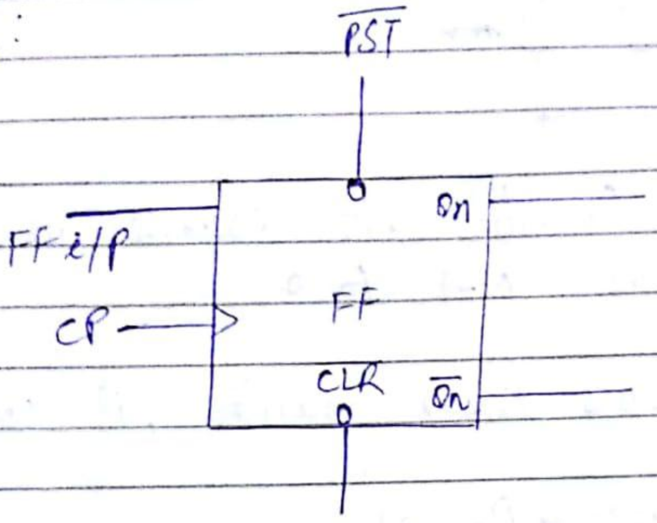
07/10/17

Date
Page

Unit-4 Counter & Register

Counter:



$\overline{PST}$	$\overline{CLR}$	Q_{n+1}	
0	1	1	(set)
1	0	0	(Reset)
1	1	F.F. works normally	
0	0	F.F. doesn't work	

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

$$\begin{tikzcd} & \text{Counter} \arrow[dl, dash] \arrow[dr, dash] & \\ \text{Asynchronous} & & \text{synchronous} \end{tikzcd}$$

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 4 up Counter, it counts as follows

$$\begin{tikzcd} \textcircled{0} \arrow[r] & \textcircled{1} \arrow[r] & \textcircled{2} \arrow[r] & \textcircled{3} \arrow[lll, bend left] \end{tikzcd}$$

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

ex. for MOD 5 Down Counter, it count as follows

$$\begin{tikzcd} \textcircled{4} \arrow[r] & \textcircled{3} \arrow[r] & \textcircled{2} \arrow[r] & \textcircled{1} \arrow[r] & \textcircled{0} \arrow[llll, bend left] \end{tikzcd}$$

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

	16	8	4	2	1
$(N-1) \rightarrow (17) \rightarrow$	1	0	0	0	1

5 ff are required.

Modulus of a Counter

Total no. of used states are is 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

$$2^n \geq 8$$
$$2^n \geq 2^3$$
$$n = 3$$

$$2^n \geq N$$
$$n \log_2 2 \geq \log_2 N$$
$$\boxed{n \geq \log_2 N}$$

06/11/17 ↗ serial/ripple/frequency divider

Asynchronus MOD 8 Counter (up)

Binary Counter : If $N = 2^n$ Ex. MOD2, MOD4, MOD8

Non-Binary Counter : If $N \neq 2^n$ Ex- MOD3, MOD5, MOD 12

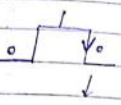
- ∵ $2^n \geq N$
  $2^3 \geq 8$
  $8 \geq 8$
  $n = 3$
  Hence 3 ff are required

[MOD N bin Counter ($N=2^n$) or n bit bin counter]

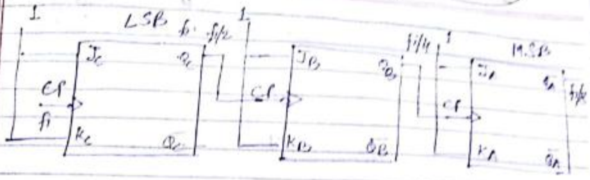
- $$\begin{tikzcd} 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & 7 \arrow[lllllll, bend left] \end{tikzcd}$$

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

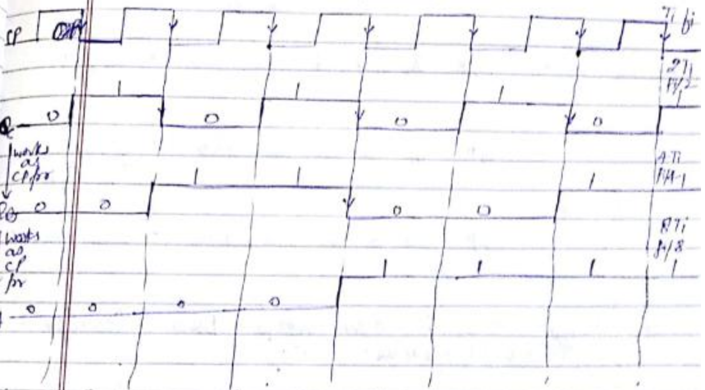
} 1 → 0



→ -ve edge triggering



* Timing Diagram



In Asynchronus counter, external CP is applied to first ff (LSB) & o/p of 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)

