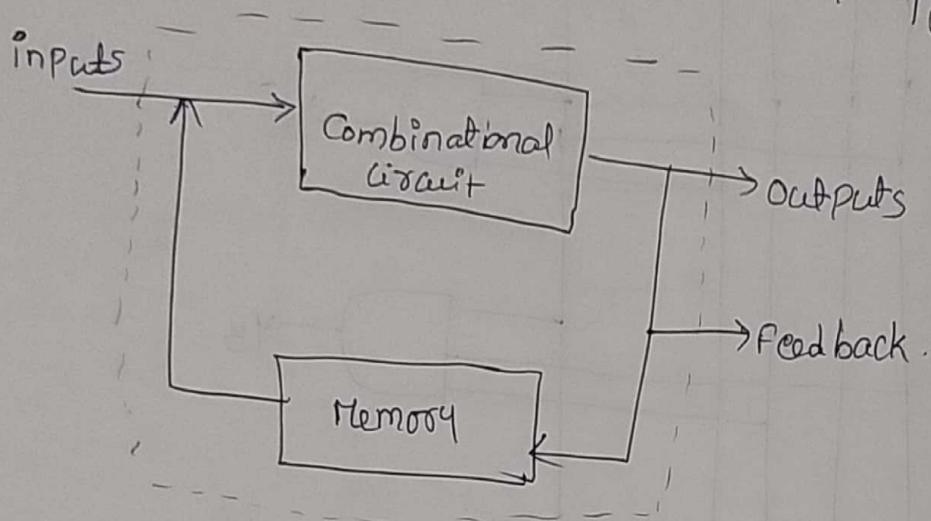


UNIT - III

Sequential Circuits

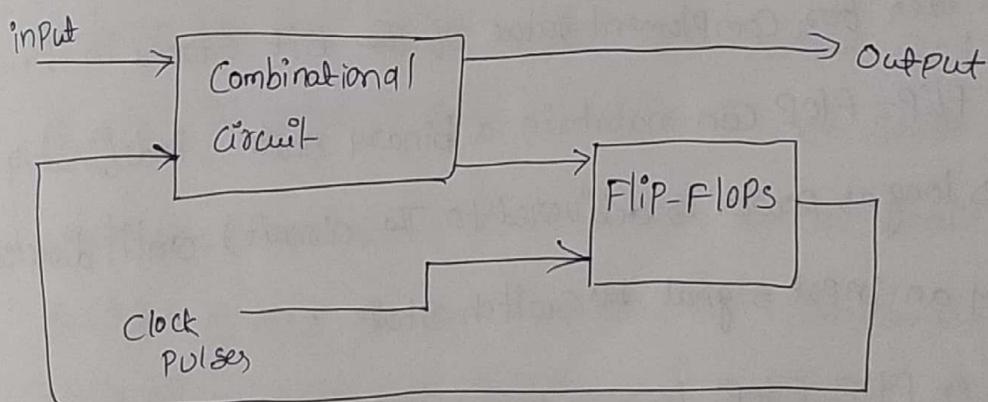
- Sequential Circuits is a Combinational Circuits with memory.
- The output of sequential circuits depends upon Present inputs & Present state (Past outputs / Inputs)



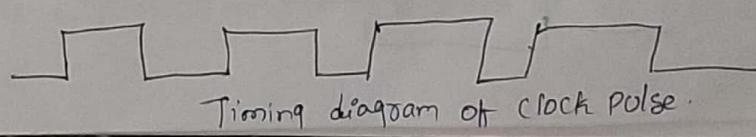
- The information stored in sequential circuits represents Present state.
- The Present state and Present inputs will define outputs & next state.
- In The Sequential Circuits we are using synchronous and Asynchronous circuits. Synchronous Runs with same Clock Pulse. Asynchronous Runs with different Clock Pulse. Sequential elements not use same clock in Asynchronous circuits.

- For synchronous sequential circuits example is **FLIP-FLOPS**
- For asynchronous circuits example is **Latches & Timedelay**

A synchronous sequential circuit employs signals that affect the storage elements at only discrete instants of time. Synchronization is achieved by a timing device called a clock generator, which provides a clock signal having the form of a periodic train of clock pulses. The clock signal is commonly denoted by the identifiers **Clock** and **Clk**. The clock signal is commonly 1. Pulses are distributed throughout the system in such a way that storage elements are affected only with the arrival of each pulse.



Synchronous Clocked Sequential Circuits



Timing diagram of clock pulse.

The storage elements used in clocked sequential circuits are called FLIPFLOPS. A flipflop is a binary storage device capable of storing one bit of information. In a

stable state the output of flipflop is either 0 or 1.

A sequential circuit may use many flip-flops to store as many bits as necessary. In the block diagram representation

of a synchronous ^{clocked} sequential circuit the outputs are formed by a combinational logic function of the inputs to the circuit or the values stored in the flip-flops.

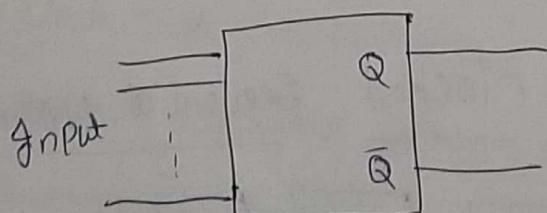
FLIP-FLOPS :-

→ Flip Flop is a memory element which is capable of storing one bit of information and it is used in clocked sequential circuits.

→ A flip flop has two outputs, one for normal value and other for complement value of the bit stored in it.

→ A flip-flop can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch state.

→ A flip-flop is also known as bistable multivibrator.



→ FLIP-FLOPs are of different types depending on how their inputs and clock pulses cause transition between two states.

→ Four types of FLIPFLOPs: SR, JK, D and T FLIP-FLOPs

Latches :- Latches are unclocked flip-flops.

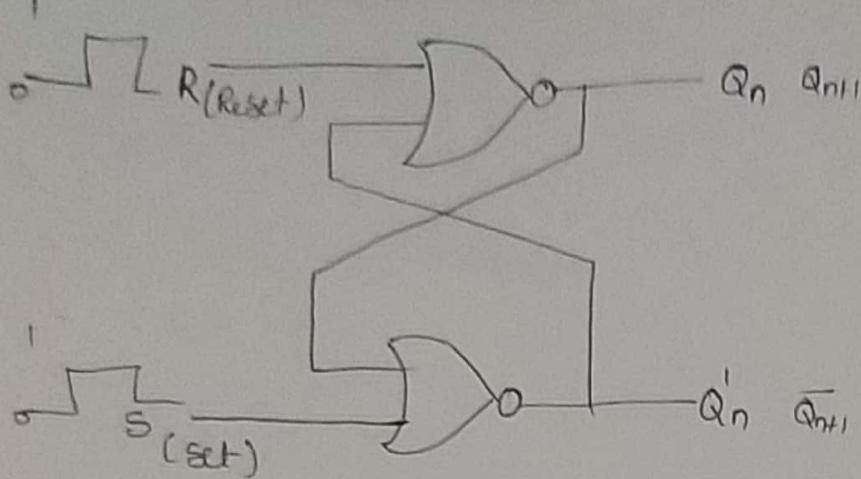
Latches are useful for storing binary information and for the design of asynchronous sequential circuits because they are the building blocks of flip-flops.

SR Latch :-

The SR Latch is a circuit with two cross-coupled NOR Gates or two cross-coupled NAND Gates and two inputs labeled S for Set and R for Reset. The SR latch constructed with two cross coupled NOR gates. The latch has two useful states when output $Q=1$ and $Q'=0$, the latch is said to be in the set state. When $Q=0$ and $Q'=1$ it is in the reset state.

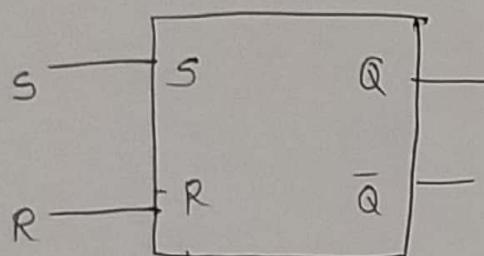
Outputs Q and Q' are normally the complement of each other. When both inputs are equal to 1 at the same time, a condition in which both outputs are equal to 0 if both inputs are switched to 0. The device will enter the unpredictable or undefined state.

SR Latch using NOR gates:-



SR Latch with NOR Gates

Logic symbol:-



Truth table

S	R	Q _n	Q _{n+1}
0	0	0	0
0	0	1	1
<hr/>			No Change
0	1	0	0
0	1	1	0
<hr/>			Reset
1	0	0	1
1	0	1	1
<hr/>			Set
1	1	0	?
1	1	1	?
<hr/>			Undefined State
<hr/>			Indeterminate State

(i) S=0 R=0

$$Q_{n+1} = \overline{0 + \bar{Q}_n} = Q_n$$

(ii) S=0 R=1

$$Q_{n+1} = \overline{1 + \bar{Q}_n} = 0$$

(iii) S=1 R=0

$$Q_{n+1} = \overline{0 + \bar{Q}_n} = Q_n$$

$$\overline{Q_{n+1}} = \overline{1 + \bar{Q}_n} = 0$$

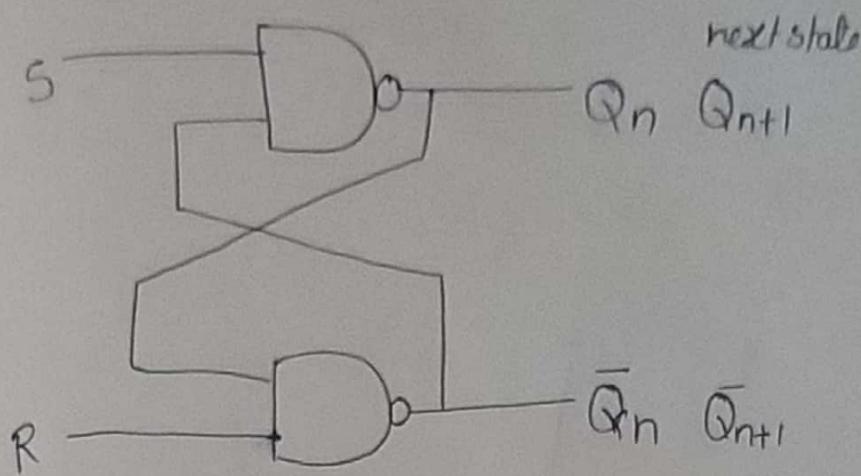
$$\overline{Q_{n+1}} = 0 \text{ Then } Q_{n+1} = 1$$

(iv) S=1 R=1

$$Q_{n+1} = \overline{1 + \bar{Q}_n} = 0$$

$$\overline{Q_{n+1}} = \overline{1 + \bar{Q}_n} = 0$$

SR Latch with NAND Gates:



S	R	Q _n	Q _{n+1}
0	0	0	I.D
0	0	1	
<hr/>			
0	1	0	1 set
<hr/>			
0	1	1	1
<hr/>			
1	0	0	0
<hr/>			
1	0	1	0 Reset
<hr/>			
1	1	0	0 No change
<hr/>			
1	1	1	1

(i) S=0 R=0

$$Q_{n+1} = \overline{0 \cdot Q_n} = 1$$

$$\bar{Q}_{n+1} = \overline{0 \cdot Q_n} = 1$$

(ii) S=0 R=1

$$Q_{n+1} = \overline{0 \cdot \bar{Q}_n} = 1$$

$$\bar{Q}_{n+1} = \overline{1 \cdot \bar{Q}_n} = \bar{Q}_n$$

(iii) S=1 R=0

$$Q_{n+1} = \overline{1 \cdot \bar{Q}_n} = Q_n = 0$$

$$\bar{Q}_{n+1} = \overline{0 \cdot \bar{Q}_n} = 1$$

(iv) S=1 R=1

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

$$\bar{Q}_{n+1} = \overline{1 \cdot Q_n} = \bar{Q}_n$$

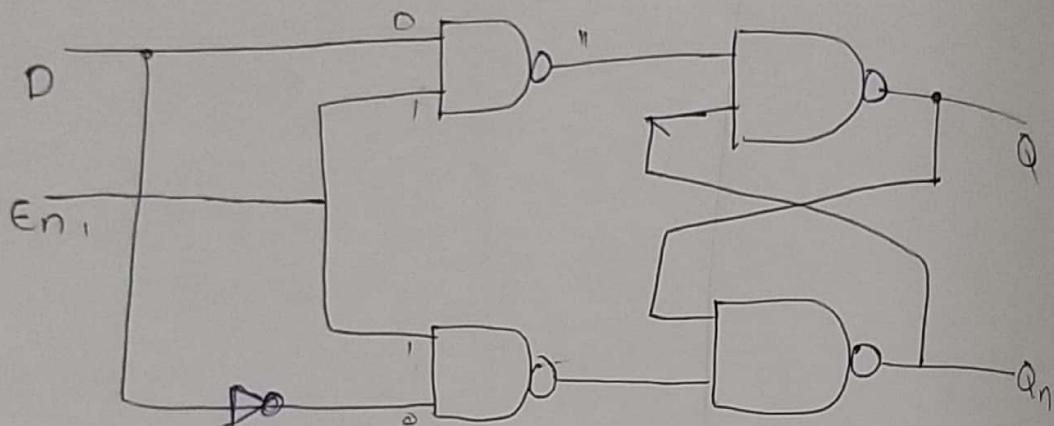
D-latch

- It can eliminate the undesirable condition of its indetermined states in the SR latch

D-latch has only 2 inputs

D - Data En - Enable

- The D input directly goes to S input and its complement is applied to the R input.



The enable input is at 0, the cross coupled SR latch has both inputs at the 1 level and the circuit cannot change state regardless of value of D. The D input is sampled when $En=1$. If $D=1$ the Q output goes to 1, placing the circuit in the set state. If $D=0$ output Q goes to 0 placing the circuit in the reset state.

The D latch receives that designation from its ability to hold data in its internal storage. It is suited to use a temporary storage for binary information between a unit and its environment.

