

## UNIT-4

### SEQUENTIAL CIRCUITS

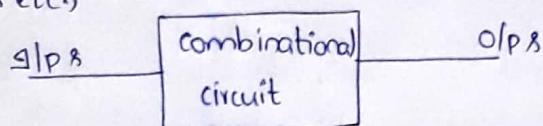
classification of digital circuits:- The digital circuits are broadly classified into two types (1) combinational circuits (2) sequential circuits.

Combinational circuit:- In combinational circuits o/p's at any point of time depends on s/p at that point of time.

\* The logic gate is the basic building block of combinational circuits

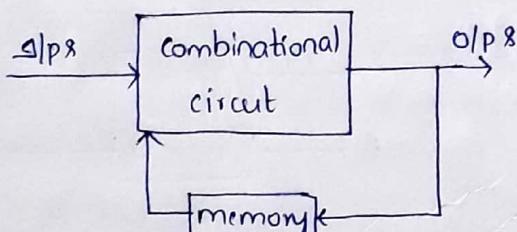
\* There is no memory requirement in combinational circuits.

Eg:- adders, subtractors, multipliers, multiplexer, encoder, decoder, demultiplexers etc.,



sequential circuit:- In sequential circuit the o/p depends on present s/p as well as past o/p. Memory is a basic building block of sequential circuits. Memory is used to store the past o/p's.

\* The sequential circuit is a combination of combinational circuit & memory.



Eg:- latches, flip-flops, registers, counters

\* Difference b/w combinational & sequential circuits?

(A)	combinational	sequential
(1)	combinational circuits are faster	(1) Sequential circuits are slower than combinational.
(2)	combinational circuits are easy to design.	(2) Sequential circuits are comparatively difficult to design.
(3)	Memory unit is not required.	(3) Memory unit is required to store past o/p's.
(4)	For combinational circuits the o/p depends on s/p at any instant of time	(4) In sequential circuit the o/p depends on present s/p & past o/p

## Classification of Sequential Circuits:-

Sequential circuits are classified into two types

(1) Synchronous sequential circuits

(2) Asynchronous sequential circuits

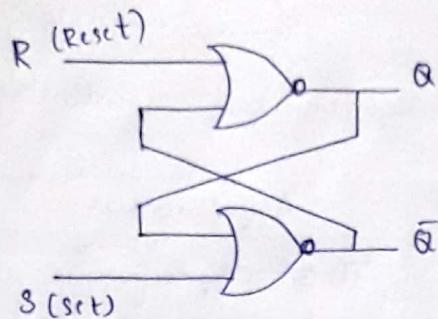
\* Difference b/w Synchronous & Asynchronous sequential circuits?

Synchronous	Asynchronous
(1) In synchronous circuit the change in S/I/P can affect the memory elements upon activation of clock signal.	(1) In asynchronous circuit the change in S/I/P can affect the memory elements at any point of time.
(2) Memory elements are clocked flip-flops.	(2) Memory elements are unclocked flip-flops.
(3) Speed of the clock is the maximum speed of circuit.	(3) Since clock is not present the circuits are faster than synchronous.
(4) These circuits are easier to design.	(4) Asynchronous circuits are difficult to design.

## Latches & Flipflops:-

Latches	Flipflops
(1) A latch is a sequential circuit used to store binary information in asynchronous arrangement.	(1) A flipflop is a sequential circuit used to store binary information in synchronous arrangement.
(2) 1 latch can store 1 bit of information.	(2) 1 flipflop can store 1 bit of information.
(3) Latch is an asynchronous device and it has no clock.	(3) Flipflop has clock S/I/P and it is synchronised with flipflop o/p.
(4) Latches are level sensitive i.e., the o/p tracks the S/I/P when clock is high.	(4) Flipflops are edge sensitive the o/p tracks the S/I/P when there is a rising or falling edge of the clock.

S-R latch using NOR gates:- SR latch has 2 inputs set & Reset. It has two outputs  $Q$  &  $\bar{Q}$ .  $\bar{Q}$  is complement of  $Q$ . S-R latch can be implemented by using NAND & NOR gates.

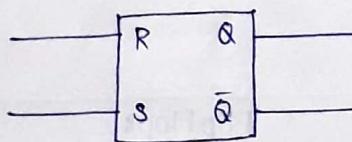


Logic diagram for S-R latch

Truth Table:- (Function Table)

S/R		Q <sub>n+1</sub>	Operation mode
S	R	Q <sub>n</sub>	
0	0	Q <sub>n</sub>	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Invalid

Logic symbol for S-R latch:-



Characteristic Equation

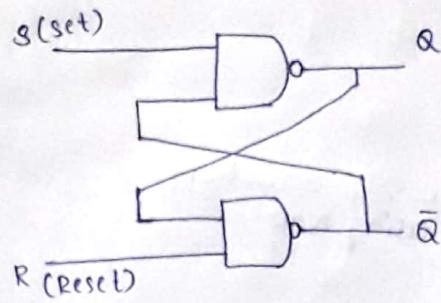
Q <sub>n</sub>	SR			
	00	01	11	10
0	0	1	X <sub>3</sub>	1 <sub>2</sub>
1	1 <sub>4</sub>	1 <sub>5</sub>	X <sub>7</sub>	1 <sub>6</sub>

K-map for SR latch

Characteristic table:-

S/R		Present state (Q <sub>n</sub> )	Next state Q <sub>n+1</sub>	Equation
S	R			Q <sub>n+1</sub> = S + R Q <sub>n</sub>
0	0	0	Q <sub>n</sub> (0)	
0	0	1	Q <sub>n</sub> (1)	
0	1	0	0	
0	1	1	0	
1	0	0	1	
1	0	1	1	
1	1	0	X	
1	1	1	X	

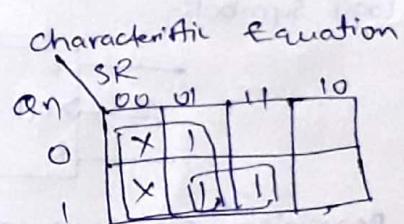
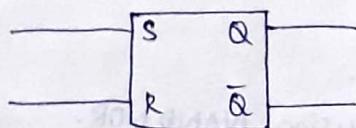
SR latch using NAND gates:-



Truth Table:-

S/p's		O/p's		Operation modes
S	R	$Q_{n+1}$		
0	0	?		Invalid
0	1	1		Set
1	0	0		Reset
1	1	$Q_n$		No change

Logic symbol:-

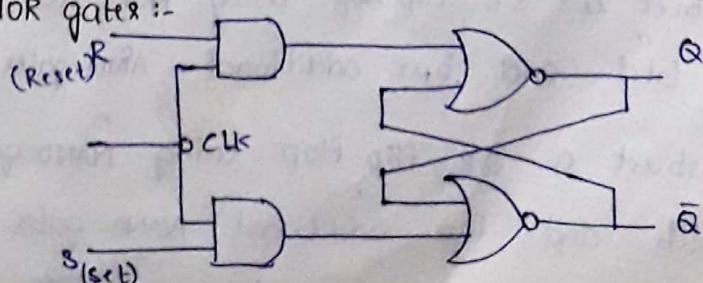


Characteristic table:- ( ~~Column 01 is invalid~~ )

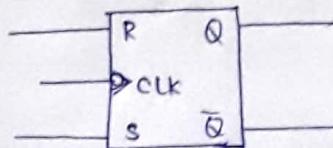
$$Q_{n+1} = \bar{S} + R Q_n$$

S/p's		present state ( $Q_n$ )	Next state ( $Q_{n+1}$ )
S	R		
0	0	0	X
0	0	1	X
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

S-R flipflop using NOR gates:-

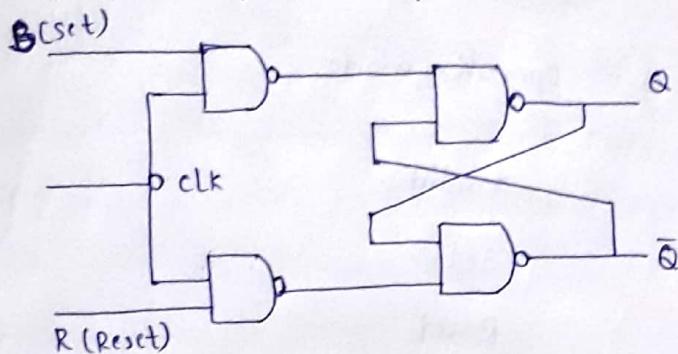


logic symbol:-

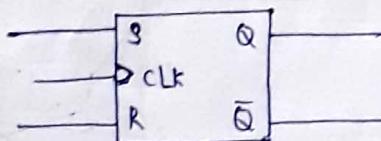


→ Truth table, Characteristic table, & equation  
Remaining are same as S-R latch using NOR.

S-R flipflop using NAND gate:-



logic symbol:-



Remaining are same as S-R latch using NAND NOR.

excitation table:- for S-R flip flop using NAND & NOR gates.

present state (Qn)	Next state (Qn+1)	Required s/p
0	0	S X
0	1	1 0
1	0	0 1
1	1	X 0

Flip-Flop:-

Def: The latch with additional control s/p is called flip-flop.

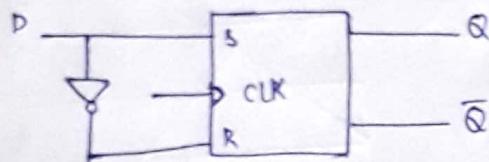
The additional control s/p is either clock (01) enable (G1).  
(CLK)

\* We construct a S-R flip-flop using NOR gates with the help of S-R NOR latch and two additional AND gates.

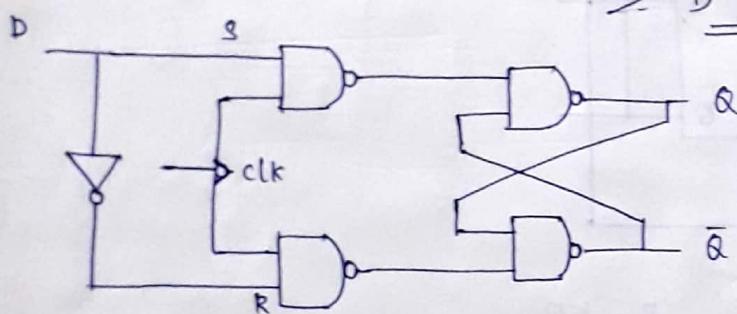
\* To construct a S-R flip-flop using NAND gates with the help of NAND latch and two additional NAND gates.

D-Flip flop (Delayed - flip flop) :- In S-R flip flop when S/R's are 0,0 there is no change in the output. When S/R's are 1,1 the output is invalid. Hence no need to consider these two states. The remaining S/R's (0,1) & (1,0) are obtained by connecting S-R inputs using a NOT gate known as D-flip flop. D-flip flop stands for delayed flip flop. In D-flip flop if S/R is 0 output is 0. If 1 is S/R the output is 1.

conversion from S-R to D-FF:

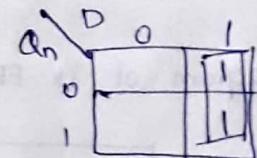
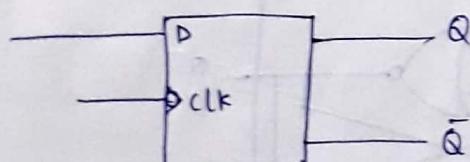


characteristic equation :  $S = D, R = \bar{D}$  D flip flop



logic diagram of D-FF

logic symbol of D-FF



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

Truth table:-

S/R's	O/p's
D	$Q_{n+1}$
0	0
1	1

characteristic table:-

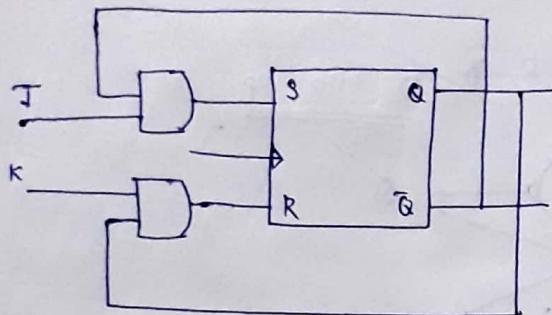
S/R's	present state	Next state
(D)	( $Q_n$ )	( $Q_{n+1}$ )
0	0	0
0	1	0
1	0	1
1	1	1

Excitation table:-

Present State (Q <sub>n</sub> )	Next State (Q <sub>n+1</sub> )	Required S/P (D)
0	0	0
0	1	1
1	0	0
1	1	1

J-K FlipFlop:- J-K flipflop works similar to S-R flipflop except for one of the S/P combination S=1 & R=1 which is not valid in S-R flipflop is valid in JK flipflop. It can be used in master slave designs.

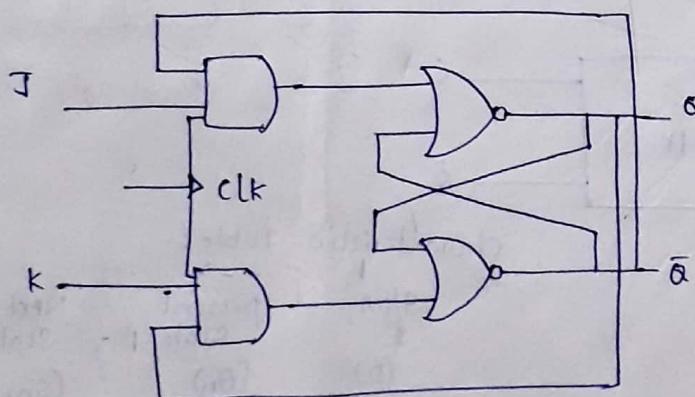
Conversion from SR to J-K flipflop:



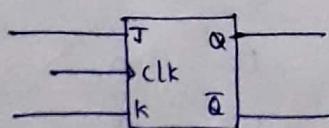
Characteristic equation

$$S = J\bar{Q}, \quad R = KQ$$

Logic diagram of JK FF:-



Logic symbol of JK-FF



Truth Table:-

S/p		Q/p		Operating mode
J	K	Q <sub>n+1</sub>	Q <sub>n</sub>	
0	0	Q <sub>n</sub>		No change
0	1	0		Reset
1	0	1		Set
1	1	Q̄ <sub>n</sub>		complement

Characteristic table :-

S/p		present state	Next state
J	K	Q <sub>n</sub>	Q <sub>n+1</sub>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Characteristic equation:-

J/K		00	01	11	10
Q <sub>n</sub>		0	1	1	0
0	0			1	1
0	1			1	0
1	0			0	1
1	1			0	0

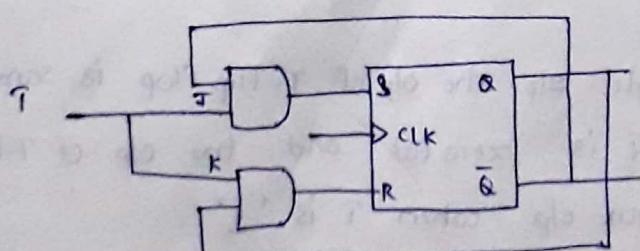
$$Q_{n+1} = J Q_n + K' Q_n$$

Excitation table:-

present state	next state	S/p
Q <sub>n</sub>	Q <sub>n+1</sub>	J      K
Q <sub>n</sub>	Q <sub>n+1</sub>	
0	0	0      X
0	1	1      X
1	0	X      1
1	1	X      0

T- flipflop :- (Toggle flipflop)

