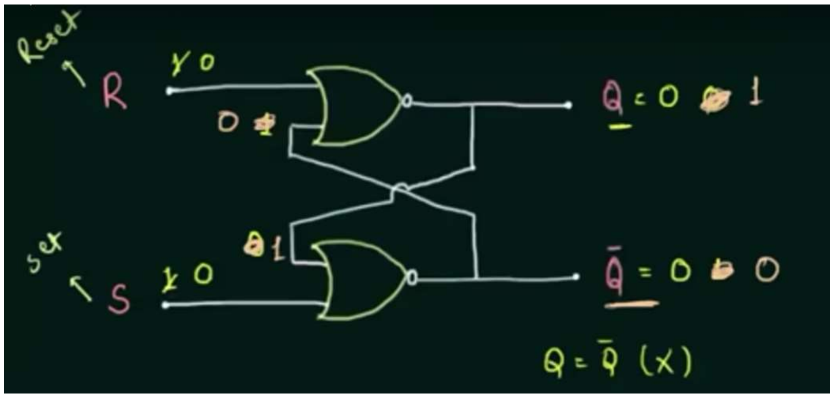
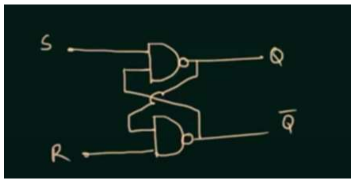
# ESE LOGIC DESIGN

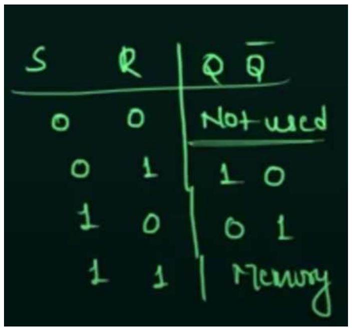
13 April 2024 01:32 AM

SR LATCH

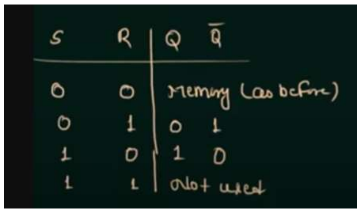
Level sensitive







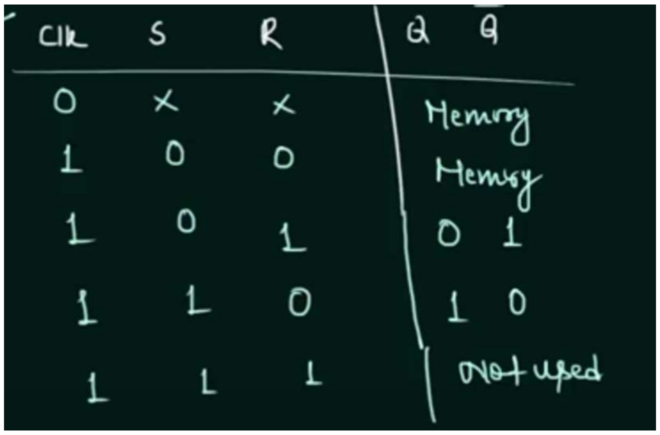
| S | R | Q $\bar{Q}$ |
|---|---|---|
| 0 | 0 | Not used |
| 0 | 1 | 1 0 |
| 1 | 0 | 0 1 |
| 1 | 1 | Memory |



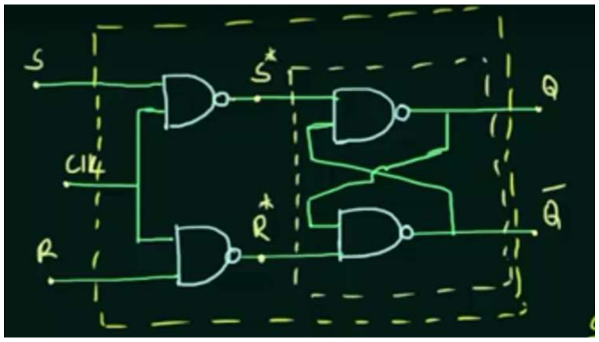
| S | R | Q | $\bar{Q}$ |
|---|---|---|---|
| 0 | 0 | Memory (as before) | |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | Not used | |

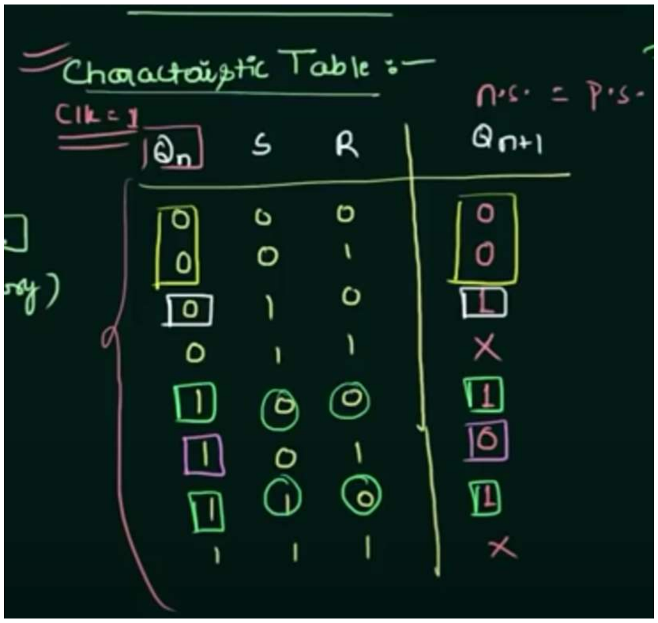
FLIP FLOP

Edge sensitive



| 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 |



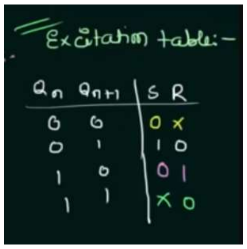


Characteristic Table :-

Clk = 1

n.s. = p.s.

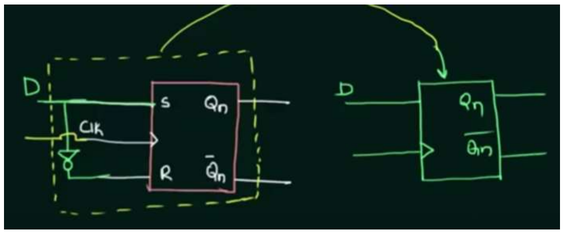
| $Q_n$ | S | R | $Q_{n+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | X |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | X |

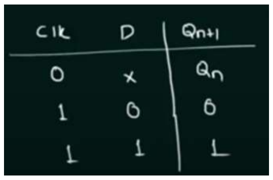


Excitation table:-

| $Q_n$ | $Q_{n+1}$ | S | R |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

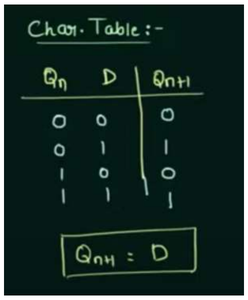
D FLIP FLOP





| Clk | D | $Q_{n+1}$ |
|---|---|---|
| 0 | x | $Q_n$ |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

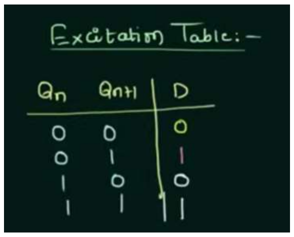
Only to store data we use this FF



Char. Table :-

| $Q_n$ | D | $Q_{n+1}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

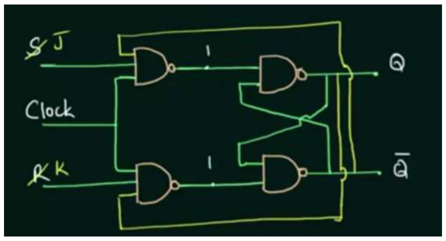
$\boxed{Q_{n+1} = D}$



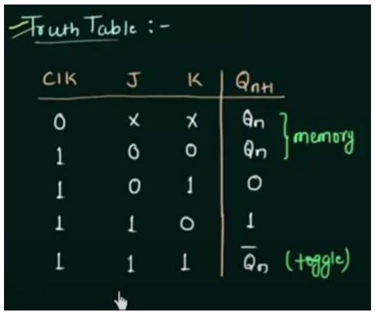
Excitation Table :-

| $Q_n$ | $Q_{n+1}$ | D |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

JK FLIP FLOP

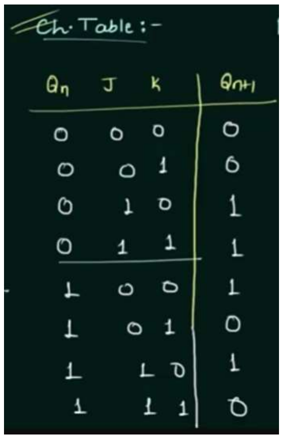


Start initially with Q=0



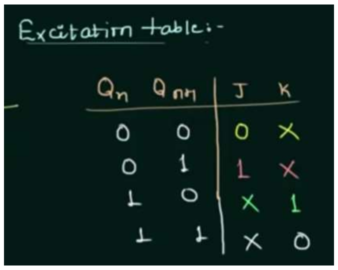
Truth Table :-

| Clk | J | K | $Q_{n+1}$ | |
|---|---|---|---|---|
| 0 | x | x | $Q_n$ | memory |
| 1 | 0 | 0 | $Q_n$ | memory |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | $\bar{Q}_n$ | (toggle) |



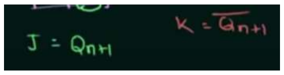
Ch. Table :-

| $Q_n$ | J | K | $Q_{n+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

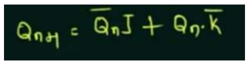


Excitation table :-

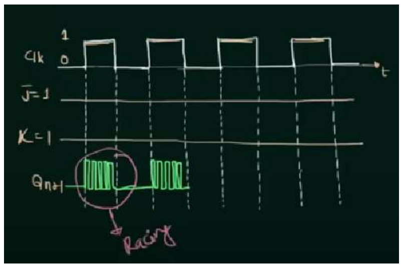
| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |



