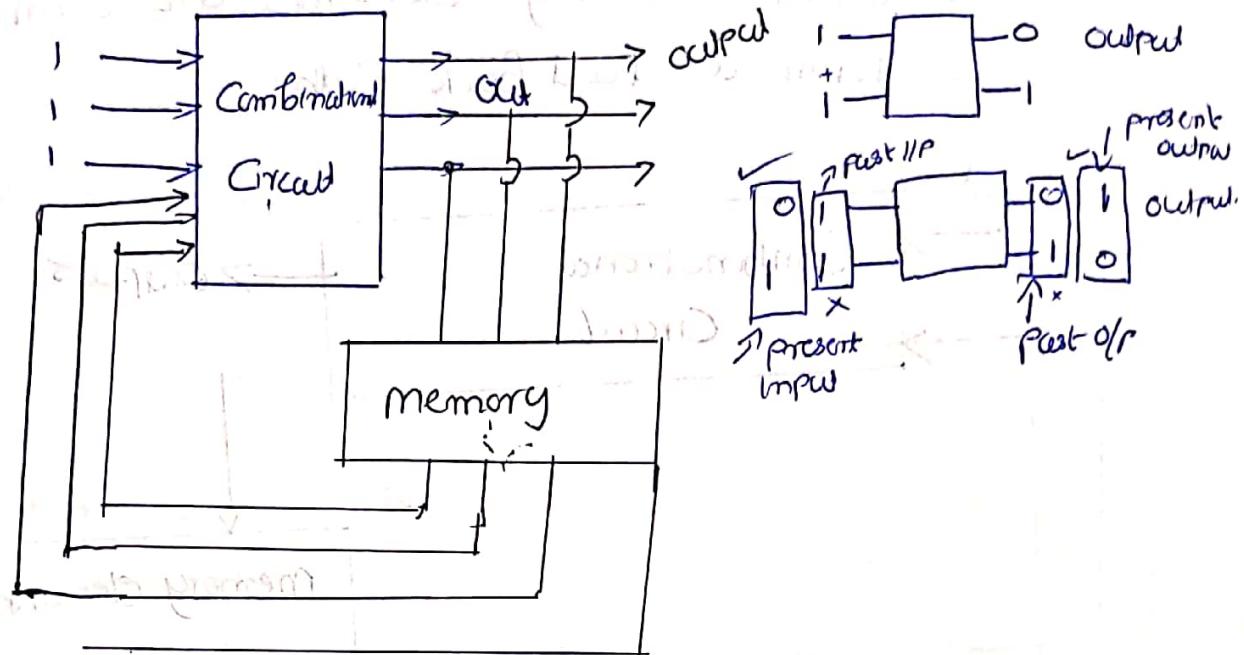


Introduction to Sequential Circuits

→ In sequential circuits, the present output depends on the present input as well as past output /outputs:



Sequential Circuits \Rightarrow required to store the previous output
 and \rightarrow they need memory to keep the output

\rightarrow Counter \rightarrow the values in the memory used as
 registers

Counters \Rightarrow past output is called feed back.

0 to 9

$$0+1 = 1$$

$$1+1 = 2$$

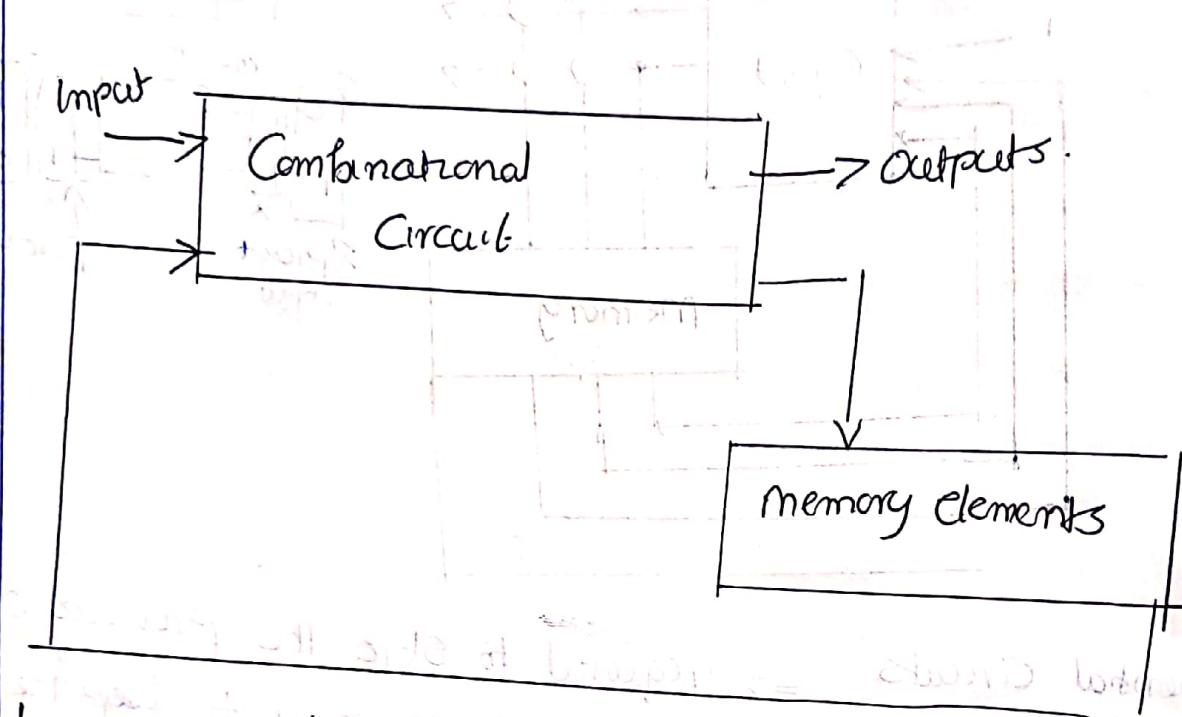
$$2+1 = 3$$

$$\text{Previous Output} + 1 = \text{Present Output}$$

\rightarrow A circuit in which output at any given time depend only on input at that time is called Combinational Circuits

→ In Sequential Circuit, the output at any given time is dependent on the input at that time as well as on their earlier outputs.

→ Sequential Circuit consists of a Combinational Circuit to which memory elements are connected to form a feed back path.



→ memory elements are devices capable of storing binary information within them.

→ The binary information stored in the memory elements at any given time defines the state of the Sequential Circuit.

→ The inputs, together with the present state of the memory elements, determine the values at output terminals.

→ The memory elements used in Clocked Sequential Circuits are called flip flops.

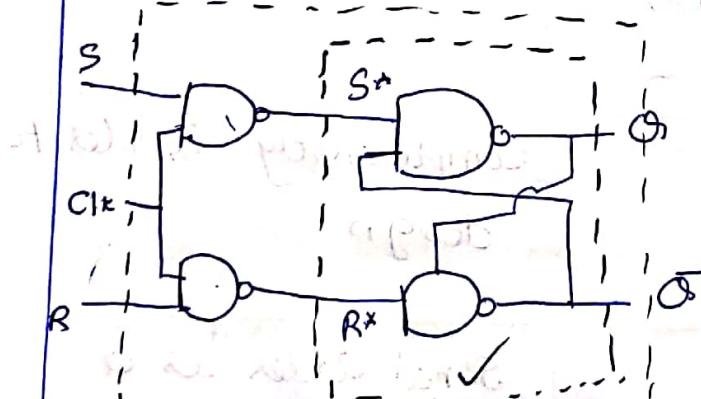
Flip Flops

- These Circuits are binary cells Capable of Storing one bit of Information
- A flip flop can maintain a binary state indefinitely, until directed by an input signal to Switch States.
- A flip flop has 2 outputs, one for normal Value and one for Complement Value of bit stored in it.
- Binary information can enter flip flop in many ways, giving rise to different types of flip flops.

UoS December 2017 (g mark)

- * Give the State table, Characteristics table, Excitation table and Characteristic equation of SR flip flop

① SR flip flops



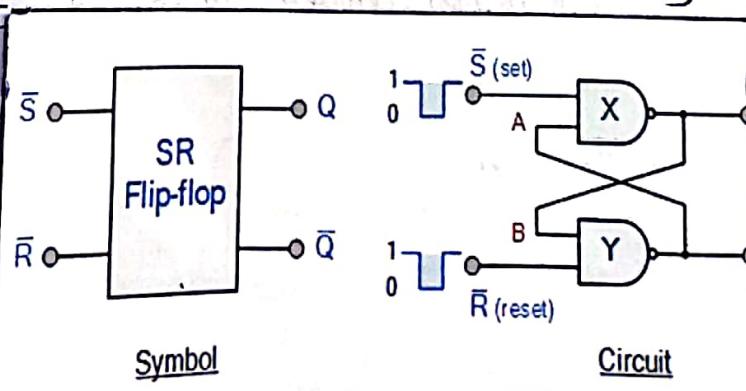
SR Latch with NOR

S*	R*	Q	Q̄
0	0	not used	
0	1	1	0
1	0	0	1
1	1	memory	

$$S^* = (S \cdot \bar{Ck}) = \bar{S} + \bar{Ck}$$

$$R^* = (\bar{R} \cdot \bar{Ck}) = \bar{R} + \bar{Ck}$$

edge triggered



Symbol

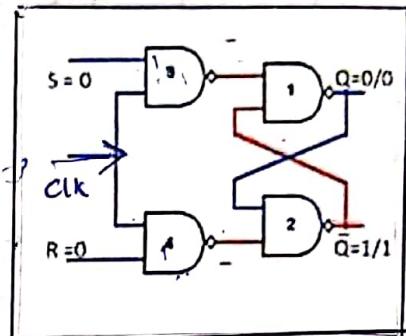
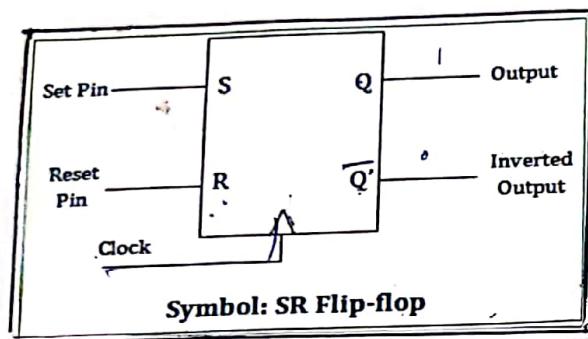
Circuit

↳ A flip flop can be constructed from 2 NAND gates, or 2 NOR gates.

↳ The cross coupled connection from the output of one gate the input of other gate constitutes a feedback path.

↳ SR flip flop has 2 inputs - Set and Reset and 2 outputs - Q and \bar{Q} .

Clocked SR- flip flops



↳ When S is 1, flip flop is set to 1.

↳ When R is 1, flip flop is reset to 0.

↳ When both S & R are 0, flip flop continues the same state.

↳ When both S & R are 1, it can be in indeterminate state.

↳ When both S & R are 1, it can be in indeterminate state.

Truth table for SR flip flop

S	R	next state
0	0	Q_{n+1}
0	1	$Q_n \rightarrow$ no change
1	0	$\bar{Q}_n \rightarrow$ Reset
1	1	$\bar{Q}_n \rightarrow$ Set
X	X	X → not allowed Concluded

Truth table for Clocked SR flip flop

CLK	S	R	Q	\bar{Q}
0	0	0	memory	memory
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1
				not used

Characteristic table & excitation table

flip flop

Truth table

Clk	S	R	Q_{n+1}	↓ next state	Q_n (Present state)
0	x	x			Q_n (Present state)
1	0	0	0		Q_n
1	0	1	0		Q_n
1	1	0	1		Q_n
1	1	1	Invalid		Q_n

Truth table

↳ A state table defines the behavior of the sequential function

↳ Truth table gives relation between inputs and output

Excitation table

↳ Used for design of flip flops and counters

↳ Truth table contains inputs and excitation table takes output as inputs.

Characteristic table

↳ Has the control input (C_{in}) as the first column, the current state as the middle column and the next state as the last column.

$Q_n \Rightarrow$ Present state

$Q_{n+1} \Rightarrow$ next state

Q_n	S	R	Q_{n+1}
Input			

Characteristic Table:

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

→ Characteristic table is determined by the truth table.

→ Truth table is the base block of these tables.

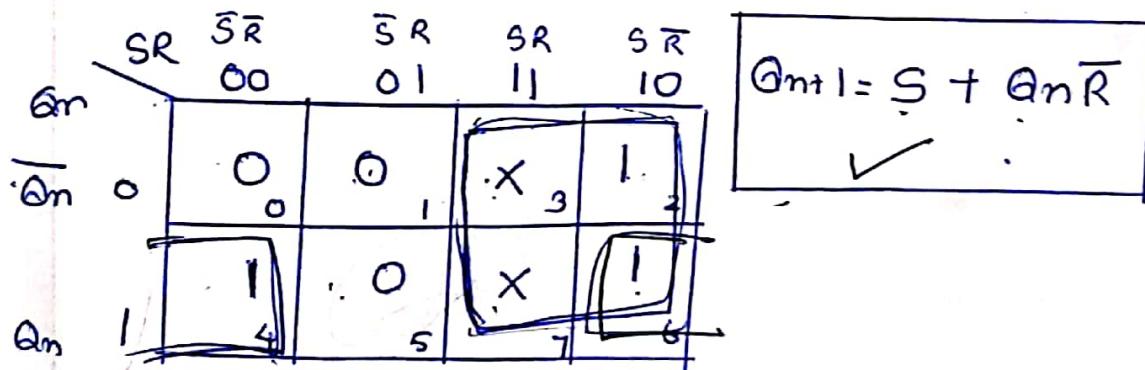
$$\begin{array}{cc} Q_n & Q_{n+1} \\ 0 & 1 \\ \hline \end{array}$$

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

→ Excitation table is determined by the truth table.

Find out the values of Q_{n+1} , then make 8 cell k-map.



Basic FlipFlop Descriptors

↳ Used in analysis

- ① Characteristic table → defines the next state of the flip-flop in terms of flip-flop inputs and current state
- ② Characteristic Equation - defines the next state of the flip-flop as a Boolean function of the flip-flop inputs and the current state.

↳ Used in design

↳ Excitation table → defines the flip-flop input variable values as function of the current state and next state.

▪ **Characteristic Table**

S	R	$Q(t+1)$	Operation
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Undefined

▪ **Characteristic Equation**

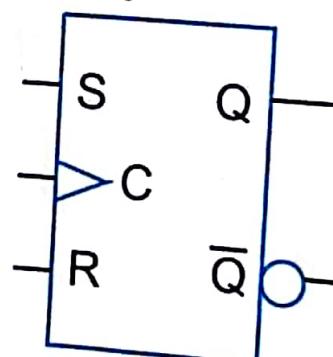
$$Q(t+1) = S + \bar{R} Q(t)$$

$S R = 0$ (S and R cannot be 1 simultaneously)

▪ **Excitation Table**

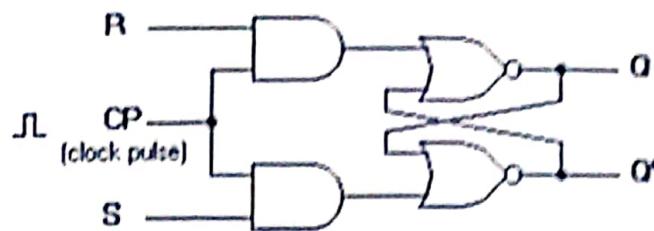
$Q(t)$	$Q(t+1)$	S	R	Operation
0	0	0	X	No change
0	1	1	0	Set
1	0	0	1	Reset
1	1	X	0	No change

Symbol



* Explain flip flop, what is clocked SR flip flop

(Circuit)



(a) Logic diagram

Q S R	Q(t+1)
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	indeterminate
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	indeterminate

(b) Truth table

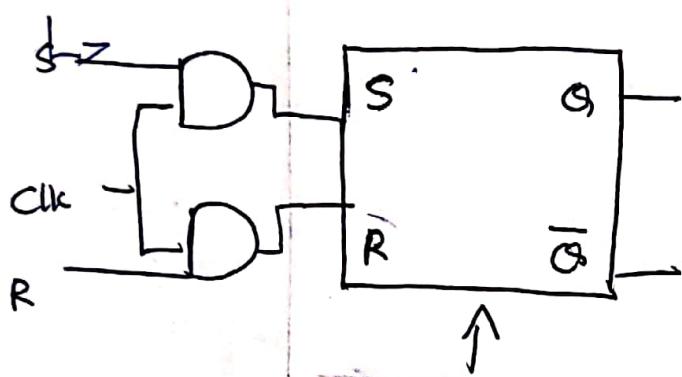
Clocked SR flip-flop

- ↳ what is flip flop?
- ↳ In digital Circuits the flip flop is a kind of bi-stable multi-vibrator.
- ↳ In sequential Circuits/ an electronic circuit which has two stable states, thereby is capable of serving as one bit of memory, bit 1 or 0.
- ↳ SR flip flop Active Low \Rightarrow NAND gates
- ↳ SR flip flop Active High \Rightarrow NOR gates.
- ↳ The most basic flip flop is called SR flip flop.
- ↳ The basic RS flip flop is an asynchronous device.
- ↳ In asynchronous device, the outputs is immediately more if the inputs change changed any time one or just as in Combinational logic circuit

→ It does not operate in step with a clock or timing
Clocked SR flip flop

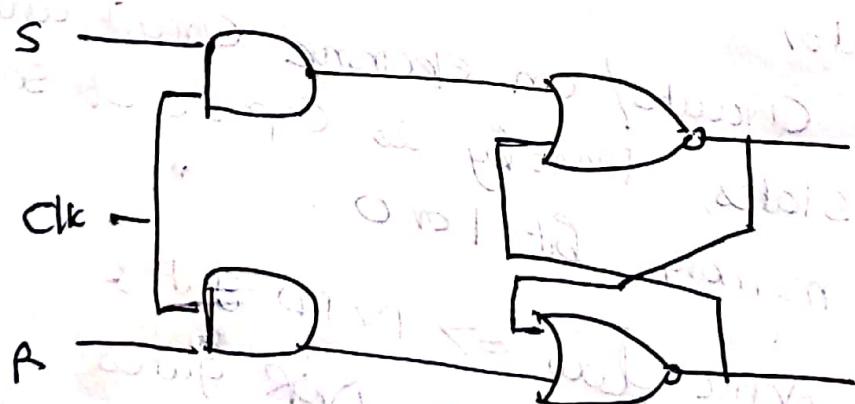
→ Additional clock input is added to change the
SR flip flop from an element used in asynchronous
sequential circuits to one, which can be used in
synchronous circuits

→ It means that the flip flop can change the
output states only when clock signal makes
a transition from Low to High.



