

## Unit III: Clocks and flip-flops

①

### 1. TTL clock and its characteristics:

→ The heart of every digital system is the system clock. The system clock provides the heartbeat without which the system would cease to function.

i) Clock characteristics

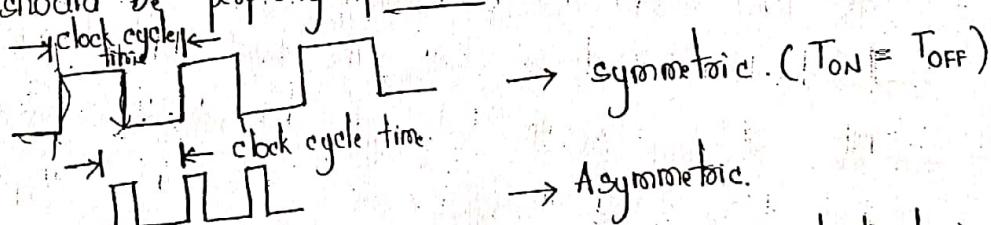
ii) Clock circuits

— Schmitt triggers are used to produce nearly ideal digital signals.

— Monostable is a basic digital timing circuit that is used in a wide variety of timing applications.

Propagation delay is the time required for a signal to pass from the input of a circuit to its output.

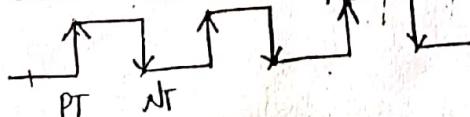
→ The clock should be perfectly periodic, and stable.



→ clock cycle time: is a basic timing interval during which logic operations must be performed.

→ A change of state will occur as the clock transitions from low to high, or as it transitions from high to low.

positive transition  
(clock circuits are called positive edge triggered)      negative transition.



Synchronous operation:

→ All circuits in a digital system are either positive-edge triggered or negative-edge triggered, & thus are synchronized with the system clock.

## → Asynchronous operation

These are few exceptions.

Ex: RESET button when pressed, result in an instant change of state that is not in synchronism with the clock.

Ex: i) What is the clock cycle time for a system that uses  
i) 500-kHz clock  
ii) 8-MHz clock

Sol<sup>10</sup>: i) Cycle time =  $\frac{1}{500 \times 10^3}$

$$= \underline{\underline{1\text{ }\mu\text{s}}}$$

ii) Cycle time =  $\frac{1}{8 \times 10^6}$

$$= \underline{\underline{125\text{ ns}}}$$

Characteristics (of ideal clock)

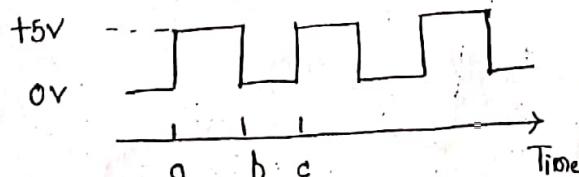
→ The clock levels must be absolutely stable. When the clock is high, the level must hold a steady value of +5V.

→ When the clock is low, the level must be an unchanging 0V.

→ The stability of the clock is much more important than the absolute value of the voltage level.

For ex: +4.8V is acceptable instead of +5.0V, provided it is a steady, unchanging, +4.8V.

a) The second characteristic deals with the time required for the clock levels to change from high to low or vice versa.

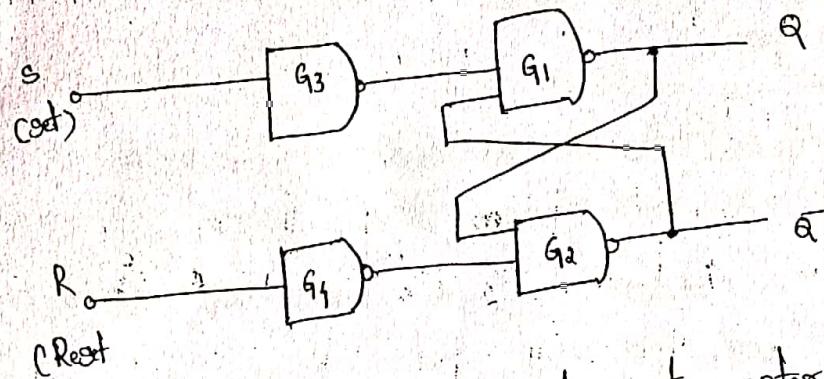


The transition occurs instantaneously - it requires zero time.  
Thus an ideal clock has zero transition time.

## 2. Basic SR Latch using NOR/NAND gates.

(2)

→ The fig. below shows basic SR latch using AND gates.



→ It consists of two inverters to enter the data  $s$  and  $r$  and two AND gates, to produce the outputs  $Q$  and  $\bar{Q}$ .  
→ This circuit works according to the truth table given below

Inputs	Output
$s \quad r$	Entry
$0 \quad 0$	$Q_n$ (No change)
$0 \quad 1$	$0$ (Reset)
$1 \quad 0$	$1$ (Set)
$1 \quad 1$	? (Forbidden)

→ There are two stable states: Set i.e.,  $Q=1$  and reset i.e.,  $Q=0$ .

→ The output  $Q$  and  $\bar{Q}$ , are always complementary.

→ The circuit remains in its set/reset state and hence it is referred to as memory, i.e., it can store 1-bit of digital information.

→ Since the information is locked or latched in this circuit, this circuit is referred to as latch.

→ When  $s=0, r=0, Q = \text{remain}$  in the same state,  
when  $s=0, r=1, Q=0$  (Reset) and  $\bar{Q}=1$   
when  $s=1, r=0, Q=1$  (Set) and  $\bar{Q}=0$

→ When  $S=1$ ,  $R=1$ ,  $Q$  and  $\bar{Q}$  try to become 1, which is not allowed and therefore, this input condition is prohibited.

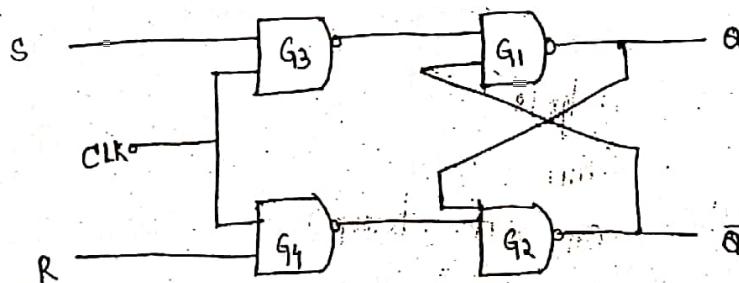
→ The state of latch is referred to as forbidden state.

### 3. Clocked SR flip-flop:

→ It is often required to set or reset the memory cell in synchronism with a train of pulses, known as clock.

→ Such a circuit is shown in fig. below and is referred to as clocked set-reset flip flop.

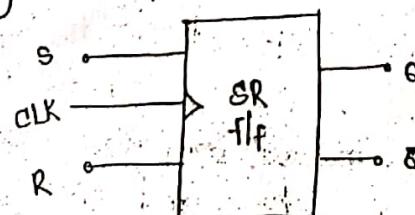
→



→ The truth table for SR flip flop is as shown below

Inputs	Output		
CLK	S	R	$Q_{nt}$
0	x	x	Qn (No change)
1	0	0	Qn (No change)
1	0	1	0 (Reset)
1	1	0	1 (Set)
1	1	1	? (Forbidden)

→ Logic symbol of clocked SR flip flop is as shown below



#### 4. "Preset and clear"

(3)

- In flip flop, when the power is switched on, the state of the circuit is uncertain. It may come to set ( $Q=1$ ) or reset ( $Q=0$ ) state.
- In many applications it is desired to initially set or reset the flip-flop i.e., the initial state of the flip flop is to be assigned.
- This is accomplished by using the direct or asynchronous inputs, referred to as preset ( $P_{\text{S}}$ ) and clear ( $C_{\text{R}}$ ) inputs.
- These inputs may be applied at any time between clock pulses and are not in synchronism with the clock.
- An SR flip flop with preset and clear is shown in fig. below

