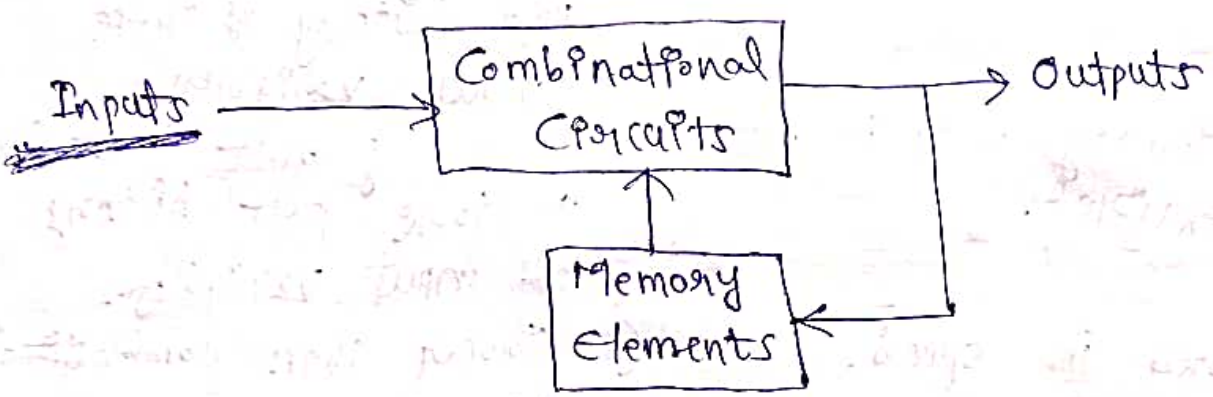
Unit-3:- Sequential Circuits

Sequential Logic Circuits: RS, Clocked RS, JK, Master Slave JK, T-Flip flops, shift Registers, Types of shift Registers, Counters, Synchronous Counters, Asynchronous Counters, Up-Down Counter.

Sequential Circuits:-

Digital outputs are generated in accordance with the sequence in which the input signals are received.



Outputs generated not only depend on the present input conditions but they also depend upon the past history of these inputs. The past history is provided by feed back from the output back to the input.

Sequential circuits can be classified as

(i) Synchronous Sequential Circuits
(ii) Asynchronous Sequential Circuits.

Synchronous sequential circuits signals can affect the memory elements at discrete instance of time.

In asynchronous sequential circuits change in input signals can affect memory element at any instant of time.

Q. Write the comparision b/w combinational and sequential circuits.

Sol:-

Combinational Circuits	Sequential Circuits
(i) At all times o/p variables depends on combination of input variables	(i) Output variables depends on not only present i/p variable but also on past history of these input variables.
(ii) Memory unit is not required.	(ii) Memory unit is required to store past history of input variables.
(iii) Fasted in speed.	(iii) Slower than combinational circuit.
(iv) Easty to design	(iv) These are harder to design.
(v) Parallel adder is a combinational circuit.	(v) Serial adder is a sequential circuit.

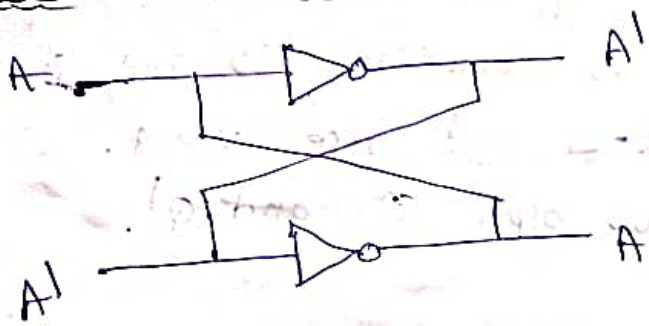
Q. Write the comparision b/w synchronous and asynchronous sequential circuit.

Sol:-

Synchronous	Asynchronous
(i) In synchronous circuits the memory elements are clocked flip flops	(i) In this, memory elements are unclocked flip flops. (or) time delay elements

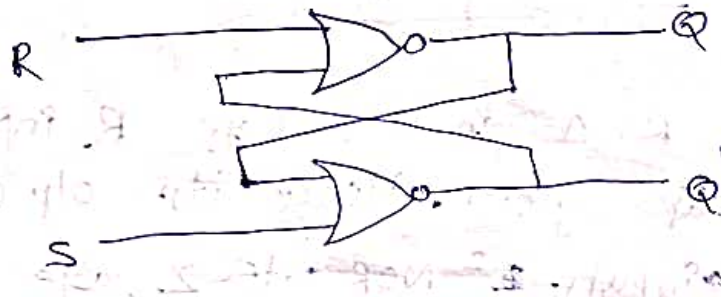
(ii) In this the change in input signals can affect memory elements upon activation of clock signal.	(ii) In this the change in input signals can affect memory elements at any instant of time.
(iii) Slower than asynchronous sequential circuits.	(iii) Faster than synchronous sequential circuits.

Basic bi-stable element:-



Latch.

RS Flip Flop:-



R	S	Q	Q'
0	0	0	1
0	1	1	0
1	0	0	1
1	1	0	0

⟸ prohibited case (or) Indeterminate case

NOR

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

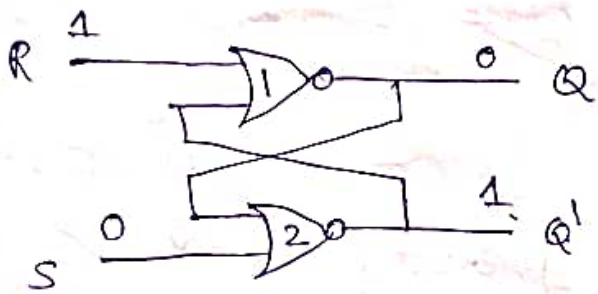
RS-Flip Flop is the most basic of all flip flops. Letters R and S here stands for "Reset" and "set" respectively.

When the flip flop is set its 'Q', output goes to 1. when it is reset it goes to '0' state. Q' is compliment of Q at all times

Two "NOR" gates are cross coupled, so that the o/p of NOR gate-1 is connected to one of the inputs of "NOR" gate-2 and vice versa.

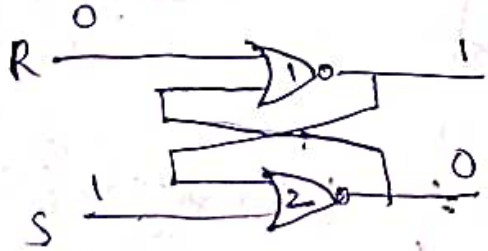
The flip flop has two o/ps Q and Q'.

Case-i:



When S=0 and R=1, in this case R input of NOR gate-1 is at logic-1. Hence its o/p Q is at logic-0. Both inputs of NOR gate-2 are now logic-0. So, its o/p Q' is at logic-1.

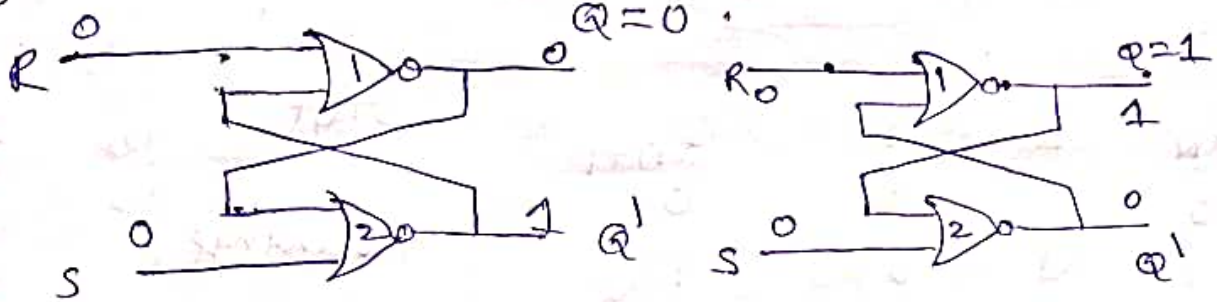
Case-ii:



When S=1 and R=0. In this case NOR gate-2 is logic-1. Hence o/p Q' is at logic-0. Both

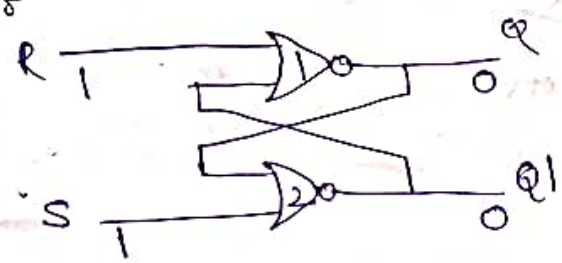
inputs 2. NOR-gate-1 are now logic-0. So that o/p Q is at logic-1.

case-iii:



when S=0 and R=0. Initially Q=1 and Q'=0 with Q'=0 both inputs 2 NOR gate-1 are at logic-0. So, its output Q is at logic-1. With Q=1, one input of NOR gate-2 is at logic-1. Hence its output Q' is at logic-0. This shows that when S and R both are low (at logic-0). The o/p state does not change, ily for Q=0 and Q'=1. The o/p state does not change.

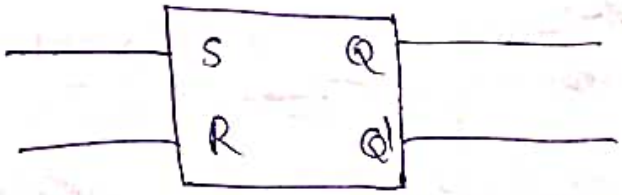
Case-iv:



Indeterminate state (or) prohibited state.

When R and S are both at logic-1 they force the o/ps of both NOR gates at the low state. That is Q=0, Q'=0. So we call this is an indeterminate (or) prohibited state and represent this by an asterisk (*). This condition must be avoided by making sure that 1s are not applied to both the inputs simultaneously.

Symbol for RS Flip-Flop



Q_n	R	S	Q_{n+1}	state
0	0	0	0	No change
1	0	0	1	
0	0	1	1	set
1	0	1	1	
0	1	0	0	Reset
1	1	0	0	
0	1	1	*	Invalid (or Prohibited (or Indeterminate state
1	1	1	*	

If Q_n = Present state

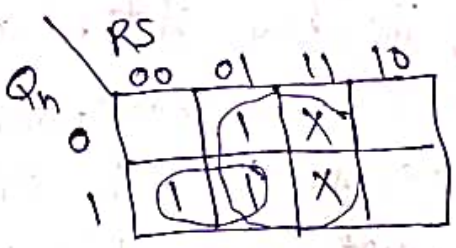
Q_{n+1} = next state.

