

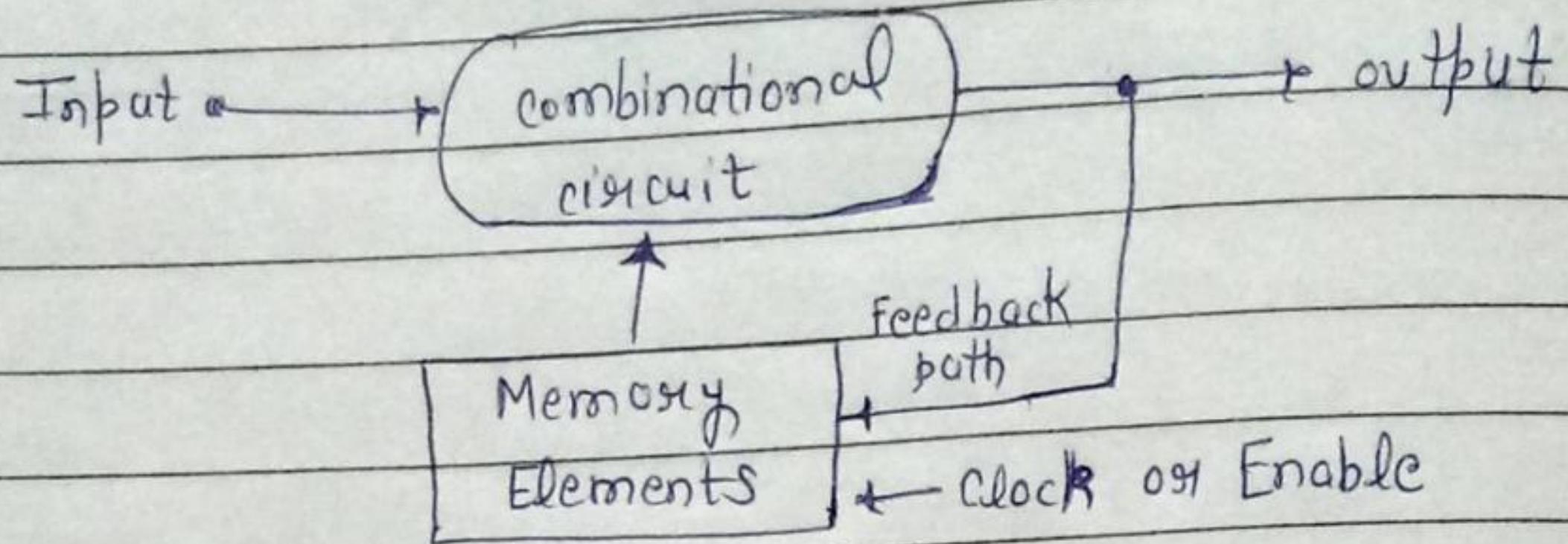
# SEQUENTIAL CIRCUITS

SN

Date: \_\_\_\_\_  
Page: \_\_\_\_\_

Sequential circuit are those which follows specific order and each stage in these circuits depends on the result of previous stage.

Sequential circuit is a digital logic with feedback.



Block diagram

## Comparison +

### Combinational Circuits

1. In this circuits, output doesn't depends upon previous state.
2. Combinational Circuits are faster in speed.
3. These are expensive in cost.
4. Memory element are not present in this circuit.
5. Combinational circuit are easy to design.

### Sequential Circuits

1. In this circuits, output depends upon present inputs as well as previous inputs.
2. Sequential Circuit are slower than combinational circuits.
3. These are cheaper.
4. Memory elements are present to store the past history of input variables.
5. Sequential circuits are difficult to design.

6. Feedback path is not used

6. Feedback path is used.

7. Examples → Adders, subtractors  
code converters, multiplexers  
de-multiplexers, encoders etc.

7. Examples → flip-flop,  
counters, Registers etc.

### Sequential Circuits

Synchronous Circuit

Asynchronous Circuit

Comparison +

Synchronous Circuit

Asynchronous Circuit

1. Memory elements are clocked  
flip-flops in synchronous  
sequential circuit.

2. These circuit are easier  
to design.

3. Memory elements is affected  
due to change in input signal  
because of activation of  
clock signal.

4. output change at certain  
time.

1. Memory elements are either  
unlocked flip-flops or time  
delay elements.

2. These circuits are more difficult  
to design as compared to  
synchronous sequential circuit.

3. Memory element is affected  
due to change in input signal  
at any instant of time.

4. Output change any time.

### Definitions +

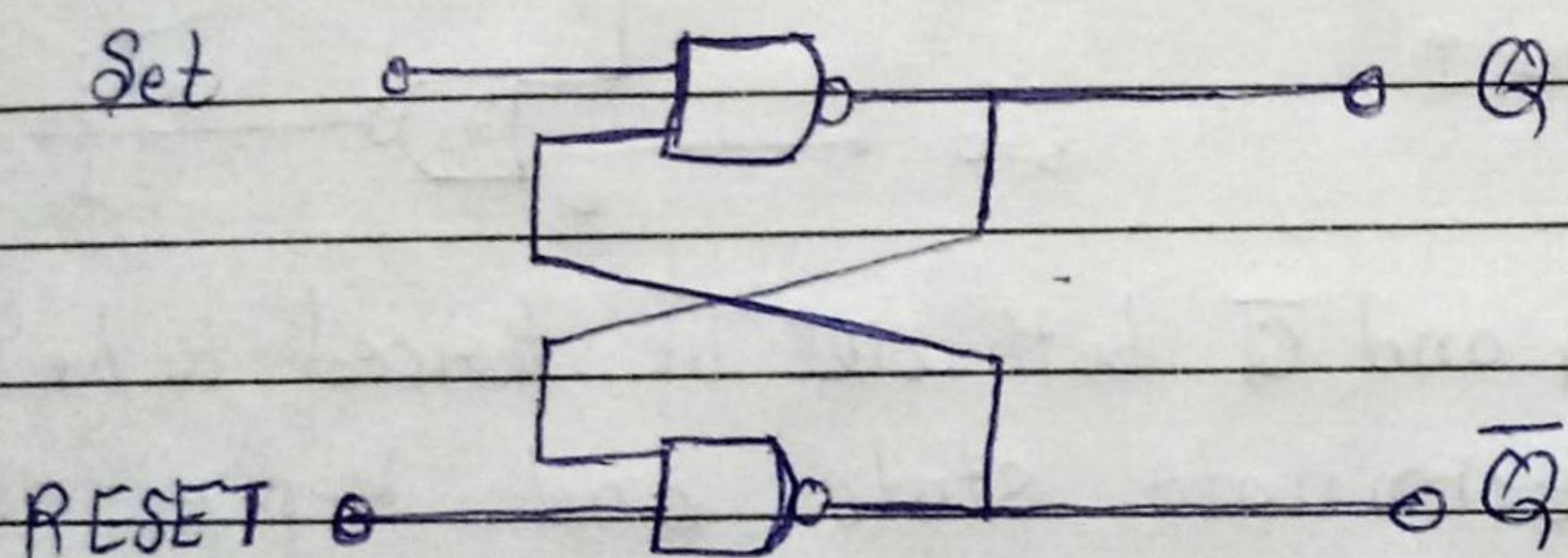
Present state + The data stored by the memory element at any given instant of time is called as the present state.

Next State + The combinational circuit operates on the external inputs and the present state to produce new outputs. Some of new outputs are stored in the memory element and called as next state.

+ The most important part of the sequential circuit seems to be the memory element. The memory element is known as Flip-Flop. It is basic memory unit.

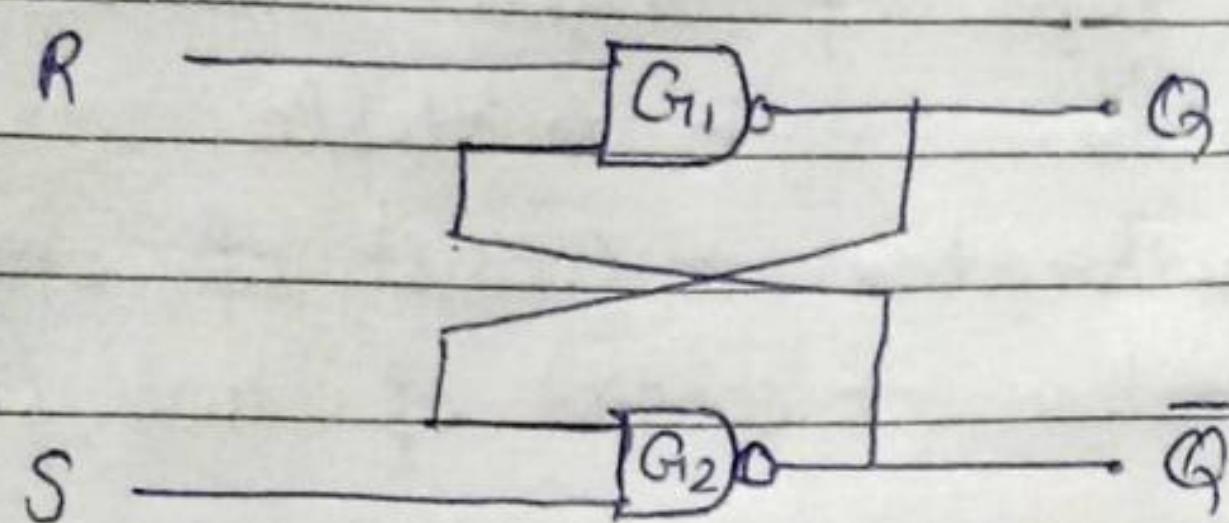
## LATCH

Latch is a type of temporary storage device which is formed by two NOR Gates. It's a one kind of a logic circuit and it is also known as bistable multivibrator. Because it has two stable states namely active high as well as active low it works like a storage device by holding the data through a feedback lane. It store 1-bit of data as long as long the apparatus is activated. One enable is declared then instantly latch can change the stored data.



S-R LATCH +

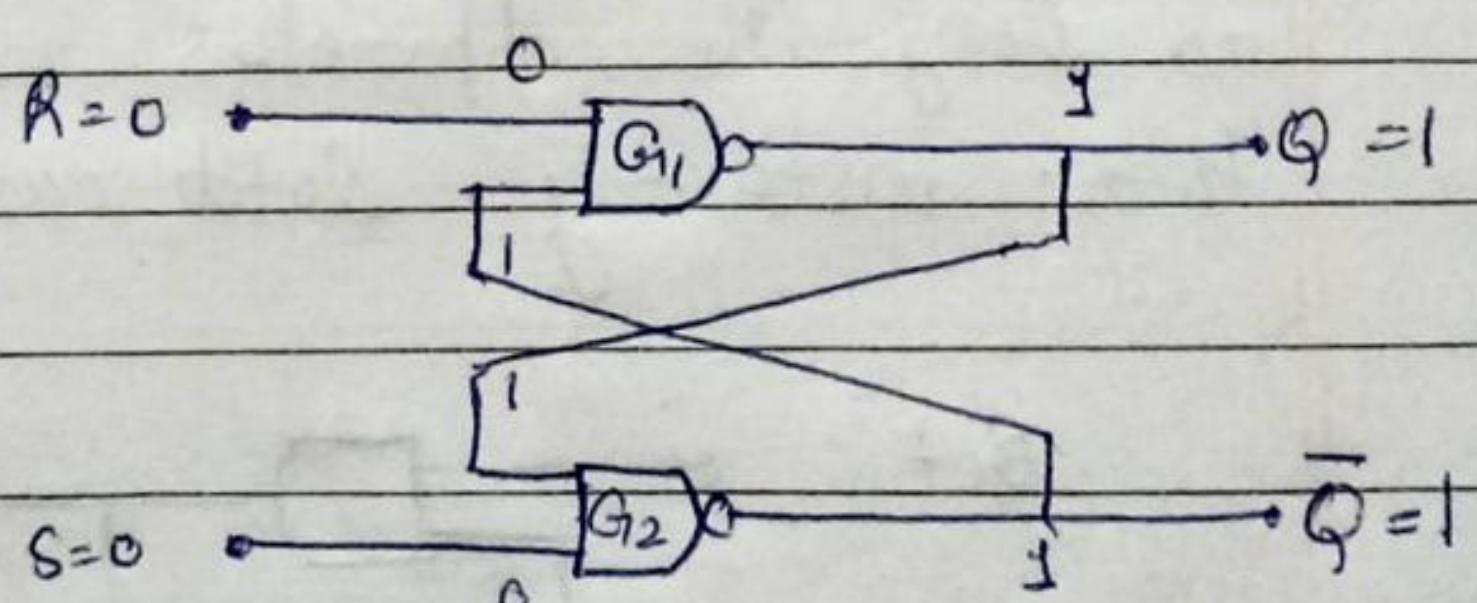
S-R Latch is the simplest type of latch circuit is known as Set-Reset Latch. It is used in electronic times.

S-R Latch using NAND Gates +

The two NAND Gates are cross-coupled so that the output of  $G_1$ , i.e. NAND 1, is connected to one of the input of  $G_2$  i.e. NAND 2 and vice-versa. Here  $Q$  and  $\bar{Q}$  are the latch output. Under normal condition, these output will always be the inverse of each other.

Case-I when  $S=0$  and  $R=0$ 

- When any one input of NAND gate becomes 0, its output is forced to 1.



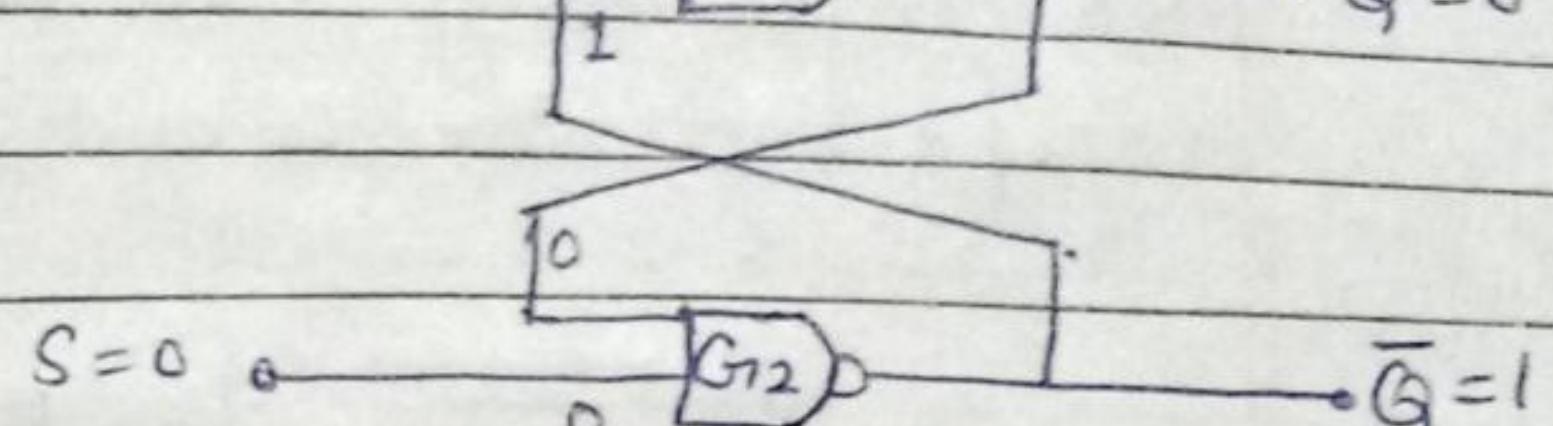
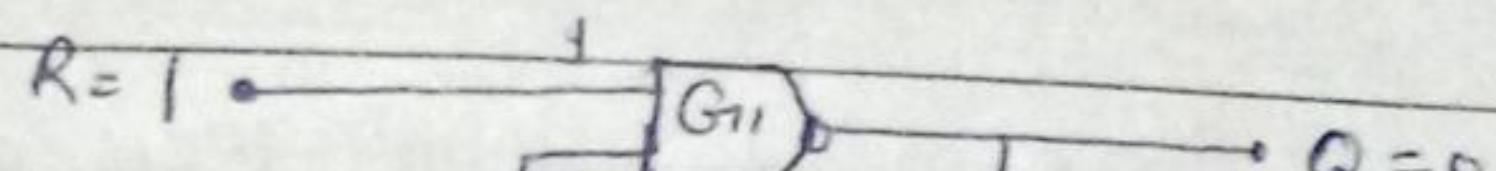
- Here  $S=R=0 \therefore Q$  and  $\bar{Q}$  both will be forced to be equal to 1.
- This is an undeterminate state and Hence should be avoided.
- This is also called as Race condition.