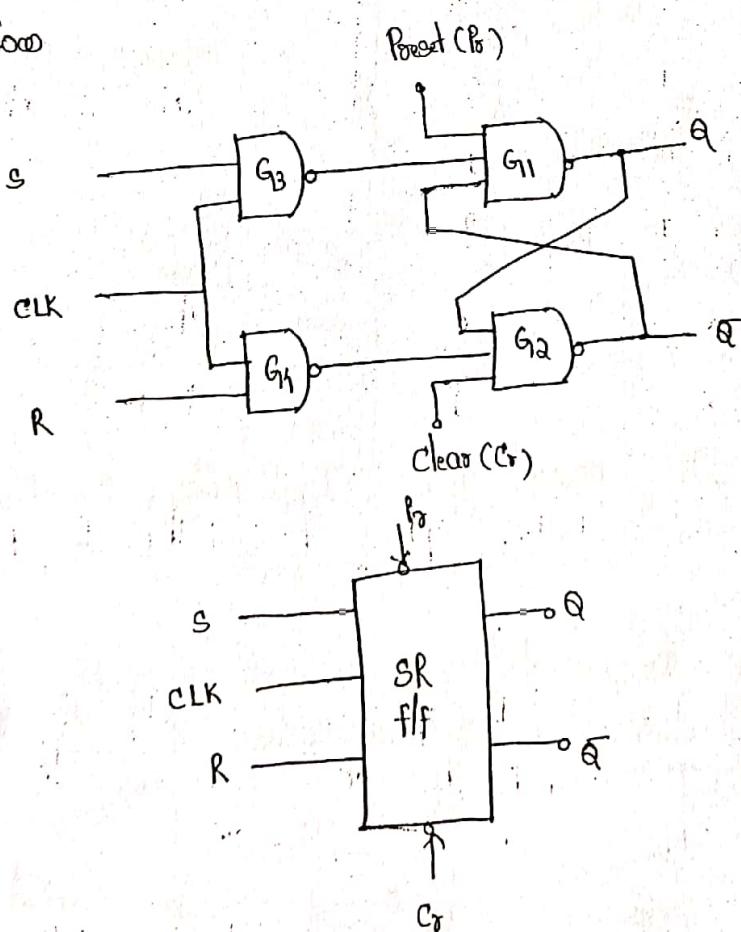


Fig. Logic symbol

- 1) If  $P_S = C_S = 1$ , circuit operates in accordance with truth table of SR flip-flop.
- 2) If  $P_S = 0, C_S = 1$ ; sets the flip-flop.
- 3) If  $P_S = 1, C_S = 0$ ; clears the flip-flop.
- 4) If  $P_S = 0, C_S = 0$ ; must not be used, since this leads to an uncertain state.

Note:  $P_S$  and  $C_S$  are active low inputs which means intended function is performed when the signal applied to  $P_S$  or  $C_S$  is low.

→ Operation of SR flip-flop is summarized as below:

Inputs				Output	Operation performed
CLK	$C_S$	$P_S$	S R	Qn	
1	1	1	0 0	0	Normal flip-flop
1	1	1	0 1		
1	1	1	1 0		
1	1	1	1 1	?	Uncertain
x	0	1	x x		
x	1	0	x x	1	Set
x	1	0	x x	0	Clear

5) JK flip-flop (JK - named after Jack Kilby, the Texas instruments engineer)

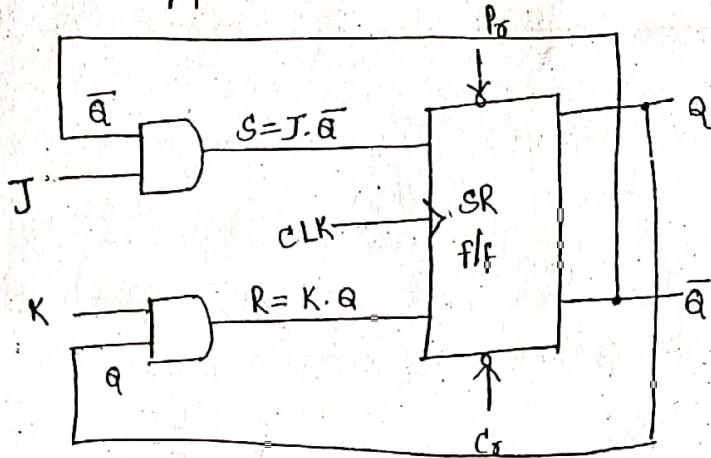
→ The uncertainty in the state of an SR flip-flop when  $S=R=1$  can be eliminated by converting it into a JK flip-flop.

→ The data inputs are J and K which are ANDed with  $\bar{Q}$  and Q respectively to obtain SR inputs, i.e.,

$$S = JQ$$

$$R = K \cdot Q$$

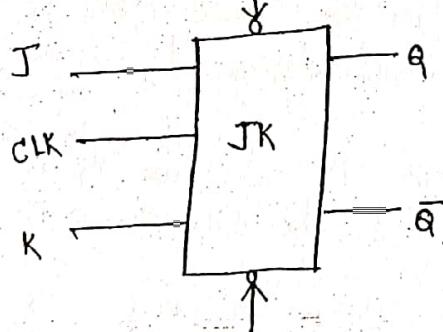
→ A JK flip flop thus obtained is shown below



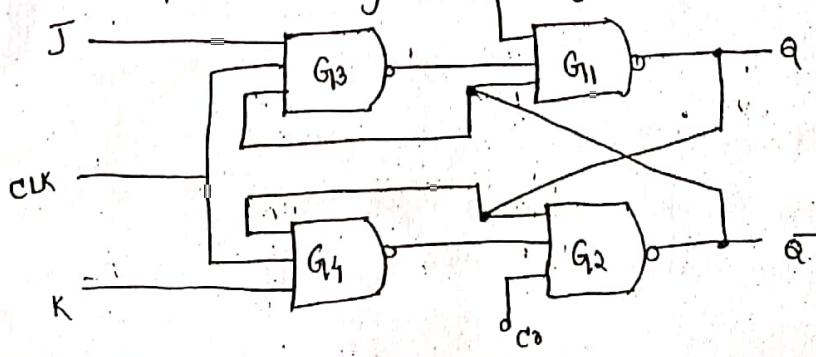
→ Truth table of JK flip flop is as shown below

Inputs		Output
J	K	Q <sub>n+1</sub>
0	0	Q <sub>n</sub> (No change)
0	1	0 (Reset)
1	0	1 (Set)
1	1	Q̄ <sub>n</sub> (Toggle)

→ Logic symbol of JK flip flop is as shown below



→ JK flip-flop using AND gates:



## The race-around condition:

- The difficulty of both inputs  $S = R = 1$  being not allowed in an SR flip flop is eliminated in a JK flip by using the feedback connections from outputs to the inputs of the gates  $G_3$  and  $G_4$ .
- The truth table of JK flip flop assumes that the inputs do not change during the clock pulse ( $CLK = 1$ ), which is not true, because of the feedback connections.
- Consider for example that, the inputs are  $J = K = 1$  and  $Q = 0$ , and a pulse as shown below is applied at the clock input.

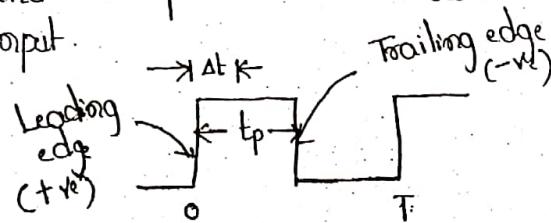


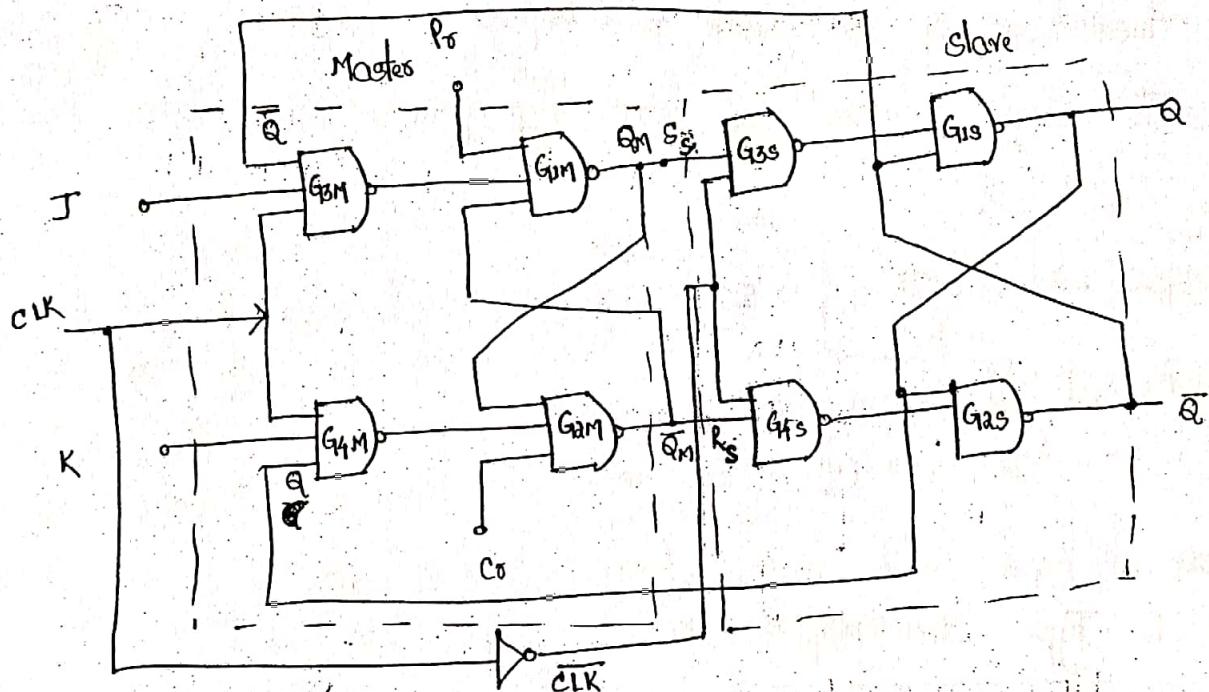
Fig. A clock pulse.

