SIR GURUDAS MAHAVIDYALAYA

Department of Computer Sc.

LAB MANUAL

Computer Organization and Architecture

(CMSA-CC5)

EXPERIMENT: 1 LOGIC GATES

AIM: To study and verify the truth table of logic gates

LEARNING OBJECTIVE:

Identify various ICs and their specification. ·

COMPONENTS REQUIRED:

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

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

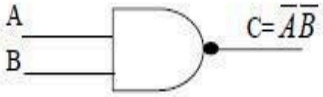
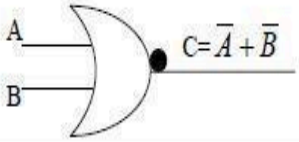
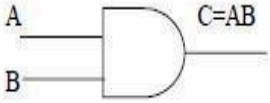
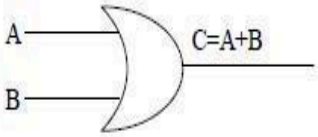
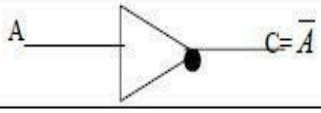
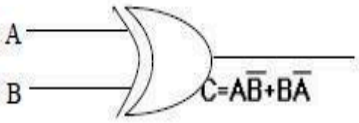
TTL and ECL are based upon bipolar transistors. TTL has a well established popularity among logic families. ECL is used only in systems requiring high-speed operation. MOS and CMOS, are based on field effect transistors. They are widely used in large scale integrated circuits because of their high component density and relatively low power consumption. CMOS logic consumes far less power than MOS logic. There are various commercial

Logi

integrated circuit chips available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

PROCEDURE:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Provide the input data via the input switches and observe the output on output LEDs

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

Logi

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: 2 REALIZATION OF A BOOLEAN FUNCTION.

AIM: To simplify the given expression and to realize it using Basic gates and Universal gates

LEARNING 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.

COMPONENTS REQUIRED:

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

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 minterm. 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.

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)

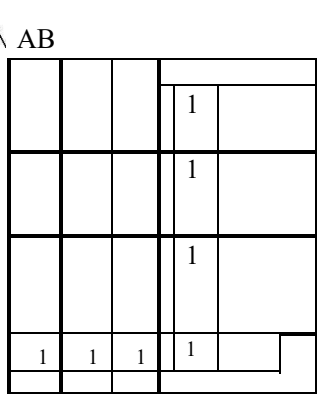
	x'y'	x'y	xy	xy'
z'	f(0,0,0)	f(0,1,0)	f(1,1,0)	f(1,0,0)
z	f(0,0,1)	f(0,1,1)	f(1,1,1)	f(1,0,1)

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

	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

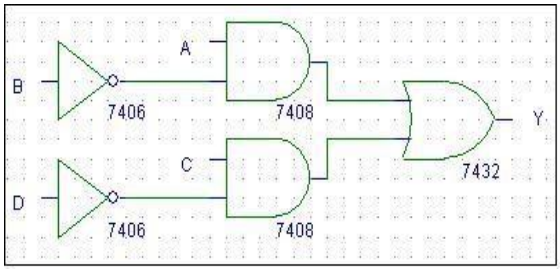
Realization of Boolean expression:

1) Y= *A B C D· A BC D· ABC D· A B C D· A B C D· A B C D · A B CD*



After simplifying using K-Map method we Y =A *B* + C *D*

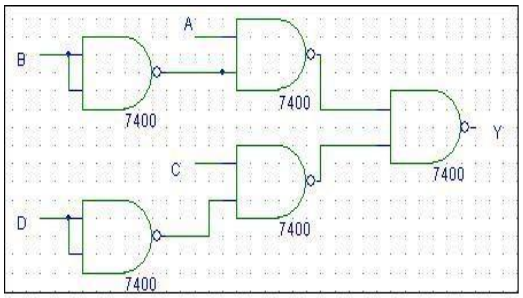
get Realization using Basic gates



TRUTH TABLE

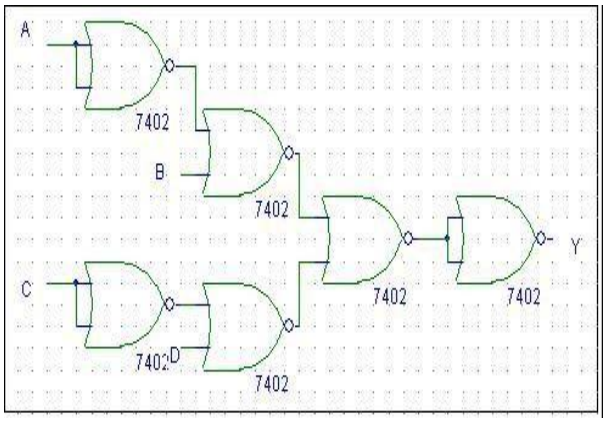
INPUTS				OUTPUT
A	**B**	**C**	**D**	**Y**
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Realization using NAND gates



Realization using NOR gates

Logi



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: 3 ADDERS AND SUBTRACTORS

AIM: To realize

i) Half Adder and Full Adder

ii) Half Subtractor and Full Subtractor by using Basic gates and NAND gates

LEARNING OBJECTIVE:

To realize the adder and subtractor circuits using basic gates and universal gates To realize full adder using two half adders

To realize a full subtractor using two half subtractors

COMPONENTS REQUIRED:

IC 7400, IC 7408, IC 7486, IC 7432, Patch Cords & IC Trainer Kit.

THEORY:

Half-Adder: A combinational logic circuit that performs the addition of two data bits, A and B, is called a half-adder. Addition will result in two output bits; one of which is the sum bit, S, and the other is the carry bit, C. The Boolean functions describing the half-adder are:

$S = A \oplus B$ $C = A\ B$

Full-Adder: The half-adder does not take the carry bit from its previous stage into account. This carry bit from its previous stage is called carry-in bit. A combinational logic circuit that adds two data bits, A and B, and a carry-in bit, Cin , is called a full-adder. The Boolean functions describing the full-adder are:

$S = (x \oplus y) \oplus Cin$ $C = xy + Cin\ (x \oplus y)$

Half Subtractor: Subtracting a single-bit binary value B from another A (i.e. A -B) produces a difference bit D and a borrow out bit B-out. This operation is called half subtraction and the circuit to realize it is called a half subtractor. The Boolean functions describing the halfSubtractor are:

$S = A \oplus B$ $C = A'\ B$

Full Subtractor: Subtracting two single-bit binary values, B, Cin from a single-bit value A produces a difference bit D and a borrow out Br bit. This is called full subtraction. The Boolean functions describing the full-subtracter are:

$D = (x \oplus y) \oplus Cin$ $Br = A'B + A'(Cin) + B(Cin)$

I. TO REALIZE HALF ADDER

TRUTH TABLE **BOOLEAN EXPRESSIONS:**

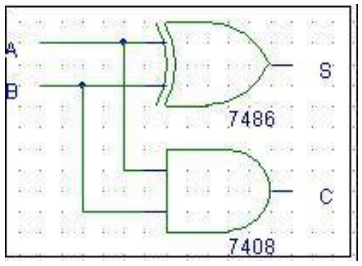
INPUTS		OUTPUTS	
A	B	S	C

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

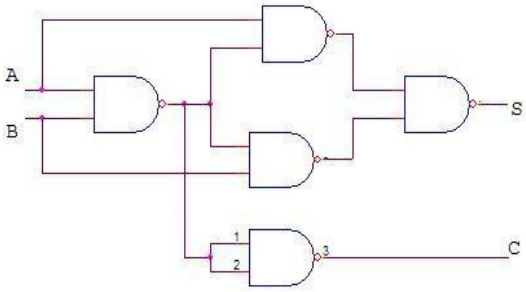
S=A ⊕ B

C=A B

i) Basic Gates



ii) NAND Gates



II. FULL ADDER

TRUTH TABLE

INPUTS			OUTPUTS	
A	**B**	**Cin**	**S**	**C**
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

