FOR INTERNAL CIRCULATION ONLY

LECTURE NOTES

ON

DIGITAL ELECTRONICS AND MICROPROCESSER

DIPLOMA 5TH SEMESTER

COMPILED BY

ER.Y.RAJANI

ASST. PROFESSOR

DEPARTMENT OF ELECTRICAL ENGINEERING

UTKAL INSTITUTE OF ENGINEERING AND TECHNOLOGY

Affiliated to SCTE&VT, Govt. of Odisha Approved by AICTE, Govt. of India

CONTENTS

CHAPTER -1

INTRODUCTION

SIGNAL

*Based on electronics Signal is defined as a time varying electric current, voltage or electromagnetic wave used to convey information or data from one place to another.

*Mathematically, signal is a function of one or more than one independent variables.

TYPES OF SIGNAL

There are two types of signal:

1-Analog signal

2-Digital signal

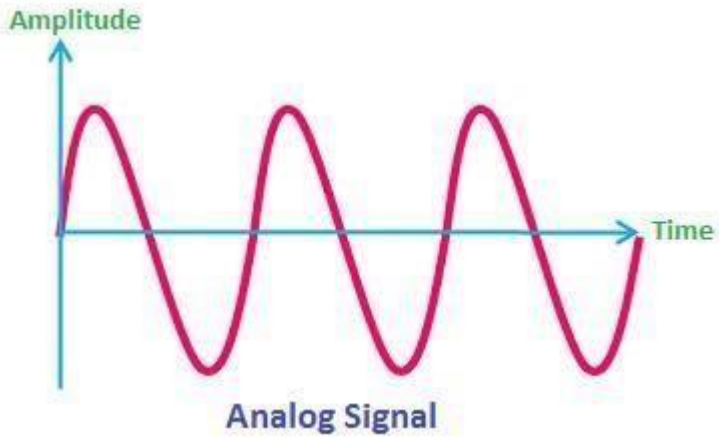
ANALOG SIGNAL

*The signal which is defined continuously for any value of an independent variable(e.g. time) is known as analog signal.

*Alternatively,it is also known as continuous signal

*Most of the commonly used signals are analog signal.

*Example

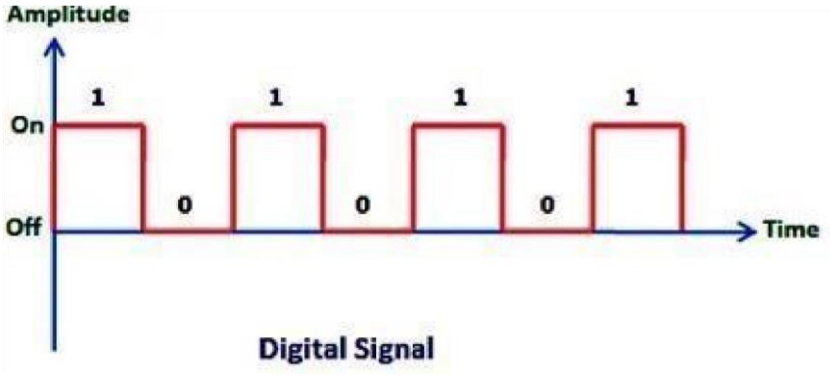


DIGITAL SIGNAL

*The signal which is defined for discrete interval of independent and dependent variables is known as digital signal.

*This signal has only two values 0 and 1.

*Example:



ANALOG ELECTRONICS

The field of electronics deals with analog or continuous signals and the devices that use or produce them is known as analog electronics.

DIGITAL ELECTRONICS

The field of electronics deals with digital or discrete signal and the devices that use or produce them is known as digital electronics.

NUMBER SYSTEM

*The number system is defined as a system to represent or express numbers by using digits or other symbols.

*It represents any data or information quantitatively.

TYPES OF NUMBER SYSTEM

There are four types of number system:

1-Decimal number system
2-Binary Number system
3-Octal number system
4-Hexadecimal number system

DECIMAL NUMBER SYSTEM

*The number system that uses total 10 symbols for representing any number is known as decimal number system.

*The symbols are : 0,1,2,3,4,5,6,7,8,9.

*Its radix value is 10.

*Examples : $(34)_{10}$, $(451)_{10}$, $(101)_{10}$ etc.

BINARY NUMBER SYSTEM

*The number system that uses total 2 symbols for representing any number is known as binary number system.

*The symbols are : 0,1.

*Its radix value is 2.

*Examples : $(110)_2$, $(11.01)_2$, $(1011)_2$ etc.

OCTAL NUMBER SYSTEM

*The number system that uses total 8 symbols for representing any number is known as octal number system.

*The symbols are : 0,1,2,3,4,5,6,7.

*Its radix value is 8.

*Examples : $(36)_8$, $(457)_8$, $(1011)_8$ etc.

HEXADECIMAL NUMBER SYSTEM

*The number system that uses total 16 symbols for representing any number is known as hexadecimal number system.

*The symbols are : 0,1,2,3,4,5,6,7,8,9.A, B, C, D,E,F.

*Its radix value is 16.

*Examples : $(39)_{16}$, $(45C)_{16}$, $(1101)_{16}$ etc.

RADIX OF NUMBER SYSTEM

The total number of symbols used to represent a number system is known as radix of that number system.

Examples :

Radix of decimal number system=10

Radix of binary number system=2

Radix of octal number system=8

Radix of hexadecimal number system=16

CHAPTER-2

BINARY ADDER

Definition of Binary Adder

The logic circuit used for adding binary numbers is known as binary adder. Types of Binary Adder

There are mainly two types of binary adder : a-Half Adder

b-Full Adder

Half and full adders are the basic building blocks of binary adder.All other binary adders are constructed by suitable combination of these two adders e.g.4-bit parallel binary adder.

HALF ADDER

*The logic circuit used for adding two binary digits is known as half adder.

*It has two inputs and two outputs

Inputs are named as : A, B

Outputs are named as : Sum, Carry *Truth Table:

Truth table shows the working of adder in a tabular form i.e. the outputs for all possible combination of inputs.

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

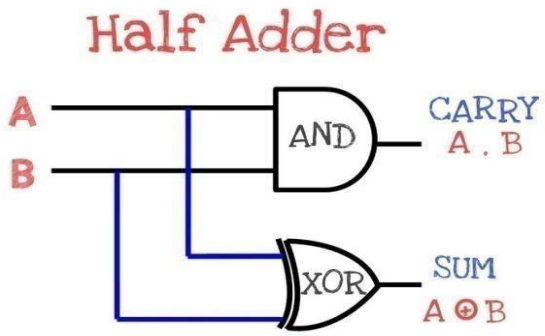
*Boolean Expression:

The Boolean expression that shows the mathematical relationship between inputs and outputs of half adder is represented by two equations, one for sum output and another for carry output as given below:

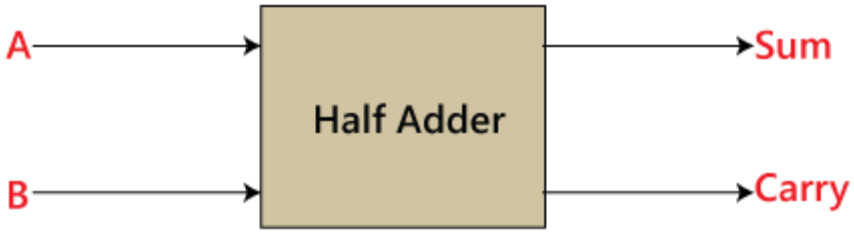
$$S = A\bar{B} + \bar{A}B$$

$$C_i = AB$$

*Logic Circuit for half adder



*Block diagram of half adder:



BINARY FULL ADDER:

*The logic circuit that adds three binary digits at a time is known as binary full adder.

*It has three inputs and two outputs.Inputs are denoted by symbols A,B,C_{in}.

Outputs are denoted by symbols S(Sum) and C(Carry). *Truth table of full adder :

Truth table shows the working of full adder in a tabular form i.e. the outputs for all possible combination of inputs.Full adder has 8 numbers of possible input states, accordingly 8 number of output states as shown below:

Inputs			Outputs	
A	B	C_{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

*Boolean Expression:

Two boolean equations, one for sum output and another for carry output are as below:

$$\text{SUM} = \bar{A}\,\bar{B}\,C_{in} + \bar{A}\,B\,\bar{C}_{in} + A\,\bar{B}\,\bar{C}_{in} + A\,B\,C_{in}$$

$$= C_{in}\,(\bar{A}\,\bar{B} + A\,B) + \bar{C}_{in}\,(A\,\bar{B} + \bar{A}\,B)$$

$$= C_{in}\,\left[(\bar{A} + B)\cdot(A + \bar{B})\right] + \bar{C}_{in}\,(A\,\bar{B} + \bar{A}\,B)$$

$$= C_{in}\,(\overline{A\,\bar{B}}\cdot\overline{\bar{A}\,B}) + \bar{C}_{in}\,(A\,\bar{B} + \bar{A}\,B)$$

$$= C_{in}\,\overline{(A\,\bar{B} + \bar{A}\,B)} + \bar{C}_{in}\,(A\,\bar{B} + \bar{A}\,B)$$

$$= C_{in} \oplus (A\,\bar{B} + \bar{A}\,B)$$

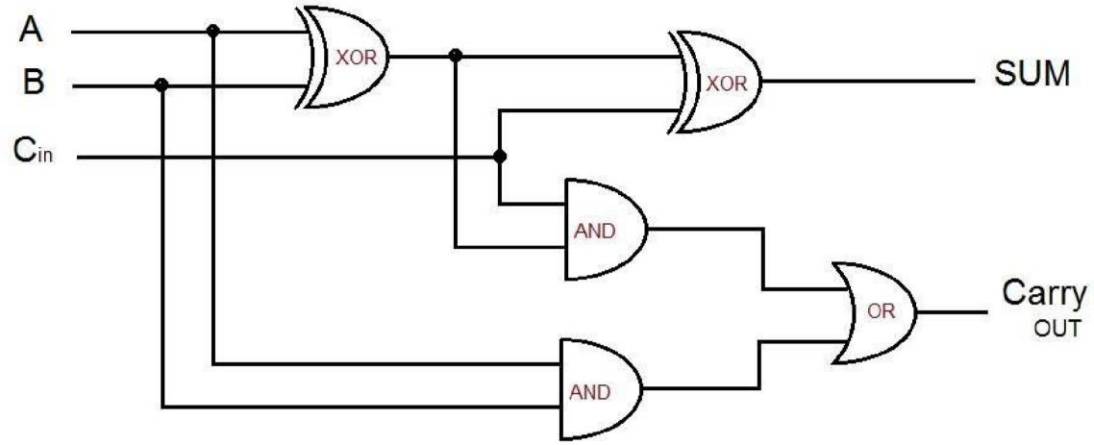
$$= C_{in} \oplus (A \oplus B)$$

$$Carry = ABC_{in} + A\bar{B}C_{in} + \bar{A}BC_{in} + \bar{A}\bar{B}\bar{C}_{in}$$

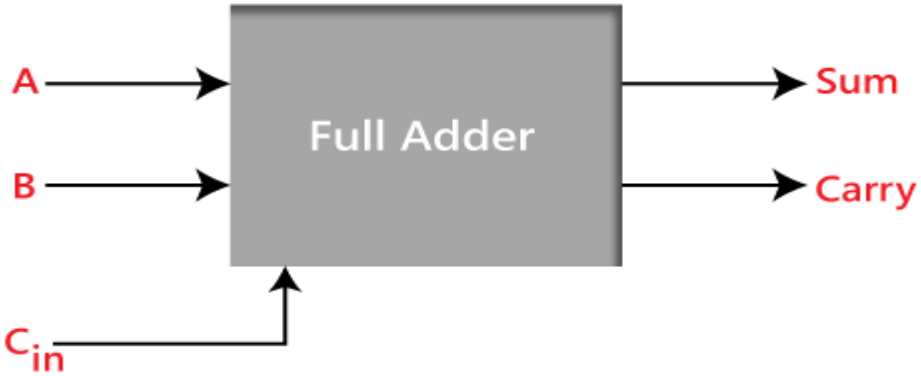
$$= C_{in}(AB + A\bar{B}) + \bar{A}B(C_{in} + \bar{C}_{in})$$

$$= C_{in}(A + B) + \bar{A}B$$

*Logic circuit of full adder :



*Block diagram of full adder:



CHAPTER-3
BINARY SUBTRACTOR

Binary Subtractor

The combinational logic circuit that performs binary subtraction is known as binary subtractor. **Types of Subtractor**

There are mainly two types of binary subtractor:
1-Half subtractor 2-Full Subtractor

Half Subtractor

*The logic circuit that subtracttwo binary bits at time is known as half subtractor.
*It has two inputs symbolised as :A, B
*It has two outputs denoted by symbols :D(Difference), B_i (Borrow). *Truth Table of half subtractor:

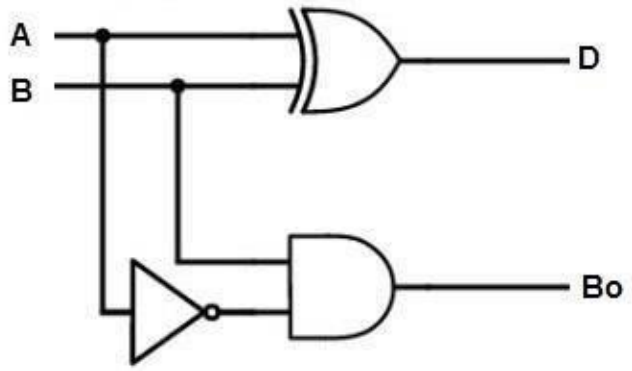
Inputs		Outputs	
A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

*Boolean Expression:
Half subtractor has two Boolean expressions, one for difference(D) output and another for borrow(B) output.

