Sequential Circuits

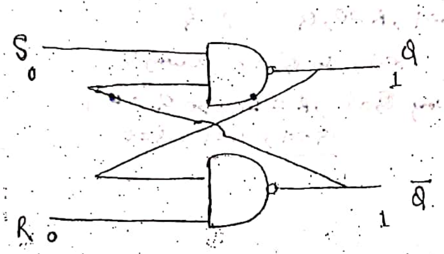
Present o/p depends on
" i/p & past o/p

flip flops:

(i) flip flop is a basic memory element
(ii) It can be used to store 1 bit.
(iii) FF's have two stable states hence it is known as bistable multivibrator.
(iv) FF's have two o/p which are complementry to each other.

SR Latch → cross coupled not Gate ⇒ 

(i) Using Nand Gate



1 or 0 } same

Nand

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

0 or 1 } same o/p 1

c → 0
d → 0

S	R	Q	$\bar{Q}$
0	0	Invalid / Forbidden State	
0	1	1	0
1	0	0	1
1	1	NC / previous state	

(Q & $\bar{Q}$ both are 1) (they req. to be compl^{ty})

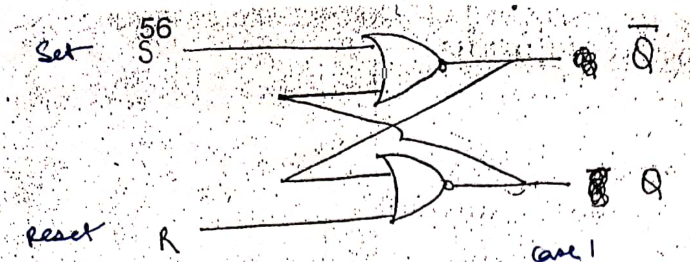
$\left(\begin{matrix} Q=0, \bar{Q}=1 \\ \text{or} \\ Q=1, \bar{Q}=0 \end{matrix} \right)$*

→ In SR latch, if both gates are disabled o/p is in invalid state

→ If both gates are enabled, o/p is in previous state ($\bar{Q}=1, Q=0$ (or) $\bar{Q}=0, Q=1$)

(ii) SR latch using NOR Gate

56



NOR Gate TT

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

d → 1
e → 0

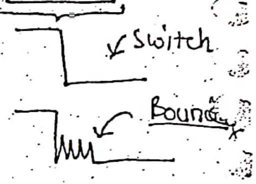
T.T

S	R	Q
0	0	previous state
0	1	0
1	0	1
1	1	invalid ($\bar{Q} = Q = 0$)

case 1
$S=0, R=1$
$R=1, Q=0$ (always)
$S=0, Q=0$ so $\bar{Q}=1$
Now, $S=0, R=0$,
Now, $\bar{Q}=1$ (prev)
so $R=0, \bar{Q}=1$, so $Q=0$
Now
$Q=0, S=0, \bar{Q}=1$
(Memory)

latch →

SR latch is used in all FF's & to eliminate bounce in switches

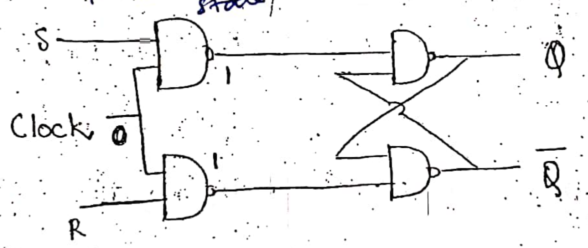


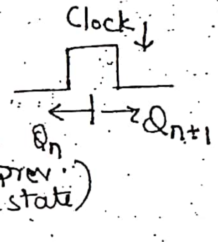
case 2
$S=1, R=0, Q$ & $\bar{Q}$
$S=1$, so $\bar{Q}=0$ (always)
Now $\bar{Q}=0, R=0$, so $Q=1$
$Q=1, S=1, \bar{Q}=0$

case 3
$S=1, R=1, Q$ & $\bar{Q}$
$S=1$, so $\bar{Q}=0$
$\bar{Q}=0, R=1$, so $Q=0$
same o/p (invalid state)

SR FF ...

Nand Gate



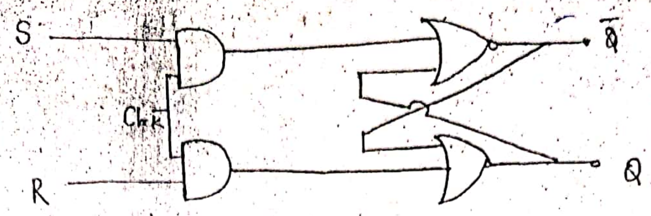


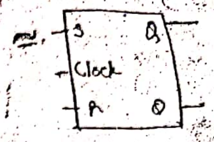
Clock	S	R	Q_{n+1}
0	0	0	Q_n
0	0	1	Q_n
0	1	0	Q_n
0	1	1	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	invalid

⟺

Clock	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n → HOLD
1	0	1	0 → Reset State
1	1	0	1 → Reset State
1	1	1	invalid ↳ unused state

(II) Using NOR latch





Clock	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

Characteristic table of SR FF

S	R	Q_n	Q_{n+1}	
0	0	0	0 (Q_n)	Q_n
0	0	1	1 (Q_n)	
0	1	0	0	0
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	X	unused
1	1	1	X	

S R
1 0

S R
1 $\bar{Q}$

R
0
0

$$Q_{n+1} = \sum m(1,4,5) + \sum d(6,7)$$

S \ RQ_n	$\bar{R}\bar{Q}_n$	$\bar{R}Q_n$	RQ_n	$R\bar{Q}_n$
$\bar{S}$		1		
S	1	1	x	x

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

and
$S.R = 0$

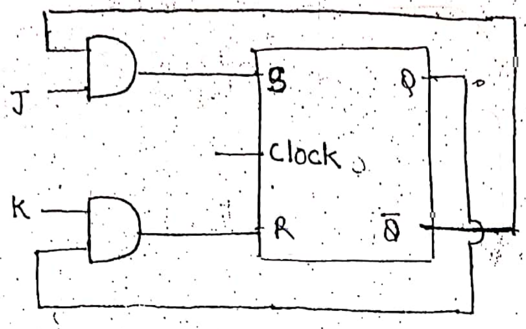
Excitation table of SR FF

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

* Disadvantage of SR FF is invalid state is present when S and R are 1. To Avoid this JK Flip flop is used.

JK flip-flop

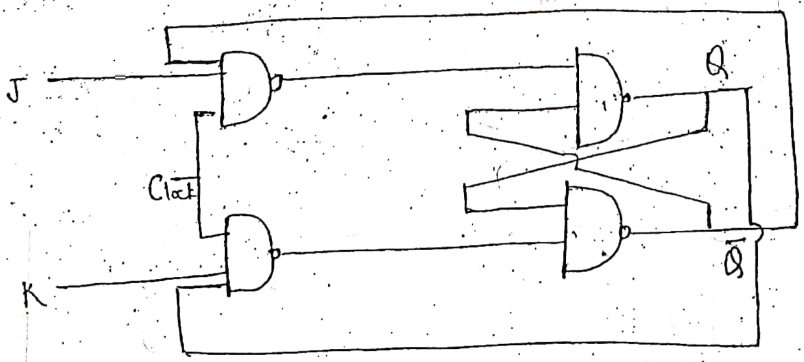
1) Using SR FF



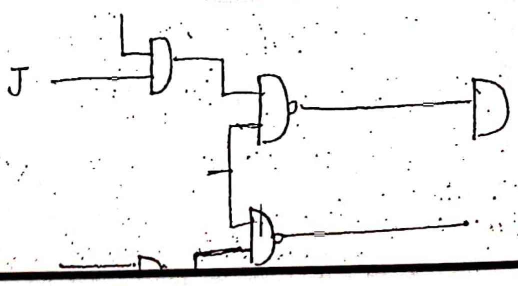
Truth table

Clock	J	K	Q_{n+1}	
0	X	X	Q_n	
1	0	0	Q_n	.... HOLD
1	0	1	0	.... RESET
1	1	0	1	.... SET
1	1	1	$\bar{Q}_n$	.... [Toggle State] → Race around condition

