combinational circuits میں کسی بھی وقت جو output ہوتی ہے وہ اس وقت جو circuits کی inputs ہوتی ہیں
صرف اسی پہ depend کرتی ہے

The current output at any time doesn't depend upon previous outputs

Chapter # 06

Sequential Logic

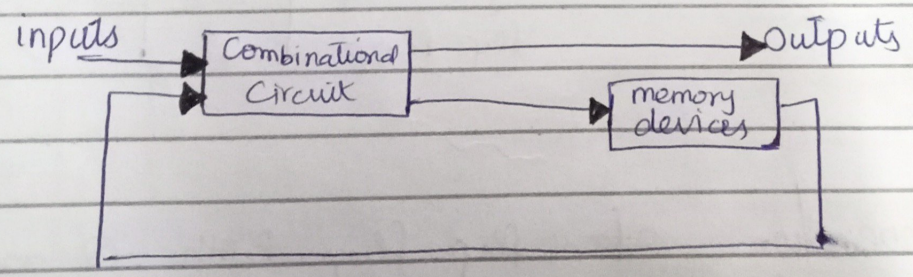
The circuits discussed so far are combinational circuits where the outputs are dependent upon the external inputs only.

However, most of the circuits include memory elements that lead to the discussion of sequential circuits. The memory elements are the circuits capable to store binary information in them

Sequential logic میں کسی بھی وقت جو output ہوتی ہے وہ نہ صرف current inputs پہ depend کرتی ہے بلکہ Previous outputs پہ بھی depend کرتی ہے

So, we have to store previous outputs
So, sequential circuits include memory. (memory elements)

★ Flip flop are basic storage device which store single bit



It is evident from block diagram of sequential circuits that external outputs are a function not only external inputs but also that of previous states (previous outputs). The binary information stored in particular memory element is called state of the memory element

used to store previous outputs

Flip Flops (Memory Elements)

A flip flop is a binary storage device that can store one bit of data in it. (0, 1)

A flip flop has 2 outputs namely Q (representing normal value of bit stored in it) and Q' (representing complement of value stored in it

* A flip flop also has two states (SET, RESET)
* The state of flip flop is determined from Q output of flip flop.

Q = 1 , Q' = 0 SET state

Q = 0 , Q' = 1 RESET

* A flip flop can maintain its state indefinitely until we change input data.

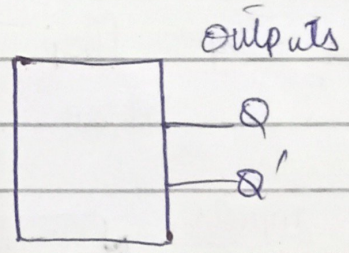
SR flip flop

D flip flop

JK flip flop

T flip flop

Inputs depend upon types of flip flop



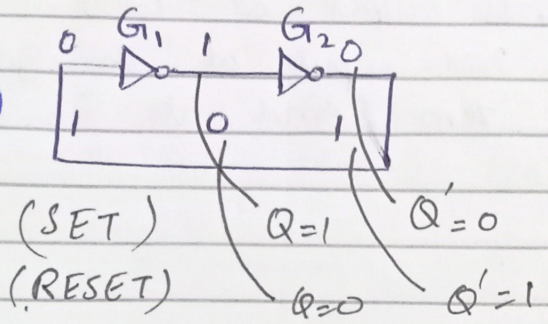
For normal operation of flip-flop, both Q and Q' must be complement to each other.

Basic

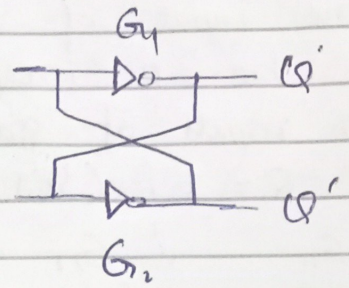
RS Flip Flop (using NOR gates)

A basic flip-flop can be constructed using either two NOR gates or two NAND gates.

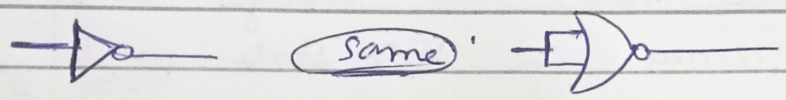
1



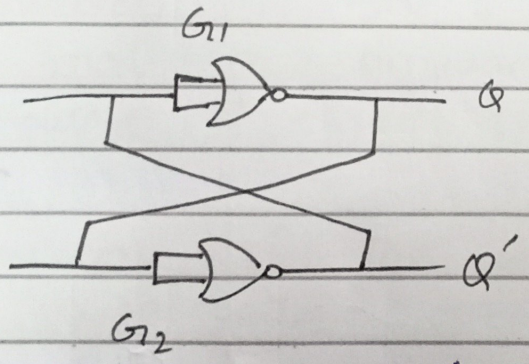
2



★ Single input NOR gate work as NOT gate.

