LAB MANUAL

DIGITAL ELECTRONICS

SUBJECT CODE – 2421303



Department of Electronics Engineering

GOVERNMENT POLTECHNIC DARBHANGA

LIST OF EXPERIMENTS

1.Functionality of OR, AND & NOT logic gates using ICs

2.Logic gates using universal NAND gate IC

3.Logic gates using universal NOR gate IC

4.Half adder using basic gates

5.Half subtractor using basic gates

6.Full Adder using NAND gates

7.Full Subtractor using NOR gates

8.Functionality of multiplexer

9.Multiplexer design using ICs (74151/74150)

10.Functionality of de-multiplexer

List of Major Laboratory Equipment, Tools and Software:

S. NO.	Name of Equipment, Tools, and Software	Broad Specifications	Relevant Experiment/Practical Number
1.	Oscilloscope	Dual Channel 20MHz	All
2.	Function generator	100MHz Function & Arbitrary Generator, 500MSa/s-DG4102	All
3.	Digital IC Trainer Kits	Power Supply: +5V, +/- 12V Display Type: 2 Digit BCD to Decimal Display	All
4.	Logic Gates ICs	Two input and 3-Input	All
5.	Bread Board	MB 102 Breadboard with Power Supply Module, Jumper Wires, Battery Clip,830 & 400 tie-Points	All
6.	Digital Multimeter	DM-86 Digital Multimeter AC Frequency Response: 40-400Hz Low Battery Display: Approx. < 7.5V	All
7.	IC Tester	1.Package: Digital ICs of 14, 16, 18,20,24,28 & 40 pins dual in line. 2. Range: Tristate, Open Collector & Bidirectional TTL/CMOS ICs. 3. Method: Truth table comparison. 4. Sockets: 20 and 40 pin ZIF. 5.Keyboard: 24 feather touch keys. 6. Display: 16 digit 0.5" Seven segment LED display. 7. Voltage: 230 volts + 10% 50Hz, AC.	All

EXPERIMENT-01

Functionality of OR, AND & NOT logic gates using ICs

Aim

To study and verify the truth tables of **AND, OR, and NOT logic gates** using ICs.

Apparatus Required

- IC 7408 (AND gate)
- IC 7432 (OR gate)
- IC 7404 (NOT gate)
- Breadboard
- Connecting wires
- DC Power Supply (+5V)
- LED indicators
- Resistors (220Ω or 330Ω)
- Switches / input keys

Theory

Logic gates are digital circuits that perform logical operations on binary inputs (0 and 1).

- **AND Gate**: Output is HIGH only when all inputs are HIGH
- **OR Gate**: Output is HIGH when any input is HIGH
- **NOT Gate**: Output is the complement of input

Each gate is implemented using specific ICs from the **7400 series**.

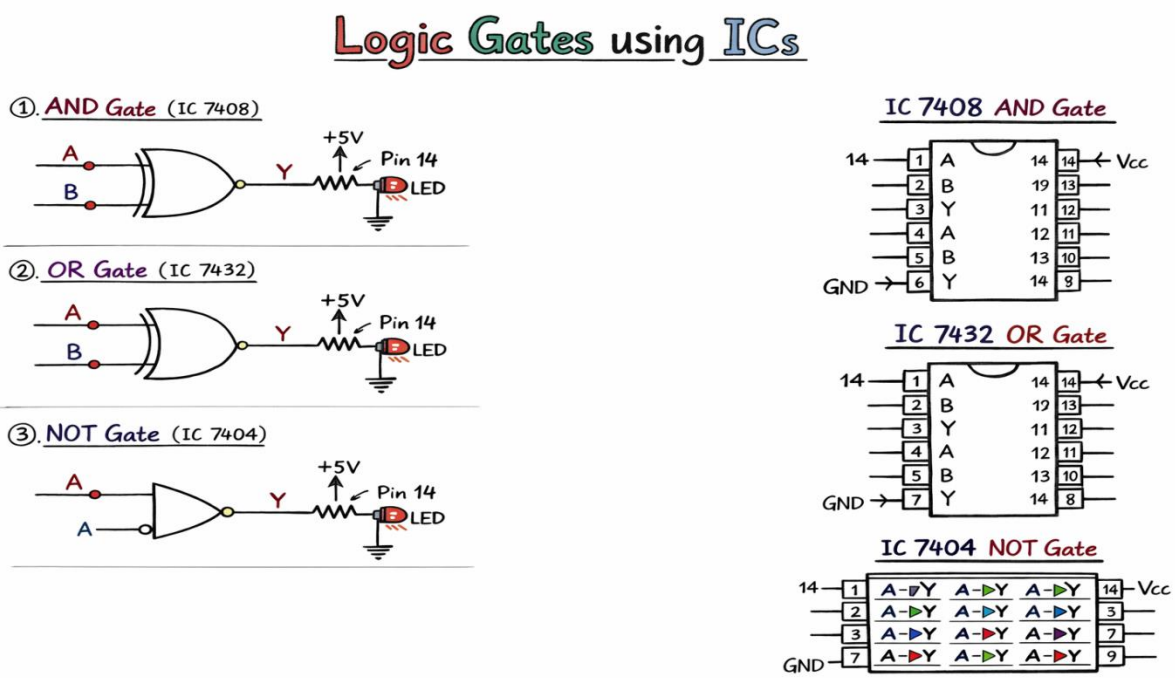
Circuit Connections

Common Connections:

- Pin 14 → Vcc (+5V)
- Pin 7 → Ground (GND)

Gate Connections:

- Inputs are given through switches
- Output is connected to LED through resistor



Procedure

1. Place the IC (7408 / 7432 / 7404) on the breadboard.
2. Connect power supply:
   - Pin 14 to +5V
   - Pin 7 to GND
3. Identify input and output pins from IC datasheet.
4. Connect inputs using switches.
5. Connect output to LED through a resistor.
6. Apply different combinations of inputs (0 and 1).
7. Observe and record the output (LED ON/OFF).
8. Repeat for all three gates.

Observation Table

AND Gate (IC 7408)

A	B	Output (Y)
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate (IC 7432)

A	B	Output (Y)
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate (IC 7404)

A	Output (Y)
0	1
1	0

Result

The truth tables of **AND, OR, and NOT gates** were successfully verified using ICs:

- AND gate produced output HIGH only when both inputs were HIGH.
- OR gate produced output HIGH when at least one input was HIGH.
- NOT gate produced output as the inverse of input.

Precautions

- Ensure correct IC orientation (check notch).
- Do not exceed +5V supply.
- Use current-limiting resistor with LED.
- Make tight and proper connections.

EXPERIMENT-02

Logic gates using universal NAND gate IC

Aim

To implement basic logic gates (**AND, OR, NOT**) using only NAND gates (IC 7400) and verify their truth tables.