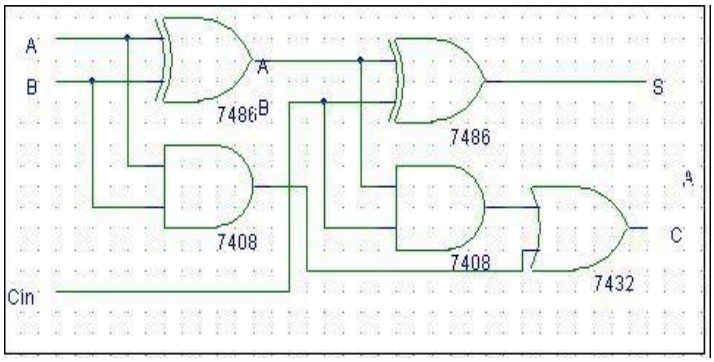
BOOLEAN EXPRESSIONS:

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

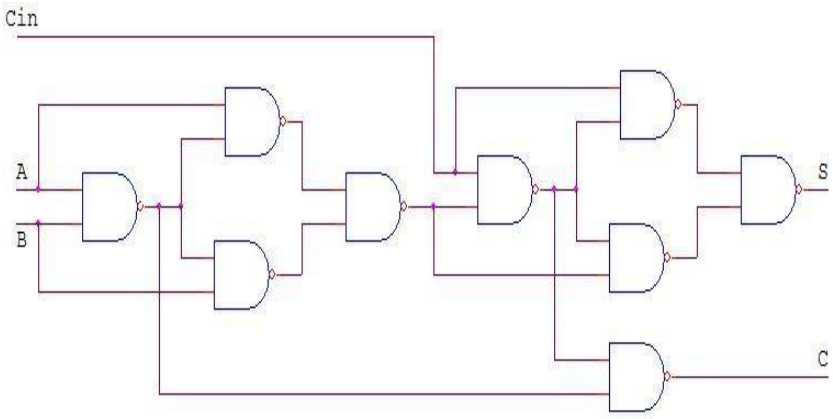
S= A ⊕ B ⊕ C

C=A B + B Cin + A Cin

i)BASIC GATES



ii) NAND GATES



III. HALF SUBTRACTOR

TRUTH TABLE **BOOLEAN EXPRESSIONS:**

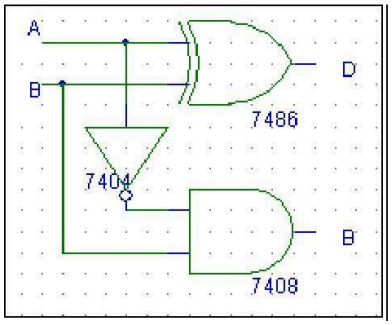
INPUTS		OUTPUTS	
A	B	D	Br
0	0	0	0
0	1	1	1

1	0	1	0
1	1	0	0

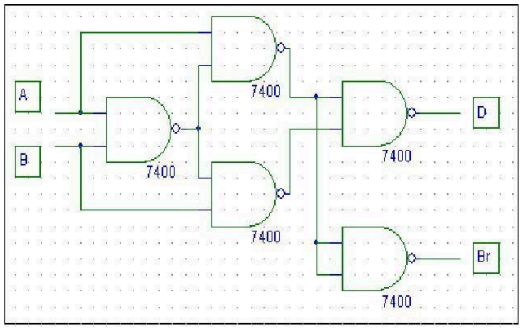
D = A ⊕ B

Br = $\overline{A}\,B$

i)BASIC GATES



ii) NAND Gates



IV. FULL SUBTRACTOR

TRUTH TABLE

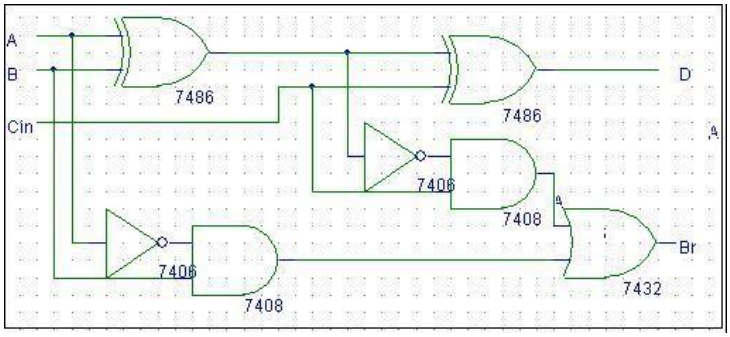
INPUTS			OUTPUTS	
A	B	Cin	D	Br
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1

BOOLEAN EXPRESSIONS:

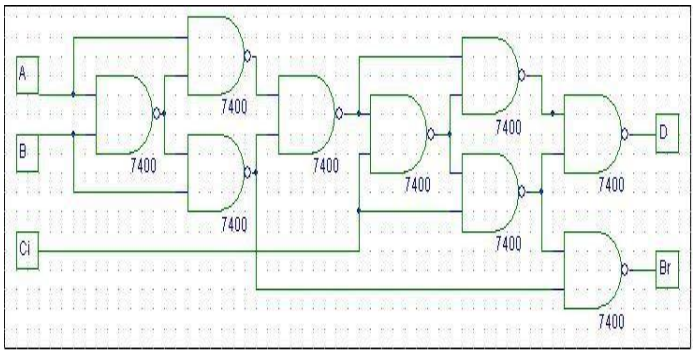
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

D= A ⊕ B ⊕ C **Br=** $\bar{A}$ **B + B Cin +** $\bar{A}$ **Cin**

i) BASIC GATES



ii) To Realize the Full subtractor using NAND Gates only



PROCEDURE:

· Check 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. RESULT:

The truth table of the above circuits is verified. VIVA

QUESTIONS: ·

·

1) What is a half adder?
2) What is a full adder?
3) What are the applications of adders?
4) What is a half subtractor?
5) What is a full subtractor?
6) What are the applications of subtractors?
7) Obtain the minimal expression for above circuits. 8) Realize a full adder using two half adders
9) Realize a full subtractors using two half subtractors

EXPERIMENT: 4 PARALLEL ADDER AND SUBTRACTOR

AIM: To design and set up the following circuit using IC 7483.

i) A 4-bit binary parallel adder. ii) A 4-bit binary parallel subtractor.

LEARNING OBJECTIVE:

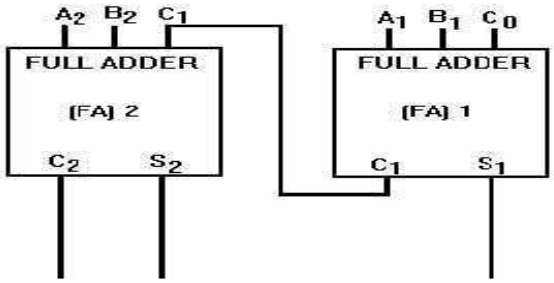
To learn about IC 7483 and its internal structure.
To realize a subtractor using adder IC 7483

COMPONENTS REQUIRED:

IC 7483, IC 7486, Patch Cords & IC Trainer Kit.

