VISHAWKARMA SCIENTIFIC WORKS

AN ISO 9001 : 2015 Certified Company

761, Dina Mandi, Ambala Cantt website: www.viskawmeters.co.in

contact: +91 9953241920, 8882030155 Email- viskawmeters@gmail.com

Viskaw® INDIA

ISO 9001

LAB MANUAL

BREAD BOARDS AND IC

Scanned with CamScanner

Index

Contents

Scanned with CamScanner

Instruction Manual

For

Digital IC Circuit Trainer

Digital IC Circuit Trainer offers a unique entry into the world of digital electronics.
The system combines simple, easy to use, IC's for GATES, arithmatic operations flip-flops, power supply, input and outpu
states with a versatile solderless bread board.
Students know how digital electronics can implement logic circuit in a matter of seconds on the bread board.
This unique approach enables the unit to be used by the absolute beginner. Yet it may also be usefully employed ir
advanced project work.

Features

DC Regulated Power Supplies

1.	Output voltages	: One fixed DC regulated power supply of 5V/1Amp One fixed output DC regulated power supply of ±15V/ 500mA
2.	Load regulation	: + 0.2%
3.	Line regulation	: + 0.05%
4.	Ripple	: Less than 3mV RMS
5.	Protections	: Short circuit & over load protected
6.	Power requirement	: 230 VAC +10% , 50 Hz
7.	Voltmeter	: 3.5 Digit Dual range (2V/20V)

Logic Input/Output

8. Inbuilt 10 logic inputs, logic '0' & logic '1' selectable using SPDT switches, 10 logic output indicators.
9. 1Hz monoshot clock pulse, automatic clock pulse generator having clock pulses output 1Hz, 1KHz, & 10KHz selectable through band switch
10. Seven Segment display with seven I/P terminals for seven logic inputs.

Bread Board Strips

10. 2 Nos vertically common strips (640 Tie points each)
11. 4 Nos horizontally common strips(100 Tie points each)

STANDARD ACCESSORIES

The following integrated circuits & patchcords are supplied with the logic computer.

1.	7400	:	Quad 2 input NAND GATE.	2.	7402	:	Quad 2 input NOR GATE.
3.	7404	:	Hex invertor.	4.	7408	:	Quad 2 input AND GATE.
5.	7411	:	Triple - 3 input AND GATES.	6.	7420	:	4 input NAND GATE.
7.	7427	:	Triple - 3 input NOR GATES.	8.	7430	:	8 - input NAND GATE.
9.	7432	:	Quad 2 inputs OR GATE.	10.	7442	:	BCD to Decimal Decoder.
11.	7447	:	BCD to 7-Segment Decoder.	12.	7472	:	AND- GATES JK Flip-Flop.
13.	7474	:	Dual D type Flip Flops	14.	7476	:	Dual JK Flip Flops **(2Pcs.)**
15.	7486	:	Quad exclusive - OR GATE.	16.	7490	:	Decade Counter.
17.	7495	:	Shift Register.	18.	74153	:	1 to 4 line Demultiplexer.
19.	74155	:	4 to 1 line Multiplexer.	20.	74193	:	Synchronous Counter.

21. Hook up wire 5 mt. suitable for Bread Board.
22. Eighteen Patchcords with 4 mm Plug on single side.

Scanned with CamScanner

Experiments to be Performed

1. Study of OR, AND, NOT, NAND, NOR, EX-OR Gates & Vertification of their truth tables.
2. Verification of Boolean Identities & Demorgan's theorems.
3. Study & Verification of truth tables of Digital Adders & Subtractors .
4. Study and verification of truth tables of comparators.
5. Study of code converters i.e. binary to gray, gray to binary code
6. Study of flip flops and verification of their truth tables.
7. Study of Counters & Shift Registers,memory register and verification of their truth tables.
8. Study of Encoders & Decoders and verification of their truth tables.
9. Study of Multiplexers, Demultiplexers and verification of their truth tables.
10. Study of TTL-CMOS & CMOS-TTL Interfacing.

Experiment 1 :-

Study of OR, AND, NOT, NAND, NOR, EX-OR GATES and verification of their truth tables.

NOTE:- Pin description of all logic IC's are given on Page No. 40.

Theory

All the digital equipment simple or complex are derived from just a ten basic circuits, called logic elements. There are two basic types of digital logic circuits i.e. "decision making" and "memory". Decision making elements monitor the binary inputs and produce outputs based on the inputs state and operational characteristics of the logic element. Memory elements are used to store binary data's. The declsion making circuits are called 'GATES' while the memory circuits are made up and called 'Flip-Flop.'

The GATES are logic circuits & act as an ON/OFF switch also. The GATE shows either ON (HIGH) or OFF (LOW) state in the digital electronics. There are three basic decision making elements of gates i.e. OR, AND and NOT (INVERTER). From these three operations, two more operations have been derived, the NAND operation and the NOR operation. These operations have become very popular and are widely used, the reason being thay only one type of gates, no. of NAND & NOR GATES are sufficient for the realization of any logical expression. Because of this reason, NAND and NOR GATES are known as universal GATES.

A GATE is a circuit with one output and two or more input channels, an output signal occurs only for certain combinations of input signals. Logic circuits are used to perform various computer functions.

Verification of 'OR' GATE :-

'OR' GATE can have two or more inputs and signal output. It is defined as whenever anyone or all the input are high then output must be high otherwise output is low. For 2-input 'OR' GATE we are using IC 7432, Pin configuration, Circuit diagram and truth table are shown in fig. 1, and table 1 respectively. Mathematically it is expressed as Y = A+B (Y equals A or B).

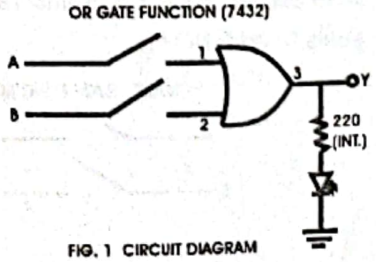


FIG. 1 CIRCUIT DIAGRAM

Scanned with CamScanner

Procedure :-

1. Put the IC in lower bread board and connect 5volts DC supply from pin No. 30 and ground .
   pin No.35 of upper bread board meant for connections.
2. Give input signal to logic GATE from pin no. 13,17, 21or 25 of upper bread board.
3. Connect output of GATE to LED indicator pin no. 40, 46, 52 or 58 of upper, bread board.
4. Connect the mains lead to 230 V AC and throw the power switch to 'ON' position. The jewel light will glow indicating that the trainer is ready for use.
5. Give the HIGH or LOW singals from debunced logic switch and observe the output (1 or 0) by (ON or OFF) indicator.
6. Verify the truth table as given in Table 1.

TRUTH TABLE - 1

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

POSITIVE LOGIC

Y = A+B

Verification of 'AND' GATE :-

'AND' GATE can have two or more inputs and a single output. It is defined as when all the inputs are high then output must be high otherwise output is low. For a 2 - inputs 'AND' GATE using IC 7408. Pin configuration, circuit diagram and truth table symbol are shown in Fig. 2 and table 2 respectively. Mathematically it is expressed as Y=A.B (Y equals A and B).

AND GATE FUNCTION (7408)

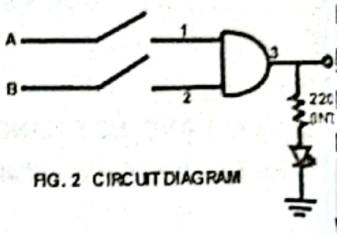


FIG. 2 CIRCUIT DIAGRAM

Procedure

Make the circuit of Fig. (2) and follow procedure of Experiment No.

1. Verify the truth table as given in Table 2.

TRUTH TABLE - 2

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

POSITIVE LOGIC

Y = AB

Verification of 'NOT' GATE :-

The simplest from of logic circuit is the INVERTER or NOT GATE. It has one input and one output terminal. It is defined as whenever input is high then output is low and vice-versa or we can say that the inverter is a logic element whose output state is always opposite of its input state. For that function we are using IC 7404. Pin configuration, circuit diagram and truth table are shown in Fig. 3 and Table 3. Mathematically it is expresses as Y = A (read as Y euals A NOT).

NOT GATE FUNCTION (7404)

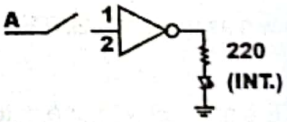


FIG. 3 CIRCUIT DIAGRAM

Procedure :-

Make the circuit of Fig. 3 and follow procedure of Experiment No. 1. Verify the truth table as given in Table 3.

TRUTH TABLE - 3

A	Y
0	1
1	0

POSITIVE LOGIC

$Y = \overline{A}$

Verification of 'NAND' GATE :-

The term 'NAND' is a contraction of the expression NOT - AND. A 'NAND' GATE therefore, is 'AND' GATE followed by an inverter. It is defined as whenever all the inputs are high then output must be low otherwise output is high. For 2 - inputs 'NAND' GATE we are using IC 7400. Pin configuration, circuit diagram and truth table are shown in Fig. 4 and Table 4. Mathematically it is expressed as Y = A.B (Y euals A nad B NOT).

TRUTH TABLE - 4

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

POSITIVE LOGIC

$Y = \overline{AB}$

NAND GATE FUNCTION (7400)

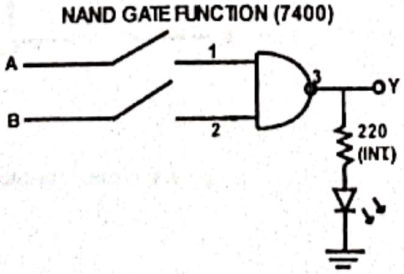


FIG. 4 CIRCUIT DIAGRAM

Scanned with CamScanner

Procedure :-

Make the circuit of Fig. 4. and follow procedure of Experiment No. 1. Verify the truth table as given in Table 4.

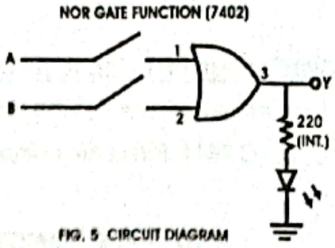


FIG. 5 CIRCUIT DIAGRAM

Verification of 'NOR' GATE :-

The 'NOR' GATE is an improved logic element used for implementing decision making logic functions. The term 'NOR' is a contraction for the expression 'NOT' 'OR'. The 'NOR' GATE is essentially a circuit containing the logic function of an 'OR' GATE and inverter. For 2 - inputs 'NOR' GATE we are using IC 7402. Pin cinfiguration diagram and truth table are shown in Fig. 5 and Table 5. It is defined as whenever all the inputs are low then output must be high otherwise output is low.

Procedure :-

Make the circuit of Fig. 5 and follow procedure of Experiment No. 1. Verify the truth table as given in 5

TRUTH TABLE - 5

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

POSITIVE LOGIC

$Y = A+B$

Verification of 'EX-OR' GATE :-

The 'Exclusive-OR' (EX-OR) operation is widely used in digital circuits. It is not a basic operation and can be performed using the basic gates. 'AND', 'OR' and 'NOT' or universal GATES 'NAND' or 'NOR'. It is similar to the basic 'OR' GATE expect that the output is low when both the inputs are high and low. The output of the gate is high when anyone output is high. For 2 - inputs 'EX-OR' GATE we are using IC 7486. Pin configuration, circuit diagram and truth table are shown in Fig. 6 and Table 6. 'Exclusive-OR' (EX-OR) circuits are used in applications like adder substraction, parity checkers etc. Mathematically expression of 'EX-OR' is given by $Y = AB + AB$.

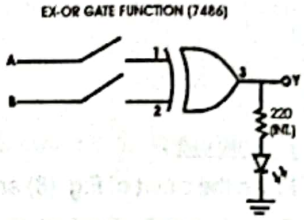


FIG .6 CIRCUIT DIAGRAM

Procedure :-

Make the circuit of Fig. 6 and follow procedure of Experiment No. 1. Verify the truth table as given in Fig. 6.

TRUTH TABLE - 6

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

POSITIVE LOGIC

$Y = AB+BA$

Verification of 3 - Input 'NOR' GATE :-

It is defined as whenever all the input are low then output must be high otherwise output is low. For 3 - inputs 'NOR' GATE we are using IC 7427. Pin configuration, circuit diagram and truth table are shown in Fig. 7 and Table 7.

Procedure

Make the cicuit of Fig. 7 and follow the procedure of Experiment no. 1 and verify the truth table as given in Table 7 .

NOR GATE FUNCTION (7427)

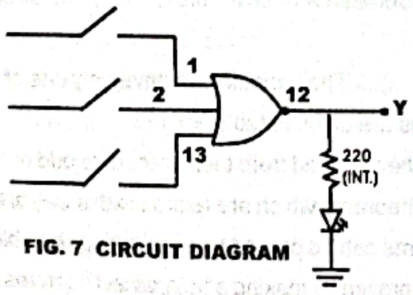


FIG. 7 CIRCUIT DIAGRAM

TRUTH TABLE - 7

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

POSITIVE LOGIC

$Y = A+B+C$

Scanned with CamScanner

Verification of 3 - Input 'AND' GATE :-

It is defined as when all the input high then output must be high otherwise output is low. For 3 input 'AND' GATE we a using IC 7411. Pin configuration, circuit diagram and truth.

table are shown in Fig. 8 and Table 8. It is mathematically expresse by Y = A.B.C.

