# EC 4010

# FAMILIARIZING WITH BOOLEAN ALGEBRA AND BINARY CODES

K.J.M.U.G.S. Eranda Jayasinghe

2021/E/075

GROUP CG8

EC4010

20.10.2023

**OBJECTIVE**:

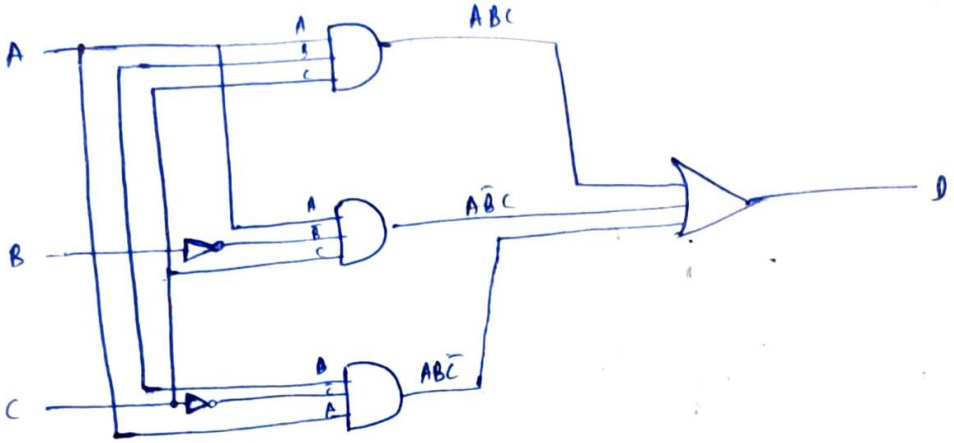
- Familiarizing with Boolean algebra and codes

**APPRATUS:**

- Logic gate ICs 7404, 7408, 7432 and 7486
- Bread board and wires
- LED

**TASK 01:**

a) $D = A \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$




b) $D = ABC + A\bar{B} \cdot C + AB\bar{C}$

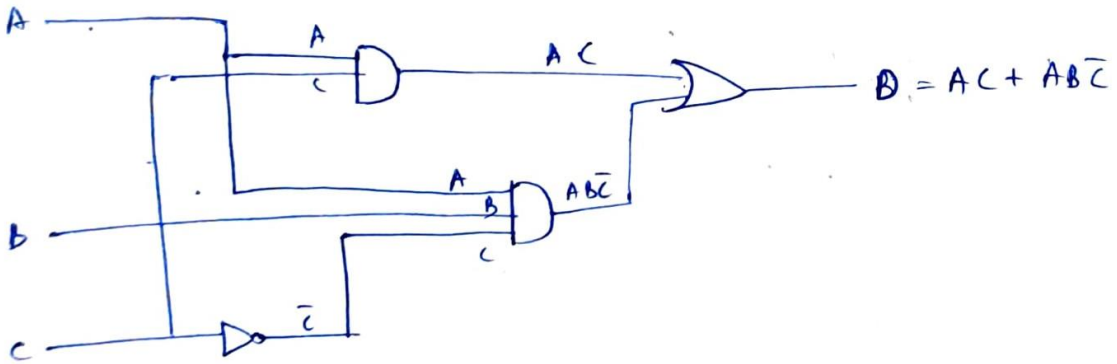
$D = AC(B + \bar{B}) + AB\bar{C}$

$D = AC + AB\bar{C}$

$D = A(B + C) = A \cdot B + A \cdot C$

$D = A \cdot B + A.C$

c)



