

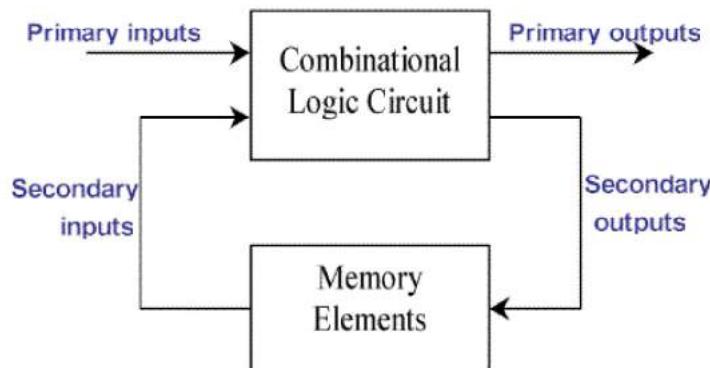
## MODULE-IV

### SEQUENTIAL LOGIC

Latches, Flip-Flops, Analysis of Clocked Sequential Circuits, Design of Synchronous Sequential Circuits, Registers, Shift Registers, Ripple Counters, Synchronous Counters, Ring Counter and Johnson Counter.

**Classification of sequential circuits:** Sequential circuits may be classed as two categories.

1. Synchronous sequential circuits
2. Asynchronous sequential circuits



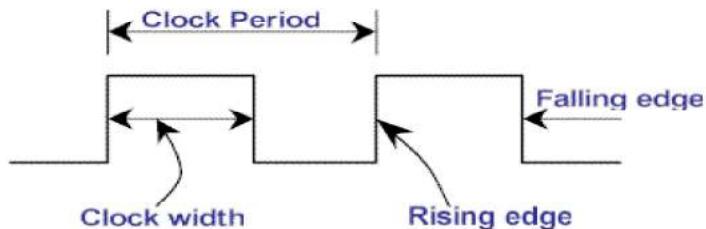
Combinational logic circuits' output depends solely on input current. Combinational logic circuits have no memory, therefore changing inputs erases their previous information. Almost every digital system has combinational circuits, but most have memory elements, which require sequential logic. Sequential logic circuits depend on both present and past input values. Sequential machines are mathematical models of sequential circuits.

**Comparison between combinational and sequential circuits**

<b>Combinational circuit</b>	<b>Sequential circuit</b>
1. In combinational circuits, the output variables at any instant of time are dependent only on the existing input variables	1. with sequential circuits the output variables at any moment of time are dependent not only on the present input variables, but also on the present state
2. memory unit is not necessary in combinational circuit	2. memory unit is necessary to retain the past history of the input variables
3. These circuits are faster because the delay between the i/p and o/p due to propagation delay of gates only	3. sequential circuits are slower than combinational circuits
4. easyto design	4.comparativelyhardtodesign

#### **Synchronous and Asynchronous Operation:**

Synchronous and asynchronous sequential circuits exist. Their classification depends on signal timing. The rising and falling edges of a free-running clock signal reflect synchronous sequential circuits' state and output values at discrete times. The clock signal is usually a square wave (Figure below).



The diagram shows that the clock period is the time between two rising or falling edges in the same direction. Synchronous sequential circuits create state transitions when the clock rises or falls. Memory data does not change between clock pulses.

The reciprocal of clock time is clock frequency. The clock width is the duration the clock signal is 1. Duty cycle is the ratio of clock width to period. State changes at the clock's rising

edge or width indicate an active high clock signal. The clock is active low else. They're also called timed sequential circuits.

Synchronous sequential circuits use flip-flops for memory. These circuits are one-bit binary cells. Flip-flop circuits have two outputs: the normal and complement values of the bit stored. Many types of flip-flops exist because binary information can enter them in different ways. The previous flip-flop tutorial covered the many fundamental flip-flop circuits and their logic.

Asynchronous sequential circuits shift from one state to another based on principal inputs, without external synchronization. Asynchronous sequential circuits use time-delayed memory, usually logic gate feedback. We can call asynchronous sequential circuits combinational circuits with feedback. Asynchronous sequential circuits can become unstable in transient settings due to logic gate feedback. Designers face many challenges due to volatility. Thus, they are less popular than synchronous systems.

**Fundamental Mode Circuits presume that:**

1. The input variables change only when the circuit is steady
2. Only one input variable can change at a given moment
3. Inputs are levels are not pulses

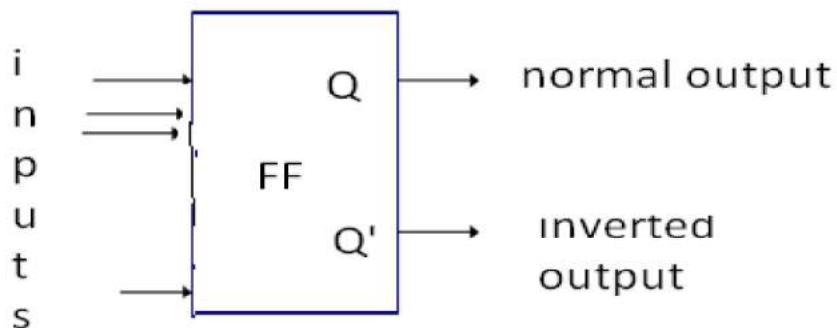
**A pulse mode circuit assumes that:**

1. The input variables are pulses instead of levels
2. The width of the pulses is long enough for the circuit to respond to the input
3. The pulse width must not be so long that it is still present after the new state is reached.

#### **4.1 Latches and flip-flops**

Information storage starts with latches and flip-flops. A latch or flip-flop can store one bit of data. As long as the enable signal is asserted, latches' outputs are always affected by their inputs, unlike flip-flops. Thus, when enabled, their content changes quickly when inputs change. Flip-flops only change content when the enable signal rises or falls. Enable is usually the controlling clock signal. Even when the input changes, the flip-flop content remains constant after the clock rises or falls.

Latches and flip-flops are usually SR, D, JK, or T. These flip-flops differ in input count and state change. Several adjustments improve each type's operations. Latch and flip-flop actions will be examined in this chapter. A flip-flop has two outputs, Q and Q'. Q is the flip flop's usual output and Q' its inverted output.



**Fig.** fundamental sign of flip-flop

A latch might be active-high or active-low. active -HIGH means the SET and RESET inputs are usually low and will be pulsed high to change latch outputs.

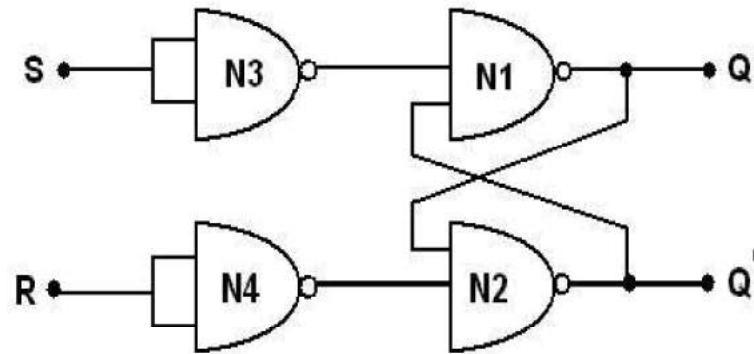
SR latch:

The latch outputs Q and Q'. When the circuit is on, the latch can do anything. Q=1 means Q'=0, SET state. Q=0 = Q'=1, RESET state. The latch will stay in SET or RESET mode as long as the power is on. The latch circuit is useless because there is no way to enter the necessary input. As detailed in the next sections, it is the main flip-flop building block.

Latch NAND

A flip-flop's main component is a NAND latch. It retains previous production if not disturbed.

NAND latch works oppositely to NOR latch. Changing 0s to 1s and 1s to 0s yields the same truth table as the NOR latch.



**NOR latch**

		S	R	Q	$\bar{Q}$	Function
0	0			$Q^+$	$\bar{Q}^+$	Storage State
0	1			0	1	Reset
1	0			1	0	Set
1	1			0-?	0-?	Indeterminate State

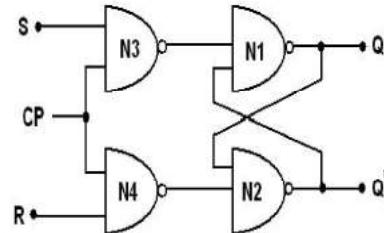
Active-HIGHNOR latch operation is examined below.

When SET=0 and RESET=0, the NOR latch rests normally and does not affect the output state.

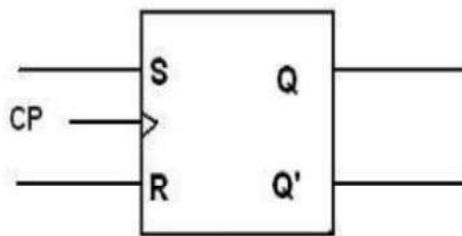
- 1.Q and  $\bar{Q}$  will remain as they were before this input condition.
2. Set  $Q=1$  and RESET=0 to maintain it even after SET returns to 0.
3. SET=0, RESET=1:  $Q=0$  remains reset even after RESET returns to 0.
4. Set=1,RESET=1: This condition simultaneously sets and resets the latch, resulting in  $Q=\bar{Q}=0$ . Returning all inputs to zero simultaneously produces a chaotic and unexpected outcome. Do not use this input

#### 4.2 RS Flip-flop:

Flip-flops are one-bit memory cells that form the basis of memory devices. It uses two NAND gates. Output of N1 is connected to input of N2, and output of N2 to input of N1. Feedback route inputs are S and R, outputs are Q and Q'. R-S flip-flop with timed input logic schematic and block design



a) Logic diagram



b) Block diagram

Figure: RS Flip-flop

Adding two NAND gates to the input latch makes the flip-flop only respond to clock pulses. Synchronization is achieved. Flip-flops can only change states at a certain time. A pulse generator generates clock pulses. The clock pulse alone affects the flip-flops.

Operation:

1. When CP=0, N3 and N4 output 1 regardless of S and R values. This feeds N1 and N2. This leaves Q and Q' unaltered.
2. When CP=1, S and R inputs can reach the latch, causing a flip-flop state shift.
3. CP=1, S=1, R=0 results in SET state ( $Q=1, Q'=0$ ).
4. CP=1, S=0, R=1 results in RESET state ( $Q=0, Q'=1$ ).
5. CP=1, S=0, R=0 does not affect flip-flop state.

6. CP=1, S=1, R=1 is not allowed as it cannot decide the next state. It is considered that this is a race-related disorder.

### Truth table

S	R	$Q_{(t+1)}$	Comments
0	0	$Q_t$	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	*	Not allowed

Triangles indicate CP input in logic symbols. When the input changes from 0 to 1, the circuit replies. The flip-flop characteristic table gives operation conditions.  $Q(t)$  represents the flip-flop's current state at time t. The state after a clock pulse is  $Q(t+1)$ .

### Edge triggered RS flip-flop:

RC circuits are sometimes used at flip-flop inputs near the clock pulse. By circuit design, the R-C time constant is much less than the clock pulse width. Therefore, output changes will only occur at specified clock pulse levels. A clock pulse from low to high fully charges the capacitor. A narrow positive spike results from this transition. Later, the trailing edge has a small negative spike. This is called edge triggering because the flip-flop only responds to clock pulse changes. Output transition at rising is called favorable edge triggering. Negative edge triggering happens at the trailing edge. Figure shows logic and block diagram..

