EXPERIMENT-1: Demonstrate Analog and digital signals

Using signal generator and oscilloscope trace and analyse various Analog and Digital signals like sinusoidal, triangle, saw tooth, pulse, and square wave.

EXPERIMENT-2: Gate characteristics and verification of basic and universal gates using IC's.

Aim:

To conduct an experiment to verify the truth-table of basic and universal gates using digital trainer kit and IC's

Components required:

Sl.no	Description	IC No	Quantity
01	IC Trainer kit	-	**1**
02	Patch chords	-	-
03	AND GATE	7408	1
04	OR GATE	7432	1
05	INVERTER GATE	7404	1
06	EXCLUSIVE OR GATE	7486	1
07	EXCLUSIVE NOR GATE	74266	1
08	NAND GATE	7400	1
09	NOR GATE	7402	1
10	EXCLUSIVE OR GATE	7486	1
11	EXCLUSIVE NOR GATE	74266	1

Theory:

Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/ output combination is called **Truth Table**. Various gates and their working is explained here.

AND Gates

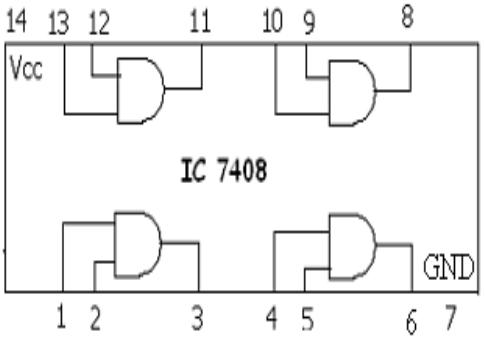
AND gate produces an output as 1, when all its inputs are 1; otherwise the output is 0. This gate can have minimum 2 inputs but output is always one. Its output is 0 when any input is 0.

SYMBOL

TRUTHTABLE

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

IC7408



OR Gate(IC 7432**)**

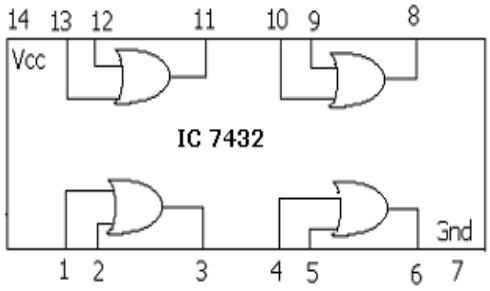
OR gate produces an output as 1, when any or all its inputs are 1; otherwise the output is 0. This gate can have minimum 2 inputs but output is always one. Its output is 0 when all input are 0.

SYMBOL

TRUTH TABLE

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

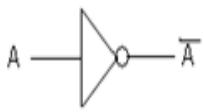
IC 7432



NOT Gate (IC 7404)

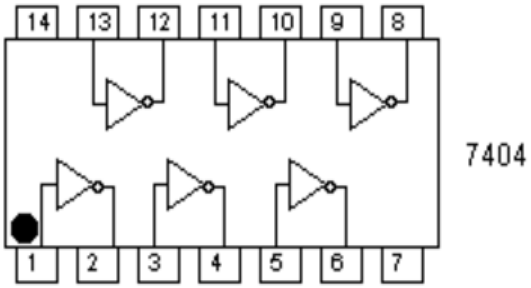
NOT gate produces the complement of its input. This gate is also called an INVERTER. It always has one input and one output. Its output is 0 when input is 1 and output is 1 when input is 0.

SYMBOL



TRUTH TABLE

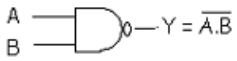
I/P (A)	O/P $\overline{(A)}$
0	1
1	0



NAND Gate (IC7400)

NAND gate is actually a series of AND gate with NOT gate. If we connect the output of an AND gate to the input of a NOT gate, this combination will work as NOT-AND or NAND gate. Its output is 1 when any or all inputs are 0, otherwise output is 1

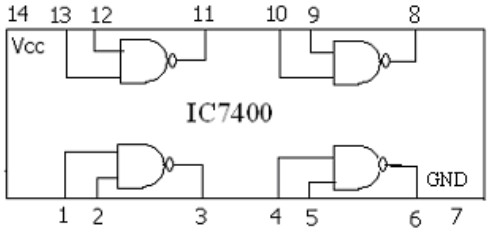
SYMBOL



TRUTHTABLE

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

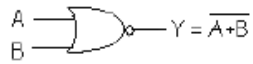
IC7408



NOR Gate (IC7402)

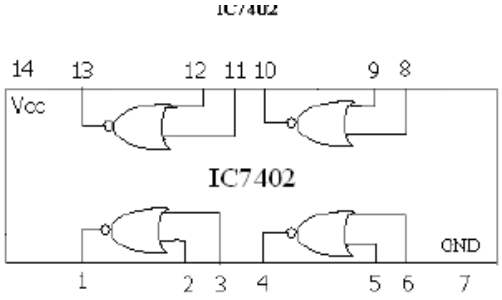
NOR gate is actually a series of OR gate with NOT gate. If we connect the output of an OR gate to the input of a NOT gate, this combination will work as NOT-OR or NOR gate. Its output is 0 when any or all inputs are 1, otherwise output is 1.

SYMBOL



TRUTH TABLE

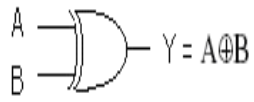
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



Exclusive OR (X-OR) Gate (IC7486)

X-OR gate produces an output as 1, when number of 1's at its inputs is **odd**, otherwise output is 0. It has two inputs and one output.

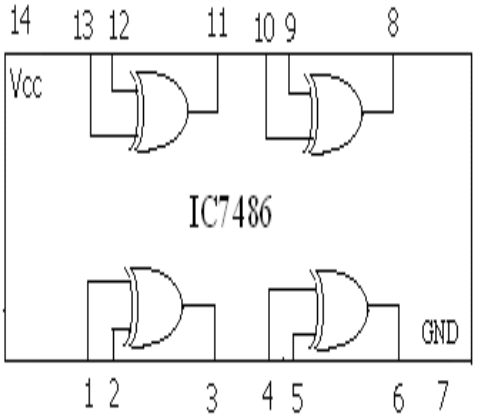
SYMBOL



TRUTH TABLE

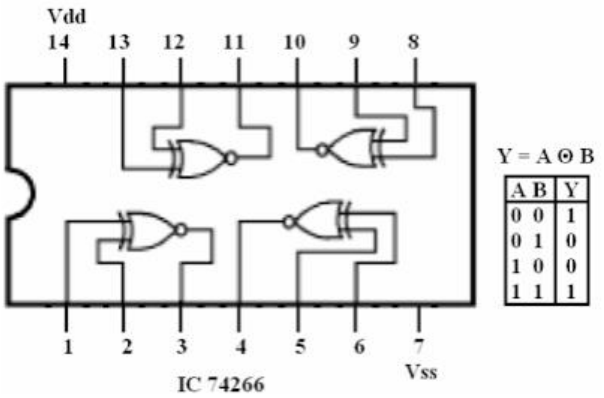
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

IC 7486



Exclusive NOR (X-NOR) Gate

X-NOR gate produces an output as 1, when number of 1's at its inputs is **not odd**, otherwise output is 0. It has two inputs and one output.



PROCEDURE

1. Collect the components necessary to accomplish this experiment.
2. Plug the IC chip into the IC Trainer kit.
3. Connect the supply voltage and ground lines to the chips. PIN7 = Ground and PIN14 = +5.
4. According to the pin-out diagram of each IC mentioned above, wire only one gate to verify its truth table.
5. Once all connections have been done, turn on the power switch of the Trainer kit.
6. Operate the switches and fill in the truth table.
7. Repeat the above steps for each IC package.
8. After finishing the experiment, turn off the power switch, disconnect the wires, take out all IC chips from the trainer, put back everything you have used and clean your table.

Result:

Exercise :

1) **Design and implement code converter**
   - **Binary to Gray**
   - **Gray to Binary Code using basic gates**

EXPERIMENT-3

Aim: Conduct an experiment to realize the universal gates using basic gates.

Experiment-4

Conduct an experiment to realize basic gates using the universal gates.

Aim

Realization of logic functions with the help of universal gates-NAND Gate.

Components required :.

Sl.no	Description	IC No
01	IC Trainer kit	-
02	Patch chords	-
03	NAND GATE	7400
04	NOR GATE	7402

Theory:

NAND gate is actually a combination of two logic gates: AND gate followed by NOT gate. So its output is complement of the output of an AND gate.

This gate can have minimum two inputs, output is always one. By using only NAND gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NOR. So this gate is also called universal gate.

