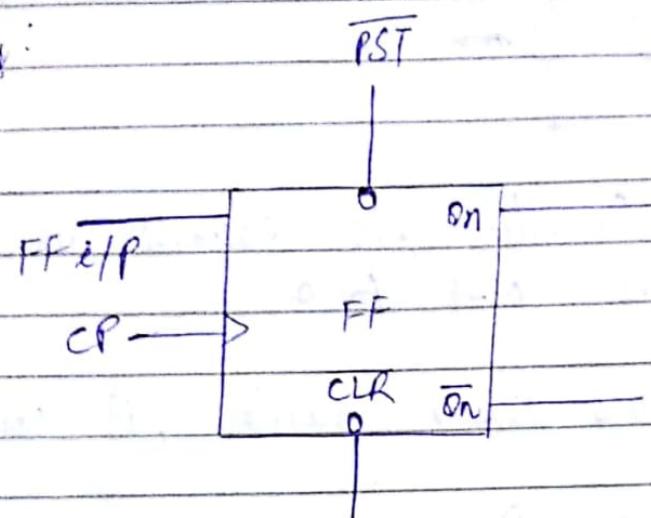


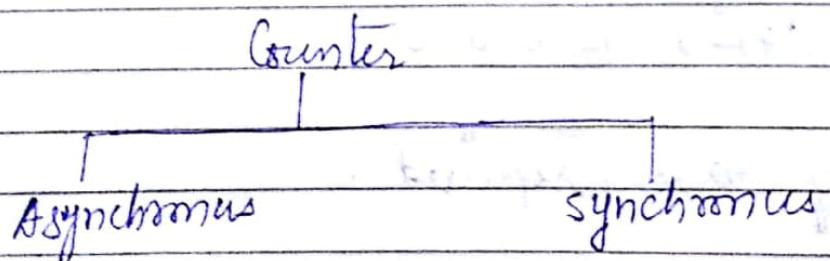
## Unit-4 Counter & Register

Counter:



PST	CLR	On <sub>n+1</sub>	
0	1	1	(set)
1	0	0	(reset)
1	1	FF	works normally
0	0	FF	doesn't work

Counter: It is used to count number of clock pulses. It is of 2 types -



Counter is also divided into 2 types - up Counter & Down Counter. In Up Counter, it counts in ascending order

from 0 to  $N-1$  for MOD N up counter.

Ex: In MOD + up Counter, it counts as follows  
 $(\overline{0}) \rightarrow (\overline{1}) \rightarrow (\overline{2}) \rightarrow (\overline{3}) \dots$

In Down Counter, it counts in descending order from  $N-1$  to 0.

Ex: for MOD 5 Down counter, it counts as follows

$(\overline{4}) \rightarrow (\overline{3}) \rightarrow (\overline{2}) \rightarrow (\overline{1}) \rightarrow (\overline{0})$

Total no. of flip-flops required to implement MOD N counter = total no. of bits required to represent  $(N-1)$

Q. Determine total no. of ff. required to implement MOD 18 counter.

$\begin{array}{r} 16 \\ 8 \\ 4 \\ 2 \\ 1 \\ \hline (N-1) \rightarrow (17) \rightarrow 1 \ 0 \ 0 \ 0 \ 1 \end{array}$

5 ff. are required.

Modulus of a Counter

Total no. of used states are called Modulus of a Counter.

$$2^n \geq N$$

where  $n$  is no. of flip flops required  
 $N$  is Modulus of a counter (used states)