Apparatus Required

- **IC 7400 Quad NAND Gate**
- Breadboard
- Connecting wires
- +5V DC supply
- LED
- Resistors (220Ω / 330Ω)
- Switches

Theory

A **NAND gate** is called a **universal gate** because all basic logic gates can be implemented using only NAND gates.

NAND Gate Expression:

$Y = \overline{A \cdot B}$

1. NOT Gate using NAND

Connection:

- Connect both inputs of NAND together (A = B)

Output:

$Y = \overline{A}$

Truth Table:

A	Y
0	1
1	0

2. AND Gate using NAND

Connection:

- First NAND → gives $\overline{A \cdot B}$
- Second NAND (as NOT) → invert output

Expression:

$Y = \overline{\overline{A \cdot B}} = A \cdot B$

Truth Table:

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

3. OR Gate using NAND

Connection:

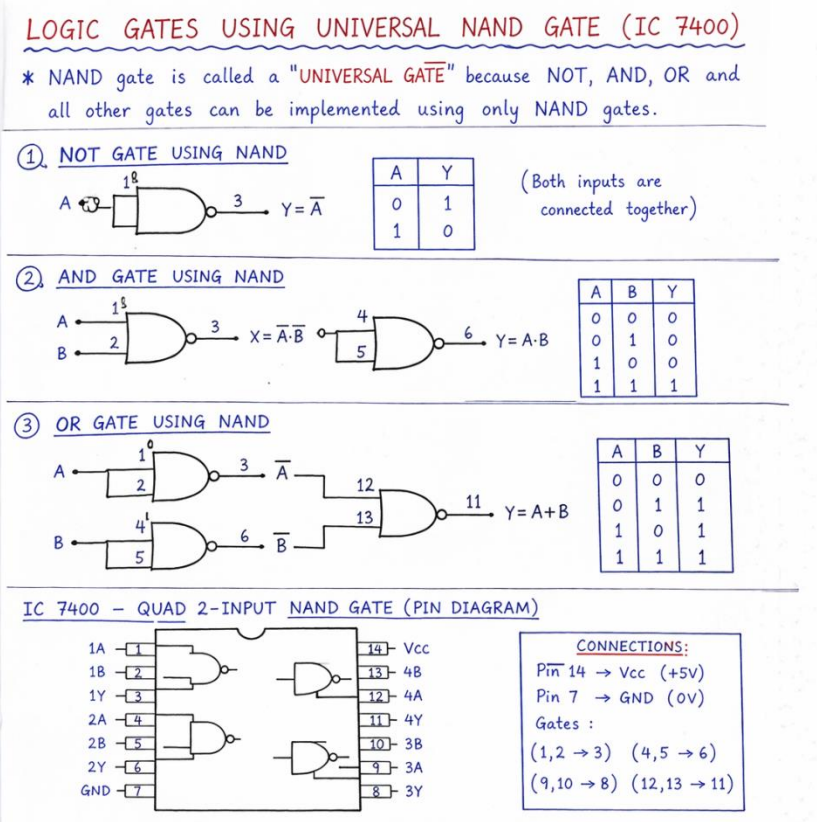
- Use NAND as NOT for inputs
- Then NAND the outputs

Expression:

$Y = \overline{\overline{A} \cdot \overline{B}} = A + B$

Truth Table:

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



IC 7400 Pin Configuration

- Pin 14 → Vcc (+5V)
- Pin 7 → GND

Gates inside IC:

- (1,2 → 3)
- (4,5 → 6)
- (9,10 → 8)
- (12,13 → 11)

Procedure

1. Place **IC 7400** on breadboard
2. Connect:
    - Pin 14 → +5V
    - Pin 7 → GND
3. Build circuits:
    - NOT using single NAND
    - AND using two NANDs
    - OR using three NANDs
4. Apply inputs using switches
5. Observe output using LED
6. Record truth tables

Result

- NOT, AND, and OR gates were successfully implemented using only NAND gates.
- Verified that NAND gate is a **universal gate**.

Precautions

- Check IC orientation (notch position)
- Use proper resistor with LED
- Ensure correct wiring connections
- Avoid short circuits

EXPERIMENT-03

Logic gates using universal NOR gate IC

Aim

To implement **NOT, AND, and OR gates** using only NOR gates and verify their truth tables.

Apparatus Required

- **IC 7402 Quad NOR Gate**
- Breadboard
- Connecting wires
- +5V DC supply
- LED
- Resistor (220Ω / 330Ω)
- Switches

Theory

A **NOR gate** is also a **universal gate** because all basic logic gates can be implemented using only NOR gates.

NOR Gate Expression:

$Y = \overline{A + B}$

1. NOT Gate using NOR

Connection:

- Connect both inputs together (A = B)

Expression:

$Y = \overline{A}$

Truth Table:

A	Y
0	1
1	0

2. OR Gate using NOR

Connection:

- First NOR → gives $\overline{A + B}$
- Second NOR (as NOT) → invert output

Expression:

Y= $\overline{\overline{A + B}}$ = A+B

Truth Table:

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

3. AND Gate using NOR

Connection:

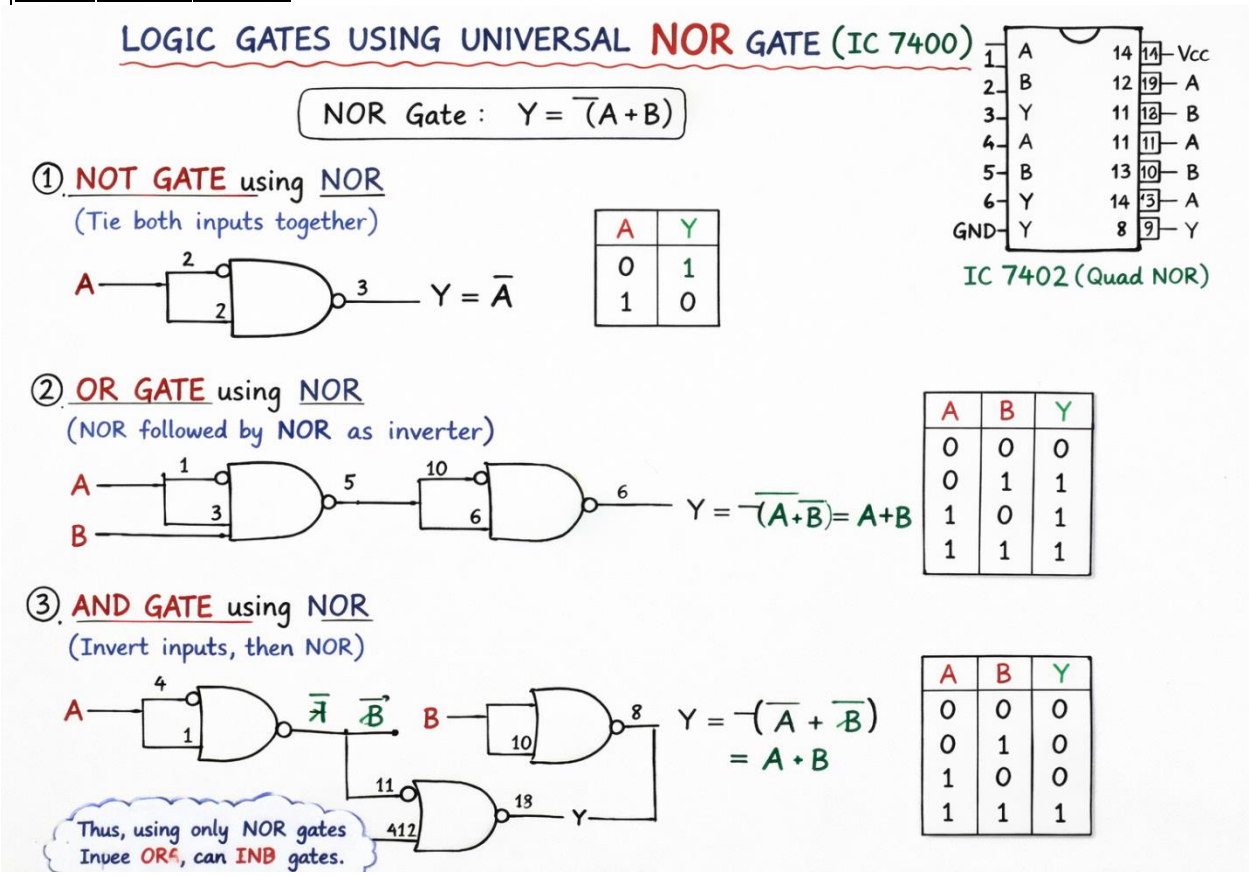
- Use NOR as NOT for inputs
- Then NOR the outputs