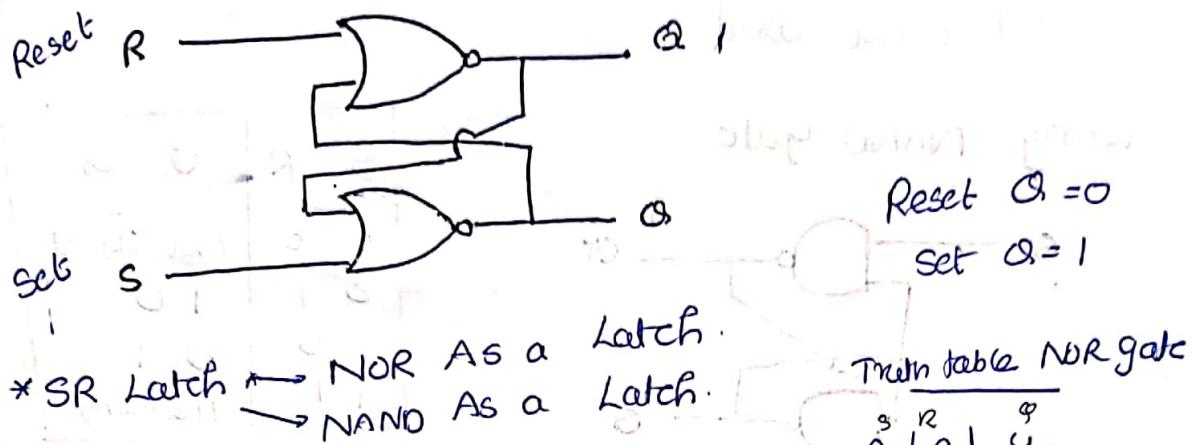
It uses NOR gate, most used

And gate in front



SR Latch

↳ The basic storage element is called LATCH. As the name suggests it latches "0" or "1".



Truth table NOR gate

S	R	Y
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Case 1

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

↳ If any output 0, Input high!

↳ $R=1, Q=0$

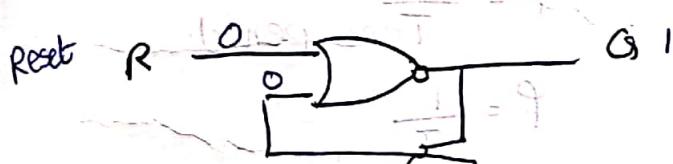
↳ When 2 input 0 \rightarrow Output 1 \rightarrow $Q=1, \bar{Q}=0$

↳ When 2 input 1 \rightarrow Output 0 \rightarrow $Q=0, \bar{Q}=1$

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

Case 2

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



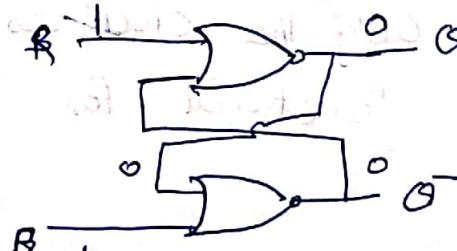
$$S=0, R=0, Q=1, \bar{Q}=0$$

\rightarrow memory



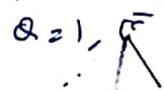
Case 3

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



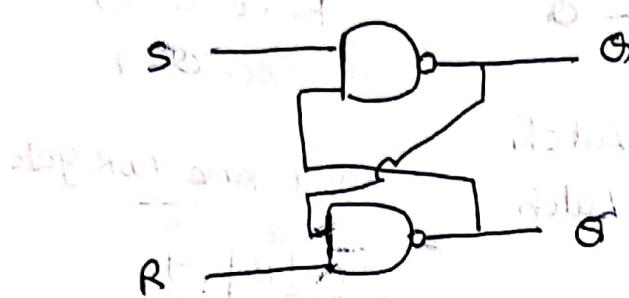
$S=1, R=1$ is not used.

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



S	R	Q	\bar{Q}
0	0	Memory	
0	1	0	1
1	0	1	0
1	1	not used.	

using NANO gate.



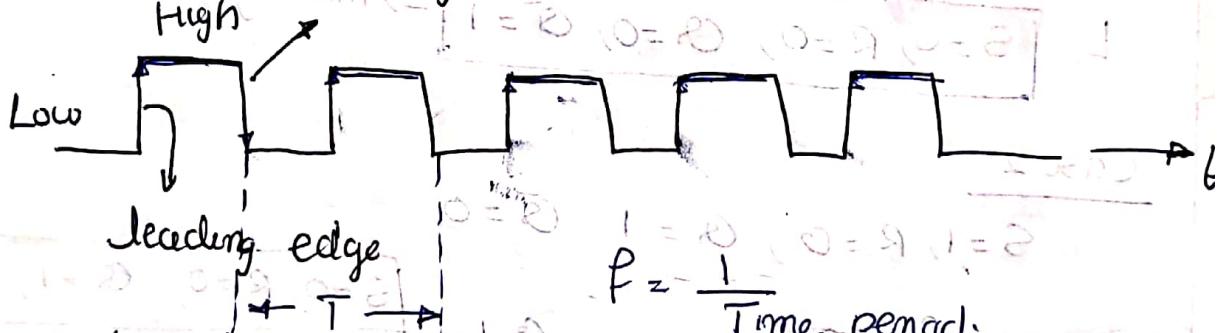
S	R	Q	\bar{Q}
0	0	Not Caged	
0	1	1	0
1	0	0	1
1	1	memory	

9

What is Clock.

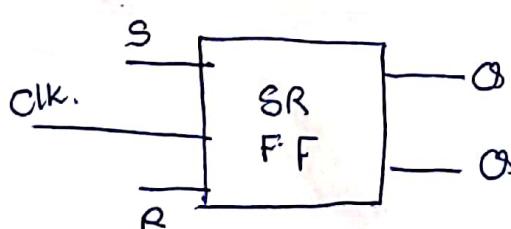
- It decides the time of inputs
- Clock is a signal that goes to low to high, high to low.

High



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

Time period.



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

- In this SR flip flop, it will work only the clock is high. It will be functional for high pulses.

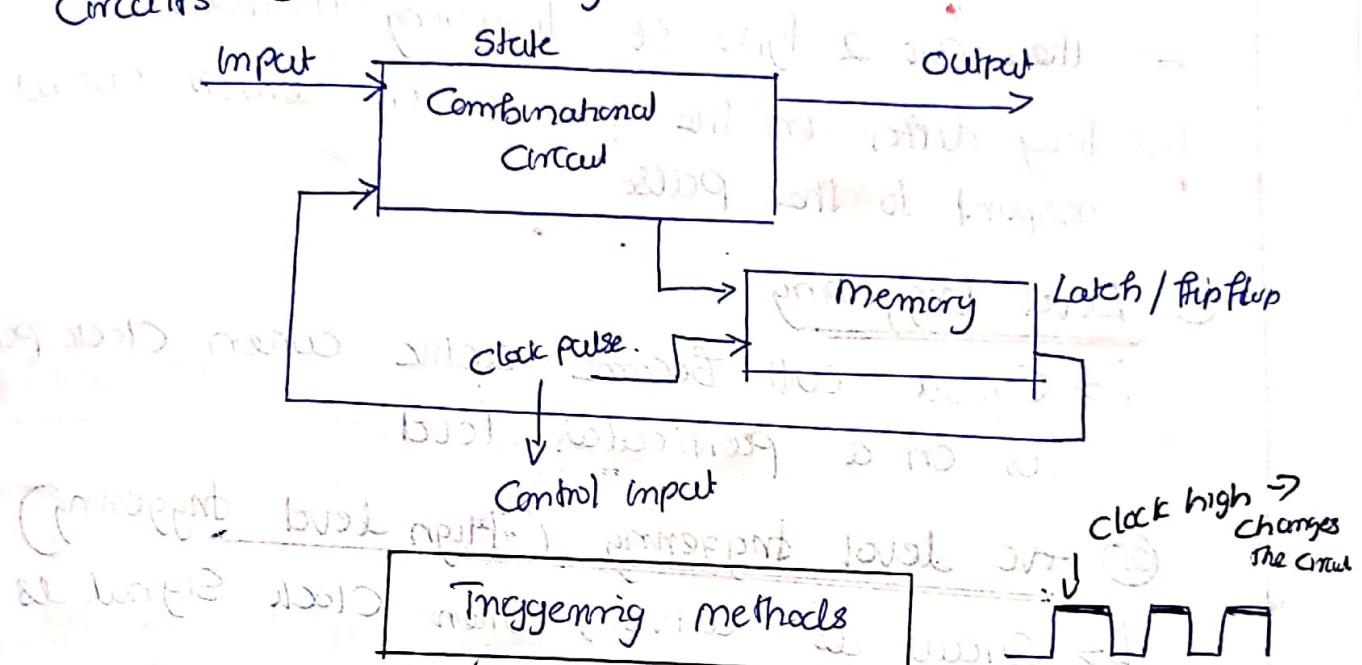
Duty cycle :- Ratio of time $\frac{\text{Signal} \uparrow}{\text{total time}}$

$$= \frac{t/2}{T}$$

$$= \frac{1/2}{1} \Rightarrow \text{duty cycle for clock } 50\%$$

Q) Explain Triggering methods?

\hookrightarrow sequential Circuits are the Combinational Circuits with memory



Combinational logic

Sequential logic

Controlled by clock

Triggering methods

Clock high \Rightarrow changes the circuit

Edge Triggering

Level Triggering

+ve edge triggering

-ve edge triggering



Low \rightarrow high
Changes happens



\rightarrow high to low
 \rightarrow there is a transition in memory elements

- ↳ Triggering means making a circuit active.
- ↳ i.e., allowing a circuit to take input & give output.
- ↳ When circuit is not triggered, even if there is input data, there will be no change in data stored or output.

- ↳ Triggering is given in the form of a clock pulse.
- ↳ There are 2 types of triggering methods.
- ↳ They differ in the manner in which circuit respond to the pulse.

① Level triggering

- ↳ Circuit will become active when clock pulse is on a particular level.

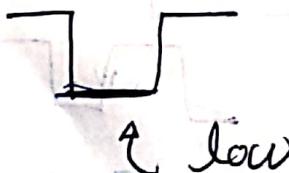
(a) +ve level triggering (High level triggering)

- ↳ Circuit is active, when clock signal is high (or 1)

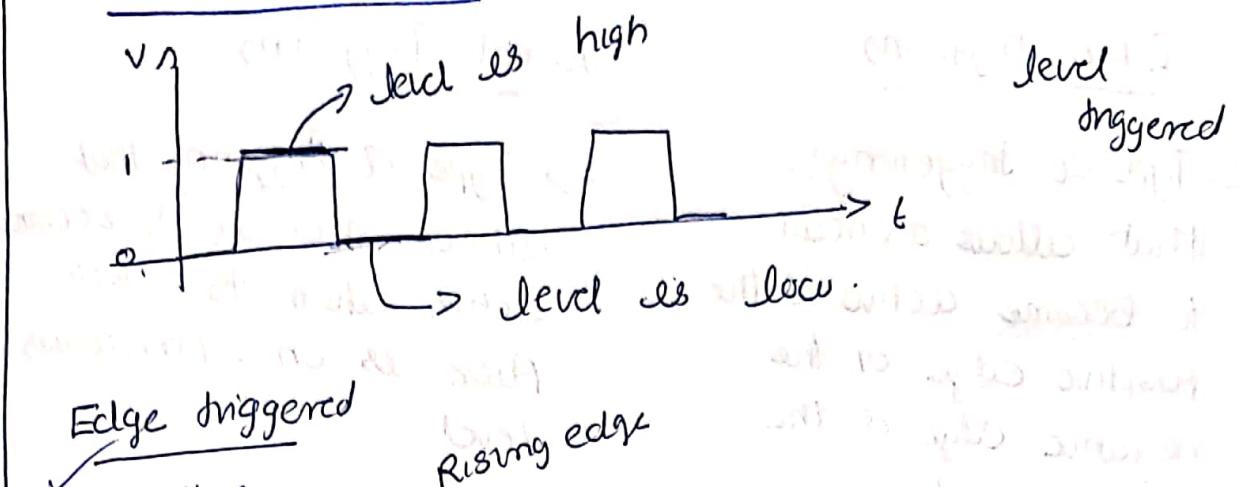


(b) -ve level triggering

- ↳ Circuit is active, when clock signal is low (or 0)



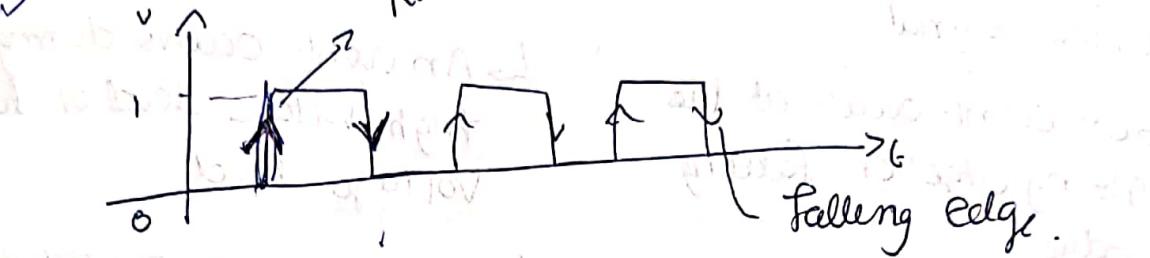
Example of level triggering



Edge triggered

Rising edge

level triggered



Double → trigger bus 2

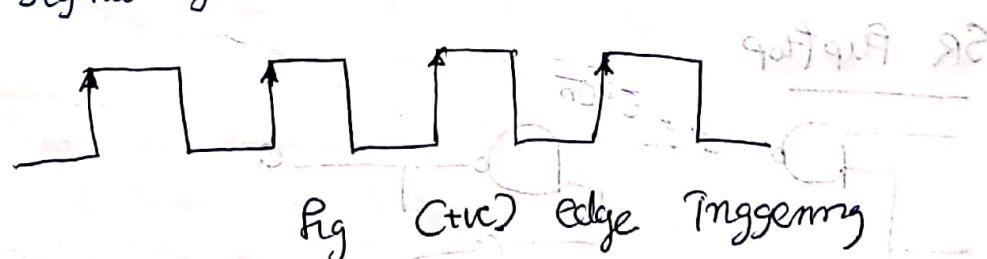
single → trigger output

② Edge triggering

↳ Circuit become active at the +ve or -ve edge of clock signal.

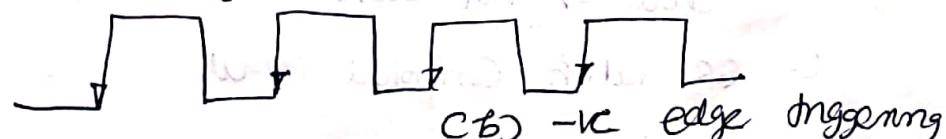
@ +ve Edge triggering

↳ Circuit take input exactly at time in which clock signal goes from low(C0) to high(C1).



C0 - ve edge triggering

↳ Circuit take input exactly at time in which clock signal goes from high(C1) to low(C0)



Example Difference

Edge Triggers

↳ Type of triggering that allows a circuit to become active at the positive edge or the negative edge of the clock signal.

↳ An event occurs at the rising edge or falling edge.

↳ Edge triggered - flipflops

Level Triggers

↳ Type of triggering that allows a circuit to become active when the clock pulse is on a particular level.

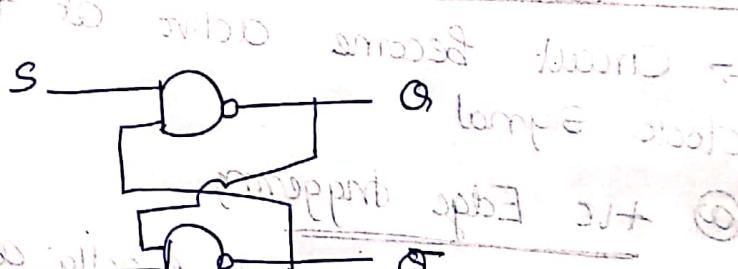
↳ An event occurs during a high voltage level or low voltage level.

↳ Level triggered → Latches.

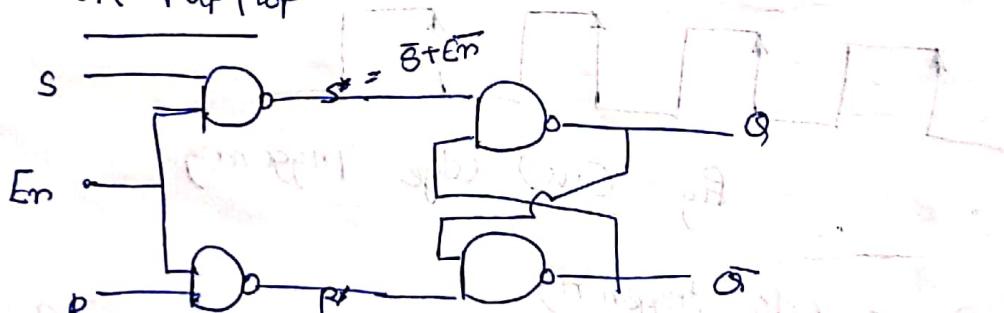
⑨

Difference between Latch and Flip flop.