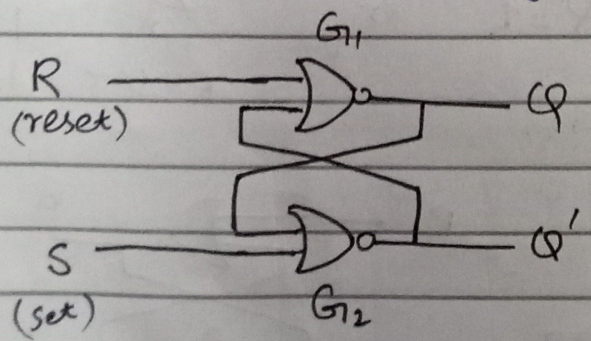


3



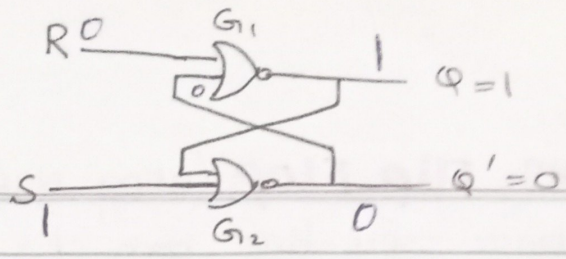
1,2,3 → same

For external control to make it SET or RESET we convert above single input NOR gate into double inputs NOR gate



output of NOR gate is 0 when any input is 1.

output is 1 when all its inputs are 0



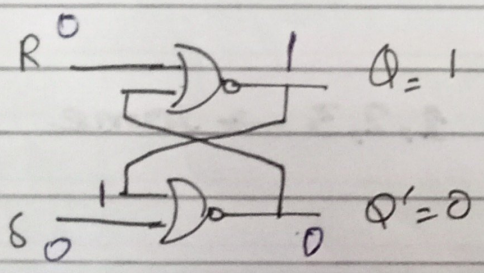
Case 1 G_1: $R=0$ G_2: $S=1$

SET state.

since one input of G_2 is 1, its output Q' will definitely be 0 irrespective of 2nd input at its gate. so both inputs of gate 1 is zero and its output $Q=1$. (SET state)

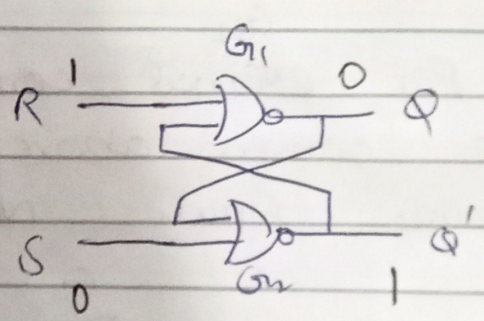
Case 2 G_1: $R=0$ G_2: $S=0$

In this situation, both inputs of G_1 will be zero so Q will be 1, making one input if $G_2 = 1$ so $Q' = 0$. So, there's no change in state and it will remain in SET state.



Case 3 $R=1$ $S=0$
G_1 G_2

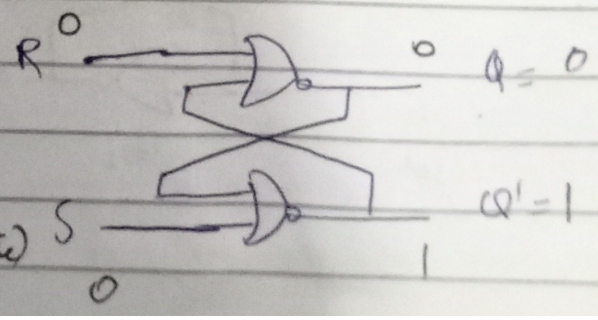
Since one input of $G_1 = 1$ its output will definitely be 0 i.r. of 2nd input
Both inputs of $G_2 = 0$ so $Q' = 1$
RESET state.



RESET state.

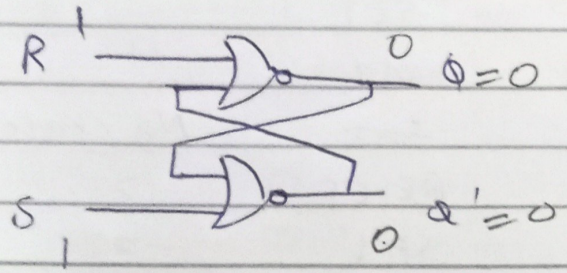
Case 4 $R=0$, $S=0$
G_1 G_2

G_2 both inputs $= 0$ so $Q' = 1$
making one input if $G_1 = 1$, $Q = 0$
No change in state, (RESET state)



Case 5 G_1: $R = 1$ G_2: $S = 1$

In this situation one of input of both gates = 1 so $Q, Q' = 0$, violating the fact that both Q and Q' must be complement to each other. So this state become avoidable / undefined.



However, if case 5 occurs and then S and R both made 0 simultaneously, a problem will occur. The output of flip flop is not determined.

