⇒ LATCH:- → it stores '0' and '1' in memory

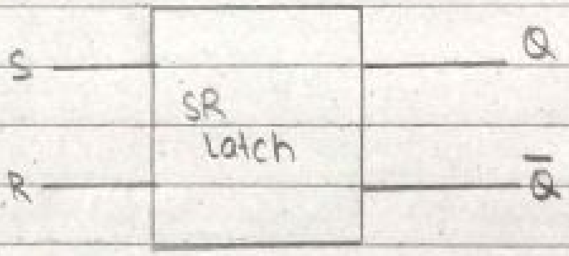
The basic storage element is called latch. As the name suggests it latches '0' or '1'.

⇒ SR LATCH:- → outputs are always complement of each other.

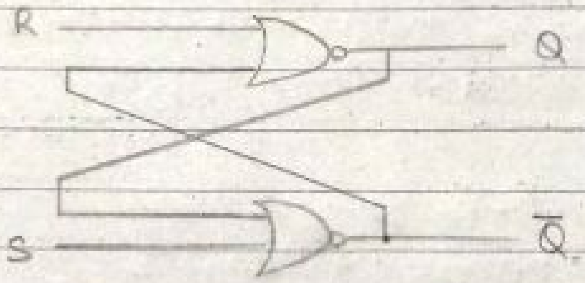
Set Reset latch.

⇒ There are two types of SR latch.

- NOR SR LATCH
- NAND SR LATCH.



⇒ NOR SR LATCH:-



⇒ When Set Q=1
⇒ When Reset Q=0

* NOR SR latch is also called "active high input SR Latch."

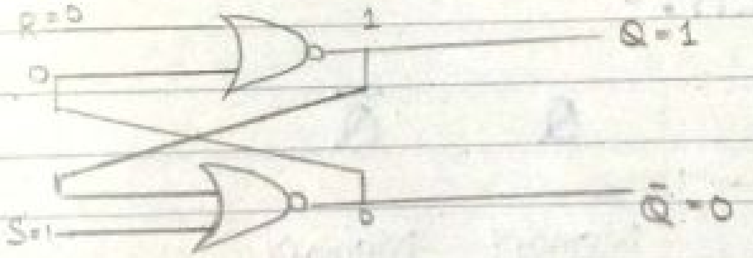
TRUTH TABLE OF NOR GATE:-

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

0

CASE : 01

When S=1, R=0

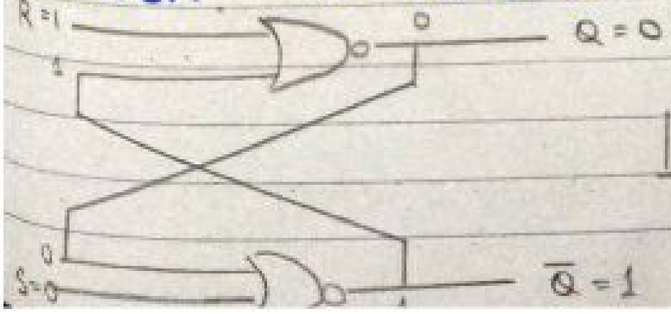


Q=1, $\bar{Q}$=0

Memory = S=0, R=0

CASE : 02

When R=1, S=0

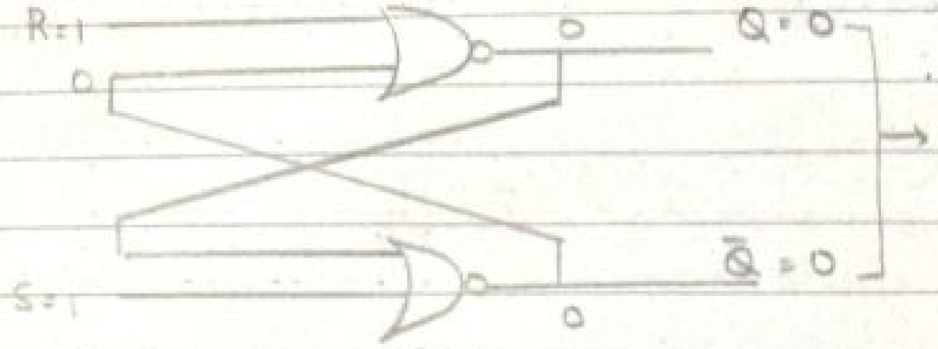


=> Q=0, $\bar{Q}$=1

=> Memory = S=0, R=0

CASE: 03

When S=1, R=1



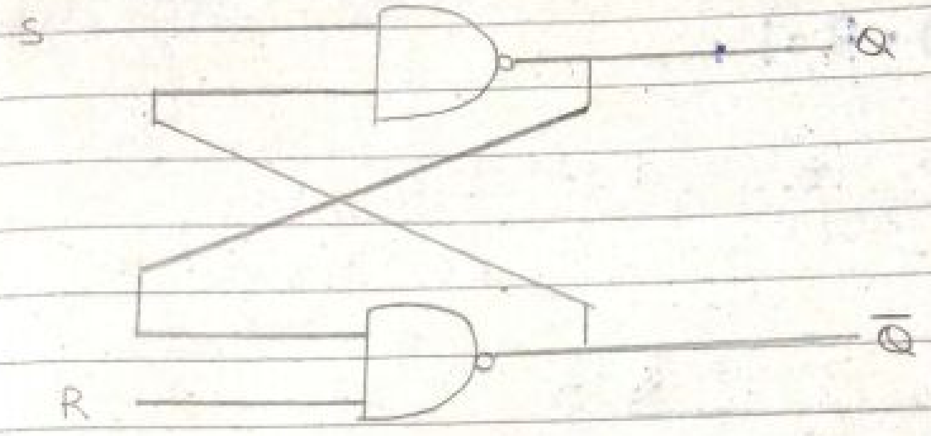
→ not complement of each other

⇒ not used case

⇒ Truth Table For Nor SR Latch:-

S	R	Q	$\overline{Q}$
0	0	Memory	Memory
0	1	0	1
1	0	1	0
1	1	Not used	Not used

=> NAND SR LATCH:-



When Set=1 Q=0
When Reset=1 Q=1

TRUTH TABLE OF NAND GATE:-

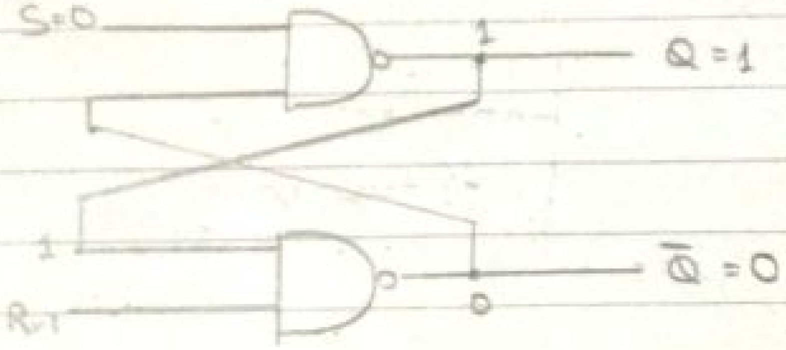
1 ←

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

* NAND SR LATCH is also called "active-low input SR latch."

=> CASE : 01

When S=0 , R=1 ,

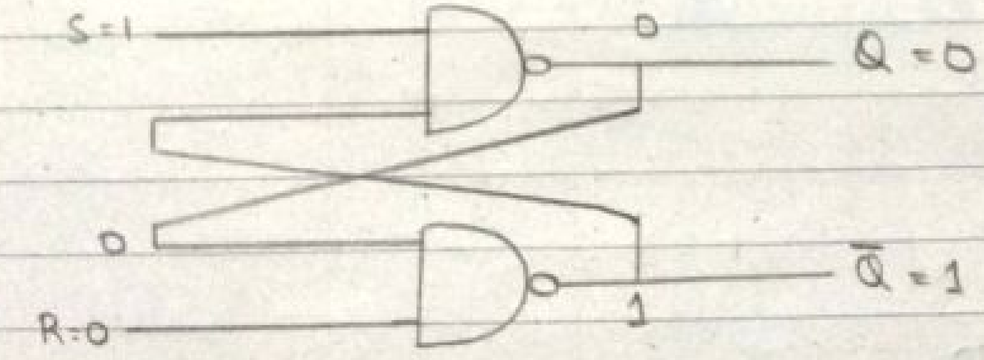


Q=1 , $\bar{Q}$=0

Memory = S=1 , R=1

CASE : 02

When S=1 , R=0

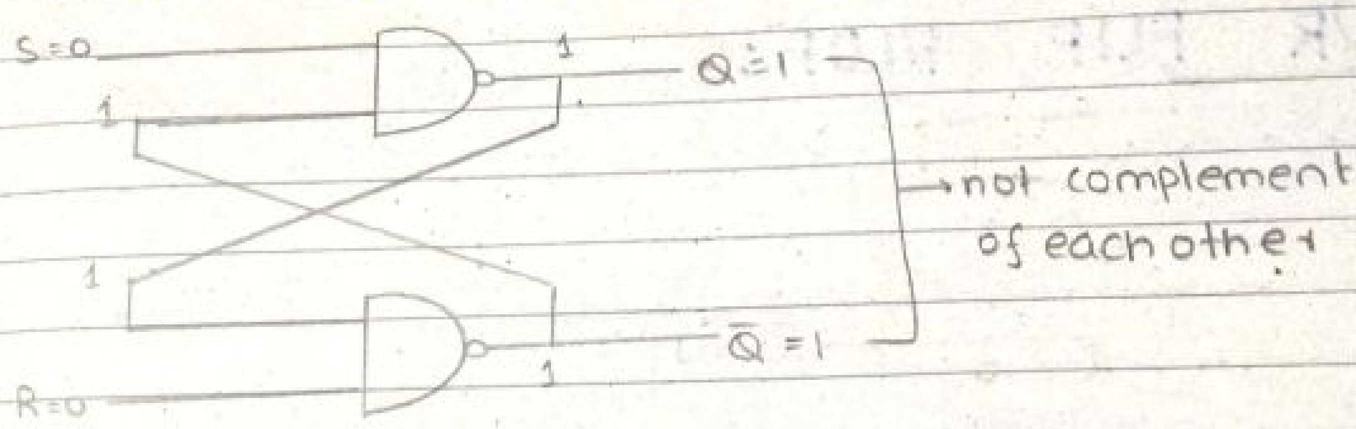


Q=0 , $\bar{Q}$=1

Memory S=1 , R=1

CASE:03

When S=0, R=0 ⇒ Not used



TRUTH TABLE FOR NAND

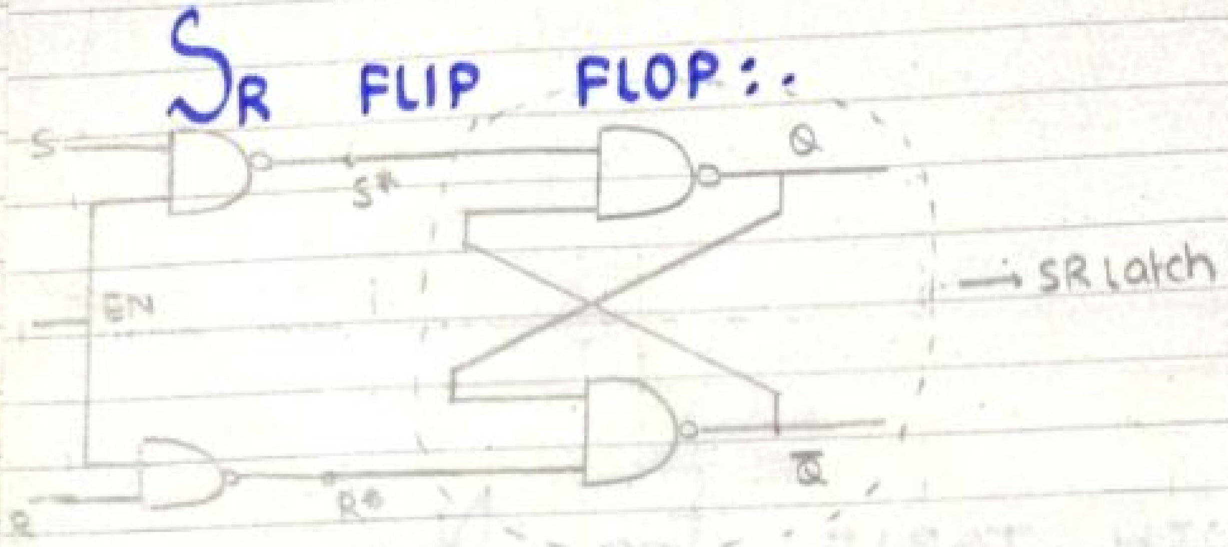
SR LATCH:-

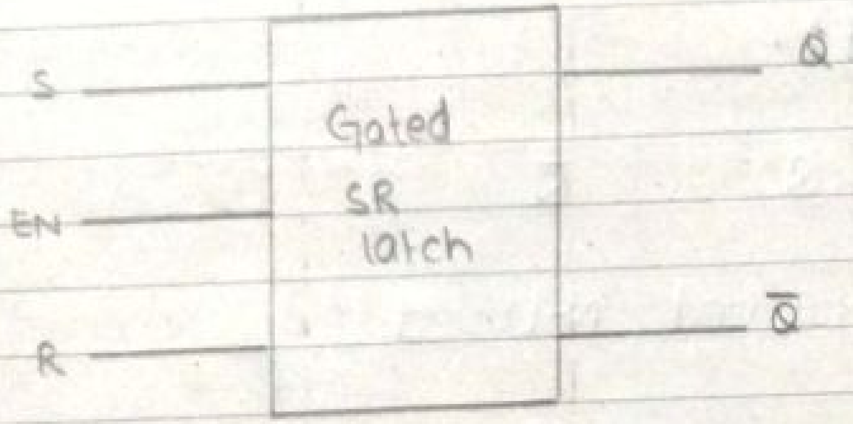
S	R	Q	$\overline{Q}$
0	0	Not used	Not used
0	1	1	0
1	0	0	1
1	1	Memory	Memory

