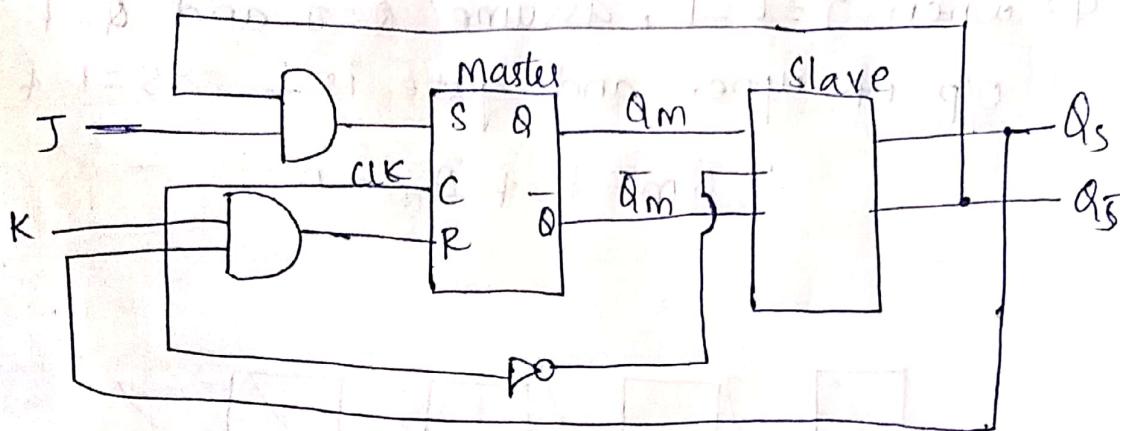


Master slave J-K FlipFlop:-

logic diagram:



Functional Table:

J	K	CLK	Q +	$\bar{Q} +$	logic symbol
X	X	Q	Q	\bar{Q}	
0	0	-	Q	\bar{Q}	
0	1	-	0	1	
1	0	-	1	0	
1	1	-	\bar{Q}	Q	

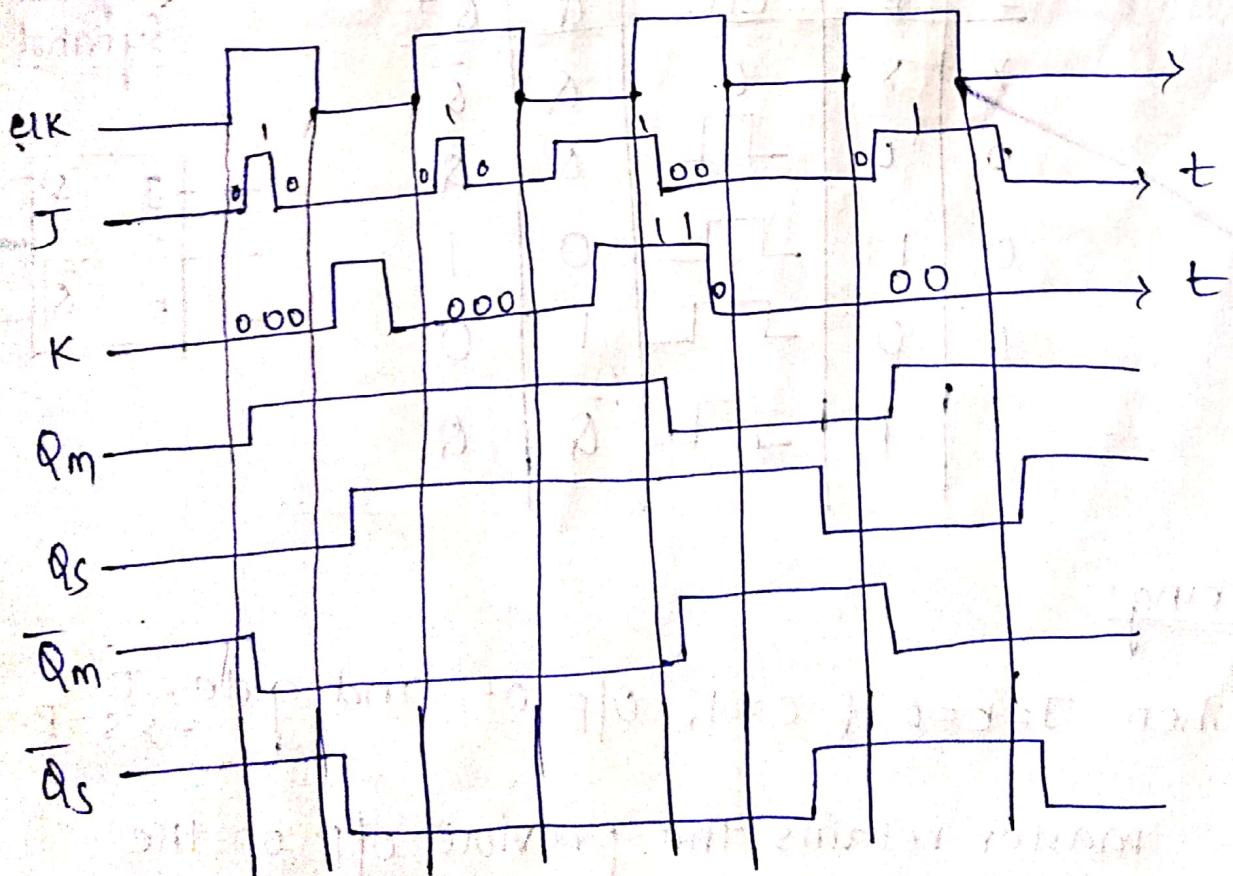
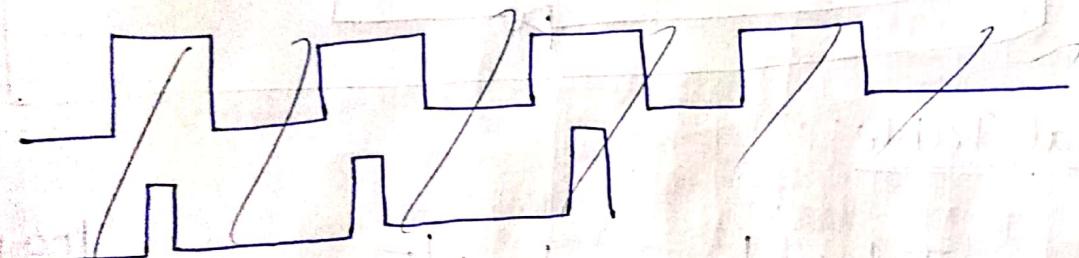
Working:

- 1.) when $J=K=0$ if $C \geq 1$, O/P of and gate $= 0$ $\Rightarrow S=R=0$
 \therefore master retains the previous O/P on the falling edge of clock, slave follows the master O/P.
- 2.) When $J=0$ and $K=1$, O/P of lower and gate is 1. i.e., $S=0$ & $R=1 \Rightarrow Q_M=0$
 $\bar{Q}_M=1$