★ Any one of the two NOR gates acts faster than other NOR gate.

It can be either SET or RESET.

It depends upon which input remain 1 longer before transition to zero.

Summary:

R=0,	S=0	NO change
R=1,	S=0	RESET
R=0,	S=1	SET
R=1,	S=1	Avoidable state

SR Latch

NAND gate → Active low

Q_t , Q_{t+1} ⇒ Q

Present state — next state

NOR Gate → SR Latch
Active high

Truth table / characteristic table

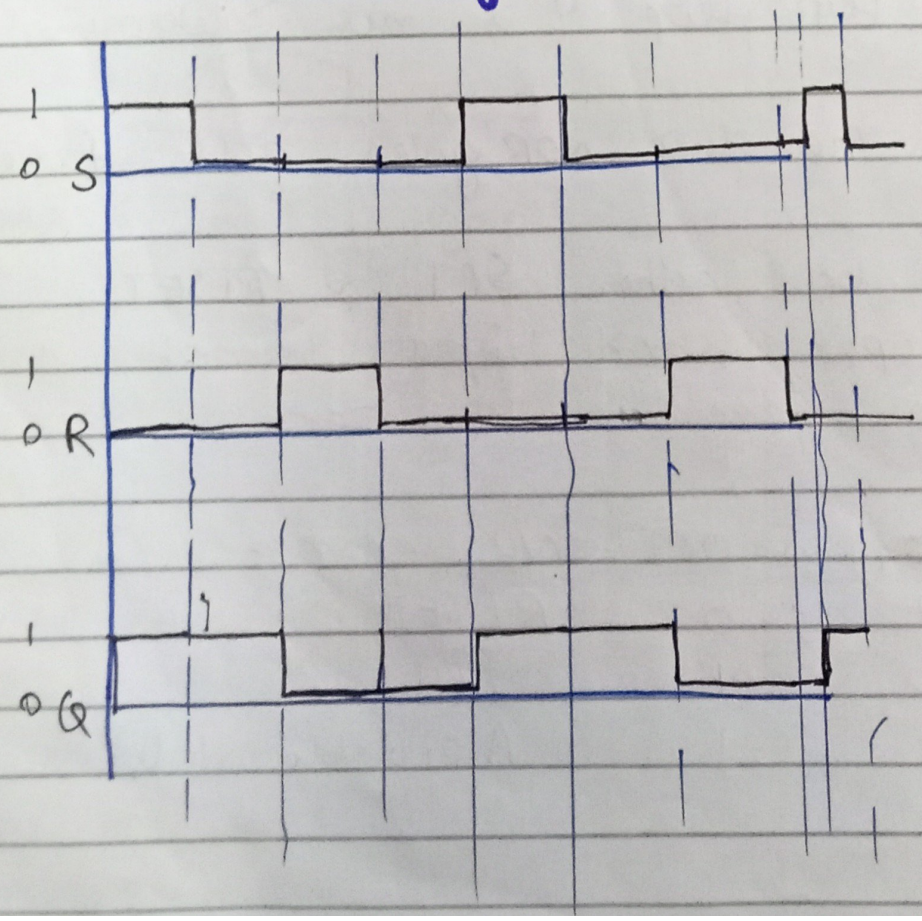
state before applying any combination

state after applying any combination

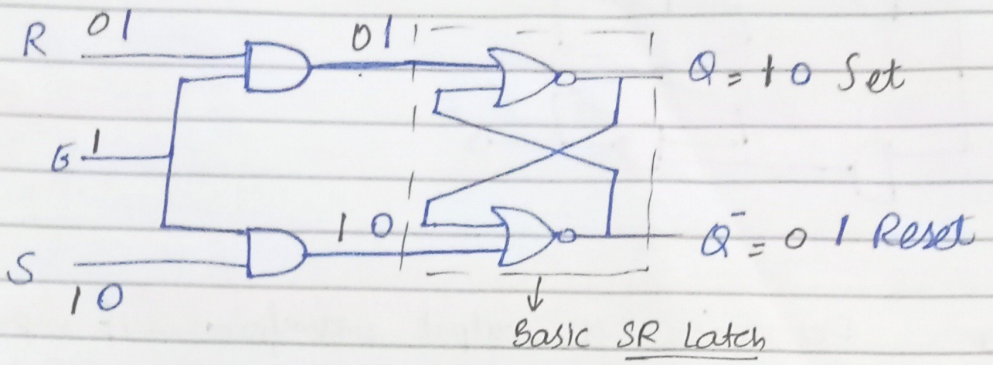
Q_t	S	R	Q_{t+1}	State	
0	0	0	0	~~RESET~~	No change
RESET 0	0	1	0	RESET	
~~SET~~ 0	1	0	1	SET	
0	1	1	X	Avoidable	
1	0	0	1	~~SET~~	No change
SET 1	0	1	0	RESET	
1	1	0	1	SET	
1	1	1	x	Avoidable	