NAND gates as NOT gate

A NOT produces complement of the input. It can have only one input, tie the inputs of a NAND gate together. Now it will work as a NOT gate. Its output is

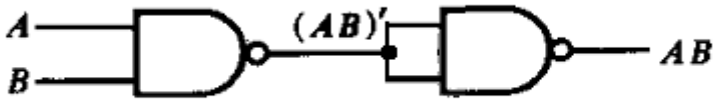
$$Y = (A.A)'$$
$$Y = (A)'$$

NOT (inverter)

NAND gates as AND gate

A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.

$$Y = ((A.B)')'$$
$$Y = (A.B)$$



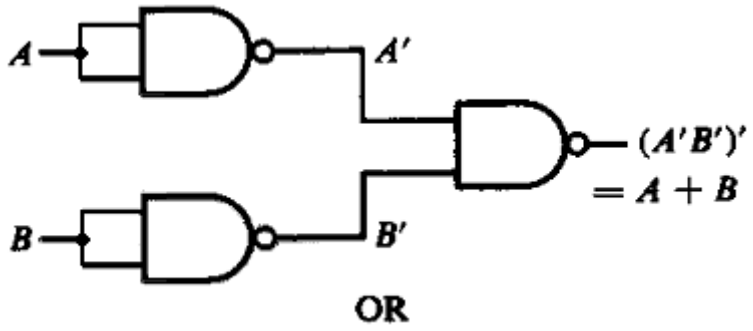
AND

NAND gates as OR gate

From DeMorgan's theorems: (A.B)' = A' + B'

(A'.B')' = A'' + B'' = A + B

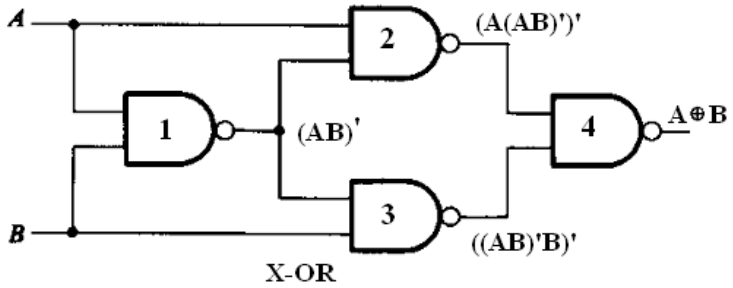
So, give the inverted inputs to a NAND gate, obtain OR operation at output.



OR

NAND gates as X-OR gate

The output of a to input X-OR gate is shown by: Y = A'B + AB'. This can be achieved with the logic diagram shown in the left side.



X-OR

Gate No.	Inputs	Output
1	A, B	(AB)'
2	A, (AB)'	(A (AB)')'
3	(AB)', B	(B (AB)')'
4	(A (AB)')', (B (AB)')'	A'B + AB'

Now the ouput from gate no. 4 is the overall output of the configuration.

Y = ((A (AB)')' (B (AB)')')'

= (A(AB)'')' + (B(AB)')''

= (A(AB)') + (B(AB)')

= (A(A' + B)') + (B(A' + B'))

= (AA' + AB') + (BA' + BB')

= (0 + AB' + BA' + 0)

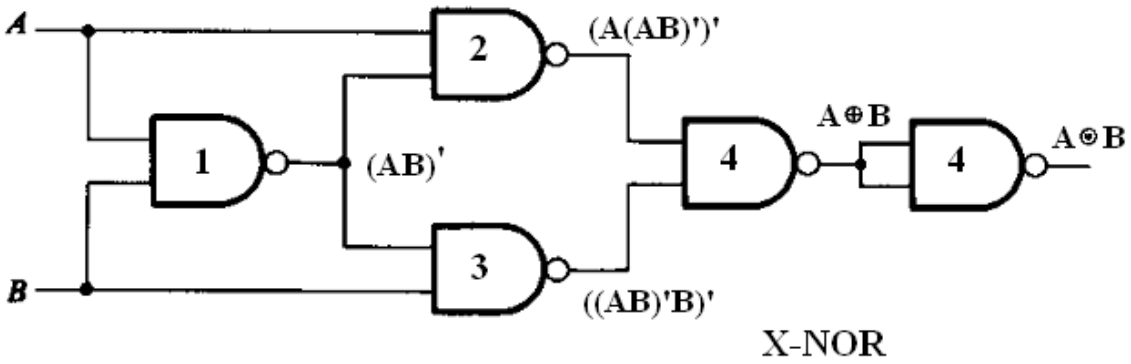
= AB' + BA'

Y = AB' + A'B

NAND gates as X-NOR gate

X-NOR gate is actually X-OR gate followed by NOT gate. So give the output of X-OR gate to a NOT gate, overall ouput is that of an X-NOR gate.

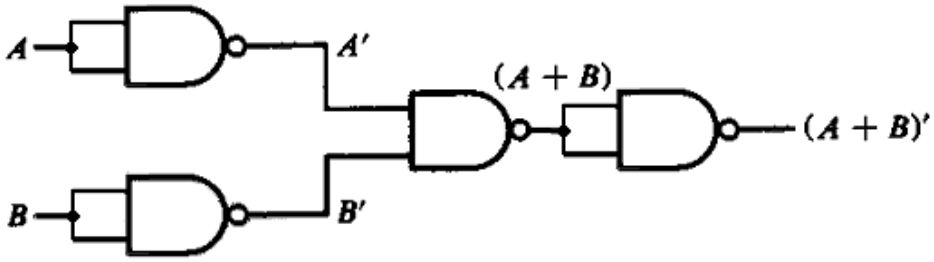
$$Y = AB + A'B'$$



X-NOR

NAND gates as NOR gate

A NOR gate is an OR gate followed by NOT gate. So connect the output of OR gate to a NOT gate, overall output is that of a NOR gate. $Y = (A + B)'$



NOR

Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the NAND gates for any of the logic functions to be realised.
3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply various input combinations and observe output for each one.
5. Verify the truth table for each input/ output combination.
6. Repeat the process for all logic functions.
7. Switch off the ac power supply.

Result:

Aim

(b)Realization of logic functions with the help of universal gates-NOR Gate.

Components required: logic trainer kit, NOR gates (IC 7402), wires.

Theory:

NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate. So its output is complement of the output of an OR gate.

This gate can have minimum two inputs, output is always one. By using only NOR gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NAND. So this gate is also called universal gate.

NOR gates as NOT gate

A NOT produces complement of the input. It can have only one input, tie the inputs of a NOR gate together. Now it will work as a NOT gate. Its output is

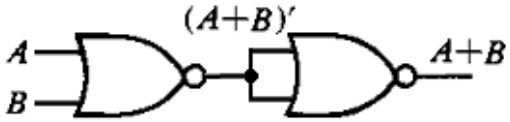
$$Y = (A+A)'$$
$$Y = (A)'$$

NOT

OR gates as OR gate

A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.

$$Y = ((A+B)')'$$
$$Y = (A+B)$$



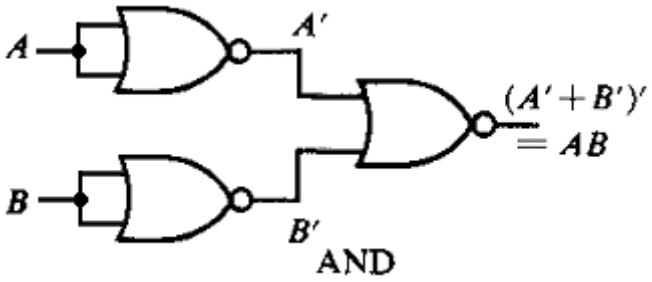
OR

OR gates as AND gate

From DeMorgan's theorems: $(A+B)' = A'B'$

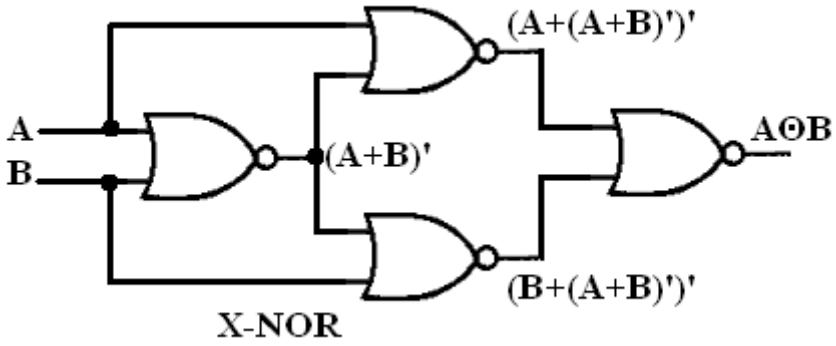
$$(A'+B')' = A''B'' = AB$$

So, give the inverted inputs to a NOR gate, obtain AND operation at output.



NOR gates as X-NOR gate

The output of a two input X-NOR gate is shown by: Y = AB + A'B'. This can be achieved with the logic diagram shown in the left side.



Gate No.	**Inputs**	**Output**
1	A, B	(A + B)'
2	A, (A + B)'	(A + (A+B)')'
3	(A + B)', B	(B + (A+B)')'
4	(A + (A + B)')', (B + (A+B)')'	AB + A'B'

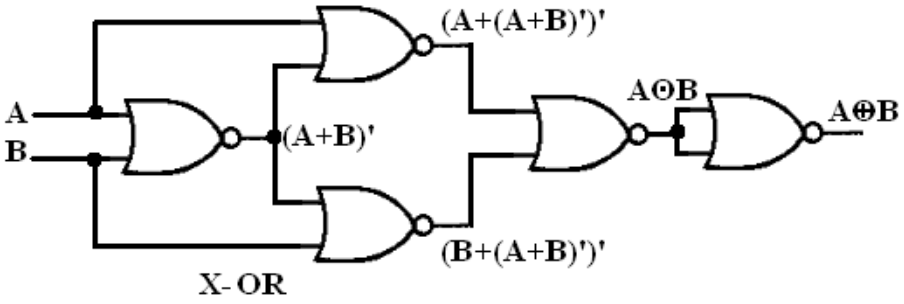
Now the ouput from gate no. 4is the overall output of the configuration.

