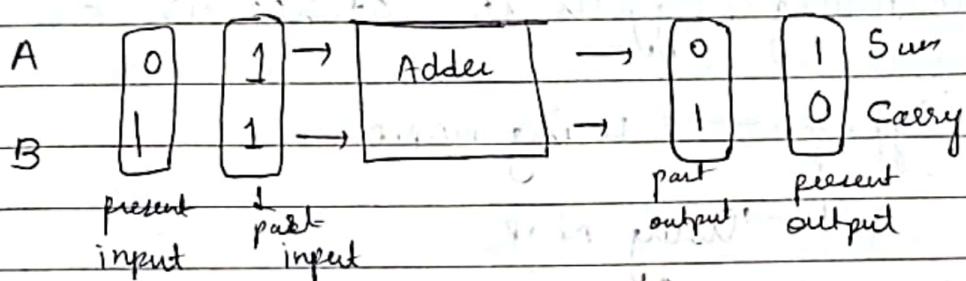


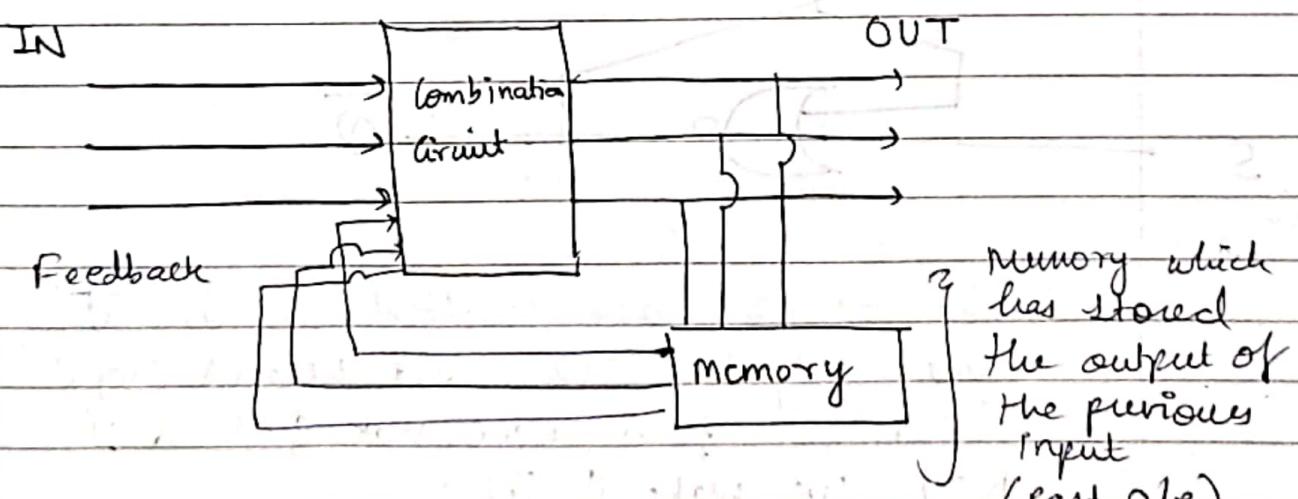
SEQUENTIAL CIRCUITS

In sequential circuits, the present output depends on the present input as well as the past output / outputs

Output / outputs
(In combinational circuits present output depends only on the present input)



This is a combinational circuit, present output does not depend on past outputs



This whole is our sequential circuit. In sequential circuit everything is combinational except memory & feedback. Memory is a flip-flop.

So basically in sequential circuits we have the memory element which is used to store the past output which is not present in combinational circuit.

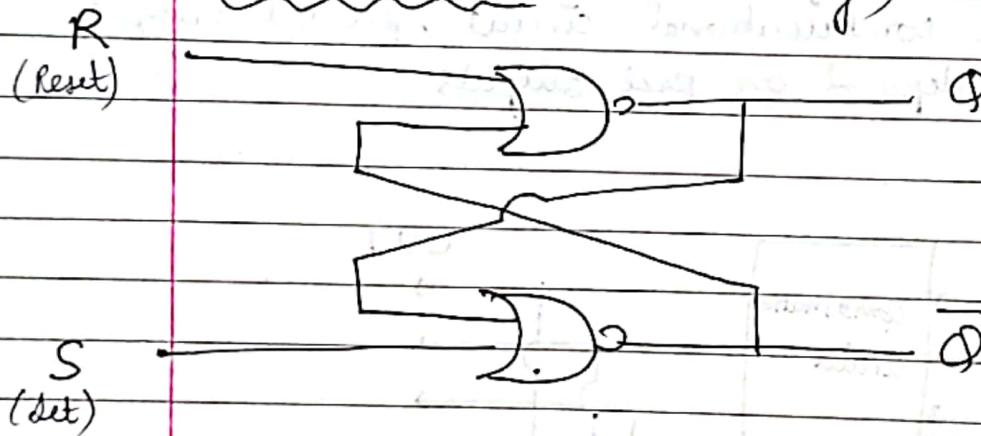
SR LATCH

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

→ Two types → using NAND

↳ Using NOR

USING NOR = (active high)



(Reset → we have resetted our circuit and output $Q=0$. Set means output $Q=1$. So if $S=1$ we make $Q=1$ & if $R=1$ we make $Q=0$)

NOR GATE :

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Case 1: $S=0, R=1$

If any input is 1, the output of NOR gate will be zero without seeing the other input. So as $R=1$, \bar{Q} will be 0 for sure.

(So when $R=1, \bar{Q}=0$ proved)

$\bar{Q}=1$ as $S=0$ and second input becomes 0.

$$\Rightarrow S=0, R=1, Q=0 \text{ & } \bar{Q}=1$$

Now we want to store the data, so if we remove the input data must be stored. So now we make $S=0, R=0$, the output should not change.

So \bar{Q} was 1 from prev case, $R=0$ and other input is 1, so $Q=0$.

$$\Rightarrow S=0, R=0, Q=0 \text{ & } \bar{Q}=1$$

So output is same as the previous case, hence we call this condition memory.

Case 2: $S=1, R=0$

Since $S=1$, \bar{Q} will be 0 due to NOR gate system. so second input with R is 0, and hence $Q=1$ (When $S=1, Q=1$ is proved)

$$\Rightarrow S=1, R=0, Q=1 \text{ & } \bar{Q}=0$$

Now again we remove the input data. And so $S=0, R=0$. Since here S is changed, \bar{Q} may change but Q will be same. So $Q=1$, second input with S is 1, $Q=0$

$$\Rightarrow S=0, R=0, Q=1 \text{ & } \bar{Q}=0$$

So again $S=0, R=0$ give us the memory state.