SR latch



SR flip flop



Enable is high \Rightarrow work

low \Rightarrow not work

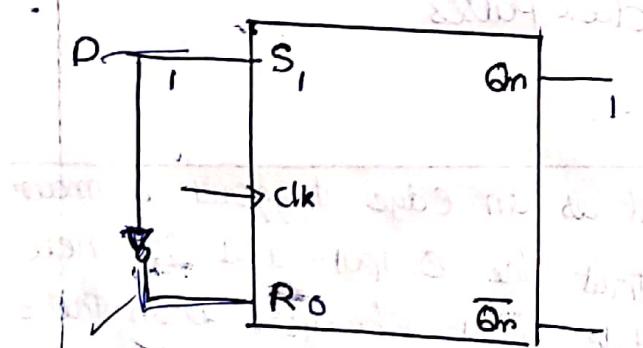
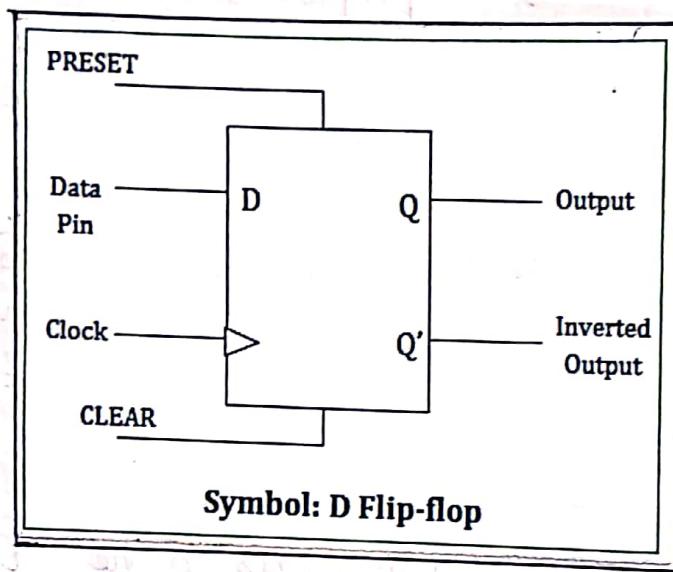
↳ SR latch Controlled input

↳ latch is level sensitive, whenever there is input, it checks for the input.

Latches	Flip flops
① Latches are building blocks of sequential circuits & these can be built from logic gates.	Flip flops are also building blocks of sequential circuits, but these can be built from the latches.
② Latch continuously checks its inputs and changes its output correspondingly	Flip flop continuously checks its inputs and changes its output correspondingly only at times determined by clocking signal
③ The latch is sensitive to the duration of the pulse & can send or receive the data when the switch is on	Flip flop is sensitive to a signal change, they can transfer data only at the single instant & data cannot be changed until the next signal change. Flip flops are used as a register
④ It is based on the enable function	It works on the basis of clock pulses
⑤ It is level triggered, it means that the output of the present state and input of the next state depends on the level that is binary input 1 or 0	If it is an edge triggered, it means that the output and the next state input changes when there is a change in clock pulse, whether it may +ve/-ve clock pulse.

Introduction to D Flip flop

- ↳ Also known as Data flip flop.
- ↳ Can be constructed from RS flip flop or JK flip flop by addition of an inverter.
- ↳ Inverter is connected so that the R input is always the inverse of S (or J input is always complementary of Ic)
- ↳ The D flip flop will act as a storage element for a single binary digit (bit).



Truth table of SR flip flop

Clk	S	R	Qn+1
0	x	x	Qn
1	0	0	Qn
1	0	1	Q
1	1	0	1
1	1	1	invalid

↳ If $S=0, R=1$, then get 0

↳ If $S=1, R=0$, then get 1

S & R are always complementary to each other.

→ To convert SR flip flop to D flip flop, use a inverter and give the output of the inverter to the R.

Truth table of D flip flop

D	Q_{n+1}
0	0
1	1

Using Clock

Clk	D	Q_{n+1}
0	x	Q_n
1	0	0 (S=0, R=1, then 0)
1	1	1

Characteristic table & Excitation table for D flip flop

→ D flip flop is the modification of SR flip flop.

→ Only one input is there → D line.

→ D is connected to S input and its complement to R input.

→ D FF is basically SR flip flop with inverter in R.

Truth table

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

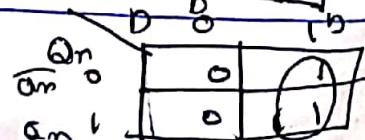
Characteristic table

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

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

Excitation Table

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



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

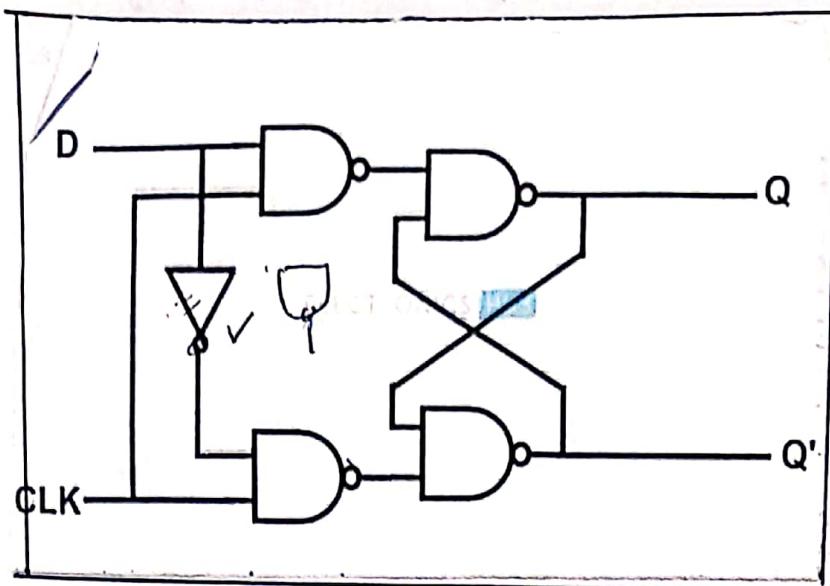


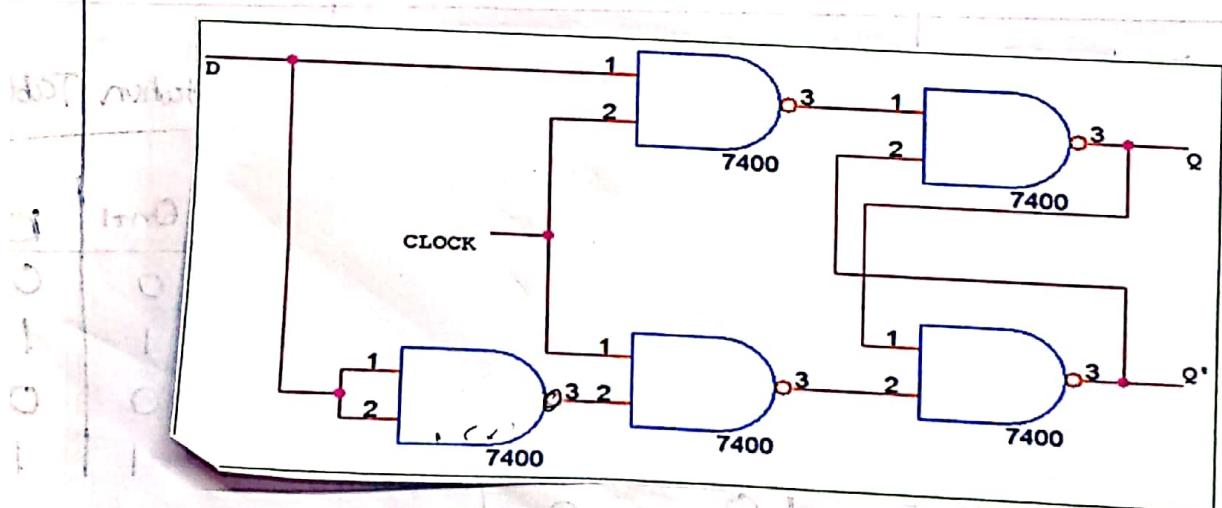
fig. - Clocked D- flip flop

* D Flip flop using NAND gate

↳ A D flip flop really is a SR flip flop, which is a set reset flip flop.

↳ The only difference is that it has an added NOT gate in front of its Q output.

↳ This NOT gate prevents the hold condition and the indeterminate condition of the SR flip flop from occurring.



Introduction to JK flip flop

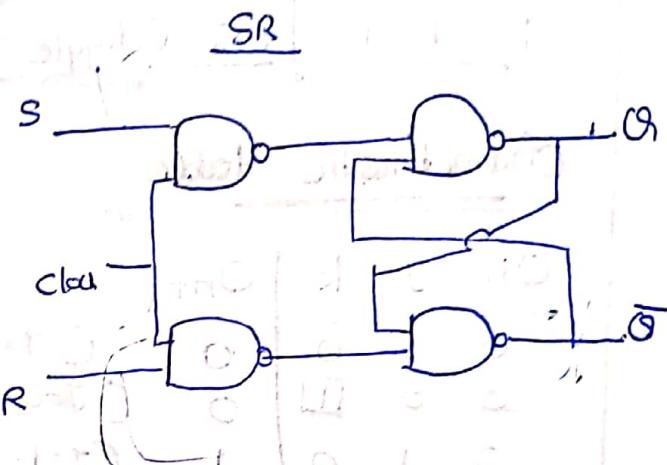
↳ A JK flip flop is a refinement of the RS flip flop

↳ The behavior of inputs J and K is same as the S and R inputs of the SR-Flipflop

↳ The letter J stands for SET and the letter K stands for CLEAR

Truth table for SR FF

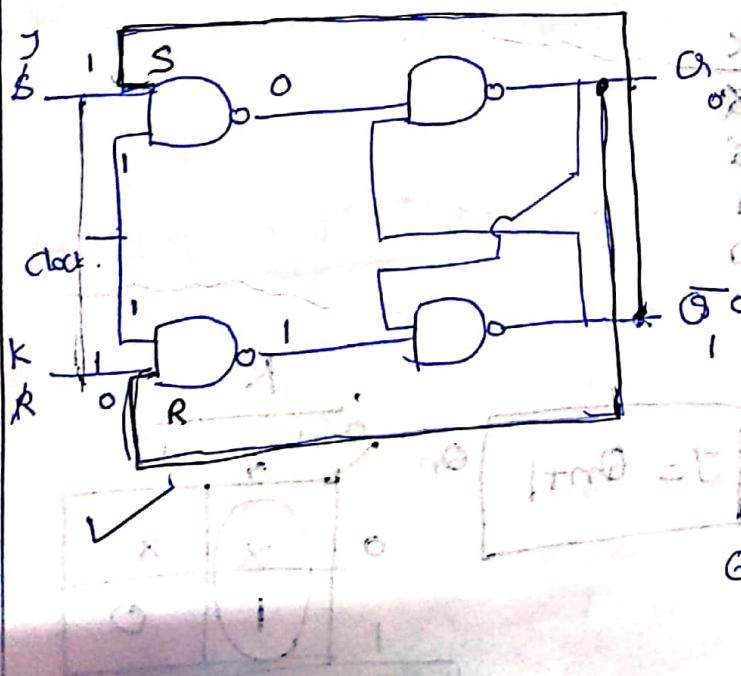
Clk	S	R	Q_{n+1}
0	x	x	Q_n (memory)
1	0	0	Q_n (memory)
1	0	1	0 (Set)
1	1	0	1 (Clear, also R)
1	1	1	not used.



↳ Last Combination $S=1, R=1$ is not used

↳ In this, we are trying that State $S=1, R=1$ will be used that is known as JK flip flop

Design of JK flipflop



Truth table

$Clk=0$	$Clk=1$	Q_{n+1}
0	0	$Clk=0, Q=1, \bar{Q}=0$
0	1	$Clk=1, J=1, K=0, Q=1, \bar{Q}=0$
1	0	$Clk=1, J=0, K=1, Q=0, \bar{Q}=1$
1	1	$Clk=1, J=1, K=1$
assume $Q=0, \bar{Q}=1$		
$Q_2 = 0, 1, 0, 1$		
$\bar{Q}_2 = 1, 0, 1, 0$		
$Q_{n+1} = ?$ (Complement of the previous state)		

Characteristic and Excitation table for JK F.F.

Truth table:-

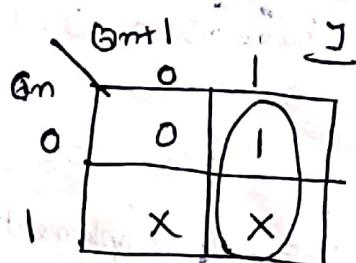
Clk	S	R	J	K	Q _n	Q _{n+1} next state.
0	x	x			Q _n	Q _n previous state
1	0	0	0	0	Q _n	memory
1	0	1	0	1	Q _n	Q _{n+1}
1	1	0	1	0	Q _n	Q _{n+1}
1	1	1	1	1	Q _n	(Toggle)

Characteristic table

Q _n	J	K	Q _{n+1}	Condition	Condition	Condition
0	0	0	0	(J=0, K=0, Q _{n+1} =0)	Q _n = 0	Q _{n+1} = 0
0	0	1	0	(J=0, K=1, Q _{n+1} =0)	Q _n = 0	Q _{n+1} = 1
0	1	0	1	(J=1, K=0, Q _{n+1} =1)	Q _n = 0	Q _{n+1} = 1
0	1	1	1	Complement of Q _n (Q _n =0, Q _{n+1} =1)	Q _n = 0	Q _{n+1} = 1
1	0	0	1	state transition from 0 to 1	Q _n = 0	Q _{n+1} = 1
1	0	1	0	state transition from 1 to 0	Q _n = 1	Q _{n+1} = 0
1	1	0	1	state transition from 1 to 1	Q _n = 1	Q _{n+1} = 1
1	1	1	0	(Toggle State)	Q _n = 1	Q _{n+1} = 0

Excitation table

Q _n	Q _{n+1}	J	K
0	0	0	*
0	1	1	*
1	0	*	1
1	1	X	0



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

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

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

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

$$\underline{Q_{n+1}}$$

8 cell k-map

		JK	$\bar{J}K$	JK	$\bar{J}K$	JK	$\bar{J}K$
		00	01	11	10		
		\bar{Q}_n	0	0	1	1	
\bar{Q}_n	0	1	0	0	1		
Q _n	1						

$$\underline{Q_{n+1}}$$

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

* What is the Race Condition in flipflops.

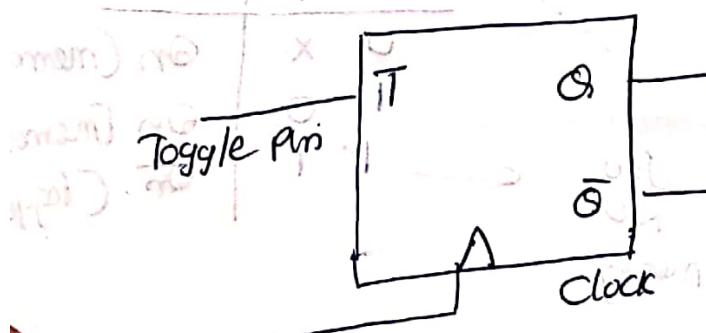
↳ When the S and R inputs of an SR flip flop is at logical 1, then the output becomes constant and it is known as race condition.

Race Around Condition of JK flip flop.

↳ In JK flip flop, as long as clock is high for the input conditions J & K equals to the output changes or complements its output from $1 \rightarrow 0$ & $0 \rightarrow 1$. This is called toggling output or uncontrolled changing or racing condition.

Explaining T-flip flop.

↳ Toggle flip flop.

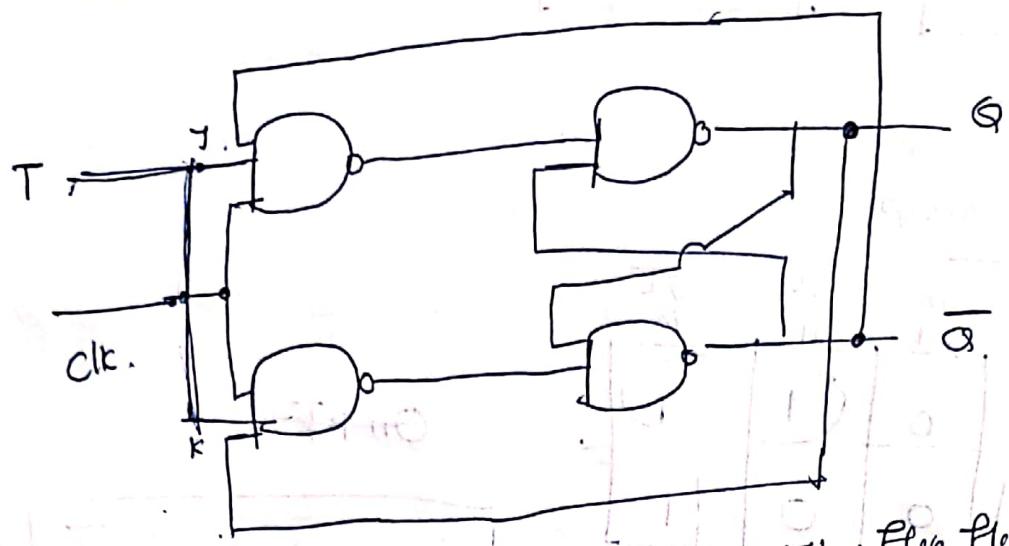


Input

Inverted Input

Symbol T FlipFlop

T- flip flop

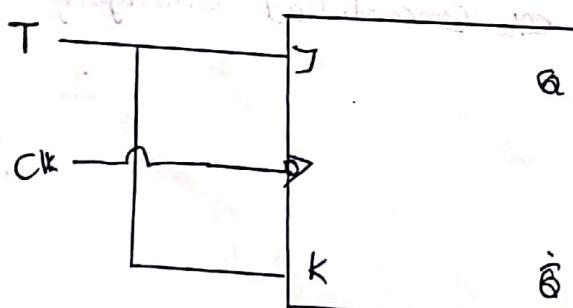


→ Single Input Version of the JK-Flip Flop,
formed by tying both inputs together.