$f_o = f_i / 2^n$

$f_o = f_i / N$

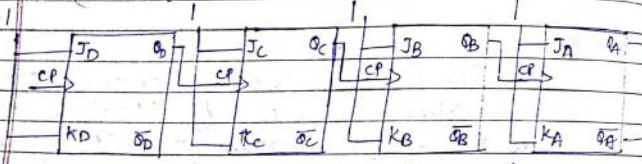
∴ If we take O/P from Q_n & CP is -ve → up counter

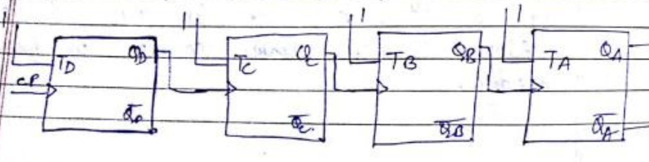
• & O/P $\overline{Q_n}$ & CP is -ve → Down Counter

O/P Q_n & CP is +ve → Down Counter

O/P $\overline{Q_n}$ & CP is +ve → Up counter

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





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

• Design MOD 6 ripple counter (Asyn.) -

(i) $2^n \geq N$

$2^n \geq 6$

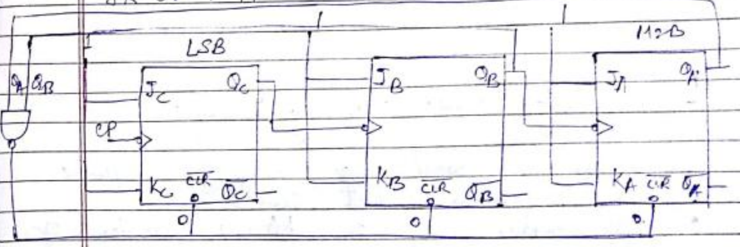
$\Rightarrow 2^3 \geq 6$

$n = 3 \Rightarrow 3$ ff are required

(ii). $0 \to 1 \to 2 \to 3 \to 4 \to 5 \cdots\cdots 6$

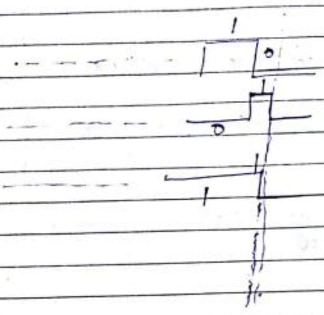
we'll get 6 also for sometime due to propagation delay

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



11 - 1011

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



→ To design MOD N Non-bin. Counter:

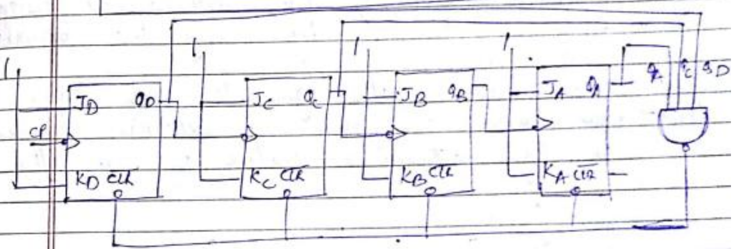
1. Determine total no. of ff required to implement that counter.
2. Now make ckt of binary counter of req. FF bits
3. Now determine binary equivalent of N by indicating Q_A Q_B Q_C ....
4. o/p of NAND gate is connected to $\overline{CLR}$ of all the FF & i/p of NAND gate are connected to o/p of that FF's where 1 comes in binary representation of N.

Q. Design MOD 11 Asyn. Counter

$\therefore 2^n \geq 11$
$2^4 \geq 11$

∴ 4 FF are required

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



Q_A	Q_B	Q_C	Q_D
1	0	1	1

11/10/17

Synchronous Counter (Parallel Counter)

In synchronous counter one ~~o/p~~ common Ci is applied simultaneously to all the flip-flops. That's why it is also called Parallel Counter.