## TABLE 01: Theoretical Truth Table

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

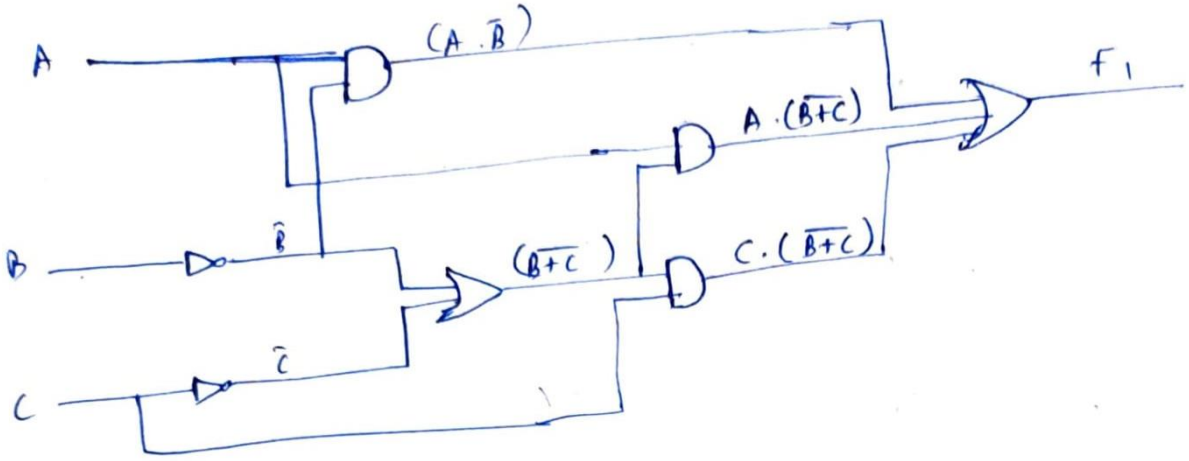
d. By using IC (7404,7408,7432), Breadboard, Wires ,LED, Resistor the circuit was completed.

## TABLE 02: Theoretical Truth Table

| A | B | C | OUTPUT |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

TASK 2:

2) $f_1 = A \cdot \bar{B} + A(\overline{B+C}) + C \cdot (\overline{B+C})$




$F_2 = A \cdot B \cdot C + \bar{C} + A\bar{B}C$

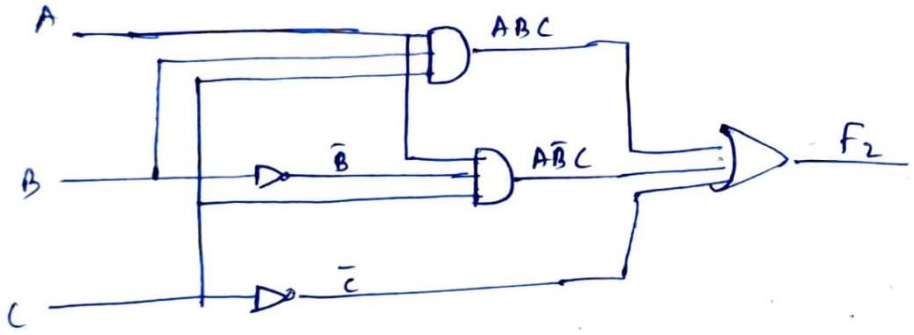



**Figure 03: Logic Circuit for F1 and F2**

## TABLE 02: Theoretical Truth Table for $F_1$

| A | B | C | Output |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

## TABLE 03: Theoretical Truth Table for $F_2$

| A | B | C | Output |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

C.

$$F_1 = A\bar{B} + A(\overline{B+C}) + C(\overline{B+C})$$

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

$$= A\bar{B}(1+\bar{C}) + 0$$

$$F_1 = A\bar{B}$$

$$f_2 = ABC + \bar{C} + A\bar{B}C$$

$$= AC(B+\bar{B}) + \bar{C}$$

$$f_2 = AC + \bar{C}$$

D.