If Q_n = Previous state

Q_{n+1} = Present state

Characteristic Equation:-

for Q_{n+1}

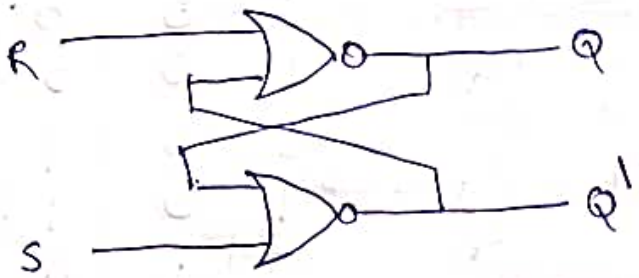


001
011
101
111

S

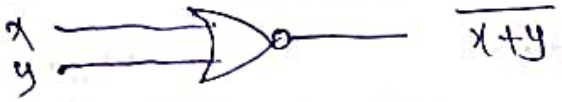
For $Q_{n+1} = S + Q_n R'$

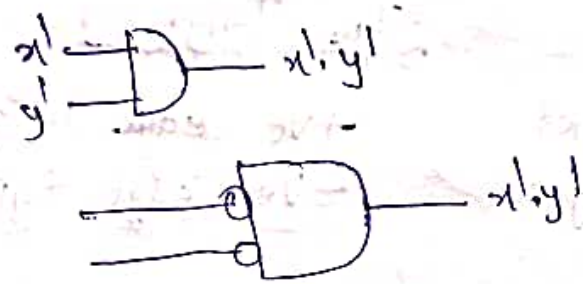
RS Flip-flop using NAND Gates:



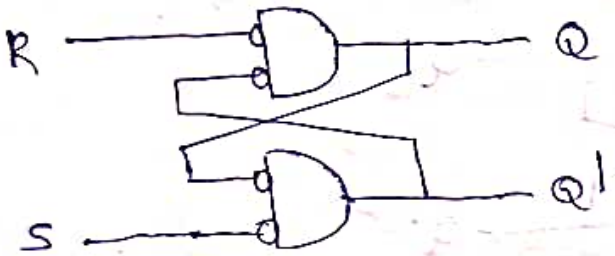
x, y inputs NOR gate o/p = $\overline{x+y}$

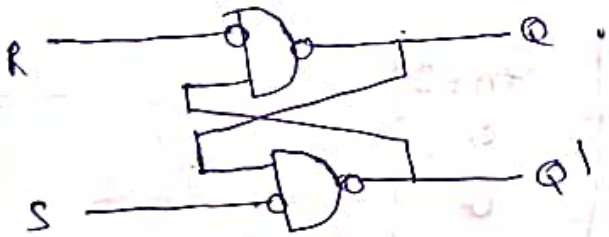
$= x' \cdot y'$

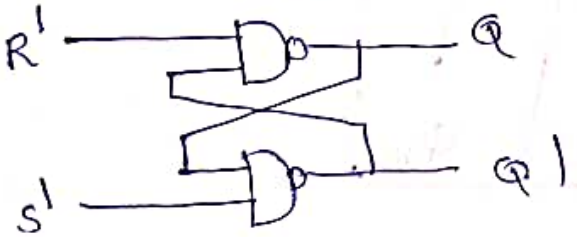




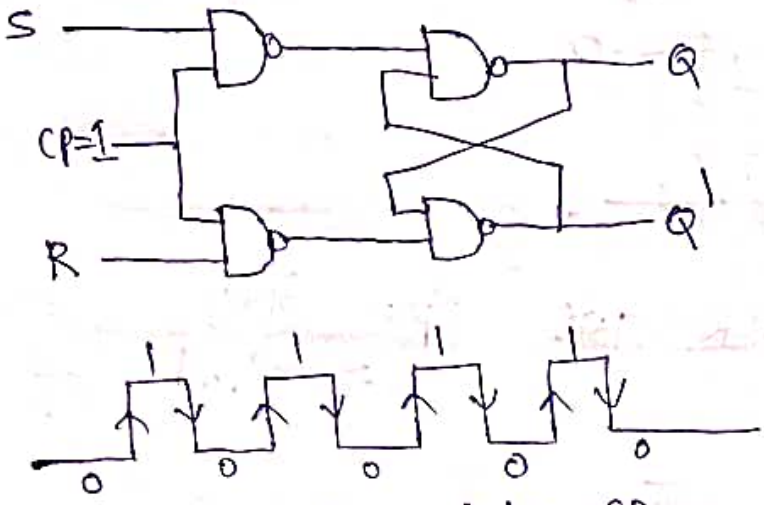
Bubbled AND Gate







Clocked RS-Flip Flop:-



CP	S	R	Q_{n+1}
0	0	0	Q_n
1	0	0	Q_n
0	0	1	Q_n
1	0	1	0
0	1	0	Q_n
1	1	0	1
0	1	1	Q_n
1	1	1	Indeterminate

0 - Inactive
1 - active

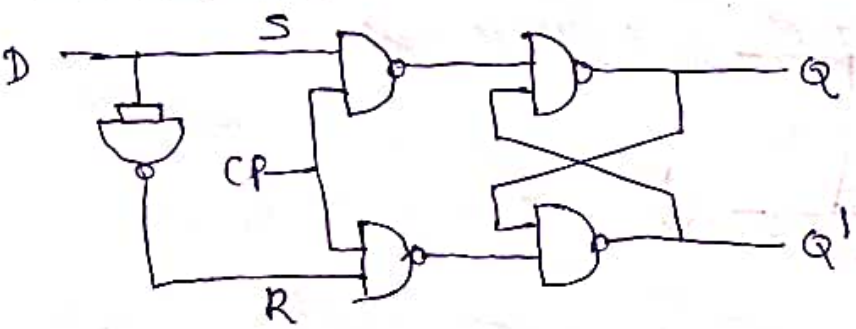
Level Triggred

+ve level triggred
-ve level triggred

Edge Triggred

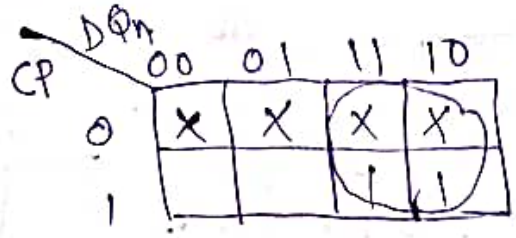
+ve edge triggred
-ve edge triggred.

D-Flip-Flop:-



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

Characteristic Equation:-



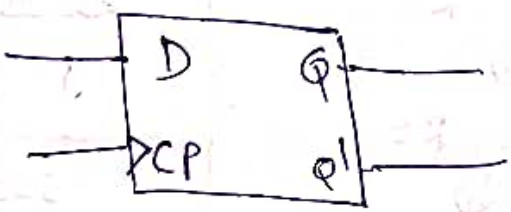
$Q_{n+1} = D$

0	11
0	10
1	11
1	10
D	

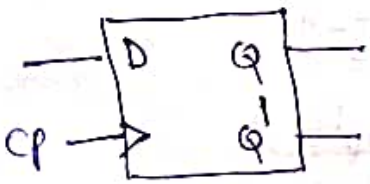
The no change and invalid conditions can be avoided by making S, R Compliment of each other. This modified SR Flip Flop is known as D-FlipFlop. As in above figure D input goes to the S input and its compliment is applied to the R through NOT gate. So only 2 input conditions occur.

These are either S=0 and R=1 or S=1 & R=0
The truth table is as shown in above figure.

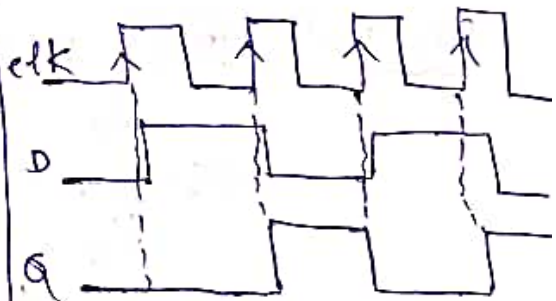
Logic Symbol is



Positive Edge Trigged D-Flip-Flop:-

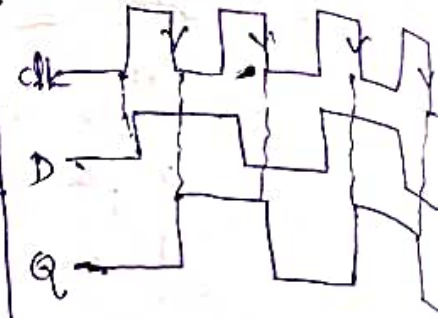


CP	D	Q_{n+1}
↑	0	0
↑	1	1
0	X	Q_n

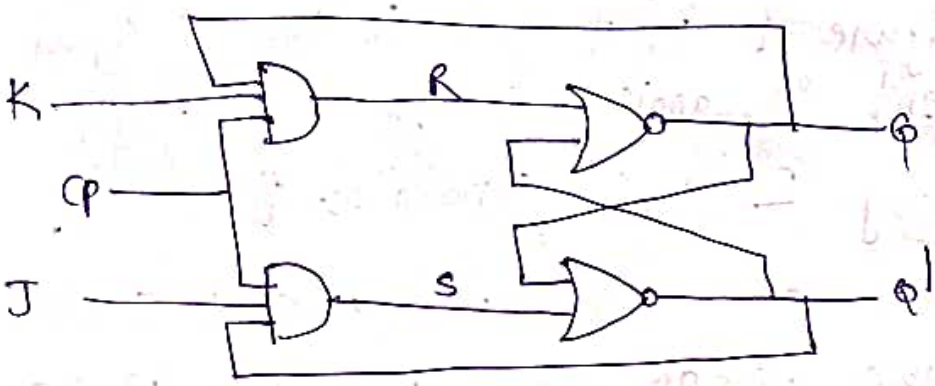


Negative Edge triggered D-Flip Flop:-

CP	D	Q_{n+1}
↓	0	0
↓	1	1
0	X	Q_n



JK- Flip Flop:-



Case-i:- J=0, K=0
⇓
S=0, R=0
⇓
No change (Q_n)

Case-ii:- J=0, K=1
⇓
S=0, R=1
⇓
Q=0 ; Q'=1

Case-iii:- J=1, K=0
⇓
S=1, R=0
⇓
Q=1 , Q'=0

Case-iv:- J=1, K=1

If Q=0, Q'=1
⇓
R=0, S=1
⇓
Q=1, Q'=0

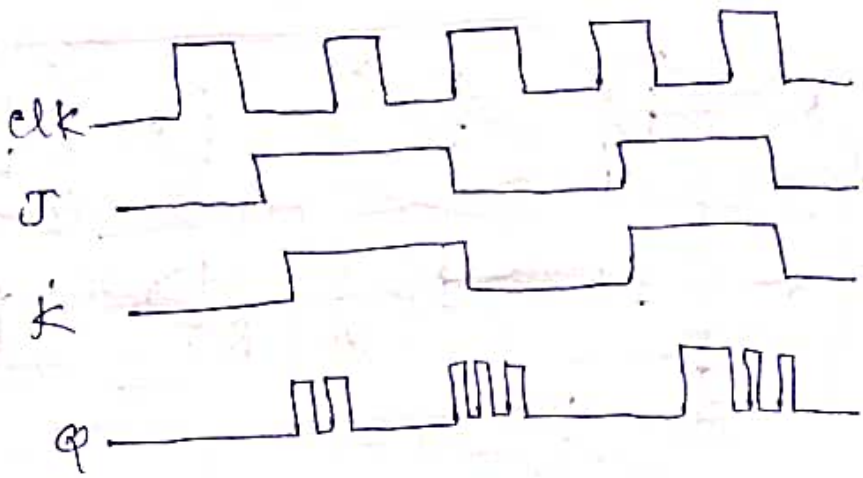
if Q=1, Q'=0
⇓
S=0, R=1
⇓
Q=0 Q'=1

↓ Toggle

Truth Table

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

Wave forms Timing Diagrams

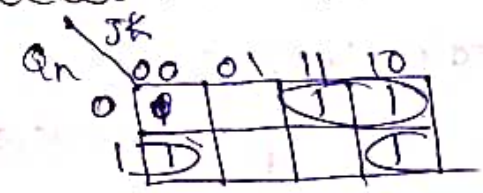


When $J=1$ and $K=1$ o/p of the flip flop will toggle on next positive clock edge. (Toggle means to switch to the compliment stage).