Scanned with CamScanner

=> GATED SR LATCH /

SR FLIP FLOP :-





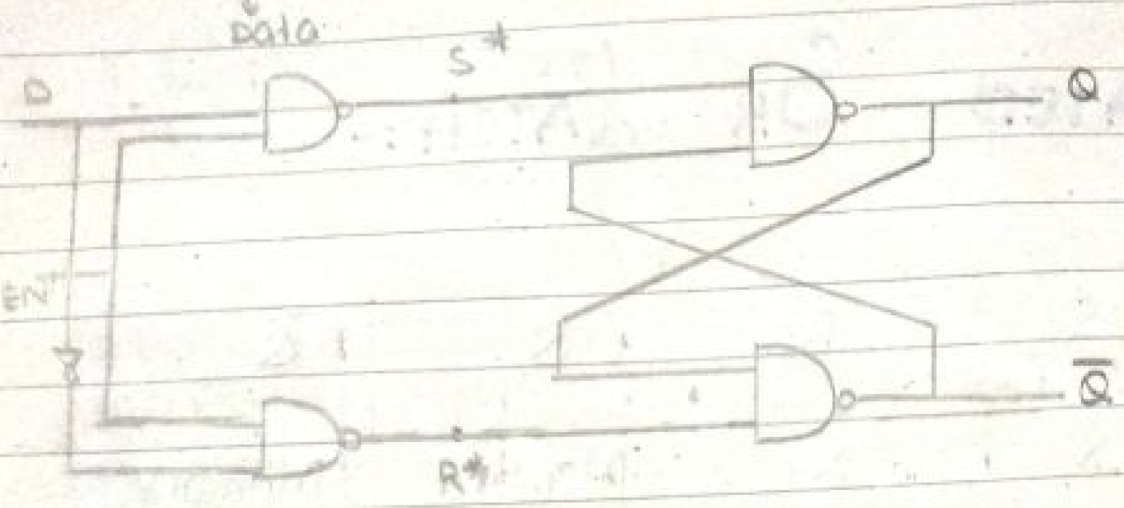
TRUTH TABLE FOR SR →NAND LATCH IS :-

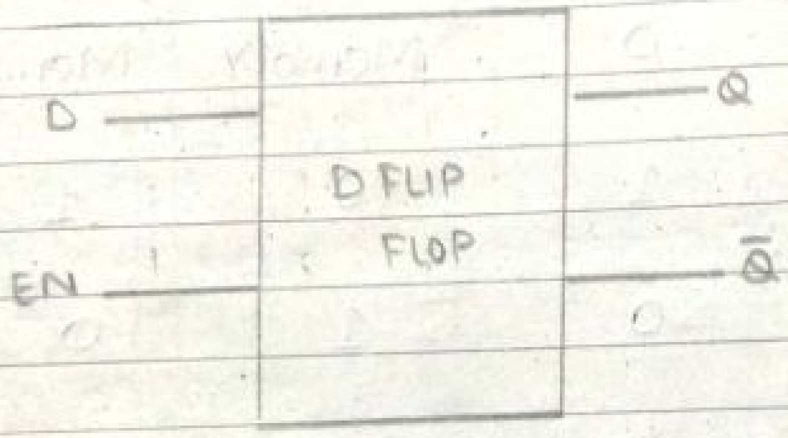
S	R	Q	$\bar{Q}$
0	0	Not used	Not used
0	1	1	0
1	0	0	1
1	1	Memory	Memory

TRUTH TABLE FOR GATED SR LATCH :-

Enable	S	R	Q	$\bar{Q}$
0	X	X	Memory	Memory
1	0	0	Memory	Memory
1	0	1	0	1
1	1	0	1	0
1	1	1	Not used	Not used.

D FLIP FLOR :-





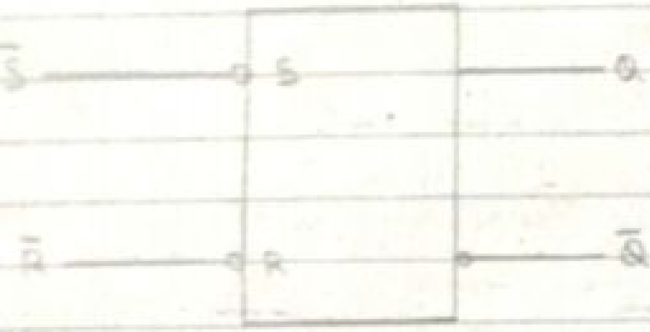
TRUTH TABLE FOR D FLIP FLOP:.

EN	D	Q	$\overline{Q}$
0	X	Memory	Memory
1	1	1	0
1	0	0	1

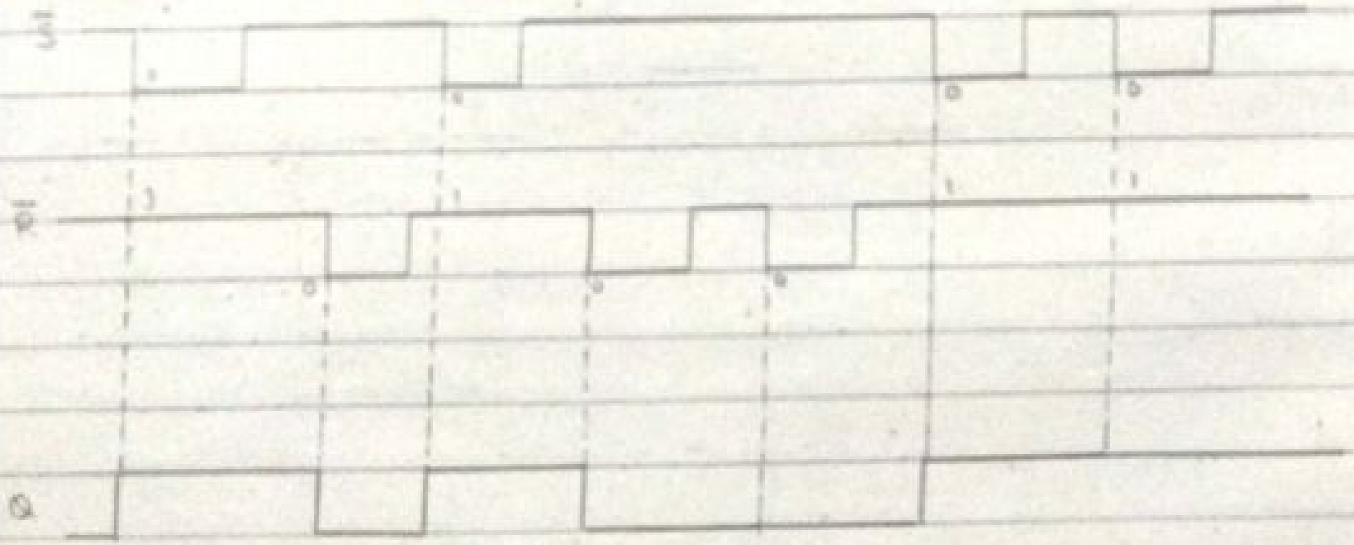
⇒ ACTIVE LOW INPUT SR LATCH

EXAMPLE : 7·1

If the $\overline{S}$ and $\overline{R}$ waveforms are applied to the inputs of the latch in active low input SR latch. Determine the waveform that will observed on the Q output. Assume that Q is initially low.



⇒ NAND SR LATCH



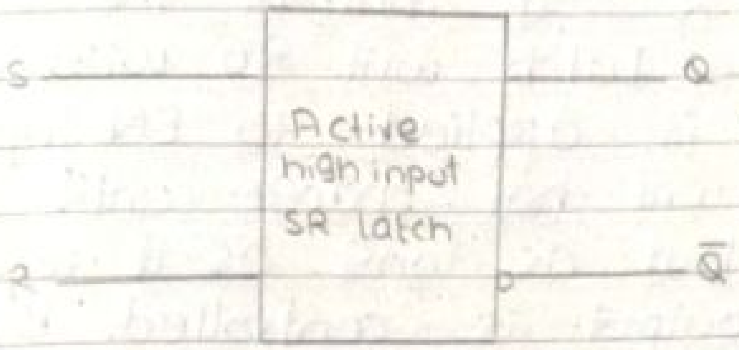
⇒ When $\overline{S}$ is low and $\overline{R}$ is high, it sets the latch

⇒ When $\overline{R}$ is low, it resets the latch

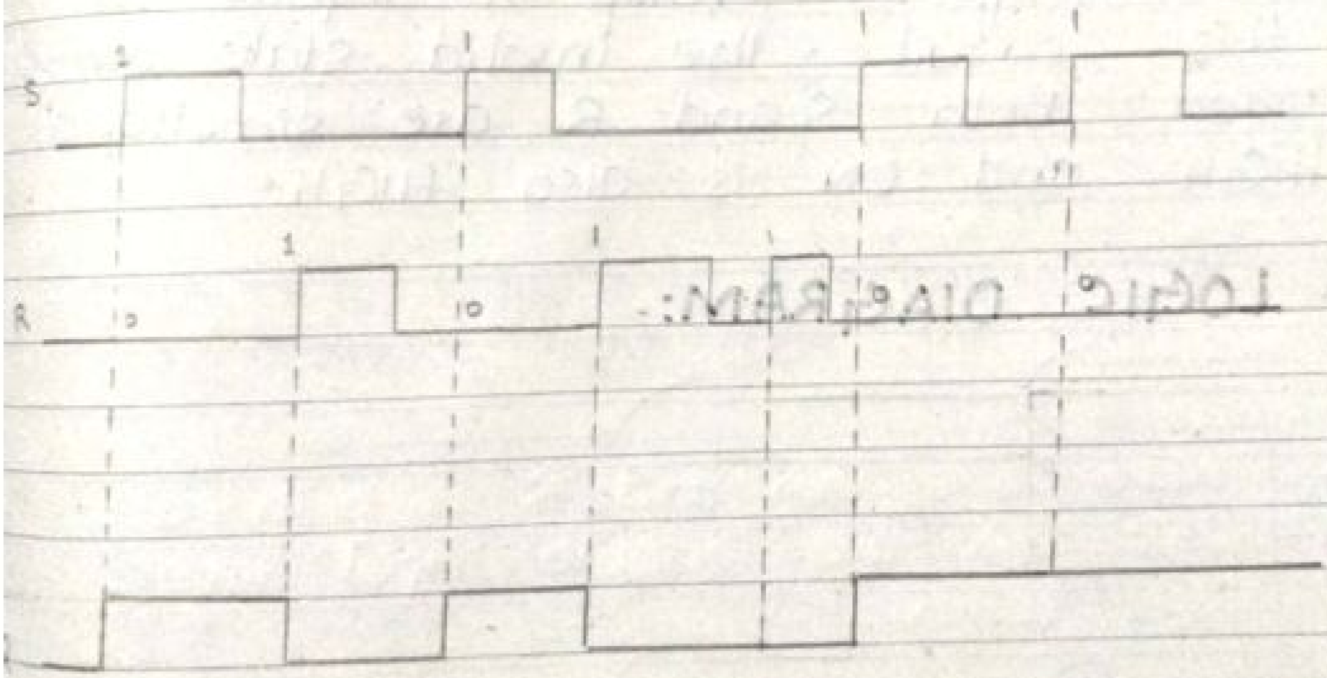
=> ACTIVE HIGH INPUT SR LATCH

RELATED PROBLEM:

Determine the Q output of an active high input S-R latch if the waveforms are inverted and applied to the inputs.



