

## Unit IV

Chapter

# 4

## Flip Flops

### Syllabus

1 bit memory cell, Clocked SR, JK, MS JK flip flop, D and T flipflops, Use of preset and clear terminals, Hold and setup time and metastability, Excitation table for flipflops, Conversion of flipflops, Typical data sheet specifications of Flip flop, Applications of flip flops, Clock skew, Clock jitter, Effect on synchronous design.

### Chapter Contents

4.1	Introduction	4.13	Preset and Clear Inputs
4.2	S-R Latch using NOR Gates	4.14	Various Representations of Flip Flops
4.3	Triggering Methods	4.15	Excitation Table of Flip-Flop
4.4	Gated Latches (Level Triggered SR Flip Flop)	4.16	Conversion of Flip Flops
4.5	The Gated S-R Latch (Clocked S-R Flip Flop)	4.17	Flip Flop Timing Considerations
4.6	The Gated D Latch (Clocked D Flip Flop)	4.18	Flip Flop Metastability
4.7	Gated JK Latch (Level Triggered JK Flip Flop)	4.19	Applications of Flip Flops
4.8	Edge Triggered Flip Flops	4.20	Analysis of Clocked Sequential Circuits
4.9	Edge Triggered D Flip Flop	4.21	Design of Clocked Synchronous State Machine using State Diagram
4.10	Edge Triggered J-K Flip Flop	4.22	Clock Jitter and it's Effects on Synchronous Design
4.11	Toggle Flip Flop (T Flip Flop)	4.23	Typical Data Sheet Specifications of Flip flop
4.12	Master Slave (MS) JK Flip Flop		



## 4.1 Introduction :

### 4.1.1 Combinational Circuits :

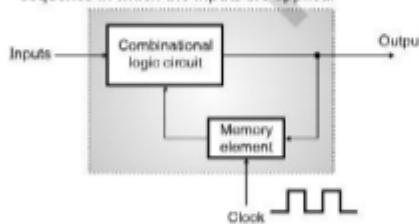
#### Definition :

- A combinational circuit is a logic circuit the output of which depends only on the combination of the inputs. The output does not depend on the past value of inputs or outputs.
- Hence combinational circuits do not require any memory (to store the past values of inputs or outputs).
- Till now we have discussed only the combinational circuits.
- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals. It does not depend on the past status of inputs.
- The combinational circuits do not use any memory. Therefore the previous states of input does not have any effect on the present state of the circuit.
- Also the sequence in which the inputs are being applied has no effect on the output of a combinational circuit.
- We do not have to use any timing and synchronization signal such as clock signal in a combinational circuit.

### 4.1.2 Sequential Circuits :

#### Definition :

- In the sequential circuit, the **timing** parameter also needs to be taken into consideration.
- The output of a sequential circuit depends on the present time inputs, the previous output (past) and the sequence in which the inputs are applied.



(c-seo) Fig. 4.1.1 : Block diagram of a sequential circuit

- In order to provide the previous input or output a memory element is required to be used.
- Thus a sequential circuit needs to use a memory element as shown in Fig. 4.1.1.

- Fig. 4.1.1 shows the block diagram of a sequential circuit which includes the memory element in the feedback path.

#### Present state of sequential circuit :

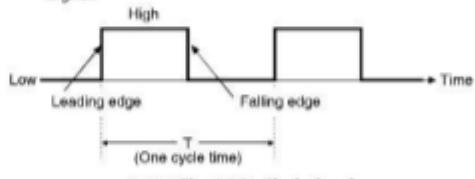
- The data stored by the memory element at any given instant of time is called as the present state of the sequential circuit.

#### Next state :

- The combinational circuit shown in Fig. 4.1.1 operates on the external inputs and the present state to produce new outputs.
- Some of these new outputs are stored in the memory element and called as the next state of the sequential circuit.
- The most important part of the sequential circuit seems to be the memory element.
- The memory element of Fig. 4.1.1 is known as Flip Flop (FF). It is the basic memory element.

### 4.1.3 Clock Signal :

- The clock signal shown in Fig. 4.1.2 is a timing signal. Every sequential signal will have this timing signal applied as an input signal as an input signal.
- Clock is a rectangular signal as shown in Fig. 4.1.2; with a duty cycle equal to 50%. That means it's on time is equal to its off time.
- The clock signal repeats itself after every  $T$  seconds. Hence the clock frequency is  $f = 1/T$ .
- The flip flops may respond to the edges of the clock or they may respond to the high or low levels of the clock signal.



(c-seo) Fig. 4.1.2 : Clock signal

### 4.1.4 Clock Skew :

SPPU : May 14, May 19

#### University Questions

- Q. 1 What is clock skew and clock jittering in synchronous circuits ? (May 14, 2 Marks)



<b>Q. 2</b>	What is clock skew and clock jitter ? (May 19, 2 Marks)
-------------	--

**Definition :**

- Clock skew is defined as the time difference between the instants at which clock edges arrive at a pair of clock inputs.
- In a perfect system, the clock signals at various clock input pins of the system arrive at exactly the same instant of time and the skew is zero.
- But in real time systems, the edges do not arrive at exactly the same time and there is some skew.
- This clock skew occurs due to different delays the clock signal experiences on different paths from the clock generator to various circuits.
- The major reasons for this are :
  1. Different length of wires (wires introduce delay).
  2. Different number of gates (buffers) on the paths.
  3. Use of flip-flops that clock on different edges (need for inverting clock for some flip-flops).
  4. The process of gating the clock to control loading of registers.
- The **maximum allowable clock skew** for the system is equal to the difference between the shortest and longest path delays.
- Clock skew is an important design parameter in high-speed clock systems.

**4.1.5 Comparison of Combinational and Sequential Circuits :** SPPU : May 10**University Questions**

<b>Q. 1</b>	Explain the difference between combinational and sequential circuits. (May 10, 4 Marks)
-------------	--

Sr. No.	Parameter	Combinational circuits	Sequential circuits
1.	Output depends on	Inputs present at that instant of time.	Present inputs and past inputs/outputs.
2.	Memory	Not necessary	Necessary
3.	Clock input	Not necessary	Necessary
4.	Examples	Adders, subtractors, code converters	Flip flops, shift registers, counters

**4.1.6 1-Bit Memory Cell**(Basic Bistable Element) : SPPU : Dec. 17**University Questions**

<b>Q. 1</b>	Write a short note on one-bit memory cell.
-------------	--

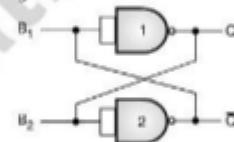
(Dec. 17, 4 Marks)

**Definition :**

- Flip-flop is also known as the basic digital memory circuit.
- It has two stable states namely logic 1 state and logic 0 state. We can design it either using NOR gates or NAND gates.

**Circuit diagram :**

- A flip-flop can be designed by using the fundamental circuit shown in Fig. 4.1.3. NAND gates 1 and 2 are basically acting as inverters. Hence this circuit is called as a cross coupled inverter.
- Output of gate 1 is connected to the input of gate-2 and output of gate 2 is connected to input of gate-1 as shown in Fig. 4.1.3.

**(IC-562) Fig. 4.1.3 : A cross coupled inverter as memory element****Operation :**

- Assume that output of gate-1 i.e.  $Q = 1$ . Hence  $\bar{Q} = 0$ .
- As  $B_2 = 1$ , output of gate-2 i.e.  $\bar{Q} = 0$ . This makes  $B_1 = 0$ .
- Hence  $Q$  continues to be equal to 1.
- Similarly we can demonstrate that if we start with  $Q = 0$ , then we end up obtaining  $Q = 0$  and  $\bar{Q} = 1$ .

**Conclusions :**

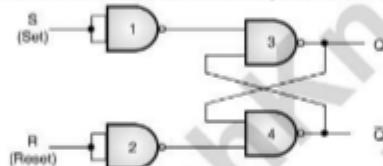
- From the above discussion we can draw the following conclusions :
- The outputs of the circuit ( $Q$  and  $\bar{Q}$ ) will always be complementary. That means if  $Q = 0$  then  $\bar{Q} = 1$  and vice versa. They will never be equal  $Q = \bar{Q} = 0$  or 1 is an invalid state.



- This circuit has two stable states. One of them corresponds to  $Q = 1, \bar{Q} = 0$  and it is called as 1 state or **set state**. Whereas the other state corresponds to  $Q = 0, \bar{Q} = 1$  and it is called as 0 state or **reset state**.
- If the circuit is in the reset state ( $Q = 0, \bar{Q} = 1$ ), then it will continue to be in the reset state and if it is in the set state ( $Q = 1, \bar{Q} = 0$ ) then it will continue to remain in the set state.
- This property of the circuit shows that it can store 1 bit of digital information. Therefore it is called as a **1-bit memory cell**.

#### 4.1.7 Latch :

- The cross coupled inverter of Fig. 4.1.3 is capable of locking or latching the information. Hence this circuit is also called as a latch.
- The disadvantage of the cross coupled inverter circuit is that we cannot enter the desired digital data into it.



(C-563) Fig. 4.1.4 : Modified memory cell

- This disadvantage can be overcome by modifying the circuit as shown in Fig. 4.1.4. This modification will allow us to enter the desired digital data into the circuit.

#### Operation :

##### Case I : $S = R = 0$ (No change) :

- The outputs of gates 1 and 2 will become 1.
- Let  $Q = 0$  and  $\bar{Q} = 1$  initially. Hence both the inputs to gate 3 are 1 and the inputs to gate 4 are (01). So gate-3 output i.e.  $Q = 0$  and gate-4 output i.e.  $\bar{Q} = 1$ .
- Thus with  $S = R = 0$ , there is no change in the state of outputs.

##### Case II : $S = 1, R = 0$ (Set) :

- Since  $S = 1$  and  $R = 0$ , one of the inputs to gate-3 will be 0. This will force  $Q$  output to 1.
- Hence both the inputs to gate-4 will be 1. This forces  $\bar{Q}$  to 0.

- Hence for  $S = 1, R = 0$ , the outputs are  $Q = 1$  and  $\bar{Q} = 0$ . This is set state.

##### Case III : $S = 0, R = 1$ (Reset) :

- If  $S = 0, R = 1$  then one of the inputs to gate-4 will be 0. This will force the  $\bar{Q}$  output to 1.
- Hence both the inputs to gate 3 will be 1. This forces  $Q$  to 0.
- Thus for  $S = 0, R = 1$ , the outputs are  $Q = 0, \bar{Q} = 1$ . This is the reset state or clear state.

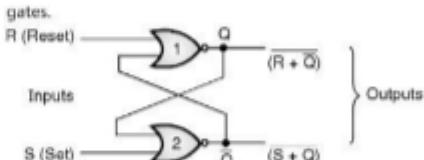
##### Case IV : $S = R = 1$ (Race : Prohibited) :

- If  $S = R = 1$  then outputs of gates 1 and 2 will be zero.
- Hence one of the inputs to gates 3 and 4 will be 0.
- So both the outputs  $Q$  and  $\bar{Q}$  will try to become 1. It is not allowed as  $Q$  and  $\bar{Q}$  should be complementary. Hence  $S = R = 1$  condition is prohibited.

#### 4.2 S-R Latch using NOR Gates :

##### Logic diagram :

- Latch is a bistable circuit which has two stable states.
- It has two outputs  $Q$  and  $\bar{Q}$  which are complements of each other. The two input terminals to this latch are set ( $S$ ) and reset ( $R$ ).
- Latch is a sequential logic circuit which monitors all its inputs continuously and will change its output as soon as the input changes. It does not wait for the clock signal.
- Generally an enable input signal is given for a latch. When the enable signal is active, the output will change in response to a change in input.
- S-R (set – reset) latch is the simplest type of latch and Fig. 4.2.1 shows S-R latch constructed using the NOR gates.



(C-564) Fig. 4.2.1 : S-R latch using NOR gates

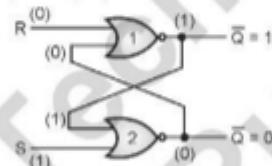
- Two NOR gates are cross coupled. That means the output of NOR gate 1 is connected to one of the inputs of the NOR gate 2. And output of NOR gate 2 is connected to one of the inputs of NOR gate 1.
- The outputs  $Q$  and  $\bar{Q}$  can be expressed using Boolean equations as follows :

$$Q = \overline{R + \bar{Q}} \text{ and } \bar{Q} = \overline{\bar{S} + Q}$$

#### 4.2.1 Operation of S-R Latch :

- Remember that the NOR gate output goes to "0" when any one its inputs becomes HIGH (1).
- If both the inputs are high (1), then the output is "0".
- Let us understand the operation under four different conditions :
  1. Case I :  $S = 1, R = 0$
  2. Case II :  $S = 0, R = 1$
  3. Case III :  $S = 0, R = 0$
  4. Case IV :  $S = 1, R = 1$
- For the operation refer Figs. 4.2.2(a), (b), (c), (d)

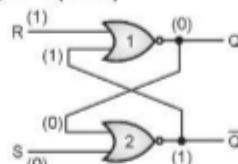
##### Case I : $S = 1, R = 0$ (Set)



(c-565) Fig. 4.2.2(a) : Operation with  $S = 1, R = 0$

- As  $S = 1$ , output of NOR gate 2 i.e.  $\bar{Q}$  becomes 0.  $R$  is already 0. Hence both inputs of NOR – 1 are 0.
- Therefore  $Q = 1$ .
- Hence both inputs to NOR – 2 are 1.
- Hence  $\bar{Q} = 0$ . Thus a stable output state of  $Q = 1$  and  $\bar{Q} = 0$  is reached.
- ∴ For  $S = 1, R = 0, Q = 1$  and  $\bar{Q} = 0$ . This is called as the "Set" mode.

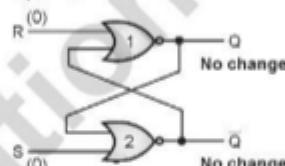
##### Case II : $S = 0, R = 1$ (Reset)



(c-566) Fig. 4.2.2(b) : Operation with  $S = 0$  and  $R = 1$

- As  $R = 1$  output of NOR-1 i.e.  $Q$  becomes 0.  $S$  is already 0. Hence both inputs to NOR-2 are 0.
- Hence output of NOR-2 i.e.  $\bar{Q} = 1$ . Thus the circuit reaches a stable state where  $Q = 0$  and  $\bar{Q} = 1$ .
- ∴ For  $S = 0$  and  $R = 1, Q = 0$  and  $\bar{Q} = 1$ . This is known as the reset mode of operation.

##### Case III : $S = 0, R = 0$



(c-567) Fig. 4.2.2(c) : Operation with  $S = R = 0$

- We know that  $\bar{Q} = \overline{S + Q}$  and  $Q = \overline{R + \bar{Q}}$

Substitute  $S = R = 0$

$$\therefore \bar{Q} = \overline{0 + Q} \text{ and } Q = \overline{0 + \bar{Q}}$$

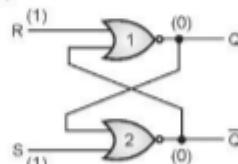
$$\therefore \bar{Q} = \bar{0} \cdot \bar{Q} \text{ and } Q = \bar{0} \cdot Q$$

according to De-Morgan's theorems.

$$\therefore \bar{Q} = 1 \cdot \bar{Q} = \bar{Q} \text{ and } Q = 1 \cdot Q = Q$$

- If  $S = R = 0$ , then  $Q$  and  $\bar{Q}$  do not change their states.

##### Case IV : $S = 1, R = 1$



(c-568) Fig. 4.2.2(d) : Operation with  $S = R = 1$

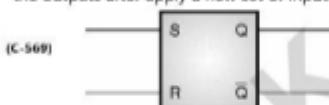
- If  $S = R = 1$ , then one input each of NOR-1 and NOR-2 is at logic 1. So outputs of both these gates will try to become 0.



- Thus both the outputs  $Q$  and  $\bar{Q}$  try to become 0 at the same time. This is not acceptable because  $Q$  and  $\bar{Q}$  should be complements of each other.
- Therefore this state is an indeterminate state and therefore should be avoided.
- This state should not be used because it violates the basic requirement of a latch i.e.  $Q$  and  $\bar{Q}$  should be complements of each other.

#### 4.2.2 Symbol and Truth Table of S-R Latch :

- The symbol and truth table of S-R latch using NOR gates are as shown in Figs. 4.2.3(a) and (b) respectively.
- In the truth table  $Q_n$  and  $\bar{Q}_n$  represent the present states of outputs i.e. these are the outputs before applying a new set of inputs.
- $Q_{n+1}$  and  $\bar{Q}_{n+1}$  represent the next states of outputs i.e. the outputs after applying a new set of inputs.



$Q_n$  and  $\bar{Q}_n$ : Present states

$Q_{n+1}$  and  $\bar{Q}_{n+1}$ : Next states

(a) Symbol

Inputs		Outputs				Comment
S	R	$Q_n$	$Q_n$	$Q_{n+1}$	$\bar{Q}_{n+1}$	
0	0	0	1	0	1	No change (NC)
0	0	1	0	1	0	
0	1	0	1	0	1	Reset
0	1	1	0	0	1	
1	0	0	1	1	0	Set
1	0	1	0	1	0	
1	1	0	1	X	X	Prohibited state
1	1	1	0	X	X	

(C-8074) (b) Truth table of S-R latch

Fig. 4.2.3

#### Summary of operation of S-R latch :

- The summary of operation of an S-R latch is as follows :
- For  $S = R = 0$  the latch output does not change.
- $S = 0, R = 1$  is called as the "Reset" condition as  $Q = 0$  and  $\bar{Q} = 1$ .

- $S = 1, R = 0$  is called as the "Set" condition as  $Q = 1$  and  $\bar{Q} = 0$ .
- $S = R = 1$  is the prohibited state. The output is unpredictable. This condition should therefore be avoided.

#### Race condition :

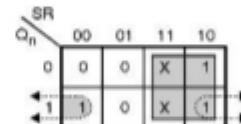
- The condition  $S = R = 1$  is called as "Race" condition.
- When any one input to a NOR gate is 1, its output becomes 0. Thus both the outputs will try to become 0. This is called as the RACE condition.

#### 4.2.3 Characteristic Equation :

- One way of explaining the behaviour of a latch or flip flop is to use its truth table.
- The truth table is also called as **excitation table**. Another way of doing it is to use special type of equations called **characteristic equations**.
- The characteristic equation of a flip-flop is the equation which relates the next state of the flip flop or latch  $Q_{n+1}$  or  $\bar{Q}_{n+1}$  to the current state and inputs ( $Q_n$ , S and R).
- Characteristic equation is actually obtained from the truth table of the flip flop or latch, using the K-map.

#### Characteristic equation of SR latch :

- Refer to the truth table of SR latch and write the K-map for the next state of output i.e.  $Q_{n+1}$  as shown in Fig. 4.2.4.



(C-570) Fig. 4.2.4 : K-map for next state

$Q_{n+1}$  of SR latch

- After simplification, the characteristic equation of SR latch is given by,

$$Q_{n+1} = S + \bar{R} Q_n \quad \dots(4.2.1)$$

#### 4.2.4 NAND Latch [S-R Latch using NAND Gates] :

SPPU : Dec. 11, May 14

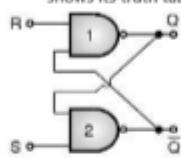
#### University Questions

- Q. 1 Draw and explain SR flip-flop using NAND gates.

(Dec. 11, May 14, 8 Marks)

**Logic diagram :**

- We can construct a S-R latch with NAND gates as shown in Fig. 4.2.5(a).
- Note that the outputs of two NAND gates are cross connected, in an identical manner as that in the NOR latch.
- Fig. 4.2.5(a) shows the NAND latch and Fig. 4.2.5(b) shows its truth table.



(a) NAND latch

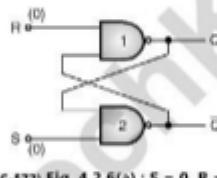
S	R	$Q_{n+1}$	$\bar{Q}_{n+1}$
0	0	RACE	RACE
0	1	0	1
1	0	1	0
1	1	(NC) $Q_n$	(NC) $\bar{Q}_n$

(b) Truth table

(C-571) Fig. 4.2.5

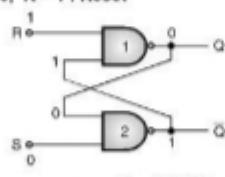
**Operation :**

- The operation of S-R NAND latch is summarised in Fig. 4.2.6.

**Case I : S = 0, R = 0 : Race**

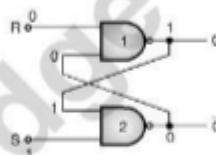
(C-572) Fig. 4.2.6(a) : S = 0, R = 0

- When any one input of a NAND gate becomes 0, its output is forced to 1.
- Here S = R = 0.  $\therefore$  Q and  $\bar{Q}$  both will be forced to be equal to 1.
- This is an undeterminate state and hence should be avoided.
- This is also called as Race condition.

**Case II : S = 0, R = 1 : Reset**

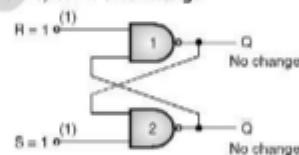
(C-573(a)) Fig. 4.2.6(b)

- Since S = 0, it forces  $\bar{Q}$  to be 1.
- Hence both inputs to NAND-1 are 1. Hence Q = 0.
- Thus with S = 0 and R = 1 the outputs are Q = 0 and  $\bar{Q} = 1$ .
- This is the reset condition.

**Case III : S = 1, R = 0 : Set**

(C-573) Fig. 4.2.6(c)

- Since R = 0, Q is forced to 1.
- Hence both inputs to NAND-2 are 1. Hence  $\bar{Q} = 0$ .
- Thus with S = 1 and R = 0 the outputs are Q = 1 and  $\bar{Q} = 0$ .
- This is the set condition.

**Case IV : S = 1, R = 1 : No change**

(C-574) Fig. 4.2.6(d)

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

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

- Using De-Morgan's theorem,

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

$$\text{and } \bar{Q}_{n+1} = \overline{S} + \overline{Q}_n$$

- Substitute  $\overline{R} = 0$  and  $\overline{S} = 0$  to get,

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

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

- Thus there is no change in the outputs if S = R = 1.



- The symbol for S-R latch is shown in Fig. 4.2.7 along with the summary of operation.

#### Circuit symbol and summary of operation of S-R latch :

- For  $S = 0, R = 0$ , both the outputs will be forced to become 1. This is RACE condition and should be avoided.
- For  $S = 0, R = 1$ , the outputs are  $Q = 0, \bar{Q} = 1$  and it is called as the Reset condition.
- For  $S = 1, R = 0$ , the outputs are  $Q = 1, \bar{Q} = 0$  and it is called as the set condition.
- For  $S = R = 1$ , there is no change in the output state.



(c-575) Fig. 4.2.7 : Symbol of S-R latch

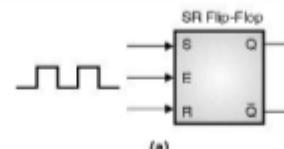
### 4.3 Triggering Methods :

- In the latches and flip-flops, we use the additional signal called clock signal.
- Depending on which portion of the clock signal the latch or flip-flop responds to, we can classify them into two types :
  - Level triggered circuits
  - Edge triggered circuits

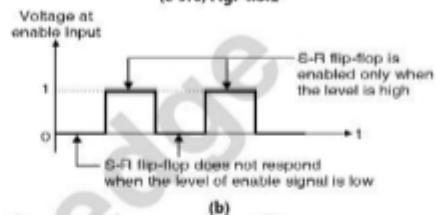
#### 4.3.1 Concept of Level Triggering :

##### Definition :

- The latch or flip-flop circuits which respond to change in their inputs, only if their enable input (E) held at an active level which may be either HIGH or LOW level are called as **level triggered** latches or flip-flops.
- Thus these circuits do not respond at the rising or falling edges of clock.
- They only respond to the steady HIGH or LOW levels of the clock signal.
- Fig. 4.3.1(a) shows the symbol of a level triggered SR flip flop and Fig. 4.3.1(b) shows the clock signal applied at its input.



(c-576) Fig. 4.3.1(a)



(c-576) Fig. 4.3.1(b) : Concept of level triggering

#### 4.3.2 Types of Level Triggered Flip-flops :

- There are two types of level triggered flip-flops :
  - Positive level triggered.
  - Negative level triggered.

##### Positive level triggered :

- If the outputs of a flip-flop respond to the input changes, only when its clock input is at HIGH (1) level, then it is called as the positive level triggered flip flop.
- The block diagram shown in Fig. 4.3.1(a) is a positive level triggered SR FF.

##### Negative level triggered FF :

- If the outputs of a flip-flop respond to the input changes, only when its clock input is at LOW (0) level, then it is called as the negative level triggered flip-flop.

**Note :** The level triggering is not used practically, due to some of its disadvantages.

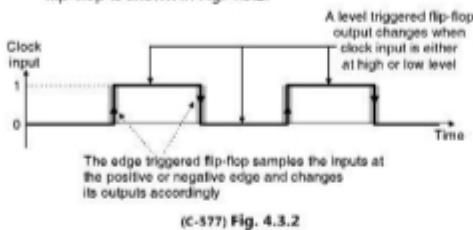
#### 4.3.3 Concept of Edge Triggering :

##### Definition :

- The flipflops which change their outputs only corresponding to the positive (rising) or negative (falling) edge of the clock input are called as **edge triggered flipflops**.
- These flip-flops are therefore said to be edge sensitive or edge triggered rather than level triggered.



- The rectangular signal applied to the clock input of a flip-flop is shown in Fig. 4.3.2.



(C-577) Fig. 4.3.2

- If the same signal is applied as the clock signal to an edge triggered flip flop, then its outputs will change only at either rising (positive) edge or at the falling (negative) edge of the clock.
- The edge triggered flip-flops do not respond to the steady state high or low level in the clock signal at all.

#### 4.3.4 Types of Edge Triggered Flip Flops :

- There are two types of edge triggered flip flops :
  - Positive edge triggered flip flops
  - Negative edge triggered flip flops.
- Positive edge triggered flip flops will allow its outputs to change in response to its inputs only at the instants corresponding to the rising edges of clock (or positive spikes).
- Its outputs will not respond to change in inputs at any other instant of time.
- Negative edge triggered flip flops will respond only to the negative going edges (or spikes) of the clock.

#### 4.4 Gated Latches (Level Triggered SR Flip Flop) :

- We have discussed the RS latches using the NAND and NOR gates.
- Now 2 more NAND gates are added to the basic SR latch and one more input called enable (E) is added, in order to obtain a gated SR latch or a level triggered SR flip-flop.
- A level voltage (0 or 1) or a clock signal can be applied to the enable (E) input.
- These flipflops will respond to the inputs if and only if we apply an **active level** at the enable input.
- This active level can be either 0 or 1 depending on the type of flip-flop.

- Such flipflops are called as level triggered flipflops or gated latches or clocked flipflops.

#### 4.4.1 Types of Level Triggered (Clocked) Flip Flops :

- There are two types of level triggered latches :
  - Positive level triggered.
  - Negative level triggered.

#### 4.5 The Gated S-R Latch (Clocked S-R Flip Flop) :

##### 4.5.1 Positive Level Triggered SR Flip-flop :

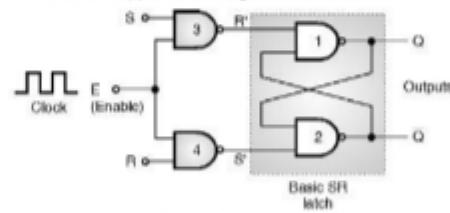
SPPU : May 19

###### University Questions

Q. 1 Draw and explain SR flip flop using NAND gates.  
(May 19, 2 Marks)

###### Logic diagram :

- The gated S-R latch is shown in Fig. 4.5.1. It is also called as clocked SR flip flop.
- It is basically the S-R latch using NAND gates with an additional "enable" (E) input. It is also called as **level triggered S-R FF**.
- The outputs of basic S-R latch used to change instantly in response to any change made at the input. But this does not happen with the gated S-R latch.



(C-578) Fig. 4.5.1 : Gated S - R latch

- For this circuit, the change in output will take place if and only if the enable input (E) is made active.
- This circuit being positive level triggered, will respond to changes in input only if the enable input is held at logic 1 level.
- In short this circuit will operate as an S-R latch if E = 1 (Enable input is active) but there is no change in the outputs if E = 0 (Enable input is inactive).

**Operation :****Case I :  $S = X, R = X, E = 0$  (No change)**

- Since enable  $E = 0$ , the outputs of NAND gates 3 and 4 will be forced to be 1 irrespective of the values of  $S$  and  $R$ .
- That means  $R' = S' = 1$ . These are the inputs of the basic S-R latch enclosed in the dotted box in Fig. 4.5.1.
- Hence the outputs of NAND latch i.e.  $Q$  and  $\bar{Q}$  will not change. Thus if  $E = 0$ , then there is no change in the output of the gated S-R latch.

**Case II :  $S = R = 0, E = 1$  : No change**

- If  $S = R = 0$  then outputs of NAND gates 3 and 4 are forced to become 1.
- Hence  $R'$  and  $S'$  both will be equal to 1. Since  $S'$  and  $R'$  are the inputs of the basic S-R latch using NAND gates, there will be no change in the state of outputs.
- Thus for  $S = R = 0$  the output state of this flip-flop remains unchanged.

**Case III :  $S = 0, R = 1, E = 1$  (Reset)**

- Since  $S = 0$ , output of NAND-3 i.e.  $R' = 1$ . And as  $R = 1$  and  $E = 1$  the output of NAND-4 i.e.  $S' = 0$ .
- Hence  $Q_{n+1} = 0$  and  $\bar{Q}_{n+1} = 1$ . This is the reset condition.

**Case IV :  $S = 1, R = 0, E = 1$  (Set)**

- Output of NAND 3 i.e.  $R' = 0$  and output of NAND 4 i.e.  $S' = 1$ .
- Hence output of S-R NAND latch is  $Q_{n+1} = 1$  and  $\bar{Q}_{n+1} = 0$ .
- This is the set condition.

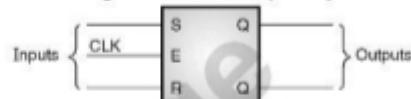
**Case V :  $S = 1, R = 1, E = 1$  (RACE)**

- As  $S = 1, R = 1$  and  $E = 1$ , the outputs of NAND gates 3 and 4 both are 0. i.e.  $S' = R' = 0$ .
- Hence the "Race" condition will occur in the basic NAND latch. This operation should be avoided as both  $Q$  and  $\bar{Q}$  will try to become 1 at the same time as discussed earlier.

$\bar{Q}$  will try to become 1 at the same time as discussed earlier.

**Symbol and truth table :**

- The symbol and truth table of the gates S-R latch are as shown in Figs. 4.5.2(a) and (b) respectively.



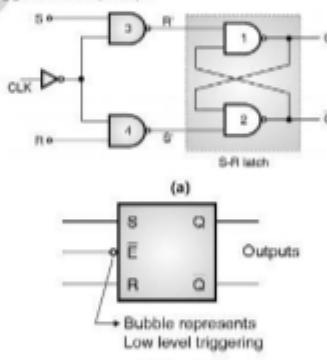
(c-579) Fig. 4.5.2(a) : Symbol for S - R latch

(c-7827) Fig. 4.5.2(b) : Truth table of gated S - R latch

Case	Enable E	S	R	Outputs		Comments
				$Q_{n+1}$	$\bar{Q}_{n+1}$	
I	0	X	X	$Q_n$	$\bar{Q}_n$	No change as E = 0
II	1	0	0	$Q_n$	$\bar{Q}_n$	No change (NC)
III	1	0	1	0	1	Reset condition
IV	1	1	0	1	0	Set condition
V	1	1	1	RACE (Indeterminate)		Avoid this condition

**4.5.2 Negative Level Triggered SR Flip Flop :****Logic diagram :**

- Fig. 4.5.3(a) shows the circuit diagram of a negative level triggered SR flip-flop.



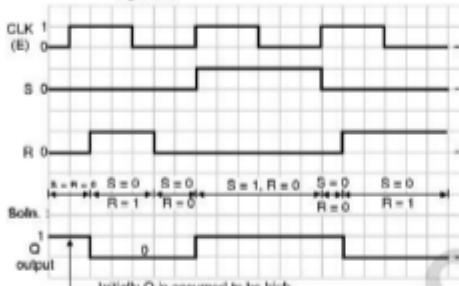
(c-580) Fig. 4.5.3 : Negative level triggered S - R latch

- It is the same circuit that we discussed in the previous section with only one additional inverter.
- Due to the additional inverter connected to the enable terminal, this circuit becomes sensitive to the low level (0) applied to the enable input. Hence it will enable the outputs if  $E = 0$ .



- If a square wave is applied to the enable input then the latch output will respond to input changes only when the square input is at its low (0) level.
- Fig. 4.5.3(b) shows the symbol of negative level triggered S-R latch. Note that there is a bubble added to the enable input which is active low input.

**Ex. 4.5.1:** Draw output waveform for the following input signals.



(c-3a2) Fig. P. 4.5.1 : Waveforms for gated S-R latch

#### 4.5.3 Disadvantage of S-R Latch :

- From the truth tables of the S-R latch using NOR and NAND gates we can draw the following important conclusion :
  - When  $S = R = 0$  or  $S = R = 1$ , the outputs  $Q$  and  $\bar{Q}$  either don't change (NC) or they are indeterminate (invalid) due to race condition.
  - This disadvantage of S-R latch can be overcome by using the gated D latch discussed in the next section.

#### 4.5.4 Application of S-R Latch :

It is used as electronic timer.

#### 4.6 The Gated D Latch (Clocked D Flip Flop) :

SPPU : May 07

##### University Questions

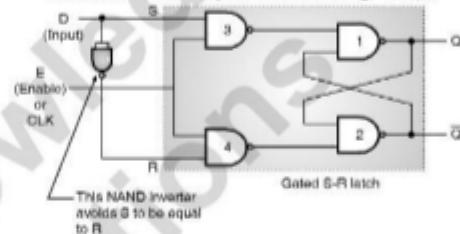
**Q. 1** Justify names delay for D flip-flop giving truth table.  
(May 07, 2 Marks)

- In some applications, the S and R inputs will always be complementary. i.e. when  $S = 0$ ,  $R = 1$  and when  $S = 1$ ,  $R = 0$ . That means  $S = \bar{R}$ .

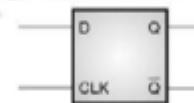
- For such applications we can use the gated D latch.

##### Logic diagram :

- The circuit diagram of the gated D latch is shown in Fig. 4.6.1(a) and its logic symbol is shown in Fig. 4.6.1(b).
- This is also called as level triggered D flip-flop or clocked D flip-flop.
- Note that D latch is the simple gated S-R latch with a small modification. A NAND inverter is connected between its S and R inputs as shown in Fig. 4.6.1(a).



(a) Gated D latch



(b) Logic symbol of gated D latch

(c-3a19) Fig. 4.6.1

- This latch has only one input denoted by D.
- Due to the NAND inverter, S and R inputs will always be the complements of each other. Hence the input conditions such as  $S = R = 0$  or  $S = R = 1$ , will never appear.
- This will avoid the problems associated with  $SR = 00$  and  $SR = 11$  conditions of SR flip-flop.

##### Truth table :

- Truth table for the gated D latch is given in Table 4.6.1.

(c-a061) Table 4.6.1 : Truth table for the gated D latch

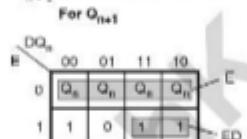
Inputs		Outputs		Comment
E	D	$Q_{n+1}$	$\bar{Q}_{n+1}$	
0	X	$Q_n$	$\bar{Q}_n$	No change (NC)
1	0	0	1	Reset condition
1	1	1	0	Set condition

**Operation :**

- If  $E = 0$  then the latch is disabled. Hence there is no change in output.
- If  $E = 1$  and  $D = 0$  then  $S = 0$  and  $R = 1$ . Hence irrespective of the present state the, next state is  $Q_{n+1} = 0$  and  $\bar{Q}_{n+1} = 1$ . This is the reset condition.
- On the other hand if  $E = 1$  and  $D = 1$ , then  $S = 1$  and  $R = 0$ . This will set the latch and  $Q_{n+1} = 1$ ,  $\bar{Q}_{n+1} = 0$  irrespective of the present state.
- From the truth table it is evident that  $Q$  output is same as  $D$  input, in otherwords  $Q$  output follows the  $D$  input.
- If  $D = 0$  then  $Q = 0$  and if  $D = 1$  then  $Q = 1$ . However output follows input after some propagation delay hence the other name of the  $D$  flip-flop is delay flip-flop.

**Characteristic equation for D latch :**

- Refer to the truth table of  $D$  latch and write the K-map for  $Q_{n+1}$  as shown in Fig. 4.6.2.



(C-584) Fig. 4.6.2 : K-map for D latch

- After simplification we get the characteristic equation of  $D$  latch as,

$$Q_{n+1} = E \cdot D + \bar{E}$$

#### 4.7 Gated JK Latch (Level Triggered JK Flip Flop) :

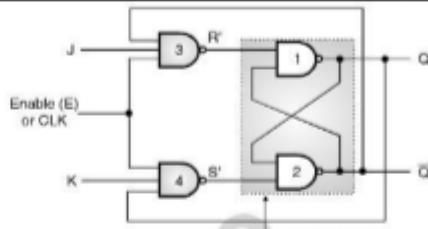
SPPU : Dec. 13

**University Questions**

- Q. 1 Draw and explain the diagram of JK flip-flop using NAND gates and explain how race around condition is avoided ? (Dec. 13, 6 Marks)

**Logic diagram :**

- The JK latch using NAND gates is shown in Fig. 4.7.1(a).
- It consists of the basic SR latch and an enable input. It is also called as level triggered JK flip-flop.



(a) A level triggered JK flip-flop



(b) Symbol of the gated JK latch

(C-3420) Fig. 4.7.1

- Note the outputs  $Q$  and  $\bar{Q}$  have been fed back and connected to the inputs of NAND gates 4 and 3 respectively.
- The JK latch of Fig. 4.7.1(a) responds to the input changes if a positive level is applied at the enable (E) input. Hence it is a positive level triggered latch.

**Operation :**

- The operation of NAND JK latch is exactly identical to that of the positive edge triggered JK flip flop discussed in section 4.10.1.
- The only difference between them is that, this circuit is level triggered.
- The operation of NAND JK latch has been summarized in Table 4.7.1.

(C-3420(a)) Table 4.7.1 : Truth table of positive level triggered JK latch

Case No.	Inputs			Outputs		State
	E	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$	
Case I	0	x	x	$Q_n$	$\bar{Q}_n$	No change
Case II	1	0	0	$Q_n$	$\bar{Q}_n$	No change
Case III	1	0	1	0	1	Reset
Case IV	1	1	0	1	0	Set
Case V	1	1	1	$Q_n$	$\bar{Q}_n$	Toggle



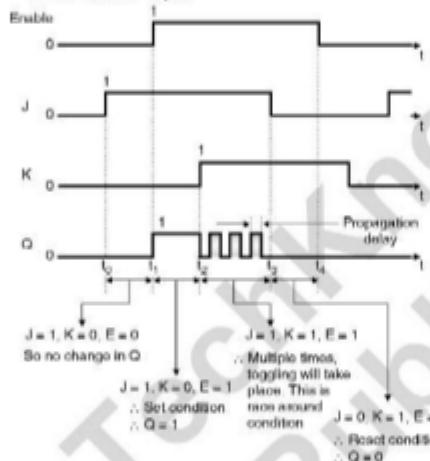
#### 4.7.1 Race Around Condition in JK Latch :

SPPU : May 06, May 07, Dec. 08, Dec. 09

##### University Questions

- Q. 1** Explain race around condition in J-K flip-flop.  
**(May 06, Dec. 08, 4 Marks)**
- Q. 2** Explain race around condition. Mention solution for it.  
**(May 07, 4 Marks)**
- Q. 3** What is race around condition ?  
**(Dec. 09, 3 Marks)**

- The "Race Around Condition" that we are going to explain occurs when  $J = K = 1$  i.e. when the latch is in the toggle mode.
- Refer Fig. 4.7.2 which shows the waveforms for the various modes, when a rectangular waveform is applied to the "Enable" input.



(C-SSE) Fig. 4.7.2 : Waveform for various modes of a JK latch

##### Interval $t_0 - t_1$ :

- During this interval  $J = 1$ ,  $K = 0$  and  $E = 0$ .
- Hence the latch is disabled and there is no change in  $Q$ .

##### Interval $t_1 - t_2$ :

During this interval  $J = 1$ ,  $K = 0$  and  $E = 1$ .

Hence this is a set condition and  $Q$  becomes 1.

##### Interval $t_2 - t_3$ : Race around

- At instant  $t_2$ ,  $J = K = 1$  and  $E = 1$ . Hence the JK latch is in the toggle mode and  $Q$  becomes low (0) and  $\bar{Q} = 1$ .