- After a time interval  $\Delta t$  equal to the propagation delay through two AND gates in series, the output will change to  $Q = 1$ .
- Now we have  $J = K = 1$  and  $Q = 1$  and after another time interval of  $\Delta t$  the output will change back to  $Q = 0$ .
- Hence, we conclude that for the duration  $t_p$  of the clock pulse, the output will oscillate back and forth between 0 & 1.
- At the end of the clock pulse, the value of  $Q$  is uncertain. This situation is referred to as race-around condition.
- The race around condition can be avoided if  $t_p \leq \Delta t \leq T$ . It is very difficult to satisfy this inequality because of very small propagation delays in ICs.

→ A more practical method for overcoming this difficulty (3) in the use of master-slave configuration discussed below

## 6. Master-slave JK flip-flop:

→ A master-slave JK flip flop is a cascade of two SR flip flops, with feedback from the outputs of the second to the inputs of the first as illustrated in fig. below.



→ Positive clock pulses are applied to the first flip-flop & the clock pulses are inverted before these are applied to the second flip-flop

→ When  $CLK=1$ , the first flip flop is enabled and the outputs  $Q_M$  and  $\bar{Q}_M$  respond to the inputs J and K according to table (truth table of JK flip-flop)

→ At this time, the second flip-flop is inhibited because its clock is low ( $\bar{CLK}=0$ )

→ When  $CLK$  goes low ( $\bar{CLK}=1$ ), the first flip flop is inhibited and the second flip-flop is enabled, because now clock is high. ( $CLK=1$ ).

→ Therefore, the outputs  $Q$  and  $\bar{Q}$  follow the outputs  $Q_M$  and  $\bar{Q}_M$

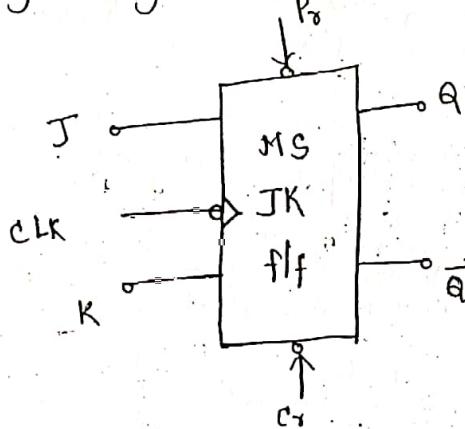
respectively.

→ Since the second flip flop simply follows the first one, referred to as slave & the first one as the master. Hence the name master-slave flip-flop.

→ In this circuit, the inputs to the gates  $G_{M1}$  and  $G_{M2}$  do not change during the CLK pulse, therefore the race-around condition does not exist.

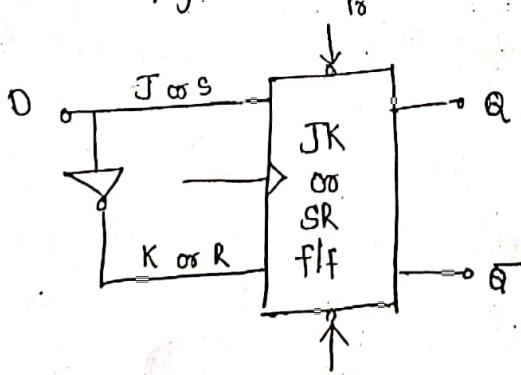
→ The state of the master-slave<sup>(crs)</sup> flip flop changes at the negative transition of the clock pulse.

→ The logic symbol of MS flip-flop is given below

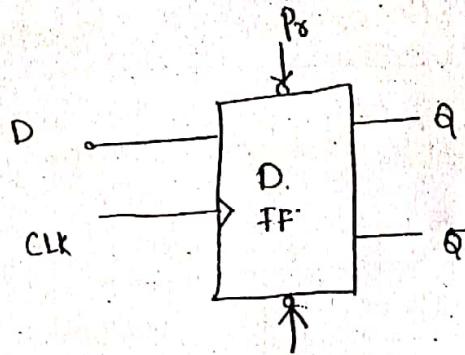


### F. D-Type flip-flop:

→ If we use only the middle two rows of the truth table of the SR or JK flip flop we obtain D flip flop as shown in fig. below



→ The logic symbol of D flip flop is as shown below



→ Truth table of D flip flop is as shown below.

Input	Output
D	Q <sub>anti</sub>
0	0
1	1

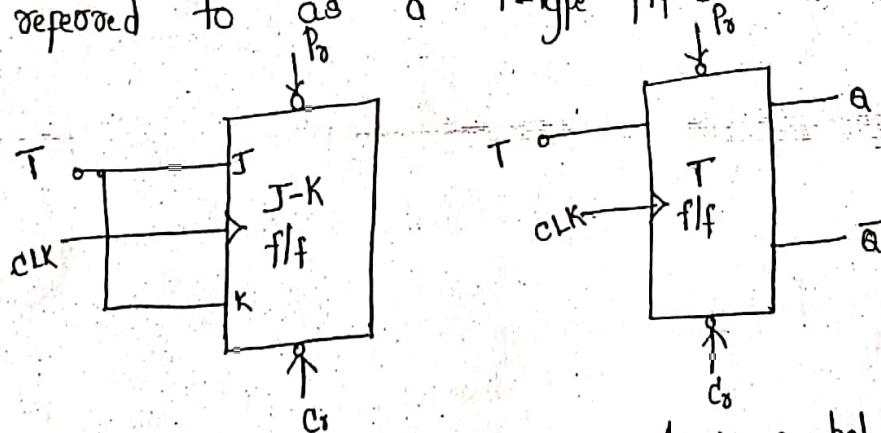
→ This is equivalent to saying that the input data appears at the end of the clock pulse.

→ Thus, the transfer of data from the input to the output is delayed & hence the name delay (D) flip flop.

→ The D type flip flop is either used as a delay device or as a latch to store 1-bit binary information.

### 8) T-type flip flop:

→ In a JK flip flop, if J=K, the resulting flip flop is referred to as a T-type flip flop is shown in fig. below.



Logic symbol.

→ It has only one input, referred to as T-input. Its

truth table is given below

Input	Output
T	Q <sub>n+1</sub>
0	Q <sub>n</sub>
1	$\bar{Q}_n$

→  $T=1$ , acts as a toggle switch. For every clock pulse the output  $Q$  changes.

Note

→ An SR flip flop cannot be converted into a T-type flip flop since  $S=R=1$  is not allowed.

## Characteristic Equations of Flip-Flops:

(2)

- are useful in analyzing circuits made of them.
- Here, next output  $Q_{n+1}$  is expressed as a function of present output  $Q_n$  and input to flip-flops
- K-map can be used to get the optimized expression and truth table and truth table of each flip-flop is mapped into it.
- Fig. below shows characteristic equation of all type of flip-flops.

i) SR f/f:

S/R	00	01	11	10
Q <sub>n</sub>	0	0	X	1
1	1	0	X	1

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

ii) D f/f:

D	0	1
Q <sub>n</sub>	0	1
1	0	1

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

iii) JK f/f

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

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

J	T	0	1
Q <sub>n</sub>	0	0	1
1	1	0	0

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

$$= T \oplus Q_n$$

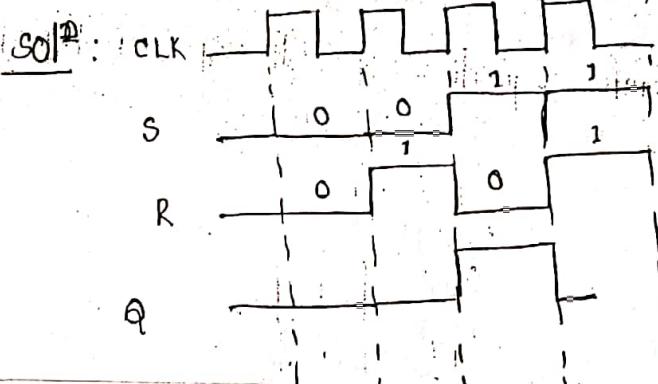
Flip-flop excitation table:

- Excitation table of a flip-flop is looking at its truth table in a reverse way.