Case-2 when  $S=0, R=1$  : Reset

→ Hence  $Q=0$

→ Thus with  $S=0$  and  $R=1$  the output are  $Q=0$  and  $\bar{Q}=1$

→ This is Reset condition



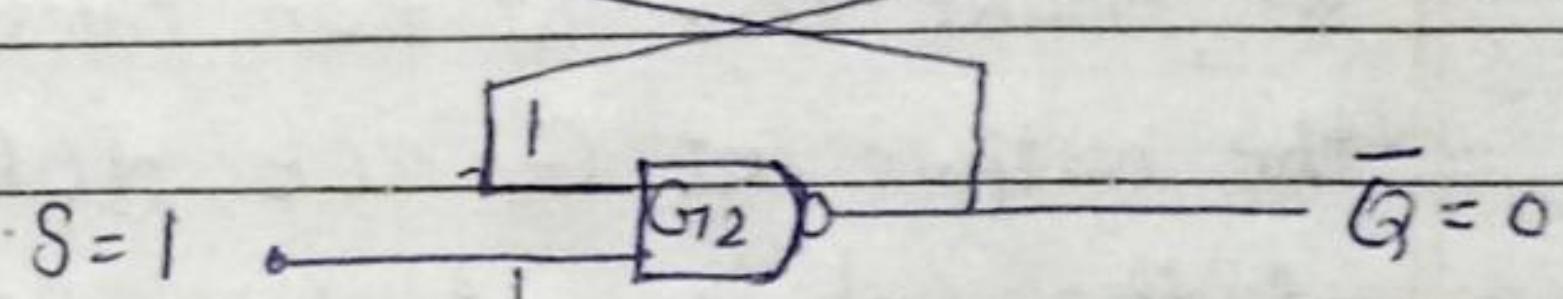
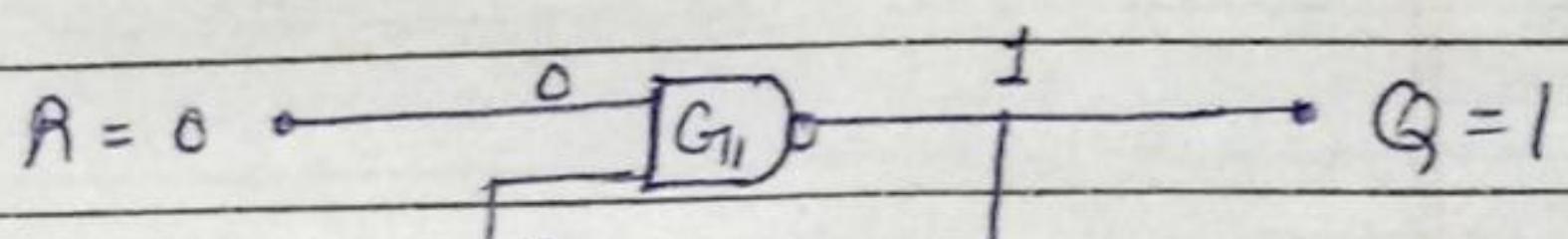
Case-3 when  $S=1, R=0$  : SET

→ Hence  $\bar{Q}=0$

→ Hence with  $S=1$  and  $R=0$

the output are  $Q=1$  and  $\bar{Q}=0$

→ SET condition.



Case-4 when  $S=1$  and  $R=1$  : Nochange

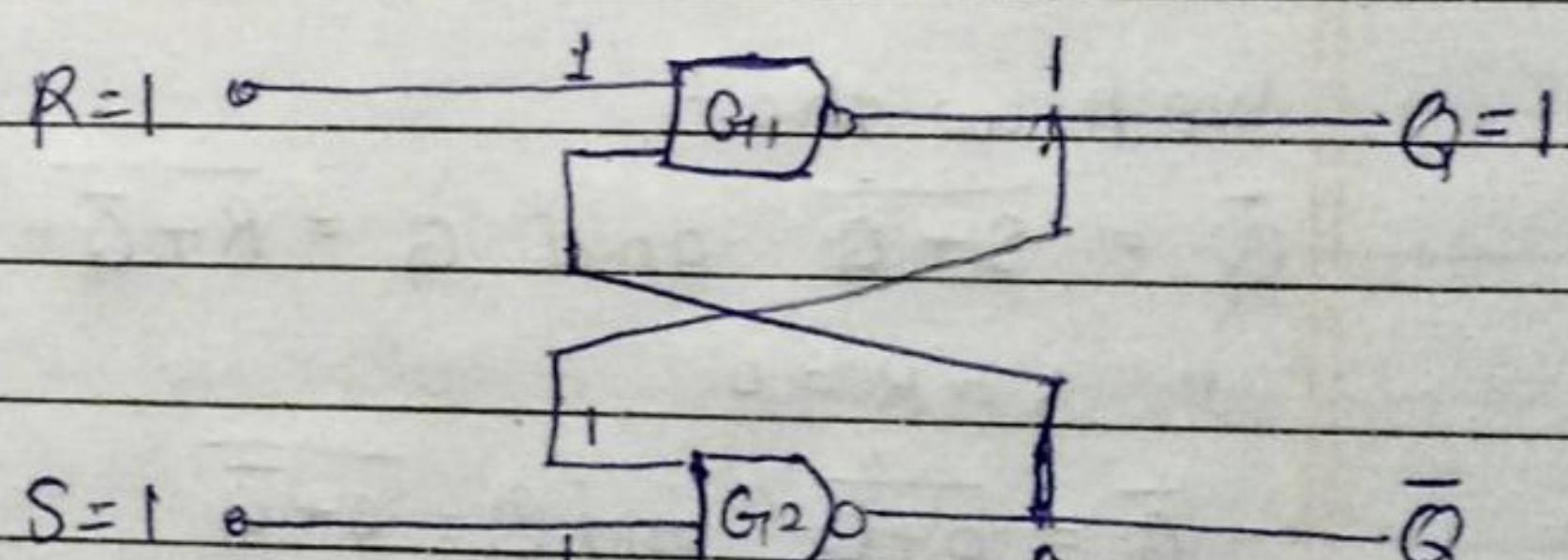
This is Nochange condition.

$$Q_{n+1} = \overline{R} \cdot \overline{Q}_n \quad \overline{Q}_{n+1} = \overline{S} \cdot Q_n \quad S=1$$

$$= \overline{R} + Q_n$$

$$= \overline{S} + \overline{Q}_n$$

$$Q_{n+1} = 0 + Q_n = Q_n \quad = 0 + \overline{Q}_n = \overline{Q}_n$$

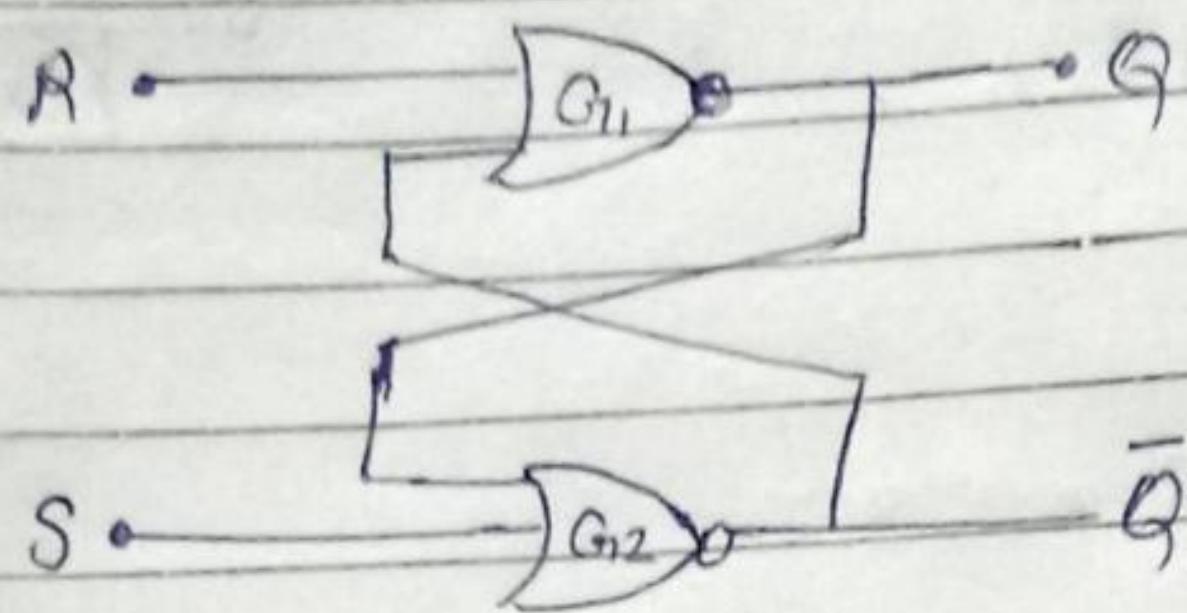


!

Input		Output	State
S	R	Q	
0	0	X	Invalid
0	1	0	Reset
1	0	1	Set
1	1	Q	Nochange

## S-R LATCH Using NOR Gates :

SR Latch using NOR gates is similar to SR NAND latch except that the  $Q$  and  $\bar{Q}$  output have reversed positions.



It makes use of two cross-coupled NOR Gates so that the output of  $G_1$  i.e. NOR 1 is stored or connected to one of the inputs of  $G_2$  i.e. NOR 2 and vice-versa.

Case-1. when  $S=0$  and  $R=0$ , we have.

We know that

$$\bar{Q} = \overline{S+Q} \text{ and } Q = \overline{R+\bar{Q}}$$

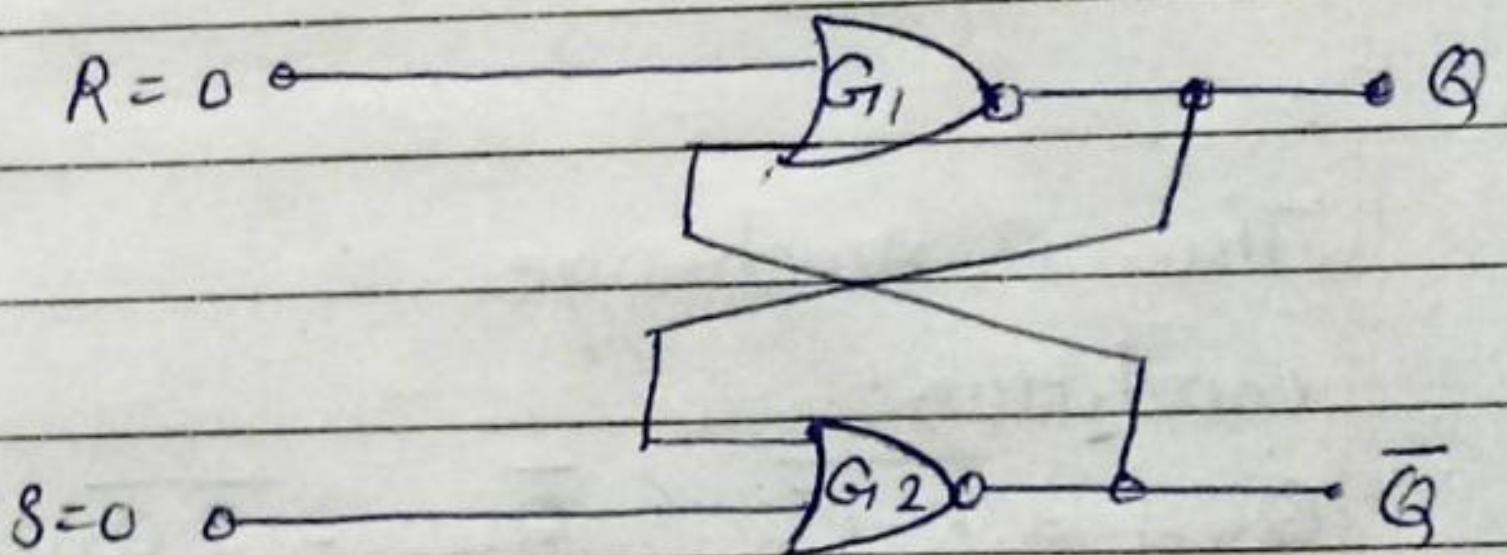
$$\therefore S = R = 0$$

$$\therefore \bar{Q} = \overline{0+Q} \text{ and } Q = \overline{0+\bar{Q}}$$

$$\bar{Q} = 0 \cdot \bar{Q} \quad Q = \bar{Q} \cdot Q$$

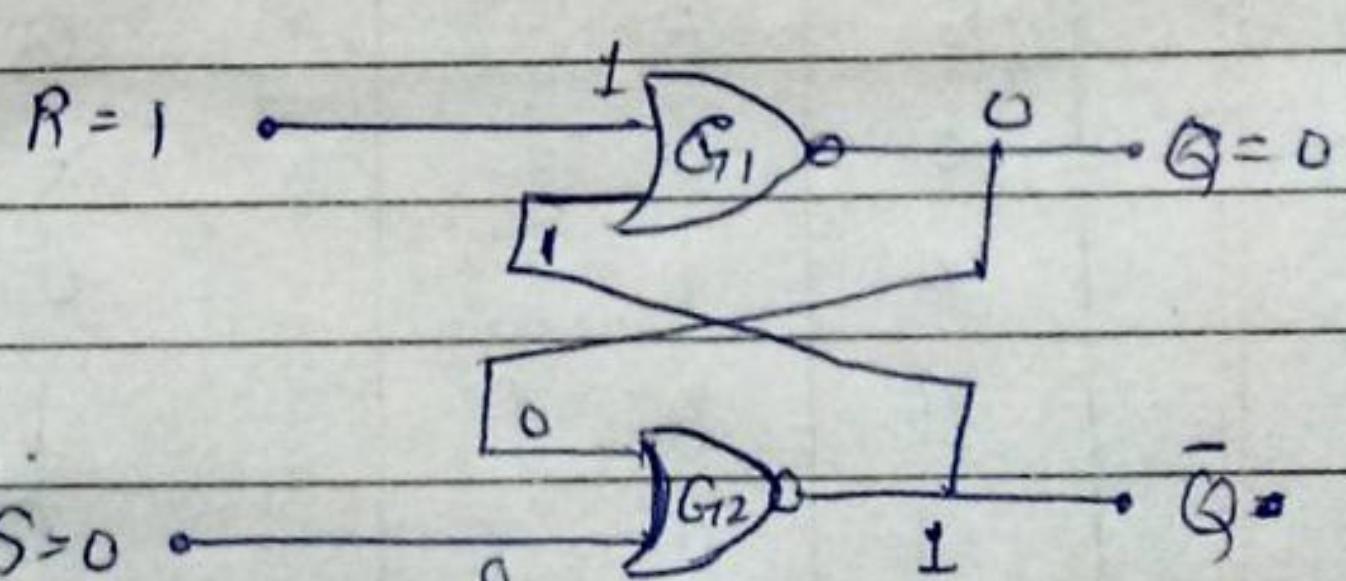
$$\bar{Q} = 1 \cdot \bar{Q} \quad Q = 1 \cdot Q$$

$$= \bar{Q} \quad = Q$$



No change condition.

Case-2. when  $S=0$  and  $R=1$



→ as  $R=1$  output of NOR-1 i.e.  $Q=0$

→ Hence both inputs to NOR-2 are 0.

