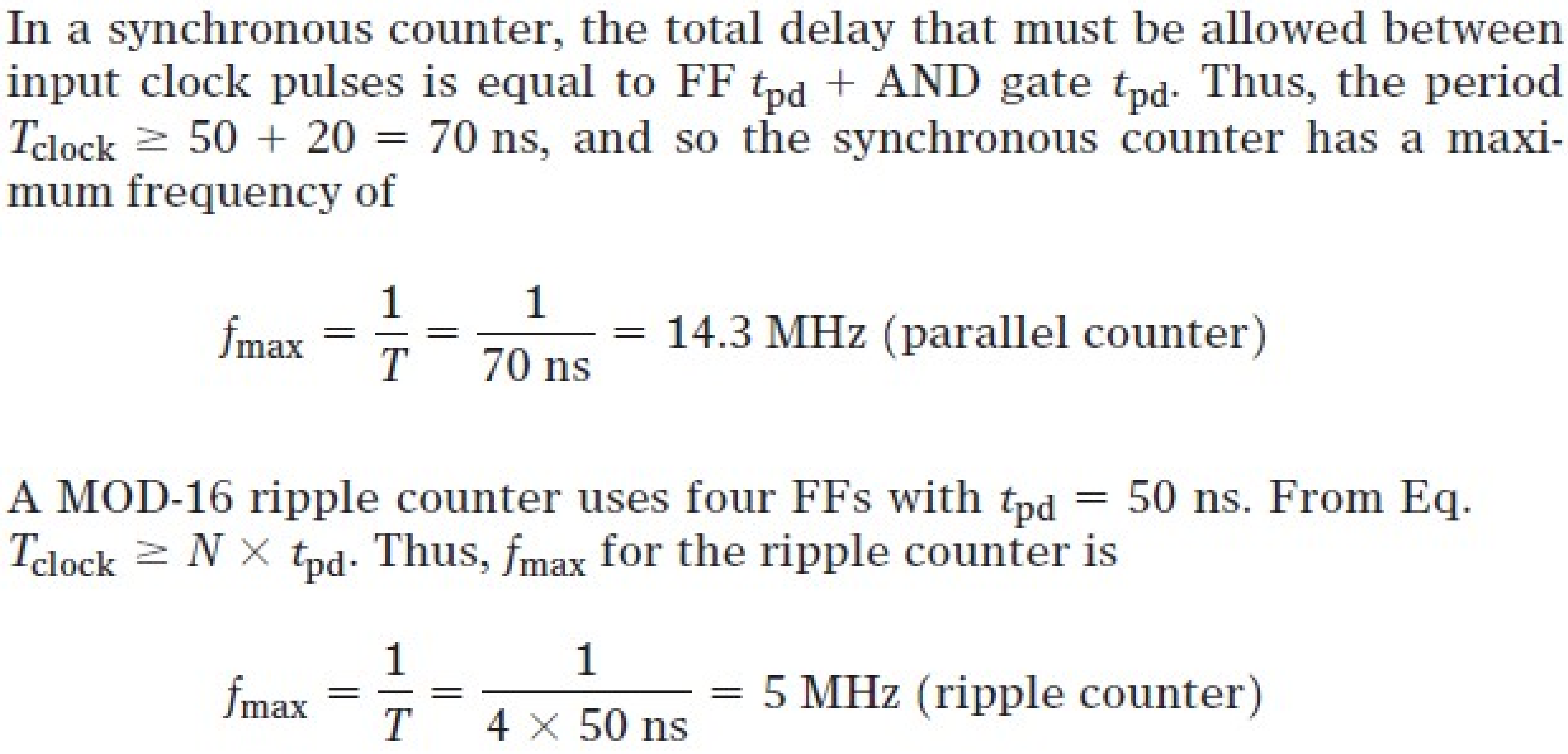
$$\text{total delay} = \text{FF } t_{\text{pd}} + \text{AND gate } t_{\text{pd}}$$

Concept Check



Determine fmax for the synchronous counter of Figure above if tpd for each FF is 50 ns and tpd for each AND gate is 20 ns. Compare this value with fmax for a MOD-16 ripple counter.

Solution



In a synchronous counter, the total delay that must be allowed between input clock pulses is equal to FF t_{pd} + AND gate t_{pd}. Thus, the period $T_{clock} \geq 50 + 20 = 70$ ns, and so the synchronous counter has a maximum frequency of

$$f_{max} = \frac{1}{T} = \frac{1}{70 \text{ ns}} = 14.3 \text{ MHz (parallel counter)}$$

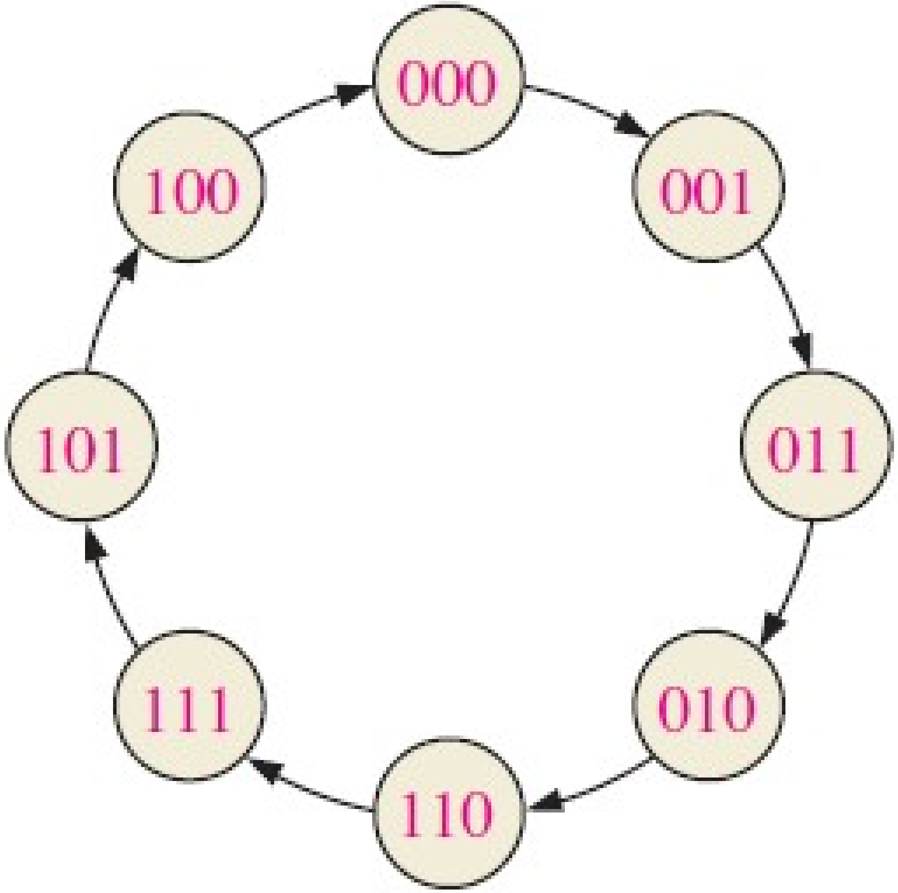
A MOD-16 ripple counter uses four FFs with $t_{pd} = 50$ ns. From Eq. $T_{clock} \geq N \times t_{pd}$. Thus, f_{max} for the ripple counter is

$$f_{max} = \frac{1}{T} = \frac{1}{4 \times 50 \text{ ns}} = 5 \text{ MHz (ripple counter)}$$

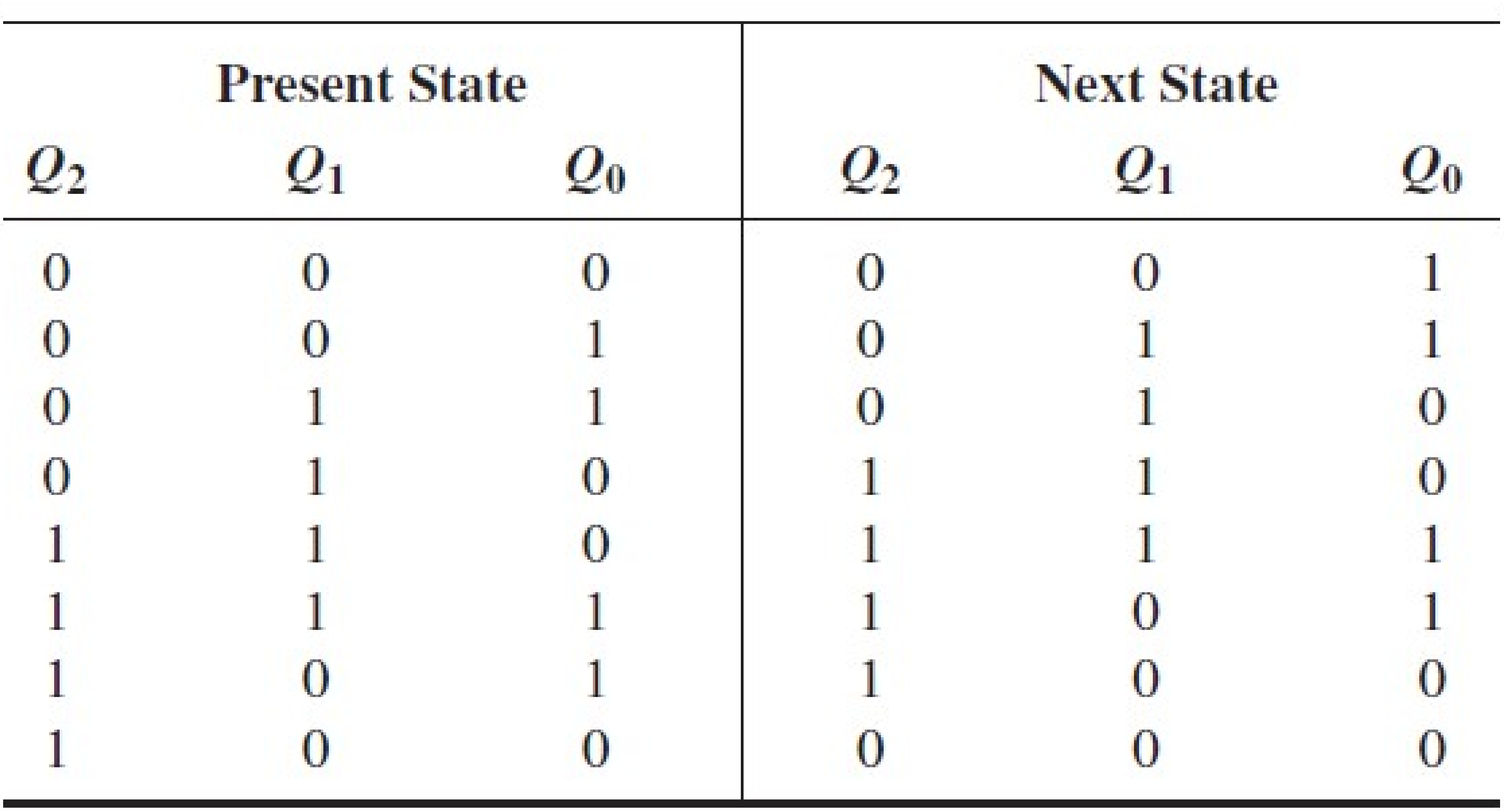
Synchronous Counter Design

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

State Diagram

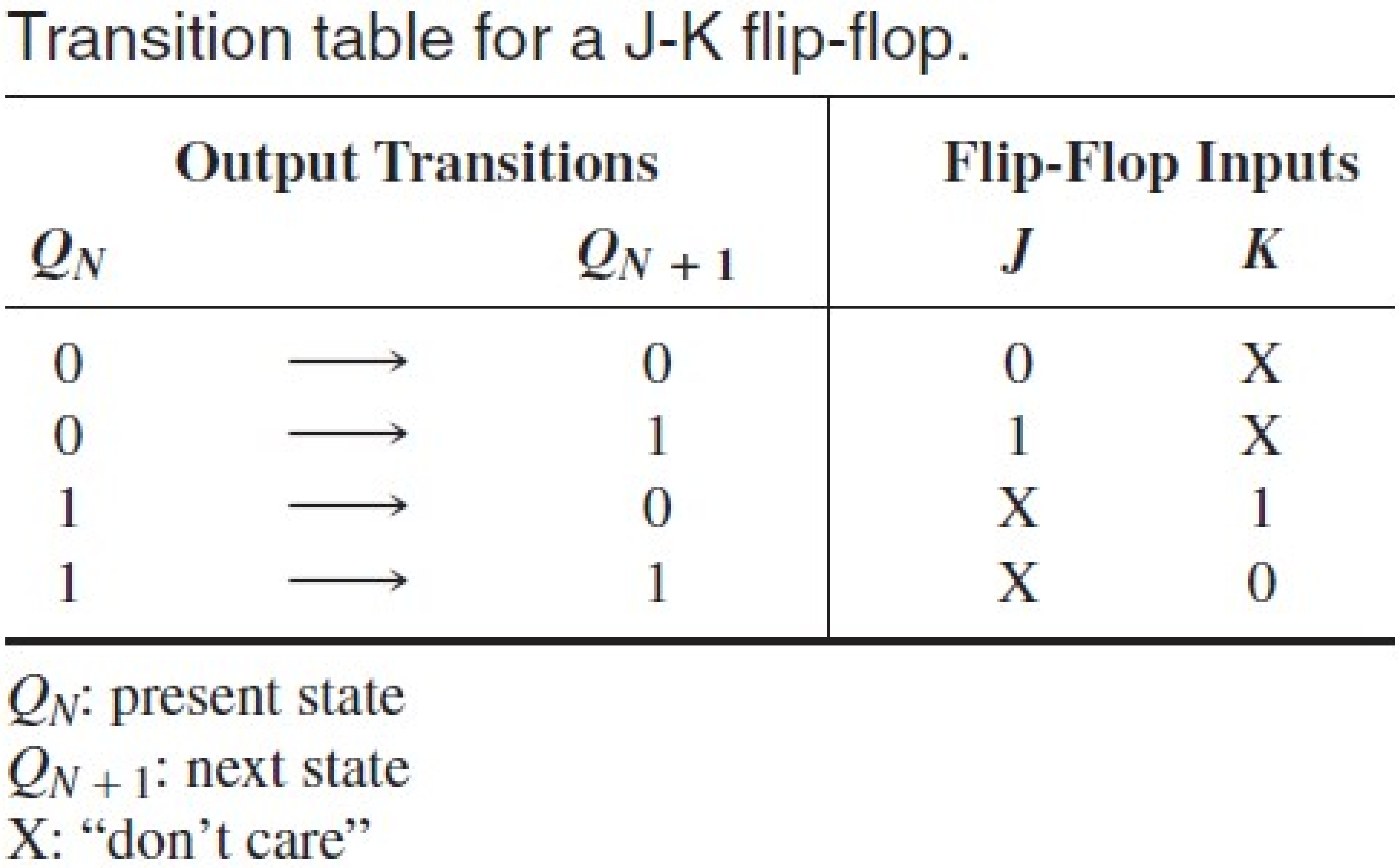


Next State Table



Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Transition Table

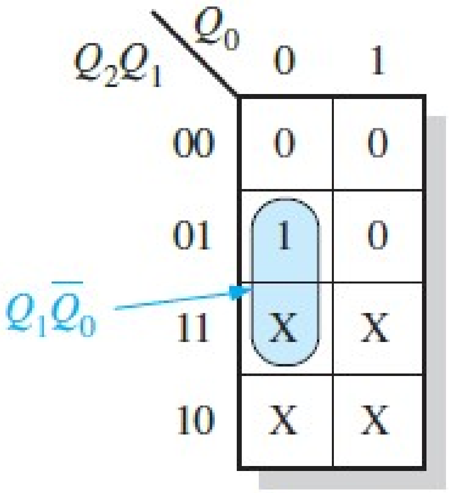


Transition table for a J-K flip-flop.

Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Q_N: present state
Q_{N+1}: next state
X: "don't care"

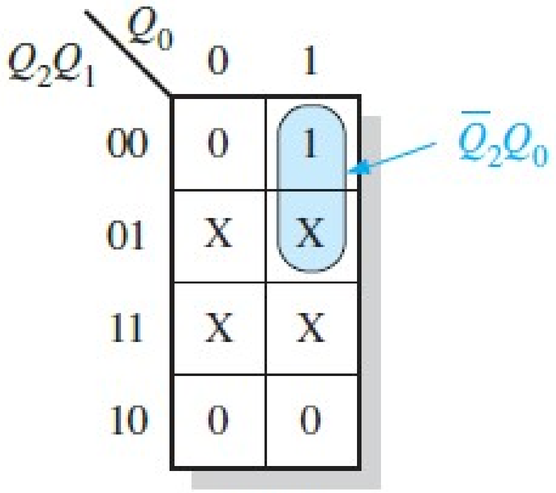
K-Map Simplifications



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

$Q_1\overline{Q}_0$

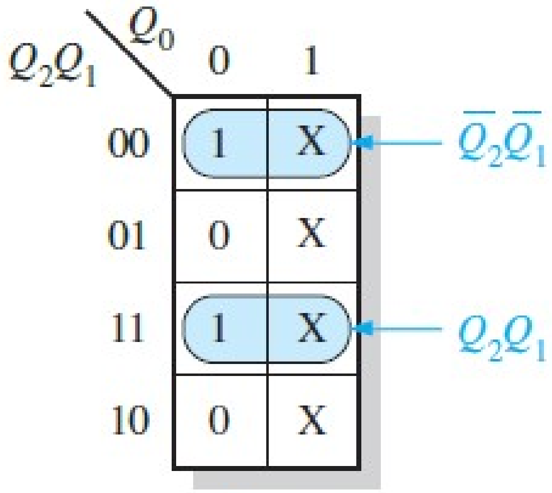
J_2 map



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

$\overline{Q}_2Q_0$

J_1 map

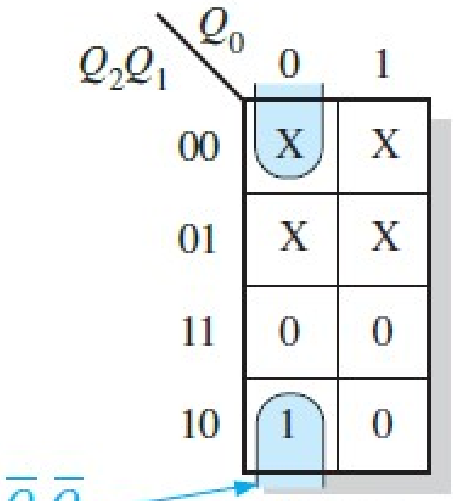


Q_2Q_1 \ Q_0	0	1
00	1	X
01	0	X
11	1	X
10	0	X

$\overline{Q}_2\overline{Q}_1$

Q_2Q_1

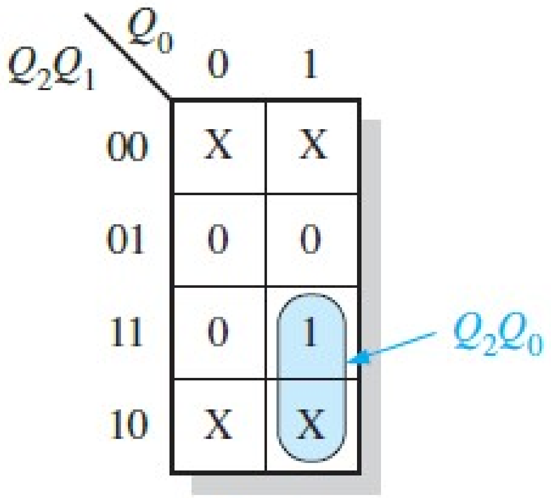
J_0 map



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

$\overline{Q}_1\overline{Q}_0$

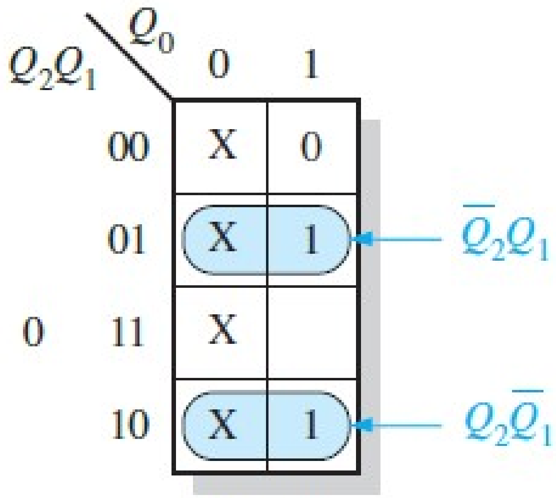
K_2 map



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

Q_2Q_0

K_1 map



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

0

$\overline{Q}_2Q_1$

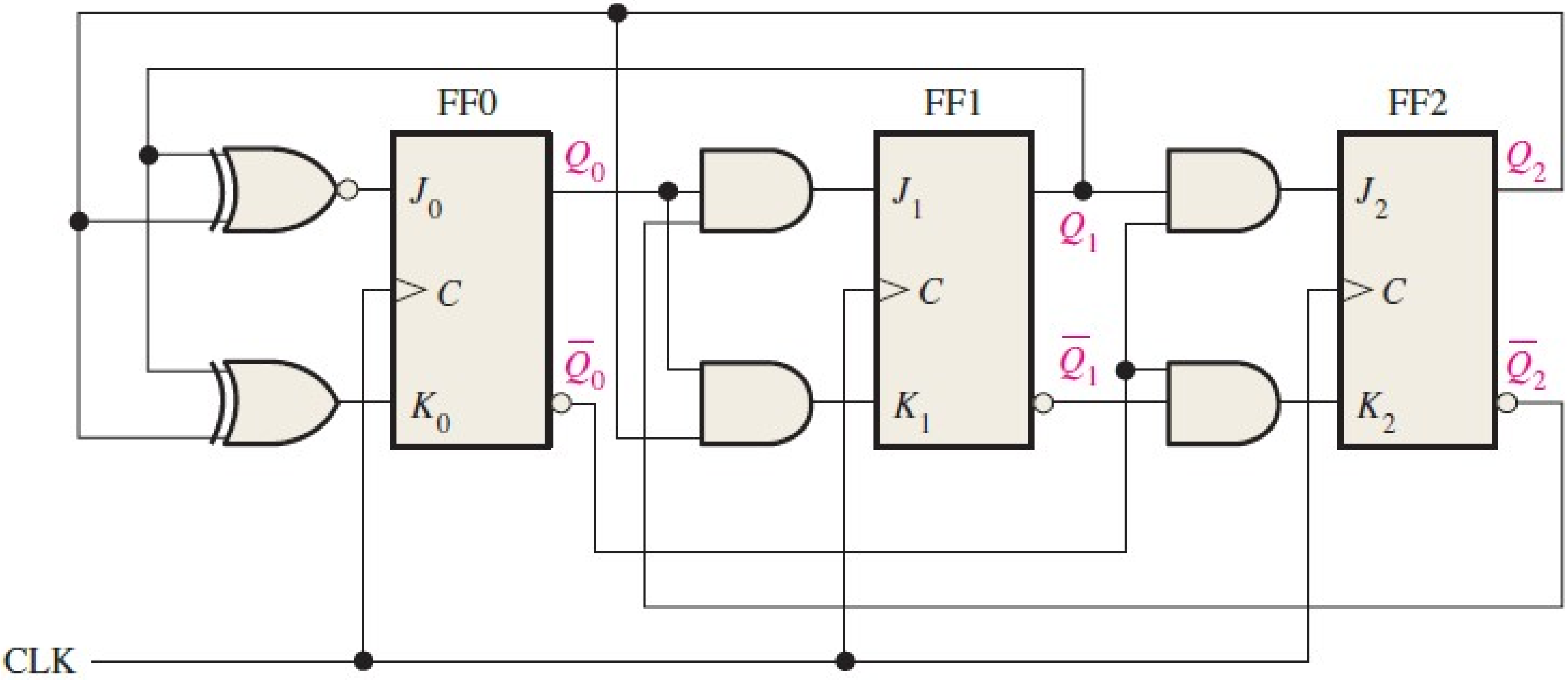
$Q_2\overline{Q}_1$

K_0 map

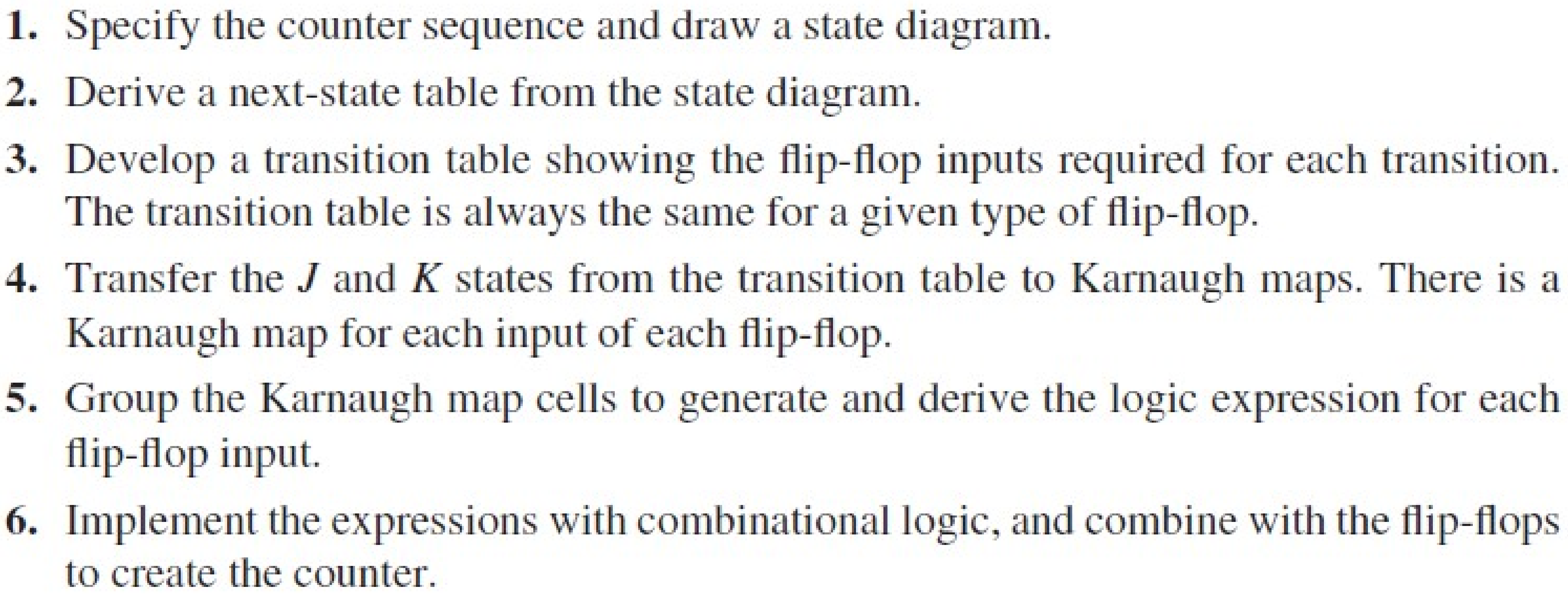
Flip-Flop Inputs

$$\begin{aligned}
J_0 &= Q_2Q_1 + \overline{Q}_2\overline{Q}_1 = \overline{Q_2 \oplus Q_1} \\
K_0 &= Q_2\overline{Q}_1 + \overline{Q}_2Q_1 = Q_2 \oplus Q_1 \\
J_1 &= \overline{Q}_2Q_0 \\
K_1 &= Q_2Q_0 \\
J_2 &= Q_1\overline{Q}_0 \\
K_2 &= \overline{Q}_1\overline{Q}_0
\end{aligned}$$