<u>En</u>	<u>D</u>	<u>Q next state</u>
0	x	no change
1	0	$Q = 0$ no set state
1	1	$Q = 1$ set state

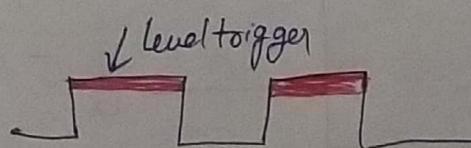
Triggers :-

The output of a Flip-flop can be changed by bringing a small change in the input signal. This small change can be brought with the help of a clock pulse or commonly known as a trigger pulse. When such a trigger pulse is applied to the input, the output changes and thus the flip-flop is said to be triggered.

Triggers are two types:

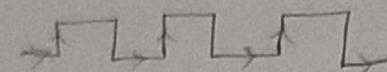
- level triggering
- edge triggering

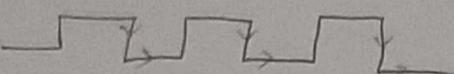
Level triggering :- Level triggered describing a circuit of component whose output is sensitive to changes of the inputs only so long as the clock input signal is high



Edge trigger

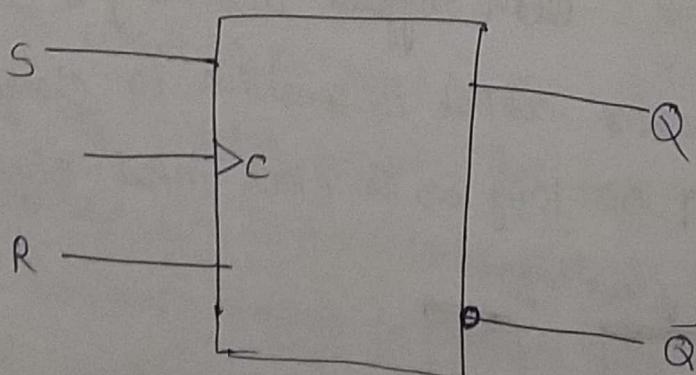
A edge triggered flip-flop changes states either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse on the control input

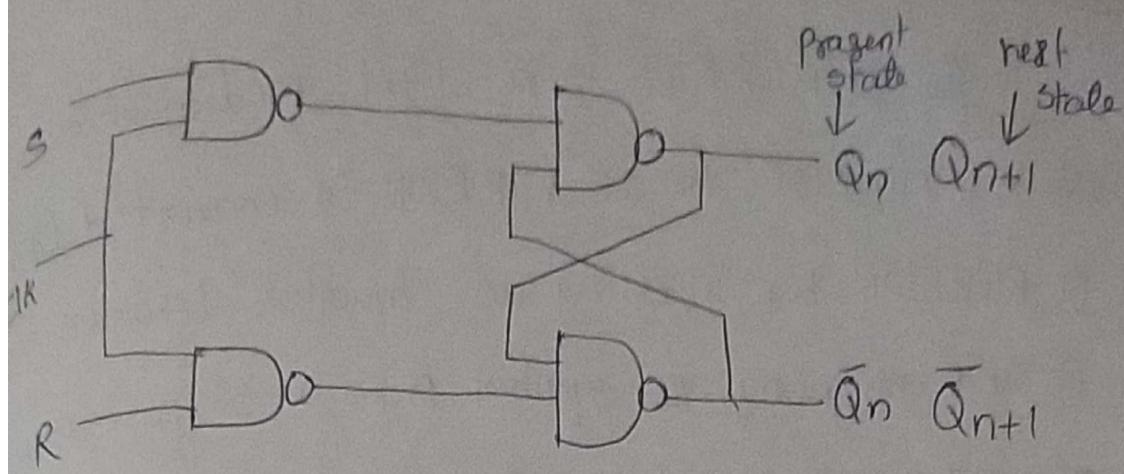
+ve edge : 

-ve edge : 

S-R FLIPFLOP

In the SR FLIPFLOP we are having three inputs labeled S(set), R(Reset) and C(clock). It has an output Q and sometimes the flipflop has a complementary output. In the SR FLIPFLOP we are having Q and \bar{Q} are the present states. By using present inputs & present state and clock pulse we are finding the next state that is Q_{n+1} and the complement of next state \bar{Q}_{n+1} .





CIK	S	R	PS Qn	NS Qn+1
↑	0	0	x	Qn - no change
↑	0	1	x	0 - Reset
↑	1	0	x	1 - Set
↑	1	1	x	ID - Indeterminate State

$$(i) S=0 \quad R=0$$

$$Q_{n+1} = \overline{0 \cdot \bar{Q}_n} = \bar{0} = \overline{1 \cdot \bar{Q}_n} = Q_n$$

$$\bar{Q}_{n+1} = \overline{\bar{0} \cdot Q_n} = \bar{0} = \overline{\bar{1} \cdot Q_n} = \bar{Q}_n$$

$$(iv) S=1 \quad R=1$$

$$Q_{n+1} = \overline{\bar{1} \cdot \bar{Q}_n} = Q_n = \overline{Q_n \cdot \bar{Q}_n} \\ = \bar{0} = 1$$

$$\bar{Q}_{n+1} = \overline{\bar{1} \cdot Q_n} = \bar{Q}_n = \overline{Q_n \cdot Q_n} \\ = \bar{0} = 1$$

$$(ii) S=0 \quad R=1$$

$$Q_{n+1} = \overline{\bar{0} \cdot \bar{Q}_n} = \bar{0} = \overline{\bar{1} \cdot \bar{Q}_n} = Q_n = 0$$

$$\bar{Q}_{n+1} = \overline{\bar{1} \cdot Q_n} = \bar{Q}_n = \overline{Q_n \cdot Q_n} = \bar{0} = 1$$

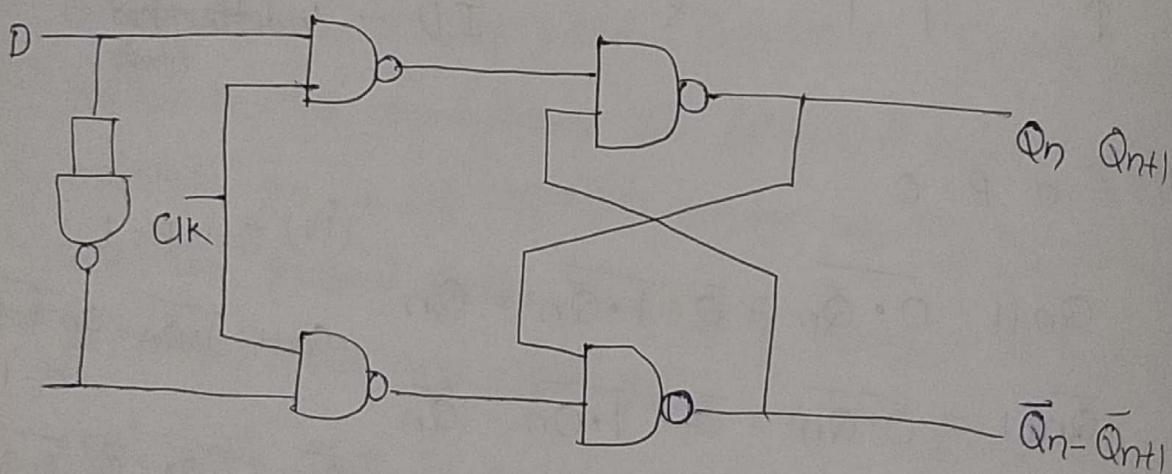
$$(iii) S=1 \quad R=0$$

$$Q_{n+1} = \overline{\bar{1} \cdot \bar{Q}_n} = Q_n = \overline{Q_n \cdot \bar{Q}_n} = \bar{0} = 1$$

$$\bar{Q}_{n+1} = 0$$

D-FLIP FLOP :-

The D-FlipFlop is a slight modification of SR flipFlop. An SR flipFlop is converted to a D-FlipFlop by inserting an inverter between S and R and assigning the symbol D to a single input. The D input is sampled during the occurrence of a clock transition from 0 to 1. If $D=1$, the output of the flip-flop goes to the 1 state, but if $D=0$, the output of the flip-flop goes to the 0 state. D-Flip-Flop is called Delay Flip-Flop / Data Flip-Flop.



CLK	D	P.S Qn	N.S Qn+1
1	0	X	0 Reset
1	1	X	1 Set

(i) $D = 0$

$$Q_{n+1} = \overline{Q \cdot \bar{Q}_n} = \overline{Q_n} = \overline{1 \cdot \bar{Q}_n} = Q_n = 0$$

$$\bar{Q}_{n+1} = \overline{D \cdot Q_n} = \bar{Q}_n = \overline{\bar{Q}_n \cdot Q_n} = \bar{Q}_n = 1$$

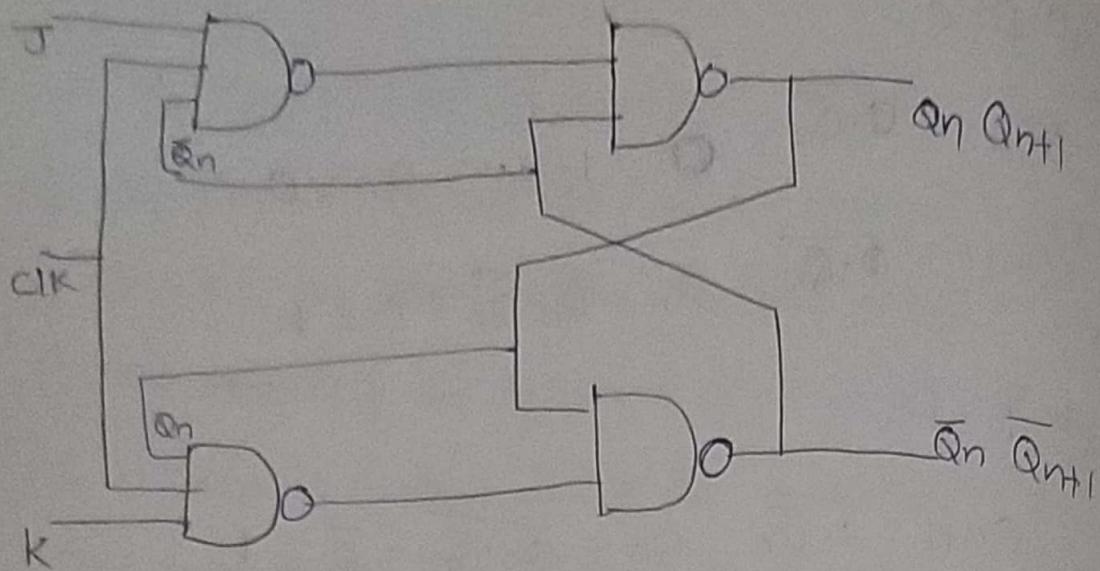
(ii) $D = 1$

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

$$\bar{Q}_{n+1} = 0$$

J-K Flip-Flop :-

In the SR-Flip Flop we are having one state as wastage state that is indetermined state. By using J-K Flip-Flop we are changing indetermined state into toggle state. A JK Flip-Flop is a refinement of the SR flip-flop in that the indetermined condition of the SR type is defined in the JK type, inputs J and K behave like inputs S and R to set and clear the flip-flop respectively. When inputs J and K are both equal to 1, a clock transition switches the outputs of the flip-flop to their complement state.



J	K	Q_n	Q_{n+1}
0	0	x	Q_n - no change
0	1	x	0 - Reset
1	0	x	1 - Set
1	1	x	\bar{Q}_n - toggle

(i) $J=0 \quad K=0$

$$Q_{n+1} = \overline{0 \cdot 1 \cdot \bar{Q}_n} = \bar{0} = \overline{1 \cdot \bar{Q}_n} = Q_n$$

$$\bar{Q}_{n+1} = \overline{0 \cdot 1 \cdot Q_n} = \bar{0} = \overline{1 \cdot Q_n} = \bar{Q}_n$$

(ii) $J=0 \quad K=1$

$$Q_{n+1} = \overline{0 \cdot 1 \cdot \bar{Q}_n} = \bar{0} = \overline{1 \cdot \bar{Q}_n} = Q_n = 0$$

$$\bar{Q}_{n+1} = \overline{1 \cdot 1 \cdot Q_n} = \bar{Q}_n = \overline{\bar{Q}_n \cdot Q_n} = \bar{0} = 1$$

(iii) $J=1 \quad K=0$

$$Q_{n+1} = \overline{1 \cdot 1 \cdot \bar{Q}_n} = Q_n = \overline{Q_n \cdot \bar{Q}_n} = \bar{0} = 1$$

$$\bar{Q}_{n+1} = 0$$

(iv) $J=1 \ K=1$

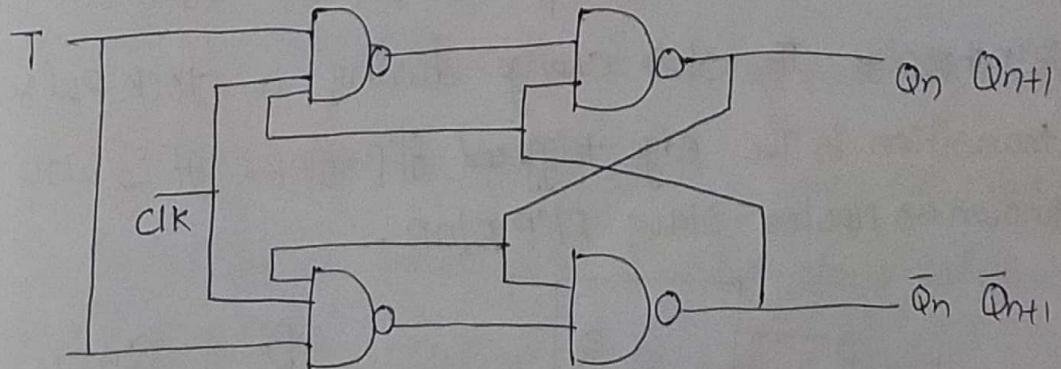
$$Q_{n+1} = \overline{J \cdot I \cdot \bar{Q}_n} = \underline{Q_n} = \overline{Q_n \cdot \bar{Q}_n} = \bar{O} = 1$$

$$Q_{n+1} = \overline{I \cdot I \cdot Q_n} = \underline{\bar{Q}_n} = \overline{Q_n \cdot \bar{Q}_n} = \bar{O} = 1$$

$$Q_{n+1} = \overline{I \cdot Q_n} = \underline{\bar{Q}_n}$$

T-FlipFlop :-

T (toggle) flip-flop is obtained from a JK type when inputs J and K are connected to provide a single input designated by T. The T flip-flop therefore has only two conditions. When $T=0$ ($J=0 \text{ and } K=0$) a clock transition does not change the state of the flip-flop. When $T=1$ ($J=K=1$) a clock transition complements the state of the flip-flop.



