

DIGITAL LOGIC DESIGN

(Computer *Science Engineering*)

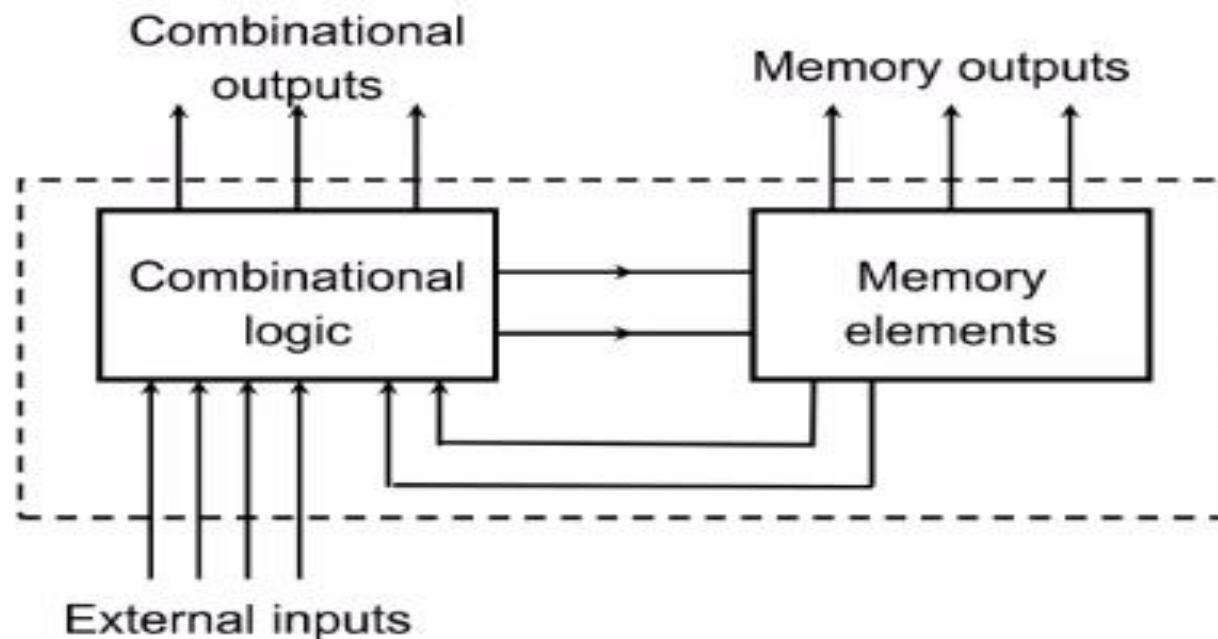
Presented by
VENKATARAMANA BOLA
Dept. of ECE

UNIT -4 & 5

FLIP-FLOPS, REGISTERS & COUNTERS

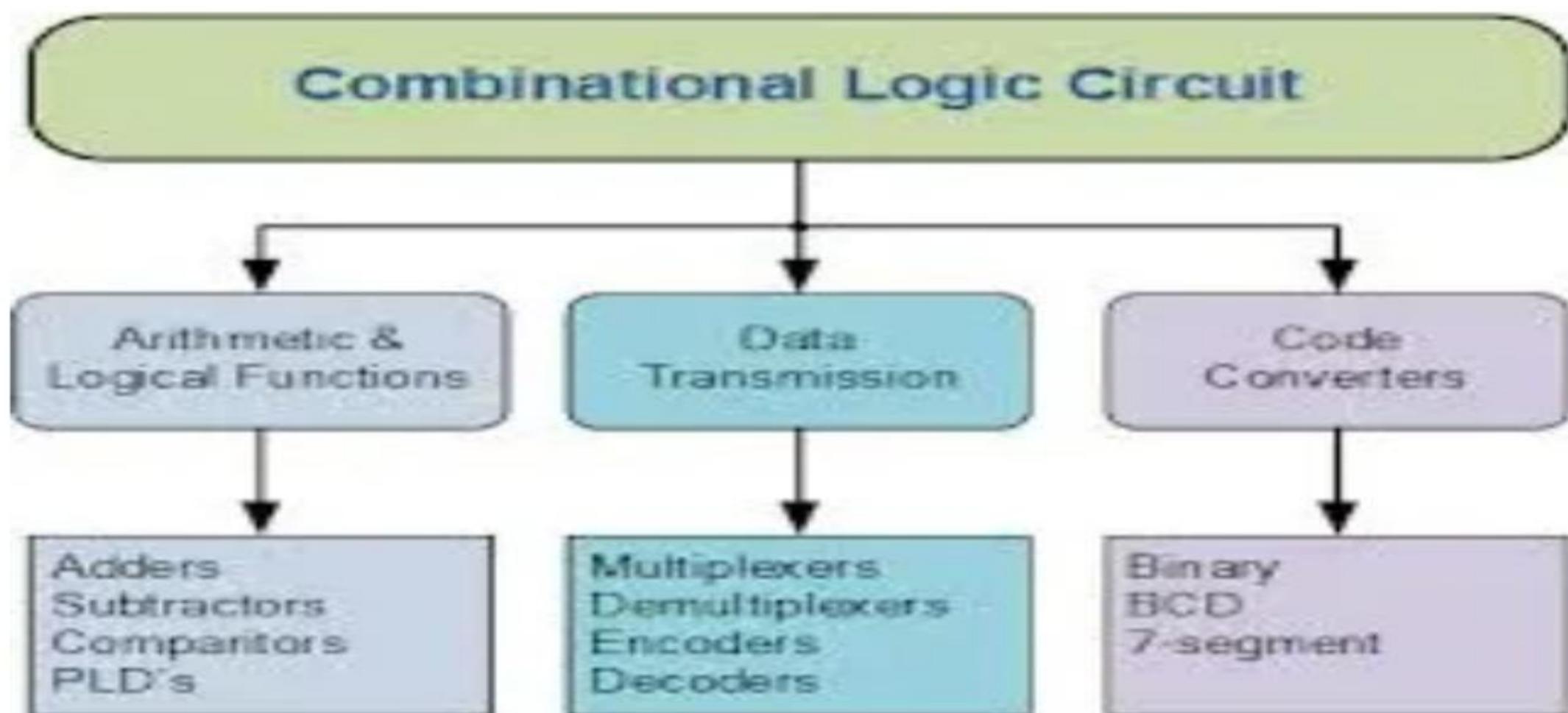
Introduction to Flip-Flops

- A **sequential circuit** consists of a *feedback path*, and employs some *memory elements*.

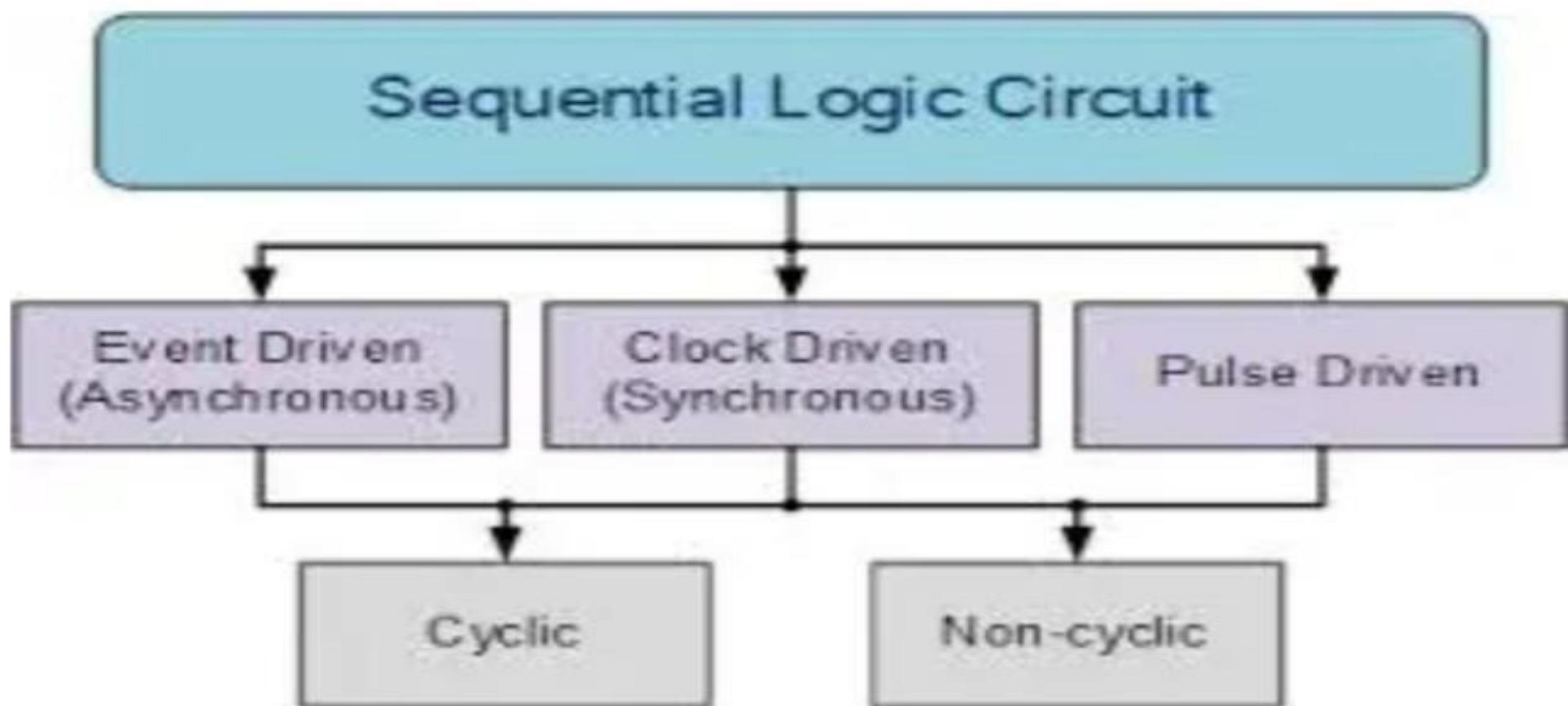


Sequential circuit = Combinational logic + Memory Elements

Continued.....



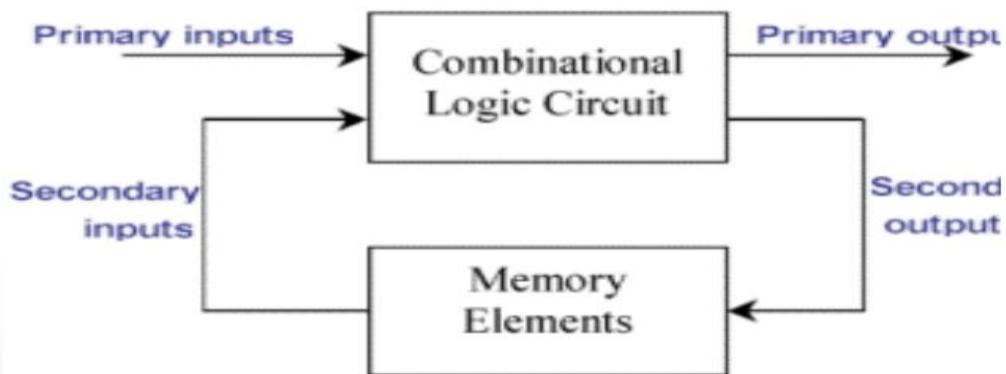
Continued.....



Continued.....

Q.1. What is sequential circuit?

Ans. These are defined as digital circuit whose output is dependent not only on the present input value but also on the past history of its input. The sequential Circuits are designed using the combinational circuits along with a memory devices known as Flip-Flops. The sequential Circuits depend over the input value as well as the stored levels.



Q.2. What is a flip flop?

Ans. Flip flop is one bit storage bistable device. Flip flop is also called latch. It stores binary value. It is the basic building block of the digital electronic systems. These are the basically the data storing devices which store the information of two stable states of the system. A flip-flop stores only a single bit of data at a time.

Continued.....

Notion of Clock

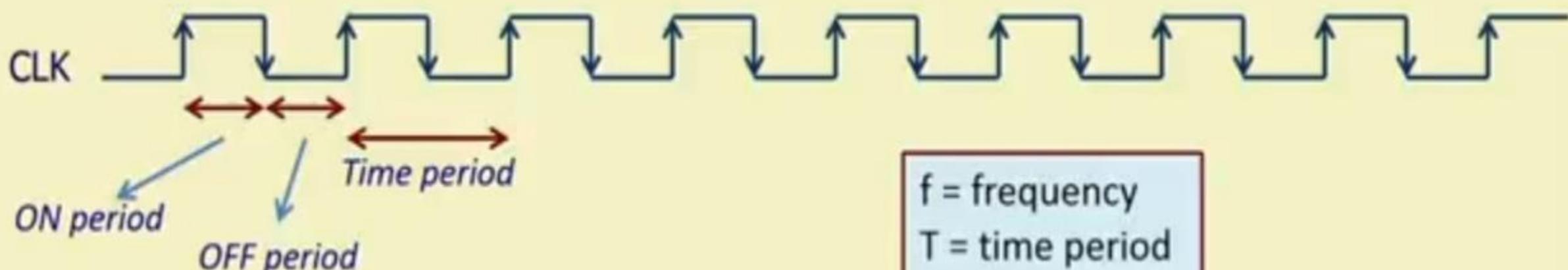
- A clock is a periodic (repetitive) rectangular pulse train that synchronizes the operation of a synchronous sequential circuit.
- For synchronous operation, the storage elements must be clock triggered.
 - Flip-flops (as opposed to latches, which are level triggered).



Continued.....

Edge-Triggered Flip-Flop

- An edge-triggered flip-flop *changes its state* in synchronism with a *clock pulse*.
 - Either at the *positive-edge* or at the *negative-edge*.
 - Useful in the design of synchronous sequential circuits, where all changes in the circuit outputs occur in synchronism with the clock.

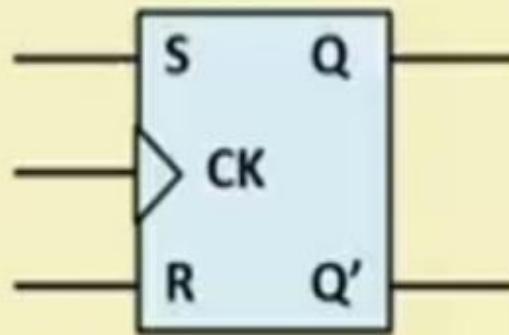


f = frequency
 T = time period
 $T = 1/f$

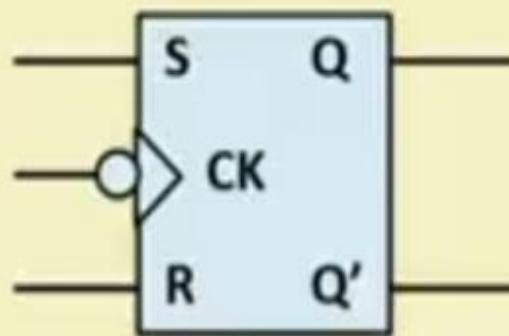
Continued.....

Edge-Triggered S-R Flip-Flop

Positive edge triggered state table



Positive edge triggered



Negative edge triggered

Q(t)	SR			
	00	01	11	10
0	0	0	X	1
1	1	0	X	1

Characteristic equation:

$$R \cdot S = 0$$

$$Q(t+1) = R' \cdot Q(t) + S$$

Continued.....

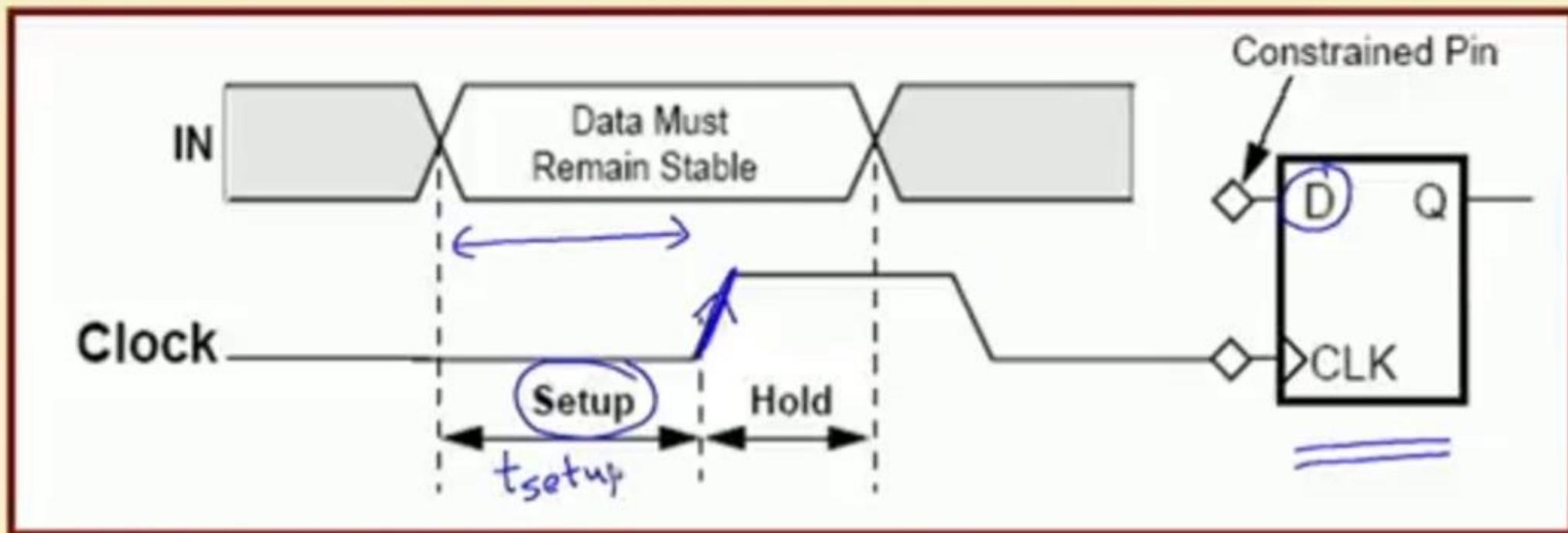
Some Definitions

- **Clock width (t_w)**: Minimum time duration for which the clock signal needs to be high in order that the flip-flops it feeds work properly.
- **Setup time (t_{setup})**: Amount of time the input to a flip-flop must be stable *before* the clock transitions high (for positive-edge triggered), or transitions low (for negative-edge triggered).
- **Hold time (t_{hold})**: Amount of time the input to a flip-flop must be stable *after* the clock transitions high (for positive-edge triggered), or transitions low (for negative-edge triggered).