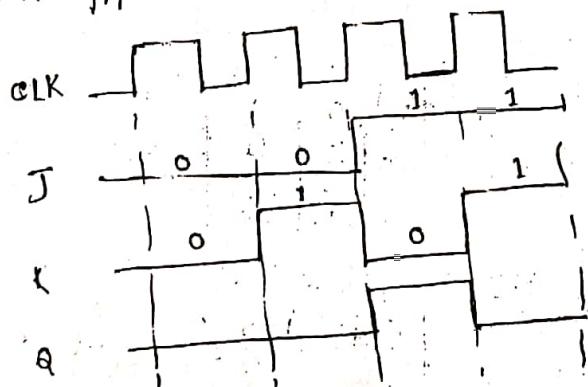
- Here, flip-flop input is presented as a dependent function of transition  $Q_n \rightarrow Q_{n+1}$  and comes later in the table.
- This is derived from flip-flop truth table or characteristic equation.

$Q_n \rightarrow Q_{n+1}$	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

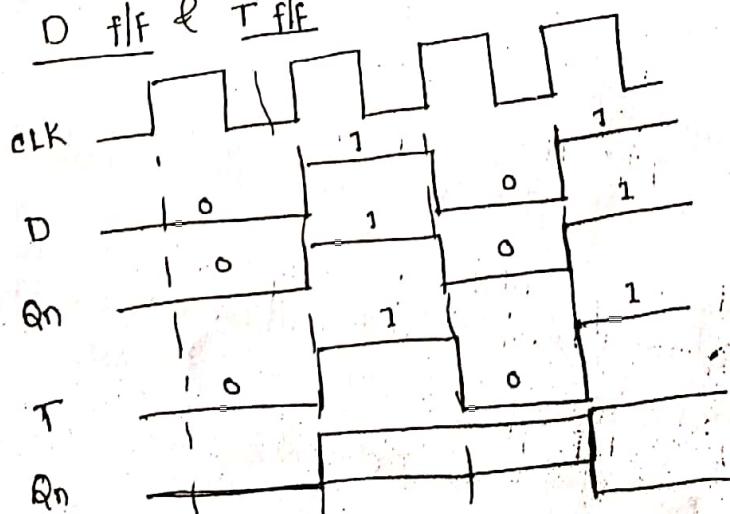
Ex: 1) Draw the output waveform for positive edge triggered SR f/f.



a) JK f/f



b) D f/f & T f/f



## Flip-flop conversion

(8)

i) Convert SR flip flop into JK flip flop.

Step 1: Write the characteristic table for available flip flop (SR).

Step 2: Write the excitation table for required flip flop (JK).

Step 3: Plot the K-map for the required flip flop to obtain its input in terms of available flip flops.

Characteristic table of SR f/f:

$Q_n \rightarrow Q_{n+1}$	S	R	J	K
00	0	0	X	0
01	1	1	0	1
10	0	0	1	X
11	1	X	0	X

Excitation table of JK f/f has to be added to the above table.

K-map: For S

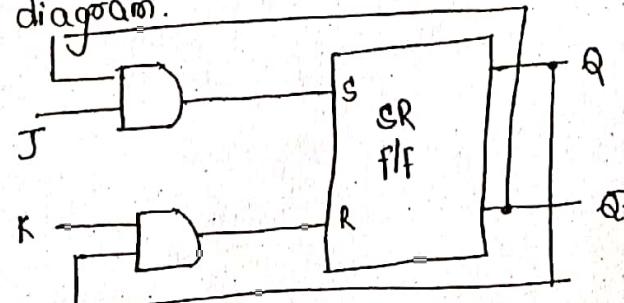
$Q_n$	JK	00	01	11	10
0	X	0	0	1	1
1	X	0	0	X	X

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

		For R:				
		JK	00	01	11	10
$Q_n$	0	X	X	0	0	
1	0		J	1	0	

$$R = K \cdot Q_n$$

→ Circuit diagram:



a) Convert JK flip flop into D flip flop

sd<sup>12</sup> i) Available flip flop is JK

ii) Required flip flop is D

→ characteristic table of JK and excitation table of D  
flip is shown below

$Q_n \rightarrow Q_{n+1}$  J K D

0 0 0 X 0

0 1 1 X 1

1 0 X 1 0

1 1 X 0 1

→ Plot K-map to obtain J and K inputs in terms  
of  $Q_n$  and D

For J:

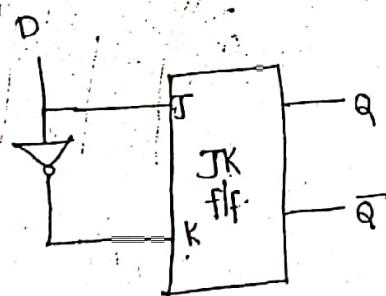
	0	1
0	0	1
1	X	X

$$J = D$$

For K:

	0	1
0	X	X
1	1	0

$$K = \bar{D}$$



3) Convert D flip flop into T flip flop

(9)

Sol<sup>2</sup>: Available flip flop  $\rightarrow$  D

Required flip flop  $\rightarrow$  T

$\rightarrow$  Write the characteristic table for D f/f and excitation table for T f/f

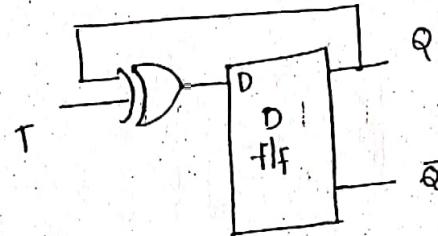
$Q_n \rightarrow Q_{n+1}$	D	T
0 0	0	0
0 1	1	1
1 0	0	1
1 1	1	0

$\rightarrow$  K map

$Q_n$	0	1
0	0 0	(1)
1	(1) 0	

$$D = \overline{Q_n}T + Q_n\bar{T}$$

$$= \underline{Q_n \oplus T}$$



4) Convert T flip flop into D flip flop

Sol<sup>2</sup>: Available flip flop  $\rightarrow$  T

Required flip flop  $\rightarrow$  D

$\rightarrow$  Write the characteristic table for T f/f and excitation table for D f/f

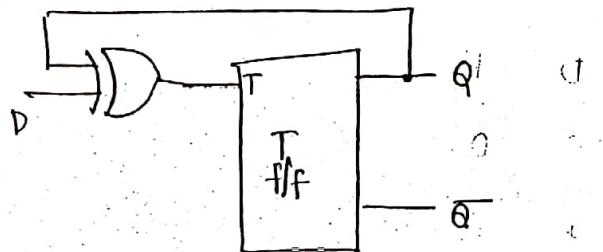
$Q_n \rightarrow Q_{n+1}$	T	D
0 0	0	0
0 1	1	1
1 0	1	0
1 1	0	1

K-map

$Q_n$	0	1
0	0	(1)
1	(1)	0

$$T = \overline{Q_n} \cdot D + Q_n \cdot \overline{D}$$

$$T = Q_n \oplus D$$



5) Convert JK flip flop into SR flip flop

SOP: Available f/f: JK

Required f/f: SR

→ Write the characteristic table for JK and excitation table for SR f/f

$Q_n \rightarrow Q_{n+1}$	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

→ Plot K-map

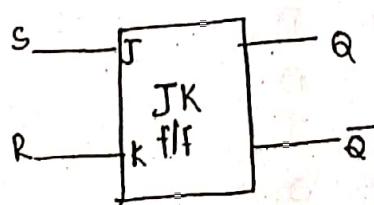
$Q_n$	00	01	11	10
0	0	0	x	1
1	x	x	x	x

for J

$$J = S$$

$Q_n$	00	01	11	10
0	x	x	x	x
1	0	1	x	0

$$K = R$$



6) Convert JK flip flop into T flip flop.

