

Unit-3 Sequential Circuits:-

Sequential Circuit:-

The circuit are dependent on clock pulse's & depends on present as well as past inputs to generate any output.

V Difference b/w Combinational & Sequential Circuit

Combinational

⇒ In this output depends only upon present input

⇒ Speed is fast

⇒ It is design easy

⇒ There is no feedback b/w Input & output

⇒ This is time independent

⇒ Elementary building blocks are logic gates

⇒ It is used for Arithmetic as well as boolean operations

Sequential

⇒ In this output depends upon present & past input

⇒ Speed is Slow

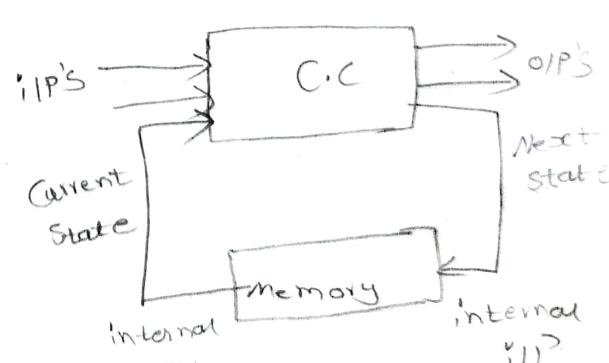
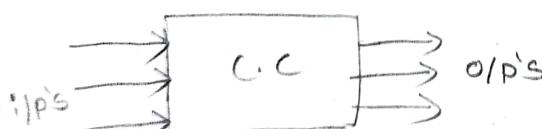
⇒ It is design ~~tasky~~ Compare to Combinational Circuit.

⇒ There exist feed back b/w Input & output.

⇒ This is time dependent

⇒ Elementary building blocks are flipflops.

⇒ It is mainly used for Storing the data.

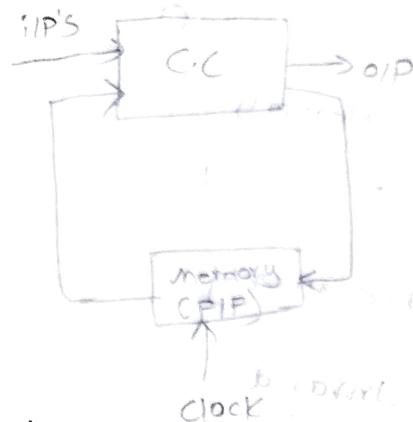


V Types of Sequential Circuits:-

1) Synchronous:-

The output behaviour depends on the input at a discrete time called Synchronous Sequential Circuit this circuit uses Clock pulses in the input's, of

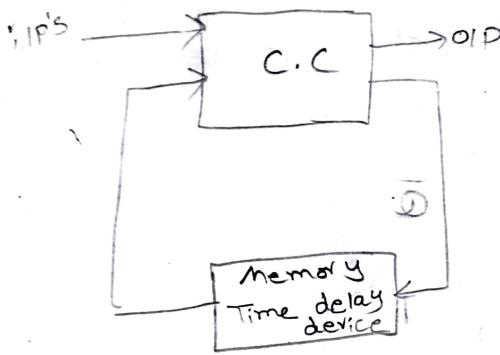
memory elements Clocked frequency Circuits.



2) Asynchronous :-

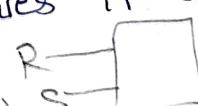
The Sequential Circuit Now's output is dependent on which input changes are called Asynchronous Sequential Circuit.

Ex:- Latch

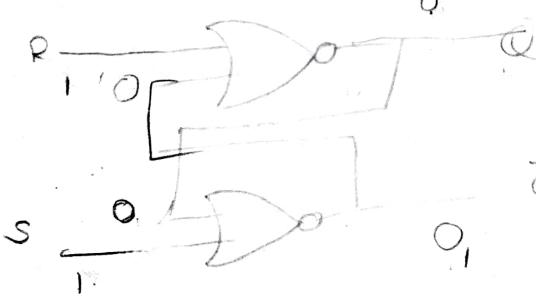


Latch:-

- These are used to design Asynchronous Sequential Circuit
- The main difference b/w latch & flipflops is clock pulses
- Latch has no clock pulses if clock pulses is given to latch it becomes flipflops.



Truth table for NOR



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

R

S

Q

\bar{Q}

0

1

1

0

0

0

Memory

1

0

0

1

0

0

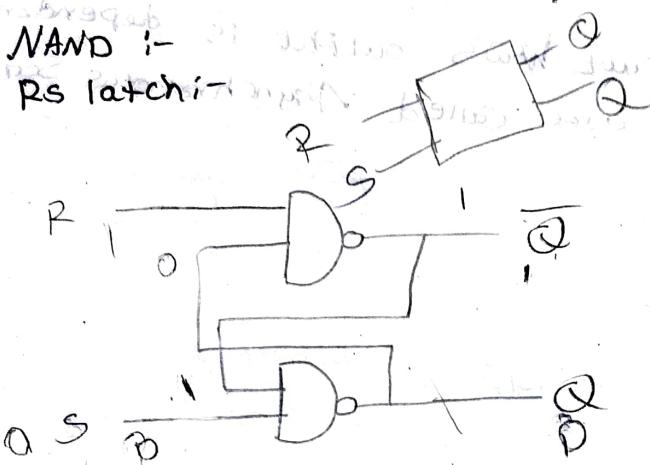
Memory

1

1

Invalid

NAND 1-2 definition
RS latch:



R

S

Q

\bar{Q}

0

1

0

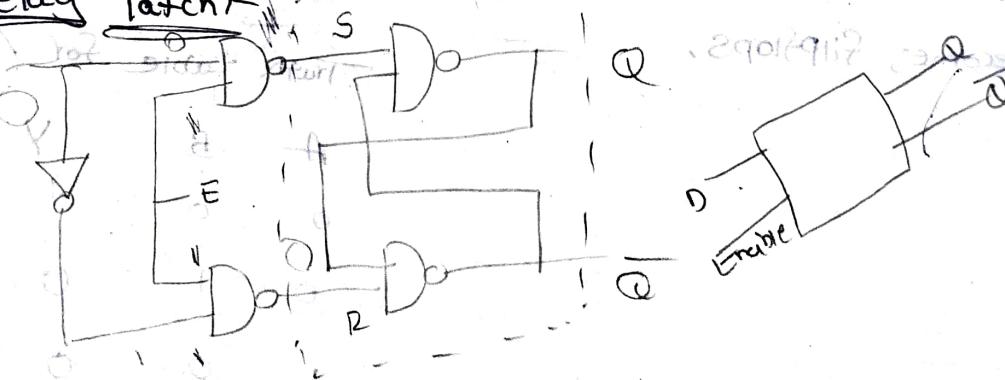
1

0

0

Invalid

Delay latch



0

1

0

1

0

1

A	B	V	R	S	\bar{Q}	Q
0	0	1	0	1	0	0
0	1	1	1	1	1	1
1	0	1	0	0	0	1
1	1	0	1	0	1	0
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	0	0	0
1	0	1	0	1	0	1
1	1	1	0	0	1	1

Ex. D \rightarrow \bar{Q}

IS 1 0 0 1 0

Result 1 1 0 1 1

Result 1 1 0 1 1

flip flop :-

A flip flop is a circuit that maintains a state
until directed by input to change the state

→ A basic flip flop can be constructed using NAND gates.

or NOR gates.

Four flip flops.

1) RS flip flop

2) JK flip flop

3) D flip flop

4) T flip flop

Applications of flip flops:-

1) Counter

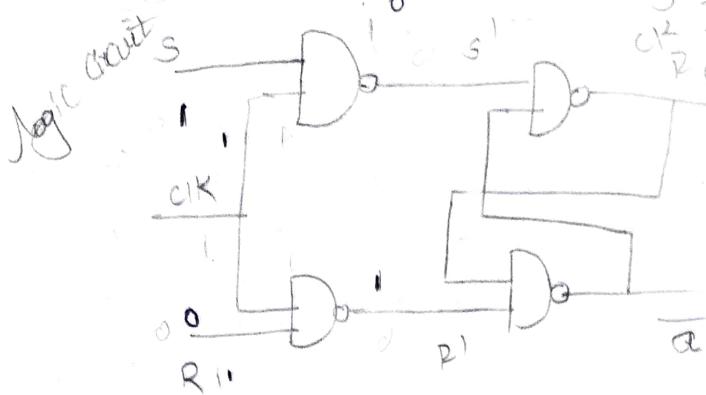
2) Register

3) Memory

data storage

data transfer

SR FLIP FLOP



A	B	V
0	0	1
0	1	1
1	0	1
1	1	0

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

Characteristic Table:-

CLK	C	S	R	Q	Q_{n+1}	State
\uparrow		0	0	0	0	Set
\uparrow		0	0	1	1	No change
\uparrow	Now	1	0	0	0	Reset
\uparrow		0	1	1	1	Set
\uparrow	1	1	0	0	1	Set
\uparrow	1	0	0	1	1	Reset
\uparrow	1	1	1	0	X	Indeterminate
\uparrow	1	1	1	1	X	Indeterminate

Excitation table :-

On Off

S

R

T

Q

Q'