- Procedure to design MOD N synchronous Counter or given sequence –

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

Step 2 – Now draw state dgm & state table

Step 3 – Now with the help of Excitation table of FF, determine value of F.F. i/ps

Q_n	Q_n^+ (Q_{n+1})	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 i/ps which are function of FF's present state.
$\downarrow$
$\boxed{Q_n}$

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

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

$$\begin{tikzcd} 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & 7 \arrow[lllllll, bend left] \end{tikzcd}$$

$2^n \geq N$.

$2^3 \geq 8$

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

Q	Q_n^+	T
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{tikzcd} 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & 7 \arrow[lllllll, bend left] \end{tikzcd}$$

P.S			N.S			F.F. i/ps		
Q_A	Q_B	Q_C	Q_A^+	Q_B^+	Q_C^+	T_A	T_B	T_C
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	1
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 :

$Q_A \backslash Q_B Q_C$	00	01	11	10
0		1	1_{3}	2
1	4	5	1_{7}	6

T_A : $Q_B Q_C$

T_B :

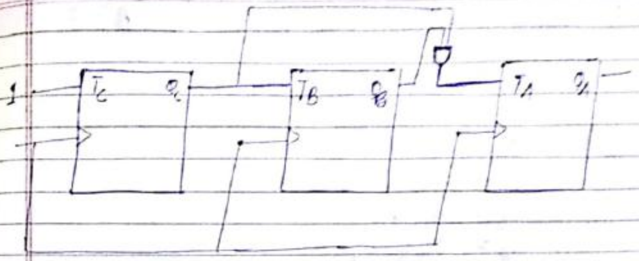
$Q_A \backslash Q_B Q_C$	00	01	11	10
0	0	1_{1}	1_{3}	2
1	4	1_{5}	1_{7}	6

T_B : Q_C

T_C :

$Q_A \backslash Q_B Q_C$	00	01	11	10
0	1_{0}	1_{1}	1_{3}	1_{2}
1	1_{4}	1_{5}	1_{7}	1_{6}

T_C : 1



: Q_C toggles after each CP $\Rightarrow$ $Q_C = 1$

Q_B

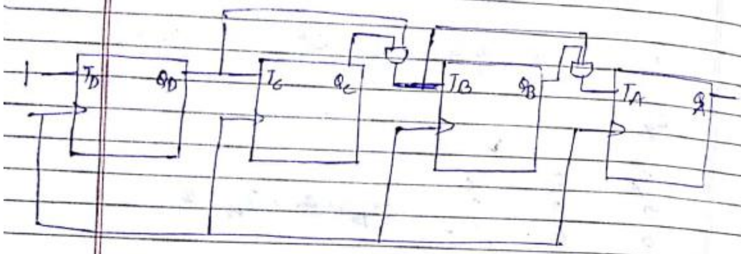
Q_A	Q_B	Q_C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$\rightarrow$ $T_B = Q_C$

$\rightarrow$ $T_A = Q_B Q_C$

Q. Design MOD 16 Syn. Counter by ~~D~~ T FF
↓
4-bit Syn. counter

$T_D = 1$
$T_C = Q_D$
$T_B = Q_C Q_D$
$T_A = Q_B Q_C Q_D$



24/10/17 Time Delay

In Asynchronous Counter -

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

n is total no of ffs & t_{pdff} is prop. delay of each ff

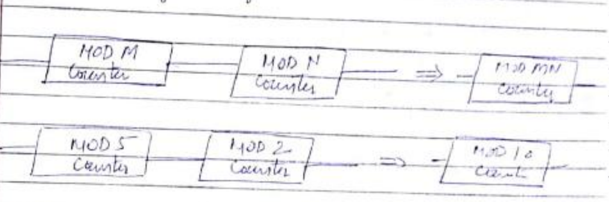
$$f = \frac{1}{T_{clock}}$$

In Synchronous Counter -

$$T_{clock} = t_{pdff} + (n-2)\, t_{pd\,AND}$$

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

For both Asyn & Syn -



for non-binary Counter -

$$o/p \text{ freq.} = \frac{i/p \text{ freq}}{\text{no of used states}}$$

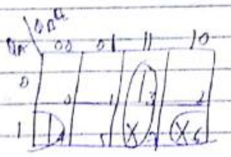
Q Design MOD 6 Synchronous counter by using D-f