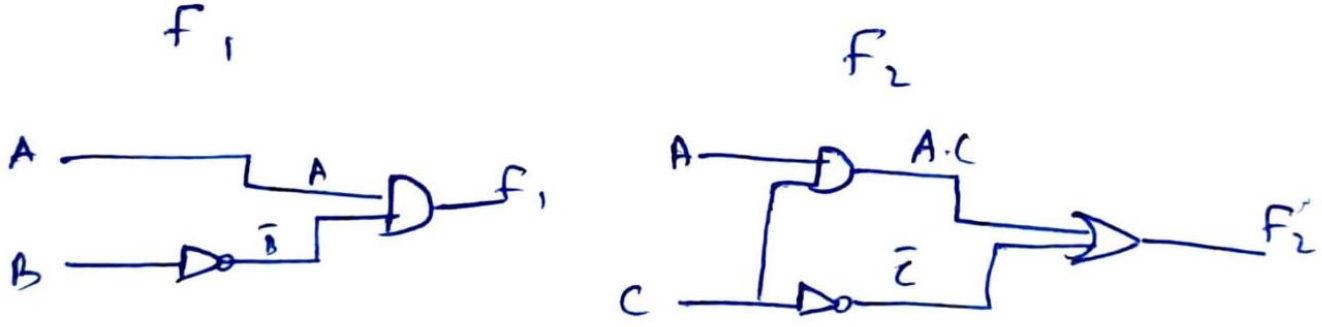



**Figure 05: Simplified Logic Circuit**