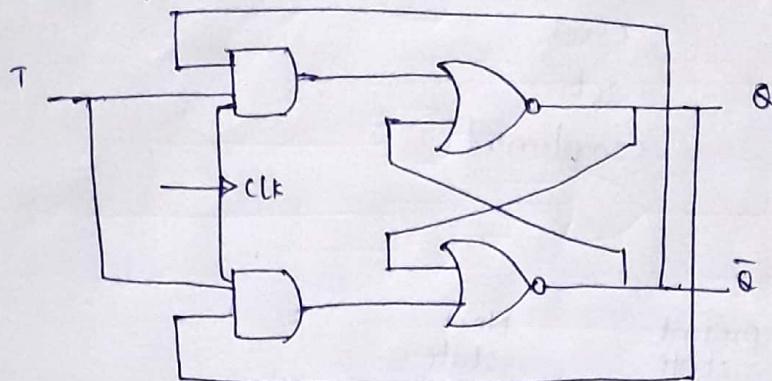
Conversion from SR to T-FF :-



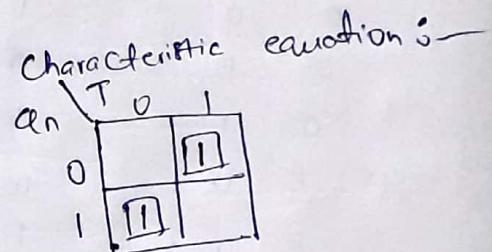
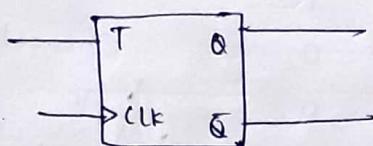
Characteristic equation:-

$$S = T\bar{Q}, \quad R = TQ$$

Logic diagram of T-FF



Logic symbol of T-FF



Truth Table :-

S/I/P	O/P	Operating mode
T	Q <sub>n+1</sub>	
0	Q <sub>n</sub>	No change
1	Q̄ <sub>n</sub>	Complement

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

Characteristic Table :-

S/I/P	P.S	N.S
T	Q <sub>n</sub>	Q <sub>n+1</sub>
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table :-

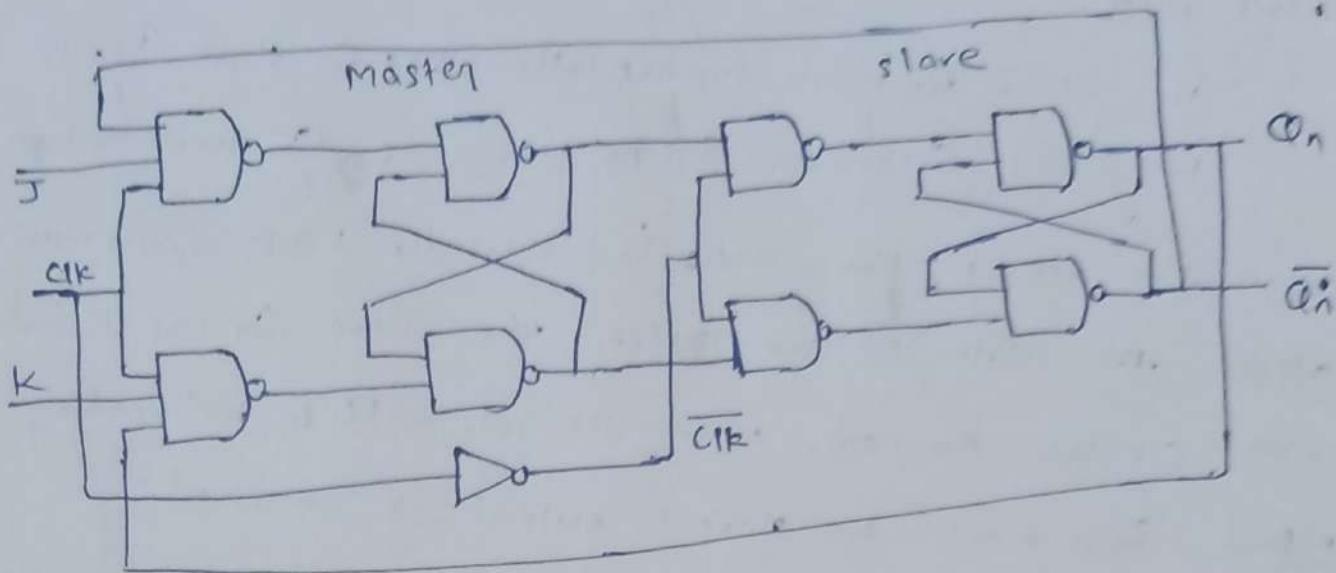
P.S	N.S	Rew S/I/P
Q <sub>n</sub>	Q <sub>n+1</sub>	T
0	0	0
0	1	1
1	0	1
1	1	0

\* The J-K S/I/P's are connected to each other to obtain the T- flip flop.

\* The T- flip flop has simple S/I/P. The O/P of T- flip flop is same as the previous O/P. When T is zero(0) and the O/P of T- flip flop is complement of previous O/P when T is '1'.

\* Race around condition:-

## The master-slave JK Flip Flop



### Race Around Condition

for JK flip flop, if  $J=K=1$ , and if  $clk=1$  for a long period of time, then Qn output will toggle as long as clk is high, which makes the output of flip flop unstable or uncertain. This problem is caused race around condition. This problem can be avoided by ensuring the clock input is at logic '1' only for a short period of time. This introduced the concept of master-slave JK flip flop.

### master - slave JK flip-flop

The master-slave JK flip flop is basically a combination of two JK flip flops connected together in a series configuration. Out of these one acts as a "master" and the other as "slave". The output from the master flip flop is connected to the two inputs of slave flip flop whose output is feedback to inputs of master flip flop.

In addition to these two flip flops, the circuit also includes an "inverter". The inverter is connected to the clock pulse.

If  $\text{clk} = 0$  for master flip flop, then  $\text{clk} = 1$  for slave flip flop.

If  $\text{clk} = 1$  for master flip flop, then  $\text{clk} = 0$  for slave flip flop.

When the  $\text{clk} = 1$ , the slave is isolated;  $J$  &  $K$  inputs may affect the state of the system. The slave flip flop is isolated until  $\text{clk}$  goes to zero. When  $\text{clk}$  goes to '0', information is passed from master to slave & output is obtained.

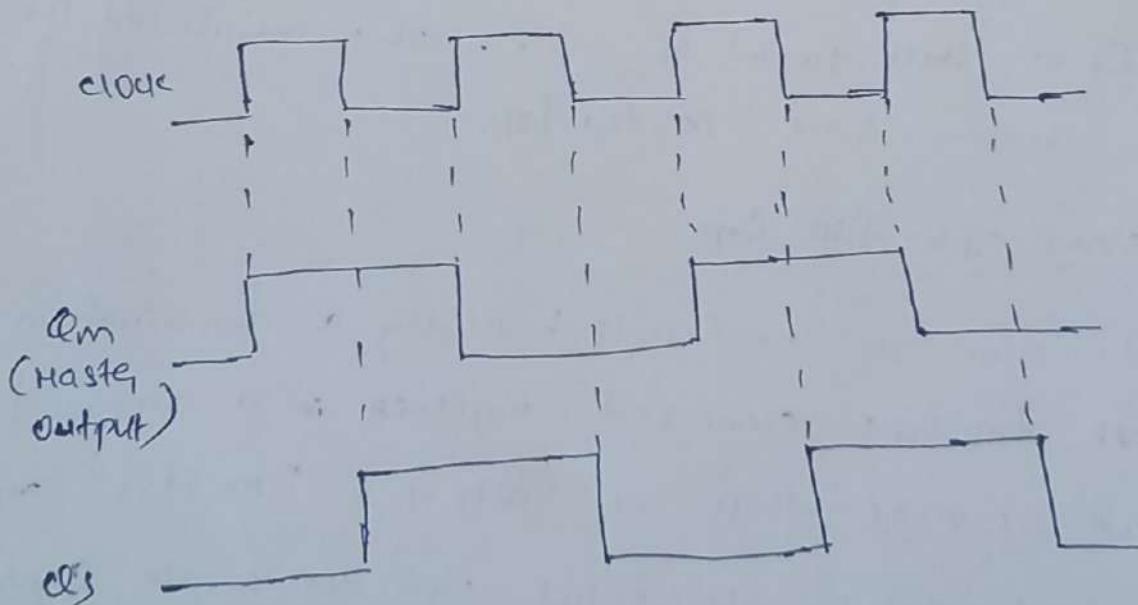
When  $J=0, K=0$ ,  $Q_n$  remains unchanged.

When  $J=0, K=1$ ,  $Q_n$  Reset

When  $J=1, K=0$ ,  $Q_n$  Set

When  $J=1, K=1$ ,  $Q_n$  toggles.

### Timing diagram of a Master-slave flip flop



(slave output). \* when  $\text{clk}$  is high, the master is operational.

\* when  $\text{clk}$  is low, the slave is operational.

## Conversion of Flipflops :-

- Steps :-
1. Identify available and required flipflop.
  2. Make characteristic table for required flipflop.
  3. Make excitation table for available flipflop
  4. Write boolean expression for available flipflop
  5. Draw the circuit.

- SR Flipflop to JK Flipflop
- JK Flipflop to SR Flipflop
- SR Flipflop to D Flipflop
- D Flipflop to SR Flipflop
- JK Flipflop to T Flipflop
- JK Flipflop to D Flipflop
- D Flipflop to JK Flipflop

### SR Flipflop to JK Flipflop :-

Step 1 : We have to convert SR to JK flipflop.

So, Here available flipflop is SR

Required flipflop is JK

Step 2 : characteristic table for JK flipflop

<u><math>Q_n</math></u>	<u>J</u>	<u>K</u>	<u><math>Q_{n+1}</math></u>
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

(22)

Step 3 : Excitation table for SR

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>S</u>	<u>R</u>
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Step 4 : To write the Boolean expression we need to find  
Conversion table using 243 steps.

<u><math>Q_n</math></u>	<u>J</u>	<u>K</u>	<u><math>Q_{n+1}</math></u>	<u>S</u>	<u>R</u>
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

(23)

we are writing this conversion table using  $Q_n$  &  $Q_{n+1}$  of  $JK$   
 to get SR  
 → let's write S, R Boolean expression using above table  
 with K-Map.

### K-Map Simplification:

For S

		JK				$\bar{Q}_n J$
		00	01	11	10	
$Q_n$	0	0	1	(1)	1	$\bar{Q}_n J$
	1	X	X	X	X	

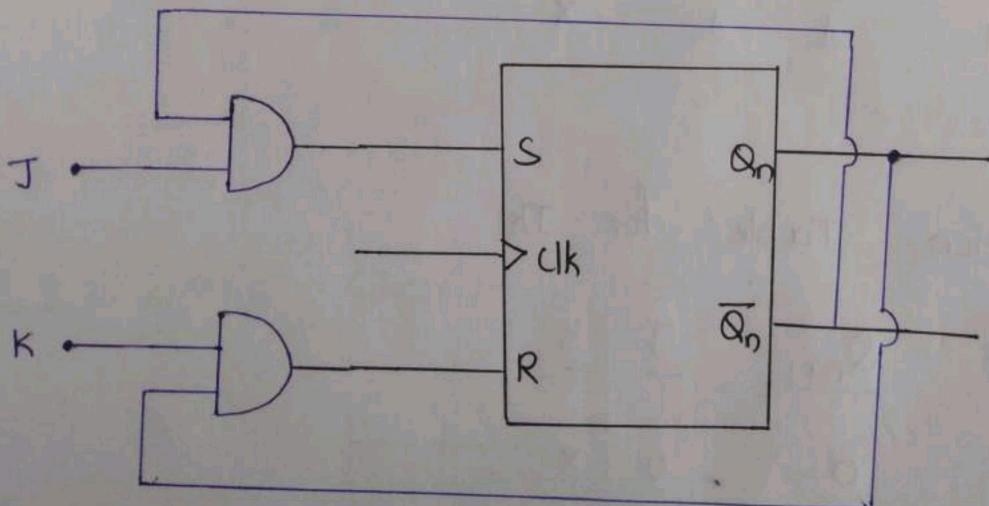
$$S = \bar{J} \bar{Q}_n$$

For R

		JK				$Q_n k$
		00	01	11	10	
$Q_n$	0	X	X	X	X	$Q_n k$
	1	1	1	1	1	

$$R = Q_n k$$

### Step 5 :- Circuit



(24)

## Jk to SR Flipflop :-

Step 1 : Available Flipflop is Jk

Required Flipflop is SR.

Step 2 : characteristic table for SR

<u><math>Q_n</math></u>	<u>S</u>	<u>R</u>	<u><math>Q_{n+1}</math></u>
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Step 3 :

Excitation Table for Jk

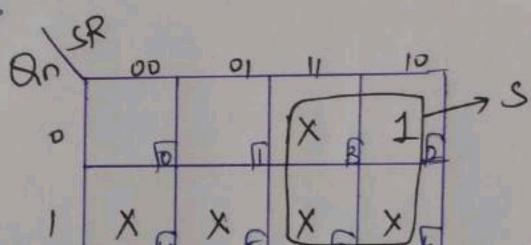
<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>J</u>	<u>K</u>
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 4 : Conversion Table

<u><math>Q_n</math></u>	<u><math>S</math></u>	<u><math>R</math></u>	<u><math>Q_{n+1}</math></u>	<u><math>J</math></u>	<u><math>k</math></u>
0	0	0	0	0	x
0	0	1	0	0	x
0	1	0	1	1	x
0	1	1	x	x	x
1	0	0	1	x	0
1	0	1	0	x	1
1	1	0	1	x	0
1	1	1	x	x	x

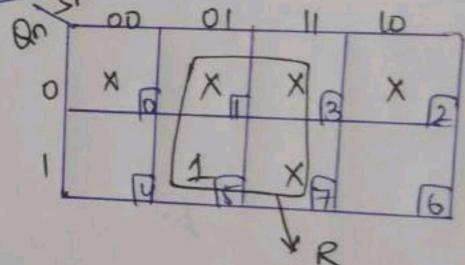
K-Map Simplification:

For  $J$



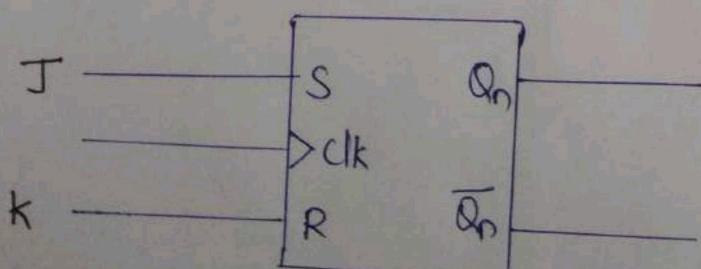
$$J = S$$

For  $K$



$$K = R$$

Step 5 : Logic circuit



SR Flipflop to D - Flipflop :-

(26)

Step 1 : Available flipflop is SR flipflop  
Required flipflop is D flipflop.

