UGC & Govt. Approved
Sonargaon University (SU)
সোনারগাঁও ইউনিভার্সিটি (এসইউ)
WE WILL RISE UP WE WILL SHINE

Department of Electrical and Electronic Engineering

DIGITAL ELECTRONICS LAB MANUAL

COURSE CODE:EEE-307

INDEX

Experiment no: 01

Name Of the Experiment: Familiarization and use of truth table for basic for logic gates.

OBJECTIVE:

- Identify various ICs and their specification.

THEORY:

The basic logic gates are the building blocks of more complex logic circuits. These logic gates perform the basic Boolean functions, such as AND, OR, NAND, NOR, Inversion, Exclusive-OR, Exclusive-NOR. Fig. below shows the circuit symbol, Boolean function, and truth. It is seen from the Fig that each gate has one or two binary inputs, A and B, and one binary output, C. The small circle on the output of the circuit symbols designates the logic complement. The AND, OR, NAND, and NOR gates can be extended to have more than two inputs. A gate can be extended to have multiple inputs if the binary operation it represents is commutative and associative.

These basic logic gates are implemented as small-scale integrated circuits (SSICs) or as part of more complex medium scale (MSI) or very large-scale (VLSI) integrated circuits. Digital IC gates are classified not only by their logic operation, but also the specific logic-circuit family to which they belong. Each logic family has its own basic electronic circuit upon which more complex digital circuits and functions are developed. The following logic families are the most frequently used.

TTL	Transistor-transistor logic
ECL	Emitter-coupled logic
MOS	Metal-oxide semiconductor
CMOS	Complementary metal-oxide semiconductor

TTL and ECL are based upon bipolar transistors. TTL has a well-established popularity among logic families. ECL is used only in systems requiring high-speed operation. MOS and CMOS are based on field effect transistors. They are widely used in large scale integrated circuits because of their high component density and relatively low power consumption. CMOS logic consumes far less power than MOS logic. There are various commercial integrated circuit chips available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

COMPONENTS REQUIRED:

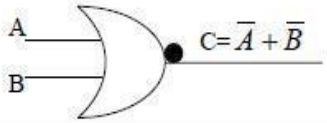
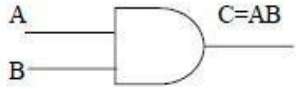
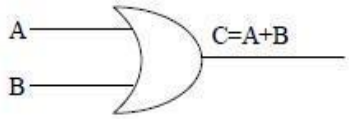
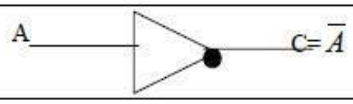
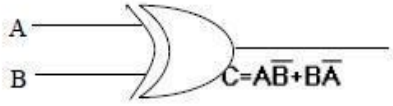
- Logic gates (IC) trainer kit.
- Connecting patch chords.
- IC 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486

PROCEDURE:

1. Turn the power (Trainer Kit) off before you build anything!
2. Make sure the power is off before you build anything!
3. Connect the +5V and ground (GND) leads of the power supply to the power and ground bus strips on your breadboard.
4. Plug the chips you will be using into the breadboard. Point all the chips in the same direction with pin 1 at the upper-left corner. (Pin 1 is often identified by a dot or a notch next to it on the chip package)
5. Connect +5V and GND pins of each chip to the power and ground bus strips on the breadboard.
6. Select a connection on your schematic and place a piece of hook-up wire between corresponding pins of the chips on your breadboard. It is better to make the short connections before the longer ones. Mark

each connection on your schematic as you go, so as not to try to make the same connection again at a later stage.

7. Get one of your group members to check the connections, **before you turn the power on**.
8. If an error is made and is not spotted before you turn the power on. Turn the power off immediately before you begin to rewire the circuit.
9. At the end of the laboratory session, collect you hook-up wires, chips and all equipment and return them to the demonstrator.
10. Tidy the area that you were working in and leave it in the same condition as it was before you started.

S.NO	GATE	SYMBOL	INPUTS		OUTPUT
			A	B	C
1.	NAND IC 7400		0	0	1
			0	1	1
			1	0	1
			1	1	0
2.	NOR IC 7402		0	0	1
			0	1	0
			1	0	0
			1	1	0
3.	AND IC 7408		0	0	0
			0	1	0
			1	0	0
			1	1	1
4.	OR IC 7432		0	0	0
			0	1	1
			1	0	1
			1	1	1
5.	NOT IC 7404		1	-	0
			0	-	1
6.	EX-OR IC 7486		0	0	0
			0	1	1
			1	0	1
			1	1	0

VIVA QUESTIONS:

1. Why NAND & NOR gates are called universal gates?
2. Realize the EX – OR gates using minimum number of NAND gates.
3. Give the truth table for EX-NOR and realize using NAND gates?
4. What are the logic low and High levels of TTL IC's and CMOS IC's?
5. Compare TTL logic family with CMOS family?
6. Which logic family is fastest and which has low power dissipation?

Experiment no: 02

Name Of the Experiment: Realization of a Boolean function.

OBJECTIVE:

- To simplify the Boolean expression and to build the logic circuit.
- Given a Truth table to derive the Boolean expressions and build the logic circuit to realize it.

THEORY:

Canonical Forms (Normal Forms): Any Boolean function can be written in disjunctive normal form (sum of min-terms) or conjunctive normal form (product of max-terms).

A Boolean function can be represented by a Karnaugh map in which each cell corresponds to a mitterm. The cells are arranged in such a way that any two immediately adjacent cells correspond to two minterms of distance 1. There is more than one way to construct a map with this property.

COMPONENTS REQUIRED:

- IC 7400
- IC 7408
- IC 7432
- IC 7406
- IC 7402
- Patch Cords & IC Trainer Kit.