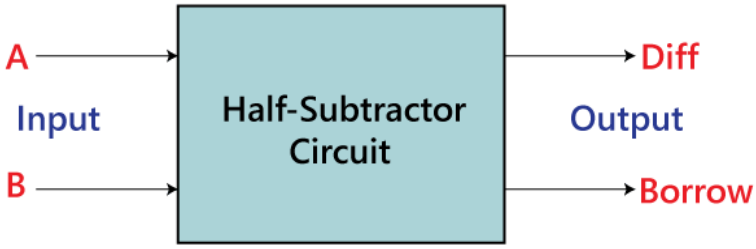
$$D = A\bar{B} + \bar{A}B$$

$$B_0 = \bar{A}B$$

*Logic Circuit:



*Block diagram of half subtractor:



Full Subtractor:

*The logic circuit that performs subtraction of three binary bits at a time is known as full subtractor.
*It has three inputs denoted by symbols:A,B,C.
It has two outputs denoted by symbols :D(Difference),B_i (Borrow). *Truth Table of full subtractor:

INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

*Boolean expression:
Full subtractor has two Boolean expression, one for output D and another for output B_i.

$D = \overline{A}\overline{B}B_{in} + \overline{A}B\overline{B}_{in} + A\overline{B}\overline{B}_{in} + ABB_{in}$

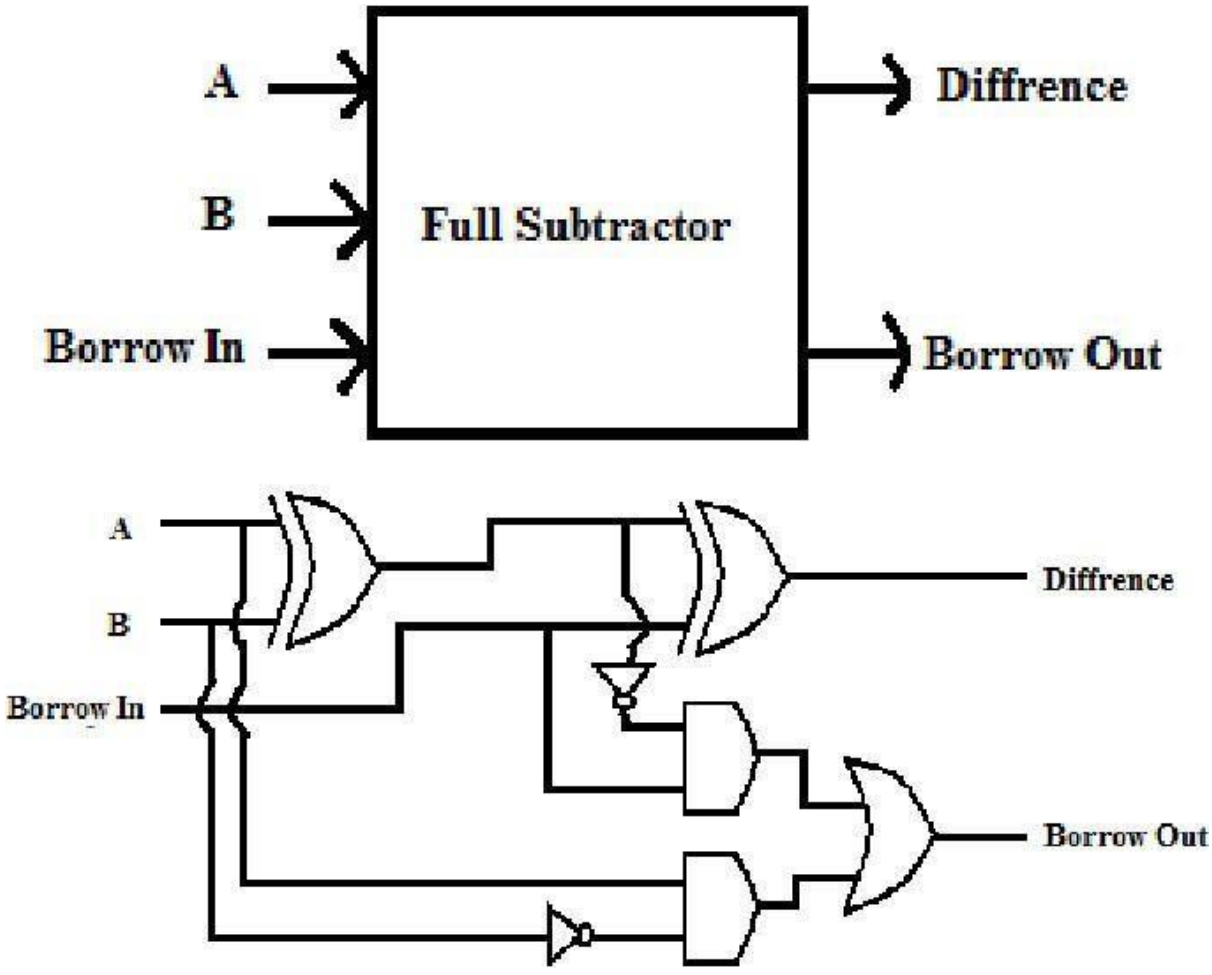
$B_{out} = \overline{A}\overline{B}B_{in} + \overline{A}B\overline{B}_{in} + \overline{A}BB_{in} + ABB_{in}$

$= \overline{A}(\overline{B}B_{in} + B\overline{B}_{in}) + B(\overline{B}\overline{B}_{in} + AA)$

$= \overline{A}(B \oplus B_{in}) + B(\overline{B \oplus B_{in}}).1$

$= \overline{A}(B \oplus B_{in}) + B(\overline{B \oplus B_{in}})$

*Logic circuit and block diagram :



CHAPTER-4

MULTIPLEXER

Multiplexer

*Multiplexer is a combinational circuit that selects any one data input out of many data inputs and passes to output.
*It has two types of inputs named as data inputs and selection inputs and one output.
*Basically , multiplexer is designated by M x1 or M:1 multiplexer. Where
M=2^n=Number of data inputs n=Number of selection inputs
*Multiplexer is also known as many to one selector.

4:1 Multiplexer

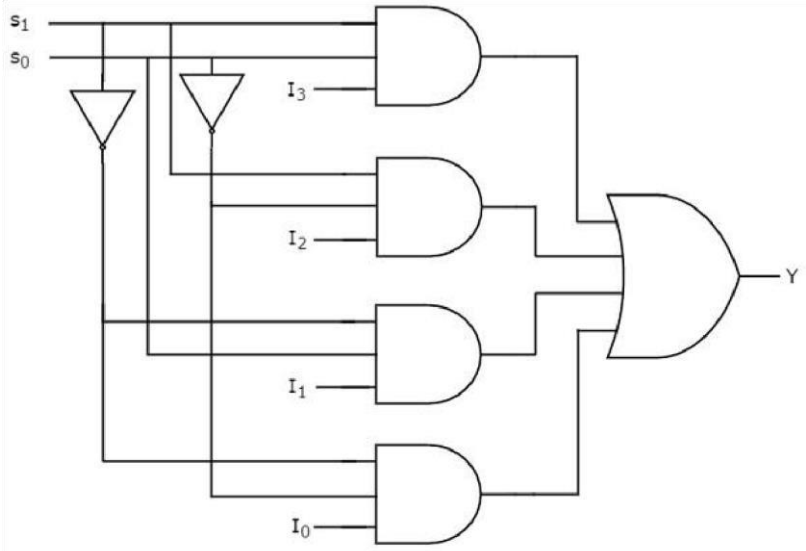
*4:1 multiplexer selects any one data input out of 4 data inputs and passes to the output.
*It has 4 number of data inputs, 2 number of selection input lines and one output line.
*It consists of four AND gates, two NOT gates and one OR gate.
*The four data inputs are denoted by I_0, I_1, I_2, I_3.
Selection inputs are denoted by S_0, S_1.
Output is denoted by Y.
*The data input to be selected and appeared on output line ,depends on the bit value of selection inputs as shown in truth table. *Truth Table:

Truth Table

S0	S1	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

So, final equation,
Y = S0'.S1'.I0 + S0'.S1.I1 + S0.S1'.I2 + S0.S1.I3

*Logic Circuit:



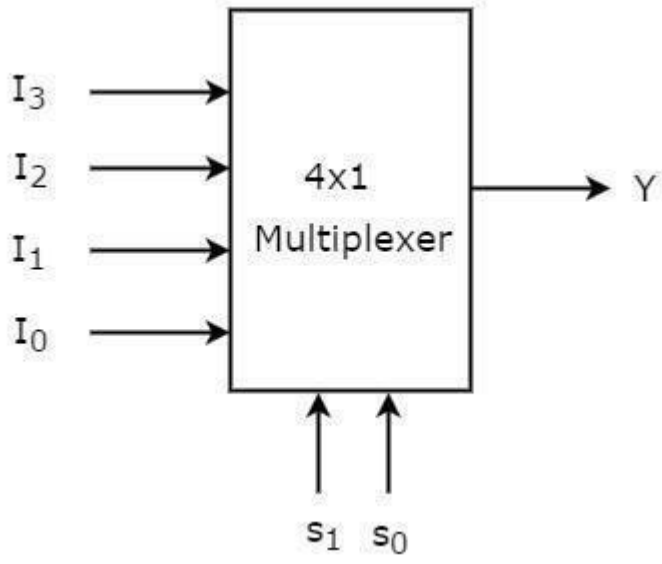
*Working Principle:

a-If S_1S_0=00, first AND gate output=I_0 and other three AND gate output=0.
Hence Y=I_0+0+0+0=I_0.

*If S_1S_0=01, second AND gate output=I_1 and all other AND gate output=0.
Hence, Y=0+I_1+0+0=I_1.

* If S_1S_0=10, third AND gate output=I_2 and all other AND gate output=0.
Hence, Y=0+0+I_2+0=I_2.