Expression:

Y= $\overline{\overline{A} + \overline{B}}$ =A·B

Truth Table:

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



IC 7402 Pin Configuration

- Pin 14 → Vcc (+5V)
- Pin 7 → GND

Gates inside IC:

- (2,3 → 1)
- (5,6 → 4)
- (8,9 → 10)
- (11,12 → 13)

Procedure

1. Place **IC 7402** on breadboard
2. Connect:
    - Pin 14 → +5V
    - Pin 7 → GND
3. Construct circuits:
    - NOT using 1 NOR
    - OR using 2 NORs
    - AND using 3 NORs
4. Apply inputs via switches
5. Observe output using LED
6. Record truth tables

Result

- Successfully implemented NOT, OR, and AND gates using only NOR gates.
- Verified that NOR gate is a **universal gate**.

Precautions

- Check IC orientation (notch position)
- Ensure proper wiring
- Use resistor with LED
- Avoid loose connections

EXPERIMENT-04

Half adder using basic gates

Aim

To design and implement a **Half Adder** using basic logic gates and verify its truth table.

Theory

A **Half Adder** is a combinational circuit used to add **two single-bit binary numbers**.

Inputs:

- A
- B

Outputs:

- **Sum (S)**
- **Carry (C)**

Logic Used

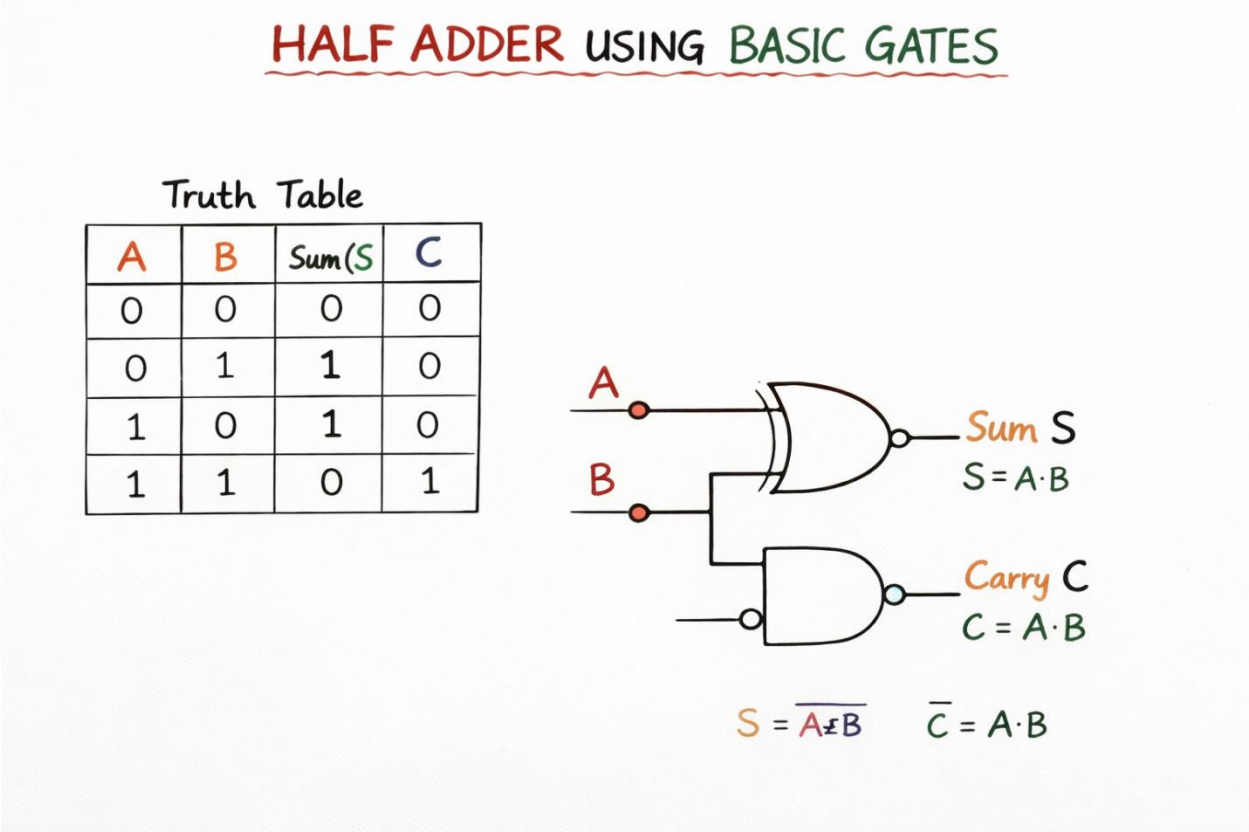
- **Sum (S)** → XOR Gate
- **Carry (C)** → AND Gate

Boolean Expressions:

$S = A \oplus B$
$C = A \cdot B$

Truth Table

A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Logic Circuit (Explanation)

- Inputs **A and B** go to:
  - XOR gate → gives **Sum (S)**
  - AND gate → gives **Carry (C)**

Handwritten-style Diagram (How to Draw in Exam)

Follow these steps:

1. Draw two inputs A and B
2. Connect them to:
   - XOR gate → label output as **S (Sum)**
   - AND gate → label output as **C (Carry)**
3. Clearly label gates and outputs

IC Implementation

- XOR Gate → **IC 7486 XOR Gate**
- AND Gate → **IC 7408 AND Gate**

Power Supply:

- Pin 14 → +5V
- Pin 7 → GND

Procedure

1. Place ICs on breadboard
2. Connect power supply
3. Give inputs using switches
4. Observe outputs using LEDs
5. Verify truth table

Result

- Half Adder circuit was successfully implemented
- Verified:
    - Sum = XOR operation
    - Carry = AND operation

Limitations

- Cannot handle carry from previous stage
- Used only for **single-bit addition**

EXPERIMENT-05

Half subtractor using basic gates

Aim

To design and implement a **Half Subtractor** using basic logic gates and verify its truth table.

