

UNIT-2

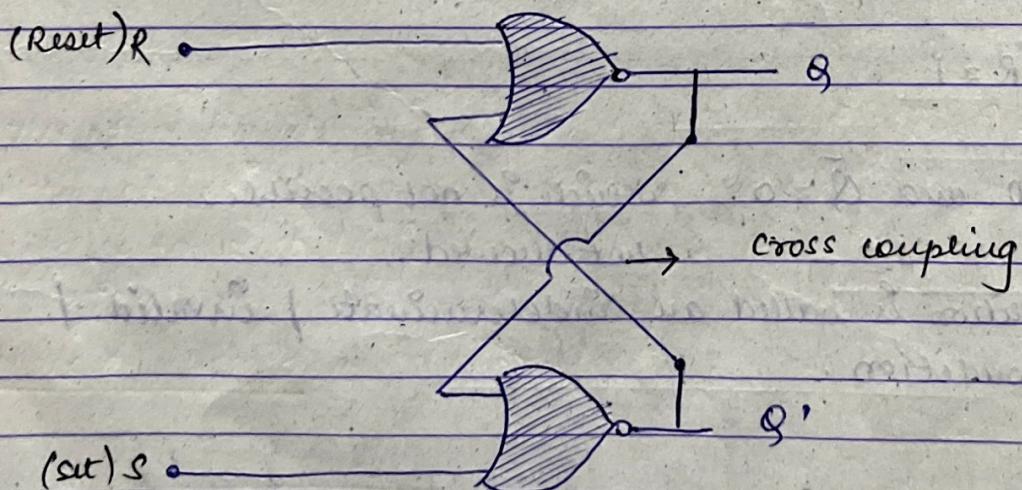
Sequential Circuit

In sequential circuits present output depends on present inputs, as well as on past output.

Therefore, it must have memory.

Latch (lock)

- 1 bit memory cell, which can hold bit '1' or bit '0' as its o/p.
- locks one bit (0 or 1). It is used to make memory (RAM, cache, etc.)
- S-R-latch (Set-Reset latch).



Let's understand its working assuming following sets of circuits are applied one by one in a sequence assuming initially the latch is set.

i.e. $Q = 1$ and $\bar{Q} = 0$ initially).

i) $S = 0, R = 1$

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

i.e. Latch is reset

ii) $S = 1, R = 0$

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

i.e. Latch is set

iii) $S = 0, R = 0$

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

i.e. it gives no change in outputs

iv) $S = 1, R = 1$

it gives

$Q = 0$ and $\bar{Q} = 0$, which is not possible.
not allowed.

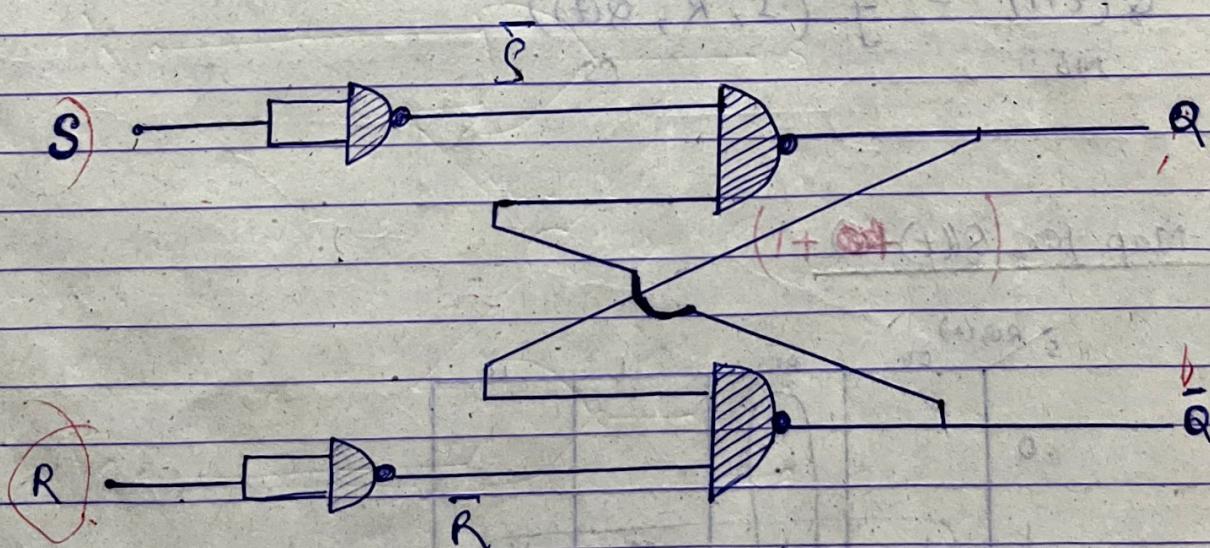
So, this condition is called an indeterminate / invalid / forbidden condition.

Truth Table
of S-R Latch

Let $Q(t)$ be present state

		N.S		
S	R	$Q(t+1)$	$\bar{Q}(t+1)$	Action
0	0	$Q(t)$	$\bar{Q}(t)$	No change in state
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Invalid state / forbidden

S-R Latch using NAND Gates



Note:- In case of S-R Latch using NAND Gates, the outputs $Q = \bar{Q} = 1$ i.e. both are same (invalid) when the inputs $S = R = 1$.

State Table of S-R latch

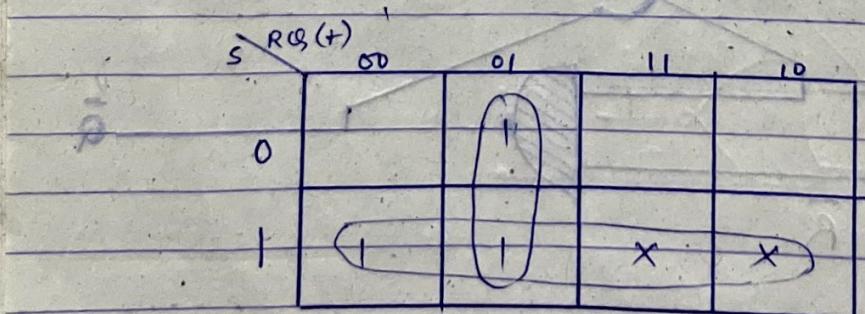
inputs		present state	next state
S	R	$Q(t)$	$Q(t+1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

State Equation - (Characteristic Equation)

$$Q(t+1) = f(S, R, Q(t))$$

NS PS

K-Map for $(Q(t) \oplus 1)$

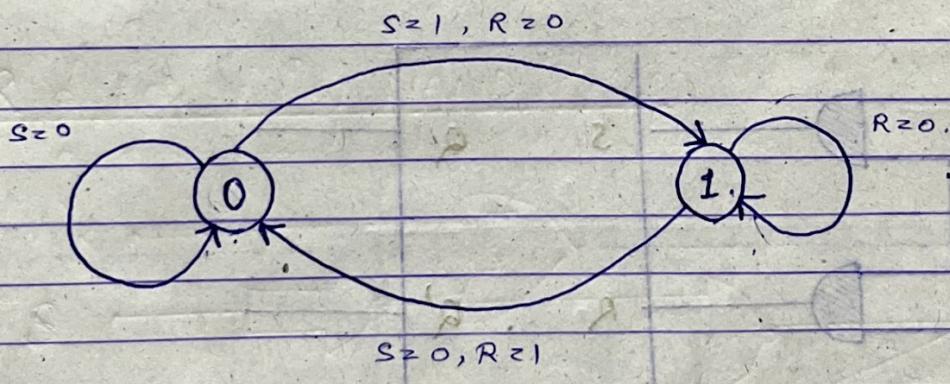


$$Q(t+1) = S + R'Q(t)$$

Excitation Table :- Combination of PS & NS as i/p &
at the o/p we have the i/ps s/R

Present State $Q(t)$	Next State $Q(t+1)$	Excitation s	Inputs R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

State Diagram.