Circuit Implementation

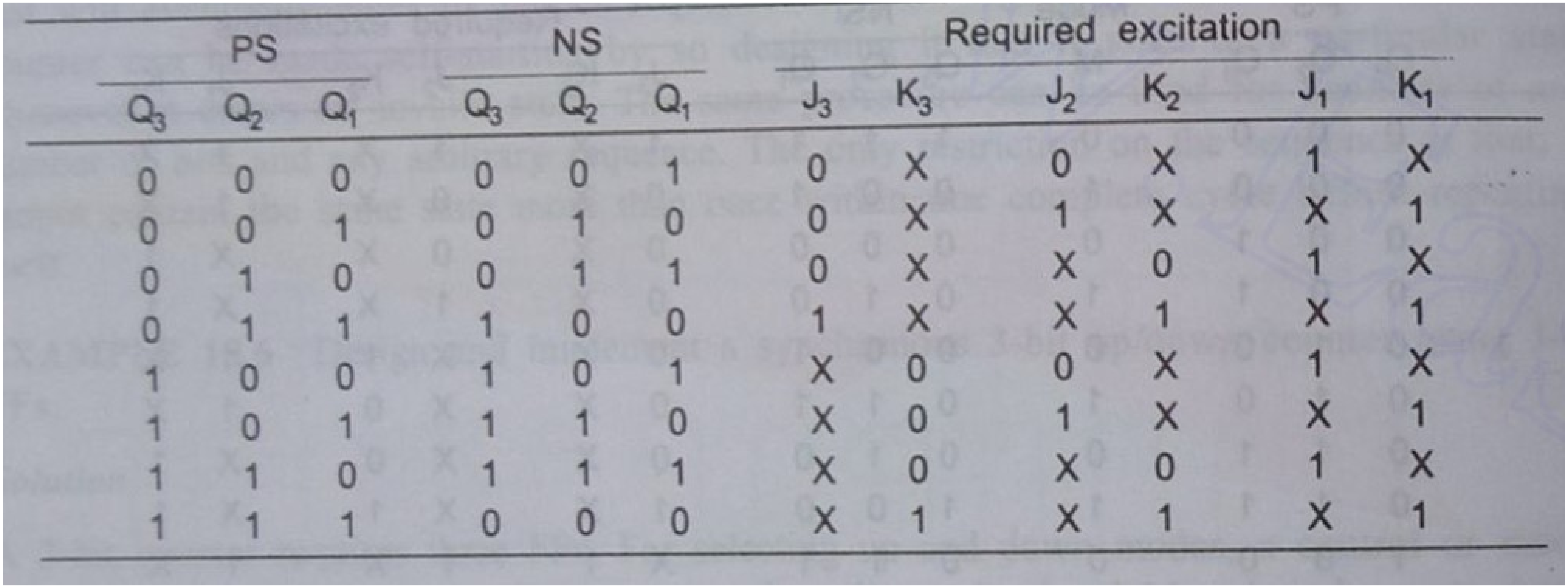


Design Summary



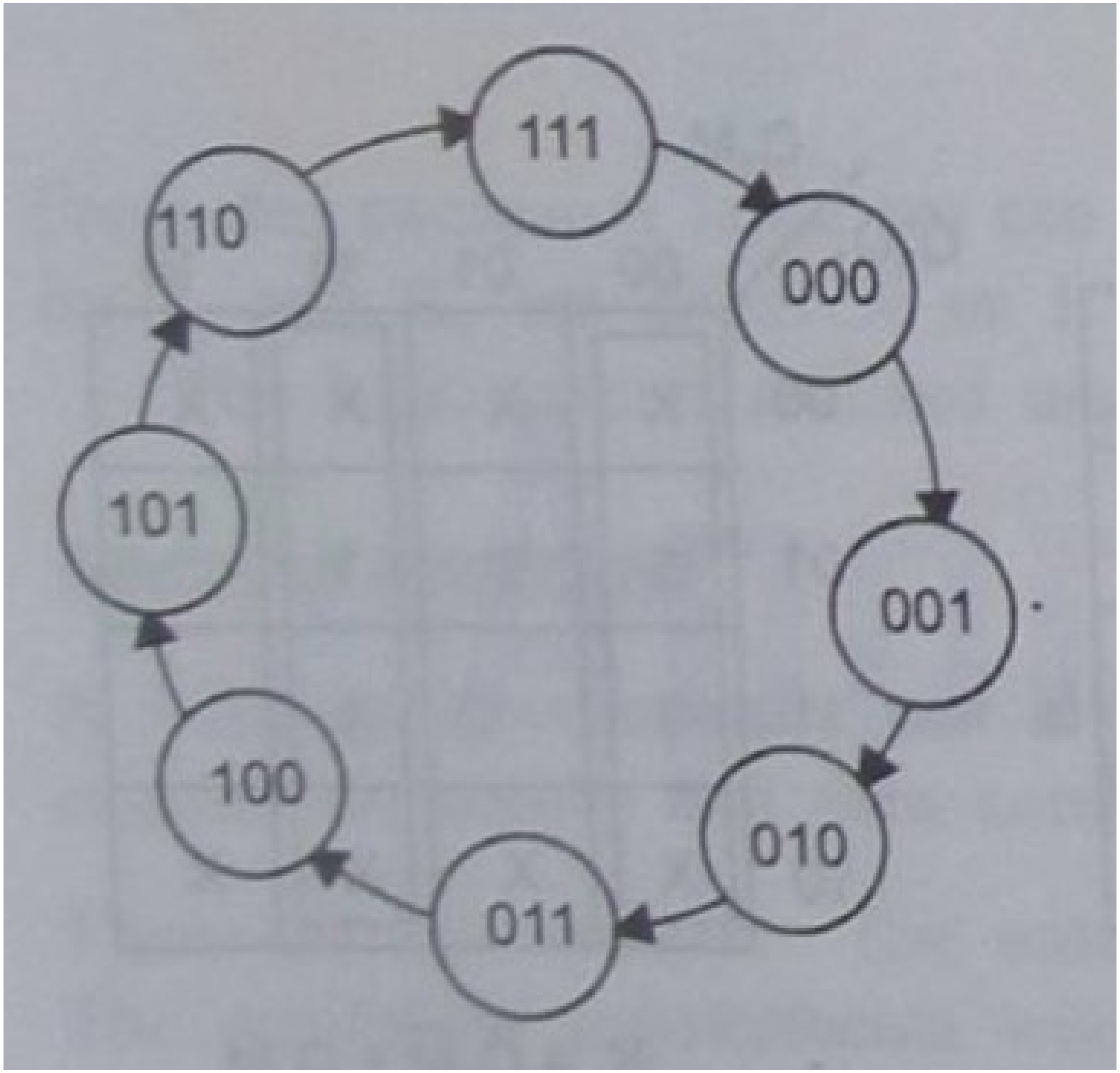
1. Specify the counter sequence and draw a state diagram.
2. Derive a next-state table from the state diagram.
3. Develop a transition table showing the flip-flop inputs required for each transition. The transition table is always the same for a given type of flip-flop.
4. Transfer the *J* and *K* states from the transition table to Karnaugh maps. There is a Karnaugh map for each input of each flip-flop.
5. Group the Karnaugh map cells to generate and derive the logic expression for each flip-flop input.
6. Implement the expressions with combinational logic, and combine with the flip-flops to create the counter.

3 – Bit UP Counter

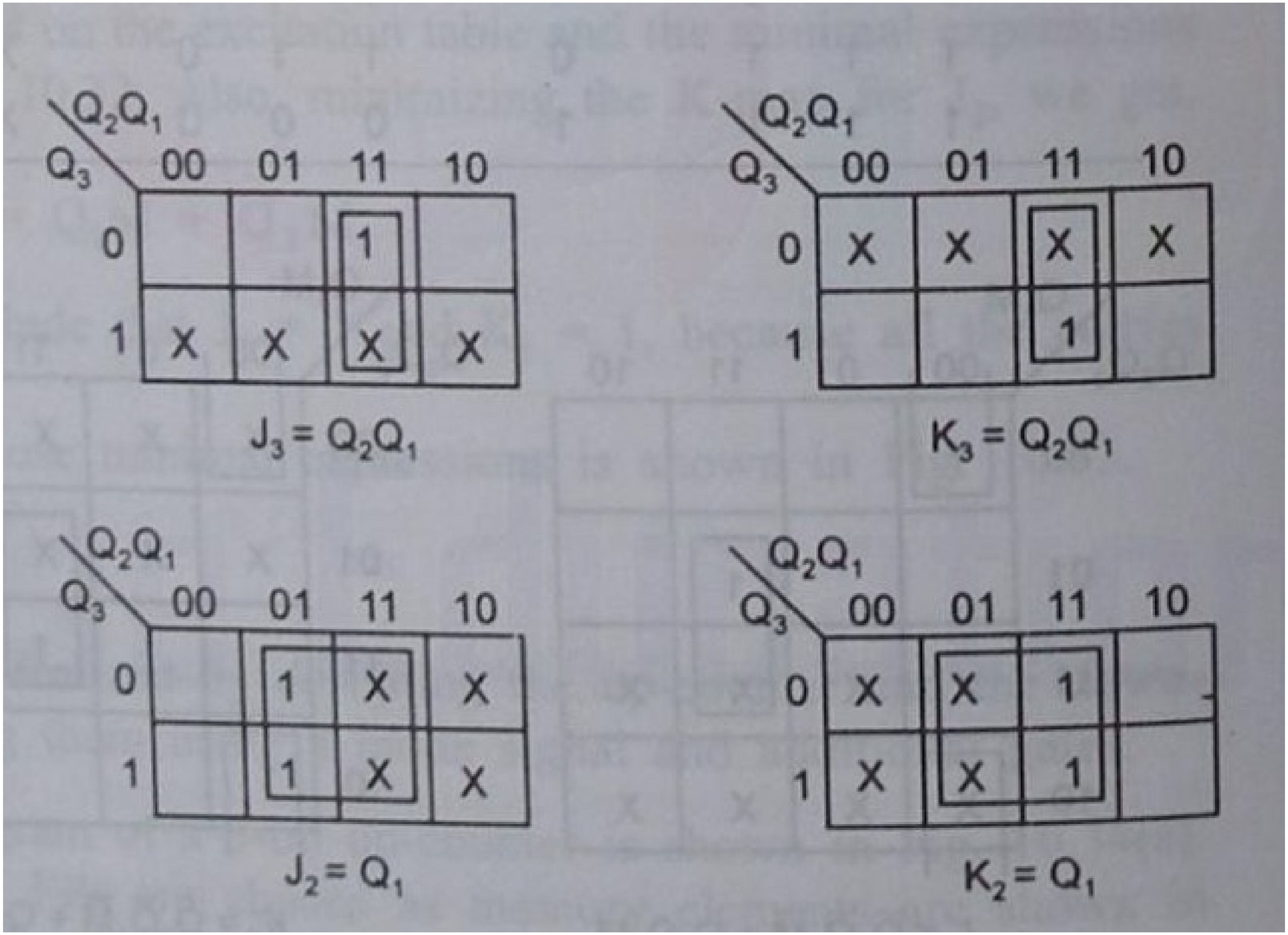


PS			NS			Required excitation					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

3 – Bit UP Counter



3 – Bit UP Counter



Q_3 \ Q_2Q_1	00	01	11	10
0			1	
1	X	X	X	X

$J_3 = Q_2Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	X	X
1			1	

$K_3 = Q_2Q_1$

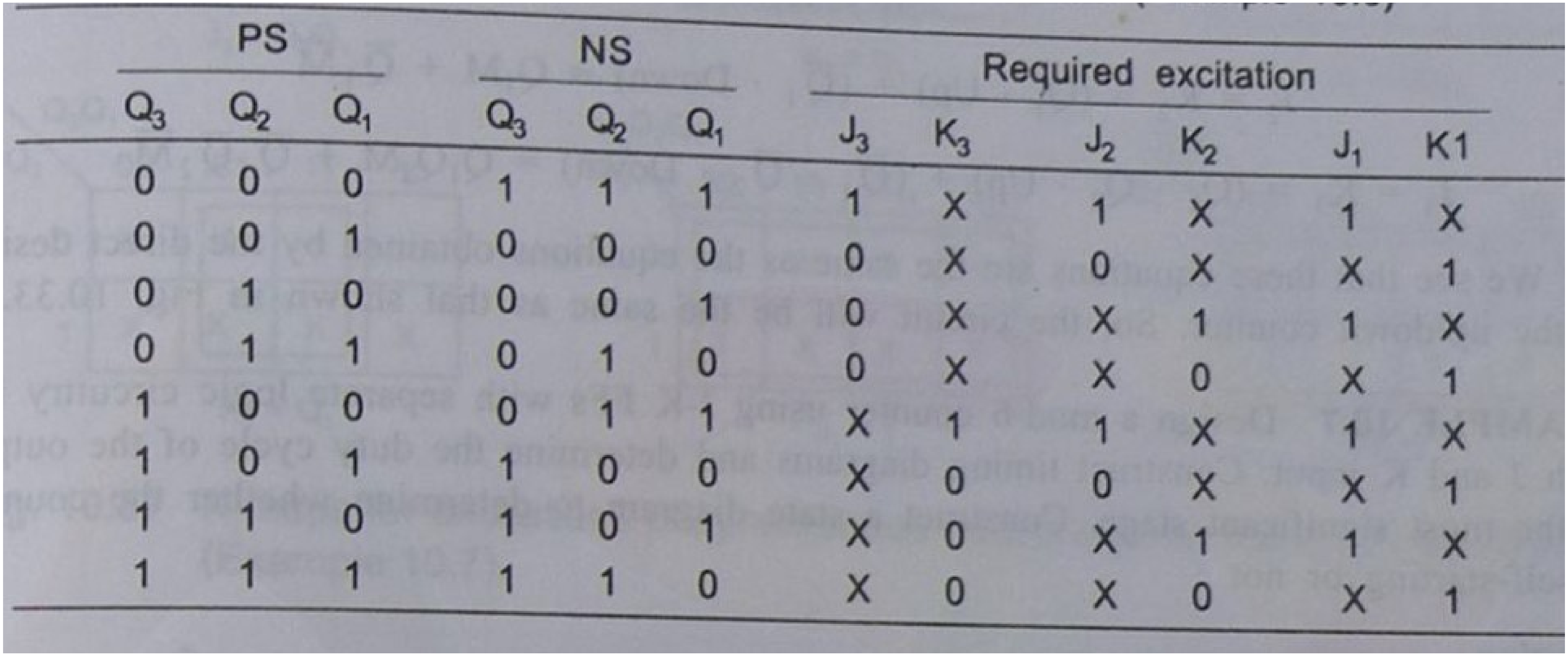
Q_3 \ Q_2Q_1	00	01	11	10
0		1	X	X
1		1	X	X

$J_2 = Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	1	
1	X	X	1	

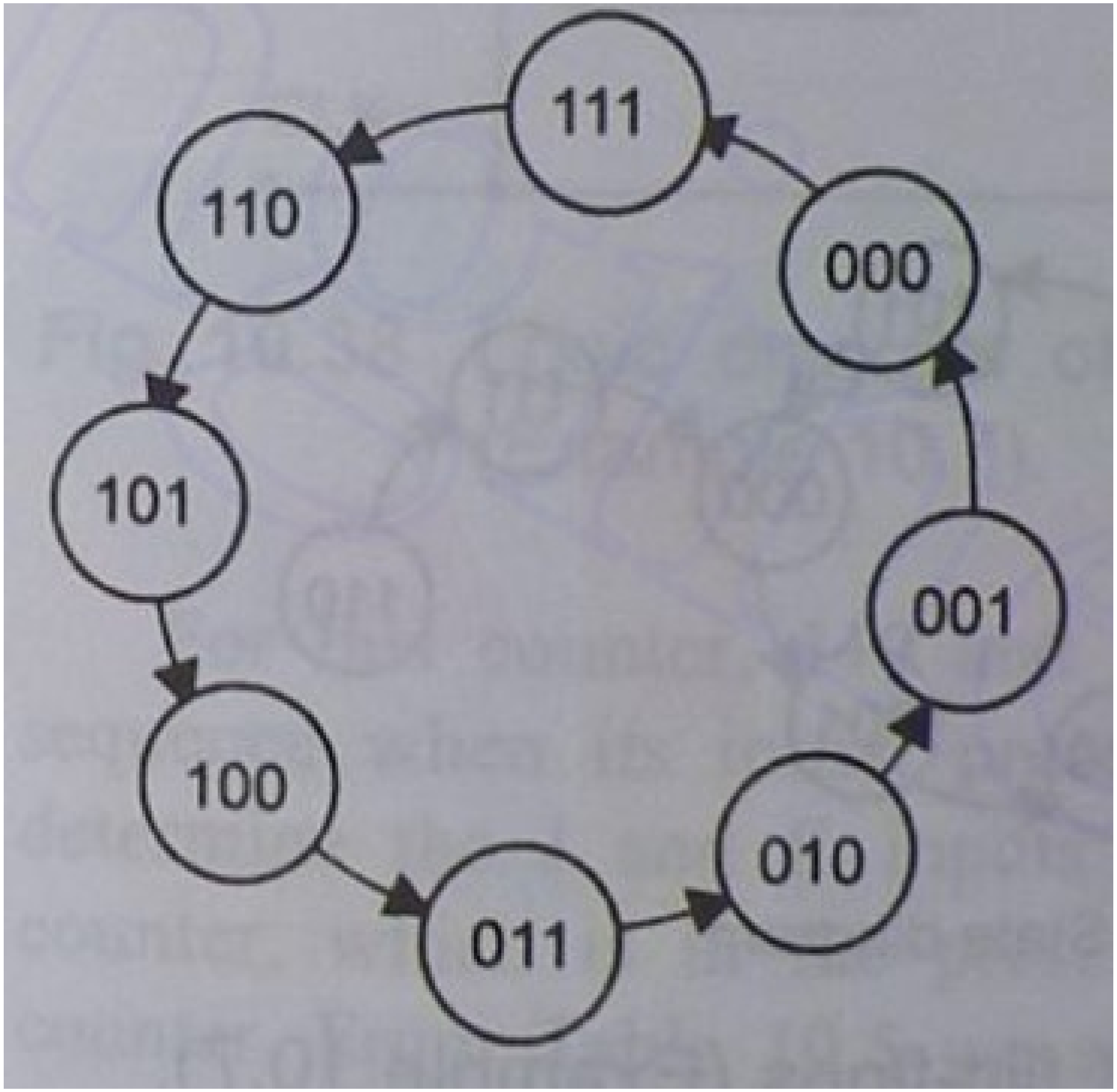
$K_2 = Q_1$

3 – Bit DOWN Counter

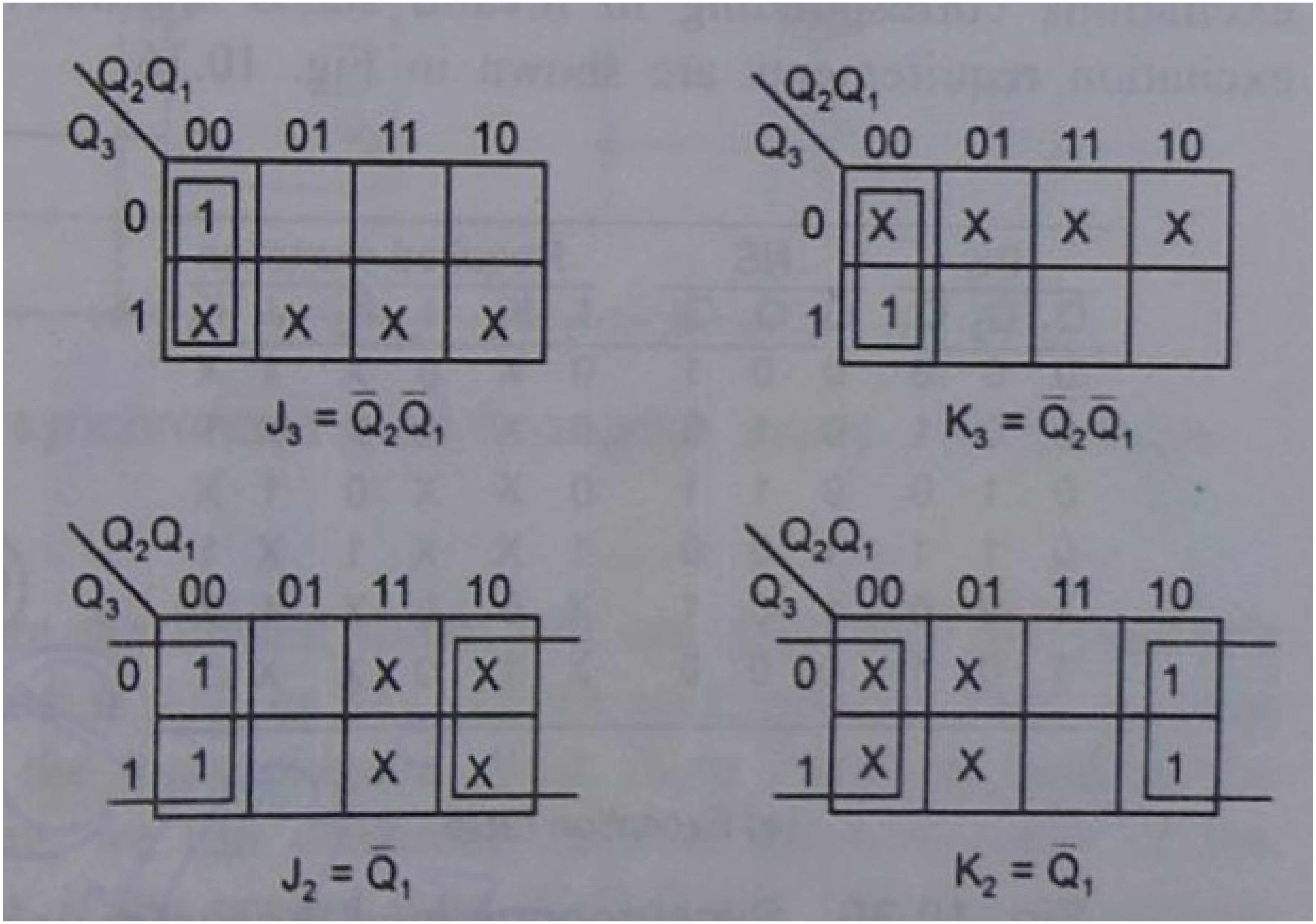


PS			NS			Required excitation					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K1
0	0	0	1	1	1	1	X	1	X	1	X
0	0	1	0	0	0	0	X	0	X	X	1
0	1	0	0	0	1	0	X	X	1	1	X
0	1	1	0	1	0	0	X	X	0	X	1
1	0	0	0	1	1	X	1	1	X	1	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	1	0	1	X	0	X	1	1	X
1	1	1	1	1	0	X	0	X	0	X	1

