Sub :

Day

Time : Date : / /

MOHSIN IBNA HOSSAIN
AIUB, DLAC NOTE'S.

#FINAL TERM:-

Sub :

Time : Date : / /

SEQUENTIAL CIRCUIT DESIGN

Sub :

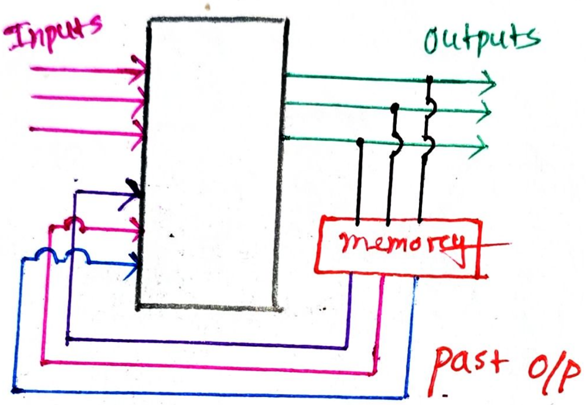
Day

Time : Date : / /

#LATCHES

Sequential Circuits:-

⇒ In Sequential circuits, the present output depends on the present input as well as past output/outputs



SR Latch:-

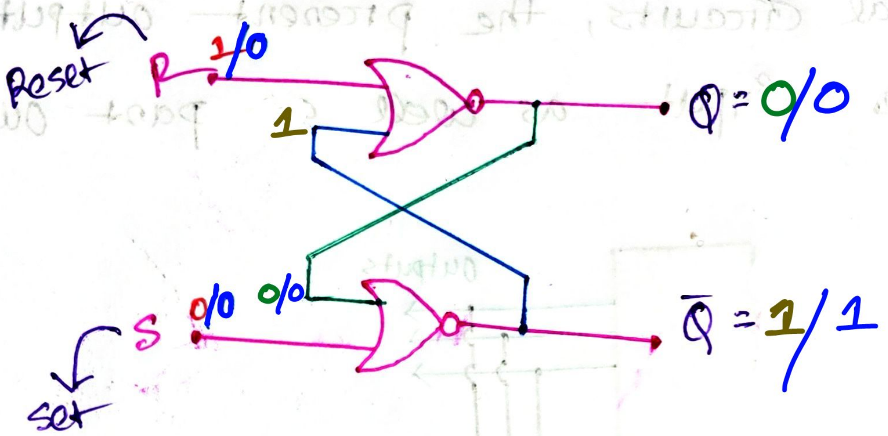
⇒ The basic storage element is called "LATCH".

There are two types:-

1. NOR SR Latch.
2. NAND SR Latch

Sub :   Day   Time :   Date : / /

SR NOR Latch:—



Note:—

Reset $Q = 0$

Set $Q = 1$

TRUTH TABLE: FOR NOR Gate:—

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

Analyse the SR Latch:—

Case 1:—

if $S = 0,\ R = 1$; then:— $Q = 0$ & $\bar{Q} = 1$

Ex:- if $\boxed{S = 0,\ R = 0 \text{ then:- } Q = 0 \ \& \ \bar{Q} = 1}$ → memory

Case 2:—

if $S = 1,\ R = 0$; then:— $Q = 1$ & $\bar{Q} = 0$

∴ Ex:- $\boxed{\text{if } S = 0,\ R = 0; \text{ then } Q = 1 \ \& \ \bar{Q} = 0}$ → memory

Sub : Day Time : Date : / /

Case: 3:—

if $S=1, R=1$ then:— $Q=0$ & $\bar{Q}=0$

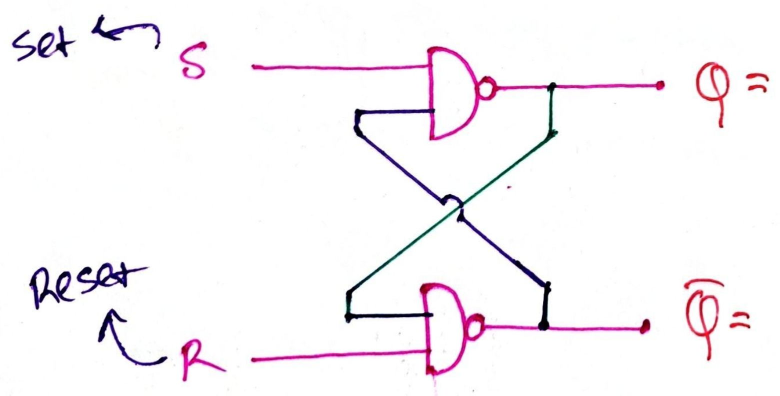
Ext if:- $S=0, R=0$; then $Q=0$ & $\bar{Q}=1$
$Q=1$ & $\bar{Q}=0$

"Not used" cz its Break the rules.

Truth Table for SR Latch:— (NOR)

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

SR NAND Latch:—



Truth table for NAND Gate:—

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

Sub :

Time : Date : / /

Analyse the Latch:-

Case 1:-

if $S = 0, R = 1$ then, $Q = 1$ & $\bar{Q} = 0$

Ex:- if $\boxed{S = 1, R = 1 \quad \text{then, } Q = 1 \text{ \& } \bar{Q} = 0}$ → memory

Case 2:-

if $S = 1, R = 0$ then:- $Q = 0$ & $\bar{Q} = 1$

Ex:- if $\boxed{S = 1, R = 1 \quad \text{then:- } Q = 0 \text{ \& } \bar{Q} = 1}$ → memory

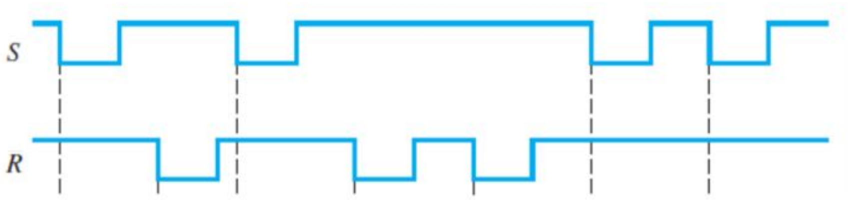
Case 3:-

if $S = 0, R = 0$, then:- $Q = 1$ & $\bar{Q} = 1$ → Not used

Truth table for SR latch (NAND):-

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

Exercise 1:

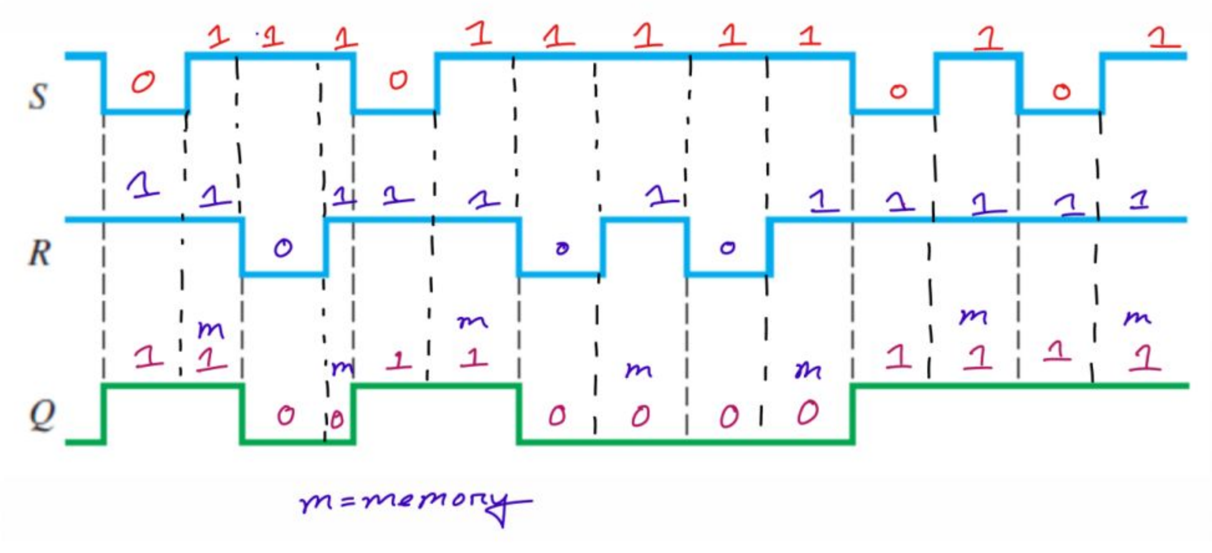


Determine the **Output(Q) Waveform** for **NAND** ***SR Latches*** that are initially **LOW**.

➔ **We Know,**

Truth table for SR Latch (NAND):

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



Day

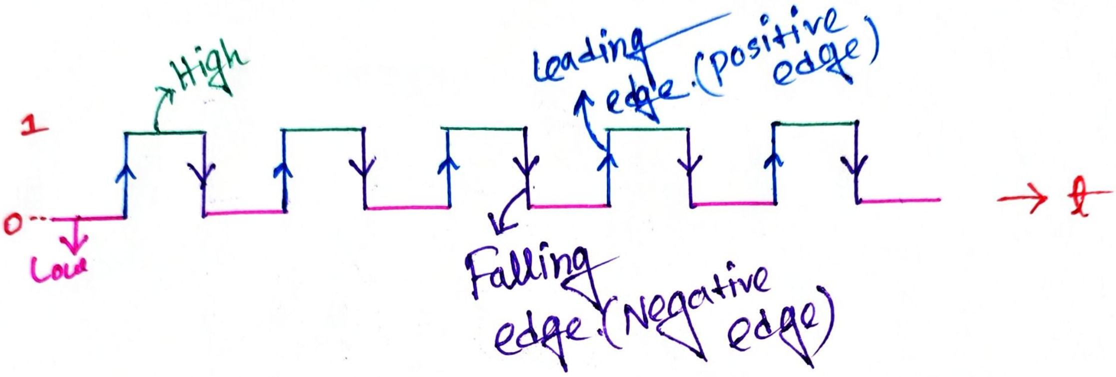
Sub :

Time : Date : / /

#Clock

Clock:-

⇒ It's a signal that goes from low to High then again low and repeats.



Triggering Methods:-

⇒ It's usually refer to how flip-flops and other Sequential elements change their states.

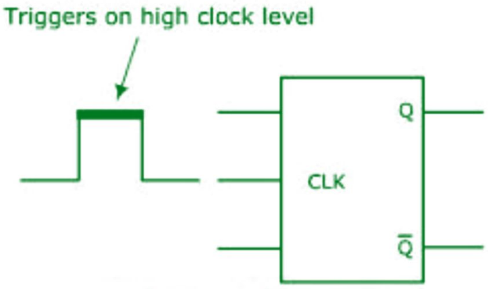
There are two types:-

1. Level - triggered.
2. Edge - triggered.

Level - triggered:-

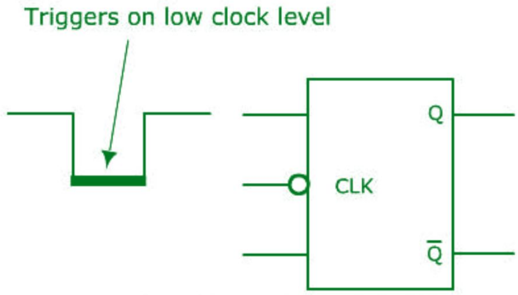
⇒ This flip-flops respond to the level of the clock signal, either high (1) or Low (0).

Block Diagram for high Level:-



High Level Triggering

Block Diagram for Low Level:-

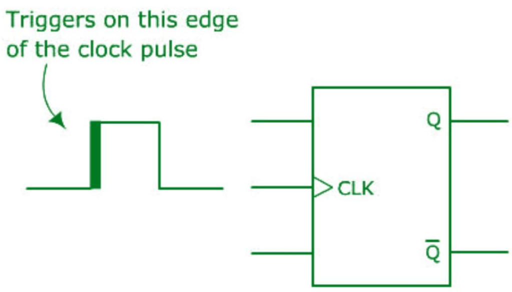


Low Level Triggering

Edge - Triggered:-

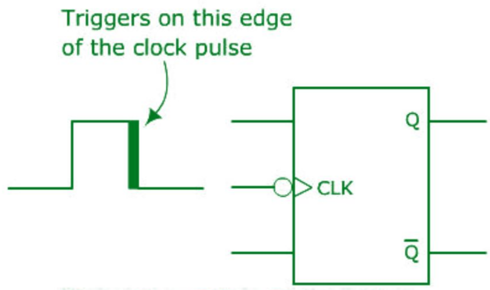
⇒ these flip-flops change state on a specific edge of the clock signal, either the rising edge (positive edge) or the falling edge (Negative edge)

Block Diagram For (+ve edge)



Positive Edge Triggering

Block Diagram for (-Ve edge)



Negative Edge Triggering

Sub :

Time : Date : / /

FLIP FLOP

Sub :

Time : Date : / /

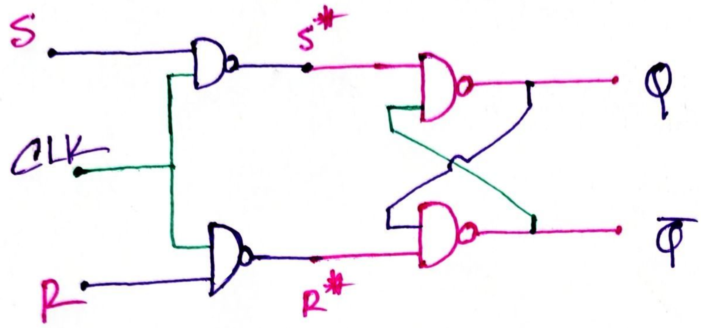
SR FLIP FLOP

Sub :

Day

Time : Date : / /

SR FLIP FLOP



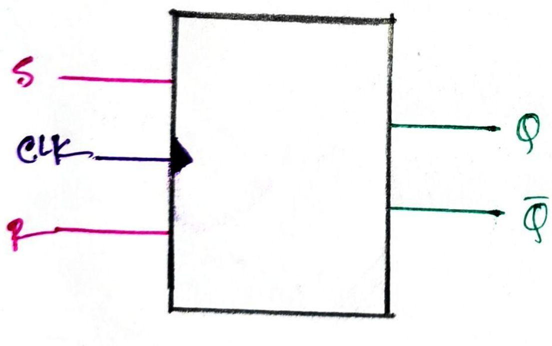
Truth table of SR Latch (NAND)

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

$\therefore S^* = (\overline{S \cdot CLK}) = \bar{S} + \overline{CLK}$

$\therefore R^* = (\overline{R \cdot CLK}) = \bar{R} + \overline{CLK}$

Block Diagram of SR Flip-Flop:-



Truth table for SR Flip-Flop:-

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	

Analyse the truth table:-

Case: 1:-

CLK = 0;

$S^* = \bar{S} + 1 = x + 1 = 1$

$R^* = \bar{R} + 1 = x + 1 = 1$

$\therefore Q \ \& \ \bar{Q}$ = memory

Case: 2:-

CLK = 1;

$S^* = \bar{S}$

$R^* = \bar{R}$

Let, $S = 0, R = 0$ — (1)

$\therefore S^* = 1$

$R^* = 1$

$\therefore Q \ \& \ \bar{Q}$ = memory

Case: 3:-

From eq (1):-

Let, $S = 0, R = 1$

$\therefore S^* = 1$

$R^* = 0$

$\therefore Q = 0 \ \& \ \bar{Q} = 1$

Case 4:-

From eq (1):-

Let, $S = 1, R = 0$

$\therefore S^* = 0$

$R^* = 1$

$\therefore Q = 1 \ \& \ \bar{Q} = 0$

Characteristic Truth Table for SR Flip-Flop:-

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Invalid(x)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Invalid (x)

Truth table for SR-Flip-flop

Clk	S	R	Q_{n+1} → next state
0	X	X	Q_n → p. state
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

Excitation table:-

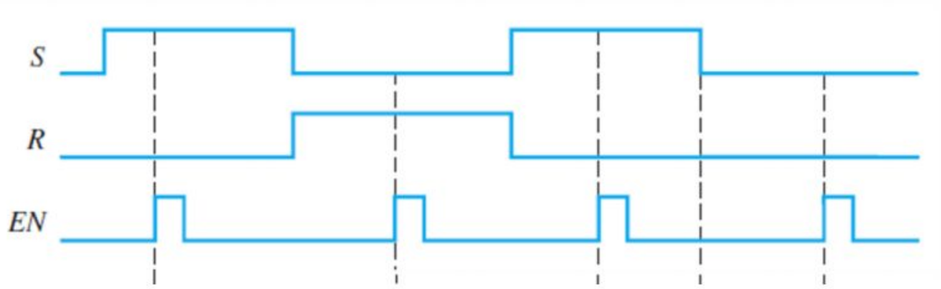
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

K-map for Q_{n+1}:-

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

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

Exercise 2:

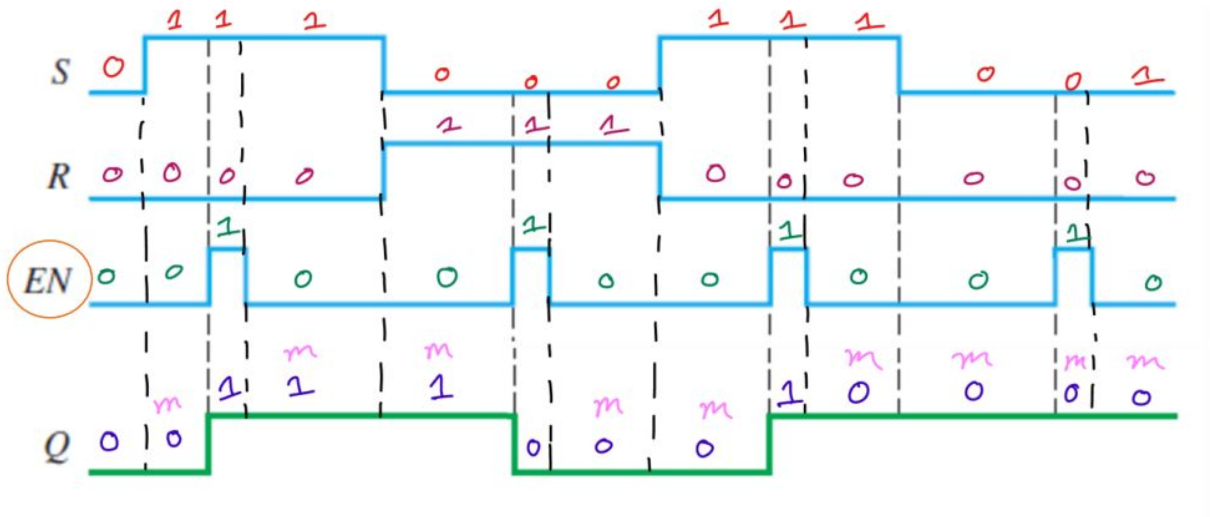


Determine the **Output(Q) Waveform** for ***SR FLIP FLOPS*** that are initially **Reset**

➔ **We Know,**

Truth table for SR-Flip-flop

Clk	S	R	Q_{n+1} → next state
0	X	X	Q_n → p. state
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid



m = memory

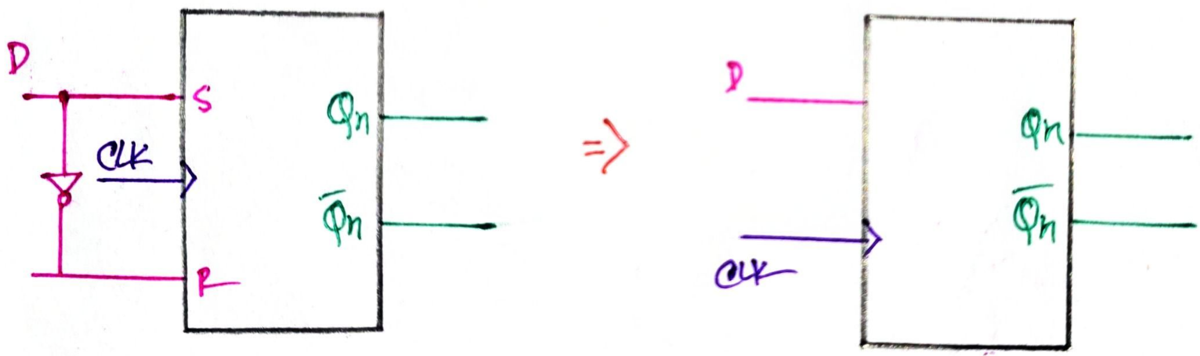
D-Flip Flop

D FLIP FLOP:-

⇒ In "NAND" SR Latch the invalid input condition of $S=0, R=0$. It in the drawback of the SR Latch.

To avoid the drawback we need an inveter to Conneet ~~the~~ between the set and Reset inputs. It ensure that at the same time S and R never equal to 1.

Logic Circuit with Block Diagram



T.T for D.F.F:-

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

Truth table for SR FLIP-FLOP:-

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	

Sub :   Time :   Date : / /

Characteristic Table for D-flip-flop:-

Q_n	D	Q_{n+1}
$\boxed{0}$	0	$\boxed{0}$
$\boxed{0}$	1	$\boxed{1}$
$\boxed{1}$	0	$\boxed{0}$
$\boxed{1}$	1	$\boxed{1}$

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

we know:-

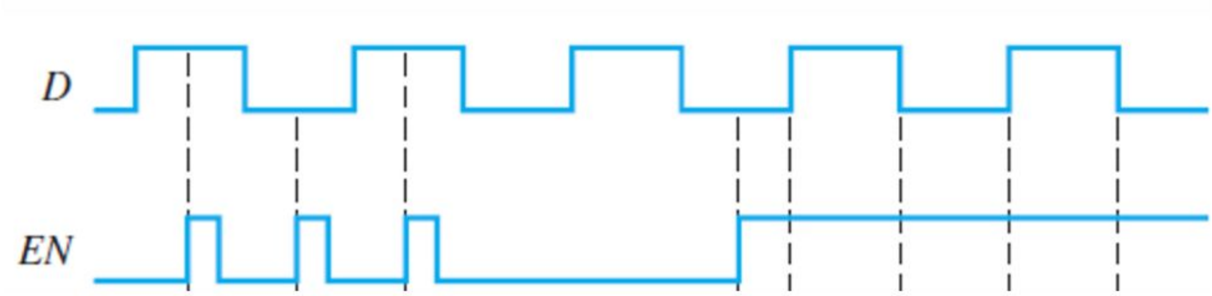
T.T for D.f.f

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

Excitation Table for D-flip-flop:-

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

Exercise 3:

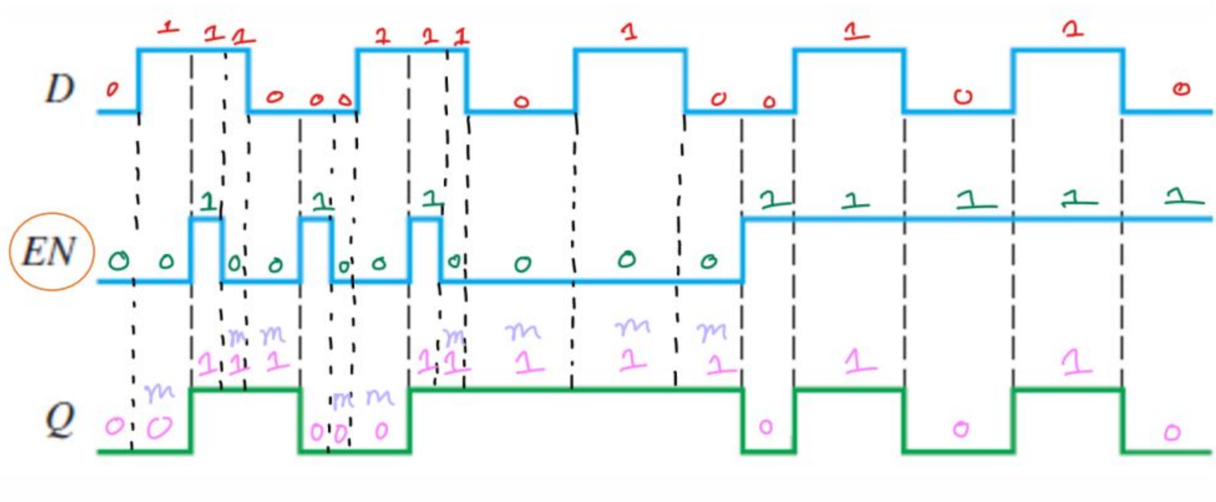


Determine the **Output(Q) Waveform** for ***D FLIP FLOPS*** that are initially **Reset**

➔ **We Know,**

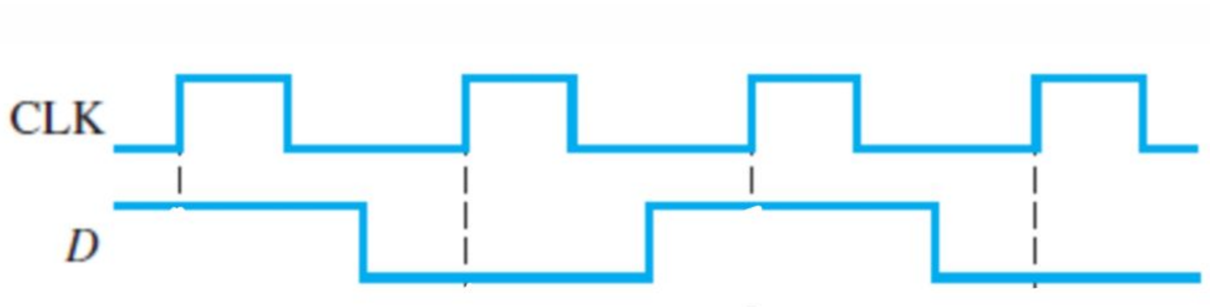
T.T for D.F.F.

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



m = memory

Exercise 4:

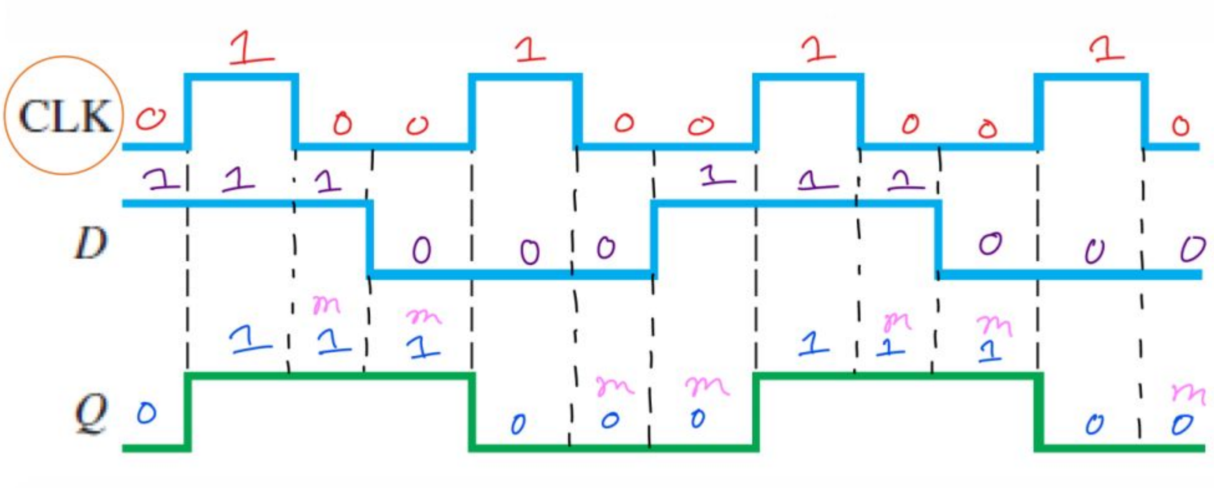


Determine the **Output(Q) Waveform** for ***D FLIP FLOPS*** that are initially **Reset**

➔ **We Know,**

T.T for D.F.F:-

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



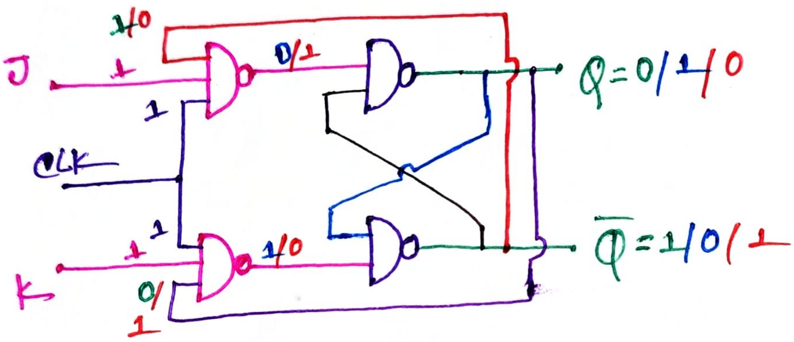
m = memory

JK FLIP FLOP

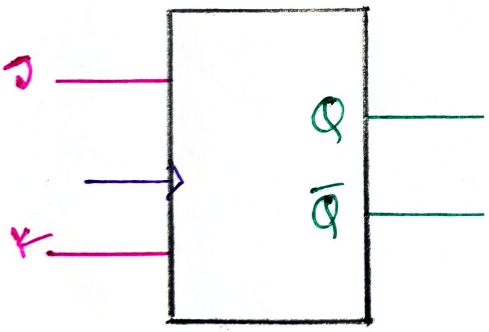
JK Flip-Flop:-

⇒ The JK flip flop is similar to the SR flipflop but there is no change in state when J=0 & K=0.

Logic Circuit



Block Diagram:-



T.T for J.K f.f

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	$\overline{Q_n}$ (toggle)

similar to SR f.f

Analyse the T.T.

Case 5:-

clk = 1

Let $Q = 0, \overline{Q} = 1$

Now $J = 1, K = 1$

$\therefore Q = 0, 1, 0, 1, \ldots$ &

$\overline{Q} = 1, 0, 1, 0, \ldots$

$\therefore Q_{n+1} = 0, 1, 0, 1, \ldots = \overline{Q_n}$

Sub : Time : Date : / /

Characteristic 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

We know, T.T. for JK ff

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

K-map for Q_{n+1}:-

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

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

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

K-map for J

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

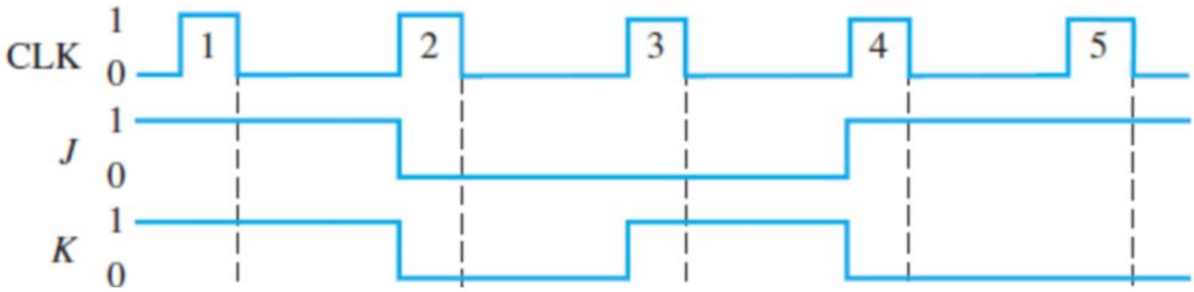
$J = Q_{n+1}$

K-map for K

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

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

Exercise 5:

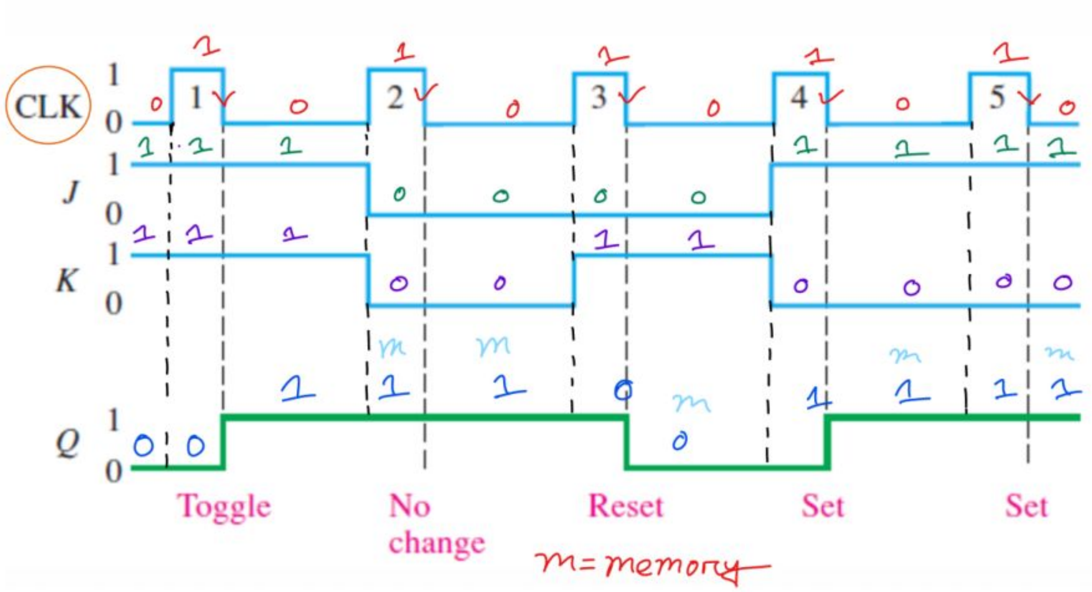


Determine the **Output(Q) Waveform** for ***JK FLIP FLOPS*** that are initially **Reset** and the flip flop is **negative edge triggered.**

➔ **We Know,**

T.T for J.K f.f

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	$\overline{Q_n}$ (toggle)



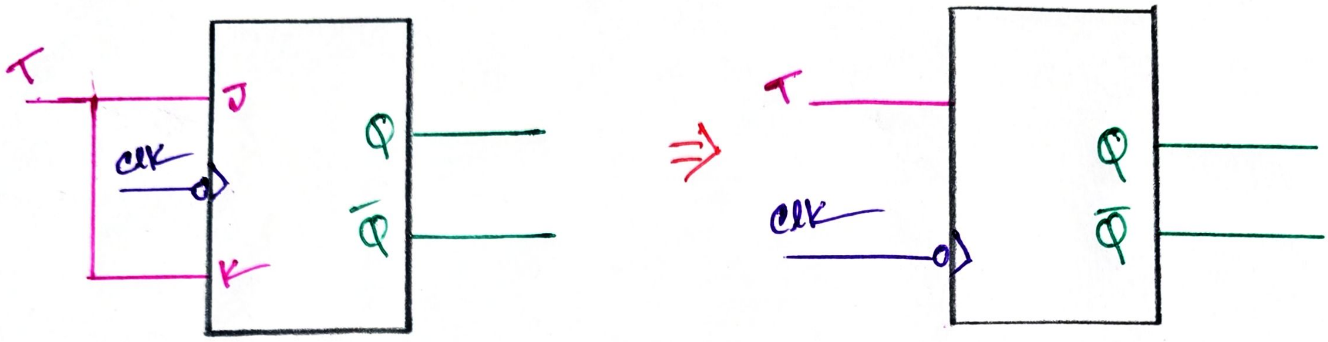
Sub :

Time : Date : / /

T- Flip Flop

T-flip-flop:-

Logic Circuit & Block Diagram:-



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)

Characteristic Table:-

Q_n	T	Q_{n+1}
$\boxed{0}$	0	$\boxed{0}$
$\boxed{0}$	1	$\boxed{1}$
$\boxed{1}$	0	$\boxed{1}$
$\boxed{1}$	1	$\boxed{0}$

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

Excitation Table:-

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

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

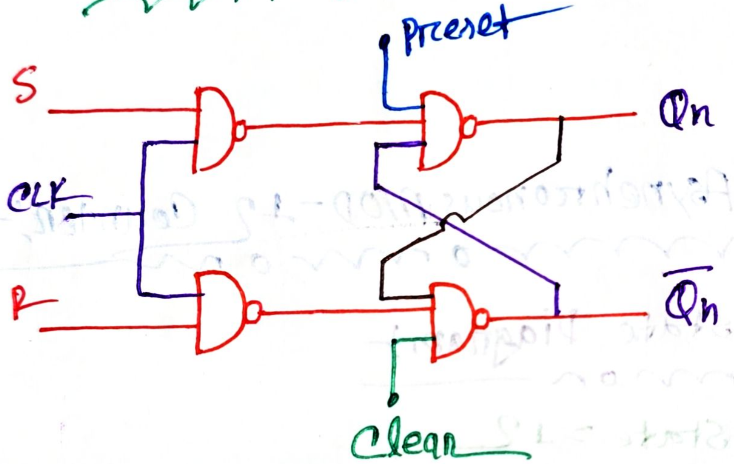
Sub: Time: Date: / /

Preset & Clear Inputs

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

2. The Synchronous inputs are S, R, J, K, D, & T

Logic Circuit:-



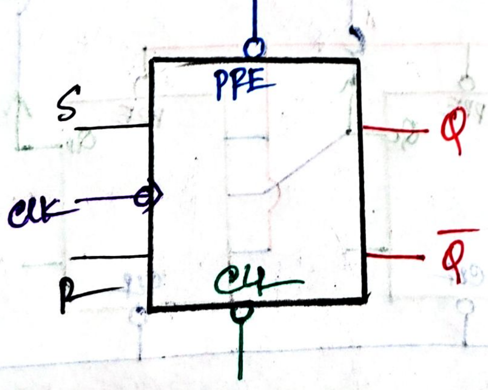
if, preset = 0, Then $Q_n = 1$ & $\overline{Q_n} = 0$

if, Clear = 0, Then $Q_n = 0$ & $\overline{Q_n} = 1$

What ever be the value of clock & synchronous inputs.

Block Diagram:-

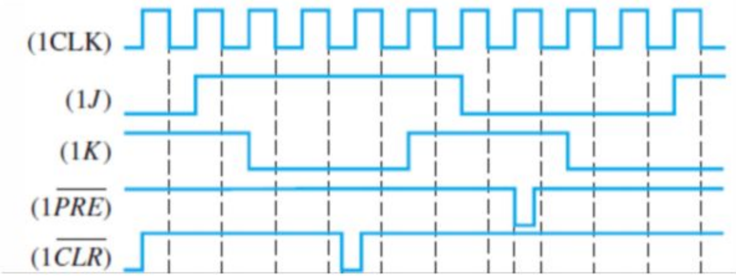
for Active Low:-



Truth Table

PR	CLR	Qn
0	0	Not used
0	1	1
1	0	0
1	1	F.F will Perform Normally.