Karnaugh Maps

For a function of two variables, say, f(x, y),

	x'	x
y'	f(0,0)	f(1,0)
y	f(0,1)	f(1,1)

For a function of three variables, say, f(x, y, z)

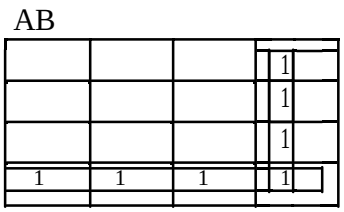
	x'y'	x'y	xy	xy'
z'	f(0,0,0)	f(0,1,0)	f(1,1,0)	f(1,0,0)
z	f(0,0,1)	f(0,1,1)	f(1,1,1)	f(1,0,1)

For a function of four variables: f(w, x, y, z)

	w'x'	w'x	wx	wx'
y'z'	0	4	12	8
y'z	1	5	13	9
yz	3	7	15	11
yz'	2	6	14	10

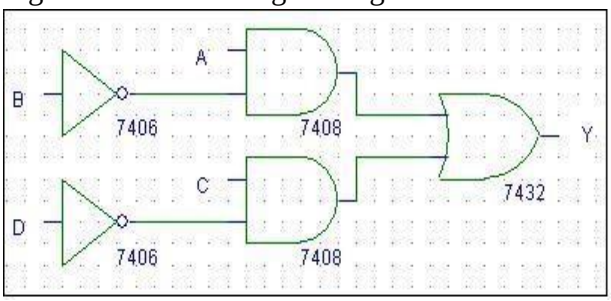
Realization of Boolean expression:

1) $Y = \bar{A}\,\bar{B}\,C\,\bar{D} + \bar{A}\,BC\,\bar{D} + ABC\,\bar{D} + A\,\bar{B}\,C\,\bar{D} + A\,\bar{B}\,\bar{C}\,\bar{D} + A\,\bar{B}\,\bar{C}\,D + A\,\bar{B}\,CD$

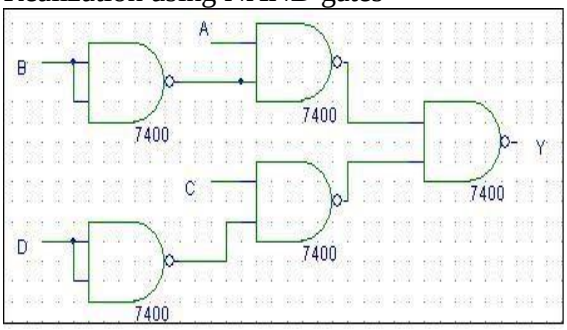


After simplifying using K-Map method we get $Y = A\,\bar{B} + C\,\bar{D}$

Realization using Basic gates



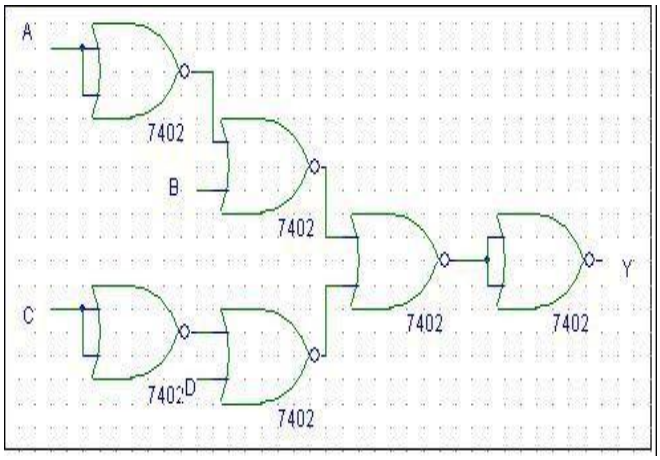
Realization using NAND gates



TRUTH TABLE

INPUTS				OUTPUT
A	**B**	**C**	**D**	**Y**
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Realization using NOR gates



2) For the given Truth Table, realize a logical circuit using basic gates and NAND gates

Inputs				Output
A	**B**	**C**	**D**	**Y**
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

PROCEDURE:

- Check the components for their working. Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Provide the input data via the input switches and observe the output on output LEDs Verify the Truth Table

RESULT: Simplified and verified the Boolean function using basic gates and universal gates

VIVA QUESTIONS:

1) What are the different methods to obtain minimal expression?
2) What is a Min term and Max term
3) State the difference between SOP and POS.
4) What is meant by canonical representation?
5) What is K-map? Why is it used?
6) What are universal gates?

Experiment no: 03

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

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.

Component:

- logic trainer kit
- NAND gates (IC 7400)
- Wires.

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

$Y = (A.A)'$

=> $Y = (A)'$

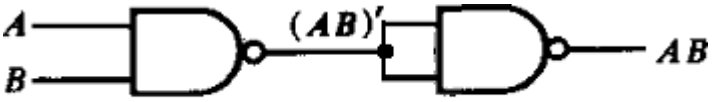
NOT (inverter)

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.