Theory

A **Half Subtractor** is a combinational circuit used to subtract **two single-bit binary numbers**.

Inputs:

- A (Minuend)
- B (Subtrahend)

Outputs:

- **Difference (D)**
- **Borrow (B_o)**

Logic Used

- **Difference (D)** → XOR Gate
- **Borrow (B_o)** → NOT + AND Gate

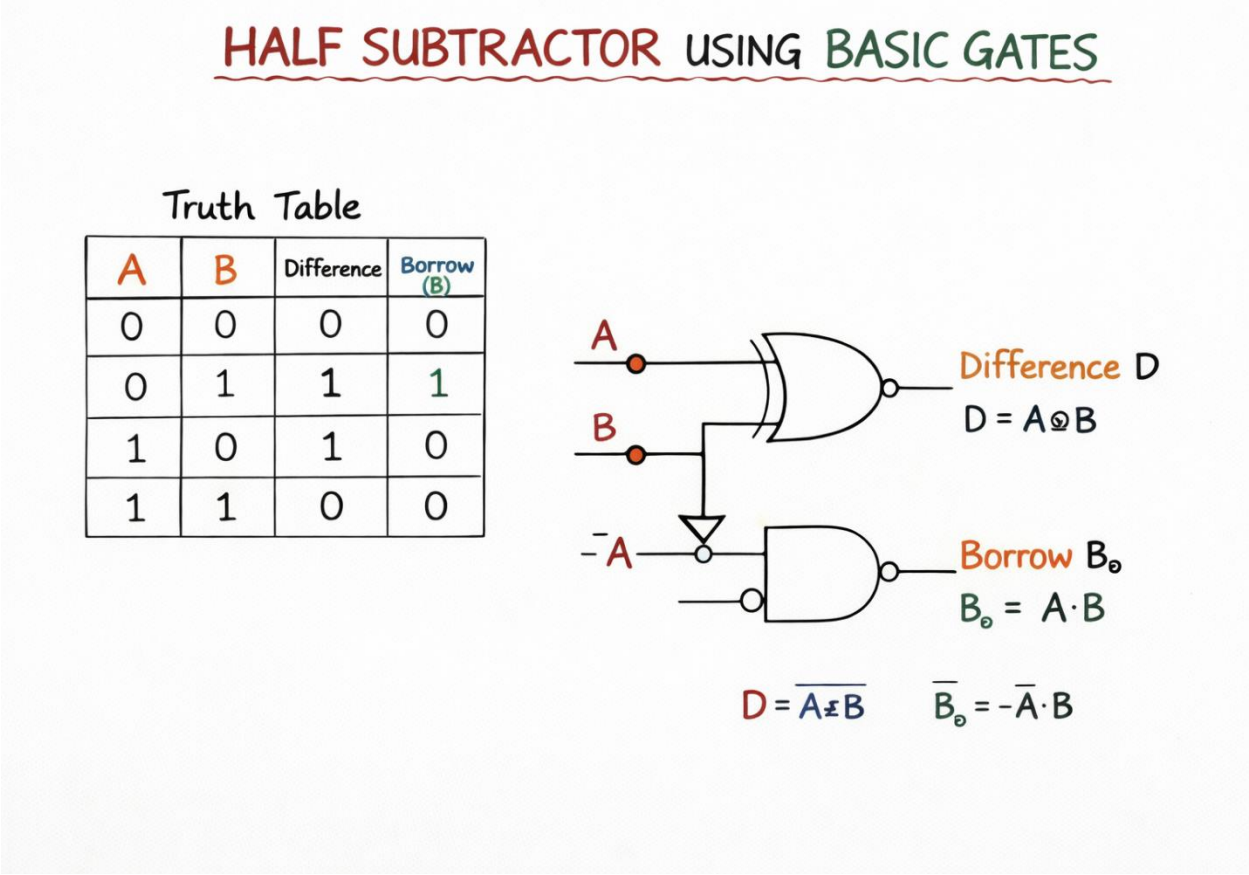
Boolean Expressions:

$D=A \oplus B$

$Bo=\overline{A} \cdot B$

Truth Table

A	B	Difference (D)	Borrow (B_o)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Logic Circuit (Explanation)

- Inputs **A and B** go to:
  - XOR gate → gives **Difference (D)**
- Input **A** is inverted using NOT gate
- Output of NOT(A) and B goes to AND gate → gives **Borrow (B_0)**

Handwritten-style Diagram (How to Draw in Exam)

Follow these steps:

1. Draw two inputs **A and B**
2. Connect them to **XOR gate → output D (Difference)**
3. Pass input **A through NOT gate**
4. Connect NOT(A) and B to **AND gate → output Borrow (B_0)**
5. Label all gates and outputs clearly

IC Implementation

- XOR Gate → **IC 7486 XOR Gate**
- AND Gate → **IC 7408 AND Gate**
- NOT Gate → **IC 7404 NOT Gate**

Power Supply:

- Pin 14 → +5V
- Pin 7 → GND

Procedure

1. Place ICs on breadboard
2. Connect power supply
3. Apply inputs using switches
4. Observe outputs using LEDs
5. Verify truth table

Result

- Half Subtractor circuit was successfully implemented
- Verified:
    - Difference = XOR operation
    - Borrow = NOT(A) AND B

Limitations

- Cannot handle borrow from previous stage
- Suitable only for **single-bit subtraction**

EXPERIMENT-06

Full Adder using NAND gates

Aim

To design and implement a **Full Adder** using only NAND gates and verify its truth table.