THEORY:

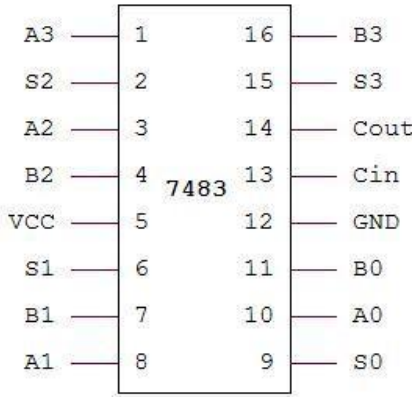
The Full adder can add single-digit binary numbers and carries. The largest sum that can be obtained using a full adder is 11_2. Parallel adders can add multiple-digit numbers. If full adders are placed in parallel, we can add two- or four-digit numbers or any other size desired. Figure below uses STANDARD SYMBOLS to show a parallel adder capable of adding two, two-digit binary numbers The addend would be on A inputs, and the augend on the B inputs. For this explanation we will assume there is no input to C_0 (carry from a previous circuit)



To add 10_2 (addend) and 01_2 (augend), the addend inputs will be 1 on A2 and 0 on A1. The augend inputs will be 0 on B2 and 1 on B1. Working from right to left, as we do in normal addition, let's calculate the outputs of each full adder. With A1 at 0 and B1 at 1, the output of adder1 will be a sum (S_1) of 1 with no carry (C_1). Since A2 is 1 and B2 is 0, we have a sum (S_2) of 1 with no carry (C_2) from adder1. To determine the sum, read the outputs (C2, S2, and

S1) from left to right. In this case, C2 = 0, S2 = 1, and S1 = 1. The sum, then, of 10_2 and 01_2 is 011_2. To add four bits we require four full adders arranged in parallel. IC 7483 is a 4- bit parallel adder whose pin diagram is shown.

	MSB				**LSB**
INPUTS					Cin
		A3	A2	A1	A0
		B_3	B_2	B_1	B_0
OUTPUT	Cout	S_3	S_2	S_1	S_0



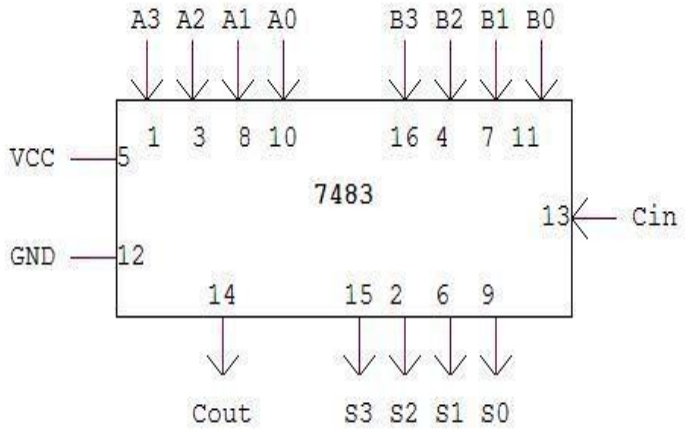
IC 7483 pin diagram

i) 4-Bit Binary Adder

An Example: 7+2=11 (1001)

7 is realized at A_3 A_2 A_1 A_0 = 0111 ·
2 is realized at B_3 B_2 B_1 B_0 = 0010
Sum = 1001

ADDER CIRCUIT:



PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply augend and addend bits on A and B and cin=0.

Verify the results and observe the outputs.

i) 4-BIT BINARY SUBTRACTOR.

Subtraction is carried out by adding 2's complement of the subtrahend.
Example: 8 – 3 = 5 (0101)

8 is realized at A_3 A_2 A_1 A_0 = 1000

3 is realized at B_3 B_2 B_1 B_0 through X-OR gates = 0011

Output of X-OR gate is 1's complement of 3 = 1100

2's Complement can be obtained by adding Cin =1

Therefore

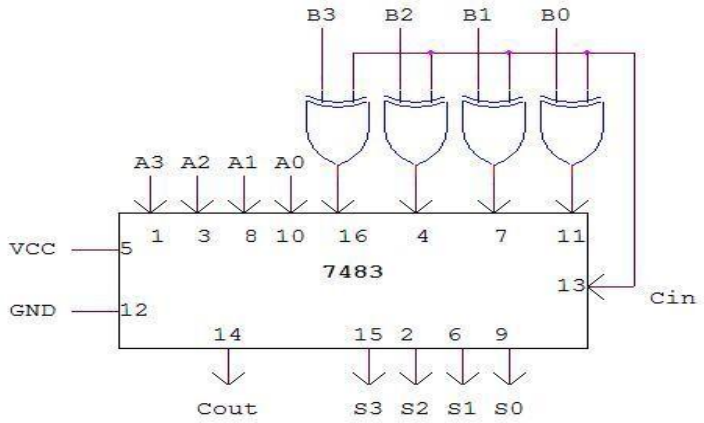
Cin = 1

A_3 A_2 A_1 A_0 = 1 0 0 0

B_3 B_2 B_1 B_0 = 1 1 0 0

S_3 S_2 S_1 S_0 = 0 1 0 1

Cout = 1 (Ignored)



PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply Minuend and subtrahend bits on A and B and cin=1.
  Verify the results and observe the outputs.

RESULTS: Verified the working of IC 7483 as adder and subtractor.

i) 4-BIT BINARY SUBTRACTOR.

Subtraction is carried out by adding 2's complement of the subtrahend.
Example: 8 – 3 = 5 (0101)

8 is realized at A_3 A_2 A_1 A_0	= 1000
3 is realized at B_3 B_2 B_1 B_0 through X-OR gates	= 0011
Output of X-OR gate is 1's complement of 3	= 1100
2's Complement can be obtained by adding Cin	=1

Therefore

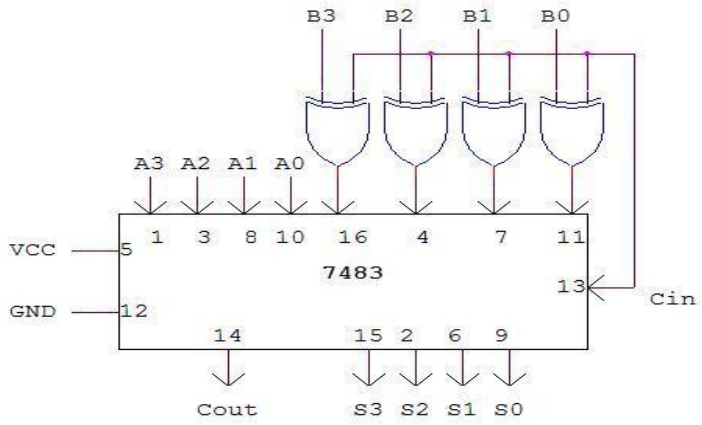
Cin = 1

$A_3 A_2 A_1 A_0 = 1\ 0\ 0\ 0$

$B_3 B_2 B_1 B_0 = 1\ 1\ 0\ 0$

$S_3\ S_2\ S_1\ S_0 = 0\ 1\ 0\ 1$

Cout = 1 (Ignored)



PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply Minuend and subtrahend bits on A and B and cin=1.
  Verify the results and observe the outputs.

RESULTS: Verified the working of IC 7483 as adder and subtractor.

