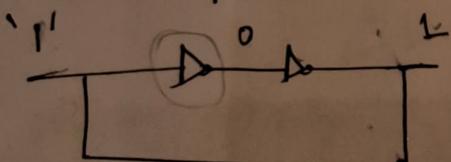
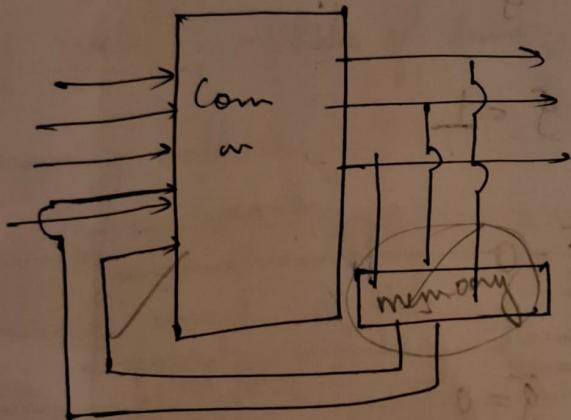


Assignment — Draw the logic circuit for BCD to decimal converter.

Interrupt request \rightarrow priority encoder

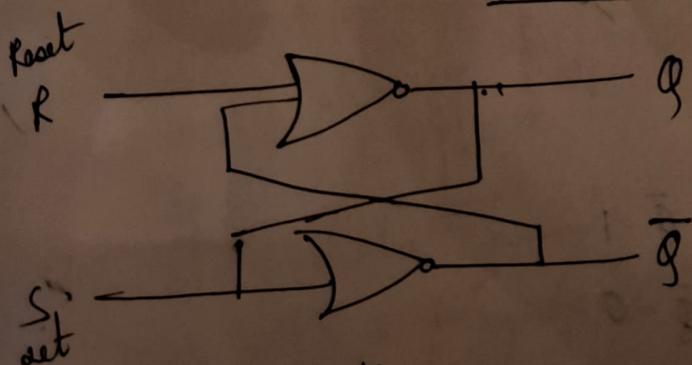
Sequential circuit

The present output depends on the present input as well as past ~~out/past~~ outputs / inputs.



cascaded NOT GATE

SR LATCH

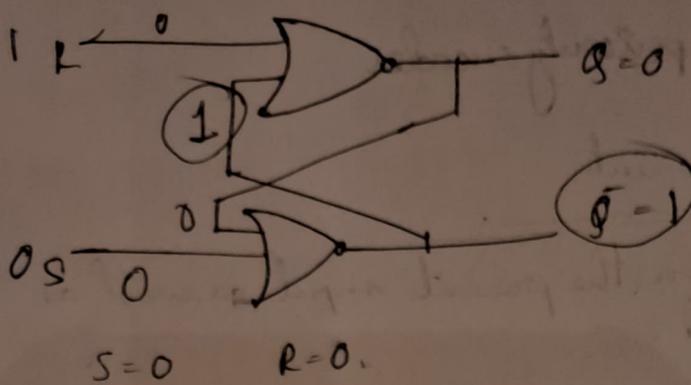


The basic storage element is called latch. It latches 0 on 1

Reset $Q=0$
set $Q=1$

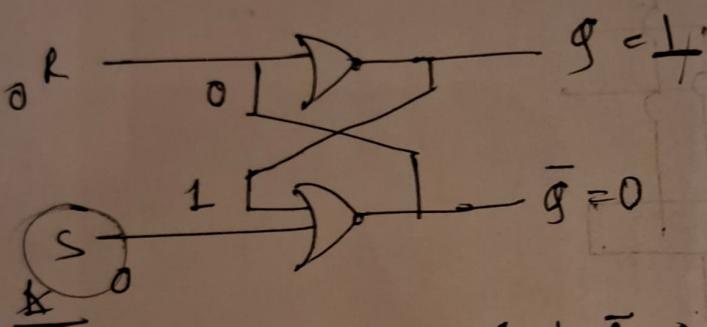
A	B	Y
0	0	1
0	1	0
1	0	0

Case 1 $S=0$ $R=1$, $Q=0 \rightarrow \bar{Q}=1$



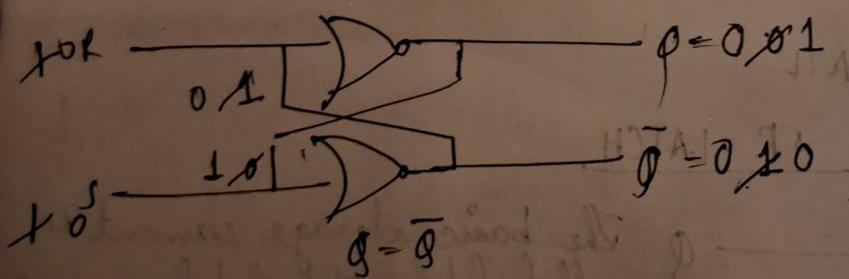
$$S=0 \quad R=0.$$

Case 2 : $S=1$ $R=0$ $Q=1 \rightarrow \bar{Q}=0$



$$S=0 \quad R=0 \quad Q=1 \quad \bar{Q}=0$$

Case 1 : $S=1$ $R=1$, $Q=0, \bar{Q}=0$

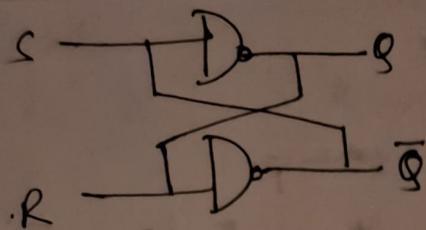


not used.

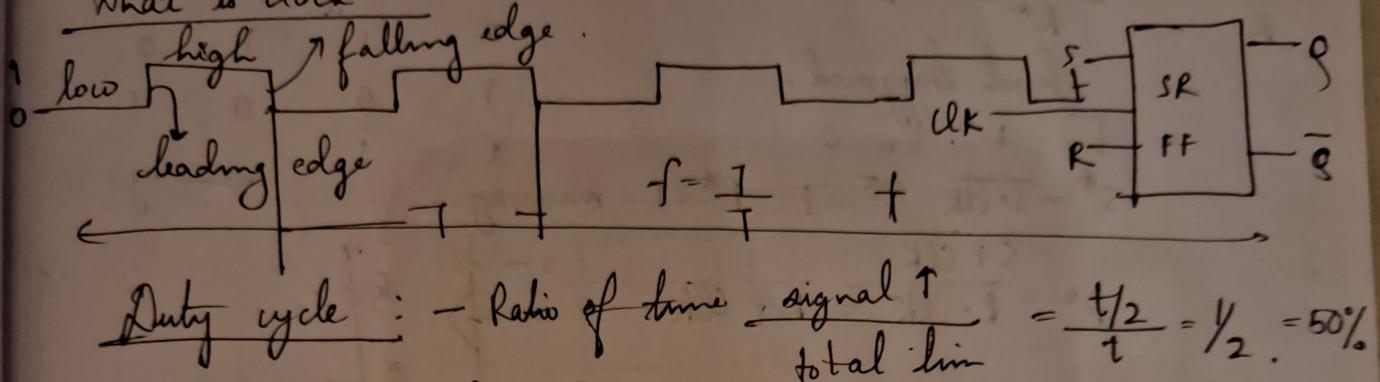
$$S=0 \quad R=0, \quad Q=0, \quad \bar{Q}=1$$