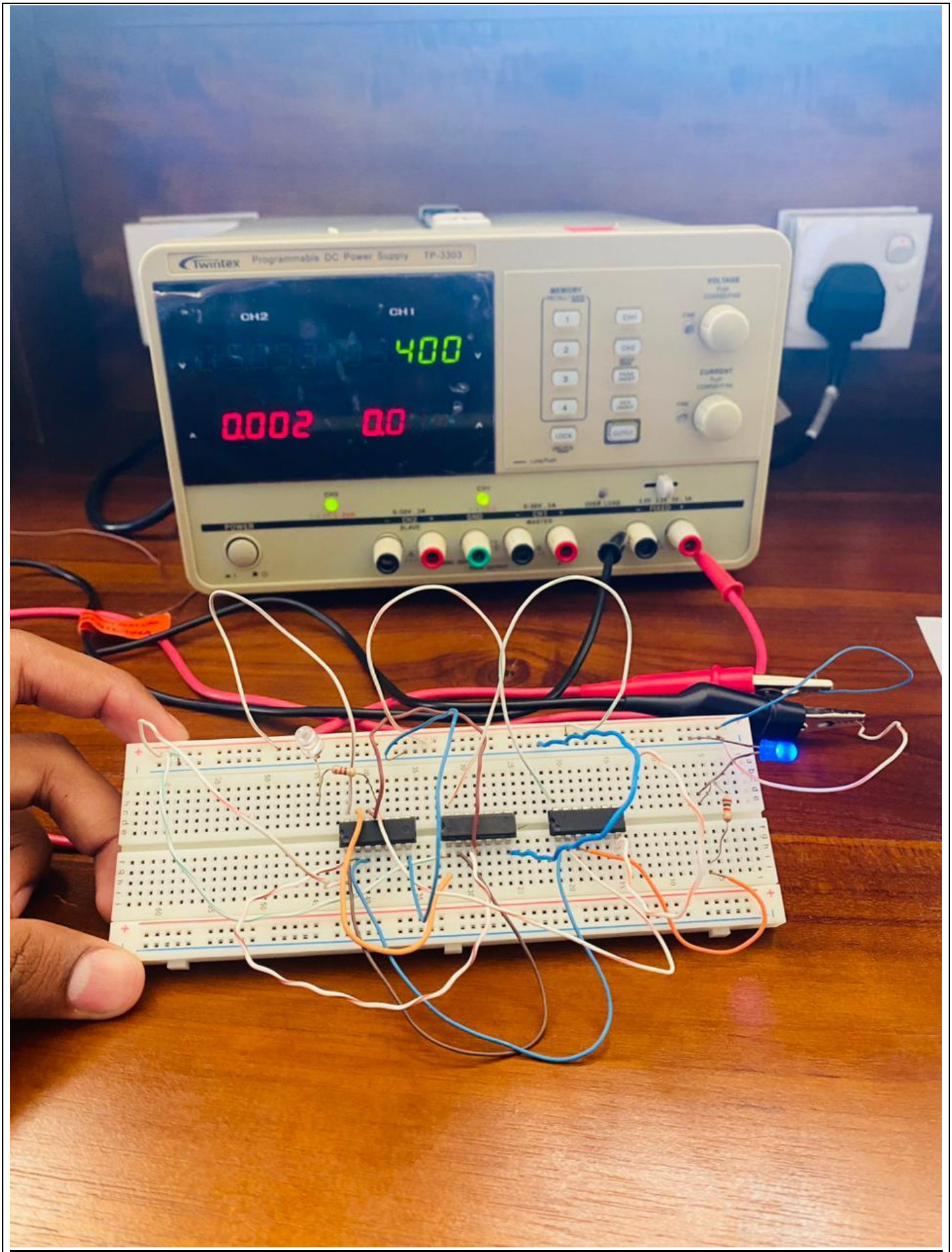
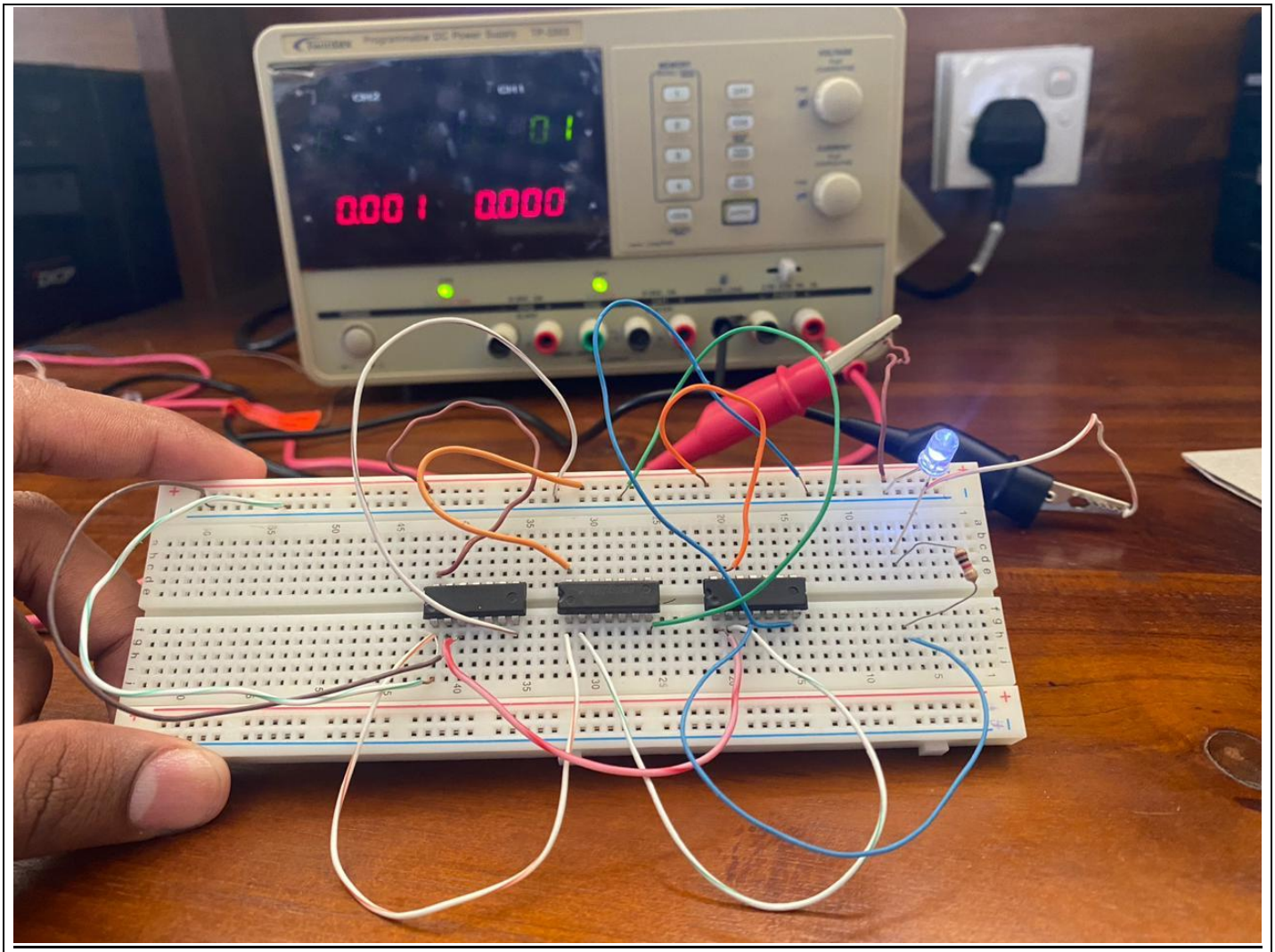



