UNIT - VI

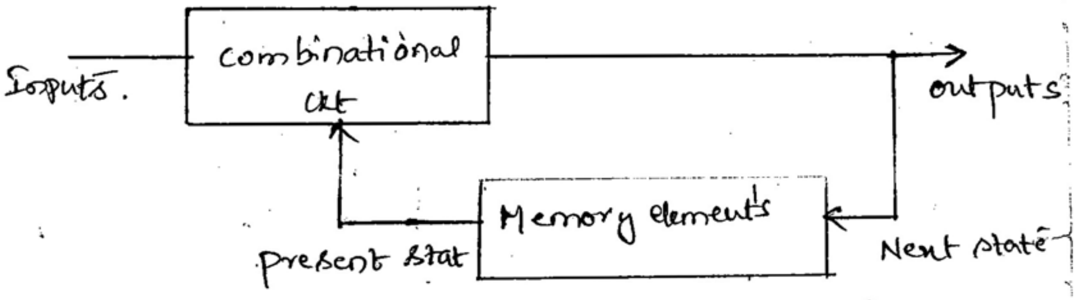
Sequential circuits-I

The digital system comprises of combinational circuit or sequential circuit. The combinational circuit is only a part of digital system. But, the major part of the digital system is sequential circuit.

we have seen the analysis & design of combinational circuit. The analysis & design of sequential circuit is also important as it forms the major part of the digital system.

There are many applications in which digital o/p's are required to be generated in accordance with the sequence in which the i/p sig's are received. This requirement cannot be achieved using a combinational circuit.

Figure below shows the block dgm of sequential circuit



In sequential circuits, the memory elements are connected to combinational circuit as feed back path. The information stored in memory elements at any given time defines the present state of sequential circuits.

The present state & external i/p's determine the o/p's & next state of sequential ckt.

Thus we can say that, the sequential circuit is a function of external i/p's, internal states (present state & next states), & o/p's.

IV

Comparison b/n combinational ckts & Sequential ckts:

Combinational ckt	Sequential ckt.
1. In combinational ckt, the o/p at all times is dependent on the combination of i/p variables. o/p = f(Input variables).	1. In sequential ckts, the o/p depends not only on the present i/p variables, but, they also depend upon the past history of these i/p variables. o/p = f(i/p variables, past history of the i/p variables).
2. Memory unit is not required.	2. Memory unit is required to store the past history of i/p variables.
3. It is faster in speed, b'cas the delay b/n i/p & o/p is due to propagation delay of gates.	3. It is slower than the combinational circuits.
4. Easy to design	4. sequential circuits are comparatively complex to design
5. Ex: Parallel adder (with carry look Ahead) is a combinational ckt.	5. Ex: Serial adder is a sequent ckt.

Classification of sequential ckts:

The sequential circuits can be classified depending on the timing of their signals as synchronous sequential ckts & Asynchronous sequential ckts.

In synchronous sequential ckts, signals can affect the memory elements only at discrete instants of time.

In Asynchronous sequential ckts, change in i/p sig's can effect memory element at any instant of time.

The memory elements used in both ckts are flip-flops which are capable of storing 1-bit information.

Syncronous sequential ckts	Asyncronous sequential ckts.
1. In syncronous ckts, memory elements are clocked flip-flops.	1. In asyncronous ckts, memory elements are either unclocked flip-flops or time delay elements.
2. In syncronous ckts, the change in i/p slg can affect memory element upon activation of clock signal.	2. In Asyncronous ckts, change in i/p slg can effect memory element at any instant of time.
3. The maximum operating speed of clock depends on time delays involved.	3. Because of absence of clock, Asyncronous ckts can operate faster than syncronous ckts.
4. Easier to design.	4. More difficult to design.

Asyncronous sequential circuits:-

It consists of a combinational ckt & delay elements connected to form feed back loops as shown in figure below.

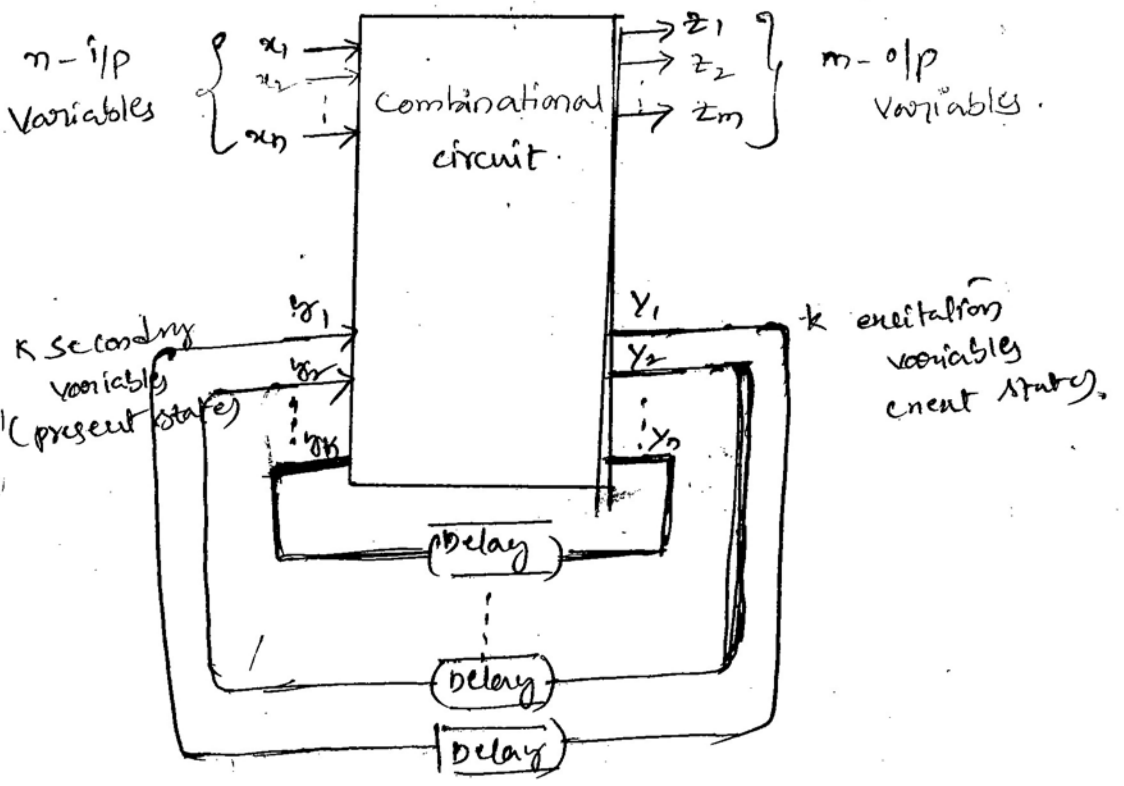


fig: Block diagram of an Asyncronous sequential ckt.

It consists of 'n' i/p variables, 'm' o/p variables and K internal states. The delay elements provide a short term memory for sequential ckt.

The present state & next state variables in asynchronous sequential ckts are called secondry variables & excitation variables, respectively.

When an i/p changes, the secondry variables, i.e $y_1, y_2 \ldots y_K$ do not change instantaneously.

26/03/12 8th

68, 76, 83, 84, 90, 97B, 108

- Because certain amount of time is required for the i/p sig to propagate from i/p terminals through the combinational ckt & delay elements. The combination ckt generates Y excitation variables, which gives the next state of the circuit. The excitation variables are propagated through delay elements to become new present state for secondry variables, i.e $y_1, y_2 \ldots y_K$.

Note: In steady state condition, excitation & secondry variables are same, but during transition they are different.

- There are two types of Asyncronous circuits, based on how i/p variables are to be considered.

1. Fundamental (Level) mode ckts &
2. pulse mode circuits.

Fundamental mode ckt assumes that:

(i) The i/p variables change only when the circuit is stable.
(ii) only one i/p variable can change at a given time &
(iii) i/p's are levels & not pulses.

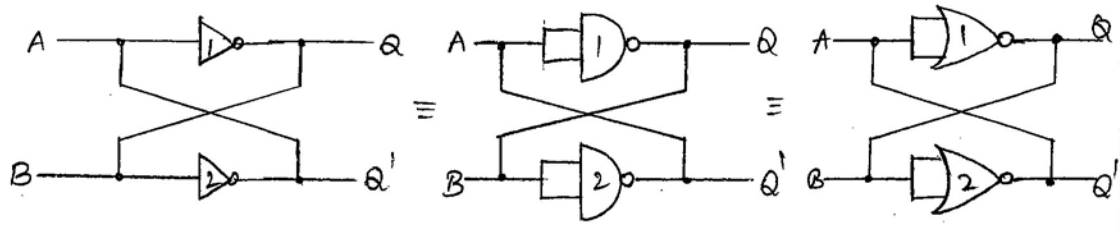
Pulse mode circuit Assumes that:

(i) The i/p variables are pulses instead of levels.
(ii) the width of pulse is long enough for the circuit to respond to the i/p &
(iii) The pulse width must not be so long that is still present after the new state is reached.

Latches & Flip Flops :-

Latches & Flipflops both are bistable elements. These are the basic building blocks of most Sequential ckts.

Fig below shows the basic bistable element used in latches & flipflops.



The basic bistable element has 2 o/p's Q & Q' It has 2 cross coupled inverters i.e, the o/p of 1st inverter is connected as an i/p to 2nd inverter & vice versa.

The basic bistable element has 2 stable states logic 0, logic 1, hence the name bistable. The 2 stable states of basic bistable elements are used to store 2 Binary elements '0' & '1'.

→ The basic bistable element constitutes :-

1. The o/p's Q & Q' are always complementary.
2. The ckt has 2 stable states. The state corresponding to Q=1 is reffered to as 1 state or Set state, & state corresponds to Q=0 is referred to as 0 state or reset state.
3. If the ckt is in Set (1) state, it will remain in set state & if the ckt is in reset (0) state, it will remain in reset state.

This property of ckt shows that it can store 1-bit of digital information.

4. The 1-bit information stored in ckt is locked or latched in the ckt.

∴ This ckt is also referred to as latch.

Modified Ckt for 1-bit Memory cell :-

In basic bistable element, when the pr. is switched on, the ckt switches to one of the stable states i.e., Q=0 or Q=1 & it is not possible to predict this state. Thus we cannot store/enter the desired digital formation in it.

By replacing inverters 1 & 2 with 2 i/p NAND gates, we can use one i/p terminal of NAND gates to enter the desired digital information & other i/p is cross coupled to o/p.

Fig below shows the modified ckt for 1-bit memory cell.

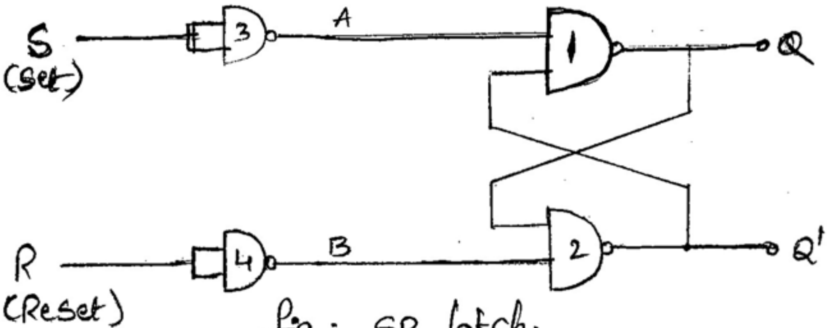


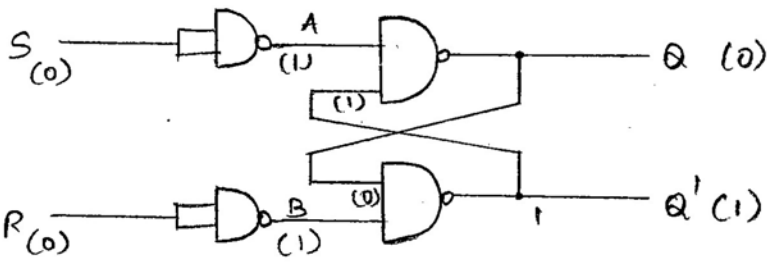
fig: SR latch.

The modified ckt consists of 2 inverters 3 & 4 connected to enter the desired digital information. I/p for gate 3 is S & i/p for gate 4 is R. Hence this latch is called SR latch

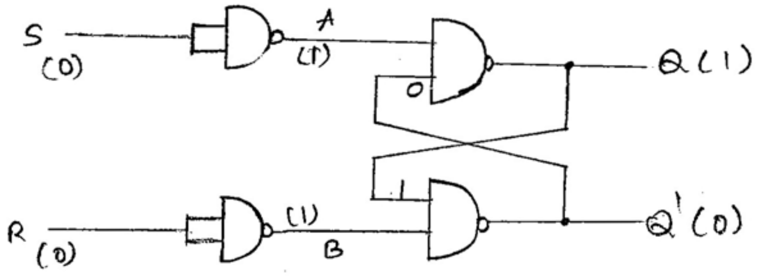
Operation of SR g latch :-

Case 1: S=0 R=0

Initial state : Q=0 , Q'=1



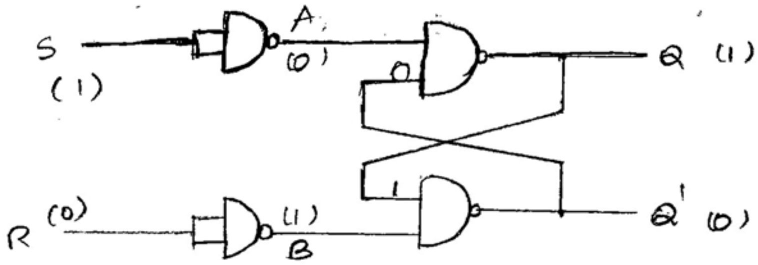
Initial state : Q=1 , Q'=0



when S=R=0 the o/p's do not change.

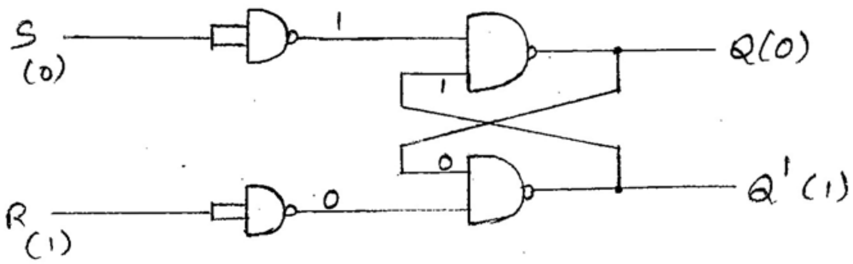
Case 2:-

S=1 & R=0



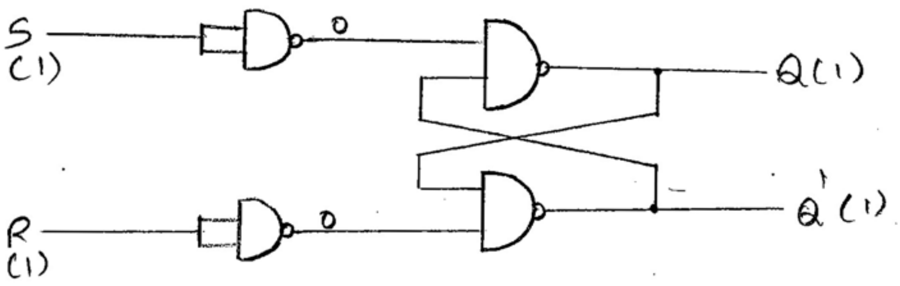
The i/p S=1, R=0 makes Q=1 i.e Set state.

Case 3:- S=0 & R=1



The i/p S=0 & R=1, makes Q=0 i.e., reset state

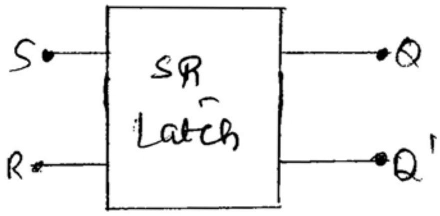
Case 4: S=1 & R=1



Note:-

When S=R=1, both the o/p's Q & Q' try to become '1' which is not allowed & therefore, this i/p condition is prevented & o/p is called Indeterminate state.

fig. below shows the symbols & truth table for SR latch



ⓐ Symbol

S	R	Qn	Qn+1	State
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	X	Indeterminate
1	1	1	X	

ⓑ Truth table

→ with both i/p's low, the o/p doesnot change & latch remains latches in its last state. The condition is called inactive state because nothing changes

→ when R=0 & S=1, the Q o/p of latch is Set (1)

→ when R=0 & S=0, the Q o/p of latch is Reset (0)

→ when R=S=1 o/p is unpredictable. This is called indeterminate condition.

The characteristic equation for SR latch can be obtained by simplifying Q_{n+1} function by K-map as:

S \ R Qn	R'Qn'	R'Qn	RQn	RQn'
S'	0	1	0	0
S	1	1	x	X

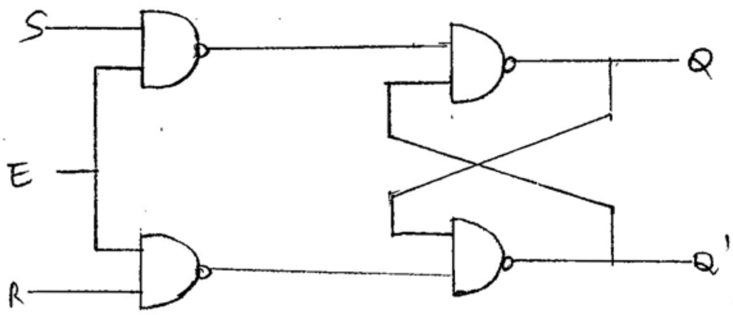
$$\boxed{Q_{n+1} = S + R'Q_n}$$

Gated Latches :-

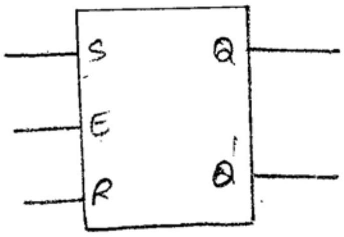
In the SR latch we have seen that o/p changes occur immediately after the i/p changes occur i.e., the latch is sensitive to its S & R i/p's at all times. However, it can be modified to create a latch i.e., sensitive to these i/p's only when an enable i/p is active. Such a latch with enable i/p is called "Gated SR latch"

Gated SR latch :-

From the truth table below, we can say that, the ckt behaves like SR latch when E=1 & retains its previous state when E=0.



ⓐ SR latch with enable i/p using NAND gates



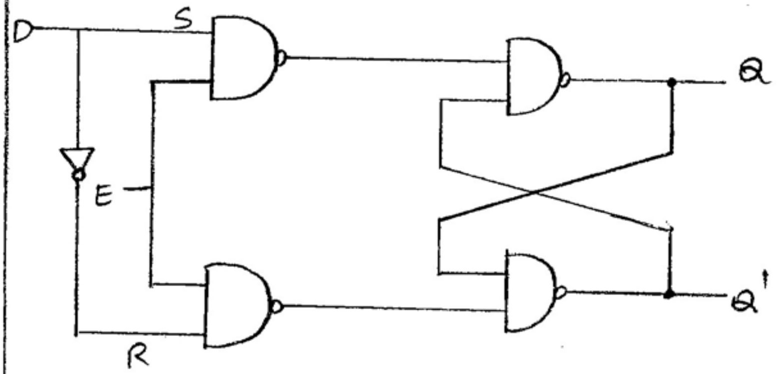
ⓑ Logic Symbol

E	S	R	Q_n	Q_{n+1}	State
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	
1	1	1	0	x	Indeterminate
1	1	1	1	x	
0	x	x	0	0	No change
0	x	x	1	1	

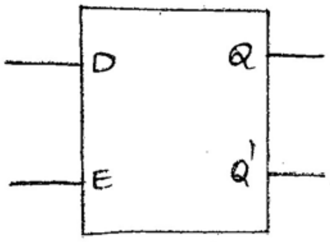
Gated D latch :-

From the truth table of SR latch we Can realize that when both i/p's are same the o/p either doesnot change or it is invalid (i/p's → 00, no change & i/p's → 11, invalid). In many practical applications, these input conditions are not required. These i/p conditions Can be avoided by making them Complement of each other.

This modified SR latch is Called Gated 'D' latch.



ⓐ D latch



ⓑ Logic symbol

E	D	Q_n	Q_{n+1}	State
1	0	X	0	Reset
1	1	X	1	Set
0	X	X	Q_n	No charge (NC)

From truth table, we Can Conclude that, the Q o/p follows the 'D' i/p. For this reason 'D' latch is also Called transparent latch

The characteristic eq'n for 'D' latch with enable 'E' is

E \ D Q_n	$D'Q_n'$	$D'Q_n$	DQ_n	DQ_n'
E'	Q_n	Q_n	Q_n	Q_n
E	0	0	1	1

$$Q_{n+1} = E'Q_n + ED$$

Flip flops :-

Latches & flipflops forms the basic building blocks of the most sequential ckts. The main difference b/w latches & flipflops is the method used for changing their state.

A simple latch forms the basis for the flip-flops. we have seen SR & D latches with enable I/p. Latches are controlled by enable sig & they are level triggered, either +ve level triggered or -ve level triggered. The o/p state is free to change according to S & R I/pls values, when active level is maintained at enable I/p.

flip flops are different from latches. flip flops are pulse or clock edge triggered instead of level triggered.

level & edge triggering :-

level triggering :

In level triggering, the o/p is changed according to i/p's when active level (either +ve or -ve) is maintained at enable i/p. There are 2 types of level triggering latches,

(1) positive level trigger &
(2) negative level " .

positive level triggered :

The o/p of latch responds to i/p change only, when enable ~~x~~ i/p is 1 (HIGH).

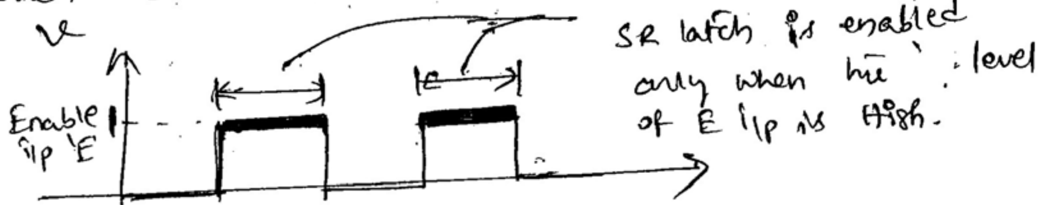


fig: positive level triggering.

Negative level Triggering

The o/p of latch responds to i/p change only, when enable i/p is 0 (LOW).

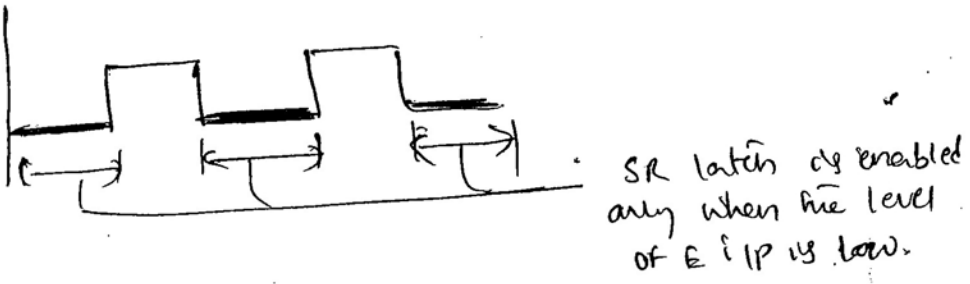
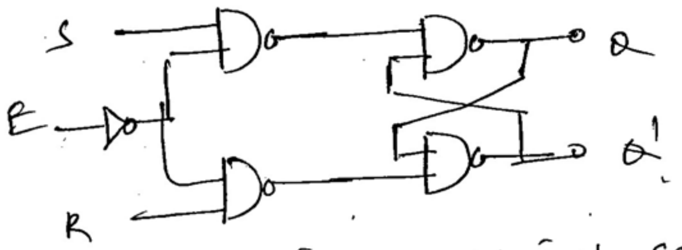
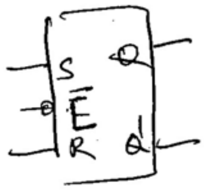


fig. below shows the ckt & symbol for negative level triggered SR latch.



ⓐ negative level triggered SR latch.



ⓑ Logic symbol)

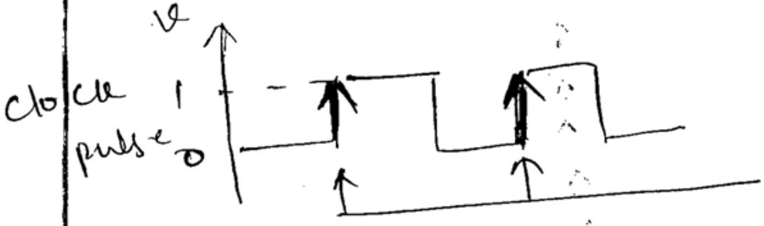
Edge triggering

In edge triggering the o/p responds to changes in i/p only at the positive or negative edge of clock pulse at the clock i/p. There are 2 types of edge triggering.

