Unit 5: Sequential Logic

Sequential Logic Circuit

Sequential circuits is a combinational logic circuit to which memory elements are connected to form a feedback path. The memory elements are devices capable of storing binary information within them. The binary information stored in the memory elements at any given time defines the *state* of sequential circuit.

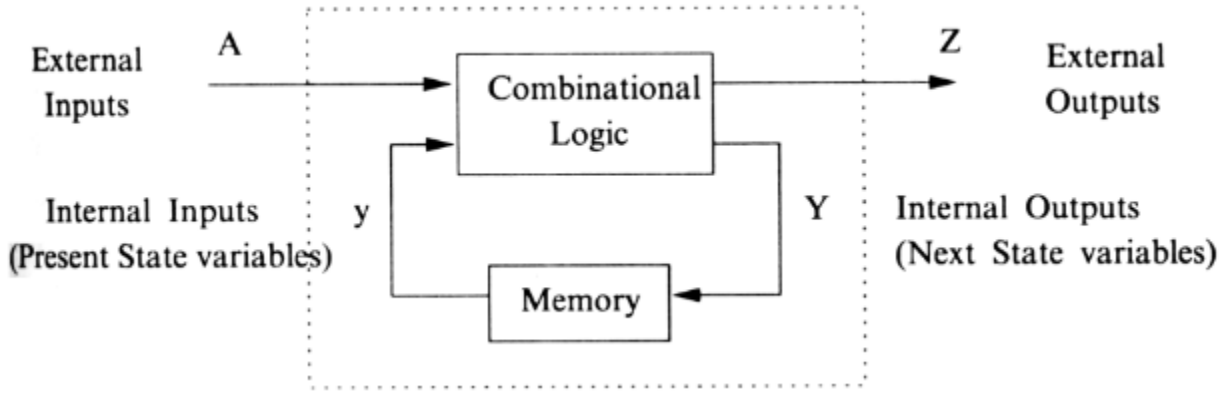
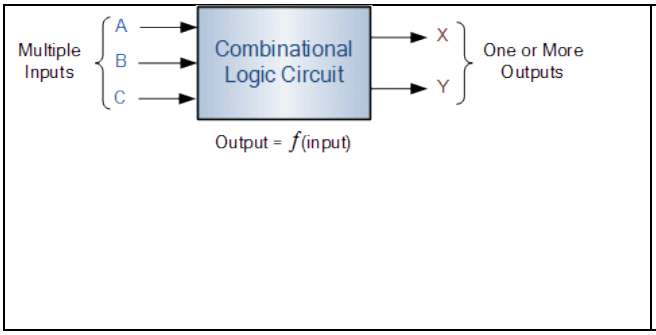


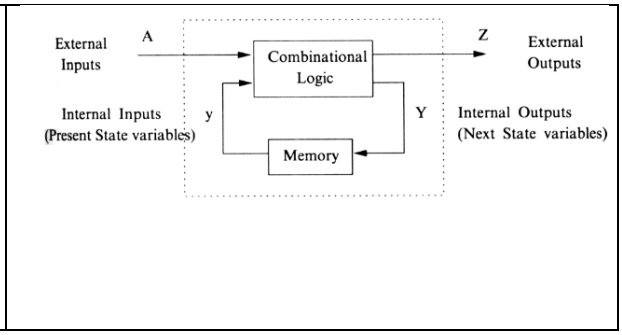
Fig: 5.1: 1: Block Diagram of sequential circuit

This sequential circuit contains a set of inputs and output(s). The output(s) of sequential circuit depends not only on the combination of present inputs but also on the previous output(s). Previous output is nothing but the present state. Therefore, sequential circuits contain combinational circuits along with memory (storage) elements. Some sequential circuits may not contain combinational circuits, but only memory elements.

- Sequential circuits employ memory elements (binary cells) in addition to logic gates
- Their outputs are function of input and the state of the memory elements. As a consequence, the output of a sequential ckt depends on not only present inputs but also on past inputs.

Combinational Circuits	Sequential Circuits
Outputs depend only on present inputs.	Outputs depend on both present inputs and present state.
Feedback path is not present.	Feedback path is present.
Memory elements are not required.	Memory elements are required.
Clock signal is not required.	Clock signal is required.
Easy to design.	Difficult to design.
Figure:	Figure:





There are two types of sequential circuits. They are:

1. Asynchronous sequential circuits
2. Synchronous sequential circuits

Asynchronous sequential circuits

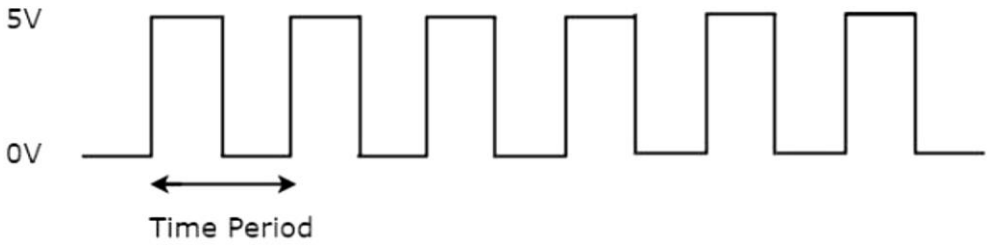
If some or all the outputs of a sequential circuit do not change (affect) with respect to active transition of clock signal, then that sequential circuit is called as Asynchronous sequential circuit. Behavior of the Asynchronous circuits depends upon the order in which its input signals change and can be affected at any instance of time. The memory elements commonly used in asynchronous sequential circuits are time delay devices.

Synchronous sequential circuits

If all the outputs of a sequential circuit change (affect) with respect to active transition of clock signal, then that sequential circuit is called as Synchronous sequential circuit. That means, all the outputs of synchronous sequential circuits change (affect) at the same time. Therefore, the outputs of synchronous sequential circuits are in synchronous with either only positive edges or only negative edges of clock signal.

