

I don't  
 depend just  
 on present  
 i/p but  
 also  
 after some  
 previous  
 history/delay  
 As soon as you give i/p, you get an o/p  
 Sequential logic → Comb logic + Memory element **papergrid**  
 UNIT - 3

19.9.16

## SEQUENTIAL LOGIC CIRCUITS

Synchronous sequential logic → Master clock controls o/p to give o/p only at a particular timing stage

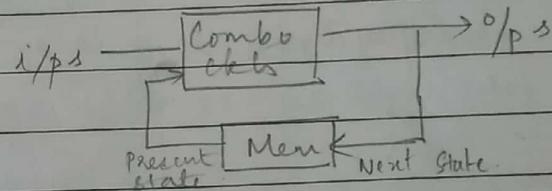
Asynchronous → Depends. You control. Not controlled by a master clock

only diff in syllabus

Combinational  
Logic  
Circuits

Sequential Logic Circuits

i/p → **Combo ckt** → o/p      synchronous      asynchronous  
seq. Ckts      seq. Ckts.



Basic Memory element in sequential circuit is a flip-flop. Called bistable elements

Registers, counters → Use flip flops.

Bistable → Flip flop can remain in one of two states indefinitely until triggered

Storing 1 in flip flop → Called set state (or preset?)  
 " " " " → Reset or clear

Cross-coupling of inverters → o/p of one given to i/p of 2<sup>nd</sup>

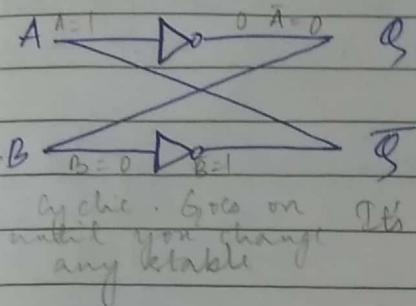
Once you switch on (give 1 or 0) it goes on indefinitely

e.g. if you give 1 to 1<sup>st</sup>. o/p of 1<sup>st</sup> → 0 ⇒ i/p of 2<sup>nd</sup> → 0 ⇒ o/p of 2<sup>nd</sup> → 0

This continues so we use NOR & NAND for better control

Flip flops → Gated latches?  
 → latches

: NOT only has  
one i/p.



No provision to give desired ips.

As soon as you switched on,  
something happened.

Cyclic. Goes on until you change any lable. It's in a deadlock

### BASIC BISTABLE ELEMENT

A circuit which can indefinitely stay in state / content 0 or 1 state is called a Basic Bistable Element figure here.

Let us assume that when power is switched on to the circuit, input A is at 1,  $\therefore \bar{A} = 0$ ,  $\therefore Q = 0$ . This implies  $B = 0$ ,  $\bar{B} = 1$  and  $\bar{Q} = 1$

So  $Q = 0$ ,  $\bar{Q} = 1$   $Q, \bar{Q} \rightarrow \text{op of each inverter} ???$

The circuit continues in this state until the power to the circuit is switched off.

This is one of the two stable states

$$\begin{aligned} \text{When } A = 0, \bar{A} = 1 &\Rightarrow Q = 1 \\ &\Rightarrow B = 1, \bar{B} = 0 \Rightarrow \bar{Q} = 0 \end{aligned} \quad \left. \begin{array}{l} \text{2nd} \\ \text{stable} \\ \text{state} \end{array} \right.$$

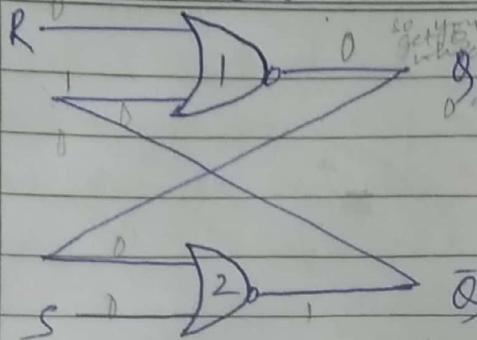
$\rightarrow$  1 or 0 stored in BBE is stable element.

It's said to be 1 state when it stores a 1  $\Rightarrow Q = 1$   
and in 0 state <sup>or reset state</sup> when it stores a 0 ( $Q = 0$ )

But actually there's a transient metastable state  $\rightarrow$  Not a valid stable state  
(when it transitions from 0 to 1 or 1 to 0 it can go into metastable state)

## SR Latch

Circuit



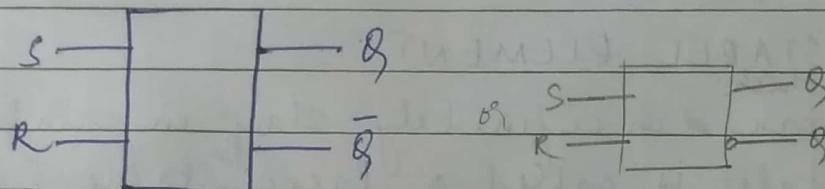
Here we have the provision for giving i/p. We have to assume a value for  $Q$ .

Assuming  $\rightarrow$  feedback.

$\rightarrow$  coz we didn't know in which state it was previously

So now we have control. We use  $R, S$  as i/p

Symbol



Assume  $Q = 0$

$R=0, S=0$        $S, R$  can have 4 combos

$$Q^+ = 0$$

A	B	Y
0	0	1

Assume  $Q = 1, R = 0, S = 0$

$$Q^+ = 1$$

0	1	0
1	0	0

TT for  
NOR gate

## FUNCTION TABLE

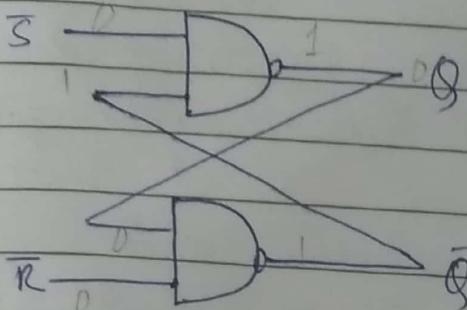
Set Reset

S	R	Q	$Q^+$	Remark
0	0	0	0	Same as $Q$ . Means it retains the previous state of flip flop
0	0	1	1	No change State of flip flops is determined by $Q$ value
0	1	0	0	Same Reset $\rightarrow S=0, R=1$ means it's in reset state so irrespective of $S, Q$ will be 0 (reset)
0	1	1	0	Change
$Q=1$ so creepin of $Q$	1	0	0	Change Set
	1	0	1	Same
1	1	0	0*	Forbidden
1	1	1	0*	Forbidden