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

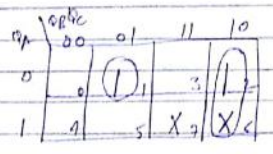
$2^3 > 6$

$\therefore$ 3 FF are required

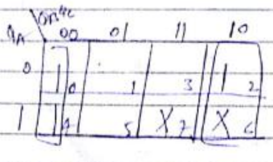
	Q_A	Q_B	Q_C	Q_A^+	Q_B^+	Q_C^+	D_A	D_B	D_C
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0
2	0	1	0	0	1	1	0	1	1
3	0	1	1	1	0	0	1	0	0
4	1	0	0	1	0	1	1	0	1
5	1	0	1	0	0	0	0	0	0
6	1	1	0	d	d	d	d	d	d
7	1	1	1	d	d	d	d	d	d

D_A :


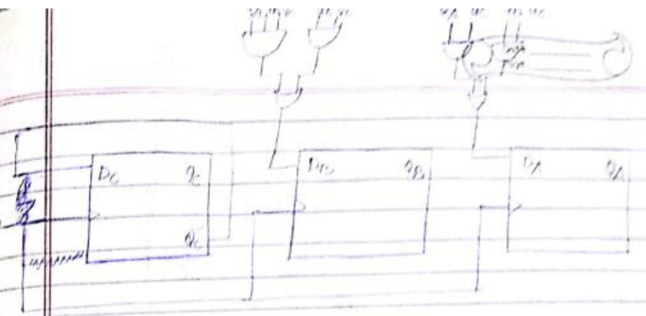
$D_A : Q_A\overline{Q_C} + Q_B Q_C$

D_B :


$D_B : \overline{Q_A}\,\overline{Q_B}\,Q_C + Q_B\overline{Q_C}$

D_C :


$D_C : \overline{Q_C}$



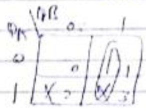
Q. Design a syn. counter which counts the following sequence by using J & K FF

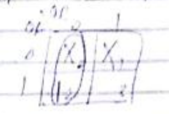
$0 \to 1 \to 3 \to 2 \to 0$

OR MOD 4 Grey sync. counter

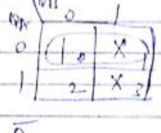
00 : 00
01 → 01
10 → 1●1
11 → 10

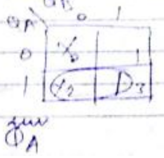
	Q_A	Q_B	Q_A^+	Q_B^+	J_A	K_A	J_B	K_B
0	0	0	0	1	0	d	1	d
1	0	1	1	1	1	d	d	0
2	1	0	0	0	d	1	0	d
3	1	1	●1	0	d	0	d	1

J_A


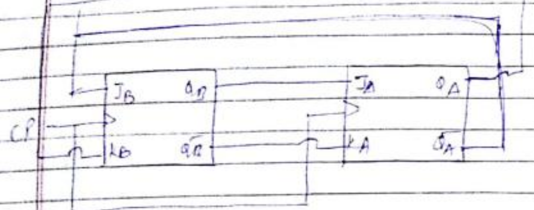
K_A


$J_A : Q_B$ $K_A : \overline{Q_B}$

J_B.


K_B


$\overline{Q_A}$ Q_A



Q. Design a syn. counter which counts the following sequence by using D-FF

$1 \to 2 \to 5 \to 7 \to 1$

		Q_A	Q_B	Q_C	Q_A^+	Q_B^+	Q_C^+	D_A	D_B	D_C
0	1	0	0	1	0	1	0	0	1	0
1	2	0	1	0	1	0	1	1	0	1
2	5	1	0	1	1	1	1	1	1	1
3	7	1	1	1	0	0	1	0	0	1

D_A:

$Q_A \backslash Q_BQ_C$	00	01	11	10
0	X_0	$_1$	X_3	1_2
1	X_4	1_5	$_7$	X_6

D_B:

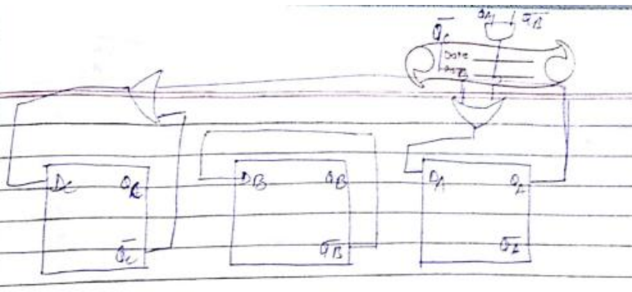
$Q_A \backslash Q_BQ_C$	00	01	11	10
0	X_0	1	X_3	$_2$
1	X_4	1	$_7$	X_6

