

Digital logic and design circuit

① Half Adder

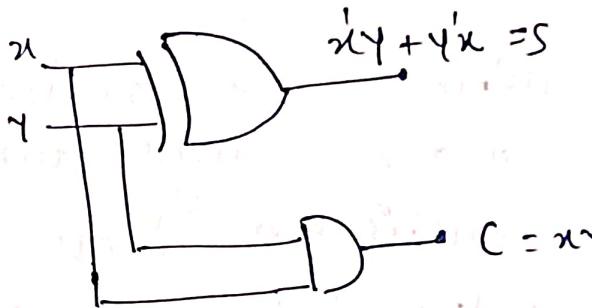
Truth Table

| x | y | carry | sum |
|---|---|-------|-----|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

UNIT-II

$$\begin{aligned} \text{sum} &= x'y + xy' = x \oplus y \\ \text{carry} &= xy \end{aligned}$$

circuit:-



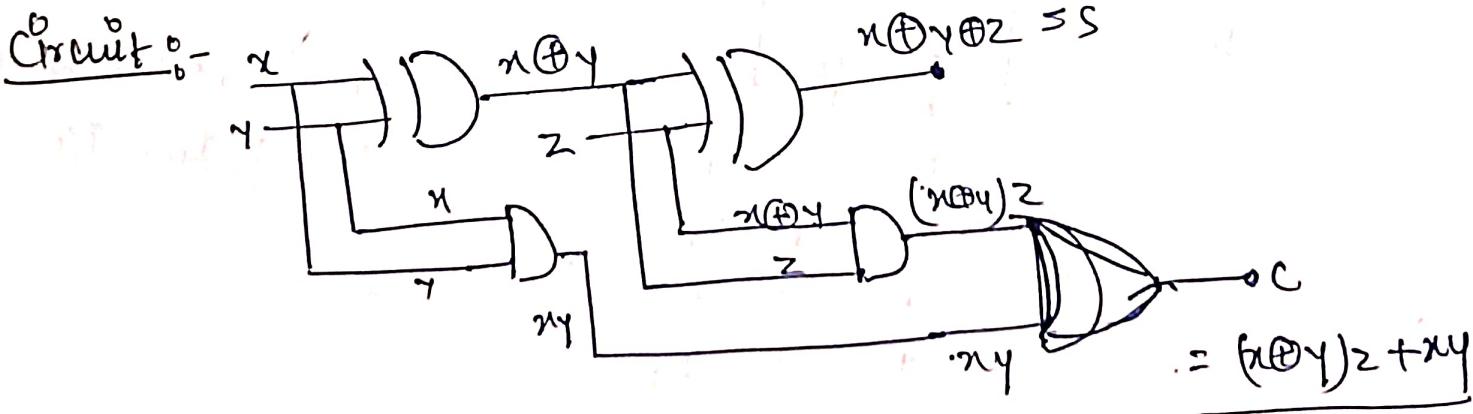
② Full Adder

| x | y | z | Carry | sum |
|---|---|---|-------|-----|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

$$\text{sum} = x \oplus y \oplus z$$

$$\text{carry} = xy + (x \oplus y)z$$

Circuit:-



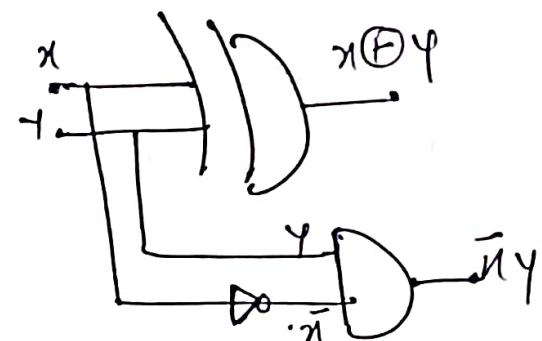
③ Half substrator:-

| x | y | Borrow sub | |
|---|---|------------|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

$$S = x \oplus y$$

$$B = x' y$$

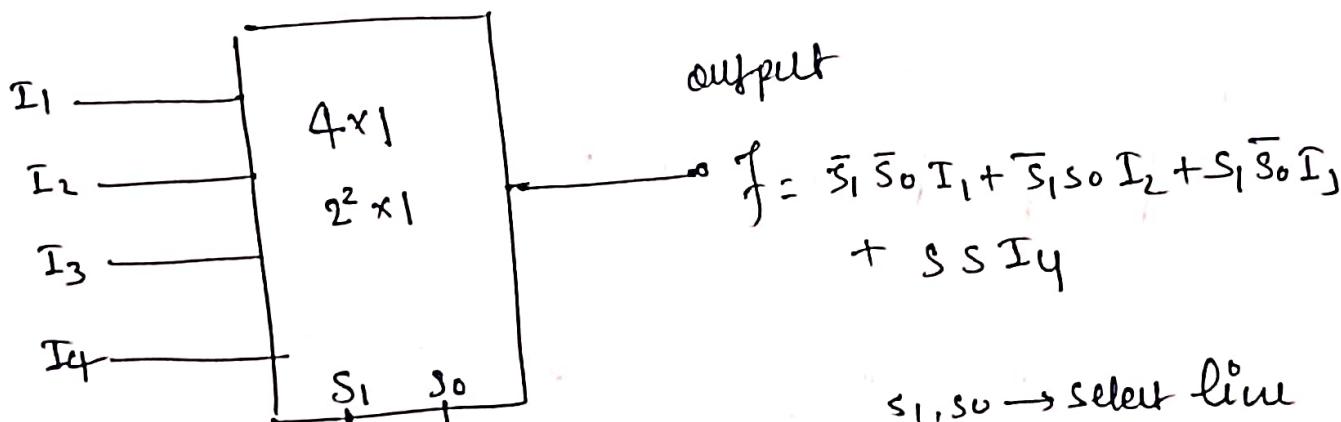
Circuit



Multiplexer

- A multiplexer is a combinational circuit that has 2^n input lines and a single output line [$2^n : 1$]
- A multiplexer is an electronic switch that can connect 1 out of 'n' input to an output
- It is functionally complete i.e all Boolean functions can be realized using one multiplexers without any other gate

⇒



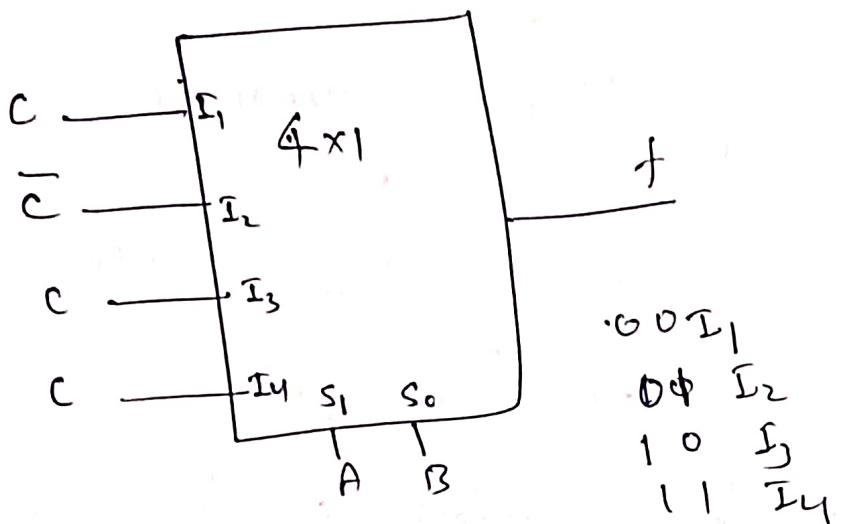
| | | |
|---|---|----------------|
| 0 | 0 | I ₁ |
| 0 | 1 | I ₂ |
| 1 | 0 | I ₃ |
| 1 | 1 | I ₄ |

$I_1, I_2, I_3, I_4 \rightarrow \text{i/p}$

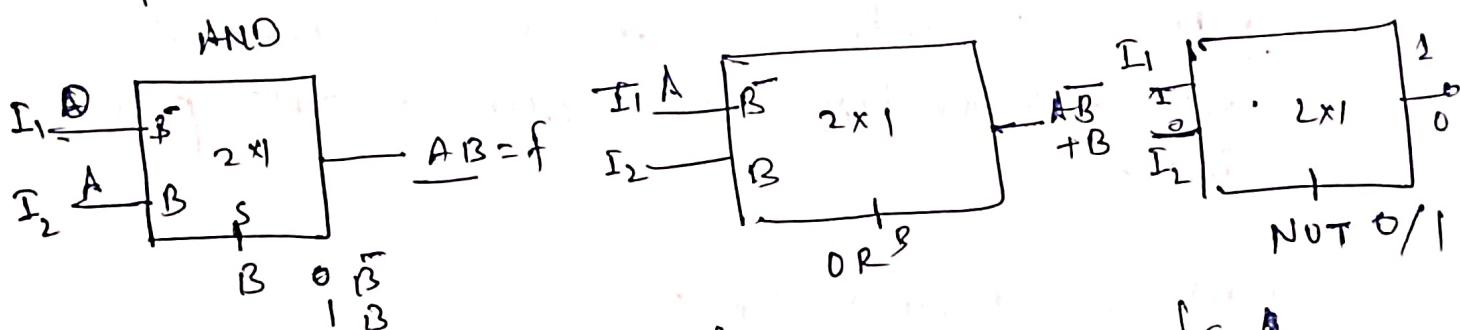
Implementing function using MUX

$$f(A, B, C) = \sum(1, 2, 5, 7) = \bar{A}\bar{B}C + \cdot\bar{A}BC + \cdot A\bar{B}C + ABC$$

| | | | |
|-------|-------|-------|-------|
| 0 0 1 | 0 1 0 | 1 0 1 | 1 1 1 |
|-------|-------|-------|-------|



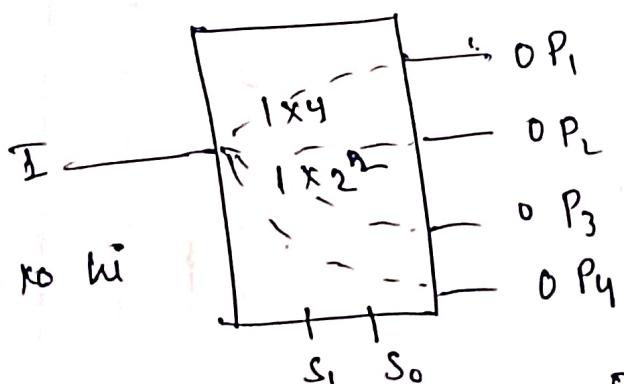
* How MUX are functionally complete



| A | B | f |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$f = AB$$

* Demultiplexers :- $(1 : 2^n)$ Select line

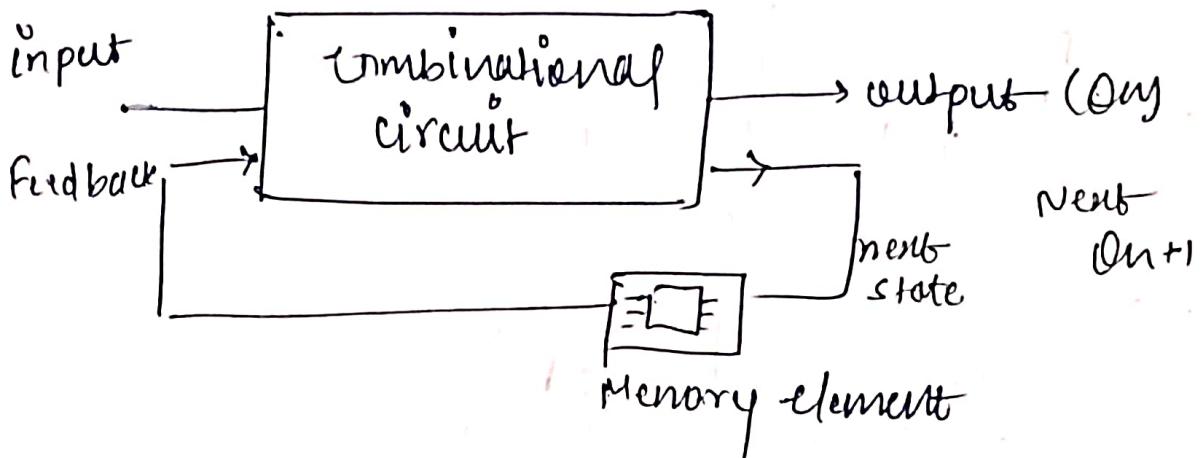


| OP_1 | OP_2 | OP_3 | OP_4 |
|--------|--------|--------|--------|
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

* EK bar me ek ko hi dega

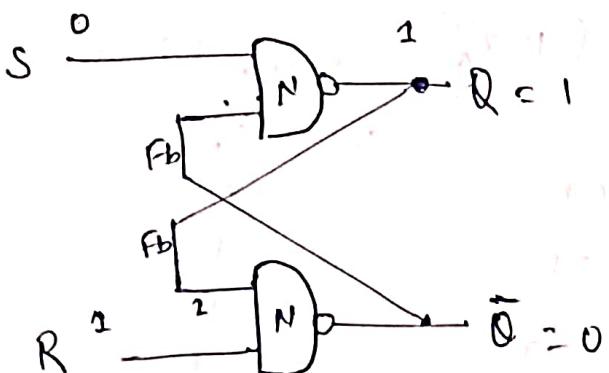
UNIT-3

(Sequential circuit Introduction)



Defⁿ: - A sequential circuit is a digital circuit that stores and processes information over time using memory elements, and where the outputs are dependent on both the current and previous inputs.