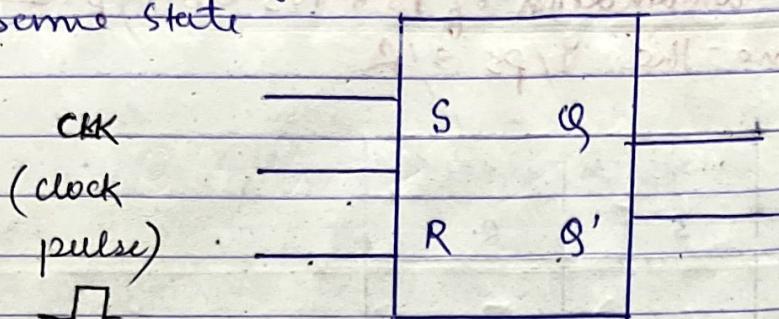


Difference b/w latch and flip-flop.

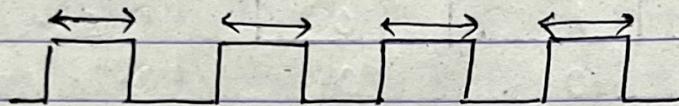
- 1) A latch with an additional enabling input i.e. clock pulse is called a flip-flop.
 A flip-flop is a clock-driven latch.

FLIP FLOP

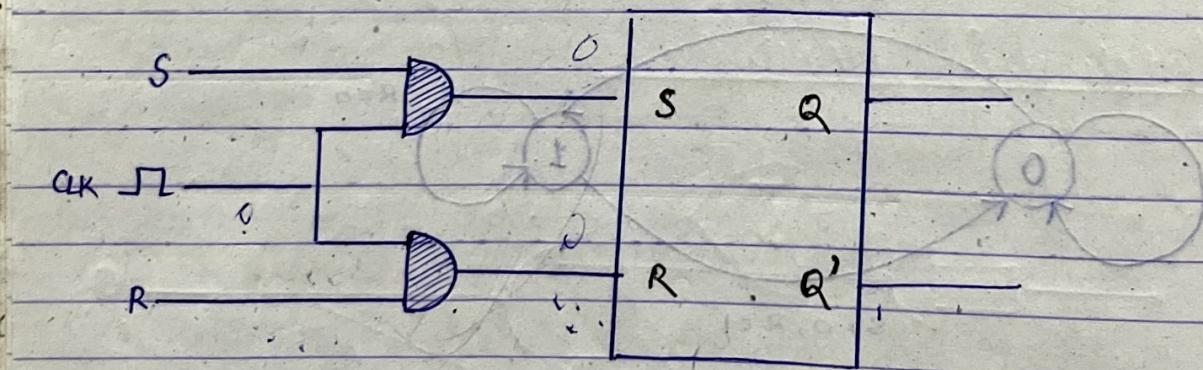
FF is a latch with an additional enabling input which is called clk.
 The significance of this clk pulse is that when a clk pulse is present at this p/p then it is enabled i.e. it responds to its p/p's in a normal way, else it is disabled i.e. ignores the p/p & remain in same state.



S-R Flip-flop



S-R Flip Flop.

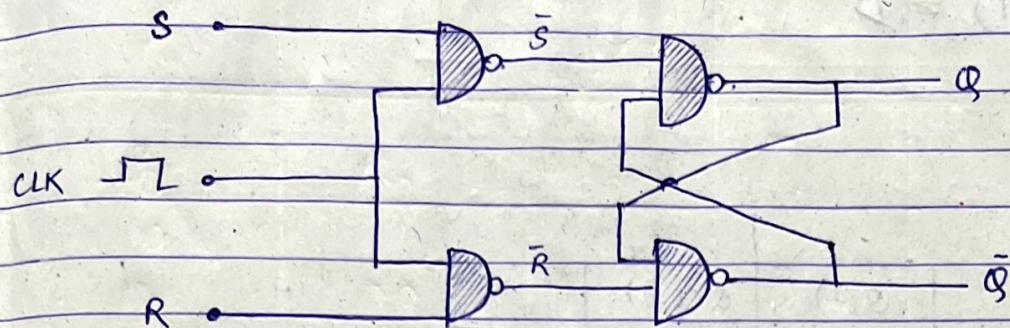


Truth Table

CLK	S	R	$Q(t+1)$
0	X	X	$Q(t)$, ✓
1	0	0	$Q(t)$
1	0	1	0
1	1	0	1
1	1	1	Invalid

Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

S-R flip flop using NAND gate



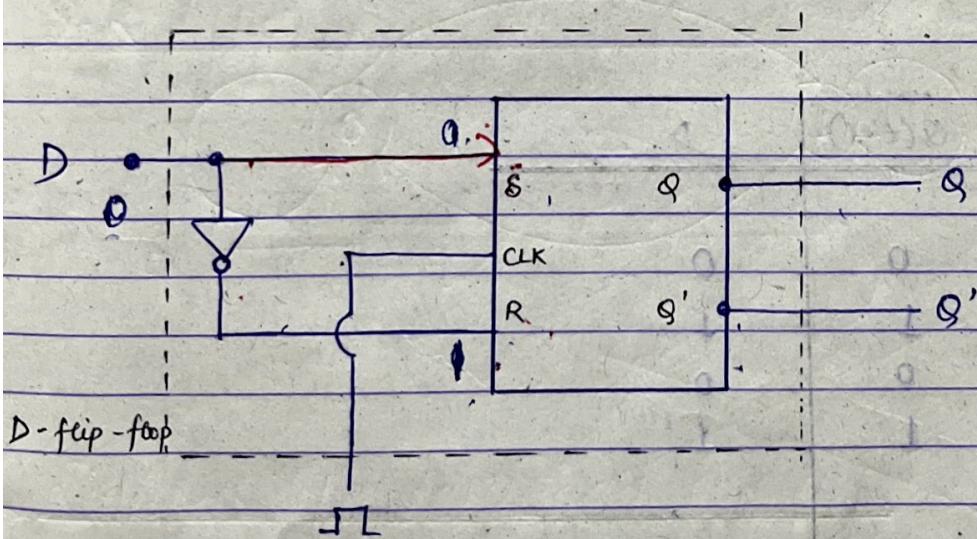
00	01	10	11
1	X	1	1

$$Q(t+1) = S + Q \bar{R}$$

characteristic eq

D - Flip-Flop

(Data/Delay)



Truth Table

	CLK	D	Q(t+1)
0	x	Q(t)	Q(t)
1	0	0	0
1	1	1	1

State Equation (Characteristic Equation)