A₁

0 0

0

X

0

00

01

10

0 1

1

0

1

01

11

10

1 0

0

1

1

01

10

11

1 1

X

0

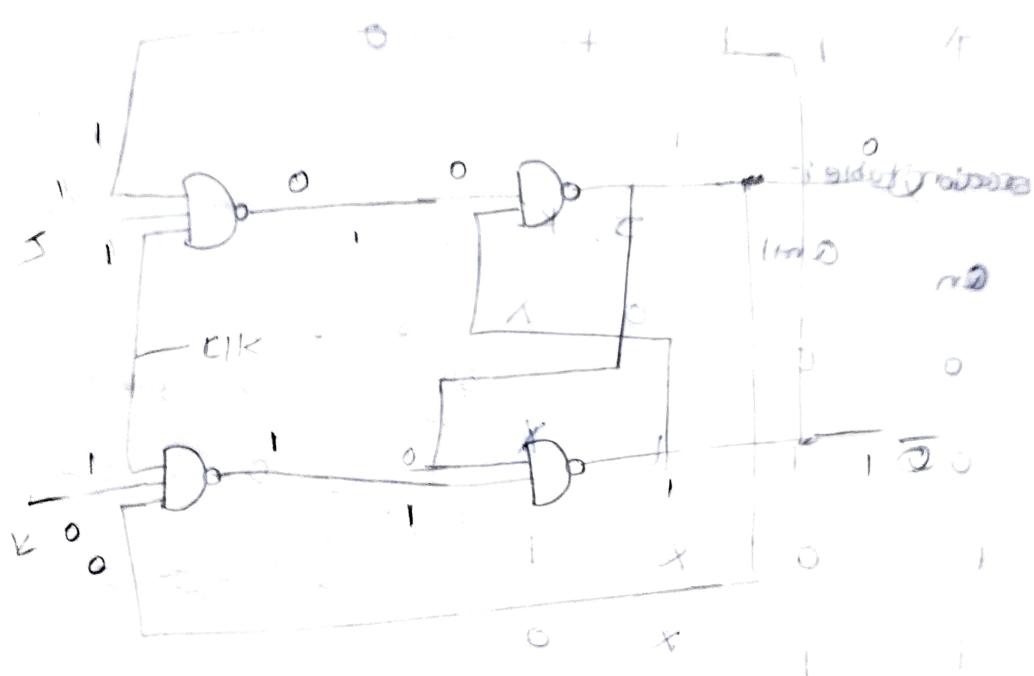
1

0

1

1

JK flip flop :-



Truth table :-

clk J K

Q Q' - toggle

0 0 0

X X

memory

0 0 1

0 0

memory

0 1 0

0 1

0 memory

0 1 1

1 0

1 0

1 0 0

1 1

1 1

1 0 1

1 0

1 0

1 1 0

0 1

0 1

1 1 1

0 0

0 0

Assume S1@=0, T1=1

Q1=0

Q2=1

Q3=0

Characteristic table

clk	J	K	Qn	Qnt	Output	State
↑	0	0	0	0	0	N.C.
↑	0	0	1	1	1	
↑	0	0	0	0	0	
↑	0	1	0	0	0	Preset
↑	0	1	1	1	0	
↑	0	0	0	0	1	Set
↑	1	0	1	1	1	
↑	1	1	0	1	1	Toggle
↑	1	1	1	0	0	

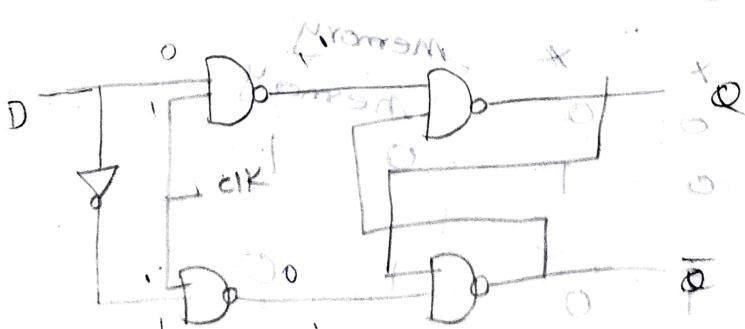
Excitation table :-

Qn	Qnt	J	K	Excitation
0	0	0	X	$\bar{Q}_n \bar{Q}_{nt}$
0	1	1	X	$\bar{Q}_n Q_{nt}$
1	0	X	1	$Q_n \bar{Q}_{nt}$
1	1	X	0	$Q_n Q_{nt}$

JK	JK	JK	JK
00	01	11	10
0	0	1	1
1	0	0	1

$$\bar{Q}_n \bar{Q}_{nt} + \bar{Q}_n J$$

SR latch Delay Flipflop



A ↓	B ↑	C ↓	D ↑
0 ↑ 0	0 ↑ 0	1	1
0 ↑ 1	1	1	1
1 ↑ 0	1	1	0
1 ↑ 1	0	0	0

R S E S T Q Q'

0 1 0 1 1 1

0 1 0 0 Invalid

memory

1 0 1 0 1 0

memory

1 1 1 1 0

\bar{Q}	D	\bar{Q}	D
0	0	1	1
1	0	0	0
0	1	1	0
1	1	0	1
0	0	0	0
1	1	1	1
0	0	1	0
1	1	0	1

\bar{Q}	D	Q_{n+1}	Q_{n+1}
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1
0	0	0	0
1	1	1	0
0	0	0	1
1	1	1	1