* SR latch using NAND GATE:-



$R \rightarrow$ Reset
 $S \rightarrow$ Set

$$(\overline{AB} = Y)$$

NAND

| S | R | $\cdot Q(n+1)$ (next) |
|---|---|-----------------------|
| 0 | 0 | Invalid state |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | hold |

1 if P is zero
0 if P is one

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

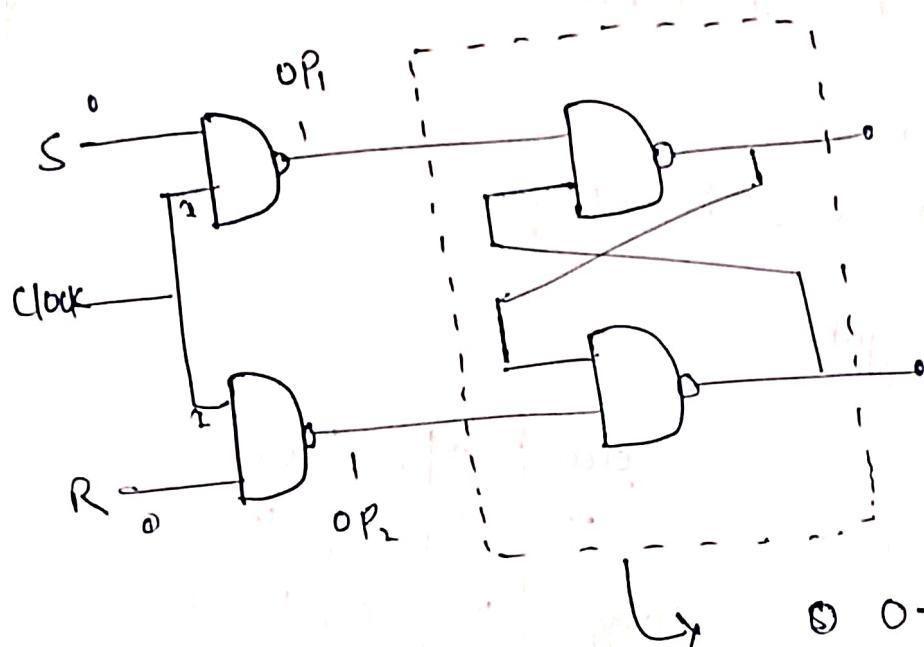
$$\textcircled{2} \quad S=1, R=0 \Rightarrow \bar{Q}=1, Q=0$$

$$\textcircled{3} \quad S=1, R=1 \Rightarrow \overline{1 \cdot \bar{Q}} = \bar{1} + \bar{Q} = 0 + 0 = Q, \overline{1 \cdot Q} = \bar{1} + Q = 1 + 1 = \bar{Q}$$

NAND
GATE
KEY

④ $S=0, R=0 \Rightarrow Q=1, \bar{Q}=1 \Rightarrow Q \neq \bar{Q}$ (invalid)

* SR flip flop using NAND GATE:



Truth Table

| Clock | S | R | Out1 |
|--------------|---|---|-----------|
| NOT Triggers | X | X | 0h |
| Triggers | 0 | 0 | Hold |
| " | 0 | 1 | 0 (Reset) |
| " | 1 | 0 | 1 (Set) |
| " | 1 | 1 | invalid |

① 0 - Invalid

0 1 → 1

1 0 → 0

1 1 → Hold

② clock = 1

$S=0, R=0$

output = 1, 1 → hold

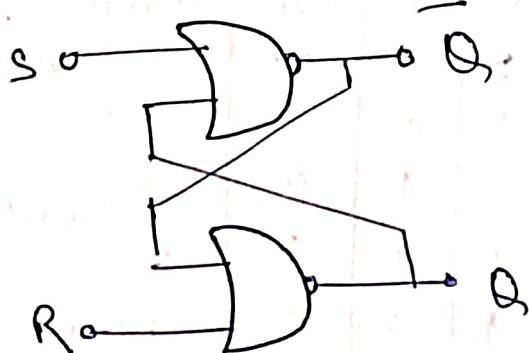
③ clock = 1

$S=0, R=1 \Rightarrow$ output $\Rightarrow OP_1=1, OP_2=0$

④ clock = 1, $S=1, R=0 \Rightarrow$ output $\Rightarrow OP_1=0, OP_2=1$

⑤ $S=1, R=1 \Rightarrow OP_1=0, OP_2=0 \Rightarrow$ invalid

* S-R latch using NOR GATE:-



| S | R | Q(out) |
|-----|---|-----------------------------|
| → 0 | 0 | Hold (No change) hold state |
| → 0 | 1 | 0 |
| → 1 | 0 | 1 |
| → 1 | 1 | invalid |

Q out
value
final state
final value

Ek bhi 0^o / P(1) आ गयी हो
तो zero output (Don't care)

\therefore NOR GATE $y = \overline{A+B}$

$$\begin{pmatrix} 0 & 0 \rightarrow 1 \\ 0 & 1 \rightarrow 0 \\ 1 & 0 \rightarrow 0 \\ 1 & 1 \rightarrow 0 \end{pmatrix}$$

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

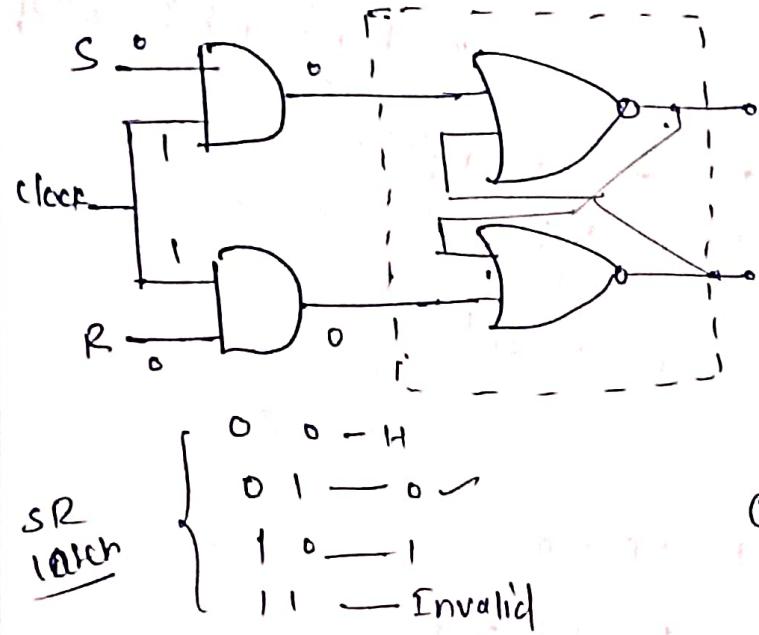
$$\bar{Q} = \overline{S+Q} = \overline{0+0} = 1$$

$$\textcircled{2} S=1, R=0 \Rightarrow \bar{Q}=0 \Leftrightarrow Q=0, \bar{Q}=1$$

$$Q = \overline{R+\bar{Q}} = \overline{0+0} = 1$$

$$\textcircled{3} S=1, R=1 \Rightarrow Q=0, \bar{Q}=0 \rightarrow \text{invalid}$$

SR Flip Flop using NOR gate



| Clock | S | R | Q(out) |
|-----------|---|---|---------|
| X | X | X | Qn |
| Triggered | 0 | 0 | Hold |
| " | 0 | 1 | 0 |
| " | 1 | 0 | 1 |
| " | 1 | 1 | Invalid |

① $\text{Clock}=1$ (Triggers)
 $\overline{S=0}, \overline{R=0} \rightarrow \text{Hold}$

$S=0, R=1 \rightarrow 01 \rightarrow 0$

$S=1, R=0 \rightarrow 10 \rightarrow 1$

$S=1, R=1 \rightarrow 11 \rightarrow \text{Invalid}$

NOTE:- SR flip flop using NAND or NOR gate truth table is same

Characteristic table:-

| S | R | Out |
|---|---|---------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Invalid |

| S | R | Q(in) | Q(out) |
|----|---|-------|------------------|
| 0 | 0 | 0 | 0 { Hold } |
| 1 | 0 | 0 | 1 { 1 } |
| 2. | 0 | 1 | 0 |
| 3. | 0 | 1 | 1 { Reset } |
| 4. | 1 | 0 | 0 { Set } |
| 5. | 1 | 0 | 1 { 1 } |
| 6. | 1 | 1 | 0 { } |
| 7. | 1 | 1 | 1 { Doubt case } |

* Characteristic eqn:-
using K-map (3) var

| \bar{Q}_n | $\bar{Q}_n \bar{Q}_n$ | $\bar{Q}_n Q_n$ | $Q_n \bar{Q}_n$ |
|-------------|-----------------------|-----------------|-----------------|
| \bar{S} | 1 | 3 | 2 |
| \bar{S} | 1 | 3 | X |
| S | 4 | 5 | X |
| S | 6 | 7 | L |

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

\Rightarrow char. eqn of SR flip flop.

* Excitation table:-

$\Rightarrow Q_n, Q_{n+1}$ ko dekna hai, unstable value par don't care (0) बताती हैं

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

\Rightarrow See characteristic table

Step:- SR LL

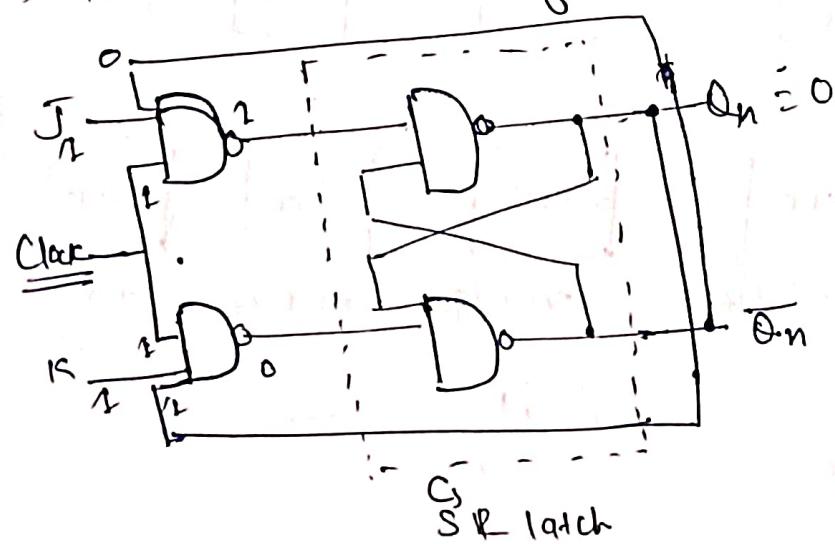
\downarrow
SR flip flop

\downarrow
TT

\downarrow
CT \rightarrow ET

JK flip flop:-

\Rightarrow almost 75% same of the SR FF



case I, $Q_n = 1$ (let), $\bar{Q}_n = 0$

J=1, K=1, clock=1

op \Rightarrow 1 and 0 see; SR latch

$$Q_{n+1} = 0$$

case-II $Q_n = 0$, $\bar{Q}_n = 1$

op \Rightarrow 0 and 1

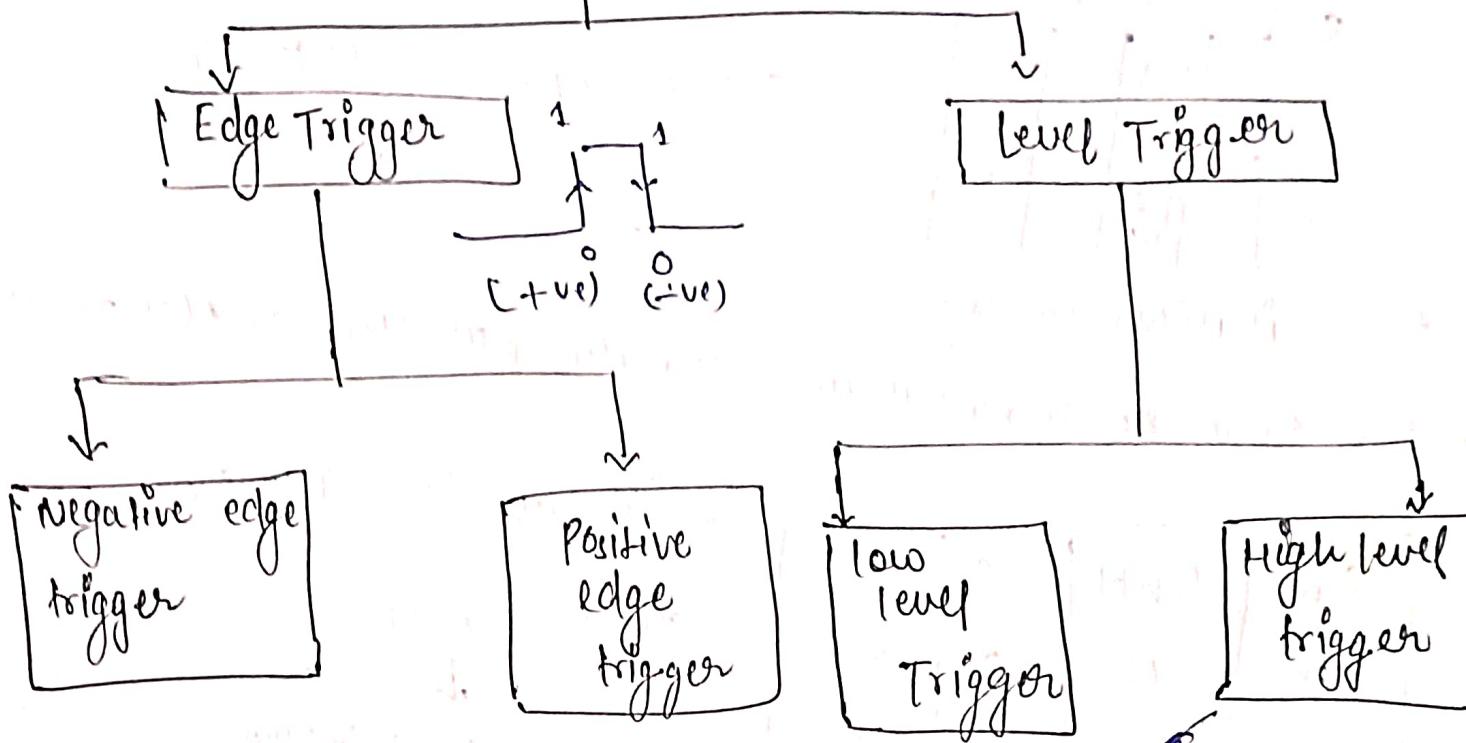
$$Q_{n+1} = 1$$

| J | K | Q_{n+1} |
|---|---|-----------|
| 0 | 0 | Hold |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

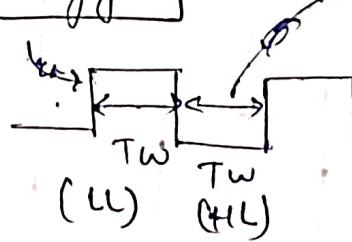
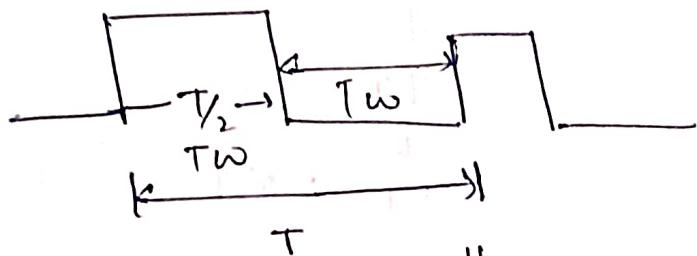
Reset
Set

\bar{Q}_n (Toggle)

Types of Triggering



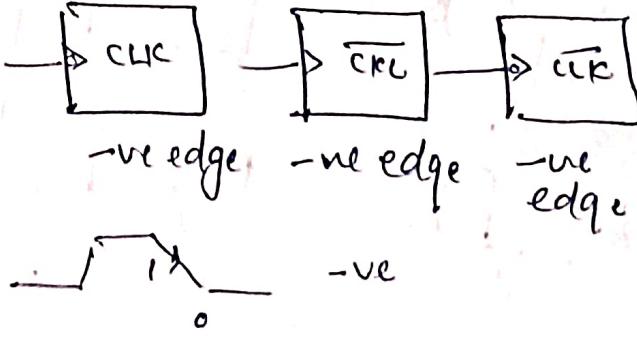
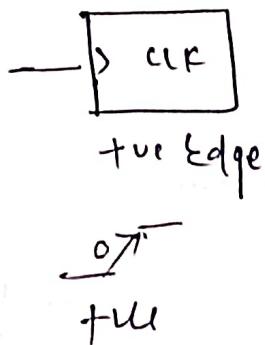
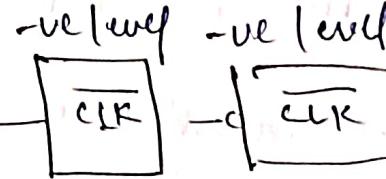
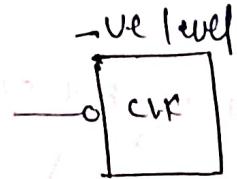
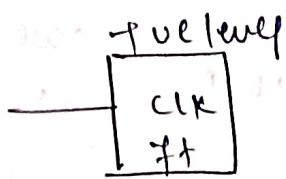
⇒



