**DEPARTMENT OF INFORMATION AND TECHNOLOGY**

**DIGITAL CIRCUITS AND SYSTEMS**

**LIST OF EXPERIMENTS**

| S.No | Name of Experiment | Equipment used for performing |
|---|---|---|
| 1 | To study & test of operation of all Logic gates for various IC's | Logic gates kit, Bread board |
| 2 | Implementation of all logic gates by NAND & NOR universal gates. | Digital full adder & sub using Nand gate, Bread board |
| 3 | Binary addition by Half adder & Full adder circuits. | 4-bit adder &subtractor kit |
| 4 | Binary subtraction by Half subtractor& Full subtractor circuits. | 4-bit adder &subtractor kit |
| 5 | Design of BCD to Excess-3 code converter. | BCD to Excess-3 trainer kit |
| 6 | Verification of Demorgan's Theorem. | Logic gate trainer kit |
| 7 | Study of RS,JK, T & D Flip Flop. | Flip Flop using Nand Gate |
| 8 | Multiplexer/Demultiplexer based Boolean function realization. | Mux/Demux trainer kit |
| 9 | Study & Applications of 555 Timer(Astable, Monostable, Schmitt trigger, VCO) | Bread board |

**EXPERIMENTS BEYOND THE CURRICULUM**

| S.No | Name of Experiment | Equipment used |
|---|---|---|
| 1 | Implementation of PLA & PAL Memories. | PLA & PAL (lab electronics)trainer kits |

## DCS

## Experiment No-1

**Aim: - To study and test of operation of all logic gates for various IC's (IC#7400, IC#7403, IC#7408, IC#74332, IC#7486**

**Apparatus Required:-Logic gate Trainer Kit**

**Theory:-**

**Logic gates**:-

Logic circuits that perform the logical operation of AND, OR & NOT are called gates. These gates produce a Logic 1 & Logic 0 o/p signal and thus know as a Logic gates. These gates are AND, OR, NOT, NAND, NOR, EXOR and EXNOR gates.

**(1). AND gate**:-The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot ( **.**) is used to show the AND operation i.e. A.B

**(2).OR gate:-**The OR gate is an electronic circuit that gives a high output (1) if **one or more**
of its inputs are high. A plus (+) is used to show the OR operation.

**(3).NOT gate:-** The NOT gate is an electronic circuit that produces an inverted version of the
input at its output.It is also known as an *inverter*. If the input variable is A, the inverted output is known as NOT A

**(4)NAND gate**: -The *NAND gate* operates as an AND gate followed by a NOT gate.
It acts in the manner of the logical operation "and" followed by negation. The output is "false"
if both inputs are "true." Otherwise, the output is "true."

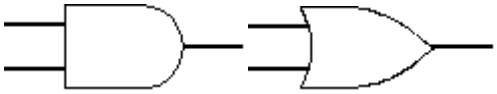
**(5).NOR gate** :-This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if **any** of the inputs are high.
The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

**(6).EXOR gate**:-The '**Exclusive-OR**' gate is a circuit which will give a high output if **either, but notboth**, of its two inputs are high. An encircled plus sign (+) is used to show the EXOR operation.

**(7).EXNOR gate:-**The '**Exclusive-NOR'** gate circuit does the opposite to the EXOR gate. It will give a low output if **either, but not both**, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

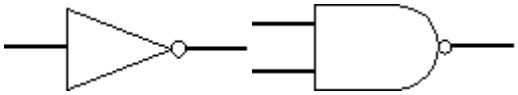
**Table 1: Logic gate symbols**

(1) **AND gate**



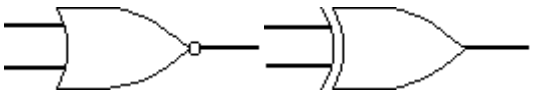
(2) **OR gate**

**(3) Inverter or NOT gate**



(4) **NAND gate**

**(5)NOR gate**



**(6) XOR gate**

(7) **XNOR gate**

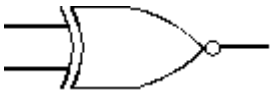


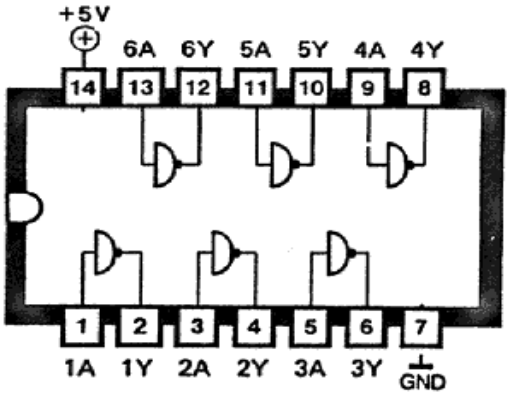
**Table 2: Logic gates representation using the Truth table**

| NOT gate | |
|---|---|
| A | $\overline{A}$ |
| 0 | 1 |
| 1 | 0 |

| INPUTS | | OUTPUTS | | | | | |
|---|---|---|---|---|---|---|---|
| A | B | AND | NAND | OR | NOR | EXOR | EXNOR |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

**1.AND GATE-(IC-7411)**

This operation is represented by a dot. The logic operation AND is interrupted by means that z=1 if and only if x and y =1 ; otherwise z = 0

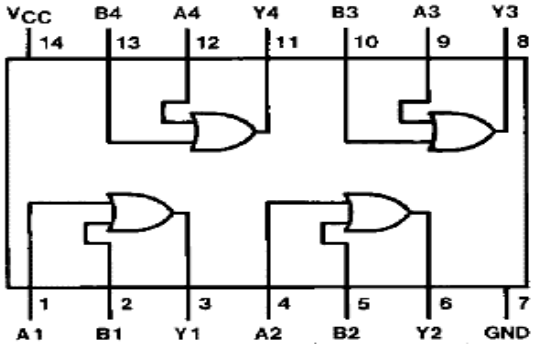



| X | Y | F=X.Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth table for AND Gate

**2. OR GATE-(IC-7412)**

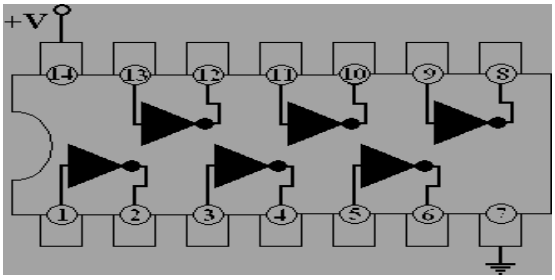
This operation is represented by '+' sign . It means that output is high in case any one of the I/Ps is high




| X | Y | F=X+Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Truth table for OR Gate

**3. NOT GATE- (IC-7404)**This operation is represented by a prime(some time a bar) .X′ =Z (X=Z) is read "X is not equal to Z" , means if x=0 ,z=1 and vice versa




| X | F=X' |
|---|---|
| 1 | 0 |
| 0 | 1 |

Truth table for NOT Gate

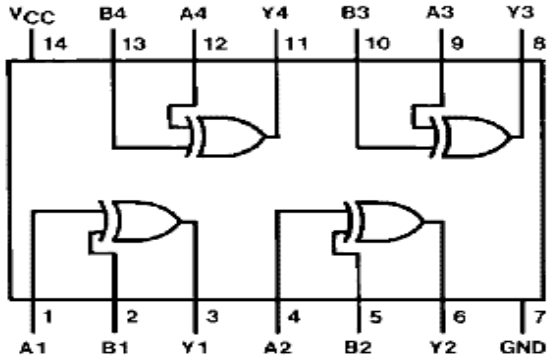
**4. EXCLUSIVE OR GATE:**

The EX-OR operation gives high output if one of the inputs is high. The Boolean for XOR operation can be written as

$$Y = X + Y$$

$$= X'Y + XY'$$

This operation can be implemented using basic AND, OR and invert gates. The symbol and truth table for XOR gate is given in Fig.(a) and (b) . The pin diagram of popular XOR gate IC is shown in fig.