(10)

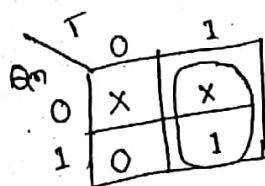
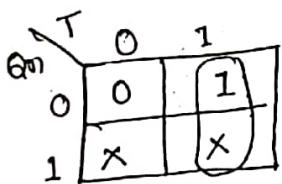
Sol<sup>2</sup> Available fl/f : JK

Required fl/f : T

→ Write the characteristic table for JK and excitation table for T fl/f

$Q_n \rightarrow Q_{n+1}$	J	K	T
0 0	0	x	0
0 1	1	x	1
1 0	x	1	1
1 1	x	0	0

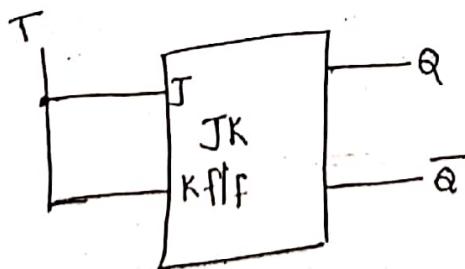
→ Plot K-map



for J

$$\boxed{J = T}$$

$$\boxed{K = T}$$

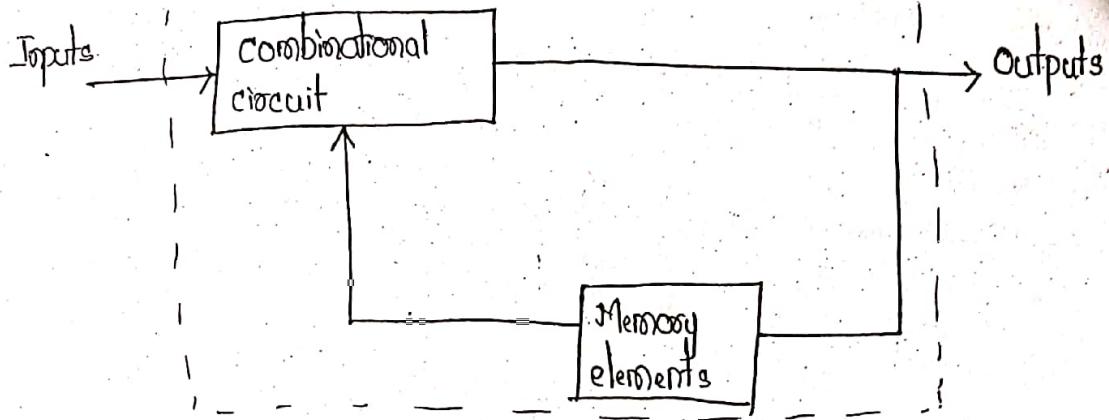


## Unit IV: Sequential Circuits

- Analysis of sequential circuits
- shift registers
  - SISO
  - SIPO
  - PISO
  - PIPO
- Ring counters
- Johnson counters
- serial Adder
- mod 'n' counter (50, 80, 100) using IC 7490 (Decade counter IC)
- Design of synchronous counters (4-bit) using
  - JK flip-flop
  - SR flip-flop
- Asynchronous counter (4-bit)
- Up counter (Down counter, UP/DOWN) counter.
- IC 7490 Decade counter.

### Sequential Circuits:

- There are many applications in which digital outputs are required to be generated in accordance with the sequence in which the input signals are received.
- This requirement cannot be satisfied using a combinational logic system.
- These applications require outputs to be generated that are not only dependent on the present input conditions but they also depend upon the past history of these inputs. The past history is provided by feedback from the output back to the input.
- Fig. below shows the block diagram of sequential circuit.



- The information stored in the memory elements at any given time defines the present state of the sequential circuit.
- The present state & the external inputs determine the outputs and the next state of the sequential circuit.
- Thus we can specify the sequential circuit by a sequence of external inputs, internal states (present state & next states), and outputs.

### Difference between combinational circuits and sequential circuits.

#### Combinational circuits

1. In combinational circuits, the output variables are at all times dependent on the combination of input variables.
2. Memory unit is not required.
3. Faster in speed, because the delay between input & output is due to propagation delay of gates.
4. Easy to design.

#### Sequential circuits

1. In sequential circuits, the output variables depends not only on the present input variables but they also depend upon the past history of these input variables.
2. Memory unit is required to store the past history of input variables.
3. Slower than combinational circuits.
4. Comparatively harder to design.

## Combinational circuits

### 5) Ex: Parallel Adder

## Sequential circuits

### 5) Ex: Serial Adder

→ The sequential circuits can be classified depending on the timing of their signals: synchronous & asynchronous.

→ In synchronous sequential circuits, signals can affect the memory elements only at discrete instants of time.

→ In synchronous sequential circuits, signals can affect the memory elements only at discrete instants of time.

→ In asynchronous sequential circuits, change in input signals can affect memory element at any instant of time.

→ The memory elements used in both circuits are flip-flops which are capable of storing 1-bit binary information.

## Synchronous sequential Circuits

i. Memory elements are clocked flip-flops

ii. Change in input signals can affect memory element upon activation of clock signal.

iii. Operatin speed of clock depends on time delays involved

iv. Easier to design

## Asynchronous Sequential Circuits

i) Memory elements are either unclocked flip or time delay elements

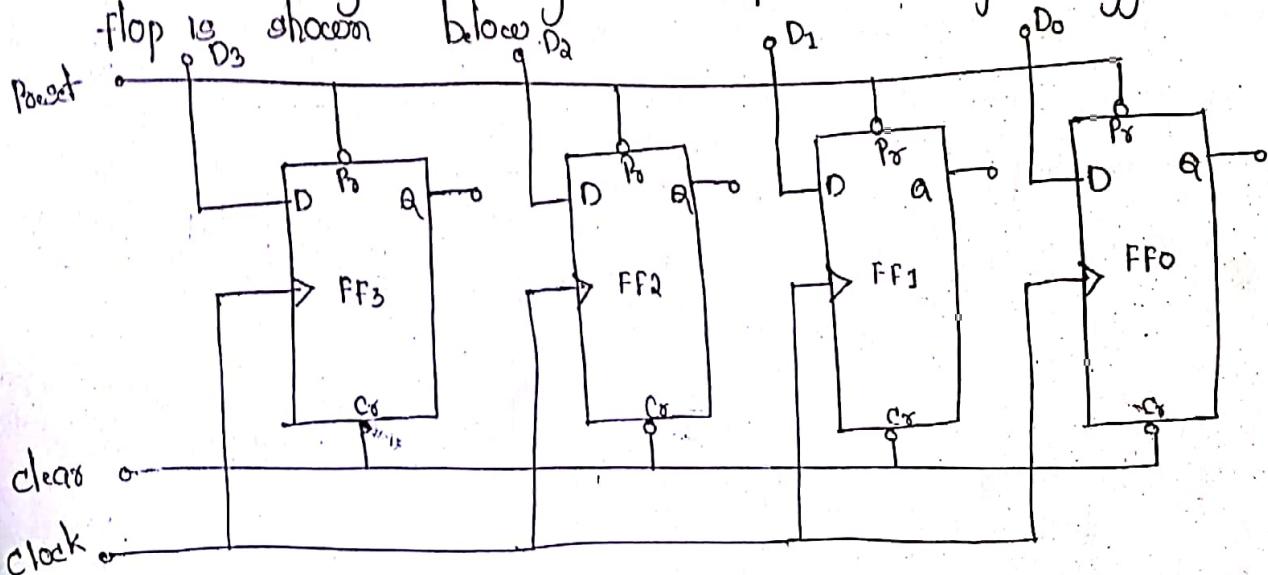
ii) Change in input signal can affect memory element at any instant of time.

iii) Because of absence of clock, asynchronous circuits can operate faster than synchronous circuits.

iv) More difficult to design

## → Shift registers:

- A register is composed of a group of flip-flops to store a group of bits (word)
- For storing an  $n$ -bit word, the no. of flip-flops required is  $n$  (one flip-flop for each bit)
- A 1-bit register using 7474 positive-edge-triggered flip-flop is shown below.



D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub> — 4-bit input  
Q<sub>3</sub> Q<sub>2</sub> Q<sub>1</sub> Q<sub>0</sub> — 4-bit output

- The bits to be stored are applied at the D-inputs which are clocked in at the leading edge of the pulse.
- In this register, the data to be entered must be available in parallel form
- Registers are classified depending upon the way in which data are entered and retrieved. There are four possible modes of operation

1. Serial In - Serial Out (SISO)
2. Serial In - Parallel Out (SIPO)
3. Parallel In - Serial Out (PISO)
4. Parallel In - Parallel Out (PIPO)

→ Registers can be designed using discrete flip-flops (SR, JK or D type) & are also available as MSI devices. (3)

Ex: IC 7495 is a 4-bit serial / parallel-in, parallel out (right shift, left shift) register.

→ Registers in which data are entered or/and taken out in serial form are referred to as shift registers, since bits are shifted in the flip flop with the occurrence of clock pulses either in the right direction (right shift register) or in the left direction (left-shift register).

① → In the bi-directional shift register, data can be shifted from left to right as well as for the reverse direction, using the mode control.

