LAKSHMI NARAIN COLLEGE OF TECHNOLOGY AND SCIENCE

LAB MANUAL

SEMESTER - III

DIGITAL SYSTEM DESIGN

EC - 303

LNCT SM
GROUP OF COLLEGES
"WORKING TOWARDS BEING THE BEST"

VISION of the department:

To be world-wide recognized for adopting and keeping innovation and entrepreneurship mindset as abreast of learning to produce professionals as valuable, ethical and moral resource for industry and society.

MISSION of the department:

- **To establish an ecosystem where students could grow with innovative practices followed in communication engineering.**
- **Adopt the global approaches to transform the young aspirant into engineering professional catering the society with ethical and patriotic zeal.**
- **Facilitate and felicitate the learners to have close interactions with the industry experts and researchers for keeping them updated of the current and future needs of the society.**
- **To develop the mindset of learners for being innovative and entrepreneurial in becoming successful professional.**

SCHEME

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

New Scheme of Examination as per AICTE Flexible Curricula

III Semester **Bachelor of Technology (B.Tech.) [Electronics & Communication Engineering]**

For batches admitted in July, 17 & July, 18 (w.e.f. July, 2018)

S.No.	Subject Code	Category	Subject Name	Maximum Marks Allotted					Total Marks	Contact Hours per week			Total Credits
				Theory			Practical						
				End Sem.	Mid Sem. Exam.	Quiz/ Assignment	End Sem	Term work / Lab Work & Sessional		L	T	P	
1.	BT301	BSC-5	Mathematics-III	70	20	10	-	-	100	3	1	-	4
2.	EC302	DC-1	Electronic Measurement & Instrumentation	70	20	10	-	-	100	3	1	-	4
3.	EC303	DC-2	Digital System Design	70	20	10	30	20	150	3	-	2	4
4.	EC304	DC-3	Electronic Devices	70	20	10	30	20	150	3	-	2	4
5.	EC305	DC-4	Network Analysis	70	20	10	30	20	150	3	-	2	4
6.	EC306	DLC-3	EMI Lab	-	-	-	30	20	50	-	-	4	2
7.	BT107	DLC-1	Evaluation of Internship-I completed at I year level	-	-	-	-	50	50			4	2
8.	BT307	DLC-4	90 hrs Internship based on using various software's –Internship -II	To be completed anytime during Third/ fourth semester. Its evaluation/credit to be added in fifth semester.									
			Total	350	100	50	120	130	750	15	2	14	24
			NSS/NCC										

1 Hr Lecture	1 Hr Tutorial	2 Hr Practical
1 Credit	1 Credit	1 Credit

SYLLABUS

Unit-1 Number Systems: Decimal, Binary, Octal and Hexadecimal systems, conversion from one base to another, Codes-BCD, Excess- 3, Gray Reflected ASCII, EBCDIC.

Logic gates and binary operations- AND, OR, NOT, NAND, NOR, Exclusive–OR and Exclusive– NOR Implementations of Logic Functions using gates, NAND–NOR implementations – Multi level gate implementations- Multi output gate implementations.

Boolean postulates and laws – De-Morgan's Theorem - Principle of Duality, Boolean function, Canonical and standard forms, Minimization of Boolean functions, Minterm, Maxterm, Sum of Products (SOP), Product of Sums (POS), Karnaugh map Minimization, Don't care conditions, Quine-McCluskey method of minimization.

Unit-2 Combinational logic circuits :Half adder – Full Adder – Half subtractor - Full subtractor– Parallel binary adder, parallel binary Subtractor – Fast Adder - Carry Look Ahead adder– Serial. Adder/Subtractor - BCD adder – Binary Multiplier – Binary Divider - Multiplexer/De-multiplexer – decoder - encoder – parity checker – parity generators – code converters - Magnitude Comparator.

Unit-3. Sequential Logic Design: Building blocks like S-R, JK and Master-Slave JK FF, Edge triggered FF, Finite state machines, Design of synchronous FSM, Algorithmic State Machines charts. Designing synchronous circuits like Pulse train generator, Pseudo Random Binary Sequence generator, Clock generation

Unit-4 Registers and Counters: Asynchronous Ripple or serial counter. Asynchronous Up/Down counter - Synchronous counters – Synchronous Up/Down counters – Programmable counters – Design of Synchronous counters: state diagram-State table –State minimization –State assignment - Excitation table and maps-Circuit. Implementation - Modulo–n counter, Registers – shift registers - Universal shift registers. Shift register counters – Ring counter – Shift counters - Sequence generators.

Unit-5 Logic Families and Semiconductor Memories: TTL NAND gate, Specifications, Noise margin, Propagation delay, fan-in, fan-out, Tristate TTL, ECL, CMOS families and their interfacing, Memory elements, Concept of Programmable logic devices like FPGA. Logic implementation using Programmable Devices.

LIST OF EXPERIMENTS

ADDITIONAL

EXPERIMENT - 1

LOGIC GATES

AIM: To study and verify the Truth Tables of AND, OR, NOT, NAND, NOR, and EXOR logic gates.

APPARATUS REQUIRED:

Digital logic trainer, and Patch cords.

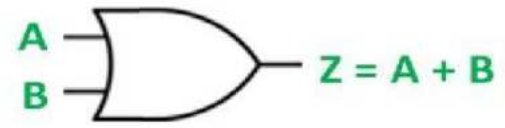
THEORY:

AND Gate: A multi-input circuit in which the output is 1 only if all inputs are 1. The symbolic representation of the AND gate is:

2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot (.) is used to show the AND operation i.e. A.B .

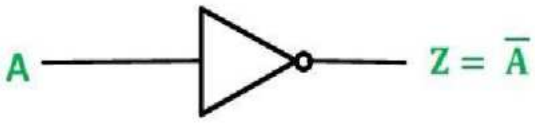
OR Gate : A multi-input circuit in which the output is 1 when any input is 1.
The symbolic representation of the OR gate is:



2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.

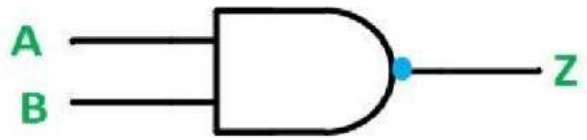
NOT gate: The output is 0 when the input is 1, and the output is 1 when the input is 0. The symbolic representation of an inverter is :



NOT gate	
A	$\overline{A}$
0	1
1	0

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top, as shown at the outputs.

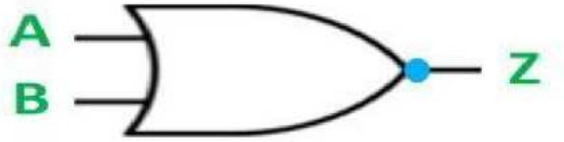
NAND gate: AND followed by INVERT. It is also known as universal gate.The symbolic representation of the NAND gate is:



2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

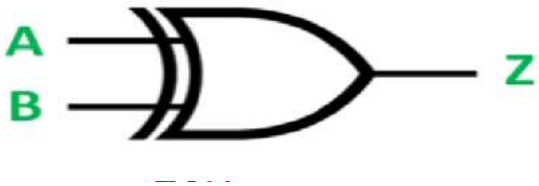
NOR gate: OR followed by inverter. It is also known as universal gate.The symbolic representation is:



2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

EXOR gate: The output of the Exclusive –OR gate, is 0 when it's two inputs are the same and its output is 1 when its two inputs are different. It is also known as Anti-coincidence gate.



2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both, of its two inputs are high. An encircled plus sign (⊕) is used to show the EOR operation.

Observation Table:

S.No	Input(A) LED	Input(B) LED	Output (OR) $Y = \overline{A}$	Output (AND) Y=AB	Output (OR) Y=A+B	Output (NAND) $Y = \overline{AB}$	Output (NOR) $Y= \overline{A+B}$	Output (XOR) Y=A⊕B
1								
2								
3								
4								

CALCULATION:

Results and Analysis:

NOT Gate: When logic 1 is applied to one of NOT gate of 7404 IC, then output becomes zero. When input LED is ON (RED), the output LED become OFF (Green) vice versa.

OR Gate: The output of an OR gate is a 1 if one or the other or both of the inputs are 1, but a 0 if both inputs are 0. When One or the other or Both of the input LEDS are ON (RED Light), then output LED is ON (RED) otherwise Output LED is OFF(Green Light)

AND Gate: The output of an AND gate is only 1 if both its inputs are 1. For all other possible inputs the output is 0.When both the LEDS are On, then output LED is ON (RED Light) otherwise Output LED is OFF.

NOR Gate: The output of the NOR gate is a 1 if both inputs are 0 but a 0 if one or the other or both the inputs are 1.

NAND Gate: The output of the NAND gate is a 0 if both inputs are 1 but a 1 if one or the other or both the inputs are 0.