Using Nand Gates



Nand do not follow Associative



Assume
$\bar{Q}=1$
$Q=0$

Gives o/p of NAND

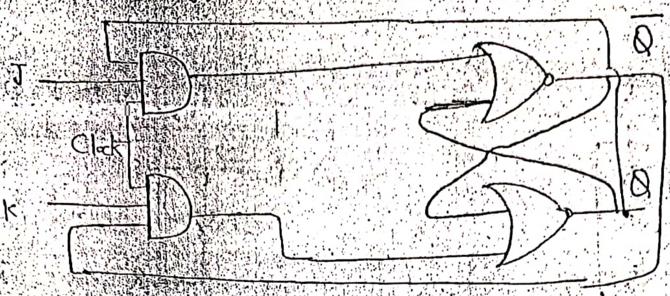
$J=1$, $clk=1$, $\bar{Q}=1$, ⇒ $0 \rightarrow Q=1$

$K=1$, $clk=1$, $Q=0$ ⇒ $1 \rightarrow \bar{Q}=0$

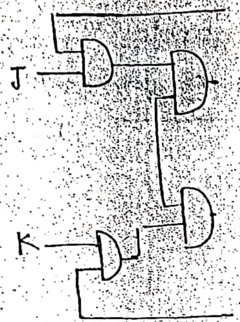
now $J=1$, $clk=1$, $\bar{Q}=0$ ⇒ $1 \rightarrow Q=Q$

$K=1$, $clk=1$, $Q=1$ ⇒ $0 \rightarrow \bar{Q}=1$

(iii) JK FF using NOR



or



Characteristic table of JK FF

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

$Q_{n+1} = m\sum(1,4,5,6)$

J \ KQ_n	$\overline{K}\overline{Q}_n$	$\overline{K}Q_n$	KQ_n	$K\overline{Q}_n$
$\overline{J}$		1		
J	1	1		1

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

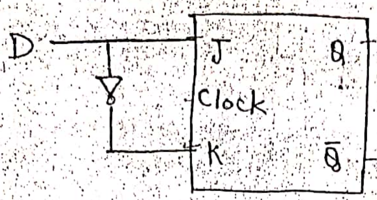
Excitation table

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

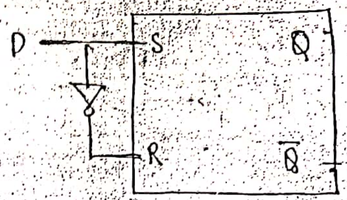
→ Disadvantage of JKFF is Race around condition.

(3) D- flip-flop

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



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



Truth table

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

D=0 Q
S=0, R=1 0
D=1
S=1, R=0 1

Characteristic table

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

$$\boxed{Q_{n+1} = D}$$

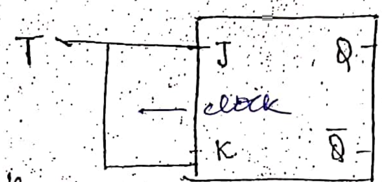
Hence also called Transparent latch

Excitation table

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

(4) T- flip flop Toggle flipflop

$J = K = T$



Truth table

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

~~Chara~~

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

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

Characteristic table

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

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

Excitation Table

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

Memory tips

→ Truth table

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

SR FF

DFF

T FF

→ Excitation table

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

→ Characteristic Eqn

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

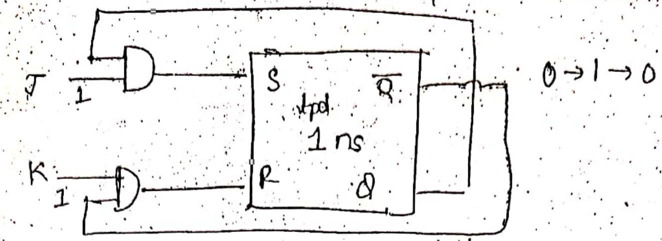
$$Q_{n+1} = J \overline{Q_n} + \bar{K} Q_n$$

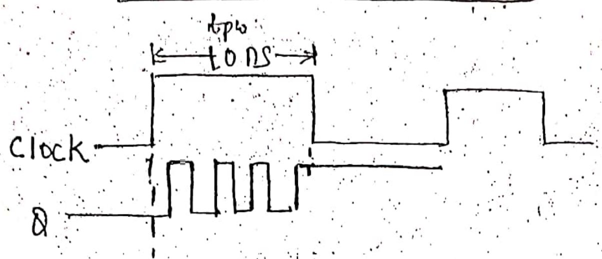
$$Q_{n+1} = D$$

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

62

Race around Condition



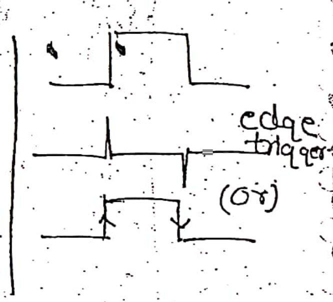


1) Race around occurs in JK FF when J = K = 1 and propogation delay of FF $\boxed{t_{p\,FF} << t_{pw}}$

) During Race around o/p will changes more than ones in single clock pulse

) Condition to avoid Race around

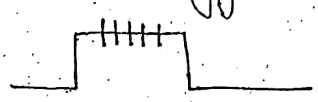
$$\boxed{t_{pw} < t_{p\,FF} < T_{clock}}$$



Master Slave JK FF is used to avoid race around condition

Triggers

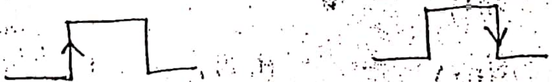
① Level trigger



② Edge triggering

+ve edge

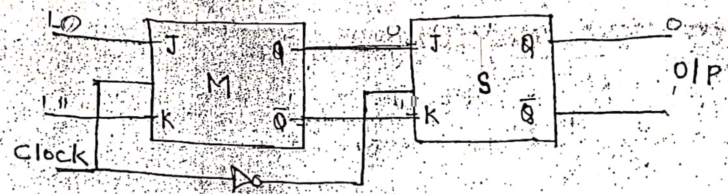
-ve edge



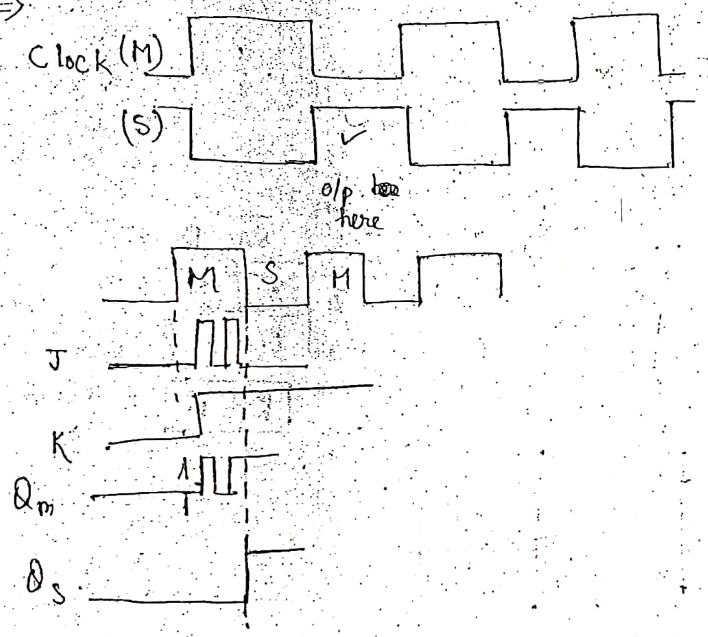
→ In edge triggering output will change only once in single clock

→ In Level trigger ckt. o/p can change many time in single cloc

Master – Slave



⟹



→ In Master – slave FF, Master is applied with i/p clock where as slave is applied with inverted clock.

→ In this o/p will changes when slave o/p is changing.

→ In Master Slave FF, Master is Level triggered whereas Slave is Edge trigger.

→ MS FF T.T is same as JK FF.

Conversions of flip flop

1. Identify available & reqd. flip flop
2. Make characteristic table for reqd. flip flop.
3. Make excitation table for available flip flop
4. Write boolean expression for available ff

i) JK ⟶ D FF

Steps (I) Identify available and required flip flop.