In this, JK-Flip Flop o/p is feedback to the input. ∴ change in the o/p results change in the input. Due to this in the two half of the clock cycle (or pulse, if J and K are both high then o/p toggles continuously as shown in points A, B, C in above o/p wave forms. This condition is known as race around condition and must be avoided.

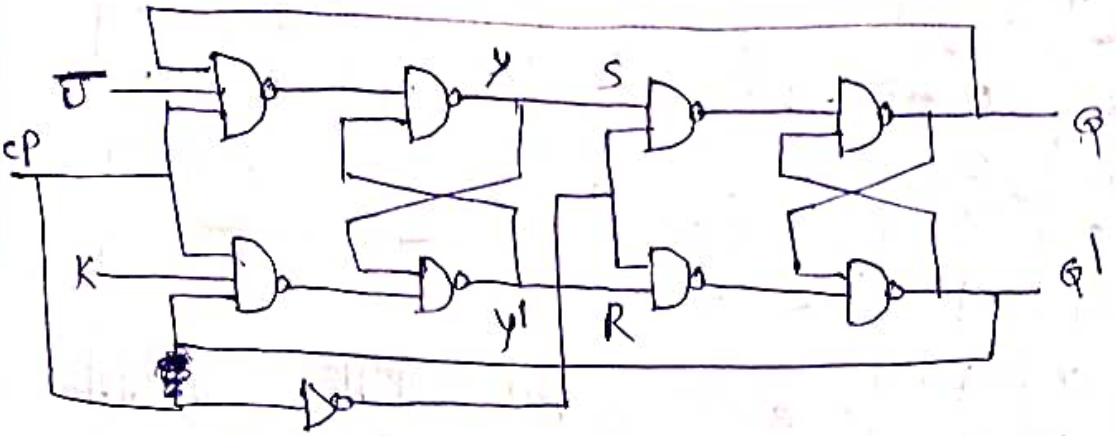
011 100 010
010 110 110

Characteristic Equation:-



$$Q_{n+1} = Q_n' J + Q_n K'$$

JK Master-slave Flip flop



J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

Case i:- $J=0, K=0$

⇒ +ve clock pulse

Y & y' same as previous state

Slave is inactive.

⇒ -ve clock pulse

Master is inactive.

Slave is active.

Slave copies the o/p of master o/p Q_n not change.

Case ii:- $J=0, K=1$.

⇒ +ve clock. master o/p is $Y=0$ & $y'=1$

Slave is inactive.

⇒ -ve clock. Slave is active.

o/p is $Q=0$, $Q'=1$.

Master is inactive.

Case iii:- $J=1, K=0$

⇒ +ve clock. Master o/p $Y=1$ & $y'=0$.

Slave is not copying the o/p of master.

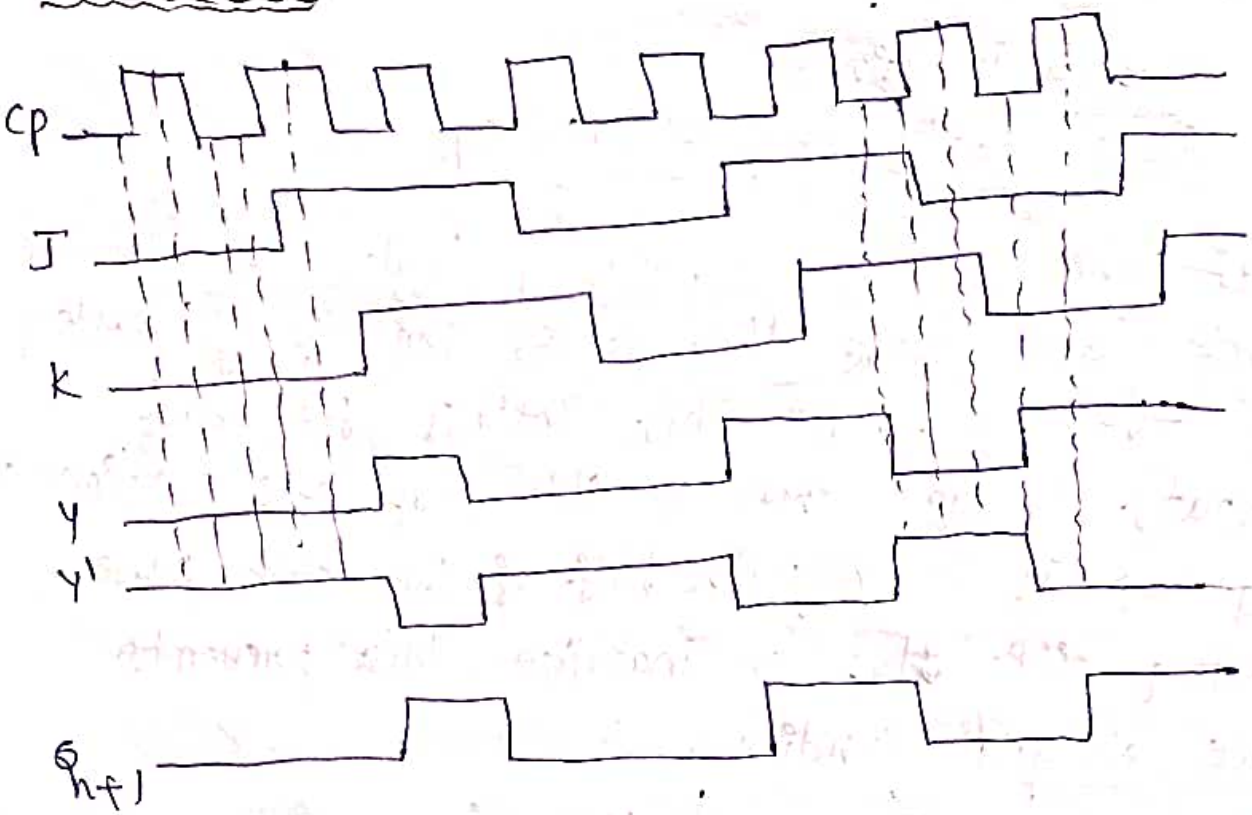
⇒ -ve clock pulse.
master inactive.
Slave copies o/p of master.

case-IV: when J=1, K=1. Master toggles on the +ve clock and slave then copies the o/p of master on -ve clock. At this instant feed back inputs to the master flip flop are complimented But as it is negative half of the clock pulse master flip flop is inactive. This prevents race around condition.

CP	Q_n	J	K	Y	Q_{n+1}
⊓	0	0	0	0	NC
⊓	0	0	0	NC	0
⊓	0	0	1	0	NC
⊓	0	0	1	NC	0
⊓	0	1	0	0	NC
⊓	0	1	0	NC	0
⊓	0	1	1	0	NC
⊓	0	1	1	NC	0
⊓	1	0	0	0	NC
⊓	1	0	0	NC	0
⊓	1	0	1	0	NC
⊓	1	0	1	NC	0
⊓	1	1	0	0	NC
⊓	1	1	0	NC	0
⊓	1	1	1	0	NC
⊓	1	1	1	NC	0

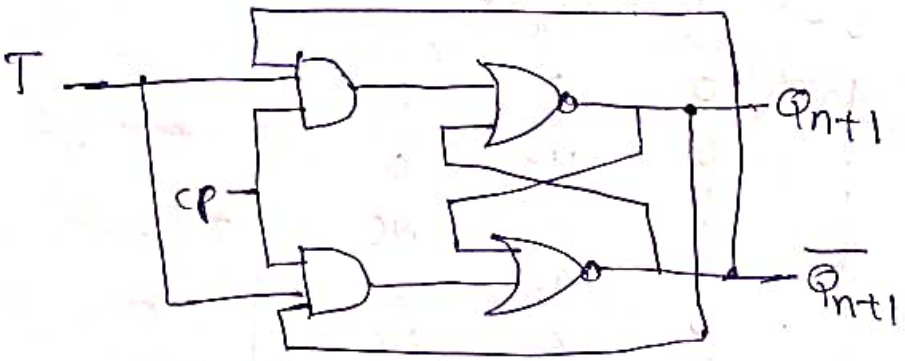
NC = No Change.

Waveforms :-



T - Flip Flop :-

Here, T means "Toggle".



Case i :- (a) $T=0$, $Q_n=0$

$Q_{n+1}=0$, $\overline{Q}_{n+1}=1$

(b) $T=0$, $Q=1$

$Q_{n+1}=1$, $\overline{Q}_{n+1}=0$.

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

Case ii :- $T=1$, $Q_n=0$, $Q_{n+1}=1$

$T=1$, $Q_n=1$, $Q_{n+1}=0$.

SR Flip Flop:-

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

JK Flip Flop:-

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

D-Flip Flop:-

D	Q_{n+1}
0	0
1	1

T Flip Flop:-

T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

Flip Flop Conversion:-

It is possible to convert one flip flop into another. The conversion of flip flop is obtained by using some additional gates or simply doing some extra connection.

SR-Flip Flop to D-Flip Flop:-

Inputs	Present state	Next state	Flip flops inputs	
D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

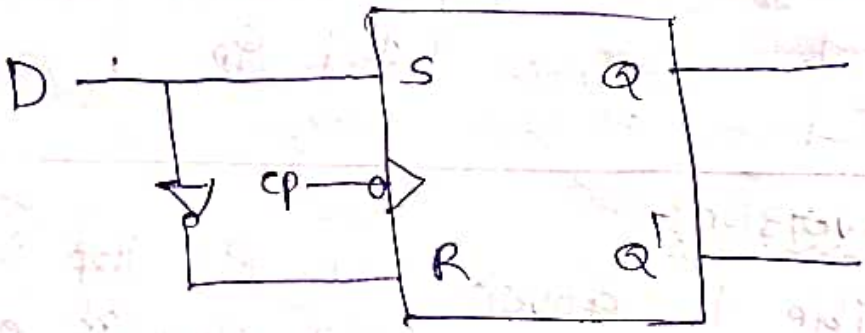
By using K-map

⇒ for S

$S = D$

⇒ for R

$R = D'$



SR-Flip Flop to JK-Flip Flop

Inputs		Present state	next state	Flip Flops inputs	
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

By using K-map,

⇒ for S

JK \ Q_n	00	01	11	10
0		X		
1	1	X		1

100
110
JQ_n'