- These changed outputs get applied at the inputs of NAND gates 3 and 4 of the JK latch. Thus the new inputs to Gates 3 and 4 are:

- NAND - 3 :  $J = 1$ ,  $E = 1$ ,  $\bar{Q} = 1$ .
- NAND - 4 :  $K = 1$ ,  $E = 1$ ,  $Q = 0$ .
- Hence  $R'$  will become 0 and  $S'$  will become 1.
- Therefore after a time period corresponding to the propagation delay, the  $Q$  and  $\bar{Q}$  outputs will change to,  $Q = 1$  and  $\bar{Q} = 0$ .
- These changed output again get applied to the inputs of NAND-3 and 4 and the outputs will toggle again.
- Thus as long as  $J = K = 1$  and  $E = 1$ , the outputs will keep toggling indefinitely as shown in Fig. 4.7.2.
- This multiple toggling in the J-K latch is called as Race Around condition. It must be avoided.

##### Interval $t_3 - t_4$ :

- During this interval  $J = 0$ ,  $K = 1$  and  $E = 1$ . Hence it is the reset condition.
- So  $Q$  becomes zero.

##### How to avoid race around condition ?

The race around condition in JK latch can be avoided by:

- Using the edge triggered JK flip flop.
- Using the master slave JK flip flop.

#### 4.7.2 Difference between Latch and Flip-flop :

##### Latches :

- Latches and flip flops both are basically the bistable elements.
- As discussed earlier a latch has got an enable input. As long as it is active, the latch output will keep changing according to the changes in its input. In other words latch is a level triggered flip flop.

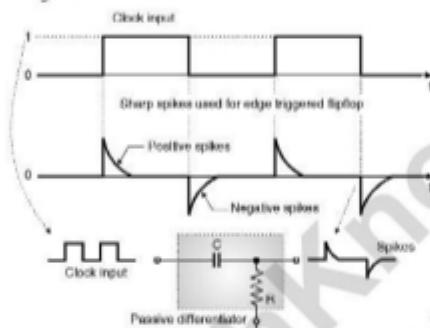
##### Flip-flop :

- But flip flop is a sequential circuit which generally samples its inputs and changes its outputs only at particular instants of time and not continuously.
- The flip-flops are therefore said to be edge sensitive or edge triggered rather than being level triggered like latches.



#### 4.8 Edge Triggered Flip Flops :

- We have already discussed the concept of edge triggering.
- For the edge triggered flip flops, it is necessary to apply the clock signal (timing signal) in the form of sharp positive and negative spikes instead of in the form of a rectangular pulse train.
- Such sharp spikes are shown in Fig. 4.8.1. These spikes can be derived from the rectangular clock pulses with the help of a passive differentiator circuit shown in Fig. 4.8.1.



(C-587) Fig. 4.8.1 : Use of differentiator to obtain sharp edges

- Thus the passive differentiator acts as a pulse shaping circuit.

#### 4.8.1 Types of Edge Triggered Flip Flops (Clocked FFs) :

- There are two types of edge triggered flip flops :
  1. Positive edge triggered flip flops
  2. Negative edge triggered flip flops.
- Positive edge triggered flip flops will allow its outputs to change in response to its inputs only at the instants corresponding to the rising edges of clock (or positive spikes).
- Its outputs will not respond to change in inputs at any other instant of time.
- Negative edge triggered flip flops will respond only to the negative going edges (or spikes) of the clock.

#### 4.8.2 Positive Edge Triggered S-R Flip Flop :

SPPU : Dec. 11

##### University Questions

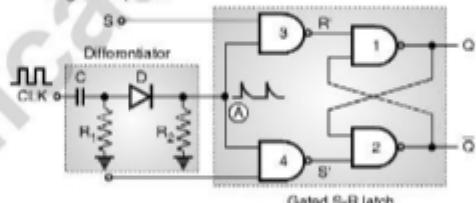
- Q. 1** Draw and explain SR flip-flop using NAND gates.  
(Dec. 11, 8 Marks)

##### Circuit diagram :

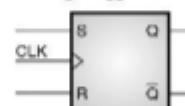
- Fig. 4.8.2(a) shows the circuit diagram of a positive edge triggered S-R flip flop and Fig. 4.8.2(b) shows the logic symbol for it.

##### Operation :

- Note that the SR flip flop of Fig. 4.8.2(a) consists of a differentiator circuit and a gated S-R latch (level triggered SR flip-flop) which we have already discussed.
- $C - R_1$  acts as a differentiator and converts the rectangular clock pulses into positive and negative spikes.
- The diode acts as rectifying diode and allows only the positive spikes to pass through it while, blocking the negative spikes.



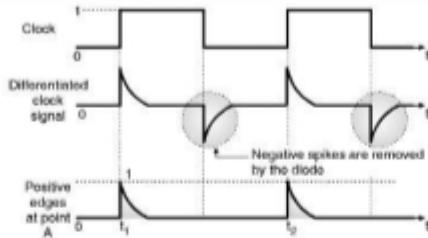
(a) Positive edge triggered S-R flip flop



(b) Logic symbol

(C-588) Fig. 4.8.2

- Thus we get only the positive spikes, corresponding to the leading edges of the clock signal, as shown in Fig. 4.8.3 at point A which is enable input of the gated S-R latch.
- Due to these sharp positive spikes applied at point "A", the gated S-R latch in the S-R flip flop will be enabled only for a short duration of time when the positive spike is present at A. (at instants  $t_1, t_2, \dots$  in Fig. 4.8.3)



(C-589) Fig. 4.8.3 : Generation of positive triggering pulses

- At these instants, the flip-flop behaviour is exactly identical to that of the enabled gated S-R latch (enabled level triggered SR flip-flop).
- The truth table of positive edge triggered S-R flip flop will also be identical to that of the gated S-R latch with only one change.
- The "enable" input will now be replaced by clock input. And the outputs will change only if a positive edge is applied to the clock input.
- The positive clock edge is denoted by an arrow ( $\uparrow$ ) in Table 4.8.1.

**Truth table :**

- The truth table of a positive edge triggered SR flip flop is shown in Table 4.8.1.

(C-6260) Table 4.8.1 : Truth table of a positive edge triggered SR flip flop

Inputs			Outputs		Remark	
CLK	S	R	$Q_{n+1}$	$\bar{Q}_{n+1}$		
FF is disabled	0	x	x	$Q_n$	$\bar{Q}_n$	No change (NC)
	1	x	x	$Q_n$	$\bar{Q}_n$	No change (NC)
	↓	x	x	$Q_n$	$\bar{Q}_n$	No change (NC)
	↑	0	0	$Q_n$	$\bar{Q}_n$	No change (NC)
	↑	0	1	0	1	Reset
	↑	1	0	1	0	Set
	↑	1	1	Race	Race	Avoid

↓ = Negative edge of clock, ↑ = Positive edge of clock

- Note that this flip-flop does not respond to the positive or negative levels in the clock signal.
- Similarly it does not respond to the negative edges of the clock.
- This flip flop will respond only to the positive edges of clock signal.

- With positive edge of the clock, the SR flip flop behaves in the following way.

$S = R = 0 \rightarrow$  No change in output

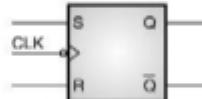
$S = 0, R = 1 \rightarrow Q_{n+1} = 0, \bar{Q}_{n+1} = 1$  Reset condition

$S = 1, R = 0 \rightarrow Q_{n+1} = 1, \bar{Q}_{n+1} = 0$  Set condition

$S = R = 1 \rightarrow$  Race condition.

**4.8.3 Negative Edge Triggered S-R Flip Flop :****Symbol and Truth table :**

- The internal circuit (with NAND gates) of the negative edge triggered S-R flip flop is exactly same as that for the positive edge triggered SR flip-flop discussed earlier.
- The differentiator circuit is slightly modified in order to make this flip flop respond to the negative (falling) edges of the clock input.
- Fig. 4.8.4 shows the circuit symbol of the negative edge triggered S-R flip flop and Table 4.8.2 shows its truth table.



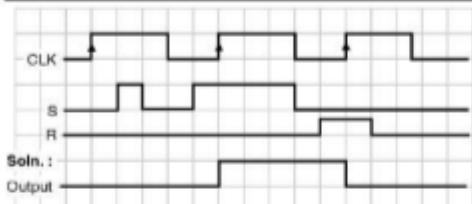
(C-590(a)) Fig. 4.8.4 : Circuit symbol of negative edge triggered SR FF

(C-6207) Table 4.8.2 : Truth table of negative edge triggered S-R FF

Inputs			Outputs		State	
CLK	S	R	$Q_{n+1}$	$\bar{Q}_{n+1}$		
> FF is disabled	0	x	x	$Q_n$	$\bar{Q}_n$	No change (NC)
	1	x	x	$Q_n$	$\bar{Q}_n$	No change (NC)
	↓	x	x	$Q_n$	$\bar{Q}_n$	No change (NC)
	↑	0	0	$Q_n$	$\bar{Q}_n$	No change (NC)
	↑	0	1	0	1	Reset
	↑	1	0	1	0	Set
	↑	1	1	Race	Race	Avoid

↑ = Positive edge of clock, ↓ = Negative edge of clock

- Ex. 4.8.1 :** Draw the waveform for the Q output for the input waveforms shown in Fig. P. 4.8.1 for the positive edge triggered SR flip-flop.



(c-591) Fig. P. 4.8.1

#### 4.9 Edge Triggered D Flip Flop :

- The edge triggered D flip flops can be of two types :
  - Positive edge triggered D flip flop
  - Negative edge triggered D flip flop.
- These flip flops can be derived from the level triggered D latch which we have discussed in section 4.6.

##### 4.9.1 Positive Edge Triggered D Flip Flop :

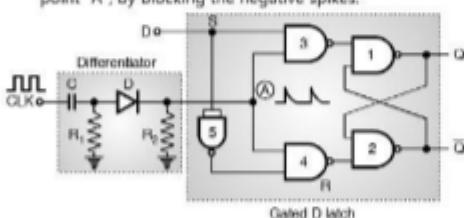
SPPU : May 07

###### University Questions

- Q. 1** Justify names delay for D flip-flop giving truth table and waveforms. **(May 07, 2 Marks)**

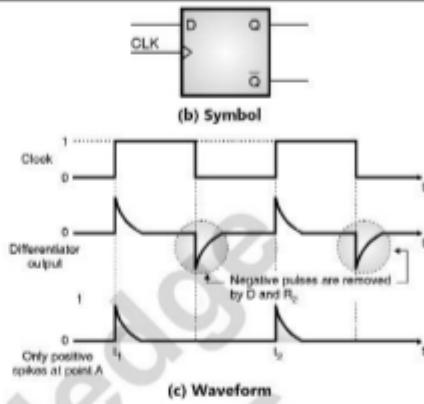
###### Logic diagram :

- Fig. 4.9.1(a) shows the positive edge triggered D flip flop. It consists of a gated D latch and a differentiator circuit. The symbol is as shown in Fig. 4.9.1(b).
- The clock pulses are applied to the circuit through a differentiator formed by  $R_1C$  and a rectifier circuit consisting of diode D and  $R_2$ .
- The NAND gates 1 through 5 form a D latch.
- The differentiator converts the clock pulses into positive and negative spikes and the combination of D and  $R_2$  will allow only the positive spikes to pass through to point "A", by blocking the negative spikes.



(a) Positive edge triggered D-flip flop

(c-592(a)) Fig. 4.9.1



(c-592(a)) Fig. 4.9.1

###### Operation :

- The operation of edge triggered D FF is exactly same as that of the gated D latch discussed earlier with only one difference.
- The D latch had an enable terminal whereas the D flip flop has a clock input.
- The gated D latch of Fig. 4.9.1 is enabled by the positive spikes obtained at point A.
- Hence the outputs will change based on the inputs at these particular instants only.
- Thus the edge triggered D flip-flop responds only to the positive (leading) edges of the clock pulses.
- At any other instants of time, the D flip flop will not respond to the changes in input.

###### Truth table :

- Table 4.9.1 shows the truth table of a positive edge triggered D flip flop.

(c-592) Table 4.9.1 : Truth table of a positive edge triggered D flip flop

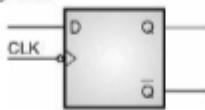
Inputs	Outputs		State		
	CLK	D	$Q_{n+1}$	$\bar{Q}_{n+1}$	
FF disabled	D	x	$Q_n$	$\bar{Q}_n$	No change
	1	x	$Q_n$	$\bar{Q}_n$	No change
	↓	x	$Q_n$	$\bar{Q}_n$	No change
FF responds only to the positive edges	↑	0	0	1	$Q$ follows D input
	↑	1	1	0	

From the truth table it is clear that the Q output of the flip flop follows the D input.

### 4.9.2 Negative Edge Triggered D Flip Flop :

#### Symbol :

- The symbol for negative edge triggered D flip flop is shown in Fig. 4.9.2.



(C-593) Fig. 4.9.2 : Symbol of negative edge triggered D flip flop

- This F/F responds only to the negative edges of the clock pulses. This is how it is different from the positive edge triggered flip flop.
- Otherwise, the operation of the negative edge triggered D flip flop is exactly same as that of the positive edge triggered D flip flop.

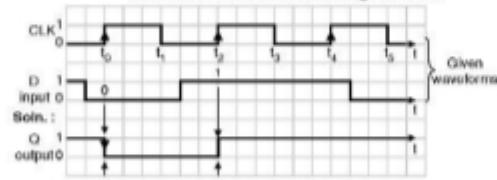
#### Truth table :

- The truth table of negative edge triggered D flip flop is shown in Table 4.9.2.

(C-6209) Table 4.9.2 : Truth table of negative edge triggered D flip flop

Inputs		Outputs		Comment
CLK	D	$Q_{n+1}$	$\bar{Q}_{n+1}$	
x	D	$Q_n$	$\bar{Q}_n$	No change
t	x	$Q_n$	$\bar{Q}_n$	No change
↑	x	$Q_n$	$\bar{Q}_n$	No change
↓	0	0	1	$Q$ output follows D input
↓	1	1	0	

Ex. 4.9.1 : Draw the output waveform for a positive edge triggered D flip flop, if the clock and D input waveforms are as shown in Fig. P. 4.9.1.



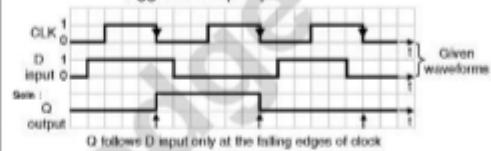
The Q output is equal to D input corresponding to every rising edge of the clock signal

(C-594) Fig. P. 4.9.1

- At instant  $t = t_0$ , the first positive edge of the clock arises, and Q output follows the D input to become zero.

- Similarly at  $t = t_1$ , Q output follows the D input at that instant.
- Note that Q output does not respond to any change in D input at any other instant time.

Ex. 4.9.2 : For the given waveforms in Fig. P. 4.9.2. Draw the output (Q) waveform for the negative edge triggered D flip flop.



(C-595) Fig. P. 4.9.2

### 4.10 Edge Triggered J-K Flip Flop :

Edge triggered J-K flip flops are of two types :

- Positive edge triggered JK flip flop.
- Negative edge triggered JK flip flop.

#### 4.10.1 Positive Edge Triggered JK Flip Flop :

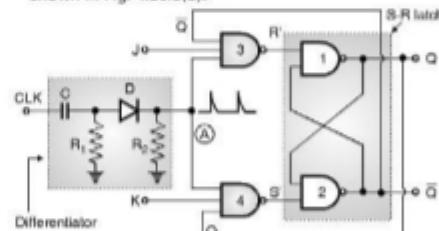
SPPU : Dec. 06

##### University Questions

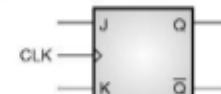
Q. 1 Draw and explain operation of JK flip-flop using logic gates with waveforms. (Dec. 06, 6 Marks)

##### Logic diagram :

- The circuit diagram of the positive edge triggered JK flip flop is shown in Fig. 4.10.1(a) and its circuit symbol is shown in Fig. 4.10.1(b).



(a) Positive edge triggered JK flip flop



(b) Symbol

(C-596) Fig. 4.10.1

**Operation :**

- The clock signal is a train of rectangular or square waves.
- It is passed through a differentiator ( $C = R_1$ ) and a rectifier (D and  $R_2$ ), to obtain only the positive spikes at point A as already discussed. The negative spikes are blocked by the diode.
- NAND gates 1 and 2 form the basic S-R latch. The other two NAND gates (3 and 4) have three inputs each. The outputs  $Q$  and  $\bar{Q}$  are fed back to the inputs of gates 4 and 3 respectively.
- Referring to Fig. 4.10.1(a) we can write the mathematical expressions for  $S'$  and  $R'$  as follows.

$$S' = \overline{K \cdot Q \cdot CLK}$$

$$\text{and } R' = \overline{J \cdot \bar{Q} \cdot CLK}$$

- Let us now understand the operation step by step.

**Case I : CLK = 0 or 1 i.e. level**

- If  $CLK = 0$  or 1 i.e. level and no pulse this flip flop is disabled, because the output of differentiator is zero for any level input.
- Therefore  $Q$  and  $\bar{Q}$  output do not change their state.

**Case II : If clock edge is falling edge ( $\downarrow$ )**

- For the falling edge of the clock, the rectifier (D -  $R_2$ ) will block the negative spike. Hence voltage at point A is logic 0.
- So one input to both the NAND gates 3 and 4 is 0.
- Hence both  $S'$  and  $R'$  will be forced to 1.
- For an SR latch using NAND gates, the outputs will remain unchanged if  $S = R = 1$ .
- Corresponding to the falling edge of clock, the outputs  $Q$  and  $\bar{Q}$  will remain unchanged.

**Case III : CLK =  $\uparrow$ , J = 0, K = 0 (No change)**

- Since  $J = 0$ ,  $R' = 1$  and as  $K = 0$ ,  $S' = 1$ .
- As  $S' = R' = 1$  the outputs  $Q$  and  $\bar{Q}$  will not change their state even though a positive edge of the clock pulse is being applied.

$$\therefore J = 0, K = 0, Q_{n+1} = Q_n \text{ and } \bar{Q}_{n+1} = \bar{Q}_n$$

No change in output.

**Case IV : CLK =  $\uparrow$ , J = 0, K = 1 (Reset)**

- Recall the expressions for  $S'$  and  $R'$ ,

$$S' = \overline{K \cdot Q \cdot CLK} \quad \text{and} \quad R' = \overline{J \cdot \bar{Q} \cdot CLK}$$

- If the previous state of  $Q$  and  $\bar{Q}$  is  $Q = 1$  and  $\bar{Q} = 0$  then

$$S' = \overline{1 \cdot 1 \cdot 1} = 0 \quad \text{and}$$

$$R' = \overline{0 \cdot 0 \cdot 1} = 1$$

- Therefore according to the operation of S-R latch if  $S' = 0, R' = 1$  then  $Q = 0$ , and  $\bar{Q} = 1$ .  
 $\therefore Q = 0$  and  $\bar{Q} = 1$
- Thus with  $J = 0$ ,  $K = 1$  and positive going clock, the JK flip flop will **reset**.

- If  $Q = 0$  and  $\bar{Q} = 1$  before the application of clock pulse, then there will not be any change in their state.

**Case V : CLK =  $\uparrow$ , J = 1, K = 0 (Set)**

- If the previous state of  $Q$  and  $\bar{Q}$  is  $Q = 0$  and  $\bar{Q} = 1$  then,

$$S' = \overline{0 \cdot 0 \cdot 1} = 1 \quad \text{and} \quad R' = \overline{1 \cdot 1 \cdot 1} = 0$$

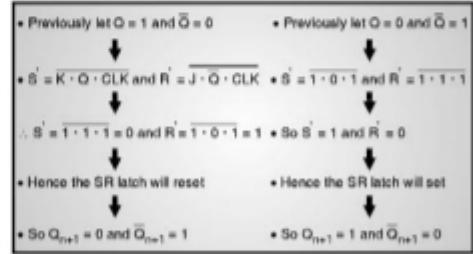
- Hence according to the operation of S-R latch, if  $S' = 1$  and  $R' = 0$  then,

$$Q = 1 \text{ and } \bar{Q} = 0 \quad \dots \text{i.e. the flip flop is set.}$$

- If  $Q$  and  $\bar{Q}$  are already 1,0 then there will not be any change in the output state.

**Case VI : CLK =  $\uparrow$ , J = 1, K = 1 Toggle mode**

- We will discuss this case for two different previous cases. (C-6399)



- From the operation discussed above, we conclude that when  $J = K = 1$  and a positive clock edge is applied, then  $Q$  and  $\bar{Q}$  outputs are inverted i.e.  $Q_{n+1} = \bar{Q}_n$  and  $\bar{Q}_{n+1} = Q_n$ .



- This is called as the **TOGGLING** mode. This is a very important mode of operation.

#### Truth table of JK flip flop :

(C-6837) Table 4.10.1 : Truth table for positive edge triggered JK FF

Case No.	Inputs			Outputs		State
	CLK	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$	
Case I	0 or 1	x	x	No change	No change	
Case II	↓	x	x	No change	No change	
Case III	↑	0	0	No change	No change	Flip flop is disabled
Case IV	↑	0	1	0	1	Reset
Case V	↑	1	0	1	0	Set
Case VI	↑	1	1	$\bar{Q}_n$	$Q_n$	Toggle

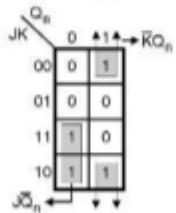
#### 4.10.2 Characteristic Equation of JK Flip Flop :

- Refer to the truth table of JK flip flop (Table 4.10.2) and write the K-map for  $Q_{n+1}$  as shown in Fig. 4.10.2.

(C-8062) Table 4.10.2 : 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

For  $Q_{n+1}$



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

(C-597) Fig. 4.10.2 : K-map for  $Q_{n+1}$

∴ The characteristic equation is  $Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

#### 4.10.3 How does an Edge Triggered JK FF Avoid Race Around Condition ?

SPPU : Dec. 07, Dec. 08, Dec. 09

##### University Questions

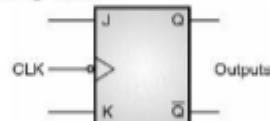
- Q. 1 Discuss methods to avoid race around condition in JK flip-flop. (Dec. 07, 4 Marks)
- Q. 2 How is race around condition is avoided ? (Dec. 08, Dec. 09, 3 Marks)

- For the race around to take place, it is necessary to have the enable input high along with  $J = K = 1$ .
  - As the enable input remains high for a long time in a JK latch, the problem of multiple toggling arises which is known as Race condition.
  - But in edge triggered JK flip flop, the positive clock pulse is present only for a very short time because it is in the form of a narrow differentiated spike.
  - Hence by the time the toggled outputs ( $Q$  and  $\bar{Q}$ ) return back to the inputs of NAND gates 3 and 4, the positive clock spike has died down to zero. Hence the multiple toggling cannot take place.
- Thus the edge triggering avoids the race around condition.

#### 4.10.4 Negative Edge Triggered JK flip-flop :

##### Symbol :

- The symbol for negative edge triggered JK flip-flop is shown in Fig. 4.10.3.



(C-598) Fig. 4.10.3 : Symbol of negative edge triggered JK FF

- This FF responds only to the negative edges of the clock pulses.
- The rest of the operation of this FF is exactly same as that of the positive edge triggered JK FF.

**Truth table :**

- The truth table of negative edge triggered JK FF is as shown in Table 4.10.3.

(C-688) Table 4.10.3 : Truth table of negative edge triggered JK FF

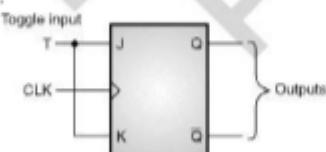
Inputs			Outputs		State
CLK	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$	
0 or 1	x	x	$Q_n$	$\bar{Q}_n$	NC (FF is disabled)
↑	x	x	$Q_n$	$\bar{Q}_n$	
↓	0	0	$Q_n$	$\bar{Q}_n$	No change
↓	0	1	0	1	Reset
↓	1	0	1	0	Set
↓	1	1	$\bar{Q}_n$	$Q_n$	Toggle

**4.11 Toggle Flip Flop (T Flip Flop) :****4.11.1 Positive Edge Triggered T-FF :****SPPU : May 07****University Questions**

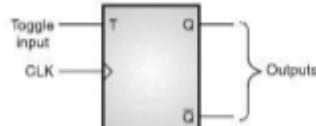
- Q. 1** Justify toggle for T flip-flop giving truth table and waveforms. **(May 07, 2 Marks)**

**Symbol :**

- Toggle flip flop is basically a JK flip flop with J and K terminals permanently connected together. It has only one input denoted by "T", as shown in Fig. 4.11.1(a).
- The symbol for positive edge triggered T flip flop is shown in Fig. 4.11.1(b) and Table 4.11.1 shows its truth table.



(a) JK FF is converted into T flip flop



(b) Logic symbol of positive edge triggered T flip flop

(C-599) Fig. 4.11.1

**Truth table :**

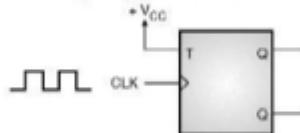
(C-689) Table 4.11.1 : Truth table of a Toggle FF (positive edge triggered)

CLK	T	Outputs		State
		$Q_{n+1}$	$\bar{Q}_{n+1}$	
↑	0	$Q_n$	$\bar{Q}_n$	No change
↓	x	$Q_n$	$\bar{Q}_n$	No change
1	x	$Q_n$	$\bar{Q}_n$	No change
0	x	$Q_n$	$\bar{Q}_n$	No change
↑	1	$\bar{Q}_n$	$Q_n$	Toggle

**Operation :**

- When  $T = 0$ ,  $J = K = 0$ . Hence the outputs  $Q$  and  $\bar{Q}$  won't change.
- But if  $T = 1$ , then  $J = K = 1$  and the outputs will toggle corresponding to every leading edge of clock signal.
- This has been illustrated in Ex. 4.11.1.

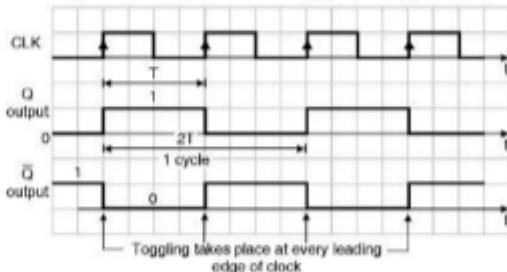
**Ex. 4.11.1 :** Determine the output Q for the set up shown in Fig. P. 4.11.1(a). What is the relation between the frequency of clock and that of Q output ?



(C-600) Fig. P. 4.11.1(a) : Given set up

**Soln. :**

- The flip flop of Fig. P. 4.11.1(a) is a toggle flip flop with  $T = 1$ . It is a positive edge triggered flip flop.
- Hence the outputs will toggle in response to each rising edge of the clock, as shown in Fig. P. 4.11.1(b).



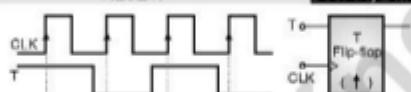
(C-601) Fig. P. 4.11.1(b) : Waveforms for a T F/F

**Relation between input and output frequency :**

- Referring to Fig. P. 4.11.1(b),  
1 cycle period of CLK signal =  $T$ ,  
 $\therefore$  Clock frequency  $f_{CLK} = \frac{1}{T}$
- 1 cycle period of Q output =  $2T$ ,  
 $\therefore$  Output frequency  $f_Q = \frac{1}{2T}$  But,  $(\frac{1}{T}) = f_{CLK}$   
 $\therefore$  Output frequency  $f_Q = \frac{f_{CLK}}{2}$  ...Ans.

The T flip flop divides the clock frequency by 2. Hence a T flip flop can be used as a frequency divider.

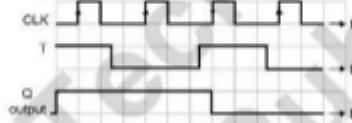
**Ex. 4.11.2 :** Refer Fig. P. 4.11.2(a) and determine the Q output waveform if the flip flop starts out RESET. **Dec. 04, 2 Marks**



(C-1344) Fig. P. 4.11.2(a)

**Soln. :**

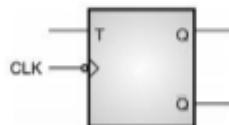
- The output waveform is shown in Fig. P. 4.11.2(b).



(C-1345) Fig. P. 4.11.2(b)

**4.11.2 Negative Edge Triggered T Flip Flop :****Symbol :**

- Fig. 4.11.2 shows the logic symbol of a negative edge triggered toggle flip flop and Table 4.11.2 gives its truth table.



(C-602) Fig. 4.11.2 : Logic symbol of a negative edge triggered toggle flip flop

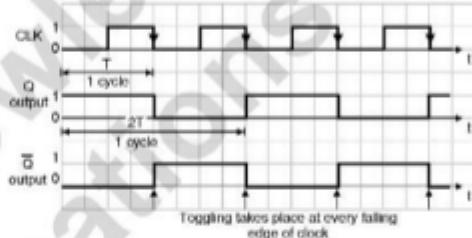
**Truth table :**

(C-6840) Table 4.11.2 : Truth table

T	CLK	Outputs		State
		$Q_{n+1}$	$\bar{Q}_{n+1}$	
0	x	$Q_n$	$\bar{Q}_n$	No change
1	↑	$Q_n$	$\bar{Q}_n$	No change
1	0	$Q_n$	$\bar{Q}_n$	No change
1	1	$Q_n$	$\bar{Q}_n$	No change
1	↓	$\bar{Q}_n$	$Q_n$	Toggle

**Waveforms :**

- The waveforms for negative edge triggered toggle FF are shown in Fig. 4.11.3.



(C-603) Fig. 4.11.3 : Waveforms for negative edge triggered T FF

- The toggle input T = 1. The outputs Q and  $\bar{Q}$  toggle corresponding to every falling edge of clock.

- This F/F also, acts as a frequency divider. The frequency of Q and  $\bar{Q}$  outputs is half the frequency of clock input.

**4.11.3 Application of T F/F :**

- The T F/F acts as the basic building block of a ripple counter.

**4.12 Master Slave (MS) JK Flip Flop :**

SPPU : Dec. 07, Dec. 08, Dec. 09, May 10,  
Dec. 16, Dec. 19

**University Questions**

- Q. 1** Discuss methods to avoid race around condition in JK flip-flop. (Dec. 07, 4 Marks)
- Q. 2** How can race around condition be avoided? (Dec. 08, Dec. 09, 3 Marks)
- Q. 3** Explain the operation of a master-slave JK flip-flop and show how the race around condition is eliminated in it? (May 10, 8 Marks)



**Q. 4** What are advantages of master-slave JK flip-flop ?  
Explain the working with a suitable diagram.

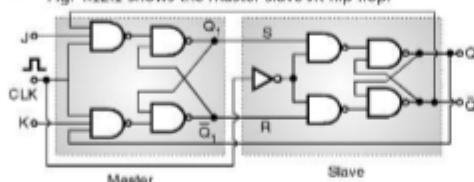
(Dec. 16, 6 Marks)

**Q. 5** Draw and explain master-slave JK flip-flop.

(Dec. 19, 6 Marks)

#### Logic diagram :

- Fig. 4.12.1 shows the master slave JK flip flop.



(c-606) Fig. 4.12.1 : Master slave JK FF

- It is a combination of a clocked JK latch (level triggered JK FF) and clocked SR latch (level triggered SRFF).
- The clocked JK latch acts as the master and the clocked SR latch acts as the slave.
- Master is positive level triggered. But due to the presence of the inverter in the clock line, the slave will respond to the negative level.
- Thus both master and slave circuits are level triggered circuits.
- Hence when the clock = 1 (positive level) the master is active and the slave is inactive.
- Whereas when clock = 0 (low level) the slave is active and the master is inactive.

#### Truth table :

- Table 4.12.1 gives truth table of master slave JK flip flop.

(c-6267) Table 4.12.1 : Truth table of master slave JK FF

Case	Inputs			Outputs		Remark
	CLK	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$	
I	X	0	0	$Q_n$	$\bar{Q}_n$	No change
II	$\overline{JL}(1)$	0	0	$Q_n$	$\bar{Q}_n$	No change
III	$\overline{JL}(1)$	0	1	0	1	Reset
IV	$\overline{JL}(1)$	1	0	1	0	Set
V	$\overline{JL}(1)$	1	1	$\bar{Q}_n$	$Q_n$	Toggle

#### Operation :

- We will discuss the operation of the master slave JK FF with reference to its truth table.
- We must always remember one important thing that in the positive half cycle of the clock, the master is active and in the negative half cycle, the slave is active. This is shown in Fig. 4.12.2.



(c-607) Fig. 4.12.2

#### Case I : Clock = x, J = K = 0 (No change)

- For clock = 1, the master is active, slave inactive. As J = K = 0.  
 $\therefore$  Outputs of master i.e.  $Q_1$  and  $\bar{Q}_1$  will not change. Hence the S and R inputs to the slave will remain unchanged.
- As soon as clock = 0, the slave becomes active and master is inactive. But since the S and R inputs have not changed, the slave outputs will also remain unchanged.  
 $\therefore$  The outputs will not change if  $J = K = 0$ .

#### Case II : Clock = $\overline{JL}(1)$ , J = K = 0 (No change) (c-6395)

This condition has been already discussed in case I.

#### Case III : Clock = $\overline{JL}(1)$ , J = 0 and K = 1 (Reset) (c-6395(a))

- Clock = 1 : Master active, slave inactive.  
 $\therefore$  Outputs of the master become  $Q_1 = 0$  and  $\bar{Q}_1 = 1$ . That means S = 0 and R = 1.
- Clock = 0 : Slave active, master inactive.  
 $\therefore$  Outputs of the slave become Q = 0 and  $\bar{Q} = 1$ . This is the RESET operation.
- Again if clock = 1 : Master active, slave inactive.  
 $\therefore$  Even with the changed outputs Q = 0 and  $\bar{Q} = 1$  fed back to master, its outputs will be  $Q_1 = 0$  and  $\bar{Q}_1 = 1$ . That means S = 0 and R = 1.
- Hence with clock = 0 and slave becoming active, the outputs of slave will remain Q = 0 and  $\bar{Q} = 1$ .
- Thus we get a stable output from the Master slave.

**Case IV : Clock =  $\overline{J\bar{K}}$ , J = 1 and K = 0 (Set) (C-6395(b))**

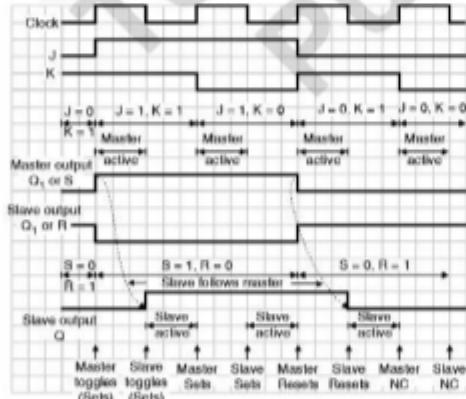
Clock = 1 : Master active, slave inactive.

- $\therefore$  Outputs of master become  $Q_1 = 1$  and  $\bar{Q}_1 = 0$  i.e. S = 1, R = 0.
- Clock = 0 : Master inactive, slave active.
- $\therefore$  Outputs of slave become Q = 1 and  $\bar{Q} = 0$ .
- Again if clock = 1 then it can be shown that the outputs of the slave are stabilized to Q = 1 and  $\bar{Q} = 0$ .

(C-6395(c))

**Case V : Clock =  $\overline{J\bar{K}}$ , J = 1 and K = 1 (Toggle without race)**

- Clock = 1 : Master active, slave inactive.
- $\therefore$  Outputs of master will toggle. So S and R also will be inverted.
- Clock = 0 : Master inactive, slave active.
- $\therefore$  Outputs of the slave will toggle.
- These changed output are returned back to the master inputs.
- But since clock = 0, the master is still inactive. So it does not respond to these changed outputs.
- This avoids the multiple toggling which leads to the race around condition. Thus the master slave flip flop will avoid the race around condition.
- The waveforms for the master slave flip flop are shown in Fig. 4.12.3.



(c-608) Fig. 4.12.3 : Waveforms of master slave JK flip flop

**Observations from the waveforms :**

- We can make the following important observations from the waveforms of the master slave JK FF.
- 1. The slave always follows the master, after a delay of half clock cycle period.
- 2. The multiple toggling or the race around condition is successfully avoided.

**4.13 Preset and Clear Inputs :**

SPPU : May 11

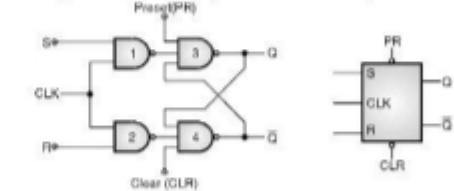
**University Questions**

Q. 1 What is meant by asynchronous inputs of flip-flop ?  
(May 11, 4 Marks)

- In the flip flops discussed so far when the power switch is turned on, the state of outputs Q and  $\bar{Q}$  can be anything that means either Q = 0,  $\bar{Q} = 1$  or Q = 1,  $\bar{Q} = 0$ . It is not predictable.
  - But this uncertainty cannot be tolerated in certain applications. In some applications it is necessary to initially set or reset the flip flops precisely.
  - This can be practically achieved by adding two more inputs to a flip flop, called **preset (PR)** and **clear (CLR)** inputs.
- These inputs are called as **direct** or **asynchronous inputs** because we can apply them any time between clock pulses without thinking about their synchronization with the clock.
- That means applying PR or CLR inputs is not related to the clock in any way.

**4.13.1 S-R Flip-Flop with Preset and Clear Inputs :****Logic diagram and symbol :**

- The S-R flip flop with preset and clear is shown in Fig. 4.13.1(a) and its symbol is shown in Fig. 4.13.1(b).



(a) S-R FF with preset and clear

(c-609) Fig. 4.13.1



- Note that both these inputs are active low inputs. This is indicated by the bubbles on these inputs in Fig. 4.13.1(b).

**Operation :**

**Case I : PR = CLR = 1**

If PR = CLR = 1 then both are inactive and the S-R FF operates as per the truth table of the conventional SR flip-flop.

**Case II : PR = 0 and CLR = 1 (Preset condition)**

- As PR = 0, the output of gate 3 of Fig. 4.14.1(a) will be 1.
- That means Q = 1.
- Therefore all the three inputs to the gate 4 will be 1 and  $\bar{Q}$  will become 0.
- Thus making PR = 0 will set the flip flop. Note that with PR = 0 and CLR = 1 the flip flop is set irrespective of the status of S, R or clock inputs. Therefore it is said that the PR input has higher priority than S, R or CLK inputs.

**Case III : PR = 1, CLR = 0 (Clear or Reset condition)**

- If PR = 1 and CLR = 0, then the output of gate-4 i.e.  $\bar{Q} = 1$ .
- Therefore all the three inputs to gate-3 will be 1 and hence Q = 0.
- Thus making CLR = 0 will reset the flip flop. Note that with CLR = 0 and PR = 1, the FF is reset irrespective of the status of S, R or clock inputs.
- Therefore it is said that CLR has higher priority than S, R or CLK inputs.

**Case IV : PR = 0, CLR = 0**

- This condition should never be used, because it leads to an uncertain state.

**Truth table :**

- The operation of SR flip flop with preset and clear inputs is summarized in Table 4.13.1.

(C-609(a)) **Table 4.13.1 : Summary of operation of SR FF**

CLK	PR	CLR	Inputs		Output	Operation performed
			1	0		
1	1	1	Q <sub>out</sub>		Normal S-R flipflop	
x	0	1	1		FF is set	
x	1	0	0		FF is reset	

- The don't care conditions (X) marked in the CLK column indicate that PR and CLR inputs have higher priority than clock input.

**4.13.2 Synchronous Preset and Clear**

**Inputs :**

- Preset and clear inputs can be of two types :
  - Asynchronous with clock.
  - Synchronous with clock.
- We have already discussed the asynchronous preset and clear inputs.

The operation of synchronous preset and clear inputs can be understood from the truth table given in Table 4.13.2.

(C-6075) **Table 4.13.2 : Truth table for S-R FF with synchronous preset and clear**

P <sub>r</sub>	C <sub>r</sub>	CLK	Inputs		Outputs	
			S	R	Q <sub>out</sub>	$\bar{Q}_{out}$
X	X	0	X	X	NC	NC
0	0	1	X	X	RACE	RACE
0	1	1	X	X	0	1
1	0	1	X	X	1	0
1	1	1	0	1	NC	NC
1	1	1	0	1	0	1
1	1	1	1	0	1	0
1	1	1	1	1	RACE	RACE

**4.13.3 JK Flip Flop with Preset and Clear**  
Inputs :

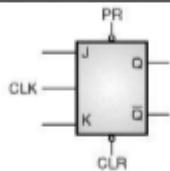
SPPU : May 11

**University Questions**

- Q. 1** Explain the operation of J-K flip-flop having preset and clear input. (May 11, 4 Marks)

**Symbol and Truth table :**

- A JK flip flop with preset and clear inputs is shown in Fig. 4.13.2. These are the synchronous preset and clear terminals.
- Both these inputs are active low and have higher priority than all the other inputs.