1. Write characteristic table of Required f.f
2. Write Excitation table of Available ff.
3. Write logical exp for inputs/excitations.

step (I) char. table of reqd. flip flop (D)

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

Step (II) excitation table for available flip flop

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

step (III) for J in terms of D & Q_n

for J

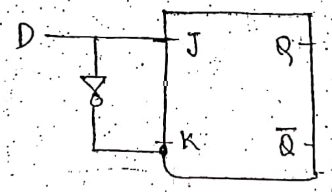
D \ Q_n	$\overline{Q_n}$	Q_n
$\overline{D}$	0	X
D	1	X

$\boxed{J = D}$

for K

D \ Q_n	$\overline{Q_n}$	Q_n
$\overline{D}$	X	1
D	X	

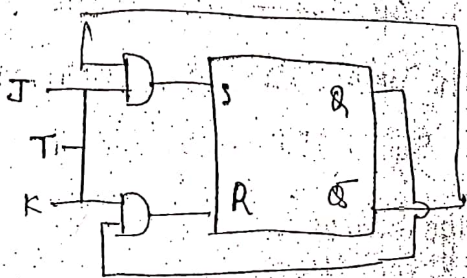
$\boxed{K = \overline{D}}$



ii) SR FF ⟶ JK FF

$S = J\overline{Q}$

$R = KQ$



iii) SR FF ⟶ D

$S = D$;
$R = \overline{D}$

iv) SR to T

$S = T\overline{Q}$

$R = TQ$

(V) 65 JK to SR

$J = S \;,\; K = R$

(VI) JK to D

$J = D \;;\; K = \overline{D}$

(VII) JK to T

$J = K = T$

(VIII) D to SR

$D = S + \overline{R}Q$

(IX) D to JK

$D = J\overline{Q} + \overline{K}Q$

(X) D to T

$D = T \oplus Q$

(XI) T to SR

$T = S\overline{Q} + RQ$

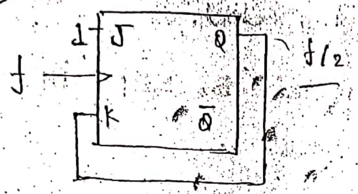
(XII) T → JK

$T = J\overline{Q} + KQ$

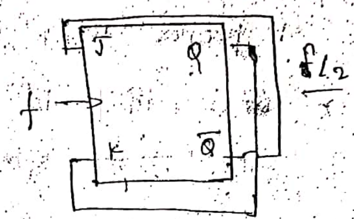
(XII) T to D

$T = D \oplus Q$

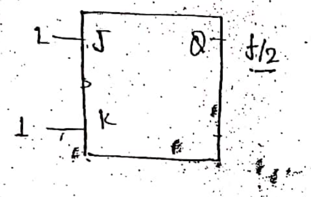
⇒



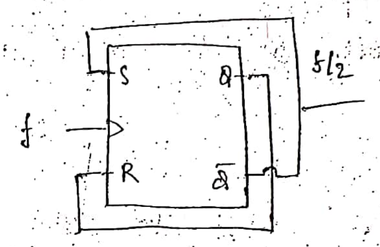
⇒



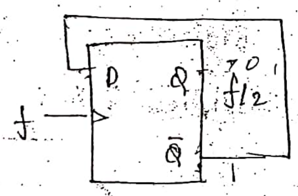
⇒



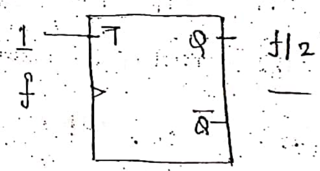
⇒



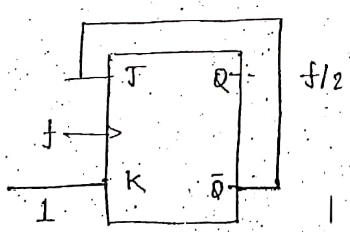
⇒



⇒



⇒



T-FF is a freq^n divider

x hz is i/p o/p will be $\frac{x}{2}$ hz.

Latch	FF
Level triggered	Edge triggered
asynchronous	synchronous.

Setup time → It is the min.^m time require to keep i/p at proper level before applying clock.

Hold time → It is the min^n time require to keep i/p at same level after applying clock.

If setup time and hold time is not properly maintain then output will go to meta stable state.

Registers

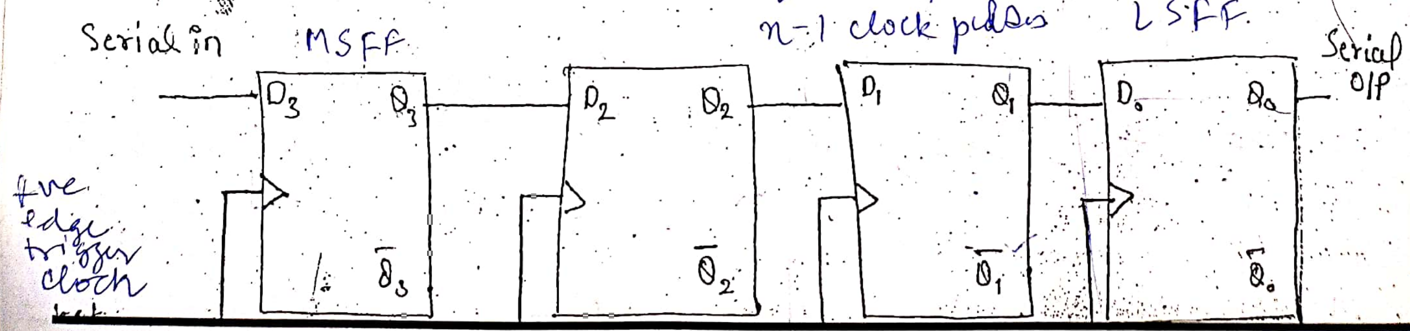
Registers are used to store group of bits.
To store n-bits, n FF's are used in Registers.
Depending on i/p and o/p registers can classified into.

SISO (ii) SIPO (iii) PISO (iv) PIPO

Based on application –
(1) Shift registers. (JK FF).
(2) Storage " (DFF)

SISO (serial in serial out)

n clock pulses
n-1 clock pulses



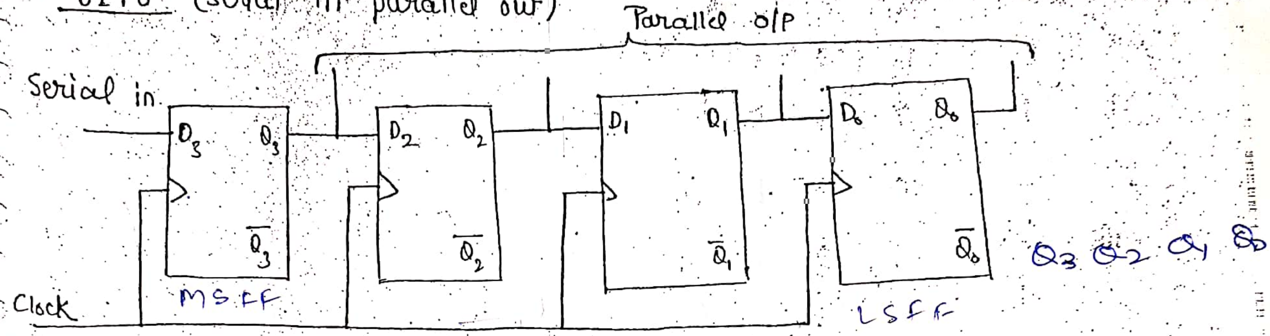
-ve triggered clock

(i) In SISO register to provide n-bit data serially in, it require n clock pulse.

(ii) SISO register is used to provide n Tclk delay to the input data

(iii) To provide n-bit data serially out in SISO register it require minimum n-1 clock pulse.