3> when $J=1$ and $K=0$, o/p of upper and gate is 1 ie., $S=1$ and $R=0 \Rightarrow Q_m=1$
 $\bar{Q}_m=0$

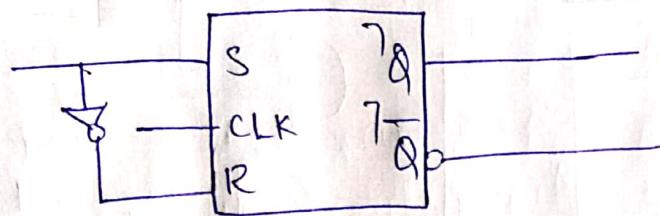
4> when $J=K=1$, assume $Q=0$ and $\bar{Q}=1$
o/p of upper and gate is 1 $\Rightarrow S=1$ & $R=0$

$$\therefore Q_m=1 \text{ & } \bar{Q}_m=0$$

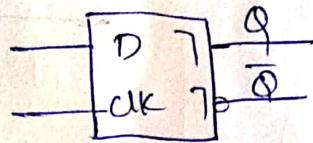


Master-slave D-flipFlop

logic diagram:-



logic symbol



Functional Table:

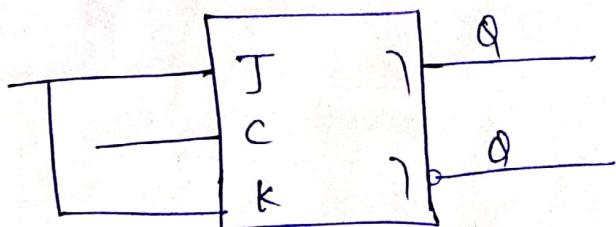
C	D	Q^+	\bar{Q}^+
0	x	Q	\bar{Q}
0	0	0	1
1	1	1	0

working:

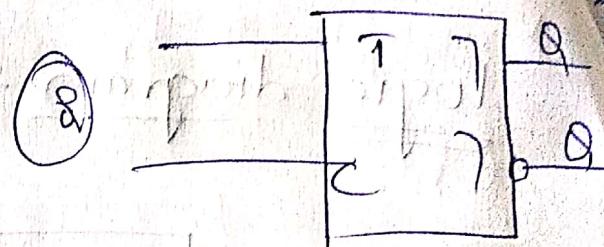
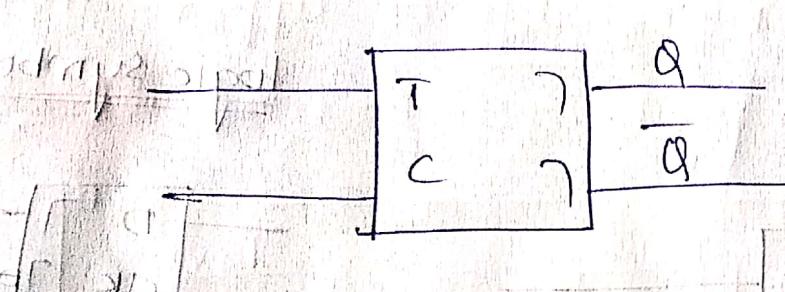
master registers the D i/p during the period when clock is high & then the state of the master is transferred to the slave at the falling edge of the clock.

Master slave T-flipflop

logic diagram:

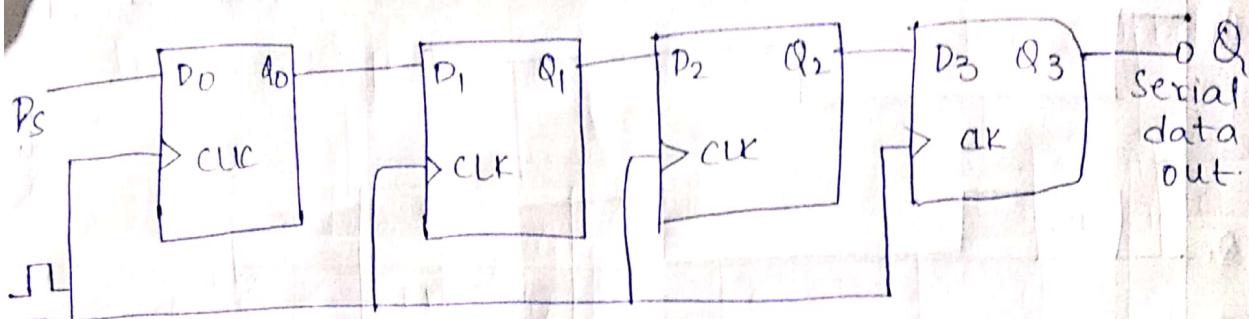


logic symbol



Sift Registers

(i) Serial-in Serial-out (SISO):



→ SISO is constructed using the Positive edge triggered D-Flipflops(FF)

→ 'Q' output of each FF is connected to 'D' input of the FlipFlop to its right

→ The control input of all FF's are connected together to a common synchronizing signal called clock.

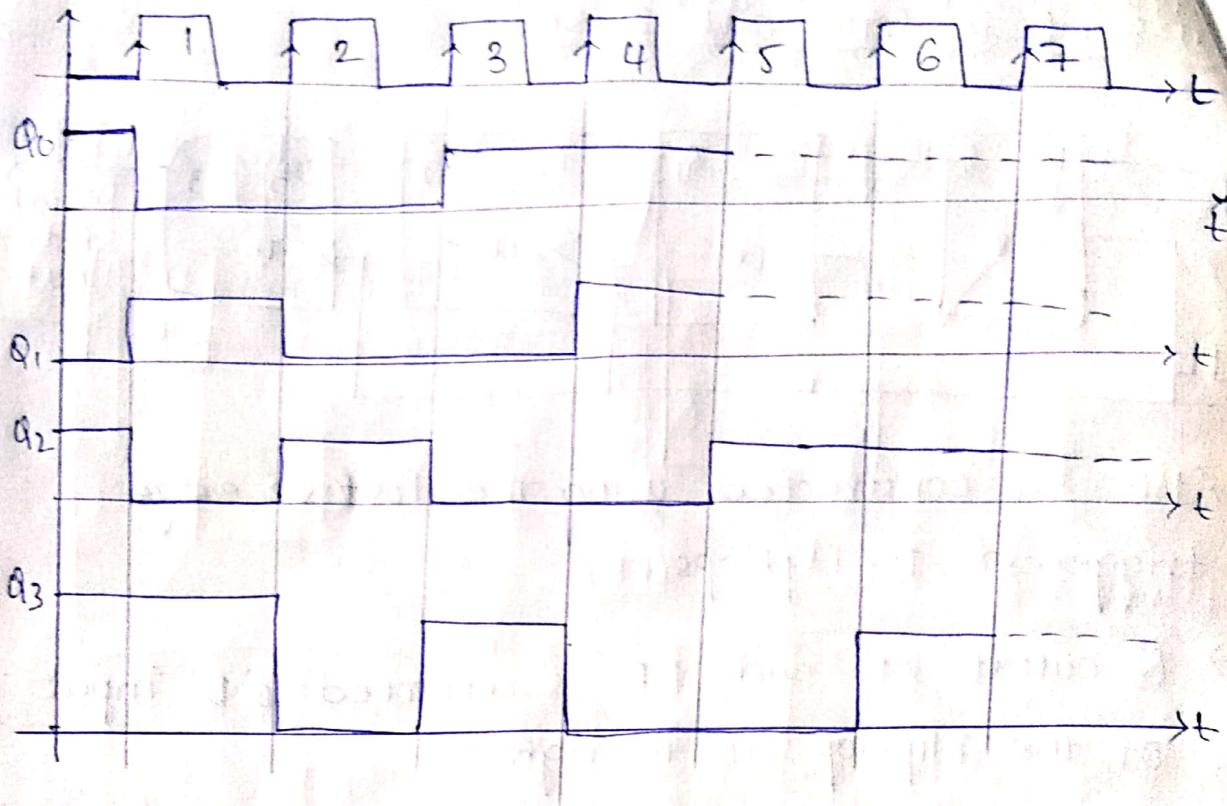
→ Upon occurrence of positive edge of the clock, the content of each FF is shifted one bit position to the right.