=> NOR - SR LATCH



=> When S is high and R is low, it sets the latch

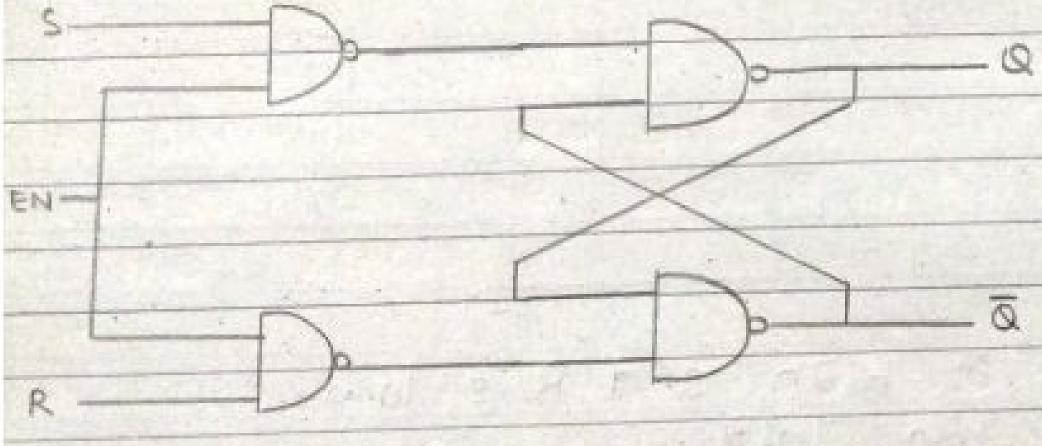
=> When R is high, it resets the latch.

SR-FLIP FLO

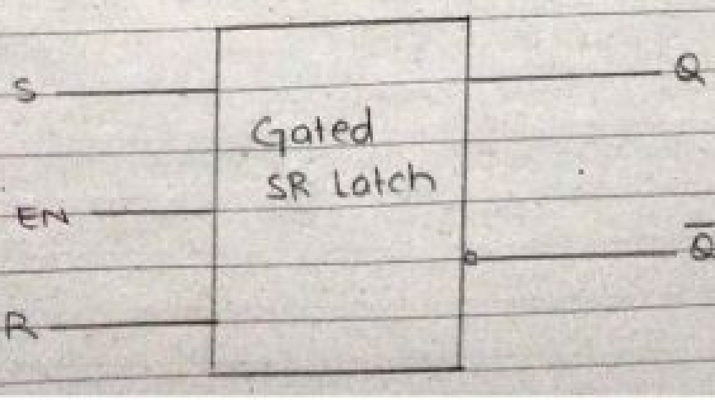
THE GATED S-R LATCH:-

A gated latch requires an enable input, EN. The S and R input controls the state to which the latch will go when a high level is applied to EN input. The latch will not change until EN is high; but as long as it remains high, the output is controlled by the state of the S and R inputs. In this circuit, the invalid state occurs when both S and R are simultaneously HIGH and EN is also HIGH.

LOGIC DIAGRAM:-

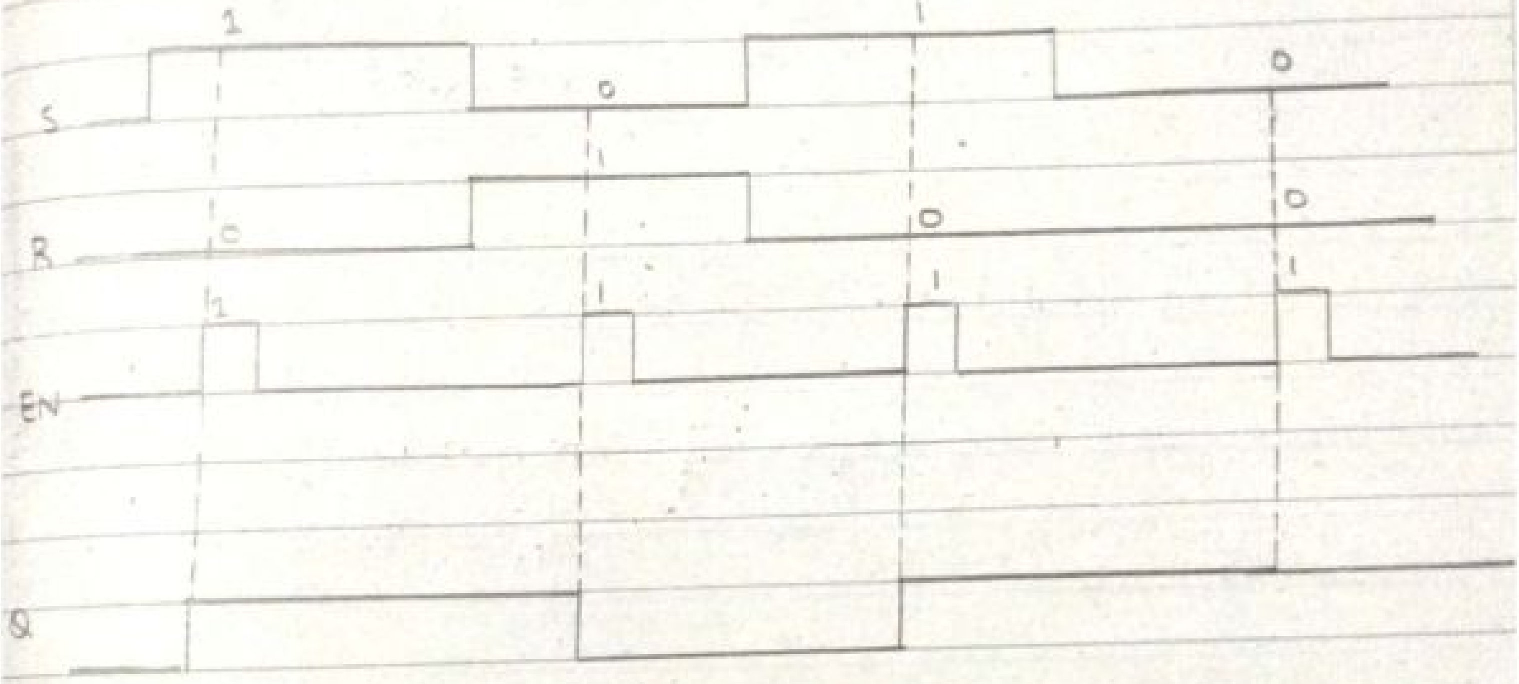


LOGIC SYMBOL



EXAMPLE 7·2

Determine the Q output waveforms if the inputs are applied to gated SR latch that is initially RESET·



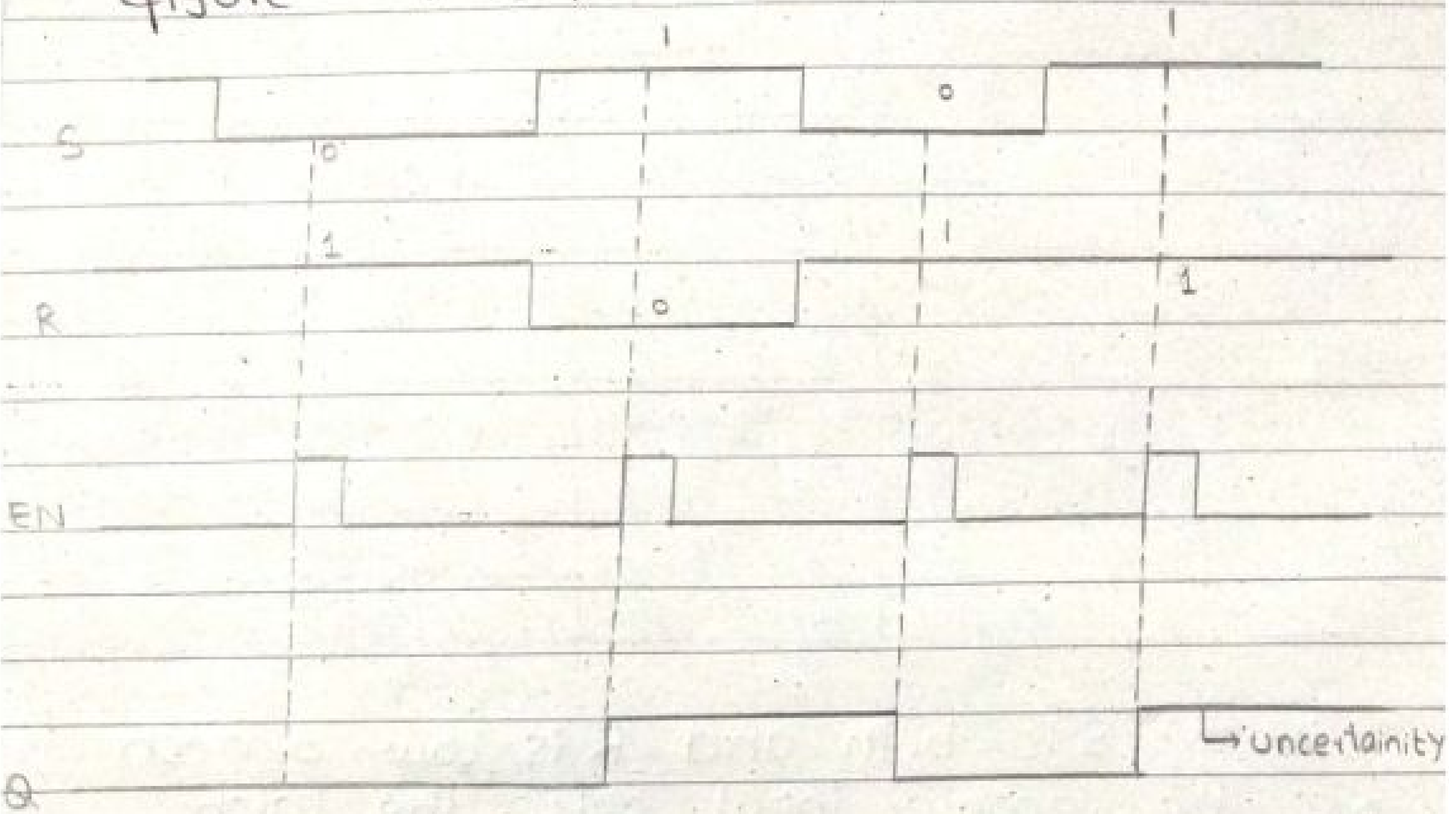
⇒ When S is high and R is low, a high on the enable input sets the latch

⇒ When S is low and R is high, a high on the enable input resets the latch·

⇒ When S is low and R is low, the Q output does not change from its Present state·

RELATED PROBLEM

Q. Determine the Q output of a gated S-R latch if the S and R inputs in figure 7-9(a) are inverted.



=> When S is low and R is high, a high' on the enable input resets the Latch.

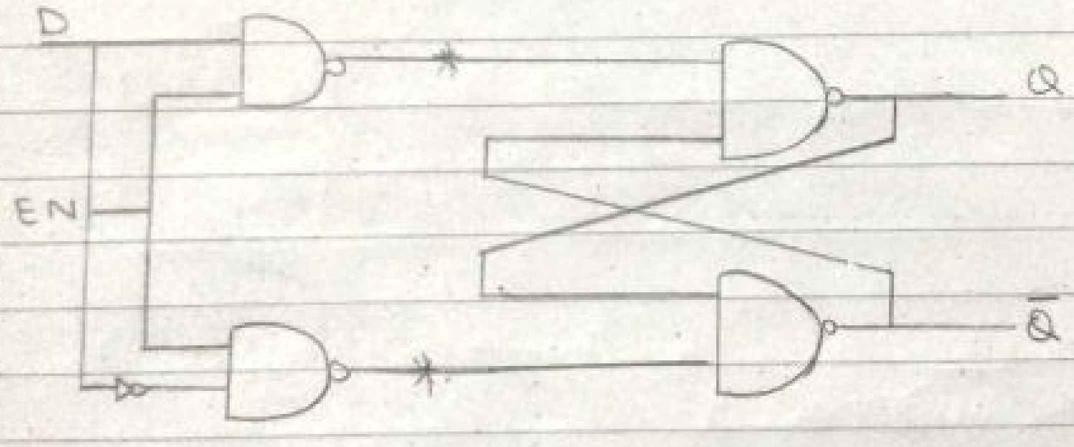
=> When S is high and R is low, a high on the enable input sets the latch

=> When S is high and R is also high, a high on the enable input gives uncertain output.