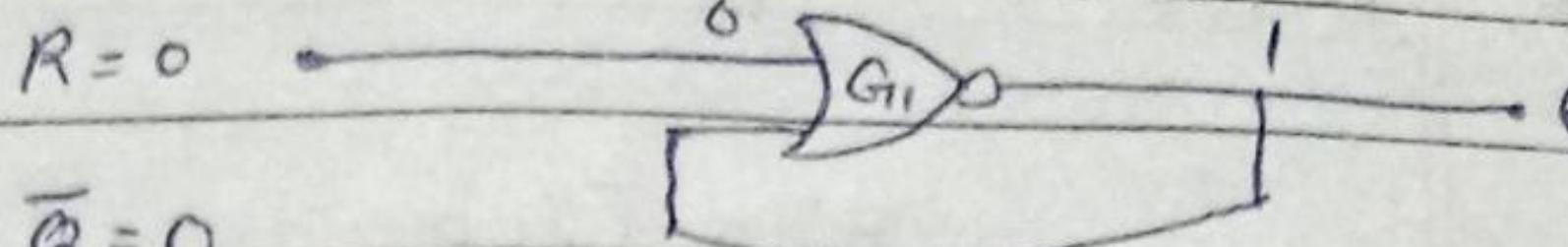
→ output of NOR-2 i.e.  $\bar{Q}=1$

∴ For  $S=0$  and  $R=1 \rightarrow Q=0$  and  $\bar{Q}=1$

Case 3 when  $S=1$  and  $R=0$

+ As  $S=1$ , output of NOR gate 2 i.e.  $\bar{Q}=0$

+ both inputs of NOR-1 are 0.



+ Therefore  $Q=1$

+ Hence both inputs to NOR-2 are 1

+ Hence  $\bar{Q}=0$

+ For  $S=1$ ,  $R=0$ ,  $Q=1$  and  $\bar{Q}=0$

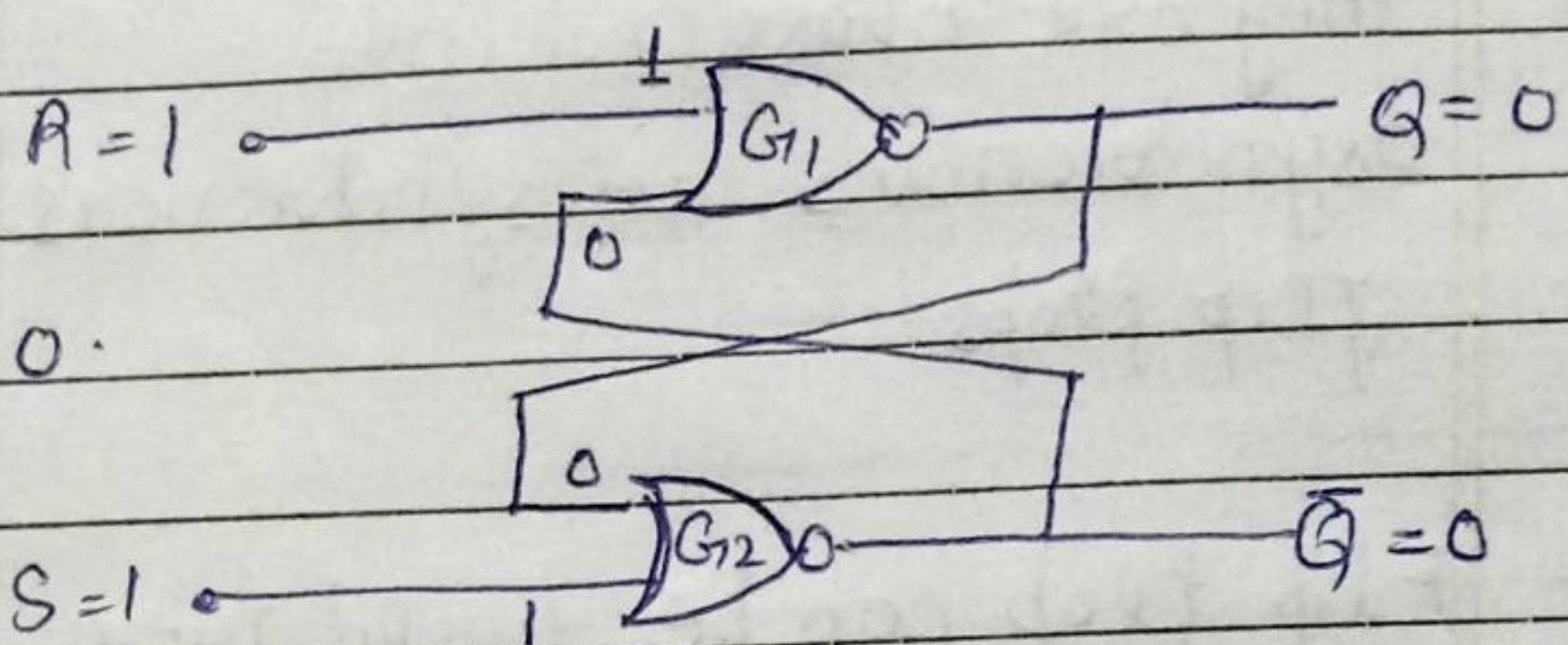
Case 4 when  $S=1$  and  $R=1$

+ If  $S=R=1$  then

outputs  $Q$  and  $\bar{Q}$  both to be 0.

+ Hence intermediate state

which should be avoided.



Inputs		Output	State
S	R	Q	
0	0	0	No change
0	1	0	Reset
1	0	1	Set
1	1	X	Invalid

## "Multivibrator"

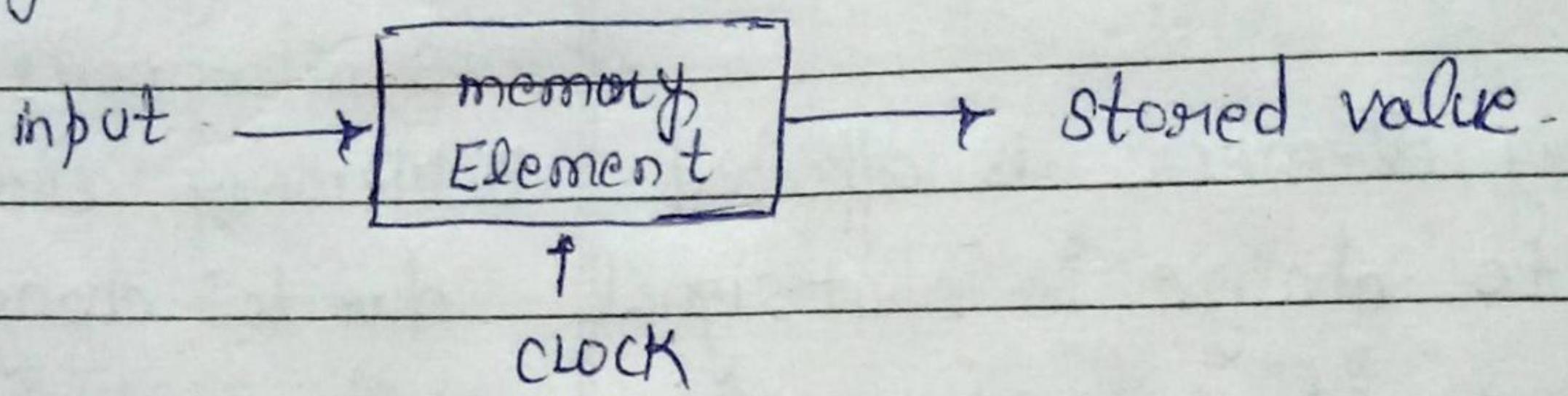
Most of the digital circuits or systems need some kind of a timing waveform for example, all clocked sequential system required a source of trigger pulses. A timing circuit which produces a rectangular waveforms are referred to as multivibrators.

There are different types of multivibrators such as-

1. Astable Multivibrator. → no stable condition
2. Monostable Multivibrator. → consist of one stable condition
3. Bistable Multivibrator. → consist of two stable condition.

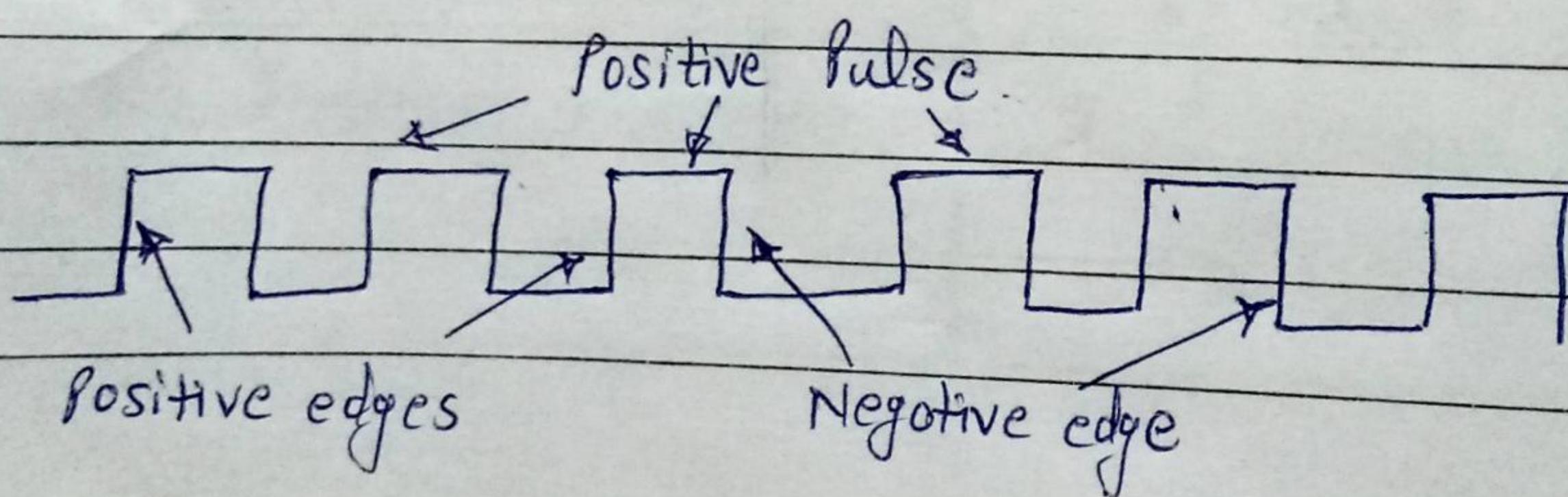
## "FLIP-FLOPS"

Flip-flop is a sequential circuit having two stable output states and it can be stay in any one of the two stable states unless external input is applied. So, it is basic memory element used to store a bit of information i.e either 0 or 1. It is a memory element which change its condition based on clock signal.

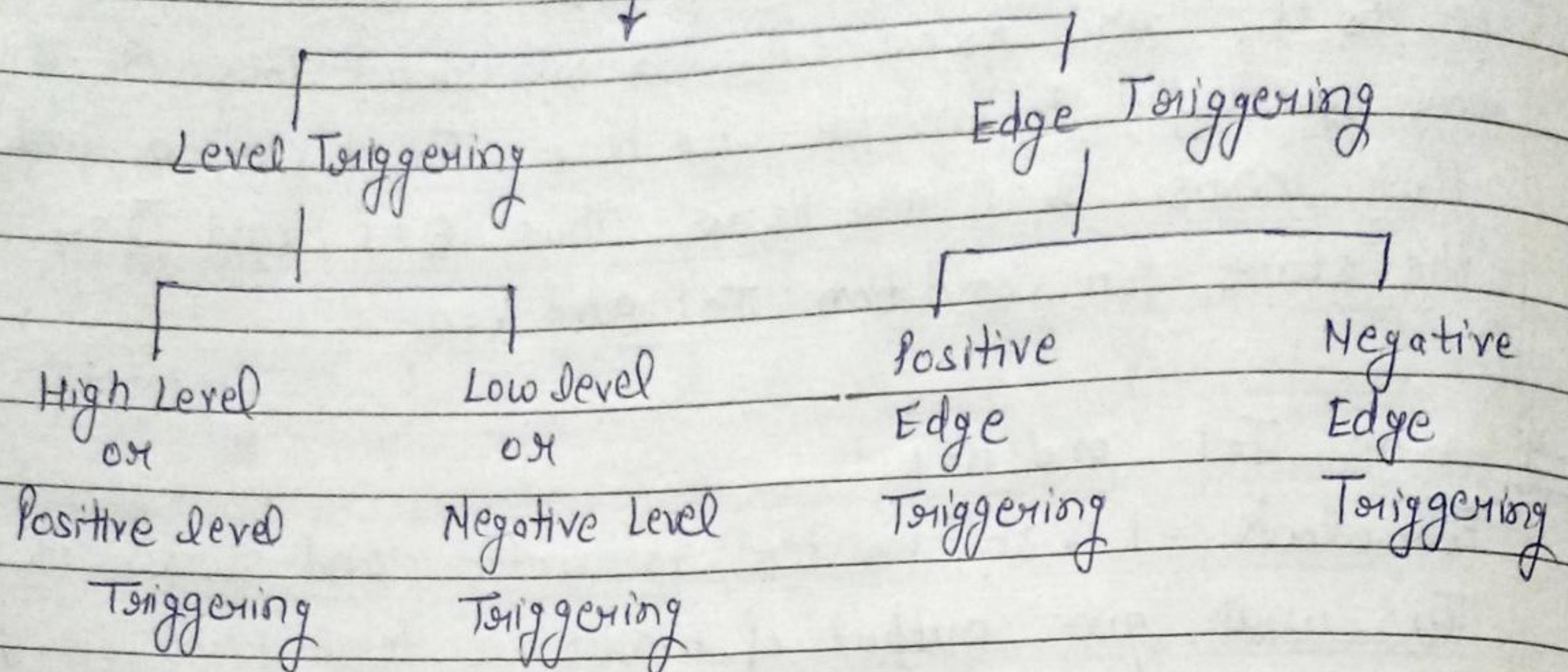


## "CLOCK"

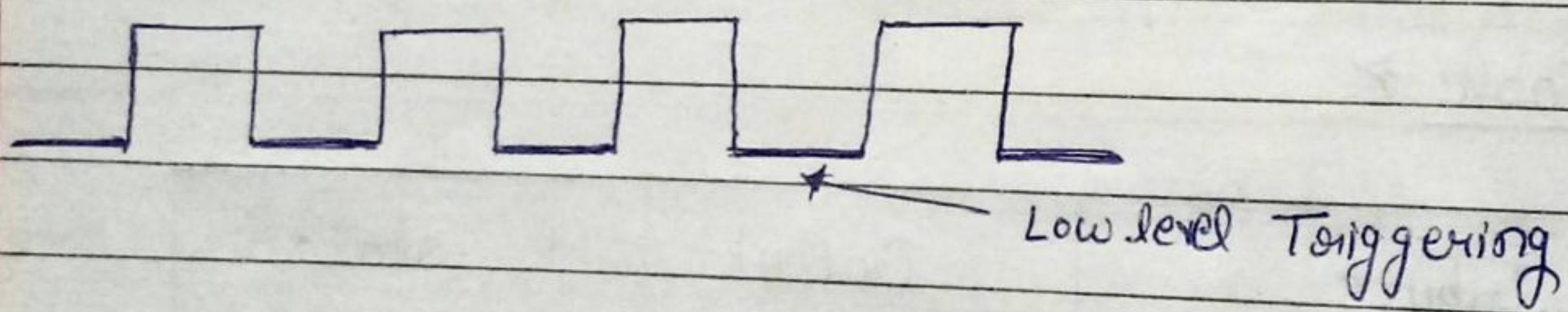
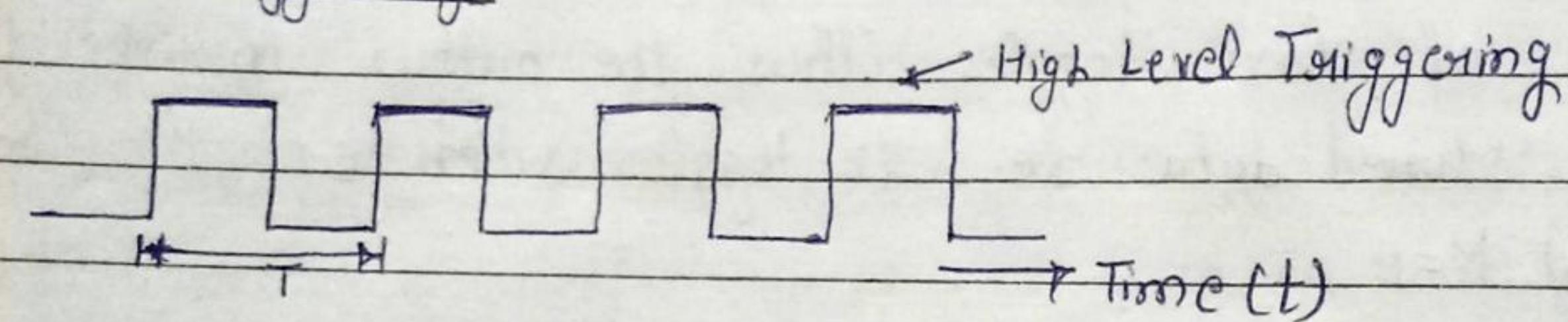
Clock is a digital signal in a square waveform.