Q. Det. min. no. of flip flop req. to implement MOD 8 counter

$$2^n \geq 8$$

$$2^n \geq 2^3$$

$$n = 3$$

$$2^n \geq N$$

$$n \log_2 2 \geq \log_2 N$$

$$\boxed{n \geq \log_2 N}$$

6/1/19

serial/ripple/frequency divider

Date \_\_\_\_\_  
Page \_\_\_\_\_

## Asynchronous MOD 8 Counter (up)

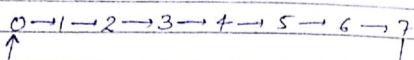
Binary Counter : If  $N = 2^n$ . Ex. 11002, MOD4, MOD8Non-Binary Counter: If  $N \neq 2^n$ . Ex. MOD3, MOD5, MOD12

$$\begin{aligned} 2^n &\geq N \\ 2^3 &\geq 8 \\ 8 &\geq 8 \end{aligned}$$

MOD N bin Counter  
( $N=2^n$ ) or n bit  
bin counters

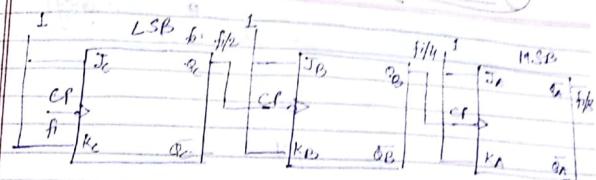
 $n=3$ 

Hence 3 flip flops are required

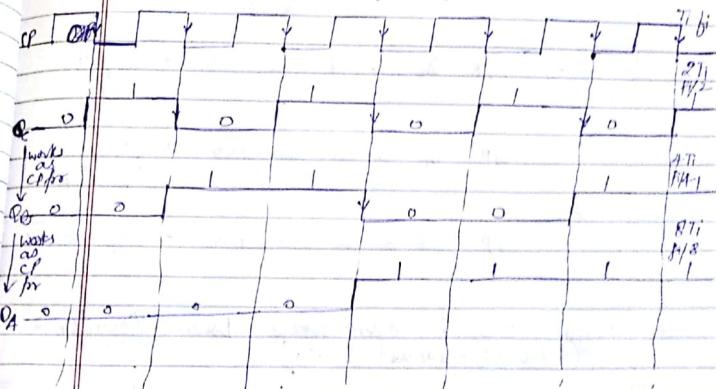


$Q_A$	$Q_B$	$Q_C$
0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
1	1	0
1	1	1

{ 1 → 0      0 → 1  
↓            ↓  
-ve edge triggering



Timing Diagram



In asynchronous counter, external CP is applied to first ff (LSB) & if it is present ff is applied to CP of next ff & so on. It is also called Ripple counter, serial counter or frequency divider.

In binary counter, O/P frequency is equal to (i/p frequency / 2)

$$f_o = f_i/2^n$$

$$f_o = f_i/N$$

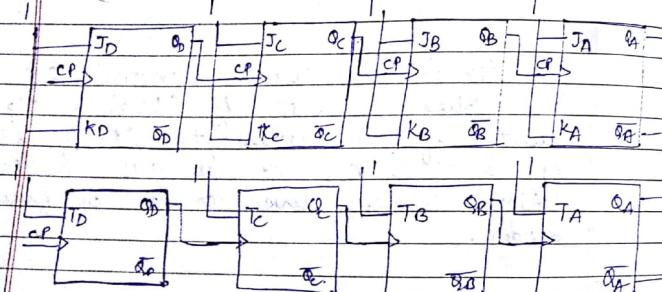
If we take O/P from  $Q_n$  & CP is -ve  $\rightarrow$  Up counter

& O/P  $Q_n$  & CP is -ve  $\rightarrow$  Down Counter

O/P  $Q_n$  & CP is +ve  $\rightarrow$  Down Counter

O/P  $Q_n$  & CP is +ve  $\rightarrow$  Up counter

Q. Design MOD 16 Asynchronous Down Counter.  
(4-bit Asynchronous Counter)



\* Design of Asynchronous Counter of MOD N if  $N \neq 2^n$  (Non-Binary Counter)