*If S_1S_0=11, fourth AND gate output=I_3 and all other AND gate output=0.
Hence, Y=0+0+0+I_3=I_3. *Block Diagram:

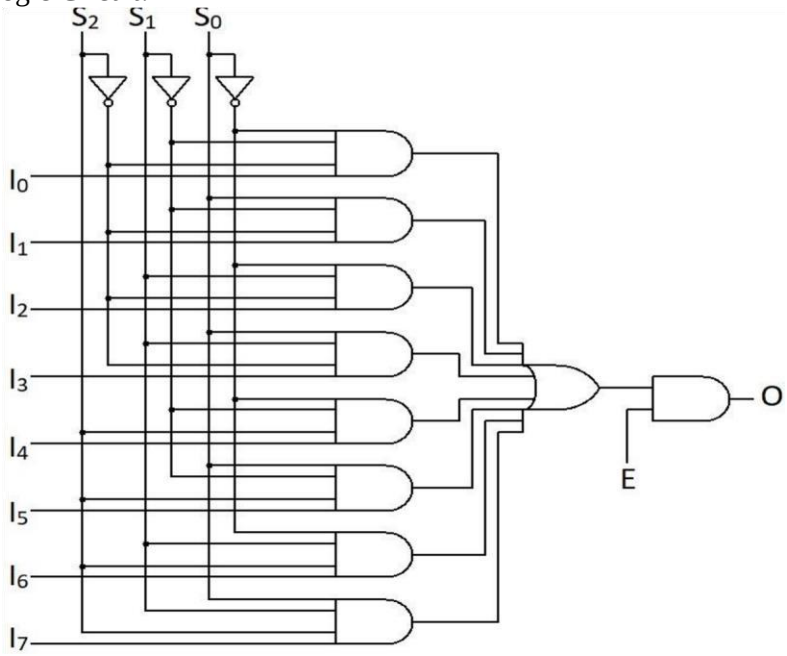


8:1 Multiplexer

*8:1 multiplexer selects any one data input out of 8 data inputs and passes to the output.
*It has 8 number of data inputs, 3 number of selection input lines and one output line.
*It consists of eight AND gates, three NOT gates and one OR gate.
*The eight data inputs are denoted by $I_0, I_1, I_2, I_3, I_4, I_5. I_6. I_7$.
Selection inputs are denoted by S_0, S_1, S_2.
Output is denoted by Y.
*The data input to be selected and appeared on output line ,depends on the bit value of selection inputs as shown in truth table. *Truth Table:

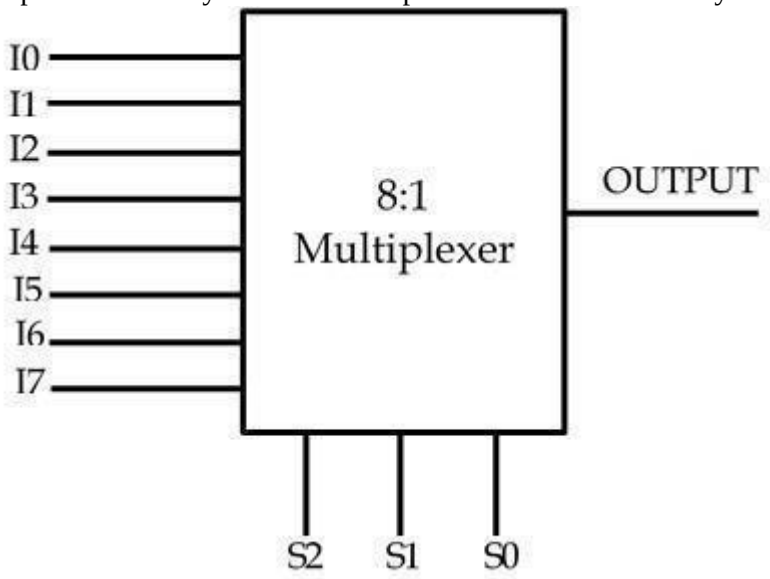
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I7

*Logic Circuit:



*Working Principle: a-If $S_2 S_1 S_0$=000, first AND gate output=I_0 and other three AND gate output=0.
Hence $Y=I_0+0+0+0+0+0+0+0=I_0$.
*If $S_2S_1S_0$=001, second AND gate output=I_1 and all other AND gate output=0.
Hence, $Y=0+I_1+0+0+0+0+0+0=I_1$.
* If $S_2 S_1S_0$=010, third AND gate output=I_2 and all other AND gate output=0.
Hence, $Y=0+0+I_2+0+0+0+0+0=I_2$.
*If $S_2 S_1S_0$=011, fourth AND gate output=I_3 and all other AND gate output=0.
Hence, $Y=0+0+0+I_3+0+0+0+0=I_3$.
*If $S_2 S_1S_0$=100, fifth AND gate output=I_4 and other three AND gate output=0.
Hence $Y=0+0+0+0+I_4+0+0+0=I_4$.
*If $S_2S_1S_0$=101, sixth AND gate output=I_5 and all other AND gate output=0.
Hence, $Y=0+0+0+0+0+I_5+0+0=I_5$.
* If $S_2 S_1S_0$=110, seventh AND gate output=I_6 and all other AND gate output=0.
Hence, $Y=0+0+0+0+0+0+I_6+0=I_6$.
*If $S_2 S_1S_0$=111, eighth AND gate output=I_7 and all other AND gate output=0.
Hence, $Y=0+0+0+0+0+0+0+I_7=I_3$.

It is clear that for each combination of selection input,only one AND gate output is active and all other AND gate output is inactive.Finally due OR operation the output of active AND gate appears at the output of multiplxer.In this way the selection input combination selects any one to be available at output. *Block Diagram:



CHAPTER-5

DEMULTIPLEXER

Demultiplexer:

*Demultiplexer is a combinational circuit that allows the single data input on any one output lines out of many output lines.

*The information is received through the single input and directed to be available on one output line.

*The output line on which data input will be available depends on the bit value of selection inputs .

*It has two types of inputs named as data input(single) and selection inputs and number of output lines..

*Basically , demultiplexer is designated by 1xM or 1:M multiplexer. Where

M=2^n=Number of output lines n=Number of selection inputs

*Demultiplexer is also known as one to many selector.

1:4 Demultiplexer

*1:4 de multiplexer selects any one output line out of 4 outputs and passes the single data input on the selected output line.

*It has single data inputs, 2 number of selection input lines and four output lines.

*It consists of four AND gates, two NOT gates.

*The single data input line is denoted by D

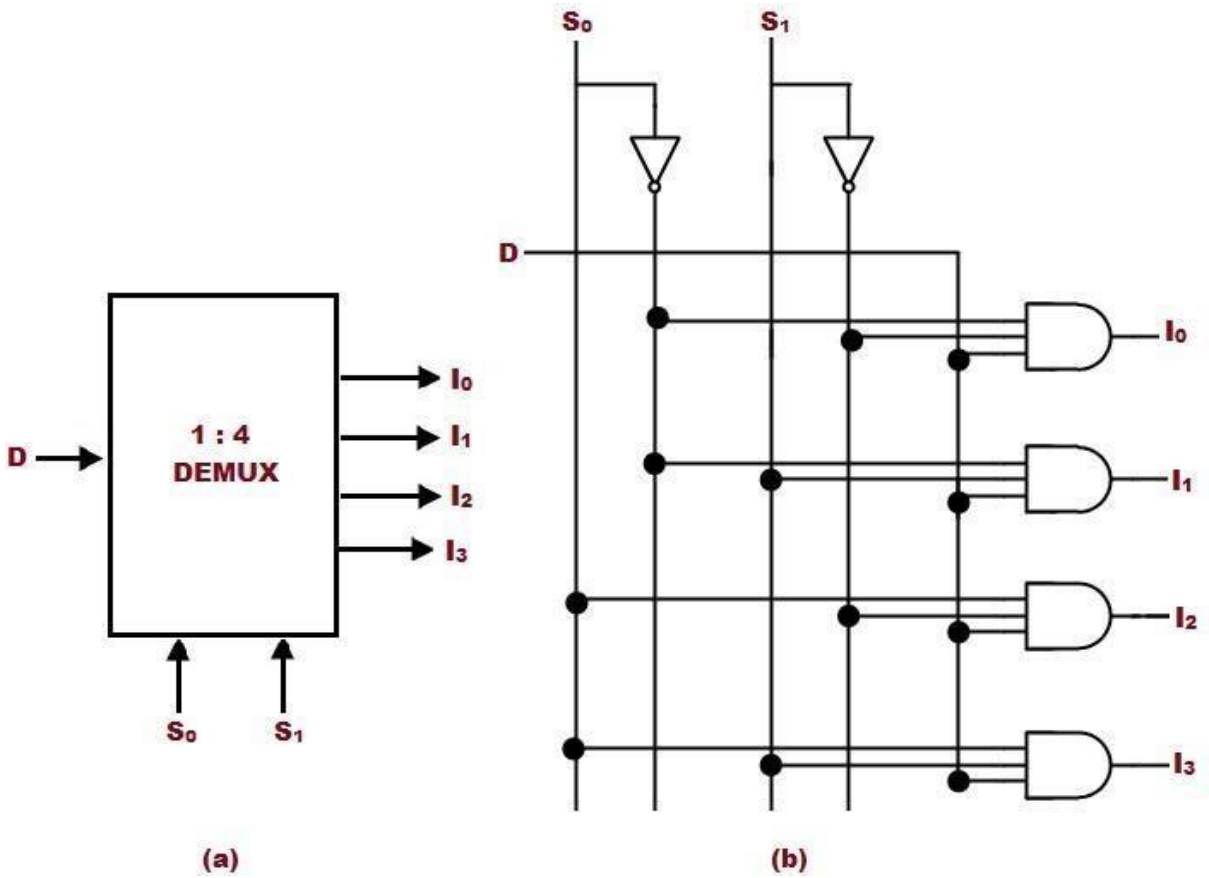
Selection inputs are denoted by S_0,S_1.

Four output lines are denoted by I_0,I_1,I_2,I_3

*The output lines over which data input will appear ,depends on the bit value of selection inputs as shown in truth table. *Truth Table:

S_1	S_0	I_3	I_2	I_1	I_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

*Logic Circuit:



*Working Principle: a-If S_1S_0=00, first AND gate output=D and other three AND gate output=0.
Hence Y_0=D
*If S_1S_0=01, second AND gate output=D and all other AND gate output=0.
Hence, Y_1=D
* If S_1S_0=10, third AND gate output=D and all other AND gate output=0.
Hence, Y_2=D
*If S_1S_0=11, fourth AND gate output=D and all other AND gate output=0. Hence, Y_3=D
*Block Diagram:

CHAPTER-6

FLIP FLOPS

Flip-Flop

*Flip-flop is a simple two state device that stores single bit of information.
*It acts as an important memory element or single bit memory.
*It is a sequential type of digital device.
*The flip-flop is made up of an assembly of logic gates.Basically, the logic gates have no storage capability.But when the several logic gates are connected in proper ways, that permit information to be stored.
*Flip- flops can have one or more inputs and two outputs.One of its output is for normal value and another is for complemented value. The inputs are used to change the output state momentarily.
*It is also known as latches or bistable multivibrator.The term latch is used for certain type of flip flops.

Application of flip-flops

*The flip-flops are used extensively as a memory cell in static random access memory(SRAM).
*It can be used to synchronise the effect of an asynchronous input.
*It is used for data or information storage and transfer operation.
*It can be used for serial data transfer e.g. in shift register.
*The frequency division can be achieved by using flip-flops.
*It can be used for counting purpose such as number of events or counting sequences,number of pulses in a signal e.g. in counter.

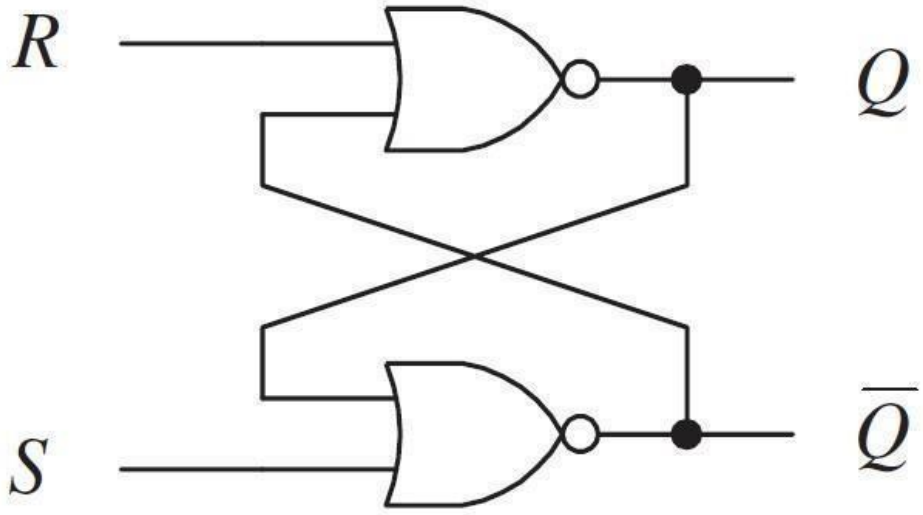
Types of Flip-flops

Flip-flops are broadly classified into six types:
a-Unclocked RS-Flip-flop/RS Latch
b-Clocked RS-Flip-flop c-D-Flipflop
d-T-Flip-flop e-JK-Flip-flop f-
Master Slave JK-flip-flop

R-S Latch

*RS latch is the simplest latch that stands for Reset-Set latch.
*R-S latch is the basic unclocked flip- flop and all other flip- flops are derived from this flip-flop.

*It has two inputs named as Set(S) , Reset(R) and two outputs named as Q and $\overline{Q}$. The two outputs are complement of each other. Hence if one is in state 1 another is in state 0.
*It can be realized by using either two NOR gates or two NAND gates or basic gates. But NOR or NAND realization is most commonly used.
*Logic circuit using NOR gate:



*The two NOR gates are cross coupled so that output of one NOR gate is connected to input of other NOR gate and vice versa.
*Its operation is based on the NOR gate functional property i.e. , if any one input of NOR gate is at logic 1, it forces its output to logic zero immediately.

*Truth table

R	S	Q
0	0	Q_n(No change/last state)
0	1	1
1	0	0
1	1	d(undefined)

*Operation of RS Latch
Its operation can be discussed for the four input possibilities as below:

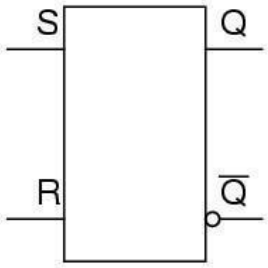
*If R=0,S=0, assuming that initially Q=1 and$\overline{Q}$ =0.With$\overline{Q}$ =0, both inputs to NOR gate 1 are at logic 0.So its output, Q is at logic 1,With Q=1, NOR gate 2 output$\overline{Q}$ =0.So the flip-flop output has no change i.e. it shows the last state or before state or initial state.