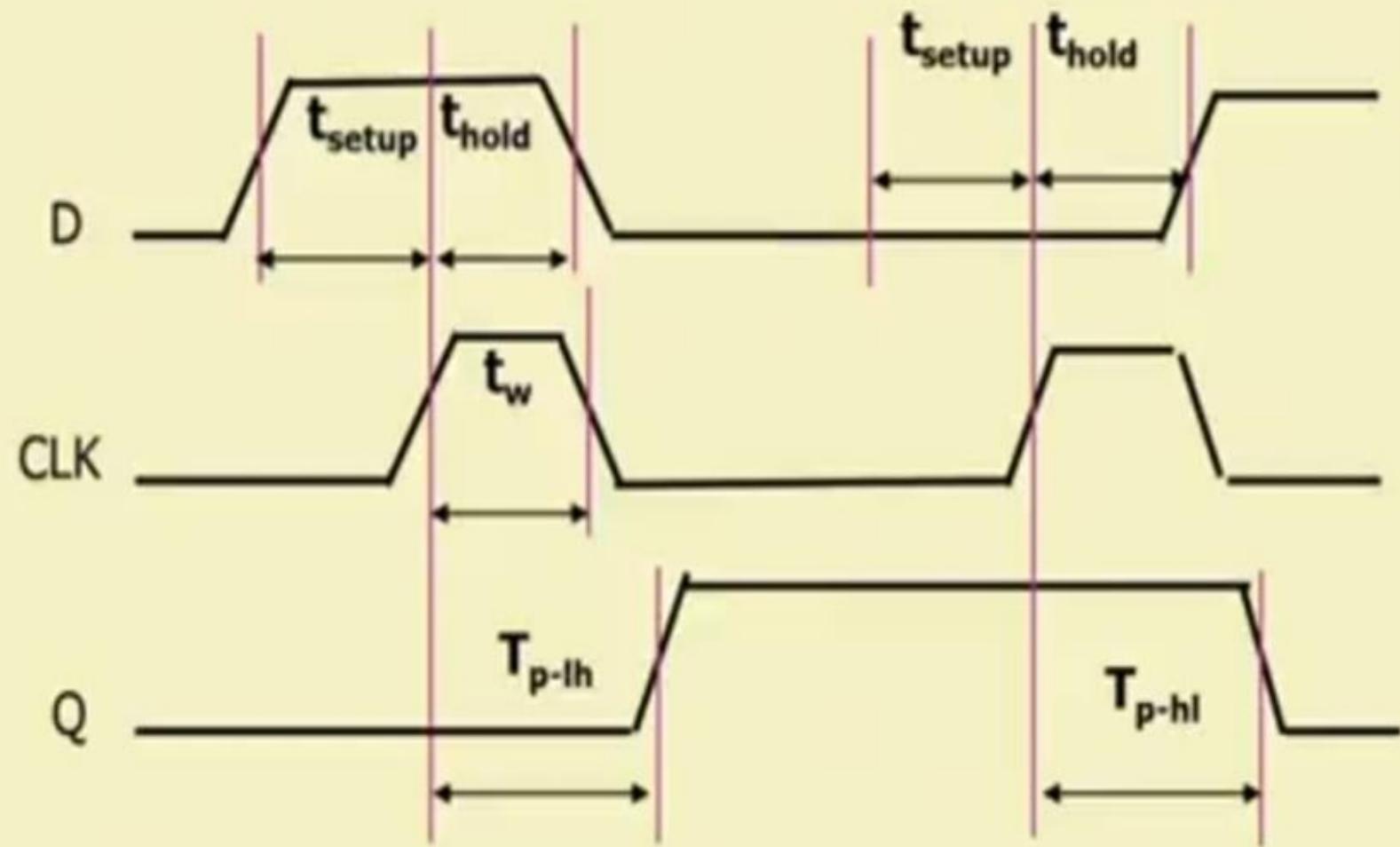
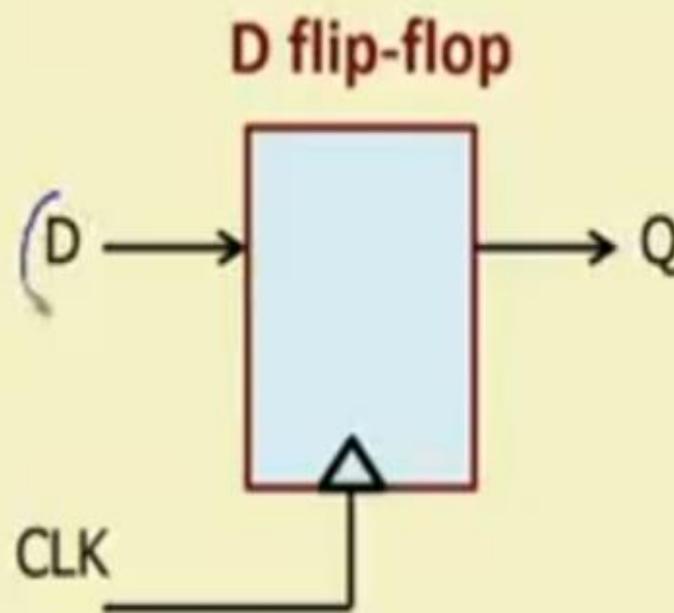


Continued.....

- *Propagation delays (t_{p-lh} and t_{p-hl})*: Delay between clocking event (low-to-high or high-to-low transition) and change in the output.



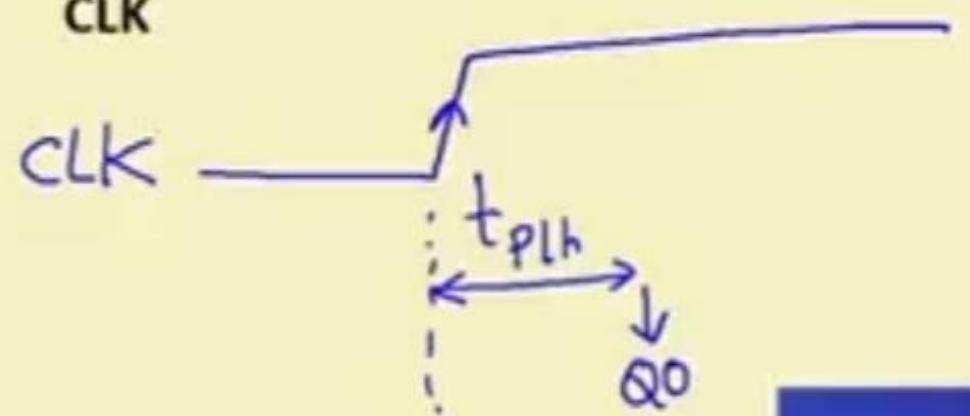
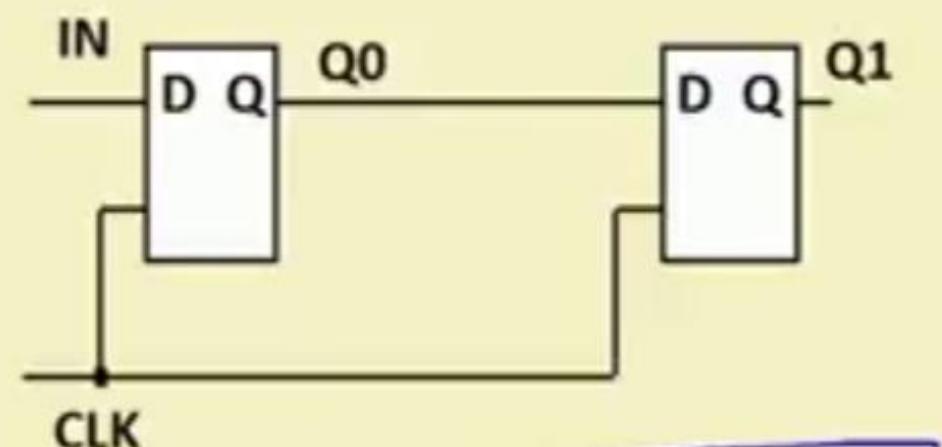
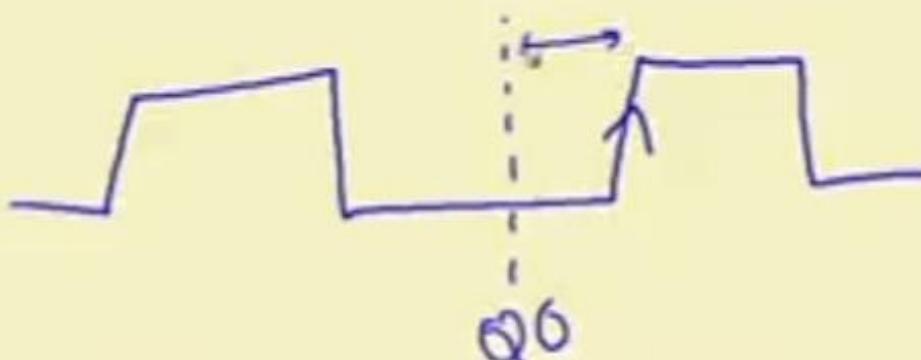
Continued.....



Continued.....

Cascading Flip-flops

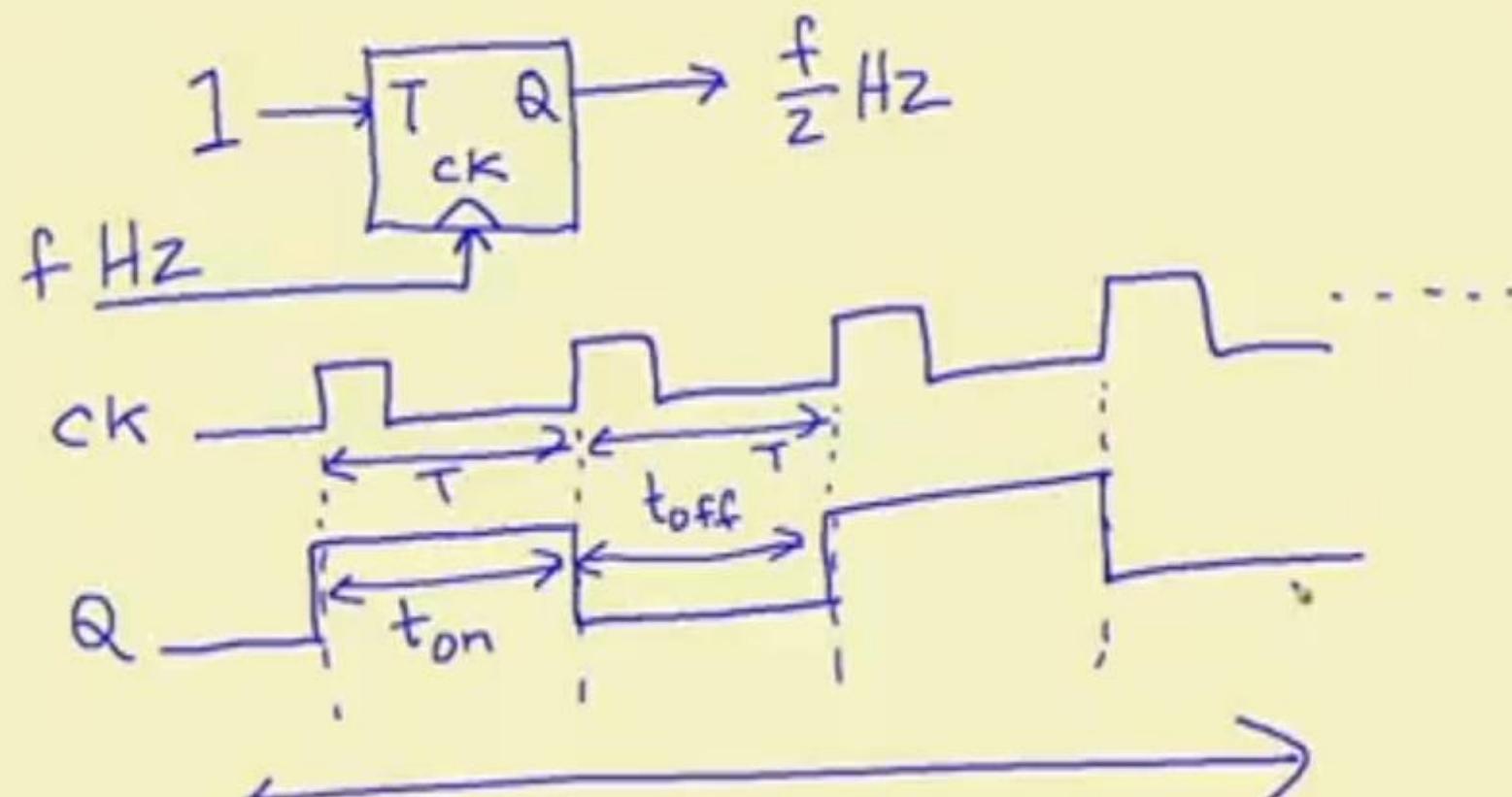
- Suppose that the flip-flop propagation delay exceeds the hold time.
- Second stage can commit its input before Q0 changes.



Continued.....

Example 1

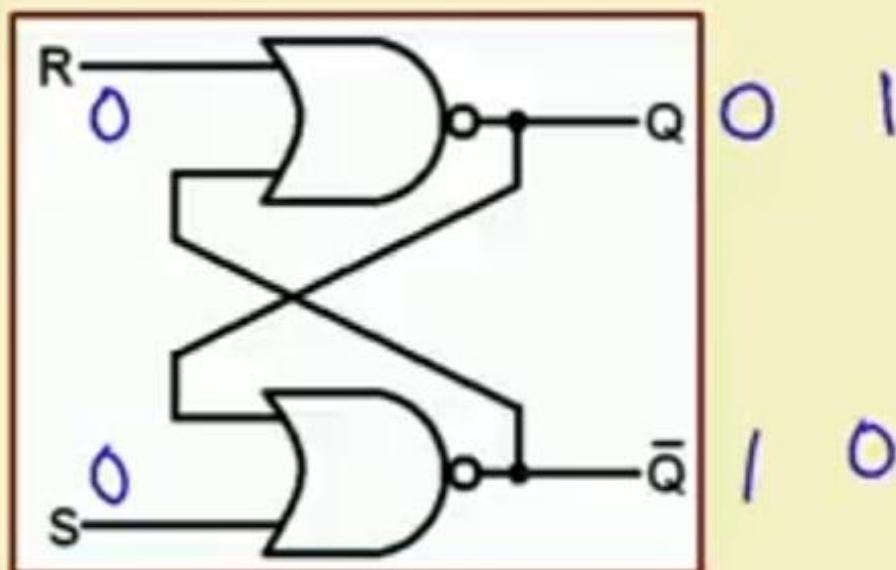
- Given a clock signal of frequency $f \text{ Hz}$, how to generate another clock of frequency $f/2 \text{ Hz}$.
 - Frequency division.



Continued.....

The Set-Reset (S-R) Latch

- Consists of a pair of cross-coupled NOR or NAND gates.
 - Two inputs (S and R) and two outputs (Q and Q').
 - The output can be set to 0 or 1 by applying suitable values on S and R inputs.

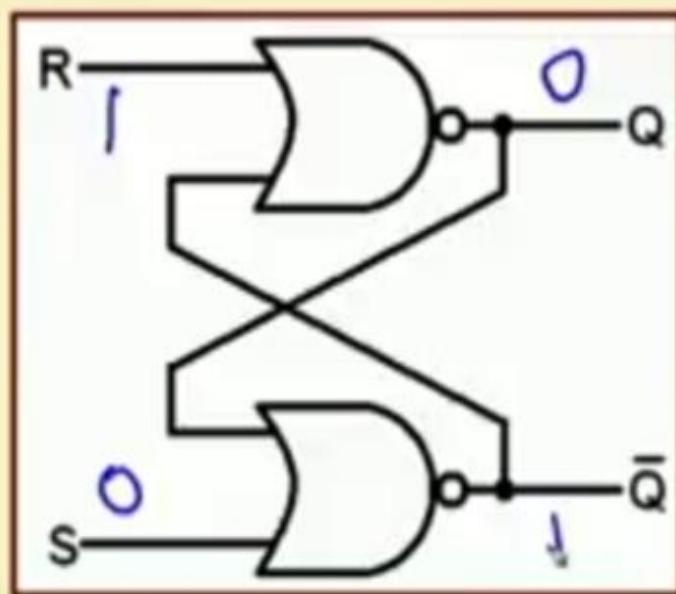


S	R	Q	Q'
0	0	<u>No change</u>	<u>No change</u>
0	1	0	1
1	0	1	0
1	1	?	?

Continued.....

The Set-Reset (S-R) Latch

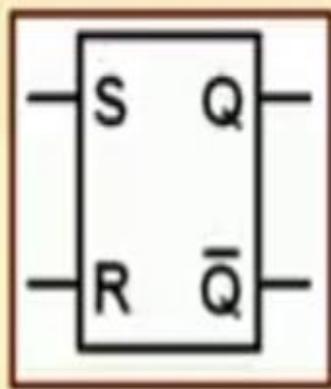
- Consists of a pair of cross-coupled NOR or NAND gates.
 - Two inputs (S and R) and two outputs (Q and Q').
 - The output can be set to 0 or 1 by applying suitable values on S and R inputs.



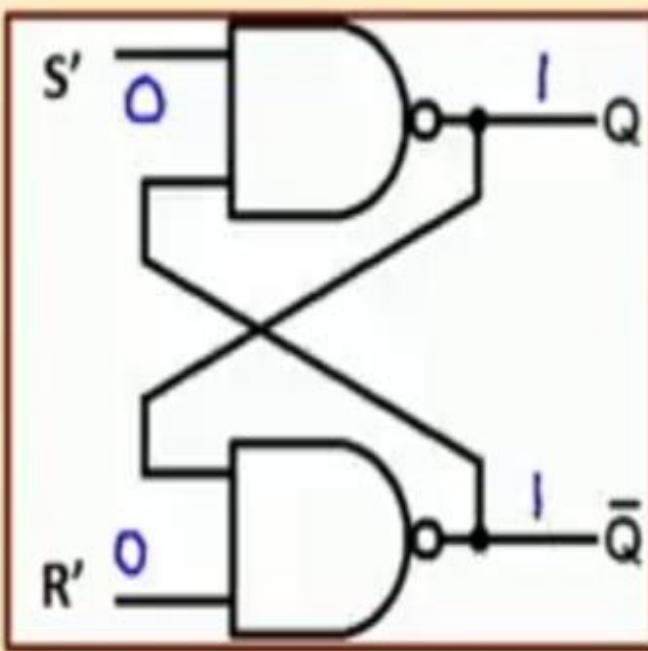
S	R	Q	Q'	
0	0	No change	No change	
0	1	0	1	
1	0	1	0	
1	1	?	?	Inva

Continued.....

Why is $S = R = 1$ an invalid input?



Logic symbol for
S-R latch

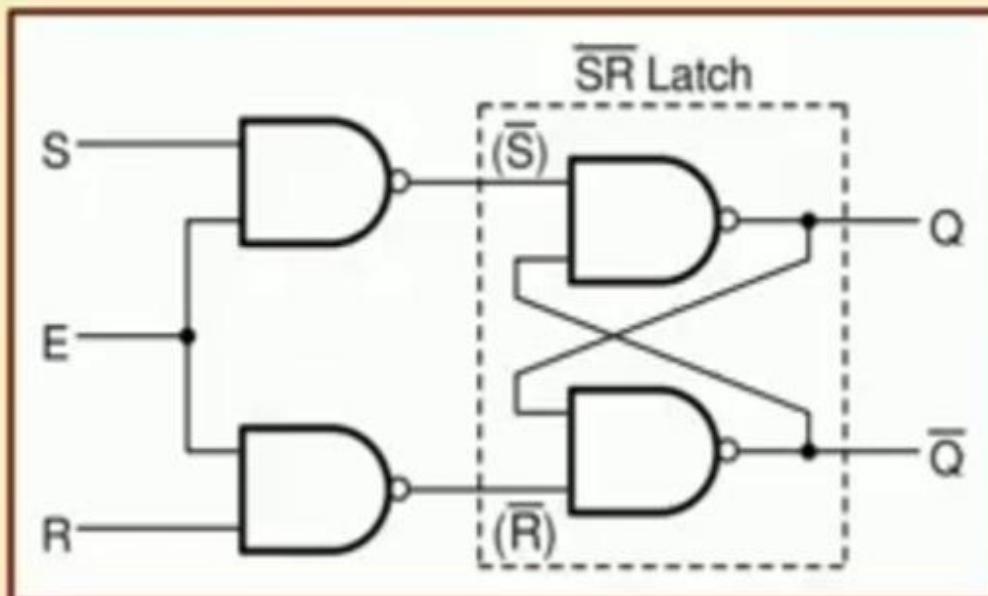


$S=0 \ R=0 \Rightarrow NC$
 $S=1 \ R=1 \Rightarrow ?$

Continued.....

Gated S-R Latch

- A gated latch requires an *enable* input (*E*).
 - When $E = 1$, the latch is *active*.
 - When $E = 0$, the latch is *de-active* and the outputs do not change.



E	S	R	Q	Q'
0	X	X	NC	NC
1	0	0	NC	NC
1	0	1	0	1
1	1	0	1	0
1	1	1	?	?

Continued.....