• Design MOD 6 ripple counter (Asyn.) -

$$2^n \geq N$$

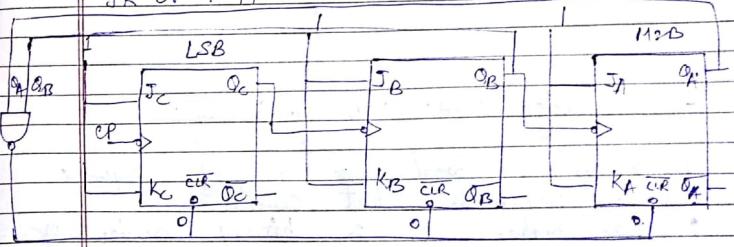
$$\begin{aligned} 2^n &\geq 6 \\ \Rightarrow 2^3 &\geq 6 \end{aligned}$$

$$n=3 \Rightarrow 3 \text{ ff are required}$$

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow \dots \rightarrow 6$$

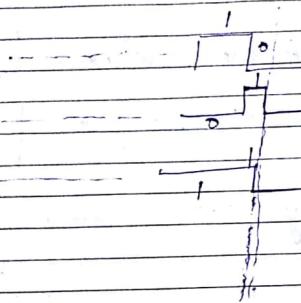
we'll get 6  
also for some  
due to propagation  
delay

• Design binary (3-bit) counter by using JK or T FF



$Q_A$	$Q_B$	$Q_C$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

11-1011



Date \_\_\_\_\_  
Page \_\_\_\_\_

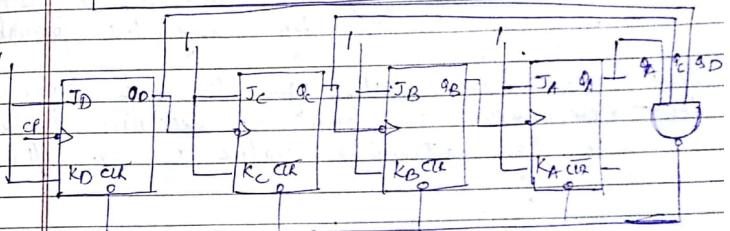
gate are connected to  $\bar{Q}_i$  of that FF's where  $i$  comes in binary representation of  $N$ .

Q. Design MOD 11 Asyn. Counter

$$\begin{aligned} 2^n &\geq 11 \\ 2^4 &\geq 11 \end{aligned}$$

: 4 FF are required

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11$



→ To design MOD N Non-bin. Counter :

1. Determine total no. of FF required to implement that counter
2. Now make ckt of binary counter of ref. FF bits
3. Now determine binary equivalent of  $N$  by indicating  $Q_1$ ,  $Q_2$ ,  $Q_3$ , ...
4.  $\bar{Q}_i$  of NAND gate is connected to CLR of all the FF & if of NAND

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

## Synchronous Counter (Parallel Counter)

In synchronous counter one common CF is applied simultaneously to all the flip-flops. That's why it is called parallel counter.

- Procedure to design MOD N synchronous counter or given sequence

Step 1- Determine no. of FF required to design synchronous counter by using the formula  $2^n \geq N$ .

Step 2- Now draw state diagram & state table

Step 3- Now with the help of excitation table of FF, determine value of FF if its

Q <sub>n-1</sub>	Q <sub>n</sub>	T	D	J K	S R
0	0	0	0	0 d	0 d
0	1	1	1	1 d	1 0
1	0	1	0	d 1	0 1
1	1	0	1	d 0	d 0

Step 4- Now with the help of K-Map, determine value of FF if its which are function of FF's present state.

Ans

Step 5- Now with the help of given FF & logic gates design synchronous counter of given sequence

- Q. Design MOD 8 sync. counter or design a syn. counter which counts the following sequence by using T FF or 3-bit bin. sync. counter

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$$

$$2^3 \geq 8$$

$n=3 \therefore 3$  FFs are required.

P.S		N.S			F.F. if ps			
Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>A'</sub>	Q <sub>B'</sub>	Q <sub>C'</sub>	T <sub>A</sub>	T <sub>B</sub>	T <sub>C</sub>
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