Y = ((A + (A+B)')' (B +(A+B)')')'
= (A+(A+B)'')'.(B+(A+B)'')''
= (A+(A+B)').(B+(A+B)')
= (A+A'B').(B+A'B')
= (A + A').(A + B').(B+A')(B+B')
= 1.(A+B').(B+A').1
= (A+B').(B+A')
= A.(B + A') +B'.(B+A')
= AB + AA' +B'B+B'A'
= AB + 0 + 0 + B'A'
= AB + B'A'
Y = AB + A'B'

NOR gates as X-OR gate

X-OR gate is actually X-NOR gate followed by NOT gate. So give the output of X-NOR gate to a NOT gate, overall ouput is that of an X-OR gate.

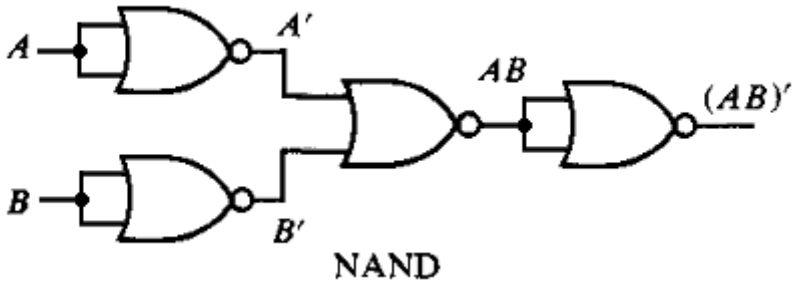
$$Y = A'B + AB'$$



NOR gates as NAND gate

A NAND gate is an AND gate followed by NOT gate. So connect the output of AND gate to a NOT gate, overall output is that of a NAND gate.

$$Y = (AB)'$$



Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the NOR gates for any of the logic functions to be realised.
3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply various input combinations and observe output for each one.
5. Verify the truth table for each input/ output combination.
6. Repeat the process for all logic functions.
7. Switch off the ac power supply.

Experiment 5: DeMorgan's Theorem

Aim:

- To verify the basic rules of Boolean algebra using logic gates
- To verify the two elements of DeMorgan's Theorem
- To develop digital circuit building and troubleshooting skills

Components:

Sl.no	Description	IC No
01	IC Trainer kit	-
02	Patch chords	-
03	NAND GATE	7400
04	NOR GATE	7402
05	AND GATE	7408
06	OR GATE	7432
07	INVERTER (NOT GATE)	7404

Discussion:

Boolean algebra is similar to other mathematical topics in that there are identities that can be used to simplify the work that needs to be done. The basic identities of Boolean algebra are:

A+0=A

A+1=1

A+A=A

A + A' = 1

A*0 = 0

A*1 = A

AA = A

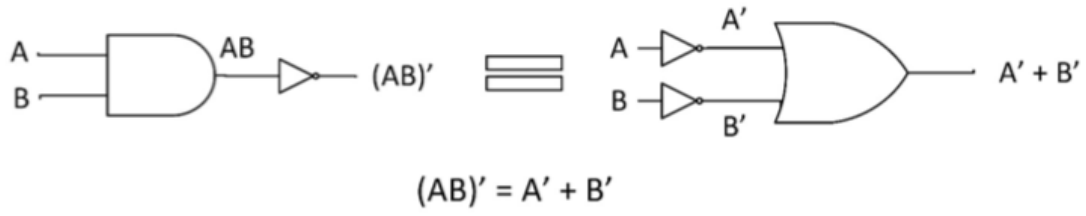
AA' = 0

A'' = A

A + AB = A
A + A'B = A + B

Since seeing is believing, this experiment will demonstrate all of the single variable Boolean identities using AND, OR and NOT gates in this experiment.

Recall that the DeMorgan's Theorem (Figure 1) simplifies the inverted product (AB)' or inverted sum (C + D)' into a form that is more usable form for overall simplification of a digital circuit. This theorem also provides a means of converting to universal gate NAND and NOR. Both parts of DeMorgan's Theorem will be proven in this experiment.



$(AB)' = A' + B'$

$(C + D)' = C'D'$

Figure 1

Procedure:

Boolean Identities

1. Design and draw the logic diagrams for the equivalent Boolean Identity circuits in **below table** using the AND, OR and NOT gates. Include the pin numbers on the gate inputs and outputs. Reference the datasheets to determine the pinout of the chips.
2. Build each circuit from procedure 1.1 using the trainer. Test each circuits output for both input states. Use the Level Switches to toggle the inputs.
3. Before turning on the power to the trainer review the wiring for errors. If at any time during the experiment the results do not match the theoretical or expected results use the logic probe to troubleshoot the circuit.
4. Turn on the power to the trainer. Record the results in Table 1.

Draw the equivalent Boolean identity circuit

Boolean Identity	Experimental Result (A, 1, 0)
A+0=A	
A+1=1	
A+A=A	
A + A’ = 1	
A*0 = 0	
A*1 = A	
AA = A	
AA’ = 0	
A’’ = A	

5. Describe your results and comment on any problems encountered during the build and testing the circuits for Table 1?

2. DeMorgan’sTheorem

1. Design and draw the first element of DeMorgan’s Theorem **(AB)’ = A’ + B’** using the AND, OR and NOT gates. Include the pin numbers on the gate inputs and outputs. Reference the datasheets to determine the pinout of the chips.

(AB)’

A’ + B’

2. Use the AND, OR and NOT gates along with the trainer to build the two logic circuits from the first DeMorgan’s Theorem **(AB)’ = A’ + B’**.Use the Level Switches to toggle the inputs.
3. Before turning on the power to the trainer review the wiring for errors.
4. Start with all the input switches as LOW. Turn on the power to the trainer. If at any time during the experiment the results do not match the theoretical or expected results use the logic probe to troubleshoot the circuit.
5. Adjust the switches to test all the input combinations and record the results in below table .

Inputs		Outputs	
A	B	(AB)'	A' + B'

Table 2

Comment on the results of above Table

Design and draw the second element of DeMorgan's Theorem **(C + D)' = C'D'** using the AND, OR and NOT gates. Include the pin numbers on the gate inputs and outputs. Reference the datasheets to determine the pinout of the chips.

(C + D)'

C'D'

- Use the AND, OR and NOT gates along with the trainer to build the two logic circuits from the second DeMorgan's Theorem **(C + D)' = C'D'**. Use the Level Switches to toggle the inputs.
- Before turning on the power to the trainer review the wiring for errors.
- Start with all the input switches as LOW. Turn on the power to the trainer. If at any time during the experiment the results do not match the theoretical or expected results use the logic probe to troubleshoot the circuit.
- Adjust the switches to test all the input combinations and record the results in Table 3.

Inputs		Outputs	
C	D	(C + D)'	C'D'

Table 3

2.12. Comment on the results of Table 3.

Questions:

1. Simplify the Boolean equation M = W'XY' + WXY' + XY'Z using Boolean identities and theorems to show that M = XY'

2. Draw the logic diagrams for **X=AB(A'+BC)'** and **Y=ABC'**.

3. Using Boolean identities and the DeMorgan's Theore prove that X=Y from question 2.

Experiment 6: MUX and DEMUX

4:1 MULTIPLEXER USING GATES

Aim: To design and set up a 4:1 Multiplexer (MUX) using gates. Components Required: IC 74153, IC 7404, 7432, 7411 and Patch Cords.

IC Trainer Kit.

Theory:

Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals.

Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. By using control signals (select lines) we can select any input to the output. Multiplexer is also called as data selector because the output bit depends on the input data bit that is selected. The general multiplexer circuit has 2n input signals, n control/select signals and 1 output signal.

Procedure:

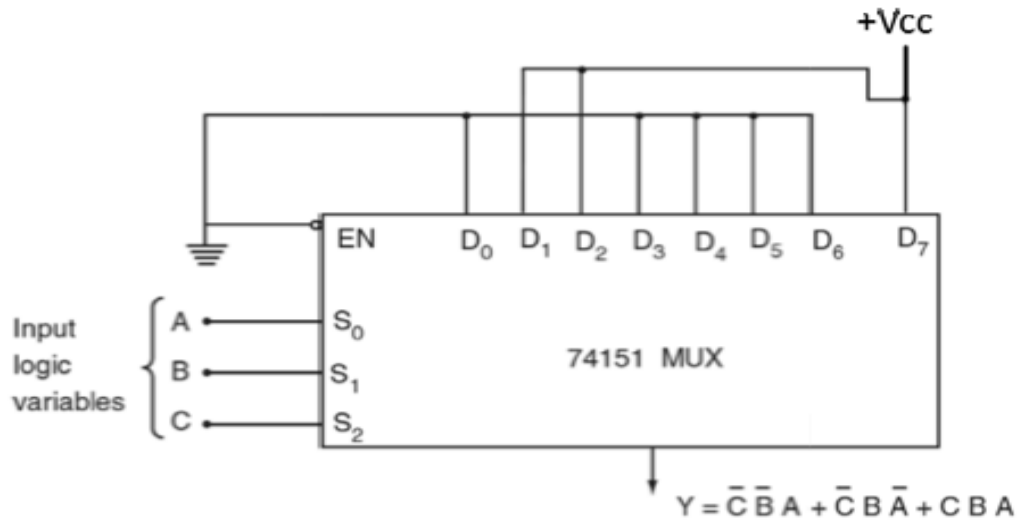
1. The connection is made as shown in the diagram.