Theory

A **Full Adder** is a combinational circuit that adds **three binary inputs**:

Inputs:

- A
- B
- Cin (Carry input)

Outputs:

- **Sum (S)**
- **Carry (Cout)**

Logic Expressions

Sum:

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

Carry:

$C_{out} = AB + BC_{in} + AC_{in}$

NAND Implementation Concept

Since NAND is a **universal gate**, we implement:

- XOR using NAND gates
- AND using NAND
- OR using NAND

Step-by-Step NAND Construction

1. XOR using NAND (for A ⊕ B)

Use **4 NAND gates**:

- X1 = A NAND B
- X2 = A NAND X1
- X3 = B NAND X1
- X4 = X2 NAND X3 → gives A ⊕ B

2. Second XOR with Cin

- S = (A ⊕ B) ⊕ Cin
  → again use 4 NAND gates

3. Carry using NAND

Expression:

$C_{out} = AB + BC_{in} + AC_{in}$

Implement:

- Generate AB, BCin, ACin using NAND combinations
- Combine using NAND (OR using NAND logic)

IC Used

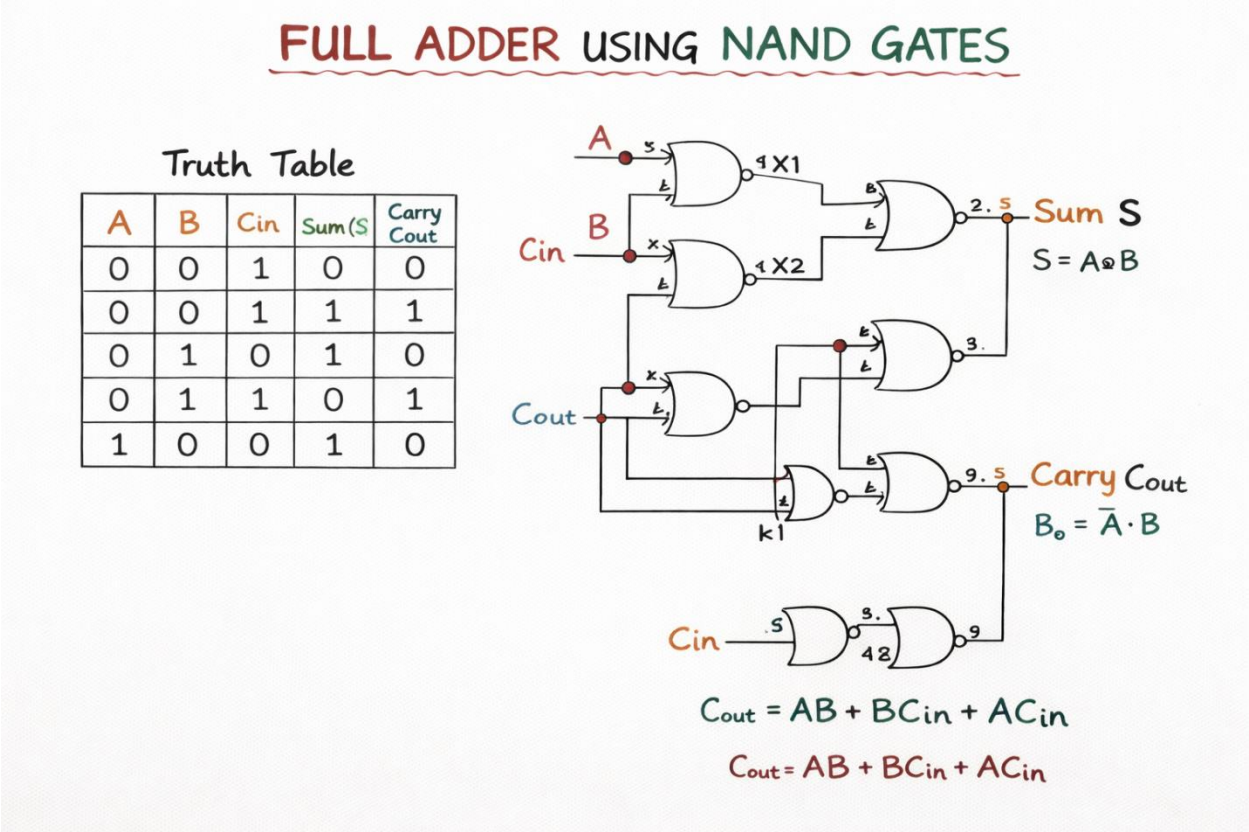
- **IC 7400 Quad NAND Gate**

Multiple ICs required (since many NAND gates are needed)

Truth Table

A	B	Cin	Sum (S)	Carry (Cout)
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

A	B	Cin	Sum (S)	Carry (Cout)
1	1	0	0	1
1	1	1	1	1



How to Draw in Exam (Important Tips)

1. First draw **XOR using NAND (4 gates)**
2. Connect output with Cin → second XOR (4 gates)
3. Draw carry part using NAND combinations
4. Label:
   - Inputs: A, B, Cin
   - Outputs: S, Cout
5. Keep diagram neat and symmetric

Procedure (Short)

1. Place IC 7400 on breadboard
2. Connect:
   - Pin 14 → +5V
   - Pin 7 → GND
3. Build XOR and Carry circuits using NAND

4. Apply inputs
5. Observe outputs using LEDs

Result

- Full Adder successfully implemented using only NAND gates
- Verified:
    - Sum = XOR logic
    - Carry = combination logic

Precautions

- Use correct pin connections
- Avoid loose wiring
- Use resistor with LED
- Check IC orientation

EXPERIMENT-07

Full Subtractor using NOR gates

Aim

To design and implement a **Full Subtractor** using only NOR gates and verify its truth table.

Theory

A **Full Subtractor** is a combinational circuit used to subtract **three binary inputs**:

Inputs:

- A (Minuend)
- B (Subtrahend)
- Bin (Borrow input)

Outputs:

- **Difference (D)**
- **Borrow (Bout)**

Logic Expressions

Difference:

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

Borrow:

$B_{out} = \overline{A}B + \overline{A}B_{in} + BB_{in}$

NOR Implementation Concept

Since NOR is a **universal gate**, we implement:

- NOT using NOR
- AND using NOR
- OR using NOR
- XOR using NOR combinations

Step-by-Step NOR Construction

1. NOT using NOR

- Tie both inputs together

$\overline{A}$ = A↓A

2. OR using NOR

- NOR followed by NOR (inverter)

3. AND using NOR

- Invert inputs, then NOR

4. XOR using NOR

- Requires multiple NOR gates (similar to NAND XOR structure)