T	Q_n	Q_{n+1}
0	x	Q_n no change state
1	x	\bar{Q}_n toggle state

(i) $T=0$

$$Q_{n+1} = \overline{0 \cdot 1 \cdot \bar{Q}_n} = \bar{0} = \overline{1 \cdot \bar{Q}_n} = Q_n - \text{no change}$$

$$\bar{Q}_{n+1} = \overline{0 \cdot 1 \cdot Q_n} = \bar{0} = \overline{1 \cdot Q_n} = \bar{Q}_n - \text{toggle}$$

(ii) $T=1$

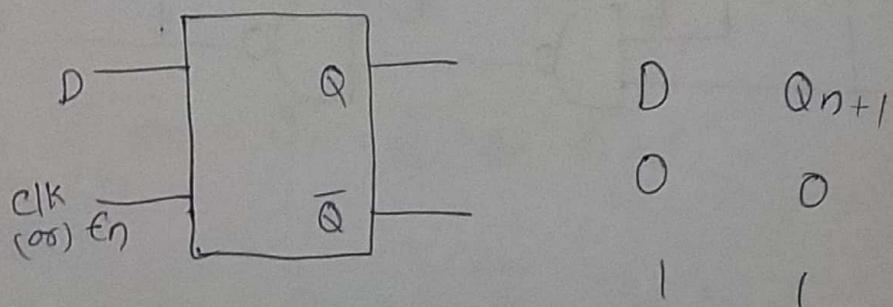
$$Q_{n+1} = \overline{1 \cdot 1 \cdot \bar{Q}_n} = Q_n = \overline{Q_n \cdot \bar{Q}_n} = \bar{Q} = 1$$

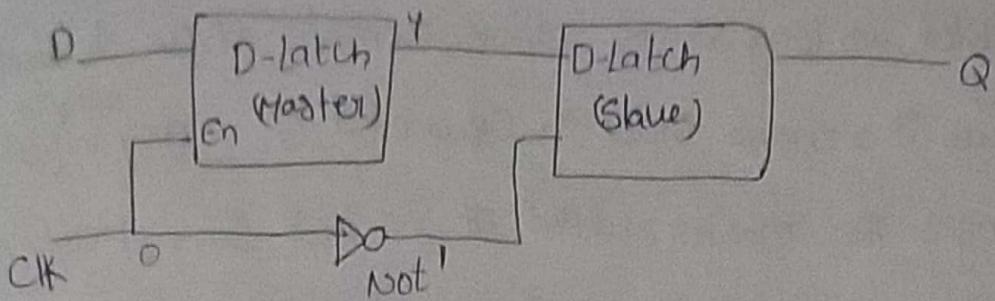
$$\bar{Q}_{n+1} = \overline{1 \cdot 1 \cdot Q_n} = \bar{Q}_n = \overline{\bar{Q}_n \cdot Q_n} = \bar{Q} = 1$$

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

Master-slave edge triggered D-Flip Flop :-

The most common type of flip-flop used to synchronize the state change during a clock pulse transition is the edge-triggered flip-flop. It is also known as Master-slave Flip Flop.





when $CLK = 0$ The slave D-Latch is enabled so

$$\text{slave} = 1 \text{ (enable)} \quad Q = Y$$

when $CLK = 1 = \text{master enable } Y = D$

when $CLK = 0 = \text{from 1 to 0} \quad Q = Y = D$

slave is dependent on Master D-latch.

The Construction of a D-FlipFlop with two D latches and an inverter is called edge triggered FlipFlop.

The first Latch is called the master and the second Latch is the slave. The circuit samples the D input and changes its output Q only at the negative edge of the synchronizing and controlling clock. When the clock is 0 the output of inverter is 1. The slave latch is enabled and its output Q is equal to the master output Y. The master latch disabled because $CLK=0$.

When the input pulse changes to the logic-1 level, the data from the external D input are

Transferred to the Master. The Slave however disabled as long as the clock remain at the 1 level, because its enable input is equal to 0. Any change in input changes the master output at Y_1 but cannot affect the slave output. When the clock pulse return to 0, the master is disabled and isolated from the D input. At the same time the slave is enable and the value of Y_1 is transferred to the output of the flip-flop at Q. A change in the output of the flip-flop can be triggered only by and during the transition of the clock from 1 to 0.

The behavior of the master-slave flip-flop is that the output may change only once, a change in the output is triggered by the +ve edge of the clock and the change may occur only during the clock's +ve level.

Differences between Combinational Circuits and Sequential Circuits:

Combinational Circuits

→ In this circuits output only depends upon present input

→ Speed is Fast

→ It is designed easy

→ There is no feedback between i/p & o/p

→ This is time independent

→ Elementary building blocks: logic gates

→ Used for mathematical as well as boolean operations

→ Combinational circuit don't have capability to store data

→ These circuits don't have clock. They don't require triggering

→ It is easy to handle

Eg: encoder, Decoder, Multiplexer, Demultiplexer

Sequential Circuits

→ In this output depends on present input & present state (Past i/p)

→ Speed is slow

→ It is designed tough as compared to Combinational Circuit.

→ There exists a feedback path between i/p & o/p.

→ This is time dependent

→ Elementary building blocks: FlipFlops, universal gates

→ Mainly used for storing data

→ Sequential Circuits have capability to store any state or retain earlier state

→ These Circuits are clock dependent. They need triggering

→ It is not easy to use and handle

Eg: FlipFlops, latches, counters

Differences between Latches & Flip Flops :-

Latch

FLIP FLOP

- Latch does not Required → FlipFlop requires Clock signal
- Clock signal
- It is level sensitive → It is edge sensitive device
- less Power required → more Power required
- Latch is Asynchronous device → synchronous device
- The operation of a Latch is Faster as they donot have a wait for any clock signal → Flip-FLOPS are comparatively slower than latch due to clock signal.

Analysis of Clocked Sequential Circuits

In this from Logical diagram write down the equations for the outputs and the Flip-Flop inputs.

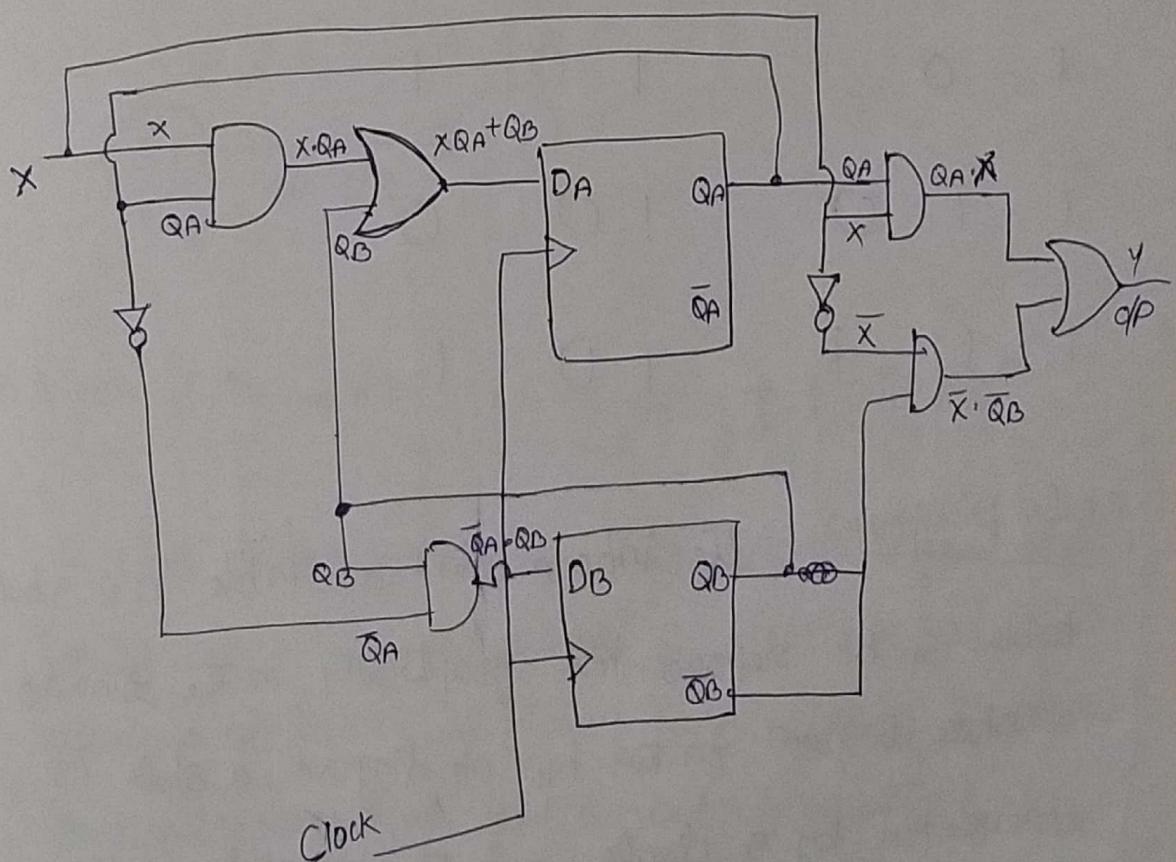
Using these equation derive a state table which describes the next state. To obtain state diagram from

The state table - It is the state table and/or state diagram that specifies behaviour of the circuit.

The Flip-Flop input equations are sometimes

Called the excitation equations. The state table is

Sometimes called a transition table.



State table :-

$$Q_A^+ = D_A = \alpha Q_A + Q_B$$

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

$$Q_B^+ = D_B = \bar{Q}_A \cdot Q_B$$

$$Q_A^+ = D_A$$

$$Y = \bar{X}Q_B + X \cdot Q_A$$

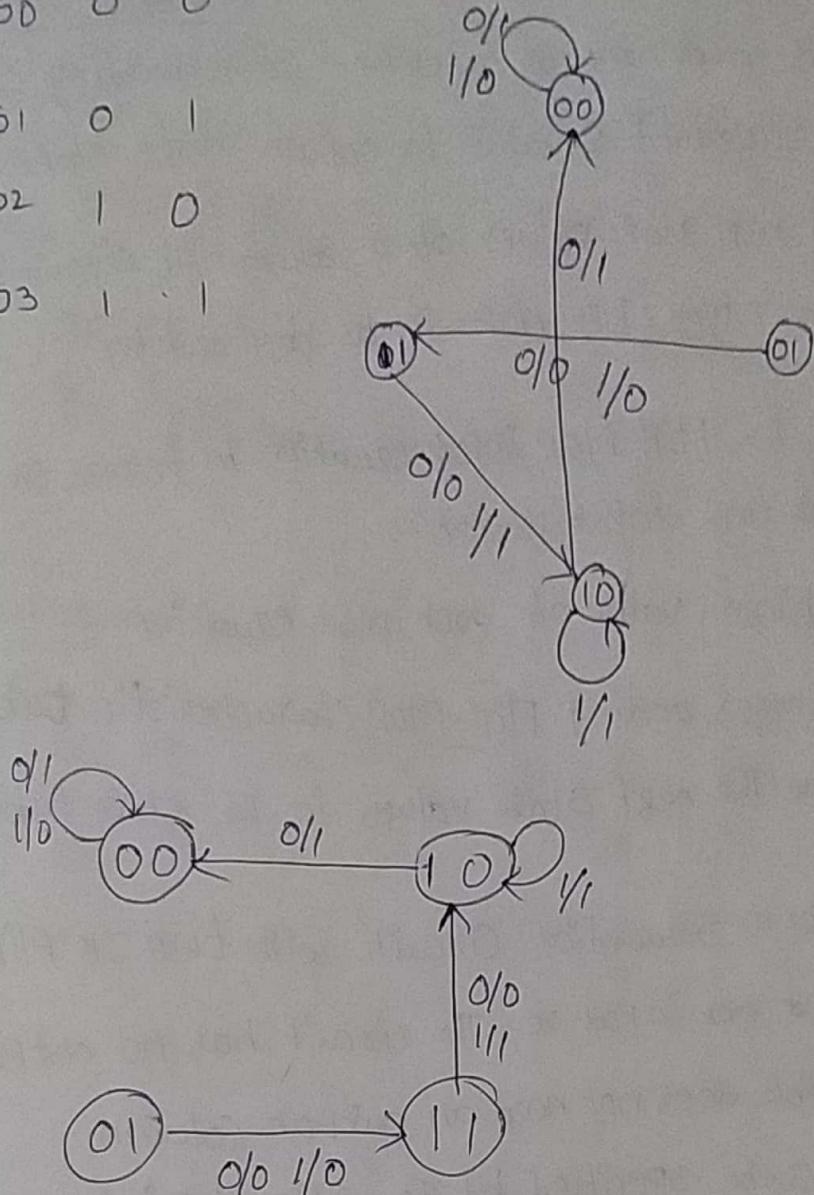
$$Q_B^+ = D_B$$

Q_A	Q_B	X	Q_A^+	Q_B^+	Y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	0	1

State Diagram :-

The information available in a state table can be represented graphically in the form of a state diagram. In this type of diagram, a state is represented by a circle and the transition between states are indicated by directed lines connecting the circles.

