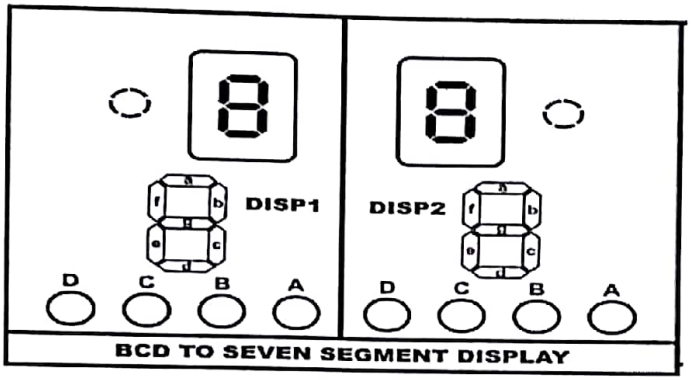
HARDWARE DESCRIPTION

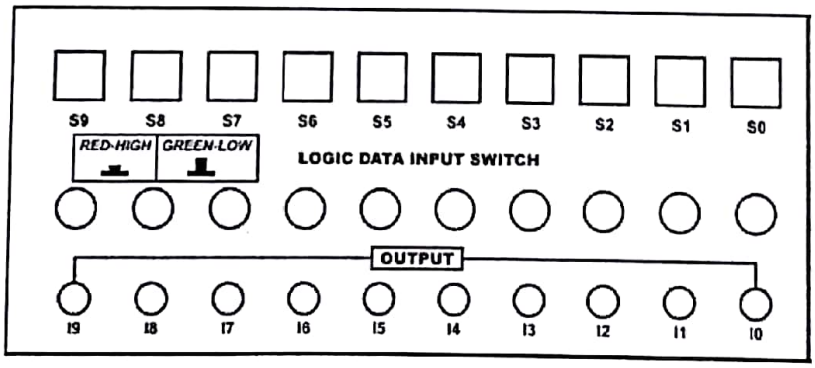
BCD To Seven Segment Display

The Digital / Analog Trainer provide Two Digit BCD to Seven Segment displays on the board so that experiments involving displays can be conducted. The four BCD to seven segment display can be used as BCD inputs



TTL/CMOS Logic Levels Input

The UADW-01 Trainer has 10 Push Key Switches S0 to S9 to generate ten TTL/CMOS input named as I0 to I9 as marked on the Trainer Board. When the Switch is in Normal (Unpressed) Mode, Logic Level Low will generate & when the switch is in Pressed Mode, Logic Level High will be generated on the 2mm banana socket provided on the kit. 10 Dual Color LEDs are used to indicate the logic input generated by each Push Key switch. The logic HIGH is indicated by the corresponding LED glowing as RED where the Logic LOW is indicated by the LED glowing as GREEN.



Logic Level Output Indicator

This Trainer has 10 Logic Level LED indicators for indicating output. The Logic high is indicated by LED glowing RED, logic low by GREEN, Pulse by YELLOW & OFF for High impedance. Logic level output is given in 2mm banana socket provided on-board.

Bread Board Area

This Trainer provides a bread board area consisting of two Terminal Strip of 630 Tie points each totaling 1260 Tie points and Four Distribution Strip of 100 Tie points each totaling 400 Tie points

Fixed DC Supply

This Trainer provides on-board Fixed DC Power Supply of +5V, GND, -5V & +12V, -12V These Power terminals are provided in 2mm Banana Socket.

Variable DC Supply

This Trainer has provision for on-board two different Variable DC Power Supplies. They are from 0 to +15V, & 0 to -15V. These Output Power terminals are provided in 2mm Banana Socket. For varying the voltage POT is provided on-board.

INTRODUCTION

This Digital Circuit Trainer has been designed with the idea of providing basic facilities essential for conducting simple experiments in the laboratory. Using these facilities one can get oneself familiarized with the various digital ICs. The system is suitable for conducting experiments on TTL as well as CMOS ICs.

INSTALLATION

The Digital Bread Board Trainer on 230V, 50Hz power supply. By connecting 230V power point is by means of the power chord of the Trainer. Please follow the below instruction while installing the Trainer.

1. Connect the Trainer kit to 230V power point.
2. Switch ON the power supply provided in the kit.
3. Now your kit is ready for conducting the experiments on different ICs.

Note: Please take precaution to turn OFF the power supply while connecting the circuits.

EXPERIMENT 1

OBJECTIVE: Study of operation of all Logic Gates

THEORY:

LOGIC GATES:

A **logic gate** is an idealized or physical device implementing a Boolean function, that is, it performs a logical operation on one or more logic inputs and produces a single logic output. Depending on the context, the term may refer to **an ideal logic gate**, one that has for instance zero rise time and unlimited fan-out, or it may refer to a non-ideal physical device.
Logic gates are primarily implemented using diodes or transistors acting as electronic switches, but can also be constructed using electromagnetic relays (relay logic), fluidic logic, pneumatic logic, optics, molecules, or even mechanical elements. With amplification, logic gates can be cascaded in the same way that Boolean functions can be composed, allowing the construction of a physical model of all of Boolean logic, and therefore, all of the algorithms and mathematics that can be described with Boolean logic.

AND GATE:

The **AND gate** is a basic digital logic gate that implements logical conjunction - it behaves according to the truth table to the right. A HIGH output (1) results only if both the inputs to the AND gate are HIGH. If neither or only one input to the AND gate is HIGH, a LOW output results.

NAND GATE:

The **Negated AND**, **NOT AND** or **NAND** gate is the opposite of the digital AND gate, and behaves in a manner that corresponds to the opposite of AND gate, as shown in the truth table on the right. A LOW (0) output results only if both the inputs to the gate are HIGH (1); if one or both inputs are LOW (0), a HIGH (1) output results.

OR GATE:

The **OR gate** is a digital logic gate that implements logical disjunction - it behaves according to the truth table to the right. A HIGH output (1) results if one or both the inputs to the gate are HIGH (1). If neither input is HIGH, a LOW output (0) results.

NOR GATE:

The **Negated OR**, **NOT NOR** or **NOR** gate is the opposite of the digital OR gate, and behaves in a manner that corresponds to the opposite of OR gate, as shown in the truth table. A HIGH (1) output results only if both the inputs to the gate are LOW (0); if one or both inputs are HIGH (1), a LOW (0) output results.

EX-OR GATE:

The **X-OR gate** (sometimes **EOR gate**, or **EXOR gate**) is a digital logic gate that implements an exclusive or; that is, a true output (1) results if one, and only one, of the inputs to the gate is true (1). If both inputs are false (0) or both are true (1), a false output (0) results.

EX-NOR GATE:

The **X-NOR gate** (sometimes **ENOR gate**, or **EXNOR gate**) is a digital logic gate that implements an exclusive nor; that is, a true output (1) results if both inputs are false (0) or both are true (1), If one, and only one, of the inputs to the gate is true (1) a false output (0) results.

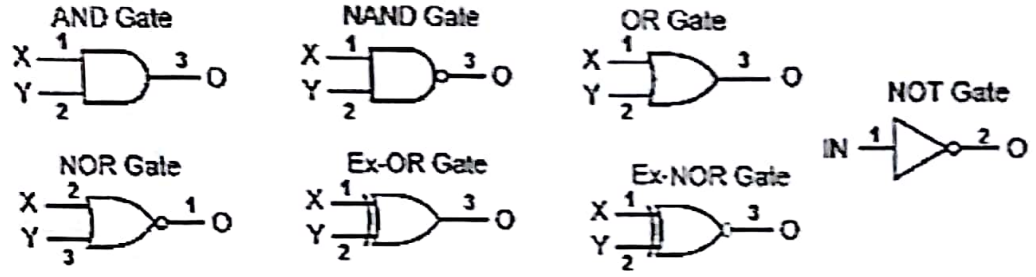
NOT GATE (INVERTER):

The **NOT gate** is also known as an inverter; simply because it changes the input to its opposite (inverts it). The NOT gate accepts only one input and the output is the opposite of the input. In other words, a low-voltage input (0) is converted to a high-voltage output (1).

QUIPMENTS NEEDED:

COMPONENTS	QUANTITY
1. TRAINER KIT	1
2. IC 7408 2 input AND Gate.	1
3. IC 7400 2 input NAND Gate.	1
4. IC 7432 2 input OR Gate.	1
5. IC 7402 2 input NOR Gate	1
6. IC 7486 2 input EX-OR Gate	1
7. IC 74266 2 input EX-NOR Gate	1
8. IC 7404 NOT Gate.	1

OGIC DIAGRAMS:



PROCEDURE:

When Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Switch is **un-pressed** it indicates switch is in **"LOW"** Position.