EXOR gate: The output of the XOR gate is a 1 if either but not both inputs are 1 and a 0 if the inputs are both 0 or both 1.

RESULT: The Truth Tables of AND, OR, NOT, NAND, NOR, and EXOR logic gates are studied and verified.

EXPERIMENT - 2

DEMORGAN'S THEOREMS

AIM:

To verify Demorgan's theorems.

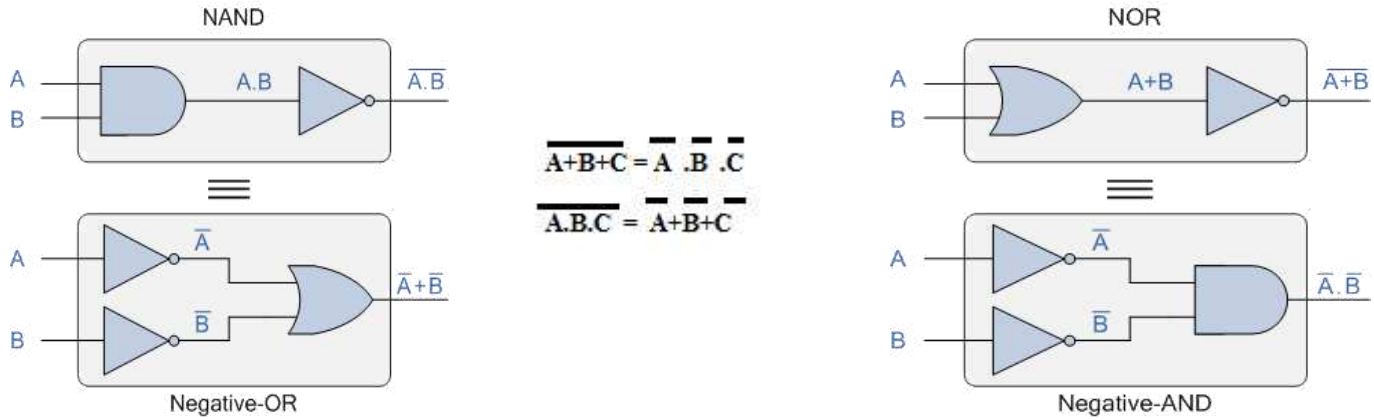
APPARATUS REQUIRED: IC 7400, IC 7402, breadboard, connecting wires, LED, and power supply.

THEORY:

The digital signals are discrete in nature and can only assume one of the two values O and 1.A number system based on these two digits is known as binary number system. This is basic of all digital systems like computers, calculators etc. Binary Variables can be represented by letter symbol such A,B,X,Y.

The variable can have only one of the two variable possible values at anytime i.e. '0' or '1'.Demorgan's Theorem can be proved by first considering the two variable case and then extending this result.

DEMORGAN'S THEOREMS:



$$\overline{A+B+C} = \overline{A}.\overline{B}.\overline{C}$$

$$\overline{A.B.C} = \overline{A}+\overline{B}+\overline{C}$$

THEOREM ONE **THEOREM TWO**

PROCEDURE:

DEMORGAN'S THEOREM -1

$$\overline{A.B.C} = \overline{A}+\overline{B}+\overline{C}$$

1. Make the circuit diagram as shown in fig and connect the inputs of the gate to the input state sockets A, B and C and output to the output indicators.
2. Set the input combinations one by one by putting input state switches A, B and C either in 0 or 1 state.
3. Now verify the output with the help of Truth Table (1)

TRUTH TABLE (1)

A	B	C	A	B	C	A.B.C	A.B.C	A+B+C
0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	1	1
0	1	0	1	0	1	0	1	1
0	1	1	1	0	0	0	1	1
1	0	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	1	1
1	1	1	0	0	0	1	0	0

DEMORGAN'S THEOREM-2:

$$\overline{A+B+C} = \overline{A}\ .\overline{B}\ .\overline{C}$$

Make the circuit diagram as shown in fig and connect the inputs o the gate to the input state sockets A, B and C and output to the output indicators.

1. Set the input combinations one by the putting input state switches A, B and C either '0' or '1' state.
2. Now verify the output with the help of Truth Table (2).

TRUTH TABLE (2)

A	B	C	A	B	C	A.B.C	A+B+C	A+B+C
0	0	0	1	**1**	1	0	1	1
0	0	1	1	**1**	0	1	0	0
0	1	0	1	**0**	1	1	0	0
0	1	1	1	**0**	0	1	0	0
1	0	0	0	**1**	1	1	0	0
1	0	1	0	**1**	0	1	0	0
1	1	0	0	**0**	1	1	0	0
1	1	1	0	**0**	0	1	0	0

RESULT: Demorgan's Theorems are verified.

EXPERIMENT - 03

HALF ADDER and FULL ADDER

AIM: To verify the truth tables of Half Adder and Full Adder.

APPARATUS REQUIRED:

Digital logic trainer, and Patch cords.

THEORY:

Half Adder

With the help of half adder, we can design circuits that are capable of performing simple addition with the help of logic gates.

Let us first take a look at the addition of single bits.

0+0 = 0

0+1 = 1

1+0 = 1

1+1 = 10

These are the least possible single-bit combinations. But the result for 1+1 is 10. Though this problem can be solved with the help of an EXOR Gate, if you do care about the output, the sum result must be re-written as a 2-bit output.

Thus the above equations can be written as

0+0 = 00

0+1 = 01

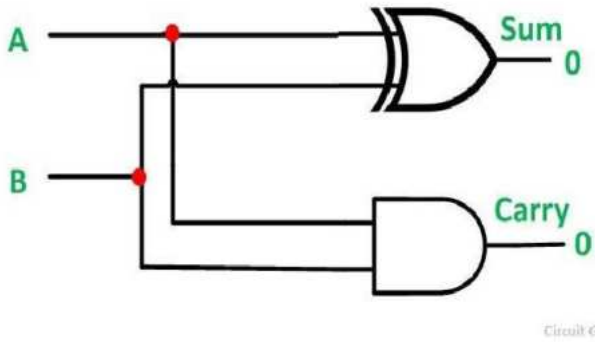
1+0 = 01

1+1 = 10

Here the output '1'of '10' becomes the carry-out. The result is shown in a truth-table below. 'SUM' is the normal output and 'CARRY' is the carry-out.

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

From the equation it is clear that this 1-bit adder can be easily implemented with the help of EXOR Gate for the output 'SUM' and an AND Gate for the carry. Take a look at the implementation below.



For complex addition, there may be cases when you have to add two 8-bit bytes together. This can be done only with the help of full-adder logic.

Full Adder

This type of adder is a little more difficult to implement than a half-adder. The main difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as CIN. When a full adder logic is designed we will be able to string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

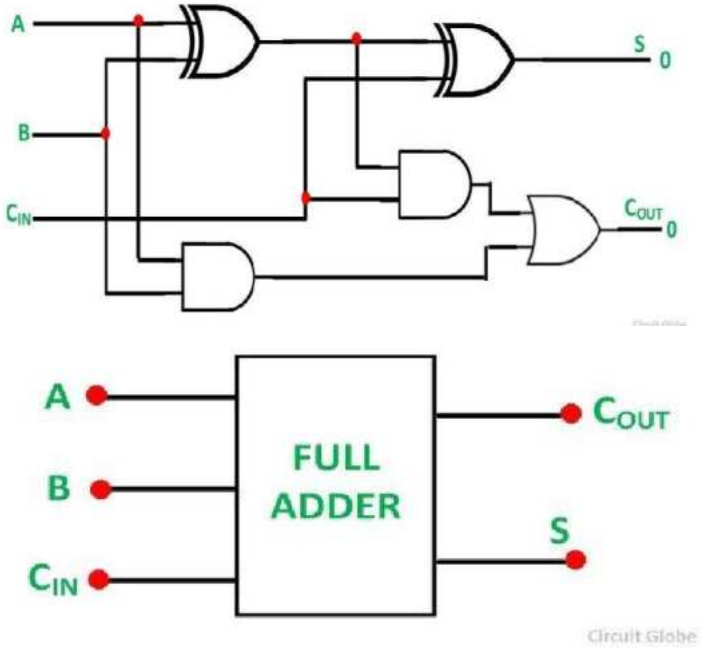
The output carry is designated as COUT and the normal output is designated as S. Take a look at the truth-table.

Input			Output	
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

From the above truth-table, the full adder logic can be implemented. We can see that the output S is an EXOR between the input A and the half-adder SUM output with B and CIN inputs. We must also note that the COUT will only be true if any of the two inputs out of the three are HIGH.

Thus, we can implement a full adder circuit with the help of two half adder circuits. The first will half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add CIN to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus, COUT will be an OR function of the half-adder Carry outputs.

.