Figure 04: F1 Circuit on the Breadboard when Output indicates False


CH2
CH1
0 1
0.001
0.000

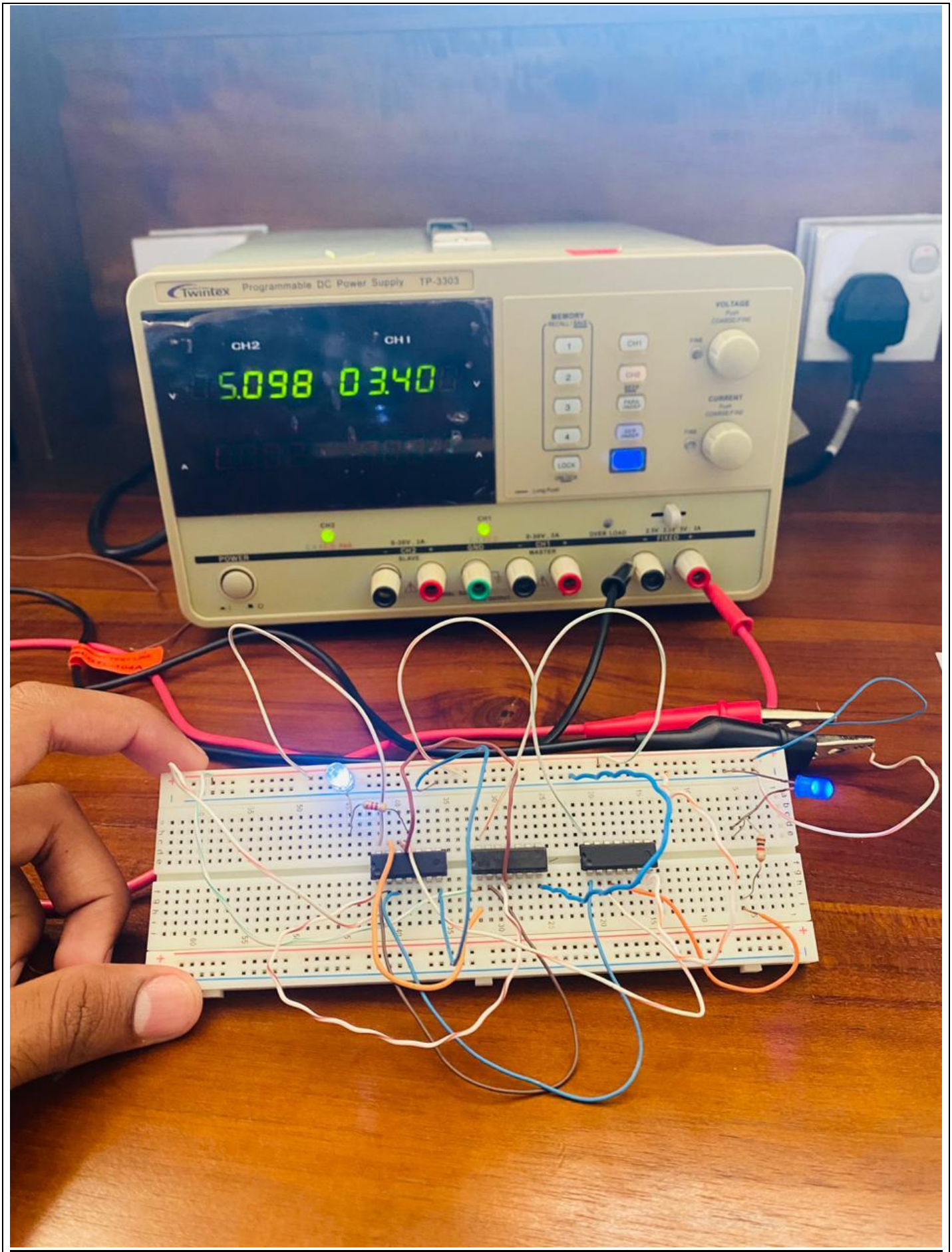



Figure 05: F2 Circuit on the Breadboard when Output indicates True

## TABLE 04: Experimental Truth Table for $F_1$

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

## TABLE 05: Experimental Truth Table for $F_2$

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

**TASK 03:**

a. **Table 06: Theoretical truth table for input and output values**

| A | B | D | Borrow |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

b.