2. Here, S1 and S0 are the channel selection lines, I0, I1, I2, I3 are the

respective data lines of the channels and Y is the output.

3. Based on the selection lines one of the inputs will be selected at the output, and thus the truth table is verified.

B. 3 Variable function using IC 74151



Select lines			Output
C	**B**	**A**	**Y**
0	**0**	**0**	**0**
0	**0**	**1**	**1**
0	**1**	**0**	**1**
0	**1**	**1**	**0**
1	**0**	**0**	**0**
1	**0**	**1**	**0**
1	**1**	**0**	**0**
1	**1**	**1**	**1**

8:1 MULTIPLEXER USING 74151

Aim: To design and set up the following circuit
3 variable function using IC 74151 (8:1 MUX)

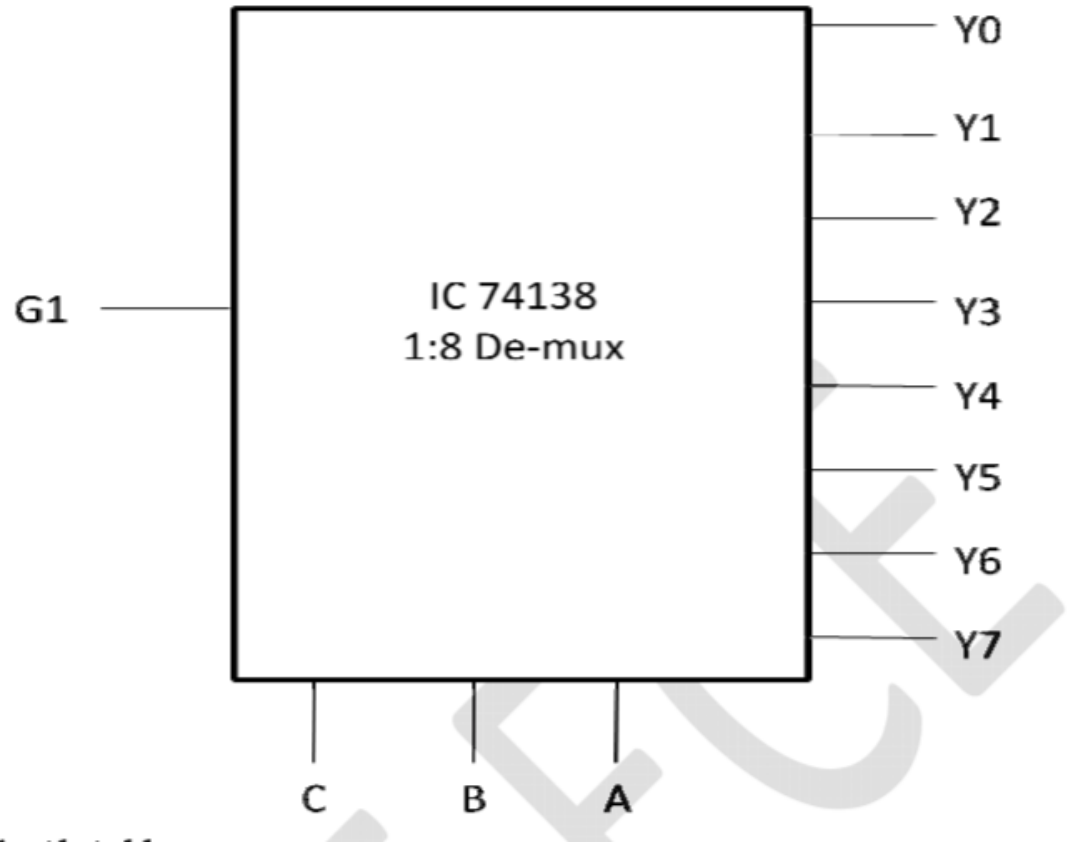
Components Required: IC 74151, Patch Cords & IC Trainer Kit.

Procedure:

1. For the given expression, a truth table is to be written.
2. An expression in SOP format is to be written.
3. The connection is made according to the obtained expression. 4. The truth table is verified for that particular expression.

Result:

1:8 De-mux and 3:8 Decoder using IC 74138



Truth table:

Select Lines			Output							
C	**B**	**A**	**Y0**	**Y1**	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	**Y7**
0	0	0	G1	1	1	1	1	1	1	1
0	0	1	1	G1	1	1	1	1	1	1
0	1	0	1	1	G1	1	1	1	1	1
0	1	1	1	1	1	G1	1	1	1	1
1	0	0	1	1	1	1	G1	1	1	1
1	0	1	1	1	1	1	1	G1	1	1
1	1	0	1	1	1	1	1	1	G1	1
1	1	1	1	1	1	1	1	1	1	G1

1:8 DEMUX AND 3:8 DECODER USING IC 74138

Aim: To design 1:8 Demux and 3:8 Decoder using IC 74138 Components Required: IC 74138, Patch Cords & IC Trainer Kit.

Theory:

A demultiplexer (or demux) is a device that takes a single input line and routes it to one of several digital output lines. A demultiplexer of 2^n outputs has n select lines, which are used to select which output line to send the input. A demultiplexer is also called a data distributor.

A decoder is a circuit which has n inputs and 2^n outputs, and outputs 1 on the wire corresponding to the binary number represented by the inputs. A standard decoder typically has an additional input called Enable. Output is only generated when the Enable input is activated.

Procedure (1:8 Demux):

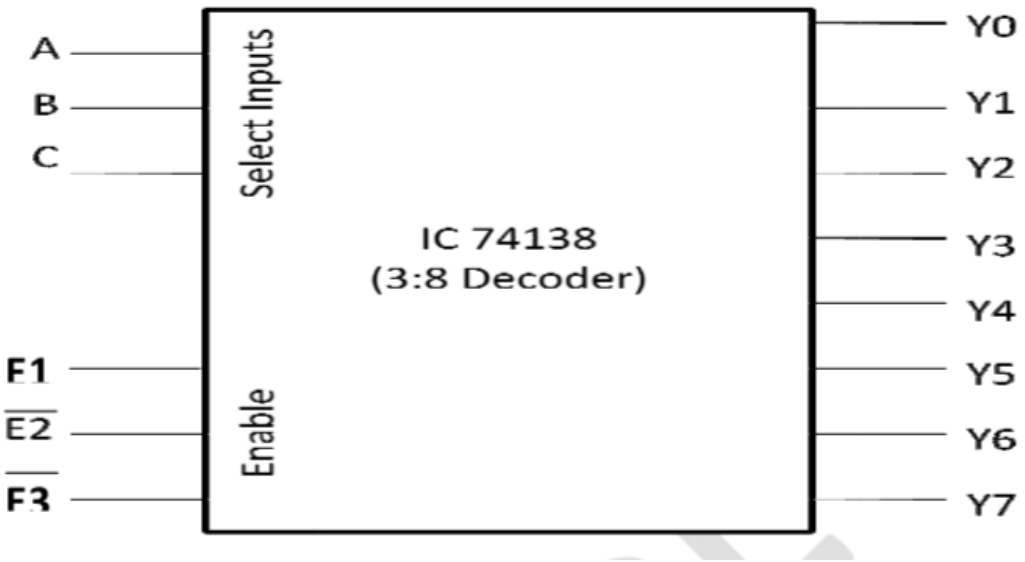
1. The pins indicated as A, B and C are the selection lines, G1 acts as an input line, Y0 to Y7 are the output channels.

2. A particular channel can be selected by choosing the proper bits for A, B and C lines.

3. As IC 74138 is having active low outputs, initially all the outputs will be high

4. Based on the bit given to G1 after selecting the channel, the same bit will be reflected in that particular channel.

5. The truth table is verified for different combination of bits on both selection lines and input G1.



Truth table:

Enable			Select Lines			Output							
E3	E2	E1	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Procedure (3:8 DECODER):

1. The connections are made as shown in the diagram
2. Enable pins are used to enable the IC
3. Using Select inputs, a particular output is selected.
4. Since IC 74138 is having an active low output, the particular line will

   give a digital low.

Result:

Experiment 6: FLIP FLOPS

Aim: To realize the following Flip-flops using NAND gates: A. Clocked SR Flip-flop

Components required: IC 7410, IC7400

Theory:

A flip-flop is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

A flip–flop is a “bit bucket”; it holds a single binary bit .Flip flops are actually an application of logic gates. With the help of Boolean logic we can create memory with them. Flip flops can also be considered as the most basic idea of a Random Access Memory [RAM].

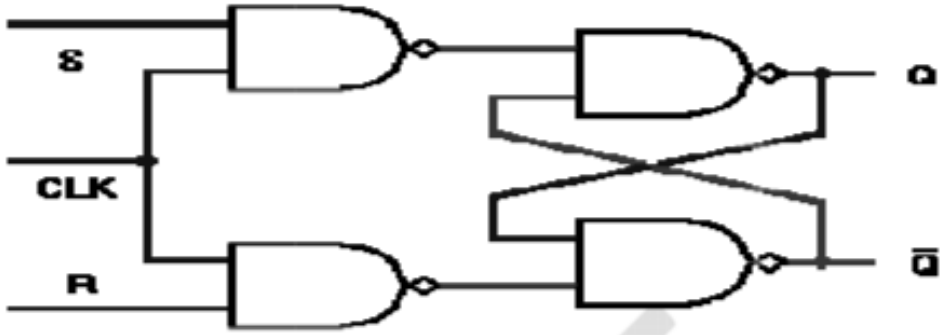
The most commonly used application of flip flops is in the implementation of a feedback circuit. As a memory relies on the feedback concept, flip flops can be used to design it.