$$Q(t+1) = D \quad \checkmark$$

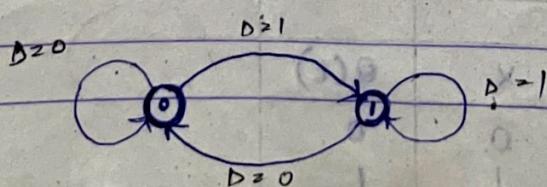
State Table

D	present state $Q(t)$	next state $Q(t+1)$
0.	0	0
0	1	0
1	0	1
1	1	1

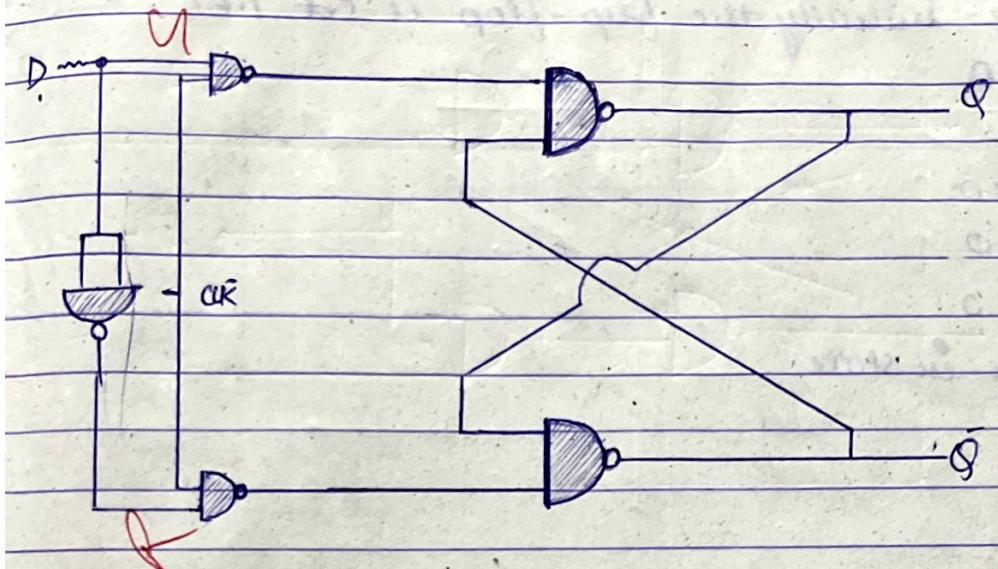
Excitation Table

present state $Q(t)$	next state $Q(t+1)$	D.
0	0	0
0	1	1
1	0	0
1	1	1

State Diagram

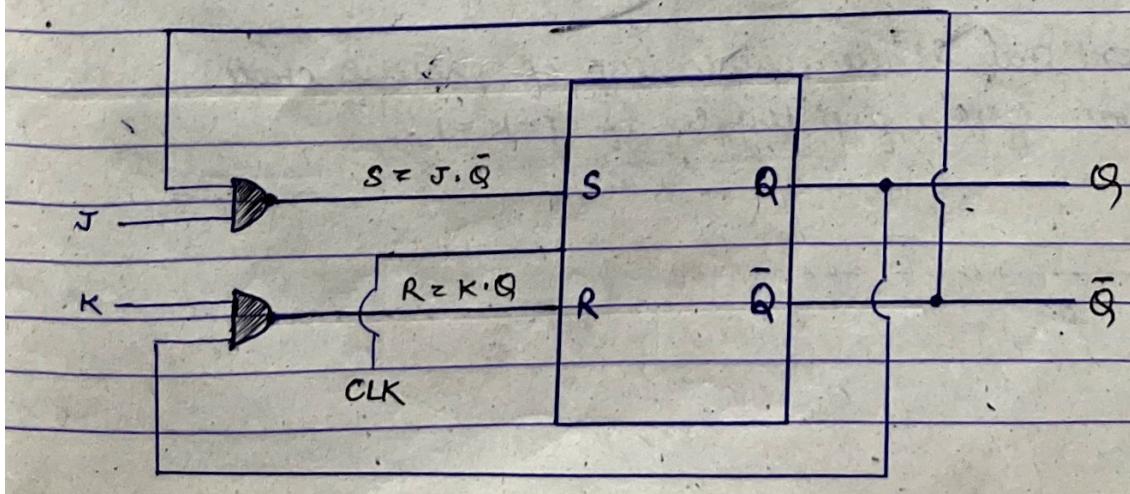


D-Flip flop using NAND Gates



J-K Flip Flops

J-K flip flop is designed in order to overcome the forbidden state of S-R flip flop which arises when both the inputs are 1. The S-R flip flop can easily converted to J-K flip flop as shown below:-



Now let's understand its working, considering following sets of inputs are applied to it in a sequence one by one assuming initially the flip-flop is set i.e. $Q=1$ and $\bar{Q}=0$

i) $J=0, K=0$

$\therefore S=0, R=0$

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

i.e. no change in state.

ii) $J=0, K=1$

$\therefore S=0, R=1$

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

i.e. flip flop is reset.

iii) $J=1, K=0$

$\therefore S=1, R=0$

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

i.e. flip flop is set.

iv) $J=1, K=1$

$\therefore S=1, R=1$

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

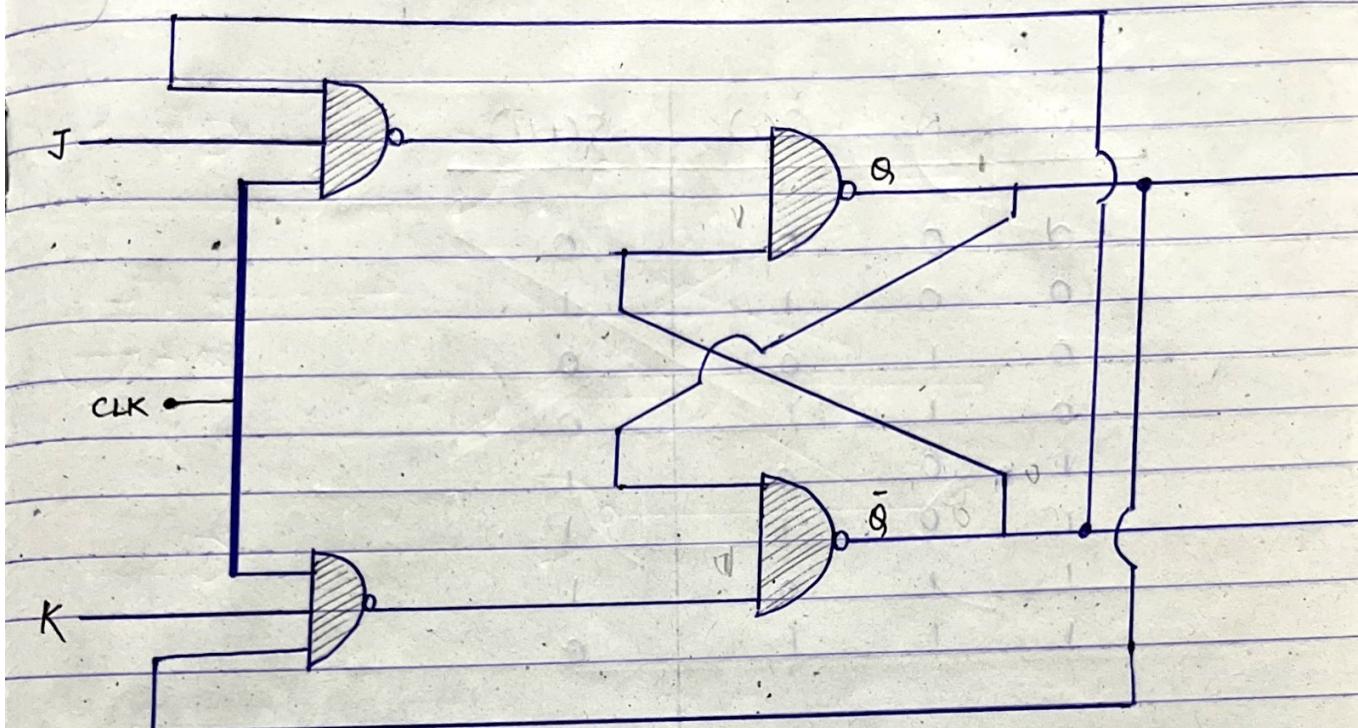
i.e. flip flop is reset.

i.e. the next state is the complement of present state

i.e. the state of flip flop toggles if $J=K=1$.

Jack Kilby

J-K flip flop using NAND Gate



Truth Table of J-K flip flop

CLK	J	K	$Q(t+1)$	$\bar{Q}(t+1)$	Action
0	x	x	$Q(t)$	$\bar{Q}(t)$	No change
1	0	0	$Q(t)$	$\bar{Q}(t)$	No change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	$\bar{Q}(t)$	$Q(t)$	Toggle

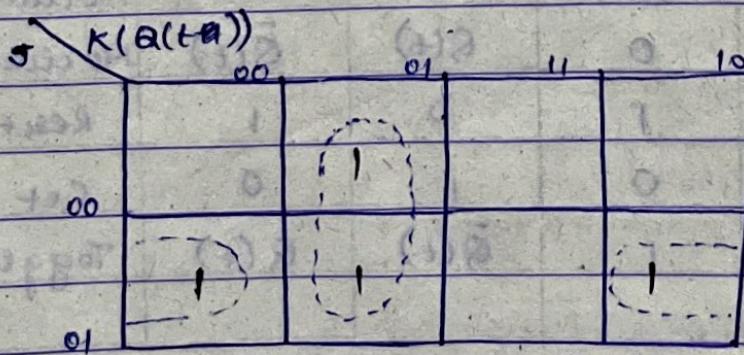
T	K	$Q(t)$	$Q(t+1)$
0	0	x	$Q(t)$
0	1	x	0
1	0	x	1
1	1	x	$Q(t)$

State table of J-K flip flop.