$\rightarrow$  we get  $Q = \bar{Q} = 1$  so also forbidden

$$0^* = Q = \bar{Q} = 0$$

Using NAND gates  
SR Latch



0	0	1
1	0	1
0	1	1
1	1	0
1	1	0

S	R	Q	Q <sup>+</sup>	Remark
0	0	0	1*	Forbidden $Q = \bar{Q} = 1$ $\bar{S} = 0, \bar{R} = 0 \Rightarrow S = 1, R = 1$
0	0	1	1*	State $Q = \bar{Q} = 1$

when it is  
in set state  
take only  $Q=1$

Reset state  
check)

take only  
 $Q=0$

coz even

0 10 will

be forbidden

otherwise

if you

check for  $Q=0$

when  $S=R=0, 1$

S	R	Q	Q <sup>+</sup>	Set
0	1	0	1	State
0	1	1	1	State

S	R	Q	Q <sup>+</sup>	Reset
1	0	1	0	State

S	R	Q	Q <sup>+</sup>	Retains
1	1	0	0	State

S	R	Q	Q <sup>+</sup>	Previous state
1	1	1	0	$\rightarrow \bar{S}=1, \bar{R}=1$

$\bar{S}=0 \Rightarrow S=1$

$\bar{S}=0 \Rightarrow S=1$

so  $Q^+$  is 1

$\Rightarrow S=0, R=0$  so it retains

previous state just

like SR latch

## SR Latch Expression

$Q^+$	$R\bar{Q}$	$RQ$	$R\bar{Q}$	
$S$	0	1	0	D
$\bar{S}$	1	1	0X	0X

$$Q^+ = SR + R\bar{Q}$$

$\times \rightarrow$  coz you don't know the intermediate change i.e.  
what's happening

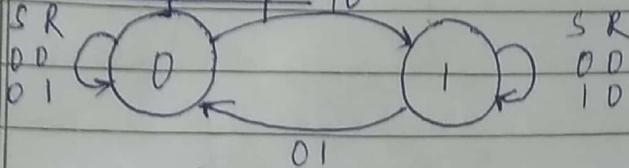
\* Characteristic Equation :

$$Q^+ = S + \bar{R}Q$$

→ Character of flip flop is analysed  
by this equation

## STATE TRANSITION DIAGRAM

SR Flip flop : 10



For which i/p's it i) stays in reset state

ii) transitions from reset to set

iii) stays in set state itself

iv) transits from 1 to 0

Means

Check when  
it already  
remains 10

For 00 it stays in reset state

01

Already 0 changes  
when it changes

When you change it to 10 it transitions from 0 to 1

Then for 00 it stays in the set state

10

When you change it to 01, it transitions from 1 to 0

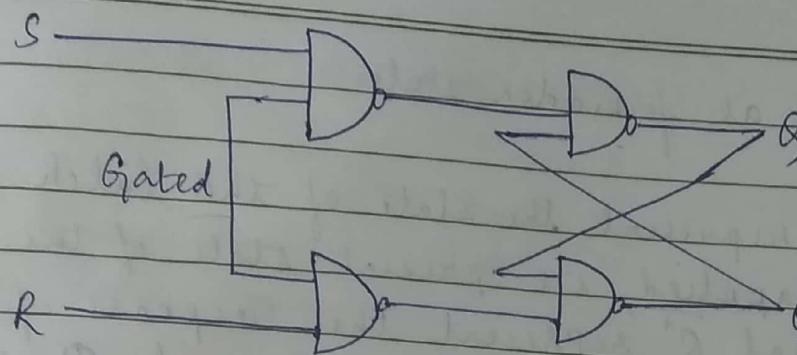
## EXCITATION TABLE

$Q$	$Q^+$	S	R	
0	0	0	X	0X means for 00 & 01 it remains in reset state
0	1	1	0	
1	0	0	1	→ Transition from 1 to 0 for 01
1	1	X	0	→ For 10 and 00 remains in set state

## GATED SR LATCH

Like an enable pin

Using 4 gates to accommodate one more i/p → i.e. clock/gate/control/enable pin



or you can use  
Not gates by  
interchanging S & R

Enable	S	R	$Q^+$	$\bar{Q}^+$	$Q^+, \bar{Q}^+ \rightarrow$
0	X	X	NO Impact		Condensed
1	0	0	No change		form of SR latch
1	0	1	Both $Q^+$ & $\bar{Q}^+$ are 0 Reset		means $Q^+ = 0$ when $Q=1$
1	1	0	Set		$Q^+ = 0$ when $Q=0$
1	1	1	Forbidden State		This is how we write condensed

$Q^+$  means output when  $Q=0$  i.e.  $\bar{Q}$

Electronic Circuits  $\rightarrow$  Have some delay called Set-up time

If  $Q$  changes <sup>is got</sup> immediately then it could be wrong coz it hasn't changed yet. So you should give it some time.

That's why we use gated SR Latch

### SR LATCH USING NOR GATES

- 1 When  $S=0, R=0 \& Q=0$  flip-flop retains the previous state  $\rightarrow$  Called as No Change state
- 2 When  $S=1, R=0, Q=0$  it makes a transition to  $Q=1$  while  $S=1, R=0 \& Q=1$  remains at  $Q=1$  and the latch is said to be in set state that's why we put check for  $Q=1$  or  $Q=0$  ie. one value of  $Q$ .
- 3 When  $S=0, R=1, Q=0$  remains  $Q=0$  while  $S=0, R=1, Q=1$  resets to  $Q=0$  and the latch is said to be in reset state
- 4  $S=R=1$ , outputs are no longer complementary but both  $Q$  and  $\bar{Q}$  are 0 and this state is called as metastable state which is undesirable & hence

Output takes some time to change (going through gates)

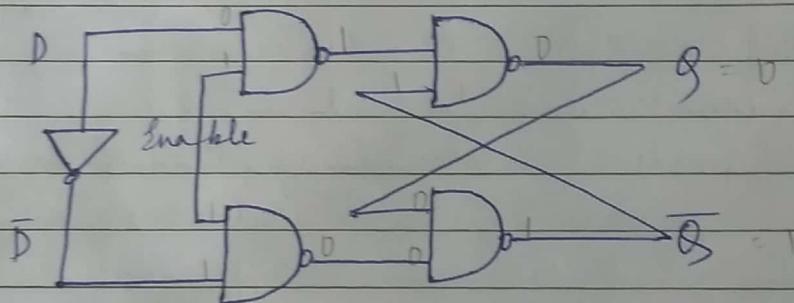
papergrid

it is called as forbidden state

NOTE  $Q$  and  $\bar{Q}$  represent the state of the latch when inputs are applied, i.e. present state of the latch while  $Q^+$  and  $\bar{Q}^+$  represent the response of the latch (output) to the inputs applied. Thus  $Q^+$  denotes the next state of the latch.

20.9.16 GATED D LATCH (flip flop gives correct op only for set & reset)

We only want the case when  $S=0, R=1$  and  $S=1, R=0$ .  
So we can use just one i/p. instead of 2 i/p's → one i/p given directly & others through inverters



Enable	D	$Q_n$	$Q_n^+$
0	X	X	No Change
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$D=0 \Rightarrow Q/p = 0$$

$D=1 \Rightarrow Q/p = 1$  So next state follows D (i/p)

$$Q^+ = D$$

Conclusions:

X	$Q_n$	$Q_n^+$
D	X	$Q_n$

$$Q_n^+ = 1$$

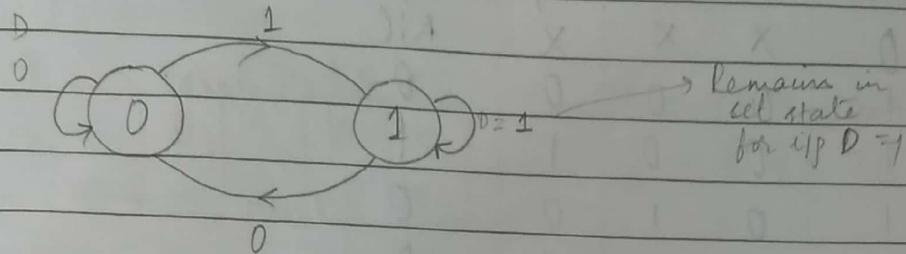
	$\bar{Q}_n$	$Q_n$
D	0	0
D	1	1

$$\Rightarrow Q_n^+ = D$$

Propagation Delay → The time it takes a change in an input signal to produce a change in an output signal

papergrid

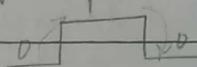
### STATE TRANSITION



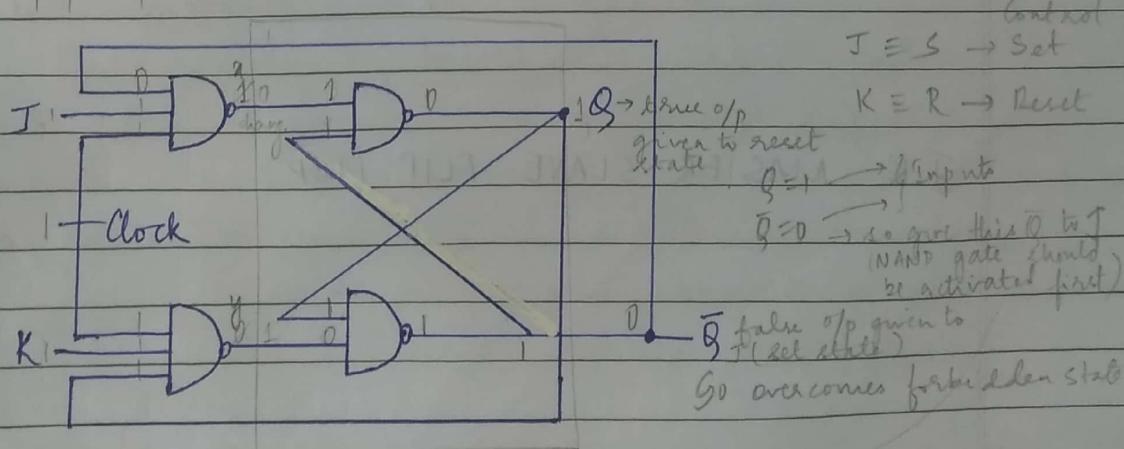
### EXCITATION TABLE

$Q$	$Q^+$	$D$	
0	0	0	$D = Q^+$
0	1	1	⇒ Next state of D flip-flop is same as $Q^+$ value
1	0	0	
1	1	1	

JK FLIP FLOP → FF means o/p do not change immediately upon Overcomes problem of forbidden state in SR giving i/p effect depends on clock Two types of triggering in FF:  
Level triggering: If flip flop changes state during positive cycle or -ve cycle (i.e. level 1 or level 0)



in edge triggering: Changes when it transitions from 0 to 1 (positive edge triggered) or when it makes trans from 1 to 0 → falling edge (negative edge triggering)



SR NAND latch  
 $S = 1, R = 0, Q = 1, \bar{Q} = 0$

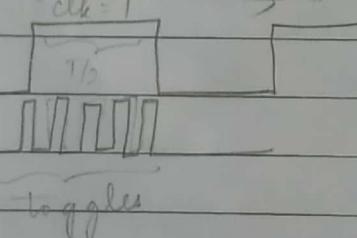
That's why JK takes more time  
 (output switches to race among  
 wires simultaneously)

When  $J=1$ ,  $K=1$ , the Q output will be in the high state after clocking i.e.  $Q_{out} = Q_1$ . This is known as toggling. The flip-flop will complement to half each time the circuit switches from papergrid to low. The ff is said to toggle.

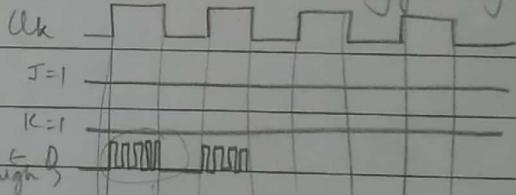
Clk	J	K	$Q$	$Q^+$	Use Mono-pulse for clock
0	X	X	X	NC	Press once
1	0	0	0	0	No change.
1	0	0	1	1	
1	0	1	0	0	{ Reset
1	0	1	1	0	
1	1	0	0	1	{ Set
1	1	0	1	1	
not controlled	1	1	0	$\overline{Q_n}$	Toggling State
+ Race Around cond	1	1	1	$\overline{Q_n} 0$	is controlled as we want to cancel race around to both which can be used in counters

Race Around cond is diff from toggle → is controlled as we want to cancel race around to both which can be used in counters. Race Around → When  $Q=0$ ,  $\bar{Q}=1$  you get  $Q^+=1$ ,  $\bar{Q}^+=0$  again these will be applied to the J NAND and K NAND Gates respectively and this continues, i.e. toggles several times within the same clock period ( $Clk=1$ ) and then becomes constant. This happens coz Q/Ps are connected to i/p/s

Toggle  
which becomes 0,  $Q_{out}$  becomes 1  
this is called  
toggling



This is level triggering



So we go for edge triggering.

So if flip-flop keeps changing only during the edges it won't be a problem. So it won't change throughout clock period.

### JK MASTER-SLAVE FLIP FLOP

Whatever the master does, the slave copies the same thing but in the next half cycle.

( $Clk=1 \Rightarrow$  Positive cycle?)

- i)  $T'$  is time taken by clk to generate output, then  $T'$  should be  $< T_2$  for racing. But since we don't want racing so 3 conditions to overcome racing
  - i)  $T_1 <$  prop delay of ff
  - ii) Edge-trig (Not enough time to race)
  - iii) Master Slave ff.

$Q_n$   
or  $Q_{n-1}$   
Present

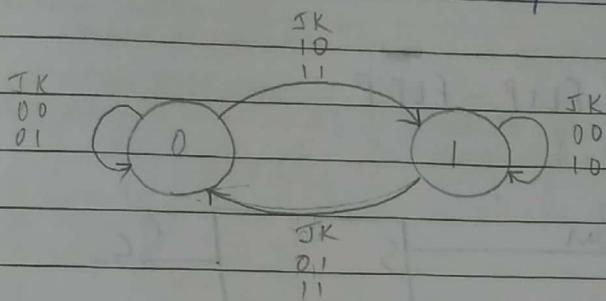
$Q_n$   
Next state  
→ Q that you get  
with present state