$T_A$ :	$Q_B Q_C$
$S_P$	$00\ 01\ 11\ 10$

$$T_A = Q_B Q_C$$

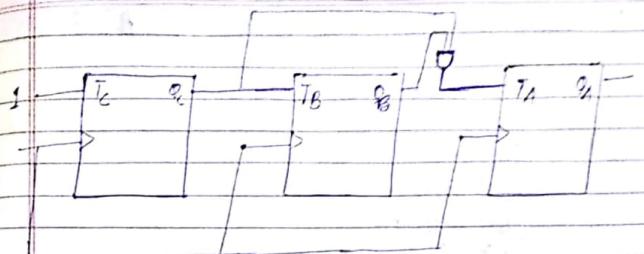
$T_B$ :	$Q_B Q_C$
$S_P$	$00\ 01\ 11\ 10$

$$T_B = Q_C$$

$T_C$ :	$Q_B Q_C$
$S_P$	$00\ 01\ 11\ 10$

$$T_C = 1$$

state  
reg



$Q_A$  toggles after each CP  $\Rightarrow Q_A = 1$

$Q_B$

$Q_F$	$Q_B$	$Q_C$
0	0	0
0	0	0
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0

$$\text{TP772} \rightarrow T_B = Q_C$$

$$\text{TP772} \rightarrow T_A = Q_B Q_C$$

Q. Design MOD 16 syn. Counter by D flip-flop

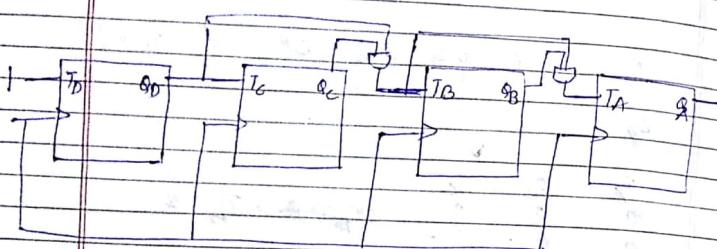
1-bit syn. counter :- ,

$$T_D = 1$$

$$T_C = Q_D$$

$$T_B = Q_C Q_D$$

$$T_A = Q_B Q_D$$



### Time Delay

In Asynchronous Counter -

$$T_{clock} = n t_{pdff}$$

n is total no of FFs &  $t_{pdff}$  is prop. delay of each FF

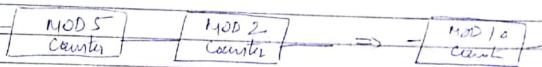
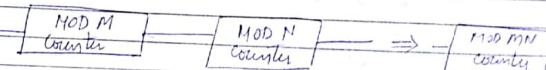
$$f = \frac{1}{T_{clock}}$$

In synchronous Counter -

$$T_{clock} = t_{pdff} + (n-1) t_{pdff AND}$$

$$f = \frac{1}{T_{clock}}$$

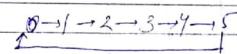
for both Asyn & syn -



for non-binary counter -

$$\text{Off freq.} = \frac{\text{off freq.}}{\text{no. of used states}}$$

Design MOD 6 Synchronous counter by using D flip-flops



$$2^3 > 6$$

∴ 3 FF are required

	$Q_A$	$Q_B$	$Q_C$	$+ + +$	$D_A$	$D_B$	$D_C$
0	0	0	0	0 0 0	0 0	0 0	0 0
1	0	0	1	0 1 0	0 1	0 1	0 0
2	0	1	0	0 1 1	0 1	0 1	1 1
3	0	1	1	1 0 0	1 0	1 0	0 0
4	1	0	0	1 0 1	1 0	1 0	1 1
5	1	0	1	0 0 0	0 0	0 0	0 0
6	1	1	0	d d d	d d d	d d d	d d d
7	1	1	1	d d d	d d d	d d d	d d d

$$D_A = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 1 & 1 & 0 \\ \hline 1 & D & X & X & C \\ \hline \end{array}$$