(b) SIPO (serial in parallel out)

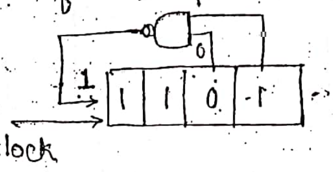


(i) In SIPO reg. to provide n bit data serially in it reque n clock pulse, and to provide parallel output no clock pulse is require

Serial — Temporial code
Parallel — Spacial code

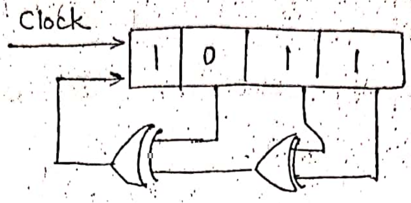
(ii) SIPO is used to $\boxed{\text{convert}}$ Temporial code into spacial code

Ques 1 Ckt in fig. is a 4 bit SIPO reg. which is initially loaded with 1101 after 3 clock pulses, the data present in register is



0 → 1 1 0 1
1 → 1 1 1 0
2 → 1 1 1 1
3 → 0 1 1 1

Ques: A 4 bit SIPO register loaded with 1011, if clock pulse are applied continuously then after how many clock pulses again data will become 1011

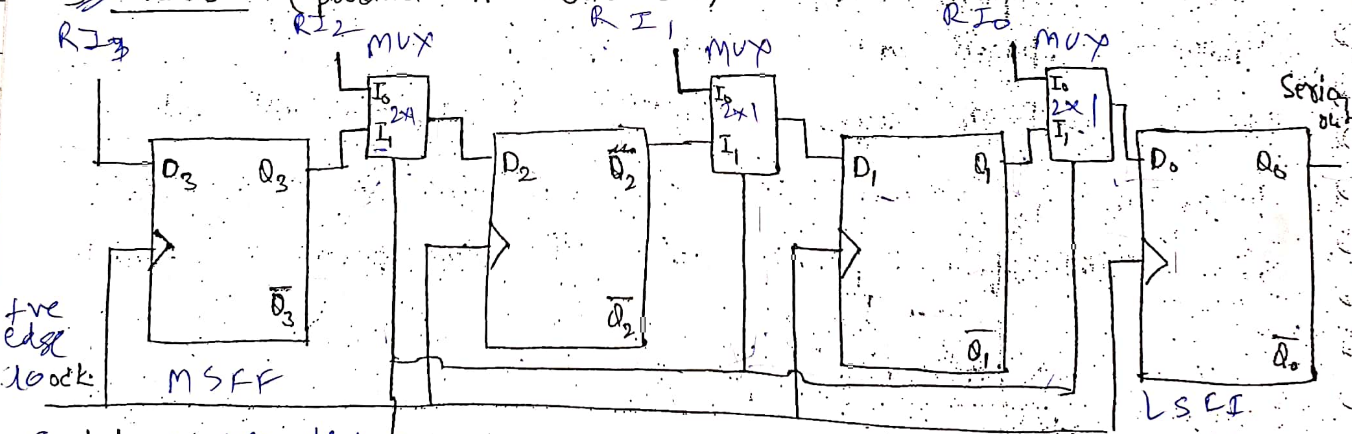


(a) 3
(b) 7
(c) 11
(d) 15

Sol:

	Clock	Q_3	Q_2	Q_1	Q_0
	0	1	0	1	1
(i)	1	0	1	0	1
(ii)	2	0	0	1	0
(iii)	3	1	0	0	1
(iv)	4	1	1	0	0
	5	1	1	1	0
	6	0	1	1	1
	7	1	0	1	1

D) PISO (parallel in Serial out)



Control 0 → Parallel In
1 → Serial Out

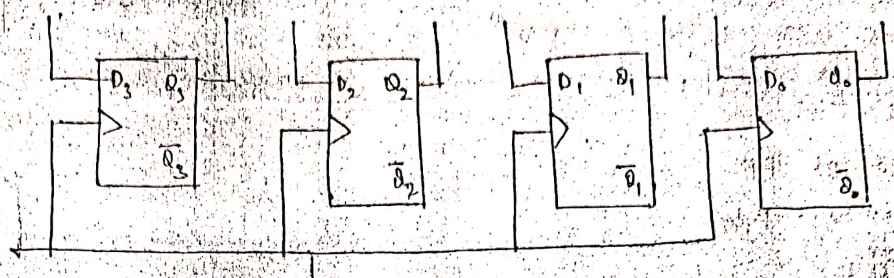
PISO I_0 → PISO (0)
SISO I_1 → SISO (1)

i) To provide ~~parap~~ pallel input 1 clock pulse is require and to provide serial out n-1 clock pulse is required

ii) Used to convert spacial code to temporal code.

71

PIPO (parallel in Parallel out)



i) Used as storage register in CPU
ii) for Parallel in 1 clock pulse is require.
iii) for parallel out 0 clock pulse is require.

	I/P	O/P	
SISO	n	n-1	Delay provision
SIPO	n	0	
PISO	1	n-1	← MUX
PIPO	1	0	(Fastest)

⟹ In shift register each shift left operation will multiply by two and shift right will divide by 2.

→ n shift left : $\times 2^n$

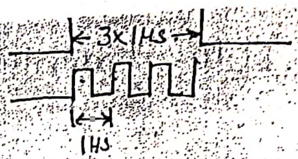
n shif righ : $\div 2^n$

COUNTERS

Imp i) Counters are basically used to count number of clock pulses applied.

ii) Used as frequency dividers
iii) Time measurement.
iv) frequency measurement
v) Range or distance measurement
vi) Pulse width measure.

Scanned with CamScanner



Basics

In Counters with n FF's max^m possible states are 2^n.

$n \rightarrow$ no. of FF

$N \rightarrow$ no. of states

2 ff $\rightarrow$ 4 states
3 ff $\rightarrow$ 8 states

$\boxed{N \leq 2^n}$

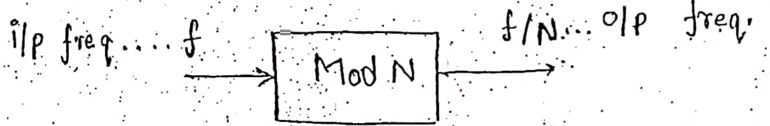
or, $\boxed{n \geq \log_2 N}$

Depending on clock pulse applied counters are of two types.

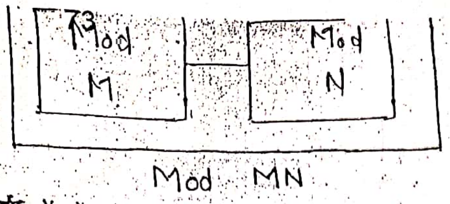
(i) Synchronous.
(ii) Asynchronous.

Synchronous	Asynchronous
(1) ALL FF's are applied with same clock	(i) Different FF's have different clock pulses
(2) faster	(ii) Slow
(3) Any count sequence can be design	(iii) fixed count sequence is possible
(4) No Decoding Errors	(iv) Decoding errors will present.
E.g. Ring Counter & Johnson counter	(v) E.g.: Ripple counter

imp. No. of states present in a counter is known as Modulus count. E.g. Mod N
$\rightarrow$ N States



(1) Decade counter (10 states) is applied with clock freq. of 100 KHz the output freq. is 10...

(ii) 

if Mod M & Mod N are cascaded then it will act as Mod MN counter.

** RIPPLE COUNTER

(i) It is an asynchronous ~~couter~~ counter.

(ii) These FF's are operated in toggle mode.

(iii) Diff^t FF's are applied with different clock pulses.

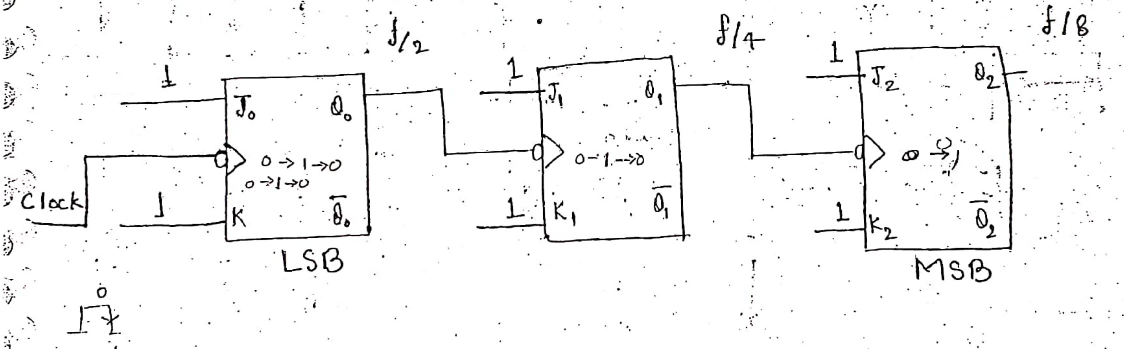
(iv) In this counter input clock is applied to only one FF and output of one FF's will act as clock to next FF.