Representation

$$T_W = T_1/2$$

$$T = 2 \times T_W$$



JK Flip flop characteristic & Excitation Table

① characteristics

| | J | K | Q_n | Q_{n+1} |
|---|---|---|-------|-----------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 |

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

⇒ characteristic eqn:-

| J | $\bar{K}Q_n$ | $\bar{K}Q_n$ | KQ_n | KQ_n |
|---|--------------|--------------|--------|--------|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

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

charact. eqn of JK FF

* Excitation table:-

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

* J, K do value same for stable but, different value for unstable

* Race Around condition

cond 1: level triggered JK flip flop

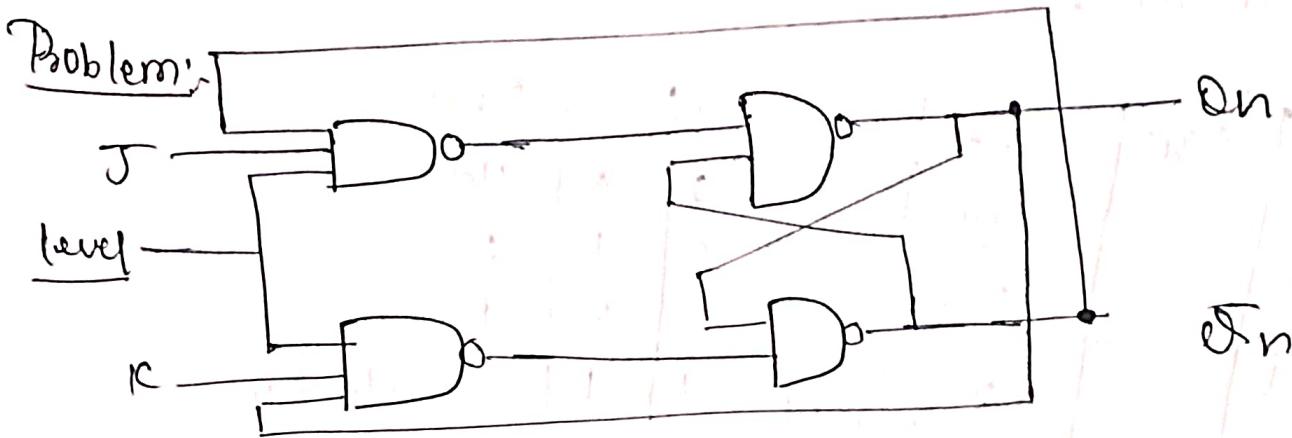
cond 2: when $J=K=1$ (Toggle mode)

cond 3: $T_w \gg T_d$

(Time delay)

→ Master slave JK flip-flop

Problem:



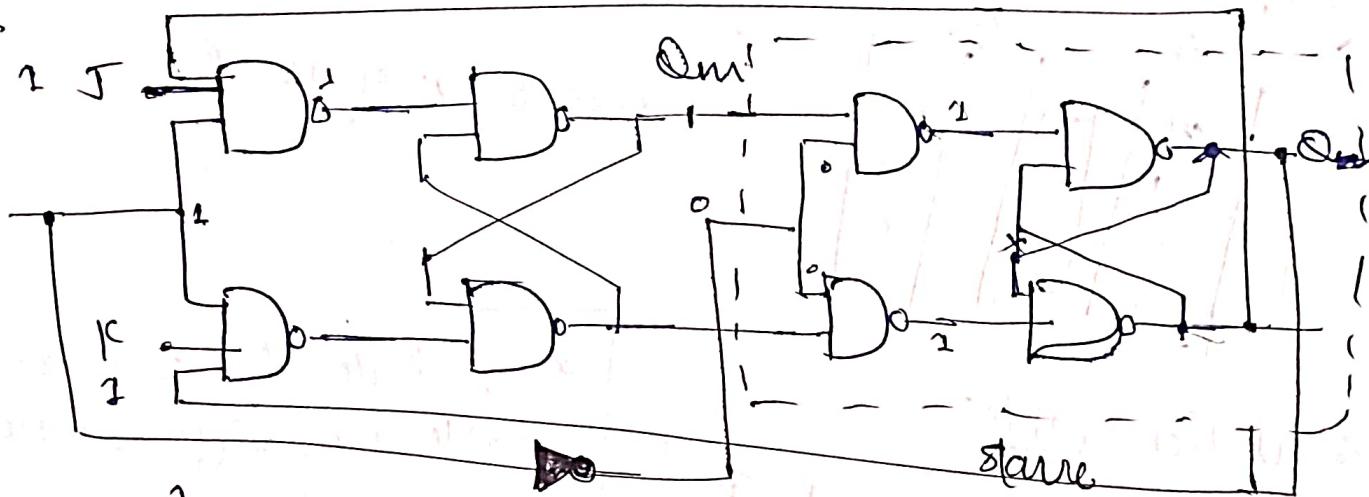
CLK → [Timing diagram showing a square wave clock signal]

J=1 → [Timing diagram showing J=1 signal]

K=1 → [Timing diagram showing K=1 signal]

Qn+1 → [Timing diagram showing the output Qn+1. A note points to it: "Race around condition"]

Qn+1-bar



CLK → [Timing diagram showing a square wave clock signal]

J=1 → [Timing diagram showing J=1 signal]

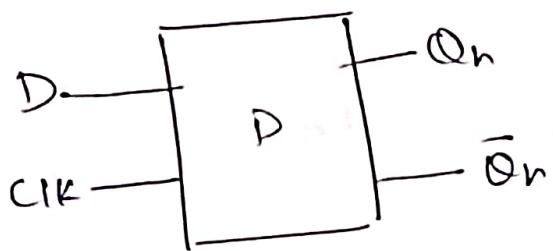
K=1 → [Timing diagram showing K=1 signal]

Qm → [Timing diagram showing the master flip-flop's output Qm]

Qn+1-bar → [Timing diagram showing the slave flip-flop's output Qn+1-bar]

→ Fach nhe Karna Bus feedback change karna hai

* D-Flip flop:-



* characteristics table

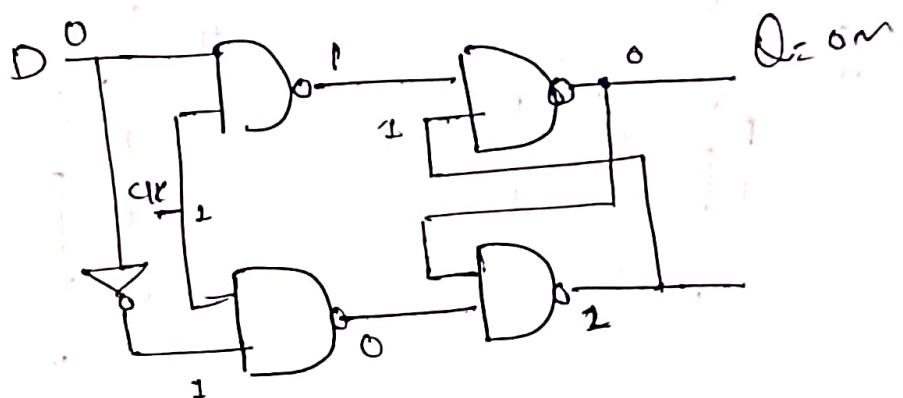
| D | Qn | Qn+1 |
|---|----|------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| D | Qn+1 |
|---|------|
| 0 | 0 |
| 1 | 1 |

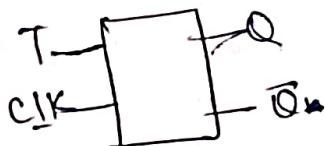
* excitation table

| Qn | Qn+1 | D |
|----|------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

* circuit diagram ($\rightarrow R$)



* T-Flip flop:- (Toggle flip flop)



| T | Qn+1 |
|---|------|
| 0 | Qn |
| 1 | Q̄n |

* characteristic Table

| T | Qn | Qn+1 |
|---|----|------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

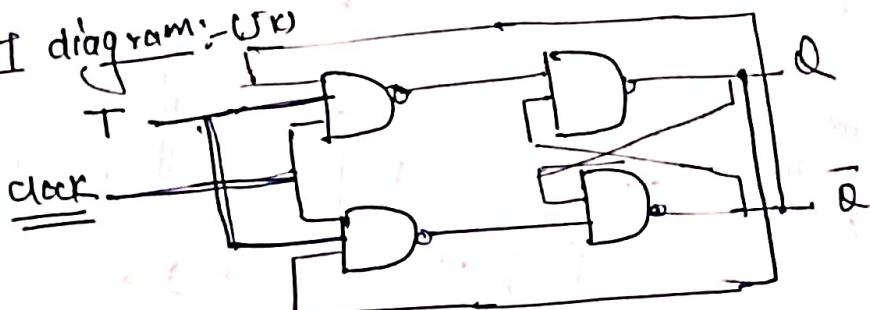
* excitation table

| Qn | Qn+1 | T |
|----|------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

eqn:-

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

* circuit diagram:-($\rightarrow R$)



* Conversion :-

① SRFF \rightarrow DFF

Given



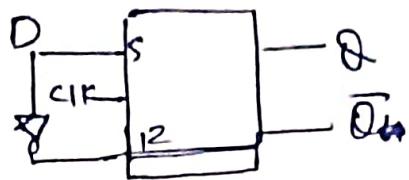
Required



Ex- \rightarrow

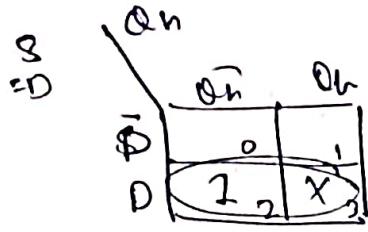
characteristic

| D | \bar{Q}_n | Q_{n+1} | S | R |
|---|-------------|-----------|---|---|
| 0 | 0 | 0 | 0 | x |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | + | 0 |
| 1 | 1 | 1 | x | 0 |

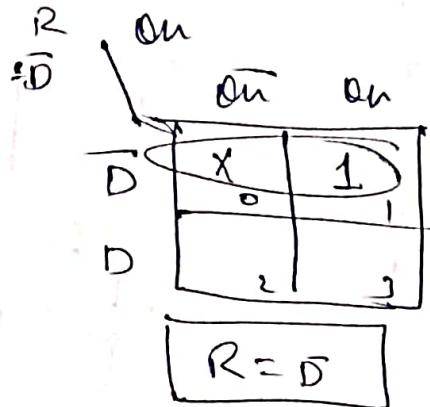


| SR - excitation | | | |
|-----------------|-----------|---|---|
| \bar{Q}_n | Q_{n+1} | S | R |
| 0 | 0 | 0 | x |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | x | 0 |

* characteristic eqn:-



$$S = D$$



$$R = \bar{D}$$

* TFF \rightarrow JF FF:-

↓

char

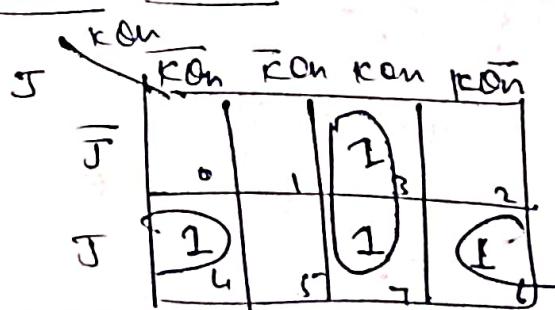
| J | K | \bar{Q}_n | Q_{n+1} | T |
|---|---|-------------|-----------|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | + | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

(Q)

* Excitation (T)

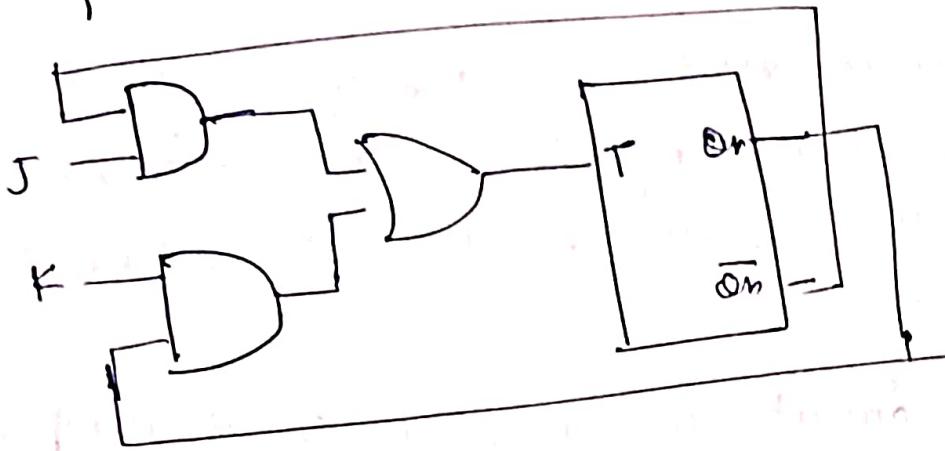
| \bar{Q}_n | Q_{n+1} | T |
|-------------|-----------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

* eqn:- (K-map)



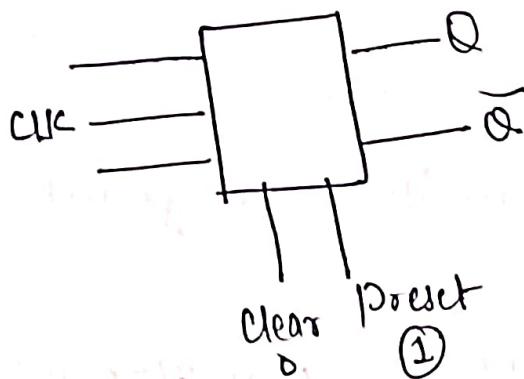
$$T = \cdot K \bar{Q}_n + J \bar{Q}_n$$

diagram:

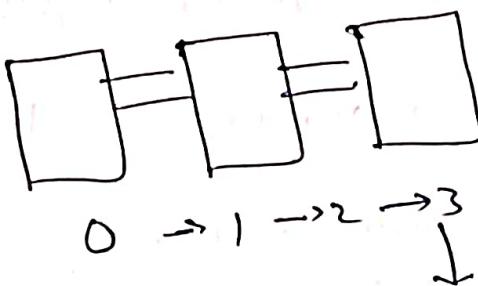


$$T = K\bar{Q} + \bar{J}Q$$

* Preset & clear inputs in flip flop (Asynchronous Inputs)

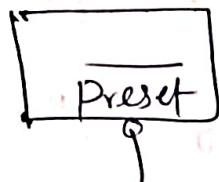
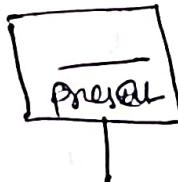
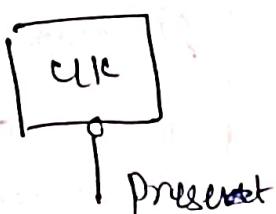
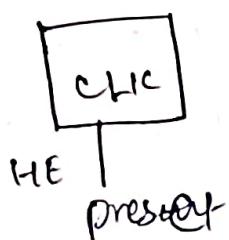
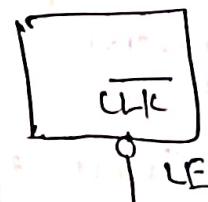
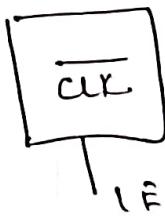
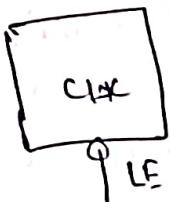
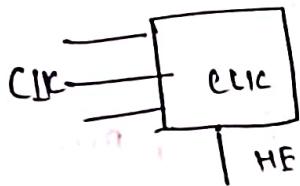