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

=> $Y = (A.B)$



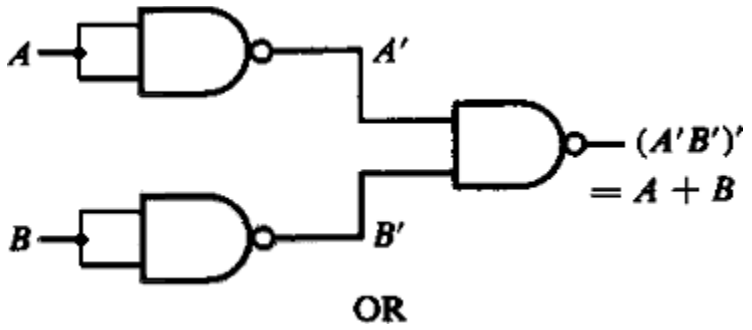
AND

NAND gates as OR gate

From DE Morgan's theorems: $(A.B)' = A' + B'$

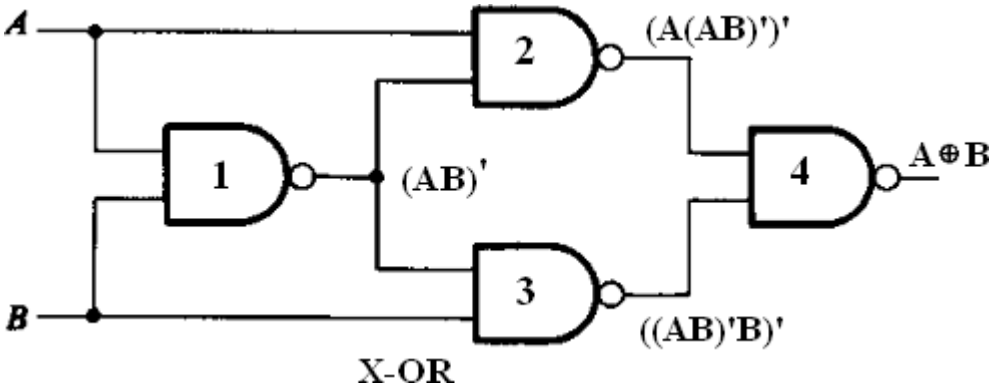
=> $(A'.B')' = A'' + B'' = A + B$

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



NAND gates as X-OR gate

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



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

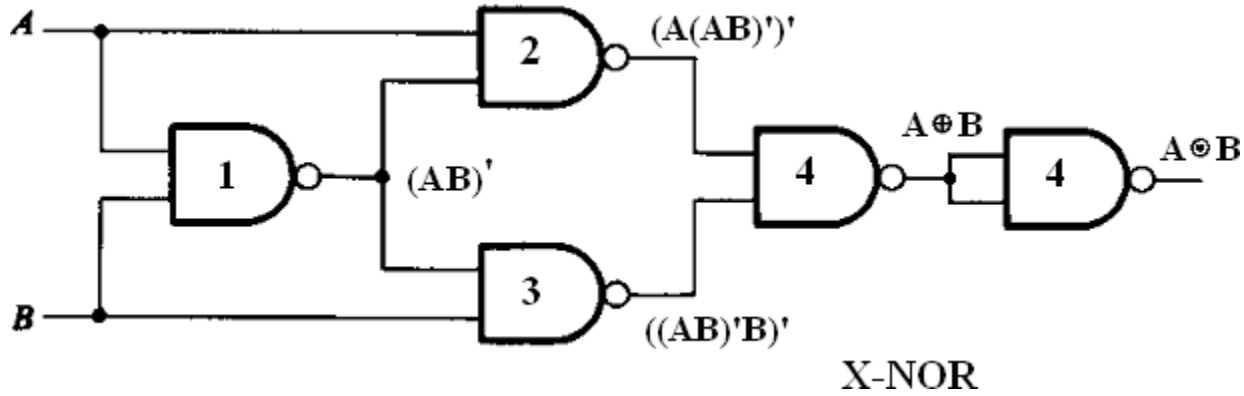
Y = ((A (AB)')' (B (AB)')')'
= (A(AB)'')'' + (B(AB)'')''
= (A(AB)') + (B(AB)')
= (A(A' + B)') + (B(A' + B'))
= (AA' + AB') + (BA' + BB')
= (0 + AB' + BA' + 0)
= AB' + BA'

=> 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 output is that of an X-NOR gate.

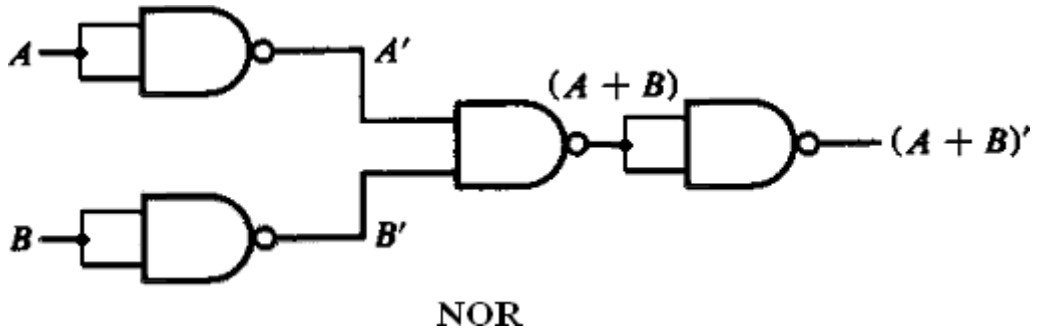
$$Y = AB + A'B'$$



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.

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



Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the NAND gates for any of the logic functions to be realized.
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 no: 04

Name of the Experiment: Design and verify the logic circuit of Half &Full adder and Full adder using logic gates.

Objective:

- To understand the principle of binary addition.
- To understand and to differentiate half & full adder concept.
- To implement half adder and full adder circuit using logic gates

Theory:

Half Adder: A half adder is a logical circuit that performs an addition operation on two binary digits. The half adder produces a sum and a carry value which are both binary digits.

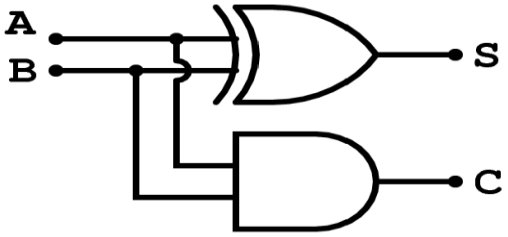


Fig 2.3: Circuit Diagram of Half Adder

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table

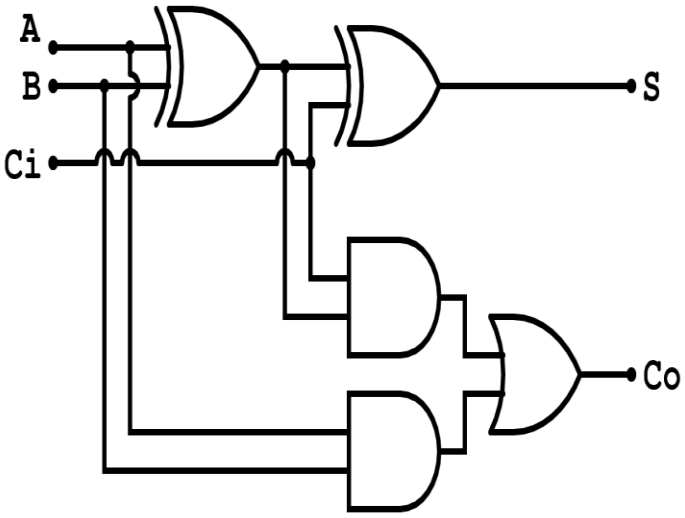
Boolean Expression: **S**= A⊕B

C=AB

Apparatus Required:

- Prototyping board (breadboard)
- DC Power Supply 5V Battery
- Light Emitting Diode (LED)
- Digital ICs: 7408 : Quad 2 input AND
  7486: Quad 2 input EXOR
  7432: Quad 2 input OR
- Connecting Wires

Full Adder: Full adder is a logical circuit that performs an addition operation on three binary digits. The full adder produces a sum and carries value, which are both binary digits. It can be combined with other full adders or work on its own.



Input			Output	
A	B	Ci	S	Co
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

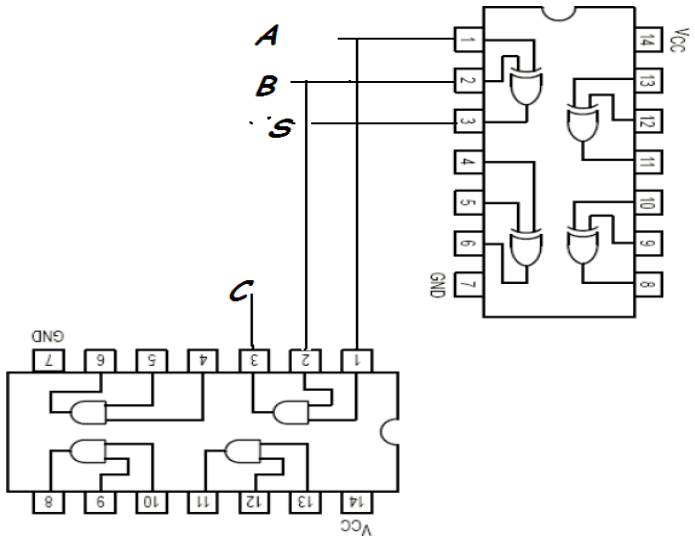
Fig 2.4: Circuit Diagram of Full Adder **Truth Table**