3 – Bit DOWN Counter



3 – Bit DOWN Counter



Q_3 \ Q_2Q_1	00	01	11	10
0	1			
1	X	X	X	X

$J_3 = \bar{Q}_2\bar{Q}_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	X	X
1	1			

$K_3 = \bar{Q}_2\bar{Q}_1$

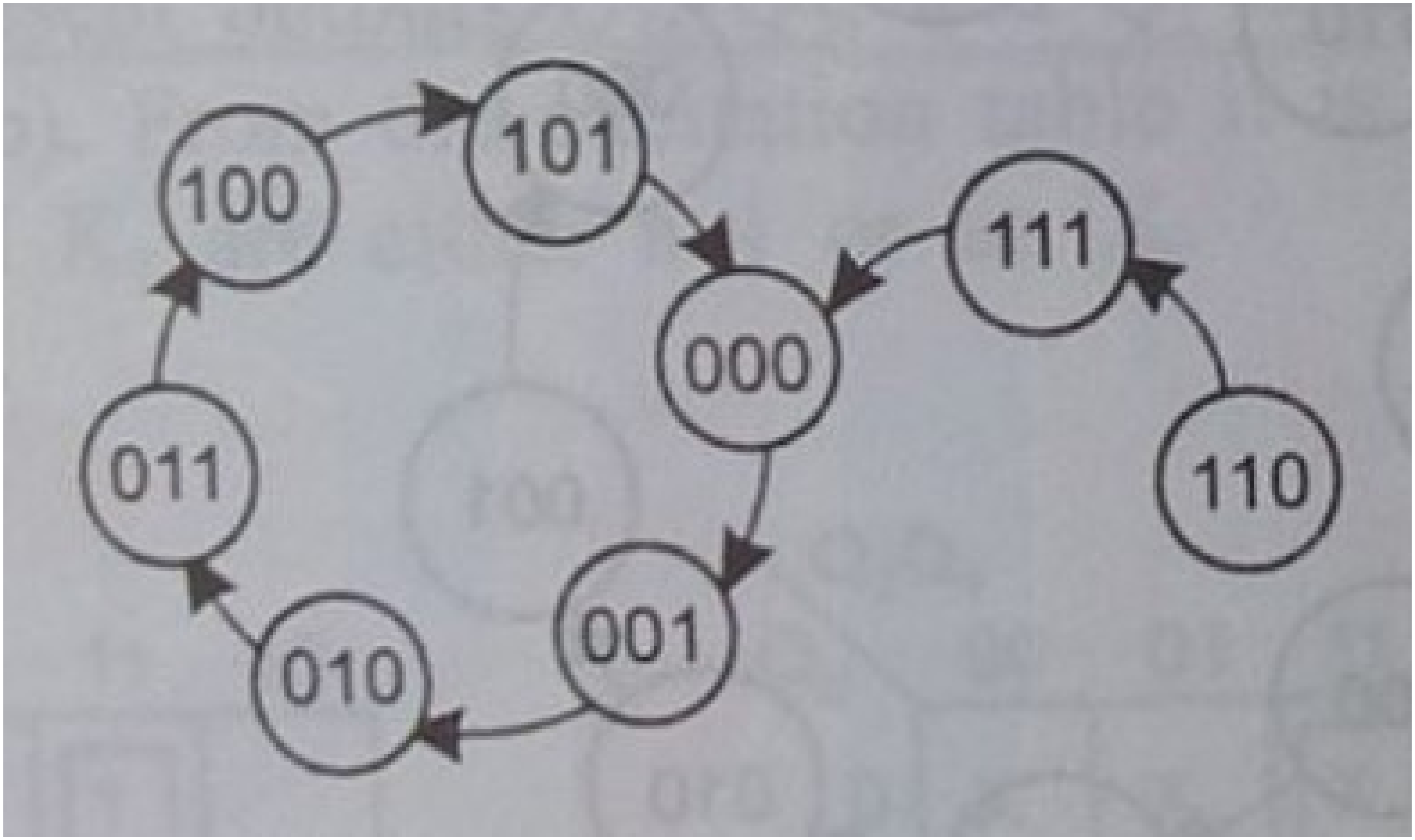
Q_3 \ Q_2Q_1	00	01	11	10
0	1		X	X
1	1		X	X

$J_2 = \bar{Q}_1$

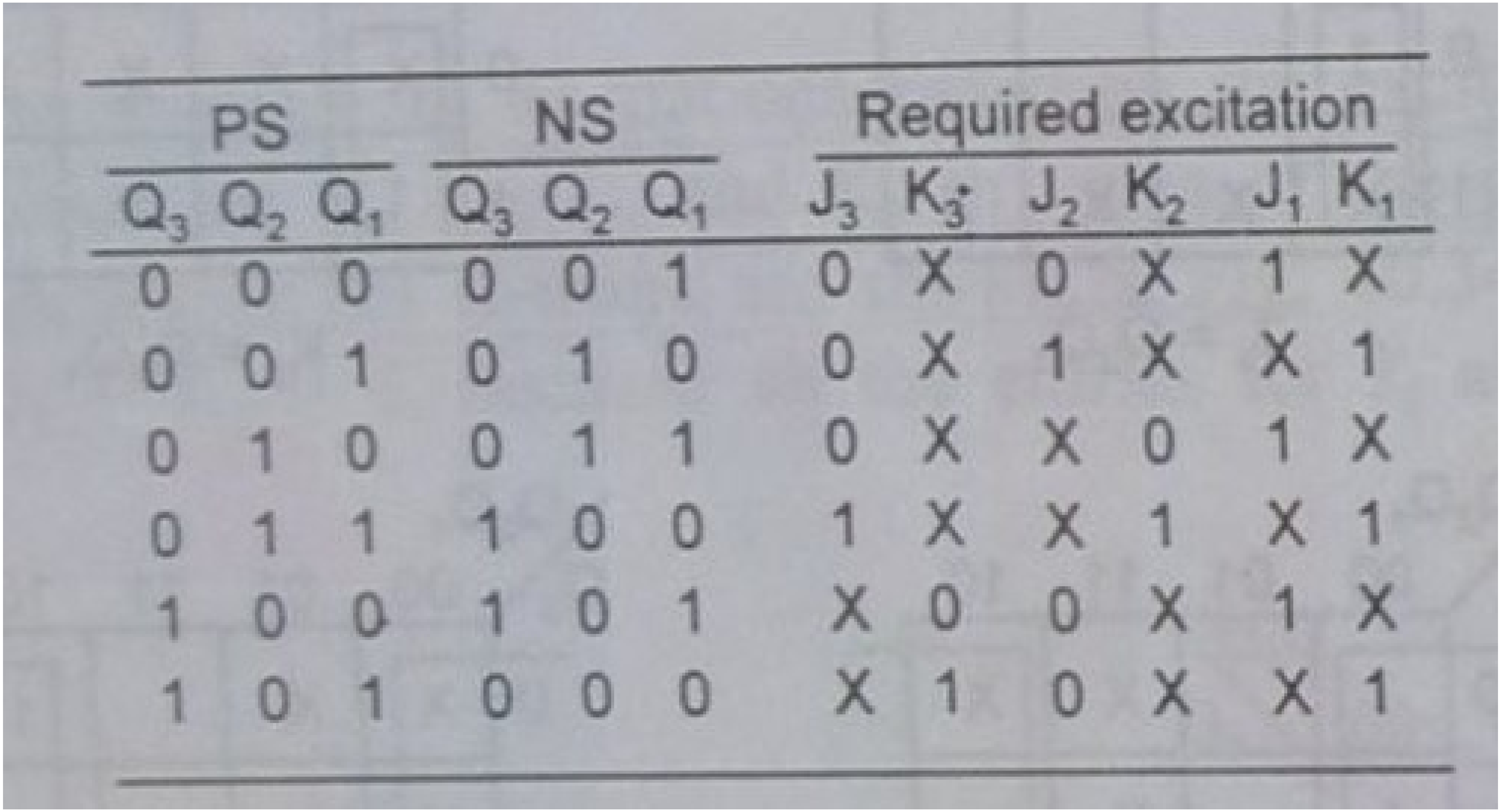
Q_3 \ Q_2Q_1	00	01	11	10
0	X	X		1
1	X	X		1

$K_2 = \bar{Q}_1$

Mod – 6 Counter

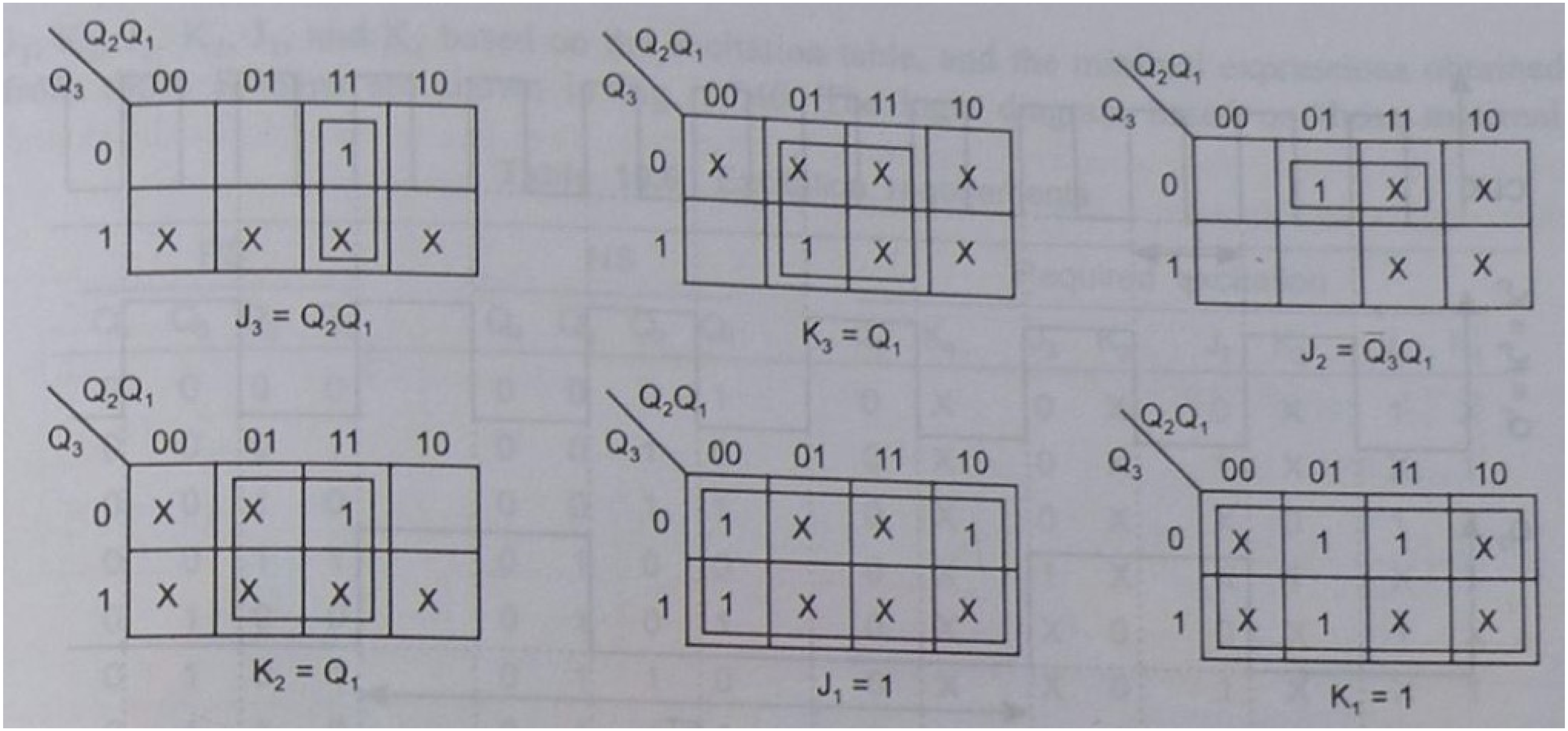


Next State Table



PS			NS			Required excitation					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

K-Map Simplifications



Q_3 \ Q_2Q_1	00	01	11	10
0			1	
1	X	X	X	X

$J_3 = Q_2Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	X	X
1		1	X	X

$K_3 = Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0		1	X	X
1			X	X

$J_2 = \bar{Q}_3Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0	X	X	1	
1	X	X	X	X

$K_2 = Q_1$

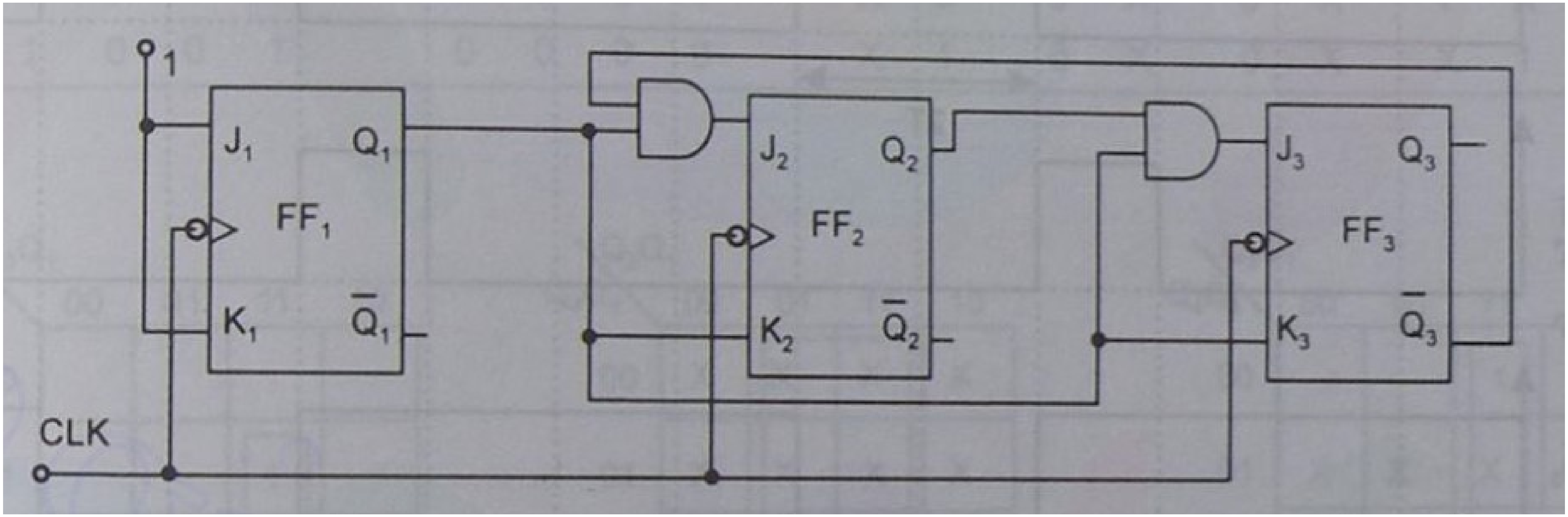
Q_3 \ Q_2Q_1	00	01	11	10
0	1	X	X	1
1	1	X	X	X