Clock signal

Clock signal is a periodic signal and it's ON time and OFF time need not be the same. We can represent the clock signal as a square wave



Latches

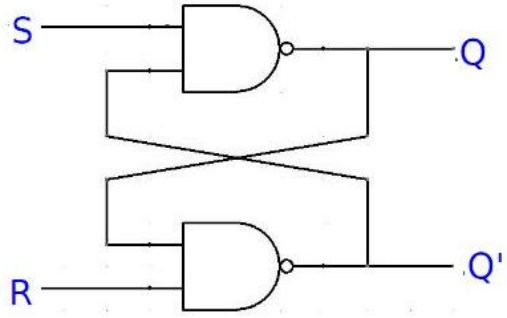
A latch is a circuit that has two stable states which can be used to store one binary digit. Flip-flops and latches are fundamental building blocks used in many sequential circuits and larger storage devices. A latch is a memory device used to store one bit of data. These are same like flip-flops, however, they are not synchronous devices. They do not work on edges of the clock as flip flops do. It means latches immediately changes the stored information when the input is changed.

- Latch are basic storage device that can store 1 bit.
- D Latches are typically used as I/O ports in asynchronous systems
- Latch is faster because it does not wait for a clock signal.
- Latches requires less power to operate

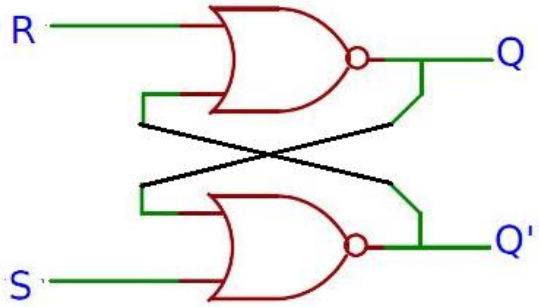
Different types of Latch are:

SR latch:

An SR (Set/Reset) latch is an asynchronous circuit, which works depending on the S-state & R-inputs. The SR-latch can be constructed using 2-NOR gates with a cross loop connection. These latches can also be built with NAND gates.



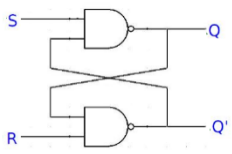
SR latch with 2 NAND gates



SR latch with 2 NOR gates

* To get a required functionality when designing a SR latch using NAND gate S input must be in upper NAND gate (with output Q) and R input must be in Lower NAND gate(with output Q').

* However the S input of SR latch designed using NOR gate must be in lower part (Complementary part i.e. Q') and R input must be in upper NOR gate (Normal part i.e. Q). Otherwise required functionality will not come.



Truth table for SR

Inputs			Next state	Comment
S	R	Present State (Q_n)	Q_{n+1}	
0	0	0	0	No Change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	i.e 0
1	0	0	1	Set
1	0	1	1	i.e 1
1	1	0	×	Invalid
1	1	1	×	Invalid

Explanation:

I. When S=0, R=0

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

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

i.e next state(output) will be Q_n whatever is the present state. [No change]

II. When S=0, R=1

$Q_{n+1}= \overline{\overline{Q_n}+1} = \overline{1} = 0$ [1+x=1]

$\overline{Q_{n+1}}= \overline{Q_n+0} = \overline{Q_n}$ [if Q_n is 0 then $\overline{Q_n}$ is 1]

So output will be 0 whatever is the Q_n(Present state)

III. When S=1, R=0

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

$\overline{Q_{n+1}}= \overline{Q_n+1} = \overline{1} = 0$ [i.e x+1=1] [if $\overline{Q_{n+1}}$=0 obviously Q_n =1]

IV. When S=1, R=1

$Q_{n+1}= \overline{\overline{Q_n}+1} = \overline{1} = 0$ [i.e. 1+x=1] 0 for both Q_{n+1} and $(Q_{n+1})'$ is not possible.

$\overline{Q_{n+1}}= \overline{Q_n+1} = \overline{1} = 0$ [i.e. 1+x'=1] So Undetermined state

Flip Flops

Flip flop is an important basic memory element for digital circuit. Flip-flop is designed by assembling different logic gates. Single logic gate does not have any information storing capacity but by combining different such gates one can make such a digital circuit which can store digital information. There are different types of flip flop with different characteristics for different application. Flip-flop is the basic building blocks of most sequential circuits.

- The memory elements used in clocked sequential circuits are called flip-flops.
- These circuits are binary cells capable of storing one bit of information.
- Flip flop have two output: one for normal value and another for the complement value

Characteristics table

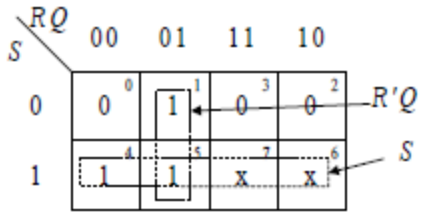
The *characteristic table* is useful during the analysis of sequential circuits when the value of flip-flop inputs are known and we want to find the value of the flip-flop output Q after the rising edge of the clock signal. As with any other truth table, we can use the k-map method to derive the characteristic equation for each flip-flop. The characteristic table is just the truth table but usually written in a shorter format. The characteristic table answers the question of what is the next state when given the inputs and the current state, and is used in the analysis of sequential circuits.

Inputs		Q_{Next}	Comment
S	R		
0	0	Qn	No Change
0	1	0	Reset
1	0	1	Set
1	1	×	Invalid

Characteristic Equation

The characteristic equation is the functional Boolean equation that is derived from the truth table. This equation formally describes the functional behavior of the flip-flop. Like the characteristic table, it specifies the flipflop's next state as a function of its current state and inputs.