Exercise 6:



Determine the **Output(Q) Waveform** for ***JK FLIP FLOPS*** that are initially **Reset** and the flip flop is **negative edge triggered.**

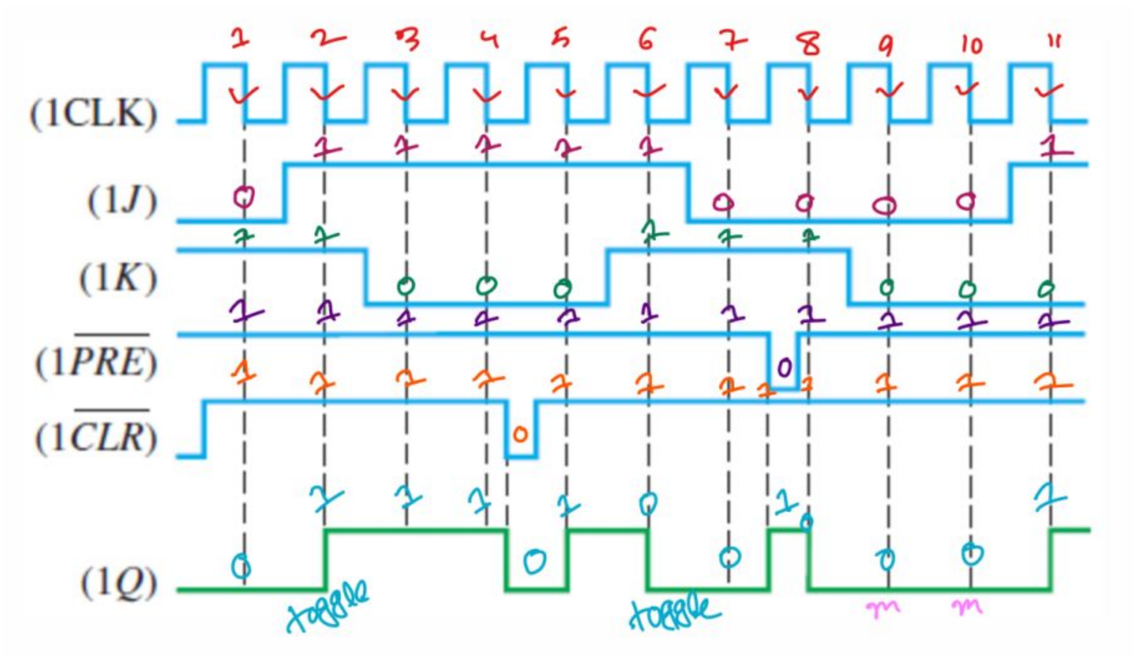
➔ **We Know,**

T.T for J.K F.F

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	$\overline{Q_n}$ (toggle)

Truth Table

PR	CLR	Q_n
0	0	Not Used
0	1	1
1	0	0
1	1	F.F will Perform Normally



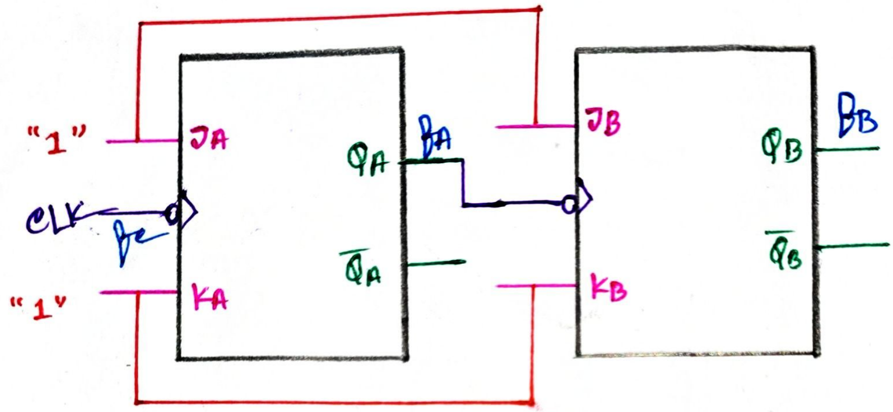
#Counters

Sub: Day Time: Date: / /

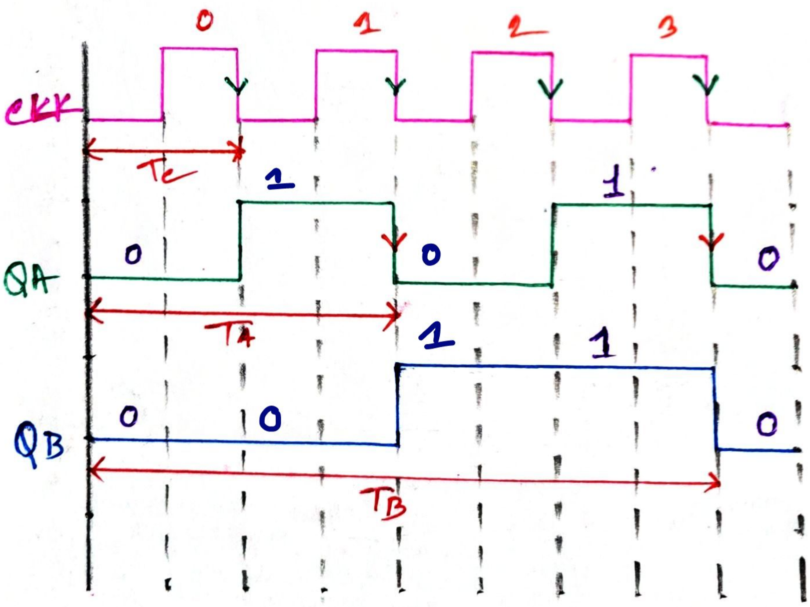
Counter:-

⇒ It's a Sequential circuit and it just simply counts.

Example:-



Analyse the Input & output clock frequency



Truth Table:-

CLK	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1

Sub: Time: Date: / /

Types of Counters:-

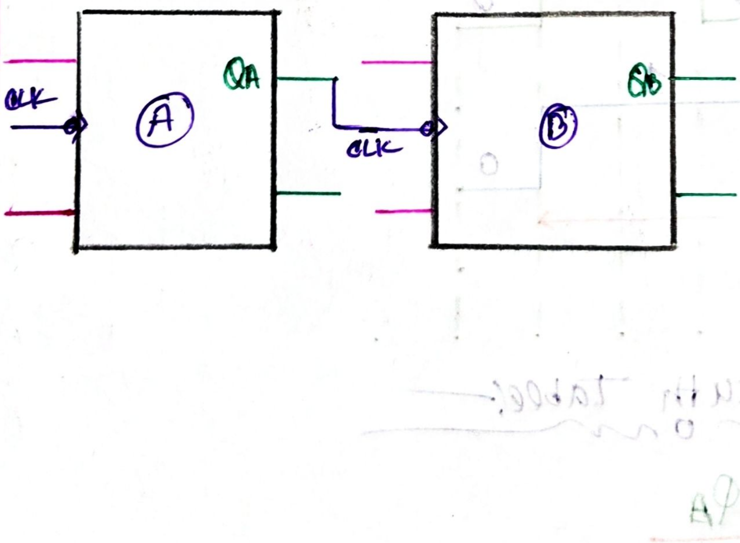
⇒ There are two types:-

1. Asynchronous Counters (Ripple Counters)
2. Synchronous Counters.

Asynchronous Vs Synchronous:-

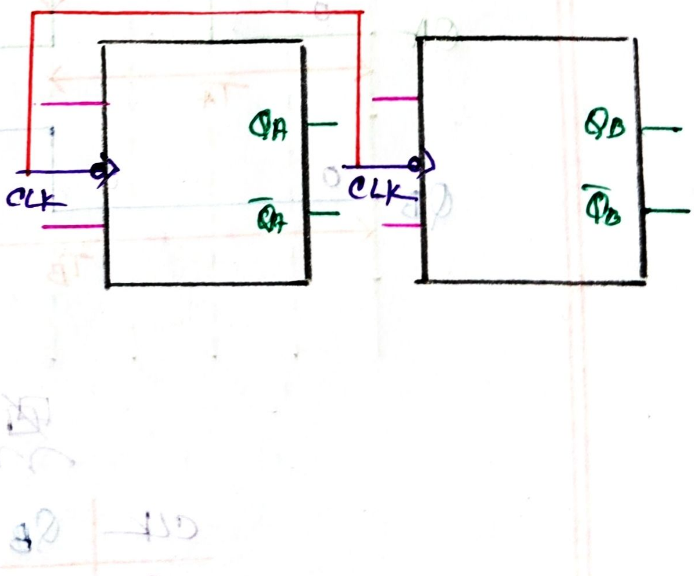
Asynchronous

⇒ Flip Flop are Connected in Such a way that the O/P of 1st flip flop drives the clock of next flip flop



Synchronous:-

⇒ There is no Connection between O/P of 1st flip flop and clock of next flip flop. F.F are clocked Simultaneously



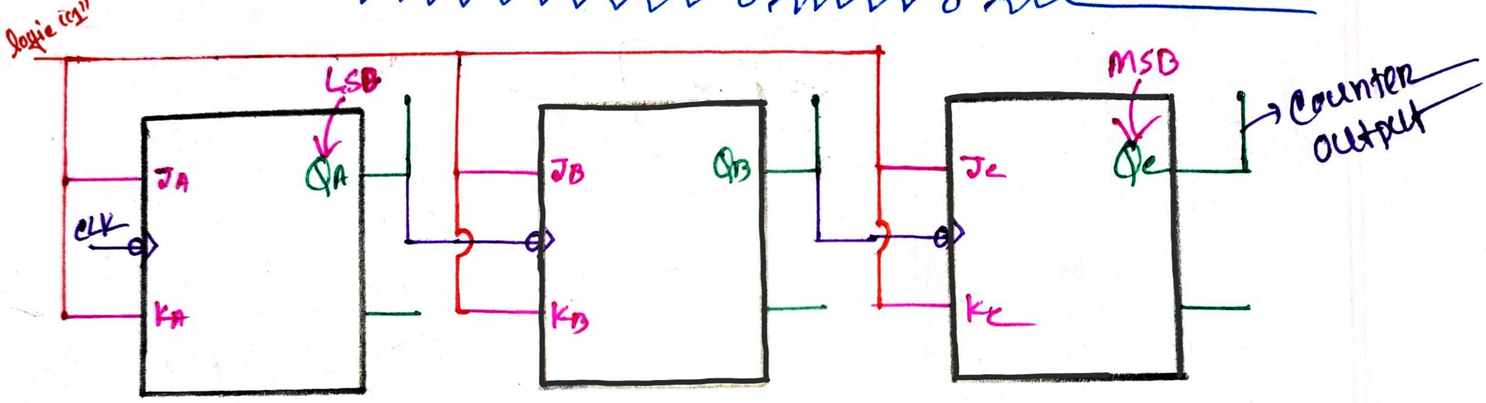
Sub :

Time : Date : / /

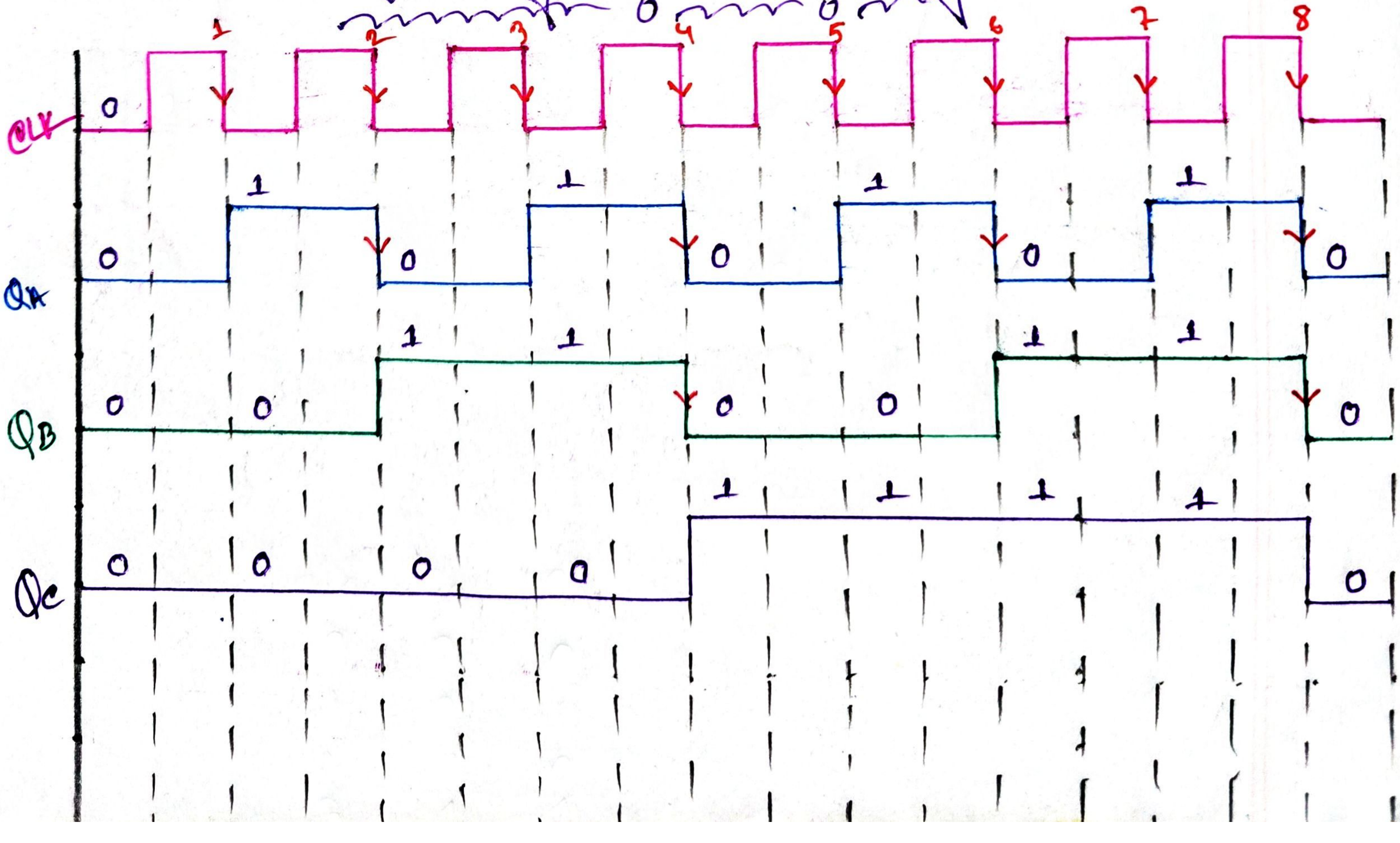
Classification of Counters:-

1. Up Counter
2. Down Counter
3. Up/Down Counter.

3 Bit Asynchronous Up Counter:-



Analyse the Clock Diagram:-

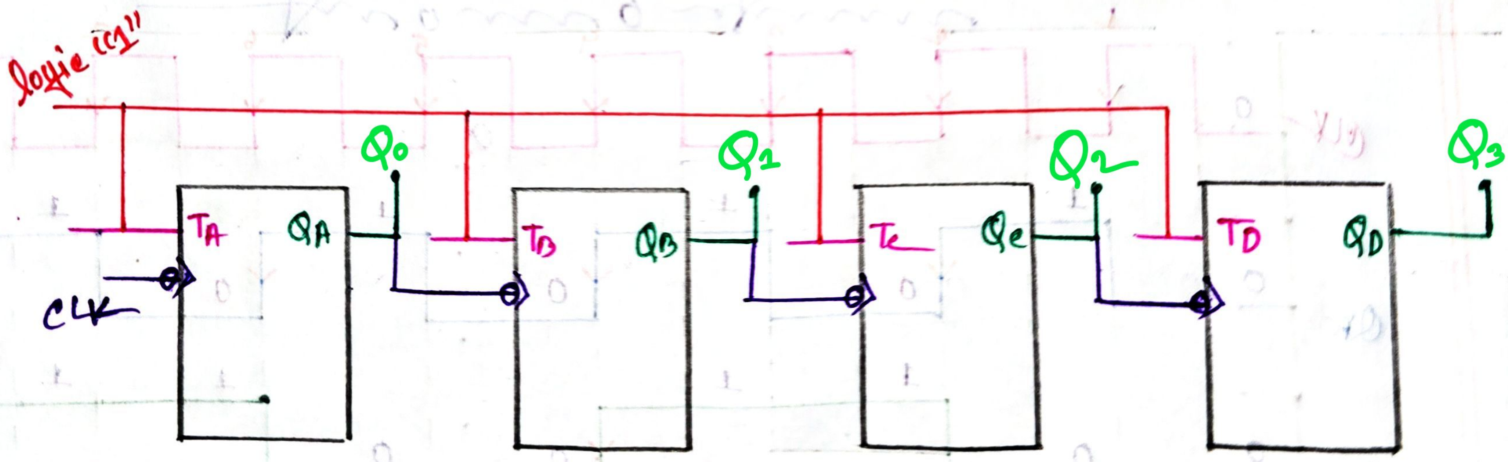


Sub :   Time :   Date :   /   /

Table for 3 bit Asys. Up Counter:-

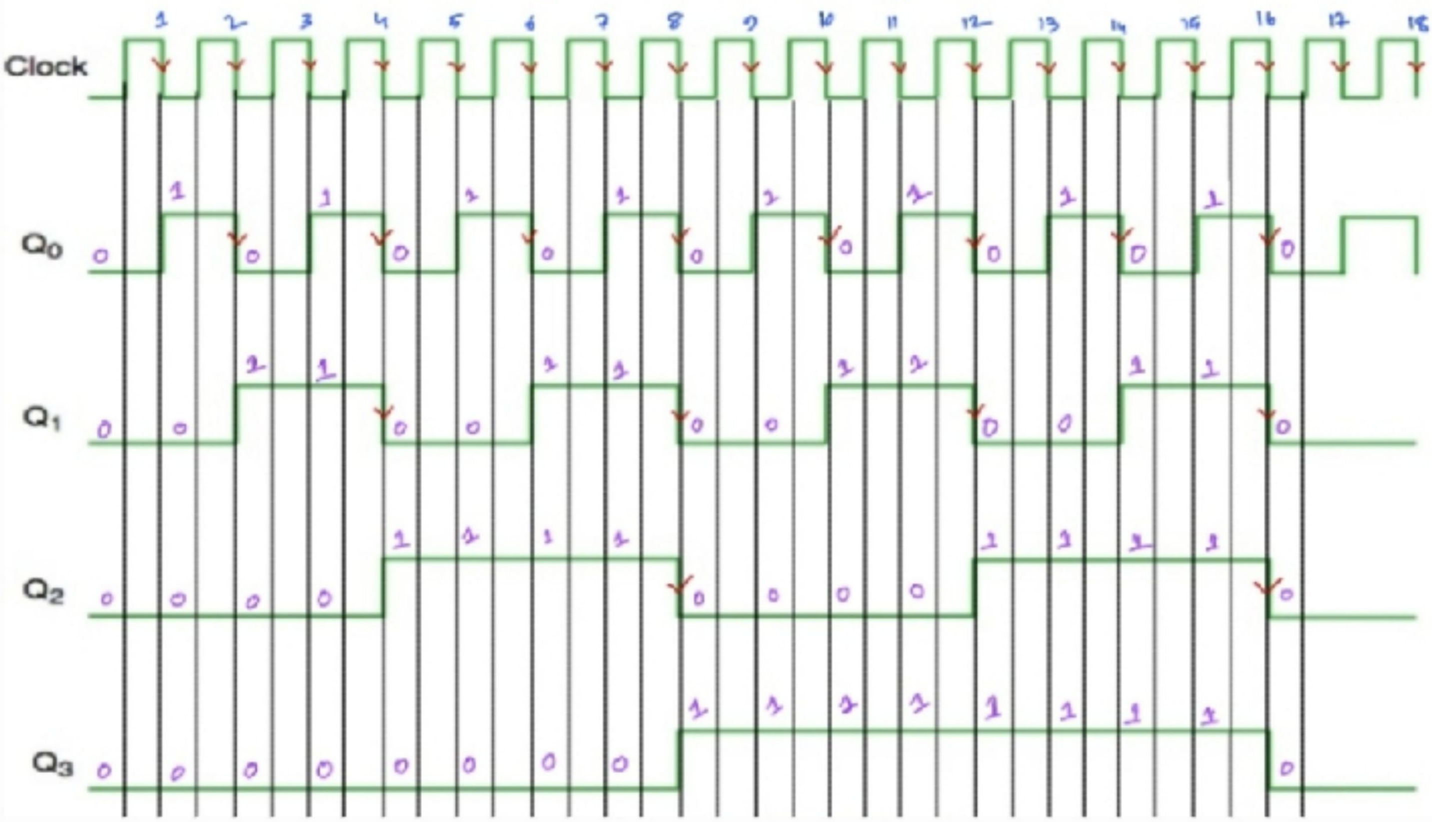
Clock	Q_C	Q_B	Q_A	Decimal eq
Initally	0	0	0	0
$1^{st}(\downarrow)$	0	0	1	1
$2^{nd}(\downarrow)$	0	1	0	2
$3^{rd}(\downarrow)$	0	1	1	3
$4^{th}(\downarrow)$	1	0	0	4
$5^{th}(\downarrow)$	1	0	1	5
$6^{th}(\downarrow)$	1	1	0	6
$7^{th}(\downarrow)$	1	1	1	7

4-Bit Asynchornous Up Counter:-



Time : Date : / /

Analyse the Clock Diagram



Truth table for 4 bit asynchronous Up Counter

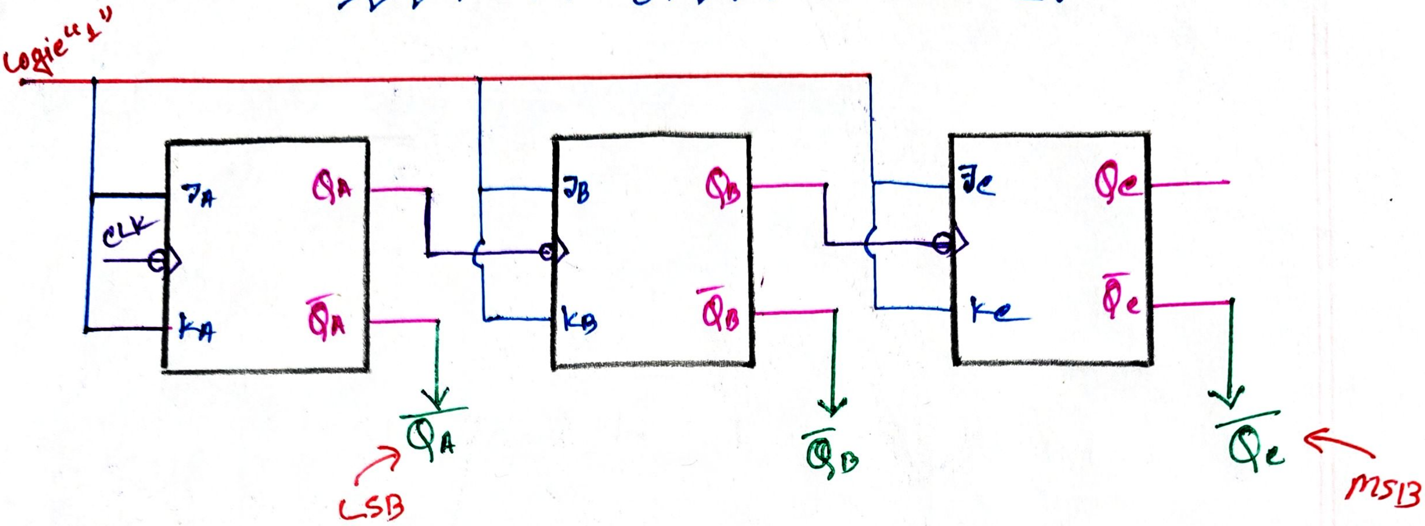
Clock Cycles	Output Bits				Decimal Value
	QA	QB	QC	QD	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	1	0	1	0	10
12	1	0	1	1	11
13	1	1	0	0	12
14	1	1	0	1	13
15	1	1	1	0	14
16	1	1	1	1	15

Sub :

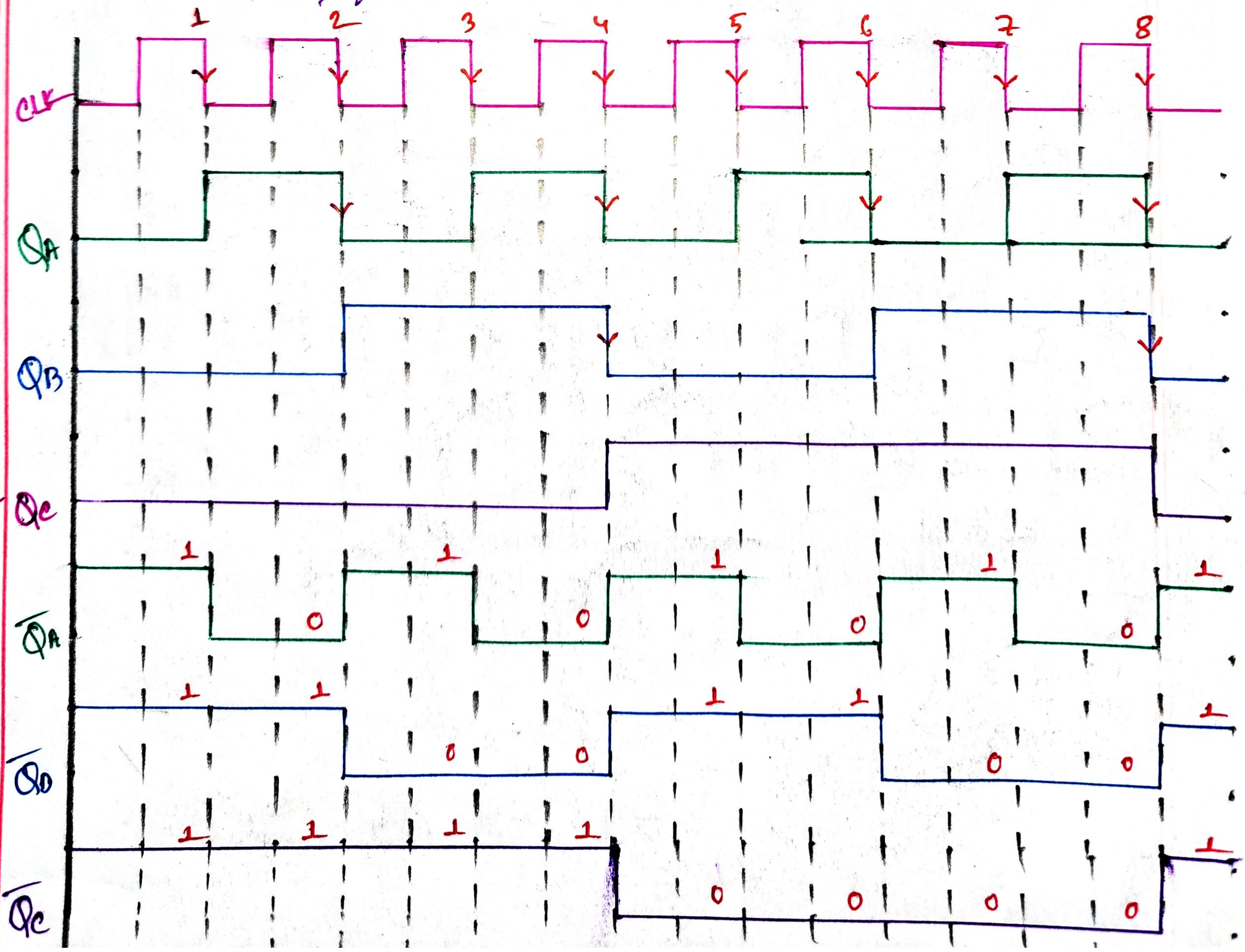
Day

Time : Date : / /

3 Bit Down Counters: (Asynchronous)



Analyse the clock Diagram:-



State Diagram of a Counter:-

⇒ 2-Bit Up Counter:-

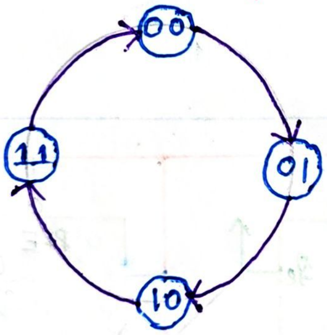
→ Total state for 2 bits = 4

→ MAX Count = 4 − 1 = 3

The combinitions are:-

Q_B	Q_A	Count
0	0	0
0	1	1
1	0	2
1	1	3

State Diagram:-



⇒ 2-Bit Down Counter:-

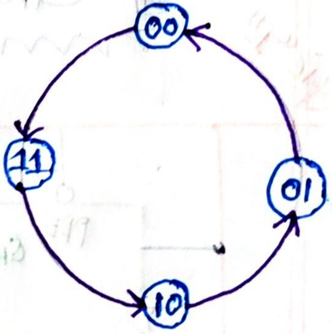
→ Total State for 2bit = 4

→ max Count = 4 − 1 = 3

The combinations are:-

Q_B	Q_A	Count
1	1	3
1	0	2
0	1	1
0	0	0

State Diagram:-



3-bit Up-down Counter:-

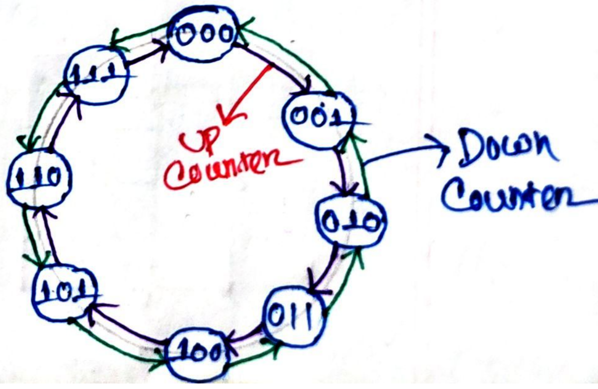
⇒ Total state for 3 bits = 8

⇒ MAX Count = 8 − 1 = 7

The Combinations are:-

Q_C	Q_B	Q_A	Count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

State Diagram:-



Sub : Time : Date : / /

MOD Counters:-

⇒ MOD Counters are cascaded counter circuits which count to a set modulus value before resetting.

Some of them are:-

1. Asynchronous Counter for 2 bit (known as MOD-4)
2. Asynchronous Counter for 3 bit (known as MOD-8)
3. Asynchronous Counter for Decade (known as MOD-10)
4. MOD-12 ripple counter
5. MOD-6 ripple counter

MOD-6 Ripple Counter using MOD-8 Counter:-

T.T for MOD-8

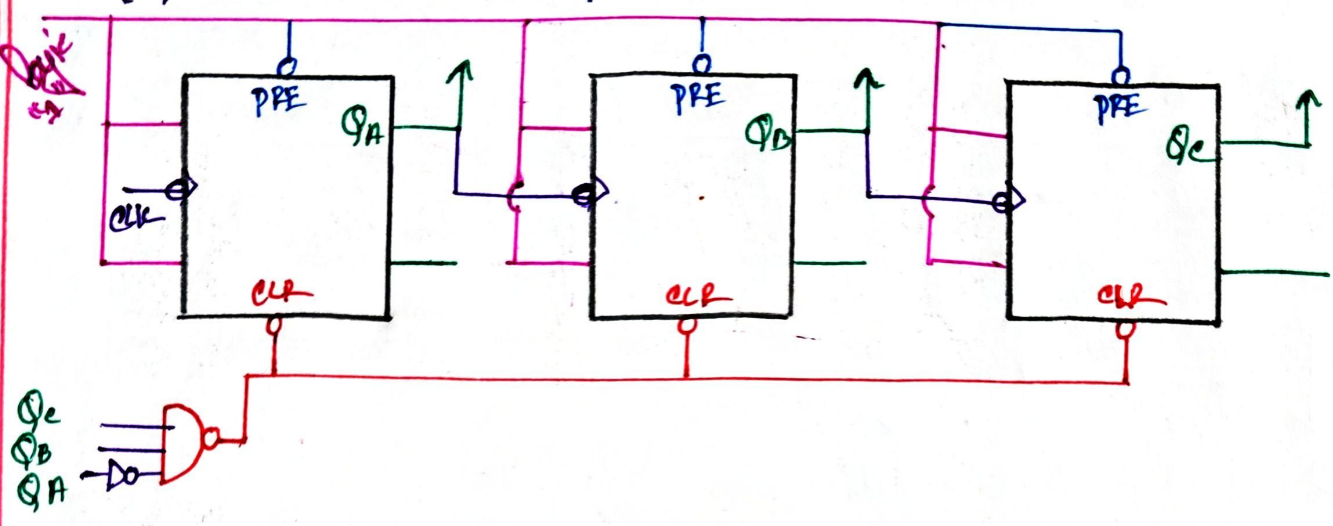
C	B	A	Decimal Value
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Max Count = 6 − 1
= 5

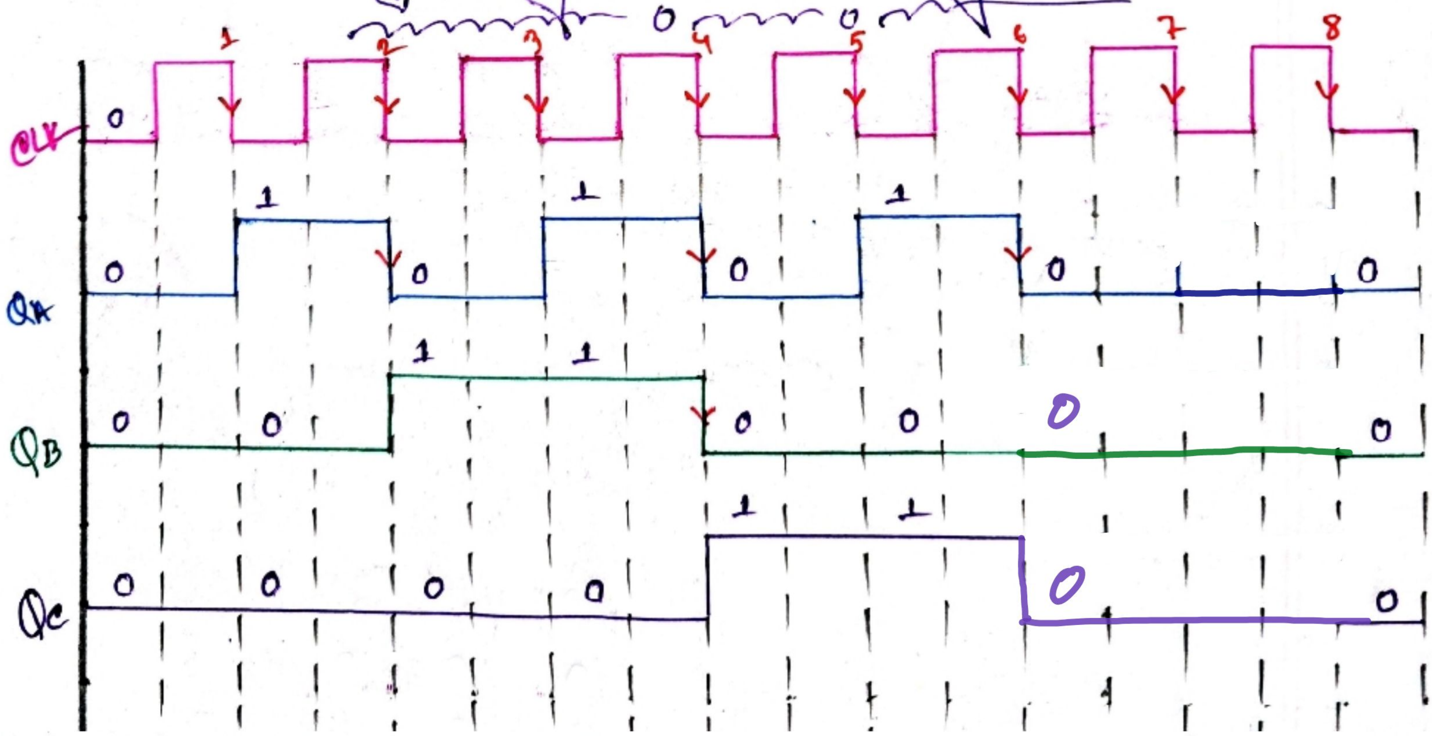
Sub :

Time : Date : / /

Logic circuit



Analyse the Clock Diagram:-



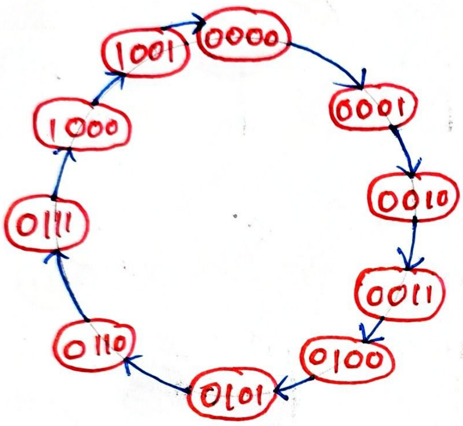
Important points:-

1. Negative edge triggered → Q is clock → Up Counter
2. Positive edge triggered → $\overline{Q}$ in clock → Up Counter
3. Negative edge triggered → $\overline{Q}$ is clock → Down Counter
4. positive edge triggered → Q is clock → Down Counter

#Decade (BCD) Ripple Counter:- (MOD 10)

⇒ Draw the State Diagram:-

here, No. of State = 10

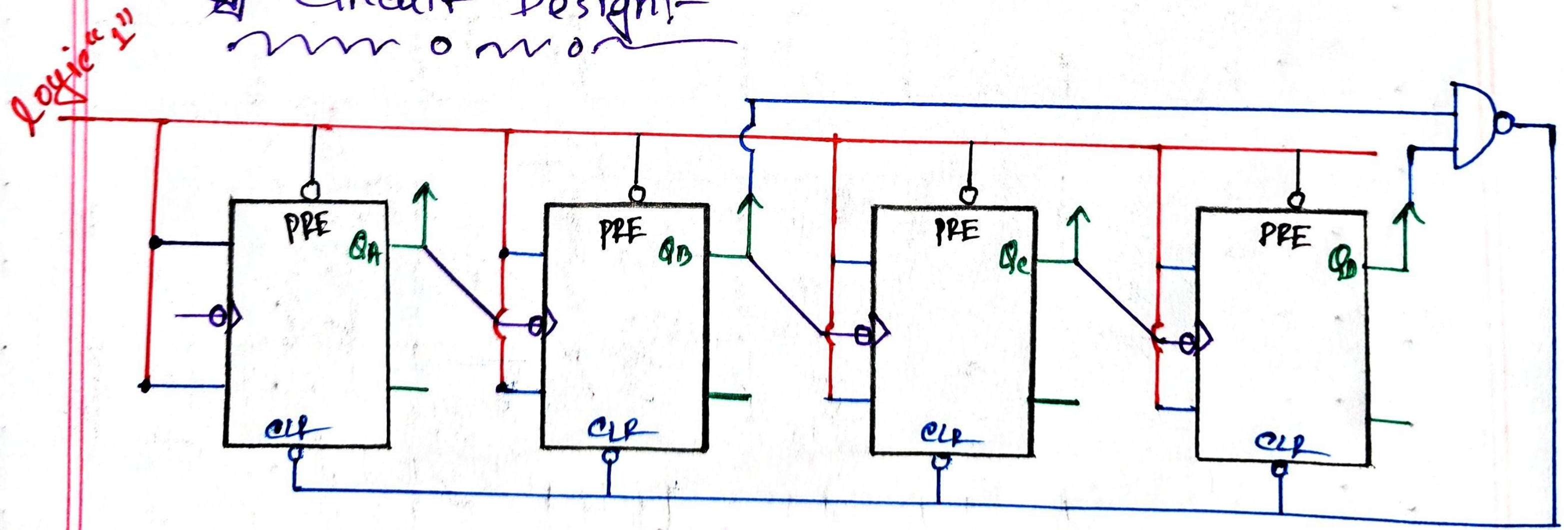


Max Count = 10 − 1
= 9

Truth Table for MOD16

CLK	Q_D	Q_C	Q_B	Q_A	BCD
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	$\boxed{1}$	0	$\boxed{1}$	0	10
12	1	0	1	1	
13	1	1	0	0	
14	1	1	0	1	
15	1	1	1	0	

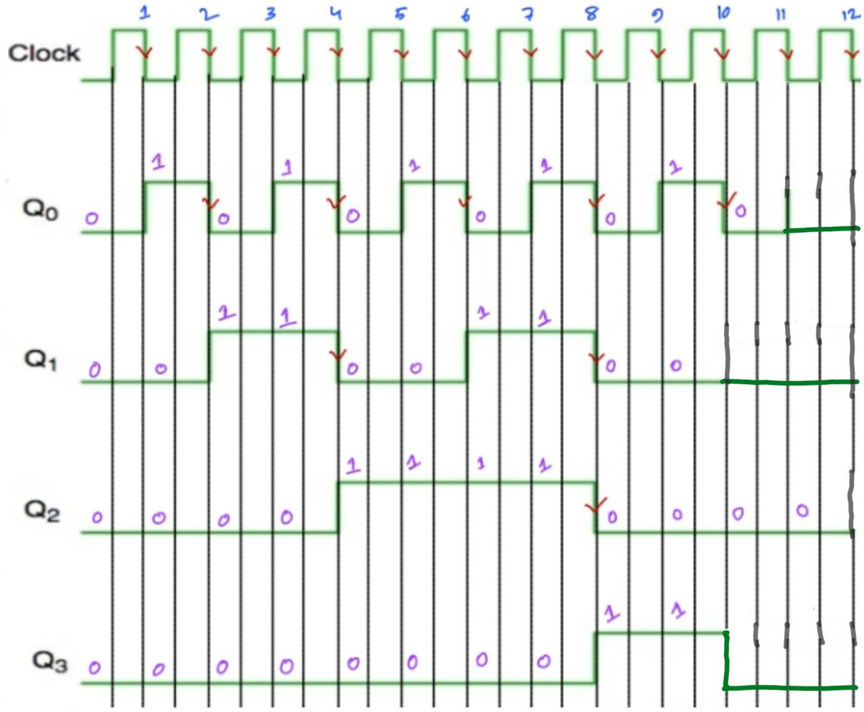
Circuit Design:-



Sub :

Day

Time : Date : / /



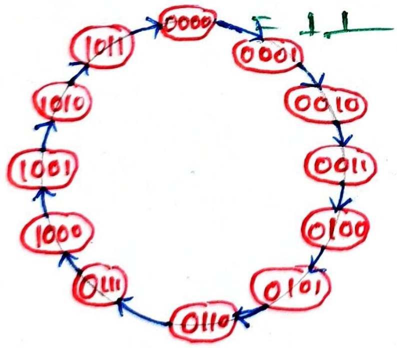
Asynchronous MOD-12 Counter:-

⇒ Draw the state Diagram:-

here, No. of State = 12

Max count = 12 − 1

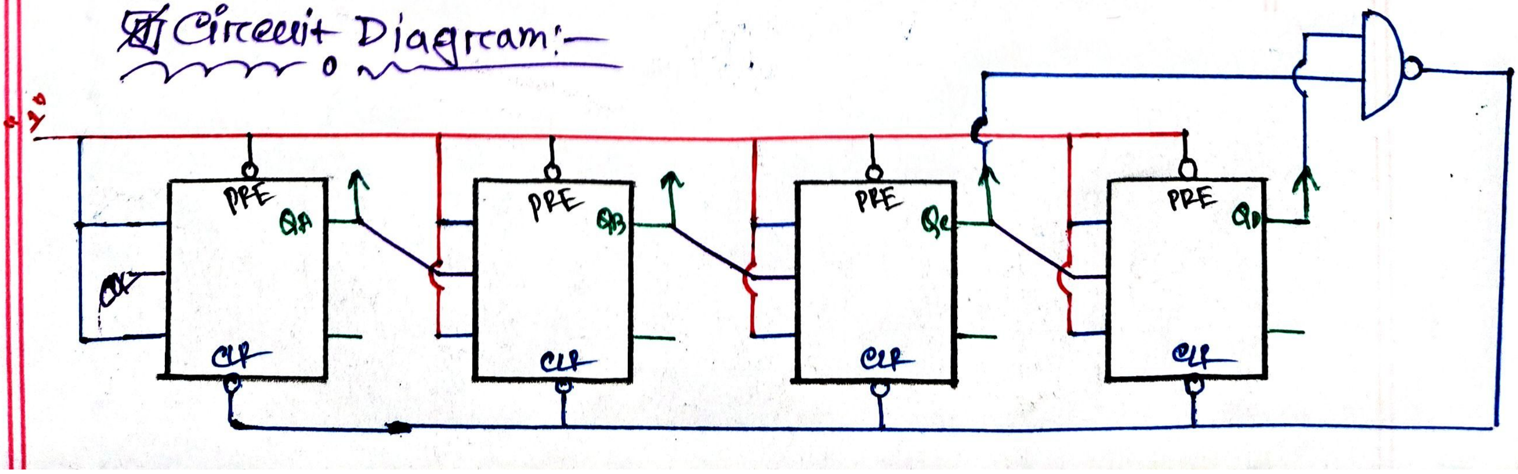
= 11



Truth Table for MOD16

clk	Q_D	Q_C	Q_B	Q_A	BCD
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	1	0	1	0	10
12	1	0	1	1	11
13	$\boxed{1}$	$\boxed{1}$	0	0	12
14	1	1	0	1	
15	1	1	1	0	

☑ Circuit Diagram:-



#Synchronous Counter:-

How to Design Synchronous Counter:-

⇒ Step 1:- Decide the number of flip-flops & which one ?