Step 2 : characteristic table for D flipflop

<u><math>Q_n</math></u>	<u>D</u>	<u><math>Q_{n+1}</math></u>
0	0	0
0	1	1
1	0	0
1	1	1

Step 3 : Excitation table for SR flipflop

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>S</u>	<u>R</u>
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

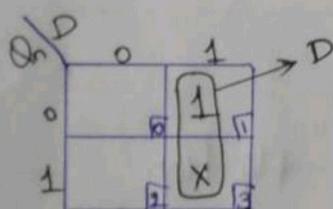
Step 4 : Conversion Table

(27)

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

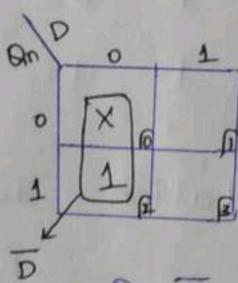
K-Map Simplification:

For S



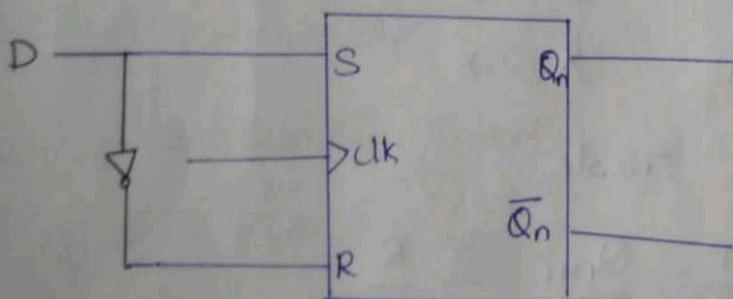
$$S = D$$

For R



$$R = \overline{D}$$

Step 5: Logic circuit



(28)

SR Flipflop to T Flipflop :-

Step 1 :

Available flipflop is SR

Required flipflop is T

Step 2 : characteristic Table for T flipflop

<u><math>Q_n</math></u>	<u>T</u>	<u><math>Q_{n+1}</math></u>
0	0	0
0	1	1
1	0	1
1	1	0

Step 3 : Excitation Table for SR flipflop

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>S</u>	<u>R</u>
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	Ø

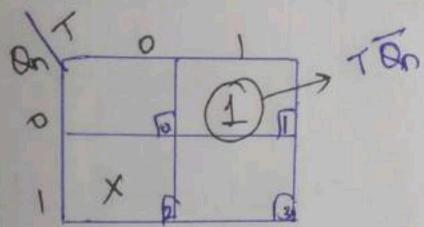
Step 4 : Conversion Table

<u><math>Q_n</math></u>	<u>T</u>	<u><math>Q_{n+1}</math></u>	<u>S</u>	<u>R</u>
0	0	0	0	X
0	1	1	1	0
1	0	1	X	Ø
1	1	0	0	1

## K-Map Simplification:

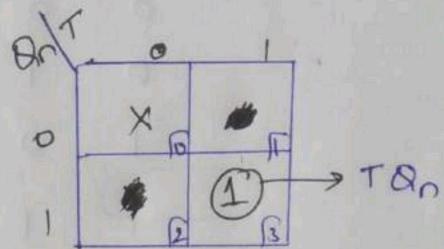
(29)

For S



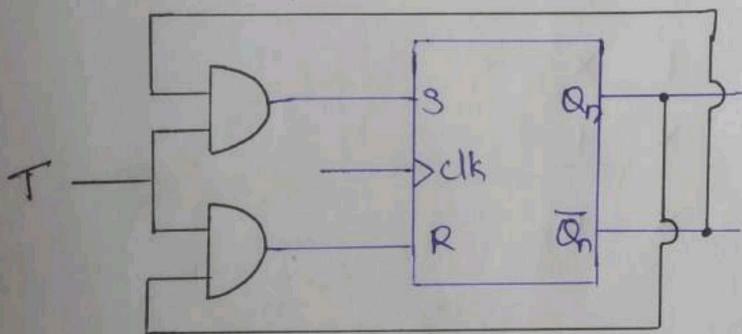
$$S = T \bar{Q}_n$$

for R



$$R = T Q_n$$

Step 5 : Logic diagram



JK Flipflop to D-Flipflop :-

Step 1 : Available flipflop is JK

Required flipflop is D

Step 2 : characteristic table for D-Flipflop:

<u><math>Q_n</math></u>	<u>D</u>	<u><math>Q_{n+1}</math></u>
0	0	0
0	1	1
1	0	0
1	1	1

Step 3: Excitation table for JK flip flop

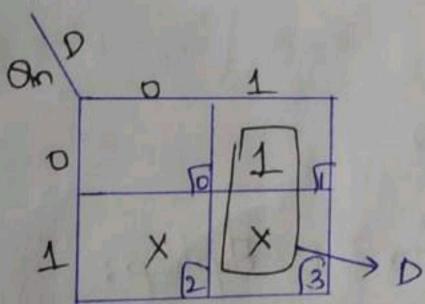
<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>J</u>	<u>K</u>
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 4: Conversion Table

<u><math>Q_n</math></u>	<u>D</u>	<u><math>Q_{n+1}</math></u>	<u>J</u>	<u>K</u>
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

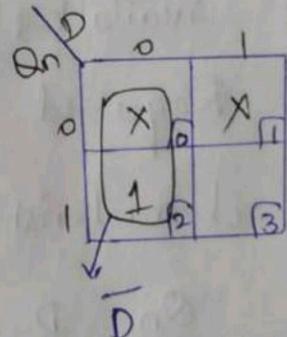
K-Map Simplification:

For J



$$J = D$$

For K

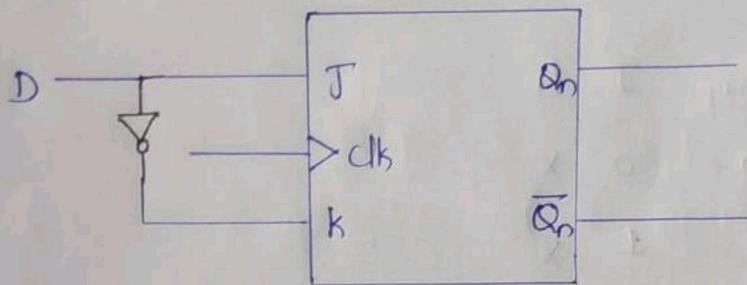


$$K = \overline{D}$$

Step 5: Logic diagram

Logic diagram

(31)



JK flipflop to T flipflop :-

Step 1 : Available flipflop is JK

Required flipflop is T

Step 2 : characteristic table for T flipflop

<u><math>Q_n</math></u>	<u>T</u>	<u><math>Q_{n+1}</math></u>
0	0	0
0	1	1
1	0	1
1	1	0

Step 3 : Excitation table for JK flipflop.

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>J</u>	<u>K</u>
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

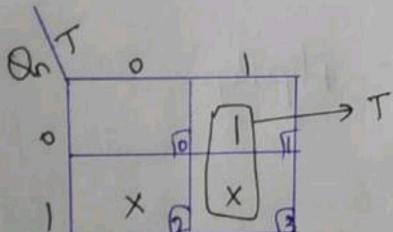
Step 4 : Conversion Table

(32)

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

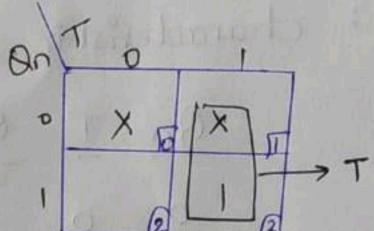
K-Map Simplification :

For  $J$ :



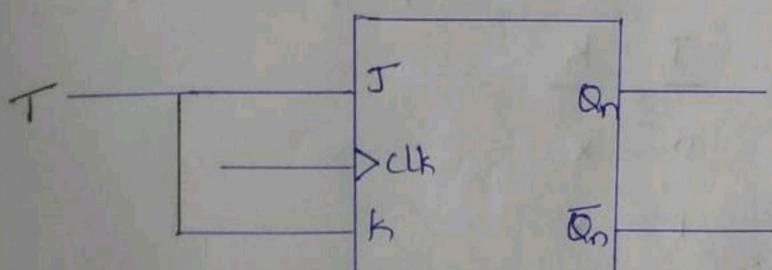
$$J = T$$

For  $K$ :



$$K = T$$

Step 5 : Logic diagram



D Flipflop to JK Flipflop:

(33)

Step 1:

Available Flipflop is D flipflop

Required Flipflop is JK flipflop

Step 2: characteristic table of JK flipflop

<u><math>Q_n</math></u>	<u><math>J</math></u>	<u><math>K</math></u>	<u><math>Q_{n+1}</math></u>
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

Step 3: Excitation Table of D flipflop

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u><math>D</math></u>
0	0	0
0	1	1
1	0	0
1	1	1

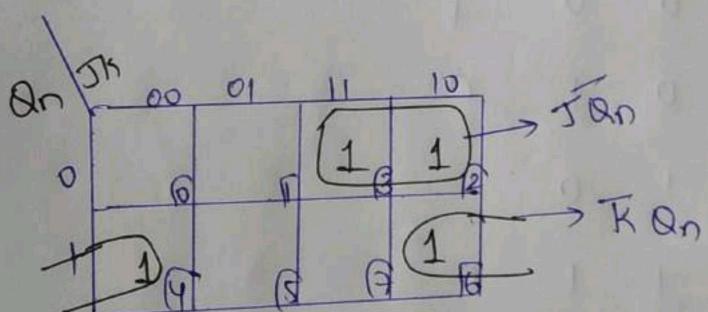
### Step 4 : Conversion table

(34)

<u><math>Q_n</math></u>	<u>J</u>	<u>K</u>	<u><math>Q_{n+1}</math></u>	<u>D</u>
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

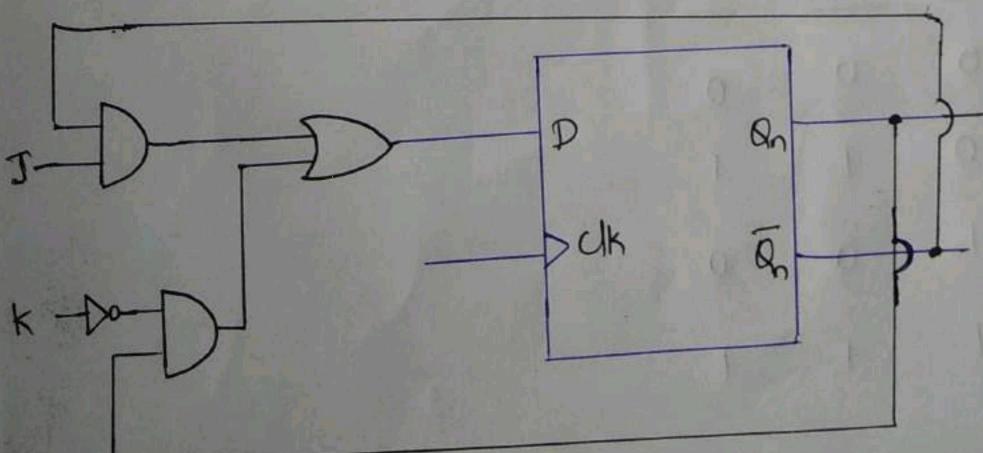
### K - Map Simplification:

For D



$$D = \overline{J} \overline{Q}_n + \overline{K} Q_n$$

### Step 5 : Logic diagram



D Flipflop to SR Flipflop :-

(35)

Step 1 : Available flipflop is D flipflop.

Required flipflop is SR flipflop.

Step 2 : characteristic table for SR flipflop.

<u><math>Q_n</math></u>	<u>S</u>	<u>R</u>	<u><math>Q_{n+1}</math></u>
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

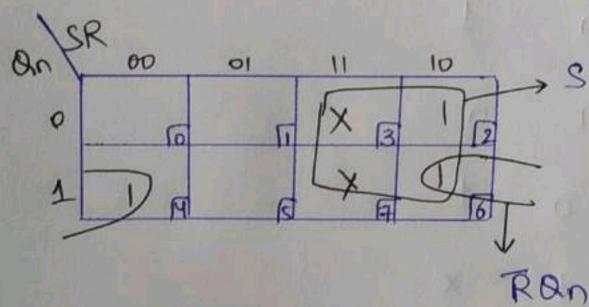
Step 3 : Excitation table for D - flipflop

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>D</u>
0	0	0
0	1	1
1	0	0
1	1	1

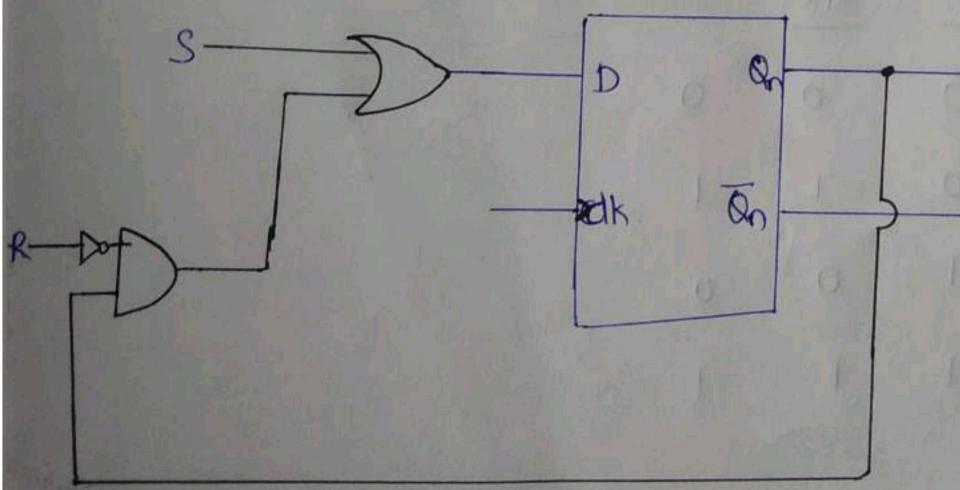
### Step 4 : Conversion Table

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

K-map



### Step 5 : Logic circuit



## D Flipflop to T- Flipflop :

(37)

Step 1 : Available flipflop is D-flipflop.

Required flipflop is T-flipflop.

Step 2 : characteristic table of T-flipflop.

<u><math>Q_n</math></u>	<u>T</u>	<u><math>Q_{n+1}</math></u>
0	0	0
0	1	1
1	0	1
1	1	0

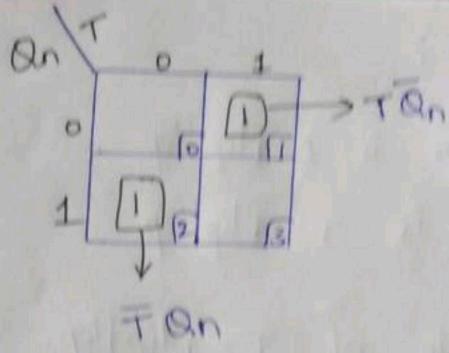
Step 3 : Excitation table of D-flipflop

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>D</u>
0	0	0
0	1	1
1	0	0
1	1	1

Step 4 : Conversion table

<u><math>Q_n</math></u>	<u>T</u>	<u><math>Q_{n+1}</math></u>	<u>D</u>
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

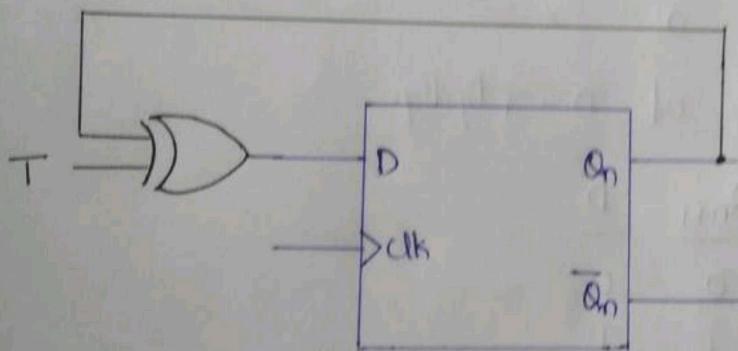
K-map Simplification :



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

$$= T \oplus Q_n$$

Step 5: Logic diagram



T flipflop to JK flipflop:

Step 4: Available flipflop is T flipflop.

Required flipflop is JK flipflop.