With this type of symbol, we can add two bits together taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude. In a computer, for a multi-bit operation, each bit must be represented by a full adder and must be added simultaneously. Thus, to add two 8-bit numbers, you will need 8 full adders which can be formed by cascading two of the 4-bit blocks.

RESULT: The truth tables of Half Adder and Full Adder are verified.

EXPERIMENT - 04

HALF SUBTRACTOR AND FULL SUBTRACTOR

AIM: To verify the truth tables of Half Subtactor & Full Subtractor.

APPARATUS REQUIRED:

Digital logic trainer kit, and Patch cords.

THEORY

The arithmetic operation, subtraction of two binary digits has four possible elementary operations, namely,

0 - 0 = 0
0 - 1 = 1 with 1 borrow
1 - 0 = 1
1 - 1 = 0

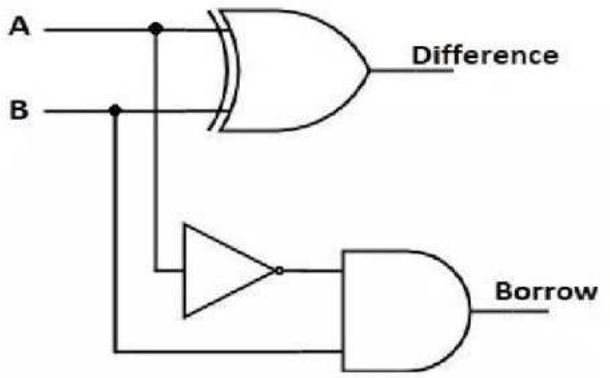
In all operations, each subtrahend bit is subtracted from the minuend bit. In case of the second operation the minuend bit is smaller than the subtrahend bit, hence 1 is borrowed.

HALF SUBTRACTOR:

A combinational circuit which performs the subtraction of two bits is called half subtractor. The input variables designate the minuend and the subtrahend bit, whereas the output variables produce the difference and borrow bits.

The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow).

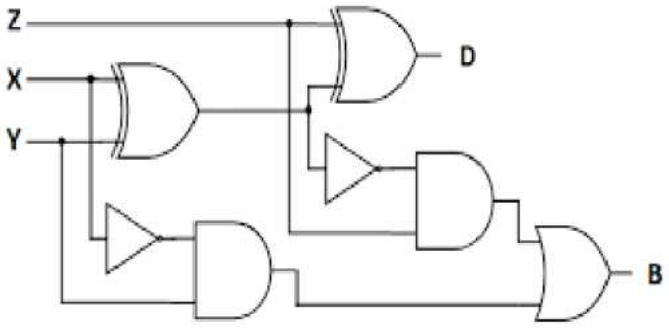
Input		Output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Full Subtractor:

A combinational circuit which performs the subtraction of three input bits is called full subtractor. The three input bits include two significant bits and a previous borrow bit. A full subtractor circuit can be implemented with two half subtractors and one OR gate. As in the case of the addition using logic gates, a full subtractor is made by combining two half-subtractors and an additional OR-gate. A full subtractor has the borrow in capability (denoted as BOR_{IN} in the diagram below) and so allows cascading which results in the possibility of multi-bit subtraction. The circuit diagram for a full subtractor is given below.

Input			Output	
A	B	C	Difference	Borrow
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



RESULT: The truth tables of Half Subtractor and Full Subtractor are verified.

EXPERIMENT - 05

MULTIPLEXER AND DE-MULTIPLEXER

AIM: -To design and verify the truth table of a 4x1 Multiplexer & 1x4 Demultiplexer.

APPARATUS REQUIRED:

S.No	Name of the Apparatus
1.	Digital IC trainer kit
2.	OR gate
3.	NOT gate
4.	AND gate (three input)
5.	Connecting wires

THEORY:

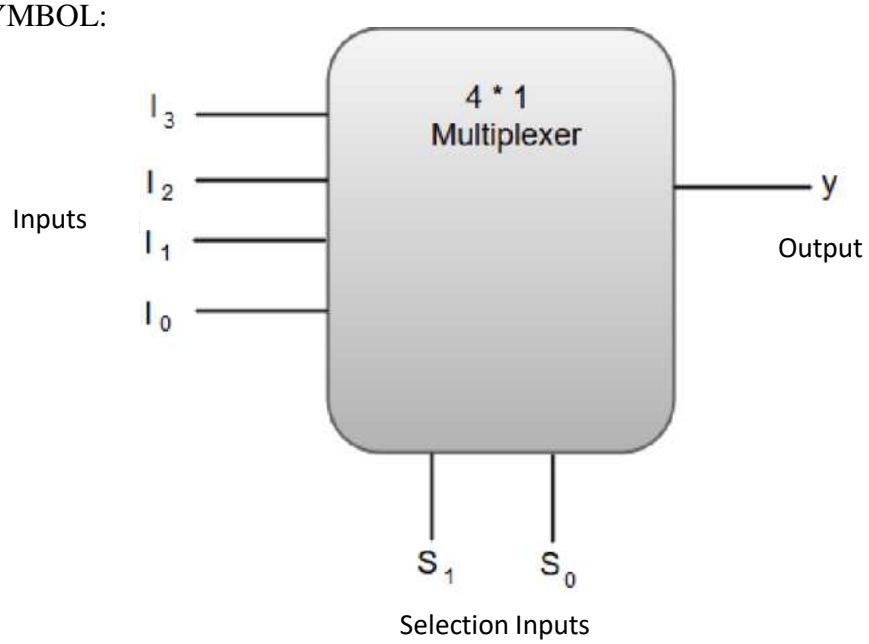
Multiplexer is a digital switch which allows digital information from several sources to be routed onto a single output line. The basic multiplexer has several data input lines and a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selector lines whose bit combinations determine which input is selected. Therefore, multiplexer is 'many into one' and it provides the digital equivalent of an analog selector switch.

A De-multiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of specific output line is controlled by the values of n selection lines.

DESIGN:

4x1 MULTIPLEXER

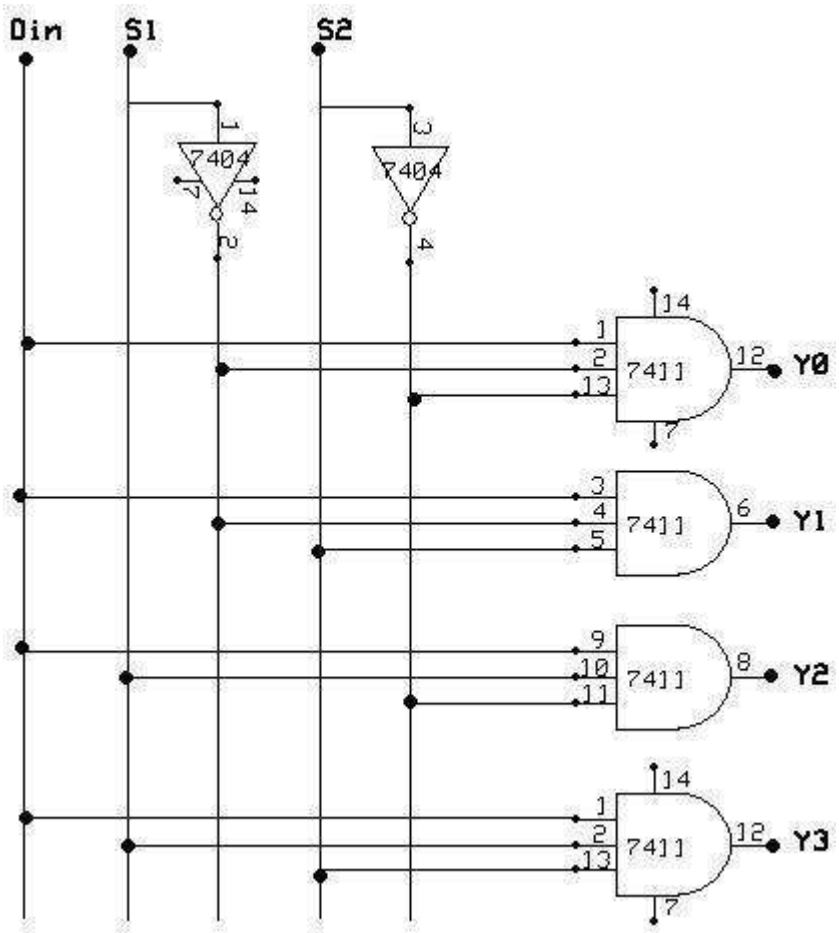
LOGIC SYMBOL:



TRUTH TABLE:

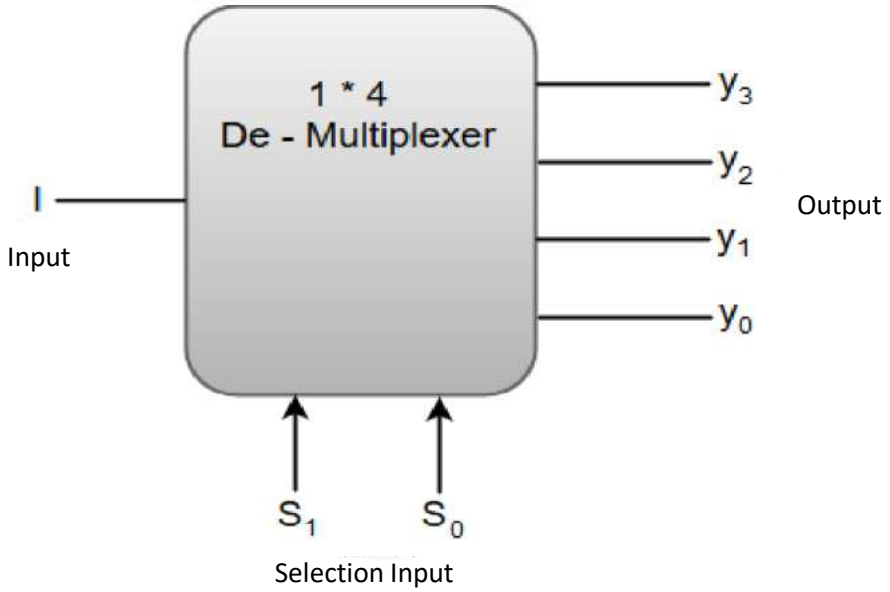
S.No	SELECTION INPUT		OUTPUT
	S1	S2	Y
1.	0	0	I_0
2.	0	1	I_1
3.	1	0	I_2
4.	1	1	I_3

CIRCUIT DIAGRAM:



1x4 DE-MULTIPLEXER

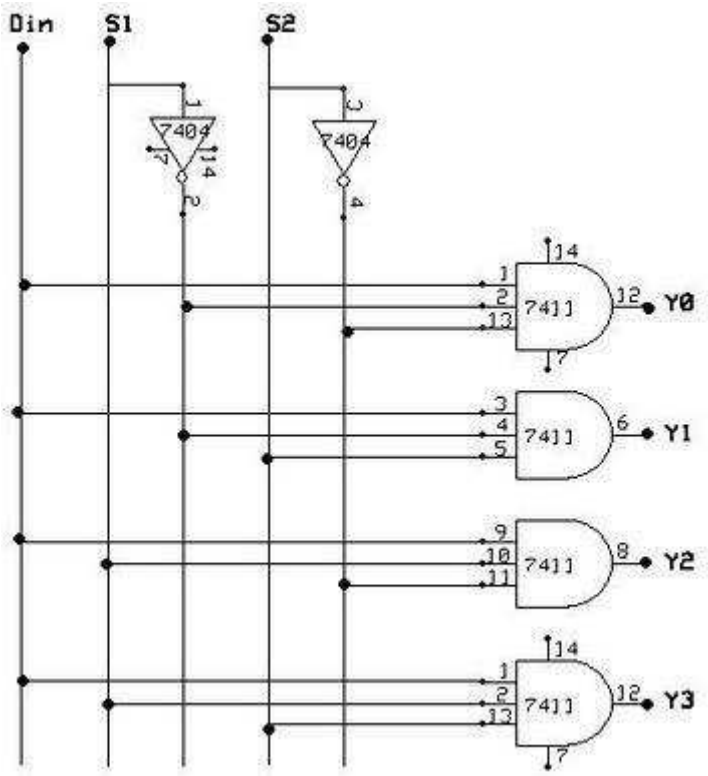
LOGIC SYMBOL:



TRUTH TABLE:

S.No	INPUT			OUTPUT			
	S1	S2	Din	Y0	Y1	Y2	Y3
1.	0	0	0	0	0	0	0
2.	0	0	1	1	0	0	0
3.	0	1	0	0	0	0	0
4.	0	1	1	0	1	0	0
5.	1	0	0	0	0	0	0
6.	1	0	1	0	0	1	0
7.	1	1	0	0	0	0	0
8.	1	1	1	0	0	0	1

CIRCUIT DIAGRAM:



PROCEDURE:

1. Connections are given as per the circuit diagrams.
2. For all the ICs 7th pin is grounded and 14th pin is given +5 V supply.
3. Apply the inputs and verify the truth table for the multiplexer & de-multiplexer.

RESULT: The truth tables of 4x1 Multiplexer and 1x4 De-multiplexer are verified.

EXPERIMENT-06

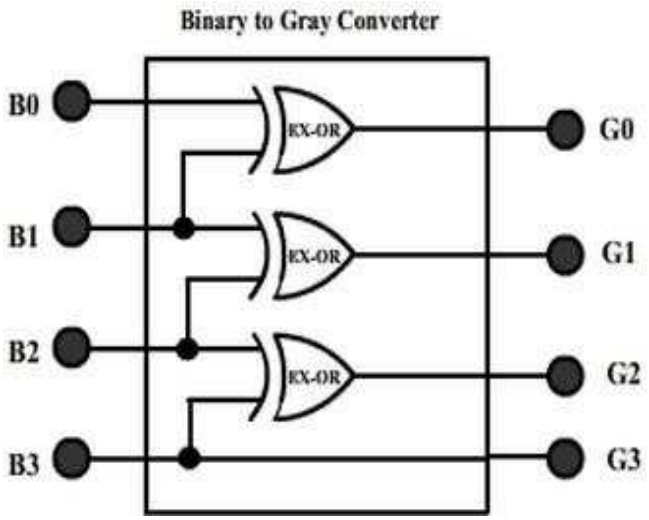
BINARY TO GRAY CODE CONVERTER

AIM: To design and verify 4-bit binary to gray code converter.

COMPONENTS REQUIRED: IC 7486, trainer kit, patch cords.

THEORY: The logical circuit which converts the binary code to equivalent gray code is known as binary to gray code converter. The gray code is a non-weighted code. The successive gray code differs in one-bit position only that means it is a unit distance code. It is also referred as a cyclic code. It is not suitable for arithmetic operations. It is the most popular of the unit distance codes. It is also a reflective code. An n-bit Gray code can be obtained by reflecting an n-1 bit code about an axis after 2n-1 rows and putting the MSB of 0 above the axis and the MSB of 1 below the axis.

Circuit implementation:



Binary to gray code converter:

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

G3 = B3

0	1	1	0
0	1	1	0
1	0	0	1
1	0	0	1

G1=B1 ⊕ B2

0	1	0	1
0	1	0	1
0	1	0	1
0	1	0	1

G2= B3 ⊕ B2

0	0	0	0
1	1	1	1
0	0	0	0
1	1	1	1

G0=B1 ⊕ B0

Truth table:

Natural-binary code				Gray code			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Result:

4-bit binary to gray code converter is studied and verified.

EXPERIMENT-07

COMPARATOR

AIM: To study and verify 2-bit comparator.

APPARATUS REQUIRED: IC trainer kit, and patch cords.

THEORY: A magnitude digital Comparator is a combinational circuit that **compares two digital or binary numbers** in order to find out whether one binary number is equal, less than or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for A > B condition, one for A = B condition and one for A < B condition.

2-Bit Magnitude Comparator –

A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers.

$$(A>B) = A1\,\bar{B1} + A0\,\bar{B1}\,\bar{B0} + \bar{B0}\,A1A0$$

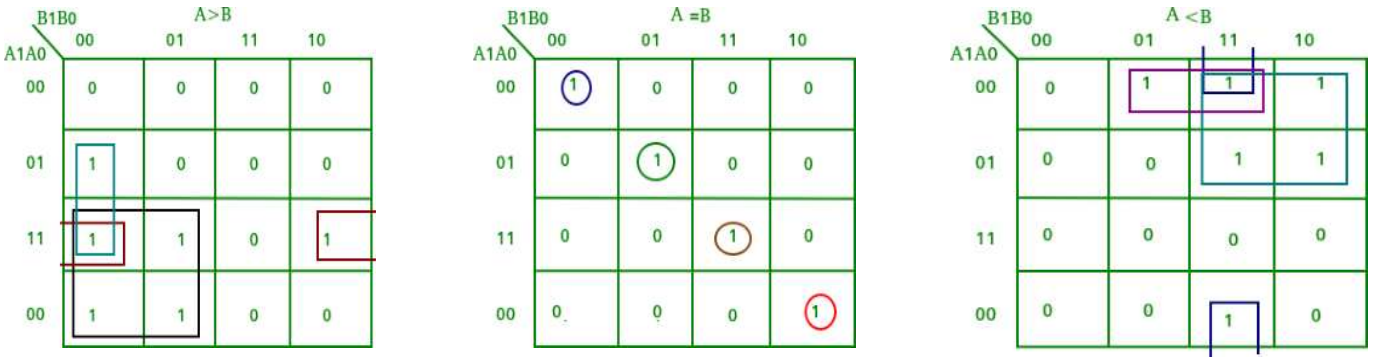
$$(A=B) = (A0 \oplus B0)\,(A1 \oplus B1)$$