	QA	QB
S0	0	0
S1	0	1
S2	1	0
S3	1	1



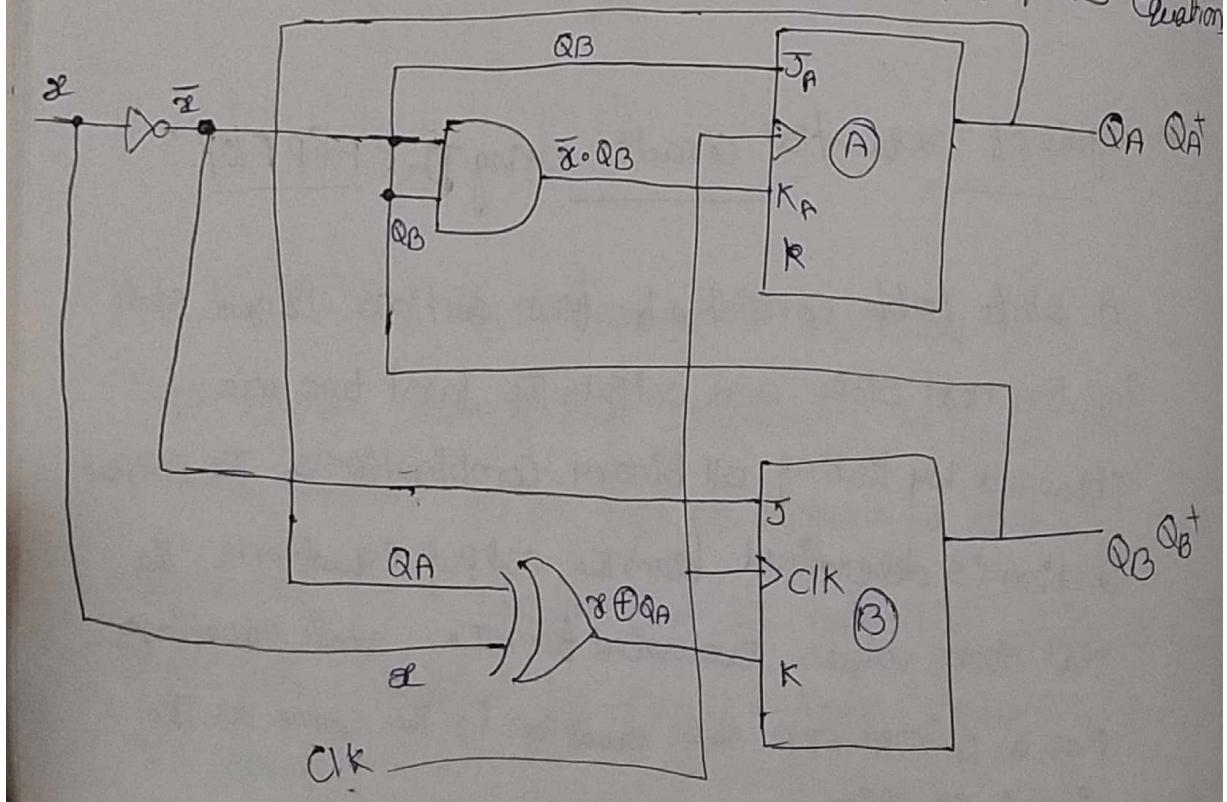
Analysis of sequential circuits using J-K Flip Flop :

A state table consist of four sections: Present state, inputs, next state, and outputs. The first two are obtained by listing all binary combinations. The output section is determined from the output equations. The next state values evaluated from the state equations. For a D-Flip-Flop' state equation is the same as the input equation.

When a flip-flop other than the D-type is used, such as a JK or T, it is necessary to refer the corresponding characteristic table or characteristic equation to obtain next state values. The next state values of a sequential circuit that uses JK- or T-type flip-flops can be derived by

1. Determine the flip-flop input equation in terms of the present state and input variables.
2. List the binary value of each input equation
3. Use the corresponding flip-flop characteristic table to determine the next state values in the state table

Consider a sequential circuit with two JK flip-flops A and B and one input α . The circuit has no outputs. The state table does not need an output column. The circuit can be specified by the flip-flop inputs equation



Input Equations :-

$$J_A = Q_B$$

$$K_A = \bar{x} \cdot Q_B = \bar{x} \cdot J_A$$

$$J_B = \bar{x}$$

$$K_B = x \oplus Q_A = \bar{x}Q_A + \bar{Q}_A x$$

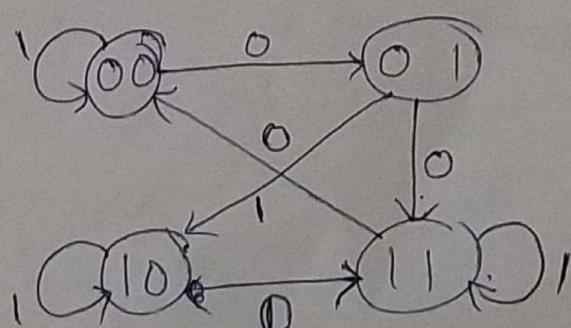
T Table for JK-Flop

J	K	Q_A	\bar{Q}_A
0	0	0	Q_n
0	1	-x	0
1	0	x	1
1	1	x	\bar{Q}_n

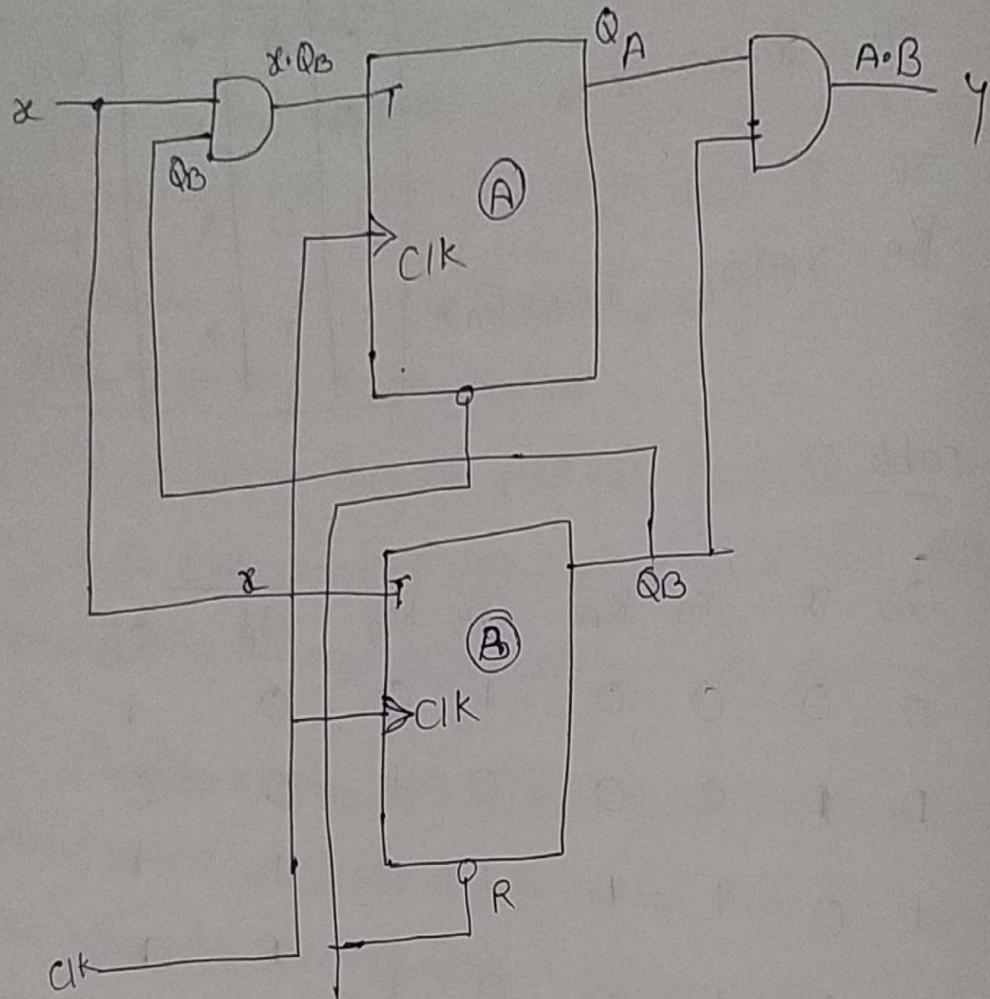
State Table :-

<u>P.S</u>						<u>N.S</u>		
Q_A	Q_B	x	J_A	K_A	J_B	K_B	Q_A^+	Q_B^+
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	0
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1

	Q_A	Q_B
S_0	00	0
S_1	0	1
S_2	1	0
S_3	1	1



Sequential Circuit with T-Flip Flop



Equations:-

$$T_A = \bar{x} \cdot Q_B$$

$$T_A \quad T_B \quad \bar{x} \quad T_A \bar{y}$$

$$T_B = \bar{x}$$

$$y = Q_A \bar{Q}_B$$

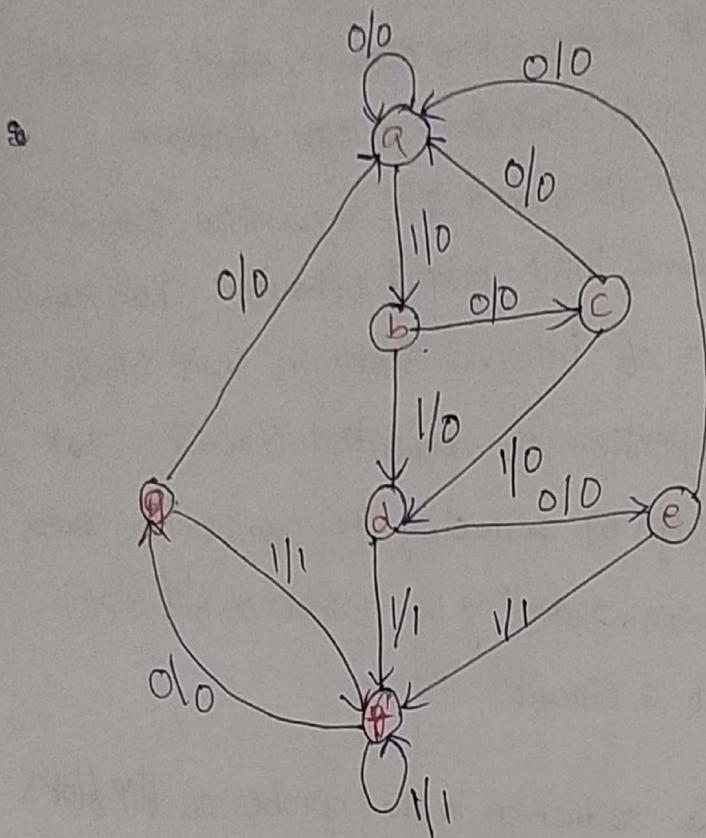
State Reduction & Assignment :

The analysis of sequential circuits starts from a circuit diagram and culminates in a state table or diagram. The design of a sequential circuit starts from a set of specifications and culminates in a logic diagram. Design procedures are presented. Two sequential circuits may exhibit the same input-output behaviour, but have a different number of internal states in their state diagram. Certain properties of sequential circuits that may simplify a design by reducing the number of gates and flip flops it uses, reducing the number of flip flops reduces the cost of a circuit.

State Reduction :- The reduction in the number of flip-flops in a sequential circuit is referred to as the state-reduction problem. State reduction algorithms are concerned with procedures for reducing the number of states in a state table, while keeping the external input/O/P requirements unchanged. Since m flip-flops produce 2^m states, a reduction in the number of states may result in a reduction in the number of flip-flops.

A state reduction procedure with an example. We start with a sequential circuit whose specification is given in the state diagram.

only the input-output sequences are important. The internal states are used merely to provide the required sequences.



State diagram

State:	a	a	b	c	d	e	f	f	g	f	g	q
input:	0	1	0	1	0	1	1	0	1	0	0	
Output:	0	0	0	0	0	1	1	0	1	0	0	

If identical input sequences are applied to the two circuits and identical outputs occurs for all input sequences then the two circuits are said to be equivalent and one may be replaced by the other.

The following algorithm for the state reduction of a completely specified state table is given here with out proof.

"Two states are equivalent if, for each member of the set of inputs, they give exactly the same output and send the circuits either to the same state or to an equivalent state" when two states are equivalent, one of them can be removed without altering the input-output relationship.

P.S	N.S		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a b		0	0
b	c d		0	0
e	a d		0	0
d	e f		0	1
e	a f		0	1
f	g f		0	1
g	a f		0	1

We look for two present states that go to the same next state and have the same output for both input combinations. States e and g are two such states. They both go to state a and f and have outputs 0 for $x=0$ and $x=1$. Therefore states g and e are equivalent and one of the states removed.

The Procedure of removing a state and replacing it by its equivalent. The row with Present State g is removed and state g is replaced by state 'e' each time it occurs in the columns headed "Next state".

Reducing the state table:

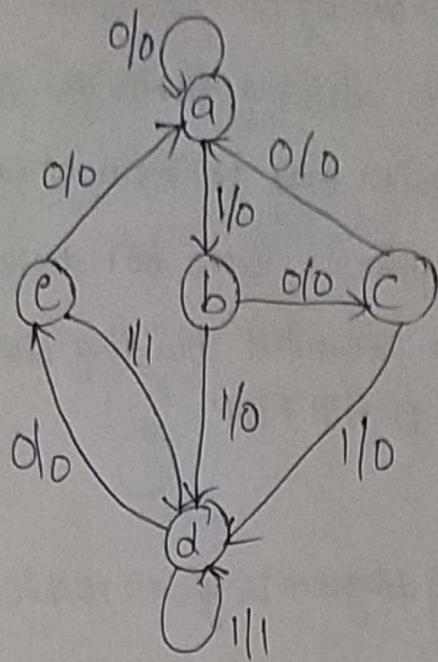
P.S	Next State		Output	
	$X=0$	$X=1$	$X=0$	$X=1$
a	b		0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Present State f has next states e and f and outputs 0 and for $x=0$ and $x=1$ respectively. The same next states and outputs appears in the row with Present State d. Therefore states f and d are equivalent, and state f can be removed and replaced by d.

Reduced state Table:-

P.S	N.S		O/P	
	$X=0$	$X=1$	$X=0$	$X=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Reduced State Diagram



The sequential circuit was reduced from seven to five states.

State Assignment

In order to design a sequential circuit with physical components, it is necessary to assign unique coded binary values to the states. For a circuit with m states, the codes must contain n bits where $2^n \geq m$. For example, with three bits, it is possible to assign codes to eight states, denoted by binary numbers 000 through 111. If the state table is used, we must assign binary values to seven states; the remaining state is unused.

State assignment is used binary number representing states

Design Procedure

Design Procedures or methodologies specifies hardware that will implement a desired behavior. The design effort for small circuits may be manual, but industry relies on automated synthesis tools for designing massive integrated circuits. The sequential building block used by synthesis tools is the D-Flip-Flop.