## CHARACTERISTIC EQUATION

	$\bar{K}Q$	$\bar{K}Q$	$KQ$	$K\bar{Q}$
$\bar{J}$	0	1	0	0
$J$	1	1	0	1

$$Q_t = \bar{J}Q + J\bar{Q}$$

## STATE TRANSITION DIAGRAM

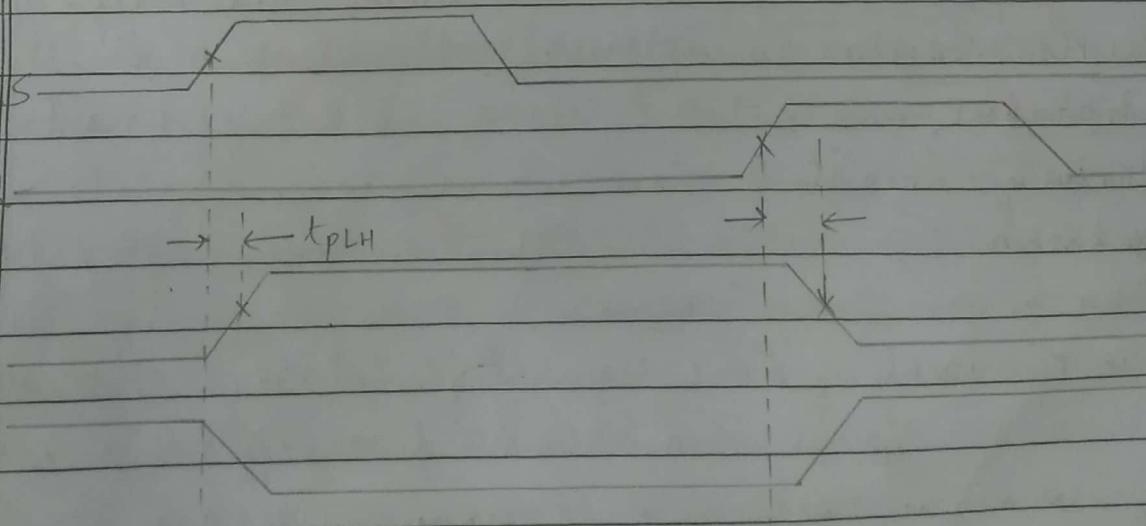


## EXCITATION TABLE

$Q$	$Q^+$	$J$	$K$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

the time it takes a change in an input signal to produce a change in an output signal

## PROPAGATION DELAYS IN AN SR LATCH w/o Clock Signal



### TIMING DIAGRAM FOR AN SR LATCH

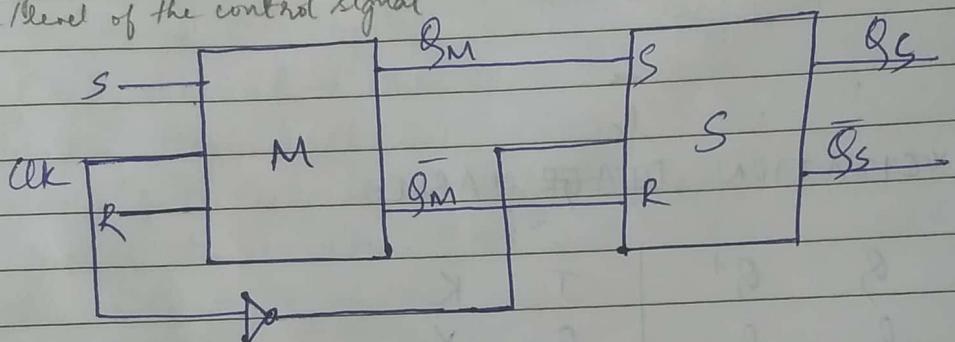
when  $S=R=1$  it is forbidden state. It can be in 0 state or 1 state  
(assume) This is a metastable state.

Now if we make  $S,R=0$  it should retain previous state. But  
we don't know previous state coz it was forbidden and we  
had just assumed it so you put a block   
coz you couldn't predict it.

→ two cascaded sections, called  
master & slave.

### MASTER SLAVE SR FLIP - FLOP

Info is entered into the master on one edge or level of  
a control signal & is transferred to the slave on  
the next edge/level of the control signal



The o/p of M goes to i/p of S. M gets activated during  
one cycle of clk & S in -ve cycle

This is why you get delayed o/p

Slave copies action of master only during next half-cycle

Master is disabled during -ve & slave is enabled

When  $S=R=0, C=0 \rightarrow$  slave retains  $S=R=0, C=1 \rightarrow$  master retains

Initially when  $C=0$ , slave retains previous state which is unknown.

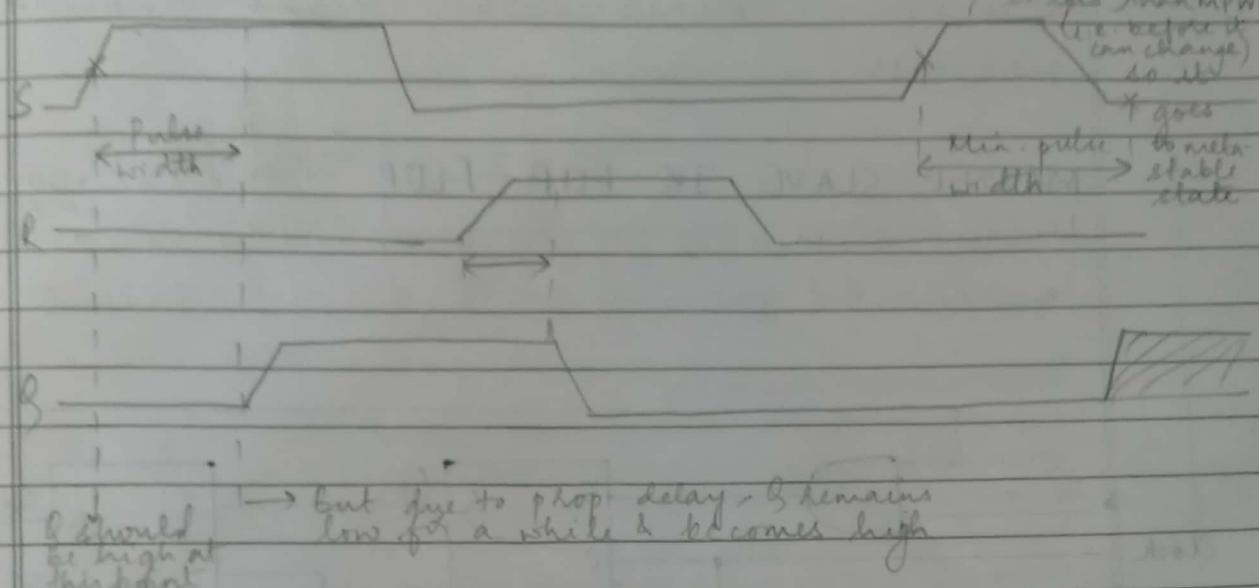
Min. of amount of time required for latching (i.e. to store or  
for getting o/p i.e. when S becomes 1 only after a min. amount of time  
(will become one since so many gates are involved))  
This is called Pulse Width  
MIN PULSE WIDTH CONSTRAINT

NPN  $\rightarrow$  Then out of time a signal must be applied in order to produce a desired result

See pg 513

Depends on constraint (D, JK, SR...) papergrid

MPW  
Minimum Pulse Width Constraint



$Q$  should be high at this point  $\rightarrow$  but due to prop. delay -  $Q$  remains low for a while & becomes high

Set-up time  $\rightarrow$  Before latching action

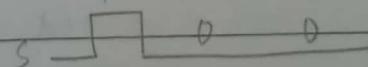
Hold Time  $\rightarrow$  after latching action some time required before going into next state

SR Master Slave

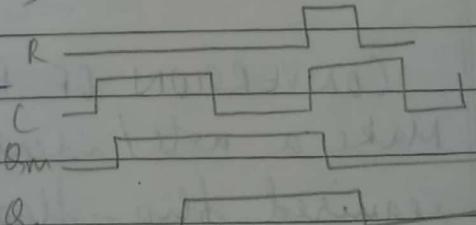
Used to remove bouncing effect of key (key-switch or keyboard key)