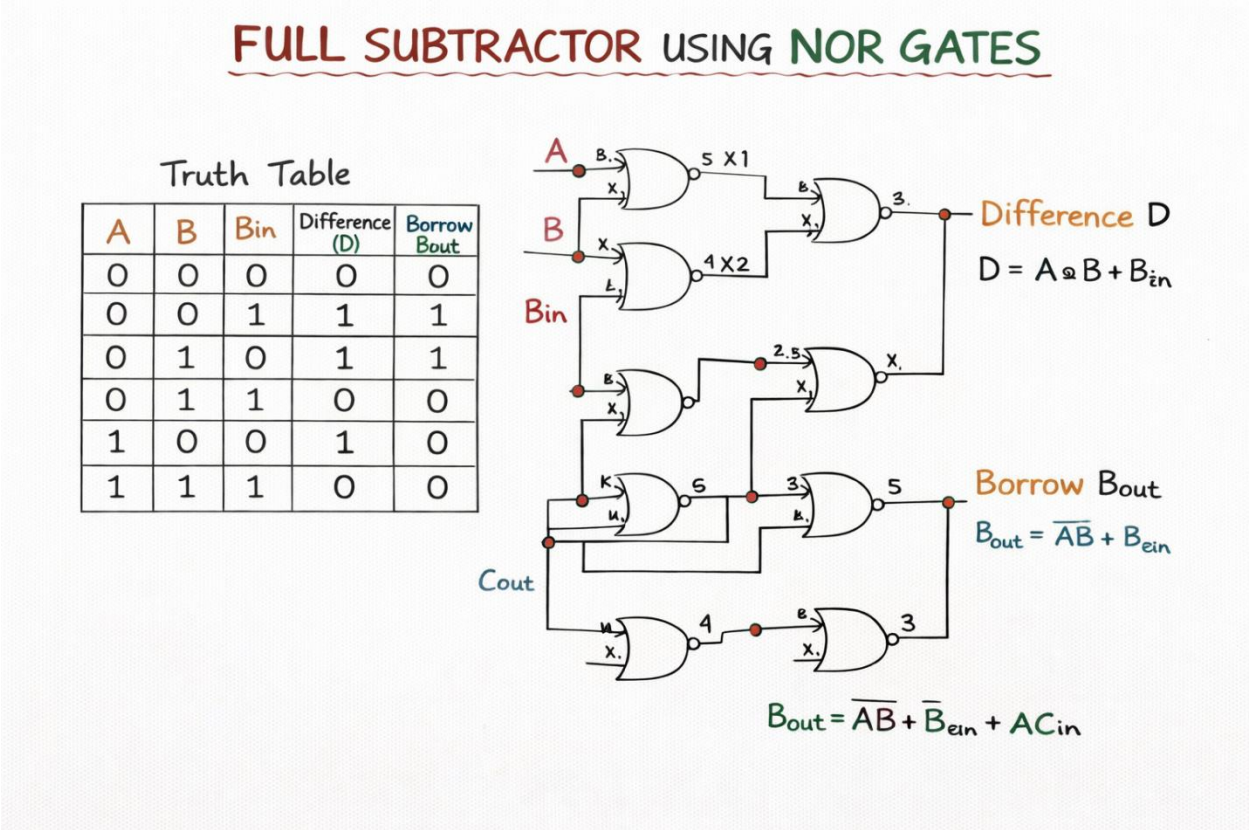
IC Used

- **IC 7402 Quad NOR Gate**

Multiple ICs required

Truth Table

A	B	Bin	Difference (D)	Borrow (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



How to Draw in Exam (Important Tips)

1. First draw **XOR using NOR gates**
2. Use second XOR with Bin → gives Difference
3. For Borrow:
   - Generate $\overline{A}$B, $\overline{A}$ B_{in},BB_{in}
   - Combine using NOR (OR logic)
4. Label:
   - Inputs: A, B, Bin
   - Outputs: D, Bout
5. Keep diagram neat and properly spaced

Procedure (Short)

1. Place IC 7402 on breadboard
2. Connect:
   - Pin 14 → +5V
   - Pin 7 → GND
3. Construct circuit using NOR gates
4. Apply inputs
5. Observe outputs using LEDs

Result

- Full Subtractor successfully implemented using NOR gates
- Verified:
    - Difference = XOR logic
    - Borrow = combination logic

Precautions

- Check pin connections carefully
- Avoid loose wiring
- Use proper resistor with LED
- Ensure correct IC orientation

EXPERIMENT-08

Functionality of multiplexer

Aim

To study and verify the **functionality of a Multiplexer (MUX)**.

Apparatus Required

- **IC 74153 Dual 4:1 Multiplexer**
- Breadboard
- Connecting wires
- +5V DC power supply
- LED indicators
- Resistors (220Ω / 330Ω)
- Switches (for inputs and select lines)

Theory

A **Multiplexer (MUX)** is a combinational circuit that selects **one input from multiple inputs** and connects it to a single output.

It works as a **data selector**.

- Number of select lines = log2(n)
- For 4 inputs → 2 select lines required

4:1 Multiplexer

Inputs:

- Data inputs → I_0, I_1, I_2, I_3
- Select lines → S_1, S_0

Output:

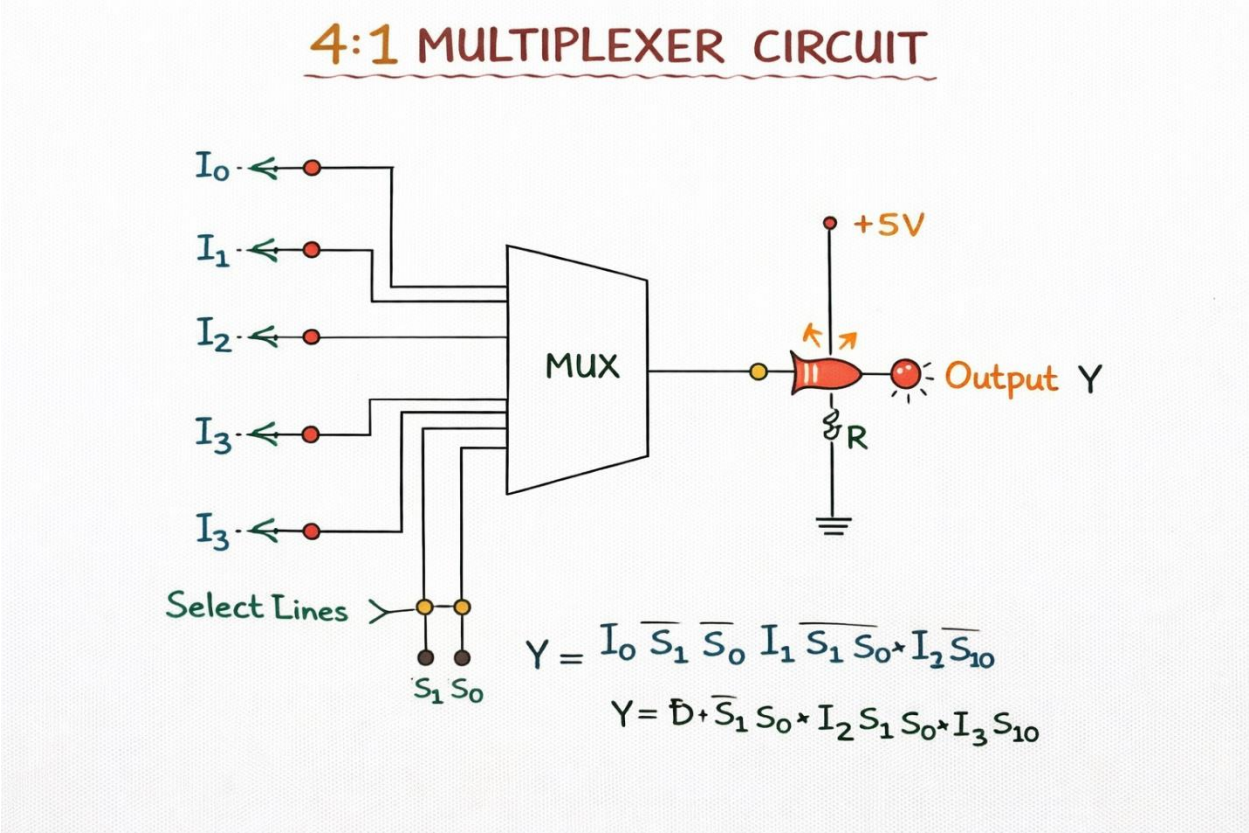
- Y

Boolean Expression

$Y = I_0 \overline{S_1 S_0} + I_1 \overline{S_1} S_0 + I_2 S1 \overline{S_0}$

Circuit Connections

- Pin 16 → Vcc (+5V)
- Pin 8 → GND
- Connect inputs (I_0–I_3) using switches
- Connect select lines (S_1, S_0) using switches
- Connect output Y to LED through resistor



Procedure

1. Place the IC on the breadboard.
2. Connect power supply:
    - Vcc → +5V
    - GND → 0V
3. Connect input lines (I_0–I_3) to switches.
4. Connect select lines (S_1, S_0) to switches.

5. Connect output to LED.
6. Apply different combinations of select inputs.
7. Observe which input is reflected at output.
8. Record the observations.

Observation Table

S_1	S_0	Output (Y)
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Result

- Multiplexer successfully selected one input at a time based on select lines.
- Verified that MUX works as a **data selector**.

Precautions

- Check IC orientation carefully
- Ensure proper connections
- Use resistor with LED
- Avoid loose wiring

EXPERIMENT-09

Multiplexer design using ICs (74151/74150)

Aim

To study and verify the working of a multiplexer using
IC 74151 and
IC 74150.

Apparatus Required

- IC 74151 / IC 74150
- Breadboard
- Connecting wires
- DC Power Supply (+5V)
- Logic switches (for inputs)
- LEDs (for output indication)
- Resistors (220Ω)

Theory

A **Multiplexer (MUX)** is a combinational circuit that selects one input from many inputs and forwards it to a single output line.

For IC 74151

- It is an **8:1 multiplexer**
- Has **3 select lines (S2, S1, S0)**
- **Enable ($\bar{G}$)** is active LOW
- Output is available in both **true (Y)** and **complement ($\bar{Y}$)** forms

Output equation:

Y = Dn (depending on select inputs)

- It is a **16:1 multiplexer**
- Has **4 select lines (S3–S0)**
- Output is **inverted**

Output:

Y = $\overline{D_n}$

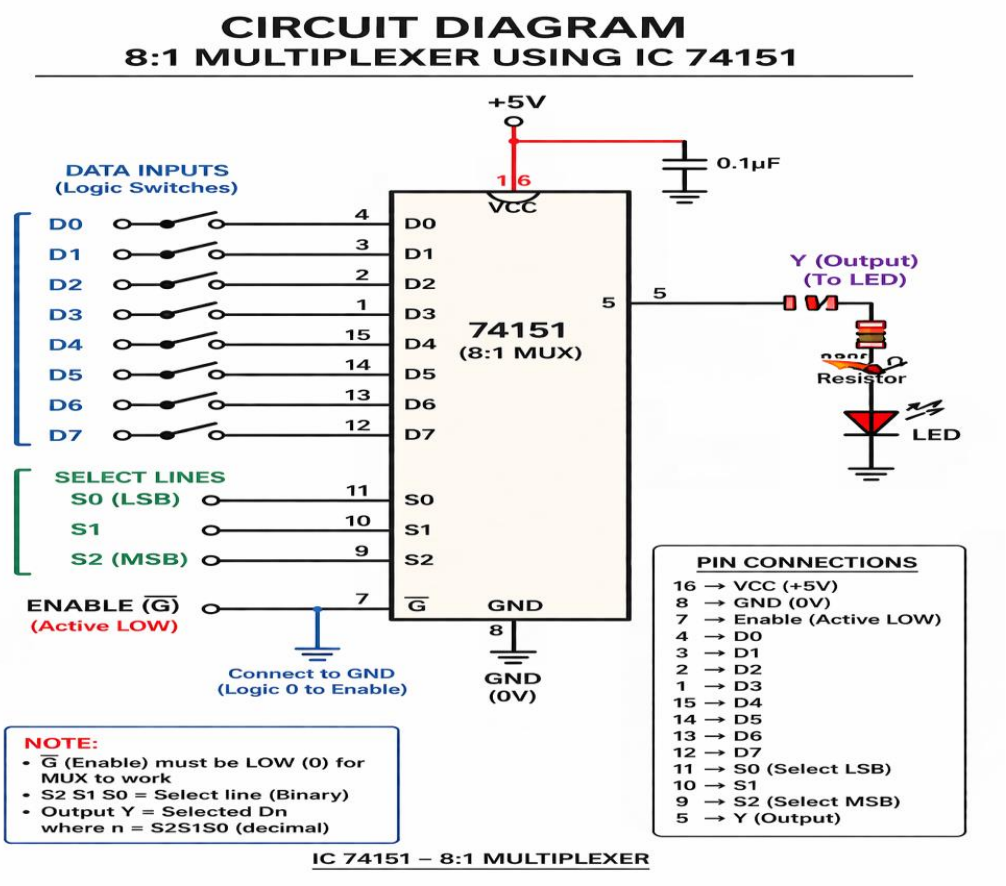
Circuit Diagram (What to Draw in Exam)

☒ For 74151:

- 8 inputs (D0–D7) on left
- Select lines (S2, S1, S0) below
- Enable grounded
- Output Y → LED

✔For 74150:

- 16 inputs (D0–D15)
- Select lines (S3–S0)
- Output (inverted) to LED



Procedure

Part A: Using IC 74151

1. Place IC on breadboard
2. Connect **Vcc (pin 16) = +5V** and **GND (pin 8)**
3. Connect **Enable ($\bar{G}$) to LOW**
4. Give inputs to D0–D7 using switches
5. Apply select inputs S2, S1, S0
6. Connect output Y to LED
7. Change select lines and observe output

Part B: Using IC 74150

1. Connect power supply properly
2. Apply inputs to D0–D15
3. Apply select lines S3–S0
4. Keep enable LOW
5. Observe output on LED

Observation Table

For 74151

S2	S1	S0	Selected Input	Output
0	0	0	D0	D0
0	0	1	D1	D1
…	…	…	…	…
1	1	1	D7	D7

For 74150

S3	S2	S1	S0	Output
0	0	0	0	$\bar{D}$0
…	…	…	…	…
1	1	1	1	$\bar{D}$15

Result

- The multiplexer operation using **74151 and 74150 ICs** was successfully verified.
- Output changes according to select lines.
- 74150 gives **inverted output**, while 74151 provides both forms.