Boolean Expression: **S= A⊕B⊕Ci**

Co=AB+Ci(A⊕B)

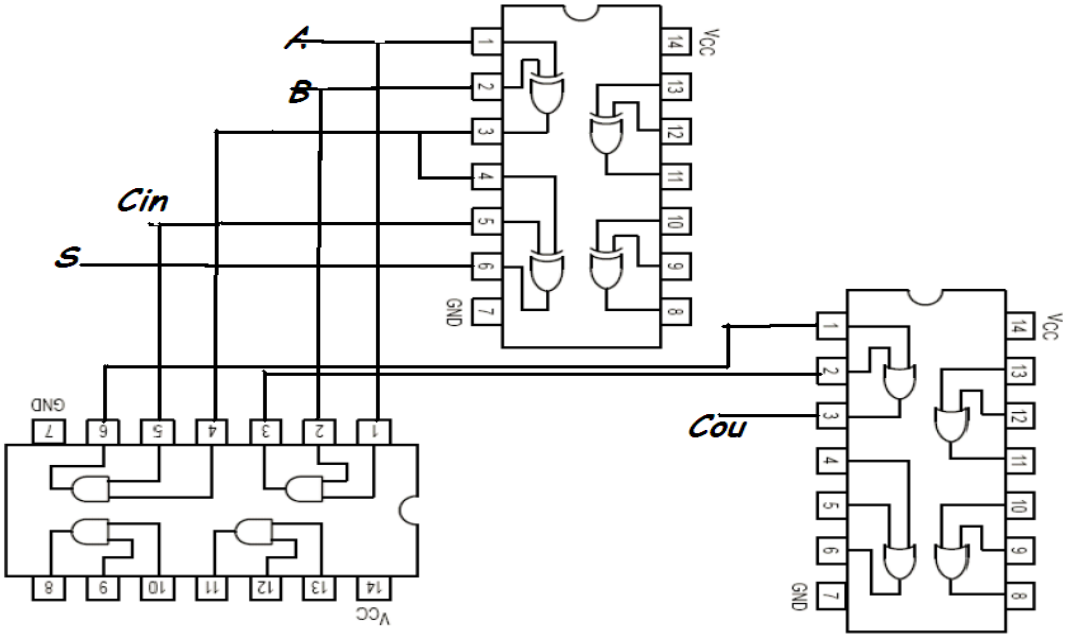
Pin Diagram:

Half Adder:



Pin Diagram of Half Adder

Full adder:



Pin diagram of Full adder

Procedure:

1. Collect the components necessary to accomplish this experiment.
2. Plug the IC chip into the breadboard.
3. Connect the supply voltage and ground lines to the chips. PIN7 = Ground and PIN14 = +5V.
4. According to the pin diagram of each IC mentioned above, make according to circuit diagram.
5. Connect the inputs of the gate to the input switches of the LED.
6. Connect the output of the gate to the output LEDs.
7. Once all connections have been done, turn on the power switch of the breadboard
8. . Operate the switches and fill in the truth table (Write "1" if LED is ON and"0" if LED is OFF apply the various combinations of inputs according to the truth table and observe the condition of Output LEDs.

Experiment no: 05

Name of Experiment: Design and implement Digital comparator.

Objective:-

To design and setup single bit comparator using logic gates and verify the truth table.

Theory

Digital or Binary Comparators compares the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

The digital comparator accomplishes this using several logic gates that operate on the principles of Boolean algebra. The purpose of a **Digital Comparator** is to compare a set of variables or unknown numbers, for example A (A1, A2, A3, ... An, etc) against that of a constant or unknown value such as B (B1, B2, B3, ... Bn, etc) and produce an output condition or flag depending upon the result of the comparison. There are two main types of Digital Comparator available and these are.

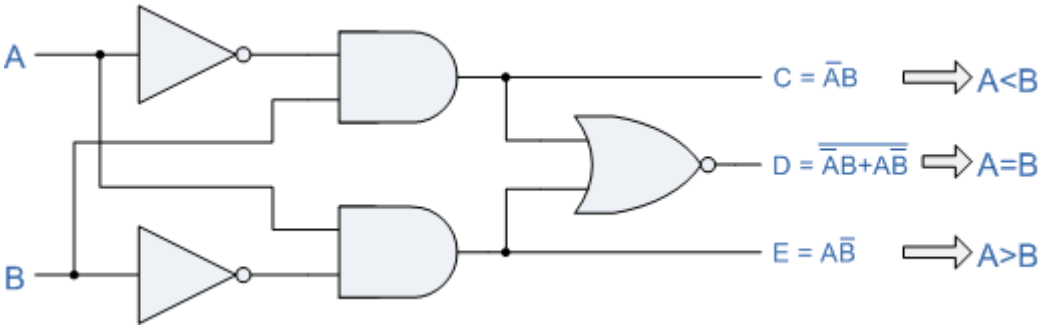
Identity Comparator - an *Identity Comparator* is a digital comparator that has only one output terminal for when the inputs A = B

Magnitude Comparator - a *Magnitude Comparator* is a type of digital comparator that has three output terminals, one each for equality (A = B), greater than (A > B) and less than
(A < B)
This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the truth table of simple 1-bit comparator.

Truth table

Input		Output		
B	**A**	**A>B**	**A=B**	**A<B**
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Digital comparators actually use Exclusive-NOR gates within their design for comparing their respective pairs of bits. When we are comparing two binary or BCD values or variables against each other, we are comparing the "magnitude" of these values, logic "0" against a logic "1" which is where the term Magnitude Comparator comes from.



RESULT

Designed and setup single bit comparator using logic gates and verified the truth table.

Experiment no: 06

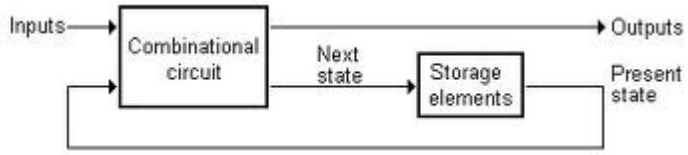
Name of Experiment: Study of Flip-Flops.

Hardware Requirement

- Digital IC Trainer Kit
- 74LS73 JK-Flip flop
- 74LS279 SR Flip flop

Theory

Digital electronic circuit is classified into combinational logic and sequential logic. Combinational logic output depends on the inputs levels, whereas sequential logic output depends on stored levels and also the input levels.