Procedure:

1. Make the connections as shown in the circuit diagrams.

2. Apply inputs as shown in the truth tables,
3. Check the outputs of the circuits; verify that they match wit5h the truth

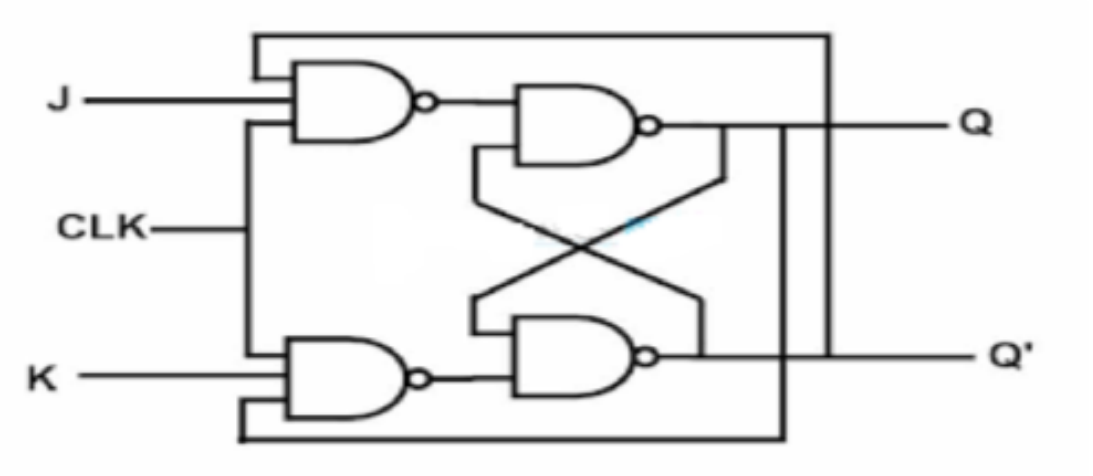
1. SR FLIPFLOP:



Truth table:

Clk	S	R	Q	Q'	States
X	0	0	Q	Q'	No Change
↑	0	0	Q	Q'	No Change
↑	0	1	0	1	Reset
↑	1	0	1	0	Set
↑	1	1	-	-	Invalid

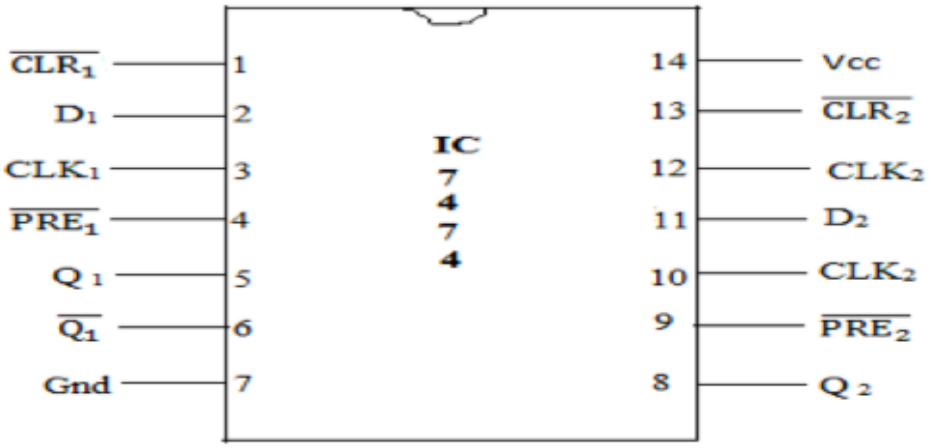
2. JK FLIPFLOP:



Clk	J	K	Q	Q'	States
X	0	0	Q	Q'	No Change
↑	0	0	Q	Q'	No Change
↑	0	1	0	1	Reset
↑	1	0	1	0	Set
↑	1	1	Q'	Q	Toggle

D – Flip – Flop:

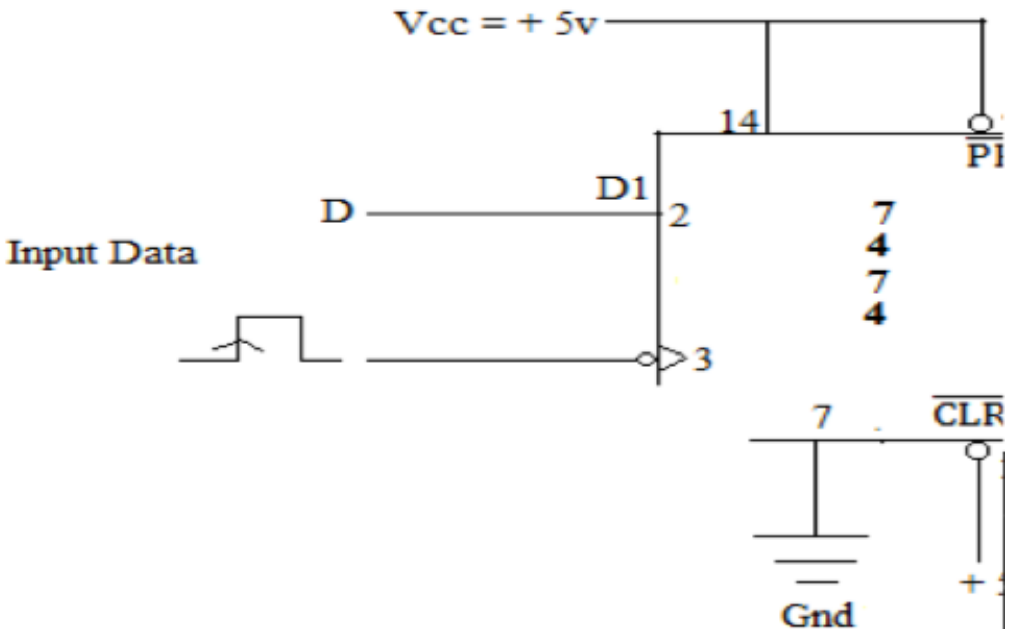
IC – 7474: Dual + ve edge triggered D- Flip Flop :



Inputs		Outputs
Q	J	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

Fig (8.3) Pin Diagram

Circuit Implementation:



T FLIP FLOP:

Inputs		Outputs
Q	J	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

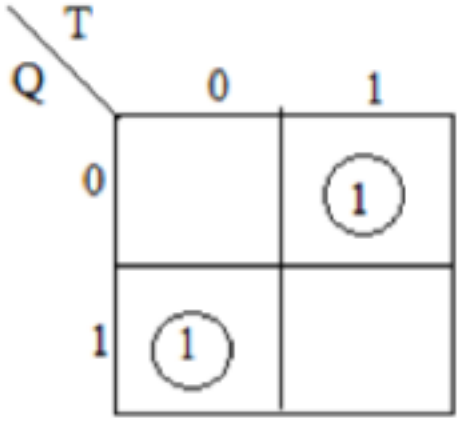
T – Flip – Flop:

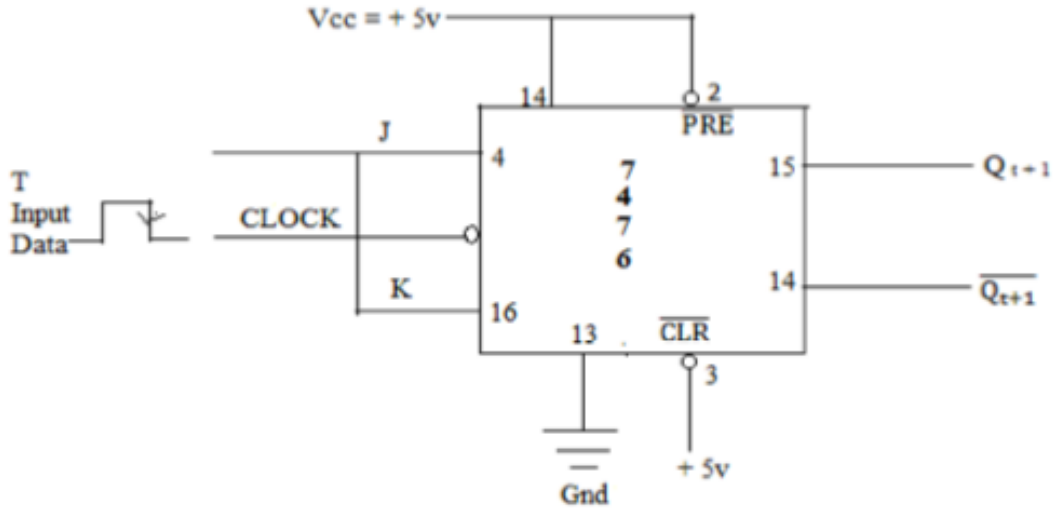
Fig (8.7) : T – Flip - Flop Circuit

Where

Q ⟶ Present State

D ⟶ Data Input





The next state of T Flip Flop is equal to Ex –OR of Present State (Q) and T input.

