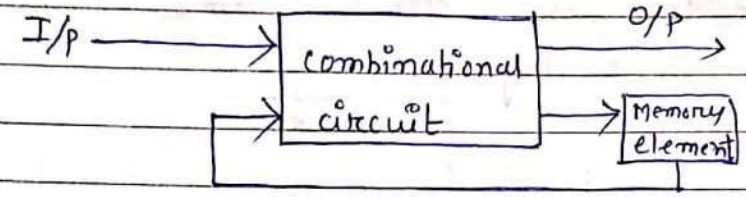
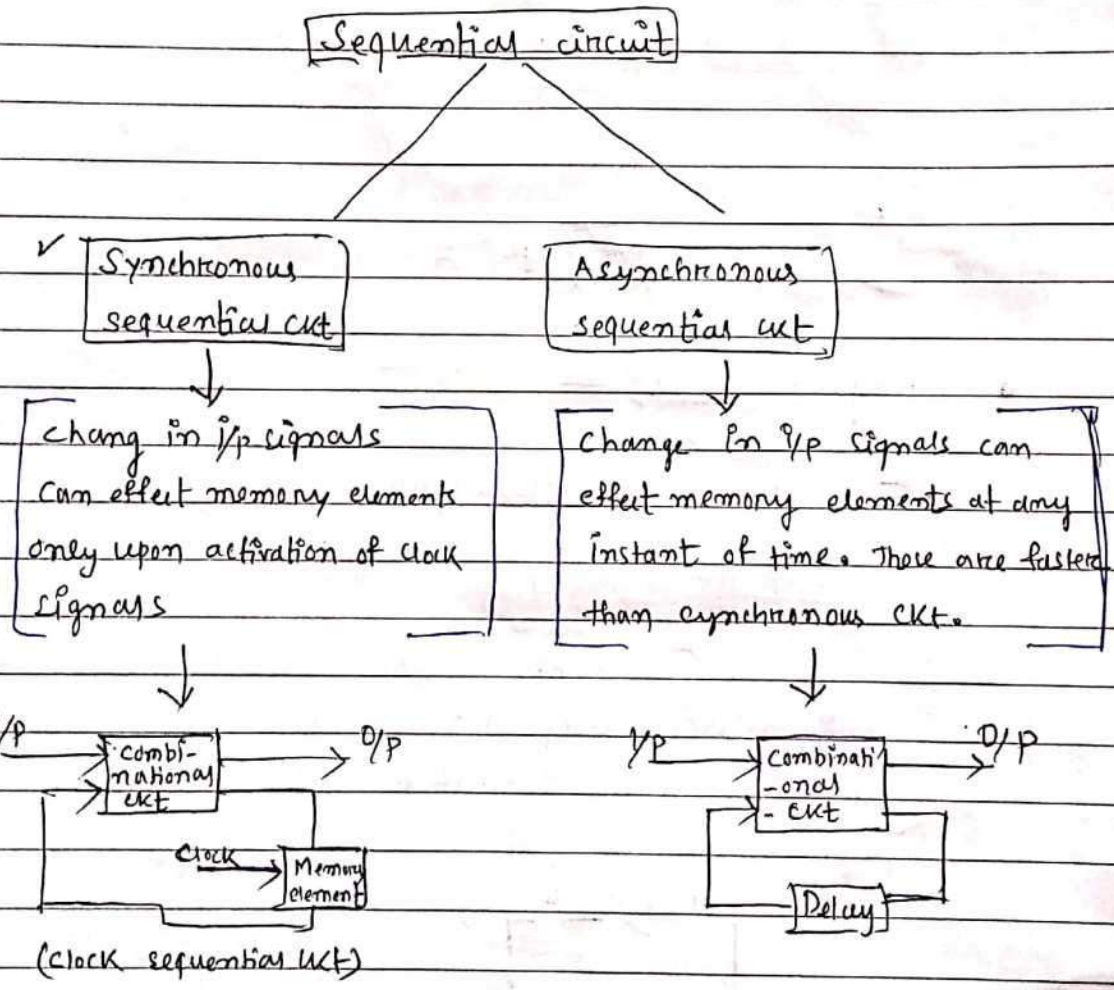
Sequential circuit

- **INTRODUCTION of Sequential circuit** —

In sequential logic circuit, the o/p is depent on upon the present i/p as well as the past I/p and output.





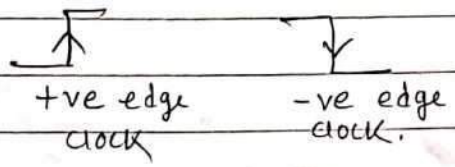
→ The storage elements (memory) used in clocked sequential circuit are called "flip-flop".

→ flip flop is a one bit memory cell which stores the 1-bit signal data (logic '0' or logic '1').

→ In synchronous sequential circuit, memory elements are clocked flip-flop and generally edge triggered.

→ In Asynchronous sequential circuit, memory elements are unclocked flip-flop / time delay elements which are generally level triggered.

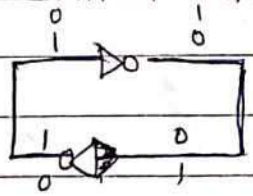
→ flip-flop circuit is also known as bistable multivibrator or latch because it has two stable state (1 state, 0 state).



• Storage Elements —

- latches. ⟶ (SR latch, D latch)
- flipflops. ⟶ (SR, JK, D, T).

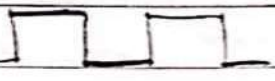
" A storage Elements in a digital circuit can ~~main~~ maintain a binary state indefinitely as long as power is delevere to the circuit, until directed by an i/p signal to switch states.



Bistable element

- **Latches**:

Storage elements that are operate with signal levels are referred to as "latches".

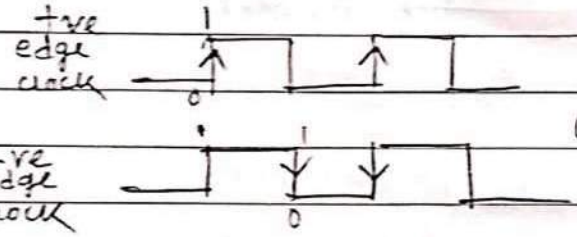


(Latches are level sensitive)

→ Latches are used in asynchronous sequential circuit for storing binary information and for the design.

- **Flip-flop**:

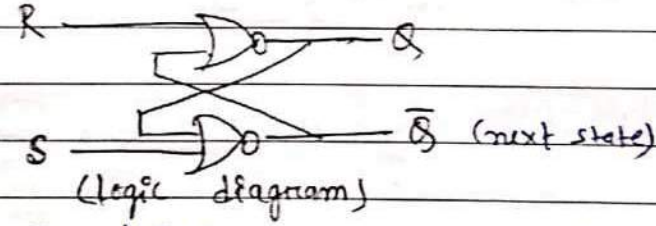
Storage elements that are controlled by a clock transition are flip-flop.



(flip-flop are edge sensitive)

→ **SR-Latch (Set-Reset Latch)** =

(1) SR Latch using NOR gate. =



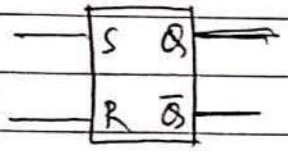
(logic diagram)

Function table =

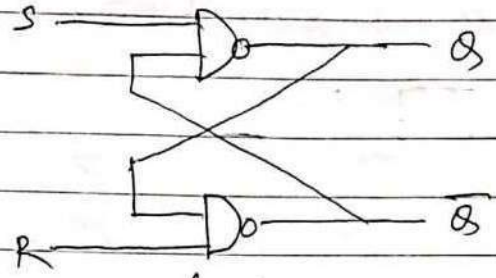
S	R	Q	$\bar{Q}$
0	0	No change	
0	1	0	1
1	0	1	0
1	1	—	

→ undefined state or forbidden state.

Symbol =



(II) SR-Latch using NAND gates —



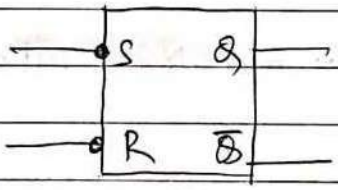
(logic diagram)

Function table —

S	R	Q	$\bar{Q}$	
0	0			→ forbidden state
0	1	1	0	
1	0	0	1	
1	1			→ NO change

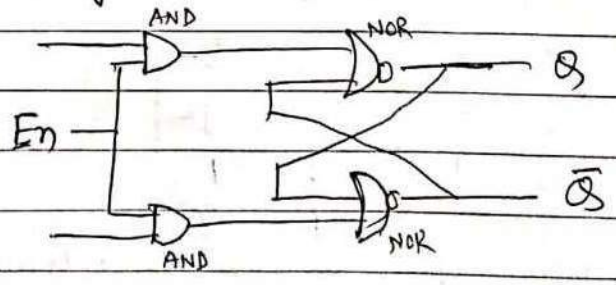
→ Opposite of function table of SR latch using NOR gate.

Symbol —



(III) SR-Latch using control inputs —

(a) using NOR gates —



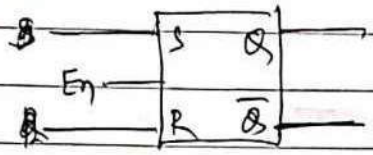
Operation –

When En = 0 (Q & $\bar{Q}$ will remains in previous state)

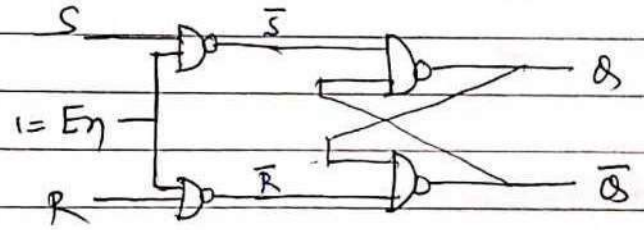
En = 1

S	R	Q	$\bar{Q}$
0	0	No-change	
0	1	0	1
1	0	1	0
1	1	~~undefine~~	→ forbidden state.

Symbol –



(b) Using NAND gate –



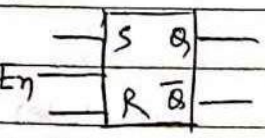
Operation –

When En = 0, (Q & $\bar{Q}$ remains in previous state)

En = 1

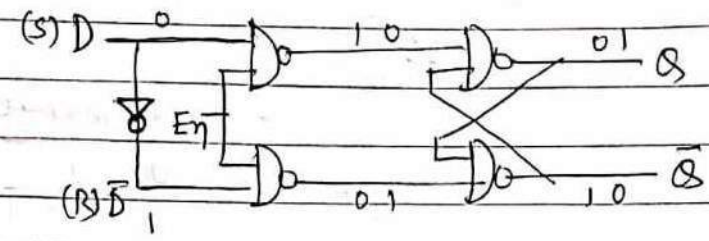
S	R	Q	$\bar{Q}$	
0	0	—	—	→ No-change
0	1	0	1	
1	0	1	0	
1	1	—	—	→ forbidden state.

Symbol –



→ Using control i/p with we will get same function table (NOR/

- **D-latch (Transparent latch) — (Delay latch)**

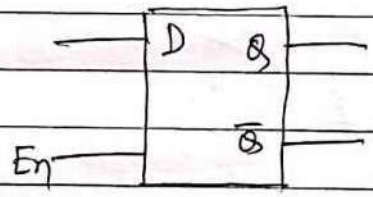


→ To remove undefine state we use D-latch.

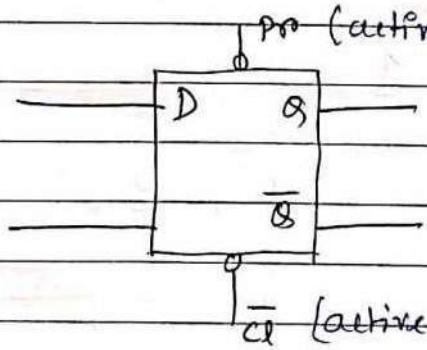
Function table —

En	D	Q	$\bar{Q}$
0	X	Previous state	
1	0	0	1
1	1	1	0

Symbol —

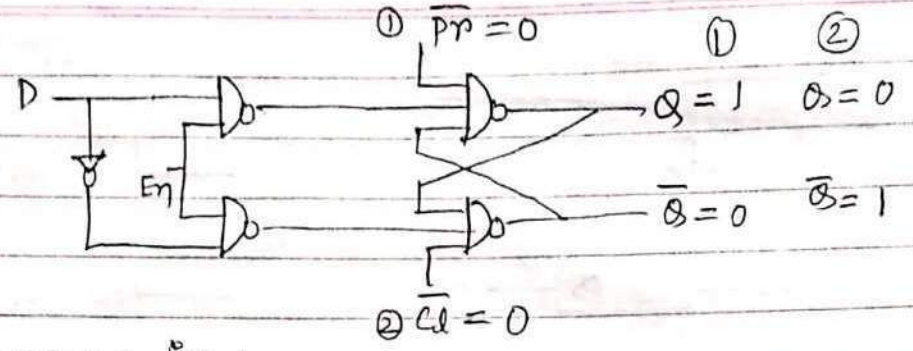


- **D-latch with preset & clear i/p's — (asynchronous i/p's)**



$\overline{Pr} = 0 \rightarrow Q = 1$ (set state)

$\overline{Cl} = 0 \rightarrow Q = 0$ (reset state)



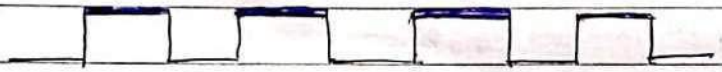
asynchronous i/p because — (operation of the i/p will not depend on the i/p of D & En)

- **Flip-Flop** —

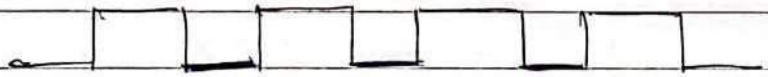
There are '4'-types of flip-flop —

(I) S-R - flip flop.
(II) J-K - flip flop.
(III) D - flip flop.
(IV) T - flip flop.

*. Clock response in latch —

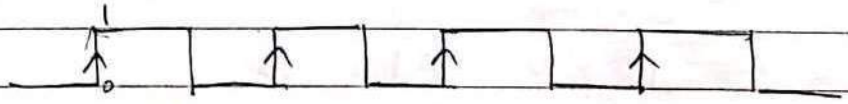


(I) response to positive level.

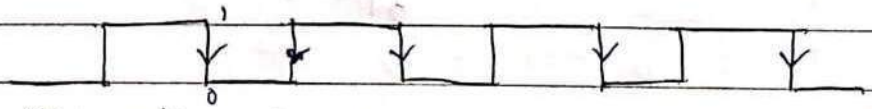


(II) response to negative level.

*. clock respone in flip-flop. —

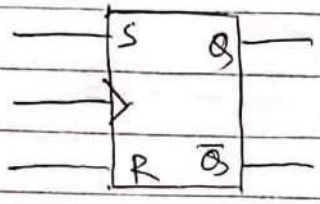


(I) Positive edge response.

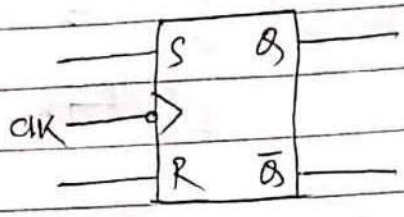


(II) negative edge response.

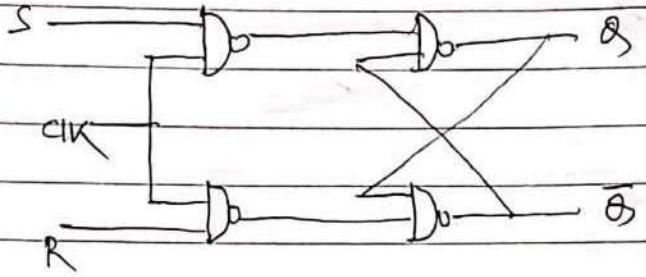
• SR-flip-flop —



+ve edge SR-FF



-ve edge SR-FF.



Truth Table / Function Table —

CLK	S	R	Q	$\overline{Q}$
0	X	X	No change	
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	Forbidden state.	

Characteristic table —

P.S			Next.S
Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	forbidden state.
1	0	0 ✓	1
1	0	1	0
1	1	0 ✓	1
1	1	1	forbidden state.

Characteristic equation -

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

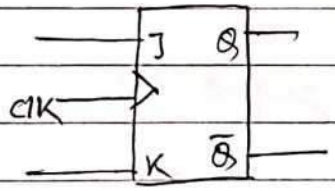
K-map

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

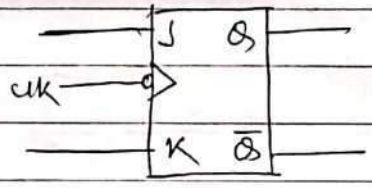
Excitation table -

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	⊗
1	0	0	1
1	1	X	0

- **J-K - flip-flop =**

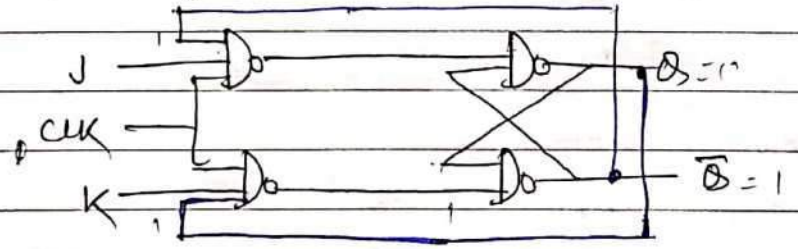


+ve edge JK-FF



-ve edge JK-FF.

Logic Diagram →



function table –

clk	J	K	Q	$\bar{Q}$
0	X	X	No-change	
↑	0	0	No-change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	previous state compliment.	

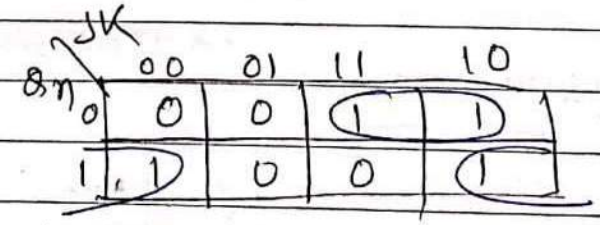
Characteristic table –

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
01	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Characteristic equation –

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

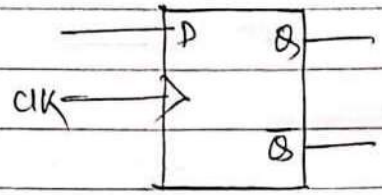
Kmap,



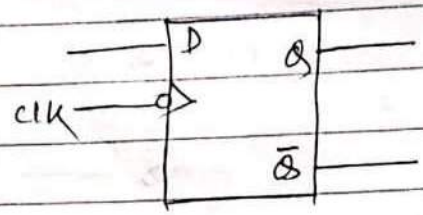
Excitation table –

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

• D-Flip-Flop —

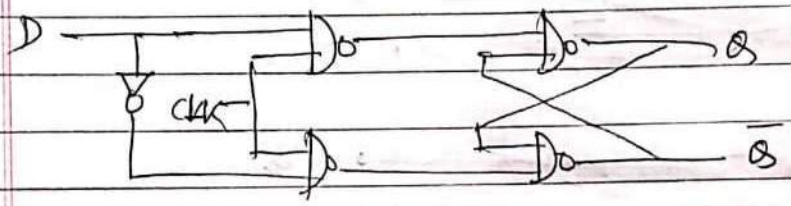


positive edge D-FF



negative edge D-FF.

logic Diagram —



Function Table —

CLK	D	Q_n	Q_{n+1}
0	X	No change	
↑	0	0	1
↑	1	1	0

Characteristic Table —

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

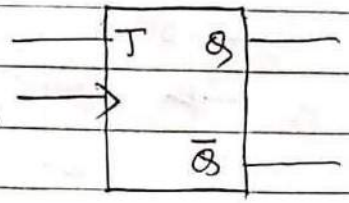
Characteristic equation —

$Q_{n+1} = D$.

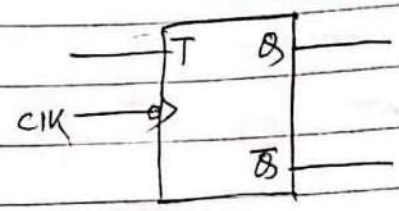
Excitation Table —

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

• T-FF (Toggle FF) —

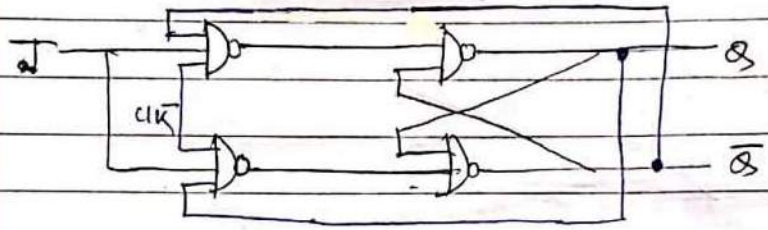


Positive edge T-FF



negative edge T-FF.

Logic Diagram —



Function table —

CLK	T	Q_n Q_{n+1}
0	X	No change. (Q_n)
↑	0	No change (Q_n)
↑	1	e.g. previous ~~as~~ state compliment ($\bar{Q}_n$)

Characteristic Table —

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

Characteristic equation —

$$Q_{n+1} = Q_n\bar{T} + \bar{Q}_n T = Q_n \oplus T$$

Q_n \ T	0	1
0	0	(1)
1	(1)	0

Excitation Table —

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

• Conversion of flip flops —

→ Characteristic table of required FF.
→ Excitation table of existing FF.

