# LABORATORY MANUAL
## Computer Organization & Architecture


**Department of Computer Science & Engineering**
**Gaya College of Engineering, Gaya**

**Instructor : Dhruba Jyoti Kalita**

| | COA LAB (PART-I) | Semester IV | L-T-P<br>0-0-4 | |
|---|---|---|---|---|

| Experiment No. | Title of the Experiment | Objective of the Experiment |
|---|---|---|
| 1 | To study and verify the truth table of logic gates | Identify various ICs and their specification<br>a. OR gate<br>b. AND gate<br>c. NAND gate<br>d. NOR gate |
| 2 | Design and implementation using NAND gate | To realize why NAND gate is known as the universal gate by implementation of :<br>a. NOT using NAND<br>b. AND using NAND<br>c. OR using NAND<br>d. XOR using NAND |
| 3 | Adders and Subtractors | To realize<br>a. Half Adder and Full Adder<br>b. Half Subtractor and Full Subtractor by using Basic gates and NAND gates |
| 4 | Multiplexer and Demultiplexer | a. To design and set up a 4:1 Multiplexer (MUX) using only NAND gates.<br>b. To design and set up a 1:4 Demultiplexer(DE-MUX) using only NAND gates. |
| 5 | FlipFlop | a. Truth Table verification of<br>1) RS Flip Flop<br>2) T type Flip Flop.<br>3) D type Flip Flop.<br>4) JK Flip Flop.<br>b. Conversion of one type of Flip flop to another |

Text book:


- Modern Digital Electronics - R P Jain
- Digital Electronics: An Introduction To Theory And Practice by William Gothmann H
- Digital Electronics by John Morris
- Fundamentals of Digital Circuits by Anand Kumar

## EXPERIMENT: 1 LOGIC GATES

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

OBJECTIVE:

Identify various ICs and their specification

a. OR gate
b. AND gate
c. NAND gate
d. NOR gate

COMPONENTS REQUIRED:

- Breadboard.
- Connecting wires.
- IC 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486

THEORY:

The basic logic gates are the building blocks of more complex logic circuits. These logic gates perform the basic Boolean functions, such as AND, OR, NAND, NOR, Inversion, Exclusive-OR, Exclusive-NOR. Fig. below shows the circuit symbol, Boolean function, and truth. It is seen from the Fig that each gate has one or two binary inputs, A and B, and one binary output, C. The small circle on the output of the circuit symbols designates the logic complement. The AND, OR, NAND, and NOR gates can be extended to have more than two inputs. A gate can be extended to have multiple inputs if the binary operation it represents is commutative and associative.

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

TTL Transistor-transistor logic

ECL Emitter-coupled logic

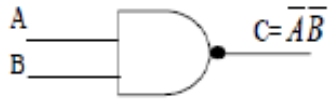
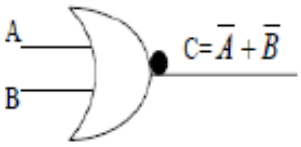
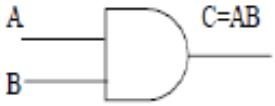
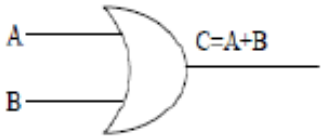
MOS Metal-oxide semiconductor

CMOS Complementary metal-oxide semiconductor

TTL and ECL are based upon bipolar transistors. TTL has a popularity among logic families. ECL is used only in systems requiring high-speed operation. MOS and CMOS, are based on field effect transistors. They are widely used in large scale integrated circuits because of their high component density and relatively low power consumption. CMOS logic consumes far less power than MOS logic. There are various commercial integrated circuit chips available. TTL ICs are usually distinguished by numerical designation as the 5400 and 7400 series.