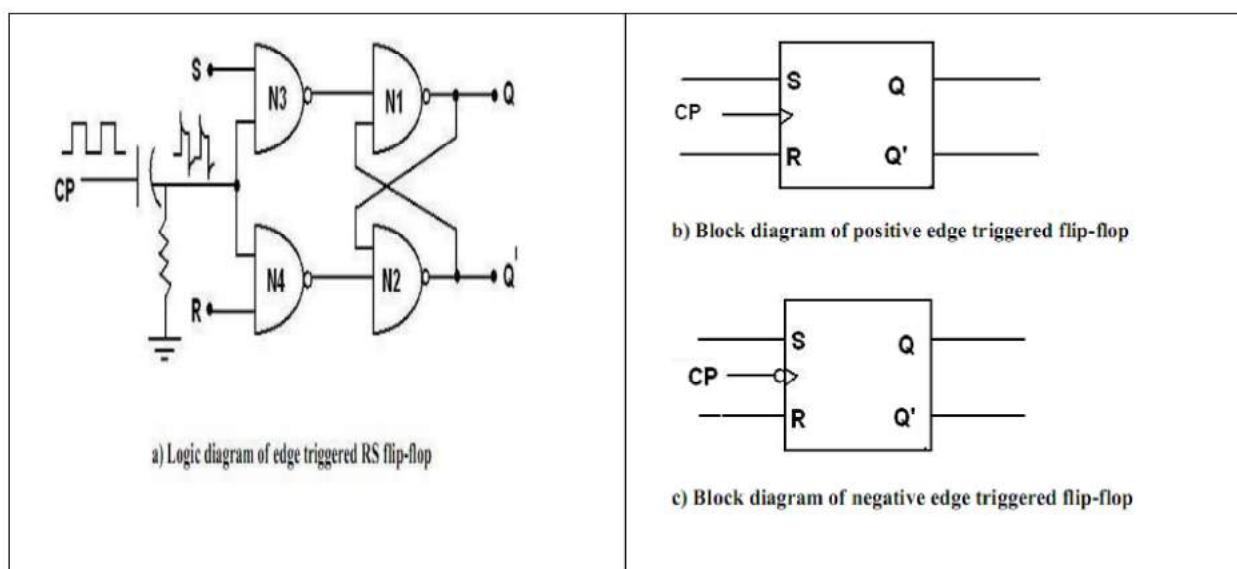
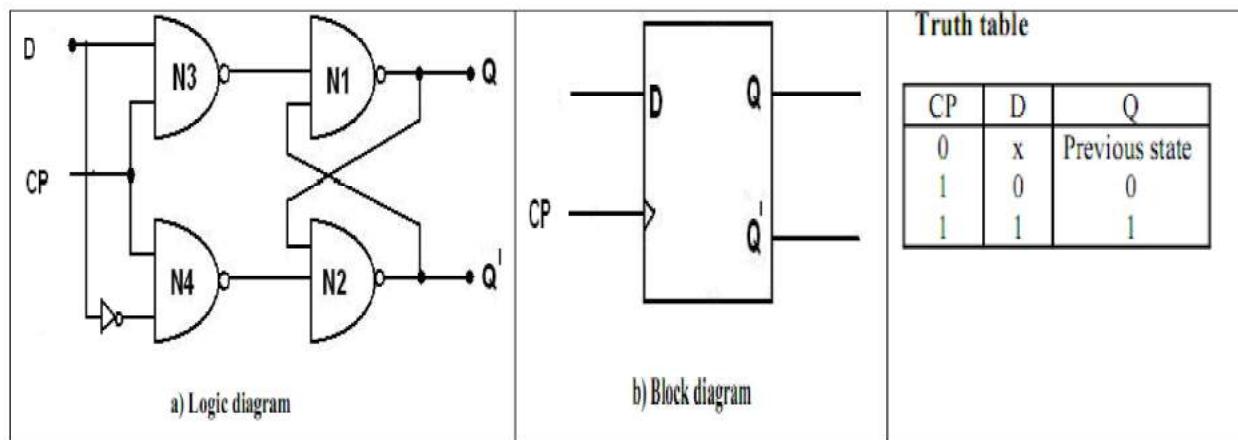


Figure: Edge triggered RS flip-flop

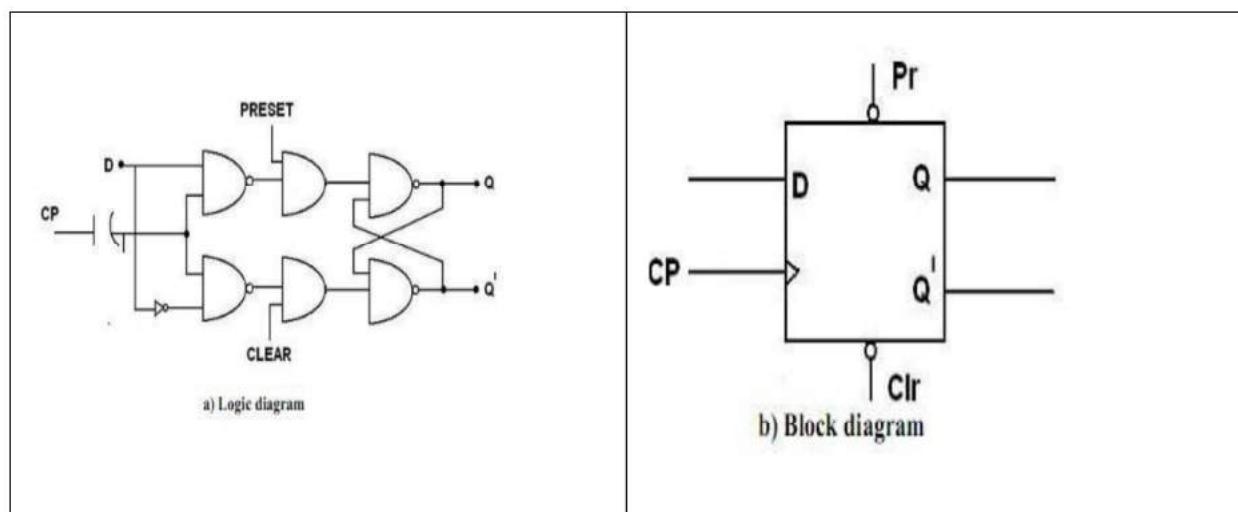
### D flip-flop:

R-S flip-flops are changed into D flip-flops. Inserting an inverter between S and R turns R-S flip-flop into D flip-flop, which takes only one input D. D is one input and its complement is another. D flip-flop with timed input logic and block diagrams



Modified R-S flip-flops are D flip-flops. An inverter between S and R converts R-S flip-flop to D flip-flop, which takes only one input D. D and its complement are inputs. The D flip-flop with timed input logic and block diagrams

### Edge Triggered D Flip-flop:

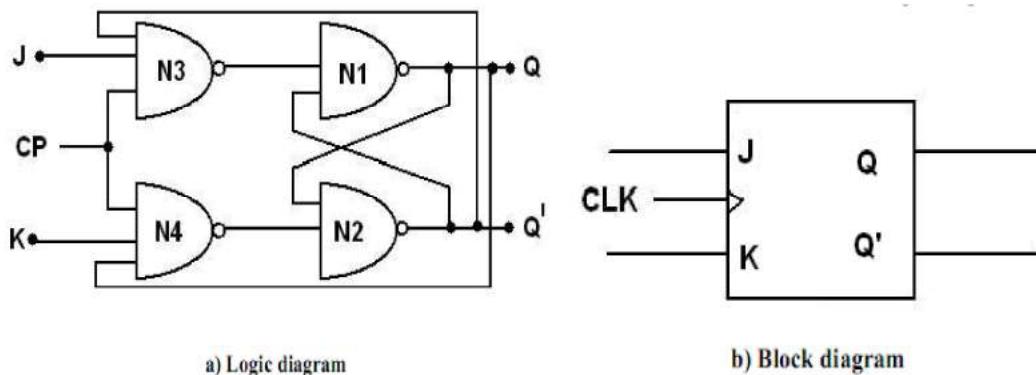


### Truth table

PRESET	CLEAR	CP	D	Q
0	0	X	X	*(forbidden)
0	1	X	X	1
1	0	X	X	0
1	0	0	X	NC
1	1	1	X	NC
1	1	↓	X	NC
1	1	↑	0	0
1	1	↑	1	1

**Figure:** truth table, block diagram, logic diagram of edge triggered flip-flop JK flip-flop (edge triggered JK flip-flop)

The race condition in RS flip-flop, when R=S=1 is eliminated in J-K flip-flop. There is a feedback from the output to the inputs. Figure depicts one approach of making a JK flip-flop.



### Truth table

J	K	$Q_{(t+1)}$	Comments
0	0	$Q_t$	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	$Q'_t$	Complement/toggle.

Figure: JK flip-flop

The J and K are called control inputs, because they determine what the flip-flop does when a positive clock edge arrives.

### Operation:

1. If J=0 and K=0, N3 and N4 will produce high output, maintaining the previous values of Q and Q'.
2. N3 outputs 1 when J=0, K=1, while N4 outputs based on Q value. Final output is Q=0, Q'=1. Reset state
3. When J=1, K=0, N4 outputs 1 and N3 depends on Q'. Final conclusion is Q=1 and Q'=0. Set state
4. J=1, K=1 allows flip-flop setting or reset based on output status. A reset condition is achieved when N4 transfers "0" to N2, resulting in Q=1, Q=0. Q becomes the complement of the last state when J=1, K=1. Flip-flops are toggled.

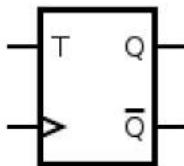
The characteristic equation of the JK flip-flop is:

$$Q_{next} = J\bar{Q} + \bar{K}Q$$

<b>JKflip-flop operation</b>									
<u>Characteristic table</u>				<u>Excitation table</u>					
<b>J</b>	<b>K</b>	<b>Q<sub>next</sub></b>	<b>Comment</b>	<b>Q</b>	<b>Q<sub>next</sub></b>	<b>J</b>	<b>K</b>	<b>Comment</b>	
0	0	Q	holdstate	0	0	0	X	Nochange	
0	1	0	reset	0	1	1	X	Set	
1	0	1	set	1	0	X	1	Reset	
1	1	Q	toggle	1	1	X	0	Nochange	

### T flip-flop:

If the T input is high, the T flip-flop changes state ("toggles") everytime the clock input is strobed. If the T input is low, the flip-flop holds the previous value. This behavior is characterized by the characteristic equation



The figure provided depicts the symbol representing a T flip flop.

This response aims to provide a more academic perspective on the topic of expanding the XOR operator. The XOR operator, also known as the exclusive disjunction operator, is a logical operator that returns true if and only if

$$Q_{next} = T \oplus Q = T\bar{Q} + \bar{T}Q$$

When the T input of a toggle flip-flop is set to a high logic level, it results in a division of the clock frequency by two. For instance, if the clock frequency is 4 MHz, the output frequency obtained from the flip-flop will be 2 MHz. This capability to divide the frequency finds utility in different digital counter applications. Additionally, a T flip-flop can be constructed using either a JK flip-flop, where the J and K pins are connected together and function as the T input, or a D flip-flop, where the T input is connected to the D input via an XOR gate.