$J_1 = 1$

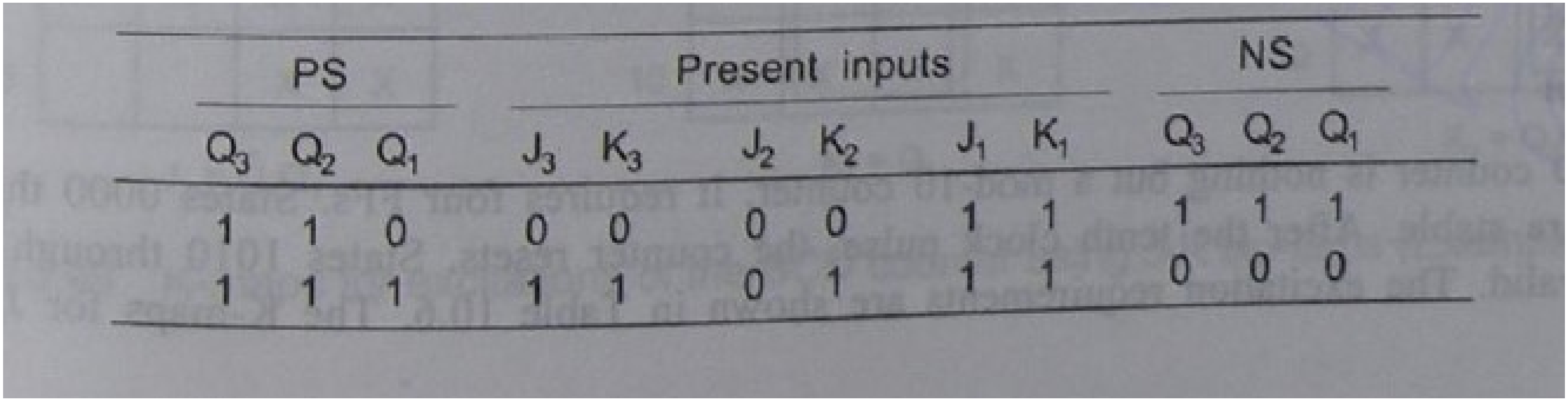
Q_3 \ Q_2Q_1	00	01	11	10
0	X	1	1	X
1	X	1	X	X

$K_1 = 1$

Connection Diagram



Check For Lockout



PS			Present inputs						NS		
Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1	Q_3	Q_2	Q_1
1	1	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	0	0	0

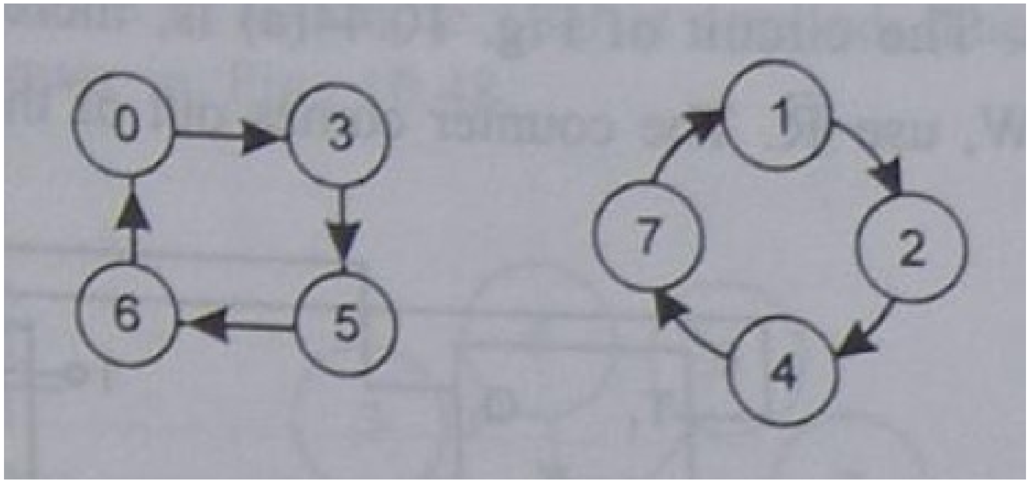
Random Sequence Counter

Design a counter using T Flip-Flops that goes through the states 0-3-5-6-0…

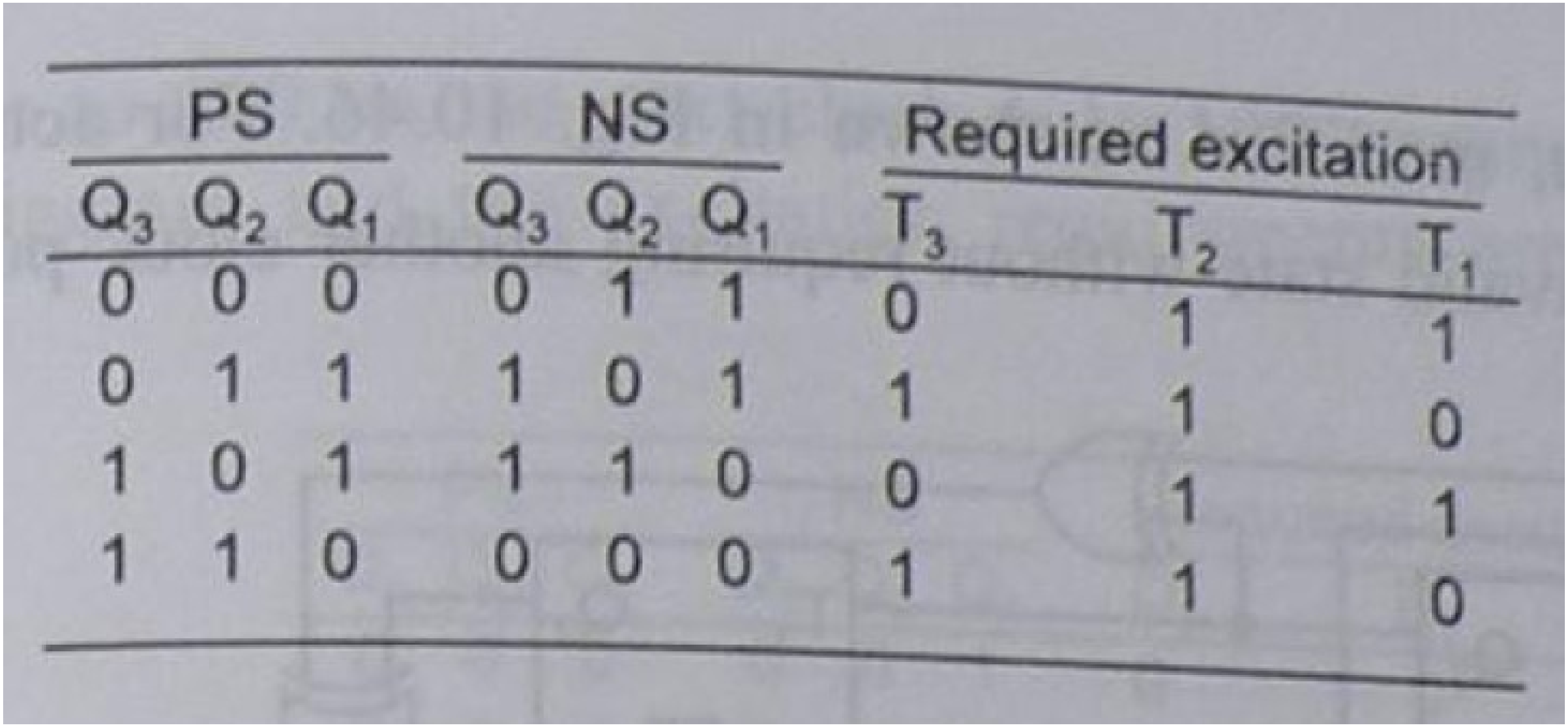
State Table

Valid States – 0,3,5,6

Invalid States – 1,2,4,7

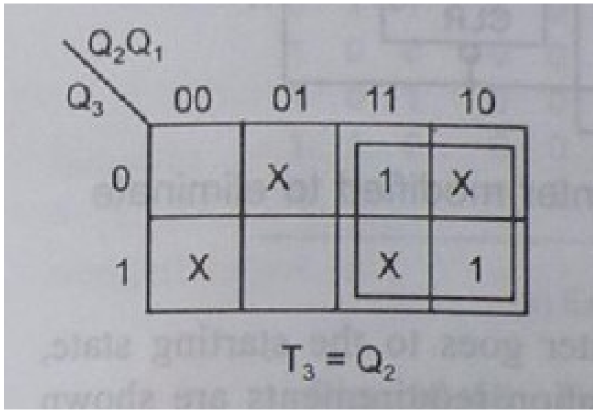


Next State Table



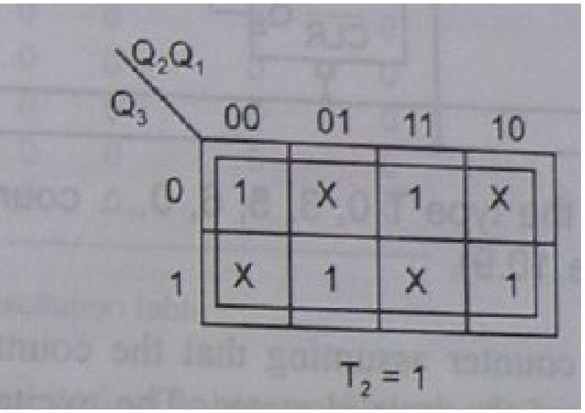
PS			NS			Required excitation		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	T_3	T_2	T_1
0	0	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	0
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0

K-Map Simplifications



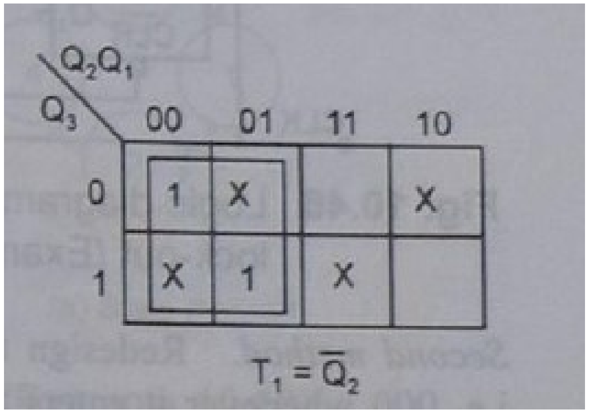
$Q_3 \backslash Q_2Q_1$	00	01	11	10
0		X	1	X
1	X		X	1