• Function Table of all FFs —

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

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

D	Q_{n+1}
0	0
1	1

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

• Excitation table of all the FFs —

Q_n	Q_{n+1}	S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0

Example - 1

SR - FF to JK - FF :

Required-FF | Existing-FF

Q_n	J	K	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

S

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

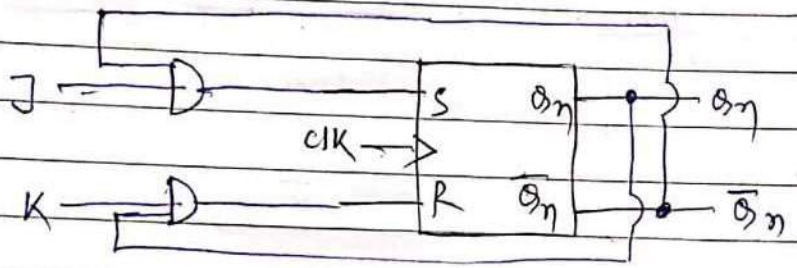
$S = \overline{Q_n} J$

R

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

$R = Q_n K$.

Block Diagram –



Example – 2

SR to D-FF :

Q_n	D	Q_{n+1}	S	R
0	0	0	0	X
0	1	1	1	0
1	0	0	0	1
1	1	1	X	0

S

Q_n \ D	0	1
0	0	1
1	0	X

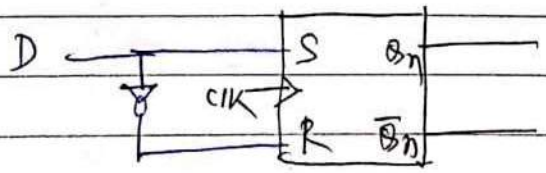
$S = D$

R

Q_n \ D	0	1
0	X	0
1	1	0

$R = \overline{D}$

Block Diagram –



Example – 3

JK-FF to SR-FF :

Q_n	S	R	Q_{n+1}	J	K
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	X
0	1	1	X	X	X
1	0	0	1	X	0
1	0	1	0	X	1
1	1	0	1	X	0
1	1	1	X	X	X

J

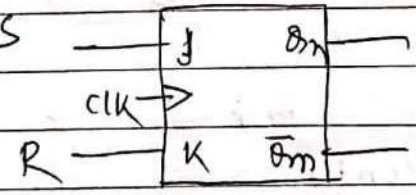
Q_n \ SR	00	01	11	10
0	0	0	X	1
1	X	X	X	X

K

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

$J = S$ $\qquad$ $K = R$

Block diagram –



Example – 4

T-FF to D-FF :

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

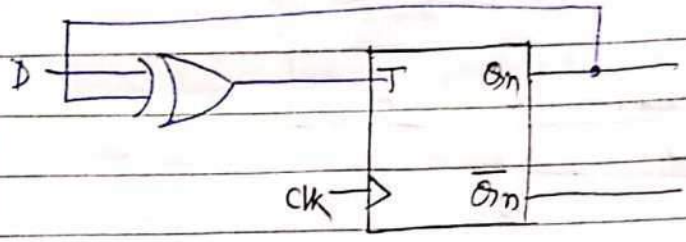
T

Q_n \ D	0	1
0	0	(1)
1	(1)	0

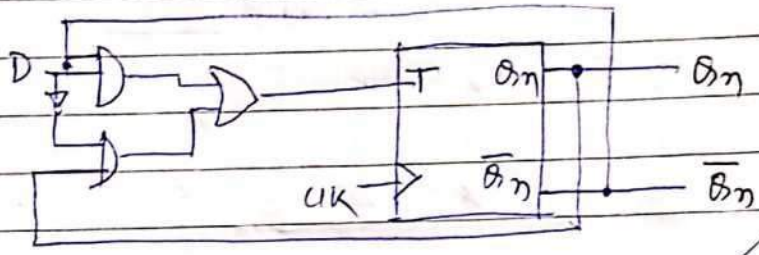
$T = \bar{Q}_n D + Q_n \bar{D}$

$= Q_n \oplus D$

Block Diagram –

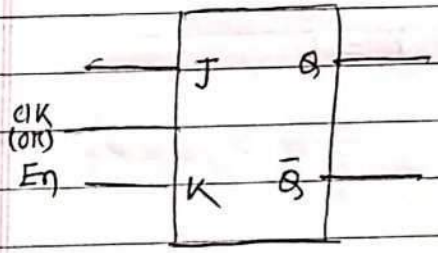


or



- **RACE AROUND CONDITION :**

→ It occures in level triggered JK-flip flop.



function table –

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

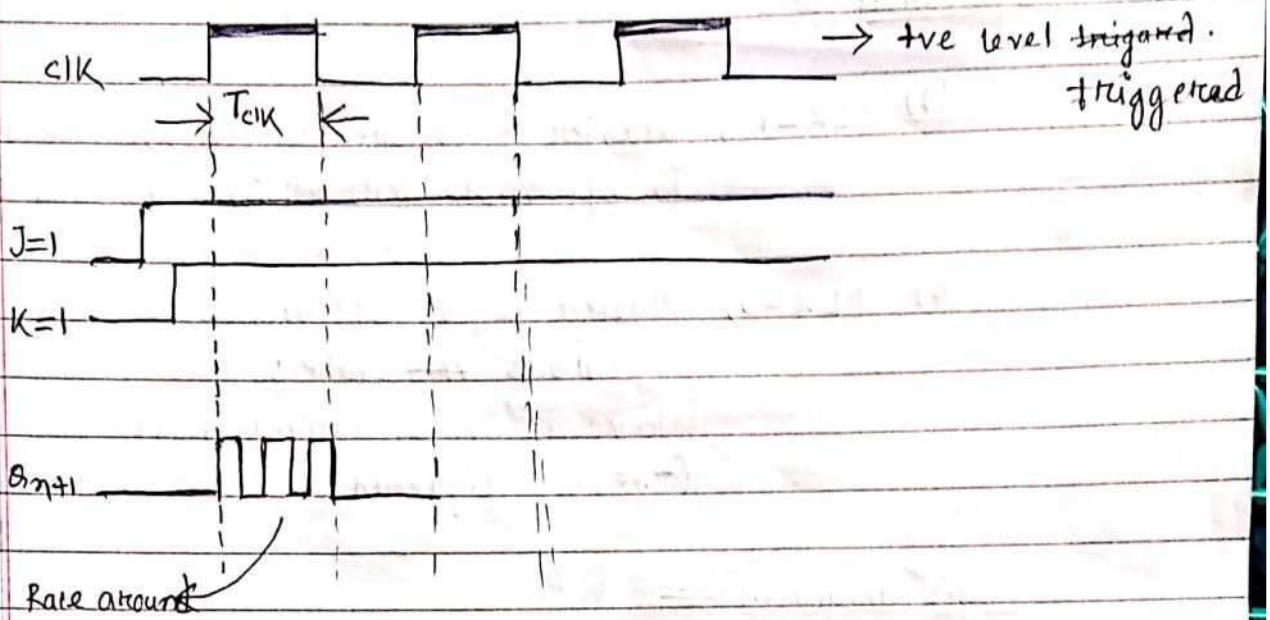
Race around condition ←

Timing Diagram –

Next page

CLK

$t_{pd} << T_{CLK}$

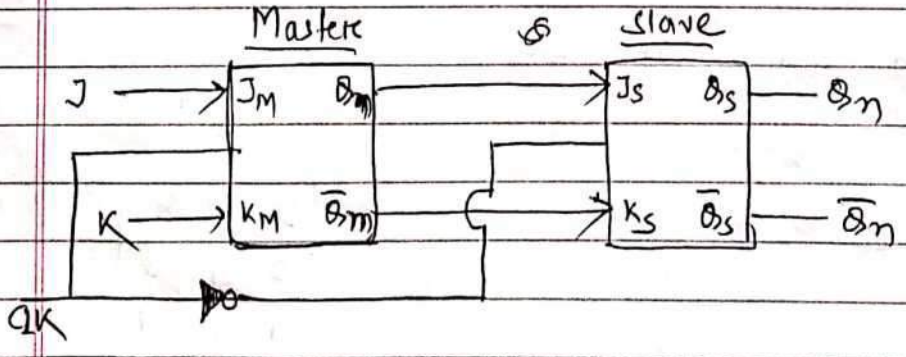


i.e when J=K=1 and $t_{pd} << T_{CLK}$ the state of the flip-flop will oscillates between '0' & '1' and at the end of the clock ~~pulse~~ pulse, the FF state is uncertain, This phenomenon is called "Race around condition".

Avoid race around condition ==

→ Master-Slave level triggered JK-FF.
→ $t_{pd} > T_{CLK}$.

Master-Slave JK-FF ==

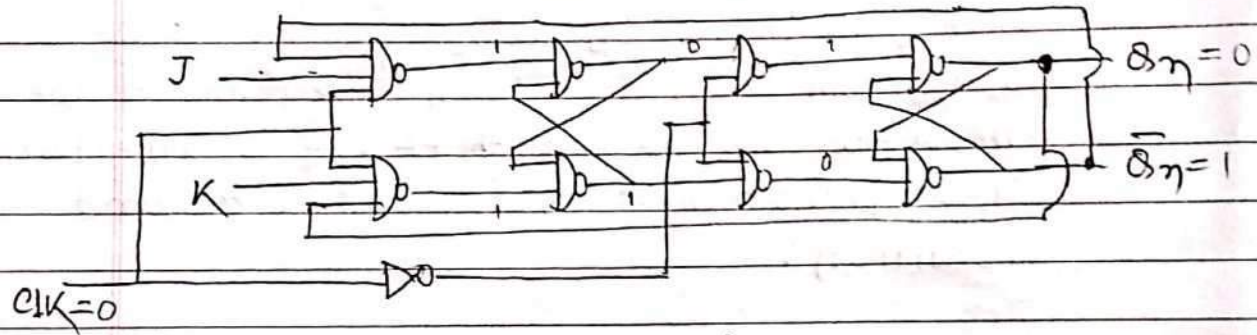


Operation =

(i) CLK=1, Master is in operating mode. Slave is in off mode (or) retains previous state.

(ii) CLK=0, Master is in off mode (or) retaining previous ~~mode~~ state; slave is in operating mode. (-ve edge triggered)

Logic diagram =



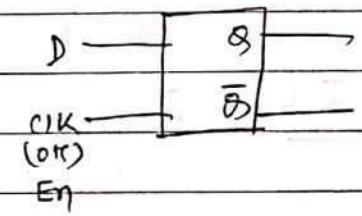
(-ve edge triggered FF)

When CLK=0, then slave ~~to~~ active & and we get result So, It is -ve edge triggered FF.

• Master-salave D Flip-Flop =

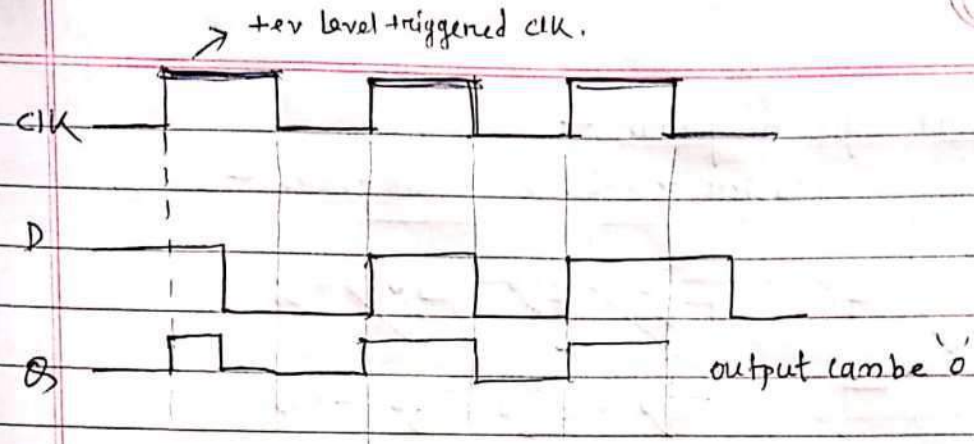
(or) D-FF (or) Edge triggered D-FF.

problem with D-latch =

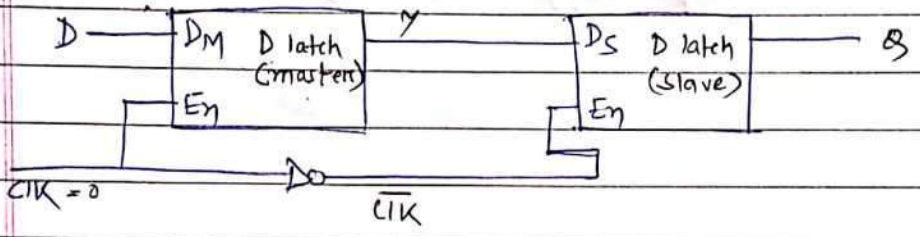


Function Table -

D	Q_n
0	0
1	1



Master slave D-FF =



Operation -

→ clk = 1 (logic-1 level). the external data D i/p is transferred to the master.

But slave is disabled (∴ $\overline{clk} = 0$) because its enable i/p is = 0

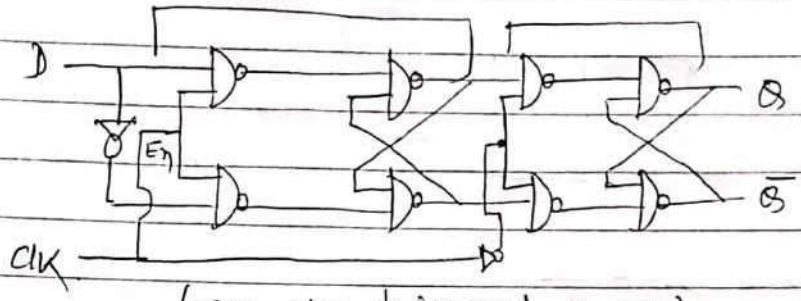
→ When clk = 0, the master latch is disable, and the slave latch is enable ($\overline{clk}_{En} = 1$) and its output Q is equal to the master output y.

behaviour of the master-slave FF ~~dict~~ dictates that -

(a) The o/p may change only once.
(b) a change in the o/p is triggered by the negative (-ve) edge of the clock.
(c) the change may occur only during the clock's -ve level.

Internal logic diagram –

Master → D-Latch SR-latch ← Slave.



(-ve edge triggered D-FF)

- **COUNTER:**

→ "A counter is sequential logic circuit capable to counting the number of clock pulses arriving at its clock i/p."

or

"A counter is a set of FF whose states changes in response to clock pulses applied at the i/p to the counter."

→ A counter can be used as a frequency divider.

→ with n-FFs maximum possible states in the counter is 2^n

$$N \leq 2^n$$

$n \rightarrow$ no. of FFs.

$N \rightarrow$ no. of states.

↳ modulus of the Counter.

Ex:

$n = 2 \longrightarrow N \leq 4 \rightarrow$ Mod 4 Counter.

$n = 3 \longrightarrow N \leq 2^3 \rightarrow N \leq 8 \rightarrow$ Mod 8 counter

↳ Mod 6 counter.

Ex:

MOD 16 counter, find $n = ?$

here,

$N \leq 16 \qquad n = 4$

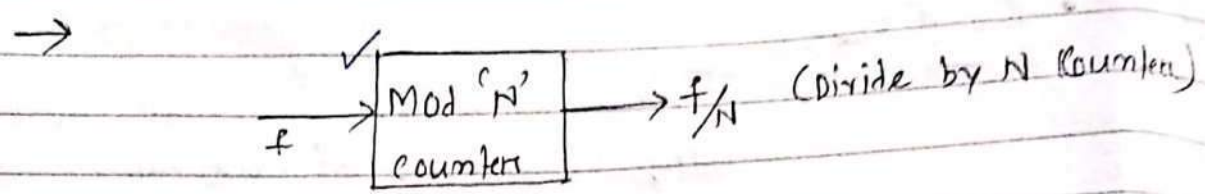
$N \leq 2^4$

Ex: MOD 6 counter, find $n = ?$

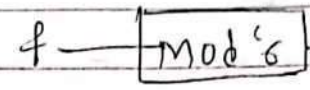
here,

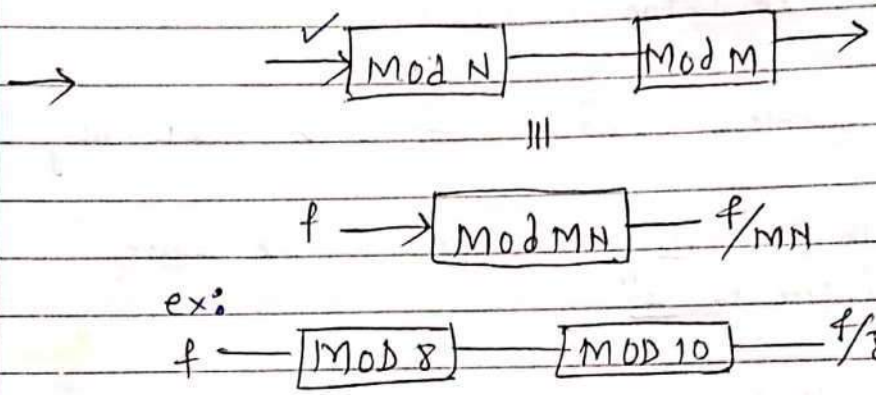
$N \leq 2^3$

$n = 3$ (minimum).



ex:





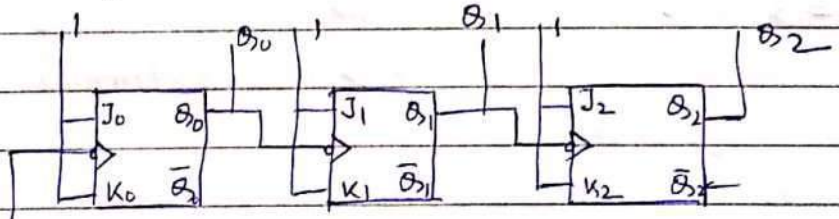
ex:

• Types of counter —

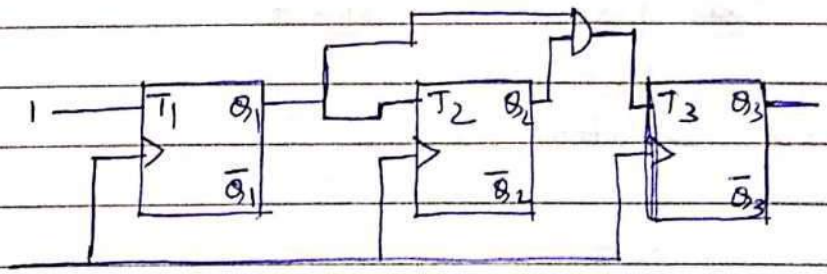
Depending on the clock pulse applied counters of two type:

→ Asynchronous counter or (ripple counter).

→ Synchronous counter.



(Asynchronous counter)



(Synchronous Counter)

Asynchronous counter	Synchronous counter
→ In this type of counter FFs are connected in such a way that the O/p of 1st FF drives the clock for the 2nd FF, the O/p of the 2nd will be the clock of 3rd FF and so on.	→ In this type of ~~counter connection b/w~~ counter there is no connection b/w the O/p of 1st FF & clock i/p of next FF and so on.
→ All the FFs are not clocked simultaneously.	→ All the FFs are clocked simultaneously.
→ Design & Implementation is very simple even for more no. of ~~FF~~ states.	→ Design and Implementation becomes complex as the no. of states increases.
→ Low speed.	→ High speed.

• Asynchronous / Ripple Counters —

→ 2-bit counter → no. of FFs = 2
$n=2 \rightarrow 2^n = 4 \rightarrow$ MOD - 4 counter.
(0, 1, 2, 3)

→ counting sequence 0, 1, 2, 3 (up-counter)
counting sequence 3, 2, 1, 0 (Down-counter)

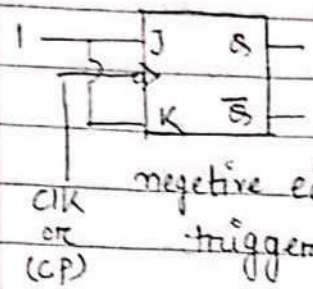
→ JK-FF and T-FF are use of ~~Asn~~ Asynchronous counter.

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

Toggle state

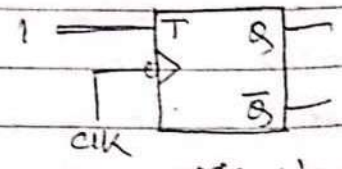
T	Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

JK-FF

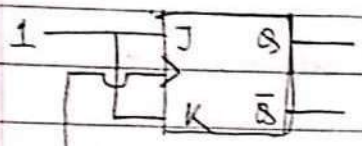


negative edge triggered.

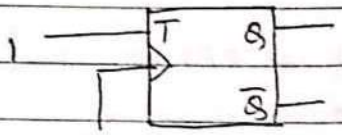
T-FF



-ve edge triggered.

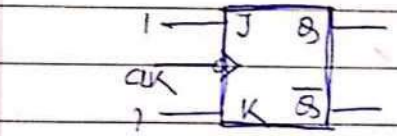


positive edge triggered

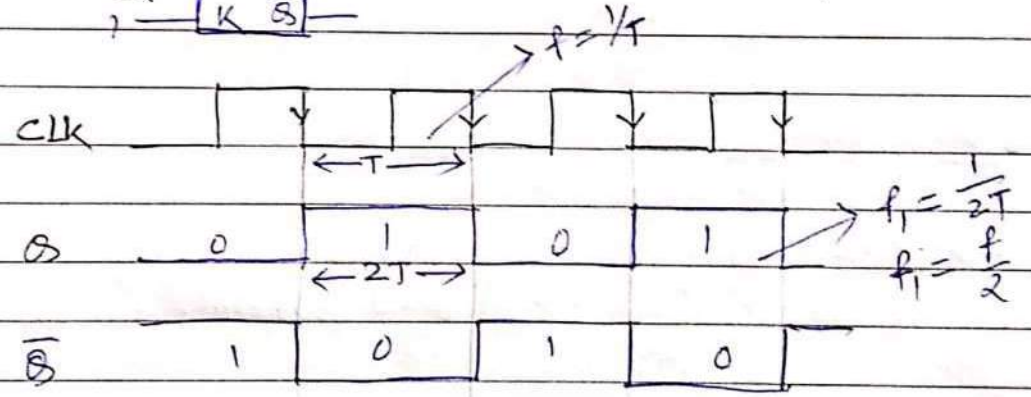


+ve edge triggered.