<b>Tflip-flopoperation</b>							
<b><u>Characteristictable</u></b>				<b><u>Excitationtable</u></b>			
<b><i>T</i></b>	<b><i>Q</i></b>	<b><i>Q<sub>next</sub></i></b>	<b><i>Comment</i></b>	<b><i>Q</i></b>	<b><i>Q<sub>next</sub></i></b>	<b><i>T</i></b>	<b><i>Comment</i></b>
0	0	0	holdstate(n o clk)	0	0	0	Nochange
0	1	1	holdstate(n o clk)	1	1	0	Nochange
1	0	1	toggle	0	1	1	Complement

1	1	0	toggle	1	0	1	Complement
---	---	---	--------	---	---	---	------------

### The operating properties of flip flops.

The operational characteristics encompass the performance metrics, operational prerequisites, and operational constraints of the circuits. The operational parameters described above are applicable to all flip-flops, irrespective of the specific configuration of the circuit.

**Propagation delay** time refers to the duration of time that is necessary for an output change to take place after the application of an input signal.

**The set-up time** refers to the minimal duration during which the logic levels on the inputs (J and K, or S and R, or D) must remain constant before the triggering edge of the clock pulse. This duration ensures that the levels may be accurately captured by the flip-flop.

**The hold time** refers to the least duration during which the logic levels on the inputs must be maintained following the triggering edge of the clock pulse, ensuring that the levels are accurately transferred into the flip-flop.

**The maximum clock frequency** refers to the uppermost frequency at which a flip-flop can be consistently and accurately triggered. Power dissipation refers to the aggregate power consumed by a given device. The value is equivalent to the multiplication of the supply voltage ( $V_{cc}$ ) and the current ( $I_{cc}$ ).

The equation  $P = V_{cc} * I_{cc}$  represents the relationship between power (P), supply

Typically, the power dissipation of a flip flop is expressed in milliwatts (mW).

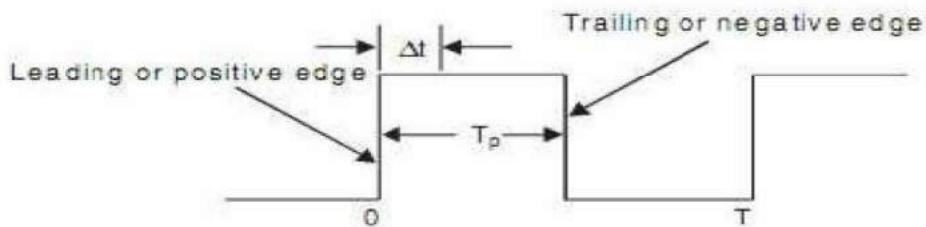
**Pulse widths** refer to the minimum durations of pulses as set by the manufacturer for the Clock, SET, and CLEAR inputs.

**Clock transition times:** In order to provide accurate triggering, it is imperative to minimize the duration of clock waveform transition times. If the clock signal exhibits a prolonged

transition time between levels, it might result in unpredictable triggering or failure to trigger the flip flop.

The phenomenon known as a "**race condition**"

The challenge associated with an S-R flip-flop, namely when both S and R are set to 1, can be resolved by implementing feedback connections from the outputs to the inputs of gate 1 and gate 2, as seen in the provided Figure. The truth tables depicted in the picture were constructed under the premise that the inputs remain constant during the duration of the clock pulse ( $CLK = 1$ ). However, the validity of this claim is questionable due to the presence of feedback linkages.



- For instance, let us consider the scenario where the values of J and K are both set to 1, and the value of Q is also set to 1. In this case, a pulse, as seen in the accompanying Figure, is applied to the clock input.
- Upon the elapse of a time interval denoted as  $t$ , which is equivalent to the propagation delay experienced by two NAND gates arranged in series, the outputs will undergo a transition, resulting in the value of Q being equal to 0. Currently, the values of J and K are both equal to 1, whereas the value of Q is 0.
- After a subsequent time period of  $t$ , the output will revert to  $Q = 1$ . Therefore, it can be inferred that during the time interval  $t_P$  of the clock pulse, the output will alternate between the binary states of 0 and 1. Consequently, with the termination of the clock pulse, the precise value of the output remains indeterminate. The present scenario is commonly denoted as a race-around condition.
- In general, the propagation delay of TTL gates is typically on the order of nanoseconds. If the duration of the clock pulse is on the scale of microseconds, it

follows that the output will undergo several changes, potentially reaching thousands of transitions inside a single clock pulse.

- The occurrence of a race-around scenario can be prevented by ensuring that the value of  $t$  falls within the range of  $t_p < t < T$ . The fulfillment of the aforementioned criteria may provide challenges due to the limited propagation delay exhibited by integrated circuits (ICs).
- A more pragmatic approach to mitigating the issue involves implementing the master-slave (M-S) setup, as elaborated upon in the subsequent discussion.

### **The utilization of flip-flops in various domains and industries.**

**Frequency Division:** In the context of digital circuits, frequency division refers to the process of generating an output signal with a lower frequency compared to the input signal. This can be achieved by applying a pulse waveform to the clock input of a J-K flip-flop that is configured to toggle. As a result, the Q output of the flip-flop produces a square wave with a frequency that is exactly half of the input clock frequency. By connecting additional flip-flops in the manner depicted in the figure provided, it is possible to achieve a greater degree of clock frequency division.

**Parallel data storage:** I'm sorry, but I need more information or text from you in order to provide an answer. In the context of data storage, a collection of flip-flops is commonly referred to as a register. In order to effectively store data consisting of  $N$  bits, it is necessary to employ  $N$  flip-flops. Given that the data is accessible in parallel form. When a clock pulse is simultaneously applied to all flip-flops, the bits will be transferred to the Q outputs of the flip flops.

**Serial data storage:** In order to store serial data consisting of  $N$  bits, a cascade connection of  $N$  D-flip-flops is utilized for data storage purposes. The clock signal is universally attached to each of the flip-flops. The serial data is inputted to the D terminal of the initial flip-flop.

**Transfer of data:** The transfer of data involves the movement of information contained in flip-flops. This transfer can occur in two ways: serially, where the data is sent bit-by-bit from the output of one flip-flop, or in parallel, where the data is transferred as a whole.

Previous State -> Present State		D
0 -> 0		0
0 -> 1		1
1 -> 0		0
1 -> 1		1

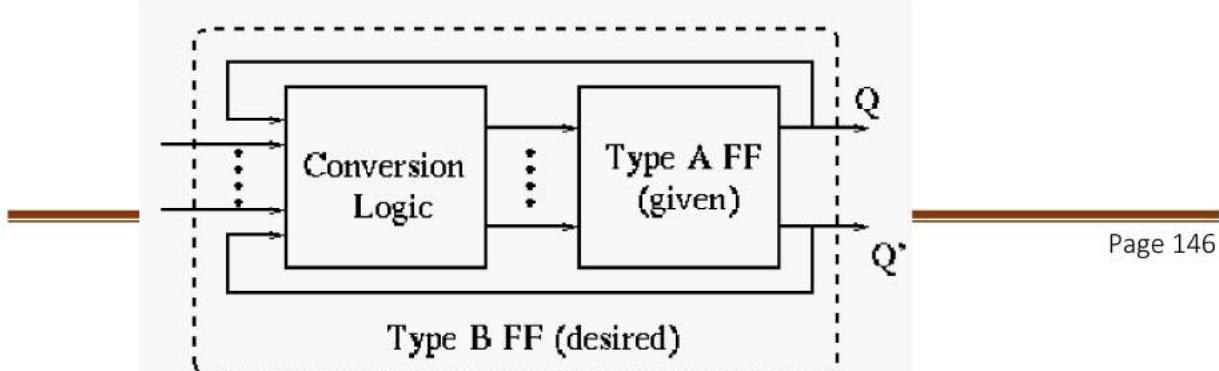
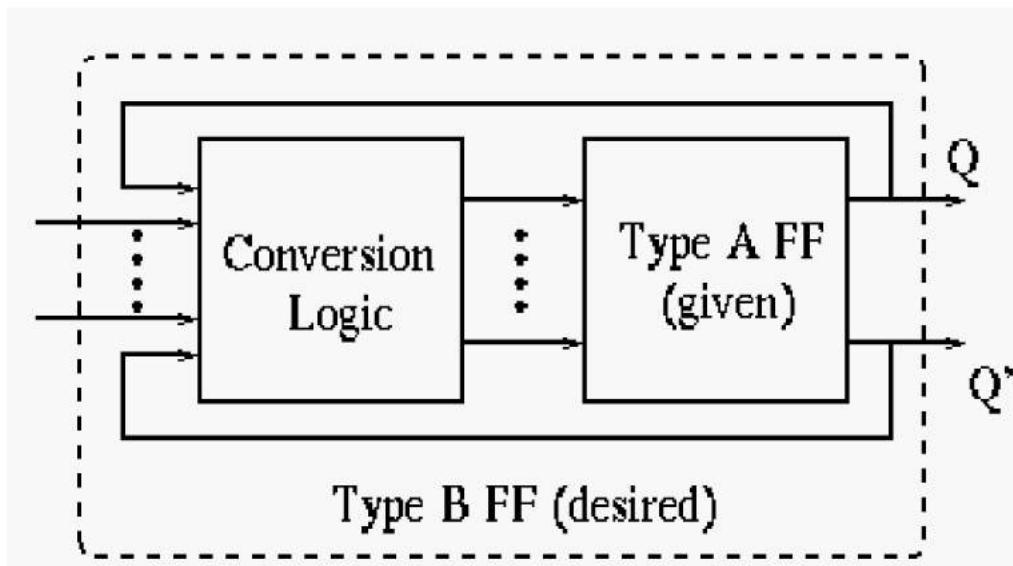
Previous State -> Present State		J	K
0 -> 0		0	X
0 -> 1		1	X
1 -> 0		X	1
1 -> 1		X	0

Previous State -> Present State		S	R
0 -> 0		0	X
0 -> 1		1	0
1 -> 0		0	1
1 -> 1		X	0

Previous State -> Present State		T
0 -> 0		0
0 -> 1		1
1 -> 0		1
1 -> 1		0

The topic of discussion pertains to **excitation tables**.

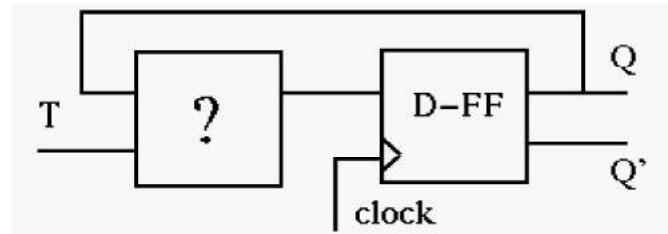
The topic of discussion pertains to the **conversions of flip-flops**.



The primary focus lies in utilizing the excitation table, which presents the essential triggering signals (S, R, J, K, D, and T) required for achieving a desired change in the state of a flip-flop.

$Q_t$	$Q_{t+1}$	S	R	J	K	D	T
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

To convert a D flip-flop (D-FF) to a T flip-flop (T-FF), a modification needs to be made to the input of the D-FF. Specifically, the D input of the D-FF should be replaced with the T input of the T

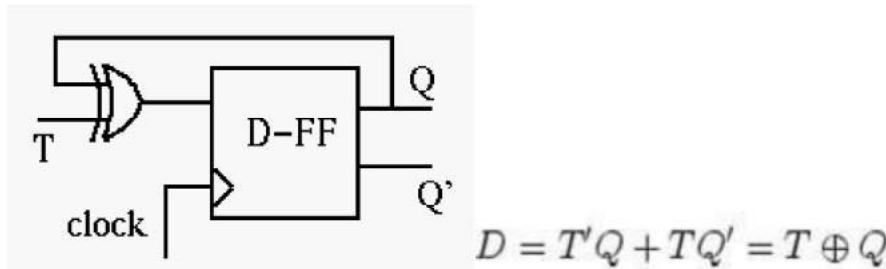


The circuit design for generating the triggering signal D is required, where D is a function of T and Q.