subscript

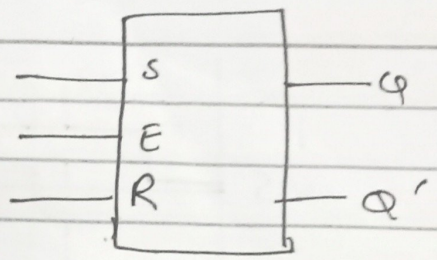
SR Latch using NOR gates



Gated SR Latch (using Enable input) using NOR gate



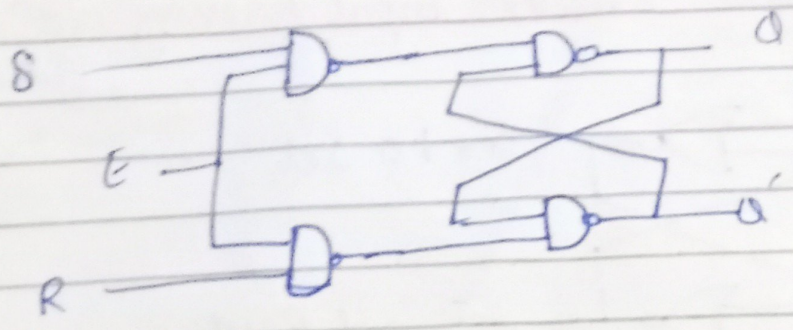
E	S	R	Q_{t+1}	State
0	0	0	0 Q_t	
0	0	1	0 Q_t	
0	1	0	0 Q_t	
0	1	1	0 Q_t	
1	0	0	0	
1	0	1	0	
1	1	0	1	
1	1	1	X	Avoidable



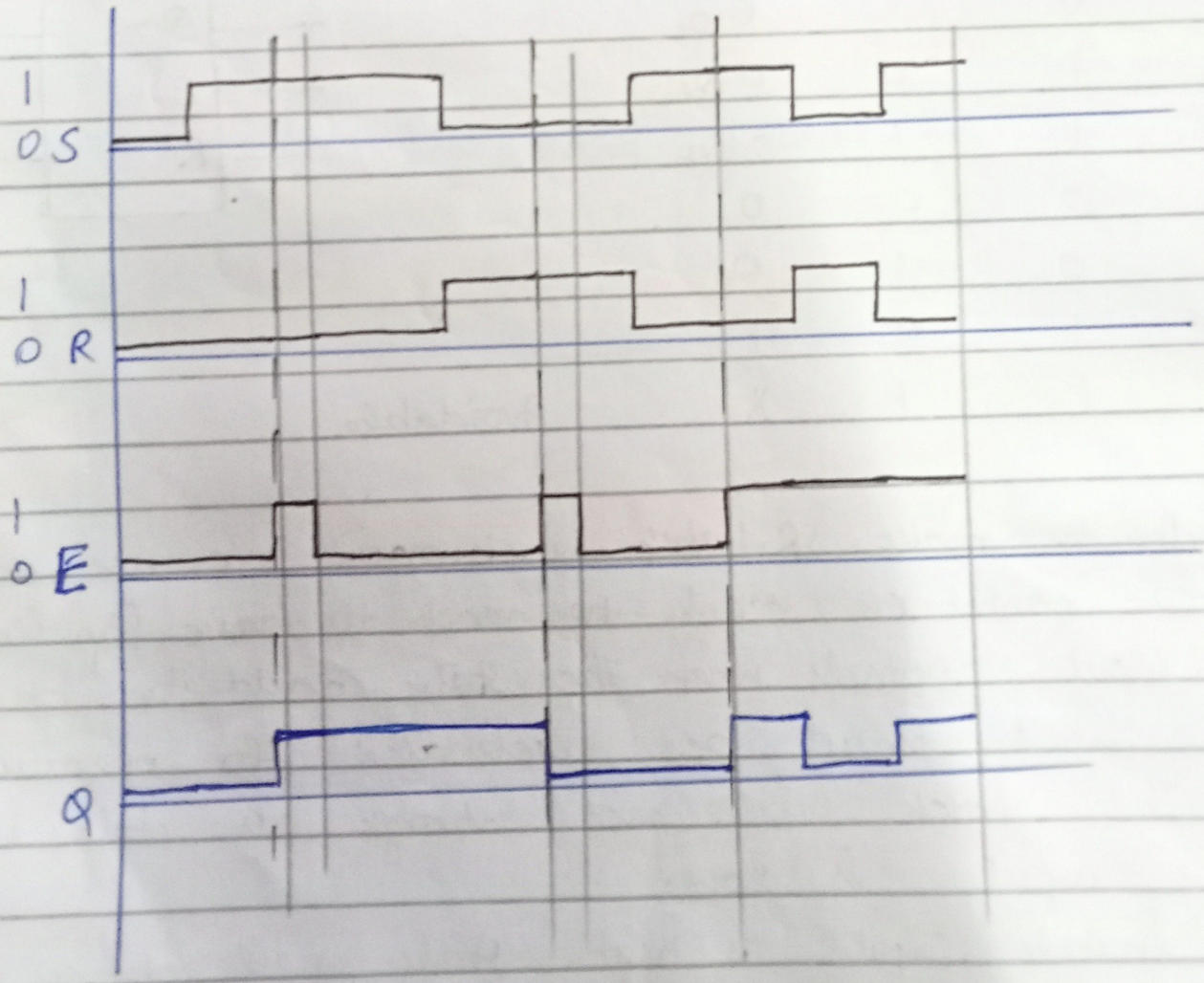
In order to make SR Latch synchronous, we use enable input to control SR inputs to reach to basic flip flop (latch). Enable input controls when the state should be changed. Enable input must be activated for operation of Latch. The latch will not change its state when enable input is low.