For example the characteristic equation of SR flip flop is derived from Kmap as



Thus, the characteristic equation for the SR flip-flop is

Q_{next} = S + R'Q.

Excitation table

During the design process we usually know the transition from present state to the next state and wish to find the flip-flop input conditions that will cause the required transition. For this reason we will need a table that lists the required inputs for a given change of state. Such a list is called the *excitation table*.

The required input conditions are derived from the information available in the characteristic table. The symbol X in the table represents a "don't care" condition, that is, it does not matter whether the input is 1 or 0.

Excitation table for SR flip flop

State		Inputs	
$Q_{Present}$	Q_{Next}	S	R
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

The common types of flip flop are:

1. SR flip flop
2. D flip flop
3. JK flip flop
4. T flip flop

S-R flip-flop

The SR flip-flop, also known as a SR Latch, can be considered as one of the most basic sequential logic circuit. This simple flip-flop is basically a one-bit memory bistable (an electronic circuit which has two stable states.) device that has two inputs, one which will

"SET" the device (meaning the output = "1"), and is labelled S and one which will "RESET" the device (meaning the output = "0"), labelled R.

Then the SR description stands for "Set-Reset". The reset input resets the flip-flop back to its original state with an output Q that will be either at a logic level "1" or logic "0" depending upon this set/reset condition.

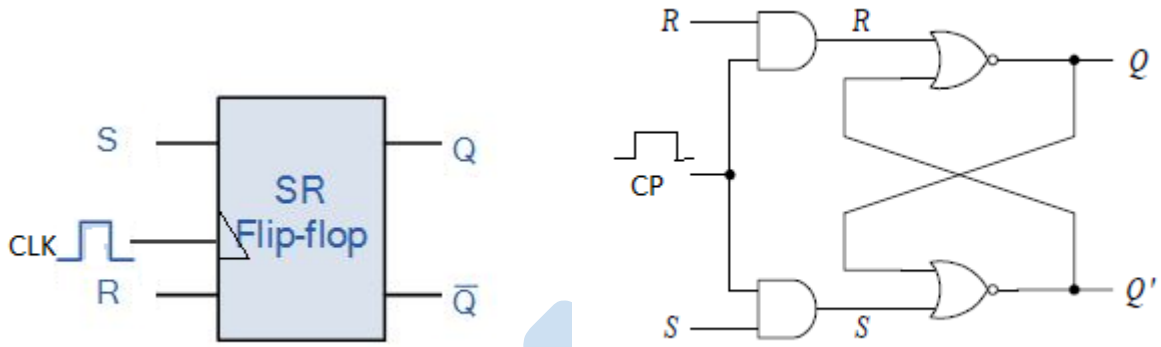


Figure 1Block diagram of SR flip flop | and logic diagram

- Each flip flop has two outputs Q and Q' and two inputs set and Reset
- A flip flop circuit can be constructed from two NAND gates or two NOT gates.
- The cross coupled connection from the output of one gate to the input of the other gate constitutes a feedback path.

Edged triggered SR flip flop

The basic flip flop is an asynchronous sequential circuit. It means there is not any clock signal which controls the output. So by adding AND gate to the inputs of the basic circuit, the flip flop can be made to respond to the input levels during the occurrence of clock pulse. The clocked RS flip flop is shown in above diagram. It consist of two NOR gates and two AND gates. The output of the two AND gates remains 0 as long as the clock pulse is 0, Regardless of S and R input values. When, the clock pulse goes to 1, information from S and R inputs is allowed to reach the basic flip flops.

The basic operation is illustrated below, along with the truth table for this type of flip-flop.

Truth table of SR flip flop

Inputs			Next state	Comment
S	R	Present State (Q_n)	Q_{n+1}	
0	0	0	0	No Change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	i.e 0
1	0	0	1	Set
1	0	1	1	i.e 1
1	1	0	-	Invalid
1	1	1	-	Invalid

Characteristics table of SR flip flop

S	R	$Q_{(next)}$	Comment
0	0	Q	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Invalid

Excitation Table of SR flip flop

State		Inputs	
$Q_{(present)}$	$Q_{(next)}$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D flip flop:

The D flip flop is a modification of clocked RS flip flop. One way to eliminate the undesirable condition of indeterminate state in SR flip flop is to ensure that inputs S and R are never equal to 1 at the same time. This is done in the D flip flop as shown in figure below. The D Input goes directly to the S input and its complement is applied to the R input.

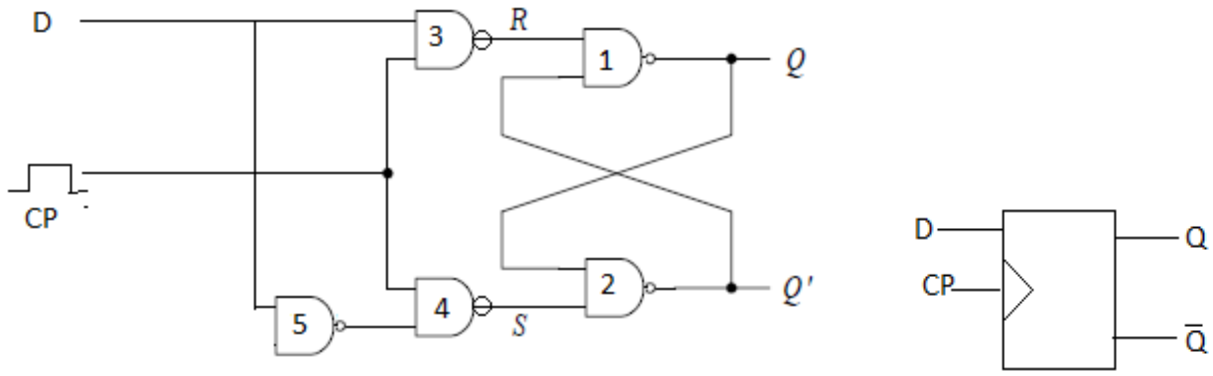


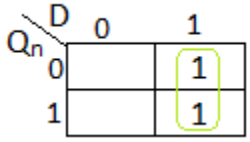
Figure 2Logic diagram of D flip flop and Graphical Symbol