*If R=0,S=1, S input of NOR gate 2 is at logic 1, hence its output$\overline{Q}$ =0 which indicates that both inputs of NOR gate 1 are at logic 0.Hence the output Q=1 and the flip-flop is said to be in 'Set' condition. *When R=1,S=0, R input of the NOR gate 1 is at logic 1, hence its output Q=0 .This indicates that both

$\overline{\text{inputs}}$ of NOR gate 2 are now at logic 0, so its output$Q$ =1.The flip-flop is said to be in 'Reset' state. *If R=1, S=1, these input state force the outputs of both NOR gates to logic 0 i.e.,Q=0 and$Q$ =0 which

$\overline{\text{is}}$ undefined or ambiguous condition because Q and $\overline{Q}$ are complement of each other, Q=Q condition is impossible. Hence care should be taken such that flip-flop never accepts this state of inputs in normal operation..

*Block diagram:

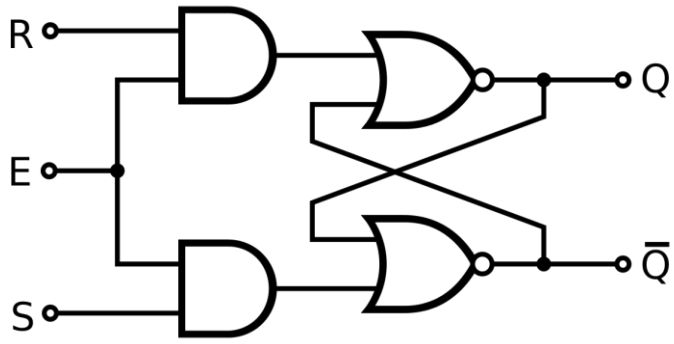


Clocked RS Flip-flop

*Clocked R-S flip-flop is an extension of R-S flip-flop.

*This flip-flop uses an external pulse type signal known as clock input signal(clk or E).The action of flipflop depends on this clock input.

*If clock input is absent, all the input combinations are blocked and the output remains unchanged. *For proper working of flip-flop, clock input is kept at high state(=1) for all input combinations. *Logic Diagram:



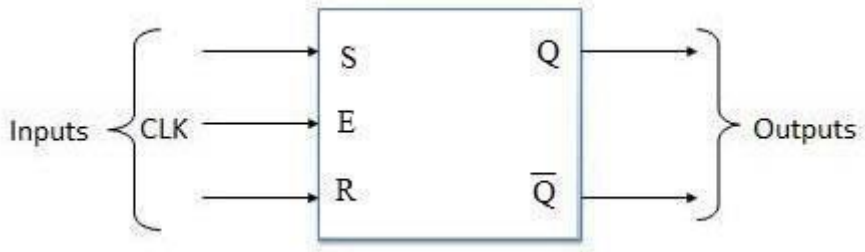
*Truth Table:

clk	R	S	Q
0	x	x	No change
1	0	0	Q_n(last state)
1	0	1	1
1	1	0	0

1	1	1	d(undefined)

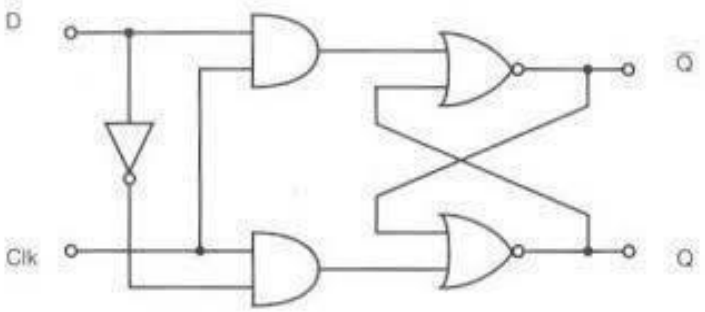
*If clock input, clk=0, whatever may be the state of inputs R and S,flip-flop output has no change i.e. it remains in initial condition.
*If clk=1,R=0,S=0, output $Q=Q_n$ i.e. last state or before state.
*If clk=1,R=0,S=1, output Q=1.The flip-flop is said to be in 'SET' state.
*If clk=1,R=1,S=0, output Q=0.The flip-flop is said to be in 'RESET' state.

*If clk=1,R=1,S=1, output Q is undefined or ambiguous condition.This can provide $Q=\overline{Q}$ which is impossible. Hence care should be taken such that flip-flop never accepts this state of inputs. *Block diagram:



D Flip-Flop

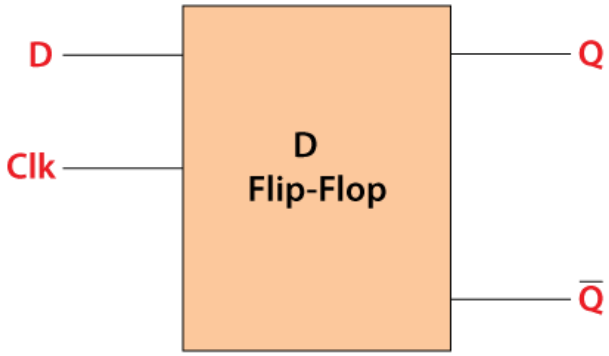
*D flip- flop avoids the undefined condition of R-S flip-flop that happens for input state R=1,S=1.
*This flip-flop is the modified version of clocked R-S flip-flop.
*The modification is basically applied at input line in such a way that the input state R=1, S=1 never happens. *Logic Diagram



*It consists of clocked R-S flip-flop and one NOT gate. It has one input named as D that is directly connected to S input. But this D input is applied to R through a NOT gate. This allows D=S, R=complemented value of D. Hence the use of NOT gate avoids the ambiguous or undefined condition or R=1,S=1. *Truth Table

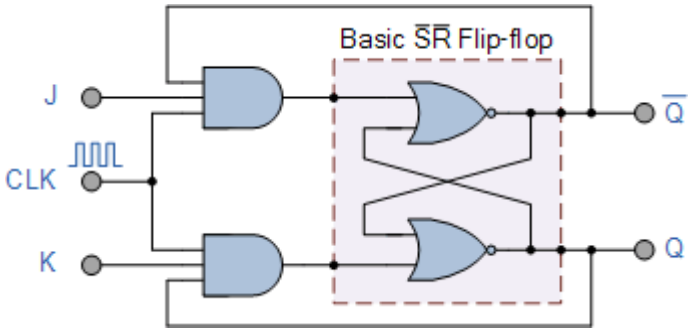
clk	D	Q
0	x	No change/last stateQ_n
1	0	0
1	1	1

*If clock input clk=0, whatever may be the state of D input, output has no change that means the output remains in last state.
*For normal working of D flip-flop, clock input should be at high state i.e. clk=1.
*When clk=1, D=0, the inputs R=1, S=0.Hence the present output Q=0. The flip-flop is said to be in 'RESET' state.
* When clk=1, D=1, the inputs R=0, S=1.Hence the present output Q=1. The flip-flop is said to be in 'SET' state.
*Block diagram:



JK Flip-flop

*JK flip-flop is a modified version of clocked RS flip-flop, that avoids the ambiguous or undefined condition of RS flip-flop happening at input state R=1, S=1.
*It consists of RS flip-flop and two AND gates.
*Logic Circuit



*It has two inputs named as J and K, two outputs named as Q and $\overline{Q}$, one clock input 'clk'. Truth Table

clk	J	K	Q(output)
0	x	x	No change
1	0	0	No change
1	0	1	0
1	1	0	1
1	1	1	Q_n

*Operation:

1-When clock input, clk=0, whatever may be the state of inputs J and K,flip-flop output has no change i.e. it remains in initial condition or last state.

2- If clk=1,J=0,K=0, both AND gates are disabled and both AND gates outputs are logic 0.These outputs are now act as R=0, S=0.Hence output has no change or last state or before state or $Q=Q_n$.

3-If clk=1,J=0,K=1, the lower AND gate is disabled and upper AND gate abled.Accordingly the upper AND gate output is logic 1 and lower AND gate output is logic 0.Hence R=1,S=0, output Q=0.Noe the flip-flop is said to be in 'Reset' state.

4- If clk=1,J=1,K=0, the lower AND gate is abled and upper AND gate disabled.Accordingly the upper AND gate output is logic 0 and lower AND gate output is logic 1.Hence R=0,S=1, output Q=1.Now the flip-flop is said to be in 'Set' state.

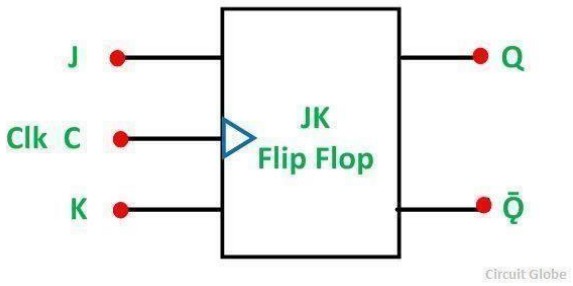
5-If clk=1,J=1,K=1,The flip-flop can be either in set or reset state depending upon the initial state.If

initially flip-flop state is in set Q=1,$\overline{Q}$ =0, the lower AND gate is disabled and upper AND gate abled.Accordingly the upper AND gate output is logic 1 and lower AND gate output is logic 0.Hence

R=1,S=0, present output Q=0, $\overline{Q}$ =1which is reset state .Similarly if initial state is in reset state, then application of J=1,K=1 will reset the flip-flop. This implies shows complement of last state or initial state.

If the last state main output is designated by Q_n the present state main output can be designated by $\overline{Q_n}$

.Hence for easy understanding we can define this state operation as when J=1,K=1, Q=$\overline{Q_n}$ i.e. output of flipflop toggles from set to reset and vice versa at positive edge of next clock. *Block diagram:



T Flip-Flop

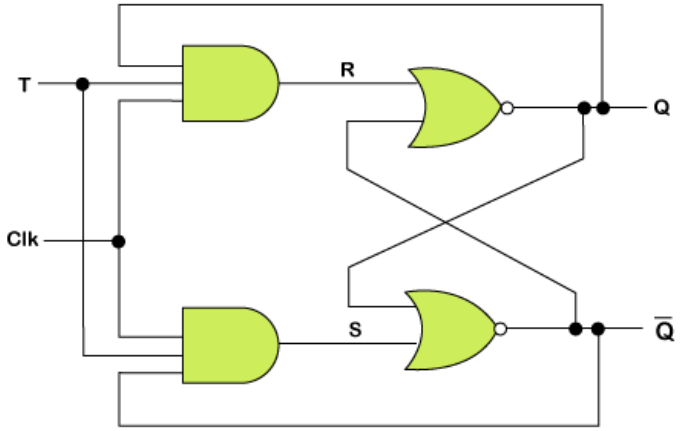
*The T flip-flop is a modification of JK flip-flop. It is made from JK flip-flop by connecting its J and K input terminals together and a single input terminal is formed known as T input.

Hence in this case J=K=T.

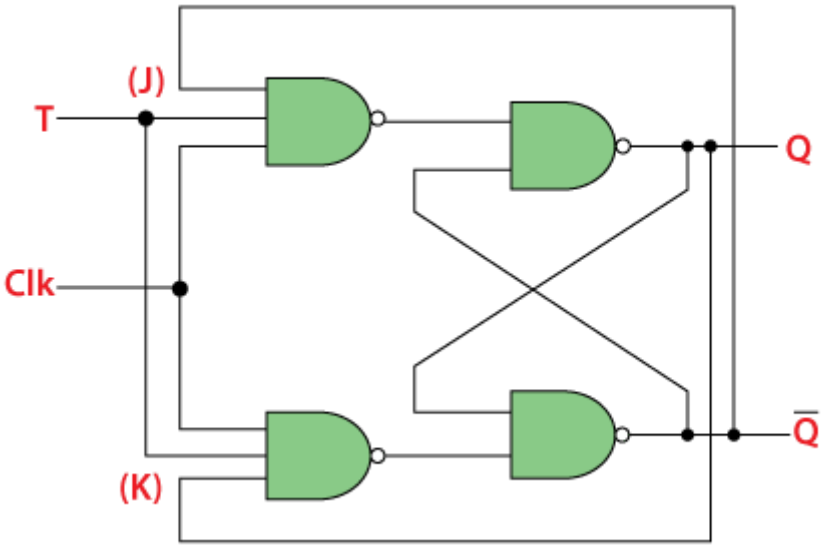
*This flip-flop is also known as toggle flip-flop. *Truth Table

Input	Before state	Present State
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

*Logic circuit of clocked T flip-flop



(a) T Flip –Flop using NOR Gate



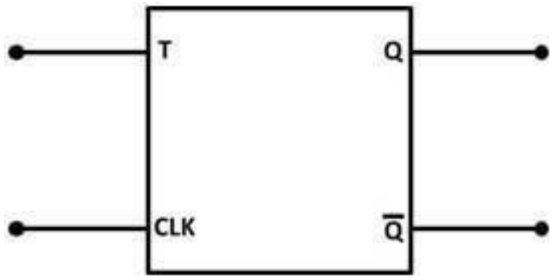
(b) T Flip-flop Using NAND gate

*Operation

1-When clock input clk=0, both AND gates are disabled whatever may be the value of T.Hence both AND gates outputs are 0.The output of the circuit is same as the last state output i.e. no change in output and present state output $Q_{n+1}=Q_n$(before state)=0.