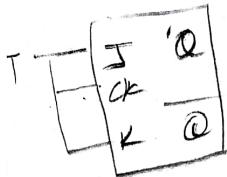
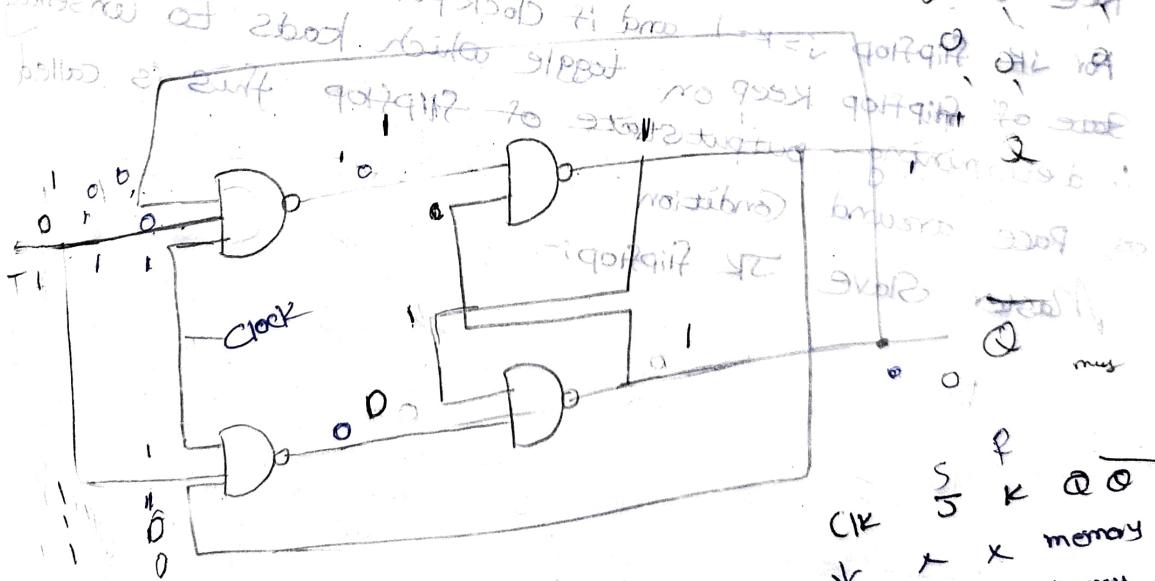
excitation table:-

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

Q_n	D	0	1	1	0
0	0	0	1	1	0
0	1	0	0	1	1

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

T flip-flop :-



CK	J	K	Q	\bar{Q}
\uparrow	T	X	memory	memory
\uparrow	0	0	0	1
\uparrow	0	1	0	1
\uparrow	1	0	1	0
\uparrow	1	1	1	0

Toggle.

RS latch		\bar{Q}	Q
+R	S	0	1
0	1	0	1
0	0	Invalid	
1	1	Memory	
1	0	1	0
1	1	Memory	

\bar{Q}	Q	\bar{Q}	Q	D	\bar{D}
1	0	0	1	1	0
0	1	1	0	0	1
1	1	0	0	1	1
0	0	1	1	0	0
1	0	0	1	1	1
0	1	1	0	0	0
1	1	0	0	1	1
0	0	1	1	0	0
1	1	1	0	1	1
0	0	0	1	0	1
1	1	1	1	1	0
0	0	1	0	0	1
1	1	0	1	1	1
0	0	0	0	0	0