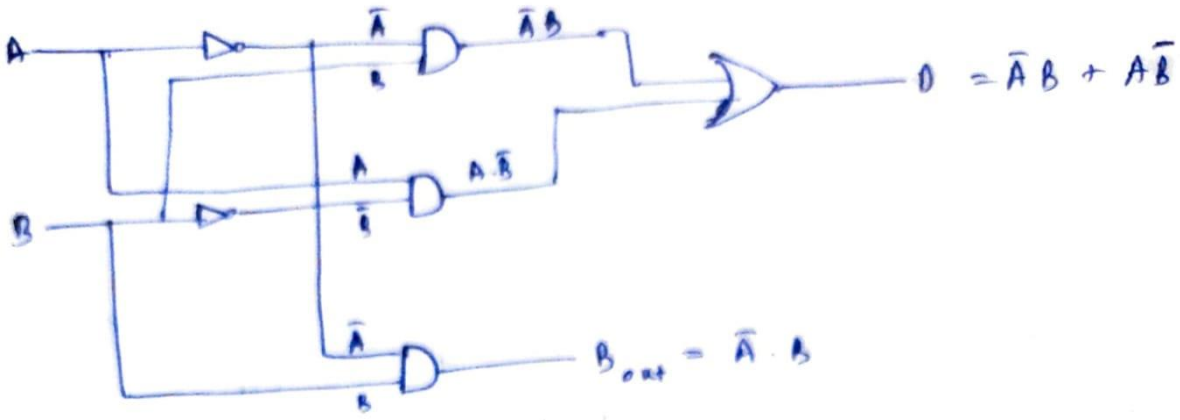
b. Difference (D) = $\bar{A}B + \bar{B}A$

Borrow ($B_{out}$) = $\bar{A}B$

c. $D = \bar{A}B + \bar{B}A$

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

d.



# DISCUSSION

## LOGIC GATES

1. AND Gate:
    - Behavior: Outputs 1 only if all inputs are 1; otherwise, it outputs 0.
    - Symbol: & (AND gate symbol)
    - Example: A AND B = Q (Q is 1 if both A and B are 1, otherwise 0)
2. OR Gate:
    - Behavior: Outputs 1 if at least one input is 1; it outputs 0 when all inputs are 0.
    - Symbol: ∨ (OR gate symbol)
    - Example: A OR B = Q (Q is 1 if A or B or both are 1, otherwise 0)
3. NOT Gate (Inverter):
    - Behavior: Produces the opposite value of the input (0 becomes 1, and 1 becomes 0).
    - Symbol: ¬ or a small circle at the gate input
    - Example: NOT A = Q (Q is the opposite of A)
4. NAND Gate:
    - Behavior: Outputs 0 only if all inputs are 1; otherwise, it outputs 1.
    - Symbol: A small circle on the output of the AND gate symbol
    - Example: A NAND B = Q (Q is 0 if both A and B are 1, otherwise 1)
5. NOR Gate:
    - Behavior: Outputs 1 only if all inputs are 0; otherwise, it outputs 0.
    - Symbol: A small circle on the output of the OR gate symbol
    - Example: A NOR B = Q (Q is 1 if both A and B are 0, otherwise 0)
6. XOR Gate (Exclusive OR):
    - Behavior: Outputs 1 when the number of 1s in its inputs is odd; outputs 0 when it's even.
    - Symbol: ⊕ or a special shape resembling a half-circle with a + inside
    - Example: A XOR B = Q (Q is 1 if A or B is 1, but not both; otherwise, 0)
7. XNOR Gate (Exclusive NOR):
    - Behavior: Outputs 1 when the number of 1s in its inputs is even; outputs 0 when it's odd.
    - Symbol: A small circle on the output of the XOR gate symbol
    - Example: A XNOR B = Q (Q is 1 if both A and B are the same, either both 1 or both 0, otherwise 0)