Characteristic table of D flip Flop:

Q	D	Q_{n+1}	Comment
0	0	0	Clear
0	1	1	Set
1	0	0	Clear
1	1	1	Set

- As long as CP is 0, the output of gate 3 and 4 are at the 1 level and the circuit cannot change state regardless of the value of D
  - If the D is 1, the Q goes to 1, placing the circuit in the set state.
  - If the D is 0, output Q goes to 0 and the circuit switches to the clear state.

Characteristic Equation:



$Q_{n+1}=D$

Excitation Table

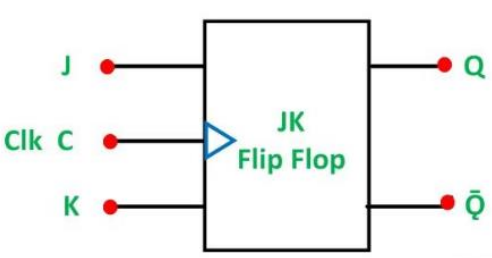
Present state	Next State	Input
Q_n	**Q_{n+1}**	**D**
0	0	0
0	1	1
1	0	0
1	1	1

Edge-triggered J-K flip-flop

The J-K flip-flop works very similar to S-R flip-flop. The only difference is that this flip-flop has NO invalid state. The JK flip flop is the improved version of SR flip flop. The outputs

toggle (change to the opposite state) when both J and K inputs are HIGH (in SR invalid state).

* JK flip flop is just like a SR flip flop but there is no undetermined state.



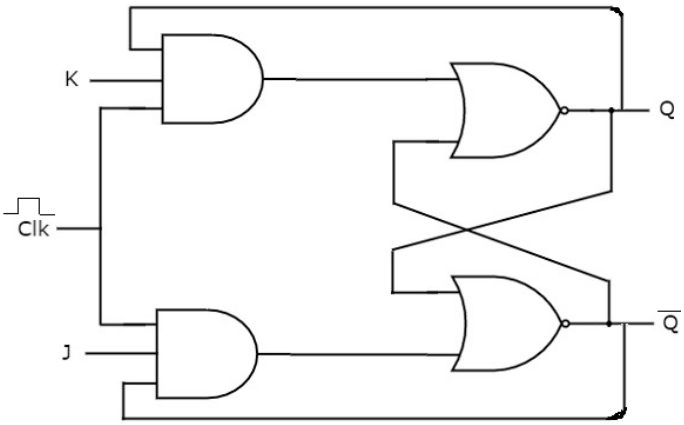


Figure 3Block Diagram of JK flip flop and Logic diagram

Inputs J and K behave like input S and R to set and reset the flip flop. In a JK flip flop, the letter J is for set and letter K is for clear. When inputs are applied to both J and K simultaneously the flip flop switches to its complement state, that is, if Q=1, it switches to Q=0 and vice versa.

In a clocked JK flip flop output Q is $ANAD_{ed}$ with K and CP inputs so that the flip flop is cleared during a clock pulse only if Q was previously 1.

Similarly, output Q' is AND_{ed} with J and CP inputs so that flip flop is set with a clock pulse only if Q' was previously 1.

The truth table is shown below.

Same as SR flip flop but toggles its state when J=1 and K=1

Inputs		Outputs	Comment
J	K	Q	
0	0	Q	No Change
0	1	0	Reset
1	0	1	Set
1	1	Q'	Toggle

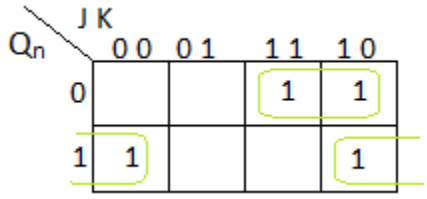
Characteristic table of JK flip flop

Inputs			Next state	Comment
J	K	Present State (Q_n)	Q_{n+1}	
0	0	0	0	No Change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	i.e 0
1	0	0	1	Set
1	0	1	1	i.e 1
1	1	0	1	toggle
1	1	1	0	toggle

Excitation Table of JK flip flop

State		Inputs	
$Q_{(present)}$	$Q_{(next)}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Characteristic Equation of JK flip flop Q_n



$Q_{n+1}= (Q_n)'J+Q_nK'$

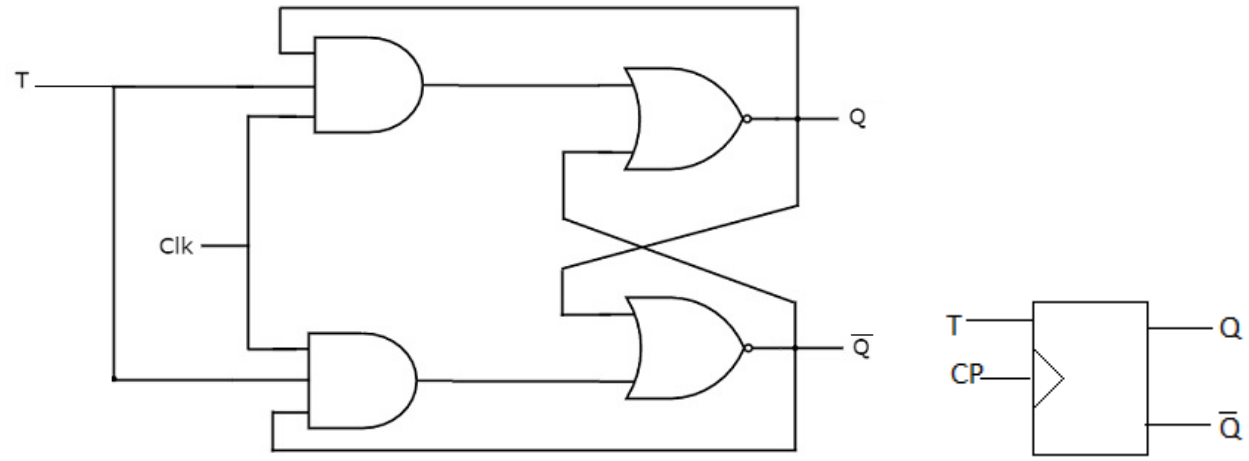
T flip flop:

T flip flop is the single input version of the JK flip flop and is obtained from the JK flip flop when both inputs are tied together. The designation T comes from the ability of the flip flop to "toggle" or complement its state. Regardless of the present state of the flip flop, it assumes the complement state when the clock pulse occurs while input T is at logic 1.

The characteristic table and characteristic equation show that:

- ⊕ When T=0, Q(t+1)=Q, that is , the next state is same as the present state and no changes occurs.

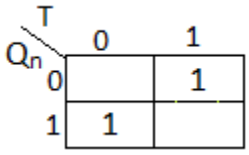
⊕ When T=1, Q(t+1)=Q, and the state of the flip flop is complemented.



Characteristic table:

Present state	Input	Next state	Comment
Q_t	T	Q_{t+1}	
0	0	0	Same as present state
0	1	1	Complemented of present state
1	0	1	Same
1	1	0	Complemented

Characteristic equation:



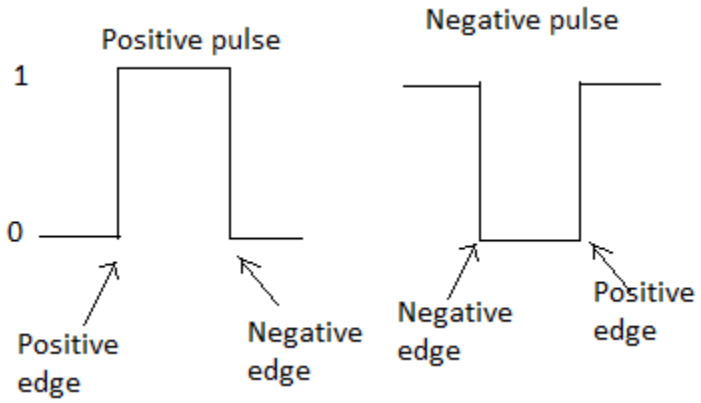
Q_{t+1}= Q'T + QT'

Triggering of Flip flops:

The state of flip flop is switched by a momentary change in the input signal. This momentary change is called a trigger and the transition it causes is said to trigger the flip flop. Clocked flip flops are triggered by pulses. A pulse starts from an initial value of 0, goes momentarily to 1, and after a short time, returns to its initial 0 value.

A clock pulse may be either positive or negative.

- A positive clock source remains at 0 during the interval between pulse and goes to 1 during the occurrence of a pulse. The pulse goes through two signal transitions: from 0 to 1 and return from 1 to 0. As shown in figure below, the positive transition is defined as the positive edge and the negative transition as negative edge.
- The definition applies also to negative pulse



One way to achieve edge triggering is to use a master slave or edge triggered flip flop.

Master slave flip flop

Master slave JK flip flop:

The output of the master JK flip flop is fed to the input of the slave flip flop. The out of the slave JK flip flop is given as a feedback to the input of the master JK flip flop. The CP is given to the master flip flop and it is sent through a NOT gate and thus inverted before passing it to the slave JK flip flop.

Master slave D flip flop:

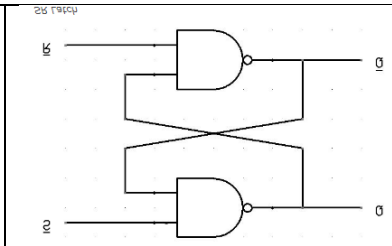
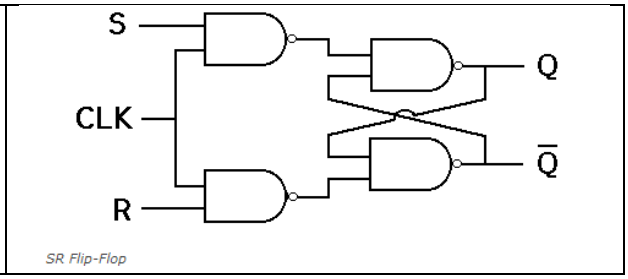
The circuit consist of two D flip flop connected together. When the clock is high the D input is stored in the first flip flop but second flip flop cannot change its state. When the clock is low, the first flip flop output is stored in the second flip flop but the first flip flop cannot change the state.

Edge- triggered flip flops

An 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. The three basic types of flip flop are: S-R, J-K and D.

The S-R, J-K and D inputs are called synchronous inputs because data on these inputs are transferred to the flip-flop's output only on the triggering edge of the clock pulse. On the other hand, the direct set (SET) and clear (CLR) inputs are called asynchronous inputs, as they are inputs that affect the state of the flip-flop independent of the clock. For the synchronous operations to work properly, these asynchronous inputs must both be kept LOW.