AND GATE:

1. Place IC 7408 on Bread board.
2. Connect **+5 V** to pin no. 14 of IC 7408 and connect GND to pin no.7. (Refer IC pin diagram).
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of AND Gate to the input switches **I0 & I1,** respectively.
4. Connect output of AND Gate i.e. pin no.3 to **L0** of Logic Output LED.
5. Switch **ON** the Trainer.
6. Set the input switches **S0 & S1** Initially to **LOW** position.
7. Observe output of AND Gate on LED **O0** of Logic Output LED.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

NAND GATE:

1. Place IC 7400 on Bread board.
2. Connect **+5** V to pin no. 14 of IC 7400 and GND to pin no. 7.(Refer IC pin diagram)
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of NAND Gate to the input switches **I2 & I3,** respectively.
4. Connect output of NAND Gate i.e. pin no.3 to **L1** of Logic Output LED
5. Switch **ON** the trainer.
6. Set the input switches **S2 & S3** Initially to **LOW** position.
7. Observe output of NAND Gate on LED **O1** of Logic Output LED.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

OR GATE:

1. Place IC 7432 on Bread board.
2. Connect +5 V to pin no. 14 of IC 7432 and connect GND to pin no.7 (Refer IC pin diagram)
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of OR Gate to the input switches **I4 & I5,** respectively.
4. Connect output of OR Gate i.e. pin no. 3 to **L2** of Logic Output LED.
5. Switch **ON** the trainer.
6. Set the input switches **S4 & S5** Initially to **LOW** position.
7. Observe output of OR Gate on LED **O2** of Logic Output LED
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

NOR GATE:

1. Place IC 7402 on Bread board.
2. Connect **+ 5V** to pin no. 14 of IC 7402 and connect **GND** to pin no.7 (Refer IC pin diagram)
3. Connect Inputs **X & Y (Pin no. 2 & 3)** of 'NOR' Gate to the input switches **I6 & I7,** respectively.
4. Connect output of 'NOR' Gate i.e. pin no. 1 to **L3** of Logic Output LED.
5. Switch ON the trainer.
6. Set the input switches **S6 & S7** Initially to **LOW** position.
7. Observe output of 'NOR' Gate on LED **O3** of Logic Output LED.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

EX-OR GATE:

1. Place IC 7486 on Bread board.
2. Connect **+5V** to pin no. 14 of IC 7486 and connect **GND** to pin no.7 (Refer IC pin diagram)
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of EX-OR Gate to the input switches **I8 & I9,** respectively.
4. Connect output of EX-OR Gate i.e. pin no. 3 to **L4** of 16 output LED indicator.
5. Switch ON the trainer.
6. Set the input switches **S8 & S9** Initially to **LOW** position.
7. Observe output of EX-OR Gate on LED **O4** of Logic Output LED.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

EX-NOR GATE:

1. Place IC 74266 on Bread board or socket.
2. Connect **+5V** to pin no. 14 of IC 74266 and connect **GND** to pin no.7 (Refer IC pin diagram)
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of EX-NOR Gate to the input switches **I0 & I1,** respectively.
4. Connect output of EX-NOR Gate i.e. pin no. 3 to **L5** of Logic Output LED.
5. Switch ON the trainer.
6. Set the input switches **S0 & S1** Initially to **LOW** position.
7. Observe output of EX-NOR Gate on LED **O5** of Logic Output LED.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

NOT GATE:

1. Place IC 7404 on Bread board.
2. Connect **+5 V** to pin 14 of IC 7404 and connect **GND** to pin no.7 (Refer IC pin diagram)
3. Connect Inputs IN (Pin no. 1) of NOT Gate to the input switch **I2.**
4. Connect output of NOT gate i.e. pin no. 2 to **L6** of Logic Output LED.
5. Switch **ON** the Trainer.
6. Set the input switch **S2** Initially to LOW position.
7. Observe output of NOT Gate on LED **O6** of Logic Output LED.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

TRUTH TABLES:

Input 1	Input 2	Output
X	Y	O
0	0	0
0	1	0
1	0	0
1	1	1

Table- AND Gate

Input 1	Input 2	Output
X	Y	O
0	0	1
0	1	1
1	0	1
1	1	0

Table- NAND Gate

Input 1	Input 2	Output
X	Y	O
0	0	0
0	1	1
1	0	1
1	1	1

Table- OR Gate

Input 1	Input 2	Output
X	Y	O
0	0	1
0	1	0
1	0	0
1	1	0

Table- NOR Gate

Input 1	Input 2	Output
X	Y	O
0	0	0
0	1	1
1	0	1
1	1	0

Table- EX-OR Gate

Input 1	Input 2	Output
X	Y	O
0	0	1
0	1	0
1	0	0
1	1	1

Table- EX-NOR Gate

Input	Output
0	1
1	0

Table- NOT Gate

CONCLUSION:

Hence we have studied the logic gates.

EXPERIMENT 2

OBJECTIVE: Study of binary adders

A. Half Adder
B. Full Adder

THEORY:

ADDER:

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar.

HALF ADDER:

Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output. If A and B are the input bits, then sum bit (S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B. From this it is clear that a half adder circuit can be easily constructed using one X-OR gate and one AND gate. Half adder is the simplest of all adder circuit, but it has a major disadvantage. The half adder can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input. So if the input to a half adder have a carry, then it will be neglected it and adds only the A and B bits. That means the binary addition process is not complete and that's why it is called a half adder.

FULL ADDER:

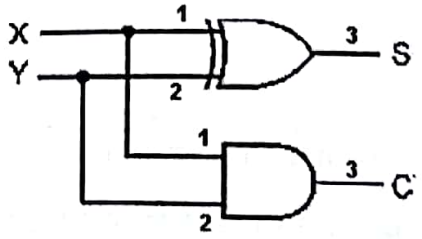
This type of adder is a little more difficult to implement than a half-adder. The main difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as CIN. When a full adder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

A. HALF ADDER

EQUIPMENTS NEEDED:

Components	Quantity
1. Trainer Kit	1
1. IC 7408 2 input AND Gate.	1
2. IC 7486 2 input EX-OR Gate.	1

LOGIC DIAGRAM:



Fig

PROCEDURE:

When Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Switch is **un-pressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board as shown in **fig.**
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of IC 7408 and IC 7486. See IC Pin diagram.
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of AND & EX-OR Gate to the input switches **I0 & I1,** respectively.
4. Connect Sum **(S)** & Carry **(C)** to **L0 & L1** of Logic Output LED, respectively.
5. Switch **ON** the instrument.
6. Set the input switches **S0 & S1** Initially to **LOW** position.
7. Observe outputs **S** and **C** on LED **O0 & O1** of 16 Logic Output LED, respectively.
8. Observe the output for different input combination as shown in Truth Table
9. Verify Truth Table

TRUTH TABLE:

Input1	Input2	Carry	Sum
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

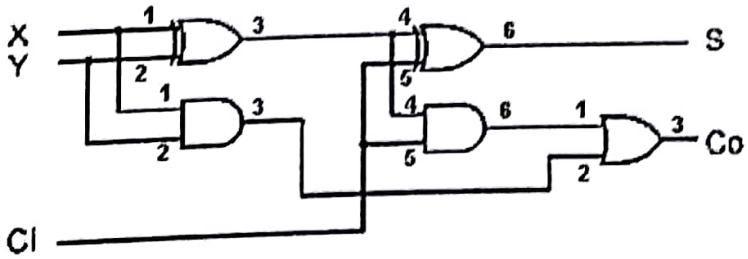
Truth Table

B. FULL ADDER:

EQUIPMENTS NEEDED:

Component	Quantity
1. IC 7408 2 input AND Gate	1
2. IC 7432 2 input OR Gate	1
3. IC 7486 2 input EX-OR Gate	1

LOGIC DIAGRAM:



Fig

PROCEDURE:

1. Make connections on bread board as shown in **fig.**
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of IC 7408, IC 7432 & IC 7486 (See IC Pin Diagram).
3. Connect Inputs **X & Y** (Pin no. 1 & 2) of AND & EX-OR Gate to the input switches **I2 & I3,** respectively.
4. Connect **CI** to input switch **I4** as shown in fig
5. Connect Sum **(S)** & Carry out **(Co)** to **L2 & L3** of Logic Output LED, respectively.
6. Switch **ON** the instrument.