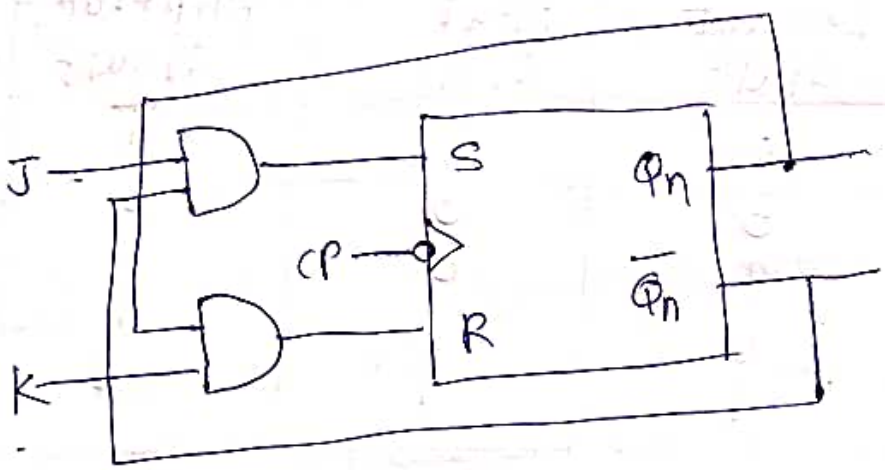
⇒ for R

JK \ Q_n	00	01	11	10
0	X		1	X
1			1	

011
111
KQ_n

$S = JQ_n'$

$R = KQ_n$

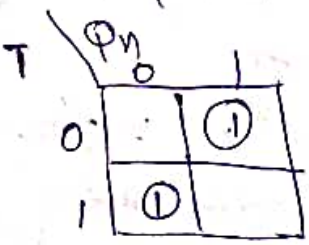


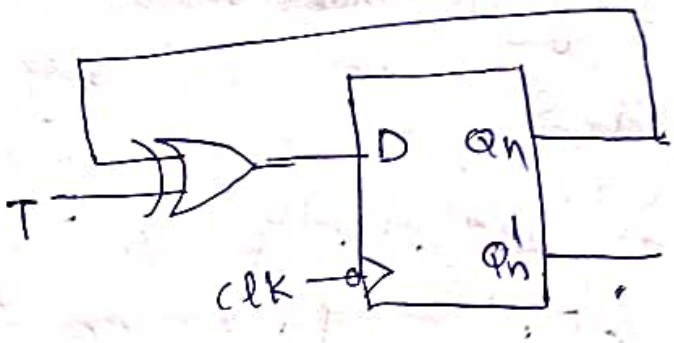
Convert D Flip Flop to T-Flip Flop:-

Inputs	Present state	Next state	Flip Flops Inputs
T	Q_n	Q_{n+1}	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

By using K-map

for D





$D = T' Q_n + T Q_n'$

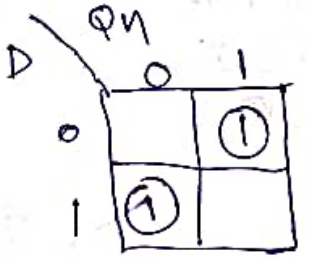
$D = T \oplus Q_n$

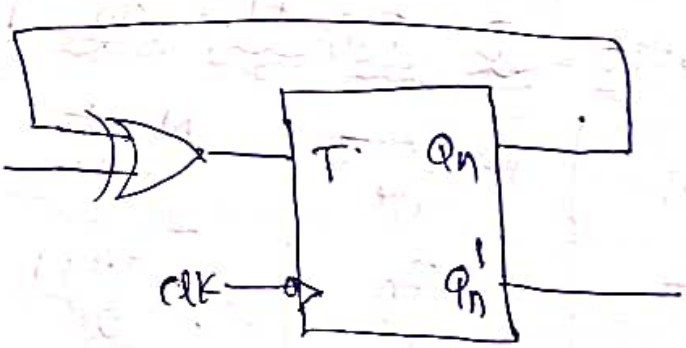
Convert T-Flip Flop to D-Flip Flop:-

Inputs	Present State	Next State	Flip Flop Inputs
D	Q_n	Q_{n+1}	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

By using K-map

for T





$T = D Q_n' + D' Q_n$

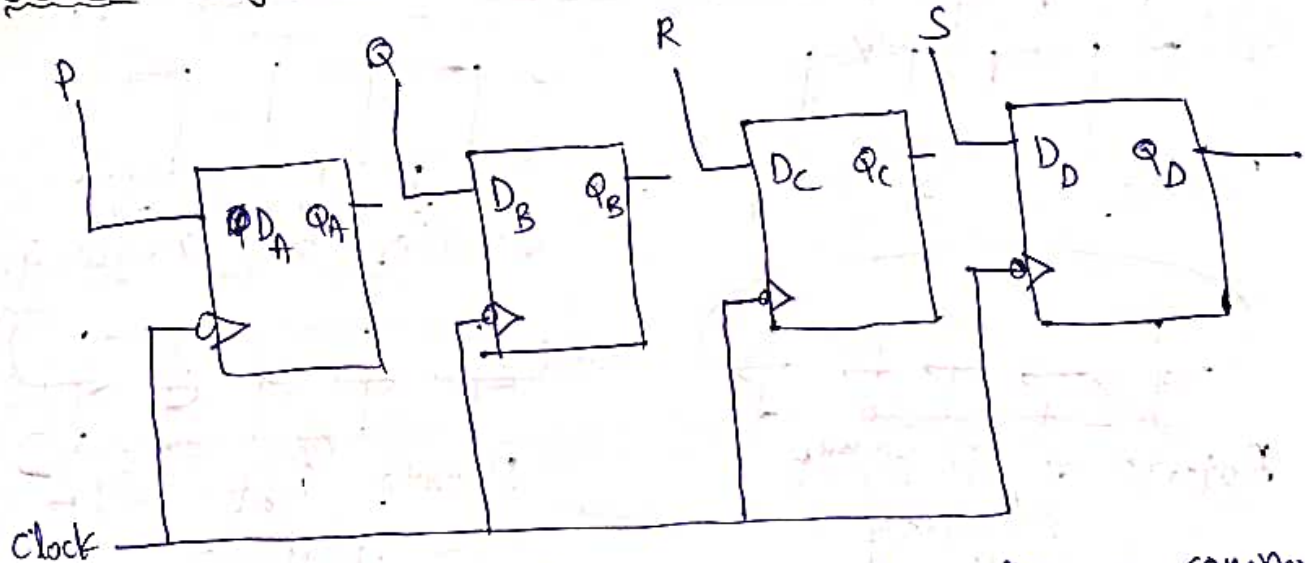
Register:-

A flip flop is nothing but a binary cell capable of storing 1 bit information and can be connected together to perform counting operation such a group of flip flops is called a counter.

The group of flip flops can be used to store a word which is called a register.

An n-bit register has a group of n-flip flops and is capable of storing any binary information / number containing n-bits.

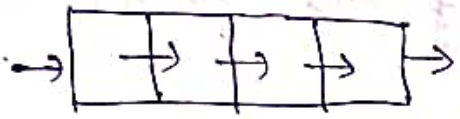
Buffer Register:-



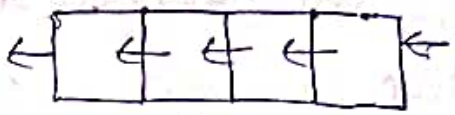
Each D flip flop is striggred with a common -ve clock pulse. The input x-bits set up the flip flop for loading when first -ve clock edge arrives then the stored binary information becomes $Q_A Q_B Q_C Q_D = PQRS$

Shift Register:-

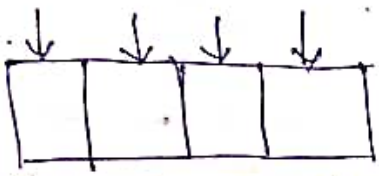
The binary information (or) data in a register can be moved from stage to stage within the register (or) in to (or) out of register upon applicatio of clock pulse. These groups of register are shifting the data called as shift registers. The are useful in micro processors and in digital systems.



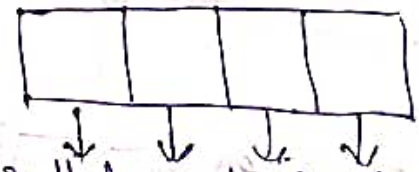
Serial shift right Register.



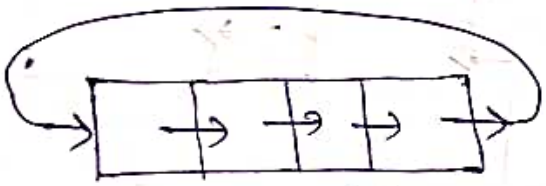
Serial shift left register.



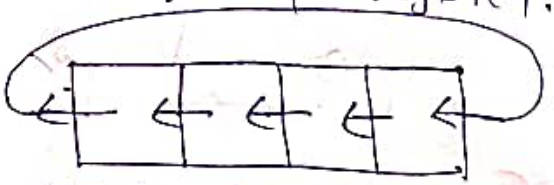
Parallel in register



Parallel out register.



Rotate right shift register



Rotate left shift register.

Types of shift Registers:-

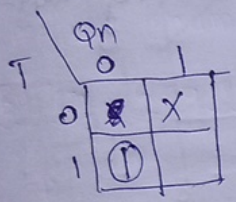
There are mainly 4 types of shift registers are present.

(i) Serial Input Serial Output register (SISO)
(ii) Serial Input Parallel Output register (SIPO)
(iii) Parallel Input serial Output register (PISO)
(iv) Parallel Input Parallel Output register (PIPO)

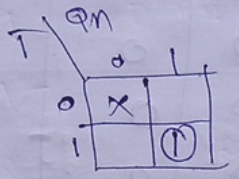
Convert SR Flip Flop to T Flip Flop:

Inputs	Present State	Next state	Flip Flop inputs		
T	Q_n	Q_{n+1}	S	R	
0	0	0	0	X	0
0	1	1	X	0	1
1	0	1	1	0	2
1	1	0	0	1	3

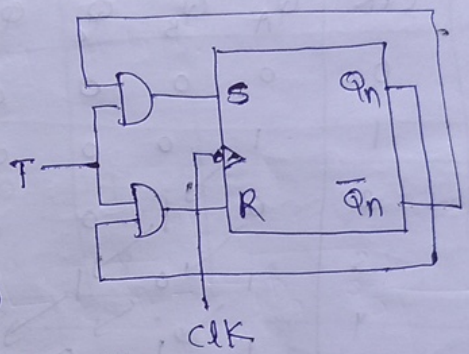
for S ... for R



$S = TQ_n'$



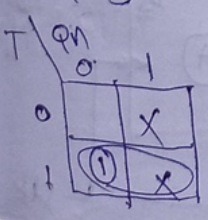
$R = TQ_n$



Convert JK Flip Flop to T-Flip Flops:

Inputs	Present state	Next state	Flip Flop inputs		
T	Q_n	Q_{n+1}	J	K	
0	0	0	0	X	0
0	1	1	X	0	1
1	0	1	1	X	2
1	1	0	X	1	3

for J

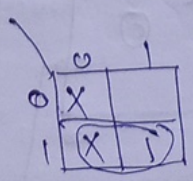


10
11
T

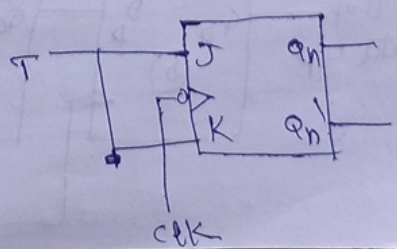
Q_0 Q_1

$J = T$

for K

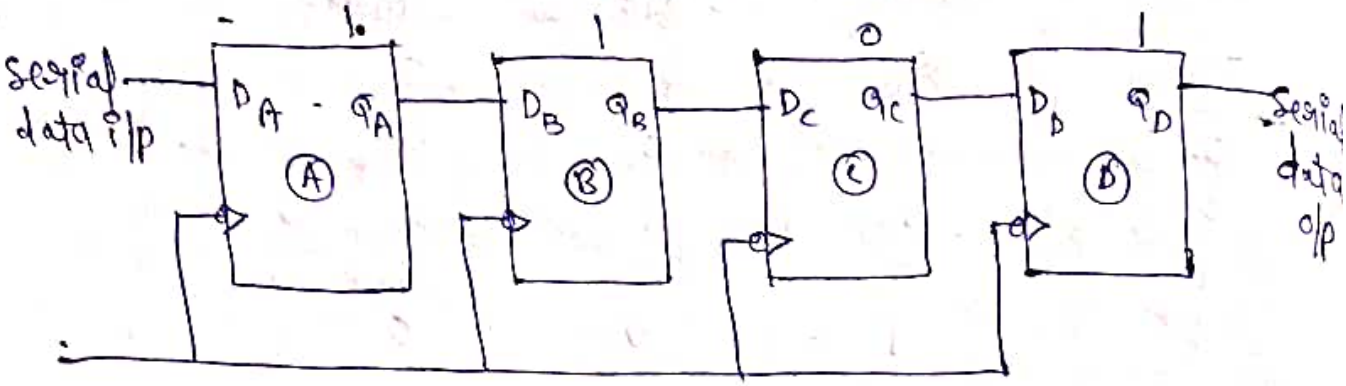


$K = T$.