1. +ve edge triggering
2. -ve edge triggering.

+ve edge triggering :-

In positive edge triggering, the o/p responds to the changes in the i/p, only at the +ve edge of the clock pulse at clock i/p.

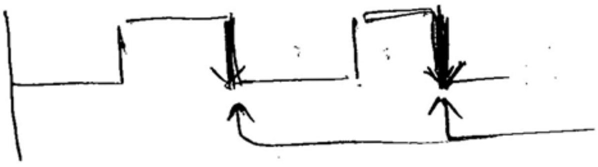


o/p's responds only at the +ve edge of the pulse.

fig: +ve edge triggering

Negative edge triggering :-

In negative edge triggering, the o/p responds to the changes in the i/p only at the -ve edge of clock pulse at clock i/p.



o/p responds only at the -ve edge of the pulse.

fig! - negative edge triggering.

Clocked SR flip-flop :

Positive edge triggered SR ff :

fig. below shows +ve edge triggered SR flip flop. The ckt is similar to SR latch except enable i/p is replaced by the clock pulse (CP) followed by the +ve edge detector ckt.

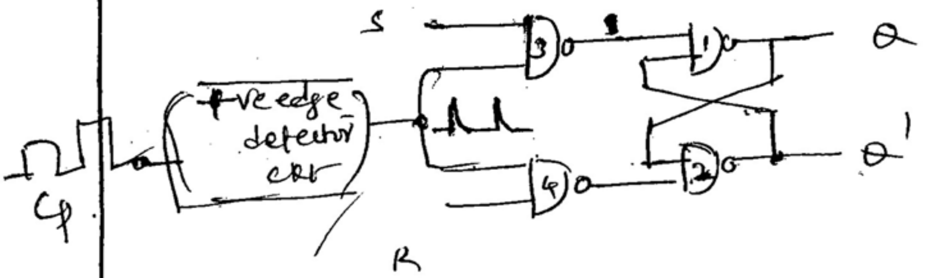
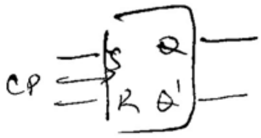


fig: Clocked SR flip flop.

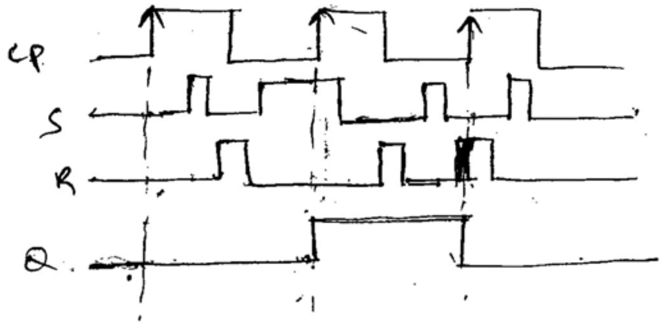
fig. below shows the logic symbol & truth table of clocked SR flip-flop.



ⓐ Logic Symbol

CP	S	R	Q_n	Q_{n+1}	State
↑	0	0	0	0	NC
↑	0	0	1	1	
↑	0	1	0	0	Reset
↑	0	1	1	0	
↑	1	0	0	1	set
↑	1	0	1	1	
↑	1	1	0	X	Indeterminate
↑	1	1	1	X	
0	X	X	0	0	NC
0	X	X	1	1	

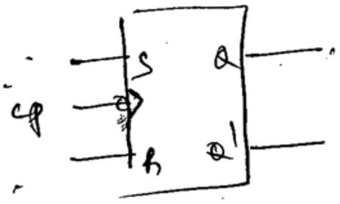
fig. below shows the i/p & o/p w/f's for a +ve edge triggered clocked SR ff.



Negative edge triggered SR flip-flop :-

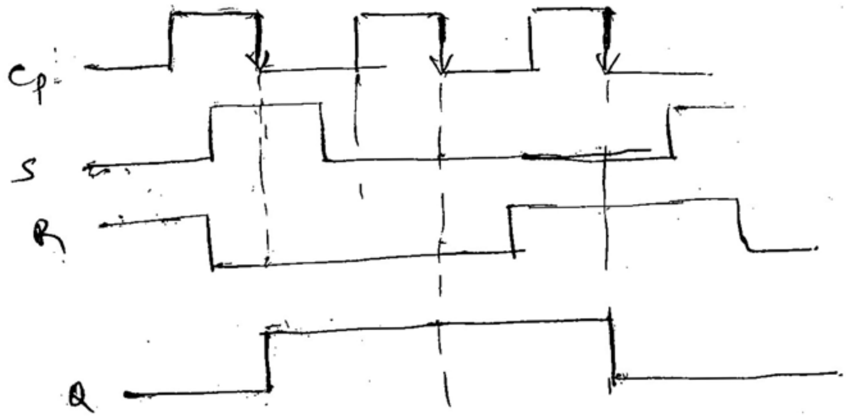
In negative edge triggered SR ff, the negative edge detector ckt is used & the ckt o/p responds at the -ve edge of the clock pulse.

fig. below shows the logic symbol & truth table of -ve edge triggered SR flip-flop.



ⓐ Logic symbol

CP	S	R	Qn	Qn+1	State.
↓	0	0	0	0	NC
↓	0	0	1	1	
↓	0	1	0	0	Reset
↓	0	1	1	0	
↓	1	0	0	1	Set
↓	1	0	1	1	
↓	1	1	0	x	Indeterminate
↓	1	1	1	x	
0	x	x	0		N.C
0	x	x	1		



Clocked D flip-flop

Like in D-latch, in D flip-flop the basic SR flip-flop is used with complemented i/p's. The D-flip flop is similar to D-latch except clock pulse followed by edge detector is used / instead of enable i/p. The edge triggered D flip-flop can be of 2 types.

1. +ve edge triggered D flip-flop.
2. -ve ''

+ve edge triggered D flip-flop:-

fig. below shows the +ve edge triggered D-flip-flop. It consists of a gated D latch & a +ve edge detector ckt..

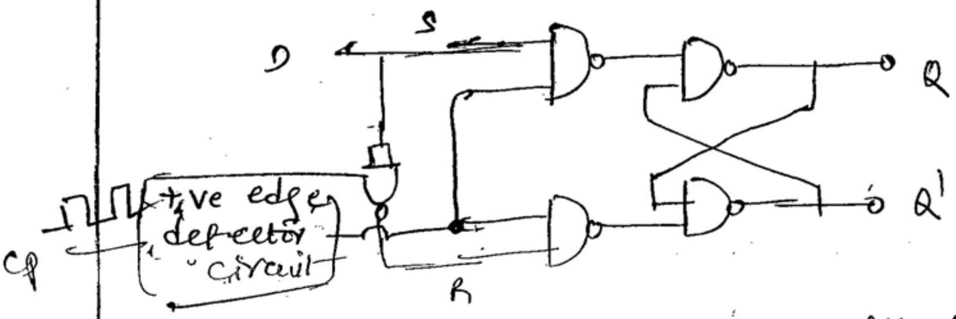
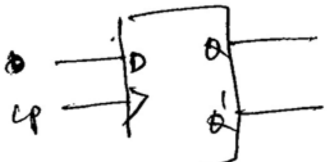


fig: +ve edge triggered D flip-flop.

fig. below shows the logic symbol, truth table & i/p & o/p w/f's for +ve edge triggered D. flip-flop.



ⓐ logic Symbol

Cp	D	Q_{n+1}
↑	0	0
↑	1	1
0	X	Q_n.

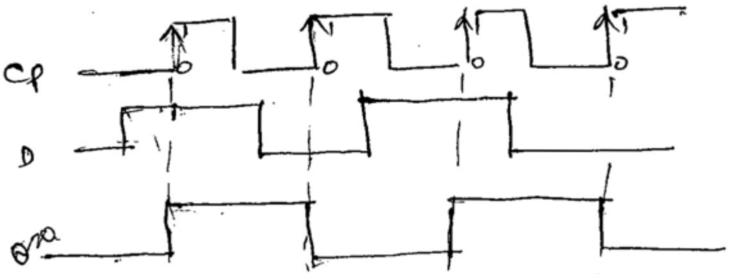


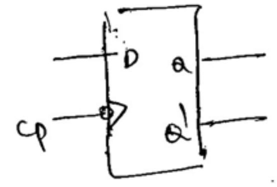
fig: I/p & o/p w/f's of clocked D flip-flop.

From truth table of D- flip-flop, we can realize that Q_{n+1} function follows D i/p at +ve edge of cp.
Hence the characteristic equation for D-flip-flop is

$$\boxed{Q_{n+1} = D.}$$

However, the o/p Q_{n+1} is delayed by one clock period. Thus D-flip-flop is also called delay flip-flop.

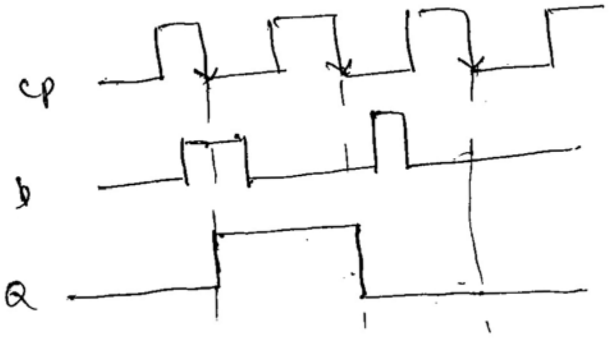
-ve edge triggered D-flip-flop :-



ⓐ Logic symbol

CP	D	Q_{n+1}
↓	0	0
↓	1	1
0	×	Q_n

ⓑ Truth table of D flip flop.



<u>Note</u>: cp stands for clock pulse.

Clocked JK flip-flop :-

The uncertainity in the state of an SR flip-flop when $S=R=1$ can be eliminated by converting it into a JK flip-flop. The data i/p's are J & K which are ANDed with Q' & Q respectively, to obtain S & R i/p's, as shown below.

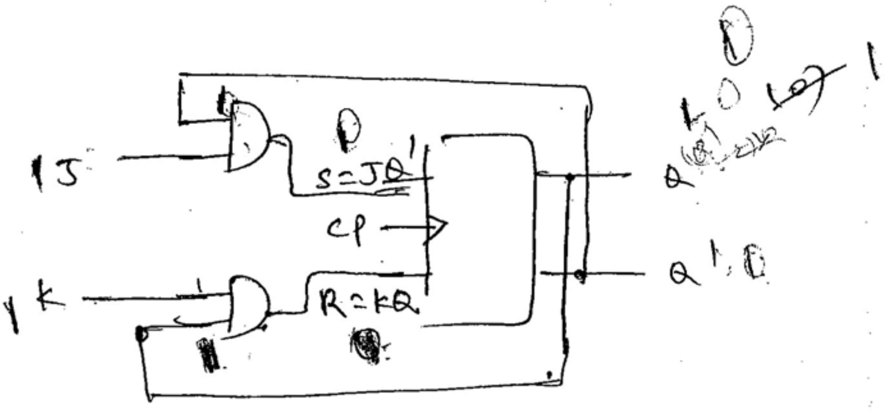


fig: JK flip-flop using SR flip-flop.

operation of JK flip flop :-

case 1: $J=k=0$.

when $J=k=0$, $S=R=0$ & according to truth table of SR flip-flop there is no change in the o/p.

→ when $J=k=0$, o/p doesnot change.

case 2: $J=0$ & $k=1$

$Q=0$, $Q'=1$: when $J=0$, $k=1$ & $Q=0$.

$S=0$ & $R=0$. Since $SR=00$, there is no change in the o/p & therefore, $Q=0$ & $Q'=1$.

$Q=1$, $Q'=0$: when $J=0$, $k=1$ & $Q=1$

$S=0$ & $R=1$. According to truth table of SR flip-flop it is a reset state & the o/p Q will be '0'.

→ when $J=0$ & $k=1$; makes $Q=0$, i.e reset state.

Case 3: $J=1$ & $K=0$.

$Q=0, Q'=1$: when $J=1$ & $K=0$, $S=1$ & $R=0$. Acc. to truth table of SR flip-flop it is set state & o/p Q will be 1.

$Q=1, Q'=0$: when $J=1$ & $K=0$. & $Q=1$, $S=0$ & $R=0$. Since $SR=00$, there is no change in o/p & therefore, $Q=1$ & $Q'=0$.

→ The i/p $J=1$ & $K=0$, makes $Q=1$, i.e set state.

Case 4: $J=K=1$,

$Q=0$; $Q'=1$: when $J=K=1$ & $Q=0$, $S=1$. & $R=0$: Acc. to truth table of SR flip-flop. it is a set state & the o/p Q will be 1.

$Q=1$, $Q'=0$: when $J=K=1$ & $Q=1$, $S=0$ & $R=1$ Acc. to truth table of SR flip-flop - it is a reset state & the o/p Q will be 0.

→ The i/p $J=K=1$, toggles the flip-flop o/p.

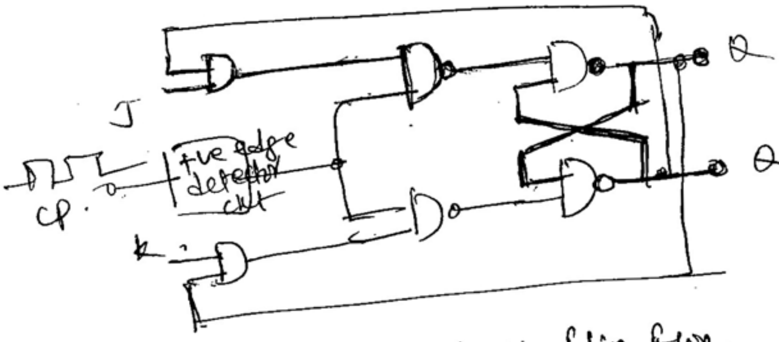


fig: clocked JK flip-flop.

logic symbol.

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

≡

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

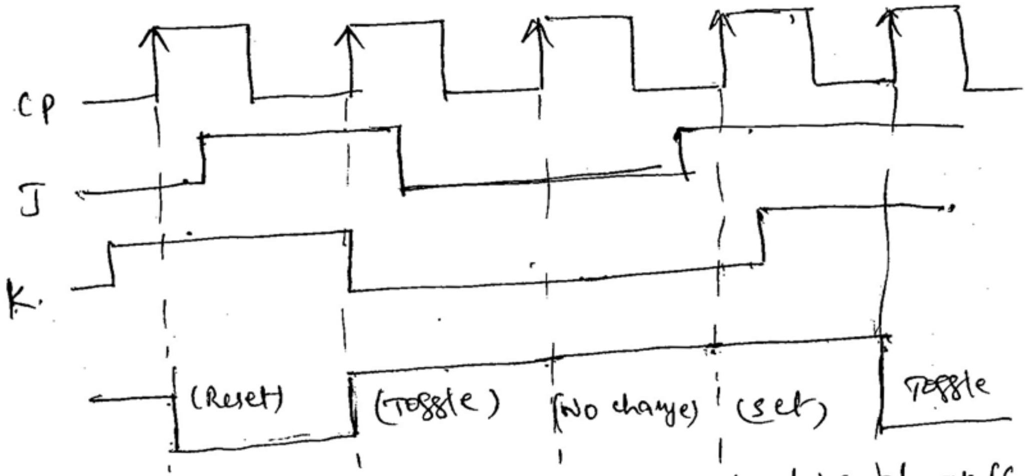
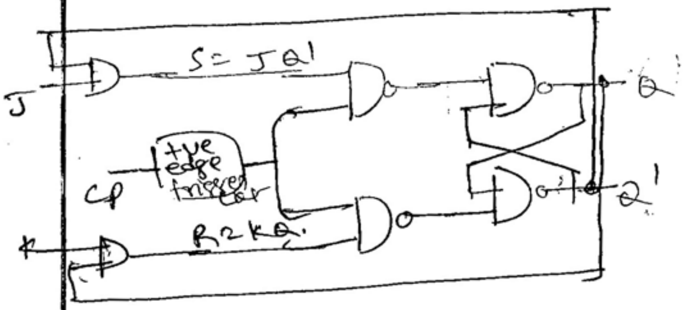


fig: I/p & O/p w/f's for +ve edge triggered JK ff.

JK flip-flop using NAND gates :-

JK flip-flop implemented using SR flip-flop & AND gates can also be performed by adding an extra i/p terminal to NAND gates 3 & 4.

modified ckt of JK flip-flop which has only NAND gates.



≡

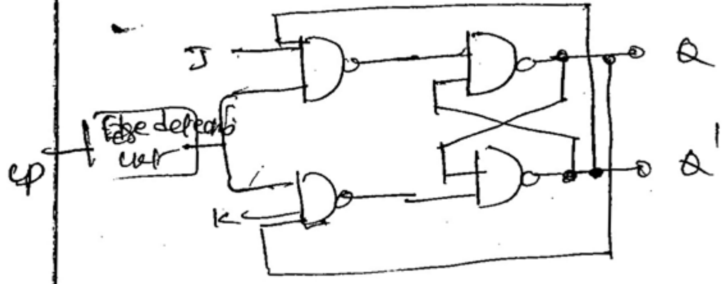


fig: JK flip-flop using NAND gates.

Race-Around condition :-

In JK flip-flop, when $J=K=1$, the o/p toggles (o/p changes either from 0 to 1 or from 1 to 0) consider that Initially $Q=0$, & $J=K=1$.

After a time Interval Δt equal to the propagation delay (t_p) through 2 NAND gates in series, the o/p will change to $Q=1$ & - after another time interval of Δt the o/p will change back to $Q=0$.

This toggling will continue untill the flip-flop is enabled & $J=K=1$. At the end of clock pulse the flip-flop is disabled & the value of Q is uncertain. This situation is referred to as Race-Around condition.

The Race-Around condition is shown in fig. below.

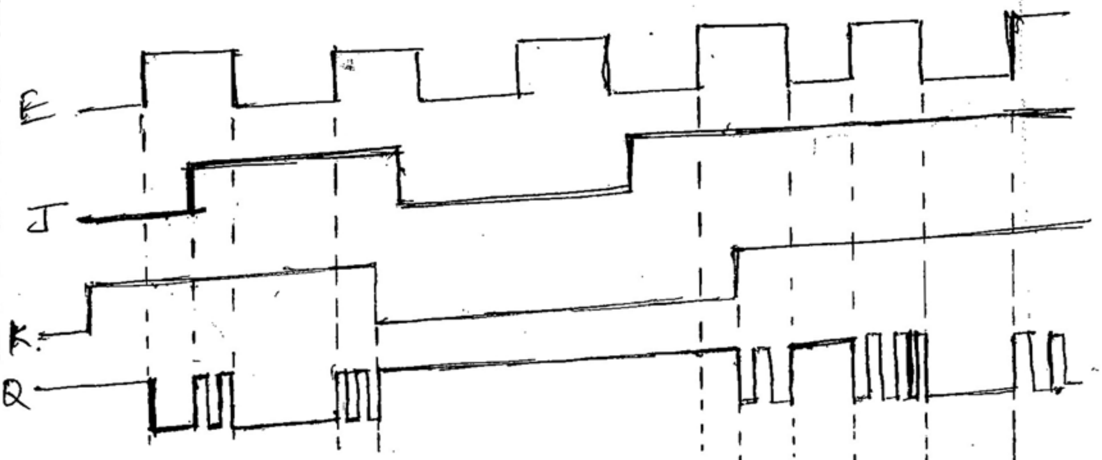


fig: I/p & o/p waveforms for clocked JK flip-flop.

Race-Around condition exists when $t_p > \Delta t$. By keeping $t_p < \Delta t$ we can avoid race-around condition.

We can keep $t_p < \Delta t$ by keeping the duration of edge less than Δt. A more practical method for overcomming this difficulty is the use of Master Slave configuration.

Master Slave flip-flops

Master Slave SR flip-flop

A master slave flip-flop is constructed from two flip-flops. one circuit serves as master & the other circuit serves as slave & the overall circuit is called master slave flip-flop.

fig below shows SR master slave flip-flop.

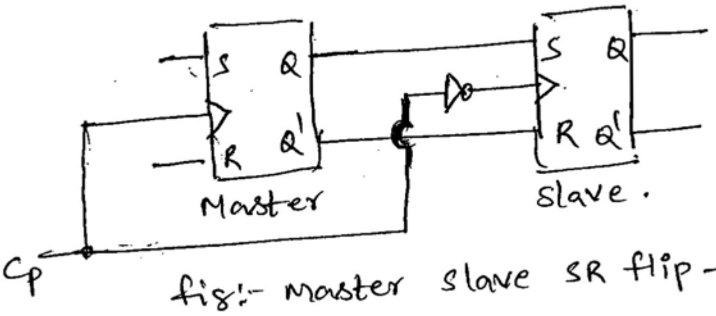


fig:- master slave SR flip-flop.

→ It consists of a master flip-flop, a slave flip-flop & an inverter. Both the flip-flops are positive level triggered, but inverter connected at the clock i/p of slave flip-flop forces it to trigger at the negative level.

The output of Master flip-flop is determined by the S & R i/p's at the positive clock pulse.

The o/p state of master is then transferred as an i/p to slave flip-flop. The slave flip-flop uses this i/p at the negative clock pulse to determine its o/p state.

The operation of master slave SR flip-flop is as shown in fig below.

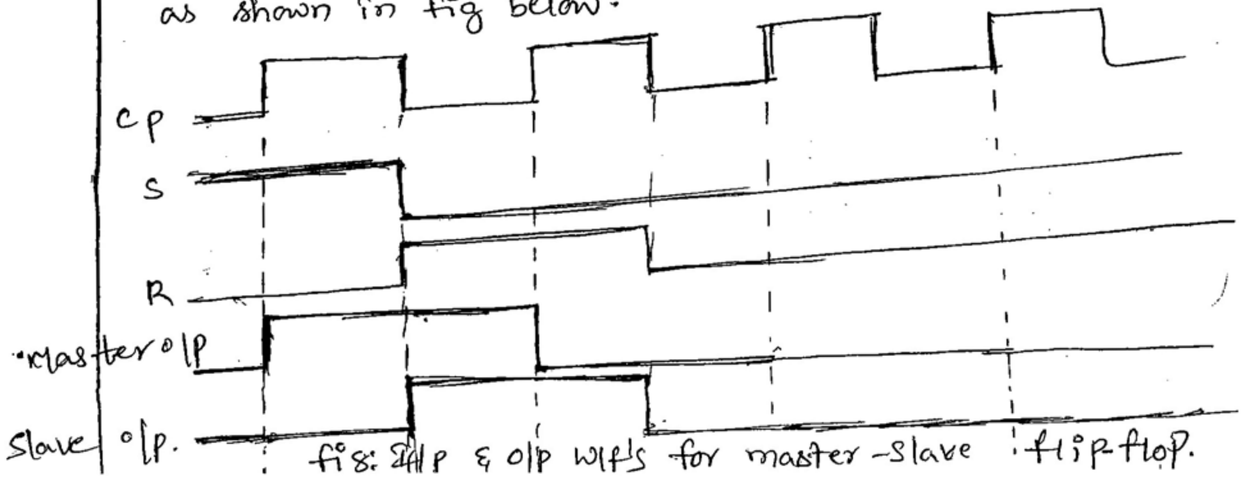


fig: i/p & o/p wif's for master-slave flip-flop.

Master Slave JK flip-flop:-

The master slave flip-flop can be constructed for any type of flip-flop.

fig. below shows one way to build a JK master-slave flip-flop. It consists of clocked JK flip-flop as a master & clocked SR flip-flop as a slave.

Like SR master slave, the o/p of master flip-flop is fed as an i/p to slave flip-flop.

The clock signal is connected directly to master flip flop, but it is connected through an inverter to slave flip-flop as shown in fig below.

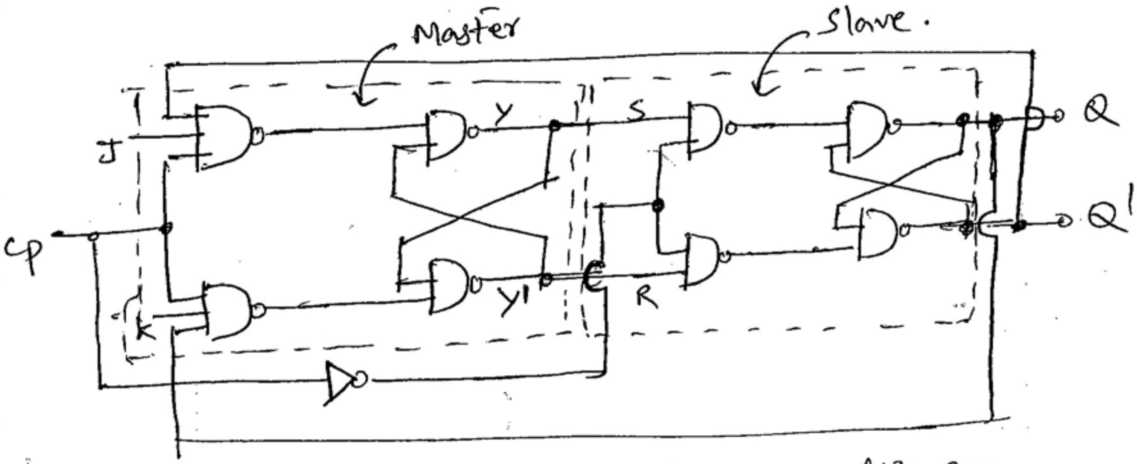


fig: Master slave JK flip-flop.