Characteristic Table:-

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

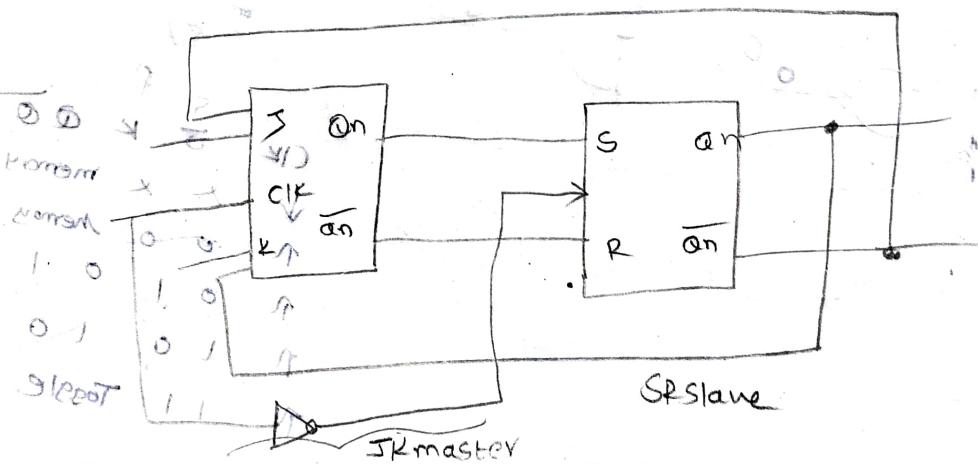
On	T	0	1
0	0	0	1
1	0	1	0

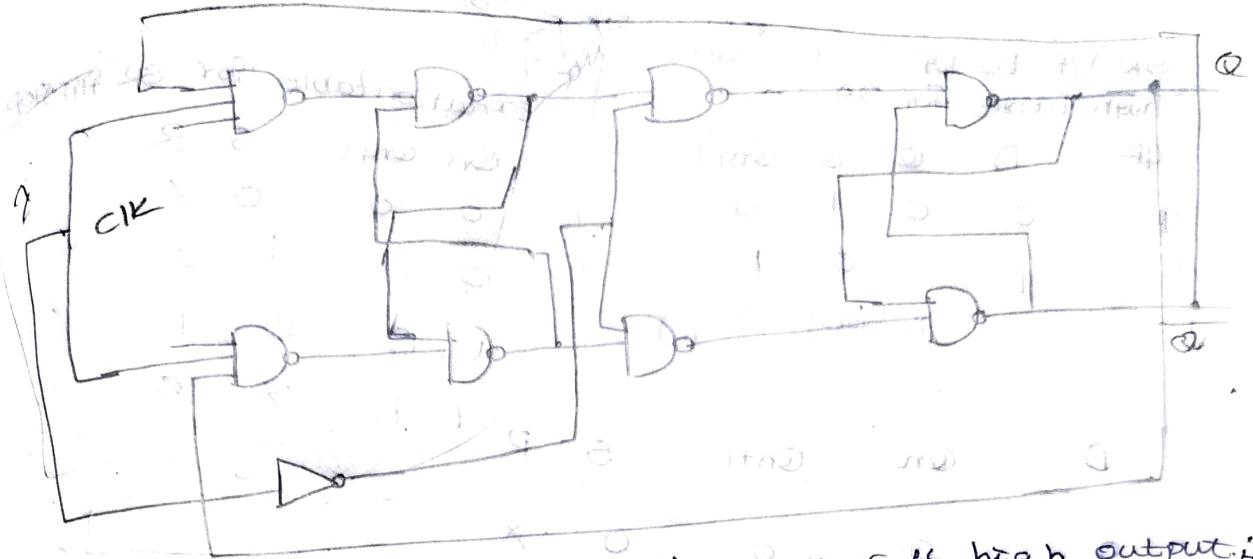
$$Q_{n+1} = Q_n \bar{T} + \bar{Q}_n T$$

Race around Condition:-

For JK flipflop $j=k=1$ and if clock pulse is ~~too~~ long then state of flipflop keep on toggle which leads to conserving output state of flipflop this is called as Race around condition.

Master Slave JK flipflop:-





- When the clock pulse is high the master is high output is remains high till the clock is low because the state is stored.
- Now the output of master becomes low^{to} , the clock pulse becomes high & remain low until the clock becomes high again.

- Does toggle takes place for a clock cycle?
- When the clock pulse is high the master is operational but not the slave the output of the slave remains ~~when~~ high till the clock becomes low.
- The clock is low slave becomes operational and remains high until the clock again becomes low.
- The clock is high during the whole process remains high during one in a cycle.
- Toggle takes place changing one in a cycle.
- Since the output is changing one in a cycle.

- ### Flip-flop Conversions
- Consider truth table for destination flip-flop and extends it as excitation table of source table.
- Draw the k-map for source flip-flop input variables it will provide expression for conversion, expression according to k-map.
- Draw the circuit according to expression.

Conversions:-

SR FF to DFF domestic table
 Truth table for DFF (or)

JK	D	Q	\bar{Q}	On +
↑	0	0	1	0
↑	1	1	0	1
↓	X	Memory	Memory	Truth table

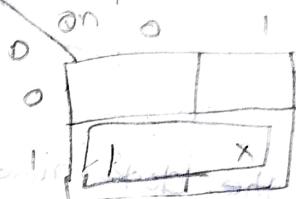
SR $\xrightarrow{JK} T$ $T \xrightarrow{SR} JK$

JK $\xrightarrow{T} SR$

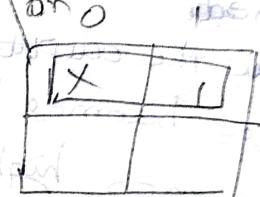
Excitation table for SR flipflop

On +	S	R
0 0	0	X
0 1	1	0
1 0	0	1

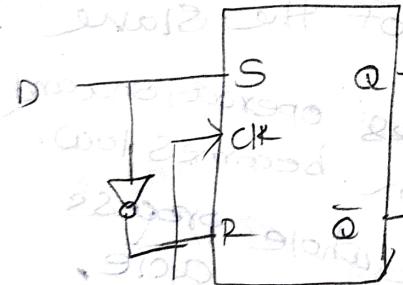
K-map for S



$S = D$ for S = 1



Excitation table for R = D



SR-JK flip flop :-

characteristic table for easy

JK flip flop truth table

SR flip flop excitation table

On +

0 0 0 X

0 0 1 0

0 1 0 1

1 0 0 0

1 1 X 1

On +	J	K	Q	\bar{Q}
↑	0	0	0	1
↑	1	0	1	0
↑	1	1	0	1

Toggle

Characteristic table for T

S	K	On	(cont)	S	R
0	0	0	0	0	X
0	0	1	0	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