Case 3: $S=1, R=1$

From NOR Truth table, both Q and \bar{Q} will be zero. So there is a contradiction $Q = \bar{Q}$ which is not possible, they should be complement of each other. Thus, this configuration is not used.

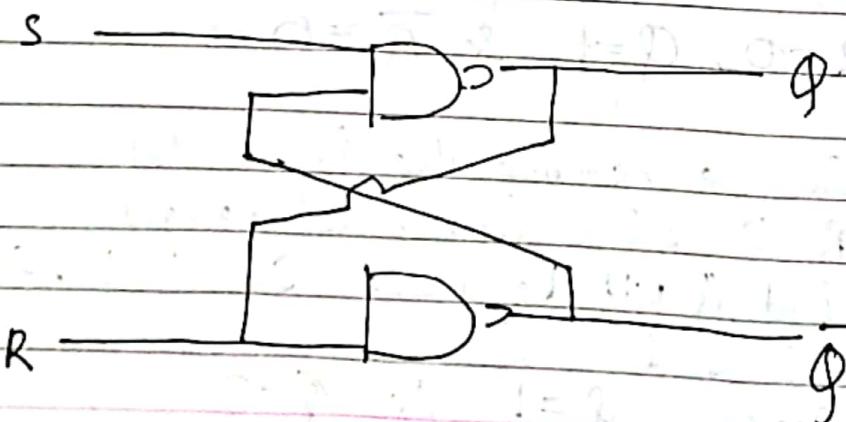
$S=1, R=1, Q=0 \& \bar{Q}=0 \times$ Not used

Now, if we make $S=0$ & $R=0$ the state is not stored as output is different.

(Case 3 is not used in SR flip flop)

S	R	Q	\bar{Q}
0	0	Memory (as before)	
0	1	0	1
1	0	1	0
1	1	(not used)	

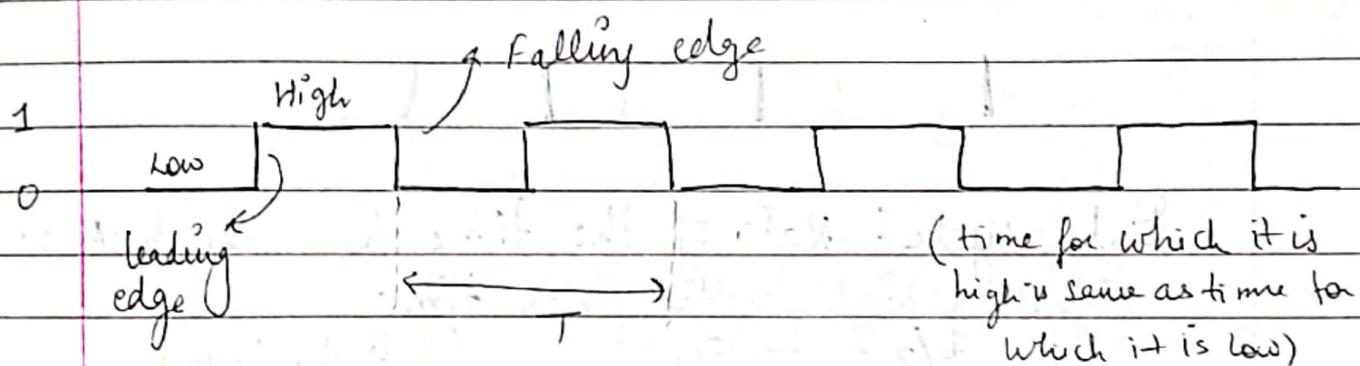
USING NAND: (active low)



S	R	Q	\bar{Q}
0	0	(Not Used)	
0	1	1	0
1	0	0	1
1	1	Memory (as before)	

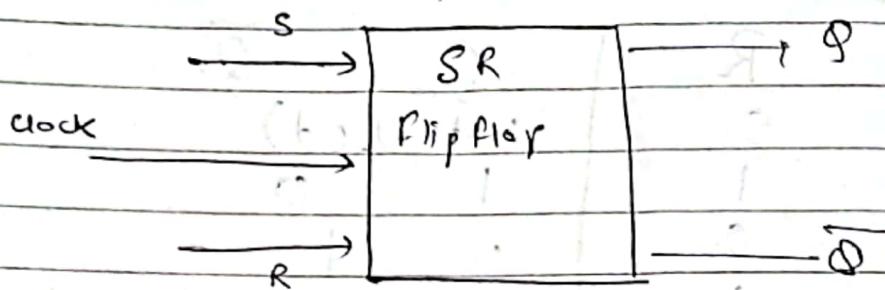
CLOCK:

Clock is just a signal that goes low to high and repeats.

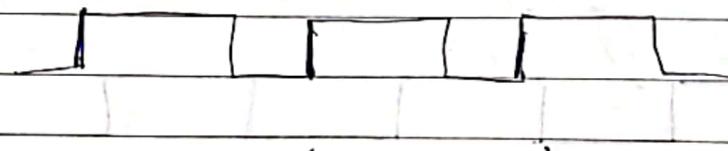
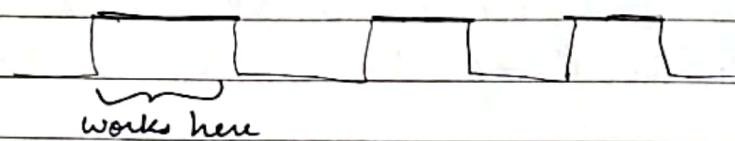


$$f = \frac{1}{\text{Time period}} = \frac{1}{T}$$

In sequential circuits there are many processes going on. The output of one circuit is input of other plus there should be control over these processes as they shouldn't work randomly. So we need clocking to time the circuit. It is called clock because it decides the time of the input.