2-When clk=1, T=0 both AND gates are disabled.Hence both AND gates outputs are 0.The output of the circuit is same as the last state output i.e. no change in output and present state output $Q_{n+1}=Q_n$(before state).If before state is at 1, present state is 1 and vice versa. Accordingly as per truth table $Q_{n+1}=Q_n=1$.

3If clk=1,T=1, implies R=S=1. The output of both AND gate depends upon the before state.If before state $Q_n=0$, lower AND gate is abled and upper AND gate is disabled producing present state output $Q_{n+1}=1$.Similarly if the before state $Q_n=1$, present state is 0.Finally the output toggles with output complement of the previous state for T=1. *Block diagram/Logic Symbol:



*Basically a clock pulse has two transitions: a-transition from 0 to 1(logic HIGH/positive level), btransition from 1 to 0(logic LOW/negative level).If the flip-flop changes state only at signal transition(from 0 to 1 or from 1 to 0) and disabled during rest the rest of the clock pulse duration is called level triggering flip-flop and the process is known as level triggering.

*If the flip-flop changes its state at positive level of clock input , it is called as positive level triggering flip-flop.

*If the flip-flop changes its state at negative level of clock input , it is called as negative level triggering flip-flop.

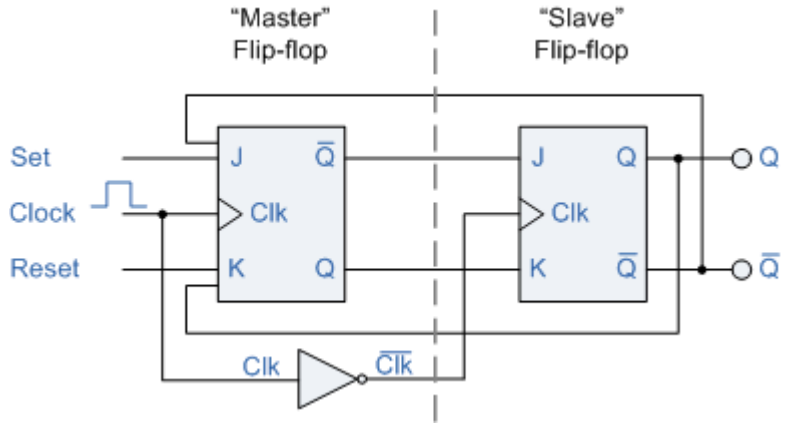
Race-arround Condition

*The ambiguous condition happening due to long duration of clock input is called race around condition.
*In case of flip-flop, let Δt=Total propagation delay of gates t_p=Duration of clock pulse If $t_p > \Delta t$ i.e.if clock pulse duration is more, more than one transitions in output state happens for a single change in input state.Hence the output oscillates between 0 and 1 and finally the output will be ambiguous.This oscillation from zero to 1 is called race around condition.
*Race arround condition can be avoided by adjusting t_p less than Δt.

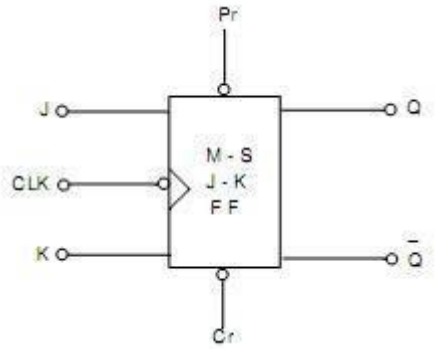
*Master-slave J-K flip-flop is the best flip-flop to avoid race-arround condition. **Master-Slave J-K Flip-flop**

*The flip-flop that avoids race around condition of J-K flip-flop is known as master-slave J-K flip-flop.
*This flip-flop is a modified version of simple J-K flip-flop.It consists of two J-K flip-flops connected in cascade as shown in logic diagram.Hence the output of first flip-flop acts as input of second flip-flop.The first flip-flop is known as master and the second flip-flop is known as slave.
*The working of this flip-flop is similar to simple J-K flip-flop.But the difference is in clock pulse triggering process.In this case both master and slave are edge triggered J-K flip-flop.
Master is a positive edge triggered J-K flip-flop i.e.for each combination of input, it changes state at positive going edge of clock pulse without using the entire duration of clock pulse.
Similarly, the slave is a negative edge triggered J-K flip-flop i.e.for each combination of input, it changes state at negative going edge of clock pulse without using the entire duration of clock pulse.
*A single clock source is used to supply clock pulse input to both master and slave flip-flop.But master is getting directly the clock input and slave is getting complement of clock input due to NOT gate.
*As a result both flip-flops can not change state simultaneously at the same clock instant.Master responds to its J and K inputs before the slave.
*If clock input, clk=0, whatever may be the state of inputs J and K,flip-flop output has no change i.e. it remains in initial condition.
*If clk=1,J=0,K=0, output $Q=Q_n$ i.e. last state or before state.
*If clk=1,J=0,K=1, output of master changes to 0 at the positive edge of clock input.The 0 output of master acts as J input of slave.So the input state of slave is J=0.K=1 that produces output Q=0(slave output) at negative edge of clock input i.e. at negative edge of clock input the slave copies the action of master.Now the flip-flop is said to be in 'RESET' state.
* If clk=1,J=1,K=0, output of master changes to 1 at the positive edge of clock input.The 1 output of master acts as J input of slave.So the input state of slave is J=1.K=0 that produces output Q=1(slave output) at negative edge of clock input i.e. at negative edge of clock input the slave copies the action of master.Now the flip-flop is said to be in 'SET' state.
*If clk=1, J=1, K=1 the master toggles or changes state on the positive edge of clock input and the slave toggles on the negative edge of clock input.
*It is concluded that no matter what the master does, the slave copies it.If the master sets, the slave sets, if the master resets, the slave resets.

*Logic Circuit



*Block diagram :



CHAPTER-7

SHIFT REGISTER

Shift Register

*A digital device that stores and shifts digital data bits is known as shift register.Simply it can be called as register.

*It operates in a synchronous mode with a clock pulse.

*It is a sequential type digital circuit whose output depends on present and past input states.

*It has two activities: a-Loading(Shifting in) b-Unloading(Shifting out) *The process of shifting the data bits into the register contents is known as loading.

*The process of shifting out the contents of register is known as unloading.

*It consists of number of flip-flops connected in series..RS, JK or D flip-flops can be used for implementing the registers. Mostly D-flip-flops are used for constructing registers.

*Each flip flop can store 1-bit of information i.e. 0 or 1.Therefore a register can contains a series of bits which can be termed as a word or a byte. Accordingly, for storing an n-bit word n-flip-flops are required.
*Registers are basically designated by n-bit register, where n=number of bits in input data =number of flip flops used in register.

Types Of Shift Register

Shift Registers are broadly classified into four classes named as follows:

1-Serial-In-Serial Out(SISO) shift Register

2-Serial-In-Parallel-Out(SIPO) shift Register

3-Parallel-in-Serial-Out(PISO) shift Register

4ParallelIn-Parallel-Out(PIPO) Shift Register **Q-Briefly Discuss various shift registers.**

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

*The shift register that shifts all data input bits in and out serially one bit after another starting from LSB(Least Significant Bit), is known as serial-in-serial-out shift register.

*For shifting each bit one clock pulse is required. As an illustration in case of 4-bit SISO, 4 clock pulses are required for shifting in 4bits of data and another 4 clock pulses are required for shifting out 4bits of data.

*It has one input line and one output line.

2-Serial-In-Parallel-Out(SIPO) shift Register

*The shift register that shifts in all the input data bits serially one bit after another starting from LSB, but shifted out all the bits simultaneously is known as serial-in-parallel-out shift register.

* For shifting in or loading each bit, one clock pulse is required.But one clock pulse is required for shifting out or unloading all data bits. As an illustration in case of 4-bit SIPO, 4 clock pulses are required for shifting in 4bits of data and one clock pulse is required for shifting out 4bits of data. *It has one input line and n number of output lines, where n is the number of data bits **3-Parallel-in-Serial-Out(PISO) shift Register** *The register that loads or shifts in all input data bits simultaneously,but unloaded or shifted out data bits serially one bit after another is known as parallel-in-serial-out shift register.

*One clock pulse is required for loading all input data bits, but for unloading each bit one clock pulse is required. As an illustration in case of 4-bit PISO, one clock pulse is required for shifting in 4bits of data and 4 clock pulses are required for shifting out 4 bits of data.

*It has as much input lines as the number of bit in input data but one output line.

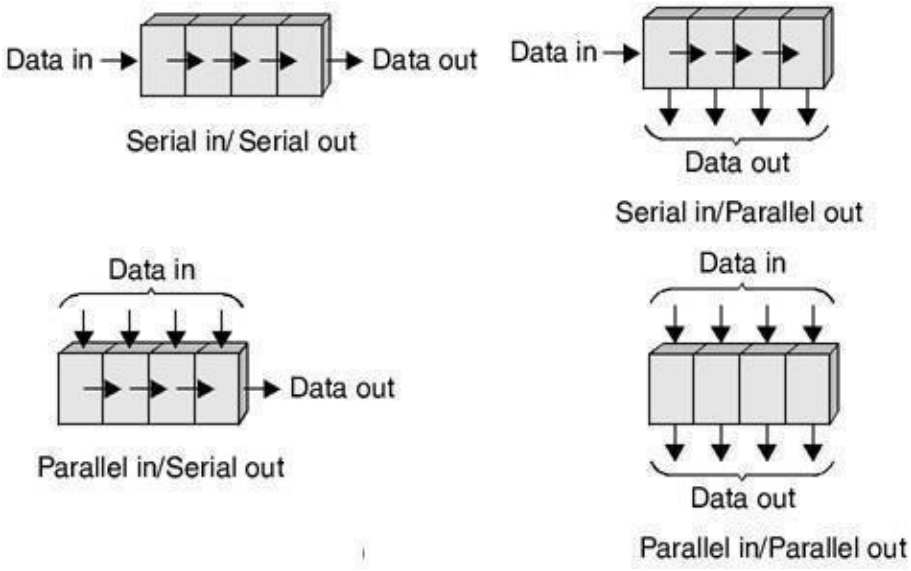
4-Parallel-In-Parallel-Out(PIPO) Shift Register

*The register that loads or shifts in all input data bits simultaneously,but unloaded or shifted out all data bits simultaneously is known as parallel-in-parallel-out shift register.

*One clock pulse is required for loading all input data bits and for unloading one clock pulse is required. As an illustration in case of 4-bit PIPO, one clock pulse is required for shifting in 4-bits of data and one clock pulses is required for shifting out 4 bits of data.

*It has as much input and output lines as the number of bit in input data.

*Block diagram of shift registers:



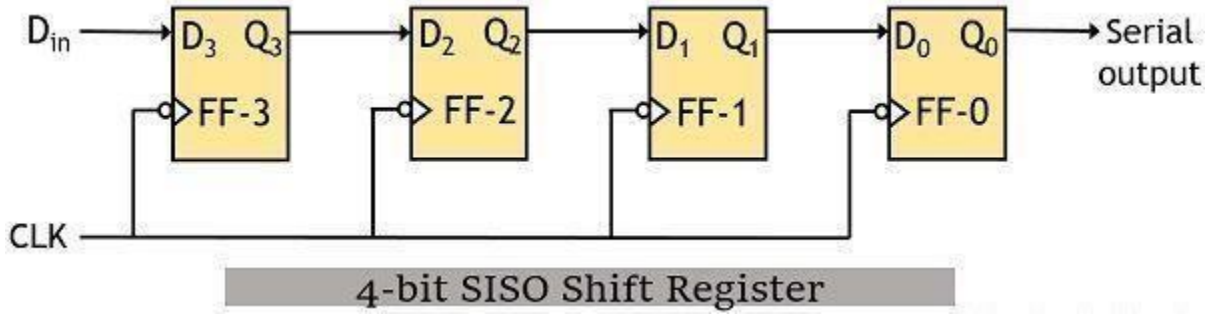
4-bit SISO Shift Register

*The register that shifts in or shifts out 4-bit data serially one bit after another is known as 4-bit SISO register.