(C-610) (a) Symbol of a JK FF with preset and clear inputs

Inputs			Output	Operation performed
CLK	PR	CLR		
1	1	1	$Q_{n+1}$	Normal JK flipflop
x	0	1	1	FF is set
x	1	0	0	FF is reset

(C-610) (a)(b) Summary of operation of JK FF

Fig. 4.13.2

#### 4.13.4 Applications of JK Flip Flop :

- 1. Shift register 2. Counters

#### 4.14 Various Representations of Flip Flops :

- There are different ways in which a flip flop can be represented.
- Each representation is suitable for a different application.
- Various representations of flip flops are :
  1. Characteristic equations.
  2. Flip flop as Finite State Machine (FSM).
  3. Flip flop excitation tables.

##### 4.14.1 Characteristic Equations :

- We have already introduced the concept of characteristic equations earlier in this chapter and we have written the characteristic equations for various flip flops.

#### 4.15 Excitation Table of Flip-Flop :

- Till now we have written the truth tables of various flip flops. The truth tables are also known as the **characteristic tables**.
- But while designing the sequential circuits, some times the present and next state of a circuit are given and we are expected to find the corresponding input condition.

- We need to use the **excitation tables** of flipflops to do this. These tables are different from the characteristic tables.
- Present state is the state prior to application of clock pulse and the next state means the state after application of clock pulse.

##### 4.15.1 Excitation Table of SR Flip Flops :

- For example the outputs of an S-R FF before clock pulse are  $Q_n = 0$  and  $\bar{Q}_n = 1$  and it is expected that these outputs should remain unchanged after application of clock.
- Then what must be the values of inputs  $S_n$  and  $R_n$  to achieve this ?
- Refer to the truth table of SR FF to answer this question.
- The answer is, for the following two conditions the outputs remain unchanged at  $Q = 0$  and  $\bar{Q} = 1$ .

##### Condition 1 :

- $S_n$  and  $R_n = 0 \rightarrow$  Refer first row of Table 4.15.1.

##### Condition 2 :

- $S_n = 0 R_n = 1 \rightarrow$  Refer third row of Table 4.15.1.

(C-657) Table 4.15.1 : Truth table of SR flip flop

CLK	$S_n$	$R_n$	$Q_{n+1}$	$\bar{Q}_{n+1}$
↑	0	0	$Q_n$	$\bar{Q}_n$
	0	1	0	1
↑	1	0	1	0
	1	1	Race	Race

- From the two conditions mentioned above we conclude that  $S_n$  input should be equal to 0 and  $R_n$  input can be 0 or 1 (don't care) if we want to maintain  $Q = 0$  and  $\bar{Q} = 1$  before and after clock.
- Similarly we can find the input conditions (Values of S and R inputs) for all the possible situations that may exist on the output side.
- The table containing all these output situations and the corresponding input conditions is called as the "excitation table" of a flip flop.
- The excitation table of SR flip flop is shown in Table 4.15.2.



(C-6581) Table 4.15.2 : Excitation table of SR flip flop

	Present state of Q output	Next state of Q output	S <sub>n</sub> input	R <sub>n</sub> input
Case I	0	0	0	X
Case II	0	1	1	0
Case III	1	0	0	1
Case IV	1	1	X	0

**Description of excitation table of SR FF :**

We have already discussed case I.

**Case II : Q should change from 0 to 1**

- This is nothing but the set condition.
- Hence S<sub>n</sub> = 1 and R<sub>n</sub> = 0 should be the inputs.

**Case III : Q should change from 1 to 0**

- This is nothing but the reset condition.
- Hence S<sub>n</sub> should be 0 and R<sub>n</sub> should be 1.

**Case IV : Q should be 1. No change**

- To satisfy this requirement, we have two possible input conditions :

**Condition 1 :** S<sub>n</sub> = R<sub>n</sub> = 0 i.e. No change in output.

**Condition 2 :** S<sub>n</sub> = 1 and R<sub>n</sub> = 0.

- From these conditions we conclude that S<sub>n</sub> can be either 0 or 1 i.e. don't care and R<sub>n</sub> = 0. Hence the inputs corresponding to this case is,
- S<sub>n</sub> = X and R<sub>n</sub> = 0
- Similarly we can write the excitation tables for the other flip flops.

**4.15.2 Excitation Table of D Flip Flop :**

- Excitation table of a D flip flop is given by Table 4.15.3.

(C-7829) Table 4.15.3 : Excitation table of a D flip flop

	Q output	Input
Present state	Next state	D <sub>n</sub>
0	0	0
0	1	1
1	0	0
1	1	1

**4.15.3 Excitation Table of JK Flip Flop :**

- Excitation table of a JK flip flop is given by Table 4.15.4.

(C-6211) Table 4.15.4 : Excitation table of JK flip flop

	Q output		Inputs	
Present state	Next state	J <sub>n</sub>	K <sub>n</sub>	
Case I	0	0	0	X
Case II	0	1	1	X
Case III	1	0	X	1
Case IV	1	1	X	0

Don't care conditions result due to toggling

- Refer to Case II of Table 4.15.4. For the change in Q output from 0 to 1 we have the following two input conditions :
- Condition I :** J<sub>n</sub> = 1 and K<sub>n</sub> = 0 i.e. set condition,
- Condition II :** J<sub>n</sub> = 1 and K<sub>n</sub> = 1 i.e. toggle.
- Hence J<sub>n</sub> = 1 and K<sub>n</sub> = X (0 or 1) corresponding to case II.
  - Similarly for case III also, the don't care condition corresponds to toggle mode.

**4.15.4 Excitation Table of T Flip Flop :**

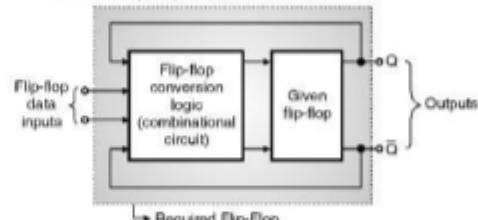
- Excitation table of T flip flop is shown in Table 4.16.5.

(C-7829) Table 4.15.5 : Excitation table of T flip-flop

	Q output		Input
Present state	Next state	T <sub>n</sub>	
0	0	0	
0	1	1	
1	0	1	
1	1	0	

**4.16 Conversion of Flip Flops :****Concept :**

- The conversion from one type of flip flop to the other (say SR FF to JKFF), needs a systematic approach using the excitation tables and K map simplifications.
- Fig. 4.16.1 shows a generalized model for conversion from one flip flop to the other.



(C-611) Fig. 4.16.1 : General model used to convert one type of FF to the other



- As shown in Fig. 4.16.1 the required flip flop is actually a combination of the given flip flop and a combinational logic circuit using gates. (Such a combinational circuit is called as the flip flop conversion logic).
- The inputs to FF conversion logic are, the flip flop data inputs and the outputs of given flip-flop i.e. the given FF and the desired FF.
- The conversion logic is designed by combining the excitation tables of both the flip flops.
- The truth table of the conversion logic has data inputs and  $Q$  and  $\bar{Q}$  outputs of the given FF as inputs whereas the inputs of the given FF are the outputs of the truth table.
- Then we draw the K map for each output and obtain the simplified expressions. The conversion logic is then implemented using gates.

#### 4.16.1 Conversion from S-R Flip Flop to D

**Flip Flop :**

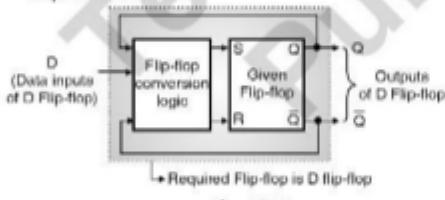
SPPU : Dec. 18,

**University Questions**

- Q. 1** With truth table, K-map and logic diagram explain how SR F/F is converted to D F/F.

(Dec. 18, 6 Marks)

- Refer Fig. 4.16.2. Here the given FF is SR FF and the required FF is D FF.



- The truth table for the conversion logic is shown in Table 4.16.1. The inputs are D and Q whereas outputs are S and R.
- The truth table is prepared by combining the excitation tables of D F/F and SR FF.

(C-6187) Table 4.16.1 : Truth table for SR to D FF conversion

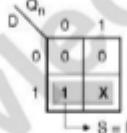
D	Inputs		Outputs	
	Present state $Q_n$	Next state $Q_{n+1}$	S	R
0	0	0	0	X
1	0	1	1	0
0	1	0	0	1
1	1	1	X	0

Entries from excitation table of D FF →

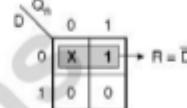
Entries from excitation table of SR FF →

- Now write the K maps for the S and R outputs as shown in Figs. 4.16.3(a) and (b).

For S output



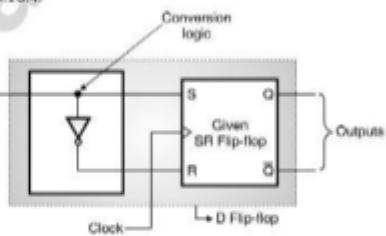
For R output



(C 613) Fig. 4.16.3

**Logic diagram :**

- The logic diagram for SR FF to D FF is shown in Fig. 4.16.4.



#### 4.16.2 Conversion of JK FF to T FF :

SPPU : May 06, May 08, Dec. 08, Dec. 09, May 10

**University Questions**

- Q. 1** Explain how J-K F/F is converted to T F/F ?

(May 06, May 08, Dec. 08, Dec. 09, May 10, 4 Marks)

**Step 1 : Write the truth table for conversion :**

- The required truth table is obtained from the excitation tables of JK and T flip flops as follows :



(C-6388) Table 4.16.2 : Truth table for conversion from JK to T FF

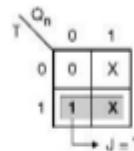
T	Inputs		Outputs	
	Present state of Q	Next state of Q	J	K
0	0	0	0	X
1	0	1	1	X
1	1	0	X	1
0	1	1	X	0

Excitation table of T FF  
Excitation table of JK FF

## Step 2 : K maps and simplification :

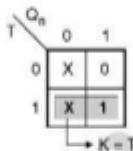
- The K maps for outputs J and K are shown in Fig. 4.16.5.

For J output



(a) K map for J output

For K output

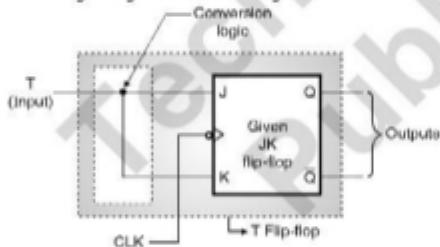


(b) K map for K output

(C-615) Fig. 4.16.5

## Step 3 : Draw the logic diagram :

- The logic diagram is shown in Fig. 4.16.6.



(C-616) Fig. 4.16.6 : Logic diagram for conversion of JK FF to T FF

## 4.16.3 SR Flip Flop to T Flip Flop :

SPPU : Dec. 07, Dec. 16

## University Questions

- Q. 1 Convert SR flip-flop to toggle flip-flop (SR to TFF).  
(Dec. 07, Dec. 16, 4 Marks)

- The stepwise conversion process is as follows :

## Step 1 : Write the truth table :

(C-809) Table 4.16.3 : Truth table for SR FF to T FF

T	Inputs		Outputs	
	Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	S	R
0	0	0	0	X
1	0	1	1	0
1	1	0	0	1
0	1	1	X	0

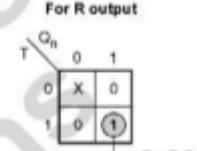
## Step 2 : Write the K maps and obtain the expressions for S and R :

For S output



(a) K map for S

For R output

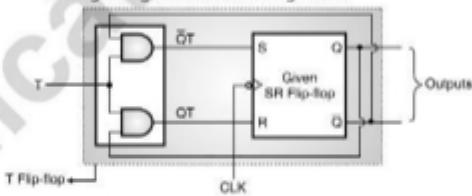


(b) K map for R

(C-617) Fig. 4.16.7

## Step 3 : Draw the logic diagram :

- The logic diagram is shown in Fig. 4.16.8.



(C-618) Fig. 4.16.8 : Conversion from SR flip flop to T flip flop

## 4.16.4 SR Flip Flop to JK Flip Flop :

SPPU : May 12, May 17, Dec. 19

## University Questions

- Q. 1 Perform the following conversion : SR FF to JK FF.  
(May 12, 4 Marks)
- Q. 2 Convert SR flip-flop to JK flip-flop.  
(May 17, Dec. 19, 6 Marks)

## Step 1 : Write the truth table for SR to JK :

- The truth table for SR to JK flip flop conversion is shown in Table 4.16.4.



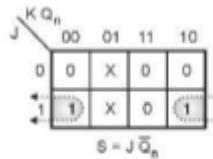
(C-6389) Table 4.16.4 : Truth table for SR to JK FF conversion

Inputs		Outputs	
J	K	Present state $Q_n$	Next state $Q_{n+1}$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1
1	0	1	X

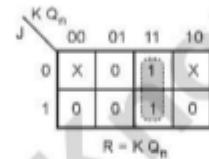
Excitation table of JK FF  
Excitation table of SR FF

#### Step 2 : K maps and simplification :

- K maps for S and R outputs are shown in Fig. 4.16.9.  
For output S      For output R



(a) K-map for S output

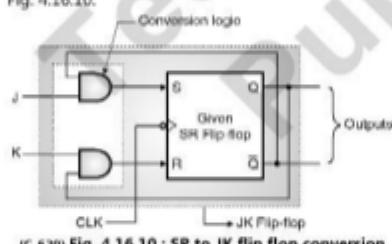


(b) K-map for R output

(C-619) Fig. 4.16.9

#### Step 3 : Logic diagram :

- The logic diagram of SR to JK flip flop is given in Fig. 4.16.10.



(C-620) Fig. 4.16.10 : SR to JK flip flop conversion

#### 4.16.5 Conversion of D Flip Flop to T Flip Flop :

SPPU : Dec. 11, May 14, May 15, Dec. 15

##### University Questions

Q. 1 Convert D to T flip-flop.

(Dec. 11, May 14, Dec. 15, 4 Marks)

Q. 2 Convert D to T and vice versa. (May 15, 6 Marks)

#### Step 1 : Write the truth table for D to T FF conversion

- The truth table is as follows :

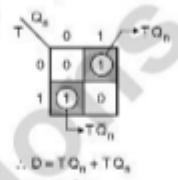
(C-6044) Table 4.16.5 : Truth table for D to T FF conversion

Inputs		Outputs	
T	Present state $Q_n$	Next state $Q_{n+1}$	D
0	0	0	0
1	0	1	1
1	1	0	0
0	1	1	1

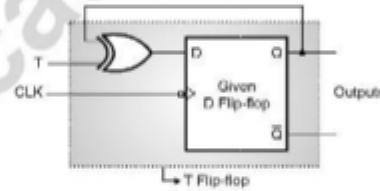
#### Step 2 : K map and simplification and logic diagram :

- K map is shown in Fig. 4.16.11(a) and logic diagram is shown in Fig. 4.16.11(b).

For output D



(a) K map and simplification



(b) Logic diagram

(C-621) Fig. 4.16.11 : D flip flop to T flip flop

#### 4.16.6 T Flip Flop to D Flip Flop Conversion :

SPPU : Dec. 11, May 15, Dec. 15

##### University Questions

Q. 1 Convert T to D flip-flop.

(Dec. 11, Dec. 15, 4 Marks, May 15, 6 Marks)

#### Step 1 : Write the truth table :

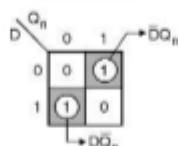
- The truth table is as given in Table 4.16.6.

(C-7831) Table 4.16.6 : Truth table for T FF to D FF conversion

Inputs		Outputs	
T	Previous state $Q_n$	Next state $Q_{n+1}$	D
0	0	0	0
1	0	1	1
0	1	0	1
1	1	1	0

**Step 2 : K maps simplification and logic diagram :**

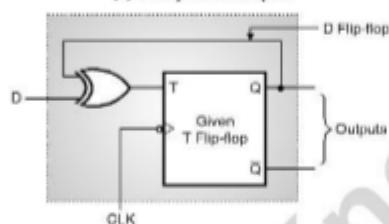
- The K map is shown in Fig. 4.16.12(a) and the logic diagram is given in Fig. 4.16.12(b).

**For output D**

$$\therefore T = DQ_n + \bar{D}Q_{\bar{n}}$$

$$\therefore T = D \oplus Q_n$$

(a) K map for D output

(b) Logic diagram of T to D flip flop conversion  
(C-622) Fig. 4.16.12**4.16.7 JK Flip Flop to D Flip Flop Conversion :**

SPPU : Dec. 09, May 10, May 12

**University Questions**

- Q. 1** Convert JK flip flop into D flip-flop.

(Dec. 09, 2 Marks, May 10, May 12, 4 Marks)

**Step 1 : Write the truth table for JK to D conversion :**

- The truth table is as follows :

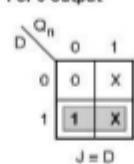
(C-6390) Table 4.16.7 : Truth table for JK to D

flip flop conversion

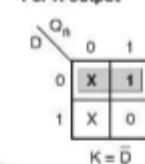
Inputs		Outputs	
D	Previous state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	J      K
0	0	0	0      X
1	0	1	1      X
0	1	0	X      1
1	1	1	X      0

← Excitation table of D FF →

← Excitation table of JK FF →

**Step 2 : K maps and simplification :****For J output**

(a) K map for J output

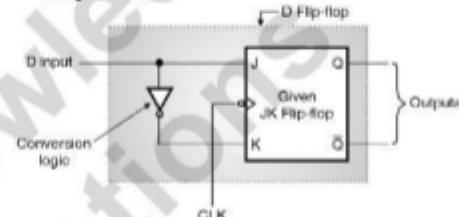
**For K output**

(b) K map for K output

(C-623) Fig. 4.16.13

**Step 3 : Draw the logic diagram :**

- The logic diagram for JK flip flop to D flip flop is shown in Fig. 4.16.14.



(C-624) Fig. 4.16.14 : Logic diagram for conversion from JK FF to D FF

**4.16.8 JK Flip Flop to SR Flip Flop**

Conversion : SPPU : Dec. 10, May 11, Dec. 12

**University Questions**

- Q. 1** Convert : JK FF to SR FF.

(Dec. 10, May 11, 6 Marks, Dec. 12, 4 Marks)

**Step 1 : Write the truth table for JK to SR :**

- The truth table is shown in Table 4.16.8.

(C-8045) Table 4.16.8 : Truth table for JK to SR conversion

S	R	Inputs		Outputs	
		Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	J	K
0	0	0	0	0	X
0	1	0	0	0	X
1	0	0	1	1	X
1	0	0	1	1	X
0	1	1	0	X	1
0	1	1	0	X	1
0	0	1	1	X	0
1	0	1	1	X	0

**Step 2 : K-maps and simplifications :**

For J output

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

(a)

For K output

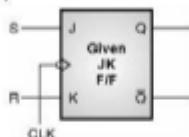
		00	01	11	10
		0	X	0	X
1		X	0	X	X
S	R	Q <sub>n</sub>			

(b)

(C-625) Fig. 4.16.15

**Step 3 : Logic diagram :**

- The logic diagram for JK to SR FF is shown in Fig. 4.16.15(c).



(C-626) Fig. 4.16.15(c) : JK to SR FF conversion

**4.16.9 D FF to SR FF Conversion :****Step 1 : Write the truth table for D to S-R conversion :**

(C-7832) Table 4.16.9 : Truth table for D to S-R conversion

Inputs		Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	Output D
S	R			
0	0	0	0	0
0	1	0	0	0
1	0	0	1	1
0	1	1	0	0
0	0	1	1	1
1	0	1	1	1

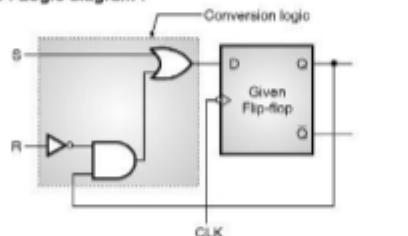
**Step 2 : K maps and simplification :**

For D

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

$$\therefore D = S + \bar{R} Q_n$$

(C-627) Fig. 4.16.16 : K-map and simplification

**Step 3 : Logic diagram :**

(C-628) Fig. 4.16.17 : Logic diagram for D to SR FF conversion

**4.16.10 T FF to SR FF Conversion :**

SPPU : Dec. 12

**University Questions**

Q. 1 Convert : T FF to SR FF. (Dec. 12, 4 Marks)

**Step 1 : Write the truth table for T to S-R conversion :**

(C-8018) Table 4.16.10 : Truth table for T to S-R conversion

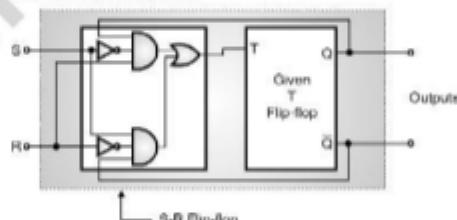
Inputs		Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	Output T
S	R			
0	X	0	0	0
1	0	0	1	1
0	1	1	0	1
X	0	1	1	0

**Step 2 : K maps and simplification for T output :**

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

$$\therefore T = \bar{S} \bar{R} \bar{Q}_n + \bar{S} R Q_n$$

(C-629) Fig. 4.16.18 : K-map and simplification

**Step 3 : Logic diagram :**

(C-630) Fig. 4.16.19 : Logic diagram for T to SR FF conversion

**4.16.11 Conversion from D FF to JK FF :**

SPPU : Dec. 10, May 19

**University Questions**

Q. 1 Convert : D FF to JK FF.

(Dec. 10, 4 Marks, May 19, 6 Marks)

**Step 1 : Write the truth table :**

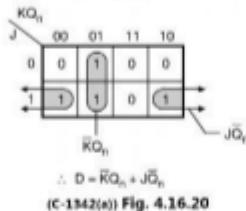
- Table 4.16.11 shows the truth table obtained from the excitation tables of D and JK FFs.



(C-8047) Table 4.16.11 : Truth table for D to JK conversion

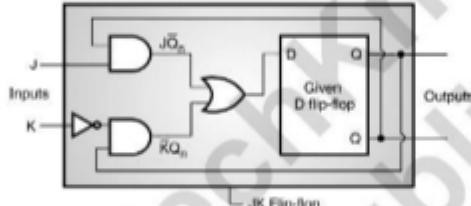
Inputs			Output D
J	K	Present state $Q_n$	Next state $Q_{n+1}$
0	X	0	0
1	X	0	1
X	1	1	0
X	0	1	1

**Step 2 : K maps and simplification :**  
For output D



**Step 3 : Logic diagram :**

- The logic diagram for D FF to JK FF is shown in Fig. 4.16.21.



## 4.17 Flip Flop Timing Considerations :

- The manufacturers of flip-flop ICs specify many important timing parameters and characteristics of flip-flops. The user/designer has to consider these, before using the flip flop for any application.
- Some of the important timing characteristics of flip-flops are :
  - Set up and hold times
  - Propagation delay
  - Maximum clocking frequency  $f_{max}$
  - Clock pulse HIGH and LOW times
  - Asynchronous active pulse width
  - Clock transition times

### 4.17.1 Set Up and Hold Times :

SPPU : Dec. 09, May 11

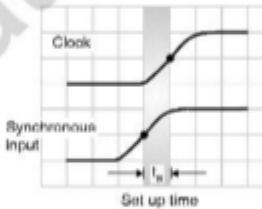
#### University Questions.

- Q. 1** Define the following terms as applied to flip-flops :  
 1. Set up time      2. Hold time  
 (Dec. 09, 2 Marks, May 11, 4 Marks)

- In order to ensure the proper operation of the clocked FF, its set up and hold time requirements must be satisfied.
- The S, R, J, K, D or T inputs of the corresponding FFs are called as the synchronous control inputs of FF while preset and clear are the asynchronous control inputs.

#### Set up time ( $t_{su}$ ) :

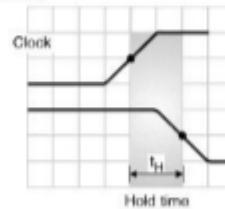
- The set up time is defined as the minimum time interval immediately before the active transition of clock signal during which we have to maintain the synchronous control input at a proper voltage level.
- This is shown in Fig. 4.17.1(a).
- The value of  $t_{su(\min)}$  is generally specified by the IC manufacturers.



(C-631) Fig. 4.17.1(a) : Set up time

#### Hold time ( $t_h$ ) :

- It is defined as the time interval immediately after the active transition of clock input during which we have to maintain the voltage synchronous input at a proper level. This is shown in Fig. 4.17.1(b).
- The value of  $t_{h(\min)}$  is generally specified by the IC manufacturers.



(C-632) Fig. 4.17.1(b) : Hold time

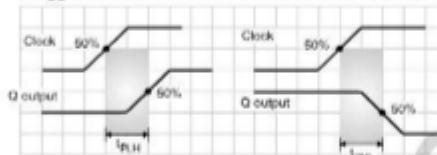


#### 4.17.2 Propagation Delays : SPPU : May 11

##### **University Questions.**

**Q. 1** Define propagation delay time as applied to flip-flop. (May 11, 2 Marks)

- When a clock signal is applied to FF input, ideally the outputs are expected to change instantaneously.
- But actually it takes some finite time for the outputs to change after application of the clock input. This is called as propagation delay.
- Fig. 4.17.2 demonstrates the two propagation delays. The flip flop is assumed to be the positive edge triggered.



(a) Delay going from  
LOW to HIGH      (b) Delay going from  
HIGH to LOW  
(C-638) Fig. 4.17.2 : Propagation delays

- $t_{PLH}$  is the propagation delay for Q output going from low to high whereas  $t_{PHL}$  is the propagation delay for Q output going from high to low.
- Note that these delays are measured between the 50% points on the input and output waveforms as shown in Fig. 4.17.2.
- Also note that the propagation delays can be defined with respect to the preset or clear inputs as well.
- The manufacturers generally define the maximum values of  $t_{PLH}$  and  $t_{PHL}$ . Their typical values are in the range from few nanoseconds to 100 nS.

#### 4.17.3 Maximum Clocking Frequency $f_{max}$ :

SPPU : May 11

##### **University Questions**

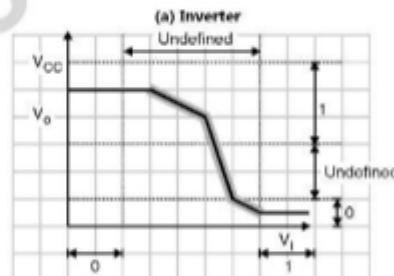
**Q. 1** Define maximum clock frequency as applied to flip-flop. (May 11, 2 Marks)

- It is defined as the maximum frequency of the clock input to a FF with reliable triggering.
- The value of  $f_{max}$  will vary from FF to FF. The typical range of  $f_{max}$  is 20 to 35 MHz.

#### 4.18 Flip Flop Metastability :

- Under the normal operating conditions, a flip flop has only two states (0 or 1). If the setup time and hold time parameters are of proper value, then the Q output of the FF can change from 0 to 1 or 1 to 0 without creating any problems.
- But if the values of set up time and hold time are not proper, then the output may enter into a third state other than 0 and 1. This state is called as the **metastable state**.
- The metastable state is an undefined state. It exists halfway between the logical 0 state and logical 1 state. The metastable state causes problems in the operation of a digital system.
- Many times the feedback existing in the flipflop circuits is responsible for its metastable state.
- Every digital circuit has a voltage gain ( $V_o/V_i$ ). Fig. 4.18.1(a) shows an inverter and Fig. 4.18.1(b) shows its transfer characteristics.

$$V_i \rightarrow \text{Inverter} \rightarrow V_o = f(V_i)$$



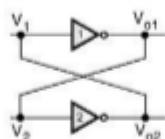
(a) Inverter  
(b) Transfer characteristics  
(C-1724) Fig. 4.18.1

- As shown in Fig. 4.18.1(b), as the input switches from 0 to 1, the output will switch from 1 to 0 and there is a time when the output voltage is undefined.
- Now consider the two cross coupled inverters of Fig. 4.18.1(c).
- The output of one becomes the input to the other. Let  $V_1$  and  $V_2$  be the inputs of the two inverters and let  $V_{o1}$  and  $V_{o2}$  be their outputs.

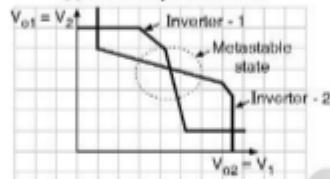


- Then  $V_{o1} = F(V_1)$  and  $V_{o2} = F(V_2)$
- We may assume that the voltage gain function  $F$  is same for both the inverters.
- Refer Fig. 4.18.1(c) to write

$$V_{o1} = V_2 \text{ and } V_{o2} = V_1$$



(c) Cross coupled inverters



(d) Voltage transfer curves for the cross coupled inverters

(C-1725) Fig. 4.18.1

- Now refer Fig. 4.18.1(d) which shows the transfer curve for the cross coupled inverters.
- One axis has been rotated by 90° to plot these characteristics.
- The metastable state exists where the two transfer curves intersect.

#### 4.18.1 Characteristics, Causes and Effects of Metastability :

1. The metastable state is present halfway between logical zero (0) and 1 states.
2. If the setup and hold time requirements or the clock pulse timing requirements are not met, then the FF may go into the metastable state.
3. The external signals connected asynchronously with the FFs can cause the metastability.
4. It is necessary to avoid metastability.

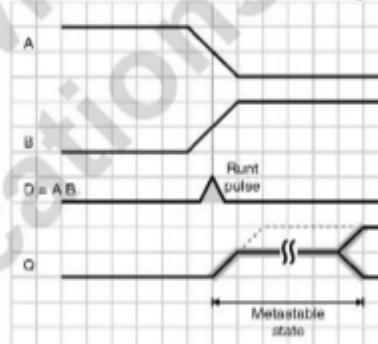
#### 4.18.2 Illustration of Metastability :

- For the illustration of metastability, consider the circuit shown in Fig. 4.18.2(a).

- The gate inputs A and B are asynchronous i.e. they can appear at any time.
- The FF is triggered. The inputs to FF are as shown in Fig. 4.18.2(b).
- Observe the appearance of a small "runt" pulse at the D input of the FF. The pulse may try to set FF (transition from 0 to 1).
- However the pulse may not have sufficient energy to carry out the transition (see Fig. 4.18.2(b)). This will leave the FF in the **metastable state**.



(a) Circuit for the illustration of metastability

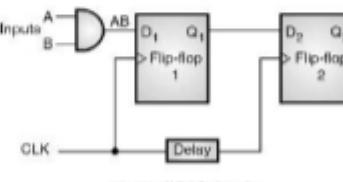


(b) Waveforms at various points

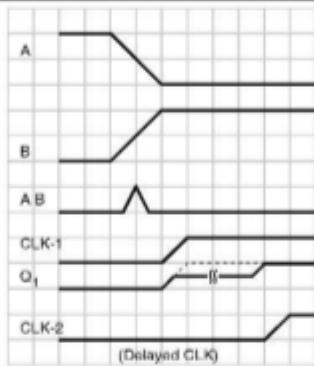
(C-1726) Fig. 4.18.2

#### Circuit modification to avoid metastability :

- The earlier circuit is modified as shown in Fig. 4.18.2(c) to reduce the probability of metastability.
- The clock pulse is delayed and applied to FF-2. This will provide some time to FF-1 to leave its metastable state, before the clock pulse gets applied to FF-2.
- The waveforms of Fig. 4.18.2(d) shows that the metastability is eliminated due to delayed clocking.



(c) Modified circuit



(d) Waveforms

(C-1727) Fig. 4.18.2

#### 4.19 Applications of Flip Flops :

- Some of the important applications of the flip flops are :
- 1. Elimination of keyboard debounce
- 2. As a memory element
- 3. In various types of registers
- 4. In counters/timers.
- 5. As a delay element.

#### 4.20 Analysis of Clocked Sequential Circuits :

- The behaviour of a clocked sequential circuit is dependent on the inputs, the outputs and the state of its flip-flops.
- The outputs as well as the next state both are function of inputs and present state.
- The analysis of the given clocked sequential circuit includes writing the state table and drawing the state diagram for the given circuit.
- In this section, we will introduce the concepts such as state diagram, state table, state equation and input equations and the step by step analysis procedure.

##### 4.20.1 State Table :

SPPU : Dec. 10, May 11, Dec. 12, Dec. 14, Dec. 17

###### University Questions

Q. 1 Explain state table.

(Dec. 10, May 11, Dec. 12, 4 Marks)

Q. 2 Explain :

1. State Table 2. State Diagram
3. State Reduction

(Dec. 14, 6 Marks)

Q. 3 Explain state diagram and state table with suitable example.

(Dec. 17, 6 Marks)

- We use the truth table for explaining the relation between its inputs and outputs of a combinational circuit.
- But it is not suitable to use the truth table to describe the input output relation of a sequential circuit.
- Instead we use the state table to describe the input output relation of the sequential circuits.
- The basic building block of a sequential circuit is a flip flop. The outputs  $Q_A, Q_B, \dots$  etc. of such flip flops will be used as **state inputs** to a sequential circuit. They are also called as **state variable**.
- In addition to this "x" represents an external input and Y represents the output of the sequential circuit.
- Y will be dependent on the state variables ( $Q_A, Q_B, \dots$ ) and the external input x.

The general format of a state table is shown in Table 4.20.1.

(C-7853) Table 4.20.1 : General format of state table

Present state	Next state		Output Y	
	x = 0	x = 1	x = 0	x = 1
$Q_A$	$Q_B$	$Q_A$	$Q_B$	$Q_A$
0	0	0 0	0 1	0 0
0	1	1 1	0 1	0 0
1	0	1 0	0 0	0 1
1	1	1 0	1 1	1 0

###### Definition :

- The state table is defined as the table that tells us about the relation between the present state, next state, external inputs and output of a sequential circuit.

##### 4.20.2 State Diagram :

SPPU : May 11, Dec. 12, Dec. 14, Dec. 17

###### University Questions

Q. 1 Explain : State diagram.

(May 11, 2 Marks, Dec. 12, 3 Marks)

**Q. 2 Explain :**

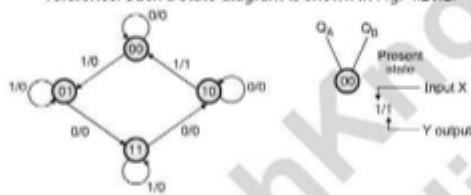
1. State Table
  2. State Diagram
  3. State Reduction
- (Dec. 14, 6 Marks)

**Q. 3 Explain state diagram and state table with suitable example.**

(Dec. 17, 6 Marks)

**Definition :**

- A state diagram is the graphical representation of a sequential circuit, which consists of states, state transitions and actions.
- State diagrams depict the permitted states and transitions as well as the events that effect these transition.
- The information available in the state table is represented graphically using the state diagram.
- The state diagram is drawn by using the state table as a reference. Such a state diagram is shown in Fig. 4.20.1.

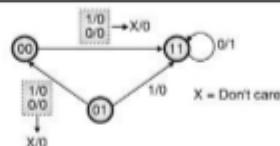


(C-653) Fig. 4.20.1 : State diagram

- The circle represents the present state. The arrows between the circles define the state transition say from 00 to 01 or 01 to 11.
- If there is directed line connecting the same circle then it means that the next state is same as the present state.
- The lines joining the circles are labeled with a pair of binary numbers with a " / " in between. For example the line joining 00 and 01 is labeled as 1/0.
- Note that 00 to 01 transition takes place when  $x = 1$  and  $y = 0$  (see row-1 of the state table). Hence 1 in 1/0 corresponds to  $x$  and 0 corresponds to  $y$ .

**Don't care condition in the state diagram :**

- Sometimes the same next state is reached for more than one present states.
- This is called as don't care condition in the state diagram, as shown in Fig. 4.20.2.



(C-654) Fig. 4.20.2 : Don't care condition in state diagram

**4.20.3 State Equation :****Definition :**

- State equation is an algebraic equation. The left side of this equation represents the next state of the flip flops.
- And the right hand side of this equation specifies the present state conditions which make the next state equal to 1.
- For example consider the following state expression.

(C-6235)

$$Q_{A(n+1)} = (\bar{Q}_A Q_B + Q_A \bar{Q}_B + Q_A Q_B) X + Q_A \bar{Q}_B X \dots$$

Next state

Combinations of present state and inputs which make the next state equal to 1.

- State equation is also called as **application equation**.
- We can derive the state equation from the state table using the K maps. The state equation is the Boolean function with time included into it.
- If the right hand side part is zero and we apply a clock pulse, then the next state will be equal to zero.

**Ex. 4.20.1 :** For the clocked D FF write the state table, draw the state diagram and write the state equation.

**Soln. :****Step 1 : Write the truth table :**

- Table P. 4.20.1(a) represents the truth table for a clocked D flip flop.

(C-6212) Table P. 4.20.1(a) : Truth table of a clocked D FF

Inputs		Outputs	
CLK	D	$Q_{n+1}$	$\bar{Q}_{n+1}$
0	X	$Q_n$	$\bar{Q}_n$
1	X	$Q_n$	$\bar{Q}_n$
↓	X	$Q_n$	$\bar{Q}_n$
↑	0	0	1
↑	1	1	0

No change in output

Q output follows

**Step 2 : Write the state table :**

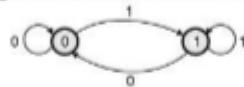
- Table P. 4.20.1(b) represents the state table for the clocked D FF.

(C-7834) Table P. 4.20.1(b) : State table of a clocked D FF

Present state $Q_n$	Next state $Q_{n+1}$	
	D = 0	D = 1
0	0	1
1	0	1

**Step 3 : State diagram :**

- State diagram of clocked D FF is shown in Fig. P. 4.20.1.



(C-659(a)) Fig. P. 4.20.1 : State diagram of a clocked D FF

**Step 4 : Write excitation table :**

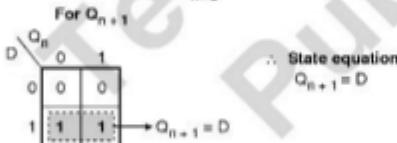
- The excitation table for D FF is given in Table P. 4.20.1(c).

(C-7835) Table P. 4.20.1(c) : Excitation table for D FF

Present state $Q_n$	Next state $Q_{n+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

**Step 5 : Write the state equation :**

- The K-map for output  $Q_{n+1}$  is shown in Fig. P. 4.20.1(a).



(C-656) Fig. P. 4.20.1(a) : K-map and state equation

**Ex. 4.20.2 :** For the clocked JK FF write the state table, draw the state diagram and write the state equation.

Soln. :

**Step 1 : Write the truth table :**

- Table P. 4.20.2(a) represents the truth table for a clocked JK flip flop.

(C-6213) Table P. 4.20.2(a) : Truth table of JK FF

Inputs			Output
CLK	J	K	$Q_{n+1}$
0	X	X	$Q_n$ (NC)
1	X	X	$Q_n$ (NC)
↓	X	X	$Q_n$ (NC)
↑	0	0	$Q_n$ (NC)
↑	0	1	0
↑	1	0	1
↑	1	1	$\bar{Q}_n$

No change in output

**Step 2 : Write the state table :**

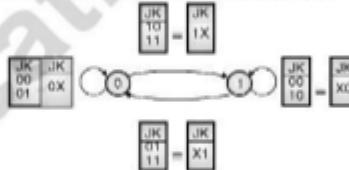
- Table P. 4.20.2(b) represents the state table for a clocked JK FF.

(C-6970) Table P. 4.20.2(b) : State table of clocked JK FF

Present state $Q_n$	Next state $Q_{n+1}$			
	JK = 00	JK = 01	JK = 10	JK = 11
0	0	0	1	1
1	1	0	1	0

**Step 3 : Draw the state diagram :**

State diagram is as shown in Fig. P. 4.20.2(a).



(C-657) Fig. P. 4.20.2(a) : State diagram of a JK flip flop

**Step 4 : Write the excitation table :**

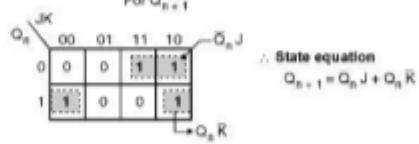
The excitation table of JK FF is as shown in Table P. 4.20.2(c).

(C-7840) Table P. 4.20.2(c) : Excitation table for JK FF

Present state $Q_n$	Next state $Q_{n+1}$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

**Step 5 : Write the state equation :**

- The K-map for output  $Q_{n+1}$  is shown in Fig. P. 4.20.2(b).