When enable input is high, state will change based upon R and S inputs the same way as discussed previously.

using NAND gate:



<u>Question</u> Determine Q output waveform if inputs show in fig are applied to gated S-R latch that is initially RESET

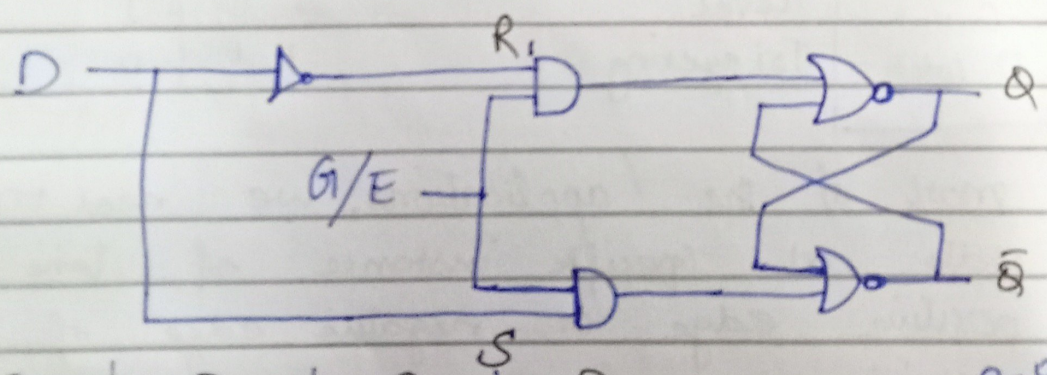


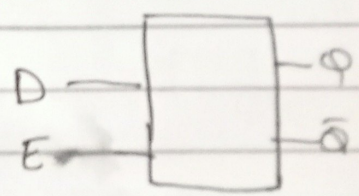
Gated D Latch

* SR latch is in The prohibited state when both R=S=1.
* D (data) input) latch is a single input version of SR latch.
* NOR gates #1 and #2 from basic flop and AND gates a and b are used for the control input.
* D input goes directly to S input (gate #b) and its complement is applied at input (gate #a)
* We get rid of undefined state of SR latch in D latch.

No change Avoidable] Both states removed

only SET or RESET





En	D	Q_t	Q_{t+1}
0	0	0	$0 = Q_t$
0	0	1	$1 = Q_t$
0	1	0	$0 = Q_t$
0	1	1	$1 = Q_t$
1	0	0	$0 = D$
1	0	1	$0 = D$
1	1	0	$1 = D$
1	1	1	$1 = D$

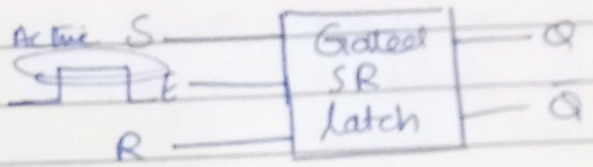
R=0, S=1
D=1

D=0
R=1, S=0

Q=1 SET
Q=0 RESET

Clocked SR Flip Flop

Latches with enable inputs can change /update their state at any time when enable input is active



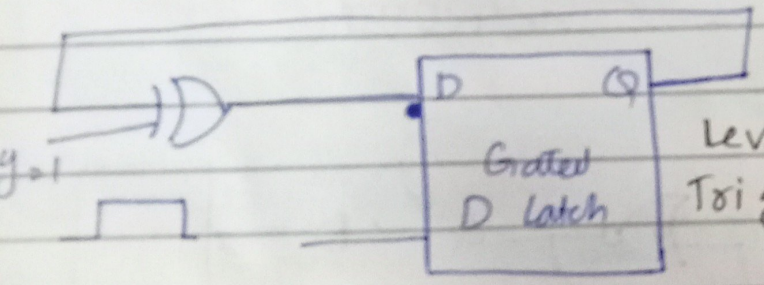
Two Types

Asynchronous
Memory elements are affected the moment input changes
e.g Basic SR flipflop