⇒ Step 2:- Excitation table for the flip-flops.

⇒ Step 3:- Draw state diagram & circuit excitation Table

⇒ Step 4: Obtain simplified equations using "K" map.

⇒ Step 5:- Draw the Logic diagram.

2-bit synchronous up Counter:-

Step: 1:-

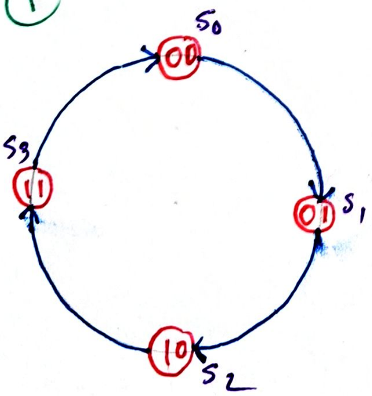
⇒ For 2bit we need two Flip Flop (JK F.F) →

Step: 2:-

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

Step: 3:-

(I)



(II) 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

Sub : Time : Date : / /

Step: 4:-

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

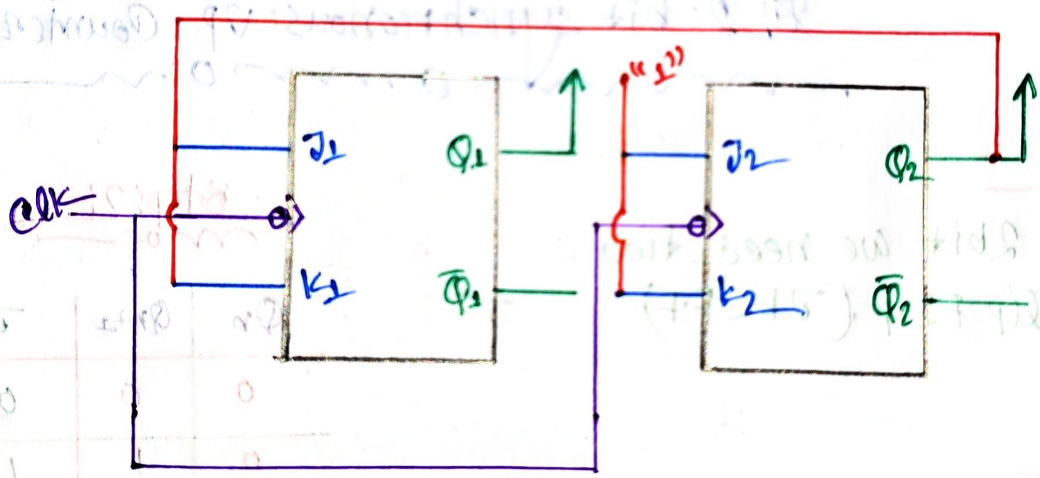
$J_2 = 1$

For K_2:-

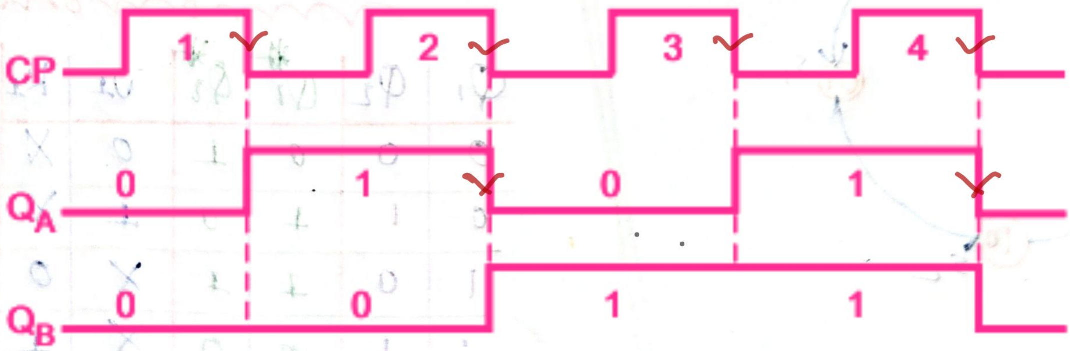
$Q_1 \backslash Q_2$	0	1
0	X	1
1	X	1

$K_2 = 1$

Step: 5:-



Alalyse the output wave:



3-Bit Synchronous Up Counter using JK flip flop:-

⇒ Step 1:-

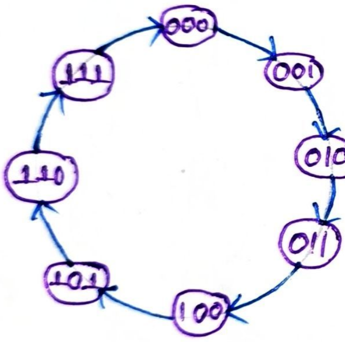
For 3 bit we need 3 JK flip flop

⇒ Step 2:-

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

⇒ Step 3:-

(i) State diagram:-



(ii) Ckt excitation table:-

Q_C	Q_B	Q_A	Q_C^*	Q_B^*	Q_A^*	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

⇒ Step 4:-

K-map for J_C

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$\therefore J_C = Q_BQ_A$

K-map for K_C

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	X	X	X	X
1	0	0	1	0

$\therefore K_C = Q_BQ_A$

K-map for J_B

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$\therefore J_B = Q_A$

K-map for K_B

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	X	X	1	0
1	X	X	1	0

$\therefore K_B = Q_A$

∴ K-map for J_A

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	1	X	X	1
1	1	X	X	1

∴ $J_A = 1$

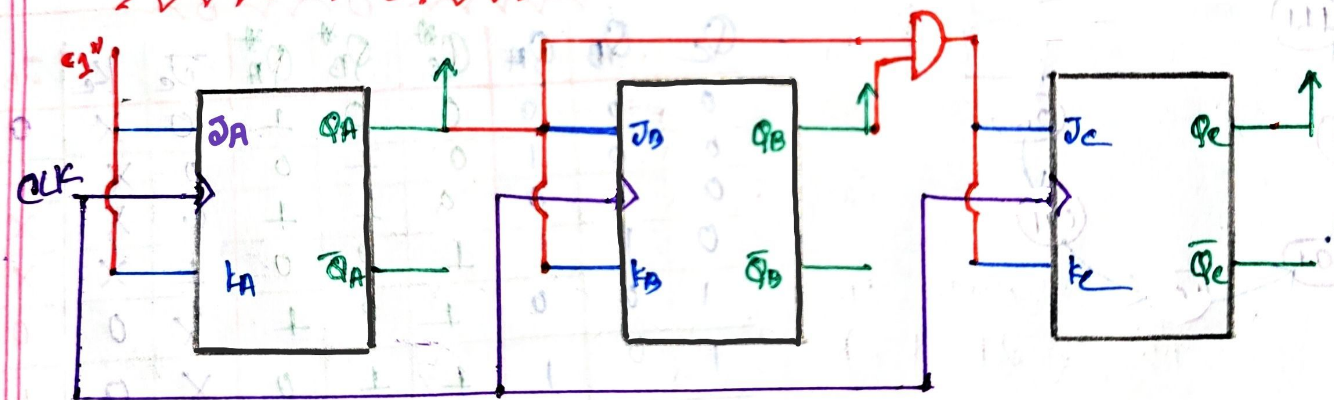
∴ K-map for K_A

$Q_C \backslash Q_BQ_A$	00	01	11	10
0	X	1	1	X
1	X	1	1	X

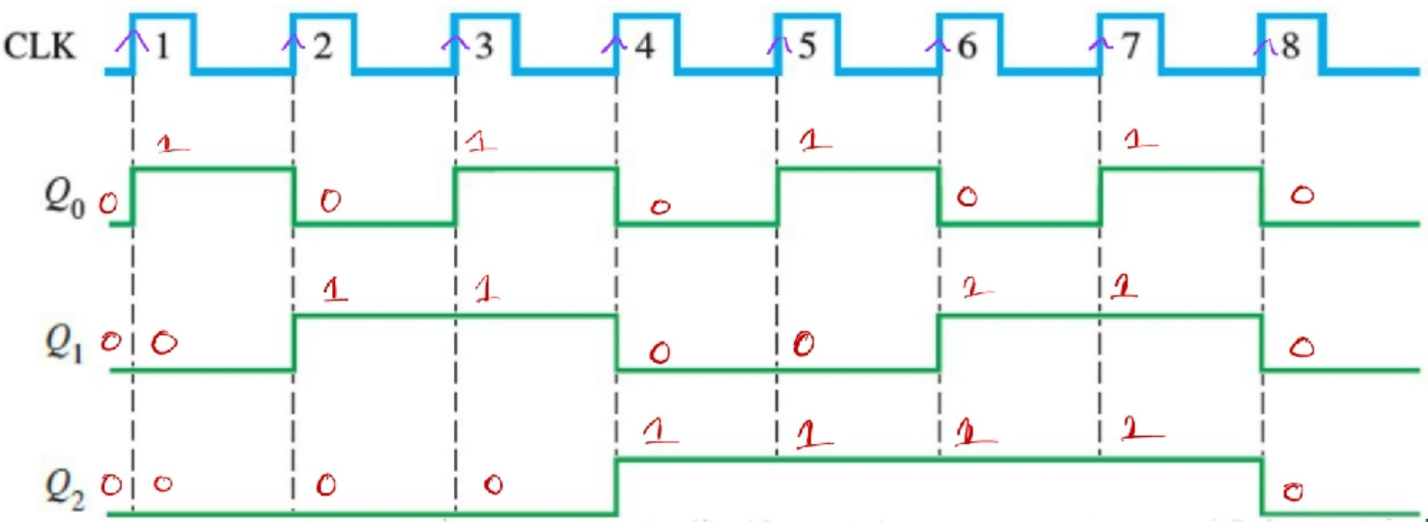
∴ $K_A = 1$

∴ Step: 5:-

⇒ Draw the logic circuit:-



Clock Diagram:-



4-Bit Synchronous Up Counter Using JK flip flop

Step 1:

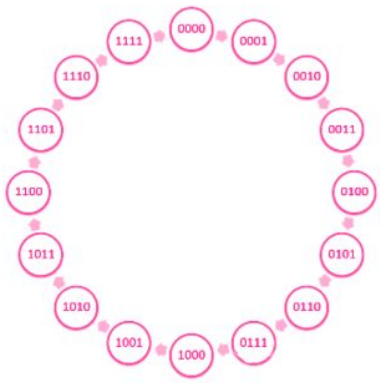
➔ For 4 bit we need 4 JK flip flop

Step 2:

➔ Draw the JK Excitation Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3.1:



Step 3.2:

CKT Excitation Table

Q_D	Q_C	Q_B	Q_A	Q_D^*	Q_C^*	Q_B^*	Q_A^*	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1
1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
1	0	1	1	1	1	0	0	X	0	1	X	X	1	X	1
1	1	0	0	1	1	0	1	X	0	X	0	0	X	1	X
1	1	0	1	1	1	1	0	X	0	X	0	1	X	X	1
1	1	1	0	1	1	1	1	X	0	X	0	X	0	1	X
1	1	1	1	0	0	0	0	X	1	X	1	X	1	X	1

Step 4:

➔K-Map for J_D:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$\therefore J_D = Q_C Q_B Q_A$

➔K-Map for K_D:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	0	0	1	0
10	0	0	0	0

$\therefore K_D = Q_C Q_B Q_A$

➔K-Map for J_C:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	0	0	1	0

$\therefore J_C = Q_B Q_A$

➔K-Map for K_C:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	X	X	X
01	0	0	1	0
11	0	0	1	0
10	X	X	X	X

$\therefore K_C = Q_B Q_A$

➔K-Map for J_B:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	X	X
01	0	1	X	X
11	0	1	X	X
10	0	1	X	X

$\therefore J_B = Q_A$

➔K-Map for K_B:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	X	1	0
01	X	X	1	0
11	X	X	1	0
10	X	X	1	0

$\therefore K_B = Q_A$

➔K-Map for J_A:

$Q_DQ_C \backslash Q_BQ_A$	00	01	11	10
00	1	X	X	1
01	1	X	X	1
11	1	X	X	1
10	1	X	X	1

$\therefore J_A = 1$

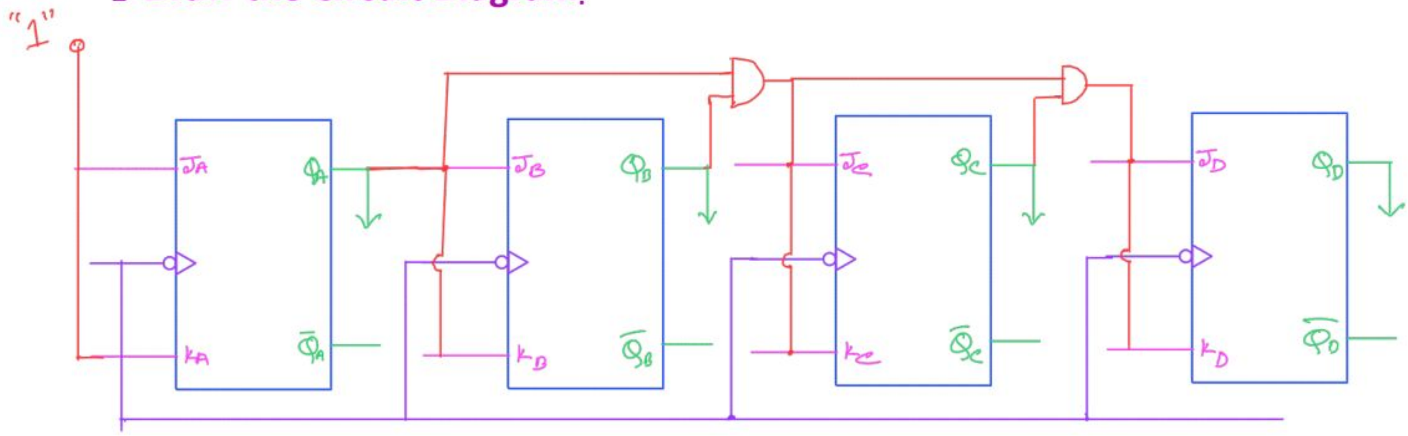
➔K-Map for K_A:

$Q_DQ_C \backslash Q_BQ_A$	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	1	1	X
10	X	1	1	X

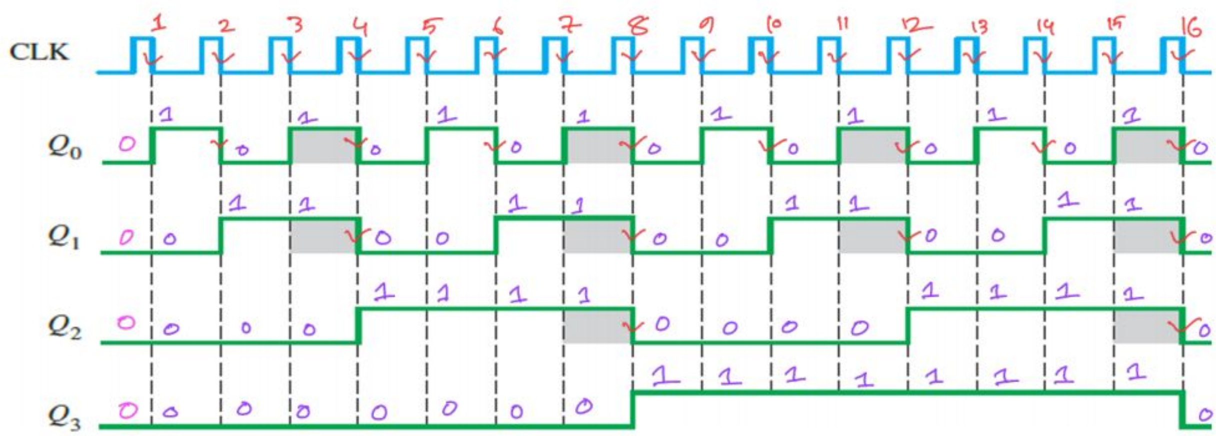
$\therefore K_A = 1$

Step 5:

➔Draw the Circuit Diagram:



Timing diagram of 4-Bit Synchronous Up Counter:



4-Bit Synchronous Decade BCD Counter Using JK flip flop

Step 1:

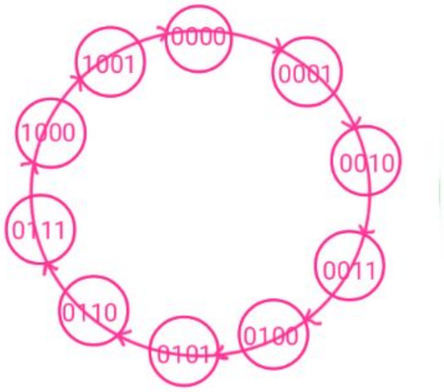
➔ For 4 bit we need 4 JK flip flop

Step 2:

➔ Draw the JK Excitation Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3.1:



Step 3.2:

CKT Excitation Table

Q_D	Q_C	Q_B	Q_A	Q_D^*	Q_C^*	Q_B^*	Q_A^*	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	X	1	0	X	0	X	X	1

Step 4:

➔ K-Map for J_D:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$\therefore J_D = Q_C Q_B Q_A$

➔ K-Map for K_D:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	X	X
10	0	1	X	X

$\therefore K_D = Q_A$

➔ K-Map for J_C:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	0	0	X	X

$\therefore J_C = Q_B Q_A$

➔ K-Map for K_C:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	X	X	X
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$\therefore K_C = Q_B Q_A$

➔ K-Map for J_B:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	X	X
01	0	1	X	X
11	X	X	X	X
10	0	0	X	X

$\therefore J_B = \overline{Q_D} Q_A$

➔ K-Map for K_B:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	X	1	0
01	X	X	1	0
11	X	X	X	X
10	X	X	X	X

$\therefore K_B = Q_A$

➔K-Map for J_A:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	1	X	X	1
01	1	X	X	1
11	X	X	X	X
10	1	X	X	X

$\therefore J_A = 1$

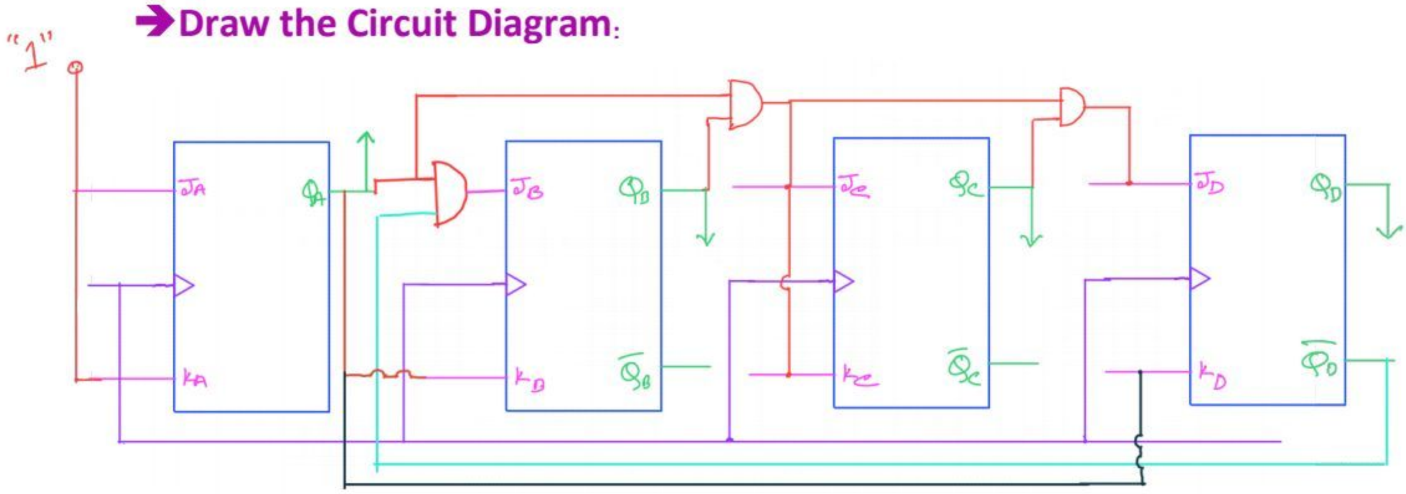
➔K-Map for K_A:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	X	X	X
10	X	1	X	X

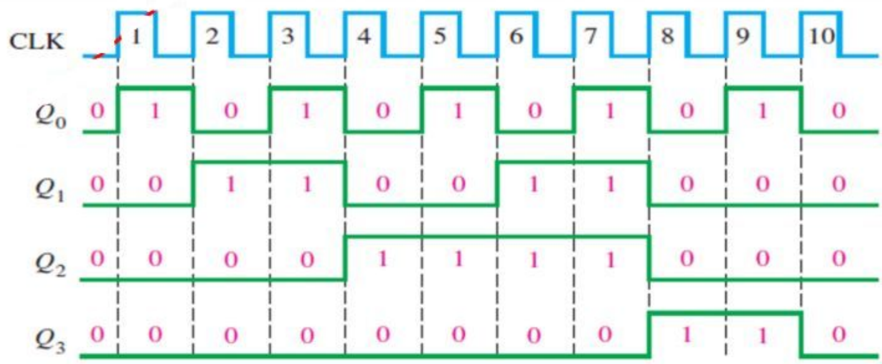
$\therefore K_A = 1$

Step 5:

➔Draw the Circuit Diagram:



Timing diagram of 4-Bit Synchronous Up Counter



UP/DOWN Counters:

3-Bit Asynchronous UP/DOWN Counter Using JK flip flop

1. We need control input **M**
2. Let, if M=0 → Up Counting → Q connected to the Clock
   if M=1 →Down Counting → Q' connected to the Clock

➔Truth Table:

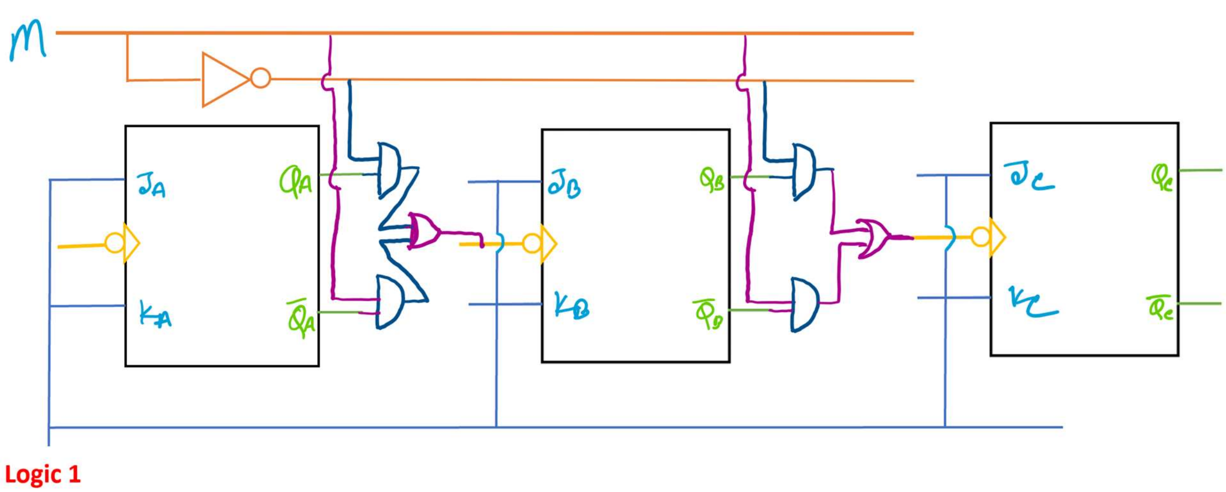
M	Q	Q'	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

➔K-Map for Y:

M \ QQ'	00	01	11	10
0	0	0	1	1
1	0	1	1	0

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

➔Draw the logic circuit:



3-Bit Synchronous UP/DOWN Counter Using JK flip flop

1. We need control input **M**
2. Let, if M=0 → Up Counting

   if M=1 →Down Counting

Step 1:

→ For 3 bit we need 3 JK flip flop

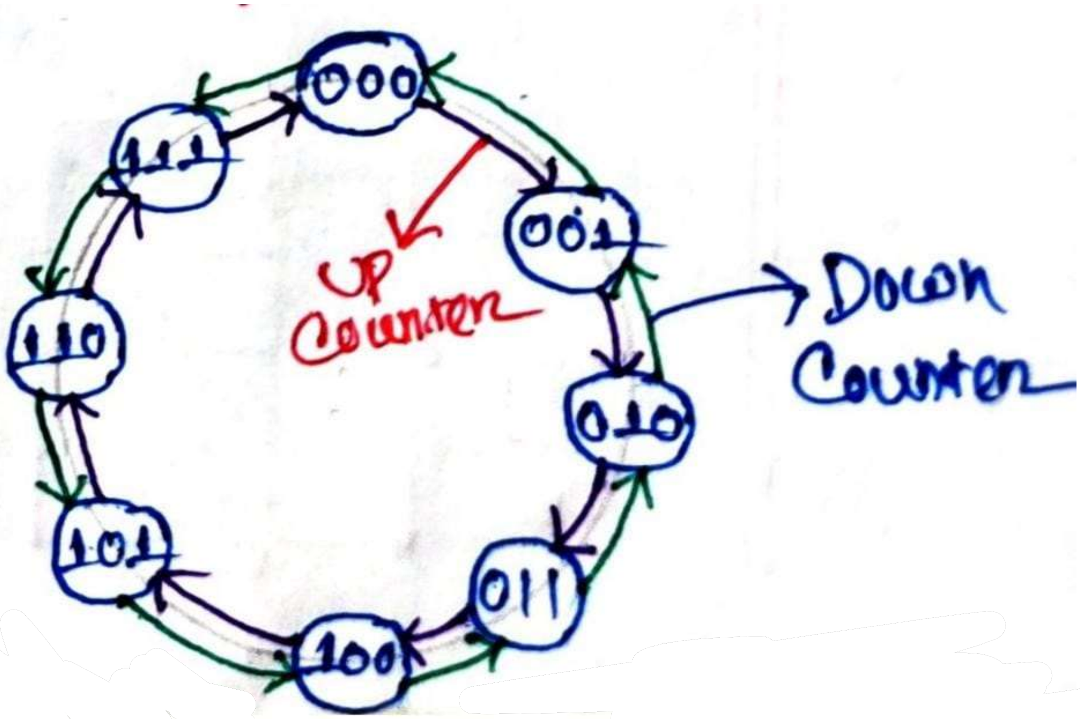
Step 2:

→ Draw the JK Excitation Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3.1:

→**Draw State Diagram:**



Step 3.2:

CKT Excitation Table

M	Q_C	Q_B	Q_A	Q_C^*	Q_B^*	Q_A^*	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	1	0	X	0	X	1	X
0	0	0	1	0	1	0	0	X	1	X	X	1
0	0	1	0	0	1	1	0	X	X	0	1	X
0	0	1	1	1	0	0	1	X	X	1	X	1
0	1	0	0	1	0	1	X	0	0	X	1	X
0	1	0	1	1	1	0	X	0	1	X	X	1
0	1	1	0	1	1	1	X	0	X	0	1	X
0	1	1	1	0	0	0	X	1	X	1	X	1
1	0	0	0	1	1	1	1	X	1	X	1	X
1	0	0	1	0	0	0	0	X	0	X	X	1
1	0	1	0	0	0	1	0	X	X	1	1	X
1	0	1	1	0	1	0	0	X	X	0	X	1
1	1	0	0	0	1	1	X	1	1	X	1	X
1	1	0	1	1	0	0	X	0	0	X	X	1
1	1	1	0	1	0	1	X	0	X	1	1	X
1	1	1	1	1	1	0	X	0	X	0	X	1

Step 4:

➔K-Map for J_C:

MQ_C \ Q_BQ_A	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	1	0	0	0

$J_C = MQ'_BQ'_A + M'Q_BQ_A$

➔K-Map for K_C:

MQ_C \ Q_BQ_A	00	01	11	10
00	X	X	X	X
01	0	0	1	0
11	1	0	0	0
10	X	X	X	X

$K_C = MQ'_BQ'_A + M'Q_BQ_A$

➔K-Map for J_B:

MQ_C \ Q_BQ_A	00	01	11	10
00	0	1	X	X
01	0	1	X	X
11	1	0	X	X
10	1	0	X	X

$J_B = MQ'_A + M'Q_A$

➔K-Map for K_B:

MQ_C \ Q_BQ_A	00	01	11	10
00	X	X	1	0
01	X	X	1	0
11	X	X	0	1
10	X	X	0	1

$K_B = MQ'_A + M'Q_A$

➔K-Map for J_A:

MQ_C \ Q_BQ_A	00	01	11	10
00	1	X	X	1
01	1	X	X	1
11	1	X	X	1
10	1	X	X	1

$J_A = 1$

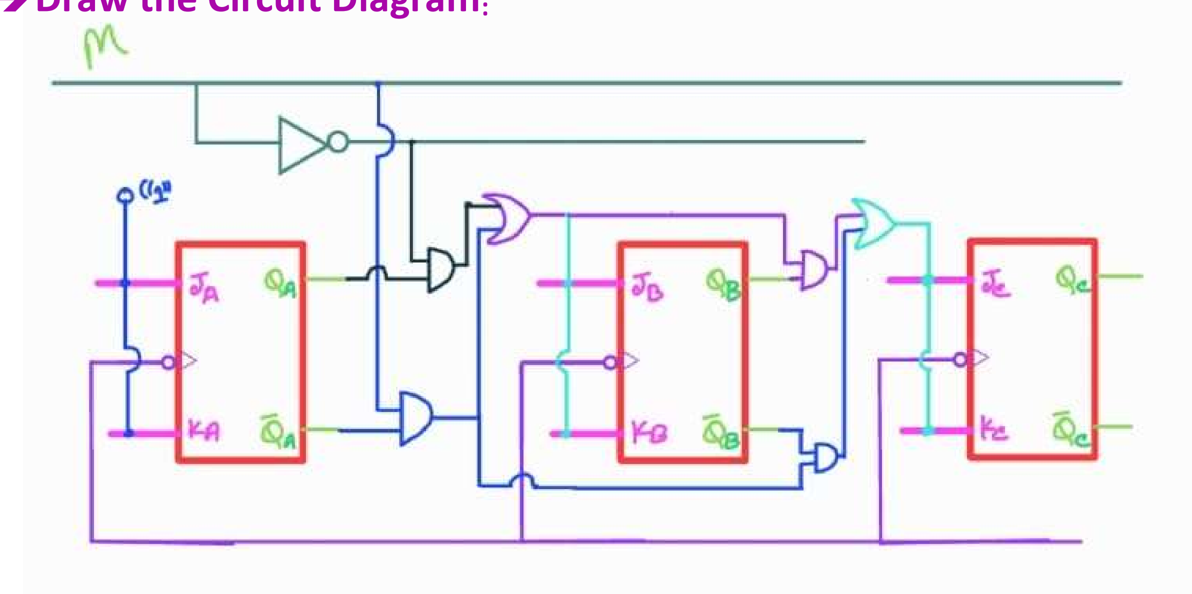
➔K-Map for K_A:

MQ_C \ Q_BQ_A	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	1	1	X
10	X	1	1	X

$K_A = 1$

Step 5:

➔Draw the Circuit Diagram:



Irregular Sequence Counter Design

Design a counter with the sequence 0,1,3,2,6,7,5 and 4 using JK flip flop

Step 1:

➔ For 3 bit we need 3 JK flip flop

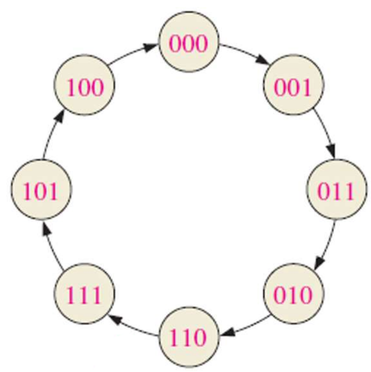
Step 2:

➔ Draw the JK Excitation Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3.1:

➔Draw State Diagram:



Step 3.2:

CKT Excitation Table

Q_C	Q_B	Q_A	Q_C^*	Q_B^*	Q_A^*	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	1	0	1	0	0	X	X	0	X	1
0	1	0	1	1	0	1	X	X	0	0	X
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	1	X	0	X	1	X	0
1	0	1	1	0	0	X	0	0	X	X	1
1	0	0	0	0	0	X	1	0	X	0	X

Step 4:

➔K-Map for J_C:

Q_C \ $Q_B Q_A$	00	01	11	10
0	0	0	0	1
1	X	X	X	X

$J_C = Q_B Q'_A$

➔K-Map for K_C:

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	X	X	X
1	1	0	0	0

$K_C = Q'_B Q'_A$

➔K-Map for J_B:

Q_C \ $Q_B Q_A$	00	01	11	10
0	0	1	X	X
1	0	0	X	X

$J_B = Q'_C Q_A$

➔K-Map for K_B:

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	X	0	0
1	X	X	1	0

$K_B = Q_C Q_A$

➔K-Map for J_A:

Q_C \ $Q_B Q_A$	00	01	11	10
0	1	X	X	0
1	0	X	X	1

$J_A = Q'_C Q'_B + Q_C Q_B$

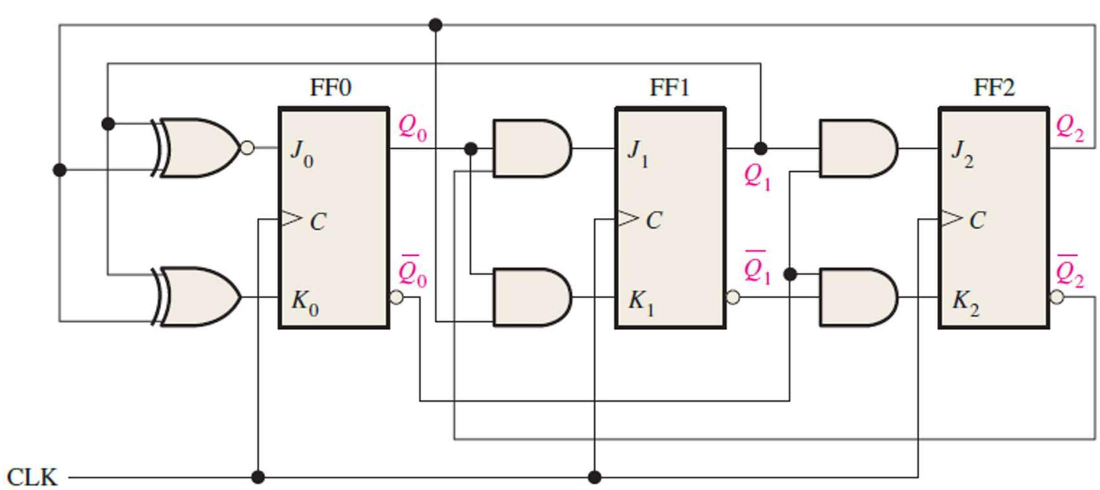
➔K-Map for J_A:

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	0	1	X
1	X	1	0	X

$K_A = Q_C Q'_B + Q'_C Q_B$

Step 5:

➔Draw the Circuit Diagram:



Design a counter with the sequence 1,2,5 and 7 using JK flip flop

Step 1:

➔ For 3 bit we need 3 JK flip flop

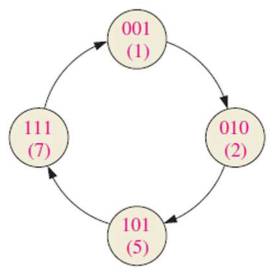
Step 2:

➔ Draw the JK Excitation Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3.1:

➔Draw State Diagram:



Step 3.2:

CKT Excitation Table

Q_C	Q_B	Q_A	Q_C^*	Q_B^*	Q_A^*	J_C	K_C	J_B	K_B	J_A	K_A
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	1	1	X	X	1	1	X
1	0	1	1	1	1	X	0	1	X	X	0
1	1	1	0	0	1	X	1	X	1	X	0

Step 4:

➔K-Map for J_C:

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	0	X	1
1	X	X	X	X

$J_C = Q_B$

➔K-Map for K_C:

Q_C \ $Q_B Q_A$	00	01	11	10
0	X	X	X	X
1	X	0	1	X

$K_C = Q_B$

➔K-Map for J_B:

Q_C \ Q_BQ_A	00	01	11	10
0	X	1	X	X
1	X	1	X	X

$J_B = 1$

➔K-Map for K_B:

Q_C \ Q_BQ_A	00	01	11	10
0	X	X	X	1
1	X	X	1	X

$K_B = 1$

➔K-Map for J_A:

Q_C \ Q_BQ_A	00	01	11	10
0	X	X	X	1
1	X	X	X	X

$J_A = 1$

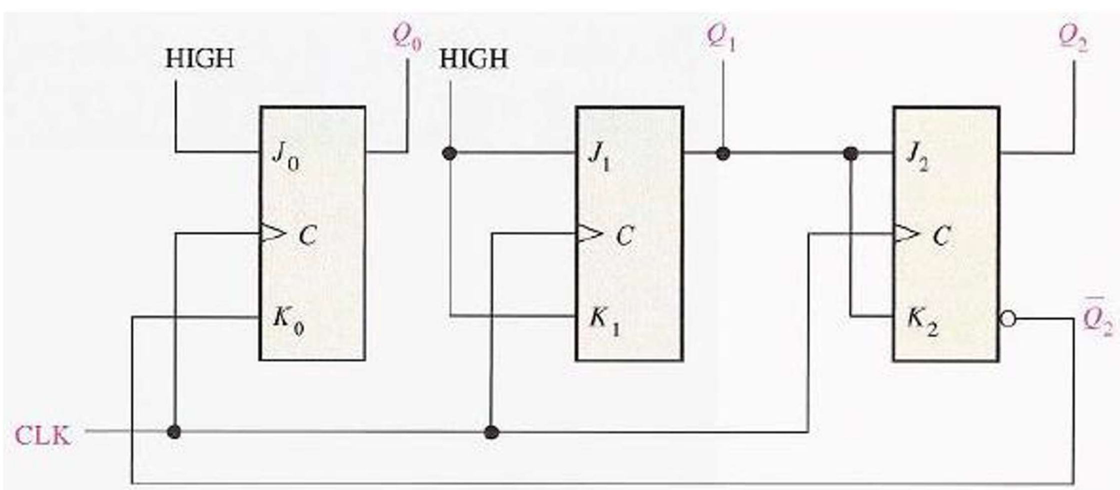
➔K-Map for J_A:

Q_C \ Q_BQ_A	00	01	11	10
0	X	1	X	X
1	X	0	0	X

$K_A = Q_C'$

Step 5:

➔Draw the Circuit Diagram:



Develop a synchronous 3-bit up/down counter with a Gray code sequence using J-K flip-flops. The counter should count up when an UP/DOWN control input is 1 and count down when the control input is 0. (0,1,3,2,6,7,5,4)

1. We need control input **Y**
2. Is given, if Y=1 → Up Counting

   if Y=0 → Down Counting

Step 1:

→ For 3 bit we need 3 JK flip flop

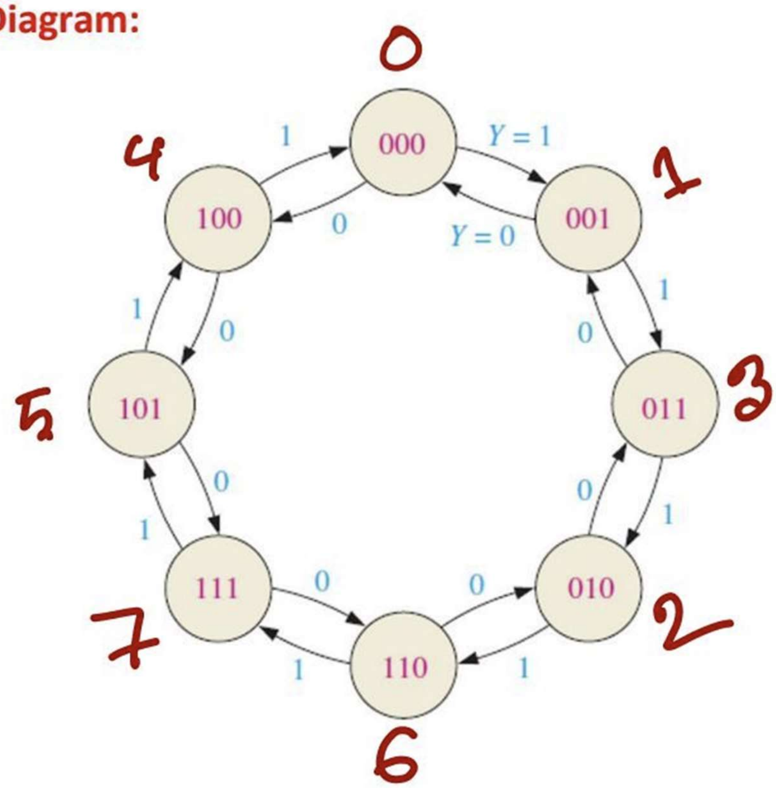
Step 2:

→ Draw the JK Excitation Table:

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 3.1:

→Draw State Diagram:



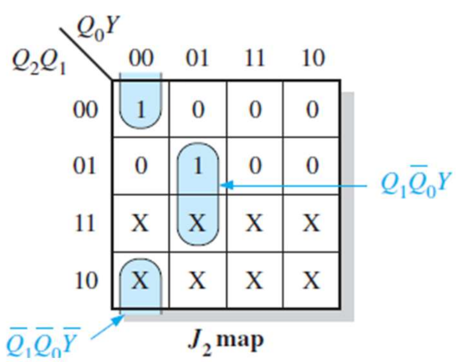
Step 3.2:

CKT Excitation Table

Q_2	Q_1	Q_0	Y	Q_2^*	Q_1^*	Q_0^*	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	1	0	0	1	X	0	X	0	X
0	0	1	0	0	0	0	0	X	0	X	X	1
0	1	1	0	0	0	1	0	X	X	1	X	0
0	1	0	0	0	1	1	0	X	X	0	1	X
1	1	0	0	0	1	0	X	1	X	0	0	X
1	1	1	0	1	1	0	X	0	X	0	X	1
1	0	1	0	1	1	1	X	0	1	X	X	0
1	0	0	0	1	0	1	X	0	0	X	1	X
0	0	0	1	0	0	1	0	X	0	X	1	X
0	0	1	1	0	1	1	0	X	1	X	X	0
0	1	1	1	0	1	0	0	X	X	0	X	1
0	1	0	1	1	1	0	1	X	X	0	0	X
1	1	0	1	1	1	1	X	0	X	0	1	X
1	1	1	1	1	0	1	X	0	X	0	X	0
1	0	1	1	1	0	0	X	0	0	X	X	1
1	0	0	1	0	0	0	X	1	0	X	X	0

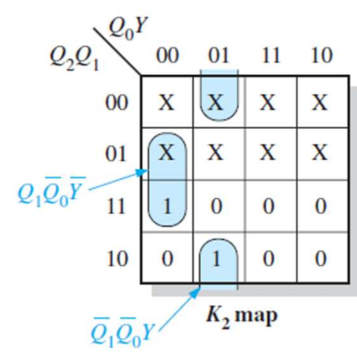
Step 4:

➔K-Map for J_C/J_2:



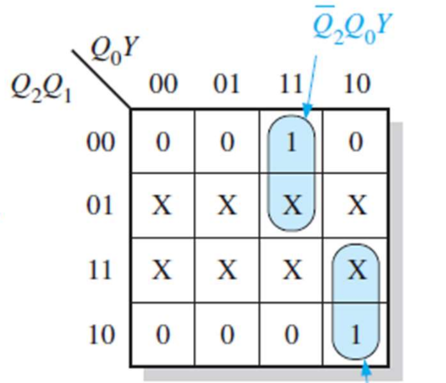
$$J_2 = Q_1\overline{Q}_0Y + \overline{Q}_1\overline{Q}_0\overline{Y}$$

➔K-Map for K_C/K_2:



$$K_2 = Q_1\overline{Q}_0\overline{Y} + \overline{Q}_1\overline{Q}_0Y$$

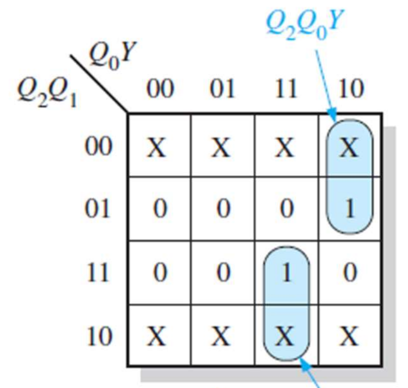
➔K-Map for J_B/J_1:



J_1 map

$$J_1 = \overline{Q}_2Q_0Y + Q_2Q_0\overline{Y}$$

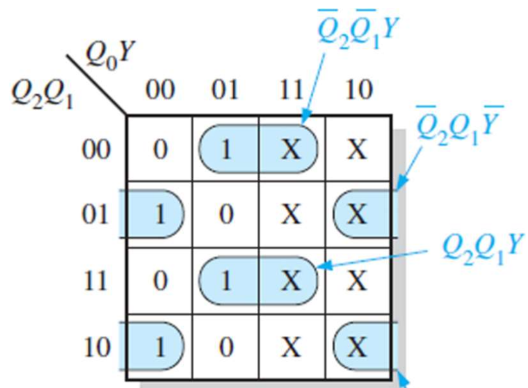
➔K-Map for K_B/K_1:



K_1 map

$$K_1 = \overline{Q}_2Q_0\overline{Y} + Q_2Q_0Y$$

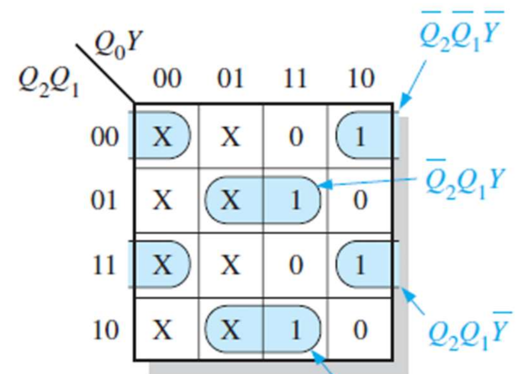
➔K-Map for J_A/J_0:



J_0 map

$$J_0 = Q_2Q_1Y + Q_2\overline{Q}_1\overline{Y} + \overline{Q}_2\overline{Q}_1Y + \overline{Q}_2Q_1\overline{Y}$$

➔K-Map for K_A/K_0:



K_0 map

$$K_0 = \overline{Q}_2\overline{Q}_1\overline{Y} + \overline{Q}_2Q_1Y + Q_2\overline{Q}_1Y + Q_2Q_1\overline{Y}$$

SHIFT REGISTERS

Shift Registers

⇨ **It is a Digital circuit with Two basic Function.**

1. **Data Storage &**
2. **Data Movement**

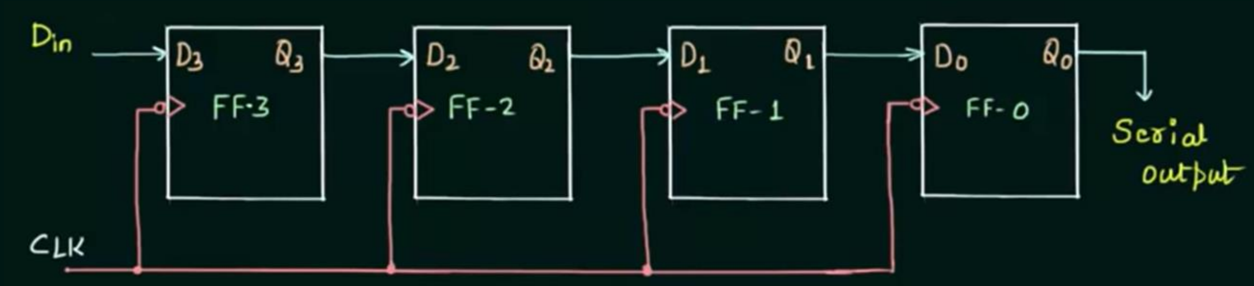
Note: Each flop-flop Stores 1 bit, & the number of flip-flops determine the registers Storage capacity.

Types of Shift Registers

1. **Serial In Serial Out (SISO)**
2. **Serial In Parallel Out (SIPO)**
3. **Parallel In Serial Out (PISO)**
4. **Parallel In Parallel Out (PIPO)**

Serial In Serial Out (SISO Mode)

⇨ We want to Store "1111" in this Shift Register.



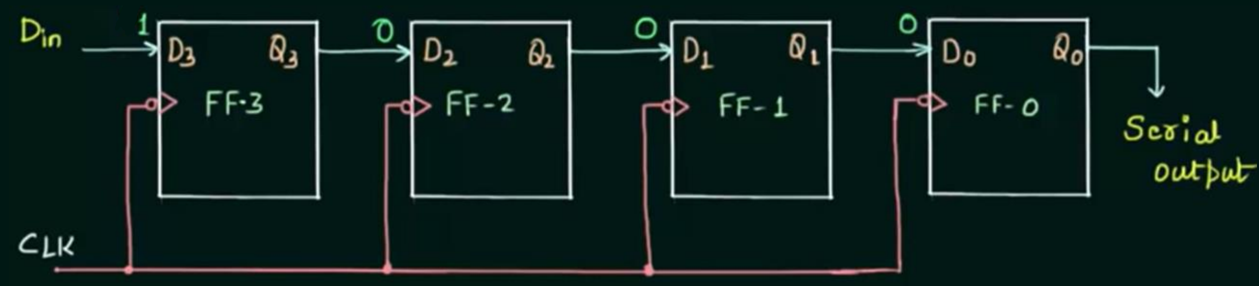
Step 1: Identify the **LSB** and **MSB.**

Step 2: Draw the D. FF TT:

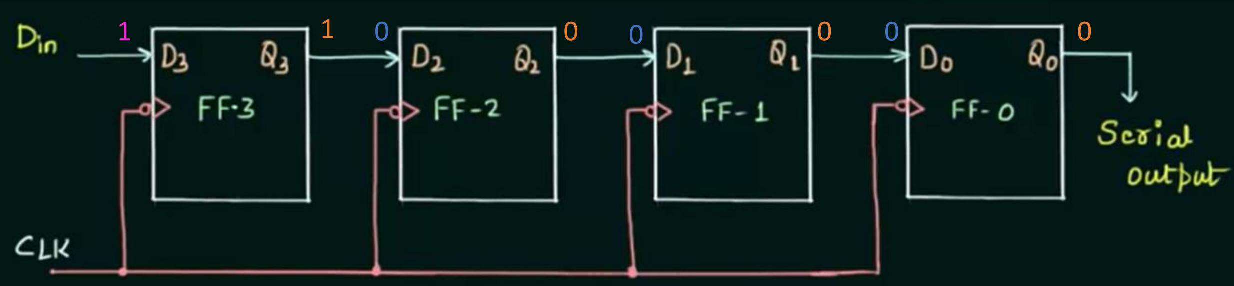
CLK	D	Q_{n+1}
0	**X**	**Q_n**
1	**O**	**0**
1	**1**	**1**

Step 3: Initially

CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓				
↓				
↓				
↓				

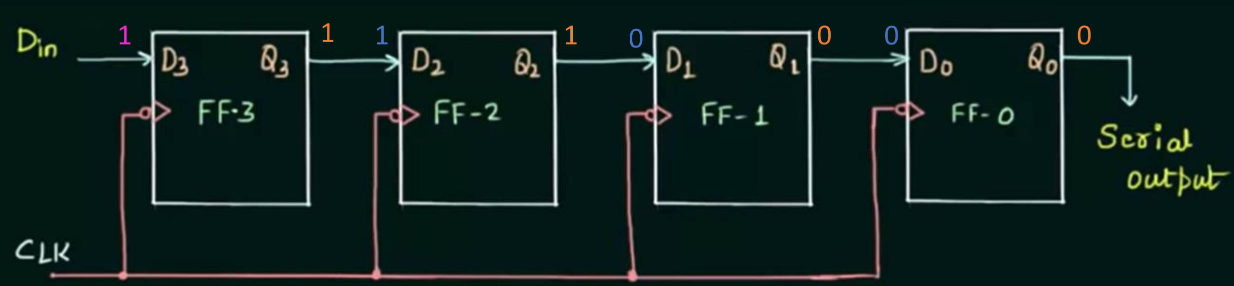


Step 4: for 1st falling edge:



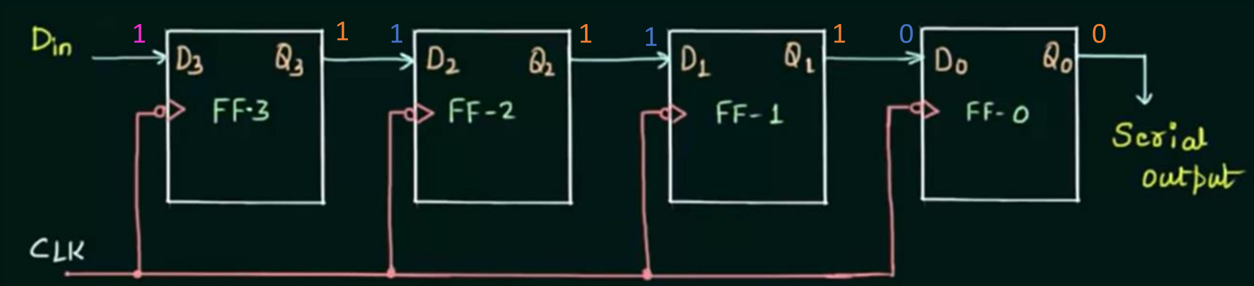
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓				
↓				
↓				

Step 5: for 2nd falling edge:



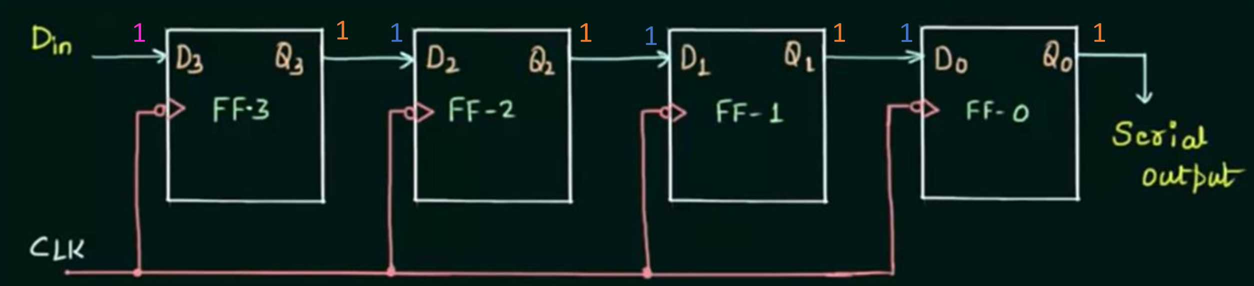
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓				
↓				

Step 5: for 3rd falling edge:



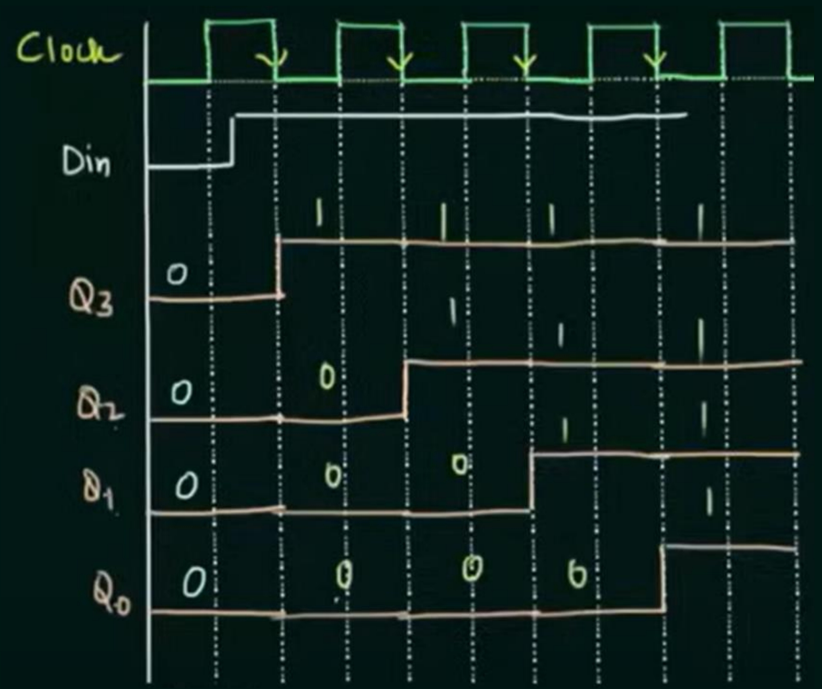
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓				

Step 6: for 4th falling edge:



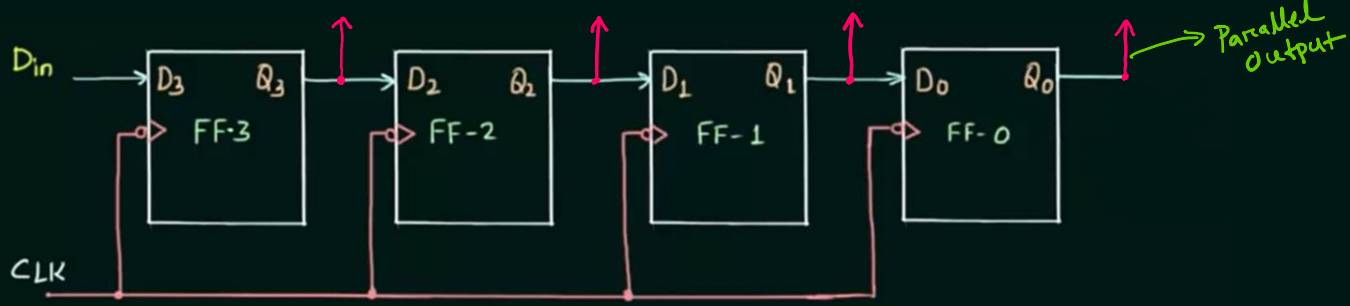
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

Analyses the clock Diagram



Serial In Parallel Out (SIPO Mode)

⇨ We want to Store "1111" in this Shift Register.



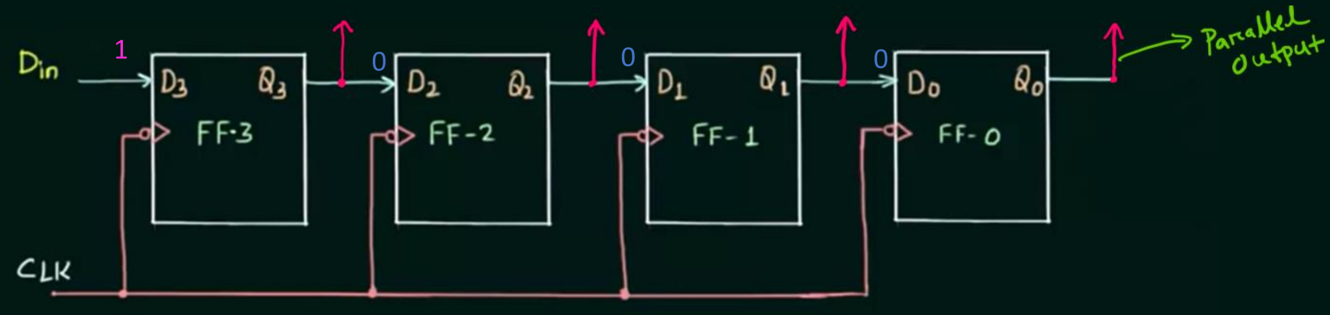
Step 1: Identify the **LSB** and **MSB.**

Step 2: Draw the D. FF TT:

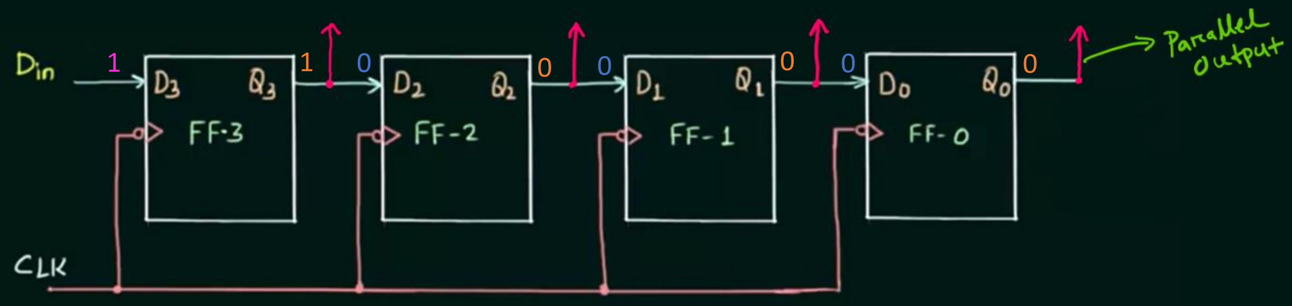
CLK	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

Step 3: Initially

CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓				
↓				
↓				
↓				

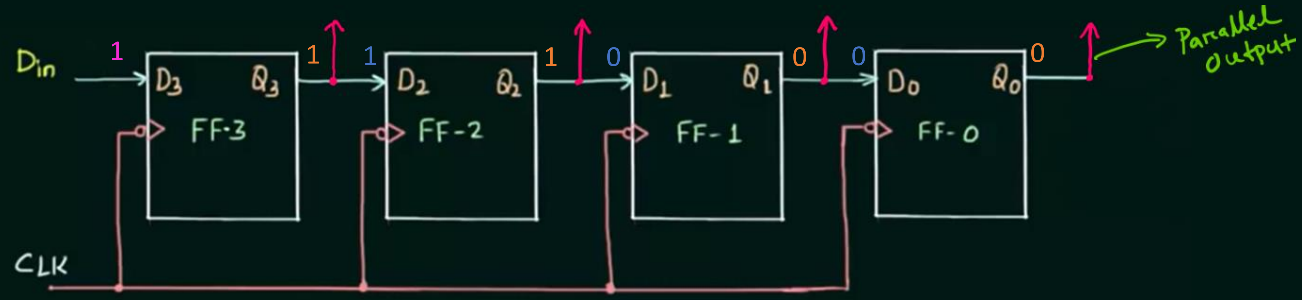


Step 4: for 1st falling edge:



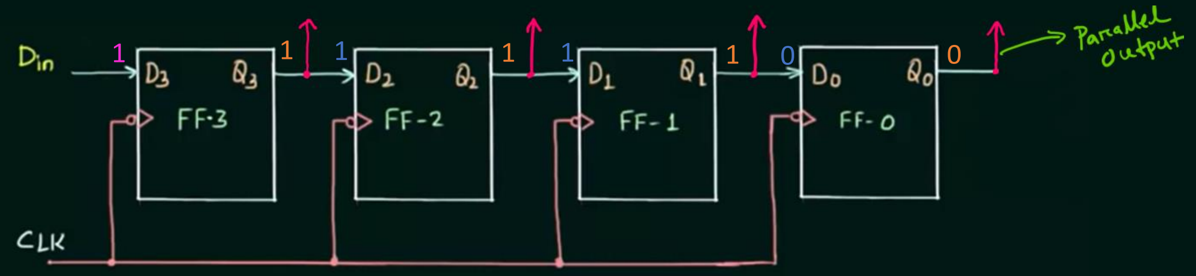
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓				
↓				
↓				

Step 5: for 2nd falling edge:



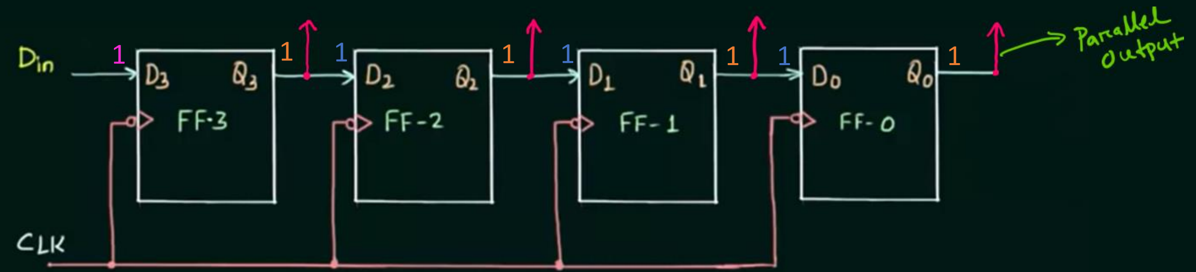
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓				
↓				

Step 5: for 3rd falling edge:



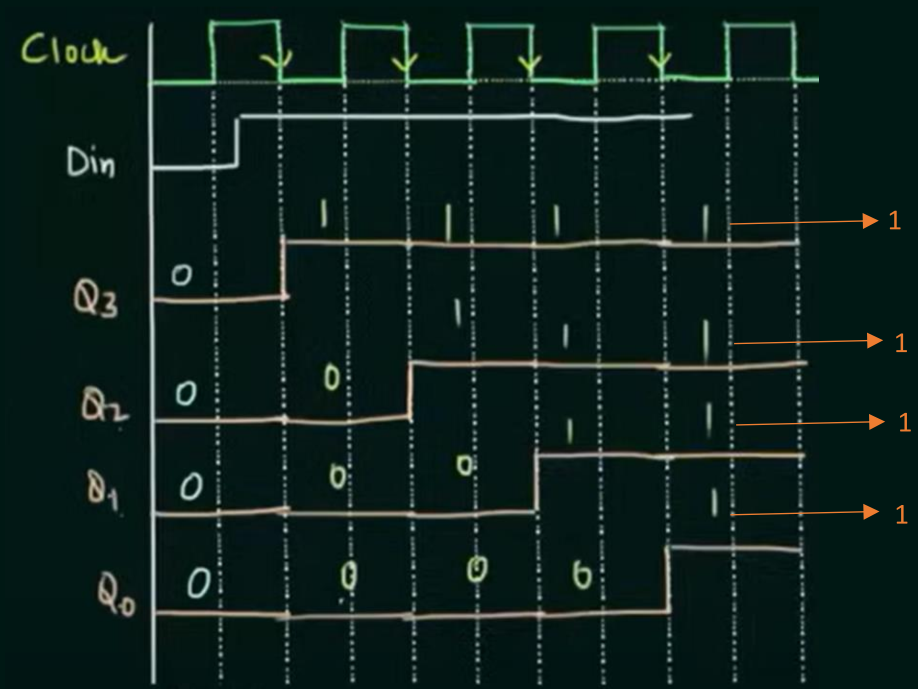
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓				

Step 6: for 4th falling edge:



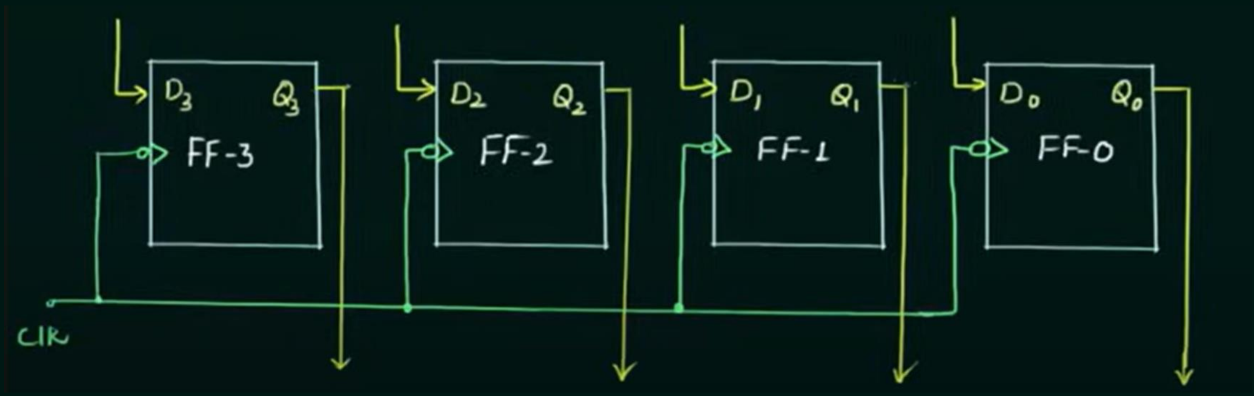
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

Analyses the clock Diagram



Parallel In Parallel Out (PIPO Mode)

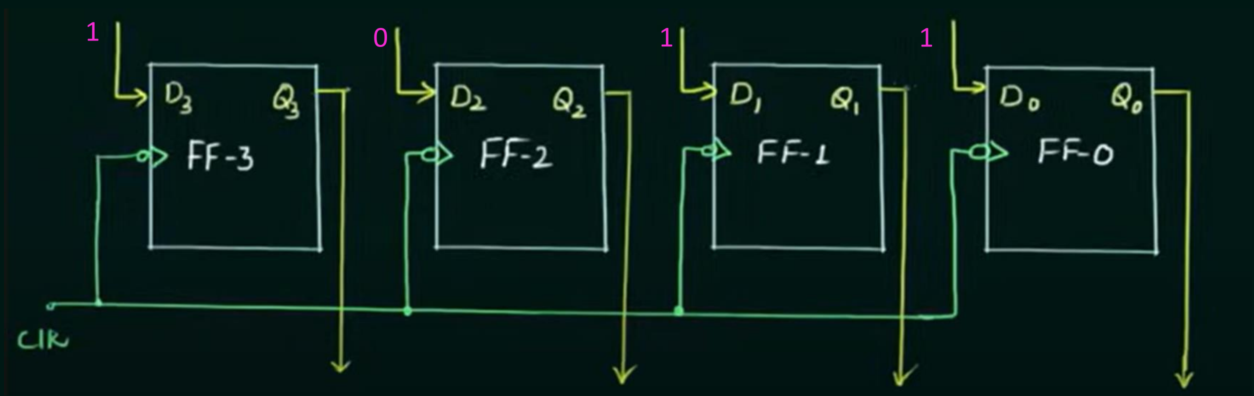
⇨ **Example 1: We want to Store "1011" in this Shift Register.**



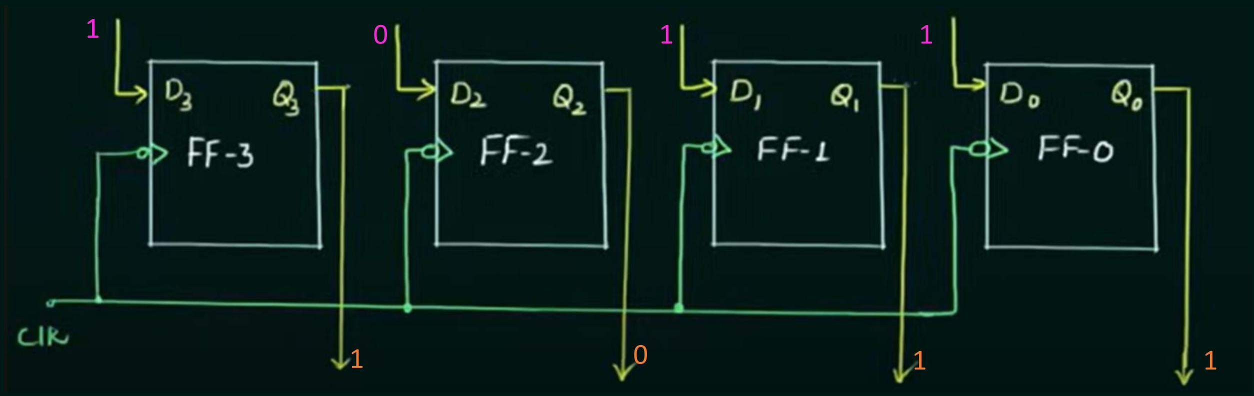
Step 1: Draw the D. FF TT:

CLK	D	Q_{n+1}
0	**X**	Q_n
1	**O**	**0**
1	**1**	**1**

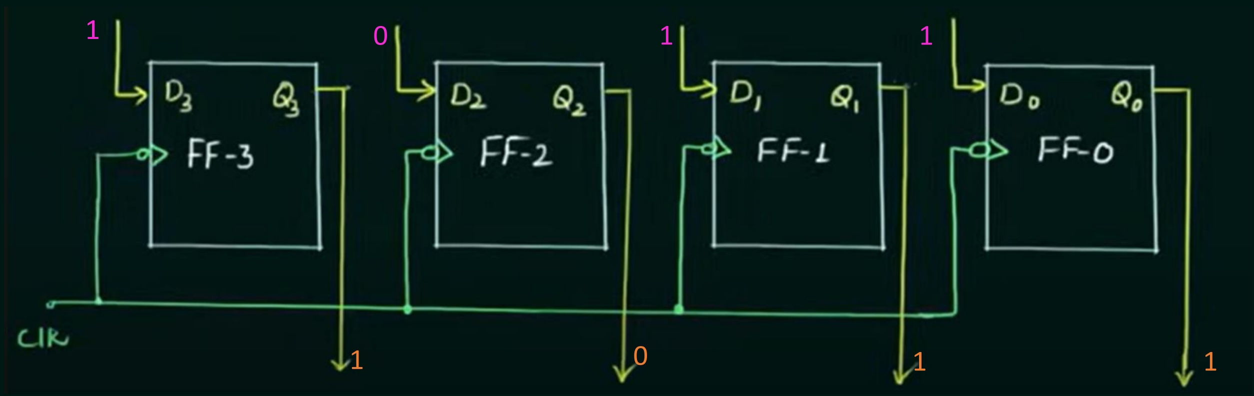
Step 2: Insert the Parallel Inputs:



Step 3: When CLK is "logic 1"

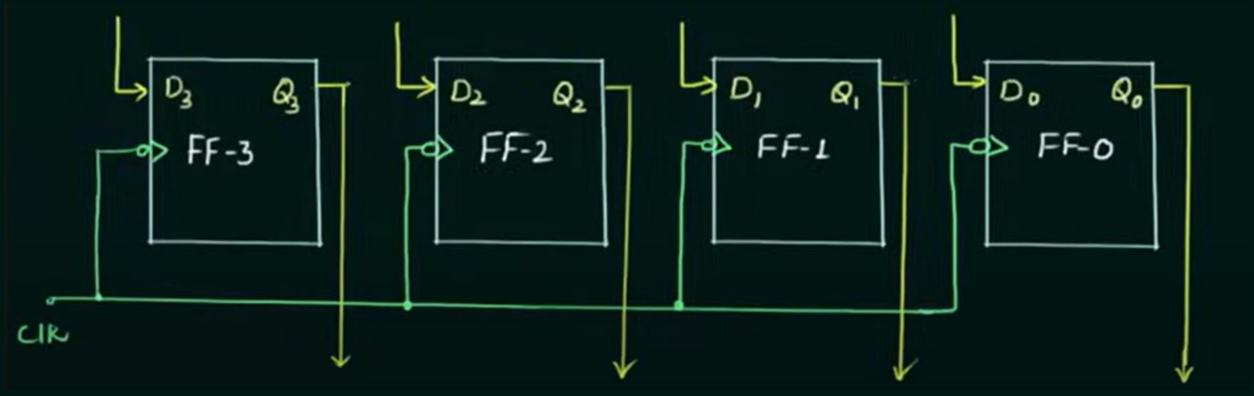


Step 4: When CLK is "logic 0"



⇨ **Its Hold the Same value since if CLK = 0 then input = Don't care Output is Memory.**

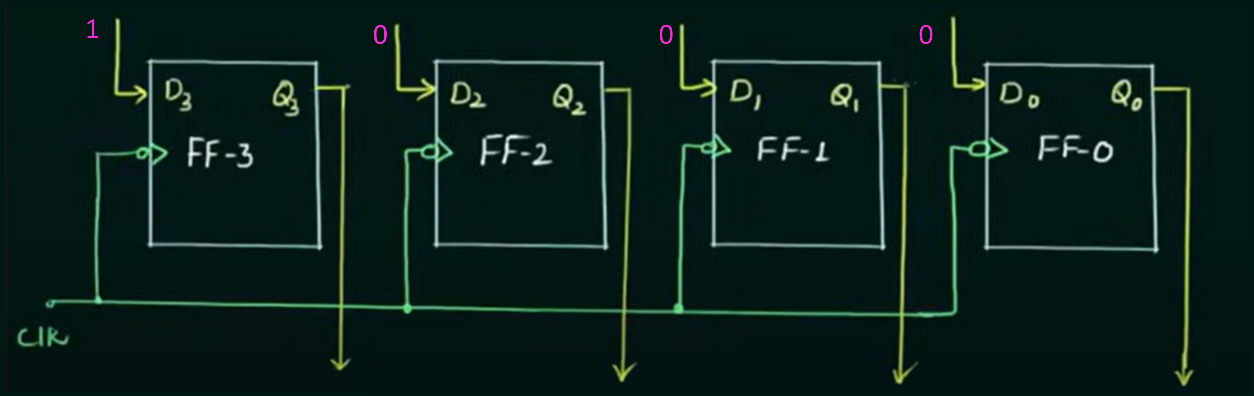
⇨ **Example 2: We want to Store "1000" in this Shift Register.**