(C-658) Fig. P. 4.20.2(b) : K-map state equation



**Ex. 4.20.3 :** For a toggle FF write the state table, draw the state diagram and write the state equation.

**Soln. :**

**Step 1 : Write the truth table :**

- Table P. 4.20.3(a) gives the truth table for a T FF.

(C-8071) **Table P. 4.20.3(a) : Truth table for a T FF**

CLK	T	$Q_{n+1}$
0	X	$Q_n$
1	X	$\bar{Q}_n$
$\uparrow$	X	$Q_n$
$\downarrow$	0	$Q_n$
$\downarrow$	1	$\bar{Q}_n$

**Step 2 : Write the state table :**

- The state table is shown in Table P. 4.20.3(b).

(C-8072) **Table P. 4.20.3(b) : State table for a T FF**

Present state $Q_n$	Next state $Q_{n+1}$	
	T = 0	T = 1
0	0	1
1	1	0

**Step 3 : Draw the state diagram :**

- The state diagram is shown in Fig. P. 4.20.3(a).



(C-659) **Fig. P. 4.20.3(a) : State diagram for a T FF**

**Step 4 : Write the excitation table :**

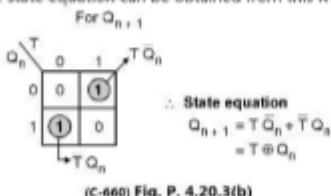
- Table P. 4.20.3(c) represents the excitation table of T FF.

(C-7842) **Table P. 4.20.3(c) : Excitation table for T FF**

Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

**Step 5 : Write the state equation :**

- The K-map for  $Q_{n+1}$  output is shown in Fig. P. 4.20.3(b). The state equation can be obtained from this K-map.



(C-660) **Fig. P. 4.20.3(b)**

## 4.21 Design of Clocked Synchronous State Machine using State Diagram :

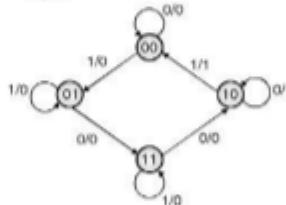
- Design of a clocked sequential circuits will start from the set of given specifications and it will end with drawing a logic diagram.
- The stepwise design procedure is as follows :

**Design steps :**

- Step 1 :** A state diagram or timing diagram or some other information is given, which describes the behaviour of the circuit that is to be designed.
- Step 2 :** Draw the state table.
- Step 3 :** The number of states can be reduced by state reduction methods.
- Step 4 :** Assign binary values to each state for the states in steps 2 and 3 using state assignment technique.
- Step 5 :** Determine the number of flip-flops required and assign a letter symbol to each one.
- Step 6 :** Decide the type of FF to be used.
- Step 7 :** Derive the circuit excitation table and output table from the state table.
- Step 8 :** Obtain the expressions for the circuit output and flip-flop inputs.
- Step 9 :** Draw the logic diagram.

- The design procedure of clocked sequential circuits will be clear by solving the following example.

**Ex. 4.21.1 :** For the state diagram given in Fig. P. 4.21.1(a) draw the clocked sequential circuit using T flip-flops.



(C-658) **Fig. P. 4.21.1(a) : Given state diagram**

**Soln. :**

**Step 1 : Write the state table :**

- The state table is written from the given state diagram as shown in Table P. 4.21.1(a).



(C-7837) Table P. 4.21.1(a) : State table

Present state	A	B	X = 0		X = 1		Output	
			A <sub>n+1</sub>	B <sub>n+1</sub>	A <sub>n+1</sub>	B <sub>n+1</sub>	X = 0	X = 1
0	0	0	0	0	0	1	0	0
0	1	0	1	1	0	1	0	0
1	0	0	1	0	0	0	1	1
1	1	0	1	0	1	1	0	0

**Step 2 : Decide the number of flip-flops :**

- As seen from the state table, there are no equivalent states. So there won't be any reduction in the state diagram.
- The circuit goes through four states. Hence we need to use 2 flip-flops.

**Step 3 : Write the circuit excitation table :**

- The circuit excitation table is shown in Table P. 4.21.1(b).
- The type of FF used is T type FF.

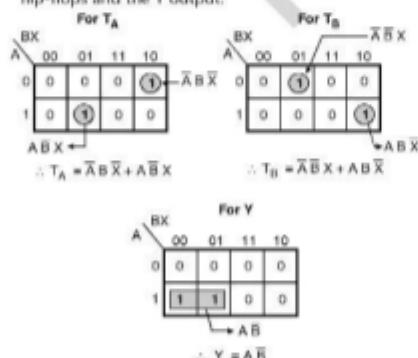
(C-7760) Table 4.21.1(b) : Circuit excitation table

Present state	A	B	Input X	Next state		FF Inputs	Output Y
				A <sub>n+1</sub>	B <sub>n+1</sub>		
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

- The shaded portion of the circuit excitation table corresponds to the excitation table of a T FF.

**Step 4 : K maps and simplifications :**

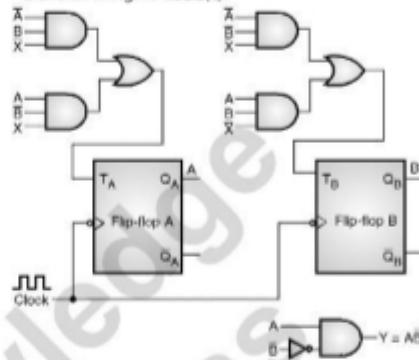
- Fig. P. 4.21.1(b) shows the K-maps and corresponding simplified expressions for T<sub>A</sub> T<sub>B</sub> i.e. inputs of the two T flip-flops and the Y output.



(C-686) Fig. P. 4.21.1(b) : K maps and simplifications

**Step 5 : Draw the logic diagram :**

- The logic diagram for required clocked sequential circuit is shown in Fig. P. 4.21.1(c).



(C-687) Fig. P. 4.21.1(c)

**Note :** We can use the design procedure in order to design the synchronous counters.

## 4.22 Clock Jitter and it's Effects on Synchronous Design :

SPPU : May 14, May 19

**University Questions**

Q. 1 What is clock skew and clock jittering in synchronous circuits ? (May 14, 2 Marks)

Q. 2 What is clock skew and clock jitter ? (May 19, 2 Marks)

- The clock jitter is defined as the shift of the clock signal with respect to the reference edge as shown in Fig. 4.22.1.
- Clock jitter represents the shift of active edge of a clock signal in the time domain.
- Clock jitter can be of two types :
  1. Long term jitter.
  2. Cycle-to-cycle jitter or edge to edge jitter.



(C-1343) Fig. 4.22.1 : Jitter



- As shown in Fig. 4.22.1, the active edges of the received clock keep changing their positions with respect to time at a very fast rate. This is jitter.
- The cycle to cycle jitter is defined as the change in clock signal between two consecutive edges.
- This type of jitter is more important when we design a high speed logic design, because this phenomenon affects the time available to the logic circuit.
- The cycle to cycle jitter (also known as short term jitter) depends on the quality and type of the clock generator.
- The long term jitter is observed due to clock edge variations over a large number of clock cycles.
- The long term jitter is due to the accumulated effects and it affects the synchronization and communication between various parts of a digital system.

#### 4.23 Typical Data Sheet Specifications of Flip flop :

##### 4.23.1 SN74LS74A : Dual D-Type Positive Edge-triggered Flip-Flop Low Power Schottky :

###### Description :

- The SN74LS/74LS74A is a dual edge-triggered flip-flop which utilizes Schottky TTL circuitry to produce high speed D-type flip-flops.
- Each flip-flop has individual clear and set inputs, as well as complementary Q and  $\bar{Q}$  outputs.

Mode select – truth table

Operating Mode	Inputs			Outputs	
	$\bar{S}_0$	$\bar{C}_0$	D	Q	$\bar{Q}$
Set	L	H	X	H	L
Reset (Clear)	H	L	X	L	H
*Undetermined	L	L	X	H	H
Load "1" (Set)	H	H	h	H	L
Load "0" (Reset)	H	H	l	L	H

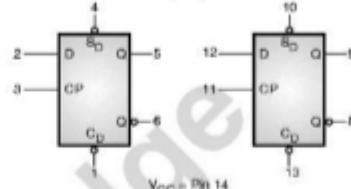
H, h = HIGH Voltage Level

L, l = LOW Voltage Level

X = Don't care

- i, h (q) = Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the LOW to HIGH clock transition.

Logic Symbol



$V_{CC} = V_0 = 14$   
 $V_{DD} = V_{IN} = 7$

J-Suffix - Case 632 - 07 (Ceramic)

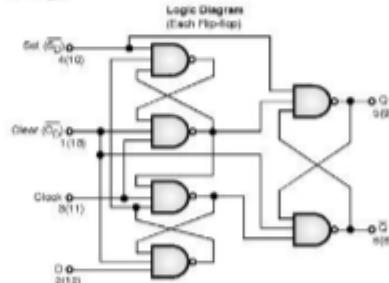
N-Suffix - Case 646 - 05 (Plastic)

(C-1722) Fig. 4.23.1

- Information applied at input D is transferred to the Q output on the positive-going edge of the clock pulse.
- Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive-going pulse.
- When the clock input is at either the HIGH or the LOW level, the D input signal does not have any effect on the Q and  $\bar{Q}$  outputs.

Now consider the mode select truth table indicates that both outputs will be HIGH while both  $\bar{S}_0$  AND  $\bar{C}_0$  are LOW, but the output states are unpredictable if  $\bar{S}_0$  and  $\bar{C}_0$  go HIGH simultaneously.

- If the levels at the set and clear are near  $V_{IL}$  maximum then we cannot guarantee to meet the minimum level for  $V_{OH}$ .



(C-3624) Fig. 4.23.2

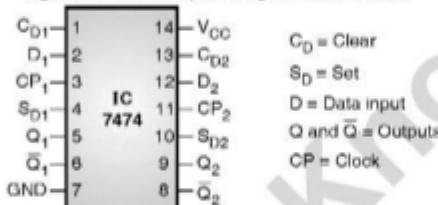


## Guaranteed operating ranges :

Symbol	Parameter		Min	Typ	Max	Unit
$V_{CC}$	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
$T_A$	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
$I_{OH}$	Output Current – High	54			-0.4	mA
$I_{OL}$	Output Current – Low	54			4.0	mA
		74			8.0	

## Pin configuration :

- Fig. 4.23.3 shows the pin configuration of IC 7474.



(C-3625) Fig. 4.23.3 : Pin configuration of IC 7474

## 4.23.2 SN74LS76A : Dual JK Flip-Flop with Set and Clear Low Power Schottky :

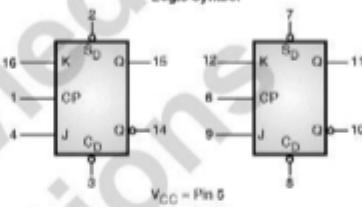
## Description :

- The SN54LS/74LS76A offers individual J, K, Clock Pulse, Direct Set and Direct Clear inputs.
- These dual flip-flops are designed so that when the clock goes HIGH, the inputs are enabled and data will be accepted.
- The Logic Level of the J and K inputs will perform according to the truth table as long as minimum set-up times are observed.
- Input data is transferred to the outputs on the HIGH-to-LOW clock transitions.

## Mode Select – Truth table

Operating Mode	Inputs		Outputs			
	$\bar{S}_0$	$\bar{C}_0$	J	K	Q	$\bar{Q}$
Set	L	H	X	X	H	L
Reset (Clear)	H	L	X	X	L	H
"Undetermined"	L	L	X	X	H	H
Toggle	H	H	H	h	$\bar{q}$	q
Load "0" (Reset)	H	H	L	h	L	H
Load "1" (Set)	H	H	H	l	H	L
Hold	H	H	L	l	q	$\bar{q}$

## Logic Symbol



(C-1723) Fig. 4.23.4

- Both outputs will be HIGH while both  $\bar{S}_0$  and  $\bar{C}_0$  are LOW, but the 7 output states are unpredictable if  $\bar{S}_0$  and  $\bar{C}_0$  go HIGH simultaneously.

H, h = HIGH Voltage Level  
L, l = LOW Voltage Level  
X = immaterial

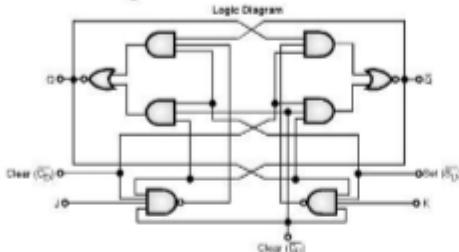
- $l, h (q)$  = Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the HIGH-to-LOW clock transition.

## Guaranteed operating ranges :

Symbol	Parameter		Min	Typ	Max	Unit
$V_{CC}$	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
$T_A$	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
$I_{OH}$	Output Current – High	54			-0.4	mA
$I_{OL}$	Output Current – Low	54			4.0	mA
		74			8.0	



- Logic diagram and pin configuration of IC 7476 is as shown in Fig. 4.23.5.



(a) Logic diagram

CP <sub>1</sub> =1	16—K <sub>1</sub>
S <sub>D1</sub> =2	15—Q <sub>1</sub>
C <sub>D1</sub> =3	14— $\bar{Q}_1$
J <sub>1</sub> =4	13—GND
V <sub>CC</sub> =5	12—K <sub>2</sub>
CP <sub>2</sub> =6	11—Q <sub>2</sub>
S <sub>D2</sub> =7	10— $\bar{Q}_2$
C <sub>D2</sub> =8	9—J <sub>2</sub>

(b) Pin configuration

(IC 3626)Fig. 4.23.5

**Review Questions**

- Q. 1 Mention the types of digital systems. Explain with block diagram their operating principle.
- Q. 2 What is a flip-flop ?
- Q. 3 State and explain the triggering methods used for flip-flops.
- Q. 4 What is the function of preset and clear inputs in flip-flop?
- Q. 5 Explain with truth table the working of clocked RS flip-flop.
- Q. 6 State the disadvantages of RS flip flop. How can they be avoided ?
- Q. 7 Explain with diagram the working of D type flip-flop. Give its truth table.
- Q. 8 Give reason why D flip-flop is called as data latch ?
- Q. 9 Draw the circuit using logic gates of a T-type flip-flop. Draw its symbol and write its truth table.
- Q. 10 Explain S-R flip flop using NOR gates.
- Q. 11 Describe how two cross-coupled NAND gates form a R-S flip-flop ? Write its truth table.
- Q. 12 What are the various types of flip-flops ?
- Q. 13 Draw the circuit of SR flip-flop using NAND gate.
- Q. 14 Draw the schematic diagram of JK flip-flop and describe its working. Write down its truth table.
- Q. 15 What is race around condition ?
- Q. 16 Draw the circuit of J-K flip-flop using NAND gate.
- Q. 17 Draw a neat diagram of master slave J-K flip-flop. Explain how race around condition is avoided using master slave J-K flip-flop ?
- Q. 18 Explain the working of the master slave JK flip-flop.
- Q. 19 Can a flip flop be used as a memory ? If so how many bits can be stored by R-S flip flop ?
- Q. 20 Explain T flip flop.
- Q. 21 Explain the following flip-flops :
1. Clocked SR
  2. JK with preset and clear
  3. Master Slave JK
  4. D type and T type
- Q. 22 Design a conversion logic to convert a JK flip flop to a D flip flop.
- Q. 23 Write a short note on race around condition in JK flip flop.
- Q. 24 Draw neat circuit diagram of clocked JK flip-flop using NAND gates. Give its truth table and explain race around condition.
- Q. 25 What is race-around condition ? How does it get eliminated in Master Slave JK FF ? Explain.
- Q. 26 Explain how JK FF is converted into :
  1. D FF
  2. T FF
- Q. 27 Carry out the following flip flop conversions (Conversion tables and K-maps expected) :
  1. S-R to D
  2. D to S-R
  3. J-K to S-R
- Q. 28 If  $\bar{Q}$  output of a D-type flip-flop is connected to D input, it acts as a toggle switch. State whether true or false ? Justify your answer.
- Q. 29 What is the basic difference between pulse-triggered and edge-triggered flip-flops ?



## Unit IV

# Chapter 5

## Registers

### Syllabus

Registers, Shift registers, Counters (Ring counters, Twisted ring counters), Sequence Generators.

### Chapter Contents

5.1 Introduction	5.8 Parallel In Parallel Out (PIPO)
5.2 Data Formats	5.9 Bidirectional Shift Register
5.3 Classification of Registers	5.10 Universal Shift Register
5.4 Buffer Registers	5.11 Applications of Shift Registers
5.5 Shift Registers	5.12 Ring Counter
5.6 Serial In Parallel Out (SIPO)	5.13 Johnson's Counter (Twisted / Switch Tail Ring Counter)
5.7 Parallel In Serial Out Mode (PISO)	5.14 Sequence Generator



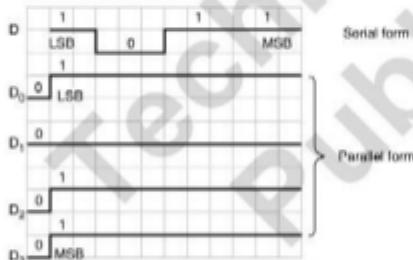
### 5.1 Introduction :

#### Definition :

- In chapter 4, we have stated various applications of flip-flops. One of them was "registers".
- 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-flops. Such a group of flip-flops is known as a **register**.
- Thus register is a group of flip-flops. The "n-bit" register will consist of "n" number of flip-flops and it is capable of storing an "n-bit" word.

### 5.2 Data Formats :

- The data can be entered in **serial** (one bit at a time) manner or in the **parallel** form (all the bits at the same time) into a register.
- And the stored data can be retrieved in the serial or parallel form.

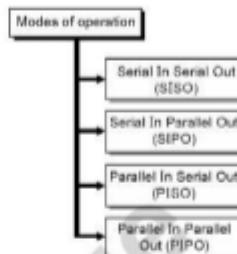


(C-729)(i) Fig. 5.2.1 : Data representation in serial and parallel forms

- A four bit digital data 1 0 1 1 is represented in the serial and parallel forms in Fig. 5.2.1.

### 5.3 Classification of Registers :

- Registers are classified on the basis of how the data bits are entered and taken out from a register. There are four possible modes as follows :



(C-729)(ii) Fig. 5.3.1 : Modes of operation of registers

- We can design all these registers using discrete flip-flops such as SR, JK or D flip-flops.
- But the registers are also available in the IC form.
- Some of the registers available in 54/74 TTL series as listed in Table 5.3.1.

Table 5.3.1 : Shift registers available in 74/54 series

IC number	Description
7491, 7491A	8 bit serial in, serial out
7494	4 bit parallel in, serial out
7495	4 bit serial/parallel in, parallel out (shift right, shift left)
7496	5 bit parallel in / parallel out, serial in / serial out.

### 5.4 Buffer Registers :

#### Logic diagram :

- The simplest type of register constructed using four D flip-flops is shown in Fig. 5.4.1.
- This is a 4 bit register, but we can construct an n-bit register by following the same principle.
- This register is also called as the **buffer register**.
- Each D-flip-flop is negative edge triggered and all the flip-flops are connected to a common clock signal.
- Hence all of them are triggered at the same instant of time, i.e. all the flip flops will change their state at the same instant of time.
- Buffer registers are used for temporary storage of digital words.

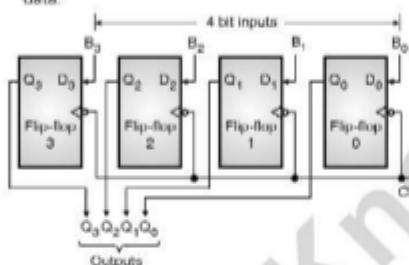
#### Operation :

- Let us assume that the word to be stored is  $B_3 B_2 B_1 B_0 = 1 0 1 0$ .



- These bits are connected to the D inputs of the four D flip-flops as shown in Fig. 5.4.1.
- Then the clock pulse is applied.
- Corresponding to the first negative edge of the clock pulse, the outputs of all the D flip-flops will follow their respective inputs.  

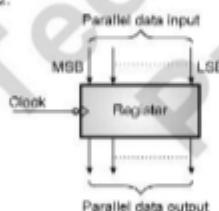
$$\therefore Q_3 Q_2 Q_1 Q_0 = B_3 B_2 B_1 B_0 = 1010$$
- Even if the inputs are now changed, the output remains latched to 1010 till the next negative edge of the clock arrives at the input.
- Thus the buffer register is capable of storing the digital data.



(C-729(j)) Fig. 5.4.1 : A four bit buffer register using D flip-flops

**Schematic diagram :**

- The schematic diagram of a buffer register is as shown in Fig. 5.4.2.



(C-730) Fig. 5.4.2 : Schematic diagram of buffer register

**Conclusions :**

- Some of the important conclusions from the discussion till now are as follows :
1. There must be one-FF for each bit to be stored. Therefore to store a 4 bit number we need four flip-flops.
  2. Note that all the four input bits  $B_3 B_2 B_1 B_0$  are loaded into the buffer register simultaneously, i.e. at the same instant of time.

- 3. Therefore this way of applying the input and taking the output is called as Parallel Input Parallel Output (PIPO) and the mode of operation is called as **parallel shifting**.

**5.5 Shift Registers :**

SPPU : May 07, May 18

**University Questions**

- Q. 1** Explain modes of operation of shift register. (May 07, 4 Marks)  
**Q. 2** State the types of shift register and explain any one of them. (May 18, 4 Marks)

**Definition :**

- The binary data in a register can be moved within the register from one flip-flop to the other or outside it with application of clock pulses.
- The registers that allow such data transfers are called as shift registers.
- Shift registers are used for data storage, data transfer and certain arithmetic and logic operations.

**Modes of operation of a shift register :**

- The various modes in which a shift register can operate are as follows :
  1. Serial Input Serial Output. (SISO).
  2. Serial Input Parallel Output. (SIPO).
  3. Parallel In Serial Out. (PISO).
  4. Parallel In Parallel Out. (PIPO).
- These modes are explained in brief in Table 5.5.1.

Table 5.5.1 : Brief explanation of various modes of shift register

Sr. No.	Mode	Illustrative diagram	Comments
1.	Serial input serial output (serial shift right)	Refer Fig. A	Data bits shift from left to right by 1 position per clock cycle.
2.	Serial input serial output (serial shift left)	Refer Fig. B	Data bits shift from right to left by 1 position per clock.
3.	Serial input parallel output	Refer Fig. C	All output bits are made available simultaneously after 4-clock pulses.



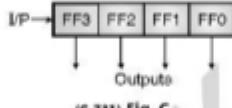
Sr. No.	Mode	Illustrative diagram	Comments
4.	Parallel input serial output	Refer Fig. D	All inputs are loaded simultaneously but output bit by bit.
5.	Parallel input parallel output	Refer Fig. E	All inputs are loaded simultaneously and are available at the output simultaneously.



(C-731) Fig. A



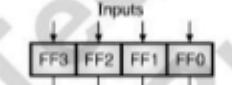
(C-731) Fig. B



(C-731) Fig. C



(C-731) Fig. D



(C-731) Fig. E

### 5.5.1 Serial Input Serial Output (Shift Left Mode) :

SPPU : May 06

#### University Questions

- Q. 1 Draw the logical diagram of a 4-bit shift register. Explain how shift left operation is performed.

(May 06, 4 Marks)

#### Principle :

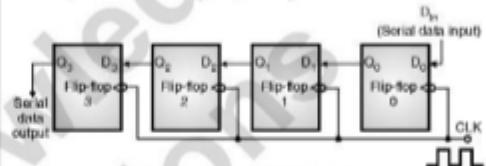
- Data bits shift from right to left by 1 position per clock cycle as shown in Fig. 5.5.1(a).



(C-731) Fig. 5.5.1(a) : Principle of Serial shift left register

#### Logic diagram :

- The serial input serial output type shift register with shift left mode is shown in Fig. 5.5.1(b).
- Let all the flip-flops be initially in the reset condition i.e. Q<sub>3</sub> = Q<sub>2</sub> = Q<sub>1</sub> = Q<sub>0</sub> = 0.
- We are going to illustrate the entry of a four bit binary number 1 1 1 1 into the register.
- When this is to be done, this number should be applied to "D<sub>n</sub>" bit by bit with the MSB bit applied first.
- The D input of FF-0 i.e. D<sub>0</sub> is connected to serial data input (D<sub>n</sub>). Output of FF-0 i.e. Q<sub>0</sub> is connected to the input of the next flip-flop i.e. D<sub>1</sub> and so on.

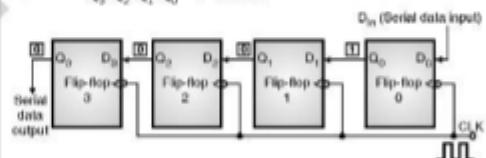


(C-732) Fig. 5.5.1(b) : Serial shift left register

#### Operation :

- Before application of clock signal let Q<sub>3</sub>, Q<sub>2</sub>, Q<sub>1</sub>, Q<sub>0</sub> = 0 0 0 0 and apply MSB bit of the number to be entered to D<sub>n</sub>. So D<sub>0</sub> = D<sub>n</sub> = 1.
- Apply the clock. On the first falling edge of clock, the FF-0 is set and the stored word in the register is,

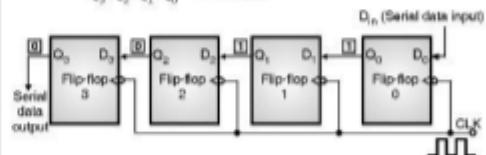
$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 0 \ 0 \ 1$$



(C-733) Fig. 5.5.2 : Shift register status after first falling clock edge

- Apply the next bit to D<sub>n</sub>. So D<sub>n</sub> = 1.
- As soon as the next negative edge of the clock hits, FF-1 will set and the stored word changes to,

$$Q_3 \ Q_2 \ Q_1 \ Q_0 = 0 \ 0 \ 1 \ 1$$



(C-734) Fig. 5.5.3 : Shift register status after the second falling edge of clock



- 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-2 will be set and the output get modified to,

$$Q_3 Q_2 Q_1 Q_0 = 0111$$

- Similarly with  $D_{in} = 1$ , and with the fourth negative clock edge arriving, the stored word in the register is,

$$Q_3 Q_2 Q_1 Q_0 = 1111$$

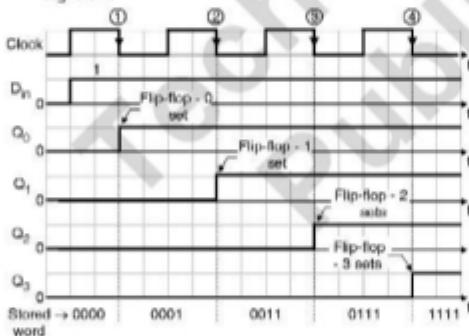
(C-737) Table 5.5.2 : Summary of shift left operation

	CLK	$Q_3$	$Q_2 = D_3$	$Q_1 = D_2$	$Q_0 = D_1$	Serial input $D_{in} = D_0$
Initially		0	0	0	0	
1 <sup>st</sup>	↓	0	0	0	1	1
2 <sup>nd</sup>	↓	0	0	1	1	1
3 <sup>rd</sup>	↓	0	1	1	1	1
4 <sup>th</sup>	↓	1	1	1	1	1

Direction of data travel →

**Waveforms for shift left operation :**

- The waveforms for the shift left operation are shown in Fig. 5.5.4.



(C-738) Fig. 5.5.4 : Waveforms for the shift left operation

**Important note :** Using the SISO mode, we needed 4-clock pulses to store a 4-bit word. So in general we can conclude that if it requires  $n$  number of clock pulses to store an  $n$  bit word using SISO mode.

**5.5.2 Serial In Serial Out (Shift Right Mode) :**

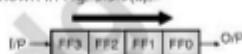
SPPU : May 06

**University Questions**

- Q. 1** Draw the logical diagram of a 4-bit shift register. Explain how shift right operation is performed.  
(May 06, 4 Marks)

**Principle :**

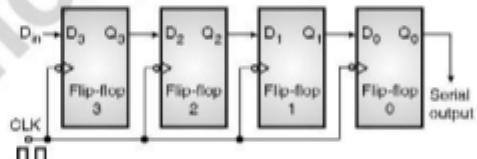
- Data bits shift from left to right by 1 position per clock cycle as shown in Fig. 5.5.5(a).



(C-731) Fig. 5.5.5(a) : Principle of Serial shift right register

**Logic diagram :**

- The serial input serial output type shift register with shift right mode is shown in Fig. 5.5.5(b).
- Let all the flip-flops be initially in the reset condition i.e.  $Q_3 = Q_2 = Q_1 = Q_0 = 0$ .
- We are going to illustrate the entry of a four bit binary number 1 1 1 1 into the register.
- When this is to be done, this number should be applied to ' $D_{in}$ ' bit by bit with the LSB bit applied first.



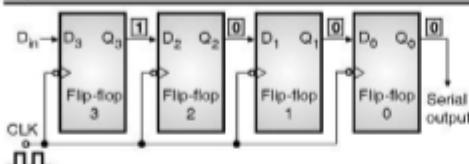
(C-739) Fig. 5.5.5(b) : Serial shift right register

- 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.

**Operation :**

- Before application of clock signal let  $Q_3 Q_2 Q_1 Q_0 = 0000$ .  
0 0 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 the stored word in the register is,

$$Q_3 Q_2 Q_1 Q_0 = 1000$$

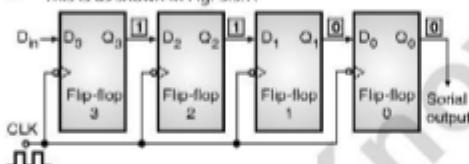


(C-740) Fig. 5.5.6 : Shift register status after first falling clock edge

- The shift register after the application of the first clock pulse is as shown in Fig. 5.5.6.
- Apply the next bit to  $D_3$ . So  $D_{in} = 1$ .
- As soon as the next negative edge of the clock is applied, FF-2 will set and the stored word changes to,

$$Q_3 Q_2 Q_1 Q_0 = 1100$$

- This is as shown in Fig. 5.5.7.



(C-741) Fig. 5.5.7 : Shift register status after the second falling edge of clock

- 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 gets applied, FF-1 will be set and the output get modified to,

$$Q_3 Q_2 Q_1 Q_0 = 1110$$

- Similarly with  $D_{in} = 1$  and with the fourth negative clock edge arriving, the stored word in the register is,

$$Q_3 Q_2 Q_1 Q_0 = 1111$$

Table 5.5.3 summarizes the shift right operation.

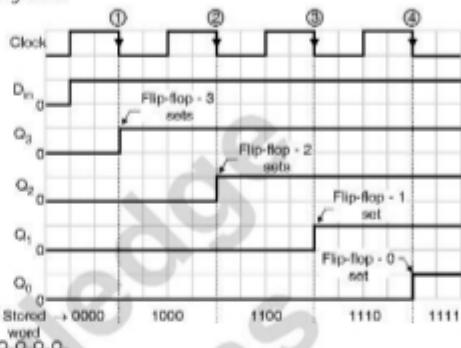
(C-744) Table 5.5.3 : Summary of shift right operation

	CLK	$D_{in} = D_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	$Q_0$
Initially			0	0	0	0
1 <sup>st</sup>	↓	1 → 1	* 0	* 0	* 0	* 0
2 <sup>nd</sup>	↓	1 → 1	* 1	* 0	* 0	* 0
3 <sup>rd</sup>	↓	1 → 1	* 1	* 1	* 1	* 0
4 <sup>th</sup>	↓	1 → 1	* 1	* 1	* 1	* 1

→ Direction of data travel

#### Waveforms for shift right operation :

The waveforms for shift right operation are as shown in Fig. 5.5.8.

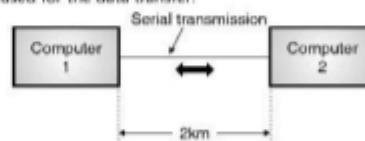


(C-745) Fig. 5.5.8 : Waveforms for the shift right operation

**Important note :** Using the SISO mode, we needed 4-clock pulses to store a 4-bit word. So in general we can conclude that it requires  $n$  number of clock pulses to store an  $n$  bit word using SISO mode.

#### 5.5.3 Applications of Serial Operation :

- The transmission of data from one place to the other takes place in serial manner as shown in Fig. 5.5.9.
- The serial shifting of data transmits one bit at a time per clock cycle.
- It takes a longer time for serial transmission, because the time required to transmit one bit is equal to the time corresponding to one clock cycle.
- The parallel transmission is much faster as it can transmit 8 bits per clock cycle. But it needs more wires for connecting a transmitter to receiver.
- However for long distance communication where the distances are in kilometres, serial communication has an advantage that only one conductor is required to be used for the data transfer.



(C-746) Fig. 5.5.9 : Application of serial operation

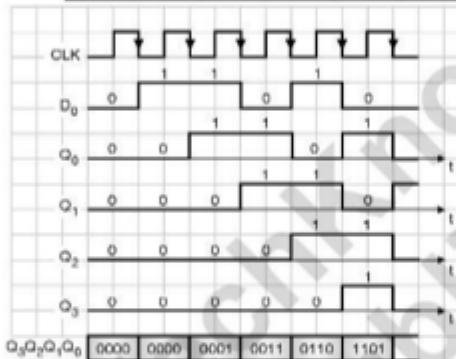


**Ex. 5.5.1 :** For a 4 bit negative edge triggered SISO shift register, draw the output waveform if the input is 01101.

Soln. :

(C-3168) Table P. 5.5.1

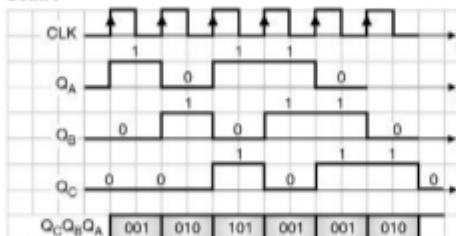
	CLK	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	D <sub>in</sub>
Initially		0	0	0	0	
1 <sup>st</sup>	↓	0*	0*	0*	0	0
2 <sup>nd</sup>	↓	0*	0*	0*	1	1
3 <sup>rd</sup>	↓	0*	0*	1*	1	1
4 <sup>th</sup>	↓	0*	1*	1*	0	0
5 <sup>th</sup>	↓	1*	1*	0	1	1



(C-3169) Fig. P. 5.5.1 : Waveforms with negative edge triggering

**Ex. 5.5.2 :** For 3 bit SISO shift register, draw the output waveform for positive edge trigger clock with input as 10110.

Soln. :



(C-3171) Fig. P. 5.5.2 : Waveforms with positive edge triggering

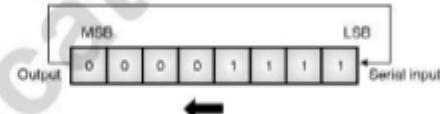
**Ex. 5.5.3 :** Draw the circuit using flip-flop to multiply the number by 16 and explain how the same circuit can be used as divide by 8 ?

**Dec. 07, 4 Marks**

Soln. :

Circuit to multiply the number by 16 :

- Let the given number be a 4 bit number. Let us use an 8 bit shift register.
- First load the number with the parallel loading mode. For example, if the number  $N = \{15\}_{10}$ , then load the 8-bit version of this number i.e.,  $N = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)_2 = \{15\}_{10}$ .
- The shift register is configured to operate in the serial shift left mode with MSB output returned to serial input.
- With every clock pulse, the number is shifted left by one place and the left shift is equivalent to multiplication by 2.
- If we left shift the number four times, then it will be multiplied by 16.



(C-1364(a)) Fig. P. 5.5.3

Table P. 5.5.3

Clock	Shift register contents	Decimal equivalent	Multipled by
-	0 0 0 0 1 1 1 1	15	-
1	0 0 0 1 1 1 1 0	30	2
2	0 0 1 1 1 1 0 0	60	4
3	0 1 1 1 1 0 0 0	120	8
4	1 1 1 1 0 0 0 0	240	16

## 5.6 Serial In Parallel Out (SIPO) :

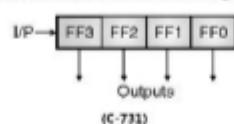
SPPU : May 18, Dec. 19

### University Questions

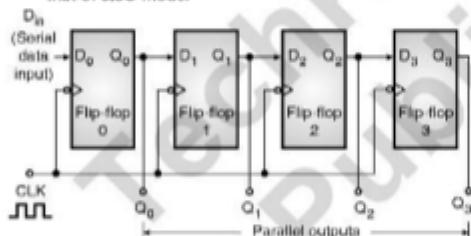
- Q. 1** State the types of shift register and explain any one of them. (May 18, 4 Marks)
- Q. 2** Draw circuit diagram of 3-bit SIPO shift register using D flipflop. Explain its working. (Dec. 19, 6 Marks)

**Principle :**

- In this operation the data is entered serially and taken out in parallel as shown in the following figure.



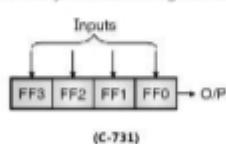
- In this operation the data is entered serially and taken out in parallel.
- That means first the data is loaded bit by bit. The outputs are disabled as long as the loading is taking place.
- As soon as the loading is complete, and all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines simultaneously.
- Number of clock cycles required to load a four bit word is 4.
- Hence the speed of operation of SIPO mode is same as that of SISO mode.



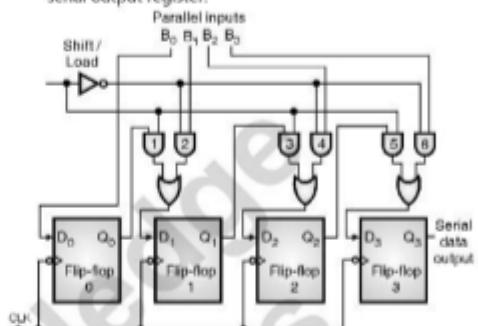
(C-747) Fig. 5.6.1 : Serial input parallel output mode

**5.7 Parallel In Serial Out Mode (PISO) :****Principle :**

- In this mode, the bits are entered in parallel i.e. simultaneously into a shift register as shown below.

**Logic diagram :**

- The circuit shown in Fig. 5.7.1 is a four bit parallel input serial output register.



(C-748) Fig. 5.7.1 : Parallel in serial out shift register

- Output of previous FF 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 :**

- In order to load the word  $B_0, B_1, B_2, B_3$  into the shift register we have to select the load mode by setting shift/ load input to 0.
- When the shift / load line is low (0), the AND gates 2, 4 and 6 become active. They will pass  $B_0, B_1$  and  $B_3$  bits to the inputs  $D_1, D_2$  and  $D_3$  of the corresponding flip-flops.  $D_0$  is directly connected to  $B_0$ .
- On the low going edge of clock, the binary inputs  $B_0, B_1, B_2, B_3$  will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

**Shift mode :**

- In order to operate the shift register in the shift mode we have to select the shift mode by applying a logic 1 to the shift/ load input.
- When the shift / load line is high (1), the AND gates 2, 4, 6 become inactive. Hence the parallel loading of the data becomes impossible.

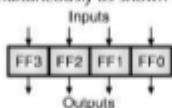


- But the AND gates 1, 3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses becomes possible.  $D_s$  acts as the data input terminal and at  $Q_s$  we get the serial data output.
- Thus the parallel in serial out operation takes place.

### 5.8 Parallel In Parallel Out (PIPO) :

#### Principle :

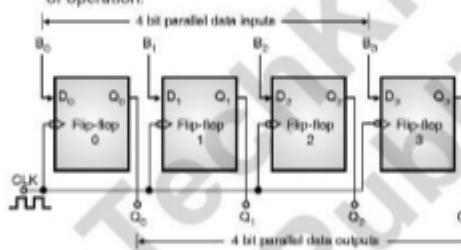
- All inputs are loaded simultaneously and are available at the output simultaneously as shown below.



(C-3167) Fig. 5.8.1 : Principle of PIPD register

#### Logic diagram :

- Fig. 5.8.2 demonstrates the parallel in parallel out mode of operation.



(C-749) Fig. 5.8.2 : Parallel in parallel out shift register

- The 4 bit binary input  $B_0, B_1, B_2, B_3$  is applied to the data inputs  $D_0, D_1, D_2$ , and  $D_3$  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 one clock pulse is essential to load all the bits. Therefore PIPD mode is the fastest mode of operation.

### 5.9 Bidirectional Shift Register :

SPPU : May 07, May 08, Dec. 08

#### University Questions

- Q. 1** Draw 4-bit bidirectional shift register.

(May 07, 4 Marks)

- Q. 2** Draw the logical diagrams of 4-bit bidirectional shift register. Explain shift left and shift right operations.