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



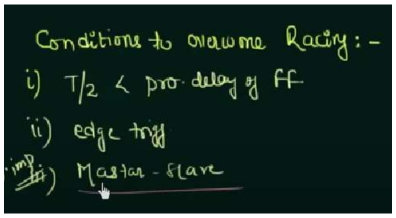
$$Q_{n+1} = \overline{Q_n}J + Q_n \cdot \overline{K}$$



Unstable
diff than toggle

Delay > T/2

Toggling is controlled



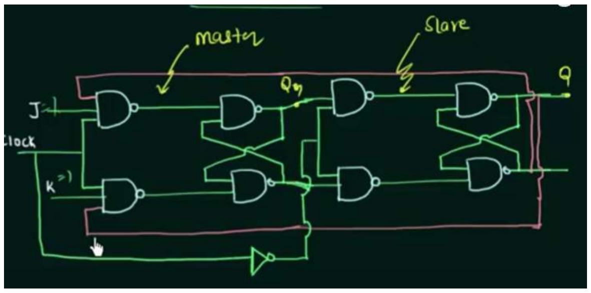
Conditions to overcome Racing :-

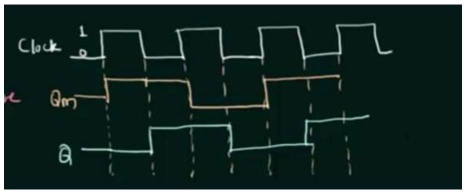
i) T/2 < pro. delay of FF

ii) edge trig

imp iii) Master - Slave

Master slave is same as neg edge trig

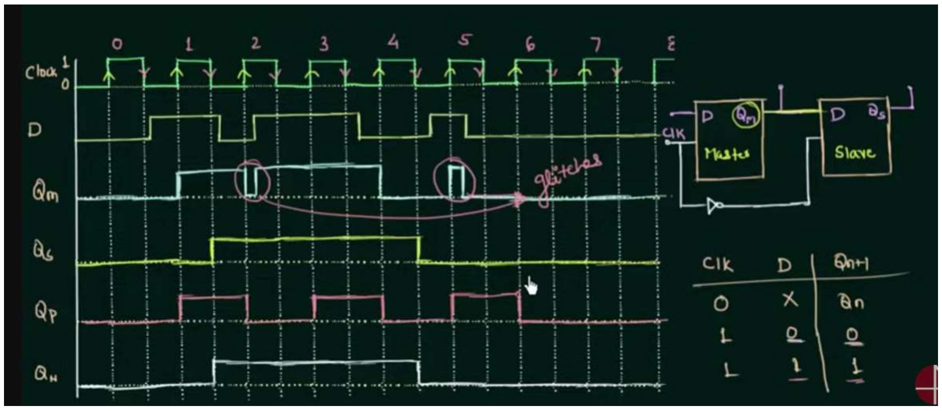




master

slave

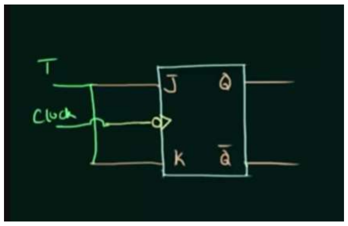
Glitches = sudden change in the output signal

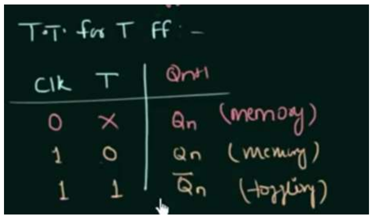


| Clk | D | $Q_{n+1}$ |
|---|---|---|
| 0 | X | $Q_n$ |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

There are no glitches in slave that why we use master slave FF
Master slave output is same as the neg edge output

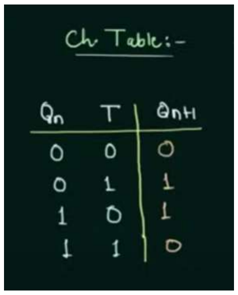
T Flip Flop - Toggle





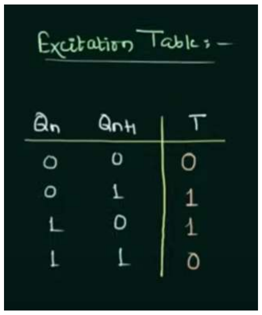
T.T. for T ff :-

| Clk | T | $Q_{n+1}$ |
|---|---|---|
| 0 | X | $Q_n$ (memory) |
| 1 | 0 | $Q_n$ (memory) |
| 1 | 1 | $\bar{Q}_n$ (toggling) |



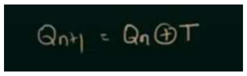
Ch. Table:-

| $Q_n$ | T | $Q_{n+1}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



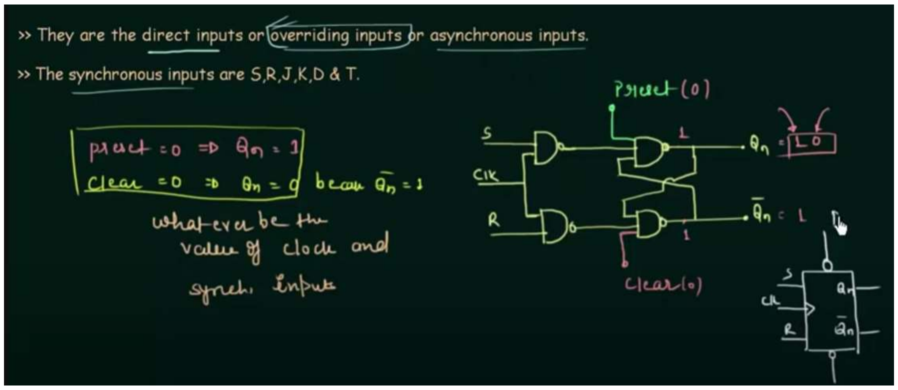
Excitation Table:-

| $Q_n$ | $Q_{n+1}$ | T |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



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

PRESET AND RESET

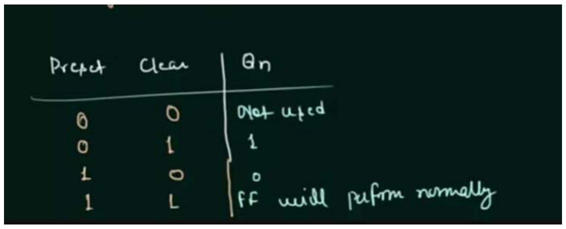


>> They are the direct inputs or overriding inputs or asynchronous inputs.

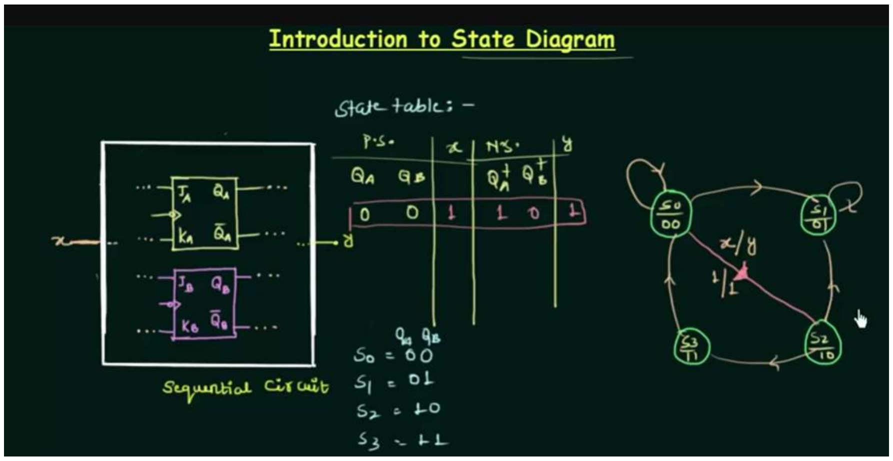
>> The synchronous inputs are S,R,J,K,D & T.

preset = 0 ⇒ $Q_n = 1$
clear = 0 ⇒ $Q_n = 0$ becau $\bar{Q}_n = 1$

whatever be the value of clock and synch. inputs



| Preset | Clear | $Q_n$ |
|---|---|---|
| 0 | 0 | not used |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | ff will perform normally |

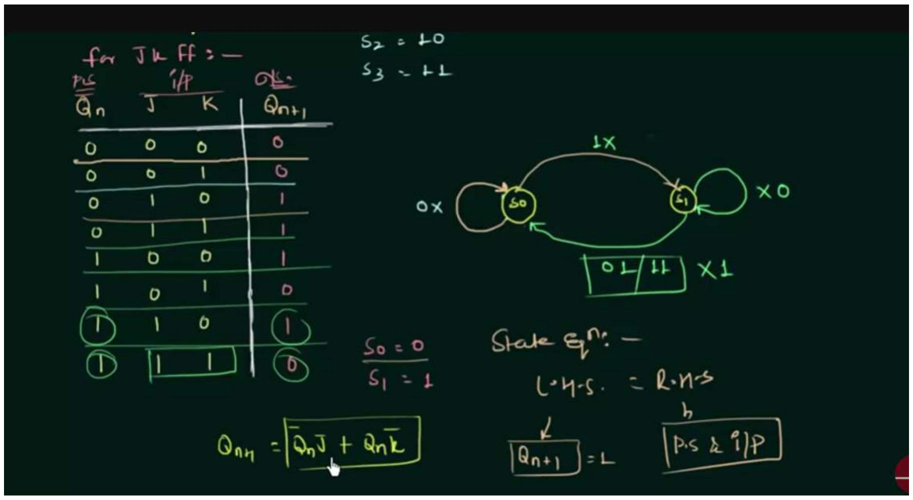


Introduction to State Diagram

State table:-

| P.S. | | x | N.S. | | y |
|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | | $Q_A^+$ | $Q_B^+$ | |
| 0 | 0 | 1 | 1 | 0 | 1 |

$Q_A$ $Q_B$
$S_0$ = 00
$S_1$ = 01
$S_2$ = 10
$S_3$ = 11



for JK FF :—

$S_2 = 10$

$S_3 = 11$

| P.S | i/P | | O/P |
|---|---|---|---|
| $Q_n$ | J | K | $Q_{n+1}$ |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$S_0 = 0$

$S_1 = 1$

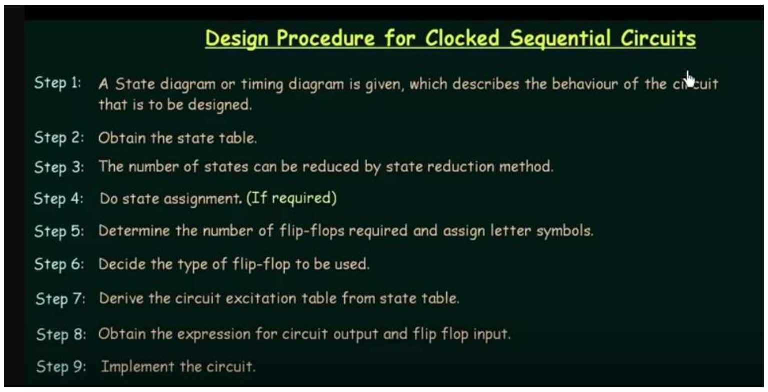
State eqn :—

L.H.S. = R.H.S

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

$\boxed{P.S \ \& \ i/P}$

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



## Design Procedure for Clocked Sequential Circuits

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

Step 2: Obtain the state table.