SISO Right shift Register:-

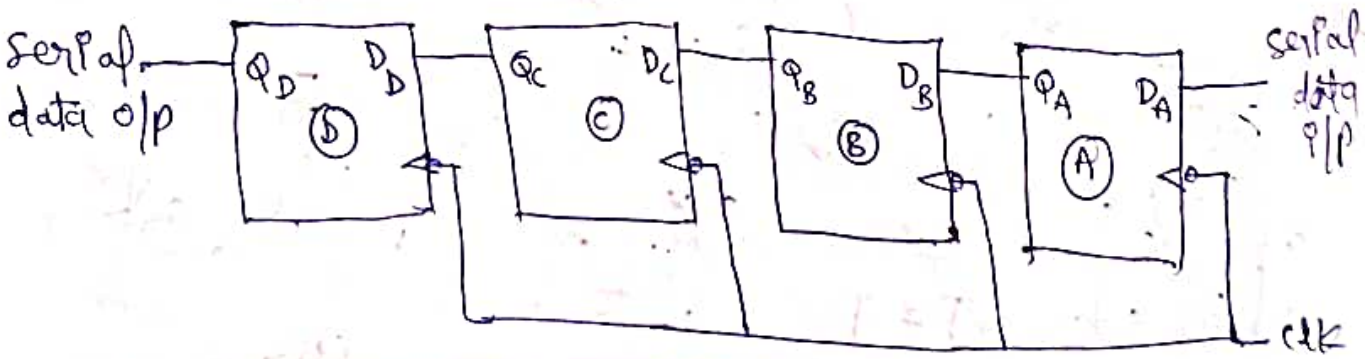
1101



CLK	Q_A	Q_B	Q_C	Q_D	o/p
0	0	0	0	0	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	0	1	0	0
4	1	1	0	1	0
5	0	1	1	0	1
6	0	0	1	1	0
7	0	0	0	1	1
8	0	0	0	0	1

Shifting data = 1101

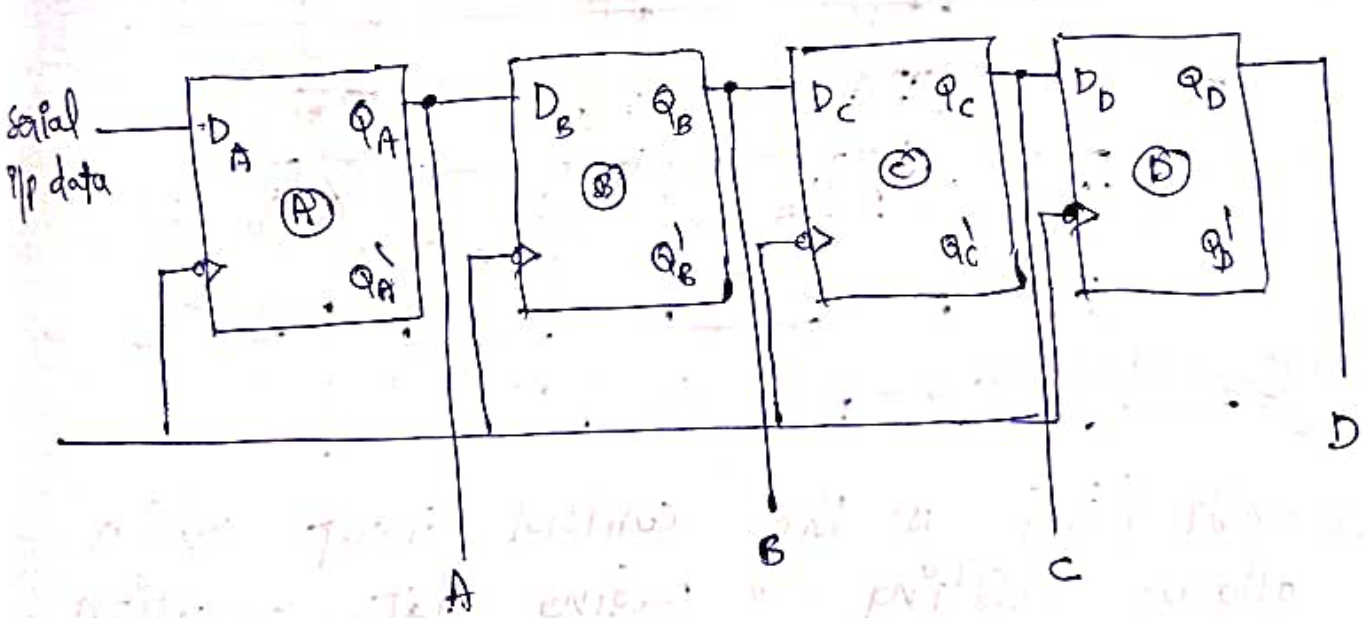
SISO Left shift Register:-



1101

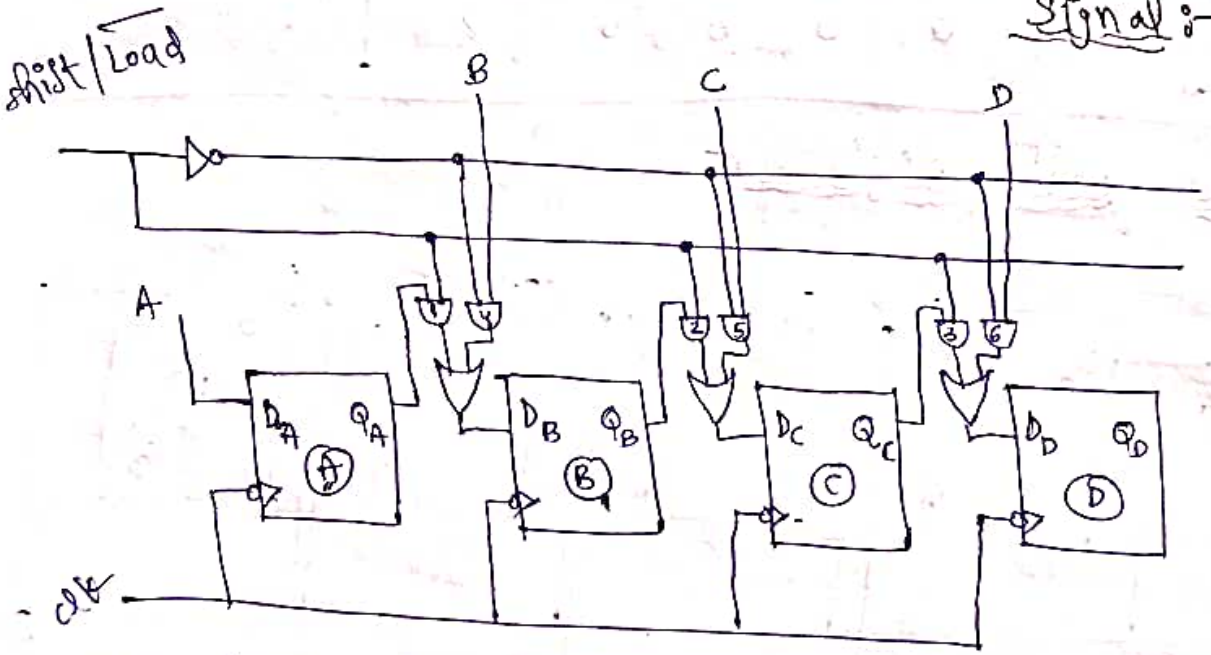
O/P	Q_D	Q_C	Q_B	Q_A	clk
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
0	0	1	0	1	3
0	1	0	1	1	4
1	0	1	1	0	5
0	1	1	0	0	6
1	1	0	0	0	7
1	0	0	0	0	8

SIPO shift Register:-



clk	Q_A	Q_B	Q_C	Q_D	Outputs A	B	C	D
0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0
2	0	1	0	0	0	1	0	0
3	1	0	1	0	1	0	1	0
4	1	1	0	1	1	1	0	1
5	0	1	1	0	0	1	1	0
6	0	0	1	1	0	0	1	1
7	0	0	0	1	0	0	0	1
8	0	0	0	0	0	0	0	0

PISO shift Register using shift / $\overline{\text{Load}}$ Control Signal :-



(i) shift / $\overline{\text{load}}$ is the control input which allows shifting & loading data operation of the register.

(ii) When shift / $\overline{\text{load}}$ is low, gates 4,5,6 are enabled. Allowing each input data bit to be

applied D input of respective flip flop.

(iii) When cp applied with D=1, will set and those with D=0 will reset. Thus all 4 bits are stored simultaneously.

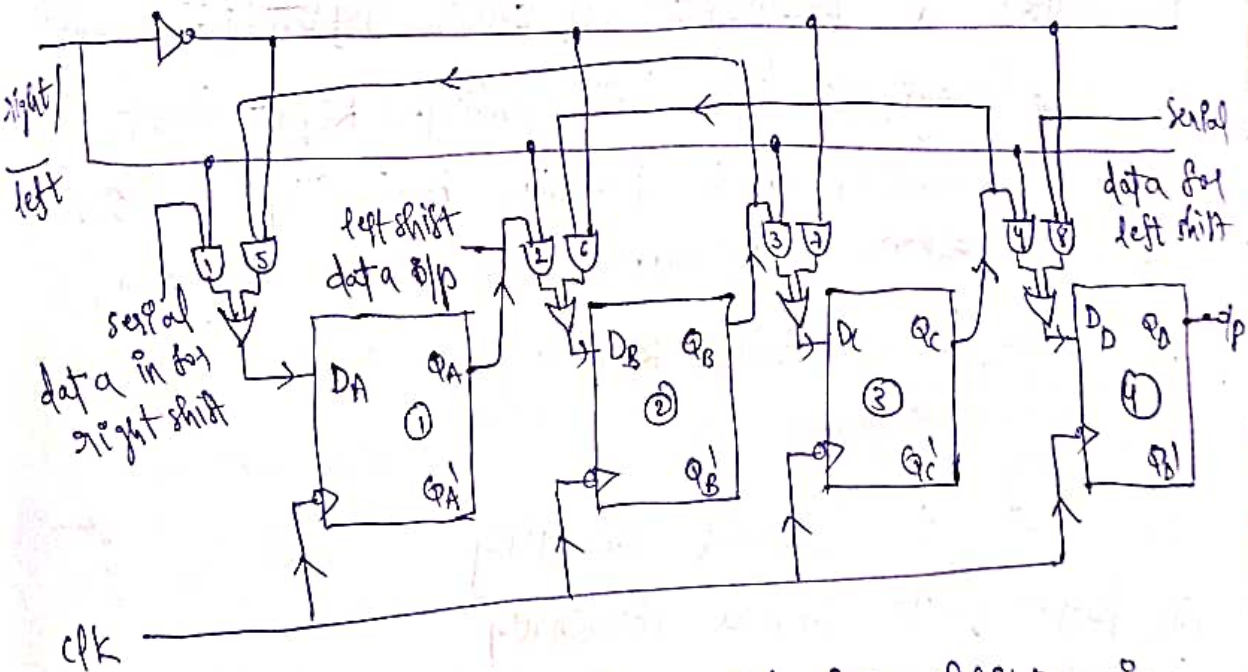
(iv) When shift/$\overline{load}$ is high, gates 4,5,6 are disabled remaining AND gates are enabled.