Step 2: characteristic Table of JK flipflop

(39)

<u><math>Q_n</math></u>	<u>J</u>	<u>K</u>	<u><math>Q_{n+1}</math></u>
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

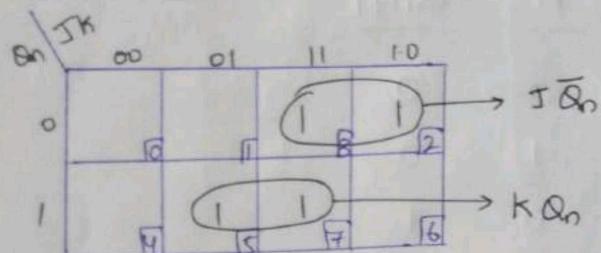
Step 3 : Excitation Table for T flipflop

<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>T</u>
0	0	0
0	1	1
1	0	1
1	1	0

Step 4 : Conversion Table

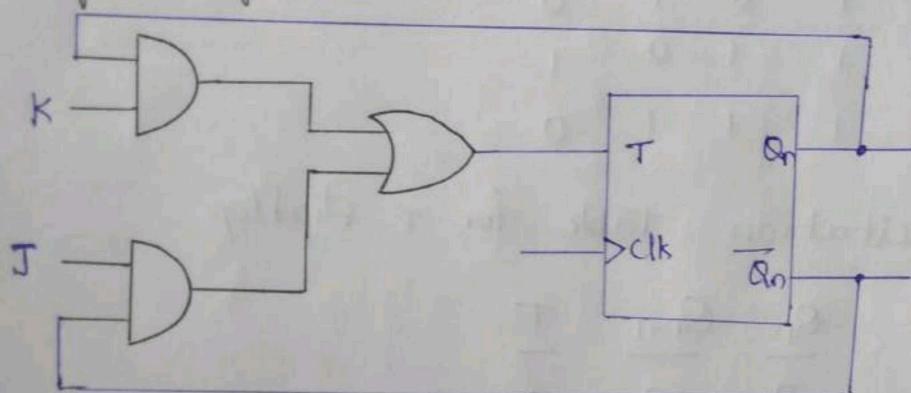
<u><math>Q_n</math></u>	<u>J</u>	<u>K</u>	<u><math>Q_{n+1}</math></u>	<u>T</u>
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

## K-Map Simplification:



$$T = \bar{J}Q_n + KQ_n$$

## Step 5: Logic diagram



## T flipflop to SR flipflop :-

Step 1: Available flipflop is T flipflop

Required flipflop is SR flipflop.

## Step 2: characteristic table for SR flipflop

<u><math>Q_n</math></u>	<u>S</u>	<u>R</u>	<u><math>Q_{n+1}</math></u>
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Step 3 : Excitation table for T

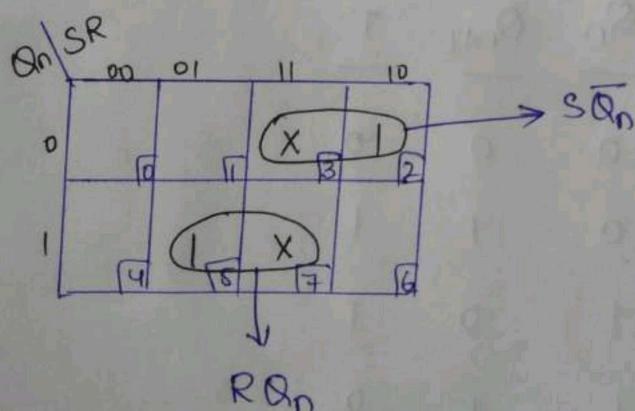
(1)

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

Step 4 : Conversion Table

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

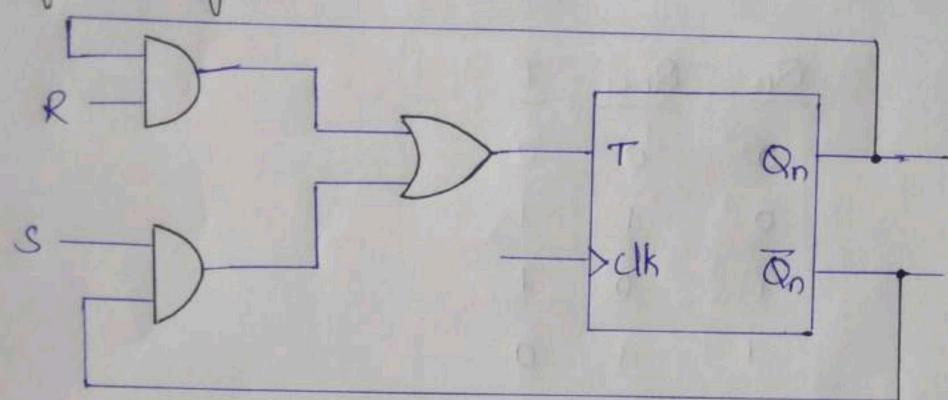
K-map Simplification:



$$T = S\bar{Q}_n + RQ_n$$

Step 5: Logic diagram

(42)



T-Flipflop to D-Flipflop :-

Step 1: Available flipflop is T flipflop

Required flipflop is D flipflop

Step 2: characteristic table of D flipflop

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

Step 3: Excitation table of T flipflop

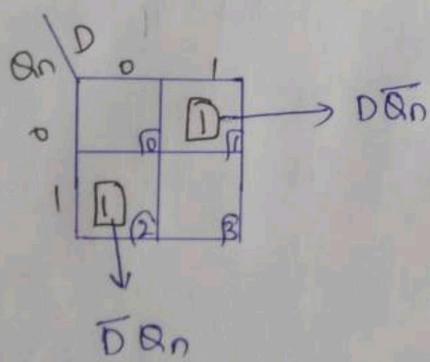
$Q_n$	$Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

#### Step 4: Conversion table

(4-3)

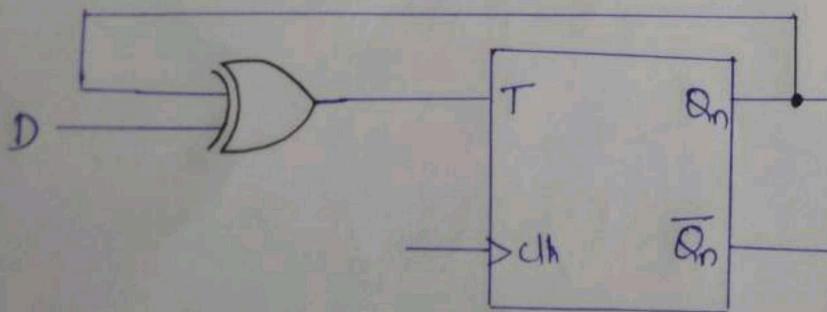
$Q_n$	$D$	$Q_{n+1}$	$T$
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

#### K-map Simplification:



$$\begin{aligned} T &= D\bar{Q}_n + \bar{D}Q_n \\ &= D \oplus Q_n \end{aligned}$$

#### Step 5: Logic diagram



In J-K flipflop when input is 1 (i.e.,  $J=1, K=1$ ), the Q/p continues to toggle from 0 to between 1 and 0 is known as 'Race Around condition'. This condition can be avoided if the flip flop is made to toggle over one clock period.

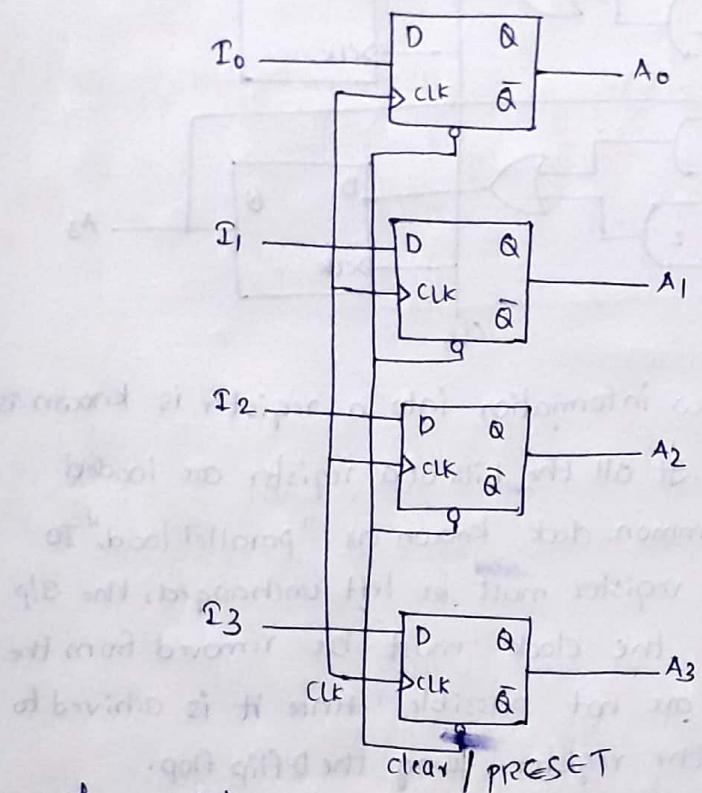
Applications of flipflops:

1. Memory
2. Registers
3. Counters.

Registers: A register is group of flip flops each one of which is capable of storing 1 bit of binary information. An n-bit register consist of  $n$  flip flops <sup>it stores  $n$  bits of information.</sup> In addition to flip flops a register may have combinational gates.

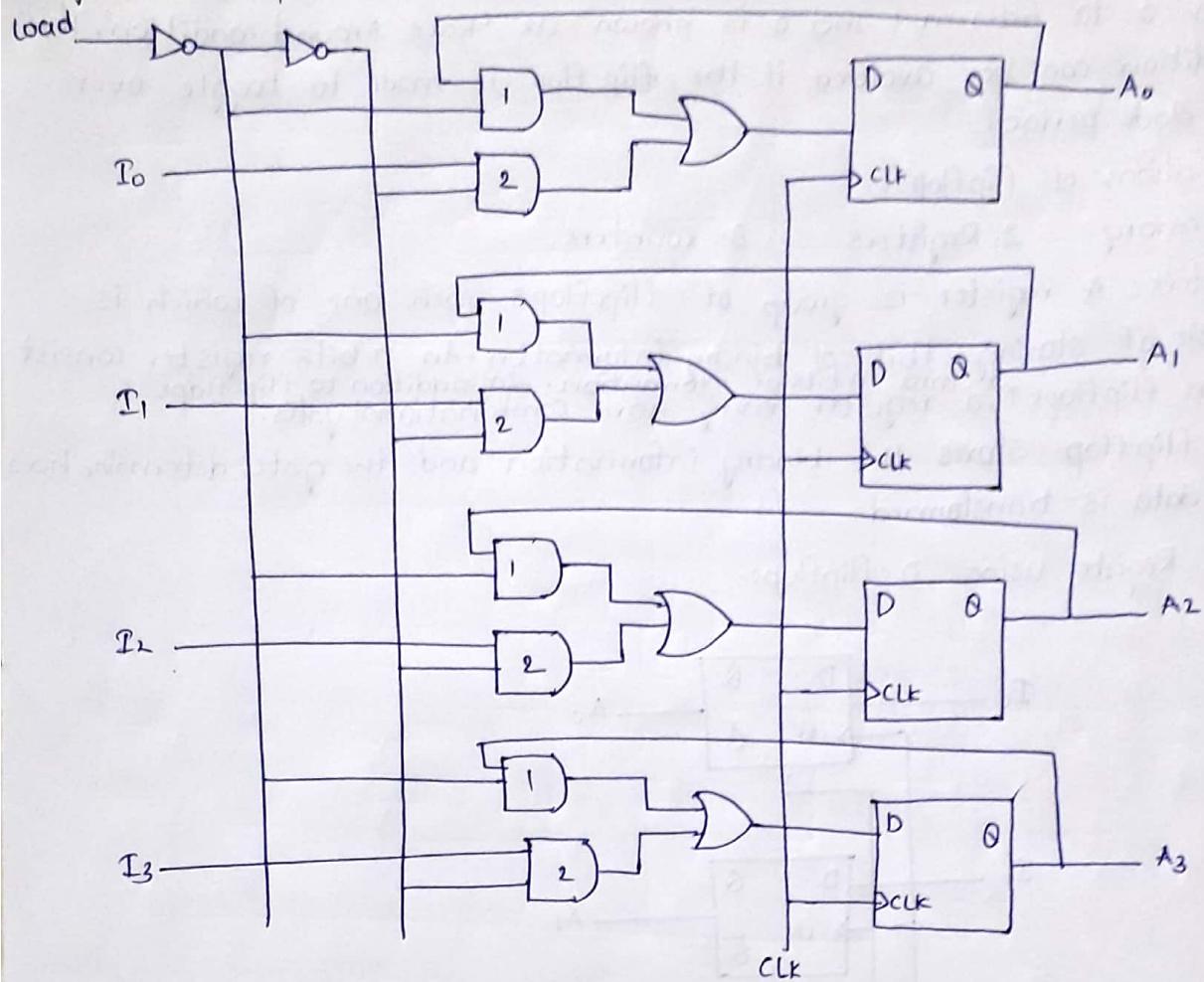
The flip flop stores the binary information and the gates determine how the data is transformed.

4bit Register using D-flipflop:-



The register shown above is a simple register which consist of only flip flop. It is a 4 bit register constructed with 4 D-flipflops and can store 4 bits of binary information. The clock input is common which triggers all the flipflops on the positive edge of the clock. The input 'clear' is a active low reset s/p when this s/p gate to zero. It resets the register.

Register with parallel load:-



\* The transformation of new information into a register is known as "loading or updating data". If all the bits of a register are loaded simultaneously with a common clock known as "parallel load." To make the contents of the register must be left unchanged, the  $g/p$  must be held constant or the clock must be removed from the circuit. These two cases are not possible. Hence it is achieved to control the operation of the register using the D-flip flop.

\* When load  $g/p$  is '1', the data at the  $g/p$ s are transferred into register. When load is '0', the o/p of the flip flops are connected to the respective flip flop's inputs.

**Shift Register**:- A register capable of shifting the binary information in the specified direction is known as shift register. The construction of shift register consist of group of flip-flops, output of each flip-flop is connected as an input of next flip flop.

There are 4 types of shift registers.

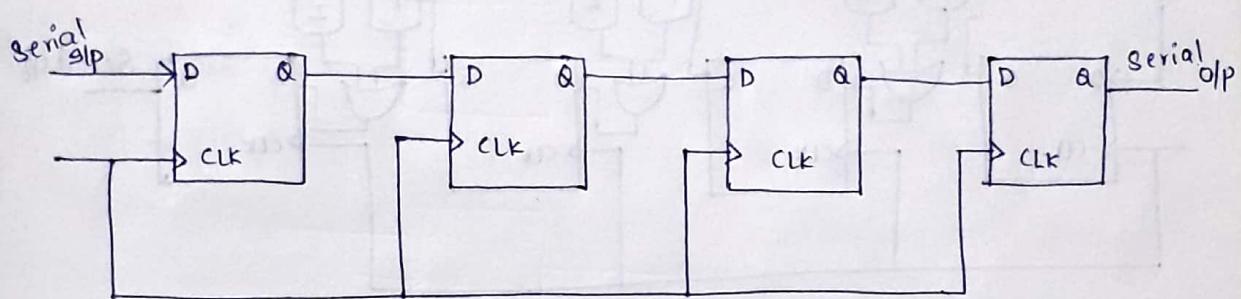
(1) serial input & serial output.