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

Characteristic Equation

K-map for $Q(t+1)$

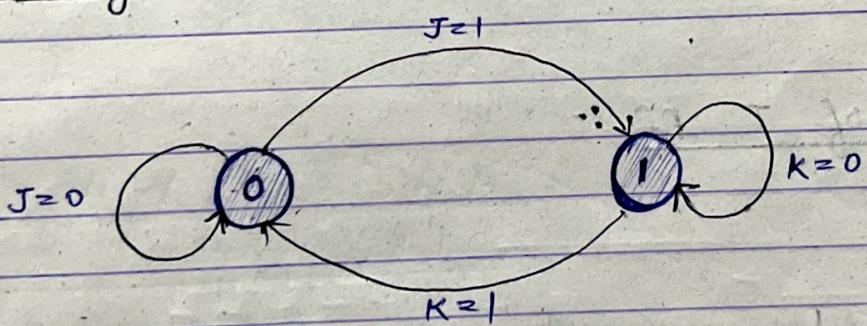


$$\therefore Q(t+1) = JQ'(t) + K'Q(t)$$

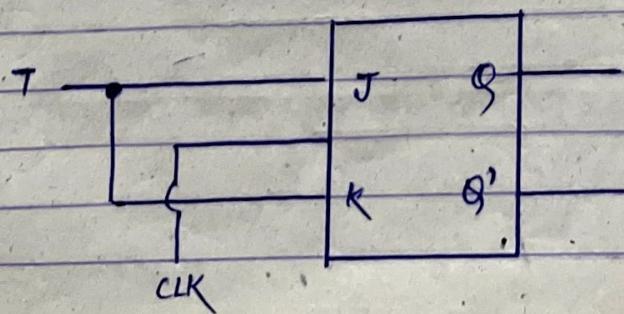
Excitation Table

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

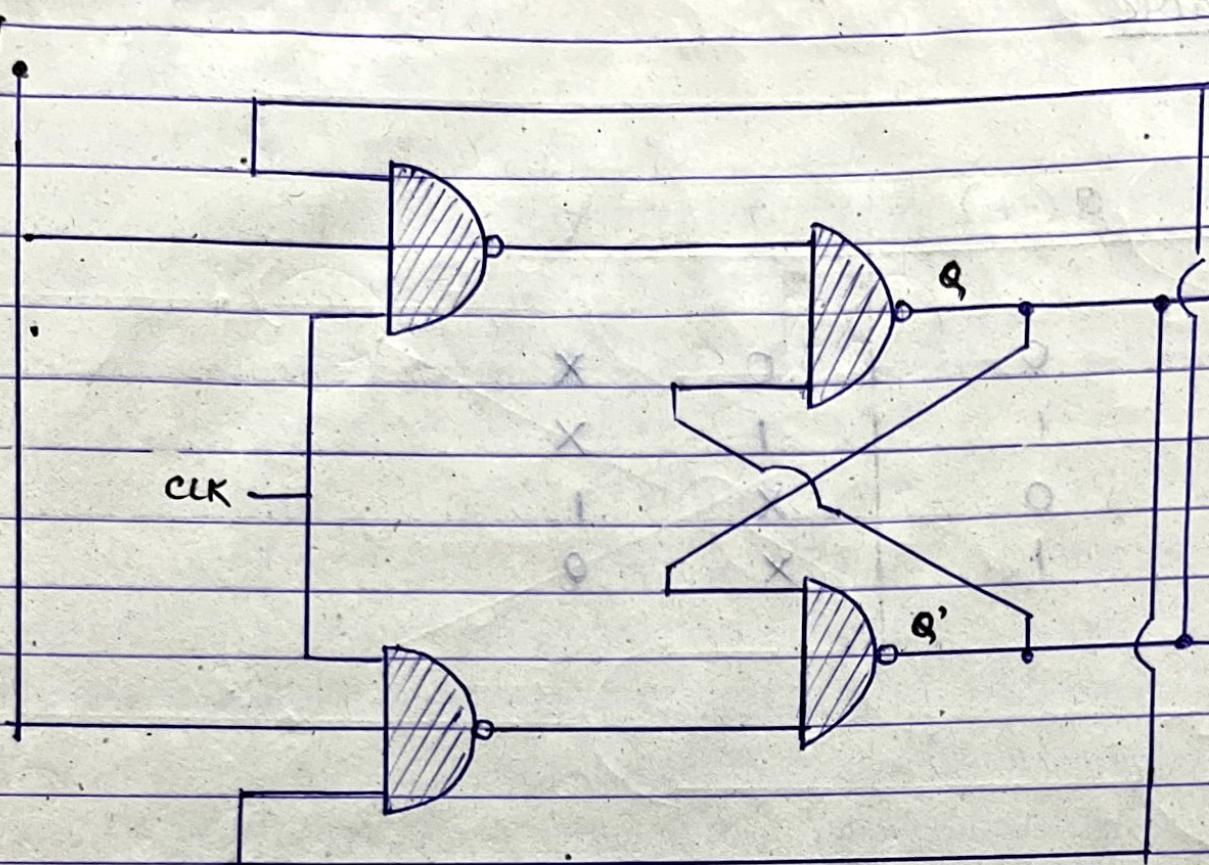
State Diagram.



Toggle Flip Flop



Using NAND Gates



Truth Table of T ff :-

CLK	T	$Q(t+1)$	$Q'(t+1)$	Action
0	x	$Q(t)$	$Q'(+) \checkmark$	No change (\because FF is disabled)
1	0	$Q(+) \checkmark$	$Q'(+) \checkmark$	No change
1	1	$Q'(+) \checkmark$	$Q(+) \checkmark$	Toggle

State Table

T	present state $Q(t)$	next state $Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic Equation

$$Q(t+1) = T \oplus Q(t)$$

T

		$Q(t)$	
		0	1
0	0	1	(1)
	1	(1)	

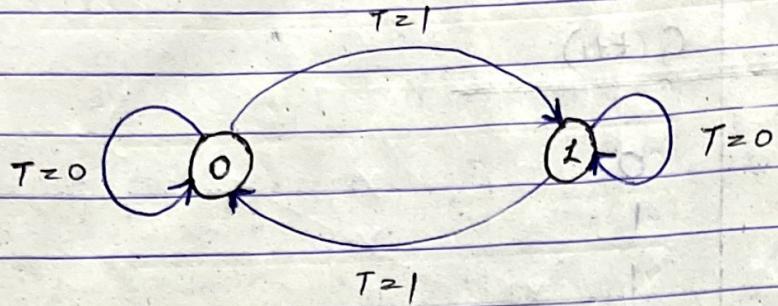
from K-map

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

Excitation Table

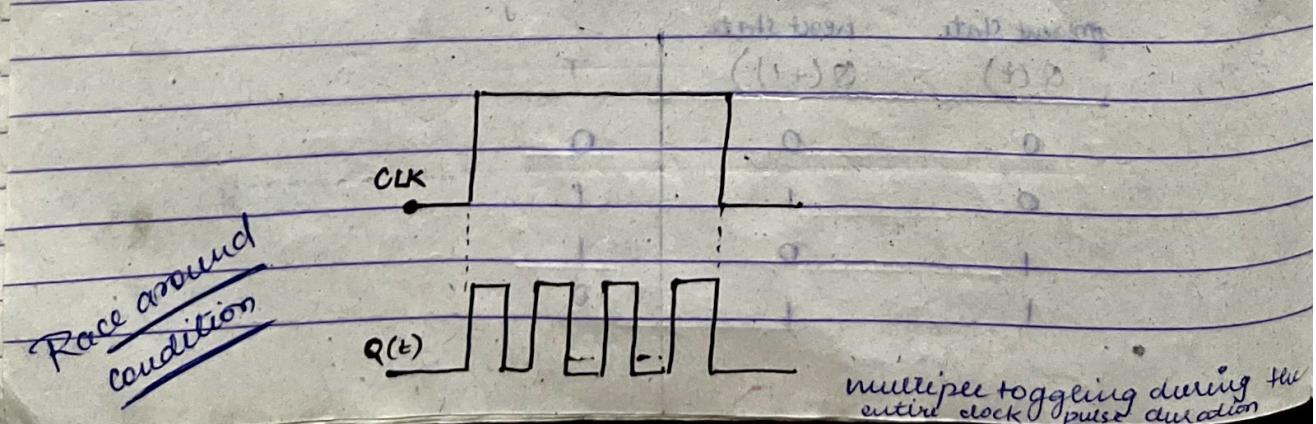
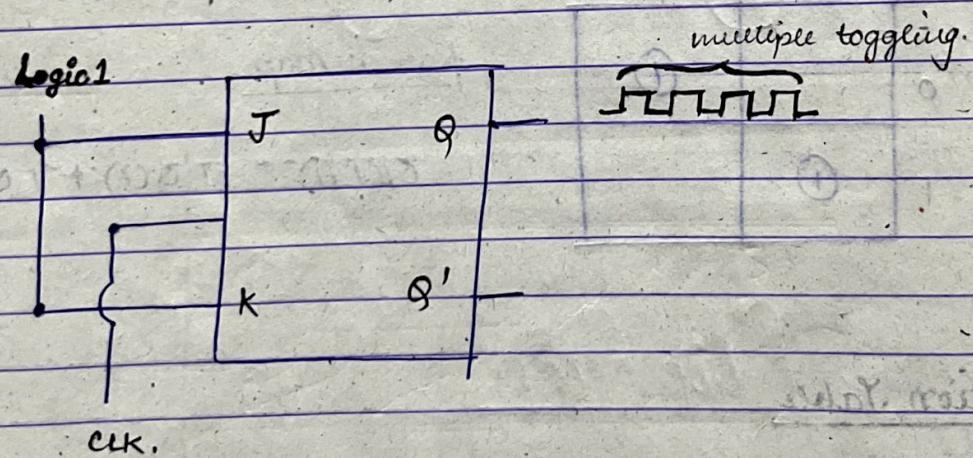
present state $Q(t)$	next state $Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