BCD adder using two 7483 IC chip

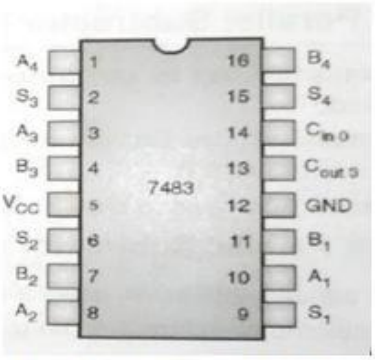


Fig3. Pin diagram of IC 7483

- A BCD adder adds two BCD digits and produces output as a BCD digit. A BCD or Binary Coded Decimal digit cannot be greater than 9.
- The two BCD digits are to be added using the rules of binary addition. If sum is less than or equal to 9 and carry is 0, then no correction is needed. The sum is correct and in true BCD form.
- But if sum is greater than 9 or carry =1, the result is wrong and correction must be done.
  The wrong result can be corrected adding six (0110) to it.
- For implementing a BCD adder using a binary adder circuit IC 7483, additional combinational circuit will be required, where the Sum output S_3—S_0
- is checked for invalid values from 10 to 15. The truth table and K-map for the same is as shown:

I/P				O/P
S3	S2	S1	S0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
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	1

Table1. Truth table for BCD numbers

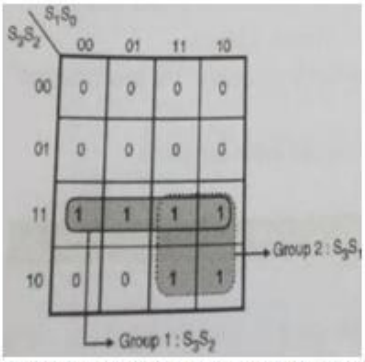


Fig. K-map for above truth table

- The Boolean expression is, $Y=S_3S_2+S_3S_1$ · · The BCD adder is shown below. The output of the combinational circuit should be 1 if Cout of adder-1 is high. Therefore Y is ORed with Cout of adder 1.

- The output of combinational circuit is connected to B1B2 inputs of adder-2 and

  $B_3=B_1+0$ as they are connected to ground permanently. This makes $B_3B_2B_1B_0$ · =

  0110 if Y' = 1.

- The sum outputs of adder-1 are applied to $A_3A_2A_1A_0$
- of adder-2. The output of combinational circuit is to be used as final output carry and the carry output of adder-2 is to be ignored.

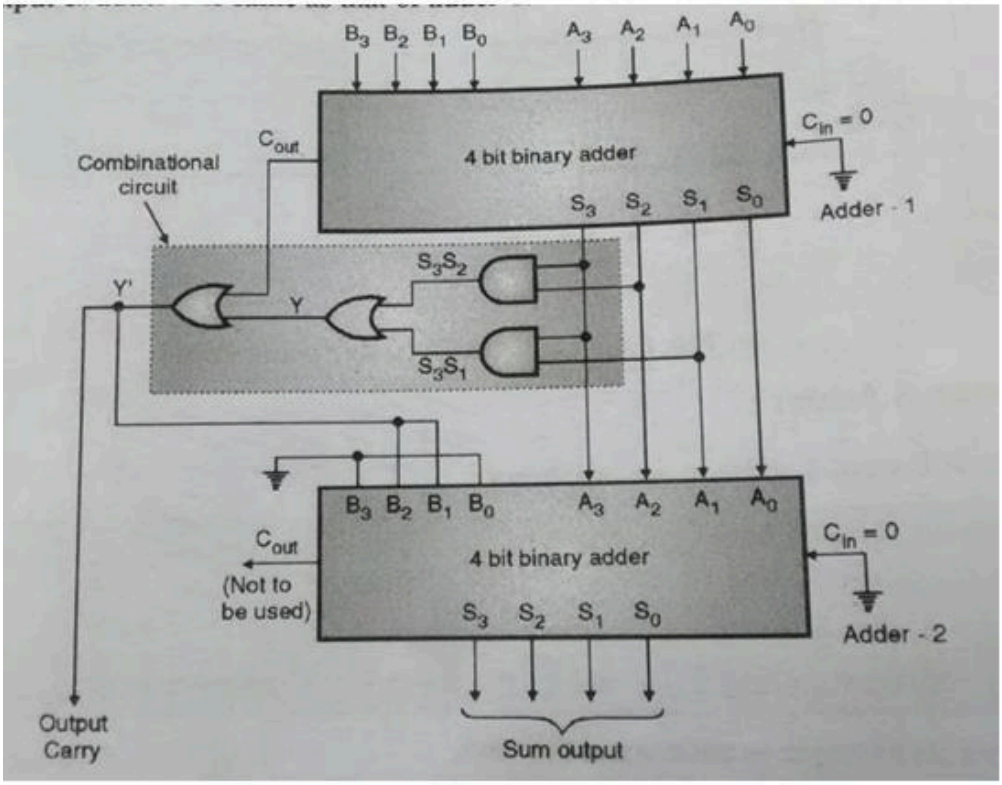


Fig. BCD addition using IC 7483

Operations of: $(011)_{BCD}+(1001)_{BCD}$

1	1	1	1	
	0	1	1	1
+	1	0	0	1
1	0	0	0	0

Thus, Cout

= 1

$S_3S_2S_1S_0$=0000

Hence, for adder, inputs will be $A_3A_2A_1A_0$=0000 and

$B_3B_2B_1B_0$=0110

This will give final output as Cout $S_3S_2S_1S_0$=10110

.

Therefore, $(0111)_{BCD}+(1001)_{BCD}$

= $(00010110)_{BCD}$

EXPERIMENT: 7 MULTIPLEXER AND DEMULTIPLEXER

AIM: To design and set up the following circuit

1) To design and set up a 4:1 Multiplexer (MUX) using only NAND gates.
2) To design and set up a 1:4 Demultiplexer(DE-MUX) using only NAND gates.
3) To verify the various functions of IC 74153(MUX) and IC 74139(DEMUX). 4) To set up a Half/Full Adder and Half/Full Subtractor using IC 74153.

LEARNING OBJECTIVE:

To learn about various applications of multiplexer and de-multiplexer

To learn and understand the working of IC 74153 and IC 74139

To learn to realize any function using Multiplexer

THEORY:

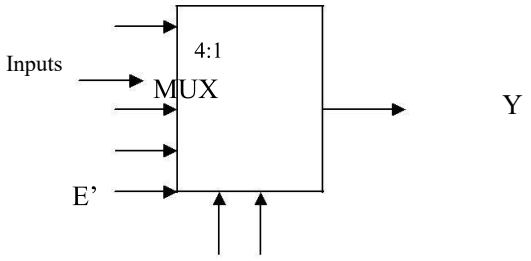
Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. 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 n depends on the input data bit that is selected. The general multiplexer circuit has 2 input signals, n control/select signals and 1 output signal.

De-multiplexers perform the opposite function of multiplexers. They transfer a small number of information units (usually one unit) over a larger number of channels under the control of selection signals. The general de-multiplexer circuit has 1 input signal, n n control/select signals and 2 output signals. De-multiplexer circuit can also be realized using a decoder circuit with enable.

COMPONENTS REQUIRED:

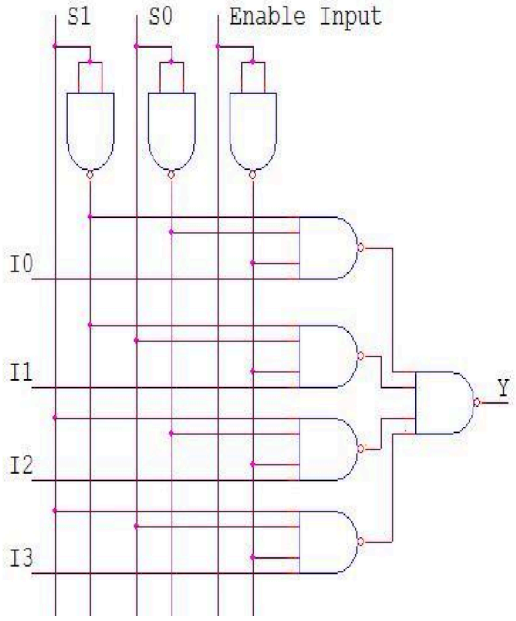
IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, Patch Cords & IC Trainer Kit.