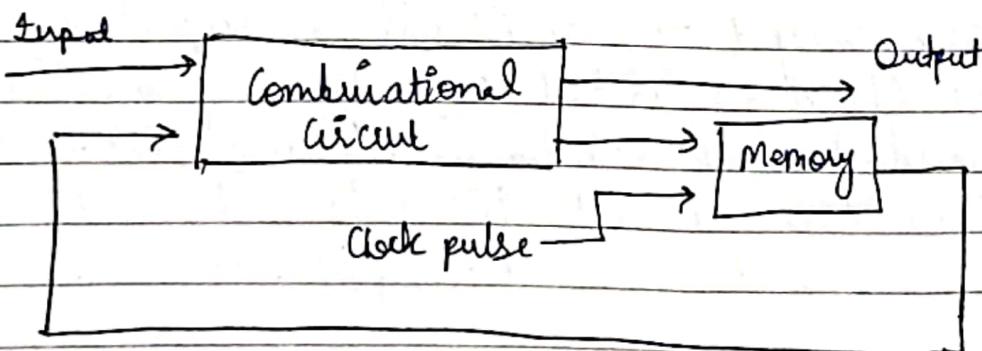
Now this flip flop (seq crkt) will work only when clock is high. Also the flip flop can be designed to function when clock goes low to high or from high to low



Duty cycle: Ratio of the time for which this signal is high to the total time.

$$= \frac{t/2}{t} = 50\% \quad (t \text{ is the total time})$$

TRIGGERING METHODS



The state of the combinational circuit is stored in the memory. Memory is either latch or flip flop.

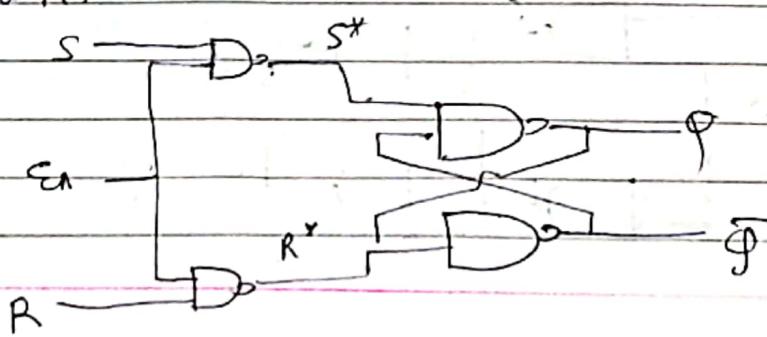
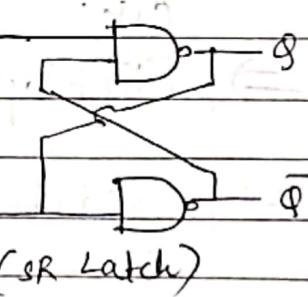
The clock pulse is the control input. In this sequential circuit we have to trigger the memory element. The clock pulse will cause a change in the latch/FF depending on the stored state.

Types of triggering:

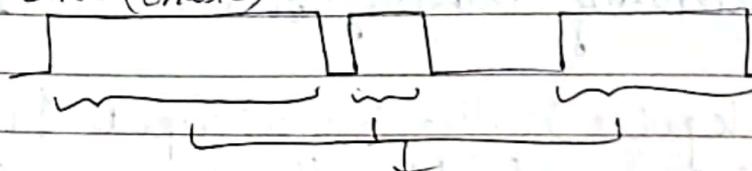
- level triggering
- edge triggering
 - positive edge triggering → memory switches state when clock goes from low to high
 - negative edge triggering → memory switches state when clock goes from high to low

Difference between latch and Flip Flop

- There is no control input given to it, so the inputs S and R can be changed accidentally or intentionally, so this circuit is not very suitable for use, & so we need to use controlled input by which we can decide when to change S and R.



This circuit works whenever E_n is high.
 Eg: E_n : (Enable)



This is a controlled latch - when $E_n = 1$,

$$S^* = \bar{S} + \bar{E}_n$$

$$S^* = \bar{S} \quad (\text{as } \bar{E}_n = 0)$$

$$\text{When } E_n = 0$$

$$S^* = \bar{S} + 1 = 1$$

$$\text{Since, } R^* = 1$$

In SR latch, both as 1 is memory.

This is SR latch with controlled input.

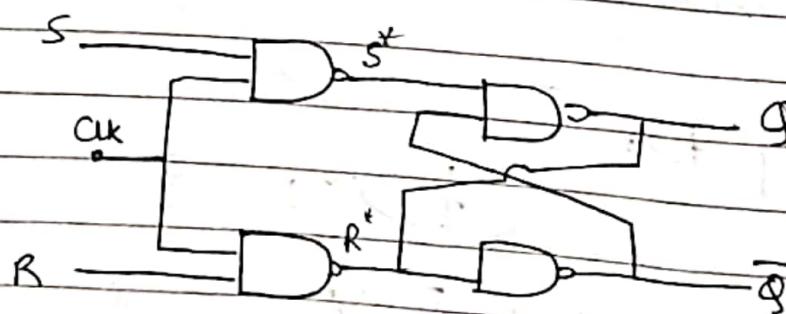
This circuit works as flip flop when we have clock in place of E_n .

Clk:

\Rightarrow FlipFlop is always edge triggered.

\Rightarrow Latch is level triggered

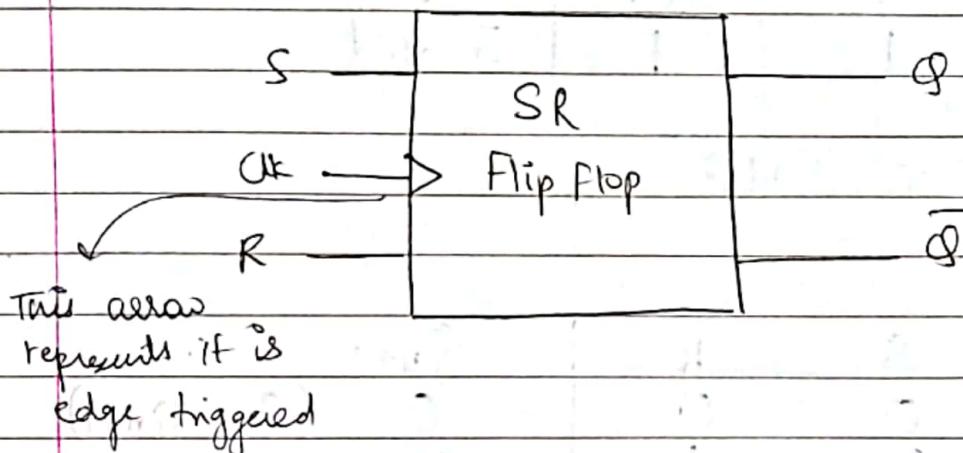
SR FlipFlop



S^*	R^*	Q	\bar{Q}
0	0	Not Used	
0	1	1	0
1	0	0	1
1	1	Memory	

$$S^* = (\overline{S} \cdot \overline{Clk}) = \overline{S} + \overline{Clk}$$

$$R^* = (\overline{R} \cdot \overline{Clk}) = \overline{R} + \overline{Clk}$$



TRUTH TABLE

Clk	S	R	Q	\bar{Q}
0	-	-	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not Used	

When $Clk=0$, $S^* = \overline{S} + 1 = 1$, $R^* = \overline{R} + 1 = 1$

⇒ Always memory

when $Clk=1$, $S^* = \overline{S}$, $R^* = \overline{R}$

Eg $\Rightarrow S=0, R=0 \Rightarrow S^*=1, R^*=1$ (Memory)

$S=1, R=1 \Rightarrow S^*=0, R^*=0$ (Not used)

Same way for other two cases.