Eg:



0 → 1 → 2 → 3
0 ← 11 clear

* Presentations:



HF - High enable
LF → low enable

Set/Clear → 0
Preset → 1

* Counters :-

- ⇒ Counter is a device that stores (and sometimes display) the numbers of times a particular event or process has occurred, often in relationship to a clock.
- ⇒ A counter circuit is usually constructed of a number of flip flop connected in cascade.
- * Difference between -synch and -Asynch. sequential circuit

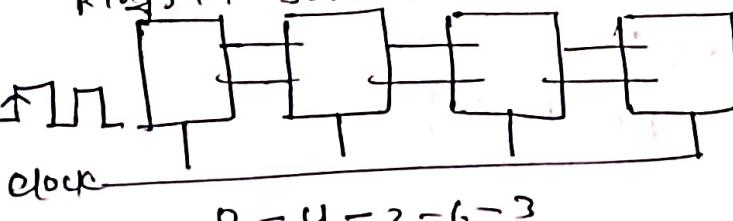
Synchronous Seq. ckt

1. These circuits are easy to design
2. A clocked flip flop acts as memory element
3. They are slower
4. The status of memory element is affected only at active edge of clock, if input is changed

Asynchronous ckt

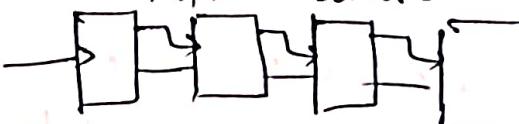
1. These circuit are difficult to design
2. an unclocked flip flop or time delay element is used as memory element
3. Faster as clock is not present
4. the status of memory elements will change any time as soon as input is changed

Ring; T-R. Johnson



Faster
complex circuit / ^{cost} High

Ripple counter



→ 0 - 1 - 2 - 3 - 4
→ 4 - 3 - 2 - 1 - 0 Slower
Easy / ~~Low~~ Low Cost

Counters (As |s)

UP counters

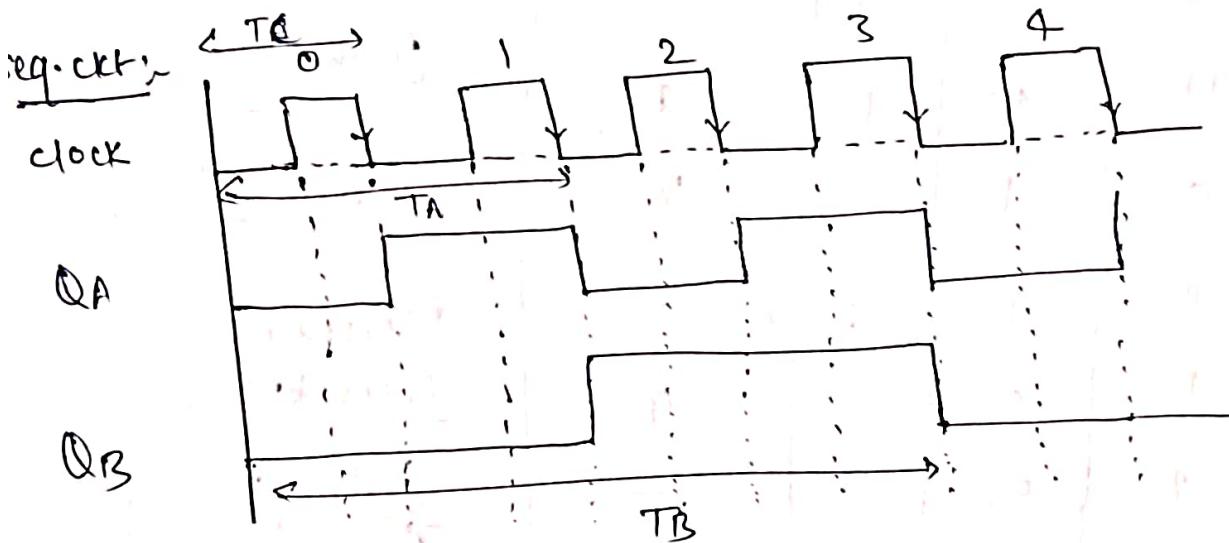
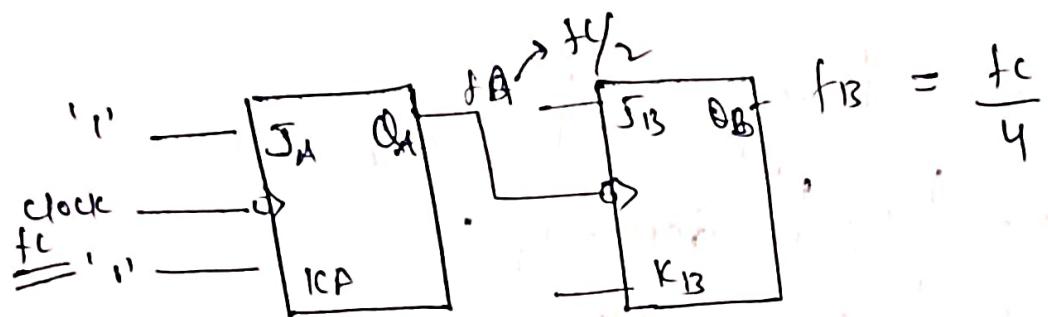
down count

Up/ down count

0-1-2-3-4-

4-3-2-1-0

Q.T. - FF as divide by 2 ckt

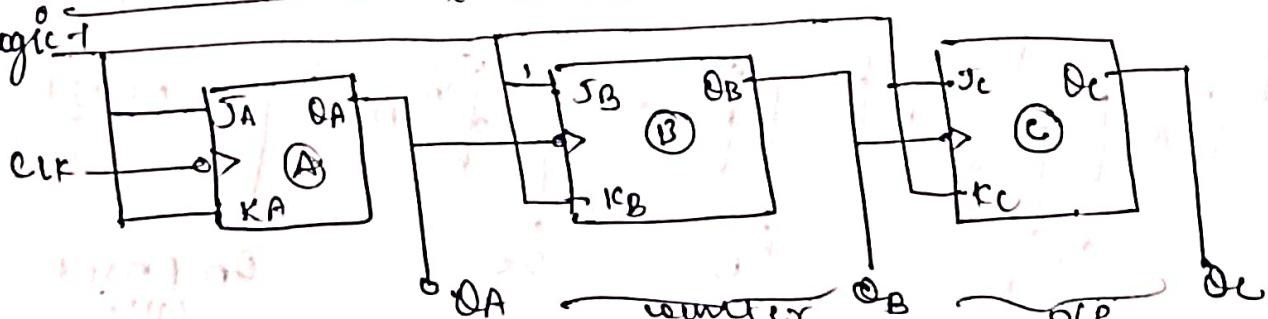


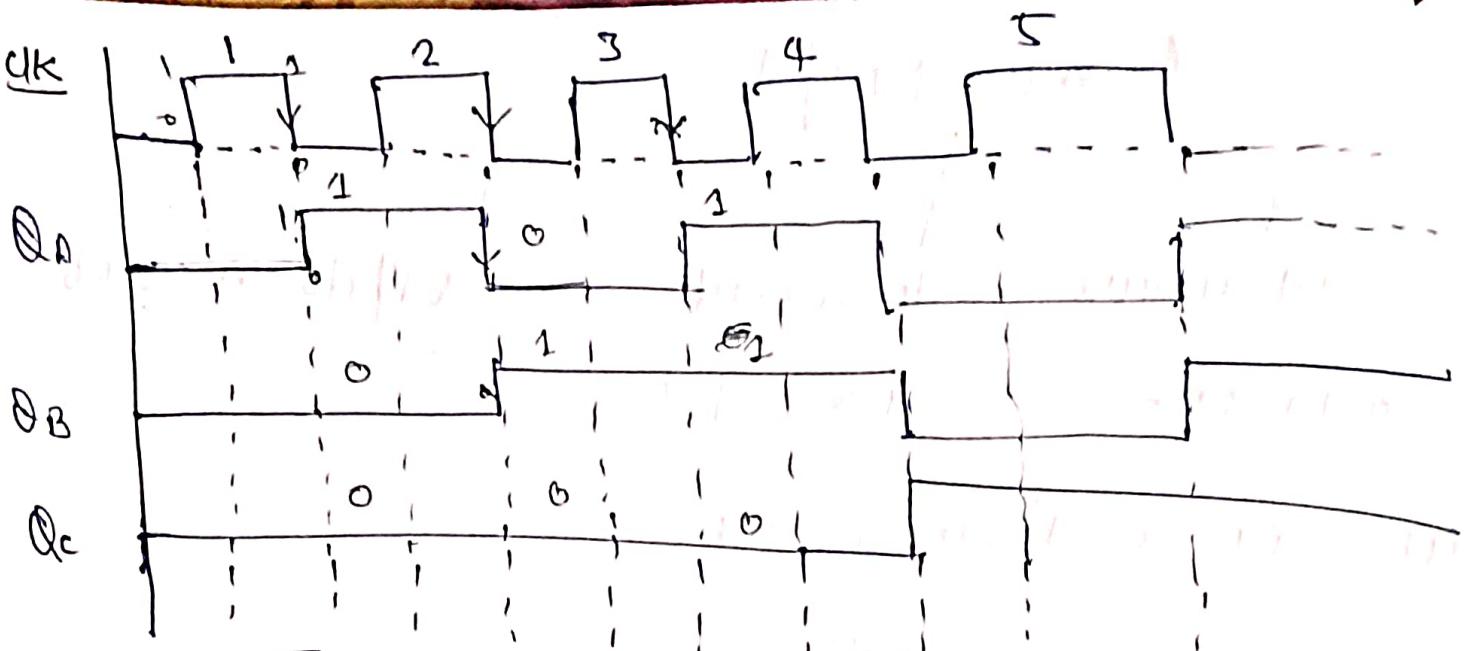
$$\Rightarrow T_A = 2T_C \Rightarrow \frac{1}{f_A} = \frac{2}{f_c} \Rightarrow f_A = \frac{f_c}{2}$$

$$T_B = 2T_A = \frac{2}{f_A} \Rightarrow \frac{1}{f_B} = \frac{2}{f_A} = \frac{1}{\frac{f_c}{2}} = \frac{f_c}{4} \Rightarrow f_B = \frac{f_c}{4}$$

3 Bit Asynchronous Up Counter

logic - 1





$T_A = k_A = 1$ (Toggling) $Q_{n+1} = \overline{Q_n}$

clock

Initially
1st (\downarrow)

2nd (\downarrow)

3rd (\downarrow)

4th (\downarrow)

5th (\downarrow)

6th (\downarrow)

7th (\downarrow)

8th (\downarrow)

| | Qc | QB | QA | decimal |
|----------------------------------|----|----|----|---------|
| Initially | 0 | 0 | 0 | 0 |
| 1 st (\downarrow) | 0 | 0 | 1 | 1 |
| 2 nd (\downarrow) | 0 | 1 | 0 | 2 |
| 3 rd (\downarrow) | 0 | 1 | 1 | 3 |
| 4 th (\downarrow) | 1 | 0 | 0 | 4 |
| 5 th (\downarrow) | 1 | 0 | 1 | 5 |
| 6 th (\downarrow) | 1 | 1 | 0 | 6 |
| 7 th (\downarrow) | 1 | 1 | 1 | 7 |
| 8 th (\downarrow) | 0 | 0 | 0 | 0 |

A state

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

Maximum
count:

$$2^n - 1 = 7$$

No of FF used

↓
Bit → Falling edge (F·r)

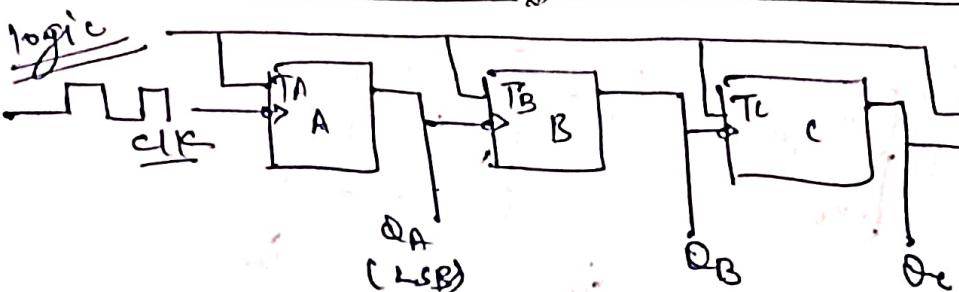
* 4-BIT Asynchronous Up counter

$$2^n = 2^4 = 16 \text{ (states)}$$

$$2^n - 1 = 16 - 1 = 15$$

↓ No of FF used

QD (MSB)
(1111)



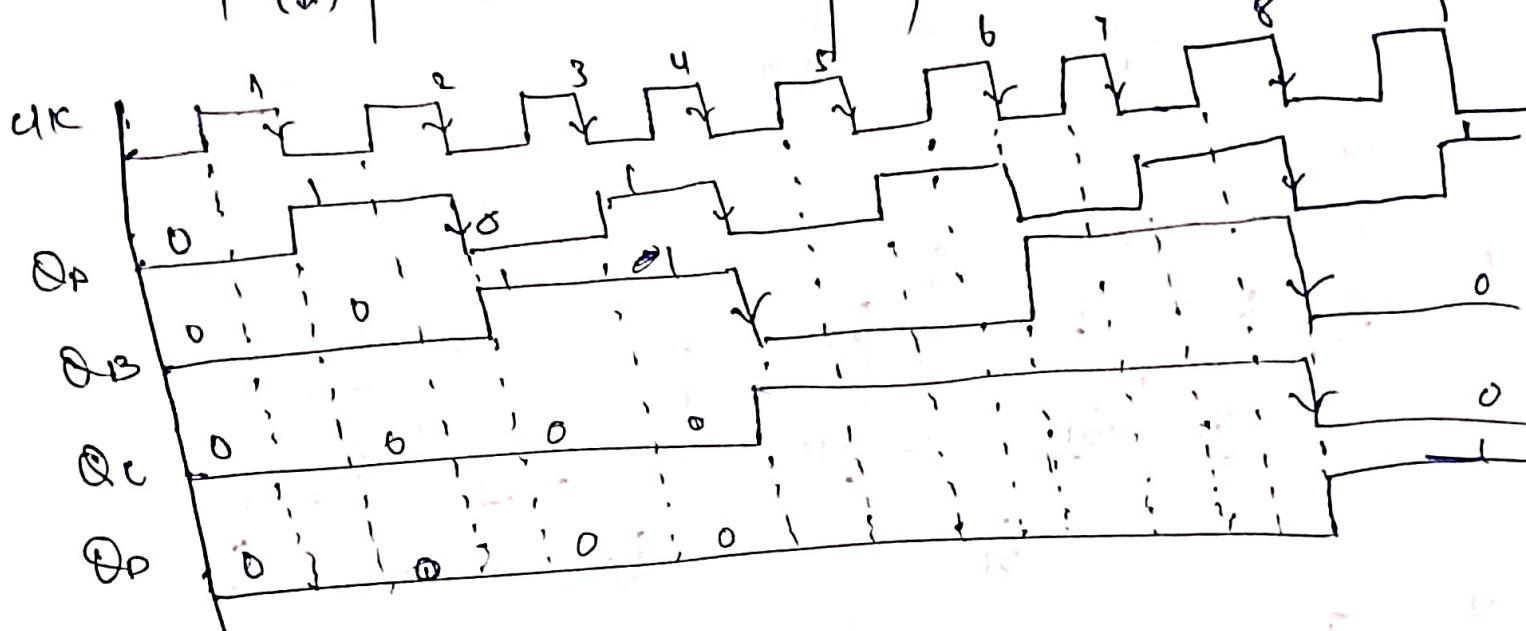
| <u>Table</u> | <u>CLK</u> | <u>QD</u> | <u>QC</u> | <u>QB</u> | <u>QA</u> | <u>Decimal</u> |
|--------------|------------|-----------|-----------|-----------|-----------|----------------|
| initial | | 0 | 0 | 0 | 0 | 0 |
| 1st (↑) | | 0 | 0 | 0 | 1 | 1 |
| 2nd (↑) | | 0 | 0 | 1 | 0 | 2 |
| 9th (↓) | | 1 | 0 | 0 | 1 | 9 |