AND GATE FUNCTION (7411)

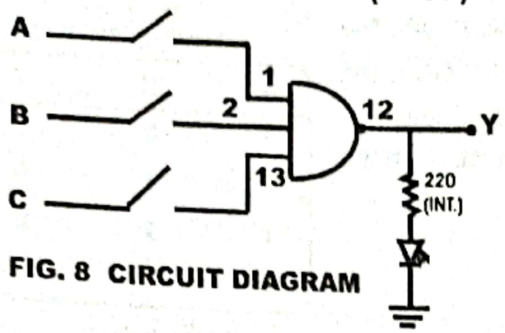


FIG. 8 CIRCUIT DIAGRAM

TRUTH TABLE - 8

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

POSITIVE LOGIC

Y = A.B.C

Procedure :-

Make the cicuit of Fig. (8) and follow the procedure of Experiment no. 1 and verify the truth table as given in Table 8.

Experiment 2 :To study the Boolean Algebraic Theorems.

Objective

To familiarise and verify the following Boolean Algebraic Theorems and to simplify and realize the following expression:

Integrated Circuit to be used :-

1. IC 7404 Hex Inverter (Six, 1 - Input 'NOT' GATES)
2. IC 7408 Quad 2 - Input 'AND GATES

   OR

   IC 7411 Triple 3 - input 'AND' GATES where necessary.
3. IC 7432 Quad 2 - Input "OR' GATES

   OR

   IC 7427 Triple 3 - input 'NOR' GATES in combination with IC 7404 to make 3 - input 'OR' GATES as follows where necessary.

Theory

The digital signals are discreate in nature and can only assume one of the two values '0' or '1'. A number system based on these two digits is known as binary number system. In the middle of 19th century, an English mathematician George Boole developed rules for manipulations of binary numbers, known as Boolean Algebra. This is the basis of all digital system like computers, calculators etc.

Binary variables can be represented by a letter symbol such as A, B, X, Y, ... The variable can have only one of the two positive values at any time, viz, '0' or '1'. The Boolean Algebraic Theorems are given in table No. (1).

From these theorems, we observe that the even numbered theorems can be obtained from their preceding odd numbered theorems by (i) interchanging and signs, and (ii) interchanging '0' and '1'.Theorems which are related in this way are called duals. Theorems 1 to 8 involve a single variable only. Each of these theorems can be proved by considering possible values of the variable.Theorema 9 to 20 involve more than one variable and can be proved by making a truth table.Theorems 21 and 22 are known as Demorgan's Theorems. These theorems can be proved by first considering the two variable case and then extending this result.To familiarise and verify the following Boolean Algebric Theorems and to simplify and relize the expression.

Scanned with CamScanner

Theorm - 1 :- **A + O=A**

Procedure :-

1. Make the circuit as shown in Fig. 9 and connect the inputs of 'OR' GATE, to the two input state sockets 'A' & 'B' and output of 'OR' GATE to the output indicator socket T_1 for LED L_1.

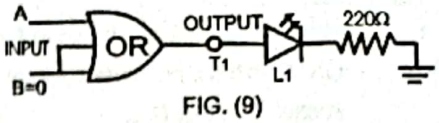


FIG. (9)

2. Set the input combinations by putting input state switches either '0' or '1' state. Input state switch of 'B' always should be zero and 'A' input state switch either '0' or '1' state.
3. Now verify the output, if input state switch of 'A' in '0' state. Output indicator LED L_1 should be 'OFF'. If 'A' in '1' state, then LED L_1 should be 'ON'. Hence the theorem is proved.

Theorem - 2 :- **A . 1 = A**

Procedure :-

1. Make the circuit as shown in Fig. 10 and connect the inputs of the 'AND' GATE, to the two input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'B' always in '1' state and switch 'A' either in '0' or '1' state.
3. Now verify the output. If B = 1 & A = 0, indicator LED L_1 should be 'OFF' and if A = 1 then LED L_1 Should be 'ON'. Hence the theorem is proved.

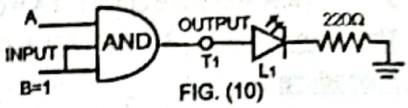


FIG. (10)

Theorem - 3 :- **A + 1 = 1**

Procedure :-

1. Make the circuit as shown in Fig. 11 and connect the inputs of 'OR' GATE, to the two input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'B' always in '1' state and switch 'A' either in '0' or '1' state.
3. Now verify the output. If B = 1 & A = 0 or 1 then indicator LED L_1 should be continuously 'ON'. Hence the theorem is proved.

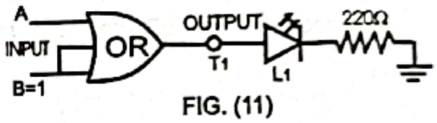


FIG. (11)

Theorem - 4 :- **A . 0 = 0**

Procedure :-

1. Make the circuit as shown in Fig. 12 and connect the inputs of 'AND' GATE, to the two input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'B' always in '0' state and switch 'A' either in '0' or '1' state.
3. Now verify the output. If B = 0 & A = 0 or 1 then indicator LED L_1 should be continuously 'OFF'. Hence the theorem is proved.

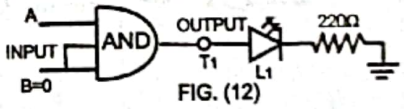


FIG. (12)

Theorem - 5 :- **A + A = A**

Procedure :-

1. Make the circuit as shown in Fig. 13 and connect the inputs of 'OR' GATE, to the input state sockets 'A' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'A' always either in '0' or '1' state.
3. Now verify the output. If A = 1 then indicator LED L_1 should be continuously 'ON' and if A = 0 then indicator LED L_1 should be 'OFF'. Hence the theorem is proved.

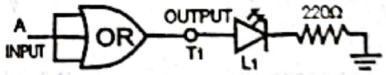


FIG. (13)

Theorem - 6 :- **A.A = A**

Procedure :-

1. Make the circuit as shown in Fig. 14 and connect the inputs of 'AND' GATE, to the input state sockets 'A' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'A' in '0' or '1' state.
3. Now verify the output. If A = 0, output indicator LED L_1 should be 'OFF' and if A = 1 should be 'ON'. Hence the theorem is proved.

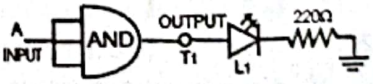


FIG. (14)

Scanned with CamScanner

Theorem - 7 :- $A + \overline{A} = 1$

Procedure :-

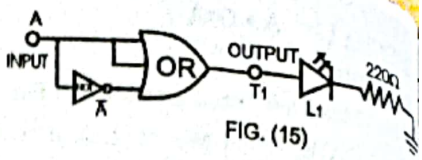


FIG. (15)

1. Make the circuit as shown in Fig. 15 and connect the inputs of the GATE, to the input state sockets 'A' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'A' either in '0' or '1' state.
3. Now verify the output. If A = 1 or 0, output indicator LED L_1 should be continuously 'ON'. Hence the theorem proved.

Theorem - 8 :- $A \cdot \overline{A} = 0$

Procedure :-

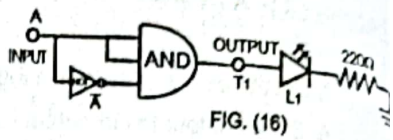


FIG. (16)

1. Make the circuit as shown in Fig. 16 and connect the GATE, to the input state sockets 'A' and output to the output indicator socket T_1 for LED L_1.
2. Put input state switch 'A' either in '0' or '1' state.
3. Now verify the output. If A = 0 then indicator LED L_1 should be continuously 'OFF'. Hence the theorem is proved.

Theorem - 9 :- $A.(B + C) = AB + AC$

Procedure :-

1. Make the circuit as shown in Fig. 17 and connect the GATE, to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state. (As shown in Truth Table No.9).
3. Now verify the output with the help of Truth Table No. 9. Output indicator LED L_1 & L_2 should be 'OFF' or 'ON' according to the Truth Table No. 9. Hence the theorem is proved.

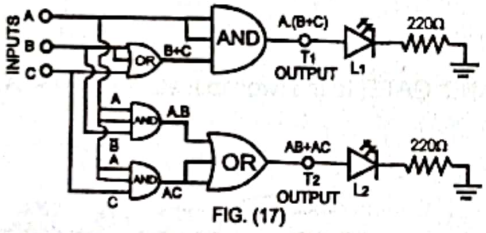


FIG. (17)

TRUTH TABLE -9

A	B	C	B+C	A.(B+C)	A.B	A.C	(AB+AC)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Theorem - 10 :- $A+BC = (A+B).(A+C)$

Procedure :-

1. Make the circuit as shown in Fig. 18 and connect the GATE, to the inputs of the GATE to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state. (As shown in Truth Table No. 10.

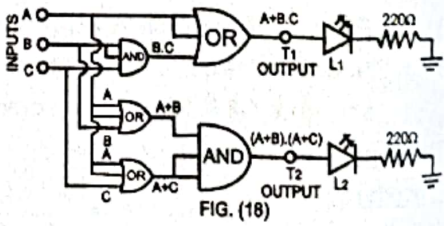


FIG. (18)

3. Now verify the output with the help of Truth Table No. 3. Output indicator LED L_1 & L_2 should be 'ON' or 'OFF' by input combinations according to the Truth Table No. 3, so it is proved that $A+BC = (A+B).(A+C)$.

Scanned with CamScanner

Theorem - 11 : $A+AB = A$

Procedure :-

1. Make the circuit as shown in Fig. (19) and connect the inputs of the GATE to the input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1.
2. Put the input state switches 'A' in '1' state and 'B' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No.11. If 'A' = '1' and 'B' = '0' or '1', the output indicator LED L_1 should be 'ON'.
4. Now verify the output also if 'A' = '0' and 'B' = '0' or '1', the output indicator LED L_1 should be 'OFF'. In this way the theorem is proved.

TRUTH TABLE - 10

A	B	C	B.C	A+B.C	A+B	A+C	(A+B).(A+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

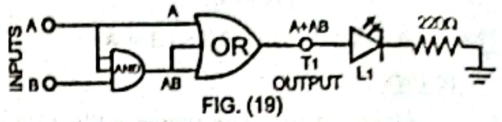


FIG. (19)

Theorem - 12 : $A.(A+B) = A$

Procedure :-

1. Make the circuit as shown in Fig. (20) and connect the inputs of the GATE to the input state sockets 'A' & 'B' and output, to the output indicator socket T_1 for LED L_1
2. Put the input state switches 'A' always in '1' or '0' state and switch.
   'B' either in 'o' or '1' state.
3. Now verify the output with the help of Truth Table No. 12. The output indicator LED L_1 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

TRUTH TABLE - 11

A	B	AB	A+AB
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

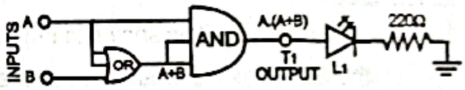


FIG. (20)

Theorem - 13 :- $A+\bar{A}B = (A+B)$

Procedure :-

1. Make the circuit as shown in Fig. 21 and connect the inputs of the GATE to the input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A' & 'B' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 13. The output indicator LED's L_1 & L_2 should be 'OFF' or 'ON' by input combinations, so it is proved that A+AB = A+B.