(2) serial input & parallel output

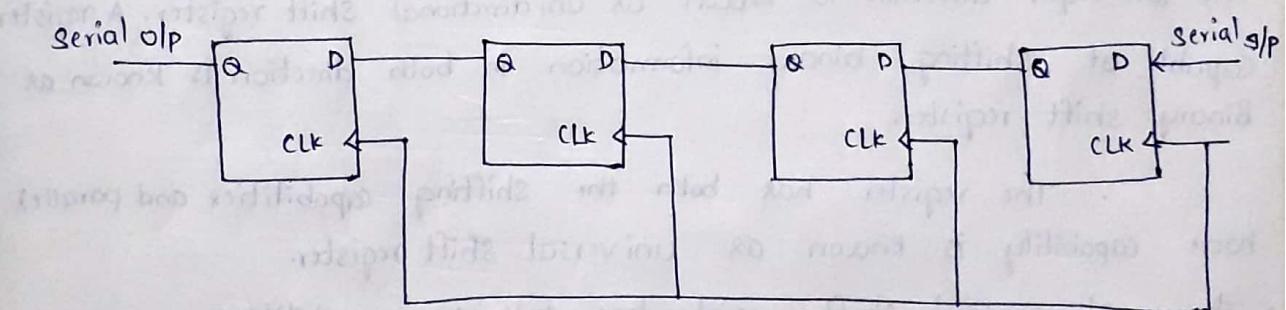
(3) parallel input & serial output

(4) parallel input & parallel output

1. Serial Input Serial Output (SISO):-



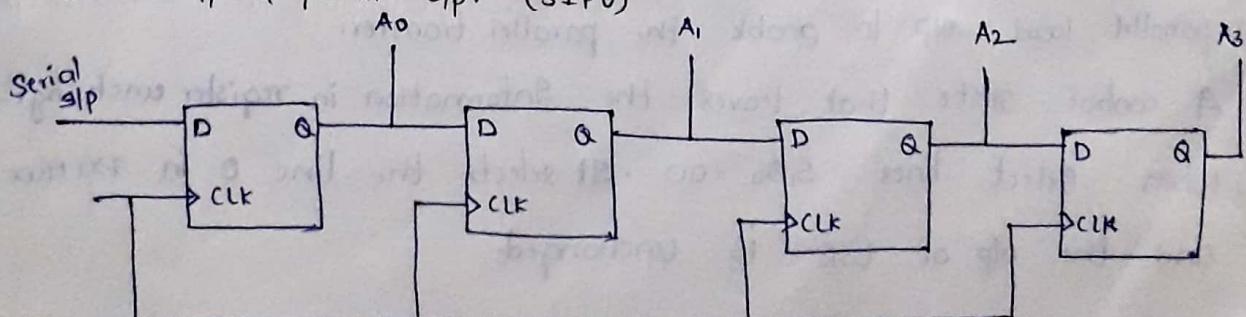
right shift register

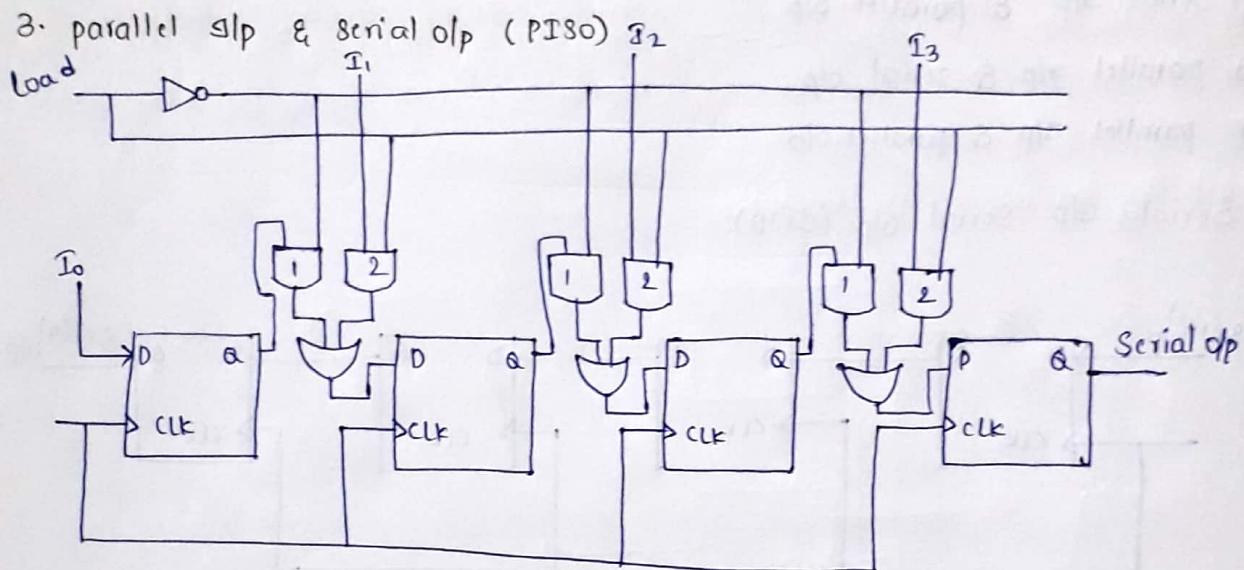
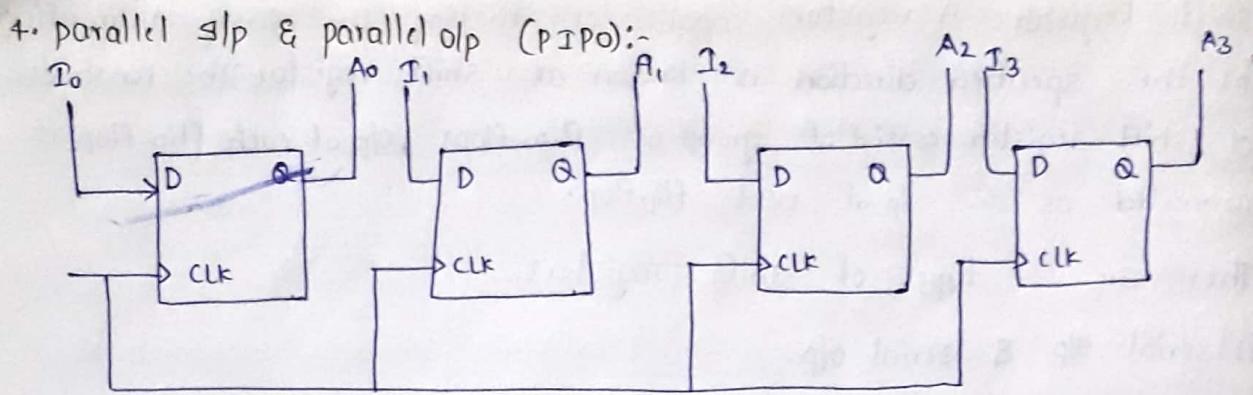


left shift register.

The figs shown above are simple shift registers as they are constructed with only flipflops. The shift registers shown above are unidirectional, i.e., they shift the data either in right direction (or) left direction.

2. Serial Input & parallel Output:- (SIPO)





### UNIVERSAL SHIFT REGISTER:- (USR)

A Register capable of shifting the binary information in left (or right direction is known as unidirectional shift register. A register capable of shifting binary information in both direction is known as Binary shift register.

The register has both the shifting capabilities and parallel load capability is known as universal shift register.

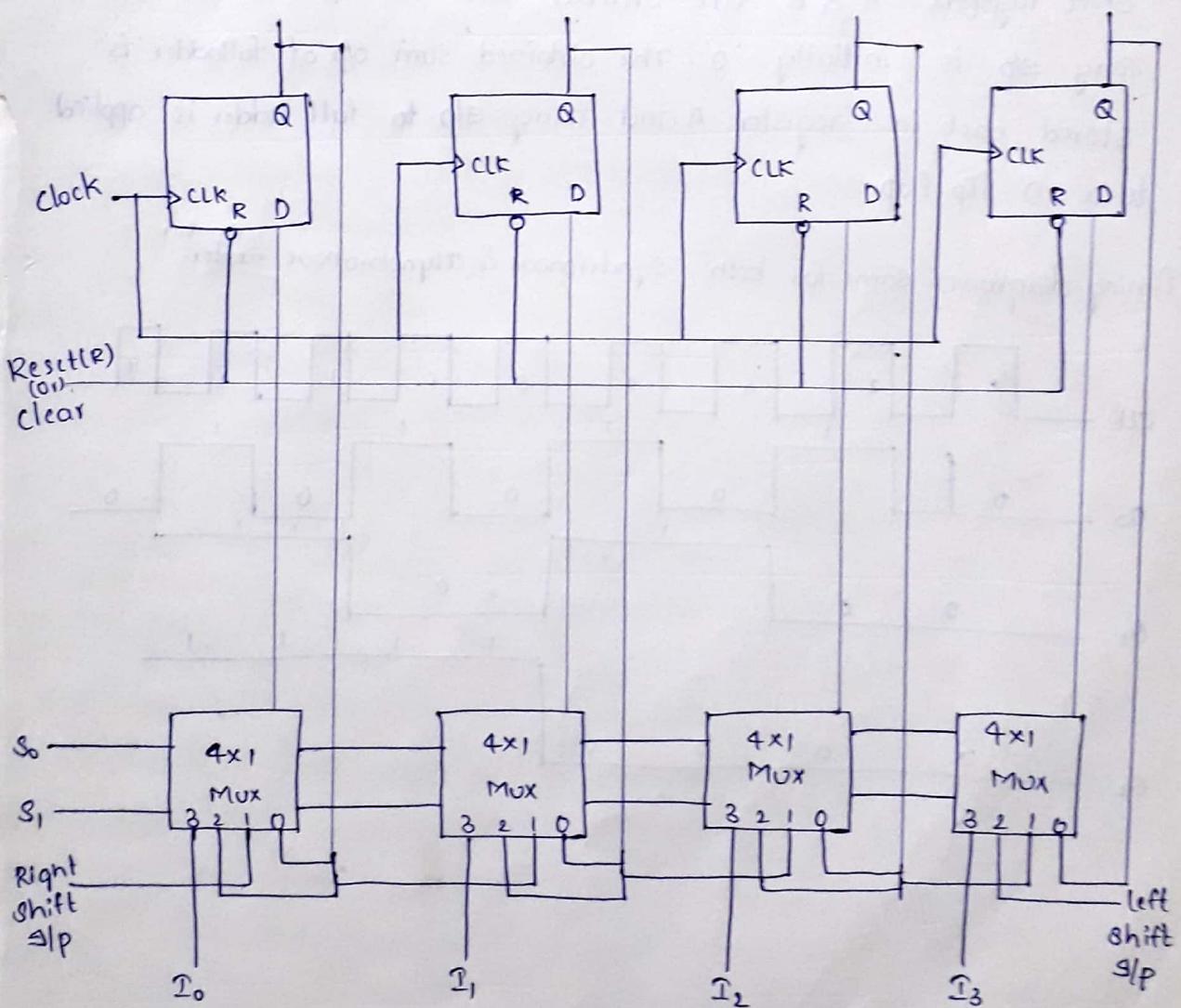
\* The most general shift register has following capabilities.

- (1) clear slp to clear the shift register.
- (2) A clock slp to synchronise the operation.
- (3) shift right control to enable right shift operation.
- (4) left shift control to enable left shift operation.
- (5) parallel load slp to enable the parallel transfer.
- (6) A control state that leaves the information in register unchanged.
- (7) When select lines  $S_1, S_0 = 00$  it selects the line 0 in  $4 \times 1$  Mux and the o/p of USR is unchanged.

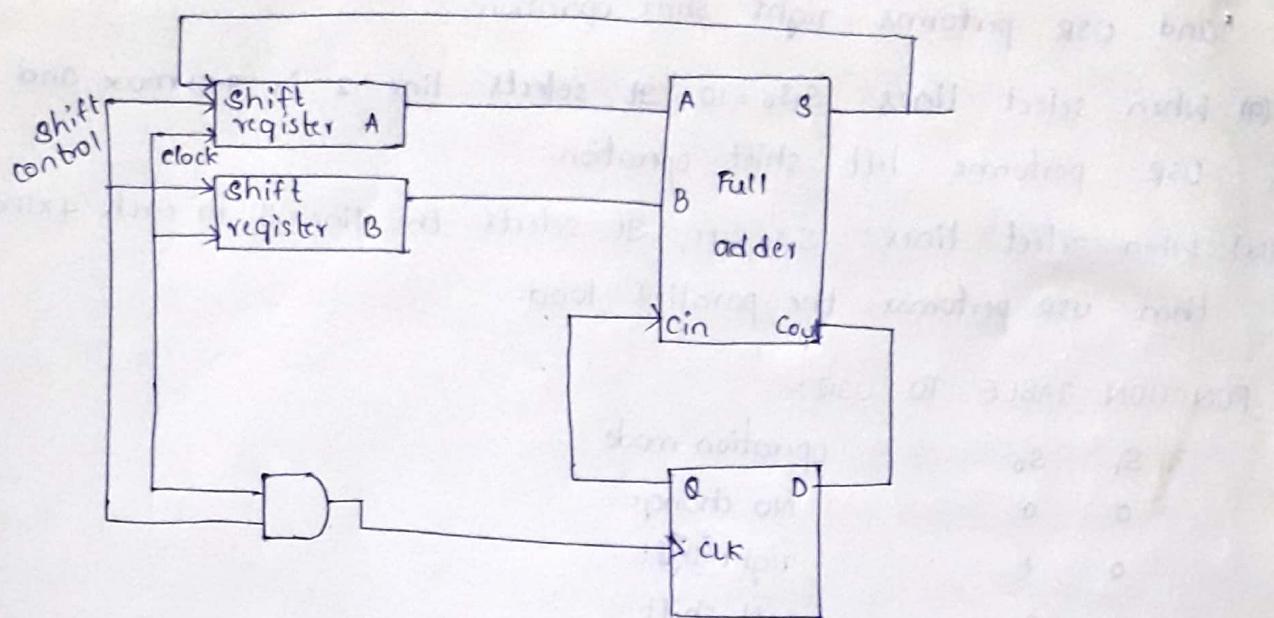
- (8) When select lines  $S_1 S_0 = 01$ , it selects the line 1 in  $4 \times 1$  MUX and USR performs right shift operation.
- (9) When select lines  $S_1 S_0 = 10$ , it selects line 2 in  $4 \times 1$  MUX and USR performs left shift operation.
- (10) When select lines  $S_1 S_0 = 11$ , it selects the line 3 in each  $4 \times 1$  MUX then USR performs the parallel load.

FUNCTION TABLE TO USR:

$S_1$	$S_0$	operation mode
0	0	No change
0	1	right shift
1	0	Left shift
1	1	parallel load

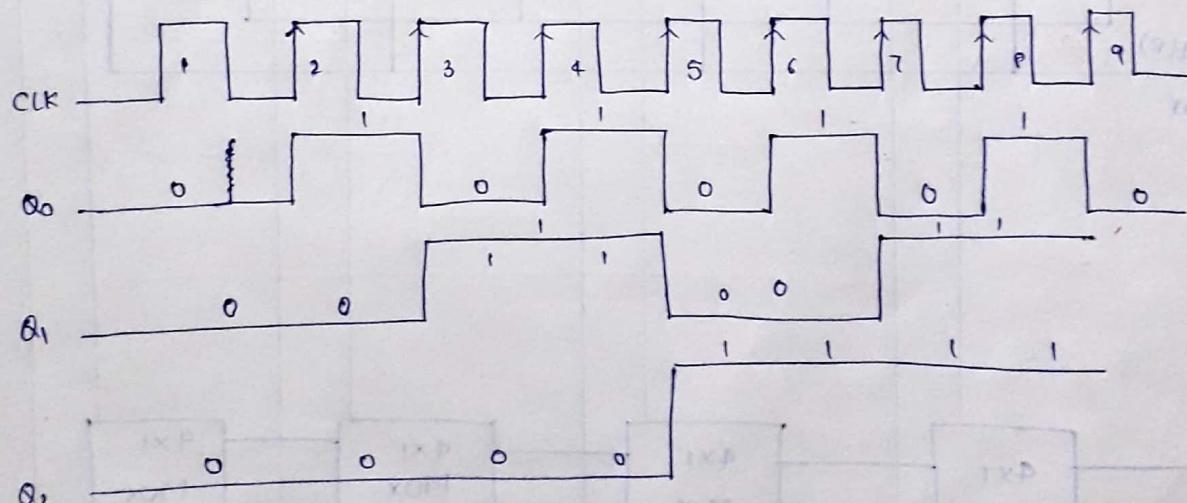


### Serial adder:-



\* When clock is applied and shift control is active then data in Shift registers A & B are shifted out as S/p's to full adder and carry S/p is initially '0'. The obtained sum o/p of full adder is stored back in register A and carry S/p to full adder is applied to a D-flip flop.