| X | Y | F=x'y+xy' |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

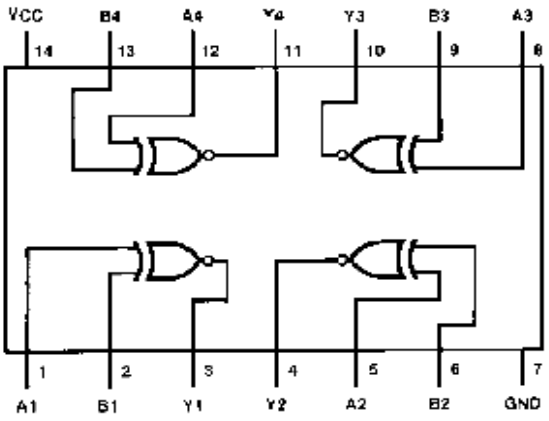
Truth table for XOR Gate

**5. EXCLUSIVE NOR GATE**:

The EX-NOR operation give high output for both inputs low and both inputs high. The Ex-NOR is written as

**F=X'Y'+XY**

This operation is implemented using basic AND, OR, and invert gates .The basic logic circuit and symbol for the EX-NOR are shown in fig.




| X | Y | F=x'y'+xy |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth table for XNOR Gate

**PROCEDURE :**

1. Connect the gates as shown in Figure
2. Set the input switches according to the truth tables as shown in the first row.
3. Monitor the output for the proper indication. If the light is off. Place a 0 under column Z and write"off" in the LED column. If the light is ON, place a 1 under column Z and write "on" in the LED column.
4. Repeat steps 2 and 3 for the remaining rows of the truth tables.
5. Repeat the above steps for verifying the truth tables in the case of a 3-input AND gate by selecting 3input switches and 073 monitoring the output on the Lamp out.

**RESULT: The** Truth table for all the logic gates is verified

**CONCLUSION:**

1.AND GATE- Output is high when all the inputs are high
2.OR GATE- Output is high when any input is high.
3.EX-OR GATE –Output is high when number of input 1's are odd.
4.EX-NOR - Output is high when number of input 1's are even.

**VIVA QUES.**

**1.What is the no system.**
**2.What is the base or radix of the number system.**
**3.State various number system.**
**4.Why is binary number system used in digital system?**
**5.What do you mean by a bit?**
**6.What do you mean by a nibble?**
**7.What do you mean by a byte?**
**8.Define a word.**
**9.What do you mean by word length?**
**10. What are logic gates.**

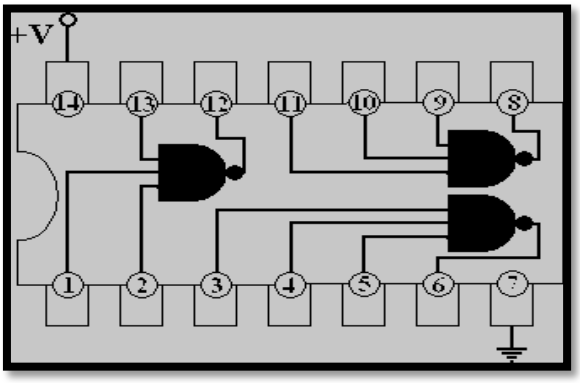
## Experiment No-2

**Aim**: - **Implementation of all logic gates by NAND & NOR universal gates.**

**Apparatus Required: - Digital full adder & sub using Nand gate**

**Theory**:-

NAND GATE-(IC-7450):-NAND function is complement of AND and is an abbreviation for "AND". Its graphic symbol which consists of an AND graphic symbol following by a small circle. The NAND & NOR gates are extensively used as standard logic gates and are in fact far more popular than AND or OR gates, NAND and NOR gates are easily constructed with transistor circuits and because Boolean function can be easily implemented with them, that is why are popularly know as universal gates.

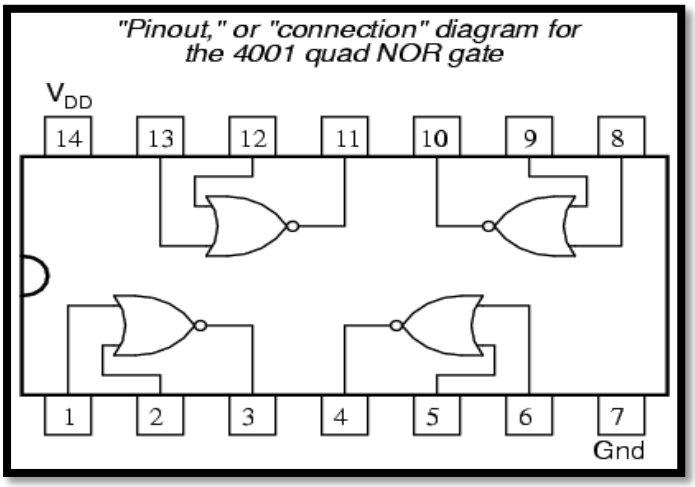



| X | Y | F= X.Y | F'=(X.Y)' |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**Truth table for NAND Gate**

NOR GATE( IC-7427):- The NOR function is the complement of the OR function and its name an OR graphic symbol followed by a small circle.

The NAND & NOR gates are extensively used as standard logic gates and are in fact far more popular than AND or ORgates , This is become NAND and NOR gates are easily constructed with transistor circuits and because Boolean function can be easily implemented with them, that is why are popularly know as universal gates.




| X | Y | F=X+ Y | F'=(X.Y)' |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

**Truth table for NOR Gate**

**PROCEDURE:**

1. Connect the gates as shown in Figure
2. Set the input switches according to the truth tables as shown in the first row.
3. Monitor the output for the proper indication. If the light is off. Place a 0 under Column Z and write "off" in the LED column. If the light is ON, place a 1 under Column Z and write "on" in the LED column.
4. Repeat steps 2 and 3 for the remaining rows of the truth tables.
5. Repeat the above steps for verifying the truth tables in the case of a 3-input AND gate by selecting 3input switches and 073 monitoring the output on the Lamp out.

**RESULT:** The Truth tables for NAND & NOR GATES are verified

**PRECAUTIONS:**

1. The connections should be made properly
2. The trainer kit should be handled properly

**VIVA QUES**

**1.How 1's complement of a binary number can be obtained?**
**2.How to obtain 2's complement of a binary number?**
**3.What is the main advantage of octal and hexadecimal system?**
**4.What are binary codes? how they are classified?**
**5. What are complements?**
**6.What are the basic operations in Boolean algebra?**
**7.What is the simplest technique for detecting error?**
**8.Why NAND & NOR are known as universal gates?**

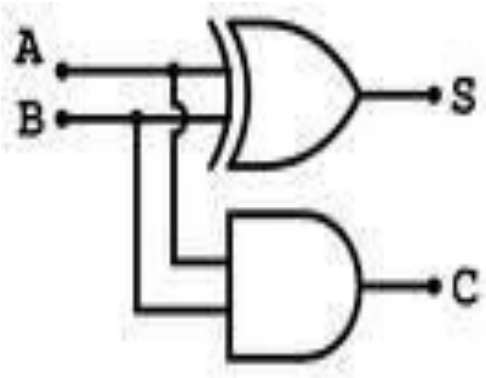
## Experiment No-3

**Aim**: - **To construct half adder and full adder**

**Apparatus required:** - **4-bit adder &subtractor kit**

**Theory**:-

**Half Adder**:-It is a combinational circuit that performs addition of two bits. This circuit has two inputs and two outputs. . Two of the inputs variables denoted by A & B represent the two significant bits to be added. The two outputs are designated as the symbols"S" for sum &"C" for carry. The binary variable S gives the value of least significant bit of the sum. The binary variable C gives the output carry.




| Input | | Output | |
|---|---|---|---|
| | | Carry | Sum |
| A | B | C | S |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