TRUTH TABLE - 12

A	B	A+B	A.(A+B)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

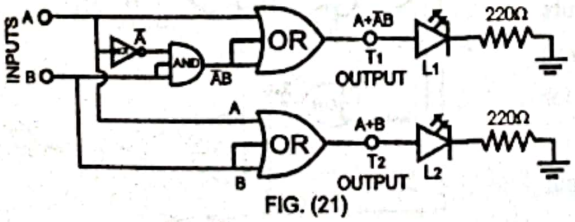


FIG. (21)

TRUTH TABLE - 13

A	B	$\bar{A}$	$\bar{A}B$	$A+\bar{A}B$	A+B
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Scanned with CamScanner

Theorem - 14 :- $A.(\bar{A}+B) = AB$

Procedure :-

1. Make the circuit as shown in Fig. 22 and connect the inputs of the GATE to the input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A' & 'B' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 14. Then output indicator LED L_1 & L_2 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

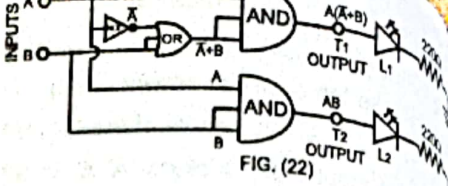


FIG. (22)

TRUTH TABLE - 14

A	B	$\bar{A}$	$\bar{A}$+B	A.($\bar{A}$+B)	AB
0	0	1	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	1	0	1	1	1

Theorem - 15 :- $AB+A\bar{B} = A$

Procedure :

1. Make the circuit as shown in Fig. (23) and connect the inputs of the GATE to the input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1.
2. Set the input combinations one by one by putting input state switches 'A' & 'B' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 15. The output indicator LED L_1 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

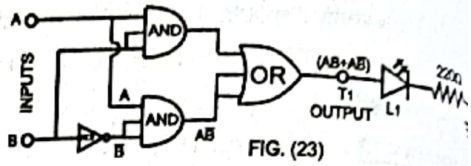


FIG. (23)

TRUTH TABLE - 15

A	B	$\bar{B}$	AB	A$\bar{B}$	AB+A$\bar{B}$
0	0	1	0	0	0
0	1	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1

Theorem - 16 : $(A+B).(A+\bar{B}) = A$

Procedure :-

1. Make the circuit as shown in Fig. (24) and connect the inputs of the GATE to the input state sockets 'A' & 'B' and output to the output indicator socket T_1 for LED L_1.
2. Set the input combinations one by one by putting input state switches 'A' & 'B' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 16. Then output indicator LED L_1 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

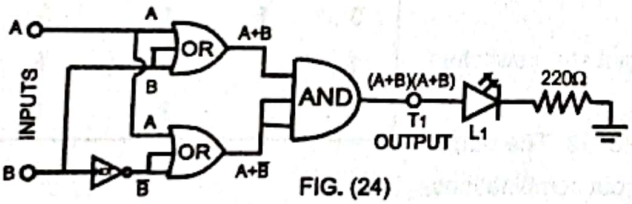


FIG. (24)

TRUTH TABLE - 16

A	B	$\bar{B}$	A+B	A+$\bar{B}$	(A+B).(A+$\bar{B}$)
0	0	1	0	1	0
0	1	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1

Theorem - 17 : $AB+\bar{A}C = (A+C).(\bar{A}+B)$

Procedure :-

1. Make the circuit as shown in Fig. (25) and connect the inputs of the GATE to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combina -tions one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state according to Truth Table - 17.

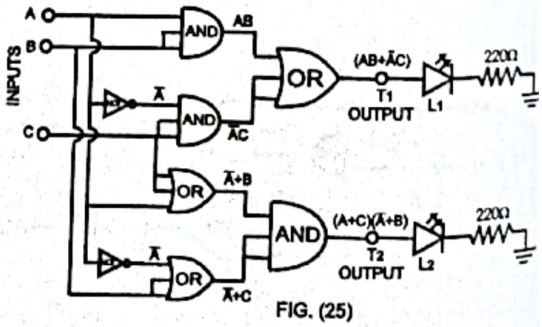


FIG. (25)

Scanned with CamScanner

3. Now verify the output with the help of Truth Table No. 10. The output indicator LED L_1 & L_2 should be 'OFF' or 'ON' by input combinations. Hence the theorem is proved.

TRUTH TABLE - 17

A	B	C	$\bar{A}$	AB	$\bar{A}C$	AB+$\bar{A}C$	A+C	$\bar{A}$+B	(A+C).($\bar{A}$+B)
0	0	0	1	0	0	0	0	1	0
0	0	1	1	0	1	1	1	1	1
0	1	0	1	0	0	0	0	1	0
0	1	1	1	0	1	1	1	1	1
1	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	1	0	0
1	1	0	0	1	0	1	1	1	1
1	1	1	0	1	0	1	1	1	1

Theorem - 18 :- **(A+B).($\bar{A}$+C) = AC+$\bar{A}$B**

Procedure :-

1. Make the circuit as shown in Fig. (26) and connect the inputs of the GATE to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state according to Truth Table - 18.
3. Now verify the output with the help of Truth Table No. 11. The output indicator LED L_1 & L_2 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

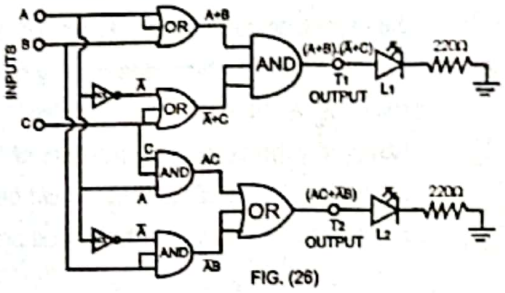


FIG. (26)

TRUTH TABLE - 18

A	B	C	$\bar{A}$	A+B	$\bar{A}$+C	(A+B).($\bar{A}$+C)	AC	$\bar{A}$B	AC+$\bar{A}$B
0	0	0	1	0	1	0	0	0	0
0	0	1	1	0	1	0	0	0	0
0	1	0	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1
1	0	0	0	1	0	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0	0	0
1	1	1	0	1	1	1	1	0	1

Theorem - 19 : AB+$\bar{A}$C+BC = AB+$\bar{A}$C

Procedure :

1. Make the circuit as shown in Fig. (27) and connect the inputs of the GATE to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 19. The output indicator LED L_1 & L_2 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

Scanned with CamScanner

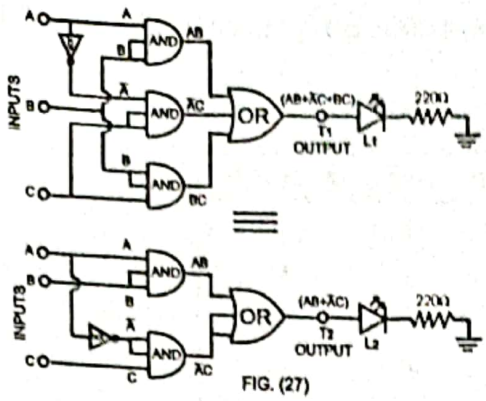


FIG. (27)

TRUTH TABLE - 19

A	B	C	$\bar{A}$	AB	$\bar{A}C$	BC	(AB+$\bar{A}$C+BC)(AB+$\bar{A}$C)
0	0	0	1	0	0	0	0
0	0	1	1	0	1	0	1
0	1	0	1	0	0	0	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	1	0	0	1
1	1	1	0	1	0	1	1

Theorem - 20 :- $(A+B).(\bar{A}+C).(B+C) = (A+B).(\bar{A}+C)$

Procedure :-

1. Make the circuit as shown in Fig. (28) and connect the inputs of the gate to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 20. The output indicator LED L_1 & L_2 should be 'ON' or 'OFF' by input combinations. Hence the theorem is proved.

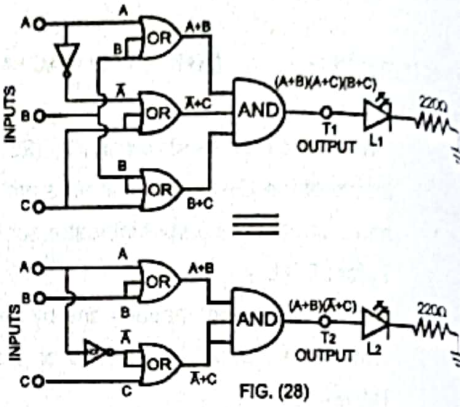


FIG. (28)

TRUTH TABLE - 20

A	B	C	$\bar{A}$	A+B	$\bar{A}$+C	B+C	(A+B).($\bar{A}$+C).(B+C)	(A+B).($\bar{A}$+C)
0	0	0	1	0	1	0	0	0
0	0	1	1	0	1	1	0	0
0	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	1	1	1
1	1	0	0	1	0	1	0	0
1	1	1	0	1	1	1	1	1

Theorem - 21 :- **Demorgan's Theorem - I** $\overline{A.B.C} = \bar{A}+\bar{B}+\bar{C}$

Procedure :-

1. Make the circuit as shown in Fig. (29) and connect the inputs of the GATE to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the following input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 21. The output indicator LED L_1 & L_2 should be 'OFF' or 'ON' by input combinations. So it is proved that A.B.C = A+B+C.

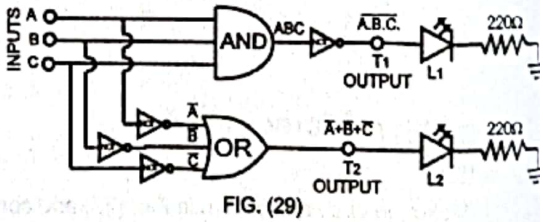


FIG. (29)

Scanned with CamScanner

TRUTH TABLE - 21

A	B	C	$\overline{A}$	$\overline{B}$	$\overline{C}$	ABC	$\overline{ABC}$	$\overline{A}+\overline{B}+\overline{C}$
0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	1	1
0	1	0	1	0	1	0	1	1
0	1	1	1	0	0	0	1	1
1	0	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	1	1
1	1	1	0	0	0	1	0	0

Theorem - 22 : Demorgan's Theorem - II $\overline{A+B+C} = \overline{A}.\overline{B}.\overline{C}$

Procedure :-

1. Make the circuit as shown in Fig. (30) and connect the inputs of the GATE to the input state sockets 'A', 'B' & 'C' and output to the output indicator socket T_1 for LED L_1 and T_2 for LED L_2.
2. Set the input combinations one by one by putting input state switches 'A', 'B' & 'C' either in '0' or '1' state.
3. Now verify the output with the help of Truth Table No. 22. The output indicator LED L_1 & L_2 should be 'OFF' or 'ON' by input combinations. So it is proved that A+B+C = A.B.C. of 'B' from, A (i.e, A-B), where 'A' & 'B' are I-bir signals is knows as a Half - Substractor. Its logic gate diagram and truth table are given in Table22. Here 'D' is the DIFFERENCE bit and 'C' the BORROW bit.

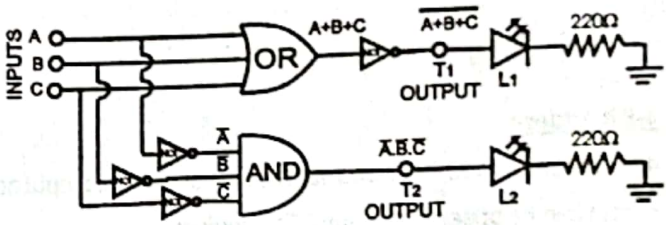


FIG. (30)

TRUTH TABLE - 22

A	B	C	$\overline{A}$	$\overline{B}$	$\overline{C}$	A+B+C	$\overline{A+B+C}$	$\overline{A}.\overline{B}.\overline{C}$
0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	0	0
0	1	0	1	0	1	1	0	0
0	1	1	1	0	0	1	0	0
1	0	0	0	1	1	1	0	0
1	0	1	0	1	0	1	0	0
1	1	0	0	0	1	1	0	0
1	1	1	0	0	0	1	0	0

Experiment- 3 Study and verification of truth tables of digital adders and subtractors.

Half Adder

Set up a half adder according to circuit fig.31 given using EX-OR (7486) & AND (7408) gates & verify the Truth Table 23.

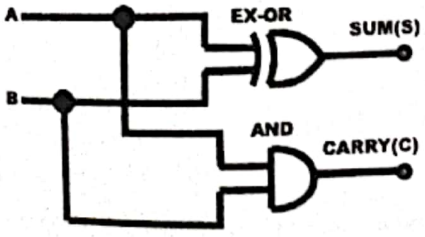


FIG. 31 HALF ADDER CIRCUIT DIAGRAM

TRUTH TABLE - 23

INPUTS		OUTPUT	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S=\overline{A}B+A\overline{B}$ $C=AB$

Scanned with CamScanner

Full - Adder

A Half - Adder has only 2-input terminals and there is no provision to add a carry coming from the lows order bits wh binary numbers are added. For this purpose a third input terminal is added and this circuit is used to add An, Bn & Cn where An & Bn are the nth order bits of the numbers 'A' & 'B' respectively and Cn-1 is the carry generated from addition the (n-1) the order bits.

The block diagram of a Full - Adder and its truth table are given in Fig. 32 & Table 24

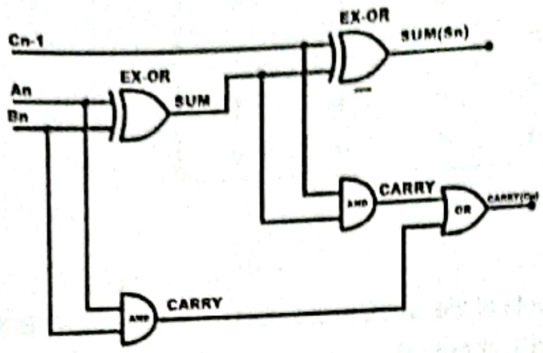


Fig 32 - FULL ADDER CIRCUIT DIAGRAM

TRUTH TABLE - 24

INPUT			OUTPUT	
An	Bn	Cn-1	Sn	Cn
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

$Sn = \bar{A}nBn\bar{C}n\text{-}1 + \bar{A}n\bar{B}nCn\text{-}1 +$

$An\bar{B}n\bar{C}n + AnBnCn\text{-}1$

$Cn = AnBn + BnCn\text{-}1 + AnCn\text{-}1$

4-Bit Adder-:

4-bit adder using IC 7483 and verify the two 4 bit data input additions. The pin description of IC 7483 is shown in Fig. 34 output can be observe through LEDs indicator.