i) 4:1 MULTIPLEXER



Output Y= E'S1'S0'I0 + E'S1'S0I1 + E'S1S0'I2 + E'S1S0I3

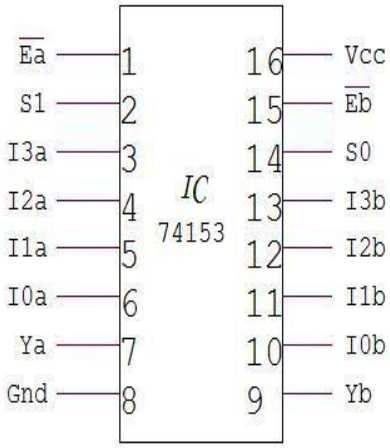
REALIZATION USING NAND GATES

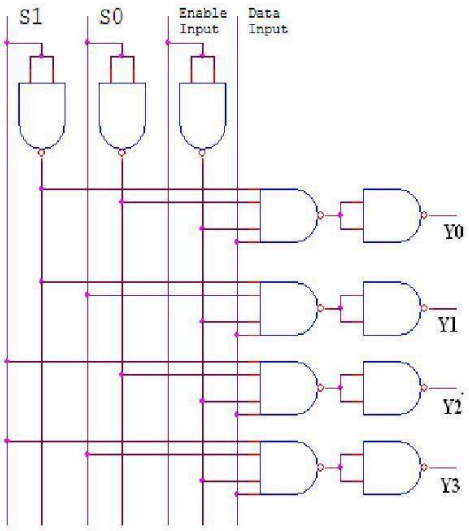


TRUTH TABLE

Select Inputs		Enable Input	Inputs				Out puts
S_1	**S_0**	**E**	**I_0**	**I_1**	**I_2**	**I_3**	**Y**
X	X	1	X	X	X	X	0
0	0	0	0	X	X	X	0
0	0	0	1	X	X	X	1
0	1	0	X	0	X	X	0
0	1	0	X	1	X	X	1
1	0	0	X	X	0	X	0
1	0	0	X	X	1	X	1
1	1	0	X	X	X	0	0
1	1	0	X	X	X	1	1

VERIFY IC 74153 MUX (DUAL 4:1 MULTIPLEXER)





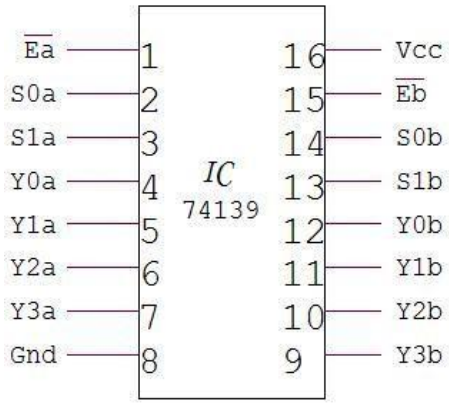
Enable Inputs	Data Input	Select Inputs		Outputs			
E	**D**	**S_1**	**S_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
1	0	X	X	X	X	X	X
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	0

ii) DE-MUX USING NAND GATES

VERIFICATION OF IC 74139 (DEMUX)

TRUTH TABLE

Inputs			Outputs			
Ea	**S1**	**S0**	**Y3**	**Y2**	**Y1**	**Y0**
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1



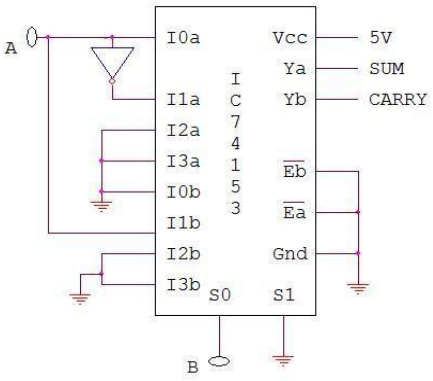
HALF ADDER USING MUX:

DESIGN:

TRUTH TABLE

Inputs		Outputs	
A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

CIRCUIT:



I0	I1
0	1
2	3
A	**A'**
I0	I1
0	1
2	3
0	**A**

Inputs			Outputs	
A	**B**	**C**	**S**	**C**
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

FULL ADDER USING MUX:

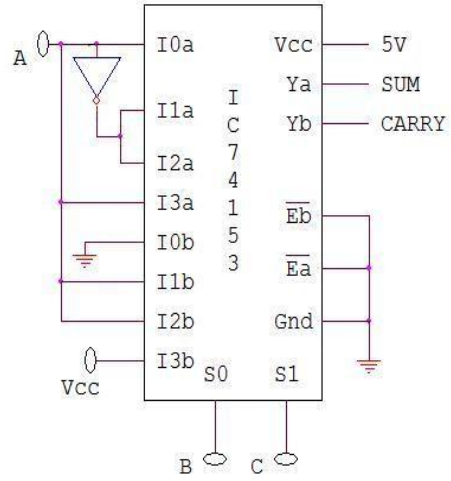
DESIGN:SUM CARRY

I0	I1	I3
0	1	2
4	5	6
A	**A'**	**A'**

I0	I1	I3	I3
0	1	2	3
4	5	6	7
0	**A**	**A**	**1**

TRUTH TABLE

FULL ADDER CIRCUIT



HALF SUBTRACTOR USING MUX:

DESIGN:

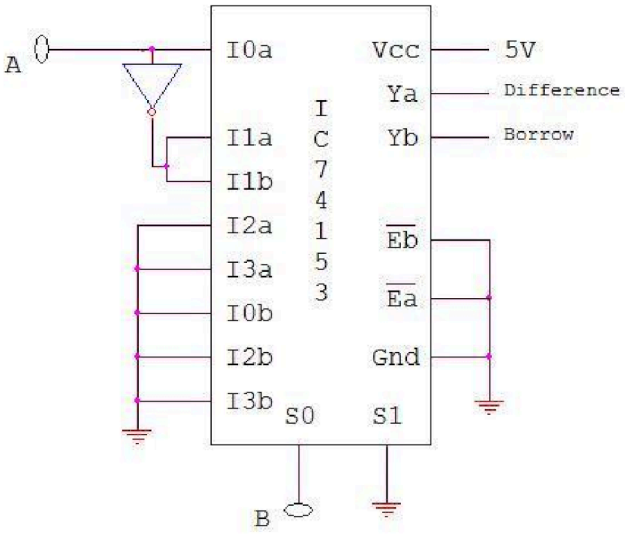
DIFFERENCE

I0	I1
0	1
2	3
A	**A'**

BORROW

I0	I1
0	1
2	3
0	**A'**

CIRCUIT:



TRUTH TABLE

Inputs		**Outputs**	
A	**B**	**D**	**Br**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

FULL SUBTRACTOR USING MUX:

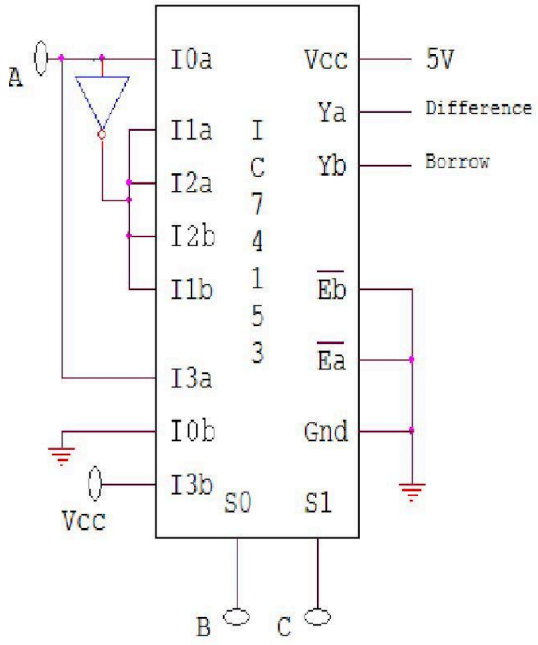
DESIGN:

DIFFERENCE

I0	I1	I2	I3
0	1	2	3
4	5	6	7
A	**A'**	**A'**	**A**

BORROW

I0	I1	I2	I3
0	1	2	3
4	5	6	7
0	**A'**	**A'**	**1**



TRUTH TABLE

Inputs			**Outputs**	
A	**B**	**C**	**D**	**Br**
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

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.