Synchronous
Memory element are affected when En = 1
e.g Gated SR Latches

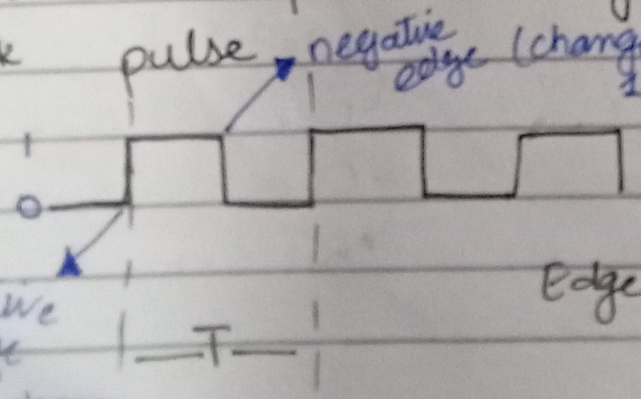
Exclusive OR

X	Y	o/p
0	0	0
0	1	1
1	0	1
1	1	0

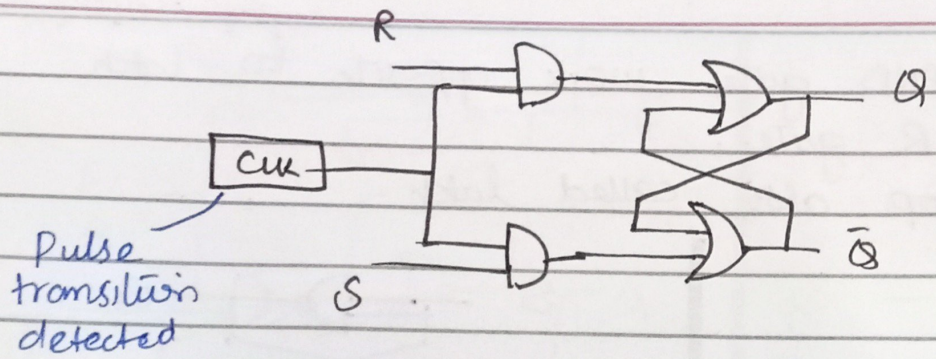


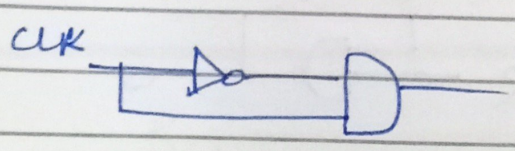
Level Triggering

However, in most of the applications, we need to update the state at specific instance of time either at positive edge or negative edge of clock pulse



Edge triggered flip flops





Q_t	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

	$\bar{S}\bar{R}$	$\bar{S}R$	SR	$S\bar{R}$
Q_t			X	1
Q	1		X	1

$$Q_{t+1} = S + Q_t \bar{R}$$

Latch using NAND gate work opposite to latch using NOR gate.

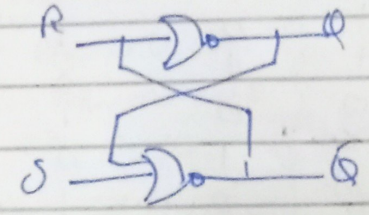
Basic SR flip flop also called latch.