PROCEDURE:

1. 1) Connections are made as per the circuit diagram.
2. 2) Apply the –ve edge triggered, +ve edge triggered and level sensitive clock pulses as required.
3. 3) Verify the truth table of all the Flip – Flops.
4. 4) Switch - off the power supply and disconnect the circuit.

RESULT:

Experiment 8 : Shift Registers

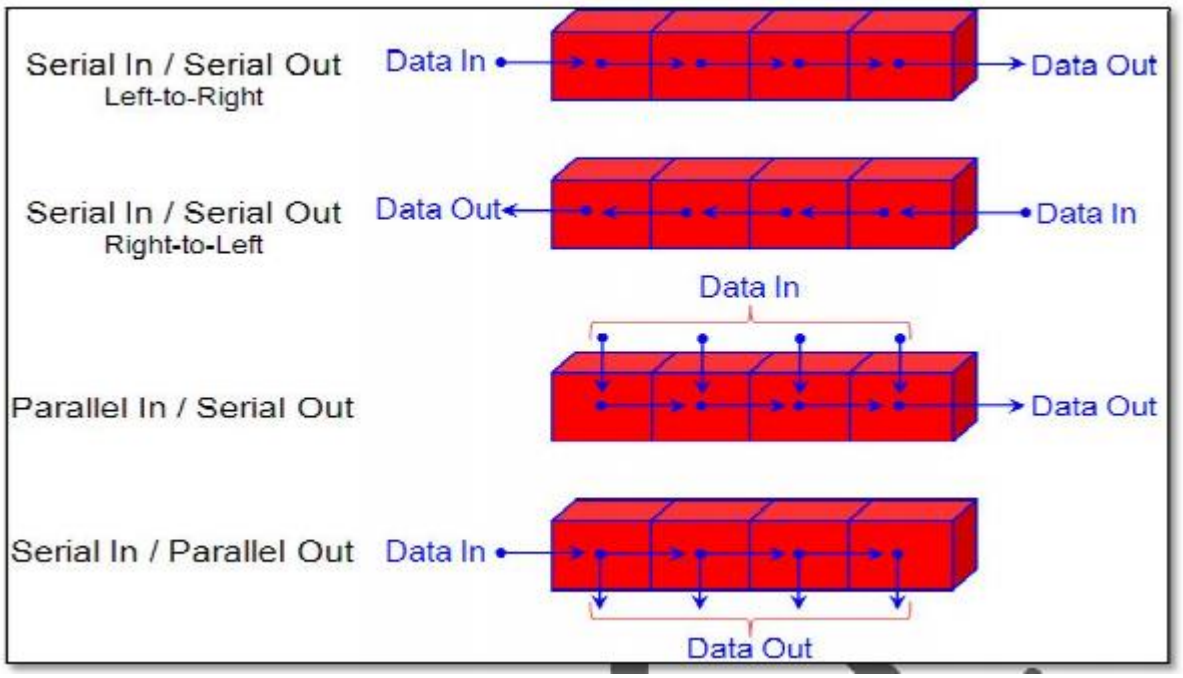
Conduct an experiment to understand the operation of shift registers.

- **SISO**
- **SIPO**
- **PISO**
- **PIPO**

Aim : - To study IC 7474, and the realization of SIPO, SISO, PISO, PIPO operations using the same.

Apparatus required : IC 7474, patch cards etcs.

Theory : - A register is simply a group of flip-flop that can be used to store a binary no. of a group of Flip- flop connected to provide either or both of these function is called shift register. To allow the data in the word to read in to the register serially. The o/p of the flip-flop is connected to the i/p of the following binary such a configuration called a Shift register. There are two ways to shift the data into a register (serial and parallel) and similarly two ways to shift the data out of the register. This leads to construction of four types of registers.



1. **SERIAL INPUT SERIAL OUTPUT (SISO):**

Procedure:

1. Connections are made as shown in the SISO circuit diagram.

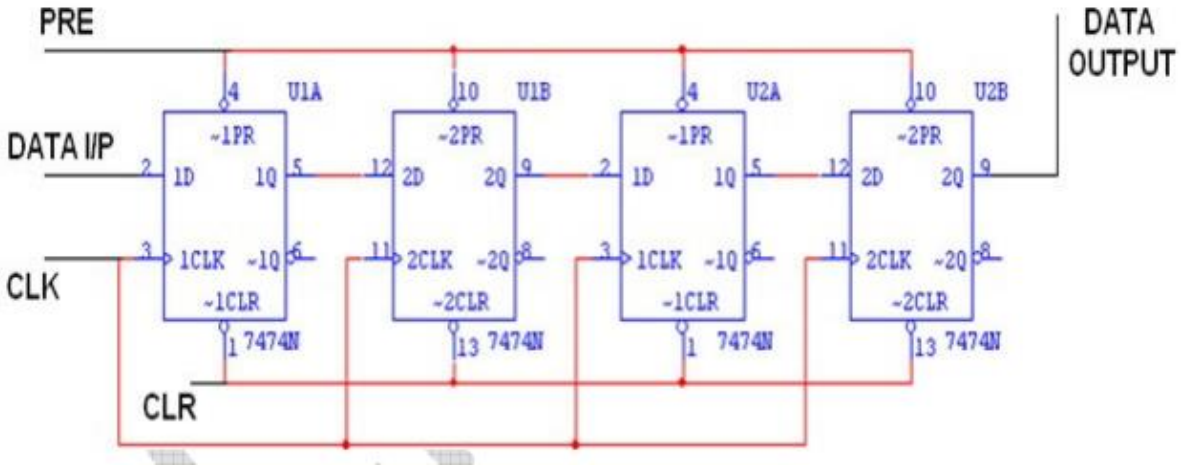
2. The shift register is loaded with 4 bits of data one by one serially.

3. At the end of the 4th clock pulse, the first data ‘d0’ appears at Q0.

4. Applying another clock pulse will push the second data bit, ‘d1’ to Q0.

5. Applying yet another clock pulse gives the third data bit, ‘d2’ at Q0, and so on.

Circuit Diagram :



Truth Table :

CLK	Serial I/P	Q3	Q2	Q1	Q0
1	D0=0	0	X	X	X
2	D1=1	1	0	X	X
3	D2=1	1	1	0	X
4	D3=1	1	1	1	0=D0
5	X	X	1	1	1=D1
6	X	X	X	1	1=D2
7	X	X	X	X	1=D3

2. SERIAL INPUT PARALLEL OUTPUT (SIPO):

Procedure:

1. Connections are made as shown in the SIPO circuit diagram.

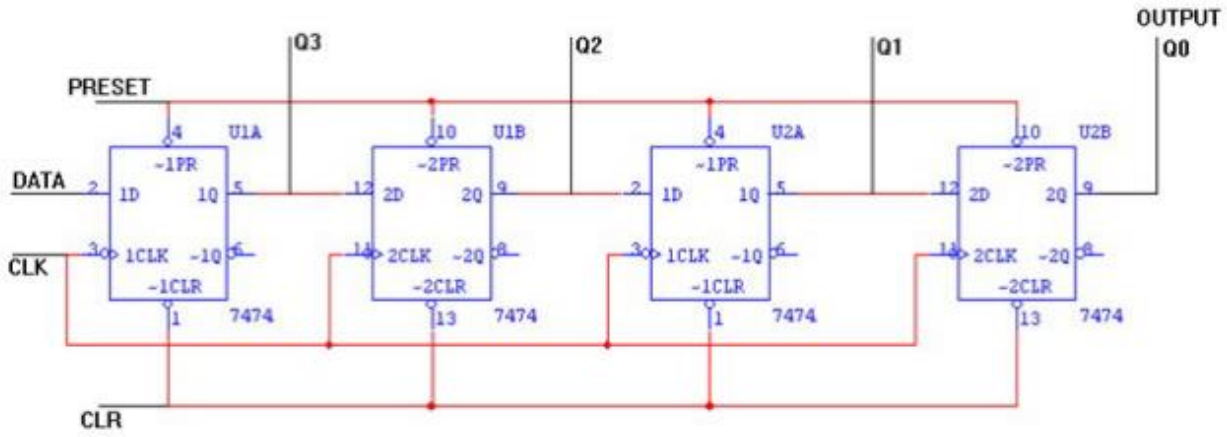
2. On applying the first bit of data and then a clock pulse, it can be observed that this data appears at (Q3).

3. Now, applying the second bit of data and a clock pulse, the bit at Q3 shifts to Q2 and Q3 will be loaded with the new data.