Timing diagram:- same for both synchronous & asynchronous Counter.



# Timing Diagram for Shift Register (SISO) / (SIPO) & (PIPO) / (PISO)

(Pin) Input = "1111"

clk	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
Initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

Table: SISO & SIPO

SISO → upto 8 clock pulses  
to get output

SIPO → 4 clock pulses to get output.

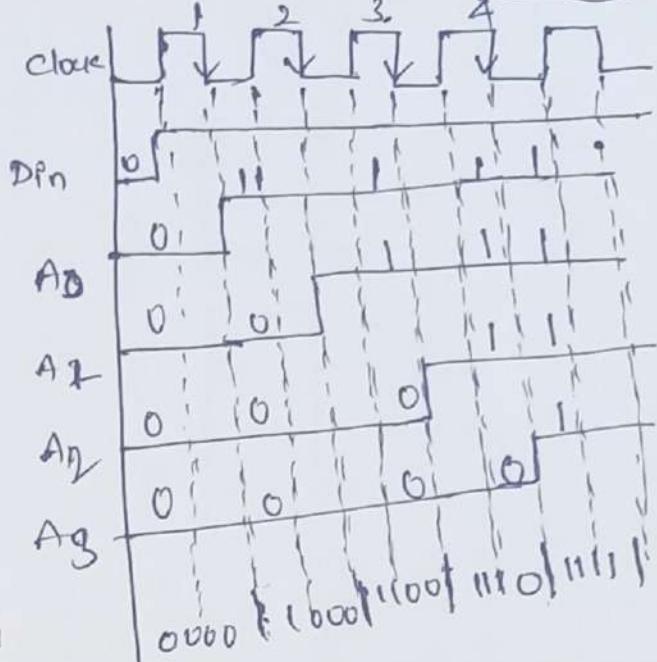


fig: SISO & SIPO

Table: PIPO

clk	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>
Initially	0	0	0	0	0	0	0	0
↓	1	1	1	1	1	1	1	1

fig: PIPO.

PIPO - only one clock cycle.

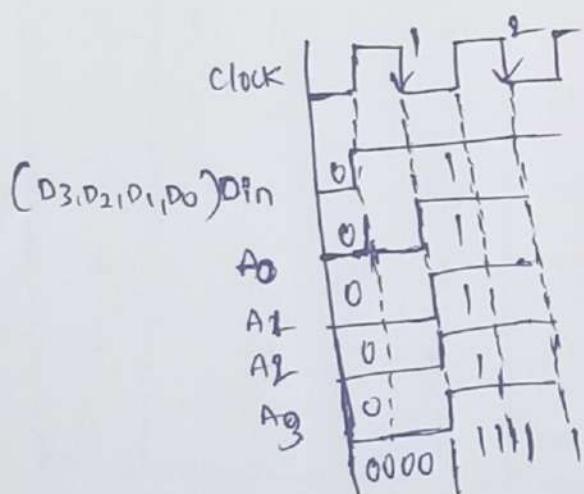
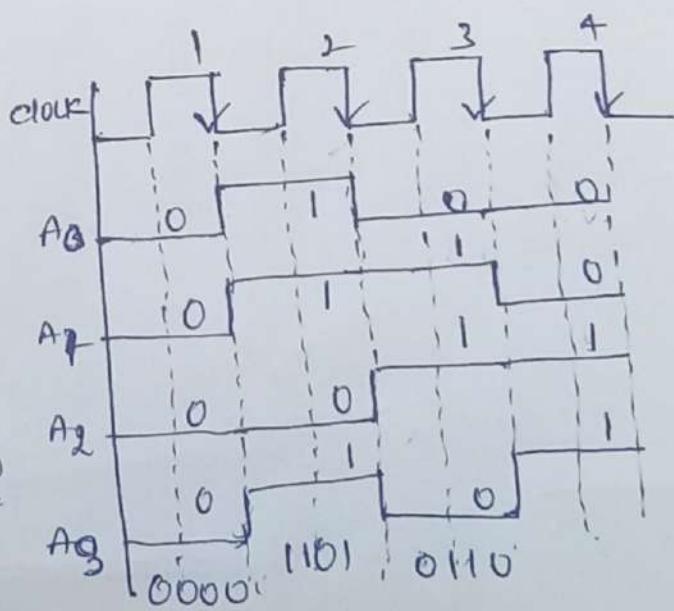


Table: PISO

clk	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	Data
Initially	0	0	0	0	"1101"
↓	1	1	0	1	
↓	0	1	1	0	
↓	0	0	1	1	

fig:  
PISO



A register that goes through a predetermined sequence of states upon the application of input pulses is called a counter. The input pulses may be clock pulses, or they may be from external source.

### Binary Counter

A counter that follows the binary number sequence is called a binary counter. An n-bit binary counter consists of n flipflops & can count  $2^n$  values.

### Types of counters

1. Ripple counter / asynchronous counters
2. Synchronous counters.

In ripple counters output of one flipflop is given as a count (C) input of next flipflop. No clock in ripple counters. In synchronous counters all flipflops receive the common clock.

### MOD Number / Modulus of the counter

Number of different logic states a counter goes through before it comes to the initial state is called MOD number.

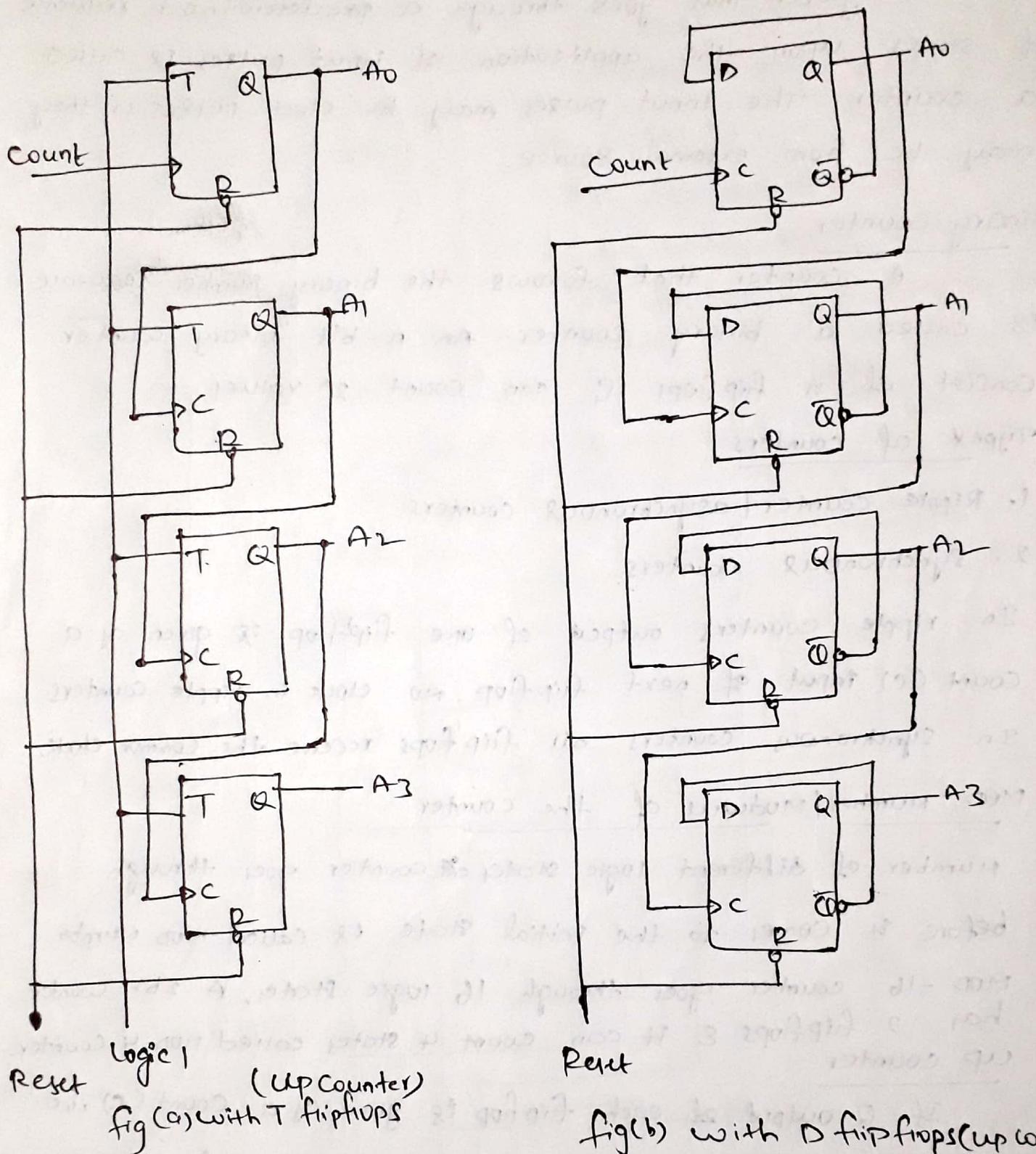
MOD -16 counter goes through 16 logic states. A 2-bit counter has 2 flipflops & it can count 4 states called MOD-4 counter.

If Q output of each flipflop is given as a count (C), i.e up counter counts the values in the order of 0, 1, 2, 3, 4, 5, 6, 7 etc.,

### Down Counter

If  $\bar{Q}$  output of each flipflop is given as a count (C), i.e down counter counts the values in the order of 7, 6, 5, 4, 3, 2, 1, 0 ..

## 4-bit binary ripple counter



fig(a) & fig(b) shows a 4 bit counter it can count  $2^4 = 16$  states i.e 0 to 15. The operation of 4-bit counter is whenever  $A_0$  goes from 1 to 0,  $A_1$  is complemented. whenever  $A_1$  goes from 1 to 0 it complements  $A_2$ . whenever  $A_2$  goes from 1 to 0 it complements  $A_3$ . For example consider a binary count sequence in table below.

Example

Binary count sequence

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	1	0
0	0	0	0
1	0	0	0

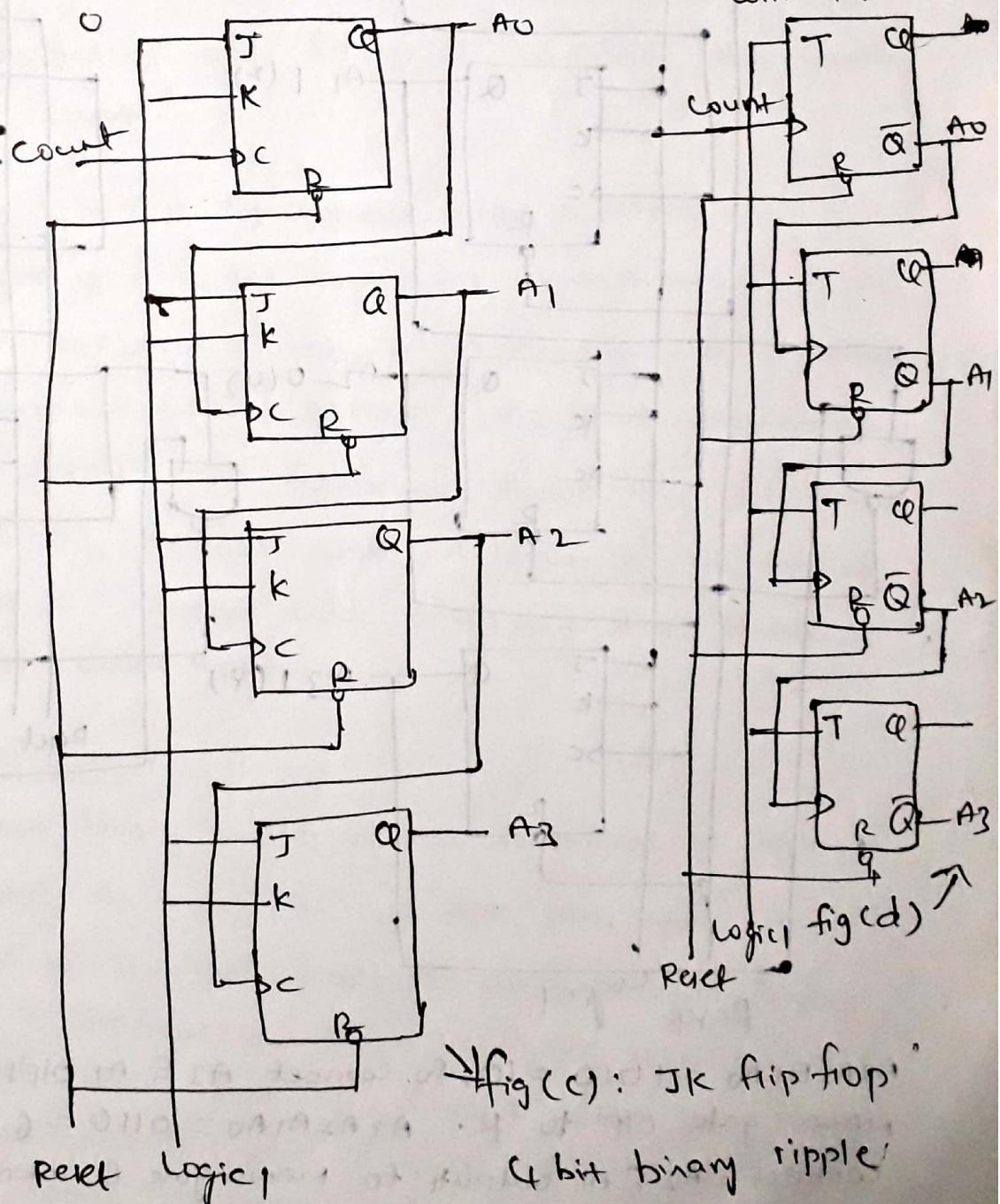
In fig(a) if  $\bar{Q}$  is connected to

C, that is called a down counter as shown in fig(d)

In fig(a) if we replace T by

JK flip flop, then it is a 4 bit binary ripple counter with JK flip flop it is shown in fig(c) below.