Ex: let the initial content be $Q_0 Q_1 Q_2 Q_3 = 1011$

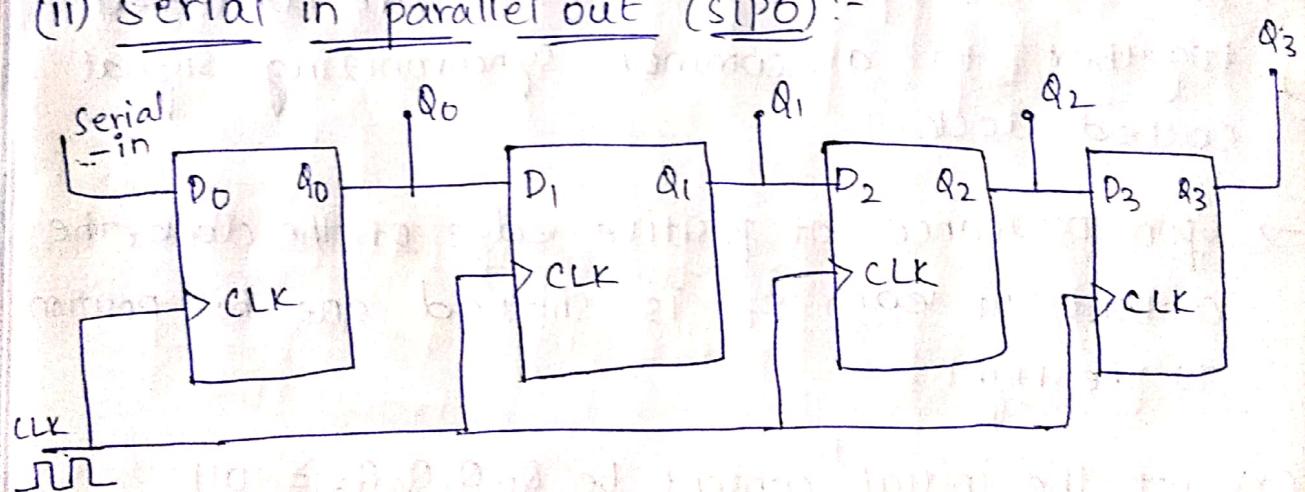
Now, let us shift the data 1100 serially to SISO.

clock pulses	CLK	DS	Q ₀ Q ₁ Q ₂ Q ₃
-	-	-	1 0 1 1
1	↑	0 (USB)	0 1 0 1
2	↑	0	0 0 1 0
3	↑	1	1 0 0 0
4	↑	1	1 1 0 0
5	↑	x	x 1 1 0
6	↑	x	x x 1 0
7	↑	x	x x x 1

Timing diagram



(ii) Serial in parallel out (SIPO) :-



- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.

when $\text{shift}/\overline{\text{load}} = 0 \Rightarrow$ loading mode to load parallel i/p
 $\text{shift}/\overline{\text{load}} = 1 \Rightarrow$ shifting mode to get serial o/p.

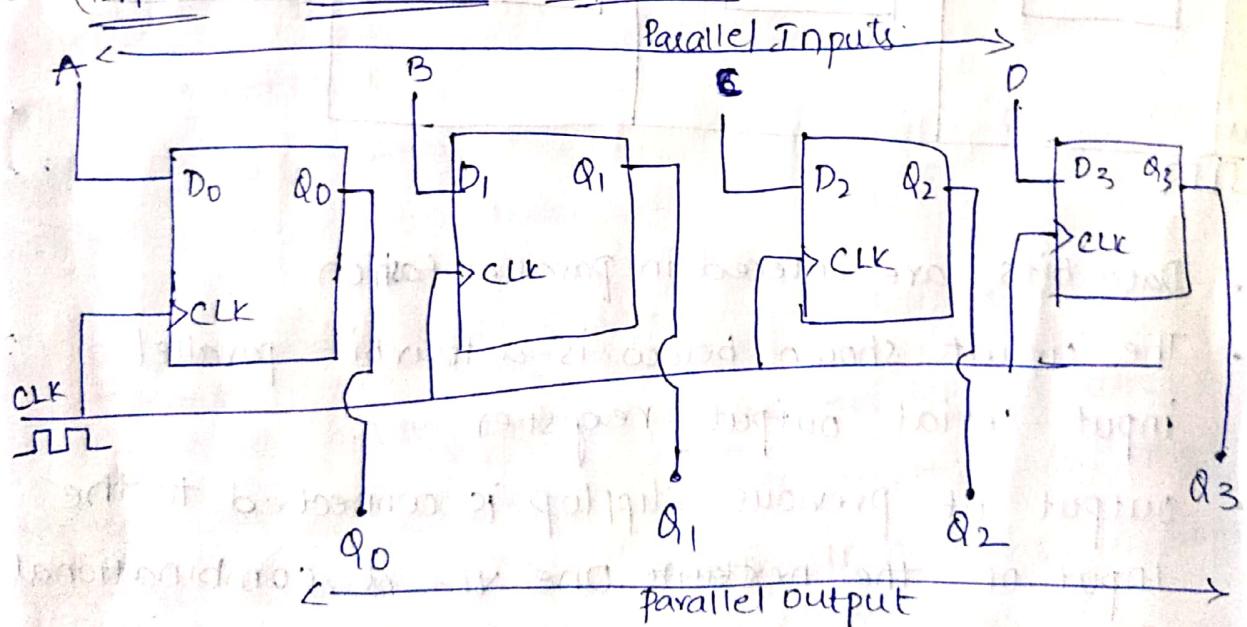
Ex:

CLK	A	B	C	D	Q ₀	Q ₁	Q ₂	Q ₃
↓	1	1	0	1	1	1	0	1
↓	x	x	x	x	x	1	1	0
↓	x	x	x	x	x	x	1	1
↓	x	x	x	x	x	x	x	1

$\text{shift}/\overline{\text{load}} = 0$

$\text{shift}/\overline{\text{load}} = 1$

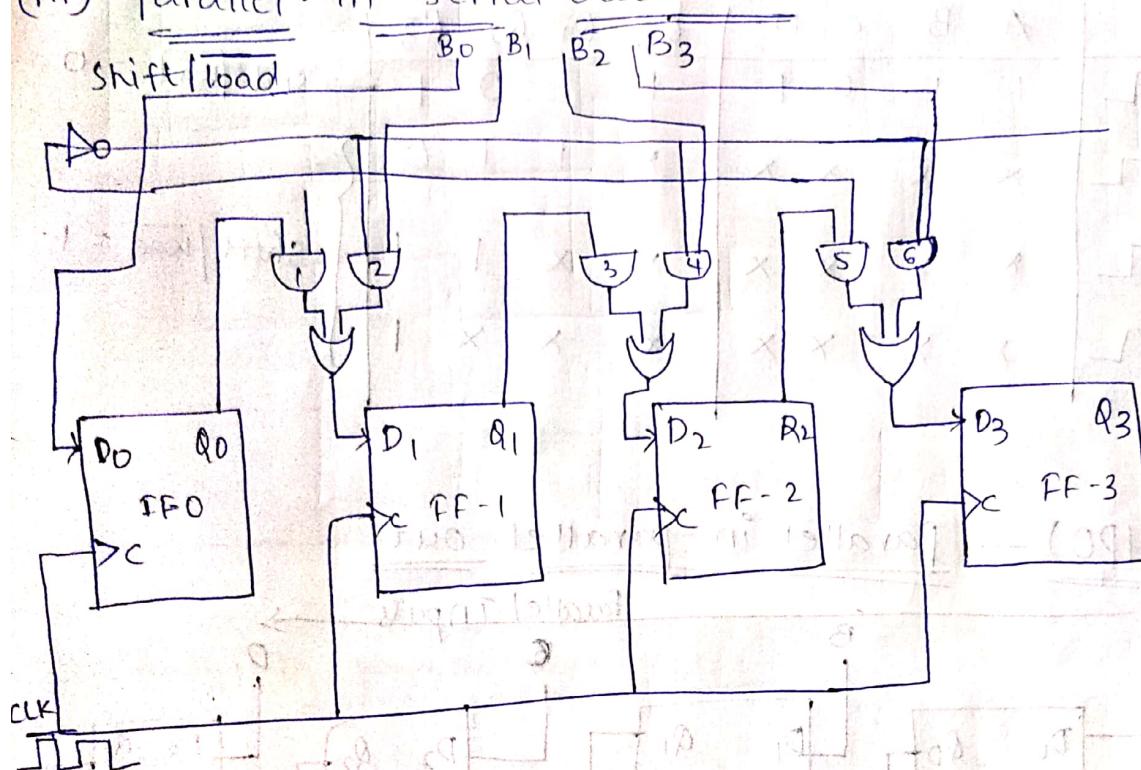
(iv) (PIPO) — Parallel in parallel out:



In this mode, the 4 bit binary input A, B, C, D is applied to the data inputs D₀, D₁, D₂, D₃ respectively. As soon as a negative edge of the four flipflops is applied, the input binary bits will be loaded into the flipflops simultaneously. The loaded bits will appear simultaneously to the output side. only clock pulse is essential to load all the bits.

• 4 clock cycles are required to load a four word, hence the speed of operation of SIPO mode is same as that of SISO mode.

(iii) Parallel-in serial out (PISO):

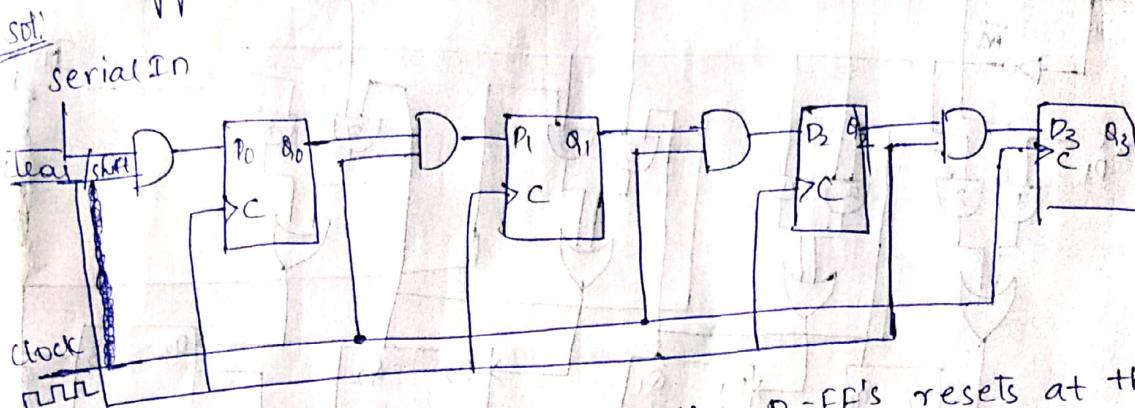


- Data bits are entered in parallel fashion.
- The circuit shown below is a fourbit parallel input serial output register.
- Output of previous flipflop is connected to the input of the next ~~one~~ one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied through the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode (or) load mode.

Bi-directional Register

Assignment:

- ④ Incorporate a synchronous clear facility in the 4-bit SISO register using the edge triggered D-FlipFlop.



when $\text{clear}/\text{shift} = 0$, then all the D-FF's resets at the next positive edge of the clock.

when $\text{clear}/\text{shift} = 1$, then shift line gets connected & shifting operation takes place.

Bi directional Shift Register:

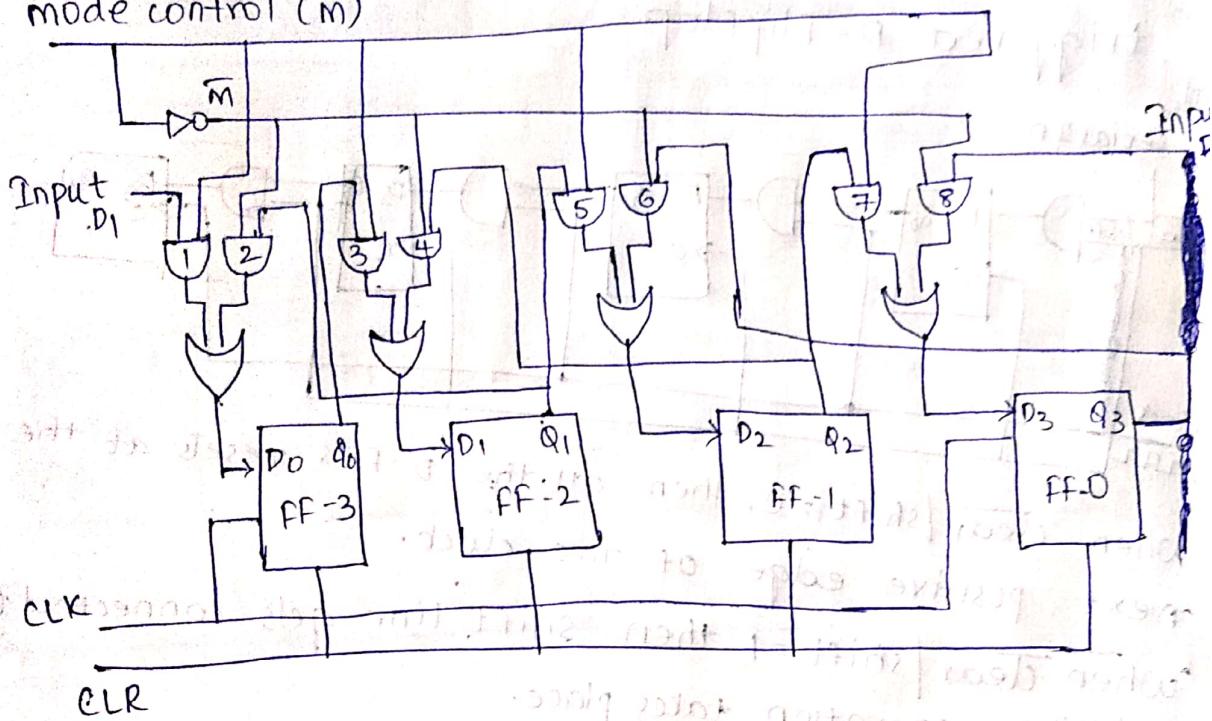
- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.

- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data bin either left or right direction.

- Such a register is called bi-directional register. A four-bit bi-directional shift register is shown in figure.

• There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL, along with a mode select input (m).

mode control (\bar{m})



S.NO	condition	operation
1.	with $m=1$ - shift right operation	If $m=1$, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6, 8 will be disabled. The data at DR is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with $M=1$ we get the serial right shift operation.
2.	with $m=0$ - shift left operation.	When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5, 7 are disabled. The data at DL is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with $M=0$ we get the serial right shift operation.

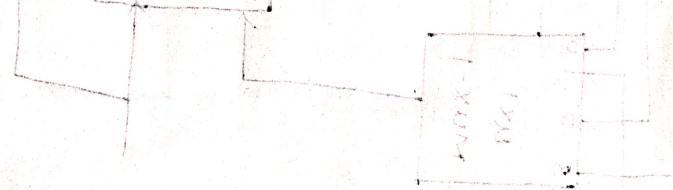
Universal Shift Register:

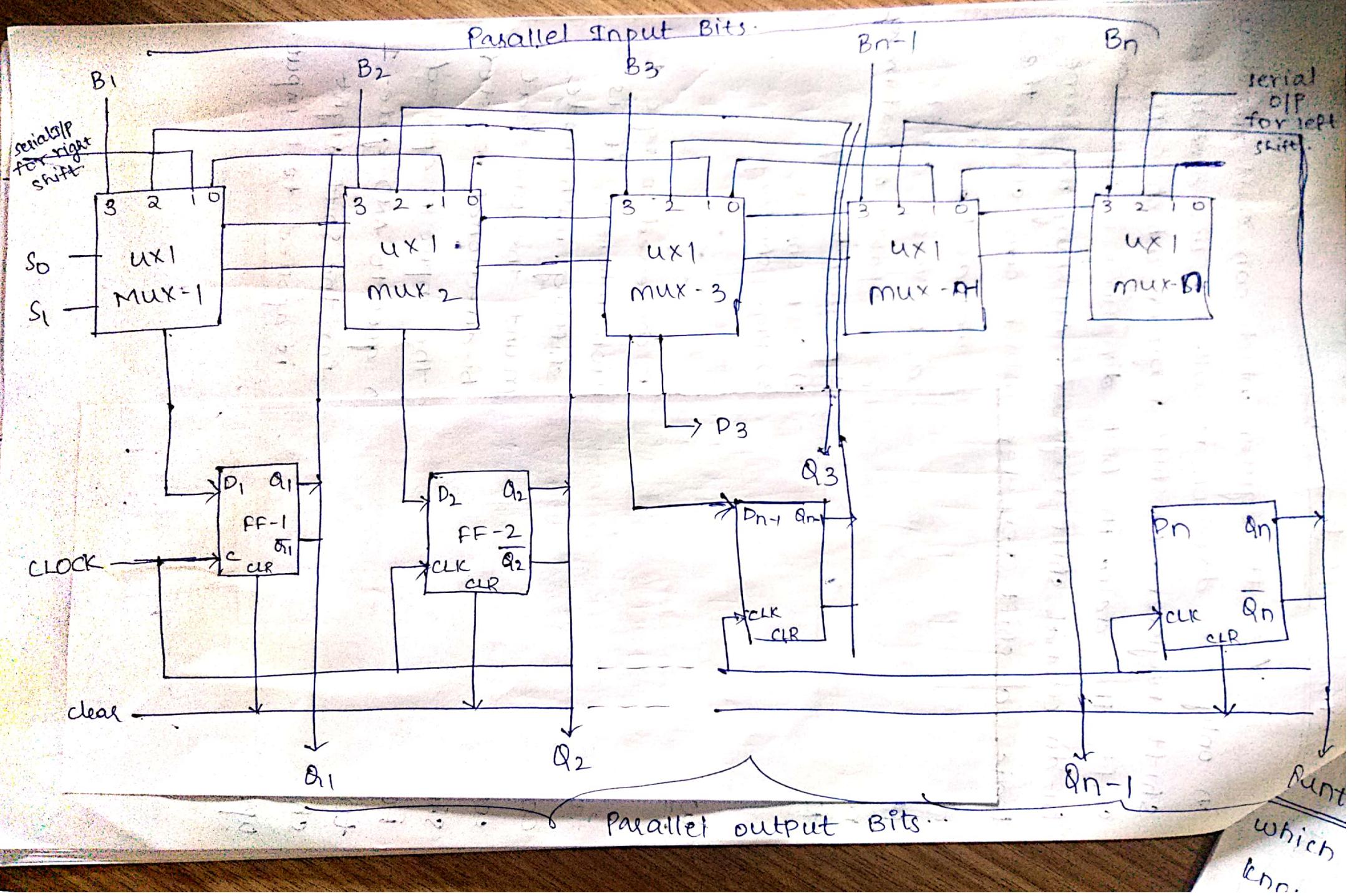
A shift register which can shift the data in only one direction is called a uni-directional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallelly, is known as a universal shift register. The shift register is capable of performing the following operations—

- 1) Parallel loading
- 2) Left shifting
- 3) Right shifting

Working:

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas, for the shift right operation, the serial input is applied to D input.