7. Set the input switches **S2, S3 & S4** initially to **LOW** position.
8. Observe outputs **S** and **Co** on LED **O2 & O3** of Logic Output LED, respectively.
9. Observe the output for different input combination as shown in Truth Table
10. Verify Truth Table

TRUTH TABLE:

Input 1	Input 2	Input Carry	Sum	Output Carry
X	Y	CI	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Truth Table

CONCLUSION:

Full adder can be implemented using two half Adder. Adders are studied and Truth Tables are verified

EXPERIMENT 3

OBJECTIVE: Study of binary Substractors

A. Half Substractor
B. Full Substractor

THEORY:

SUBTRACTOR:

A subtractor can be designed using the same approach as that of an adder.
The arithmetic operation, subtraction of two binary digits has four possible elementary operations, namely,

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

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

HALF SUBTRACTOR:

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

FULL SUBTRACTOR:

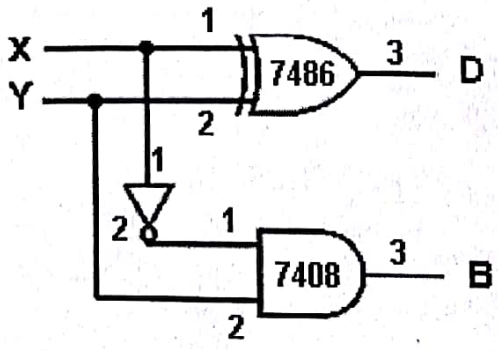
A full-subtractor is a combinational circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage

A. HALF SUBSTRACTOR

EQUIPMENTS NEEDED:

Components	Quantity
1. Trainer Kit	1
1. IC 7408 2 input AND Gate.	1
2. IC 7486 2 input EX-OR Gate.	1
3. IC 7404 NOT Gate	1

LOGIC DIAGRAM:



Fig

PROCEDURE:

When Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Switch is **un-pressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board as shown in **fig.**
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of IC 7408, 7404 and IC 7486. See IC Pin diagram.
3. Connect Inputs **X & Y** to the input switches **I0 & I1,** respectively.
4. Connect Difference **(D)** (i.e. pin no. 3 of IC 7486) & Borrow **(B)** (i.e. pin no. 3 of IC7408) to **L0** & **L1** of Logic Output LED, respectively.
5. Switch **ON** the instrument.
6. Set the input switches **S0 & S1** Initially to **LOW** position.
7. Observe outputs **D** and **B** on LED **O0** & **O1** of Logic Output LED, respectively.
8. Observe the output for different input combination as shown in Truth Table.
9. Verify Truth Table.

TRUTH TABLE:

Input1	Input2	Diff	Borrow
X (S0)	Y(S1)	D (L0)	B (L1)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

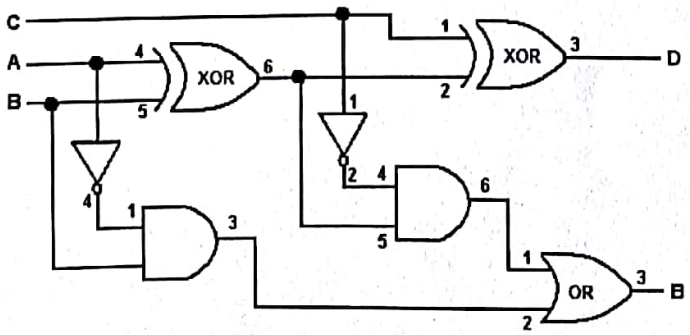
Truth Table

B. FULL SUBSTRACTOR:

EQUIPMENTS NEEDED:

Component	Quantity
1. IC 7408 2 input AND Gate	1
2. IC 7432 2 input OR Gate	1
3. IC 7486 2 input EX-OR Gate	1
4. IC 7404 NOT Gate	1

LOGIC DIAGRAM:



Fig

PROCEDURE:

1. Make connections on bread board or socket as shown in **fig.**
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of IC 7408, 7404, 7432 and IC 7486. See IC Pin diagram.
3. Connect Inputs **A, B & C** the input switches **I0, I1 & I2,** respectively.
4. Connect Diff **(D)** & Borrow **(B)** to **L0** & **L1** of Logic Output LED, respectively.
6. Set the input switches **S0, S1 & S2** Initially to **LOW** position.
7. Switch **ON** the Trainer.
8. Observe outputs **D** and **B** on LED **O0** & **O1** of Logic Output LED, respectively.
9. Observe the output for different input combination as shown in **Truth Table**
10. Verify Truth Table

TRUTH TABLE:

INPUT			OUTPUT	
A	B	C	DIFFERENCE	BORROW
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Truth Table

CONCLUSION:

Subtractors are studied and Truth Tables are verified.

EXPERIMENT 4

OBJECTIVE: Study of Binary to Gray Code Conversion

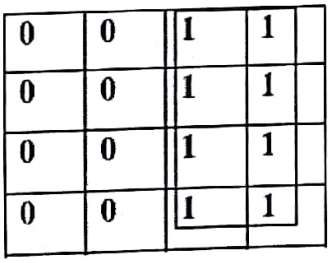
EQUIPMENTS NEEDED:

COMPONENTS	QUANTITY
1 TRAINER KIT	1
2. IC 7486 2 Input EX-OR Gate.	1

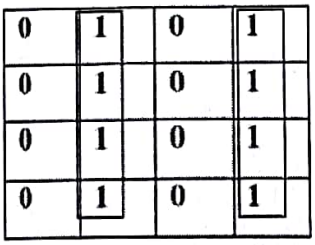
TRUTH TABLE:

Binary Code				Gray Code			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

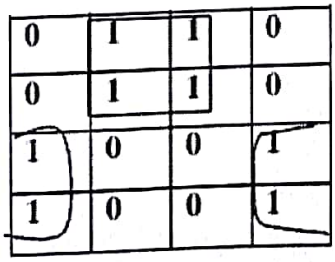
SIMPLIFICATION USING K MAP



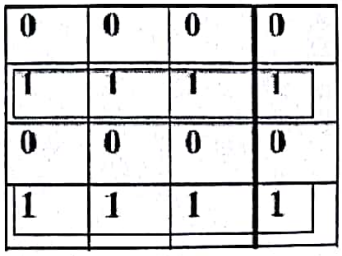
G3 = B3



G2= B3 ⊕ B2

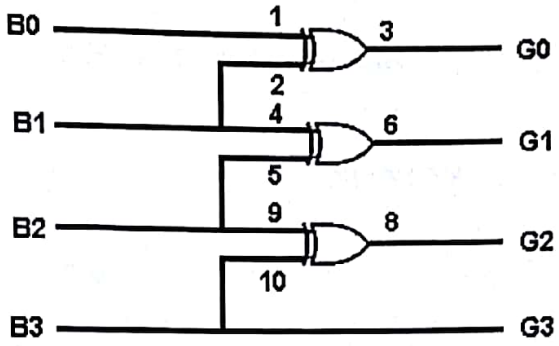


G1=B1 ⊕ B2



G0=B1 ⊕ B0

LOGIC DIAGRAM:



Fig

PROCEDURE:

When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **un-pressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board or socket as shown in **fig.**
2. Connect +5V to pin no. 14 and GND to pin no.7 of ICs. See IC Pin diagram.
3. Connect Inputs **I0-13** to **B0-B3** respectively.
4. Connect outputs **G0-G3** to **O1-O4** of 16 Bits LED Indicator, respectively.
5. Set the input Switches **S0-S4** initially to **LOW** Position.
6. Switch ON the Trainer.
7. Observe outputs G3, G2, G1, G0 on LED **L4, L3, L2 & L1** of 16 bits LED Indicator, respectively.
8. Verify Truth Table for different input combinations as shown in truth Table

CONCLUSION:

Hence we have studied the Binary to gray Code conversion

EXPERIMENT 5

OBJECTIVE: Study Gray Code to Binary Code Conversions

EQUIPMENTS NEEDED:

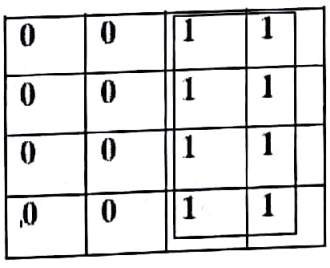
COMPONENTS	QUANTITY
1. TRAINER KIT	1
2. IC 7486 2 Input EX-OR Gate.	1

TRUTH TABLE:

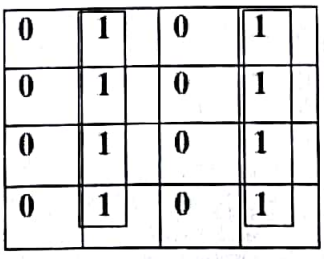
Gray Code				Binary Code			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