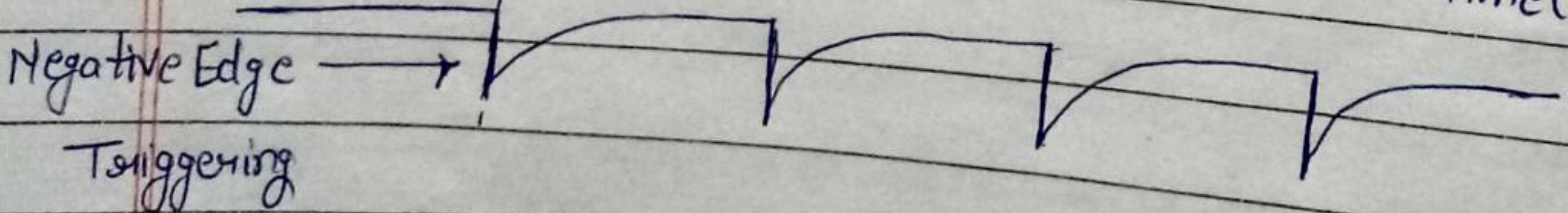
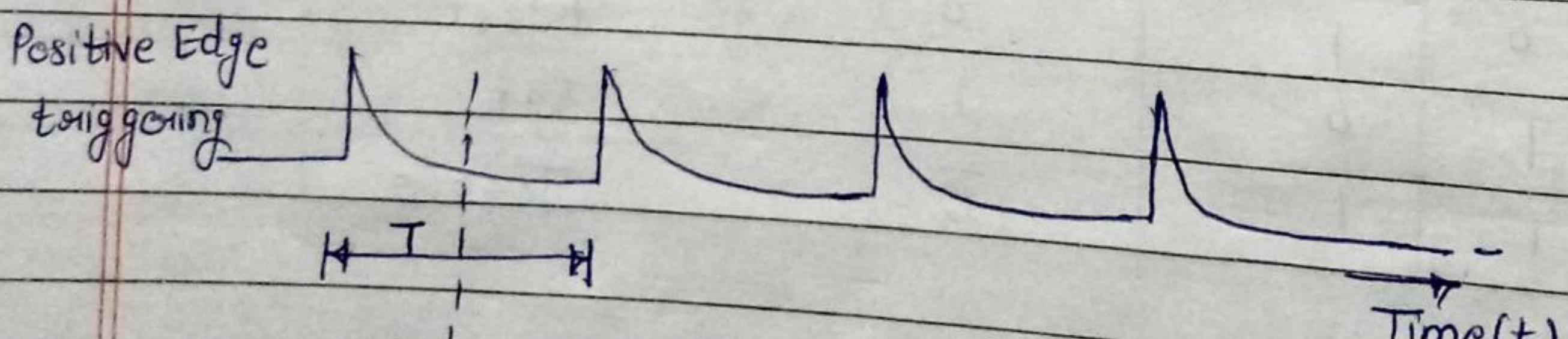
## Triggering of Flip-flop



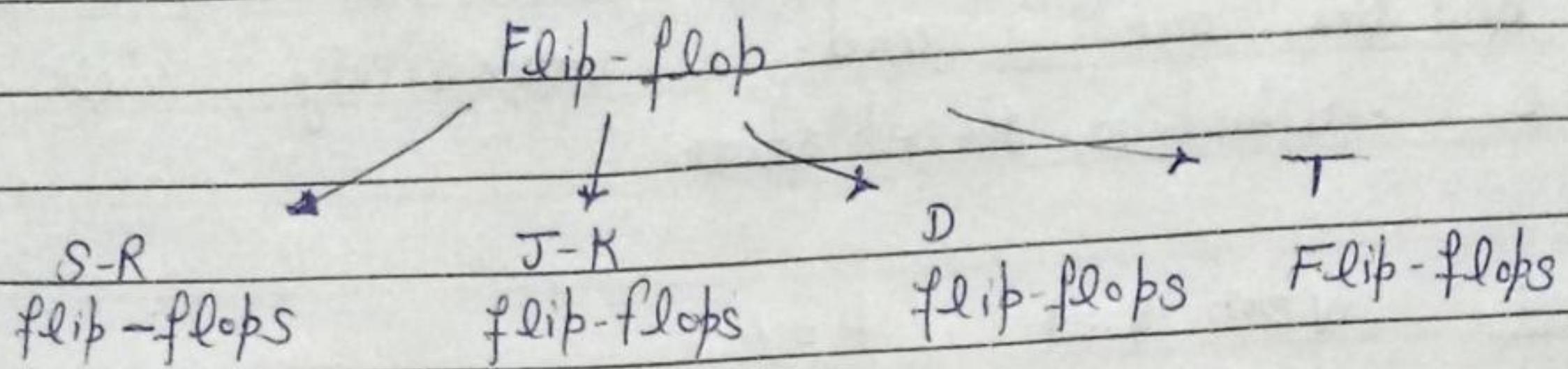
(a) Level Triggering →



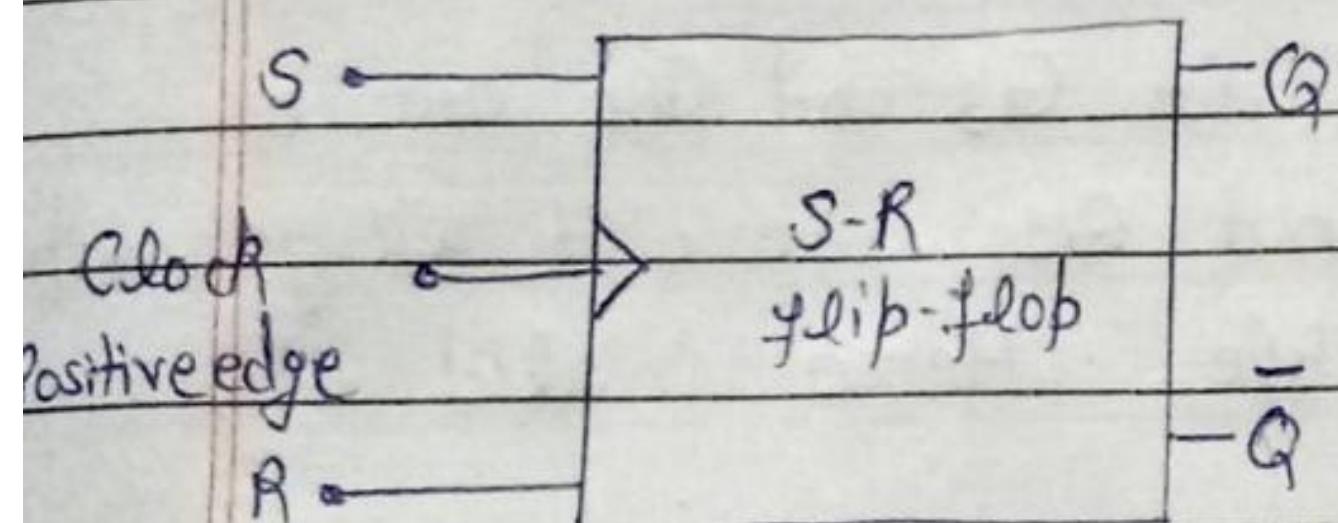
(b) Edge Triggering →



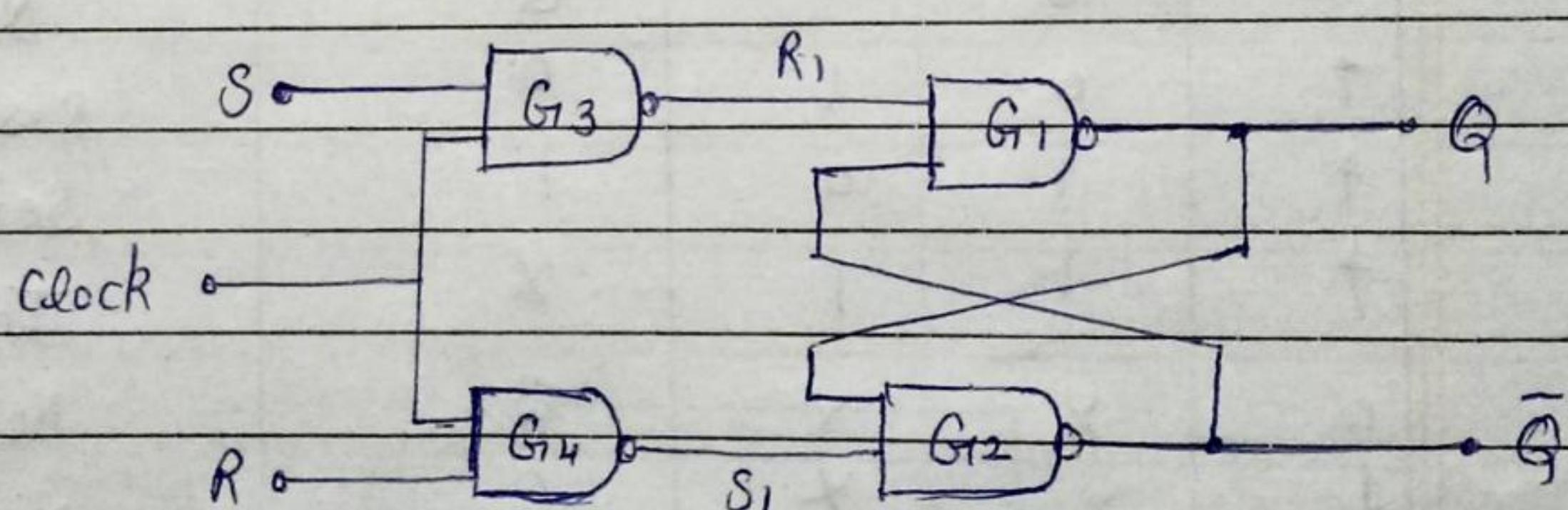
## "Types of Flip-flop"



1 S-R-flip flop +



## Logic Symbol



S-R - flip-flop

Case-1 & when  $s=0$  and  $R=0$

output of gates  $G_3$  and  $G_4$  are always 1. Thus 1 is input to Gates  $G_1$  and  $G_2$  which provides the output for no change.

Case-2 when  $S=0$   $R=1$

Output of gates  $G_3$  is always 1 and  $G_4$  is 0 as second input in clock. Thus the inputs for gates  $G_1$  and  $G_2$  are 1 and 0 respectively, we get output  $Q=0$ , which is Reset state.

Case-3 when  $S=1$   $R=0$

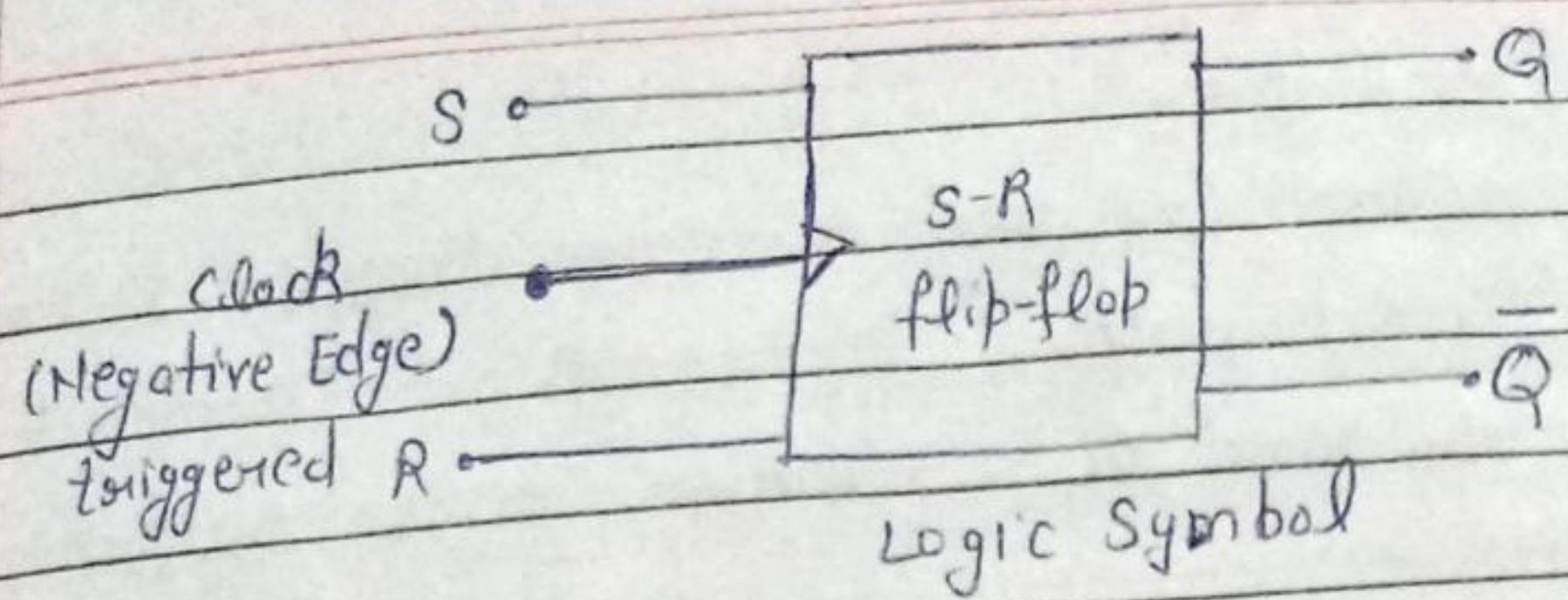
Output of gate  $G_4$  is always 1 and  $G_3$  is 0. Thus the inputs for gates  $G_1$  and  $G_2$  are 0 and 1 respectively, we get output  $Q=1$ , which is Set state.

Case-4 when  $S=1$  and  $R=1$

Output of both gates i.e.  $G_3$  and  $G_4$  are 0. Thus inputs for gates  $G_1$  and  $G_2$  are 0, we get output  $Q=\bar{Q}=1$ , which is not possible. Hence invalid state.