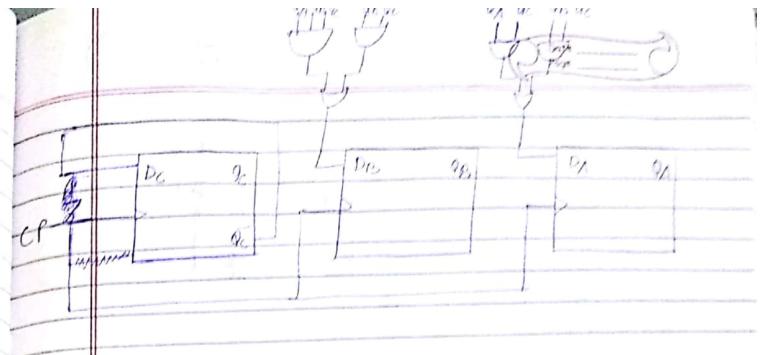
$$D_A = Q_A \bar{Q}_C + Q_B \bar{Q}_C$$

$$D_B = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & X & X \\ \hline \end{array}$$

$$D_B = Q_A \bar{Q}_C + Q_B \bar{Q}_C$$

$$D_C = \begin{array}{|c|c|c|c|} \hline & 00 & 01 & 11 & 10 \\ \hline 0 & 1 & 0 & 1 & 3 \\ \hline 1 & 1 & 0 & X & X \\ \hline \end{array}$$