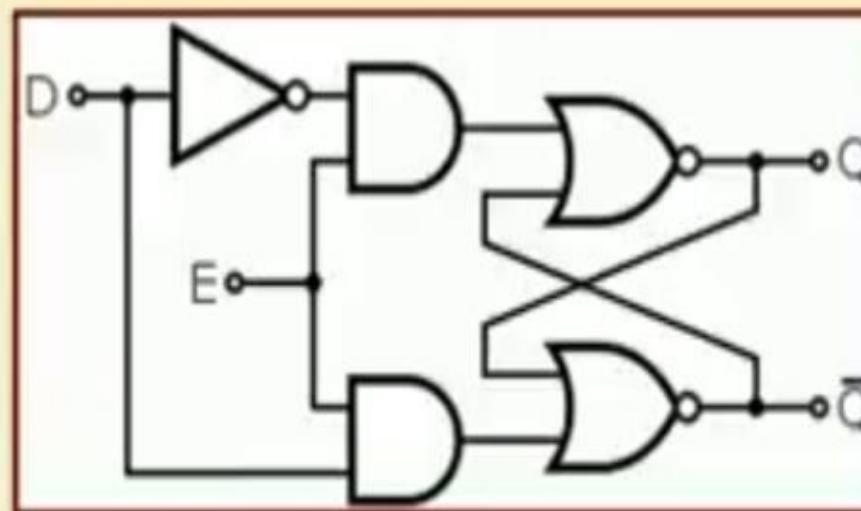
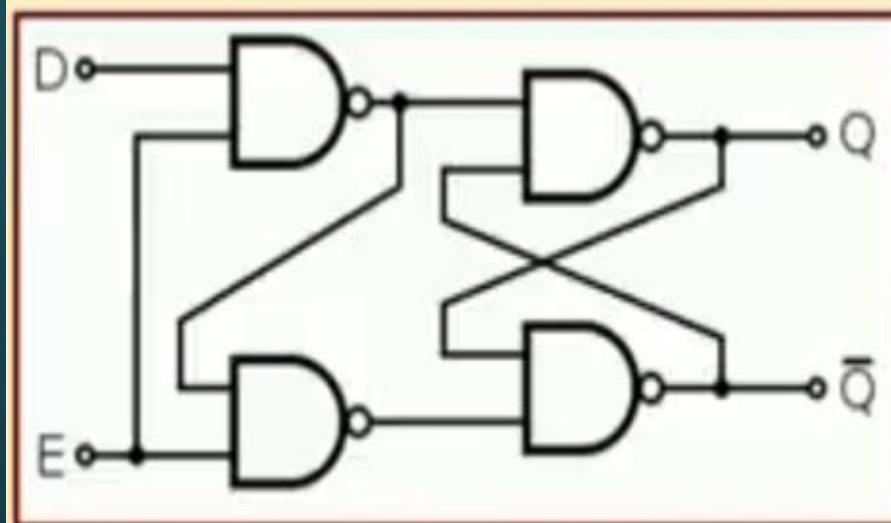
Excitation table for S-R flip-flop:

Circuit changes		Required value	
From	To	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Continued.....

Gated D Latch

- A *D-latch* has a single input *D*.
 - When the latch is enabled, the value at *D* gets stored in *Q*.
- A *gated D-latch*:



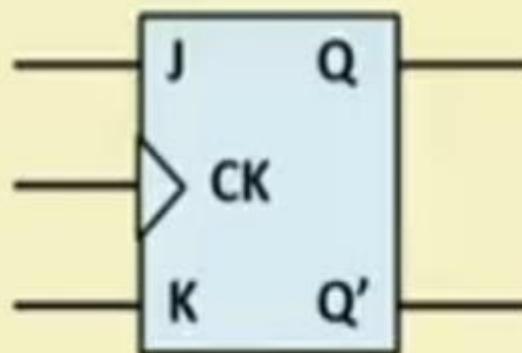
<i>E</i>	<i>D</i>	<i>Q</i>	<i>Q'</i>
0	X	NC	NC
1	0	0	1
1	1	1	0

Continued.....

Edge-Triggered J-K Flip-Flop

- The J-K flip-flop is the most versatile flip-flop.
 - Like in S-R flip-flop, it has two inputs J and K, but does not have any invalid inputs.

Positive edge triggered state table



Positive edge triggered

CK	J	K	$Q(t+1)$	$Q(t+1)'$
0/1	X	X	$Q(t)$	$Q(t)'$
↑	0	0	$Q(t)$	$Q(t)'$
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$Q(t)'$	$Q(t)$

Continued.....

JK	00	01	11	10
Q(t)	0	0	1	1
1	1	0	0	1

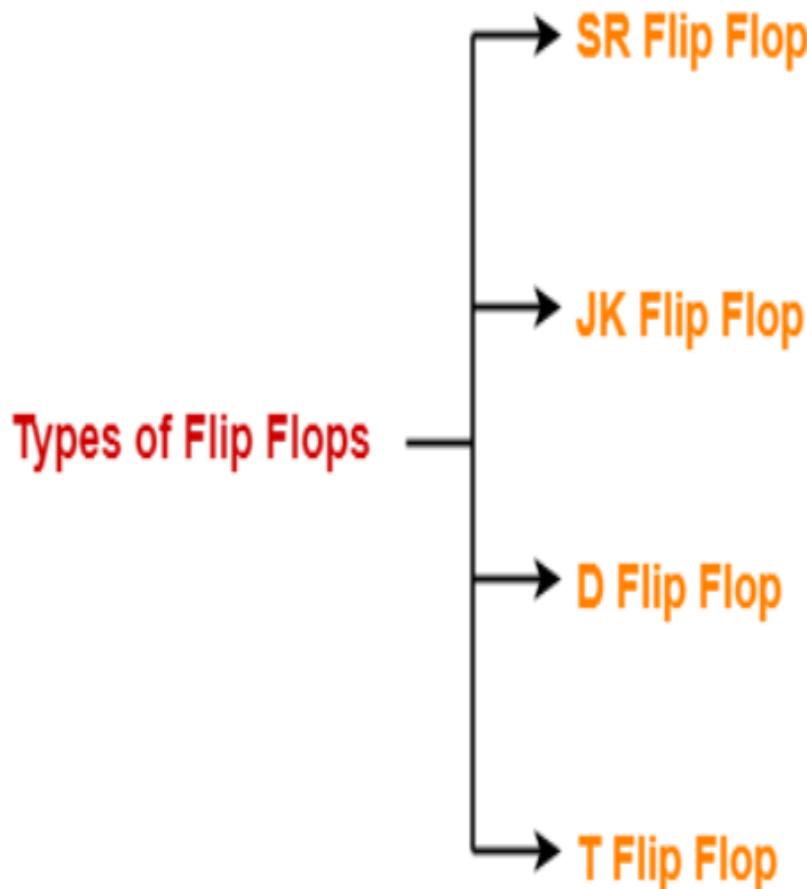
Characteristic equation:

$$Q(t+1) = J \cdot Q(t)' + K' \cdot Q(t)$$

Excitation Table

Circuit changes		Required value	
From	To	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Continued.....



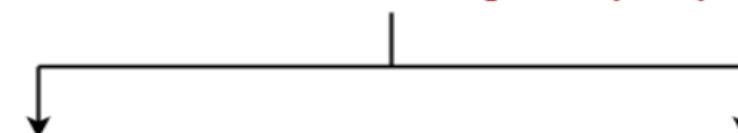
SR Flip Flop-

- SR flip flop is the simplest type of flip flops.
- It stands for **Set Reset flip flop**.
- It is a clocked flip flop.

Construction of SR Flip Flop-

There are following two methods for constructing a SR flip flop-

Methods for Constructing SR Flip Flop



By Using NOR Latch

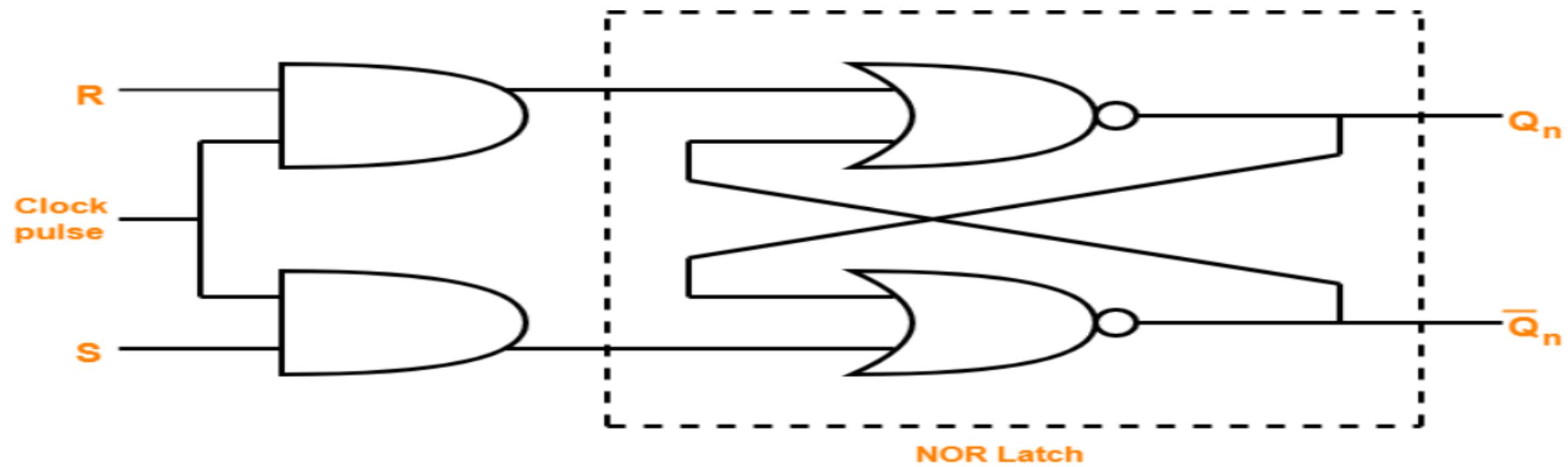
By Using NAND Latch

1. By using NOR latch
2. By using NAND latch

Continued.....

Logic Circuit-

The logic circuit for SR Flip Flop constructed using NOR latch is as shown below-



SR Flip Flop Using NOR Latch

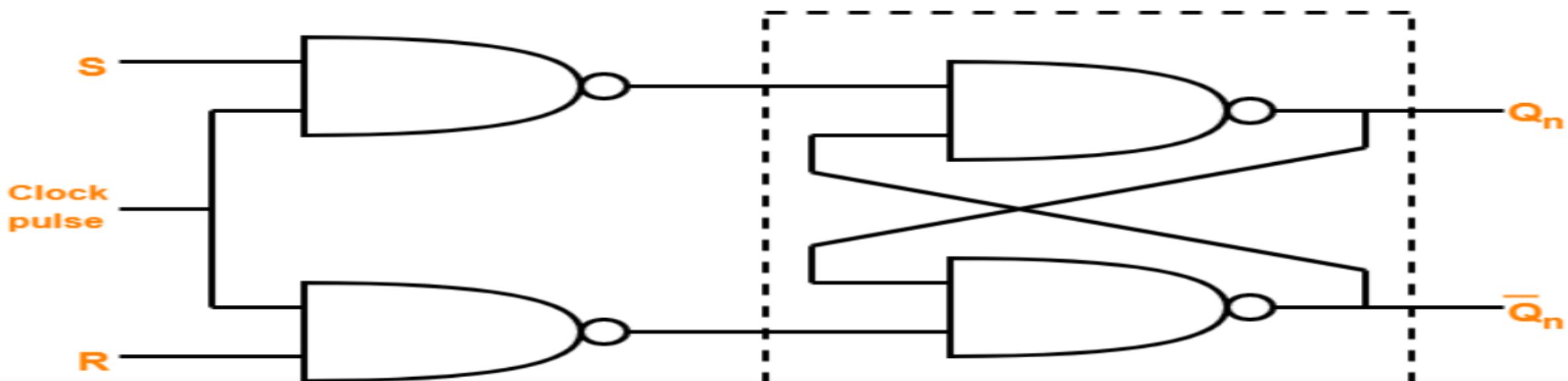
Continued.....

This method of constructing SR Flip Flop uses-

- NAND latch
- Two NAND gates

Logic Circuit-

The logic circuit for SR Flip Flop constructed using NAND latch is as shown below-



Continued.....

Logic Symbol-

The logic symbol for SR Flip Flop is as shown below-



Logic Symbol

Continued.....

The truth table for SR Flip Flop is as shown below-

INPUTS			OUTPUTS
S	R	Q_n (Present State)	Q_{n+1} (Next State)
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	Indeterminate
1	1	1	Indeterminate

Truth Table

Continued.....

The above truth table may be reduced as-

INPUTS			OUTPUTS	REMARKS
S	R	Q_n (Present State)	Q_{n+1} (Next State)	States and Conditions
0	0	X	Q_n	Hold State condition $S = R = 0$
0	1	X	0	Reset state condition $S = 0, R = 1$
1	0	X	1	Set state condition $S = 1, R = 0$
1	1	X	Indeterminate	Indeterminate state condition $S = R = 1$

Truth Table

Continued.....

Characteristic Equation-

Draw a k map using the above truth table-



From here-

$$Q_{n+1} = (SR + SR') (Q_n + Q'_n) + Q_n (S'R' + SR')$$

$$Q_{n+1} = S + Q_n R'$$

Continued.....

What is excitation table?

For a given combination of present state Q_n and next state Q_{n+1} , excitation table tell the inputs required.

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

Excitation Table

Continued.....

JK Flip Flop-

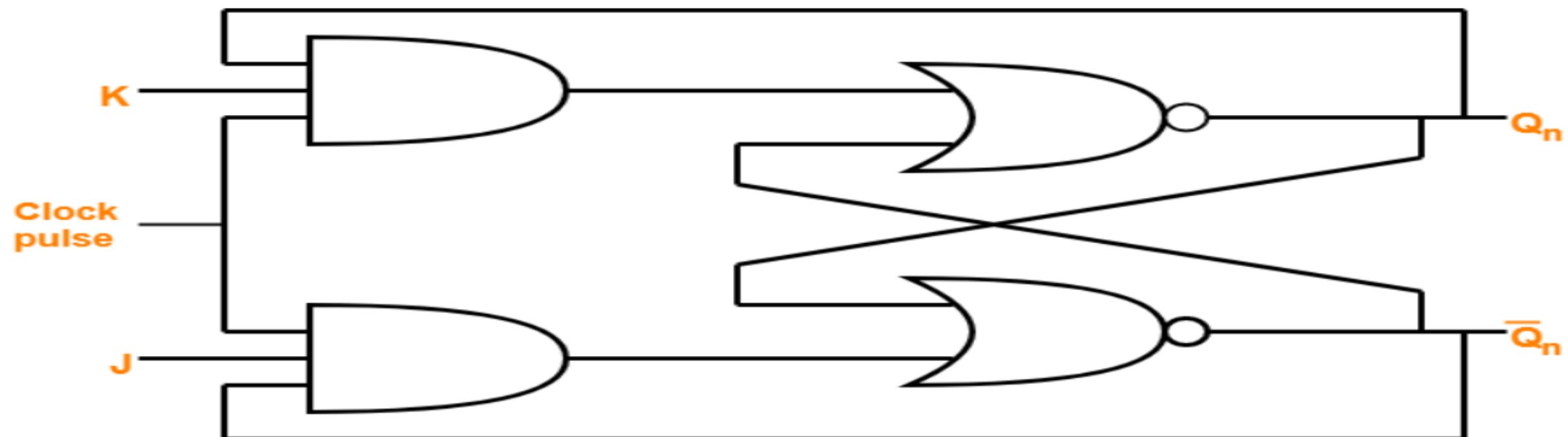
JK flip flop is a refined & improved version of **SR Flip Flop** that has been introduced to solve the problem of indeterminate state that occurs in SR flip flop when both the inputs are 1.

In JK flip flop,

- Input J behaves like input S of SR flip flop which was meant to set the flip flop.
- Input K behaves like input R of SR flip flop which was meant to reset the flip flop.

Continued.....

The logic circuit for JK Flip Flop constructed using SR Flip Flop constructed from NOR latch is as shown below-

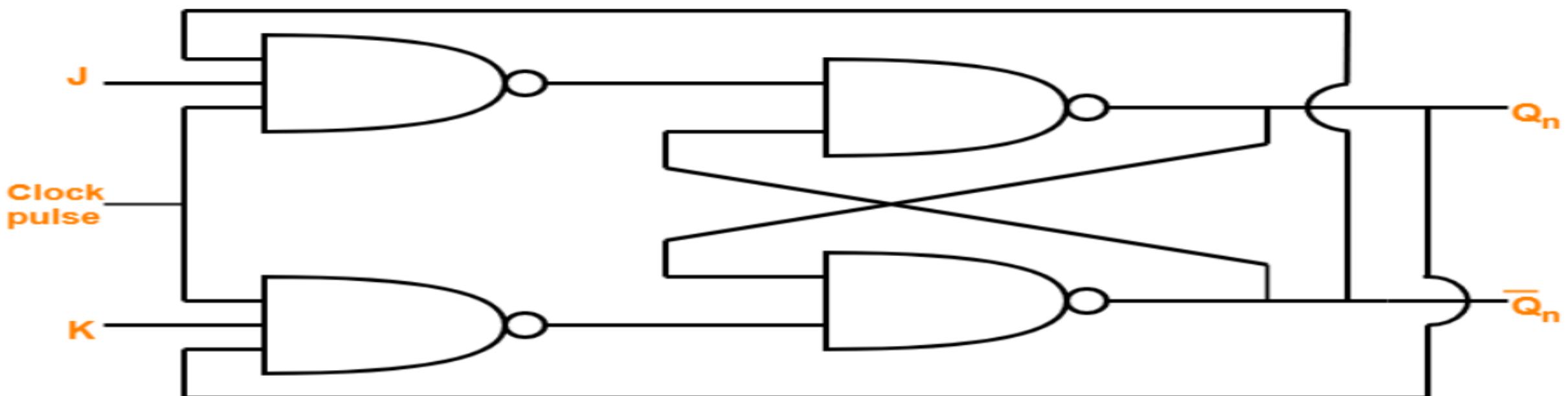


**Logic Circuit For JK Flip Flop Using SR Flip Flop
(Constructed From NOR Latch)**

Continued.....

Logic Circuit-

The logic circuit for JK Flip Flop constructed using SR Flip Flop constructed from NAND latch is as shown below-



Logic Circuit For JK Flip Flop Using SR Flip Flop
(Constructed Using NAND Latch)

Continued.....

The truth table for JK Flip Flop is as shown below-

INPUTS		Q_n (Present State)	OUTPUTS	
J	K		Q_{n+1} (Next State)	
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

Continued.....

The above truth table may be reduced as-

INPUTS			OUTPUTS	REMARKS
J	K	Q_n (Present State)	Q_{n+1} (Next State)	States and Conditions
0	0	X	Q_n	Hold State condition $J = K = 0$
0	1	X	0	Reset state condition $J = 0, K = 1$
1	0	X	1	Set state condition $J = 1, K = 0$
1	1	X	Q'_n	Toggle state condition $J = K = 1$

Truth Table

Continued.....

Draw a k map using the above truth table-



From here-

$$Q_{n+1} = Q'_n (JK + JK') + Q_n (J'K' + JK')$$

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

Continued.....

Excitation Table-

The excitation table of any flip flop is drawn using its truth table.

What is excitation table?

For a given combination of present state Q_n and next state Q_{n+1} , excitation table tell the inputs required.

Q_n	Q_{n+1}	S	R
0	0	0	\times
0	1	1	\times
1	0	\times	1
1	1	\times	0

Excitation Table

Continued.....

Q.6. How race around condition can be eliminated?

Ans. It is essential to understand the race around condition before the development of edge triggered flip flop. As we know that the conditions $S==1$ and $R==1$ are not allowed in flip flop by the use of feedback correction. Under this situation when input J and K are 1 and 1 output will change from 0 to 1. To avoid race around condition, we use master slave flip flop. It has two different flip flop, which are connected serially.

Continued.....

What are Flip-Flops?

The flip-flops are basically the circuits that maintain a certain state unless and until directed by the input for changing that state. We can construct a basic flip-flop using four-NOR and four-NAND gates.