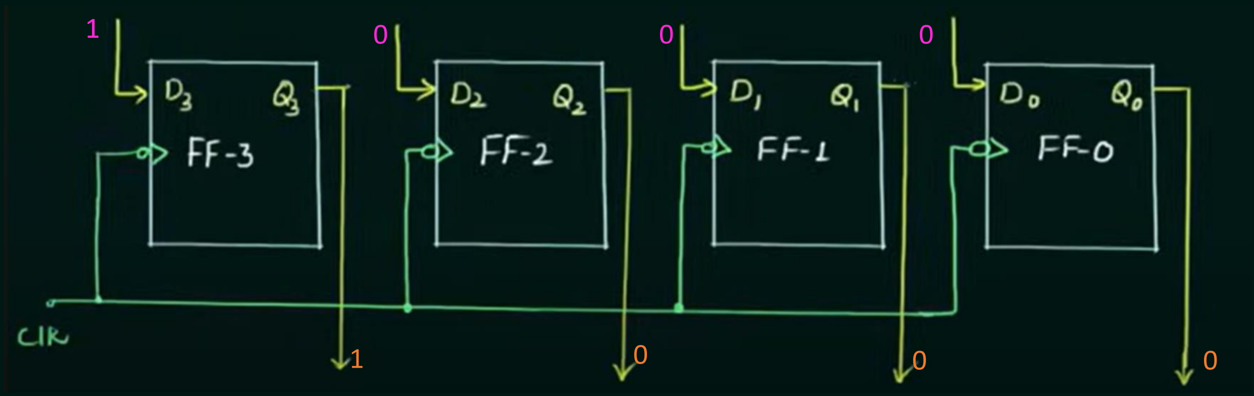
Step 1: Draw the D. FF TT:

CLK	D	Q_{n+1}
0	**X**	$\mathbf{Q_n}$
1	**O**	**0**
1	**1**	**1**

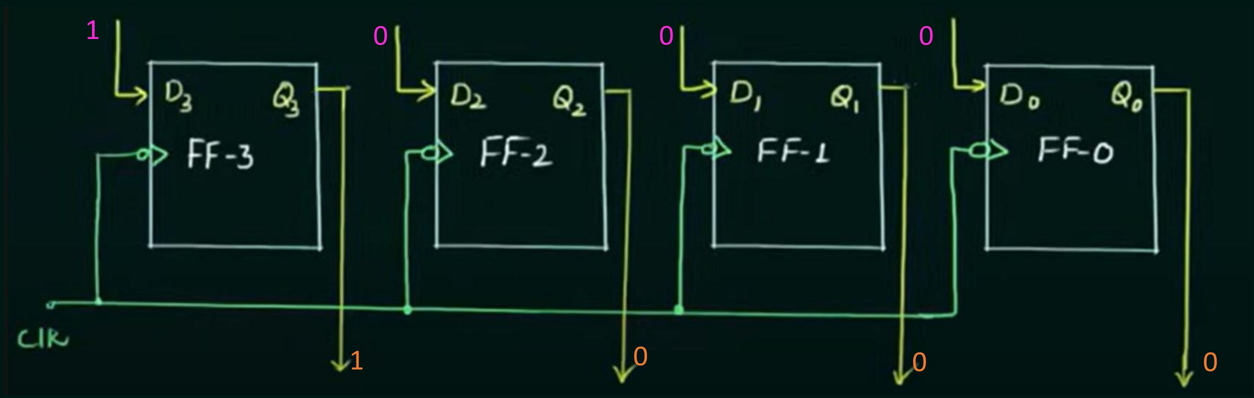
Step 2: Insert the Parallel Inputs:



Step 3: When CLK is “logic 1”



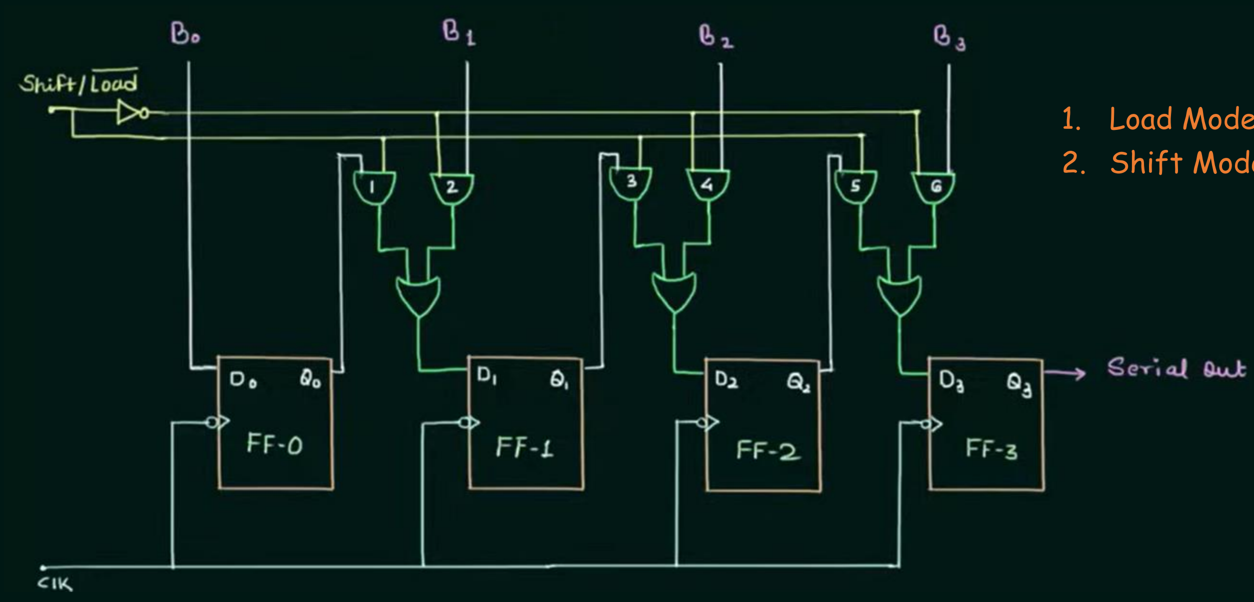
Step 4: When CLK is “logic 0”



⇨ **Its Hold the Same value since if CLK = 0 then input = Don’t care Output is Memory.**

Parallel In Serial Out (PISO Mode)

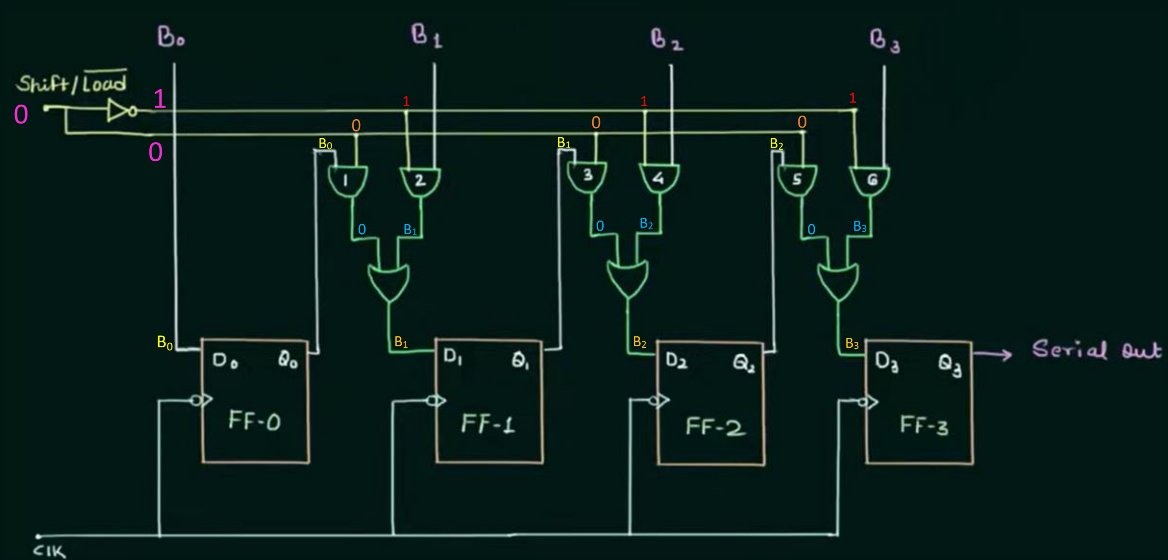
⇨ **The circuit Diagram is:**



1. Load Mode
2. Shift Mode

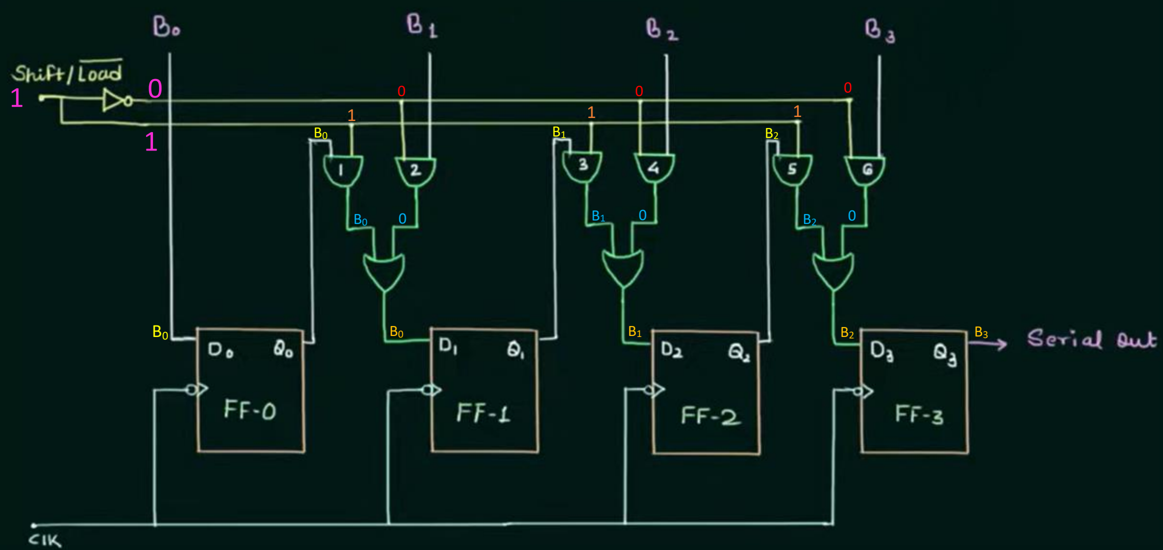
1. How is Load Mode Achieved?

⇨ **For that Shift/Load' is = 0**



2. How is Shift Mode Achieved?

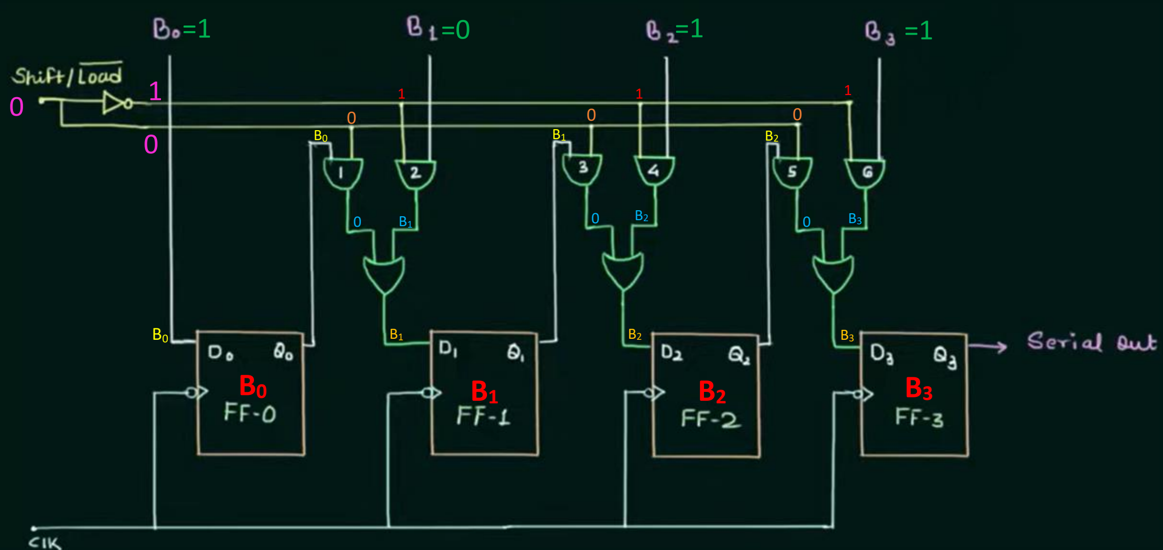
⇨ For that Shift/Load' is = 1



⇨ We want to Store "1011" in this Shift Register.

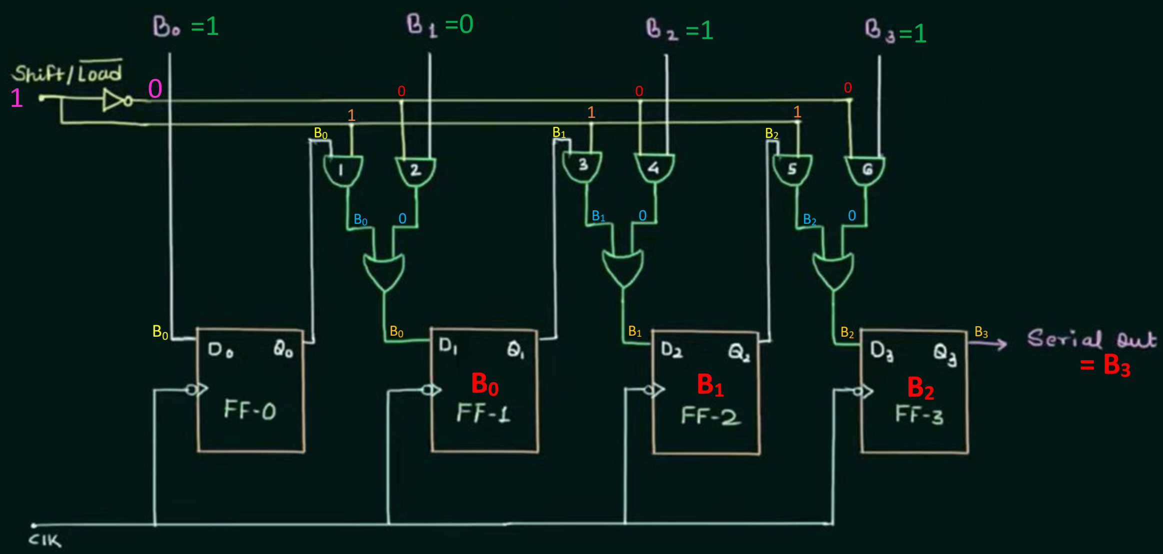
Step 1: 1st Achieved Load Mode?

⇨ For that Shift/Load' is = 0



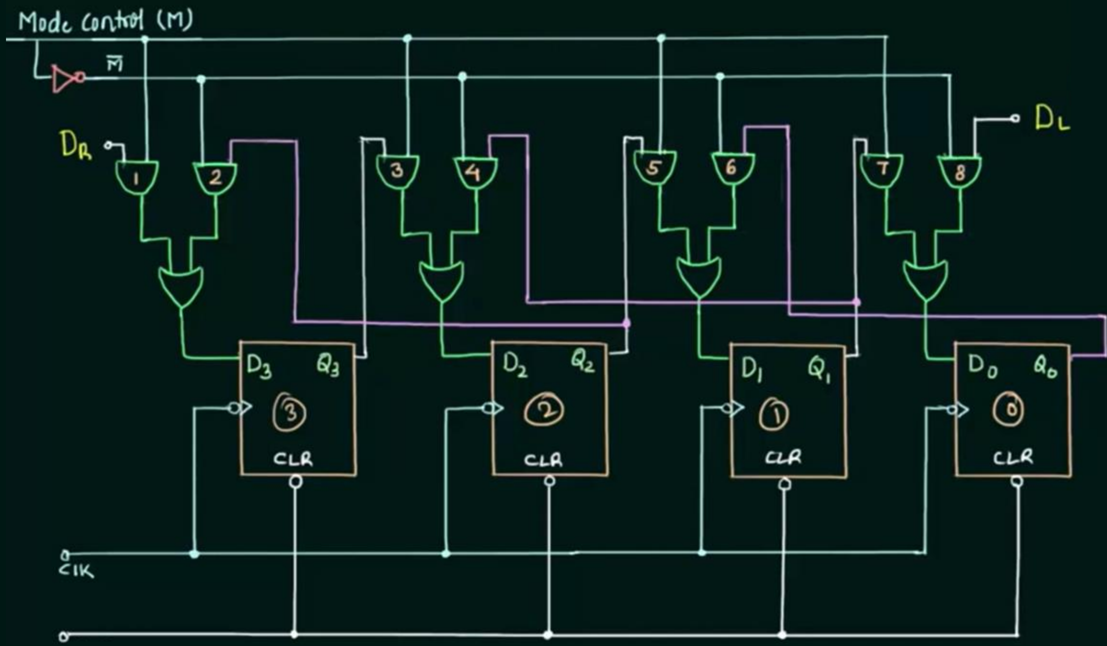
⇨ **Step 2: 2^{nd} Achieved Shift Mode?**

⇨ **For that Shift/Load' is = 1**

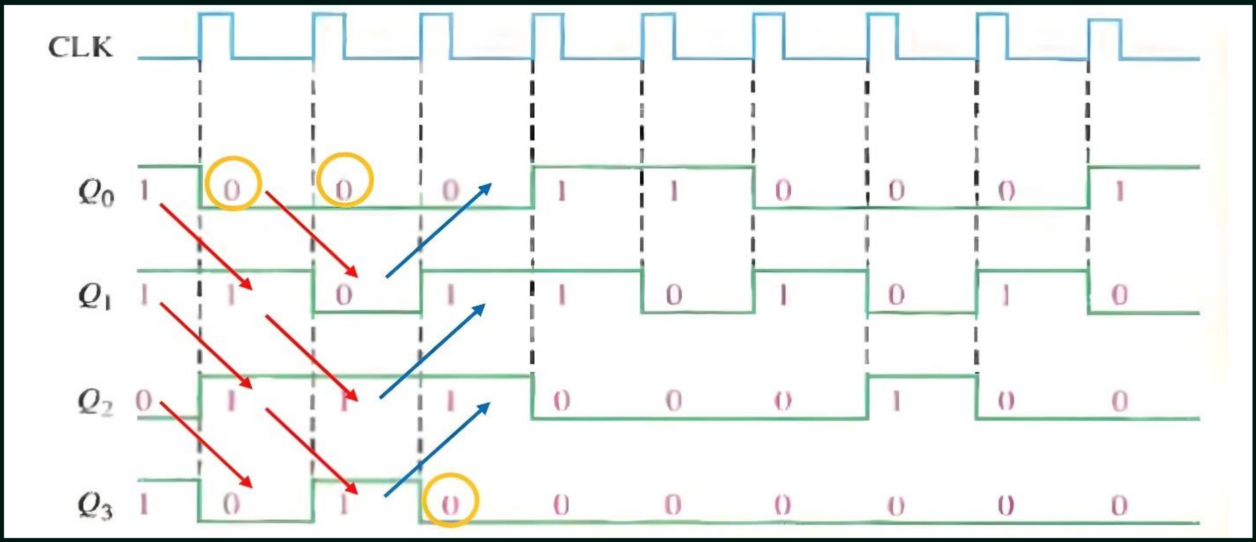


Bidirectional Shift Register

⇨ **The circuit Diagram is:**



Example 1: Determine the state of the shift register after each clock pulse for the given RIGHT/LEFT control input waveform. Assume Q_0=1, Q_1=1, Q_2=0, and Q_3=1 and the serial data input line is LOW.



Shift Register Counters

- A shift register counter is a shift register with its serial output connected to its serial input, creating a specific sequence of states.

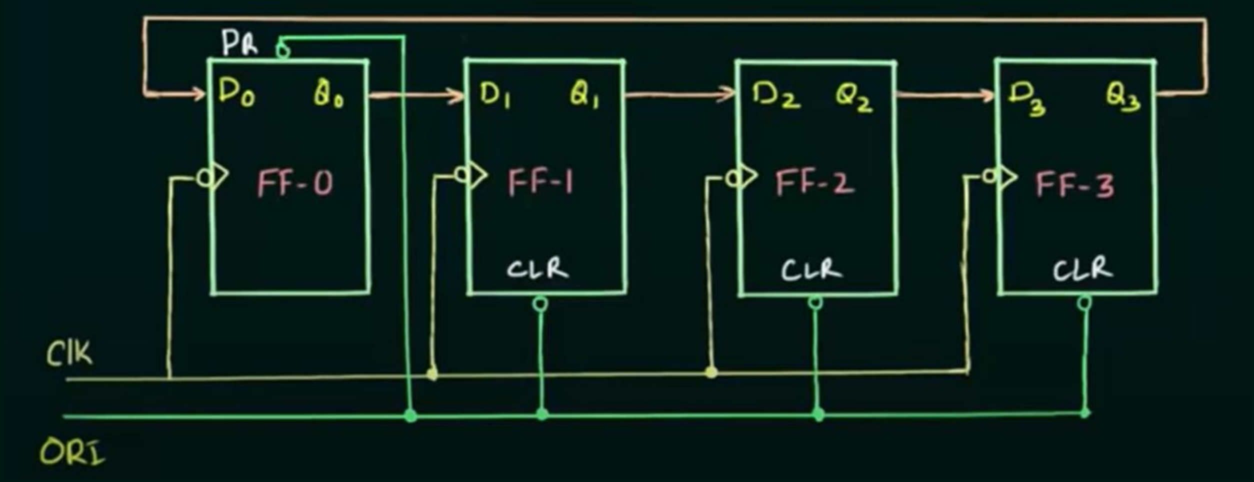
There are 2 types:

1. **Ring Counter**
2. **Johnson Counters**

RING COUNTER

Ring Counter

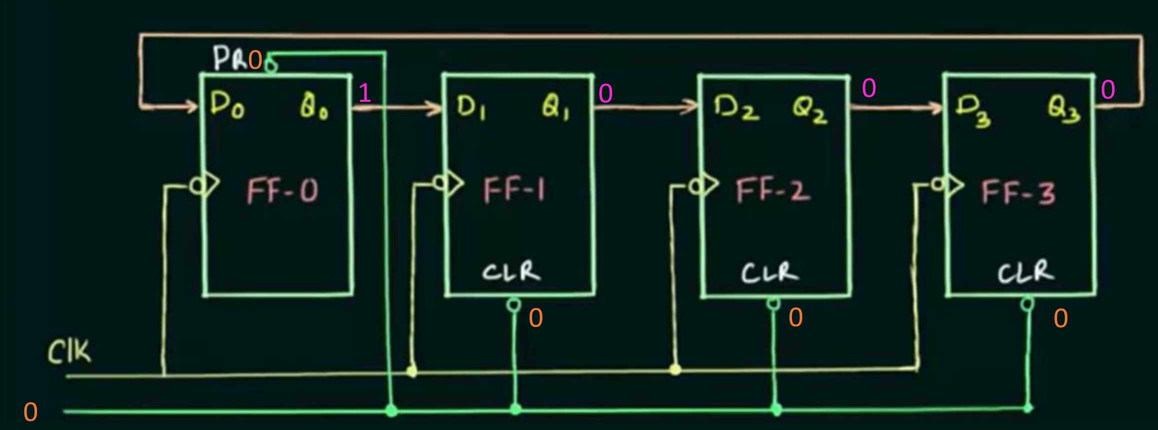
>> Ring counter is a typical application of shift register
>> The only change is the otput of last ff is connected to the input of first ff.



Note: NO of State = No. of flip-flops Used in the circuit.

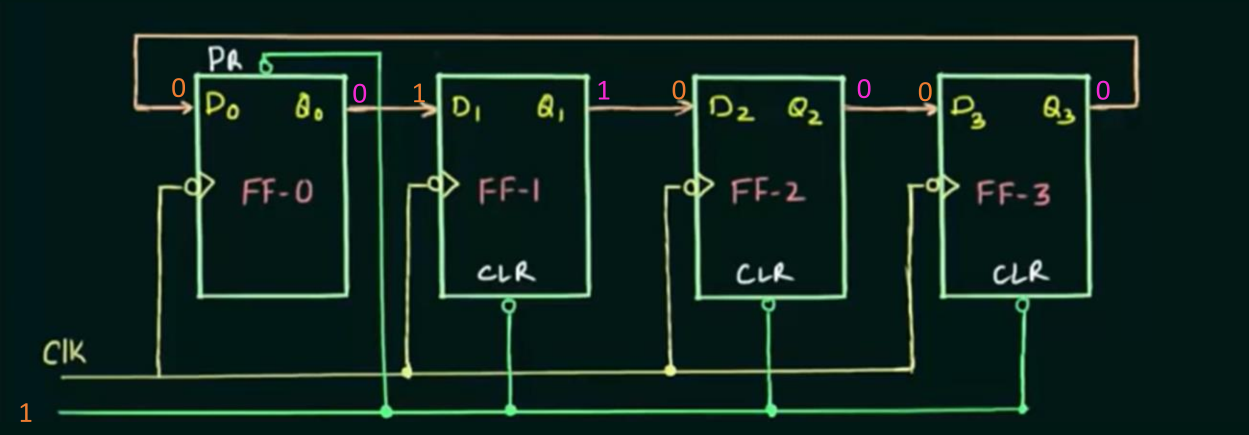
4-bit Ring counter

Step 1: Assume that Initially ORI=0



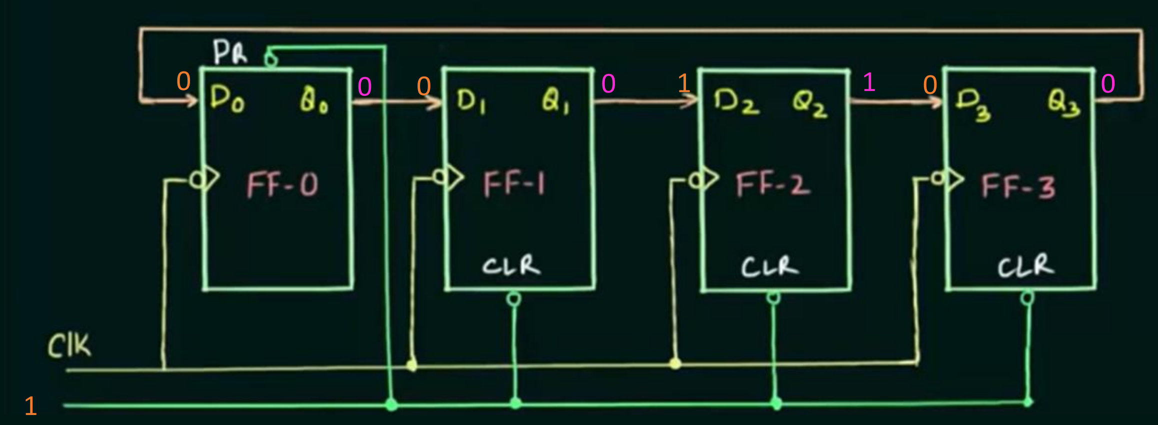
ORI	CLK	Q_0	Q_1	Q_2	Q_3
0	X	1	0	0	0
1	↓				
1	↓				
1	↓				
1	↓				

Step 1: **Now** for 1^{st} falling edge & **rest of the time that ORI=1:**



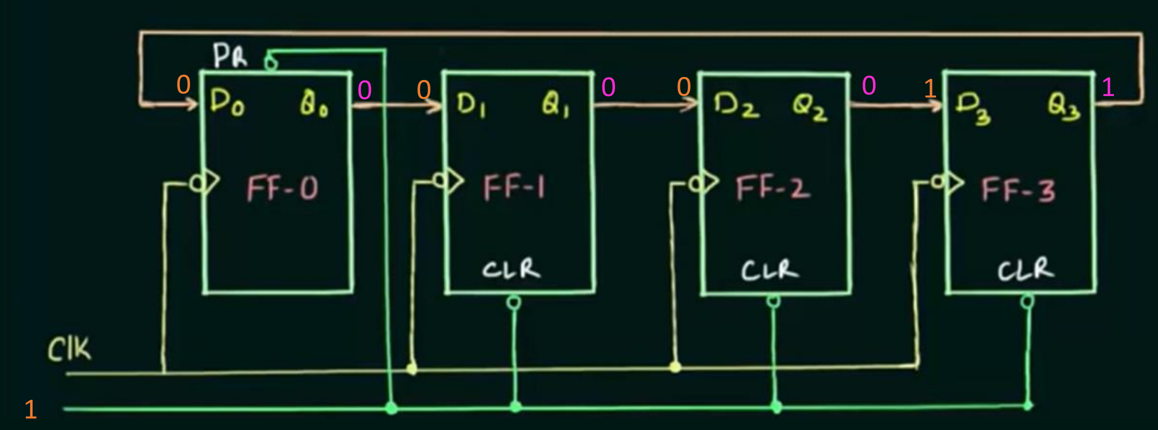
ORI	CLK	Q_0	Q_1	Q_2	Q_3
0	X	1	0	0	0
1	↓	0	1	0	0
1	↓				
1	↓				
1	↓				

Step 2: **Now** for 2nd falling edge:



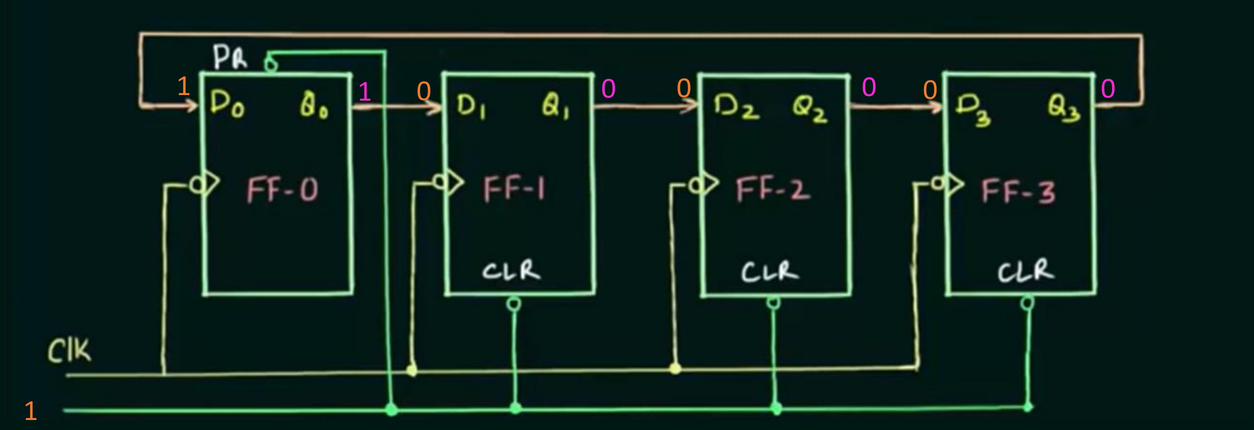
ORI	CLK	Q_0	Q_1	Q_2	Q_3
0	X	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓				
1	↓				

Step 3: **Now** for 3rd falling edge:



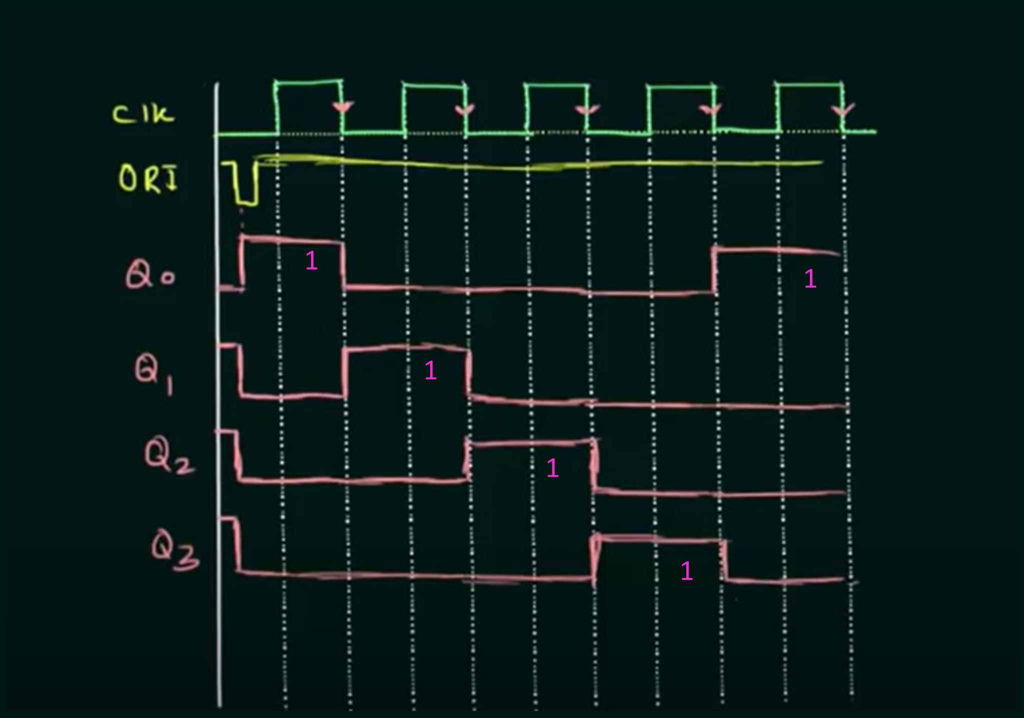
ORI	CLK	Q_0	Q_1	Q_2	Q_3
0	X	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓				

Step 4: **Now** for 4th falling edge:



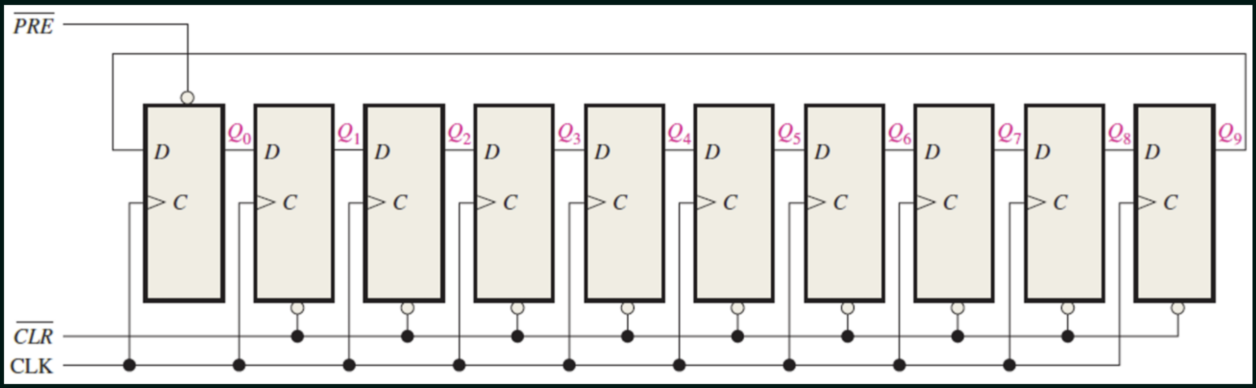
ORI	CLK	Q_0	Q_1	Q_2	Q_3
0	X	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

Analyses the clock Diagram



10-bit ring counter

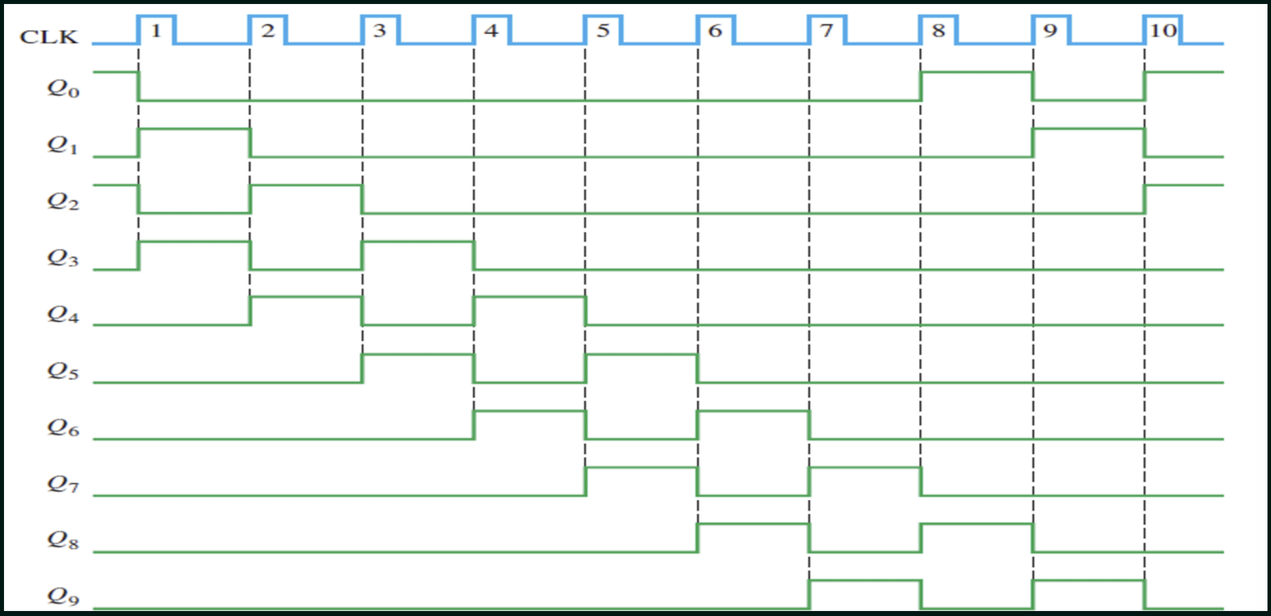
⇨ **The circuit Diagram is:**



⇨ **The 10-bit Ring Counter Sequence is:**

Clock Pulse	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

Analyses the clock Diagram



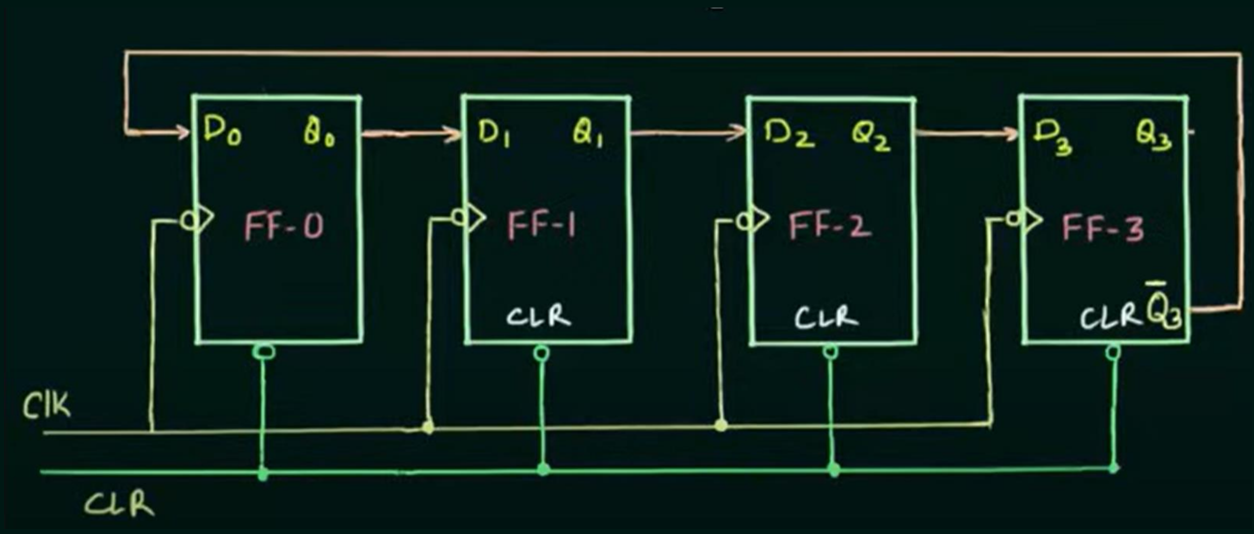
JOHNSON COUNTER

Johnson Counter

>> Its also known as Twisted/Switch tail Ring Counter.