$T_3 = Q_2$



$Q_3 \backslash Q_2Q_1$	00	01	11	10
0	1	X	1	X
1	X	1	X	1

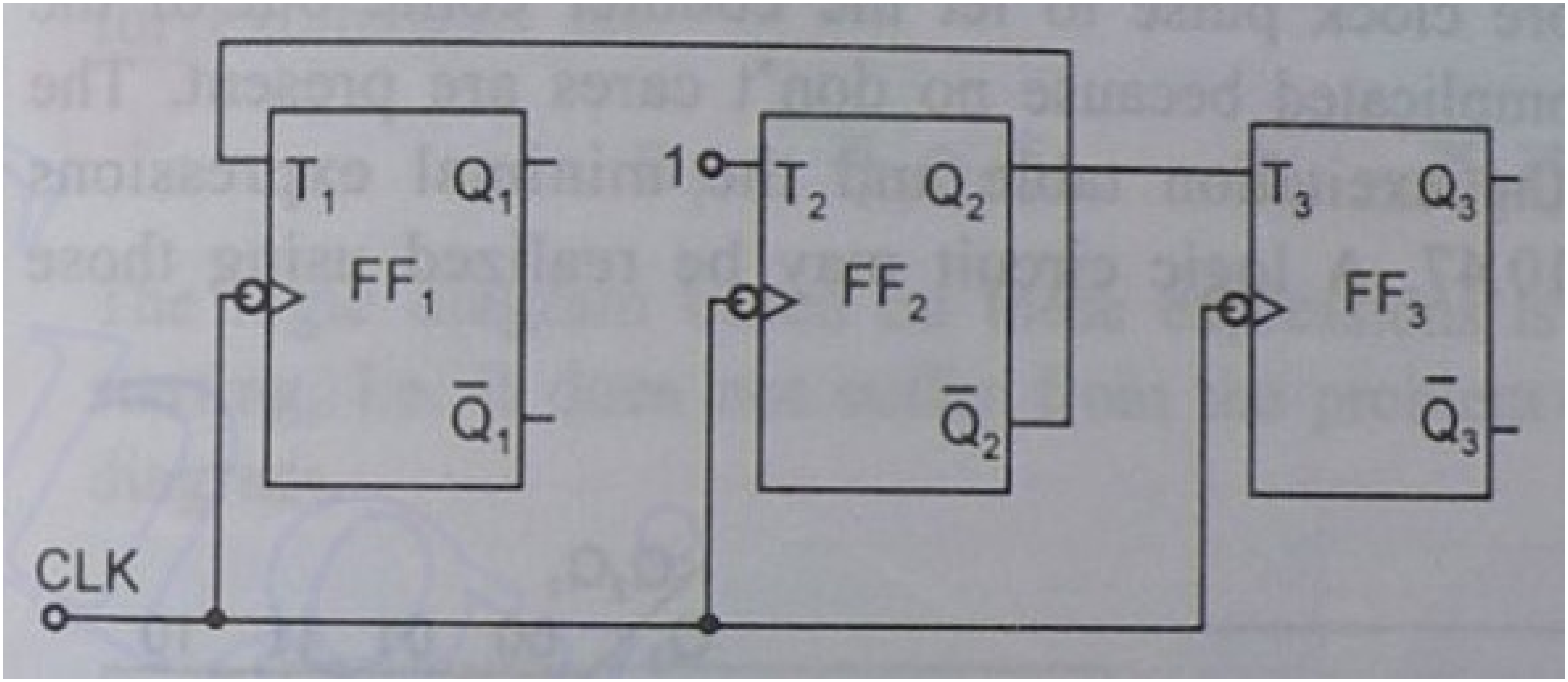
$T_2 = 1$



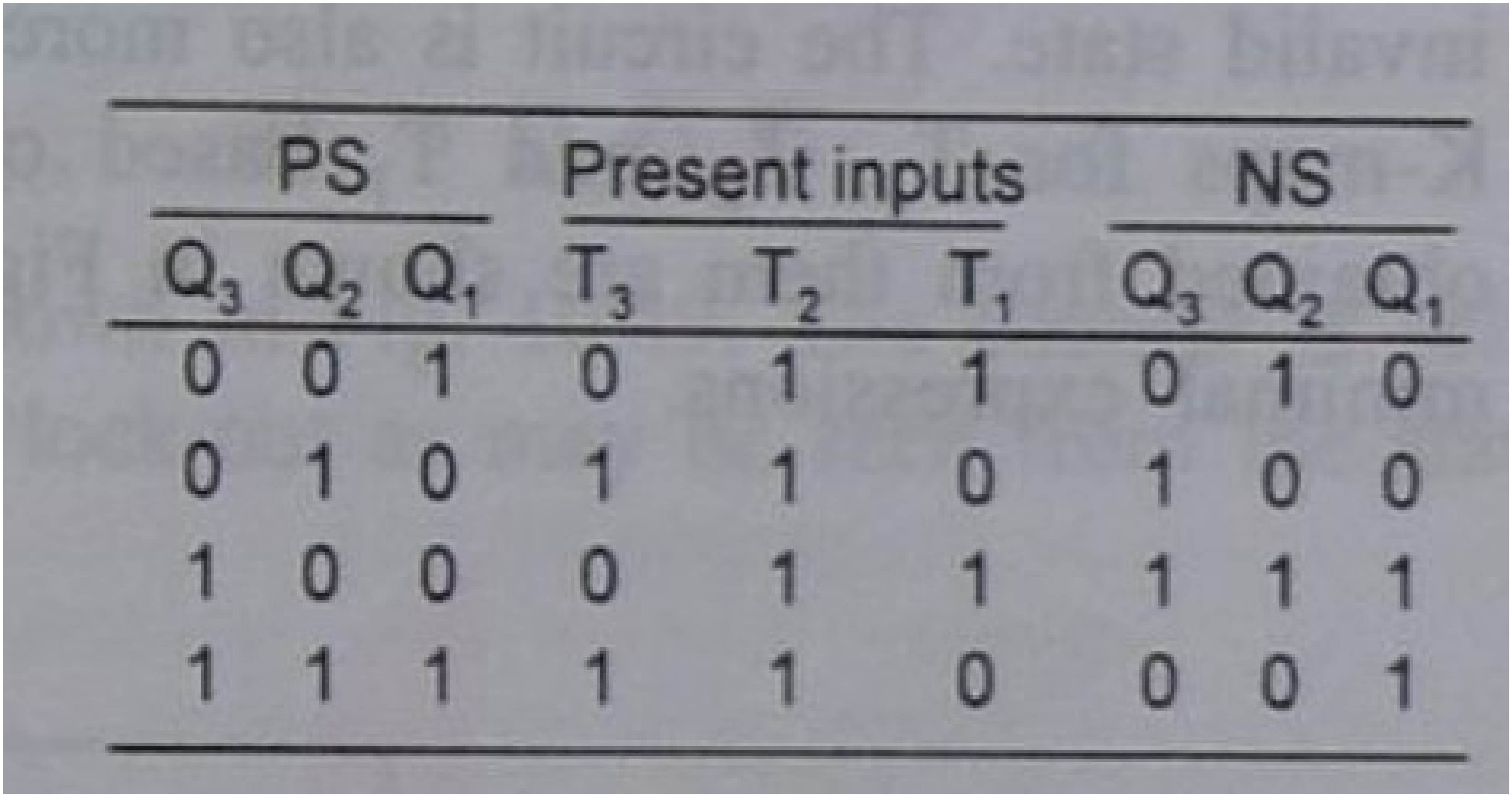
$Q_3 \backslash Q_2Q_1$	00	01	11	10
0	1	X		X
1	X	1	X	

$T_1 = \overline{Q}_2$

Connection Diagram

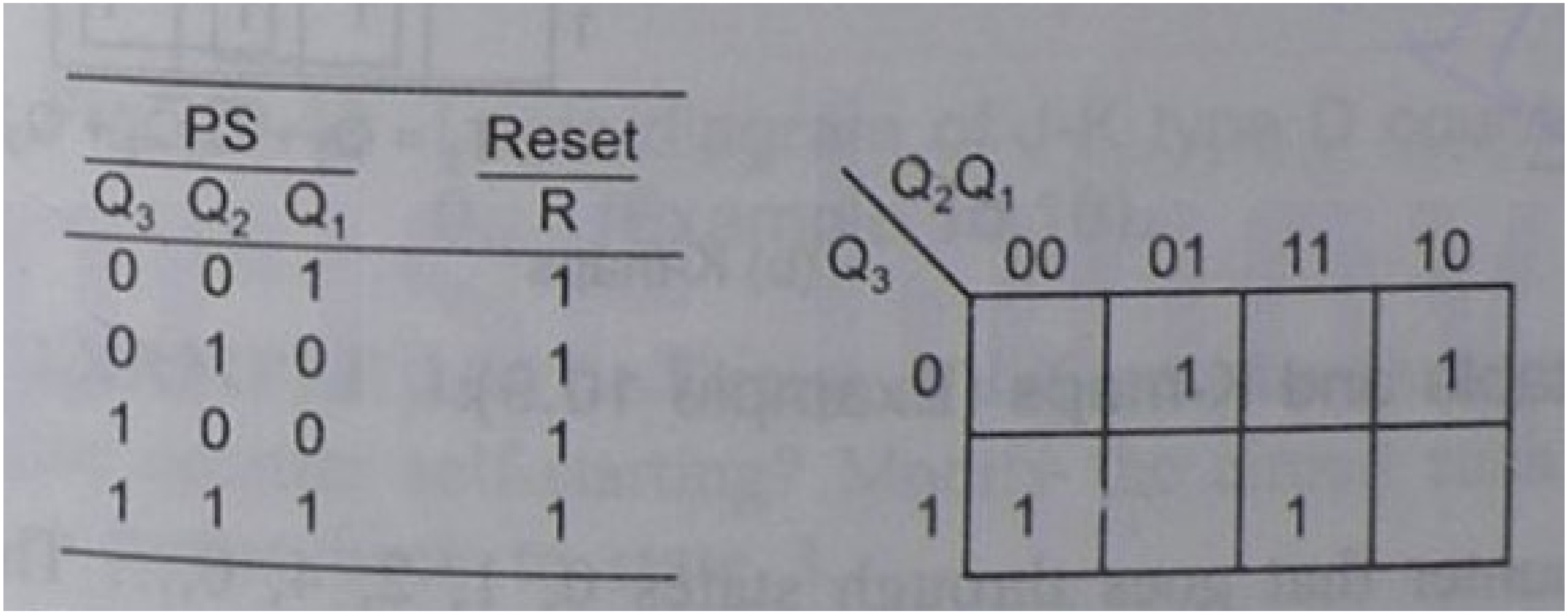


Check For Lockout



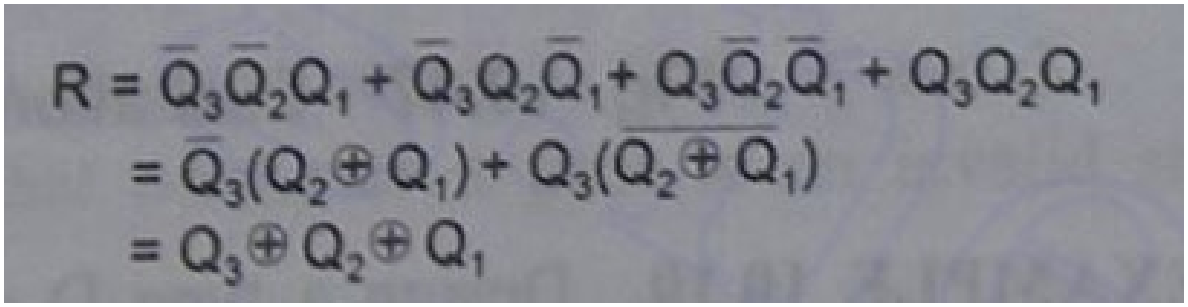
PS			Present inputs			NS		
Q_3	Q_2	Q_1	T_3	T_2	T_1	Q_3	Q_2	Q_1
0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
1	0	0	0	1	1	1	1	1
1	1	1	1	1	0	0	0	1

Elimination of Lockout (Method -1)



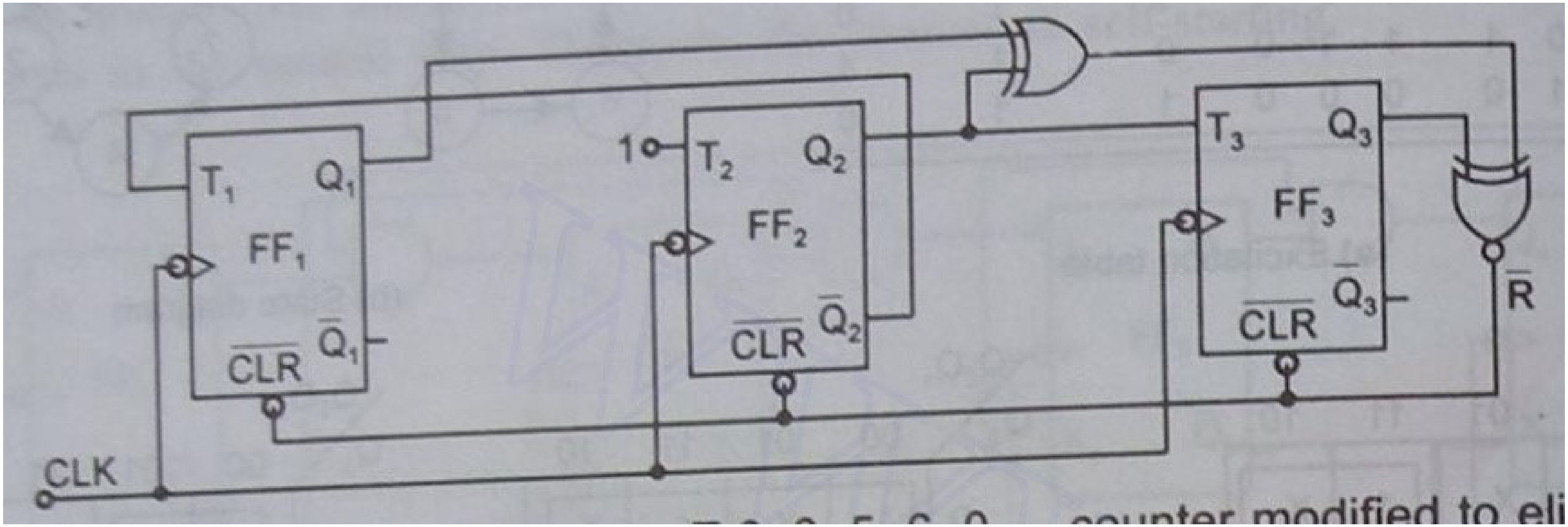
PS			Reset
Q_3	Q_2	Q_1	R
0	0	1	1
0	1	0	1
1	0	0	1
1	1	1	1

$$\begin{array}{c|cccc} Q_3 \backslash Q_2Q_1 & 00 & 01 & 11 & 10 \\ \hline 0 & & 1 & & 1 \\ 1 & 1 & & 1 & \end{array}$$

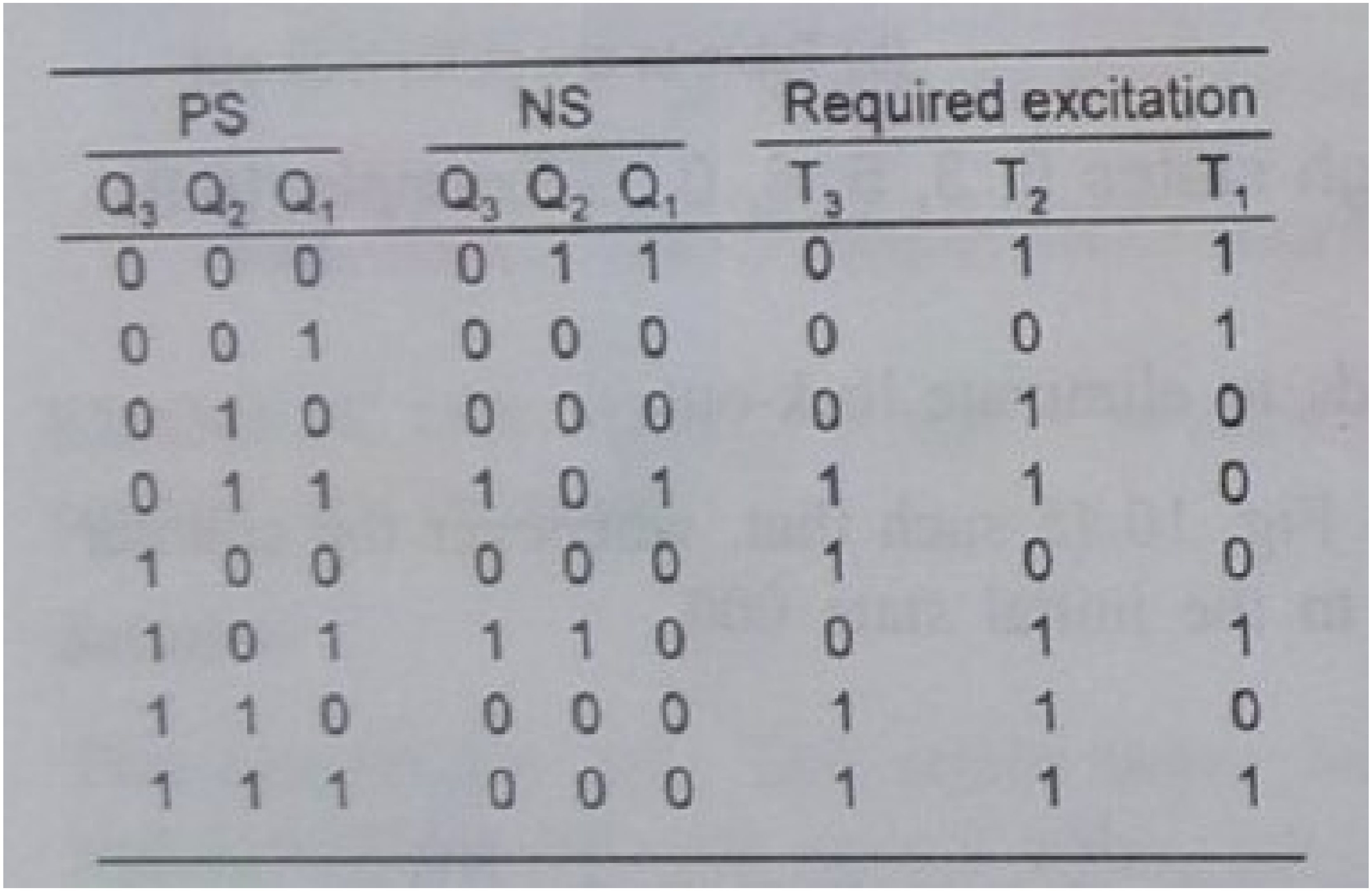


$$\begin{aligned} R &= \bar{Q}_3\bar{Q}_2Q_1 + \bar{Q}_3Q_2\bar{Q}_1 + Q_3\bar{Q}_2\bar{Q}_1 + Q_3Q_2Q_1 \\ &= \bar{Q}_3(Q_2 \oplus Q_1) + Q_3(\overline{Q_2 \oplus Q_1}) \\ &= Q_3 \oplus Q_2 \oplus Q_1 \end{aligned}$$