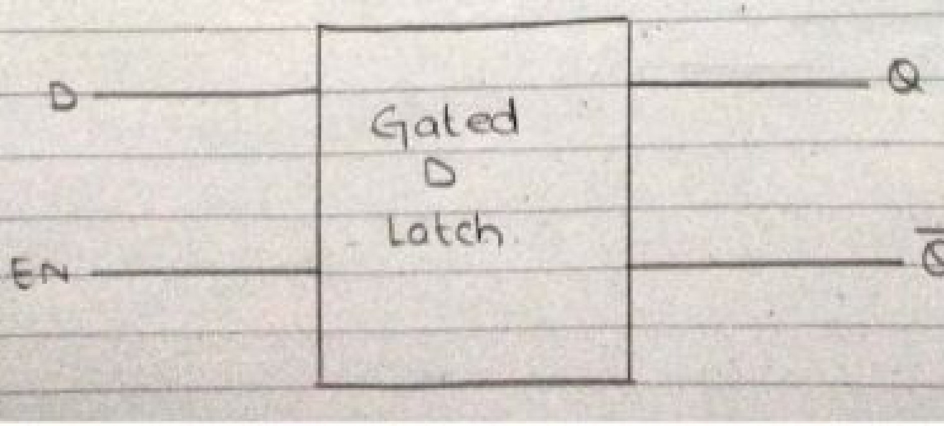
⇒ THE GATED D LATCH:-

⇒ Another type of gated latch is called the D Latch. It differs from the S-R Latch because it has only one input in addition to EN. This input is also called D (Data) input. When the D input is high and EN is high, the latch will set. When the D input is low and EN is high, the latch will reset. stated another way, the output Q follows the input D when EN is high.

⇒ LOGIC DIAGRAM:

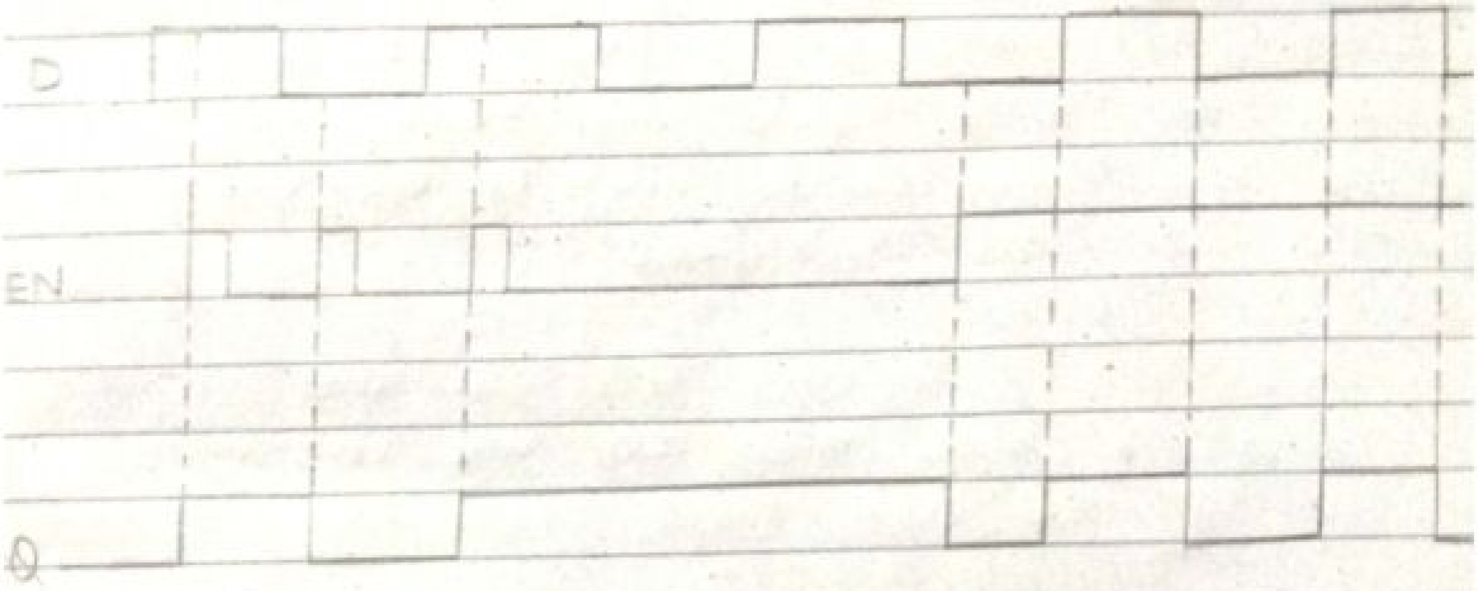


LOGIC SYMBOL:-



EXAMPLE: 7·3

Determine the Q output waveform if the inputs are applied to a gated D latch, which is initially reset.

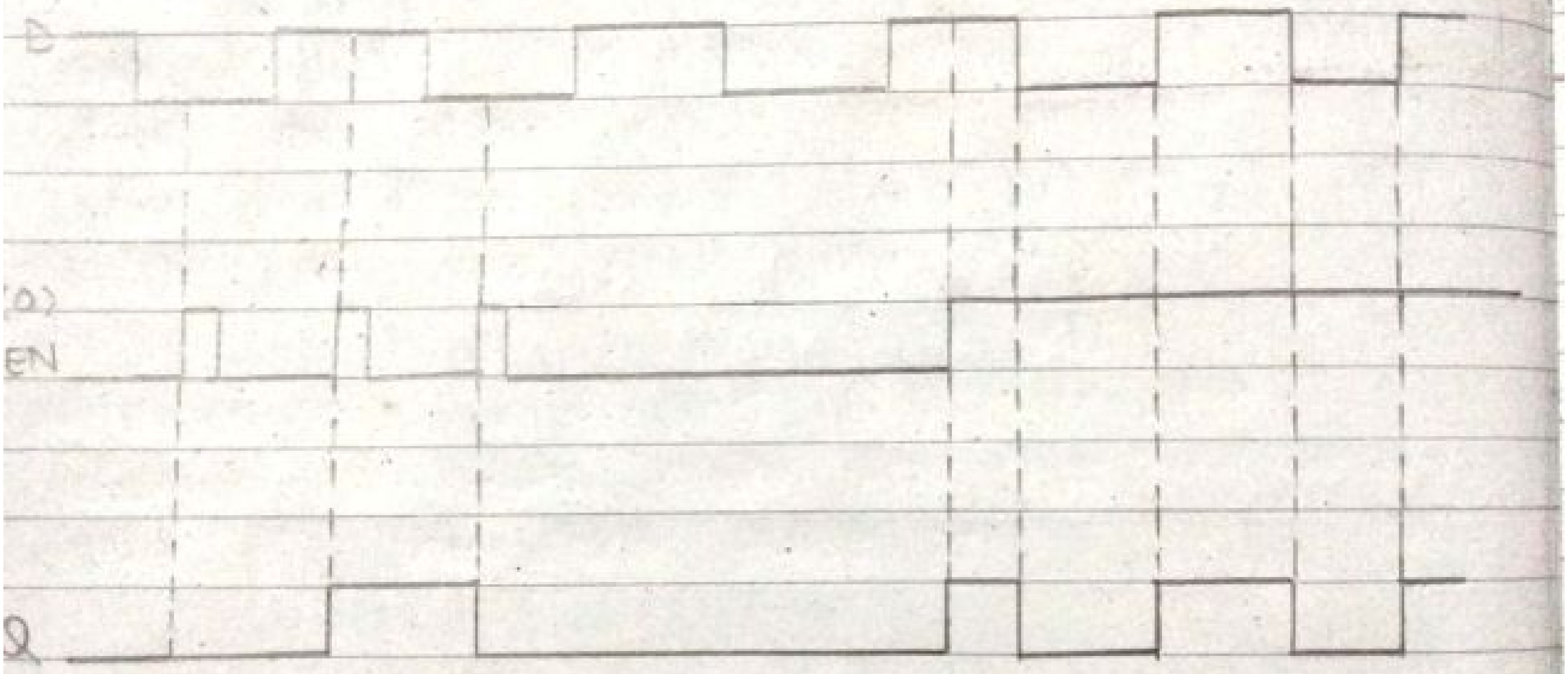


=> When D is high and Enable is high, the Q goes high

=> When D is low and Enable is high, the Q goes low.

RELATED PROBLEM

Q. Determine the Q output waveform, if the inputs are applied to a gated D Latch, which is initially RESET.



⇒ When D is low and Enable is high, the output 'Q' goes low.

⇒ When D is high and Enable is high, the output 'Q' goes high.

=> Edge Triggered Flip Flops:-

Flip-flops are synchronous bistable devices, also known as "bistable multivibrators". In this case, the term synchronous means that the output changes state only at a specified point on the triggering input called the clock (CLK), which is designated as a control input, C; that is, changes in the output occur in synchronization with the clock.

=> An edge triggered flipflop changes state either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse and is sensitive to its inputs only at this transition of the clock.

=> Three types of flip flops are:

1- SR Flip-flop

2- D Flip-flop

3- J-K Flip-flop.

=> The Edge Triggered S-R Flip Flop:-

The S and R inputs of the S-R flip flop are called synchronous inputs because data on these inputs are transferred to the flip flops output only on the triggering edge of the clock pulse

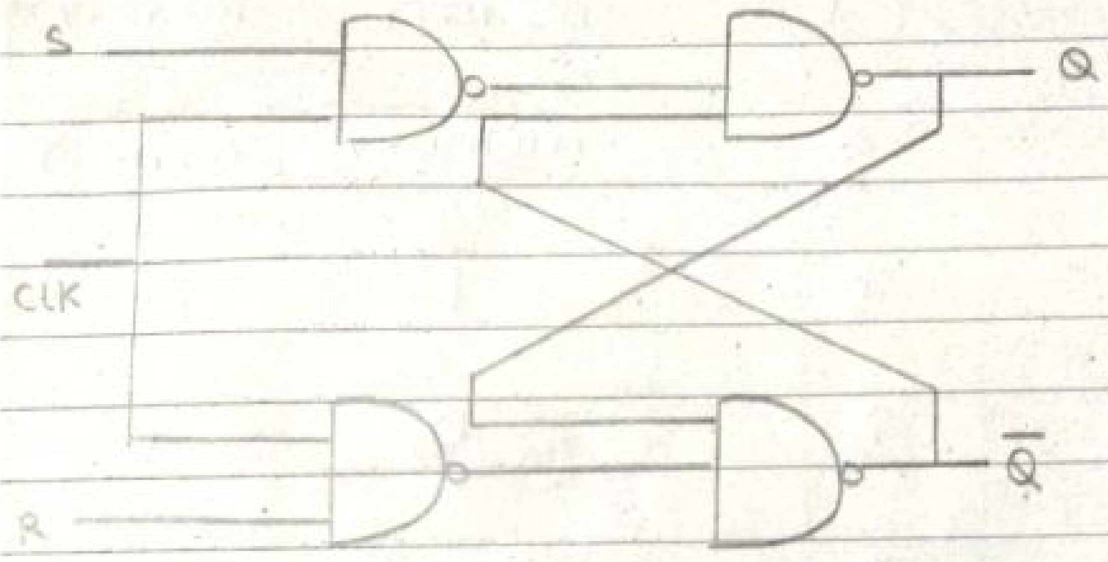
=> When S is high and R is low, the output Q goes high on the triggering edge of the clock pulse. → flip flop is set

=> When S is low and R is high, the output Q goes low on the triggering edge of the clock pulse → flip flop is reset

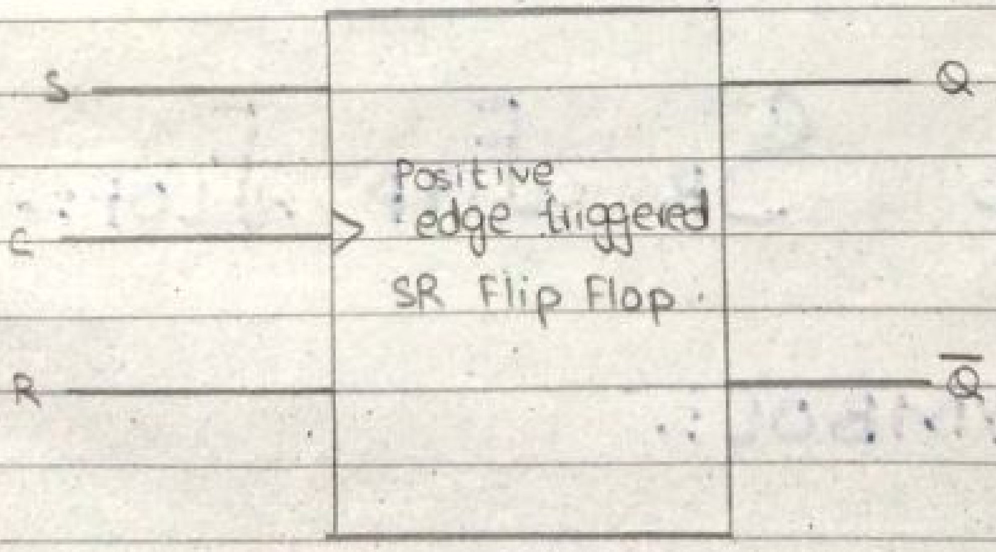
=> When both S and R are low, the output Q does not change from its prior state. ↳ memory

=> When both S and R are high, an invalid condition occurs on triggering edge of the clock pulse ↳ Invalid condition.