	Latches		**Flip flops (FF)**
	Latches are Asynchronous		Flip flops are Synchronous
1	Structure of latches are built logic gates	1	Flip flops are designed with latches by adding an extra clock signal.
2	Latch does not contain any clock signal	2	Flip flop contains a clock signal
3	Latches are very fast as compared to flip flops	3	Flip flops are slow compared to latches
4	Latches consumes less power	4	Flip flops consumes more power.
5	Latches are not capable of working as a register because of the lack of a CLK signal.	5	FFs are capable of working as a register because they come by a CLK
6	The working manner of latch is asynchronous which means, the output produced from the latch will depends on the inputs.	6	The flip continuously checks its inputs and changes its output only at times determined by clock signal.

7		7	

Flip-Flop operating characteristics

The operating characteristics specify the performance, operating requirements, and operating limitations of the circuit.

Propagation Delay Time - is the interval of time required after an input signal has been applied for the resulting output change to occur.

Set-Up Time - is the minimum interval required for the logic levels to be maintained constantly on the inputs (J and K, or S and R, or D) prior to the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop.

Hold Time - is the minimum interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop.

Maximum Clock Frequency - is the highest rate that a flip-flop can be reliably triggered.

Power Dissipation - is the total power consumption of the device.

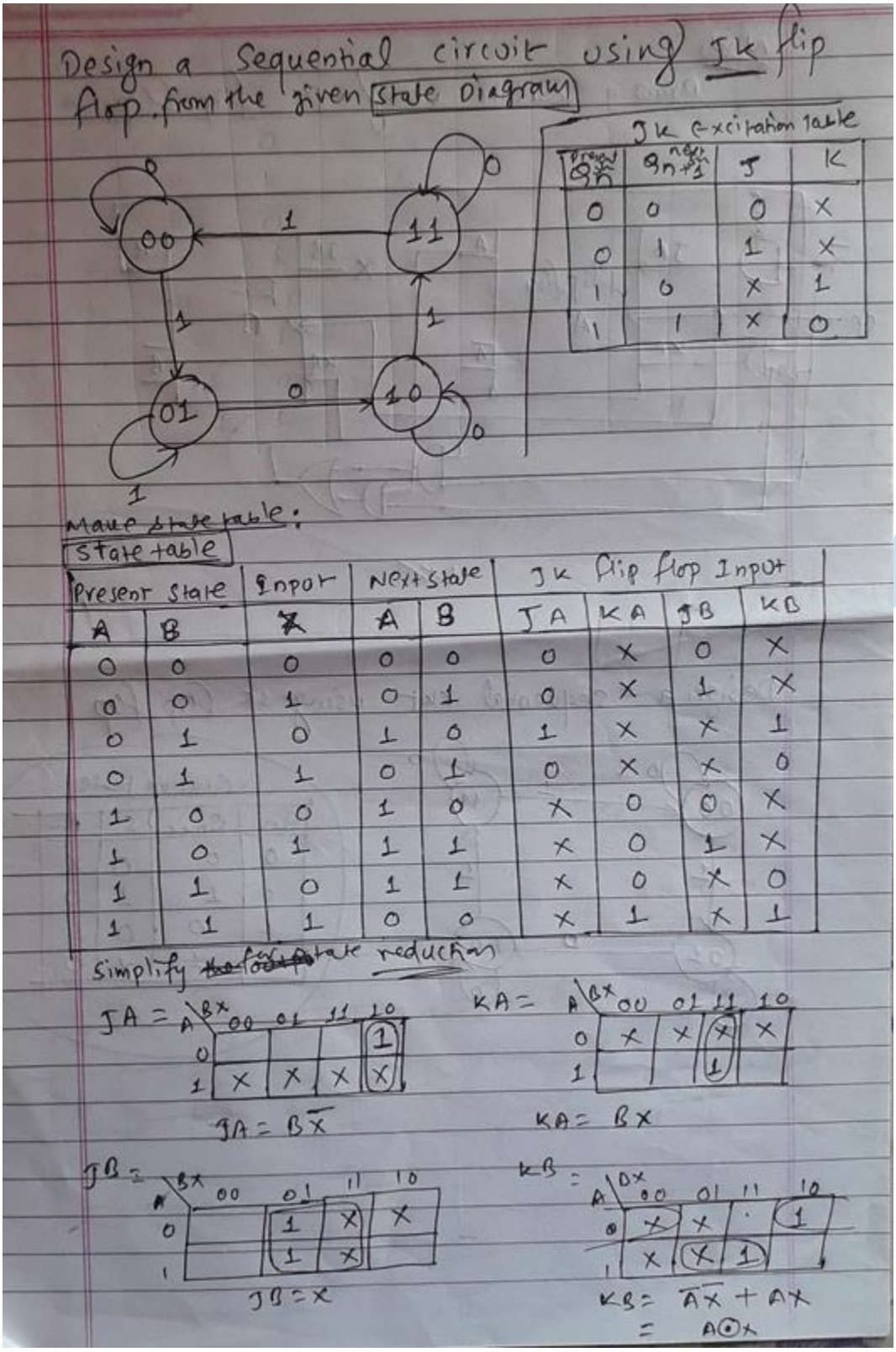
Pulse Widths - are the minimum pulse widths specified by the manufacturer for the Clock, SET and CLEAR inputs.