$D_A: \bar{Q}_C + Q_A\bar{Q}_B$ $\qquad$ $D_B: \bar{Q}_B$

D_C:

$Q_A \backslash Q_BQ_C$	00	01	11	10
0	X_0	$_1$	X_3	1_2
1	X_4	1_5	1_7	X_6

$D_C: \bar{Q}_C + Q_A$



Lock out Cond^n

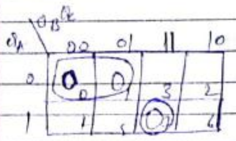
When counter goes from one used state to any unused state & again it goes to another unused state, counter never arrives in used state. This is called lock out Cond^n in counter.

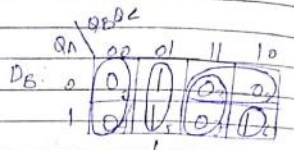
To avoid lock out cond^n we develop such a mechanism if counter goes any unused state then it must go to used state so we use unused state in front of used state.

Q. Draw a syn. counter of following sequence $1 \to 2 \to 5 \to 7 \to 1$ & avoid lock out cond^n.

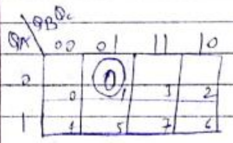
$$\begin{tikzcd}
 & 1 \arrow[r] & 2 \arrow[r] & 5 \arrow[r] & 7 \arrow[r] & 1 \\
0 \arrow[ur, dashed] & & 3 \arrow[ur, dashed] & 4 \arrow[u, dashed] & 6 \arrow[u, dashed] &
\end{tikzcd}$$

	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

D_A:


D_B:


$D_B : \overline{Q_B}Q_C + Q_A Q_B \overline{Q_C}$

D_C:


$D_A : Q_A + Q_B$; $D_B : Q_B Q_C$

$D_A : Q_A \cdot Q_B$; $D_B : (Q_B + Q_C)$

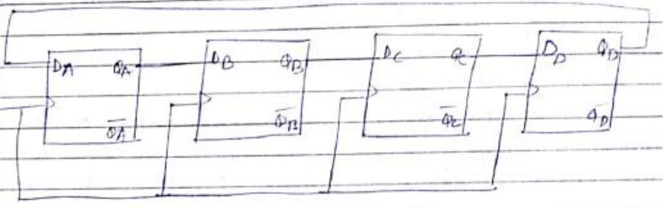
$D_C :$ ~~Q_A+~~ $Q_A(Q_B + \overline{Q_C})$

* **Ring Counter**

To implement n-bit Ring Counter we require n-ffs. In this at one time only one o/p is high. ~~o/p of last ff~~ Uncomplemented o/p of last ff is connected to i/p of first ff

$\nearrow n$

Four-bit Ring counter



CP	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

Right shift → $/2$

$/2^2$

$/2^3$

L.S → $\times 2$, $\times 2^2$, $\times 2^3$

Total used states are $4(n)$

Total Unused states are $2^n - n$ (12)

o/p freq = f_i/n

↳ phase shift → $\frac{360}{90}$ = 4 = $f_i/4$

Scanned by CamScanner

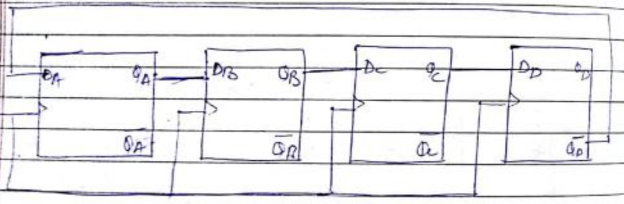
Q. Det. total no of unused state in 8-bit Ring counter

A → $2^8 - 8 = 248$

o/p fq = $f_i/8$

* <u>Johnson Counter / Twisted Ring Counter / ~~Cr~~ Creeping Counter / Switch Tale Counter / Mobile Counter</u>

To implement n-bit Johnson counter, we require n-ff. In this complemented o/p of last ff is connected to i/p of first ~~ckt~~ ff.



CP	Q_A	Q_B	Q_C	Q_D	$\bar{Q}_D$
0	0	0	0	0	1
1	1	0	0	0	1
2	1	1	0	0	1
3	1	1	1	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

25/1

<u>Register</u>

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.