Truth table

$Q_{(t)}$	T	$Q_{(t+1)}$
0	0	0
0	1	1

→ T- stands for toggle.



Truth table for T

Flip Flop

Clk	T	$Q_{(t+1)}$
0	X	On (memory)
1	0	On (memory)
1	1	Off (toggle)

* Characteristic & Excitation Table for T Flip Flop

Truth Table :-

CLK	T	Q_{n+1}
0	X	$Q_n \}$ memory
1	0	$\overline{Q_n} \}$ toggle
1	1	$\overline{Q_n} \}$ toggle
0	1	$\overline{Q_n} \}$ Complement of previous state. $Q_n = \overline{Q_n}$

Characteristic table

present state	Q_n	T	Q_{n+1}	next state
0	0	0	0	0
0	0	1	1	1
1	0	0	1	1
1	1	1	0	0

$T=1$
then Q_n
 $Q_n=0, \overline{Q_n}=1$

$T=0 = Q_n$
previous state = 0

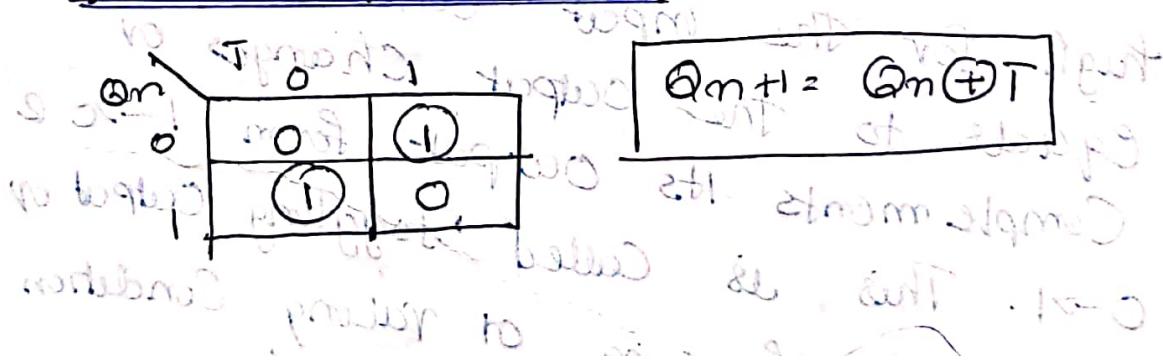
Excitation Table.

↳ Calculated from Characteristic table.

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

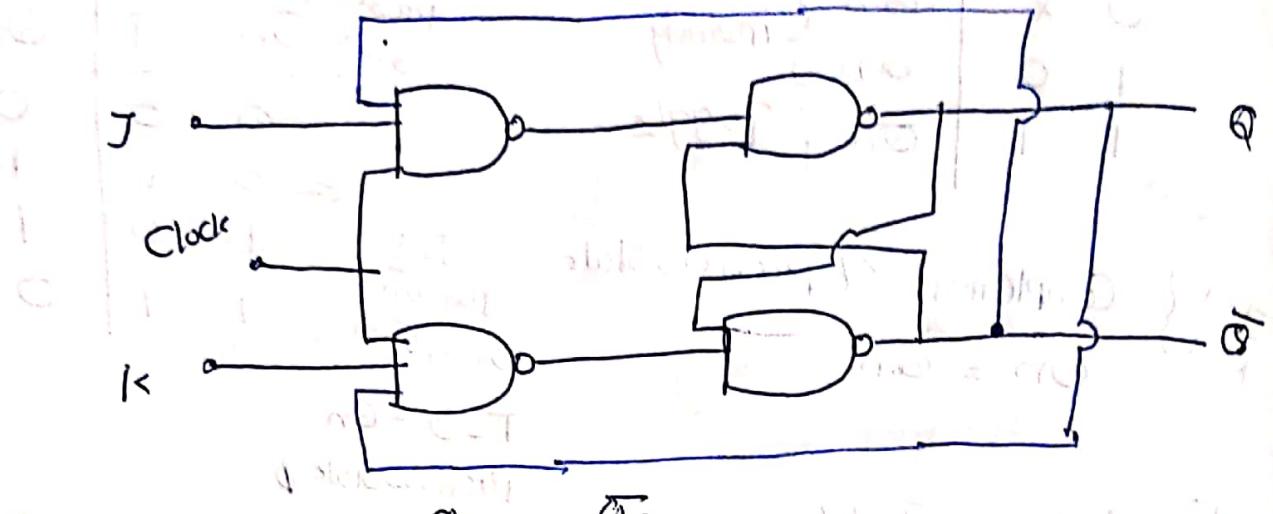
00	0	1	0
00	0	0	1
01	1	0	1
10	1	1	1
11	1	1	0

Characteristic equation of Q_{n+1} if Q_n and T



Race Around Condition in JK Flip Flop (Q mark)

JK Flipflop



Clock	J	K	Q _{n+1}	Q̄ _{n+1}	Output conditions		
			Q _{n+1}	Q̄ _{n+1}	T	Time	Q̄
0	x	x	Q _n	Q̄ _n			
1	0	0	Q _n	Q̄ _n	1	Time	0
1	0	1	0	1	0	0	0
1	1	0	1	0	1	0	0
1	1	1	?	?	0	1	1

→ In JK flip flop, as long as clock is high for the input conditions J & K equals to the output changes or complements its output form 0 → 1. This is called controlled changing of racing condition.

Racing of JK flip flop

↳ Steps to avoid Racing Condition in JK flipflop.

① If the clock on or high time is less than the propagation delay of the flip flop then racing can be avoided. This is done by using edge triggering rather than level triggering.

② If the flip flop is made to toggle over one clock period then racing can be avoided. This is introduced in the concept of master-slave JK flip flop.

Conditions to Over Come Racing:-

① $T/2 <$ propagation delay of Flip Flop

② Edge Triggering with suitable width

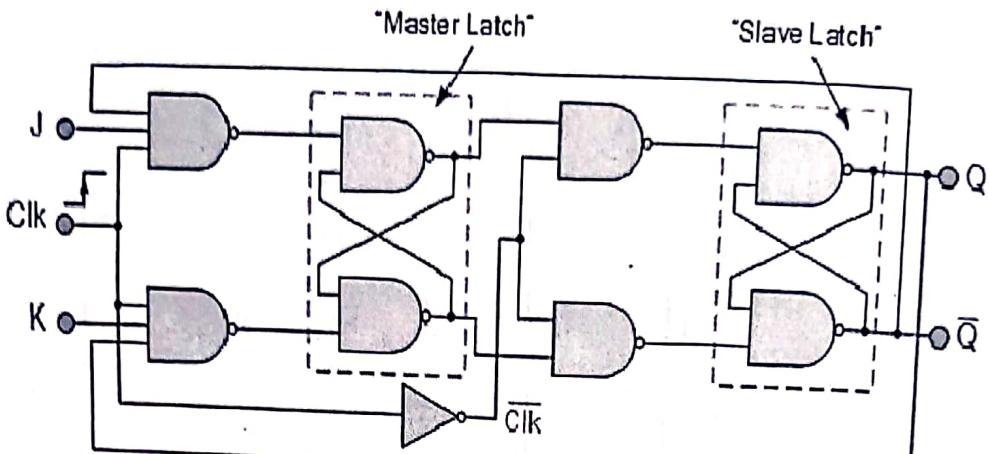
③ Master Slave JK flip flop.

Race Avoided

↳ Toggling of the output before the occurrence of next clock pulse

Master-Slave JK Flip Flop

MASTER-SLAVE



- ↳ The master-slave flip flop eliminates all the timing problems by using two SR FF connected together in a series configuration.
- ↳ One flip flop acts as the master circuit, which triggers on the leading edge of the clock pulse while the other acts as the slave circuit, which triggers on the falling edge of the clock pulse. This results in the two sections being enabled during opposite half-cycles of the clock signal.

Case 1

- ↳ When clock is not given, both master & slave are inactive and there is no change in outputs.

Case 2

For $\text{Clock} = 1$, master is active, slave inactive. As $J = K = 0$, output for master is Q and \bar{Q} will not change.

As soon as clock goes to 0, slave becomes active, and master inactive but since input to slave S and R is same, output of slave will also remain same.

Truth table of Master Slave Jk FF

Case	Inputs			Outputs		Remarks
	CLK	J	K	Q_{n+1}	\bar{Q}_{n+1}	
I	X	0	0	Q_n	\bar{Q}_n	No Change
II	$\neg \neg C_0$	0	0	Q_n	\bar{Q}_n	No Change
III	$\neg \neg C_0$	0	1	0	1	Reset
IV	$\neg \neg C_0$	1	0	1	0	Set
V	$\neg \neg C_0$	1	1	Q_n	\bar{Q}_n	Toggle

Case 3

$Clk(C_0), J=0, K=1$, Output will be Reset

Case 4

$Clk(C_0), J=1, K=0$ Output will be Set

Case 5 ~~Initial State of outputs after which~~
 $\neg \neg J=1, \neg \neg K=1, Clk=1$, master output will toggle - so S and R will invert but slave remains inactive all the times since clock is 0

State Table

- ↳ The time sequence of I/Ps, O/Ps and flip-flops states are represented in a table called State table.
- ↳ It consists of 3 sections labeled present state, next state & output.
- ↳ Present state shows States of FF before the occurrence of Clock pulse.
- ↳ Next state shows state of FF after the application of Clock pulse.
- ↳ Output lists the values of O/P variables during the present state.

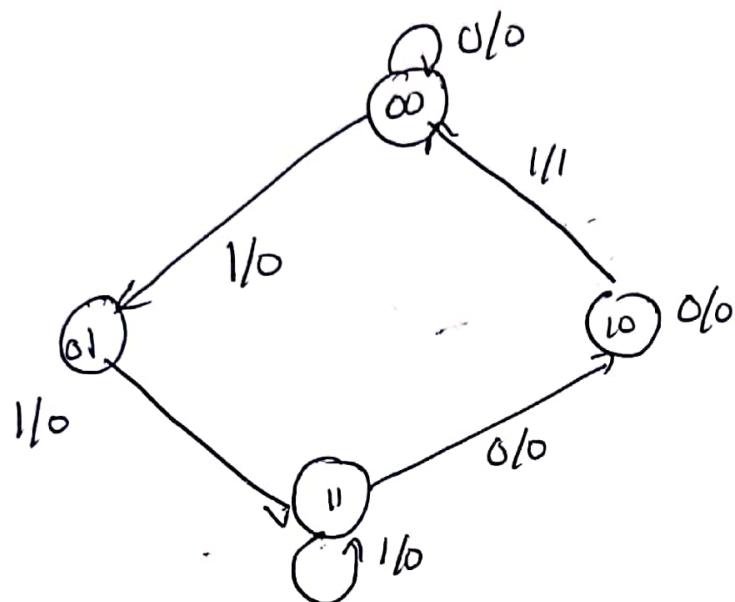
State table for the example.

Present State	Next State		Output	
AB	AB	AB	$x=0$	$x=1$
00	00	00	g	g
01	01	01	0	0
10	10	10	0	0
11	11	11	0	1

State diagram

- ↳ Information available in State table can be represented graphically in a State diagram

↳ A state is represented by a circle and to monitor the states as indicated by directed lines by connecting the circles



State equations

↳ State equation is an algebraic expression that specifies the condition for a flip flop state transition.

$$A(t+1) = (\bar{A}B + A\bar{B} + AB)x + AB^c \bar{x}$$

	00	01	11	10
00	1			
01				
11				
10				

	00	01	11	10
0	1	1	1	1
1				

$$\begin{aligned}
 A(t+1) &= Bx + (B+x)A \\
 &= Bx + (Bx)^T A \\
 &= S + RA
 \end{aligned}$$

$$\begin{aligned}
 B(t+1) &= \bar{A}x + (\bar{A}+x)B \\
 &\Rightarrow \bar{A}x + (\bar{A}x)^T B \\
 &\Rightarrow S^T R B
 \end{aligned}$$

Flip Flop Conversions.

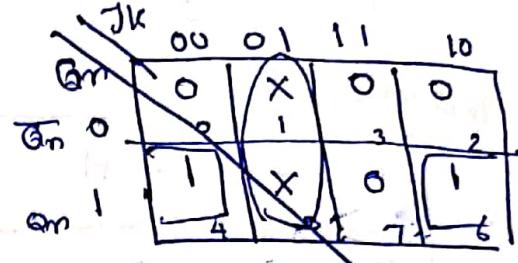
↳ Find the truth table for destination flip flop, find the excitation table for source flip flop.

Truth table of JK

J	K	Q_m	Q_{m+1}
0	0	X	$\overline{Q_m}$
0	1	X	0 - Reset
1	0	X	1 - Set
1	1	X	$Q_m \Rightarrow \text{doubt}$

$x = 0, 1$

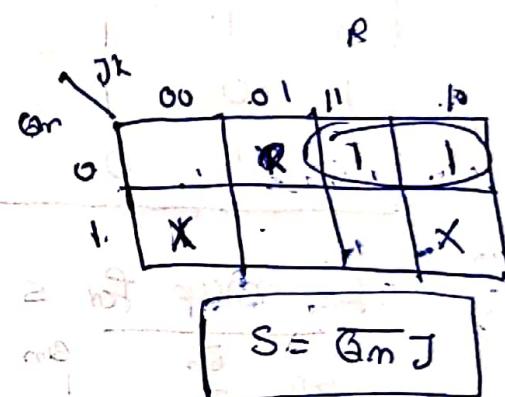
k-map of S



Excitation table of SR

Q_m	Q_{m+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

PS.

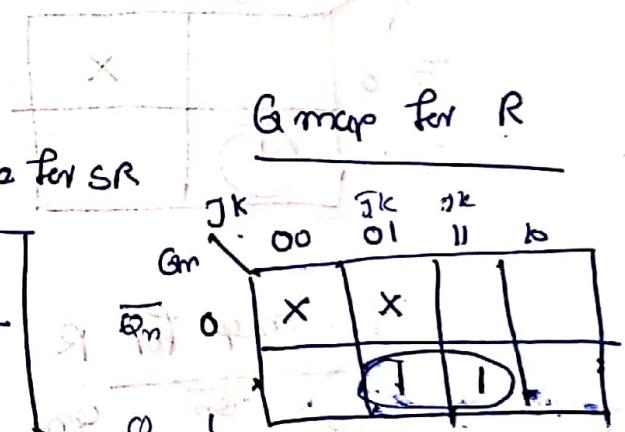


SR flip flop to JK FF

Truth table for JK

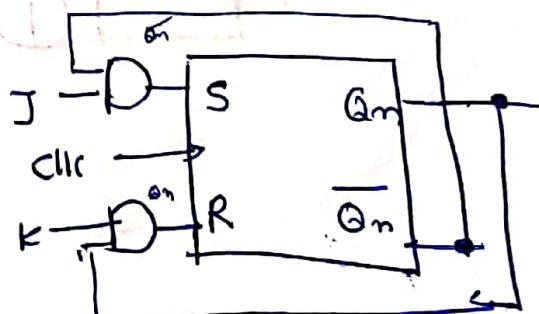
J	K	Q_m	Q_{m+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	1	0	0	X
0	1	1	1	0	1
1	0	1	0	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	0	0	0	1
1	1	1	1	X	0

Excitation table for SR



$G \text{ map for } R$

$\Rightarrow R = Q_m K$



⑦ SR flipflop to T flip flop

Truth table of T

T	Q_n	Q_{n+1}
0	x	\bar{Q}_n
1	x	Q_n

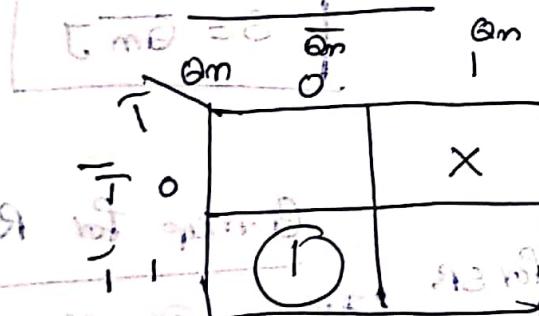
Excitation table of SR

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

Truth table of T

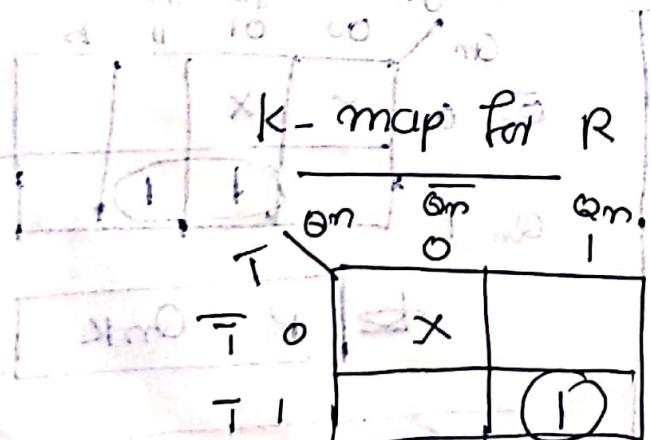
T	Q_n	Q_{n+1}	S	R
0	0	0	0	x
0	1	1	x	0
1	0	1	1	0

K-map for S



$$S = \overline{T} \cdot \overline{Q}_n + T \cdot \overline{Q}_n$$

K-map for R



$$R = T \cdot Q_n$$

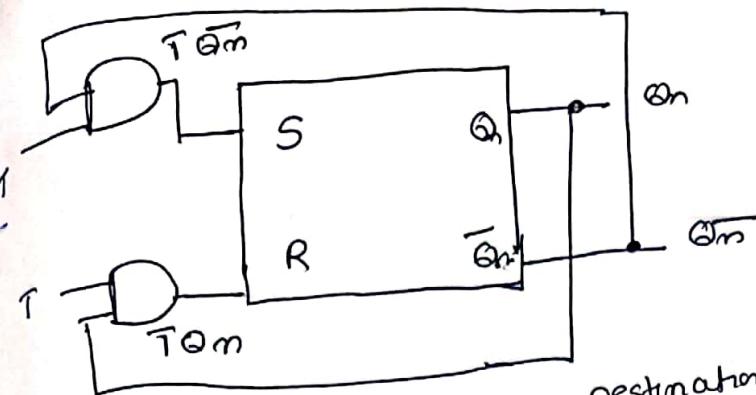


Fig:- SR Flip Flop ~~Conversion~~
to T Flip Flops.