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

S	R	Q	\bar{Q}
0	0	memory	
0	1	0	1
1	0	1	0
1	1	Not used	

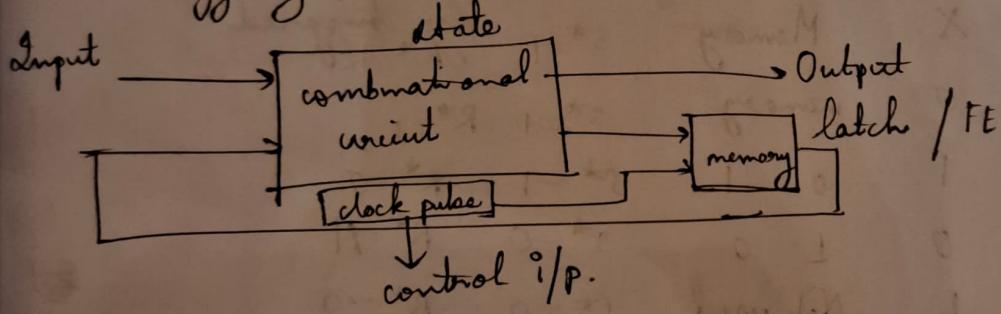


What is clock



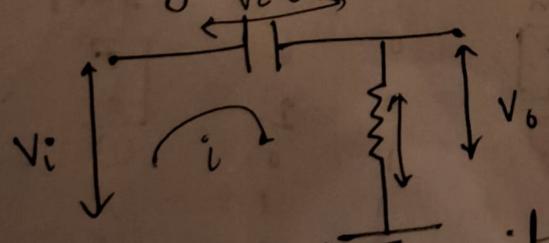
Duty cycle : - Ratio of time $\frac{\text{signal } \uparrow}{\text{total time}} = \frac{t/2}{T} = \frac{1}{2} = 50\%$

Triggering methods.



edge - \oplus edge \ominus edge

Edge triggering



Differentiator = circuit

value of C & R is very low.

$$V_i - V_o + V_o = 0$$

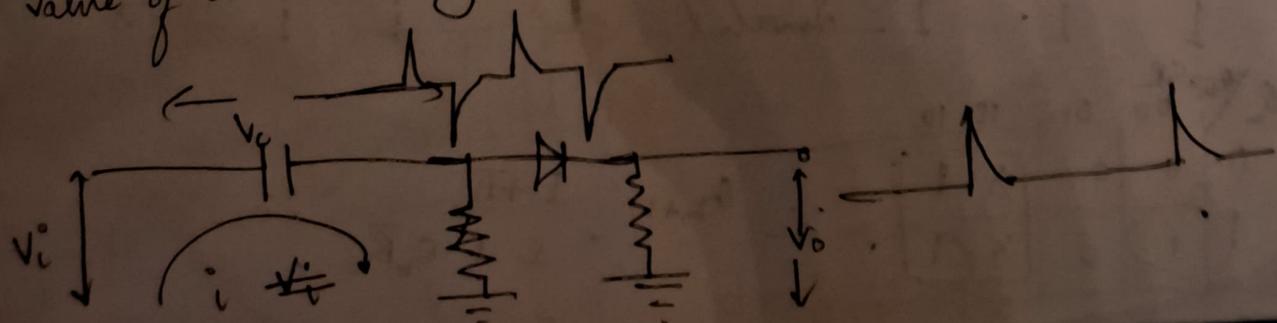
$$V_i = V_o + V_o$$

$$Q = CV_C$$

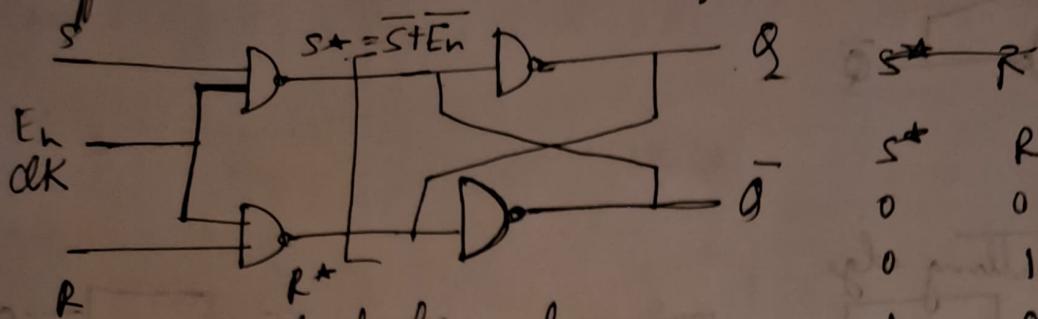
$$\Delta t = C V_C \quad \frac{dV_o}{dt} = \frac{1}{RC} V_o$$

$$V_C = \frac{1}{C} \int dt \quad \frac{dV_o}{dt} = \frac{1}{RC} V_o$$

$$Rt = V_o - R \frac{dV_o}{dt}$$



Difference between latch and flip flop.



level triggered.

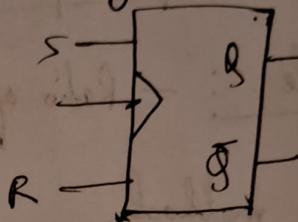
$$E_n = 1 \cdot E_n = 0$$

$$S^* = \overline{S \cdot \text{CLK}} \quad S^* = 1 \quad R^* = 1 \quad \text{memory}$$

$$= \overline{S} + \overline{\text{CLK}}$$

$$R^* = \overline{R \cdot \text{CLK}} = \overline{R} + \overline{\text{CLK}}$$

CLK	S	R	Q	\bar{Q}
0	X	X	Memory	
1	0	0	memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not used	



$S^* = 1$ edge triggered.
 $R^* = 1$

CLK	S	R	Q	\bar{Q}
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Truth Table next state characteristic

CLK	S	R	Q_{n+1}	Q_n	Q_m	Characteristics		Q_{n+1}
						S	R	
0	X	X	Q_n	0	0	0	0	$\begin{matrix} 0 \\ 0 \end{matrix}$
1	0	0	Q_n	0	0	1	0	$\begin{matrix} 1 \\ 0 \end{matrix}$
1	0	1	0	0	0	1	1	$\begin{matrix} X \\ 1 \end{matrix}$
1	1	0	1	1	1	0	1	$\begin{matrix} 1 \\ 0 \end{matrix}$
1	1	1	Invalid.	1	1	1	1	$\begin{matrix} X \\ X \end{matrix}$

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

$$Q_{m+1} = \frac{I + II}{S + Q_m R}$$

Exita

Q_n
0
0
1
1

T.R
clk
0
1
1
1
1
1
0

J
K
R
K'

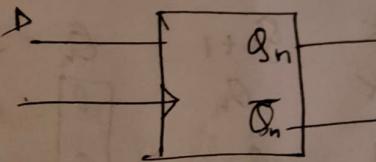
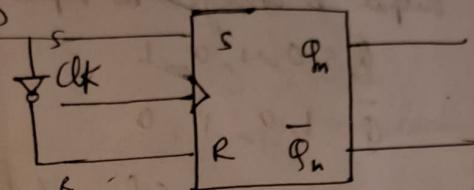
Excitation Table

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

D flip flop

T-T for SR flip flop

clk	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



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

$S=0 \quad R=1$
 $S=1 \quad R=0$

Excitation Table

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

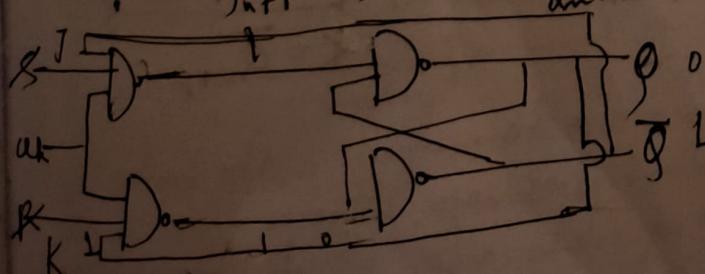
Char table

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

Excitation Table

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

$Q_{n+1} = D$ Introduction + JK flipflop.