Step 3: The number of states can be reduced by state reduction method.

Step 4: Do state assignment. (If required)

Step 5: Determine the number of flip-flops required and assign letter symbols.

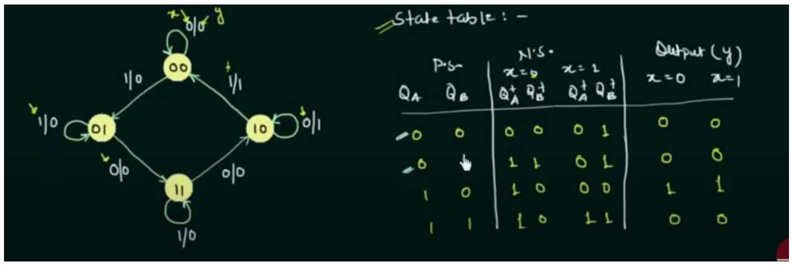
Step 6: Decide the type of flip-flop to be used.

Step 7: Derive the circuit excitation table from state table.

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

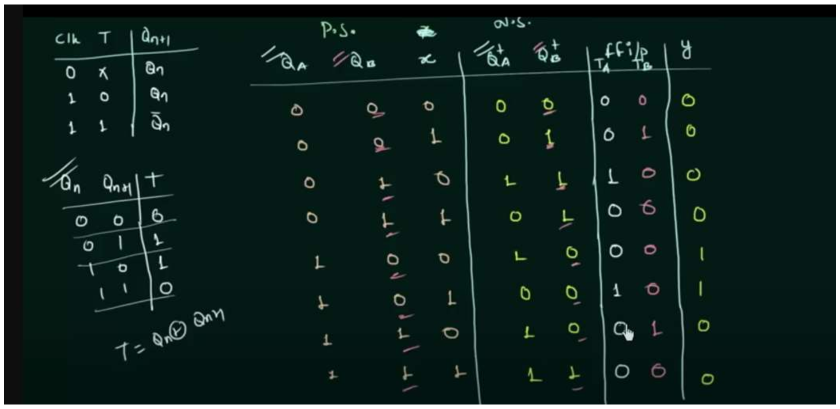
Step 9: Implement the circuit.

Next stage as well as the output must be same for two states for them to be reduced



State table :—

| P.S. | | N.S. | | | | Output (y) | |
|---|---|---|---|---|---|---|---|
| | | x=0 | | x=1 | | x=0 | x=1 |
| $Q_A$ | $Q_B$ | $Q_A^+$ | $Q_B^+$ | $Q_A^+$ | $Q_B^+$ | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | [illegible] | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

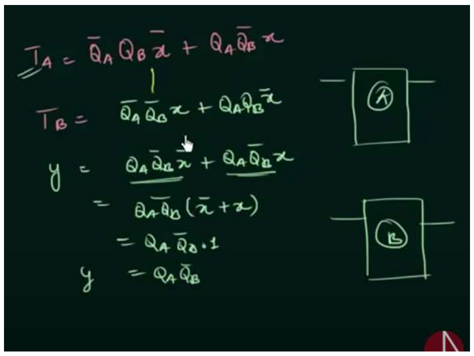


| Clk | T | $Q_{n+1}$ |
|---|---|---|
| 0 | x | $Q_n$ |
| 1 | 0 | $Q_n$ |
| 1 | 1 | $\bar{Q}_n$ |

| $Q_n$ | $Q_{n+1}$ | T |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

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

| P.S. | | | N.S. | | ff i/p | | y |
|---|---|---|---|---|---|---|---|
| $Q_A$ | $Q_B$ | $x$ | $Q_A^+$ | $Q_B^+$ | $T_A$ | $T_B$ | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |



$T_A = \bar{Q}_A Q_B \bar{x} + Q_A \bar{Q}_B x$

$T_B = \bar{Q}_A \bar{Q}_B x + Q_A Q_B \bar{x}$

$y = \underline{Q_A \bar{Q}_B} \bar{x} + \underline{Q_A \bar{Q}_B} x$

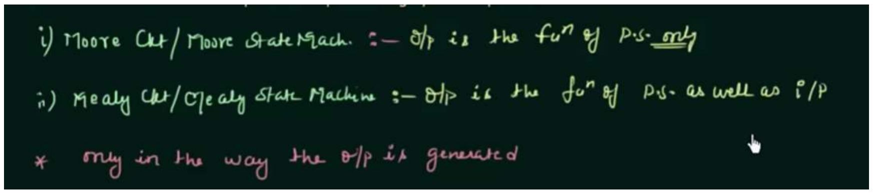
$= Q_A \bar{Q}_B (\bar{x} + x)$

$= Q_A \bar{Q}_B \cdot 1$

$y = Q_A \bar{Q}_B$

MEALY AND MOORE CIRCUITS

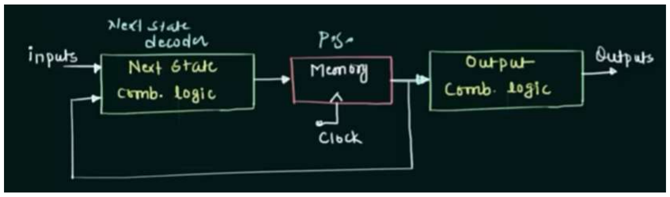
MOORE = Doesn't depend upon the input
Diff is the way the output is calc



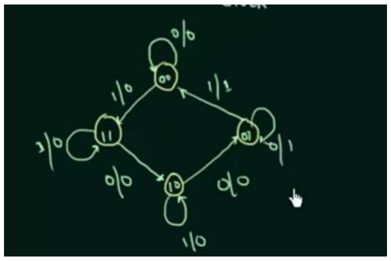
i) Moore Ckt / Moore State Mach. :— O/p is the fun of P.S. only

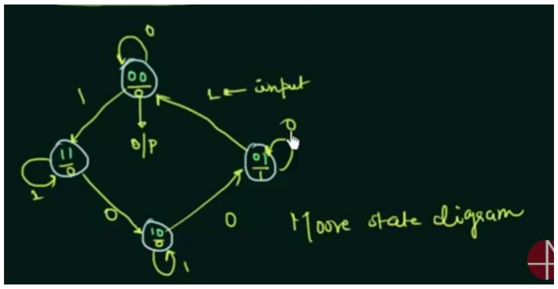
ii) Mealy Ckt / Mealy State Machine :— O/p is the fun of P.S. as well as i/p

* only in the way the o/p is generated

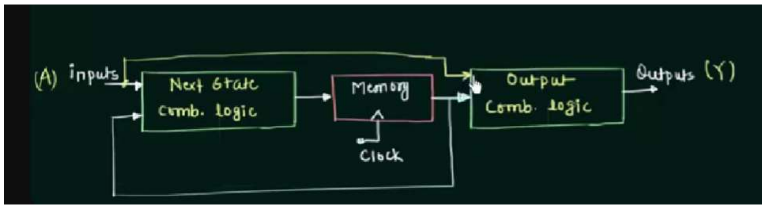


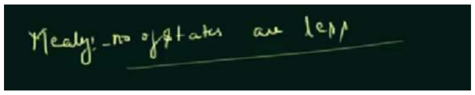
moore





MEALY



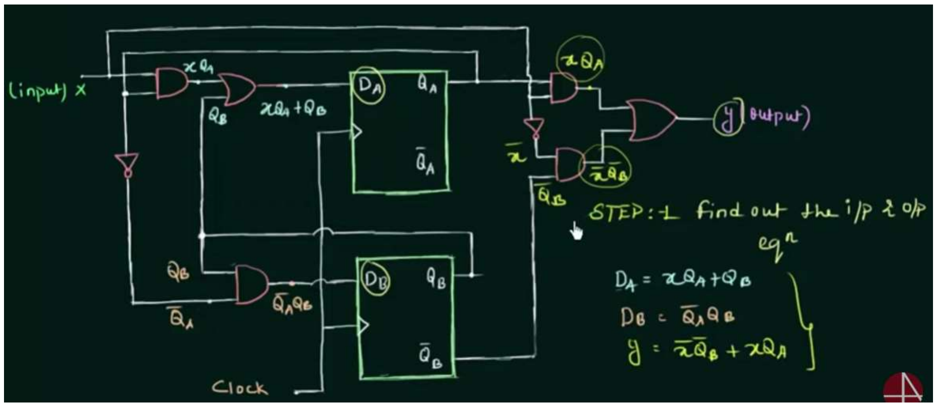


Mealy: no of states are less

If same logic is applied

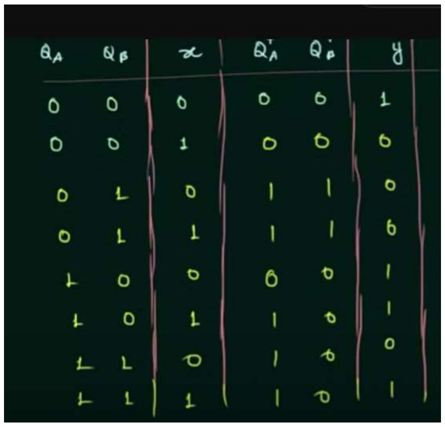
ANALYSIS OF CLOCKED SEQUENTIAL CIRCUITS

Write the equations for the inputs and outputs
Then write the state table
Then make the state diagram