Step 1: A state diagram or timing diagram is given, which describes the behaviour of the circuit.

Step 2: Obtain the state table.

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

Step 4: Do state assignment (if required)

Step 5: Determine the number of FlipFlops required and assign letter symbols.

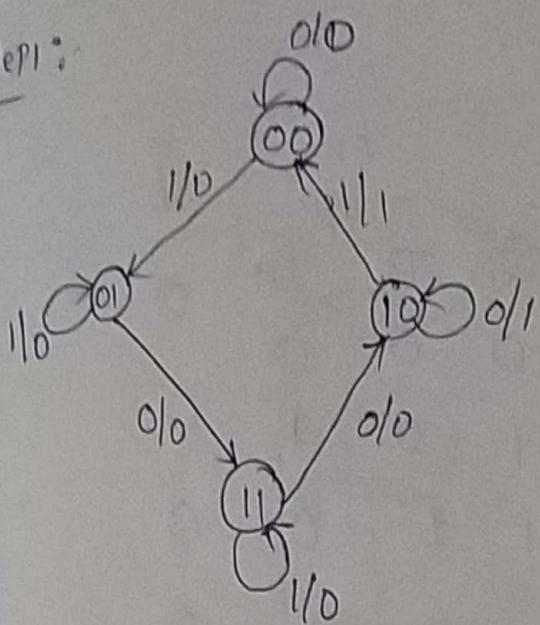
Step 6: Decide the type of Flip-Flop to be used

Step 7: Derive the circuit excitation table for state table

Step 8: Obtain the expression for circuit o/p and flip-flop input

Step 9: Implement the circuit

Step1:-



→ Step2:-

Q_A	Q_B	$N.S$ $x=0$	$x=1$	Output(y)
Q_A^+	Q_B^+	Q_A^+	Q_B^+	$x=0$ $x=1$
0	0	0	0	0 0
0	1	1	1	0 0
1	0	1	0	1 1
1	1	1	1	0 0

→ Step3:- No State Reduction Possible

→ Step4:- Don't have to assign

→ Step5:- ~~as~~ we are using 2 flip flop

$$(A) \rightarrow Q_A$$

$$(B) \rightarrow Q_B$$

→ Step6:- using T flip flop.

Circuit excitation table:

PS		N-S		FF I/P	Y		
QA	QB	x	Q_A^+	Q_B^+	T_A	T_B	0
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	0	0	0	1

→ Step 8:-

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

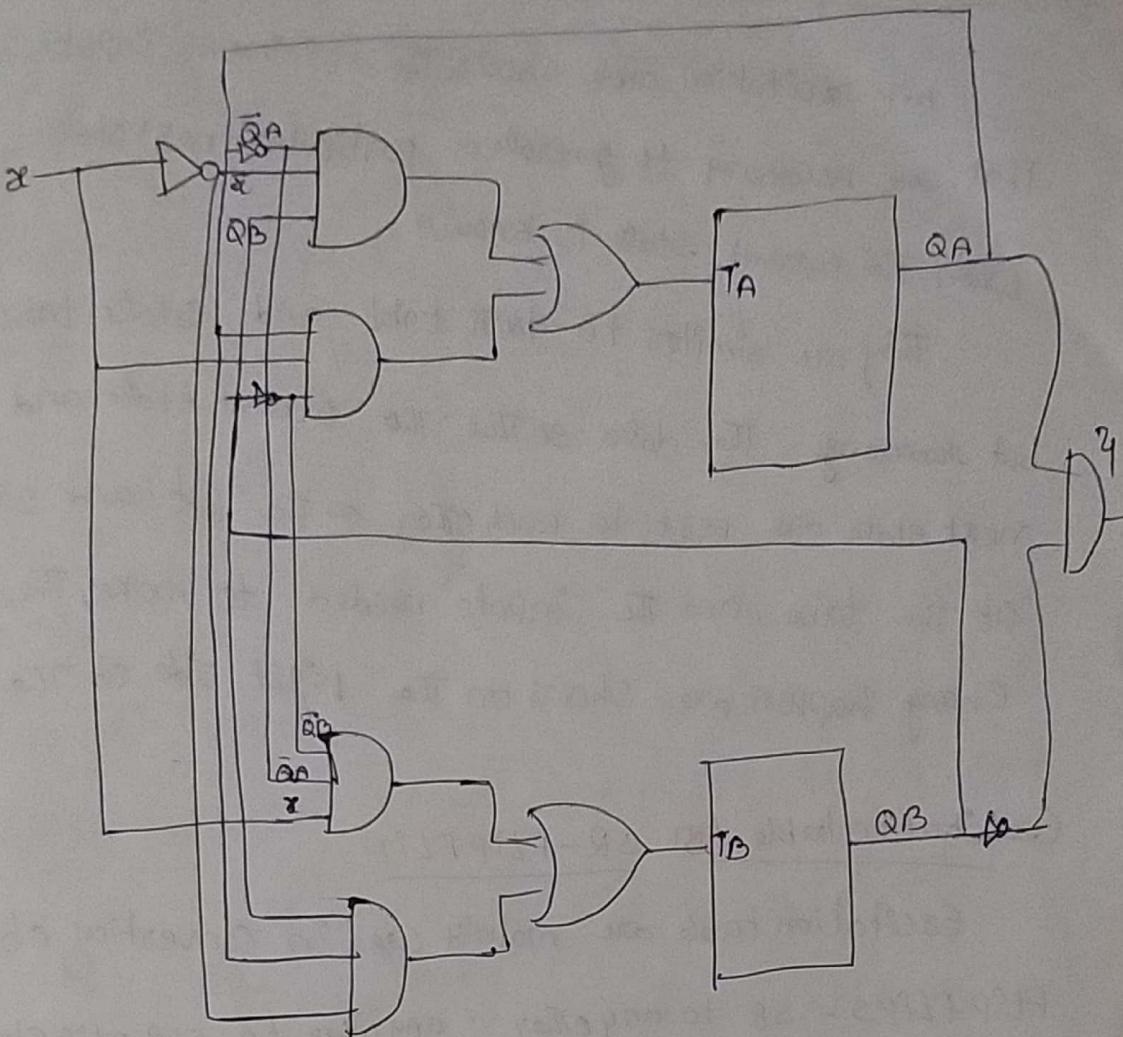
$$T_B = \bar{Q}_A \bar{Q}_B x + Q_A Q_B \bar{x}$$

$$Y = Q_A \bar{Q}_B \bar{x} + Q_A \bar{Q}_B x$$

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

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

$$Y = Q_A \bar{Q}_B$$



Flip-Flop Synchronous Excitation table :-

An excitation table shows the minimum inputs that are necessary to generate a particular next state when the current state is known.

They are similar to TOTL table and State table but rearrange the data so that the current state and next state are next to each other on the left hand side of the table, and the inputs needed to make the state change happen are shown on the right side of the table.

Excitation table for SR-FLIP FLOP :-

Excitation table are mainly used in conversion of FLIP-FLOPs. SR -to any other . any other to S-R FLIP FLOP.

TOTL table of SRFLIPFLOP :-

S	R	P_S Q_n	Q_{n+1}	S	R
0	0	x 1	0 1	0	0
0	1	x 1	0 0	0	1
1	0	x 1	1 1	1	0
1	1	x 1	ID	-	-

Excitation table for S-R FlipFlop

P	S	N.S	Mandatory Input S	X ← don't care R
Qn		Qn+1		
0	0	0	0	X
0	1	1	1	0
1	0	0	0	1
1	1		X	0

Characteristic table :-

S	R	Qn	Qn+1
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	X
1	1	1	X

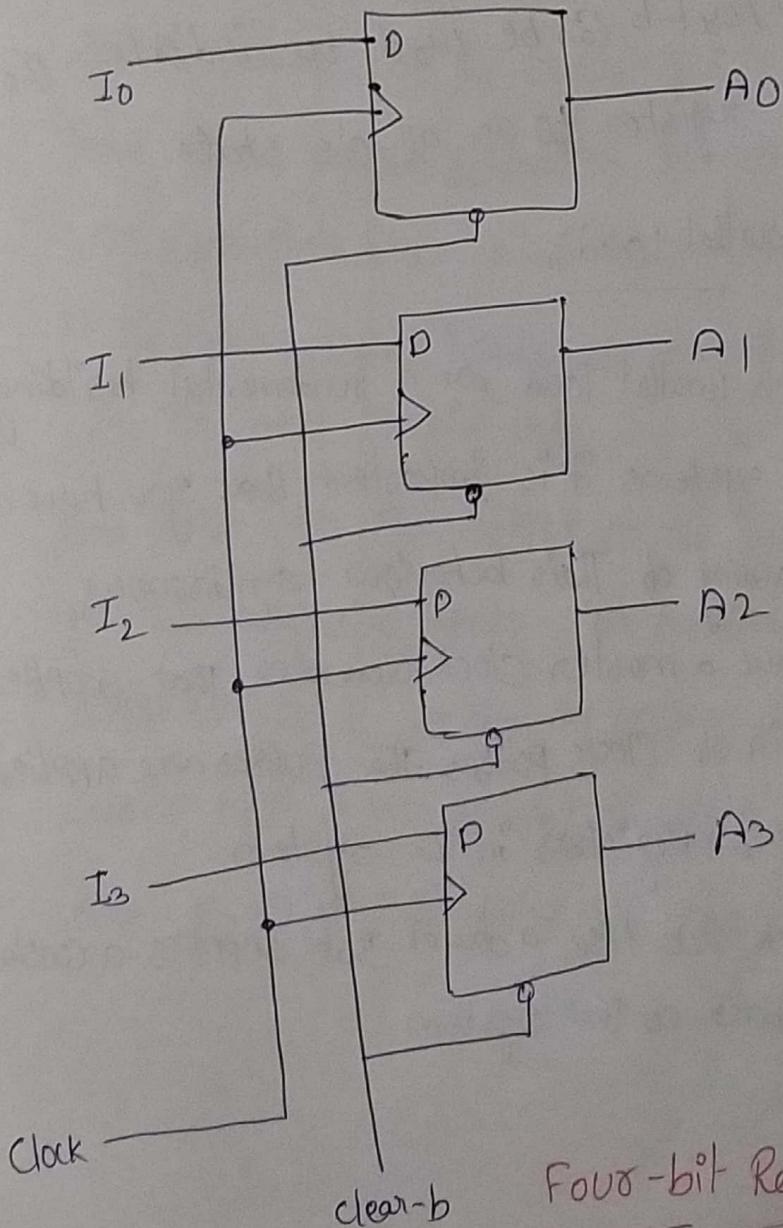
Registers

A clocked sequential circuit consists of a group of flip-flops and combinational gates. The flip-flops are essential because, in their absence, the circuit reduces to a purely combination circuits. A circuit with flip-flops is considered a sequential circuit even in the absence of combinational gates. A circuit ~~with~~ ^{That include} flip-flops considered are usually classified by the function they perform rather than by the name of the sequential circuit. Two such circuits are registers & counters.

A register is a group of flip-flops each one of which shares a common clock and is capable of storing n bits of information. An n -bit register consists of a group of n flip-flops capable of storing n bits of binary information. In addition to flip-flops a register may have combinational gates that perform certain data processing tasks.

A Counter is essentially a register that goes through a predetermined sequence of binary states. The gates in the counter are connected in such a way as to produce the prescribed sequence of states.

The simplest register is one that consists of only flip flops with out any gates. A register constructed with four D-flip-flops to form a four-bit data storage register and the binary data available at the four inputs are transferred into the register. The value of (I_3, I_2, I_1, I_0) immediately before of the clock edge determines the value of (A_3, A_2, A_1, A_0) after the clock edge.



The four outputs can be sampled at any time to obtain the binary information stored in the register. The input clear-b goes to active-low R (reset) input of all four flip-flops. When this input goes to 0 all flip-flops are reset asynchronously. The clear-b input is useful for clearing the register to all 0's prior to its clocked operation. The R inputs must be maintained at logic 1 during normal clocked operation. Note that depending on the flip flop, either clear, clear-b, reset, reset-b can be used to indicate transfer of the register to an all 0's state.

Registers with Parallel Load

Registers with parallel load are a fundamental building block in digital systems. It is important that you have a thorough understanding of their behaviour. Synchronous digital systems have a master clock generator that supplies a continuous train of clock pulses. The pulses are applied to all flip-flops & registers in the system.