Precautions

- Ensure correct IC orientation
- Do not exceed +5V supply
- Make proper ground connections
- Avoid loose wiring

EXPERIMENT-10

Functionality of de-multiplexer

Aim

To study and verify the functionality of a De-Multiplexer using a standard IC such as IC 74138 *(commonly used as DEMUX)*.

Apparatus Required

- IC 74138
- Breadboard
- +5V DC power supply
- Connecting wires
- Toggle switches (for inputs/select lines)
- LEDs
- 220Ω resistors

Theory

A **De-Multiplexer (DEMUX)** is a combinational circuit that takes **one input** and distributes it to one of several outputs depending on select lines.

The IC IC 74138 works as a **1-to-8 DEMUX** when:

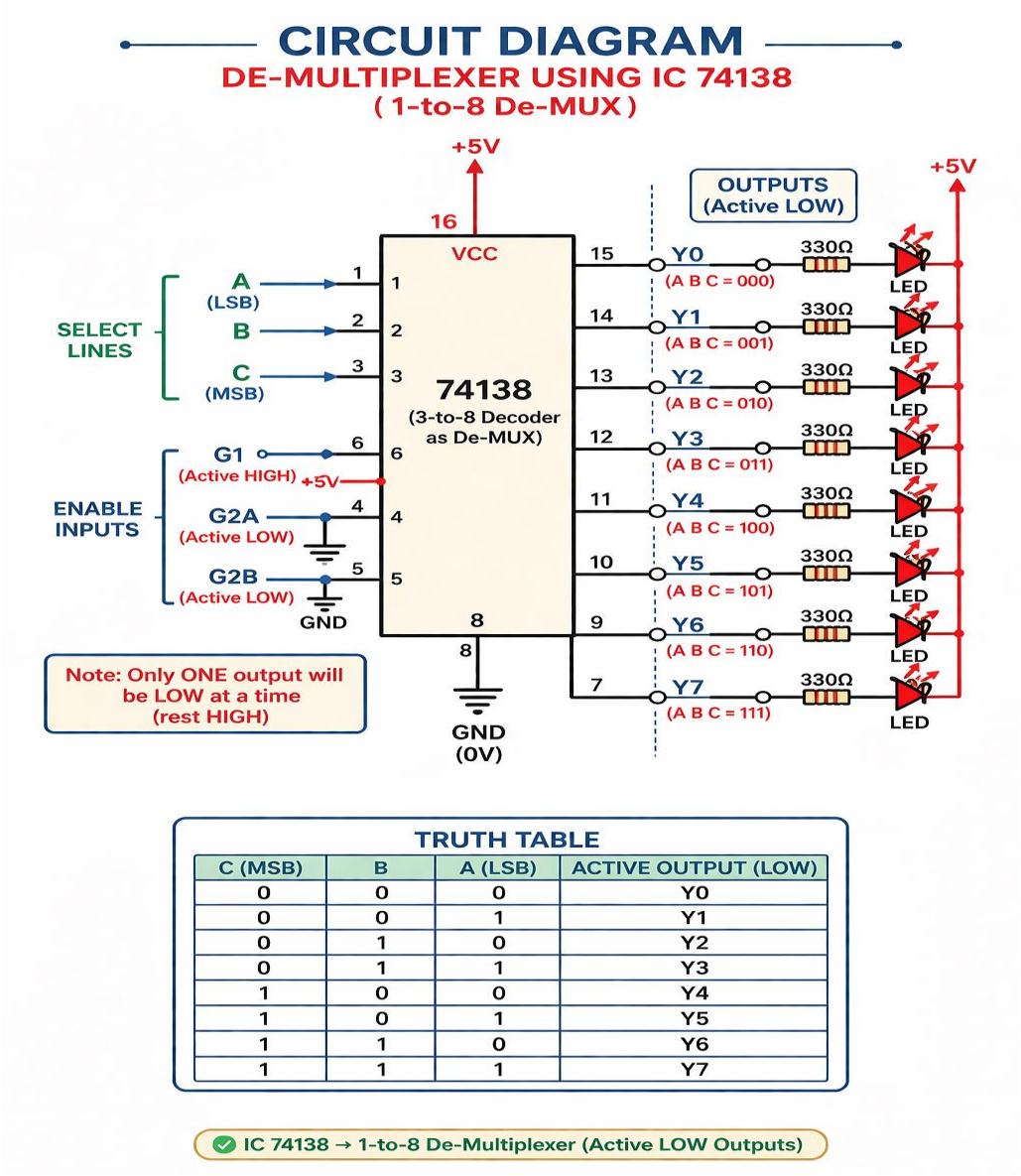
- One of its enable pins is used as **data input**
- Remaining enable pins are used for activation

Key Features

- 3 select lines → A, B, C
- 8 outputs → Y0 to Y7
- Outputs are **active LOW**

Working Principle

- Only **one output line becomes LOW (active)** at a time
- The selected output depends on **select inputs (A, B, C)**
- The input signal is routed to that output



TRUTH TABLE

C (MSB)	B	A (LSB)	ACTIVE OUTPUT (LOW)
0	0	0	Y0
0	0	1	Y1
0	1	0	Y2
0	1	1	Y3
1	0	0	Y4
1	0	1	Y5
1	1	0	Y6
1	1	1	Y7

IC 74138 → 1-to-8 De-Multiplexer (Active LOW Outputs)

Circuit Connections

1. Place IC on breadboard
2. Connect:
   - Vcc → +5V
   - GND → Ground
3. Enable pins:
   - G1 → HIGH
   - G2A, G2B → LOW
4. Use select lines:
   - A (LSB), B, C (MSB)
5. Connect outputs Y0–Y7 to LEDs via resistors

Procedure

1. Power ON the circuit
2. Apply input signal (logic HIGH/LOW)
3. Change select lines (A, B, C)
4. Observe which LED glows
5. Verify that only one output is active at a time

Observation Table

C	B	A	Active Output
0	0	0	Y0
0	0	1	Y1
0	1	0	Y2
0	1	1	Y3
1	0	0	Y4
1	0	1	Y5
1	1	0	Y6
1	1	1	Y7

Note: Output will be **LOW (LED ON)** due to active LOW outputs

Result

- The De-Multiplexer operation was successfully verified
- Only one output line is selected based on select inputs
- IC 74138 works effectively as a DEMUX

Precautions

- Ensure correct IC orientation
- Use proper voltage (+5V only)
- Avoid loose connections
- Check enable pins carefully