>> In a Johnson counter, the complement of the last flip-flop's output connects to the first flip-flop's input, generating a unique sequence. A Johnson counter has 2N states for N flip-flops (e.g., 4-bit = 8 states, 5-bit = 10 states).

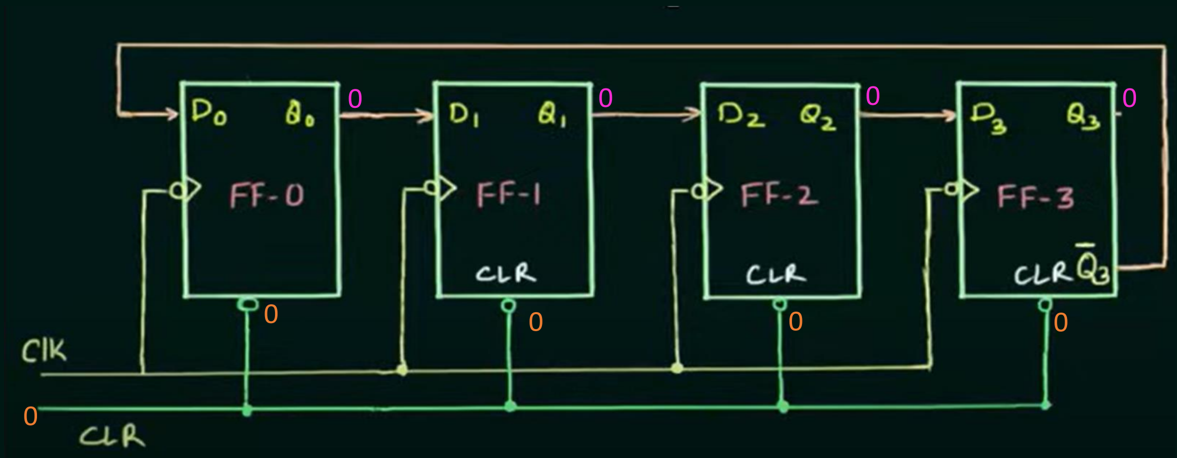
- ⇨ **The circuit Diagram is:**



Note: NO of State = 2 * No. of flip-flops Used in the circuit.

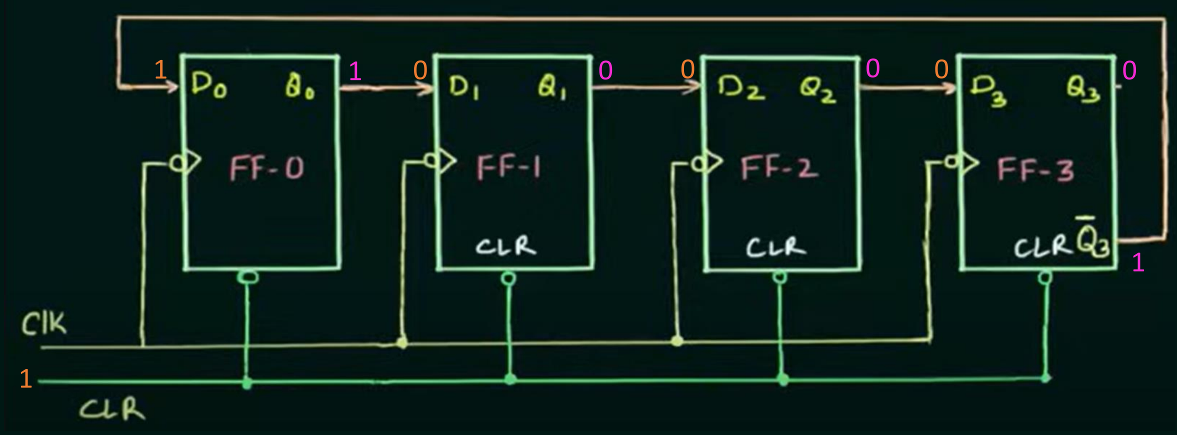
4-bit Johnson counter

Step 1: Assume that Initially CLR=0



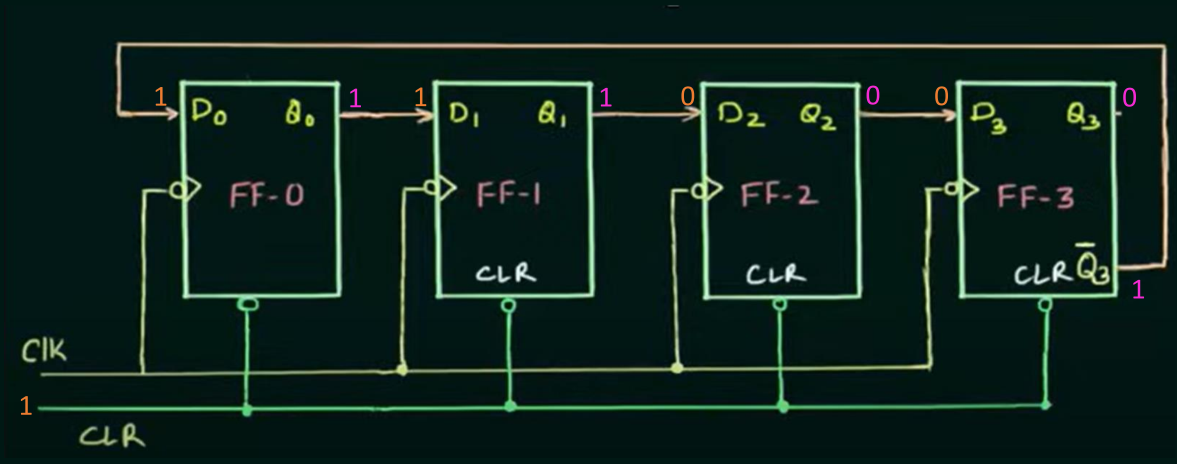
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓				
1	↓				
1	↓				
1	↓				

Step 1: **Now** for 1st falling edge & **rest of the time that CLR=1:**



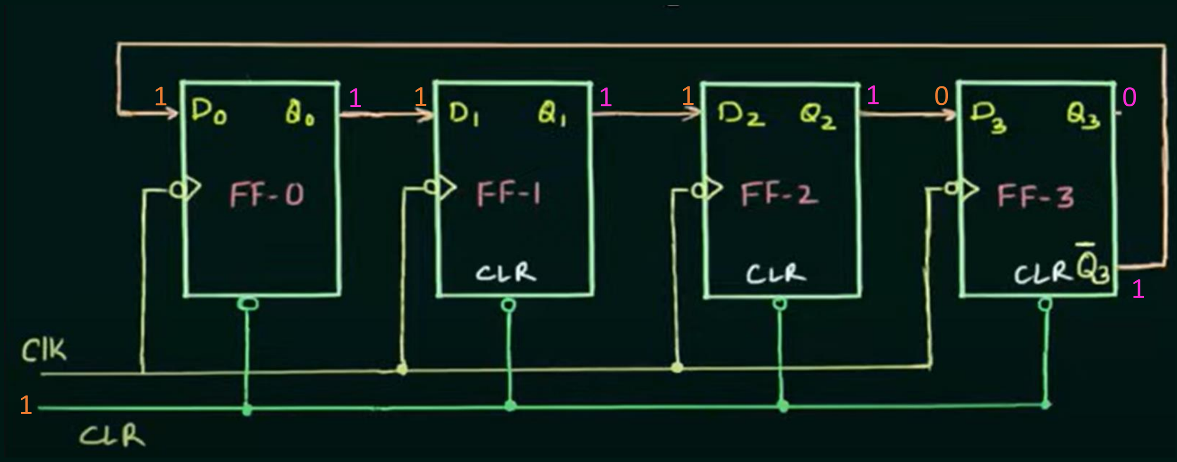
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓				
1	↓				
1	↓				

Step 2: Now for 2nd falling edge:



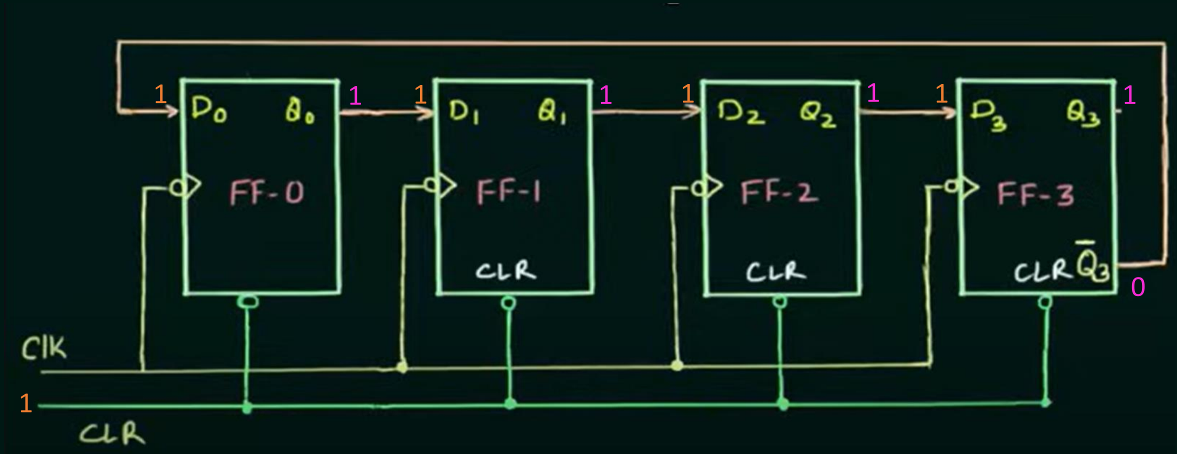
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓				
1	↓				

Step 3: Now for 3rd falling edge:



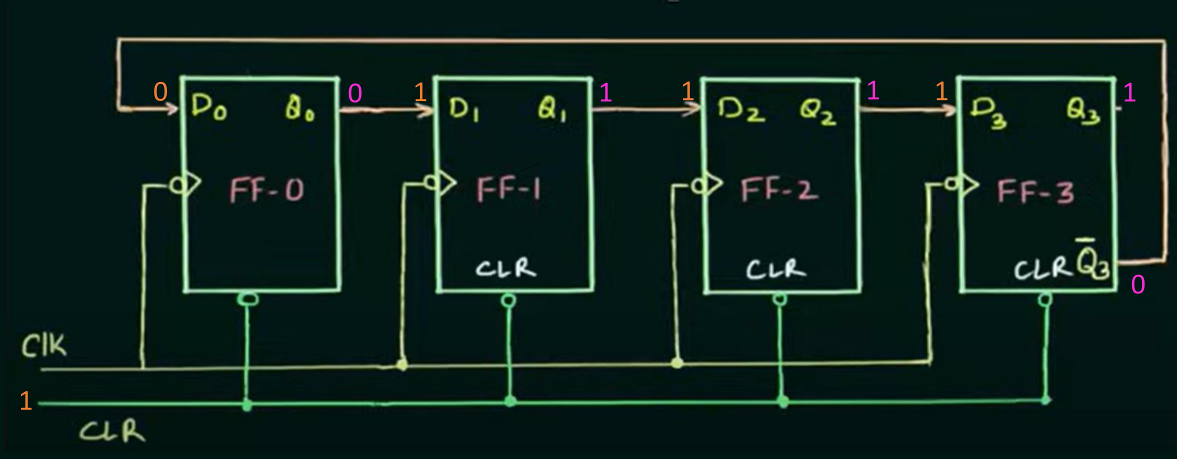
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓				

Step 4: Now for 4th falling edge:



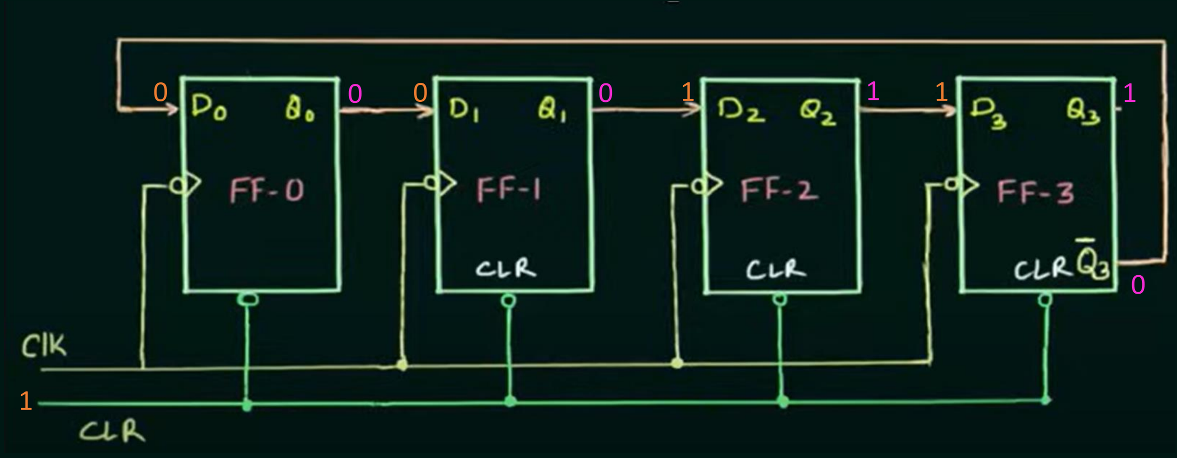
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1

Step 5: **Now** for 5th falling edge:



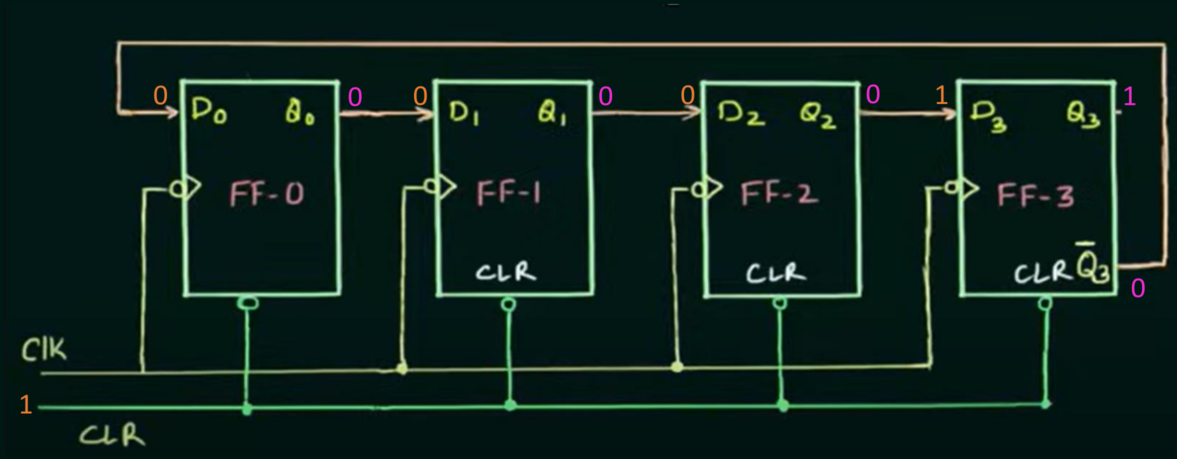
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1

Step 6: **Now** for 6th falling edge:



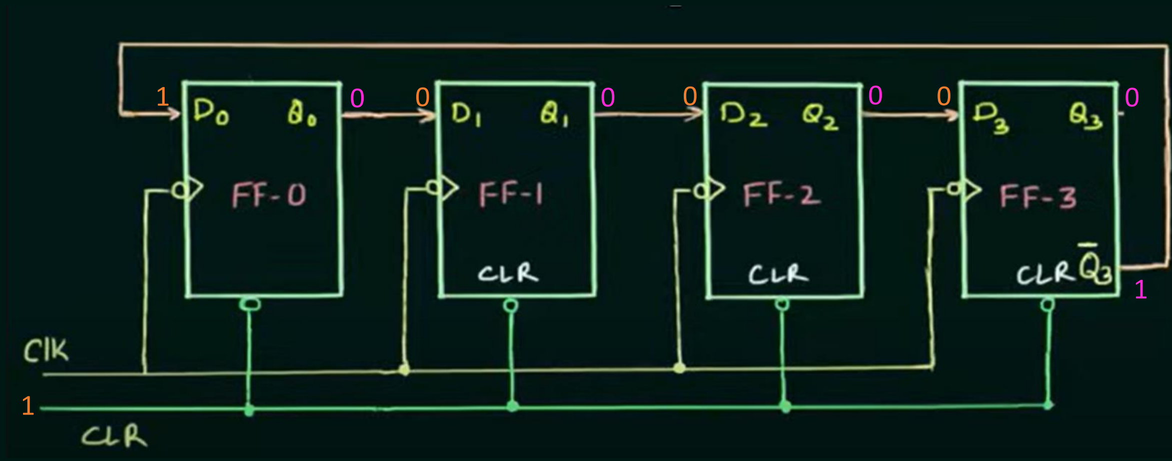
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1

Step 7: Now for 7th falling edge:



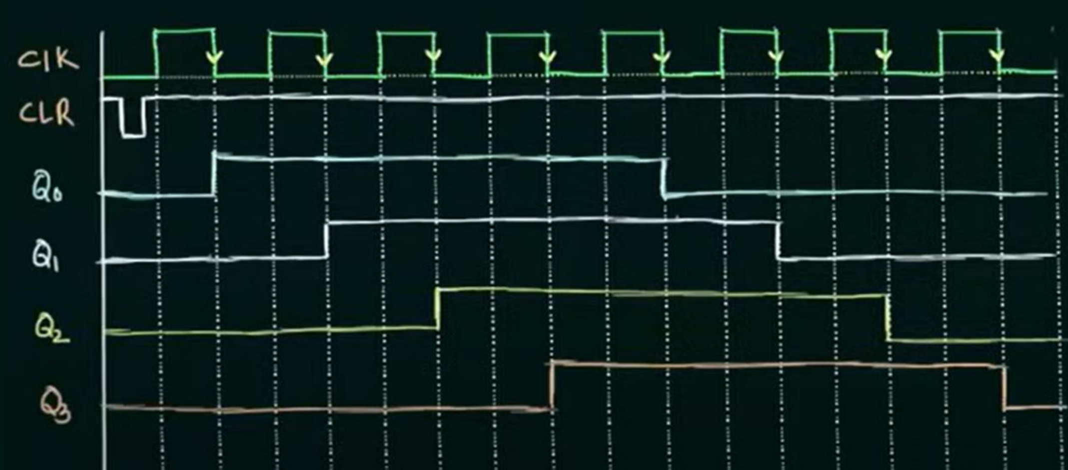
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1
1	↓	0	0	0	1

Step 8: Now for 8^{th} falling edge:



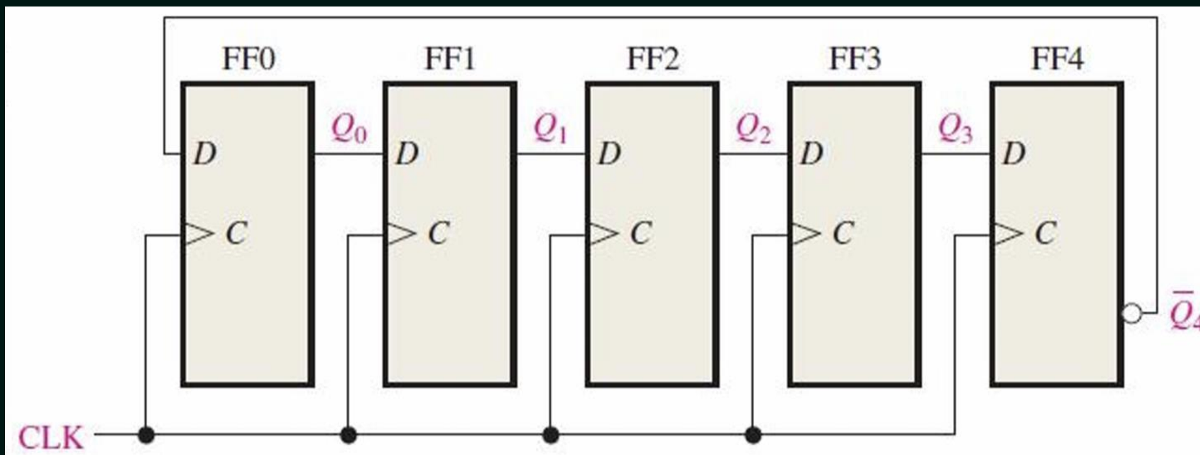
CLR	CLK	Q_0	Q_1	Q_2	Q_3
0	X	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1
1	↓	0	0	0	0

Analyses the clock Diagram



5-bit Johnson counter

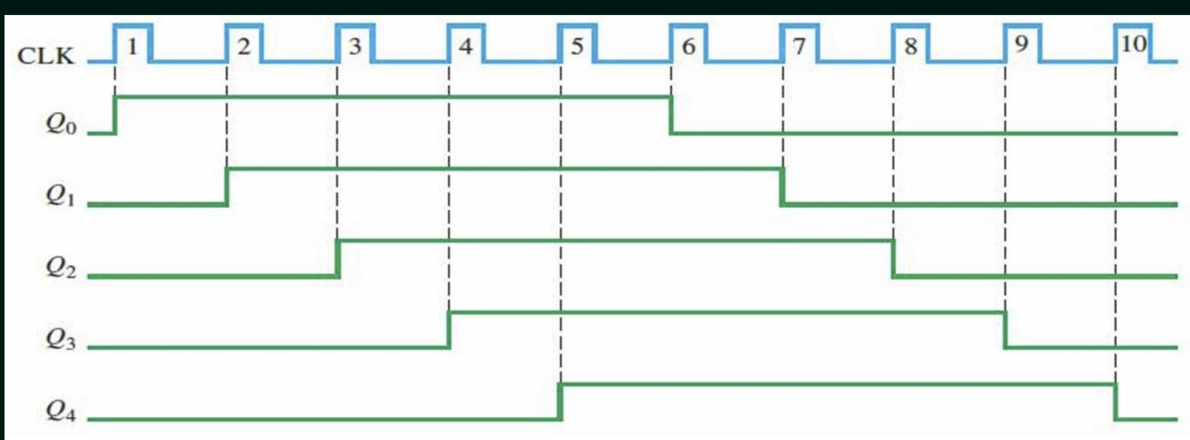
⇨ The circuit Diagram is:



⇨ The 5-bit Johnson_Counter Sequence is:

Clock Pulse	Q_0	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	1	1	1	1	0
5	1	1	1	1	1
6	0	1	1	1	1
7	0	0	1	1	1
8	0	0	0	1	1
9	0	0	0	0	1

Analyses the clock Diagram



DIGITAL SYSTEM DESIGN WITH FINITE STATE MACHINES (FSM)

Digital System Design with Finite State Machines (FSM)

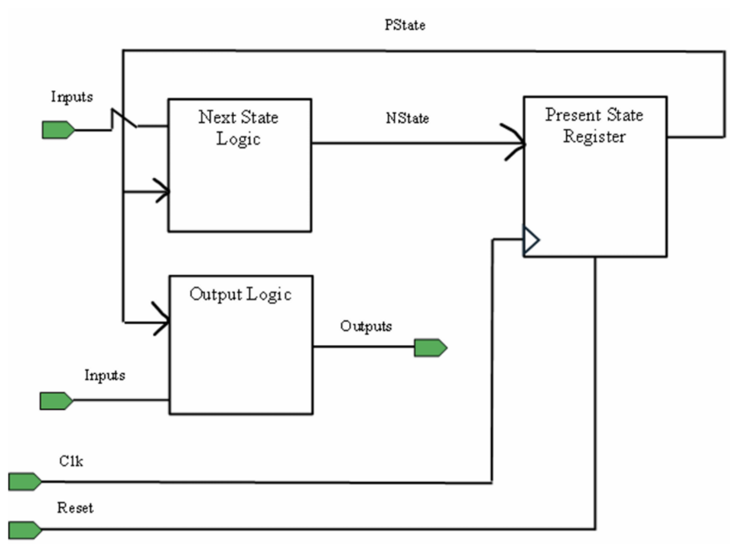
➔ **FSM/FSA**: Automata with a finite number of states; executes a predetermined sequence of operations.

- Self-regulating, automatic machine without external assistance.
- Computational model used in hardware/software for designing programs and circuits.
- Terminology:
  - "Finite State Automata (FSA)" in Mathematics/Computer Science.
  - "Finite State Machines (FSM)" in Electronics/Engineering.
- **Applications:**
  - Pattern recognition.
  - Modeling digital circuits.
  - Designing algorithms and programs.
  - Building compilers/interpreters.
  - Language models and machine learning.
- **Purpose:** Automate computational tasks.
- **FSM described by:**
  - Finite states.
  - Finite inputs.
  - State transitions.
  - Initial state.
  - Final state.
- In digital logic: FSMs design control circuits and represent sequential logic circuits.

FSM Architecture

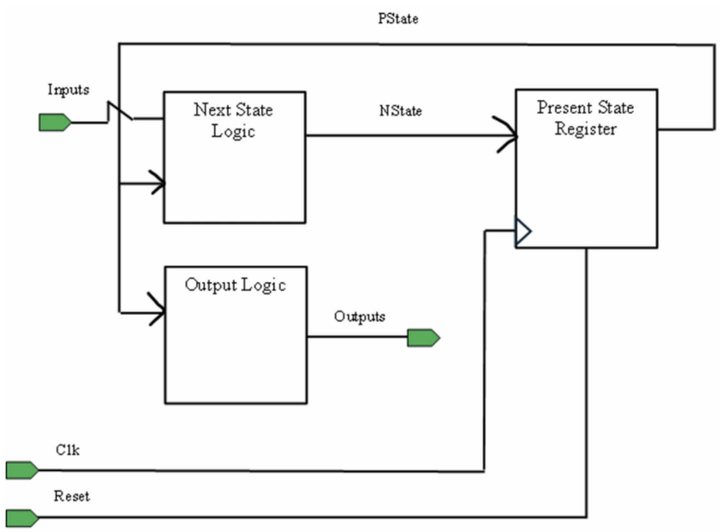
➔**FSM Architecture Components**:

- **Present State Register**: Sequential logic.
- **Next State Logic**: Combinational logic.
- **Output Logic**: Combinational logic.



Moore vs Mealy FSMs

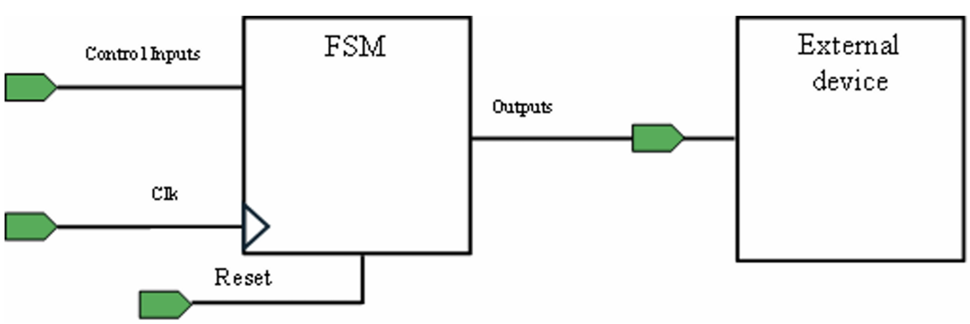
- **Moore FSMs**:
  - Outputs depend only on the present state.
  - Produce only Moore outputs.
  - Architecture focuses on state-based output generation.
- **Mealy FSMs**:
  - Produce one or more Mealy outputs.
  - Outputs depend on both the present state and inputs.
  - Architecture includes input-based output generation for faster response.



Control-dominated FSMs vs Data-dominated FSMs

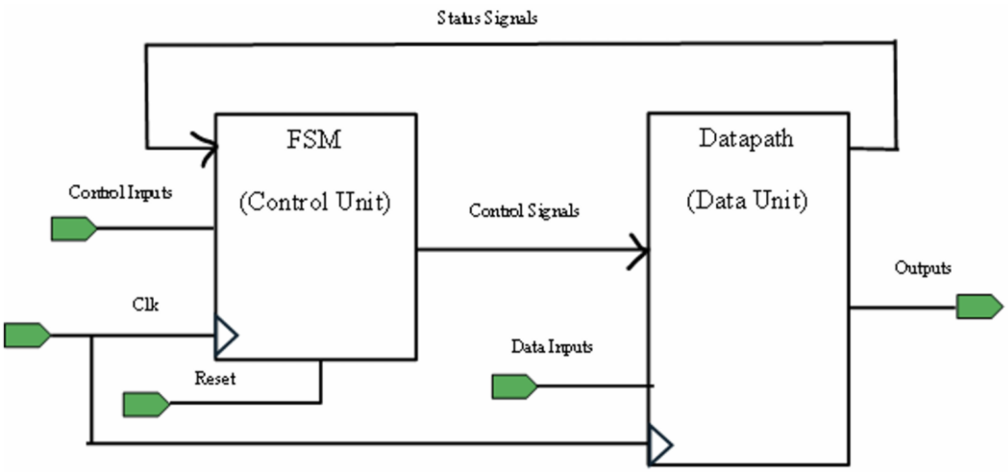
⇨ **Control-Dominated Designs Using FSMs**:

- Examples: Signal generators (e.g., traffic light controllers) and stepper motor controllers.
- FSMs manage and control the operation of external devices like LEDs or stepper motors.



➪ **Data-Dominated Designs Using FSMs**:

- Examples: Microprocessors.
- FSMs control logical and arithmetic operations for computations.
- **Controlled Elements**:
  - Arithmetic and Logic Units (ALU).
  - Storage devices (registers).
  - Data steering circuits (multiplexers and demultiplexers).
- **Control Signals**:
  - ALU_Control signals, select signals, load/enable signals.
- FSM reads device output states as **status signals** for proper control.
- Computations occur in the **datapath** (data unit), while the FSM functions as the **control unit** to guide and sequence operations correctly.



Example:

➪ **Design an FSM that:**

- Starts in an idle state, generating an output "Ready" and evaluating the input "Go."
- If "Go" is low, it stays in the idle state.
- If "Go" is high, it generates three outputs (Red, Green, Yellow) consecutively for 3 seconds, 3 seconds, and 1 second, respectively.
- **The clock frequency is 1 Hz.**
- Use binary state encoding.

Show:

1. **FSM architecture**
2. **State chart**
3. **State table**
4. **Logic equations for next states and outputs.**

➔Solution:

⇨ **Is given:**

$f = 1Hz$

$\therefore T = \frac{1}{f} = 1s$ [that means each State is 1 sec long]

FSM Design:

- **Objective**: The system needs to output "Ready," "Red," "Green," and "Yellow" in a sequence based on a 1 Hz clock.

States & Outputs:

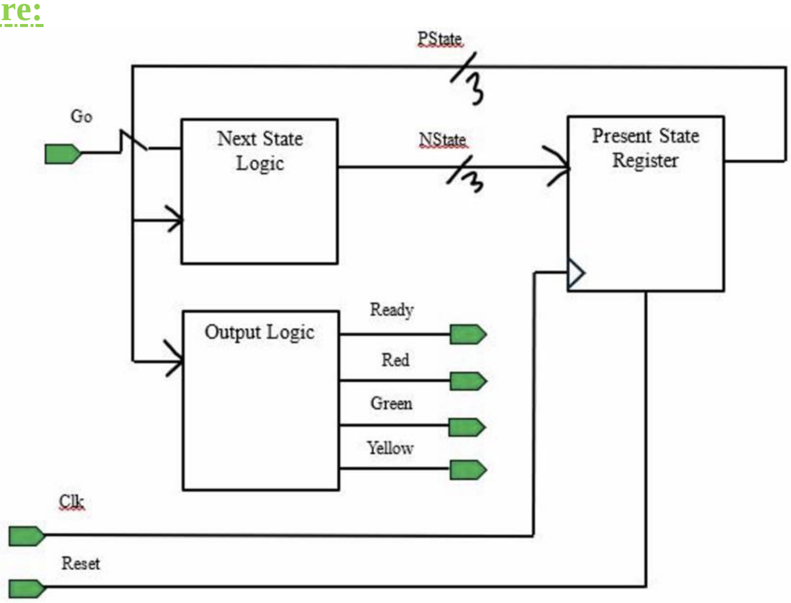
1. **Idle State (State 0)**: Outputs "Ready" (only for 1 clock cycle).
2. **Red States (State 1, 2, 3)**: Outputs "Red" for 3 clock cycles.
3. **Green States (State 4, 5, 6)**: Outputs "Green" for 3 clock cycles.
4. **Yellow State (State 7)**: Outputs "Yellow" for 1 clock cycle.

Total States:

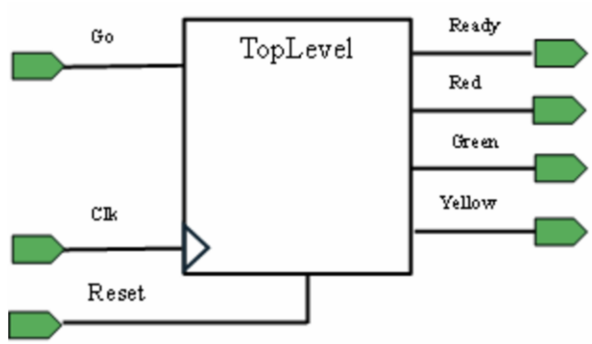
- The system **needs 8 states in total** (000 to 111 in binary), since **it has 8 distinct tasks**: 1 for Ready, 3 for Red, 3 for Green, and 1 for Yellow.

(a)

⇨ **FSM architecture:**

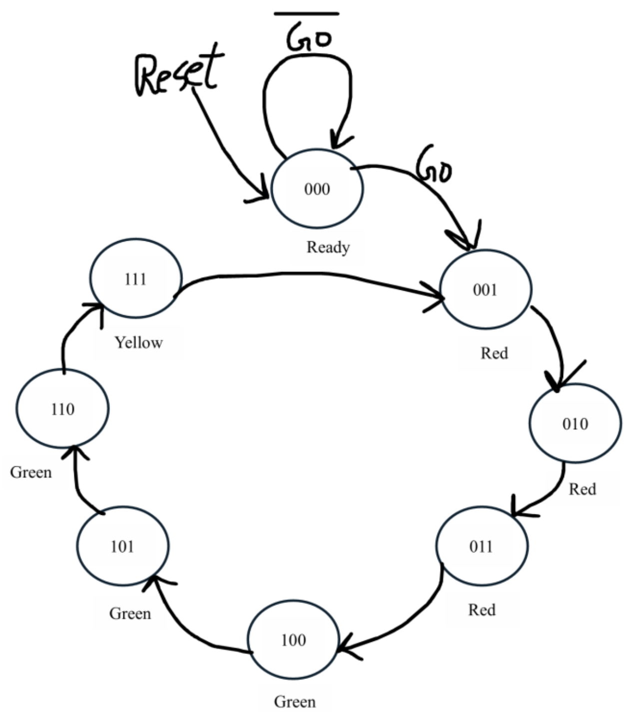


FSM Block Diagram:



(b)

⇨ State chart/State Diagram:



(c)

➪ **State Table:**

Reset	Input	PState			NState			Outputs			
	Go	PS_2	PS_1	PS_0	NS_2	NS_1	NS_0	Ready	Red	Green	Yellow
1	X	X	X	X	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
	1				0	0	1				
0	X	0	0	1	0	1	0	0	1	0	0
0	X	0	1	0	0	1	1	0	1	0	0
0	X	0	1	1	1	0	0	0	1	0	0
0	X	1	0	0	1	0	1	0	0	1	0
0	X	1	0	1	1	1	0	0	0	1	0
0	X	1	1	0	1	1	1	0	0	1	0
0	X	1	1	1	0	0	1	0	0	0	1

(d)

➪ **Logic equations for next states and outputs:**

➔ **Let, NState = NS**

PState = PS

∴ Equation for NState:

➔ $NS0 = \overline{Reset}.(\overline{PS2}.\overline{PS1}.\overline{PS0}.GO + \overline{PS2}.PS1.\overline{PS0} + PS2.\overline{PS1}.\overline{PS0} + PS2.PS1.\overline{PS0} + PS2.PS1.PS0)$

➔ $NS1 = \overline{Reset}.(\overline{PS2}.\overline{PS1}.PS0 + \overline{PS2}.PS1.\overline{PS0} + PS2.\overline{PS1}.PS0 + PS2.PS1.\overline{PS0})$

➔ $NS1 = \overline{Reset}.(\overline{PS2}.PS1.PS0 + PS2.\overline{PS1}.\overline{PS0} + PS2.\overline{PS1}.PS0 + PS2.PS1.\overline{PS0})$

∴ Equation for Outputs:

➔ $Ready = \overline{PS2}.\overline{PS1}.\overline{PS0}$

➔ $RED = (\overline{PS2}.\overline{PS1}.PS1 + \overline{PS2}.PS1.\overline{PS0} + \overline{PS2}.PS1.PS0)$

➔ $Green = (PS2.\overline{PS1}.\overline{PS0} + PS2.\overline{PS1}.PS0 + PS2.PS1.\overline{PS0})$

➔ $Yellow = (PS2.PS1.PS0)$

Sub : Day Time : Date : / /

#555 Timer IC

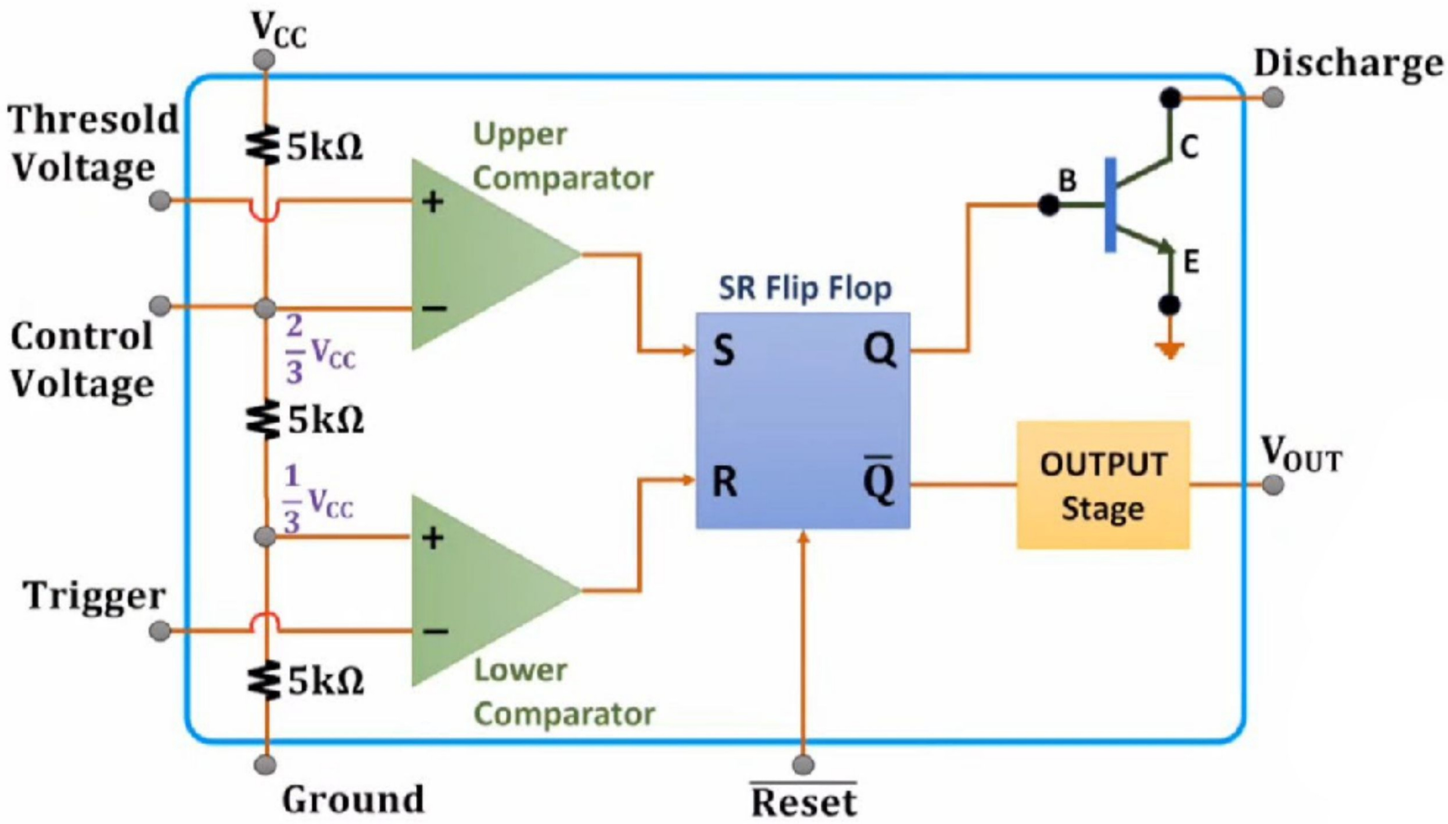
555 Timer IC

⇒ It can be used as a Timer & Counter.