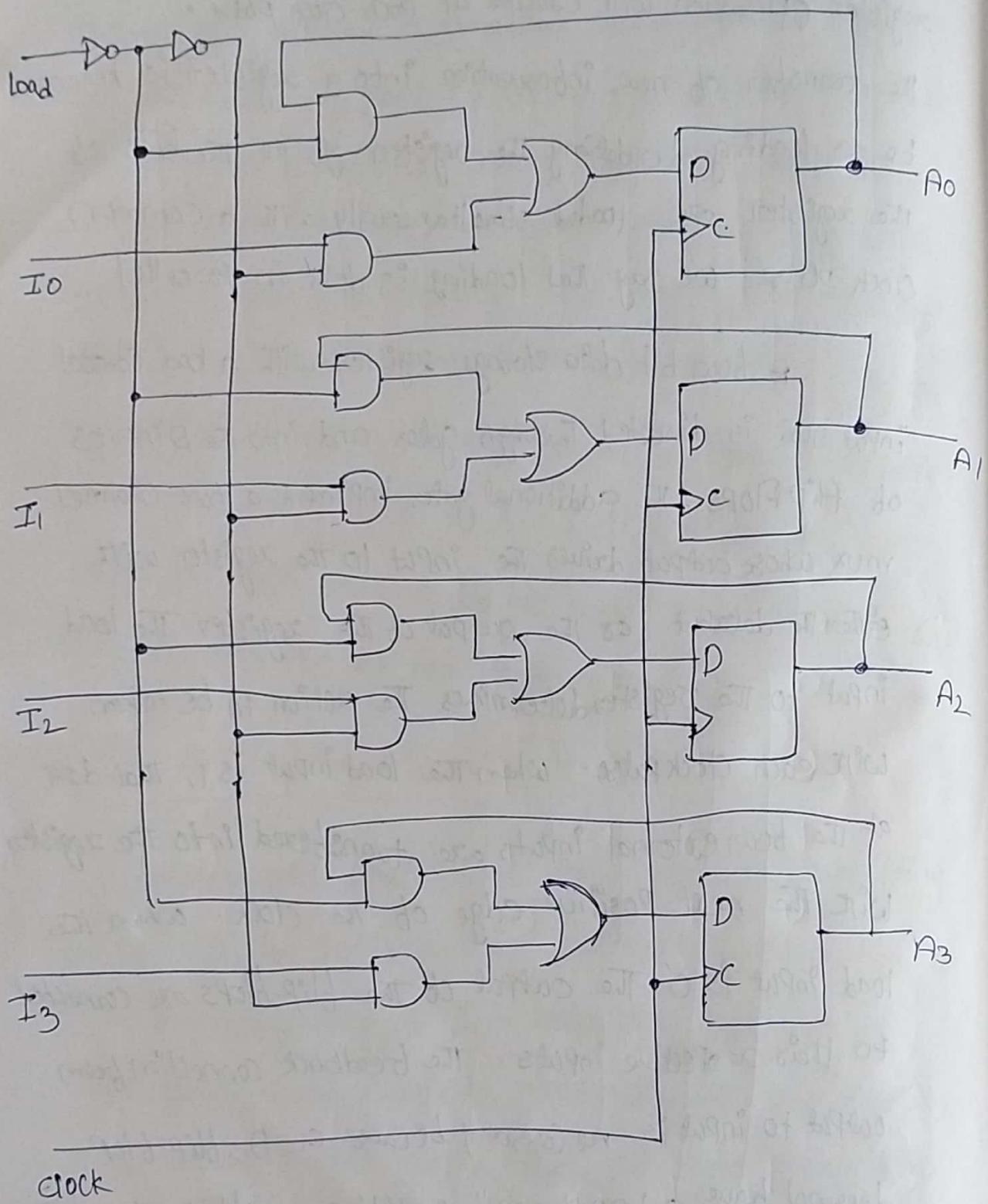
The master clock acts like a drum that supplies a constant beat to all parts of the system.

A separate control signal must be used to decide which register operation will execute at each clock pulse.

The transfer of new information into a register is referred to as loading or updating the register. If all the bits of the register are loaded simultaneously with a common clock pulse, we say that loading is done in parallel.

A four bit data storage register with a load control input that is directed through gates and into the D inputs of flip flops. The additional gates implement a two-channel MUX whose output drives the input to the register with either the data bus or the output of the register. The load input to the register determines the action to be taken with each clock pulse. When the load input is 1, the data at the four external inputs are transferred into the register with the next positive edge of the clock. When the load input is 0, the output of the flip-flops are connected to their respective inputs. The feedback connection from output to input is necessary because a D-flip-flop does not have a 'no change' condition. With each clock edge, the D input determine the next state of the register.

Synchronous



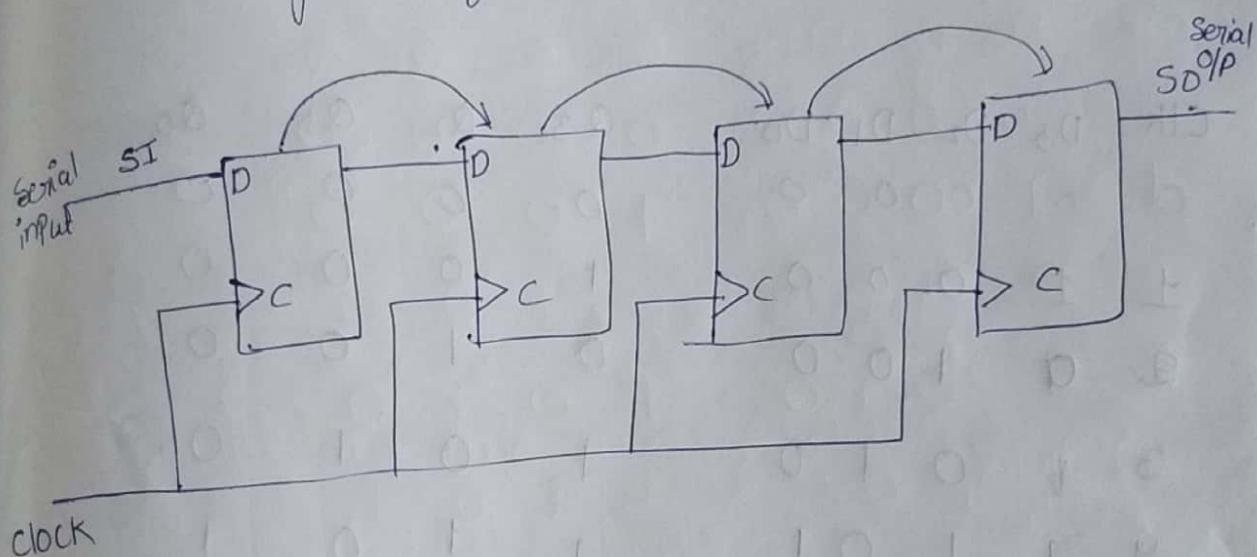
Four-bit register with parallel load

Shift Registers

A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction is called shift register. A register capable of shifting its binary information to the left or right.

LSR → Left Shift Register

RSR → Right Shift Register



In Registers data transfer is in two types

→ Serial :- In this data transfer 1-bit

→ Parallel :- data transfer all at a time.

Data transfer entering & Retriving Methods:

→ SISO 1-bit to 1-bit

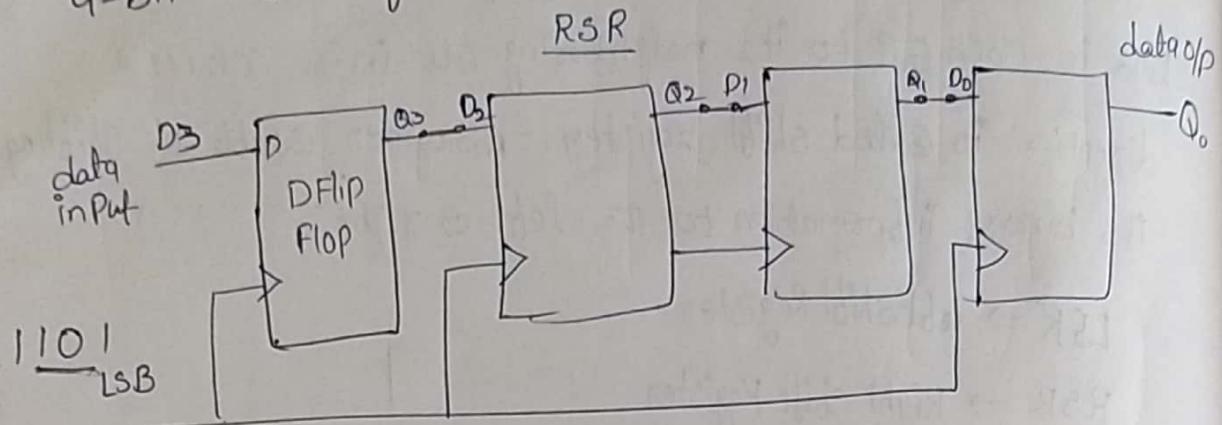
→ SIPO 1-bit to All

→ PIPO All bits

→ PISO

SISO :- Right Shift Register

We are going to insert 4 bit data we need 4-bit shift register.



CLK	D ₃	D ₂	D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0
2	a	1	0	0	0	1	0	0
3	1	0	1	0	1	0	1	0
4	1	1	0	1	1	1	0	1
5	0	1	1	0	0	1	1	0

$$\text{Time} = T$$

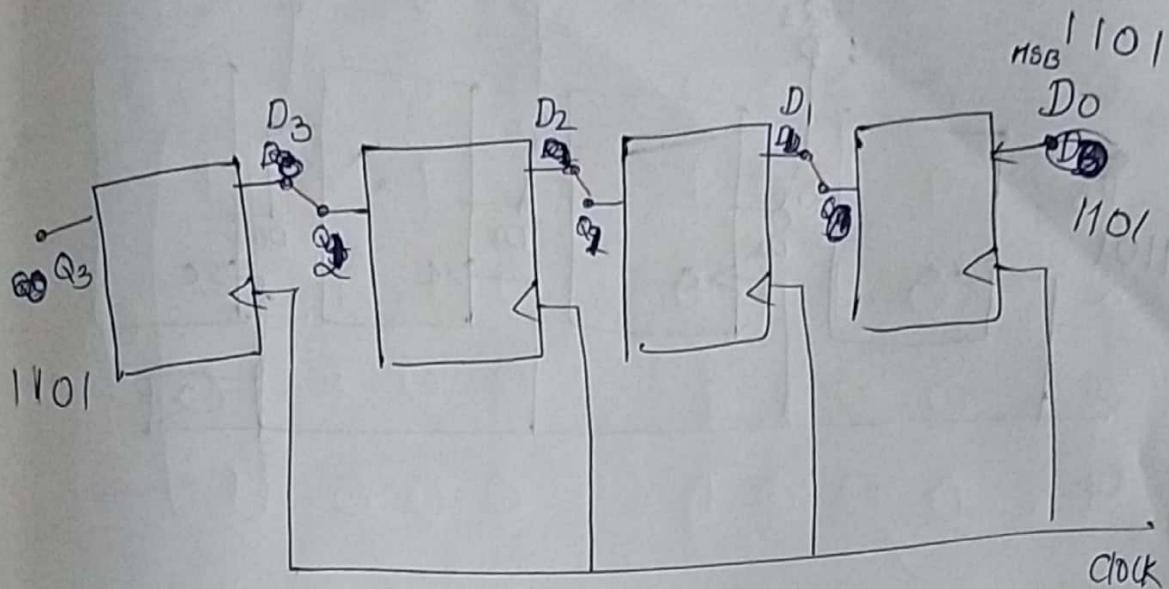
n-bits = n-clock pulse

$$\text{Data transfer time delay} = n \times T$$

6	0	0	1	1	0	0	1	1
7	0	0	0	1	0	0	0	1

n-1 = clock pulse used to receive data at output

4 bit Shift Register



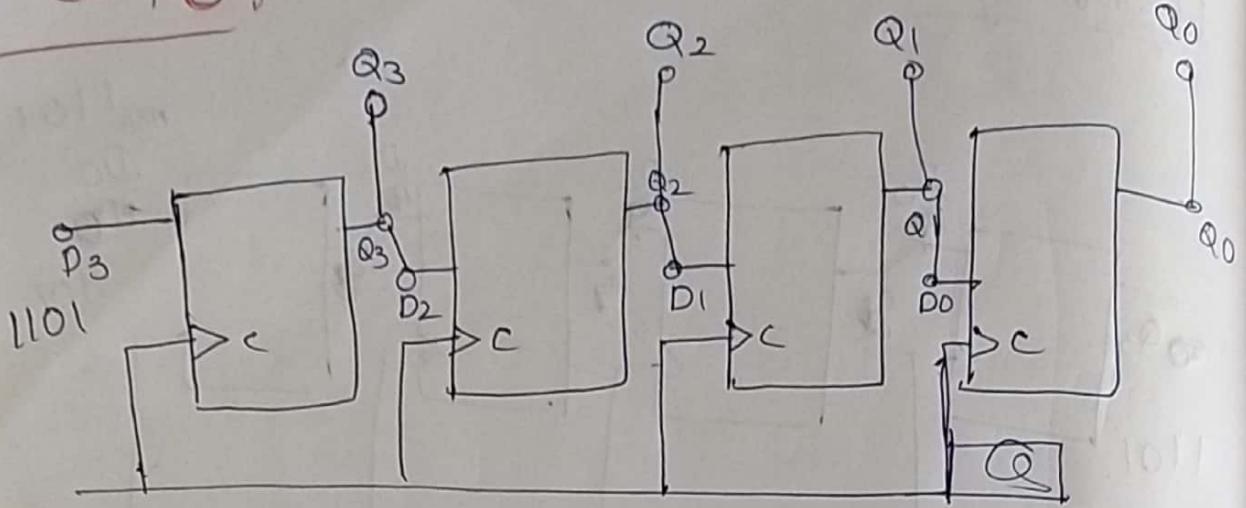
Input = 1101

	D ₃	D ₂	D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	1
3	0	1	1	0	0	1	1	0
4	1	1	0	1	1	0	1	0
5	1	0	1	0	1	0	0	0
6	0	1	0	0	0	1	0	0
7	1	0	0	0	0	0	0	0

Clock Pulse

↓ O/P

SI PO:

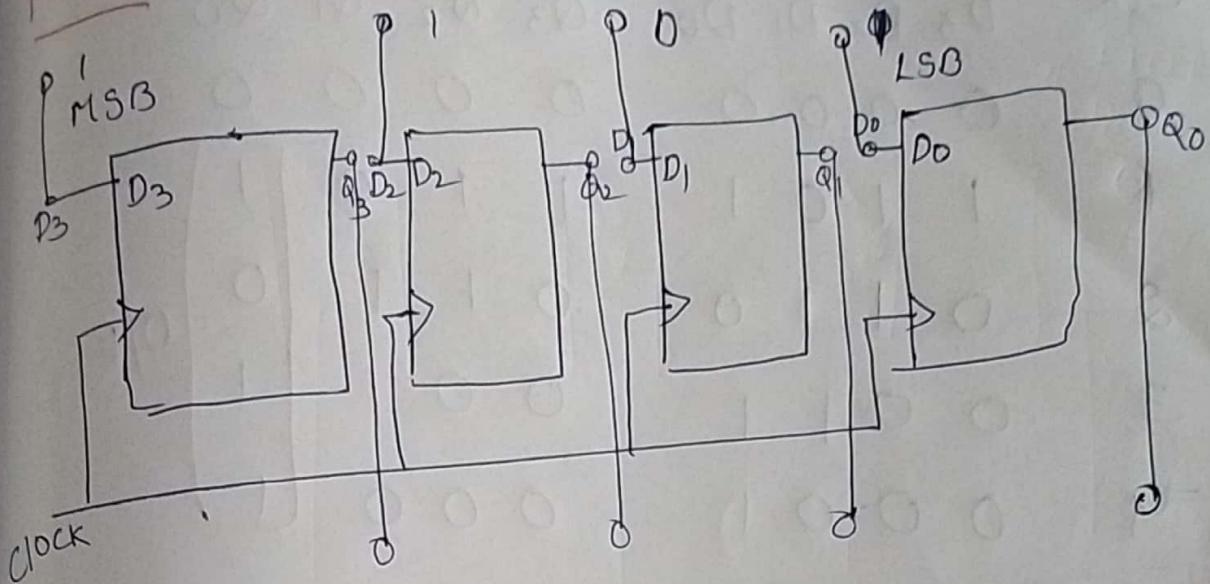


Clk	D ₃	D ₂	D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0
2	0	1	0	0	0	1	0	0
3	1	0	1	0	1	0	1	0
4	1	1	0	1	1	1	0	1

nxt

Output is connected in parallel so we want to get o/p no need of clock pulse we connected o/p in parallel so output will get directly.