- Negative edge JK flip flop —



→ Divide by 2 counter.



Truth Table

	CLK	Q	$\bar{Q}$
initial	0	0	1
↓	1	1	0
↓	2	0	1
↓	3	1	0

up counting | down counting

$f_1 = f/2$

freq divider

1 bit counter.
or
MOD-2 counter.

• 2-bit Up counter using negative-edge triggered JK-FF:

$n = 2$, $N = 2^2 = 4$

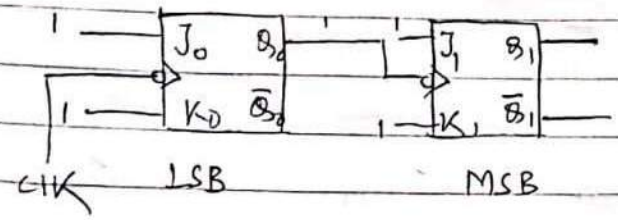
↓ no. of FFs

MOD - 4 counter

or

Divide by 4 counter.

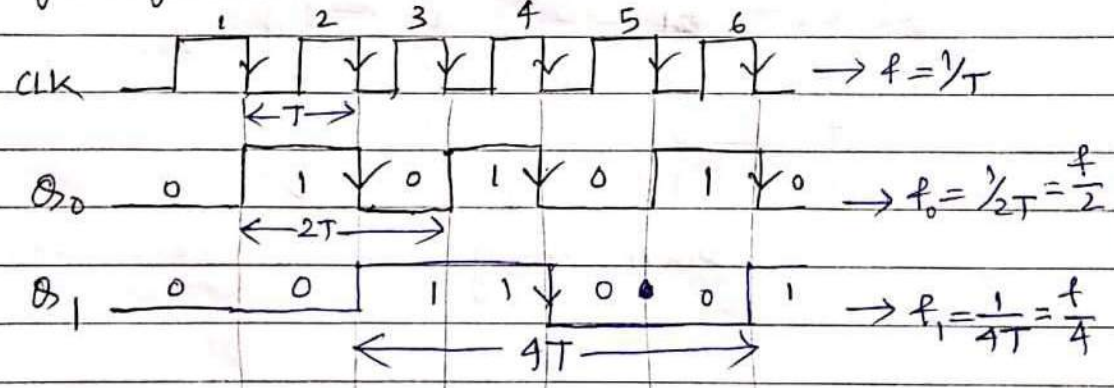
00
01
10
11



CLK	Q_1	Q_0	$\bar{Q}_1$	$\bar{Q}_0$
0	0	0	1	1
↓	0	1	1	0
↓	1	0	0	1
↓	1	1	0	0
↓	0	0	1	1

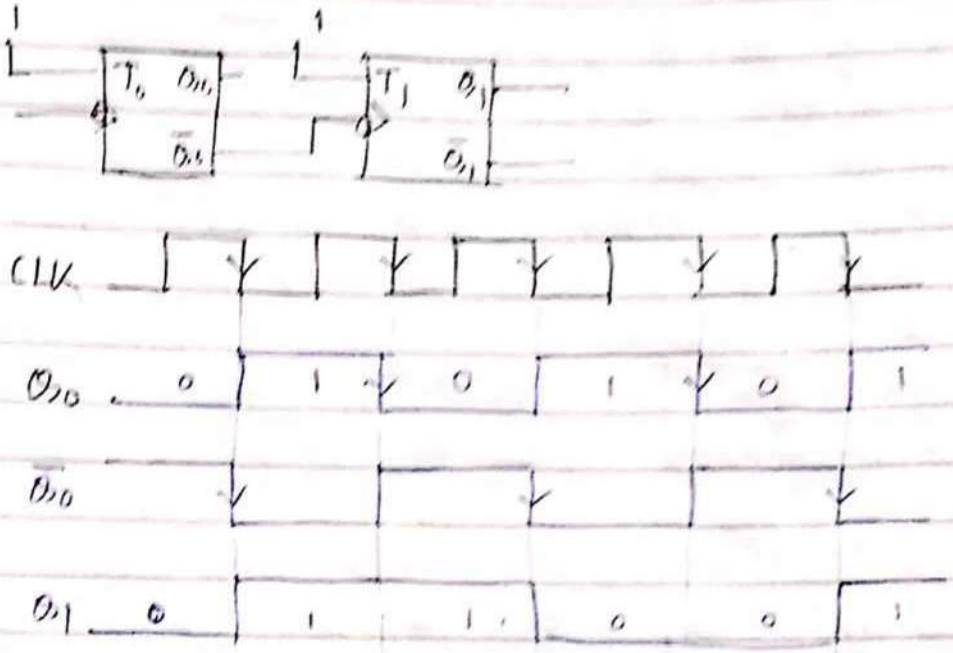
(up counting) (down counting)

Timing Diagram :-



Divide by 4-counter.

- 2 bit Down Counter using -ve edge triggered T-F:



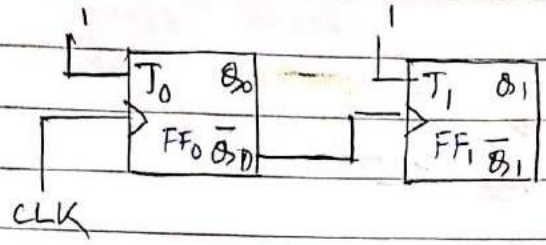
(Down - counting)

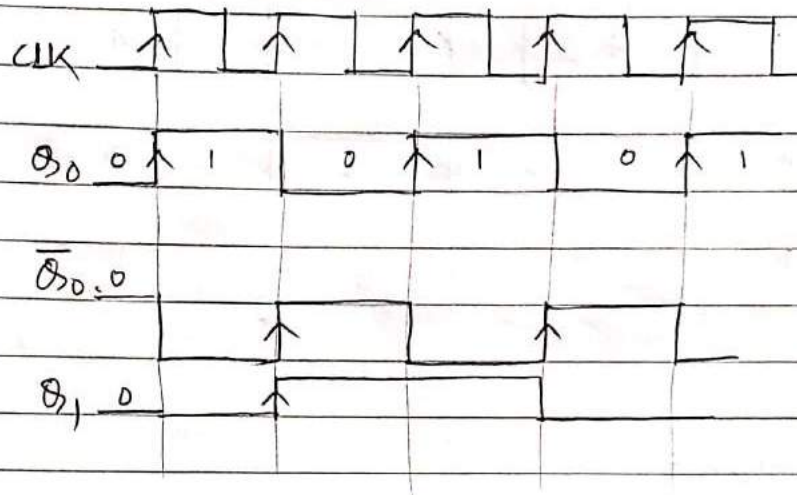
Truth table —

	CLK	Q_1	Q_0	$\bar{Q}_1$	$\bar{Q}_0$
	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
↓	5	1	1	0	0

(Down counting)

- 2-bit up-counter using +ve edge-triggered J-FF =





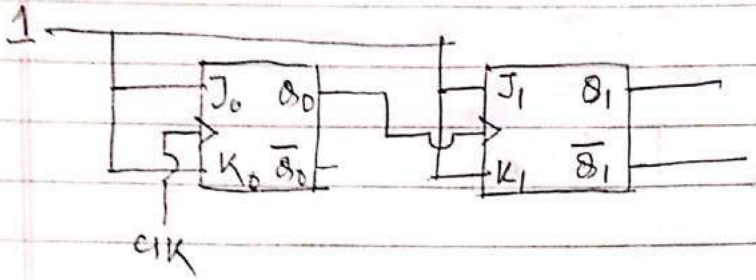
Truth Table =

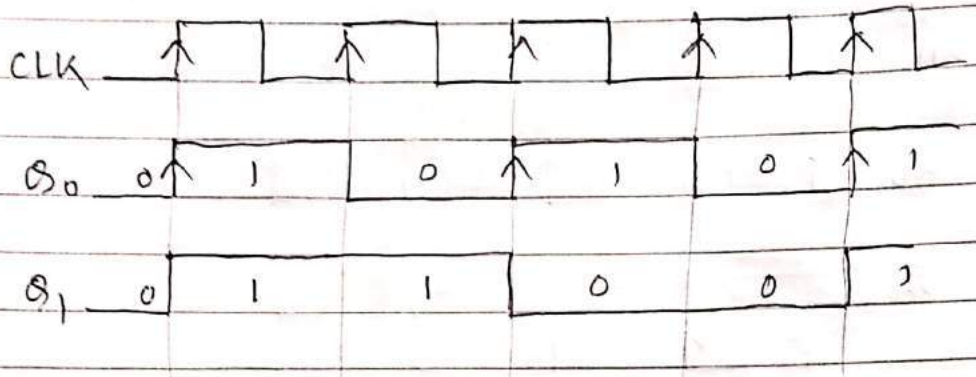
CLK	Q_1	Q_0	$\bar{Q}_1$	$\bar{Q}_0$
0	0	0	1	1
↑ 1	0	1	1	0
↑ 2	1	0	0	1
↑ 3	1	1	0	0
↑ 4	0	0	1	1
↑ 5	0	1	1	0

up counting
MOD-4 Up-counter
divide by 4 counter.

Q. CLK frequency $f = 1$ MHz, using 2-bit counter
What is O/P frequency =
→ $f = \frac{1}{2^2} = \frac{1}{4}$ MHz.

• 2-bit Down counter using +ve edge-triggered JK-FF =





Truth Table =

	CLK	Q_1	Q_0	$\bar{Q}_1$	$\bar{Q}_0$
↓	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
.	5	1	1	0	0

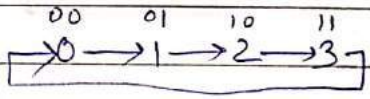
Down counter (MOD -4 counter)

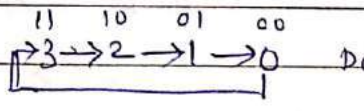
Note: $n = 2 \rightarrow 2^n = 4$

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

$n = 4 \rightarrow 2^n = 16$

$N = 16$





n-bit counter

or MOD-N counter.

↓

max no. of states

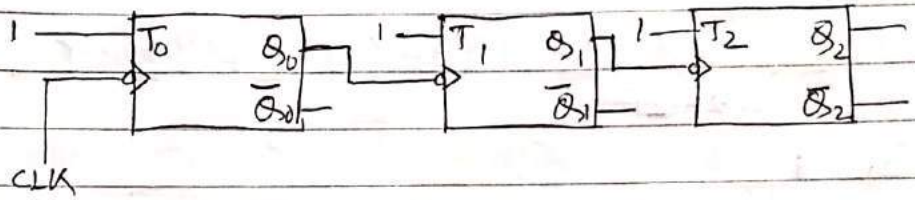
or

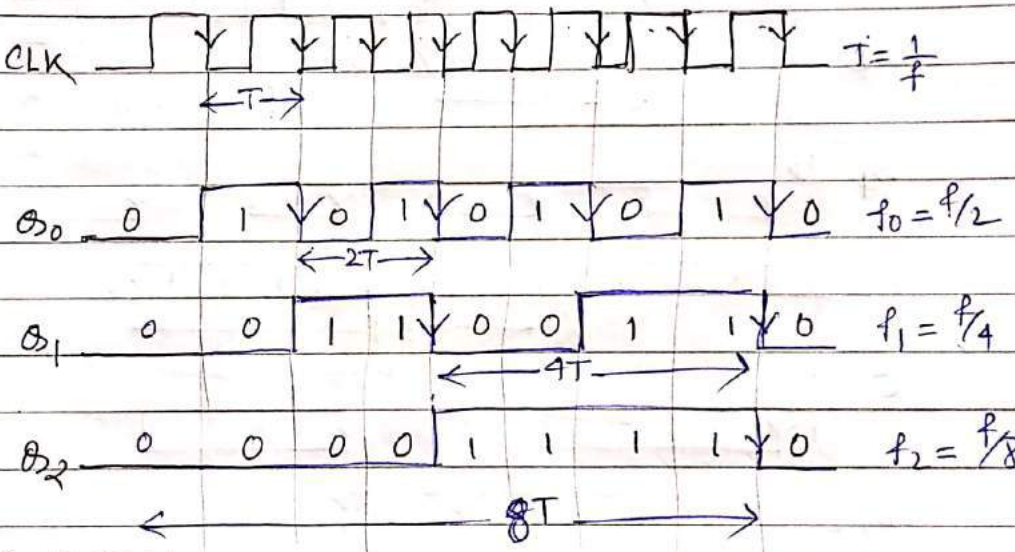
Divide by N counter. = f/N

Note:

(I) −ve Edge triggered FF → Q as CLK → Up counter.
(II) −ve " " " → $\bar{Q}$ " " → Down counter.
(III) +ve " " " → Q " " → Down " .
(IV) +ve " " " → $\bar{Q}$ " " → Up " .

- **3 bit Ripple Counter** = (up counter)
  or MOD 8 - Ripple counter (0, 1, 2, 3, 4, 5, 6)





Truth Table

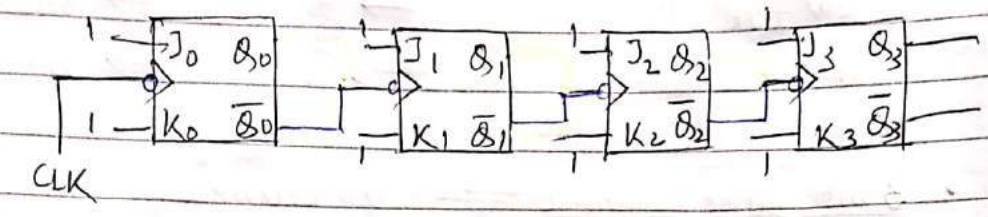
CLK	Q_2	Q_1	Q_0	$\bar{Q}_2$	$\bar{Q}_1$	$\bar{Q}_0$
0	0	0	0	1	1	1
1	0	0	1	1	1	0
2	0	1	0	1	0	1
3	0	1	1	1	0	0
4	1	0	0	0	1	1
5	1	0	1	0	1	0
6	1	1	0	0	0	1
7	1	1	1	0	0	0
8	0	0	0	1	1	1

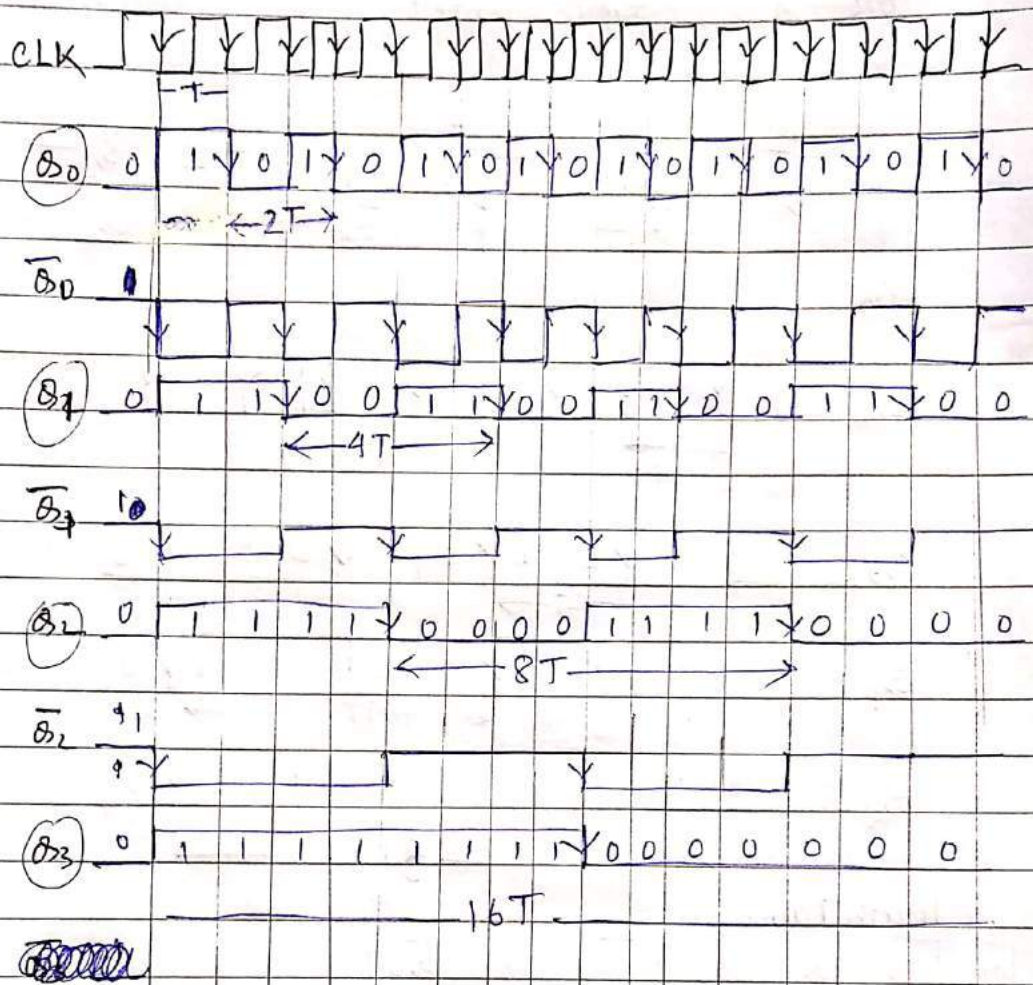
$T = \frac{1}{f}$

0 → 1 → 2 → 3 → 4 → 5 → 6 → 7

- MOD -16 Ripple counter / 4 bit Ripple counter =

$N = 16 = 2^{④} \rightarrow n$ (Down Asynchronous counter)





Truth Table — Ripple (Down) Counter

CLK	Q_3	Q_2	Q_1	Q_0	
→ 0	0	0	0	0	(0)
↓ 1	1	1	1	1	(15)
↓ 2	1	1	1	0	(14)
↓ 3	1	1	0	1	(13)
↓ 4	1	1	0	0	(12)
↓ 5	1	0	1	1	(11)
↓ 6	1	0	1	0	(10)
↓ 7	1	0	0	1	(9)
↓ 8	1	0	0	0	(8)
↓ 9	0	1	1	1	(7)
↓ 10	0	1	1	0	(6)
↓ 11	0	1	0	1	(5)
↓ 12	0	1	0	0	(4)
↓ 13	0	0	1	1	(3)
↓ 14	0	0	1	0	(2)
↓ 15	0	0	0	1	(1)
↓ 16	0	0	0	0	(0)

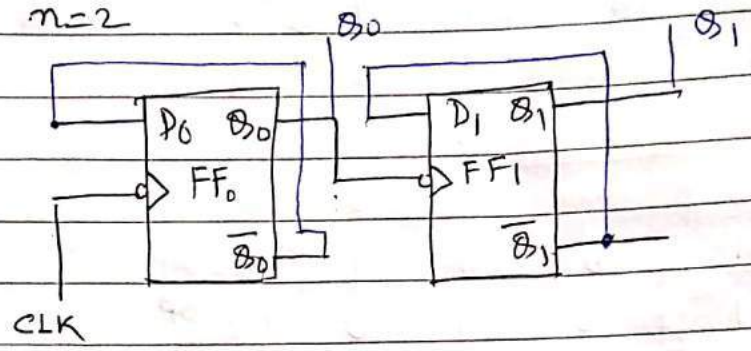
Ripple down
MOD-16 counter

$f_3 = f/16$.

• Ripple Counter using D-FF =

→ FF should be in toggle mode.

• 2-bit Ripple Counter =



Timing Diagram =

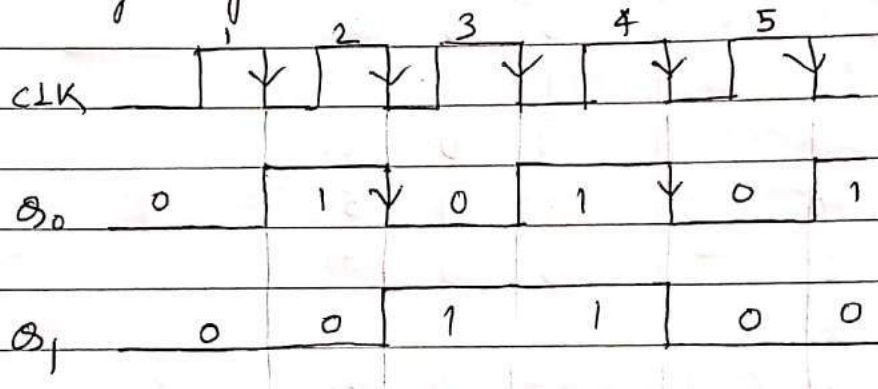


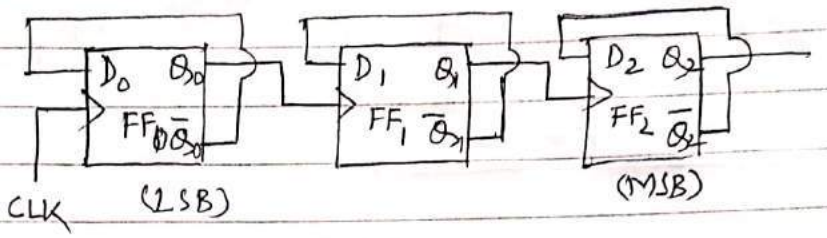
Table –

CLK	Q_1	Q_0	$\overline{Q_1}$	$\overline{Q_0}$
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
↓ 5	0	1	1	0

~~Up Cou~~

MOD-4 Ripple Up counter.