PROCEDURE:

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

| S.NO | GATE | SYMBOL | INPUTS | | OUTPUT |
|---|---|---|---|---|---|
| | | | A | B | C |
| 1. | NAND IC 7400 |  | 0 | 0 | 1 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 0 |
| 2. | NOR IC 7402 |  | 0 | 0 | 1 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 0 |
| 3. | AND IC 7408 |  | 0 | 0 | 0 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 1 |
| 4. | OR IC 7432 |  | 0 | 0 | 0 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 1 |

## EXPERIMENT: 2 DESIGN AND IMPLEMENTATION USING NAND GATE

AIM: To design and implementation using NAND gate

OBJECTIVE:

- To realize why NAND gate is known as the universal gate by implementation of :

a. NOT using NAND
b. AND using NAND
c. OR using NAND
d. XOR using NAND

COMPONENTS REQUIRED:

- Breadboard.
- Connecting wires.
- IC 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486

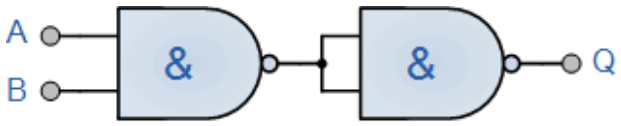
THEORY:

Boolean algebra is a branch of mathematical logic, where the variables are either true (1) or false (0).In order to construct NOT, AND, OR, XOR gates from NAND gates only, we need to be familiar with the following boolean algebra laws:

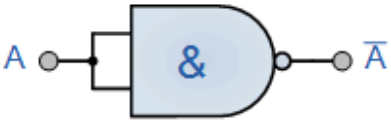
1. Involution Law

2. Idempotency (Idempotent) law

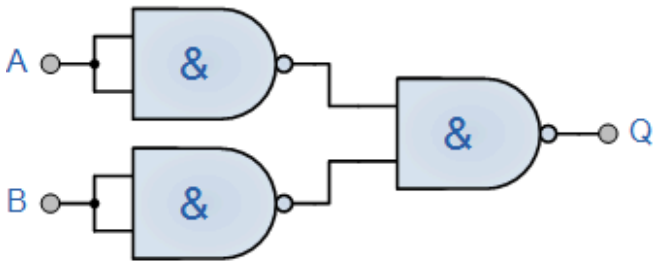
3. DeMorgan's Law



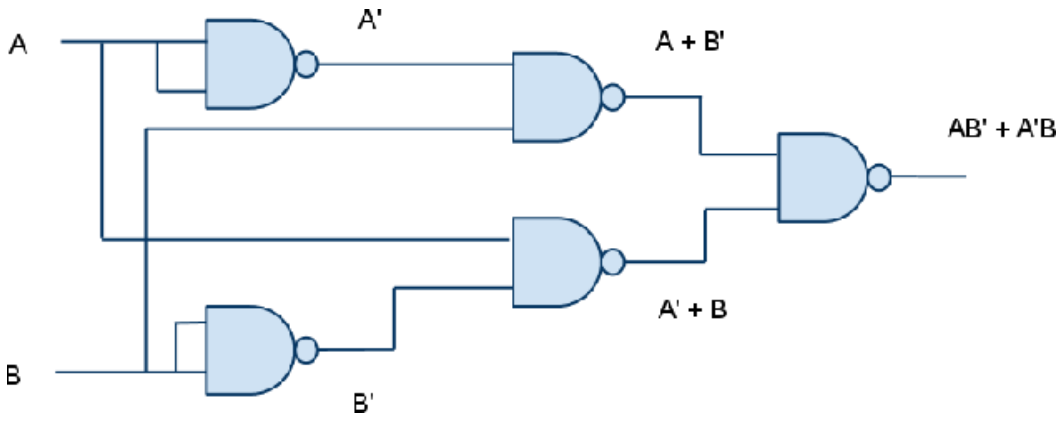
2-input AND Gate



NOT Gate



2-input OR Gate




2 input XOR gate

PROCEDURE:

Check the components for their working.

Insert the appropriate IC into the breadboard.

Make connections as shown in the circuit diagram.

Provide the input data via the input switches and observe the output on output LEDs

Verify the Truth Table

RESULT:

NOT, AND, OR, XOR gate is realized using NAND gate.

## EXPERIMENT:4 ADDERS AND SUBTRACTORS

AIM: To realize

i) Half Adder and Full Adder

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

LEARNING OBJECTIVE:

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

COMPONENTS REQUIRED:

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

THEORY:

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

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

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

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

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

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

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

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

I.