Truth Table

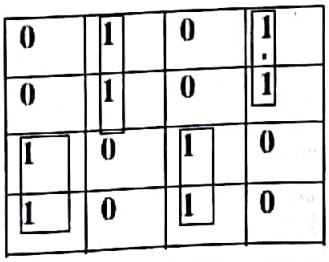
SIMPLIFICATION USING K MAP



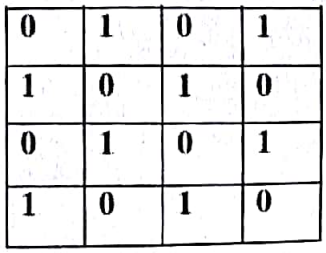
B3 = G3



B2=G3 ⊕ G2

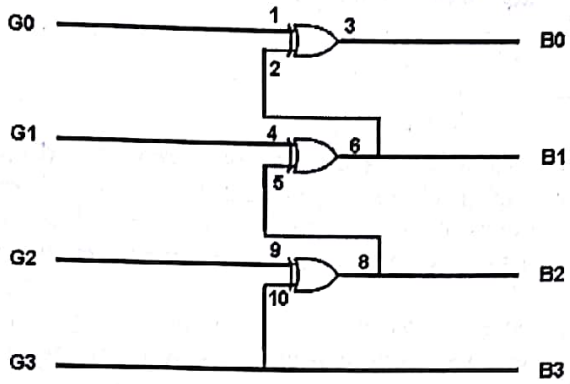


B1=G3 ⊕ G2 ⊕ G1



B0=G3 ⊕ G2 ⊕ G1 ⊕ G0

LOGIC DIAGRAM:



Fig

PROCEDURE:

When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **un-pressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board or socket as shown in **fig**
2. Connect +5V to pin no.14 and GND to pin no.7 of ICs. See IC Pin diagram.
3. Connect Inputs **I0-I3** to **G0-G3** respectively.
4. Connect outputs **B0-B3** to **O1-O4** of 16 Bits LED Indicator, respectively.
5. Set the input Switches **S0-S4** initially to **LOW** Position.
6. Switch ON the Trainer.
7. Observe outputs B3, B2, B1, B0 on LED **L4, L3, L2 & L1** of 16 bits LED Indicator, respectively.
8. Verify Truth Table for different input combinations as shown in truth Table

CONCLUSION:

Hence we have studied the gray to Binary Code conversion

EXPERIMENT 6

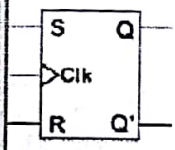
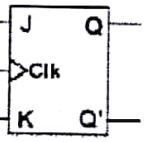
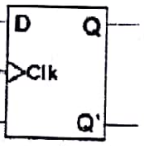
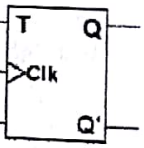
OBJECTIVE: To Study of Characteristics of various types of Flip-Flops

THEORY:

FLIP FLOP:

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

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

Flip-Flop Name	Flip-Flop Symbol	Characteristic Table	Characteristic Equation	Excitation Table
SR		S R Q(next) 0 0 Q 0 1 0 1 0 1 1 1 ?	Q(next) = S + R'Q SR = 0	Q Q(next) S R 0 0 0 X 0 1 1 0 1 0 0 1 1 1 X 0
JK		J K Q(next) 0 0 Q 0 1 0 1 0 1 1 1 Q'	Q(next) = JQ' + K'Q	Q Q(next) J K 0 0 0 X 0 1 1 X 1 0 X 1 1 1 X 0
D		D Q(next) 0 0 1 1	Q(next) = D	Q Q(next) D 0 0 0 0 1 1 1 0 0 1 1 1
T		T Q(next) 0 Q 1 Q'	Q(next) = TQ' + T'Q	Q Q(next) T 0 0 0 0 1 1 1 0 1 1 1 0

The characteristic table in the third column of Table defines the state of each flip-flop as a function of its inputs and previous state. Q refers to the present state and Q(next) refers to the next state after the occurrence of the clock pulse

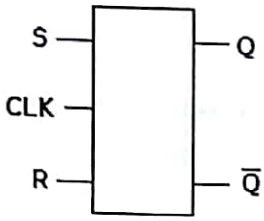
RS FLIP FLOP:

The **SR flip-flop**, also known as a *SR Latch*, can be considered as one of the most basic sequential logic circuit possible. This simple flip-flop is basically a one-bit memory bistable device that has two inputs, one which will "SET" the device (meaning the output = "1"), and is labelled S and another which will "RESET" the device (meaning the output = "0"), labelled R. Then the SR description stands

for "Set-Reset". The reset input resets the flip-flop back to its original state with an output Q that will be either at a logic level "1" or logic "0" depending upon this set/reset condition

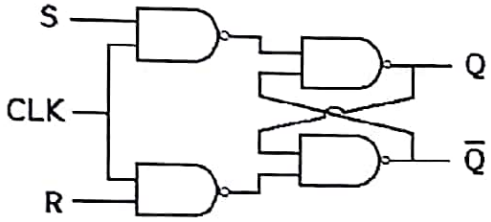
CLOCKED RS FLIP-FLOP:

The clocked RS flip-flop is like an SR flip-flop but with an extra third input of a standard clock pulse CLK. The logic symbol for this flip-flop is shown below



Fig

And one equivalent implementation using NAND gates is illustrated here

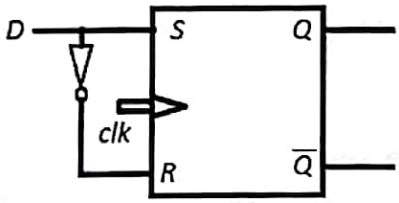


Fig

Bearing in mind that the NAND implementation of an SRFF is driven by 0s then it can be seen that the extra two NAND gates in front of the standard SRFF circuitry mean that the circuit will function as a usual SRFF when S or R are 1 and the clock pulse is also 1 ("high"). Therefore this flip-flop is synchronous. Specifically, a 0 to 1 transition on either of the inputs S or R will only be seen at the output if the clock is 1.

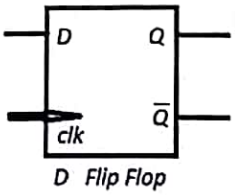
D FLIP FLOP:

The D (delay) flip-flop has only one input called the delay (D) input and two outputs Q and $\overline{Q}$. It can be constructed from an SR flip flop by a inserting an inverter between S and R and assigning the symbol D to the S input. The structure of D flip flop is shown below



Fig

It can be represented by logic symbol



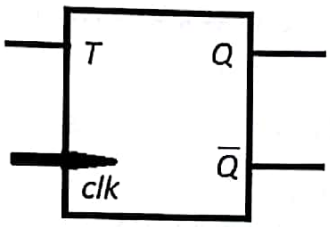
Fig

JK FLIP FLOP:

This simple **JK flip-Flop** is the most widely used of all the flip-flop designs and is considered to be a universal flip-flop circuit. The sequential operation of the JK flip-flop is exactly the same as for the previous SR flip-flop with the same "set" and "reset" inputs. The difference this time is that the JK flip-flop has no invalid or forbidden input states of the SR Latch (when S and R are both 1).

T FLIP FLOP:

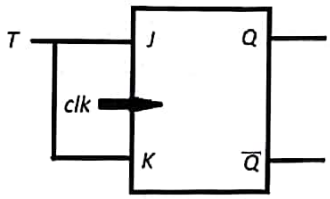
T flip flop or toggle flip flop , has only one input , a clock input and two output Q and $\overline{Q}$. When T= 1 the output is toggled or inverted. The symbolic representation of T flip flop is shown below



T Flip Flop

Fig

The T flip flop is obtained from a JK flip flop by connecting its J and K input together .The diagram is shown below



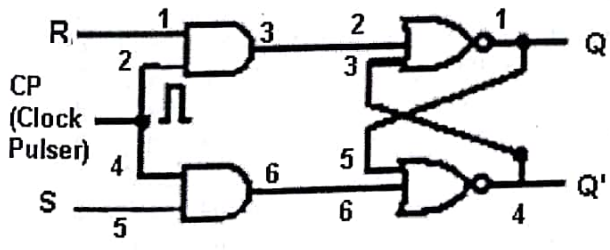
T Flip Flop Using JK Flip Flop

Fig

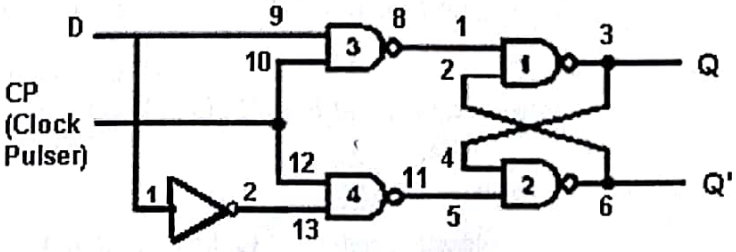
EQUIPMENTS NEEDED:

COMPONENT	Quantity
1. Trainer	1
2. IC 7411 3 input AND Gate	1
3. IC 7408 2 input AND Gate	1
4. IC 7400 2 input NAND Gate	1
5. IC 7402 2 input NOR Gate	1
6. IC 7404 NOT Gate	1

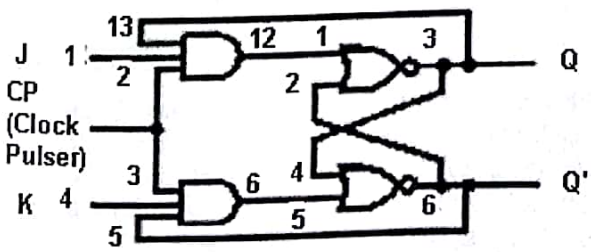
LOGIC DIAGRAM:



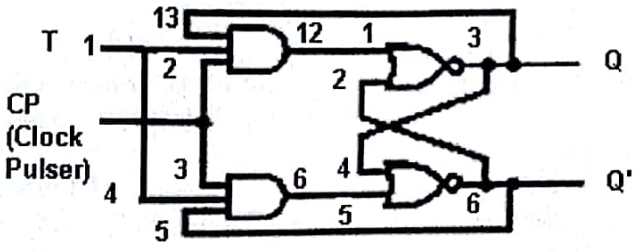
Fig



Fig



Fig



Fig

PROCEDURE:

When Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Switch is **un-pressed** it indicates switch is in **"LOW"** Position.

RS FLIP FLOP

1. Make connections on bread board as shown in **fig** for **RS Flip flop**. (Refer pin diagram of ICs)
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of ICs. See IC Pin diagram.
3. Connect inputs **R & S** of RS flip flop to Inputs **I0** and **I1**, respectively.
4. Connect **Manual Pulser (T2)** to clock input CP of flip-flop
5. Connect outputs of flip-flop **Q** and **Q'** to **L0** and **L1** of Logic Output LED, respectively.
6. Set the Input Switches **S0 & S1** initially to **LOW** position.
7. Switch **ON** the Trainer.
8. Press **Pulsar switch** to get output.
9. Observe outputs **Q** and **Q'** on LED **O0** & **O1** of Logic Output LED, respectively.
10. Observe the output for different input combination as shown in **Truth Table.**
11. Verify Truth Table

J K FLIP FLOP

1. Make connections on bread board as shown in **fig** for **J K Flip flop**. (Refer pin diagram of ICs)
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of ICs. See IC Pin diagram..
3. Connect inputs **J & K** of JK flip flop to Inputs **I2** and **I3**, respectively.
4. Connect **Manual Pulser (T2)** to clock input CP of flip-flop
5. Connect outputs of flip-flop **Q** and **Q'** to **L2** and **L3** of Logic Output LED, respectively.
6. Set the Input Switches **S2 & S3** initially to **LOW** position.
7. Switch **ON** the Trainer.
8. Press **Pulsar** switch to get output.
9. Observe outputs **Q** and **Q'** on LED **O2** & **O3** of Logic Output LED, respectively.
10. Observe the output for different input combination as shown in **Truth Table.**
11. Verify Truth Table

D FLIP FLOP

1. Make connections on bread board as shown in **fig** for D Flip flop. (Refer pin diagram of ICs)
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of ICs. See IC Pin diagram.
3. Connect input **D** of D flip flop to Input **I4**.
4. Connect **Manual Pulser (T2)** to clock input CP of flip-flop
5. Connect output of flip-flop **Q** and **Q'** to **L4** and **L5** of Logic Output LED, respectively.
6. Set the Input Switch **S4** initially to **LOW** position.
7. Switch **ON** the Trainer.
8. Press **Pulsar** switch to get output.
9. Observe outputs **Q** and **Q'** on LED **O4** & **O5** of Logic Output LED, respectively.
10. Observe the output for different input combination as shown in **Truth Table.**
11. Verify Truth Table

T FLIP FLOP

1. Make connections on bread board shown in **fig** for **T Flip flop**. (Refer pin diagram of ICs)
2. Connect **+5V** to pin no. 14 and **GND** to pin no.7 of ICs. See IC Pin diagram.
3. Connect input **T** of T flip flop to Input **I5.**
4. Connect **Manual Pulser (T2))** to clock input **CP** of flip-flop
5. Connect output of flip-flop **Q** and **Q'** to **L6** and **L7** of Logic Output LED, respectively.
6. Set the Input Switch **S5** initially to **LOW** position.
7. Switch **ON** the Trainer.

8. Press **Pulsar** switch to get output.
9. Observe outputs **Q** and **Q'** on LED **O6** & **O7** of Logic Output LED, respectively.
10. Observe the output for different input combination as shown in **Truth Table-7.4**
11. Verify Truth Table

TRUTH TABLE:

Clocked RS Flip Flop			
Present State	Input 1	Input 2	Next State
Q	S	R	Q (t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	In determinant
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	In determinant

Truth Table

J K Flip Flop			
Present State	Input 1	Input 2	Next State
Q	J	K	Q (t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Truth Table

D Flip Flop		
Present State	Input	Next State
Q	D	Q (t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Truth Table

T Flip Flop		
Present State	Input	Next State
Q	T	Q (t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table

CONCLUSION:

Thus we have studied the flip-flops.

EXPERIMENT-7

OBJECTIVE: Study of Multiplexer and De-multiplexer Circuit

THEORY:

MULTILPLEXER:

In electronics, a **multiplexer** (or **MUX**) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a **data selector**. They are used in CCTV, and almost every business that has CCTV fitted, will own one of these.

DEMULTIPLEXER:

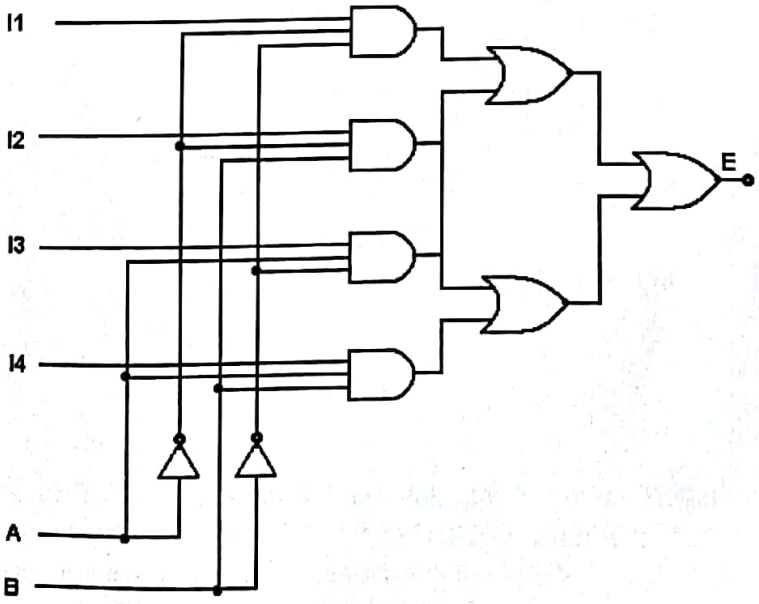
The data distributor, known more commonly as a **Demultiplexer** or "Demux", is the exact opposite of the **Multiplexer**. The demultiplexer takes one single input data line and then switches it to any one of a number of individual output lines one at a time. The **demultiplexer** converts a serial data signal at the input to a parallel data at its output lines.

