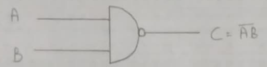
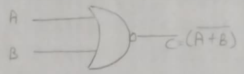
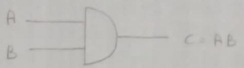
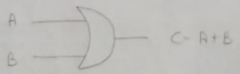
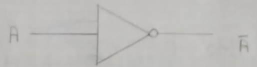
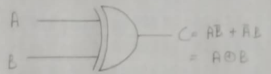
| S No | Gate | Symbol | Input A | Input B | Output C |
|---|---|---|---|---|---|
| 1. | NAND<br>IC7400 |  | 0 | 0 | 1 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 0 |
| 2. | NOR<br>IC7402 |  | 0 | 0 | 1 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 0 |
| 3. | AND<br>IC7408 |  | 0 | 0 | 0 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 1 |
| 4. | OR<br>IC7432 |  | 0 | 0 | 0 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 1 |
| 5. | NOR<br>IC7404 |  | 0 | – | 1 |
| | | | 1 | – | 0 |
| | | | – | – | |
| | | | – | – | |
| 6. | Ex OR<br>7486 |  | 0 | 0 | 0 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 0 |

Date 8/2/2023

Expt. No. 1

Page No. 1

# Realization of Functions Using Basic and Universal Gates ( SOP and POS Forms)

## AIM

1. To study & verify truth table of basic gates
2. To simplicy the given expression & to realize it using basic gates, NAND gate and NOR gate

$$f = \sigma\ m(2,6,8,9,10,11,14)$$
$$f = \Pi\ M(3,6,7)$$

## COMPONENTS REQUIRED

Logic Gates (IC), trainer kit, Connecting patch cords.
IC7400, IC 7408, IC ,7432, IC7486, IC7404, IC7410
IC7411, IC7402

## THEORY

The basic Logic gates are the building blocks of more complex logic circuits. These logic gates perform the basic Boolean functions, such as AND, OR, NAND, NOR, Inversion, Exclusive-OR, ExclusiveNOR. Fig below shows the circuit symbol, Boolean function, and truth table. Each gate has one or two binary inputs A&B and one binary output C. The small circles on the output of the circuit symbols designates the logic complement. The AND, OR, NAND and NOR gates can be extended to have more than two inputs.

Teacher's Signature

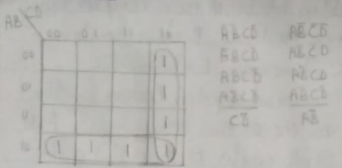
SOP : $F(A,B,C,D) = \Sigma m(2,6,8,9,10,11,14)$

POS : $F(A,B,C) = \Pi(3,6,7)$

Truth Table for SOP

| A | B | C | D | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

K Map

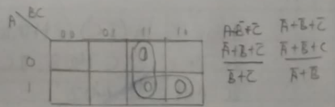


$F = A\bar{B} + C\bar{D}$

Truth Table for POS

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

K Map



$F = (\bar{A}+\bar{B}) \cdot (\bar{B}+\bar{C})$

A gate can be extended to have multible inbuts if the binary oberation it rebresents is commutative and associative.

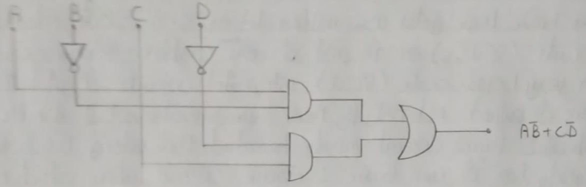
These basic logic gates are imblemented as small scale integrated circuits (SSICs) or as part of more comblex medium scale (MSI) or very large-scale (VLSI) integrated circuits. Digital IC gates are classified not only by their logic oberation, but also the sbecific logic-circuit family to which they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

TTL - Transistor-Transistor Logic
ECL - Emitter-Coubled Logic
MOS - Metal-Oxide Semiconductor
CMOS - Comblementary Metal-Oxide Semiconductor

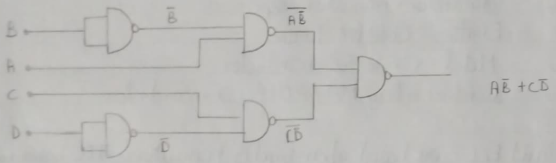
TTL and ECL are based ubon bibolar transistors. TTL has a well established bobularity among logic families. ECL is used only in systems requiring high-sbeed oberation. MOS and CMOS, are based on field effect transistors. They are widely used in large scale integrated circuits because of their high combonent density and relatively low bower consumbtion. CMOS logic consumes far less bower than MOS logic. There are various commercial integrated circuit chibs available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