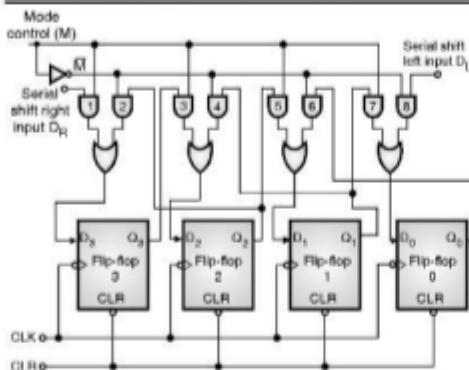
(May 08, Dec. 08, 8 Marks)

#### Principle :

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2.
- On the other hand if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2. This is illustrated below.
- Let a four bit number  $Q_3, Q_2, Q_1, Q_0 = 0010 = (2)_{10}$  is existing in a shift register.
- Now with a 0 applied at the input, if we shift these contents by one position to left then we get  $Q_3, Q_2, Q_1, Q_0 = 0100 = (4)_{10}$ .
- Thus the shift left is equivalent to multiplying by 2. Now shift it right by one position to get  $Q_3, Q_2, Q_1, Q_0 = 0001 = (1)_{10}$ . Thus shifting right is equivalent to dividing 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 as and when we want.

#### Block diagram :

- Such a register is called as a bi-directional register. A four bit bi-directional shift register is shown in Fig. 5.9.1.
- There are two serial inputs namely the serial right shift data input  $D_s$  and the serial left shift data input  $D_L$  alongwith a Mode control input (M).



(C-750) Fig. 5.9.1 : A 4-bit bi-directional shift register

**Operation :****With  $M = 1$  : Shift right operation**

- 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.
- Hence the data at  $D_R$  (shift right input) is shifted 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.

**With  $M = 0$  : Shift left operation**

- 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.
- Therefore the data at  $D_L$  (shift left input) is shifted left bit by bit from FF-0 to FF-3 on application of the clock pulses.
- Thus with  $M = 0$  we get the serial left shift operation.
- Note that M should be changed only when CLK = 0, otherwise the data stored in the register may get changed in an undesirable manner.

**5.9.1 A 3-bit Bidirectional Register using the JK Flip Flops :** SPPU : Dec. 06, Dec. 09**University Questions**

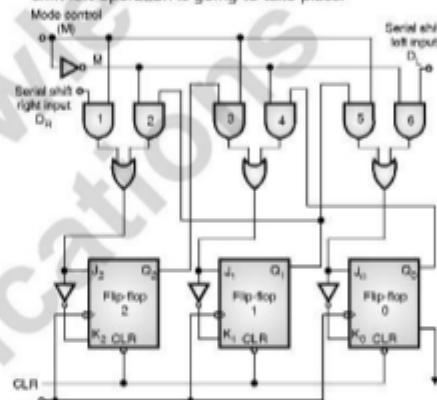
- Q. 1** Draw neat circuit diagram of 3-bit bi-directional shift register using JK flip-flop and explain its operation.

(Dec. 06, 6 Marks)

**Q. 2** Draw and explain the operation of 3-bit bidirectional shift register. (Dec. 09, 4 Marks)

**Logic diagram :**

- A 3-bit bi-directional shift register using JK flip-flops is shown in Fig. 5.9.2.
- This circuit is same as the 4-bit bi-directional shift register discussed in the previous section.
- The only modification is the inclusion of an inverter between the J and K inputs of each flip-flop.
- With  $M = 1$  the shift right operation will take place as discussed in the previous section and with  $M = 0$  the shift left operation is going to take place.



(C-751) Fig. 5.9.2 : A 3-bit bi-directional shift register using JK flip flops

**5.10 Universal Shift Register :**

SPPU : Dec. 05, Dec. 18

**University Questions**

- Q. 1** Explain with the help of neat diagram the operation of a 3-bit universal shift register. (Dec. 05, 8 Marks)

- Q. 2** Explain universal shift registers with a diagram. (Dec. 18, 6 Marks)

**Definition :**

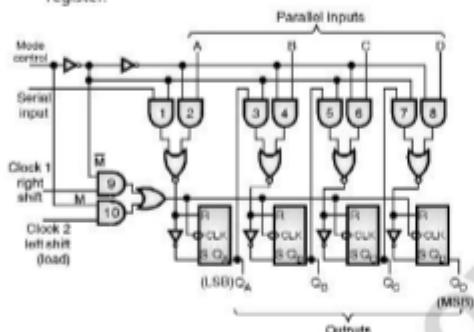
- A shift register which can shift the data in only one direction is called as a unidirectional shift register.
- A shift register which can shift the data in both the directions is called as a bi-directional shift register.



- Applying the same logic, a shift register which can shift the data in both the directions (shift right or left) as well as load it parallel, then it is called as a **universal shift register**.

#### Logic diagram :

- Fig. 5.10.1 shows the logic diagram of a universal shift register.



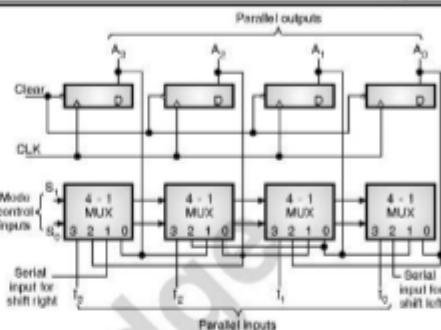
(C-752) Fig. 5.10.1 : Logic diagram of a universal shift register

- This shift register is capable of performing the following operations :
  - Parallel loading (parallel input parallel output).
  - 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 in Fig. 5.10.1.
- Whereas for the shift right operation, the serial input is applied to D input (input of AND gate 8).
- The well known example of universal shift register in the IC form is IC 7495.

#### 5.10.1 Universal Shift Register using Multiplexers and D-flip flops :

##### Logic diagram :

- Fig. 5.10.2 shows the diagram of a 4 bit universal shift register. It consists of four D-flip flops and four multiplexers.



(C-753) Fig. 5.10.2 : 4-Bit universal shift register

#### Operation :

- The  $S_1$  and  $S_0$  inputs are the select lines of all the four multiplexers and they are connected together. These lines act as the **mode control inputs**.
- The mode of operation is dependent on the status of the  $S_1$  and  $S_0$  lines as shown in Table 5.10.1.

(C-807) Table 5.10.1 : Function table of universal shift register

Mode control		
$S_1$	$S_0$	Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

##### 1. Operation for $S_1 S_0 = 00$ (No change) :

- When  $S_1 S_0 = 00$  input 0 of each multiplexer is selected.
- As the output of each FF is connected to input 0 of the corresponding MUX, the present value of the register is applied to D inputs of the flip flops.
- On the next positive clock edge, this present value gets transferred to the output.
- That means, the next state will be same as the present state. Thus there is **no change** in state for  $S_1 S_0 = 00$ .

##### 2. Operation for $S_1 S_0 = 01$ (Right shift) :

- When  $S_1 S_0 = 01$ , terminal 1 of multiplexer input has a path to flip flop input D. This will cause the shift right operation.

**3. Operation for  $S_1 S_0 = 10$  (Left shift) :**

- When  $S_1 S_0 = 10$ , the input 2 of each multiplexer get connected to the D input of corresponding flip flop.
- This will result in shift left operation.

**4. Operation for  $S_1 S_0 = 11$  (Parallel load) :**

- When  $S_1 S_0 = 11$ , the input 3 of each multiplexer gets connected to the D input of corresponding flip flop.
- This will result in the parallel load operation.

**5.11 Applications of Shift Registers :****SPPU : May 05****University Questions****Q. 1** State various applications of shift registers.**(May 05, 3 Marks)**

- Some of the common applications of a shift register are :
- 1. For temporary data storage.
- 2. For multiplication and division.
- 3. As a delay line.
- 4. Serial to parallel converter.
- 5. Parallel to serial converter.
- 6. Ring counter.
- 7. Twisted ring counter or Johnson counter.
- 8. Serial data transmission.
- We have already seen the use of shift register for temporary storage of data and for multiplication or division.

**5.11.1 Serial to Parallel Converter :****SPPU : May 10****University Questions****Q. 1** Explain how shift register is used as serial to parallel converter. **(May 10, 2 Marks)**

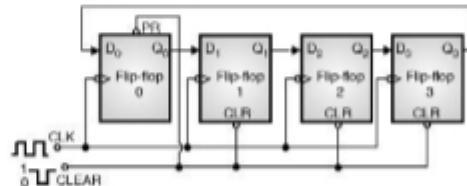
- In many applications, the data in serial form needs to be converted in the parallel form.
- The Serial Input Parallel Output Mode (SIPO) of the shift register is used for such applications.

**5.11.2 Parallel to Serial Converter :****SPPU : May 10****University Questions****Q. 1** Explain how shift register is used as parallel to serial converter. **(May 10, 2 Marks)**

- The data communication between two computers takes place in the serial transmission form.
- But internally the data processing takes place in the parallel form.
- Hence we need to use a parallel to serial converter to convert the internal parallel data into a serial data suitable for transmission.
- We can use the parallel to serial mode of the shift register for this operation.

**5.12 Ring Counter :****SPPU : May 10, May 12, Dec. 17****University Questions****Q. 1** Explain how shift register is used as ring counter ? **(May 10, May 12, 4 Marks)****Q. 2** Draw and explain 4-bit ring counter. **(Dec. 17, 6 Marks)****Logic diagram :**

- Fig. 5.12.1 shows a typical application of shift registers called Ring Counter.

**(C-871) Fig. 5.12.1 : A four bit ring counter**

- The connections reveal that they are similar to the connections for shift right operation, except for one change.
- Output of FF-3 is connected to data input  $D_0$  of FF-0. Ring counter is a special type of shift register.

**Operation :**

- Initially a low clear (CLR) pulse is applied to all the flip-flops.



- Hence FF-3, FF-2 and FF-1 will reset but FF-0 will be preset. So the outputs of the shift register are :

$$Q_3 Q_2 Q_1 Q_0 = 0\ 0\ 0\ 1.$$

- Now the clear terminal is made inactive by applying a high level to it.
- The clock signal is then applied to all the flip-flops simultaneously.
- Note that all the flip-flops are negative edge triggered.

**On the first negative going CLK edge :**

- As soon as the first falling edge of the clock hits, only FF-1 will be set because  $D_0 = D_1 = 1$ .
- The FF-0 will reset because  $D_0 = Q_3 = 0$  and there is no change in the status of FF-2 and FF-3.
- Hence after the first clock pulse the outputs are

$$Q_3 Q_2 Q_1 Q_0 = 0\ 0\ 1\ 0.$$

**On the second falling edge of clock :**

- At the second falling edge of the clock, only FF-2 will be set because  $D_2 = Q_1 = 1$ .
- FF-1 will reset since  $D_1 = Q_0 = 0$ . There is no change in status of FF-3 and FF-0.
- So after the second clock pulse the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0\ 1\ 0\ 0.$$

- Similarly after the third clock pulse the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 1\ 0\ 0\ 0.$$

- And after the fourth one the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0\ 0\ 0\ 1.$$

- These are the outputs from where we started. Hence the operation repeats from this point onwards.

**Number of output states :**

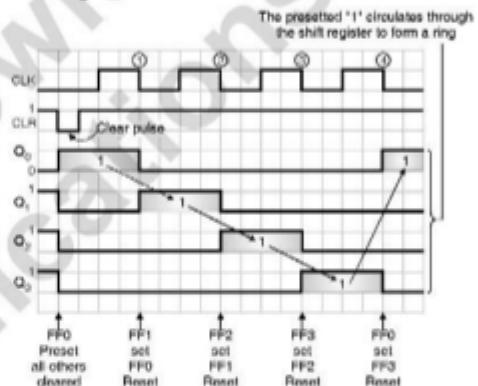
- The number of output states for a ring counter will always be equal to the number of flip-flops.
- So for a 4-bit ring counter the number of states is equal to 4.
- The operation of a four bit ring counter is summarized in Table 5.12.1.

(C-871(a)) Table 5.12.1 : Summary of operation of a ring

counter						
CLR	CLK	$Q_3$	$Q_1$	$Q_2$	$Q_0$	
	X	1	0	0	0	
1	↓	0	1	0	0	← FF-0 Preset, others cleared
1	↓	0	0	1	0	The presetted 1 follows a circular path to form a ring
1	↓	0	0	0	1	
1	↓	1	0	0	0	

**Waveforms for the ring counter :**

- The waveforms for the 4-bit ring counter are as shown in Fig. 5.12.2.



(C-1360) Fig. 5.12.2 : Waveforms of a four bit ring counter

- These waveforms clearly show that the presetted "1" shifts one bit per clock cycle and forms a ring. Hence the name ring counter.

**Applications of ring counter :**

- Ring counters are used in those applications in which several operations are to be controlled in a sequential manner.
- For example in resistance welding the operations called squeeze, hold, weld and off are to be performed sequentially.
- We can use a ring counter to initiate these operations.



### 5.13 Johnson's Counter (Twisted / Switch Tail Ring Counter) :

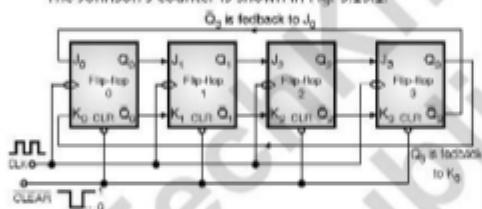
SPPU : May 07, Dec. 08, May 12, May 19

#### University Questions

- Q. 1** Design Johnson's counter using 2-bit shift register. Draw waveforms. (May 07, Dec. 08, 4 Marks)
- Q. 2** Explain how shift registers are used as twisted ring counter. (May 12, 4 Marks)
- Q. 3** Design 3 bit twisted ring counter. (May 19, 6 Marks)

#### Logic diagram :

- In the ring counter the outputs of FF-3 were connected directly to the inputs of FF-0 i.e.  $Q_3$  to  $J_0$ ,  $\bar{Q}_3$  to  $K_0$ .
- Instead if the outputs are cross coupled to the inputs i.e. if  $Q_j$  is connected to  $K_0$  and  $\bar{Q}_j$  is connected to  $J_0$  then the circuit is called as twisted ring counter or Johnson's counter.
- The Johnson's counter is shown in Fig. 5.13.1.



(C-1302) Fig. 5.13.1 : Twisted ring counter or Johnson counter

- All the flip-flops are negative edge triggered, and clock pulses are applied to all of them simultaneously.
- The clear inputs of all the flip-flops are connected together and connected to an external clear signal. Note that all these clear inputs are active low inputs.

#### Operation :

- Initially a short negative going pulse is applied to the clear input of all the flip-flops. This will reset all the flip-flops. Hence initially the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 0 0 0 0$ .
- But  $\bar{Q}_3 = 1$  and since it is coupled to  $J_0$  it is also equal to 1.  
 $\therefore J_0 = 1$  and  $K_0 = 0$  .... Initially

#### On the first falling edge of clock pulse :

- As soon as the first negative edge of clock arrives, FF-0 will be set. Hence  $Q_0$  will become 1.
- But there is no change in the status of any other flip-flop.
- Hence after the first negative going edge of the clock the flip-flop outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 0 0 0 1$

#### On the second negative going clock edge :

- Before the second negative going clock edge,  $Q_3 = 0$  and  $Q_2 = 1$ . Hence  $J_0 = 1$  and  $K_0 = 1$ . Also  $Q_0 = 1$ . Hence  $J_1 = 1$ .
- Hence as soon as the second falling clock edge arrives, FF-0 continues to be in the set mode and FF-1 will now set. Hence  $Q_1$  will become 1 and  $\bar{Q}_1 = 0$ .
- There is no change in the status of any other flip-flop.
- Hence after the second clock edge the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 0 0 1 1$ .

Similarly after the third clock pulse, the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0 1 1 1$$

And after the fourth clock pulse, the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 1 1 1 1$$

Note that now  $\bar{Q}_3 = 0$  i.e.  $J_0 = 0$  and  $K_0 = 1$ .

- Hence as soon as the fifth negative going clock pulse strikes, FF-0 will reset.
- But the outputs of the other flip-flops will remain unchanged. So after the fifth clock pulse, the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 1 1 1 0$  .... after the 5<sup>th</sup> clock pulse
- This operation will continue till we reach the all zero output state. (i.e.  $Q_3 Q_2 Q_1 Q_0 = 0 0 0 0$ ).
- The operation of Johnson's counter is summarised in Table 5.13.1.



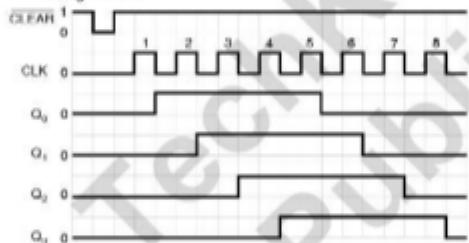
(C-6295) Table 5.13.1 : Summary of operation of Johnson's counter

CLEAR	CLK	$Q_3$	$Q_2$	$Q_1$	$Q_0$	State number	Decimal equivalent
1	↓	0	0	0	1	1	0
1	↓	0	0	1	1	3	3
1	↓	0	1	1	1	4	7
1	↓	1	1	1	1	5	15
1	↓	1	1	1	0	6	14
1	↓	1	1	0	0	7	12
1	↓	1	0	0	0	8	8
1	↓	0	0	0	0	1	0

- Note that there are 8 distinct states of output.
- In general we can say that the number of states of a Johnson's counter is twice the number of flip-flops used. Therefore for a 4-flip-flop Johnson's counter, there are 8-distinct output states.

#### Waveforms for Johnson's counter :

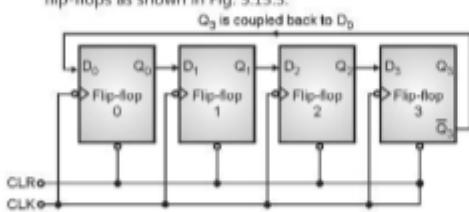
- The waveforms for a 4-bit Johnson's counter are shown in Fig. 5.13.2.



(C-875) Fig. 5.13.2 : Waveforms of Johnson counter

#### 5.13.1 Johnson's Counter using D Flip-flops :

- It is possible to construct the Johnson counter using D flip-flops as shown in Fig. 5.13.3.



(C-1164(b)) Fig. 5.13.3 : Johnson counter using D flip-flops

- The operation of this circuit is summarized in Table 5.13.2.

(C-6296) Table 5.13.2 : Summary of operation of Johnson counter

CLR	CLK	$Q_0$	$Q_1$	$Q_2$	$Q_3$
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1
1	↓	0	0	0	1
1	↓	0	0	0	0

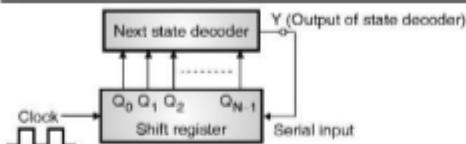
## 5.14 Sequence Generator :

#### Definition :

- A sequence generator is a sequential circuit which generates a desired sequence at its output. The output sequence is in synchronization with the clock input.
- It is possible to design a sequence generator using counters or using the shift registers.

#### 5.14.1 Sequence Generator using Shift Register :

- The sequence generator is a circuit which generates a desired sequence of bits at its output in synchronization with the clock.
- Some of the applications of the sequence generator are as follows :
  - Random bit generator
  - Counters
  - Code generators
  - Period and sequence generator.
- Fig. 5.14.1 shows the basic structure of a sequence generator using a shift register.
- The outputs of an N bit shift register ( $Q_0$  through  $Q_{n-1}$ ) are applied as inputs to a combinational circuit called "Next state decoder".



(C-1513) Fig. 5.14.1 : Basic structure of a sequence generator

- And the output (Y) of the next state decoder is applied as the serial input to the shift register.
- The "Next state decoder" is designed by keeping in mind the required sequence.
- Design of a sequence generator using shift registers has been demonstrated in the following examples.

**Ex. 5.14.1 :** Design a sequence generator to generate the following sequence 1 0 1 1 0 ...

**OR**

Design pulse train generator using shift register to generate the following pulse :

.....10110.... **Dec. 12, 8 Marks, Dec. 13,**

**Dec. 14, Dec. 16, May 19, 6 Marks**

**Soln. :**

(C-6242) Table P. 5.14.1 : State stable for the sequence generator

Number of clock pulses	Flip flop outputs			Decimal equivalent
	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	
1	1	0	1	5 ← same
2	0	1	0	2
3	1	0	1	6 ←
4	1	1	0	6
5	0	1	1	3

↓  
Prescribed sequence  
delayed by 2 bits  
↓  
Prescribed sequence  
delayed by 1 bit  
↓  
Prescribed sequence

The length of given sequence = L = 5 bits.

1. The minimum number of flip flops (N) required to generate a sequence of length L is given by,
2. Length of the sequence L =  $2^N - 1$  .....(1)
3. Here L = 5. Therefore number of flip flops N = 3. Thus three flip-flops will have to be used.
4. Write the states of the circuit as shown in Table P. 5.14.1. Note that the column for Q<sub>2</sub> output

consists of the prescribed sequence, whereas the column for Q<sub>1</sub> consists of 1 bit delayed version of the prescribed sequence and that for Q<sub>0</sub> consists of 2 bit delayed version.

5. The use of N = 3 i.e. three flip-flops will lead us to the correct generation of prescribed sequence, if and only if all the states in the state table are distinct. Otherwise we will have to increase the number of flip-flops.
6. In Table P. 5.14.1, all the states are not distinct. States 1 and 3 are same. Hence only three flip-flops are not sufficient. So assume N = 4 and prepare a new state table as Table P. 5.14.1(A) in a similar manner as the previous state table.

(C-6243) Table P. 5.14.1(A) : Modified state table using N = 4

Clock pulse number (state)	Flip flop outputs				Decimal equivalent
	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	
1	1	0	1	1	11
2	0	1	0	1	5
3	1	0	1	0	10
4	1	1	0	1	13
5	0	1	1	0	6

- Note that with 4 flip-flop, all the states are distinct states.

#### 7. Design of next state decoder :

- We have to find the output (Y) of the next state decoder which is connected to the serial input of the shift register so that the state changes from present to next as per our requirements.
- For example if Q<sub>3</sub> Q<sub>2</sub> Q<sub>1</sub> Q<sub>0</sub> = 1 0 1 1 i.e. (11 decimal). Then what should be the serial input so as to get Q<sub>3</sub> Q<sub>2</sub> Q<sub>1</sub> Q<sub>0</sub> = 0 1 0 1 (i.e. 5 decimal) ? The answer is, serial input should be "0" (right shifting is assumed).
- Applying same logic to Y which is output of the next state decoder and serial input to the shift register we find Y for each state as shown in Table P. 5.14.1(B).



(C-6244) Table P. 5.14.1(B) : Output Y of the next state decoder

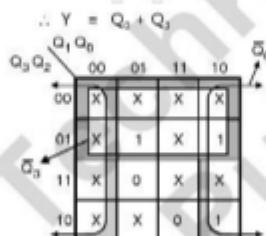
Number of clock pulse (state)	Flip flop outputs				Output Y	State (Decimal equivalent)
	$Q_3$	$Q_2$	$Q_1$	$Q_0$		
1	1	0	1	0	0	11
2	0	1	0	0	1	5
3	1	0	1	0	1	10
4	1	1	0	1	0	13
5	0	1	1	0	1	6

Valid states

Note that the Y output of the next state decoder is applied to  $D_3$  (input of FF - 3) and the shift register is operated in the right shift mode.

#### 8. K-map and simplifications :

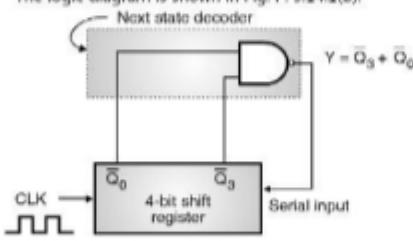
- The states 11, 5, 10, 13 and 6 shown in Table P. 5.14.1(B) are called as the stable states. So in the K-map corresponding to these states we will enter the corresponding value of Y (0 or 1 as per Table P. 5.14.1(B)).
- In the remaining boxes of K-maps we should enter the don't care (X) conditions.
- The K-map is shown in Fig. P. 5.14.1(a).



(C-1513) Fig. P. 5.14.1(a) : K-map and simplifications

#### 9. Draw the logic diagram :

- The logic diagram is shown in Fig. P. 5.14.1(b).



(C-1514) Fig. P. 5.14.1(b) : Logic diagram of the sequence generator

Ex. 5.14.2 : Design a sequence generator to generate the following pulse train using shift register

1 1 0 0 1 1 1 0.

Soln. :

#### Step 1 : Decide the number of flip-flops :

- The length of given sequence L = 8

$$L = 2^N - 1 \text{ where } N = \text{Number of flip-flops}$$

$$\therefore 8 = 2^N - 1$$

$$\therefore 2^N = 9 \quad \therefore N = 4$$

#### Step 2 : Write the state table for the sequence generator :

- The state table for the sequence generator is shown in Table P. 5.14.2(a).

(C-6245) Table P. 5.14.2(a) : State table for the generator

State	D	C	B	A	Decimal equivalent
1	1	0	0	0	12
2	1	0	0	1	9
3	0	0	1	1	8
4	0	1	1	1	7
5	1	1	1	0	14
6	1	1	0	1	13
7	1	0	1	1	11
8	0	1	1	0	6

- Note that we have delayed the sequence in the opposite direction. Hence the directions of all the arrows have reversed.

- All the states in Table P. 5.14.2(a) are distinct states. Hence 4 flip-flops will be sufficient.

#### Step 3 : Find Y :

- Refer Table P. 5.14.2(b) which shows the values of Y i.e. the output of the next state generator.

(C-6246) Table P. 5.14.2(b) : State table to obtain Y

State	D	C	B	A	Y
1	1	1	0	0	1
2	1	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	0
5	1	1	1	0	1
6	1	1	0	1	1
7	1	0	1	1	0
8	0	1	1	0	0

**Step 4 : K-map and simplifications :**

DC	BA	00	01	11	10	DC
00	X X	X	X	X	X	DC
01	X X	X	0	0		DC
11	1 1	X	X	1		
10	X 1	0	X			

(C-151) Fig. P. 5.14.2(a) : K-map and simplification

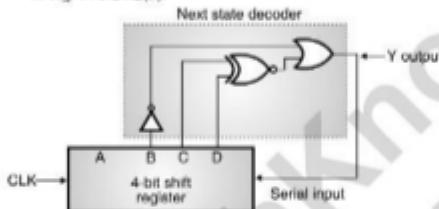
- The K-map for output Y is as shown in Fig. P. 5.14.2(a).

$$\therefore Y = \bar{B} + DC + \bar{D}\bar{C}$$

$$\therefore Y = \bar{B} + (D \oplus C)$$

**Step 5 : Draw the logic diagram :**

- The logic diagram of the sequence generator is shown in Fig. P. 5.14.2(b).



(C-1516) Fig. P. 5.14.2(b) : Logic diagram of the required sequence generator

**Ex. 5.14.3 :** Design and implement the following sequence generator using shift register :  
1010... Dec. 06, 4 Marks

**Soln. :**

- The length of the sequence L = 2 bits.

$$L = 2^N - 1$$

$$\therefore 2^N = 3$$

$$\therefore N = 2 \text{ so use three flip flops.}$$

- Write the states of the circuit as shown in Table P. 5.14.3(a).

(C-6299) Table P. 5.14.3(a)

Clock pulse	FF outputs		Decimal equivalent
	Q <sub>1</sub>	Q <sub>0</sub>	
1	1	0	2
2	0	1	1

**Design of next state decoder :**

- Refer Table P. 5.14.3(b) which shows the value of Y i.e. output of the next state generator.
- Table P. 5.14.3(b) state table to obtain Y.

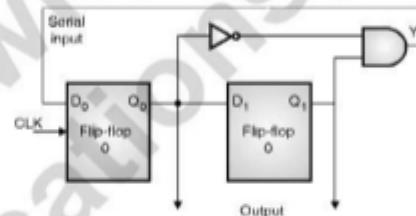
Table P. 5.14.3(b)

State	Q <sub>1</sub>	Q <sub>0</sub>	Y
1	1	0	1
2	0	1	0

$$\therefore \text{Output } Y = Q_1 \bar{Q}_0$$

**Logic diagram :**

- The sequence generator is as shown in Fig. P. 5.14.3.



(C-3560) Fig. P. 5.14.3 : Logic diagram of sequence generator

**Ex. 5.14.4 :** Design a pulse train generator using a shift register for the following pulse train :  
.....1000 110... May 12, 10 Marks

**Soln. :****Step 1 : Decide number of flip flop :**

Given sequence = ...1000110...

The length of given sequence L = 7

$$\therefore L = 2^N - 1 \text{ where } N = \text{Number of flip flops}$$

$$\therefore 7 = 2^N - 1$$

$$\therefore 2^N = 8 \therefore N = 3$$

**Step 2 : Write the state table for pulse train generator :**

- The state table for sequence generator is as shown in Table 5.14.4(a).



(C-3861) Table 5.14.4(a)

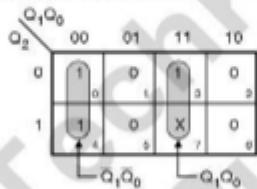
Number of clock pulse	Flip-flop outputs			Decimal equivalent
	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	
1	1	0	1	5
2	0	1	0	2
3	0	0	1	1
4	0	0	0	0
5	1	0	0	4
6	1	1	0	6
7	0	1	1	3

Step 3 : Find output Y :

(C-3864) Table P. 5.14.4(b)

State	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Y
1	1	0	1	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	1

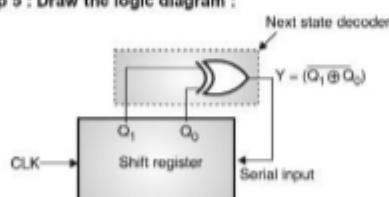
Step 4 : K-map and simplification :



(C-3862) Fig. P. 5.14.4(a)

$$\therefore Y = Q_1 Q_0 + \bar{Q}_1 \bar{Q}_0 = (Q_1 \oplus Q_0)$$

Step 5 : Draw the logic diagram :



(C-3863) Fig. P. 5.14.4(b) : Logic diagram

**Ex. 5.14.5 :** Design a sequence generator using shift register to generate sequence 1101.

**Dec. 07, 4 Marks****Soln. :** Similar to Ex. 5.14.3

**Ex. 5.14.6 :** Design a sequence generator to generate the following sequence : -1101011- using shift register.

**(May 08, 8 Marks)****Soln. :** Similar to Ex. 5.14.3

### Review Questions

- Q. 1 What is the function of a shift register ? Give its applications.
- Q. 2 State the types of shift registers.
- Q. 3 With a neat diagram explain the operation of 4-bit left shift register. Give its truth table and timing diagram.
- Q. 4 With a neat diagram explain the operation of 4-bit SISO (Serial-In-Serial-Out) register. Draw the timing diagram and give its truth table.
- Q. 5 With a neat diagram explain the operation of 4-bit Serial-In-Parallel Out (SIPO) register. Give the truth table and timing diagram.
- Q. 6 With a neat diagram explain the operation of 4-bit Parallel-In-Serial-Out (PISO) register. Give the truth table and timing waveforms.
- Q. 7 What is meant by "Universal shift register" ?
- Q. 8 What do you understand by a bi-directional shift register ? Explain its operation.
- Q. 9 List any one shift register IC. Sketch its schematic diagram and pin configuration. Give its specifications.



Q. 10 Draw circuit diagram of 3 bit SIPO shift register. Use D flip flops. Explain its working.

Q. 11 Define bi-directional shift register. Draw 3 bit bi-directional shift register using D flip flop.

■ ■ ■

TechKnowledge  
Publications

---

**Unit IV**

Chapter

**6****Counters****Syllabus**

Ripple counters, MOD-N counters, Up/down counters, Synchronous counters, Lock out, Sequence generators.

**Chapter Contents**

6.1 Introduction	6.8 Modulo – N Synchronous Counters
6.2 2-Bit Asynchronous / Ripple Up Counters	6.9 UP / DOWN Synchronous Counter
6.3 Asynchronous Down Counters	6.10 Lock Out Condition
6.4 UP / DOWN Counters	6.11 Bush Diagram
6.5 Modulus of the Counter (MOD-N Counter)	6.12 Applications of Counters
6.6 Problems Faced by Ripple Counters	6.13 Sequence Generator
6.7 Synchronous Counters	



## 6.1 Introduction :

### Definition :

- The digital circuit used for counting pulses is known as counter. It is a sequential circuit.
- Counter is the most widely used application of flip-flops. It is a group of flip-flops with a clock signal applied.
- Counter counts the number of clock pulses. Therefore with some modifications we can use them for measuring frequency or time period.

### 6.1.1 Types of Counters :

- Counters are basically of two types :
  1. Asynchronous or ripple counters.
  2. Synchronous counters.

#### 1. Asynchronous or ripple counters :

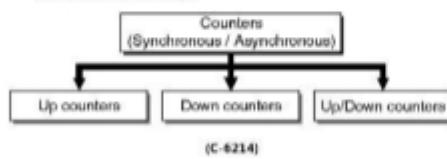
- For these counters the external clock signal is applied to one flip flop and then the output of preceding flip-flop is connected to the clock of next flip-flop.

#### 2. Synchronous counters :

- In synchronous counters all the flip-flops receive the external clock pulse is applied to all the flip-flops simultaneously.
- Ring counter and Johnson counter are the examples of synchronous counters.

### 6.1.2 Classification of Counters :

- Depending on the way in which the counter outputs change, the synchronous or asynchronous counters are classified as follows :



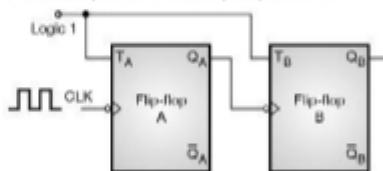
- **Up counters** are the counters that count from small to big count. Their output goes on increasing as they receive clock pulses.
- For example, the output of an up counter will be 0-1-2-3...

- **Down counters** are the counters that count from large to small count. Their output goes on decreasing as they receive clock pulses.
- For example, the output of a down counter will be 7-6-5-4-3...
- **Up/Down counter** is the combination of up counter and down counter.

## 6.2 2-Bit Asynchronous / Ripple Up Counters :

### Logic diagram :

- Fig. 6.2.1 shows the logic diagram of a 2-bit ripple up counter.
- The number of flip-flops used is 2. Note that the number of bits will always be equal to the number of flip-flops. Thus a 4 bit counter will use four flip-flops.
- The toggle (T) flip-flops are being used. But we can use the JK flip-flops also with J and K connected permanently to logic 1.
- External clock is applied to the clock input of flip-flop A which is the LSB flip-flop and  $Q_A$  output is applied to the clock input of the next flip-flop i.e. FF-B.



(C-771) Fig. 6.2.1 : A two bit asynchronous binary up counter

### Operation of the counter :

- Initially let both the flip-flops be in reset condition.  
 $\therefore Q_A = Q_B = 00$

### On the first negative going clock edge :

- As soon as the first falling edge of the clock hits FF-A, it will toggle as  $T_A = 1$ . Hence  $Q_A$  will become equal to 1.
- $Q_A$  is connected to clock input of FF-B. Since  $Q_A$  has changed from 0 to 1, it is treated as the positive clock edge by FF-B.
- There is no change in the status of  $Q_B$  because FF-B is a negative edge triggered FF.



- Therefore after the first clock pulse the counter outputs are

$$Q_B Q_A = 01 \quad \dots \text{After the first CLK pulse}$$

#### At the second falling edge of clock :

- On the arrival of second falling clock edge, FF-A toggles again and  $Q_A$  changes from 1 to 0.
- $\therefore Q_A = 0$  Corresponding to 2<sup>nd</sup> negative clock edge.
- This change in  $Q_A$  (from 1 to 0) acts as a negative clock edge for FF-B. So it will also toggle, and  $Q_B$  will change from 0 to 1.

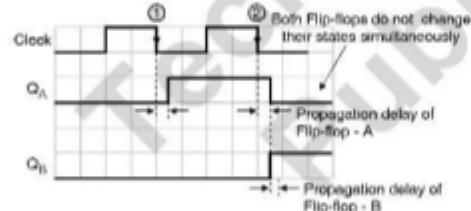
$$\therefore Q_B = 1$$

- Hence after the second clock pulse the counter outputs are

$$Q_B Q_A = 10 \quad \dots \text{After the second CLK pulse}$$

- Note that both the outputs are changing their state.
- But both the changes do not take place simultaneously.  $Q_A$  will change first from 1 to 0 and then  $Q_B$  will change from 0 to 1.
- This is due to the propagation delay of FF-A. So both flip-flops will never trigger at the same instant.

**Therefore the counter is called as an asynchronous counter.** This is shown in Fig. 6.2.2.



(C-772) Fig. 6.2.2 : FFs do not change their state simultaneously

#### At the third falling edge of clock :

- On arrival of the third falling edge, FF-A toggles again and  $Q_A$  becomes 1 from 0.
- Since this is a positive going change, FF-B does not respond to it and remains inactive. So  $Q_B$  does not change and continues to be equal to 1.

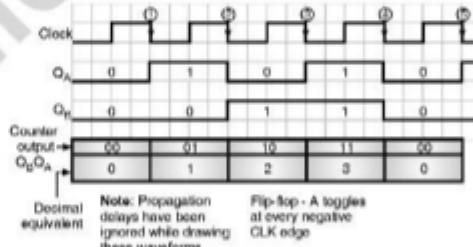
$$\therefore Q_B Q_A = 11 \quad \dots \text{After the third CLK pulse}$$

#### At the 4<sup>th</sup> negative clock edge :

- On the 4<sup>th</sup> falling clock edge, FF-A toggles and  $Q_A$  changes from 1 to 0.
- This negative going change in  $Q_A$  acts as a negative clock pulse for FF-B. Hence it toggles to change  $Q_B$  from 1 to 0.
- $\therefore Q_B Q_A = 00 \quad \dots \text{After the fourth CLK pulse}$
- So the counter has reached the original state. The counter operation will now repeat.
- Table 6.2.1 summarizes the operation of the counter and Fig. 6.2.3 shows the timing waveforms.

(C-6843) Table 6.2.1 : Summary of operation of a 2-bit binary ripple up counter

Clock	Counter outputs		State number	Decimal equivalent of counter output
	$Q_B$	$Q_A$		
Initially	0	0	-	0
1 <sup>st</sup> (↓)	0	1	1	1
2 <sup>nd</sup> (↓)	1	0	2	2
3 <sup>rd</sup> (↓)	1	1	3	3
4 <sup>th</sup> (↓)	0	0	4	0



(C-773) Fig. 6.2.3 : Timing diagram for a 2-bit ripple up counter

#### Why is it called counter ?

- See Fig. 6.2.3. The decimal count corresponds to the number of clock pulses, which counter has received.
- Thus this circuit counts the clock pulses. Hence it is called as counter.

#### Number of states :

- As seen from Table 6.2.1, this counter has four distinct states of output namely 00, 01, 10 and 11.



- In general the number of states =  $2^n$  where n is equal to the number of flip-flops.

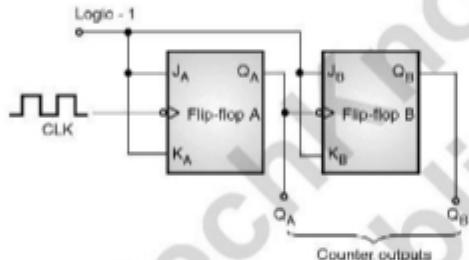
**Maximum count :**

- As seen from Table 6.2.1, the maximum count is 3 (decimal) i.e. 1 1 binary.  
Maximum count =  $3 = 2^2 - 1$
- In general the maximum count =  $(2^n - 1)$ , where n = Number of flip-flops.

### 6.2.1 Two Bit Asynchronous Up Counter using JK Flip-Flops :

**Logic diagram :**

- A 2 bit asynchronous counter up using JK flip-flops is shown in Fig. 6.2.4.
- Note that the J and K inputs of both the flip-flops are connected to logic 1 so actually JK flip-flops are converted into T flip-flops.



(C-774) Fig. 6.2.4 : Two bit ripple up counter using JK flip-flops

- The operation of this circuit is exactly same as that of the counter using the T flip-flops.

### 6.2.2 3 Bit Asynchronous Up Counter :

SPPU : May 07, Dec. 09, Dec. 12, Dec. 16

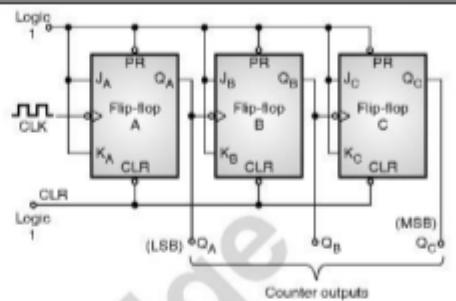
**University Questions**

- Q. 1** Draw 3-bit asynchronous counter. Explain timing diagram for the same.

(May 07, Dec. 09, Dec. 12, Dec. 16, 8 Marks)

**Logic diagram :**

- Fig. 6.2.5 shows the logic diagram of a 3-bit ripple up counter. Since it is a 3-bit counter, we need to use 3-flip-flops.



(C-775) Fig. 6.2.5 : 3-bit ripple up counter

- We can apply all the basic concepts which were introduced for the 2-bit ripple up counter to the 3-bit ripple up counter.
- Operation of the 3-bit ripple up counter takes place in exactly similar manner as that of a 2-bit counter.