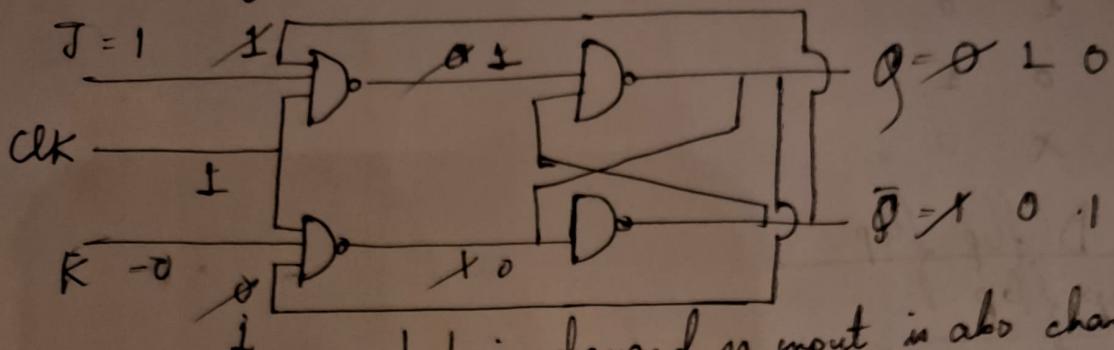


$JK = 0 -$ memory
 $JK = 1 \quad J = 1, K = 0, Q = 1, \bar{Q} = 0$

$$clk = 1 \quad J=0 \quad K=1 \quad Q=0 \quad \bar{Q}=1$$

$$clk = 1 \quad J=1 \quad K=1$$

assume $Q=0 \quad \bar{Q}=1$



output is changed so input is also changed.

$$Q = 0, 1, 0, 1$$

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

Truth Table

clk	J	K	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	0	1	1
1	1	1	$\overline{Q_n}$ (Toggle)
1	1	1	0

Char Table

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

Excitation Table

On Int + JK	0 0 0 X	0 1 2 X
0	0 1 2 X	0 1 0 X
1	1 0 0 X	1 1 1 X
X	X 0 0 X	X 1 1 X

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

Q_n	00	01	11	10
J	0	0	1	1
K	0	0	0	1

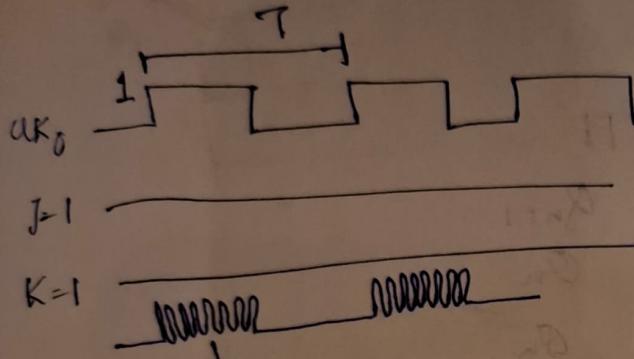
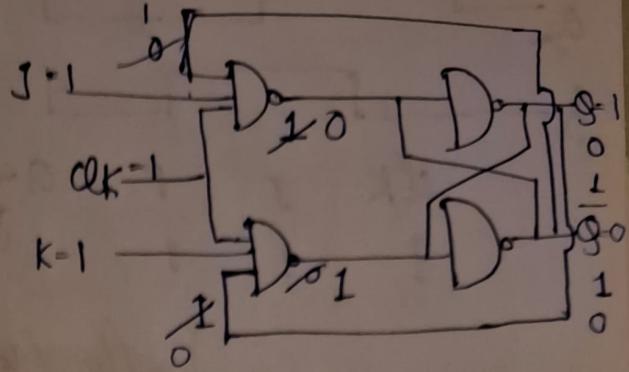
$$Q_{n+1} = \bar{Q}_n J + Q_n \cdot K$$

if

Q_n	0	1
J	X	X
K	1	0

$$K = \overline{Q_{n+1}}$$

Clk J K On Q_{n+1} \bar{Q}_{n+1}



$$T/2 > \text{Delay}$$

Delay $> T/2 \rightarrow \text{clock will be low}$

Racing

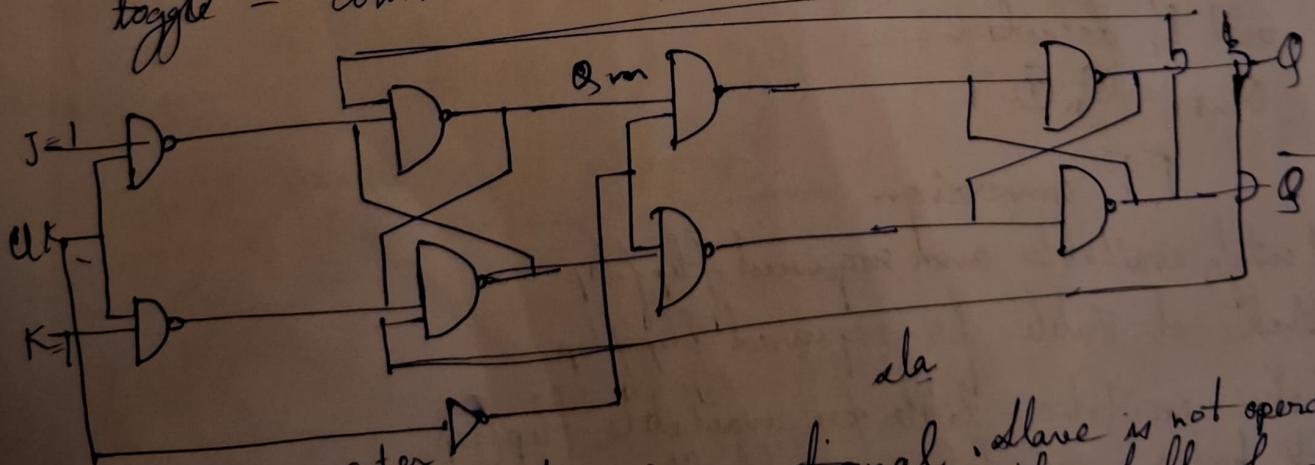
Conditions to overcome racing.