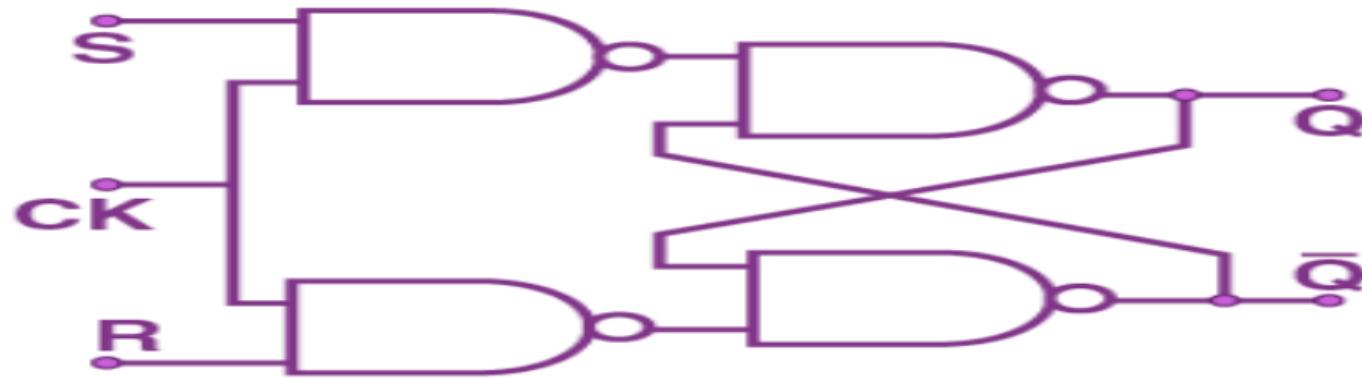
Types of Flip-Flops

The flip-flops are of the following types:

1. S-R Flip Flop
2. J-K Flip Flop
3. T Flip Flop
4. D Flip Flop

Below, you can find the logic diagrams along with the truth tables of all the various types of flip-flops:

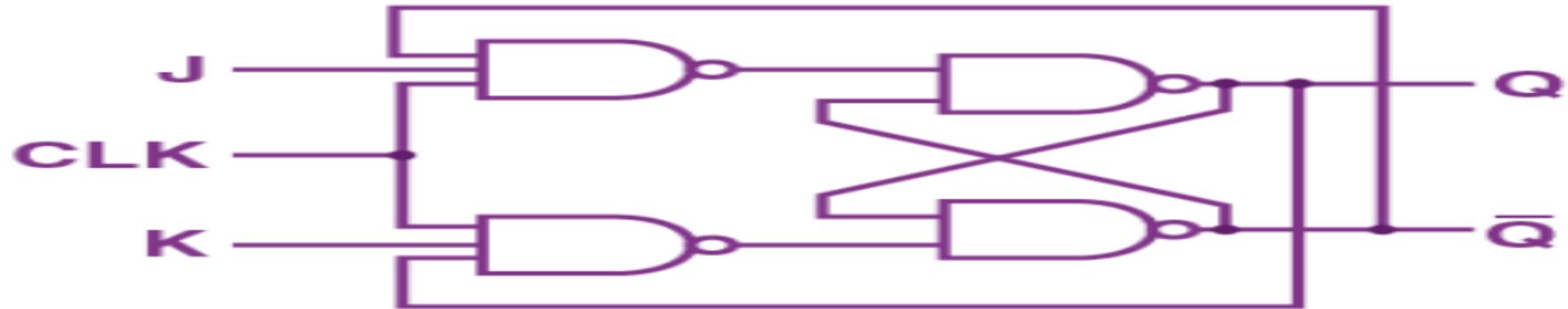
S-R FLIP-FLOP



Truth Table

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

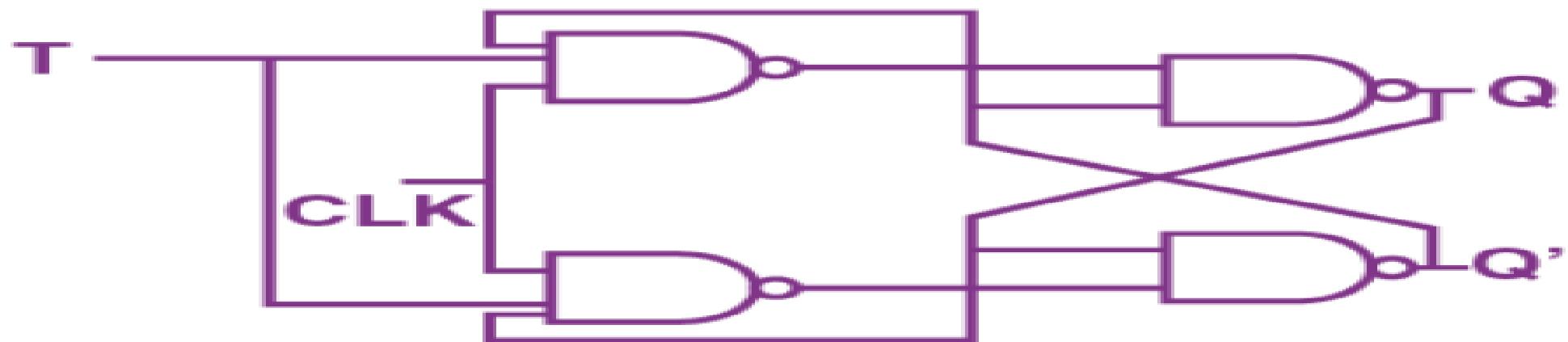
J-K FLIP-FLOP



Truth Table

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

T(Toggle)-FLIP-FLOP

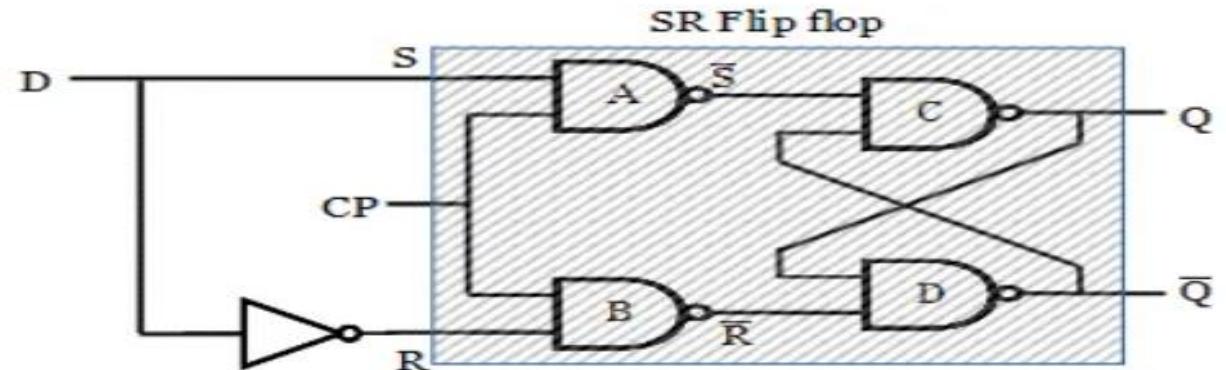
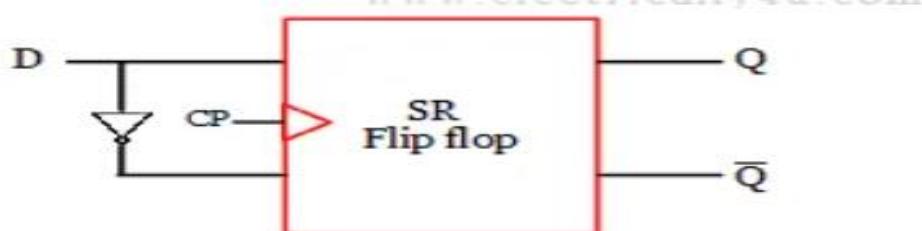


Truth Table

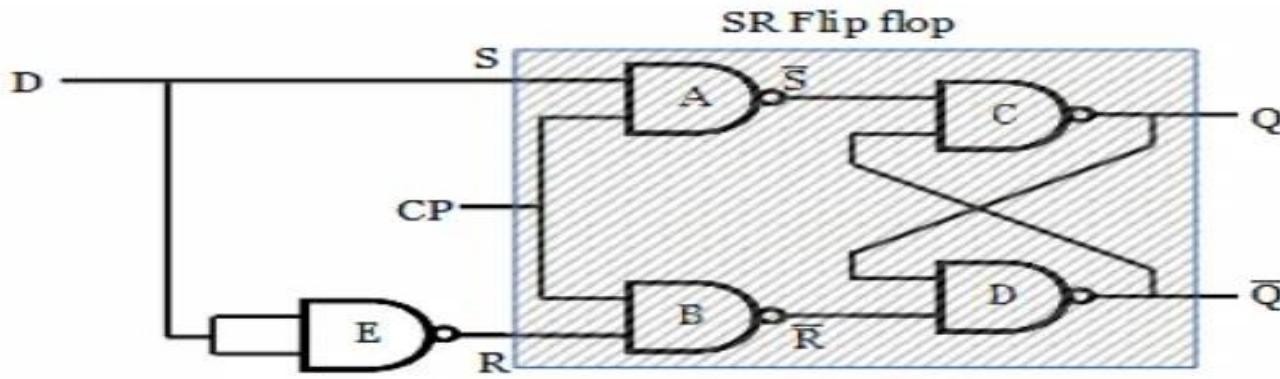
T	Q_N	Q_{N+1}
0	0	0
0	1	1
1	0	1
1	1	0

D(Delay)-FLIP-FLOP

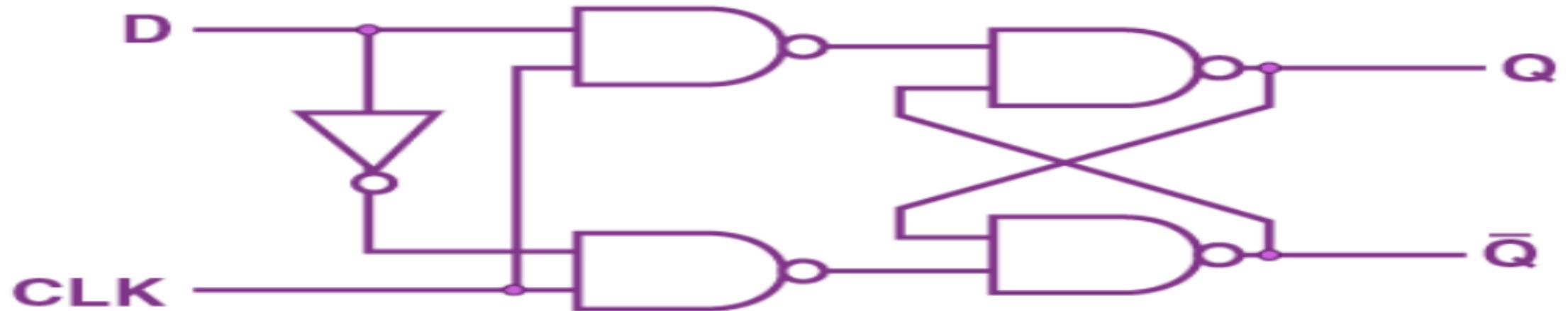
D Flip Flop is the most important of all the clocked flip-flops as it ensures that both the inputs S and R are never the same at the same time. It is constructed by joining the S and R inputs with an inverter in between them, as shown below. Thus the **D flip flop** has single input(D).



Replacing the NOT gate with single input NAND gate, the D flip flop circuit becomes



D(Delay)-FLIP-FLOP



Truth Table

Q	D	$Q_{(t + 1)}$
0	0	0
0	1	1
1	0	0
1	1	1

FLIP-FLOP CONVERSIONS

Conversion for Flip-Flops

Excitation Table

Q_N	Q_{N+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

Converting Flip-Flops

Here we will discuss the steps that one must use to convert one given flip-flop to another one. Let us assume that we have the required flip-flops that are to be constructed using the sub-flip-flops:

1. Drawing of the truth of the required flip-flop.
2. Writing of the corresponding outputs of those sub-flip-flops that are to be used from the given excitation table.
3. Drawing of the K-Maps using the required inputs of the flip-flops and then obtaining the excitation functions for the inputs of the sub-flip-flops.
4. Construction of the logic diagram in accordance with the functions that we have obtained.

Continued.....

i) Conversion of SR to JK Flip-Flop

J	K	Q_N	Q_{N+1}	S	R
0	0	0	0	0	x
0	0	1	1	x	0
0	1	0	0	0	x
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	x	0
1	1	0	1	1	0
1	1	1	0	0	1

Continued.....

Excitation Functions

$$S = JQ_N$$

$$KQ_N$$

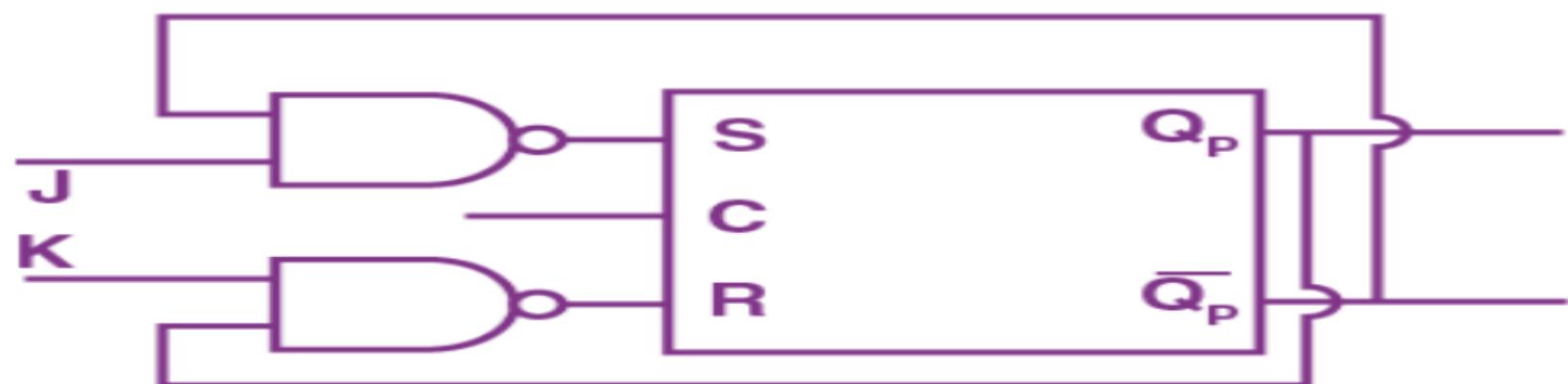
$$J$$

0	x	0	0
1	x	0	1

$$R = KQ_N$$

$$KQ_N$$

x	0	1	x
0	0	1	0



Continued.....

Conversion of S-R to D Flip-Flop:

D	Q_N	Q_{N+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

Excitation Functions

$$R = D'$$

$$S = D$$

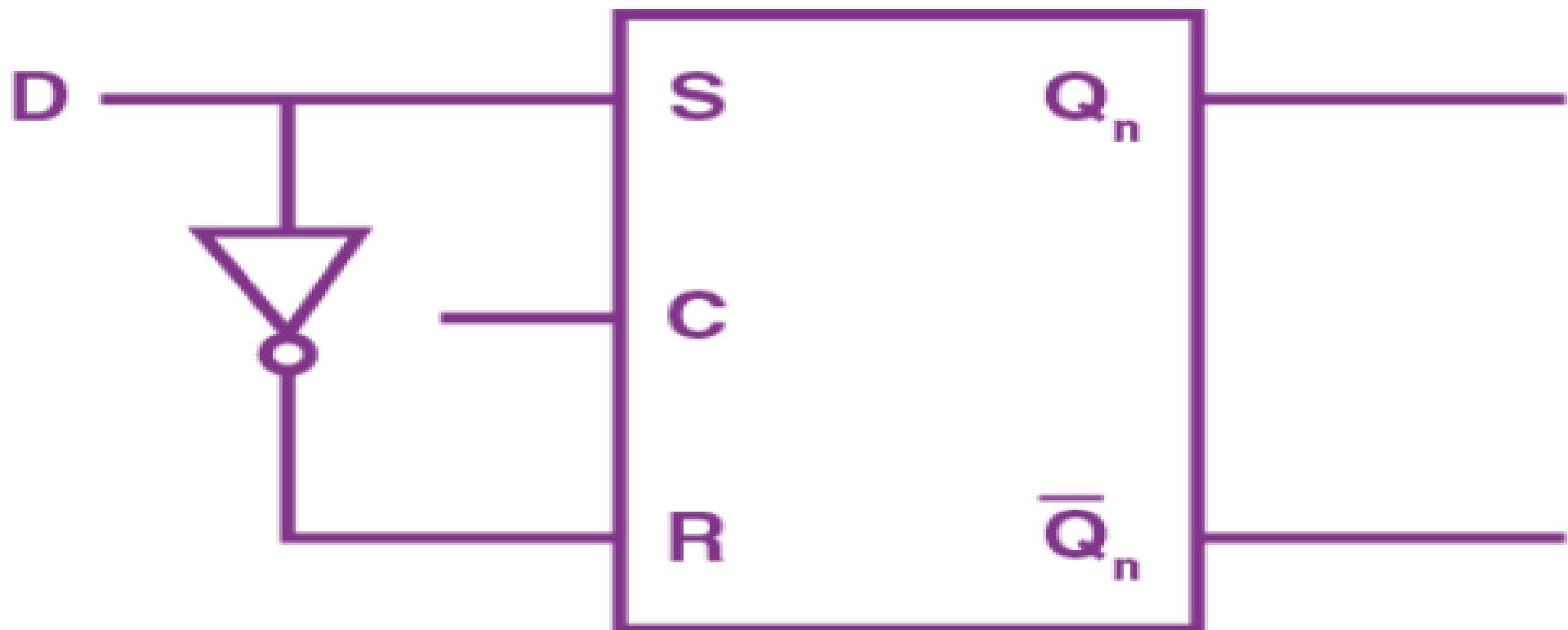
S:

D, Q_N		
	0	0
	1	X

R:

D, Q_N		
	X	1
	0	0

Continued.....



Logic Diagram

Continued.....

Applications of Flip-Flops

In this article, we have summed up the different types of flip-flops that we use in digital electronic circuits. You can find the various applications of the flip-flops below:

- Frequency dividers
- Counters
- Storage registers
- Shift registers
- Data storage
- Bounce elimination switch
- Latch
- Data transfer
- Memory
- Registers

Continued.....

Conversion of JK Flip Flop to D Flip Flop

1. Truth Table for D Flip-Flop

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

2. Excitation Table for JK Flip-Flop

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

3. Conversion Table

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

4. K-map Simplification

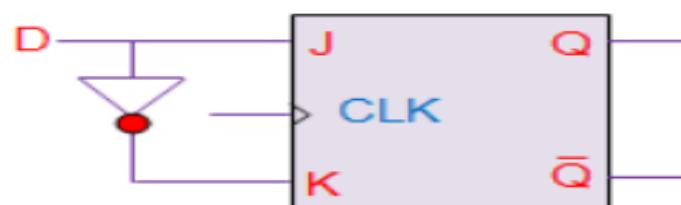
D	Q_n	0	1
0	0	0	X
1	1	X	X

$$J = D$$

D	Q_n	0	1
0	X	1	X
1	X	0	0

$$K = \bar{D}$$

5. Circuit Design



Continued.....

J-K to T-Flip-Flop conversion:

1. Truth Table for T Flip-Flop

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

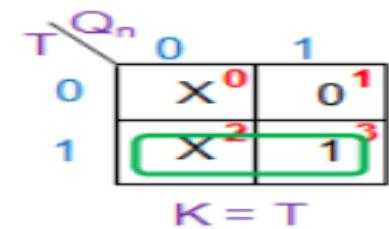
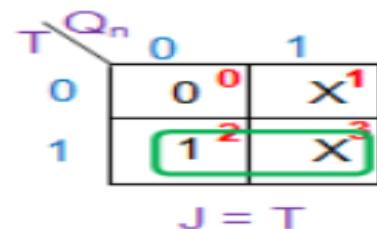
2. Excitation Table for JK Flip-Flop

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

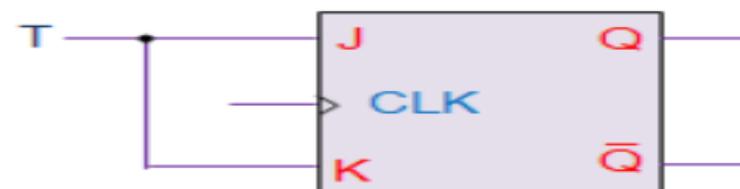
3. Conversion Table

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

4. K-map Simplification



5. Circuit Design



Continued.....

