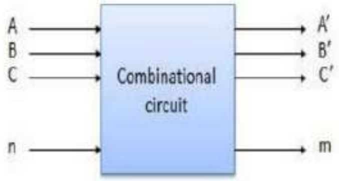
KTU NOTES

MODULE 4 – FLIP FLOPS

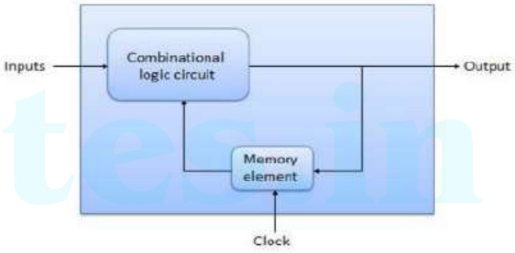
Combinational circuits

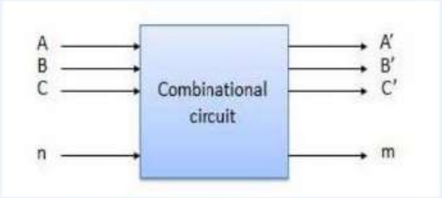
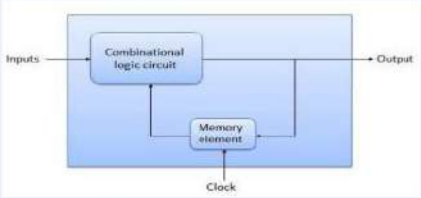
Output at any instant of time is entirely dependent upon the inputs present at that time



Sequential Circuit

- Sequential circuit consists of a combinational circuit to which memory elements are connected to form a feedback path.
- Memory elements are devices that are capable of storing binary information within them
- Binary information stored in the memory elements at any given time defines the state of the sequential circuit
- The external inputs, together with the present state of the memory elements determine the binary value at the output terminals
- The next state of the memory elements is also a function of the external inputs and the present state
- Sequential Circuit is specified by the time sequence of inputs, outputs and internal states



Combinational circuit	Sequential Circuit
Present Output depends only on Present input	Present Output depends on Present input and present state
It contains no Memory element	It contains Memory element
Its behavior is described by the set of output functions	Its behavior is described by the set of next state functions and the set of output functions
 A B C n Combinational circuit A' B' C' m	 Inputs Combinational logic circuit Output Memory element Clock
E.g.. Half Adder, Full Adder, Multiplexer	E.g.. Flip flop, counters, registers

Types of Sequential Circuits

- Flip flops
- Counters
- Registers

Flip Flops

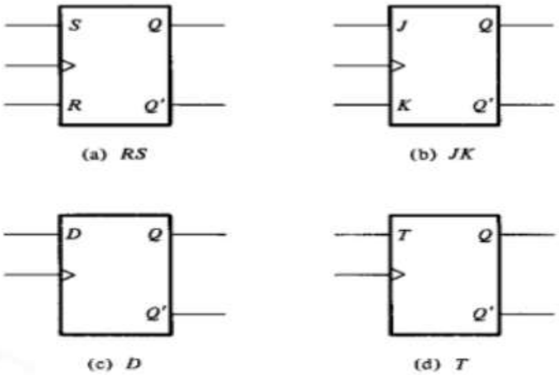
- The basic memory element in a sequential circuit is a flip flop.
- The flip flop can store a 1 or 0 indefinitely and is said to possess 2 stable states.
- A flip flop circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch states.

Types of Flip flops

- SR Flip flops: ‘S’- Set, ‘R’- Reset
- D Flip flop: ‘D’- Delay
- JK Flip flop: ‘J’- Jack, ‘K’- Kilby
- T Flip flop: ‘T’- Toggle
- Master Slave JK Flip flop

Graphic Symbols

- The input letter symbols in each diagram designate the type of flip flop such as SR, JK, D or T
- The clock pulse input is recognized in the diagram from the arrow head shaped symbol (dynamic indicator symbol)- denotes that the flip flop responds to a positive edge transition of the clock
- The presence of a small circle outside the block along the dynamic indicator designates a negative transition for triggering the flip flop
- The letter symbol C is used for the clock input when the flip flop responds to a pulse level rather than a pulse transition
- The output of the flip flop are marked with the letter symbol Q and Q’ within the block
- The flip flop may be assigned a different variable name even though Q is written inside the block. Letter symbol for flip flop output is marked outside the block along the output line



(a) RS

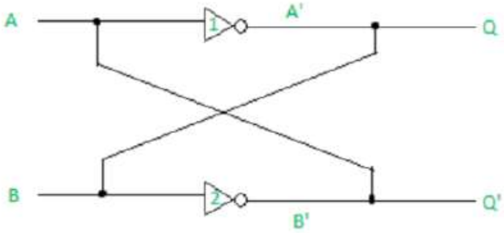
(b) JK

(c) D

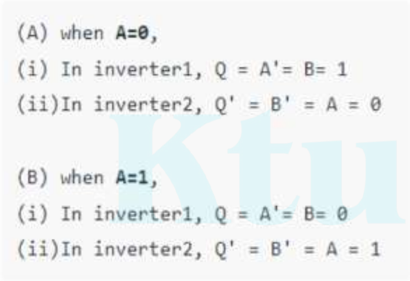
(d) T

1- bit memory cell

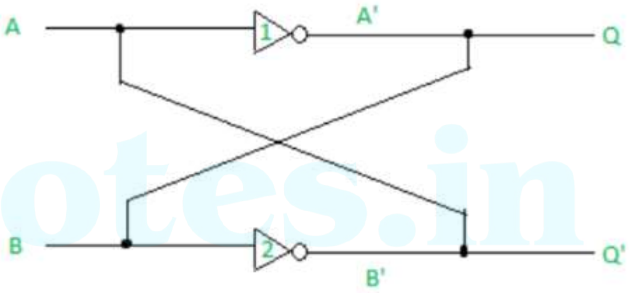
- Basic building block for various types of flip flops
- **One Bit memory** cell is also called **Basic Bistable element.**
- It has two cross-coupled inverters, 2 outputs Q and Q'.
- It is called 'Bistable' as the circuit has two stable states logic 0 and logic 1.



1- bit memory cell (contd.)



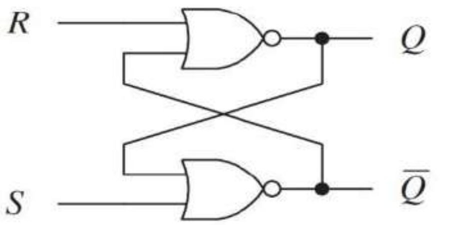
```
(A) when A=0,
(i) In inverter1, Q = A'= B= 1
(ii)In inverter2, Q' = B' = A = 0

(B) when A=1,
(i) In inverter1, Q = A'= B= 0
(ii)In inverter2, Q' = B' = A = 1
```



Some key points:

- The 2 outputs are always complementary.
- The circuit has 2 stable states. when Q=1, it is **Set state**. when Q=0, it is **Reset state**.
- The circuit can store 1-bit of digital information and so it is called one-bit memory cell.
- The one-bit information stored in the circuit is locked or latched in the circuit. This circuit is also called **Latch**.
- Inverter-based flip-flop is just for understanding the working but it doesn't have any practical uses as there is no provision for applying any inputs

RS Latch using NOR gates



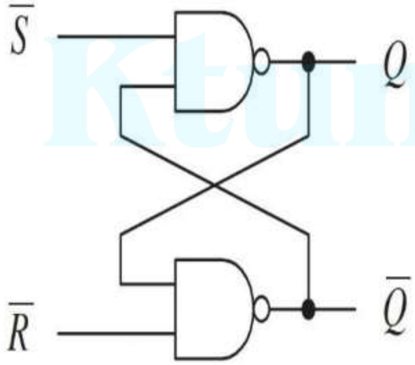
Q = (R + Q')'
Q' = (S + Q)'

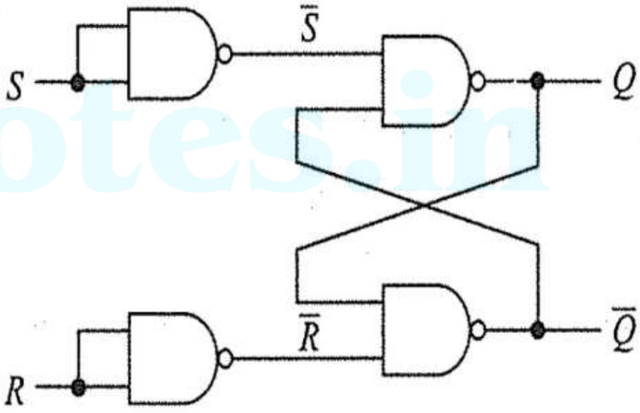
SR Latch using NAND gates

An SR latch can also be designed by cross coupling of two NAND gates

In SR Latch using NOR gates,
Q = (R + Q')' = (R'.Q)
Q' = (S + Q)' = (S'.Q')

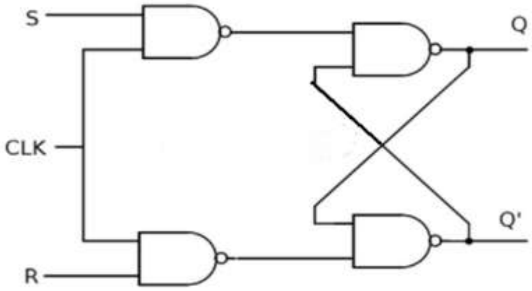
In SR Latch using NAND gates,
Q' = (R'.Q)'
Q = (S'.Q')'

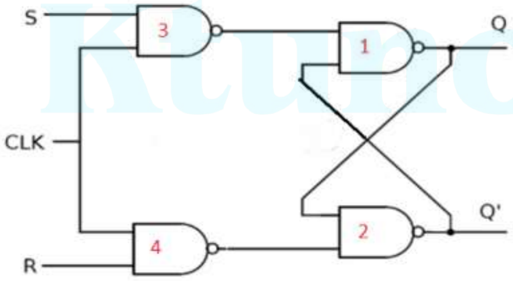




SR Flip – Flops

- SR Latch is limited to asynchronous mode of operation
- In many digital systems, it is desirable that the circuit responds only during some prescribed time, determined with an enable or clock input. With this clock, the circuit operates synchronously with all others in the system. These circuits are then referred to as flip flops
- SR Flip flop circuit is formed by adding 2 NAND gates to a NAND based SR latch.
- A clock pulse is given as input to both the extra NAND gates.