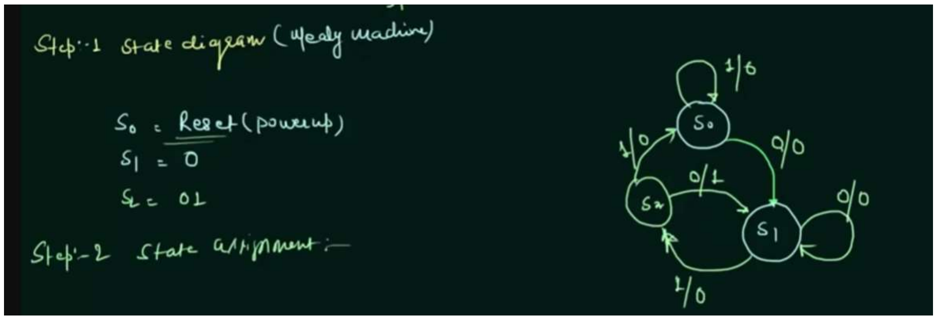
STEP: 1 find out the i/p & o/p eq^n

$$D_A = xQ_A + Q_B$$
$$D_B = \bar{Q}_A Q_B$$
$$y = \bar{x}\bar{Q}_B + xQ_A$$



| $Q_A$ | $Q_B$ | $x$ | $Q_A'$ | $Q_B'$ | $y$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

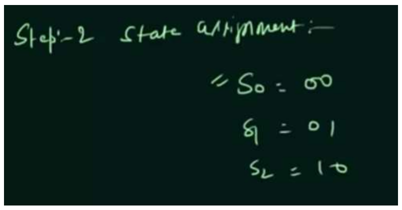
PATTERN DETECTOR



Step: 1 State diagram (Mealy machine)

$S_0$ = Reset (powerup)
$S_1$ = 0
$S_2$ = 01

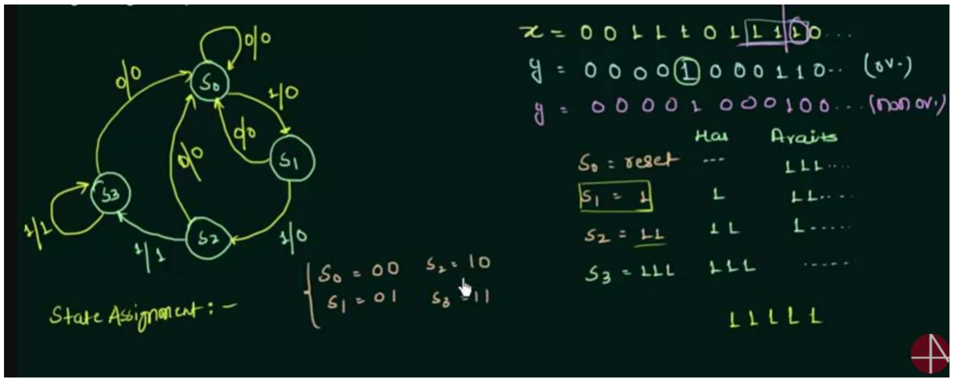
Step-2 State assignment :-

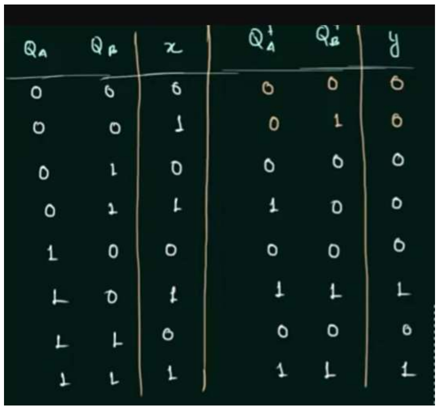


Step-2 State assignment :-

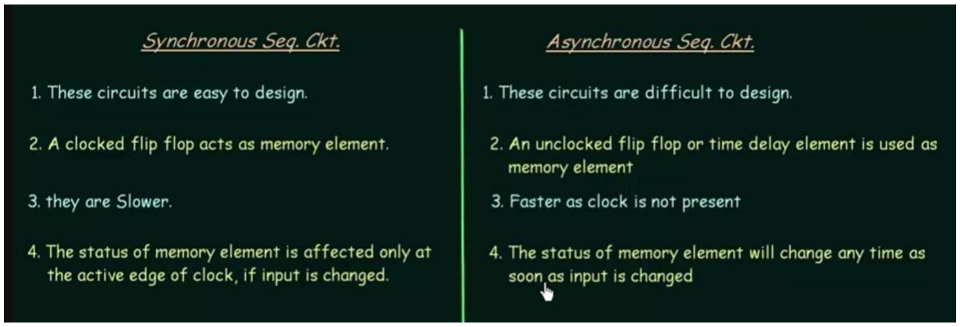
$S_0$ = 00
$S_1$ = 01
$S_2$ = 10

Considering OV



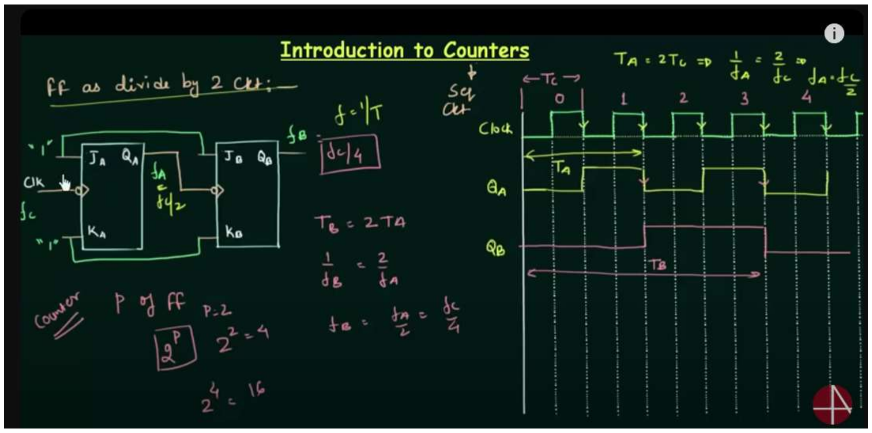


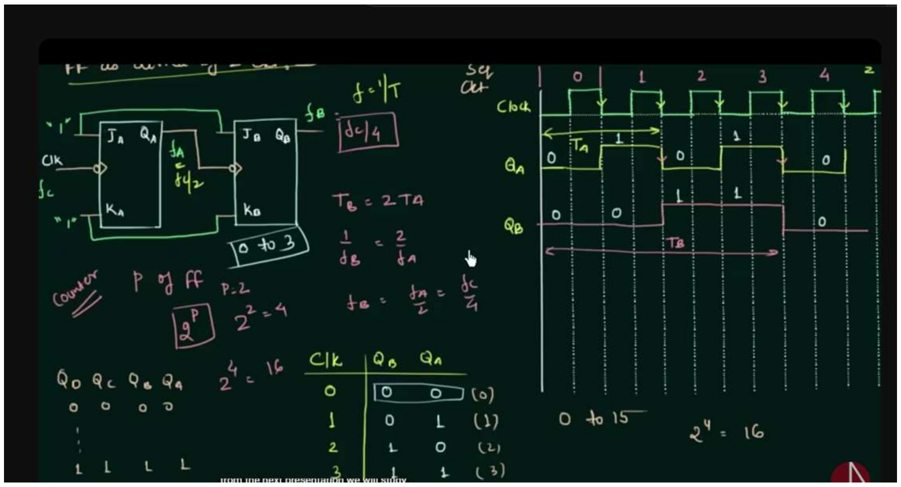
| $Q_A$ | $Q_B$ | $x$ | $Q_A^+$ | $Q_B^+$ | $y$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

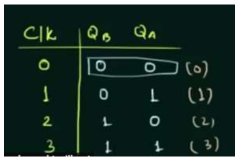


| *Synchronous Seq. Ckt.* | *Asynchronous Seq. Ckt.* |
|---|---|
| 1. These circuits are easy to design. | 1. These circuits are difficult to design. |
| 2. A clocked flip flop acts as memory element. | 2. An unclocked flip flop or time delay element is used as memory element |
| 3. they are Slower. | 3. Faster as clock is not present |
| 4. The status of memory element is affected only at the active edge of clock, if input is changed. | 4. The status of memory element will change any time as soon as input is changed |

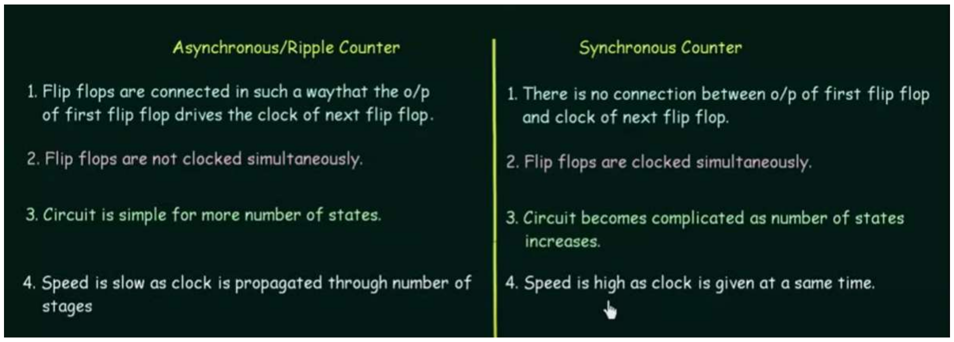
COUNTERS



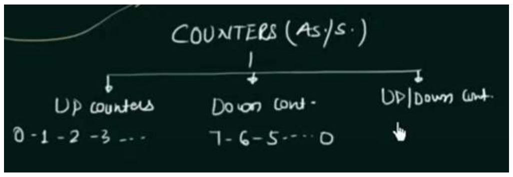




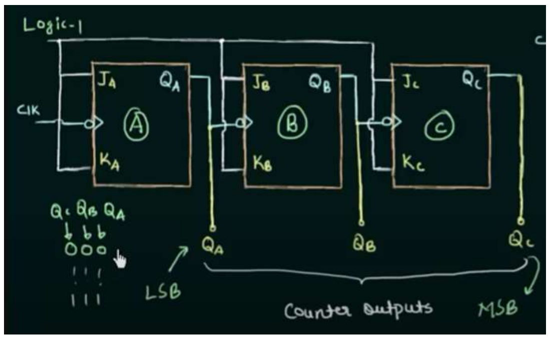
| Clk | $Q_B$ | $Q_A$ | |
|---|---|---|---|
| 0 | 0 | 0 | (0) |
| 1 | 0 | 1 | (1) |
| 2 | 1 | 0 | (2) |
| 3 | 1 | 1 | (3) |



| Asynchronous/Ripple Counter | Synchronous Counter |
|---|---|
| 1. Flip flops are connected in such a waythat the o/p of first flip flop drives the clock of next flip flop. | 1. There is no connection between o/p of first flip flop and clock of next flip flop. |
| 2. Flip flops are not clocked simultaneously. | 2. Flip flops are clocked simultaneously. |
| 3. Circuit is simple for more number of states. | 3. Circuit becomes complicated as number of states increases. |
| 4. Speed is slow as clock is propagated through number of stages | 4. Speed is high as clock is given at a same time. |