Conversion of SR Flip Flop to JK Flip Flop

1. Truth Table for JK flip-flop

Inputs		Outputs	
J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

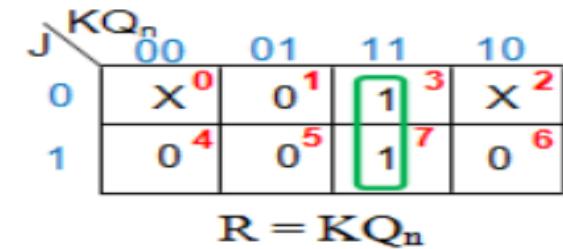
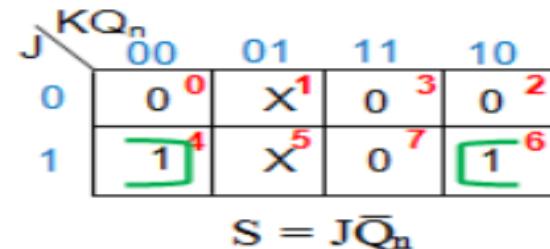
2. Excitation Table for SR flip-flop

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

3. Conversion Table

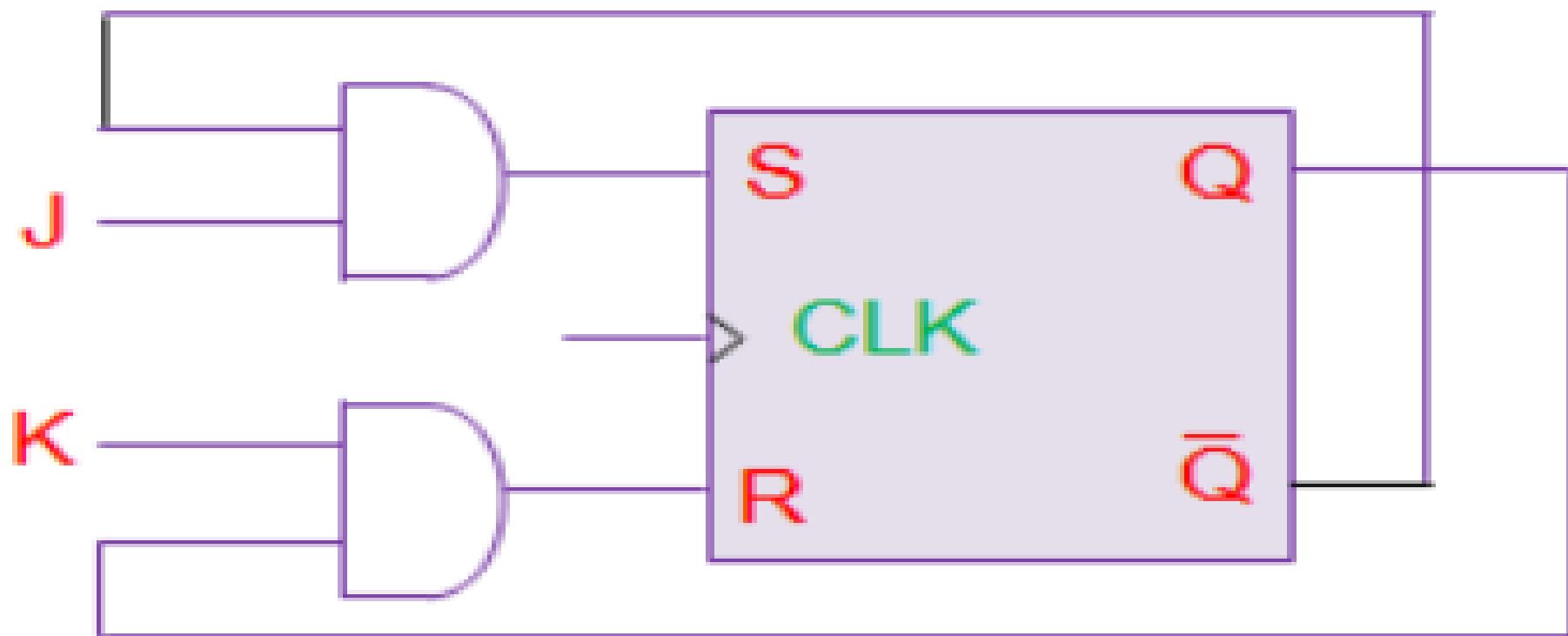
J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

4. K-map Simplification



Continued.....

5. Circuit Design



Continued.....

Conversion of SR Flip Flop to D Flip Flop

1. Truth Table for D Flip Flop

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

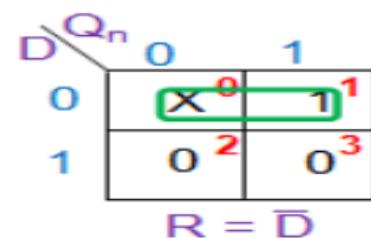
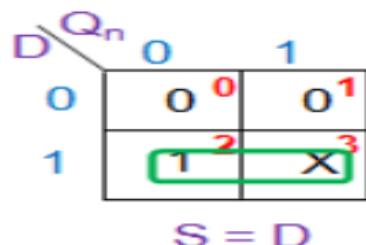
2. Excitation Table for SR Flip Flop

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

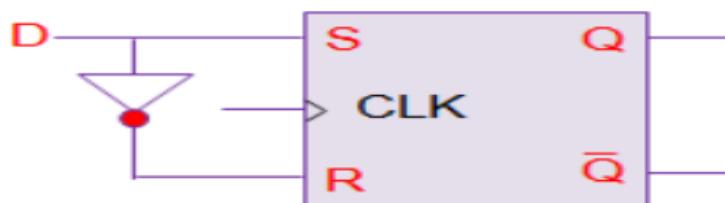
3. Conversion Table

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

4. K-map Simplification



5. Circuit Design



Continued.

Conversion of SR Flip Flop to T Flip Flop

1. Truth Table for T flip-flop

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

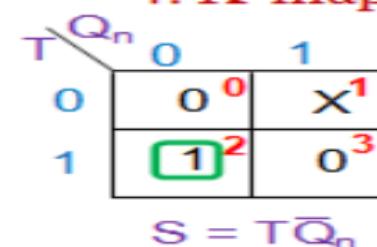
2. Excitation Table for SR flip-flop

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

3. Conversion Table

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

4. K-map Simplification

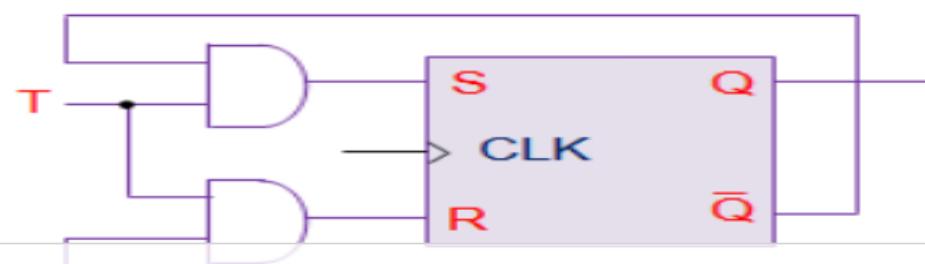


$$S = T\bar{Q}_n$$

$$R = TQ_n$$

$$R = TQ_n$$

5. Circuit Design



Registers & Counters

Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a **Register**. The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit** word.

The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

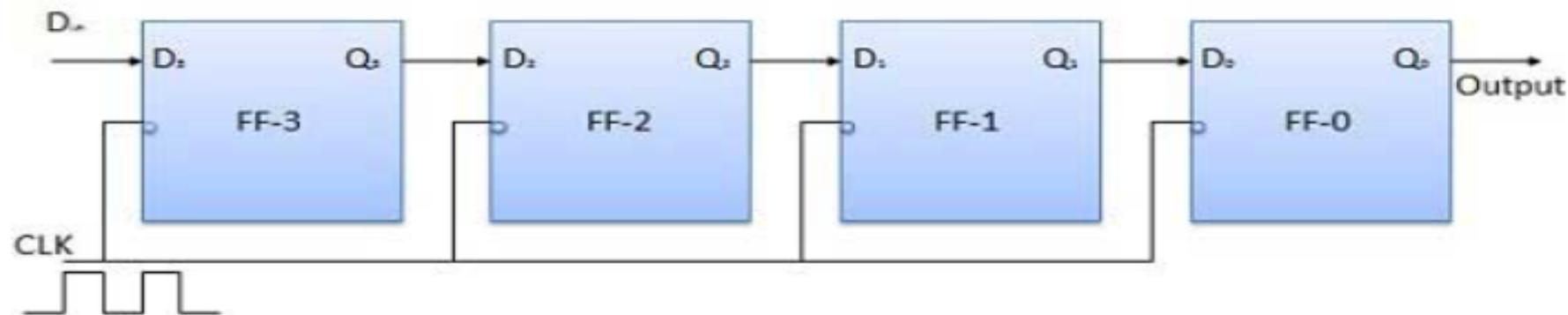
- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

Continued.....

Serial Input Serial Output

Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to D_{in} bit with the LSB bit applied first. The D input of FF-3 i.e. D_3 is connected to serial data input D_{in} . Output of FF-3 i.e. Q_3 is connected to the input of the next flip-flop i.e. D_2 and so on.

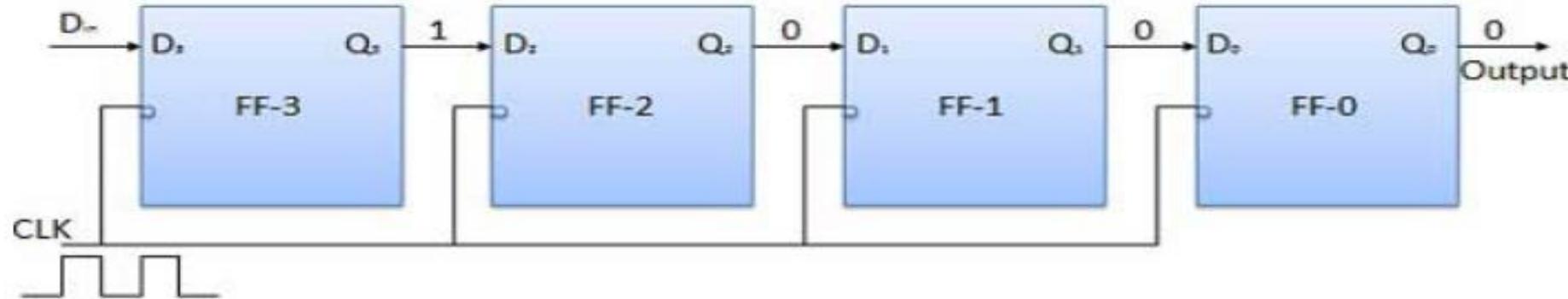
Block Diagram



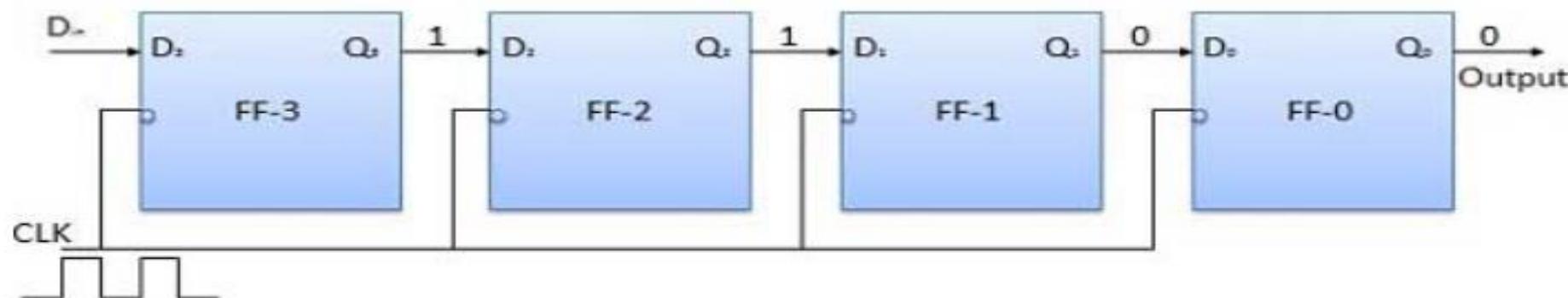
Operation

Before application of clock signal, let $Q_3 Q_2 Q_1 Q_0 = 0000$ and apply LSB bit of the number to be entered to D_{in} . So $D_{in} = D_3 = 1$. Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1000$.

Continued.....

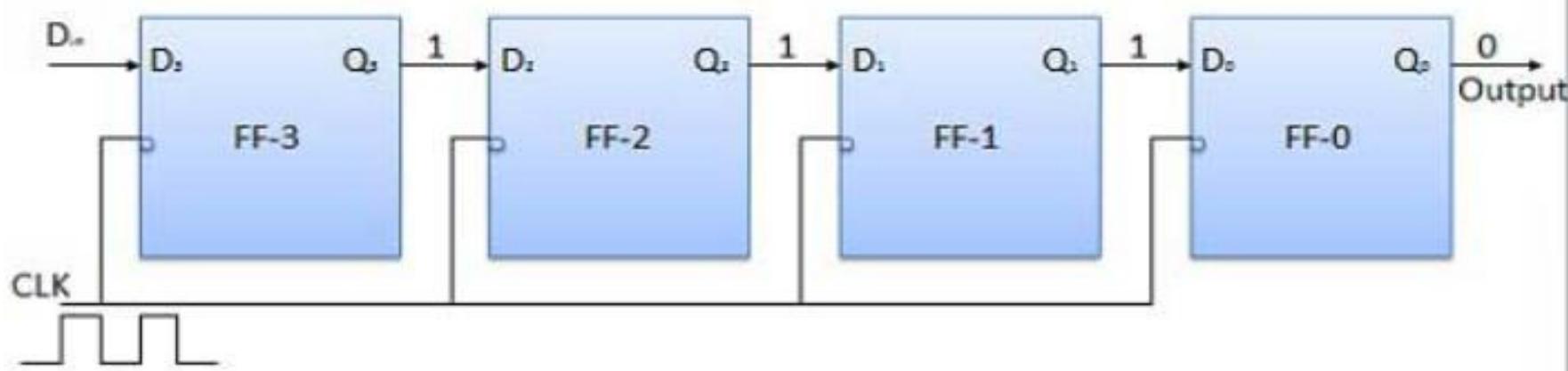


Apply the next bit to D_{in} . So $D_{in} = 1$. As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to $Q_3 Q_2 Q_1 Q_0 = 1100$.

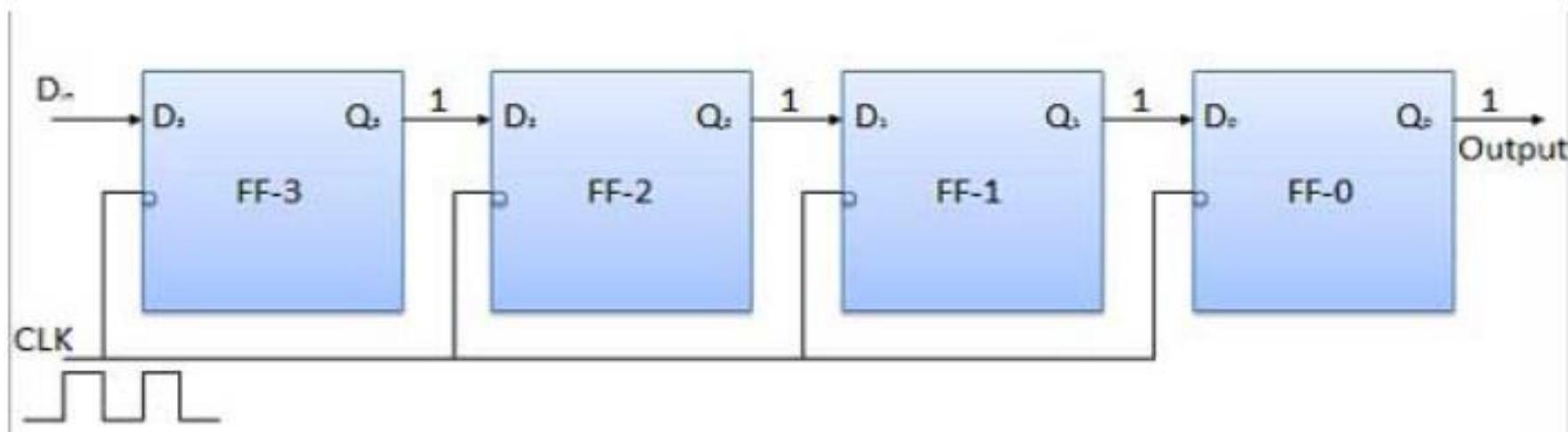


Apply the next bit to be stored i.e. 1 to D_{in} . Apply the clock pulse. As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to $Q_3 Q_2 Q_1 Q_0 = 1110$.

Continued.....



Similarly with $D_{in} = 1$ and with the fourth negative clock edge arriving, the stored word in the register is $Q_3 Q_2 Q_1 Q_0 = 1111$.



Continued.....

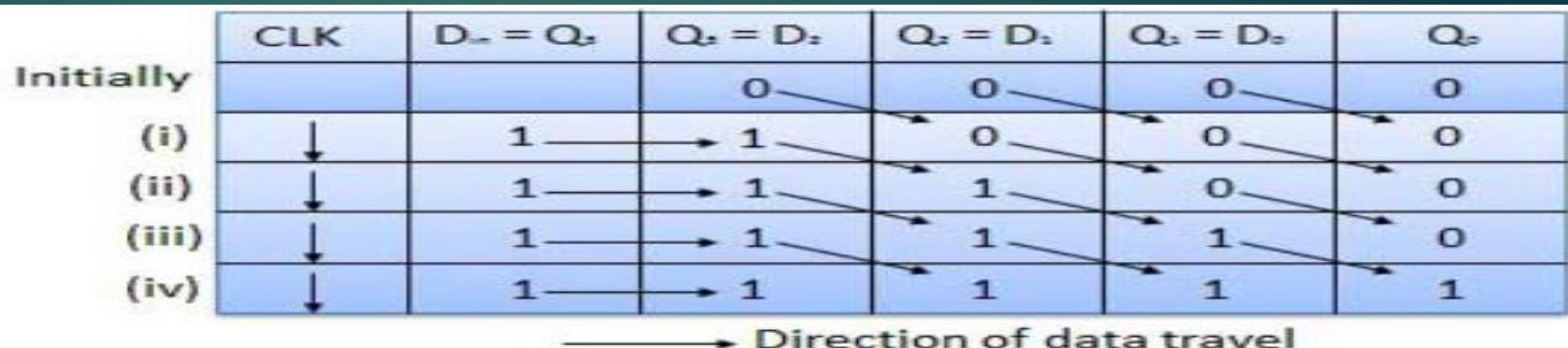
Truth Table

Initially

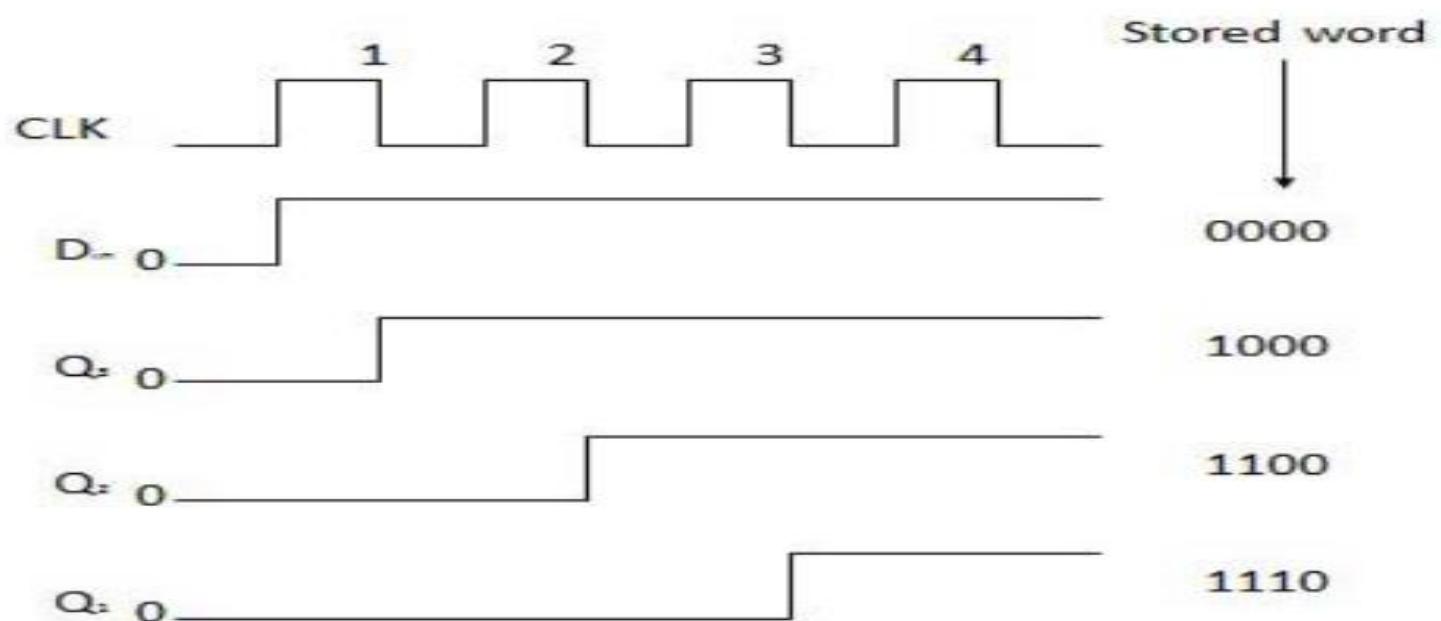
	CLK	$D_n = Q_0$	$Q_1 = D_0$	$Q_2 = D_1$	$Q_3 = D_2$	Q_4
			0	0	0	0
(i)	↓	1 → 1	0	0	0	0
(ii)	↓	1 → 1	1	0	0	0
(iii)	↓	1 → 1	1	1	1	0
(iv)	↓	1 → 1	1	1	1	1

→ Direction of data travel

Continued.....