Logic Diagram:-



Logic Symbol:-



Scanned with CamScanner

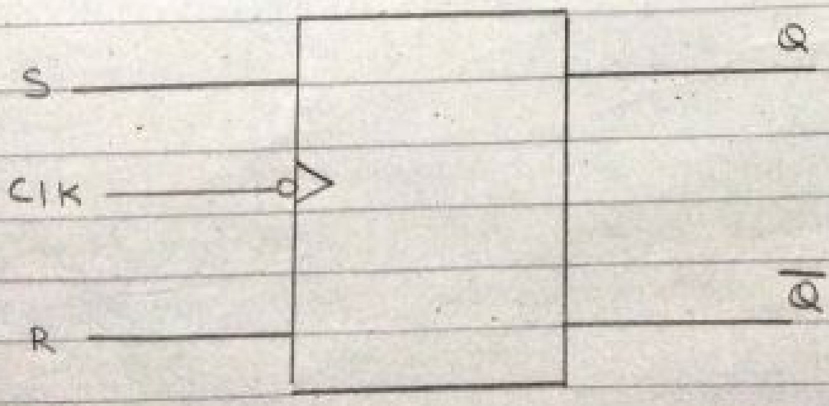
Scanned with CamScanner

TRUTH TABLE:- → positive edge triggered SR-Flip flop

S	R	CLK	Q	$\overline{Q}$
X=0	X=0	↓	memory → Q_0	memory → $\overline{Q_0}$
0	0	↑	memory → Q_0	memory → $\overline{Q_0}$
1	0	↑	1	0
0	1	↑	0	1
1	1	↑	IC	IC

=> NEGATIVE EDGE TRIGGERED SR FLIP FLOP:-

LOGIC SYMBOL:-



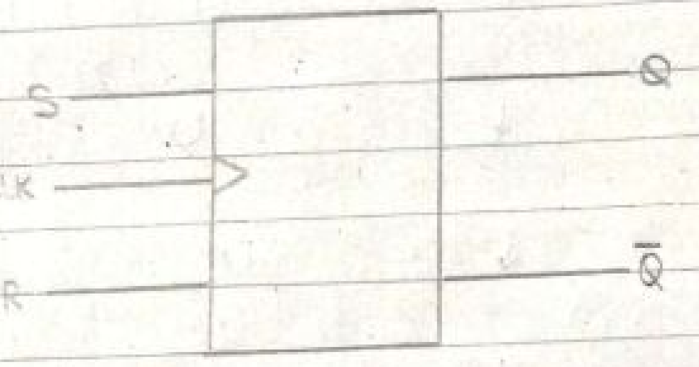
TRUTH TABLE:-

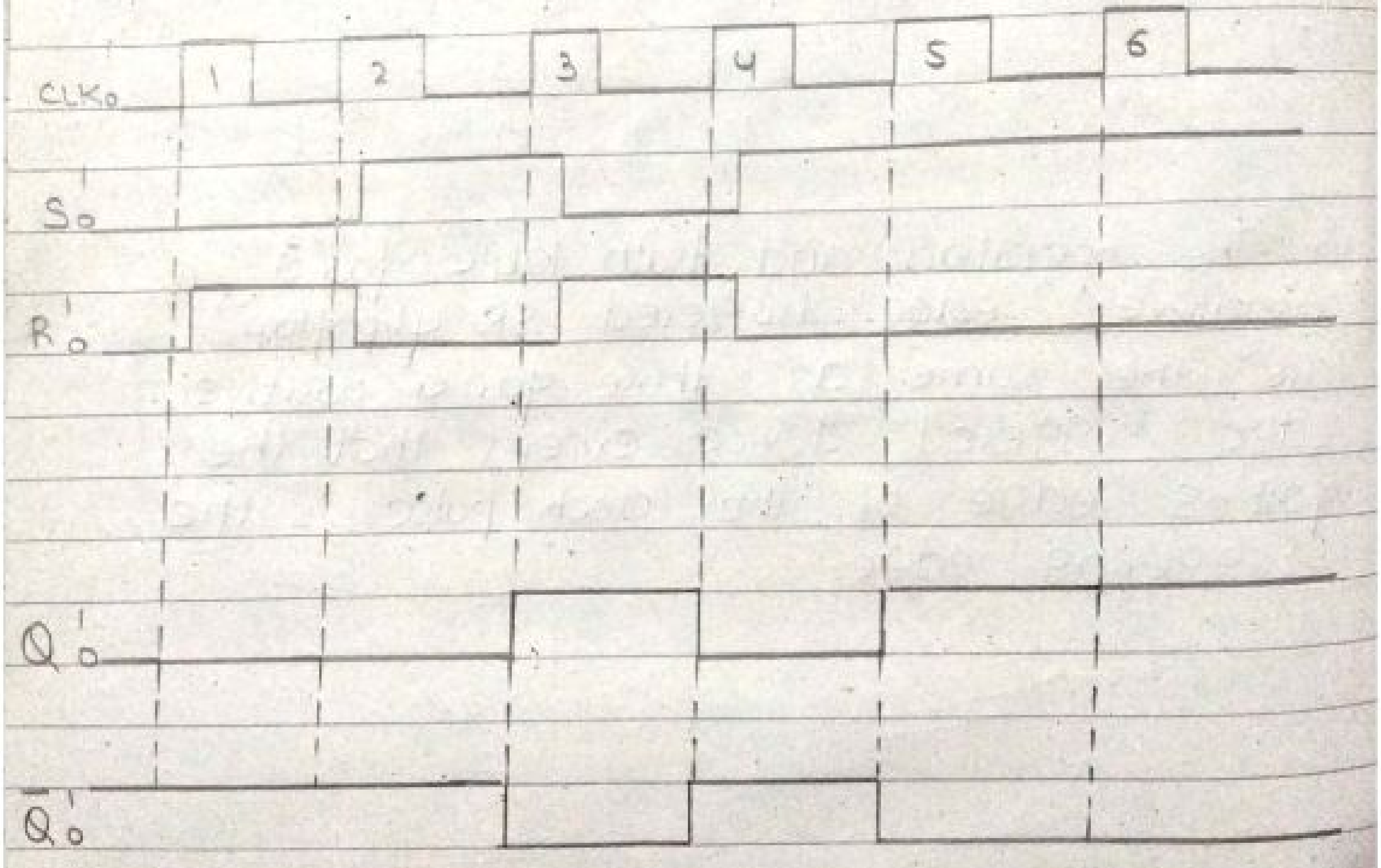
S	R	CLK	Q	$\bar{Q}$
0	0	↓	memory	memory
1	0	↓	1	0
0	1	↓	0	1
1	1	↓	IC	IC

=> The operation and truth table for a negative edge triggered SR flip flop are the same as those for a positive edge triggered device except that the falling edge of the clock pulse is the triggering edge.

7.4

Deteimine the Q and $\overline{Q}$ output wavefoims of the flip-flop for the S, R and CLK inputs. Assume that the positive edge-triggered flip flop is initially reset.





⇒ At clock pulse 1, S is Low and R is low, so Q does not change.

=> At clock pulse 2, S is Low and R is high, so Q remains Low (RESET)

=> At clock pulse 3, S is HIGH and R is Low, so Q goes HIGH. (SET)

=> At clock pulse 4, S is LOW and R is HIGH, so Q goes Low. (RESET)

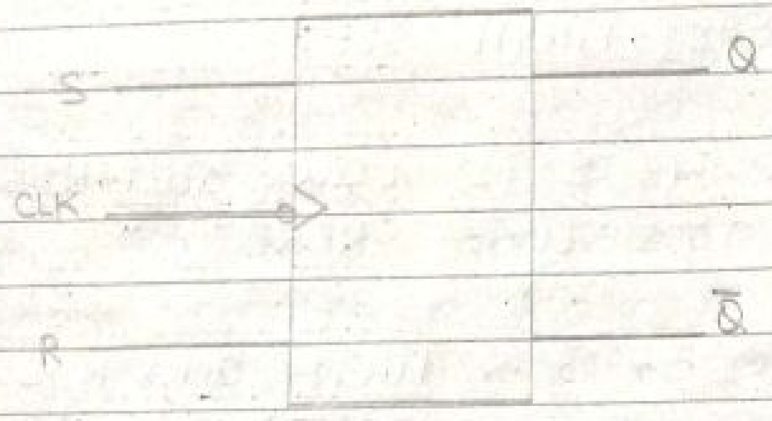
=> At clock pulse 5, S is HIGH and R is LOW, so Q goes HIGH. (SET)

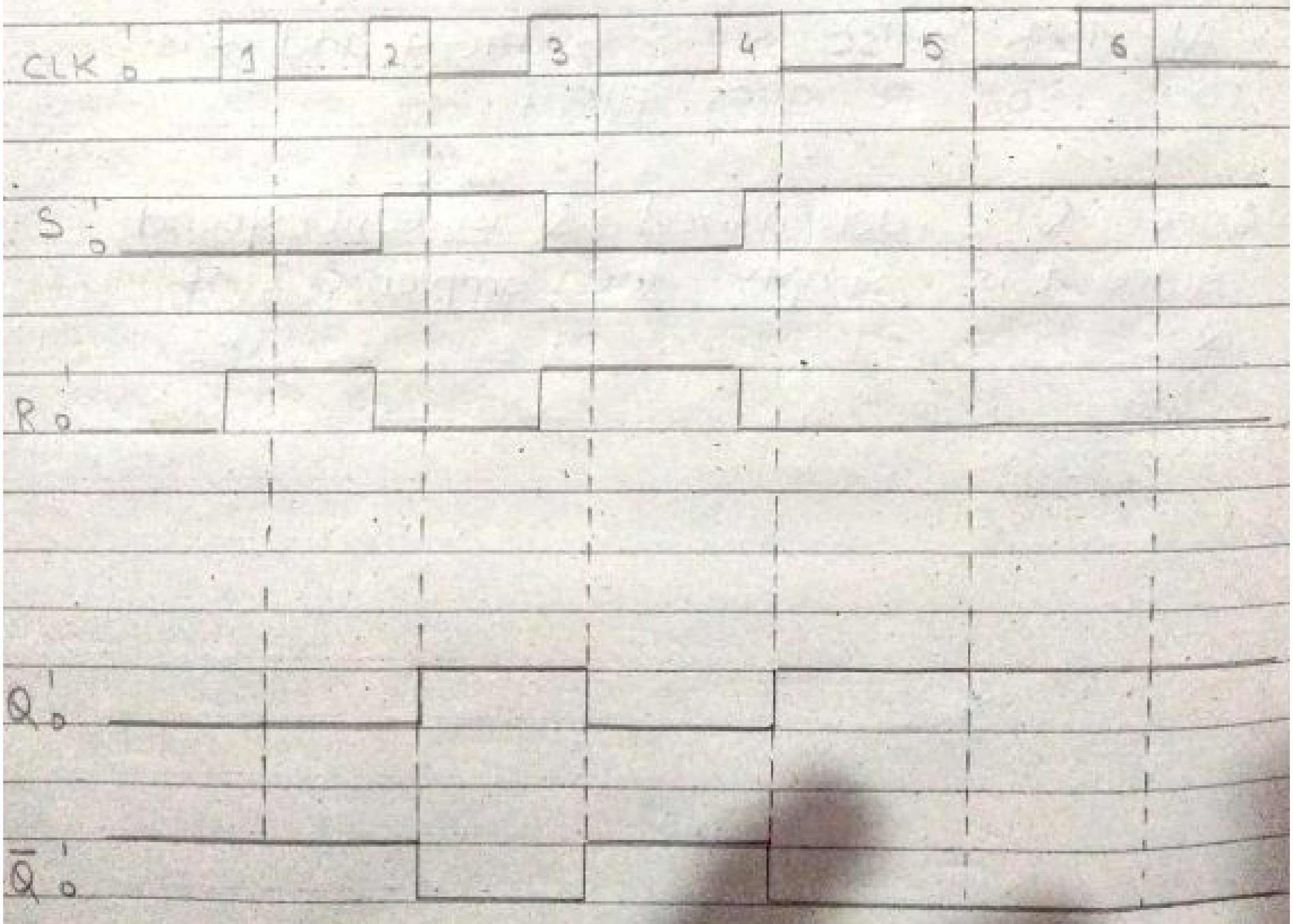
=> At clock pulse 6, S is HIGH and R is Low, so Q goes HIGH

=> Once Q is determined, $\overline{Q}$ is easily found since it is simply the complement of Q.

RELATED PROBLEM:-

Determine Q and $\overline{Q}$ for the S and R inputs if the flip flop is a negative edge triggered device.





⇒ At clock pulse 1, S is Low and R is HIGH, So Q remains Low. (Reset)

⇒ At clock pulse 2, S is HIGH and R is Low, so Q goes HIGH. (set)

⇒ At clock pulse 3, S is low and R is HIGH, so Q goes Low. (Reset)

⇒ At clock pulse 4, S is HIGH and R is low, So Q goes HIGH. (set)

⇒ At clock pulse 5, S is HIGH and R is low, So Q remains HIGH. (set)

⇒ At clock pulse 6, S is HIGH and R is low, so Q remains HIGH. (set)

⇒ Once Q is determined, $\overline{Q}$ is easily found since it is simply the complement of Q.