(v) In this counter the FF applied with input clock will acts as LSB.

FF → LSB

Clock pulse

3 bit Ripple Counter



(i) The ~~sh~~ counter shown in figure, Q_0 toggle for every clock pulse -ve edge.

(ii) Q_1 toggles when Q_0 is changing from $1 \rightarrow 0$

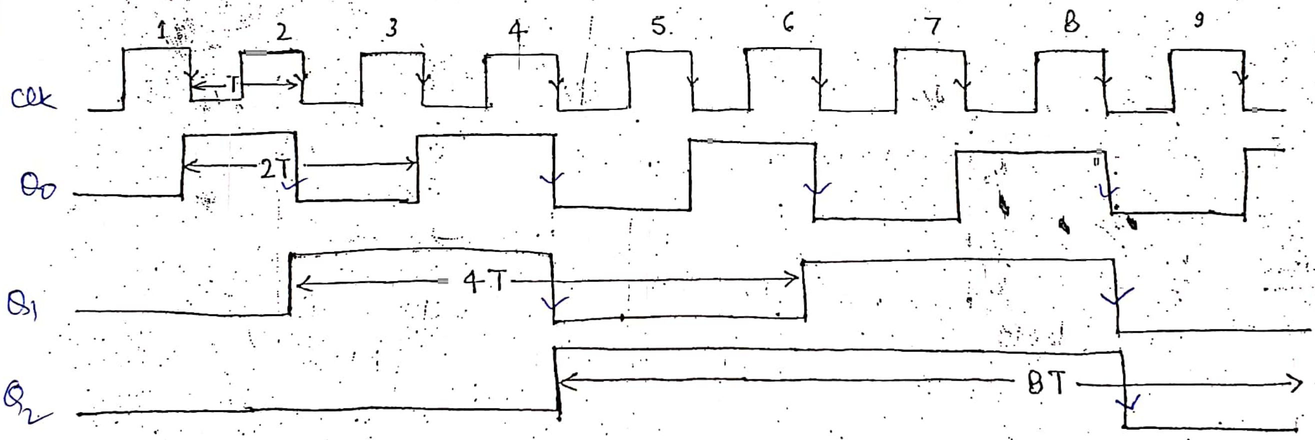
(iii) Q_2 " " Q_1 " " " $1 \rightarrow 0$

Clock	74	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
5		1	0	1
6		1	1	0
7		1	1	1
8		0	0	0

→ −ve edge triggered
→ Q_0 as clock

In n bit ripple counter max.^{m} possible states with n FF's are 2^n.   3 bit counter ⟶ 8 states

Timing diagram



In n bit ripple counter propogation delay of each ff's is tpd ff, then time period of clock

$$\boxed{\begin{aligned} T_{CLK} &\geq n\, t_{pd}\, ff \\ f_{CLK} &\leq \frac{1}{n\, t_{pd}\, ff} \\ f_{max} &\leq \frac{1}{n\, t_{pd}\, ff} \end{aligned}}$$

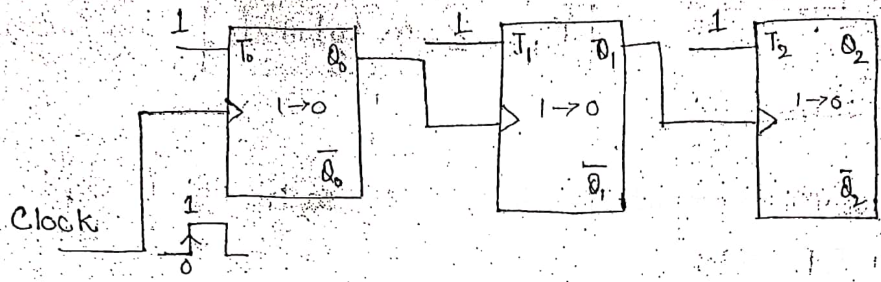
(11) 75 Ans. In ripple counters, decoding errors or transient states will present due to propogation delay. (Decoding errors will be present)

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

(v) (a) –ve edge trig. Q as clock UP counter.

(b) +ve edge trig. $\bar{Q}$ as clock UP Counter.

Down Counter



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

+ve edge trig. Q as clock Down Counter

–ve edge trig. $\bar{Q}$ as clock Down

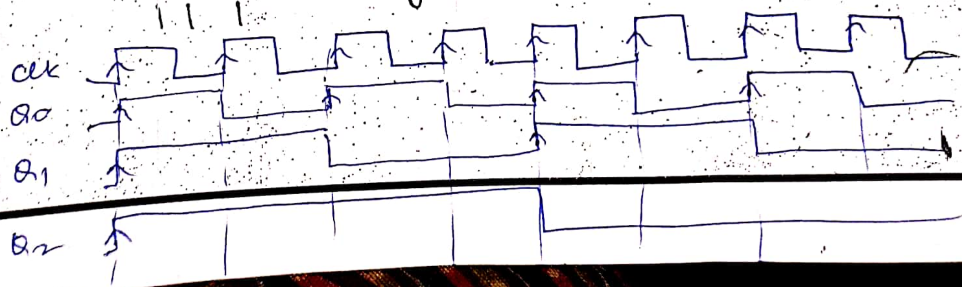
~~+ve Q~~
~~–ve $\bar{Q}$~~
~~+ve Q~~
~~–ve Q~~

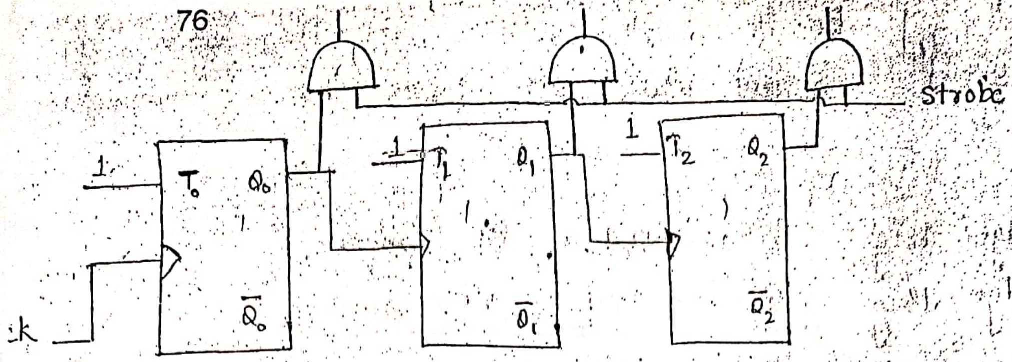
imp.

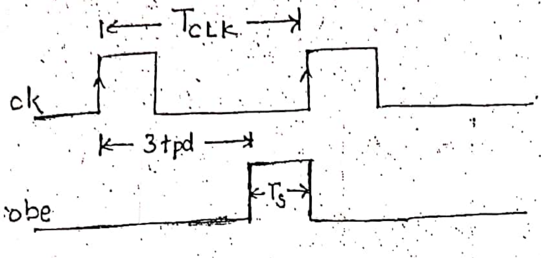
+ve	Q	Down
+ve	$\bar{Q}$	up
–ve	Q	up
–ve	$\bar{Q}$	Down

→ To avoid Decoding error, strobe signal is used

0 0 0
0 0 1
0 1 1 } → Decoding errors
1 1 1







*** $\boxed{T_{CLK} \geq n\, t_{pd} + T_s}$

In n bit ripple counter propogation deloy of each ff is tpd ff and Strobe is T_s, then