② Jk flip flop To T Conversion

source \rightarrow destination

Truth table of T

T	Q _n	Q _{n+1}
0	x	Q _n
1	x	Q _n

Excitation table of J_k.

Q _n	Q _{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Extended Truth table

Truth table

T	Q _n	Q _{n+1}	J	K
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

excitation

T	Q _n	Q _{n+1}
0	0	0
0	1	1
1	0	1
1	1	1

K-map of J

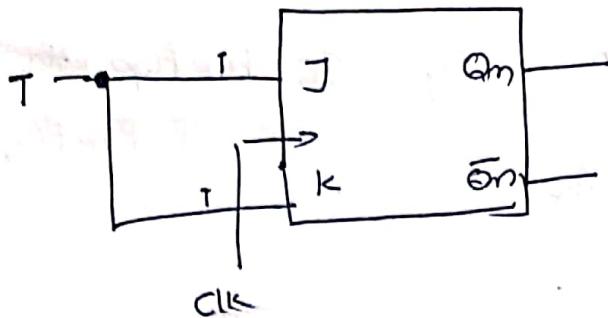
T	Q _n	Q _{n+1}
0	0	x
1	1	x

$$J = T$$

K-map of K

T	Q _n	Q _{n+1}
0	0	x
1	1	1

$$K = \bar{T}$$



JK FF To D Flip Flop

Truth table of D

D	Q _n	Q _{n+1}
0	X	0
1	X	1

Excitation table of JK

Q _n Q _{n+1}	J	K
0 0	0	X
0 1	1	X
1 0	X	1
1 1	X	0

Extended Truth table

D	Q _n	Q _{n+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

k-map for J

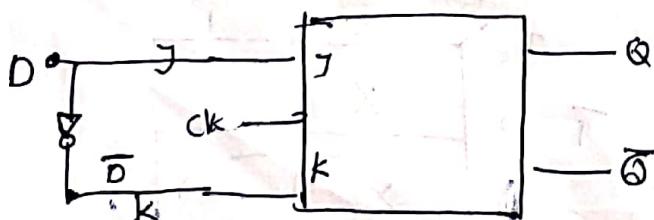
Q _n	0	1
0	0	X
1	X	0

$$J = D$$

k-map for K

D	0	1
0	0	X
1	X	0

$$K = \bar{D}$$



T-Flip flop to D- flip flop Conversion

Step 1
Truth table of D

D	Qn	Qn+1
0	x	0
1	x	1

Excitation table for T

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

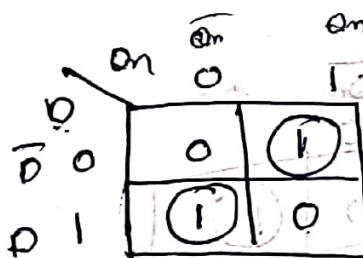
Extended Truth table

D	Qn	Qn+1	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

D	Qn	Qn+1	T
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	0

Step 2

k-map for T

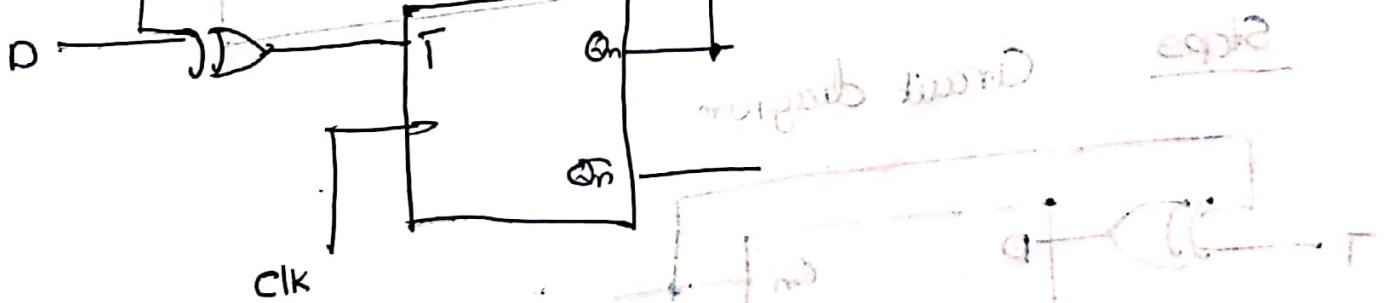


$$\bar{A}B + A\bar{B} = A \oplus B$$

$$T = \overline{D} \overline{Qn} + D \overline{Qn}$$

$$= D \oplus Qn$$

$$\bar{D}T + \bar{D}\bar{Qn}T = D \oplus Qn$$



D Flipflops to T- flipflop

Step 1

Truth table of T

T	Q _n	Q _{n+1}
0	X	Q _n
1	X	Q _n

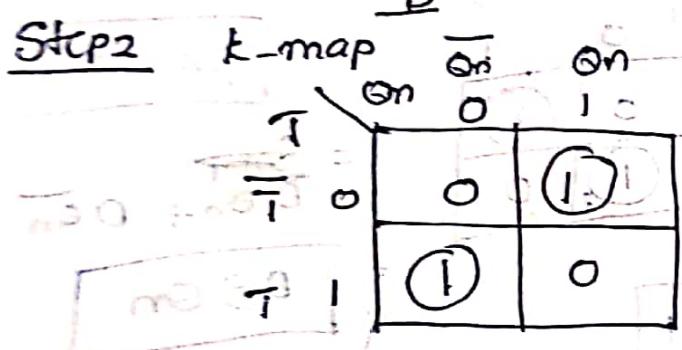
Excitation table for D

Q _n	Q _{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Extended Truth table

T	Q _n	Q _{n+1}	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Step 2



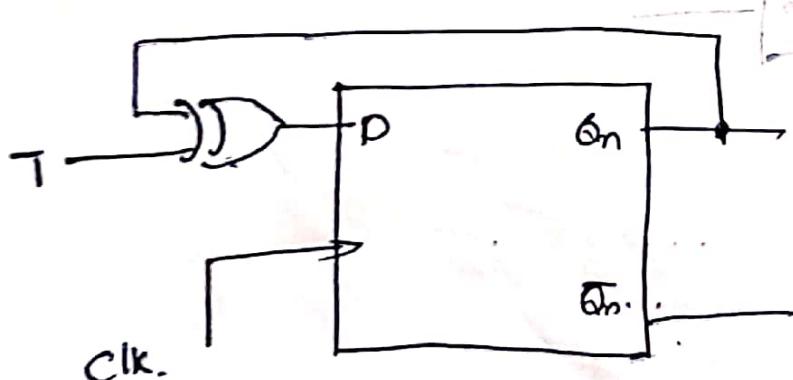
$$T \oplus Q_n = D$$

$$D = T \bar{Q}_n + \bar{T} Q_n$$

$$D = T \oplus Q_n$$

Step 3

Circuit diagram



University Questions

April 2018

- ① What is master-slave JK flip flop explain its working with a timing diagram (4)
- ② Give the characteristic & excitation table of RS flip flop and JK flip flop (4)
- ③ Compare Synchronous & Asynchronous Sequential circuits (3)
- ④ Convert JK flip flop to T flip flop (6)
- ⑤ Explain race condition in JK flip flop.

July 2017

(3)

- ① what is meant by race condition in a flip flops
- ② Explain Clocked Sequential Circuits can be designed with state equations using an example (9)

December 2018

- ① Write the excitation table of SR, JK & T flip flops
- ② Given below is a sequential Circuit using D flip flop, write the state table and draw a state diagram (9)
- ③ Explain master-slave. (9)



Module - V

① Registers

↳ Group of flip flops + gates = registers

↳ digital circuit for storage purpose

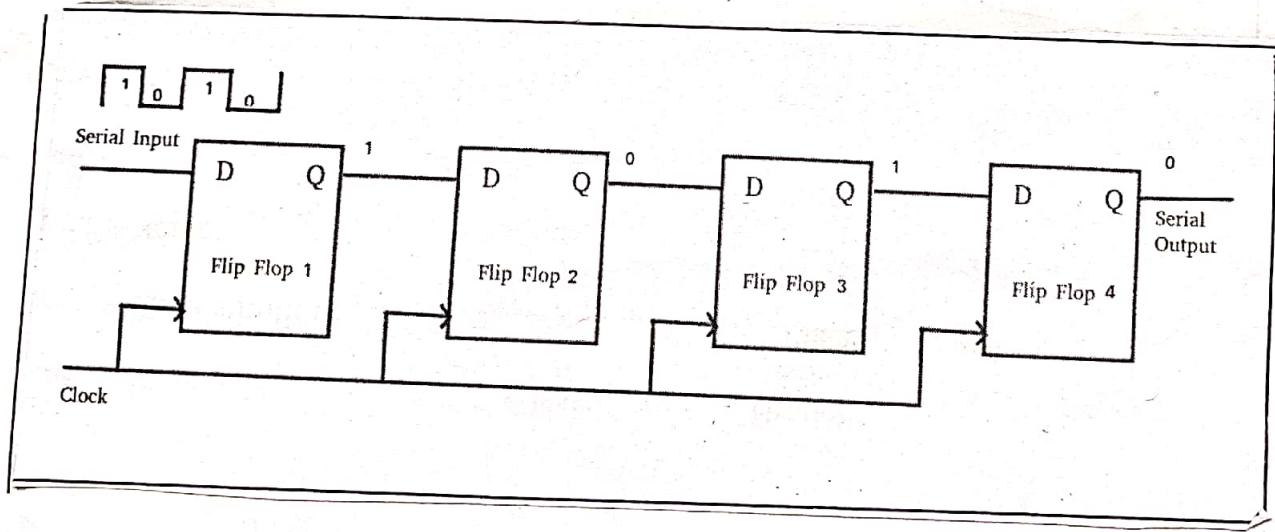
↳ A flip flop is capable of storing one bit of information and holds binary information.

↳ Gates - Controls when and how new information is transferred to the register.

↳ Simplest register only contains flip flops with no external gates.

Figures:- 4 bit register using D flip flop.

↳ Same clock pulse is given to the all flip flops



↳ A register that represents a group of flip flops in parallel constructed using D flip flop

↳ A register that responds to pulse duration called a gated latch.

Register with parallel load.

① Loading the register

Transfer of new information into

a register

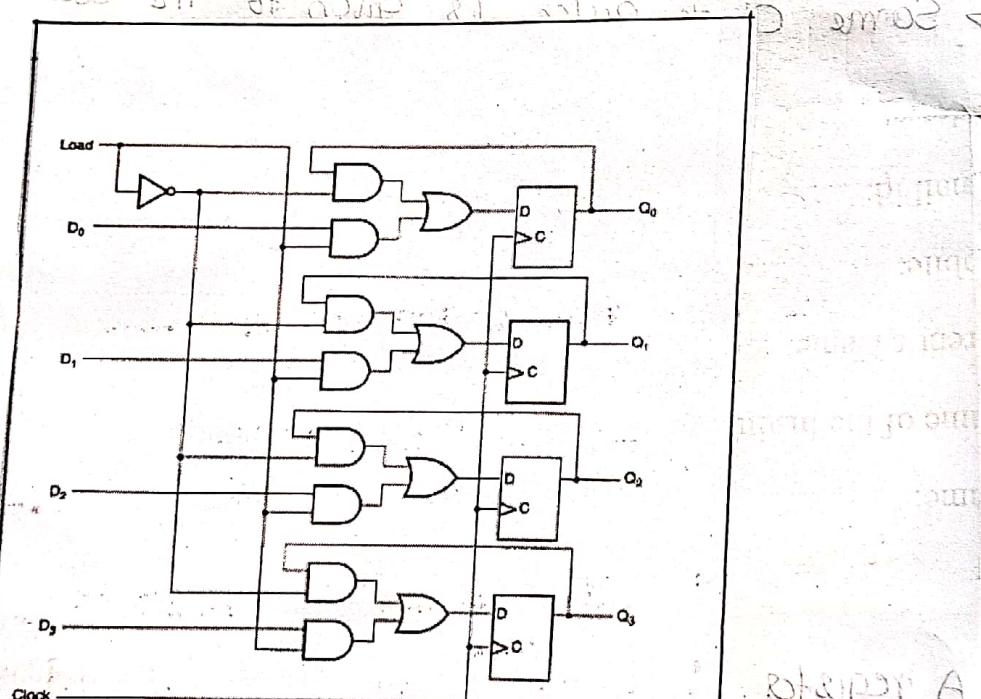
② Parallel loading

all the bits are

loaded simultaneously

4 bit register with parallel load.

Load input determines whether the next clock pulse will accept new information or will retain the previous input.

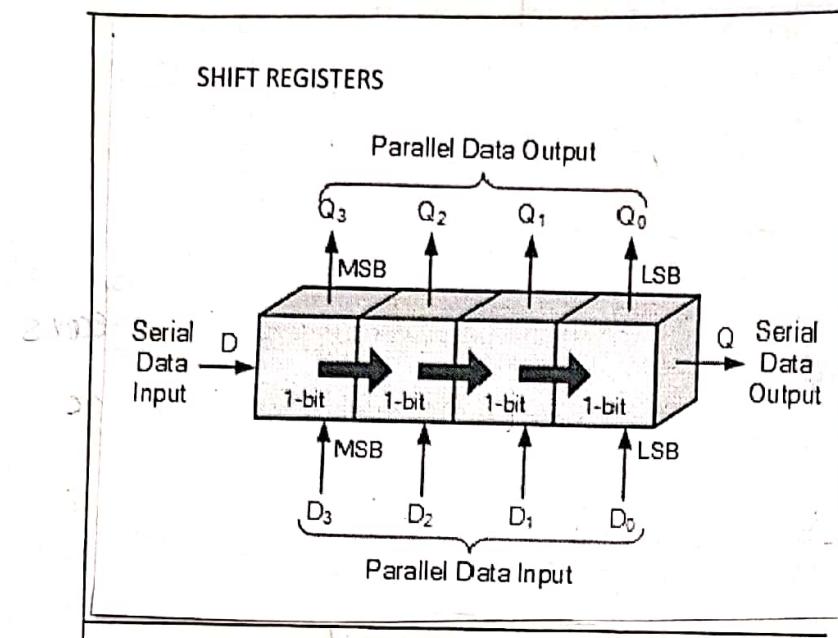


4-bit register with parallel load

The transfer of new information into a register is referred to as loading the register.

② Shift registers

↳ Shifting binary information

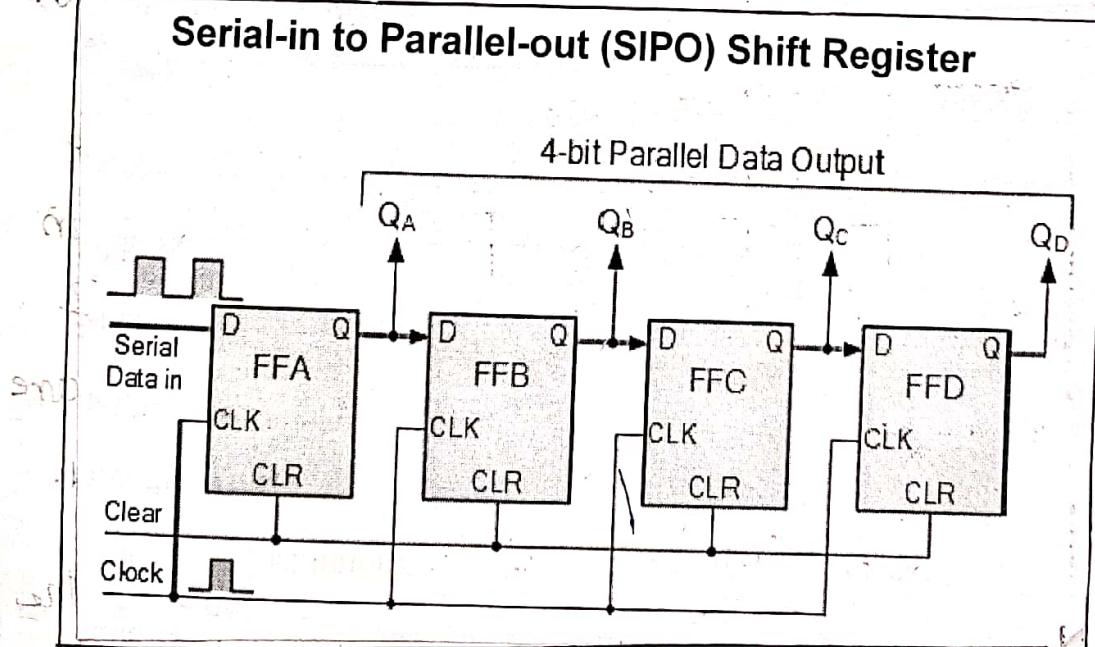


↳ A register capable of shifting its binary information either to the right or to the left is called shift register.