→ The information present at the J & K i/p's is transmitted to the o/p of master flip-flop on positive - clock pulse & it is held there untill the negative clock pulse occurs, after which it is allowed to pass through to the o/p of slave flip-flop. The o/p of slave flip-flop is connected as a third i/p of the master JK flip-flop.

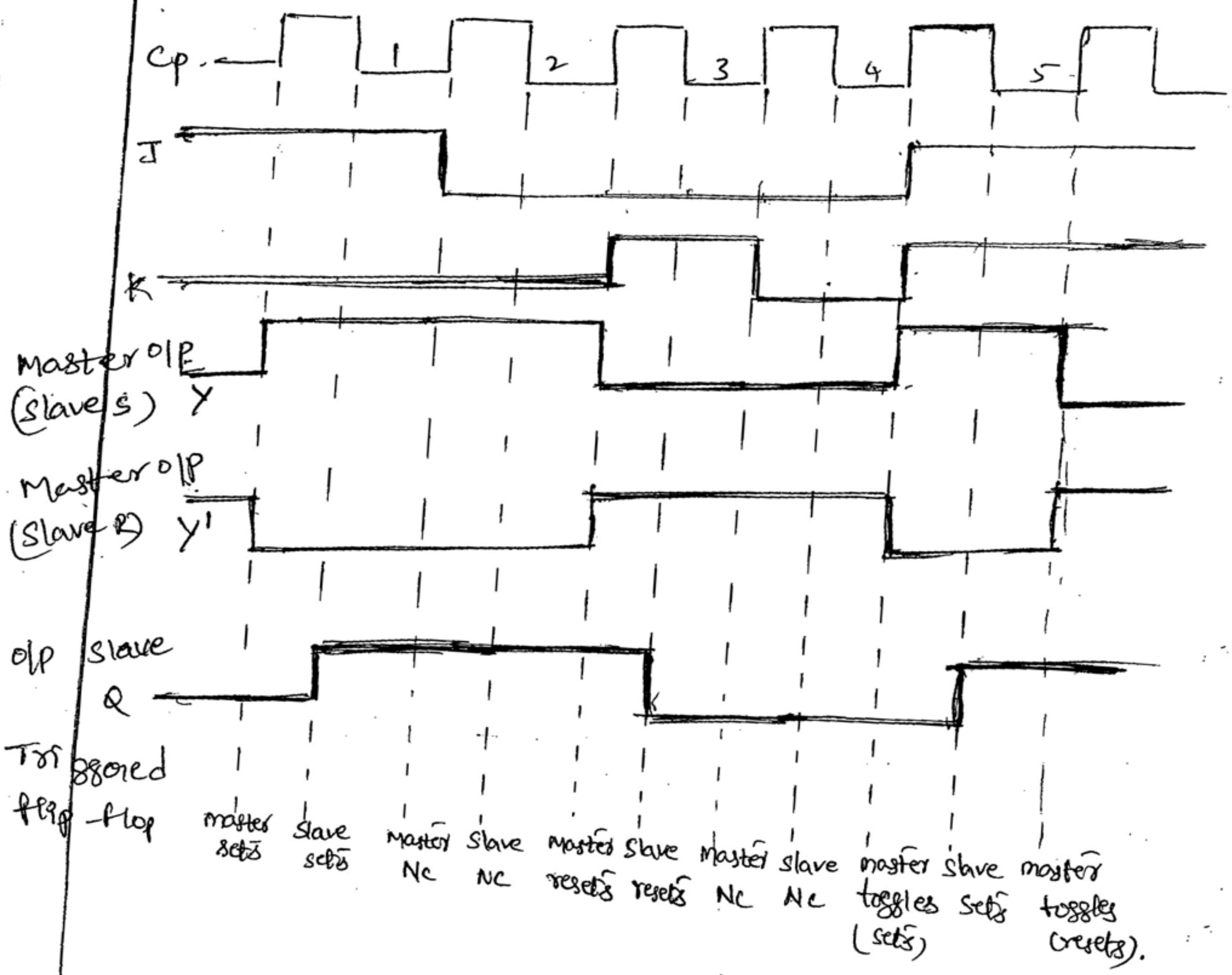
→ when J=1 & K=0, the master sets on positive clock the high 'y' o/p of master drives the 's' i/p of the slave, so at negative clock, slave sets, copying the action of Master.

→ when J=0 & K=1, the master resets on the positive clock. The high Y' o/p of master goes to 'R' i/p of slave. Therefore, at the negative clock slave resets, again copying the action of master.

→ when J=1 & K=1, master toggles on positive clock & slave then copies the o/p of master on the negative clock. At this instant, feedback i/p's to the master flip-flop are complemented but as it is negative half of the clock pulse master flip-flop is inactive.

This prevents race-around condition.

fig. below shows i/p & o/p waveforms of master slave JK flip-flop.



Clocked T flip-flop :-

T flip-flop is also known as "Toggle flip-flop". The T flip-flop is a modification of JK flip-flop. the T flip-flop is obtained from a JK flip-flop by connecting both i/p's, J & K together.

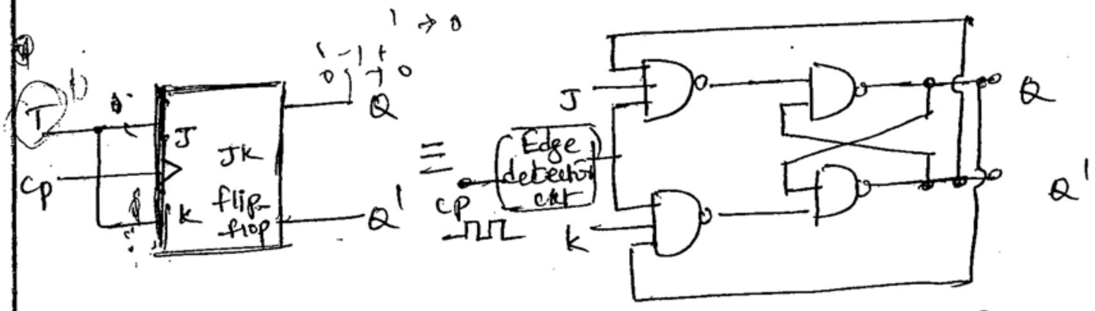


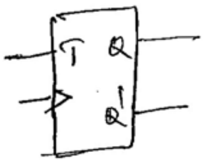
fig: T-flip-flop using NAND gates.

operation of T-flip-flop:

→ when T=0, J=K=0 & hence there is no change in the o/p.

→ when T=1, J=K=1 & hence o/p toggles.

The fig below shows Logic symbol, truth table & the characteristic equation for T-flip-flop.



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

≡

T	Q_{n+1}
0	Q_n
1	Q_n'

Q_n \ T	T'	T
Q_n'	0	1
Q_n	1	0

$$\therefore Q_{n+1} = TQ_n' + T'Q_n$$

ⓐ Logic symbol ⓑ truth table. ⓒ characteristic equation.

Characteristic equations of flip-flops :-

flip-flop	characteristic equation
SR	$Q_{n+1} = S + R' Q_n$
D	$Q_{n+1} = D$.
JK	$Q_{n+1} = J Q_n' + K Q_n$
T	$Q_{n+1} = T Q_n' + T' Q_n$.

Various representation of flip-flops :-

Flip-flop as ~~state~~ Finite state machine :-

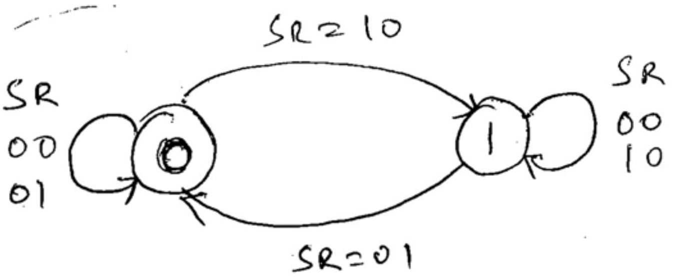
In FSM, the functional behaviour of the circuit is represented using state transition diagram.

The representation of flip-flop in their state transition diagrams is shown below.

SR flip-flop :-

S	R	Q_n	Q_{n+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

ⓐ

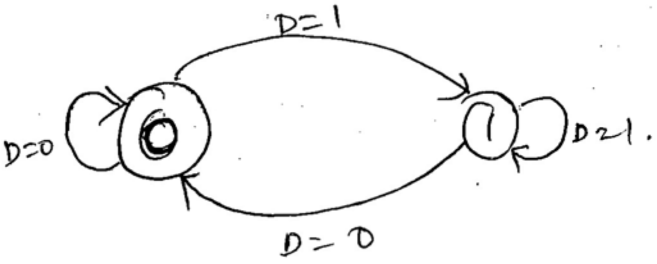


ⓑ.

fig:— ⓐ Truth table . ⓑ state transition diagram

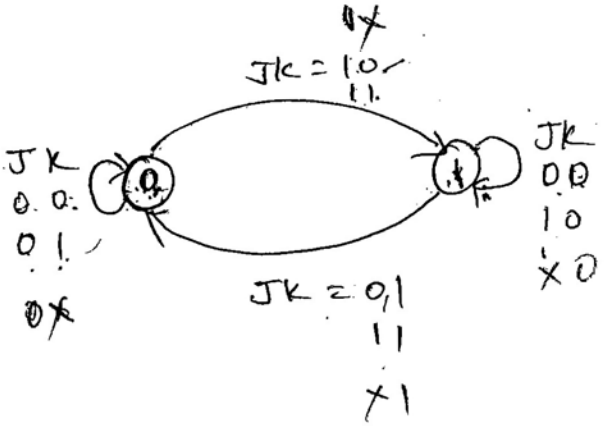
D - flip-flop :-

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



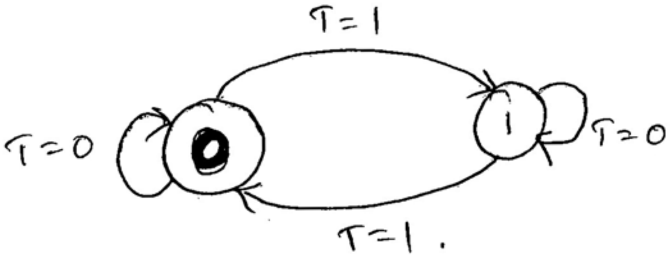
JK flip-flop :-

J	K	Q_n	Q_{n+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



T - flip-flop :-

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



Flip-flop excitation table

During the design process we know, from the transition table, the sequence of states, i.e the transition from each present state to its corresponding next state. From this, we can find the flip-flop i/p conditions that causes the required transition.

The table that lists the required i/p condition for a given change of State is known as excitation table.

→ The excitation table consists of two columns Qn & Qn+1 & a column for each input to show how the required transition can be achieved.

RS flip-flop :-

R	S	Qn+1
0	0	Qn
0	1	1
1	0	0
1	1	X

ⓐ RS truth table.

Qn	Qn+1	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

ⓑ. Excitation table.

JK flip-flop :-

J	K	Qn+1
0	0	Qn
0	1	0
1	0	1
1	1	Qn'

JK truth table

Qn	Qn+1	J	K.
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation table.

D flip-flop

D	Qn	Qn+1
0	0	0
0	1	0
1	0	1
1	1	1

fig: D-flip-flop truth table

Qn	Qn+1	D
0	0	0
0	1	1
1	0	0
1	1	1

fig: Excitation table

T flip-flop :-

T	Qn	Qn+1
0	0	0
0	1	1
1	0	1
1	1	0

fig: T flip-flop truth table

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0.

fig: T flip-flop excitation table.

Conversion of flip-flops:-

It is possible to convert one flip-flop into another flip-flop with some additional gates or simply doing some extra connections.

SR flip-flop to D flip-flop:

The excitation table for above conversion is shown in table below.

I/P	present state	Next state	flip-flop I/pls.	
D	Qn	Qn+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 simplification:

D	Qn'	Qn
D'	0	0
D.	1	X

S = D

D	Qn'	Qn
D'	X	1
D.	0	0

R = D'.

Logic diagram

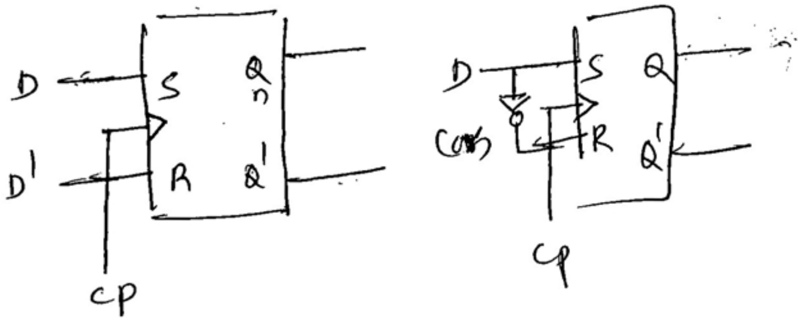


fig: SR to D flip-flop conversion.

SR flip-flop to JK flip-flop:-

The excitation table for above conversion is shown in table below.

J	K	Q_n	Q_{n+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

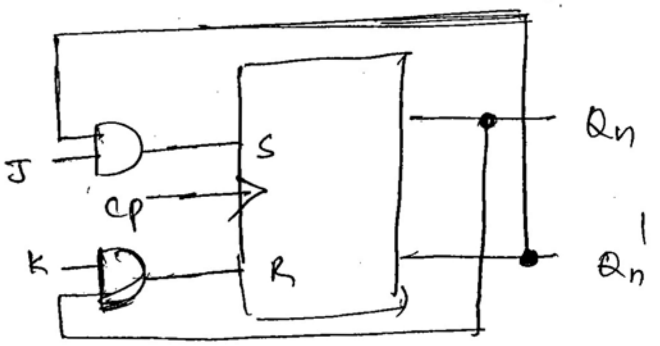
K-map simplification:

J \ KQ_n	$K'Q_n'$	$K'Q_n$	KQ_n	KQ_n'
J'	0	X	0	0
J	1	X	0	1

$S = JQ_n'$

J \ KQ_n	$K'Q_n'$	$K'Q_n$	KQ_n	KQ_n'
J'	X	0	1	X
J	0	0	1	0

$R = KQ_n$.



SR flipflop to T flip-flop:

I/p	Present state	next state	Flip-flop i/p's	
T	Q_n	Q_{n+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 simplification:-

T \ Q_n	Q_n'	Q_n
T'	0	X
T	1	0

$S = T Q_n'$

T \ Q_n	Q_n'	Q_n
T'	X	0
T	0	1

$R = T Q_n.$

Logic dgm:-

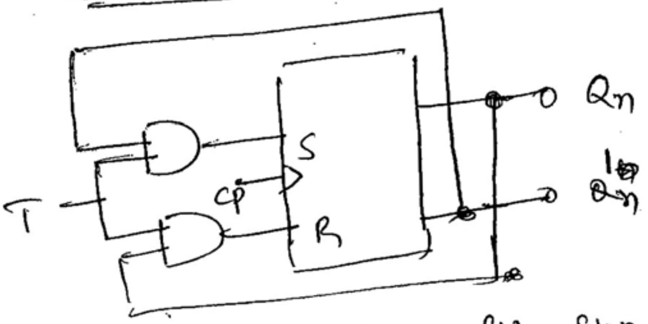
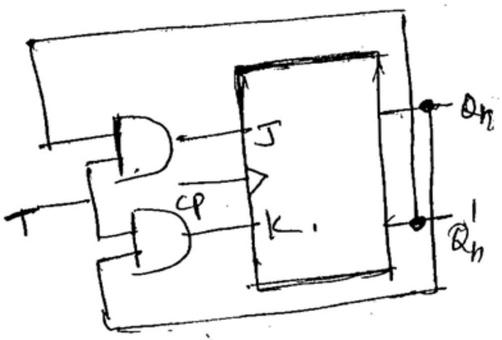


fig: SR to T flip-flop.

JK flip-flop to T flip-flop:

T	Q_n	Q_{n+1}	J	K.
0	0	0	0	X
0	1	1	X	0.
1	0	1	1	0.
1	1	0	0	1



K-map simplification.

T \ Q_n	Q_n'	Q_n
T'	0	X
T	1	0

$J = T Q_n'$

T \ Q_n	Q_n'	Q_n
T'	X	0
T	0	1

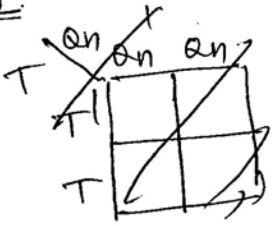
$K = T Q_n$

D flip-flop to T flip-flop :-

T	Qn	Qn+1	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

K-map simplification :-

T \ Qn	Qn'	Qn
T'	0	1
T	1	0



$D = T Q_n' + T' Q_n$.

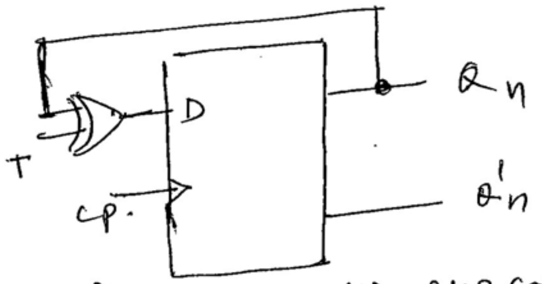


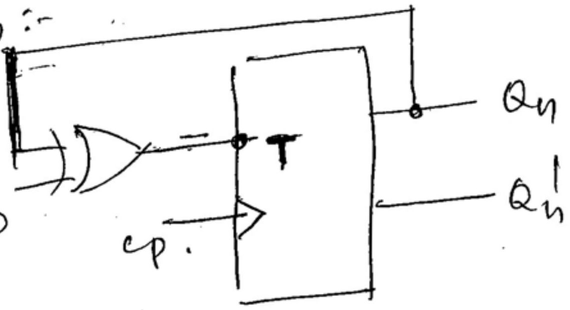
fig: D to T flip-flop conversion.

T flip-flop to D flip-flop:

D	Qn	Qn+1	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

K-map simplification :-

D \ Qn	Qn'	Qn
D'	0	1
D	1	0



$T = D Q_n' + D' Q_n$

Registers :-

A flip-flop can store 1-bit of binary information

Definition:

A register is a group of flip-flops which can store a group of binary information.

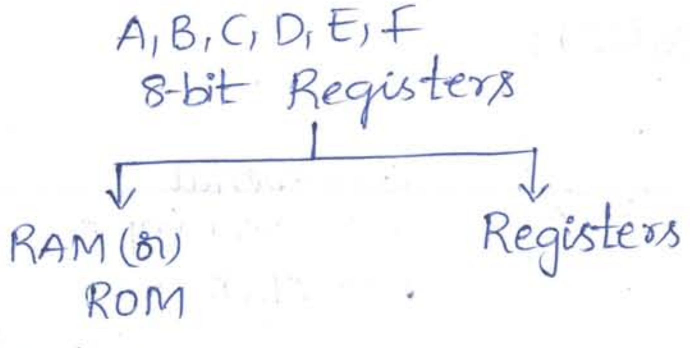
→ n-bit Registers consists of n, number of flip-flops, it can store n-bit of binary information

→ Available IC's :-

4-bit Registers & 8-bit Registers, IC's which are used in microprocessors i.e. A, B, C, D, E, F

Applications :-

→ Used as general purpose register in microprocessor. 8085, 8086



Shift Register :-

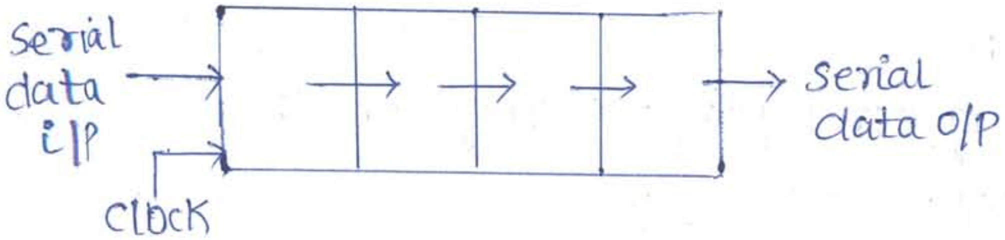
A register capable of shifting the binary information held in each cell to its neighbouring cells, in a selected direction is called "shift Register".

Types of shift Register:

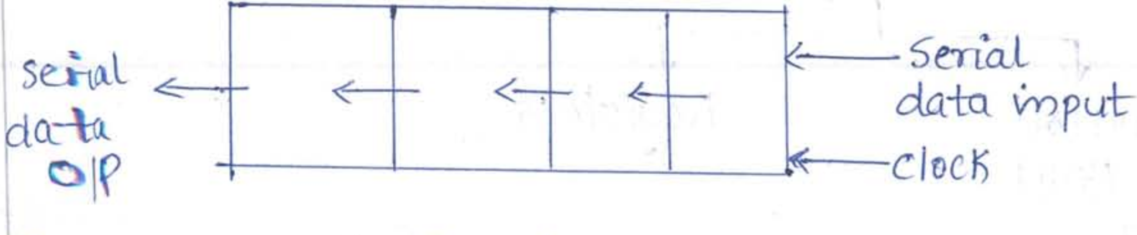
(1) Serial in Serial out (SISO) shift Register
(2) Serial in Parallel out (SIPO) shift Register
(3) Parallel in Serial out (PISO) shift Register.
(4) Parallel in parallel out (PIPO) shift Register.
(5) Bidirectional shift Register
(6) Universal shift Register

(1) SISO shift register:-

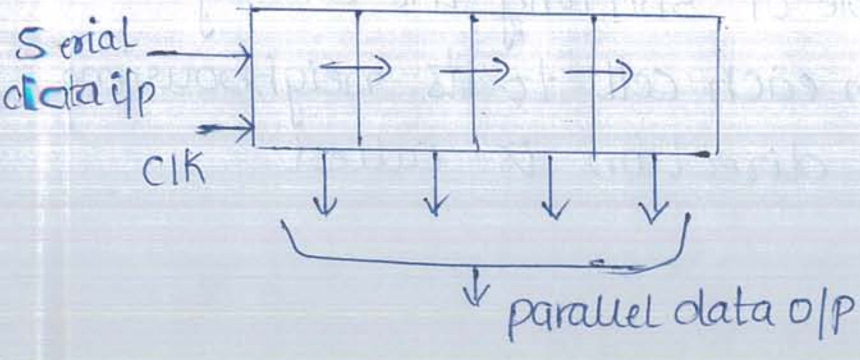
(a) SISO Right shift register;



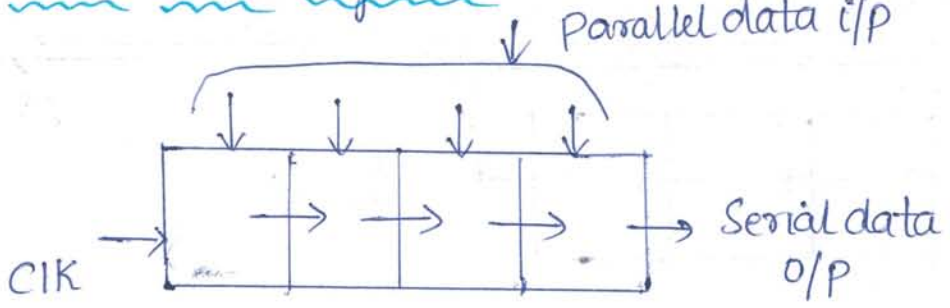
(b) SISO Left shift Register;