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

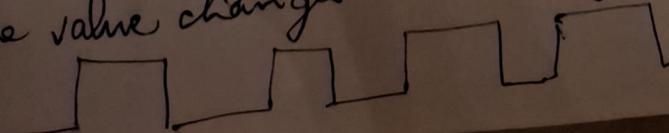
(ii) edge triggering

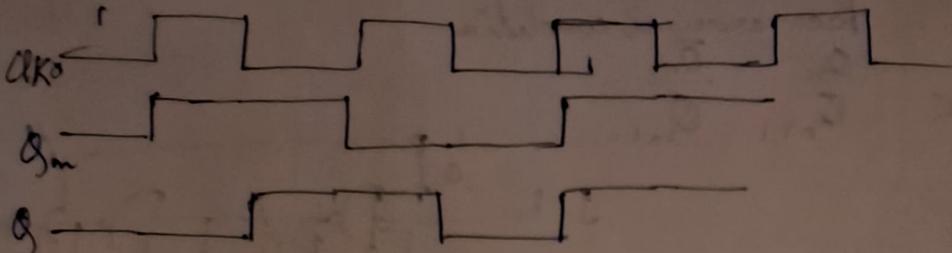
(iii) master-slave (-ve edge triggering)

Racing - uncontrolled phenomena.
Toggle - controlled



when Clk is high master is operational. Slave is not operational
so it will keep the memory. When vice versa, the feedback does not work. So the value changes one in a cycle.

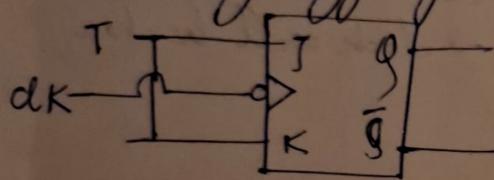




clk J K Q_{n+1}
 1 1 1 \bar{Q}_n Toggling

T flip flop

The Only Toggling action



T.T for TFF
 clk T Q_{n+1}
 0 X Q_n
 1 O Q_n
 $J=0 \quad K=0$

1 1 \bar{Q}_n (Toggling)

On Table

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

Excitation Table

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

odd 1's detector

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

FF conversion

- ① Identify available and required flip flop.
- ② Make On Table for required flip flop
- ③ Make excitation table for available flip flop
- ④ Write boolean exp for available ff
- ⑤ Draw the circuit

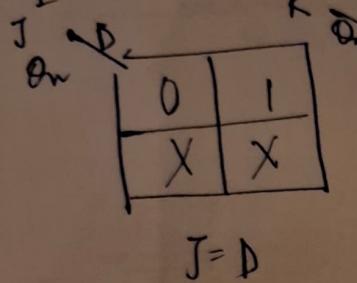
$JK \rightarrow D$ flip flop

aw ff = JK

req ff = D

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

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



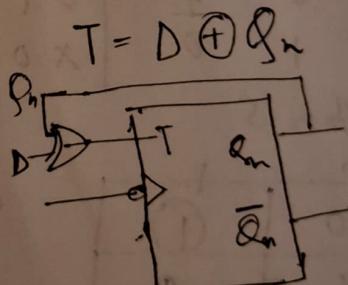
~~T flip flop to D flip flop Conversion~~

aw. FF = T

req FF = D.

Ch Table

Q_n	D	Q_{n+1}	Excitation FF		
			T	Q_m	Q_{n+1}
0	0	0	0	0	0
0	1	1	1	0	1
1	0	0	1	1	0
1	1	1	0	1	1



~~SR flip flop to JK flip flop~~

aw SR

req JK

Ch Table

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

Excitation Table

Excitation Table

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

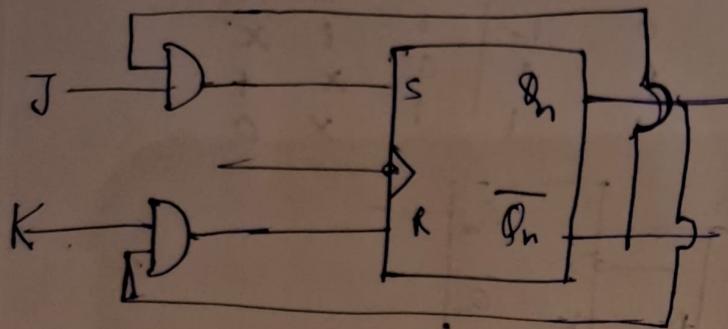
Q_n	JK	for S
0	00 01 11 10	0 0 1 1
1	X 0 0 X	X 0 1 X

for R

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

$$S = \overline{Q_n} J$$

$$R = Q_n K$$



↗ JK to SR
 ↗ T to SR

Ch Table SR FF			to T FF		
Q_n	T	Q_{n+1}	S R	Q_n	Q_{n+1}
0	0	0	0 X	0	0 X
0	1	1	1 0	0	0 0
1	0	1	0 1	0	1 0
1	1	0	X 0	1	0 1

Excitation Table

for S

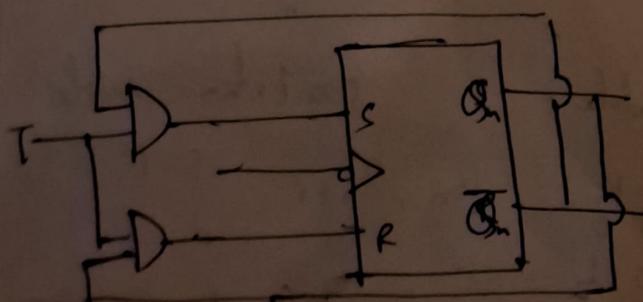
Q_n	T	0 1
0	0	0
1	X	0

$$S = \overline{Q_n} T$$

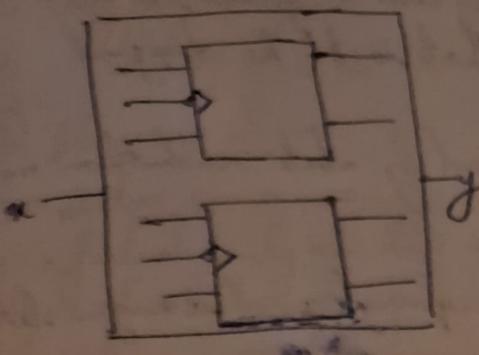
for K

Q_n	T	0 1
0	X	0
1	0	0

$$R = Q_n T$$



State Diagram

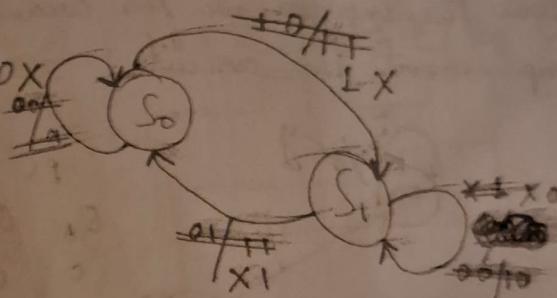


State Table

$P = S$	x	$N = S$	y
$Q_n \quad Q_n$			
0 0	1	1 0	1
$S_0 = Q_n Q_n$			
$S_1 = 0 1$			
$S_2 = 1 0$			
$S_3 = 1 1$			



Q_n	J	K	Q_{n+1}	\bar{Y}_P
1	0	0	0	
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	1	$S_1 = 1$
1	0	1	0	$S_0 = 0$
1	1	0	1	
1	1	1	0	



State eq

$$LHS = RHS$$

$$Q_{n+1} = P S \& \bar{J} P$$

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

Design

Design Procedure for clocked sequential circuit

Step 1: A state diagram or timing diagram is given, which describe the behaviour of the circuit that is to be designed.

Step 2: Obtain the state table

Step 3: The no of states can be reduced by state reduction meth.