K map for S

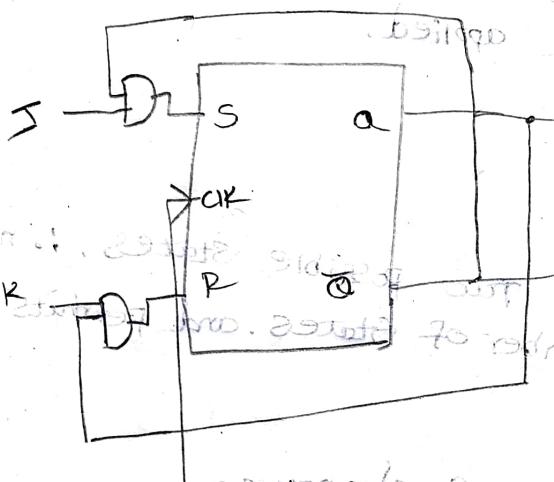
JK	JK	JK	JK	JK
00	01	11	10	
00	01	11	10	
00	01	11	10	

$$S = \overline{Q_{n-1} Q_n}$$

K map for R

JK	JK	JK	JK	JK
00	01	11	10	
00	01	11	10	
00	01	11	10	

$$R = \overline{Q_n K}$$



SR-T flip/flop mapping

characteristic table for T

QK	T	On	(cont)
10	0	0	0
10	1	1	1

↑ Position and performance of SR
↑ Position and performance of T
↑ Position and performance of JK

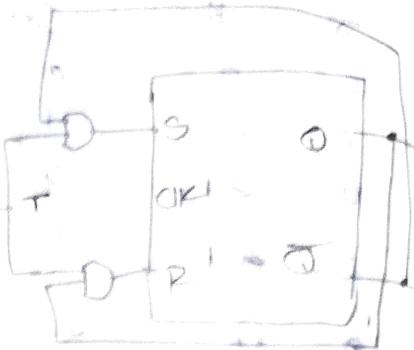
SR flip/flop excitation table

On	On	S	R
0	0	0	X
0	1	1	0
1	0	0	1

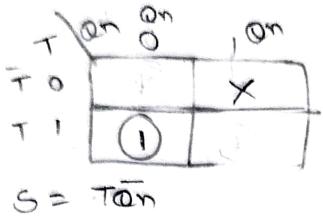
↑ Position and performance of SR

Characteristic table

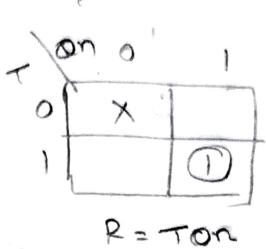
CLK	T	on	Cont	S	R
↑	0	0	0	0	X
↑	0	01	1	1	0
↑	1	0	1	X	0
↑	1	1	0	0	1



K map for S



K map for R



Counters:-

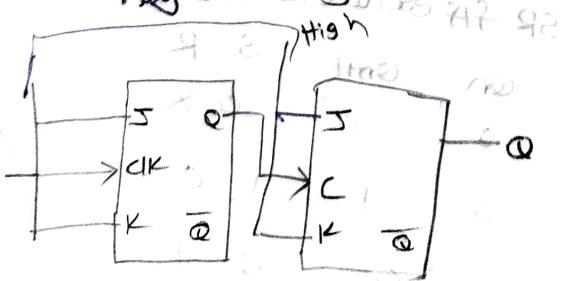
- A Counter is a Sequential Circuit. It is used for Counting pulses.
- Counter is the widest application of flipflops it is a group of flipflops with a signal applied.
- Counters are of two types

1) Asynchronous.

2) Synchronous

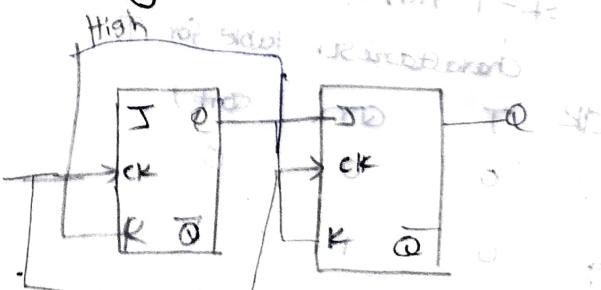
→ we know that a flip flop has two possible states, $\therefore n$ flip flop there will be 2^n number of states and permits counting from zero to $2^n - 1$.

Asynchronous



Output of the first flip-flop driving the clock for the next flip-flop.