Flip Flop Application

Flip flops are the main components of sequential circuits. Flip flops can be used in many field in digital circuits. Particularly, edge triggered flip flops are very resourceful devices that can be used in wide range of applications like storing of binary data, counter, transferring binary data from one location to other etc. Some of the most common applications of flip – flops are

- Counters
- Registers

- Frequency Divider circuits
- Data transfer



Design a Sequential circuit using JK flip flop from the given state diagram

JK excitation table

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

Make state table:

State table

Present state		Input	Next state		JK flip flop Input			
A	B	X	A	B	JA	KA	JB	KB
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Simplify ~~the~~ state reduction

JA =

A \ BX	00	01	11	10
0				1
1	X	X	X	X

$JA = B\bar{X}$

KA =

A \ BX	00	01	11	10
0	X	X	X	X
1			1	

$KA = BX$

JB =

A \ BX	00	01	11	10
0		1	X	X
1		1	X	

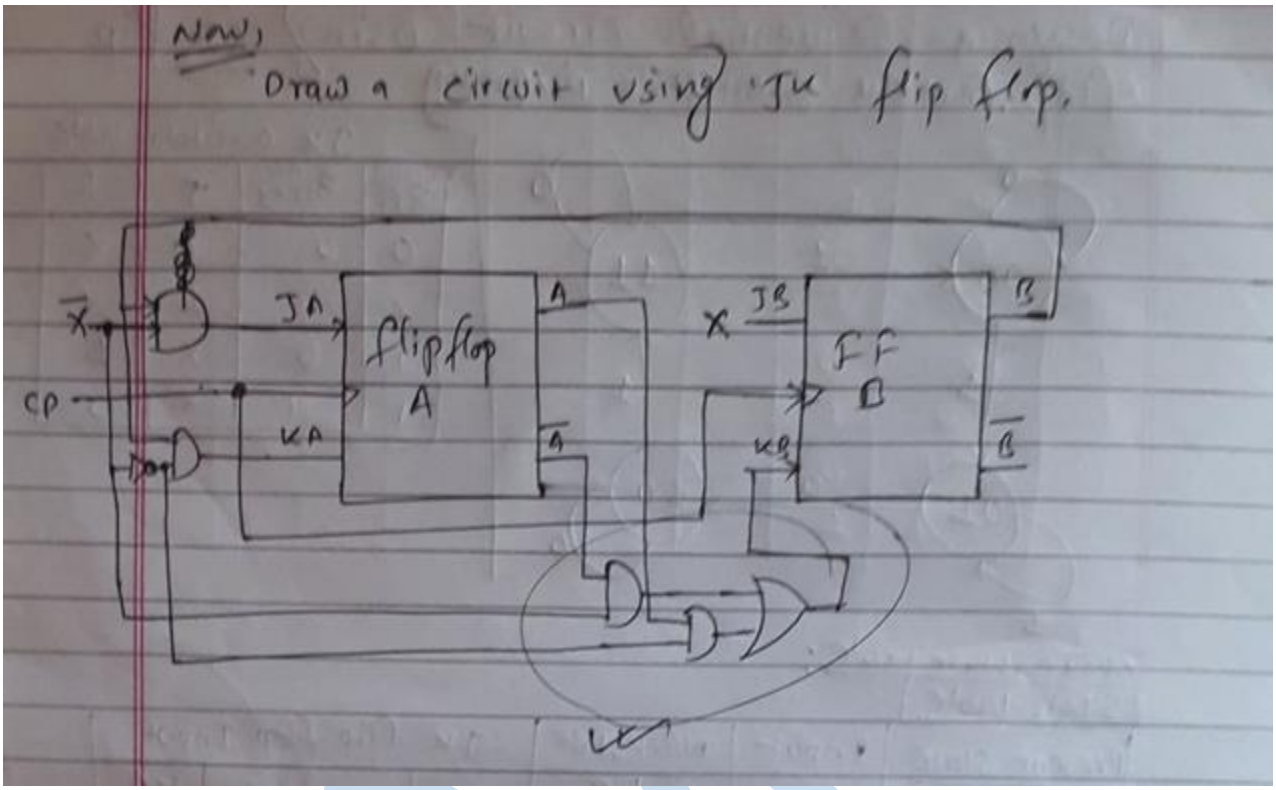
$JB = X$

KB =

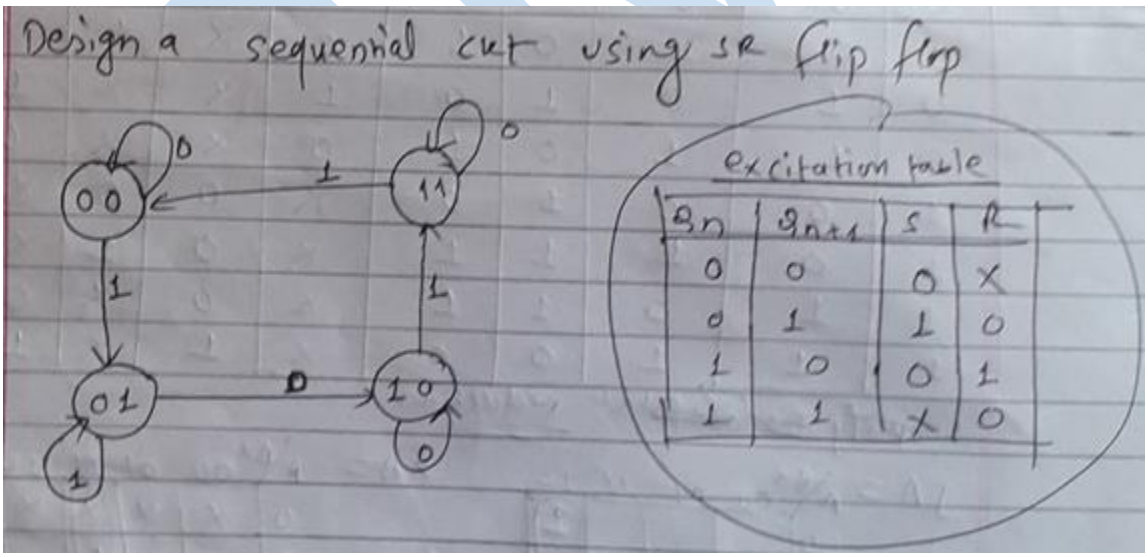
A \ BX	00	01	11	10
0	X	X		1
1	X	X	1	

$KB = \bar{A}X + AX$

$= A \odot X$



Q.N. Design a sequential circuit using SR flip flop from the given state diagram



Present state		Input	Next state		flip flop Inputs			
A	B	x	A	B	SA	RA	SB	RB
0	0	0	0	0	0	x	0	x
0	0	1	0	1	0	x	1	0
0	1	0	1	0	1	0	0	1
0	1	1	0	1	0	x	x	0
1	0	0	1	0	x	0	0	x
1	0	1	1	1	x	0	1	0
1	1	0	1	1	x	0	x	0
1	1	1	0	0	0	1	0	1