Modified Connection Diagram

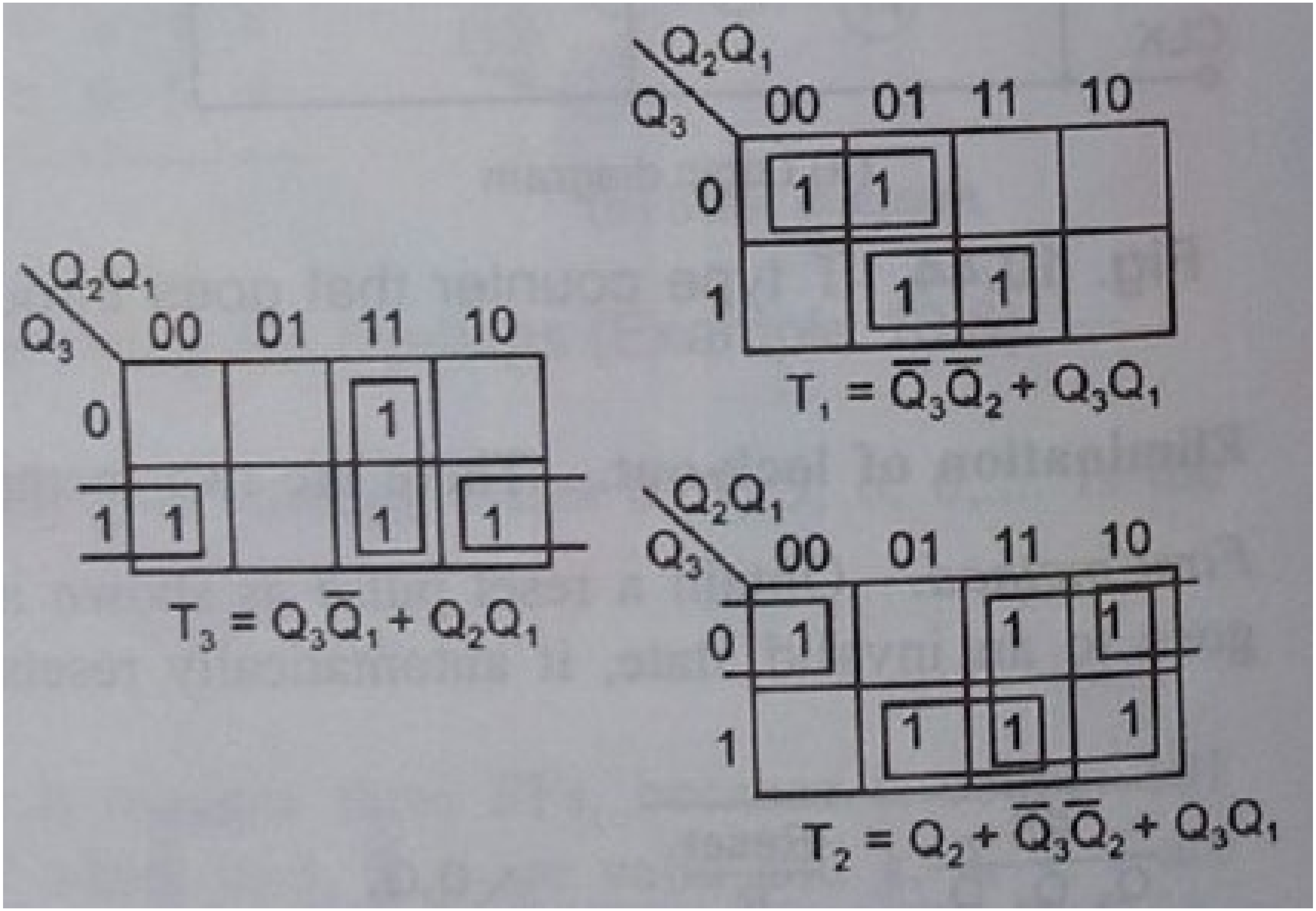


Elimination of Lockout (Method -2)



PS			NS			Required excitation		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	T_3	T_2	T_1
0	0	0	0	1	1	0	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0
0	1	1	1	0	1	1	1	0
1	0	0	0	0	0	1	0	0
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0
1	1	1	0	0	0	1	1	1

K-Map Simplification



Q_3 \ Q_2Q_1	00	01	11	10
0	1	1		
1		1	1	

$T_1 = \overline{Q}_3\overline{Q}_2 + Q_3Q_1$

Q_3 \ Q_2Q_1	00	01	11	10
0			1	
1	1		1	1

$T_3 = Q_3\overline{Q}_1 + Q_2Q_1$

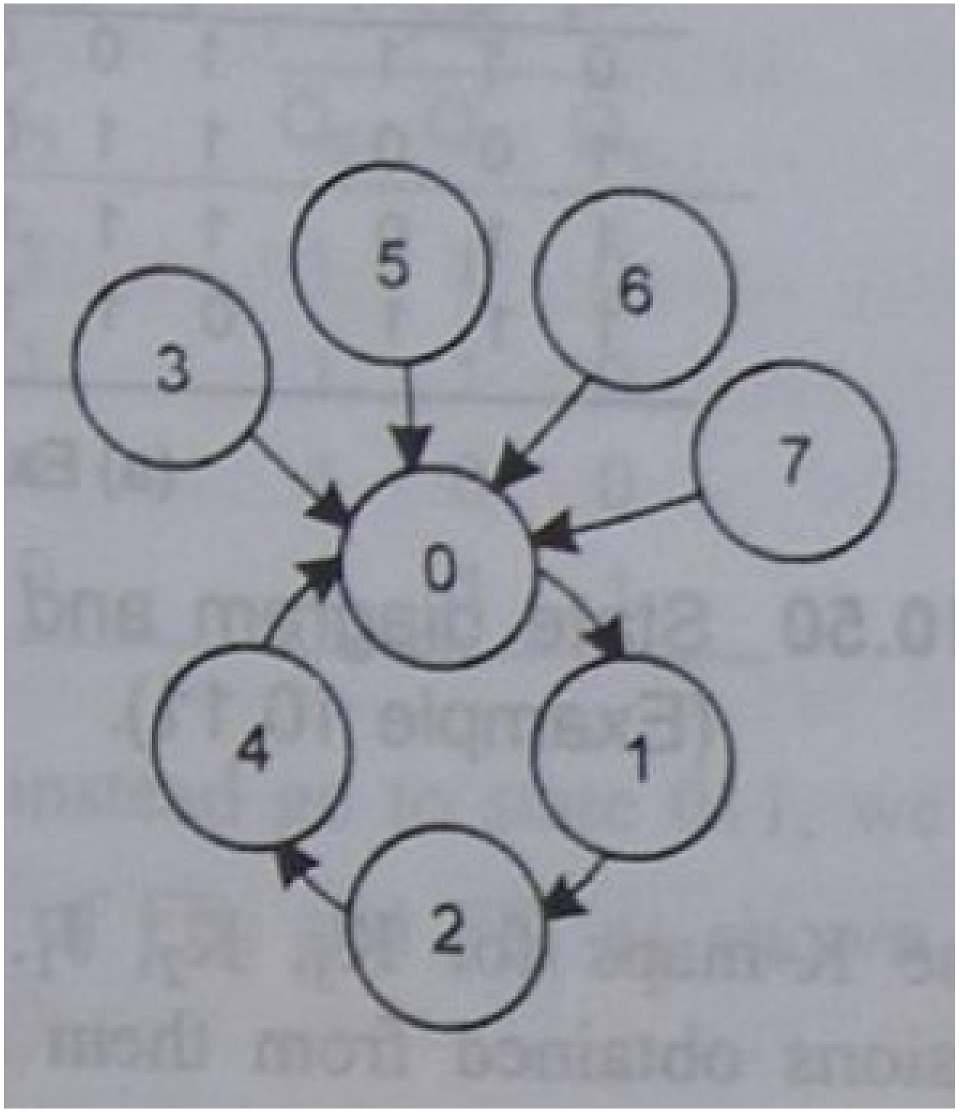
Q_3 \ Q_2Q_1	00	01	11	10
0	1		1	1
1		1	1	1

$T_2 = Q_2 + \overline{Q}_3\overline{Q}_2 + Q_3Q_1$

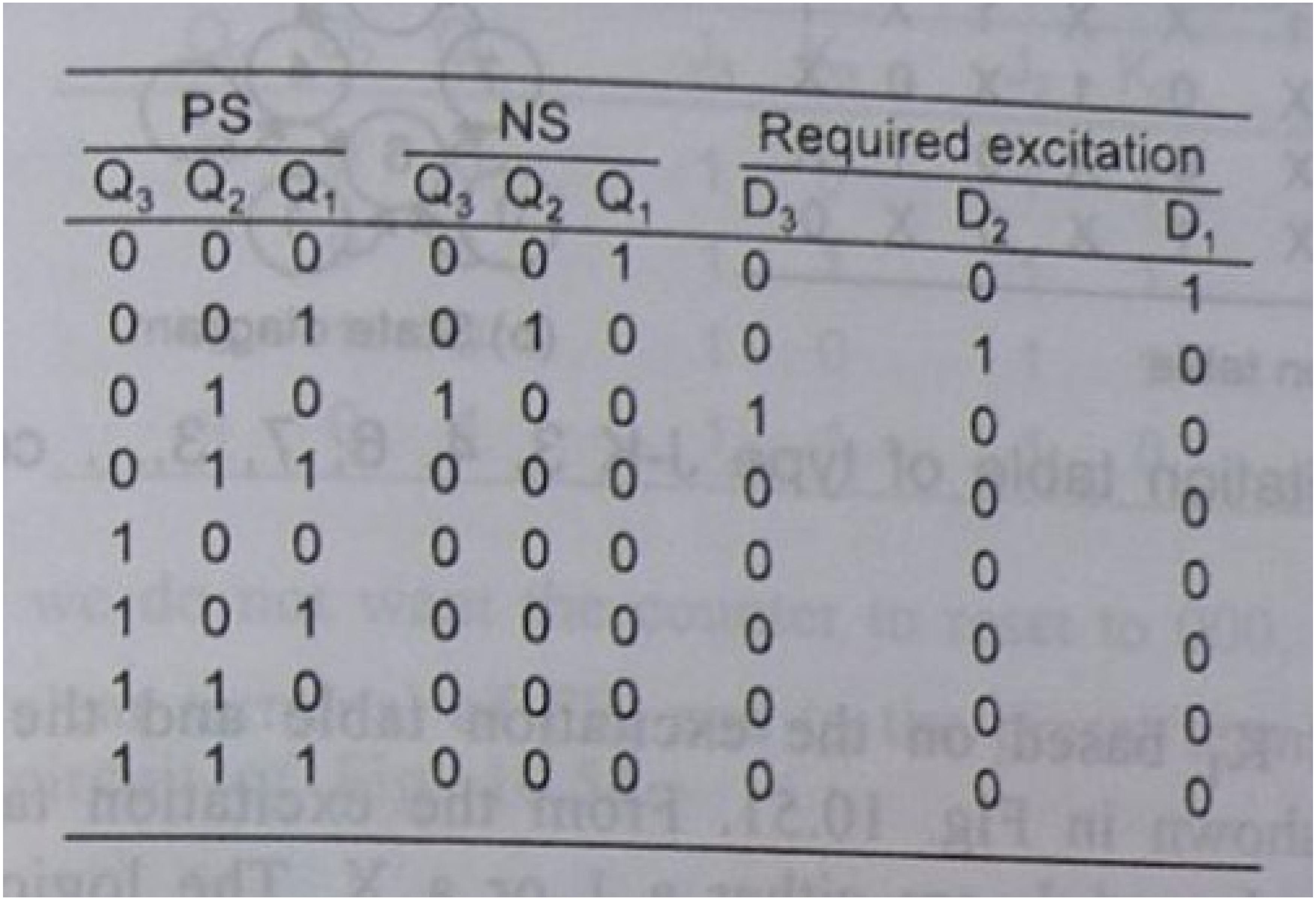
RSC - 2

Design a counter using D Flip-Flops which will count the sequence 0-1-2-4-0. If the counter is found in the undesired state, it must go to the initial (000) state on the very next clock pulse.

State Diagram



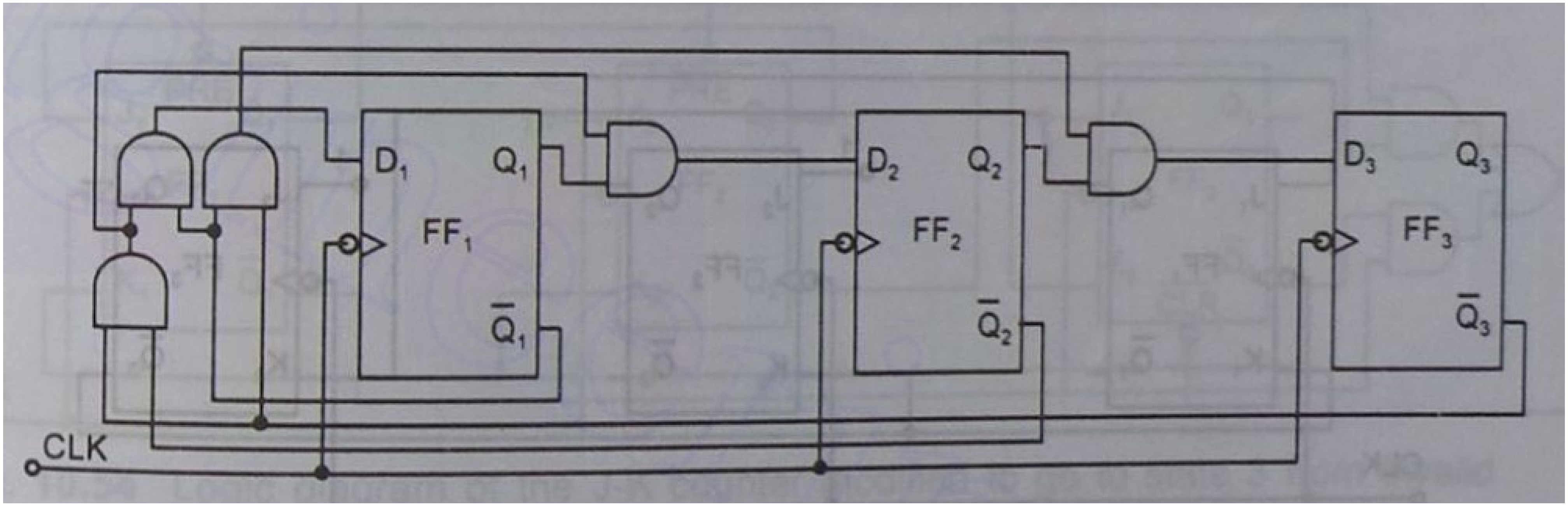
Next State Table



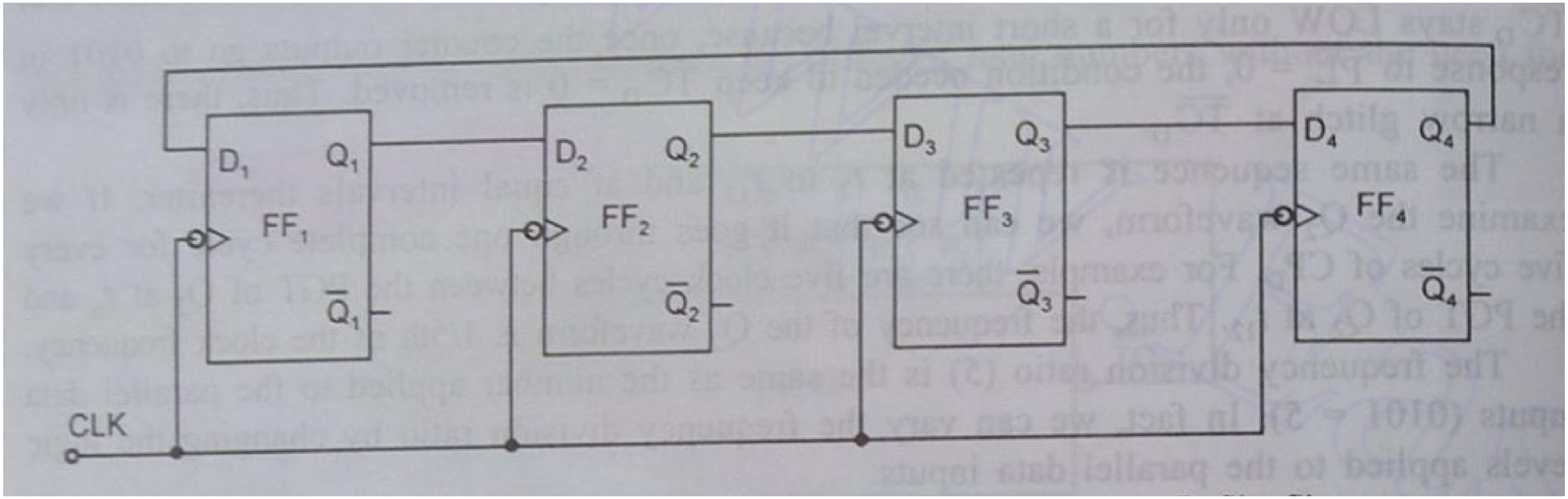
PS			NS			Required excitation		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	D_3	D_2	D_1
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	1	0	0
0	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Simplifications and Connection Diagram