*In this register the 4 input bits are loaded serially one bit after another starting from LSB. The loaded 4bits are shifted out serially one bit after another.

*For each bit loading/unloading one clock pulse is required. Hence 4 clock pulses are required for loading 4-bit data input and another 4 clock pulses are required for unloading 4-bit of data. * The practical application of SISO register is to act as a delay element.

*Logic Circuit:



*It consists of four D-flip flops connected in series.Each flip flop is triggered synchronously by clock input pulse.Each flip-flop changes state at negative edge of clock input.

*It has one input named as D_{in} and one output named as $D_{out}=Q_0$.

*Initially all the flip-flops are reset to 0 by applying clear input.Hence the contents of register is$Q_3Q_2Q_1Q_0=0000$.

*Suppose the purpose is to load 4-bit binary number 1101 into register.

Step-1

First the LSB bit 1 is applied at D_{in}. Then at the negative edge of first clock pulse,flip-flop 3 set to 1,but other flip-flops show no change their input $D_2=0$, $D_1=0$, $D_0=0$.

Hence the contents of register after application of first clock pulse are $Q_3Q_2Q_1Q_0=1000$.

Step-2

When second clock pulse negative edge reaches, $D_{in}=0$,$D_3=0$, $D_2=1$(output of FF-3),$D_1=0$,$D_0=0$.After negative edge of second clock input, contents of register are $Q_3Q_2Q_1Q_0=0100$.

Step-3

When third clock pulse negative edge reaches, $D_{in}=1$,$D_3=D_{in}=1$, $D_2=0$(output of FF-3),$D_1=1$,$D_0=0$.After negative edge of third clock input, contents of register are Q3Q2Q1Q0=1010.

Step-4

When fourth clock pulse negative edge reaches, $D_{in}=1$,$D_3=D_{in}=1$, $D_2=1$(output of FF-3),$D_1=0$,$D_0=1$.After negative edge of fourth clock input, contents of register are $Q_3Q_2Q_1Q_0=1101$.

The above discussion provides a conclusion that at each negative edge of clock, data input bits shifts from one flip-flop to next flip-flop.Finally, after four clock pulses all the input bits are shifted into the register. The contents 1101 can be shifted out by using another 4 clock pulses starting from LSB.

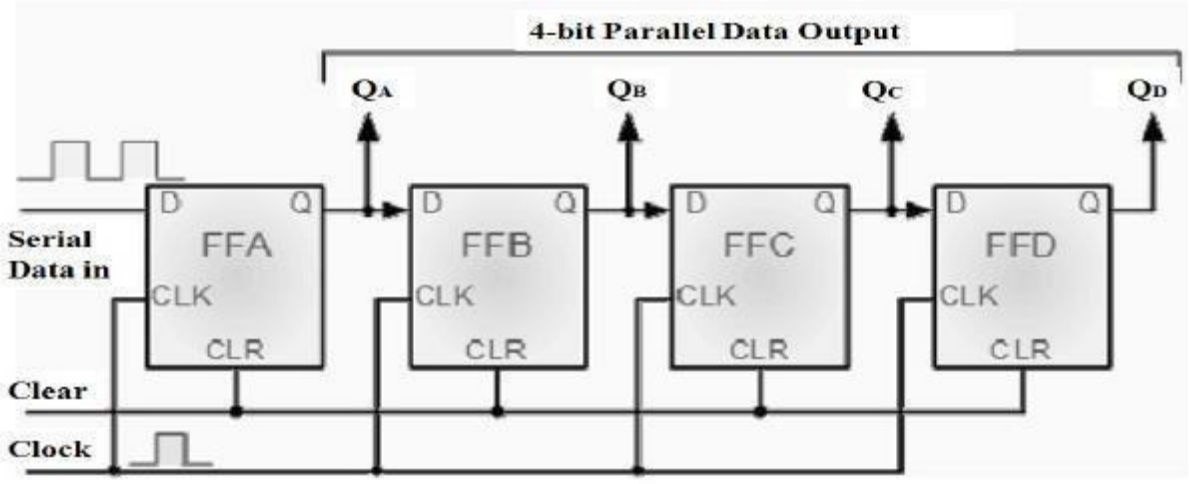
4-bit SIPO Shift Register

*The register that shifts in 4-bit data serially one bit after another and shifted out the loaded 4-bit data simultaneously is known as 4-bit SIPO register.

*Accordingly,in this register the 4 input bits are loaded serially one bit after another starting from LSB. But the loaded 4-bits are shifted out in a parallel fashion.

*For each bit loading one clock pulse is required.Hence 4 clock pulses are required for loading 4-bit data input and a single clock pulse is required for unloading 4-bit of data.

*The practical application of SIPO register is to convert serial data format on a single wire to parallel format on multiple wires.They are used in communication lines where demultiplexing of a data line into several parallel lines is required. *Logic Circuit:



*It consists of four D-flip flops connected in series.All these flip-flops are synchronous with each other since the same clock input pulse is applied simultaneously to each flip-flop.Each flip-flop changes state at positive edge of clock input.

*It has one input named as serial data in D_{in} and 4 outputs named as Q_A,Q_B,Q_C,Q_D.

*Initially all the flip-flops are reset to 0 by applying clear input.Hence the contents of register is $Q_AQ_BQ_CQ_D$=0000.

*Suppose the purpose is to load/shift in 4-bit binary data 1101 into register.

Step-1

First the LSB bit 1 is applied at D_{in}. Then at the positive edge of first clock pulse,flip-flop A set to 1,but other flip-flops show no change because their input D_B=0, D_C=0, D_D=0.Basically the 0 at input of fourth flip-flop D_D will be shifted to output Q_D,0 at input of third flip flop D_C shifted to its output Q_C, the 0 at input of second flip-flop D_B shifted to its output Q_B and finally, the 1 at input of first flip-flop D_A(=D_{in}) shifted to its output Q_A.

Hence the contents of register after application of first clock pulse are $Q_AQ_BQ_CQ_D$=1000. Similarly , the shifting process for other 3-bits can be continued. Step-2

When second clock pulse positive edge reaches, D_{in}=0,D_A=0, D_B=1(output of FF-A),D_C=0,D_D=0.After positive edge of second clock input, contents of register are $Q_AQ_BQ_CQ_D$=0100.

Step-3

When third clock pulse positive edge reaches, D_{in}=1,D_A=D_{in}=1, D_B=0(output of A),D_C=1,D_D=0.After positive edge of third clock input, contents of register are $Q_AQ_BQ_CQ_D$=1010.

Step-4

When fourth clock pulse positive edge reaches, D_{in}=1,D_A=D_{in}=1, D_B=1(output of A),D_C=0,D_D=1.After positive edge of fourth clock input, contents of register are $QAQ_BQ_CQ_D$=1101.

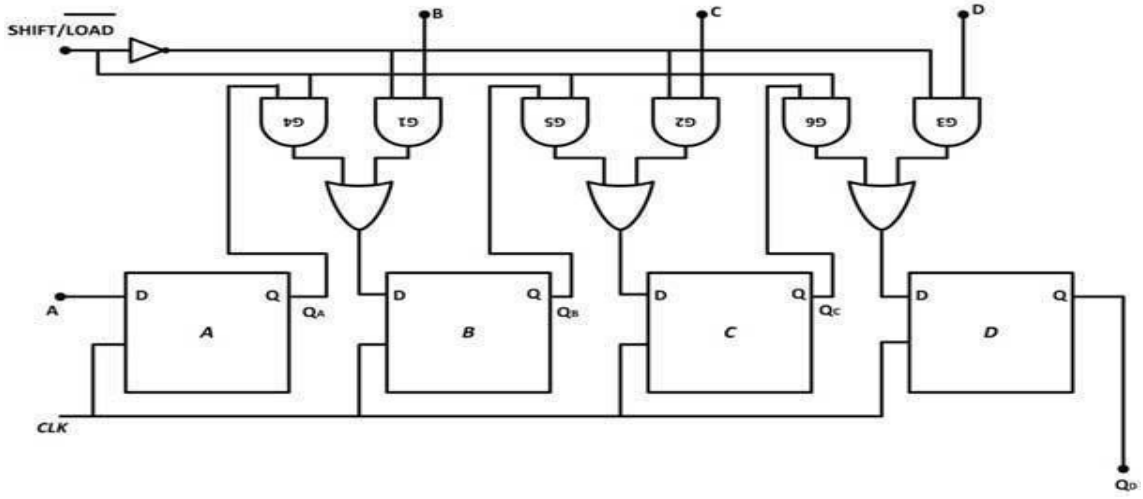
It is evident from above discussion that at each negative edge of clock pulse, data input bits shifts from one flip-flop to next flip-flop.Finally, after four clock pulses all the input bits are shifted into the register. The contents 1101 can be shifted out by using a single clock pulse.

4-bit PISO Shift Register

*The shift register that shift in 4-bit data in parallel or simultaneously and shifted out the loaded 4-bit data serially one bit after another.

*In this register the 4 input bits are loaded simultaneously by the application of a single clock pulse.But the loaded 4-bits are unloaded serially one bit after another starting from LSB by the application of 4 clock pulses.For each bit unloading one clock pulse is required.

*A PISO register used to convert parallel data to serial data. *Logic Circuit:



*It consists of four D-flip flops.All these flip-flops are synchronous with each other since the same clock input pulse is applied simultaneously to each flip-flop.Each flip-flop changes state at positive edge of clock input.
*It has 4 inputs named as A,B,C,D and one output named as Q_D
*Initially all the flip-flops are reset to 0 by applying clear input.Hence the contents of register is $Q_AQ_BQ_CQ_D$=0000.

*It has a control input named as *Shift* / $\overline{Load}$, that controls the loading and unloading of data bits. If

Shift / $\overline{Load}$ =0, all the 4 input bits will be loaded simultaneously at the positive edge of a single clock pulse.

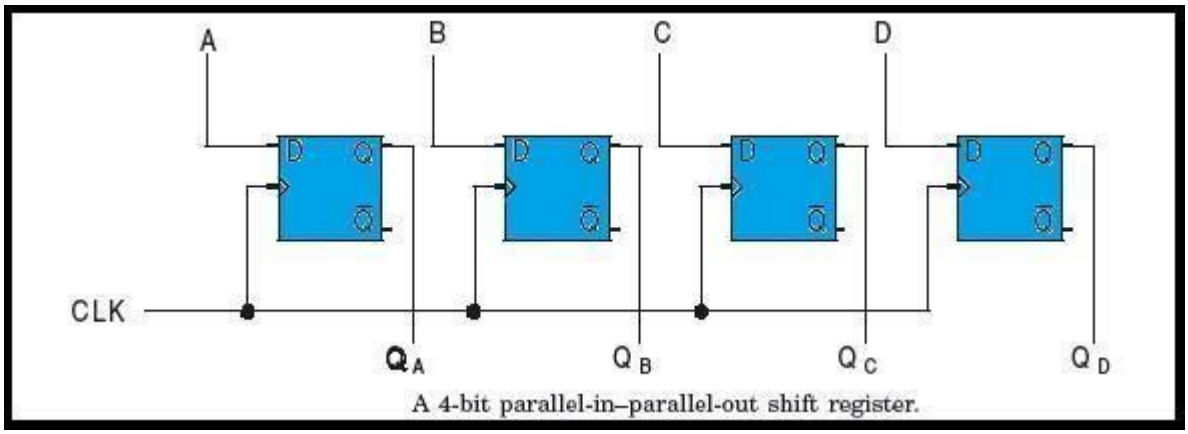
If *Shift* / $\overline{Load}$ =1, the stored 4-bits are unloaded serially using 4 clock pulses. *Suppose the purpose is to load/shift in 4-bit binary data 1101 into register.

For loading the control input condition is *Shift* / $\overline{Load}$ =0.The 4 inputs are A=1, B=1,C=0.D=1. Under this condition the AND gates G_4,G_5,G_6 are disabled because both inputs of these gates are 0 hence output of these gates are 0.But the AND gates G_1,G_2,G_3 are enabled, because one input of these gates are 1.Accordingly, the output of AND gate G1=1.B=B,AND gate G2=1.C=C, AND gate G3=1.D=D.Now the output of first OR gate=0+B=B=1=D_B,2nd OR gate=0+C=C=0=D_C, 3rd OR gate=0+D=D=1=D_D,D_A=A=1.Hence at positive edge of clock pulse, Q_A=1,Q_B=1,Q_C=0,Q_D=1.
In this way a single clock pulse shifted all input data bits simultaneously.
Basically the AND and OR gate combination at input of flip-flops acts as a multiplexer.