Counters: counter is a sequential circuit which is used for counting pulses is known as counter. It is a widest application of flipflops; it is a group of flipflops with a clock signal applied.

(or)

Counters:

- It is an example of Register
- flipflops are considered for output on application of a clock
- It produces a specific - output pattern, hence it is called as pattern generator.
- The combinations of 0 & 1 that are stored in counters is called as pattern.
ex: 1101, 1000 etc.
- The output pattern is called as state of the counter.
- The total number of states is called as modulus.

Binary Ripple Counter: (Asynchronous counter):

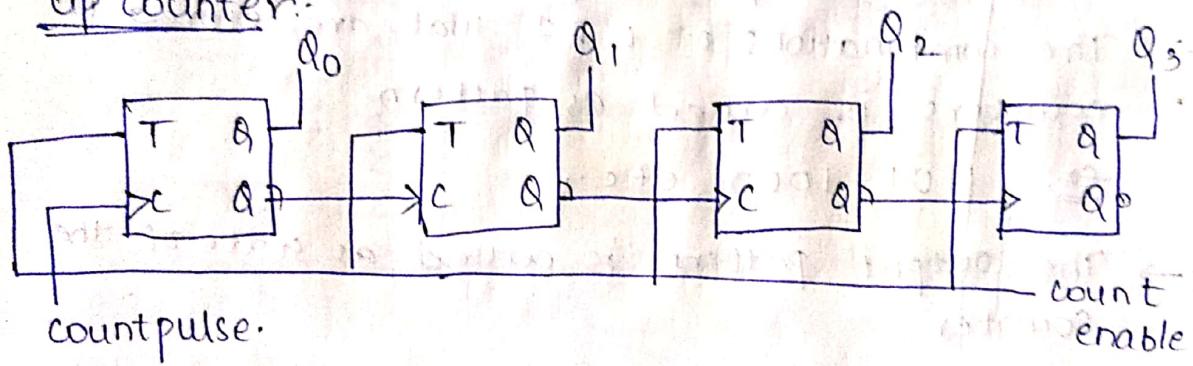
It is one of whose counting sequence / pattern corresponds to that of a binary number. modulus of binary counter is 2^n , where n is the number of flipflops.

ex) if $n=4$, $2^4 = 16$ modulus.

4-bit binary Ripple counter:

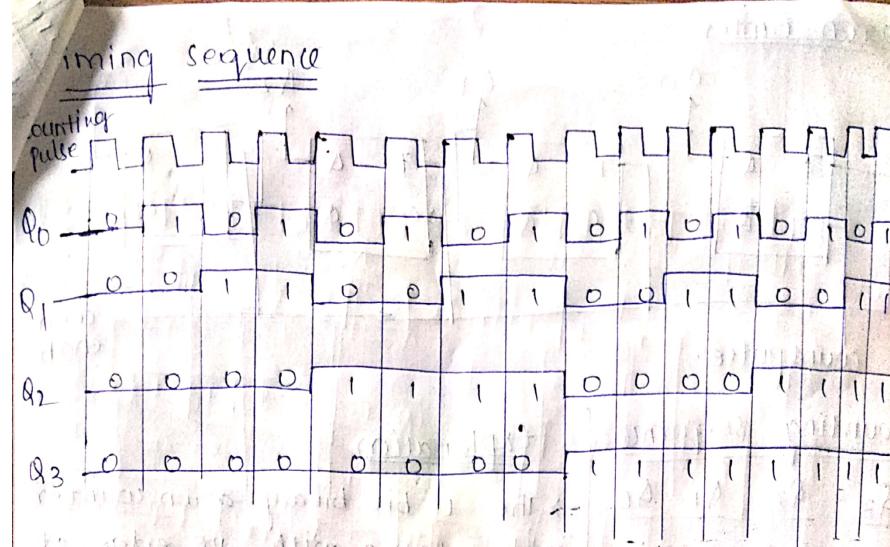
- It consists of 4 flipflops in cascaded form. [cascade- arranging in series form].
- The counter will count from 0000 to 1111 for upcounter and 1111 to 0000 for down counter.
- Ripple counter is a Asynchronous counter where Input of one flipflop is given as clock of the next flipflop.
- only the clock of first flipflop is given to clock bit.

UP counter:-



Counting Sequence:-

Q_3 / Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0 / 0	0	0	1	0	0	0
0 / 0	0	1	1	0	0	1
0 / 0	1	0	1	0	1	0
0 / 0	1	1	1	0	1	1
0 / 1	0	0	1	1	1	0
0 / 1	0	1	1	1	0	1
0 / 1	1	0	1	1	1	0
0 / 1	1	1	1	1	1	1

Explanation

→ The block diagram shows the 4 bit binary up counter implementation with positive edge triggered.

→ T=FF

→ It can also be constructed by connecting together J-K terminals of +ve edge triggered JK FF and labelled this common terminal as T.

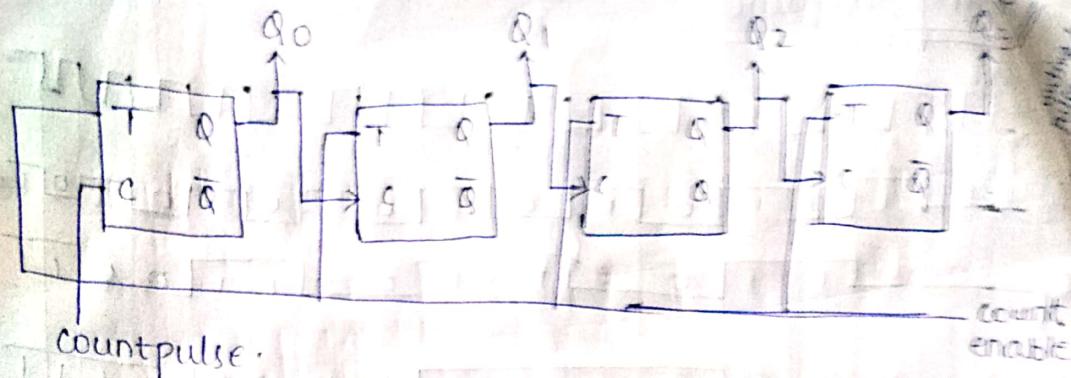
→ logic-1 is held at T, i.e., on application of clock o/p of flip flop toggles.

→ Count pulse is applied directly to first FF. Count pulse of remaining FF's are connected to \bar{Q} of previous one
∴ when $\bar{Q}=1$, +ve edge trigger occur at C of next FF.

Like $Q_3 Q_2 Q_1 Q_0 = 0000$, the next o/p will be 0001

It continues till - 1111 ($Q_3 Q_2 Q_1 Q_0$), and turns to 0000.

Down counter:



Counting sequence

<u>Q_3</u>	<u>Q_2</u>	<u>Q_1</u>	<u>Q_0</u>
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
0	0	1	0
0	0	0	1
1	0	0	0

Explanation

→ The 4 bit binary down counter implementation with -ve edge triggered T-FF's.