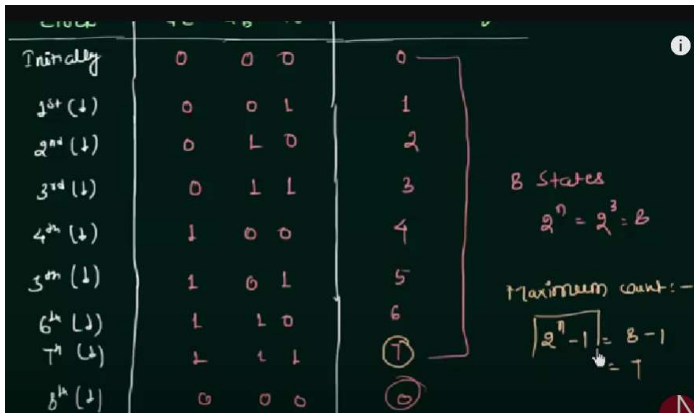


3 BIT ASYNC COUNTER



No of states = 2^n

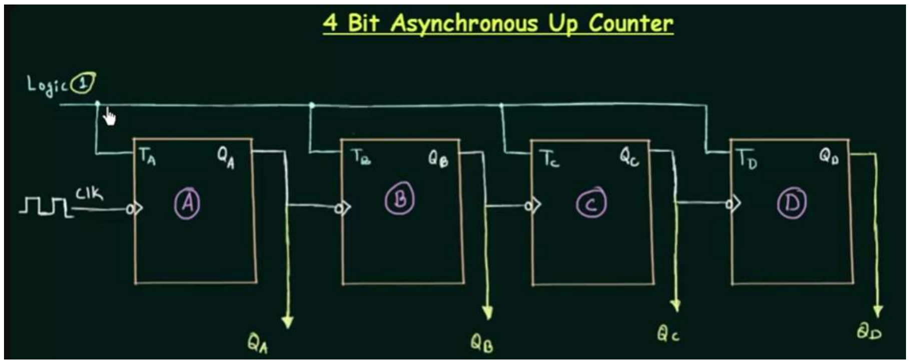
N bit = n FF



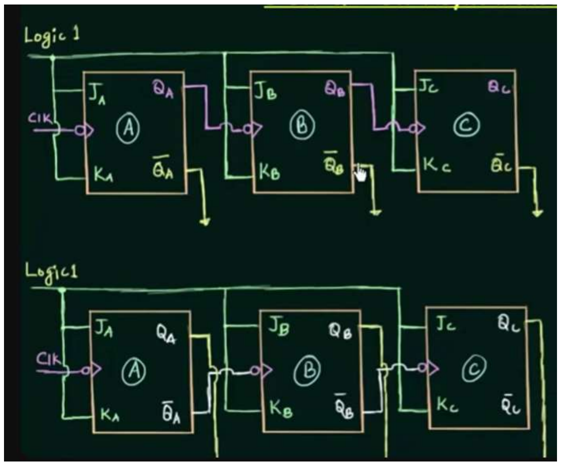
| | | | | |
|---|---|---|---|---|
| Initially | 0 | 0 | 0 | 0 |
| 1st (↓) | 0 | 0 | 1 | 1 |
| 2nd (↓) | 0 | 1 | 0 | 2 |
| 3rd (↓) | 0 | 1 | 1 | 3 |
| 4th (↓) | 1 | 0 | 0 | 4 |
| 5th (↓) | 1 | 0 | 1 | 5 |
| 6th (↓) | 1 | 1 | 0 | 6 |
| 7th (↓) | 1 | 1 | 1 | 7 |
| 8th (↓) | 0 | 0 | 0 | 0 |

8 States $2^n = 2^3 = 8$

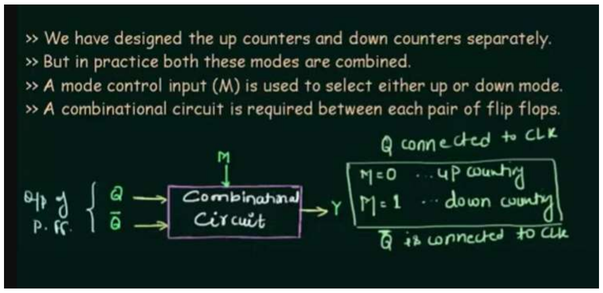
Maximum count:- $\boxed{2^n - 1} = 8 - 1 = 7$



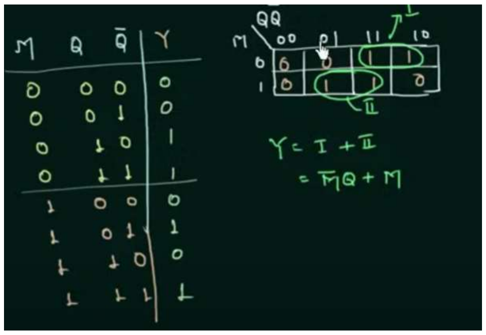
3 BIT DOWN COUNTER


Logic 1
CLK
$J_A$ $Q_A$ A $K_A$ $\bar{Q}_A$
$J_B$ $Q_B$ B $K_B$ $\bar{Q}_B$
$J_C$ $Q_C$ C $K_C$ $\bar{Q}_C$
Logic1

UPDOWN COUNTER



>> We have designed the up counters and down counters separately.
>> But in practice both these modes are combined.
>> A mode control input (M) is used to select either up or down mode.
>> A combinational circuit is required between each pair of flip flops.

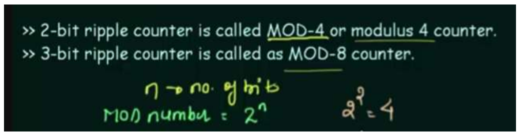


| M | Q | $\bar{Q}$ | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

$$\begin{array}{c|cccc} M \backslash Q\bar{Q} & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{array}$$

$$Y = I + II$$
$$= \bar{M}Q + M$$

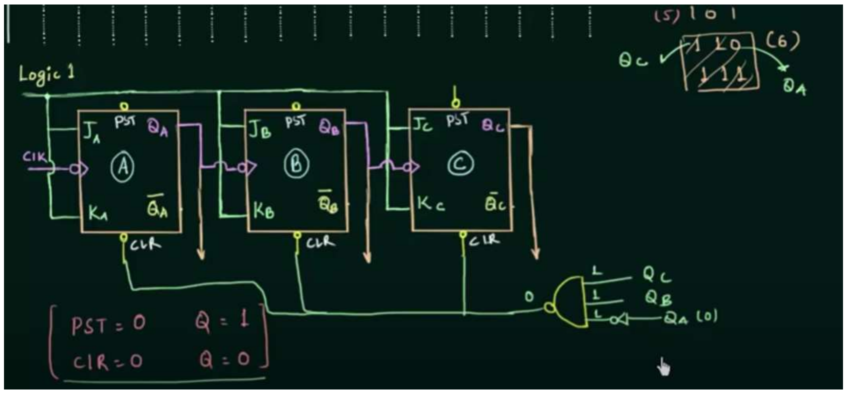
MODULUS COUNTER



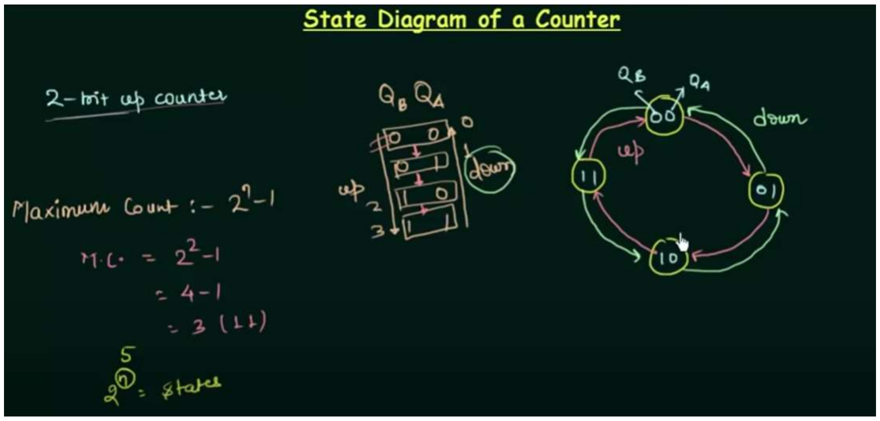
>> 2-bit ripple counter is called MOD-4 or modulus 4 counter.
>> 3-bit ripple counter is called as MOD-8 counter.