Pressing the key -  $Q$  0 to 1 transition

Releasing the key - 1 to 0



Using a pull-down resistor



TIMING DIAGRAM FOR A GATED D LATCH

Ideal  
 $Q^+ = D$  so one-to-one correspondence except for propagation delay

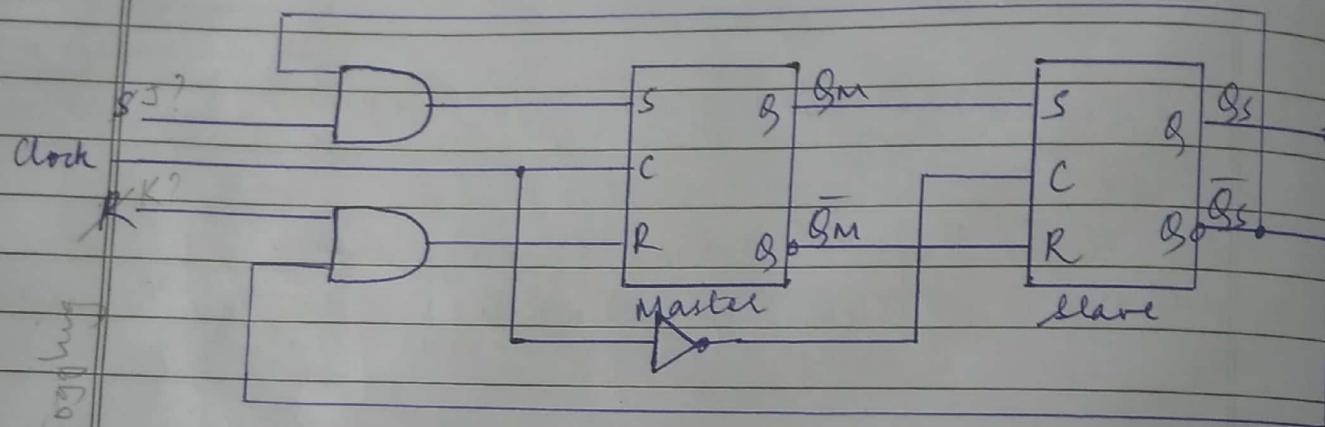
edge triggering  $\rightarrow$  you need some set-up time & hold time

$\bar{S}, \bar{R} \rightarrow$  use NAND Gates for min. gates;  
directly giving  $S, R \rightarrow$  Using AND Gates

papergrid

If you don't give it min. required amt. of time for it to change it goes into metastable state and "Unpredictable response"

MASTER-SLAVE JK FLIP-FLOP Toggling happens only once each half-cycle  
Modify SR M-S flip flop by using AND Gates so no race-around cond.  
Coz NAND will give  $\bar{S}, \bar{R}$  to J, K too many times toggle



When  $J=K=1, Q=0, Q^+=1$  i.e. changes from 0 to 1

Synchronous up  $\rightarrow$  JK, SR etc (along with clock)

Asynchronous ups  $\rightarrow$  Direct dips to force the Flip-Flop to be in set or reset called Preset State  $\rightarrow$  for set state

Clear State  $\rightarrow$  for reset state

### CONVERSION OF FLIP-FLOPS

Step 1 Make a note / identify the available flip flop & 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 from the expression

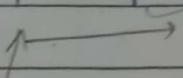
Q) Convert from SR to D flip flop.

(Soln) Available flip-flop = SR

Required flip-flop = D

2) Characteristic Table for D flip-flop

D	$Q^-$	$Q^+$	S	R	$(Q^+ = D)$
0	0	0	0	X	
0	1	0	0	1	
1	0	1	1	0	
1	1	1	X	0	



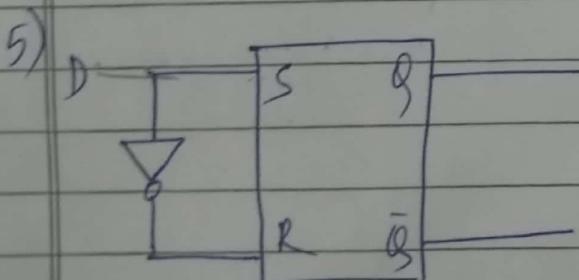
3) Excitation table for available (SR)

4) Boolean Exp for S & R

D	$Q^-$	$\bar{Q}$	$Q$	D	$Q^-$	$Q$
0	0	0	0	0	X	1
0	1	X	1	1	0	0

$$S = D$$

$$R = \bar{D}$$



Q1 Convert from SR to JK flip flop and JK to D.