There are three modes of operation:—

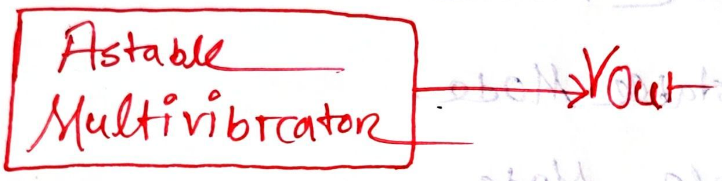
1. Astable Mode.
2. Monostable Mode.
3. Bistable Mode.

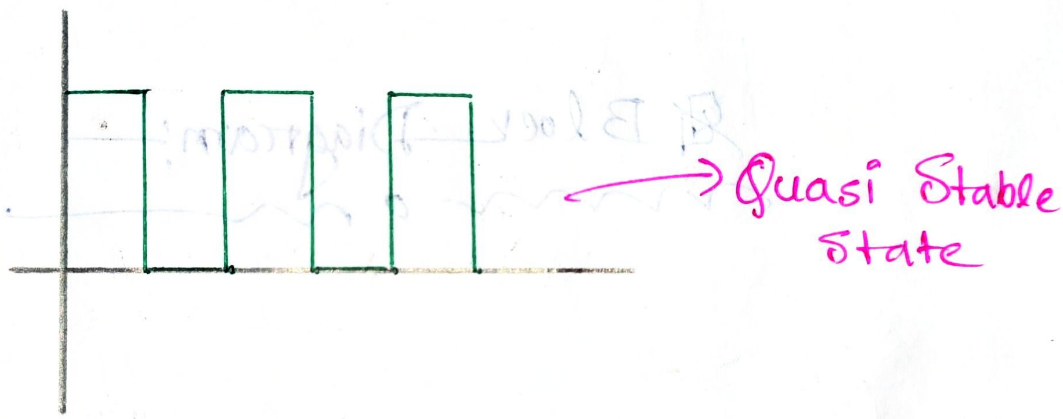
Block Diagram:—



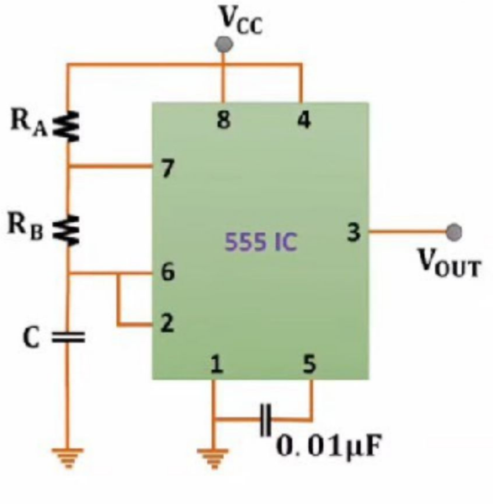
Astable Multivibrator

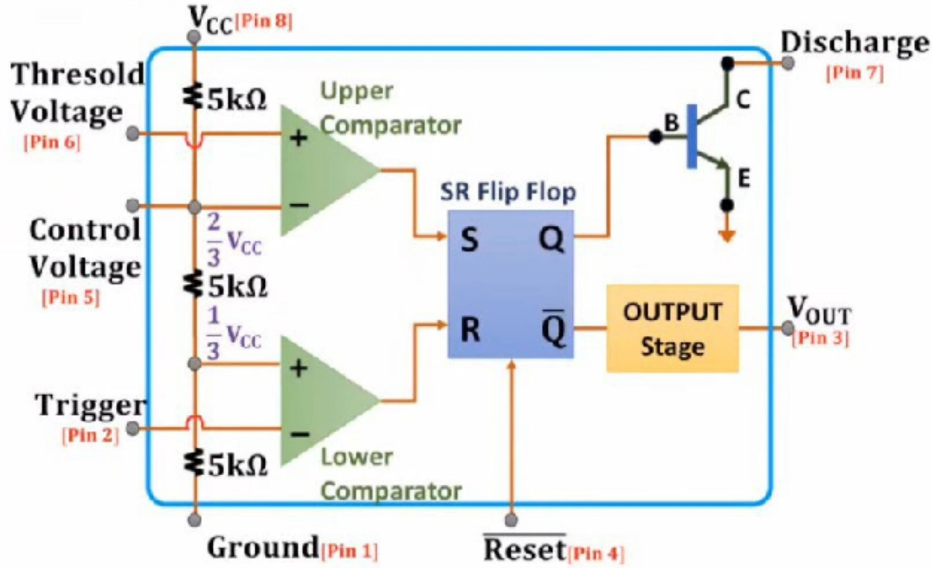
⇒ Astable Multivibrator has no stable state. Both states are Quassi-Stable state. It canbe uses as a square wave generator.





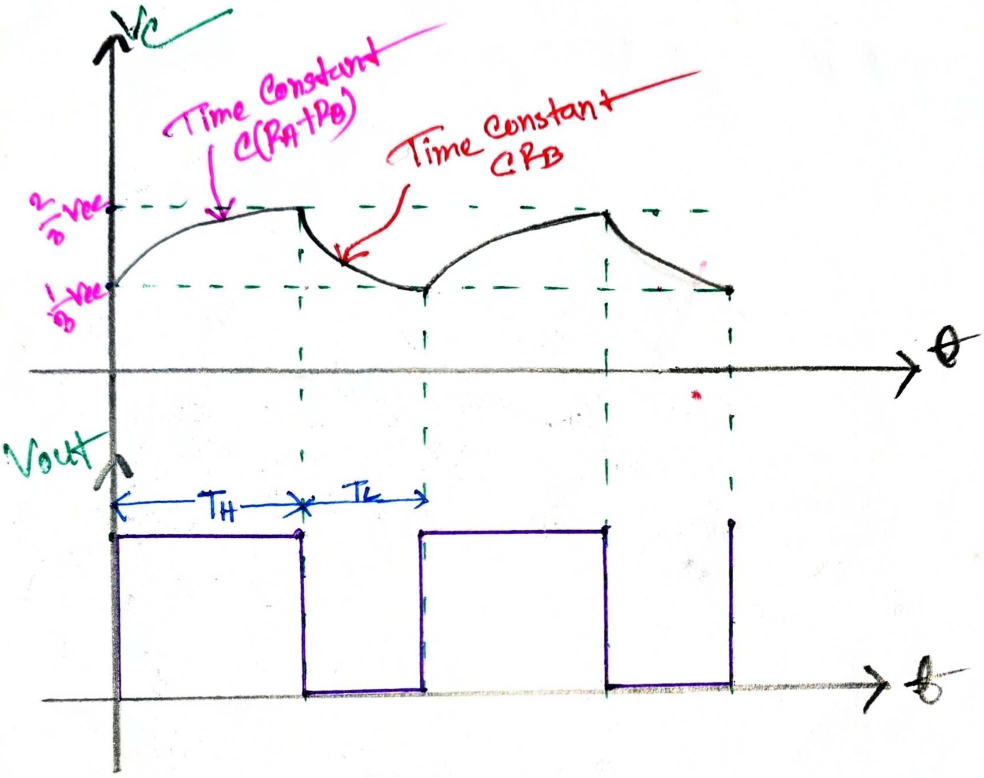
Block Diagram





Sub :   Time :   Date :   /   /

Waveforms of Astable Multivibrator



$$\therefore\ V_c = V_F + (V_I - V_F)e^{-t/RC}$$

∴ During Charging :-

$t = T_H,\ R = R_A + R_B$

$$\therefore\ \frac{2}{3}V_{cc} = V_{cc} + \left\{\left(\frac{1}{3}V_{cc}\right) - V_{cc}\right\} e^{-t/C(R_A+R_B)}$$

$$\Rightarrow\ -\frac{1}{3} = -\frac{2}{3}e^{-T_H/C(R_A+R_B)}$$

$$\Rightarrow\ \frac{1}{2} = e^{-T_H/C(R_A+R_B)}$$

$$\therefore\ \boxed{T_H = 0.693\,C_1(R_A+R_B)}$$

During Discharging:-

$t = T_L \; , \; R = R_B$

$\therefore \Rightarrow \frac{1}{3} V_{cc} = 0 + \left(\frac{2}{3} V_{cc} - 0\right) e^{-T_L/CR_B}$

$\Rightarrow \frac{1}{2} = e^{-T_L/CR_B}$

$\Rightarrow \boxed{T_L = 0.693\, C R_B}$

If the frequency is given:-

We know:-

$$\boxed{T = T_H + T_L}$$

$$\therefore \; F = \frac{1}{T}$$

$$F = \frac{1}{T_L + T_H}$$

$$= \frac{1}{0.693\, C R_B + 0.693\, C (R_B + R_A)}$$

$$\therefore \; \boxed{F = \frac{1}{0.693\, C (R_A + 2R_B)}}$$

Sub :

Day

Time : Date : / /

For Duty Cycle:-

$$D \cdot C = \frac{T_H}{T}$$

$$\therefore \boxed{D \cdot C = \frac{T_H}{T_H + T_L}}$$

problem: 1:-

⇒ Design a oscillator for a frequency of 200 Hz with duty cycle of 78%. Determine

(a) Time period.

(b) High & low time.

(c) R_2 & R_1 where $C_1 = C = 10\mu F$

⇒ Is given:-

$$f = 200 Hz$$

$$D \cdot C = 78\% = 0.78$$

(a)

We know:-

$$T = \frac{1}{f} = \frac{1}{200} = 5 \times 10^{-3} sec$$

Sub : 	Day 	Time : 	Date : / /

(b)

⇒ We know:—

$$D.C = \frac{T_H}{T}$$

$\therefore T_H = D.C \times T$

$= 0.78 \times 5 \times 10^{-3}$

$= 3.9\,ms$

$T = T_H + T_L$

$\therefore T_L = T - T_H$

$= 5ms - 3.9ms$

$= 1.1ms$

Ans:—

(c)

⇒ We know:—

$T_H = 0.693\,C(R_1 + R_2)$

$\therefore R_1 = (T_H / 0.693C) - R_2$ —— (1)

& $T_L = 0.693\,C R_2$

$\therefore R_2 = T_L / 0.693C$

$$= \frac{1.1ms}{0.693 \times 10\mu}$$

$= 158.73\,\Omega$

$T_L = 1.1ms$
$C = 10\mu F$

Sub : Day Time : Date : / /

Now, put the value of R_2 in eq (i):-

$$\therefore R_1 = \frac{T_H}{0.693C} - R_2$$

$$= \frac{3.9ms}{0.693 \times 10\mu} - 158.73$$

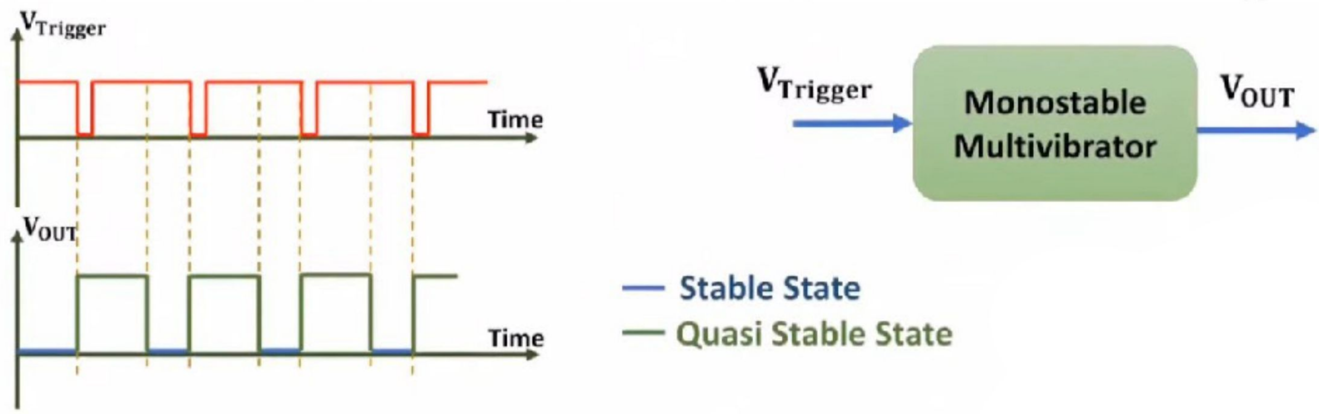
$$= 404.04\,\Omega$$

$T_H = 3.9m$
$C = 10\mu F$
$R_2 = 158.73\Omega$

Ans:-

Monostable Multivibrator:-

- Monostable has one stable state and the second state is not stable, it is Quasi stable State.
- Stable State: This state doesn't change. Stable State changes by external trigger.
- Quasi-stable State: This state changes based on the circuit. It changes automatically.

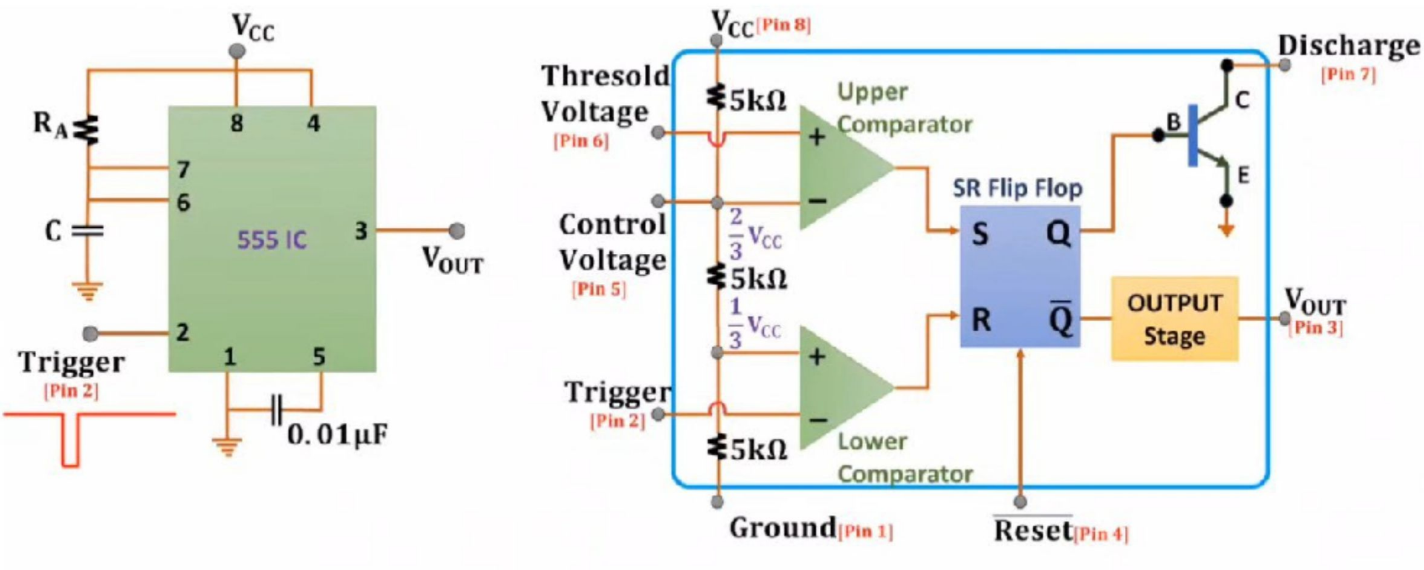


Sub :

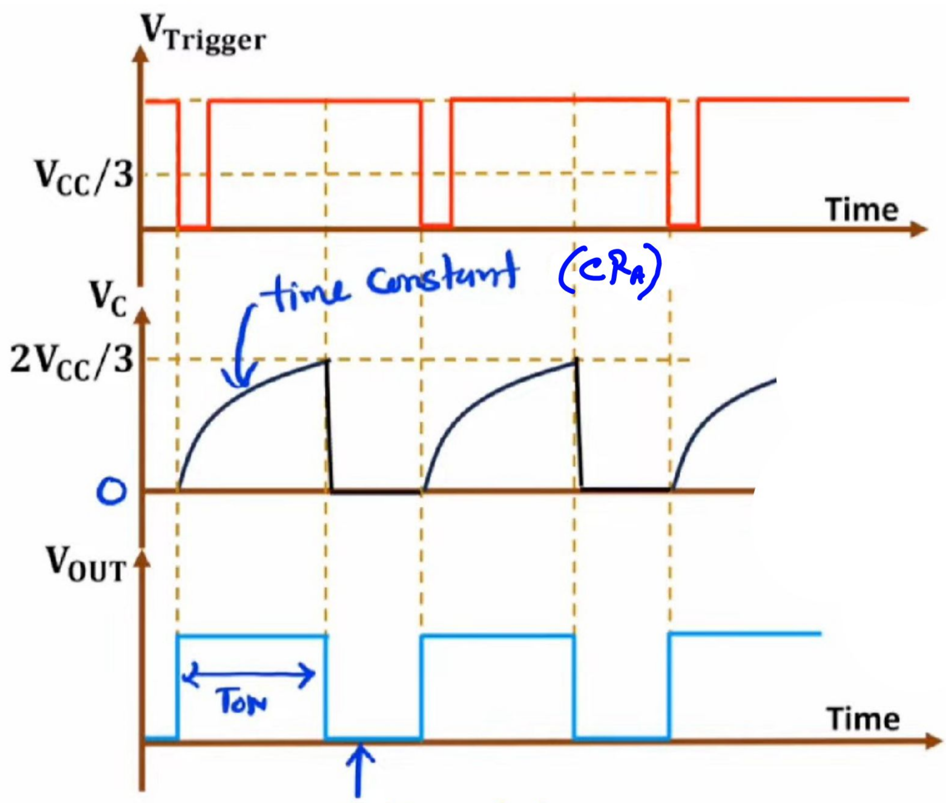
Day

Time : Date : / /

Block Diagram :-



Waveforms of MonoStable Multivibrator



Sub :

Time : Date : / /

$\therefore \Rightarrow V_C = V_F + (V_I - V_F) e^{-t/RC}$

During Charging:-

$\Rightarrow t = t_H, R = R_A$

$\therefore \Rightarrow \frac{2}{3} V_{CC} = V_{CC} + (0 - V_{CC}) e^{T_H/R_AC}$

$\therefore \Rightarrow -\frac{1}{3} V_{CC} = -V_{CC} \cdot e^{-T_H/R_AC}$

$\Rightarrow 1.1 = T_H / CR_A$

$\therefore \boxed{T_H = 1.1 R_A C}$

Probleme 2:-

$\Rightarrow$ if the frequency of 400Hz with 70% duty cycle. Determine

(a) Time period.

(b) High time if $C = 20\mu F$

(c) Value of R_A

$\Rightarrow$ Is given:-

$D.C = 70\% = 0.7$

$f = 200 + 200 = 400 Hz$

$C = 20\mu F$

Sub :

Day

Time :   Date : / /

(a)

⇒ We know:-

$$T = \frac{1}{F} = \frac{1}{400} = 2.5ms$$

Ans:-

(b)

⇒ We know:-

$$D.C = \frac{T_H}{T}$$

$$\therefore T_H = D.C \times T$$

$$= 0.7 \times 2.5ms$$

$$= 1.75ms$$

(c)

⇒ We know:-

$$T_H = 1.1 CR_A$$

$$\therefore R_A = \frac{T_H}{1.1 \times 20uF}$$

$$= \frac{1.75ms}{1.1 \times 20u}$$

$$= 79.54 \,\Omega$$

Sub :

Time : Date : / /

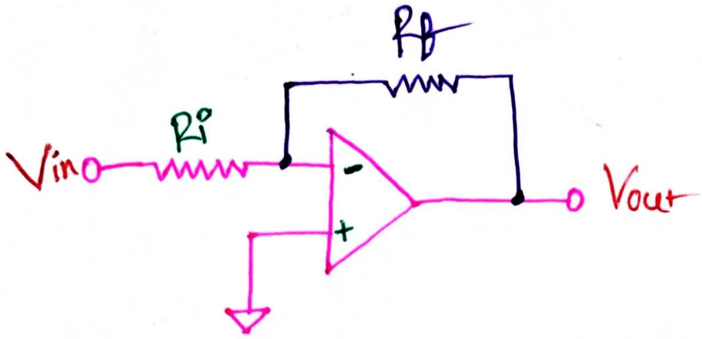
Digital to Analog Converter (DAC)

Sub :

Day

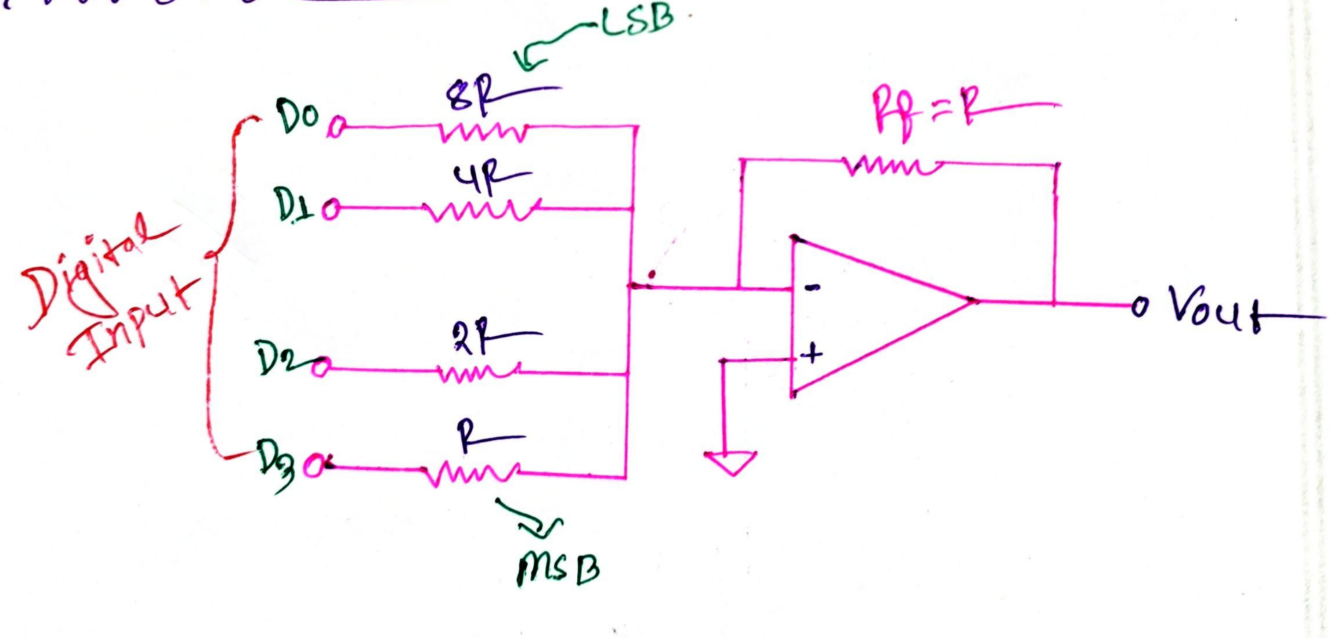
Time :   Date : / /

Binary Weighted DAC



$$\therefore V_{out} = -V_{in}\left(\frac{R_f}{R_i}\right)$$

Example: 1:



$$\therefore V_{out} = -\left(\frac{R}{8R}\times V_0 + \frac{R}{4R}V_1 + \frac{R}{2R}V_2 + \frac{R}{R}V_3\right)$$

$$\Rightarrow V_0 = -\left(\frac{1}{8}V_0 + \frac{1}{4}V_1 + \frac{1}{2}V_2 + \frac{1}{1}V_3\right)$$

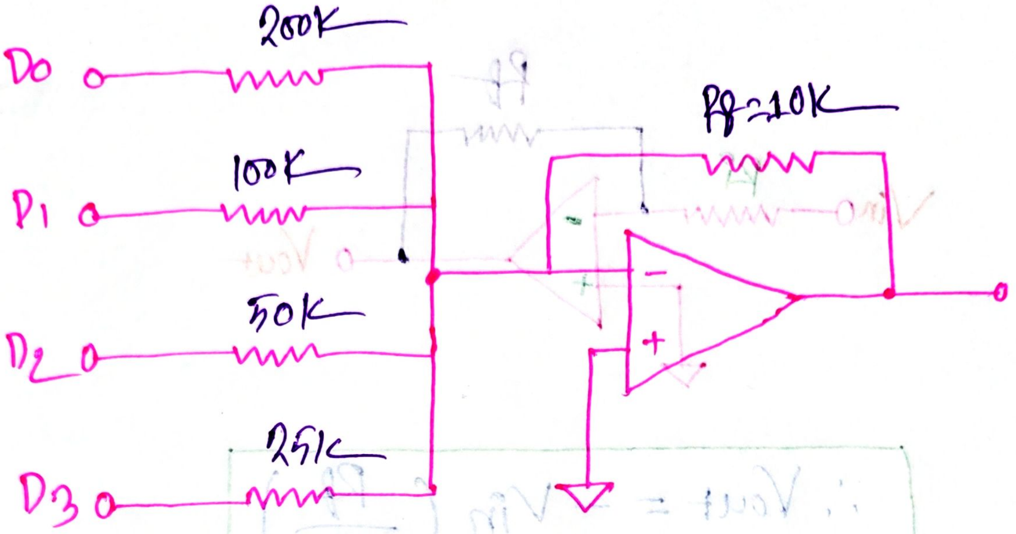
$$\Rightarrow \boxed{V_0 = -\left(\frac{V_0}{8} + \frac{V_1}{4} + \frac{V_2}{2} + \frac{V_3}{1}\right)}$$

Sub :

Day

Time :   Date :   /   /

Example 1:-



⇒ Determine the output of the DAC if waveform representing a sequence of 4-bit numbers are applied to the inputs. D0 is the LSB.

⇒ Since 4-bit,
There are 16 combinations.

For: 0001:-

We know:-

$$V_{out} = -\left(\frac{V_0 R_f}{R_0} + \frac{V_1 R_f}{R_1} + \frac{V_2 R_f}{R_2} + \frac{V_3 R_f}{R_3}\right) \quad \text{--- (1)}$$

In given:-

$R_f = 10K$

$V_0 = 5V \longrightarrow [\because 0001]$

$V_1 = 0$

$V_2 = 0$

$V_3 = 0$

$R_0 = 200K\ \Omega$

$R_1 = 100K\ \Omega$

$R_2 = 50K\ \Omega$

$R_3 = 25K\ \Omega$

Sub :

Day

Time : Date : / /

put the values in eq. (1) ⇒

$$\therefore V_{out} = -\left(\frac{5V \times 10K}{200K} + 0 + 0 + 0\right) = -0.25V$$

Now, for : 0010 :—

$V_0 = 0,\ V_1 = 5V,\ V_2 = 0V,\ V_3 = 0V$

$$\therefore V_{out} = -\left(\frac{V_1 \times R_f}{R_1} + 0 + 0 + 0\right) =$$

$$= -\left(\frac{5 \times 10K}{100K} + 0 + 0 + 0\right) = -0.5V$$

Now, for : 0100 :—

$V_0 = 0,\ V_1 = 0V,\ V_2 = 5V,\ V_3 = 0V$

$$\therefore V_{out} = -\left(\frac{5 \times 10K}{50K} + 0 + 0 + 0\right) = -1V$$

Now, for : 1000 :—

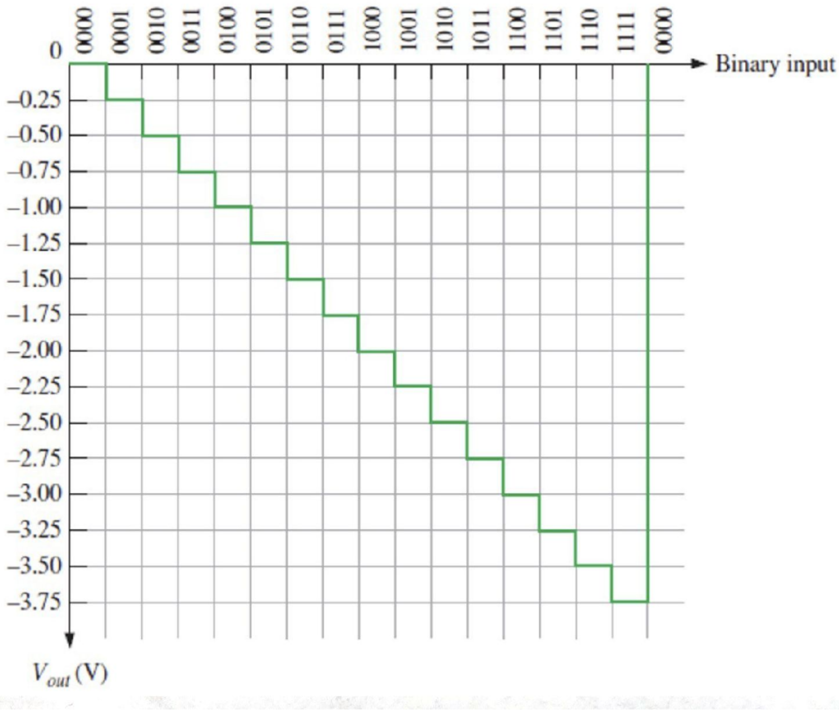
$V_0 = 0,\ V_1 = 0,\ V_2 = 0,\ V_3 = 05V$

$$\therefore V_{out} = -\left(\frac{5 \times 10K}{25K} + 0 + 0 + 0\right) = -2V$$

So, Now you know the game, Rest of the Combinations are Similar.

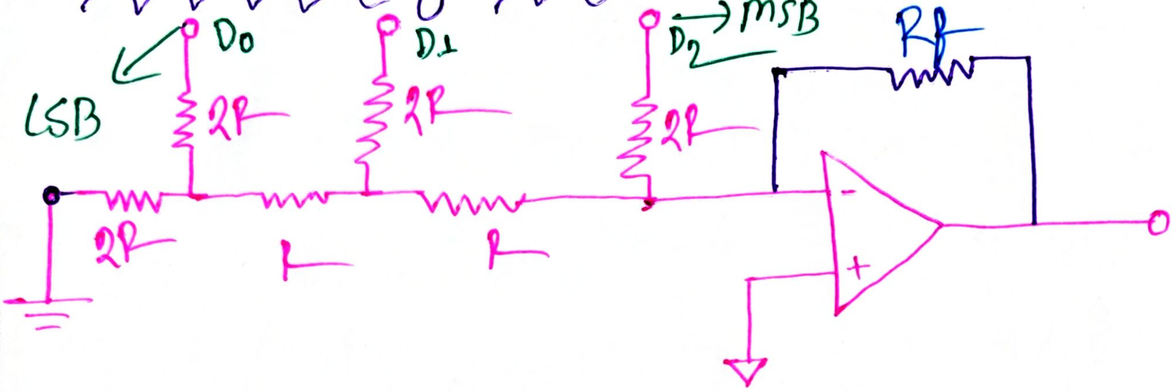
Data Table:

D3	D2	D1	D0	Vout (V)
0	0	0	0	0
0	0	0	1	−0·25
0	0	1	0	−0·50
0	0	1	1	−0·75
0	1	0	0	−0·1 0
0	1	0	1	−1·25
0	1	1	0	−1·50
0	1	1	1	−1·75
1	0	0	0	−2·00
1	0	0	1	−2·25
1	0	1	0	−2·50
1	0	1	1	−2·75
1	1	0	0	−3·00
1	1	0	1	−3·25
1	1	1	0	−3·50
1	1	1	1	−3·75

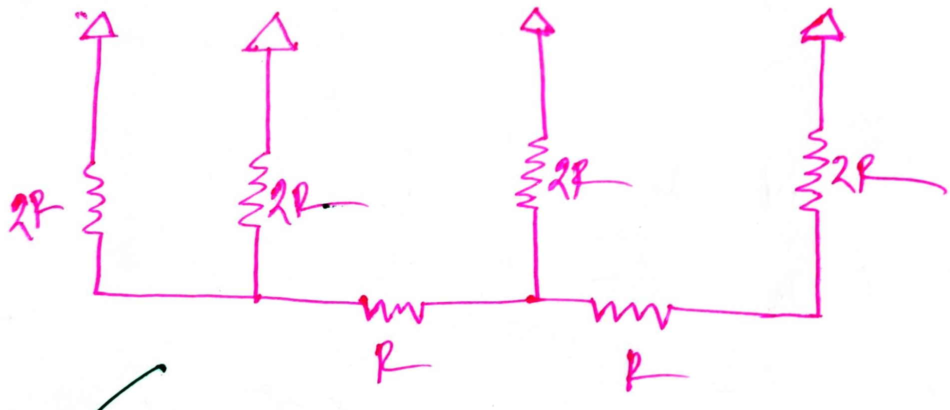


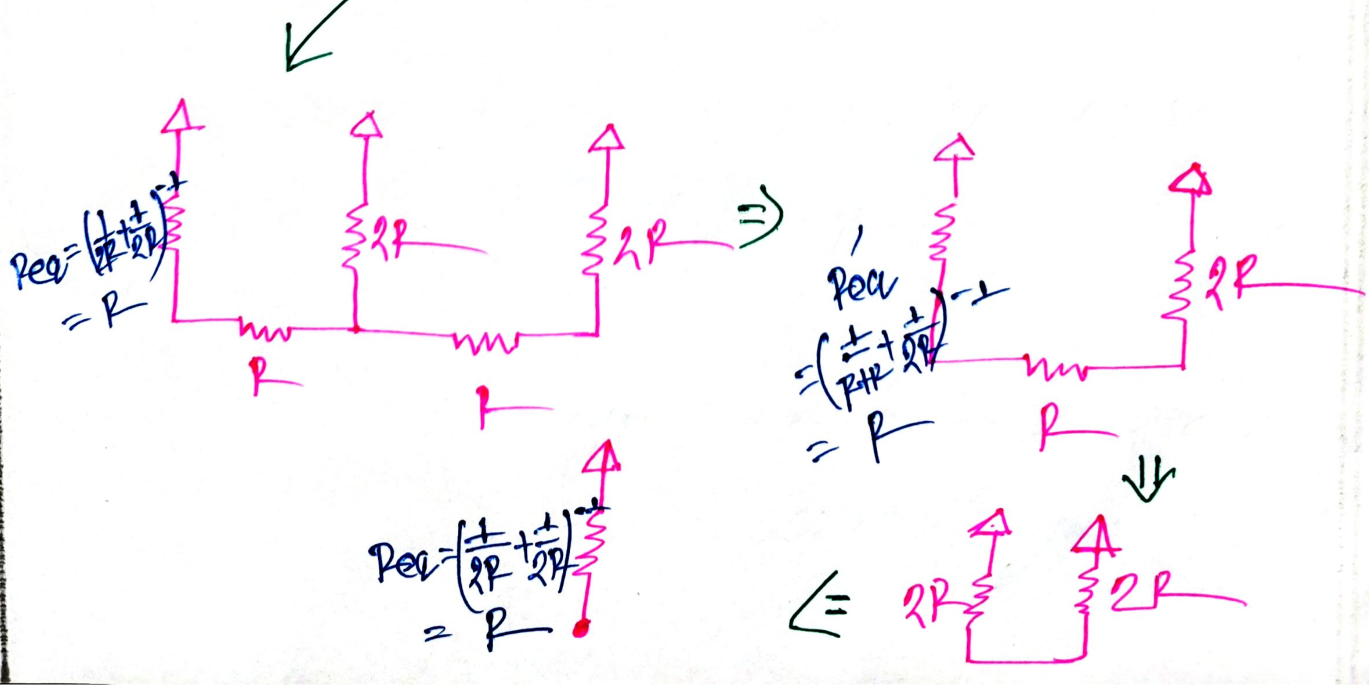
R-2R Ladder DAC

3 bit R-2R Ladder:-



⇒ Find the impedence of Network





$$R_{eq} = \left(\frac{1}{2R}+\frac{1}{2R}\right)^{-1} = R$$

$$R_{eq} = \left(\frac{1}{R+R}+\frac{1}{2R}\right)^{-1} = R$$

$$R_{eq} = \left(\frac{1}{2R}+\frac{1}{2R}\right)^{-1} = R$$

Sub :

Day

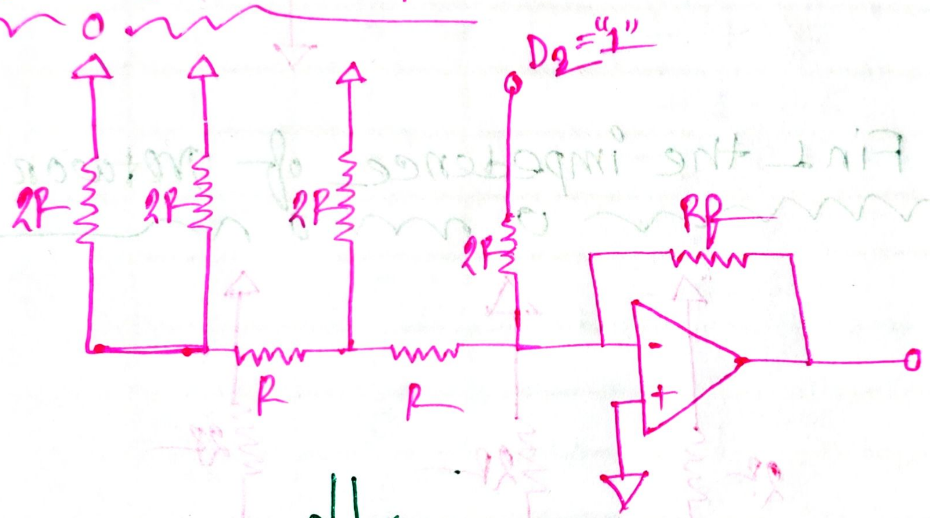
Time : Date : / /

Note!– The impedence of network is R, regardless of number of bits.