$$D_C = \bar{Q}_C$$



Q. Design a sync counter which counts the following sequence by using JK FF

0 → 1 → 3 → 2 → 0

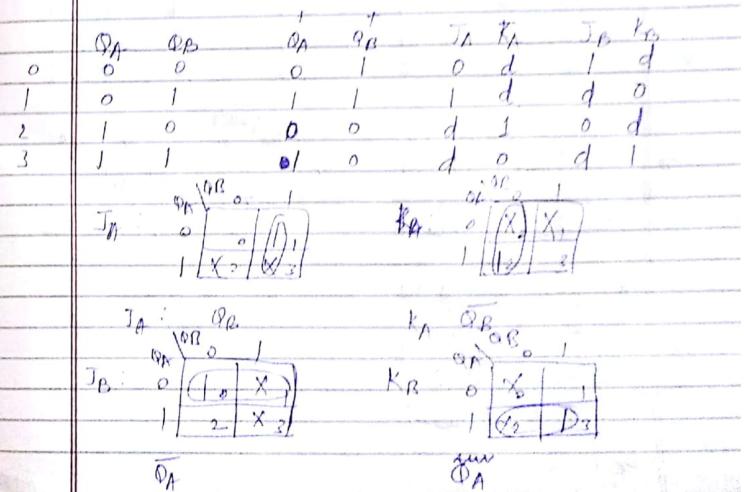
00 → 00

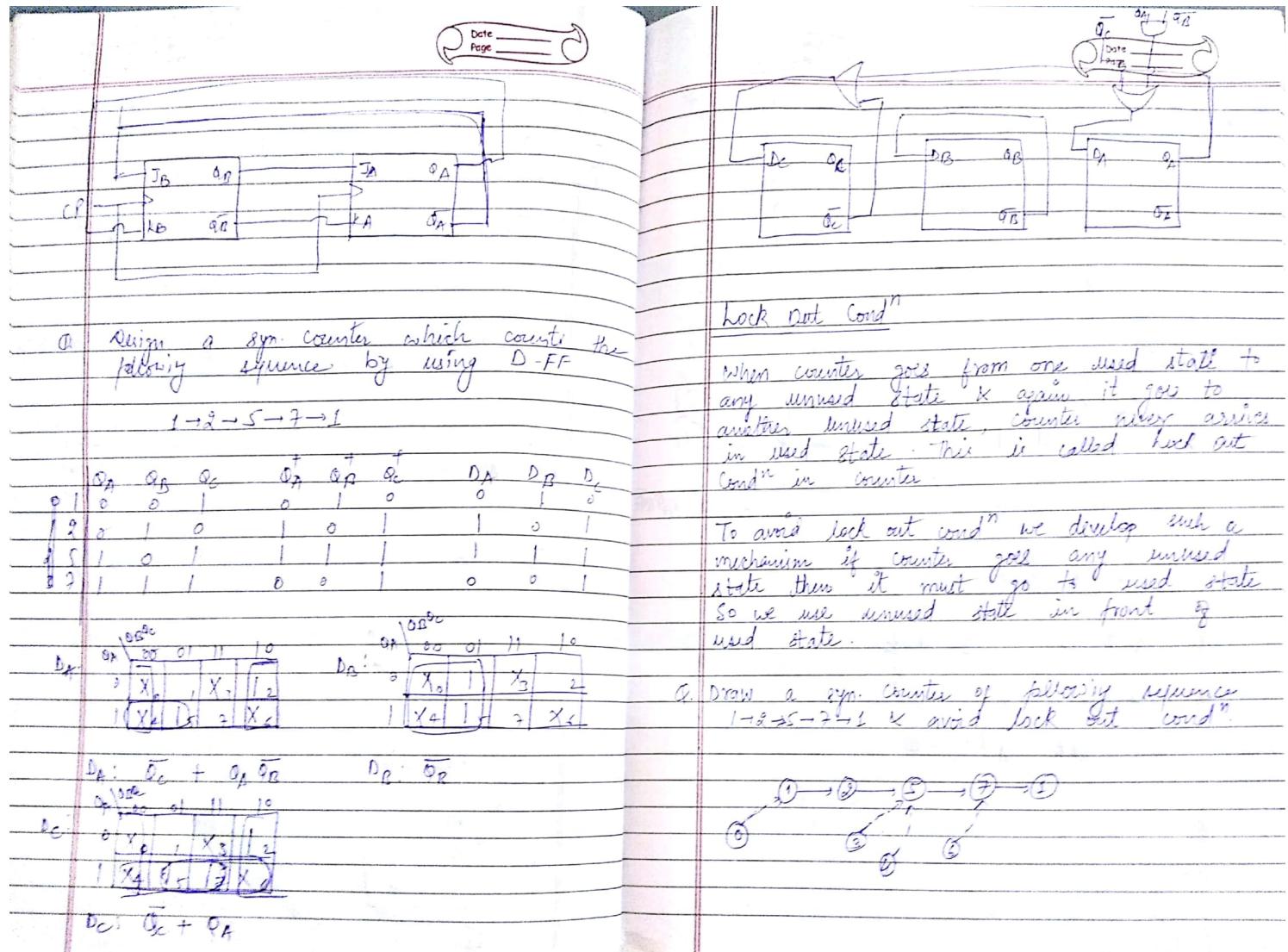
01 → 01

10 → 10

11 → 11

Q. MOD 4 Gray sync counter





Date \_\_\_\_\_  
Page \_\_\_\_\_

	$Q_A$	$Q_B$	$Q_C$	$Q_A$	$Q_B$	$Q_C$	$D_A$	$D_B$	$D_C$
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0
2	0	1	0	1	0	1	1	0	1
3	0	1	1	1	0	1	1	0	1
4	1	0	0	1	0	1	1	0	1
5	1	0	1	1	1	1	1	1	1
6	1	1	0	1	1	1	1	1	1
7	1	1	1	0	0	1	0	0	1