*OBSERVATION*

**Truth table for half adder**

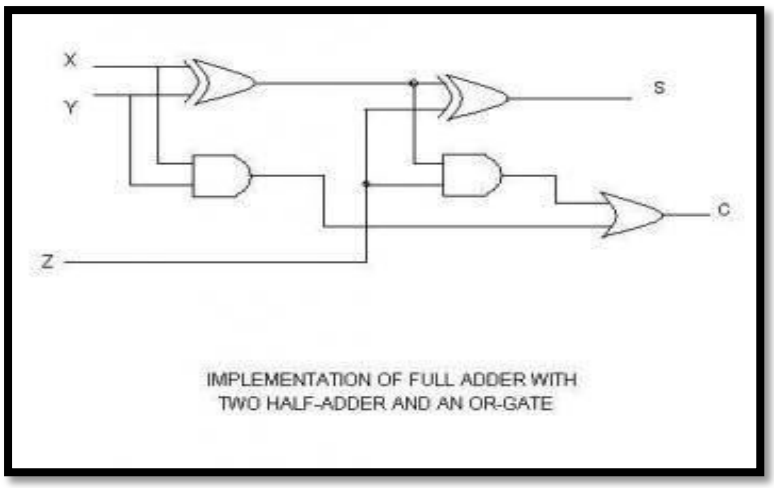
**Full Adder**:-A full adder is a combinational circuit that forms arithmetic sum of three input bits. It consists of three inputs & two outputs. Two of the inputs variables denoted by X&Y represent the two significant bits to be added. The third input Z represents the carry from the previous lower significant position. The need for the two outputs arises because the arithmetic sum of three binary digits ranges in the value from 0-3 and binary 2&3 need two digits. The two outputs are designated as the symbols"S" for sum &"C" for carry. The binary variable S gives the value of least significant bit of the sum. The binary variable C gives the output carry.

**the truth table logical expression for S & C outputs are:**

$$S = A \oplus B$$

$$S=A'B+AB'$$

$$C=AB$$




IMPLEMENTATION OF FULL ADDER WITH TWO HALF-ADDER AND AN OR-GATE

| X | Y | Z | C | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**Truth table for full adder**

**From the truth table logical expression for S & C outputs are:**

$$S = X \oplus Y \oplus Z$$

$$S=XY'Z'+X'YZ' +XYZ+X'Y'Z$$

$$C=XY+XZ+YZ$$

**PROCEDURE:**

1. Connect the gates as shown in Figure
2. Set the input switches according to the truth tables as shown in the first row.
3. Monitor the output for the proper indication. If the light is off. Place a 0 under column Z and write"off" in the LED column. If the light is ON, place a 1 under column Z and write "on" in the LED column.
4. Repeat steps 2 and 3 for the remaining rows of the truth tables.
5. Repeat the above steps for verifying the truth tables

**RESULT :**The Truth table for Full Adder and Half Adder is verified

**PRECAUTIONS:**

1. The connections should be made properly
2. The trainer kit should be handled properly

**VIVA QUES.**

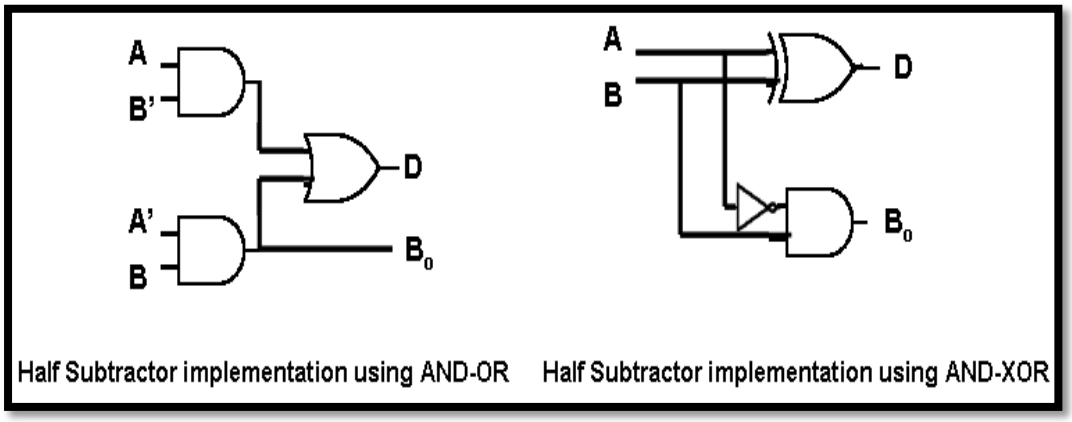
**1. What are arithmetic circuits?**
**2. What is Half adder?**
**3. What is Full adder?**
**4. What is carry propagation delay of a adder?**
**5. What is serial adder?**

## Experiment No-4

**Aim**: - **To construct of half subtractor and full subtractor circuits**

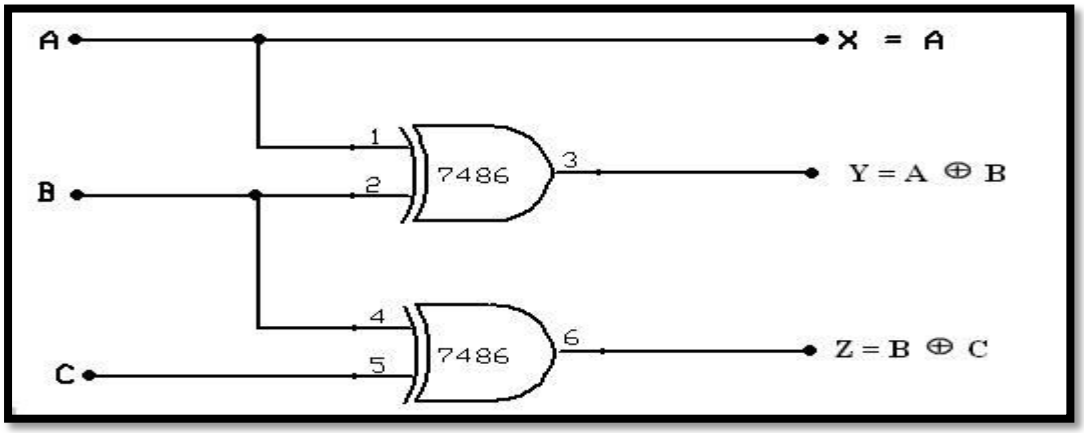
**Apparatus Required:-4-bit adder &subtractor kit**

**Theory**:-**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.




**Difference = A ⊕ B**
**Borrow=A'.B**

**FULL SUBTRACTOR:**
A combinational circuit which performs the subtraction of three input bits is called full subtractor. The three input bits include two significant bits and a previous borrow bit. A full subtractor circuit can be implemented with two half subtractors and one OR gate.




**Difference, DIFF= A'B'C + A'BC' + AB'C' + ABC**
**Borrow, BORR = A'BC + AB'C + ABC' +ABC**

**HALF SUBTRACTOR**

**TRUTH TABLE:**

| S.No | INPUT | | OUTPUT | |
|---|---|---|---|---|
| | A | B | DIFF | BORR |
| 1. | 0 | 0 | 0 | 0 |
| 2. | 0 | 1 | 1 | 1 |
| 3. | 1 | 0 | 1 | 0 |
| 4. | 1 | 1 | 0 | 0 |

## FULL SUBTRACTOR

**TRUTH TABLE:**

| S.No | INPUT | | | OUTPUT | |
|---|---|---|---|---|---|
| | A | B | C | DIFF | BORR |
| 1. | 0 | 0 | 0 | 0 | 0 |
| 2. | 0 | 0 | 1 | 1 | 1 |
| 3. | 0 | 1 | 0 | 1 | 1 |
| 4. | 0 | 1 | 1 | 0 | 1 |
| 5. | 1 | 0 | 0 | 1 | 0 |
| 6. | 1 | 0 | 1 | 0 | 0 |
| 7. | 1 | 1 | 0 | 0 | 0 |
| 8. | 1 | 1 | 1 | 1 | 1 |

**RESULT :**The Truth table for Full subtractor and Half subtractor is verified

**PRECAUTIONS:**

1. The connections should be made properly
2. The trainer kit should be handled properly

**VIVA QUES.**

**1.Why serial adders are slower than parallel adders?**
**2.How does look ahead carry adder speed up addition process?**
**3.What do you mean by serial data?**
**4.What is Half subtractor?**
**5.What is Full subtractor?**
**6.What are the applications of adders?**
**7.What are the applications of subtractors?**
**8.Realize a full adder using two half adders**
**9.Realize a full subtractors using two half subtractors**