Inputs			Output	State	
Clk	S	R	Q	No change	
↑	0	0	Q	No change	Positive Edge
↑	0	1	0	Reset	
↑	1	0	1	Set	
↑	1	1	X	Invalid	
0	X	X	Q	No change	Flip-flop is disabled.
1	X	X	Q	No change	
↑	X	X	Q	No change	

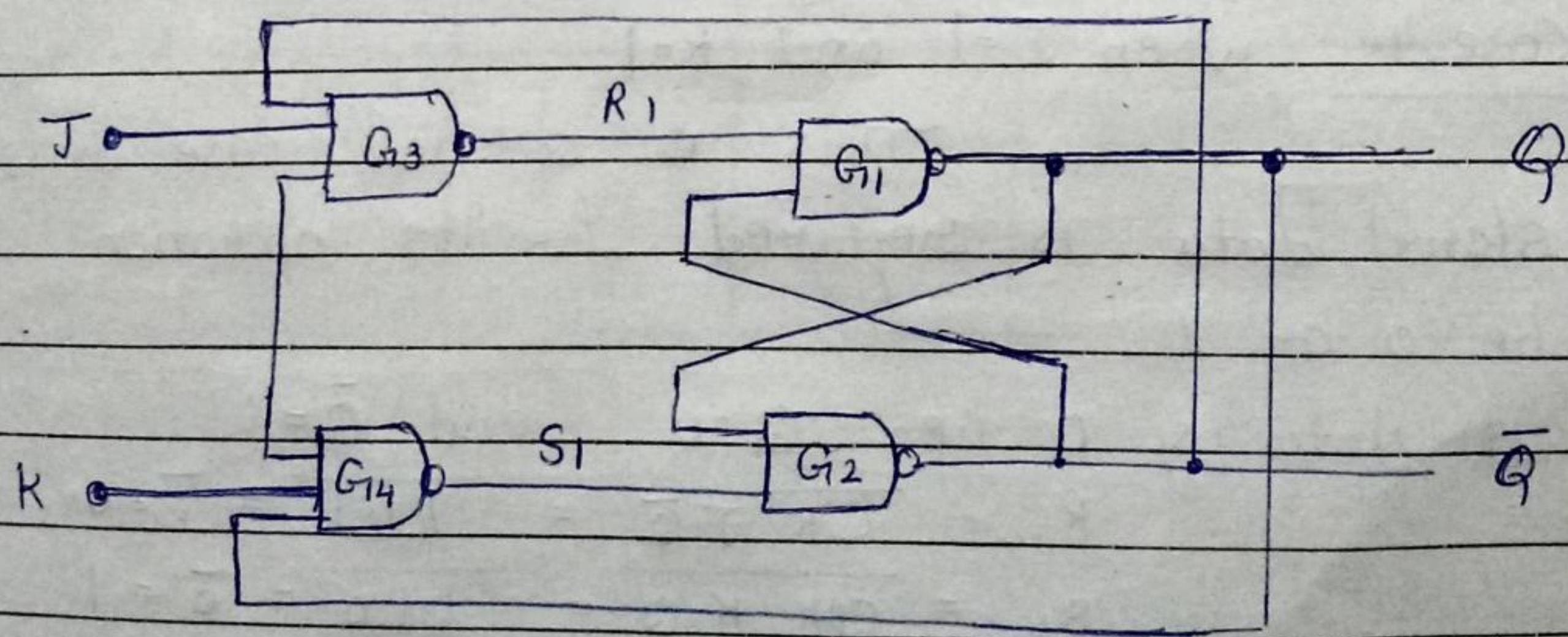
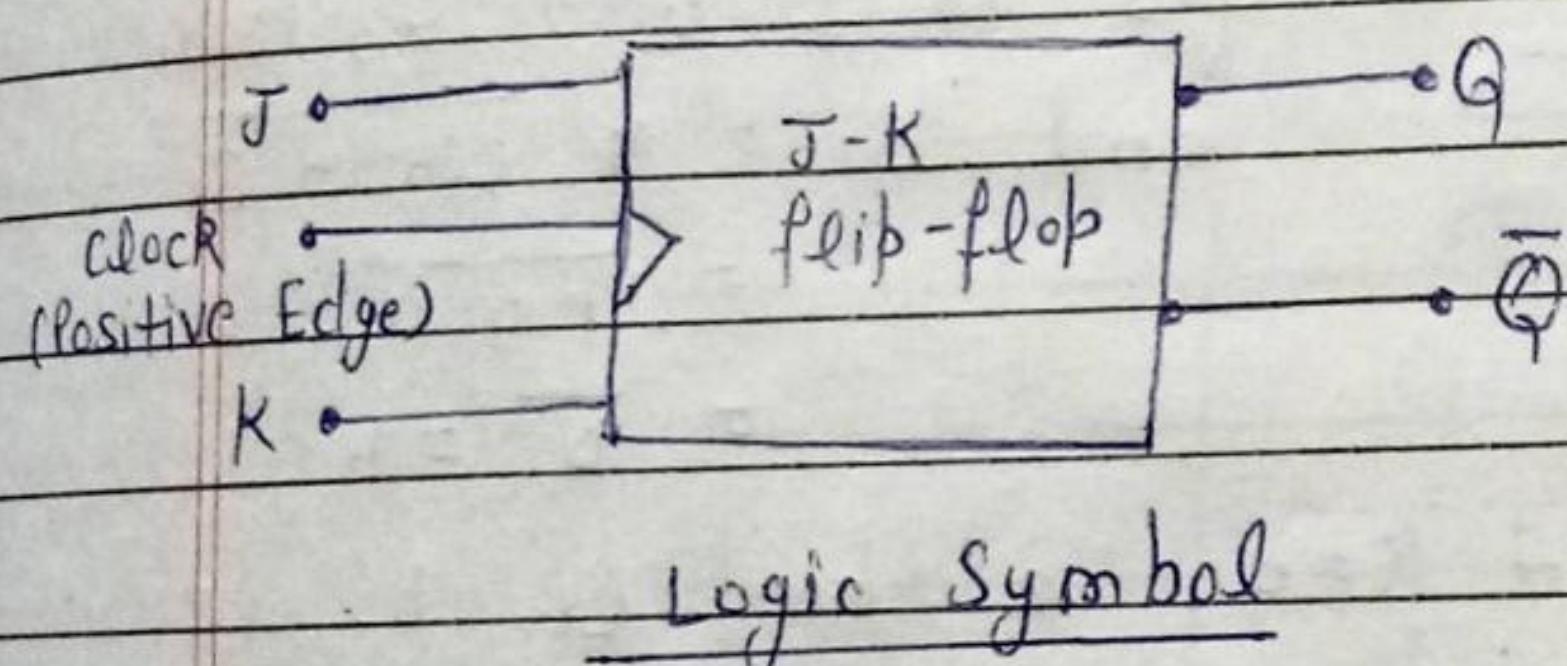
Similarly, the truth table for S-R flip-flop using negative edge triggered clock can be obtained. The only difference that it responds during the negative edge of the clock input.



## 2. J-K flip-flop +

The disadvantage of S-R flip flop is its invalid state i.e.  $Q = \bar{Q}$  which is not possible. To overcome this disadvantage J-K flip-flop designed.

The operation of J-K flip-flop is similar to that of S-R flip-flop except invalid state.



J-K flip-flop

Case-1 when  $J=0$  and  $K=0$

Output of gates  $G_3$  and  $G_4$  are always 1 irrespective of clock signal and output feedbacks. Thus 1 is input for gates  $G_1$  and  $G_2$ , which gives output for no change.

Case-2 when  $J=0$  and  $K=1$

output of  $G_3$  is always 1 and  $G_4$  is 0

$$\text{so } R_1 = \frac{\overline{\text{CK}} \cdot K \cdot \overline{J} \cdot \overline{Q}}{1 \cdot 0 \cdot 0} \quad \text{and } S_1 = \frac{\overline{\text{CK}} \cdot K \cdot Q}{1 \cdot 1 \cdot 1}$$

$$= \overline{0} = 1 \quad = \overline{1} = 0$$

therefore, when  $J=0$  and  $K=1$  output  $Q=0$  i.e. Reset state

Case-3 when  $J=1$  and  $K=0$

output of  $G_4$  is always 1 and  $G_3$  is 0

$$\text{so } R_1 = \frac{\overline{\text{CK}} \cdot K \cdot J \cdot \overline{Q}}{1 \cdot 1 \cdot 1} \quad \text{and } S_1 = \frac{\overline{\text{CK}} \cdot K \cdot Q}{1 \cdot 0 \cdot 0}$$

$$= \overline{1} = 0 \quad = \overline{0} = 1$$

therefore when  $J=1$  and  $K=0$  output  $Q=1$  i.e. Set state.

Case-4 when  $J=1$  and  $K=1$

This is special case in which previously stored data is required for its operation. That data may be 0 or 1.

If data is 0 i.e.  $Q=0$  and  $\overline{Q}=1$

$$R_1 = \frac{\overline{\text{CK}} \cdot K \cdot J \cdot \overline{Q}}{1 \cdot 1 \cdot 1} = \overline{1} = 0 = \overline{T} = 0$$

$$S_1 = \frac{\overline{\text{CK}} \cdot K \cdot Q}{1 \cdot 1 \cdot 0} = \overline{1} = 0 = \overline{0} = 1$$

Thus  $R_1=0$  and  $S_1=1$

from SR latch output  $Q=1$  i.e. state SET

If we take previous data is 1,  $Q=1$  and  $\bar{Q}=0$ .

$$R_1 = \overline{\text{COK} \cdot J \cdot \bar{Q}} = \overline{1 \cdot 1 \cdot 0} = \bar{0} = 1$$

$$S_1 = \overline{\text{COK} \cdot K \cdot Q} = \overline{1 \cdot 1 \cdot 1} = \bar{1} = 0$$

Thus  $R_1=1$  and  $S_1=0$

From SR latch output  $Q=0$  i.e. Reset state.

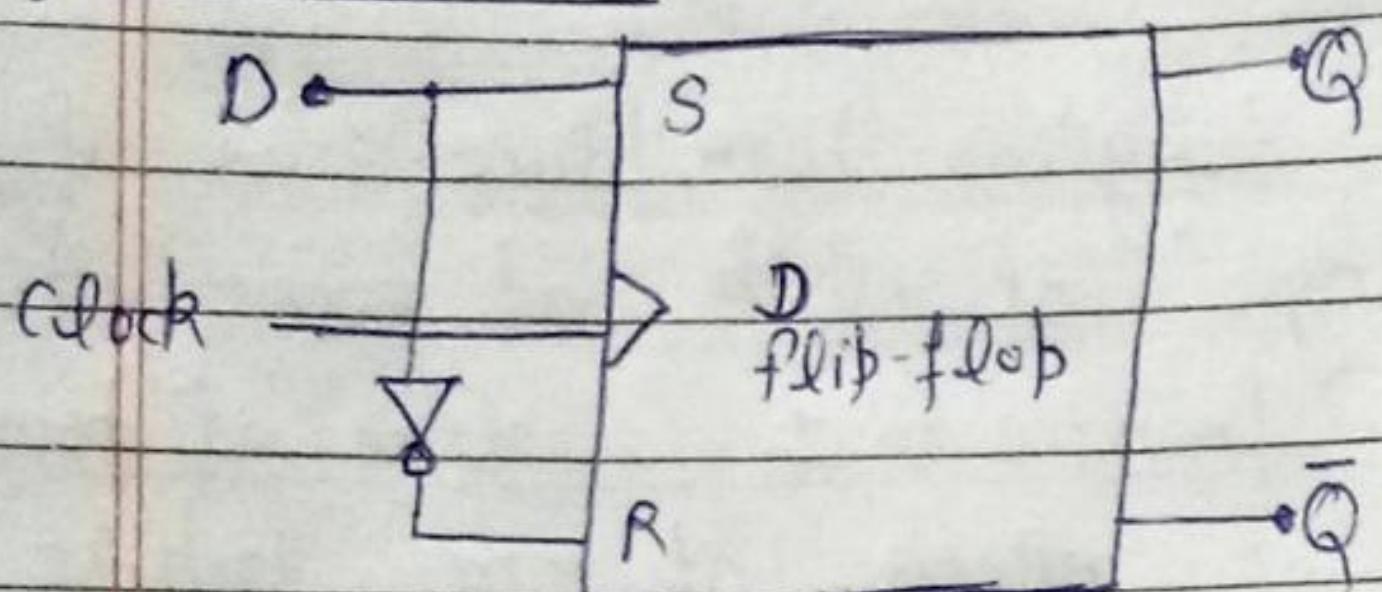
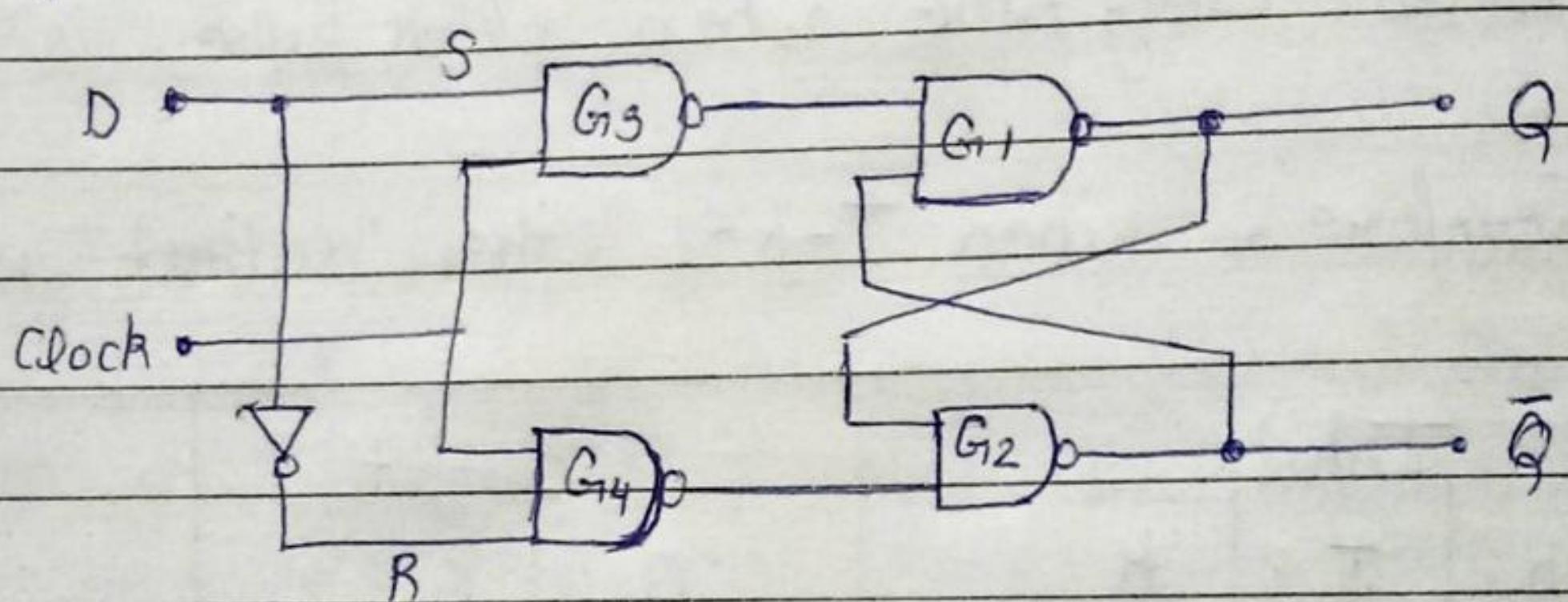
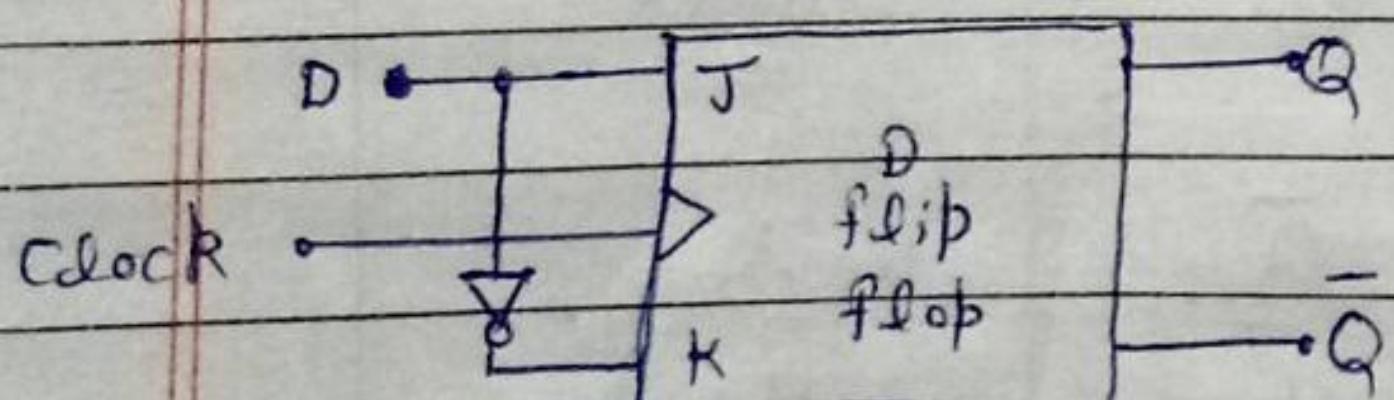
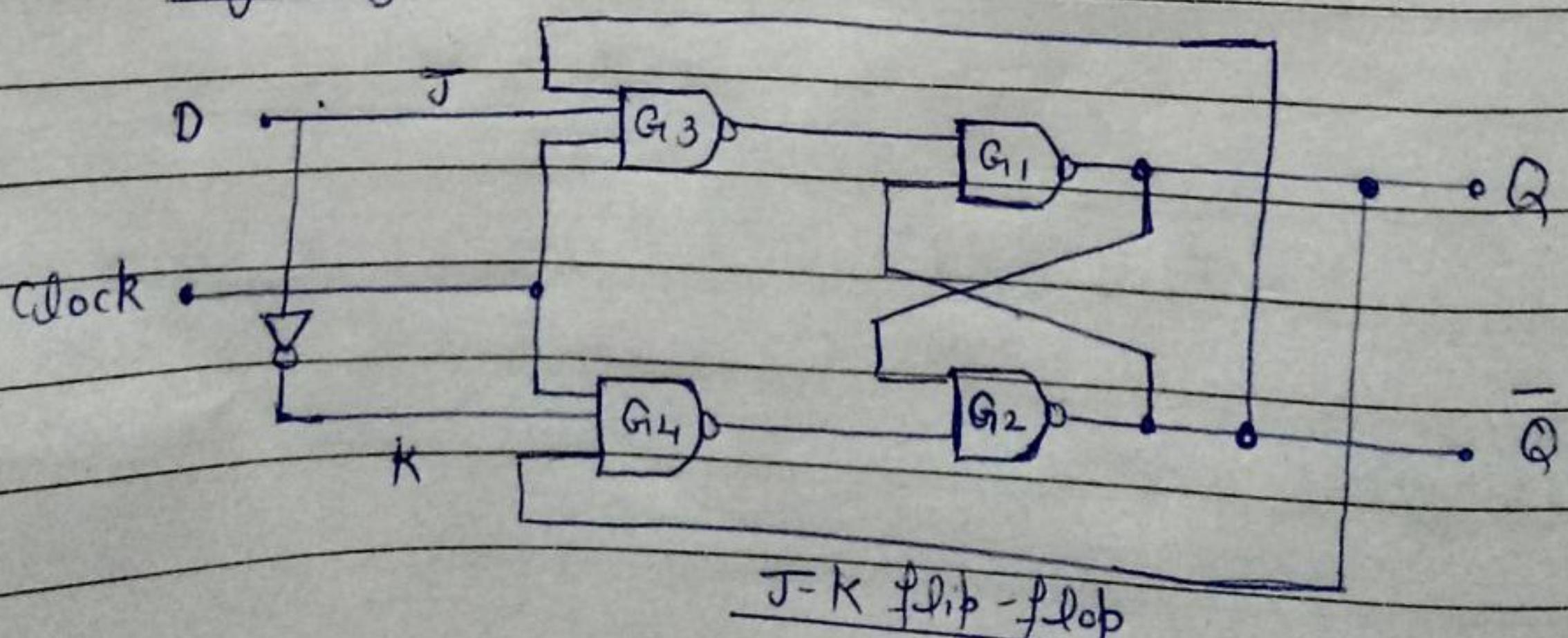
It is clear that when  $J=K=1$  toggling take place. i.e if previous data was 0 with the passage of clock at some inputs  $J=K=1$ , the outputs toggles from 0 to 1. Also if previous data was 1 with passage of clock at some inputs  $J=K=1$ , the output toggles from 1 to 0. Thus, outputs are inverted every time when clock pulse is provided.