**Truth table :**

- Table 6.2.2 summarizes the operation of the 3-bit asynchronous up counter.
- Note that the asynchronous preset and clear terminals are also being used.
- Both of them are active low inputs. Hence for the normal output of the counter preset and clear terminals should be connected to logic 1.

(C-776(a)) Table 6.2.2 : Summary of operation of a 3-bit ripple up counter

Clock	Flip-flop outputs			State	Decimal equivalent
	$Q_C$ (MSB)	$Q_B$	$Q_A$ (LSB)		
Initially	0	0	0	1	0
$2^{nd}$ (-)	0	0	1	2	1
$2^{nd}$ (-)	0	1	0	3	2
$3^{rd}$ (-)	0	1	1	4	3
$4^{th}$ (-)	1	0	0	5	4
$5^{th}$ (-)	1	0	1	6	5
$6^{th}$ (-)	1	1	0	7	6
$7^{th}$ (-)	1	1	1	8	7
$8^{th}$ (-)	0	0	0	1	0

**Number of states :**

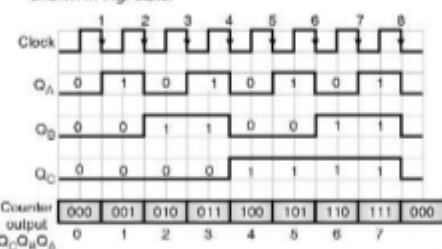
- Number of states =  $2^n = 2^3 = 8$ .
- The 3 bit ripple up counter can have 8 distinct states i.e.  $Q_C Q_B Q_A$  can take up values from 000, 001, 010, ....110, 111.

**Maximum count :**

- Maximum count =  $2^8 - 1 = 8 - 1 = 7$ . Refer Table 6.2.2 the maximum count is  $Q_C Q_B Q_A = 1\ 1\ 1$  i.e. decimal 7. Note that  $Q_C$  is treated as MSB and  $Q_A$  as LSB.

**Timing diagram :**

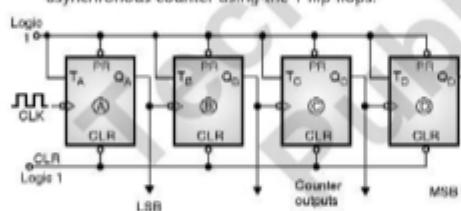
- The timing diagram of a 3-bit ripple up counter are as shown in Fig. 6.2.6.



(C-776) Fig. 6.2.6 : Timing diagram for a 3-bit ripple up counter

**6.2.3 4 Bit Asynchronous up Counter :****Logic diagram :**

- Fig. 6.2.7(a) shows the circuit diagram of a 4 bit asynchronous counter using the T flip flops.



(C-777) Fig. 6.2.7(a) : 4 bit asynchronous up counter

- Since it is a 4 bit ripple up counter, we need to use four flip flops.
- Initially all the flipflops have zero output.  
 $\therefore Q_D Q_C Q_B Q_A = 0000$ .
- All the flip flop are negative edge triggered. CLK signal is applied to the clock input of FF-A whereas Q outputs of every F/F is applied to the clock input of next F/F.

- For example  $Q_A$  to CLK of FF-B,  $Q_B$  to CLK of FF-C and so on.

**Truth table :**

- Table 6.2.3 shows the truth table for 4 bit asynchronous up counter.
- Its output passes through 16 states from 0000 i.e.  $(0)_10$  to  $(1111)_10$  i.e.  $(15)_10$ .
- After 1111, the output again become 0000 and the operation repeats itself.
- Fig. 6.2.7(b) shows the timing diagram for the 4-bit asynchronous up counter.
- $Q_D$  acts as MSB of the output whereas  $Q_A$  acts as the LSB.

(C-6844) Table 6.2.3 : Truth table for a 4-bit asynchronous up counter

Clock	FF outputs				Decimal
	$Q_D$	$Q_C$	$Q_B$	$Q_A$	
Initially	0	0	0	0	0
$1^{\text{st}} (\downarrow)$	0	0	0	1	1
$2^{\text{nd}} (\downarrow)$	0	0	1	0	2
$3^{\text{rd}} (\downarrow)$	0	0	1	1	3
:	:	:	:	:	:
$14^{\text{(th)}} (\downarrow)$	1	1	1	0	14
$15^{\text{(th)}} (\downarrow)$	1	1	1	1	15
$16^{\text{(th)}} (\downarrow)$	0	0	0	0	0

**Number of states :**

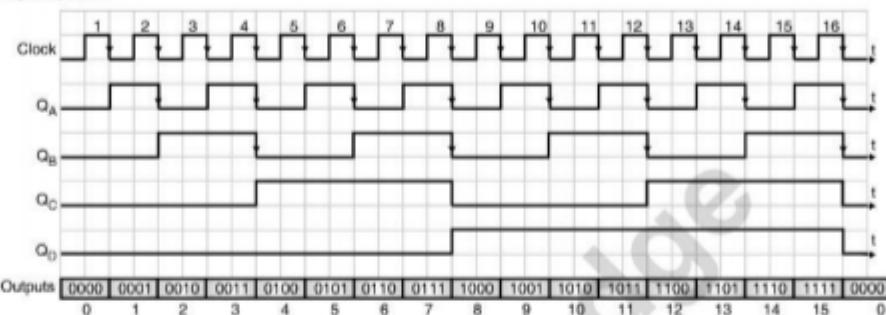
- The number of state through which the output of a 4 bit up counter passes is 16 (from 0 to 15).

**Maximum count :**

- The maximum count is 15 i.e. 1111. Thus a 4-bit ripple up counter will count from 0000 to 1111.



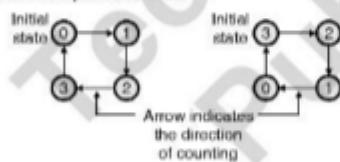
Timing diagram :



(C-778) Fig. 6.2.7(b) : Waveforms of a 4 bit asynchronous up counter

#### 6.2.4 State Diagram of a Counter :

- The state diagram of a counter represents the states of a counter graphically.
- For example for a 2-bit up counter the state diagram is shown in Fig. 6.2.8(a) and for a 2-bit down counter the state diagram is shown in Fig. 6.2.8(b).
- The number written inside a circle represents the state number, whereas the arrow shows the direction of counter (up or down).
- Note that in the counters only the state is important. Hence in the state diagram we have not shown any input or output conditions.



(a) For a 2-bit up counter (b) For a 2-bit down counter

(C-779) Fig. 6.2.8 : State diagram

#### 6.3 Asynchronous Down Counters :

##### 6.3.1 3- Bit Asynchronous Down Counter :

Truth table and state diagram :

- All the counters discussed so far have counted upwards from zero. So they can be called as **up counters**.
- But the counters which can count in the **downward** direction i.e. from the maximum count to zero are called **down counters**.

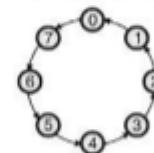
- The countdown sequence for a 3-bit asynchronous down counter is as follows :

(C-697) Table 6.3.1 : Truth table of 3 bit down counter

CLK	Output			Decimal Count
	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	
↓	0	0	0	0
	1	1	1	7
	1	1	0	6
	1	0	1	5
	1	0	0	4
	0	1	1	3
	0	1	0	2
	0	0	1	1
↓				0

Maximum count  
Direction of counting

Repeats



(C-2762) Fig. 6.3.1 : State diagram

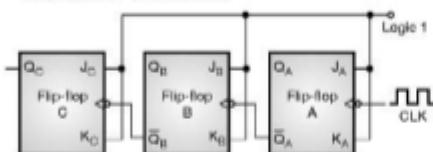
- Thus counting takes place as follows :  
 $Q_C\ Q_B\ Q_A = 111, 110, 101, 100, 011, 010, 001, 000$ .
- From this sequence it is evident that FF-A should toggle at every negative going clock edge but FF-B should change its state only at those instants when  $Q_A$  changes from LOW (0) to HIGH (1) and FF-C should change its state only when  $Q_B$  changes from LOW to HIGH.
- Thus in a down counter, each FF except the first one (FF-A) should toggle when the output of its preceding flip-flop changes from LOW to HIGH.



- If all the FFs are negative edge triggered i.e. responding to the negative CLK edge, then we can place an inverter in front of every CLK input or we can drive the CLK input of next FF from the  $\bar{Q}$  output of the preceding FF and not from the Q outputs as shown in Fig. 6.3.2.

#### Logic diagram :

- A 3-bit asynchronous down counter is shown in Fig. 6.3.1. The clock input is applied directly to the clock input of FF-A. But  $\bar{Q}_A$  is connected to clock of FF-B,  $\bar{Q}_B$  to clock of FF-C and so on.



(C-781) Fig. 6.3.2 : A 3-bit asynchronous down counter

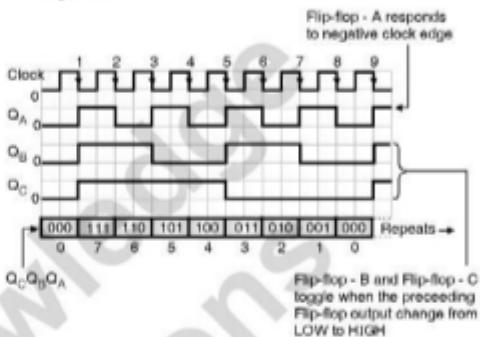
#### Operation :

- Initially let all the flip-flops be in the reset condition.  
 $\therefore Q_C Q_B Q_A = 0\ 0\ 0$
- As soon as the first falling clock pulse arrives, FF-A toggles. So  $Q_A$  becomes 1 and  $\bar{Q}_A$  changes from 1 to 0.
- The negative going change in  $Q_A$  acts as a clock to FF-B. Hence FF-B will change its state. So  $Q_B$  becomes 1 and  $\bar{Q}_B$  changes from 1 to 0.
- This negative going change in  $\bar{Q}_B$  acts as a clock to FF-C. Hence FF-C will change its state. So  $Q_C$  becomes 1 and  $\bar{Q}_C$  becomes 0.
- Thus after the first clock pulse the output of counter are,  
 $Q_C Q_B Q_A = 1\ 1\ 1 \quad \dots \text{After the } 1^{\text{st}} \text{ CLK pulse}$
- Corresponding to the second falling clock edge, FF-A toggles.  $Q_A$  becomes 0 and  $\bar{Q}_A$  becomes 1. This positive going change in  $\bar{Q}_A$  does not alter the state of FF-B. So  $Q_B$  remains 1 and  $\bar{Q}_B$  remains 0. So there is no change in the state of FF-C. Hence after the second clock pulse the counter outputs are,  
 $Q_C Q_B Q_A = 1\ 1\ 0 \quad \dots \text{After the } 2^{\text{nd}} \text{ CLK pulse}$
- The down counting will thus take place. Similarly the counter will count down to pass through the states 101,

100, 011, 010, 001 and 000. The operation repeats itself thereafter.

#### Timing diagram :

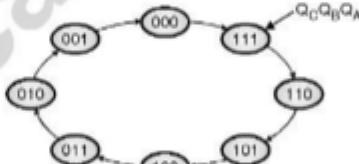
- The timing diagram for the down counter is shown in Fig. 6.3.3.



(C-782) Fig. 6.3.3 : Timing diagram of a 3-bit down counter

#### State diagram :

- The state diagram of the 3-bit down counter is shown in Fig. 6.3.4.



(C-783) Fig. 6.3.4 : State diagram of a 3 bit ripple down counter

**Note :** The numbers inside the circles correspond to the state of  $Q_C Q_B Q_A$ .

#### 6.4 UP / DOWN Counters :

- We have designed the up counters and the down counters separately.
- But in practice both these modes are generally combined together and an UP/DOWN counter is formed.
- A mode control (M) input is also provided to select either up count or down count mode of operation.
- A combinational circuit is required to be designed and used between each pair of flip-flops in order to achieve the up/down operation.

**Types of up/down counters :**

- The up/down counters are of two types :
  1. UP/DOWN ripple counters.
  2. UP/DOWN synchronous counters.

**6.4.1 UP/DOWN Ripple Counters :**

- In the up/down ripple counter all the FFs operate in the toggle mode. So either T flip-flops or JK flip-flops are to be used.
- The LSB flip-flop receives clock directly. But the clock to every other FF is obtained from Q or  $\bar{Q}$  output of the previous FF.

**UP counting mode ( $M = 0$ ) :**

- The CLK signal is applied directly to the clock input of the LSB flip-flop.
- For the remaining flip-flops, the Q output of the preceding FF is connected to the clock of the next stage if up counting is to be achieved. For this mode, the mode select input M is at logic 0 ( $M = 0$ ).

**DOWN counting mode ( $M = 1$ ) :**

- The clock signal is applied directly to the clock input of the LSB flip-flop. For the remaining flip-flop, the  $\bar{Q}$  output of the preceding FF is connected to the clock of the next FF.
- This will operate the counter in the down counting mode. For down counting mode the mode select input M is kept at logic 1 ( $M = 1$ ).

**6.4.2 3-bit Up Down Ripple Counters :**

- The design of 3-bit up down ripple counters has been illustrated in the following examples.

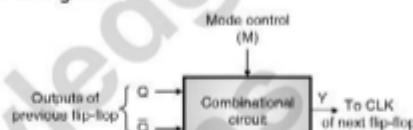
**Ex. 6.4.1 :** Design a 3-bit binary up/down ripple counter.  
Draw the timing diagram.

**May 07, Dec. 08, 6 Marks**

**Soln. :**

- The requirements of counter are :
  1. 3-bit : Hence three FFs are required.
  2. UP/DOWN : So a mode control input is essential.
- We know that for a ripple up counter, the Q output of preceding FF is connected to the clock input of the next one.

- And for a ripple down counter, the  $\bar{Q}$  output of the preceding FF is connected to the clock input of the next one.
- Let the selection of Q or  $\bar{Q}$  output of the preceding FF be controlled by the mode control input M such that,
  - If  $M = 0$  UP counting. So connect Q to CLK
  - If  $M = 1$  DOWN counting. So connect  $\bar{Q}$  to CLK
- Let us design a combinational logic to satisfy all the requirements stated above.

**Block diagram :**

(C-786) Fig. P. 6.4.1(a) : Block diagram of combinational circuit

**Truth table of combinational circuit :**

- The truth table of such a combinational circuit is shown in Table P. 6.4.1.

(C-6215) Table P. 6.4.1 : Truth table

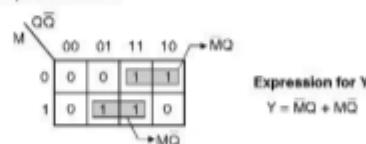
Inputs			Output
M	Q	$\bar{Q}$	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$Y = Q$   
for up  
counting

$Y = \bar{Q}$   
for down  
counting

**K-map and simplified expression for output :**

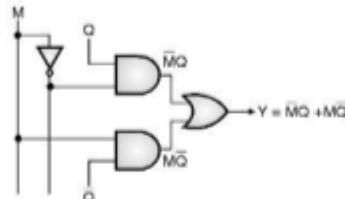
- Fig. P. 6.4.1(b) shows the K-map and simplified expression for Y.



(C-787) Fig. P. 6.4.1(b) : K-map for Y

**Logic diagram for combinational circuit :**

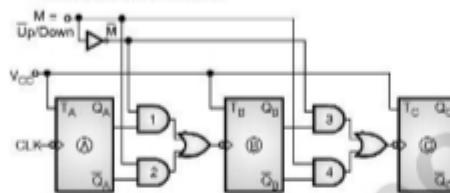
- The logic diagram for the combinational circuit is shown in Fig. P. 6.4.1(c).



(C-787) Fig. P. 6.4.1(c)

**Logic diagram of a 3-bit up/down counter :**

- The combinational circuit is connected between every pair of flip-flops to obtain the 3-bit up/down counter as shown in Fig. P. 6.4.1(d).



(C-788) Fig. P. 6.4.1(d) : A 3-bit up / down ripple counter

**Operation of 3-bit up down ripple counter :****1. With  $M = 0$  (Up counting mode) :**

- If  $M = 0$  and  $\bar{M} = 1$ , then the AND gates 1 and 3 in Fig. P. 6.4.1(d) will be enabled whereas the AND gates 2 and 4 will be disabled.
- Hence  $Q_A$  gets connected to the clock input of FF-B and  $Q_B$  gets connected to the clock input of FF-C.
- These connections are same as those for the normal up counter.
- Thus with  $M = 0$  the circuit works as an up counter.

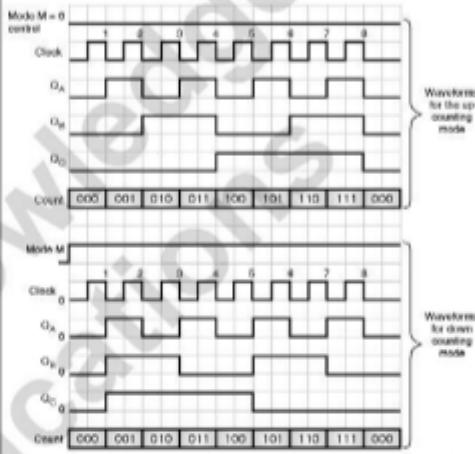
**2. With  $M = 1$  (Down counting mode) :**

- If  $M = 1$ , then AND gates 2 and 4 in Fig. P. 6.4.1(d) are enabled whereas the AND gates 1 and 3 are disabled.

- Hence  $\bar{Q}_A$  gets connected to the clock input of FF-B and  $\bar{Q}_B$  gets connected to the clock input of FF-C. As discussed earlier, these connections will produce a down counter. Thus with  $M = 1$  the circuit works as a down counter.

**Timing diagram for a 3-bit up/down ripple counter :**

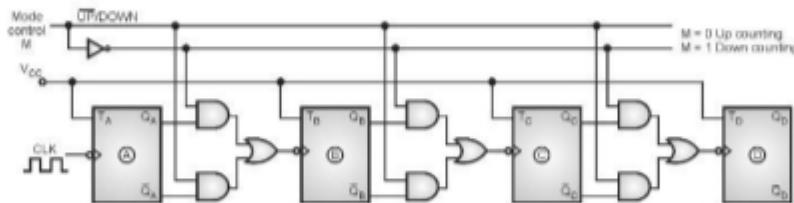
- The timing diagram for a 3-bit up/down ripple counter is shown in Fig. P. 6.4.1(e).



(C-2076(e)) Fig. P. 6.4.1(e) : Timing diagram for 3-bit up/down ripple counter

**Ex. 6.4.2 :** Design a 4-bit up down ripple counter.**Dec. 04, Dec. 12, 8 Marks****Soln. :**

- Refer Ex. 6.4.1. The combinational circuit is going to be the same. But we have to use 4 flip-flops. The 4-bit up down counter is as shown in Fig. P. 6.4.2.
- The circuit will work as up counter with  $M = 0$  whereas it will operate as a down counter when  $M = 1$ .



(C-789) Fig. P. 6.4.2 : 4-bit up/down ripple counter



## 6.5 Modulus of the Counter (MOD-N Counter) :

**Definition :**

- Modulus (MOD) of a counter represents the number of states through which the counter progresses during its operation. It is denoted by N.
  - Thus MOD-N counter means the counter progresses through N states.
  - A MOD-4 counter will have 4 states. A MOD-6 counter will have 6 states etc.
  - Thus a 3 bit counter which has 8 states is a MOD-8 counter, and a 4 bit counter is a MOD-16 counter.
  - In general m number of flip-flops are required to construct mod-n counter, where  $N \leq 2^m$ .
  - We can design a modulo counter with the help of the basic ripple counter structure and a combinational logic called reset logic.
  - 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 =  $2^n$
- So we can conclude that **modulus** of a counter represents the **number of states** through which the counter progresses during its operation.
  - Can we have a modulo-5 counter using a 3-bit ripple counter ? That means can we restrict the number of states to only 5 instead of 8 under normal conditions ?
  - The answer is yes. We can design such modulo counters with the help of the basic ripple counter structure and a combinational logic called reset logic.
  - Table 6.5.1 shows the relation between 2, 3 and 4 bit counters and their modulus.

(c-8049) Table 6.5.1

Counter type	Modulus
2 bit up or down	MOD-4
3 bit up or down	MOD-8
4 bit up or down	MOD-16

### 6.5.1 Design of Asynchronous MOD Counters :

- Ex. 6.5.1 :** Design a Modulo-5 ripple counter using a 3-bit ripple counter.

**OR**

Design MOD-5 asynchronous counter, and also draw the waveforms.

**Dec. 11, Dec. 14, 6 Marks**

**Soln. :**

**Step 1 : Draw the state diagram :**

- The state diagram of MOD-5 ripple counter is as shown in Fig. P. 6.5.1(a).



(C-794) Fig. P. 6.5.1(a) : State diagram of a MOD-5 ripple counter

**Step 2 : Write truth table for the reset logic :**

- Table P. 6.5.1 shows the truth table for the reset logic.

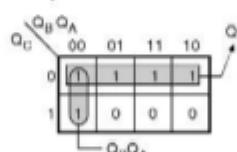
(C-6216) Table P. 6.5.1 : Truth table for the reset logic

State	Flip-flop outputs			Output Y of reset logic
	$Q_C$	$Q_B$	$Q_A$	
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

> Valid states  
> Invalid states

**Step 3 : K-map :**

For output Y



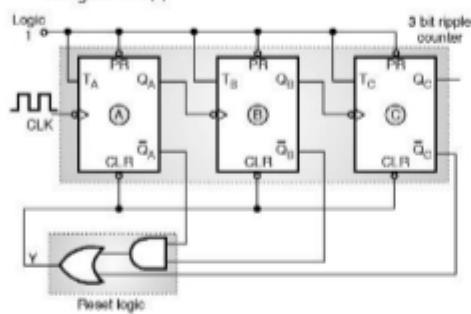
Expression for Y  
 $Y = \bar{Q}_C + Q_B \bar{Q}_A$

(C-795) Fig. P. 6.5.1(b) : K-map and simplification

- The K map is as shown in Fig. P. 6.5.1(b).
- The states 0 through 4 are valid states and the output Y of reset logic (Y) is inactive (0) for them.
- The states 5, 6 and 7 are invalid states. If counter enters into any one of these states that  $Y = 0$  (active) and will reset all the flip-flops.

#### **Step 4 : Logic diagram :**

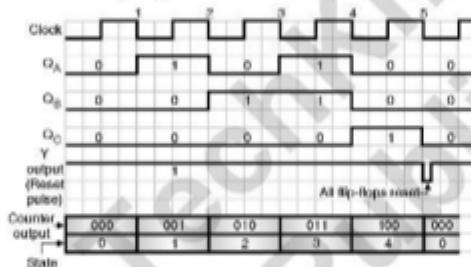
- The logic diagram of a MOD-5 ripple counter is shown in Fig. P. 6.5.1(c).



(c-796) Fig. P. 6.5.1(c) : Logic diagram of  
MOD-5 ripple counter

#### **Step 5 : Timing diagram :**

- The timing diagram is as shown in Fig. P. 6.5.1(d).



(C-797) Fig. P. 6.5.1(d) : Timing diagram of MOD-5 ripple counter

**Ex. 6.5.2 :** For MOD-11 asynchronous up counter :

1. Draw circuit diagram. Use T flip-flop.
  2. Write truth table.
  3. Draw timing diagram.
  4. If the output frequency is 11 kHz what is the clock input ?

Solv.

**Step 1 : Write the truth table :**

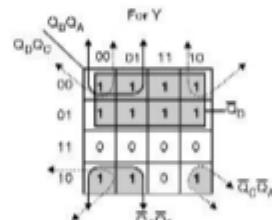
(C-6268) Table P. 6.5.2

$Q_3$	$Q_2$	$Q_1$	$Q_0$	Y output
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

#### **Step 2 : Draw the K map :**

- The K map is shown in Fig. P. 6.5.2(a).
  - The simplified equation for Y output of the reset logic is as follows :

$$Y = \bar{Q}_D + \bar{Q}_C (\bar{Q}_A + \bar{Q}_B)$$

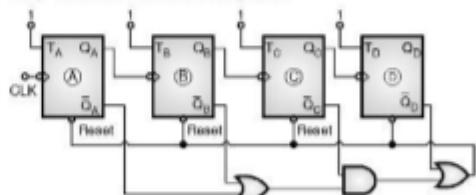


$$\therefore Y = \bar{Q}_D + \bar{Q}_C \bar{Q}_A + \bar{Q}_C \bar{Q}_B$$

$$\therefore Y = \bar{Q}_D + \bar{Q}_C(\bar{Q}_A + \bar{Q}_B)$$

(C-801) Fig. P. 6.5.2(a) : K map

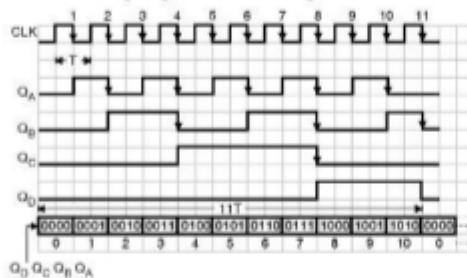
### **Step 3 : Draw the circuit diagram :**



(C-802) Fig. P. 6.5.2(b) : MOD 11 counter

**Step 4 : Timing diagram :**

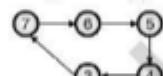
- The timing diagram is shown in Fig. P. 6.5.2(c).



(C-803) Fig. P. 6.5.2(c) : Timing diagram

- Output frequency  $f_o = \frac{f_{CLK}}{11}$
- $\therefore f_{CLK} = 11 f_o = 11 \times 11 \text{ kHz}$
- $\therefore f_{CLK} = 121 \text{ kHz}$  ...Ans.

**Ex. 6.5.3 :** Design a ripple counter for the state diagram shown in Fig. P. 6.5.3(a).



(C-804) Fig. P. 6.5.3(a) : Given state diagram

**Soln. :**

The observations from the given state diagram are :

- The highest state is 7 and the counter is a down counter.
- 7, 6, 5, 4, 3 are the valid states.
- 0, 1, 2 are the invalid states.
- As soon as the count reaches 3 i.e. 011, all the flip-flops should be set and counter should return back to state 7.

**Note :** In order to make the counter to DOWN COUNT, we must treat  $\bar{Q}_C$ ,  $\bar{Q}_B$  and  $\bar{Q}_A$  as outputs for deciding the state.

**Step 1 : Decide the number of flip-flops (n) :**

- Since the highest state is 7. The number of flip-flops (n) is given by,

$$\text{Highest state} = 2^n - 1$$

$$\therefore 8 \leq 2^n \therefore n = 3$$

- So 3 flip-flops should be used.

**Step 2 : Truth table for the reset logic :**

Truth table for the reset logic is shown in Table P. 6.5.3.

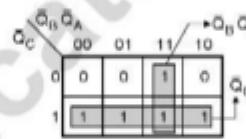
(C-6277) Table P. 6.5.3 : Truth table of reset logic

State	Flip-flop outputs			Output of reset logic Y
	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	
Invalid states	0	0	0	0
	1	0	0	0
	2	0	1	0
	3	0	1	1
Valid states	4	1	0	0
	5	1	0	1
	6	1	1	0
	7	1	1	1

**Note :** The  $\bar{Q}_C$ ,  $\bar{Q}_B$ ,  $\bar{Q}_A$  outputs are used because the counter is a DOWN counter.

**Step 3 : K-map and simplification for output Y :**

For output Y

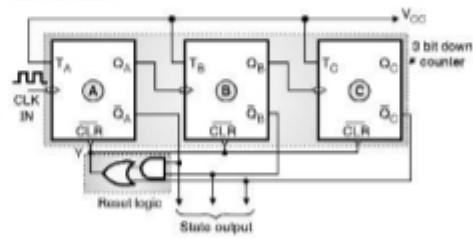


Expression for Y  
 $Y = \bar{Q}_C + \bar{Q}_B \bar{Q}_A$

(C-805) Fig. P. 6.5.3(b) : K-map and simplification for Y output

**Step 4 : Draw the logic diagram :**

The logic diagram of the required counter is shown in Fig. P. 6.5.3(c).



(C-806) Fig. P. 6.5.3(c) : Logic diagram of required counter

**Ex. 6.5.4 :** A certain counter is being pulsed by a 256 kHz clock signal. The output frequency from the last flip-flop is 2 kHz : 1. Determine the MOD number. 2. Determine the counting range.

**Dec. 05, 2 Marks**

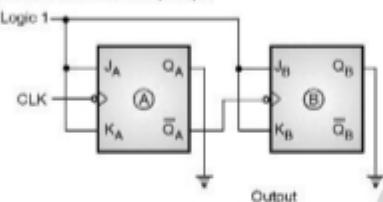
**Soln. :**

- MOD number ( $n$ ) =  $\frac{\text{Clock frequency}}{\text{Output frequency}} = \frac{256 \text{ kHz}}{2 \text{ kHz}} = 128$ .
- Counting range = 0 to  $(n - 1) = 0$  to  $(128 - 1) = 0$  to 127.

**Ex. 6.5.5 :** Design MOD-4 down-counter using JK flip-flop.  
**May 07, 6 Marks**

**Soln. :**

- Fig. P. 6.5.5 shows the MOD-4 down counter and Table P. 6.5.5 shows its truth table.
- For MOD-4 counter, the number of states is 4 so we have to use two JK flip flops.



(C-1423(a)) Fig. P. 6.5.5 : MOD 4 down counter

(C-8255) Table P. 6.5.5 : Truth table

CLK	$Q_A$	$\bar{Q}_A$
Initially	0	0
↓	1	1
↓	1	0
↓	0	1
↓	0	0

**Ex. 6.5.6 :** Design MOD 6 asynchronous counter using T flip-flop and also draw the waveforms.  
**May 12, 8 Marks, May 15, 6 Marks**

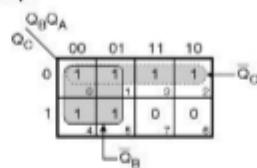
**Soln. :****Step 1 : Truth table :**

- The truth table of divide by 6 (MOD 6) asynchronous counter is as shown in Table P. 6.5.6(a).
- Y represents output of reset logic used for resetting the flip-flops.

(C-6297) Table P. 6.5.6(a)

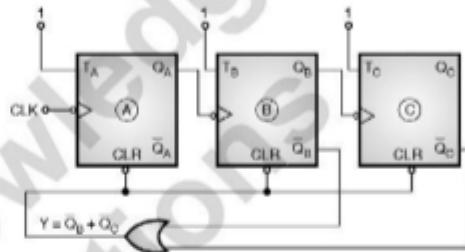
CLK	$Q_C$	$Q_B$	$Q_A$	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Invalid states  
(Reset all FFs)

**Step 2 : K-map :**

$$\therefore Y = Q_B + Q_C$$

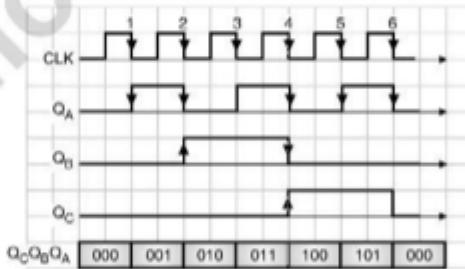
(C-3422) Fig. P. 6.5.6(a)

**Step 3 : Draw the circuit :**

(C-3423) Fig. P. 6.5.6(b) : Logic diagram

**Step 4 : Timing diagram :**

The timing diagram is as shown in Fig. P. 6.5.6(c).



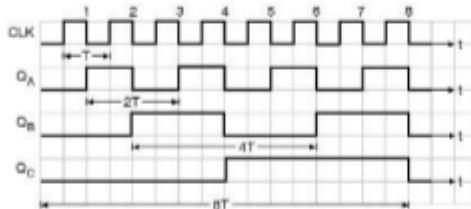
(C-3424) Fig. P. 6.5.6(c) : Timing diagram

**6.5.2 Frequency Division Taking Place in Asynchronous Counters :**

- In the chapter on flip-flops, we have seen that a flip-flop in toggle mode divides the clock frequency by 2.
- That means the frequency of Q or  $\bar{Q}$  output waveform of a toggle flip-flop is exactly half of the clock frequency.
- The concept of frequency division is applicable to the counters as well because we use flip-flops in the toggle mode for counters.



- Refer to the timing waveforms of a 3-bit asynchronous up counter and observe the frequencies of  $Q_A$ ,  $Q_B$  and  $Q_C$  waveforms with respect to the clock frequency.



(c-807) Fig. 6.5.1 : Timing waveforms of a 3-bit asynchronous up counter

**Conclusions :**

- Let one cycle period of the clock signal be  $T$  sec. Hence the clock frequency  $f_{CLK} = (1/T)$  Hz.
- Output of the least significant flip-flop (i.e. FF-A) has a one cycle period of  $(2T)$  as shown in Fig. 6.5.1. Hence,

$$\text{Frequency of } Q_A \text{ output} = f_A = \frac{1}{2T} = \frac{f_{CLK}}{2}$$

- The one cycle period of  $Q_B$  is  $4T$ . Hence the frequency of  $Q_B$  is given by,

$$f_B = \frac{1}{4T} = \frac{f_{CLK}}{4} \quad \dots \text{since } \frac{1}{T} = f_{CLK}$$

- Similarly the frequency of  $Q_C$  output is given by

$$f_C = \frac{1}{8T} = \frac{f_{CLK}}{8}$$

**Note :** In any counter, the signal at the output of last FF (i.e. MSB) will have a frequency equal to the input clock frequency divided by the MOD number of the counter.

**6.5.3 Decade (BCD) Ripple Counter :**

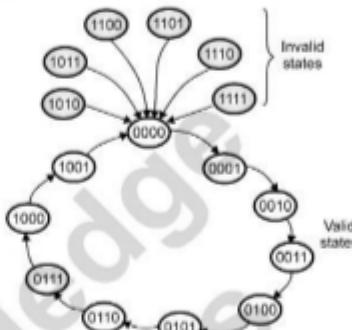
SPPU : Dec. 07, Dec. 09, May 10

**University Questions**

- Q. 1** Design BCD ripple counter. (Dec. 07, 4 Marks)  
**Q. 2** Design and implement MOD-10 ripple counter using J-K flip-flops and explain with output waveforms. (Dec. 09, 8 Marks)  
**Q. 3** Design and implement a mod-10 asynchronous counter using T flip-flops. (May 10, 8 Marks)

**State diagram :**

- The state diagram of an BCD ripple counter is shown in Fig. 6.5.2.



(c-811) Fig. 6.5.2 : State diagram of a BCD counter

- It shows that a BCD counter counts from 0000 to 1001 i.e. from 0 to 9.
- Four JK flip-flops with a clear input are to be used. The output of the **reset logic** is to be connected to the clear input of all the FFs.
- The output of the reset logic  $Y = 1$  for all the valid states from 0 to 9 and  $Y = 0$  for all the invalid outputs of the counter i.e. from 10 to 15.

If the counter enters into any invalid state then it will be reset automatically.

**Truth table :**

- The truth table of the reset logic is given in Table 6.5.2.

(c-6269) Table 6.5.2 : Truth table of a BCD counter

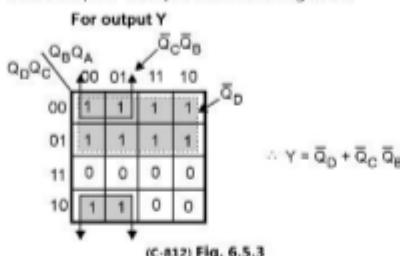
CLK	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Output of reset logic
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
-	1	0	1	0	0
-	1	0	1	1	0
-	1	1	0	0	0
-	1	1	0	1	0
-	1	1	1	0	0
-	1	1	1	1	0

&gt; Valid states

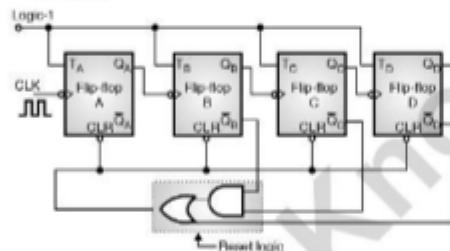
&gt; Invalid states

**K-map and simplifications :**

- The K-map for Y output is shown in Fig. 6.5.3.

**Logic diagram :**

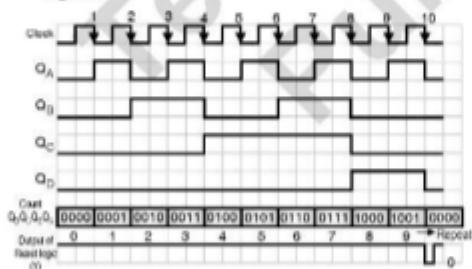
- The logic diagram of a BCD counter are as shown in Fig. 6.5.4.



(C-813) Fig. 6.5.4 : Logic diagram of a BCD counter

**Timing diagram :**

- The timing diagram of a BCD counter is shown in Fig. 6.5.5.



(C-814) Fig. 6.5.5 : Timing diagram of a BCD ripple counter

**6.6 Problems Faced by Ripple Counters :****SPPU : Dec. 05****University Questions**

- Q. 1** Explain using suitable waveforms, problems faced by ripple counter. **(Dec. 05, 6 Marks)**

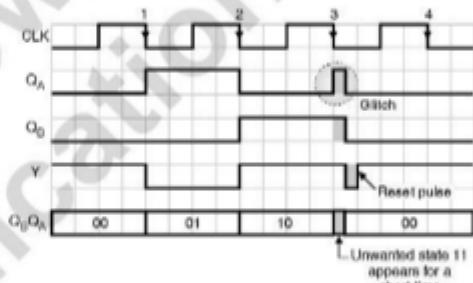
- The two major problems associated with the ripple counters are as follows :
  - Generation of unwanted short duration pulses called glitch.
  - Propagation delay.

**6.6.1 Glitch :****SPPU : Dec. 07, Dec. 11****University Questions**

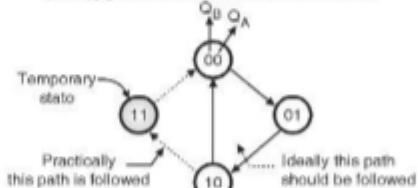
- Q. 1** Explain the problem of glitches in asynchronous counter circuits and solution for the same. **(Dec. 07, 4 Marks)**

- Q. 2** Mention significance of glitch. **(Dec. 11, 4 Marks)**

- Glitch is a short duration pulse or spike that appears in the outputs of a counter with mod- number  $< 2^n$ .
- Consider the waveforms of a MOD-3 counter shown in Fig. 6.6.1(a).



(C-809)(a) Waveforms of a MOD-3 counter

**(b) State diagram****(C-809) Fig. 6.6.1**

- The counter is supposed to pass through the states 00, 01 and 10 only and return to 00 state as shown in the state diagram of Fig. 6.6.1(b).
- But practically what happens is something different. At the third falling edge of CLK, the  $Q_3$  and  $Q_2$  become 11.
- Hence output of reset logic goes low but after its own propagation delay.



- Therefore the  $Q_B Q_A$  outputs are allowed to be 11 for a short period of time as shown in Fig. 6.6.1(a).
- Note that 11 is an unwanted state. In the  $Q_A$  output waveform a short pulse called "Glitch" will appear as shown in Fig. 6.6.1(a).

### 6.6.2 Disadvantages of Ripple Counters :

- Every flip-flop has its own propagation delay. In ripple counter the output of the previous FF is used as clock for the next FF.
- Hence the propagation delay goes on accumulating. For a 3-bit ripple counter the propagation delay of the first FF gets added to that of the second FF to decide the transition time for the third stage.
- This accumulated time delay is the main problem with the ripple counters, because the propagation delay goes on increasing with increase in number of flip-flops.
- This will put a limitation on the maximum clock frequency.

## 6.7 Synchronous Counters :

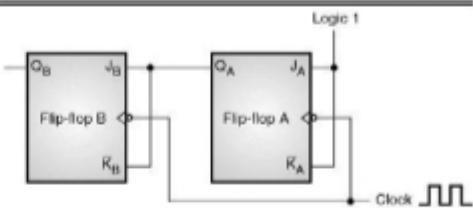
### Definition :

- If the "clock" pulses are applied to all the flip-flops connected in a counter simultaneously, then such a counter is called as synchronous counter.
- These counters are also known as parallel counters. The state of all the flip flops will change simultaneously in the synchronous counter.

### 6.7.1 2-Bit Synchronous up Counter :

#### Logic diagram :

- A 2-bit or MOD-4 synchronous counter is shown in Fig. 6.7.1.
- The  $J_A$  and  $K_A$  inputs of FF-A are tied to logic 1. So FF-A will work as a toggle flip-flop. The  $J_B$  and  $K_B$  inputs are connected to  $Q_A$ .
- Hence FF-B will toggle if  $Q_A = 1$  and there won't be any state change if  $Q_A = 0$ , at the instant when the negative clock edge is applied.