<u>Types of registers</u>:

a). On the basis of Data in & Data out [serially & parallely]

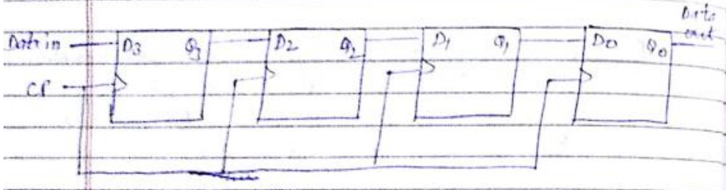
~~Data in~~
- Serial In serial out (SISO)
- SIPO (serial In parallel out)
- PIPO
- PISO

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

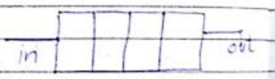
Right shift reg. (Data will shift from L to R)

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

* 4-bit Serial In Serial Out Right Shift Register



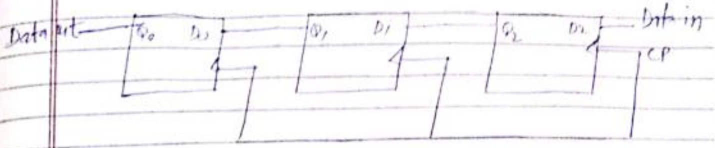
Data 1 0 1 1



CP	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1

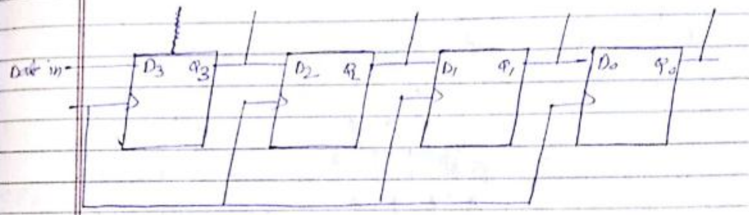
$T_{clock} = n \cdot t_{pdff}$

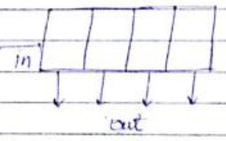
3-bit SISO left shift Register



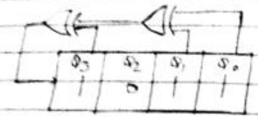
* 4-bit SIPO Shift Register

← Parallel o/p →





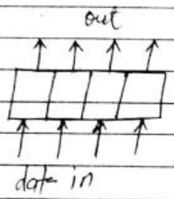
Q. In SIPO shift register what will be value of Q_3, Q_2, Q_1, Q_0 after 3 C.P in the following dgm.

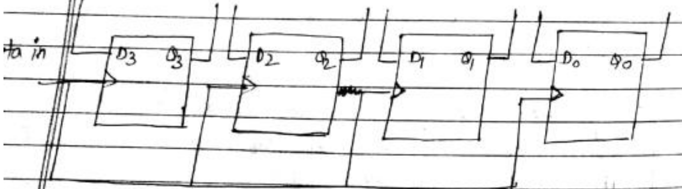


CP	Q_3	Q_2	Q_1	Q_0
0	1	0	1	1
1	1	1	0	1
2	0	1	1	0
3	1	0	1	1

Ans → 1011

* PIPO Shift Register





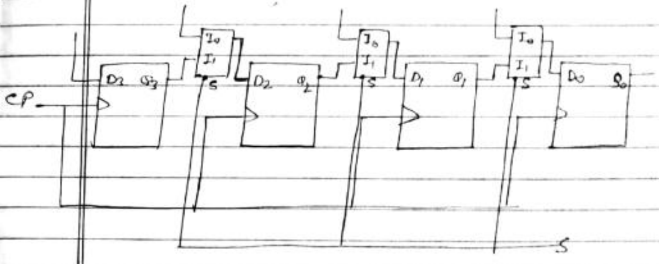
In this input is taken parallely (simultaneously) & o/p is also taken parallely

$$T_{clock} = t_{pd_{ff}}$$

* ~~PIPO~~ PISO shift Register

In this i/p is ~~taken~~ given parallely but o/p is taken serially (1-bit at a time)

→ 4-bit PISO shift Register



S = 0 ⇒ I_0 will work (parallel in)

S = 1 ⇒ 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 Register are those registers which perform the following operations.

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

* Applications of Shift Register –

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