Therefore, when  $J=K=1$ , the output is toggle state.

Input			Output		State
COK	J	K	Q	$\bar{Q}$	
↑	0	0	Q	$\bar{Q}$	No change
↑	0	1	0	1	Reset
↑	1	0	1	0	Set
↑	1	1	$\bar{Q}$	Q	Toggle
Positive Edge triggered clock					
0	X	X	Q	$\bar{Q}$	No change
1	X	X	$\bar{Q}$	Q	No change
↑	X	X	Q	$\bar{Q}$	No change
					No change

3. D flip-flop →

It can be designed from S-R flip-flop and J-K flip-flop by putting an inverter or NOT gate between S and R (or) J and K inputs and connected together to make a single input D.

Using S-R flip-flop →Logic SymbolD-flip-flopUsing J-K flip-flop →Logic SymbolJ-K flip-flop

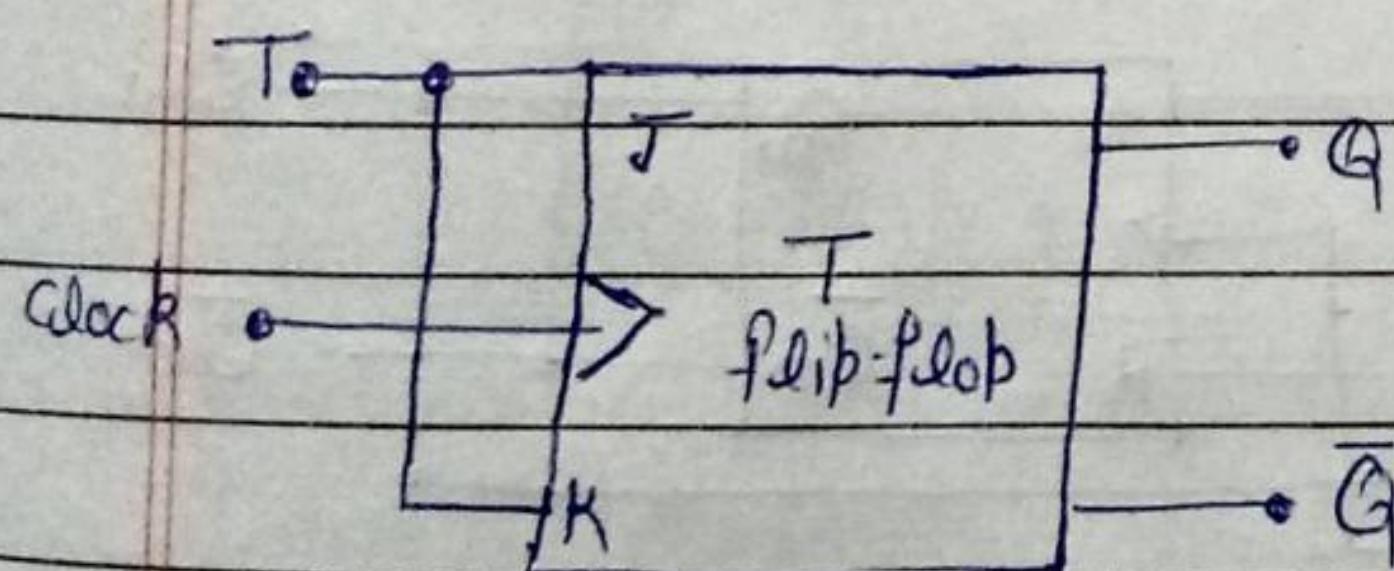
For  $D=0$  either  $S=0$  and  $R=1$  or  $J=0$  and  $K=1$  for this input output  $Q$  is always Reset i.e. 0, similarly for  $D=1$  either  $S=1$  and  $R=0$  or  $J=1$  and  $K=0$ , for this input  $Q$  is always set i.e. 1. Hence D is called Data input thus it is named as D-flip-flop.

Inputs		Output	state
clock	D	Q	
↑	0	0	Reset
↑	1	1	Set
0	X	Q	No change

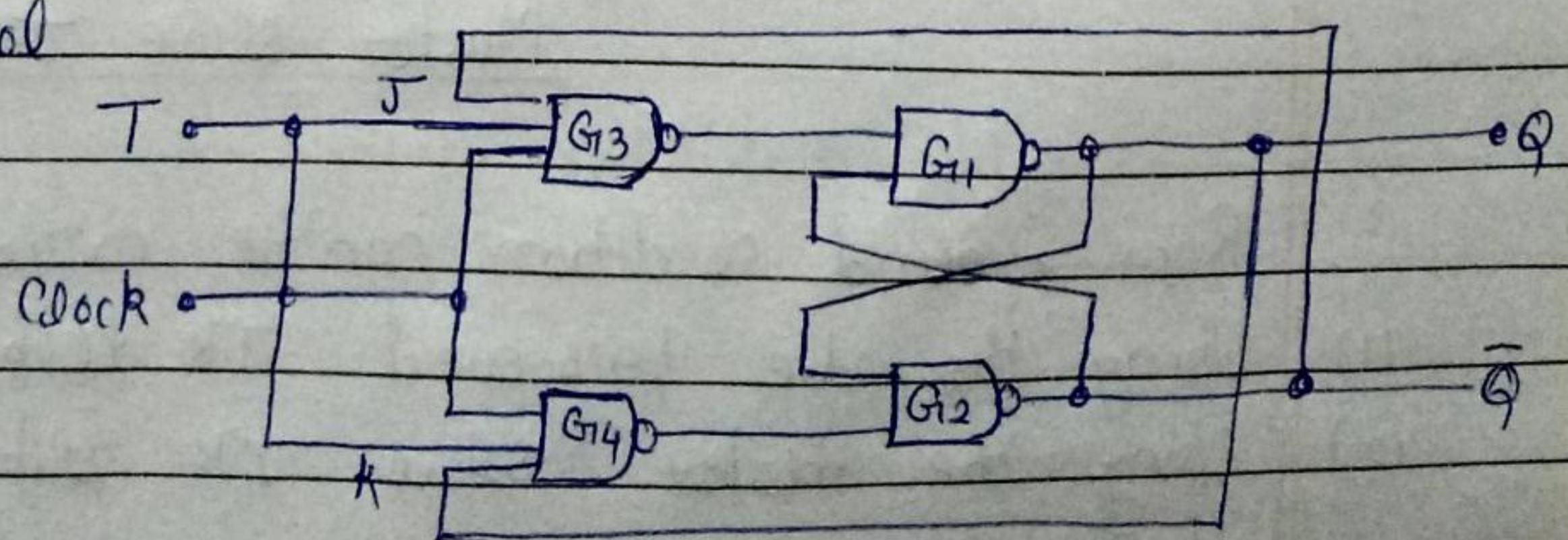
It is similar to D-latch. It is important because if we want to store 0 and 1 in the flip-flop then, some inputs are applied to D. Thus it is very useful in register for storage of data.

#### 4. T flip-flop + (TOGGLE FLIP-FLOP)

It can be designed using JK flip-flop.



Logic symbol



T-flip-flop

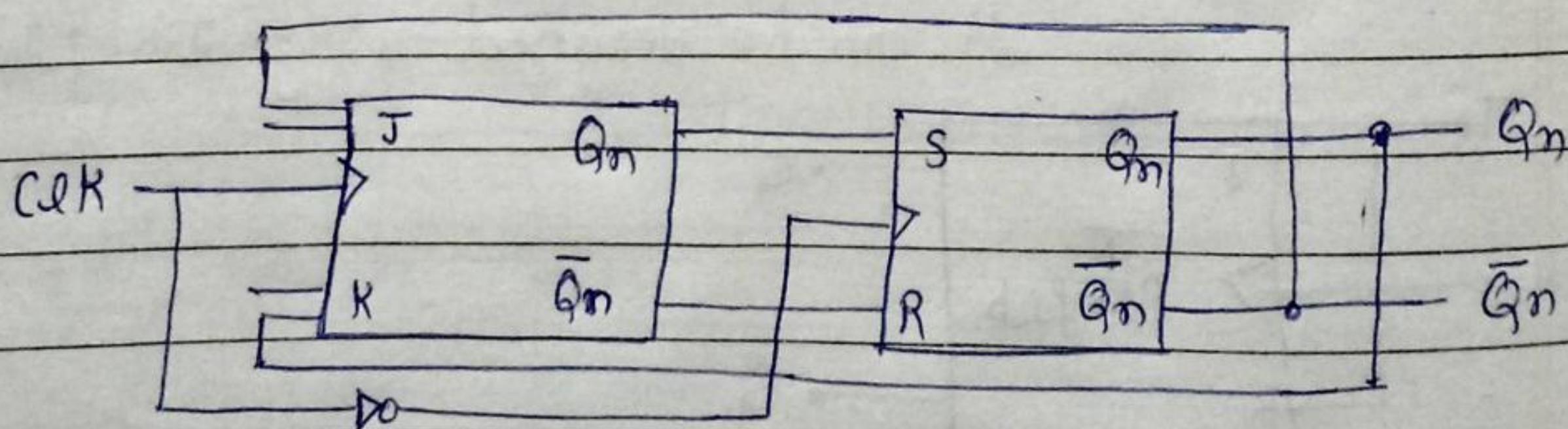
Its operation is similar to the inputs of J-K flip-flop with  $J=K=0$  or  $J=K=1$  as inputs.

Due to toggling property it is known as toggle flip-flop and are extensively used in counters to count the pulse.

Input		Output	State
CLK	T	Q	
↑	0	Q	No change
↑	1	$\bar{Q}$	Toggle
0	X	Q	No change

### "Race Around Condition"

For J-K flip-flop if  $J=K=1$  and if clock is too long then state of flip-flop keep on toggle which leads to uncertainty in determining output state of flip-flop. This condition is called "Race around condition".



Master slave J-K flip-flop.

Race around condition can be avoided by -

- (1) Using the edge triggered JK flip flop.
- (2) Using the master slave JK flip flop.