EQUIPMENTS NEEDED:

Component	Quantity
1. Trainer	1
2. IC 7411 3 input AND Gate.	2
3. IC 7432 2 Input OR Gate.	1
4. IC 7404 NOT Gate.	1

LOGIC DIAGRAM:

4 To 1 Line Multiplexer



Fig

PROCEDURE:

When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **unpressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board as shown in **fig** (Refer pin diagram of ICs).
2. Connect **+ 5 V** to pin no. 14 and **GND** to pin no.7 of ICs.
3. Connect Inputs **I1-I4** to inputs **I0-I3** of logic data switches, respectively.
4. Apply inputs 1 or 0 to input lines 11, 12, 13, 14.
5. Connect Selection Switches **A & B** to the input switches **I4 & I5,** respectively.
6. Set the input switches **S4 & S5** initially to **LOW** position.
7. Connect Output **E** to **L0** of Logic Output LED.
7. Switch **ON** the power supply.
8. Observe outputs on LED **O0** of Logic Output LED.
9. Observe the output for different input combinations as shown in **truth Table.**

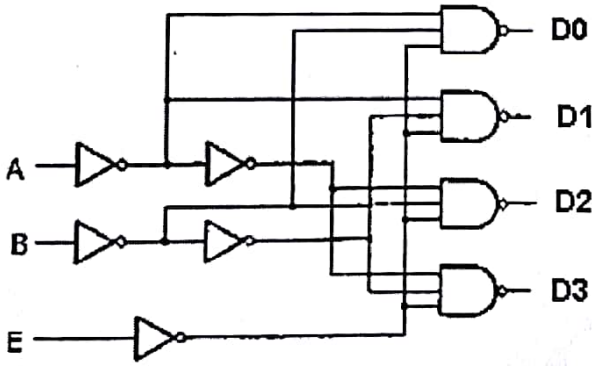
TRUTH TABLE

Selection line 1 (A) S4	Selection line 2 (B) S5	Input Selected (E)
0	0	I1
0	1	I2
1	0	I3
1	1	I4

Truth Table

LOGIC DIAGRAM:

1 To 4 Line De-multiplexer :



Fig

PROCEDURE:

When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **unpressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board as shown in **fig** (Refer pin diagram of ICs).
2. Select **+ 5 V** to pin no. 14 and **GND** to pin no.7 of ICs.
3. Connect Input **E** to **I0** of Input switch
4. Apply inputs 1 or 0 to input E.
5. Connect selection lines **S0** and **S1** to **I1** and **I2** of Logic data Input Switches, respectively.

6. Set the Input Switches **S0-S2** initially to **LOW** position.
7. Connect outputs **D0-D3** to **L0-L3** of Logic Output LED, respectively.
8. Switch **ON** the instrument
9. Observe outputs on LEDs **O0, O1, O2 & O3** of Logic Output LED.
10. Observe the output for different input combinations as shown in **truth Table.**

TRUTH TABLE:

Input	Selection line	Selection line	Output Line	Output Line	Output Line	Output Line
E	A	B	D0	DI	D2	D3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Truth Table.

CONCLUSION:

Thus we have studied the Multiplexer and De-multiplexer Circuit.

EXPERIMENT-8

OBJECTIVE: Study of Demorgan's Theorem

THEORY:

There are actually two theorems that were put forward by De-Morgan. On the basis of DE Morgan's laws much Boolean algebra are solved. Solving these types of algebra with De-Morgan's theorem has a major application in the field of digital electronics. De Morgan's theorem can be stated as follows:-

Theorem 1:

The compliment of the product of two variables is equal to the sum of the compliment of each variable. Thus according to **De-Morgan's laws** or De-Morgan's theorem if A and B are the two variables or Boolean numbers. Then accordingly

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

Theorem-2:

The compliment of the sum of two variables is equal to the product of the compliment of each variable. Thus according to De Morgan's theorem if A and B are the two variables then.

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

EQUIPMENTS NEEDED:

Component	Quantity
1. Trainer	1
2. IC 7400 2 input NAND Gate.	1
3. IC 7432 2 Input OR Gate.	1
4. IC 7404 NOT Gate.	1
5. IC 7402 2 input NOR Gate	1
6. IC 74010 2 input AND Gate.	1

LOGIC DIAGRAM:

Theorem-1 $\overline{A.B} = \overline{A} + \overline{B}$

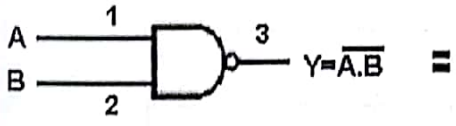


Fig-9.1 (LHS)

=

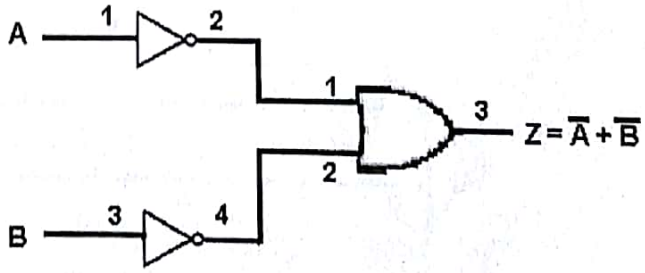


Fig-9.2 (RHS)

Theorem-2 $\overline{A+B} = \overline{A}.\overline{B}$

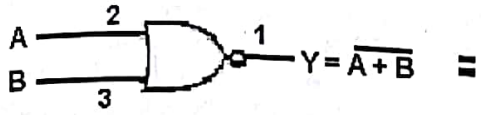


Fig-9.3 (LHS)

=

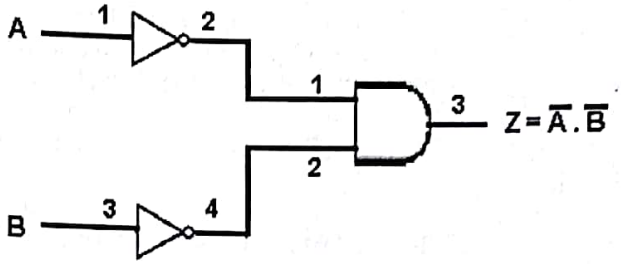


Fig-9.4 (RHS)

PROCEDURE:

Let us assume,

When **RED** LED glows it indicates logic **High** or **1**
When **GREEN** LED glows **(Output LED Indicator)** it indicates logic **Low** or **0.**
When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **unpressed** it indicates switch is in **"LOW"** Position.

Theorem-1 $\overline{A.B} = \overline{A} + \overline{B}$

Left Hand Side: (A.B)'

1. Make connections on bread board as shown in fig-9.1 (Refer pin diagram of ICs).
2. Select **+ 5 V** it to pin no. 14 and connect **GND** to pin no.7 of IC 7400.
3. Connect Inputs **A & B** (i.e. pin no.1 & 2) to **I0 & I1** of Input Switches, respectively.
4. Set the input switches **S0 & S1** initially to **LOW** position.
7. Connect Output **Y** (i.e. pin no. 3) to **L0** of Logic Output LED
7. Switch **ON** the power supply.
8. Observe output on LED **O0** of Logic Output LED
9. Observe the output for different input combinations as shown in **truth Table-9.1**

TRUTH TABLE:

A	B	(A.B)	Y = (A. B)'
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Truth Table 9.1

Right Hand Side: (A'+B')

1. Make connections on bread board as shown in fig-8.2 (Refer pin diagram of ICs).
2. Select **+ 5 V** it to pin no. 14 and connect **GND** to pin no.7 of IC 7404 & IC 7432.
3. Connect Inputs **A & B** (i.e. pin no. 1 & 3) to **I2 & I3** of Input Switches, respectively.
4. Set the input switches **S2 & S3** initially to **LOW** position.
7. Connect Output **Z** (i.e. pin no. 3 of IC 7432) to **L1** of Logic Output LED.
7. Switch **ON** the power supply.
8. Observe output on LED **O1** of Logic Output LED.
9. Observe the output for different input combinations as shown in **truth Table-9.2**

TRUTH TABLE:

A	B	A'	B'	Z = A'+ B'
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

Truth Table-9.2

CONCLUSION: From the truth tables, it is clear that, Y = Z and hence Demorgan's Theorem 1 is verified.

Theorem-2 $\overline{A + B} = \overline{A} . \overline{B}$

Left Hand Side: (A+B)'

1. Make connections on bread board as shown in fig-8.3 (Refer pin diagram of ICs).
2. Select **+ 5 V** to pin no. 14 and connect **GND** to pin no.7 of IC 7402.
3. Connect Inputs **A & B** (i.e. pin no.2 & 3) to **I4 & I5** of Input Switches, respectively.
4. Set the input switches **S4 & S5** initially to **LOW** position.
7. Connect Output **Y** (i.e. pin no. 1) to **L2** of Logic Output LED.
7. Switch **ON** the power supply.
8. Observe output on LED **O2** of Logic Output LED
9. Observe the output for different input combinations as shown in **truth Table-9.3**

TRUTH TABLE:

A	B	A + B	Y = (A+B)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Truth Table-9.3

Right Hand Side: (A'.B')

1. Make connections on bread board as shown in fig-8.4 (Refer pin diagram of ICs).
2. Select **+ 5 V** to pin no. 14 and connect **GND** to pin no.7 of IC 7404 & IC 74010.
3. Connect Inputs **A & B** (i.e. pin no. 1 & 3 of NOT Gate) to **I6 & I7** of Input Switches, respectively.
4. Set the input switches **S6 & S7** initially to **LOW** position.
7. Connect Output **Z** (i.e. pin no. 3 of IC 74010) to **L3** of Logic Output LED.
7. Switch **ON** the power supply.
8. Observe output on LED **O3** of Logic Output LED.
9. Observe the output for different input combinations as shown in **truth Table-9.4**

TRUTH TABLE:

A	B	A'	B'	Z = A'. B'
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Truth Table-9.4

CONCLUSION:

From the truth tables, it is clear that, Y = Z and hence Demorgan's Theorem 2 is verified.