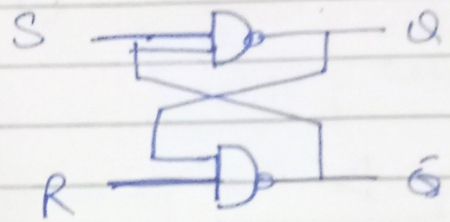
R=1, S=0 Reset

R=0, S=1 Set

R=0, S=0 No change

R=1, S=1 Avoidable





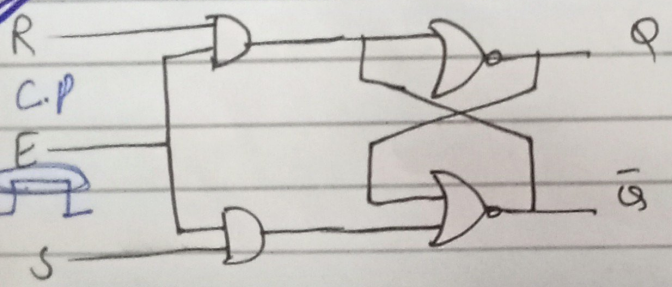
S=0, R=1 Set

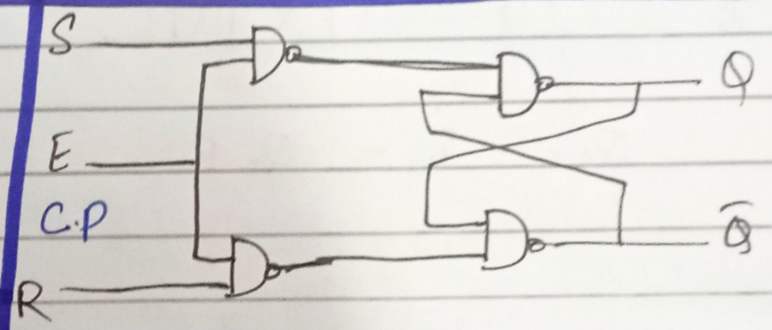
S=1, R=0 Reset

R=1, S=1 No change

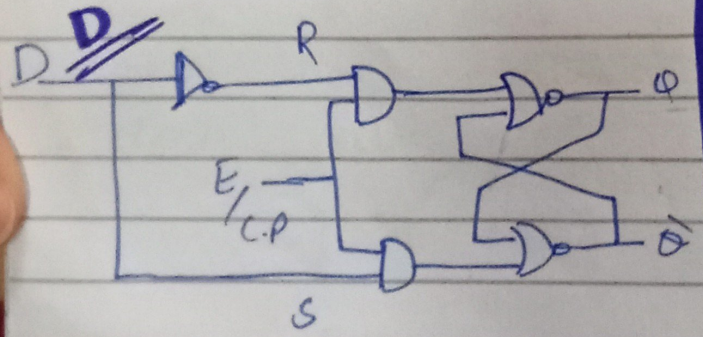
R=0, S=0 Avoidable

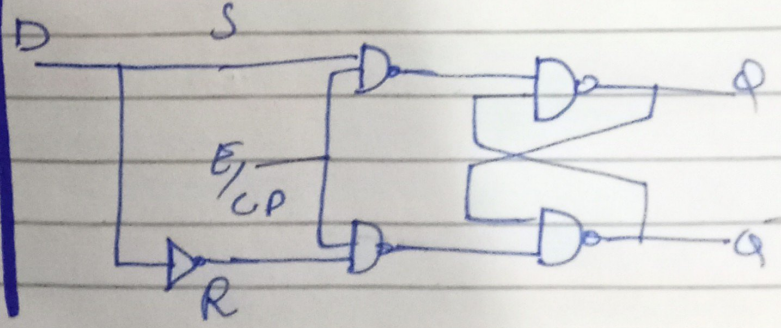
RS Gated RS latch



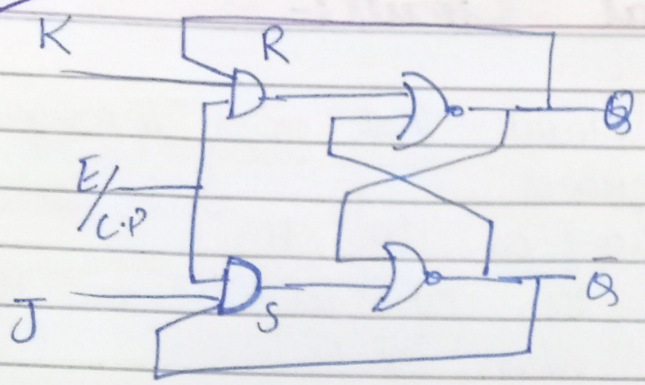


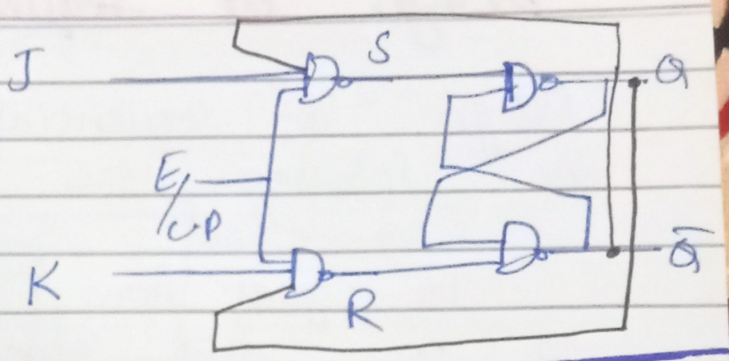
D



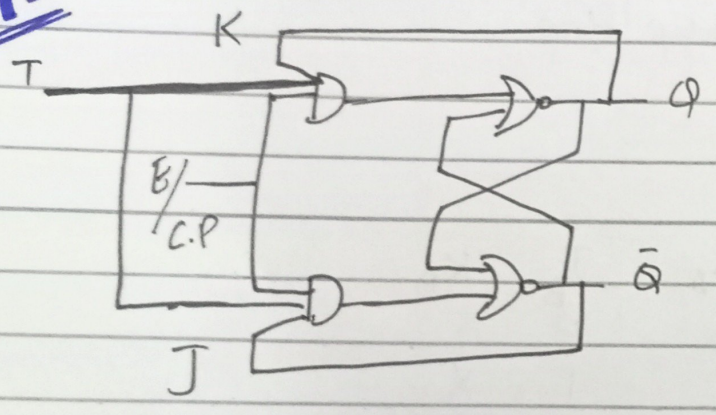


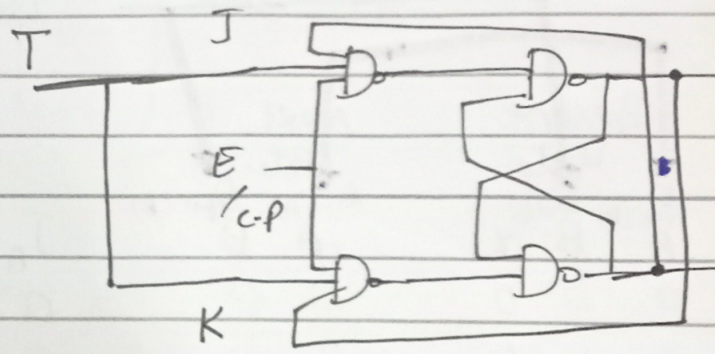
JK





T.