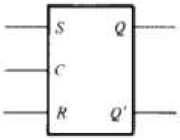




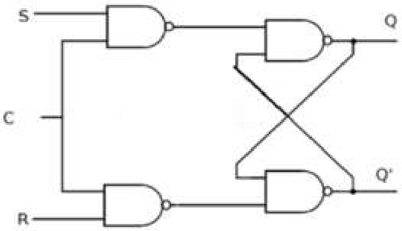
S	R	Q_{n+1}	State
0	0	Q_n	NO CHANGE
0	1	0	RESET
1	0	1	SET
1	1	?	FORBIDDEN

- The outputs of NAND gates 3 and 4 stay at logic 1 level as long as the CLK input remains at 0
- When the input goes to 1, the information at S and R input is allowed to reach the output.

SR Flip – Flops



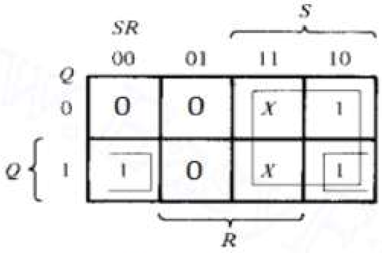
a. Graphic Symbol



c. Logic Diagram

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

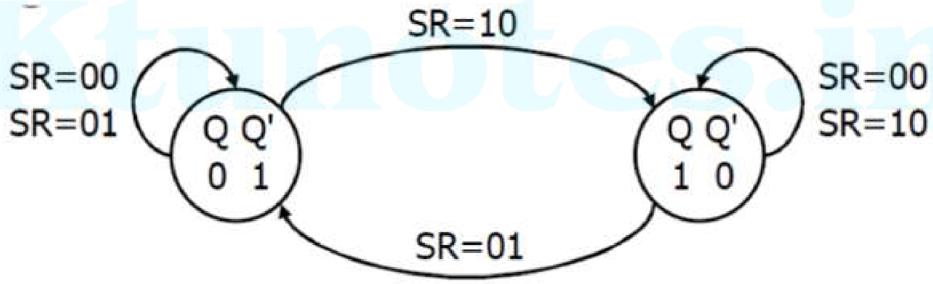
b. Characteristic table



$Q_{n+1} = S + R'Q$

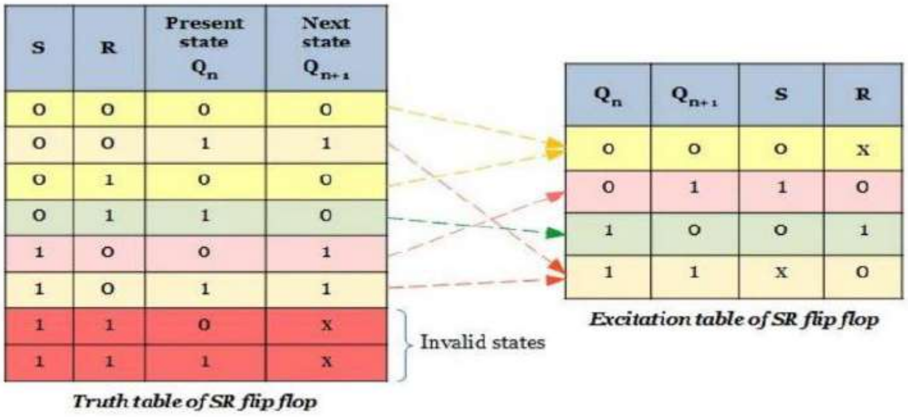
$SR = 0$

d. Characteristic equation



State Transition Diagram

Excitation Table of SR Flip flop



S	R	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

Invalid states

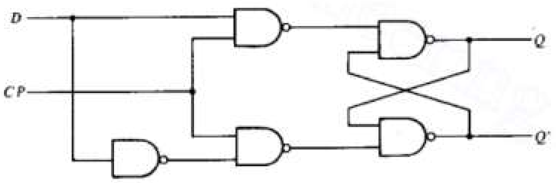
Truth table of SR flip flop

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

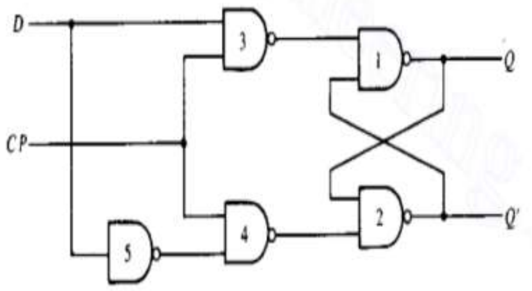
Excitation table of SR flip flop

D Flip flop

- Eliminates the undesirable condition of the indeterminate (invalid/ forbidden) state in the RS flip flop
- Ensures that the inputs S and R are never 1 at the same time
- D flip flop has only 2 inputs- D and CP
- D goes directly to S input, its complement goes to CP input



- As long as pulse input (CP) is at 0, outputs of gates 3 and 4 are at 1 level and the circuit cannot change state, regardless of the value of D.
- The D input is sampled when CP = 1
- If D = 1, then Q = 1 (circuit becomes SET state)
- If D = 0, then Q = 0 (circuit becomes RESET/CLEAR state)



- Next state of the flip flop is independent of the present state since Q(t+1) = D whether Q = 0 or Q = 1
- The binary information present at the input of the D flip flop is transferred to the Q output when the CP input is enabled
- Input pulse will transfer the value of input D into the output of the flip flop independent of the value of the output before the pulse was applied

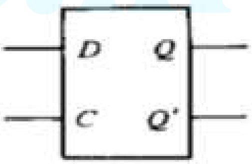
Q	D	$Q(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1

Characteristic table

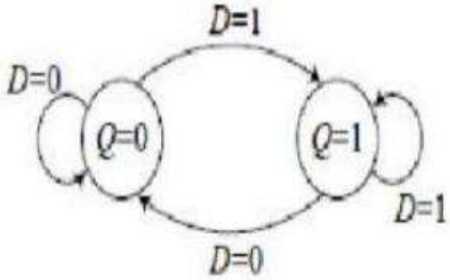
Q \ D	0	1
0		1
1		1

$Q(t+1) = D$

Characteristic equation

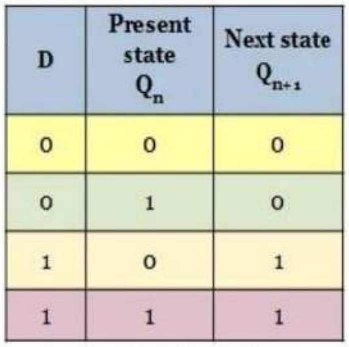


Graphic symbol



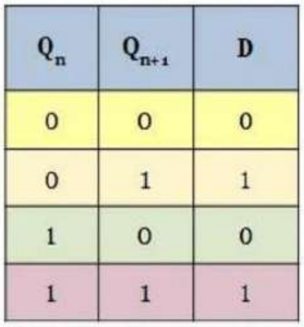
State- transition diagram

Excitation Table of D Flip flop



D	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Truth table of D flip flop



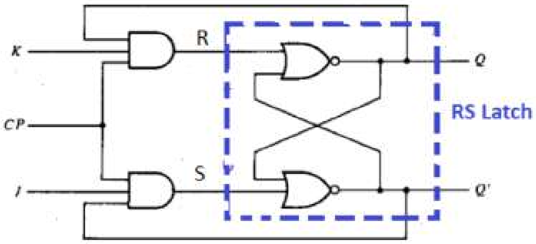
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

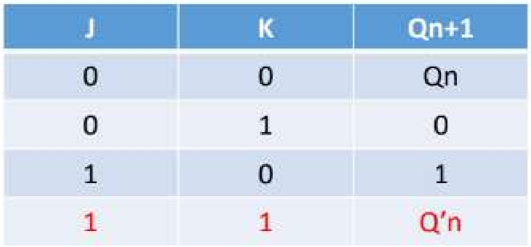
Excitation table of D flip flop

JK Flip Flop

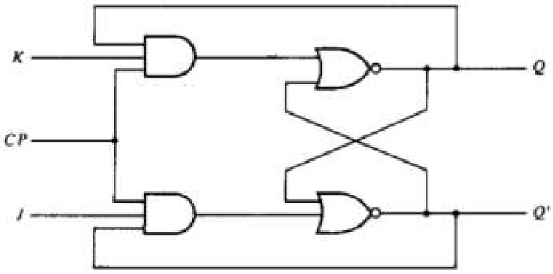
- It is a refinement of the RS flip flop in that the indeterminate state of the RS flip flop is defined in the JK type.
- The inputs, J and K, behave like inputs S and R, to set and clear the flip flop respectively
- When both inputs J and K are equal to 1, the flip flop switches to its complement state (if Q = 1, it switches to Q = 0 and vice versa)