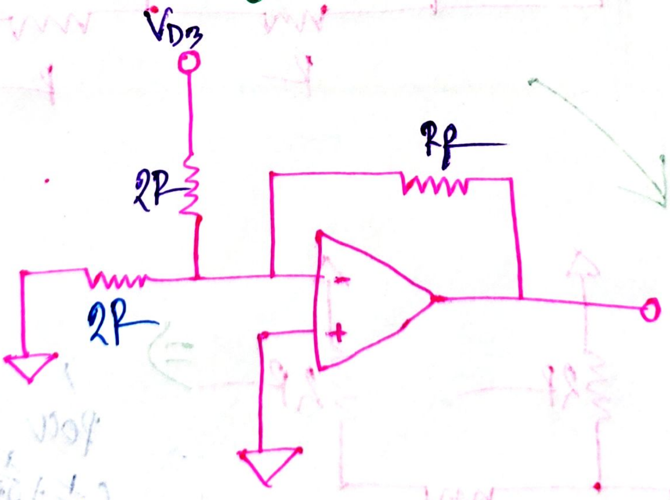
Since 3 bit, so there are 8 combinations!–

⇒ For!– $D_2 = 1,\ D_0 = D_1 = 0$

⇒ Redraw the circuit!–



⇓



$$\therefore R_{Th} = \left(\frac{1}{2R} + \frac{1}{2R}\right)^{-1} = R$$

$$\therefore V_{Th} = V_{D_2} \times \left(\frac{2R}{2R+2R}\right) = \frac{V_{D2}}{2}$$

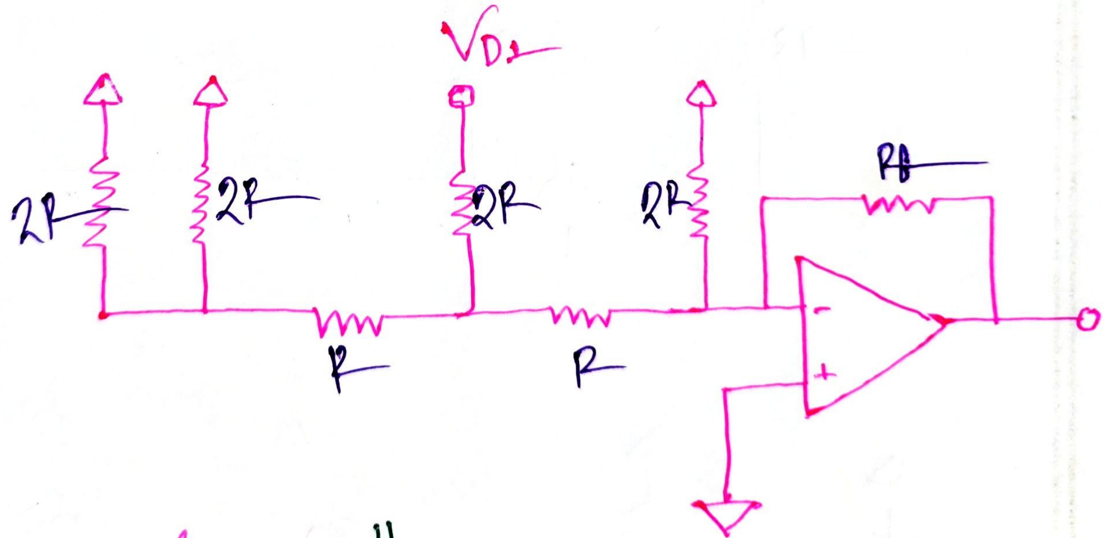
Sub : Day Time : Date : / /

∴ equivalent circuit :-

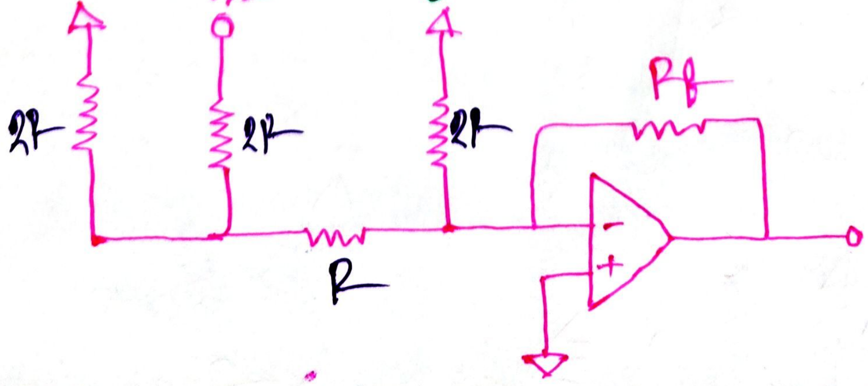
$$\therefore V_o = -\left(\frac{V_{Th} \times R_f}{R_{Th}}\right) = -\left(\frac{V_{D2}}{2} \times \frac{R_f}{R}\right)$$

⇒ For $D_1 = 1$, $D_2 = D_0 = 0$:-

⇒ Redraw the circuit :-



⇓

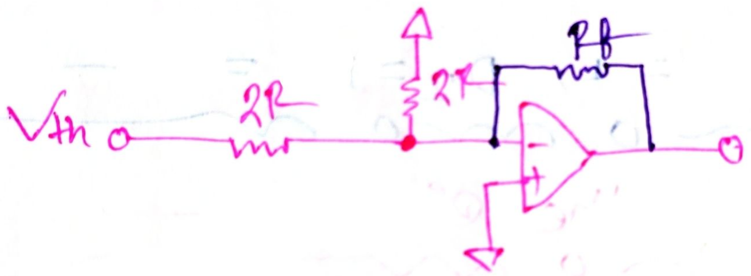


$$\therefore R_{Th} = \left(\frac{1}{2R} + \frac{1}{2R}\right)^{-1} = R$$

$$\therefore V_{th} = V_{D2} \times \left(\frac{2R}{2R+2R}\right) = \frac{V_{D2}}{2} \times 1$$

∴

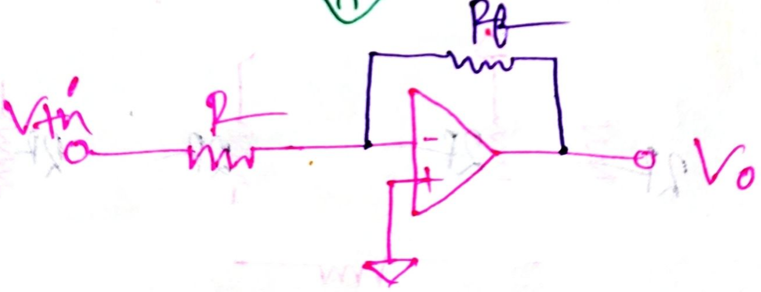
$\Downarrow$



$$\therefore V_{th}' = \frac{\left(+\frac{V_{D1}}{2}\right) \times 2R}{2R} = \frac{V_{D1}}{4}$$

$$R_{th} = \left(\frac{1}{2R} + \frac{1}{2R}\right)^{-1} = R$$

$\Downarrow$



$$\boxed{\therefore V_o = -\left(\frac{V_{th} \times R_f}{R}\right) = -\left(\frac{V_{D2}}{4} \times \left(\frac{R_f}{R}\right)\right)}$$

$\therefore$ For $\underbrace{D_0 = 1}_{}$, $\underbrace{D_1}_{0} = \underbrace{D_2}_{0} = 0V$:-

$$\boxed{V_o = -\left(\frac{V_{D0}}{8} \times \frac{R_f}{R}\right)}$$

Sub :

Day

Time : Date : / /

∴ The generalised eq :—

$$\boxed{V_0 = -\left(\frac{R_f}{R}\right) \times V\left(\frac{D_2}{2} + \frac{D_1}{4} + \frac{D_0}{8}\right)}$$

$$\boxed{\therefore V_0 = -V\left(\frac{D_2}{2} + \frac{D_1}{4} + \frac{D_0}{8}\right)} \quad [\because R_f = R]$$

∴ Digital data to Analog

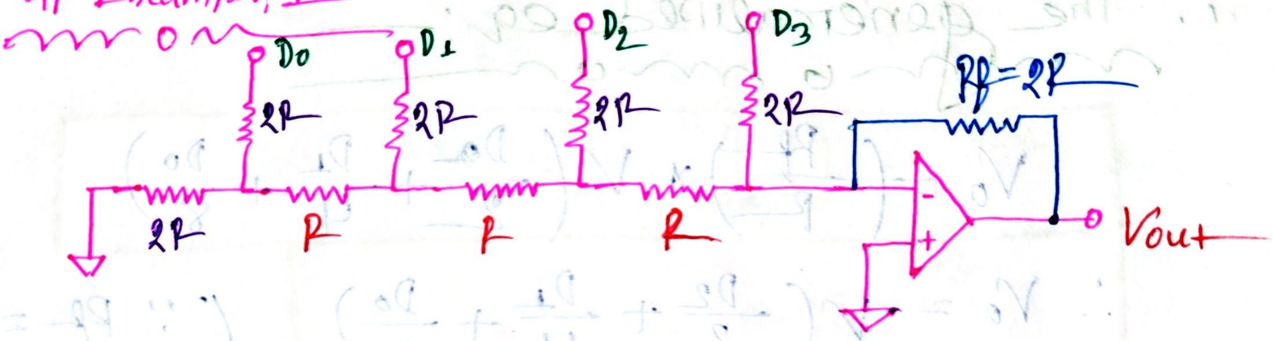
D_2	D_1	D_0	Analog V_0
0	0	0	0
0	0	1	$-V/8$
0	1	0	$-2V/8$
0	1	1	$-3V/8$
1	0	0	$-4V/8$
1	0	1	$-5V/8$
1	1	0	$-6V/8$
1	1	1	$-7V/8$

Sub :

Day

Time : Date : / /

Example 1



⇒ Find the value of V_0 for (i) $D_3 = 1,\ D_2 = D_1 = D_0 = 0$.

(ii) $D_2 = 1,\ D_3 = D_1 = D_0 = 0$

(iii) $D_1 = 1,\ D_3 = D_2 = D_0 = 0$

(iv) $D_0 = 1,\ D_3 = D_2 = D_1 = 0$

⇒ We know:—

$$V_0 = -\left(\frac{R_f}{R}\right)\left(\frac{D_3}{2} + \frac{D_2}{4} + \frac{D_1}{8} + \frac{D_0}{16}\right) \times V$$

∴ Now, case 1:—

$D_3 = 1,\ D_2 = 0,\ D_1 = 0,\ D_0 = 0$

$$\therefore V_0 = -\left(\frac{2R}{R}\right) \times \left(\frac{1}{2}\right) V$$

$$= -(2) \times \left(\frac{1}{2}\right) \times 5$$

$$= -5V$$

here,

$V = 5V$

$D_3 = 1$

$R_f = 2R$

$R = R$

∴ Now case 2:—

$D_3 = 0,\ D_2 = 1,\ D_1 = 0,\ D_0 = 0$

Sub :

Day

Time : Date : / /

$$\therefore V_0 = -\left(\frac{2R}{R}\right) \times \left(\frac{1}{4}\right) \times 5$$
$$= -2.5V$$

Since,

$D_2 = 1$

Now, case 3:-

$D_3 = 0$, $D_2 = 0$, $D_1 = 1$, $D_0 = 0$

$$\therefore V_0 = -\left(\frac{2R}{R}\right) \times \left(\frac{1}{8}\right) \times 5$$
$$= -1.25V$$

Since,

$D_1 = 1$

$\therefore$ Now case 4:-

$D_3 = 0$, $D_2 = 0$, $D_1 = 0$, $D_0 = 0$

$$\therefore V_0 = -\left(\frac{2R}{R}\right) \times \left(\frac{1}{16} \times 5\right)$$
$$= -0.625V$$

Since,

$D_0 = 1$

Sub :

Day

Time : Date : / /

Resolutin of DAC

Resolution of n-bit DAC:-

$$\boxed{RDAC = \frac{1}{2^n - 1} \times 100}$$

problem:1:-

⇒ Determine the resolution expressed as a percentage of the following:-

(a) An 8-bit DAC

(b) A 12 bit DAC.

⇒ We Know:-

$$RDAC = \frac{1}{2^n - 1} \times 100$$

(a)

For 8-bit DAC:-

$$RDAC = \frac{1 \times 100}{2^8 - 1} = 0.392\%$$

(b)

For 12-bit DAC:-

$$RADC = \frac{1}{2^{12} - 1} \times 100 = 0.0244\%$$

Sub :

Day

Time : Date : / /

Analog To DIGITAL CONVERSION (ADC)

Sub :

Day

Time : Date : / /

Flash ADC

⇒ Flash ADC is fastest among all ADC. It takes only one clock cycle to have the Conversion.

Structure of Flash ADC

⇒ Consist three major components.

1. High speed comparator [$2^n - 1$]
2. Resistive voltage divider Network [2^n]
3. Priority Encoder [1]

8 bit Flash ADC

⇒ In given:

$n = 8$.

So 1. Need comparator $= 2^n - 1 = 2^3 - 1 = 8 - 1 = 7$

2. No. of Resistor $= 2^n = 2^3 = 8$

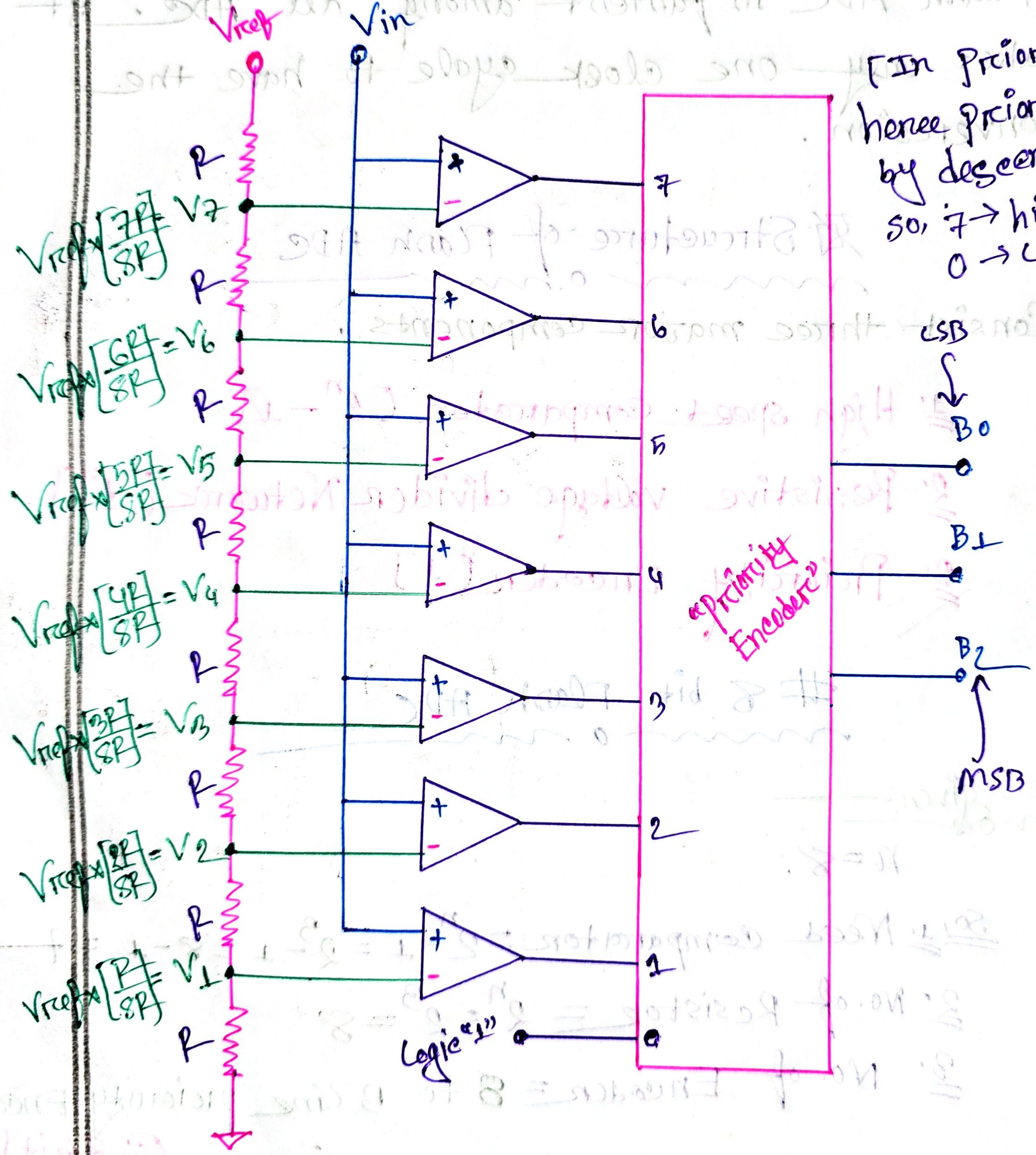
3. No. of Encoder = 8 to 3 Line priority Encoder
[∵ 8 bit]

Sub :

Day

Time : Date : / /

Logic circuit:



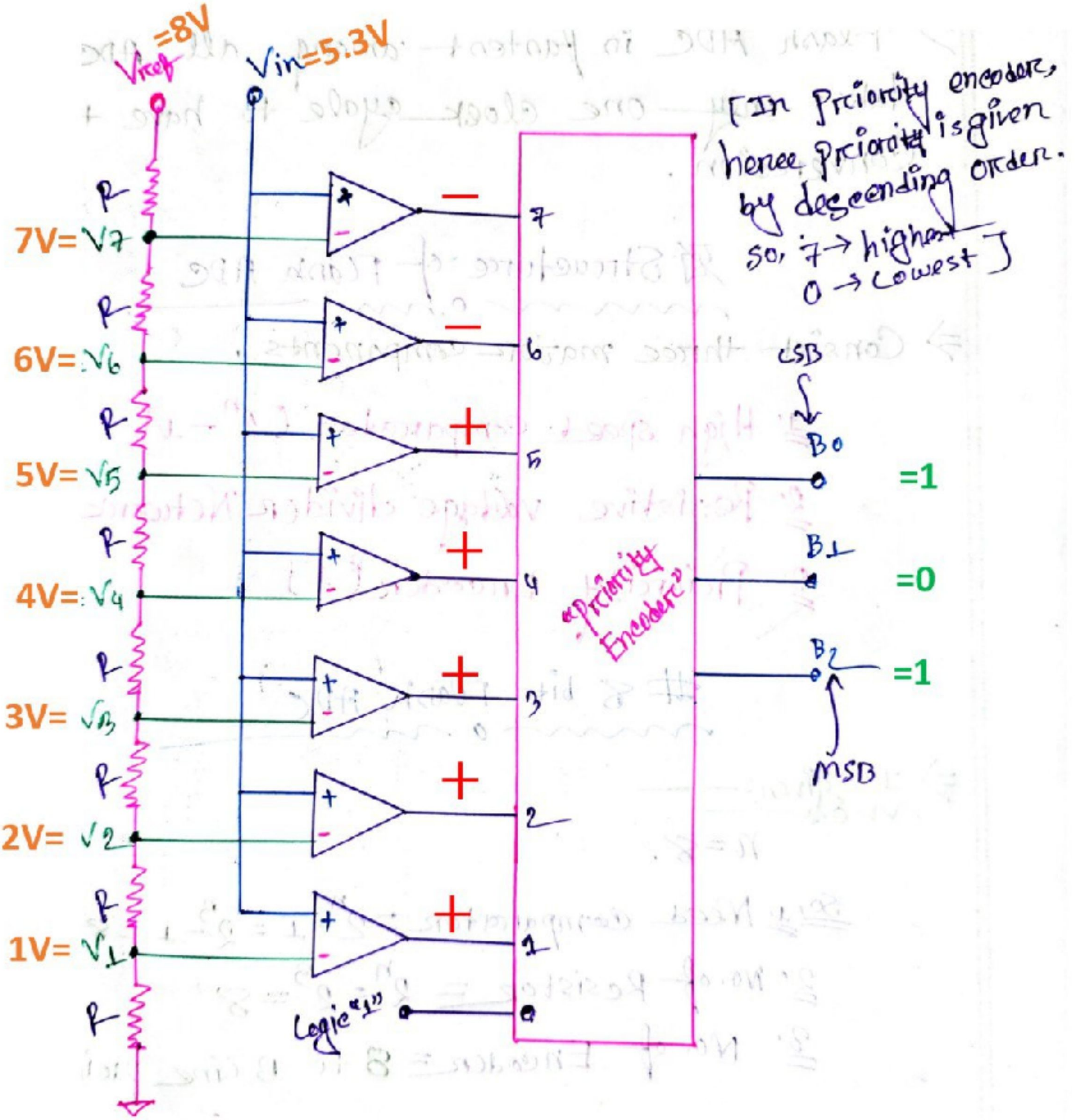
[In Priority encoder, hence priority is given by descending order. so, 7 → highest, 0 → lowest]

Sub :

Day

Time : Date : / /

Example 1:

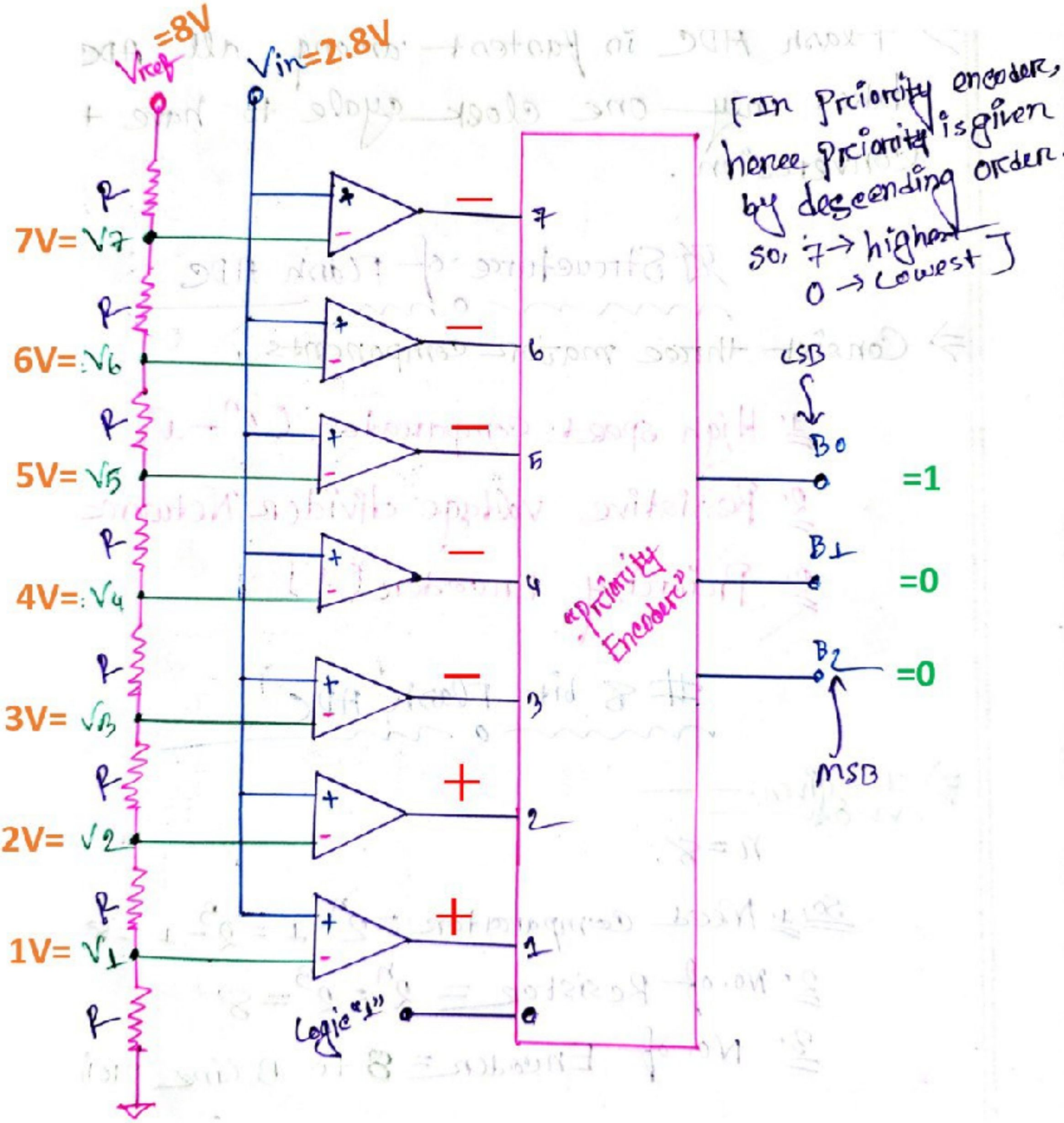


Sub :

Day

Time : Date : / /

Example 2:



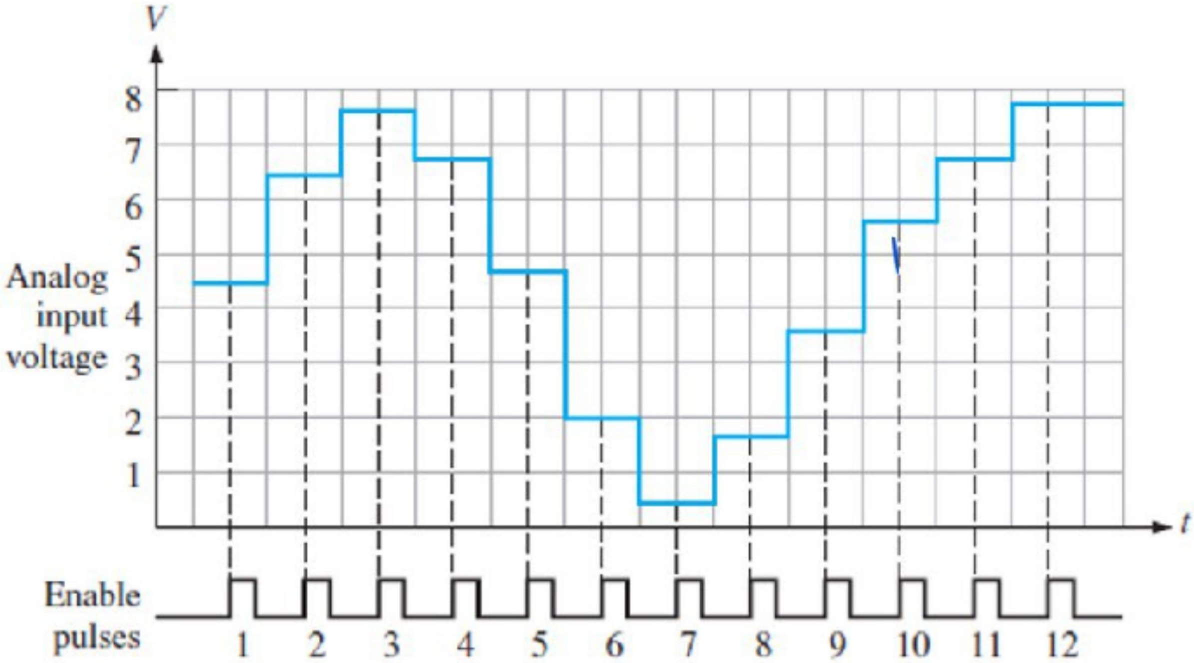
Sub :

Day

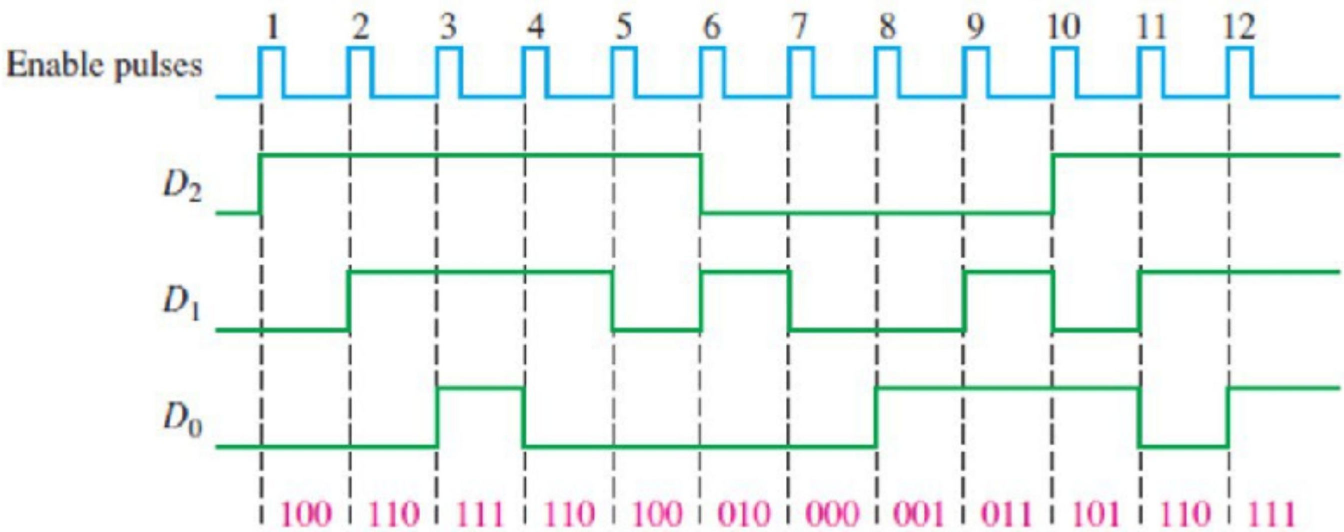
Time : Date : / /

Example 3:

⇨ Determine the binary code output of a **3-bit ADC** for the input signal and **encoder enable pulses shown**. For this example, $V_{REF} = +8V$



➔Solution:

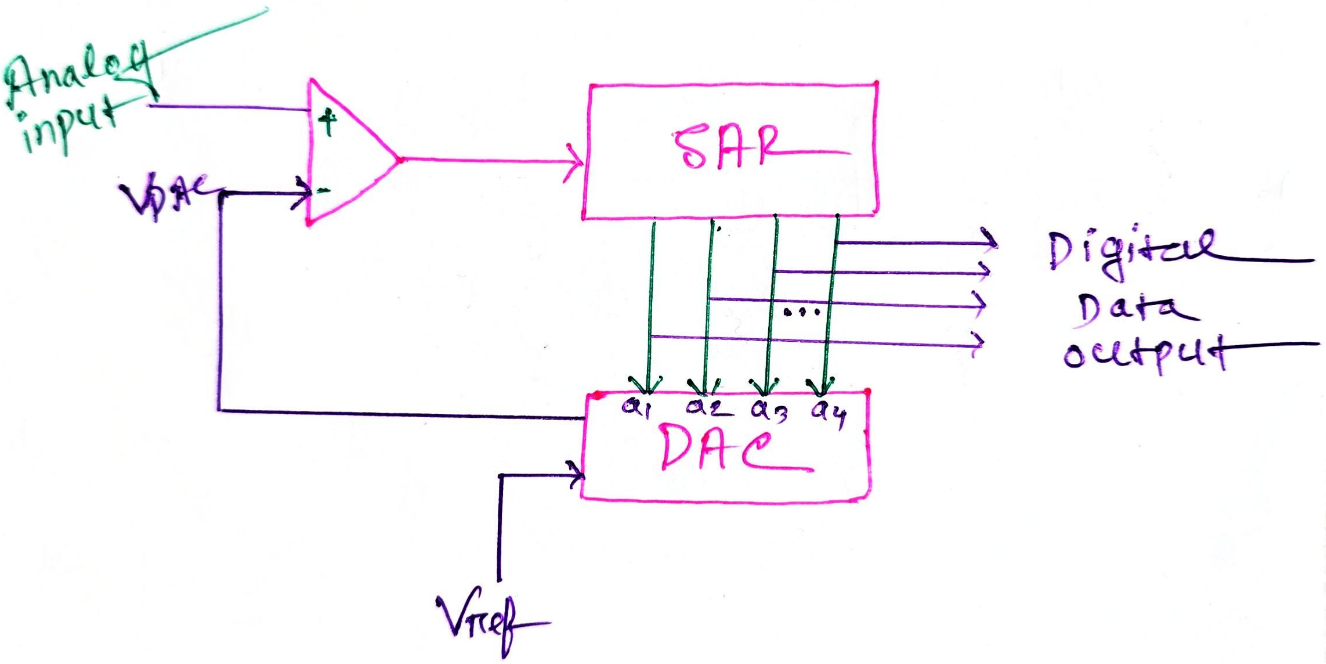


Sub :

Successive Approximation ADC

⇒ This ADC reduces conversion time and its conversion time does not depends on input analog voltage.

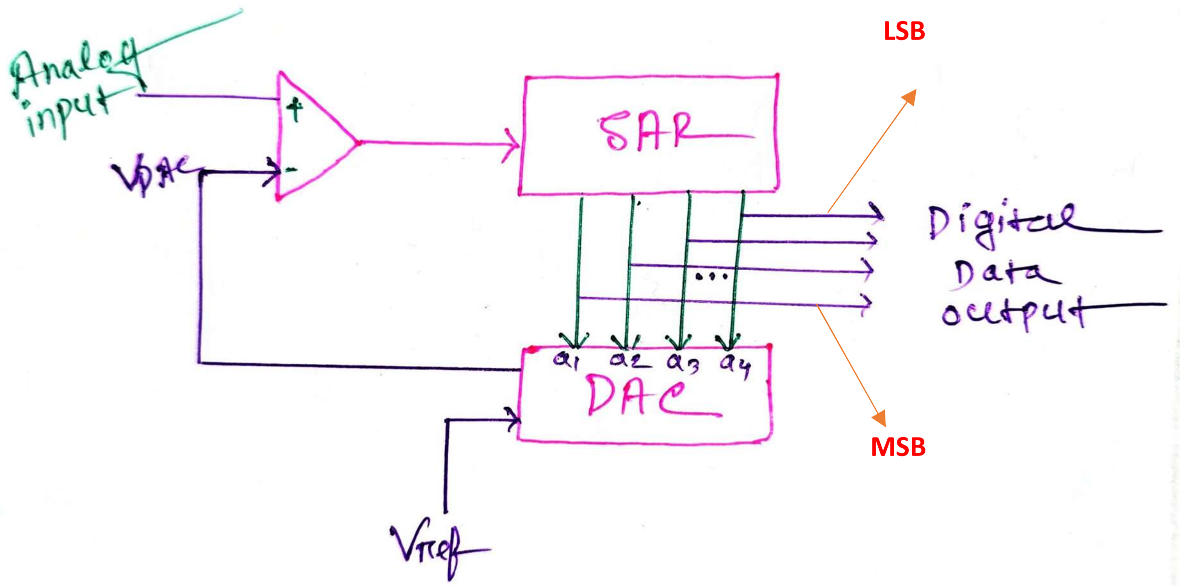
Structure of Successive A·ADC



$$\therefore V_{DAC} = V_{ref}\left[\frac{a_1}{2} + \frac{a_2}{4} + \frac{a_3}{8} + \frac{a_4}{16}\right]$$

SOL

3 Bit Structure of Successive A.ADC



$$\therefore V_{DAC} = V_{ref}\left[\frac{a_1}{2} + \frac{a_2}{4} + \frac{a_3}{8} + \frac{a_4}{16}\right]$$

➔It starts by MSB, SAR = 1000

➔In next clock pulse, if $V_{in} > V_{DAC}$ $then\ SAR = 1100$

➔In next clock, if $V_{in} < V_{DAC}$ $then\ SAR = 0100\ \&\ Continue$

Sub :   Time :   Date : / /

Problem 1:

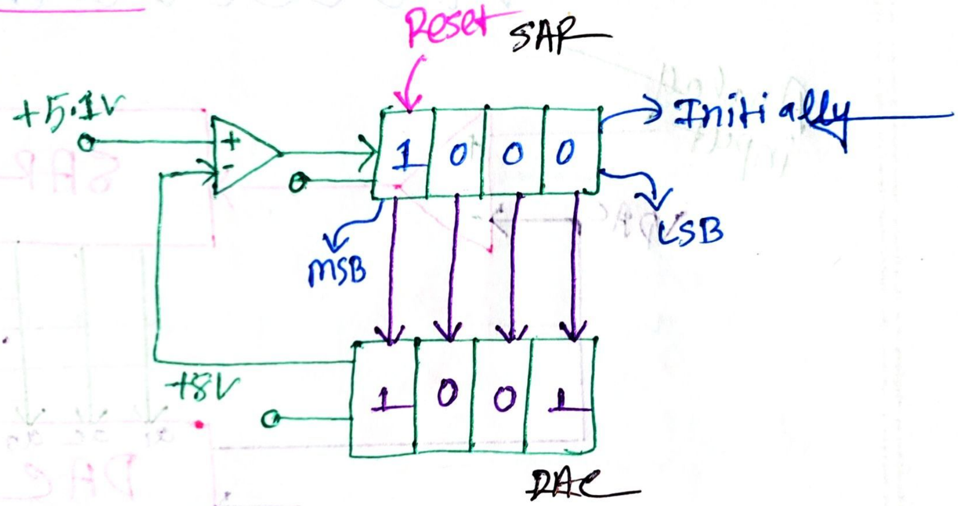
⇒ Construct 6-3=3 bit analog to digital Converter (ADC) using successive approximation method. Show the step wise conversion of an analog signal of 5.1V into 3 bit digital binary value.

⇒ For 1st clk pulse:

$V_{in} < V_{DAC}$

$5.1 < 8$

So, reset

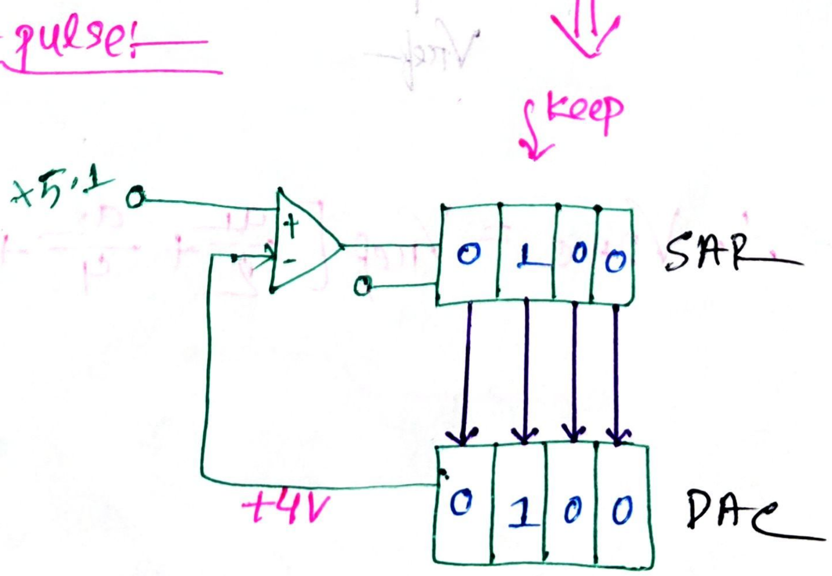


For 2nd clk-pulse:

$V_{in} > V_{DAC}$

$5.1 > 4V$

So, keep



Sub :

Day

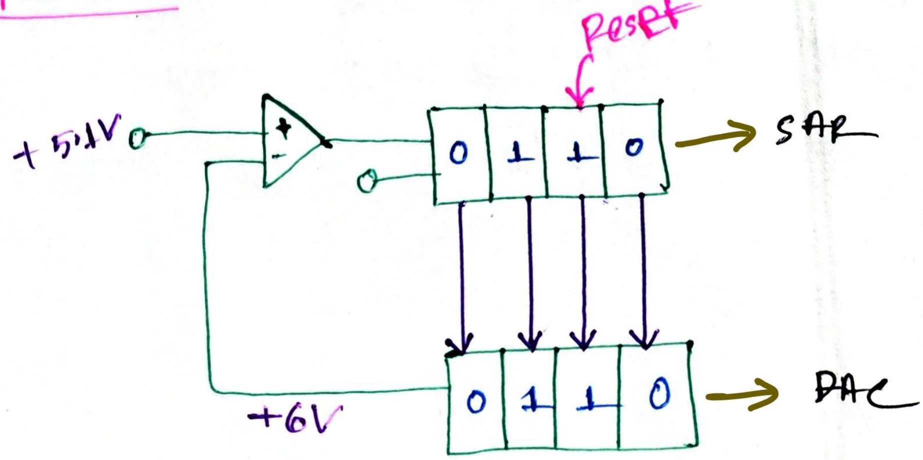
Time : Date : / /

For 3rd clk-pulse:-

Vin < VDAC

5.1 < 6

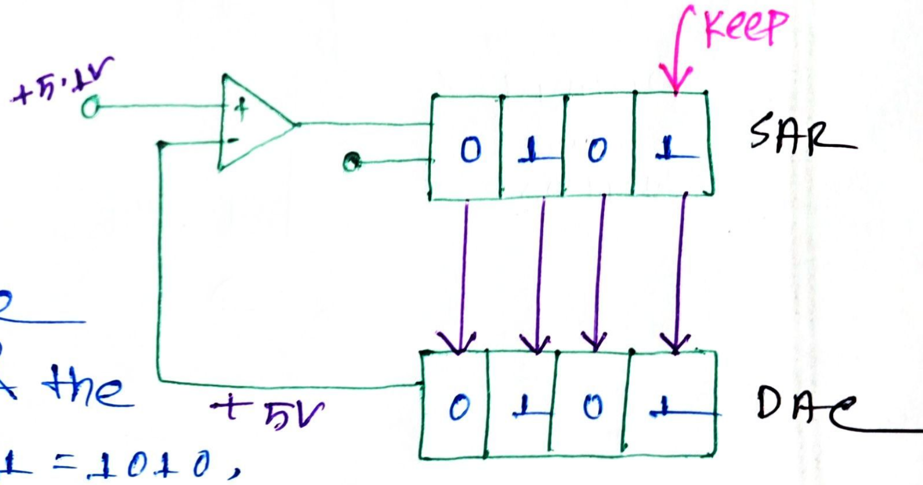
So Reset



For 4th clk-pulse:-

Vin = VDAC

5.1 = 5



So After 4 clk pulse
All bits are tested & the Digital binary for 5.1 = 1010, representing the 5V.