RESULT: Adder and subtractor circuits are realized using multiplexer IC 74153.

VIVA QUESTIONS:

1) What is a multiplexer?
2) What is a de-multiplexer?
3) What are the applications of multiplexer and de-multiplexer?

4) Derive the Boolean expression for multiplexer and de-multiplexer. 5) How do you realize a given function using multiplexer

6) What is the difference between multiplexer & demultiplexer?

7) In 2n to 1 multiplexer how many selection lines are there?

8) How to get higher order multiplexers?

9) Implement an 8:1 mux using 4:1 muxes?

EXPERIMENT: 7 MULTIPLEXER AND DEMULTIPLEXER

AIM: To design and set up the following circuit

1) To design and set up a 4:1 Multiplexer (MUX) using only NAND gates.
2) To design and set up a 1:4 Demultiplexer(DE-MUX) using only NAND gates.
3) To verify the various functions of IC 74153(MUX) and IC 74139(DEMUX).
4) To set up a Half/Full Adder and Half/Full Subtractor using IC 74153.

LEARNING OBJECTIVE:

To learn about various applications of multiplexer and de-multiplexer

To learn and understand the working of IC 74153 and IC 74139

To learn to realize any function using Multiplexer

THEORY:

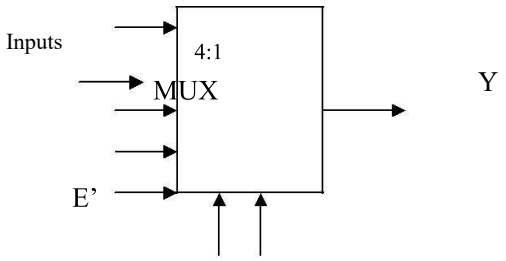
Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output. 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 2^n input signals, n control/select signals and 1 output signal.

De-multiplexers perform the opposite function of multiplexers. They transfer a small number of information units (usually one unit) over a larger number of channels under the control of selection signals. The general de-multiplexer circuit has 1 input signal, n control/select signals and 2^n output signals. De-multiplexer circuit can also be realized using a decoder circuit with enable.

COMPONENTS REQUIRED:

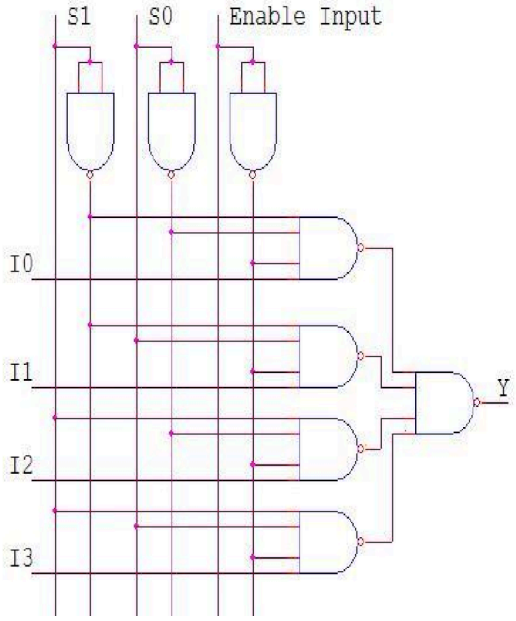
IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, Patch Cords & IC Trainer Kit.

i) 4:1 MULTIPLEXER



Output Y= E'S1'S0'I0 + E'S1'S0I1 + E'S1S0'I2 + E'S1S0I3

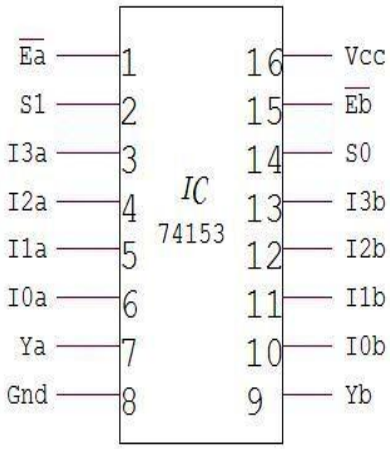
REALIZATION USING NAND GATES

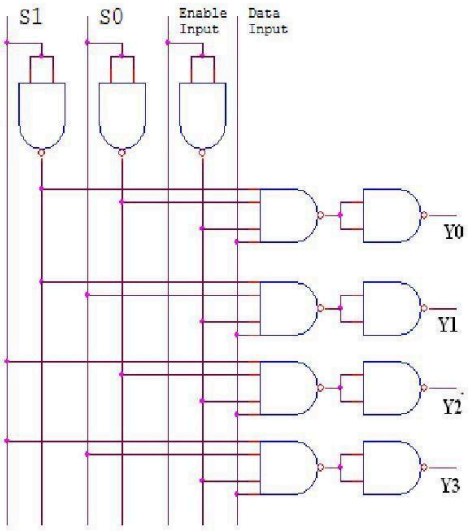


TRUTH TABLE

Select Inputs		Enable Input	Inputs				Out puts
S_1	**S_0**	**E**	**I_0**	**I_1**	**I_2**	**I_3**	**Y**
X	X	1	X	X	X	X	0
0	0	0	0	X	X	X	0
0	0	0	1	X	X	X	1
0	1	0	X	0	X	X	0
0	1	0	X	1	X	X	1
1	0	0	X	X	0	X	0
1	0	0	X	X	1	X	1
1	1	0	X	X	X	0	0
1	1	0	X	X	X	1	1

VERIFY IC 74153 MUX (DUAL 4:1 MULTIPLEXER)



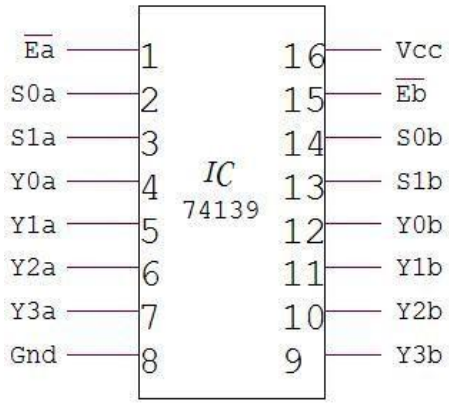


Enable Inputs	Data Input	Select Inputs		Outputs			
E	**D**	**S_1**	**S_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
1	0	X	X	X	X	X	X
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	0

ii) DE-MUX USING NAND GATES

VERIFICATION OF IC 74139 (DEMUX) TRUTH TABLE

Inputs			Outputs			
Ea	**S_1**	**S_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1



L ogi

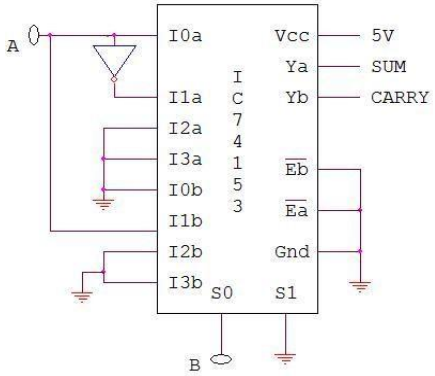
HALF ADDER USING MUX:

Inputs		Outputs	
A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

I0	I1
0	1

2	3
0	**A**
I0	I1
0	1
2	3
A	**A'**

D ESIGN:



Inputs			Outputs	
A	**B**	**C**	**S**	**C**
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

FULL ADDER USING MUX:

DESIGN:

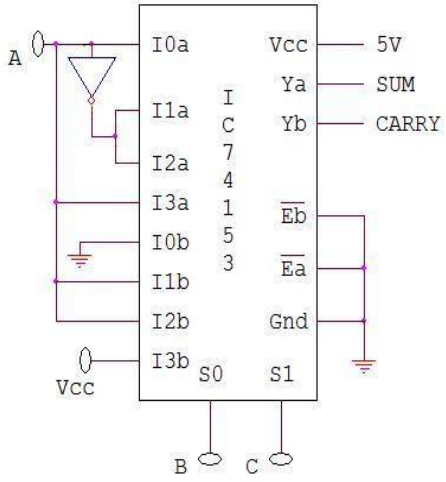
SUM CARRY

I0	I1	I3
0	1	2
4	5	6
A	**A'**	**A'**

I0	I1	I3	I3
0	1	2	3
4	5	6	7
0	**A**	**A**	**1**

TRUTH TABLE

FULL ADDER CIRCUIT



HALF SUBTRACTOR USING MUX:

DESIGN:

DIFFERENCE

I0	I1
0	1
2	3

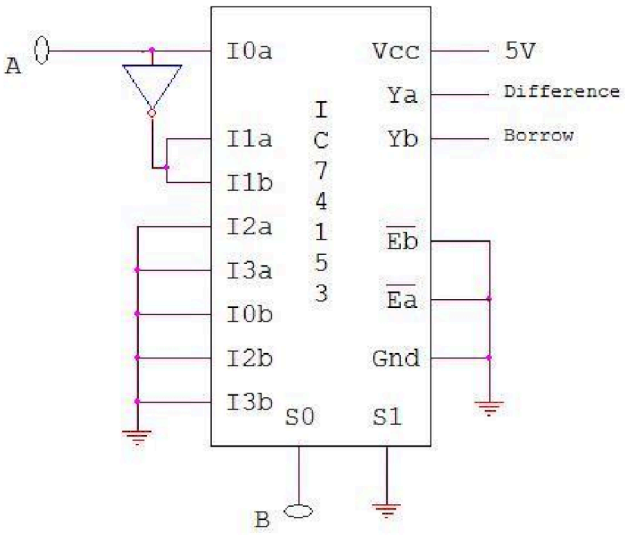
BORROW

I0	I1
0	1
2	3

A	A’

0	A’

CIRCUIT:



TRUTH TABLE

Inputs		Outputs	
A	**B**	**D**	**Br**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

FULL SUBTRACTOR USING MUX:

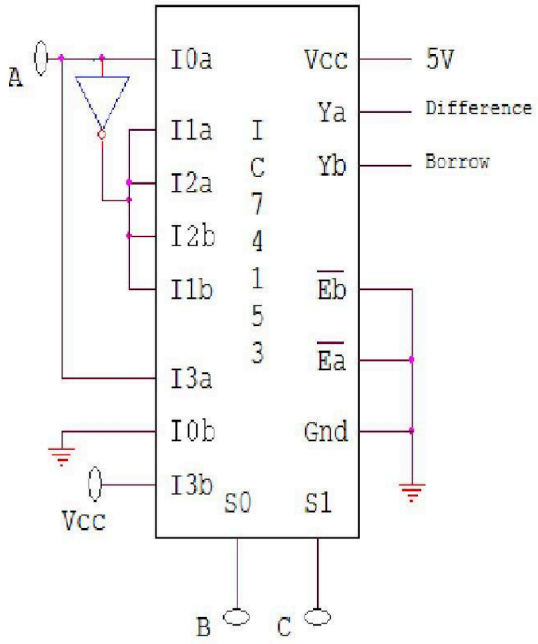
DESIGN:

DIFFERENCE

I0	I1	I2	I3
0	1	2	3
4	5	6	7
A	**A'**	**A'**	**A**

BORROW

I0	I1	I2	I3
0	1	2	3
4	5	6	7
0	**A'**	**A'**	**1**



TRUTH TABLE

Inputs			Outputs	
A	**B**	**C**	**D**	**Br**
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

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.

RESULT: Adder and subtractor circuits are realized using multiplexer IC 74153.

VIVA QUESTIONS:

1) What is a multiplexer?
2) What is a de-multiplexer?
3) What are the applications of multiplexer and de-multiplexer?

4) Derive the Boolean expression for multiplexer and de-multiplexer. 5) How do you realize a given function using multiplexer

6) What is the difference between multiplexer & demultiplexer?

7) In 2n to 1 multiplexer how many selection lines are there?

8) How to get higher order multiplexers?

9) Implement an 8:1 mux using 4:1 muxes?

EXPERIMENT: 9 DECODERS

AIM: To realize a decoder circuit using basic gates and to verify IC 74LS139

LEARNING OBJECTIVE:

To learn about working principle of decoder

To learn and understand the working of IC 74LS139

To realize using basic gates as well as universal gates

COMPONENTS REQUIRED:

IC74LS139, IC 7400, IC 7408, IC 7432, IC 7404, IC 7410, Patch chords, & IC Trainer Kit

THEORY:

A decoder is a combinational circuit that connects the binary information from 'n' input lines to n a maximum of 2 unique output lines. Decoder is also called a min-term generator/max-term generator. A min-term generator is constructed using AND and NOT gates. The appropriate output is indicated by logic 1 (positive logic). Max-term generator is constructed using NAND gates. The appropriate output is indicated by logic 0 (Negative logic).

The IC 74139 accepts two binary inputs and when enable provides 4 individual active low outputs. The device has 2 enable inputs (Two active low).

CIRCUIT DIAGRAM:

2:4 DECODER (MIN TERM GENERATOR):

TRUTH TABLE:

INPUT		**OUTPUT**			
A	**B**	**Y0**	**Y1**	**Y2**	**Y3**
0	**0**	**1**	**0**	**0**	**0**
0	**1**	**0**	**1**	**0**	**0**
1	**0**	**0**	**0**	**1**	**0**
1	**1**	**0**	**0**	**0**	**1**

BOOLAEN EXPRESSIONS:

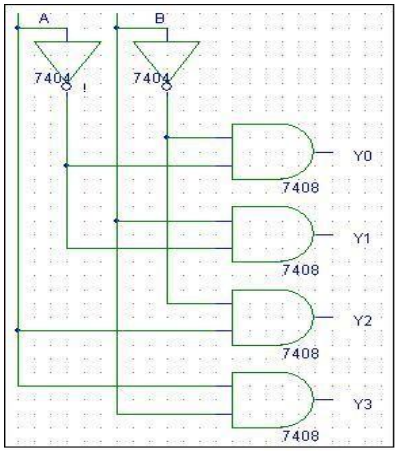
$Y0 \cdot \overline{AB}$

$Y1 \cdot A\overline{B}$

$Y2 \cdot AB$

$Y3 \cdot AB$

CIRCUIT DIAGRAM:

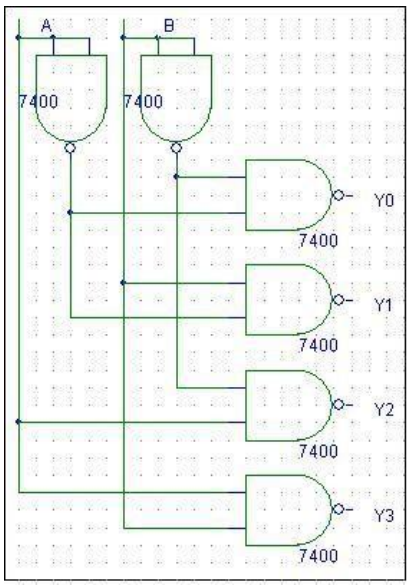


2:4 DECODER (MAX TERM GENERATOR):

TRUTH TABLE:

INPUT		**OUTPUT**			
A	**B**	**Y0**	**Y1**	**Y2**	**Y3**
0	**0**	**0**	**1**	**1**	**1**
0	**1**	**1**	**0**	**1**	**1**
1	**0**	**1**	**1**	**0**	**1**
1	**1**	**1**	**1**	**1**	**0**

CIRCUIT DIAGRAM:



PROCEDURE:

1. Make the connections as per the circuit diagram.
2. Change the values of G1, G2A, G2B, A, B, and C, using switches.
3. Observe status of Y0, to Y7 on LED's. 4. Verify the truth table.