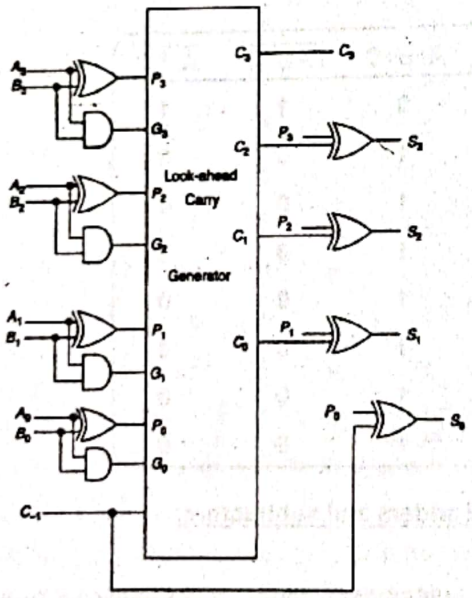


Fig.33

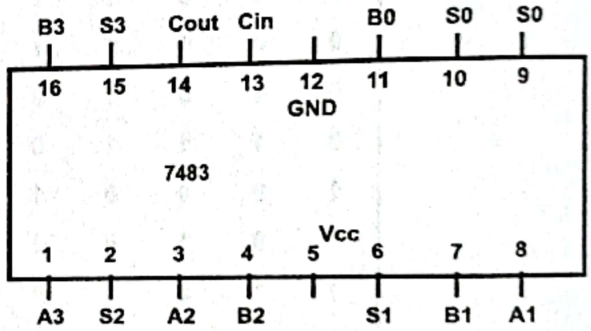


Fig.34

Where A3 A2 A1 A0--------First 4-bit Data

B3 B2 B1 B0--------Second 4-bit Data

S3 S2 S1 S0---------Adder output

Cout---------output Carry

Full Subtractor

Set up Full - Substractor circuit using 7486, 7408, 7432 & 7404 IC's as given in Fig. (35).

Verify the truth table experimentally as given in Fig. (25).

Scanned with CamScanner

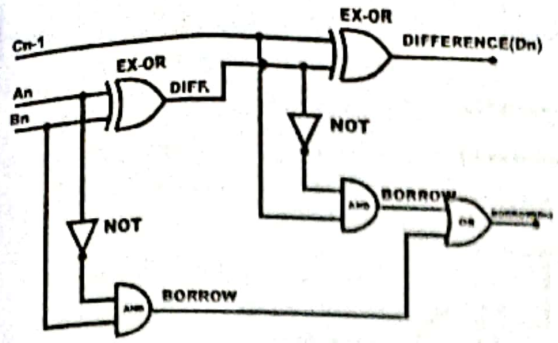


Fig 35 - FULL SUBTRACTOR CIRCUIT DIAGRAM

TRUTH TABLE - 25

INPUT			OUTPUT	
An	Bn	Cn-1	Dn	Bn
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

$Dn = \bar{A}nBn\bar{C}n\text{-}1 + \bar{A}n\bar{B}nCn\text{-}1 + An\bar{B}n\bar{C}n\text{-}1 + AnBnCn\text{-}1$

$Bn = \bar{A}nBn + \bar{A}nCn\text{-}1 + BnCn\text{-}1$

Experiment - 4 Study of Code Converters

Procedure

Binary to Gray Code Converter

Table No.26 shows the Gray Code of 4 Bit Binary Code.

Simplified expression for Gray Code from 4 Bit Binary code using K-map is

G3 '= B3

G2 = B2 ⊕ B3

G1 = B1 ⊕ B2

G0 = B0 ⊕ B1

Procedure

Set up binary to gray code converter using circuit given in figure no. 36. The circuit is using 3 EX-OR gate using IC 7486. Varify the gray code of respective 4 bit binary data.

Circuit Diagram

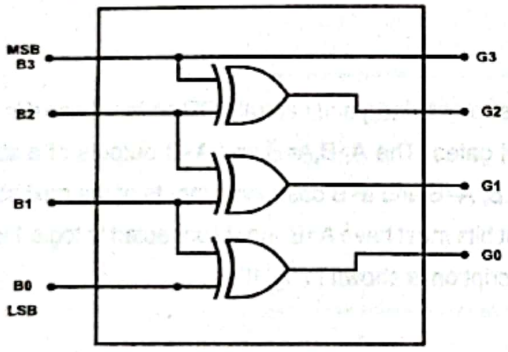


Fig.36

Binary to Gray Code Table No.26

B3,B2,B1,B0	G3,G2,G1,G0
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 0
0 1 0 0	0 1 1 0
0 1 0 1	0 1 1 1
0 1 1 0	0 1 0 1
0 1 1 1	0 1 0 0
1 0 0 0	1 1 0 0
1 0 0 1	1 1 0 1
1 0 1 0	1 1 1 1
1 0 1 1	1 1 1 0
1 1 0 0	1 0 1 0
1 1 0 1	1 0 1 1
1 1 1 0	1 0 0 1
1 1 1 1	1 0 0 0

Gray to Binary Code Converter

The simplified expression for binary code from gray code using K-Map is -

B3 = G3

B2 = G2 ⊕ G3

B1 = G1 ⊕ G2 ⊕ G3

B0 = G0 ⊕ G1 ⊕ G2 ⊕ G3

Procedure:

Set up gray to binary code converter using circuit given in figure no. 37. The circuit is using 3 EX-OR gate using IC 7486. Varify the binary code of respective gray code.

Scanned with CamScanner

Circuit for Gray to Binary

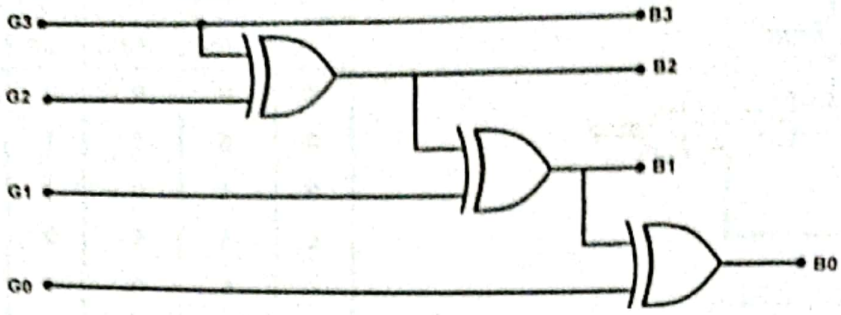


Fig.37

Experiment -5 Study and Verification of Truth Tables of Comparator.

Procedure

One Bit Comparator

1. Truth Tables of One Bit Comparator.

1- Bit Comparator Table No.27

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

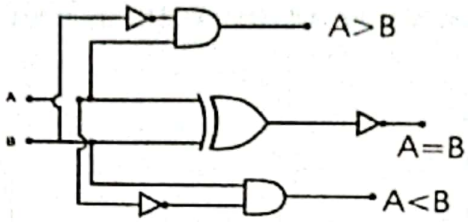


Fig.38

The simplified expression for single bit comparator

A>B ——— $A\bar{B}$
A=B ——— $AB+\bar{A}\bar{B}$
A<B ——— $\bar{A}B$

Procedure

Set up the circuit according to the fig.38 . The operation can be performed through IC's 7486 (EX-OR) gate , 7404 (NOT GATE) & 7408 (AND GATE)

4-Bit Comparator

4-bit comparator are available in MSI (7485) which can be compare straight binaryand natural BCD codes. these ICs ca be cascaded to compare words of greater lenghts without external gates. The A>B,A=B and A<B outputs of a stag handling less significant bits are connected to the corresponding A.B, A=B and a<B cascading inputs of the next stag handling more significant bits. The stage handling the least-significant bits must have A=B input connected to logic 1 lev and A>B and A<B inputs connected to logic 0 or 1 level.The pin description is shown in fig.40.

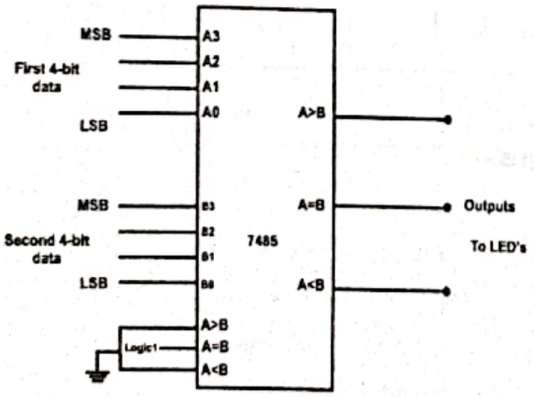


Fig.39

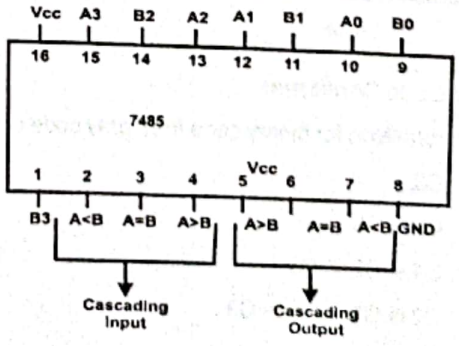


Fig.40

Scanned with CamScanner

Experiment - 6 Study of flip flops and verification of their truth tables.

Objective-: Verification of D-type flip-flop (7474).

Theory-: Delay flip flop or D type flip-flop is used to give delay of clock time to the input given signal. The D type flip-flop is used to divide the frequency of input square wave. Fig.41 shows a D type flip flop and truth table is given in Table No.28.

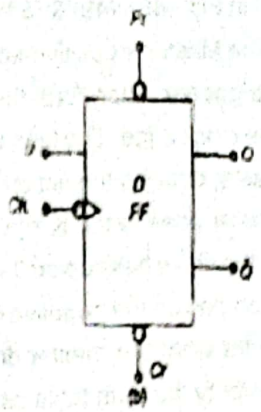


Fig. 41

Procedure

IC 7474 is a dual D-type positive edge triggered flip-flop. Pin configuration, circuit diagram and truth table are shown in Fig (45), (46) & table no.(30). D-type flip-flop has only one input referred to as D-input (Data Input).Make the circuit as shown in Fig. (46). Set Preset and Clear as giver in truth table, when clock is low, the flip-flop is latched in its last state. When clock is high the value od D-input is important. A high D-input sets the flip-flop, while a low D-input reset the flip-flop.Verify the truth table as given in table 30.

Objective -: Verification of J-K flip-flop (7476).

Truth table of a D-type FLIP-FLOP

Input D_n	*Output* Q_{n+1}
0	0
1	1

Procedure

IC 7476 is a dual J-K flip-flop with Preset & Clear. Pin configuration, circuit diagram and truth table are shown in Fig (47), (48) & table no.31. This one input is negative edge reset (O). Again set it by using the PR terminal.

In each case, observe Q also. Which is complement of Q. Verify that the presetting and clearing operations are independent of J & K inputs.

Basic Fig.

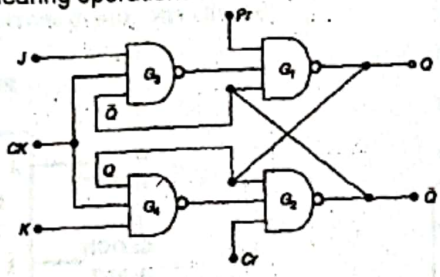


Fig.42

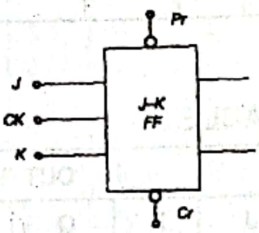


Fig.43

Inputs J_n	*Inputs* K_n	*Output* Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	$\overline{Q}_n$

Flip Flop Operation

Set PR = CR = 1. Apply the clock manually by using a bounce a elimination switch and observe the output for all possible combinations of the inputs.Verify the truth table given in Fig. (47)

Object :-Verification of JK Master/ Slave flip-flop (7472).

Procedure

For this flip-flop we are using IC 7472 JK Master/ Slave flip-flop with Preset and Clear. Pin configuration circuit diagram and truth table are

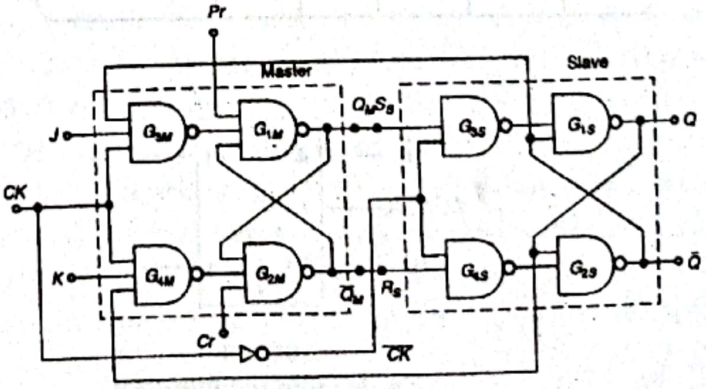


Fig.44

Scanned with CamScanner

shown in Fig. (48), (49) & (50).In JK Master/ Slave flip-flop two JK flip-flop are connected by invert clock as shown in Fi (44). The Master is positive-edge-triggered and the Slave is negative-
edge-triggered. Therefore, the master responds to its J and K control inputs before the slave. If J = 1, the master sets o positive clock edge. The high Q out of the master drives the J input of the slave. So, when the negative clock edge hits, th slave sets, copying the action of the master.If K =1, the master resets on the leading edge of the clock. The high Q out the master goes to the K input of the salve. Therefore, the arrival of the clock's trailling edge forces the slave to res Again, the slave has copied the master.If the master's J and K inputs are high, it toggles on the positive clock edge and th slave toggles on the negative clock edge.