The storage elements (Flip -flops) are devices capable of storing 1-bit binary info. The binary info stored in the memory elements at any given time defines the state of the Sequential circuit. The input and the present state of the memory element determine the output. Storage elements next state is also a function of external inputs and present state.

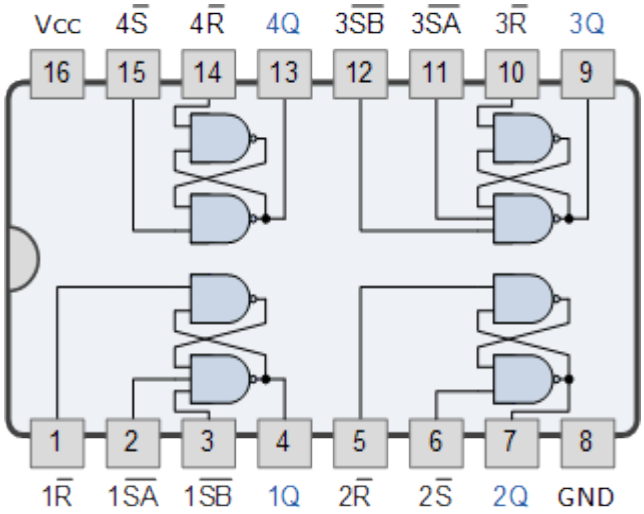
Flip-Flops and their properties

Flip-flops are synchronous bi-stable devices. The term synchronous means the output changes state only when the clock input is triggered. That is, changes in the output occur in synchronization with the clock. A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the stored bit. Since memory elements in sequential circuits are usually flip-flops, it is worth summarizing the behavior of various flip-flop types before proceeding further. All flip -flops can be divided into four basic types: **SR**, **JK**, **D** and**T**. They differ in the number of inputs and in the response invoked by different value of input signals.

1. RS FLIP-FLOP:

RS flip-flop is the simplest possible memory element. It can be constructed from two NAND gates or two NOR gates. Let us understand the operation of the RS flip-flop using NOR Gates as shown below using the truth table for 'A NOR B' gate. The inputs R and S are referred to as the Reset and Set inputs, respectively. The outputs Q and Q' are complements of each other and are referred to as the normal and complement outputs, respectively. The binary state of the flip-flop is taken to be the value of the normal output. When Q=1 and Q'=0, it is in the *set state* (or 1-state). When Q=0 and Q'=1, it is in the *reset/clear state* (or 0-state).

Quad SR Bi-stable Latch 74LS279



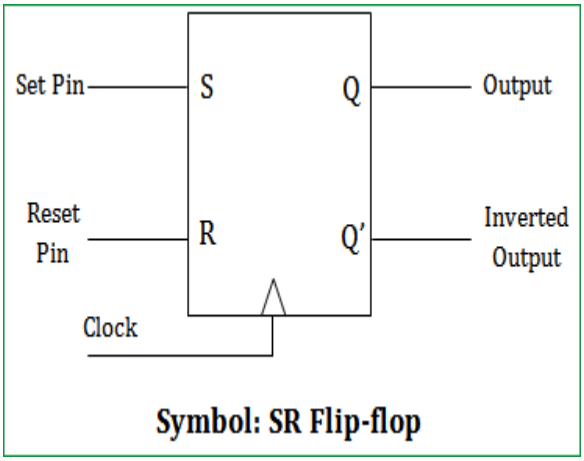


Fig.3. Circuit Diagram of RS Flip-Flop

- **S=1 and R=0:** The output of the bottom NOR gate is equal to zero, Q'=0. Hence both inputs to the top NOR gate are equal to 0, thus, Q=1. Hence, the input combination S=1 and R=0 leads to the flip-flop being **set** to Q=1.
- **S=0 and R=1:** Similar to the arguments above, the outputs become Q=0 and Q'=1. We say that the flip-flop is **reset**.
- **S=0 and R=0:** Assume the flip-flop was previously in set (S=1 and R=0) condition. Now changing S to 0 results Q' still at 0 and Q=1. Similarly, when the flip-flop was previously in a reset state (S=0 and R=1), the outputs do not change. Therefore, with inputs S=0 and R=0, the flip-flop holds its state.
- **S=1 and R=1:** This condition violates the fact that both outputs are complements of each other since each of them tries to go to 0, which is not a stable configuration. It is impossible to predict which output will go to 1 and which will stay at 0. In normal operation this condition must be avoided by making sure that 1's are not applied to both inputs simultaneously, thus making it one of the main disadvantages of RS flip-flop.

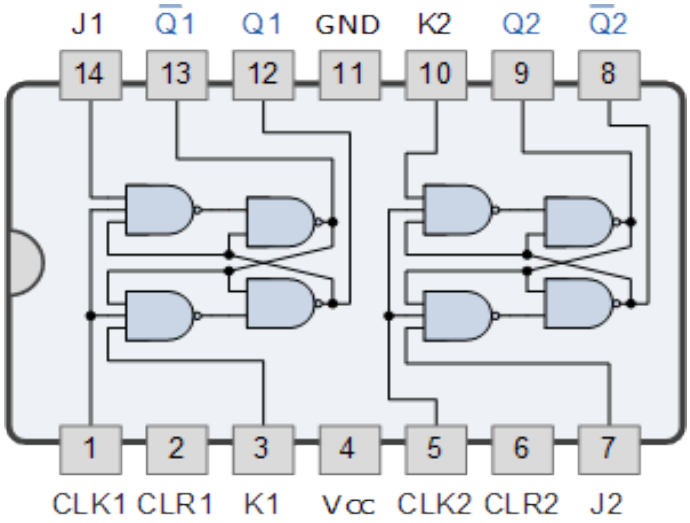
All the above conditions are summarized in the characteristic table below:

Characteristic Table:

R	S	Q	Q'	Comment
0	0	Q	Q'	Hold state
0	1	1	0	Set
1	0	0	1	Reset
1	1	?	?	Indeterminate

2. JK FLIP-FLOP:

The JK flip flop (JK means Jack Kilby, a Texas instrument engineer, who invented it) is the most versatile flip-flop, and the most commonly used flip flop. Like the RS flip-flop, it has two data inputs, J and K, and an EN/clock pulse input (CP). Note that in the following circuit diagram NAND gates are used instead of NOR gates. It has no undefined states, however. The fundamental difference of this device is the feedback paths to the AND gates of the input, i.e. Q is AND-ed with K and CP and Q' with J and CP.