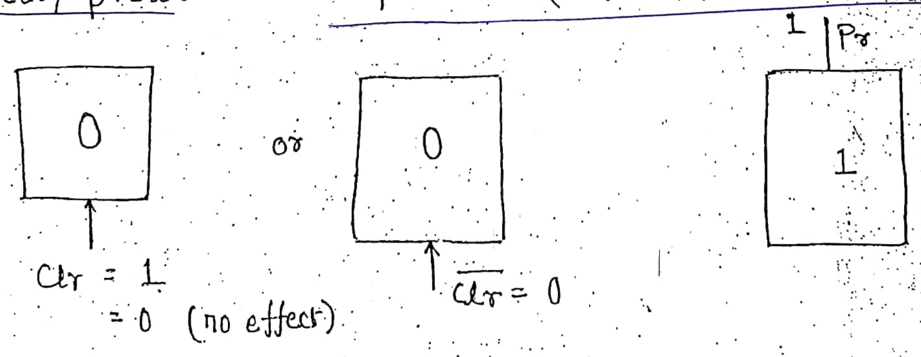
$$\boxed{T_{CLK} \geq n \cdot t_{pd} + T_s}$$

Non Binary Ripple counter

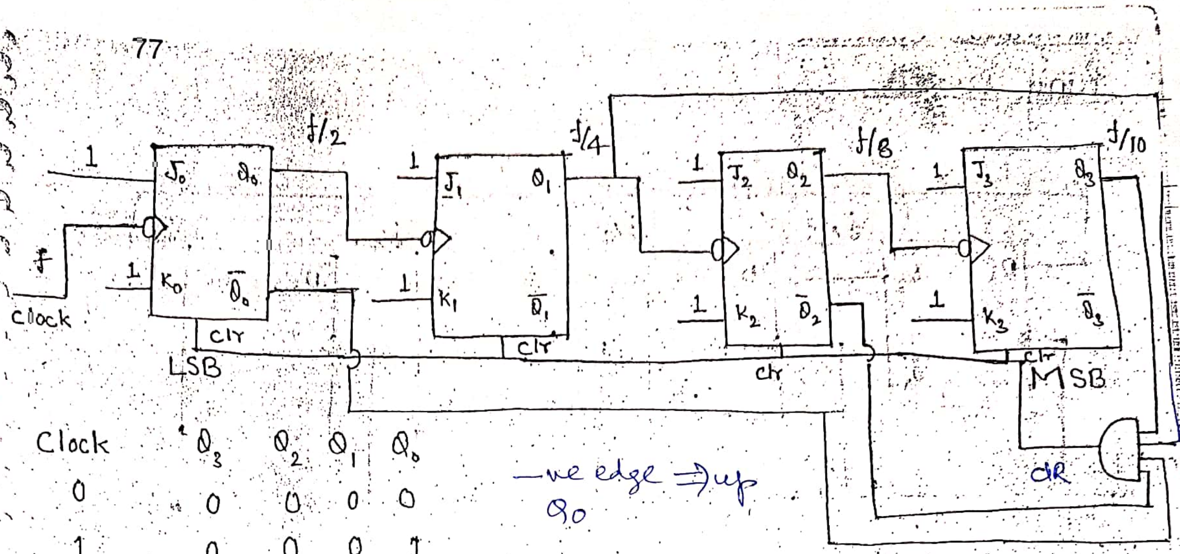
{Asynchronous inputs ⇒ clear or preset signals}

To Design Non binary counter, clear or preset signal is used

clear/ present → Asynchronous (affect o/p without having clock)

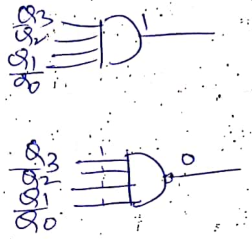


BCD counter



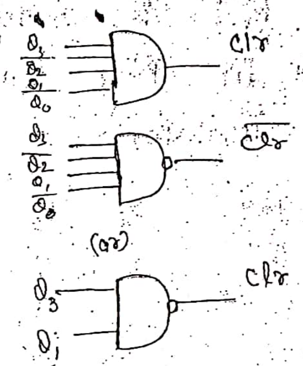
Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

−ve edge ⇒ up
Q_0



$Q_3 Q_2 Q_1 Q_0$
1 0 1 0
$Q_3 \overline{Q_2} Q_1 \overline{Q_0}$
$Q_3 \bar{Q}_2 Q_1 \bar{Q}_0$
⇐ 1010

also

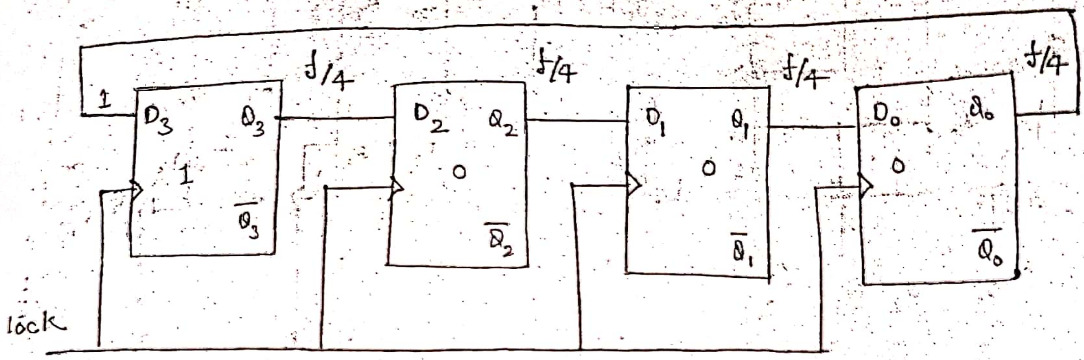


BCD counter is a Decade counter but, all Decade counter are not BCD counters.

→ Ripple Counter:

(i) Trigger < +ve, −ve
(ii) clock < Q, $\bar{Q}$
(iii) up / down
(iv) clear / present — Clr − 000, Pre − 111
(v) Observe decoding logic (Terminal Count)

Ring Counter



i) Ring counter is a shift register in this output of last FF (Q) is connected to 1^{st} FF input.

- In Ring counter only one FF output is high and it circulates among all FF based on clock pulse.

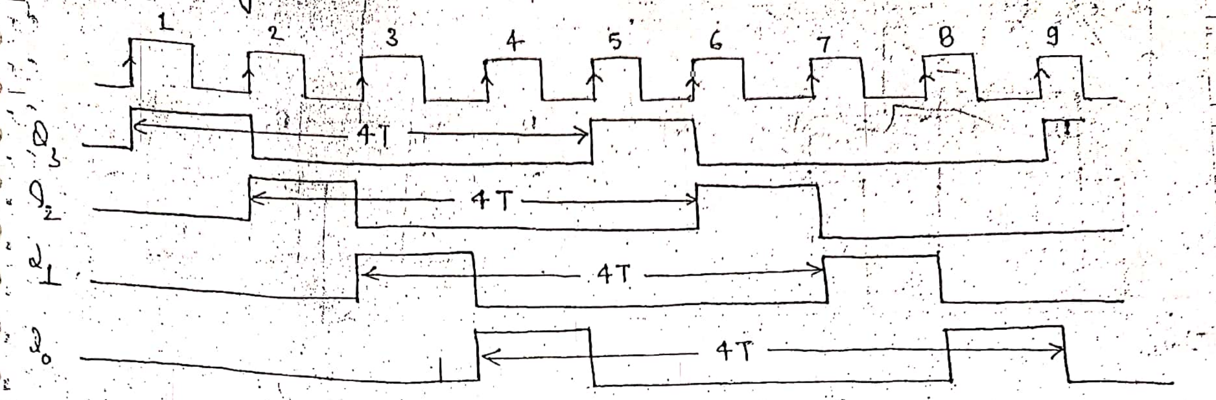
Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
	0	0	0	1
9	1	0	0	0

ii) In Ring counter with n flip flops, maximum possible states are n.