State Diagram



Race around Condition

occurs in level triggered J-K flip flop when J and K inputs are connected logic 1.



as we see in the figure when the inputs S and K are tied the logic 1, then its output keeps toggling from high to low and low to high during the entire clock pulse duration. At the end of clock pulse what value the outputs attain is totally ambiguous.

This is called Race Around Condition.

This can be overcome if instead of level triggered J-K FF we use edge triggered J-K FF.

FF Conversion

1) Convert SR to D (Realize D FF using SR FF)

Excitation Table of S-R FF

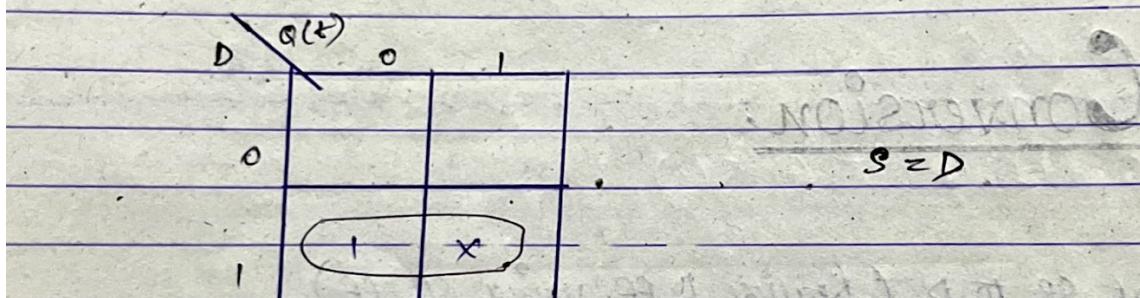
$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

- 1) Consider the TT of Destination FF & extend it as excitation Table of Source FF
- 2) Draw the K-map for source FF i/p variables it will provide Expression for conversion
- 3) Draw the cbt according to the K-map Expression.

Conversion Table :-

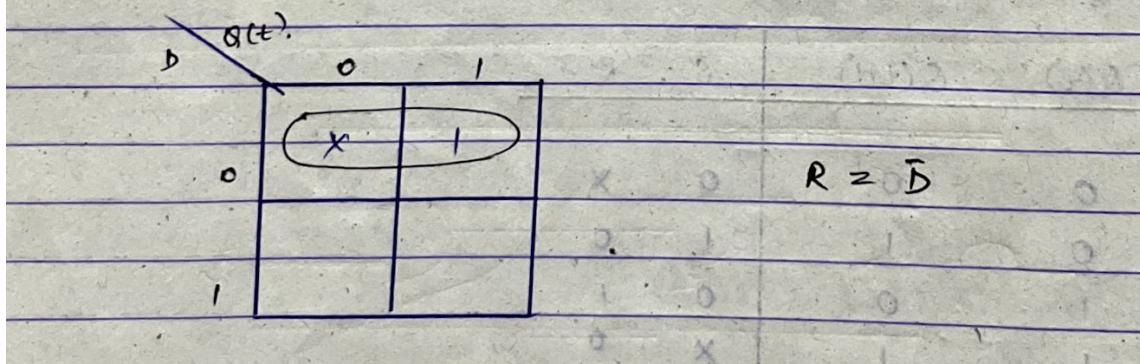
D	$Q(t)$	$Q(t+1)$	S	R
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

K-Map for S

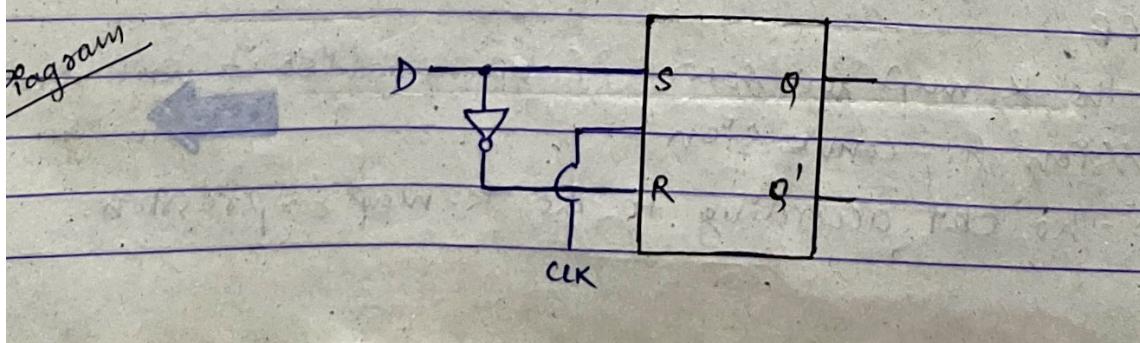


$$S = D$$

K-Map for R



$$R = \bar{D}$$



2) Convert SR to JK (Realize JK FF using SR FF)

Excitation Table of SR FF

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Conversion Table :-

J	K	$Q(t)$	$Q(t+1)$	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

$$S = J \cdot Q(t)$$

$$R = K \cdot Q(t).$$

K-Map for S

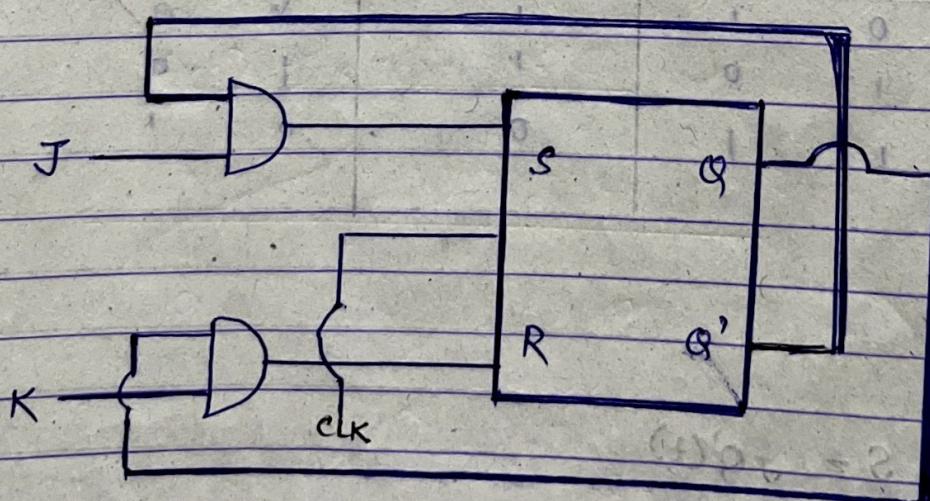
		00	01	11	10
		J	K $\bar{Q}(t)$		
J	K $\bar{Q}(t)$	00	X		
		01	1	X	1

$$S = J \cdot K\bar{Q}(t)$$

K-Map for R

		00	01	11	10
		J	K $\bar{Q}(t)$		
J	K $\bar{Q}(t)$	00	X		
		01	X	1	1

$$R = K \cdot Q(t)$$



3) Convert JK to T (Realize TFF using JK FF)