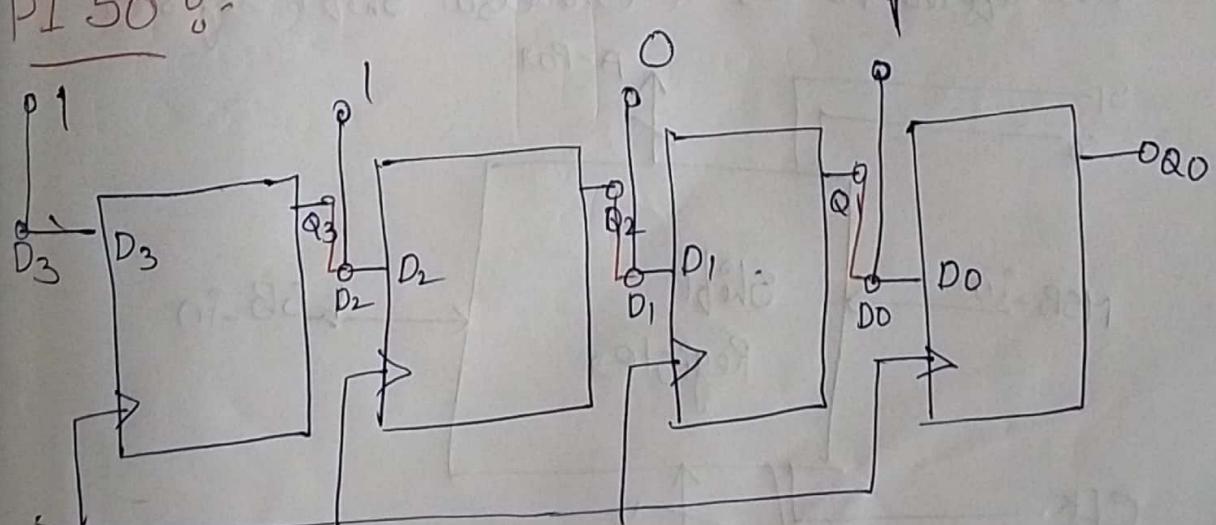
PIPO :-



Clock	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	0	1

If we want to retain data we need '0' clock pulse
because already output is get directly in 1/le shift
register.

PI SO :-



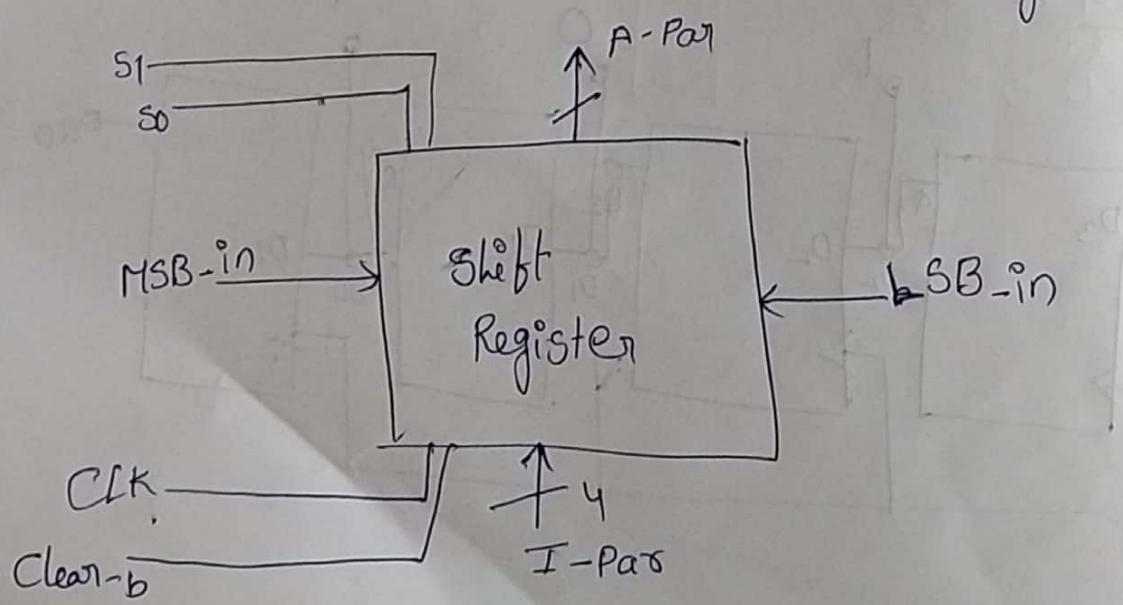
CLK	D ₃	D ₂	D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	0	1	1
2	0	1	1	0	0	1	1	0
3	0	0	1	1	0	0	1	1
4	0	0	0	1	0	0	0	1

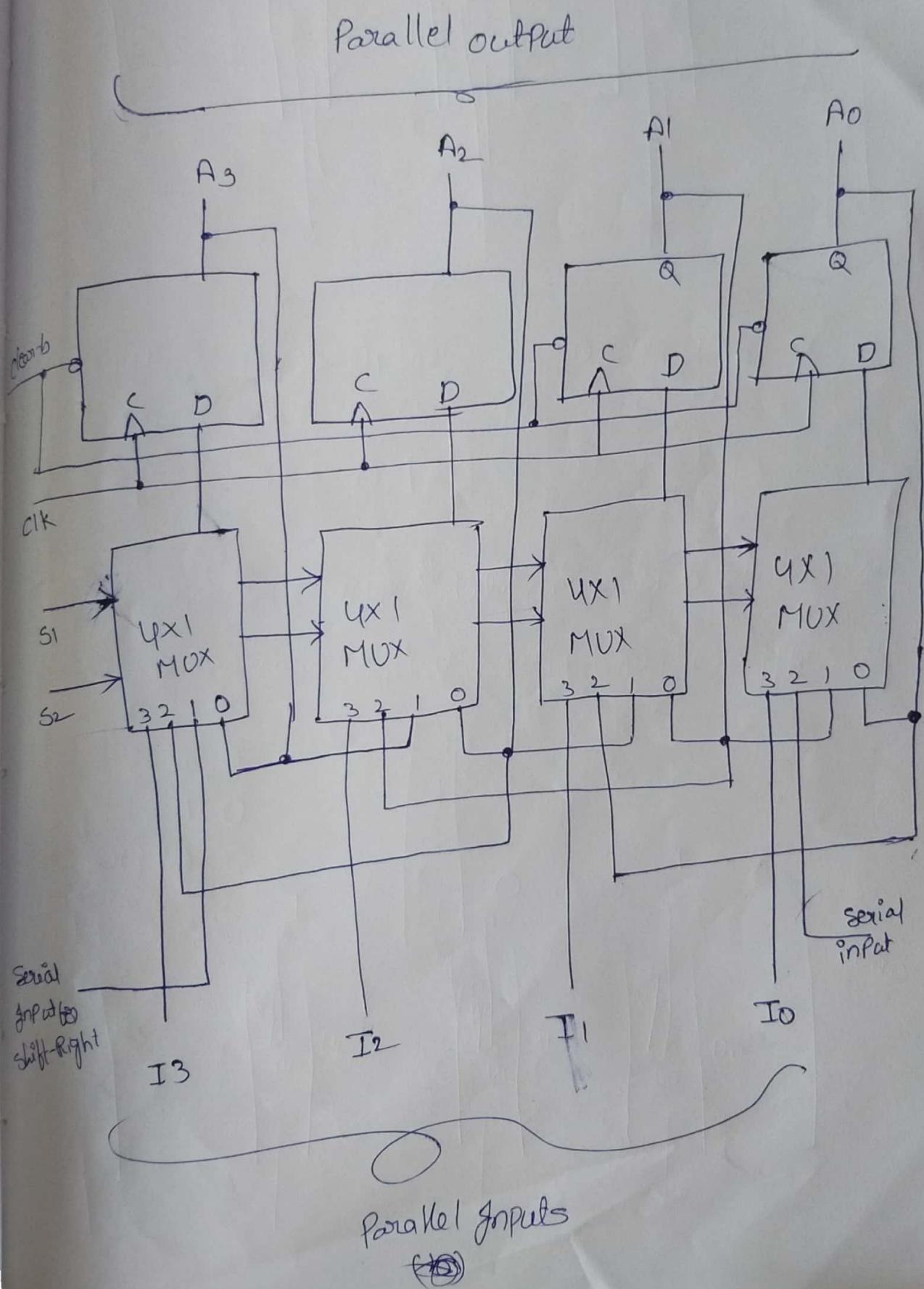
1 clock pulse - enter
 n-1 clock pulse - set to zero

Universal Shift Register:

Universal Shift Register is shift bits left and Right shifts. If a shift register doing Left and Right Shifts is called Universal Shift Register.

If the register has both shifts and Parallel-load capabilities it is referred to as a Universal shift register.





Counters:

→ Counter is a digital device used to count number of pulses & it can also be used as frequency divider.

→ Counter Generally used for Counting.

→ Counter can count in two ways

1. Upcount ($0, 1, 2, \dots, N$)

2. Downcount ($N, N-1, \dots, 1, 0$)

→ Up counter is used in EVM's, down counters are used in space related counts.

→ Present count of the counter represents state of the counter.

→ Counter contains set of flip-flops, A n bit counter need n -flipflops & 2^n states $N = 2^n$

$$\text{Max count} = 2^n - 1 \quad (0-7) \text{ if } 2^3 = 8$$

→ Each state frequency = $\frac{\text{total frequency}}{2^n}$

→ Mod 2, Mod 10, Mod 16

→ State frequency is nothing but in each state how much time it is in.

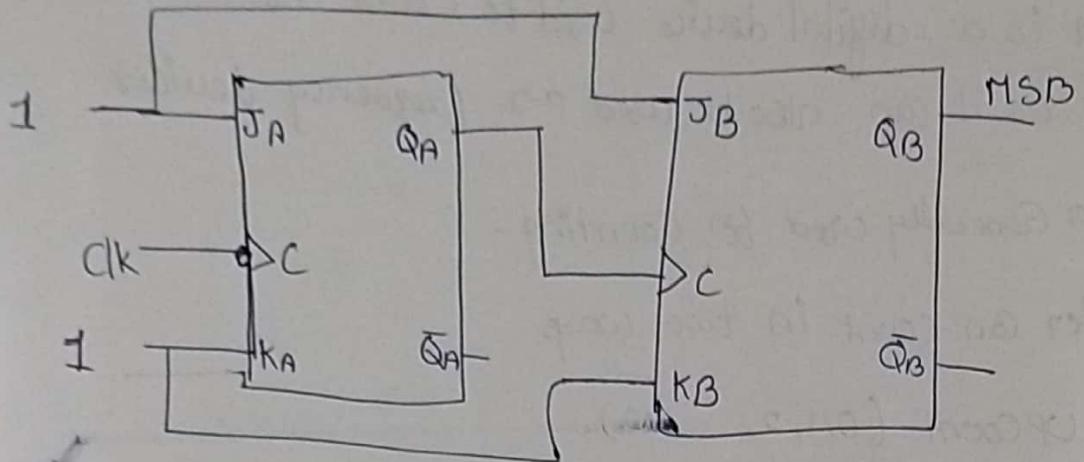
→ according to clock pulse counters are divided into two types

→ Asynchronous counter.

→ Synchronous counter.

Synchronous counter is nothing but all the flipflops connected to single clock.

Asynchronous counter is nothing but each flip-flop connected with different clock pulse.

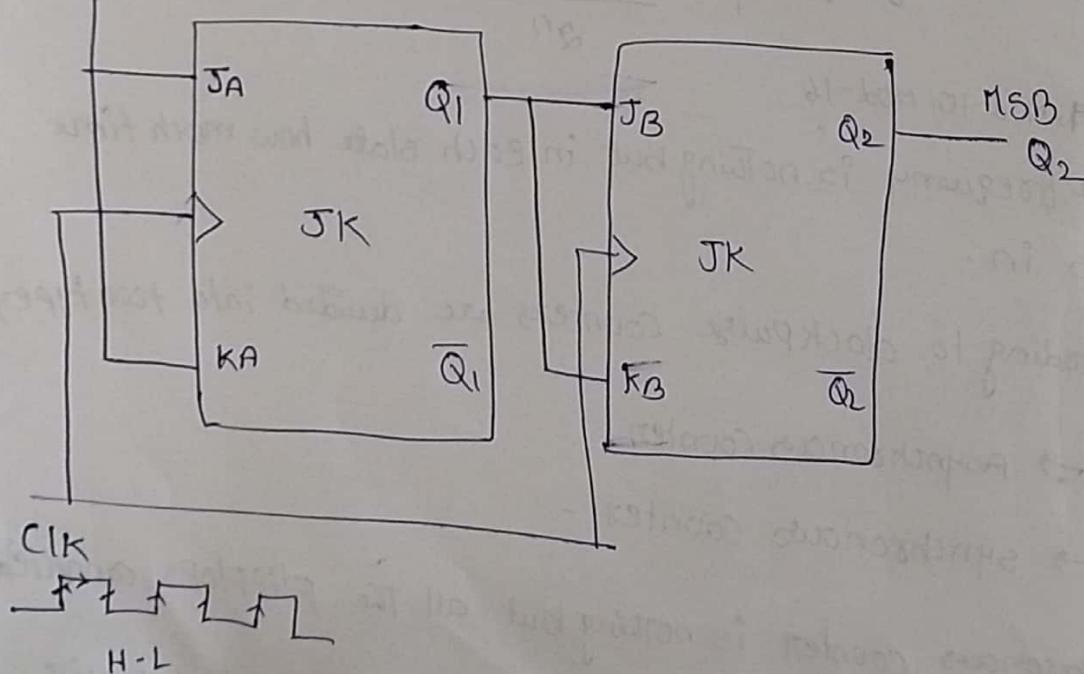


Asynchronous Counter

Synchronous Counter

In synchronous counter we need to use similar clock pulse to the flip-flops.

high (Logic 1)



Clk	JA	KA	Q1	JB	KB	Q2	Q2	Q1
0	x	x	0	x	x	0	0	0
1	1	1	1	0	0	0	0	1
2	1	1	0	1	1	1	1	0
3	1	1	1	0	0	0	1	1
4	1	1	0	1	1	0	0	0

for the synchronous down counter we are using -ve triggered clock pulse the input to second flip flop is going to connected with complement output of first Flip Flop.