Next falling
ko dekhna
hai

Note

wave form:

QA - done one
ice bad QA, QB
↓ down fall dekhna



* state diagram of a counter

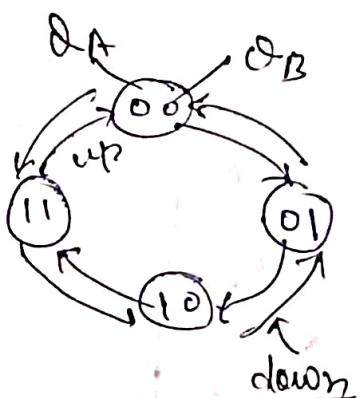
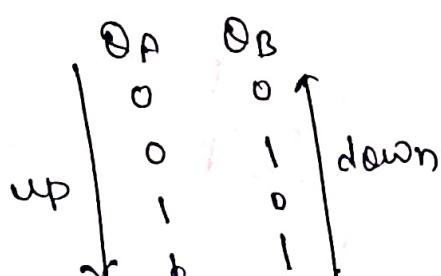
2 Bit up counter

$$\text{Maximum count} = 2^n - 1$$

$$MC = 2^2 - 1$$

$$= 4 - 1 = 3$$

$$= 3(11)$$

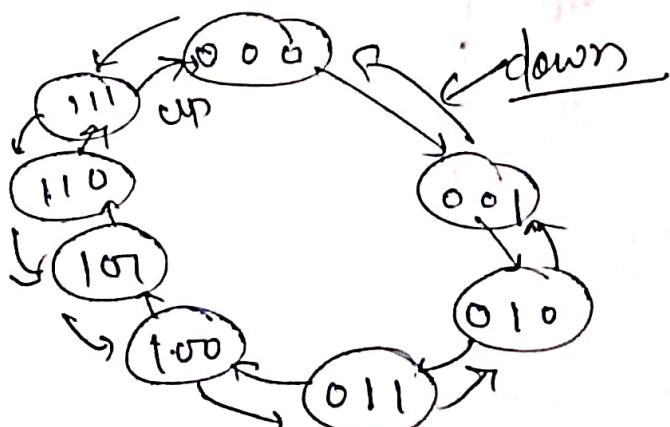


3 BIT counter :- (up)

$$n = 3, \text{ no of states} = 8 = 2^3$$

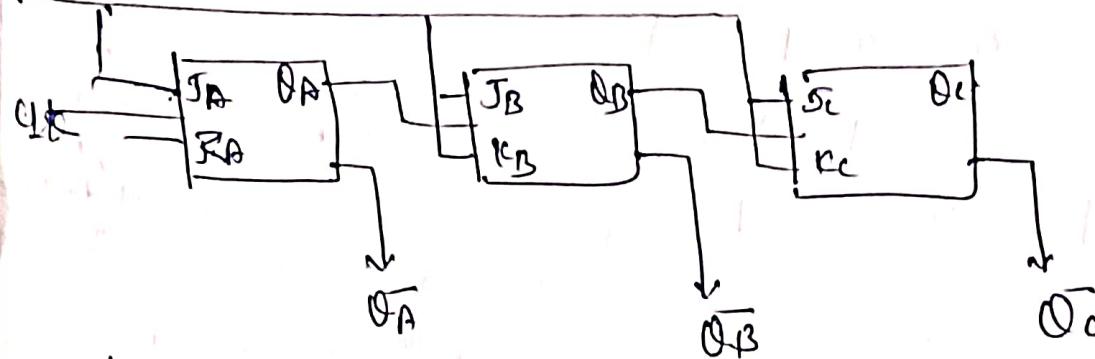
$$MC = 2^n - 1 = 8 - 1 = 7$$

$$= 7(011\Phi)$$

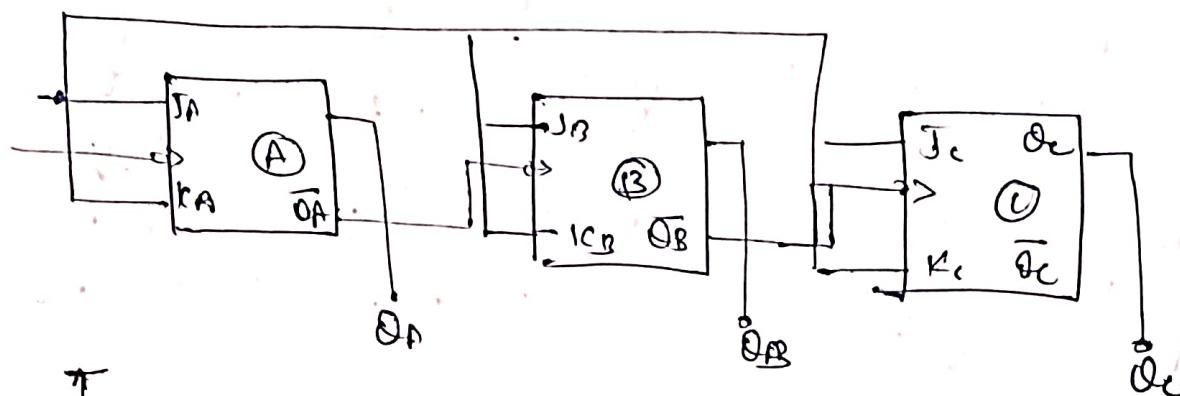


* 3 Bit & 4 bit Asynchronous Down Counter

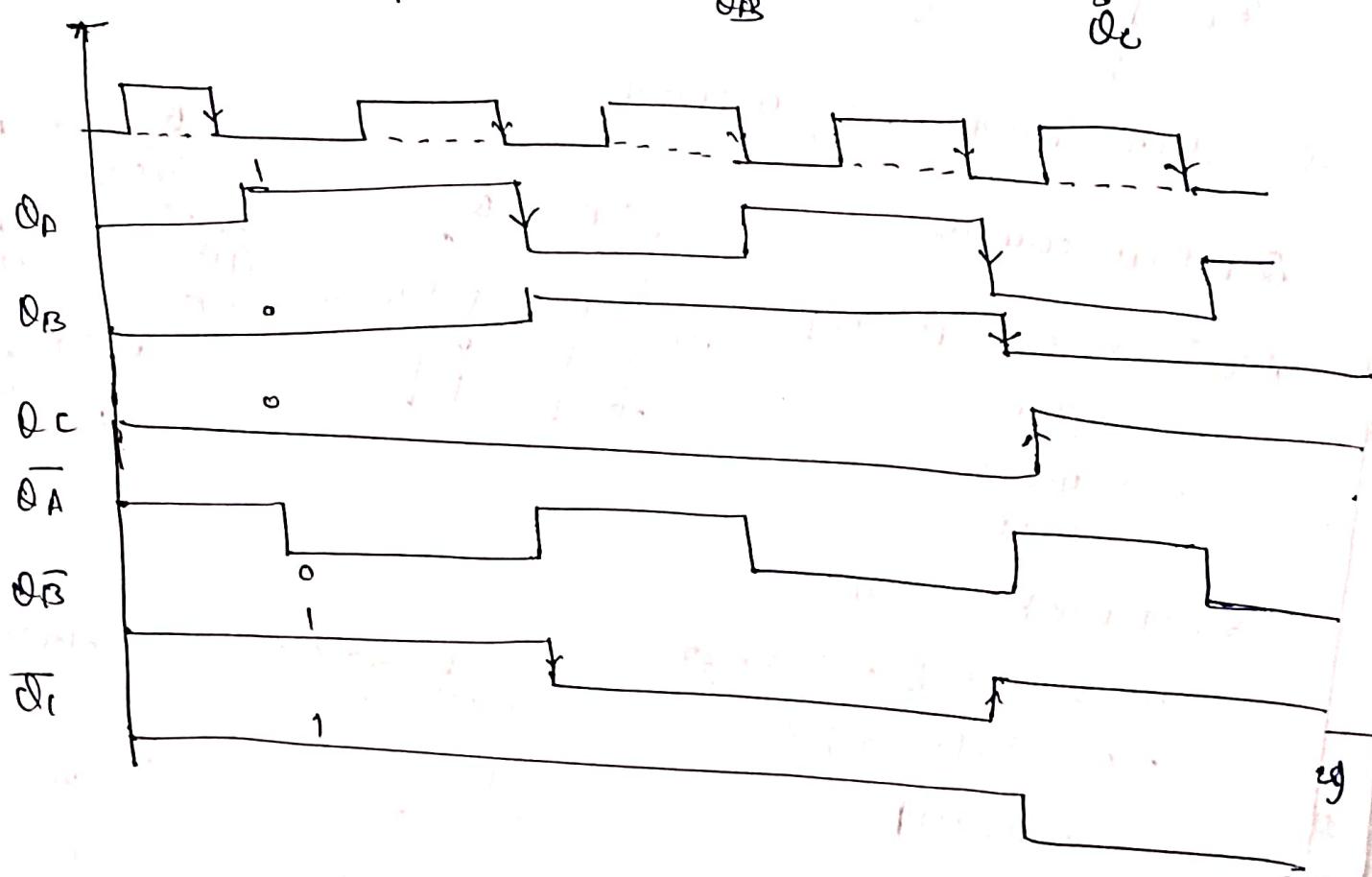
logic-1



logic-2



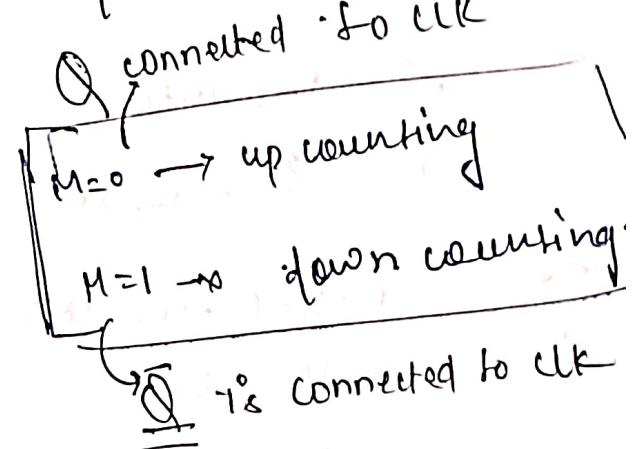
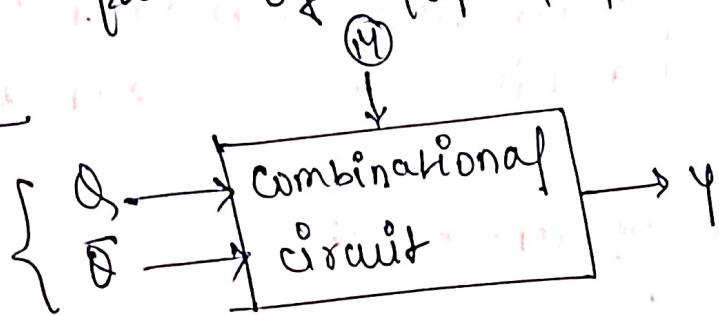
Q1



* 3 Bit and 4-Bit UP/Down Ripple counters

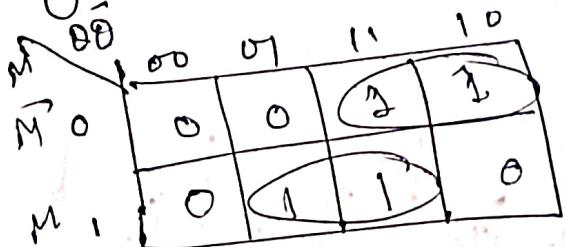
- we have designed the up counters and down counters separately
- But in practice both these modes are combined
- A mode control input (M) is used to select either up or down mode
- Combinational circuit is required between each pair of flip flops

Logic - 1



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

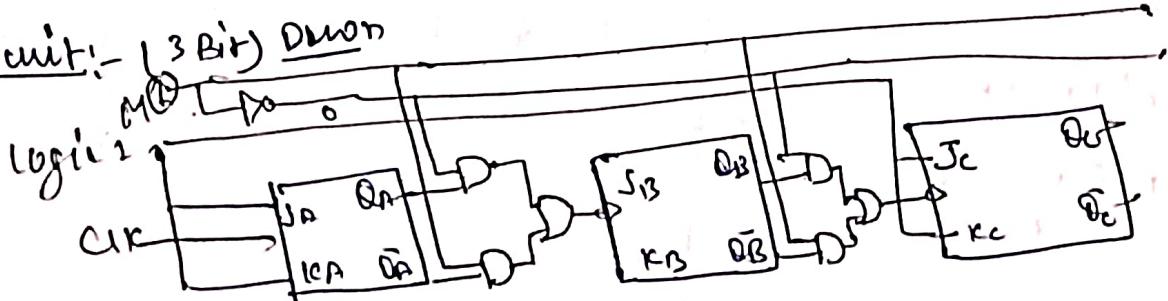
using k-map



$$Y = I + II$$

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

Circuit:- (3 Bit) Down



- * Modulus of the counter & counting up to particular value
- 2-bit ripple counter is called MOD 4 or modulus 4 counter
- 3-bit ripple " " as MOD-8 counter
- $n \rightarrow$ no of bit

$$\text{MOD number} = 2^n$$

$$2^2 = 4 \text{ counter}$$

$$2^3 = 8 \text{ counter}$$

$$\text{state} = 2^n$$

2-bit up or down \rightarrow MOD-4

3-bit up or down \rightarrow MOD 8

4-bit " MOD 16

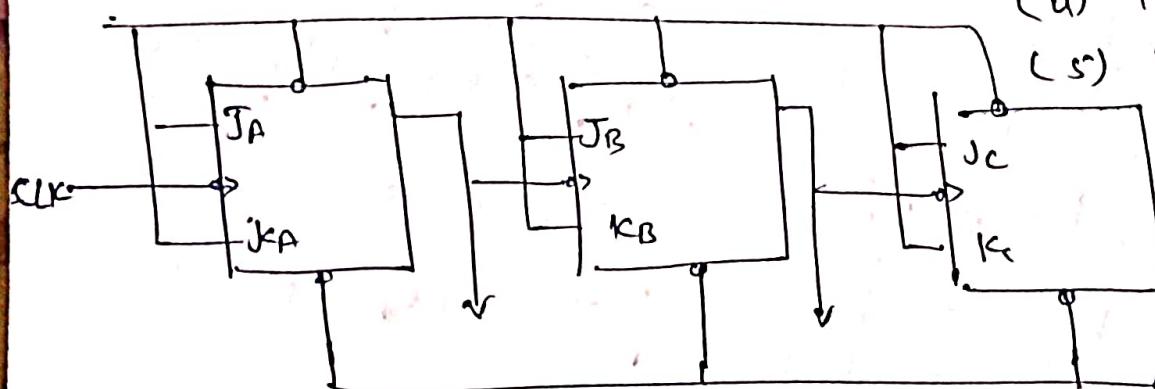
Eg

MOD-6 counter using MOD8 counter

$$\text{state} = 6$$

$$\text{MC} = 6-1 = 5$$

Ckt



| |
|----------------------------------|
| $PST = 0 \Rightarrow \theta = 1$ |
| $CUR = 0 \Rightarrow \theta = 0$ |