No matter what the master does, the slave copies it. If the master sets, the slave sets, if the master resets, the sla resets.Verify the truth table as given in table (32

TRUTH TABLE 29

INPUTS				OUTPUTS	
PRE SET	CLE AR	CLO CK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	I	H	H	L
H	H	I	L	L	H
H	H	L	X	Qo	$\overline{Qo}$

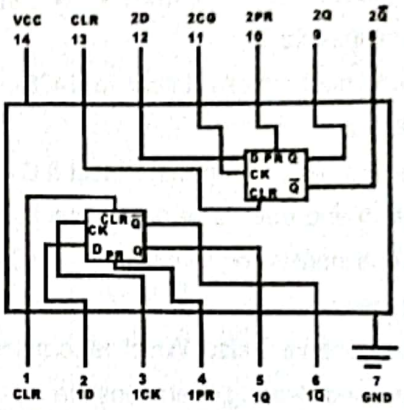


Fig.45 - PIN CONFIGURATION IC 7474

TRUTH TABLE 30

INPUTS				OUTPUTS		
PRE SET	CLE AR	CLO CK	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	Ω	L	L	Qo	$\overline{Qo}$
H	H	Ω	H	L	H	L
H	H	Ω	L	H	L	H
H	H	Ω	H	H	TOGGLE	

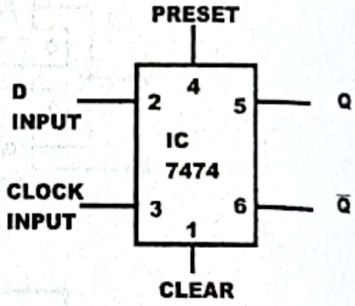


Fig. 46 - CIRCUIT DIAGRAM

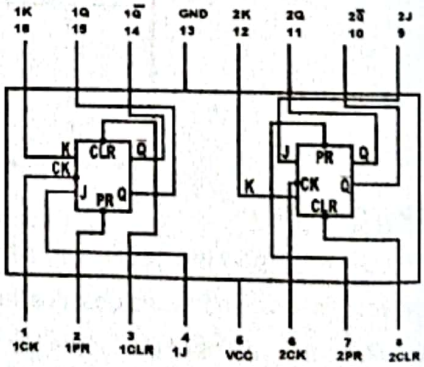


Fig.47 - PIN CONFIGURATION IC 7476

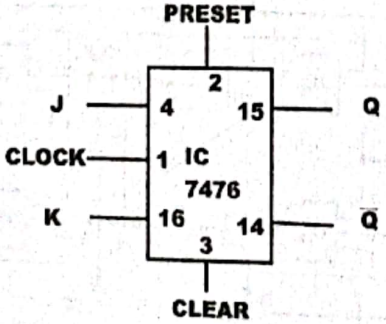


Fig. 48 - CIRCUIT DIAGRAM

Scanned with CamScanner

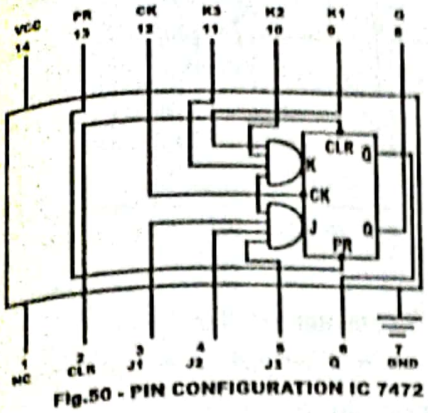
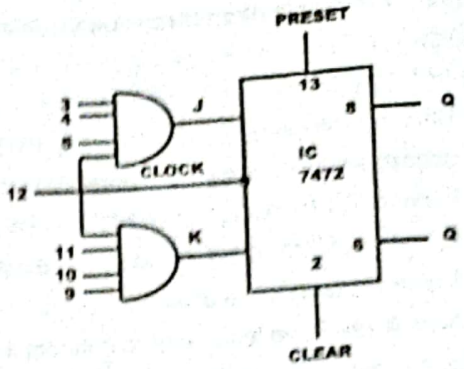


Fig.50 - PIN CONFIGURATION IC 7472



POSITIVE LOGIC: J=J1.J2.J3
K=K1.K2.K3

Fig.50 - CIRCUIT DIAGRAM.

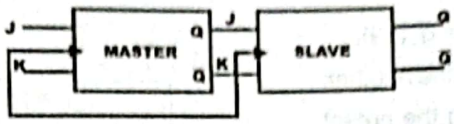


Fig. 51 - MASTER/SLAVE FLIP-FLOP

TRUTH TABLE -31

INPUTS					OUTPUTS	
PRESET	CLEAR	SCLOCK	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	W	L	L	Qo	$\overline{Qo}$
H	H	W	H	L	H	L
H	H	W	L	H	L	H
H	H	W	H	H	TOGGLE	

Experiment 7 Study of Counters & Shift Registers and verification of their truth tables.

Theory

A counter driven by a clock can be used to count the no. of clock pulses. There are basically 2 different types of counters.

1. Synchronous Counter (Parallel Counter)
2. Asynchronous Counter (Ripple Counter)

Here we have used ripple counter which is simple and straight forward in operation & construction and usually requires a minimum of hardware. Here each flip-flop is triggered by the previous flip-flop, which is why it is also called as a serial counter.For the 4 bit ripple counter, we have used two 7476 IC's each comprises of two flip-flops.For up counting, counter counts the no. of clock transition up to a maximum of 15. Clock pulses are applied at the clock input of first flip-flop & output of first flip-flop; i.e. QA is used to drive flip-flop B & QB is used to drive flip-flop C and so on. The counter begins at count 0000 & advances one count for each clock transition until it reaches count 1111. At this point, it resets back to 0000 & begains the count cycle all over again.For down counting, clock pulses are applied at the clock input of first flip-flop & compliment of QA is used to drive flip-flop B & QB is used to drive flip-flop C & so on. The counter begins at count 1111 & counter contents are reduced by one count with each clock transition until it counts 0000. At this point, it resets back to 1111 & begins the counts cycle all over again.

Scanned with CamScanner

Objective- 4 Bit Binary Asynchronous Counters

Procedure

A. Forward Counter

1. Connect the circuit as shown in Fig. (52) by connecting clock output to ck input of first flip flop, connect Q1 to ck2,Q2 to ck3,Q3 to ck4. Also connect Q1, Q2, Q3 & Q4 to LED indicators. Connect reset points of all the flip flops & keep one lead open to connect it to ground point to reset the counter to 0000.

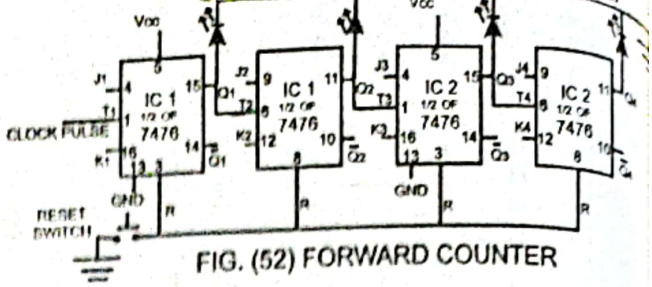


FIG. (52) FORWARD COUNTER

Conclusion:- counter counts the pulses up to decimel 15 & then on the application of next pulse, resets to zero automatic
The binary counter advances one bit each time at the trailing edge of the clock pulse.

B. Reverse Counter :-

1. Connect the circuit as shown in Fig. (53).
2. Connect preset of all flip-flops to one another.
3. Reset the flip-flop by connecting the preset input i.e. PR to the ground momentarily. Observe that output at all the indicators are low (i.e. all the LED's are in OFF condi tions).
4. Apply clock pulses one by one using pulser switch. Note all the four output's at each pulse & verify it with the table no. 33.
5. Draw a graph of the output wave form with respect to the input pulses.

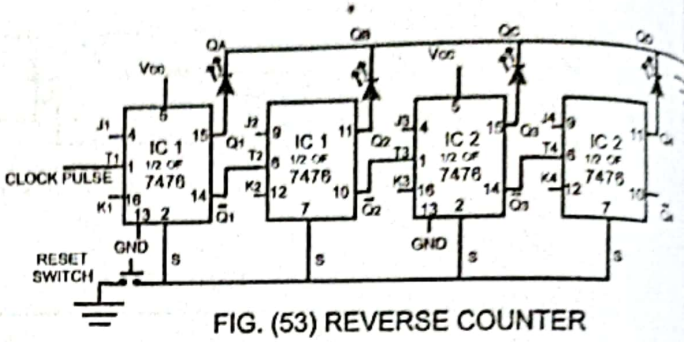


FIG. (53) REVERSE COUNTER

TRUTH TABLE - 32

INPUT	OUTPUT			
Clock	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

TRUTH TABLE - 33

INPUT	OUTPUT			
Clock	QD	QC	QB	QA
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
10	0	1	0	1
11	0	1	0	0
12	0	0	1	1
13	0	0	1	0
14	0	0	0	1
15	0	0	0	0

C. Module Counter

Theory

4 bit ripple counter counts upto 15 & then resets to '0'. But it can be made to reset at any clock pulse. For this purpos we operate the reset connections automatically using 4-input NAND GATE. As you know that output of a NAND GATE low if & only if all the inputs are high. Suppose we want to convert 4-bit forward counter to Decade counter. For Decad

Scanned with CamScanner

counter it is required that counter should count upto 9 & than resets to zero at tenth clock pulse. We know that at 10th clock pulse output of 2nd & 4th flip flop is high& output of 1st & 3rd flip flop is low. So we choose Q output of 2nd & 4th flip flop & Q output of 1st & 3rd flip flop for the inputs of 4-input NAND GATE. Also connect output of NAND GATE to common reset of all the flip flops. As you can program the counter for any number of Pulses.

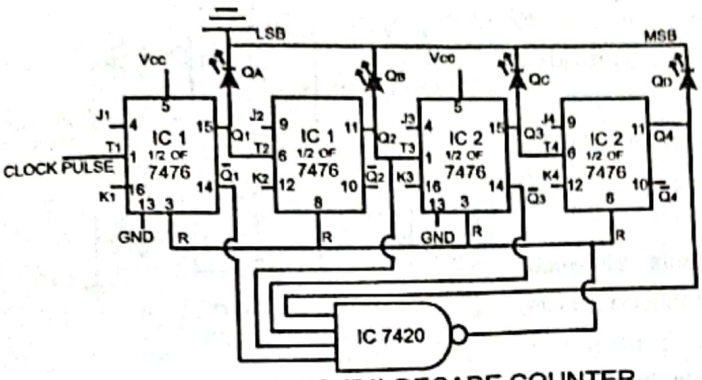


FIG (54) DECADE COUNTER

TRUTH TABLE - 34

INPUT	OUTPUT			
Clock	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

).

Excess 3 Counter

This is another form 8 BCD code, in which each decimal digit 13 coded into a 4 bit binary code. The code for each decimal digit 13 obtained by adding decimal 3 to the natural BCD code of the digit . For example decimal 5 is coded as 0101 + 0011=1000 in excess 3 code.

An excess 3 counter can be made by converting the updown counter to its excess- 3code.