→ It can also be done by JK flip flop.

→ Count pulse is directly applied to 1st FF. Count pulse of remaining FF's are connected to Q of previous FF.

$$\text{let } Q_3 Q_2 Q_1 Q_0 = 1111$$

if $T=1$ for 1st +ve edge triggered of a clock Q_0 changes to 0.

Q_0 is clock to next FF. As it is 0, next Q will not change.

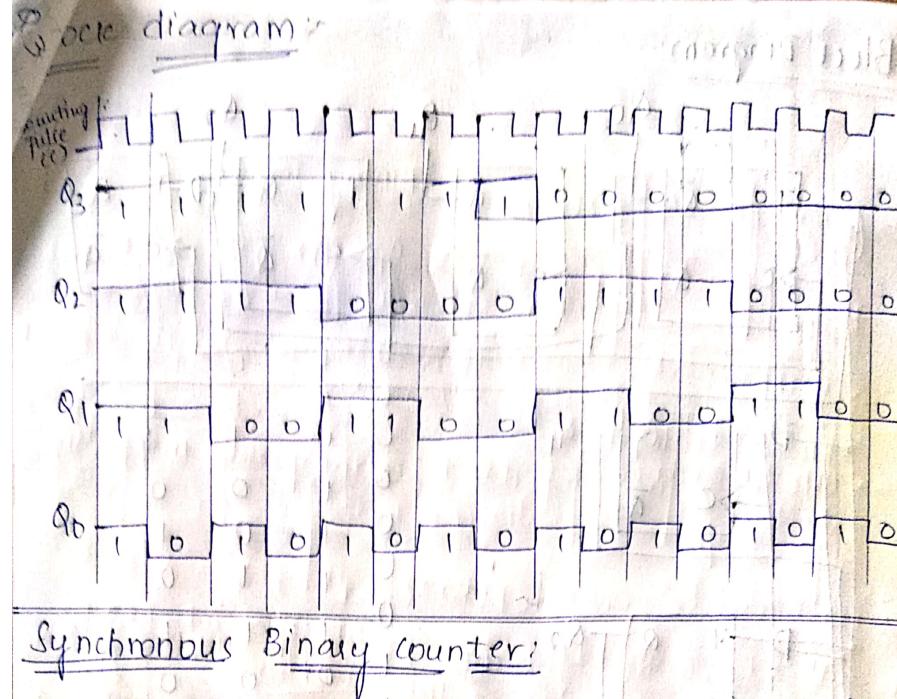
$$\therefore Q_3 Q_2 Q_1 Q_0 = 1110$$

on occurrence of next positive edge of clock Q_0 changes to 1. CLK is provided to 2nd FF.

∴ Q_1 changes to 0

$$\Rightarrow Q_3 Q_2 Q_1 Q_0 = 1101$$

like wise it decreases to 0000 & turns to 1111.



In Asynchronous counter we see the problem of setting of o/p. As FF's clock's are dependent on previous outputs, it takes more time to get final o/p.

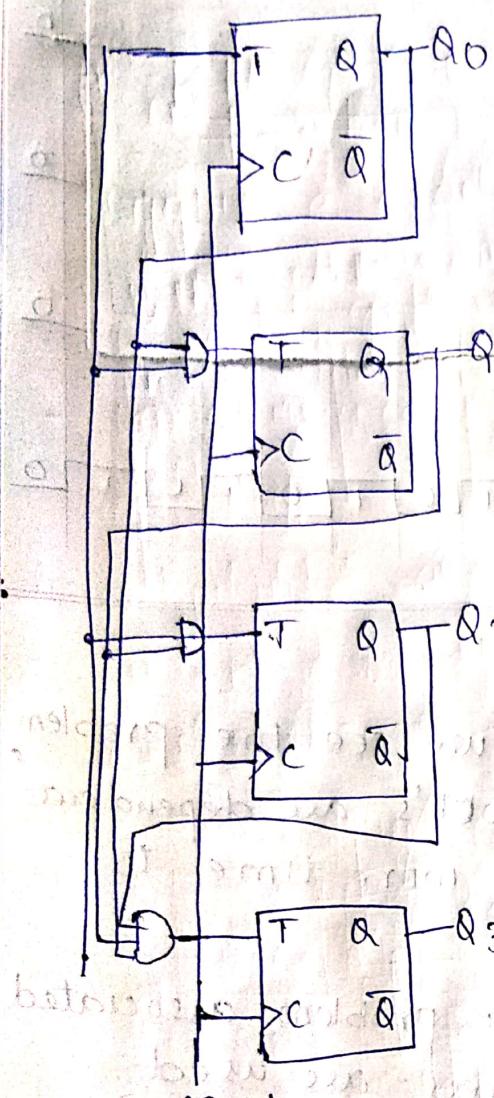
→ To avoid this setting time problem associated with these synchronous counters are used.

→ For synchronous counters clk pulses are directly applied to the control inputs C of all clk FF's.

→ All FF's change simultaneously with the edge triggered of clock.

→ The figure below shows binary up counter (4 bit) following the count sequence.

Block Diagram:-



Binary Sequence

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Working

→ It counts from 0000 to 1111.

→ T input is given high i/p.

→ There are 4 FF's. First FF gets i/p 1.

2nd FF gets input from a two input AND gates which gets input from T and Q_0 .

→ when $T = Q_0 = 1$, only that time o/p of 2nd FF will toggle.

→ 3rd FF gets input from a three input And gate which gets input from T, Q_0 , Q_1 .

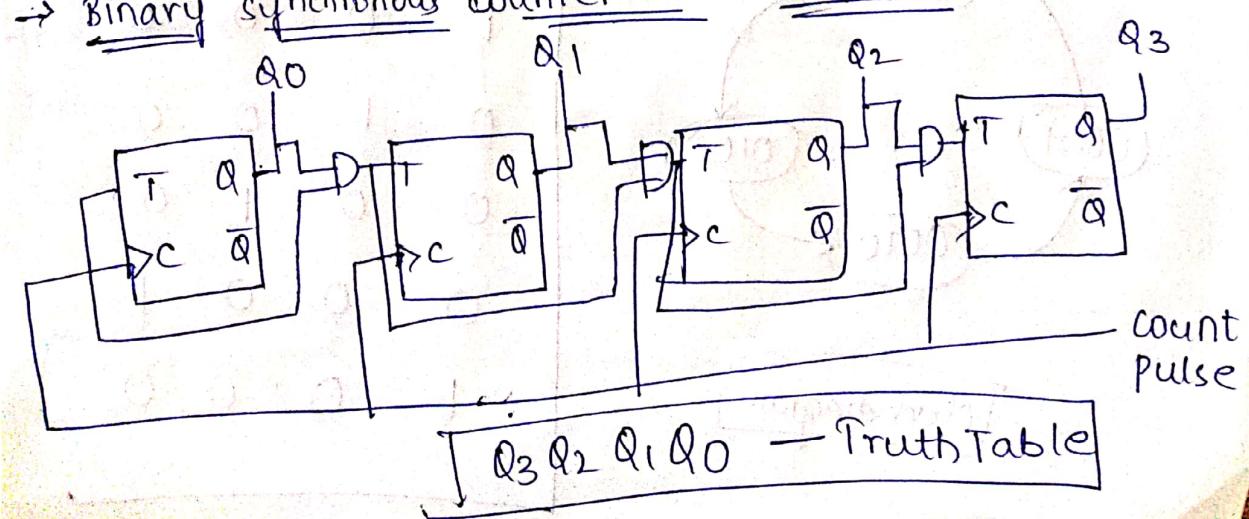
only when $T = Q_0 = Q_1 = 1$, The o/p Q_1 will toggle.