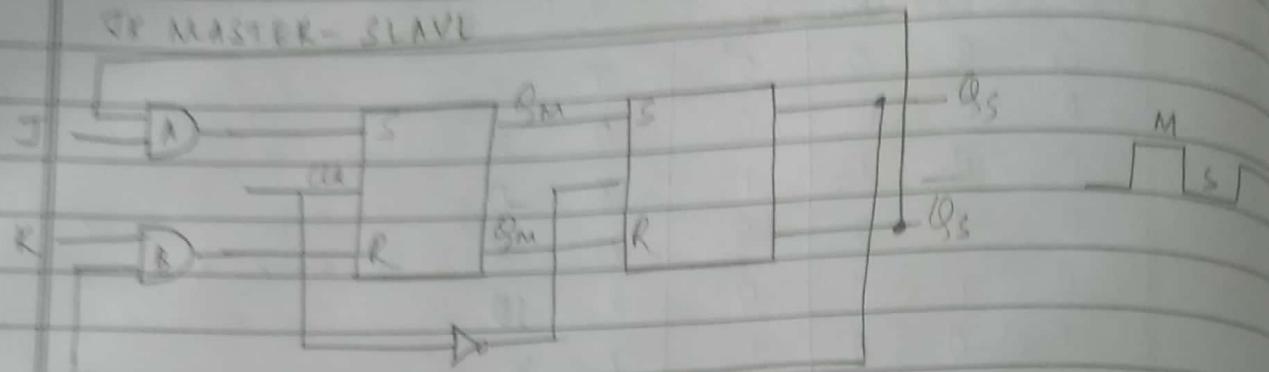
i) SR to JK

Available: SR

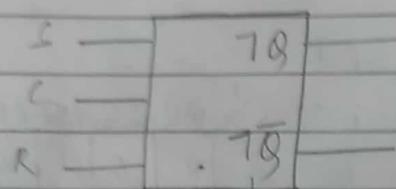
Required: JK

## 27.9.16 FLIP-FLOPS AND ITS APPLICATIONS

FOR MASTER-SLAVE



Clock decides at what time output should be obtained



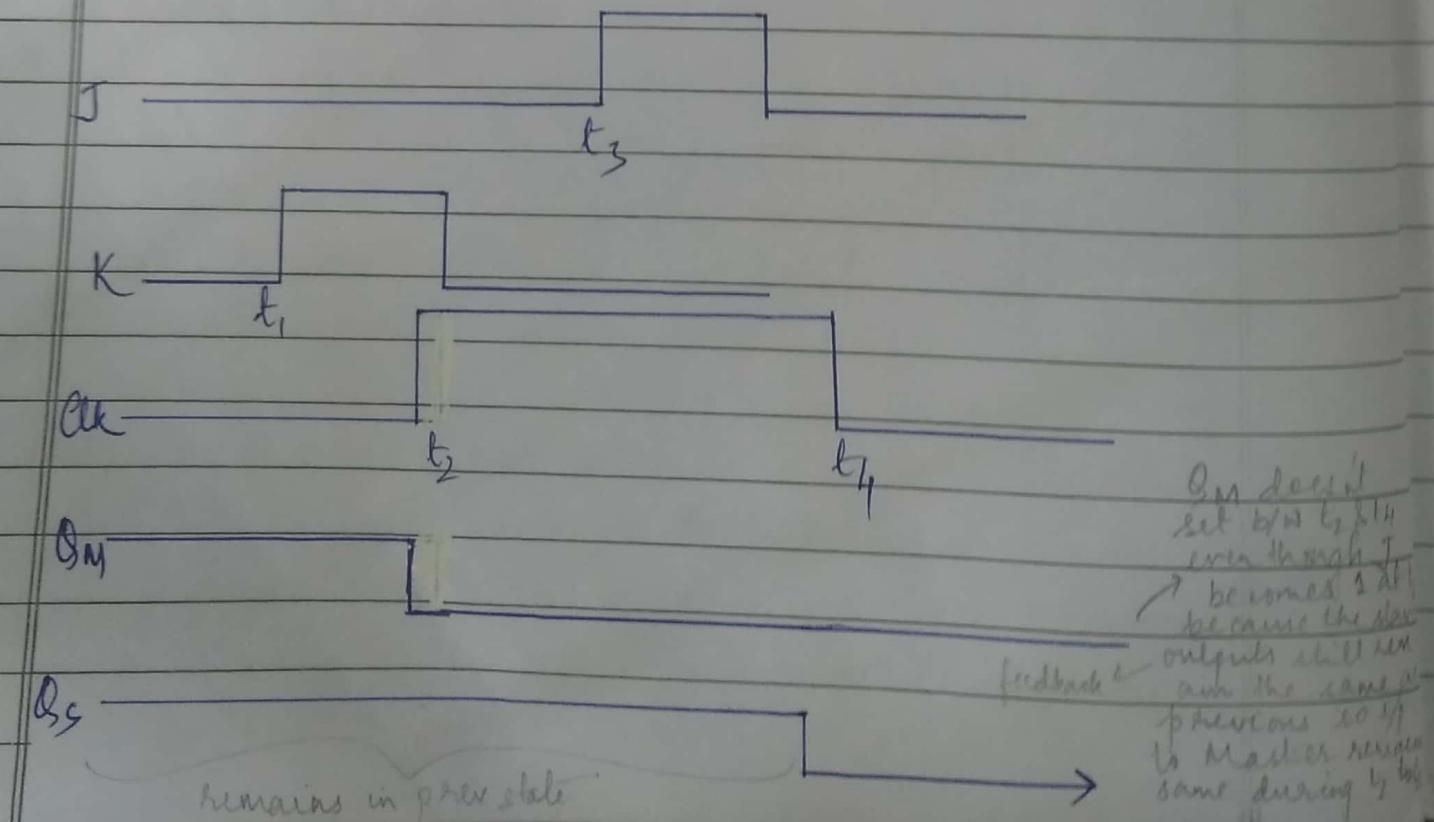
↳ postponed output

To the next cycle

- Master-slave use

- Output from slave given back to AND

### O's CATCHING



$$\bar{Q}_M - \bar{Q}_S = 0$$

Let the flip-flop be set ( $Q_M = Q_S = 1$ )

At time  $t_1$ , K goes high because  $Q_S = 1$

And at time  $t_2$ , when clock goes high, output of second AND Gate (B) becomes high (1) and the master resets. responds only when  $\text{clk} = 1$

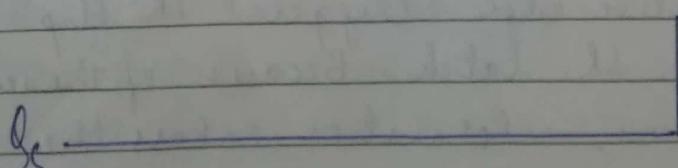
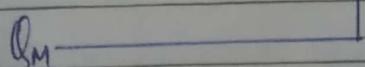
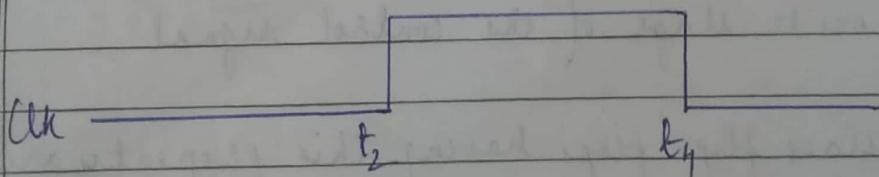
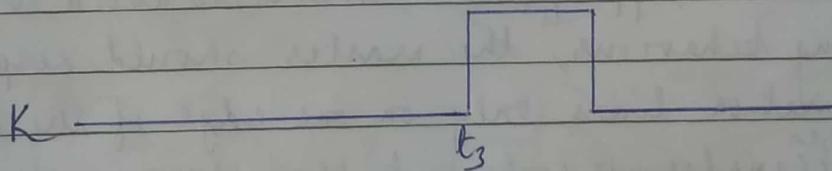
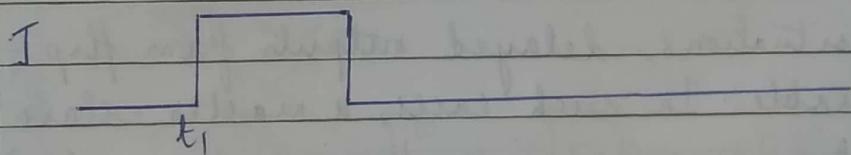
Note that  $Q_S$  is still at logic 1  $\rightarrow$  it resets only during next half-cycle

At time  $t_3$ , J input becomes 1 because  $Q_S$  is still zero; output of AND Gate A is 0 and  $J=1, K=0$  goes unrecognized

$\rightarrow$  only when  $\text{clk} = 0$   
Slave resets at time instant  $t_4$  (Negative cycle). This is called 0's catching

$\because Q_M$  doesn't become 1 again during after  $t_2$  during this clock cycle, it is called 0's catching ( $\because$  it remains 0)

### 1's CATCHING



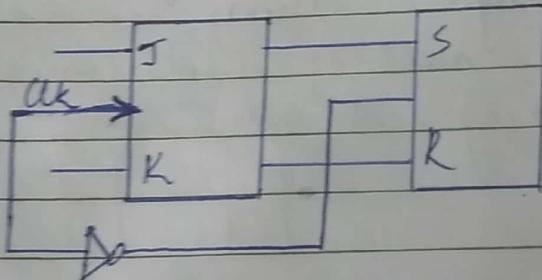
- +ve edge triggered
- -ve edge triggering

To avoid the catching problem, the signals on the info. lines are latched from changing during the time the master is enabled. Once the ~~latch~~ <sup>papergrid</sup> occurs, the ~~latch~~ <sup>papergrid</sup> remains unresponsive to info. input changes until the next trigger edge.