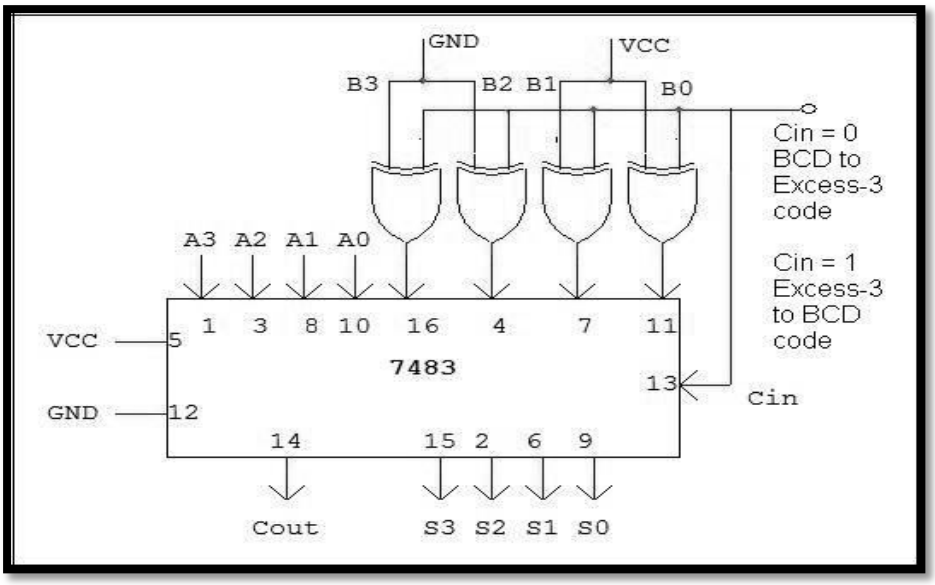
## Experiment No-5

**Aim**: - **Design of BCD to Excess-3 code converter.**

**Apparatus Required:** - **BCD to Excess-3 trainer kit**

**Theory**:-
Code converter is a combinational circuit that translates the input code word into a new correspondingword.The excess-3 code digit is obtained by adding three to the corresponding BCD digit. To Construct a BCD-to-excess-3-code converter with a 4-bit adder feed BCD-code to the 4-bit adder as the first operand and then feed constant 3 as the second operand. The output is the corresponding excess-3 code.To make it work as a excess-3 to BCD converter, we feed excess-3 code as the first operand and then feed 2's complement of 3 as the second operand. The output is the BCD code.

CIRCUIT DIAGRAM




TRUTH TABLE:

**i) BCD - EXCESS-3 CODE**

| BCD | | | | EX-3 | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

**ii)EXCESS-3 TO BCD CODE**

| EX-3 | | | | BCD | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

**PROCEDURE:**

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Apply BCD code as first operand(A) and binary 3 as second operand(B) and cin=0 for

**RESULT: Realized** BCD code to Excess-3 code conversion and vice versa using 7483 IC

**VIVA QUESTIONS:**

**1) What is the internal structure of 7483 IC?**
**2) What do you mean by code conversion?**
**3) What are the applications of code conversion?**
**4) How do you realize a subtractor using full adder?**
**5) What is a ripple Adder? What are its disadvantages?**

## Experiment No-6

**Aim**: - **Verification of Demorgan's Theorem.**

**Apparatus Required:** - **Logic gate Trainer Kit**

**Theory**:-
De Morgan's Theorem was created by Augustus De Morgan, a 19th-century mathematician who developed many of the concepts that make Boolean logic work with electronics.

**DeMorgan's Theorems:**
a. (A + B) = A* B
b. A*B = A + B
Note: * = AND operation

**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'

**RESULT:** Demorgans law has been verified

**PRECAUTIONS:**

1. The connections should be made properly
2. The trainer kit should be handled properly

**VIVA QUESTIONS:**

1) **What are the different methods to obtain minimal expression?**
2) **What is a Min term and Max term**
3) **State the difference between SOP and POS.**
4) **What is meant by canonical representation?**
5) **State Demorgan's theorem.**

**6) What are universal gates?**

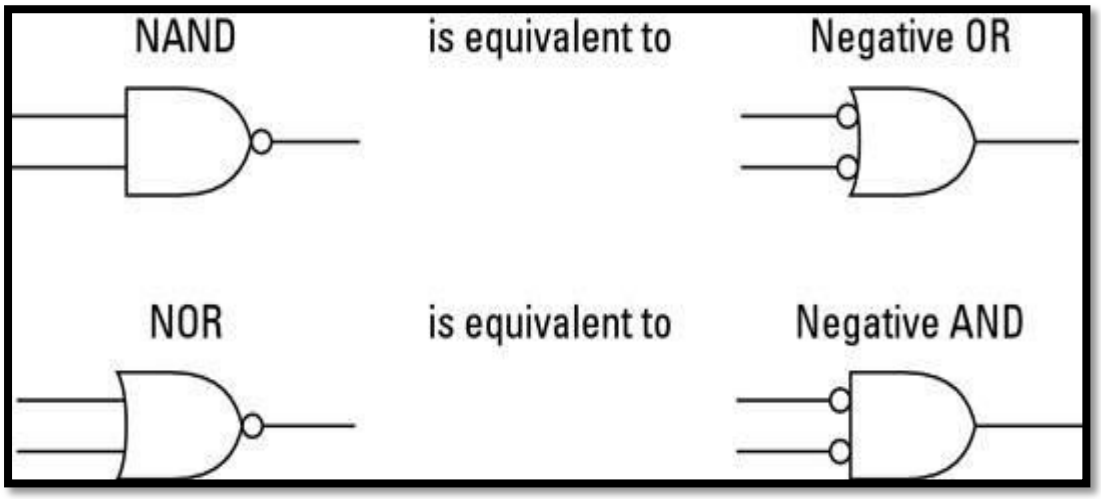
**Proof by Tabular method**

| A | B | A.B | (A.B)’ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

| A | B | A’ | B’ | (A’+B’) |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

| A | B | A+B | (A+B)’ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

| A | B | A’ | B’ | A’.B’ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |



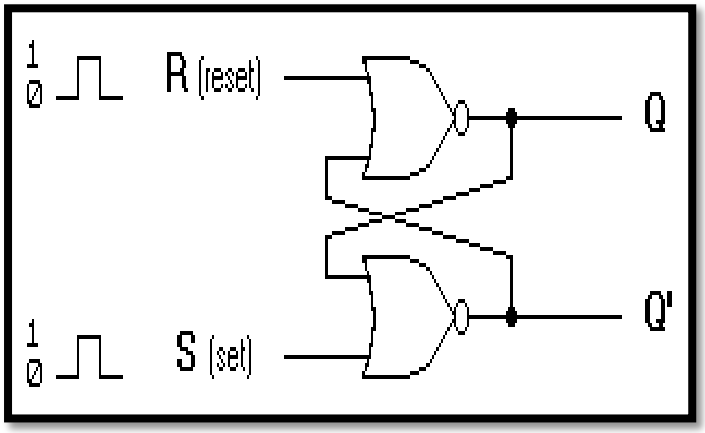
## Experiment-No-7

**Aim**: - **Study of RS,JK, T & D Flip Flop.**

**Apparatus Required:** - **Flip Flop using Nand Gate**

**Theory**:-

**1) Basic flip flop** : - A flip flop can be constructed from two NAND gates or two NOR gates .the cross coupled connections from the o/p of one gate be i/p of another constitute the feedback path ,each flip flop has two o/p Q & Q' and two i/p set and reset ,the type of flip flop is sometimes called direct coupled RS flip flop or SR latch.In basic flip flop circuit with NOR gate we must remember that o/p is 1 only when all i/p are 0. as starting point is assume that set i/p is 1 and reset i/p is 0, since gate 2 has an i/p of 1, it's o/p must be 0,its o/p Q' must be 0, which puts both input of gates to the o/p Q is 1, when the set i/p is returned to 0,the o/p remains the same at one i/p at gate 2 at 1.That causes o/p Q to stay at 0 which leaves both i/p of gates no. 1 at 0.So Q is at 1. A f-f has two useful states when Q =1 and Q'=0 ,it is in the set state . When Q =0&Q'=1 it is clear state .Under normal conditions both i/p remains at 0 unless the state of the flip flop has to be changed. The application of momentary 1 to the set i/p causes the flip flop to go to set state .The set state i/p must go back to 0 before 1 is supplied to reset i/p .The NAND basic flip flop circuit . Operates with both i/p normally at 1 under the state of f-f has to be changed. The application at momentary 0 to set i/p has o/p Q to go to 1 and Q' to 0. The flip flop in the set state .After the set i/p causes a transition to the clear state when both i/p go to 1, a condition avoided in the f-f operation .