(C-815) Fig. 6.7.1 : A 2-bit (MOD-4) synchronous counter

### Operation :

- Initially let both the FFs be in the reset state. Let FF-A be the LSB flip-flop and FF-B be the MSB flip-flop.  
 $\therefore Q_B Q_A = 00 \dots$  Initially

#### At the 1<sup>st</sup> negative clock edge :

- As soon as the first negative clock edge is applied, FF-A will toggle and  $Q_A$  will change from 0 to 1.
- But at the instant of application of negative clock edge,  $Q_A = 0 \therefore J_B = K_B = 0$ . Therefore FF-B will not change its state. So  $Q_B$  will remain 0.  
 $\therefore Q_B Q_A = 01 \dots$  After the first clock pulse

#### At the 2<sup>nd</sup> negative clock edge :

- At the instant when we apply the second negative clock edge, FF-A toggles again and  $Q_A$  changes from 1 to 0.
- But at this instant  $Q_A$  was 1. So  $J_B = K_B = 1$  and FF-B also will toggle. Hence  $Q_B$  changes from 0 to 1.  
 $\therefore Q_B Q_A = 10 \dots$  After the second clock pulse

#### Next negative clock edges :

- Similarly on application of the third falling clock edge, FF-A will toggle from 0 to 1 but there is no change of state for FF-B.  
 $\therefore Q_B Q_A = 11 \dots$  After the third clock pulse
- On application of the next clock pulse,  $Q_A$  will change from 1 to 0 as  $Q_B$  will also change from 1 to 0. Hence  
 $\therefore Q_B Q_A = 00 \dots$  After the fourth clock pulse
- This is the original state. The operation of counter will repeat after this. The operation is summarised in Table 6.7.1 and the timing diagram is shown in Fig. 6.7.2.
- In this way the 2-bit synchronous counter has four distinct states namely  $Q_B Q_A = 00, 01, 10$  and  $11$ .
- The maximum count of a 2-bit counter is  $(11)_2$  or  $(3)_{10}$ .



(C-6224) Table 6.7.1 : Summary of operation of a 2-bit synchronous counter

Clock	Counter outputs	
	$Q_B$ (MSB)	$Q_A$ (LSB)
Initially	0	0
1 <sup>st</sup> (↓)	0	1 ↗
2 <sup>nd</sup> (↓)	1	0
3 <sup>rd</sup> (↓)	1	1
4 <sup>th</sup> (↓)	0	0

(C-818) Fig. 6.7.2 : Timing diagram for a 2-bit synchronous counter

### 6.7.2 3-Bit Synchronous Binary up Counter :

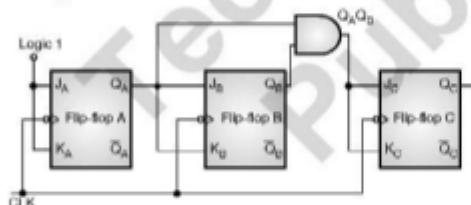
SPPU : Dec. 10

#### University Questions

- Q. 1 Design Mod 8 synchronous counter using T flip-flop.  
(Dec. 19, 6 Marks)

#### Logic diagram :

- We can extend the principle of operation of a 2-bit synchronous counter to a 3-bit counter shown in Fig. 6.7.3.



(C-817) Fig. 6.7.3 : A 3-bit synchronous binary counter

- FF-A acts as a toggle FF since  $J_A = K_A = 1$ .
- $Q_A$  output of FF-A is applied to  $J_B$  as well as  $K_C$ . Hence if  $Q_A = 1$  at the instant of triggering, then FF-B will toggle but if  $Q_A = 0$  then FF-B will not change its state.
- $Q_A$  and  $Q_B$  are ANDed and the output of AND gate is applied to  $J_C$  and  $K_C$ .
- Hence when  $Q_A$  and  $Q_B$  both are simultaneously high, then  $J_C = K_C = 1$  and FF-C will toggle. Otherwise there is no change in the state of FF-C.

#### Operation :

- Initially all the FFs are in their reset state.

$$\therefore Q_C Q_B Q_A = 0 \ 0 \ 0$$

#### 1<sup>st</sup> clock pulse :

- FF-A toggles and  $Q_A$  changes to 1 from 0. But since  $Q_A = 0$  at the instant of application of 1<sup>st</sup> falling clock edge,  $J_B = K_C = 0$  and  $Q_B$  does not change state.  $\therefore Q_B$  remains 0.
- Similarly  $Q_C$  also does not change state.

$$\therefore Q_C = 0.$$

$$\therefore Q_C Q_B Q_A = 0 \ 0 \ 1 \quad \dots \text{After 1}^{\text{st}} \text{ clock pulse}$$

#### 2<sup>nd</sup> clock pulse :

- FF-A toggles and  $Q_A$  becomes 0.
- But at the instant of application of 2<sup>nd</sup> falling clock edge  $Q_A$  was equal to 1. Hence  $J_B = K_C = 1$ . Hence FF-B will toggle and  $Q_B$  becomes 1.
- Output of AND gate is 0 at the instant of negative clock edge. So  $J_C = K_C = 0$ . Hence  $Q_C$  remains 0.

$$\therefore Q_C Q_B Q_A = 0 \ 1 \ 0 \quad \dots \text{After the 2}^{\text{nd}} \text{ clock pulse}$$

#### 3<sup>rd</sup> clock pulse :

- After the 3<sup>rd</sup> clock pulse, the outputs are  $Q_C Q_B Q_A = 0 \ 1 \ 1$ .

#### 4<sup>th</sup> clock pulse :

- Note that  $Q_B = Q_A = 1$ . Hence output of AND gate = 1 and  $J_C = K_C = 1$ , at the instant of application of 4<sup>th</sup> negative edge of the clock.
- Hence on application of this clock pulse, FF-C will toggle and  $Q_C$  changes from 0 to 1.
- FF-A toggles as usual and  $Q_A$  becomes 0.
- Since  $Q_A$  was equal to 1 earlier, FF-B will also toggle to make  $Q_B = 0$ .
- $\therefore Q_C Q_B Q_A = 1 \ 0 \ 0 \quad \dots \text{After the 4}^{\text{th}} \text{ clock pulse}$
- Thus the counting progresses.
- After the 7<sup>th</sup> clock pulse the output is 111 and after the 8<sup>th</sup> clock pulse, all the flip-flops toggle and change their outputs to 0. Hence  $Q_C Q_B Q_A = 0 \ 0 \ 0$  after the 8<sup>th</sup> pulse and the operation repeats.
- Table 6.7.2 summarizes operation of the three bit synchronous counter.

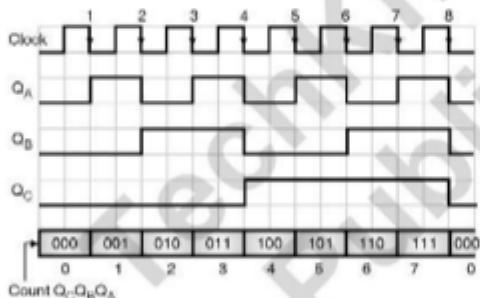


(C-6225) Table 6.7.2 : Summary of operation of a 3-bit synchronous counter

Clock	$Q_C$	$Q_B$	$Q_A$
0	0	0	0
1 <sup>st</sup> (↑)	0	0	1
2 <sup>nd</sup> (↑)	0	1	0
3 <sup>rd</sup> (↑)	0	1	1
4 <sup>th</sup> (↑)	1	0	0
5 <sup>th</sup> (↑)	1	0	1
6 <sup>th</sup> (↑)	1	1	0
7 <sup>th</sup> (↑)	1	1	1

Timing diagram :

- Timing diagram for a 3-bit synchronous counter is shown in Fig. 6.7.4.
- The number of states through which this counter progresses is 8 namely  $Q_C Q_B Q_A = 000, 001, 010, 011, 100, 101, 110, 111$ .
- The maximum count is  $(111)_2$  or  $(7)_{10}$ .
- Note that the waveforms of synchronous counter are exactly same as those of an asynchronous counter.



(C-818) Fig. 6.7.4 : Timing diagram for a 3-bit synchronous counter

### 6.7.3 Design of the 3 Bit Synchronous Counter :

SPPU : Dec. 06, May 12

#### University Questions

- Q. 1** Design 3-bit synchronous counter using J-K flip-flop and explain. (Dec. 06, 8 Marks)
- Q. 2** Design and implement 3 bit synchronous counter using JK FF. (May 12, 8 Marks)

- Let us now design a 3 bit synchronous counter first using the T flip flops and then using JK flip flops.

#### Design using T flip flops :

##### Steps to be followed :

- Step 1 : Decide the number of flip flops.
- Step 2 : Write the excitation table of T flip flop.
- Step 3 : Write the excitation table of the counter.
- Step 4 : From the circuit excitation table write K-maps and obtain simplified equations.
- Step 5 : Draw the logic diagram.

##### Step 1 : Decide number of FFs :

- A 3 bit counter goes through 8 states. So it needs three flip flops.

##### Step 2 : Excitation table of T FFs :

- Table 6.7.3(a) shows the excitation table of T FF.

(C-7842) Table 6.7.3(a) : Excitation table of a T FF

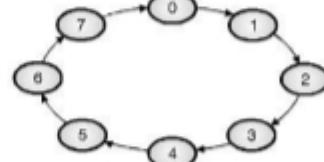
Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

##### Step 3 : State diagram and circuit excitation table :

Count sequence for a 3 bit up counter is given in Table 6.7.3(b) and Fig. 6.7.5 shows the corresponding state diagram.

(C-6226) Table 6.7.3(b)

$Q_C$	$Q_B$	$Q_A$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



(C-819) Fig. 6.7.5 : State diagram

- Table 6.8.3(c) shows the circuit excitation table.

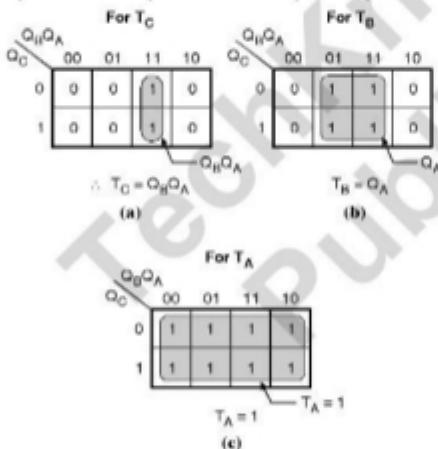


(C-753B) Table 6.7.3(c) : Circuit excitation table

Present state			Next state			Flip flop input		
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_C$	$T_B$	$T_A$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	0	0	0	1	1
1	1	0	1	1	0	0	0	1
1	1	1	0	0	0	1	1	1

- Table 6.7.3(c) has been written by referring to the present and next state of an output and the required value to input is written as per the excitation table of T FF.
- For example consider the shaded columns of Table 6.7.3(c). i.e.  $Q_C, Q_{C+1}$  and  $T_C$ .
- Consider the first row,  $Q_C = 0, Q_{C+1} = 0$ . So as per the excitation table of a T FF  $T_C$  should be 0. Similarly all other entries are made.

**Step 4 : Write K maps and obtain simplified equations :**

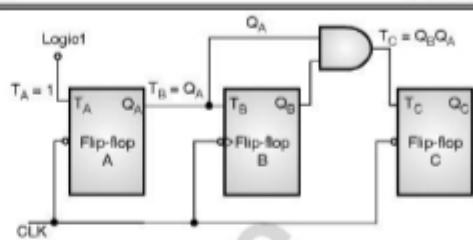


(C-1506) Fig. 6.7.6 : K-maps for different FF inputs

- Refer Figs. 6.7.6(a), (b), (c) for the K-maps corresponding to all the FF inputs. The simplified equations for  $T_A, T_B$  and  $T_C$  also are shown.

**Step 5 : Draw the logic diagram :**

- By using the simplified equations for  $T_A, T_B$  and  $T_C$  we can draw the logic diagram for the 3 bit synchronous shown in Fig. 6.7.7.



(C-1507) Fig. 6.7.7 : Logic diagram of a 3-bit synchronous counter

**Design using JK flip flops :**

**Step 1 : Number of flip flops :**

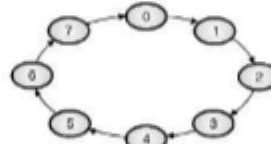
- For a 3-bit counter, we need 3 flip-flops.

**Step 2 : Excitation table of JK FF :**

(C-7761) Table 6.7.4 : Excitation table of JK FF

Present state $Q_n$	Next state $Q_{n-1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

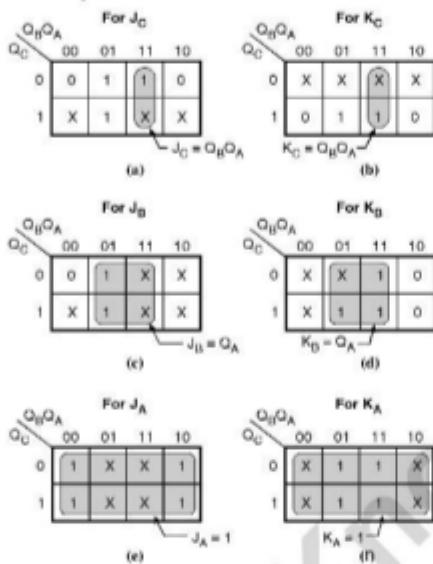
**Step 3 : State diagram and circuit excitation table :**



(C-1508) Fig. 6.7.8 : State diagram

(C-7050) Table 6.7.5 : Circuit excitation table

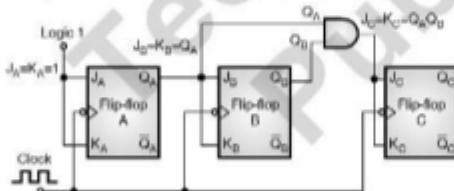
Present state	Next state	Flip flop inputs									
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

**Step 4 : K maps and simplified expressions for all FF****Inputs :**

(C-1509) Fig. 6.7.9 : K-maps and simplified equations for different FF inputs

**Step 5 : Logic diagram :**

- Fig. 6.7.10 shows the logic diagram of a 3 bit synchronous counter using JK flip-flops.

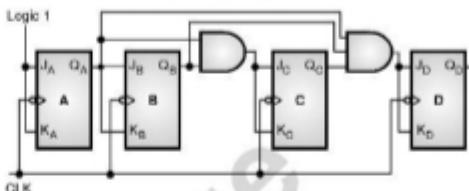


(C-1510) Fig. 6.7.10 : 3 bit synchronous counter using JK FFs

**6.7.4 Four Bit Synchronous Up Counter :****Logic diagram :**

- All the concepts of a synchronous counter are extended to design a 4-bit synchronous counter shown in Fig. 6.7.11.
- Note that  $J_D$  and  $K_D$  of the FF-D are connected to the output of an AND gate and the inputs of this AND gate come from the outputs of the three preceding FFs.

- Operating principle of this counter is same as that of the 3-bit counter discussed earlier.



(C-1391) Fig. 6.7.11 : A four bit synchronous counter

**6.8 Modulo – N Synchronous Counters :**

- We have already discussed the MOD-N asynchronous counters. Now let us discuss the Modulo-N synchronous counters.

- The steps to be followed to design a MOD-N synchronous counter are as follows.

**Steps to be followed :**

- Step 1 : Decide the number of FFs and type of FF to be used.
- Step 2 : Write the circuit excitation table.
- Step 3 : From the circuit excitation table write down the K-maps and obtain simplified expressions for the outputs.
- Step 4 : Draw the logic diagram and timing diagram.

**6.8.1 Synchronous Decade Counter :****SPPU : Dec. 05, May 06, Dec. 09****University Questions**

- Q. 1** Design and implement synchronous BCD counter using T flip flops.

(Dec. 05, May 06, Dec. 09, 8 Marks)

- For the design of a synchronous decade counter follow the steps given below :

- Step 1 : Write the excitation table for T FF and circuit excitation table :**

- Excitation table for T FF is shown in Table 6.8.1.

(C-7842) Table 6.8.1 : Excitation table for T flip-flop

Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

- The circuit excitation table is shown in Table 6.8.2.

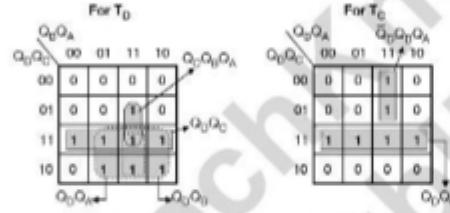


(C-7839) Table 6.8.2 : Circuit excitation table

Present state			Next state			Flip flop inputs					
$Q_D$	$Q_C$	$Q_B$	$Q_A$	$Q_{D+1}$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_D$	$T_C$	$T_B$	$T_A$
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	1
1	0	1	0	0	0	0	0	1	0	1	0
1	0	1	1	0	0	0	0	1	1	0	0
1	1	0	0	0	0	0	0	1	1	1	0
1	1	1	1	0	0	0	0	1	1	1	1

**Step 2 : K-maps and simplifications :**

K-maps for  $T_D$ ,  $T_C$ ,  $T_B$ ,  $T_A$  and their simplified expressions are given in Figs. 6.8.1(a), (b), (c) and (d).



$$\begin{aligned} T_D &= Q_D Q_B Q_A + Q_D Q_B + Q_D Q_A + Q_D Q_A \\ \therefore T_D &= Q_D Q_B Q_A + Q_D (Q_B + Q_A + Q_A) \end{aligned}$$

(a) K map for  $T_D$ 

$$\begin{aligned} T_C &= Q_D Q_B Q_A + Q_D Q_B + Q_D Q_A \\ \therefore T_C &= Q_D Q_B Q_A + Q_D (Q_B + Q_A + Q_A) \end{aligned}$$

(b) K map for  $T_C$ 

$$\begin{aligned} T_B &= Q_D Q_B + \bar{Q}_D Q_A \\ \therefore T_B &= Q_D Q_B + \bar{Q}_D Q_A \end{aligned}$$

(c) K map for  $T_B$ 

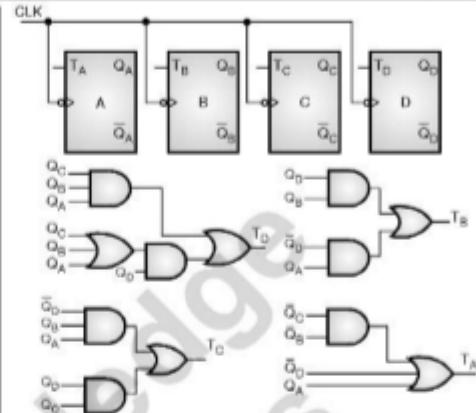
$$\begin{aligned} T_A &= \bar{Q}_D + \bar{Q}_C \bar{Q}_B + Q_A \\ \therefore T_A &= \bar{Q}_D + \bar{Q}_C \bar{Q}_B + Q_A \end{aligned}$$

(d) K map for  $T_A$ 

(C-847) Fig. 6.8.1

**Step 3 : Draw the logic diagram :**

Fig. 6.8.2 shows the logic diagram of a synchronous decade counter.



(C-848) Fig. 6.8.2 : Logic diagram of a synchronous decade counter

**Ex. 6.8.1 :** Design a synchronous counter for the sequence shown in Fig. P. 6.8.1(a).

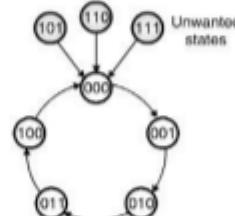
(C-6227) Table P. 6.8.1(a) : Desired sequence

$Q_C$	$Q_B$	$Q_A$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

**Soln. :****Step 1 : Determine the desired number of FFs :**

From the given sequence the number of FFs is equal to 3. This is a MOD-5 synchronous counter since the number of states is 5.

The state diagram is shown in Fig. P. 6.8.1(a) which shows that 101, 110, 111 are unwanted states.



(C-849) Fig. P. 6.8.1(a) : State diagram

**Step 2 : Write the excitation table and state table :**

- The type of FF used is JK flip-flop. The excitation table for a JK FF is as shown in Table P. 6.8.1(b).
- We have already seen how to write the excitation table for JK FF.

**Excitation table of JK FF :**

- Table P. 6.8.1(b) exhibits the excitation table of a JK FF.

(C-7840) Table P. 6.8.1(b) : Excitation table of JK FF

Present state $Q_n$	Next state $Q_{n+1}$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

**Circuit excitation table :**

- The circuit excitation table is as shown in Table P. 6.8.1(c).

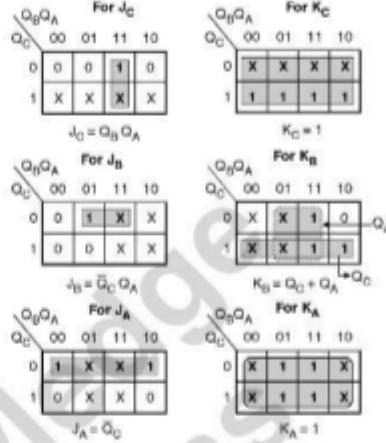
(C-7841) Table P. 6.8.1(c) : Circuit excitation table

Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	Present state		Next state		Flip flop inputs						
			Q <sub>C</sub> =0	Q <sub>B</sub> =1	Q <sub>C</sub> =1	Q <sub>B</sub> =0	Q <sub>A</sub> +1	J <sub>C</sub>	K <sub>C</sub>	J <sub>B</sub>	K <sub>B</sub>	J <sub>A</sub>	K <sub>A</sub>
1	0	0	0	0	1	0	x	0	x	1	x		
1	0	1	0	1	0	0	x	1	x	x	1		
1	1	0	0	1	1	0	x	x	0	1	x		
1	1	1	1	0	0	1	x	x	1	x	1		
0	0	0	0	0	0	x	1	0	x	0	x		
0	0	1	0	0	0	x	1	0	x	x	1		
0	1	0	0	0	0	x	1	x	1	0	x		
0	1	1	0	0	0	x	1	x	1	x	1		

- Refer to the shaded portion of the circuit excitation table. This is nothing but the excitation table of FF-C. The J<sub>C</sub> and K<sub>C</sub> values have been derived based on Q<sub>C</sub> and Q<sub>C+1</sub>.
- Similarly the entries for J<sub>B</sub> and K<sub>B</sub> are based on Q<sub>B</sub> and Q<sub>B+1</sub> whereas those for J<sub>A</sub> and K<sub>A</sub> are based on Q<sub>A</sub> and Q<sub>A+1</sub>.

**Step 3 : K-maps and simplifications :**

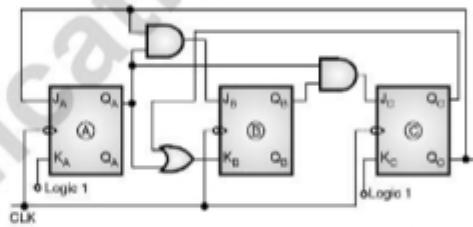
- K-maps for the J and K inputs of all the FFs and the corresponding simplified equations are shown in Fig. P. 6.8.1(b).
- Note that the inputs to this combinational circuit are Q<sub>A</sub>, Q<sub>B</sub>, Q<sub>C</sub> and outputs are J<sub>A</sub>, K<sub>A</sub> through J<sub>C</sub>, K<sub>C</sub>.



(C-850) Fig. P. 6.8.1(b) : K-maps and simplifications

**Step 4 : Logic diagram :**

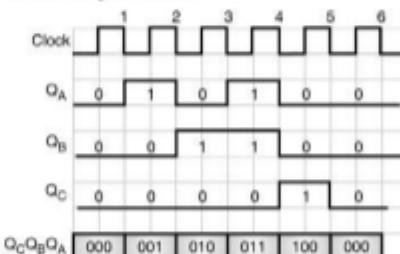
- The logic diagram of MOD-5 synchronous counter is shown in Fig. P. 6.8.1(c).



(C-851) Fig. P. 6.8.1(c) : Logic diagram of MOD-5 synchronous counter

**Step 5 : Draw the timing diagram :**

- The timing diagram of MOD-5 synchronous counter is shown in Fig. P. 6.8.1(d).



(C-852) Fig. P. 6.8.1(d) : Timing diagram of MOD-5 counter



**Ex. 6.8.2 :** Design a MOD-5 synchronous counter using T flip-flops. **Dec. 07, 4 Marks**

**Soln. :**

**Step 1 : Decide number of FFs :**

- Since the number of states is 5 we need to use 3 flip-flops.

**Step 2 : Excitation tables :**

- The excitation table of a T flip-flop is shown in Table P. 6.8.2(a).

(C-7842) Table P. 6.8.2(a) : Excitation table of a T flip-flop

Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

**Circuit excitation table :**

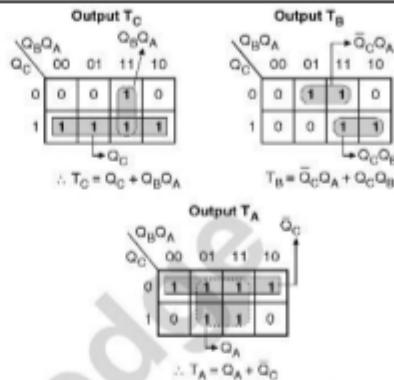
- The circuit excitation table is shown in Table P. 6.8.2(b).
- Refer to the shaded portion of the circuit excitation table.
- This is nothing but the excitation table of FF-C. The  $T_C$  values are decided based on  $Q_B$  and  $Q_{B+1}$ .
- Similarly the entries for  $T_B$  are based on  $Q_A$  and  $Q_{A+1}$ , whereas those for  $T_A$  are based on  $Q_C$  and  $Q_{C+1}$ .

(C-7843) Table P. 6.8.2(b) : Circuit excitation table

Present state	Next state			Flip flop inputs		
	$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$
0 0 0	0	0	1	0	1	0 0 1
0 0 1	0	1	0	1	0	0 1 1
0 1 0	0	1	1	1	0	0 0 1
0 1 1	1	0	0	0	0	1 1 1
1 0 0	0	0	0	0	0	1 0 0
1 0 1	0	0	0	0	1	0 1 0
1 1 0	0	0	0	0	1	1 1 0
1 1 1	0	0	0	0	1	1 1 1

**Step 3 : K-maps and simplifications :**

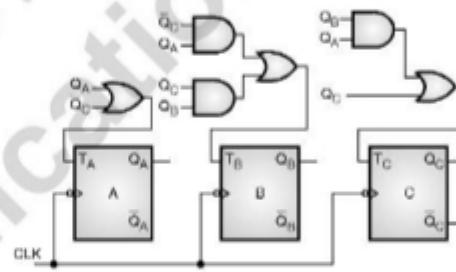
- K-maps for the  $T_A$ ,  $T_B$  and  $T_C$  inputs of the three FFs and the corresponding simplified equations are shown in Fig. P. 6.8.2(a).



(C-853) Fig. P. 6.8.2(a) : K-maps and simplifications

**Step 4 : Draw the logic diagram :**

- The logic diagram of MOD-5 synchronous counter using T FFs is shown in Fig. P. 6.8.2(b).



(C-854) Fig. P. 6.8.2(b) : MOD-5 synchronous counter using T flip-flops

**Ex. 6.8.3 :** Design MOD 13 synchronous counter using T flip-flop. **Dec. 12, 8 Marks**

**Soln. :**

For the design of a synchronous decade counter follow the steps given below :

**Step 1 : Write the excitation table for T FF and circuit excitation table :**

- Excitation table for T FF is shown in Table P. 6.8.3.

(C-7842) Table P. 6.8.3 : Excitation table for T flip-flop

Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

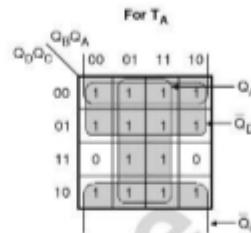
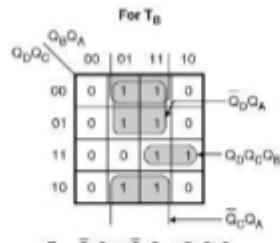
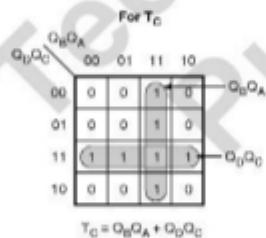
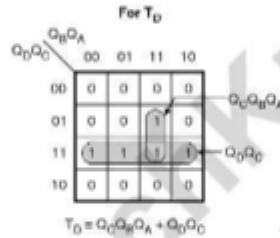


- The circuit excitation table is shown in Table P. 6.8.3(a).

(C-7844) Table P. 6.8.3(a) : Circuit excitation table

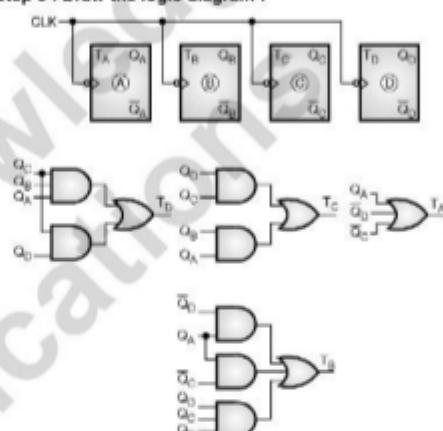
Present state			Next state			Flip flop inputs					
$Q_D$	$Q_C$	$Q_B$	$Q_A$	$Q_{D+1}$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_D$	$T_C$	$T_B$	$T_A$
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	1	0	1	0	0	0	1	1
1	0	1	0	1	0	1	1	0	0	0	1
1	0	1	1	1	1	0	0	0	1	1	1
1	1	0	0	0	0	0	0	1	1	0	0
1	1	0	1	0	0	0	0	1	1	0	1
1	1	1	0	0	0	0	0	1	1	1	0
1	1	1	1	0	0	0	0	1	1	1	1

### Step 2 : K-maps and simplification :



(C-3454) Fig. P. 6.8.3

### Step 3 : Draw the logic diagram :



(C-3455) Fig. P. 6.8.3(a)

Ex. 6.8.4 : Design and implement a synchronous up decade counter using D FF.

Dec. 15, 6 Marks

Soln. :

Step 1 : Write the excitation table for D FF and circuit excitation table :

- Excitation table for D FF is shown in Table P. 6.8.4(a).

(C-7828) Table P. 6.8.4(a) : Excitation table for D flip-flop

Present state	Q output		Input
	Next state	D <sub>n</sub>	
0	0	0	0
0	1	1	1
1	0	0	0
1	1	1	1

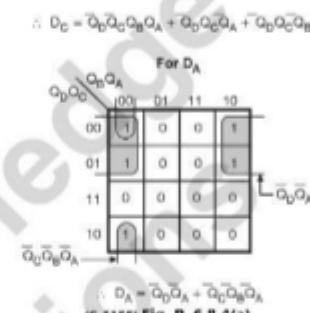
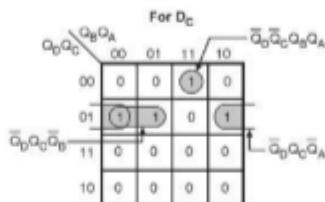
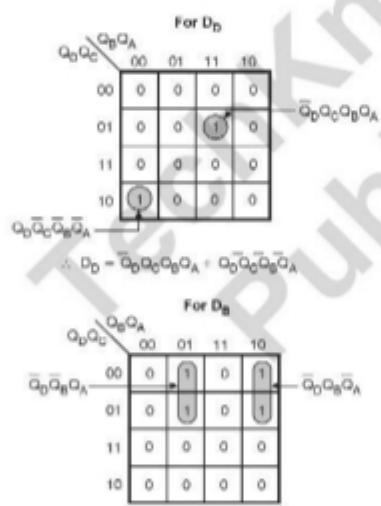
- The circuit excitation table is shown in Table P. 6.8.4(b).



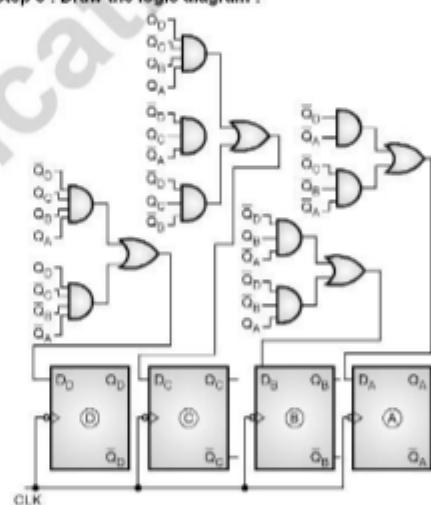
(C-828) Table P. 6.8.4(b) : Circuit excitation table

Present state				Next state				Flip Flop inputs			
$Q_0$	$Q_C$	$Q_B$	$Q_A$	$Q_{D+1}$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$D_0$	$D_C$	$D_B$	$D_A$
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	1	1	0	0	1	1
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	0	0	0	1	0	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0

Step 2 : K-maps and simplifications :



Step 3 : Draw the logic diagram :



(C-5156) Fig. P. 6.8.4(b) : Logic diagram using D flip-flop

Ex. 6.8.5 : Design MOD-6 synchronous counter using Toggle FF (T FF). May 14, May 17, 6 Marks

Solt. :

Step 1 : Decide number of FFs. Since the number of states is 6 we need to use 3 flip-flops.

**Step 2 : Excitation tables :**

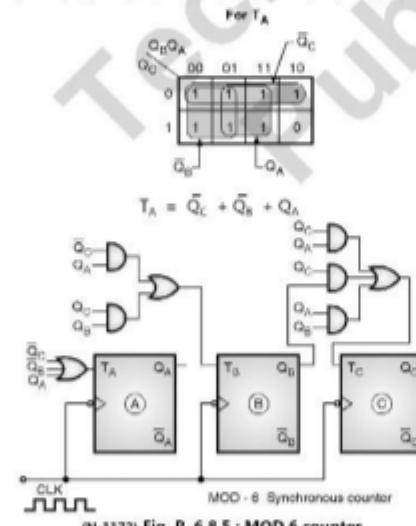
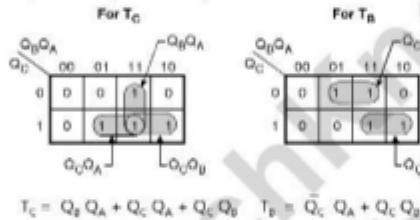
(C-7842) Table P. 6.8.5 : Excitation table of a T flip flop

Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	T
0	0	0
0	1	1
1	0	1
1	1	0

**Circuit excitation table :**

(C-8052) Table P. 6.8.5(a)

Present state			Next state			Flip flop inputs		
Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	Q <sub>C+1</sub>	Q <sub>B+1</sub>	Q <sub>A+1</sub>	T <sub>C</sub>	T <sub>B</sub>	T <sub>A</sub>
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	0	0	0	1	0	1
1	1	0	0	0	1	1	1	0
1	1	1	0	0	0	1	1	1

**Step 3 : K-maps and simplifications :**

(N-1172) Fig. P. 6.8.5 : MOD 6 counter

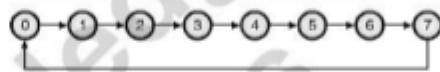
**Ex. 6.8.6 : Design 3 bit synchronous counter using Delay flip flop (D-FF).**  
**May 18, 6 Marks**

**Soln. :****Step 1 : Decide number of FFs :**

- A 3 bit synchronous counter goes through 8 states. So it needs three flip-flops.

**Step 2 : State diagram and circuit and circuit excitation table :**

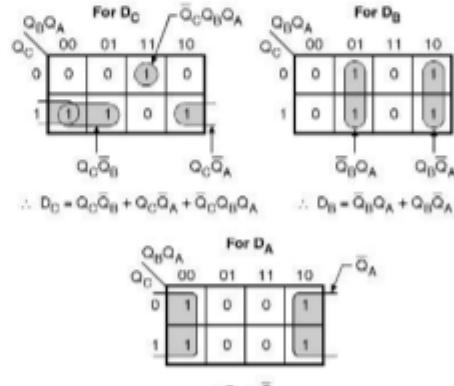
- Fig. P. 6.8.6(a) shows the corresponding state diagram and Table 1 shows the circuit excitation table.



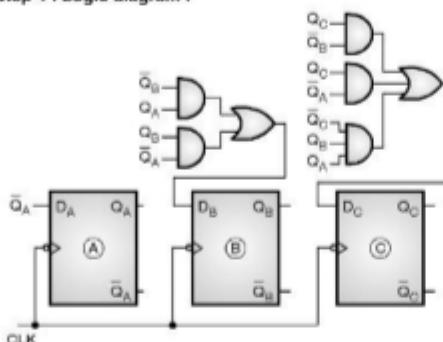
(C-7163) Fig. P. 6.8.6(a) : State diagram

(C-8282) Table P. 6.8.6 : Circuit excitation table

Present state			Next state			Flip Flop inputs		
Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	D <sub>C</sub>	D <sub>B</sub>	D <sub>A</sub>
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	0	0	0	1	0	1
1	1	0	0	0	1	1	1	1
1	1	1	0	0	0	0	0	0

**Step 3 : K-maps and simplification :**

(C-7164) Fig. P. 6.8.6(b) : K-maps

**Step 4 : Logic diagram :**

(C-7185) Fig. P. 6.8.6(c) : Logic diagram using D flip-flop

**6.9 UP / DOWN Synchronous Counter :**

- The operating principle of a synchronous up down counter is same as that of the up/down ripple counters.
- But the design procedure and realization are different.
- The mode control (M) is used for selecting the mode of operation.
- The steps to be followed for the design are as follows:

**Steps to be followed :****Step 1 :** Write the circuit excitation table.**Step 2 :** Write K-maps and obtain the simplified expressions.**Step 3 :** Draw the logic diagram.**6.9.1 3-bit Up/Down Synchronous Counter :**

- The number of FFs to be used is three. We will use three toggle flip-flops.
- Let upcounting takes place with  $M = 0$  and down counting takes place for  $M = 1$ .

**Step 1 : Write the circuit excitation table :**

- The circuit excitation table is shown in Table 6.9.1.

(C-6228) Table 6.9.1 : Excitation table for a 3-bit up/down synchronous counter

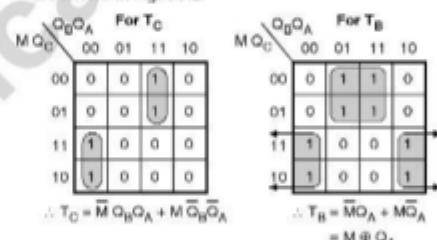
Mode control M	Present state				Next state			Flip-flop inputs		
	$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_C$	$T_B$	$T_A$	
0	0	0	0	0	0	1	0	0	1	
0	0	0	1	0	1	0	0	1	1	
0	0	1	0	0	1	1	0	0	1	
0	0	1	1	1	0	0	1	1	1	
0	1	0	0	1	0	1	0	0	1	
0	1	0	1	1	1	0	0	1	1	
0	1	1	0	1	1	1	0	0	1	
0	1	1	1	0	0	0	1	1	1	
1	0	0	0	1	1	1	1	1	1	
1	0	0	1	0	0	0	0	0	1	
1	0	1	0	0	0	1	0	1	1	
1	0	1	1	0	1	0	0	0	1	
1	1	0	0	0	1	1	1	1	1	
1	1	0	1	1	0	0	0	0	1	
1	1	1	0	1	0	1	0	1	1	
1	1	1	1	1	1	0	0	0	1	

Up counting

Down counting

**Step 2 : K-maps and simplified equations :**

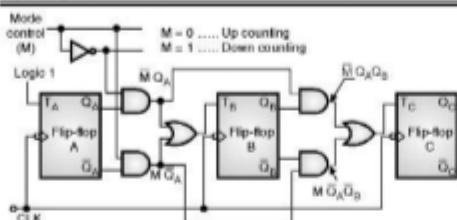
- The K-maps and simplified equations for  $T_C$ ,  $T_B$  and  $T_A$  are shown in Fig. 6.9.1.



(C-855) Fig. 6.9.1 : K-maps and simplified expressions

**Step 3 : Draw the logic diagram :**

- The logic diagram for a 3-bit synchronous up down counter is shown in Fig. 6.9.2.



(IC-856) Fig. 6.9.2 : Logic diagram of a 3-bit synchronous up down counter

### 6.9.2 Advantages of Synchronous Counter :

SPPU : Dec. 05

#### University Questions

**Q. 1** What is the advantage of a synchronous counter over an asynchronous counter ? What is the disadvantage ? **(Dec. 05, 2 Marks)**

- All the FFs receive the clock signal simultaneously i.e. at the same instant, the propagation delay problem is reduced to a great extent. Recall that the propagation delay is one of the problems of the ripple counter.
- The total propagation delay between the instant of application of clock edge and the instant at which the MSB output changes is equal to sum of propagation delay of one FF and that of one AND gate.  
∴ Total propagation delay =  $t_d$  of one FF +  $t_d$  of one gate
- This is much less than the propagation delay of an asynchronous (ripple) counter.

Due to reduced propagation delay, the synchronous counters can operate at a much higher clock frequency than the asynchronous counters.