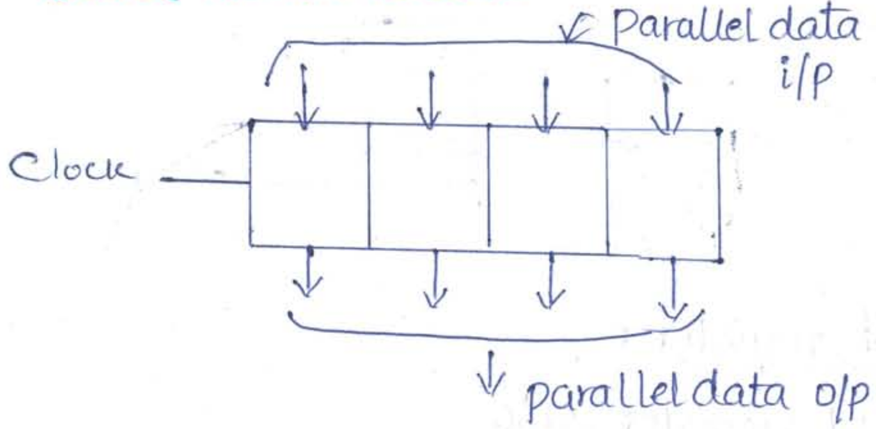
(2) SIPO shift Register:-



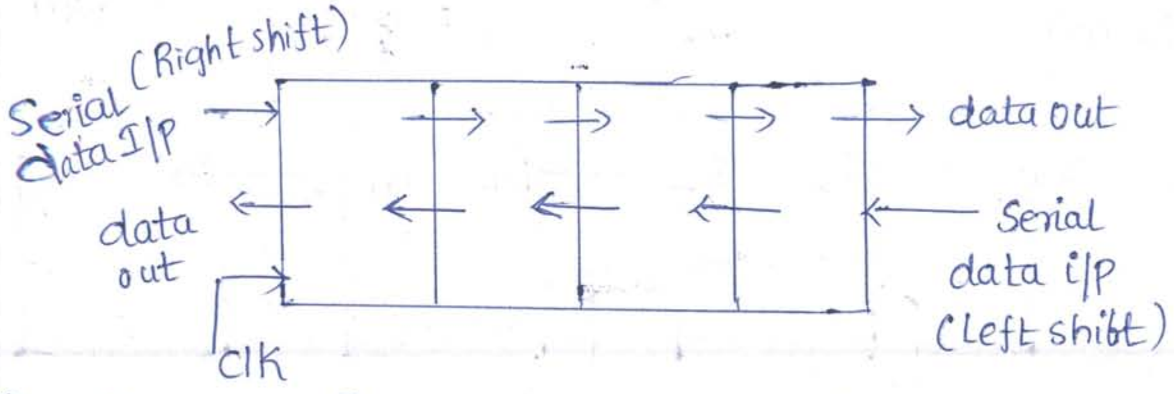
(3) PISO Shift Register :-



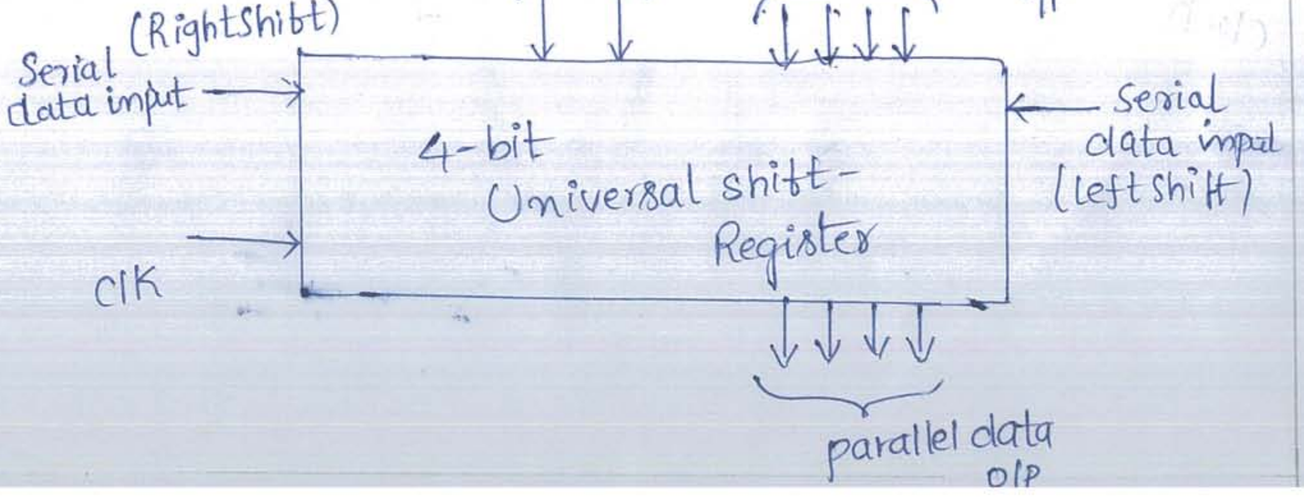
(4) PIPO Shift Register :- (General register)



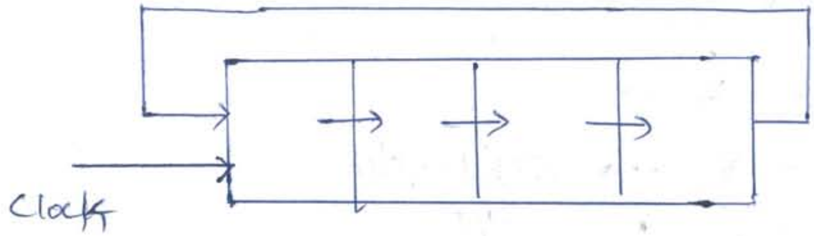
(5) Bi-directional shift Register :-



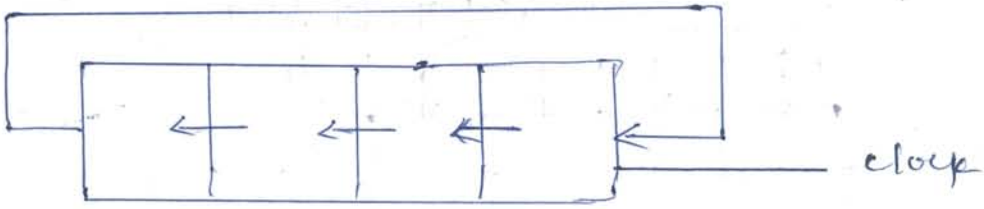
(6) Universal Register :-



(7) Rotate Right shift :-

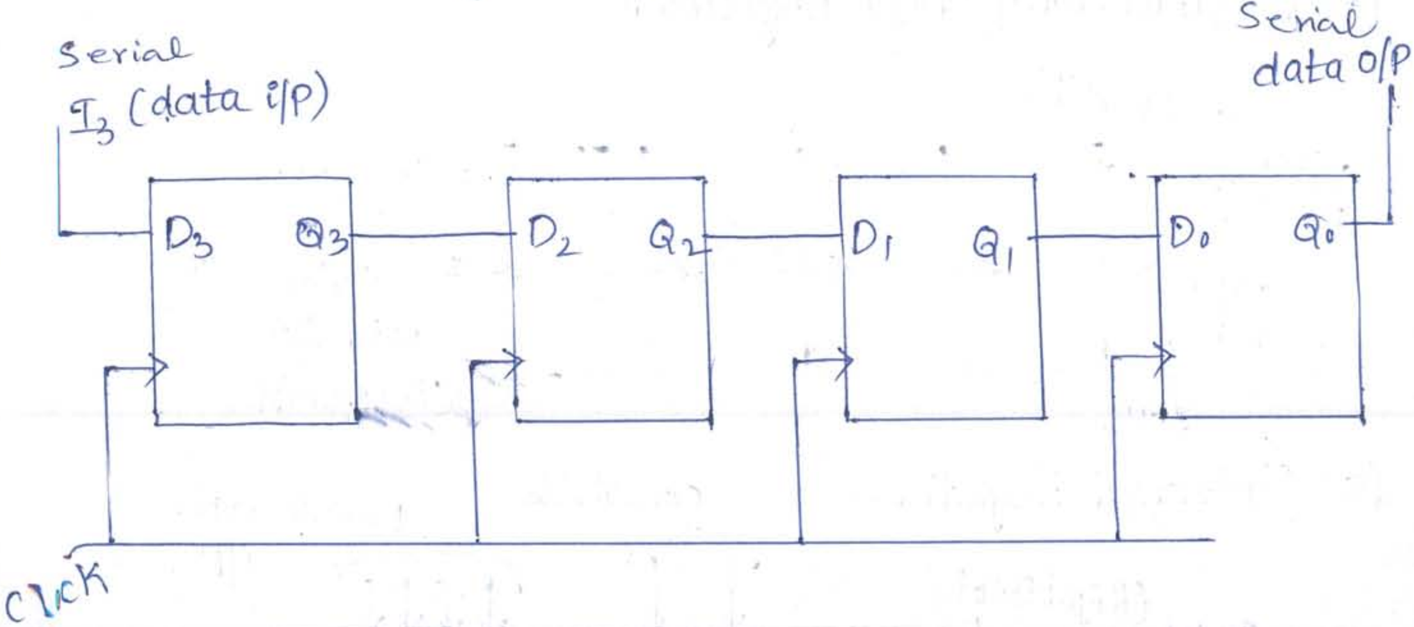


(8) Rotate Right shift :-



(1)(i) SISO shift right register :-

4-bit shift right register SISO

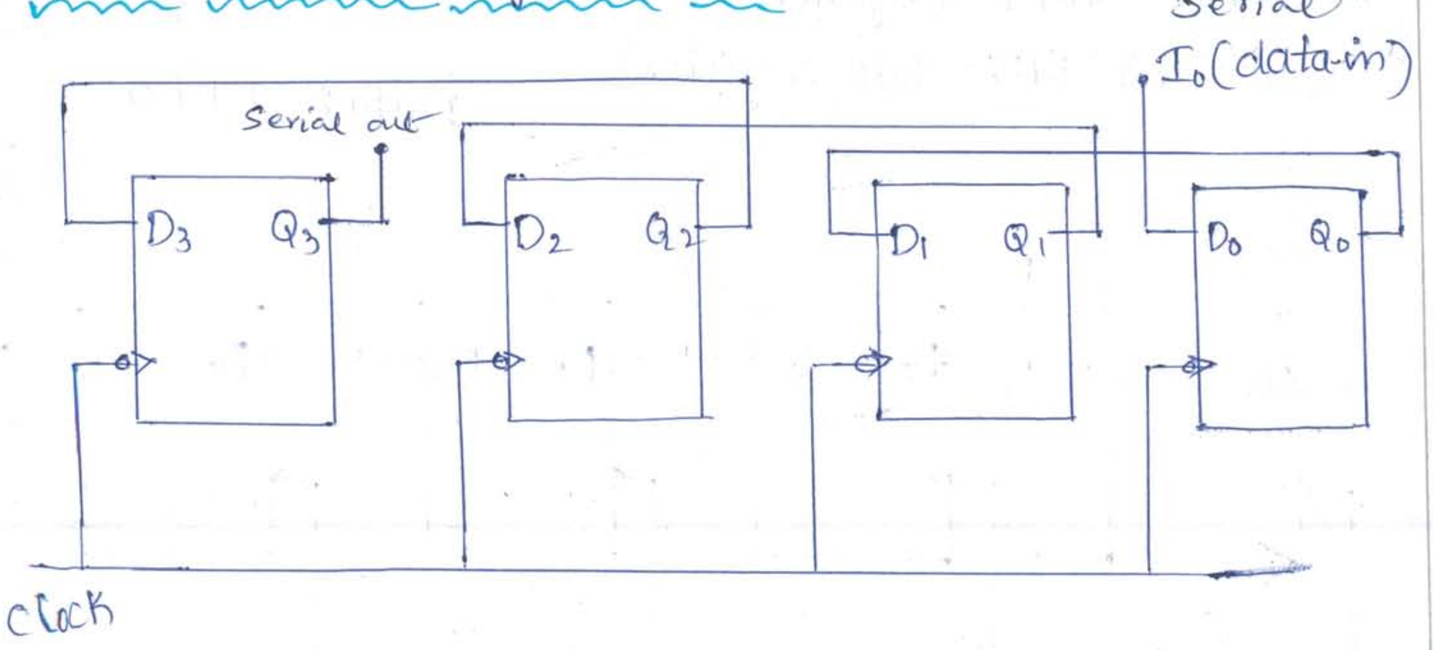


Operation :-

Data :- 1 0 1 0
↑ (MSB) ↑ (LSB)

clock	Data-in	Q_3	Q_2	Q_1	Q_0	
0	—	0	0	0	0	
↑ 1	0	0	0	0	0	
↑ 2	1	1	0	0	0	
↑ 3	0	0	1	0	0	
↑ 4	1	1	0	1	0	0
↑ 5	0	0	1	0	1	1
↑ 6	0	0	0	1	0	0
↑ 7	0	0	0	0	1	1

(ii) 4-bit Shift left register SISO :-



Operation :-

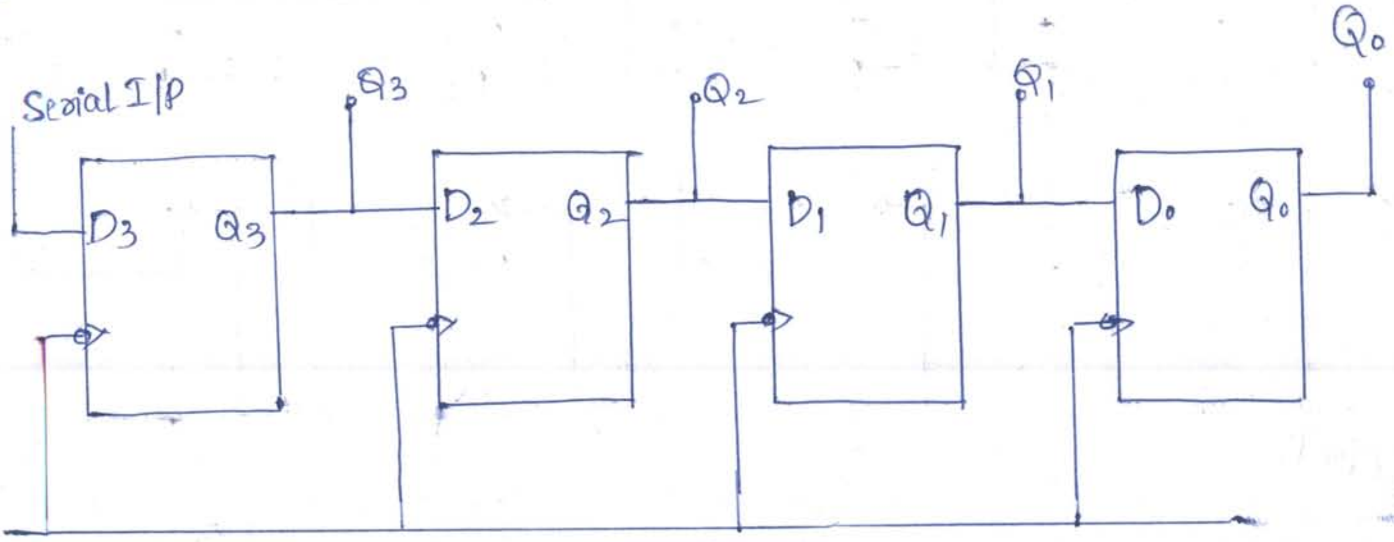
Data: 1010
↑
MSB

Clock		Q_3	Q_2	Q_1	Q_0	Data-in
0		0	0	0	0	—
↓ 1		0	0	0	1	1
↓ 2		0	0	1	0	0
↓ 3		0	1	0	1	1
↓ 4	$\boxed{1}$ ←	1	0	1	0	0
↓ 5	0 ←	0	1	0	0	0
↓ 6	1 ←	1	0	0	0	0
↓ 7	0 ←	0	0	0	0	0

(2) SIPO Shift Register :-

(4-bit SIPO shift register)

Data :- 1110



Operation :-

Data = 1110

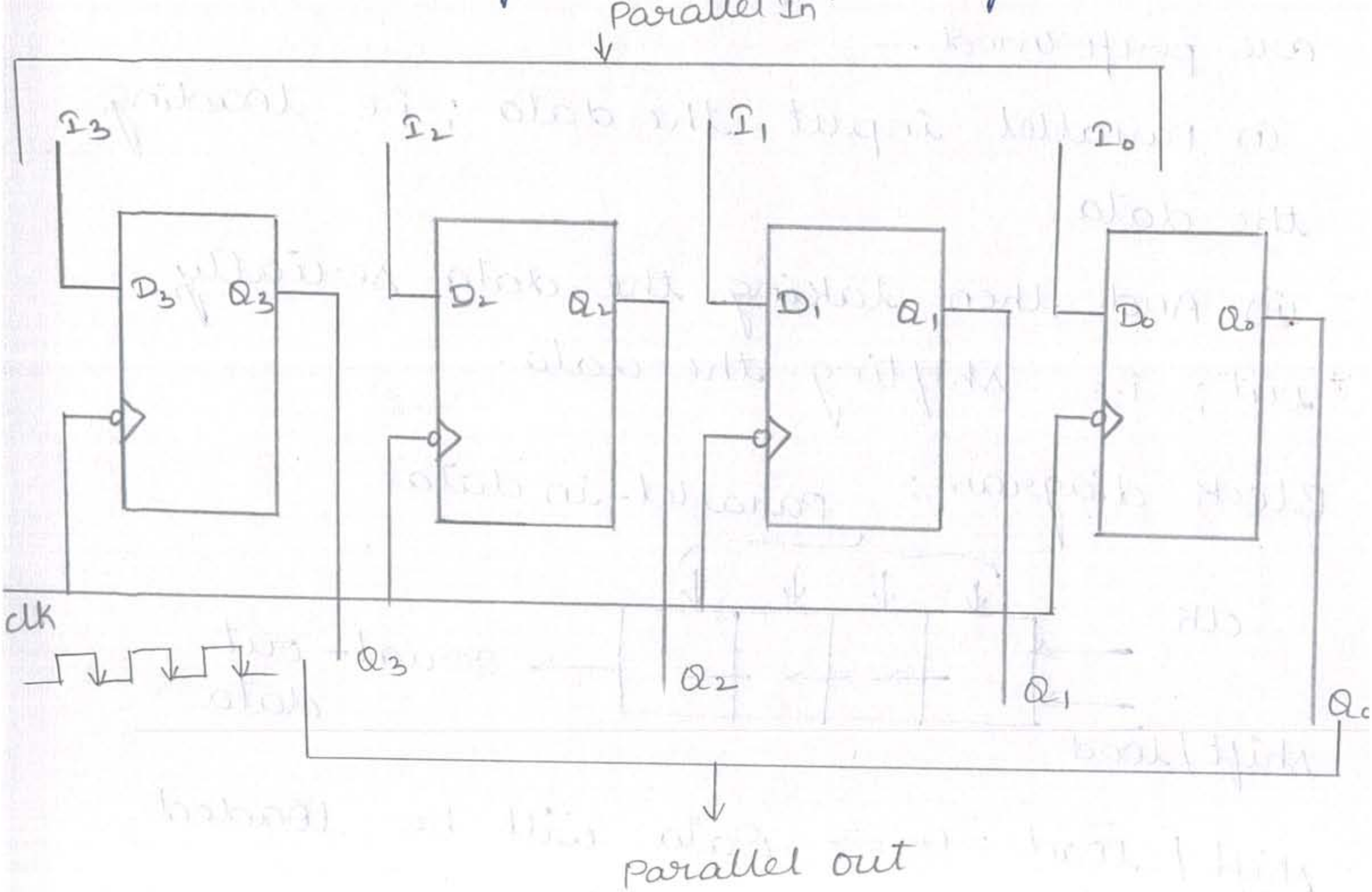
Clock	Din	Q_3	Q_2	Q_1	Q_0
0	—	0	0	0	0
↓ 1	0	0	0	0	0
↓ 2	1	1	0	0	0
↓ 3	1	1	1	0	0
↓ 4	1	1	1	1	0

$\boxed{1 \quad 1 \quad 1 \quad 0}$ → parallel output

→ After 4 clock pulses, the serial data is entered & After 4th clock pulse, the data can be taken out from all the flipflop o/p's i.e Q_3 Q_2 Q_1 Q_0.

3) PIPO shift register:

(Buffer register / storage register)



Data : 1100

operation:

clk	I3 I2 I1 I0	Q3 Q2 Q1 Q0
0	–	0 0 0 0
↓1	1100	1 1 0 0

Application:

1. storage register
2. Acts as normal register.

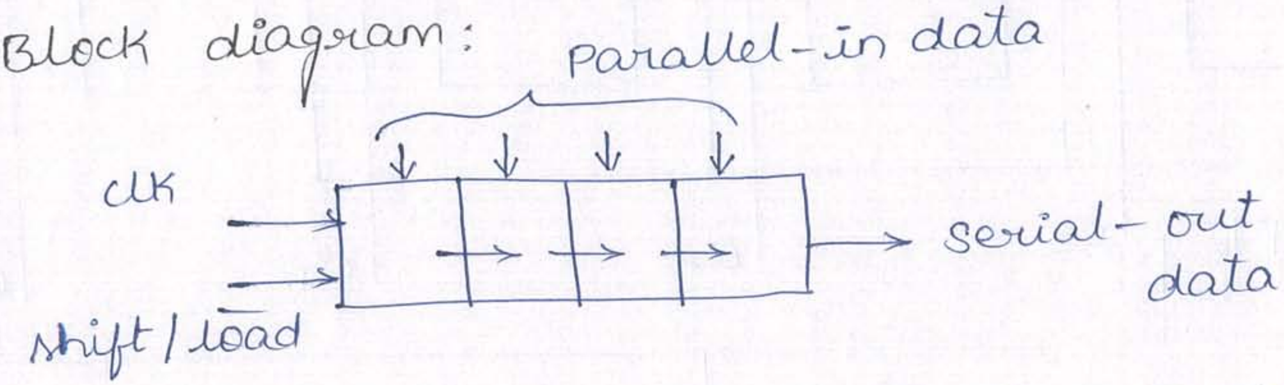
)PISO shift register:

→ In PISO shift register, two operations are performed.

(i) Parallel input the data ; i.e loading the data.

(ii) And then taking the data serially out ; i.e shifting the data.

Block diagram:



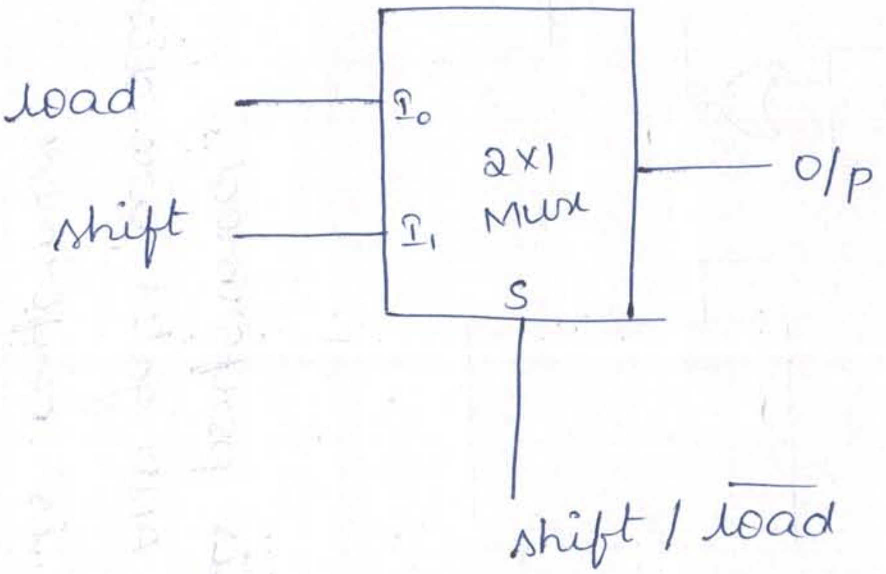
shift / $\overline{load}$ = 0 ⇒ Data will be loaded parallel into the register.

shift / $\overline{load}$ = 1 ⇒ Data will be shifted towards right.

→ Two operations are to be performed i.e first loading the data and then shifting the data to obtain serial o/p.

So, we need to design a circuit that performs both the operations at a time.

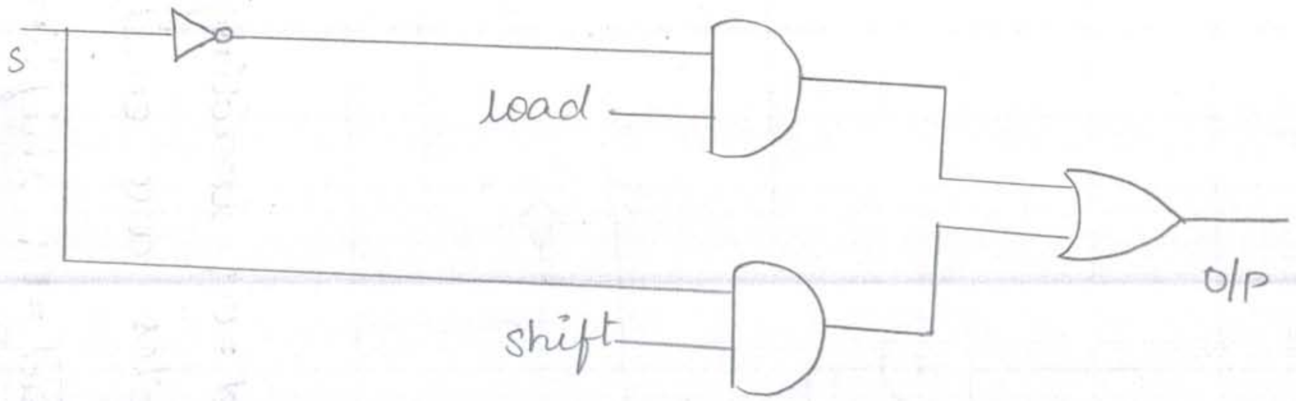
→ The multiplexer can be used to perform the above operation.