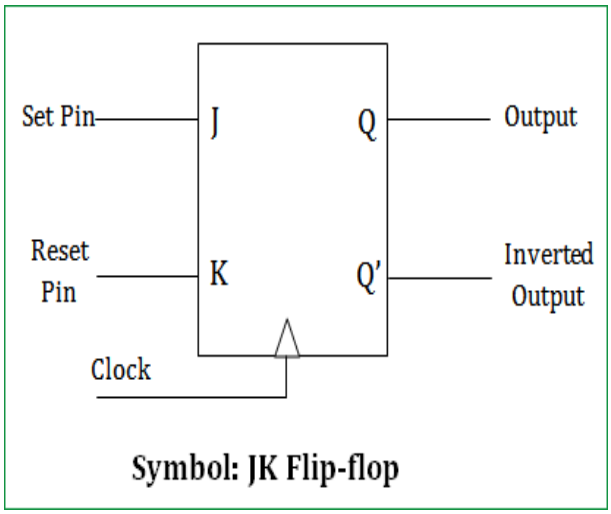


Fig. Circuit Diagram of JK Flip-Flop

Characteristic Table:

INPUTS					OUTPUTS	
$\overline{PRE}$	$\overline{CLR}$	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H†	H†
H	H	⎍	L	L	Q_0	$\overline{Q}_0$
H	H	⎍	H	L	H	L
H	H	⎍	L	H	L	H
H	H	⎍	H	H	TOGGLE	

The JK flip-flop has the following characteristics:

- If one input (J or K) is at logic 0, and the other is at logic 1, then the output is set or reset (by J and K respectively), just like the RS flip-flop.
- If both inputs are 0, then it remains in the same state as it was before the clock pulse occurred; again like the RS flip flop. CP has no effect on the output.
- If both inputs are high, however the flip-flop changes state whenever a clock pulse occurs; i.e.,the clock pulse toggles the flip-flop again and again until the CP goes back to 0 as shown in the shaded rows of the characteristic table above. Since this condition is undesirable, it should be eliminated by an improvised form of this flip-flop as discussed in the next section.

3. MASTER-SLAVE JK FLIP-FLOP:

Although JK flip-flop is an improvement on the clocked SR flip-flop it still suffers from timing problems called "race" if the output Q changes state before the timing pulse of the clock input has time to go "OFF", so the timing pulse period (T) must be kept as short as possible (high frequency). As this is sometimes not possible with modern TTL IC's the much improved Master-Slave J-K Flip-Flop was developed. This eliminates all the timing problems by using two SR flip-flops connected together in series, one for the "Master" circuit, which triggers on the leading edge of the clock pulse and the other, the "Slave" circuit, which triggers on the falling edge of the clock pulse.

The master-slave JK flip flop consists of two flip flops arranged so that when the clock pulse enables the first, or master, it disables the second, or slave. When the clock changes state again (i.e., on its falling edge) the output of the master latch is transferred to the slave latch. Again, toggling is accomplished by the connection of the output with the input AND gates.

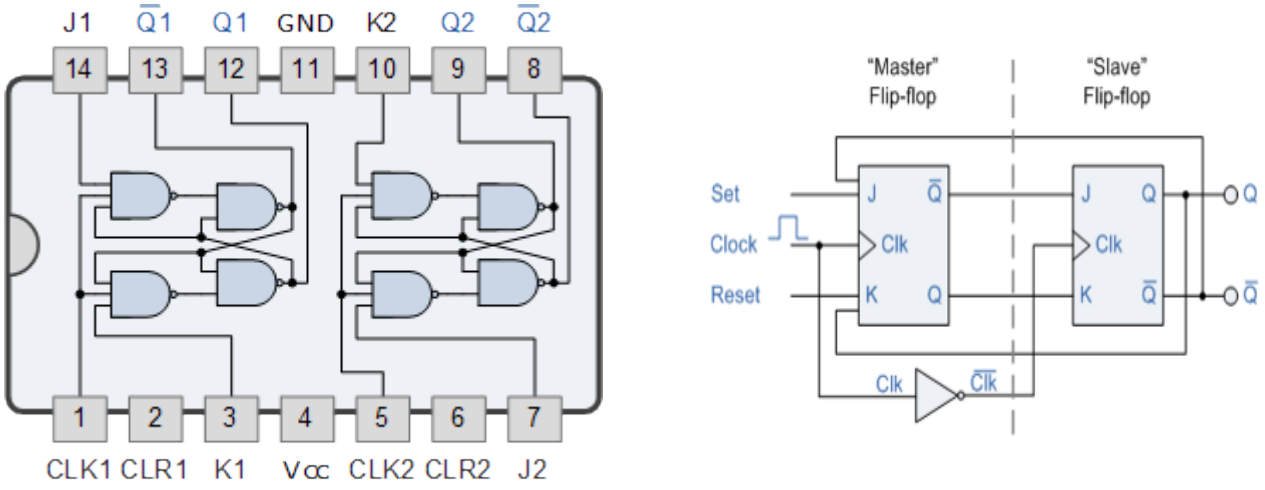


Fig.5. Circuit Diagram of Master-Slave JK Flip-Flop

Characteristic Table

CP	J	K	Q_m	$\overline{Q}_m$	Q_n	$\overline{Q}_n$
0⟶1	0	0	Hold		Hold	
1⟶0	0	0	Hold		Hold	
0⟶1	0	1	0	1	Hold	
1⟶0	0	1	Hold		0	1
0⟶1	1	0	1	0	Hold	
1⟶0	1	0	Hold		1	0
0⟶1	1	1	Toggle		Hold	
1⟶0	1	1	Hold		Toggle	

H HIGH Logic Level
X Either LOW or HIGH Logic Level
L LOW Logic Level
↑ Positive-going transition of the clock.
Q0 The output logic level of Q before the indicated input conditions was established.
IC7473-JK- Flip flop Connection Diagram with Function Table

Experiment no: 07

Name of Experiment: Design and implement encoder circuit using logic GATE and encoder IC 74148

Objective:

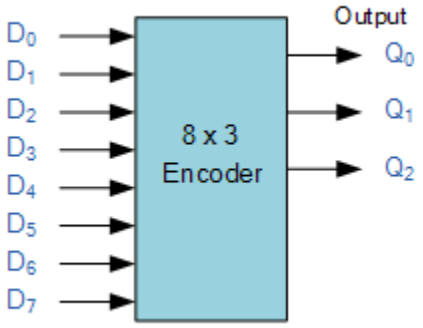
- To learn about various applications of Encoders
- To learn and understand the working of IC 74148
- To learn to do code conversion using encoders

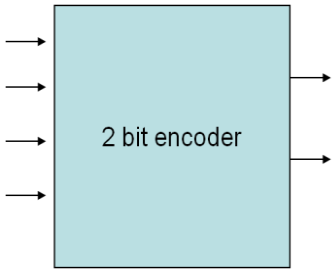
Theory:-

An encoder performs a function that is the opposite of decoder. It receives one or more signals in an encoded format and output a code that can be processed by another logic circuit. One of the advantages of encoding data, or more often data addresses in computers, is that it reduces the number of required bits to represent data or addresses. For example, if a memory has 16 different locations, in order to access these 16 different locations, 16 lines (bits) are required if the addressing signals are in 1 out of *n* format. However, if we code the 16 different addresses into a binary format, then only 4 lines (bits) are required. Such a reduction improves the speed of information processing in digital systems.