$$(A<B) = B1\,\bar{A1} + B0\,\bar{A1}\,\bar{A0} + \bar{A0}B1B0$$

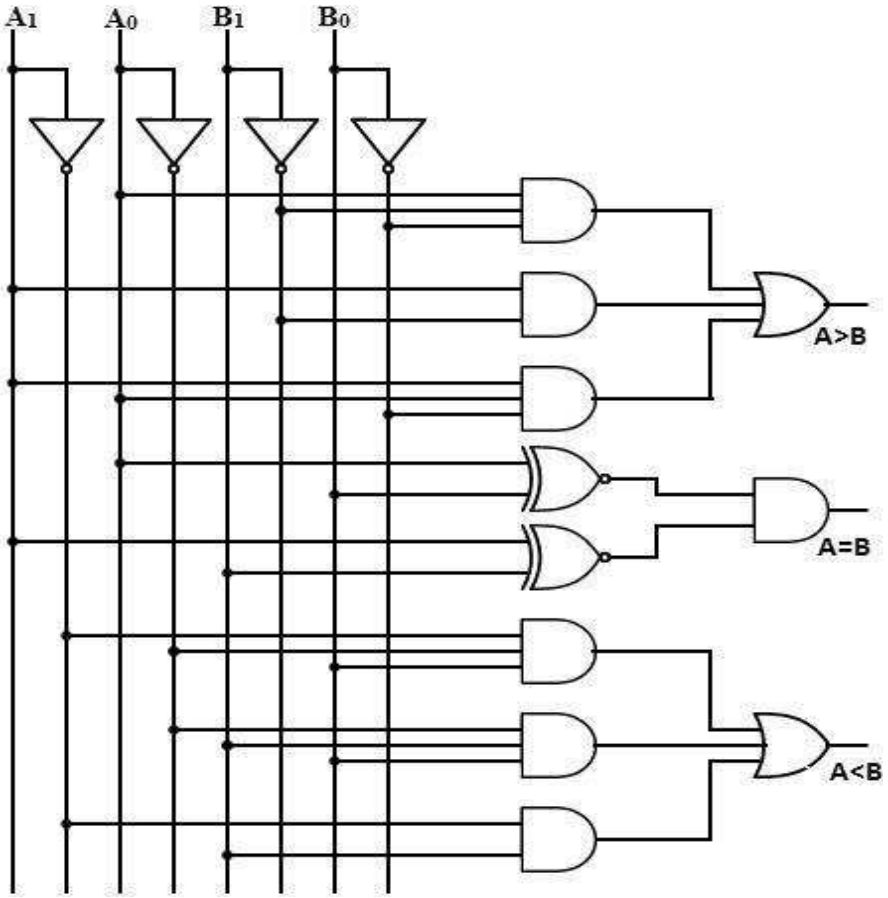
The truth table for a 2-bit comparator is given below:

Inputs				Outputs		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

K-MAP:



LOGIC DIAGRAM:



RESULT: 2-bit comparator is studied and verified.

EXPERIMENT-08

FLIP FLOPS

AIM: -To verify the characteristic table of RS, D, JK, and T Flip flops .

COMPONENTS REQUIRED:

S.No	Name of the Apparatus
1.	Digital IC trainer kit
2.	NOR gate
3.	NOT gate
4.	AND gate (three input)
5.	NAND gate
6.	Connecting wires

THEORY:

A Flip Flop is a sequential device that samples its input signals and changes its output states only at times determined by clocking signal. Flip Flops may vary in the number of inputs they possess and the manner in which the inputs affect the binary states.

RS FLIP FLOP:

The clocked RS flip flop consists of NAND gates and the output changes its state with respect to the input on application of clock pulse. When the clock pulse is high the S and R inputs reach the second level NAND gates in their complementary form. The Flip Flop is reset when the R input high and S input is low. The Flip Flop is set when the S input is high and R input is low. When both the inputs are high the output is in an indeterminate state.

D FLIP FLOP:

To eliminate the undesirable condition of indeterminate state in the SR Flip Flop when both inputs are high at the same time, in the D Flip Flop the inputs are never made equal at the same time. This is obtained by making the two inputs complement of each other.

JK FLIP FLOP:

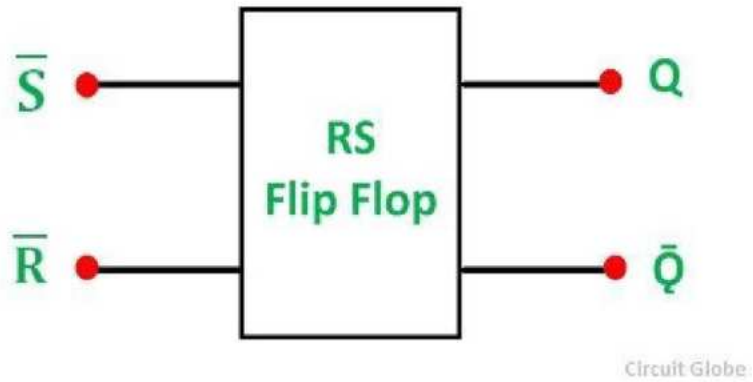
The indeterminate state in the SR Flip-Flop is defined in the JK Flip Flop. JK inputs behave like S and R inputs to set and reset the Flip Flop. The output Q is ANDed with K input and the clock pulse, similarly the output Q' is ANDed with J input and the Clock pulse. When the clock pulse is zero both the AND gates are disabled and the Q and Q' output retain their previous values. When the clock pulse is high, the J and K inputs reach the NOR gates. When both the inputs are high the output toggles continuously. This is called Race around condition and this must be avoided.

T FLIP FLOP:

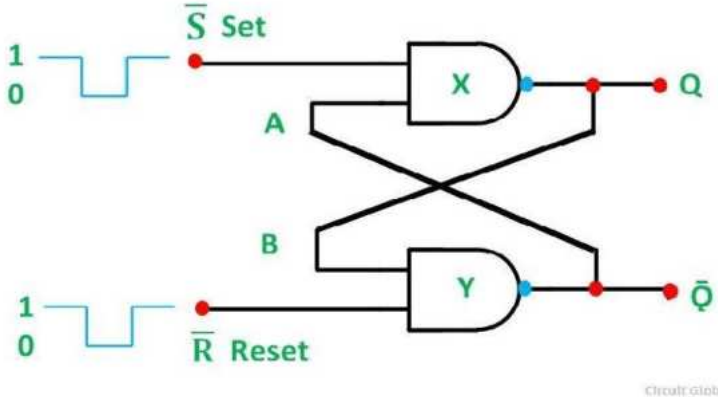
This is a modification of JK Flip Flop, obtained by connecting both inputs J and K inputs together. T Flip Flop is also called Toggle Flip Flop.

RS FLIP FLOP

LOGIC SYMBOL:



CIRCUIT DIAGRAM:

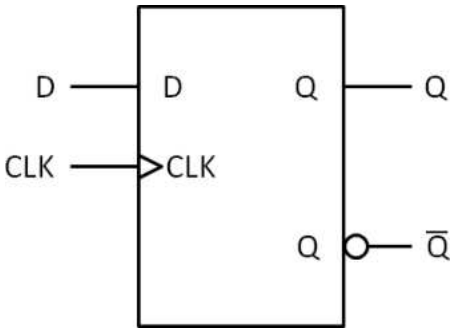


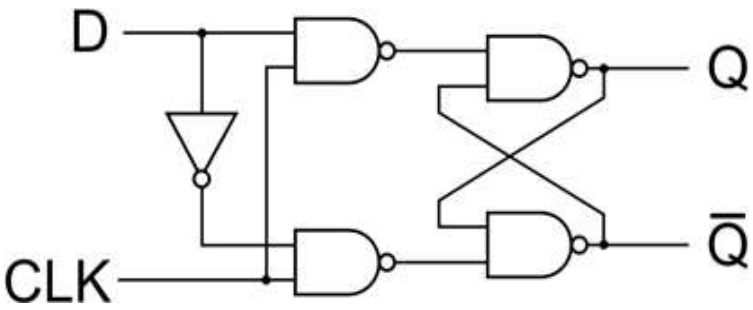
CHARACTERISTIC TABLE:

CLOCK PULSE	INPUT		PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
	S	R			
1	0	0	0	0	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	0	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	X	
8	1	1	1	X	

D FLIP FLOP

LOGIC SYMBOL AND CIRCUIT DIAGRAM:



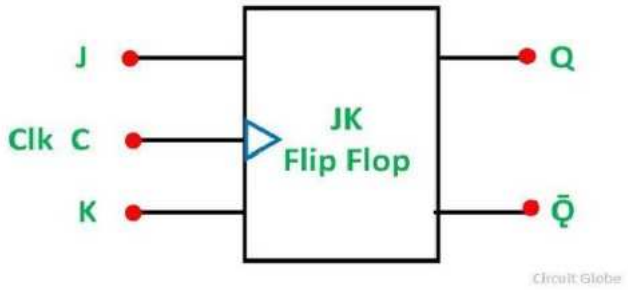


CHARACTERISTIC TABLE:

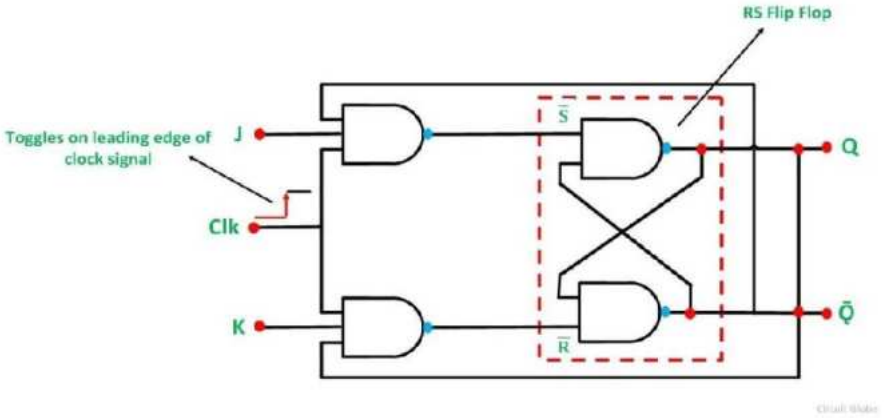
CLOCK PULSE	INPUT D	PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
1	0	0	0	
2	0	1	0	
3	1	0	1	
4	1	1	1	

JK FLIP FLOP

LOGIC SYMBOL:



CIRCUIT DIAGRAM:

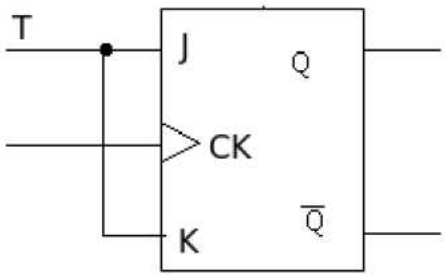


CHARACTERISTIC TABLE:

CLOCK PULSE	INPUT		PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
	J	K			
1	0	0	0	0	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	0	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	1	
8	1	1	1	0	

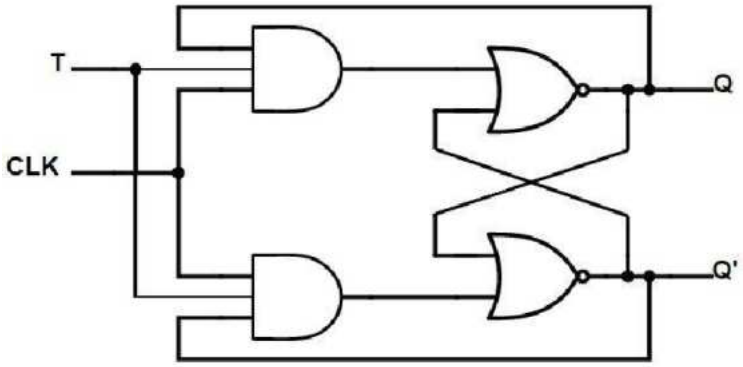
T FLIP FLOP

LOGIC SYMBOL:



CIRCUIT DIAGRAM:

:



CHARACTERISTIC TABLE:

CLOCK PULSE	INPUT T	PRESENT STATE (Q)	NEXT STATE(Q+1)	STATUS
1	0	0	0	
2	0	1	0	
3	1	0	1	
4	1	1	0	

PROCEDURE:

1. Connections are given as per the circuit diagrams.
2. For all the ICs 7th pin is grounded and 14th pin is given +5 V supply.

3. Apply the inputs and observe the status of all the flip flops.

RESULT: The characteristic tables of RS, D, JK, and D Flip Flops are verified.

EXPERIMENT-09

REGISTERS

AIM: To study 4-bit SISO, SIPO, PISO, and PIPO shift registers.

COMPONENTS REQUIRED: IC 7495, IC 7474, Patch Cords & IC Trainer Kit.

THEORY: Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. All the flip-flops are driven by a common clock, and all are set or reset simultaneously.

SISO-

The serial in/serial out shift register accepts data serially – that is, one bit at a time on a single line. It produces the stored information on its output also in serial form.

SIPO-

The serial in/parallel out shift register accepts data serially – that is, one bit at a time on a single line. It produces the stored information on its output in parallel form.

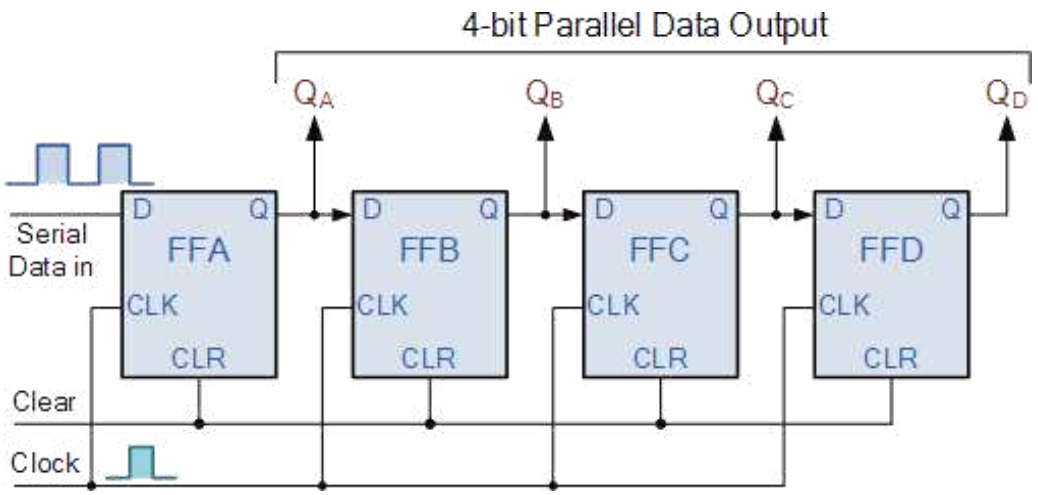
PISO-

The parallel in/serial out shift register accepts data in parallel. It produces the stored information on its output also in serial form.

PIPO-

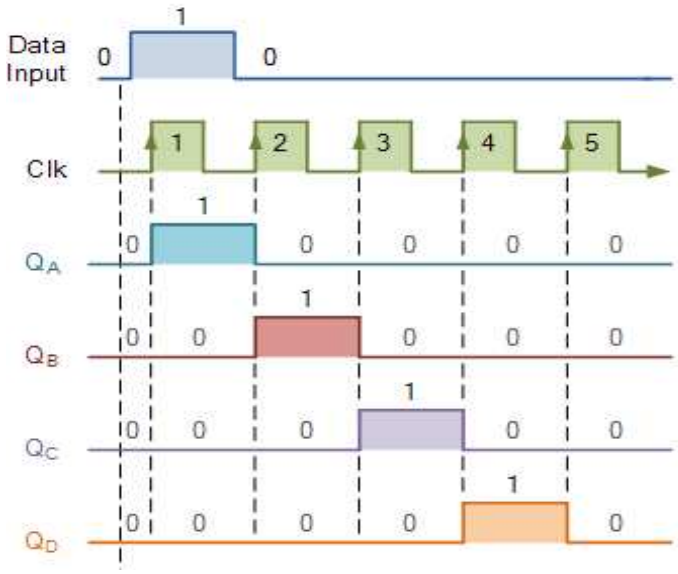
The parallel in/parallel out shift register accepts data in parallel. It produces the stored information on its output in parallel form

SERIAL-IN PARALLEL-OUT SHIFT REGISTER (SIPO):

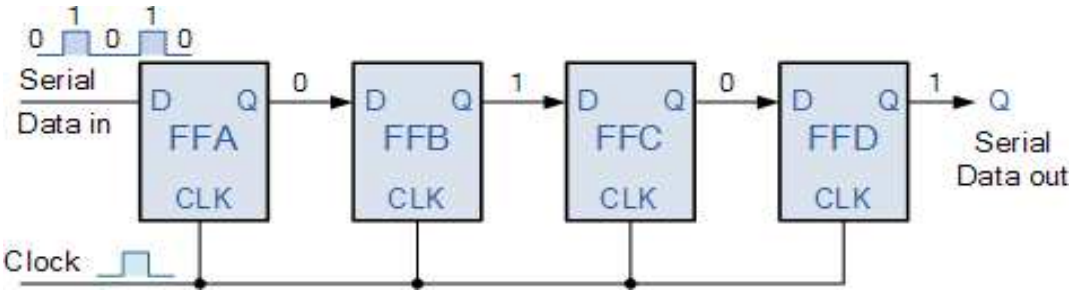


TRUTH TABLE AND OUTPUT WAVEFORM:

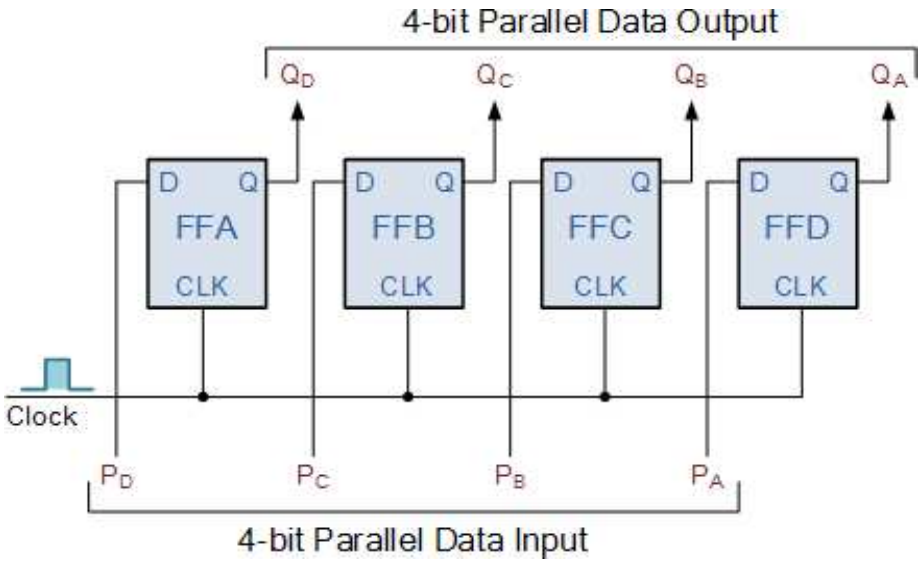
Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0



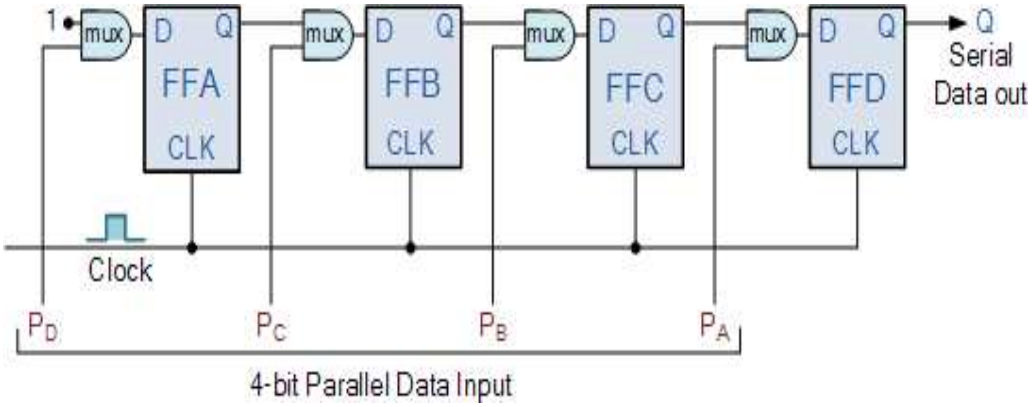
SERIAL-INPUT SERIAL-OUTPUT REGISTER (SISO):



PARALLEL-INPUT PARALLEL-OUTPUT SHIFT REGISTER (PIPO):



PARALLEL-INPUT SERIAL-OUTPUT SHIFT REGISTER (PISO):



RESULT: 4-bit SISO, SIPO, PISO, and PIPO shift registers are studied.

EXPERIMENT-10

COUNTERS

AIM: To study and verify 4-bit Synchronous UP Counter, and 4-bit Asynchronous UP Counter.

COMPONENTS REQUIRED: IC Trainer Kit, and Patch Cords.

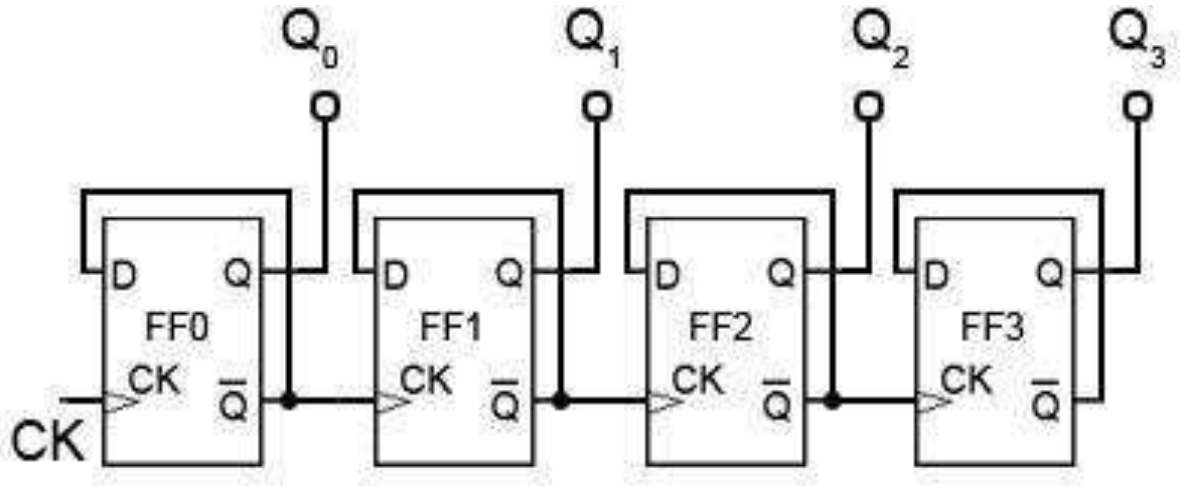
THEORY: A Counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit**.**

Asynchronous 4-bit UP counter

Asynchronous counters are those whose output is free from the clock signal. Because the flip flops in asynchronous counters are supplied with different clock signals, there may be delay in producing output.

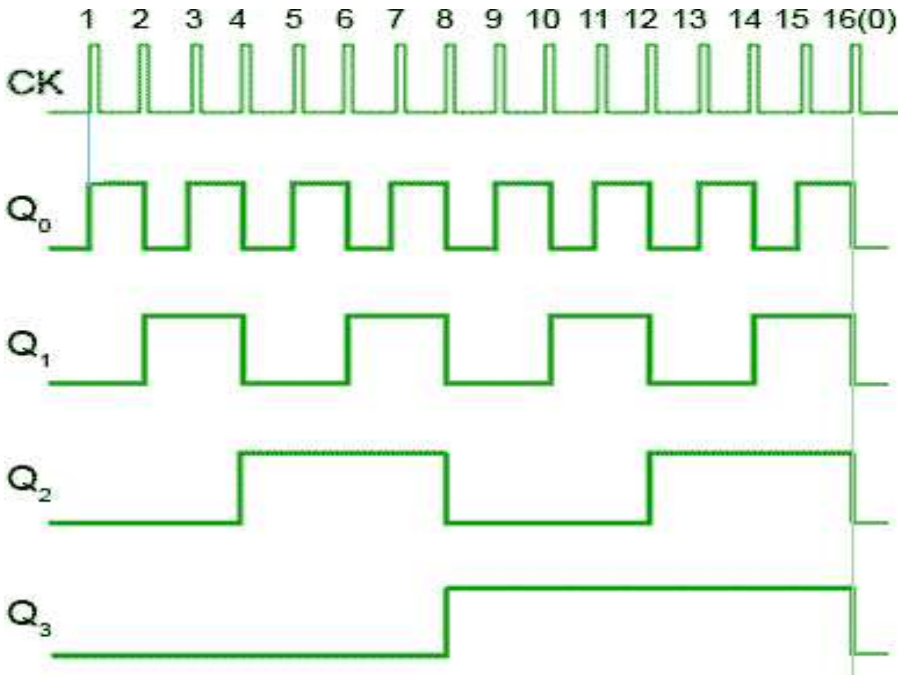
A 4 bit asynchronous UP counter with D flip flop is shown in the below diagram. It is capable of counting numbers from 0 to 15. The clock inputs of all flip flops are cascaded and the D input (DATA input) of each flip flop is connected to a state output of the flip flop.
That means the flip flops will toggle at each active edge or positive edge of the clock signal. The clock input is connected to first flip flop. The other flip flops in counter receive the clock signal input from Q' output of previous flip flop. The output of the first flip flop will change, when the positive edge on clock signal occurs.
In the asynchronous 4- bit up counter, the flip flops are connected in toggle mode, so when the when the clock input is connected to first flip flop FF0, then its output after one clock pulse will become 20.
The rising edge of the Q output of each flip flop triggers the clock input of its next flip flop. It triggers the next clock frequency to half of its applied input. The Q outputs of every individual flip flop (Q0, Q1, Q2, Q3) represents the count of the 4 bit UP counter such as 20 (1) to 23 (8).