SA =

A \ Bx	00	01	11	10
0				1
1	x	x		x

$SA = B\bar{x}$

RA =

A \ Bx	00	01	11	10
0	x	x	x	
1			1	

$RA = Bx$

SB =

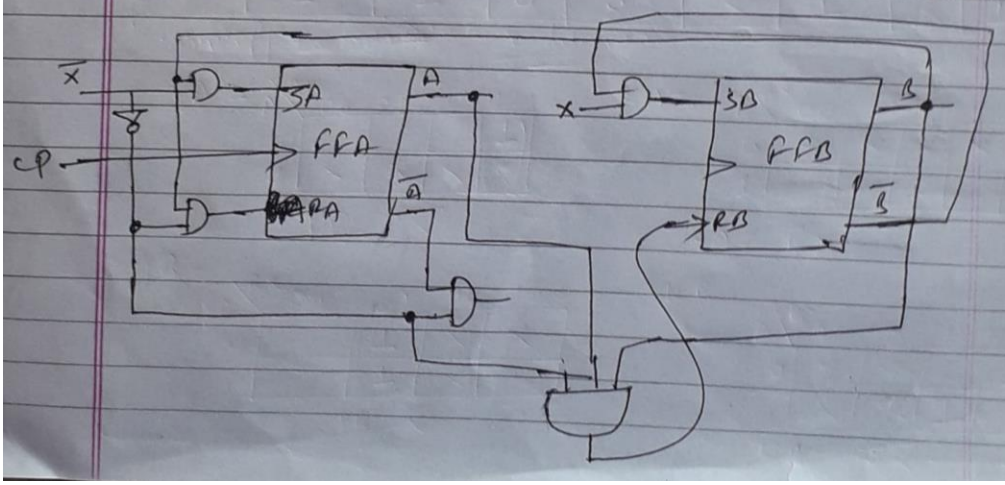
A \ Bx	00	01	11	10
0		1	x	
1		1		x

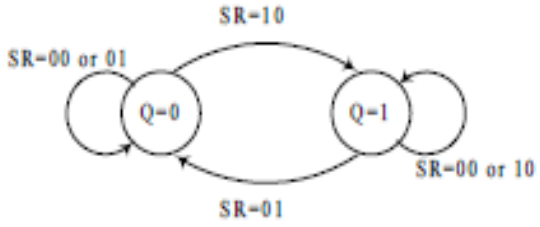
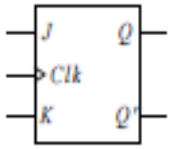
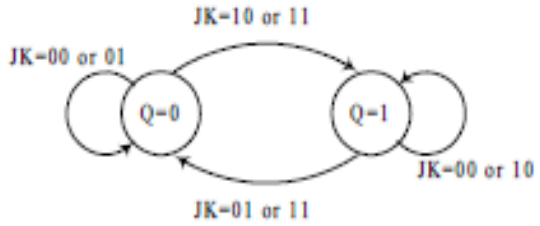
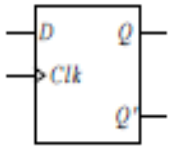
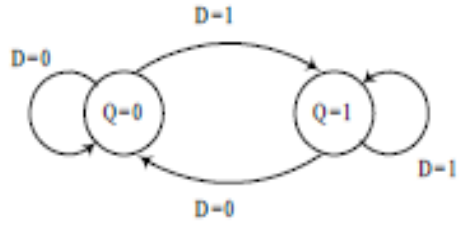
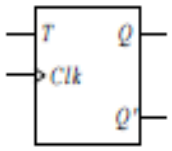
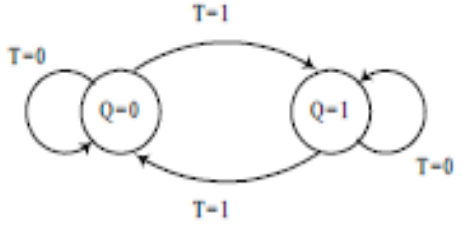
$SB = \bar{B}x$

RB

A \ Bx	00	01	11	10
0	x			1
1	x		1	

$RB = \bar{A}\bar{x} + ABx$



Name / Symbol	Characteristic (Truth) Table	State Diagram / Characteristic Equations	Excitation Table
SR 	S R Q Qnext 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 ×	 $Q_{next} = S + R'Q$ $SR = 0$	Q Qnext S R 0 0 0 × 0 1 1 0 1 0 0 1 1 1 × 0
JK 	J K Q Qnext 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	 $Q_{next} = J'K'Q + JK' + JKQ'$ $= J'K'Q + JK'Q + JK'Q' + JKQ'$ $= K'Q(J'+J) + JQ'(K'+K)$ $= K'Q + JQ'$	Q Qnext J K 0 0 0 × 0 1 1 × 1 0 × 1 1 1 × 0
D 	D Q Qnext 0 × 0 1 × 1	 $Q_{next} = D$	Q Qnext D 0 0 0 0 1 1 1 0 0 1 1 1
T 	T Q Qnext 0 0 0 0 1 1 1 0 1 1 1 0	 $Q_{next} = TQ' + T'Q = T \oplus Q$	Q Qnext T 0 0 0 0 1 1 1 0 1 1 1 0

- End of Unit 5 -