• 3-bit Ripple Down counter using +ve edge triggered D-FF =



Table

421

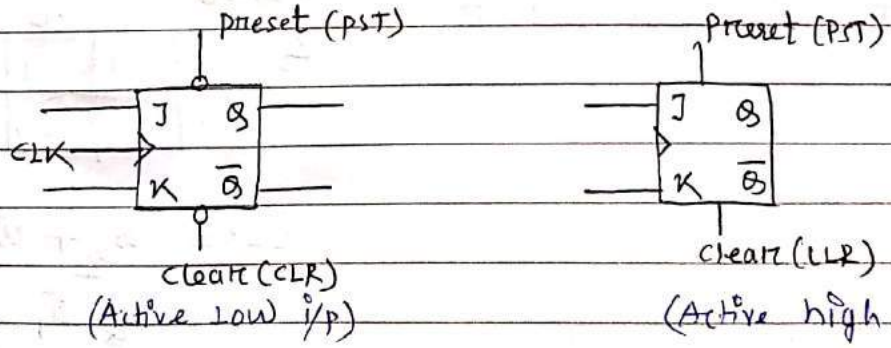
CLK	Q_2	Q_1	Q_0	$\bar{Q}_2$	$\bar{Q}_1$	$\bar{Q}_0$
→ 0	0	0	0 - 0	1	1	1
↑ 1	1	1	1 - 7	0	0	0
↑ 2	0	1	0 - 2	1	0	1
↑ 3	0	0	1 - 1	1	1	0
↑ 4	0	0	0 - 0	1	1	1
↑ 5	1	1	1 - 7	0	0	0
↑ 6	0	1	0 - 2	1	0	1
↑ 7	0	0	1 - 1	1	1	0
↑ 8	0	0	0 - 0	1	1	1

7 → 2 → 1 → 0

(MOD - 8 Ripple Down counter)

• MOD - N Ripple Counter =
↓
modulus

JK-FF with preset & clear inpts =



active low i/p

PST	CLR	Q
1	0	0
0	1	1
1	1	FF works normally
0	0	X

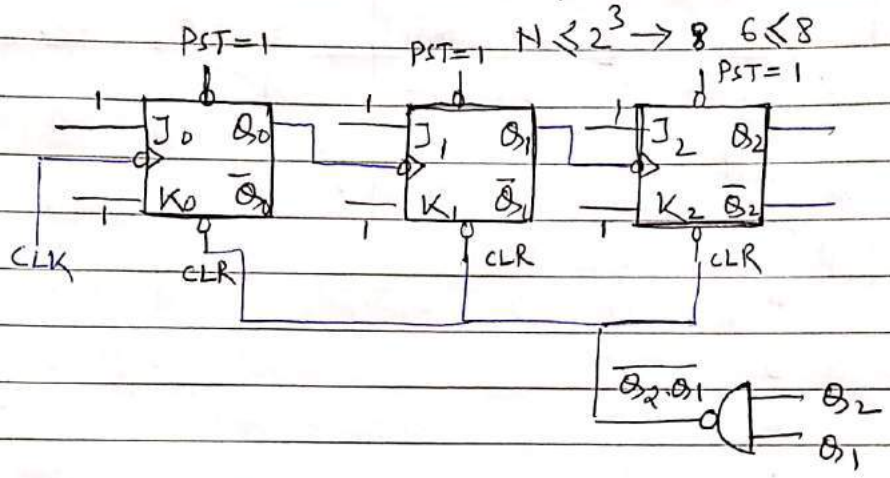
active high i/p

PST	CLR	Q
1	0	1
0	1	0
1	1	X
0	0	FF works normally

- **MOD-6 Ripple counter =**

$N \leq 2^n \rightarrow$ no of flip flops.

$N \leq 2^3 \rightarrow 8 \quad 6 \leq 8$



Table

CLK	Q_2	Q_1	Q_0	
↓ 0	0	0	0	→ 0
↓ 1	0	0	1	→ 1
↓ 2	0	1	0	→ 2
↓ 3	0	1	1	→ 3
↓ 4	1	0	0	→ 4
↓ 5	1	0	1	→ 5
↓ 6	1	1	0	→ 6
↓ 7	1	1	1	→ 7

000 } CLR = 1
101 }

110 } CLR = 0
111 }

CLR

Q_2 \ Q_1Q_0	00	01	11	10
0	1	1	1	1
1	1	1	X	0

$CLR = \bar{Q}_2 + \bar{Q}_1$

$= \overline{Q_2 \cdot Q_1}$

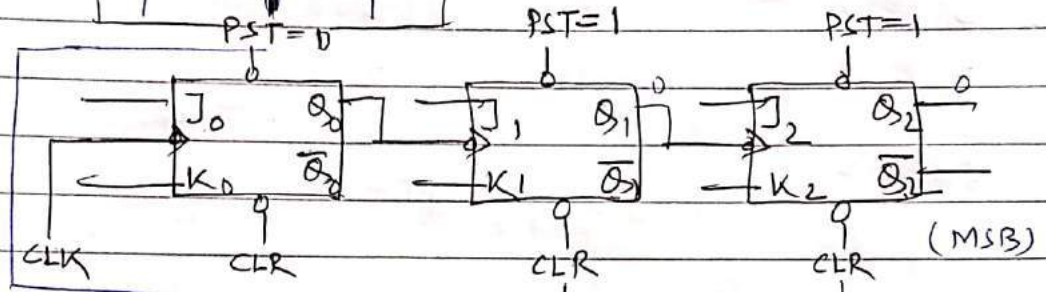
Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Count from 001 to 100.

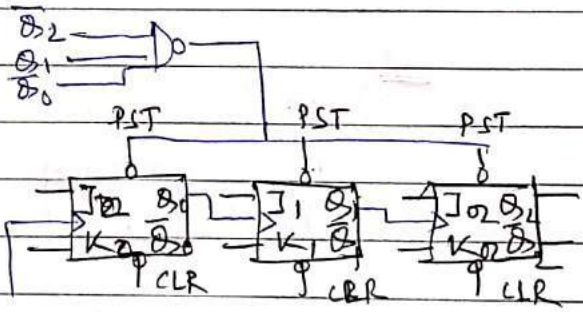
count

$1 \rightarrow 2 \rightarrow 3$

MOD – 3 Up Counter



CLK	Q_2	Q_1	Q_0
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0
8	1	1	1



count $7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow$

MOD – 5 Down counter.

MOD-9 Ripple counter

→ UP counter $N \leq 2^n$

$9 \leq 2^n = 2^4$

4-FFs

CLK	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
⋮	⋮	⋮	⋮	⋮
15	1	1	1	1

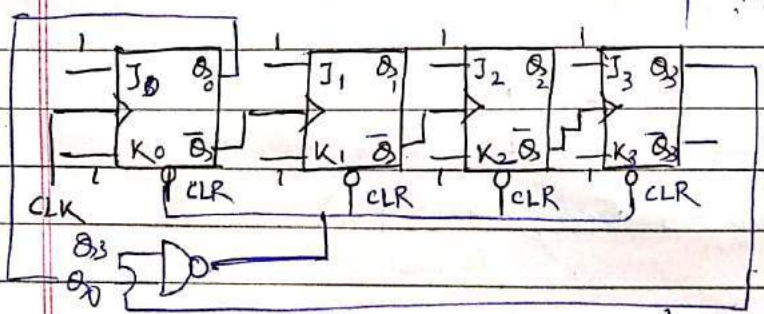
→ 0000

CLR

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1_0	1_1	1_3	1_2
01	1_4	1_5	1_7	1_6
11	X_{12}	X_{13}	X_{15}	X_{14}
10	1_8	0_9	X_{11}	X_{10}

$CLR = \overline{Q_3} + \overline{Q_0}$

$= \overline{Q_3 Q_0}$



MOD-10 Ripple counter

Down counter

$10 \leq 2^n = 2^4$

4-FFs

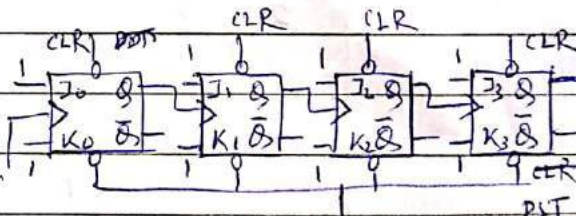
CLK	Q_3	Q_2	Q_1	Q_0
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
10	0	1	0	1
⋮	⋮	⋮	⋮	⋮
15	0	0	0	0

15 14 13 ⋮ 7 6 5 1 0

CLR

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	X_0	X	X	X
01	X	0	1	1
11	1	1	1	1
10	1	1	1	1

$CLR = Q_3 + Q_1 = \overline{\overline{Q_3}\,\overline{Q_1}}$



$\overline{Q_3}$
$\overline{Q_1}$

(Decade counter)

or

(Divide by 10 counter)

- **BCD Ripple Counter** — (alway up counting)

0000 ↓ 1001

up
MOD-10 Ripple counter
(or)
Decade Counter
(or)
Divisible by 10 Ripple counter.
(or)
BCD Ripple counter.

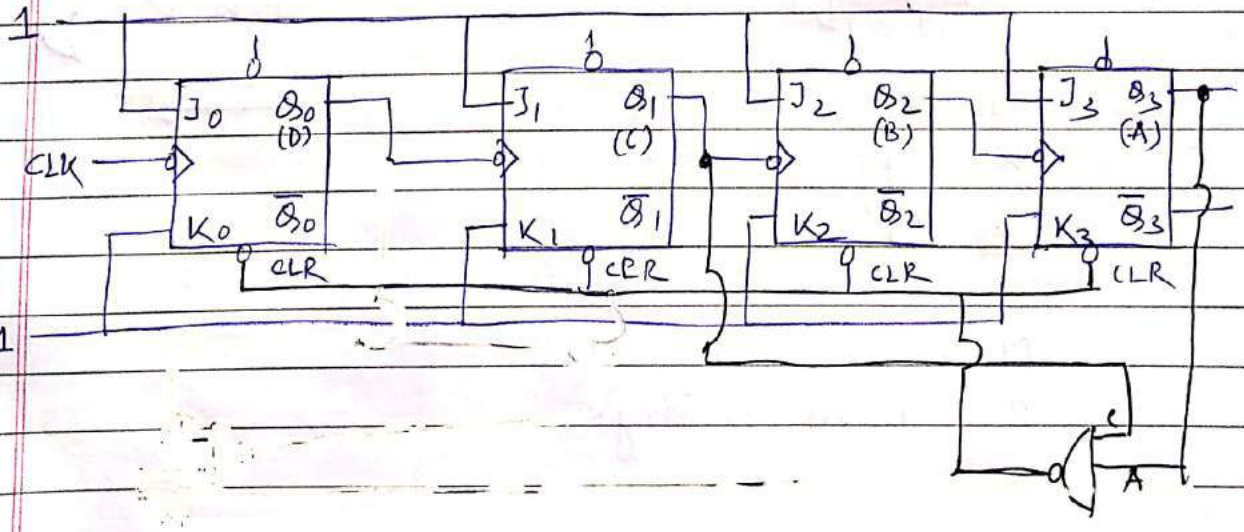
CLK	A	B	C	D	CLR
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
⋮	⋮	⋮	⋮	⋮	

CLR

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	X	X	X

$CLR = \overline{A} + \overline{B}$

$= \overline{AB}$.



• UP/Down Ripple Counter =

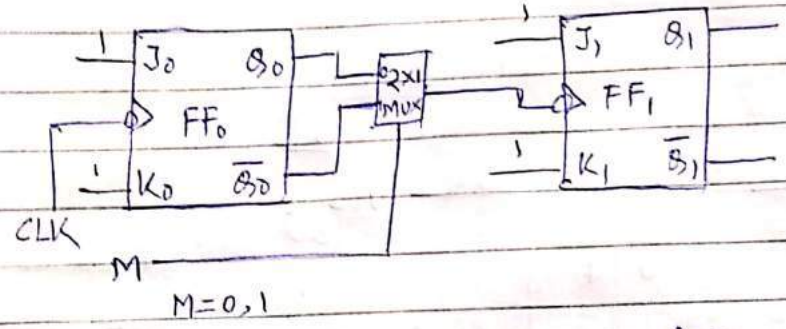
① 2-bit UP/Down Ripple counter =

$n = 2 \rightarrow N \leq 2^n$

$N = 4$

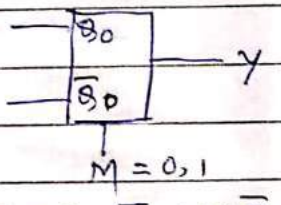
$0 \rightarrow 1 \rightarrow 2 \rightarrow 3$

$3 \rightarrow 2 \rightarrow 1 \rightarrow 0$



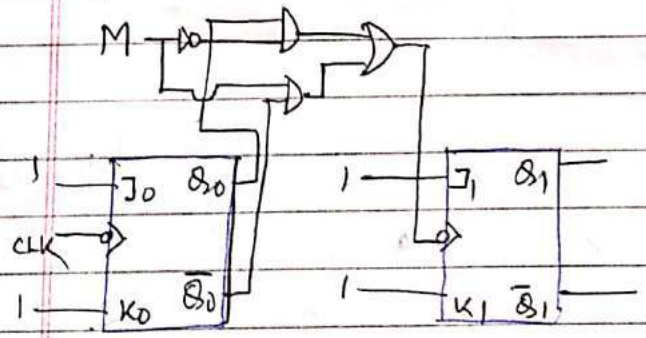
When, M = 1, ~~Q0 will be~~ Output of Q_0 will be the clock of second flip-flop. (UP counting)

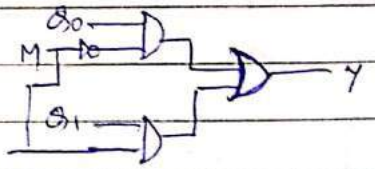
M = 0, output of $\bar{Q}_0$ will be the clock of second FF. (Down counting)



$Y = Q_0 \bar{M} + M \bar{Q}_0$

(Same function)



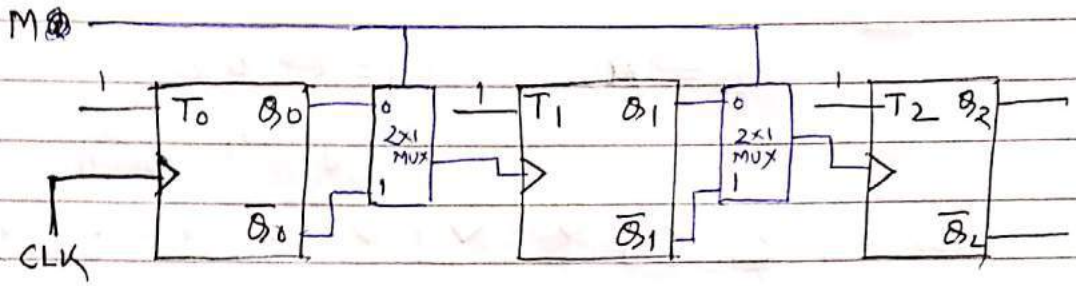


M = 0, UP counting.

M = 1, Down counting.

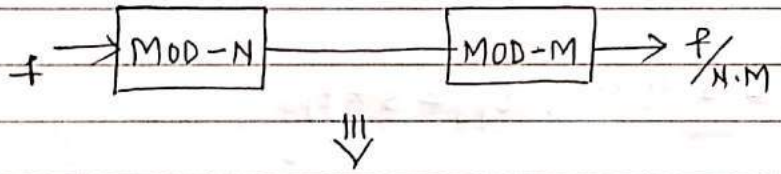
(11) 3-bit Ripple Up/Down Counters

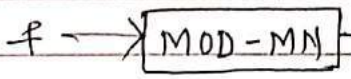
$n=3$, $N = 2^3 = 8$



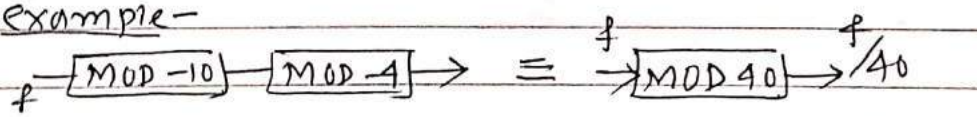
When, M = 0 (Down counting)
M = 1 (Up counting).

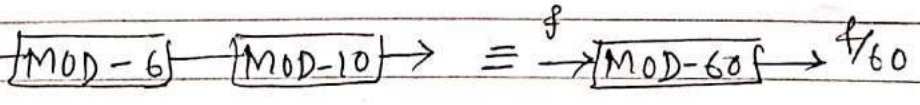
- **Cascading of Ripple Counters**





Example-

(i) 

(ii) 

- **Drawback of Ripple counter or Asynchronous Counter**

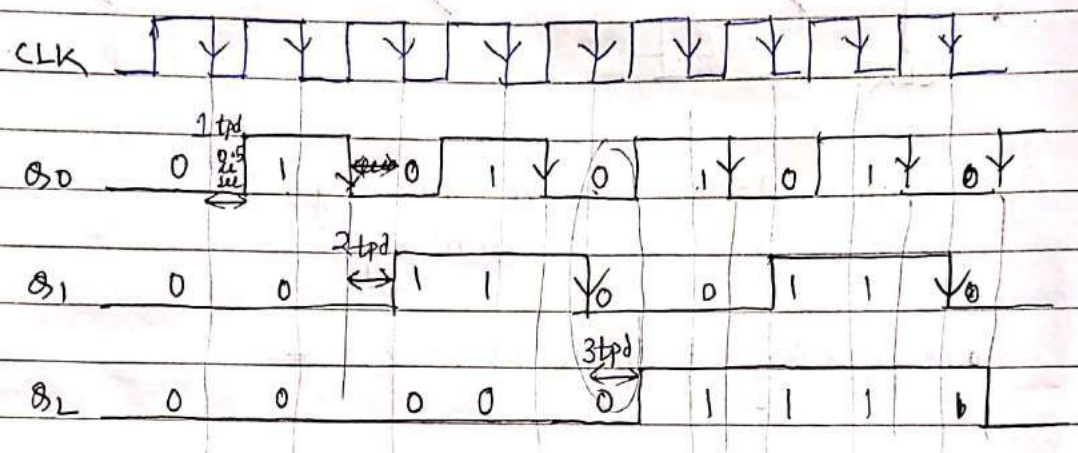
→ low speed counters.
If no of FF - n then, Max ~~Prop~~ ~~tpd~~ delay = $n t_{pd}$.
t_{pd} ⇒ propagation delay of '1' FF.

→ If the propagation delay is large compare to clock period, then counter may skip a state.

Ex—

$T_{CLK} = 1\ \mu sec$ $\quad t_{pd} = 0.5\ \mu sec$

↓ propagation delay.



Ex—

$T_{CLK} = 1\ \mu sec$, $t_{pd} = 3 * t_{pd}$

$= 3 * 0.5$

$= 1.5$

$\boxed{T_{CLK} < t_{pd}}$

CLK	Q_2	Q_1	Q_0	
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	→ skiped
5	1	0	1	
6	1	1	0	
7	1	1	1	

** for avoid this problem —

t_{pd} = pro delay of FF

n = no. of FF.

$\boxed{T_{CLK} \geq n.t_{pd}}$

$\frac{1}{f_{CLK}} \geq n.t_{pd} \Rightarrow f_{CLK} \leq \frac{1}{n.t_{pd}}$.

• Synchronous Counters :

• Required steps to design Synchronous Counter —

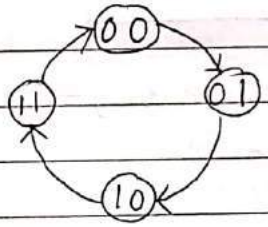
→ No. of FFs required. — $n=2,\ N \leq 2^n = 4$
→ state diagram. — $0 \to 1 \to 2 \to 3$
→ choice of FF & excitation table of FF.
→ minimal expressions for the excitations.
→ Draw the logic diagram

• Design of a Synchronous 2-bit UP Counter —

$n = 2$ (no. of FF) $\quad N \leq 2^n$

$N \leq 4 \quad 0 \to 1 \to 2 \to 3$

(i) no. of FF required = 2.

(ii) 

state diagram.

(iii) D-FF

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

(iv)

CLK	PS $Q_1 Q_0$	NS $Q_1 Q_0$	D_1	D_0	required excitation
1	0 0	0 1	0	1	
2	0 1	1 0	1	0	
3	1 0	1 1	1	1	
4	1 1	0 0	0	0	

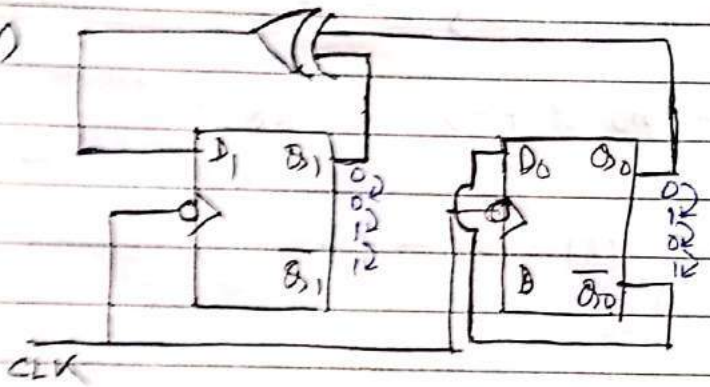
$D_1 = Q_1\bar{Q}_0 + Q_0\bar{Q}_1 = Q_0 \oplus Q_1$

Q_1 \ Q_0	0	1
0	0	1
1	1	0

$D_2 = \bar{Q}_0$

Q_1 \ Q_0	0	1
0	1	0
1	1	0

(i)



total propagation delay = t_{pd}

	FS Q_1 Q_0	NS Q_1 Q_0
CLK 1	0 0	0 1
CLK 2	0 1	1 0
CLK 3	1 0	1 1
CLK 4	1 1	0 0

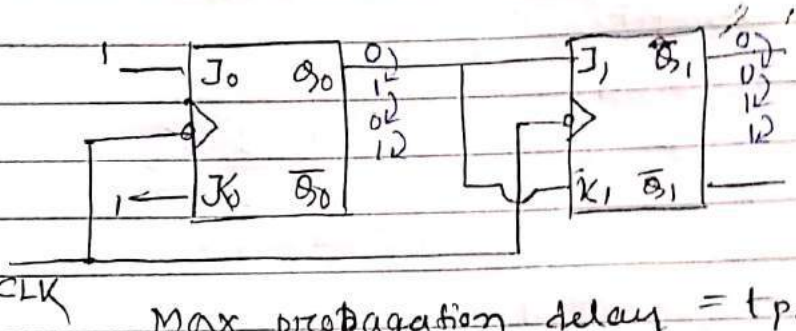
Step-(5):

Draw the logic diagram -

$J_0 = K_0 = 1$

$J_1 = K_1 = Q_0$

CLK → -ve edge triggering.