↳ The data can be entered into shift register either serially or in parallel. also O/P can be taken either serially or in parallel.

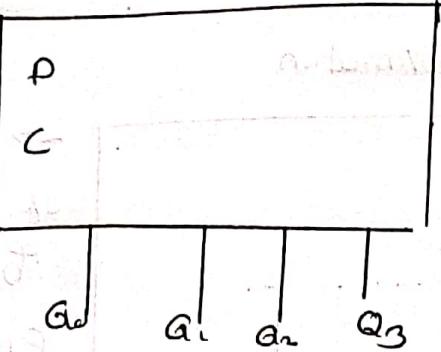
Types of Shift Registers ① SIPO

Serial-in to Parallel-out (SIPO) Shift Register



↳ This register accepts data serially - one bit at a time & produces o/p also in serial form parallel.

Data I/P



- ↳ Once the data are stored, each bit appears
on its respective O/p line & all bits are
available simultaneously.

Serial-In-parallel-Out

Basic Data Movement Through A Shift Register

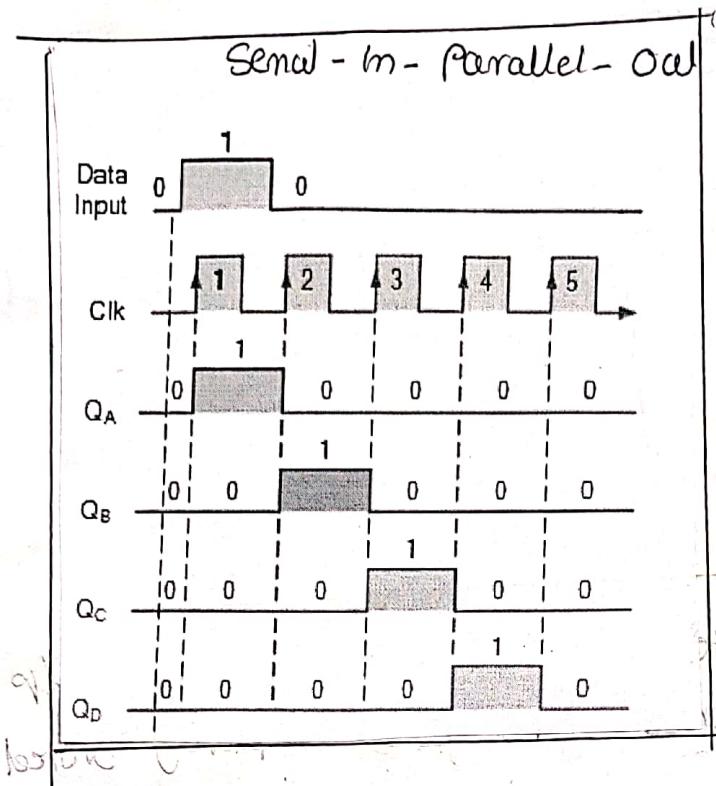
Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

↳ figure shows
the basic
data move
ment in
Serial-In
Parallel Out.

↳ data I/P
entering in
serial &
Output are
parallel.

↳ one of the
important
Shift register.

Timing diagram

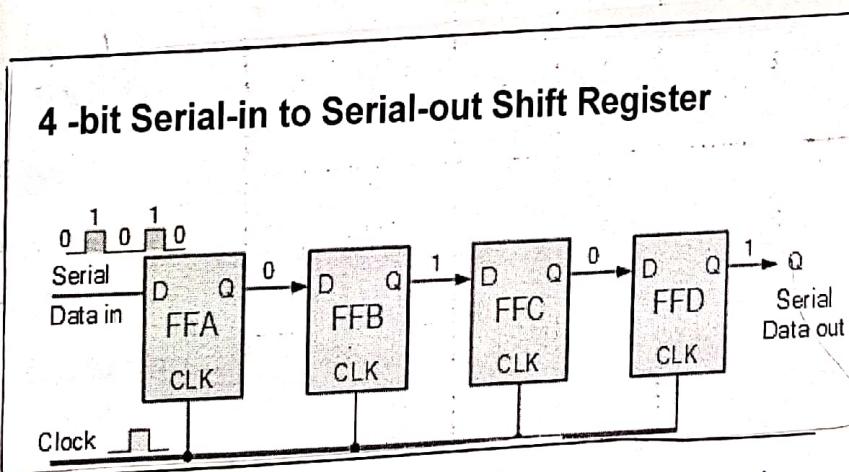


↳ figure shows the timing diagram of Serial - In - parallel - Out.

↳ with 4 stages, thus register can store upto 4 bits of data.

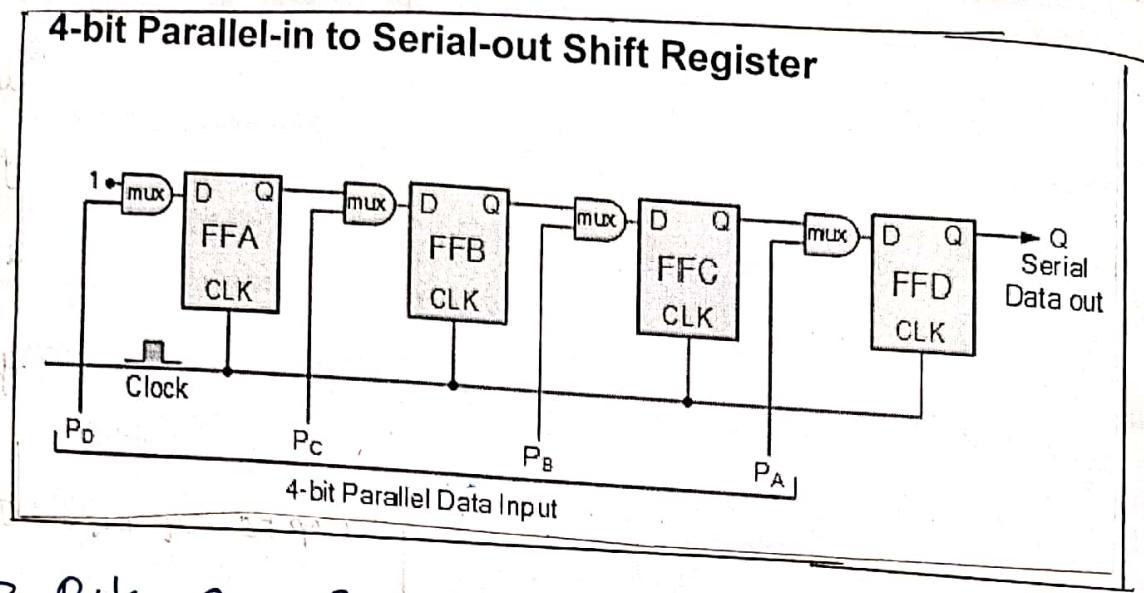
↳ based on the clock signal, the shift register is cascading. steps of 1 bit at a time.
 ↳ 5 clock signals are given in the above diagram.

② 4 bit Serial - In - to - Serial - Out Shift Registers



↳ This register accepts data serially - one bit at a time & produces o/p also in serial form.

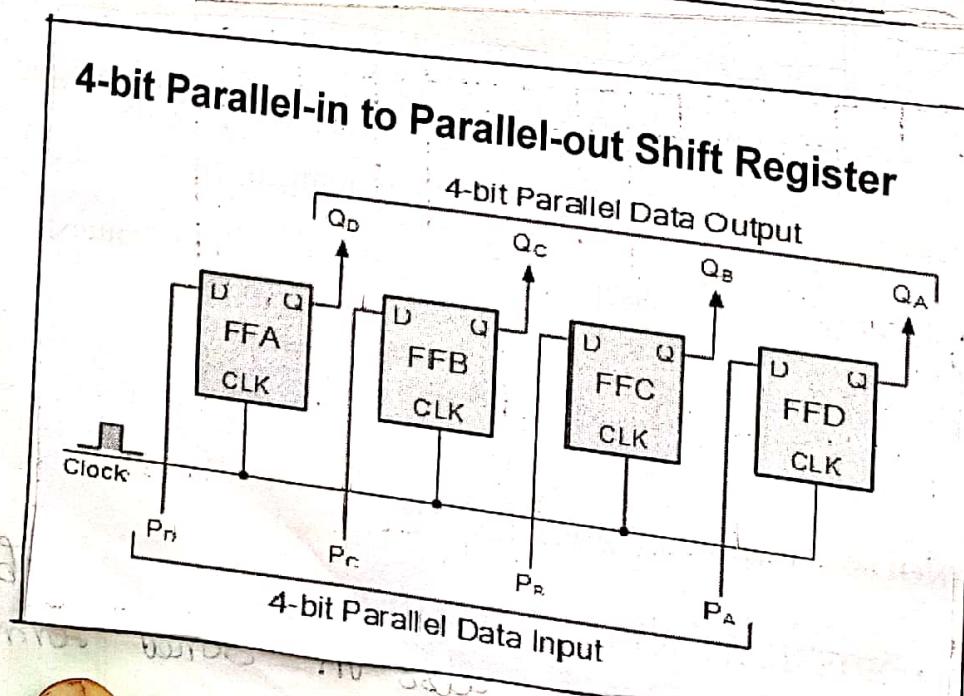
③ 4-bit Parallel-in to Serial-out Shift Registers



↳ Bits are entered simultaneously & serial output is obtained once data are completely stored in registers.

↳ 4 data input lines - P_0, P_1, P_2, P_3

④ 4-bit parallel-in to parallel-out Shift Registers

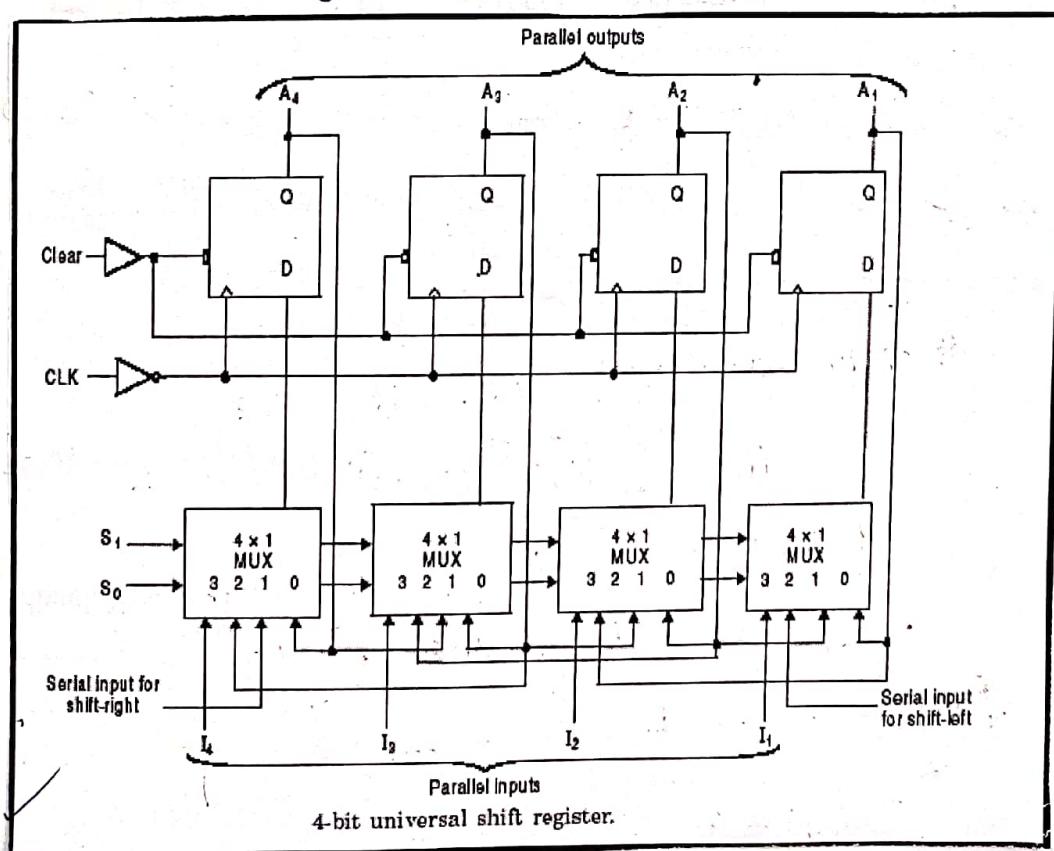


↳ Accepts 4 bits in
Parallel out is also in Parallel

(iii) Universal Shift Registers

(Bidirectional Shift registers
with parallel load)

- ↳ A register Capable of shifting both right & left is called a bidirectional shift register.
- ↳ If the register has both shift and parallel load capabilities it is called a shift register with parallel load.
- ↳ It is used as memory elements in computers.
- ↳ A unidirectional shift register is capable of shifting in only one direction.
- ↳ It can be configured to load and retrieve the data in any mode (either serial or parallel) by shifting it either towards right or left.



④ Serial adder

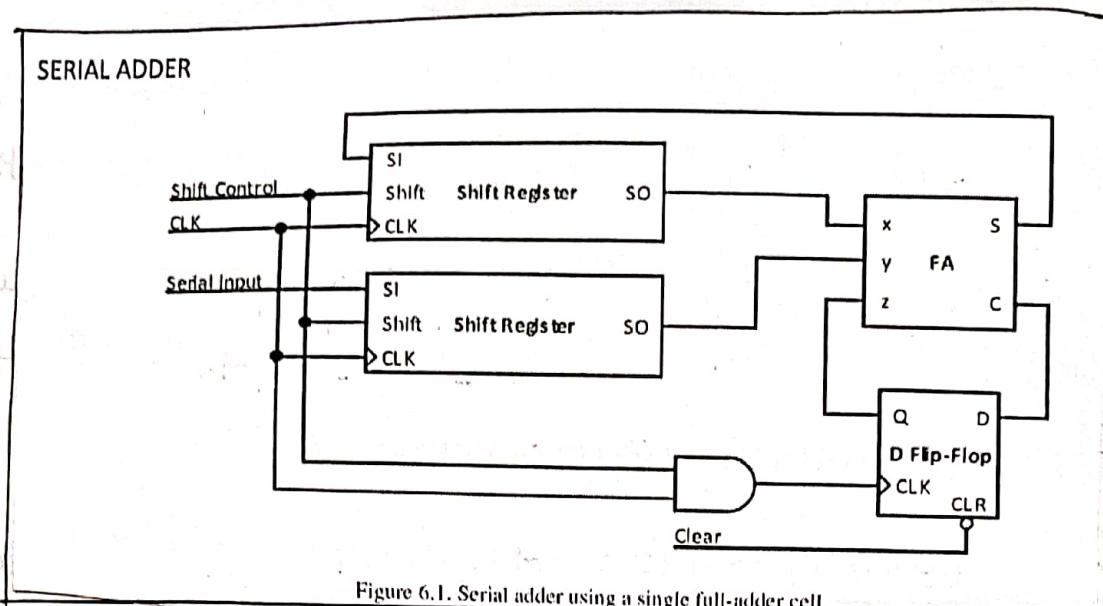


Figure 6.1. Serial adder using a single full-adder cell

→ The 2 binary nos to be added serially are

stored in 2 shift registers

→ bits are added one pair at a time

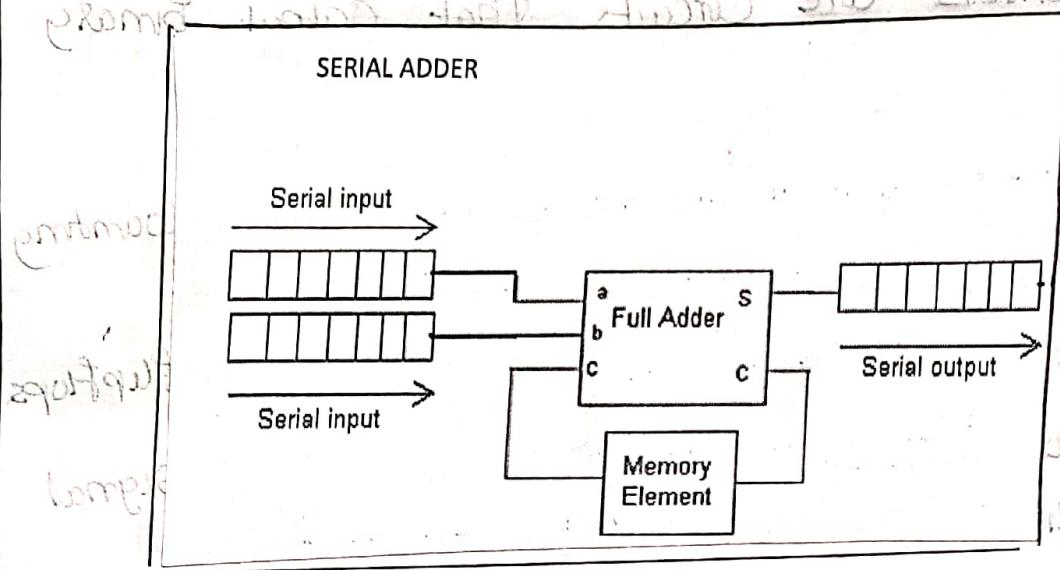
sequentially, through a single full adder circuit.

→ The carry out of the full adder is transferred to a D-flip flop. The output of this flip flop is then used as an input carry for next pair of significant bits.

→ The 2 shift registers are shifted to the right for one word - time period

→ The sum bits from the S o/p of full adder could be transferred into a 3rd shift register.

Serial adder + Full adder + memory element.



- ↳ by shifting & summing into a while bits. A are shifted out as possible to use one register for storing both augend and sum bits.
- ↳ The serial I/p (SI) of register B is able to receive a new binary no while addend bits are shifted out during addition.
- ↳ Serial adder is a binary adder that is capable of forming sum and carry outputs for addend and augend words of greater than one bit in length.
- ↳ The individual bits of the addend and augend starting with the least significant bit, are presented in sequence, together with a carry, to the adder, which then forms sum & carry outputs.