$n \rightarrow$ no. of bits
MOD number $= 2^n$ $\quad 2^2 = 4$

Modulus of counter = number of states



PST = 0   Q = 1
CLR = 0   Q = 0



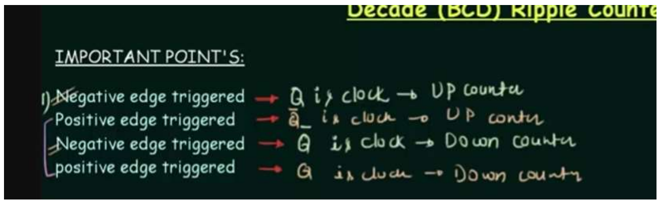
## State Diagram of a Counter

2-bit up counter

Maximum Count :- $2^n-1$

M.C. $= 2^2-1$
$= 4-1$
$= 3\ (11)$

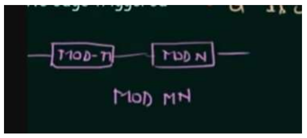
$2^{n} = 8$ states

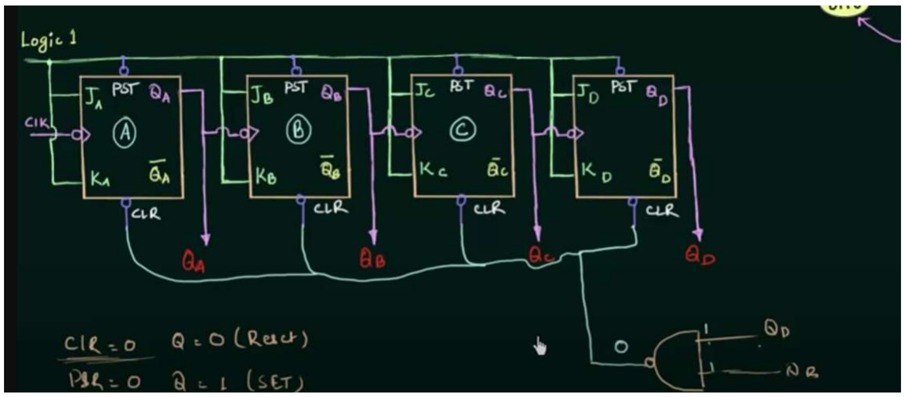


Decade (BCD) Ripple Counte

IMPORTANT POINT'S:

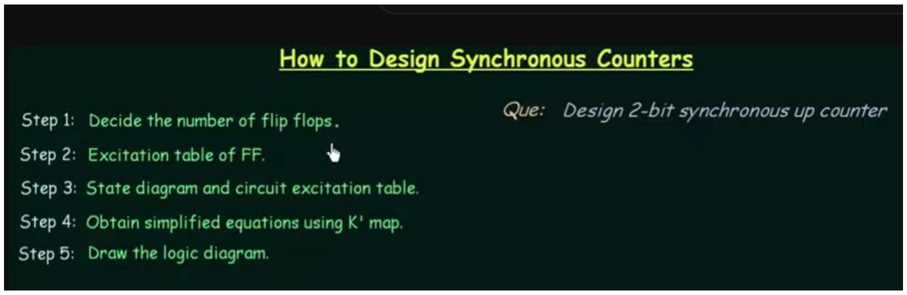
1) Negative edge triggered → Q is clock → UP counter
Positive edge triggered → $\bar{Q}$ is clock → UP conter
Negative edge triggered → $\bar{Q}$ is clock → Down counter
positive edge triggered → Q is clock → Down counter





CLR = 0   Q = 0 (Reset)
PST = 0   Q = 1 (SET)

Mod 10 counter / BCD coutner



## How to Design Synchronous Counters

Step 1: Decide the number of flip flops.

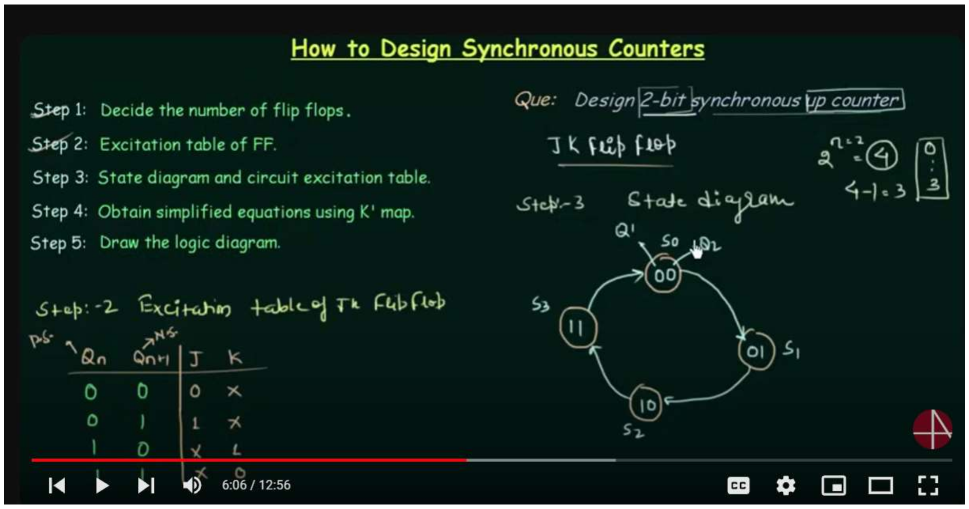
Step 2: Excitation table of FF.

Step 3: State diagram and circuit excitation table.

Step 4: Obtain simplified equations using K' map.

Step 5: Draw the logic diagram.

*Que: Design 2-bit synchronous up counter*



## How to Design Synchronous Counters

Step 1: Decide the number of flip flops.

Step 2: Excitation table of FF.

Step 3: State diagram and circuit excitation table.

Step 4: Obtain simplified equations using K' map.

Step 5: Draw the logic diagram.

*Que: Design 2-bit synchronous up counter*

JK flip flop

$2^{n=2} = 4$ 0 : 3

$4-1=3$

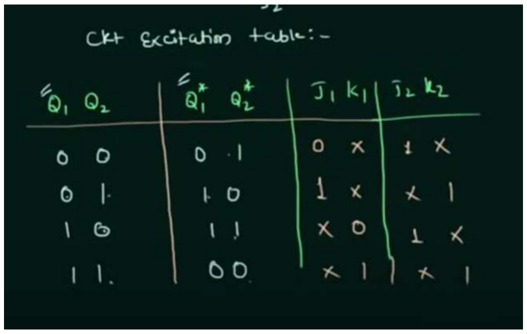
Step:-2 Excitation table of JK flip flop

| P.S. $Q_n$ | N.S. $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | x |
| 0 | 1 | 1 | x |
| 1 | 0 | x | 1 |
| 1 | 1 | x | 0 |

Step:-3 State diagram

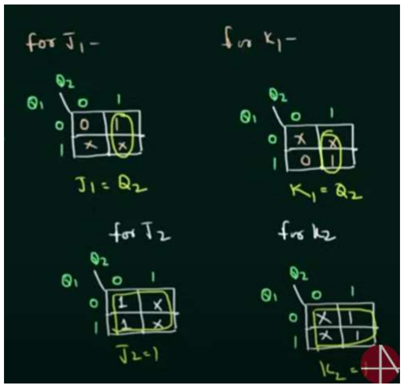
$$\begin{tikzcd}
 & 00 \arrow[dr] & \\
11 \arrow[ur] & & 01 \arrow[dl] \\
 & 10 \arrow[ul] &
\end{tikzcd}$$

Q'
S0
Q2
S1
S2
S3
6:06 / 12:56



Ckt Excitation table:-

| $Q_1$ | $Q_2$ | $Q_1^*$ | $Q_2^*$ | $J_1$ | $K_1$ | $J_2$ | $K_2$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | x | 1 | x |
| 0 | 1 | 1 | 0 | 1 | x | x | 1 |
| 1 | 0 | 1 | 1 | x | 0 | 1 | x |
| 1 | 1 | 0 | 0 | x | 1 | x | 1 |



for $J_1$-

| $Q_1 \backslash Q_2$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | x | x |

$J_1 = Q_2$

for $K_1$-

| $Q_1 \backslash Q_2$ | 0 | 1 |
|---|---|---|
| 0 | x | x |
| 1 | 0 | 1 |

$K_1 = Q_2$

for $J_2$

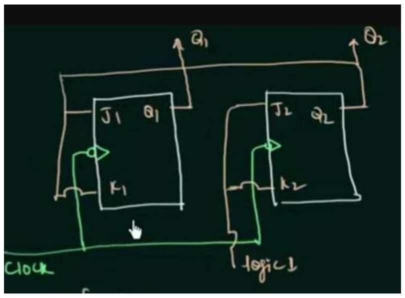
| $Q_1 \backslash Q_2$ | 0 | 1 |
|---|---|---|
| 0 | 1 | x |
| 1 | 1 | x |

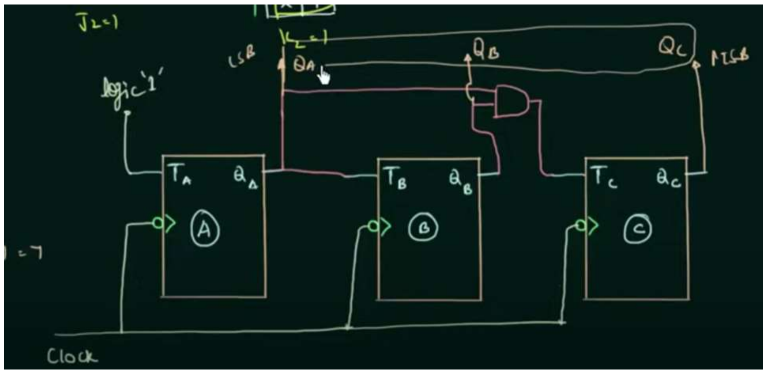
$J_2 = 1$

for $K_2$

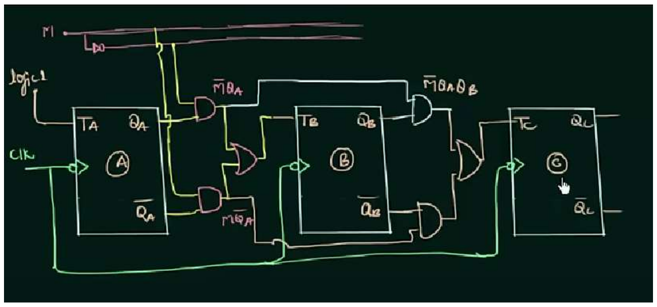
| $Q_1 \backslash Q_2$ | 0 | 1 |
|---|---|---|
| 0 | x | 1 |
| 1 | x | 1 |