## Characteristic Table

OR  
Truth Table and Excitation Table of All flip-flop +

For S-R flip-flop +

Input		Output		Present state	Next state	Inputs	
S	R	$Q_n$	$\bar{Q}_n$	$Q_n$	$Q_{n+1}$	S	R
0	0	no change		0	0	0	x
0	1	0 (Reset)		0	1	1	0
1	0	1 (Set)		1	0	0	1
1	1	intermediate int.		1	1	x	0

Truth Table

Excitation Table

For J-K flip-flop +

Input		Output		Present	Next state	Inputs	
J	K	$Q_n$	$\bar{Q}_n$	$Q_n$	$Q_{n+1}$	J	K
0	0	No change	No change	0	0	0	x
0	1	0 (Reset)	1	0	1	1	x
1	0	1 (Set)	0	1	0	x	1
1	1	Toggle	Toggle	1	1	x	0

Truth Table

Excitation Table

For D flip-flop +

Input		Output		Present state	Next state	Input
D		$Q_n$	$\bar{Q}_n$	$Q_n$	$Q_{n+1}$	D
0		0	1	0	0	0
1		1	0	0	1	1
				1	0	0
				1	1	1

Truth Table

Excitation Table

for T-Flip-flop →

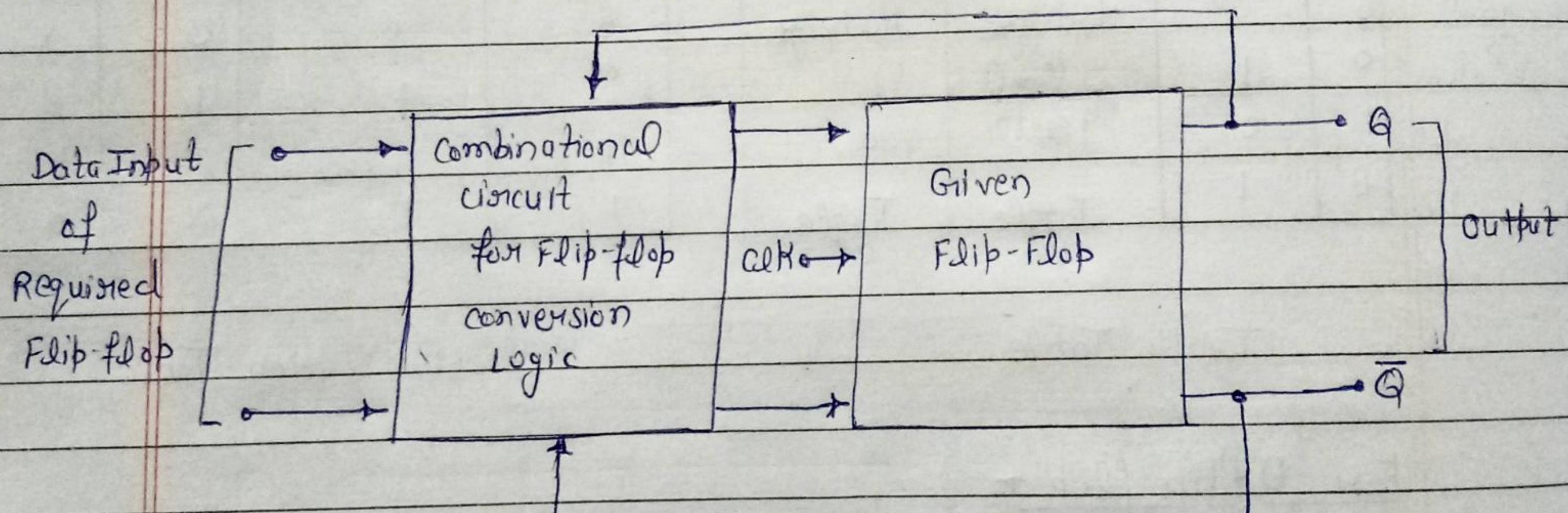
Input		Output		Present state		Next state		Input	
T		$Q_n$	$\bar{Q}_n$	$Q_n$		$Q_{n+1}$		T	
0		No change	No change	0		0		0	
1		Toggle	Toggle	0		1		1	

Truth Table

Excitation Table.

### Flip-Flop conversion →

conversion of one flip-flop into another flip-flop is possible by adding some extra gates or connections.



## I. Conversion of SR Flip-Flop to JK Flip-Flop +

Step-1 Write Truth Table for SR to JK +

Inputs			Output		
J	K	presentstate Q	Nextstate Qn+1	S	R
0	0	0	0	0	X
0	1	0	0	0	X
1	0	0	1	1	0
1	1	0	1	1	0
0	1	1	0	0	1
1	1	1	0	0	1
0	0	1	1	X	0
1	0	1	1	X	0

← Excitation Table of JK FF →

↑ ← Excitation Table of SRFF → ↑

## Step-2 K-mop +

For output S

J	KQn	00	01	11	10
0	0	X	0	0	
1	-D	X	0	(1)	

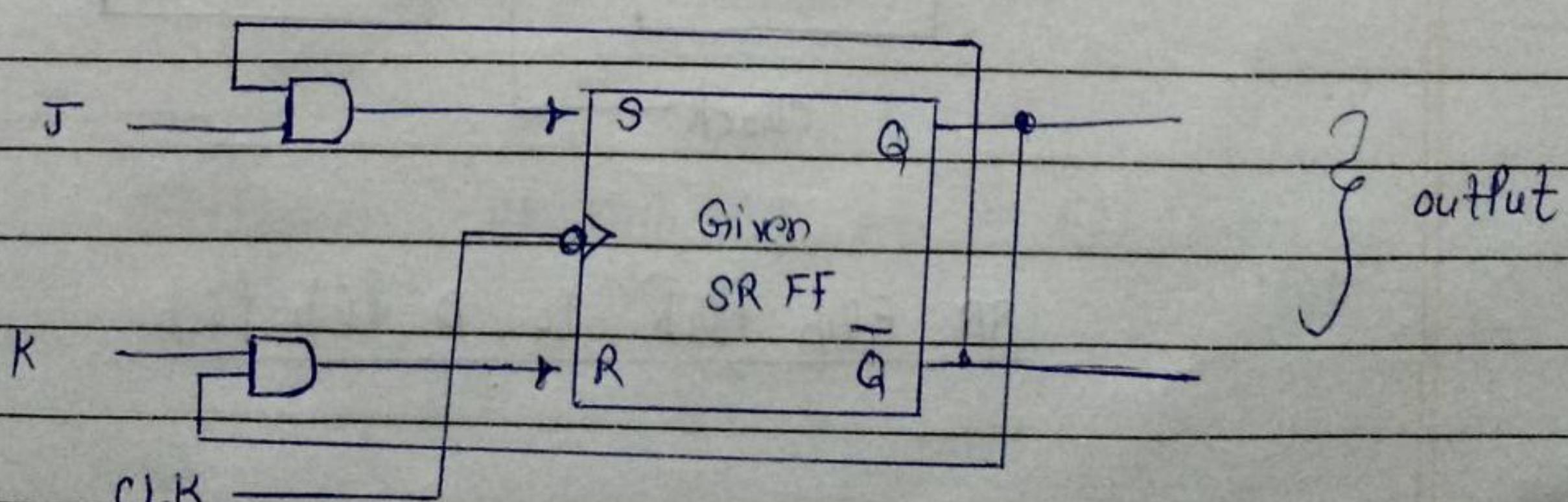
$$S = J \bar{Q}_n$$

For Output R

J	KQn	00	01	11	10
0	X	0	(1)	X	
1	0	0	(1)	0	

$$R = K Q_n$$

## Step-3 Logic Diagram +



S-R to J-K Flip-flop conversion

Conversion from S-R flip-flop to D flip-flop +

Step-1

		Inputs		Outputs	
D	Present state $Q_n$	Next State $Q_{n+1}$	S	R	
0	0	0	0	X	
1	0	1	1	0	
0	1	0	0	1	
1	1	1	X	0	

+ Entries from Excitation table of D +

+ Entries from Excitation of SR FF +

Step-2

K-map +

For S output

$Q_n$		D	
0	1	0	1
0	0	0	
1	1	X	

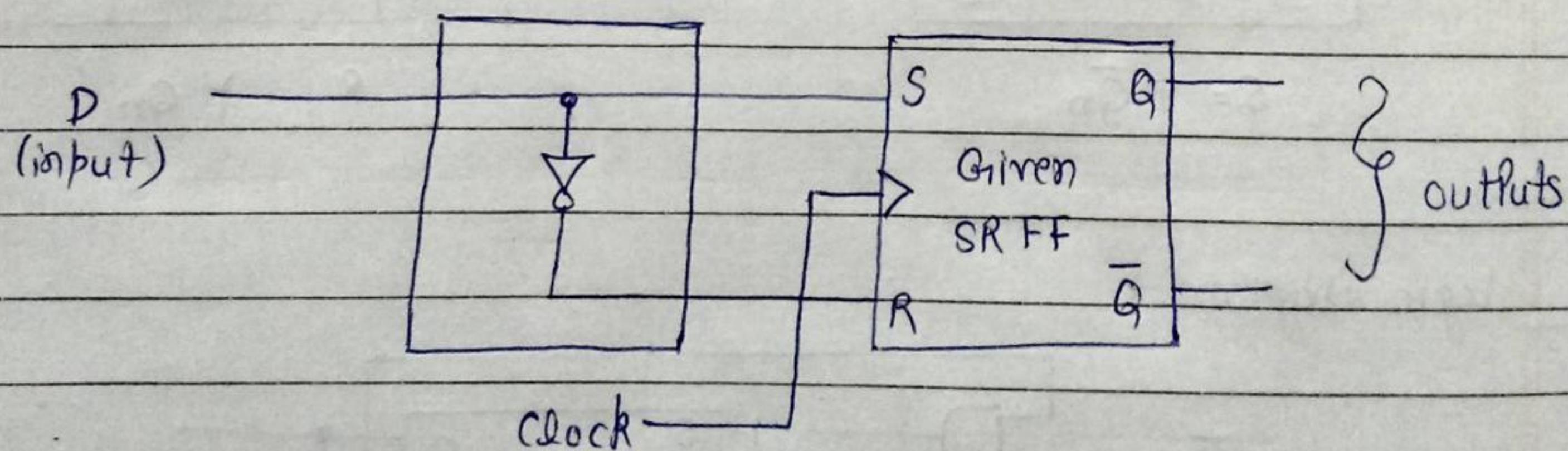
For R output

$Q_n$		D	
0	1	X	1
0	0		
1	0		

$$\therefore R = \overline{D}$$

$$\therefore S = D$$

Step-3 LOGIC DIAGRAM +



SR Flip Flop to D flip-flop.

Conversion of S-R flip-flop to T flip-flop →

Step-1 Truth Table →

T input			Outputs	
T	Present state $Q_n$	Next state $Q_{n+1}$	S	R
0	0	0	0	X
1	0	1	1	0
1	1	0	0	1
0	1	1	X	0

Step-2 K-map →

for S output

$\bar{T} \bar{Q}_n$	0	1
0	0	X
1	1	0

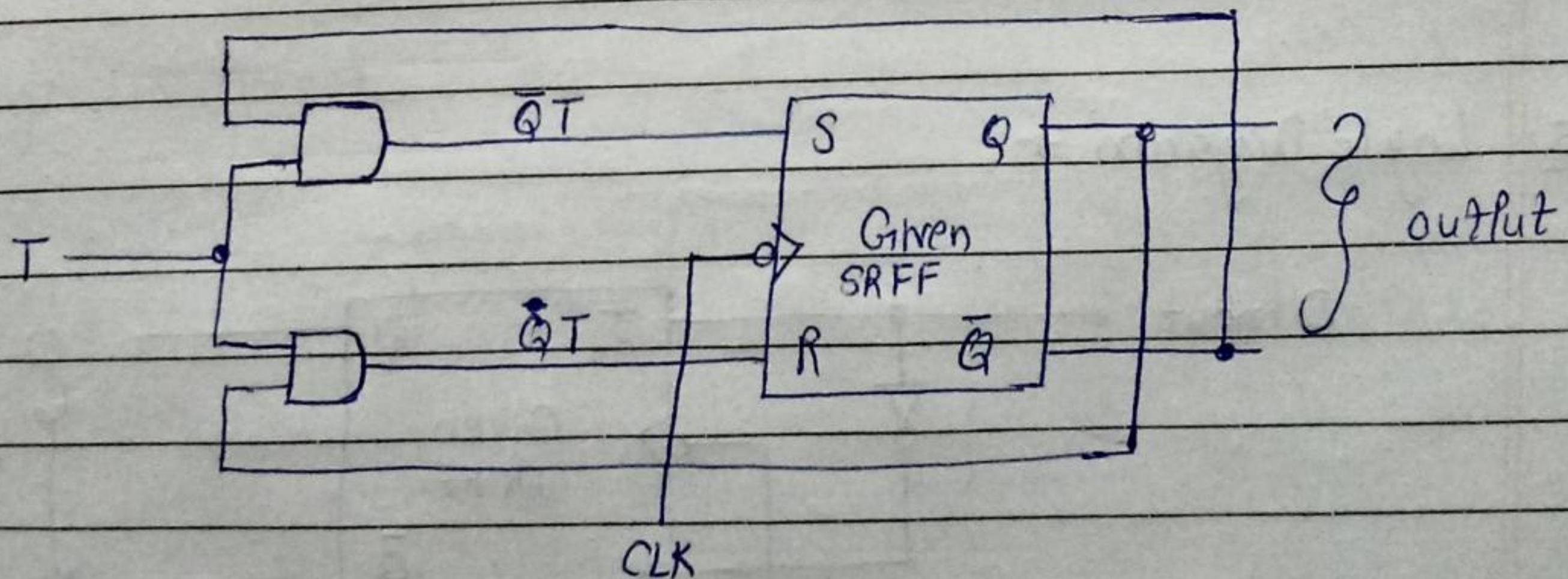
$$S = T \bar{Q}_n$$

For R output

$T \bar{Q}_n$	0	1
0	X	0
1	0	1

$$R = T Q_n$$

Step-3 Logic diagram →



Conversion SR Flip Flop to T Flip Flop

## "Conversion of J-K flip-flop to D flip-flop"

Step-1 Truth Table →

Inputs			Outputs	
D	Previous state Q <sub>n</sub>	Next State Q <sub>n+1</sub>	J	K
0	0	0	0	X
1	0	1	1	X
0	1	0	X	1
1	1	1	X	0

→ Excitation Table of D FF

← Excitation table of J-K FF

Step-2 K-map +

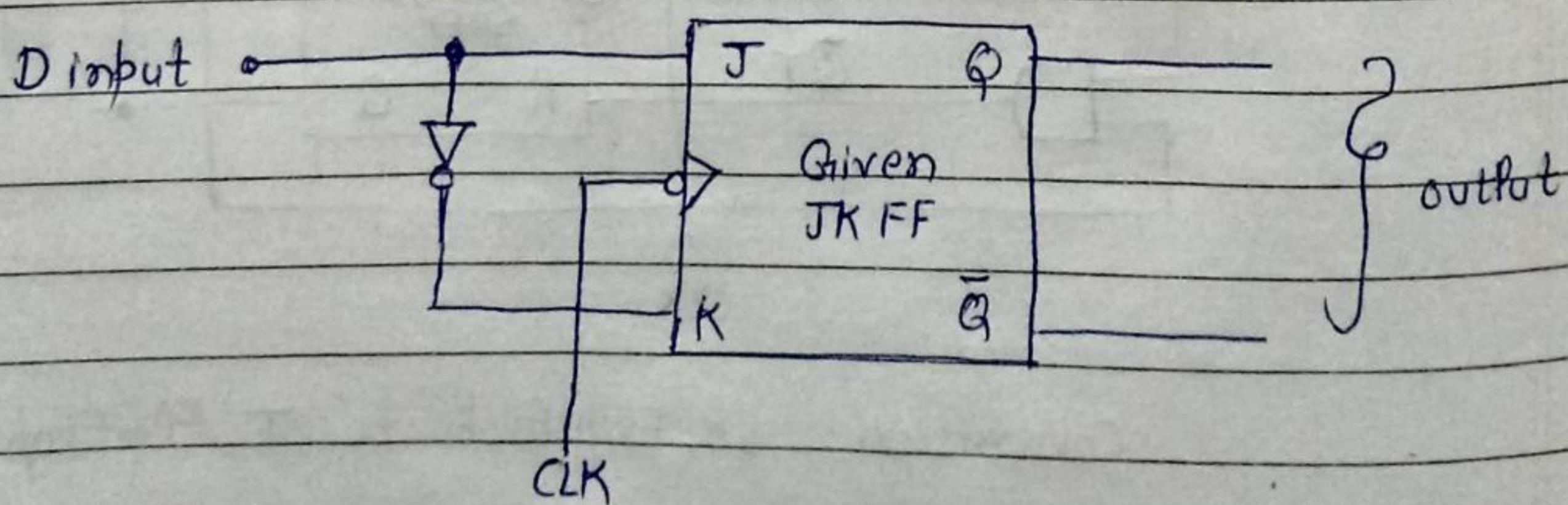
For J output	
D	Q <sub>n</sub>
0	0 X
1	(1) X

$$J = D$$

For K output	
D	Q <sub>n</sub>
0	(X) 1
1	X 0

$$R = \overline{D}$$

Step-3 Logic Diagram +



Conversion from JK flip flop to D FF

## Conversion of J-K Flip-flop to T Flip-flop

### Step-1 Truth Table +

Inputs			Output	
T	Present State Q	Next State Q	J	K
0	0	0	0	X
1	0	1	1	X
1	1	0	X	1
0	1	1	X	0

← Excitation table of T FF →

← Excitation Table of J-K FF →

### Step-2 K-map +

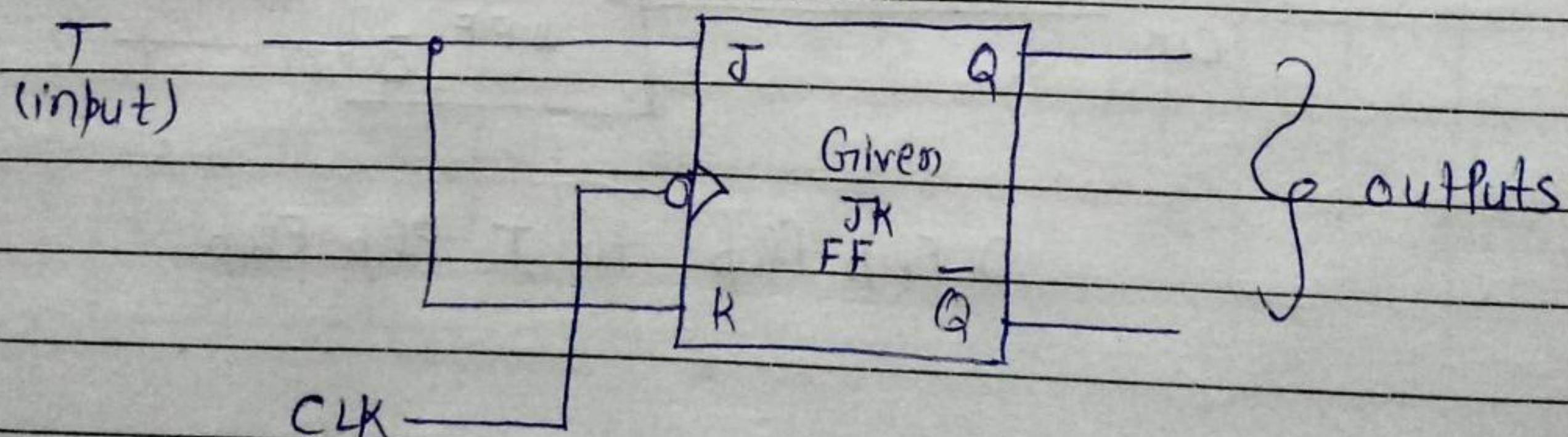
For J output →

		Q <sub>n</sub>
		0 1
0		0 X
1		(1 X)
J = T		

For K output →

		Q <sub>n</sub>
		0 1
0		X 0
1		(X 1)
K = T		

### Step-3 Logic diagram +



Logic diagram for JK FF to T FF.