(v) OR gate at the D input of the flip flops allow either parallel data operations or shift operation depending on AND gates enabled by shift/$\overline{load}$ control input.

Bi-directional shift Register:-

(i) The below diagram shows 4 stages bi-directional shift register.

(ii) This bi-directional shift register shifts in both direction i.e, right and left directions.



(iii) Right/$\overline{left}$ control line is used for shifting in one the direction.

(iv) Right/$\overline{\text{left}}$ = 1 AND gates 1, 2, 3, & 4 are enable & remaining AND gates disabled.

(v) A upon the giving clock pulse to all D-flip flops & the input data at D transfers to o/p Q.

(vi) While right shifting the data, the serial data moves are shifts AND gates 1, 2, 3, 4 through the D-flip flops. The o/p data collected at right most flip flop.

(vii) When right/$\overline{\text{left}}$ = 0, AND gates 5, 6, 7 & 8 are enable & used for left shift

---

Universal shift Register:-

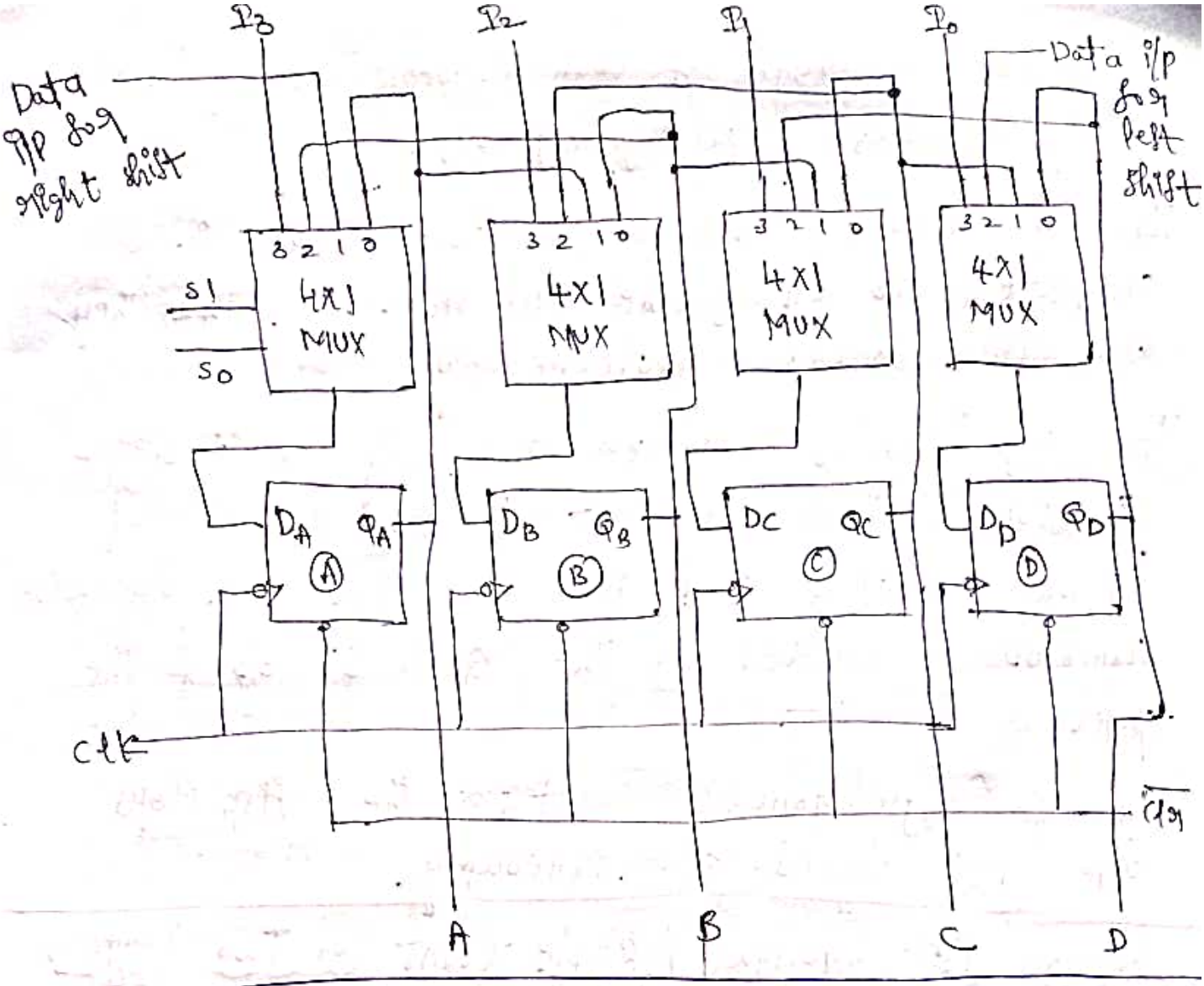
A register capable of shifting in one direction only is called a uni-directional shift register.

A register capable of shifting in both directions is called a bi-directional shift register.

If the register has both shifts (Right shift and Left shift) with parallel load capabilities. It is referred as universal shift register.

Applications of shift Registers:-

(i) Delay line
(ii) Serial to Parallel Converter
(iii) Parallel to Serial Converter
(iv) shift register Counter.
(v) Pseudo Random binary sequence generators.



Counters:

A counter is a register capable of counting the number of clock pulses arriving at its clock input. Count represents the number of clock pulses arrived.

A specified sequence of states appears as the counter o/p this is the main difference b/w a register and a counter.

A specified sequence of states is different for different types of counters.

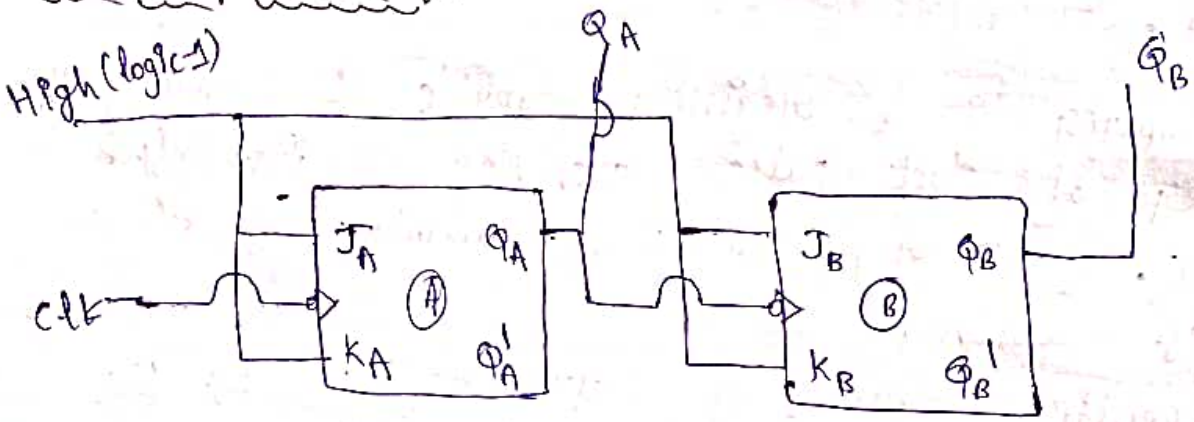
There are 2 types of Counters.
(i) Synchronous (ii) Asynchronous.

In synchronous counter the clock input is connected to all of the flip flops, and thus they are called clocked simultaneously.

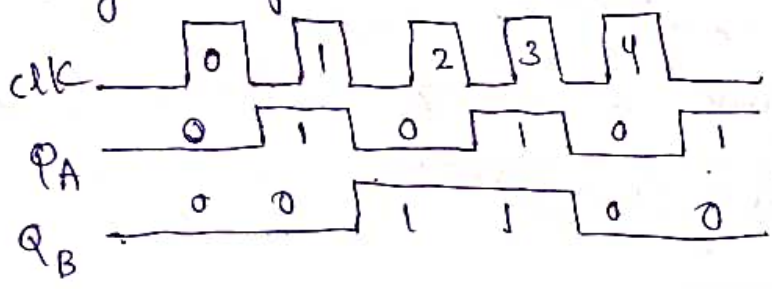
In asynchronous counter commonly called ripple counter in this the first flip flop is clocked by the external clock pulse and then each successive flip flop is clocked by the Q (or) Q' o/p of the previous flip flop

In an asynchronous counters the flip flops are not clocked simultaneously.