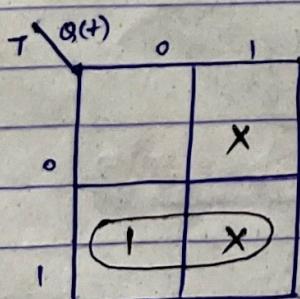
Excitation table of JK FF

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

Conversion Table :-

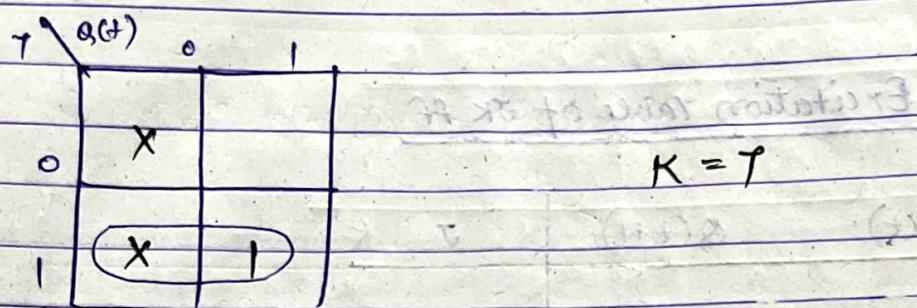
T	$Q(t)$	$Q(t+1)$	J	K
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

K-Map for J

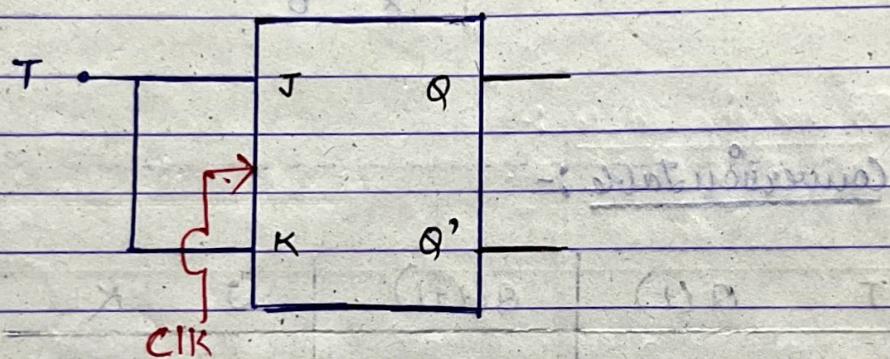


$$J = T$$

K-map for T



~~Diagram~~



- 4) Convert T to D (Realize D FF using T FF)

Excitation Table of T

$Q(t)$	$\bar{Q}(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

Conversion Table :-

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

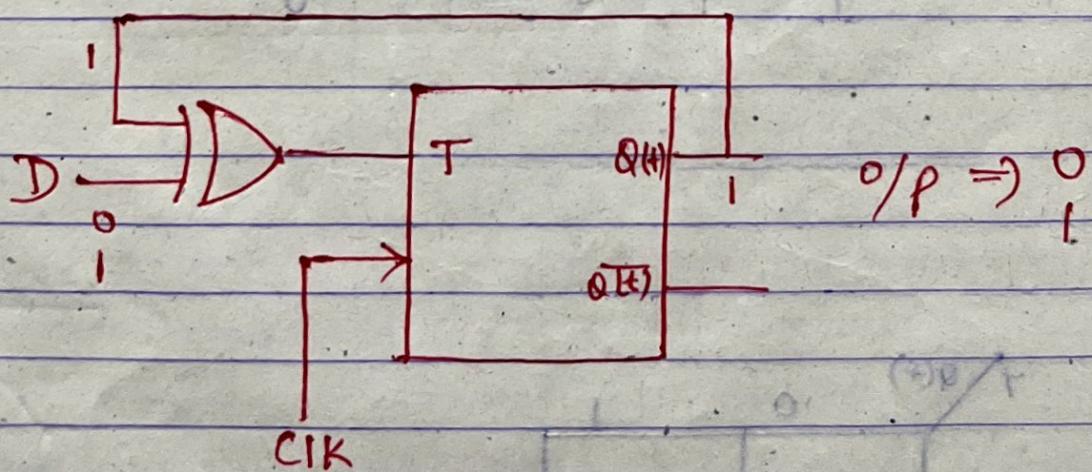
K-map for T

K-map for T:

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

$$T = D'Q(t) + D \cdot Q'(t)$$

$$T = D \oplus Q(t)$$



5) Convert D to T (Realize TFF using DFF)

Excitation Table of D-FF

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

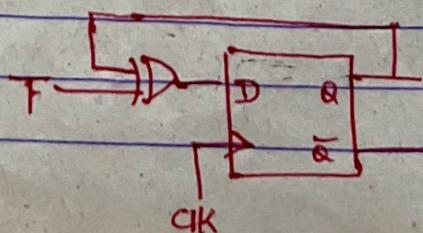
Conversion Table :-

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

K-Map for D

	$Q(t)$	0	1
0	0	0	1
1	1	1	0

$$D = T \oplus Q(t)$$



6) Convert SR to T (Realize TFF using SR FF)

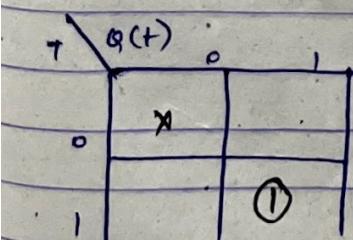
Excitation Table for SR FF

$S(t)$	$Q(t+1)$	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Conversion Table :-

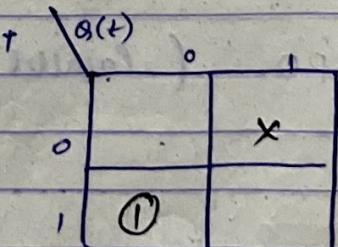
T	$Q(t)$	$Q(t+1)$	S	R
0	0	0	0	x
0	1	1	x	0
1	0	1	1	0
1	1	0	0	1