4. This repeats until all 4 data bits are loaded.

5. At the end of the 4th clock pulse, all 4 bits are available at the parallel output pins Q3 through Q0.

Circuit Diagram :



Truth Table :

CLK	Serial I/P	Q3	Q2	Q1	Q0
1	1	1	X	X	X
2	0	0	1	X	X
3	1	1	0	1	X
4	1	1	1	0	1

3. PARALLEL INPUT PARALLEL OUTPUT (PIPO):

Procedure:

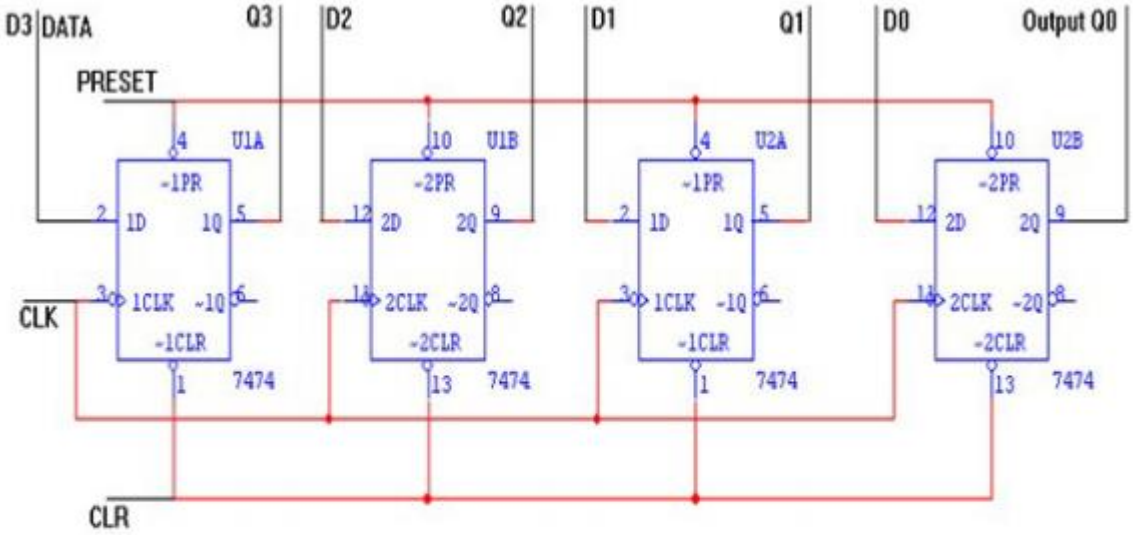
1. Connections are made as shown in the PIPO mode circuit diagram.

2. Apply the 4 data bits as input to D3, D2, D1, D0.

3. Apply one clock pulse

4. Note that the 4 bit data at parallel inputs appears at the parallel output pins Q3, Q2, Q1, Q0 respectively.

Circuit Diagram :



Truth Table:

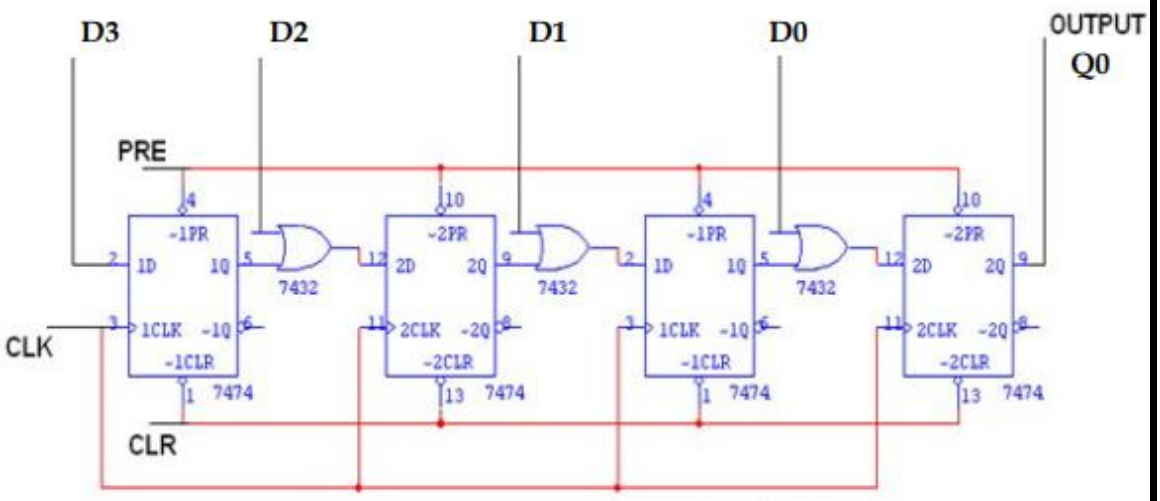
	Parallel I/P				Parallel O/P			
CLK	**D3**	**D2**	**D1**	**D0**	**Q3**	**Q2**	**Q1**	**Q0**
1	1	0	1	1	1	0	1	1

4. PARALLEL INPUT SERIAL OUTPUT (PISO):

Procedure:

1. Connections are made as shown in the PISO circuit diagram.
2. Apply the 4-bit data at the parallel I/P pins D3, D2, D1, D0 and apply single clock pulse to load the data to all 4 registers.
3. Now make all data bits as 0 and apply clock pulse one by one to get the data bit by bit at the output line.
4. The data applied at the parallel input pins will shift and comes out serially at the output lineQ0.

Circuit Diagram :



Truth Table:

Mode	Clock	Parallel I/P				O/P			
M	CLK	D3	D2	D1	D0	Q3	Q2	Q1	Q0
1	1	**1**	**0**	**1**	**1**	1	0	1	**1**
0	2	0	0	0	0	X	1	0	**1**
0	3	0	0	0	0	X	X	1	**0**
0	4	0	0	0	0	X	X	X	**1**

Experiment 9: Asynchronous Mod-N counters.

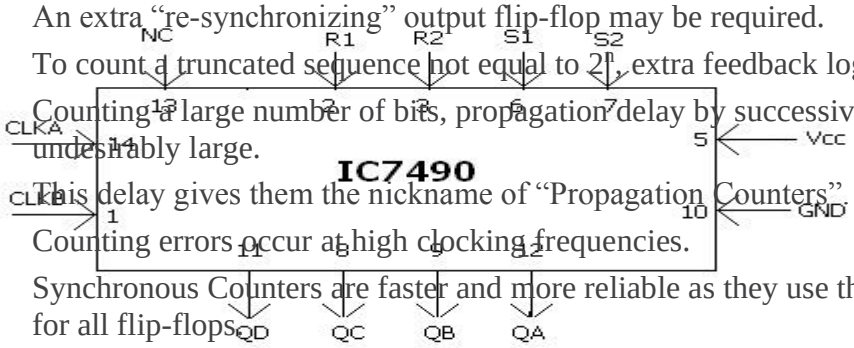
Conduct an experiment to verify the operation of a counter using IC-7490.

Then to summarise some of the advantages of Asynchronous Counters:

- *Asynchronous Counters* can easily be made from Toggle or D-type flip-flops.
- They are called "Asynchronous Counters" because the clock input of the flip-flops are not all driven by the same clock signal.
- Each output in the chain depends on a change in state from the previous flip-flops output.
- Asynchronous counters are sometimes called ripple counters because the data appears to "ripple" from the output of one flip-flop to the input of the next.
- They can be implemented using "divide-by-n" counter circuits.
- Truncated counters can produce any modulus number count.

Disadvantages of Asynchronous Counters:

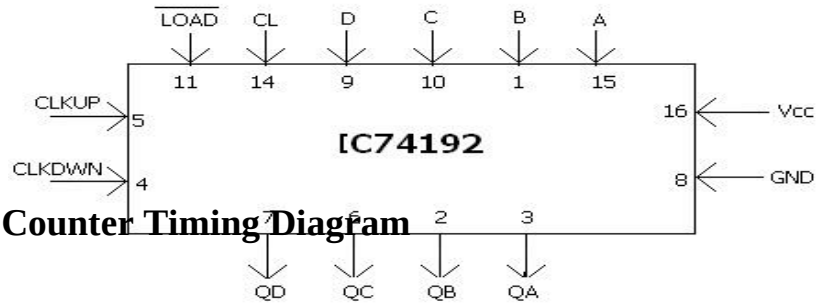
- An extra "re-synchronizing" output flip-flop may be required.
- To count a truncated sequence not equal to 2^n, extra feedback logic is required.
- Counting a large number of bits, propagation delay by successive stages may become undesirably large.
- This delay gives them the nickname of "Propagation Counters".
- Counting errors occur at high clocking frequencies.
- Synchronous Counters are faster and more reliable as they use the same clock signal for all flip-flops.