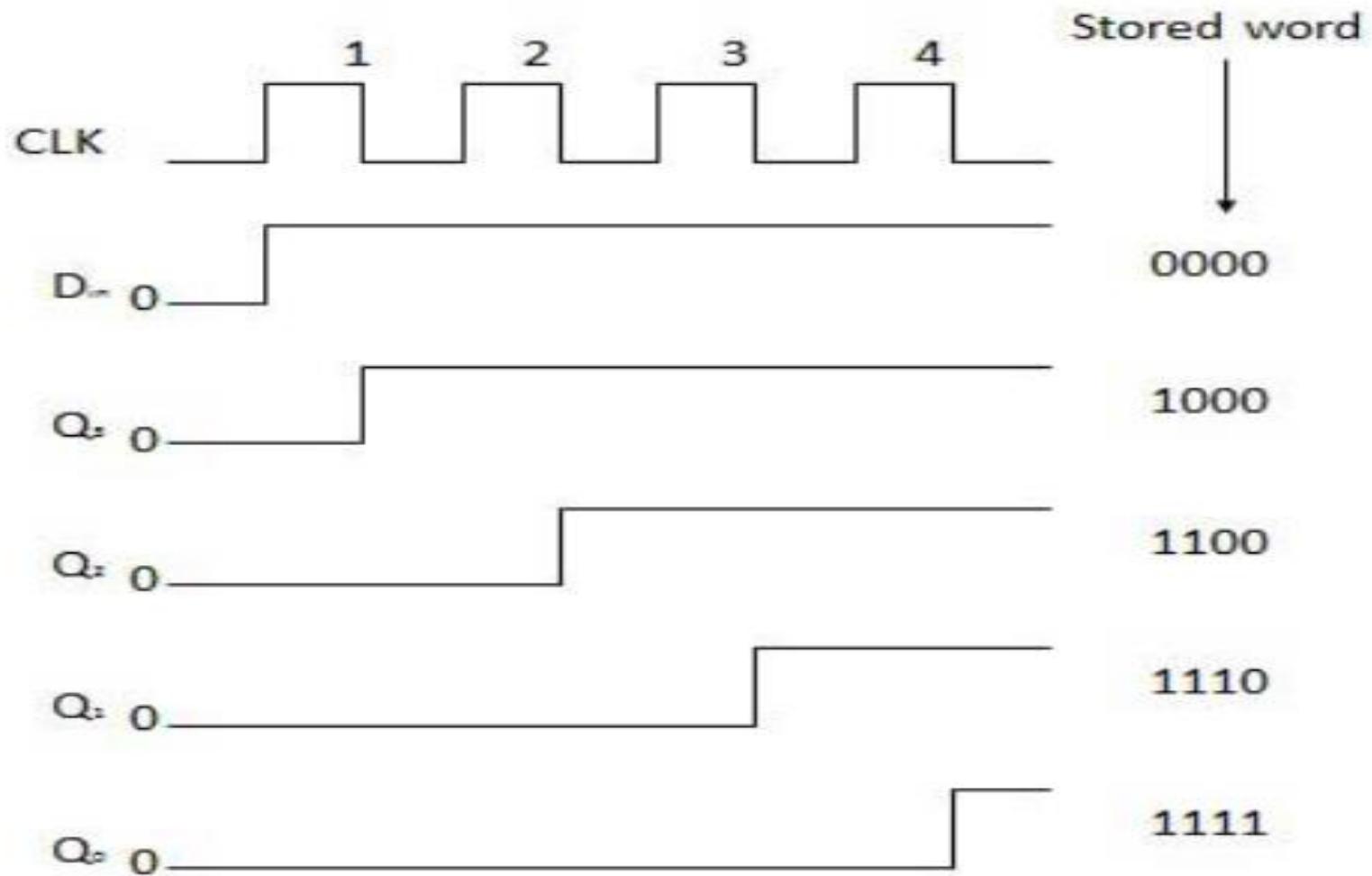


Waveforms



Continued.....

Waveforms

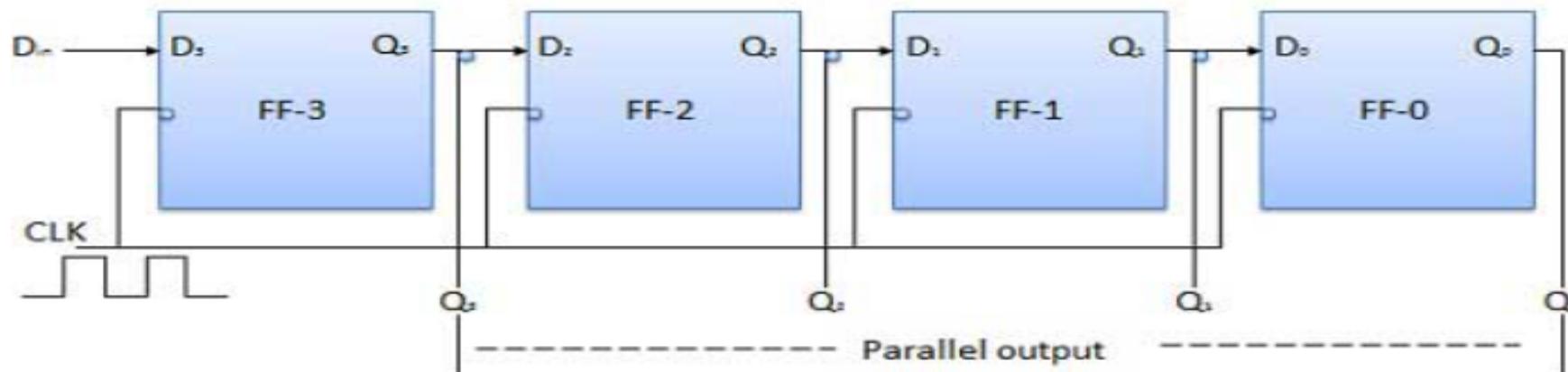


Continued.....

Serial Input Parallel Output

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

Block Diagram



Continued.....

Parallel Input Serial Output (PISO)

- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied through the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

Load mode

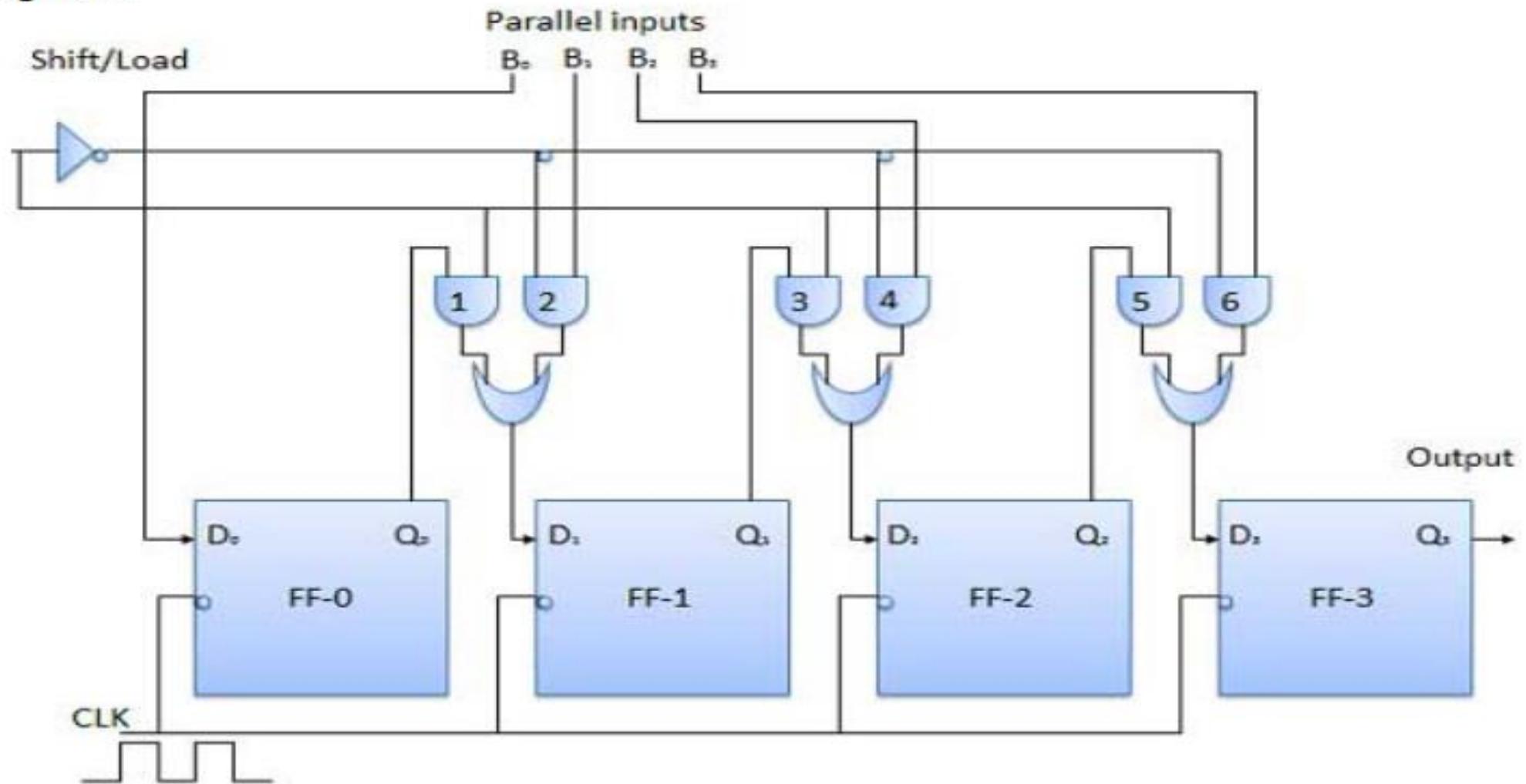
When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

Continued.....

Block Diagram

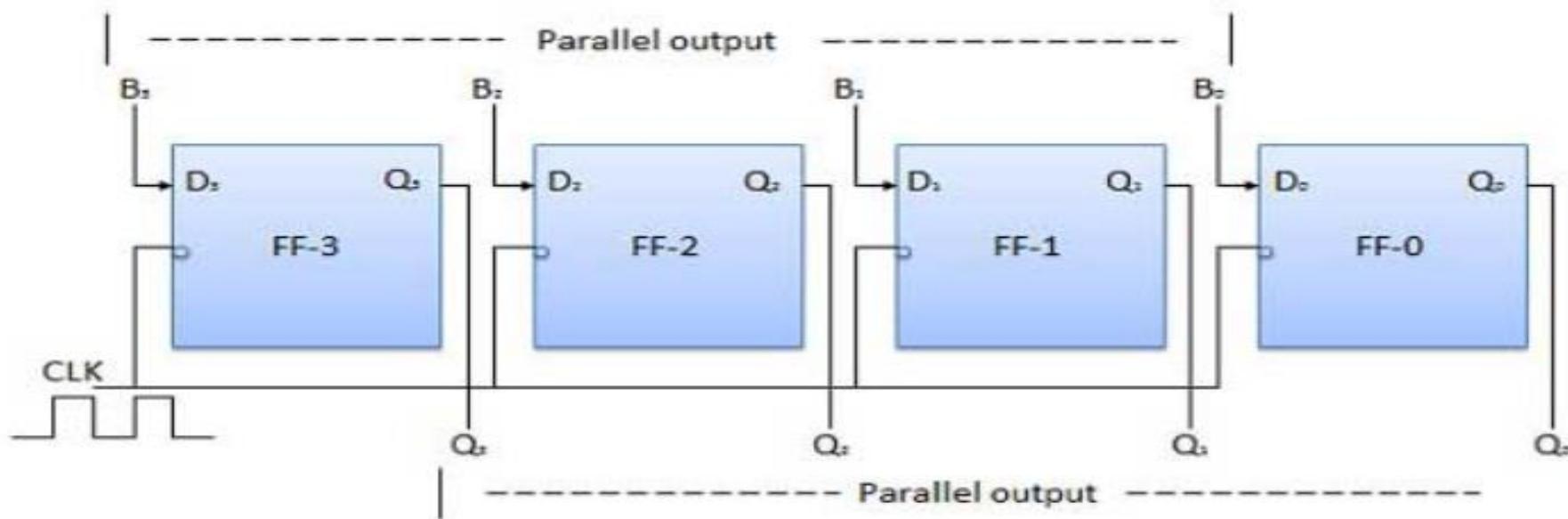


Continued.....

Parallel Input Parallel Output (PIPO)

In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.

Block Diagram

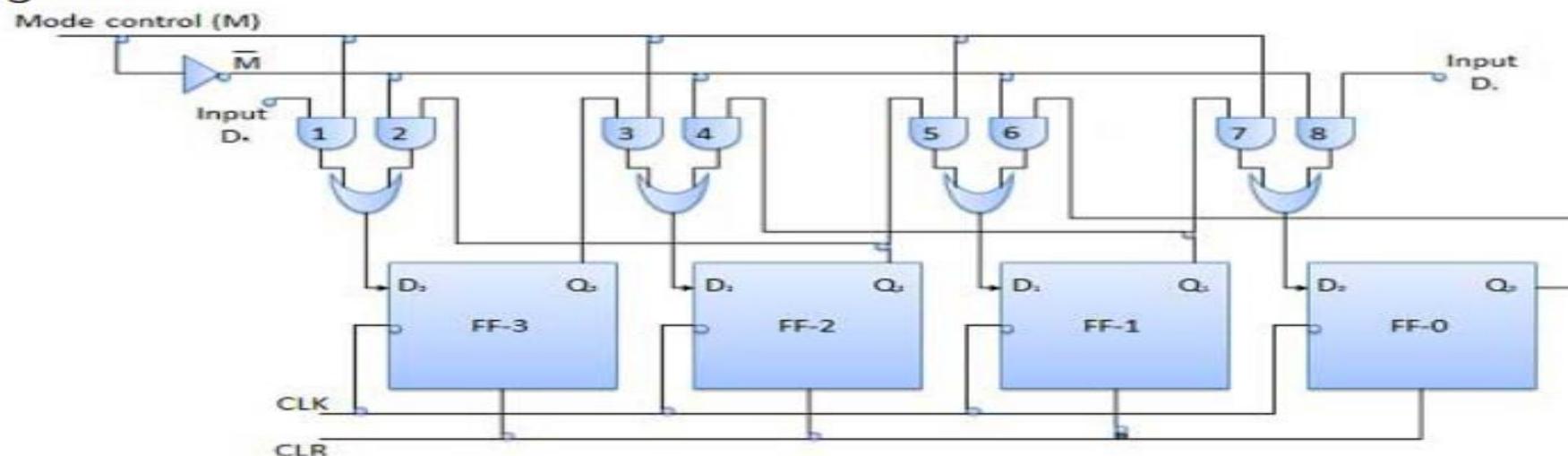


Continued.....

Bidirectional Shift Register

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

Block Diagram



Continued.....

Operation

S.N.	Condition	Operation
1	With $M = 1$ – Shift right operation	<p>If $M = 1$, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled.</p> <p>The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with $M = 1$ we get the serial right shift operation.</p>
2	With $M = 0$ – Shift left operation	<p>When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled.</p> <p>The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with $M = 0$ we get the serial right shift operation.</p>

Continued.....

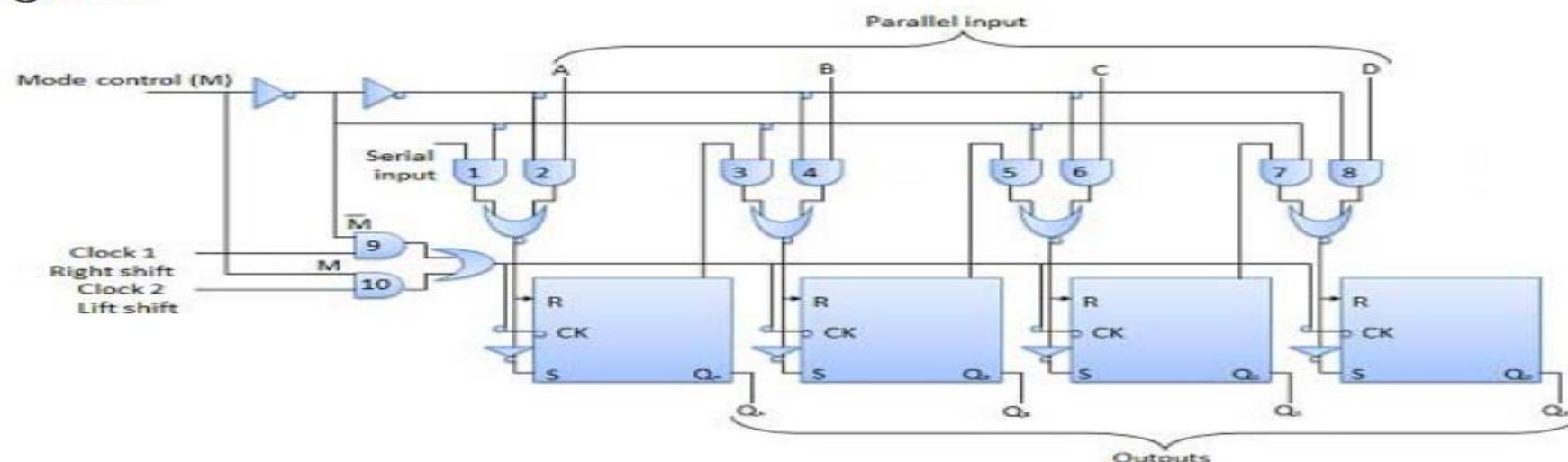
Universal Shift Register

A shift register which can shift the data in only one direction is called a uni-directional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallelly, is known as a universal shift register. The shift register is capable of performing the following operation –

- Parallel loading
- Left Shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

Block Diagram



Continued.....

Parallel Input Serial Output (PISO)

- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied through the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

Load mode

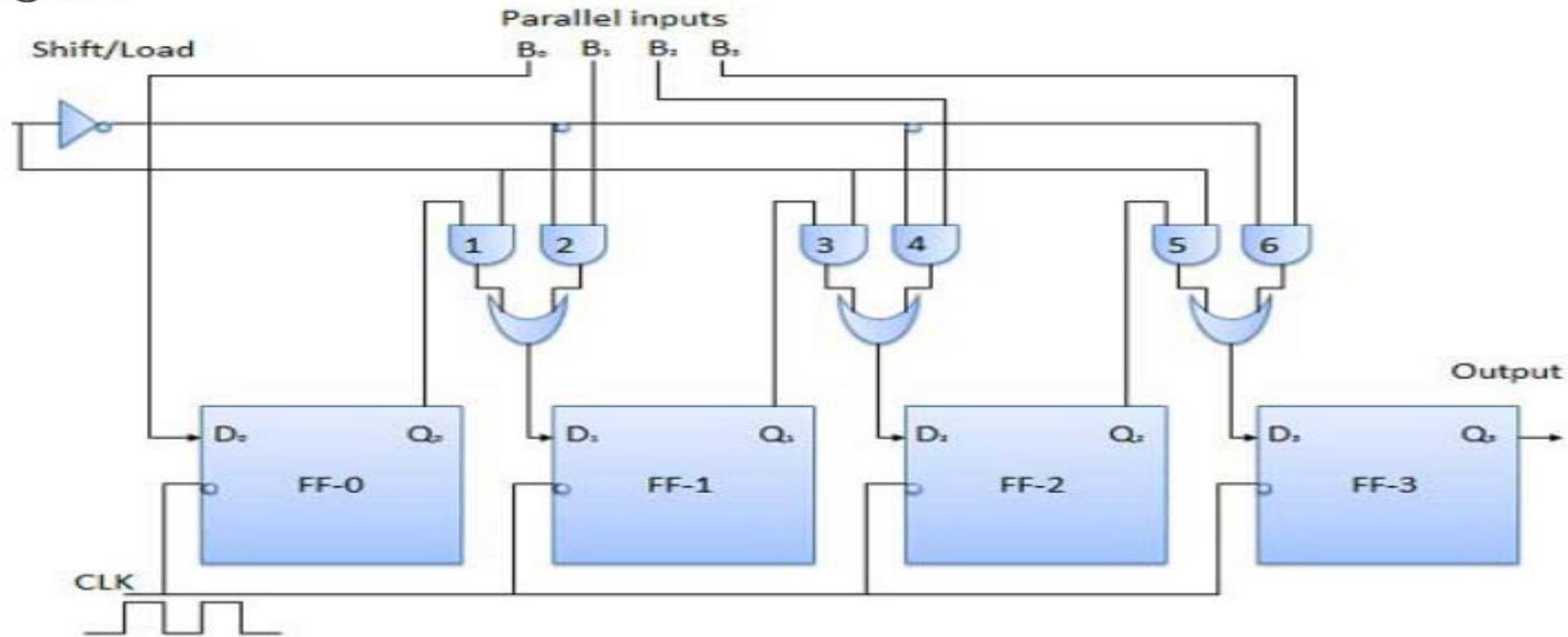
When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

Continued.....