Step 4: Do state assigned

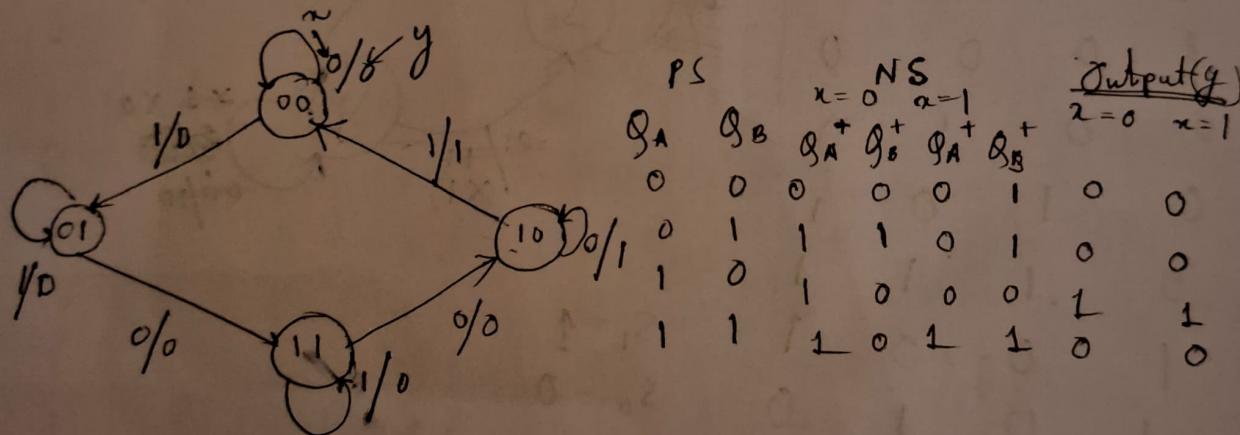
Step 5: Determine the no of flip-flops required and assign left symbols

Step 6: Decide the type of flip-flop to be used

Step 7: Draw the circuit excitation table from state table

Step 8: Obtain the expression for circuit output and flop flop input

Step 9: Implement the circuit.



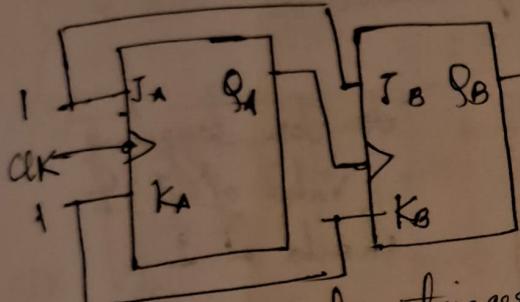
Synchronous

- ① Easy to design
- ② clocked ff act as memory
- ③ they are slower
- ④ status of memory element is affected only at the active edge of clock, if input is changed.

A synchronous

difficult to design
unlocked ff or time delay element is used as memory element faster as clock is not present
The status of memory element will change any time as soon as input is changed.

Counters

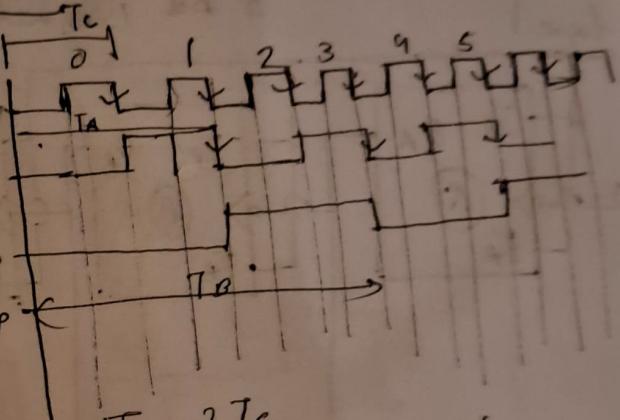


→ edge triggered flipflop
→ edge so toggling

↑ no of flipflop $J=1 \ K=1$, ϕ
2 P → frequency will be divided by 16.

CLK	Q _B	Q _A
0	0	0
1	0	1
2	1	0
3	1	1

from 4 it repeats.

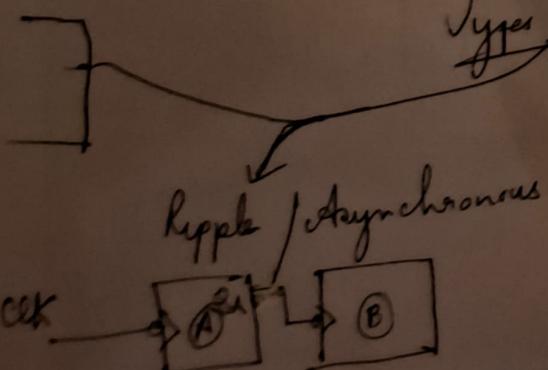


$$T_A = 2T_C \Rightarrow f_A = \frac{f_c}{2}$$

$$T_B = 2T_A$$

$$f_B = \frac{f_A}{2} = \frac{f_c}{4}$$

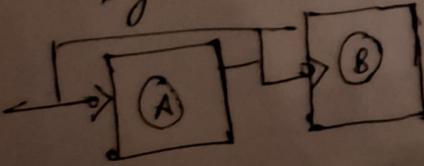
Types of Counters



ripple / asynchronous

Up
Down
up/down

synchronous.



Asynchronous counter

1. If are connected in such a way that o/p of first ff drives the clock of the next ff.

2. ff are not clocked simultaneously ff are clocked simultaneously.

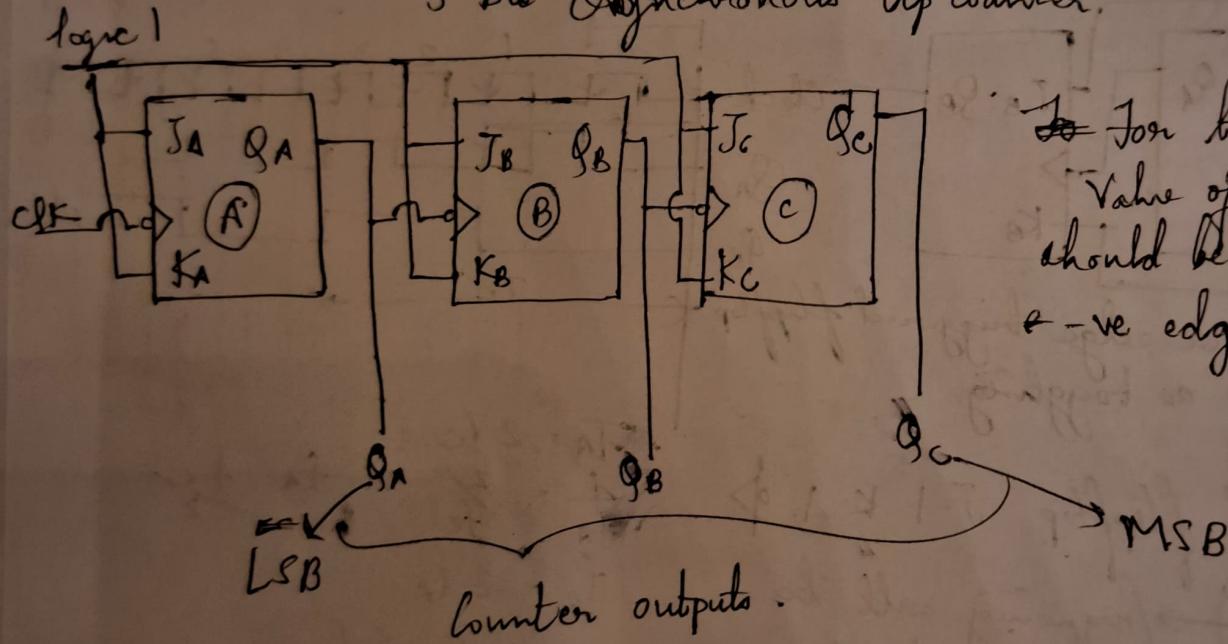
3. Circuit is simple for more no of states circuit becomes complicated as no of states increases.