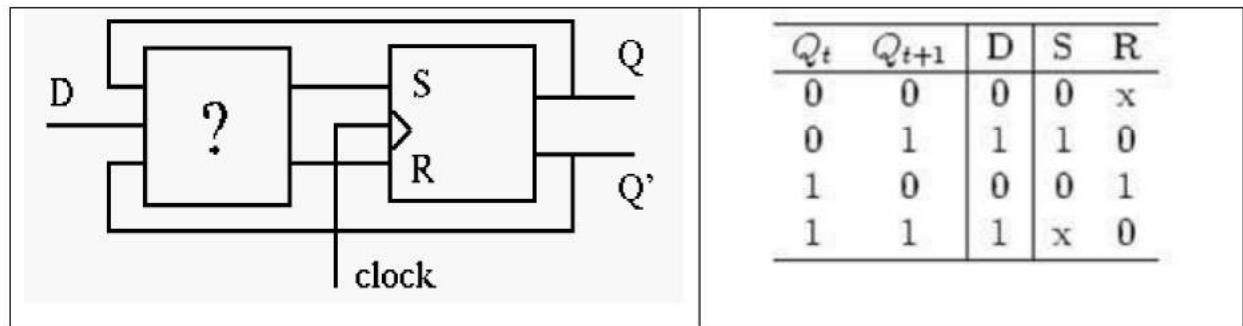
I'm sorry, but I need more context or information in order to provide an academic rewrite  
The excitation table is to be considered.

$Q_t$	$Q_{t+1}$	T	D
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

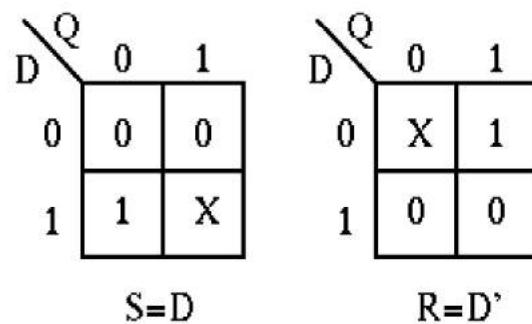
When considering it as a function of both the variable and the current FF state, we can observe.



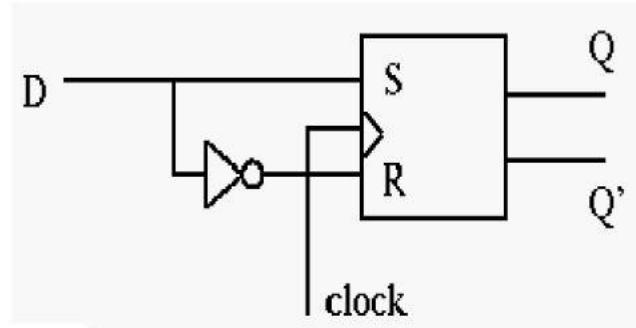
To convert an **RS-FF (Reset-Set Flip-Flop)** to a **D-FF (Data Flip-Flop)**, the following steps can be followed.



The task at hand necessitates the development of a circuit that can create the triggering signals S and R based on certain functions and in accordance with the provided excitation table.

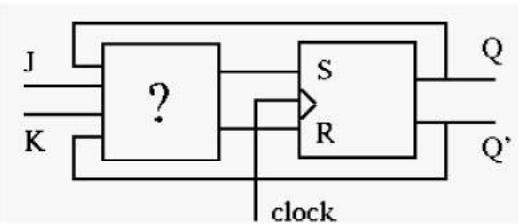


The intended signal can be derived as a function of the present FF state and input variables by utilizing the Karnaugh maps.



$$S = D, \quad R = D'$$

To convert an **RS-FF (Reset-Set Flip-Flop)** to a **JK-FF (Jack-Kilby Flip-Flop)**, the following steps can be followed.



It is necessary to devise a circuit that can generate the triggering signals S and R, which are dependent on the variables J and K.

$Q_t$	$Q_{t+1}$	J	K	S	R
0	0	0	x	0	x
0	1	1	x	1	0
1	0	x	1	0	1
1	1	x	0	x	0

The excitation table can be analyzed by examining the intended signal and its dependence on the present flip-flop state, which can be derived from the Karnaugh maps.

Karnaugh maps, commonly referred to as K-maps, are a graphical tool used

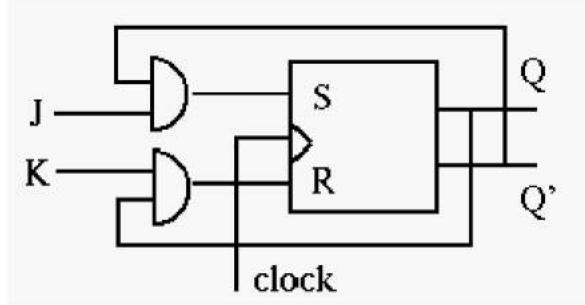
K-maps:

	QJ	00	01	11	10
K	0	0	1	X	X
	1	0	1	0	0

$$S = Q'J$$

	QJ	00	01	11	10
K	0	X	0	0	0
	1	X	0	1	1

$$R = QK$$



$$S = Q'J, \quad R = QK$$

The **Master-Slave JK Flip-flop** is a type of sequential logic circuit that is commonly used in digital electronics.

The Master-Slave Flip-Flop consists of two gated SR flip-flops interconnected in a series arrangement, where the slave flip-flop receives an inverted clock pulse. The feedback loop of the system involves connecting the outputs from the "Slave" flip-flop, denoted as Q and Q̄, to the inputs of the "Master" flip-flop. In turn, the outputs of the "Master" flip-flop are connected to the two inputs of the "Slave" flip-flop. The feedback setup, which involves connecting the output of the slave to the input of the master, results in the distinctive toggling behavior of the JK flip-flop, as depicted in the diagram below.

The input signals J and K are linked to the gated "master" SR flip-flop, which retains the input state when the clock (Clk) input is at a logic level of "1". Due to the inverse relationship

between the clock inputs of the "slave" and "master" flip-flops, the "slave" SR flip-flop does not exhibit toggling behavior. The outputs generated by the "master" flip-flop are exclusively observed by the gated "slave" flip-flop when the clock input transitions to a logic level of "0". When the clock signal is in a "LOW" state, the outputs of the "master" flip-flop are captured and all further modifications to its inputs are disregarded. The gated "slave" flip-flop exhibits a response to the condition of its inputs that is determined by the information carried across by the "master" section. During the transition of the clock pulse from low to high, the inputs of the master flip-flop are transmitted to the gated inputs of the slave flip-flop. Conversely, during the transition from high to low, the same inputs are mirrored on the output of the slave flip-flop. This characteristic renders this type of flip-flop as edge or pulse-triggered. Subsequently, the circuit receives input data during the period when the clock signal is in a "HIGH" state, and transmits the data to the output during the descending phase of the clock signal. The Master-Slave JK Flip-flop can be classified as a "Synchronous" device due to its characteristic of transmitting data exclusively in synchronization with the clock signal.

### 1.3 The present study focuses on the analysis and design of **clocked sequential circuits**.

The operational characteristics of a clocked sequential circuit are established based on the inputs, outputs, and the internal state of its flip-flops. Both the outputs and the next state are dependent on both the inputs and the current state. The examination of a sequential circuit involves the acquisition of a table or diagram that represents the chronological progression of inputs, outputs, and internal states.

A clocked sequential circuit is characterized by the presence of flip-flops with clock inputs within its logic layout. The flip-flops utilized in the system can vary in type, while the logic diagram may or may not incorporate combinational logic gates. Within this particular section, we give a method of expressing the condition for the subsequent state in algebraic terms, utilizing the current state and input variables. The behavior of the sequential circuit is thereafter described through the presentation of a state table and state diagram.

### An illustrative instance to traverse the procedural stages.

The given specification states that the circuit shall consist of a single input, denoted as X, and a single output, denoted as Z. The value of output Z will be 0 in all cases, except when the

input sequence 1101 is the most recent sequence of four inputs received on X. In such an instance, the resulting value would be equal to 1.

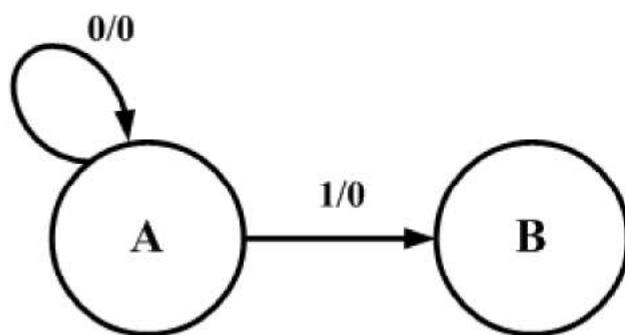
### The production of a state diagram

Developing states and ascribing significance to them.

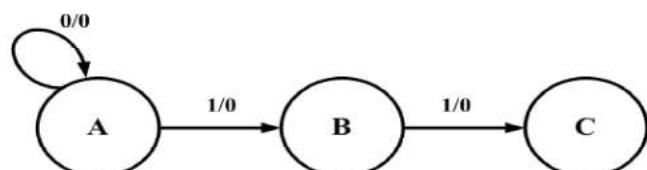
In State A, the most recent input was a zero, while the preceding inputs remain undisclosed.

The reset state can also be considered. In State B, the most recent input was a 1, but the preceding input was a 0. The commencement of a potential new series.

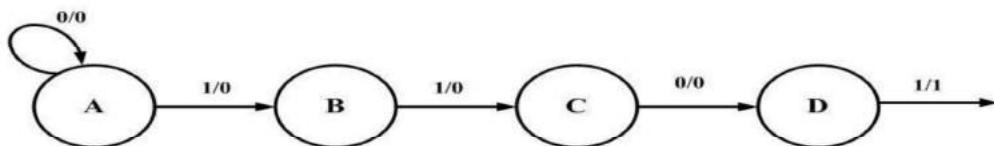
Please represent this concept using a state diagram.



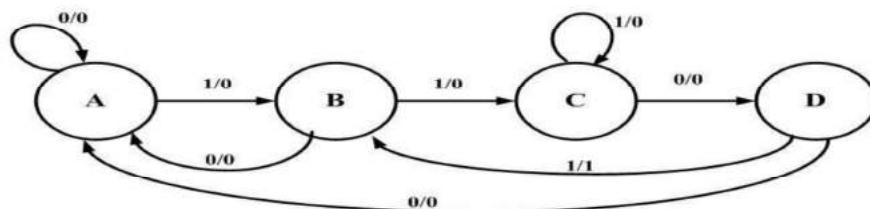
- Capture this in a state diagram
- Circles serve as symbolic representations of the many states.
- Lines and arcs are utilized to depict the transition that occurs between different states.
- The notation "Input/output" on the line or arc denotes the input that triggers this transition and the corresponding output associated with this change of state.
- Incorporate state C - The input sequence 11, which marks the initiation of the sequence, has been detected.



- Incorporate an additional state into the existing context.
- State D has successfully identified the third input at the beginning of a series, which is a zero.
- The number 110. If the subsequent input in State D is equal to 1, the sequence is recognized and a 1 is produced as the output.



- The diagram provided earlier was found to be lacking in its entirety.
- In any condition, the subsequent input has the potential to be either a 0 or a 1. It is imperative that this be incorporated.



- The state table is a tabular representation that outlines the various states and their corresponding characteristics or properties inside a given system or context.
- The task at hand can be accomplished in a straightforward manner by utilizing the state diagram.