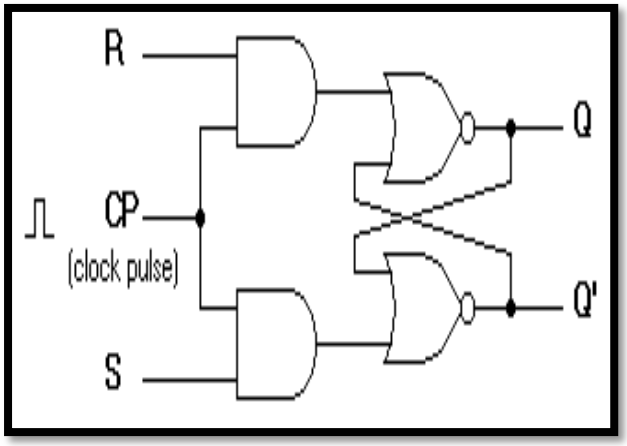

Logic diagram

| S | R | Q | Q' | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 0 | (after S=1, R=0) |
| 0 | 1 | 0 | 1 | |
| 0 | 0 | 0 | 1 | (after S=0, R=1) |
| 1 | 1 | 0 | 0 | |

Transition table

**2) Clocked RS Flip Flop :** By adding Gates to the i/p of the basic circuit. of the flip flop can be made to respond to the i/p level during the occurrence clock pulse. This f-f consist of basic NOR and two NAND gates the o/p of the two AND gates remains at 0 as long as the clock pulse is 0 regardless of S & R. i/p values when clock pulse goes to 1 information from S & R i/p is allowed to reach the basic f-f , the set state is 1

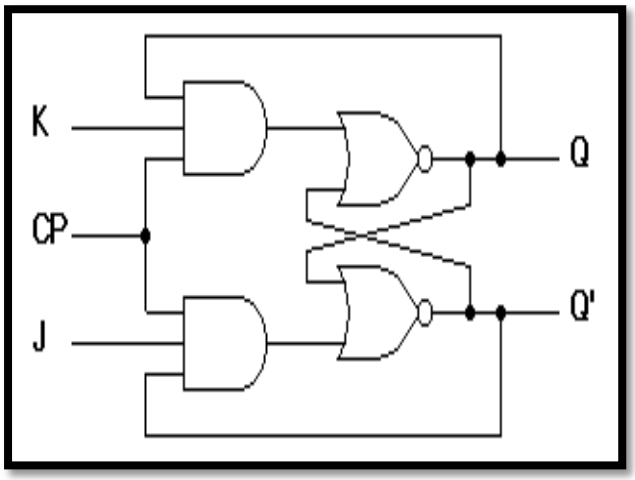
with set R=0 & CP = 1.To change to clear state the i/p must be S=0, R=1 & CP =1, with both S = 1, R =1, the occurrence of clock pulse causes both i/p to go 0.
Q(t+1)=S+R'Q
It is an algebraic exp. for binary information of characteristics tables




| Q | S | R | Q(t+1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | indeterminate |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | indeterminate |

**Logic diagram** **Transition table**

**J K Flip Flop :** J K Flip Flop is a refinement of RS Flip Flop in that the indeterminate state of RS type is defined in JK type. Input J & K Behaves like input S & R. to set and clear Flip Flop when inputs are applied to both J & K Simultaneously Flip Flop switches to its complementary State. In this type of Flip Flop output Q is AND with Clock pulse, inputs so that the i/p are available to the Flip Flop only when Q and CP both are 1.




| 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 |

**Logic diagram** **Transition table**

**PROCEDURE:**
1.Connect the input terminal of flip flop to high and low terminal of the logic circuit .
2.Connect the output terminal to LED.
3.Switch on the kit .
4.Vary the input (high/low) in the input terminals of various Flip flops.

5.Make the verification of truth table.

**T flip-flop (Triggered / Toggle)**

The T type flip-flop is a single input device: T (trigger). Two outputs: Q and Q' (where Q' is the inverse of Q).The operation of the T type flip-flop is as follows: A '0' input to 'T' will make the next state the same as the present state (i.e. T = 0 present state = 0 therefore next state = 0). However a '1' input to 'T' will change the next state to the inverse of the present state (i.e. T = 1 present state = 0 therefore next state = 1).The T type flip-flop is an edge driven device. Therefore you should not associate 1 and 0 with levels, but instead 1 should be considered as a pulse, and 0 as no pulse.

## D type flip-flop (Delay)

The D type flip-flop has one data input 'D' and a clock input. The circuit edge triggers on.the clock input. The flip-flop also has two outputs Q and Q' (where Q' is the reverse of Q).

The operation of the D type flip-flop is as follows: Any input appearing (present state) at the input D, will be produced at the output Q in time T+1 (next state). e.g. if in the present state we have D = 0 and Q = 1, the next state will be D = anything and Q = 0.

.

**RESULT:**
Truth table for RS ,JK,T & D flip-flop is verified and the result was checked through LED'S.

**PRECAUTIONS:**
1.The connections should be made properly
2.The trainer kit should be handled properly

**VIVA QUESTIONS:**

1. **What is the difference between Flip-Flop & latch?**
2. **Give examples for synchronous & asynchronous inputs?**
3. **What are the applications of different Flip-Flops?**
4. **What is the advantage of Edge triggering over level triggering?**
5. **What is the relation between propagation delay & clock frequency of flip-flop?**
6. **What is race around in flip-flop & how to over come it?**
7. **Convert the J K Flip-Flop into D flip-flop and T flip-flop?**
8. **List the functions of asynchronous inputs?**
9. **What do you mean by Toggling?**

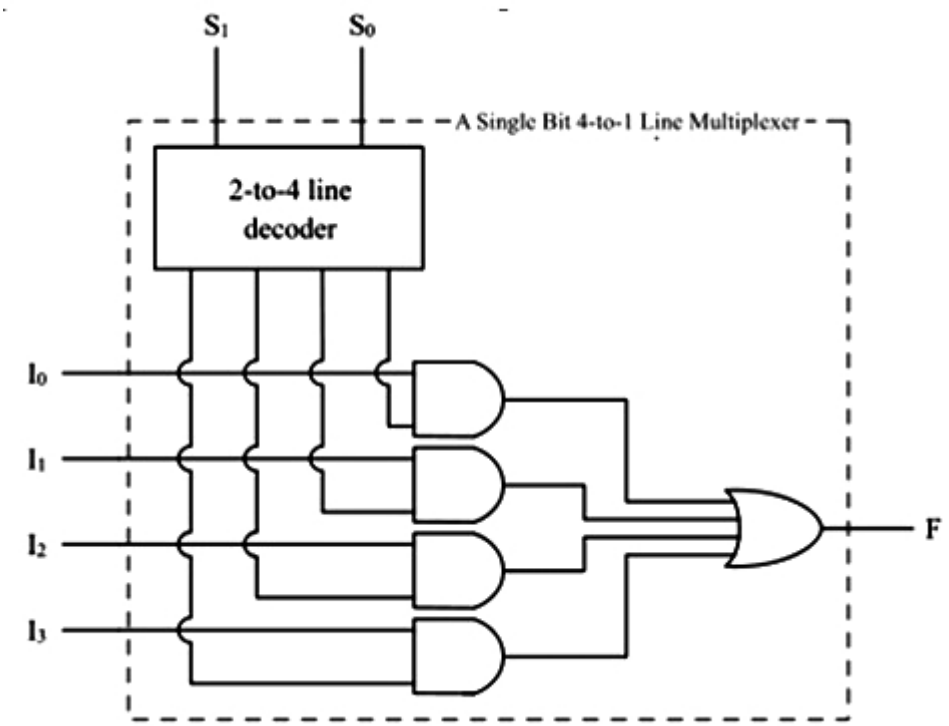
## Experiment-No-8

**Aim**: - **Multiplexer/Demultiplexer based Boolean function realization.**

**Apparatus Required:** - **Mux/Demux trainer kit**

**Theory**:-

Multiplexer means transmitting a large no. of information unit over a small no. of channels or lines. A digital multiplexer is a combinational circuit. That receives binary information from one of $2^n$ input data lines & directs it to a Single output lines. the selection of a particular input data line for output is Determine by a set of selection input. A $2^n$ x 1 multiplexer has $2^n$ input data lines & n input selection lines whose bit combinations determine which input data are selected for the output. Each of the four data inputs $I_0$ through $I_3$ is applied to one input of an and gate. The two selection inputs $s_1$ &$s_0$ are decoded to select a particular AND gate . The outputs of the AND gates are applied to single OR gate to provide the single output. The multiplexer has six inputs and one output. A multiplexer is also called data selectors, since it selects one of the many inputs and steers the binary information to the output line.