Max propagation delay = t_{pd}

↓

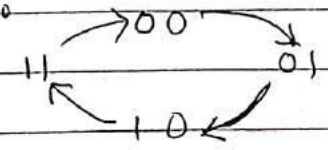
(prop delay of 1-FF)

	P.S Q_1 Q_0	N.S Q_1 Q_0
↓ CLK1	0 0	0 1
↓ CLK2	0 1	1 0
↓ CLK3	1 0	1 1
↓ CLK4	1 1	0 0

- 2-bit Synchronous Up Counter using T-FF :

Step-(1): $n = 2$ $\quad 2^n = 4$

Step-(2):



Step-(3): T-FF

excitation table of T-FF

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

Step-(4):

	P.S		N.S		Required excitation	
CLK	Q_0	Q_1	Q_0	Q_1	T_0	T_1
1	0	0	0	1	0	1
2	0	1	1	0	1	1
3	1	0	1	1	0	1
4	1	1	0	0	1	1

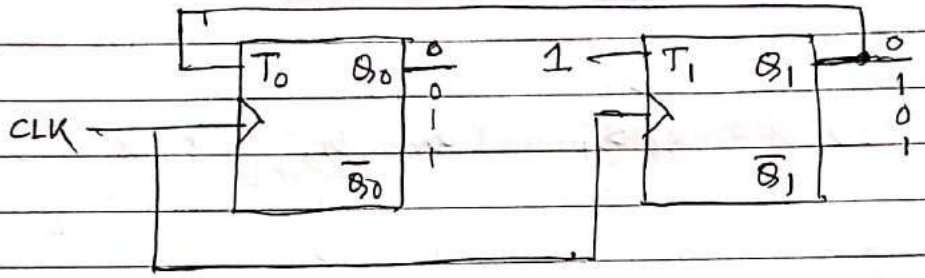
T_0

Q_0 \ Q_1	0	1
0	0	1
1	0	1

$T_1 = 1$

$T_0 = Q_1$

Step-(5): Draw the Diagram –

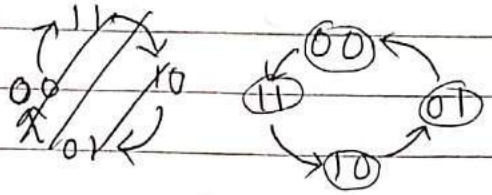


	P.S		N.S	
CLK	Q_0	Q_1	Q_0	Q_1
1	0	0	0	1
2	0	1	1	0
3	1	0	1	1
4	1	1	0	0

• 2-bit Synchronous Down Counter —

(1) $n = 2$

(2)



(3) JK-FF

excitation table.

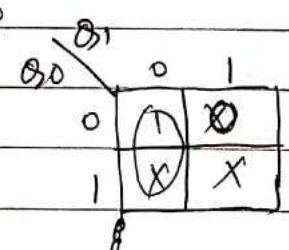
Q_n	Q_{n+1}	Q_J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(4)

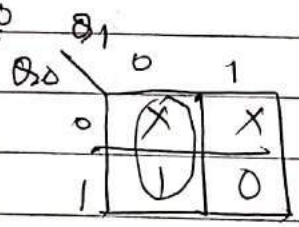
	P.S		N.S					
CLK	Q_0	Q_1	Q_0	Q_1	J_0	K_0	J_1	K_1
1	0	0	1	1	1	X	1	X
2	1	1	1	0	0	X	X	1
3	1	0	0	1	X	1	1	X
4	0	1	0	0	X	0	X	1

(5) ~~Diagram~~ +

J_0



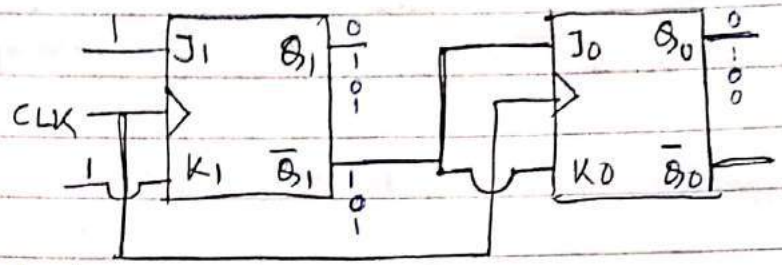
K_0



$J_0 = \overline{Q_1}$ $\quad$ $K_0 = \overline{Q_1}$

$J_1 = 1$ $\quad$ $K_1 = 1$

(5)



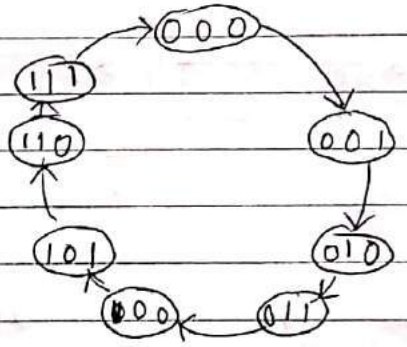
Verification =

	P.S		N.S	
CLK	Q_0	Q_1	Q_0	Q_1
1	0	0	1	1
2	1	1	1	0
3	1	0	0	1
4	0	1	0	0

- 3-bit synchronous up counter
  or
  MOD-8 Synchronous up counter.

(1) $n = 3(FF), N = 2^3 = 8$ (state)

(2) state diagram



(3) choice FF & excitation table = (T-FF)

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

(4) Minimal expressions for excitations —

	P.S			N.S			Required excitation		
CLK	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
1	0	0	0	0	0	1	0	0	1
2	0	0	1	0	1	0	0	1	1
3	0	1	0	0	1	1	0	0	1
4	0	1	1	1	0	0	1	1	1
5	1	0	0	1	0	1	0	0	1
6	1	0	1	1	1	0	0	1	1
7	1	1	0	1	1	1	0	0	1
8	1	1	1	0	0	0	1	1	1

T_2

Q_2 \ Q_1Q_0	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$T_2 = Q_1 Q_0$

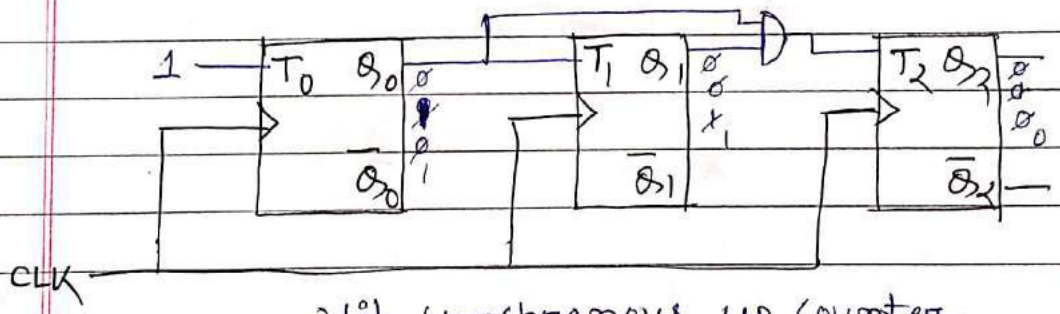
T_1

Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$T_1 = Q_0$

$T_0 = 1$

(5) Draw Diagram —



3 bit synchronous up counter.

Verification

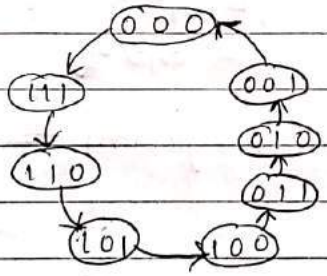
CLK	Q_2 Q_1 Q_0	Q_2 Q_1 Q_0
1	0 0 0	0 0 1
2	0 0 1	0 1 0
3	0 1 0	0 1 1
4	0 1 1	1 0 0
5	1 0 0	1 0 1
6	1 0 1	1 1 0
7	1 1 0	1 1 1
8	1 1 1	0 0 0

- **3-bit synchronous Down Counter =**
  (or)
  MOD-8 synchronous Down Counter =

(1) $n = 3$, $N = 2^3 = 8$

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

(2)



8 → 7 → 6 → 5 → 4 → 3 → 2 → 1

(3) Choice flipflop & excitation table –
D-FF

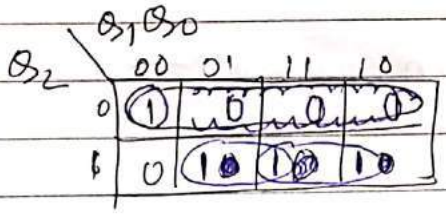
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

④ minimal expressions for excitations =

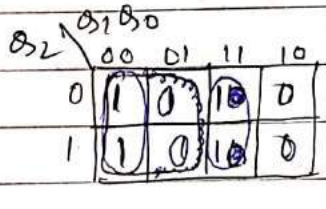
CLK	$Q_2\ Q_1\ Q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
1	0 0 0	1 1 1	1 1 1
2	1 1 1	1 1 0	1 1 0
3	1 1 0	1 0 1	1 0 1
4	1 0 1	1 0 0	1 0 0
5	1 0 0	0 1 1	0 1 1
6	0 1 1	0 1 0	0 1 0
7	0 1 0	0 0 1	0 0 1
8	0 0 1	0 0 0	0 0 0

* Careful when design K-map in case of down counting

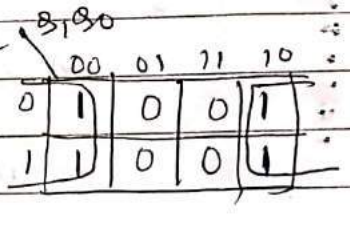
D_2



D_1



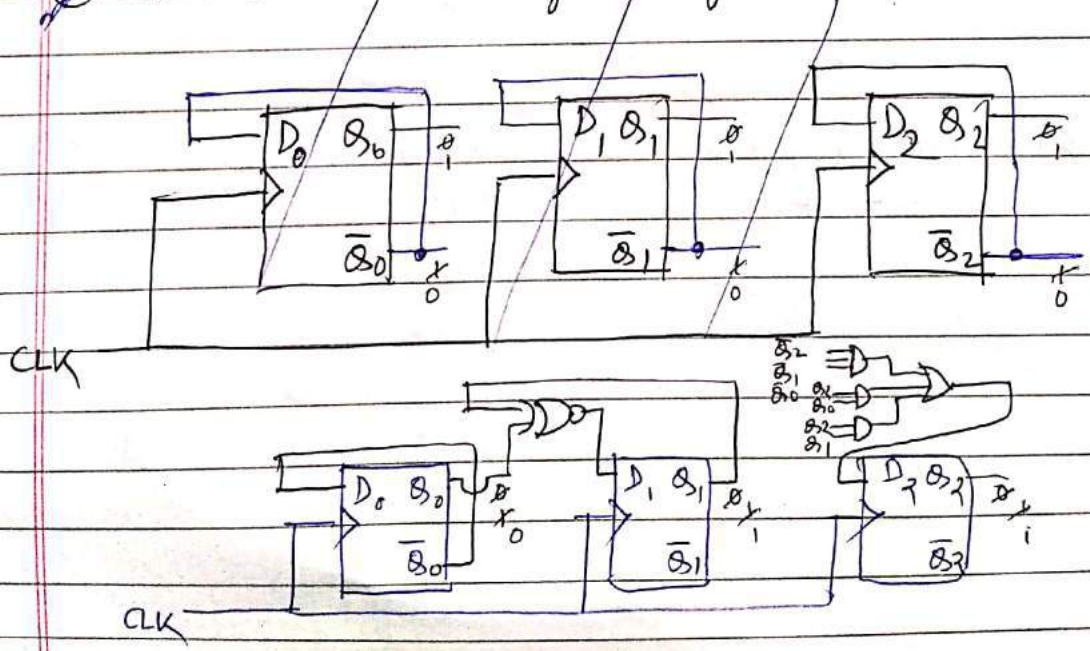
D_0



$D_2 = \bar{Q}_2 Q_1 \bar{Q}_0 + Q_2 Q_0 + Q_2 Q_1$ $D_1 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0$ $D_0 = \bar{Q}_0$

$= Q_1 \odot Q_0$

⑤ Draw the Diag Logic Diagram =



verification -

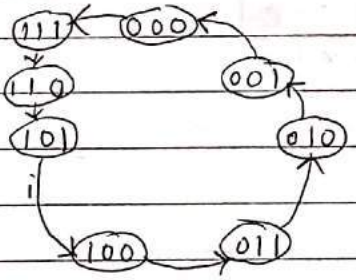
CLK	$Q_2 Q_1 Q_0$	$Q_2 Q_1 Q_0$
1	0 0 0	1 1 1
2	1 1 1	1 1 0
3	1 1 0	1 0 1
4	1 0 1	1 0 0
5	1 0 0	0 1 1
6	0 1 1	0 1 0
7	0 1 0	0 0 1
8	0 0 1	0 0 0

- 3-bit Synchronous Down counter using JK-FF :

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

(I) $n = 3$ $N = 2^3 = 8$ (MOD -8 synchronous down counter.

(II) State diagram -



(III) JK-FF & excitation table.

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

(iv)

CLK	Q_2	Q_1	Q_0	Q_2'	Q_1'	Q_0'	J_2	K_2	J_1	K_1	J_0	K_0
1	0	0	0	1	1	1	1	X	1	X	1	X
2	1	1	1	1	1	0	X	0	X	0	X	01
3	1	1	0	1	0	1	X	0	X	10	1	X
4	1	0	1	1	0	0	X	0	0	X	X	01
5	1	0	0	0	1	1	X	01	1	X	1	X
6	0	1	1	0	1	0	0	X	X	0	X	01
7	0	1	0	0	0	1	0	X	X	10	1	X
8	0	0	1	0	0	0	0	X	0	X	X	1

$J_0 = K_0 = 1$

J_1

Q_2 \ Q_1Q_0	00	01	11	10
0	1	0	X	X
1	1	0	X	X

$J_1 = \bar{Q}_0$

K_1

Q_2 \ Q_1Q_0	00	01	11	10
0	X	X	0	1
1	X	X	0	1

$K_1 = \bar{Q}_0$

J_2

Q_2 \ Q_1Q_0	00	01	11	10
0	1	0	0	0
1	X	X	X	X

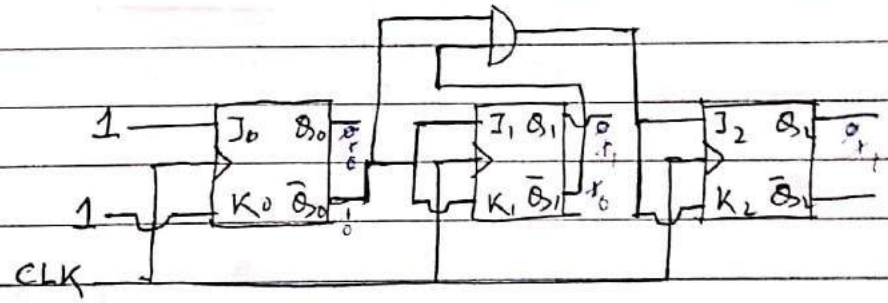
$J_2 = \bar{Q}_1\bar{Q}_0$

K_2

Q_2 \ Q_1Q_0	00	01	11	10
0	X	X	X	X
1	1	0	0	0

$K_2 = \bar{Q}_1\bar{Q}_0$

(v) Logic Diagram =



→ We use JK-FF and T-FF ~~for~~ to get simple circuit.

(Work)

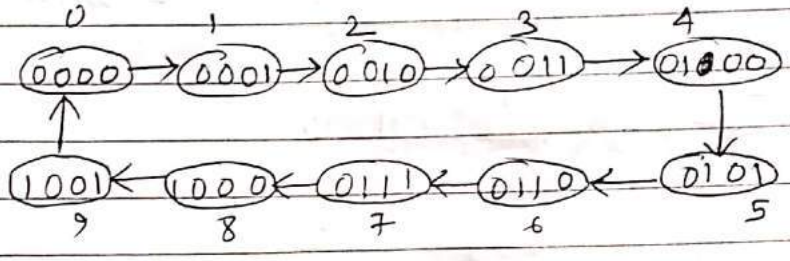
- Design of a MODULUS-10 synchronous up counter =

(I) $n = 4$, $N = 2^4 = 16$

$N \leq 2^n$

$10 \leq 2^4$

(II)



Decade counter

(or) BCD Counter.

(III) T-FF & excitation table =

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

(IV)

	P.S				N.S							
CLK	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0
1	0	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	1	0	0	1	0	0	0	1	1
3	0	0	1	0	0	0	1	1	0	0	0	1
4	0	0	1	1	0	1	0	0	0	1	1	1
5	0	1	0	0	0	1	0	1	0	0	0	1
6	0	1	0	1	0	1	1	0	0	0	1	1
7	0	1	1	0	0	1	1	1	0	0	0	1
8	0	1	1	1	1	0	0	0	1	1	1	1
9	1	0	0	0	1	0	0	1	0	0	0	1
10	1	0	0	1	0	0	0	0	1	0	0	1

$T_0 = 1$

$T_1 =$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	1	1	0
01	0	1	1	0
10	X	X	X	X
11	0	0	X	X

$T_1 = \bar{Q}_3 Q_0$

$T_2 =$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	1	0
01	0	0	1	0
10	X	X	X	X
10	0	0	X	X

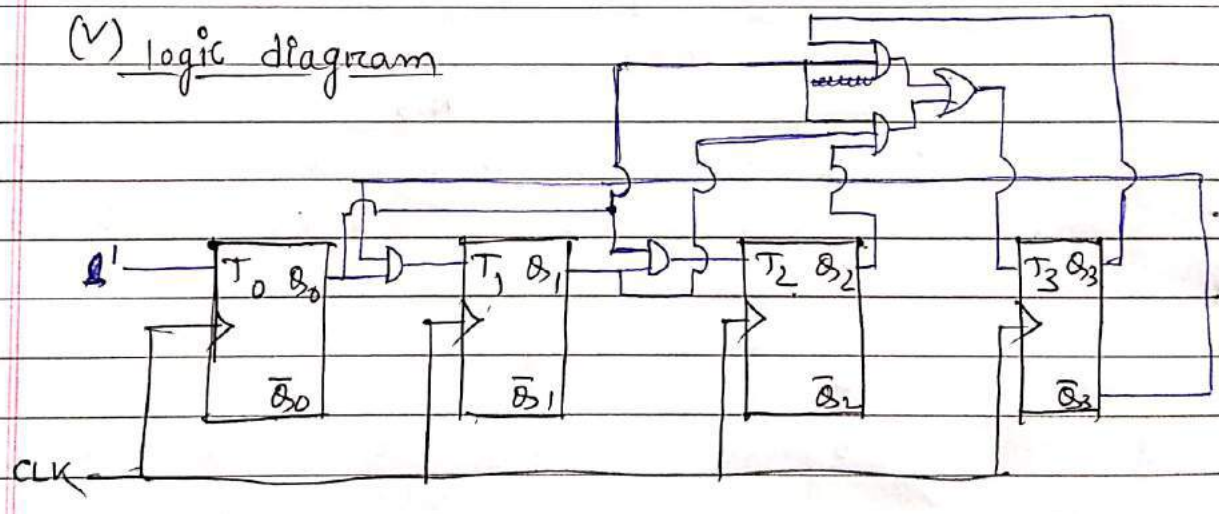
$T_2 = Q_1 Q_0$

$T_3 =$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	1	X	X

$T_3 = Q_2 Q_1 Q_0 + Q_3 Q_0$

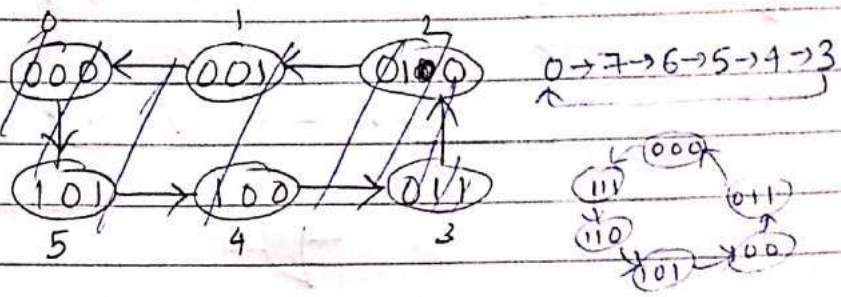
(V) Logic diagram



• Design of a Synchronous MOD-6 Down Counter —

(1) $N \leq 2^n$, $n = 3$, $6 \leq 2^3$

(2)



(3) JK-FF & its excitation table —

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

(4)

CLK	P.S Q_2	Q_1	Q_0	N.S Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
1	0	0	0	1	0	1	1	X	0	X	1	X
2	1	0	1	1	0	0	X	0	0	X	X	1
3	1	0	0	0	0	1	X	0	X	0	1	X
4	0	0	1	0	0	0	0	X	0	X	X	1
5	0	0	0	0	1	1	0	X	1	X	1	X
6	0	1	1	0	0	0	0	X	X	1	X	1

$J_0 = K_0 = 1$

$J_1 =$

Q_2 \ Q_1Q_0	00	01	11	10
0	1	X	X	X
1	1	0	X	X

$J_1 = Q_1 Q_0$

$K_1 =$

Q_2 \ Q_1Q_0	00	01	11	10
0	X	X	1	X
1	X	X	0	1

$K_1 = Q_1 Q_0 \bar{Q}_2 + \bar{Q}_0$

$J_2 =$

Q_2 \ Q_1Q_0	00	01	11	10
0	1	X	0	X
1	X	X	X	X

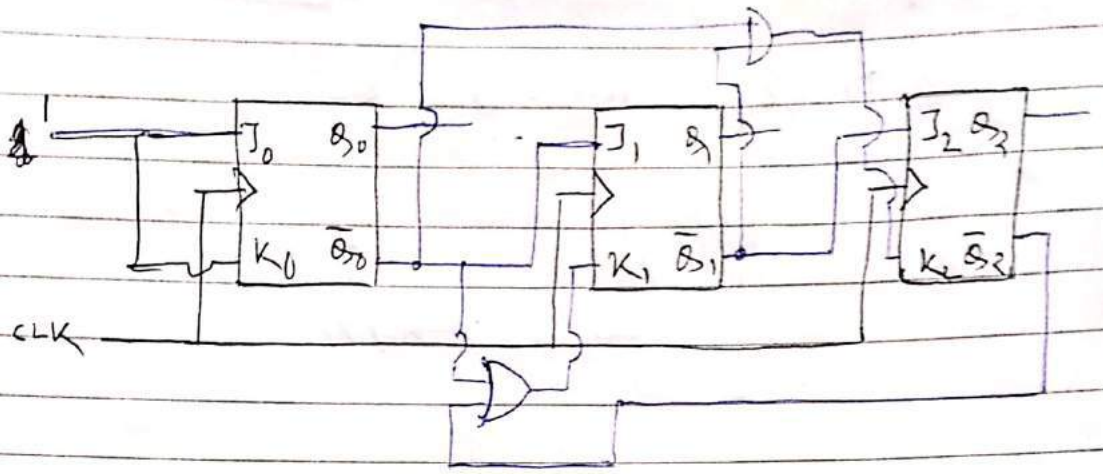
$J_2 = Q_1 Q_0 \bar{Q}_1 \bar{Q}_0$

$K_2 =$

Q_2 \ Q_1Q_0	00	01	11	10
0	X	X	0	X
1	1	0	X	0

$K_2 = Q_1 Q_0 \bar{Q}_1 \bar{Q}_0$

(5)



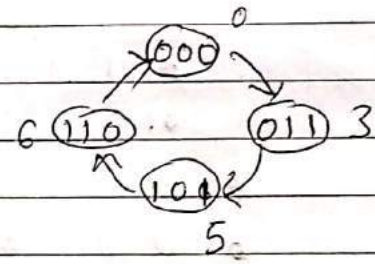
MOD-6 synchronous Down counter.