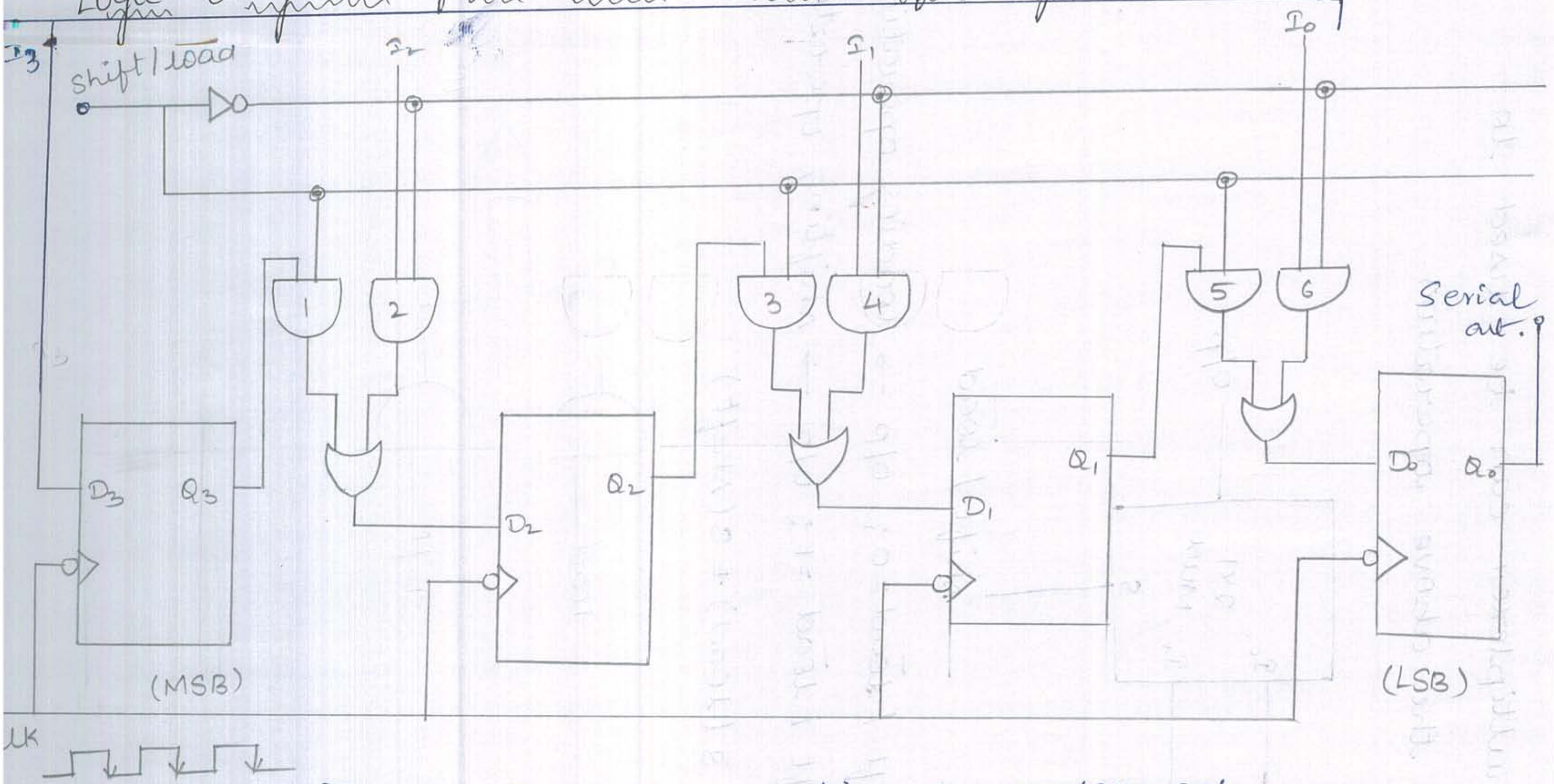
S=0 ⇒ shift / $\overline{\text{load}}$ = 0 ; O/p → loading operation

S=1 ⇒ shift / $\overline{\text{load}}$ = 1 ; O/p → shifting operatio

$$O/p = \overline{S}(\text{load}) + S(\text{shift})$$



Logic diagram for 4-bit PISO shift register:



→ When shift / $\overline{load}$ = 0; loading operation is performed.
2,4,6 AND gates are enabled ; 1,3,5 AND gates are disabled.

→ When shift / $\overline{load}$ = 1; shifting operation is performed.
2,4,6 AND gates are disabled; 1,3,5 AND gates are enabled.

Operation:

Data : 1100 ← LSB
↗ MSB

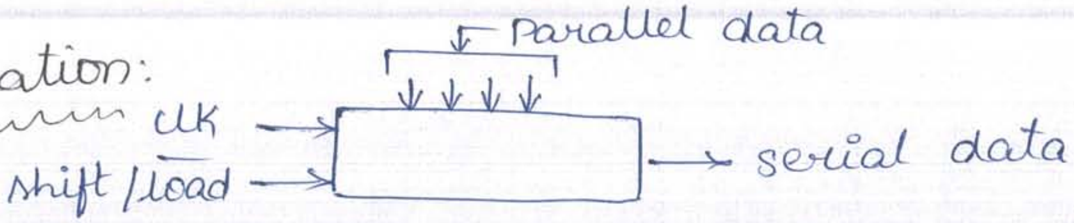
clk	shift/ load	parallel data $I_3 I_2 I_1 I_0$	$Q_3 Q_2 Q_1 Q_0$	serial O/P
0	–	–	0 0 0 0	
↓1	0	1 1 0 0	1 1 0 0	0
↓2	1	–	1 1 1 0	0
↓3	1	–	1 1 1 1	1
↓4	1	–	1 1 1 1	1

parallel i/p

serial data

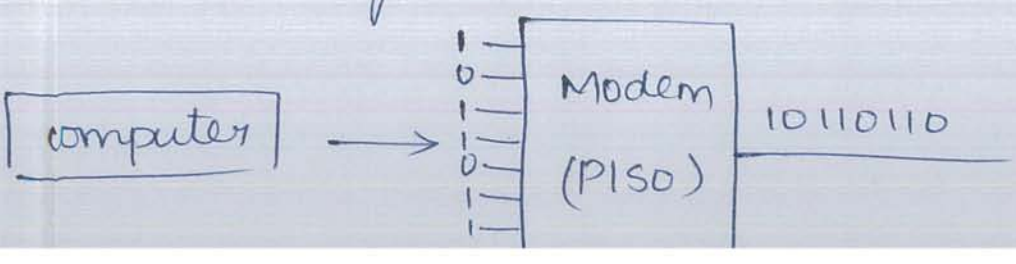
Total no. of clock pulses required to perform PISO = 4 i.e 1 clock to load and 3 clocks to take serial O/P.

Application:



→ parallel to serial conversion in telephone mo

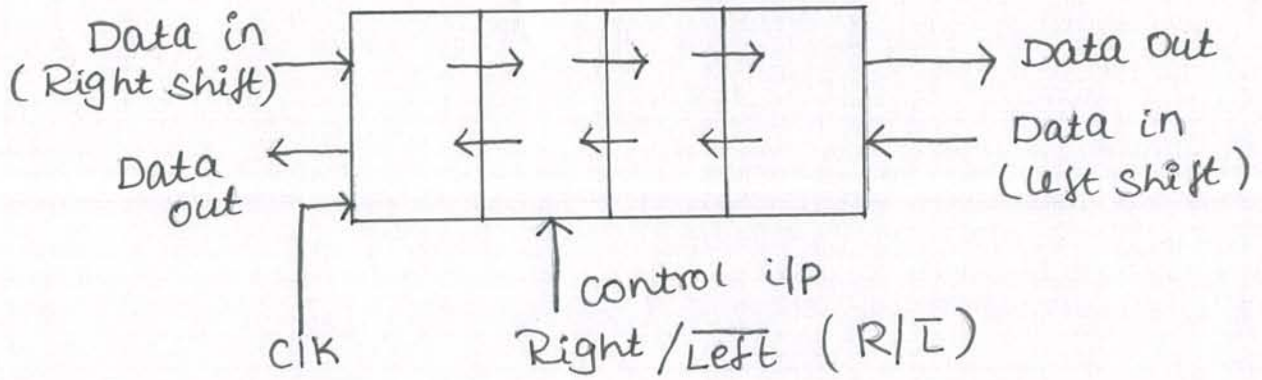
→ Used as registers in micro processors.



BI DIRECTIONAL SHIFT REGISTER :

In Bidirectional shift Register, the binary data can be shifted from Right to left (or) from left to Right.

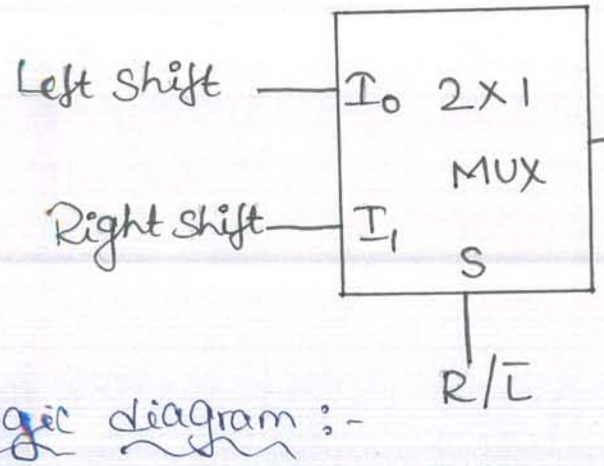
But one shift can be done at a time.



↳ If $R/\overline{L} = 0$; Data will be shifted to left

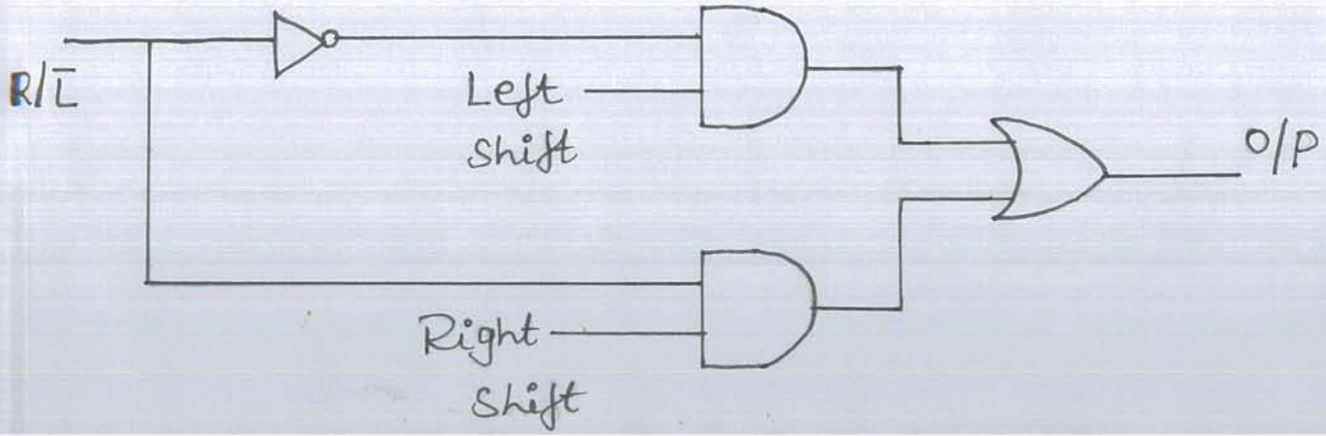
↳ If $R/\overline{L} = 1$; Data will be shifted to Right.

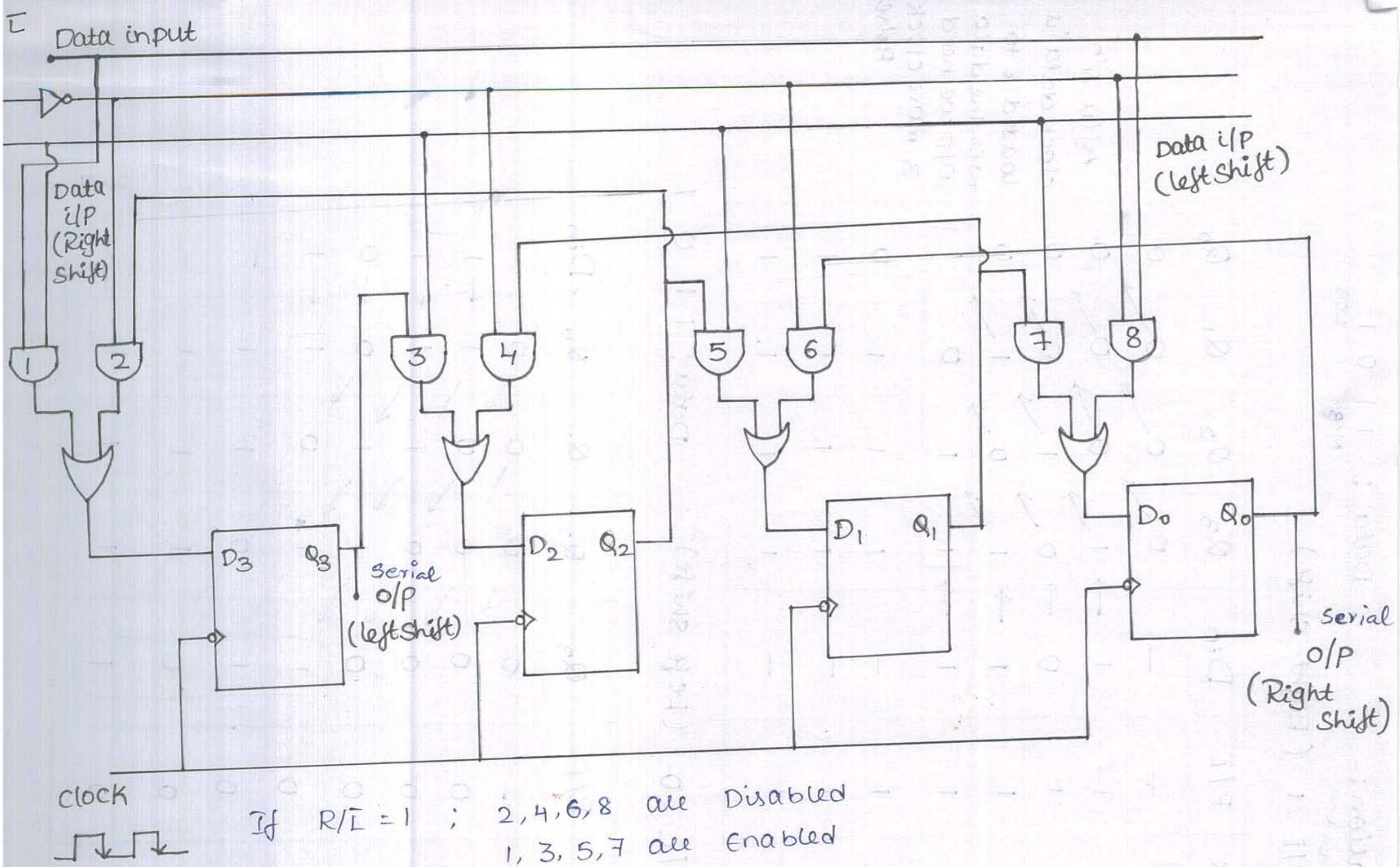
A multiplexer is used to perform one operation at a time.



$$o/p = (R/\overline{L})' \text{ left shift} + (R/\overline{L}) \text{ Right shift}$$

Logic diagram :-





If $R/\bar{L} = 1$; 2, 4, 6, 8 are Disabled
1, 3, 5, 7 are Enabled

If $R/\bar{L} = 0$; 1, 3, 5, 7 are Disabled.
2, 4, 6, 8 are Enabled.

Operation:- Data : 1 1 0 1

MSB LSB

For $R/\bar{L} = 1$ (Right shift)

Clk	$R/\bar{L}$	D_{in}	Q_3	Q_2	Q_1	Q_0
0	—	—	0	0	0	0
↓ 1	1	1 →	1	0	0	0
↓ 2	1	0 →	0	1	0	0
↓ 3	1	1 →	1	0	1	0
↓ 4	1	1 →	1	1	0	1
↓ 5	1	—	1	1	1	0
↓ 6	1	—	1	1	1	1
↓ 7	1	—	1	1	1	1

After 4th clock data is loaded & to take the data O/P we need 3 more clock Pulse

For $R/\bar{L} = 0$ (Left shift) Data : 1 1 0 1

Clk	$R/\bar{L}$	Q_3	Q_2	Q_1	Q_0	D_{in}
0	—	0	0	0	0	—
↓ 1	0	0	0	0	1	← 1
↓ 2	0	0	0	1	1	← 1
↓ 3	0	0	1	1	0	← 0
↓ 4	0	1	1	0	1	← 1
↓ 5	0	1	0	1	1	—
↓ 6	0	0	1	1	1	—
↓ 7	0	1	1	1	1	—

Universal Shift Register:

The most general shift register has following capabilities.

1. A clear control to clear the register to 0.
2. A clock i/p to synchronize the operations.
3. A shift right control to enable the shift right operation and the serial input and output lines associated with shift right.
4. A shift left control to enable the shift left operation and the serial input and output lines associated with shift left.
5. A parallel load control to enable paralle transfer and the ~~n~~ input lines associated with the parallel transfer.

Unidirectional Shift Register:

A Register capable of shifting data in one direction i.e either left shift (or) right shift.

Bidirectional Shift Register:

A register that can shift in both directions is called Bi-directional shift register.

Universal Shift Registers:

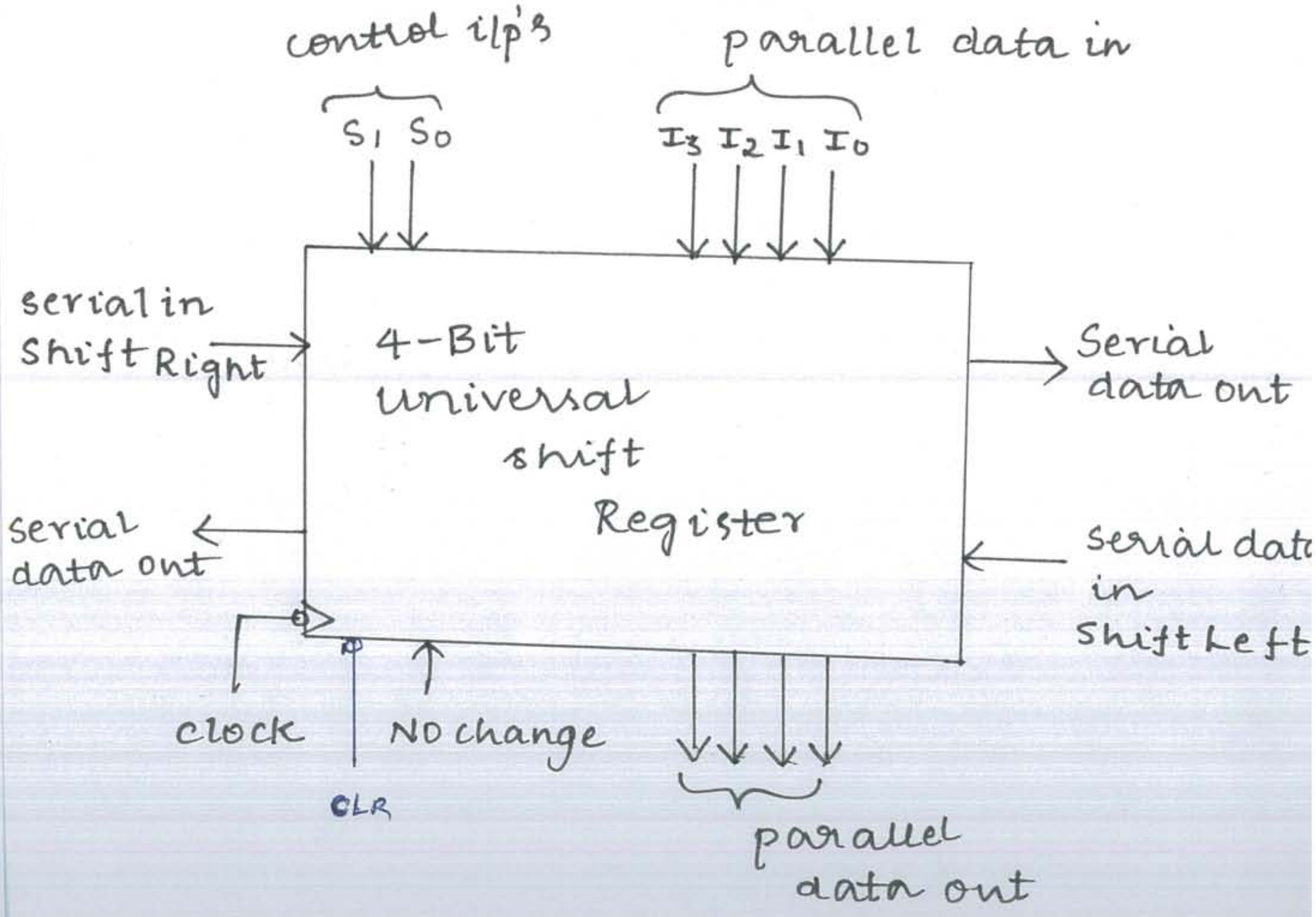
A register that has both shifts and parallel load capability is called as universal Shift Register.

Shift Registers:

SISO
SIPO } → unidirectional shift Registers
PISO

PIPO → No shift just acts like normal register.

Bidirectional: performs both shifts i.e left shift and Right shift.



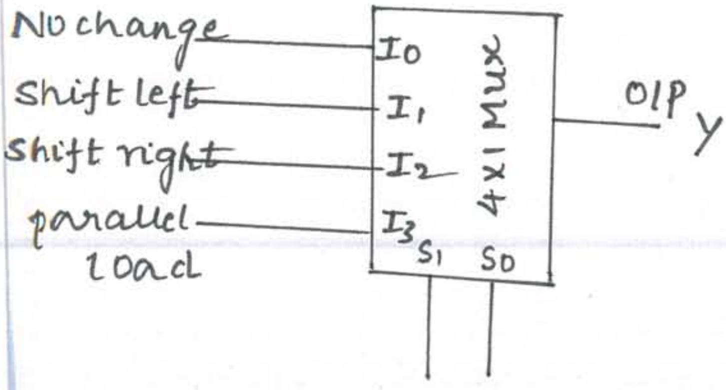
↳ The diagram of a 4-bit universal shift Register that has all the functions (shift left, shift right and parallel load) is shown in fig.

↳ It consists of 4 D-flip flops and 4 Mux.

↳ The 4 Mux's has 2 common selection i/p's S_1 and S_0

#I_0 in each mux is selected when $S_1S_0=00$
I_1 in each mux is selected when $S_1S_0=01$
I_2 in each mux is selected when $S_1S_0=10$
I_3 in each mux is selected when $S_1S_0=11$

The selection i/p's S_1S_0 controls the mode of operation of register as shown in function table.



function table for Register:

Mode control		Register Operation
S_1	S_0	
0	0	No change
0	1	shift left
1	0	shift Right
1	1	parallel load

↳ when $S_1S_0 = 00$, the present value of register is applied to D i/p's of the flipflops. This condition forms a path from the flipflops o/p into the i/p of same flipflop. So, the next clock edge transfers into each flipflop the binary value it held previously. So, no change of state occurs.

↳ when $S_1S_0 = 01$, terminal I_0 of mux i/p has a path to D i/p's of the flipflop. This causes a shift left register operation, with serial i/p transferred into flip flop Q_2

↳ when $S_1S_0 = 10$, a shift right operation results with other serial i/p going into flipflop Q_1.

↳ When $S_1S_0 = 11$, the binary information on the parallel i/p lines is transferred into registers simultaneously during next clock edge.