At time instant  $t_1$ ,  $J=1$  and  $K$  at time  $t_2$ , when clock becomes high,  $Q_S = 1$ , AND Gate A becomes 1 and the master sets ( $Q_M = 1$ ).  $Q_S$  is still at 0 and this keeps AND Gate B at 0 when K becomes 1 at  $t_3$ . Thus  $J=0$ ,  $K=1$  input goes unrecognized.

Slave sets at  $t_4$  (Negative half-cycle) of This is called 1's catching so we go for edge triggering

### MS ~~FB~~<sup>FFs</sup> WITH DATA LOCKOUT



Clock used here is an edge triggered clock

In some situations, delayed outputs from flip-flops are desirable. In such cases, a master-slave configuration is appropriate. However, to avoid 0's and 1's catching behaviour, the master should respond to the information lines only on one edge of the control signal & transfer its content to the slave on the next opposite stage of the control signal

Master slave flip-flops having this property are said to have data lockout.

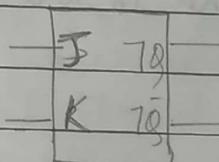
### CONSTRUCTION

Master uses positive edge-triggered JK flip-flop and slave uses SR latch. Because of the inverter between two sections, information enters the Master

To eliminate race-around you can use either Master-slave FF or edge-triggering

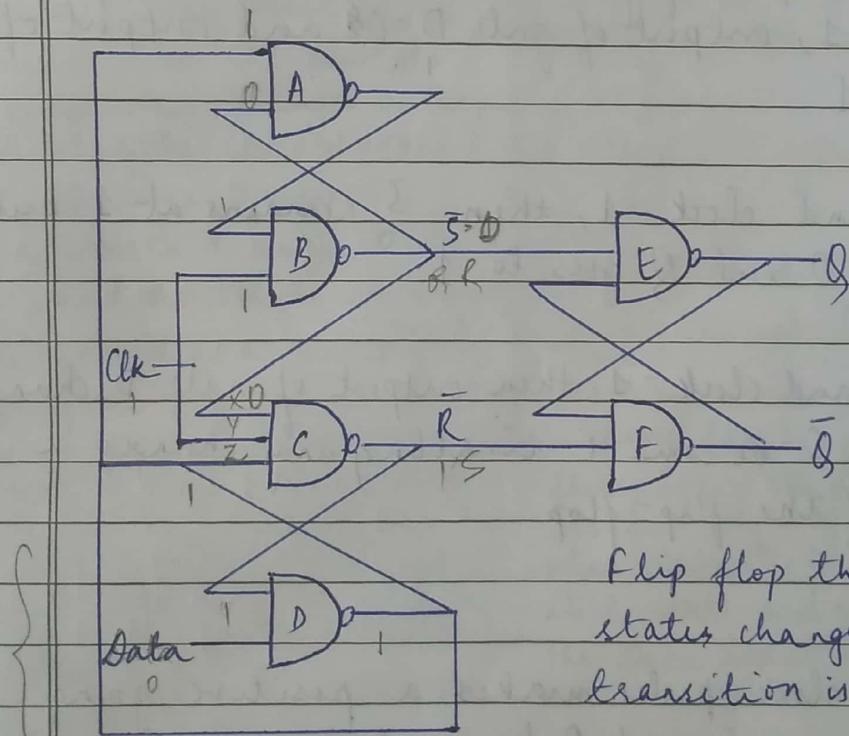
Master is triggered during rising edge  
Slave is triggered during falling edge  
on positive edge of the control signal  
Because Master is an edge-triggered flip-flop,  
any changes on J & K info lines while control  
signal is 1 are disregarded.

Content of M is transferred to Slave during negative  
edge of the control signal hence desired output delay  
is achieved



Here master only responds during  
edge so only J, K values at these  
edges are taken into consideration  
instead during the entire  $clk = 1$   
data (used in registers)

POSITIVE EDGE TRIGGERED FF 3 basic flip flops



Flip flop that synchronises the  
status changes during a clock pulse  
transition is edge triggered FF

It consists of 3 basic flip-flops where NAND gates  
A and B make one flip-flop  
NAND gates C and D form second flip-flop  
NAND Gates E & F form the third flip-flop

\* When you use edge-triggered is called flip-flop  
level-triggered " latch

papergrid

Take middle as  $\bar{S}$  or  $\bar{R}$

and give the outputs of the circuit

The inputs  $\bar{S}$  and  $\bar{R}$  of 3rd basic flip-flop must be maintained at logic 1 for outputs to remain in their steady state

When  $\bar{S}=0$ ,  $\bar{R}=1$  output goes to set state

Inputs  $S$  and  $R$  are determined from the states of other two basic flip-flops

These two basic flip-flops respond to the external inputs (D and clock)

When clock pulse = 0, input Data may be 0 or 1

In either case, a 0 on clock causes outputs of gates B and C to go to 1. Then  $D=0$  irrespective of Data

when  $B=0$ , output of gate D = 1 <sup>This is fed back to A</sup> and output of gate A = 0

when Data = 1, output of gate D = 0 <sup>and output of gate A = 1</sup>

A becomes 1

during

means 0 to 1 transition we change Data & so

If Data = 0 and clock = 1, then  $\bar{S}$  remains at 1 but

$\bar{R}$  changes to 0 and  $Q$  goes to 1

Take  $Q=1$  (i.e., 1st state)  $\Rightarrow \bar{S}=0, \bar{R}=1$

$\bar{S}=0$  only if 2/p to B = 1

right?

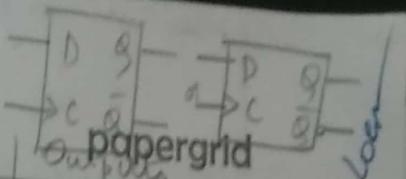
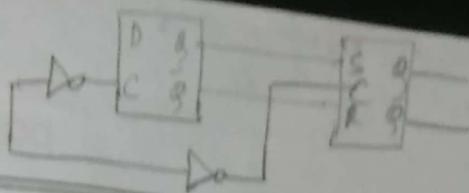
If Data = 1 and clock = 1 then output of gate D changes but R and S become 1 disabling any changes in the output of the flip-flop.

so Q remains 1 since  $\bar{S}$  remains 0 & R as 1 so this shows that it is only triggered during edge and it doesn't matter even when Data changes from 0 to 1 during  $\bar{A}k=1$

Summary:

When input clock pulse makes a positive going transition the value of Data is transferred to  $Q$ . A negative pulse transition does not affect the output, nor does it when Clock = 0.

Hence the edge-triggered flip-flop eliminates any feedback problems in sequential circuits just



as a Master Slave <sup>FF</sup> does

NOTE Let up and hold time must be taken into consideration while using this type of flip-flop

When Clock is 1, then the three inputs to gate C are equal to 1 and output of gate C is 0  $\Rightarrow \bar{R} = 0$  and  $\bar{S} = 1$  & output Q resets.