fig(d):- 4 bit binary ripple Counter (down counter) with T flip flops



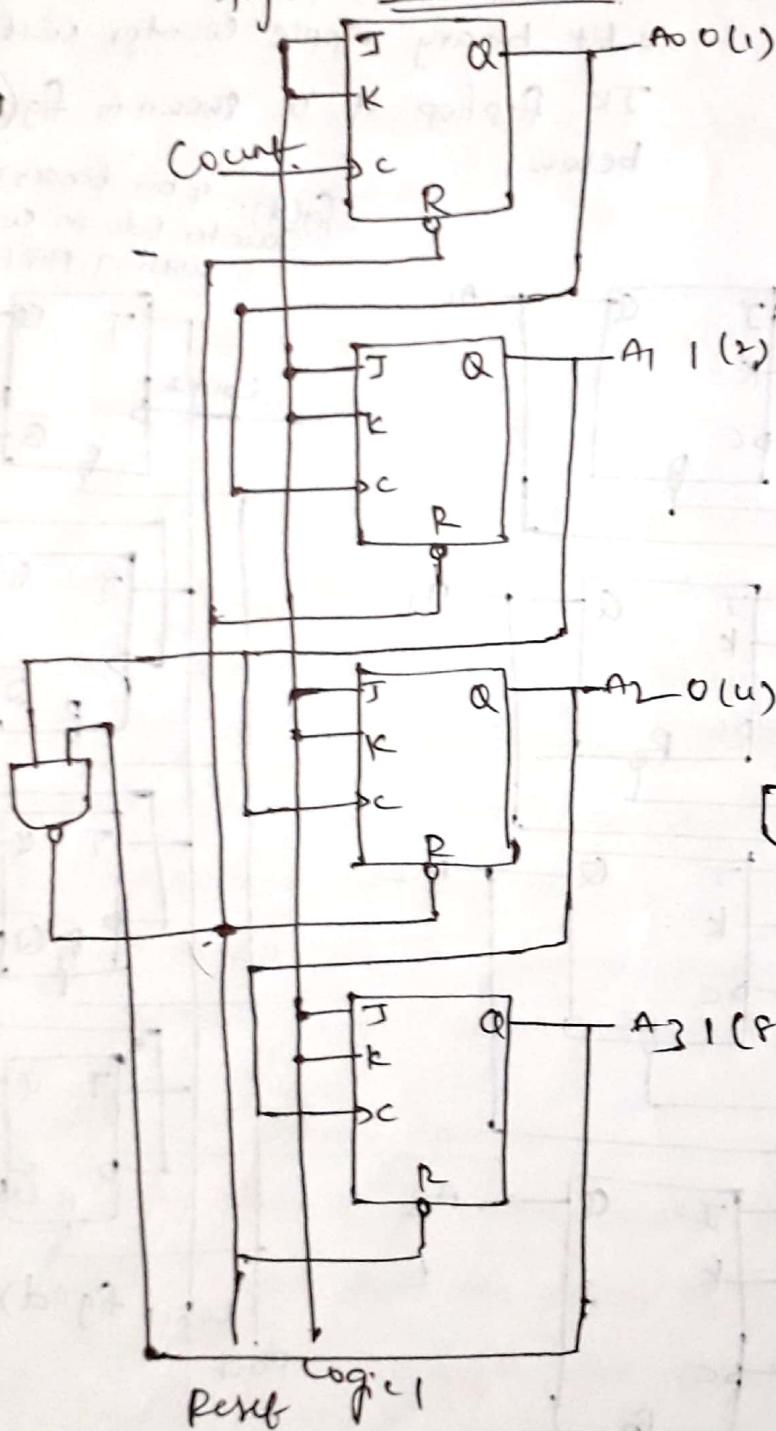
Fig(c): JK flip flop

4 bit binary ripple counter (up counter).

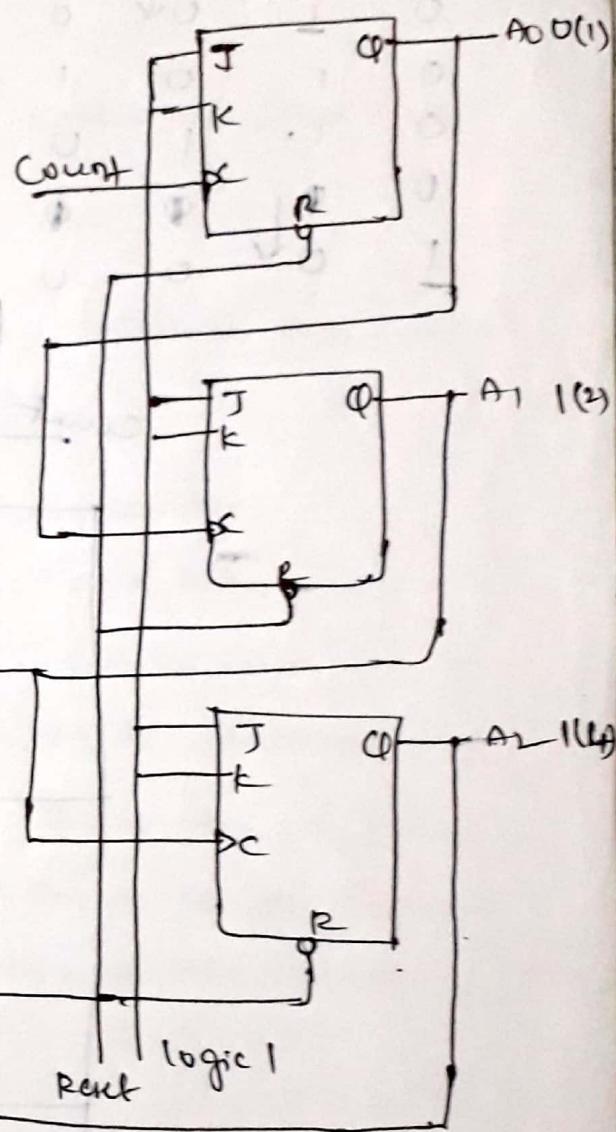
# BCD Counter / BCD ripple Counter / decimal counter / MOD-10 counter

A BCD counter follows a sequence of 10 states from 0 to 9 and returns to 0. This counter must have atleast 4 flipflops to count 10 states because 3 flipflops can count only  $2^3 = 8$  states.

fig(a): MOD-10 Counter



fig(b): MOD-6 Counter



$A_3 A_2 A_1 A_0 = 1010 = 10$ , so connect  $A_3$  &  $A_1$  OLP's to NAND gate & NAND gate OLP to R.  $A_3 A_2 A_1 A_0 = 0110 = 6$ , i.e MOD 6 Counter, connect  $A_2$  &  $A_1$  outputs to NAND gate & NAND gate output to R. 3 flipflops are enough for MOD-6, because 3 flipflops can count  $2^3 = 8$  states (8 MORS). MOD-6 requires only 6 states.

## Synchronous Counters

Synchronous counters are different from ripple counters (asynchronous counters). In ripple counters output of first flip flop is given as a input (clock) of next flip flop. In synchronous counters common clock pulse is applied to all flip flops.

### Up Counter

If Q output of each flip flop is considered, then counter performs up count.

### Down Counter

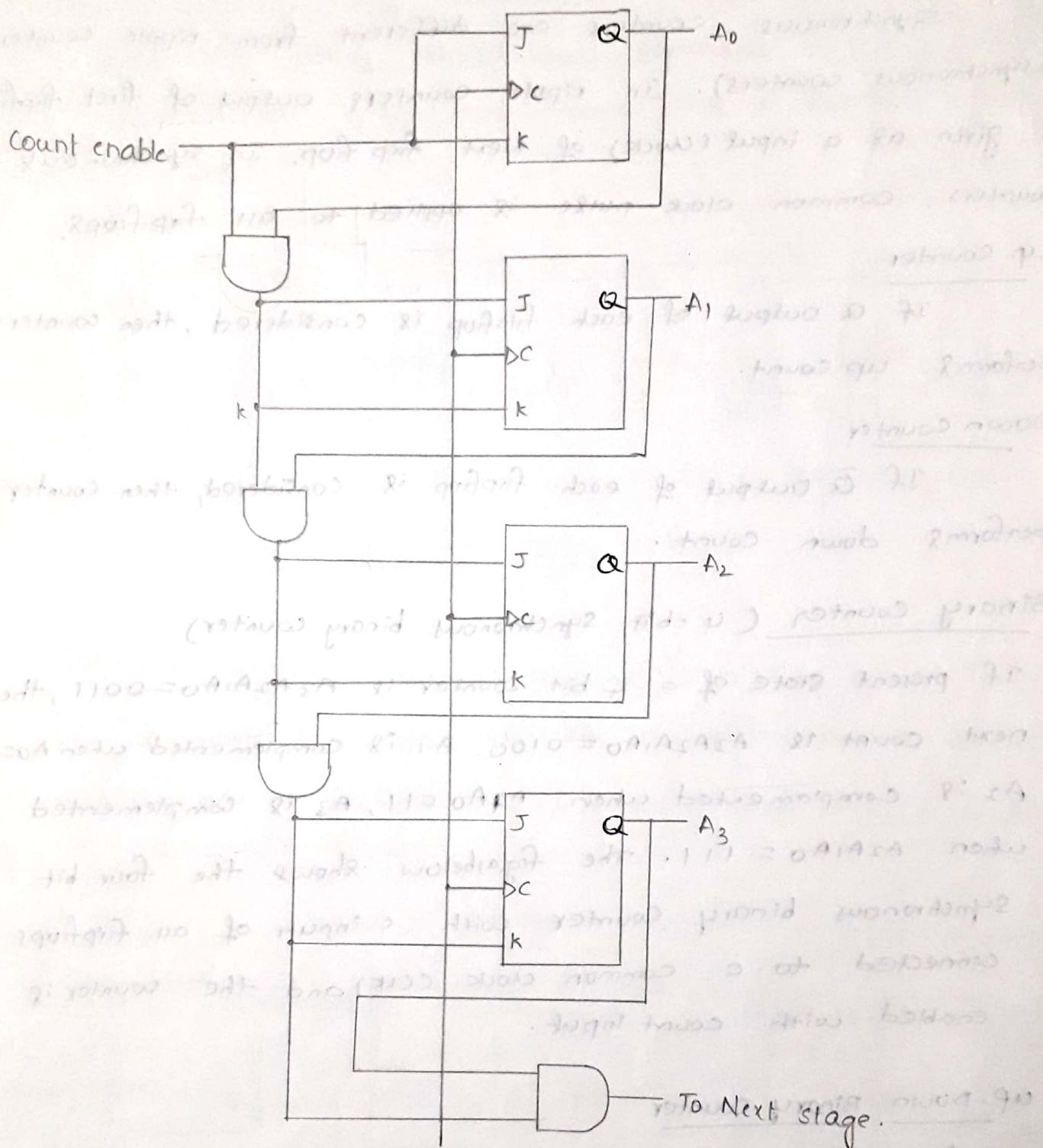
If  $\bar{Q}$  output of each flip flop is considered, then counter performs down count.

### Binary Counter (4-bit Synchronous binary counter)

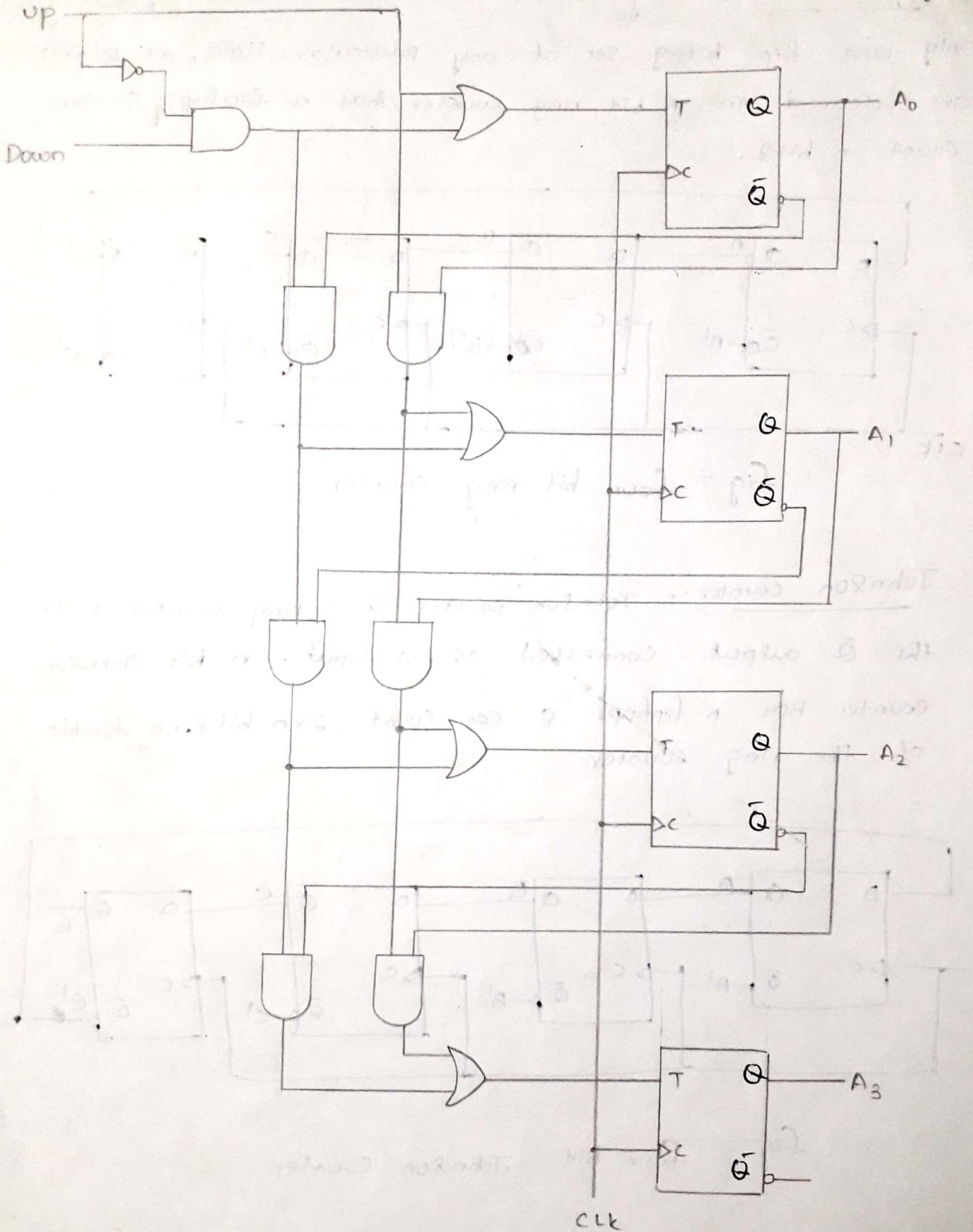
If present state of a 4 bit counter is  $A_3A_2A_1A_0 = 0011$ , the next count is  $A_3A_2A_1A_0 = 0100$ .  $A_1$  is complemented when  $A_0 = 1$ ,  $A_2$  is complemented when  $A_2A_0 = 11$ ,  $A_3$  is complemented when  $A_2A_1A_0 = 111$ . The fig(a) below shows the four bit synchronous binary counter with  $\bar{Q}$  inputs of all flip flops connected to a common clock (CLK) and the counter is enabled with count input.