$2^n - n$

79

Timing diagram



→ $f \rightarrow \frac{f}{n}$

→ $\theta = \frac{360}{n}$

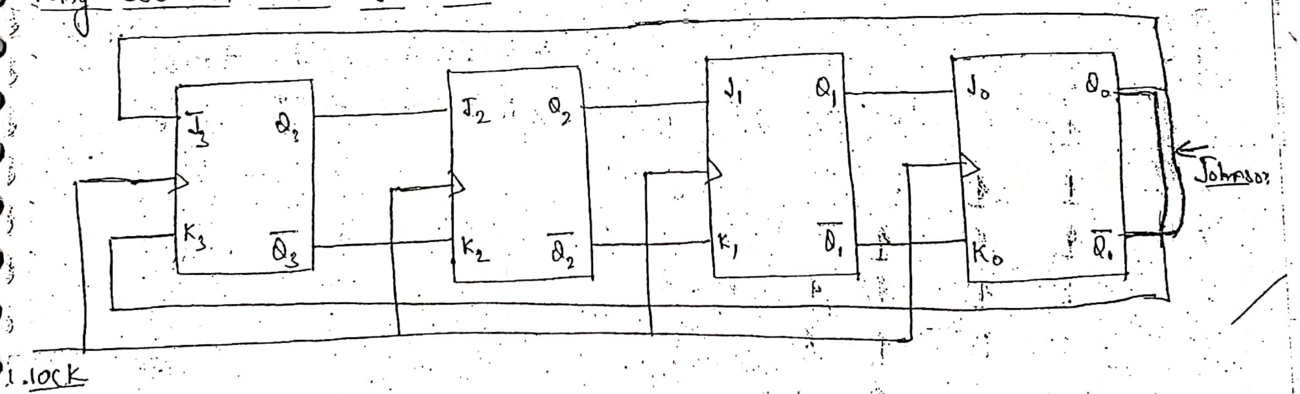
(iv) Ring counter is used in SAR (successive approx. register) type ADC

(v) Also used in stepper motor

(vi) To generate waveform

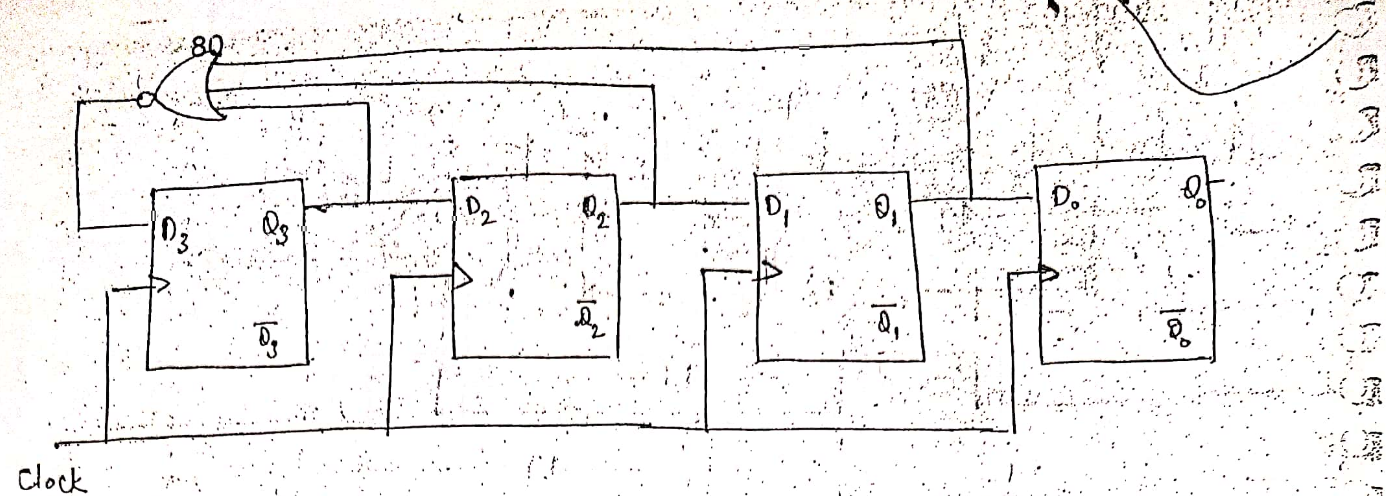
→ No. of unused states present in ring counter is $2^n - n$.

Ring counter with JK FF

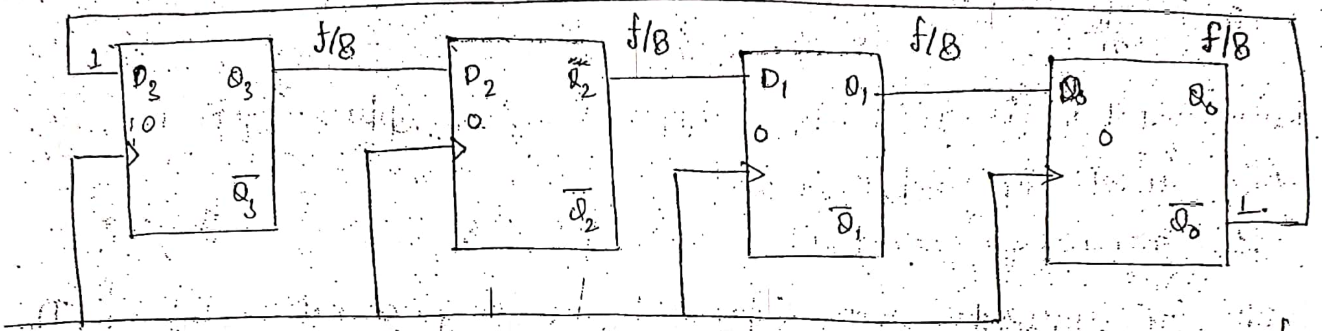


→ Advantage of Ring counter is its decoding is easy.

Self starting Ring Counter (To avoid Lock out)



Johnson Counters : (Twisted ring counter) or Switch tail Counter (or) Creeping Counter (or) Mobies Counter (or) Walking counter



Clock	Q_3	Q_2	Q_1	Q_0	Decoding logic
0	0	0	0	0	$\bar{Q}_3 \bar{Q}_0$
1	1	0	0	0	$Q_3 \bar{Q}_2$
2	1	1	0	0	$Q_2 \bar{Q}_1$
3	1	1	1	0	$Q_1 \bar{Q}_0$
4	1	1	1	1	$Q_3 Q_0$
5	0	1	1	1	$\bar{Q}_3 Q_2$
6	0	0	1	1	$\bar{Q}_2 Q_1$
7	0	0	0	1	$\bar{Q}_1 Q_0$
8	0	0	0	0	$\bar{Q}_3 \bar{Q}_0$
~~9~~	~~1~~	~~0~~	~~0~~	~~0~~	$\cancel{Q_3 \bar{Q}_2}$

→ if f is input freq. $f \rightarrow \frac{f}{2n}$ o/p freq.

→ 4 ff → 8 States

→ n ff → $2n$ states

> No. of unused ~~use~~ states in Johnson counter is $2^n - 2n$

Scanned with CamScanner

① J, K, Q_n | Q_{n+1}
② Q_n Q_{n+1} | J K

Flip flop conversion

(I) Identify available & required flipflop.

(II) Characteristic table for required flip flop.

(III) Excitation table for available flip flop.

(IV) Boolean expression for available flip flop.

(V) Draw ckt. diagram.

JK to D

available flip flop = JK

Reqd. flip flop = D

(I) char. table of D

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

(I) Excitation table for available flip flop = JK

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

(II) Boolean expression

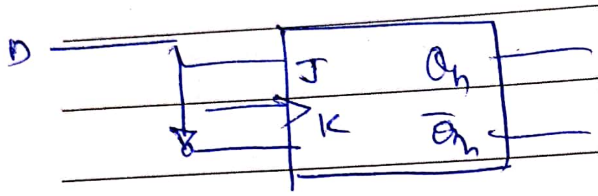
D	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