CIRCUIT DIAGRAM



Characteristic Table and Output Waveform:

	toggle Frequency			
	f/16	f/8	f/4	f/2
	↓	↓	↓	↓
Decimal	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1



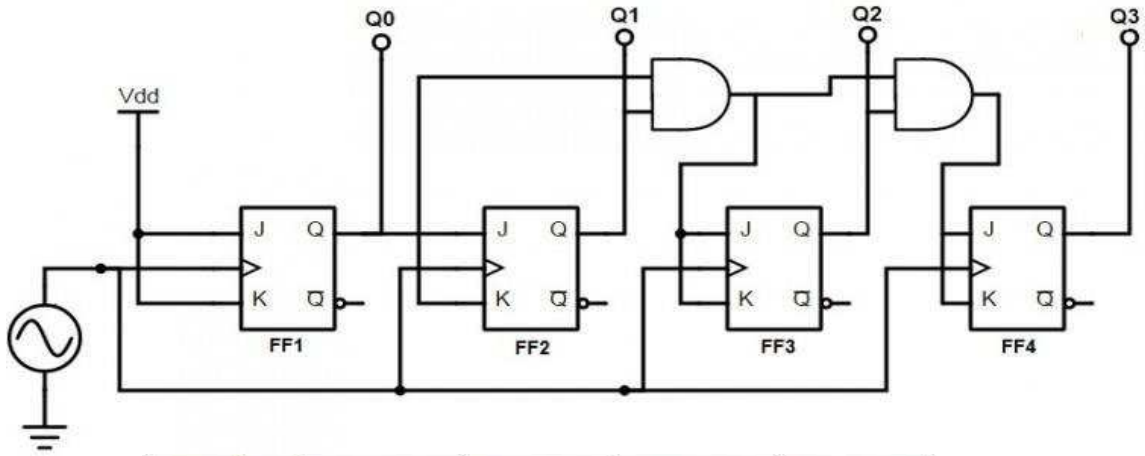
Synchronous 4-bit UP Counter

The counters which use clock signal to change their transition are called "Synchronous counters". This means the synchronous counters depends on their clock input to change state values. In synchronous counters, all flip flops are connected to the same clock signal and all flip flops will trigger at the same time.

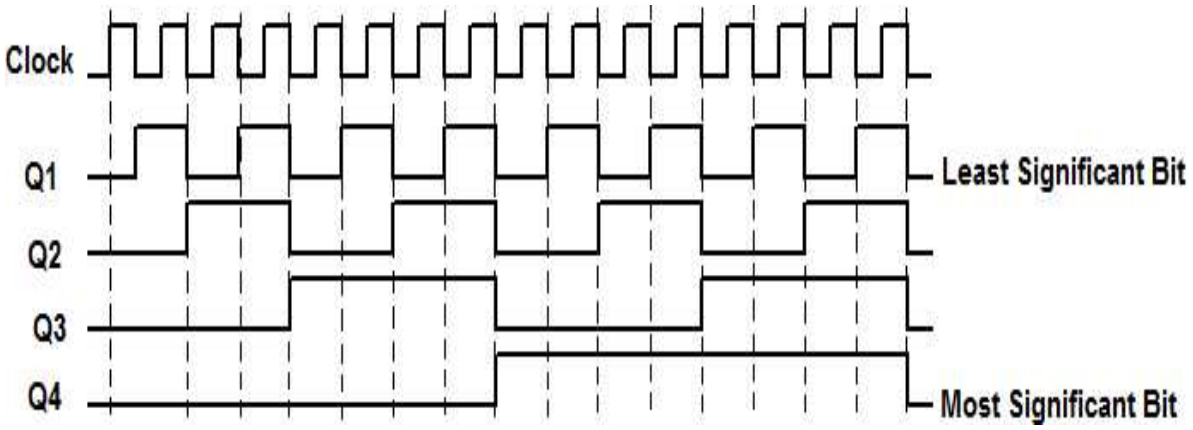
The 4 bit up counter shown in below diagram is designed by using JK flip flop. External clock pulse is connected to all the flip flops in parallel.

For designing the counters JK flip flop is preferred .The significance of using JK flip flop is that it can toggle its state if both the inputs are high, depending on the clock pulse.

The inputs of first flip flop are connected to HIGH (logic 1), which makes the flip flop to toggle, for every clock pulse entered into it. So the synchronous counter will work with single clock signal and changes its state with each pulse.



Clock	rst	bin_out(3)	bin_out(2)	bin_out(1)	bin_out(0)
X	X	0	0	0	0
↑	1	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
↑	1	1	1	0	0
↑	1	1	1	0	1
↑	1	1	1	1	0
↑	1	1	1	1	1



The output of first JK flip flop (Q) is connected to the input of second flip flop. The AND gates (which are connected externally) drives the inputs of other two flip flops . The inputs of these AND gates , are supplied from previous stage lip flop outputs.

If inputs of FF2 are connected directly to the Q1 output of FF1 , the counter would not function properly. This is because , the Q1 value is high at count of 210 , this means that the FF2 flip flop will toggle for the 3rd clock pulse. This results in wrong counting operation, gives the count as 710 instead of 410.

To prevent this problem AND gates are used at the input side of FF2 and FF3. The output of the AND gate will be high only when the Q0, Q1 outputs are high. So for the next clock pulse, the count will be 00012.

Similarly, the flip flop FF3 will toggle for the fourth clock pulse when Q0, Q1 and Q2 are high. The Q3 output will not toggle till the 8th clock pulse and will again remain high until 16th clock pulse. After the 16th clock pulse, the q outputs of all flip flops will return to 0.

RESULT: 4-bit Synchronous UP Counter, and 4-bit Asynchronous UP Counter are studied and verified.

VIVA QUESTIONS

1. Differentiate BCD & Excess 3 code
2. What is BCD?
3. What is the base for Binary?
4. How will you convert Binary to Gray?
5. What is Gray Code?
6. Give the conversion process of Excess 3 to BCD
7. What are the logic gates used in converting Gray code to Binary?
8. What is the need of code converters?
9. How will you design Code converters?
10. What is the difference between Positive and Negative logic?
11. Distinguish between Full Adder , 4 bit Binary adder
12. What is BCD?
13. How many pins are there in Ic 7483?
14. What is the need of Binary adder?
15. What is a Carry Look ahead Adder?
16. Is IC 7483 can be used for Adders and Subtractors? How?
17. Give the difference between a Half adder and a Parallel adder
18. How multipliers are utilized using adders
19. What is 4 bit subtraction?
20. Compare BCD adder , Binary adder
21. What is Mux?
22. What is the use of select lines?
23. What is a De-multiplexer?
24. Differentiate 2:1, 4:1 Multiplexer

25. What is called as Distributor? Why?
26. What is the use of IC 74150?
27. Why Mux is called as selector?
28. What is the use of IC 74154?
29. Will you design 8:1 Multiplexor using logic gates?
30. Why 74154 is used rather than logic gates?
31. What is a counter?
32. What are the types of counter?
33. Which flip-flop is most commonly used for counter applications?
34. Why asynchronous counter is called ripple counter?
35. What is Up/Down counter?
36. What are the applications of Flip-flops?
37. What is the use of a Clock signal?
38. What are the types of Triggering methods in F/Fs
39. What is a Register?
40. What is a Shift Register?
41. What is the basic device used in a Shift register?
42. What is the of Shift registers?
43. Give one application of shift register
44. What is SISO shift register? What is the IC used for it?
45. What is a Ring Counter?
46. What is a Bi-directional Shift register?
47. What is a PISO shift register?
48. Which is faster?

REFERENCE BOOKS:

- Malvino & Leach, “Digital Principles and Applications”, TMH.
- M. Morris Mano, “Digital Logic Design”, PHI
- R.P. Jain, “Modern Digital Design”, TMH.
- S. Salivahanan & S. Arivazhagan, “Digital Circuits and Design”, Vikas Publishing.
- D. Roy Chaudhuri, Digital Circuits, “An Introduction Part -1 & 2”, Eureka Publisher.
- Ronald J Tocci , “Digital Systems, Principles and Applications”, PHI.
- Taub & Schilling, “Digital Integrated Electronics”, TMH.