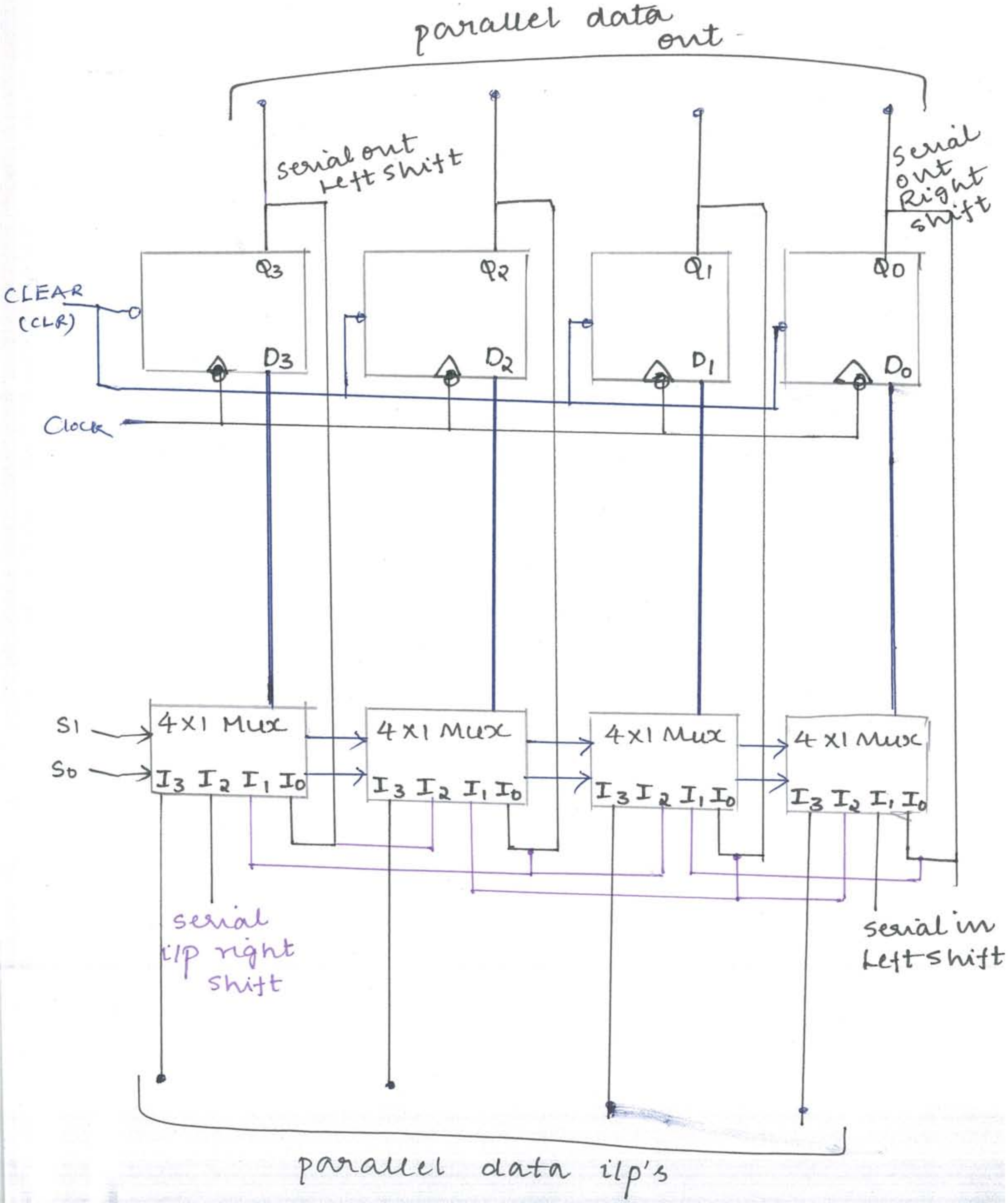


Fig: 4-bit universal shift Register.

Design of Modulo - N countors:

steps involved in design of syncronous counters:

1. obtain the transition table from given ckt information.
2. Determine the no. of flip-flops.
3. choose the type of flip-flop to be used.
4. From transition table, derieve the ckt excitation table.
5. use k-map or any other simplification method to derive the circuit flip-flop input functions.
6. Draw the Logic dgm.

Design of a syncronous Mod-6 counter using clocked JK flip-flops:

The counter with n flip-flops has maximum mod number 2^n.

Design of mod-6 counter using JK flip-flop:

step 1:- Find no. of flip-flops required to build the counter.

flip flops required are: $2^n \geq N$.

Here $N=6$ $\therefore n=3$.

i.e 3 flip-flops are required.

step 2: Determine the transition table or state table.

Present	State		Next	State		flip-flop i/ps.					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	F
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

step 3: k-map simplification for flip-flop i/ps.

for J_A

$Q_A \backslash Q_BQ_C$	$Q_B'Q_C'$	$Q_B'Q_C$	Q_BQ_C	Q_BQ_C'
Q_A'	0	0	1	0
Q_A	X	X	X	X

$J_A = Q_BQ_C$.

for K_A

$Q_A \backslash Q_BQ_C$	$Q_B'Q_C'$	$Q_B'Q_C$	Q_BQ_C	Q_BQ_C'
Q_A'	X	X	X	X
Q_A	0	1	X	X

$K_A = Q_C$

for J_B,

$Q_A \backslash Q_BQ_C$	$Q_B'Q_C'$	$Q_B'Q_C$	Q_BQ_C	Q_BQ_C'
Q_A'	0	1	X	X
Q_A	0	0	X	X

$J_B = Q_A'Q_C$

for K_B,

$Q_A \backslash Q_BQ_C$	$Q_B'Q_C'$	$Q_B'Q_C$	Q_BQ_C	Q_BQ_C'
Q_A'	X	X	1	0
Q_A	X	X	X	X

$K_B = Q_C$

for J_C,

$Q_A \backslash Q_BQ_C$	$Q_B'Q_C'$	$Q_B'Q_C$	Q_BQ_C	Q_BQ_C'
Q_A'	1	X	X	1
Q_A	1	X	X	X

$J_C = 1$

for K_C,

$Q_A \backslash Q_BQ_C$	$Q_B'Q_C'$	$Q_B'Q_C$	Q_BQ_C	Q_BQ_C'
Q_A'	X	1	D	X
Q_A	X	1	X	X

$K_C = 1$.

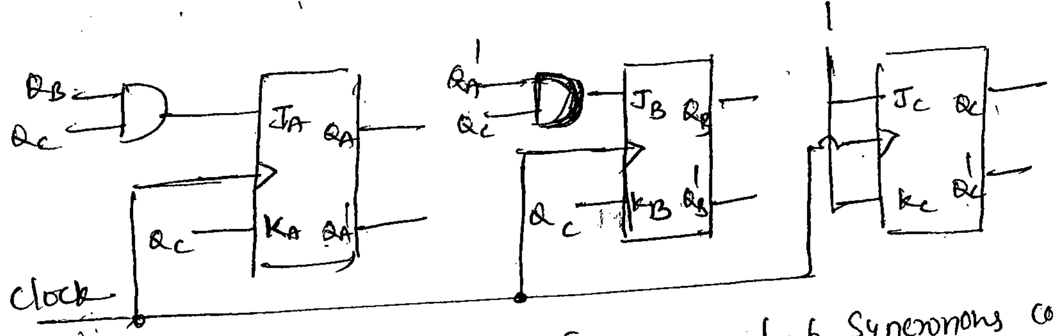


fig: Implementation of mod-6 Syncronons counter using JK-flip-flo

Design of syncronons Decade counter (Mod-10 counter):

Table below shows the excitation table for syncronons decade counter using T-flip-flops.

Present state				Next state				Flip-flop i/p's			
Q_D	Q_C	Q_B	Q_A	Q_{D+1}	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_D	T_C	T_B	T_A
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	1
1	0	1	0	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X

Table: Excitation table.

for T_A,

Q_DQ_C \ Q_BQ_A	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$Q_D'Q_C'$	1	1	1	1
$Q_D'Q_C$	1	1	1	1
Q_DQ_C	X	X	X	X
Q_DQ_C'	1	1	X	X

$T_A = 1$

for T_B

Q_DQ_C \ Q_BQ_A	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$Q_D'Q_C'$	0	1	1	0
$Q_D'Q_C$	0	1	1	0
Q_DQ_C	X	X	X	X
Q_DQ_C'	0	0	X	X

$T_B = Q_D' Q_A$.

for T_D,

$Q_DQ_C \backslash Q_BQ_A$	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$Q_D'Q_C'$	0	0	0	0
$Q_D'Q_C$	0	0	1	0
Q_DQ_C	X	X	X	X
Q_DQ_C'	0	1	X	X

for T_D,

$Q_DQ_C \backslash Q_BQ_A$	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$Q_D'Q_C'$	0	0	1	0
$Q_D'Q_C$	0	0	1	0
Q_DQ_C	X	X	X	X
Q_DQ_C'	0	0	X	X

$T_D = Q_D Q_A + Q_C Q_B Q_A$ $\quad T_C = Q_B Q_A$

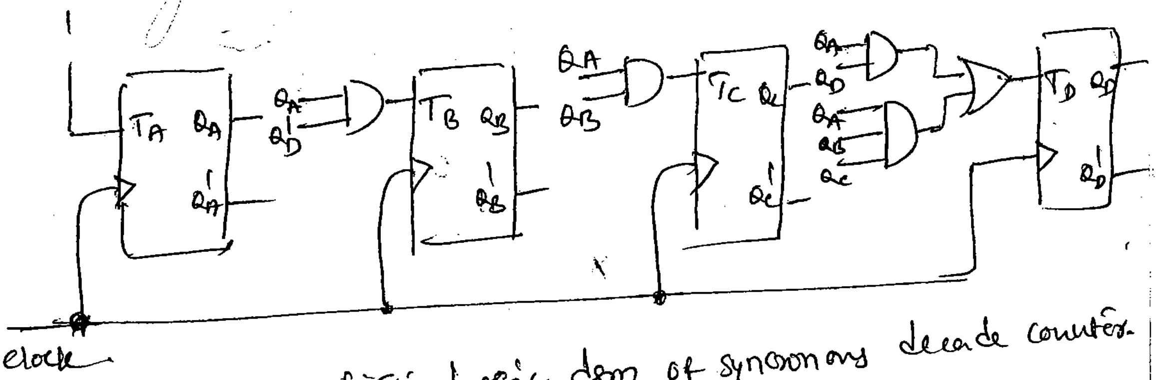


fig: Logic dgm of synchronous decade counter.

Design of up/down synchronous counter:

A up/down counter is one that is capable of progressing in increasing order or decreasing order through a certain sequence.

- → An up/down counter is also called 'Bidirectional counter'
- → up/down counter is controlled by up/down signal.
- → when up/$\overline{\text{down}}$ sig is HIGH, counter goes through up-sequence, i.e 0, 1, 2, 3 ... n.
- → when up/$\overline{\text{down}}$ sig is Low, counter follows reverse sequence i.e n, n−1, ... 0.

For 3-bit counters these sequences are : 0, 1, 2, 3, 4, 5, 6, 7 for up operation & 7, 6, 5, 4, 3, 2, 1, 0 for Down operation.

The arrows in table indicates the state-to-state movement of the counter for both its up & its Down modes of operation.

Cp	up	Qc	QB	QA	Down.
0		0	0	0	
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

Design of 3-bit up/DOWN synchronous counter, using T-flip-flop.

i/p UP/DOWN (UD)	Present State			Next State			Flip-flop i/ps		
	Qc	QB	QA	Qc+1	QB+1	QA+1	Tc	TB	TA
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	1	1
0	0	1	1	0	1	0	0	0	1
0	1	0	0	0	1	1	1	1	1
0	1	0	1	1	0	0	0	0	1
0	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	0	0	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	1
1	1	0	1	1	1	0	0	1	1
1	1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	1

K-map simplification:-

for T_C,

$UD\,Q_C \backslash Q_BQ_A$	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$UD'Q_C'$	1	0	0	0
$UD'Q_C$	1	0	0	0
$UD\,Q_C$	0	0	1	0
$UD\,Q_C'$	0	0	1	0

$T_C = UD'Q_B'Q_A' + UD\,Q_BQ_A$

for T_B,

$UD\,Q_C \backslash Q_BQ_A$	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$UD'Q_C'$	1	0	0	1
$UD'Q_C$	1	0	0	1
$UD\,Q_C$	0	1	1	0
$UD\,Q_C'$	0	1	1	0

$T_B = UD'Q_A' + UD \cdot Q_A$

for T_A,

$UD\,Q_C \backslash Q_BQ_A$	$Q_B'Q_A'$	$Q_B'Q_A$	Q_BQ_A	Q_BQ_A'
$UD'Q_C'$	1	1	1	1
$UD'Q_C$	1	1	1	1
$UD\,Q_C$	1	1	1	1
$UD\,Q_C'$	1	1	1	1

$T_A = 1.$

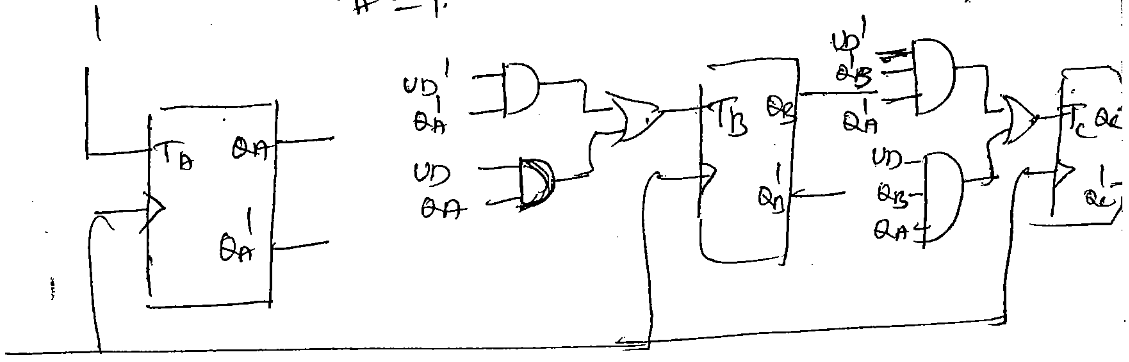


fig: Logic dgm of syncronous up/down counter

Ring counter:-

The ring counter can be used for counting no. of pulses. The no. of pulses counted is read by noting which flipflop is in state -1 NO decoding circuit is required.

construction:-

The o/p of each stage is connected to 'D' i/p of the next stage & o/p of last stage is fed back to i/p of 1st stage. is

The $\overline{CLR}$ followed by $\overline{PRE}$ makes the o/p of 1st stage to '1' and remaining o/p's are '0' i.e., $Q_A=1$ & $Q_B=Q_C=Q_D=0$

operation:-

The 1st clock pulse produces $Q_B=1$ & remaining o/p's are '0'. According to clock pulse applied at the clock i/p CP, a sequence of of 4 states is produced.

These states are listed in table

clock pulse	Q_A	Q_B	Q_C	Q_D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

Ring counter sequence 4-bits

→ As shown in table, '1' is always retained in counter & simply shifted '*' 'around the ring' moving one state to right for each clock pulse.

→ In case of 4-bit sequence, 4 flip-flops are used. so, a sequence of 4 states is produced & repeated.

Design of Ringcounter using D-flipflop:-

State-table:-

Present State				Next state				FlipFlop I/p's			
A	B	C	D	A^+	B^+	C^+	D^+	D_A	D_B	D_C	D_D
0	0	0	0	x	x	x	x	x	x	x	x
0	0	0	1	1	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0	0	1
0	0	1	1	x	x	x	x	x	x	x	x
0	1	0	0	0	0	1	0	0	0	1	0
0	1	0	1	x	x	x	x	x	x	x	x
0	1	1	0	x	x	x	x	x	x	x	x
0	1	1	1	x	x	x	x	x	x	x	x
1	0	0	0	0	1	0	0	0	1	0	0
1	0	0	1	x	x	x	x	x	x	x	x
1	0	1	0	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x	x

Simplification using K-map:-

D_A:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	x_0	1_1	x_3	0_2
A'B	0_4	x_5	x_7	x_6
AB	x_{12}	x_{13}	x_{15}	x_{14}
AB'	0_8	x_9	x_{11}	x_{10}

→ D

$D_A = D$

DB:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	X_0	0_1	X_3	0_2
A'B	0_4	X_5	X_7	X_6
AB	X_{12}	X_{13}	X_{15}	X_{14}
AB'	1_8	X_9	X_{11}	X_{10}

A

$D_B = A$

Dc:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	X_0	0_1	X_3	0_2
A'B	1_4	X_5	X_7	X_6
AB	X_{12}	X_{13}	X_{15}	X_{14}
AB'	0_8	X_9	X_{11}	X_{10}

→ C

$D_C = B$

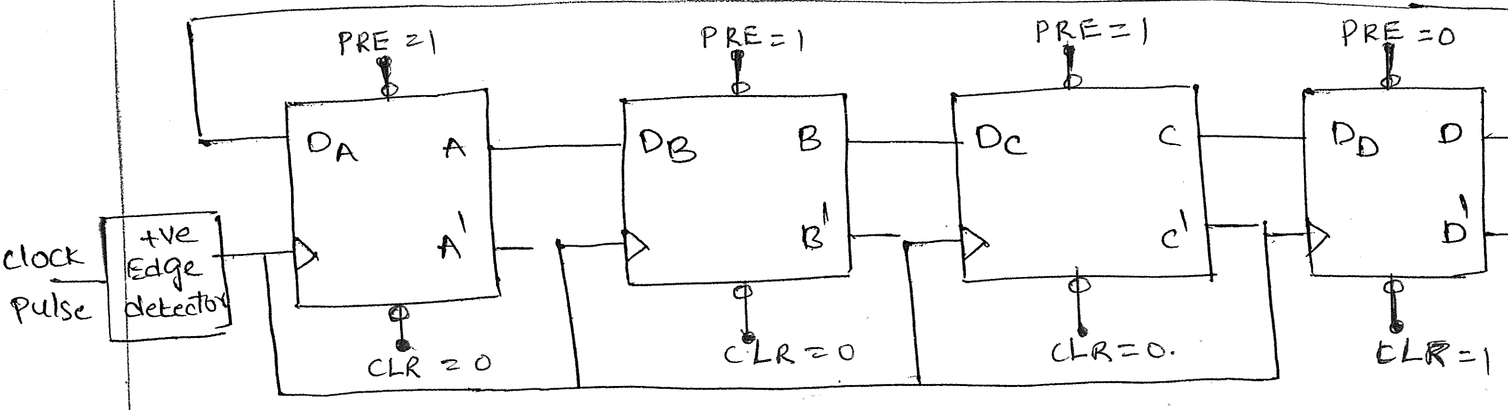
DD:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	X_0	0_1	X_3	1_2
A'B	0_4	X_5	X_7	X_6
AB	X_{12}	X_{13}	X_{15}	X_{14}
AB'	0_8	X_9	X_{11}	X_{10}

→ C

$D_D = C$

Implement using D-flipflop:-



Application:-

The ring counter can be used for counting no. of pulses. The no. of pulses counted is read by noting which flip-flop is in state

→ The o/p of last stage Q_D is '0'. Therefore, complement o/p of last stage, Q_D is '1'. This is connected back to 'D' i/p of 1st stage. So D_A is '1'.

→ The first raising clock edge produces $Q_A=1$ & $Q_B=0$, $Q_C=0$, $Q_D=0$ since D_B, D_C & D_D are '0'.

→ The next rising clock edge produces $Q_A=1$, $Q_B=1$, $Q_C=0$ & $Q_D=0$

The sequence of states is shown in statetable

→ After '8' states, the sequence is repeated.

To design a counter of 5-bit sequence, it requires a total of 10 states.

Design of Johnson counter:-

Present state A B C D	Nextstate $A^+ B^+ C^+ D^+$	Flip Flop I/ps D_A D_B D_C D_D
0 0 0 0	1 0 0 0	1 0 0 0
0 0 0 1	0 0 0 0	0 0 0 0
0 0 1 0	× × × ×	× × × ×
0 0 1 1	0 0 0 1	0 0 0 1
0 1 0 0	× × × ×	× × × ×
0 1 0 1	× × × ×	× × × ×
0 1 1 0	× × × ×	× × × ×
0 1 1 1	0 0 1 1	0 0 1 1
1 0 0 0	1 1 0 0	1 1 0 0
1 0 0 1	× × × ×	× × × ×
1 0 1 0	× × × ×	× × × ×

No decoding circuit is required. since there is one pulse at the o/p for each of N clock pulses.

Note:-

→ Ring counter is also referred to as divide by N counter (or) an N:1 scalar.

→ Ring counters can be instructed for any desired mod No; ie, Mod-N ring counter requires N-flipflops.

Johnson (or) Twisting Ring (or) switch tail counter:-

In a Johnson counter, the 'Q' o/p of each stage of flip-flop is connected to 'D' i/p of the next stage, except that the complement o/p of last stage flipflop is connected back to D-i/p of 1st flipflop.

Initially, the registors (ie, all flipflop's) are cleared. so all o/p's Q_A, Q_B, Q_C & Q_D are '0'.

clock pulse	Q_A	Q_B	Q_C	Q_D.
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

1 0 1 1	X X X X	X X X X
1 1 0 0	1 1 1 0	1 1 1 0
1 1 0 1	X X X X	X X X X
1 1 1 0	1 1 1 1	0 1 1 1
1 1 1 1	0 1 1 1	0 1 1 1

Simplification using K-map:-

DA:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	1_{0}	0_{1}	0_{3}	X_{2}
A'B	X_{4}	X_{5}	0_{7}	X_{6}
AB	1_{12}	X_{13}	0_{15}	1_{14}
AB'	1_{8}	X_{9}	X_{11}	X_{10}

D'

DA = D'

DB:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	0_{0}	0_{1}	0_{3}	X_{2}
A'B	X_{4}	X_{5}	0_{7}	X_{6}
AB	1_{12}	X_{13}	1_{15}	1_{14}
AB'	1_{8}	X_{9}	X_{11}	X_{10}

A

DB = A

Dc:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	0_{0}	0_{1}	0_{3}	X_{2}
A'B	X	X	1	X
AB	1	X	1	1
AB'	0	X	X	X

→ D.

Dc = B

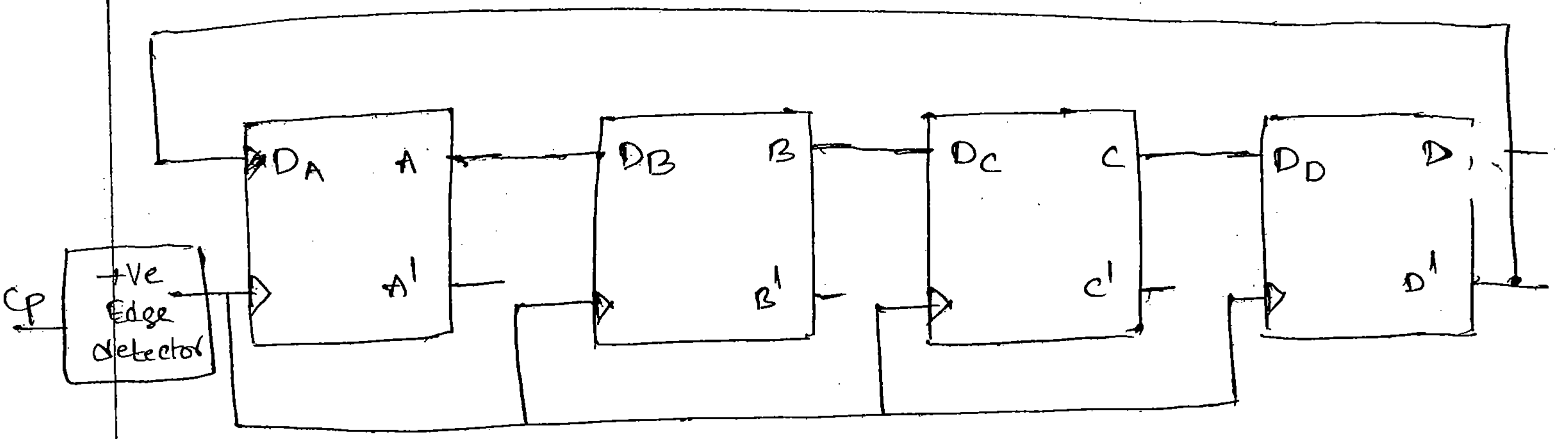
DD:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	0_{0}	0_{1}	1_{3}	X_{2}
A'B	X_{4}	X_{5}	1_{7}	X
AB	0_{12}	X_{13}	1_{15}	1_{14}
AB'	0_{8}	X_{9}	X_{11}	X_{10}

→ C

DD = C

Implement using D-flipflop:-



Note:-

In General, a n-stage Johnson counter will produce a modulus of 2×n, where 'n' is the no. of stages (ie, flipflops) in the counter.

As shown in table, the counter will "fill up" with 1's from left to right & then fill up with 0's again.

Advantage:-

NO decoding is required.

(Readily decoded with two i/p And Gates)

Counters:

A Counter is a sequential circuit that goes through a predetermined sequence of binary states upon the application of input pulses.

\# A Counter is used to count clock pulses.

\# A counter can also be used as frequency divider.

⇒ with n-flip flops maximum possible states in the counter is 2^n.

$$\boxed{N \leq 2^n}$$

where N - No. of States,
n - No. of flip flop's.

Ex: $n = 2$, $N \leq 2^2 \Rightarrow N \leq 4$, ⇒ Mod 4 counter.
$n = 3$, $N \leq 2^3 \Rightarrow N \leq 8$ ⇒ Mod 8 counter,

Types of Counters:

Depending on clock pulse applied, counters are of 2 types.