Truth table gives characteristic Table
Characteristic table gives excitation table

POPU

Page No.:

Date:

Characteristic table & Excitation table for
SR flipflop:

Truth Table:

Clk	S	R	Q_{n+1}
0	-	-	$Q_n \rightarrow P.S$
1	0	0	Q_n (Memory) [Next state = previous state]
1	0	1	0
1	1	0	1
1	1	1	Invalid

Characteristic Table:

$Clk = 1$	Q_n	S	R	Q_{n+1}
	0	0	0	0 ($Q_n = Q_{n+1}$)
	0	0	1	0
	0	1	0	1
0	1	1	1	Invalid
1	0	0	0	1 ($Q_n = Q_{n+1}$)
1	0	1	1	0
1	1	0	0	1
1	1	1	1	Invalid

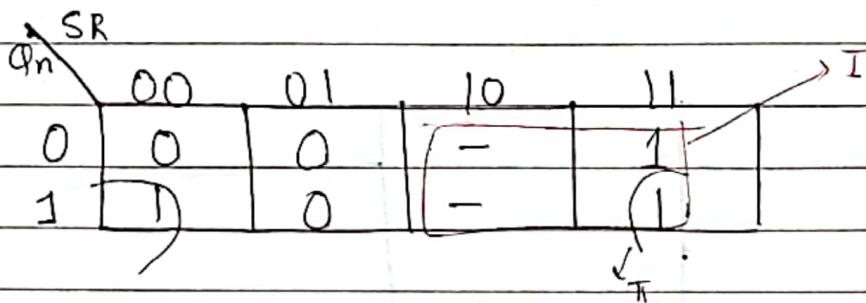
$Clk = 0 \Rightarrow S \rightarrow X \quad R \rightarrow X$

Always Memory

Excitation Table:

Q_n	Q_{n+1}	S	R
0	0	0	- 0/1 (Any of the two)
0	1	1	0
1	0	0	1
1	1	0/1	0

K-Map for $Q_{n+1} \Rightarrow$



$$Q_{n+1} = I + II$$

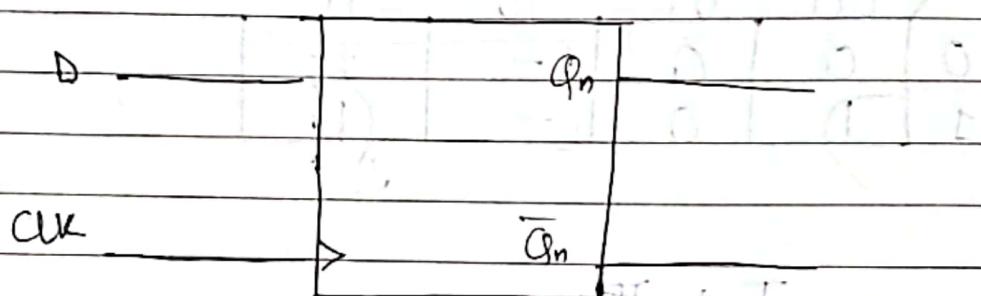
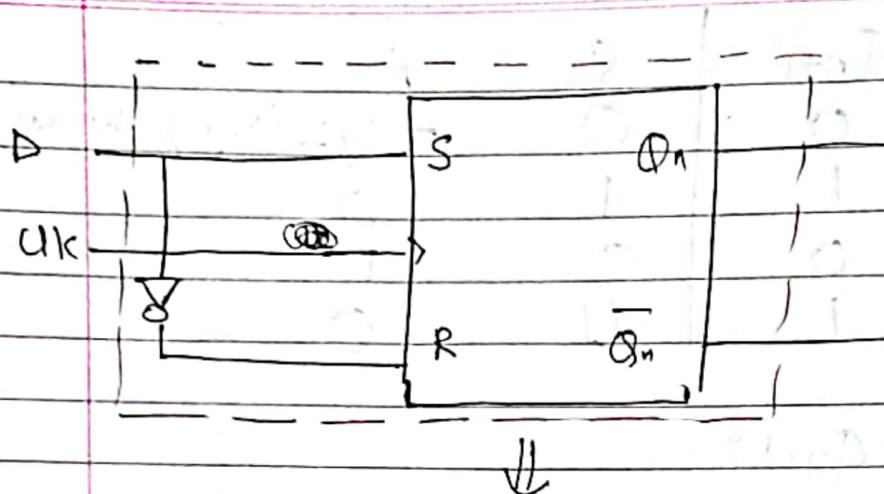
$$Q_{n+1} = S + Q_n R$$

D - FLIP FLOP

T.T for SR flip flop:-

CK	S	R	Q_{n+1}
0	-	-	Q_n memory
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	invalid } \rightarrow but this case S & R are not complement but anyways the case is invalid

So in D flip flop, S & R are complement of each other



TRUTH TABLE

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

$S=0, R=1$
 $S=1, R=0$

Characteristic table and excitation table for
D-FF :