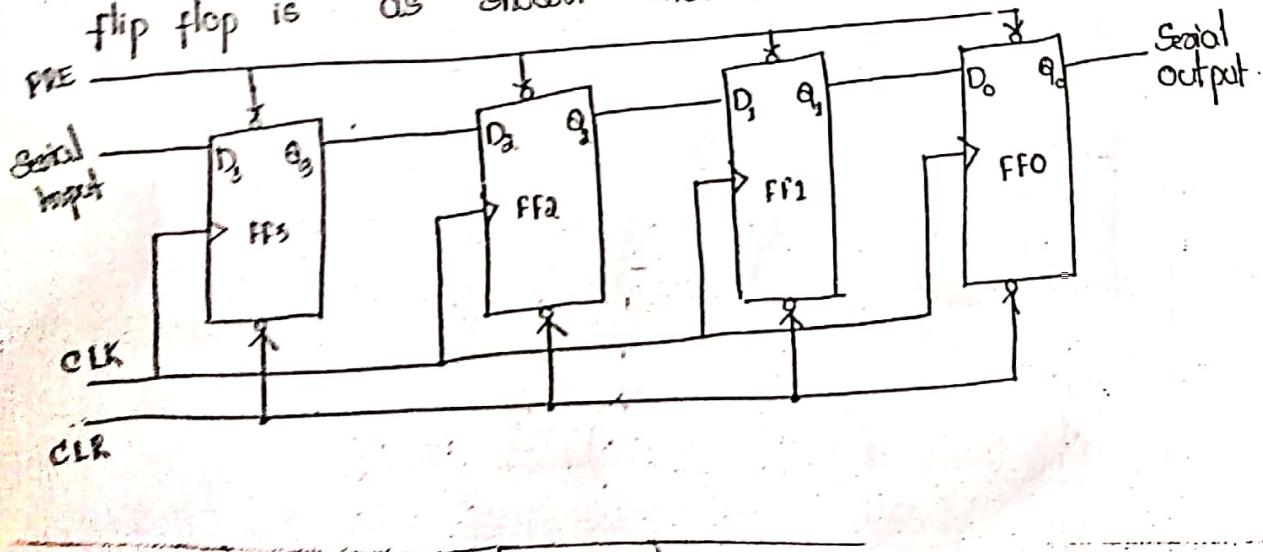
Ex: IC 7496 is a bidirectional shift register.

→ A register is referred to as a universal register if it can be operated in all the four possible modes & also as a bi-directional register.

Ex: IC 74194 is a universal register.

1) Serial In - Serial Out (SISO) shift register:

→ SISO shift register (4-bit) using positive edge triggered D flip flop is as shown below



- It is an example for right-shift register.
- First, all the flip-flops are cleared using the asynchronous input CLR. ( $CLR=0$ ; clears all flip-flops)
- PRE and CLR are both active low inputs.
- Serial input is provided to the 'D' input of FF<sub>3</sub>, which is clocked at the leading edge of the clock pulse.
- The output 'Q' of FF<sub>3</sub> is connected to 'D' input of FF<sub>2</sub>, Q of FF<sub>2</sub> is connected to 'D' input of FF<sub>1</sub>, Q output of FF<sub>1</sub> connected to 'D' input of FF<sub>0</sub>.
- The serial input bit get shifted by one position upon every leading edge of the clock pulse.
- The serial output is received bit by bit at FF<sub>0</sub>.
- 4-bit SISO shift register requires 4 flip-flops to store 4-bits of data, 4 CLK pulses to enter the data into the flip-flops, another 3 CLK pulses to retrieve the data serially.
- In general, n-bit SISO shift register requires  $(dn-1)$  CLK pulses to store and retrieve the data bits
- The working of 4-bit SISO right-shift register is as shown in table below. Assume that data bits  $D_3 D_2 D_1 D_0 = \underline{1110}$

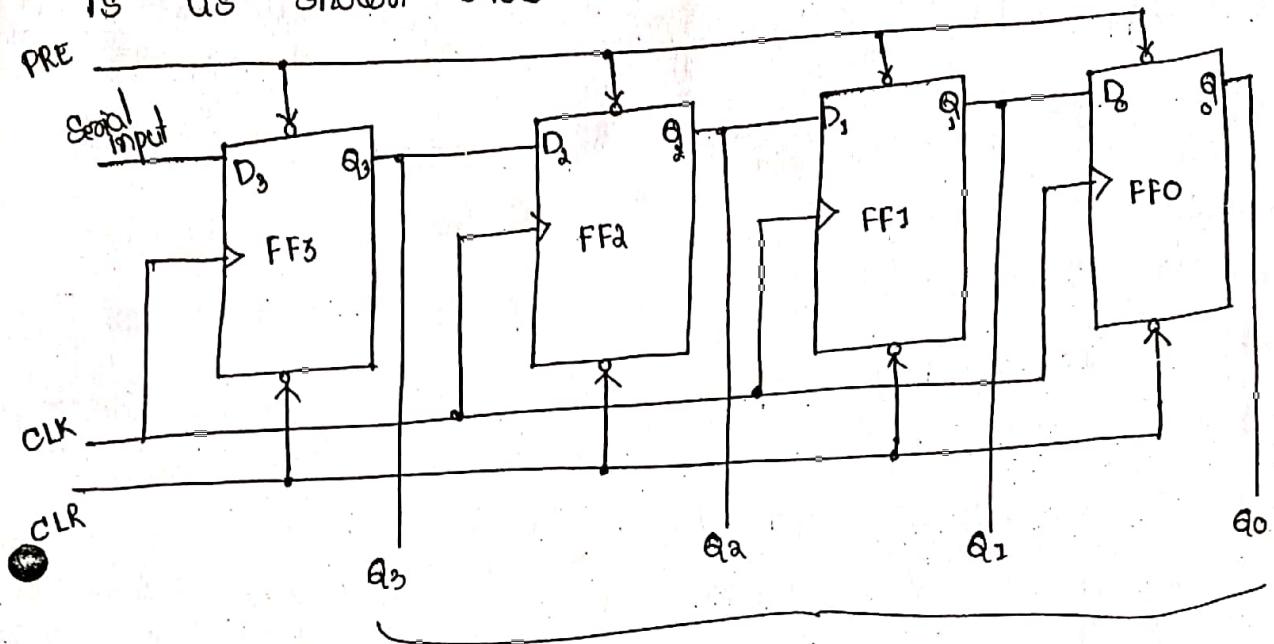
CLR	PRE	CLK	Serial input $D_3$	outputs $Q_3 \ Q_2 \ Q_1 \ Q_0$			
0	1	0	x	0	0	0	0
1	1	1	0 (LSB)	0	0	0	0
2	1	2	1	1	0	0	0
3	1	3	1	1	1	0	0
4	1	4	1 (MSB)	1	1	1	0
5	1	5	x	x	1	1	1
6	1	6	x	x	x	1	1
7	1	7	x	x	x	x	1

(Data available in FFs)      (Serial output)

## d) Serial - In Parallel - Out (SIPO) shift register:

(4)

→ SIPO shift register (4-bit) using positive edge triggered D flip flop is as shown below



Parallel outputs

shift register.

→ It is an example for eight shift register.

→ First, all the flip-flops are cleared using the asynchronous input CLR ( $CLR = 0$ ; clears all flip-flops)

→ PRE and CLR are both active low inputs

→ Serial input is provided to the 'D' input of FF3, which is clocked in at the leading edge of the clock pulse.

→ The output 'Q' of FF3 is connected to 'D' input of FF2, 'Q' of FF2 is connected to 'D' input of FF1, 'Q' output of FF1 connected to 'D' input of FFO.

→ The serial input bit get shifted by one position upon every leading edge of the clock pulse

→ After 4 CLK pulses, 4-bit data is available in 4 flip-flops

→ The output is obtained parallelly from all Q outputs ( $Q_3 Q_2 Q_1 Q_0$ )

→ So, 4-bit SIPO shift register requires 4 clock pulses to load 4-bit data and to read the output parallelly

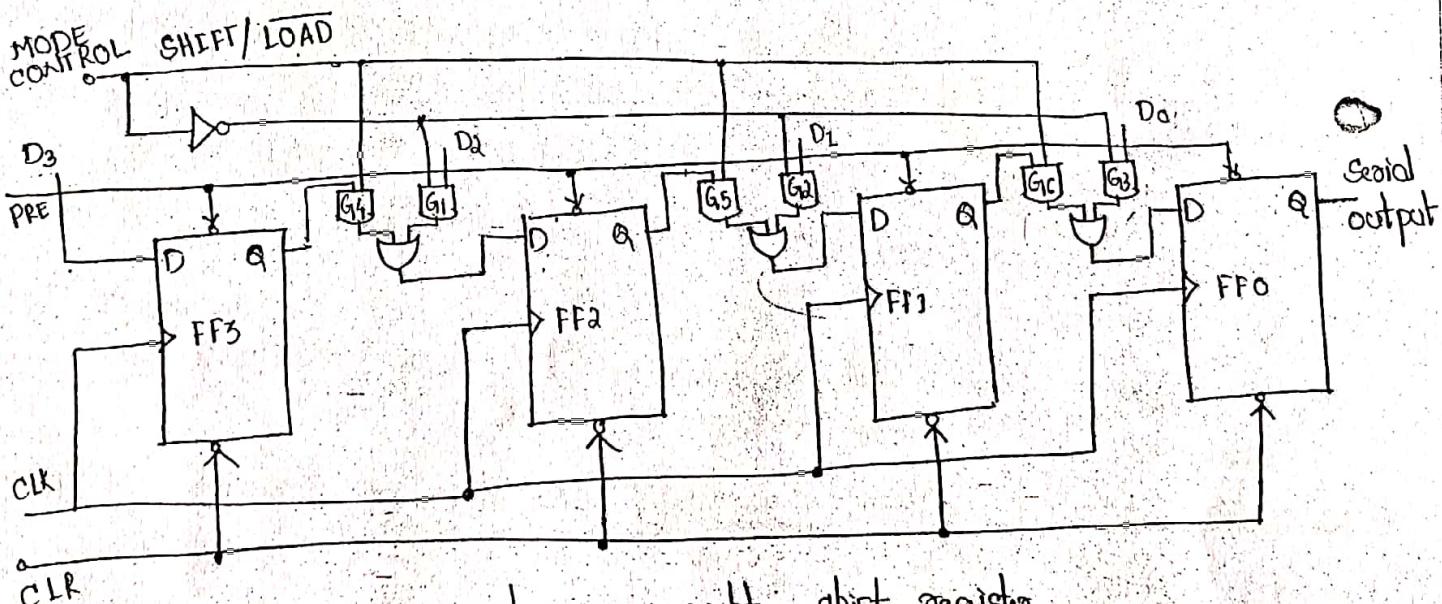
→ In general, n-bit SIPO shift register requires 'n' clock pulses to store and retrieve the data bits.