→ MOD-6 ~~up count~~ synchronous up counter is self starting.

• Design a counter that goes through states 0, 3, 5, 6, 0.

(I) no. of FFS $n = 3$.

(II)



1, 2, 4, 7

(III) Choice of FF & excitation flipflop –

T-FF

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

CLK	P.S. $Q_2\ Q_1\ Q_0$	N.S $Q_2\ Q_1\ Q_0$	required excitation $T_2\ T_1\ T_0$
1	0 0 0	0 1 1	0 1 1
2	0 1 1	1 0 1	1 1 0
3	1 0 1	1 1 0	0 1 1
4	1 1 0	0 0 0	1 1 0

(iv) minimal expression =

$T_2 = Q_1$

Q_2 \ Q_1Q_0	00	01	11	10
0	0	X	1	X
1	X	0	X	1

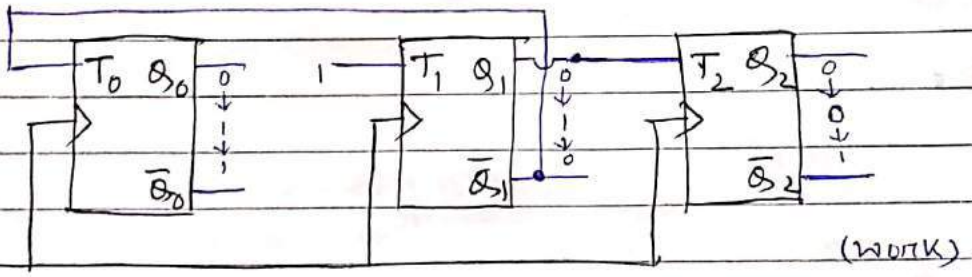
$T_1 = 1$

Q_2 \ Q_1Q_0	00	01	11	10
0	1	X	1	X
1	X	1	X	1

$T_0 = \overline{Q_1}$

Q_2 \ Q_1Q_0	00	01	11	10
0	1	X	0	X
1	X	1	X	0

(v)



(WORK)

(1,2,4,7)

CLK	P.S $Q_2\ Q_1\ Q_0$	N.S $Q_2\ Q_1\ Q_0$	$Q_1 = 1$: T_2	T_1	$\overline{Q_1}$: T_0
1	(1) 0 0 1	0 1 0 (2)	0	1	1
2	0 1 0	0 0 0 (4)	1	1	0
3	1 0 0	1 1 1 (7)	0	1	1
4	1 1 1	0 0 1 (1)	1	1	0

$1 \rightarrow 2 \rightarrow 4 \rightarrow 7$ → Lock out problem

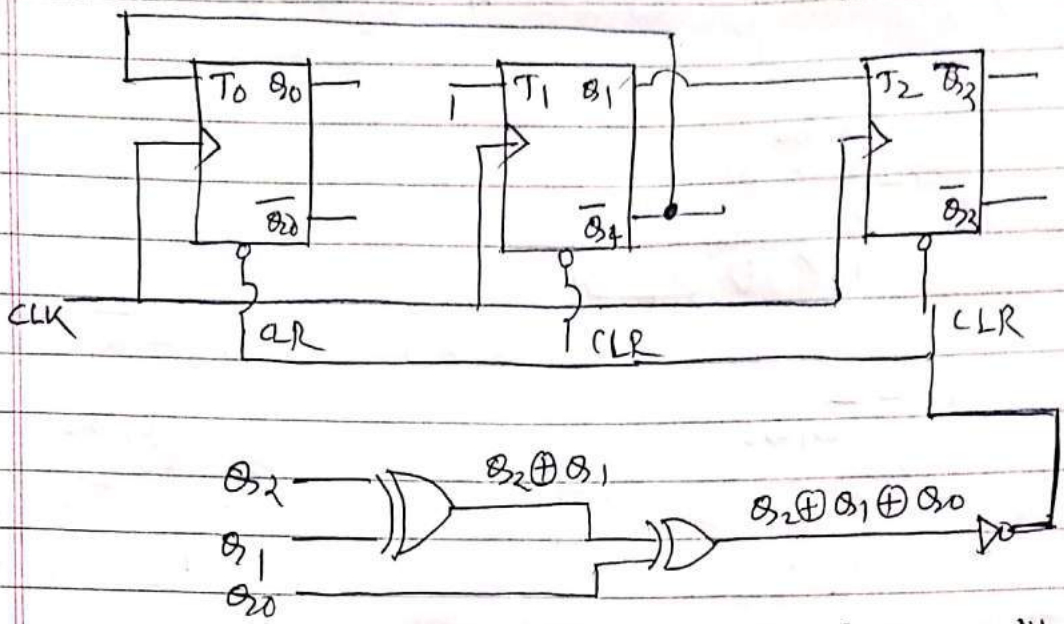
to solve this problem =

$Q_2\ Q_1\ Q_0$	CLR
(1) 0 0 1	1
(2) 0 1 0	1
(4) 1 0 0	1
(7) 1 1 1	1

CLR =

Q_2 \ Q_1Q_0	00	01	11	10
0	⊗	(1)	0	(1)
1	(1)	0	(1)	0

$\Sigma(1,2,4,7) = Q_2 \oplus Q_1 \oplus Q_0$

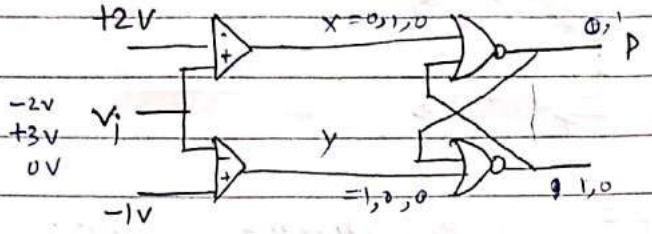


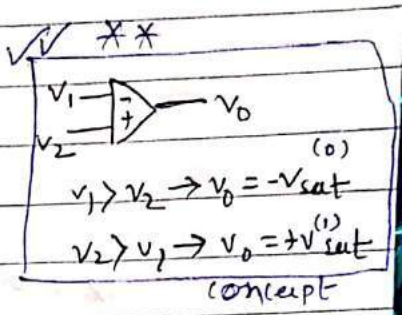
When (1, 2, 4, 7) any of them will come Counter will be CLR. (000).

Scanned by CamScanner

GATE QUESTION ON SEQUENTIAL CIRCUIT :

Gr-1987 (Q1)





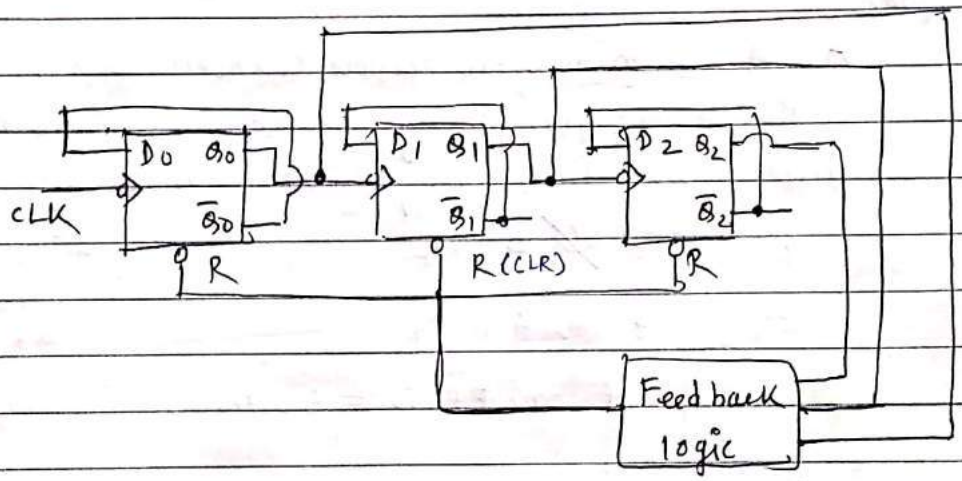
(I) For $V_i = -2V$, P=0 ($x = 0, y = 1$)

(II) For $V_i = +3V$, P=0 ($x = 1, y = 0$)

(III) For $V_i = 0V$, P=0 always ($x = 0, y = 0$)

(IV) For $V_i = 0V$, P can be either '0' or '1'.

Gr-1987 (Q2)



A Ripple counter using -ve triggered D-FF is shown. The FFs are cleared to '0' at R i/p. The Feedback logic is to be designed to obtain the count sequence shown below.

(a) 000 → 001 → 010 → 011 → 100 → 101

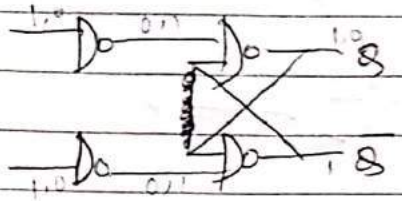
→ MOD-6 Asynchronous counter.

(6)
110
$Q_2 Q_1 Q_0$

$\overline{Q_2 Q_1 \bar{Q}_0}$

→ Feed back logic is = $\overline{Q_2 Q_1 \bar{Q}_0}$.

G=1988 (Q3) The circuit shown below is a



→ SR (latch)

~~(a)~~ X (a) JK-FF X (b) Jonson's counter

X (c) RS-FF (d) None of the above.

→ ** [it is not FF because it have not any clock]
[if it is not FF, then it can't be counter]

G=1990 (Q4)

A 4-bit modulo-16 ripple counter uses a JK-FF. If the propagation delay of each FF is 50 ns. The max clock frequency that can be used is equal to:

(a) 20 MHz (c) 5 MHz

(b) 10 MHz (d) 4 MHz

→ t_{pd} of one FF is = 50 ns

** $T_{clk} \geq n\, t_{pd}$

** $f = \frac{1}{T_{clk}}$

$= \frac{1}{4 \times 50 \times 10^{-9}}$

$= \frac{10^9}{200} = \frac{10^7}{2}$

$= 5 \times 10^6$

$= 5$ MHz

G=1991 (Q5)

An S-R-FF can be connected in a T-FF by connecting to R & S to S̄.

S·R → T

Q_n	T	Q_{n+1}	S	R
0	0	0	0	X
0	1	1	1	0
1	0	1	X	0
1	1	0	0	1

S \ T	0	1
0	0	1
1	X	0

$S = \bar{S} \cdot T$

S \ T	0	1
0	X	0
1	0	1

$R = S \cdot T$

or 1992

(Q6) A pulse train with a freq of 1 MHz is counted using a modulo-1024 ripple counter built with JK-FF. For proper operation of the counter. The max permissible prop delay per FF stage is 100 nsec.

→ max t_{pd} = ?

$T_{CLK} = n \times t_{pd}$

$t_{pd} = \frac{T_{CLK}}{n}$

$= \frac{10^{-6}}{10}$

$= 10^{-7}$ sec

$= 100 \times 10^{-9}$ sec

$= 100$ nsec

$f_{req} = 1$ MHz

$T_{CLK} = \frac{1}{f_{CLK}}$

$= \frac{1}{10^6}$

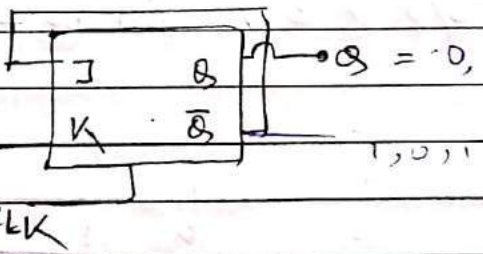
$= 10^{-6}$ sec

MOD-1024 ripple (N)

$N = 2^{10} = 1024$

$n = 10$

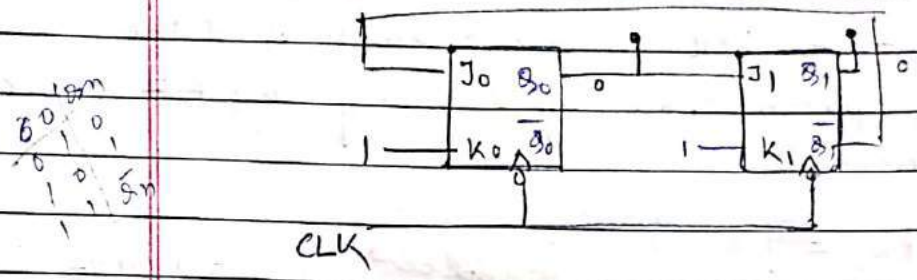
(Q7) In a JK-FF we have $J = \bar{Q}$, $K = 1$. Assuming the FF was initialy cleared & then clocked for 6 pulses, sequence at the Q o/p will be



→

CLK	Q
1	1
2	0
3	1
4	0
5	1
6	0

GATE-1998 (Q8) Fig shows a Mod-K counter, Here K = ?



(a) 1 (b) 2 (c) 3 (d) 4

→ $n = 2$

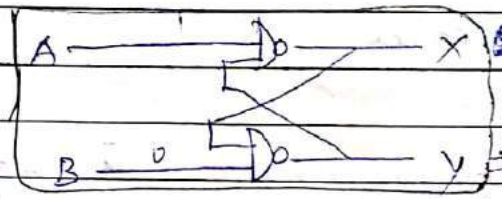
$N \leq 2^n$

$N \leq 2^2$

$N \leq 4$

CLK	P.S Q_1 Q_0	N.S Q_1 Q_0	J_1 K_1	J_0 K_0
1	0 0	0 1	0 1	1 1
2	0 1	1 0	1 1	1 1
3	1 0	0 0	0 1	0 1
4				

GATE-1998 (Q9)



A = 1, B = 1, the i/p 'B' is now replaced by a sequence 101010.

The o/p x & y will be –

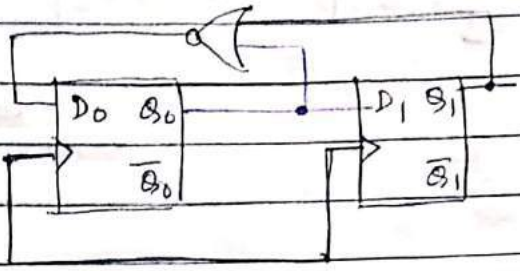
→

A	B	X	Y
1	1	0	1
1	0	0	1
1	1	0	1
1	0	0	1
1	1	0	1
1	0	0	1

X = 0
Y = 1

q → 2007

(Q.10) For the circuit shown, the counter state $(Q_1 Q_0)$ follows the sequence –



(a) 00, 01, 10, 11, 00.

(b) 00, 01, 10, 00, 01.

(c) 00, 01, 11, 00, 01.

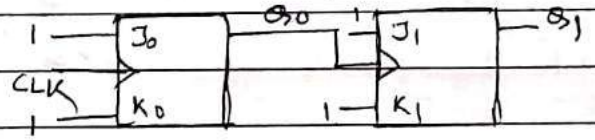
(d) 00, 10, 11, 00, 10.

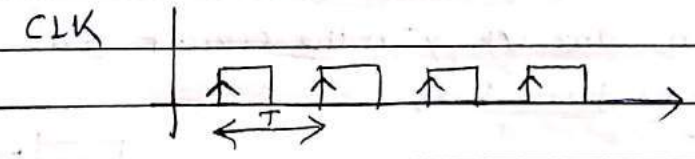
→

CLK	P.S Q_1	P.S Q_0	N.S Q_1	N.S Q_0	D_1	D_0
					Q_0	$\overline{Q_1 + Q_0}$
1	0	0	0	1	0	1
2	0	1	1	0	1	0
3	1	0	0	0	0	0
4	0	0	0	1	0	1
5	0	1	1	0	1	0

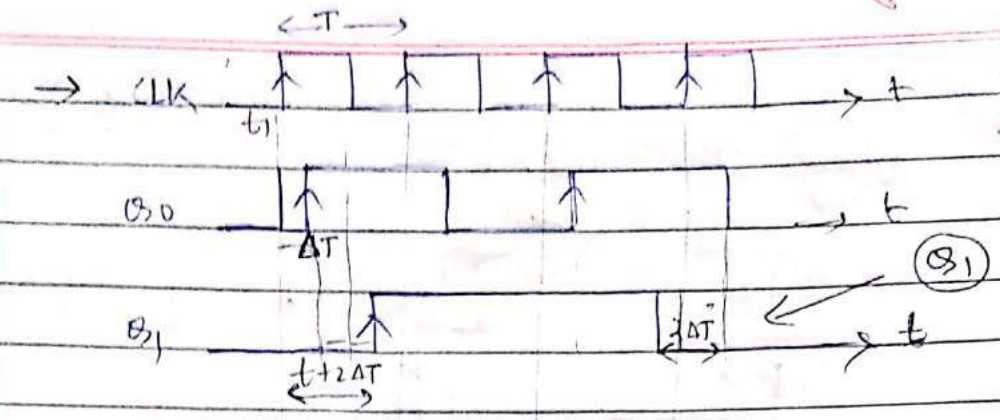
q → 2008

(Q.11) For each of the +ve edge triggerd JK-FF used in the following fig, the propagation delay is ΔT.

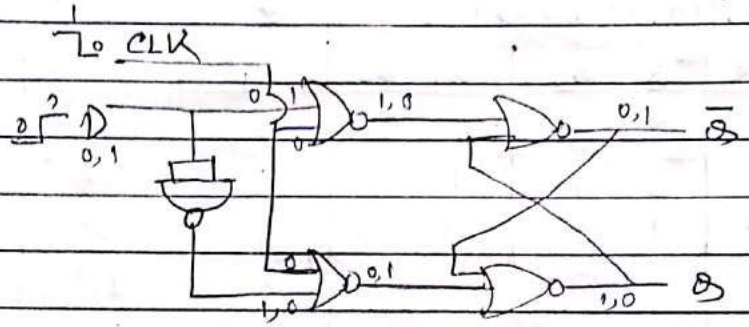




Draw the o/p wave ~~from~~ form at Q_1



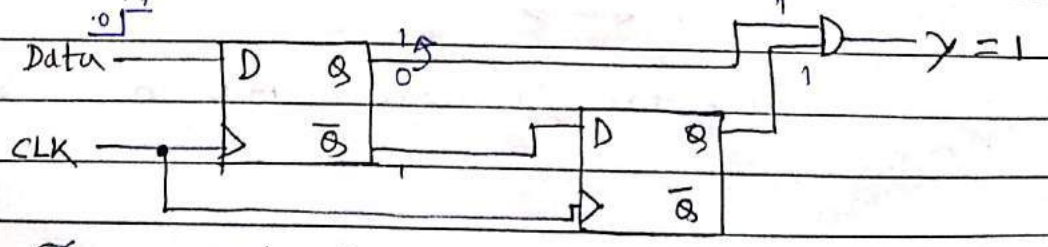
w-2008 (Q.12) For the circuit shown in the fig, D has a transition from 0 to 1 after CLK changes from 1 to 0. Assume gate delays are ge negligible.