Block Diagram



Continued.....

Modulus Counter (MOD-N Counter)

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n-bit ripple counter is called as modulo-N counter. Where, MOD number = 2^n .

Type of modulus

- 2-bit up or down (MOD-4)
- 3-bit up or down (MOD-8)
- 4-bit up or down (MOD-16)

Application of counters

- Frequency counters
- Digital clock
- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator.

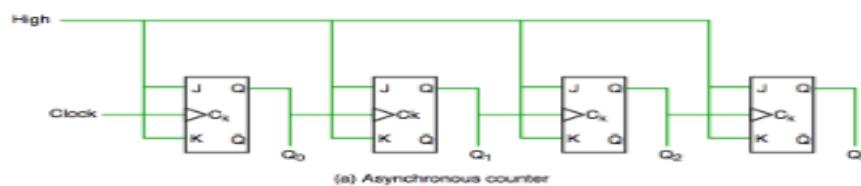
Continued.....

Counters are broadly divided into two categories

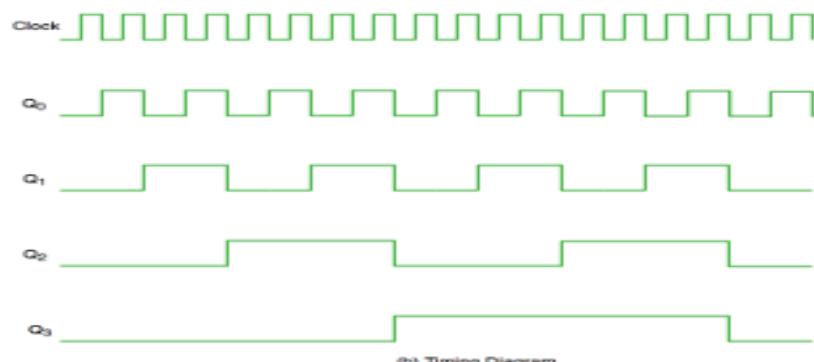
1. Asynchronous counter
2. Synchronous counter

1. Asynchronous Counter

In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops. We can understand it by following diagram-



(a) Asynchronous counter



(b) Timing Diagram

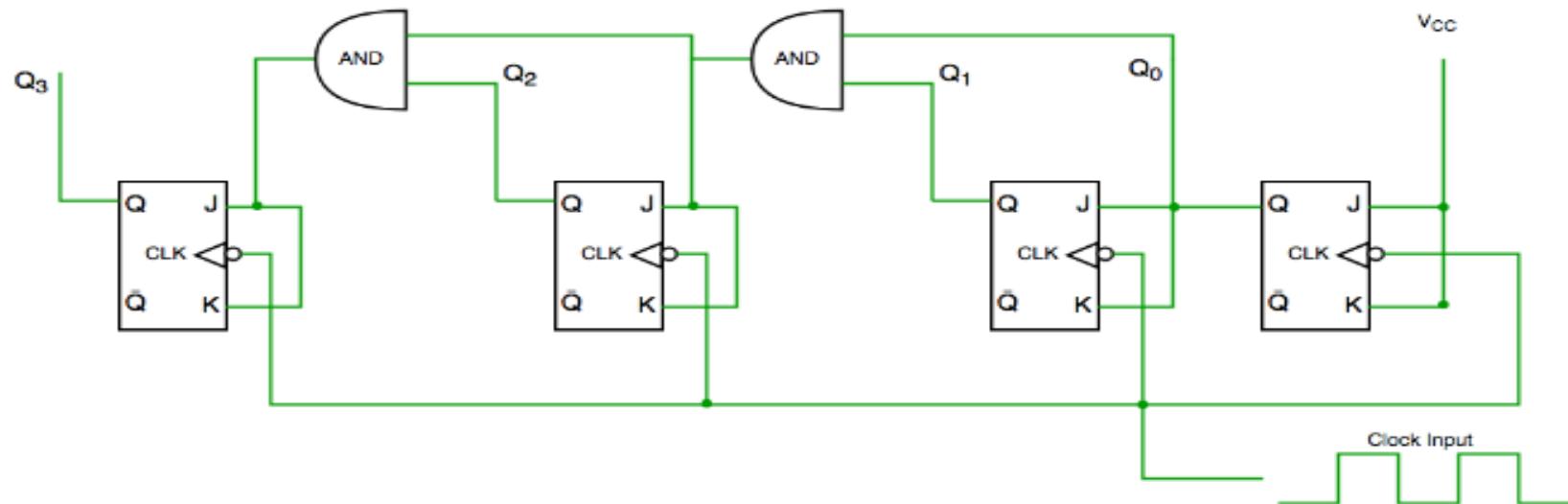
Continued.....

It is evident from timing diagram that Q0 is changing as soon as the rising edge of clock pulse is encountered, Q1 is changing when rising edge of Q0 is encountered (because Q0 is like clock pulse for second flip flop) and so on. In this way ripples are generated through Q0, Q1, Q2, Q3 hence it is also called **RIPPLE counter**. A ripple counter is a cascaded arrangement of flip flops where the output of one flip flop drives the clock input of the following flip flop

Continued.....

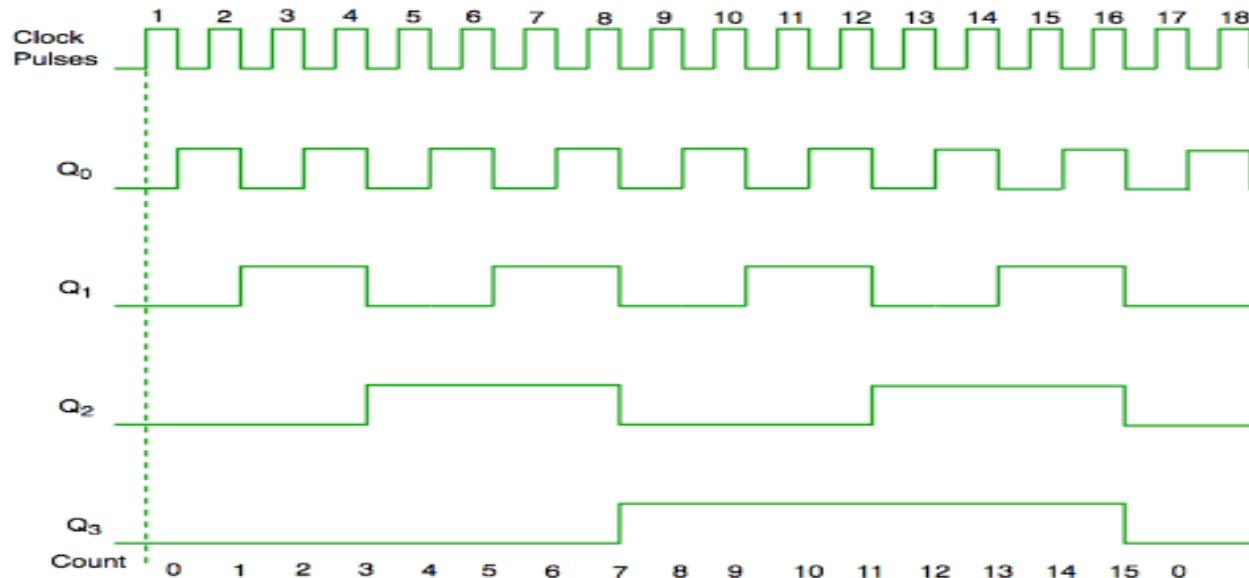
2. Synchronous Counter

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop.



Continued.....

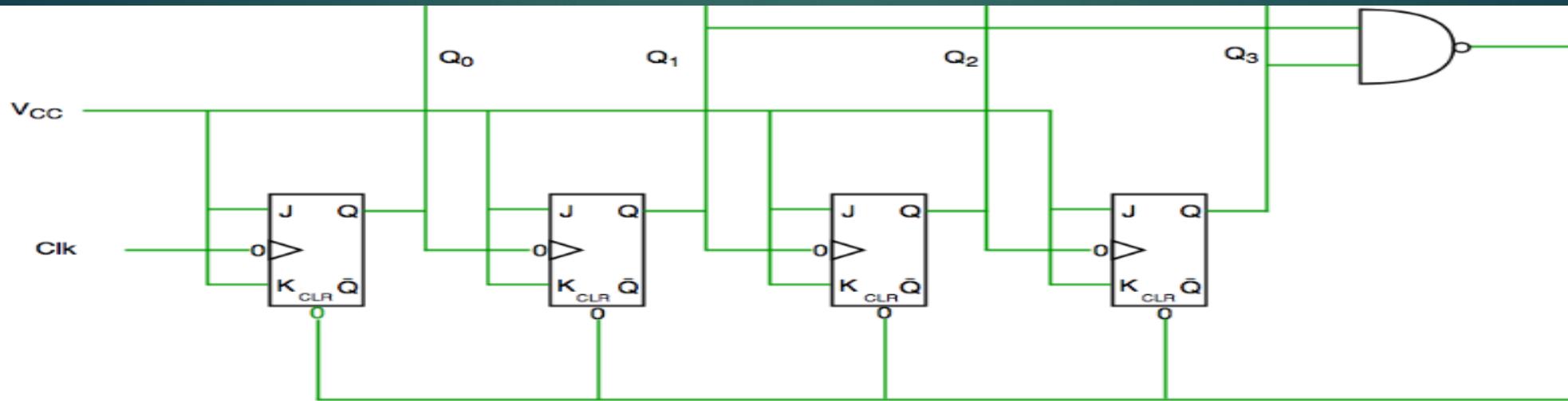
Synchronous counter circuit



Timing diagram synchronous counter

From circuit diagram we see that Q0 bit gives response to each falling edge of clock while Q1 is dependent on Q0, Q2 is dependent on Q1 and Q0 , Q3 is dependent on Q2,Q1 and Q0.

Continued.....



Decade counter circuit diagram

We see from circuit diagram that we have used nand gate for Q₃ and Q₁ and feeding this to clear input line because binary representation of 10 is—

1010

And we see Q₃ and Q₁ are 1 here, if we give NAND of these two bits to clear input then counter will be clear at 10 and again start from beginning.

Important point: Number of flip flops used in counter are always greater than equal to $(\log_2 n)$ where n=number of states in counter.

Continued.....

Decade Counter

A decade counter counts ten different states and then reset to its initial states. A simple decade counter will count from 0 to 9 but we can also make the decade counters which can go through any ten states between 0 to 15(for 4 bit counter).

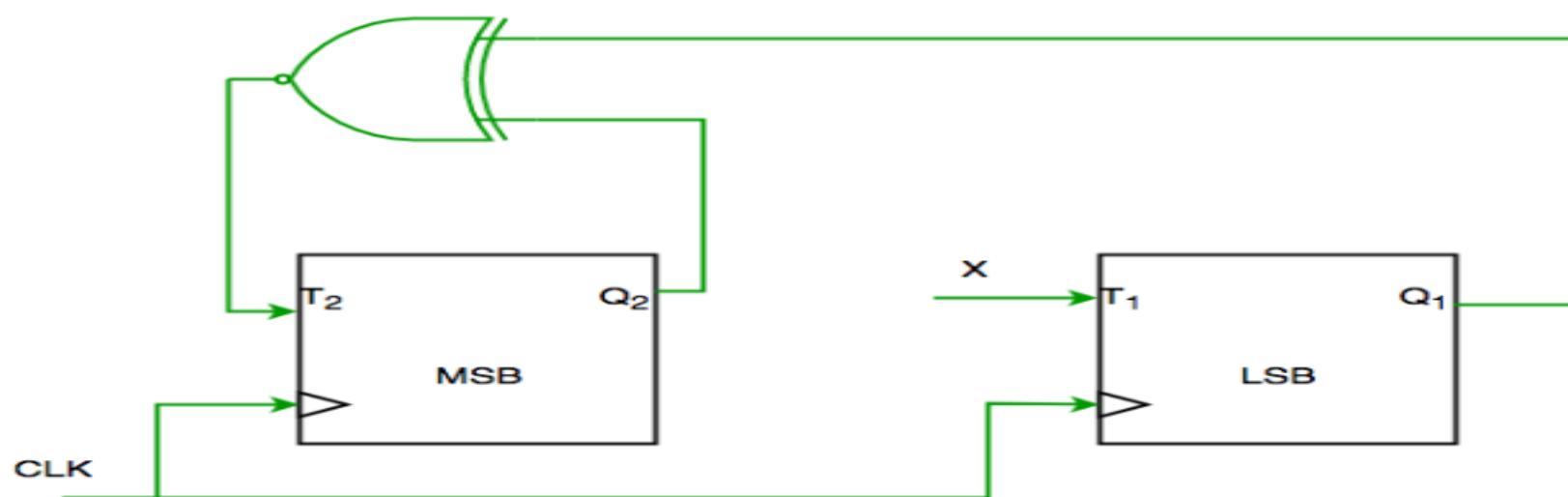
Clock pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1

Continued.....

Clock pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Continued.....

Q1. Consider the partial implementation of a 2-bit counter using T flip-flops following the sequence 0-2-3-1-0, as shown below



To complete the circuit, the input X should be

- (A) Q_2 ?
- (B) $Q_2 + Q_1$
- (C) $(Q_1 ? Q_2)'$
- (D) $Q_1 ? Q_2$

(GATE-CS-2004)

Continued.....

Solution:

From circuit we see

$$T1 = XQ1' + X'Q1 \text{ -- (1)}$$

AND

$$T2 = (Q2 ? Q1)' \text{ -- (2)}$$

AND DESIRED OUTPUT IS $00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$

SO X SHOULD BE $Q1Q2' + Q1'Q2$ SATISFYING 1 AND 2.

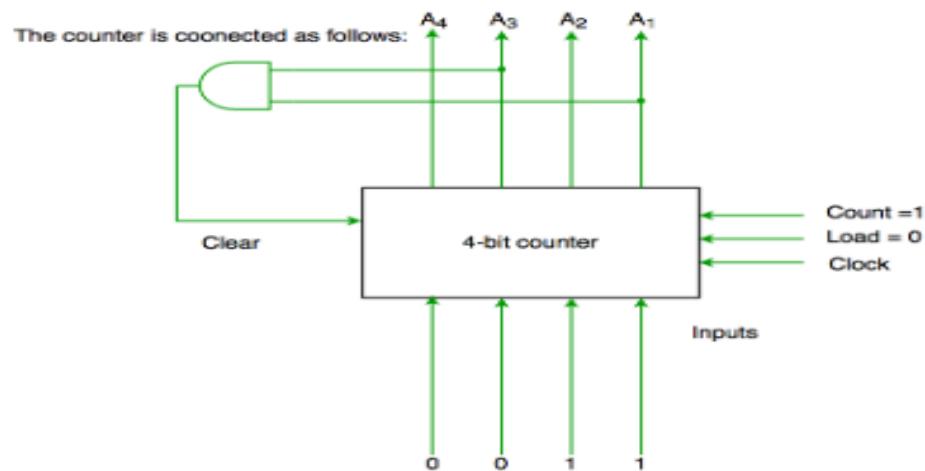
SO ANS IS (D) PART.

Continued.....

Q2. The control signal functions of a 4-bit binary counter are given below (where X is "don't care")

The counter is connected as follows:

Clear	Clock	Load	Count	Function
1	X	X	X	clear to 0
0	X	0	0	No change
0	↑	1	X	Load Input
0	↑	0	1	Count next



Assume that the counter and gate delays are negligible. If the counter starts at 0, then it cycles through the following sequence:

Continued.....

Assume that the counter and gate delays are negligible. If the counter starts at 0, then it cycles through the following sequence:

- (A) 0,3,4
- (B) 0,3,4,5
- (C) 0,1,2,3,4
- (D) 0,1,2,3,4,5

(GATE-CS-2007)

Solution:

Initially A1 A2 A3 A4 =0000

Clr=A1 and A3

So when A1 and A3 both are 1 it again goes to 0000

Hence 0000(init.) \rightarrow 0001 (A1 and A3=0) \rightarrow 0010 (A1 and A3=0) \rightarrow 0011 (A1 and A3=0) \rightarrow 0100 (**A1 and A3=1**) [clear condition satisfied] \rightarrow 0000(init.) so it goes through 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4

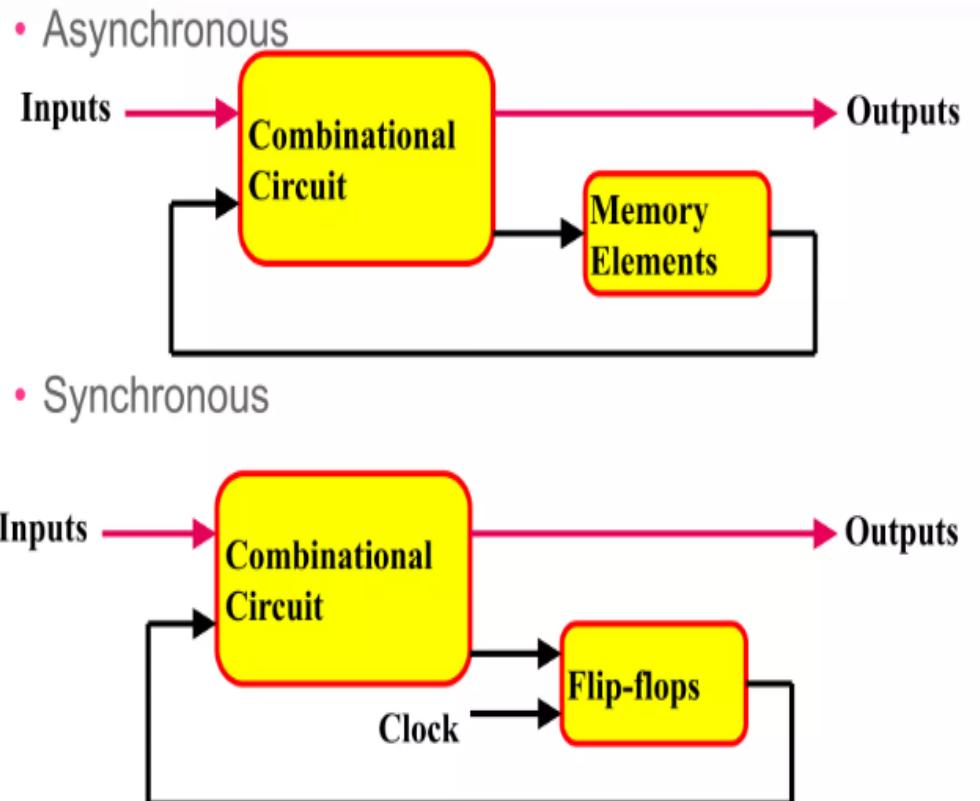
Ans is (C) part.

Continued.....

Continued.....

- There are two types of sequential circuits:
 - ❖ *synchronous*: outputs change only at specific time
 - ❖ *asynchronous*: outputs change at any time
- *Multivibrator*: a class of sequential circuits. They can be:
 - ❖ *bistable* (2 stable states)
 - ❖ *monostable* or *one-shot* (1 stable state)
 - ❖ *astable* (no stable state)
- Bistable logic devices: *latches* and *flip-flops*.
- Latches and flip-flops differ in the method used for changing their state.

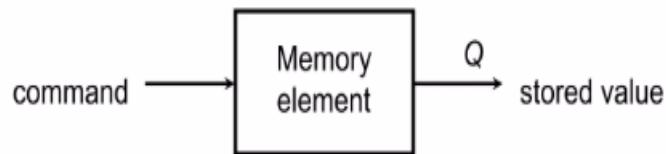
SEQUENTIAL CIRCUITS



Continued.....

MEMORY ELEMENTS

- **Memory element:** a device which can remember value indefinitely, or change value on command from its inputs.