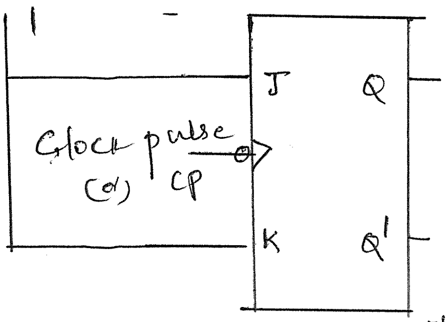
1. Asyncronous | Ripple counters. &
2. Syncronous counters.

⇒ The main difference, between Asyncronous & syncronous is that, in syncronous same clock is given to all flip flops for triggering simultaneously, But in Asyncronous, the output of one flip flop

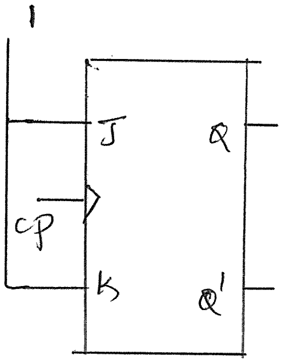
Asyncronous Counter's:

Generally, JK & T flip flops are used to design Asyncronous counters, because they ~~have~~ produce complemented outputs, when J=1 & K=1; (or) T=1.

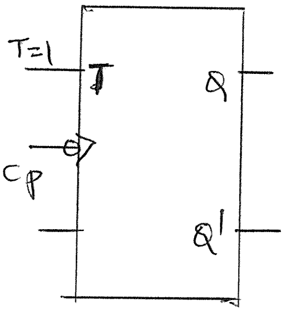
We can use either +ve edge or -ve edge triggered 'JK' or 'T' flip flops to design Asyncronous counters. i.e.



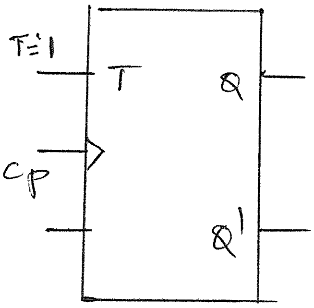
-ve edge Triggered JK ff
If bubble is present,



+ve edge Triggered JK-ff



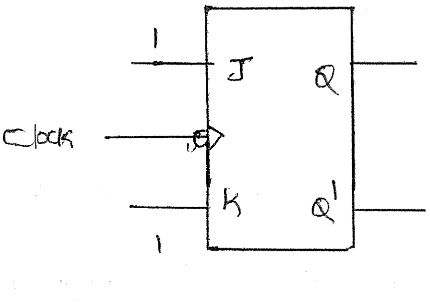
-ve edge Triggered T-ff



+ve edge Triggered T-ff.

→ In Asyncronous counters, the inputs of J & K is 1; i.e to obtain toggle state.

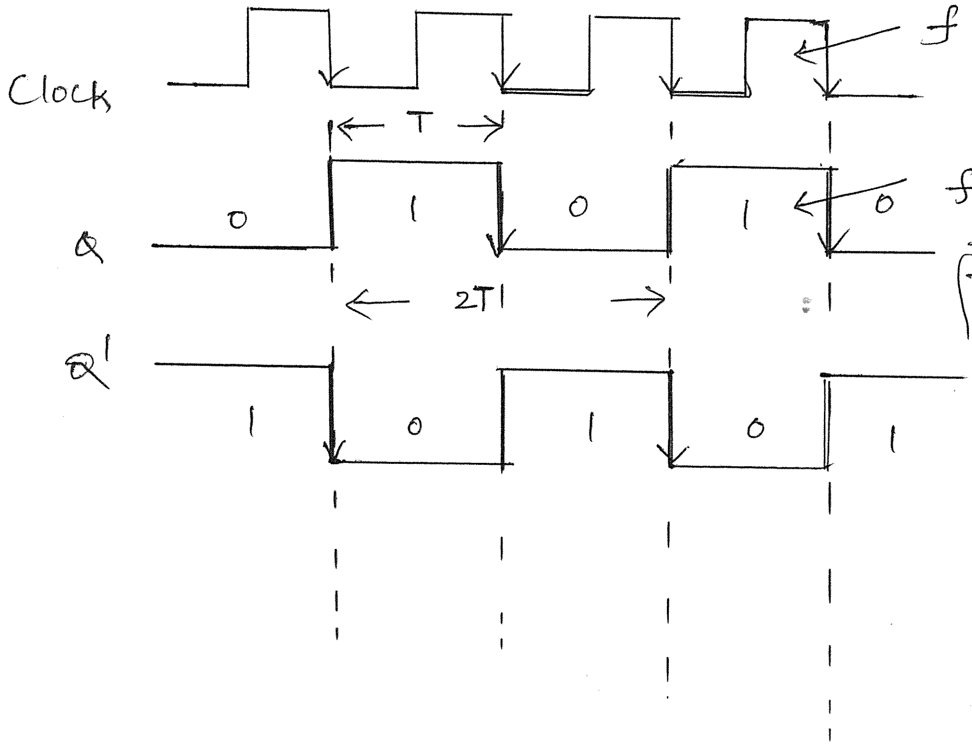
1-bit counter (or) Mod-2 counter (or) Divide by 2 counter:



Truth Table (or) Operati

Clock	Q	Q'
Initial 0	0	1
↓ 1	1	0
↓ 2	0	1
↓ 3	1	0
	up Countin	Down counti

Timing diagram:



$f = \frac{1}{T}$

$f_1 = \frac{1}{2T}$

$\left(f_1 = \frac{f}{2}\right)$

↓

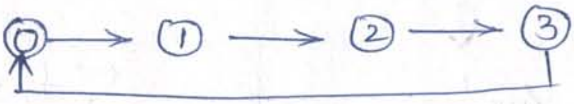
So, 1-bit counter is also called Divide frequency by 2 counter.

ASYNCRONOUS COUNTER:

* 2-bit ripple : counter:

(Mode-4 counter (or) Divide by 4)

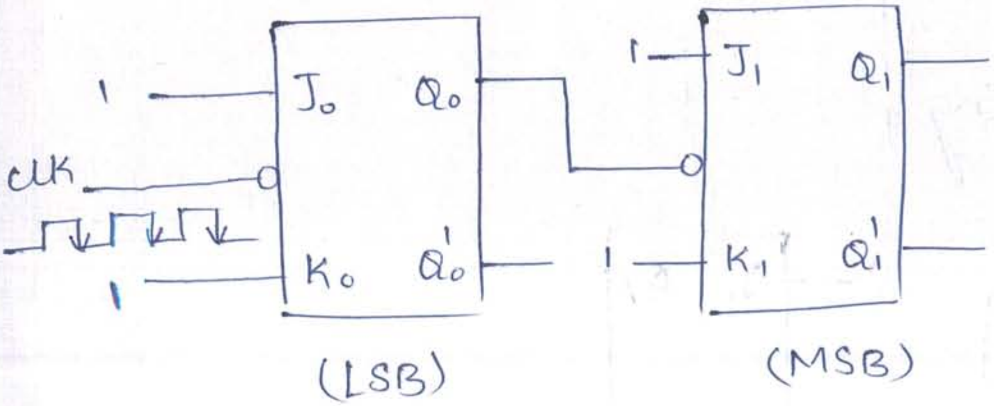
-ve edge trigerring:



$2^n \geq N$

$2^n \geq 4$

$n = 2$ (flip flop's required)



clk	$Q_1\ Q_0$	$Q_1'\ Q_0'$
0	0 0	1 1
↓ 1	0 1	1 0
↓ 2	1 0	0 1
↓ 3	1 1	0 0
↓ 4	0 0	1 1

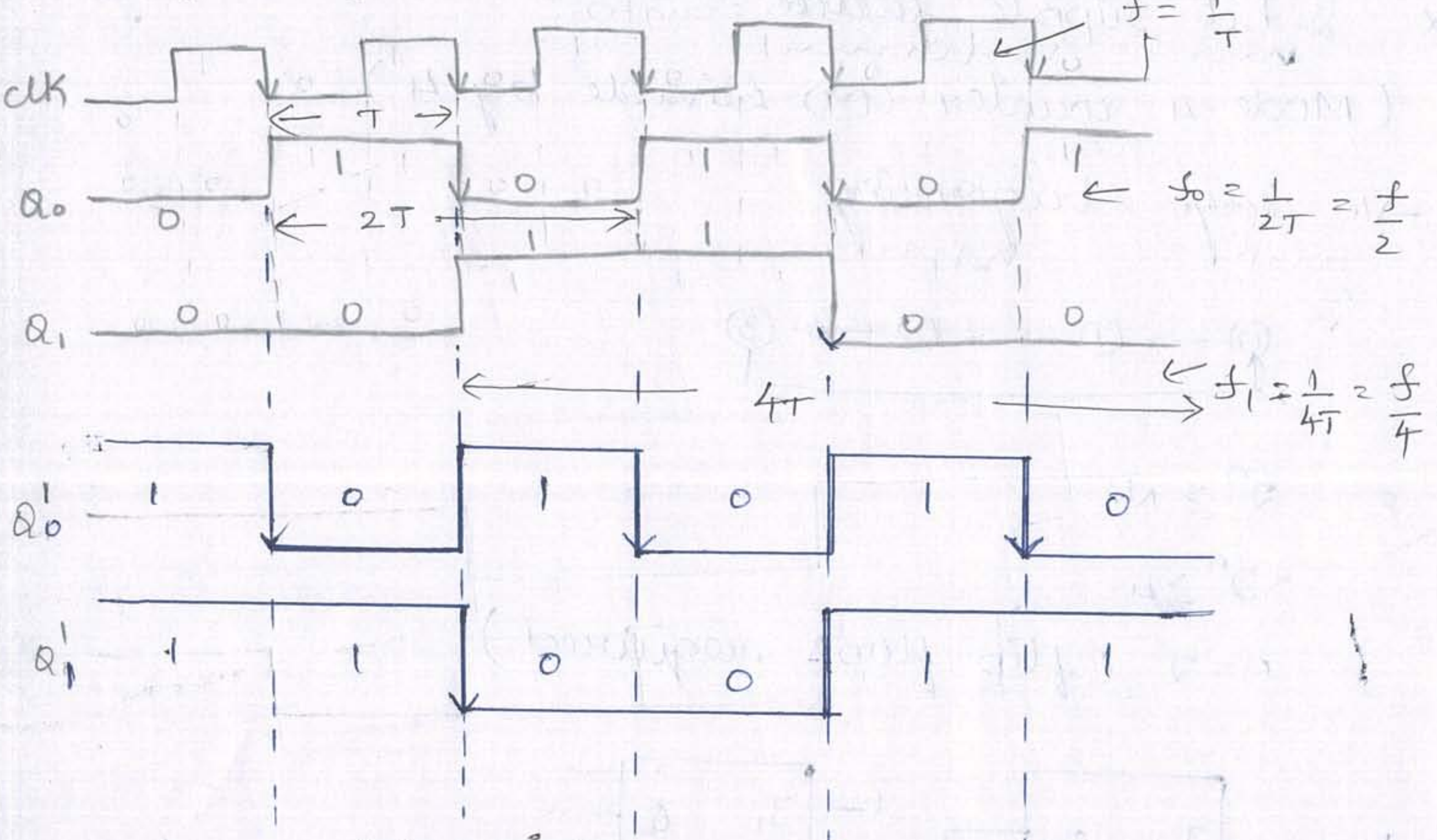
↓ up counting

↓ Down counting.

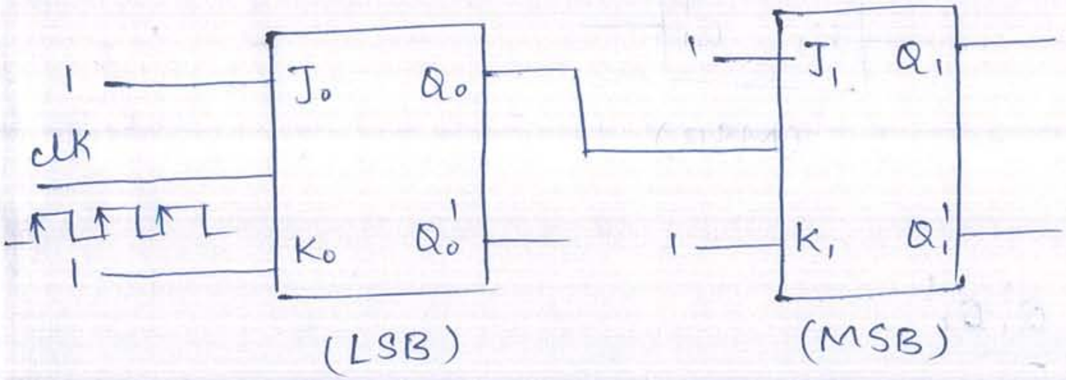
$Q_1 Q_0 \rightarrow$ up counter

$Q_1' Q_0' \rightarrow$ Down counter

Timing diagram:



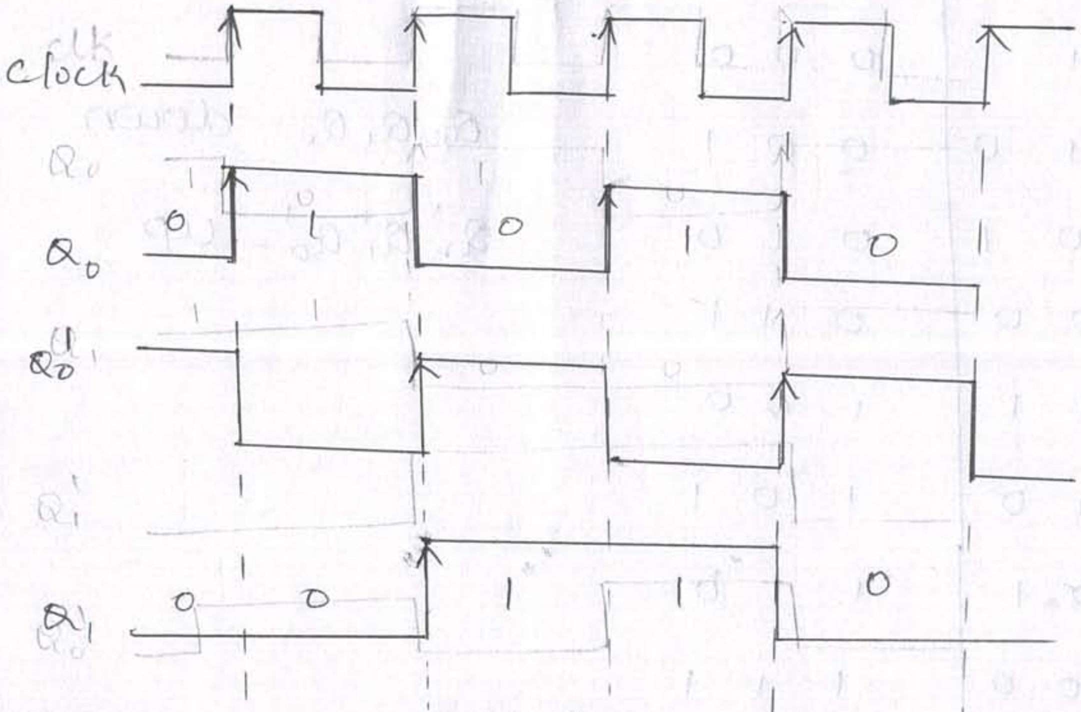
+ve edge triggering:



clk	Q_1 Q_0	Q_1' Q_0'
0	0 0	1 1
↑ 1	1 1	0 0
↑ 2	1 0	0 1
↑ 3	0 1	1 0
↑ 4	0 0	1 1

$Q_1 Q_0$ → Down counter

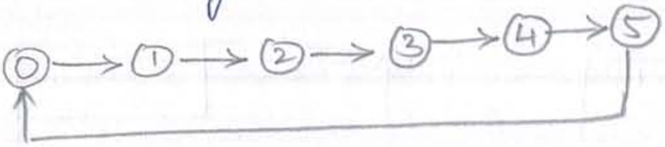
$Q_1' Q_0'$ → up counter

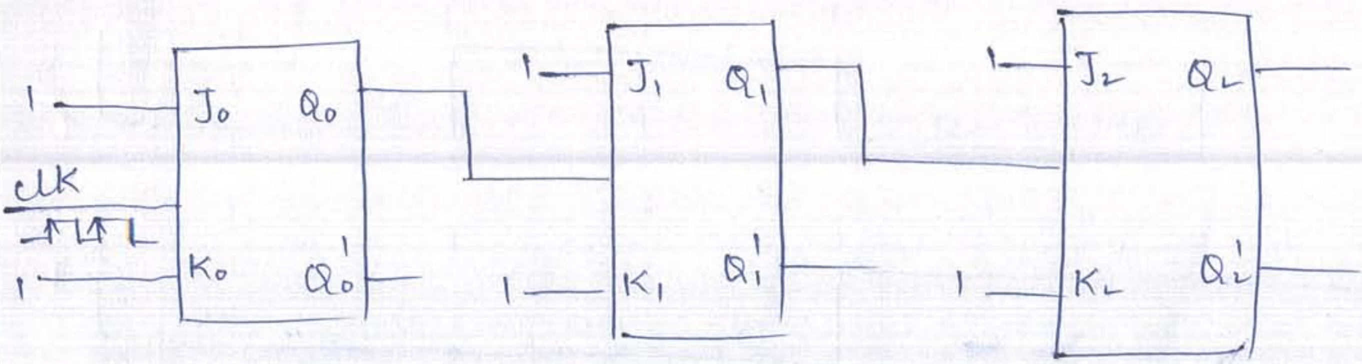


3- Bit ripple ~~factor~~ counter:-

[Mode-6 counter (or) Divide by 8 counter]

+ve edge triggering:

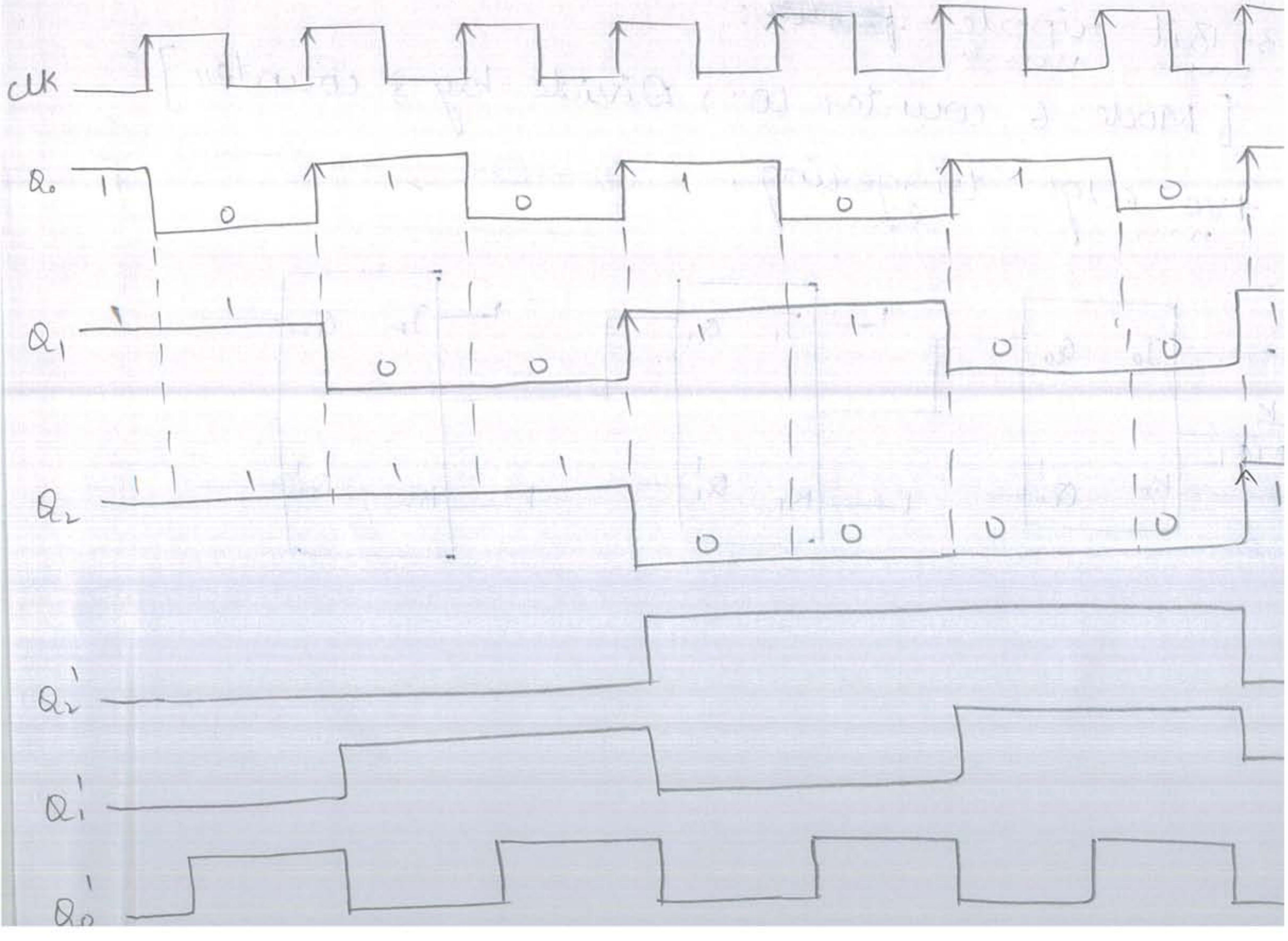




clk	Q_2 Q_1 Q_0	Q_2' Q_1' Q_0'
0	0 0 0	1 1 1
↑ 1	1 1 1	0 0 0
↑ 2	1 1 0	0 0 1
↑ 3	1 0 1	0 1 0
↑ 4.	1 0 0	0 1 1
↑ 5	0 1 1	1 0 0
↑ 6	0 1 0	1 0 1
↑ 7	0 0 1	1 1 0
↑ 8	0 0 0	1 1 1

$Q_2 Q_1 Q_0$ – down

$Q_2' Q_1' Q_0'$ – up

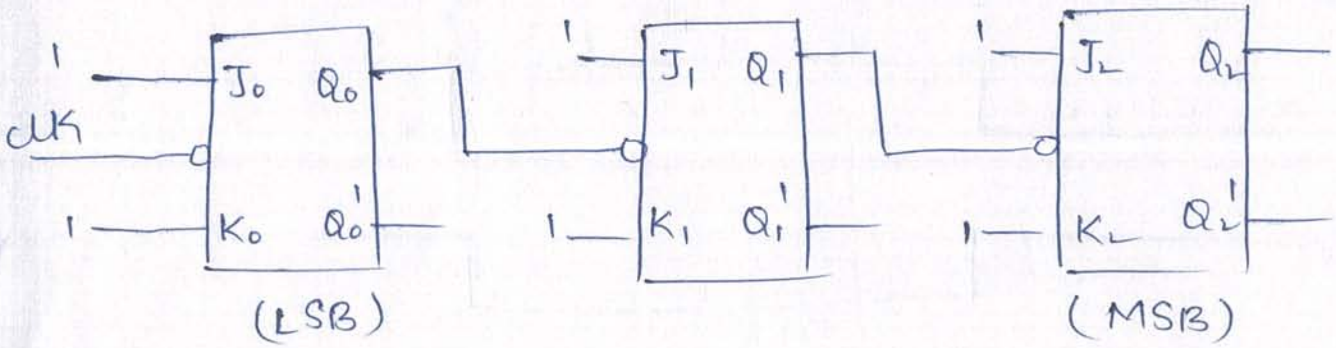


-ve edge triggering:

$2^n \geq N$

$2^3 \geq 6$

3 : NO. of flip flop's

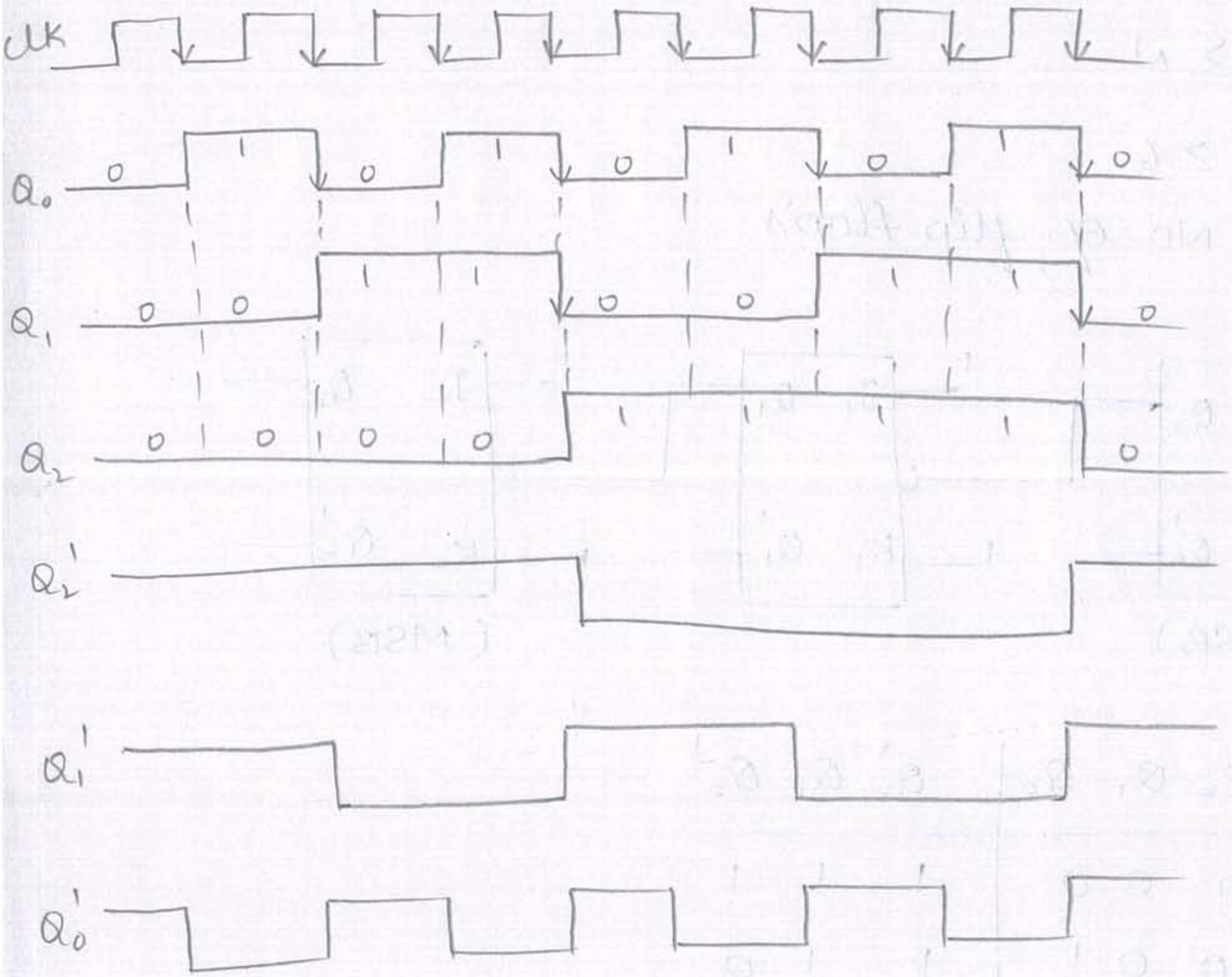


clk	Q_2	Q_1	Q_0	Q_2'	Q_1'	Q_0'
0	0	0	0	1	1	1
↓ 1	0	0	1	1	1	0
↓ 2	0	1	0	1	0	1
↓ 3	0	1	1	1	0	0
↓ 4	1	0	0	0	1	1
↓ 5	1	0	1	0	1	0
↓ 6	1	1	0	0	0	1
↓ 7	1	1	~~0~~1	0	0	0
↓ 8	0	0	0	1	1	1

$Q_2 Q_1 Q_0$ → up

$Q_2' Q_1' Q_0'$ - down

Timing diagram:



4 Bit Asynchronous Counter

Asynchronous 4-bit Down counter :-

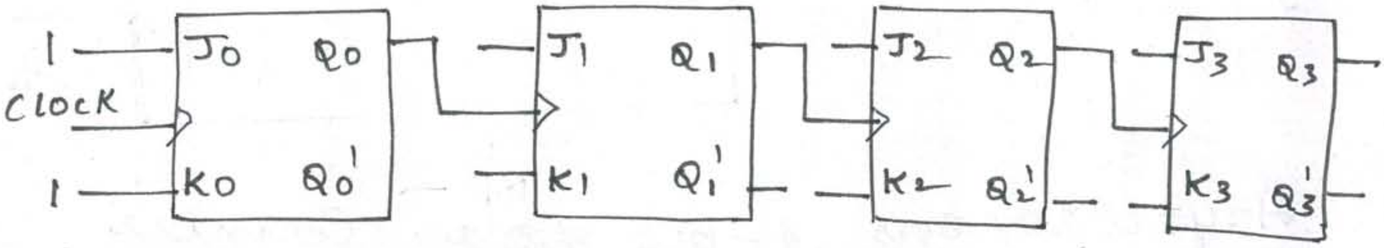
Mod-10 Counter

$$2^n \geq N$$

$$2^4 \geq 10$$

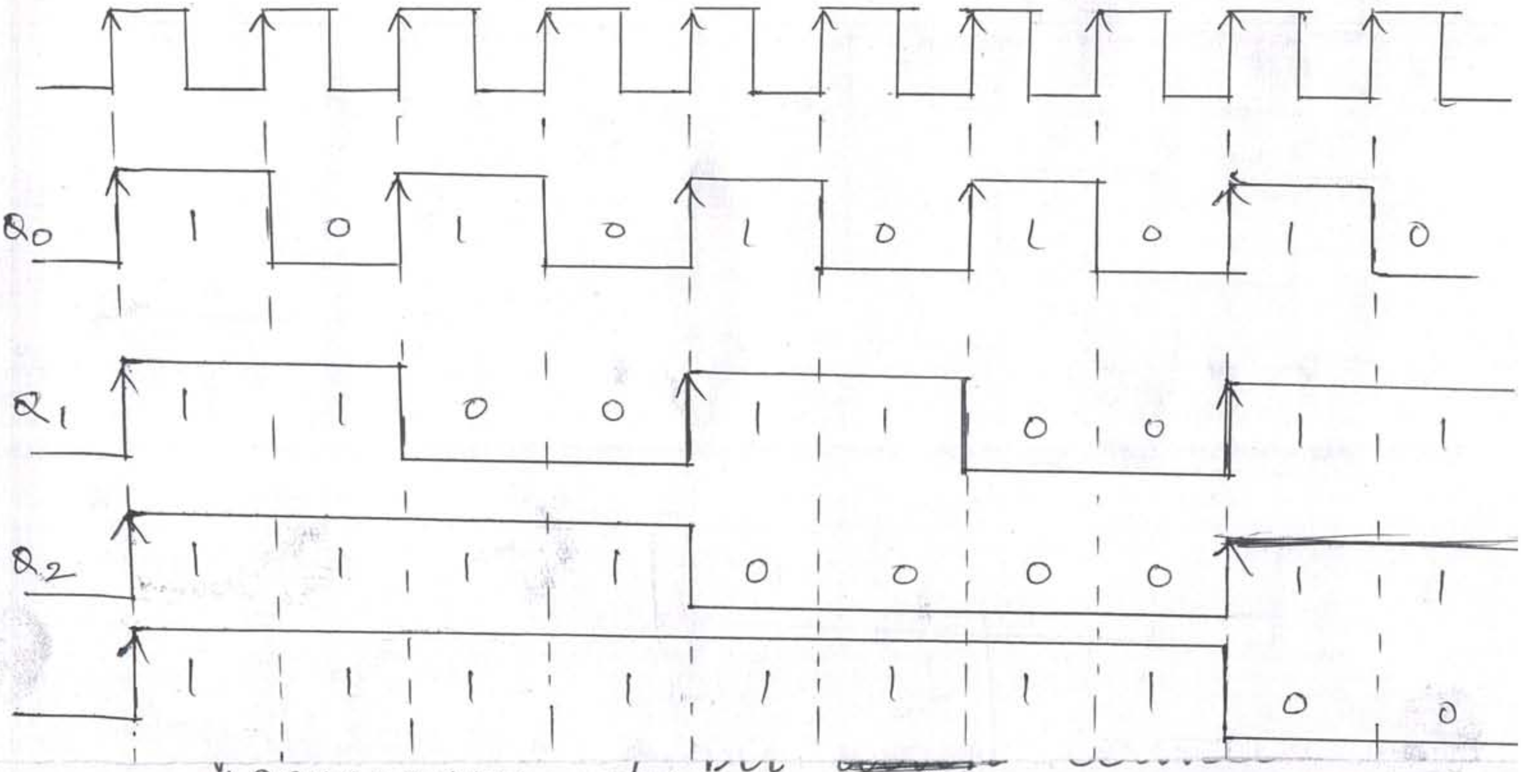
$$n = 4$$

∴ 4 flip flops are required to design a Mod-10 counter

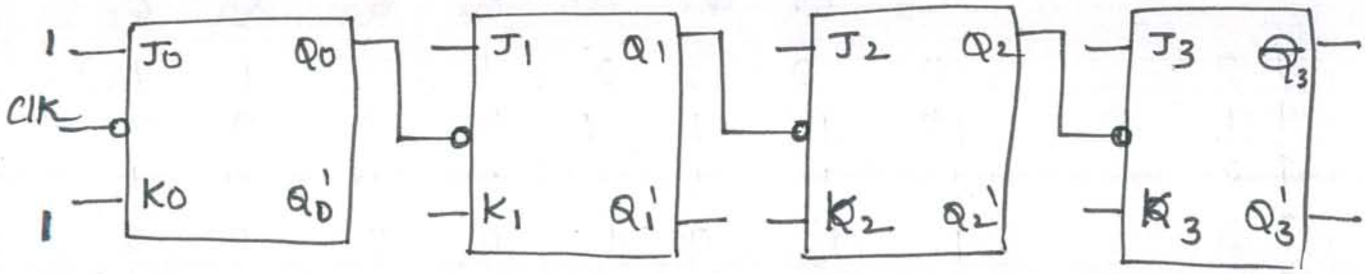


Clock	Down				up			
	Q_3	Q_2	Q_1	Q_0	Q_3'	Q_2'	Q_1'	Q_0'
↑ 0	0	0	0	0	1	1	1	1
↑ 1	1	1	1	1	0	0	0	0
↑ 2	1	1	1	0	0	0	0	1
↑ 3	1	1	0	1	0	0	1	0
↑ 4	1	1	0	0	0	0	1	1
↑ 5	1	0	1	1	0	1	0	0
↑ 6	1	0	1	0	0	1	0	1
↑ 7	1	0	0	1	0	1	1	0
↑ 8	1	0	0	0	0	1	1	1
↑ 9	0	1	1	1	1	0	0	0
↑ 10	0	1	1	0	1	0	0	1
↑ 11	0	1	0	1	1	0	1	0
↑ 12	0	1	0	0	1	0	1	1
↑ 13	0	0	1	1	1	1	0	0
↑ 14	0	0	1	0	1	1	0	1
↑ 15	0	0	0	1	1	1	1	0
↑ 16	0	0	0	0	1	1	1	1

Timing diagram:



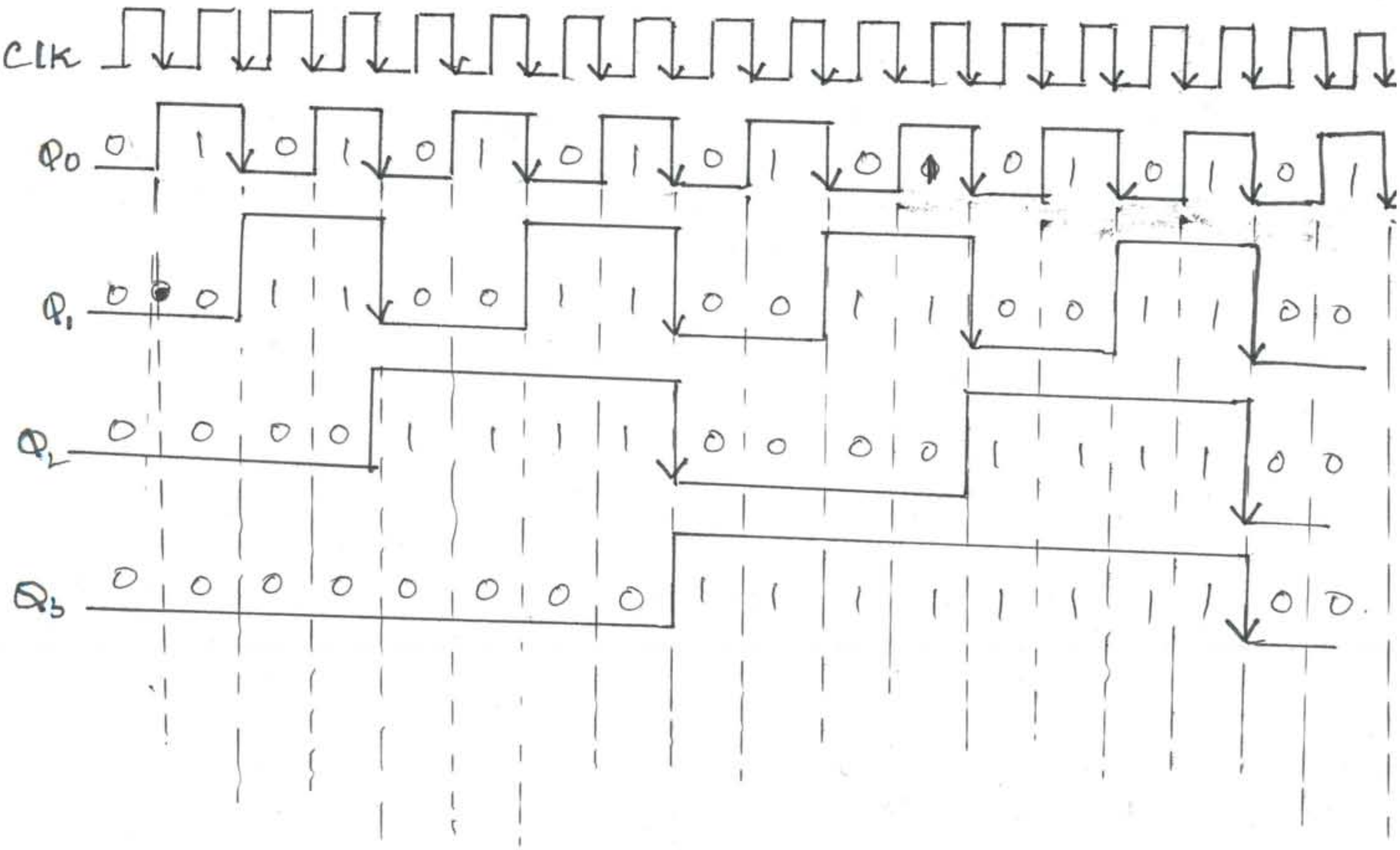
Asyncronous 4-Bit up counter:-



Clock	Q3	Q2	Q1	Q0	Q3'	Q2'	Q1'	Q0'
0	0	0	0	0	1	1	1	1
↓1	0	0	0	1	1	1	1	0
↓2	0	0	1	0	1	1	0	1
↓3	0	0	1	1	1	1	0	0
↓4	0	1	0	0	1	0	1	1
↓5	0	1	0	1	1	0	1	0
↓6	0	1	1	0	1	0	0	1
↓7	0	1	1	1	1	0	0	0

↓8	1 0 0 0	0 1 1 1
↓9	1 0 0 1	0 1 1 0
↓10	1 0 1 0	0 1 0 1
↓11	1 0 1 1	0 1 0 0
↓12	1 1 0 0	0 0 1 1
↓13	1 1 0 1	0 0 1 0
↓14	1 1 1 0	0 0 0 1
↓15	1 1 1 1	0 0 0 0
↓16	0 0 0 0	1 1 1 1

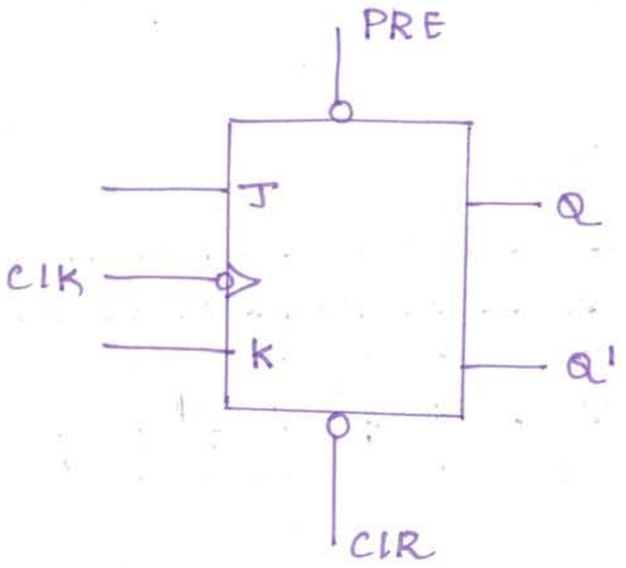
Timing Diagram:



Present and Clear Inputs :

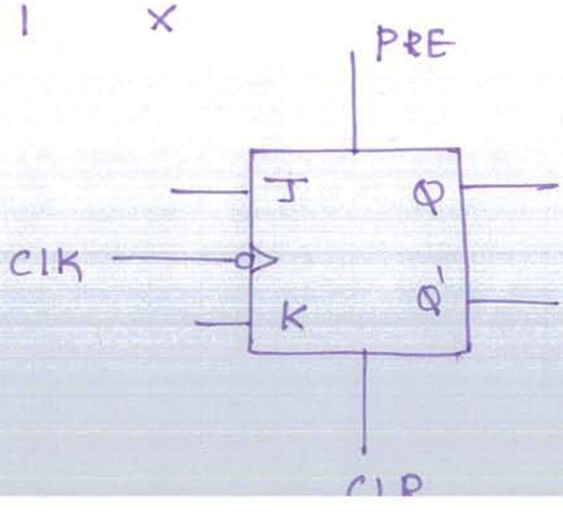
Active Low

PRE	CLR	Q
0	1	1
1	0	0
1	1	Normal JK
0	0	X

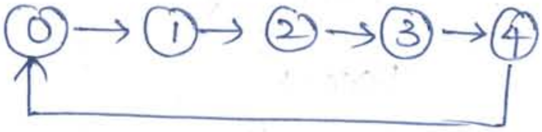


Active high:

PRE	CLR	Q
0	0	Normal JK
0	1	0
1	0	1
1	1	X



ClK	Q_2	Q_1	Q_0	CLR
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	X
7	1	1	1	X



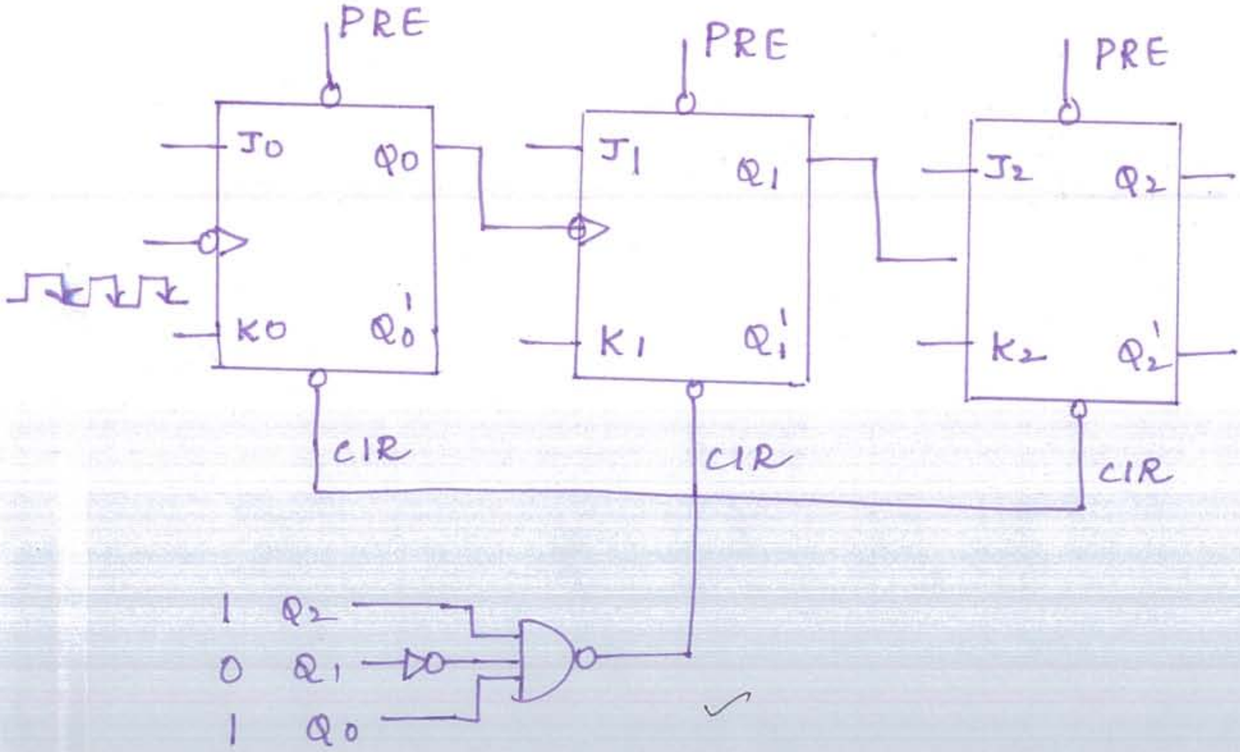
K-map Simplification

ClR

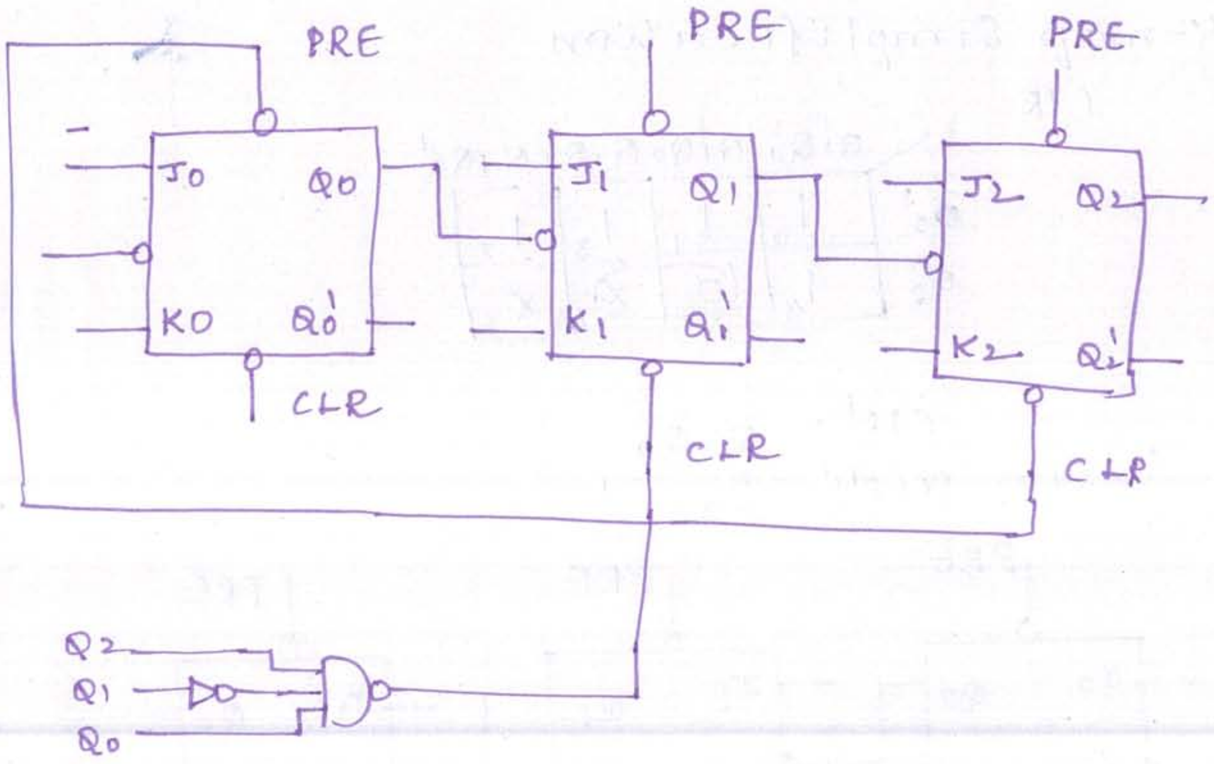
	$Q_1'Q_0'$	$Q_1'Q_0$	Q_1Q_0	Q_1Q_0'
Q_2'	1_0	1_1	1_3	1_2
Q_2	1_4	0_5	X_7	X_6

$$ClR' = Q_2Q_0$$

$$CLR = (CLR')' = (Q_2Q_0)'$$



	CLK P	Q2	Q1	Q0
	0	0	0	0
	1	0	0	1
	2	0	1	0
	3	0	1	1
Tsec→ ↕	4	1	0	0
Tsec→	5	1	0	1
	6	1	1	0
	7	1	1	1



Comparison between Asyncronous counters & syncronous counters:

Asyncronous Counters	Syncronous Counters.
1. In Asyncronous Counters, the flip flop o/p drives the clock of next flip flop. i.e the o/p of 1st flip flop drives the Clock for second flip flop, & o/p of 2nd flip flop drives the clock for 3rd.	1. In Syncronous counters, there is no connection between the o/p of 1st flip flop & clock i/p of next.
2. All flip flops are not clocked simultaneously	2. All flip flops are clocked simultaneously.
3. Design & Implementation is very simple even for more number of states.	3. Design & Implementatio is complex as no. of states increases. [i.e Excitation tables, flip flop i/p's requ
4. Low speed.	5. High speed. i.e All flip flops are clocked simultaneously.