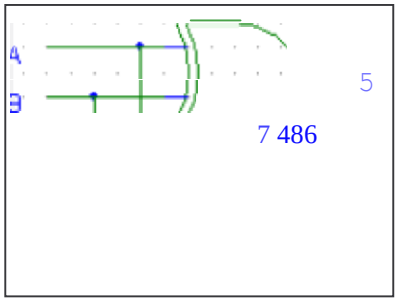
TRUTH TABLE

| INPUTS | | OUTPUTS | |
|---|---|---|---|
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| i | 0 | i | 0 |
| 1 | 1 | 0 | 1 |

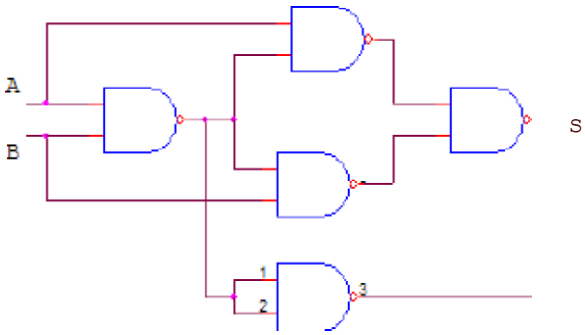
BOOLEAN EXPRESSIONS:

S=A & B
C=A B

i} Basic Gates




ii} NAND Gates




ii. FULL ADDER

TRUTH TABLE

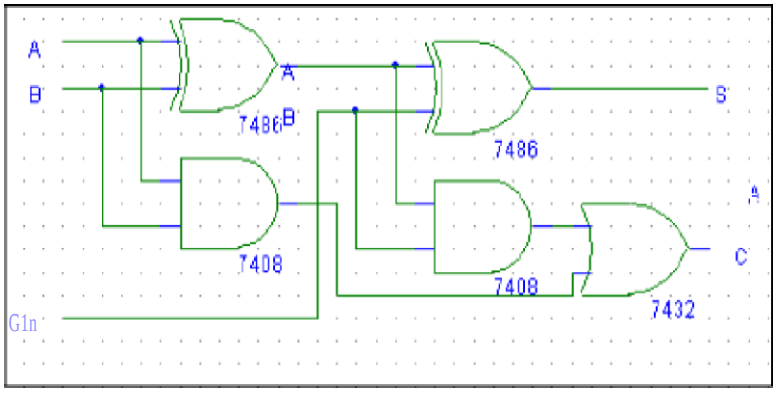
| INPUTS | | | OUTPUTS | |
|---|---|---|---|---|
| A | B | Cin | ' | C |
| 0 | 0 | 0 | 0 | 0 |
| | | | | |
| | | | | |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| | 0 | 1 | 0 | 1 |
| | 1 | 0 | 0 | 1 |
| | | | | |

BOOLEAN EXPRESSIONS:

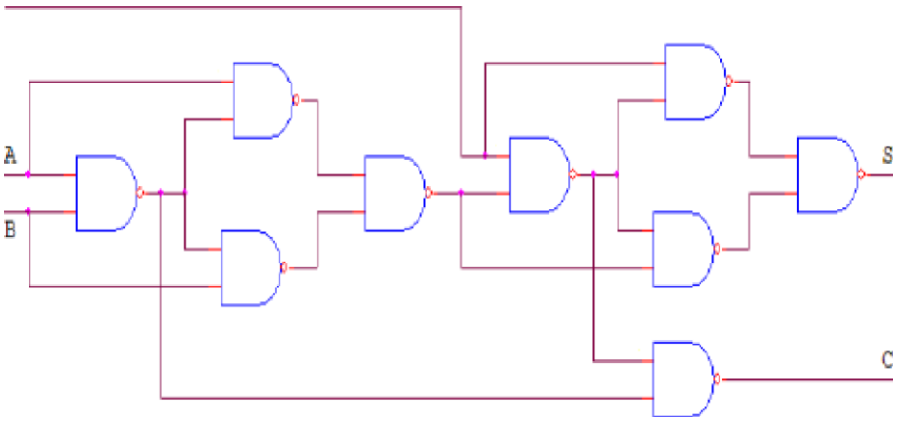
S= A & B *B* C

C=A B + B Cin + A Cin

ilBASIC GATES




**ii) NAND GATES**




III. HALF SUBTRACTOR

TRUTH TABLE

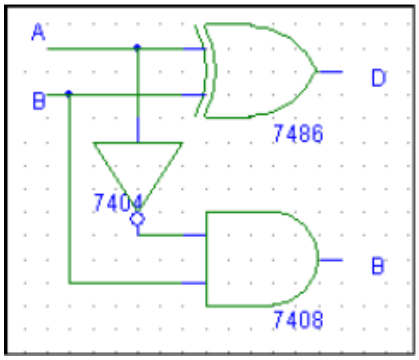
| INPLTS | | OUTPUTS | |
|---|---|---|---|
| A | B | D | Br |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| i | 0 | i | 0 |
| 1 | 1 | 0 | 0 |

BOOLEAN EXPRESSIONS:

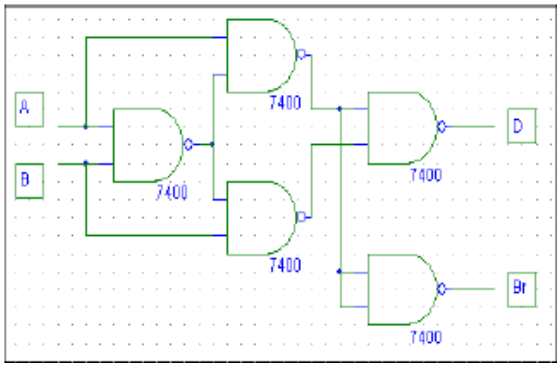
D = A & B

Br = $A\bar{B}$

**i)BASIC GATES**




**ii) NAND Gates**




## IV. FULL SUBTRACTOR

**TRUTH TABLE**

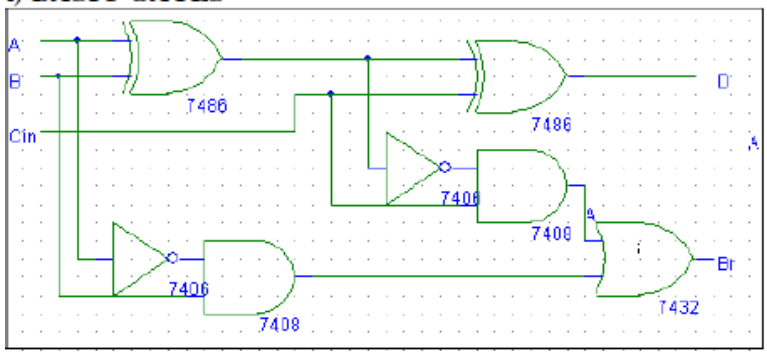
| INPUTS | | | OUTPUTS | |
|---|---|---|---|---|
| A | B | Cin | D | Br |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**BOOLEAN EXPRESSIONS:**