EXPERIMENT-9

OBJECTIVE: Study of 8 to 3 Line Encoder and 3 to 8 Line Decoder

THEORY:

ENCODER:

An **encoder** is a device used to change a signal (such as a bit stream) or data into a code. The code may serve any of a number of purposes such as compressing information for transmission or storage, encrypting or adding redundancies to the input code. translating from one code to another. This is usually done by means of a programmed algorithm, especially if any part is digital, while most analogue encoding is done with analog circuitry.

DECODER:

A **decoder** is a device which does the reverse operation of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode. It is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

In digital electronics, a decoder can take the form of a multiple-input, multiple-output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different. E.g. n-to-2^n, binary-coded decimal decoders. Enable inputs must be on for the decoder to function, otherwise its outputs assume a single "disabled" output code word. Decoding is necessary in applications such as data multiplexing, 7 segment display and memory address decoding.

A. 8 X 3 ENCODER:

EQUIPMENTS NEEDED:

Component	Quantity
1. Trainer	1
2. IC 4072 4 Input OR Gate.	1

LOGIC DIAGRAM:

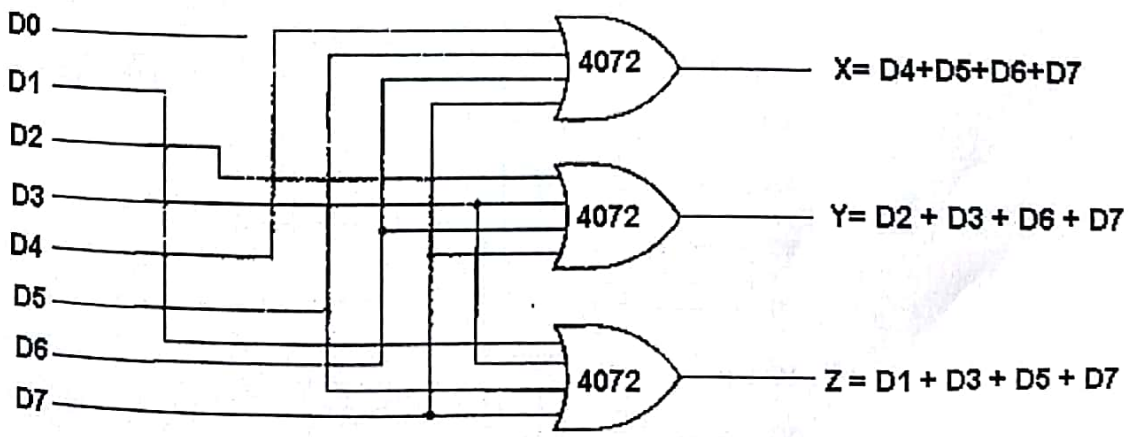


Fig-10.1

PROCEDURE:

When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **unpressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board as shown in **fig** (Refer pin diagram of IC).
2. Select **+ 5 V** to pin no. 14 and connect **GND** to pin no.7 of IC 4072.
3. Connect Inputs **(D0-D7)** to **(I0- I7)** of Input Switches, respectively.
4. Set the input switches **S0-S7** initially to **LOW** position.
7. Connect Outputs X, Y & Z to **L0, L1 & L2** of Logic Output LED, Respectively.
7. Switch **ON** the power supply.
8. Observe outputs on LED **O0, O1 & O2** of Logic Output LED.
9. Set the input switches **S0-S7** to various Positions as shown in the **Truth Table.**

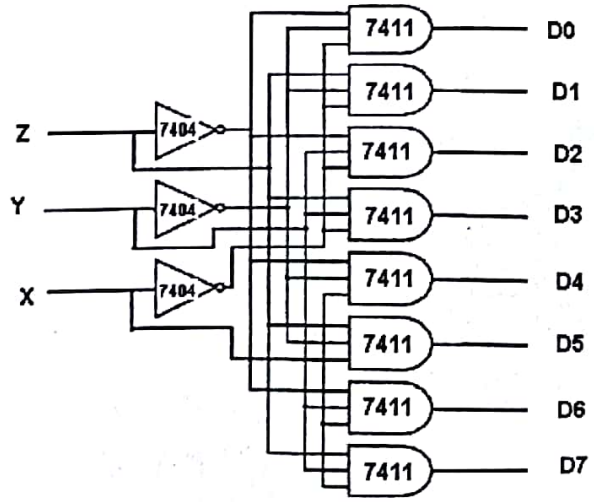
TRUTH TABLE:

INPUTS								OUTPUTS		
D0 (S0)	D1 (S1)	D2 (S2)	D3 (S3)	D4 (S4)	D5 (S5)	D6 (S6)	D7 (S7)	X (O0)	Y (O1)	Z (O2)
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Truth Table

B. 3 X 8 DECODER:

LOGIC DIAGRAM:



Fig

PROCEDURE:

When Input Switch is **pressed** it indicates switch is in **"HIGH"** Position.
When Input Switch is **unpressed** it indicates switch is in **"LOW"** Position.

1. Make connections on bread board as shown in **fig** (Refer pin diagram of IC).
2. Select **+ 5 V** to pin no. 14 and connect **GND** to pin no.7 of IC 4072.
3. Connect Inputs **(X, Y & Z)** to **(I0- I2)** of Input Switches, respectively.
4. Set the input switches **S0-S2** initially to **LOW** position.
7. Connect Outputs **(D0-D7)** to **L0-L7** of Logic Output LED, Respectively.
7. Switch **ON** the power supply.
8. Observe outputs on LED **O0-O7** of Logic Output LED.
9. Set the input switches **S0-S2** to various Positions as shown in the **Truth Table.**

TRUTH TABLE:

INPUTS			OUTPUTS							
X (S0)	Y (S1)	Z (S2)	D0 (O0)	DI (O1)	D2 (O2)	D3 (O3)	D4 (O4)	D5 (O5)	D6 (O6)	D7 (O7)
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Truth Table

CONCLUSION:

Thus we have studied the 8 to 3 Line Encoder and 3 to 8 Line Decoder

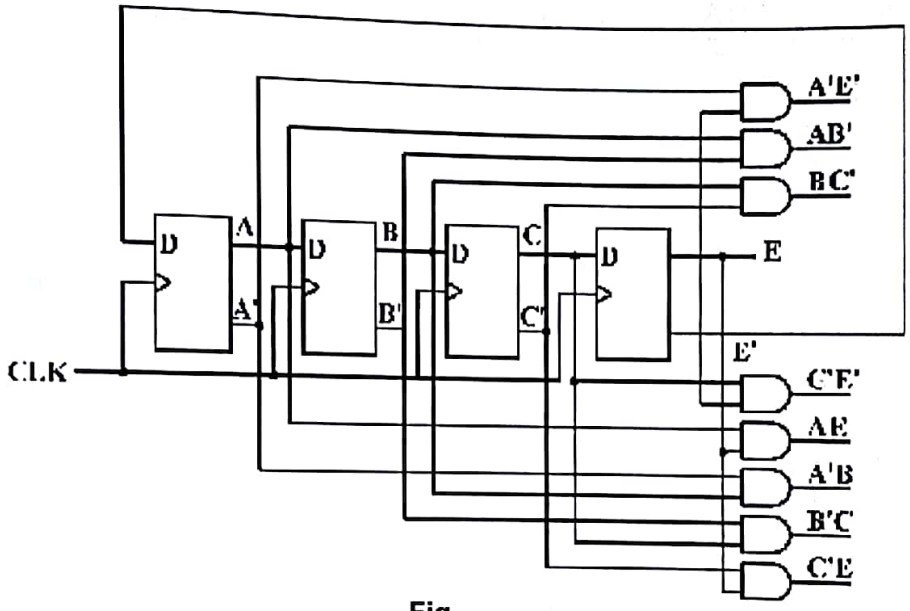
EXPERIMENT 10

OBJECTIVE: Study of Johnson Counter

EQUIPMENTS NEEDED:

Components	Quantity
1. Trainer	1
2. IC 7474 Dual D Flip Flop	2
3. IC 7408 2 input AND Gate	2

LOGIC DIAGRAM:



Fig

PROCEDURE:

1. Make connections on bread Board as shown in figure (Refer to pin diagram of ICs)
2. Connect **+5** V to pin no. 14 and **GND** to pin no. 7of ICs.
3. Connect GND to clear input of flip flop and disconnect it.
4. Connect **T1** of Manual pulsar switch to **CLK** input of flip-flops.
5. Switch **ON** the instrument.
6. Press Manual pulsar switch **SW7.**
7. Observe outputs on Logic Output LED.
8. Press Manual pulsar switch **SW7** and observe change in output. It will follow Sequence as shown in Truth Table.
9. Verify Truth Table.
10. AND & NOT Gates can be used to make product terms of last column and can be observed on Logic Output LED to know when a particular Combination occurs.