For Ex. using IC's 74193 (Asynchronous up down counter) output up counter add binary three in the 4 bit adder

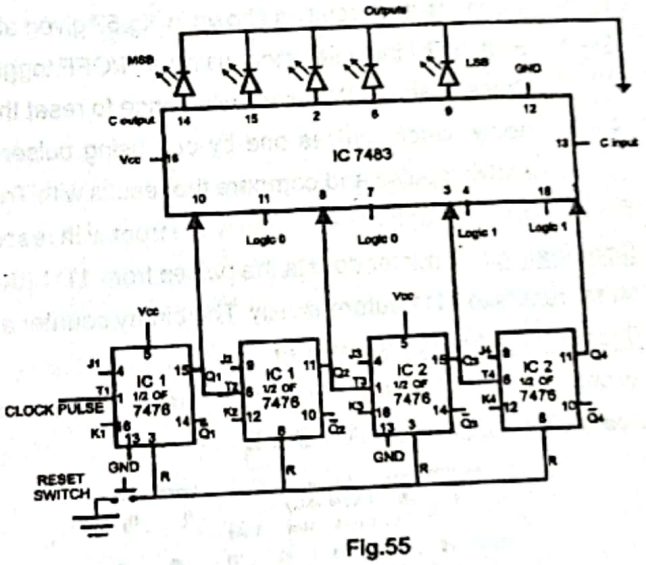


Fig.55

Objective- 4 Bit up Down Binary Synchronous Counter

Theory

A counter is sequential logic circuit made up of flip flops and is used to count the number of pulses apply to it. The input pulse change the status of the flip flops in such a way that by observing the output levels, the total number of input pulses applied can be determined. The ripple counter is simplest to build, but there is a limit to its highest operating frequency. Each flip flop has a delay time. In a ripple counter these delay times are additive and the total settling time for the counter is approximately delay time times the total number of flip flops. This speed limitation can be overcome by the use of a synchronous or parallel counter. The difference here is every flip flop is triggered by the clock. If the asynchronous operation of a counter is changed so that all flip flops are clocked simultaneously (synchronously) by the input pulses, the propagation delay time may be reduced considerably. Repetition rate is limit by the delay of anyone flip flop plus the propagation time of any control GATE required.

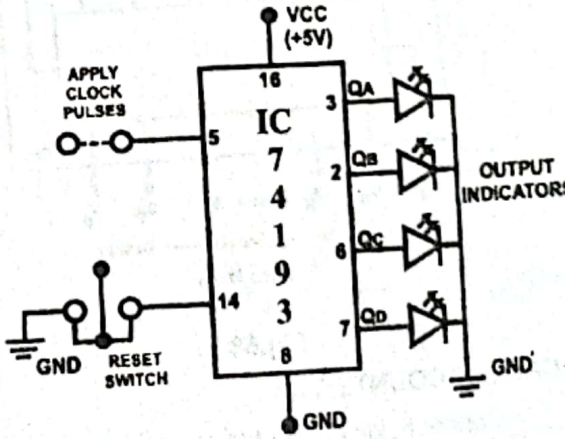


FIG. (56)

Scanned with CamScanner

Typically, the maximum frequency of operation of a 4-bit synchronous counter using TTL logic is 32MHz, which is abou twice that of a ripple counter, another advantage of the synchronous counter is that no decoding spikes appear at the output since all flip flops change state at the same time. Hence the strobe pulse is required when decoding a synchronous counter.

Procedure

Up (Forward) Counter

1. Connect the circuit as shown in Fig.56.
2. Switch ON the instrument using ON/OFF toggle switch provided on front panel.
3. Press push to OFF reset switch once to reset the output at logic 0000.
4. Apply clock pulses one by one using pulser switch. Note down the status of all the four outputs at each applying pulse and compare the results with truth table no.(1).[as in case of asynchronous up counter]
5. Draw a graph of the output wave from with respect to the input pulses.

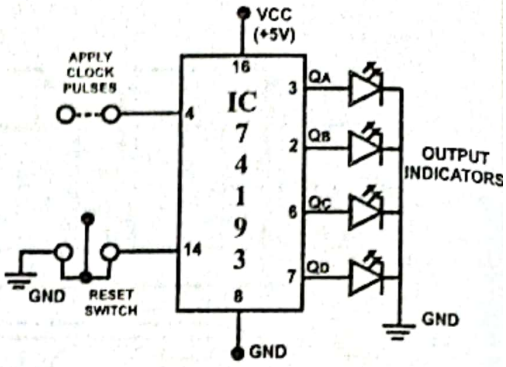


FIG. (57)

Conclusion :- Counter counts the pulses from 0000(1) upto decimal 15 and then the application of next pulse, resets to zero automatically. The binary counter advances one bit each time at the tariling edge of the clock pulse.

Down (Reverse) Counter :-

1. Connect the circuit as shown in fig.57 given above.
2. Switch ON the instrument using ON/OFF toggle switch provided on front panel.
3. Press push to ON reset switch once to reset the output of logic 0000.
4. Apply clock pulses one by one using pulser switch. Note down the status of all the four outputs all each applying pulse and compare the results with Truth Table no.(2). [as in case of asynchronous down counter].
5. Draw a graph of the output wave from with respect to the input pulses.

Conclusion :- Counter counts the pulses from 1111 (decimal 15) upto 0000 (decimal 0) and then the application of next pulse, resets to 1111 automatically. The binary counter advances one bit each time at the tariling edge of the clock pulse.

Decade Synchronous Counter

All counters in 74190 are fully programmable .The counter can be preset to any desired BCD digit as determined by the states of the data inputs A,B,C & D.

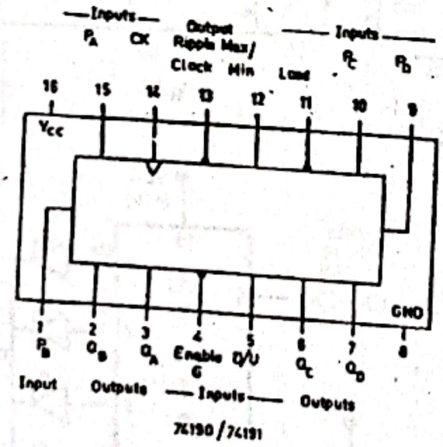


Fig.58

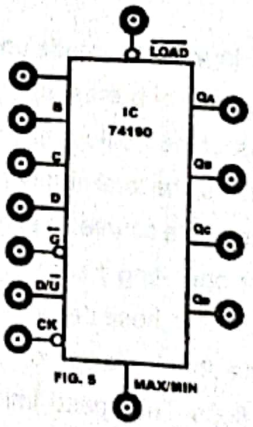


Fig.59

DECADE UP COUNTER

1. Connect clock output (1 Hz) to CK input of the IC 74190 through single point patch cords.
2. Connect LOAD to logic input '0' (ground point) and also D/U pins to ground point . Apply logic 0 on G pin of IC .
3. Connect 4 logic outputs to output indicators.
5. Apply logic 1 on LOAD pin of IC.

Scanned with CamScanner

6. Apply clock pulses one by one using pulser switch . Note down all the four outputs at the application of each pulse & verify the truth table.

Conclusion: Counter counts the pulses up to 9 i.e. from 0000 to 1001and then on the application of next pulse, reset to 0000 automatically.

Decade Down Counter

7. Apply logic 1 to D/U pin of IC
8. Apply clock pulses one by one using pulser swithc. Note down all the four outputs at the application of each pulse and verify that the counter counts from 1001 to 0000 and then on application of next pulse, reset to 0000 automatically.

Objective- Study of SHIFT Register

Serial to Parallel and Parallelto Serial SHIFT Register using IC 7495

Theory

A SHIFT register is basically a storage medium where one or more binary words may be stored. Like a counter, it is also made up of binary storage elements, usually flip-flops. These elements are cascaded in such a way that the bit stored there can be moved or shifted from one element to another adjacent element. All of the storage registers are activated simultaneously by a single input 'CLOCK' or 'SHIFT' pulse. When a shift pulse is applied, the data stored in the shift register is moved one position either to left or to right as desired. Because of its ability to move the data, one bit at a time, the SHIFT Register is valuable in performing a wide variety of logic operations like counting, frequency dividing or perforiming airthmatic operations.

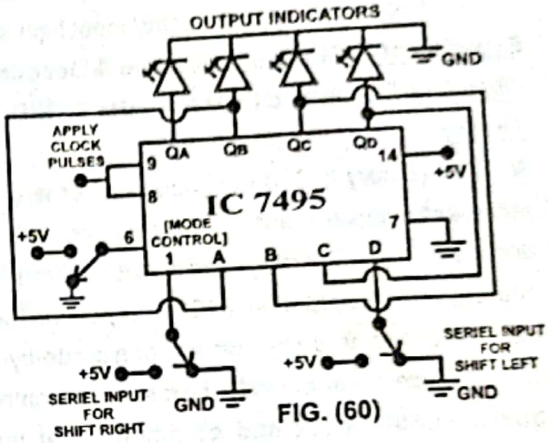


FIG. (60)

The storage capacity of a SHIFT Register depends on the number of binary elements used. Most SHIFT Register operations are serial operations, but many circuits are provided with parallel inputs and parallel outputs such SHIFT Registers permit data to be preset in parallel and data to be readout in parallel. This property makes the SHIFT Register an ideal circuit for performing serial to parallel and parallel to serial conversions. SHIFT Registers can also be used to perform arithmatic operations. Shifting the data stored in a SHIFT Register, to the right or to the left by one bit is equivalent to multiplying or dividing that number by two. In addition, SHIFT Registers can also be used for generating a sequence of control pulses for a logic circuit. In some applications, SHIFT Registers are used for counting and frequency division.

Procedure for Serial to Parallel SHIFT Register

1. Connect the circuit as shown in Fig. (60).
2. Switch ON the instrument using ON/ OFF toggle switch provided on front panel.
3. Put the mode control switch to ground i.e. 0 volt to enable the IC perform right SHIFT operation.
4. Put serial input and D input to ground. Apply four clock pulses and record the final output state as below ABCD=_____
5. Put serial input to +5V. Apply four clock pulses again and record the final outputs in the same manner. Note the direction in which the data shifts.
6. Set serial input to ground. Apply four clock pulses and note the direction in which the data shifts.
7. Put mode control switch to +5V. Also put the D input to +5V. Apply SHIFT pulses and note down the direction of shifting. Record the final state after four pulses.

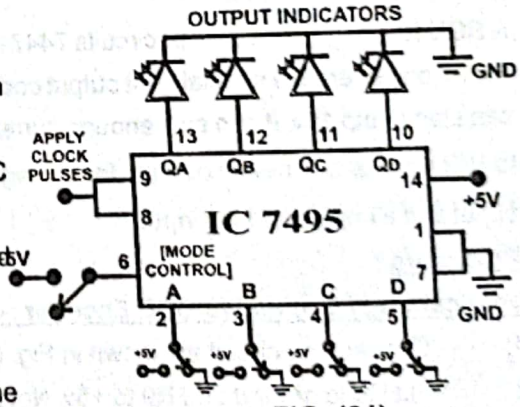


FIG. (61)

Scanned with CamScanner

8. Set D input to binary 0 i.e. to ground. Apply two shift pulses and record the contents of the register in the same manner.

Procedure for Parallel to Serial SHIFT Register

1. Connect the circuit as shown in Fig. (61)
2. Switch ON the instrument using ON-OFF toggle switch provided on front panel.
3. Set Data inputs ABC and D to ground i.e. at 0 Volt.
4. Put mode control switch to +5V and apply one clock pulse. Note output levels and record the observations.
5. Set the Data inputs A B C and D to +5 V. Apply one clock pulse and record the observations in the same manner.
6. Put mode control input to ground. Apply four clock pulses one by one and note the outputs each time.

Note :- Any parallel data from 0000 to 1111 at inputs A B C and D can be loaded into the SHIFT Register keeping the mode control input high and applying a clock pulse.

Experiment -8 Study of Encoders & Decoders and verification of their truth tables.

Objective-Decimal to BCD Encoder & BCD TO 7-Segment Decoder

Theory

A BCD (Binary coded decimal) counter is a sequential circuit that counts by tens. It has ten discrete states which represent decimal numbers from 0 through 9. Because of its ten state nature, A BCD counter is also referred to as a decade counter. The integrated circuit 7490 is a BCD counter using the standard 8421 binary code. The IC consists of one divide by -two and one divide-by-five circuit, which are independent except for a common power supply input and common reset input terminals.In addition to counting circuits, the 7490 also contains two numbers of 2-input NAND GATES. A low input to at least one input of each GATE is necessary to enable counting. If both inputs to GATES G1 and G2 are high, the counter is reset to binary 0000. These inputs are marked as Ro(1) and Ro (2). On the other hand, if both inputs to gates G3 and G4 go high, the counter is reset to binary 1001 (decimal 9) . These inputs are marked as R9 (1) and R2.

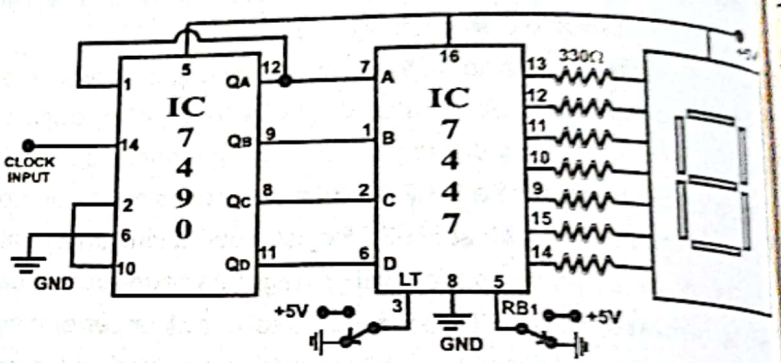


FIG. (62)

A BCD to 7-segment decoder circuits 7447 is a special form of decoder circuit that accept the standard 8421 BCD input code and generate a special 7-bit output code that is used to operate a 7-segment readout.The output transistors in 7447 can stand upto 15v. It can sink enough current to drive common anode type LED 7-segment displays directly. In addition to BCD inputs and seven outputs for driving segments, the IC have a lamp test input , a blanking input/ripple blanking ouput and a ripple blanking input.