Binary Asynchronous / Ripple Count (or) Two stage (or) MOD-4 Counter :-

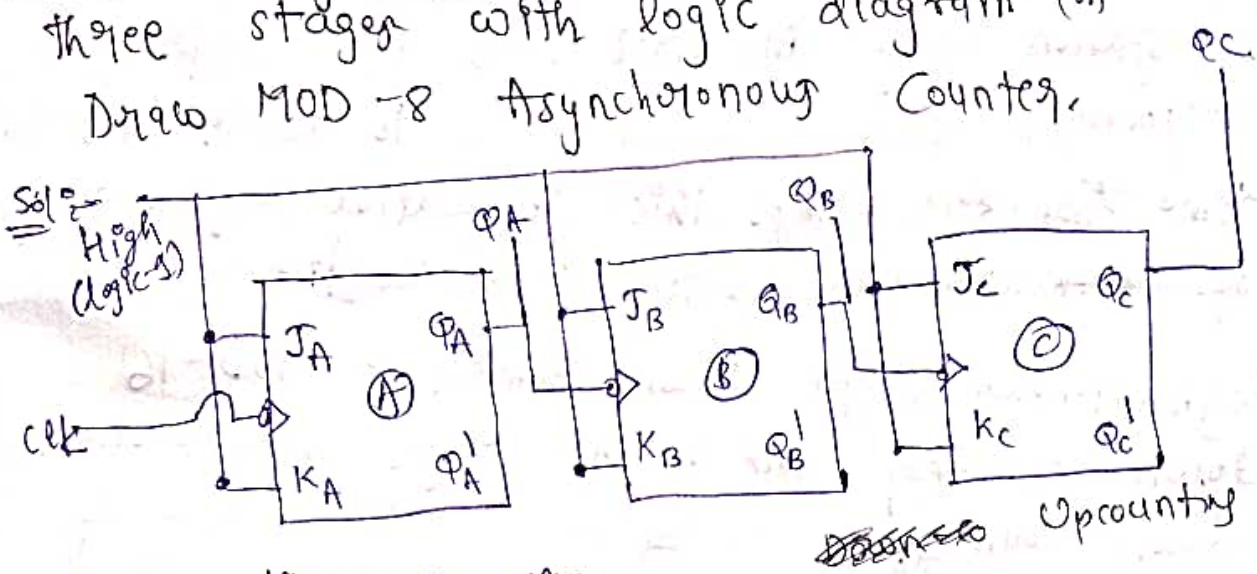


Timing Diagrams are

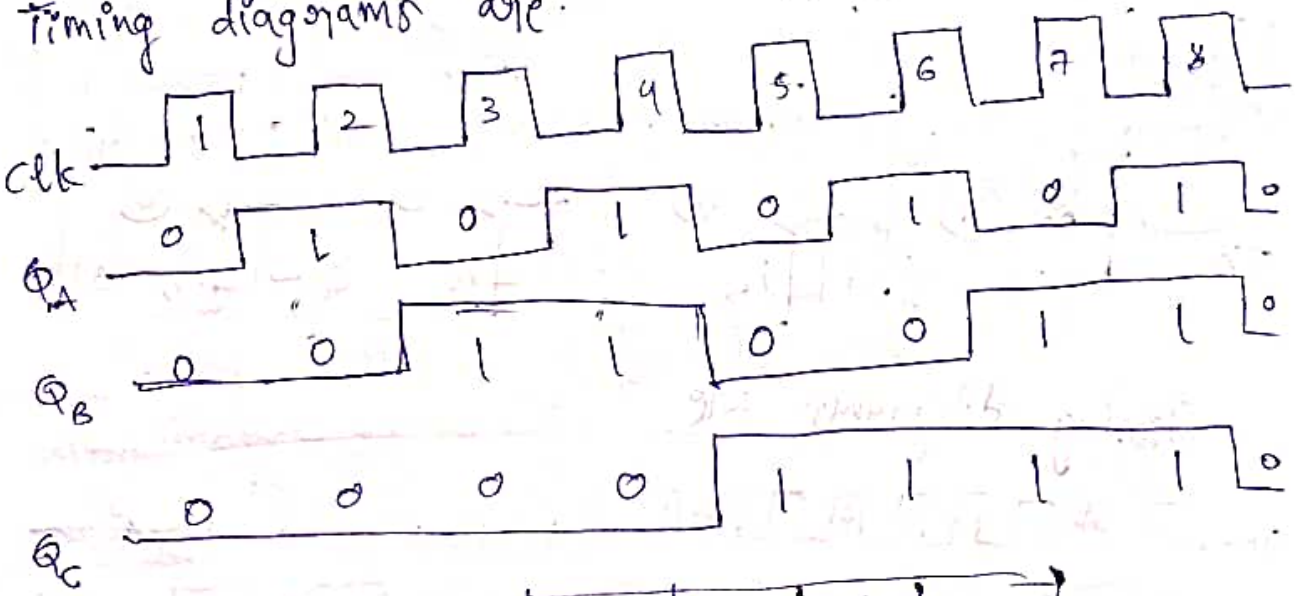


clk	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0

① Draw Asynchronous (or) Ripple Counter for three stages with logic diagram (or) Draw MOD-8 Asynchronous Counter.

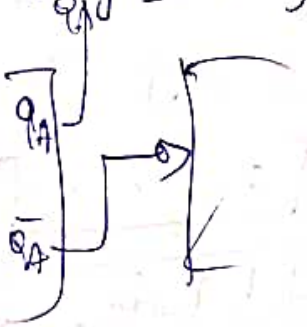


Timing diagrams are.



clk	Q_C	Q_B	Q_A
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

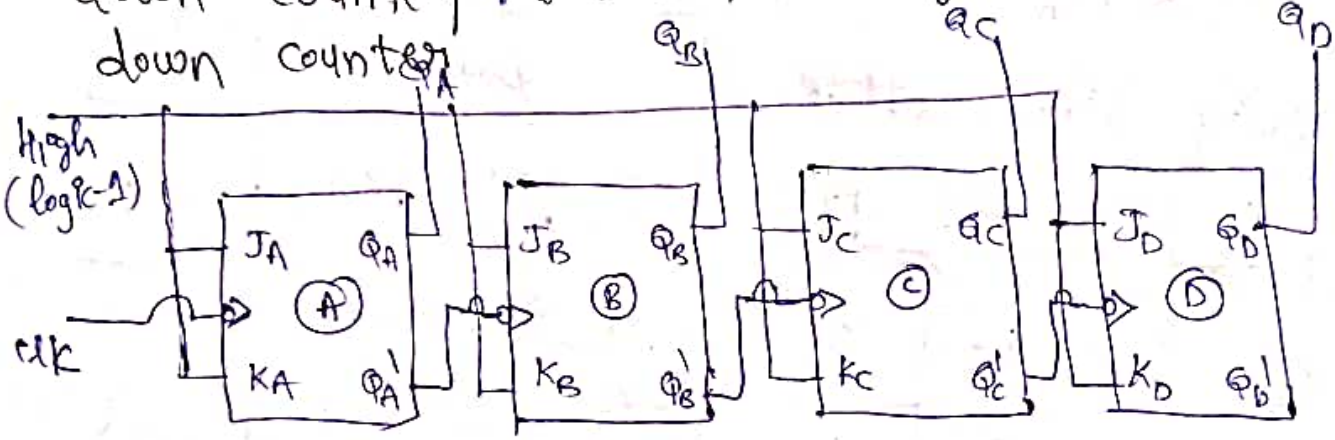
Ily, Draw down counter using 3 stages



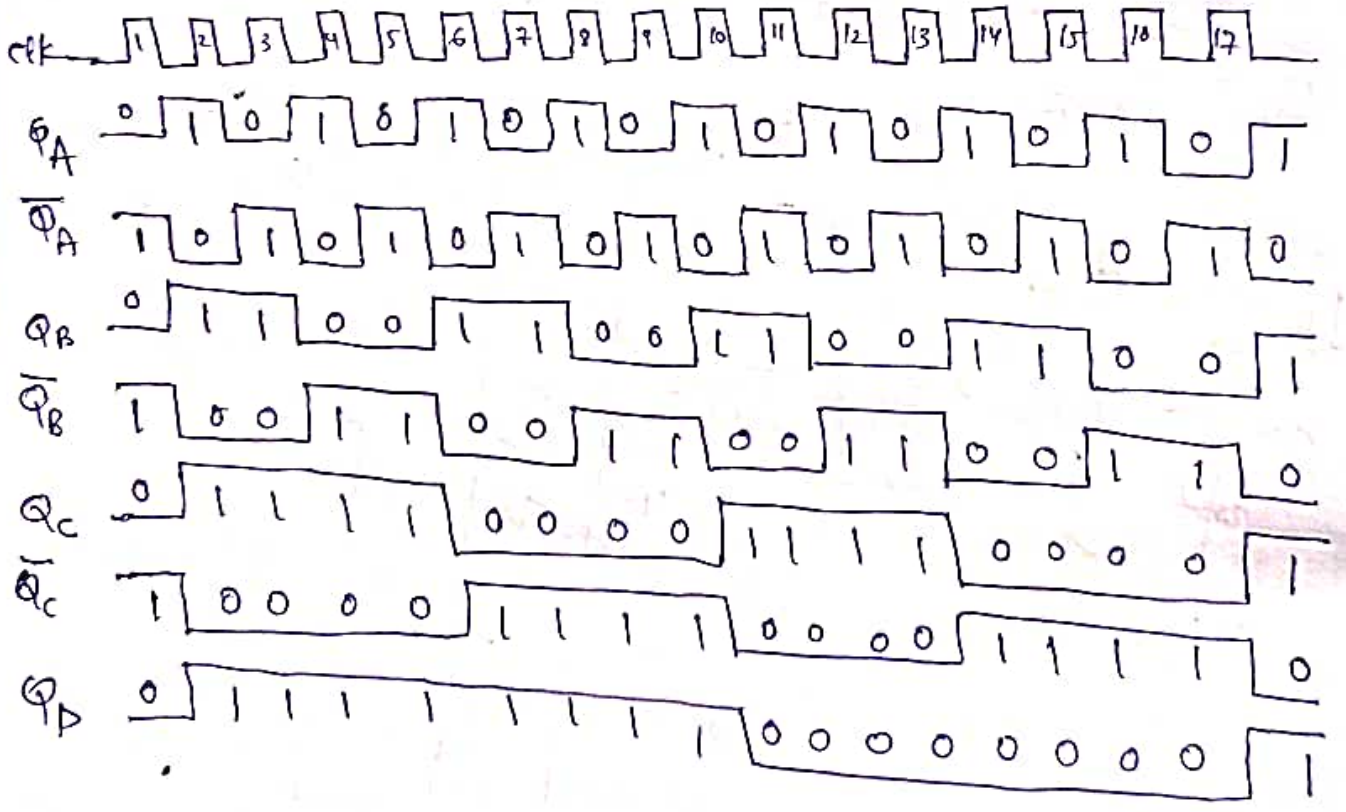
Disadvantage in Asynchronous Counters

Propagation delay in first stage is added in the propagation delay of second stage to deside transition time for 3rd stage this cumulative major delay of asynchronous counter is a disadvantage.

② Asynchronous Ripple Down Counter (or MOD-16 down counter. (or 4 stage asynchronous down counter



Timing diagrams are



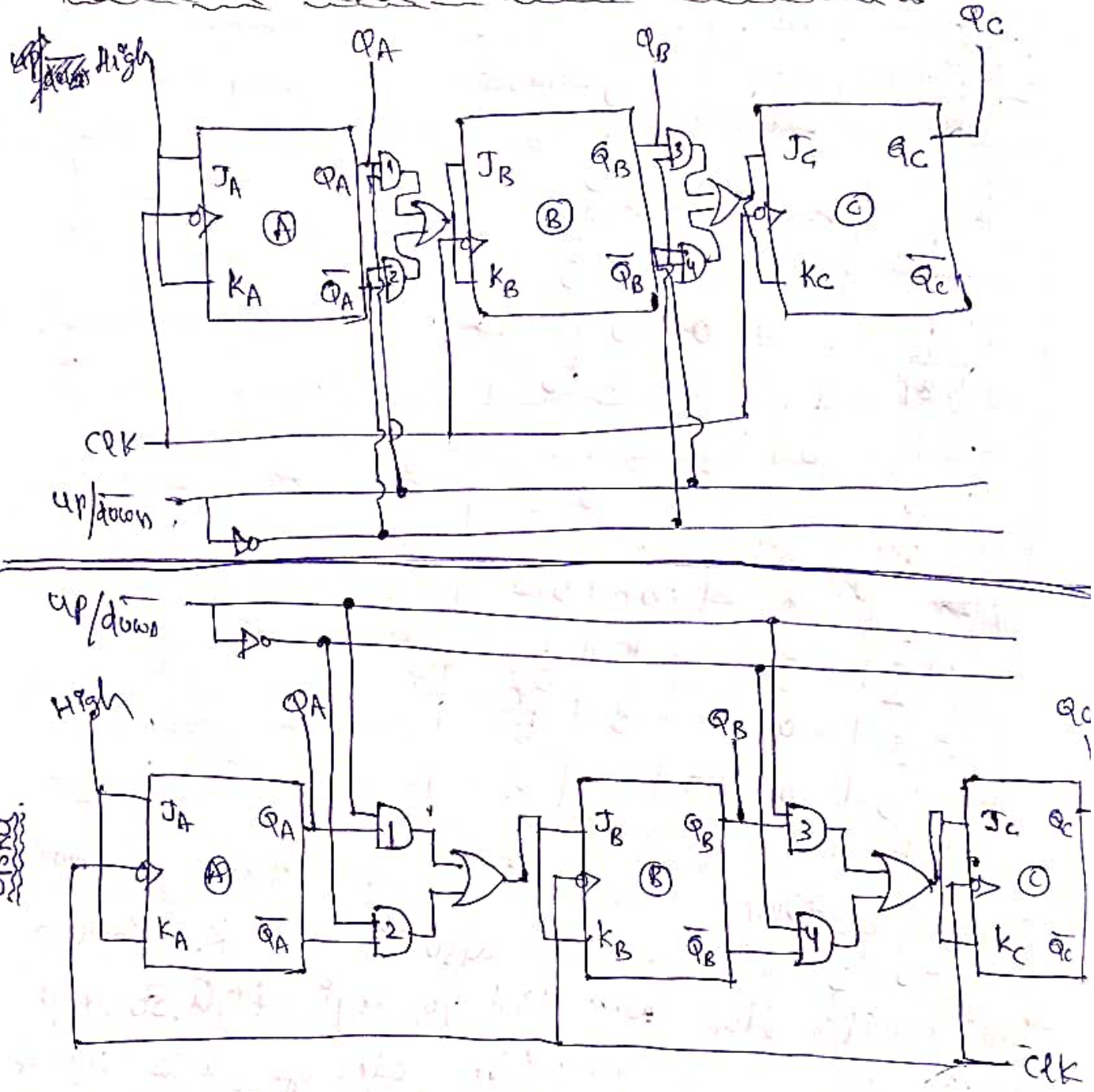
clk		Q_D	Q_C	Q_B	Q_A
1	15	1	1	1	1
2	14	1	1	1	0
3	13	1	1	0	1
4	12	1	1	0	0
5	11	1	0	1	1
6	10	1	0	1	0
7	9	1	0	0	1
8	8	1	0	0	0
9	7	0	1	1	1
10	6	0	1	1	0
11	5	0	1	0	1
12	4	0	1	0	0
13	3	0	0	1	1
14	2	0	0	1	0
15	1	0	0	0	1
0	0	0	0	0	0

(i) The ripple down counter is count from a maximum count to "zero". In this also J and K inputs of JK flip flop are tied to logic high. So o/p toggle for each negative edge of clock signal.