With Data = 1, when clock becomes 1;  $\bar{S} = 0$ ,  $\bar{R} = 1$  and output Q sets

QMS Once Data input gets latched at the output during the rising edge of the clock any changes in Data between positive edges goes unrecognized.

eg Let Q be 1 with  $\bar{S} = 0$  and  $\bar{R} = 1$  as Data = 1 at the rising edges of the clock. Immediately after the rising edge of the clock, let Data = 0; output of Gate D is 1. Now  $Y = 1$ ,  $Z = 1$ ,  $X = 0$  keeps  $\bar{R}$  at 1

Both inputs of Gate B are at 1 and  $\bar{S}$  continues at 0 and output 'Q' continues to be 1 or the flip-flop remains in the set state though Data has become 0 after the positive edge of the clock.

Conclusion: All the changes between positive edges are ignored.

### APPLICATIONS OF SR LATCH

Pushing keys or push buttons

Pressing & depressing - Because of spring it doesn't immediately

When you press a key & CC which way goes from B to A. On its way we cannot say whether A & B have 0 or 1. This is called high impedance state. They may have some intermediate value (like metastable). Resistors which make  $A = B = 0$  and make it no-change. Same while transition from 0 to 1.

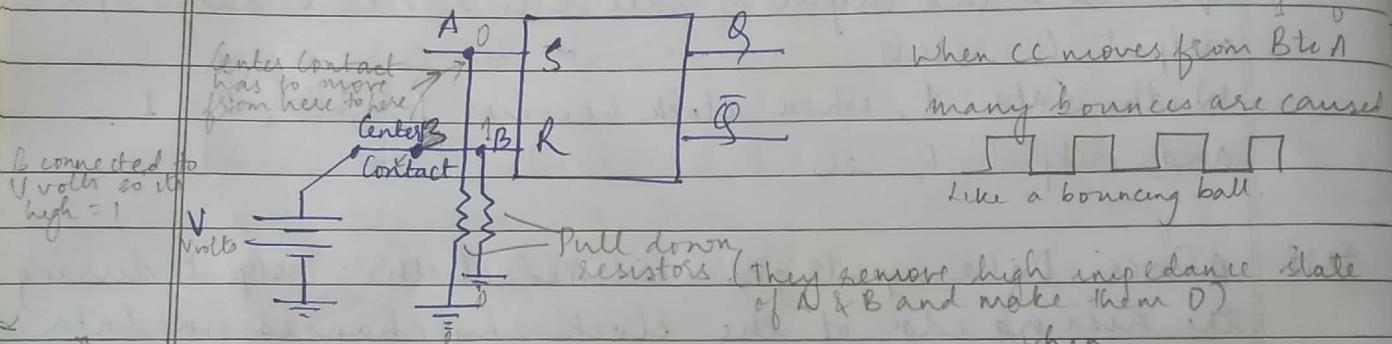
**papergrid**

go to 1 state when you press. & it bounces many times. Spring in key causes propagation delay and toggle effect due to spring. This effect is called high impedance state.

Remove bouncing effect using SR Latch

SR  $\rightarrow$  Called Debouncing Circuit

### SWITCH DEBOUNCER CIRCUIT



We're using the fact that in an SR latch,  $S = R = 0$ , it retains previous state.

While pressing, it bounces.

Cannot predict value of A & B

High impedance state

Appears like key has been pressed multiple times

Hence 2 resistors connected to A & B called pulldown res to bring S and R to 0 to retain the previous state & toggles are removed.

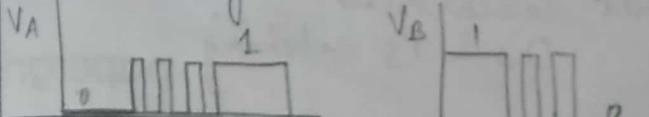
28.9.16

Q Design a JK flip-flop using a 2:1 MUX, D flip-flop & an inverter if necessary

Write func<sup>n</sup> table for available JK

Group 0s & 1s together for Q here.

Voltage at A starts at 0. When you press key it has to go from 0 to 1. But it changes several times (i.e. bounces) and finally becomes 1 (immediately)



papergrid

JK ff

$Q$   $J \neq K$   $Q^+$

0 0 0

0 Here  $Q^+ = J$

0 0 1

D best inspite of  $S$   $K \rightarrow$

0 1 0

1 Set

0 1 1

1 Toggles

1 0 0

1

1 0 1

0 Here  $Q^+ = \bar{K}$

1 1 0

1

1 1 1

0

selected line

So

2:1

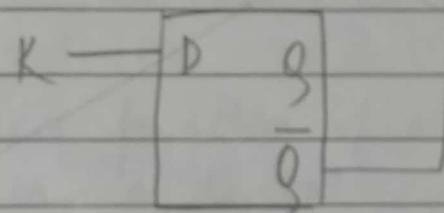
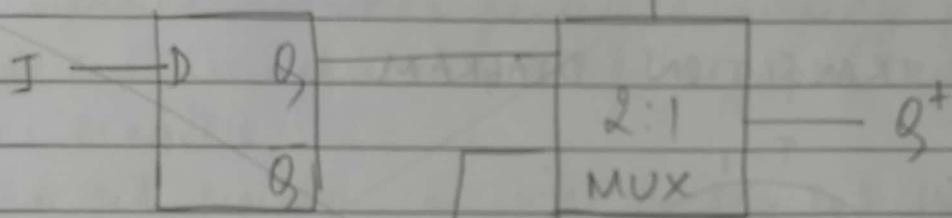
MUX

clk

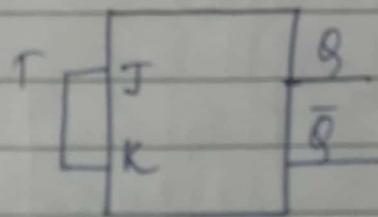
D

Q

$\bar{Q}$



Toggle FLIP-FLOP : This is used to get toggling state



When  $J=K=0$ ,  $T=0$  and it retains

$Q$  value

When  $J=K=1$ ,  $T=1$  it toggles

Combining J & K together & calling it as T

Edge-triggered  $\rightarrow$  Lesser gates  
 Overcomes race-around  
 " 0's & 1's catching papergrid

28.9.16

## REGISTERS

FUNCTION TABLE

T	Q	$Q^+$
0	0	0
0	1	1
1	0	1
1	1	0

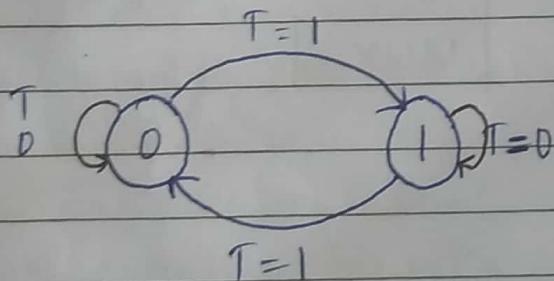
CHARACTERISTIC EQUATION  $\rightarrow$  Useful for analysing func<sup>n</sup> of FF

$T^Q$	0	1
0	0	1
1	1	0

$$Q^+ = \overline{Q} T + \overline{T} Q$$

$$= Q \oplus T$$

## STATE TRANSITION DIAGRAM



EXCITATION TABLE  $\rightarrow$  Useful for conversion and for counter construction

Q	$Q^+$	T
0	0	0
0	1	1
1	0	1
1	1	0

can be used for storing or shifting data  
 in flip-flops