*For shifting out the 4-bits, *Shift* / $\overline{Load}$ =1.This makes one input of AND gates G1,G2,G3 zero thus disabling these gates and data input lines B,C,D.But enables the AND gates G4,G5,G6.Hence the output of first OR gate=Q_A=D_B,Output of second OR gate=Q_B=D_C, output of third OR gate=Q_C=D_D.The circuit is equivalent to cascade connection with output of one flip-flop connected to input of next flip-flop and A=D_A is the single data input line.This allows the data bits to shift from one stage to next stage.Then using 4 clock pulses 4 bits are shifted out serially.

4-bit PIPO Register

*The shift register that shifted in 4-bit data in parallel or simultaneously and shifted out the loaded 4-bit data in parallel or simultaneously is known as PIPO register.
*In this register the 4 input bits are loaded simultaneously by the application of a single clock pulse.The loaded 4-bits are unloaded simultaneously by the application of a single clock pulse. *A PIPO register used as a temporary stirage device.It also acts as a delay element. *Logic Circuit:



A 4-bit parallel-in–parallel-out shift register.

*It consists of four D-flip flops.All these flip-flops are synchronous with each other since the same clock input pulse is applied simultaneously to each flip-flop.Each flip-flop changes state at negative edge of clock input simultaneously.

*There are no interconnections between the individual flip-flops, since no serial shifting of the data is required.Data is given as input separately for each flip-flop and output also collected individually from each flip flop.

*It has 4 inputs named as A,B,C,D and four outputs named as Q_A,Q_B,Q_C,Q_D.

*The inputs A,B,C,D are directly passed to the data inputs D_A,D_B,D_C,D_D of the respective flip flop. The bits of the binary input are loaded to the flip flops when negative clock edge is applied.At the same time the data read in from each of the inputs is passed out at the corresponding output Q_A to Q_D

*Initially all the flip-flops are reset to 0 by applying clear input.Hence the contents of register is $Q_AQ_BQ_CQ_D$=0000.

*Suppose the purpose is to load and unload 4-bit data 1101.

*By applying one clock pulse, at its negative edge A=D_A=1,B=D_B=1,C=D_C=0, D=D_D=1.Hence Q_A=1,Q_B=1,Q_C=0,Q_D=1 will be available simultaneously.

CHAPTER-8

BINARY COUNTER

Counter

*Counter is a sequential type digital device that counts the number of occurrence of events or number of clock pulses in clock input signal.

*It can be used as a frequency divider and for generating timing sequences.

*Counter is basically designated by modulus or bit form i.e. Mod-M or n-bit counter. Examples
:
Mod-16 or 4-bit counter
Mod-8 or 3-bit counter
Mod-4 or 2-bit counter
*The relation between M and n is given by : $M=2^n$.
*Modulus(Mod) implies number of counting sequences or states of the counter and bit value implies number of bits in each counting sequence.
*A counter consists of set of flip-flops whose states change in response to clock pulses applied at the input to the counter.Clock input is the only one input to the counter.
Flip-flops are interconnected in such a manner that each clock pulse application advances one counting sequence.

*An n-bit counter has n number of flip-flops.

Applications of Counter

*Counters are used for counting events like number of clock pulses in a clock input.
*Counters are used in different arithmetic circuits.
*One important application of counter is frequency divider.
*Counter can be used for the purpose of sequential addressing in computer or microprocessor system.

Classification of Counter

Counters are fundamentally classified into three types named as :
1-Serial Counter
2-Parallel Counter
3-Combinational Counter
Serial counter alternatively called as series or asynchronous or ripple counter, Parallel counter is also named as synchronous counter.

Series Counter

*When the flip-flops are connected in such a manner that the output of one flip-flop drives next flip-flops ,that is known as series counter.
*In serial counter the main clock input from clock source is applied to the first flip-flop and the first flipflop is triggered by this clock input. But other flip-flops are triggered by the output of previous flip-flop. As an illustration, in case of a Mod-8 or 3-bit counter the number of flip-flops are 3. First flip-flop is triggered by clock input directly. The output of first flip-flop acts as the clock input of second flipflop.Hence second flip-flop is triggered by the output of first flip-flop and so on.
*In series counter all the flip-flops are not triggered or clocked simultaneously. Hence first flip-flop changes state in response to main clock input.But second flip-flop changes state in response to output of first flip-flop and similarly for other flip-flops.
*In case of series counter the triggers move through the flip-flops like a ripple in water, accordingly the counter is called as ripple counter.
*The overall propagation delay time is sum of individual delays of flip-flops used in counter.

Advantages of Series Counter

1-Construction of series counter is very simple.

2-It deals with straight forward operation.
3-It requires fewer components.
4-This type of counter can be implemented for any Mod value easily using D-flip flop or T-flip flop.

Disadvantages of Series counter

1-Since the propagation delay is more, the speed of operation is slow.These type of counters are mostly used in cases where speed of operation is not so important.
2-To obtain a truncated sequence(Mod $<2^n$) ,additional feedback logic is needed.

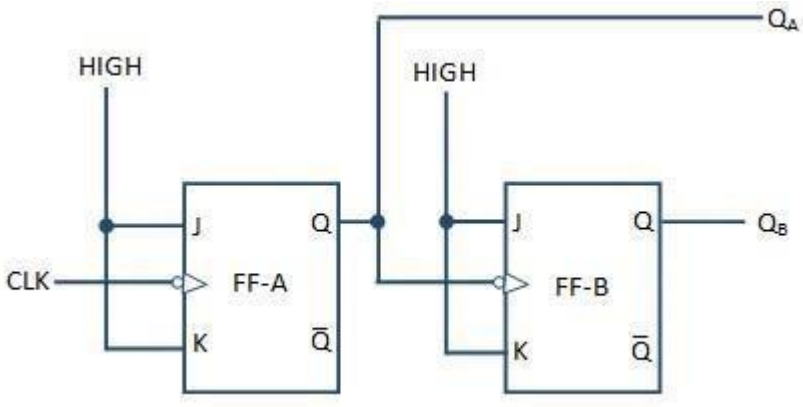
3-Counting errors may occur for high clock frequencies due to propagation delay.

Applications of Series Counters

1-Series counters are used as frequency dividers, as divide by N counters.
2-They are used in designing asynchronous decade counters.
3-They are used for low noise emission and low power applications.
4-They are used in Ring counter and Johnson counter.
5-They are used for designing any Mod counter.
Types of Series Counter
There are many types of asynchronous counters. Some of them are listed below. 1-Up-Counter 2-Down Counter
3-Ripple Up/Down Counter
4-Ripple BCD Counter/Decade Counter

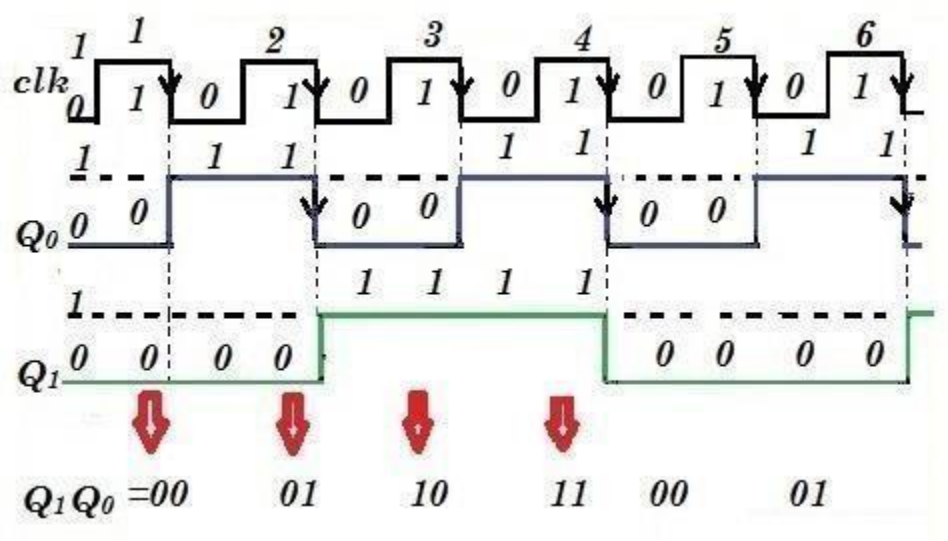
Mod-4/2-bit Series Up-Counter

*Mod-4 series up counter counts 4 number of sequences in ascending mode starting from 00 ton 11.After counting sequence 11 again it starts counting from 00.Each counting sequence has 2-bits. *Logic Diagram



*It consists of 2 master slave JK flip flops.
*The flip-flops are connected in cascade. The main clock source is applied directly to first flip-flop.The output of first flip-flop acts as clock input for second flip-flop.
*All the J and K inputs are tied to logic high(=1).Hence each flip-flop changes state(or toggle) at each negative edge of its clock input.
*Flip-flop A changes state at each negative edge of main clock input and it changes its state before triggering flip-flop B.

*Flip-flop B changes state at each negative edge of its clock input coming from output A. *It is clear that triggers move like a ripple in water.Hence this is known as ripple counter. *Wave Form/Timing diagram :



*Truth Table

Clock transition	B A
0	0 0
1	0 1
2	1 0
3	1 1
4	0 0

Operation

*Initially, the flip-flops are reset to 0 output.Considering A as the LSB(Least Significant Bit) and B as the MSB, we get BA=00.This is the initial state of the counter.

*Flip-flop A changes state at each negative edge of main clock input.Flip-flop B changes state at each negative edge of output A.

*When the first clock pulse is applied, flip-flop A changes state from 0 to 1 at netgative edge of input clock pulse.But flip-flop B remains unchanged at negative edge of first clock pulse because of clock input=0.Hence output of counter at the end of first clock pulse negative edge transition is BA=01. *At the end of second clock pulse negative edge transition A changes from 1 to 0 and B changes from 0 to 1 ..Hence after application of second clock pulse contents are BA=10 .

*At the end of third clock pulse A changes from 0 to 1 and no change in B.Hence the contents of counter after application of third clock pulse is BA=11.

But after 4^{th} clock pulse again contents returns to BA=00.

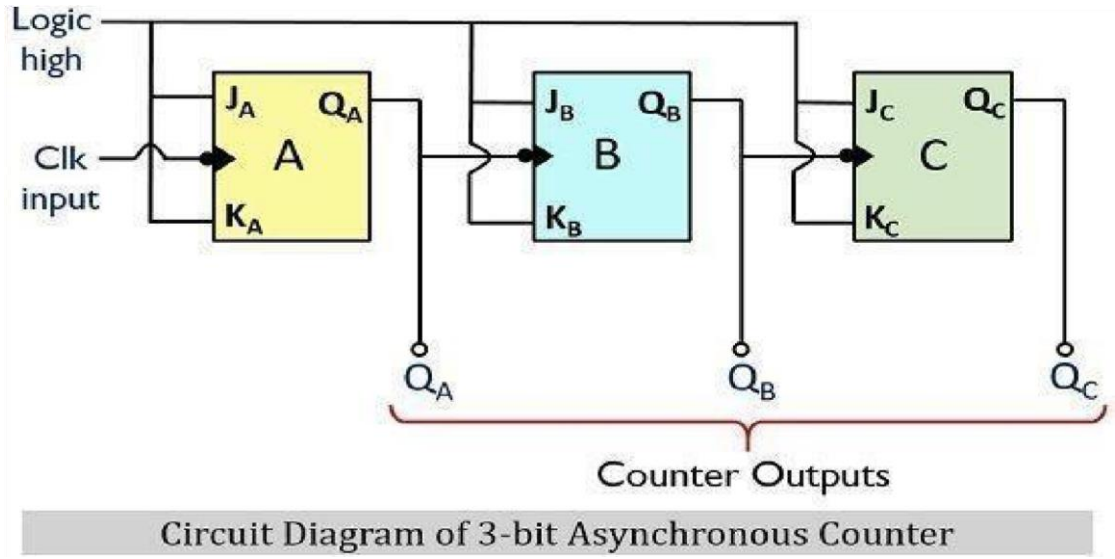
*For further application of clock pulses, the counter proceeds again from 00 to 11.This process continues till time main clock input is available.

It is concluded that maximum count by Mod -4 counter is 11(3 in decimal) and total number of counting sequence is 4 starting from 00(decimal 0) to 11.

*The timing diagram shows that the frequency of wave form A is one-half that of main clock input, frequency of wave form B is one-half that of A. **Mod-8 or 3-bit Series Up-Counter**

*Mod-8 series up counter counts 8 number of sequences in ascending mode starting from 000 ton 111.After counting sequence 111 again it starts counting from 000.Each counting sequence has 3-bits.