Which of the following statement is true?

(a) Q goes to 1 at the clock transition & stays at '1'
(b) Q goes to 0 " " " " '0'
(c) Q goes to 1 " " & goes to 0 when D goes to
(d) Q goes to 0 " " & " " '1' when D goes to 1

(Q.13) When the O/p y in the circuit below is '1', it implies Data has



(a) changed from 0 to 1 (d) not changed.
(b) changed " 1 to 0
(c) changed in either direction

Q.14 (Sol) Two D-FFs are connected as a synchronous counter that goes through the following $Q_B Q_A$ sequence $00 \to 11 \to 01 \to 10 \to 00 \to \cdots$. The connections to the i/p's D_A & D_B are —

(a) $D_A = Q_B$, $D_B = Q_A$

(b) $D_A = \overline{Q}_A$, $D_B = \overline{Q}_B$

(c) $D_A = (Q_A \oplus Q_B)$, $D_B = Q_A$

(d) $D_A = (Q_A \odot Q_B)$, $D_B = \overline{Q}_B$

	D_B		D_A			
CLK	Q_B	Q_A	Q_B	Q_A	D_B	D_A
1	0	0	1	1	0	1
2	1	1	0	1	1	1
3	0	1	1	0	0	0
4	1	0	0	0	1	0

D_B	Q_B \ Q_A	0	1
	0	0	1
	1	1	0

$D_B = \overline{Q}_B$

D_B	Q_B \ Q_A	0	1
	0	X	0
	1	0	X

$D_B = Q_A \odot Q_B$

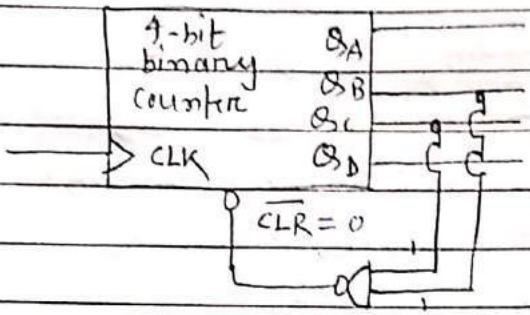
D_A	Q_B \ Q_A	0	1
	0	0	1
	1	1	0

$D_A = Q_A$ or $D_A = \overline{Q}_B \overline{Q}_A + Q_A Q_B = \odot$

$D_A = \overline{Q}_A \oplus \overline{Q}_B$

q-2015 (Q15)

A MOD-n counter using a synchronous binary up-counter with synchronous clear i/p is shown in the fig. The value of n is —



→

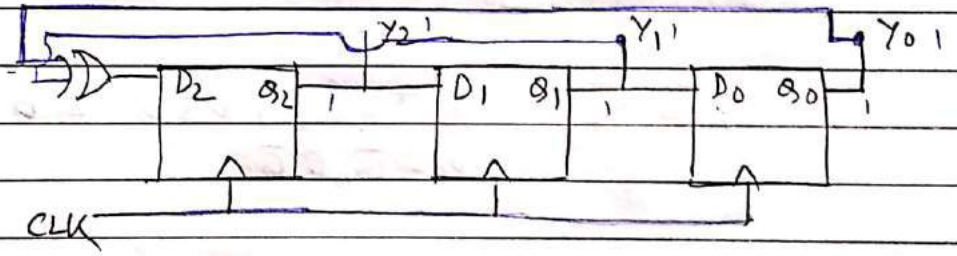
Q_A	Q_B	Q_C	Q_D	
0	1	1	0	→ 6

0 → 1 → 2 → 3 → 4 → 5 → 6

MOD-7 Counter.

q-2015 (Q16)

A 3-bit pseudo random number generater is shown. Initially the value of i/p $y = Y_2 Y_1 Y_0$ is set to 111. The value of i/p y after 3 clock cycles is.

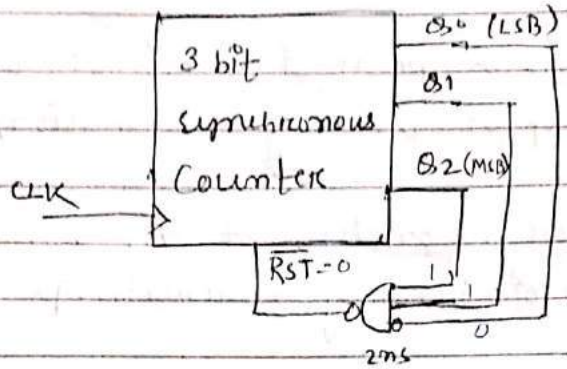


	P.S	N.S	$Q_1 \oplus Q_0$	Q_2	Q_1
→ CLK	Q_2 Q_1 Q_0	Q_2 Q_1 Q_0	D_2	D_1	D_0
1	1 1 1	0 1 1	0	1	1
2	0 1 1	0 0 1	0	0	1
3	0 0 1	1 0 0	1	0	0
		Y =			

$1\ GHz = 10^9$ $\qquad$ 10^{-9} sec = 1 nsec

Page No :
Date : 1-4-19

(8.11) For a circuit shown in the fig, The delay of the bubbled NAND gate is 2ns & that of the counter is assumed to be zero.



If the clock freq is 1 GHz, then the counter behaves as a -

(a) Mod - 5 counter
(b) " - 6 "
(c) " - 7 "
(d) " - 8 "

→ $f_{CLK} = 1\ GHz$
$= 10^9$ sec

$T_{CLK} = \frac{1}{10^9 sec} = 10^{-9}$ sec = 1 ns.

$N \leq 2^n \leq 2^3$

$N = 8$

$Q_2\ Q_1\ Q_0$
1 1 0 → 6 + 2 → 8

Mod - 8

Page No :
Date : 180

• REGISTERS :

→ a Flip-Flop can store 1-bit of binary information.

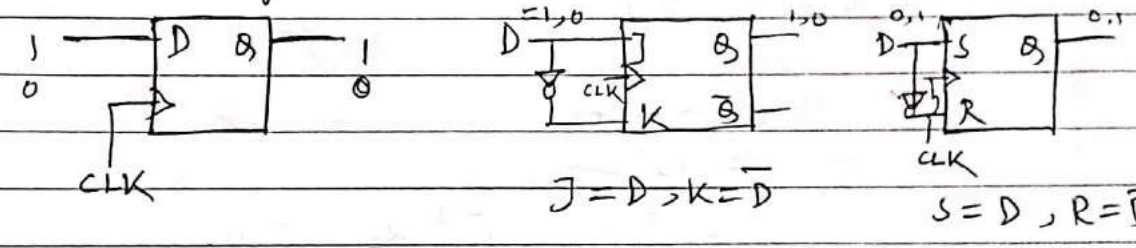
→ A register is a group of flip-flops which can store a group of binary information.

→ n-bit register, consists of n, no. of FFs, it can store n-bit of binary information.

→ 4-bit register, 8 bit register.
A, B, ··· E, F

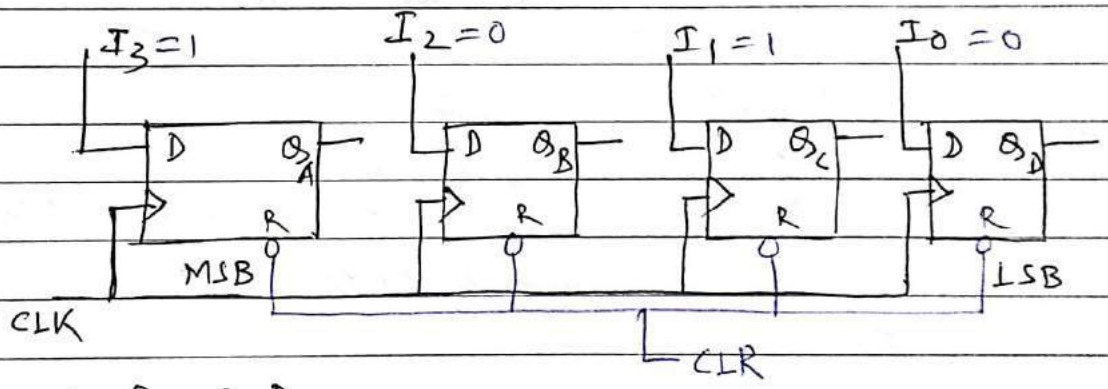
• 4-bit register :

→ generally D-FF are use in register.



$J = D, K = \bar{D}$

$S = D, R = \bar{D}$



A	B	C	D
1	0	1	0
(MSB)			(LSB)

Page No :
Date : 18

Operation =

(1) $CLR = 0$, $R = 0$

Q_A	Q_B	Q_C	Q_D
0	0	0	0

(2)

$I_3 = 1$

$I_2 = 1$

$I_1 = 0$

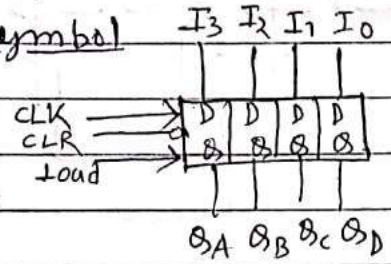
$I_0 = 0$

$D_3 = 1, D_2 = 1, D_1 = 0, D_0 = 0$

When we apply a clk

	Q_A	Q_B	Q_C	Q_D
CLK-1→	1	1	0	0

• Symbol



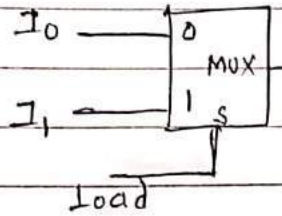
Load = 0 → The O/p of register will remain same.

Load = 1 → The Data which is available at the i/p of the D-FF will be transferred.

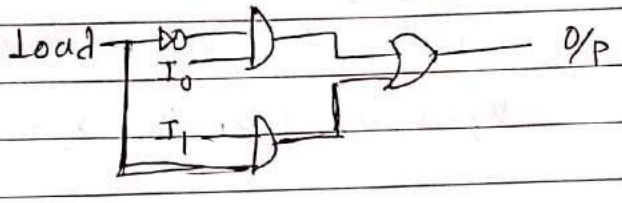
→ This Load can be
- Synchronous
  → (Load↑, CLK↑)
- Asynchronous
  → Load↑
  (It will not depend on the CLK)

Page No :
Date : 1 8 2

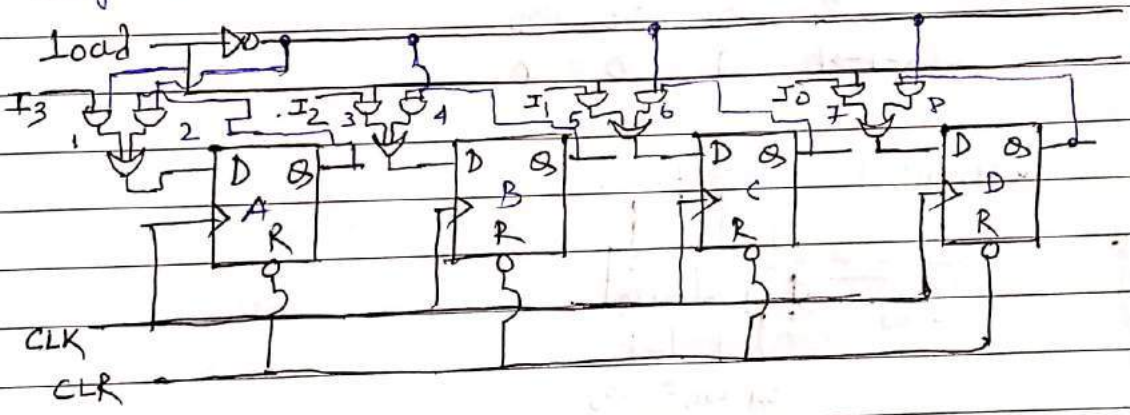
Load = 0 → same O/P.
Load = 1 → Load new information.



$O/P = I_0 \overline{Load} + I_1 Load$.



• Register with parallel Load —



Load = 0 → 2, 4, 6, 8 AND gates (Enables) will be ON
1, 3, 5, 7 → OFF Disable

Load = 1 → 1, 3, 5, 7 → (Enable)
2, 4, 6, 8 → (Disable)

✓ (Clear understanding)

Page No :
Date : 1 8 3

• Applications of Registers =

→ will be used as a general purpose registers in the microprocessors.

8085, 8086

A, B, C, ... E, F, G

8-bit register

semiconductor

RAM or ROM

(permanent storage memory)

(program store)

registers. (temporary storage memory)

(result store)

ex: ADD A.

→ Shift Register → SISO
→ SIPO
→ PIPO
→ PISO

• Shift register :

Types of shift registers =

→ Serial-in serial out (SISO) shift register.
→ Serial-in parallel-out (SIPO) shift register.
→ Parallel-in serial out (SIPO) shift register.
→ Parallel-in parallel out (PIPO) shift register.

(1) SISO shift register =

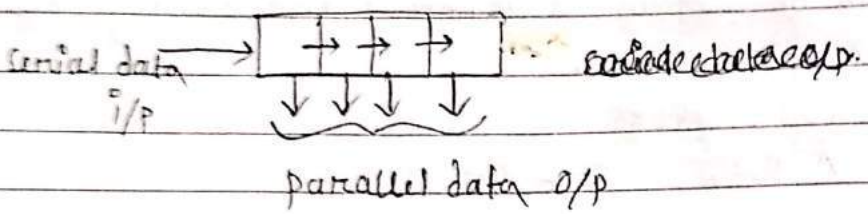
(a)

Shift-Right SISO

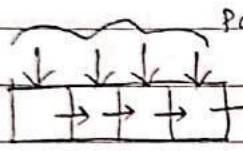
(b)

Shift-left SISO.

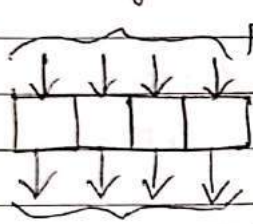
(2) SIPO shift register =



(3) PISO shift register =

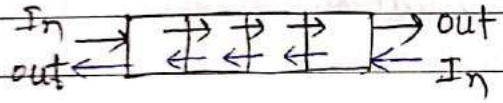


(4) PIPO shift Register =

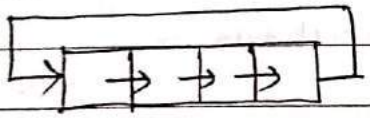


(5) Bidirectional shift register =

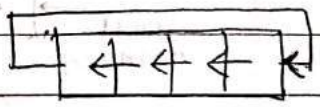
SISO, shift-left, shift-right.



(6) Rotate right shift register =

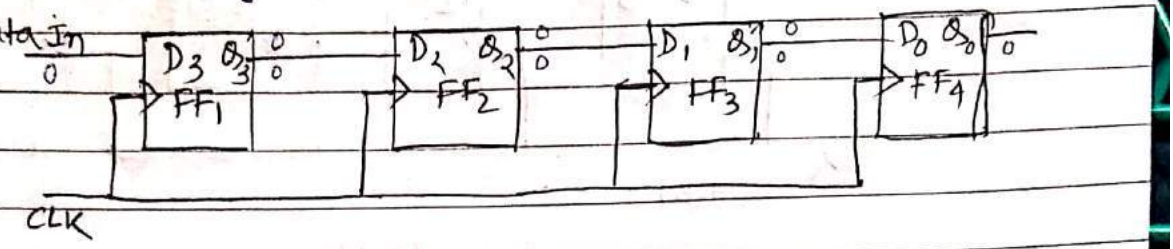


(7) rotate-left shift reg =



- SISO (Serial in serial out) shift registers =

(a) Shift - Right :



State Q_3 Q_2 Q_1 Q_0

1 0 1 0 → LSB

MSB ↙ ⟶

Operation :

In

CLK	Data In	Q_3	Q_2	Q_1	Q_0
Initially	–	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

↑ Stored data.

Out :

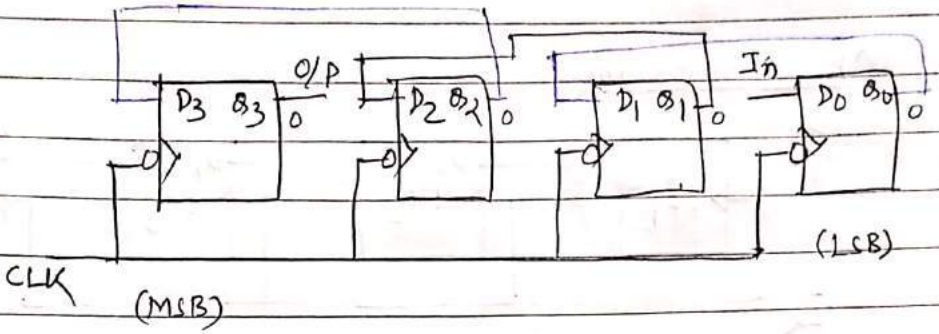
CLK	Data In	Q_3	Q_2	Q_1	Q_0	serial output
Initially	–	1	0	1	0	0 (LSB)
5	0	0	1	0	1	1
6	0	0	0	1	0	0
7	0	0	0	0	1	1 → MSB

→ Total '7' CLK pulses required to store the data serially & taken out serially.

→ If n-bit shift reg SISO

(2n-1) → clock pulses required.

(b) Shift left :



Store 1010 (MSB → LSB)

Operation =

CLK	Q_3	Q_2	Q_1	Q_0	In
Initially	0	0	0	0	-
1	0	0	0	1	1
2	0	0	1	0	0
3	0	1	0	1	1
4	1	0	1	0	0

Stored data.

output	CLK	Q_3	Q_2	Q_1	Q_0	In
1	initially.	1	0	1	0	0
0	1	0	1	0	0	0
1	2	1	0	0	0	0
0	3	0	0	0	0	0

→ Total - 7 CLK pulses required to complete the operation.

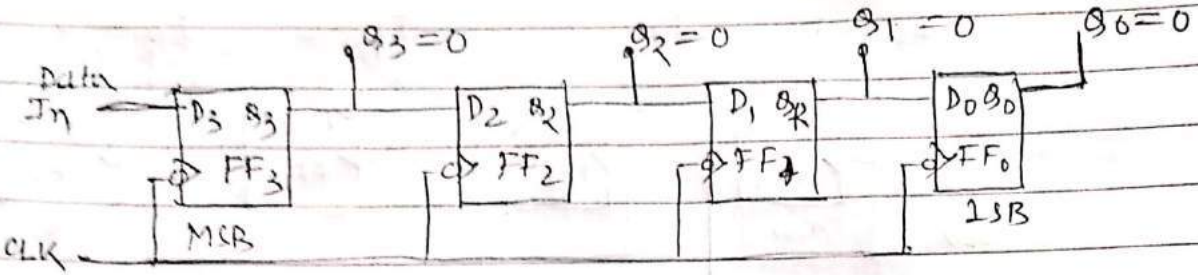
→ 1010 — [7 CLK pulses] — 1010

T_{CLK} = 1 μsec

1010 — [7 μsec] — 1010

→ SISO → Time delay application (used digital signal processing)

- 4-Bit SIPO (Circular in serial out parallel) shift register =



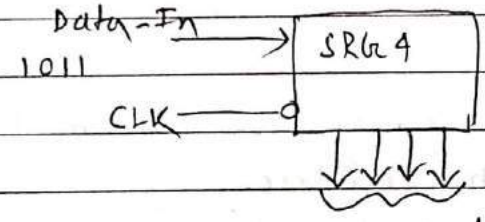
Store 1011 → LSB
MSB

Operation

CLK	Data-In	Q_3 Q_2 Q_1 Q_0
Initially	–	0 0 0 0
1	1 →	1 0 0 0
2	1 →	1 1 0 0
3	0 →	0 1 1 0
4	1 →	1 0 1 1 store

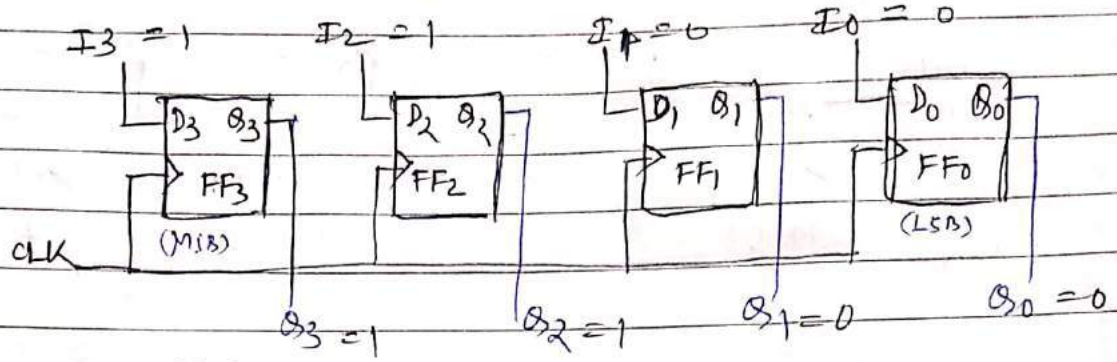
→ After 4 CLK pulses the data is available.

Application of SIPO shift-Registers =



→ serial to parallel conversion of data.

- PIPO (parallel in parallel out) shift - Register =

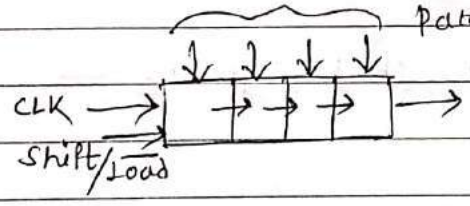


also called
↓
Register / Storage register / Buffer register.