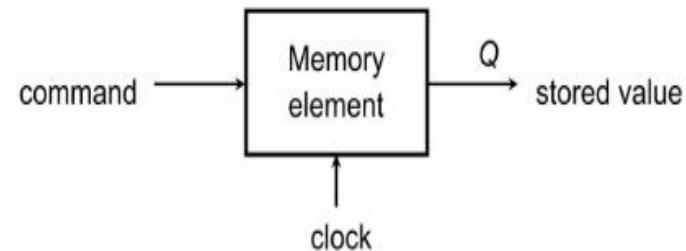
- **Characteristic table:**

Command (at time t)	$Q(t)$	$Q(t+1)$
Set	X	1
Reset	X	0
Memorise / No Change	0	0
1	1	1

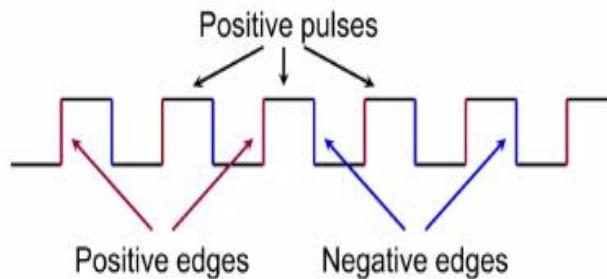
$Q(t)$: current state
 $Q(t+1)$ or Q^+ : next state

MEMORY ELEMENTS

- Memory element with clock. Flip-flops are memory elements that change state on clock signals.



- Clock is usually a square wave.



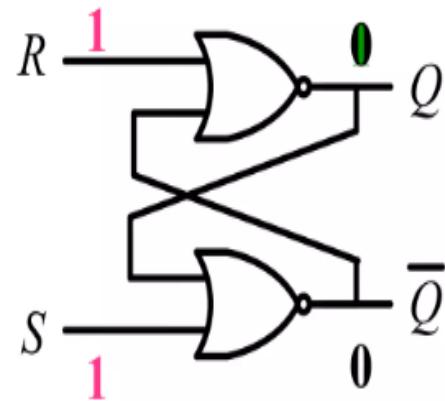
Continued.....

MEMORY ELEMENTS

- Two types of triggering/activation:
 - ❖ pulse-triggered
 - ❖ edge-triggered
- Pulse-triggered
 - ❖ latches
 - ❖ ON = 1, OFF = 0
- Edge-triggered
 - ❖ flip-flops
 - ❖ positive edge-triggered (ON = from 0 to 1; OFF = other time)
 - ❖ negative edge-triggered (ON = from 1 to 0; OFF = other time)

LATCHES

• SR Latch



S	R	Q_0	Q	Q'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

$Q = Q_0$

$Q = 0$

$Q = 1$

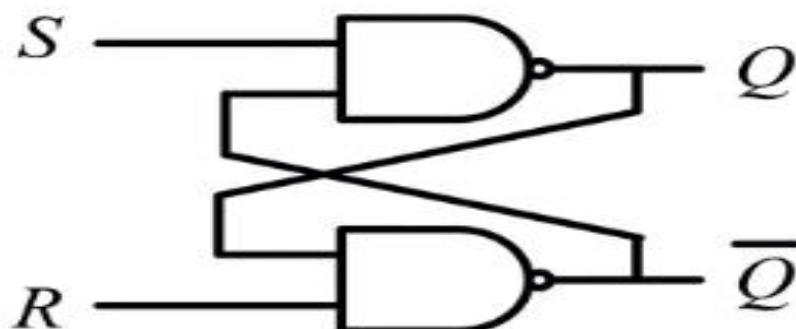
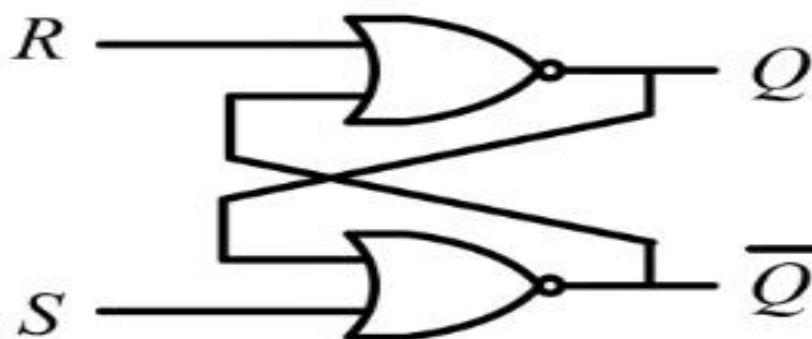
$Q = Q'$

$Q = Q'$

Continued.....

LATCHES

- SR Latch



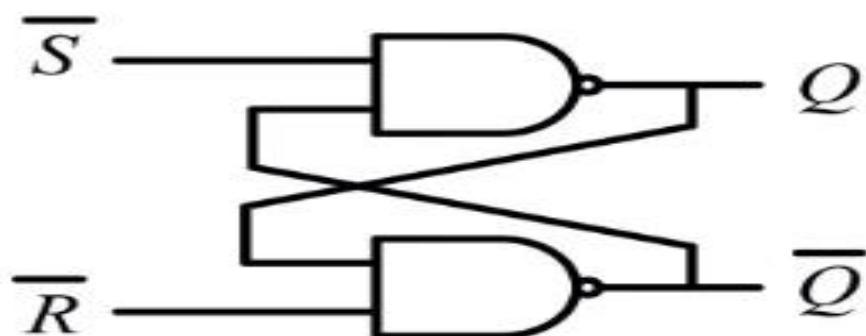
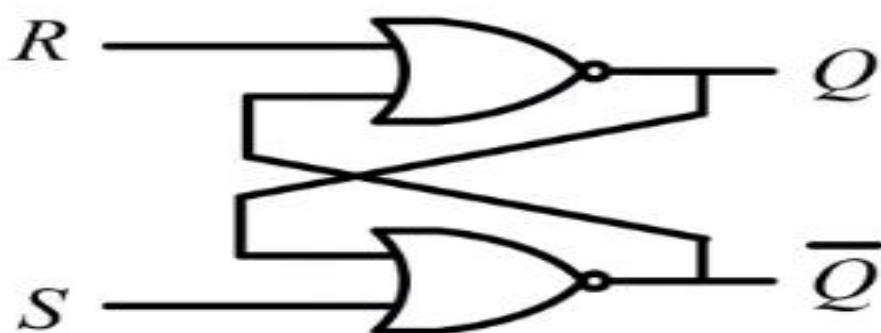
S	R	Q	
0	0	Q_0	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q=Q'=0$	Invalid

S	R	Q	
0	0	$Q=Q'=1$	Invalid
0	1	1	Set
1	0	0	Reset
1	1	Q_0	No change

Continued.....

LATCHES

- SR Latch



S	R	Q
0	0	Q_0
0	1	0
1	0	1
1	1	$Q=Q'=0$

No change
Reset
Set
Invalid

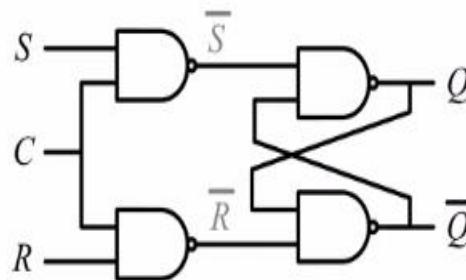
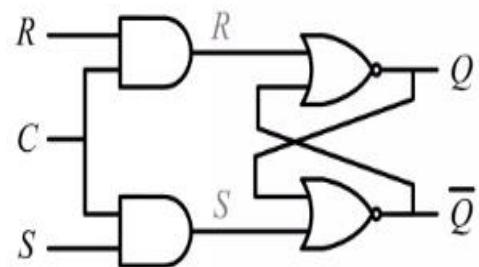
S^*	R^*	Q
0	0	$Q=Q'=1$
0	1	1
1	0	0
1	1	Q_0

Invalid
Set
Reset
No change

Continued.....

CONTROLLED LATCHES

- SR Latch with Control Input

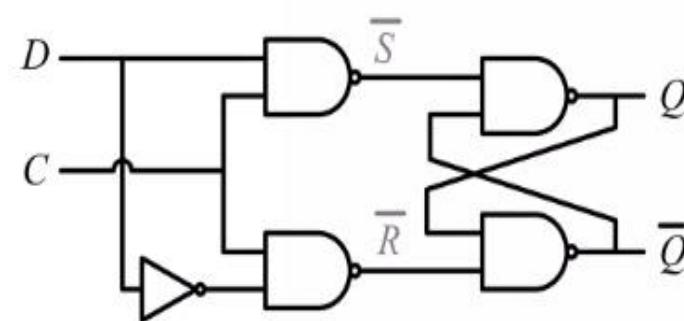


C S R	Q
0 x x	Q_0
1 0 0	Q_0
1 0 1	0
1 1 0	1
1 1 1	$Q=Q'$

No change
No change
Reset
Set
Invalid

CONTROLLED LATCHES

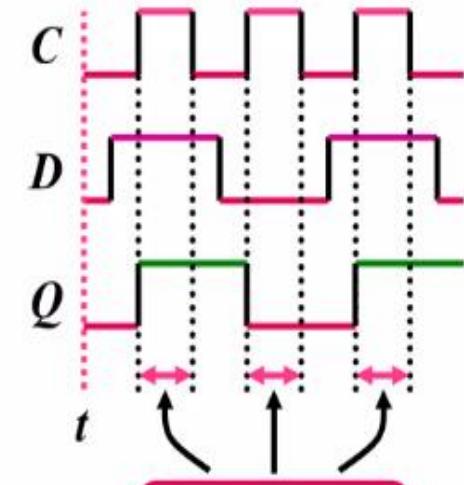
- D Latch ($D = Data$)



C D	Q
0 x	Q_0
1 0	0
1 1	1

No change
Reset
Set

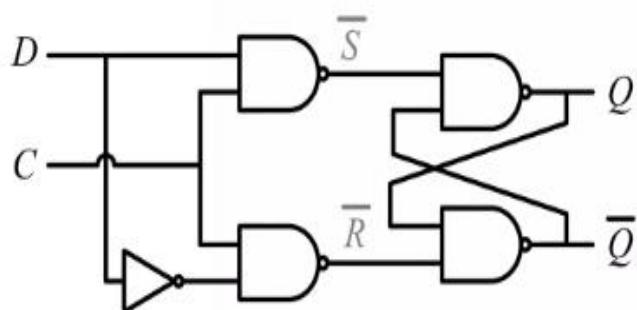
Timing Diagram



Continued.....

CONTROLLED LATCHES

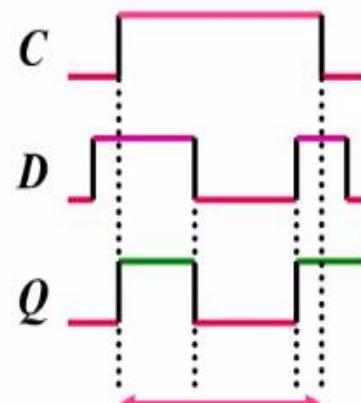
- D Latch ($D = \text{Data}$)



C	D	Q
0	x	Q_0
1	0	0
1	1	1

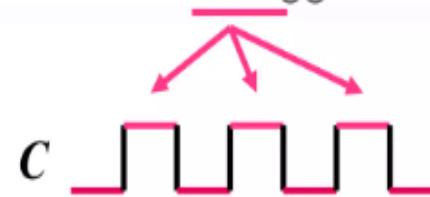
No change Output may change Reset Set

Timing Diagram



FLIP-FLOPS

- Controlled latches are level-triggered



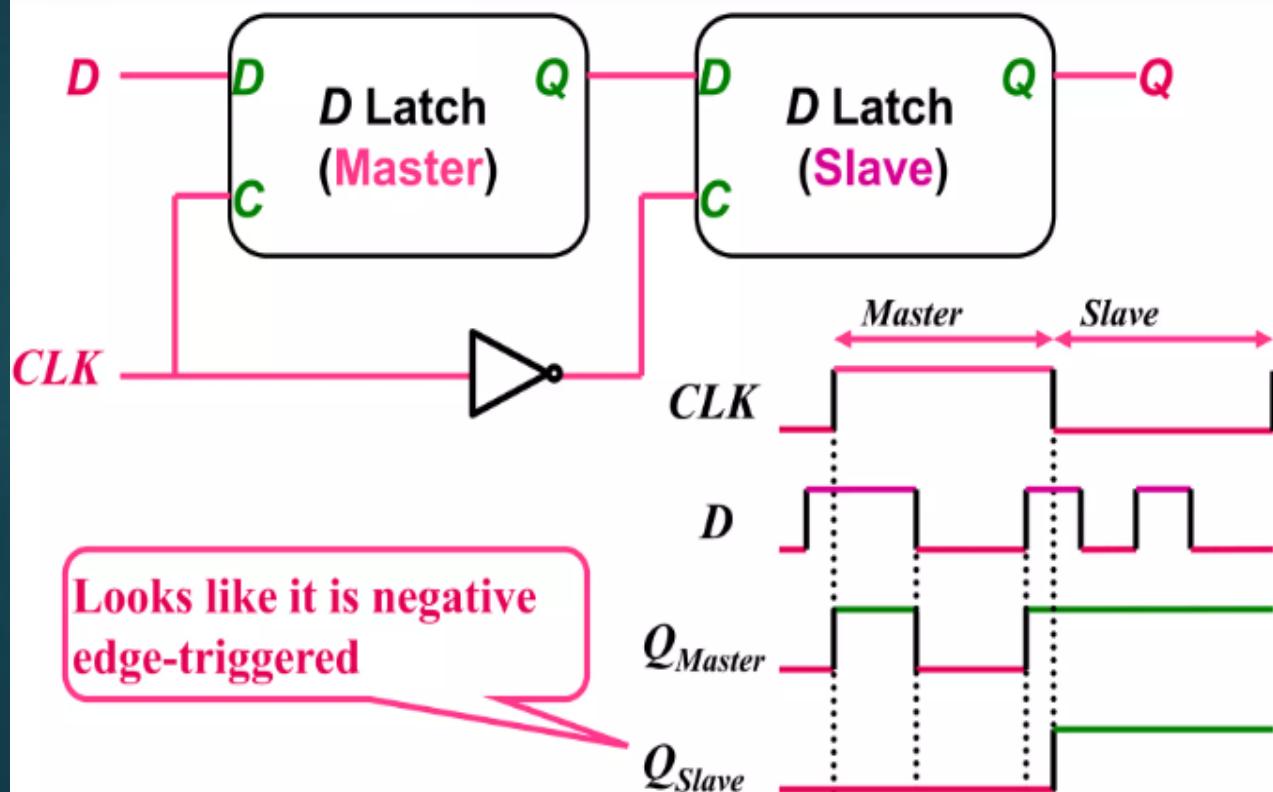
- Flip-Flops are edge-triggered



Continued.....

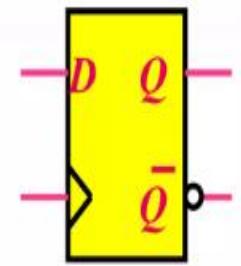
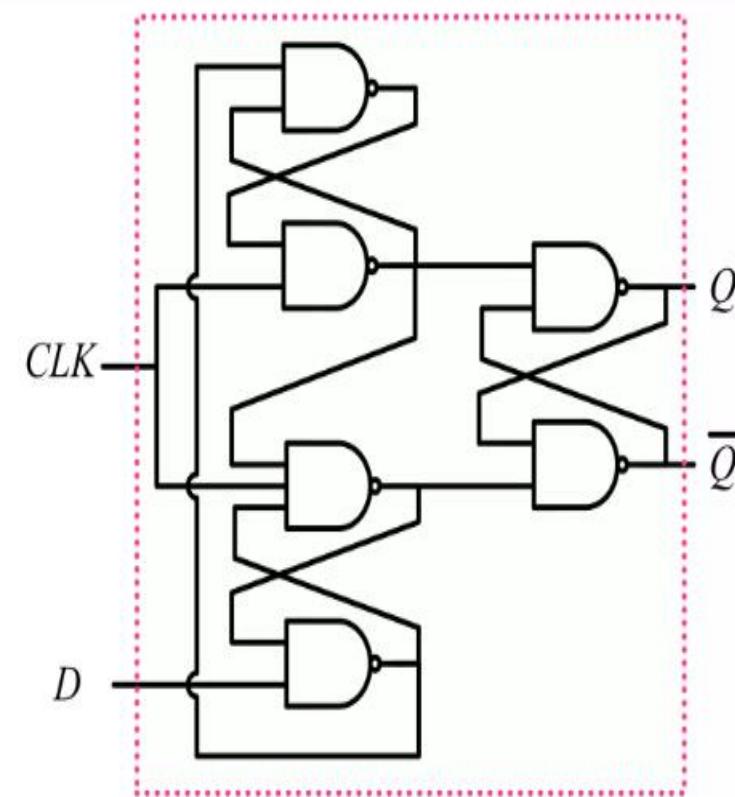
FLIP-FLOPS

- Master-Slave D Flip-Flop

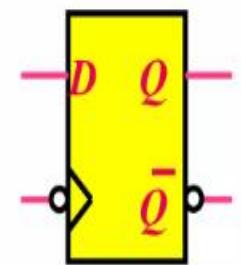


FLIP-FLOPS

- Edge-Triggered D Flip-Flop



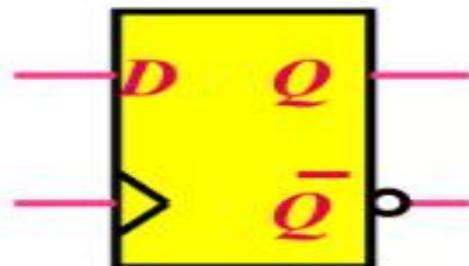
Positive Edge



Negative Edge

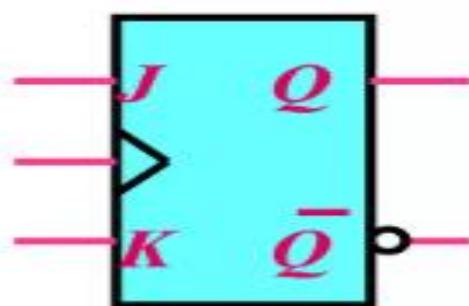
Continued.....

FLIP-FLOP CHARACTERISTIC TABLES



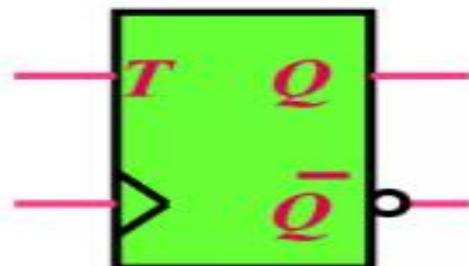
D	$Q(t+1)$
0	0
1	1

Reset
Set



J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

No change
Reset
Set
Toggle

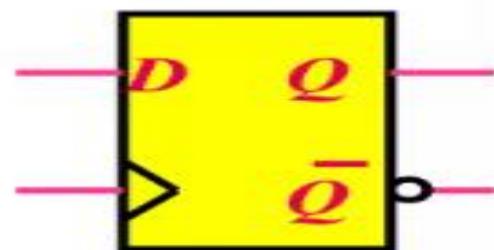


T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

No change
Toggle

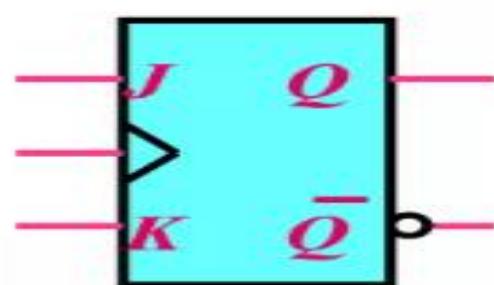
Continued.....

FLIP-FLOP CHARACTERISTIC EQUATIONS



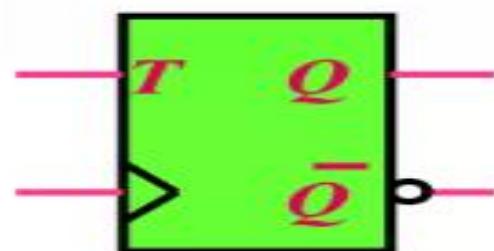
D	$Q(t+1)$
0	0
1	1

$$Q(t+1) = D$$



J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

$$Q(t+1) = JQ' + K'Q$$



T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

$$Q(t+1) = T \oplus Q$$

Continued.....