Procedure :

Verification of Decimal to BCD Encoder :-

1. Connect the circuit as shown in Fig. (62).
2. Put Ro to ground and R9 to +5v. Note the output levels.
3. Put R0 to +5v and R9 to ground. Note the output levels.
4. Put both R0 and R9 to ground. Apply clock pulses one by one and note the output levels when the clock pulse goes high and when it goes low.
5. Repeat step 4 above for about 20 clock pulse.
6. Tabulate your observations as shown in truth table.
7. Draw output waveforms with reference to the clock pulse

Scanned with CamScanner

Verification of BCD to decimal decoder

1. Connect the circuit as shown in Fig. (62).
2. Switch ON the instrument using ON/ OFF toggle switch provided on front panel. The seven segment display shows any arbitrary Fig.
3. Connect lamp-test (LT) of IC 7447 to ground. All LED segments should light up showing a Fig. 8.
4. Make lamp test input high. Apply clock pulses one by one and see that the figures displayed by the LED repeat sequentially from 0 through 9.
5. Apply clock pulses till figures zero is displayed. Now connect input marked RB1 to ground. The display should disappear.
6. With pin 5 connected to ground, apply clock pulses and confirm that all figures except 0 are displayed sequentially.

TRUTH TABLE - 35

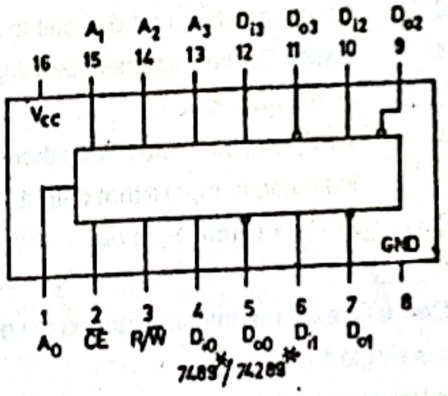
CLOCK PULSE	BCD OUTPUTS D	C	B	A	7 SEGMENT OUTPUT
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9

Truth Table No. (19) for Decimal to BCD Encoder & BCD to 7 Segment Decoder

Objective- Study of Memory Register

Theory

IC 7489 is a 16 x 4 RAM in which 16 words of 4 bits length can be read or written. Memory is a device used for the storage of digital data. Earlier, only magnetic memory devices were possible, but now a days it has become possible to make memory devices using semiconductors. Semiconductor memories have become very popular because of their smaller size (available in the IC form) Flip- Flops and other digital systems such as shift -registers can be used for the storage of digital data. In shift registers, the data can be stored for any desired length of time and then read out in a serial or parallel form. In this form of memory any location for storing (Writing) or reading the data can be accessed only sequentially. Hence, this form of memory is very slow and also has limited storage capacity. Therefore, this type of arrangement is not suitable for storing large data.Another form of memory in which any bit can be accessed in a random fashion, requiring the same time for each location, is known as the random access memory. **RAM** is a read and write memory, i.e. data can be written into it or read from it. In IC 7489 the outputs are open collector and active low. CE is the 'Chip Enable' terminal and when this is at logic '0', the memory is enabled. To write into any memory location apply the address of the chosen memory location at the address input terminals, apply the data to be stored at the data input terminals, apply logic '0' at the Read/ Write (R/ W) line. To read from any memory location at the address input terminals, apply logic1 at the R/W lin



Procedure

1. Connect logic inputs '0' & '1' selectable through SPDT switches to ADDRESS inputs as well as to DATA inputs through patchchords. Also connect CE (Chip Enable) & R/ W pin to logic inputs through patchchords.
2. Connect Four logic outputs (D01, D02, D03 & D04) to four logic output indicators (LED) through patchchords.
3. Switch ON the instrument using ON/ OFF toggle switch provided on the front panel.
4. To write into any memory location, apply logic input '1' at CE as well as at R/ W pins as given in SR. No. 1 of Functional Table. Apply the ADDRESS of the choosen memory location at the ADDRESS input pins (0000 - 1111) and apply the DATA (0011) to be stored at the DATA input pins through SPDT switches given on the trainer board.

Scanned with CamScanner

5. To read the above stored DATA on output indicators proceed as SR. No. 2 of Functional Table-16, change P/W input from logic '1' to '0'. We will observe that output indicators also shows output 0011.
6. Change the DATA from 0001 - 1110 and repeat the steps 4 & 5 of procedure.

Experiment -9 Study of Multiplexers, Demultiplexers and verification of their truth tables.

Objective- 4 to 1 LINE MULTIPLEXER using IC 74153 :

Theory

Multiplexer means many into one i,e multiplexer is a logic circuit which has many inputs but single output. A multiplexer accepts several data inputs but allow only one of them at a time to get through to the output. The block diagram of a multiplexer is shown in Fig.63 In this diagram the inputs and outputs are indicated by means of broad arrows to indicate that there may be one or more inputs Depending upon the digital code applied at the select inputs, one out of the N data source (D_0, D_1,———D_{n-1}) is selected and transmitted to the single output channel. A 4 to 1 line multiplexer has four inputs but only single output.

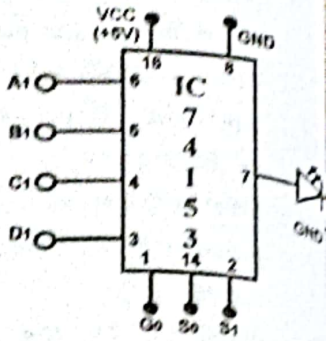


FIG. (63)

To perform 4 to 1 line multiplexer experiment, we have used IC 74153. It has 4-line inputs (A1, B1, C1, D1) and only one outputs Y1. G_0 is the strobe input (active low). S1, and So are select lines these lines select one out of four inputs at output for e.g. if we will apply oo at S1 & S0 first input A1 will be selected.

TRUTH TABLE - 36

Sr. No.	INPUT			OUTPUT
	STROBE	SELECT INPUT		
	G0	S1	S0	Y
1	0	0	0	A1
2	0	0	1	B1
3	0	1	0	C1
4	0	1	1	D1

Procedure

1. Connect the circuit as shown in Fig. (63) according to S.No. 1 of Truth Table No. (36).
2. Connect output of the circuit to output indicator.
3. Switch On the instrument by using ON/OFF switch provided on the front panel.
4. Verify the truth table for other sets of input and observe the output indicator, compare that output with truth table.

Objective -1 :4 Demultiplexer

Theory

Demultiplexer means one into many i,e Demultiplexer is a logic circuit which has single input

but many output. It accepts a single input & distributes it over several outputs.The block diagram of a fig.

Demultiplexer is shown in fig.64.

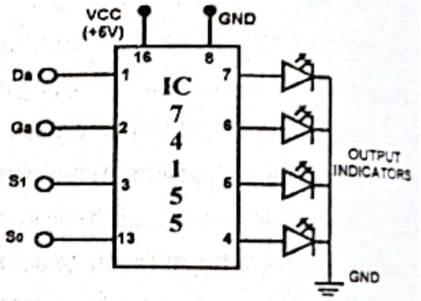


FIG. (64)

The select input code determines to which output the data input will be transmitted. A 1 to 4 line Demultiplexer has 1 input & 4 outputs.

To perform 1 to 4 line Demultiplexer experiment, we have used IC 74155. It has 2 data select line inputs (S0 & S1) and one strobe input Ga (active low).

TRUTH TABLE - 37

Sr. No.	INPUT			OUTPUT
	STROBE	SELECT INPUT		
	Ga	S1	S0	Y
1	0	0	0	Y0
2	0	0	1	Y1
3	0	1	0	Y2
4	0	1	1	Y3

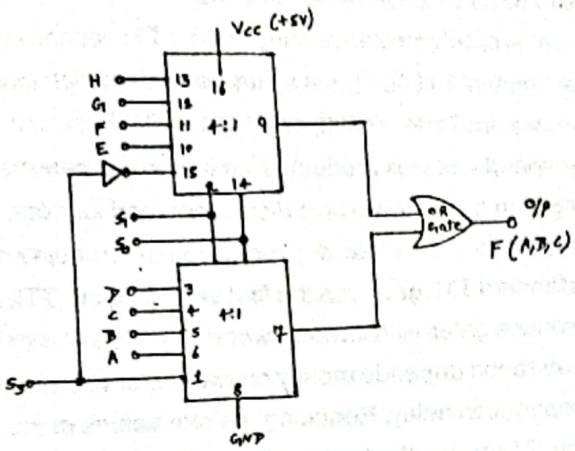
Procedure

1. Connect the circuit according to S.No. 1 of truth table given below:-
2. Switch On the instrument by using ON/OFF switch provided on the front pannel.
3. Apply data inputs at data input Ga & Also connect Ga to "0" level.s
4. Verify the truth table for other sets of input and observe the output indicator, compare that output with truth table.

Scanned with CamScanner

Objective: 8:1 Multiplexer

To perform 8:1 multiplexer experiment we have use the same IC 74153. Since IC74153 has two 4:1 multiplexer wer can perform the 8:1 operation by making a multiplexer tree of 4:1 multiplexers.



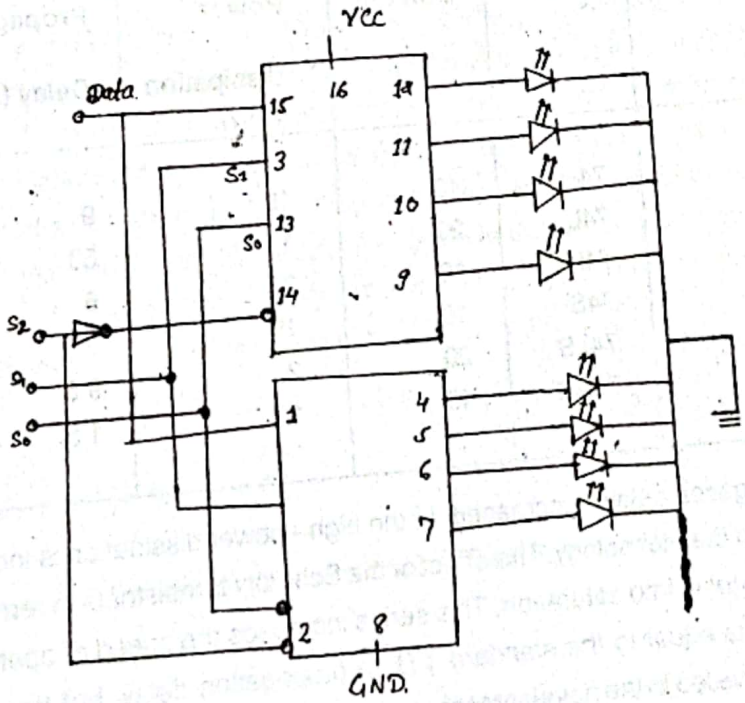
TABLE

SR NO.	S_2	S_1	S_0	O/P Y
0	0	0	0	A
1	0	0	1	B
2	0	1	0	C
3	0	1	1	D
4	1	0	0	E
5	1	0	1	F
6	1	1	0	G
7	1	1	1	H

Objective: 1:8 Demultiplexer

Similarly 1:8 demultiplexer operation can be performed by making a demultiplexer tree of two 1:4 demultiplexers .

1:8 Demultiplexer



Scanned with CamScanner

Experiment 10 :

To Conduct CMOS to TTL IC and TTL to CMOS IC Interfacing Experiment using pull up resistors and transistors.

Theory

Transistor Transistor Logic (TTL)

The original basic TTL gate was a slight improvement over the DTL gate. As the TTL technology progressed. Additional improvements were added to the point where this. Logic family became the most widely used family in the design of digital systems. There are several subfamilies or series of the TTL technology. The names and characteristics of seven TTL series appear in table 1 commercial TTL ICs have a number designation that starts with 74 and follows with a suffix that identifies the series type. Examples are 7404, 74s86, and 74ALS161. Fan - out, power dissipation and propagation delay were defined in section 2. The speed - power product is an important parameter for comparing the various TTL series. This is the product of the propagation delay and power dissipation and is measured in picojoules (pJ). A low value for this parameter is desirable, because it indicates that a given propagation delay can be achieved without excessive power dissipation, and vice versa.The standard TTL gate was the first version in the TTL family. This basic gate was then designed with different resistor values to produce gates with lower power dissipation or with higher speed. The propagation delay of a transistor circuit that goes into saturation depends mostly on two factors : storage time and *RC* time constants. Reducing the storage time decrease the propagation delay. Reducing resistor values in the circuit reduces the RC time constants and decreases the propagation delay. Of course, the trade - off is higher power dissipation because lower resistances draw more current

Table- 1

TTL Series and Their Characteristics

From the power supply. The speed of the gate is inversely proportional to the propagation delay.