4. Speed is slow as clock is propagated through no of stages speed is high as clock is given at a same time.

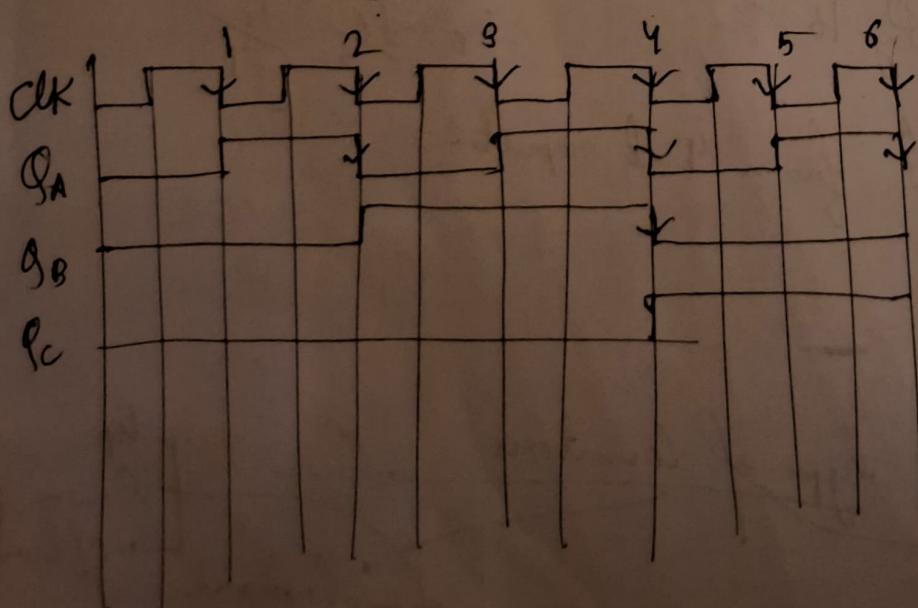
Synchronous

There is no correction of let between o/p of first ff and clock of next ff.

3 Bit Asynchronous Up counter.



To For toggling
Value of J, K
should be 1.
e - ve edge triggering



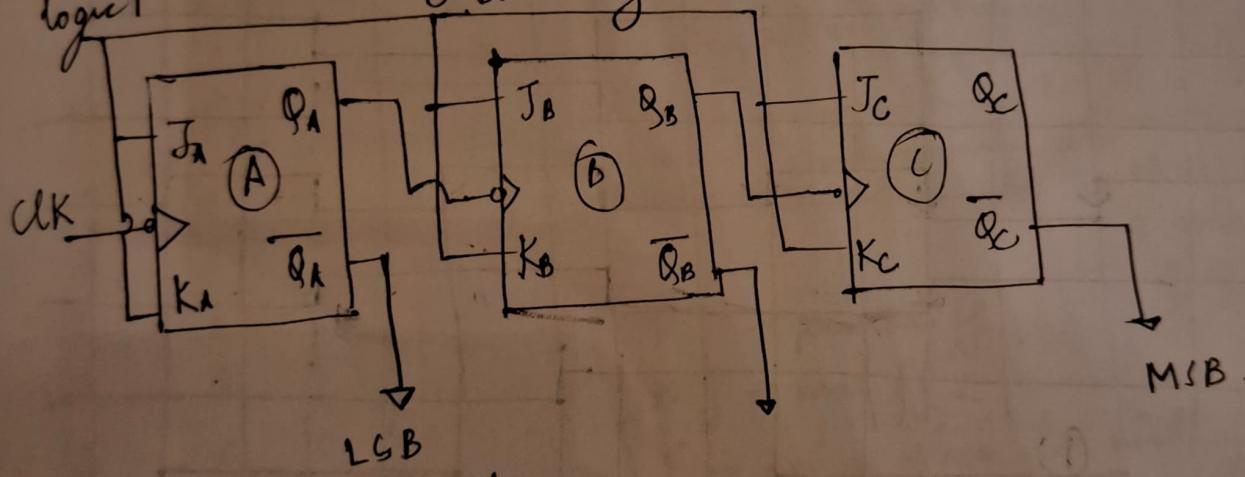
$$J_A = K_A = 1 \text{ (Jogging)}$$

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

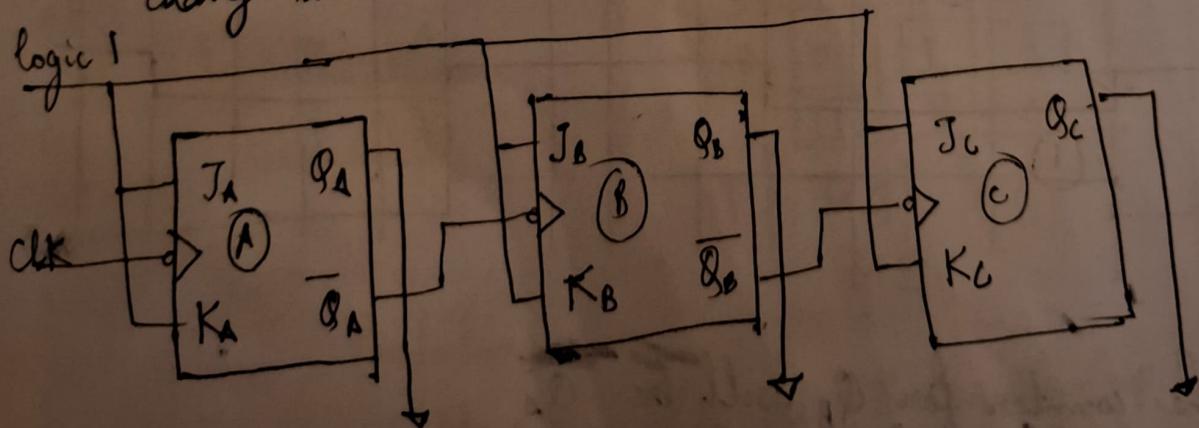
Clock	Q_2	Q_1	Q_0	Decimal Eq
Initially	0	0	0	0
1st	0	0	1	1 8 states
2nd	0	1	0	2 2^n
3rd	0	1	1	3. $n = \text{no of flipflop}$
4th				
7th	1	1	1	7
8th	0	0	0	0