TRUTH TABLE:

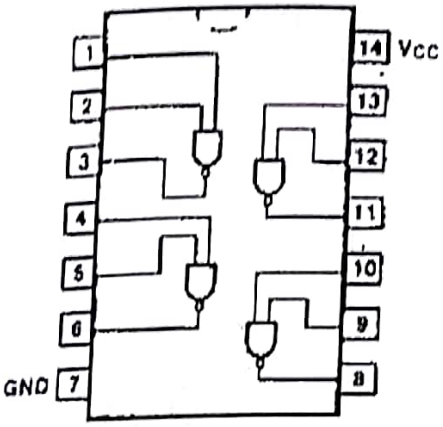
Sr. No.	Flip flop output A	B	C	E	AND Gate required for output
1	0	0	0	0	A'E'
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	A'B
7	0	0	1	1	B'C
8	0	0	0	1	C'E

CONCLUSION:

Johnson counter gives 2k distinguishable states with k flip-flop.

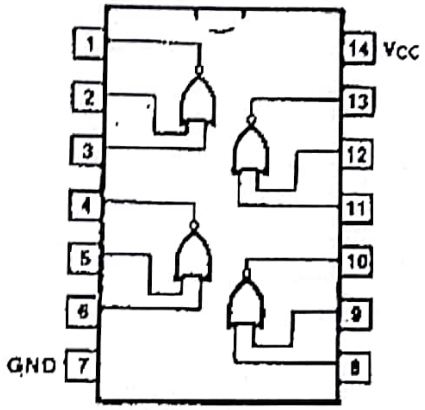
PIN DIAGRAMS OF ICS

54/7400
54H/74H00
54S/74S00
54LS/74LS00
QUAD 2-INPUT NAND GATE

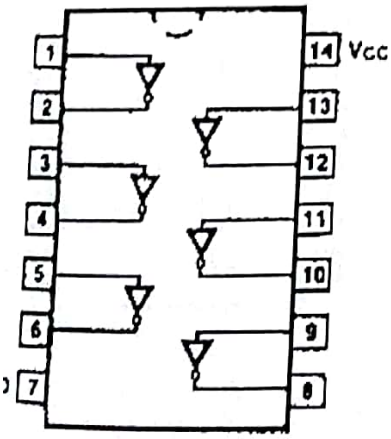


54/7402
54S/74S02
54LS/74LS02

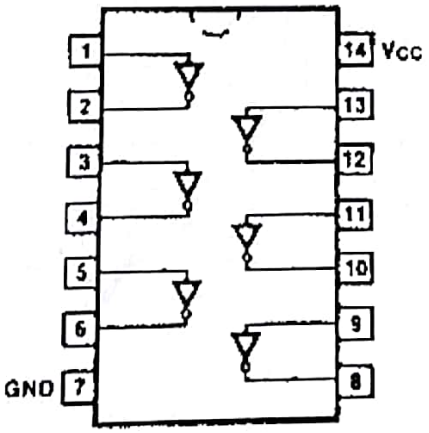
QUAD 2-INPUT NOR GATE



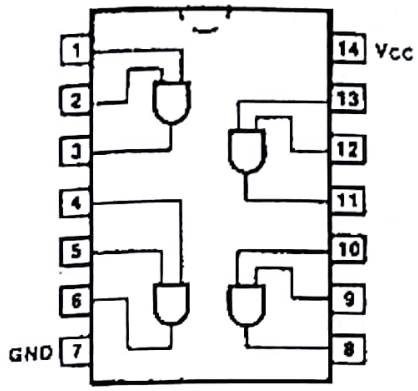
54/7404
54H/74H04
54S/74S04
54S/74S04
54LS/74LS04
HEX INVERTER



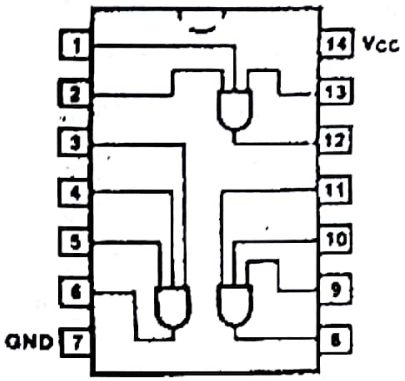
54/7406
HEX INVERTER BUFFER/ DRIVER
(WITH OPEN-COLLECTOR
A HIGH- VOLTAGE OUTPUT)



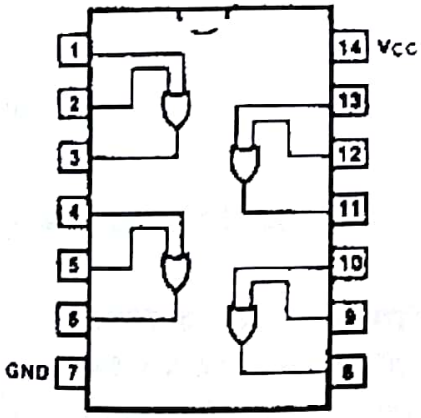
54/7408
54H/74H08
54S/74S08
54LS/74LS08
QUAD 2-INPUT AND GATE



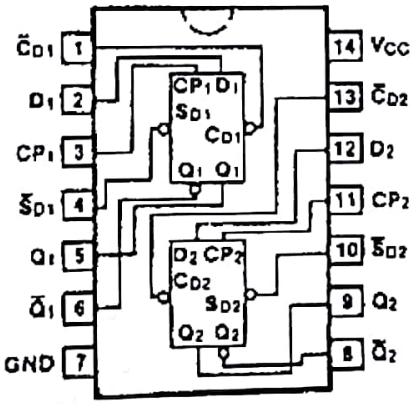
54/7411
54H/74H11
54S/74S11
54LS/74LS11
TRIPLE 3-INPUT AND GATE



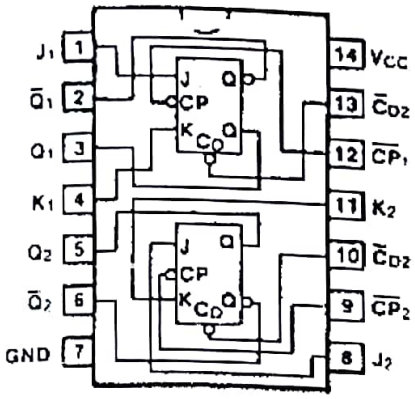
4/7432
4S/74S32
4LS/74LS32
AD 2-INPUT OR GATE



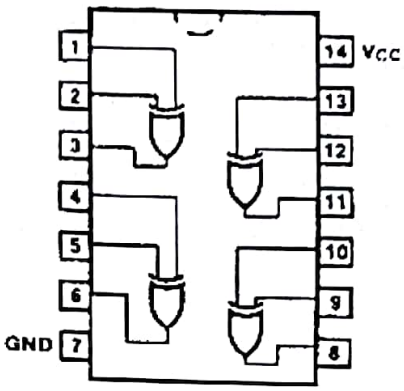
54/7474
54H/74H74
54S/74S7
54LS/74LS74
DUAL D-TYPE POSITIVE
EDGE TRIGGERED FLIP FLOP



54/74107
54LS/74LS107
DUAL JK FLIP FLOP
(WITH SEPARATE
CLEARSAND CLOCKS)



54LS/74LS86
QUAD 2-INPUT EXCLUSIVE OR GATE



OPTIONAL-

Experimental Module Boards-

1. BASIC logic gates (using digital IC's, diodes & Transistor)
2. Astable, Monostable, Bistable, RS Flip Flop
3. Boolean algebric Eqn & Demorgans Theorem
4. ALU using 74181
5. 8x1 Multiplexer, 1x8 demultiplexer, 8x3 encoder 3x8 decoder
6. RS, D, Type flip flop Trainer
7. JK, T, M/S Type JK flip flop Trainer
8. Half/Full adder- Substractor Trainer
9. 4 bit Asynchronus up/down & BCD counter using 7473
10. Modulo-n-counter (n=2 to 15)
11. 4 bit decade counter & 4 bit shift register
12. A to D Converter Trainer
13. D to A Converter Trainer
14. PISO, SIPO,SISO, PIPO Trainer
15. Seven Segment Decoder Trainer