Characteristic Table

$Clk=1$	Q_n	D	Q_{n+1}
	0	0	0
	0	1	1
	1	0	0
	1	1	1

$$Q_{n+1} = D$$

Excitation Table

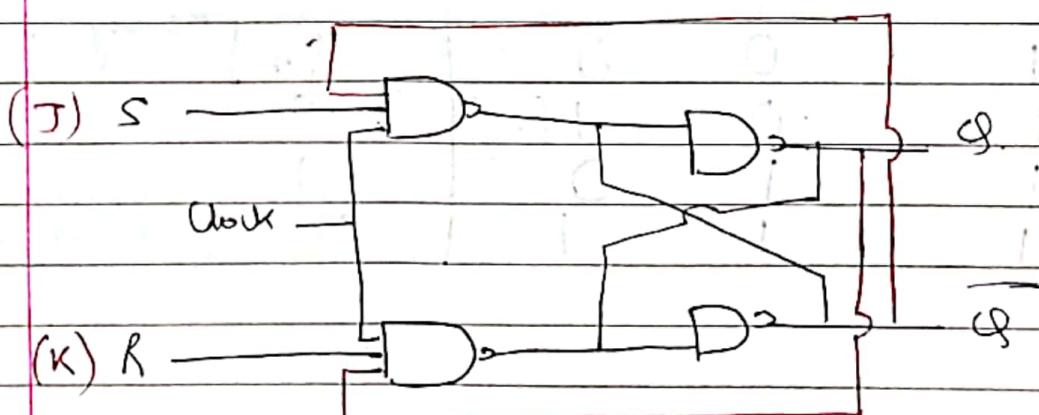
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

JK FlipFlop

TT for SR flipflop \rightarrow

Ck	S	R	Q_{n+1}
0			Q_n (memory)
1	0	0	Q_n (memory)
1	0	1	0
1	1	0	1
1	1	1	Not Used

This last combination where $S=1$ and $R=1$ is a not used state. So JK flipflop uses this state in such a way that it becomes valid.



(First 4 cases of SR and JK are same)

$\boxed{\text{Clk} = 0}$ Memory

$\boxed{\text{Clk} = 1}$

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

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

$$\text{Now, } J=1, K=1$$

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

$$\begin{array}{lll} Q = 0, 1, 0, 1 \dots & \left. \begin{array}{l} \text{Both race} \\ \text{very fast} \end{array} \right\} \\ \bar{Q} = 1, 0, 1, 0 \dots & \end{array}$$

When $Q=0$ then it becomes 1 then 0
then 1 and so on.

Characteristic and Excitation table for
JK flipflop:

TRUTH TABLE

<u>Clk</u>	<u>J</u>	<u>K</u>	<u>Q_{n+1}</u>
0	-	-	Q_n ? memory
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	\bar{Q}_n (toggle)

CHARACTERISTIC TABLE

Q_n	J	K	Q_{n+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

EXCITATION TABLE

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

For J:

Q_n	0	1	
0	0	1	$J = Q_{n+1}$
1	-	-	

For K:

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

For Q_{n+1} :

Q_n	00	01	11	10
0	0 → 0	0 → 1	1 → 1	1 → 0
1	1 → 1	0 → 0	0 → 1	1 → 1

$$Q_{n+1} = \overline{Q_n}J + Q_nK$$

RACE AROUND CONDITION

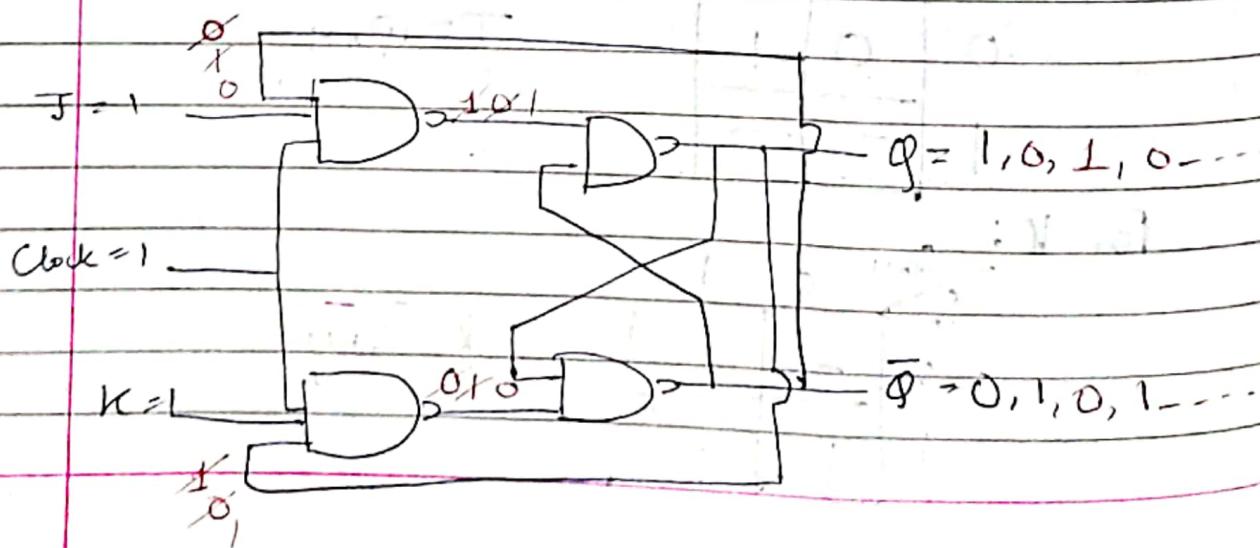
For JK Flipflop

Clock	J	K	Q_{n+1}	\overline{Q}_{n+1}
0	x	x	Q_n	\overline{Q}_n
1	0	0	Q_n	\overline{Q}_n
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