$K_2 =$

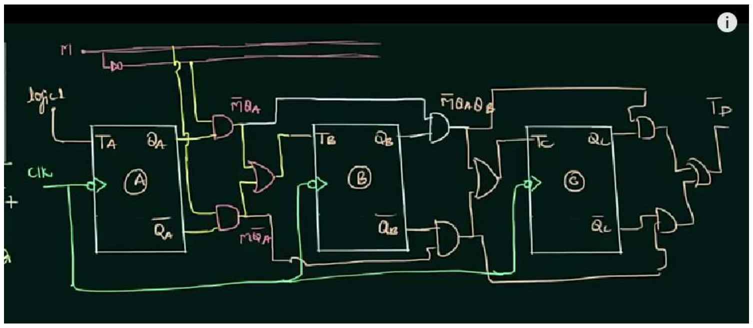




3 bit sync counter



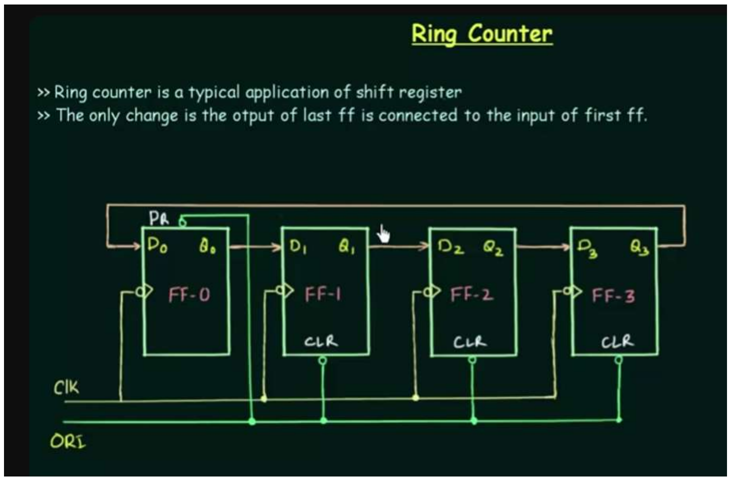
3 bit updown sync counter



4 bit updown counter

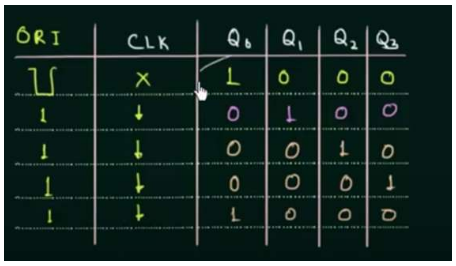
RING COUNTER

Difference from register = output of the last FF is connected to the input of the first FF

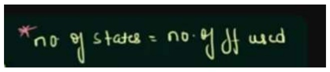


## Ring Counter

>> Ring counter is a typical application of shift register
>> The only change is the otput of last ff is connected to the input of first ff.



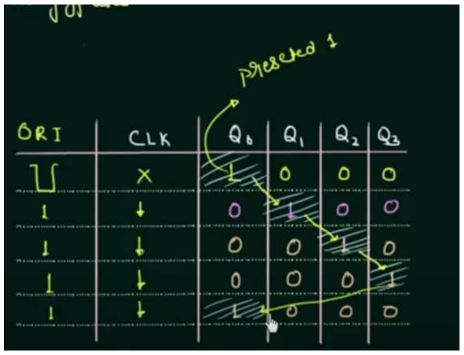
| ORI | CLK | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|---|---|
| ⊔ | X | 1 | 0 | 0 | 0 |
| 1 | ↓ | 0 | 1 | 0 | 0 |
| 1 | ↓ | 0 | 0 | 1 | 0 |
| 1 | ↓ | 0 | 0 | 0 | 1 |
| 1 | ↓ | 1 | 0 | 0 | 0 |



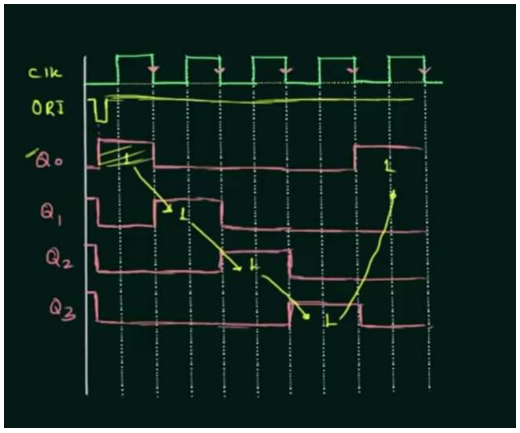
*no of states = no. of ff used

ORI = overidding inputs

Number of states = n

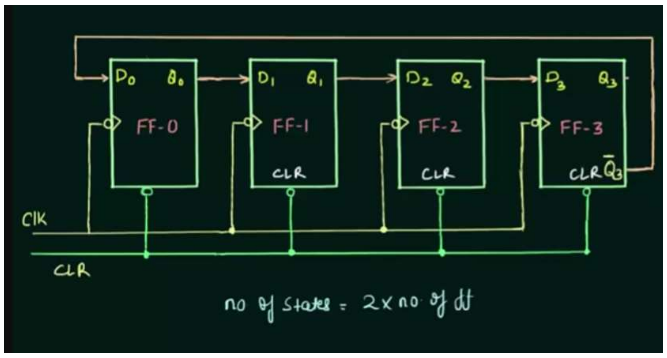


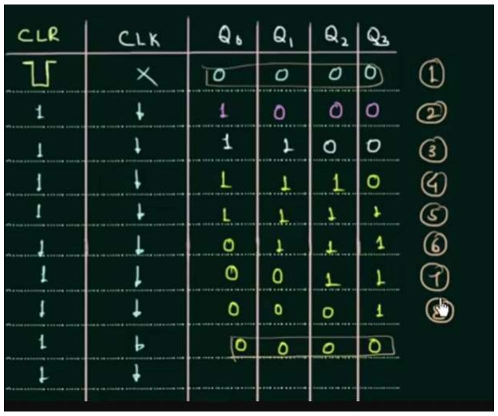
| ORI | CLK | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|---|---|
| ⊔ | × | 1 | 0 | 0 | 0 |
| 1 | ↓ | 0 | 1 | 0 | 0 |
| 1 | ↓ | 0 | 0 | 1 | 0 |
| 1 | ↓ | 0 | 0 | 0 | 1 |
| 1 | ↓ | 1 | 0 | 0 | 0 |



Johnson Counter

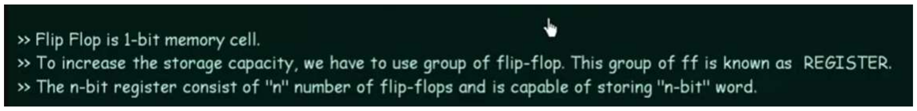
Better than the ring counter due to more number of states
Number of states = 2n



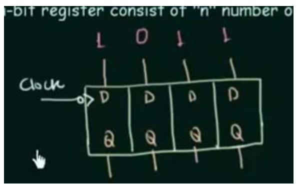


| CLR | CLK | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ | |
|---|---|---|---|---|---|---|
| ⊔ | × | 0 | 0 | 0 | 0 | 1 |
| 1 | ↓ | 1 | 0 | 0 | 0 | 2 |
| 1 | ↓ | 1 | 1 | 0 | 0 | 3 |
| 1 | ↓ | 1 | 1 | 1 | 0 | 4 |
| 1 | ↓ | 1 | 1 | 1 | 1 | 5 |
| 1 | ↓ | 0 | 1 | 1 | 1 | 6 |
| 1 | ↓ | 0 | 0 | 1 | 1 | 7 |
| 1 | ↓ | 0 | 0 | 0 | 1 | 8 |
| 1 | ↓ | 0 | 0 | 0 | 0 | |
| 1 | ↓ | | | | | |

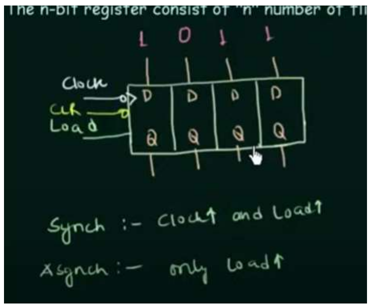
REGISTER



>> Flip Flop is 1-bit memory cell.
>> To increase the storage capacity, we have to use group of flip-flop. This group of ff is known as REGISTER.
>> The n-bit register consist of "n" number of flip-flops and is capable of storing "n-bit" word.

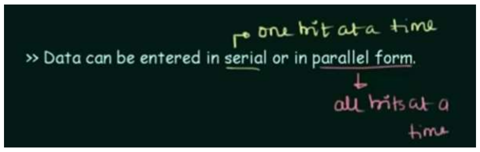


Clock is applied to all directly
This can be a problem as the data stored will get changed as clock pulse changes


The n-bit register consist of "n" number of

Synch :- Clock↑ and Load↑

Hence load are applied

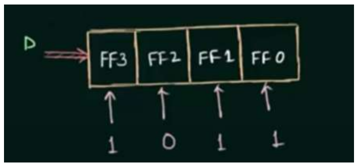


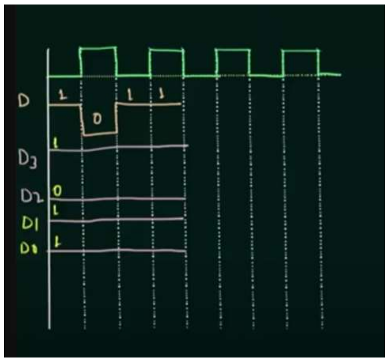
>> Data can be entered in serial or in parallel form.

one bit at a time

all bits at a time

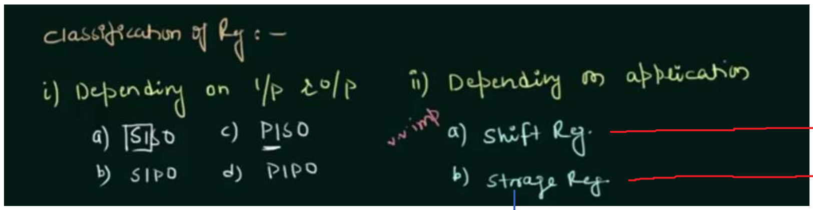
serial





parallel

Serail = temporal code
Parallel = special code



classification of Reg :-

i) Depending on I/p & O/p

a) SISO c) PISO

b) SIPO d) PIPO

ii) Depending on application

v.v. imp a) Shift Reg.

b) Storage Reg.