⑤ Counters

↳ Counters are Circuits that output Binary number sequence.

↳ A digital circuit which is used for a counting pulses is known as Counter.

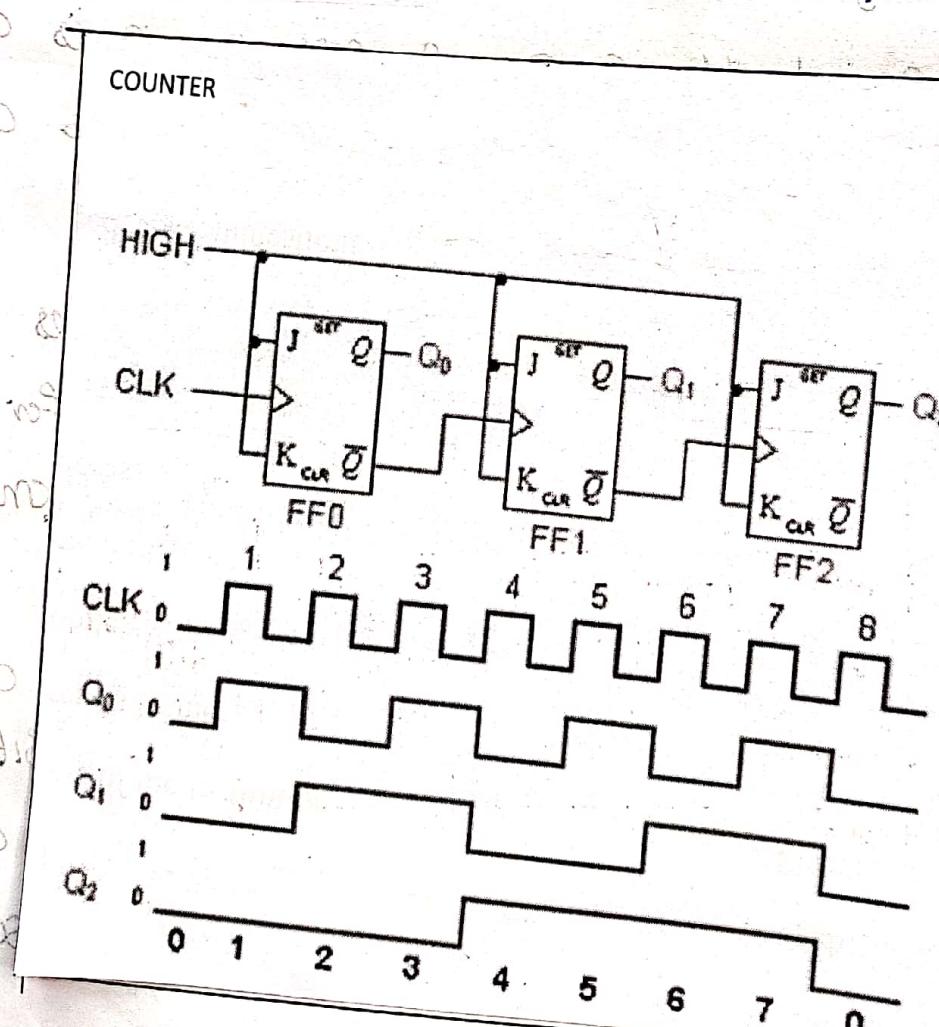
↳ Counter is the easiest application of flipflops.

↳ It is a group of flipflops with clock signal applied.

Ques A) ↳ Counters are of two types

① Asynchronous Counter

② Synchronous Counter



* Asynchronous Counter & Ripple Counter

↳ different flip flops are triggered with different clock, not simultaneously.

↳ state of one flip flop is set before other flip flop.

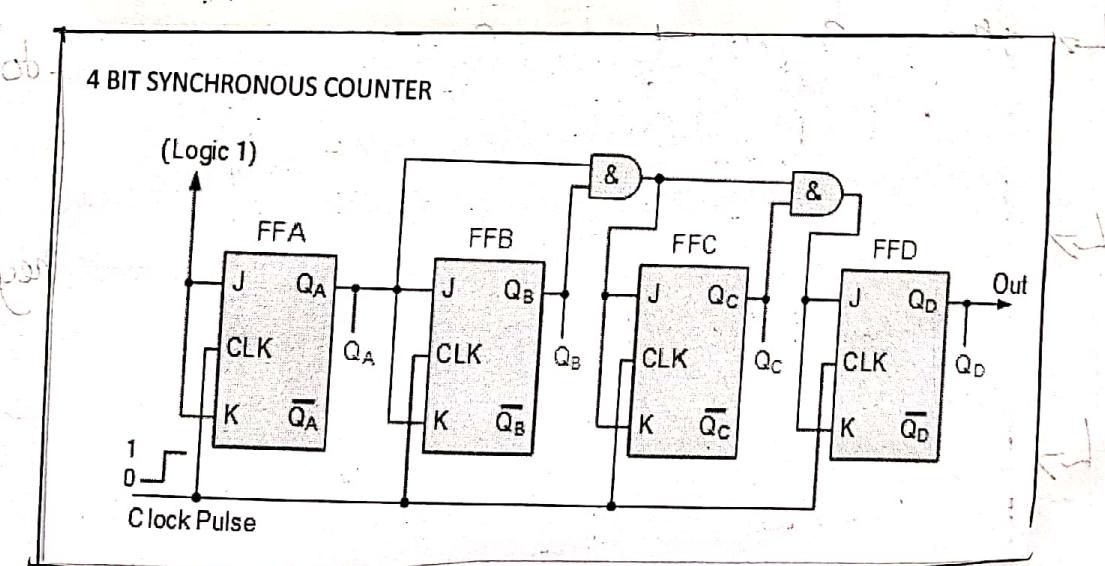
* Synchronous Counter

↳ All flip flops are triggered simultaneously at the same clock time.

↳ Synchronous Counter does not produce any decoding errors.

↳ It contains all flip flops which are all synchronized with each other because of the same clock pulse.

⑧ 4 bit Synchronous Counter



↳ In this, the clock input across all the flip flops use the same source & create the same clock signal at the same time.

↳ So, a Counter which is using the Same Clock Signal from the Same Source at the Same Time is called Synchronous Counter.

↳ It is one whose output bit change state simultaneously, with no proprie

↳ The only way we can build such a Counter Circuit from JK flip flops is to connect all the clock inputs together, so that each and every flip flop receives the exact same clock pulse at the exact same time.

Synchronous UP Counters

↳ Count up from zero and provide a change in state or output upon reaching predetermined values.

↳ others count down from a preset value to zero to provide an o/p state change

↳ The counters are synchronous, but they are asynchronously presetable.

↳ used for the counting purpose

Synchronous Counter

- ↳ A Counter is a Sequential Circuit that goes through a predetermined Sequence of States upon the Application of Clock Pulses.
- ↳ Common Clock Signal.
- ↳ The Change of State is determined from the Present State.

BCD Ripple Counters

Decade Counters



asynchronous

- * Negative edge triggered $\Rightarrow Q_1$ is used as clock \Rightarrow up counter
- * Positive edge triggered $\Rightarrow Q_0$ is used as clock \Rightarrow down counter

- * Negative edge triggered \Rightarrow

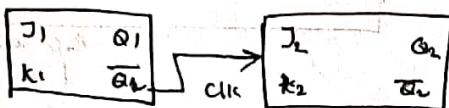


\Rightarrow down counter

- * Positive edge triggered \Rightarrow

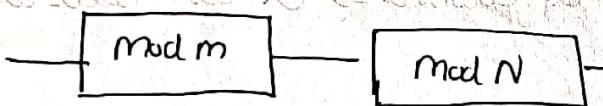


Q_0 is clock \Rightarrow up counter.



Q_1 is clock \Rightarrow down counter

② Cascading of the Counters

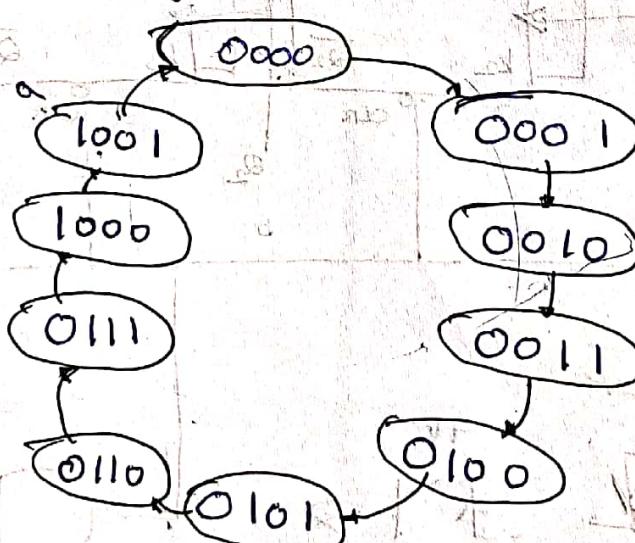


Cascading, the resulting flip flop = mod mN

Decade Counters

0 - 9 (10 States)

No of States = 10
4 bits is used to represent maximum count
 $= \text{State} - 1$
 $= 10 - 1$
 $= 9$

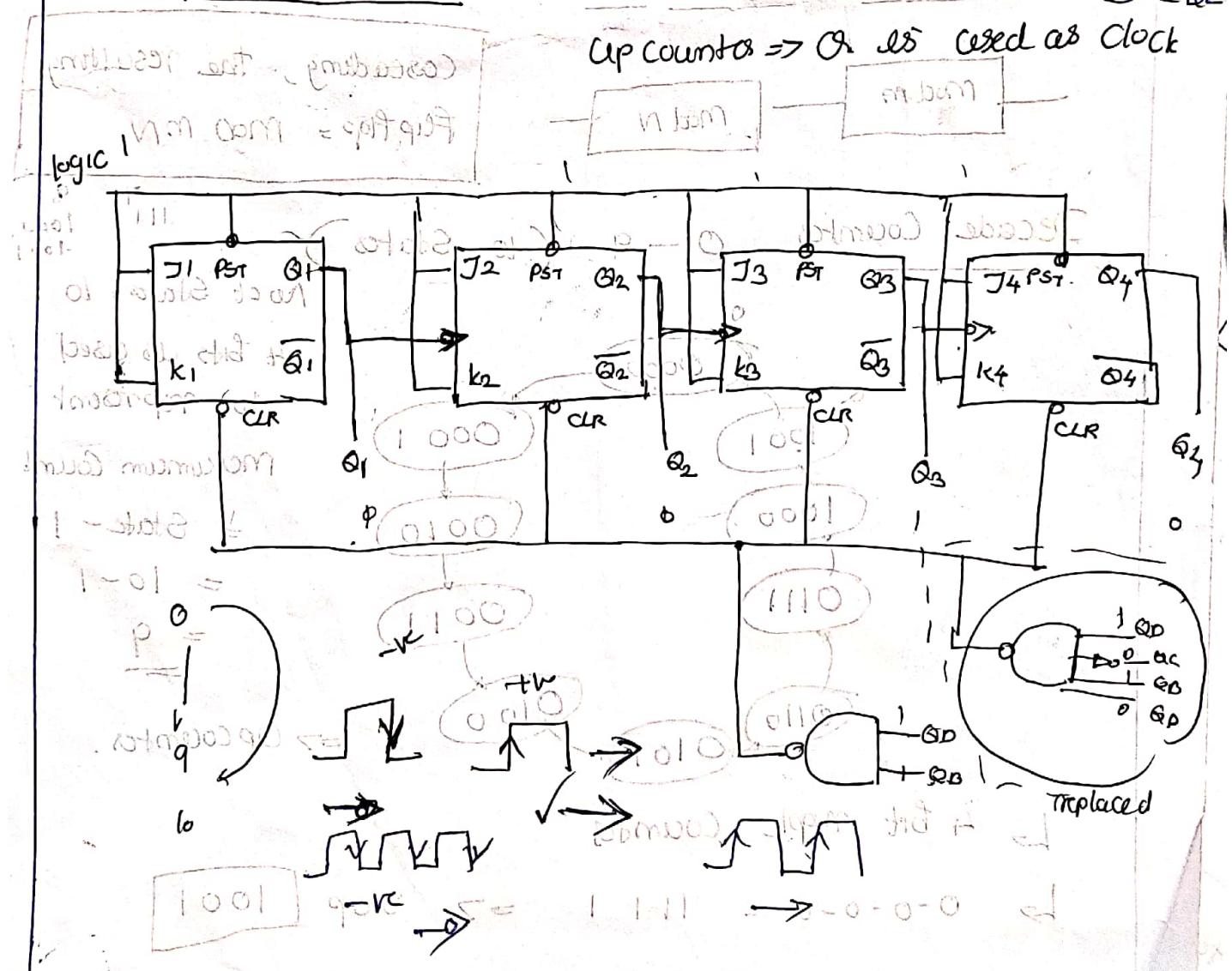


\Rightarrow Up counter.

\hookrightarrow 4 bit ripple counters

$0-0-0-0 - 1111 \Rightarrow \text{Stop}$

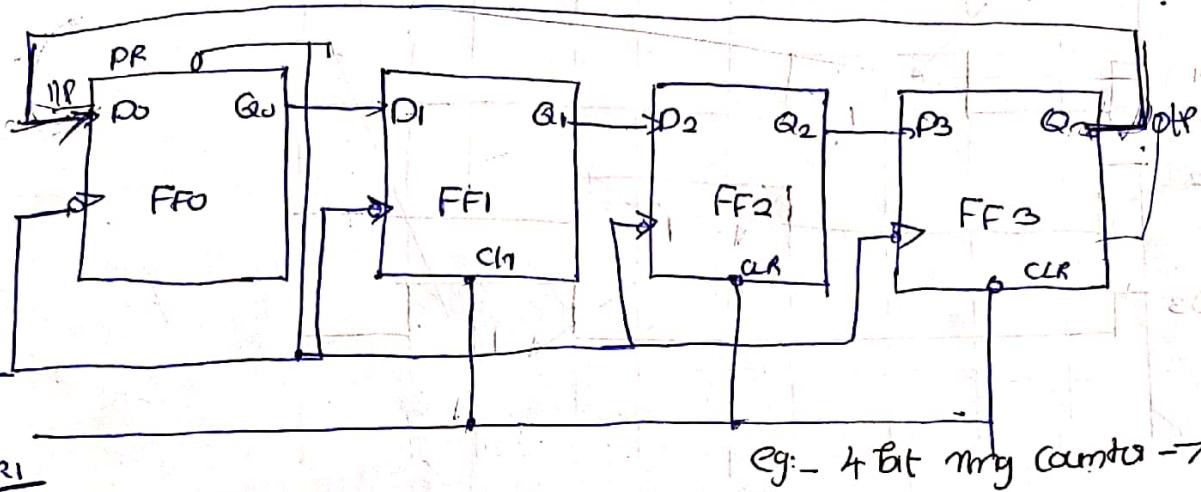
Diagram 4-bit Ripple Counter



Ring Counter (Shift register Counter)

↳ Ring Counter is a typical application of Shift registers.

↳ The only change is the output of last flip flop is connected to the input of first flip flop.



Eg:- 4 Bit ring counter \rightarrow 4 states

$$2^n = 2^4 \Rightarrow 16 \text{ States} \Rightarrow \text{Counter}$$

5 Bit ring counter \Rightarrow 5 States

Ring Counter = no of states = no of flipflop used

1000
 0100
 0010
 0001
 0000

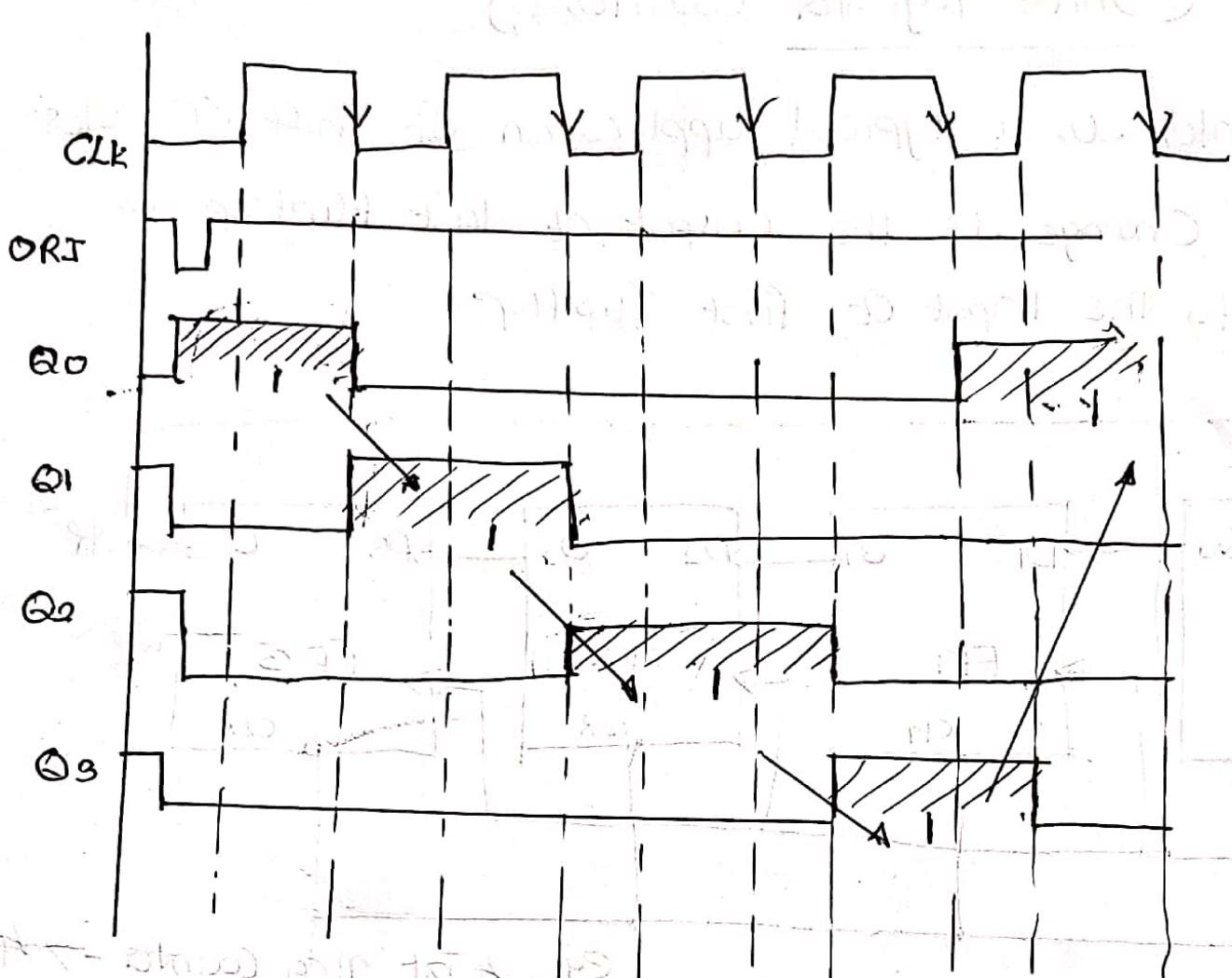
ORI = Over Riding Input of the flip flop