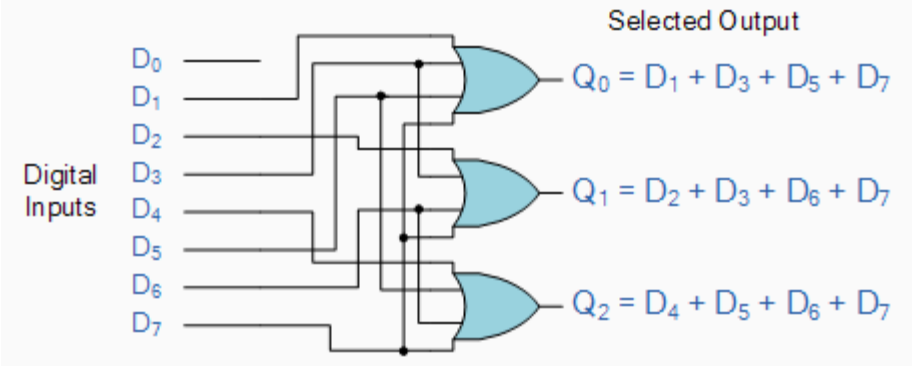
Generally encoders produce 2-bit, 3-bit or 4-bit code.

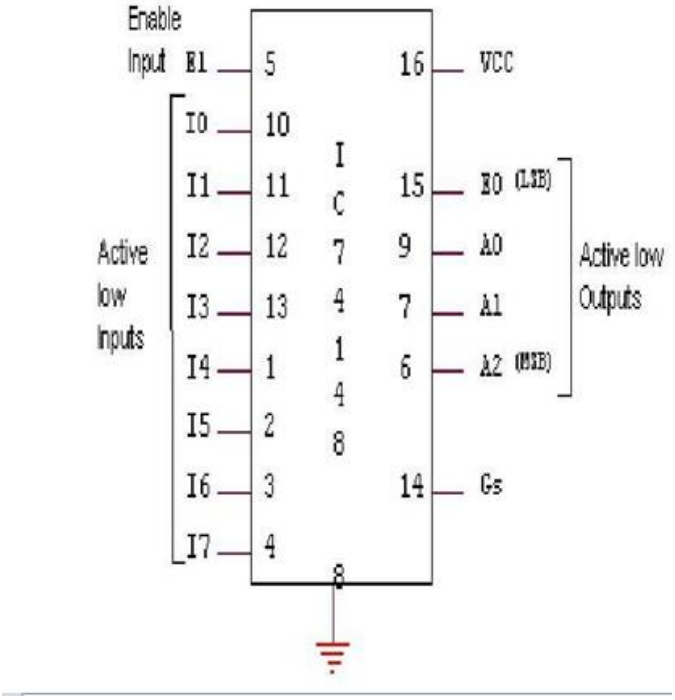
- n bit encoder has 2^n input





Inputs								Outputs		
D0	D1	D2	D3	D4	D5	D6	D7	Q2	Q1	Q0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1





ENCODER TABLE

Inputs									Outputs				
EI	7	6	5	4	3	2	1	0	A3	A2	A1	GS	EO
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	X	X	X	X	X	X	X	0	0	0	0	1
0	1	0	X	X	X	X	X	X	0	0	1	0	1
0	1	1	0	X	X	X	X	X	0	1	0	0	1
0	1	1	1	0	X	X	X	X	0	1	1	0	1
0	1	1	1	1	0	X	X	X	1	0	0	0	1
0	1	1	1	1	1	0	X	X	1	0	1	0	1
0	1	1	1	1	1	1	0	X	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

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.

VIVA QUESTIONS:

1. What is a priority encoder?
2. What is the role of an encoder in communication?
3. What is the advantage of using an encoder?
4. What are the uses of validating outputs?

Experiment no: 08

Name of Experiment: Design and implement Decoder using logic GATE and Decoder IC 74138.

Objectives:

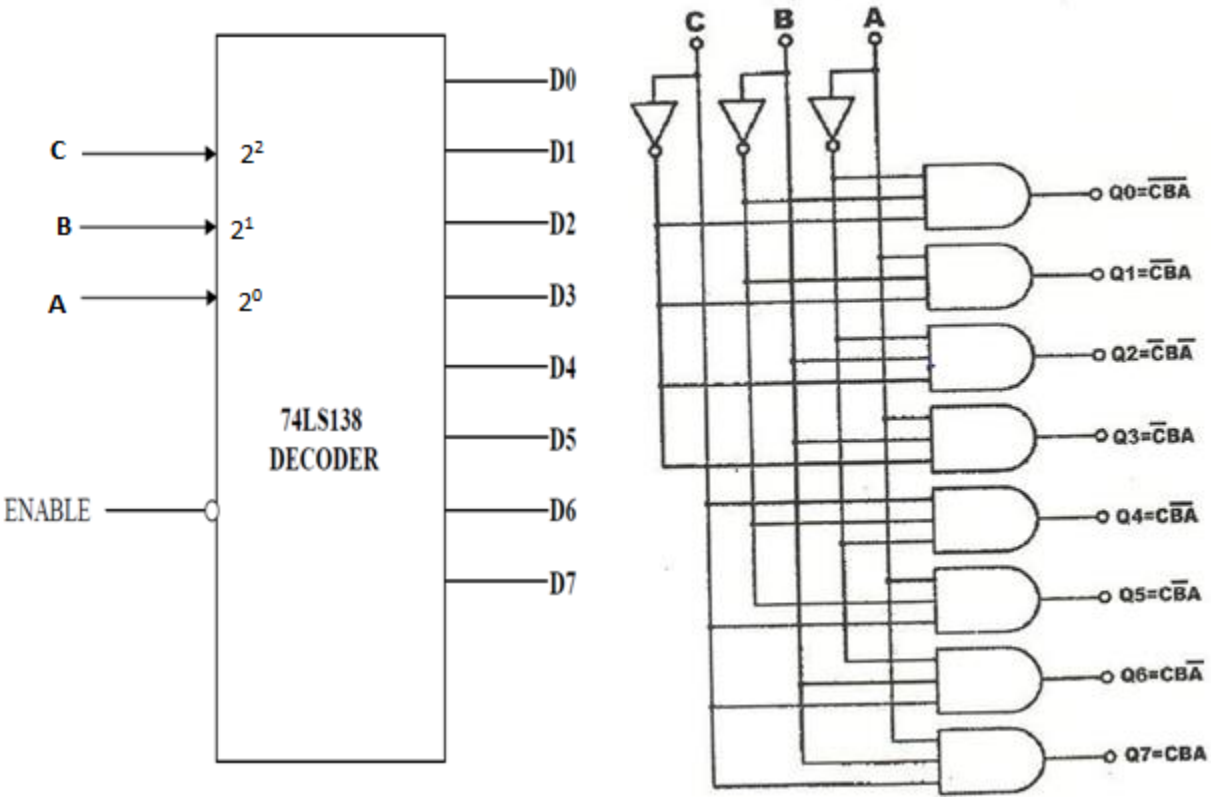
1. Understanding the construction and operational principles of digital decoders and encoders.