K-Map for R

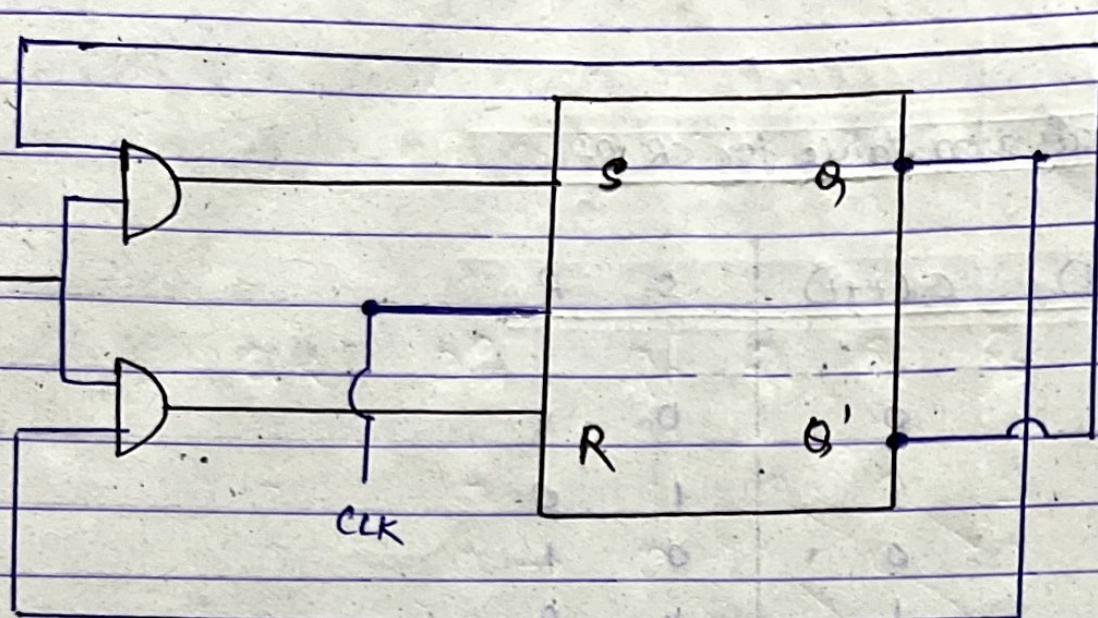


$$R = T \cdot Q(t)$$

K-Map for S

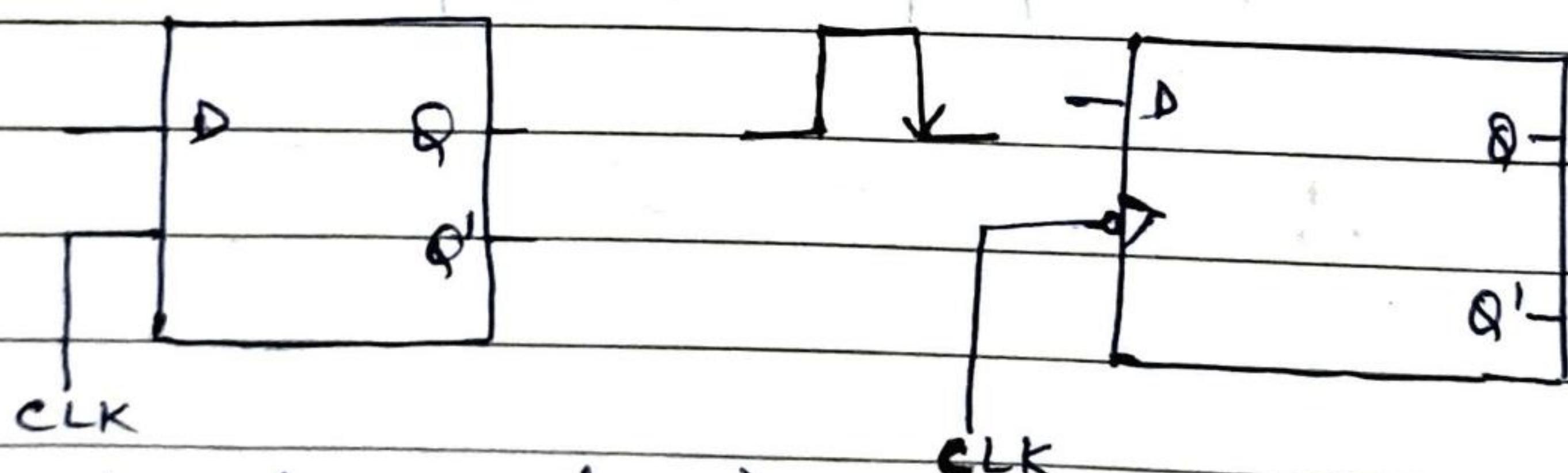


$$S = T \cdot Q'(t)$$



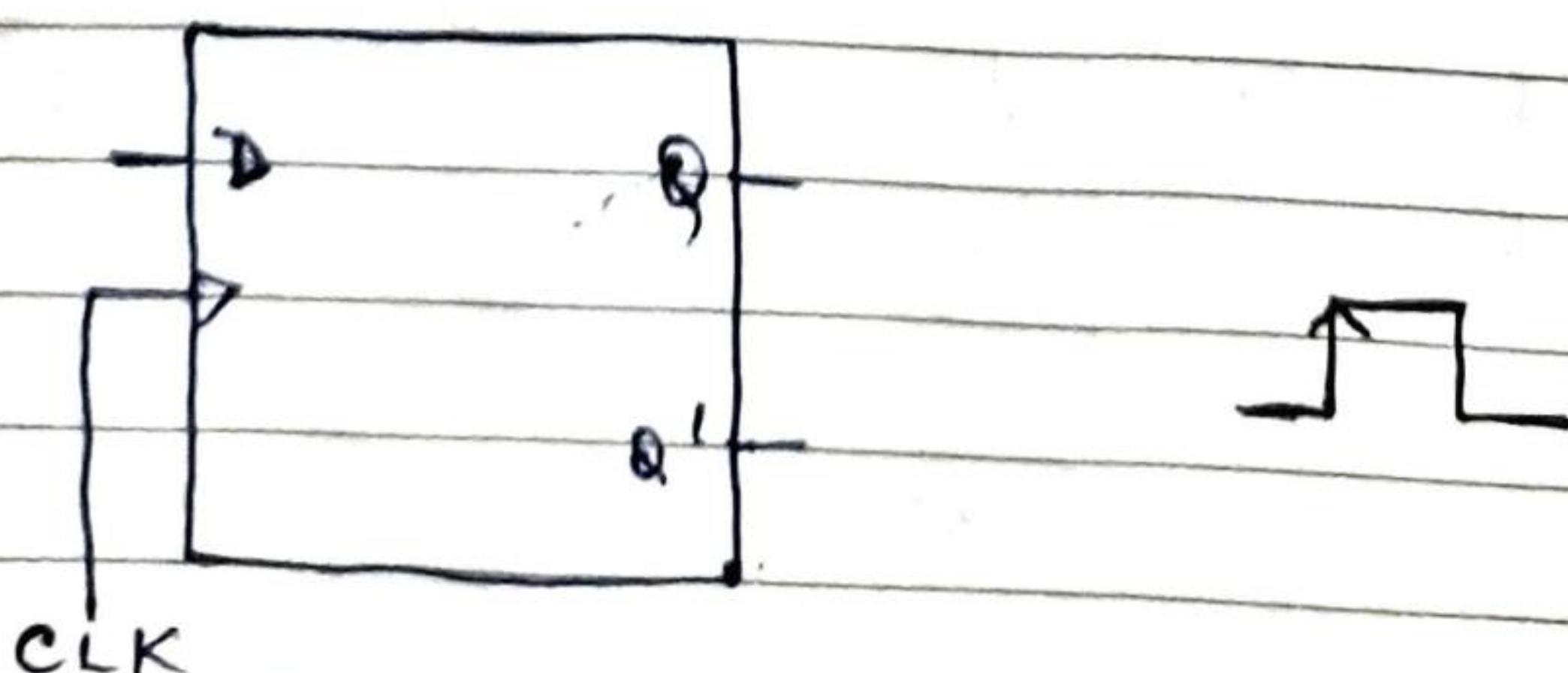
Level Triggered FFs vs Edge Triggered FFs:

Level Triggered FFs remains enabled for entire clock pulse duration whereas the Edge Triggered FFs are enabled for a moment when the clock pulse makes transition either from low state to high state (in case of positive edge triggered flip flop) or it makes transition from high to low state (in case of negative edge triggered flip flop).



(Level Triggered FF)

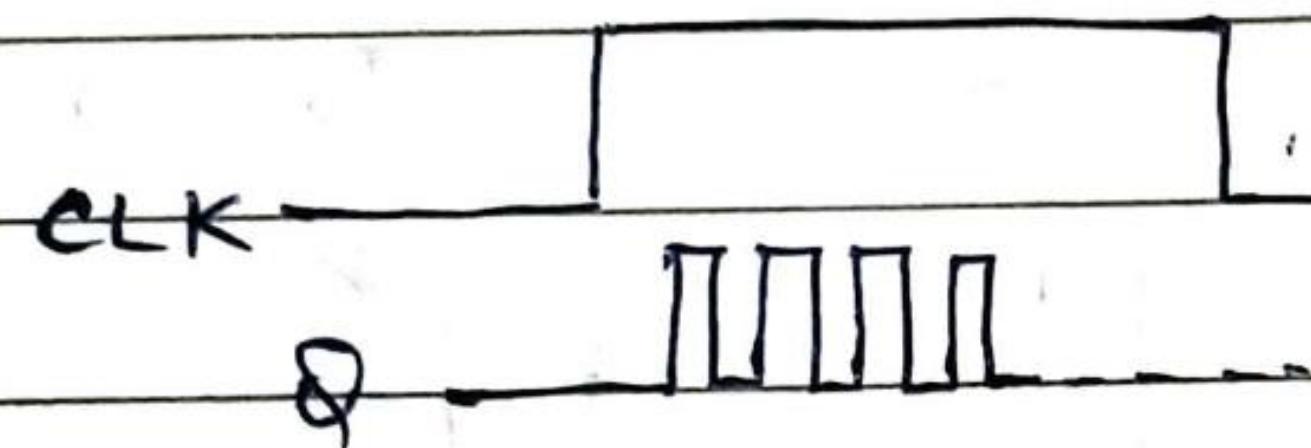
(-ve edge Triggered FF)



(+ve edge Triggered FF)

✓ Race around Condition in JK ff :-

In case of level Triggered JK FF when $J=K=1$, then multiple toggling will be taking place at the o/p of J-K FF, (since) it is level triggered as shown below) during the entire duration of clock pulse



So, it is clear that at the end of clock its o/p is ambiguous. This undesired condⁿ is called Race-around condⁿ.