### Applications:

1. Digital Circuit Design:
    - Form the foundation of all digital systems and devices, such as computers, smartphones, and digital cameras.
2. Arithmetic Operations:
    - Used to perform addition, subtraction, multiplication, and division in digital arithmetic units.
3. Memory Units:
    - Utilized in building flip-flops and latches, fundamental in constructing memory elements within computers and digital systems.
4. Control Units:
    - Employed in microprocessors to manage the flow of instructions and data, coordinating various functions of a computer system.
5. Data Encryption:
    - Complex logic gates are employed in cryptographic algorithms, enhancing data security and ensuring confidentiality during transmission and storage.
6. Logic Controllers in Industrial Systems:
    - Utilized in industrial settings to create logic controllers for automation in manufacturing processes and machinery.
7. Signal Processing:

- Used in filtering, amplification, modulation, and demodulation of signals in various communication systems.

8. Robotics:
    - Embedded within control systems of robots, enabling decision-making processes based on various sensor inputs.
9. Automated Systems:
    - Found in automated gates, traffic light systems, and various sensors, ensuring efficient and controlled operations in modern infrastructures.
10. Consumer Electronics:
    - Incorporated into numerous everyday devices, such as TVs, remote controls, and kitchen appliances, to perform specific logical functions.

3. Advanced Logic Gates:

Advanced logic gates extend beyond the basic operations of AND, OR, NOT, and their derivatives, offering specialized functionalities. Multiplexers, used as data selectors, enable the routing of information from multiple inputs to a single output based on control signals, essential in memory addressing and data transmission systems. Demultiplexers perform the reverse function, directing a single input to one of many outputs. Encoders convert multiple inputs into a coded output, simplifying data representation, while decoders perform the inverse operation, translating coded inputs to multiple outputs. These advanced gates play pivotal roles in complex digital circuit designs, providing the capability to manipulate and control data, address memory elements, and perform intricate signal routing tasks fundamental to modern electronic systems.

## ❖ HALF SUBTRACTOR

A half subtractor is a combinational digital circuit used in digital electronics to subtract two single-bit binary numbers (A and B) and provide two outputs: the difference (D) and the borrow (B_out). It is a fundamental building block for more complex subtractors like full subtractors and is typically used in binary subtraction operations. Here's an explanation of the half subtractor's operation and its components:

1. Inputs:

A (Minuend): The first binary number, from which the subtraction is performed.
B (Subtrahend): The second binary number that is subtracted from the first.

2. Outputs:

- D (Difference): This output provides the result of subtracting B from A. It represents the difference between A and B for the respective bit positions.
- B_out (Borrow Out): This output indicates whether a borrow is required from the next lower significant bit when subtracting B from A. If B_out is 1, it means that a borrow is needed.

3. Truth Table:

The truth table for a half subtractor is as follows:

| A | B | D | B_Out |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |

- When both A and B are 0, there is no need for a borrow, and the difference is 0.
- When A is 0 and B is 1, a borrow is needed, and the difference is 1.
- When A is 1 and B is 0, no borrow is needed, and the difference is 1.
- When both A and B are 1, there is no need for a borrow, and the difference is 0.

4. Logic Diagram:

The logic diagram of a half subtractor consists of two basic logic gates: an XOR gate for the difference and an AND gate for the borrow output.

- The XOR gate computes the difference D as it performs binary subtraction. - The AND gate determines the borrow B_out based on the inputs A and B.

5. Applications:

Half subtractors are used in various digital systems where binary subtraction is required. They are the building blocks for more complex subtractors like full subtractors, which can handle multi-bit binary subtraction. Full subtractors incorporate additional inputs for handling the borrow from the previous stage in multi-bit subtraction operations.

In summary, a half subtractor is a basic digital circuit used to subtract two single-bit binary numbers, providing the difference and the borrow outputs. It plays a fundamental role in binary subtraction operations and serves as the basis for more advanced subtractor circuits.