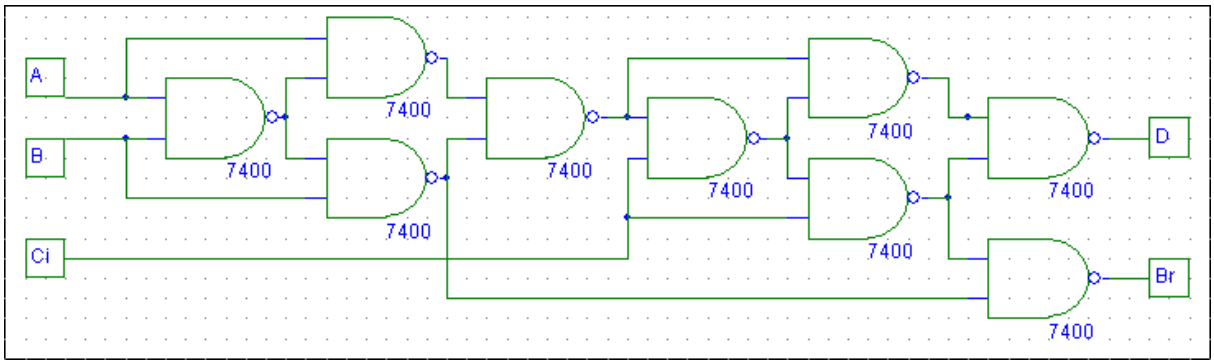
$$D = A \oplus B \oplus C$$

$$Br = \bar{A}\,B + B\,Cin + \bar{A}\,Cin$$

**i) BASIC GATES**




**ii) To Realize the Full subtractor using NAND Gates only**




**PROCEDURE:**

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

**RESULT:**

## EXPERIMENT: 4 MULTIPLEXER AND DEMULTIPLEXER

AIM: To design and set up the following circuit

1) To design and set up a 4:1 Multiplexer (MUX) using only NAND gates.
2) To design and set up a 1:4 Demultiplexer(DE-MUX) using only NAND gates.

LEARNING OBJECTIVE:

1. To learn about various applications of multiplexer and de-multiplexer

THEORY:

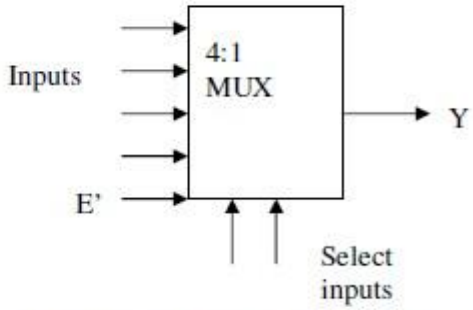
Multiplexers are very useful components in digital systems. They transfer a large number of information units over a smaller number of channels, (usually one channel) under the control of selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but only one output.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

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

COMPONENTS REQUIRED:

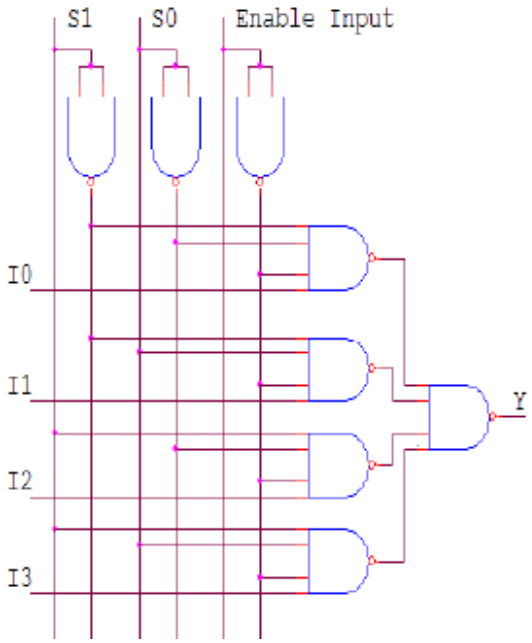
IC 7400, IC 7410, IC 7420, IC 7404, IC 74153, IC 74139, Patch Cords & IC Trainer Kit.

**i) 4:1 MULTIPLEXER**




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

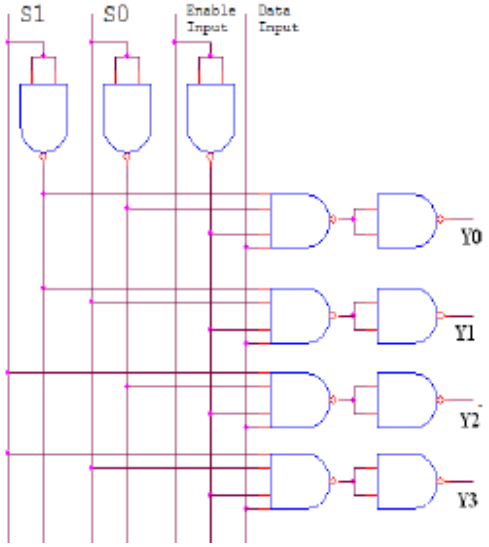
**REALIZATION USING NAND GATES**




**TRUTH TABLE**

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

**ii) DE-MUX USING NAND GATES**




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

**PROCEDURE:**

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

**EXPERIMENT: 5 FLIP FLOPS**

**AIM**: Truth Table verification of
1) RS Flip Flop
2) T type Flip Flop.
3) D type Flip Flop.
4) JK Flip Flop.

**OBJECTIVE*:***

- To learn about various Flip-Flops
- To learn about applications of FFs
- Conversion of one type of Flip flop to another