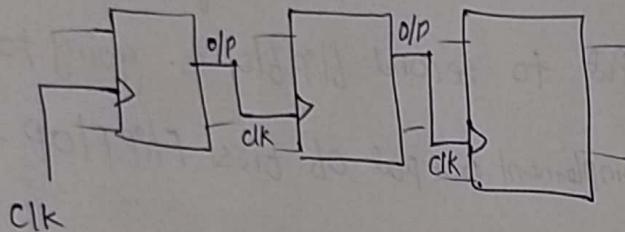
synchronous counter are all clocked together at the same time with same clock signal. Due to this common clock pulse all output states switch or change simultaneously. In synchronous counters, the clock inputs of all the flip flops are connected together and are triggered by the input pulses. Thus all the flip flops change simultaneously.

Asynchronous Counter:

Asynchronous refers to states that doesn't have a fixed time relationship with each other.

In Asynchronous counter Flip Flops does not have a common clock pulse so their state doesn't change exactly at same time.

1st Flip-Flop has clock, output of each flip-flop will act as clock to the next flip-flop in the counter.



example for Asynchronous Counter is Ripple Counter.
In any Asynchronous Counter we are using JK-FlipFlop (or) T FlipFlop
That all in toggle condition mode

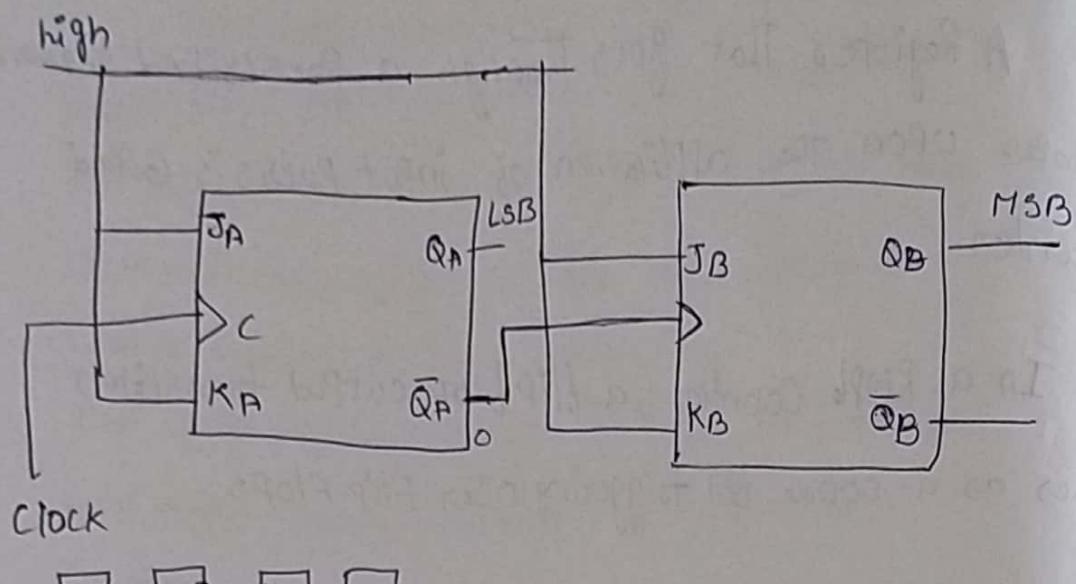
Ripple Counter

A Register That goes through a Prescribed sequence of states upon the application of input pulses is called a Counter.

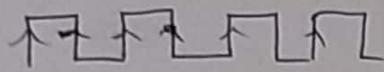
In a Ripple Counter , a flip flop output transition serves as a source for triggering other Flip-Flops

Binary Ripple Counter: A binary Ripple Counter consists of a series connection of complementing flip-flops, with the output of each flip-flop connected to the C input of the next higher order flip flop. A Complementing flip flop can be obtained from a JK-Flip Flop with the J and K inputs tied together or from a T-Flip Flop. A third possibility is to use a D Flip Flop with the Complement output connected to the D input. The D input is always the complement of the present state, and the next clock pulse will cause the flip flop to complement.

Ripple UP Counter:-

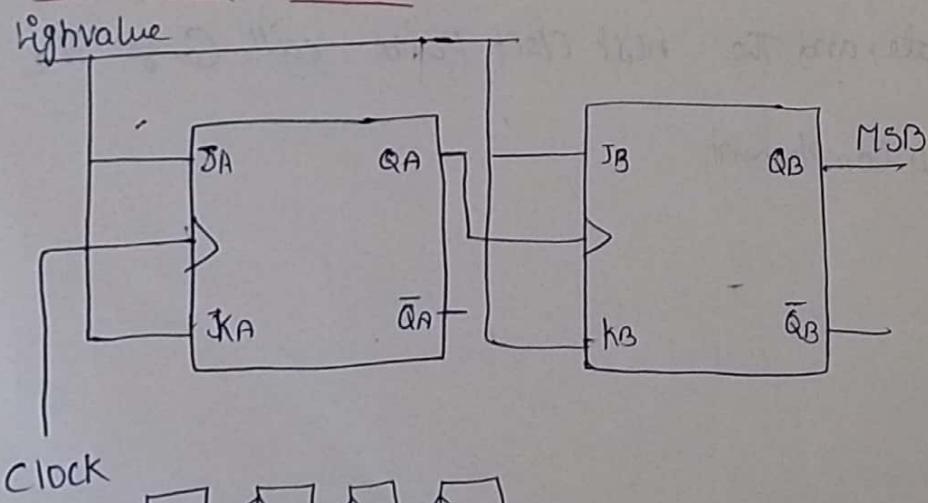


Clock

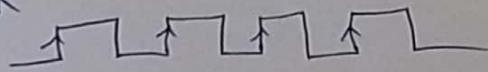


CLOCK	J _A	K _A	J _B	K _B	Q ₂	Q ₁	Q ₁ '
0	&	&	&	&	0	0	0
1	1	1	x	x	0	1	0
2	1	1	1	1	1	0	1
3	1	1	x	x	1	0	1
4	1	1	1	1	1	1	0
					0	0	1

Ripple Down Counter:-



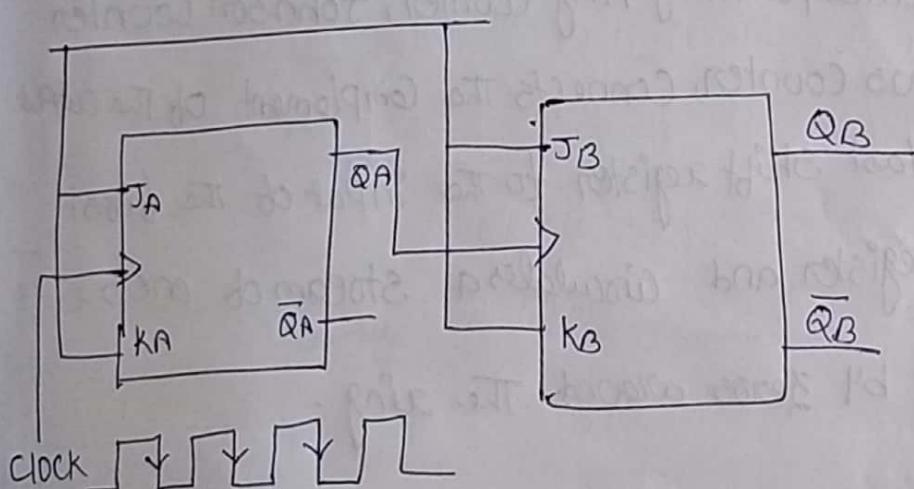
CLOCK



positive edge triggered clock

CLOCK	J _A	K _A	J _B	K _B	Q ₂	Q ₁	
0	x	x	x	x	0	0	
1	1	1	1	1	1	1	3
2	1	1	x	x	1	0	2
3	1	1	1	1	0	1	1
4	1	1	x	x	0	0	0

-ve edge trigger Ripple Up Counter:



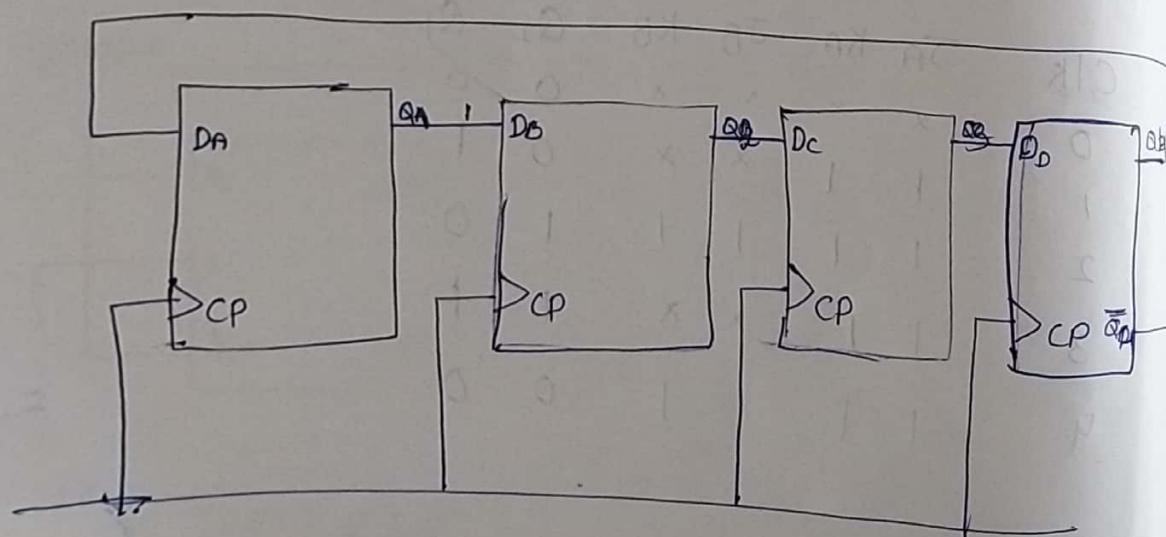
C K	J _A	K _A	J _B	K _B	Q ₂	Q ₁	
0	x	x	x	x	0	0	
1	1	1	x	x	0	1	
2	1	1	1	1	1	0	
3	1	1	x	x	1	1	
4	1	1	1	1	0	0	

Other Counters:

Twisted Ring Counter (or) Johnson's Counter:-

→ A ring counter is a type of counter composed of flip-flops connected into a shift register with the output of the last flip-flop fed to the input of the first, making a "circular" or "ring" structure.

A twisted ring counter also called switch-to-ring counter, walking ring counter, Johnson counter or Möbius counter, connects the complement of the output of the last shift register to the input of the first shift register and circulates a stream of ones followed by zeros around the ring.



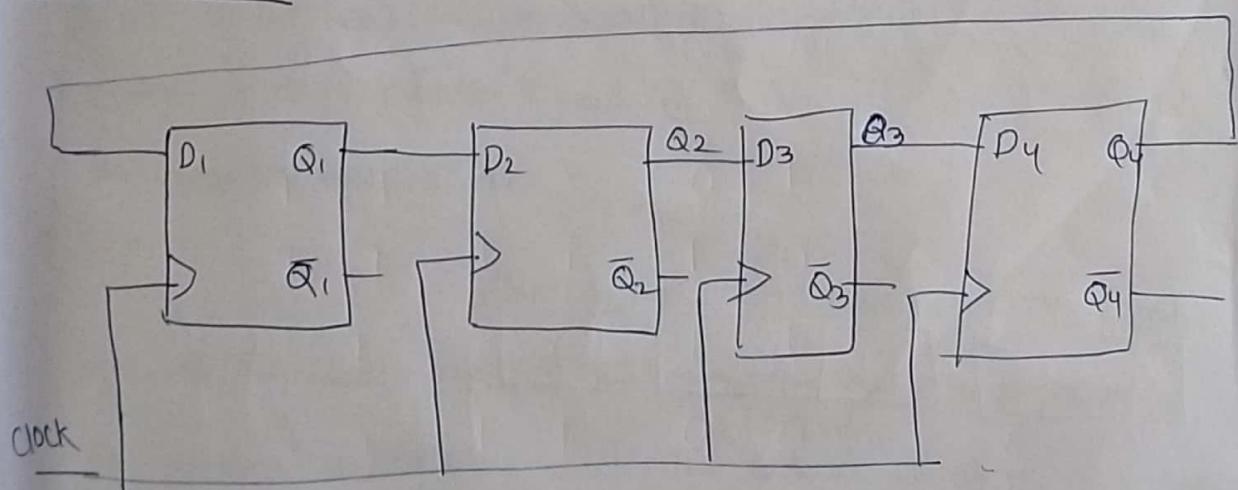
Johnson Counter

Johnson counter :-

CLK	Q ₄	Q ₃	Q ₂	Q ₁
0	0	0	0	1
1	0	0	1	1
2	0	1	1	1
3	1	1	1	1
4	1	1	1	0
5	1	1	0	0
6	1	0	0	0
7	0	0	0	0
8	0	0	0	1

Each state frequency = $\frac{\text{total frequency}}{2N}$

Ring counter :



θ_{ik} θ_4 θ_3 θ_2 θ_1

0	0	0	0	/
1	0	0	/	0
2	0	/	0	0
3	/	0	0	0
4	0	0	0	0
5	0	0	0	/