## SOP Circuit

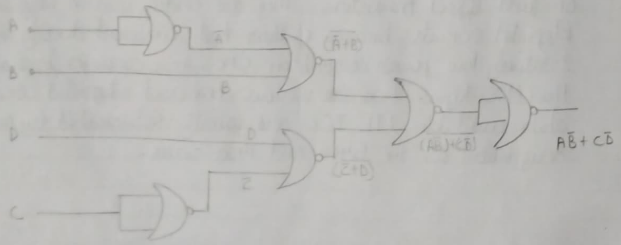
i) Using Basic Gates



ii) Using NAND Gates



iii) Using NOR Gates



## PROCEDURE

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

## REALIZATION OF BOOLEAN EXPRESSION

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

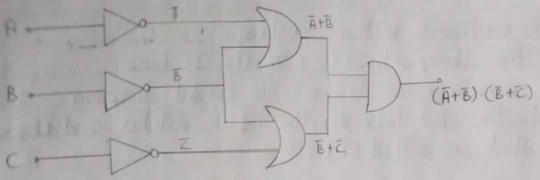
A boolean function can be represented by a Karnaugh map in which each cell corresponds to a minterm. The cells are arranged in such a way that any two immediately adjacent cells corresponds to two minterms of distance 1. There is more than one way to construct a map with this property.

SOP: $f(A,B,C,D) = \Sigma m(2,6,8,9,10,11,14)$
$= A\bar{B}+C\bar{D}$

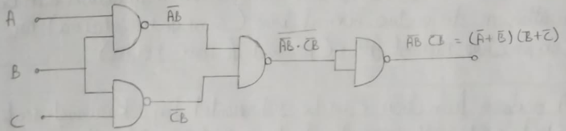
POS: $f(A,B,C) = \Pi(3,6,7)$
$= (B+\bar{C})\cdot(\bar{A}+\bar{B})$

## POS Circuit

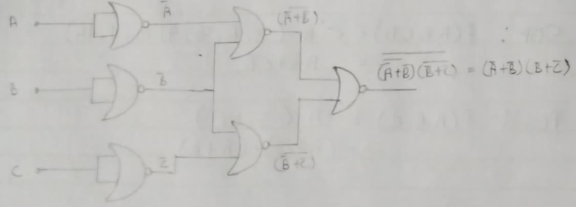
I) Using Basic Gates



II) Using NAND Gates


A
B
C
$\overline{AB}$
$\overline{CB}$
$\overline{AB}\cdot\overline{CB}$
$\overline{\overline{AB}\,\overline{CB}}$ = (Ā+B̄)(B+C̄)

III) Using NOR Gates


A
B
C
Ā
B̄
C̄
$\overline{(\bar A+\bar B)}$
$\overline{(\bar B+\bar C)}$
$\overline{\overline{(\bar A+\bar B)}\,\overline{(\bar B+\bar C)}}$ = (Ā+B̄)(B+C̄)

- Implementing the boolean functions using NAND gate

SOP : $f = A\bar{B} + C\bar{D}$ Taking Double Complement

$f = (\overline{\overline{A\bar{B} + C\bar{D}}})$ Demorgans Law

$f = (\overline{(\overline{A\bar{B}})(\overline{C\bar{D}})})$

POS : $F = (\bar{B}+\bar{C})\cdot(\bar{A}+\bar{B})$ Taking Double Complement

$F = ((\overline{\overline{\bar{B}+\bar{B}}) \cdot (\overline{\bar{B}+\bar{C}})})$ Demorgans Law

$F = (\overline{AB + BC})$

$F = (\overline{AB})(\overline{BC})$

- Implementing the boolean functions using NOR gates

SOP : $f = A\bar{B} + C\bar{D}$

$f = (\overline{\overline{A\bar{B} + C\bar{D}}})$ Taking Double compliment

$f = ((\overline{\overline{A\bar{B}})(\overline{C\bar{D}}}))$ Demorgans Law

$f = \overline{(\bar{A} + B)} + \overline{(\bar{C} + D)}$

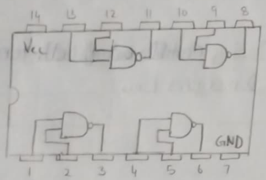
POS: $F = (\bar{A} + \bar{B})(\bar{B} + \bar{C})$ Taking Double complement

$F = ((\overline{\overline{(\bar{A}+\bar{B})(\bar{B}+\bar{C})}}))$ Demorgans Law

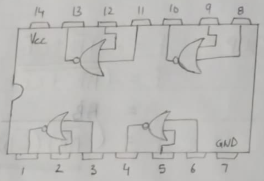
$F = \overline{(\overline{\bar{A}+\bar{B}})(\overline{\bar{B}+\bar{C}})}$