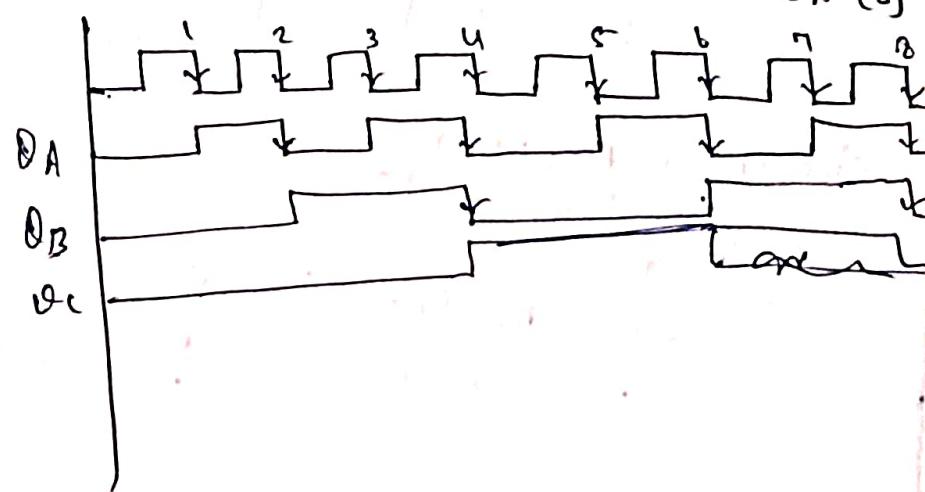
wave-diagram

$$Q_A \rightarrow Q_B (\text{DF} \rightarrow Q_A)$$

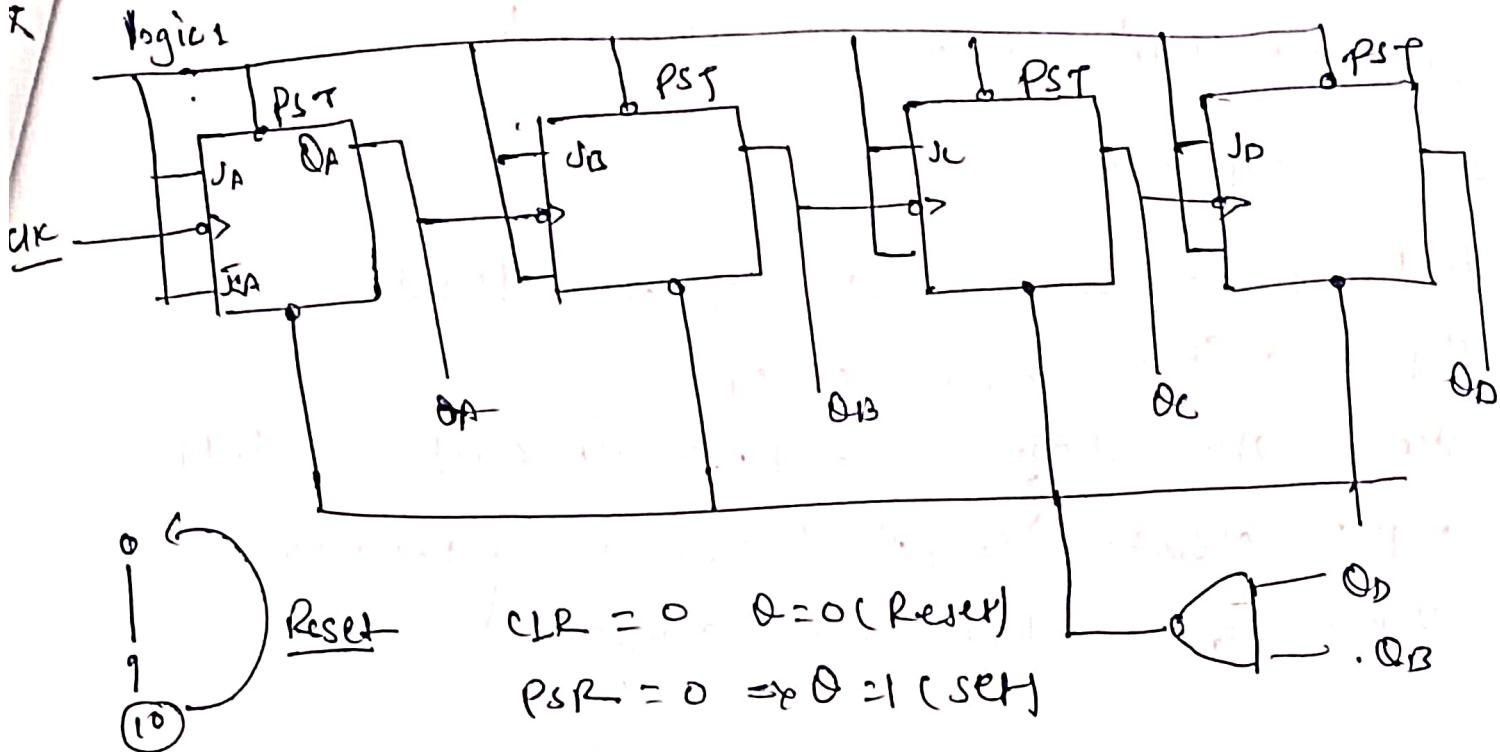
$$Q_B \rightarrow Q_C (\text{Downfall } Q_B).$$

| | |
|-----------|---|
| (0) 0 0 0 | ← |
| (1) 0 0 1 | |
| (2) 0 1 0 | |
| (3) 0 1 1 | |
| (4) 1 0 0 | |
| (5) 1 0 1 | |

MOD-6



* Decade (BCD) 12 Ripple Counter

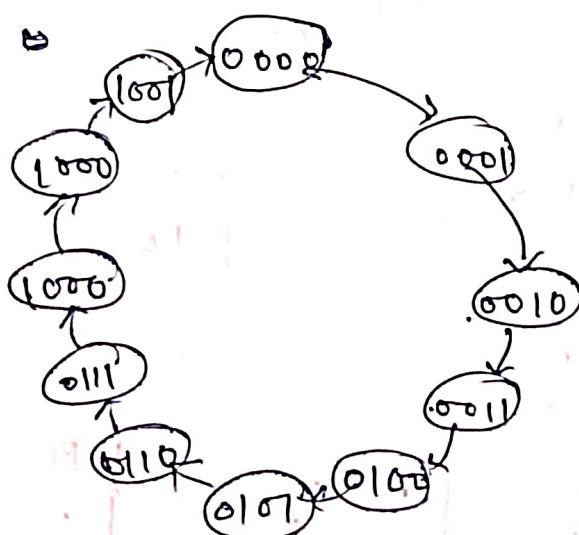


- Imp:-
- i) negative edge triggered $\rightarrow Q$ is clock \rightarrow UP counter
 - ii) positive " " " $\rightarrow \bar{Q}$ is clock \rightarrow UP counter
 - iii) negative " " " $\rightarrow Q$ is clock \rightarrow Down counter
 - iv) positive " " " $\rightarrow \bar{Q}$ is clock \rightarrow Down counter



No of states = 10

Max. count = 10129



| clock | QD | Q ₂ | Q ₁ | Q ₀ |
|------------------|----|----------------|----------------|----------------|
| Initially | 0 | 0 | 0 | 0 |
| 1 st | 0 | 0 | 0 | 1 |
| 2 nd | 0 | 0 | 1 | 0 |
| 3 rd | 1 | 0 | 0 | 0 |
| 4 th | 1 | 0 | 0 | 1 |
| 5 th | 1 | 0 | 1 | 0 |
| 6 th | 0 | 1 | 0 | 0 |
| 7 th | 0 | 1 | 0 | 1 |
| 8 th | 0 | 1 | 1 | 0 |
| 9 th | 1 | 0 | 1 | 0 |
| 10 th | 1 | 0 | 1 | 0 |

Decade counter

How to Design synchronous Counting

Q. Design 2-bit synchronous up counter

→ JK Flip Flop

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 eqn. using k-map

Step-5:- Draw the logic diagram

① JK Flip Flop

② Excitation table of JK FF

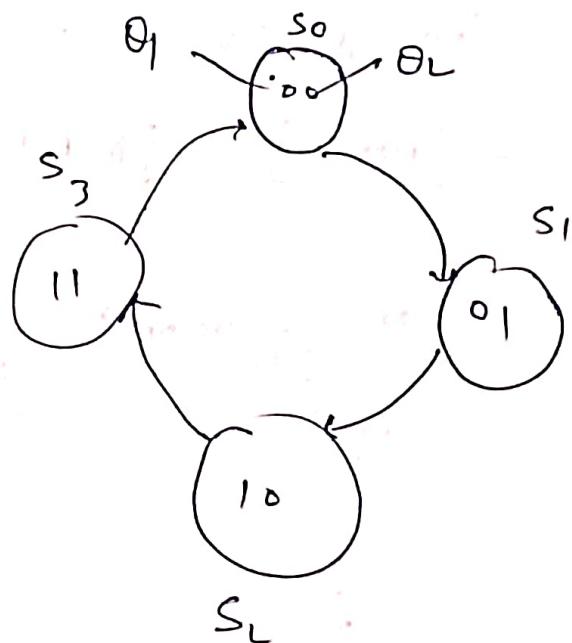
| Qn | Out. | J ₁ | K |
|----|------|----------------|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

Step ③

State diagram

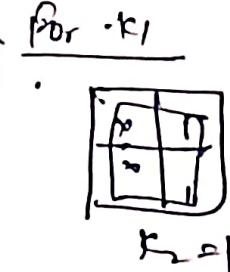
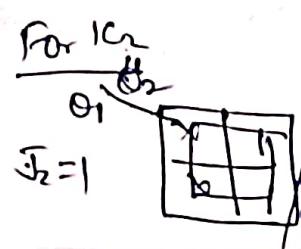
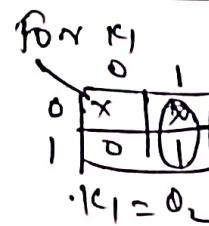
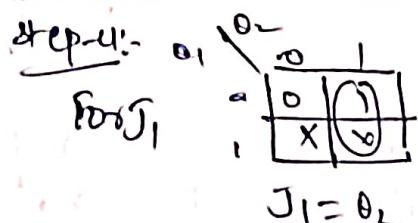
$$\text{No. of state} = 2^{n=2} = 4$$

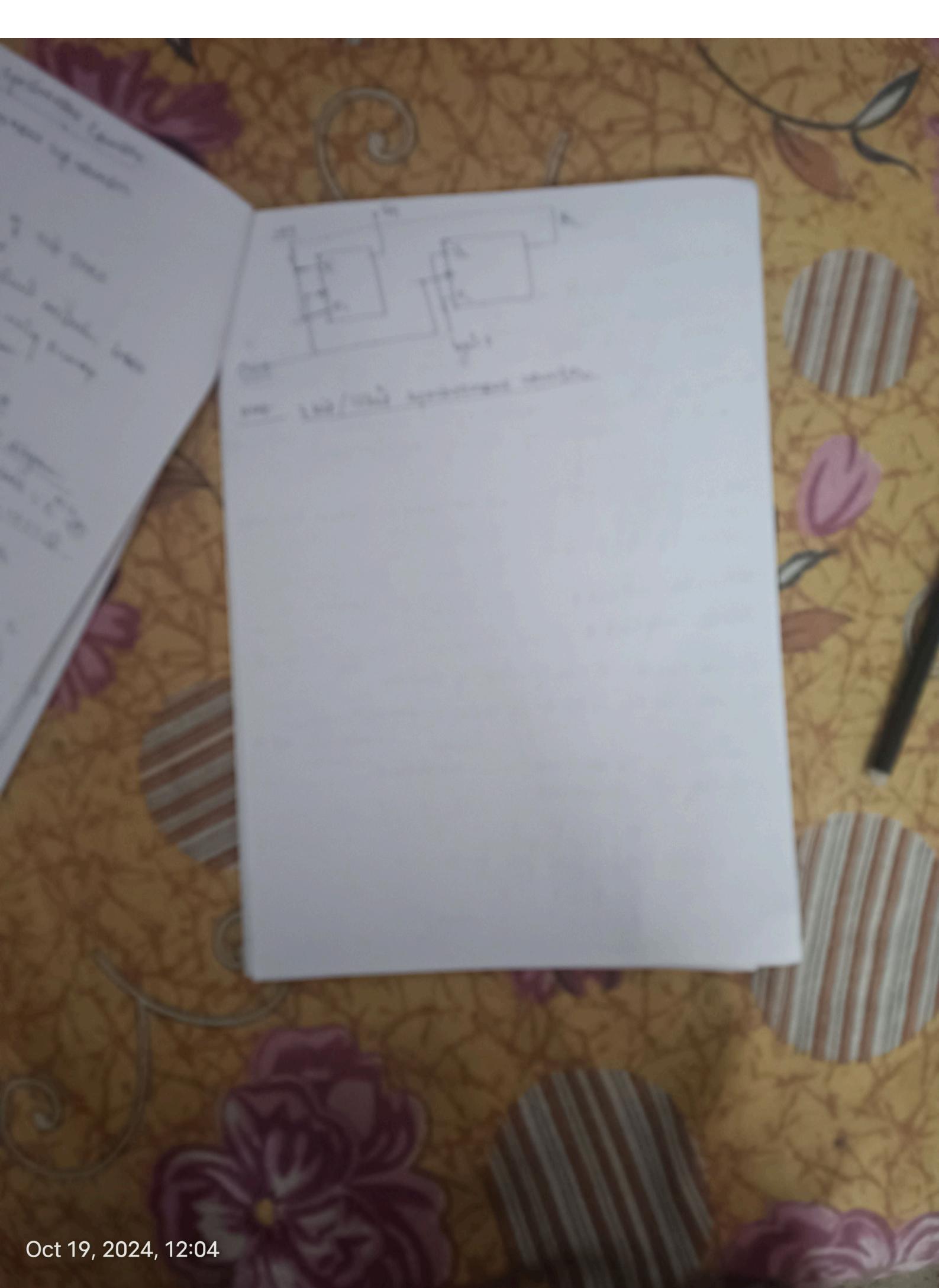
$$m_C = 4 - 1 = 3$$



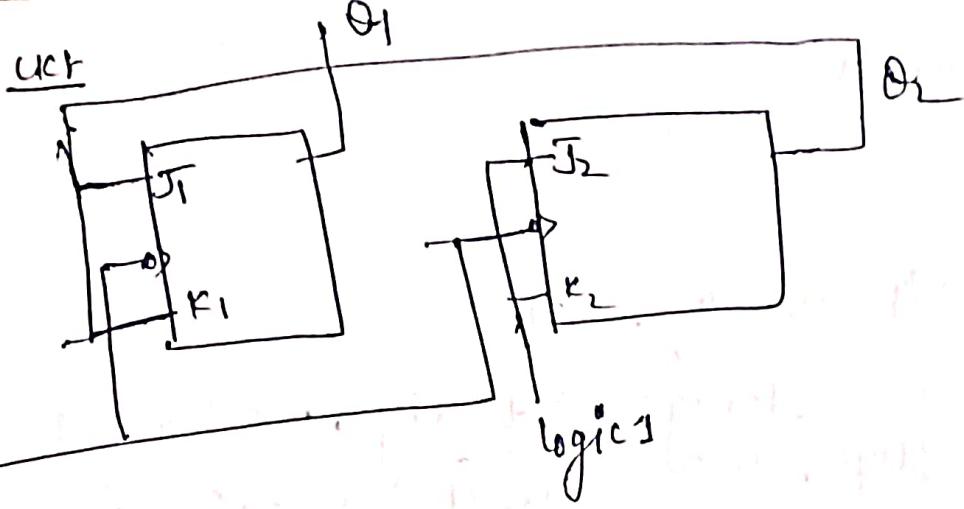
⇒ Circuit excitation table

| Q ₁₁ | Q ₁₂ | Q ₁₁ Q ₁₂ | S ₁ K ₁ | J ₂ K ₂ |
|-----------------|-----------------|---------------------------------|-------------------------------|-------------------------------|
| 0 | 0 | 0 0 | 0 X | 1 X |
| 0 | 1 | 0 1 | 1 X | X 1 |
| 1 | 0 | 1 0 | X 0 | 1 X |
| 1 | 1 | 1 1 | X 1 | X 1 |