In the low - power TTL gate, the resistor values are higher than in the standard gate to reduce the

TTL Series Name Power	Prefix	Fan out	Power Dissipation (mW)	Propagation Delay (ns)	Speed Product (pJ)
Standard	74	10	10	9	90
Low Power	74L	20	1	33	33
High Speed	74H	10	22	6	132
Schottky	74S	10	19	3	57
Low Power Schottky	74LS	20	2	9.5	19
Advanced Schottky	74AS	40	10	1.5	15

power dissipation, but the propagation delay is increased. In the high - power dissipation is increased. The Schottky TTL gate was the next improvement in the technology. The effect of the Schottky transistor is to remove the storage time delay by preventing the transistor from going into saturation. This series increases the speed of operation without an excessive increase in power dissipation. It is equal to the standard TTL in propagation delay, but has only one - fifth the power dissipation. Recent innovations have led to the development of the advanced schottky series. It provides an improvement in propagation delay over the schottky has the lowest -

power dissipation. The advanced low - power schottky has the lowest - speed - power product and is the most efficient series. It is replacing all other low - power versions in new designs.

All TTL series are available an available in SSI and in more complex forms as MSI and LSI components.

The differences in the TTL series are not in the digital logic that they perform, but rather in the internal

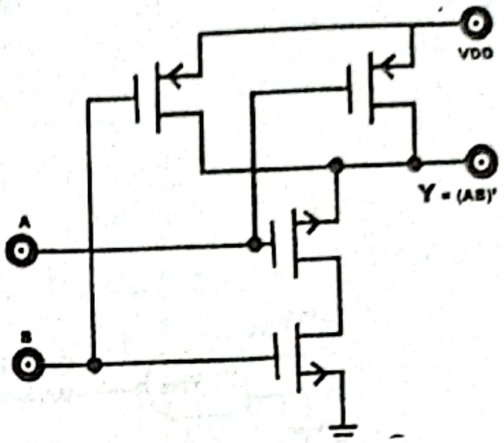
Scanned with CamScanner

construction of the basic NAND gate. In any case, TTL gates in all available series come in three different types of output configuration

1. Open - collector output
2. Totem - pole output
3. Three - sate (or tristate) output

These three types of outputs will be considered in conjunction with the circuit description of the basic TTL gate.

Complementary MOS circuits take advantage of the fact that both n - channel and p - channel devices can be fabricated on the same substrate. CMOS circuits consist of both types of MOS devices interconnected to from logic functions. The basic circuit is the inverter, which consists of one p - channel transistor and one n - channel transistor, as shown in fig .1(a). The source terminal of the p - channel device is at ground. The value of V_{DD} may be anywhere from +3 to +18V. The two voltage levels are 0V for the low level and V_{DD} for the high level.



To understand the operation of the inverter, we must review the behaviour of the MOS trasnsistor from the previous section.

The n - channel MOS conducts when its gate - to source voltage is positive.The p - channel MOS conducts MOS conducts when its gate to source voltage is negative . Either type of device if turned off if its gate- to- source voltage is zero .A two input NAND gate consists of two p-type units in parallel and two n-type units in series ,as shown in fig.1. If all inputs are hihg, both p- channel transistors turn off and both n-channel transistor turn ON. The output has a low imedance to ground and produces a low stat. If any input is low ,the associated n-channel transistor is turnned off and the associated p-channel transistor is turned off and the associated p-channel transistor is turned ON. The out put is coupled to VDD and goes to the high state. Multiple input NAND gate may be formed by placing equal number of p-type numbers of p-type and n-type transistor in parallel and series ,respectively, in an arrangement similar to that shown in fig.when its gate - to - source voltage is negative.Either type of device is turned off if its gate - to - source voltage is zero.A two - input NAND gate consists of two p - type units in parallel and two n - type units in series, as shown in fig.1. If all inputs are high, both p - channel transistors turn off and both n - channel transistors turn on. The output has a low impedance to ground and produces a low state. If any input is low, the associated n - channel transistor is turned off and the associated p - channel transistor is turned on. The output is coupled to VDD and goes to the high state. Multiple input NAND gates may be formed by placing equal number of p - type numbers of p - type and n - type transistors in parallel and series, respectively, in an arrangement similar to that shown in fig.

Procedure

TTL to CMOS

1. Arrange the circuit according to the fig.61.
2. Give the supply of +5V & +12V from the main board.
3. Connect the output of TTL NAND gate (i.e) pin no. 3 of 7400 to the input of CMOS NAND gate (i.e) pin no. 1 of IC 74C00.
4. The output will be arround 2.75V.
5. If the CMOS is connected through interface network the output is high of about 8.5 to 9.5V.

CMOS to TTL

1. Arrange the circuit according to the fig.64.
2. Connect the output of TTL NAND gate (i.e) pin no. 3 of 74C00 to the pin no. 1 of IC 7400 without interface network.
3. The output will be zero.
4. connect the CMOS output (i.e) pin no. 3 of IC 74C00 via interface network as shown in diagram.
5. The output will be high.

Scanned with CamScanner

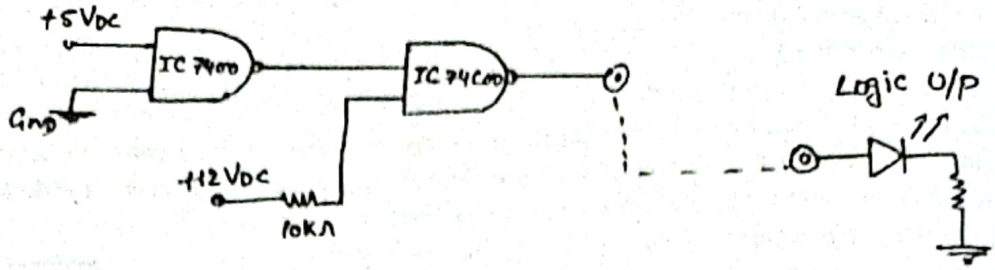


Fig.67

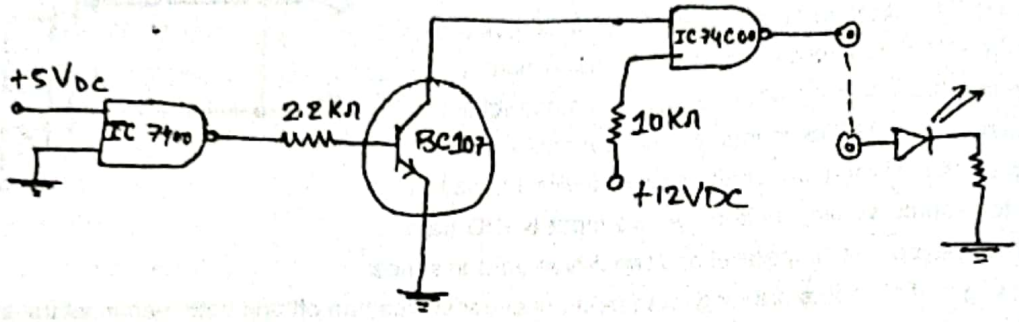


Fig.68

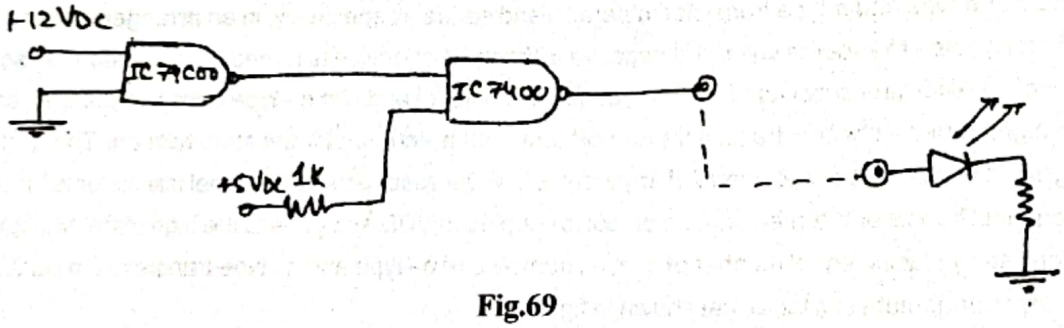


Fig.69

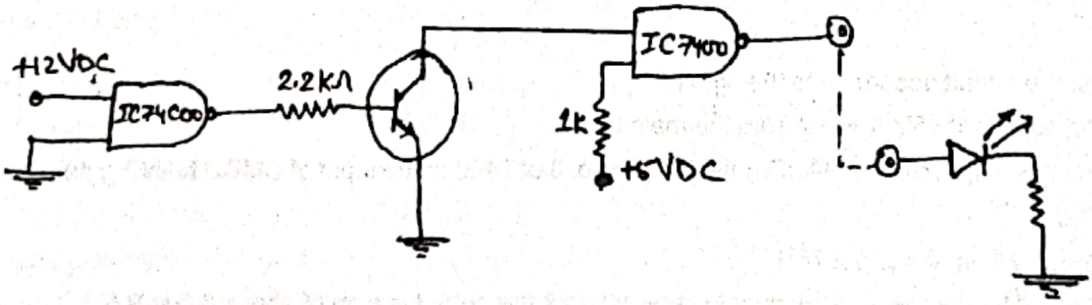


Fig.70

Scanned with CamScanner

PIN CONFIGURATION OF LOGIC GATE IC's

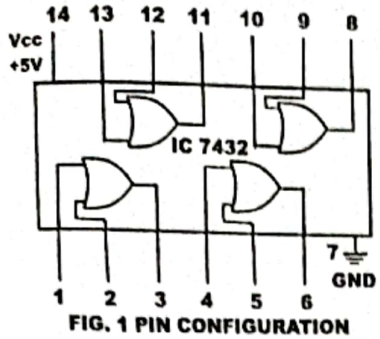


FIG. 1 PIN CONFIGURATION

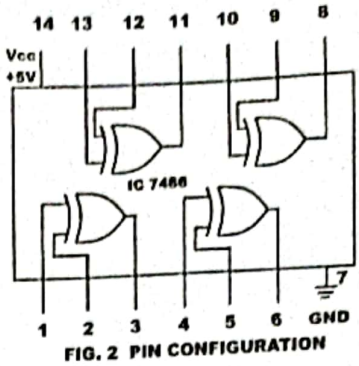


FIG. 2 PIN CONFIGURATION

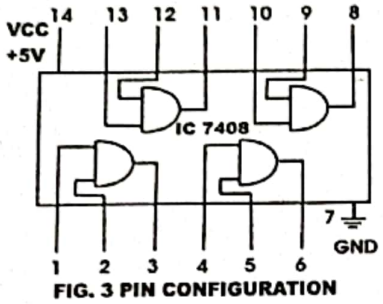


FIG. 3 PIN CONFIGURATION

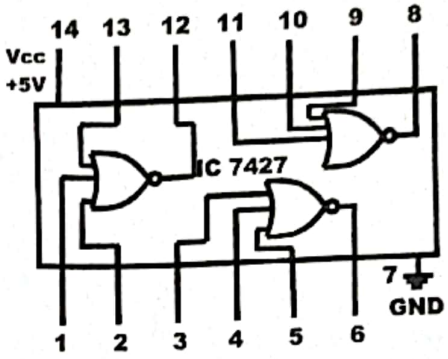


FIG. 4 PIN CONFIGURATION.

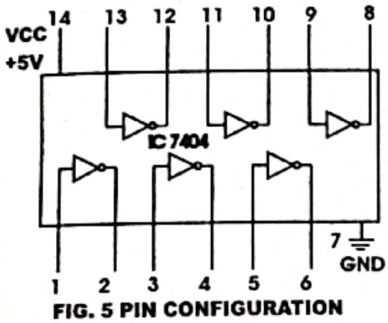


FIG. 5 PIN CONFIGURATION

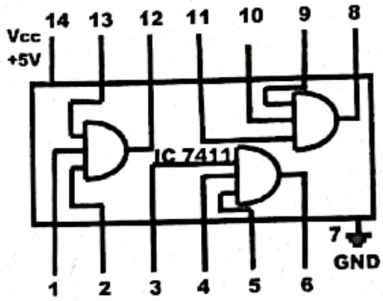


FIG. 6 PIN CONFIGURATION.

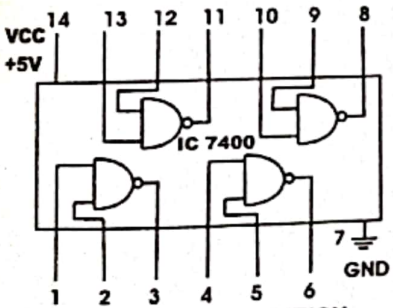


FIG. 7 PIN CONFIGURATION

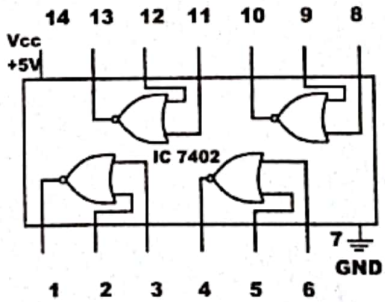


FIG. 8 PIN CONFIGURATION

Scanned with CamScanner