$Q_A \oplus Q_B$

$Q_A$	00	01	11	10
0	00	01	11	10
1	11	00	01	10

$Q_A \oplus Q_B$

$Q_A$	00	01	11	10
0	00	01	11	10
1	11	00	01	10

$D_B = Q_B \oplus C$

$D_B = Q_B \oplus C + Q_A \bar{Q}_B \bar{Q}_C$

$Q_B \oplus C$

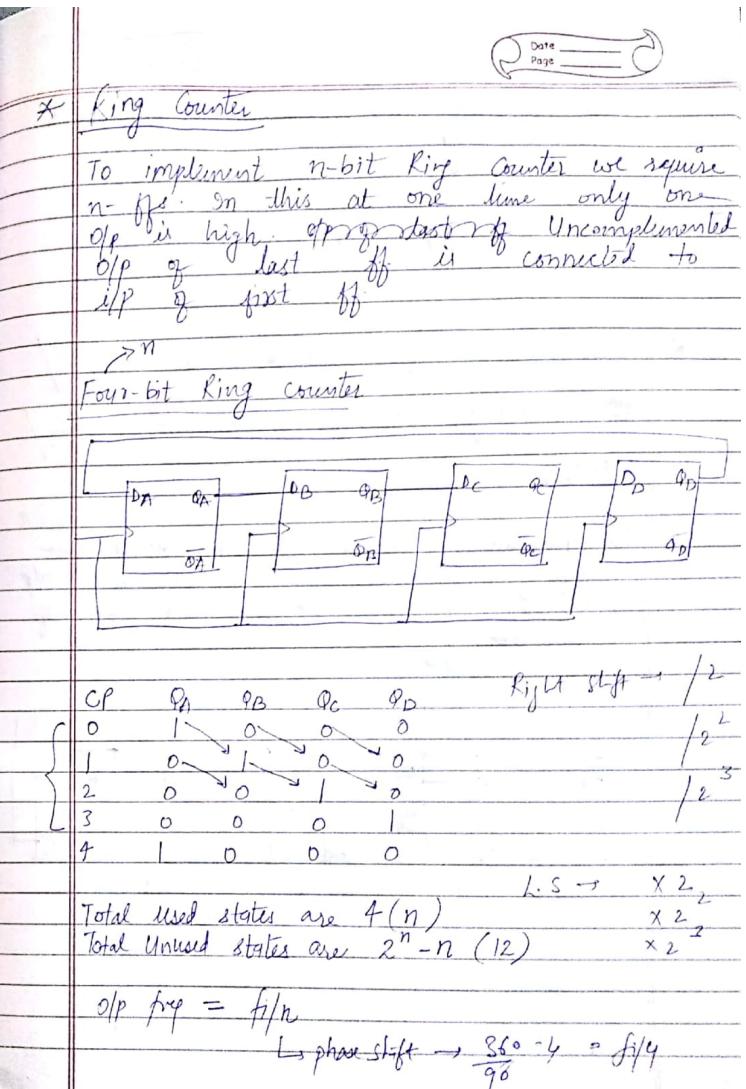
$Q_A$	00	01	11	10
0	00	01	11	10
1	11	00	01	10

$D_B = Q_B \oplus C + Q_A \bar{Q}_B \bar{Q}_C$

$D_A = Q_A + Q_B ; D_B = Q_B \oplus C$

$D_C = Q_A \oplus Q_B (Q_B + \bar{Q}_C)$



Q) Det. total no. of unused states in 8-bit Ring Counter

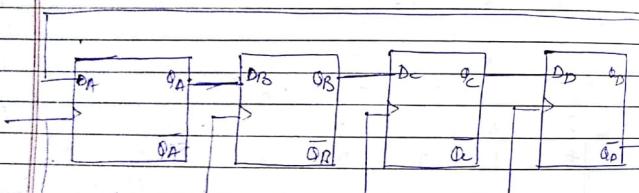
$$A \rightarrow 2^8 - 8 = 240$$

$$dp fm = f/8$$

\* Johnson Counter / Twisted Ring Counter / Creeping Counter / Switch Tail Counter

Mobile Counter

To implement n-bit Johnson counter, we require n-FF. In this complemented output of last FF is connected to input of first FF.



CP	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>	Q <sub>D'</sub>
0	0	0	0	0	1
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	1
4	1	1	1	1	0
5	0	1	1	1	0
6	0	0	1	1	0
7	0	0	0	1	0
8	0	0	0	0	1

### Register

Register is group of FFs which is used to store temporary data bits.

To implement n-bit register, we require n-FFs & it will store n-bits.

Types of registers :

a) On the basis of Data in & Data out  
[Serially & parallelly]

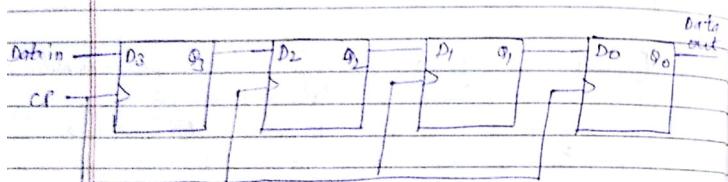
Serial in serial out (SISO)  
SIPO (serial in parallel out)  
PIPO  
PISO

b) On the basis of data shifting from left to right or right to left

Right shift  
e.g.  
(Data will shift from L to R)