4 bit Asynchronous Up counter

3 bit Asynchronous Down Counter



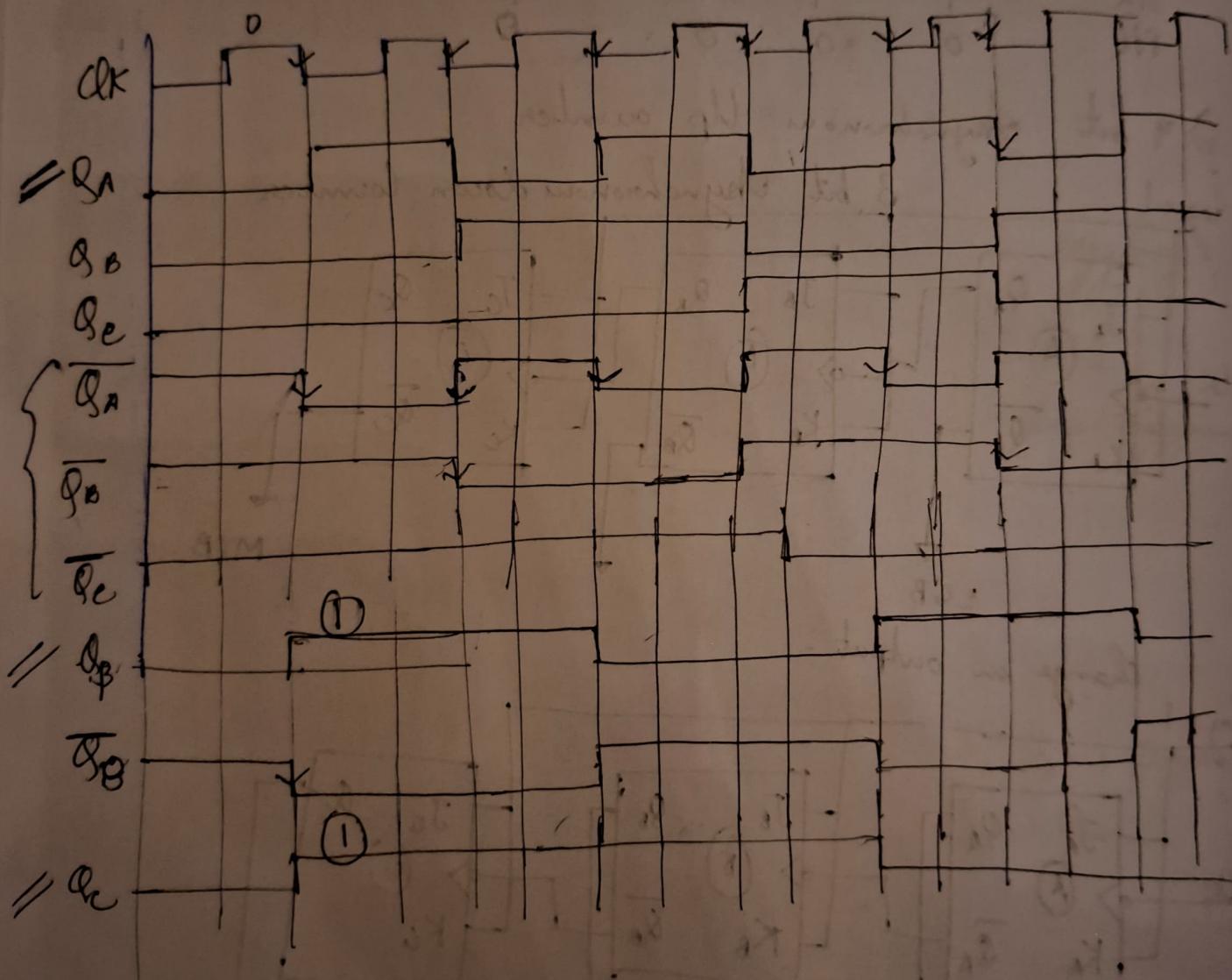
change in output.



CLK \bar{Q}_0 \bar{Q}_1 \bar{Q}_2 Decimal

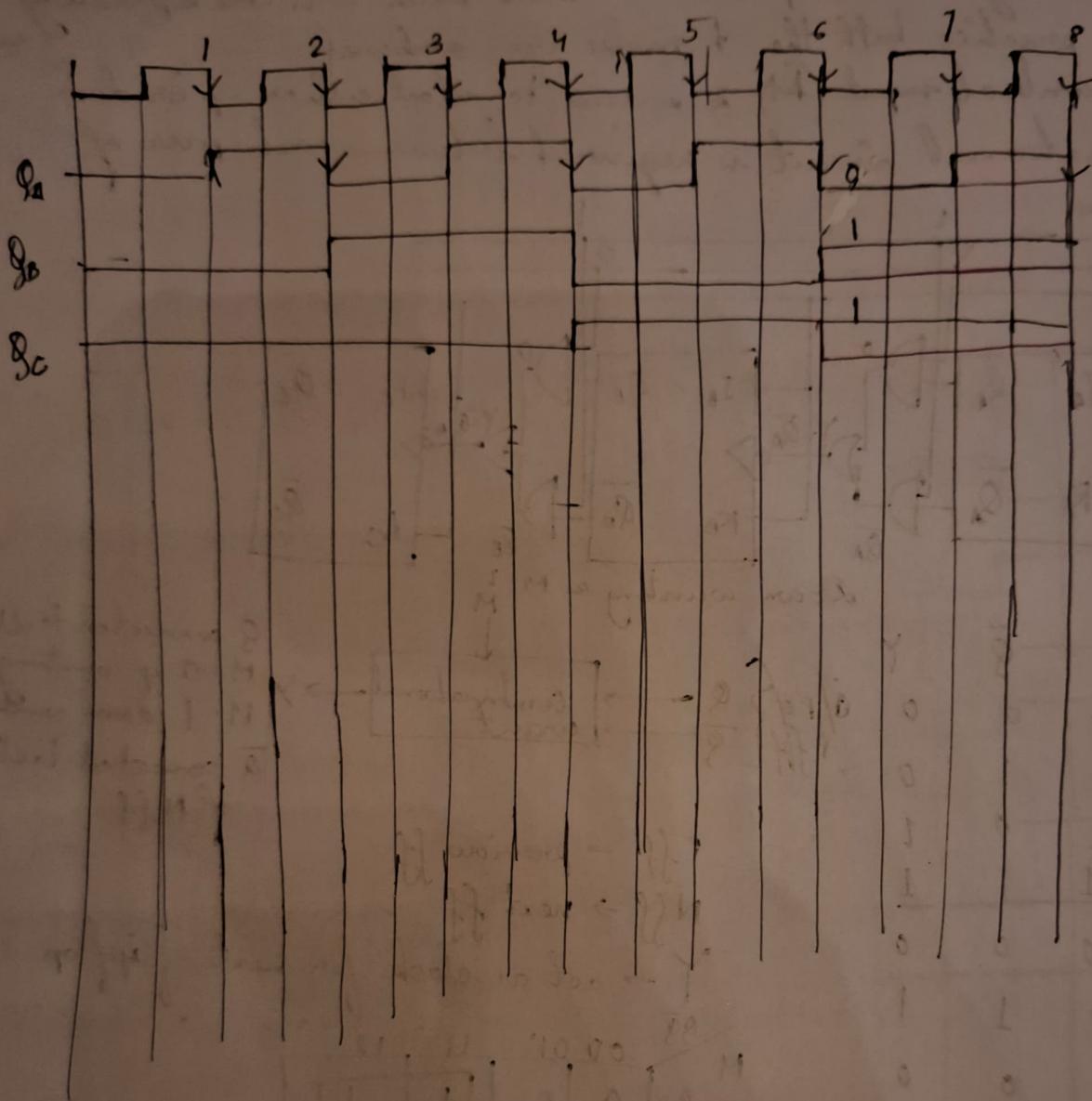
Initially	1	1	1	7
1st	1	1	0	
2nd	1	0	1	

7th 0 0 0 0



~~→ →~~ The counter for \bar{Q}_2 will be \bar{Q}_3

Modulus of counter represents the no of states the counter is having.
how many counts the counter is doing.