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

- There is a current requirement to do a state assignment.

Please choose a state assignment.

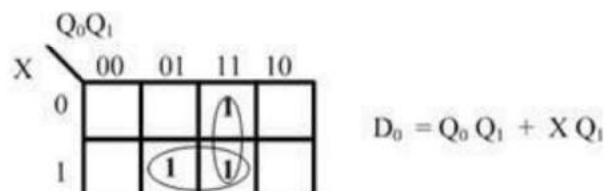
The decision will be made to employ a gray encoding.

In the given encoding scheme, state A is represented by the binary sequence 00, state B by 01, state C by 11, and state D by 10.

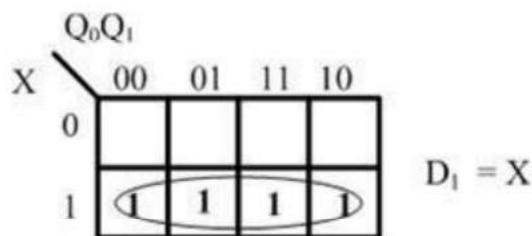
Present State	Next State		Output	
	X=0	X=1	X=0	X=1
00	00	01	0	0
01	00	11	0	0
11	10	11	0	0
10	00	01	0	1

The equations governing the input **behavior of a flip-flop**

- Derive the mathematical expressions for the inputs of a flip-flop.



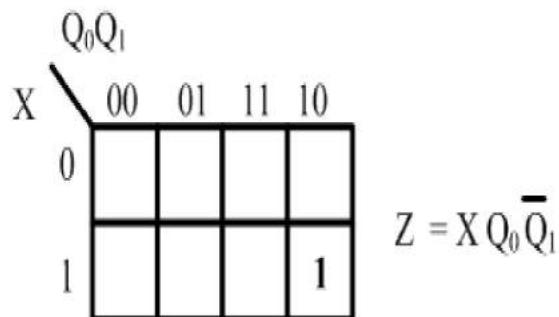
- Deriving the D0 equation
- Derive the D1 equation.



The equation representing the output

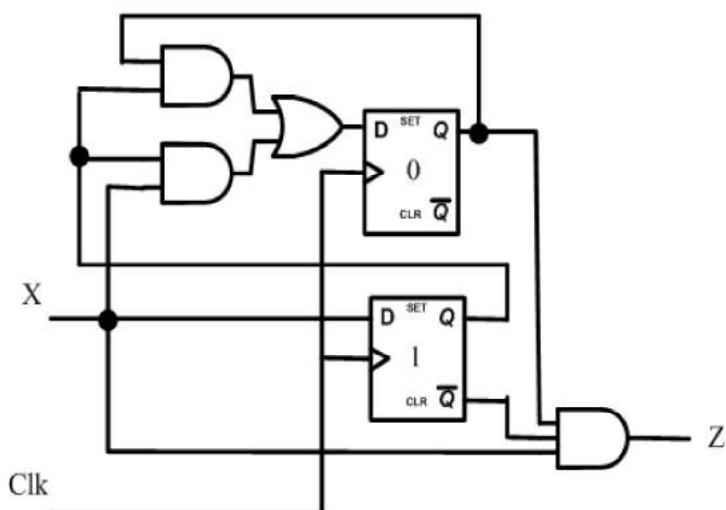
The subsequent stage involves formulating the equation for the output variable Z and identifying the necessary components required for its generation.

Generate a Karnaugh map based on the provided truth table.



Next, proceed with the task of creating a circuit diagram.

The circuit is equipped with two D-type flip-flops.



#### 1.4 The topic of discussion pertains to **registers**.

A register is a collection of flip-flops that are synchronized by a common clock signal and have the ability to store a single binary digit. A register with a capacity of n bits is comprised of a collection of n flip-flops that have the ability to retain n bits of binary data. In addition to the flip-flops, a register may incorporate combinational gates that execute certain data-processing functions. A register, in its most comprehensive interpretation, encompasses a

collection of flip-flops together with associated gates that exert influence on their functioning. The flip-flops are responsible for storing the binary information, while the gates play a crucial role in determining the mechanism via which the information is transmitted into the register.

A wide range of registers can be found in the commercial market. The most elementary form of a register is a configuration solely composed of flip-flops, devoid of any logical gates. The diagram presented below illustrates the construction of a register utilizing four D -type flip-flops, resulting in the formation of a four-bit data storage register.

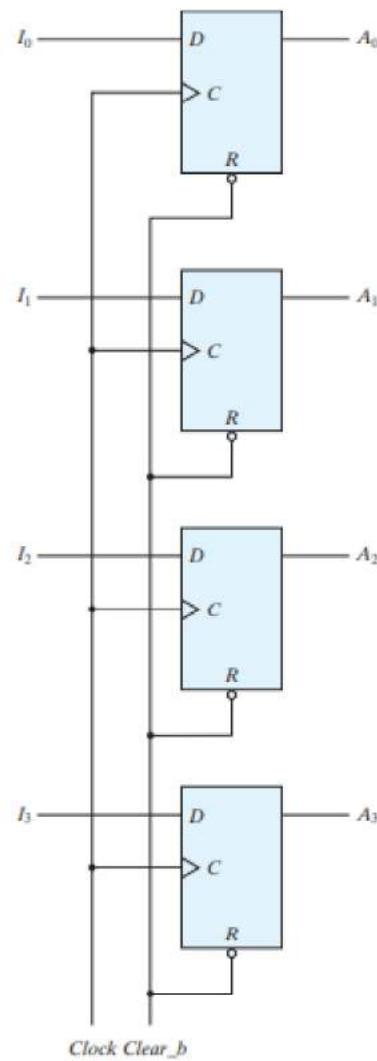


Figure. The topic of discussion is a four-bit register.

The flip-flops in the register are triggered by a shared clock input, namely on the positive edge of each pulse. At this moment, the binary data present at the four inputs is transferred into the register. The state of ( I<sub>3</sub> , I<sub>2</sub> , I<sub>1</sub> , I<sub>0</sub> ) at the moment just prior to the occurrence of the clock edge determines the state of ( A<sub>3</sub> , A<sub>2</sub> , A<sub>1</sub> , A<sub>0</sub> ) immediately following the clock edge. The binary information stored in the register can be obtained by sampling the four outputs at any given time. The input signal Clear\_b is connected to the active-low R (reset) input of each of the four flip-flops. When the input reaches a value of 0, the reset operation of all flip-flops occurs asynchronously. The Clear\_b input serves the purpose of resetting the register to a state where all bits are set to 0 before its timed action. In regular clocked operation, it is imperative to maintain the R inputs at logic 1, often known as de-asserted state. It should be noted that, depending on the type of flip-flop being used, either the Clear, Clear\_b, reset, or reset\_b signals can be employed to signify the transition of the register to a state where all bits are set to 0.

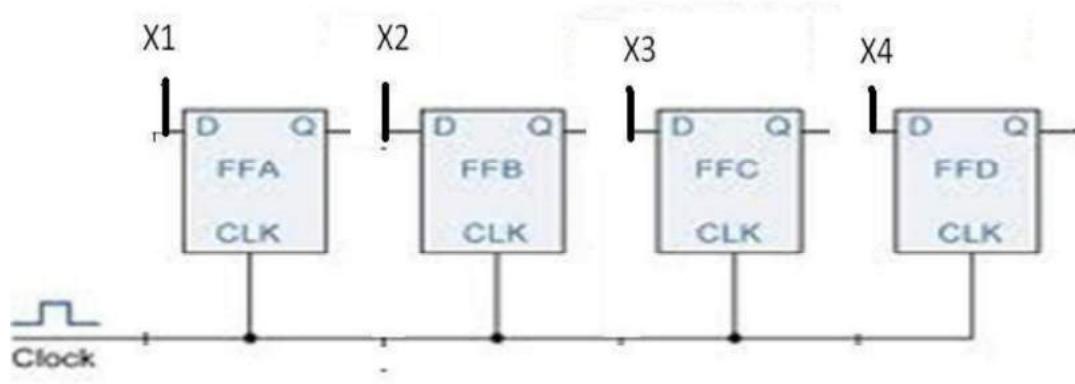
**4.5 Shift registers** are electronic devices that are utilized in digital circuits for the purpose of storing and transferring data.

Digital shift registers are interconnected flip-flops that operate on the same clock signal. Each flip-flop's output is linked to the "data" input of the next flip-flop. Thus, this arrangement allows the register's "bit array" to shift one position. The register shifts in the data at its input and out the final bit in the array during each clock input transition. A shift register can have several dimensions with bit array "data in" and stage outputs. Operate multiple shift registers with identical bit-lengths simultaneously.

Shift registers support parallel and serial inputs and outputs. Many of these setups are serial-in, parallel-out (SIPO) or parallel-in, serial-out. There are also devices with serial and parallel input and output capabilities. Additionally, bi-directional shift registers allow for shifting in both directions (L→R or R→L). Connecting a shift register's serial input and last output creates a circular shift register.

Shift registers are counter-like logic devices. These gadgets mostly store and transfer digital data. The buffer register is a component used in computer systems to temporarily store data during the transfer between different parts of the system.

The buffer register refers to a basic collection of registers. The purpose of this system is to save the binary representation of a word. The buffer in question has the potential to be a controlled buffer. The majority of buffer registers employ D Flip-flops.



The provided figure depicts a logic diagram illustrating a 4-bit buffer register.

The provided illustration depicts a buffer register with a capacity of four bits. The binary word is put to the data terminals for storage. When a clock pulse is applied, the output word is identical to the word that was applied at the terminals. The input word is loaded into the register through the application of a clock pulse.

Upon the arrival of the positive clock edge, the stored word is transformed into  $Q_4Q_3Q_2Q_1=X_4X_3X_2X_1$ .

The variable Q is equal to X.

A controlled buffer register is a type of register that is designed to store and control the flow of data within a computer system.

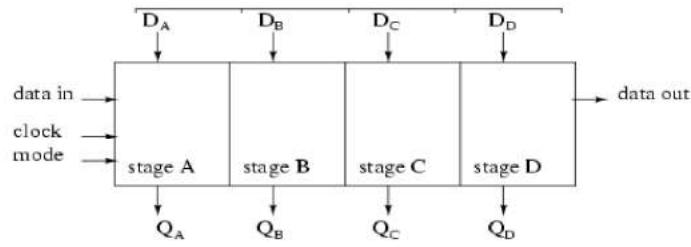
When the control signal CLRG is set to a low voltage level, it triggers the resetting of all flip-flops inside the system. As a result, the output of the system is set to a binary value of 0000, represented as  $Q=0000$ .

When the Control Line Register (CLR) is in a HIGH state, it indicates that the register is prepared and available for execution. The control input in this context is referred to as LOAD. When the LOAD signal is in a HIGH state, the data bits X are able to propagate to the D inputs of the flip-flops.

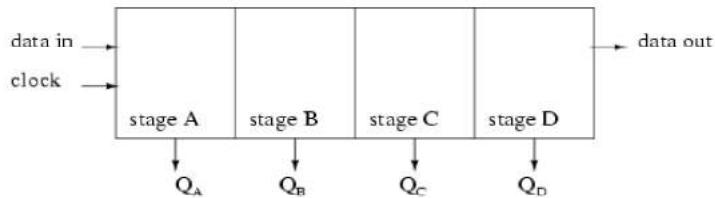
The equation  $Q_4Q_3Q_2Q_1 = X_4X_3X_2X_1$  can be simplified as  $Q = X$

When the load is at a low level, the X bits are unable to propagate to the flip-flops (FFs).