### Up-Down Binary Counter

A count down binary counter can be constructed as shown in fig(a), except that the inputs to the AND gates must come from  $\bar{Q}$ , instead of  $Q$ . The circuit of an up-down binary counter using T flip flops as shown in fig(b). When the UP input is 1, the circuit counts up, since T receives input from  $Q$ . When the DOWN input is 1 and the UP input is 0, the circuit counts down, since T receives input from  $\bar{Q}$ .



fig(a) four bit Synchronous binary counter (up counter)



fig(b) four bit up-down binary counter

# Timing Diagram for counters

Table

Clock	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Count
Initially	0	0	0	0	0
↓	0	0	0	1	1
↓	0	0	1	0	2
↓	0	0	1	1	3
↓	0	1	0	0	4
↓	0	1	0	1	5
↓	0	1	1	0	6
↓	0	1	1	1	7
↓	1	0	0	0	8
↓	1	0	0	1	9
↓	1	0	1	0	10
↓	1	0	1	1	11
↓	1	1	0	0	12
↓	1	1	0	1	13
↓	1	1	1	0	14
↓	1	1	1	1	15

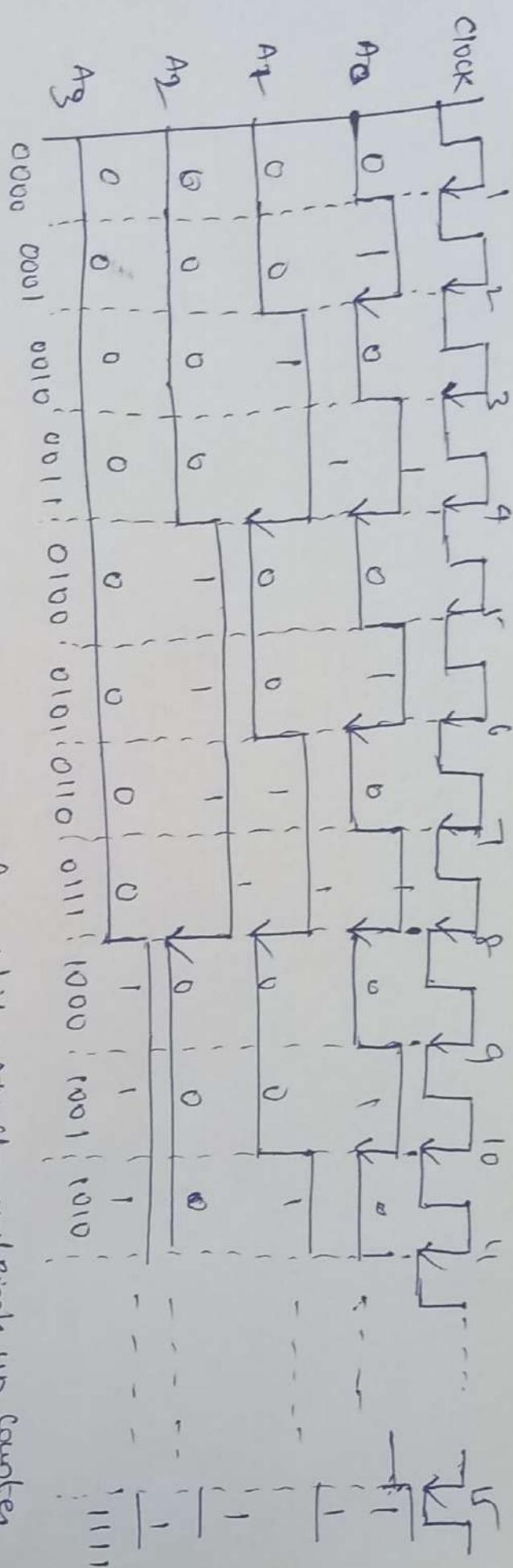
Table. 4 bit Asynchronous /

Ripple up Counter and

Synchronous up counter.

Note:- for 3 bit counter, only the outputs A<sub>2</sub>, A<sub>1</sub>, A<sub>0</sub> are taken upto the count value '7'.

fig: Timing diagram for 4 bit asynchronous ripple up counter and synchronous up counter



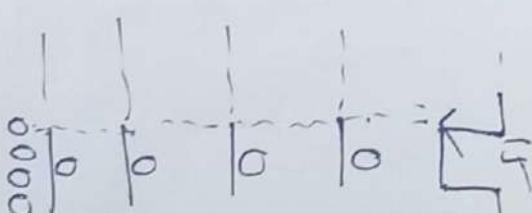
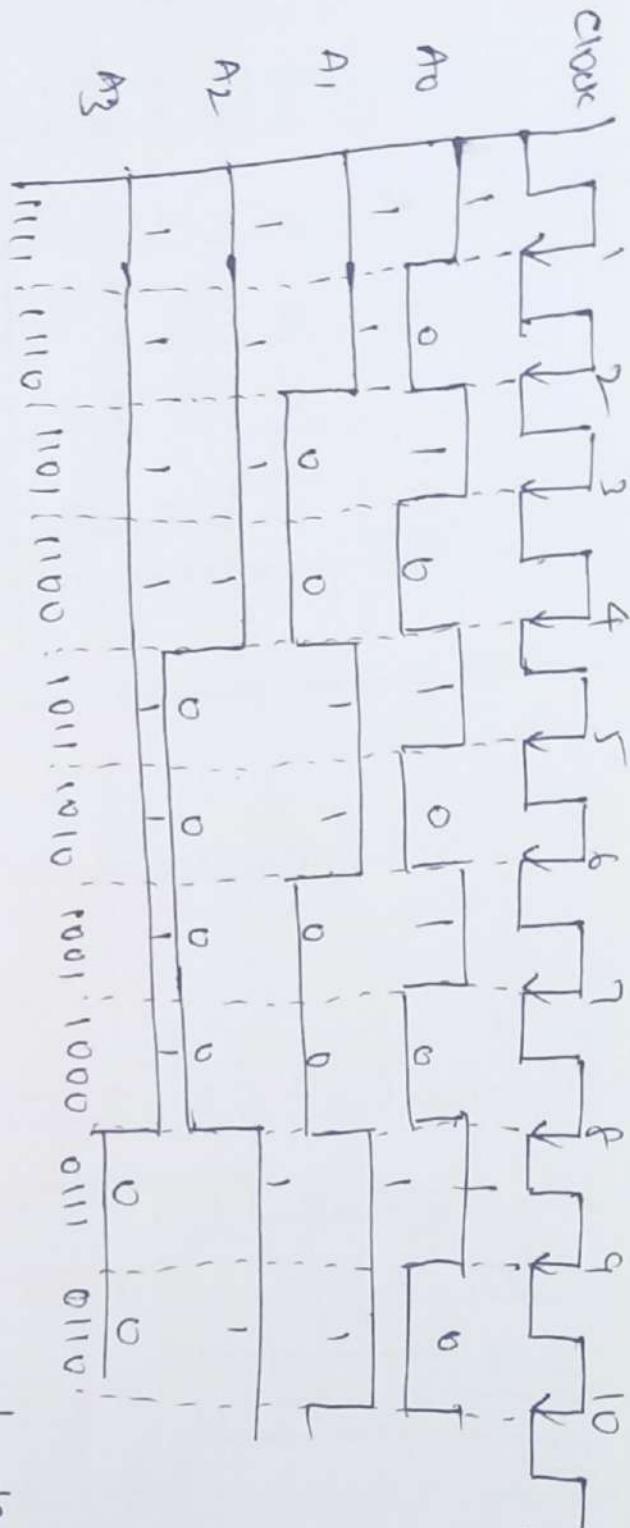
Table

clock	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Count
Initial	1	1	1	1	0
↓	1	1	1	0	1
↓	1	1	0	1	2
↓	1	1	0	0	3
↓	1	0	1	1	4
↓	1	0	1	0	5
↓	1	0	0	1	6
↓	1	0	0	0	7
↓	0	1	1	1	8
↓	0	1	1	0	9
↓	0	1	0	1	10
↓	0	1	0	0	11
↓	0	0	1	1	12
↓	0	0	1	0	13
↓	0	0	0	1	14
↓	0	0	0	0	15

Table: 4 bit Asynchronous/Ripple down counter, and Synchronous down counter.

Note: If the counter is MOD-10, write the table & timing diagram for first 10 clock pulses (0 to 9). If, for MOD-6 counts 6 clock pulses (0 to 5).

fig: Timing diagram for 4 bit Asynchronous/Ripple down counter and synchronous down counter



## Other counters

Ring counter :- A ring counter is a circular shift registers with only one flip being set at any particular time, all others are cleared. An n bit ring counter has n flipflops & can count n bits.

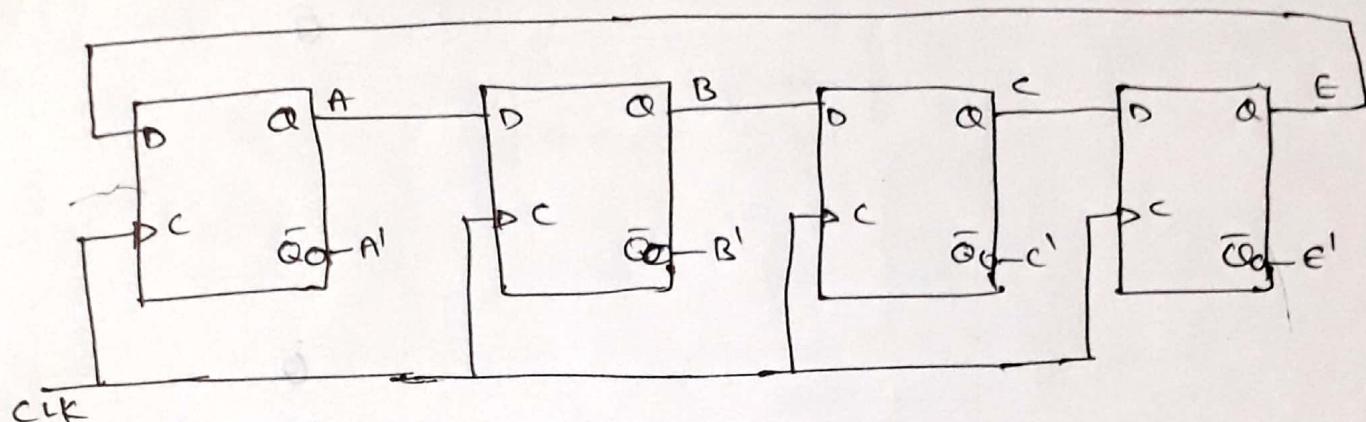


fig: four bit ring counter

Johnson counter :- Johnson counter is a ring counter with the  $\bar{Q}$  output connected as a input. n bit Johnson counter has n flipflops & can count  $2 \times n$  bits, i.e double of the ring counter.

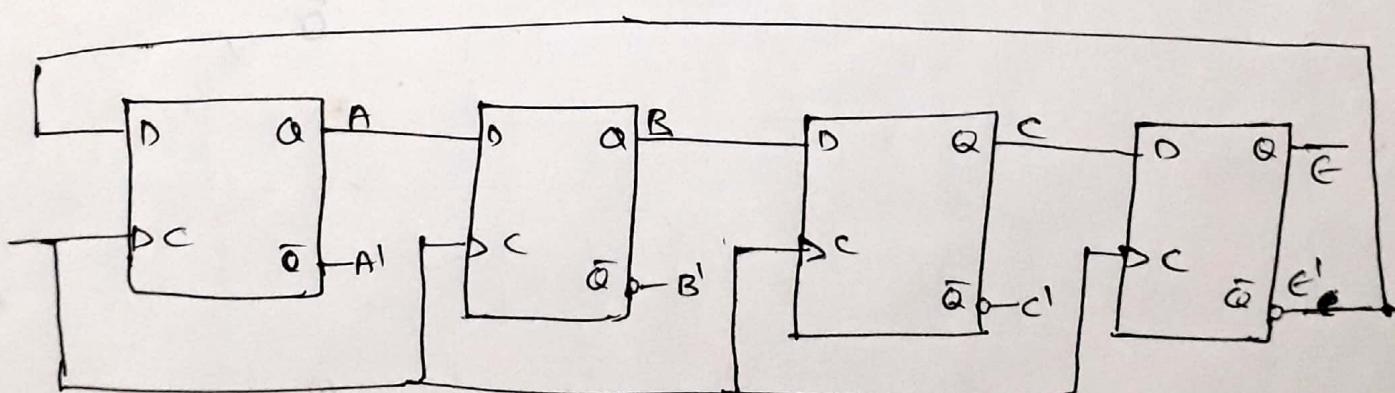


fig: four bit Johnson counter

## Ring Counter

predefined value = "1000"

Table

CLK	A	B	C	E
X	1	0	0	0
↓	0	1	0	0
↓	0	0	1	0
↓	0	0	0	1
↓	1	0	0	0

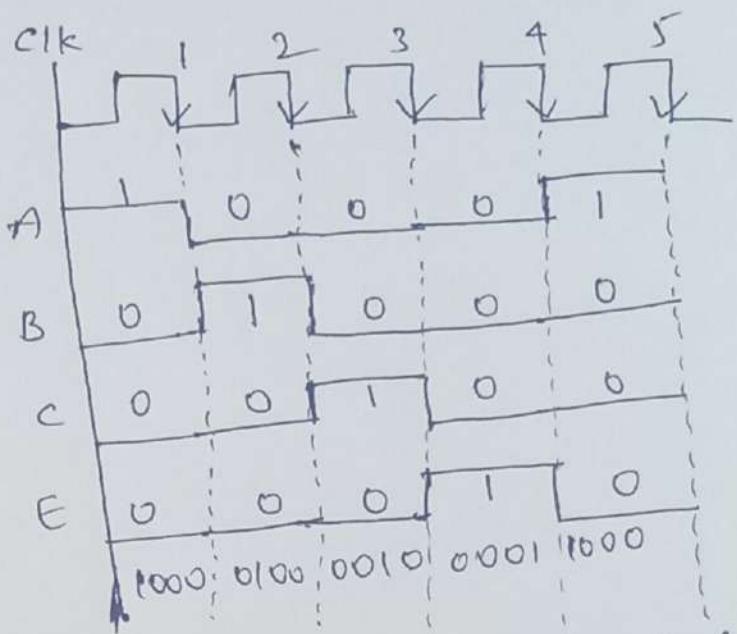


fig: Timing diagram for ring counter

## Johnson's Counter (Twisted / switch Tail Ring Counter)

Table

CLK	A	B	C	E
X	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1
↓	0	1	1	1
↓	0	0	1	1
↓	0	0	0	1
↓	0	0	0	0

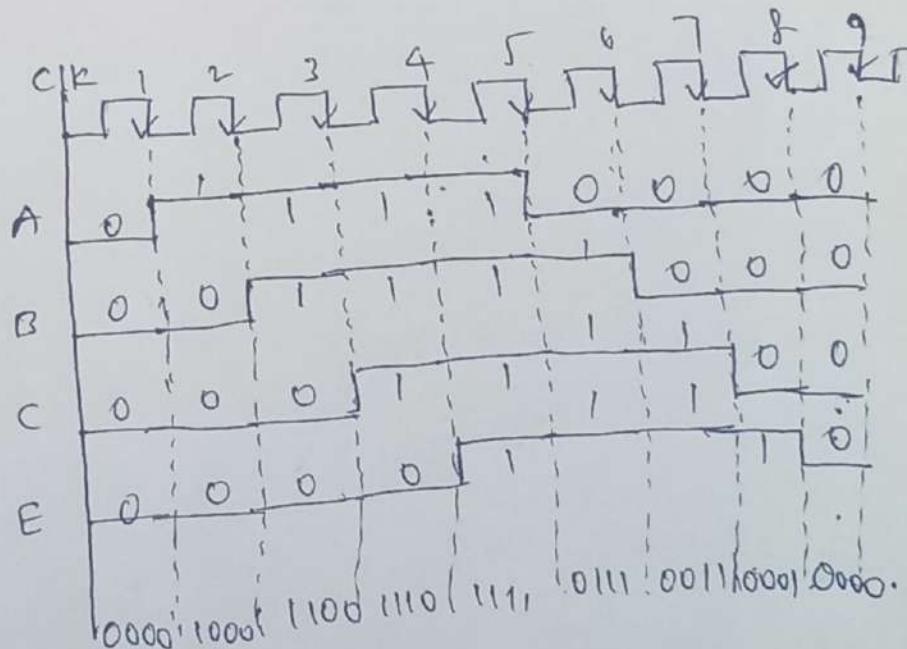


fig: Timing diagram for Johnson's Counter.