**Circuit Diagram**

# Truth Table

| $S_1$ $S_0$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | F |
|---|---|---|---|---|---|
| 0 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 1 |
| | 0 | 1 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 1 | 1 |
| | 0 | 1 | 1 | 0 | 0 |
| | 0 | 1 | 1 | 1 | 1 |
| | 1 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 1 |
| | 1 | 0 | 1 | 0 | 0 |
| | 1 | 0 | 1 | 1 | 1 |
| | 1 | 1 | 0 | 0 | 0 |
| | 1 | 1 | 0 | 1 | 1 |
| | 1 | 1 | 1 | 0 | 0 |
| | 1 | 1 | 1 | 1 | 1 |

| $S_1$ $S_0$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | F |
|---|---|---|---|---|---|
| 0 1 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 0 | 1 |
| | 0 | 0 | 1 | 1 | 1 |
| | 0 | 1 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 1 | 0 |
| | 0 | 1 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 1 | 1 |
| | 1 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 1 | 0 | 1 |
| | 1 | 0 | 1 | 1 | 1 |
| | 1 | 1 | 0 | 0 | 0 |
| | 1 | 1 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 0 | 1 |
| | 1 | 1 | 1 | 1 | 1 |

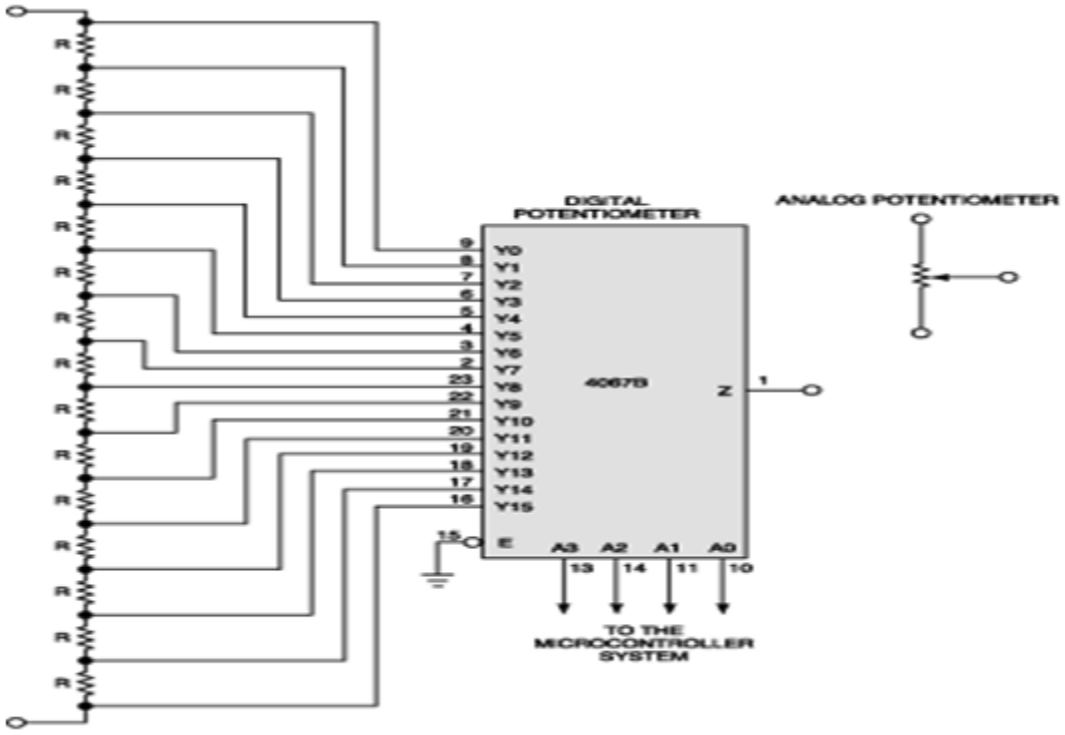
| $S_1$ $S_0$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | F |
|---|---|---|---|---|---|
| 1 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 0 |
| | 0 | 1 | 0 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 1 |
| | 0 | 1 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 1 | 1 |
| | 1 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 1 | 0 | 0 |
| | 1 | 0 | 1 | 1 | 0 |
| | 1 | 1 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 1 |
| | 1 | 1 | 1 | 0 | 1 |
| | 1 | 1 | 1 | 1 | 1 |

| $S_1$ $S_0$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | F |
|---|---|---|---|---|---|
| 1 1 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 0 |
| | 0 | 1 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 1 | 0 |
| | 0 | 1 | 1 | 0 | 0 |
| | 0 | 1 | 1 | 1 | 0 |
| | 1 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 1 | 1 |
| | 1 | 0 | 1 | 0 | 1 |
| | 1 | 0 | 1 | 1 | 1 |
| | 1 | 1 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 1 |
| | 1 | 1 | 1 | 0 | 1 |
| | 1 | 1 | 1 | 1 | 1 |

Demultiplexer is a digital device that directs information from a single input to one of several output which means one is to many. This enable to transmit the information on one of the 2n possible output lines .the decoder can function as Demultiplexer if e line is taken as data input line and line a and b are taken as selection lines. This is shown in figure of Demultiplexer. a decoder with an enable input is referred to as decoder/Demultiplexer. It is enable to input that makes a circuit a Demultiplexer.Decoder/Demultiplexer circuits can be connected together to form a larger decoder circuits. fig. shows two 3*8 decoders which enable inputs connected to form 4*16 decoders.

**PROCEDURE :**

1. Switch on the trainer and check the power supply .
2. Note that the supply voltage is +5v.
3. Study pin configuration of the 74150 circuit 7493 is provided mechanical switches which has been in the trainer to choose correct input (1 Hz) to pin 14 of 7493 and corresponding patch the output of 7493 ABCD TO THE INPUT OF 74150 .
4. The LED,s are connected to the 7493 counts from 0 to 15 the single output channel of the multiplexer is energized alternately the 16 inputs to 74150 IC can be given through logic switches also thus the trainer highlights how a 16 bit information is otherwise indicated as 16*1 multiplexing..
5. Study the pin configuration of 74154-ckt. The device consists of 16 counter which is provided in a trainer to give input to the Demultiplexer.
6. The clock input (1 Hz.) to pin 14 of 7493 and corresponding path the output of 7493 ABCD to the input of 74154 G1 and G2.the slope points of 74154 are to grounded
7. The LED's are connected to output of demultiplexer.As the 7493 count from 0-15 the output channel 1-16 of Demultiplexer are alternatively ABCD input can be given through logic switch also.
8. The trainer highlights has a 4 bit information is being transmitted through 16 channels which is indicated 4-16 Demultiplexer




**CIRCUIT DIAGRAM**

OBSERVATIONS:

The truth table obtained matches with function of logic gates and hence verified.
The truth table that has been prepared as per the requirements and connection made. All the functions of the multiple matches are verified as per the truth table.

RESULT:
Truth table for the MUX & DEMUX has been verified.