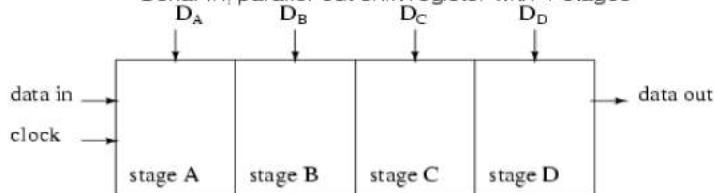
The topic of discussion pertains to the process of **data transmission within shift registers**.



Parallel-in, parallel-out shift register with 4-stages



Serial-in, parallel-out shift register with 4-stages

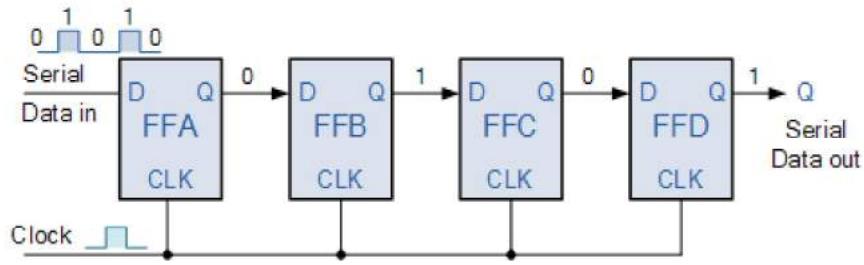


Parallel-in, serial-out shift register with 4-stages

A shift register refers to a series of flip-flops that are interconnected, enabling the transfer of data into and out of them by shifting operations. The transfer of data into or out of the register can occur either in serial form or in parallel form. There exist four fundamental categories of shift registers.

1. Serialin,serialout, shiftright,shiftregisters
2. Serialin,serialout,shiftleft,shiftregisters
3. Parallelin,serialoutshiftregisters
4. Parallelin,paralleloutshiftregisters

#### Serial IN, serial OUT, shift right, shift left register:



The concept of **serial in, serial out**, shift right, shift registers refers to a type of digital circuit that is commonly used in computer systems and other electronic devices. These shift registers are designed to store and manipulate binary data in a sequential manner. The "serial in" aspect refers to the input of data being received one bit at a time, while the "serial out"

The concept of **serial in, serial out**, shift left, shift registers refers to a type of digital circuit that is commonly used in many applications. These registers are designed to store and manipulate binary data in a sequential manner. The "serial in" aspect denotes the input of data in a serial fashion, meaning that the bits are fed into the register one at a time.

**Parallel-in, serial-out** shift registers are electronic devices that allow for the parallel loading of data and the serial shifting of that data out.

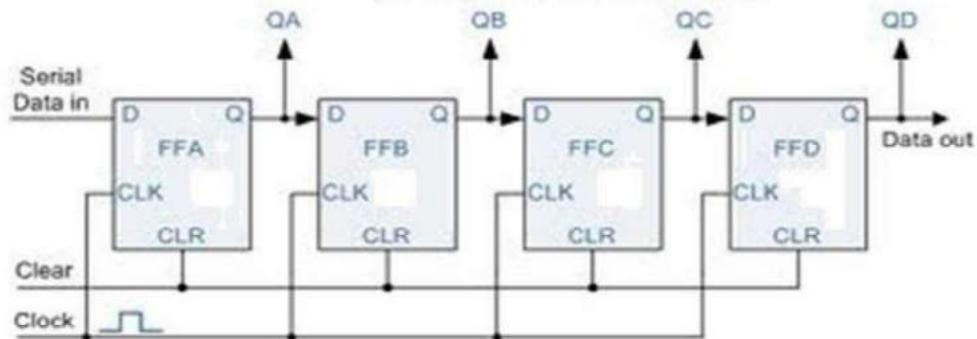
The concept of parallel in, parallel out shift registers refers to a type of digital circuit that allows for the simultaneous input and output of data in a parallel manner.

The **serial input, serial output**, shift right, and shift left registers are components commonly used in digital systems for data storage and manipulation.

The present document presents the logic diagram of a four-stage, four-bit serial-in, serial-out right shift register. The register has the capacity to hold a maximum of four bits of data. The input receives a stream of serial data.

The first fundamental frequency (FF) is denoted as D. The output Q of the first flip-flop is connected to the input D of another flip-flop. The data is generated as output from the Q terminal of the last flip-flop.

In the process of transferring serial data into a register, the introduction of each new bit occurs at the positive transition of each clock pulse, causing it to be clocked into the first flip-flop (FF). The data previously held in the first flip-flop is transferred to the second flip-flop. The data previously held in the Second Flip-Flop (FF) is moved to the third FF.

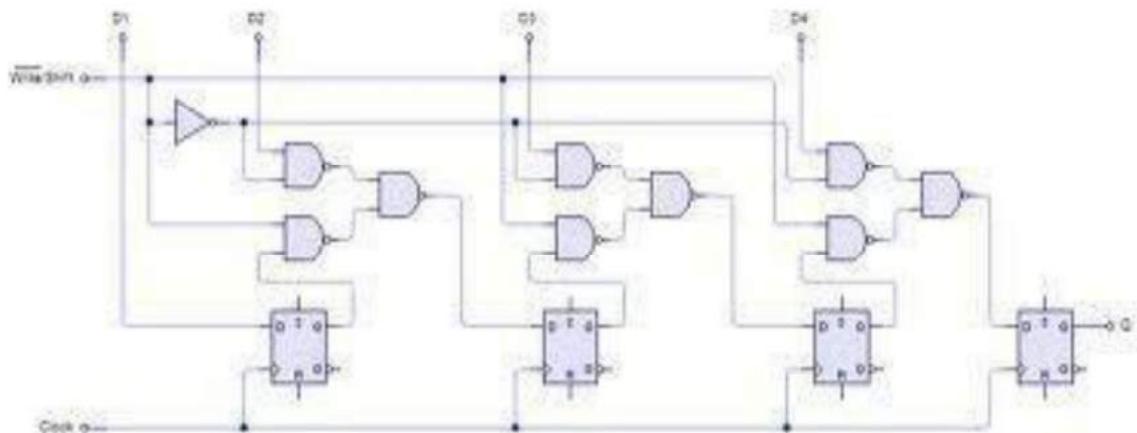


The shift register under consideration is a **serial-in, parallel-out** configuration.

In this particular register configuration, the data bits are sequentially inputted into the register. However, the data stored within the register is subsequently shifted out in a parallel manner.

After the storage of data bits, each bit is presented on its corresponding output line, allowing for simultaneous access to all bits instead of a sequential output of bits. The shift register with a serial-in, parallel-out configuration can also function as a serial-in, serial-out register by extracting the output from the Q terminal of the last flip-flop.

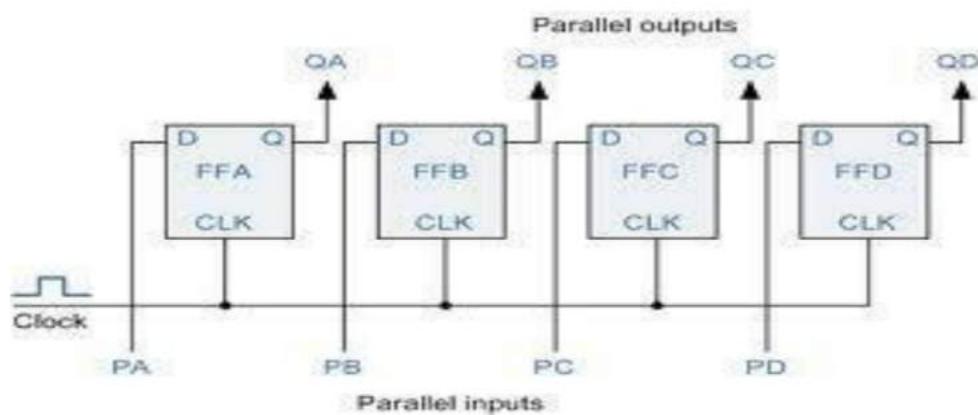
The **parallel-in, serial-out shift register** is a type of digital circuit that allows for the parallel input of data and the serial output of that data.



In the context of a parallel-in, serial-out shift register, the data bits are introduced concurrently into their respective stages through parallel lines, as opposed to the sequential transmission of serial data bits from the register. In a sequential manner, examining each individual unit within a singular line.

The data is fed into the register in parallel manner using four data lines, namely A, B, C, and D. The signal shift/load function facilitates the parallel input of data into the register, while enabling the serial output of data from terminal Q4.

The parallel-in, parallel-out shift register is a digital circuit that allows for the simultaneous loading and retrieval of several bits of data.



The parallel-in, parallel-out shift register operates by receiving data in parallel format and also transmitting data in parallel format. The data is inputted into the D terminals of the flip-flops. Upon the application of a clock pulse, namely at the positive transition of the pulse,

the D inputs are transferred and stored into the Q outputs of the flip-flops. The data is now stored within the register. The data that has been stored can be accessed immediately and can be transmitted in a parallel manner.

A **bidirectional shift register** is a digital circuit that is capable of shifting data in both directions, either to the right or to the left.

A bidirectional shift register is a type of register that allows for the shifting of data bits in both the left-to-right and right-to-left directions. The provided diagram illustrates the logical representation of a 4-bit serial-in, serial out, bidirectional shift register. The mode signal serves as an indicator for the right/left operation in the logic circuit, where a value of 1 signifies the shift-register functionality. To enable bidirectional operation, each stage of the circuit employs two NAND gates and one OR gate, in conjunction with the mode signal.

When the control input on the right or left is set to a HIGH state, it activates AND gates G1, G2, G3, and G4, while deactivating AND gates G5, G6, G7, and G8. As a result, the Q output state of each flip-flop is transmitted via the gate and applied to the D input of the subsequent flip-flop. When a clock pulse is triggered, the data bits undergo a process in which they are moved by one position towards the right, resulting in an effective shift. When a LOW signal is applied to the right/left control inputs, it activates AND gates G5, G6, G7, and G8, while deactivating AND gates G1, G2, G3, and G4. Additionally, the output Q of each flip-flop is connected to the input D of the preceding flip-flop. When a clock pulse is triggered, the data bits undergo a process in which they are shifted one position to the left, resulting in an effective shift. Therefore, the circuit functions as a bidirectional shift register.

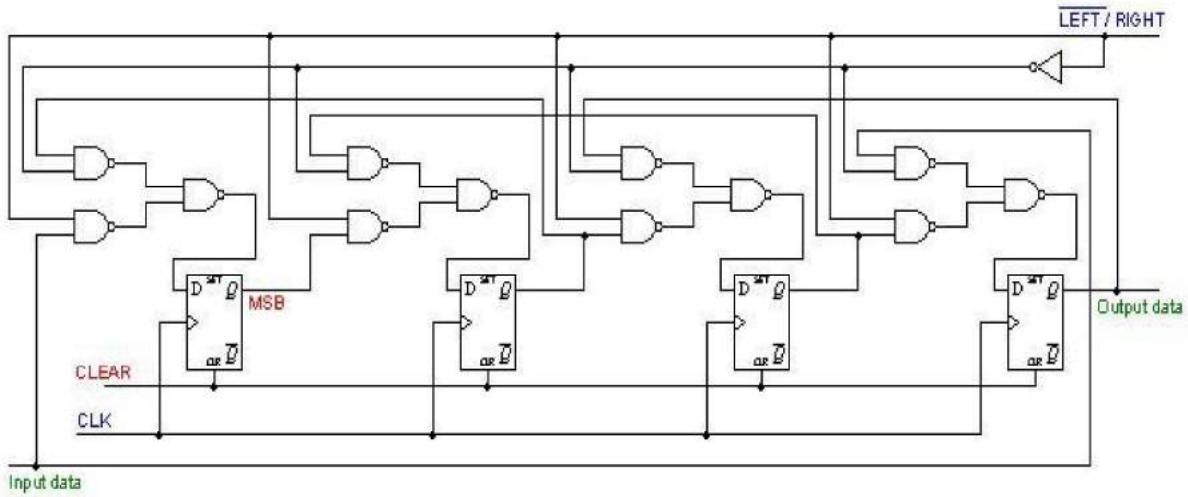


Figure. The following is a depiction of a logic diagram illustrating a 4-bit bidirectional shift register.