Teacher's Signature

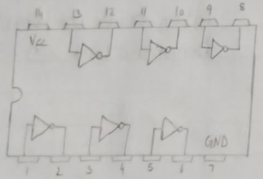
## IC Pinouts



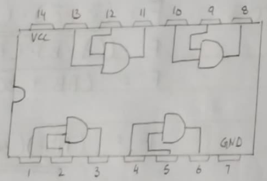
7400



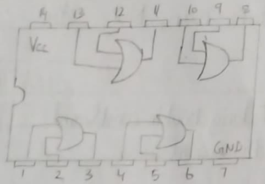
7402



7404



7408



7432

## RESULT

Realised :

1. SOP and POS functions using basic and Universal gates
2. Verified the truth table of basic logic gates

- Half Adder

Truth Table

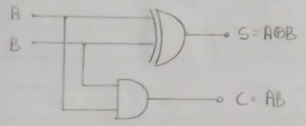
| Input | | output | |
|---|---|---|---|
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Boolean Function

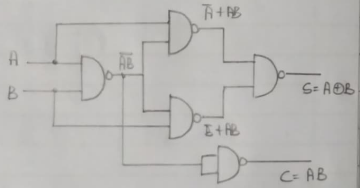
$S = A \oplus B$

$C = AB$

i) Using Basic Gates



ii) Using NAND Gates



- Full Adder

Truth Table

| Input | | | Output | |
|---|---|---|---|---|
| A | B | Cin | S | C |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

K Map

| A \ BCin | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | | 1 |
| 1 | 1 | | 1 | |

| A \ BCin | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | |
| 1 | | 1 | 1 | 1 |

$S = \bar{A}\bar{B}C_{in} + A\bar{B}\overline{C_{in}} + ABC_{in} + \bar{A}B\overline{C_{in}}$

$S = C_{in} \oplus (A \oplus B)$

$C = Ab + C_{in}A + C_{in}B$

# Adders and Subtractors

## AIM

i) To realize half adder, full adder, half subtractor and full subtractor using basic gates & universal gates

ii) To design and set up 4 bit binary parallel adder/subtractor circuit using IC 7483 chip

## COMPONENTS REQUIRED

IC trainer kit, Patch Chords, IC 7483, IC 7486, IC 7400, IC 7408 IC 7432,

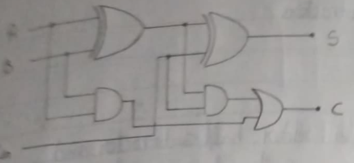
## THEORY

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

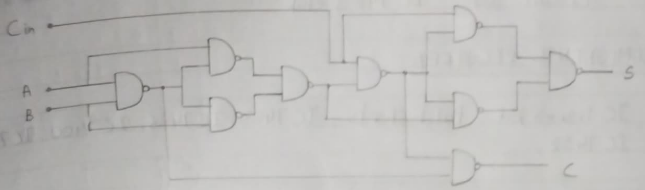
$S = A \oplus B$ $\qquad$ $C = AB$

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

i) Using Basic Gates



ii) Using NAND Gates



- <u>Half-Adder</u>

Truth Table

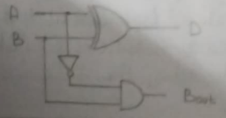
| Input | | Output | |
|---|---|---|---|
| A | B | D | Bout |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

Boolean Function

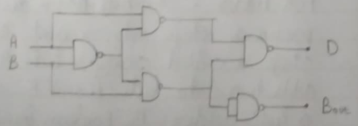
$D = A \oplus B$

$B_{out} = \bar{A}B$

i) Using Basic Gates



ii) Using NAND Gates



Full-adder are:

$S = (A \oplus B) \oplus C_{in}$ $\quad C = AB + C_{in}(A \oplus B)$

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

$D = A \oplus B$ $\quad B_{out} = \bar{A}B$

Full Subtractor: Subtracting two single-bit binary values, B, Cin from a single bit value A produce a difference D and a borrow out Bout

$D = (A \oplus B) \oplus B_{in}$ $\quad B_{out} = \bar{A}B + B_{in}(\overline{A \oplus B})$

4 Bit Binary Adders: It is a circuit that produces the sum of two binary numbers of 4 bit. It can be constructed with full-adder connected with output carry

For eg: 7 + 2 = 9

7 realizes as $A_3 \; A_2 \; A_1 \; A_0$
0 1 1 1

2 realizes as $B_3 \; B_2 \; B_1 \; B_0$
0 0 1 0

Sum = 1 0 0 1

- ## Full Subtractor

Truth Table

| Input | | | Output | |
|---|---|---|---|---|
| A | B | Bin | D | Bout |
| 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 |