→ The working of 4-bit SIPO right-shift register is as shown in table below. Assume that data bits  $D_3 D_2 D_1 D_0 = \underline{1110}$

CLR	PRE	CLK	Serial input	Outputs
				$Q_3 \ Q_2 \ Q_1 \ Q_0$
0	1	0	x	0 0 0 0 (clear)
1	1	1	0 (LSB)	0 0 0 0
1	1	2	1	1 0 0 0
1	1	3	1	1 1 0 0
1	1	4	1 (MSB)	1 1 1 0 (Data available in flip-flops and is read parallelly)

### 3) Parallel In Serial Out: (PISO)

→ PISO shift register (4-bit) using positive edge triggered D flip-flop is as shown below.



→ It is an example for right-shift register.

→ First all the flip-flops are cleared using the asynchronous PRE input (PRE=0; clears all flip-flops).

→ PRE and CLR are both active low inputs.

→ The parallel input (4-bit  $D_3 D_2 D_1 D_0$ ) is applied through the MODE CONTROL (SHIFT/LOAD) ⑤

→ When MODE CONTROL = 0; loading operation takes place through the AND gates  $G_1, G_2, G_3$

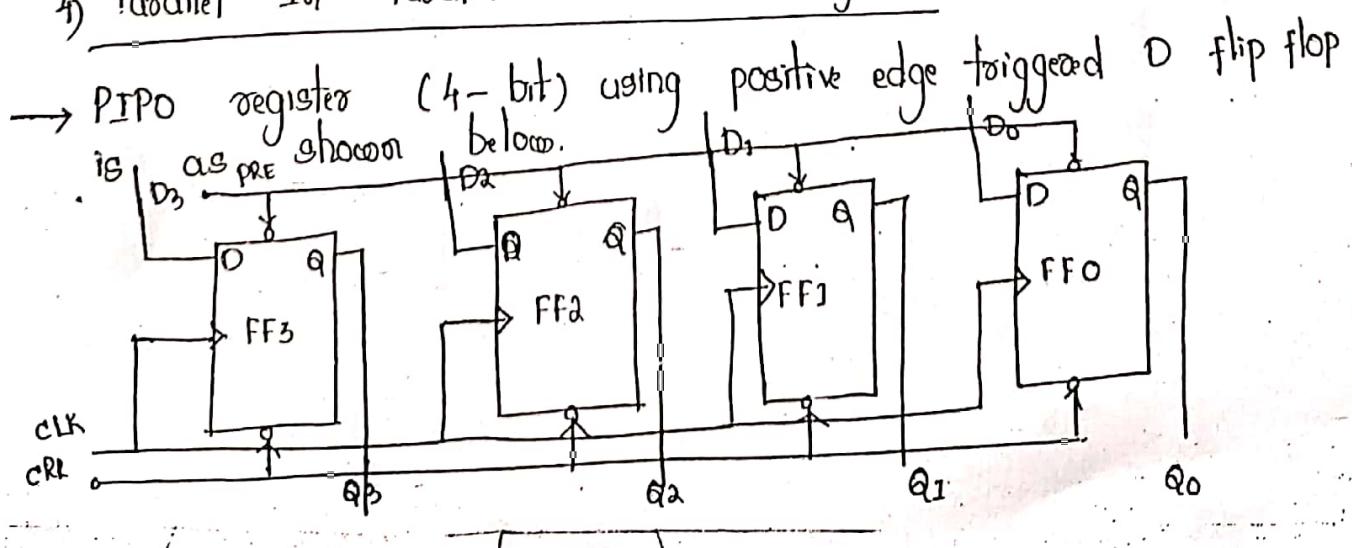
→ When MODE CONTROL = 1; shifting operation takes place through the AND gates  $G_4, G_5, G_6$  which is required in order to obtain the output bits serially at  $Q_3$

→ The working of 4-bit PISO shift register is as shown below. Assume that data bits  $D_3 D_2 D_1 D_0 = 1110$

MODE	CLR	PRE	CLK	Parallel inputs $D_3 D_2 D_1 D_0$	Output (serial) $Q_3 Q_2 Q_1 Q_0$
X	0	1	0	0 0 0 0	0 0 0 0 (clear)
0 (load)	1	1	1	1 1 1 0	1 1 1 0
1 (shift)	1	1	2	X X X X	X 1 1 1 } serial output
1	1	1	3	X X X X	X X 1 1 }
1	1	1	4	X X X X	X X X 1 }

→ 'n' bit PISO shift register requires 'n' clock pulses, to load parallel and to receive the data bits serially.

#### 4) Parallel In Parallel Out (PIPO) Register:



→ It is the simplest of all registers

→ Data bits are loaded parallelly and read parallelly

→ The working of PIPo register is as shown below

CLR	PRE	CLK	Parallel inputs	Parallel outputs	
0	1	0	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	(clear)
1	1	1	{ 1 1 1 0 }	{ 1 1 1 0 }	parallel input      parallel output

→ PIPo shift register requires one clock pulse to load and read data parallelly.

## Applications of registers / shift registers:

(6)

- The primary use of shift registers are temporary data storage & bit manipulations.

### 1) Delay line:

- A SISO shift register can be used to introduce time delay  $\Delta t$  in digital signals given by
- $$\Delta t = N \times \frac{1}{f_c}$$
- where 'N' is the no. of stages and  $f_c$  is the clock frequency
- The amount of delay can be controlled by the clock frequency or the no. of flip flop in the shift register.

### 2) Serial-to-parallel converter:

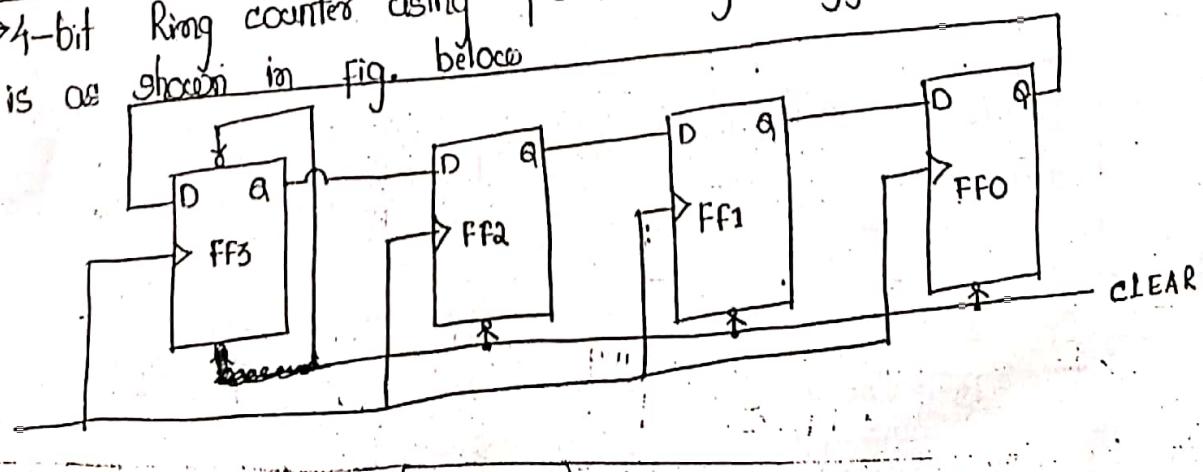
- SIPO shift register can be used to perform serial-to-parallel conversion.

### 3) Parallel-to-serial converter:

- PISO shift register can be used to perform parallel-to-serial conversion.