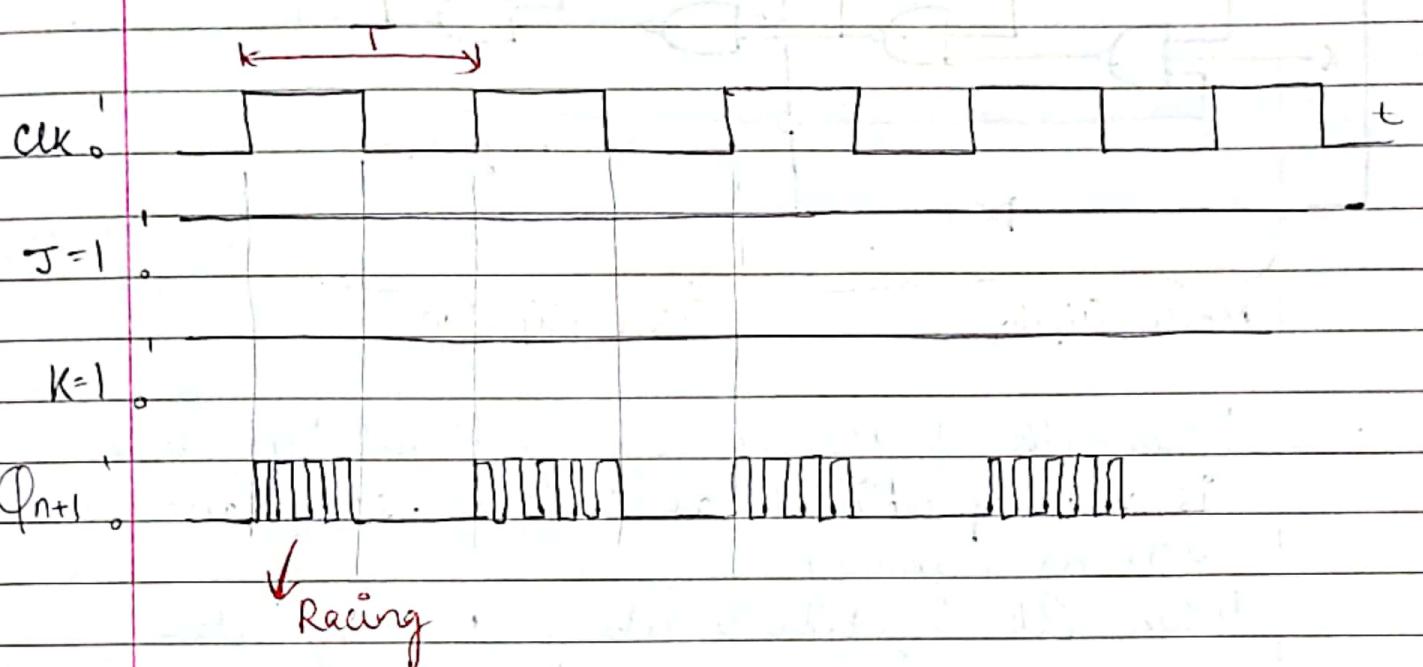
In JK flipflop,

$$\overline{J}=1 \quad \text{Clock}=1 \quad K=1$$

Assuming $Q=1$, $\overline{Q}=0$



- This constant changing of Q and \bar{Q} when $J, K, CLK = 1$ is race around condition
- Its different from toggling. Toggling is controlled while this isn't.



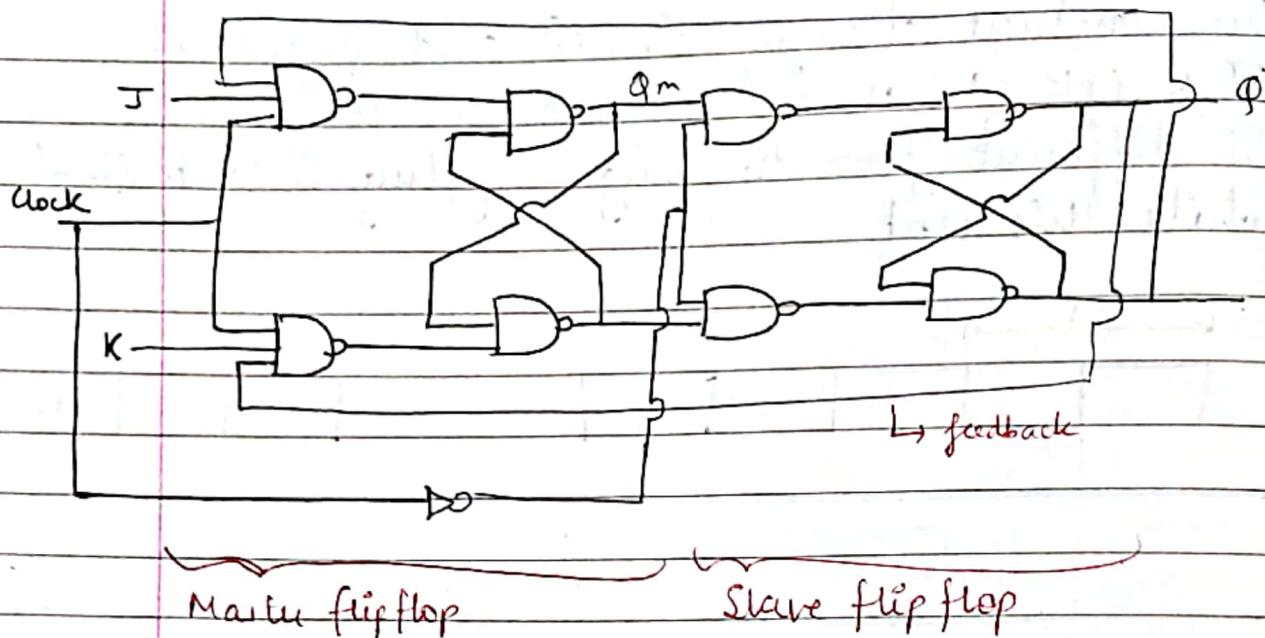
- Half time period of clock $T_1/2 >$ delay of flip flop
as circuit is operational only when clock is high

Conditions to overcome racing:

- $T_1/2 <$ propagation delay of FF (not practically used generally)
- Edge triggering (In this there's not enough time for circuit to race)
- Master Slave