- JK flip flop is constructed with two cross coupled NOR gates and two AND gates
- Output Q is ANDed with K and CP inputs
- Output Q' is ANDed with J and CP inputs

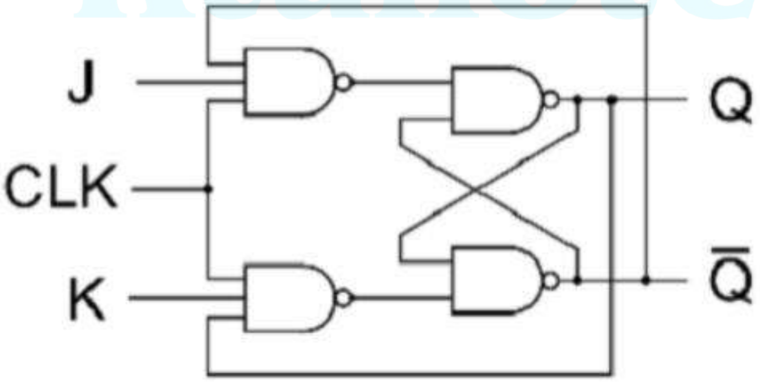


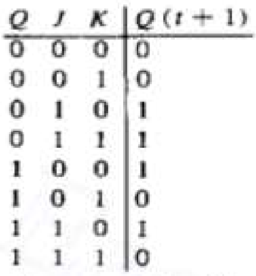


J	K	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	Q'n



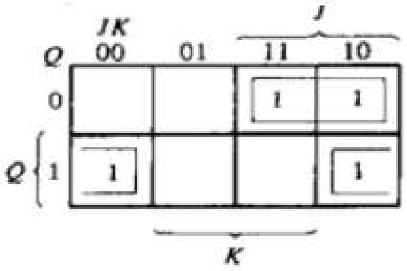
JK FF using NAND gates





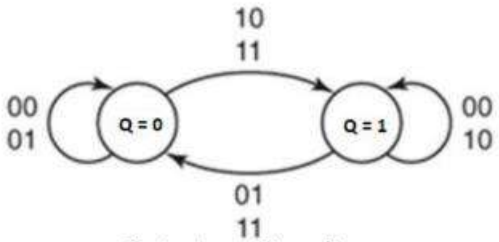
Q	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Characteristic table



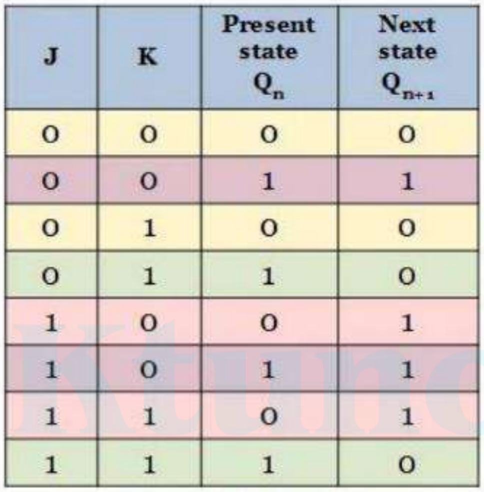
$$Q(t+1) = JQ' + K'Q$$

Characteristic equation



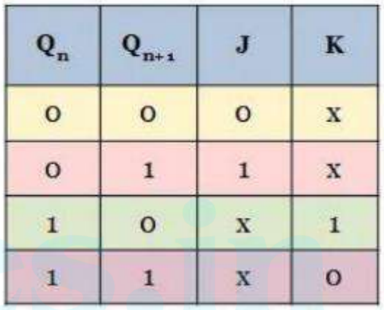
State transition diagram

Excitation Table of JK Flip flop



J	K	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Truth table of JK flip flop



Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation table of JK flip flop

Limitation of JK Flip flop

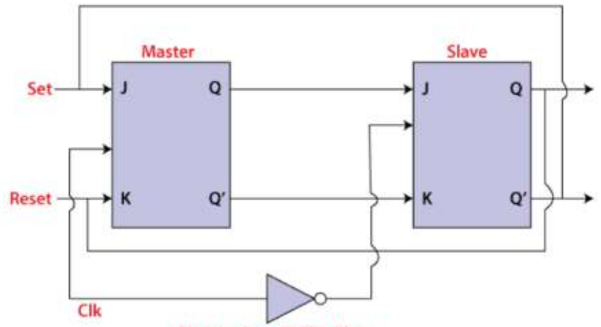
- With the inputs J = K = 1 and Q = 0, the present state changes to 1 after the propagation delay tp of the gates.
- Now we have J = K = 1 and Q = 1, and after another time interval tp, the output will change back to Q = 0
- Hence for a duration T of the clock pulse, the value of Q is ambiguous. This ambiguity is referred to as the race around condition

Methods to eliminate race around condition

- By using a clock satisfying the condition T < tp. This is not a feasible solution.
- Use of a master-slave configuration (pulse triggering)
- Use of edge triggering (as race around condition occurs only in level triggered flip flops)

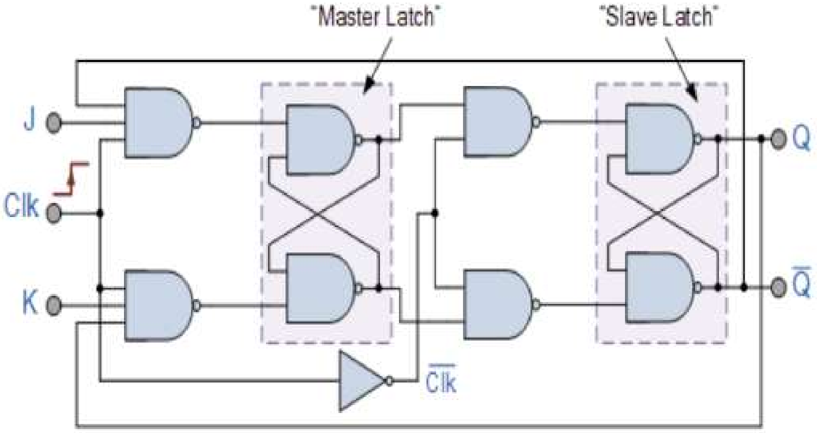
Master Slave JK Flip flop

- It is realized by cascading two JK flip flops, one acting as a master, while the other as slave
- When master flip flop is enabled, slave flip flop is disabled, and vice versa
- Both flip flops are driven from the same clock pulse, with the slave receiving its inverted version
- During the positive cycle of the clock pulse, master flip flop is enabled and its output is determined by the present input and the state of the flip flop.
- The variation occurring at the output of the master is transferred to the slave flip flop only when the clock goes from high to low



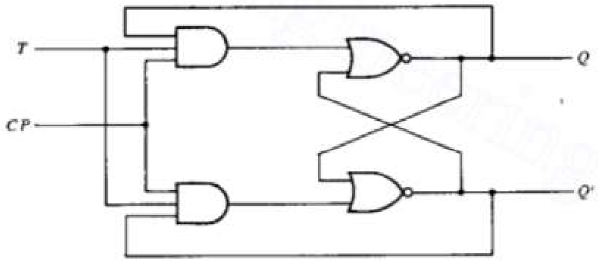
Master-slave JK Flip-Flop

- First half of the circuit is master JK and second half is slave JK
- Output of master Flip flop is fed as input to slave flip flop
- Clock is connected directly to master flip fop, but is connected through inverter to slave flip flop
- There is a feedback from the output of slave flip flop to the input of master flip flop

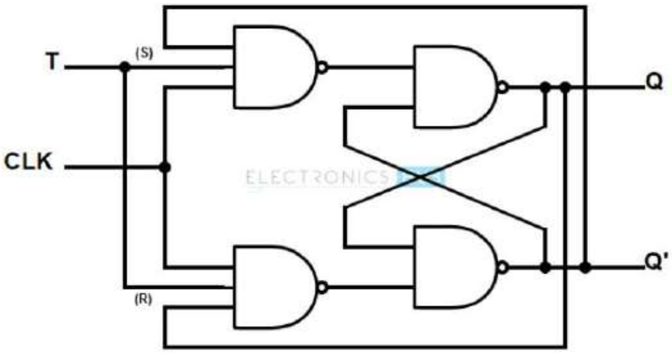


T Flip flop

- T flip flop is a single input version of the JK flip flop
- T flip flop is obtained from the JK flip flop when both inputs are tied together
- It is named 'T' because of the ability of the flip flop to 'toggle' or complement its state



T FF using NAND gates



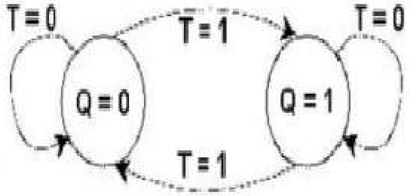
Q	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic table