PRECAUTIONS:

1. The connections should be made properly
2. The trainer kit should be handled properly.

VIVA QUESTIONS:

1) What is a multiplexer?
2) What is a de-multiplexer?
3) What are the applications of multiplexer and de-multiplexer?
4) Derive the Boolean expression for multiplexer and de-multiplexer.
5) How do you realize a given function using multiplexer
6) What is the difference between multiplexer &demultiplexer?
7) In 2n to 1 multiplexer how many selection lines are there?
8) How to get higher order multiplexers?
9) Implement an 8:1 mux using 4:1 muxes?

## Experiment No-9

**Aim**: -**Study & Applications of 555 Timer(Astable, Monostable, Schmitt trigger, VCO)**

**Apparatus Required: - Bread board**

**Theory**:-

One of the most versatile linear ICs is the **555 timer** which was first introduced in early 1970 by Signetic Corporation giving the name as **SE/NE 555 timer**. This IC is a monolithic timing circuit that can produce accurate and highly stable time delays or oscillation. Like other commonly used op-amps, this IC is also very much reliable, easy to use and cheaper in cost. It has a variety of applications including **monostable** and **astable multivibrators**, **dc-dc converters**, digital logic probes, **waveform generators**, analog frequency meters and tachometers, **temperature measurement** and control devices, **voltage regulators** etc. The timer basically operates in one of the two modes either as a monostable (one-shot) multivibrator or as an astable (free-running) multivibrator.The **SE 555** is designed for the operating temperature range from – 55°C to 125° while the **NE 555** operates over a temperature range of 0° to 70°C.

**Pin 1**: **Grounded Terminal:** All the voltages are measured with respect to this terminal.

**Pin 2: Trigger Terminal:** This pin is an inverting input to a comparator that is responsible for transition of **flip-flop** from set to reset. The output of the timer depends on the amplitude of the external trigger pulse applied to this pin.

**Pin 3:Output Terminal:** Output of the timer is available at this pin. There are two ways in which a load can be connected to the output terminal either between pin 3 and ground pin (pin 1) or between pin 3 and supply pin (pin 8). The load connected between pin 3 and ground supply pin is called the normally on loadand that connected between pin 3 and ground pin is called the **normally off load**.

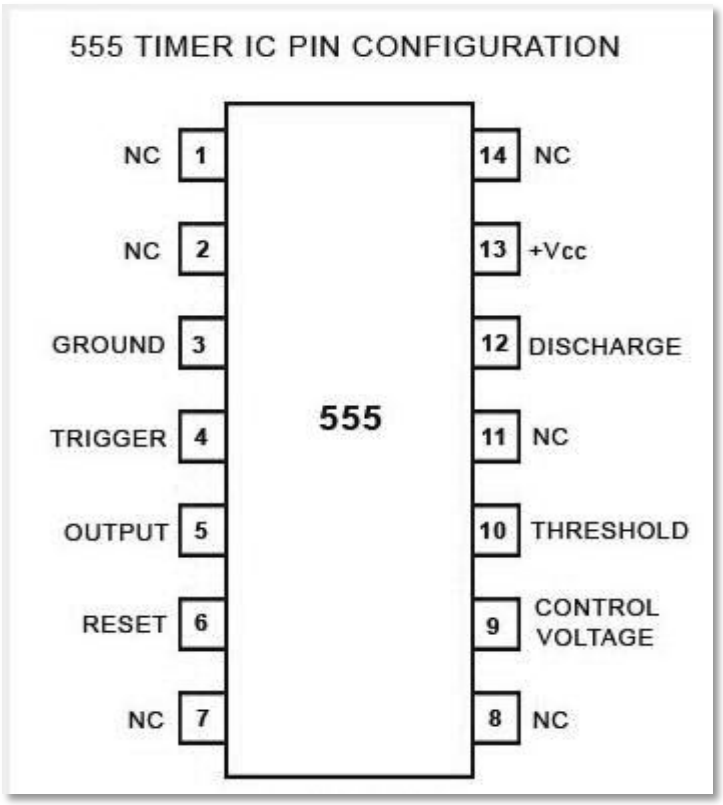
**Pin 4:Reset Terminal:** To disable or reset the timer a negative pulse is applied to this pin due to which it is referred to as reset terminal. When this pin is not to be used for reset purpose, it should be connected to + $V_{CC}$ to avoid any possibility of false triggering.

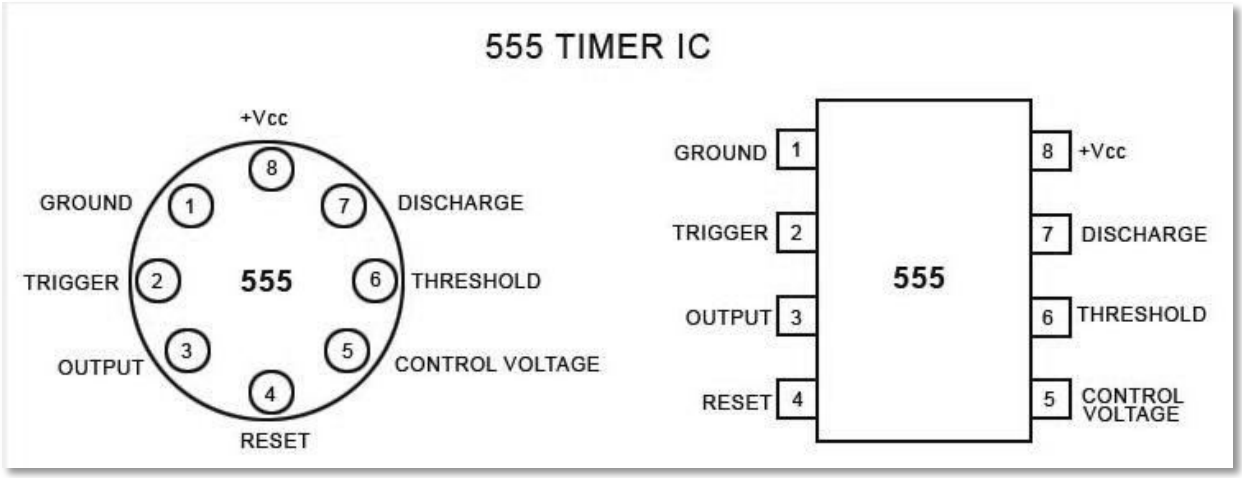
**Pin 5:Control Voltage Terminal:** The function of this terminal is to control the threshold and trigger levels. Thus either the external voltage or a pot connected to this pin determines the pulse width of the output waveform. The external voltage applied to this pin can also be used to modulate the output waveform. When this pin is not used, it should be connected to ground through a 0.01 micro Farad to avoid any noise problem.

**Pin 6**: **Threshold Terminal:** This is the non-inverting input terminal of comparator 1, which compares the voltage applied to the terminal with a reference voltage of 2/3 $V_{CC}$. The amplitude of voltage applied to this terminal is responsible for the set state of flip-flop.

**Pin 7** :**Discharge Terminal:** This pin is connected internally to the collector of transistor and mostly a capacitor is connected between this terminal and ground. It is called discharge terminal because when transistor saturates, capacitor discharges through the transistor. When the transistor is cut-off, the capacitor charges at a rate determined by the external resistor and capacitor.

**Pin 8:Supply Terminal:** A supply voltage of + 5 V to + 18 V is applied to this terminal with respect to ground (pin 1).



**RESULT:**
Study of IC-555 Timer has been completed.

**PRECAUTIONS:**
1.The connections should be made properly
2.The trainer kit should be handled properly.

**VIVA QUESTIONS:**

1.What is Astablemultivibrator?
2.What is Bistablemultivibrator?
3.What is Monostablemultivibrator?
4.What are the applications of IC-555 Timer.

## Experiment No-10

**Aim**: - **Implementation of PLA & PAL Memories.**

**Apparatus Required: - PLA & PAL (lab electronics)trainer kits**

**Theory**:-
**Read Only Memory (ROM)** - a fixed array of AND gates and a programmable array of OR gates
**Programmable Array Logic (PAL)** - a programmable array of AND gates feeding a fixed array of OR gates.
**Programmable Logic Array (PLA)** - a programmable array of AND gates feeding a programmable
array of OR gates

**READ ONLY MEMORY**
_ Read Only Memories (ROM) or Programmable Read Only Memories (PROM) have:
• N input lines,
• M output lines, and
• 2N decoded minterms.
_ Fixed AND array with 2N outputs implementing all N-literal minterms.
_ Programmable OR Array with M outputs lines to form up to M sum of minterm expressions.
_ A program for a ROM or PROM is simply a multiple-output truth table
• If a 1 entry, a connection is made to the corresponding minterm for the corresponding output
• If a 0, no connection is made
_ Can be viewed as a memory with the inputs as addresses of data (output values), hence ROM or
PROM names!