The **universal shift register** is a type of digital circuit that is capable of shifting data in both directions, either left or right. It is a fundamental component in digital systems and

A shift register that is capable of changing in only one way is referred to as a unidirectional shift register. A bidirectional shift register is capable of shifting in both directions. A register that possesses the ability to do both shifts and parallel load operations is commonly known as a universal shift register. The universal shift register is a versatile register that supports bidirectional data transfer. It can accept input in either serial or parallel format, and it can also produce output in either serial or parallel format.

The most comprehensive shift register possesses the subsequent functionalities.

1. A distinct mechanism for resetting the register to zero.
2. The provision of a clock input to facilitate the synchronization of operations.
3. The inclusion of a shift-right control facilitates the activation of the shift-right operation, while also providing serial input and output lines that are linked to the shift-right process.
4. The inclusion of a shift-left control facilitates the activation of the shift-left operation, while also providing serial input and output lines that are specifically linked to the shift-left process.

5. A control mechanism for parallel loads is implemented to facilitate parallel transfer, whereby the  $n$  input lines connected to the parallel transfer are utilized.
6. There are  $N$  parallel output lines.
7. A control state is a state in which the information stored in the register remains unaltered when the clock signal is present.

A universal shift register can be implemented by employing multiplexers. The figure presented below depicts the logical diagram of a 4-bit universal shift register that possesses a comprehensive range of capabilities. The system comprises of four D flip-flops and four multiplexers. The four multiplexers are equipped with two shared selection inputs, denoted as  $s_1$  and  $s_0$ . The selection of input 0 in each multiplexer occurs when the binary control inputs  $S1S0$  are set to 00. Similarly, input 1 is chosen when  $S1S0$  is set to 01, input 2 is chosen when  $S1S0$  is set to 10, and input 4 is chosen when  $S1S0$  is set to 11. The selection inputs control the mode of operation of the register according to the functions entries. When the signal  $S1S0$  equals zero, the current value of the register is transferred to the D inputs of the flip-flops. The condition establishes a connection between the output and input of each flip-flop, forming a closed loop. During the transition of the next clock edge, the binary value that each flip-flop holds is transferred without any change in state. When the value of  $S1S0$  is equal to 01, the multiplexer's terminal 1 is connected to the D inputs of the flip-flop. This results in a shift-right operation, where the serial input is transferred into flip-flop A4. When the value of  $S1S0$  is equal to 10, performing a shift left operation causes the other serial input to be directed into flip-flop A1. The transfer of binary information from the parallel input lines into the register occurs concurrently during the subsequent clock cycle, namely when  $S1S0$  is equal to 11.

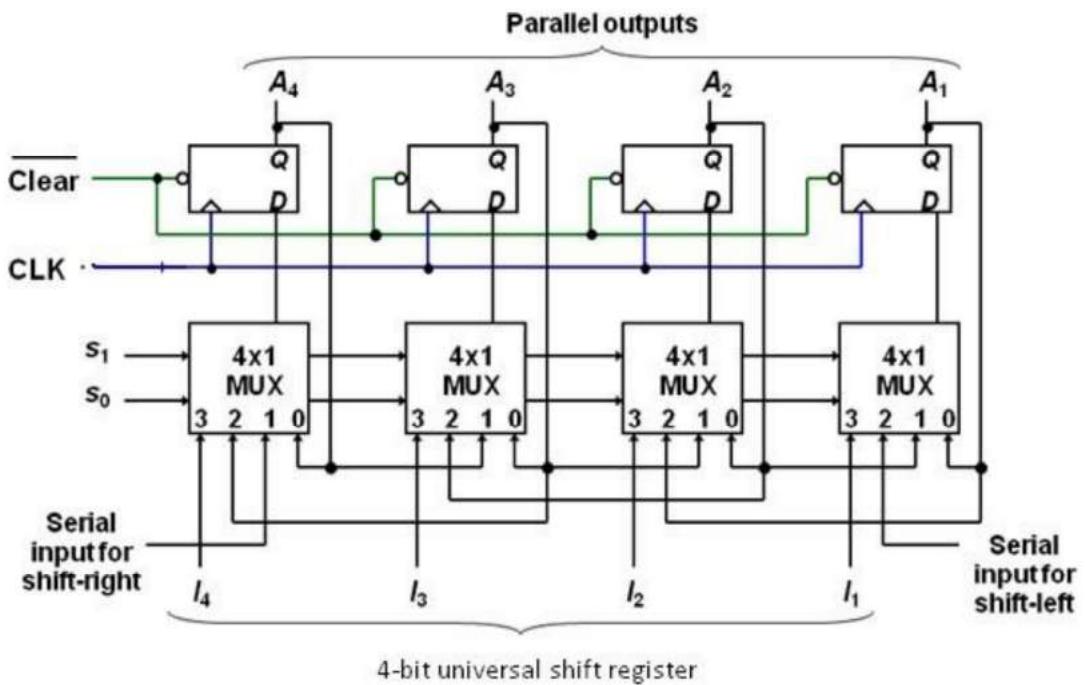


Figure. The purpose of this study is to present a logic diagram of a 4-bit universal shift register.

The following is a function table representing the register.

modecontrol		
S0	S1	registeroperation
0	0	Nochange
0	1	ShiftRight
1	0	Shiftleft
1	1	Parallelload

#### 4.5 Counters: An Examination

- A counter records and occasionally displays the frequency of an event or process, usually in relation to a clock signal. A digital counter is a group of flip flops that change state in response to input pulses. Asynchronous and synchronous counters exist. Also called ripple counters, asynchronous counters.
- Register-type devices like flip-flops make counter implementation easy in electronics. Counter categories vary. The asynchronous counter, also known as a ripple counter, utilizes the changing state bits as clocks for the subsequent state flip-flops.
- A synchronous counter is a type of counter where all state bits undergo a change in response to a single clock signal.
- The decade counter is a circuit that sequentially cycles through ten distinct states inside each stage.
- The up/down counter is a type of counter that has the ability to count in both ascending and descending order. This functionality is achieved by the control input, which dictates the direction of the counting operation.
- A ring counter is a sequential logic circuit that consists of a shift register with a feedback connection arranged in a circular configuration.
- The Johnson counter is a type of sequential logic circuit that operates as a twisted ring counter. It is commonly used in digital electronics for many applications. Additionally, cascaded counters are a configuration where multiple counters are connected in series to achieve a larger counting range or to do more complex counting operations.
- The modulus counter is a computational tool used to determine the remainder of a division operation.

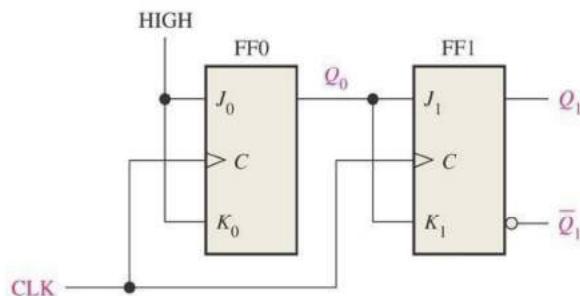
Each of these tools has a distinct purpose in various applications. Counter circuits are typically of a digital form and operate by counting in natural binary. Various sorts of counter circuits are readily available as digital building blocks. For instance, the 4000 series encompasses a range of chips that are designed to implement different types of counters.

There are instances where utilizing a counting sequence other than the natural binary sequence, such as the binary coded decimal counter, a linear feed-back shift register counter, or a gray-code counter, can offer certain benefits.

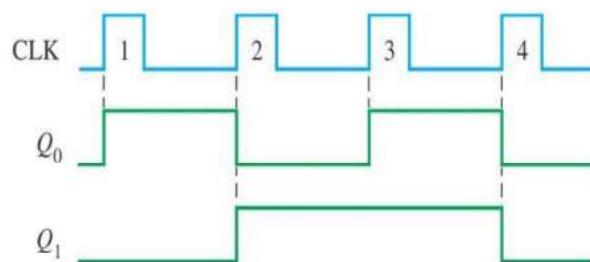
Counters serve as valuable components in various applications such as digital clocks, timers, oven timers, VCR clocks, and similar devices.

**Synchronous counters** refer to a type of digital circuit that utilizes a common clock signal to control the timing of its operations.

flops. □ A 2



Asynchronous counters can be classified as serial counters. The reason for their delayed operation is because each flip-flop is only able to transition to a new state until all preceding flip-flops have completed their state transitions. In the event when the clock frequency is significantly elevated, it is possible for the asynchronous counter to omit certain states. The aforementioned issue is resolved through the utilization of synchronous counters or parallel counters. Synchronous counters are a type of counter where all the flip flops are activated simultaneously by the clock pulses. These counters are characterized by the application of a common clock pulse to all the flip flops in the circuit. Specifically, the mentioned counter is a 4-bit synchronous binary counter.



The **design of synchronous counters** refers to the process of creating circuits that can count and display binary numbers in a sequential manner.

In order to achieve a systematic approach to the construction of synchronous counters. The aforementioned methodology is employed.

**Step 1:** State Diagram: Create a state diagram that illustrates all the potential states. A state diagram, also known as an nth transition diagram, visually represents the series of states that the counter undergoes.

**Step 2:** Determining the Number of Flip-Flops To address the problem at hand, it is necessary to ascertain the appropriate number, denoted as n, of flip-flops. This value of n should be chosen such that the number of states, represented by N, is less than or equal to  $2^n$ . Additionally, the chosen number of flip-flops should align with the intended counting sequence.

**Step 3:** Determination of Flip-Flop Excitation Table: The user is required to choose the appropriate type of flip-flop and thereafter construct the excitation table. An excitation table is a tabular representation that enumerates the present state (ps), the next state (ns), and the corresponding needed excitations.

**Step 4:** Derivation of Minimal Expressions for Excitations: Determine the minimal expressions for the excitations of the flip-flops by utilizing K-maps that are constructed for the excitation of the flip-flops in relation to the present states and inputs.

**Step 5:** Logic Diagram: Create a logic diagram that is based on the minimal expressions.

This paper presents the **design of a synchronous 3-bit up-down counter utilizing JK flip-flops.**

The **first step** involves determining the necessary quantity of flip-flops. A three-bit counter necessitates the utilization of three flip-flops. The system consists of eight states, namely 000, 001, 010, 011, 101, 110, and 111. All of these states are considered valid within the context of the system. Therefore, there is a lack of concern or indifference. In order to choose between up and down modes, the system necessitates the presence of a control or mode signal denoted as M. The mode signal M is set to 1, and it decrements when M is equal to 0. The clock signal is uniformly distributed to all flip-flops (FFs) in a simultaneous manner.

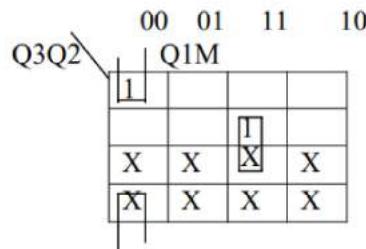
**Next, the state** diagrams for the 3-bit up-down counter are illustrated.