Oct 19, 2024, 12:04



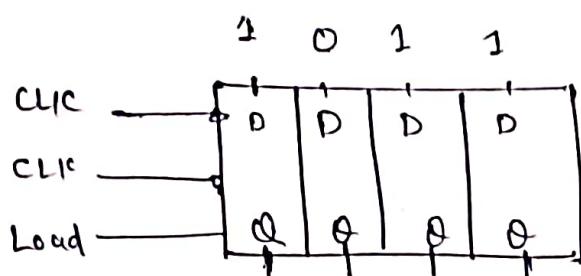
next 3 bit / 4 bit synchronous counter

UNIT - IV

Register

Introduction:-

- » 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 flop and is capable of storing "n-bit" word



Synch:- clock ↑ and load ↑

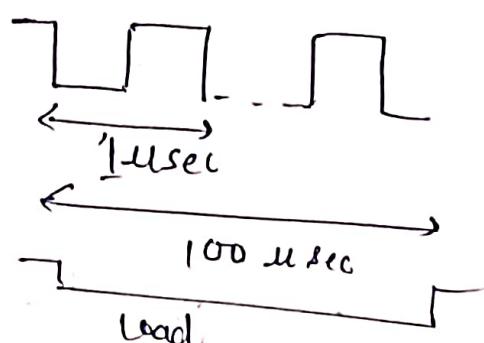
Asynchi:- only load ↑

we are bound to follow the clock

$$f = 1 \text{ MHz}$$

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

$$\boxed{T = 1 \mu\text{sec}}$$

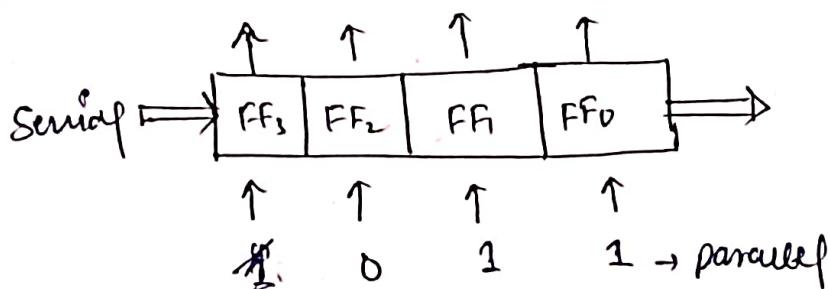


Data Formats & Classification of Registers

- » Data can be entered in serial or parallel form

↳ one bit at time ↳ all bits at a time
 serial form: Temporal code parallel form: spatial code

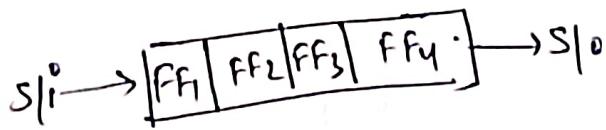
serial form: Temporal code
 parallel form: spatial code



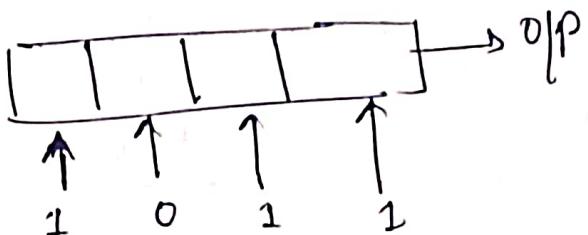
* Classification of Register

i) depending on I/P & O/P

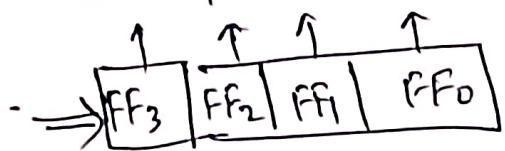
a) SISO



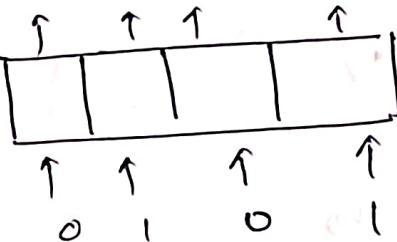
b) PI SO



c) SIPD



d) PIPO

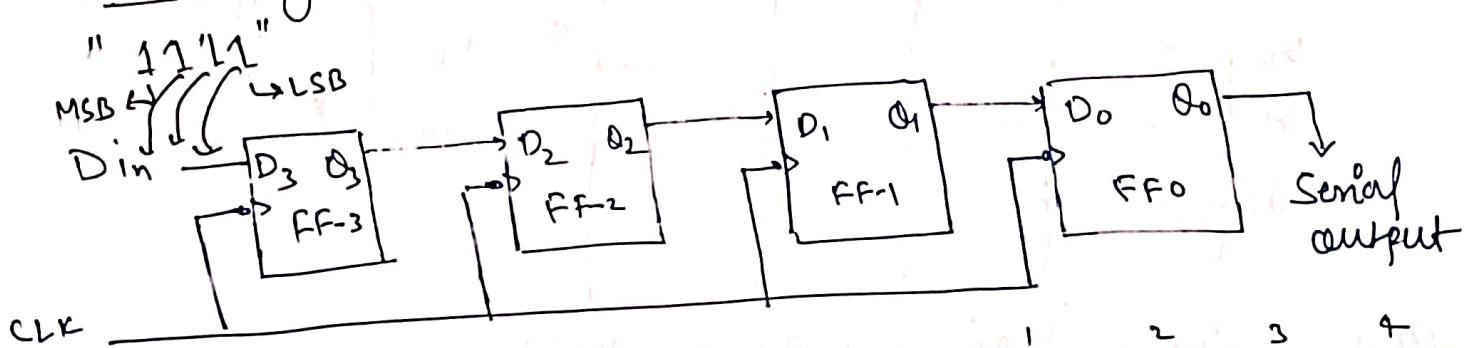


ii) Depending on application

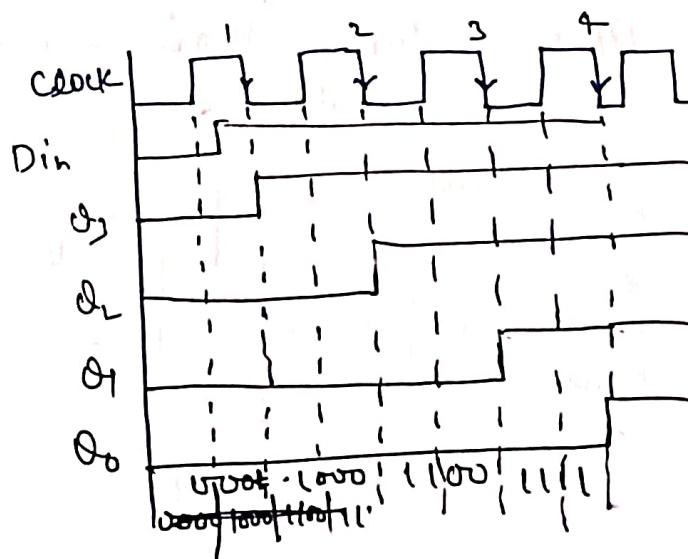
a) shift Reg.

b) storage Reg (PIPO)

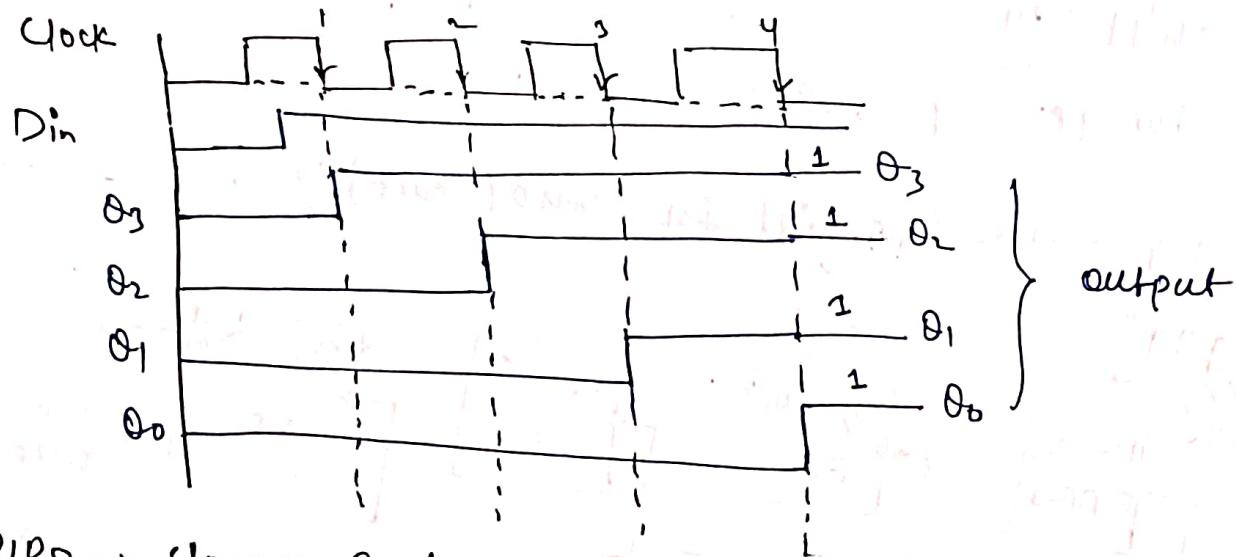
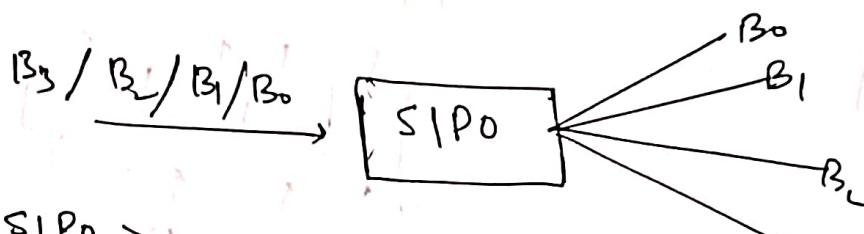
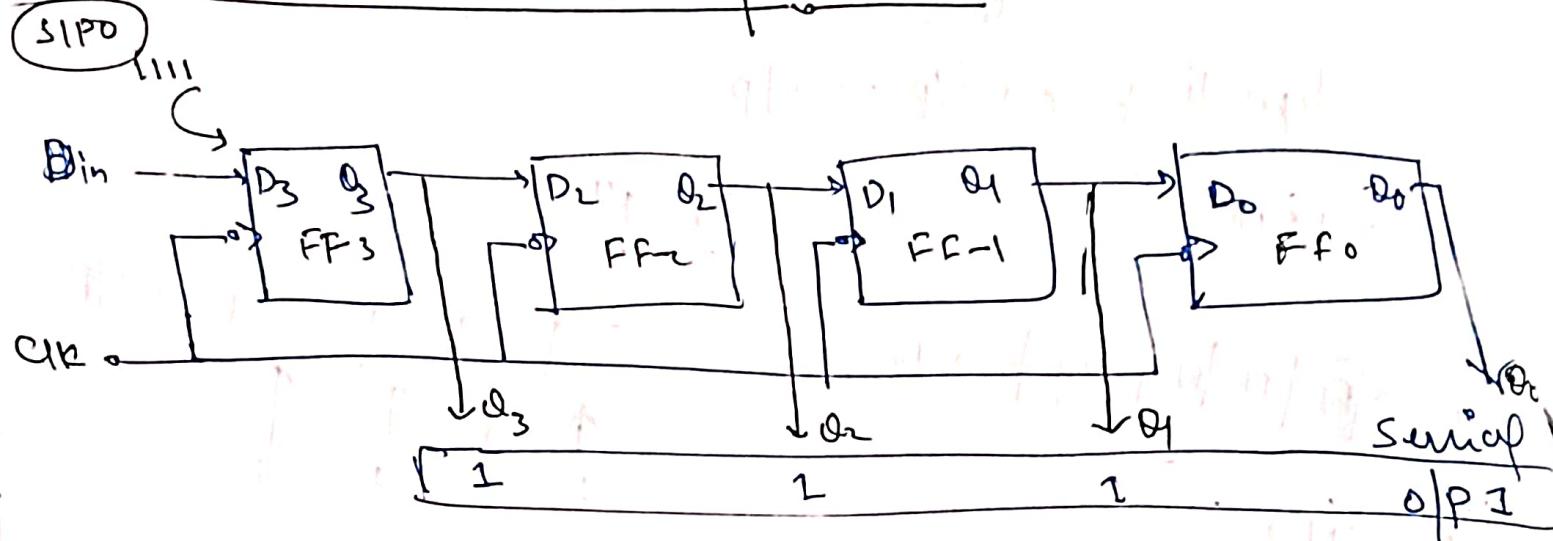
Shift Register (serial In serial Out)



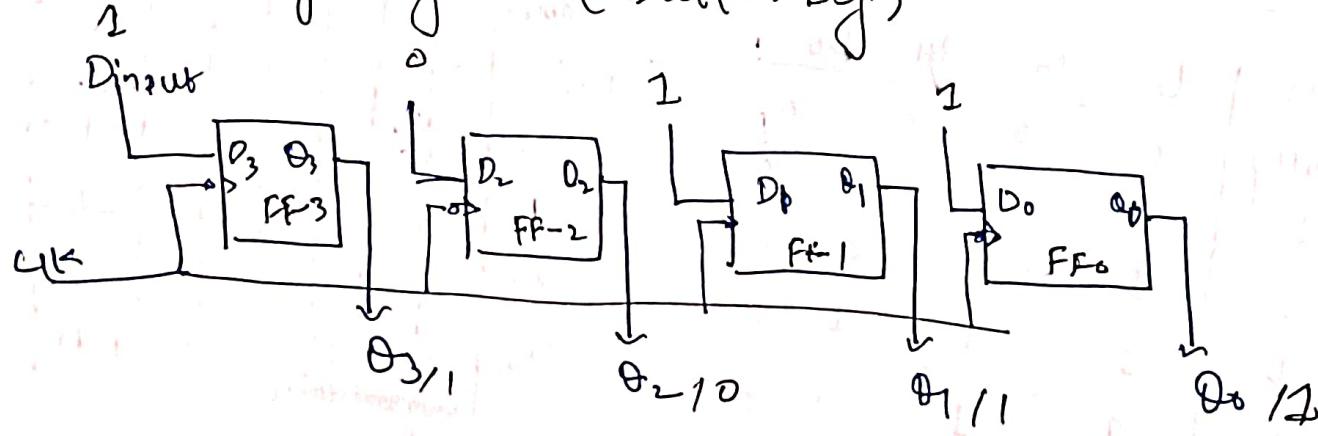
| CLK | Q_3 | Q_2 | Q_1 | Q_0 |
|-----------|-------|-------|-------|-------|
| Initially | 0 | 0 | 0 | 0 |
| ↓ | 1 | 0 | 0 | 0 |
| ↓ | 1 | 1 | 0 | 0 |
| ↓ | 1 | 1 | 1 | 0 |
| ↓ | 1 | 1 | 1 | 0 |



S1PO & P1PO mode shift Register



P1PO → storage Register (Buffer Reg.)