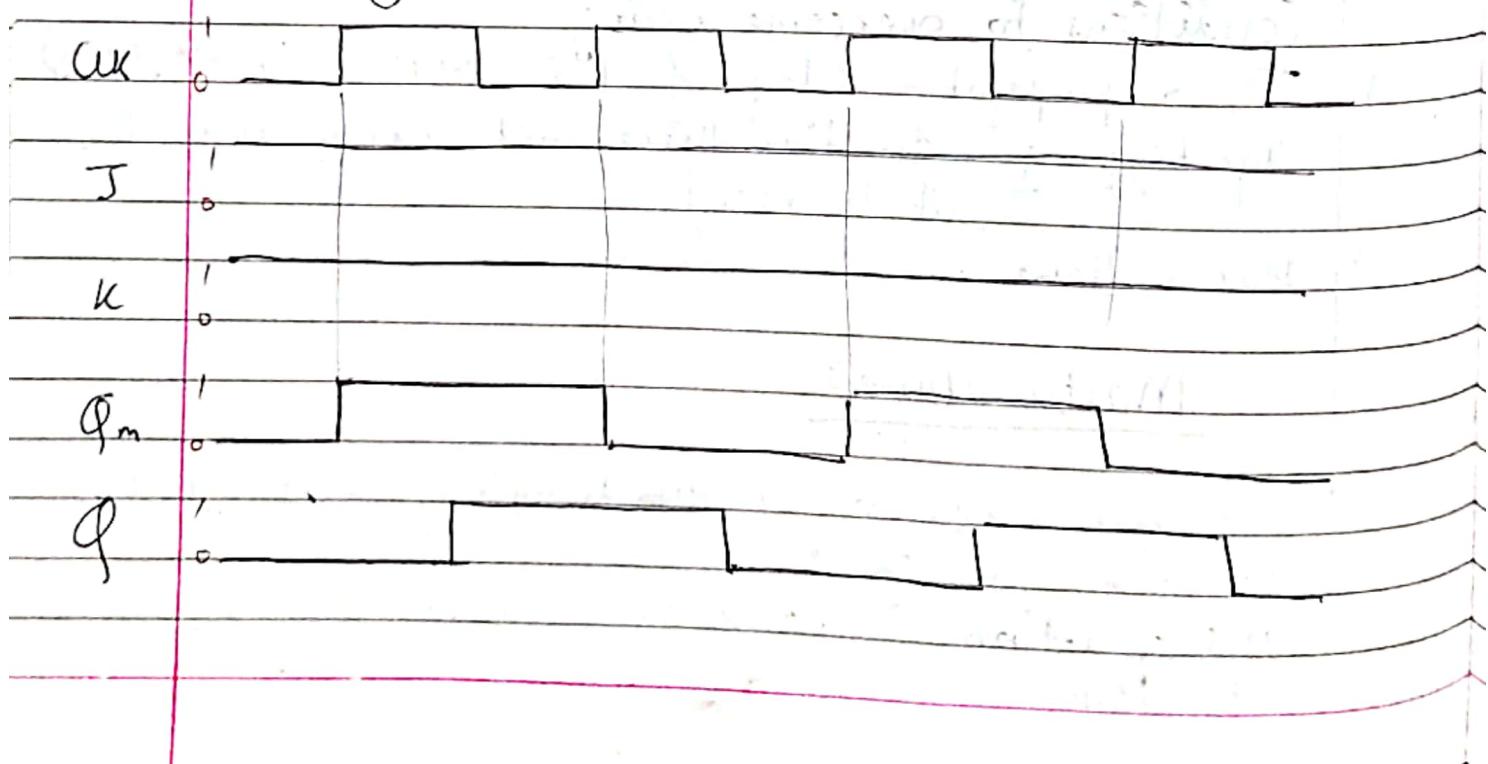
Master Slave:

- We use ways to get ~~toggling~~ out of the racing condition.
- M-S operation is same as negative edge triggered flip flop.



When $\text{Clk} = 1$, Master is operational but slave is not operational as $\text{Clk} = 0$, so slave acts as memory.

When $\text{Clk} = 0$, Master acts as memory, slave gets functions. When output changes, as we have feedback but it has no effect as master is low and hence no effect of racing.



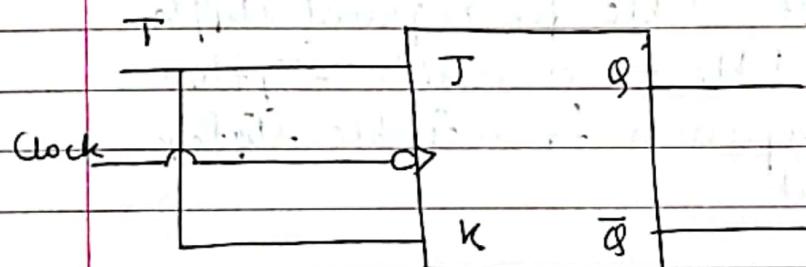
So when $Clk = 1$, $J = 1$, $K = 1$

$Q_{n+1} = \overline{Q_n} \rightarrow \text{toggling}$

Used in the counter

T-FlipFlop

→ If we only want toggling action, then we will use the T-flipflop.
(T stands for Toggle)



TRUTH TABLE:

$Clk = 1$	T	Q_{n+1}
0	X	Q_n (memory)
1	0	Q_n (memory)
1	1	$\overline{Q_n}$ (Toggling)

CHARACTERISTIC TABLE

$Clk = 1$	Q_n	T	Q_{n+1}
0	0	0	0
0	0	1	1
1	0	0	1
1	1	1	0

$$Q_{n+1} = Q_n \oplus T$$

(X=OK)

EXCITATION TABLE~~Ans =)~~

<u>Q</u>	<u>Q_{n+1}</u>	<u>T</u>
0	0	0
0	1	1
1	0	1
1	1	0

Flip flop conversions →

- 1) Identify available and required flip flop
- 2) Make characteristic table for required flip flop.
- 3) Make excitation table for available flip flop
- 4) Write boolean expression for available flip flop
- 5) Draw the circuit

J.K to D conversion:

Char Table for D ff.

<u>Q_n</u>	<u>D</u>	<u>Q_{n+1}</u>
0	0	0
0	1	1
1	0	0
1	1	1

Excit table for JK ff

<u>Q_n</u>	<u>Q_{n+1}</u>	<u>J</u>	<u>K</u>
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

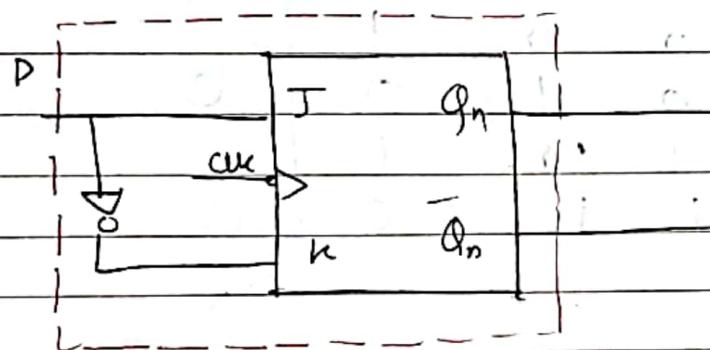
<u>Q_n</u>	<u>D</u>	<u>Q_{n+1}</u>	<u>J</u>	<u>K</u>
0	0	0	0	x
0	1	1	1	x
1	0	0	x	1
1	1	1	x	0