### 6.9.3 Comparison of Synchronous and Asynchronous Counters :

SPPU : May 07, Dec. 07

#### University Questions

**Q. 1** Differentiate between synchronous and asynchronous counters. **(May 07, 4 Marks, Dec. 07, 2 Marks)**

- Comparison of synchronous and asynchronous counters is given in Table 6.9.2.

Table 6.9.2 : Comparison of synchronous and asynchronous counters

Sr. No.	Parameter	Asynchronous counter	Synchronous counter
1.	Circuit complexity	Logic circuit is simple	With increase in number of states, the logic circuit becomes complicated.
2.	Connection pattern	Output of the preceding FF, is connected to clock of the next FF.	There is no connection between output of preceding FF and CLK of next one.
3.	Clock input	All the FFs are not clocked simultaneously.	All FFs receive clock signal simultaneously.
4.	Propagation delay	P.D. = $n \times (t_d)$ where $n$ is number of FFs and $t_d$ is p.d. per FF.	P.D. = $(t_d)_{FF} + (t_d)_{gate}$ . It is much shorter than that of asynchronous counter.
5.	Maximum frequency of operation	Low because of the long propagation delay.	High due to shorter propagation delay.

### 6.10 Lock Out Condition :

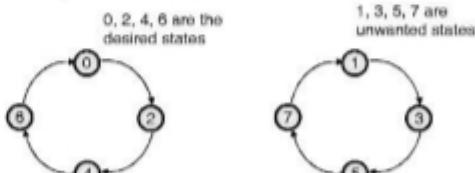
SPPU : Dec. 06, May 07

#### University Questions

**Q. 1** Comment on : Lock out condition. **(Dec. 06, 2 Marks)**

**Q. 2** Describe lock out condition. **(May 07, 2 Marks)**

- A counter is supposed to follow the sequence of only the desired states as shown in Fig. 6.10.1(a).
- If it enters into an unused or unwanted state, then it is expected to return back to a desired state.



(a) State diagram showing the desired sequence

(b) State diagram showing the lockout condition

(C-851) Fig. 6.10.1



- Instead if the next state of an unwanted state is again an unwanted state as shown in Fig. 6.10.1(b) then the counter is said to have gone into the lockout conditions.

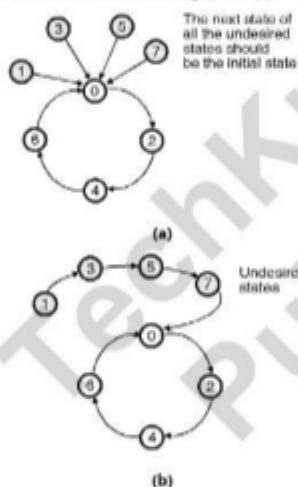
### 6.10.1 Bushless Circuit :

#### Definition :

- The sequential circuit which enters into the lockout condition is called as the bushless circuit.

#### How to avoid lockout situation ?

- In order to avoid the lockout condition, we have to use some additional logic circuit to ensure that the next state of the counter is its initial state, if it enters into an undesired state.
- Thus the next state of each unwanted state should be the initial state as shown in Fig. 6.10.2(a).



(C-832) Fig. 6.10.2 : Ways to avoid lockout condition

- However it is not necessary to terminate all the undesired states into the initial state as shown in Fig. 6.10.2(a).
- It will be sufficient to terminate only one unused state into the initial state as shown in Fig. 6.10.2(b).
- If the circuit enters into say state "3" then its next states will be "5" and "7". As "7" has next state of "0" which is the desired state, the lockout condition is avoided.

#### Sequential counter design using JK and T flip flops :

- Ex. 6.10.1 :** By using suitable flip-flops design a counter to go through the following states.

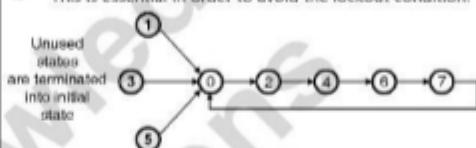
0 – 2 – 4 – 6 – 7 – 0

Avoid the lockout condition.

**Soln. :**

#### Step 1 : Draw the state diagram :

- The state diagram is shown in Fig. P. 6.10.1(a). It shows that the next state of all the unused states (1, 3 and 5) is forced to be the initial state (0).
- This is essential in order to avoid the lockout condition.



(C-833) Fig. P. 6.10.1(a) : State diagram of desired counter

#### Step 2 : Number of flip-flops and type of flip-flop :

- Since the highest state is 7 i.e. 111 we need to use three flip-flops. Let us use JK flip-flops.

#### Step 3 : Write the state table :

- Table P. 6.10.1(a) shows the state table for the required counter.

(C-833) Table P. 6.10.1(a)

Present state	Next state			Flip flop inputs								
	$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0 0 0	0	0	0	1	0	0	0	X	1	X	0	X
0 0 0	0	1	0	0	0	0	0	X	0	X	X	1
0 1 0	0	1	0	0	0	0	1	X	X	1	0	X
0 1 1	1	0	0	0	0	0	0	X	X	1	X	1
1 0 0	0	1	1	1	0	0	X	0	1	X	0	X
1 0 0	1	0	0	0	0	0	X	1	0	X	X	1
1 1 0	0	1	0	1	1	1	X	0	X	0	1	X
1 1 1	1	0	0	0	0	0	X	1	X	1	X	1

#### Step 4 : K maps and simplification :

- Fig. P. 6.10.1(b) illustrates the K-maps for all the FF inputs and the corresponding simplified expressions.

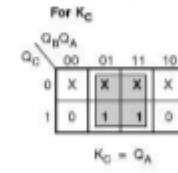
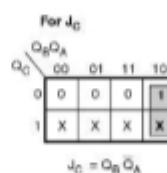
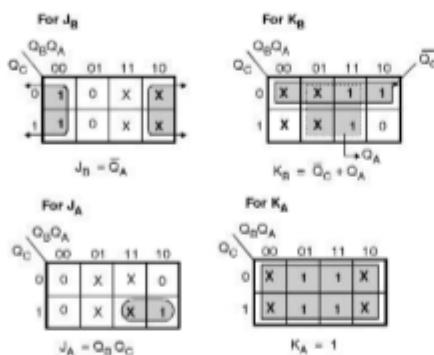


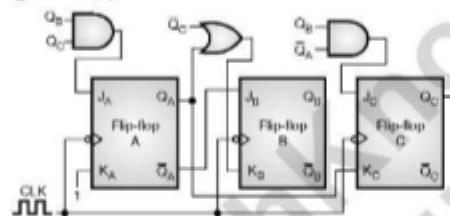
Fig. P. 6.10.1(b) (Cont...)



(C-834) Fig. P. 6.10.1(b) : K-maps and simplification

**Step 5 : Draw the logic diagram :**

The logic diagram for the required counter is shown in Fig. P. 6.10.1(c).

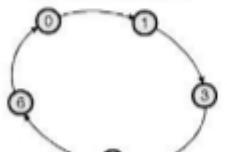


(C-835) Fig. P. 6.10.1(c) : Logic circuit of the required counter

**Ex. 6.10.2 :** By using suitable FFs design a counter to go through the states 0-1-3-4-6-0. Draw the logic diagram. Examine the action of counter for the unused states.

**Soln. :****Step 1 : Draw the state diagram :**

- The state diagram is as shown in Fig. P. 6.10.2(a). Let us use three T flip-flops.



(C-836) Fig. P. 6.10.2(a) : State diagram

**Step 2 : Write the state table :**

- Assuming the next state of unused states to be don't care conditions the state table is written as shown in Table P. 6.10.2(a).

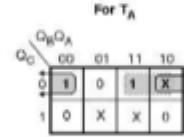
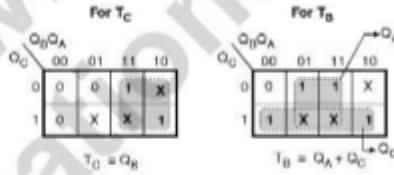
(C-8270) Table P. 6.10.2(a) : State table

		Present states			Next states			Flip-flop inputs		
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_C$	$T_B$	$T_A$		
0	0	0	0	0	1	0	0	1		
0	0	1	0	1	1	0	1	0		
0	1	0	X	X	X	X	X	X		
0	1	1	1	0	0	1	1	1		
1	0	0	1	1	0	0	0	1	0	
1	0	1	X	X	X	X	X	X		
1	1	0	0	0	0	1	1	0		
1	1	1	X	X	X	X	X	X		

Next state of the unused state is considered as don't care

**Step 3 : K-maps and simplifications :**

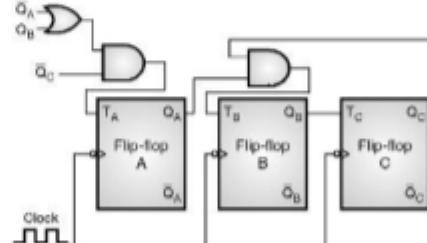
- The K-maps and the corresponding simplified expressions are as shown in Fig. P. 6.10.2(b).



(C-837) Fig. P. 6.10.2(b) : K-maps and simplifications

**Step 4 : Draw the logic diagram :**

- The logic diagram of the required counter is shown in Fig. P. 6.10.2(c).



(C-838) Fig. P. 6.10.2(c) : Logic circuit of the required counter

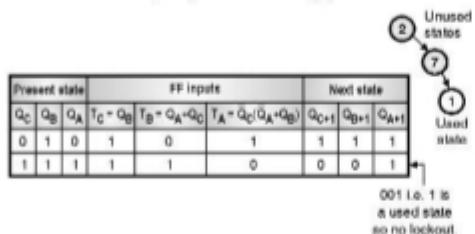


**Step 5 : Examine the counter action for the unused states :**

**For state 010 i.e. 2 :**

$$Q_C = 0, Q_B = 1 \text{ and } Q_A = 0.$$

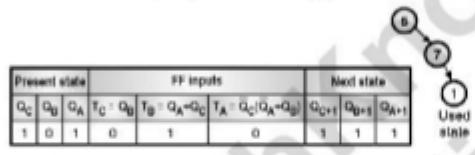
(C-839) Table P. 6.10.2(b)



**For state 101 i.e. 5 :**

- Refer Table P. 6.10.2(c) to know the behaviour of the counter for state 5.

(C-840) Table P. 6.10.2(c)



- In Table P. 6.10.2(b) we have already proved that the counter returns to state "1" from state "7", which is used state.

**For state 111 i.e. 7 :**

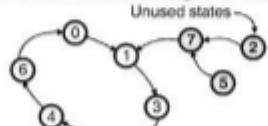
- In Table P. 6.10.2(b) we have already proved that the counter returns to state 1 from "7".

#### Conclusion :

The counter returns to a used state 1 if by chance it enters into one of the unused states 2, 5 or 7. Hence lockout condition is automatically avoided.

**Step 6 : Draw the modified state diagram :**

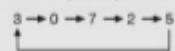
- The modified state diagram showing the used as well as the unused states is shown in Fig. P. 6.10.2(d).



(C-841) Fig. P. 6.10.2(d) : State diagram showing used and unused states

**Ex. 6.10.3 : Design synchronous counter which will go through the following step, using JK flip-flop. 1**

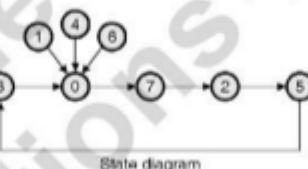
(C-6298)

(Avoid lock out condition). **May 11, 8 Marks****Soln. :**

**Step 1 : State diagram :**

- To avoid the lock out condition the next state of all the unused states (1, 4, 6) is forced to the initial state (0).

**Step 2 : To decide the number of flip-flops required :**



(C-1571) Fig. P. 6.10.3

- Since the highest state is 7 =  $(111)_2$  the number of flip-flops required are 3.

**Step 3 : Write the state table :**

(C-8283)

Present state			Next state			Flip Flop Inputs					
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$J_0$	$K_0$	$J_1$	$K_1$	$J_2$	$K_2$
0	0	0	1	1	1	1	X	1	X	1	X
0	0	1	0	0	0	0	0	X	0	X	X
0	1	0	1	0	1	1	X	X	1	1	X
0	1	1	0	0	0	0	0	X	X	1	X
1	0	0	0	0	0	X	1	0	X	0	X
1	0	1	0	1	1	X	1	1	X	X	0
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	1	1	0	X	1	X	0	X	1

**Step 4 : K-maps and simplification :**

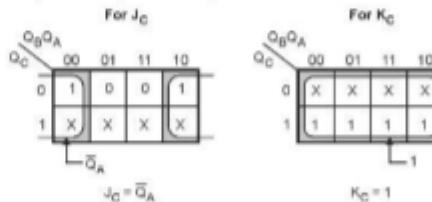


Fig. P. 6.10.3(a)(Contd...)



For $J_B$		For $K_B$	
$Q_B Q_A$	00 01 11 10	$Q_B Q_A$	00 01 11 10
0	0 1 0 X (X)	0	X C
1	0 1 X X	1	0 C
			$\bar{Q}_C \bar{Q}_A$

$J_B = \bar{Q}_C \bar{Q}_A + Q_C Q_A$

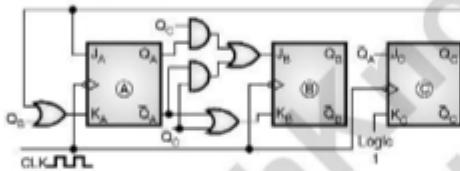
  

For $J_A$		For $K_A$	
$Q_B Q_A$	00 01 11 10	$Q_B Q_A$	00 01 11 10
0	0 1 X X 1	0	X C
1	0 X X 1	1	0 B
			$\bar{Q}_B$

$J_A = \bar{Q}_C$

$K_A = \bar{Q}_C + Q_B$

(C-1572) Fig. P. 6.10.3(a)

**Step 5 : Logic diagram :**

(C-1573) Fig. P. 6.10.3(b)

**6.11 Bush Diagram :**

- In the previous section we have discussed the meaning of lock-out condition.
- The lock-out condition can be avoided by taking a precaution called "bushing". Bushing means adding branches in the state diagram.
- The invalid states of a counter are branched in the bush diagram in such a way that even if the counter enters into an invalid state, it will return to one of its valid states after one, two or three clock cycles.
- The concept of bushing can be well understood by referring to the following example.

**Ex. 6.11.1 :** For the given state diagram in Fig. P. 6.11.1(a) of a counter draw the bush diagrams required to bring the counter from invalid to a valid state in one, two or three clock cycles.

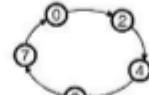
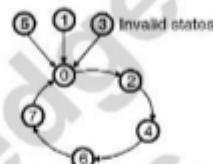


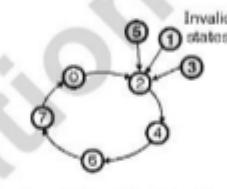
Fig. P. 6.11.1(a) : Given state diagram

**Soln. :**

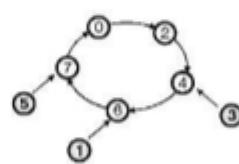
- Refer Figs. P. 6.11.1(b) to (d).



(b) Entry from invalid to valid state after only one clock pulse



(c) Entry from invalid to valid state after only one clock pulse



(d) Entry from invalid to valid state after 1, 2 or 3 clock cycles

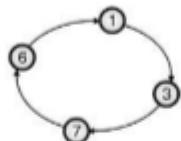
(C-842) Fig. P. 6.11.1

- Fig. P. 6.11.1(b) shows that the next state of each invalid state 1, 3 or 5 is the initial state 0. Hence even if the counter enters into an unused state, it will return to a valid state after only one clock cycle.
- Similarly for the state diagram shown in Fig. P. 6.11.1(c), it will take only one clock pulse to bring the counter into a used state if it enters into an unused one.
- Refer the state diagram of Fig. P. 6.11.1(d). If the counter enters into state "7" then it will return to "0" after one clock pulse.



- If it is in the state "5" then it will return to 0 state after two clock pulses and if it is in the state "1" then it will return to "0" state after three clock pulses.

**Ex. 6.11.2 :** Design a synchronous counter from the state diagram shown in Fig. P. 6.11.2(a). Avoid lockout condition. Draw the bush diagram to avoid the lockout condition.

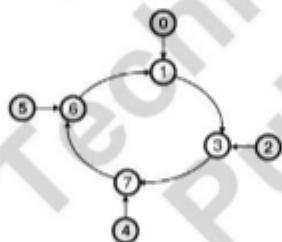


(C-843) Fig. P. 6.11.2(a) : Given state diagram

Soln. :

- So as to avoid the lockout condition, we have to ensure that the counter should return back to a valid state, as soon as it enters into an unused state. The bush diagram is shown in Fig. P. 6.11.2(b).

0, 2, 4 and 5 are the unused states. They are forcibly terminated into 1, 3, 7 and 6 respectively.



(C-844) Fig. P. 6.11.2(b) : Bush diagram

#### Step 1 : Number of FFs and the type of FF :

- Let us use T-type FF.
- Since the highest state in the state diagram is 7, we have to use three T flip-flops.

#### Step 2 : Write the circuit excitation table :

- Let us obtain the circuit excitation table from the bush diagram of Fig. P. 6.11.2(b).

- The excitation table for a T flip-flop is shown in Table P. 6.11.2(a) and the circuit excitation table is shown in Table P. 6.11.2(b).

(C-7842) Table P. 6.11.2(a) : Excitation table for a toggle FF

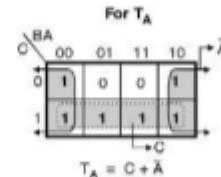
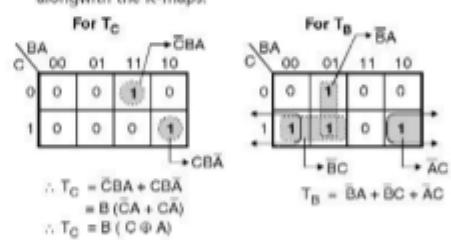
Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

(C-6229) Table P. 6.11.2(b) : Circuit excitation table

Present state	Next state			FF inputs					
	C	B	A	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$T_C$	$T_B$	$T_A$
0 0 0	0	0	0	0	0	1	0	0	1
0 0 1	0	0	1	0	1	1	0	1	0
0 1 0	0	1	0	0	1	1	0	0	1
0 1 1	1	1	1	1	1	1	1	0	0
1 0 0	1	0	0	1	1	1	0	1	1
1 0 1	1	0	1	1	1	0	0	1	1
1 1 0	1	1	0	0	0	1	1	1	1
1 1 1	1	1	1	1	0	0	0	0	1

#### Step 3 : Draw the K-maps and obtain simplified expressions :

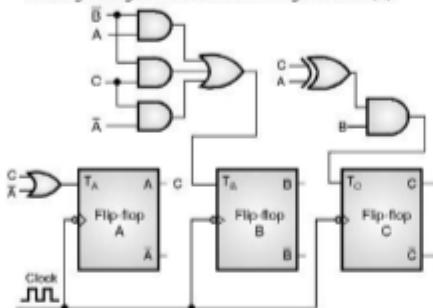
Fig. P. 6.11.2(c) shows the K-maps for various T inputs. The simplified Boolean expressions also are given alongwith the K-maps.



(C-845) Fig. P. 6.11.2(c) : K-maps and simplifications

**Step 4 : Logic diagram :**

- The logic diagram is as shown in Fig. P. 6.11.2(d).



(C-846) Fig. P. 6.11.2(d) : Logic diagram of the required counter

**6.12 Applications of Counters :**

Some of the applications of counters are :

- In digital clock.
- In the frequency counters.
- In time measurement.
- In digital voltmeters.
- In the counter type A to D converter.
- In the digital triangular wave generator.
- In the frequency divider circuits.

**6.12.1 Comparison of Counters and Registers :**

Sr. No.	Parameter	Registers	Counters
1.	Basic building block	R-S, JK flip-flops	T flip-flops, JK flip-flops.
2.	Modes of operation	Serial, parallel	Serial up or down
3.	Change output state	Output need not follow a sequence	Output will always follow a sequence either in the upward or downward direction.
4.	Types	SISO, SIPO, PIPO and PISO	Up counters, down counters, up/down counters.

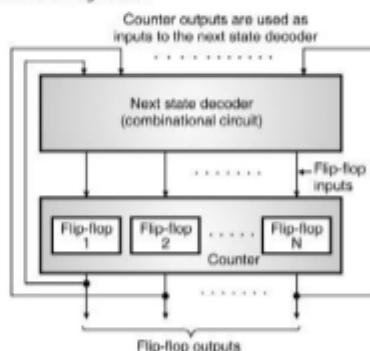
Sr. No.	Parameter	Registers	Counters
5.	Direction of data transfer	Bidirectional	Bidirectional
6.	Number of flip-flops per bit	One	One
7.	Applications	Data storage, data shifting, multiplication or division.	Time and frequency measurement, clock, A to D converter.

**6.13 Sequence Generator :****Definition :**

- A sequence generator is a sequential circuit which generates the prescribed sequence at its output. The output sequence is produced in synchronization with the clock input.
- It is possible to design a sequence generator using either counters or using the shift registers.

**6.13.1 Sequence Generator using Counters :**

- The general block diagram of a sequence generator constructed using counter is shown in Fig. 6.13.1.
- The next state decoder is a combinational circuit. The inputs to it are obtained from the flip-flop outputs and its outputs are applied to the inputs of the flip-flops as shown in Fig. 6.13.1.



(C-857) Fig. 6.13.1 : A general block diagram of a sequence generator

**Deciding the number of flip-flops :**

- We will be given the required sequence of 1's and 0's that is to be generated such as 1011011 ....



- Based on the given sequence we have to decide the number of flip-flops as follows :
  1. Count the number of 1's and 0's in the given sequence.
  2. Choose the higher number of the two. Let this number be N.
  3. The number of FFs (n) is then calculated as,  $N \leq 2^{n-1}$
- Let us apply this principle to the given sequence, i.e. 1011011 ...
- Number of 1's = 5, number of 0's = 2. So select higher one of them i.e. 5. So N = 5.
- But  $N = 2^{n-1}$ . Hence  $5 \leq 2^{n-1}$   
 $\therefore n = 4 \dots$  so four FFs will be required.

**Ex. 6.13.1:** For the following sequence, design a sequence generator using JK flip-flops.  
1 0 0 1 0 0 1.

Soln. :

**Step 1 : Decide the number of flip-flops :**

- Number of 1's = 3 and number of 0's = 4.
- So select higher one of them i.e. 4. So N = 4.
- Hence  $N = 2^{n-1}$ , so  $4 \leq 2^{n-1}$
- Therefore number of flip-flops n = 3.

**Step 2 : Write the state table :**

- Assume that the required sequence is obtained at the output of flip-flop C. The state assignment for the other outputs is shown in Table P. 6.13.1(a).

(C-6247) Table P. 6.13.1(a)

Flip flop outputs			State
C	B	A	
1	0	0	4
0	0	1	1
0	1	0	2
1	1	1	7
0	0	0	0
0	1	1	3
1	1	0	6

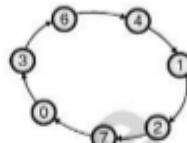
These entries have been made in an arbitrary manner

This is the given sequence

- There is only one unused state i.e. 5. We assume that the next state of 5 is don't care XXX.

**Step 3 : Draw the state diagram :**

- From the state table, draw the state diagram as shown in Fig. P. 6.13.1(a).



(C-658) Fig. P. 6.13.1(a) : State diagram

**Step 4 : Write the excitation table :**

- The excitation table of a JK FF is given in Table P. 6.13.1(b) and the circuit excitation table is shown in Table P. 6.13.1(c).

(C-7761) Table P. 6.13.1(b) : Excitation table of JK FF

Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(C-6398) Table P. 6.13.1(c) : Circuit excitation table

Present state	Next state		Flip-flop inputs									
	C	B	A	C <sub>n+1</sub>	B <sub>n+1</sub>	A <sub>n+1</sub>	J <sub>C</sub>	K <sub>C</sub>	J <sub>B</sub>	K <sub>B</sub>	J <sub>A</sub>	K <sub>A</sub>
0 0 0	0	0	1	1	1	0	0	X	1	X	1	X
0 0 1	0	1	0	0	0	0	0	X	1	X	X	1
0 1 0	1	0	1	1	1	1	1	X	X	0	1	X
0 1 1	1	1	1	0	1	0	1	X	X	0	X	1
1 0 0	0	0	0	1	X	1	0	X	1	X	1	X
1 0 1	X	X	X	X	X	X	X	X	X	X	X	X
1 1 0	1	0	0	0	0	0	0	X	1	0	X	0
1 1 1	0	0	0	0	X	1	X	1	X	1	X	1

The next state and FF inputs are don't care for the unused state.

**Step 5 : K-maps and simplified Boolean expressions :**

- Fig. P. 6.13.1(b) shows the K-maps and corresponding simplified boolean expressions for all the FF inputs.

For J<sub>C</sub>

BA	C			
	00	01	11	10
0	0	0	1	1
1	X	X	X	X

$$J_C = B$$

For K<sub>C</sub>

BA	C			
	00	01	11	10
0	X	X	X	X
1	1	1	X	1

$$K_C = A + B$$

Fig. P. 6.13.1(b)(Contd...)

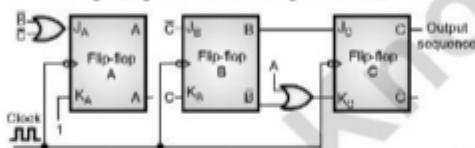


For $J_B$	For $K_B$
$J_B = \bar{C}$	$K_B = C$
For $J_A$	For $K_A$
$J_A = B + \bar{C}$	$K_A = 1$

(c-859) Fig. P. 6.13.1(b) : K-map and simplification

**Step 6 : Draw the logic diagram :**

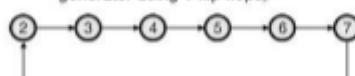
- The logic diagram is shown in Fig. P. 6.13.1(c).



(c-860) Fig. P. 6.13.1(c) : Logic diagram of the required sequence generator

- Note that the required sequence is obtained at the C output.
- All the flip-flops are clocked simultaneously.

**Ex. 6.13.2 :** Design and implement the following sequence generator using T flip-flops.



(c-861) Fig. P. 6.13.2(a) : Given state diagram

**Soln. :**

- From the state diagram it is evident that the maximum count is 7. So we need three T FFs.

**Step 1 : Write the circuit excitation table :**

- Table P. 6.13.2 shows the circuit excitation table for the required sequence generator.

(IC-8300) Table P. 6.13.2								
Present state		Next state			Flip-flop inputs			
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_C$	$T_B$	$T_A$
0	0	0	0	1	0	0	0	1
0	0	1	0	1	0	0	0	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	1	0	1	0	1

- Note that the unused states (0 and 1) are terminated to state 2.

**Step 2 : Write the K-maps for  $T_C$ ,  $T_B$ ,  $T_A$  :**

- The K-maps and simplified expressions are shown in Figs. P. 6.13.2(b), (c) and (d).

For $T_C$		For $T_B$		For $T_A$	
$Q_C$	$Q_B Q_A$	$Q_C$	$\bar{Q}_C \bar{Q}_B \bar{Q}_A$	$Q_C$	$\bar{Q}_C \bar{Q}_B \bar{Q}_A$
0	0 0 0 1	0	1 1 1 0	0	1 1 1 0
1	0 0 1 0	1	1 1 0 0	1	0 0 0 0

$$T_C = Q_B Q_A$$

$$\therefore T_B = Q_C Q_A + Q_B Q_A + Q_C Q_B \\ = Q_A (Q_C + Q_B) + Q_C Q_B$$

(b)

(c)

For $T_A$	
$Q_C$	$Q_A$
0	0 1 1 1
1	1 1 1 1

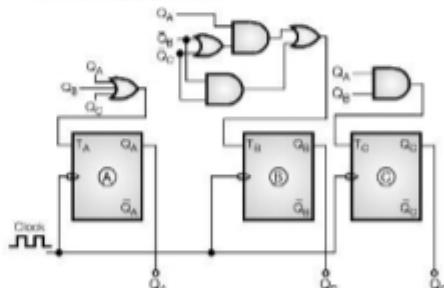
$$T_A = Q_C + Q_B + Q_A$$

(d)

(c-862) Fig. P. 6.13.2(d) : Simplified K-maps

**Step 3 : Draw the logic diagram :**

- The logic diagram of the required sequence generator is shown in Fig. P. 6.13.2(e).



(c-870) Fig. P. 6.13.2(e) : Logic diagram of sequence generator



**Ex. 6.13.3 :** Design a pulse-train generator to generate a pulse train 110011..... using 'D' flip-flop.

**May 06, 8 Marks**

**Soln. :**

**Step 1 : To decide the number of flip-flops :**

Given sequence to be generated is 110011...

Number of 1's in the sequence = 4,

Number of 0's in the sequence = 2

$$\therefore N = 4$$

$$N \leq 2^{n-1} \text{ where } n \rightarrow \text{number of flip-flops}$$

$$\therefore 4 \leq 2^{n-1} \therefore n = 3$$

Hence number of D flip-flops required = 3

**Step 2 : To write the state table :**

- Let the required sequence be obtained output at flip-flop D.

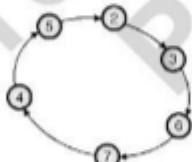
(C-828) Table P. 6.13.3 (a) : State table

Flip flop outputs			State
C	B	A	
1	0	0	4
1	0	1	5
0	1	0	2
0	1	1	3
1	1	0	6
1	1	1	7

- The only unused states are 0 and 1. Let the next state of 0 and 1 be don't care i.e. XXX.

**Step 3 : To draw the state diagram :**

- From the state table, the state diagram can be drawn, for the given sequence, as shown in Fig. P. 6.13.3(a).



(C-1553) Fig. P. 6.13.3(a) : State diagram

**Step 4 : To write the excitation table :**

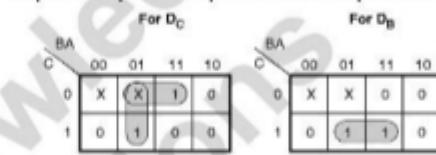
(C-7828) Table P. 6.13.3(b) : Excitation table for D flip-flop

Q output		
Present state	Next state	D <sub>n</sub>
0	0	0
0	1	1
1	0	0
1	1	1

(C-828) Table P. 6.13.3(c) : Circuit excitation table

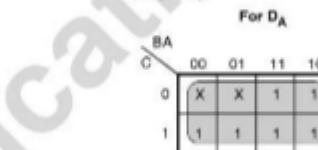
Present state			Next state			Flip Flop inputs		
C	B	A	C <sub>n+1</sub>	B <sub>n+1</sub>	A <sub>n+1</sub>	D <sub>C</sub>	D <sub>B</sub>	D <sub>A</sub>
0	0	0	X	X	X	X	X	X
0	0	1	X	X	X	X	X	X
0	1	0	0	1	1	0	0	1
0	1	1	1	1	0	1	0	1
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	1	1	1
1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1

**Step 5 : K maps and simplified Boolean expressions :**



$$D_C = \overline{BA} + \overline{CA} \\ = A(B + \overline{C})$$

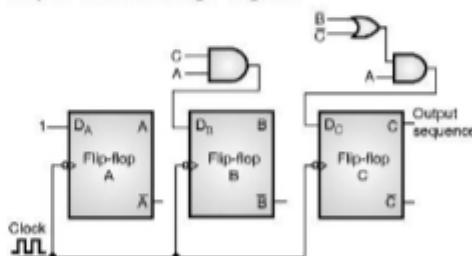
$$D_B = CA$$



$$D_A = 1$$

(C-1554) Fig. P. 6.13.3(b) : K-maps and simplification

**Step 6 : To draw the logic diagram :**



(C-1555) Fig. P. 6.13.3(c) : Logic diagram for required sequence generator

**Ex. 6.13.4 :** Design synchronous counter which will go through the following steps JK flip-flop :

**Dec. 06, 8 Marks, May 12, 10 Marks**

$$0 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 7 \rightarrow 3$$

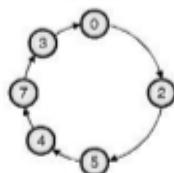
(C-6301)



Soln. :

## Step 1 : Draw the state diagram :

- The state diagram is as shown in Fig. P. 6.13.4(a).



(C-4085) Fig. P. 6.13.4(a) : State diagram

## Step 2 : Number of FF's :

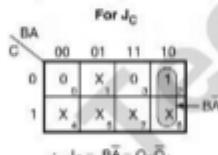
- Since the highest state is 7 i.e. 111. So we have to use three JK FFs.

## Step 3 : Write excitation table :

(C-4286) Table P. 6.13.4(a)

Present state			Next state			Flip Flop inputs					
C	B	A	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	1	0	0	X	1	X	0	X
0	0	1	X	X	X	X	X	X	X	X	X
0	1	0	1	0	1	1	X	X	1	1	X
0	1	1	0	0	0	0	X	X	1	X	1
1	0	0	1	1	1	X	0	1	X	1	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	0	1	1	X	1	X	0	X	0

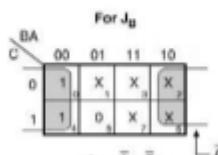
## Step 4 : K-maps and simplification :



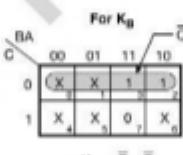
$$\therefore J_C = \bar{B}A = Q_B\bar{Q}_A$$



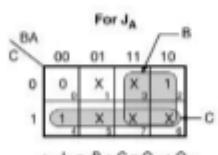
$$\therefore K_C = B = Q_B$$



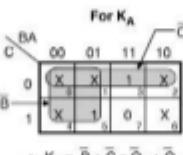
$$\therefore J_B = \bar{A} = \bar{Q}_A$$



$$\therefore K_B = \bar{C} = \bar{Q}_C$$



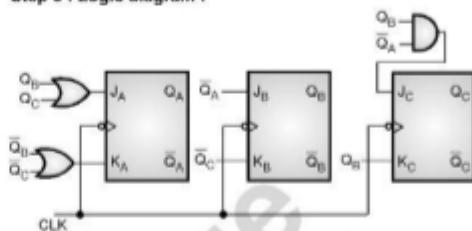
$$\therefore J_A = B + C = Q_B + Q_C$$



$$\therefore K_A = \bar{B} + \bar{C} = \bar{Q}_B + \bar{Q}_C$$

(C-4086) Fig. P. 6.13.4(b)

## Step 5 : Logic diagram :



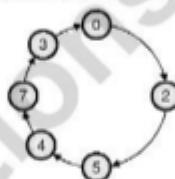
(C-4087) Fig. P. 6.13.4(c) : Logic diagram

Ex. 6.13.5 : Design a circuit to generate sequence 0-2-5-4-7-3 using T FF. Dec. 15, 6 Marks

Soln. :

## Step 1 : Draw the state diagram :

- The state diagram is as shown in Fig. P. 6.13.5(a).



(C-4085) Fig. P. 6.13.5(a) : State diagram

## Step 2 : Number of FF's :

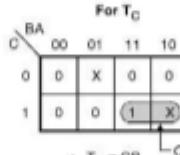
- Since the highest state is 7 i.e. 111. So we have to use three JK FFs.

## Step 3 : Write excitation table :

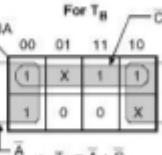
(C-4286) Table P. 6.13.5

Present state			Next state			Flip Flop output					
C	B	A	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$T_C$	$T_B$	$T_A$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$
0	0	0	0	1	0	0	0	1	X	X	X
0	0	1	X	X	X	X	X	X	X	X	X
0	1	0	1	0	1	0	1	1	X	X	X
0	1	1	0	0	0	0	0	1	1	X	X
1	0	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	0	0	0	0	0	0	1
1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	0	1	1	1	1	0	0	0	0

## Step 4 : K-maps and simplification :



$$\therefore T_C = CB = Q_B\bar{Q}_A$$



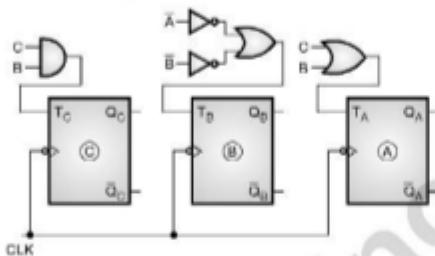
$$\therefore T_B = \bar{A} = \bar{Q}_A$$

Fig. P. 6.13.5(b)(Contd...)



For $T_A$			
C	B	A	$T_A$
0	00 01 11 10	X 1 1 1	$\bar{C}B$
1	1 1 0 0	1 0 X X	$\bar{C}B + \bar{C}B$ $T_A = B \oplus C$

(C-5153) Fig. P. 6.13.5(b)

**Step 5 : Logic diagram :**

(C-5154) Fig. P. 6.13.5(c) : Logic diagram

**Ex. 6.13.6 :** Design the sequence generator using JK flip-flop :  $0 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 0$

Dec. 19, 6 Marks

Soln. :

- Since the maximum count is 6, we can use three JK flip-flops.

**Step 1 : Excitation table :**

The excitation table of JK FF is shown in Table P. 6.13.6(a).

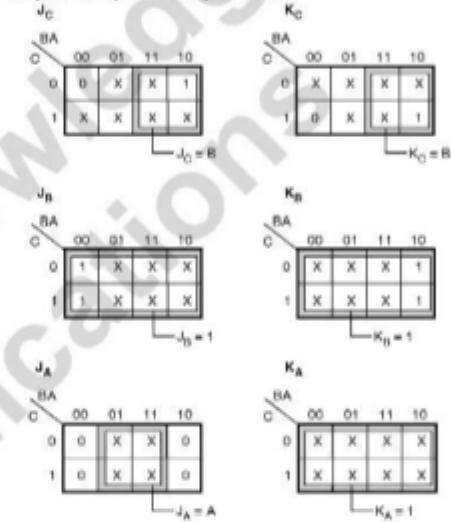
(C-7761) Table P. 6.13.6(a) : Excitation table of JK FF

Present state $Q_n$	Next state $Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

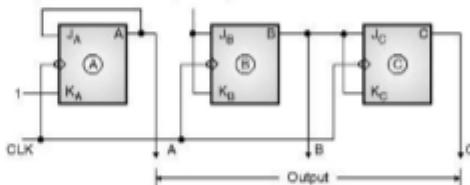
- The circuit excitation table is shown in Table P. 6.13.6(b).
- For the unused states the next state and FF inputs are don't care conditions.

(C-8287) Table P. 6.13.6(b) : Circuit excitation table

Present state	Next state			Flip Flop inputs								
	C	B	A	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0 0 0	0	1	0	0	0	0	0	X	1	X	0	X
0 0 1	1	X	X	X	X	X	X	X	X	X	X	X
0 1 0	0	1	0	0	1	0	X	X	1	0	X	
0 1 1	X	X	X	X	X	X	X	X	X	X	X	X
1 0 0	1	1	1	0	X	0	1	X	0	1	X	D
1 0 1	X	0	X	X	X	X	X	X	X	X	X	X
1 1 0	0	X	0	X	0	X	1	X	1	0	X	
1 1 1	X	0	X	X	X	X	X	X	0	X	X	X

**Step 2 : K-maps and simplification :**

(C-3610) Fig. P. 6.13.6(a) : K maps and simplification

**Step 3 : Draw the logic diagram :**

(C-3611) Fig. P. 6.13.6(b) : Sequence generator

**Review Questions**

- Q. 1 What is counter ? What are its types ?  
Q. 2 Explain the terms "synchronous" and "asynchronous".



- |       |   |       |  |
|-------|---|-------|--|
| Q. 3  | What is the difference between synchronous and asynchronous counter ?                       | Q. 13 | Explain the working of 4-bit ripple counter (asynchronous counter) using T flip-flop with suitable circuit diagram and timing diagram. |
| Q. 4  | What are the merits and demerits of the synchronous counter over the asynchronous counter ? | Q. 14 | Write a note on Mod-8 synchronous up-counter using T flip-flop.  |
| Q. 5  | State the procedure for designing mod counters from N-bit ripple counter.                   | Q. 15 | Write a note on Mod-8 synchronous down counter using T flip-flop.  |
| Q. 6  | What is the similarity between a decade counter and a binary counter ? Explain in brief.    | Q. 16 | Draw the divide by 7 asynchronous up counter using T flip flop. Write truth table. Draw timing diagram.                                |
| Q. 7  | What are the applications of counter ?  | Q. 17 | Draw the circuit for mod-12 counter. Explain the same with neat waveforms.   |
| Q. 8  | How is a counter different from a register ?  | Q. 18 | Compare synchronous and ripple counters.   |
| Q. 9  | Write the count sequence of three bit binary down counter.                                  | Q. 19 | Compare counters and registers.  |
| Q. 10 | How does a counter works as frequency divider ?   | Q. 20 | Design a 3 bit synchronous counter using JK flip flops.  |
| Q. 11 | What is synchronous counter ?   | Q. 21 | What is lock out ? How is it avoided ?   |
| Q. 12 | Define modulus of counter. State the states of MOD 5 counter.                               | Q. 22 | Explain decade counter.  |

□□□