In Step 3, the user is instructed to choose the appropriate type of flip flop and proceed to create the excitation table. Specifically, the user has selected JK flip-flops for the purpose of designing a 3-bit up-down counter. The resulting excitation table for this counter, utilizing JK flip-flops, is depicted in the accompanying figure.

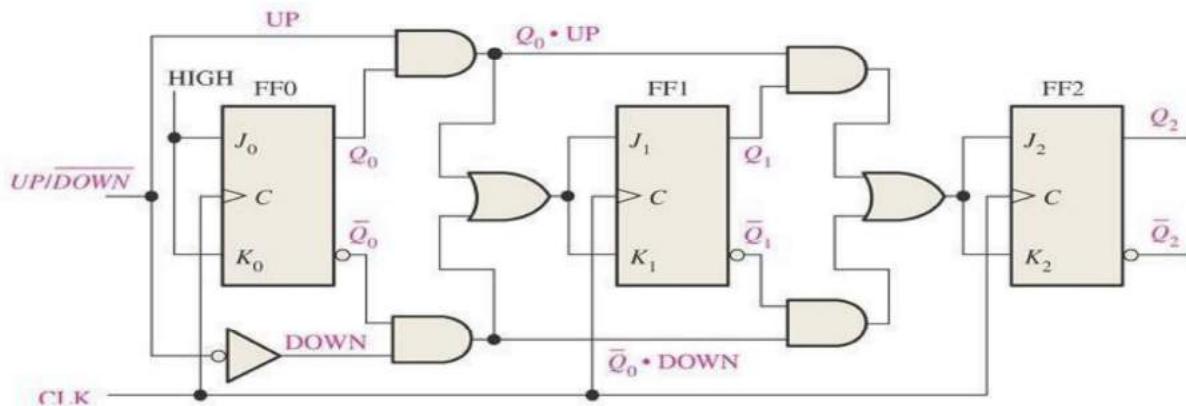
PS				mode	NS			requiredexcitations						
Q3	Q2	Q1	M	Q3	Q2	Q1	J3	K3	J2	K2	J1	K1		
0	0	0	0	1	1	1	1	x	1	x	1	x		
0	0	0	1	0	0	1	0	x	0	x	1	x		
0	0	1	0	0	0	0	0	x	0	x	x	1		
0	0	1	1	0	1	0	0	x	1	x	x	1		
0	1	0	0	0	0	1	0	x	x	1	1	x		
0	1	0	1	0	1	1	0	x	x	0	1	x		
0	1	1	0	0	1	0	0	x	x	0	x	1		
0	1	1	1	1	0	0	1	x	x	1	x	1		
1	0	0	0	0	1	1	x	1	1	x	1	x		
1	0	0	1	1	0	1	x	0	0	x	1	x		
1	0	1	0	1	0	0	x	0	0	x	x	1		
1	0	1	1	1	1	0	x	0	1	x	x	1		
1	1	0	0	1	0	1	x	0	x	1	1	x		

1	1	0	1	1	1	1	x	0	x	0	1	x
1	1	1	0	1	1	0	x	0	x	0	x	1
1	1	1	1	0	0	0	x	1	x	1	x	1

**Step4:** obtain the minimal expressions: From the excitation table we can conclude that  $J1=1$  and  $K1=1$ , because all the entries for  $J1$  and  $K1$  are either X or 1. The K-maps for  $J3, K3, J2$  and  $K2$  based on the excitation table and the minimal expression obtained from them are shown in fig.



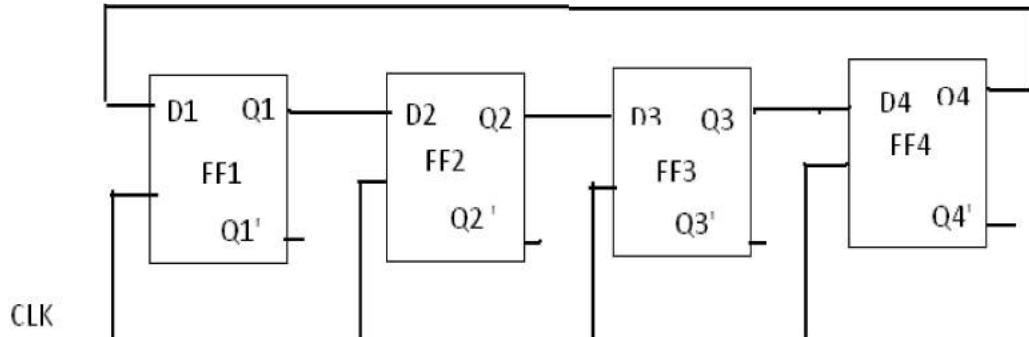
**Step5:** draw the logic diagram: a logic diagram using those minimal expressions can be drawn as shown in fig.



### Shift register counters:

One of the applications of shift register is that they can be arranged to form several types of counters. The most widely used shift register counter is ring counter as well as the twisted ring counter.

**4.6 Ring counter:** this is the simplest shift register counter. The basic ring counter using D flip-flops is shown in fig. the realization of this counter using JK FFs. The Q output of each



stage is connected to the D flip-flop connected back to the ring counter.

Fig. logic diagram of 4-bit ring counter using D flip-flops

The register holds one 1 and circulates it as clock pulses are applied. Initial FF is 1. Initial state is 1000, with Q1=1, Q2=0, Q3=0, and Q4=0. The register contents are shifted right by one bit after each clock pulse, and Q4 returns to Q1. Sequence repeats after four clock pulses. Number of states in the ring counter, or mod, equals number of FFs used. A ripple counter can count  $2^n$  bits, while a ring counter can only count n. Ring counters are more expensive than ripple counters, but they don't need a decoder because we can read the count by noting which FF is set. It is quick since it is synchronous and requires no gates or external FFs.

Timing diagram:

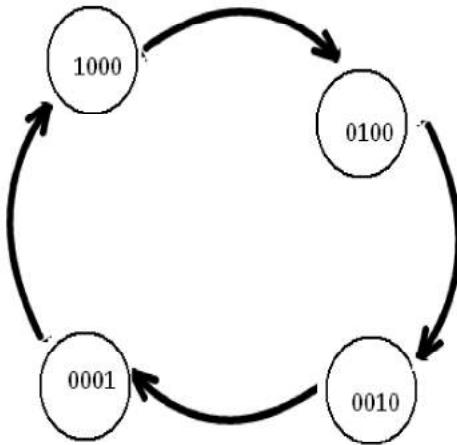
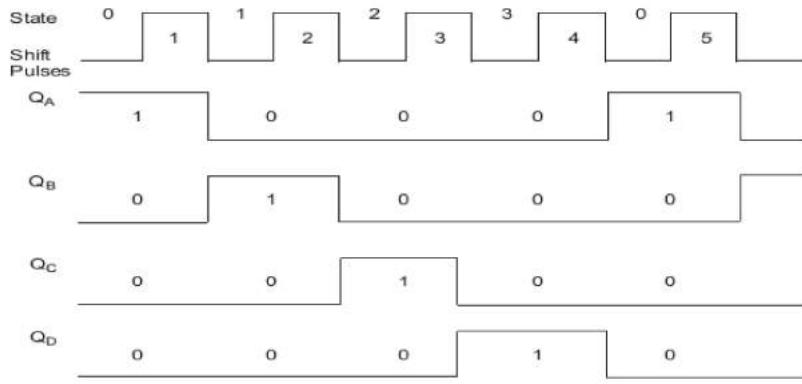


Fig. state diagram

#### 4.7 Twisted Ring counter (Johnson counter):

This counter is created from a serial-in, serial-out shift register by feeding the inverted output of the final FF to the first FF's D input. The name twisted ring counter comes from the fact that the Q output of each stage is connected to the D input of the next stage, but the Q' output of the last stage is connected to the D input of the A unique state sequence results from this feedback system.

Figure shows a 4-bit Johnson counter logic diagram utilizing D FF. Fig. shows J-K FF implementation. Figure shows state diagram and sequence table. Figure shows a Johnson counter timing diagram.

Let the counter start at 0000 with all FFs reset. After each clock pulse, Q1 is moved to Q2, Q2 to Q3, Q3 to Q4, and Q4 to Q1 in the manner shown in fig.

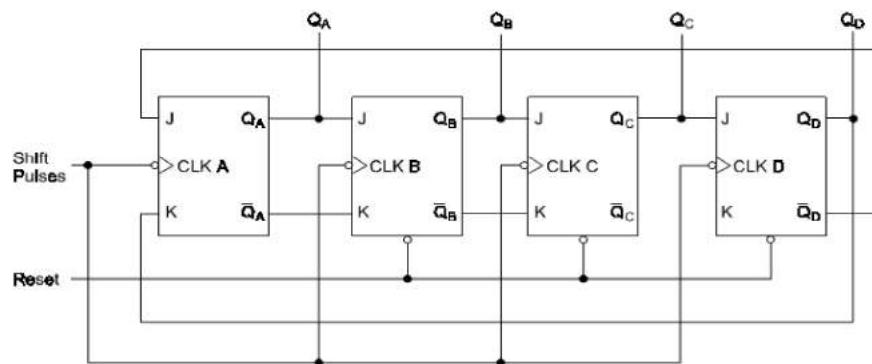


Fig. Johnson counter with JK flip-flops

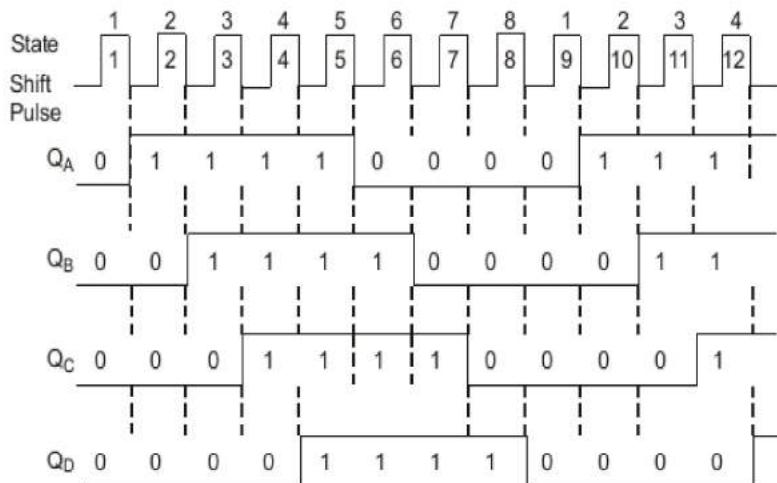
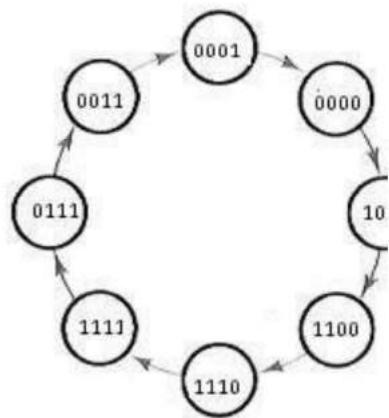


Figure: timing diagram

**Statediagram:**

				afterclock pulse
Q1	Q2	Q3	Q4	
0	0	0	0	0
1	0	0	0	1
1	1	0	0	2
1	1	1	0	3
1	1	1	1	4
0	1	1	1	5
0	0	1	1	6
0	0	0	1	7
0	0	0	0	8



**Excitationtable**