**PROGRAMMABLE LOGIC ARRAY (PLA)**
_ Compared to a ROM and a PAL, a PLA is the most flexible having a programmable set of ANDscombined with a programmable set of ORs.
_ Advantages
• A PLA can have large N and M permitting implementation of equations that are impractical
for a ROM (because of the number of inputs, N, required
• A PLA has all of its product terms connectable to all outputs, overcoming the problem of
the limited inputs to the PAL Ors
• Some PLAs have outputs that can be complemented, adding POS functions
_ Disadvantages
• Often, the product term count limits the application of a PLA.
• Two-level multiple-output optimization is required to reduce the number of product terms
in an implementation, helping to fit it into a PLA.
• Multi-level circuit capability available in PAL not available in PLA. PLA requires external
connections to do multi-level circuits.

**PROGRAMMABLE ARRAY LOGIC (PAL)**

_ The PAL is the opposite of the ROM, having a programmable set of ANDs combined with fixed ORs.

_ Disadvantage

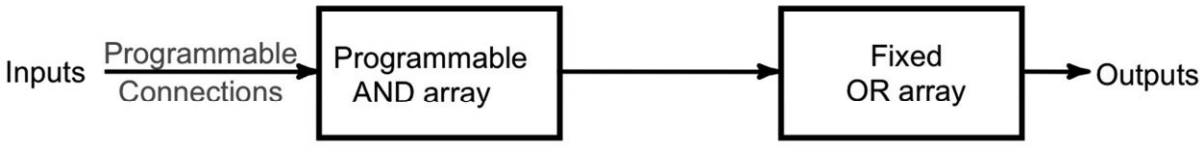
• ROM guaranteed to implement any M functions of N inputs. PAL may have too few inputs
to the OR gates.

_ Advantages

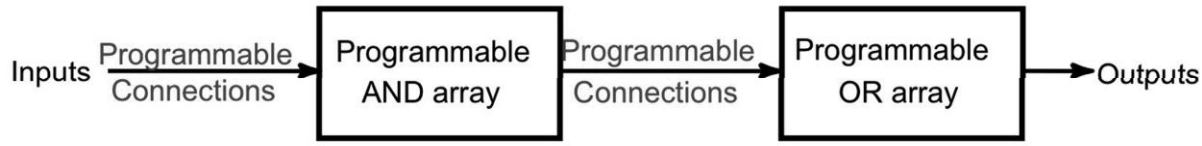
• For given internal complexity, a PAL can have larger N and M

• Some PALs have outputs that can be complemented, adding POS functions

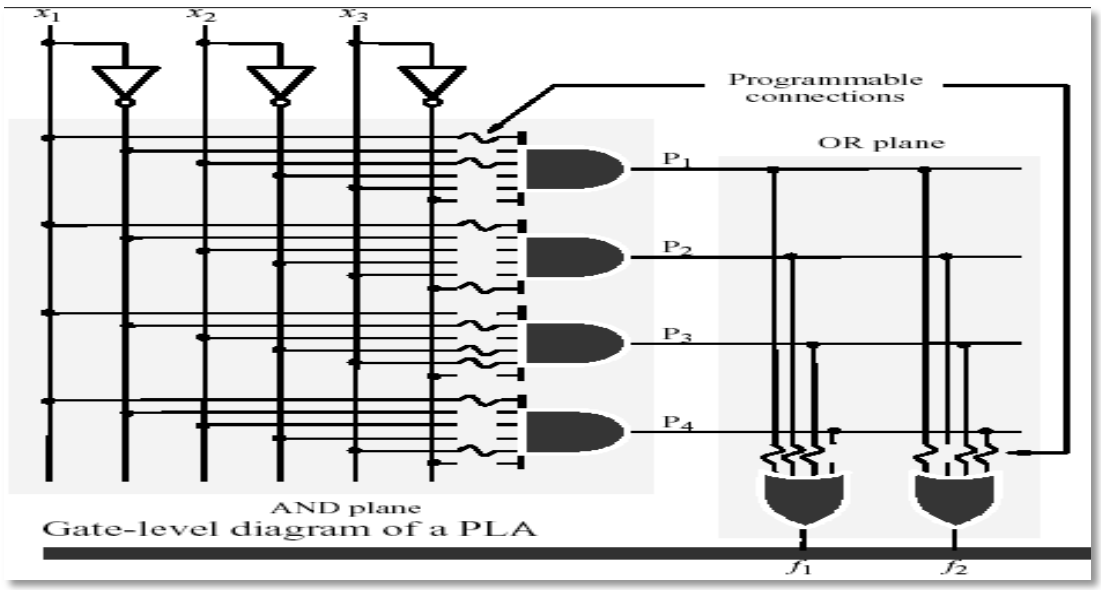
• No multilevel circuit implementations in ROM (without external connections from output
to input). PAL has outputs from OR terms as internal inputs to all AND terms, making implementation of multi-level circuits easier.

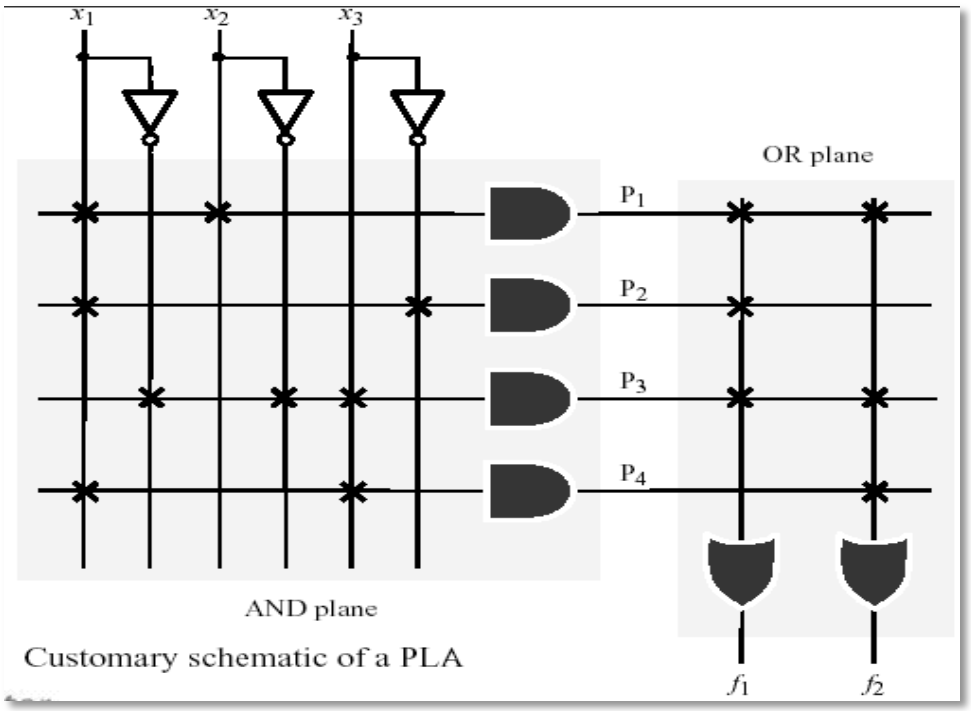



(b) Programmable array logic (PAL) device




(c) Programmable logic array (PLA) device





Customary schematic of a PLA

**RESULT:-**PLA & PAL Verified

**PRECAUTIONS:**

1.The connections should be made properly
2.The trainer kit should be handled properly.

**VIVA QUESTIONS:**

1) **What is memory?**
2) **What is program memory?**
3) **What is a storage element called?**
4) **What is ROM?**
5) **What is Volatile & Non Volatile memory?**
6) **What is PLA?**
7) **How size of PLA can be specified?**
8) **What are the applications of ROM?**
9) **What is PAL?**