Shifts the data

Stores the data

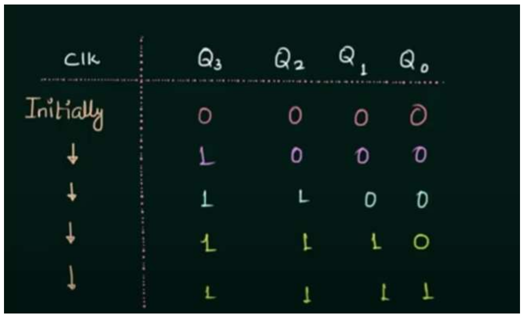
PIPO

Serial IN and Serial OUT

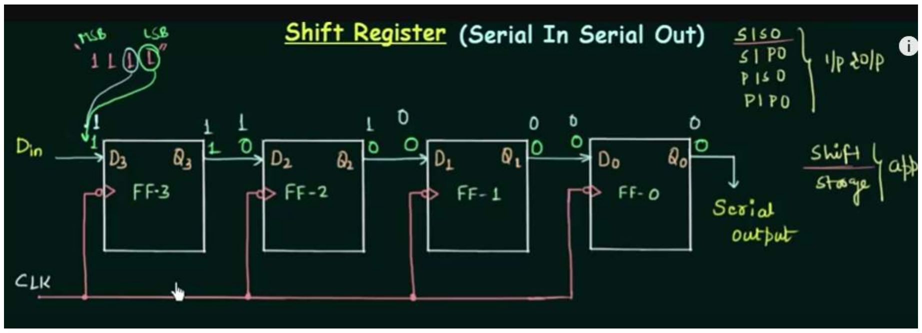
On input = 1111
The clock stores on rising edge

Beneficial for long distance transmission
Only one input line is required

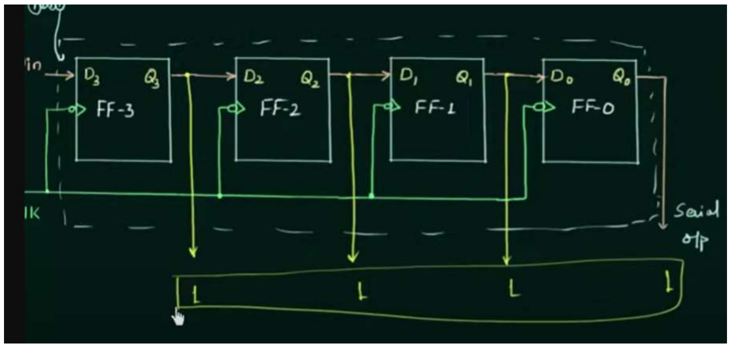
Number of clock pulses required to store data = 4
But disadv is that we need 4 pulses to output the data as well



| CLK | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
|---|---|---|---|---|
| Initially | 0 | 0 | 0 | 0 |
| ↓ | 1 | 0 | 0 | 0 |
| ↓ | 1 | 1 | 0 | 0 |
| ↓ | 1 | 1 | 1 | 0 |
| ↓ | 1 | 1 | 1 | 1 |

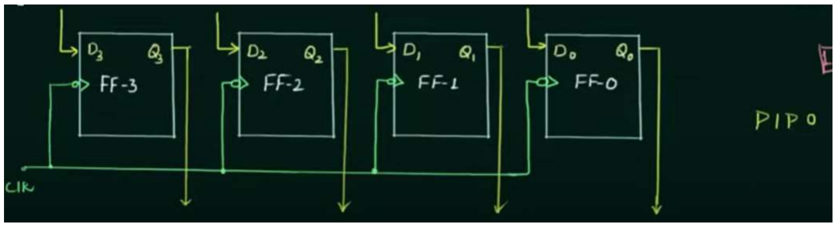


Serial IN and Parallel OUT



4 clock pulses to store the data
1 clock pulse to output the data

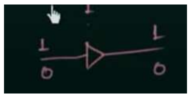
Parallel IN and Parallel OUT



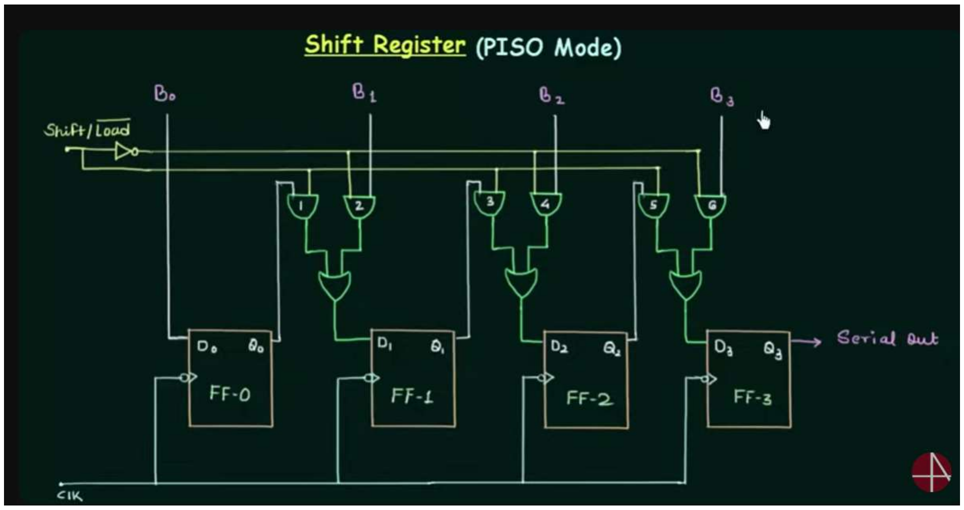
Input the number before clocks falling edge and then make clock =0
This stores the data
Output is given parallely

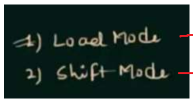
1 clock pulse to store the data

Storage register / buffer regsiter



Buffer
Input in = output out

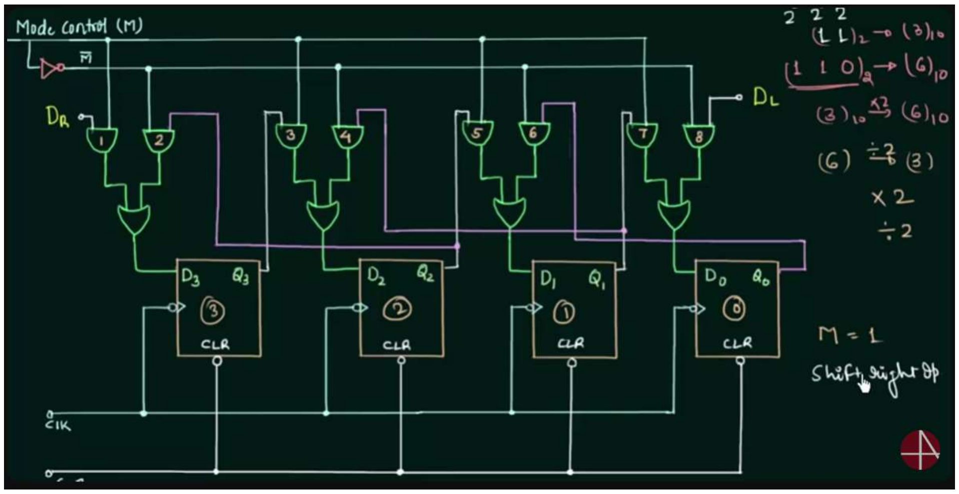


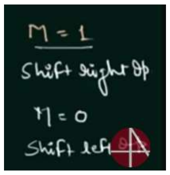

1) Load Mode
2) Shift Mode

For input. Works when load =0

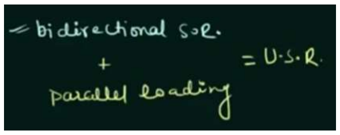
Shifts the data towards the right. Works when load =1

BIDIRECTIONAL SHIFT REGISTER



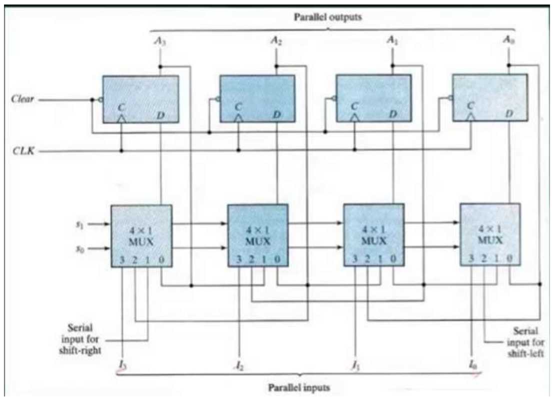


M = 1
Shift right op
M = 0
Shift left op

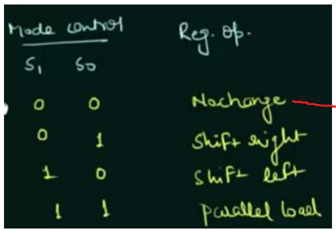


bidirectional S.R. + parallel loading = U.S.R.

UNIVERSAL SHIFT REGISTER



All SISO SIPO PIPO PISO are possible in this , hence universal



| Mode control $S_1$ | $S_0$ | Reg. op. |
|---|---|---|
| 0 | 0 | No change |
| 0 | 1 | Shift right |
| 1 | 0 | Shift left |
| 1 | 1 | parallel load |

Save data in flip flops

FSM = fininte state machine

Flip flop are edge sensitive while lathes are level sensitive

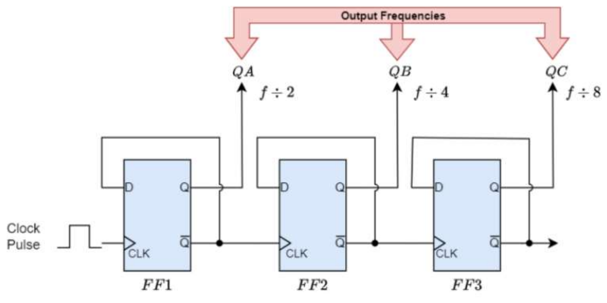


Fig-1: Freq Division Counter