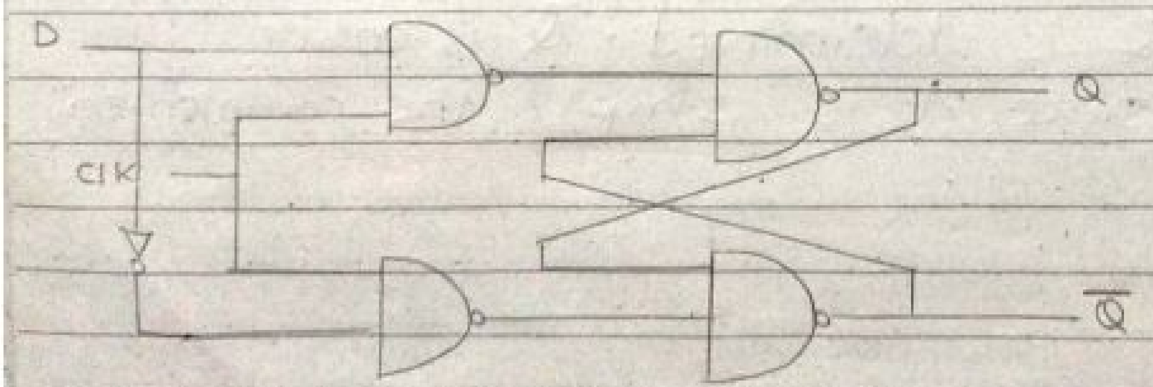
The Edge Triggered D Flip Flop:-

=> The D flip-flop is useful' When a single data bit (1 or 0) is to be stored. The addition of an inverter to an S-R flip flop creates a basic D flipflop The D (data) flip flop has only one input, in addition to the clock.

=> When 'D' is HIGH, the output 'Q' goes HIGH on the triggering edge of the clock pulse. → flip flop is Set.

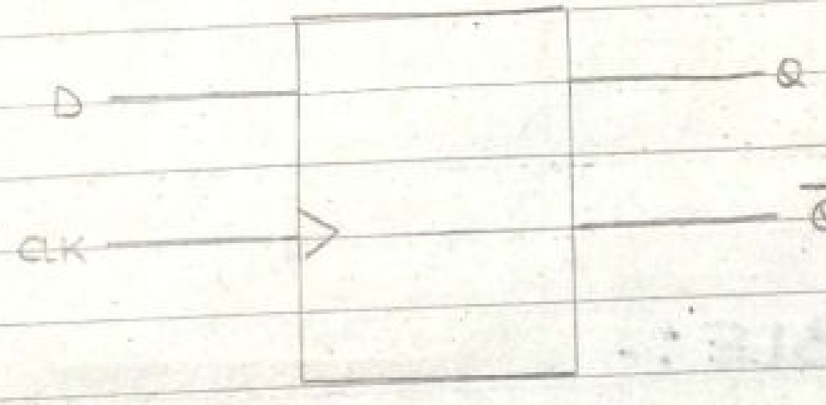
=> When 'D' is Low, the output 'Q' goes Low on the triggering edge of the clock pulse. → flip flop is Reset

Logic Diagram:-



POSITIVE EDGE TRIGGERED D - FLIP FLOP :-

=> LOGIC SYMBOL:-



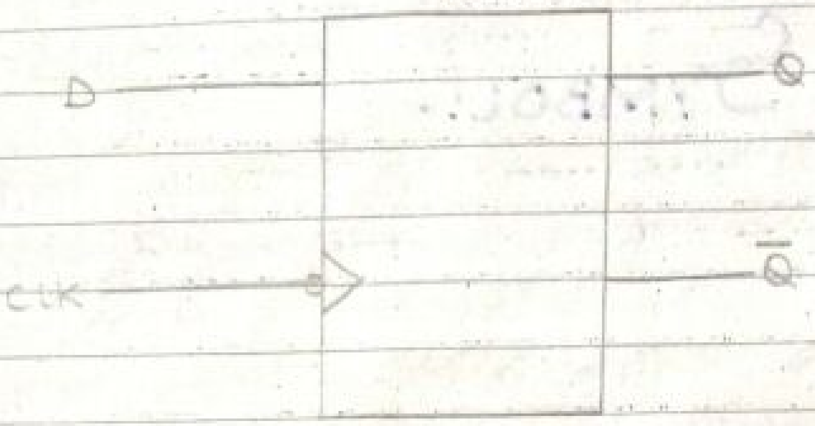
=> TRUTH TABLE:-

CLK	D	Q	$\overline{Q}$
↑	0	0	1
↑	1	1	0

↑ = clock transition Low to HIGH.

=> NEGATIVE EDGE TRIGGERED D FLIP FLOP:-

LOGIC SYMBOL:

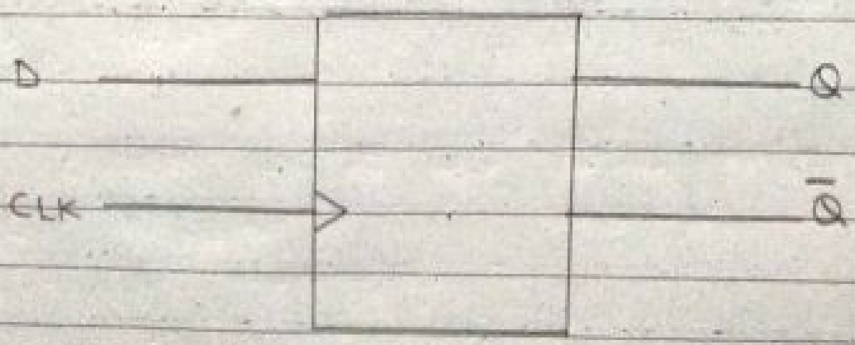


TRUTH TABLE:-

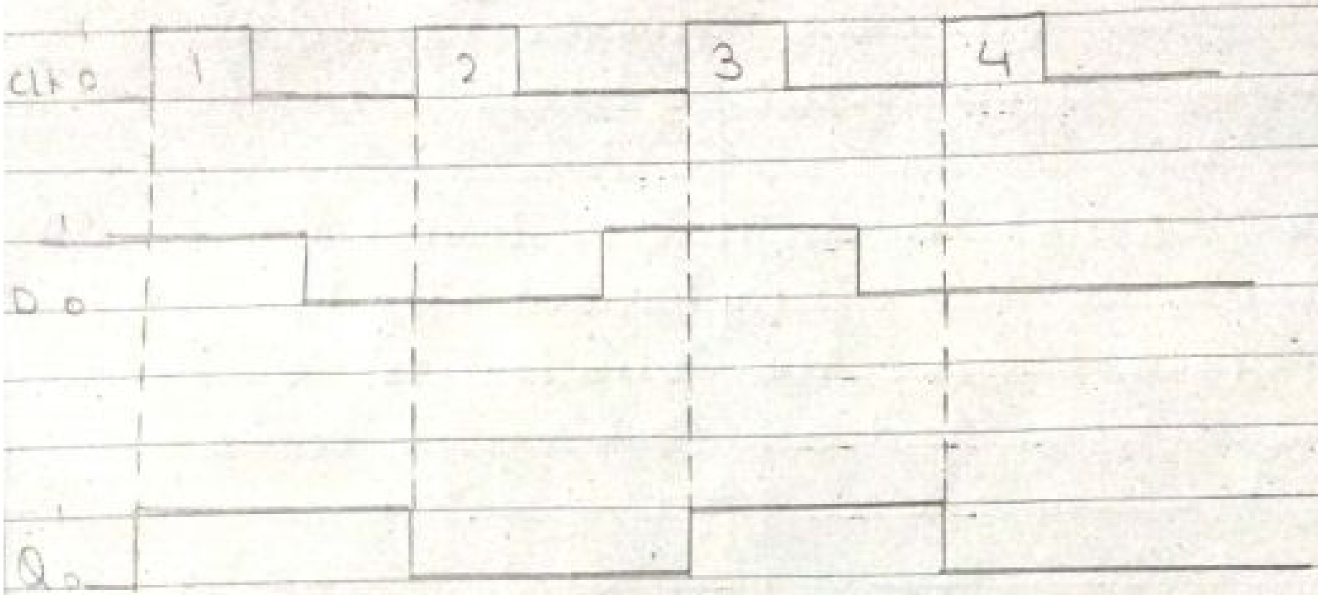
CLK	D	Q	$\overline{Q}$
↓	0	0	1
↓	1	1	0

EXAMPLE: 7·5

Given the waveforms (a) for the D input and the clock, determine the Q output waveform if the flip flop starts out RESET.



Sol:-



=> At clock pulse 1, D is HIGH, so Q goes HIGH. (Set)

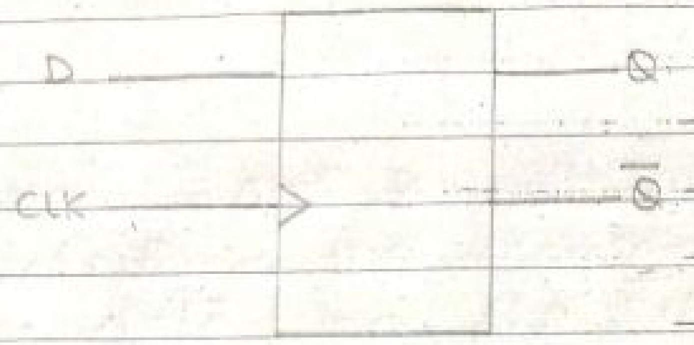
=> At clock pulse 2, D is Low, So Q goes Low (Reset)

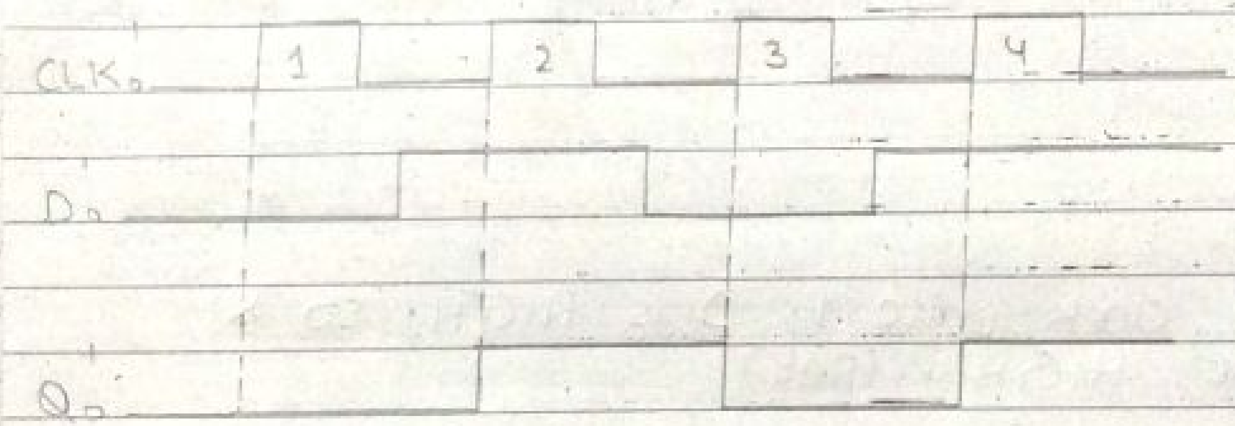
=> At clock pulse 3, D is HIGH, So Q goes HIGH (set)

=> At clock pulse 4, D is Low, so Q goes Low. (Reset)

RELATED PROBLEM:-

Q. Determine the Q output for the D flip-flop if the D input is inverted.





=> At clock pulse 1, D is LOW, so Q remains low. (RESET)

=> At clock pulse 2, D is HIGH, so Q goes HIGH (SET)

=> At clock pulse 3, D is LOW, so Q goes low. (RESET)

=> At clock pulse 4, D is HIGH, so Q goes HIGH. (SET)

The Edge Triggered

J-K flip Flop:-

The J-K flip flop is a versatile and is a widely used type of flipflop. The functioning of the J-K flipflop is identical to that of the S-R flip flop in the SET, RESET and no change conditions of operation. The difference is that J-K flip flop has no invalid state as does the SR flipflop. It differs from the S-R edge triggered flip flop in that the Q output is connected back to the input of gate G_2 and the $\bar{Q}$ output is connected back to the input of gate G_1.