- 4th FF gets i/p from 3rd FF's AND gate, whose inputs are from T, Q_0, Q_1, Q_2 .
 $\therefore Q_3 = 1$ only when $T = Q_0 = Q_1 = Q_2 = 1$.
- All FF's are dependent on previous outputs and T i/p.

- For positive edge of 1st clock, $T = 1$.
Let $Q_3, Q_2, Q_1, Q_0 = 0000$
 $\therefore Q_0$ changes to '1'.
like this counter counts upto 1111 and return back to 0000.

* The disadvantage of this type counter is the number of inputs to AND gate will go on increasing for higher order sequences, which is practically difficult. So to avoid this we use two input AND gates synchronous counters.

- Binary synchronous counter with two i/p AND gates.

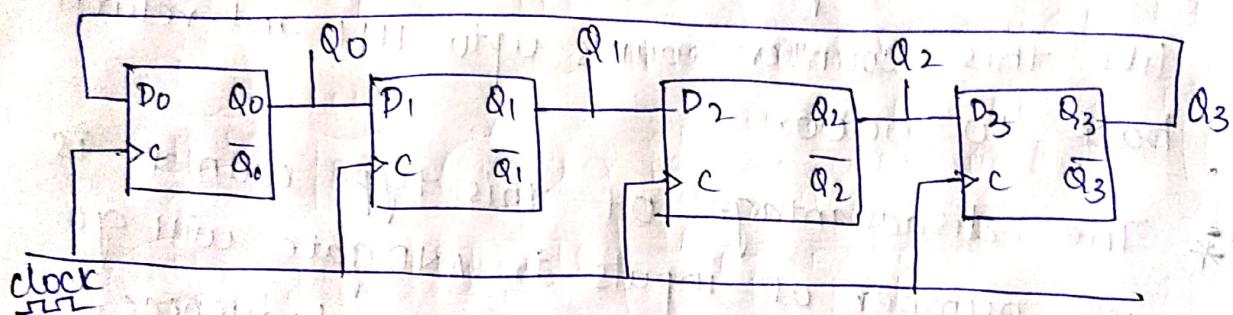


* RING COUNTER

It is a circular shift register, which is initialised such that only one of its FF is in 1 state and rest others are in 0 state.

→ On occurrence of each count pulse, the 1 is shifted to its adjacent FF.

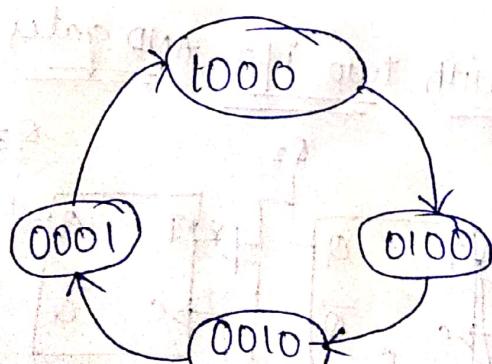
→ 4 D-FF's, are connected in series. O/P of 1 FF is connected to D i/p of next FF. ~~the~~ ~~initial~~



Let $Q_0 Q_1 Q_2 Q_3 = 1000$, on providing +ve edge of 1 clock, we get $Q_0 Q_1 Q_2 Q_3 = 0100$. like this the 1 is rotated in the form of ring. Hence called as ring counter.

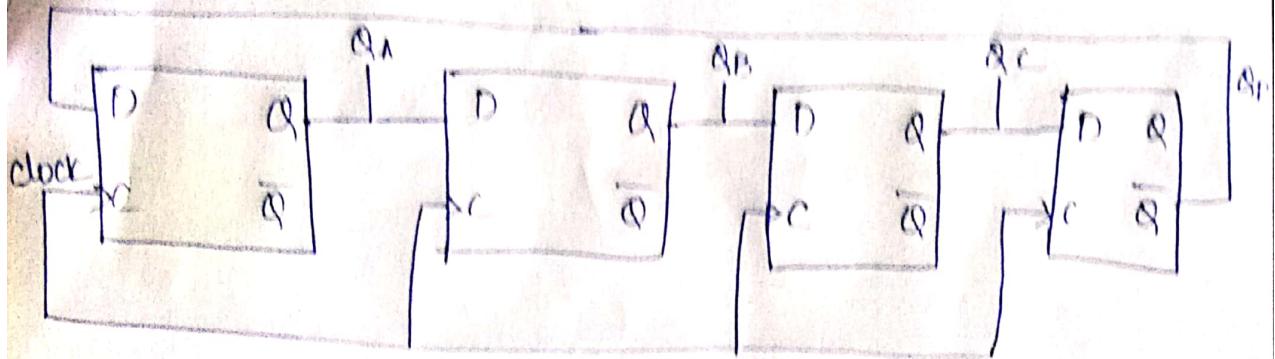
Truth Table

<u>Q_0</u>	<u>Q_1</u>	<u>Q_2</u>	<u>Q_3</u>
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0



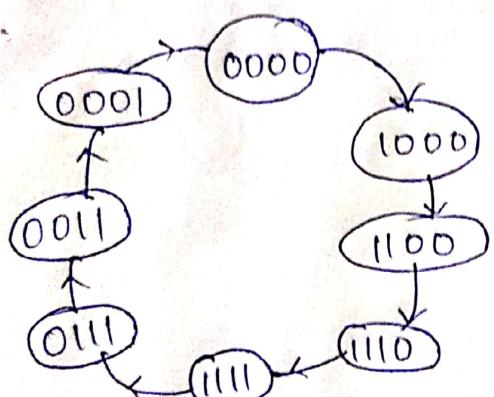
Clock diagram

Johnson Counter / switchtail / Twisted ring counter



→ Here Q_A is connected to front D i/p of 1st FF

→ The o/p of other FF are connected to i/p of next FF ie; Q_A, Q_B, Q_C are connected to D i/p of respective next i/P's



→ This structure always has 2^n states ($n = \text{no. of FF's used}$).

Truth table

Hence, modulus of the above Johnson counter is $2 \times 4 = 8$
∴ It has 8 states.

Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

clock diagram