Q \ T	0	1
0		1
1	1	

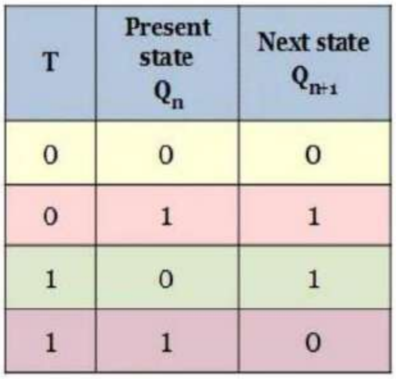
$Q(t+1) = TQ' + T'Q$

Characteristic equation



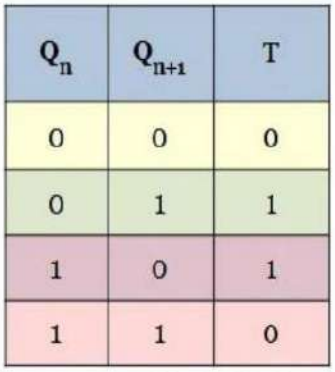
State transition diagram

Excitation Table of T Flip flop



T	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

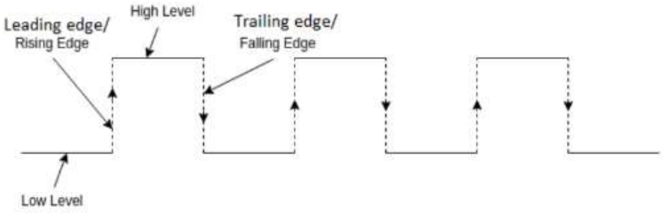
Truth table of T flip flop



Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table of T flip flop

Triggering of flip flops



A clock signal

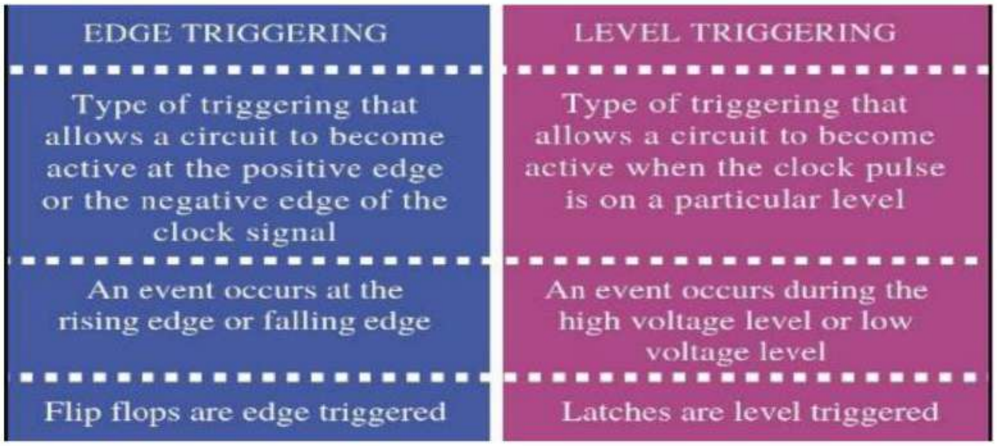
Three types of triggering

- Level Triggering
- Edge Triggering
- Pulse Triggering

Pulse triggering

- Pulse triggered flip flops are also referred to as master-slave flip flops
- It has a master section and a slave section cascaded together
- Master registers the data on one level of the input clock signal, which is transferred to the slave on the other level of the clock signal

Edge Triggering vs Level Triggering

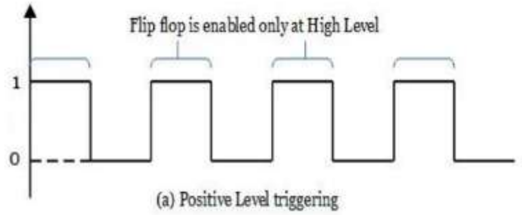


EDGE TRIGGERING	LEVEL TRIGGERING
Type of triggering that allows a circuit to become active at the positive edge or the negative edge of the clock signal	Type of triggering that allows a circuit to become active when the clock pulse is on a particular level
An event occurs at the rising edge or falling edge	An event occurs during the high voltage level or low voltage level
Flip flops are edge triggered	Latches are level triggered

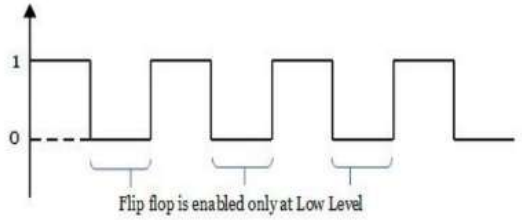
Level triggering

In this, the flip flop is triggered only during the high-level or the low level of the clock pulse.

- ***Positive level triggering*** – If the flip flop is triggered at the positive level of the clock pulse, then it is said to be a positive level triggering.
- ***Negative level triggering*** – If the flip flop is triggered at the negative level of the clock pulse, then it is said to be negative level triggering.



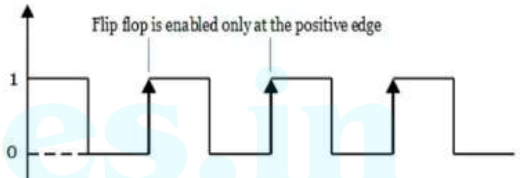
(a) Positive Level triggering



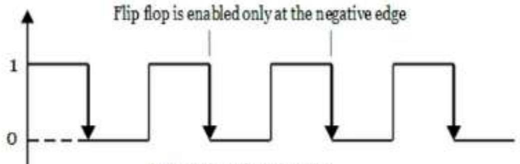
(b) Negative Level triggering

Edge triggering

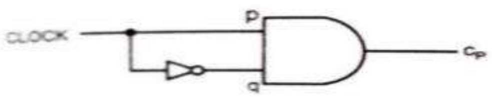
- In edge triggering, the flip flop changes its state during the positive edge or negative edge of the clock pulse. There are two types of edge triggering.
- ***Positive edge triggering*** – When the output responds to the change in the input only at the positive edge of the clock pulse, then the clock pulse is said to be a positive edge triggered.
- ***Negative edge triggering*** – When the output responds to the change in the input only at the negative edge of the clock pulse, then the clock pulse is said to be a negative edge triggered.



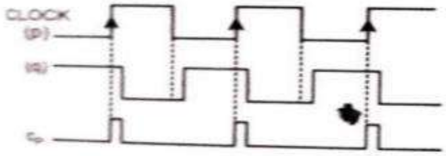
(a) Positive Edge triggering



(b) Negative Edge triggering

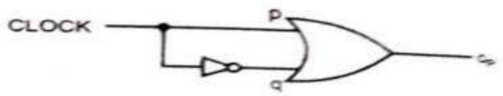


(a) Positive-edge detection circuit

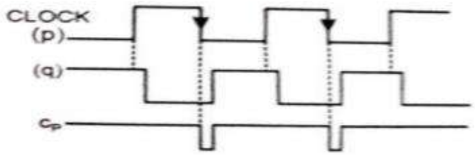


(c) Wave forms corresponding to (a)

- The clock input ‘p’ and its inverted version are ANDed to get cp.
- q is the inverted version of p that is delayed by propagation delay of the NOT gate
- When both p and q are ANDed, the output will be high only for a short duration concentrated at the leading edge of the clock input as shown



(b) Negative-edge detection circuit



(d) Wave forms corresponding to (b)

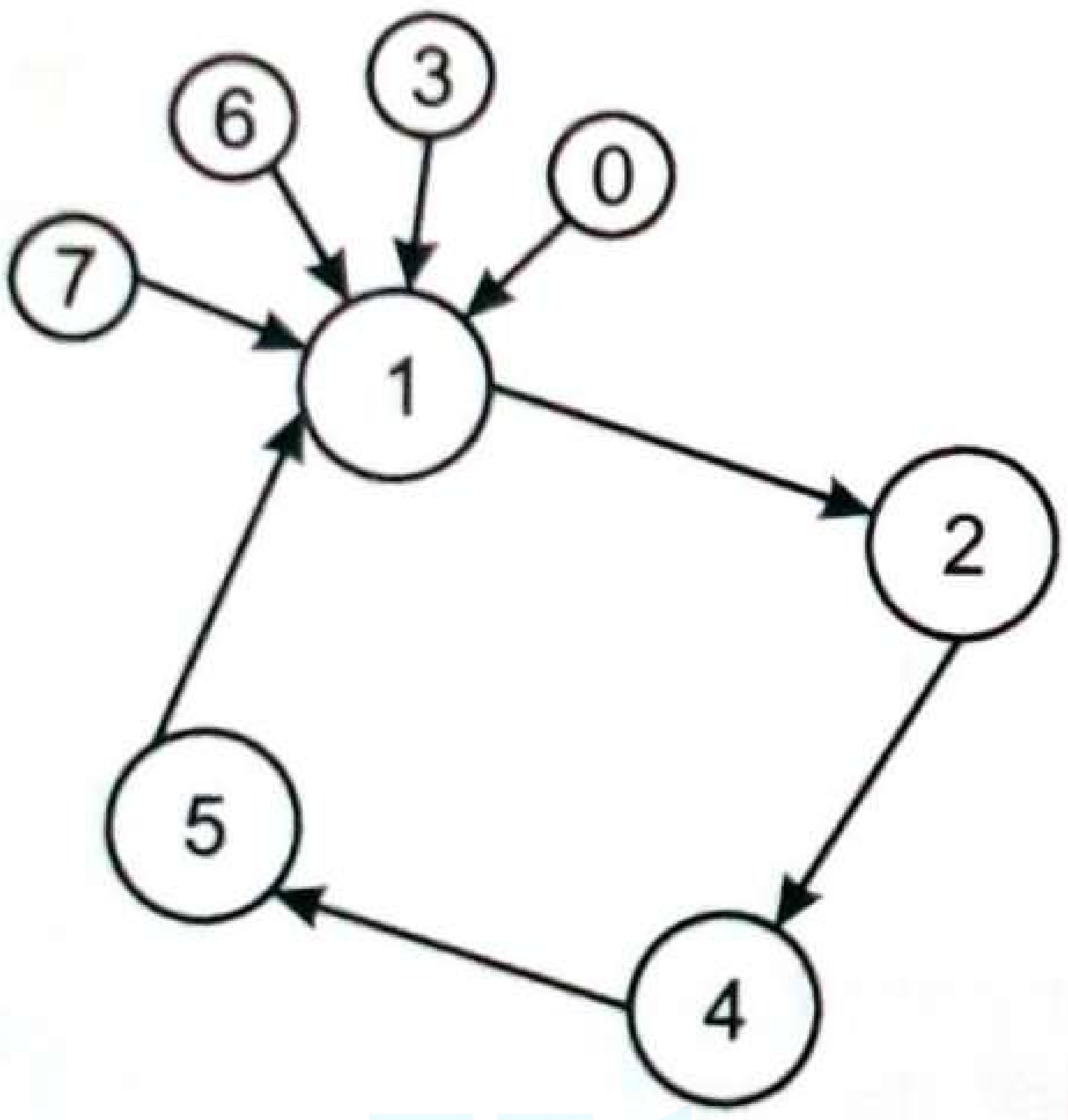
- The clock input ‘p’ and its inverted version are ORed to get cp.
- q is the inverted version of p that is delayed by propagation delay of the NOT gate
- When both p and q are ORed, the output will be high only for a short duration concentrated at the trailing edge of the clock input as shown