(ii) Here the clock signal is connected to the clock input of only first flip flop.

(iii) This connection is same as asynchronous ripple up counter but the clock input of the remaining flip flop is triggered by $\overline{Q}_A$ o/p of the previous stage. instead of Q_A o/p of previous stage.

Synchronous up-down counter (3-stage):



(i) A parallel counter (synchronous counter) (or synchronous down counter are using inverted flip flop outputs to drive the following JK inputs.

(ii) The parallel up counter can be converted to a down counter by connecting $\bar{Q}_A$, $\bar{Q}_B$, $\bar{Q}_C$ output in place of Q_A, Q_B and Q_C respectively

(iii) The up/$\overline{\text{down}}$ is used to control whether the normal flip flop outputs are the inverted flip flop output are fed to the J and K inputs of the following flip flops.

(iv) The up-down counter that will count from 000 upto 111 when the up/$\overline{\text{down}}$ control input is 1. and from 111 down to 000 when the up/$\overline{\text{down}}$ control input is 0.

(v) A logic-1 on the up/$\overline{\text{down}}$ enables AND gates 1 and 3 and disables AND gates 2 and 4. This allows Q_A and Q_B outputs through to the J and K inputs to the next flip flop.

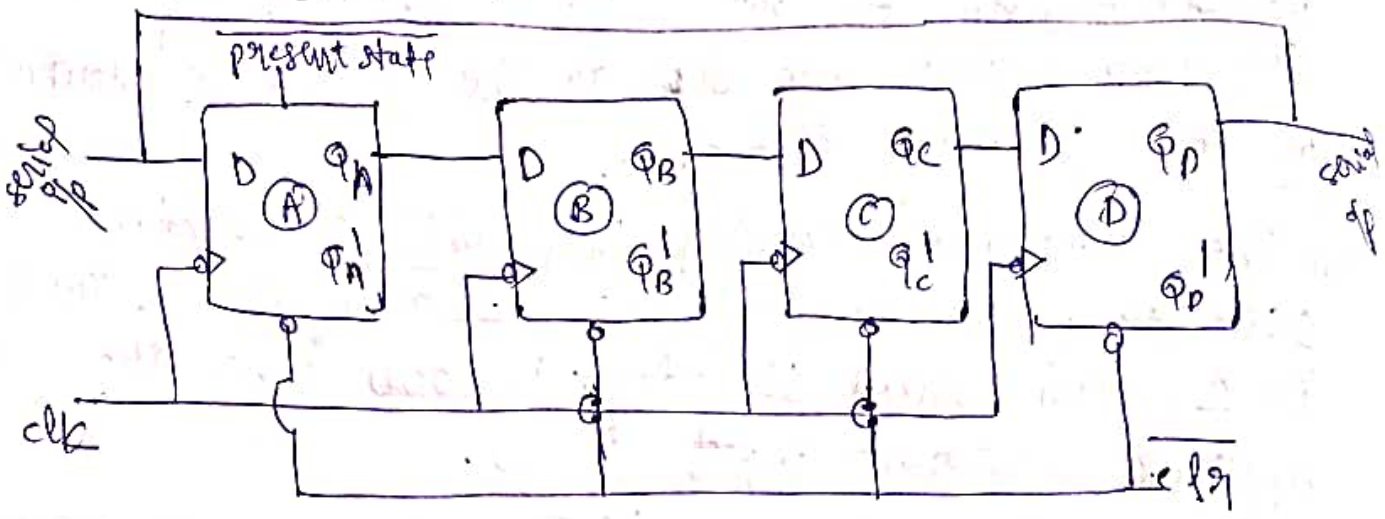
(vi) So that the counter will count up as pulses are applied.

(vii) When up/$\overline{\text{down}}$ line is logic-0 AND gates 1 and 3 are disables and AND gates 2 and 4 are enables. This allows the $\overline{Q}_A$, $\overline{Q}_B$ outputs through to the J and K inputs of the next flip flop. So that the counter will count down as pulses are applied.

~~Ring shift registers counters~~

~~Shift Registers Counters~~

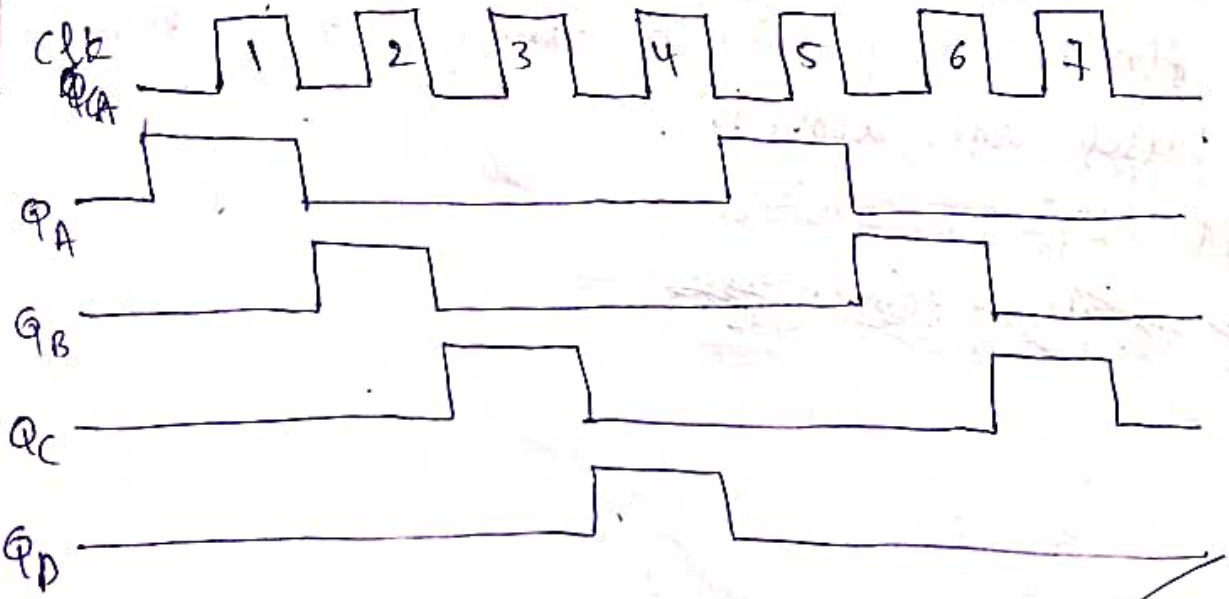
Shift Register Counters:



clk	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
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1

It is Ring Counter

Timing Diagram:



A shift register with the serial output connected back to the serial input is called shift register counter. The most commonly used shift register counters are

(i) Ring Counter
(ii) Twisted Ring Counter / Johnson Counter.

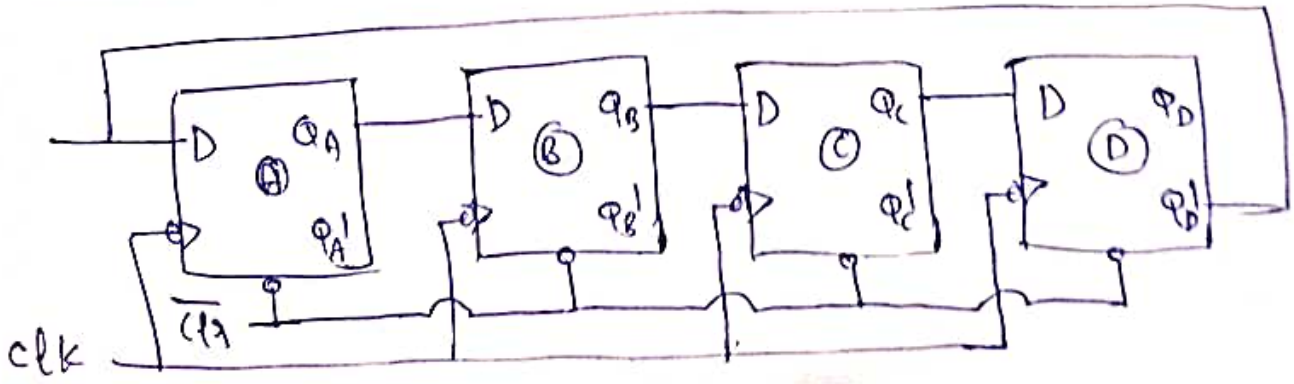
In the ring counter of above logic diagram. The Q output of each stage is connected to the D input of the next stage and o/p of the last stage Q_D is feed back to the input of the first stage.

The $\overline{clr}$ followed by $\overline{pre}$ makes the o/p of the first stage to 1 and remaining outputs are zero.

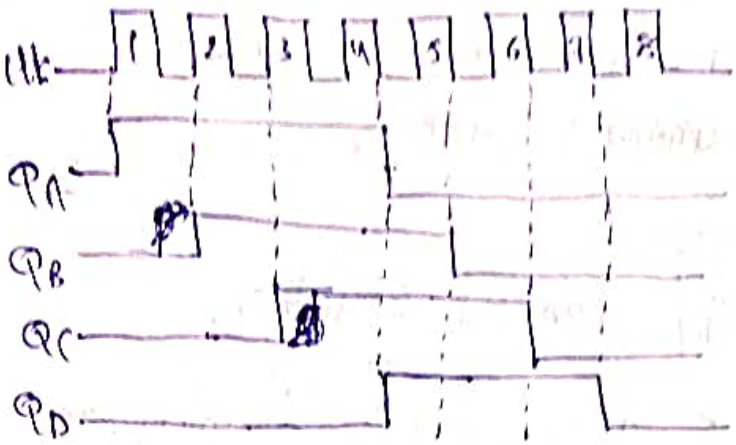
i.e, Q_A is 1 and Q_3, Q_C, Q_D are 0's.

This is repeated after 4 states. So, for n. no of stages 'n' states are occured and count 'n' no. of clock pulses.

Johnson Counter (or) Twisted Ring Counter:-



Timing Diagram



clk	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
8	0	0	0	0

In the ring counter the Q output of each stage of flipflop is connected to the D input of the next stage. The single exception in johnson counter is that the complement of the last flipflop is connected back to the D input of the first D flipflop as shown in above fig.

There is a feedback from the right most flipflop complement output to the left most flipflop input. Here for n no. of flip flop 2^n states counting is possible as shown in fig.