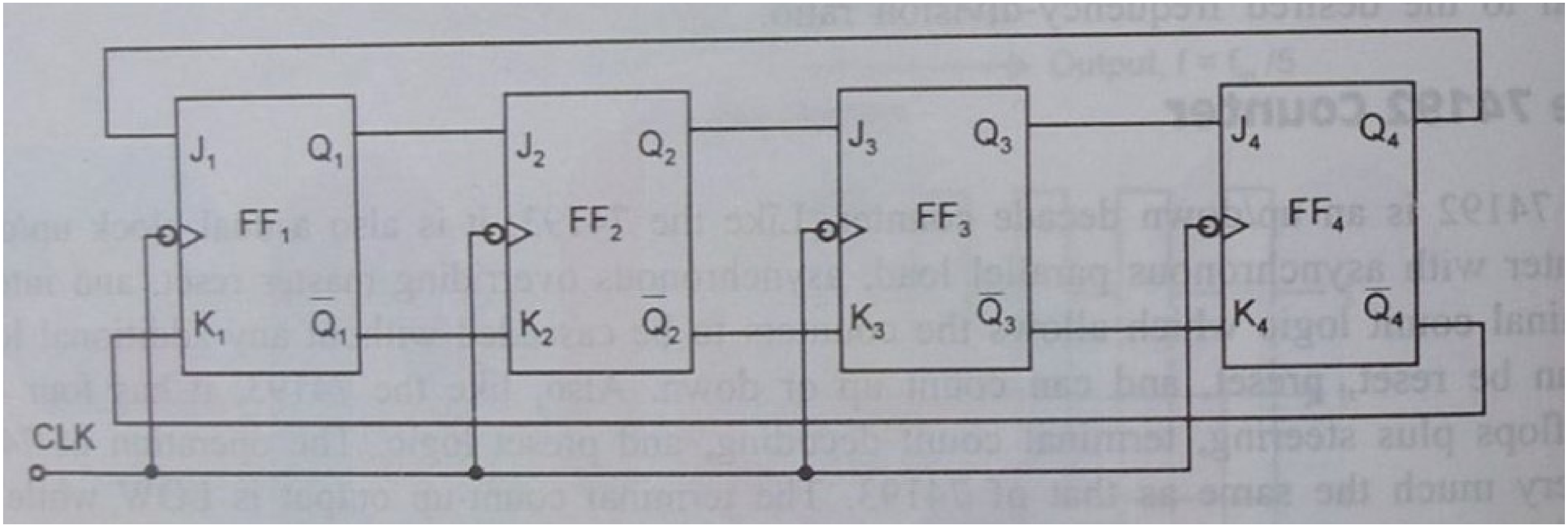
$$D_3 = \overline{Q}_3 Q_2 \overline{Q}_1; \qquad D_2 = \overline{Q}_3 \overline{Q}_2 Q_1; \qquad D_1 = \overline{Q}_3 \overline{Q}_2 \overline{Q}_1$$



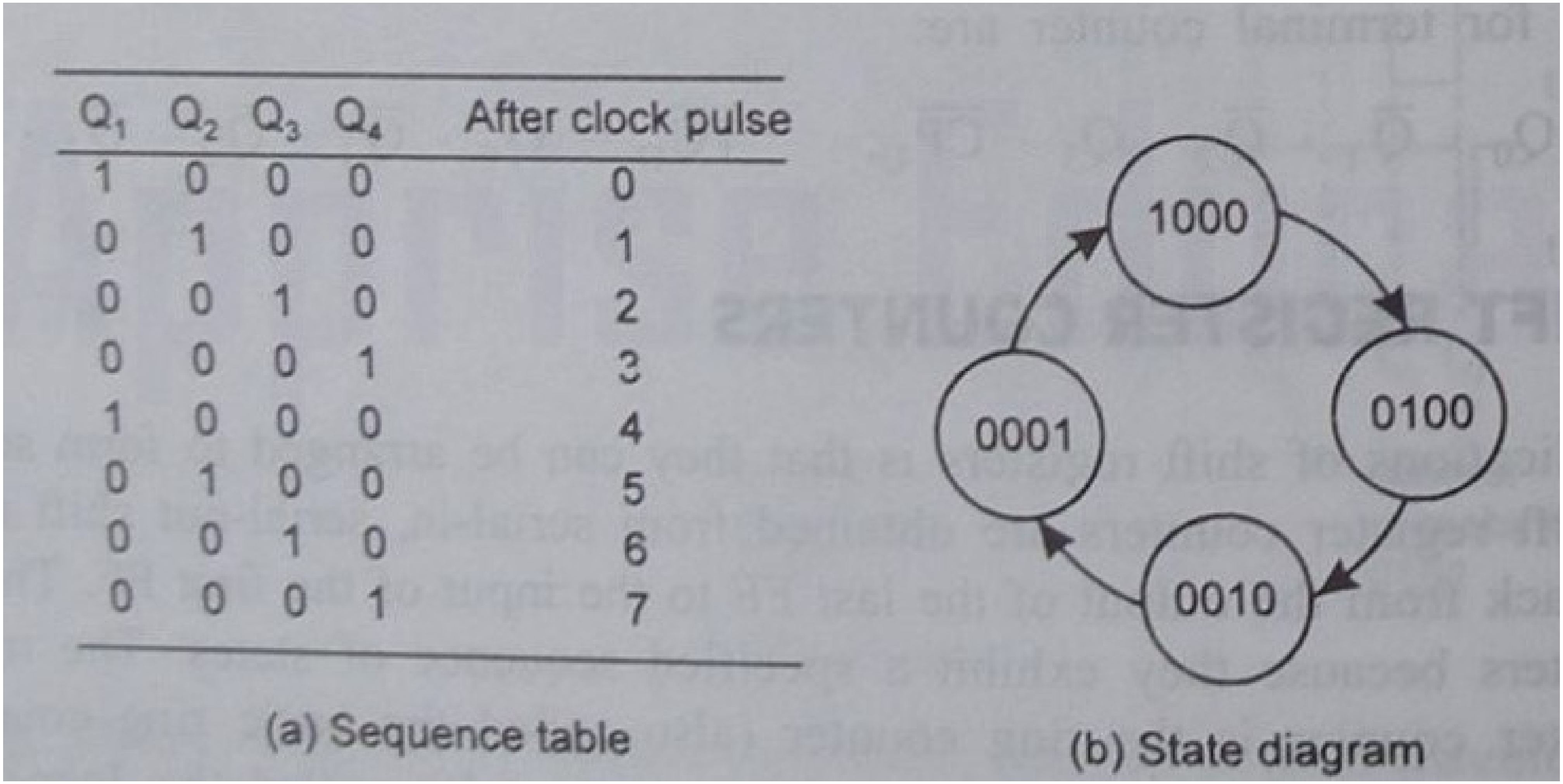
Ring Counter



Ring Counter



Ring Counter

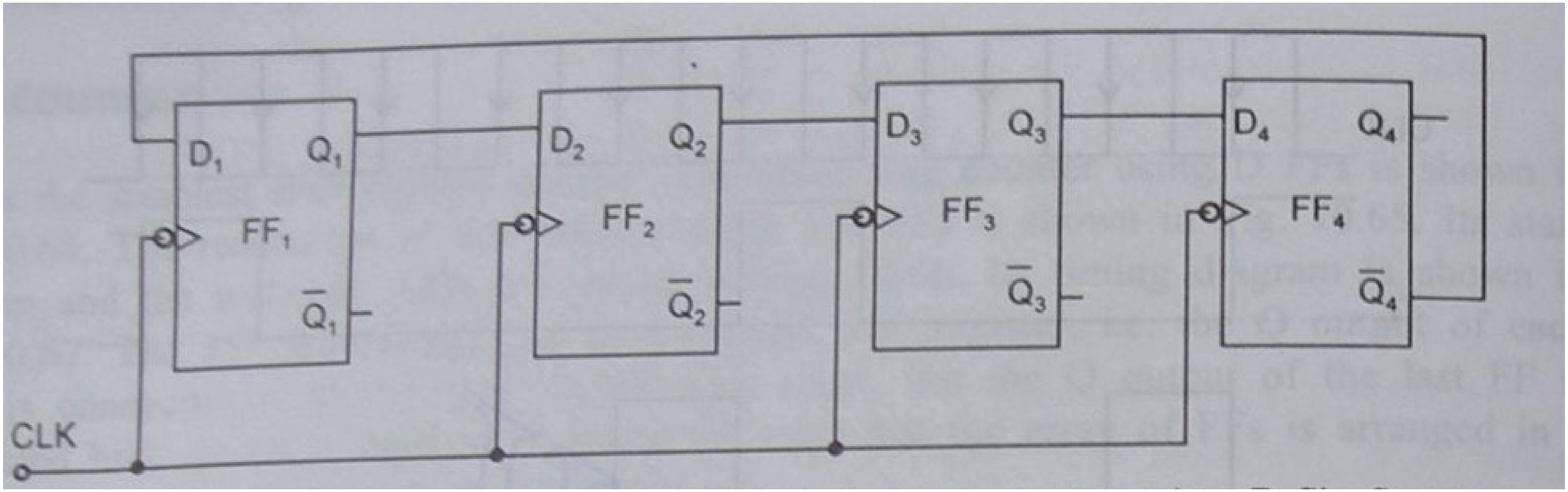


Q_1	Q_2	Q_3	Q_4	After clock pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7

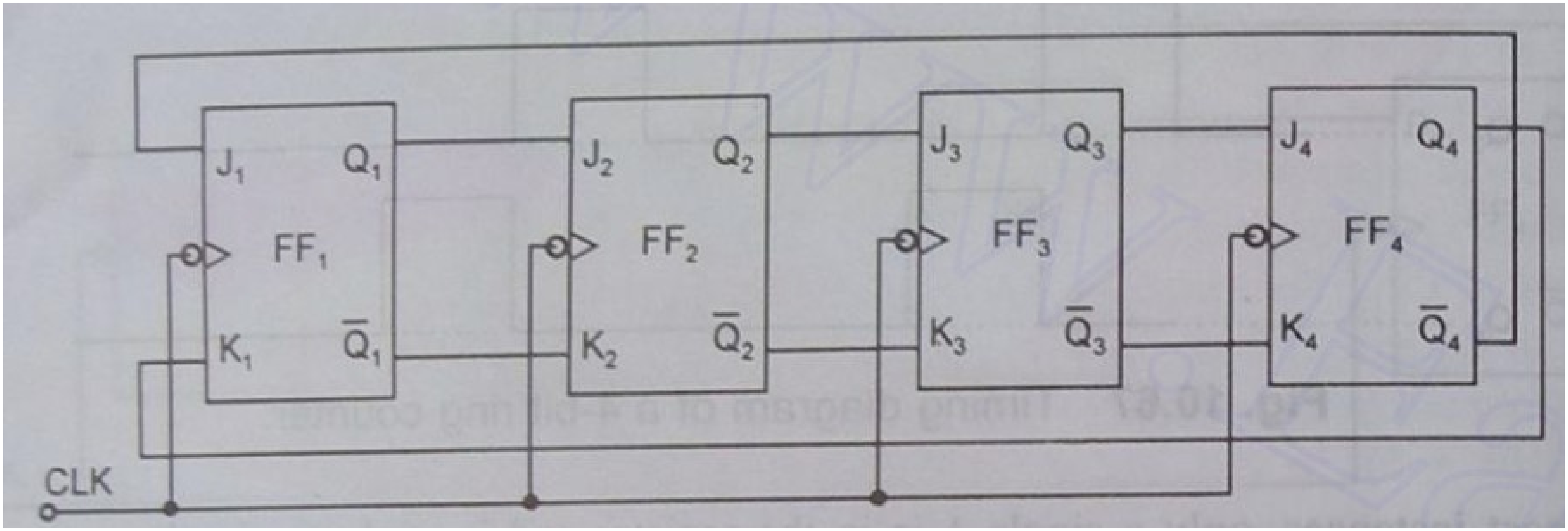
(a) Sequence table

(b) State diagram

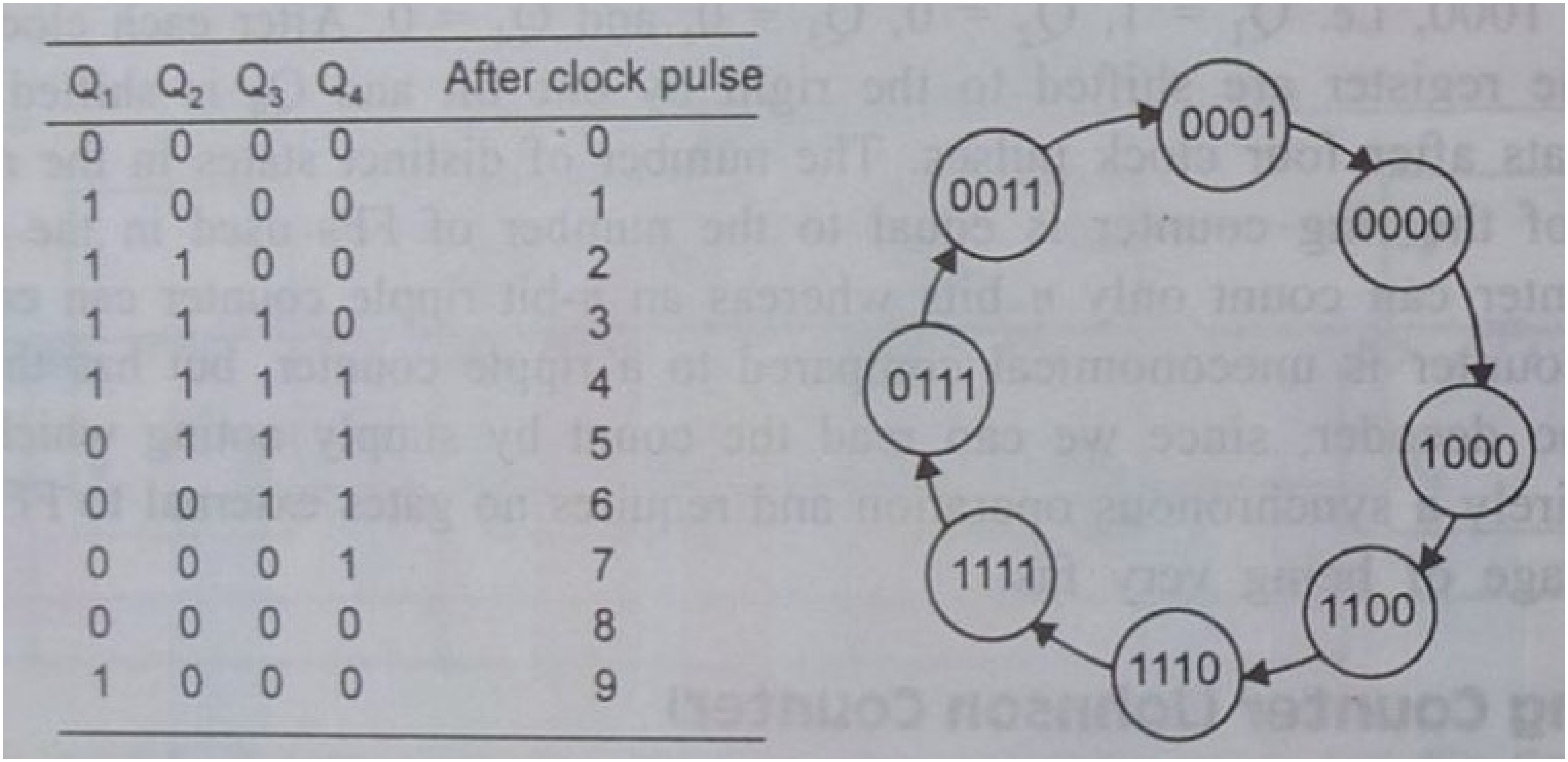
Twisted Ring Counter



Twisted Ring Counter



Twisted Ring Counter



Q_1	Q_2	Q_3	Q_4	After clock pulse
0	0	0	0	0
1	0	0	0	1
1	1	0	0	2
1	1	1	0	3
1	1	1	1	4
0	1	1	1	5
0	0	1	1	6
0	0	0	1	7
0	0	0	0	8
1	0	0	0	9

To B Continued…