Example 6.5 *Design a synchronous counter to count the sequence 1 - 2 - 4 - 5 - 1 using T flip-flops.*

Solution:

Number of flip-flops, $n = \lceil log_2 5 \rceil = 3$

The state transition diagram and the state transition table is shown in Fig. 6.58 (a) and (b). Here the counting sequence is 1-2-4-5-1. There are other sequences counted by a 3-bit counter such as 0, 3, 6 and 7. To ensure that the counter do not



(a)

Count	Present	State		Next	State	
	Q_C	Q_B	Q_A	Q_C	Q_B	Q_A
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	1	0	0
3	0	1	1	0	0	1
4	1	0	0	1	0	1
5	1	0	1	0	0	1
6	1	1	0	0	0	1
7	1	1	1	0	0	1

(b)

Figure 6.58: (a) STD (b) STT

switch between unwanted states, it is advisable to direct the unwanted state to any one of the given possible state. it means that if the sequence happened to be 0, 3, 6, or 7, the next state can be directed to any one of the possible states, such as 1, 2, 4 or 5. In this case, it is assumed that when flip-flop reaches any other state (namely 0, 3, 6 or 7) the next state is directed to 1.

The flip-flop to be used is T whose excitation table is shown in Fig. 6.59 (a). Using the excitation table, the input conditions are derived as shown in Fig. 6.59 (b).

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

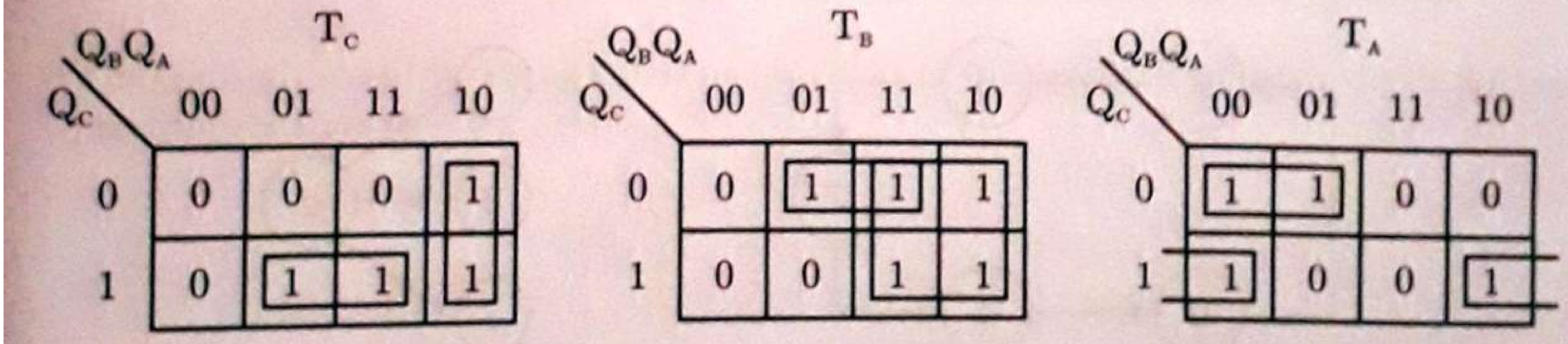
(a)

Present	State		Next	State				
Q_C	Q_B	Q_A	Q_C	Q_B	Q_A	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	1	0	0	1	1	0
0	1	1	0	0	1	0	1	0
1	0	0	1	0	1	0	0	1
1	0	1	0	0	1	1	0	0
1	1	0	0	0	1	1	1	1
1	1	1	0	0	1	1	1	0

(b)

Figure 6.59: (a) Excitation table for T-flip-flop (b) Input required for a T flip-flop

The K-map for inputs T_C, T_B and T_A are shown in Fig. 6.60.



T_C

Q_C \ Q_BQ_A	00	01	11	10
0	0	0	0	1
1	0	1	1	1

T_B

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	1	1
1	0	0	1	1

T_A

Q_C \ Q_BQ_A	00	01	11	10
0	1	1	0	0
1	1	0	0	1

Figure 6.60: K-map representation

Simplified expressions are given below:

$$T_C = Q_CQ_A + Q_B\overline{Q_A} \tag{6.18}$$

$$T_B = Q_B + \overline{Q_C}Q_A \tag{6.19}$$

$$T_A = \overline{Q_C}\,\overline{Q_B} + Q_C\overline{Q_A} \tag{6.20}$$

The circuit schematic for the required sequence can be drawn as shown in Fig. 6.61 by incorporating eq (6.18) to eq (6.20) and following the fundamental principles for a synchronous counter.

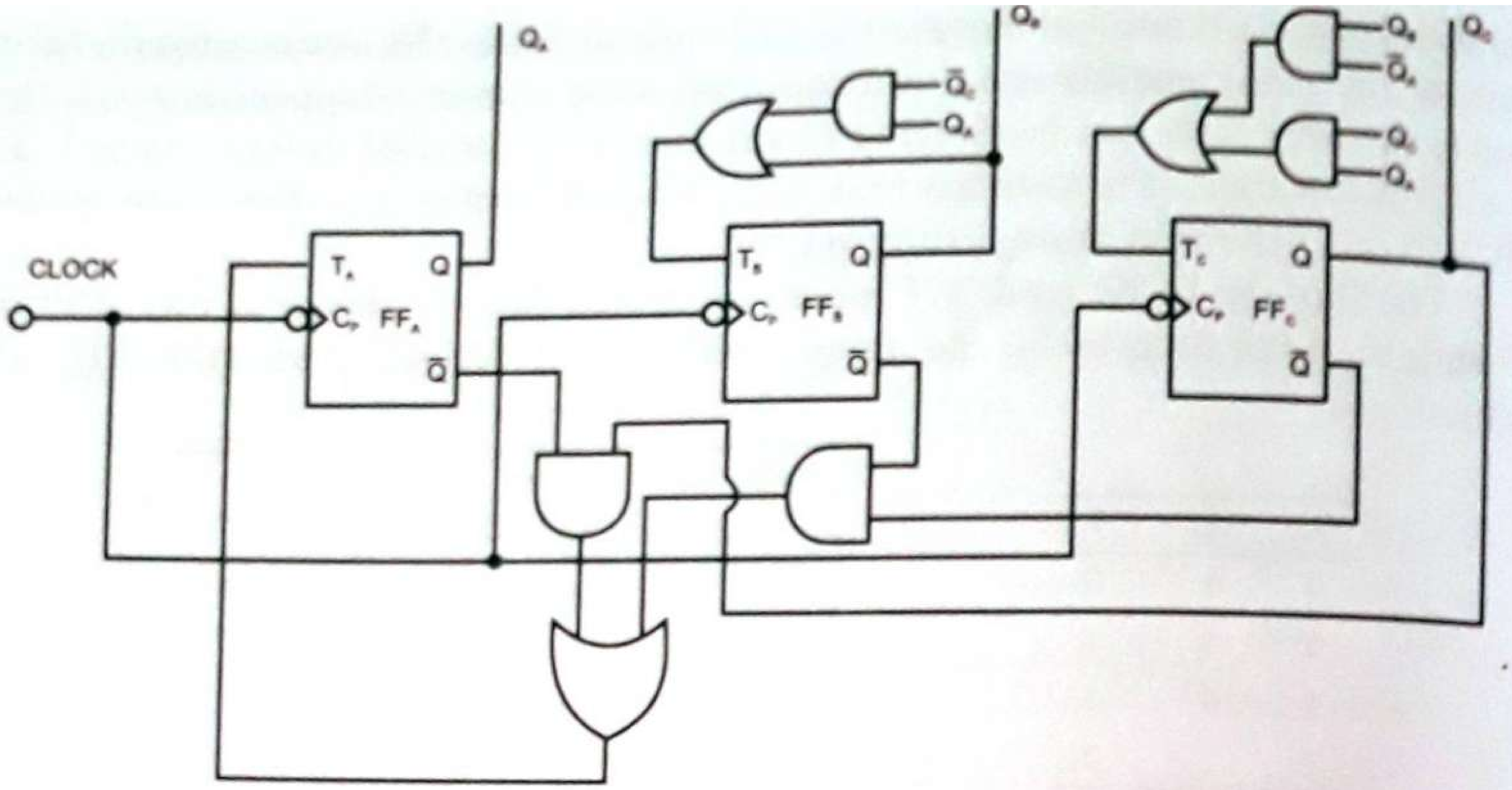


Figure 6.61: The synchronous counter for the required pattern

Example 6.6 *Design a synchronous counter to count the sequence 7 - 5 - 2 - 4 - 0 - 7..... Use D flip-flops. The undesired input conditions can be in don't care state.*

Solution:

Number of flip-flops, $n = \lceil \log_2 7 \rceil = 3$

The state transition diagram is shown in Fig. 6.62.