Left shift reg.  
(Data will shift from R to L)

\* 4-bit Serial In serial Out Right Shift Register

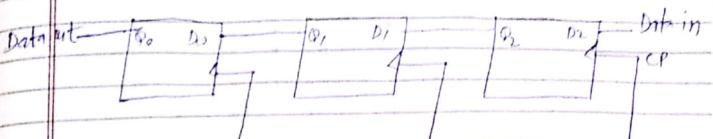


Data  $D_3 \ D_2 \ D_1 \ D_0$   
Pulse 1 0 1 1

CP	$Q_2$	$Q_1$	$Q_0$	in	out
0	0	0	0		
1	1	0	0		
2	1	1	0		
3	0	1	1		
4	1	0	1		

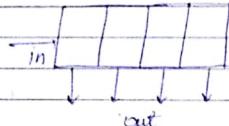
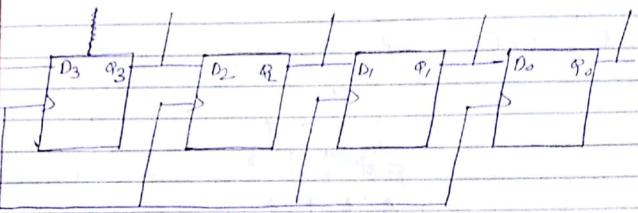
$$T_{clock} = n \cdot t_{\text{delay}}$$

3-bit SISO Left shift Register

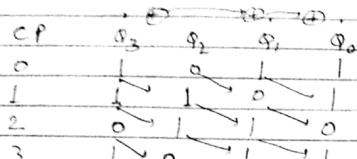


\* 4-bit SIPO shift Register

parallel o/p

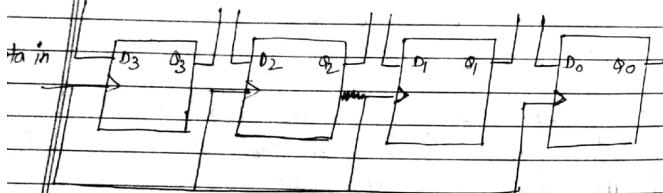
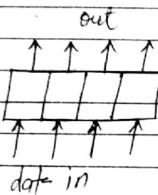


Q. In SIPO shift register What will be value of  $Q_3, Q_2, Q_1, Q_0$  after 3 CP in the following 2gm.



$D_0 \rightarrow 1011$

### \* PIFO Shift Register



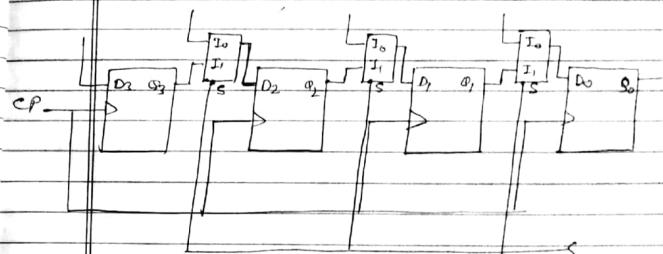
In this input is taken parallelly (simultaneously) & output is also taken parallelly

$$T_{clock} = t_{trig}$$

### \* FIFO PISO shift Register

In this input is taken serially (given) & output is taken parallelly but output is taken parallelly (1-bit at a time)

### \* 4-bit PISO shift register



$S=0 \Rightarrow I_0$  will work (parallel in)

$S=1 \Rightarrow I_1$  will work (serial out)

### Bi-directional Shift Register

It will shift data either in left to right direction or right to left direction.

### Universal Shift Register

Universal shift Registers are those registers which perform the following operations.

- a) SISO
- b) SIPO
- c) PISO
- d) PIPO
- e) Bi-dir. Shift register

### \* Applications of Shift Registers -

1. To store temporary data in Microprocessor
2. Left Shift
3. Right Shift
4. Multiplication & Division
5. To convert serial data into parallel data. (SIPO)
6. To convert parallel data into serial data (PISO)
7. Ring Counter & Johnson Counter