$$\begin{array}{ll} PR = 0 & Q = 1 \\ CLR = 0 & Q = 0 \end{array}$$

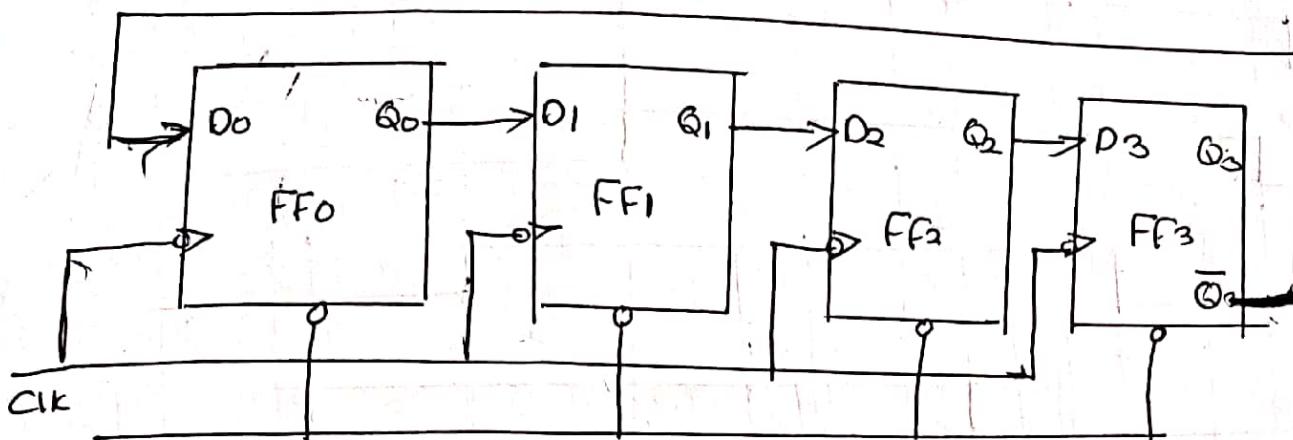
↑ predicted = 1

ORI	CLK	Q0	Q1	Q2	Q3
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

Timing diagrams showing the state transitions of the ring counter. The CLK signal is a square wave. The Q0, Q1, Q2, and Q3 signals show the state changes at each clock edge. The final state is 0001.



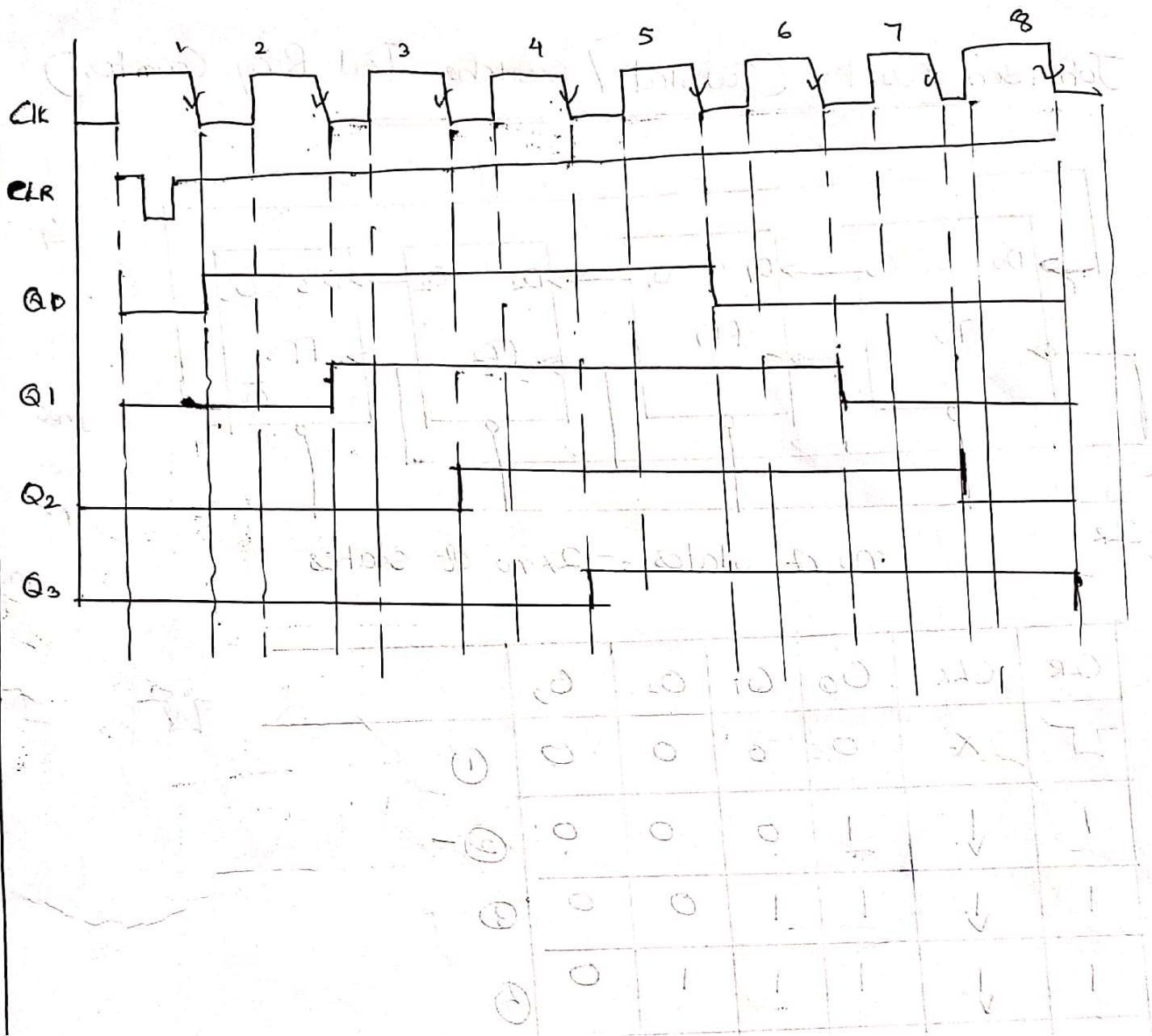
Johnson Counter (Twisted / Switched Tail Ring Counter)



CLR

no of states = $2 \times$ no of states

CLR	Clk	Q ₀	Q ₁	Q ₂	Q ₃	
1	X	0	0	0	0	①
1	↓	1	0	0	0	②-
1	↓	1	1	0	0	③
1	↓	1	1	1	0	④
1	↓	1	1	1	1	⑤
1	↓	0	1	1	1	⑥
1	↓	0	0	1	1	⑦-
1	↓	0	0	0	1	⑧-
1	↓	0	0	0	0	0



① Design a Counter that has a repeated sequence of the following six states: 000, 001, 101, 100, 101, 110

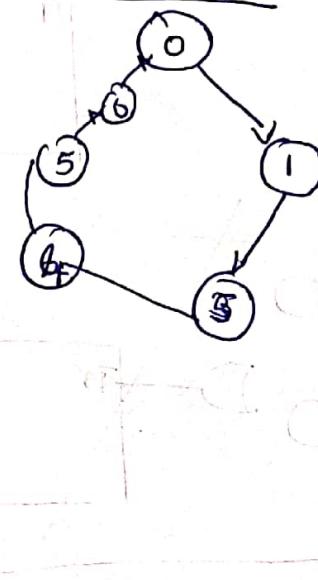
Solution

Step 1

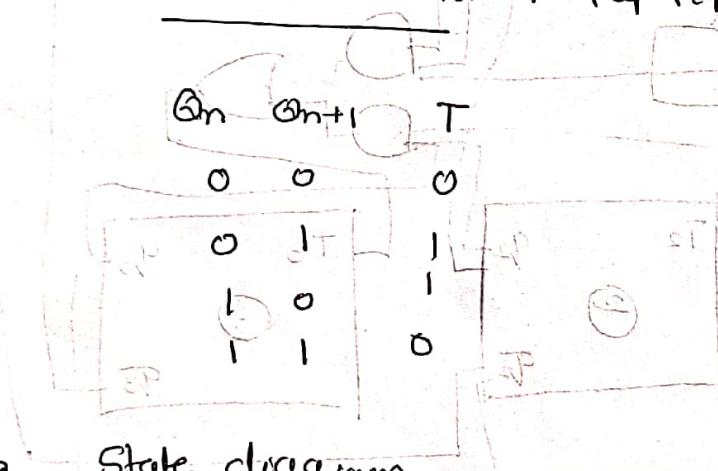
1/P

States	Present State			Next State			T_B	T_A
	q_3	q_2	q_1	\bar{q}_3	\bar{q}_2	\bar{q}_1		
0	0	0	0	0	0	1	0 0	1 0
1	0	0	1	1	0	1	1 0	0 0
5	1	0	1	1	0	0	0 0	1 5
4	1	0	0	1	0	1	0 0	1 4
5	1	0	1	1	1	0	0 0	1 5
6	1	1	0	0	0	0	1 0	0 0

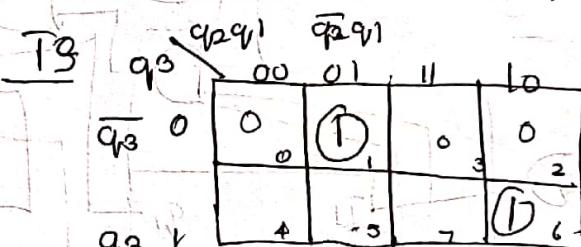
State diagram



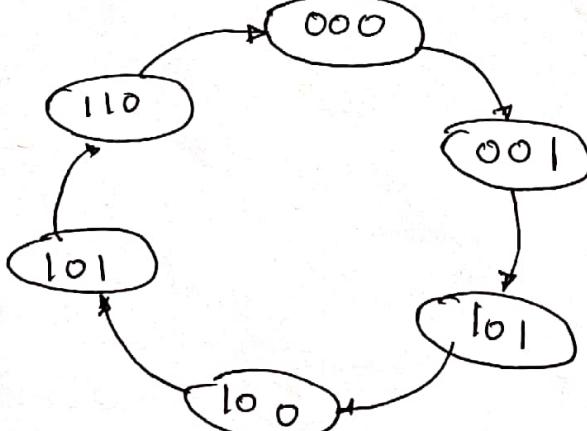
Step 2 Excitation table for T flip flops



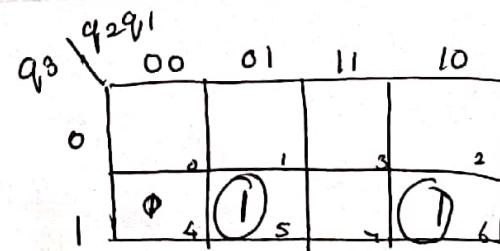
Step 4 k-map



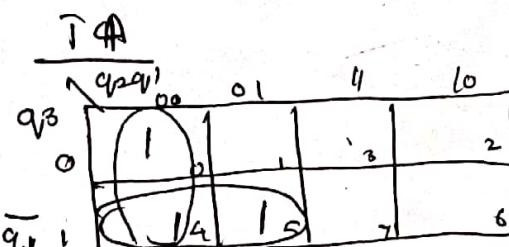
Step 3 State diagram



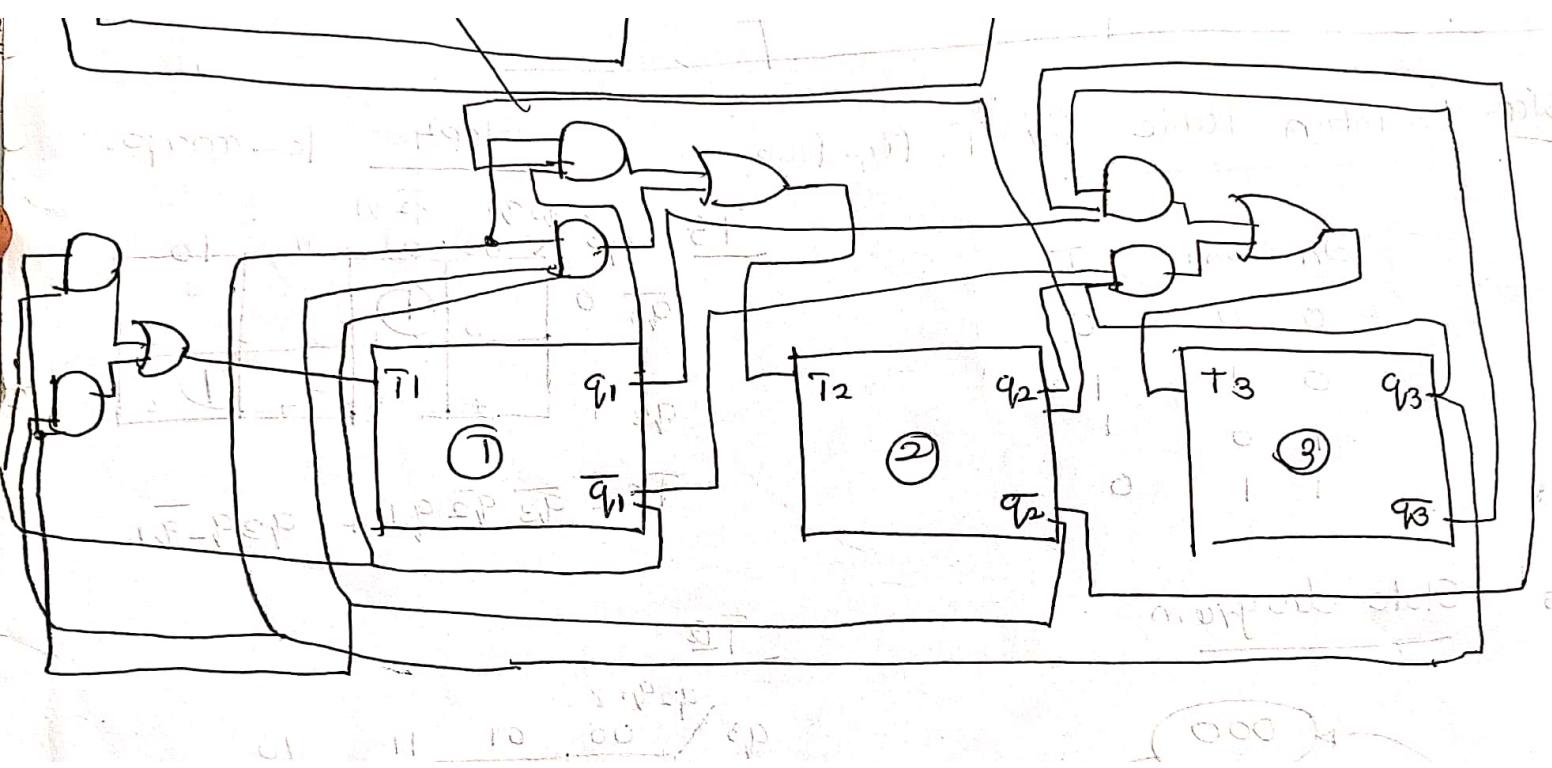
Step 5



$$T_A = q_3 q_2 q_1 + q_3 \bar{q}_2 q_1$$



$$T_A = q_3 q_2 + q_2 q_1$$



University Questions

January 2017

Page No.

- ① What is state diagram? write down two advantages of state reduction techniques. (3)
- ② what is Universal shift registers (2)
- ③ explain how a shift register is used as a converter from (1) serial to parallel data.
② parallel to serial data. (3)
- ④ how does ripple Counter differ from synchronous Counter. (3)
- ⑤ design a synchronous Counter. (3)
- ⑥ Draw the block diagram of a 4 bit Ripple Counter.

Dec 2018

- ① Compare ring Counter & Johnson Counter (2)
- ② Explain the working of 3-bit Universal shift registers. (8)
- ③ Give 2 applications of shift register (2)
- ④ Explain Johnson Counter with timing diagram (8)

April 2018

- ① Design a BCD Ripple Counter with timing diagram (10)
- ② Explain the block diagram of serial adder. Using full adder & shift registers (5)
- ③ Explain BCD adder with truth table (5)

Dec 2017

- Ques. No. _____ Date _____
- (1) Design a Synchronous Counter Using JK flip flop. (10)
 - (2) Draw & explain 4 bit Johnson Counter. (10)
 - (3) Draw & explain the different types of Shift registers. (8)
 - (4) List down the applications of shift registers. (6)

July 2017

- (1) Design and Implement 4 bit binary Synchronous up counter. (10)
- (2) Explain Johnson Counter & its working. (10)