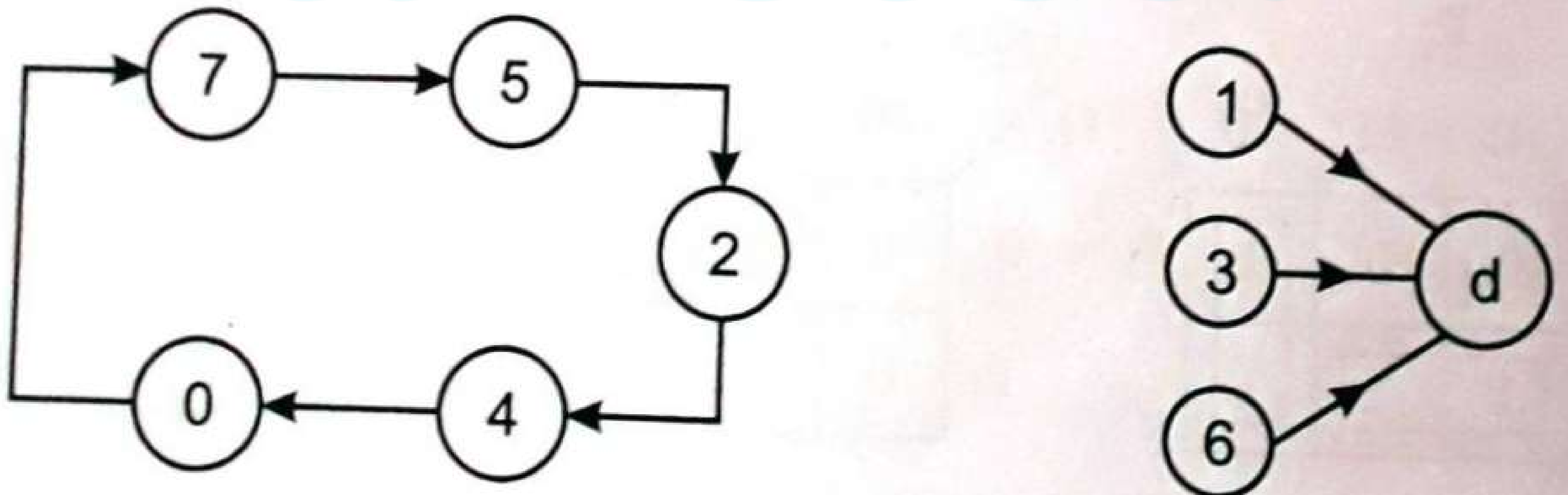


Figure 6.62: state transition diagram

The flip-flop to be used is D whose excitation table is shown in Fig. 6.63 a. State transition table along with required flip-flop inputs are shown in Fig. 6.63 b. K-map can be drawn as shown in Fig. 6.64 to represent the required input conditions.

The simplified expressions are given below:

$$D_C = \overline{Q_C} + Q_B \tag{6.21}$$

$$D_B = \overline{Q_C}\,\overline{Q_B} + \overline{Q_B}Q_A \tag{6.22}$$

$$D_A = \overline{Q_C}\,\overline{Q_B} + Q_C Q_B \tag{6.23}$$

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

(a)

Present	State		Next	State				
Q_C	Q_B	Q_A	Q_C	Q_B	Q_A	D_C	D_B	D_A
0	0	0	1	1	1	1	1	1
0	0	1	X	X	X	X	X	X
0	1	0	1	0	0	1	0	0
0	1	1	X	X	X	X	X	X
1	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	1	0
1	1	0	X	X	X	X	X	X
1	1	1	1	0	1	1	0	1

(b)

Figure 6.63: (a) Excitation table (b) Input requirements

D_C

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	1	X	X	1
1	0	0	1	X

D_B

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	1	X	X	0
1	0	1	0	X

D_A

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	1	X	X	0
1	0	0	1	X

Figure 6.64: K map representation

The desired circuit by using equations eqs 6.21 to 6.23 and following the basic design principle is shown in Fig. 6.65.

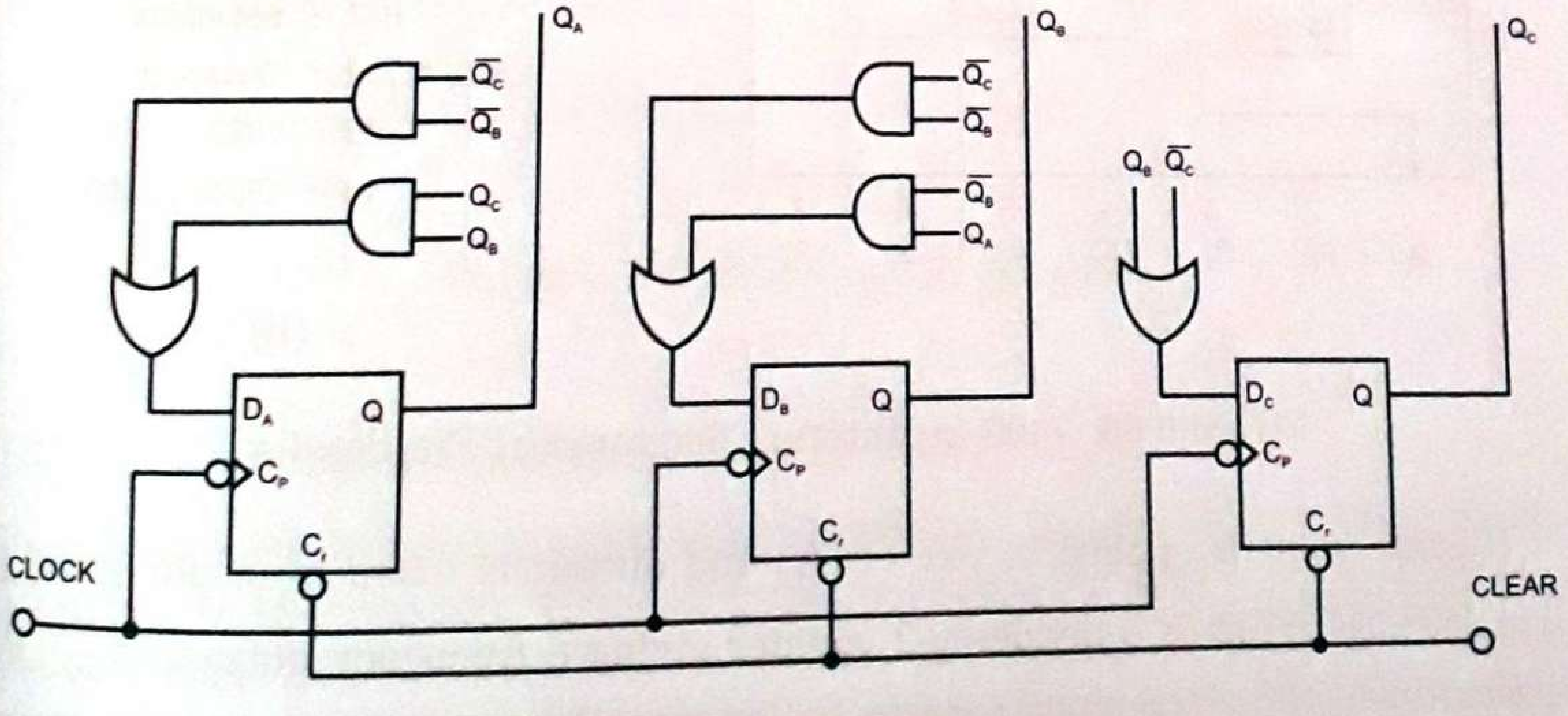


Figure 6.65: The complete schematic

6.4 Conversion of Flip-flops

So far, we have discussed various flip-flops. It is possible to convert one flip-f into the other by including additional logic circuits. The procedure for convers is listed below:

- Get the characteristic table of desired flip-flop.
- Fill the excitation values for inputs of given flip-flop for each combination present state and next state using the excitation table.
- Get the simplified expression for each excitation input (can use K-maps required).
- Draw the circuit diagram of the required flip-flop based on the simplified expressions using given flip-flop and additional logic gates.

Example 6.1 *Convert a given JK flip-flop to SR flip-flop.*

Solution:

Step 1: Here SR flip-flop is the desired one. Write its truth table or characteristic (Table 6.13).

Table 6.13: Truth table of SR flip-flop

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	Invalid
1	1	1	Invalid

Step 2: Fill the excitation values for inputs of JK flip-flop using the excitation table of JK flip-flop shown in Table 6.14. Excitation values corresponding to JK inputs are filled in the truth table of SR flip-flop using JK excitation table as shown in Table 6.15.