Theory:

A decoder is a logic circuit that will detect the presence of a specific binary number or word. The input to the decoder is a parallel binary number and the output is a binary signal that indicates the presence or absence of that specific number.

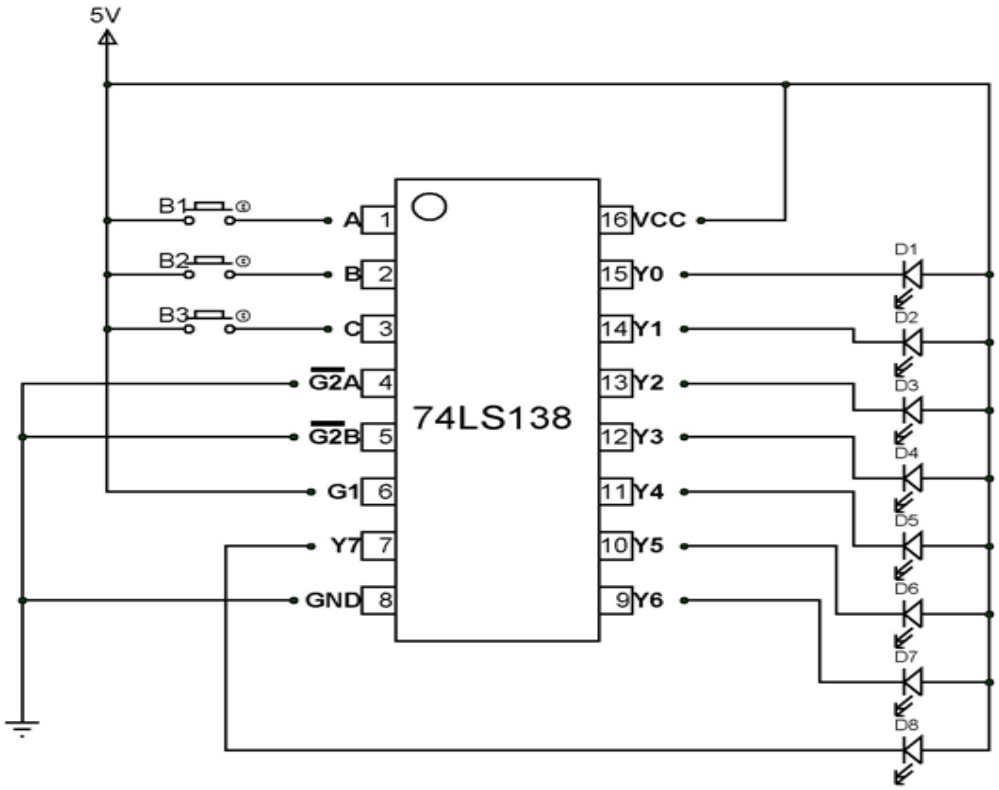
It is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

3-to-8 decoder



How to use 74LS138 Decoder

For understanding the working of device let us construct a simple application circuit with a few external components as shown below.



About circuit: Here the outputs are connected to LED to show which output pin goes LOW and do remember the outputs of the device are inverted. We are using a single device so we will connect G2A and G2B pin to ground followed by connecting G1 to VCC to enable the chip. The three buttons here represent three input lines for the device.
For understanding the working let us consider the truth table of the device.

Input						Output							
E1	E2	E3	A0	A1	A2	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

PROCEDURE:

- Make the connections as per the circuit diagram.
- Change the values of G1, G2A, G2B, A, B, and C, using switches.
- Observe status of Y0, to Y7 on LED's.
- Verify the truth table.

RESULT: Verified the Operation of 3 to 8 Decoder VIVA QUESTIONS:

- What are the applications of decoder?
- What is the difference between decoder & encoder?
- For n- 2^n decoder how many i/p lines & how many o/p lines?
- What are the different codes & their applications?
- What are code converters?

Experiment no: 09

Name of Experiment: BCD to 7-segment decoder/driver using IC 7447.

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

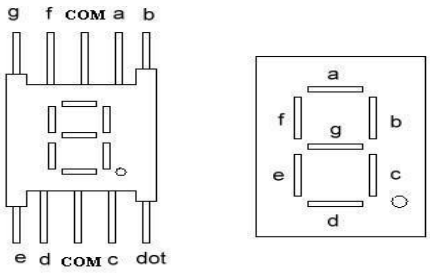
Components required:

- IC7447
- 7-Segment display (common anode)
- Patch chords
- Resistor (1K) &
- IC Trainer Kit

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.



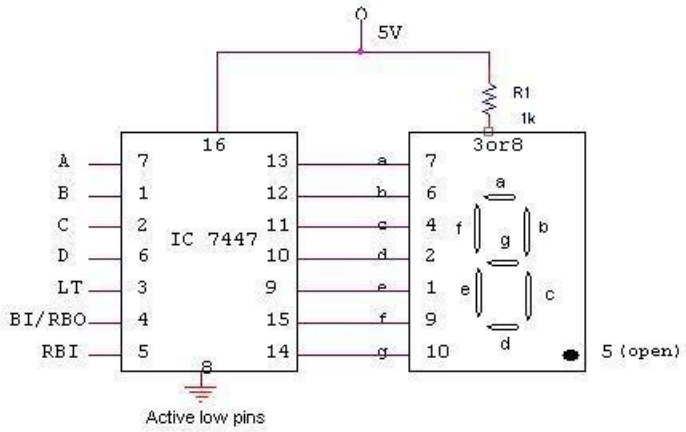
Seven-Segment Display

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.

VIVA QUESTIONS:

1. What are the different types of LEDs?
2. Draw the internal circuit diagram of an LED.
3. What are the applications of LEDs?