★ PISO \rightarrow 1 clock pulse to store the data.

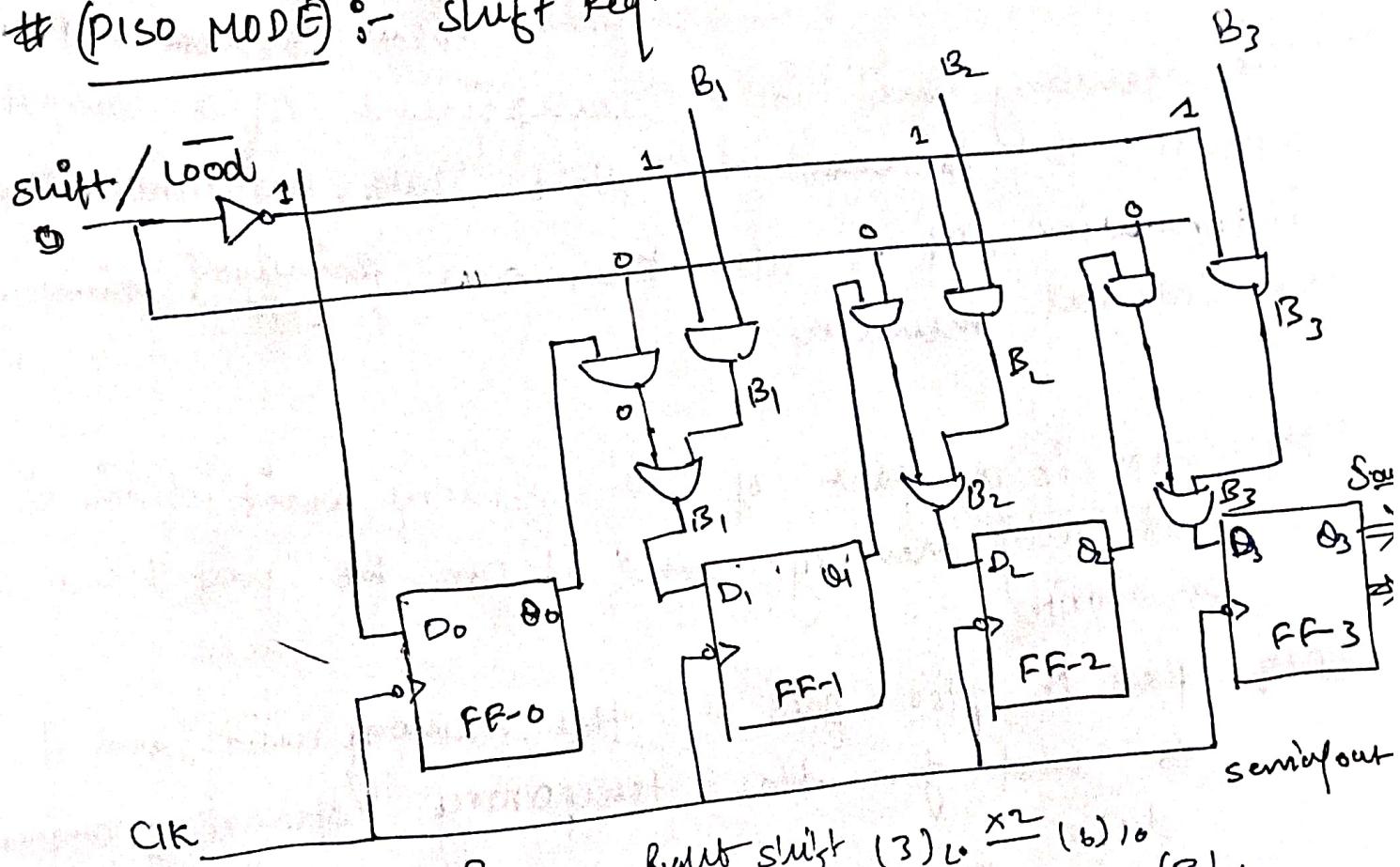
$$D=0 \Rightarrow Q_{n+1} = 0$$

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

$$D=2 \Rightarrow Q_{n+1} = Q_n$$

$$D=x$$

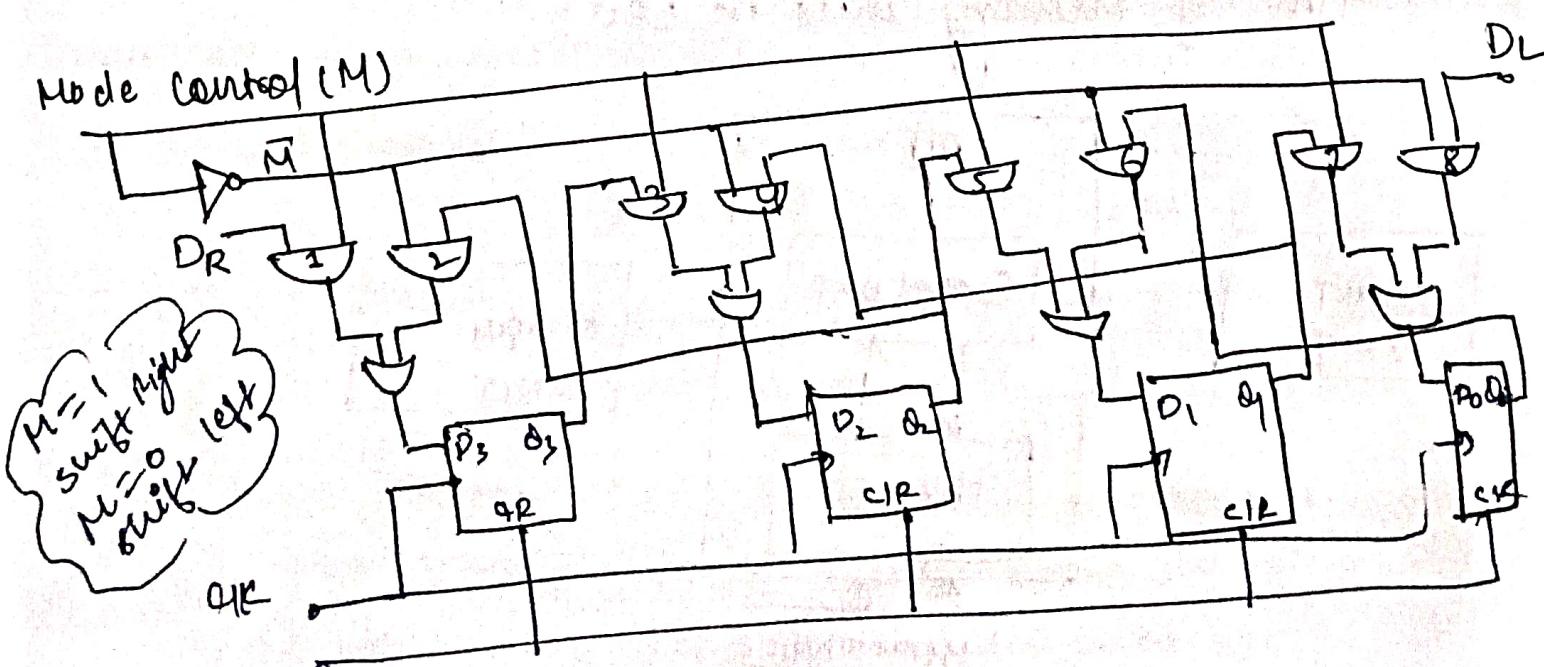
(PISO MODE) :- shift Reg'



* Bidirectional shift Reg.

$$\begin{aligned} \text{Right shift } (3)_0 &\xrightarrow{x_2} (6)_0 \\ \text{left shift } (6)_0 &\xrightarrow{x_2} (3)_0 \end{aligned}$$

Mode Control (M)



universal shift Register

Memory UNIT

ROM

(Read only Memory)

RAM

(Random access Memory)

PROM

EEPROM

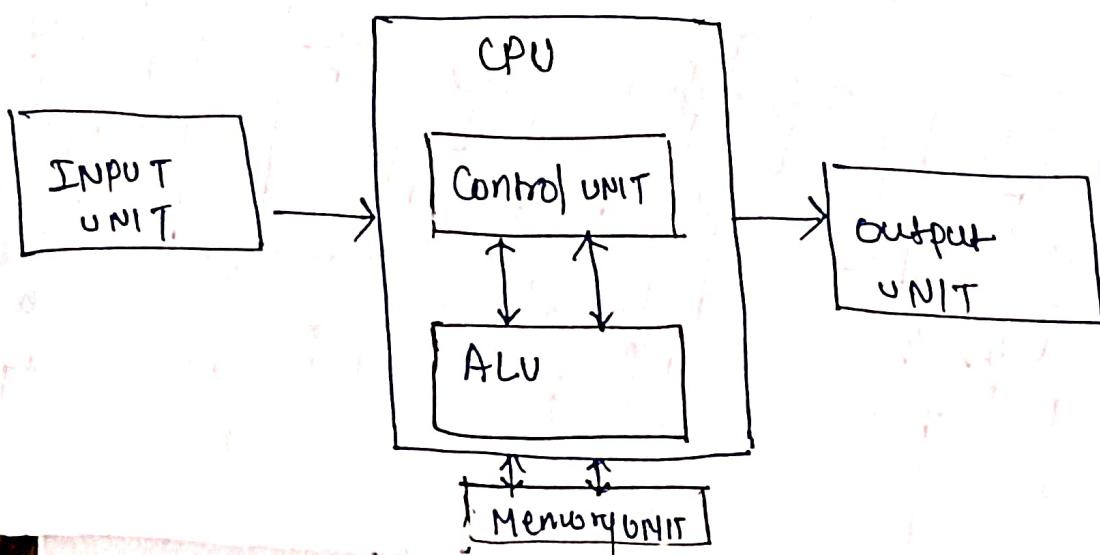
EPROM

⇒ The memory unit is a component of a computer system. It is used to store data, instructions, or information. It is also known as principal / primary memory.

ROM:- ROM is a part of the memory unit. This is read only memory, and it can be not be used to write.

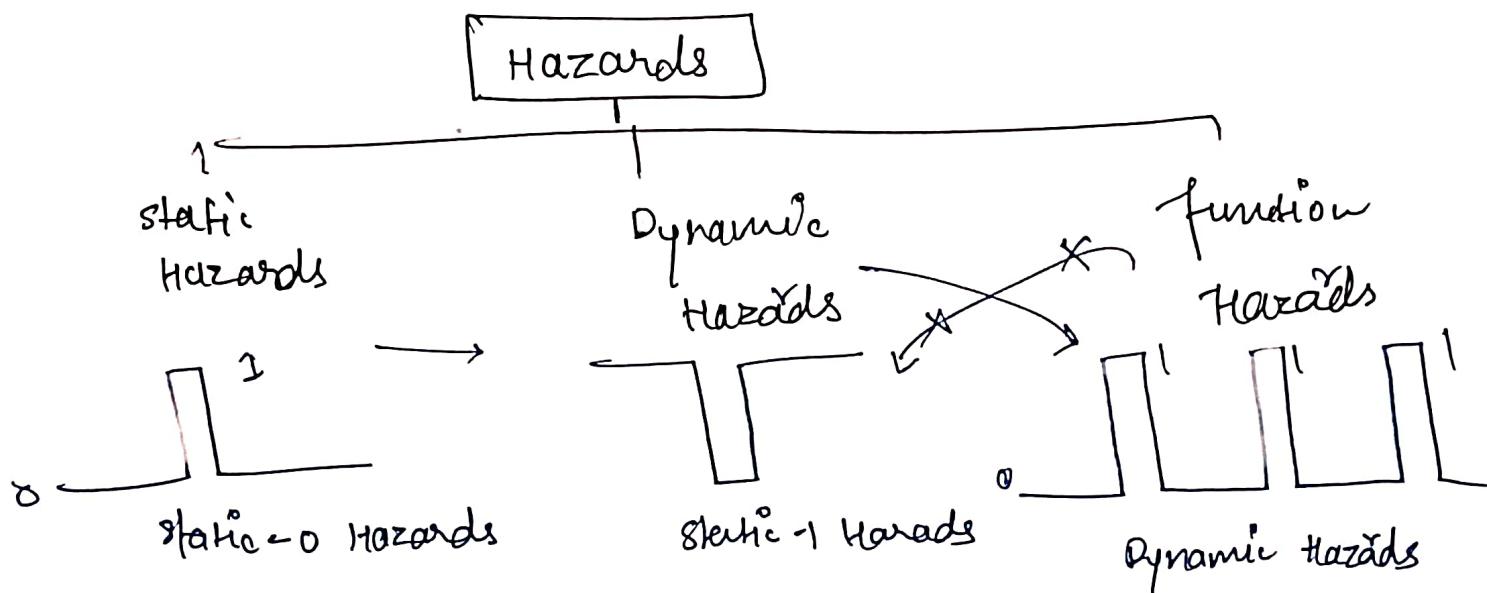
RAM:- RAM is also part of the memory unit, and it is used for the temporary storage of program data.

* Function of Memory unit in CPU:-



- * PROM:- Programmable Read only Memory can be programmed by a user only once
- * EEPROM:- Erasable Programmable Read only memory can be reprogrammed by exposing it to ultraviolet light for about 40 minutes.
- * EEPROM:- Electrically Erasable programmable Read only Memory can be programmed and erased electrically many times.

* Hazards



There are three types of Hazards

- Static-0 Hazards:- The o/p temporarily changes to 1 when it should have been stayed a 0
- Static-1 Hazards:- the o/p temporarily changed to 0 when it should have stayed a 1
- Dynamic Hazards:- The o/p changes multiple times when it should have made a single logic transition