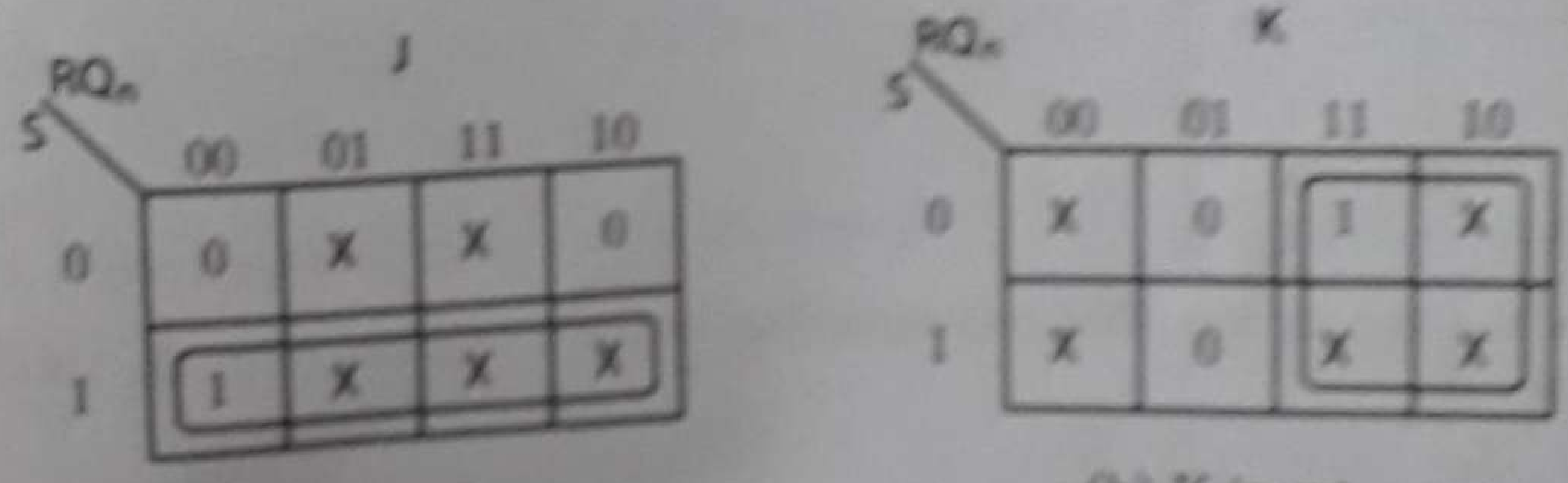
Table 6.14: Excitation table of JK flip-flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 6.15: Conversion table

S	R	Q_n	Q_{n+1}	J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	0	Invalid	X	X
1	1	1	Invalid	X	X

Step 3: Simplify for each excitation inputs J and K using K-maps shown in Fig 6.24 (a) and (b).



S \ RQ_n	00	01	11	10
0	0	X	X	0
1	1	X	X	X

(a) J input

S \ RQ_n	00	01	11	10
0	X	0	1	X
1	X	0	X	X

(b) K input

Figure 6.24: K-maps

From the K-map, we have

$$J = S \tag{6.3}$$

and

$$K = R \tag{6.4}$$

Step 4: Draw the circuit diagram. Using Eq. (6.3) and (6.4), the diagram can be drawn as in Fig 6.25.

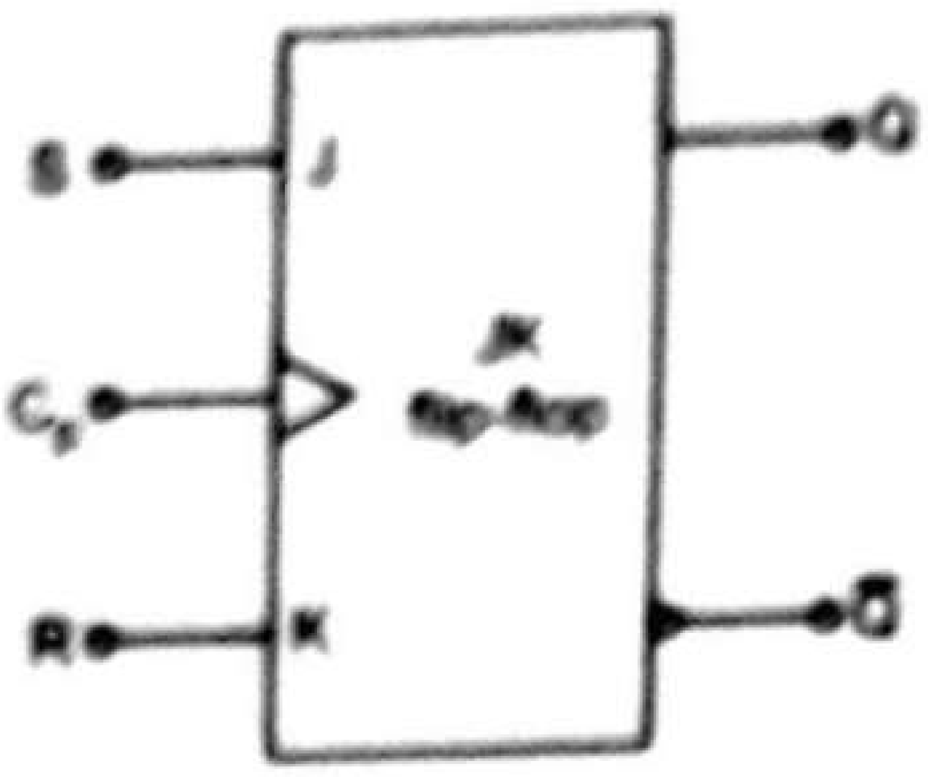


Figure 6.25: SR flip-flop derived from JK flip-flop

Example 6.2 *Convert a given JK flip-flop to D flip-flop.*

Solution:

Step 1: Write the truth table of D flip-flop

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Step 2: Write the excitation table of JK flip-flop.

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Now the inputs are filled corresponding to the transition from Q_n to Q_{n+1} as shown below.

D	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

Step 3: Simplify using K-maps (Fig 6.26) for inputs J and K.

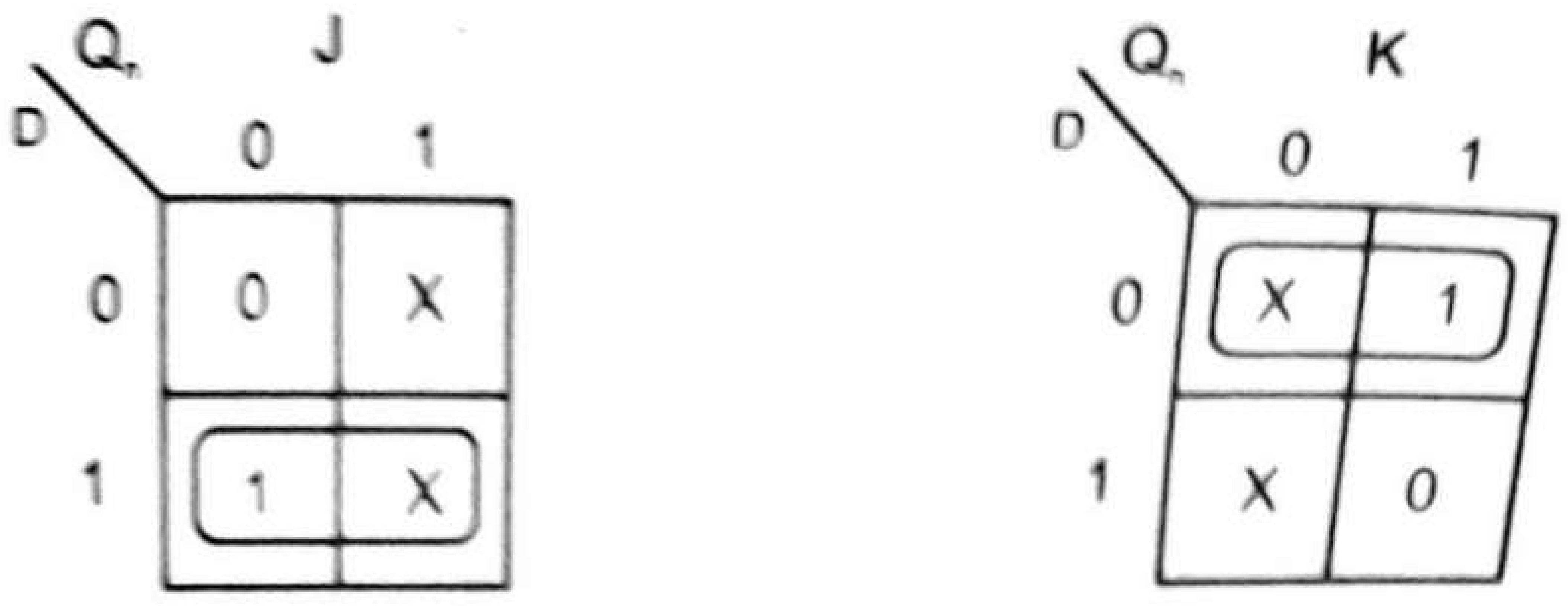
From the K-maps, we have

$$J = D \qquad (6.5)$$

and

$$K = \bar{D} \qquad (6.6)$$

Step 4: Draw the circuit diagram using equations 6.5 and 6.6 (Fig 6.27).