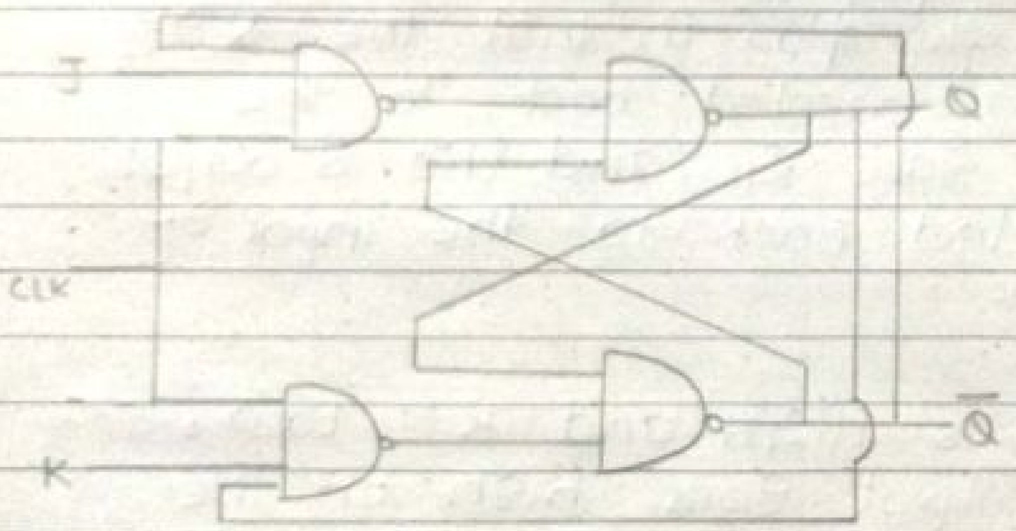
=> When J is HIGH and K is low, the Q output goes high on the triggering edge of the clock pulse.
↳ flip-flop is SET

=> When J is low and K is HIGH, the Q output goes low on the triggering edge of the clock pulse
↳ flip-flop is RESET

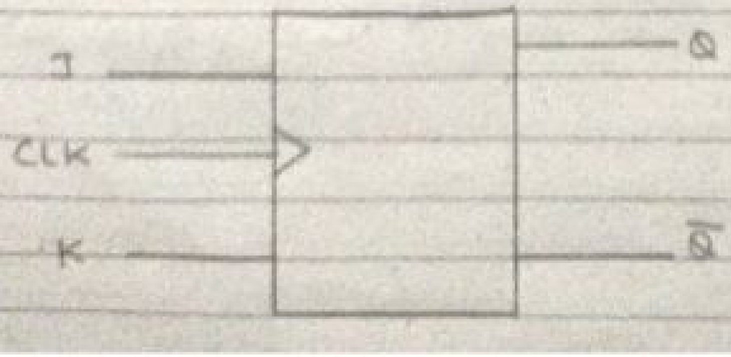
=> When J and K both are low, the Q output does not change from its prior state on the triggering edge of the clock pulse. ↳ Memory

=> When J and K both are high, the Q output is the complement of its prior state on the triggering edge of the clock pulse. ↳ Toggle mode.

=> LOGIC DIAGRAM:-



=> POSITIVE EDGE TRIGGERED J-K FLIP FLOP:-



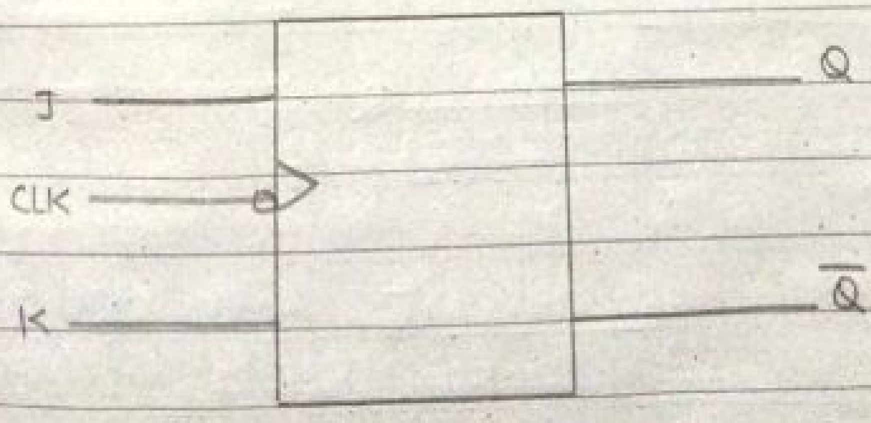
TRUTH TABLE:-

CLK	J	K	Q	$\overline{Q}$
↑	1	0	1	0
↑	0	1	0	1
↑	0	0	Memory	Memory
↑	1	1	$\overline{Q_n}$	Q_n

↑ = clock transition Low to HIGH

Qn: output level prior to clock transition.

⇒ NEGATIVE EDGE TRIGGERED J-K FLIP FLOP:-

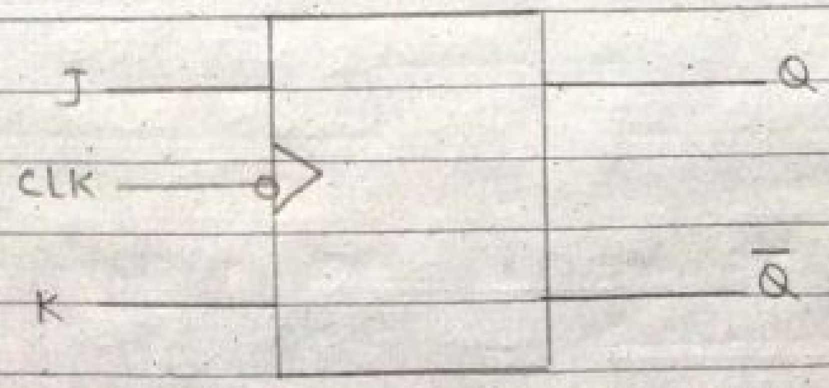


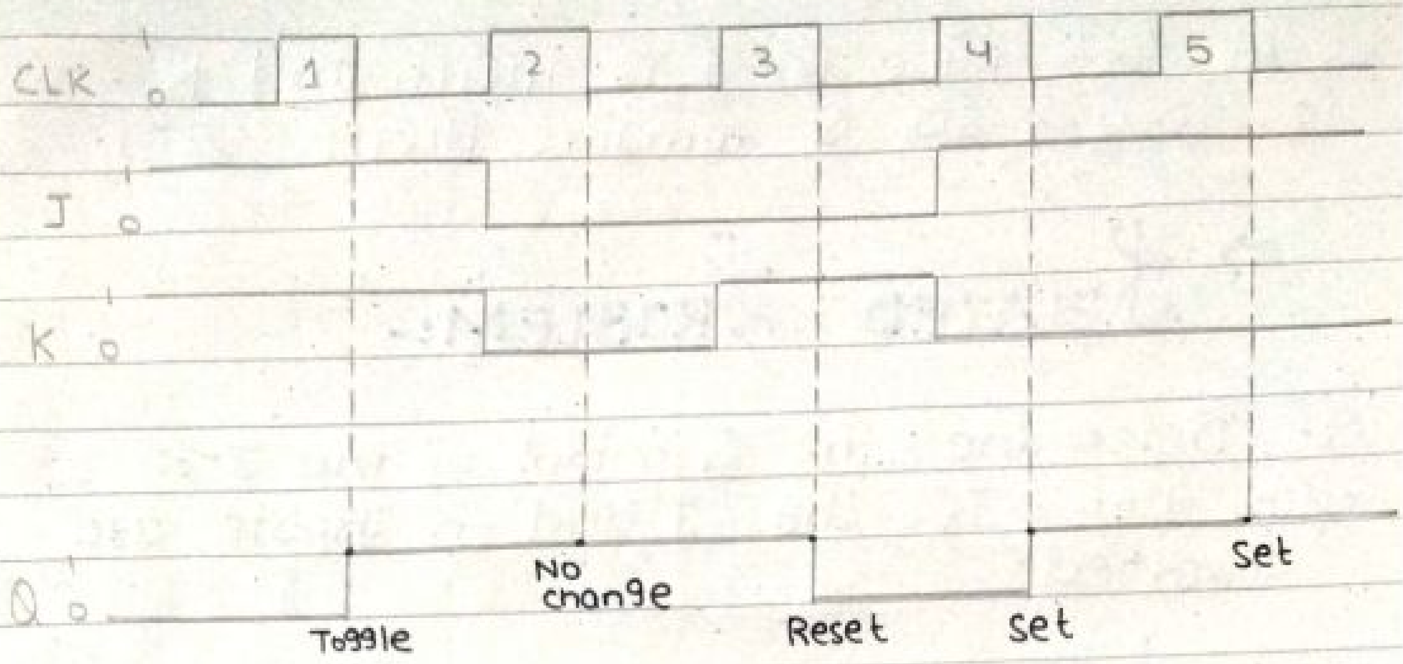
TRUTH TABLE:-

CLK	J	K	Q	$\overline{Q}$
↓	1	0	1	0
↓	0	1	0	1
↓	0	0	Memory	Memory
↓	1	1	$\overline{Qn}$	Qn

=> EXAMPLE 7-6

The waveforms are applied to the J, K and clock inputs as indicated. Determine the Q output, assuming that the flip flop is initially RESET.





=> Since, this is a negative edge triggered flip flop, as indicated by the "bubble" at the clock input, the Q output will change only on the negative going edge of the clock pulse.

=> At clock pulse 1, J and K both are high, and because it is a toggle condition Q goes HIGH. (SET).

=> At clock pulse 2, J and K both are low, so Q remains HIGH (SET)

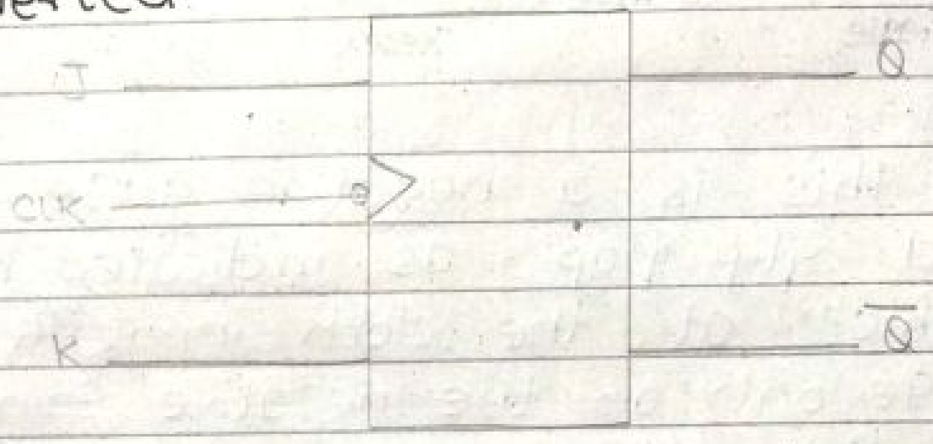
=> At clock pulse 3, J is Low and K is HIGH, so Q goes low. (RESET)

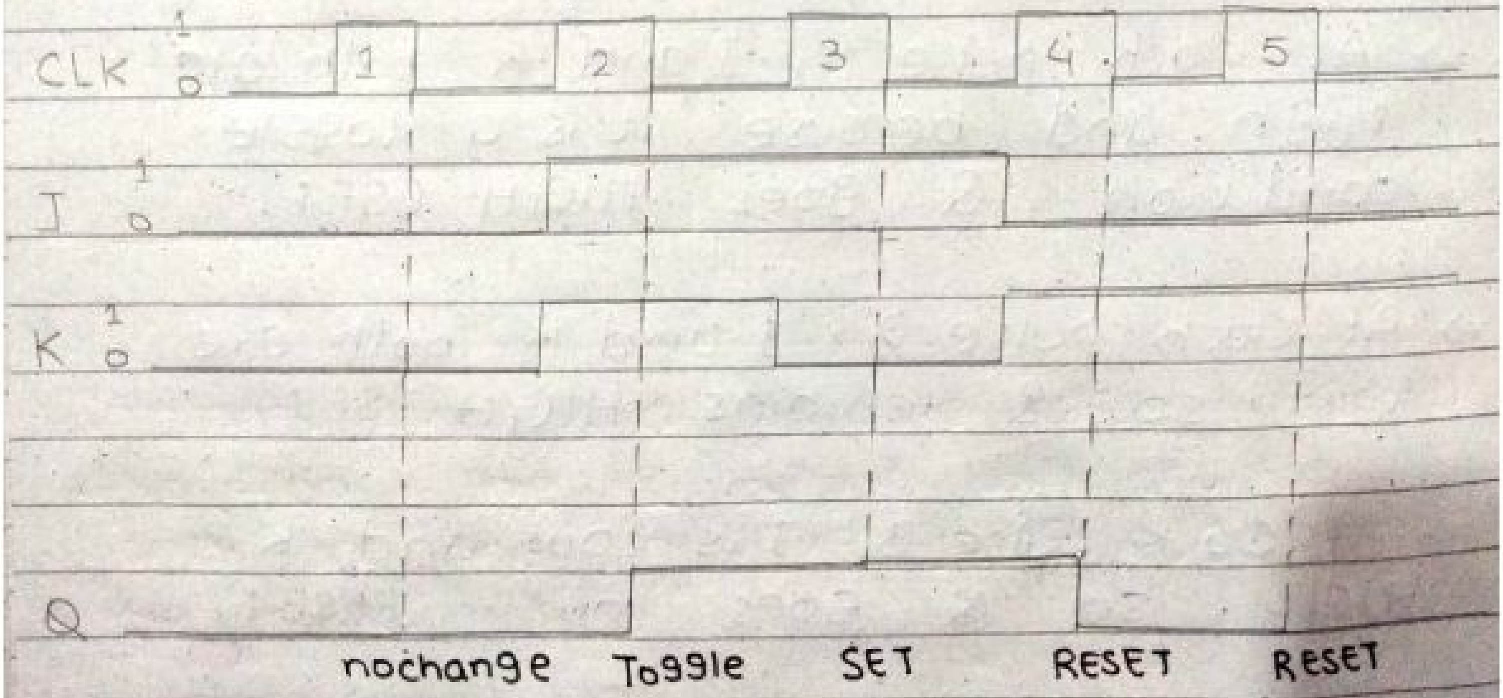
=> At clock pulse 4, J is HIGH and K is Low, so Q goes HIGH. (SET)

=> At clock pulse 5, J is HIGH and K is low, so Q remains HIGH. (SET).