K Map

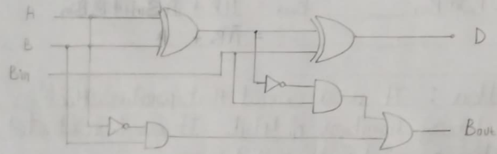
| A \ B Bin | 00 | 01 | 11 | 0 |
|---|---|---|---|---|
| 0 | | 1 | | 1 |
| 1 | 1 | | 1 | |

$D = A \oplus B + B_{in}$

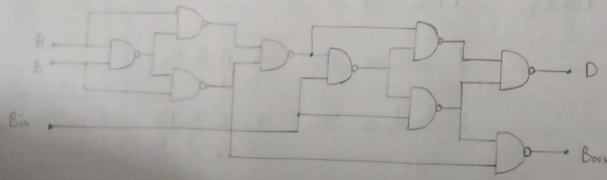
| A \ B Bin | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | 1 | 1 |
| 1 | | | 1 | |

$B_{out} = \bar{A}B + B_{in}(A \oplus B)$

i) Using Basic Gates



ii) Using NAND Gates



Date

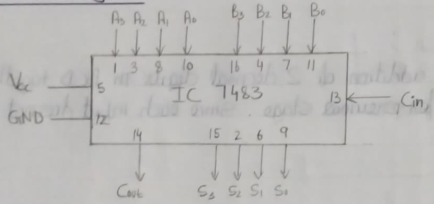
Expt. No. 2

Page No. 8

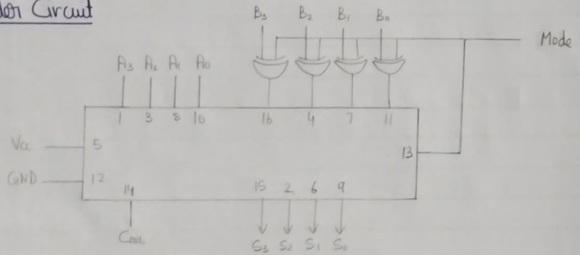
4 Bit BCD Adder :

Teacher's Signature

## IC 7483 Pin Diagram



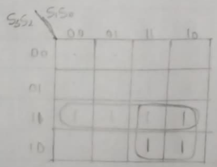
## Adder Circuit



## 4 Bit BCD Adder

| BCD | | | | carry |
|---|---|---|---|---|
| $S_3$ | $S_2$ | $S_1$ | $S_0$ | |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

### K Map for Correction Circuit



| $S_3S_2$ \ $S_1S_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | 1 | 1 | 1 | 1 |
| 10 | | | 1 | 1 |

$$Y = S_3S_2 + S_3S_1 + C_{out}$$

Date

Expt. No. 2

## PROCEDURE

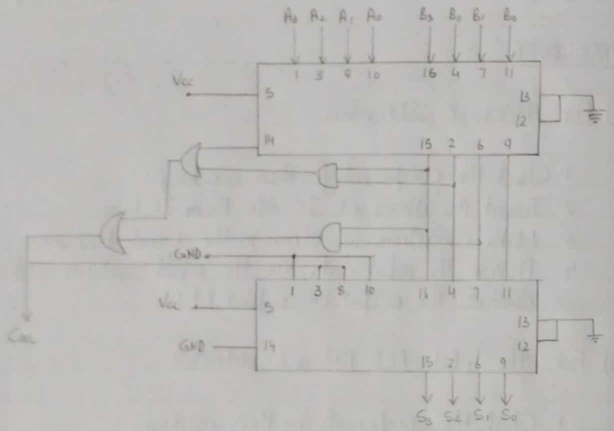
i) For Adders & Subtractors

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

ii) For the 4 Bit BCD Adders & Subtractors

1. Check the components for their working.
2. Insert appropriate IC into their IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

Teacher's Signature



Verification Table

i) 4 Bit Adder/Subtractor

| Mode | $A_3$ | $A_2$ | $A_1$ | $A_0$ | $B_3$ | $B_2$ | $B_1$ | $B_0$ | $C_{out}$ | $S_3$ | $S_2$ | $S_1$ | $S_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

ii) BCD Adder

| $A_3$ | $A_2$ | $A_1$ | $A_0$ | $B_3$ | $B_2$ | $B_1$ | $B_0$ | $C_{out}$ | $S_3$ | $S_2$ | $S_1$ | $S_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |

RESULT

Realised and verified the Truth Table for

i) half-adder, full-adder, half-subtractor and full-subtractor

ii) verified the working of IC 7483 as adder & Subtractor of 4 bits & as BCD adder with the help of verification table.