D \ Q_n	0	1
0	0	X
1	1	X

(a) J input

D \ Q_n	0	1
0	X	1
1	X	0

(b) K input

Figure 6.26: K-maps

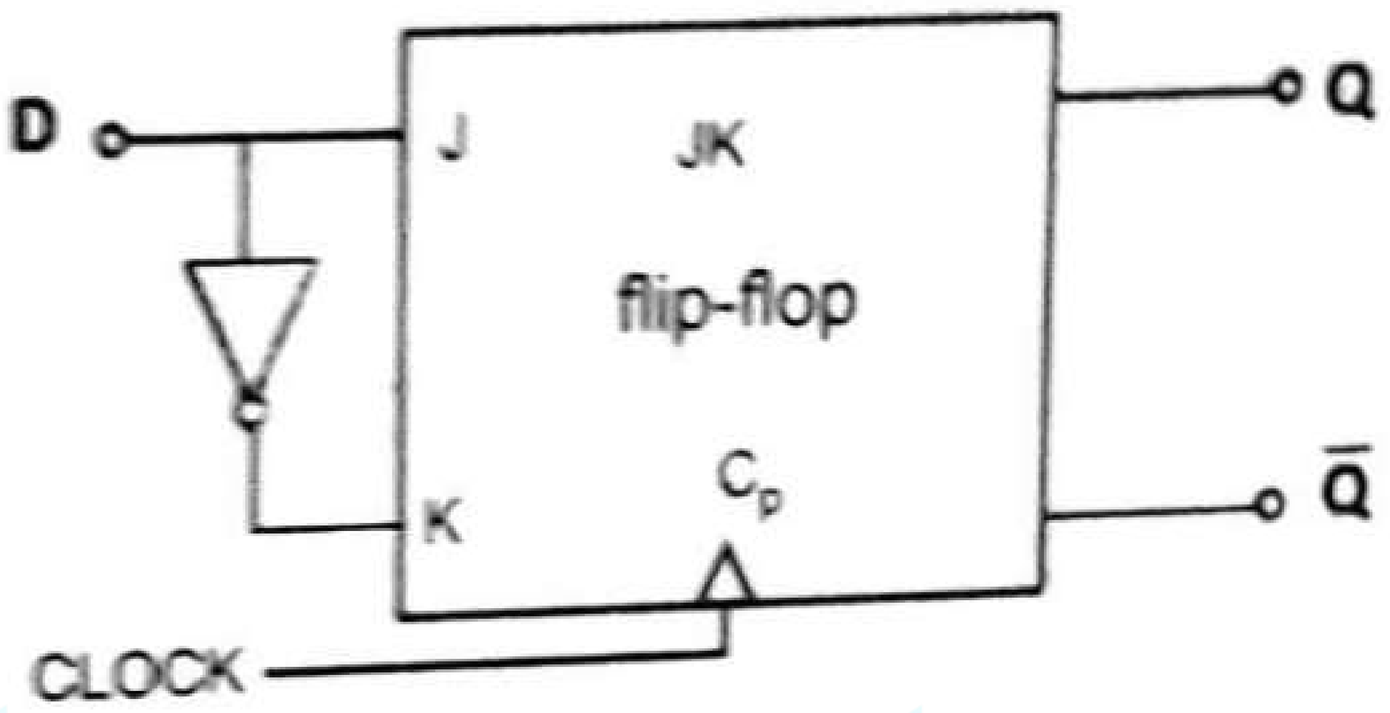


Figure 6.27: D flip-flop using JK flip-flop

Example 6.3 *Convert a given T flip-flop to JK flip-flop.*

Solution:

Step 1: Write the truth table for JK flip-flop.

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

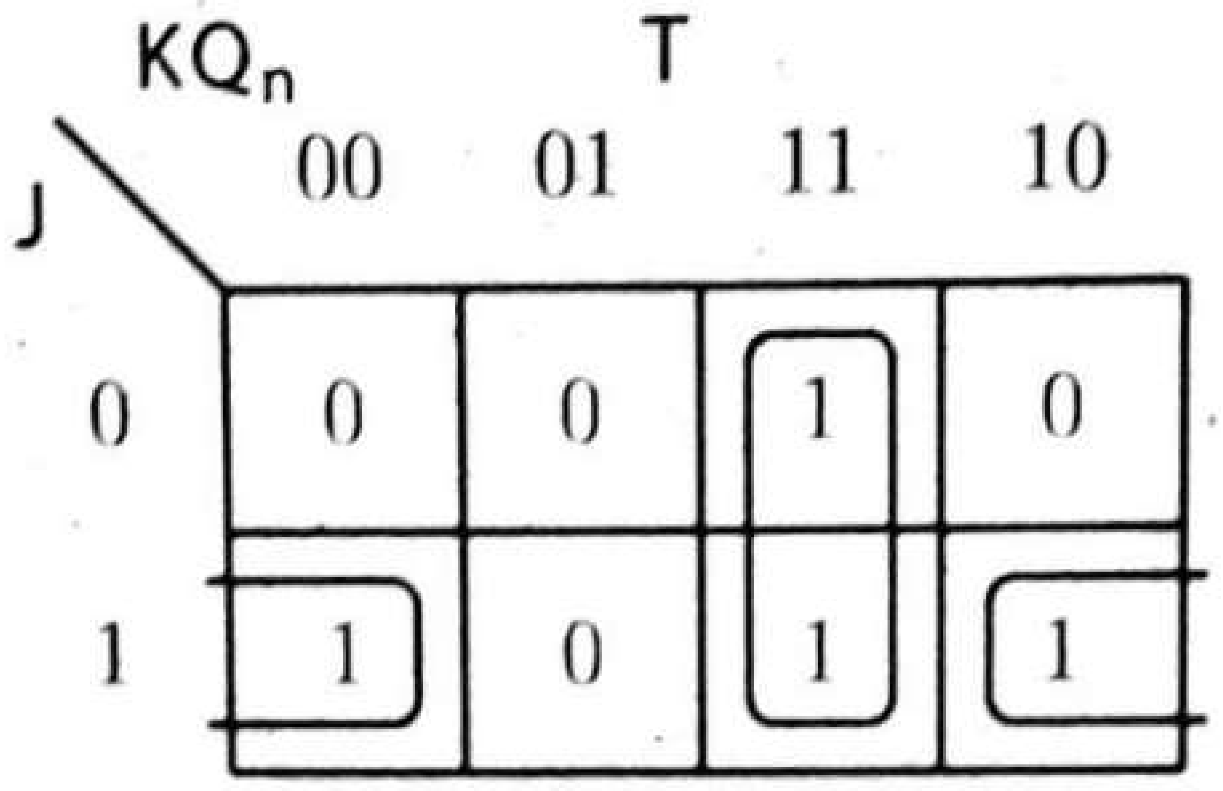
Step 2: Write the excitation table of T flip-flop.

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

The inputs are to be filled corresponding to the transition from Q_n to Q_{n+1} as shown in below.

J	K	Q_n	Q_{n+1}	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

Step 3: Simplify using K-map for the input T (Fig 6.28).



J \ KQ_n	00	01	11	10
0	0	0	1	0
1	1	0	1	1

Figure 6.28: K-map for T input

From the K-map, we have,

$$T = KQ_n + J\bar{Q_n} \tag{6.7}$$

Step 4: Using the equation 6.7, the required circuit can be drawn as shown in Fig 6.29.

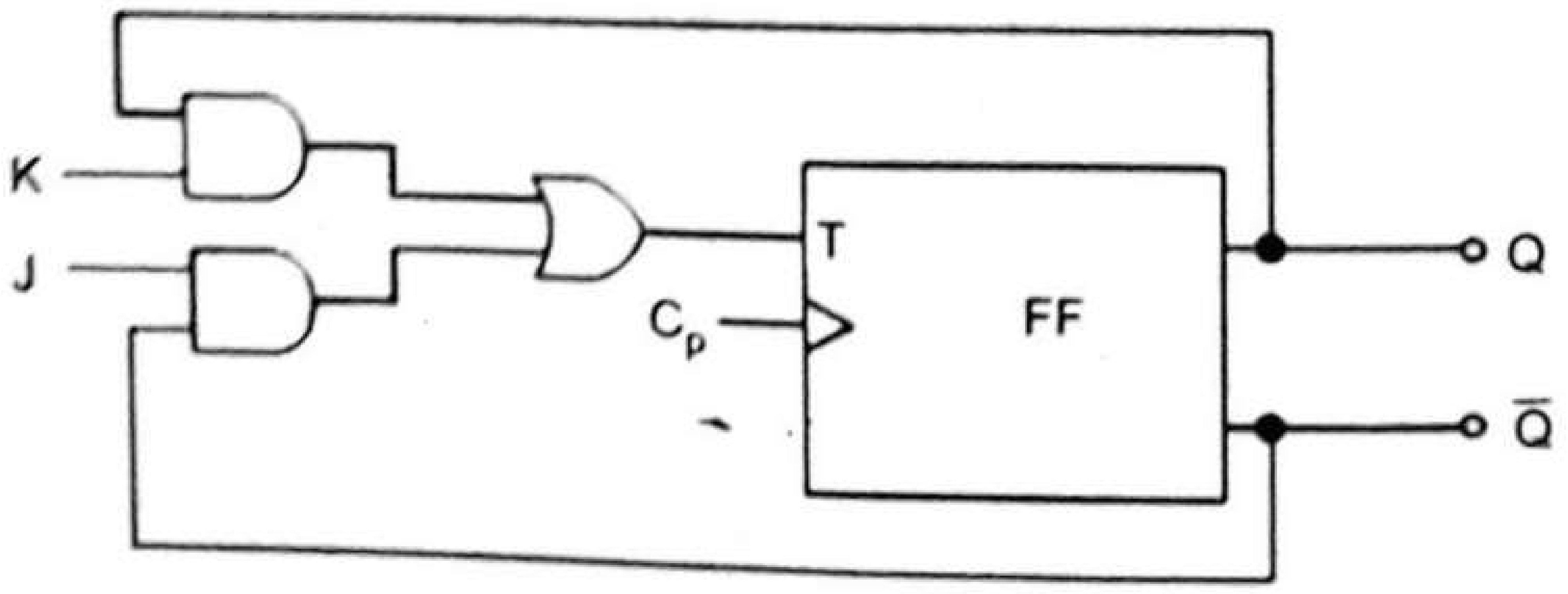


Figure 6.29: JK flip-flop derived from T flip-flop

6.5 Registers

In the discussion of flip-flops we have seen that 1 flip-flop can store 1-bit of information. If 2 bits are to be stored, then 2 flip-flops are to be connected together, for