Logic Diagram



Circuit Diagram of 3-bit Asynchronous Counter

*It consists of 3 master slave JK flip flops named as A,B,C.

*The flip-flops are connected in cascade.The main clock source is applied directly to first flip-flop.The output of first flip-flop acts as clock input for second flip-flop and output of second flip-flop acts as clock input for third flip flop.

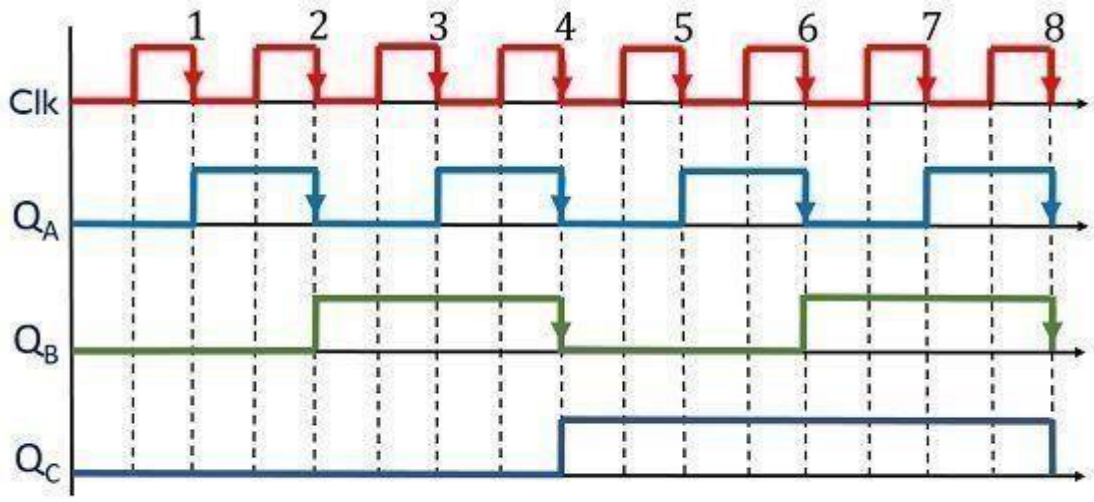
*All the J and K inputs are tied to logic high(=1).Hence each flip-flop changes state(or toggle) at each negative edge of its clock input.

*Flip-flop A changes state at each negative edge of main clock input and it changes its state before triggering flip-flop B.

*Flip-flop B changes state at each negative edge of its clock input coming from output A and it changes its state before triggering flip-flop C.

*Similarly flip-flop C changes state at each negative edge of output from flip-flop B.

*It is clear that triggers move like a ripple in water.Hence this is known as ripple counter. *Wave Form/Timing diagram :



Timing Diagram of 3-bit Asynchronous Counter

*Truth Table

Clock Transition	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Operation

*Initially, the flip-flops are reset to 0 output.Considering A as the LSB(Least Significant Bit) and C as the MSB, we get $Q_CQ_BQ_A$=000.This is the initial state of the counter.

*Flip-flop A changes state at each negative edge of main clock input.Flip-flop B changes state at each negative edge of output A.Flip-flop C changes state at each negative edge of output B.

*When the first clock pulse is applied, flip-flop A changes state from 0 to 1 at negative edge of input clock pulse.But flip-flop B and C remain unchanged at negative edge of first clock pulse because of clock input=0.Hence output of counter at the end of first clock pulse negative edge transition is $Q_CQ_BQ_A$=001.

*At the end of second clock pulse negative edge transition A changes from 1 to 0 and B changes from 0 to 1 and no change in C.Hence after application of second clock pulse contents are $Q_CQ_BQ_A$=010 . *At the end of third clock pulse A changes from 0 to 1 and no change in B and C.Hence the contents of counter after application of third clock pulse is $Q_CQ_BQ_A$=011.

*At the end of fourth clock pulse A changes from 1 to 0, B changes from 1 to 0 and C from 0 to 1.Hence contents of counter $Q_CQ_BQ_A$=100.

*At the end of fifth clock pulse, A changes from 0 to 1, no change in B and C.Now contents $Q_CQ_BQ_A$=101. *At the end of sixth clock pulse, A changes from 1 to 0, B changes from 0 to 1 and no change in C. Hence contents of counter after sixth clock pulse, $Q_CQ_BQ_A$=110.

*At the end of seventh clock pulse A changes from 0 to 1,no change in B and C. Finally after 7^{th} clock pulse contents, $Q_CQ_BQ_A$=111.

But after 8^{th} clock pulse again contents returns to $Q_CQ_BQ_A$=000.

*For further application of clock pulses, the counter proceeds again from 000 to 111.This process continues till time main clock input is available.

It is concluded that maximum count by Mod -8 counter is 111(7 in decimal) and total number of counting sequence is 8 starting from 000(decimal 0) to 111.

*The timing diagram shows that the frequency of wave form A is one-half that of main clock input, frequency of wave form B is one-half that of A and frequency of wave form C is one-half that of B. *The timing diagram shows that the frequency of wave form A is one-half that of main clock input, frequency of wave form B is one-half that of A and frequency of wave form C is one-half that of B.Frequency of wave form D is one-half that of C.

Note: Series up counters of higher(>Mod-16) modulus can be designed in similar way as discussed above.

Parallel/Synchronous Counter

*In parallel counter, the clock input is applied to all the flip-flops simultaneously.Hence all the flip-flops change their state simultaneously in synchronism with clock input. *It uses the same clock signal from the same source at the same time.

* It avoids the demerits of series counter. Series counter propagation delay time or settling time is additive that reduces speed.This speed limitation can be avoided by parallel counter.

*In this counter the LSB flip-flop J an K inputs are tied to logic high(0r 1).Accordingly this flip-flop toggles (or complement)with each clock pulse from one state to another.

*The other position flip-flops toggle with a pulse, under a condition that all the outputs in the lower-order position flip-flops are equal to 1. **Applications of Parallel Counter** 1-Used in alarm clock.

2-AC timer setting.

3-Used in camera to set timing to take the picture.

4-It is used in automobiles as flashing light indicator.

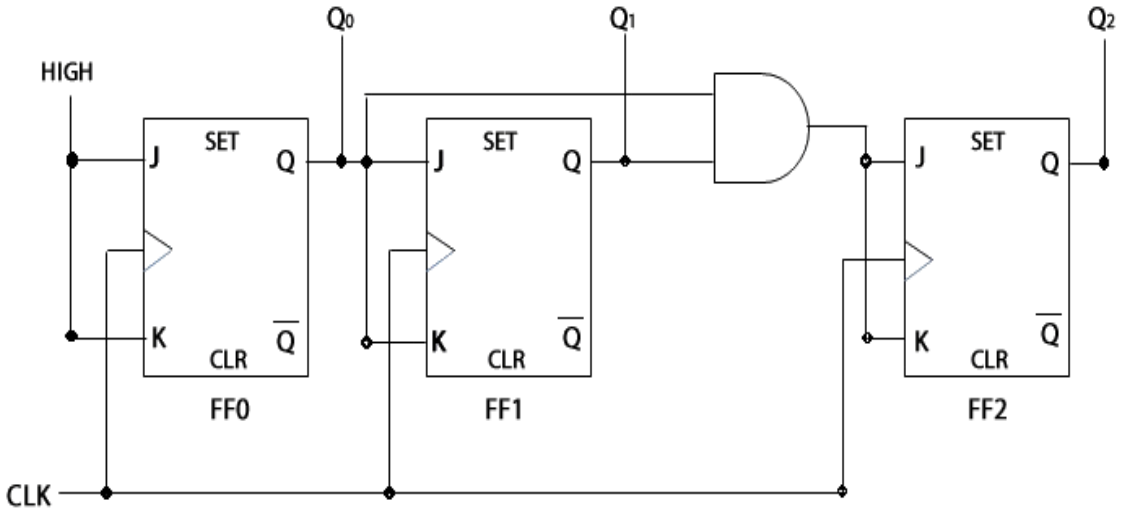
5-It can be used in car parking .

6-It is also used in counting the event or time allotted for special process.

Mod-8/ 3-Bit Parallel Up Counter

*Mod-8 parallel up counter counts 8 numbers of sequences in ascending mode starting from 000 ton 111.After counting sequence 111 again it starts counting from 000.Each counting sequence has 3-bits.

Logic Diagram

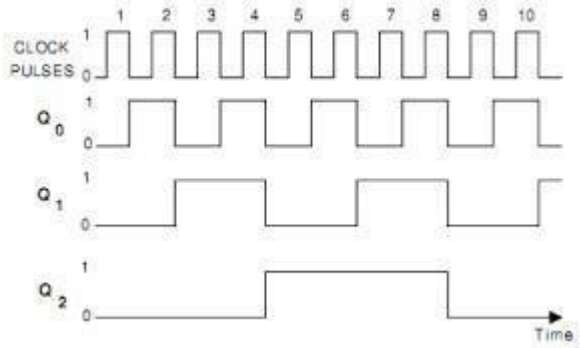


*The flip-flops are connected as in logic circuit.The main clock source is applied directly to all the three flip-flops from the same clock source at the same time.

*The J and K inputs of first flip-flop are tied to logic high(=1).Hence the first flip-flop changes state(or toggle) at each negative edge of its clock input.

*The output Q_0 of first flip-flop is connected to both J and K inputs of second flip-flop.Hence flip-flop Q_2 toggles only when output Q_0 is at logic 1.

*The output Q_0 of first flip-flop and output Q_1 of second flip-flop are applied to an AND gate.The AND gate output is connected to both J and K inputs of third flip-flop.Flip-flop Q_2 changes state only when AND gate output is high(or 1). AND gate output will be high only when both output Q_0 and Q_1 are at logic 1. Hence Flip-flop Q_2 toggles only when output Q_0 and Q_1 are at logic1. *Timing diagram:



*Truth Table

Clock Transition	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Operation

*Initially, the flip-flops are reset to 0 output.Considering Q0 as the LSB(Least Significant Bit) and Q_2 as the MSB, we get Q_2Q1Q_0=000.This is the initial state of the counter.

*Flip-flop Q0 changes state at each negative edge of main clock input.Flip-flop Q_1 changes state at each negative edge of output Q_0.Flip-flop Q_2 changes state at each negative edge of output Q_1.

*When the first clock pulse is applied, flip-flop Q0 changes state from 0 to 1 at netgative edge of input clock pulse.But flip-flop Q_1 and Q_2 remain unchanged at negative edge of first clock pulse because the J and K inputs of both second and third flip-flop are 0 i.e. $J_1=K_1=J_2=K_2=0$.Hence output of counter at the end of first clock pulse negative edge transition is $Q_2Q_1Q_0$=001.

*At the end of second clock pulse negative edge transition Q_0 changes from 1 to 0 and Q_1 changes from 0 to 1 i.e. both flip-flops toggle because of $J_0=K_0=1$ and $J_B=K_B=1$.But C remains unchanged because $J_2=K_2=0$.Hence after application of second clock pulse contents are $Q_2Q_1Q_0$=010 .

*At the end of third clock pulse Q_0 changes from 0 to 1 and no change in Q_1 and Q_2 because $J_1=K_1=J_2=K_2=0$.Hence the contents of counter after application of third clock pulse is $Q_2Q_1Q_0$=011. *At the end of fourth clock pulse Q_0 changes from 1 to 0, Q_1 changes from 1 to 0 and Q_2 from 0 to 1.Hence contents of counter $Q_2Q_1Q_0$=100.

*At the end of fifth clock pulse, Q_0 changes from 0 to 1, no change in Q_1 and Q_2.Now contents $Q_2Q_1Q_0$=101.

*At the end of sixth clock pulse, Q0 changes from 1 to 0, Q_1 changes from 0 to 1 and no change in Q_2.Hence contents of counter after sixth clock pulse, $Q_2Q_1Q_0$=110.

*At the end of seventh clock pulse Q_0 changes from 0 to 1,no change in Q_1 and Q_2.Finally after 7^{th} clock pulse contents, $Q_2Q_1Q_0$=111.

But after 8^{th} clock pulse again contents returns to $Q_2Q_1Q_0$=000.

*For further application of clock pulses, the counter proceeds again from 000 to 111.This process continues till time main clock input is available.

It is concluded that maximum count by Mod -8 counter is 111(7 in decimal) and total number of counting sequence is 8 starting from 000(decimal 0) to 111.

*The timing diagram shows that the frequency of wave form Q_0 is one-half that of main clock input, frequency of wave form Q_1 is one-half that of Q_0 and frequency of wave form Q_2 is one-half that of Q_1

REFERENCES

1-Digital Electronics –Principle and Application-By S.K. Mandal
2-Fundamental of digital Electronics – By Ananda Kumar
3-Digital Electronics- By B. R. Gupta & V. Singhal