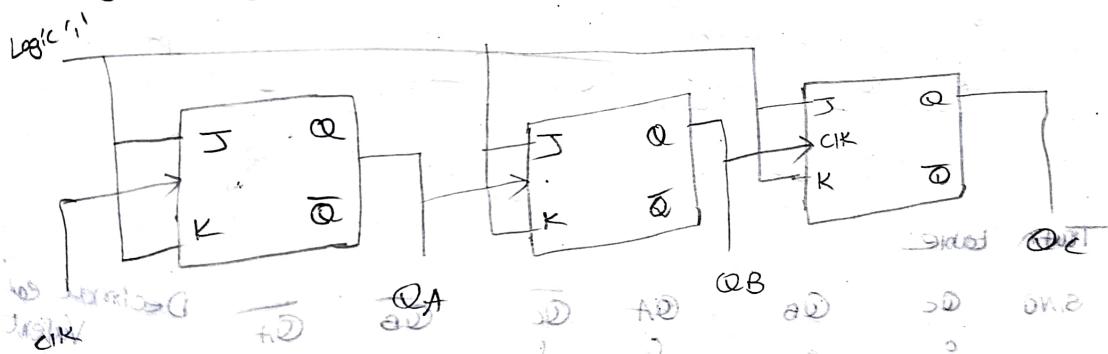
Synchronous



→ No connection b/w output of first flip & clock input of the next flip.

- ⇒ All the flipflops are not clocked simultaneously ⇒ All the flipflops are clocked simultaneously
- ⇒ Logic circuit is very simple ⇒ Logic circuit is complex.
- ⇒ These counters are slow etc. ⇒ These counters are fast. (Cause of propagation delay.)
- ⇒ Counter can count in two ways:
 - i) Upcounter $[0, 1, 2, \dots, n]$.
Ex - EVM
 - ii) Down counter $[n, n-1, n-2, \dots, 0]$.
Ex - Space
- ⇒ Present count of the counter represents state.
- ⇒ Counter contains set of flipflops. A n bit counter will have n flipflops & 2^n states.
- ⇒ Each state frequency = $\frac{\text{Total frequency}}{2^{n \text{ bits stored in register}}}$

Asynchronous (Ripple) Up Counter :-



Truth table:-

S.NO	J	Q _C	Q _B	Q _A
0	0	0	0	0
1	0	0	1	0
2	0	0	1	1
3	0	1	0	0
4	1	0	0	0
5	1	0	0	1
6	1	1	1	0
7	1	1	1	1
8	0	0	0	0

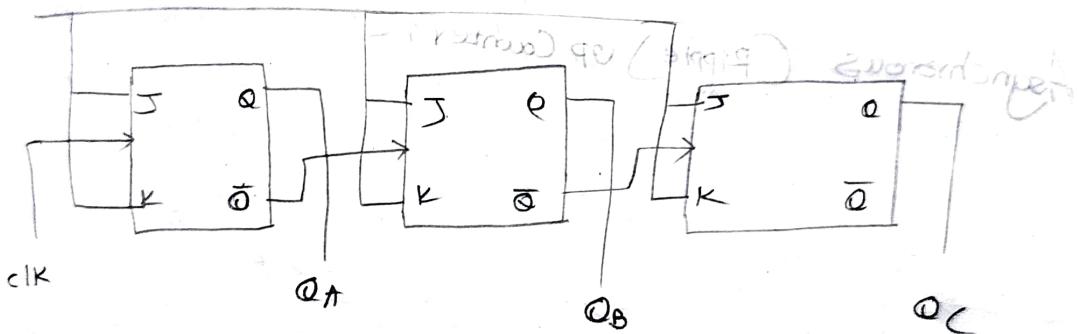
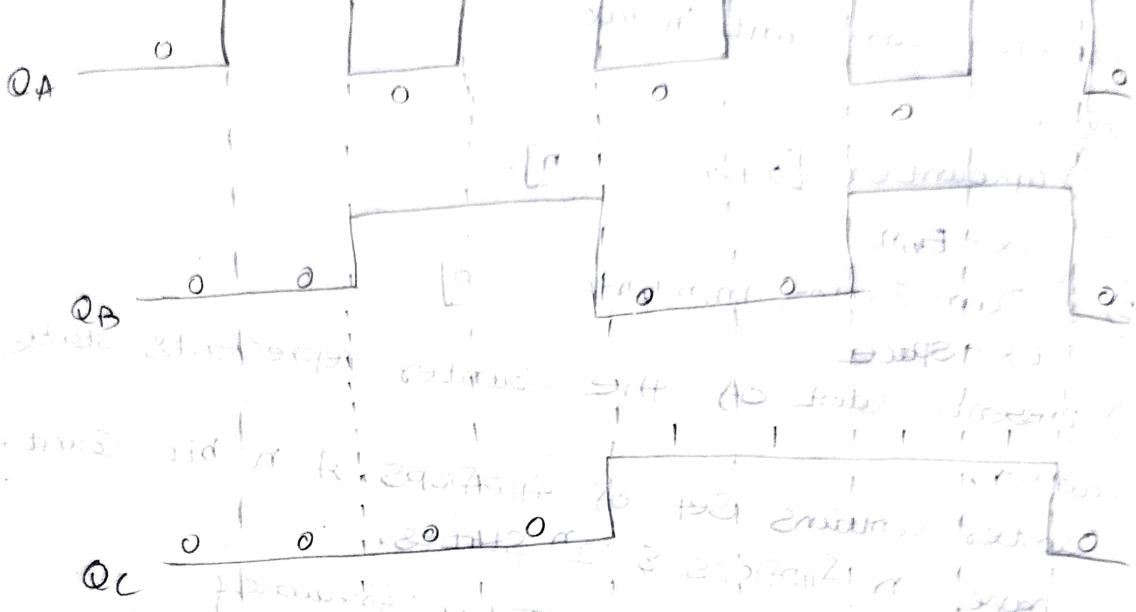
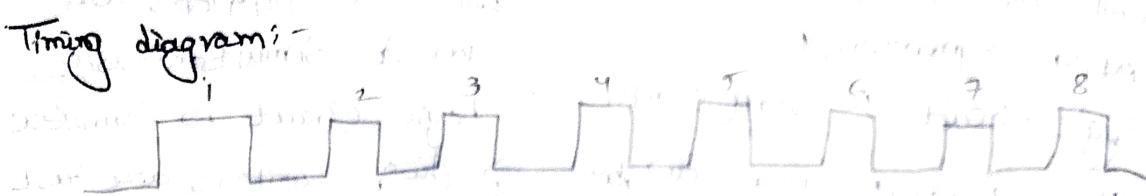
Decimal equivalent