~~Ans~~

## "Conversion to D flip-flop to T Flip-flop"

### Step-1 Truth Table +

Inputs			Output
T	Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	D
0	0	0	0
1	0	1	1
1	1	0	0
0	1	1	1

### Step-2 K-map +

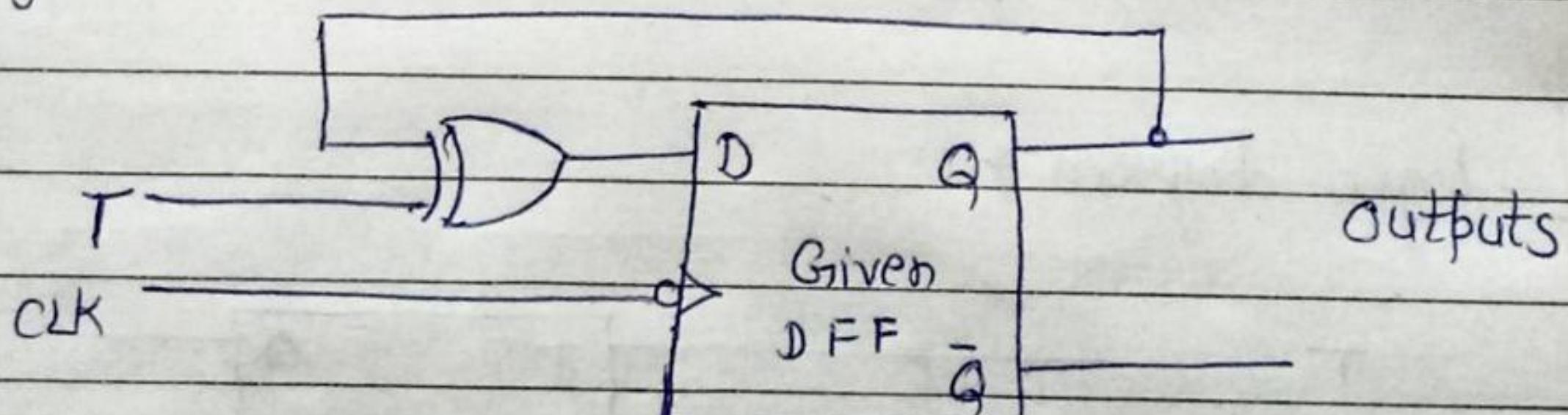
For output D →

$T \setminus Q_n$

	0	1
0	0	(1)
1	(1)	~0

$$\begin{aligned} D &= T\bar{Q}_n + \bar{T}Q_n \\ &= T \oplus Q_n \end{aligned}$$

### Step-3 Logic Diagram +



D Flip-flop to T Flip-flop.

### Applications of Flip-Flop +

- These are used in registers where transfer of data take place from one flip-flop to another.

2. These are used in counters and timers.
3. These are used in frequency division and delay element.
4. These are used in the elimination of keyboard debounce.
5. These are used for storage of data or information.

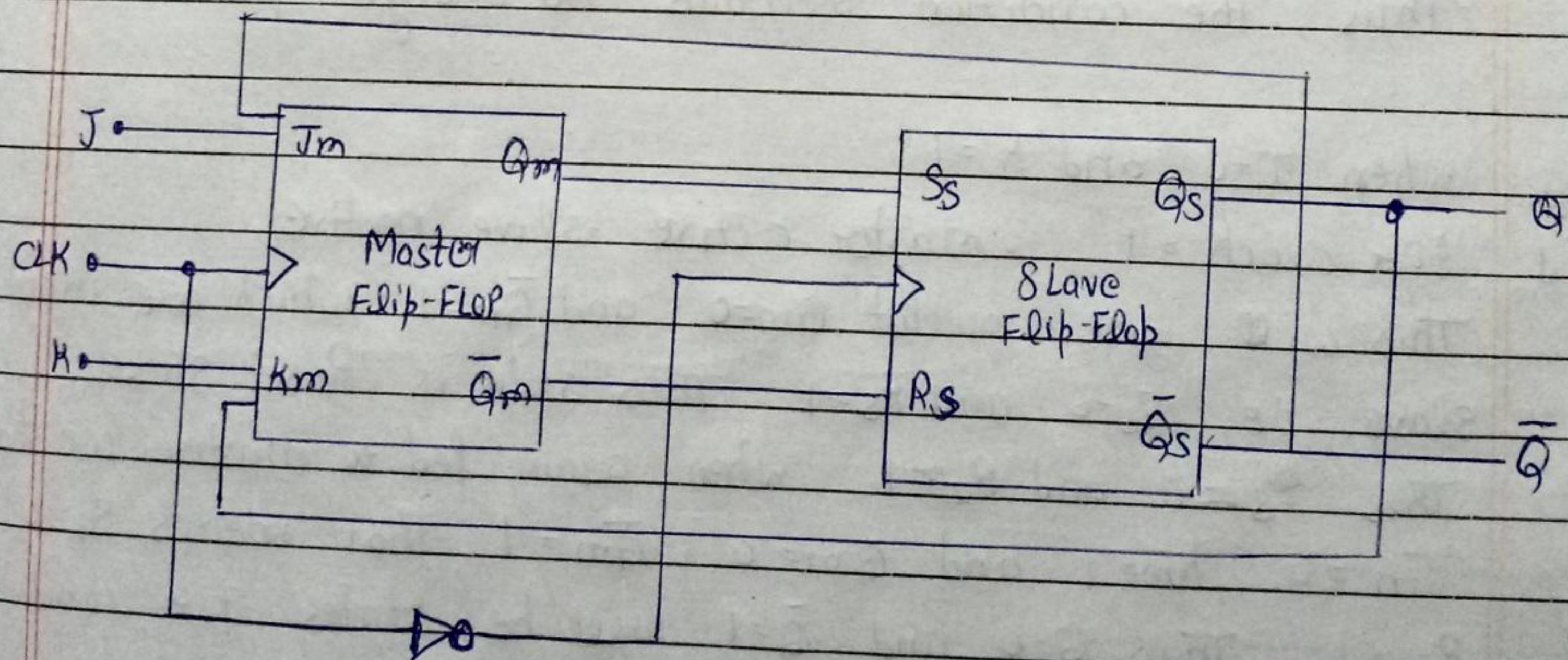
### Master Slave Flip-Flop

A master slave flip-flop is constructed from two flip-flops. One flip-flop acts as a master and other acts as slave. Slave follows output of master. It is used to convert the level triggered flip-flop to edge triggered flip-flop. Thus used to eliminate race around condition.

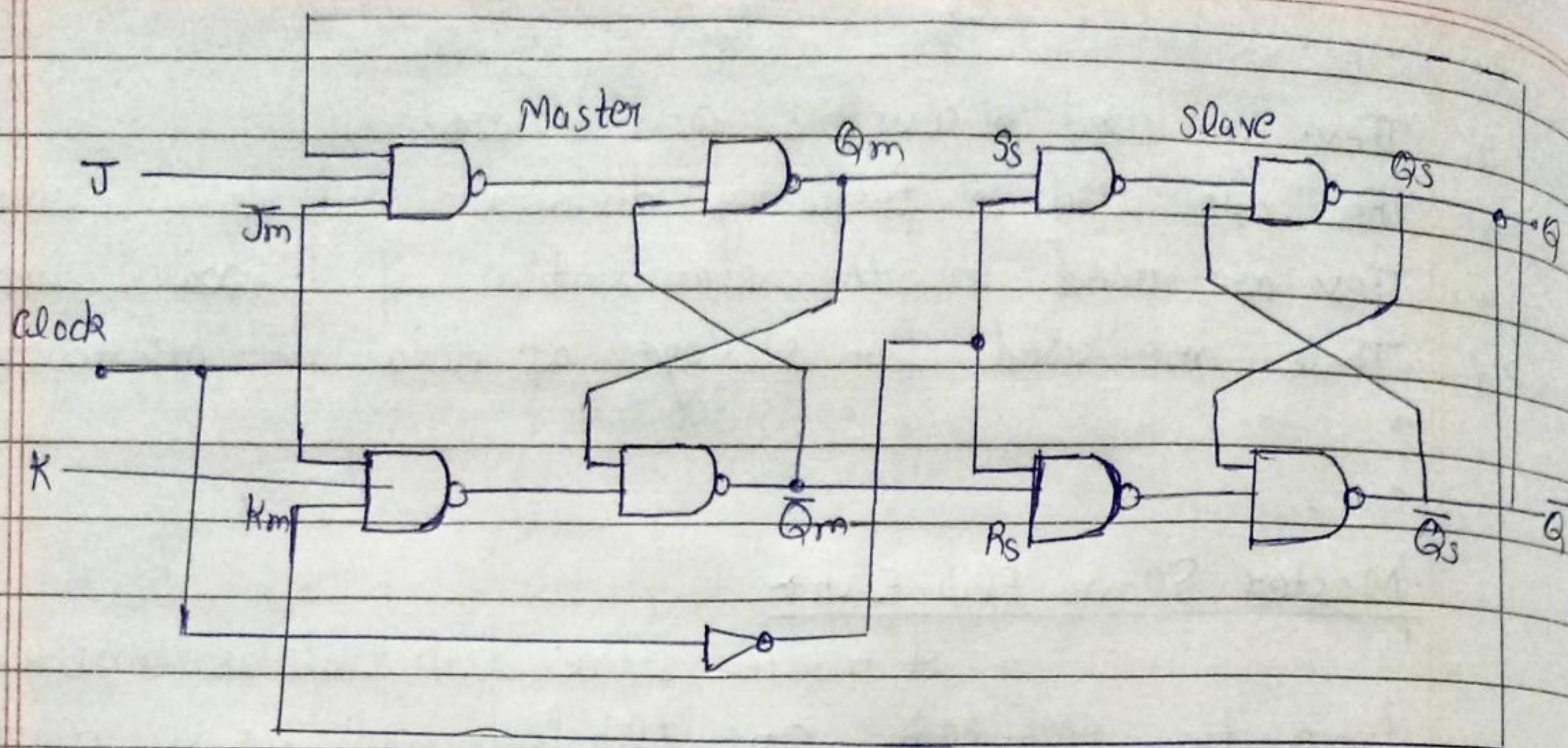
There are three basic types of master slave flip-flops - i.e. S-R, D, JK master-slave flip-flops. Mostly JK slave flip-flop is used because of its easy availability in integrated circuit forms.

~~Ans~~

### "JK Master Slave JK flip-flop"



Block diagram of JK Master slave Flip-Flop



J-K-Master Slave Flip-Flop

Case-1 If  $J=0, K=0$

1. For  $\text{clock} = 1$ , master is active, slave inactive. During  $J=K=0$ , the output of master will not change, as there is no change condition when  $J=K=0$  in J-K flip-flop.
2. For  $\text{clock} = 0$ , the slave is active and master is inactive. Thus, the condition remains No change for  $J=K=0$ .

Case-2 when  $J=0$  and  $K=1$

- 1. For  $\text{clock} = 1$ , master active, slave inactive. This will give output  $Q_m=0$  and  $\bar{Q}_m=1$  which are input for slave i.e.  $S_s=0$  and  $R_s=1$ . This state is Reset state. Thus  $S_s=0$  and  $R_s=1$  when again fed to master, we have  $J_m=0$ ,  $K_m=1$  and  $Q_m=0$ ,  $\bar{Q}_m=1$ . That means  $S_s=0$  and  $R_s=1$ . Thus  $Q=0$  and  $\bar{Q}=1$  will be states for condition  $J=0$  and  $K=1$ .

Case-3 When  $J=1$  and  $K=0$

for  $\text{clock} = 1$  the master is active and slave is inactive

This will give output  $Q_m=1$  and  $\bar{Q}_m=0$  which are inputs for slave i.e.  $S_s=1$  and  $R_s=0$ . This is set state. Thus  $S_s=1$  and  $R_s=0$  when again fed to master we have  $J_m=1$ ,  $K_m=0$  and  $Q_m=1$ ,  $\bar{Q}_m=0$  that means  $S_s=1$  and  $R_s=0$  thus  $Q=1$  and  $\bar{Q}=0$  will be the states for condition  $J=1$  and  $K=0$ .

Case-4 when  $J=1$  and  $K=1$

For clock = 1, the master is active and slave is inactive. This will give output of master in toggle form i.e. previous data will invert. The state is Toggle state. Thus S and R data is also inverted.

For clock = 0, the master is inactive and slave will be active, again the output of slave toggles. Thus the output inverts from previously stored data i.e. it toggles whenever the input are  $J=1$  and  $K=1$ .

Truth Table ↗

Inputs			Output	state
CLK	J	K	Q	
↑↓	0	0	Q	No change
↑↓	0	1	0	Reset
↑↓	1	0	1	Set
↑↓	1	1	$\bar{Q}$	Toggle

## # Difference b/w LATCH and Flip-Flop \*

## Flip-Flop

## LATCH

- | Flip-Flop   | LATCH  |
|---|--|
| 1. Flip-flop is a device i.e. it has two stable states that represented as 0 and 1. | 1. Latch is also a bistable device whose states are also represented as 0 and 1. |
| 2. It is edge triggered device.   | 2. It is a level triggered device.   |
| 3. They are classified into asynchronous or synchronous flip-flops.                 | 3. There is no such classification in latches                                    |
| 4. Flip-flop can be build from Latches.   | 4. LATCHES can't build from Gates  |
| 5. Gates like NOR, NOT, AND are building blocks of flip-flops.                      | 5. These are also made up of gates.  |
| 6. Ex- D Flip-flop and J-K flip-flop.   | 6. Ex- SR Latch - D latch.   |

"THANK - YOU"