This can be overcome by using an edge Triggered JK FF instead of Level Triggered FF.

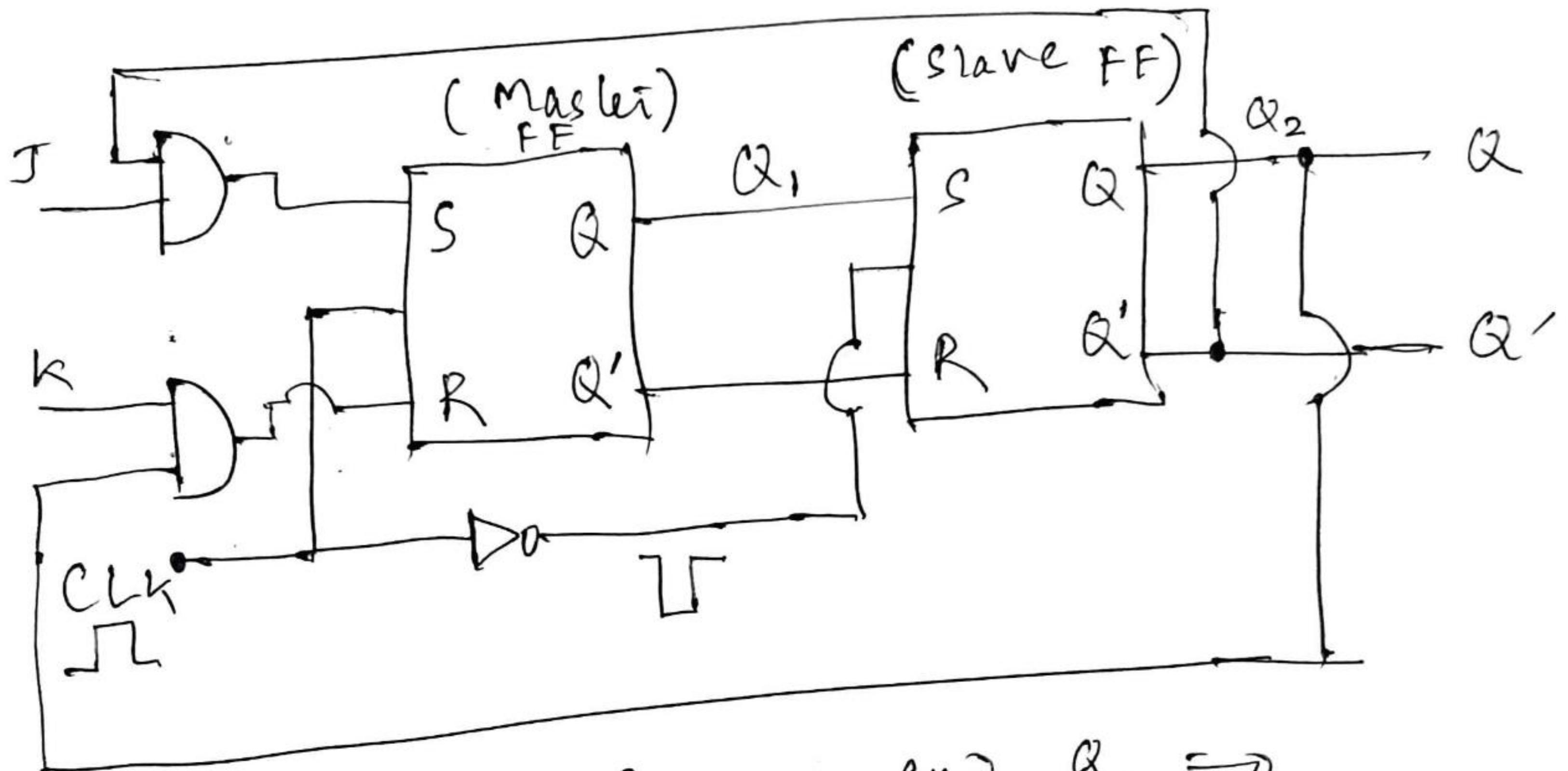
In this case only one toggling per clock pulse takes place when $J=K=1$. Since the flip-flop gets enabled for a moment either at (+)ve edge or at (-)ve edge of clock

Master slave JK FF

It overcomes the race around condition in level triggered J-K flip flop.

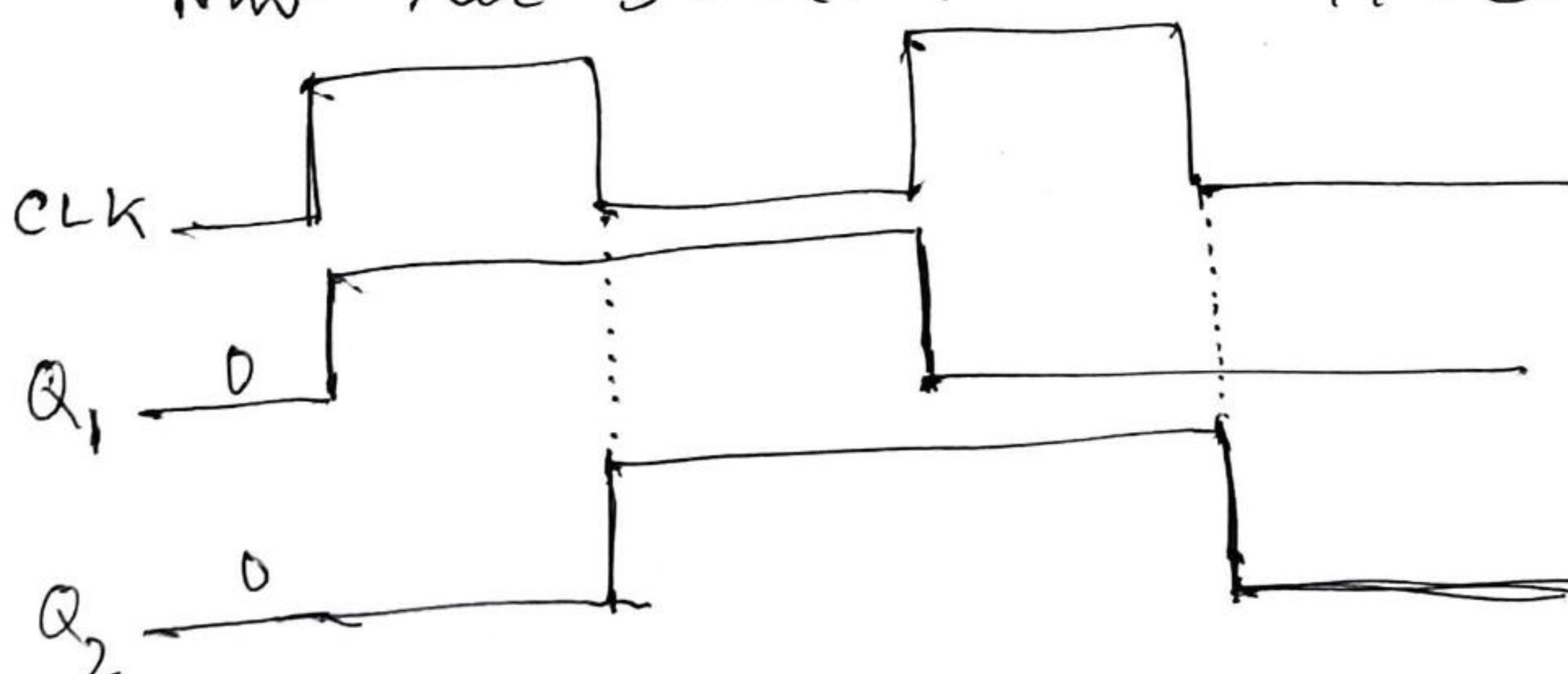
When $J = K = 1$.

It ensures only single toggling per clock pulse.



Let initially $Q_1 = 0$ and $Q_2 = 0$
 $(Q'_1 = 1)$ $(Q'_2 = 1)$

Now let $J = K = 1$ is applied.



So we can observe the final output Q_2 toggles only once per clock cycle.