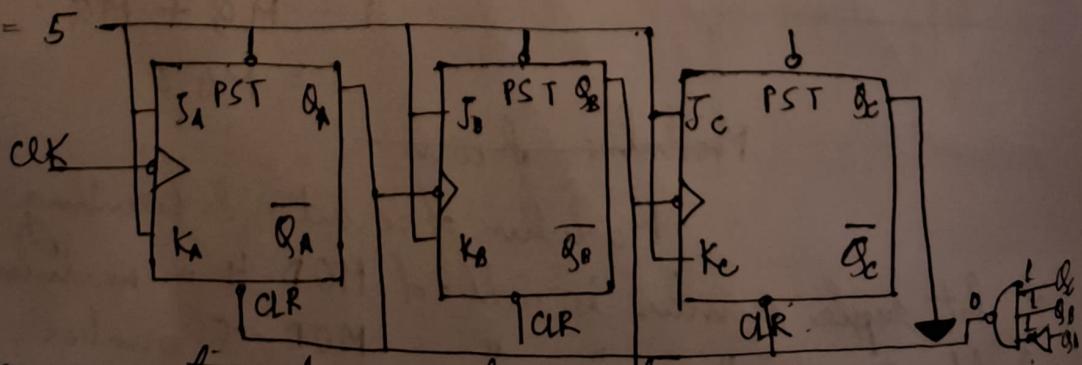
Ex - MOD 6 counter using MOD-8 counter (ripple)

States = 6. MOD 8 counter 3 bit ripple counter

max count = 5

000 (1)
001 (2)

101 (5)

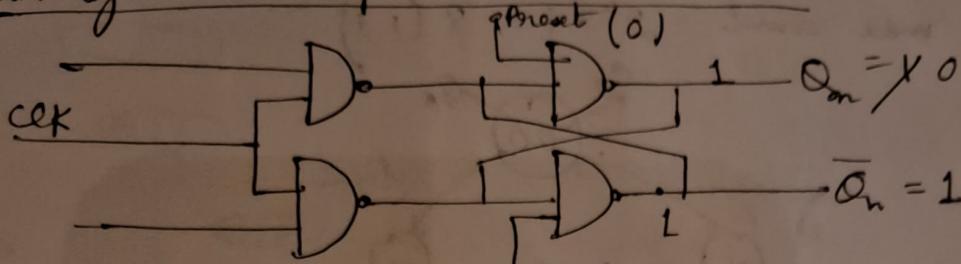


Preset and clear are active low signal ie whenever they are low
Whenever they are low the ff will be overriding

Preset and Clear Inputs in FF

They are direct or over overriding inputs in asynchronous inputs.

Other synchronous inputs are S P J K D T

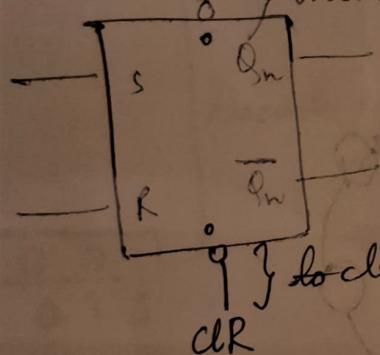


preset = 0 $\rightarrow Q_n = 1$ CLR (0)

clear = 0 $\rightarrow Q_n = 0$ as $\bar{Q}_n = 1$

whatever be the value of clock and synchronous inputs.

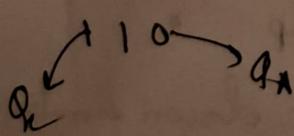
\overline{PR} } indicates active low signal.



} so clear is low

	Reset	Clear	On
0	0	0	Not used
0	0	1	1
1	1	0	0
1	1	1	ff will perform normally

$$\begin{bmatrix} PST = 0 & Q = 1 \\ CLR = 0 & Q = 0 \end{bmatrix}$$



As soon as counter 6 is encountered the clock is reset.

How to make the counter count till 5. There are 2 more state 110 (6) and 111 (7). How to skip these states. As soon as 6 is arrived we need to reset our counters.

Its clear is zero, the output is 0 and Q_A NAND Q_B NAND $\bar{Q}_A = 0$

As soon as counter 6 is encountered the clock is reset.

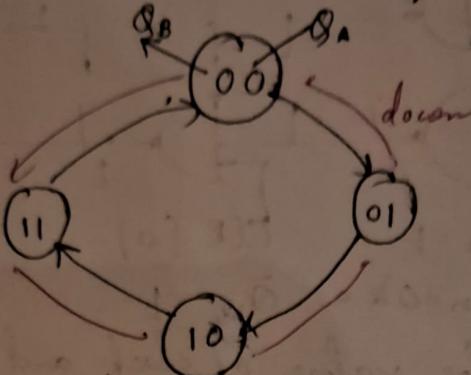
State diagram of a Counter

2 bit up counter 2 ff (JK or T)

$$Q_B \ Q_A \quad \text{max count} = 2^n - 1 = 3 \ (11)$$

0	0
0	1
1	0
1	1

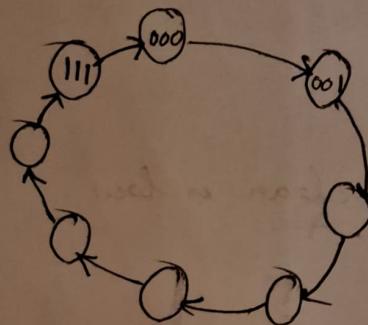
down



3 bit up counter

$$Q_2 \ Q_1 \ Q_0 \quad MC = 2^n - 1 = 7$$

0 0 0



BCD Ripple counter

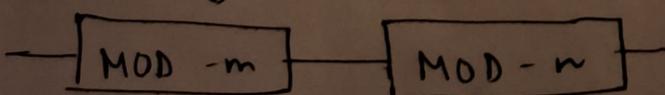
1) Negative edge triggered $\rightarrow Q$ is clock \rightarrow up counter

" " " $\rightarrow \bar{Q}$ is clock \rightarrow down counter

Positive " " " $\rightarrow \bar{Q}$ " " \rightarrow Up " "

" " " " $\rightarrow Q$ " " \rightarrow ~~up~~ down counter

2) Cascading counter



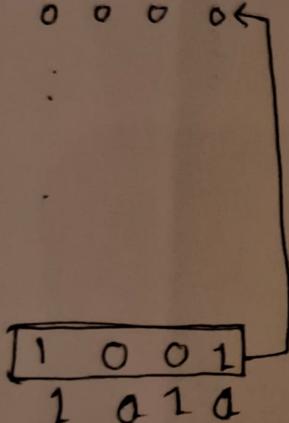
MOD mn

$$MC = 2^h - 1 = 2^{10} - 1 = 9$$

~~h = 10 states.~~

$$\begin{aligned} \text{no. of states } (n) &= 10 \\ \# MC &= 10 - 1 = 9 \end{aligned}$$

$Q_0 \ Q_1 \ Q_2 \ Q_3$
0 0 0 0 ←



Decade counter PR = 0

Convert 4 bit asynchronous
counter to decade counter

Clk = 0 $Q = 0$ set

Clk = 1 $Q = 1$ set

We want to make Clk = 0 as soon as
we reach 10 (1010)

, , , ,

—————