CLK	I_3	I_2	I_1	I_0	Q_3	Q_2	Q_1	Q_0
initially		—			0	0	0	0
1	1	1	0	0	1	1	0	0

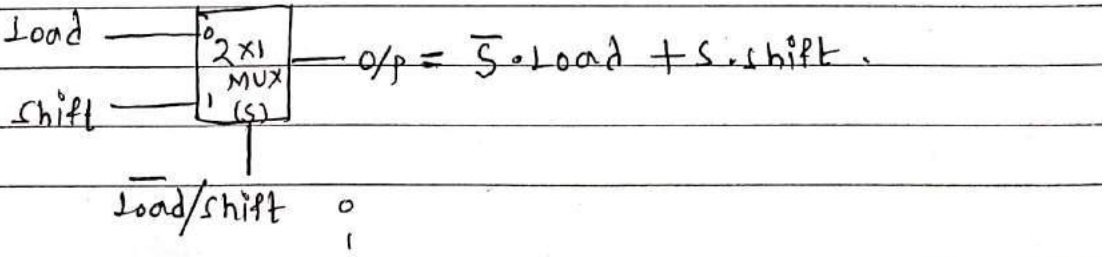
(shifting is not happening in this reg.)

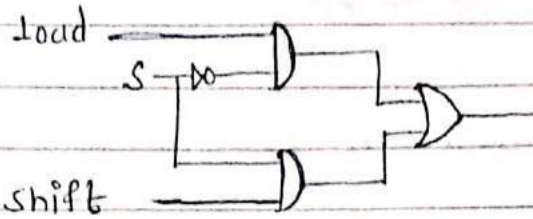
- PISO (parallel-in-serial-out) shift-registers =



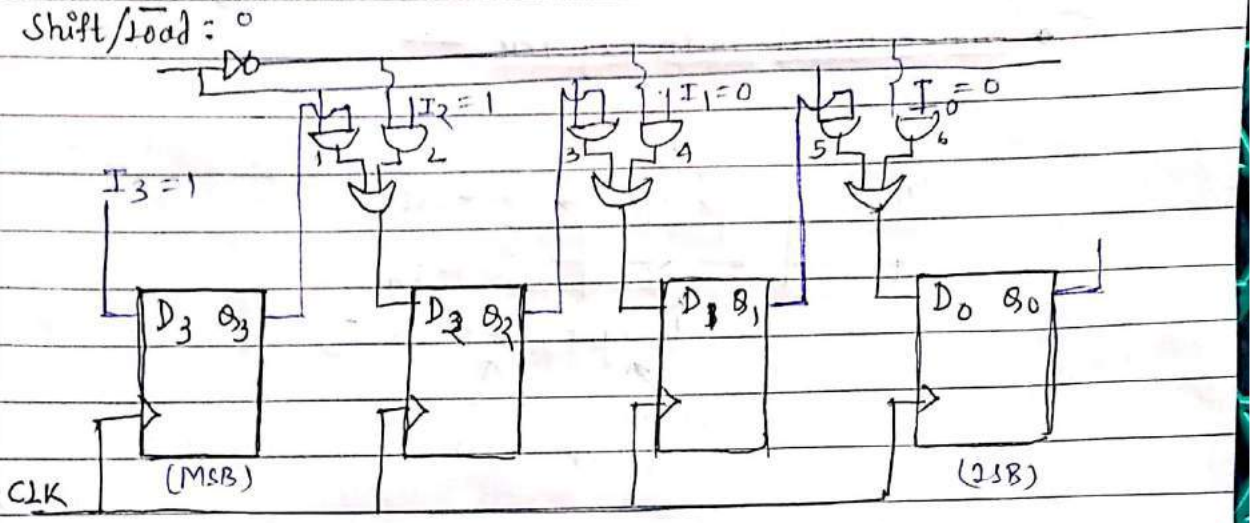
shift/$\overline{Load}$ = 0 → parallel data will be loaded into the Registers.

shift/$\overline{Load}$ = 1 → The Data will be shifted towards Right side.





• 4-bit PISO shift register — (Store 1100)



Operation — MSB 1 1 0 0 LSB

CLK	shift/load	parallel-data $I_3\ I_2\ I_1\ I_0$	$Q_3\ Q_2\ Q_1\ Q_0$	serial output
initially	0 —	—	0 0 0 0	
↑ (1)	0	1 1 0 0	1 1 0 0	0
↑ (2)	1	1 1 0 0	1 1 0 0	0
↑ (3)	1	1 1 0 0	1 1 1 0	1
↑ (4)	1	1 1 0 0	1 1 1 1	1

Total 4- clock pulses required to complete the operation. (Loading & shifting)

When, shift/load = 0, (2, 4, 6) AND gate are enable.
(1, 3, 5) " " are disable.

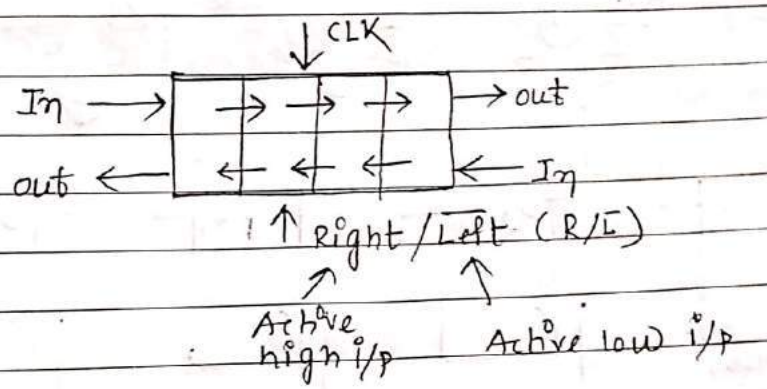
shift/load = 1, (1, 3, 5) AND gate are enable.
(2, 4, 6) AND gate are disable.

- Application =

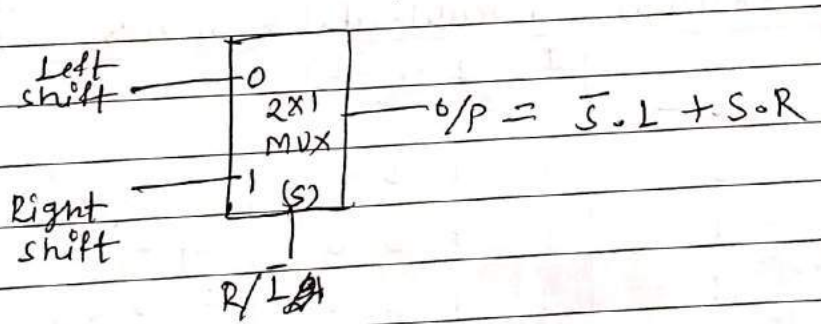
→ parallel-to-serial Conversion

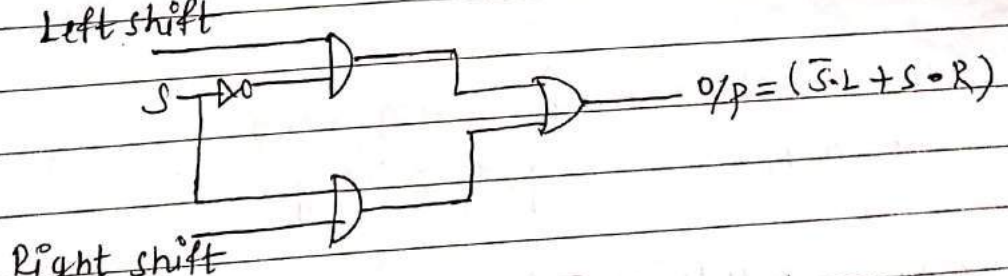
→ used in microprocessors (8085, 8086).

- Bidirectional shift Register =



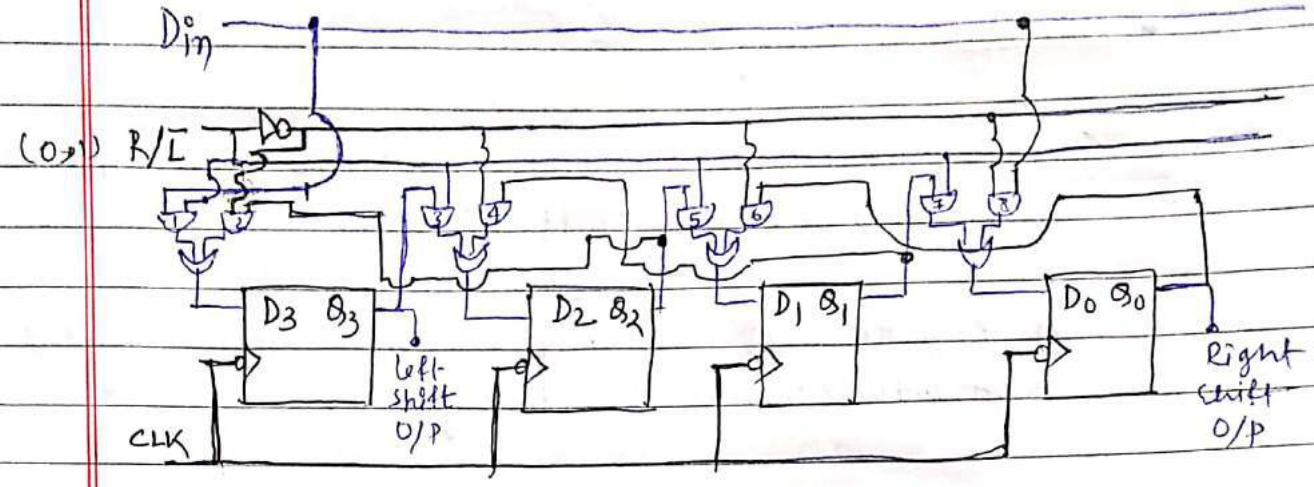
→ R/L̄ = 0 → Left-shift
→ R/L̄ = 1 → Right-shift





• Logic diagram of 4-bit bidirectional shift reg =

MSB 1 1 0 1 LSB



operation — 1101

CLK	R/I	Din	Q_3 Q_2 Q_1 Q_0	O/P
Initially	–	–	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	
~~DELAY~~			Q_0 Q_1 Q_2 Q_3	
↓ (5)	0	1	0 0 0 0	
↓ (6)	0	1	1 0 0 0	
↓ (7)	0	0	0 1 1 0	
↓ (8)	0	1	1 0 1 1	Q_3 Q_2 Q_1 Q_0 = 1 1 0 1

(right shifting; left shifting)

Page No :
Date : 19/2

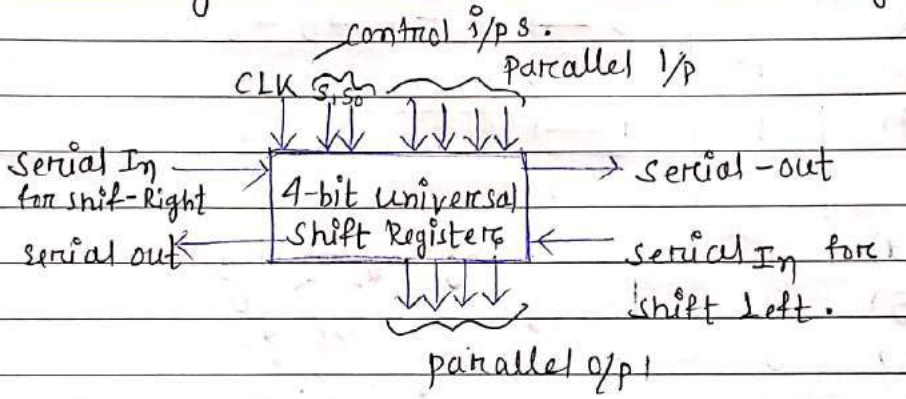
- **Universal shift register =**

→ A Register capable of shifting in one direction only is a unidirectional shift register.

→ A Register that can shift in both directions is a bidirectional shift register.

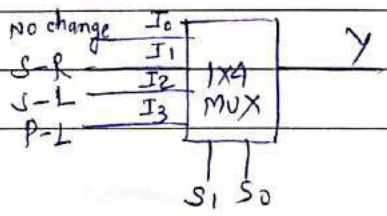
→ If the register has both shifts and parallel load capabilities, it is referred to as a universal shift register.

- **Block diagram of 4-bit Universal shift register =**

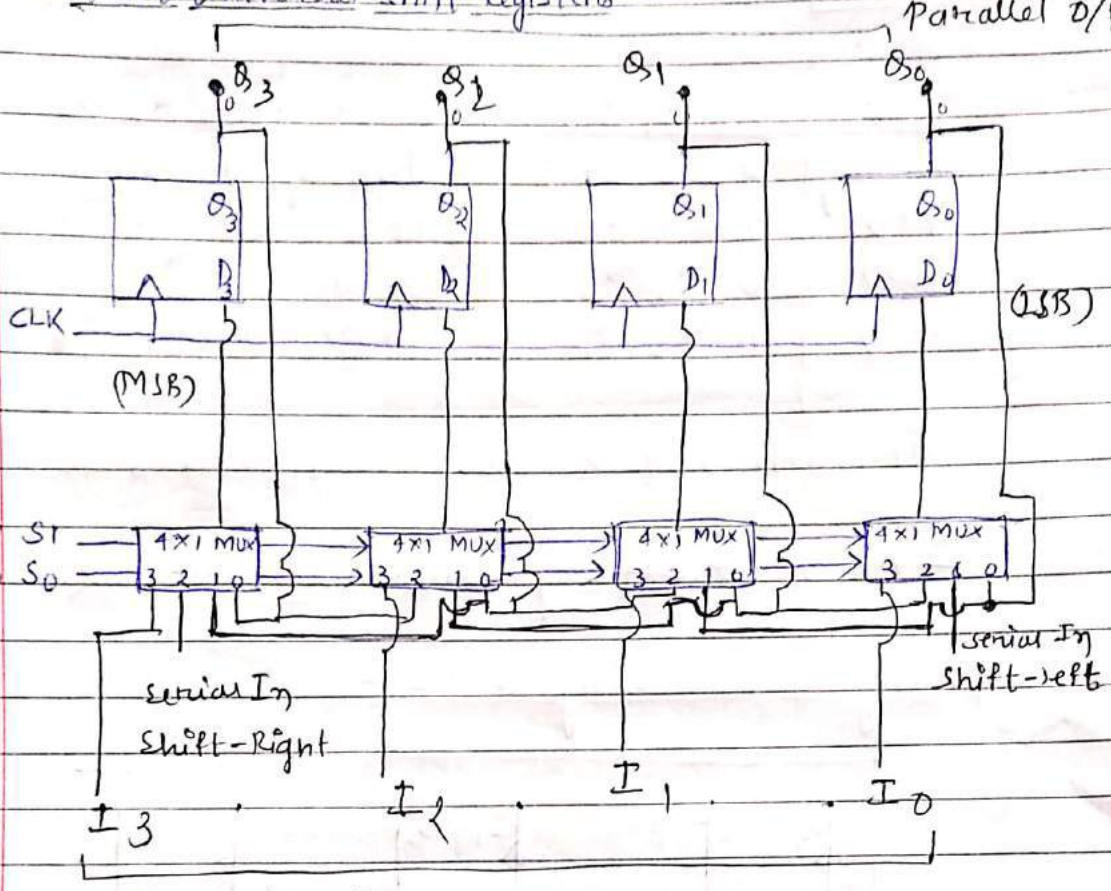


- **function table —**

S_1	S_0	Register operation
0	0	No change
0	1	shift left
1	0	shift Right
1	1	Parallel load.



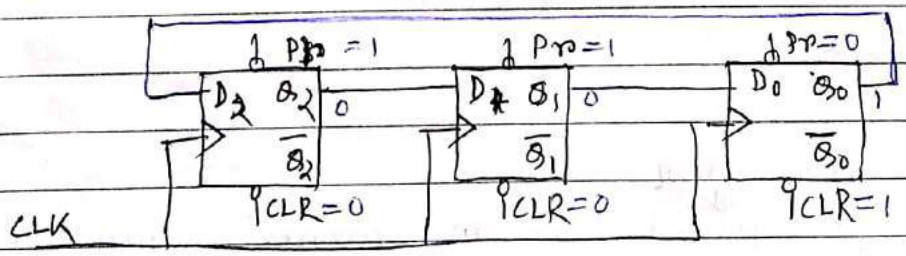
• 4-Bit Universal shift Register =



• Shift Register Counters =

→ Ring counter.
→ Johnson Counter (Twisted ring Counter)

• Ring Counter =



3-bit Ring Counter

Pr = 0, Q = 1
CLR = 0, Q = 0

operation =

CLK	Q_2 Q_1 Q_0
Initially	0 0 1
↑ (1)	1 0 0
↑ (2)	0 1 0
↑ (3)	0 0 1

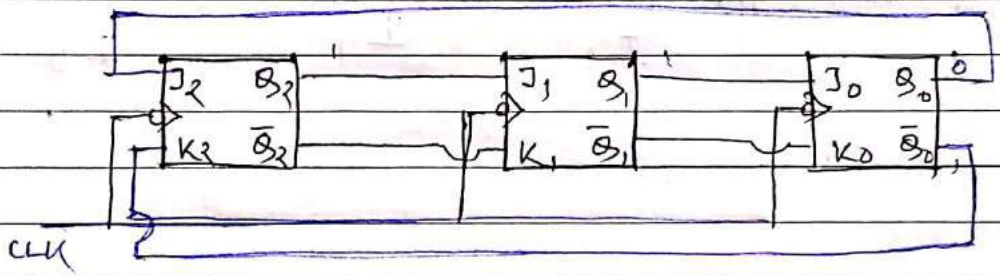
It counts 3 clock pulses.

CLK	Q_2 Q_1 Q_0
Initially	1 0 0
↑ (1)	0 1 0
↑ (2)	0 0 1
↑ (3)	1 0 0

It counts 3 CLK pulses.

** ⇒ An n bit ~~Ring~~ Ring Counter count n - clock pulses.

• Ring counter using JK-FF =



operation =

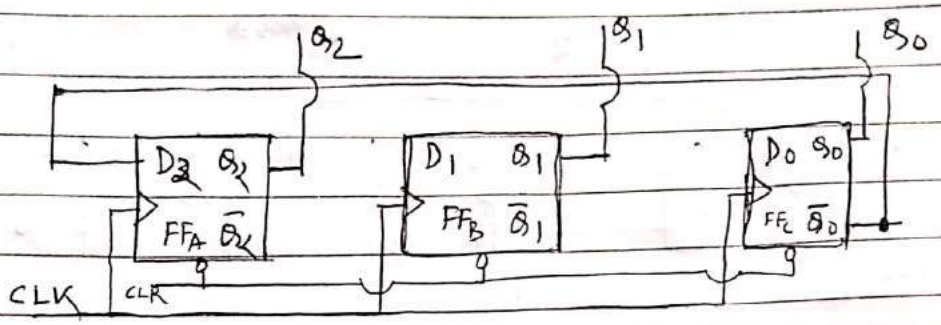
CLK	Q_2 Q_1 Q_0
Initially	1 1 0
↓ (1)	0 1 1
↓ (2)	1 0 1
↓ (3)	1 1 0

It counts 3 CLK pulses.

Application

⇒ pseudo random sequence generator.
(use in)

• Johnson's Ring Counter (Twisted Ring counter) =



operation =

CLK	$\bar{Q}_0$	Q_2 Q_1 Q_0
Initially		0 0 0
(1)	1	1 0 0
(2)	1	1 1 0
(3)	1	1 1 1
(4)	0	0 1 1
(5)	0	0 0 1
(6)	0	0 0 0

→ A 3-bit Twisted Ring counter can count 6 - clk pluses.

** ** → An n-bit Twisted Ring counter can count 2n-cp.

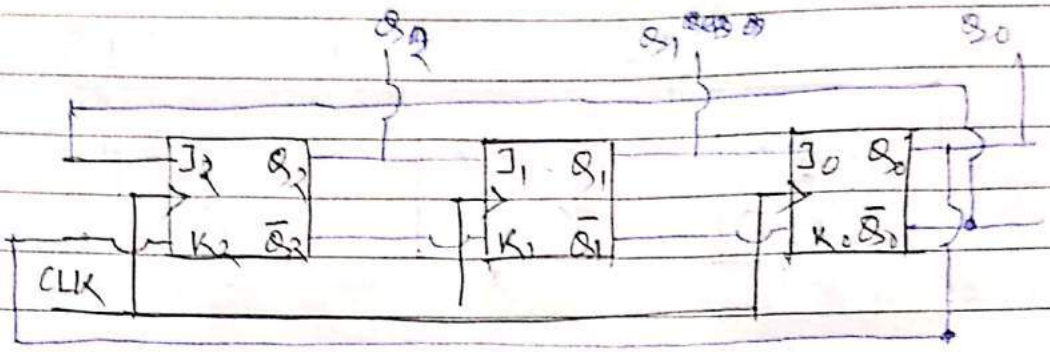
→ n-bit Ripple counter | frequency division = $f/2n$

→ " Ring counter = f/n

→ " Johnson's counter = $f/2n$

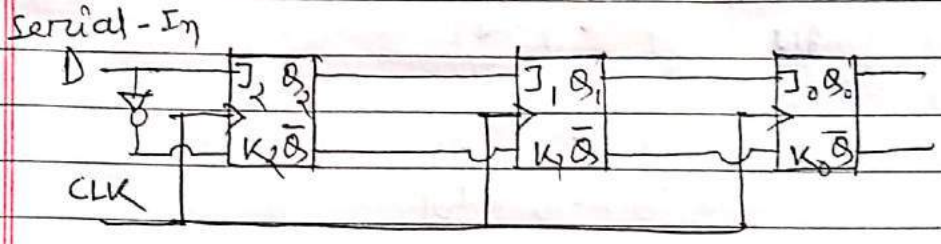
Page No :
Date : 196

• Twisted Ring Counter using JK-FF =



→ It can count - 2n clock pulses.

• 3-bit SISO Shift register using JK-FF =



$J = D$

$K = \bar{D}$