D \ Q_n		
	0	X
	1	X

D \ Q_n		
	X	1
	X	0

For J $\boxed{J = D}$ For K $\boxed{K = \overline{D}}$

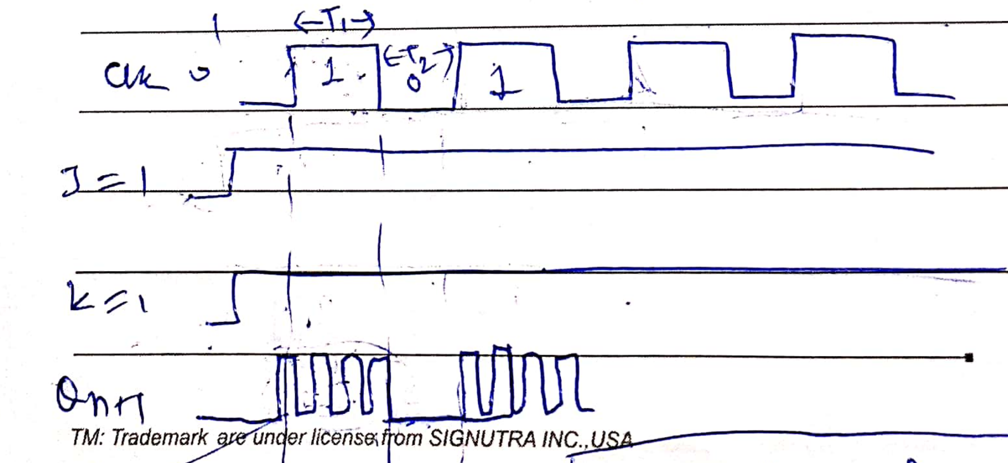
(V) Diagram



(VI) Race Around condition

CLK	J	K	Q_{n+1}	$\bar{Q}_{n+1}$
0	X	X	Q_n	~~$\bar{Q}_n$~~
1	0	0	Q_n	~~$\bar{Q}_n$~~
1	0	1	0	~~1~~
1	1	0	1	~~0~~
1	1	1	$\bar{Q}_n$	Q_n

Q_n
$\bar{Q}_n$
Q_n
$\bar{Q}_n$
Q_n
$\bar{Q}_n$
Q_n
$\bar{Q}_n$



delay of flip flop Racing

$\boxed{T_1 > \text{delay of JK flip flop.}}$

$\boxed{T/2 > \text{delay of flip flop}}$

time period of clock

Conditions to overcome Racing

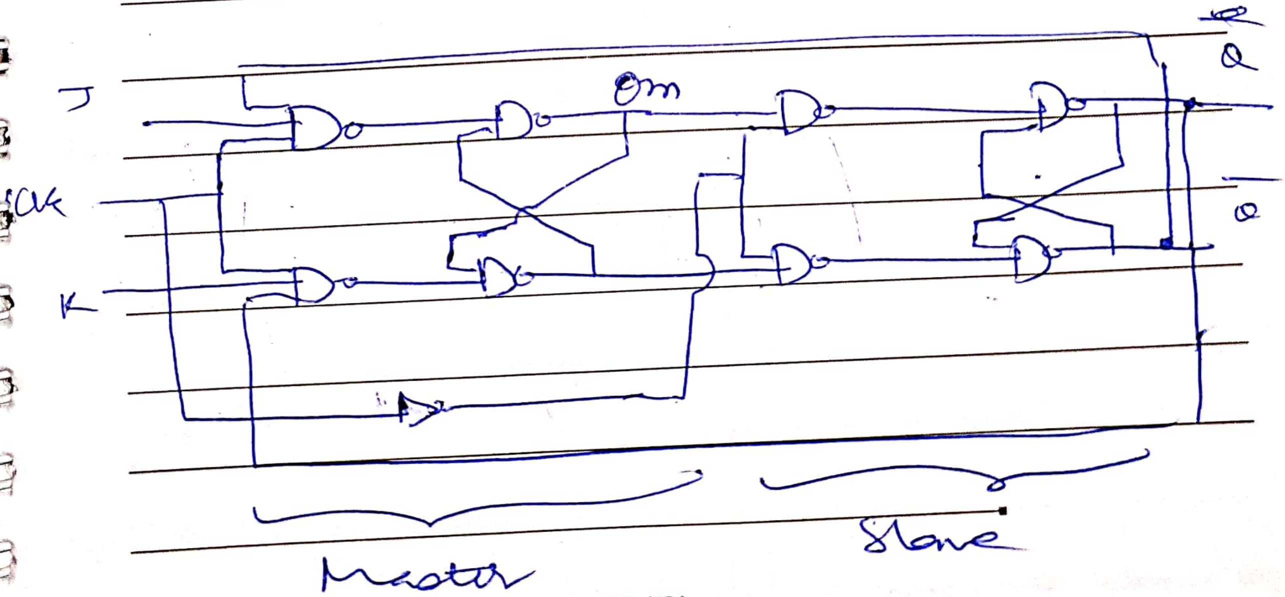
(i) $T/2 <$ prop. delay of flip flop

(ii) edge triggering

Imp (iii) Master slave

Master slave flip flop (JK)

(i) same as negative edge triggering flip flop clocking



clk = 1

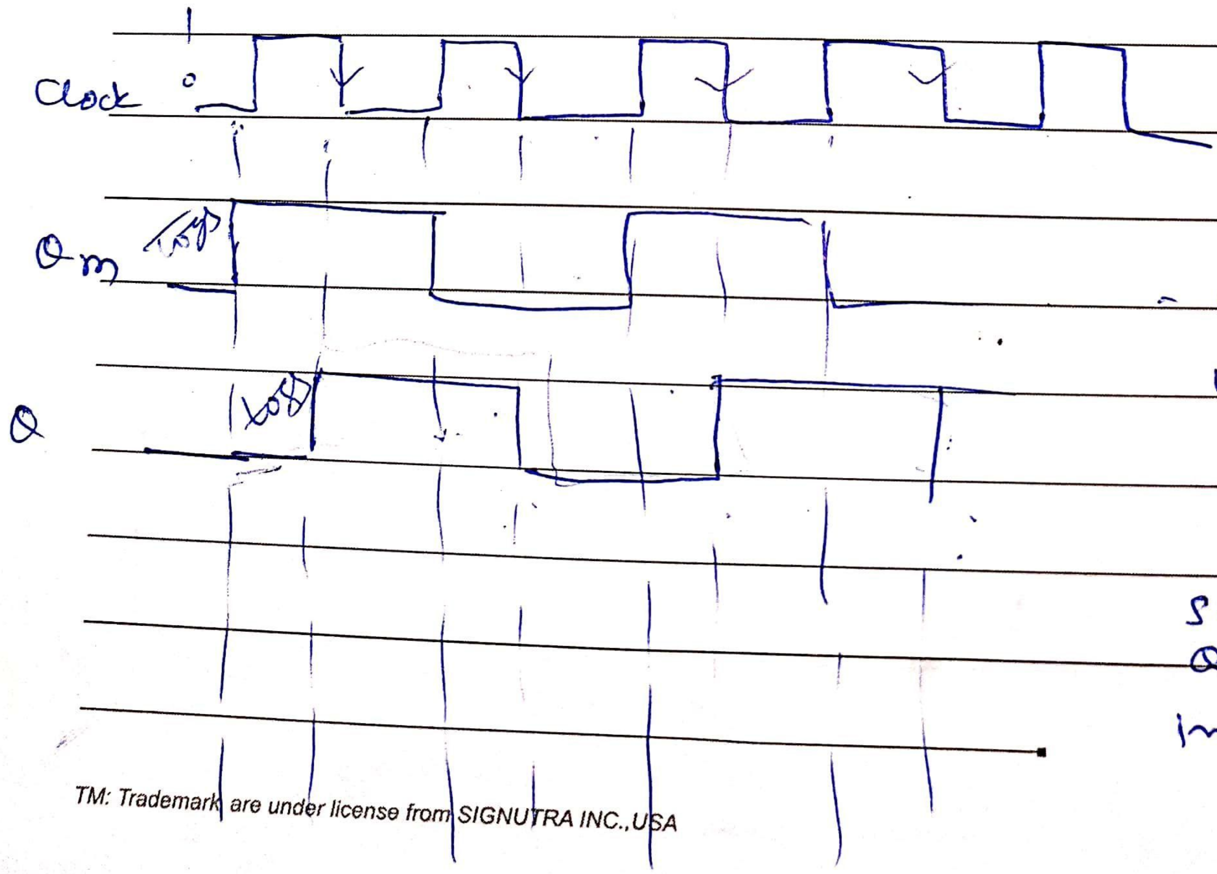
Master = operational

clk = 0

slave = functional

- Instead of o/p changing continously, the output changes once in a cycle.

- Say, J = K = 1



- Assume Q_m was low initially.

Say Q = 0 initially