RESULT: Verified the Operation of 3 to 8 Decoder

VIVA QUESTIONS:

1. What are the applications of decoder?
2. What is the difference between decoder & encoder? $_{n}$
3. For n- 2 decoder how many i/p lines & how many o/p lines?
4. What are the different codes & their applications?
5. What are code converters?
6. Using 3:8 decoder and associated logic, implement a full adder?
7. Implement a full subtractor using IC 74138? 8. What is the difference between decoder and de-mux?

EXPERIMENT: 10 BCD TO 7-SEGMENT DECODER/DRIVER

AIM: To set up and test a 7-segment static display system to display numbers 0 to 9.

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

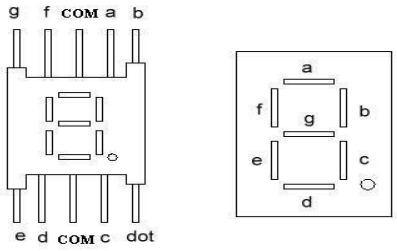
COMPONENTS REQUIRED:

IC7447, 7-Segment display (common anode), Patch chords, 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.

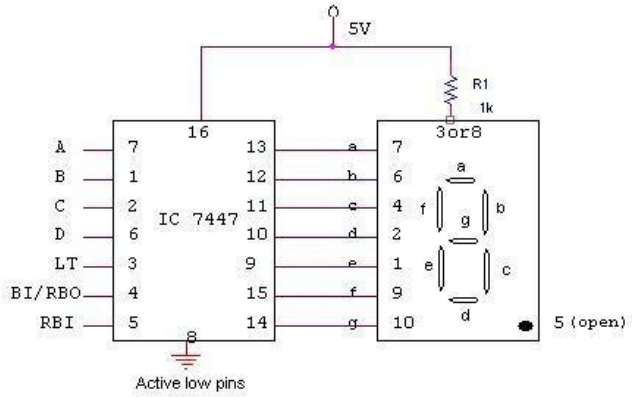


Seven-Segment Display

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 n to a maximum of 2 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?

EXPERIMENT: 11 ENCODERS

AIM:

1. To set up a circuit of Decimal-to-BCD Encoder using IC 74147.
2. To design and set up a circuit of Hexadecimal-to-Binary Encoder using IC 3. 74148 Encoders and IC 74157 Multiplexer

LEARNING OBJECTIVE:

To learn about various applications of Encoders To learn and understand the working of IC 74147 , IC 74148 & IC 74157

To learn to do code conversion using encoders

COMPONENTS REQUIRED:

IC 74147, IC 74148, IC 74157, Patch chords & IC Trainer Kit

THEORY:

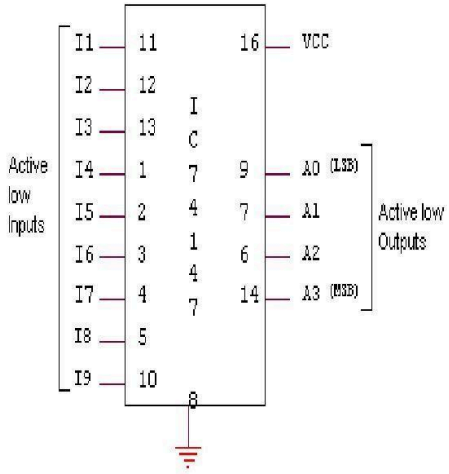
An encoder performs a function that is the opposite of decoder. It receives one or more signals in an encoded format and output a code that can be processed by another logic circuit. One of the advantages of encoding data, or more often data addresses in computers, is that it reduces the number of required 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.

CIRCUIT DIAGRAM:

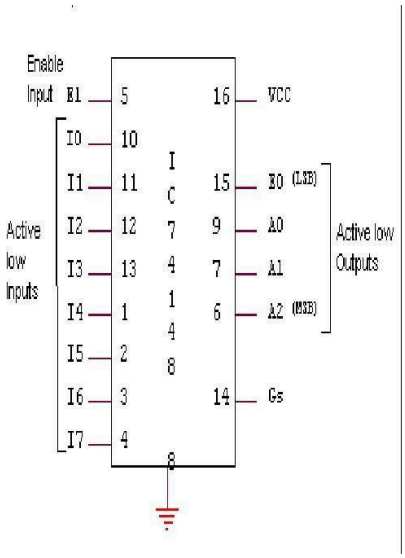
1) DECIMAL-TO BCD ENCODER USING IC 74147. TRUTH TABLE

INPUTS									OUTPUTS			
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	A_3	A_2	A_1	A_0
1	1	1	1	1	1	1	1	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1

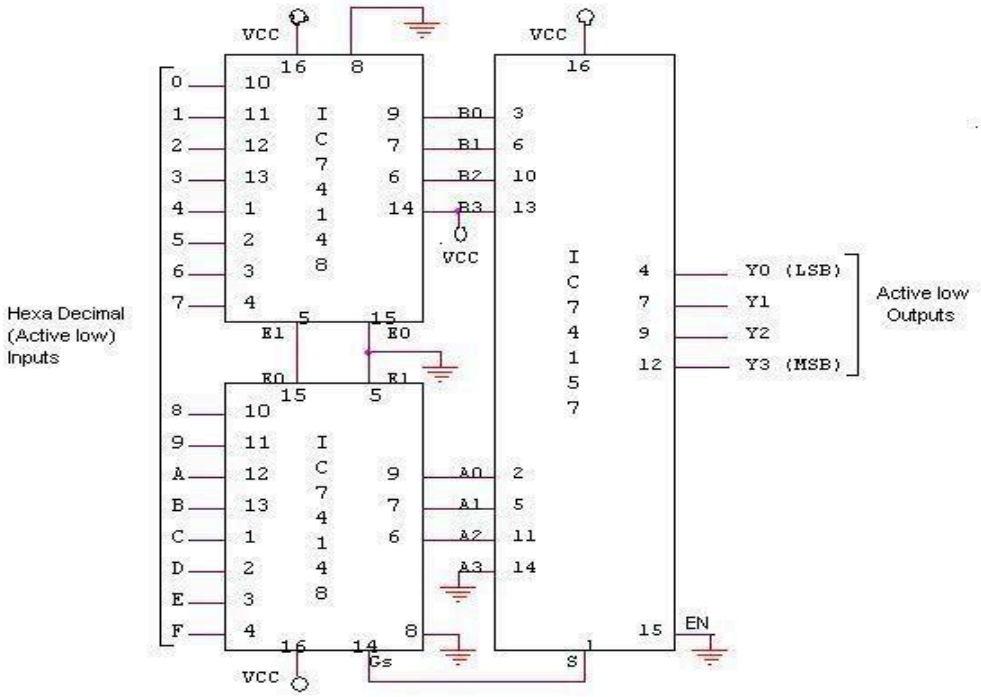


2) OCTAL TO BINARY ENCODER USING IC 74148. TRUTH TABLE

Inputs									Outputs				
E_1	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A_2	A_1	A_0	G_S	E_0
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	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1



3) HEXADECIMAL TO BINARY ENCODER



TRUTH TABLE

INPUTS																OUTPUTS			
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	Y_3	Y_2	Y_1	Y_0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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?