=> RELATED PROBLEM:-

Q. Determine the Q output of the J-K flip flop. If the J and K inputs are inverted.





=> since, this is a negative edge triggered flip flop, as indicated by the "bubble" at the clock input, the Q output will change only on the negative going edge of the clock pulse.

=> At clock pulse 1, J and K both are low, so Q does not change to its prior/initial state and remains RESET

=> At clock pulse 2, J and K both are HIGH, so because it is a toggle condition Q goes HIGH.
↳ SET

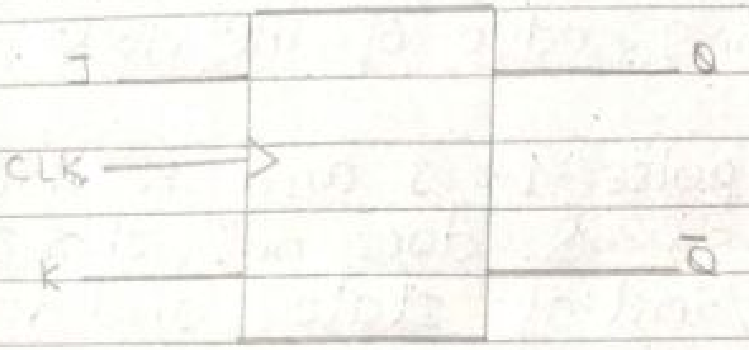
=> At clock pulse 3, J is HIGH and K is LOW, so Q remains HIGH.
↳ SET

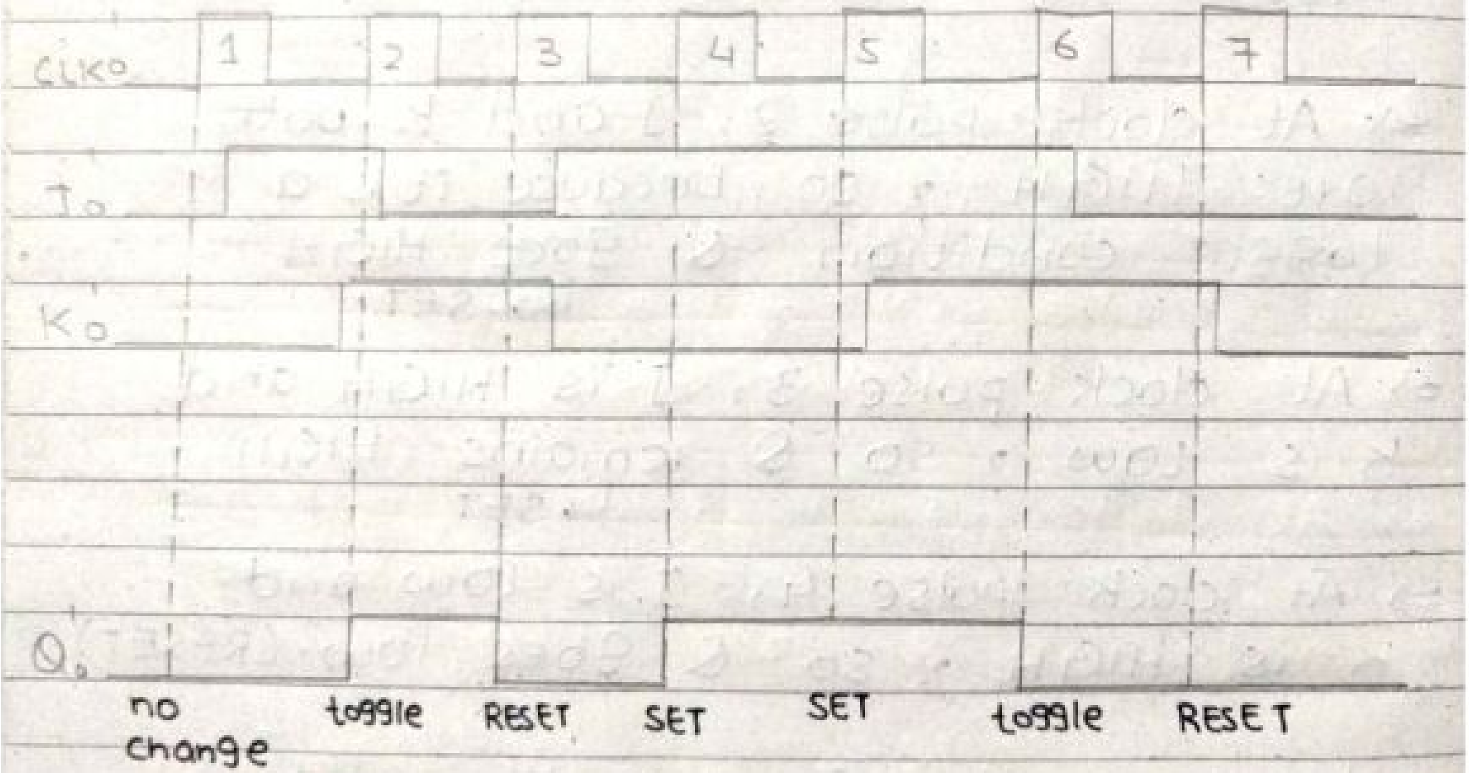
=> At clock pulse 4, J is LOW and K is HIGH, so Q goes low. (RESET)

=> At clock pulse 5, J is low and K is HIGH, so Q remains low. (RESET)

EXAMPLE : 7-7

The waveforms are applied to the flip flop as shown. Determine the Q output, starting in the RESET state.





=> Since, this is a positive edge triggered flip flop, the Q output will change only on the positive going edge of the clock pulse

=> At clock pulse 1, J is Low and K is Low, so Q does not change from its initial state and remains RESET.

=> At clock pulse 2, J and K both are HIGH, because it is a toggle condition, Q goes HIGH. (SET)

=> At clock pulse 3, J is low and K is HIGH, so Q goes Low. (RESET)

=> At clock pulse 4, J is HIGH and K is Low, so Q goes HIGH. (SET)

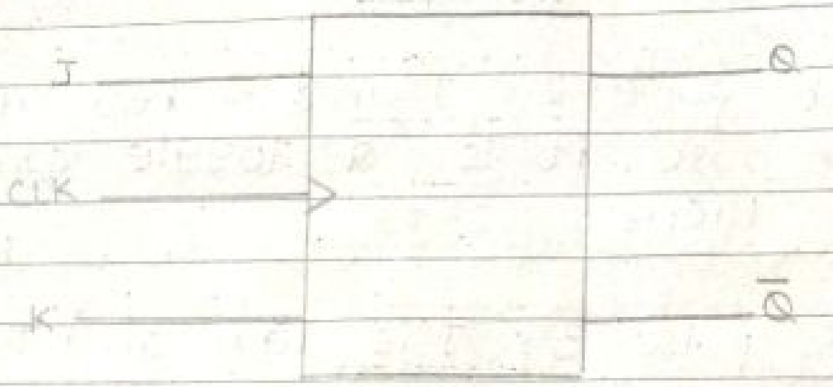
=> At clock pulse 5, J is HIGH and K is Low, so Q remains HIGH (SET)

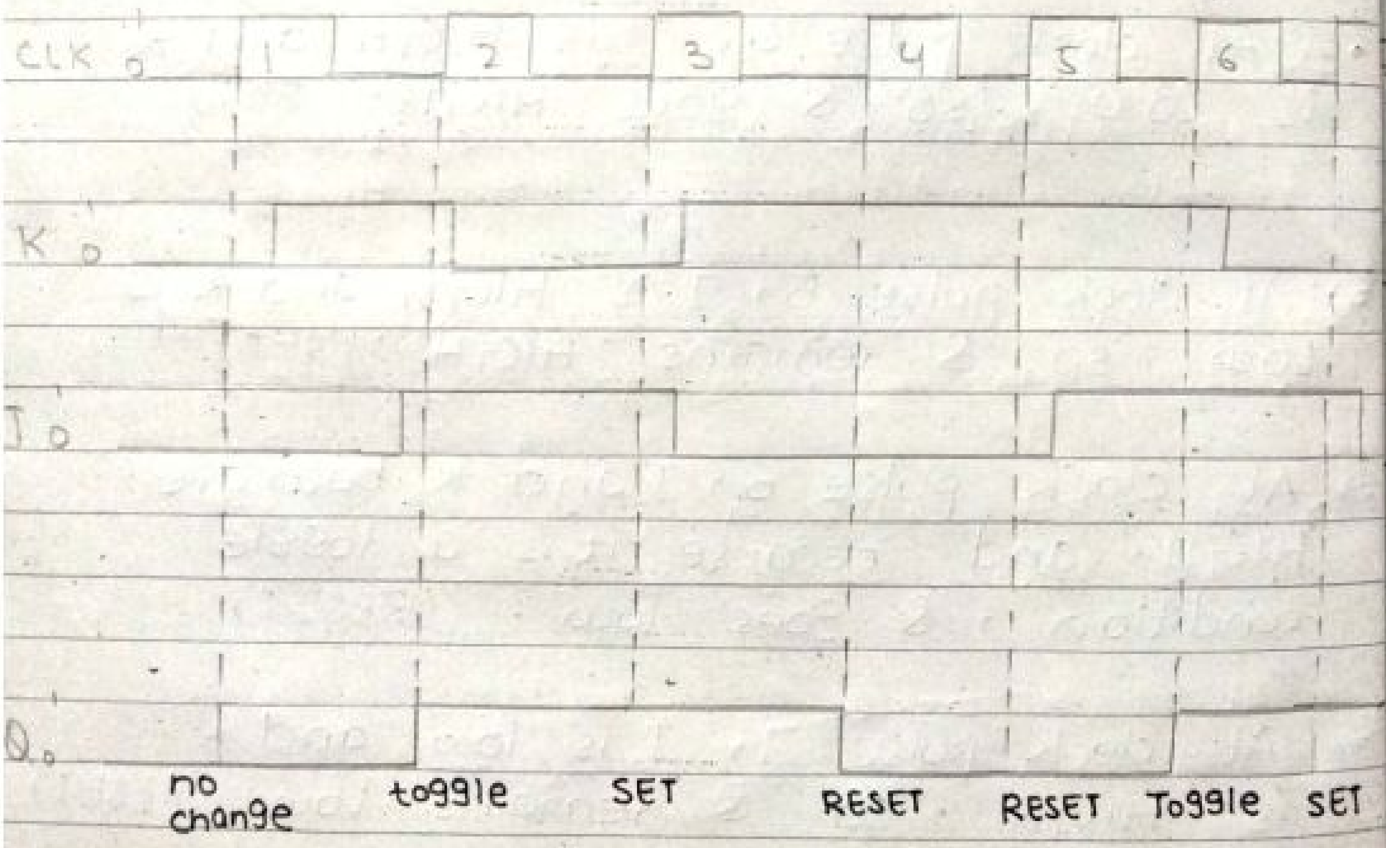
=> At clock pulse 6, J and K both are HIGH, and because it is a toggle condition, Q goes low. (RESET)

=> At clock pulse 7, J is low and K is HIGH, so Q remains low. (RESET)

Related Problem

Interchange the J and K inputs determine the resulting Q output.





=> Since, this is a positive edge triggered flip flop, the Q output will change only on the positive going edge of the clock pulse.

=> At clock pulse 1, J is low and K is low, so Q does not change from its initial state and flip flop remains RESET.

=> At clock pulse 2, J and K both are high, and because it is a toggle condition, Q goes high. (SET)

=> At clock pulse 3, J is HIGH and K is Low, so Q remains HIGH. (SET)

=> At clock pulse 4, J is low and K is HIGH, so Q goes low. (RESET)

=> At clock pulse 5, J is low and K is HIGH, so Q remains low. (RESET)

=> At clock pulse 6, J and K both are HIGH, and because it is a toggle condition Q goes HIGH.

=> At clock pulse 7, J is HIGH and K is Low, so Q remains HIGH. (SET)