### 4) Ring counter:

- 4-bit Ring counter using positive edge triggered D flip-flop is as shown in fig. below



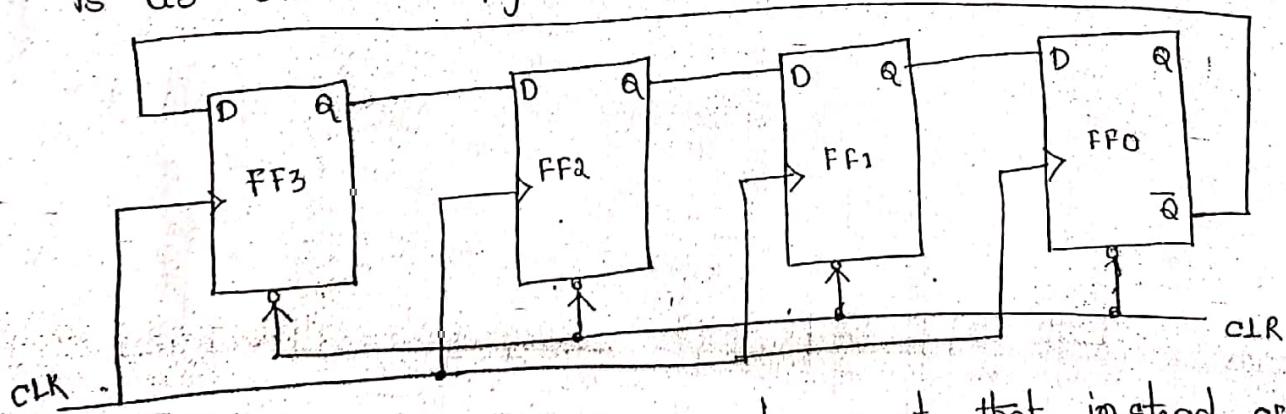
- It is similar to 4-bit SISO shift register, with the exception that the output  $Q_0$  (FF0) is connected back to the input of first flip-flop (FF3).
- Also, the preset input of first flip-flop (FF3) is connected to clear inputs of FF2, FF1 and FF0.
- As a result, when  $CLR=0$ , FF3 is set, FF2, FF1 and FF0 are cleared as shown in table below.
- Here, the preset input of FF2, FF1 and FF0 are unused and clear input of FF3 is unused.

CLR	CLK	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	1	0	0	0
1	1	0	1	0	0
1	2	0	0	1	0
1	3	0	0	0	1
1	4	1	0	0	0

- After 4 clock pulses, again the sequence repeats, hence the name ring counter.

### 5) Johnson Counter:

- 4-bit Johnson counter using positive edge triggered 'D' flip flop is as shown in fig. below.



- It is similar to ring counter except that instead of  $Q$ ,  $\bar{Q}$  is connected back to FF3's D input.

→ Here preset input of all flip-flop remains unused.

(7)

→ The working of Johnson counter is as shown in table below.

CLR.	CLK	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	0	0	0	0	0
1	1	1	0	0	0
1	2	1	1	0	0
1	3	1	1	1	0
1	4	1	1	1	1
1	5	0	1	1	1
1	6	0	0	1	1
1	7	0	0	0	1
1	8	0	0	0	0
1	9	1	0	0	0

#### 6) Serial Adder:

- For 8-bit addition we need 8 FA units. Then the addition is done in parallel.
- Using shift register we can convert this parallel addition to serial one and reduce number of FA units to only one.
- The benefit of this technique is more pronounced if the hardware unit that's needed to be used in parallel is very costly.
- Fig. below shows how serial addition takes place in a time-multiplexed manner and also provides a snapshot of the register values.

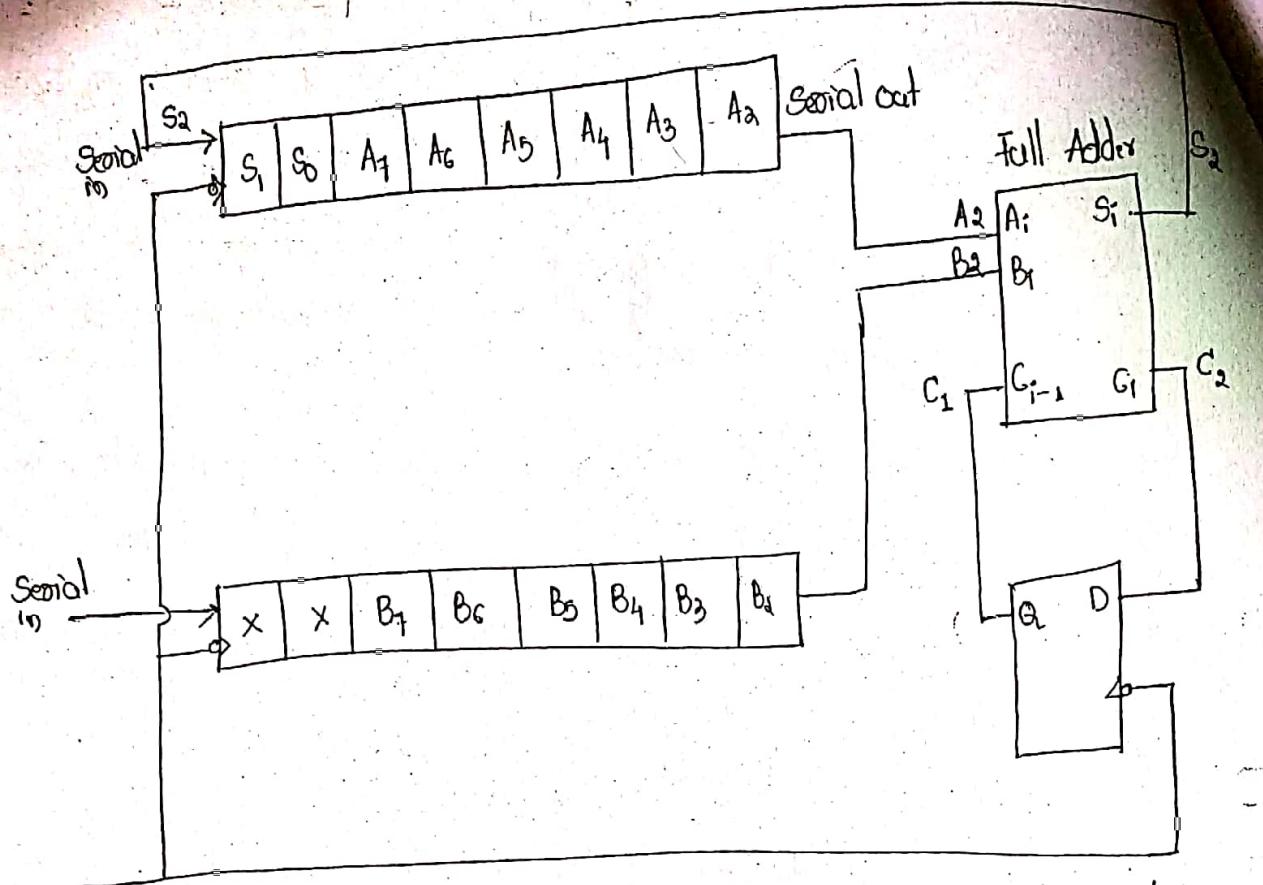


Fig. Serial addition of two 8-bit numbers (Register values shown are at 3rd clock cycle)

- Two 8-bit numbers to be added ( $A_7 \dots A_0$  and  $B_7 \dots B_0$ ) are loaded in two 8-bit shift registers A and B.
- The LSB of each number appears in the rightmost position in two registers.
- Serial data out of A and B are fed to data inputs of full adder.
- The carry-in is fed from its own carry output delayed by one clock period by a D flip-flop, which is initially cleared.
- Both registers and D flip-flop are triggered by same clock. The sum ( $S$ ) output of FA is fed to serial data in of shift register A.
- The serial addition takes place like this. The LSBs of two numbers ( $A_0$  and  $B_0$ ) appearing at serial out of respective registers are added by FA during 1st clock cycle and generate sum ( $S_0$ ) and

and carry ( $C_0$ )

(7ii)

→  $S_0$  is available at serial data input of register A and  $C_0$  at input of D flip-flop.

→ At NT of clock shift registers shift its content to right by one unit.

→  $S_0$  becomes MSB of A and  $C_0$  appears at D flip-flop output

→ Therefore in the second clock cycle FA is fed by second bit (A<sub>1</sub> and B<sub>1</sub>) of two numbers and previous carry ( $C_0$ )

→ This process goes on and is stopped by inhibiting the clock after 8 clock cycles.

→ At that time shift register A stores the sum bits,  $S_7$  in leftmost (MSB) position and  $S_0$  in rightmost (LSB) position.

→ The final carry is available at D flip-flop output

→ The limitation of this scheme is that the final addition result is delayed by eight clock cycles.

→ In parallel adder the result is obtained almost instantaneously, after nano seconds of delay.