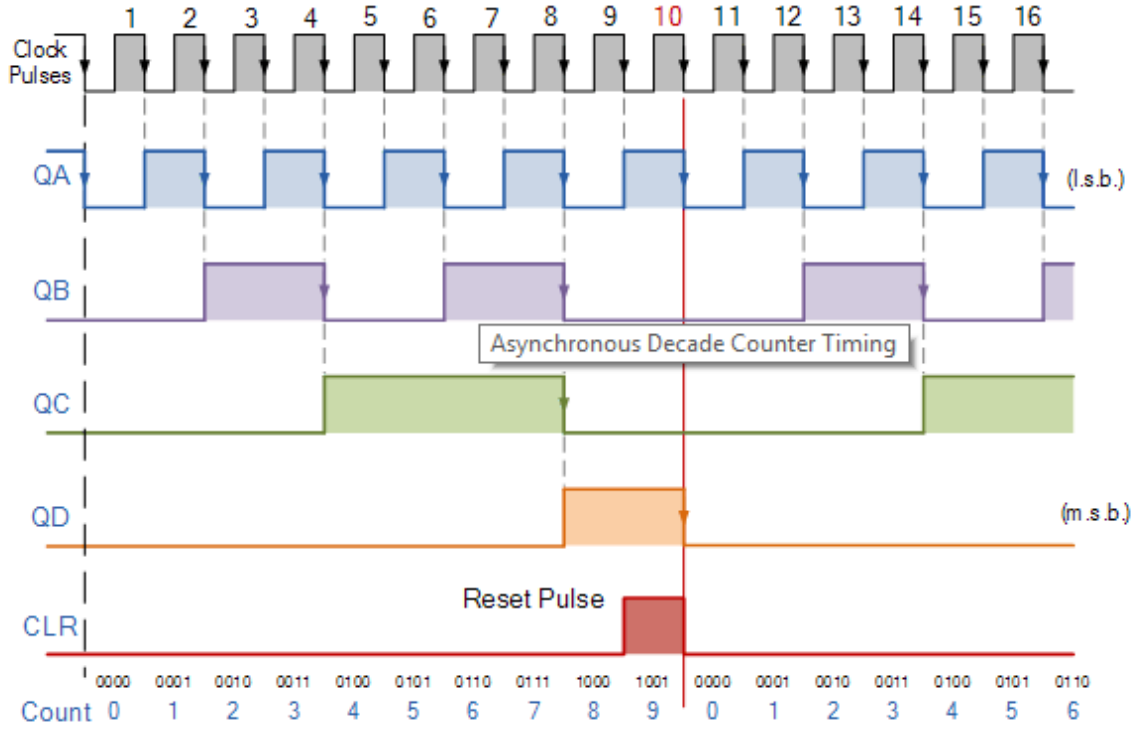
Truth Table :

Clock Count	Output bit Pattern				Decimal Value
	QD	QC	QB	QA	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Outputs back to Zero				

Pin Diagram of IC7490



Decade Counter Timing Diagram



Experiment 10: Synchronous UP/DOWN counters.

Conduct an experiment to verify the operation of UP/DOWN counter using IC74192.

Procedure:-

1) Rig up the circuit as shown in the diagram.

2) Apply the inputs to these counters as per the Truth table and observe the o/p verify with the truth table.

Pin Details of IC-74192

Truth Table :

COUNT-UP Mode				*COUNT-DOWN Mode*			
States	Q_C	Q_B	Q_A	*States*	Q_C	Q_B	Q_A
0	0	0	0	7	1	1	1
1	0	0	1	6	1	1	0
2	0	1	0	5	1	0	1
3	0	1	1	4	1	0	0
4	1	0	0	3	0	1	1
5	1	0	1	2	0	1	0
6	1	1	0	1	0	0	1
7	1	1	1	0	0	0	0

EXPERIMENT-11:

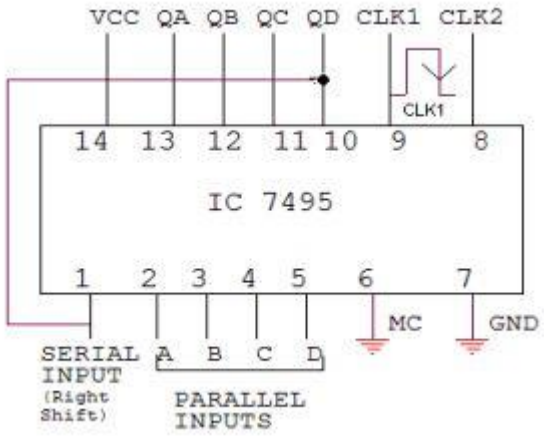
Ring Counter

Aim: Conduct an experiment to design and verify ring counter.

COMPONENTS REQUIRED: 7495

THEORY: Ring counter is a basic register with direct feedback such that the contents of the register simply circulate around the register when the clock is running. Here the last output that is QD in a shift register is connected back to the serial input.

Circuit diagram



Truth table

CLK	QA	QB	QC	QD
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1
8	1	0	0	0

Result : The truth table & working of Ring is verified

Exercise:

1) Conduct an experiment to design and verify Johnson counter.
2) Design ring counter and Johnson counter using D flip flop.
3) Design and set up four bit Johnson and ring counter using JK FF
4) Design a ring counter/Johnson counter with mode control using JK flip flop

EXPERIMENT-12

Conduct an experiment to verify 7- segment display decoders.

BCD TO 7-SEGMENT DECODER/DRIVER

LEARNING OBJECTIVE:

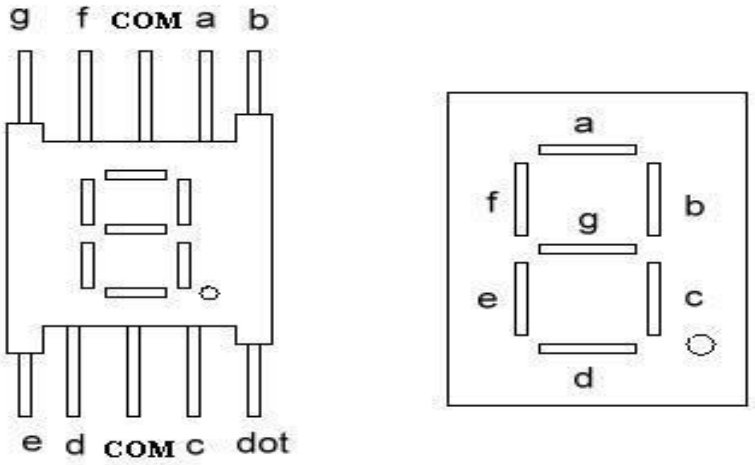
- To learn about various applications of decoder
- To learn and understand the working of IC 7447
- To learn about types of seven-segment display

COMPONENTS REQUIRED: IC7447, 7-Segment display (common anode), Patch chords

THEORY: The Light Emitting Diode (LED) finds its place in many applications in these modern electronic fields. One of them is the Seven Segment Display. Seven-segment displays contains the arrangement of the LEDs in “Eight” (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). The purpose of arranging it in that passion is that we can make any number out of that by switching ON and OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement. The Light Emitting Diode (LED), finds its place in many applications in this modern electronic fields. One of them is the Seven Segment Display. Seven-segment displays contains the arrangement of the LEDs in “Eight” (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). The purpose of arranging it in that passion is that we can make any number out of that by switching ON and OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement.

LED‟s are basically of two typesCommon Cathode (CC) -All the 8 anode legs uses only one cathode, which is common. Common Anode (CA)-The common leg for all the cathode is of Anode type. A decoder is a combinational circuit that connects the binary information from „n‟ input lines to a maximum of 2n unique output lines. The IC7447 is a BCD to 7-segment pattern converter. The IC7447 takes the Binary Coded Decimal (BCD) as the input and outputs the relevant 7 segment code.

Circuit diagram

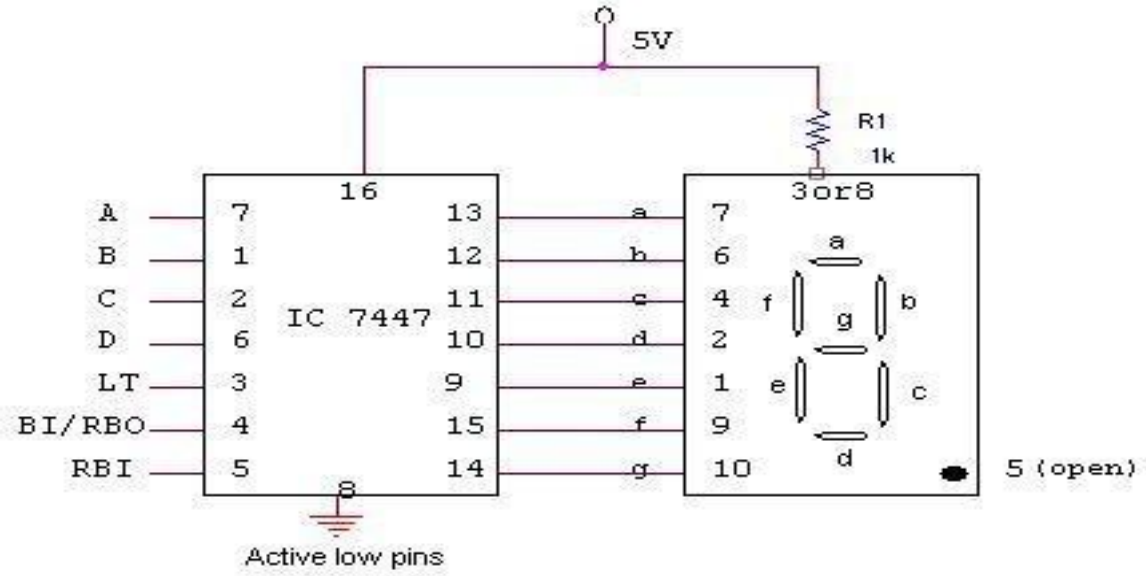


Seven-Segment Display

LED‟s are basically of two types-
Common Cathode (CC) -All the 8 anode legs uses only one cathode, which is common. Common Anode (CA)-The common leg for all the cathode is of Anode type.

A decoder is a combinational circuit that connects the binary information from „n‟ input lines to a maximum of 2^n unique output lines. The IC7447 is a BCD to 7-segment pattern converter. The IC7447 takes the Binary Coded Decimal (BCD) as the input and outputs the relevant 7 segment code.

CIRCUIT DIAGRAM:



TRUTH TABLE:

BCD Inputs				**Output Logic Levels from IC 7447 to 7-segments**							**Decimal number display**
D	C	B	A	A	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Verify the Truth Table and observe the outputs.