L.Q

Excitation Table

Q_t	Q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

L.Q

① using Rs flip flop
② Down counter
③ when $x = 0$, change
$x = 1$, No change

Design of Sequential Circuit:-

Design a sequential circuit that goes Through the following state sequence.

00 01 10 11

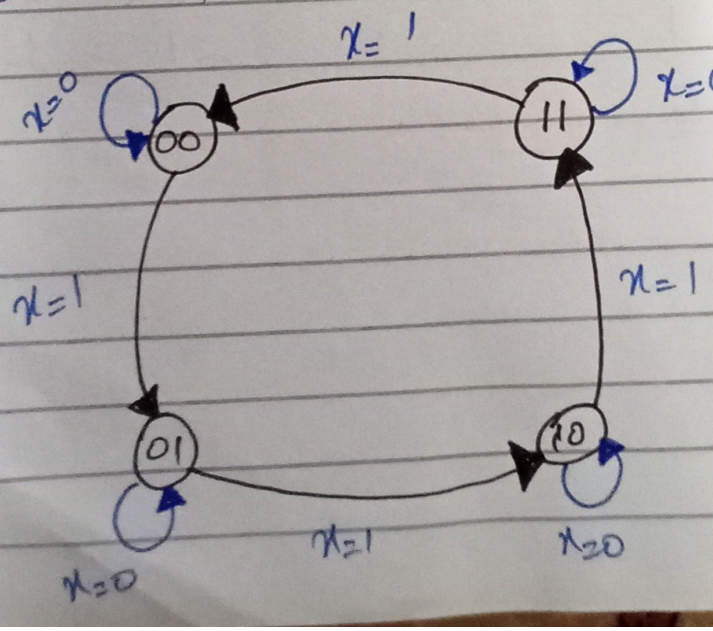
It goes from 00 to 01 to 10 to 11 to 00 and repeat when $x = 1$

It doesn't change its state when $x = 0$

use JK flip flop.

Present state $Q_t(A)$ A	$Q_t(B)$ B	X	Next state $Q_{t+1}(A)$ A	$Q_{t+1}(B)$ B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	0	1	0	X	X	0
0	1	1	1	0	1	X	X	1
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

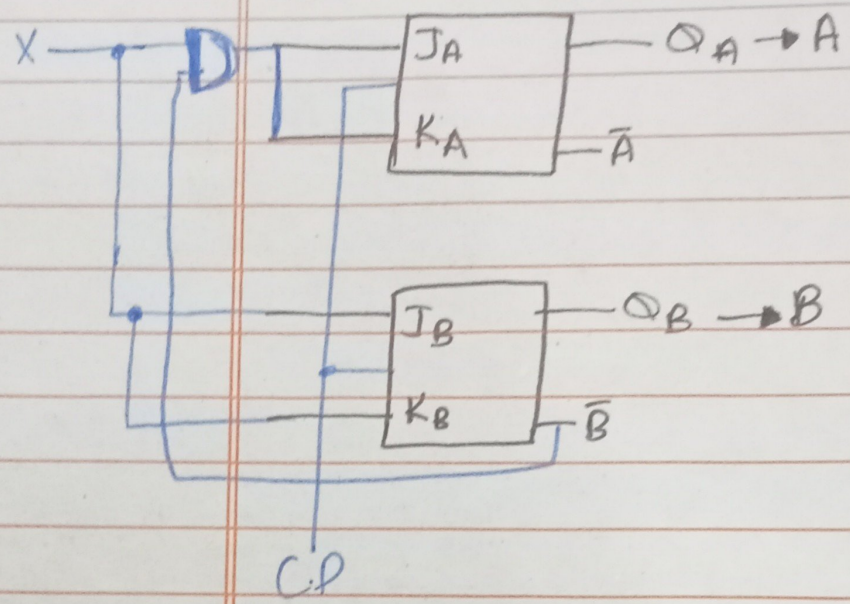
Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



State Diagram

NOTES Subject:

SR. NO.	DATE	TOPIC
① State Diagram ② State Table		2-bit Counter



For J_A

	$\bar{B}\bar{X}$	$\bar{B}X$	BX	$B\bar{X}$
$\bar{A}$			1	
A	X	X	X	X

$J_A = BX$

For K_A

X	X	X	X
		1	

$K_B = BX$

For J_B

	1	X	X
	1	X	X

$J_B = X$

For K_B $K_B = X$

RIZVI PAPER PRODUCTS