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

$$\begin{aligned} \text{States} &= 2^3 \\ &= 2^3 \\ &= 8 \end{aligned}$$

$$\text{Max. Count} = 2^3 - 1$$

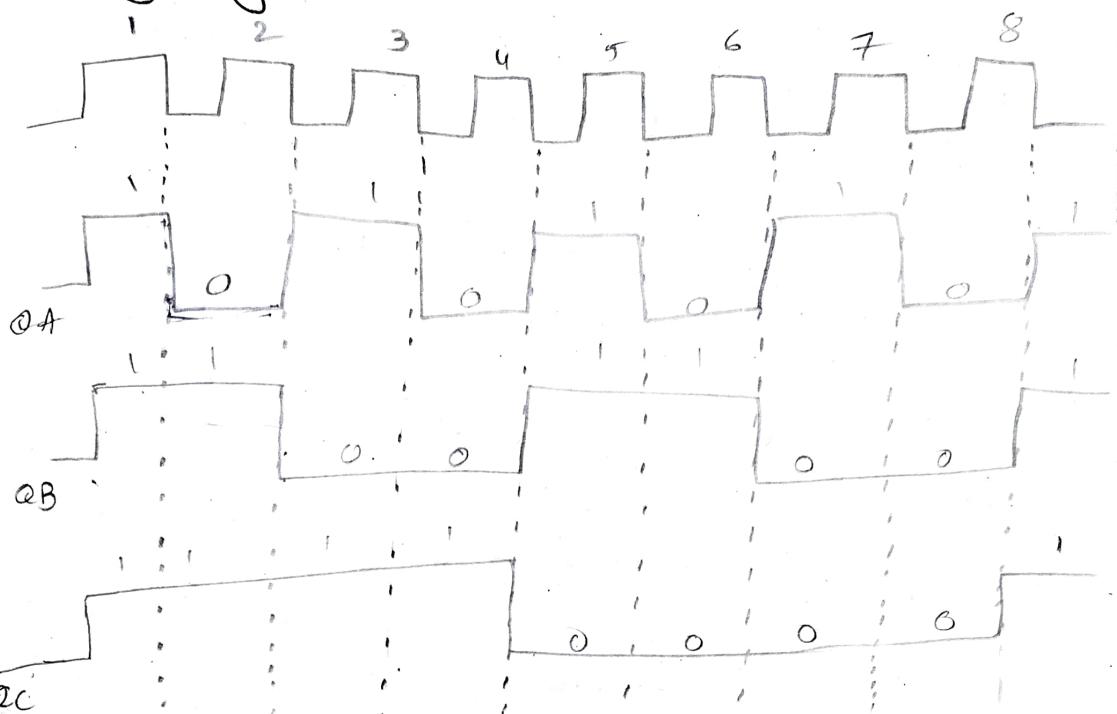
$$\begin{aligned} &= 8 - 1 \\ &= 7 \end{aligned}$$



Truth table

S.No	Q_C	Q_B	Q_A	\bar{Q}_C	\bar{Q}_B	\bar{Q}_A	Decimal equivalent
1	0	0	0	1	1	1	7
2	0	1	0	1	0	1	6
3	0	1	1	1	0	0	5
4	1	0	0	0	1	0	4
5	1	0	1	0	1	1	3
6	1	1	0	0	0	1	2
7	1	1	1	0	0	0	1
8	0	0	0	1	1	1	7

Timing diagram:-



3-bit up-down Counter

- A mode control input (m) is used to select either up or down mode.
- When $m=0$ it will do the up counting & connected to the clock of the next flipflop.
- When $m=1$ it will do the down counting & connected to the clock of the next flipflop.

m	Q_1	\bar{Q}_1	Y
0	0	0	0
0	0	1	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Here Q_1 is the o/p

m	Q_1	\bar{Q}_1	Y
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Q_1 is the o/p.