Q_n	D	0	1
0	0	1	(J)
1	x	x	

Q_n	D	0	1
0	x	x	(K)
1	1	0	

$$J = D$$

$$K = \overline{D}$$



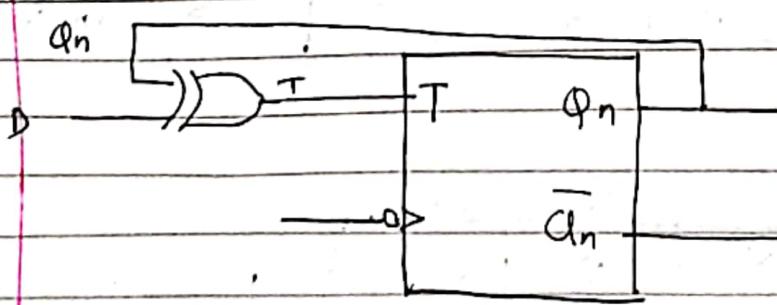
T ff to D ff :

Char table for D ff

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

Excitation table for T ff

$$T = D \text{ xor } Q_n = D \oplus Q_n$$



SR to JK FF:

Char table of J-K

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

Exit table of SR

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

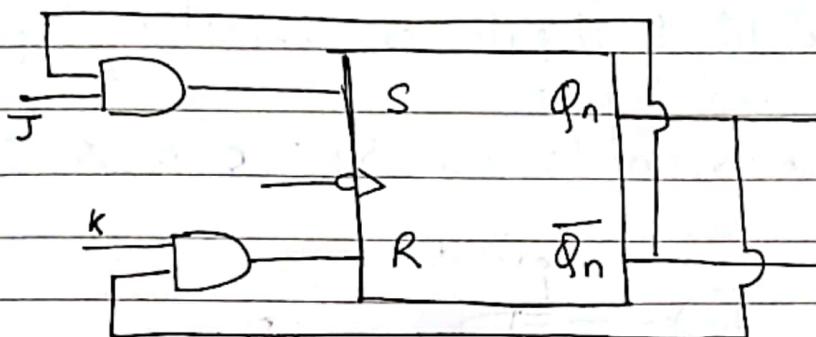
Q_n	JK	00	01	11	10	01	
0	0	0	0	(1)	(1)		
1	x	0	0	x			

$$S = \overline{Q_n} \cdot J$$

For S:

Q_n	J	K	00	01	10	11	
0	x	x	0	0	0	0	
1	0	0	(1)	(1)	0	0	

$$R = Q_n R$$



SR to T ff:

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

a_n	T	0	1
0	0	0	1
1	x	0	0

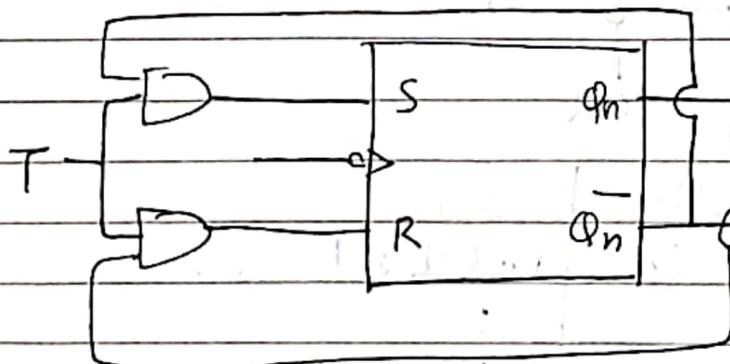
Q_n	T	0	1
0	0	x	0
1	0	0	1

(S)

(R)

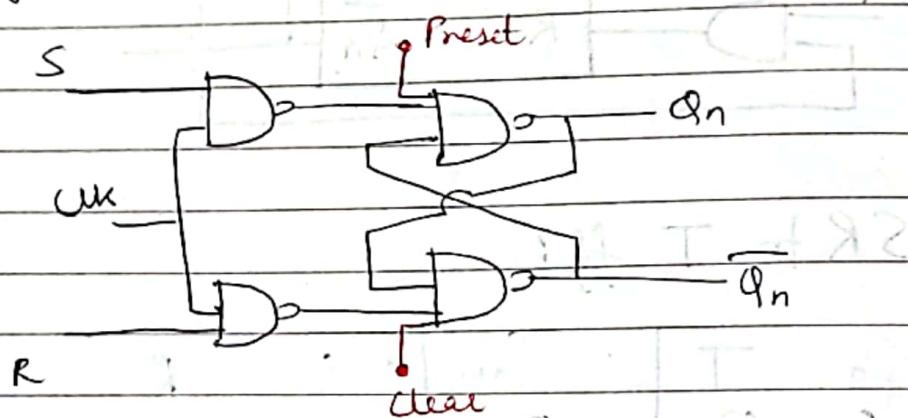
$$S = \bar{Q}_n T$$

$$R = Q_n T$$



RESET AND CLEAR INPUTS

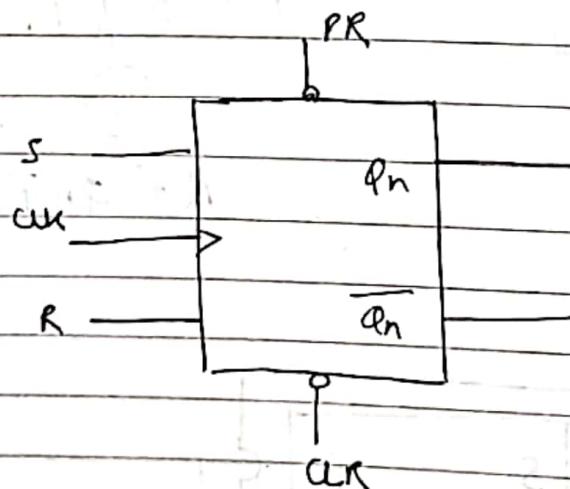
- » They are the direct inputs or overriding inputs or asynchronous inputs.
- » The synchronous inputs are S, R, J, K, D, & T



$$\text{Preset} = 0 \Rightarrow Q_n = 1$$

$$\text{Clear} = 0 \Rightarrow Q_n = 0 \text{ because } \overline{Q_n} = 1$$

whatever be the value of clock and synchronous inputs-



Preset	Clear	Q _n
0	0	Not Used
0	1	1
1	0	0
1	1	FF will perform normally

Diff b/w Synch. and Asynch. Sequential Circuits:

Synchronous Seq. Ckt

1. These circuits are easy to design.
2. A clocked flip flop acts as memory element.
3. They are slower.
4. The status of memory element is effected only at active edge of clock, if input is changed.

Asynchronous Seq. Ckt

- These are difficult to design.
- An unlocked flip flop or time delay element is used as memory element.
- Faster as clock is not present.
- The status of memory element will change at any time as soon as input is changed.