Sub-Group: A-7
Experiment 4: Study of Boolean Algebra

Sayan Karmakar

22MS163

1 Aim:

To study Boolean algebra by verifying them in circuits using logic gates (integrated circuits.)

2 Theory:

In digital electronics signals are represented by discrete bands of analog levels, rather than continuous signals as commonly seen in analog devices, In most cases in digital electronics, there are only two discrete levels. One of them is 0 volts. And the other one is 5 Volts. Voltage value of 0 corresponds to "false" or "0" or "off" state, and the voltage value of 5 volts corresponds to "true" or "1" or "on" state. These states satisfies the Boolean algebra.

Boolean algebra is set of rules of that we can use to do arithmetic with inputs 1 and 0. The rules are given for two inputs and the outcome of the operations are normally truth tables where all the inputs and outputs are presented. The basic boolean functions are NOT, AND, OR, XOR, NAND, NOR gates. These gates are explained below.

2.1 NOT gate:

NOT gate is single input and single output gate. It takes one value and gives the negation of the input. So, if the input is 1, it returns 0 and if the input is 0, then it returns 1. If the input is A, then NOT operation on $\mathbf{A}$ as $\overline{\mathbf{A}}$.
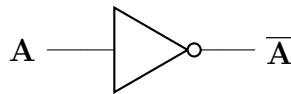

Figure 1: Representation of NOT Gate

2.2 AND gate:

AND gate takes two input and returns one output. If both the inputs are 1 then it returns 1 and if even one of the inputs is 0 then the output is 0. If the inputs are **A** and **B**, we represent the operation as **A** · **B**.
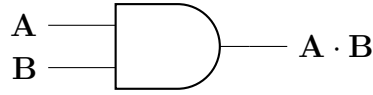

Figure 2: Representation of AND Gate

2.3 OR gate:

OR gate is also a gate with two inputs and one output. If both the inputs of OR gates are 0 only then the output is 0, otherwise the output is 1. It is represented as **A** + **B** if the inputs are **A** and **B**.
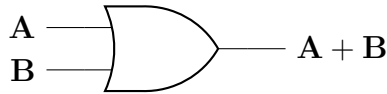

Figure 3: Representation of OR Gate

2.4 XOR gate:

XOR gate also takes two inputs and returns one output. It represented as **A** ⊕ **B** if the inputs are **A** and **B**. It returns 0 if both the inputs are same. Analytical expression of the XOR gate is $\mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B}$.
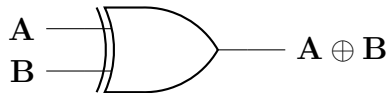

Figure 4: Representation of XOR Gate

2.5 NAND gate:

NAND gate is the composition of AND and NOT gate. So taking the inputs it gives the negation of the AND output. For inputs **A** and **B**, the ouput is represented as $\overline{\mathbf{A} \cdot \mathbf{B}}$.
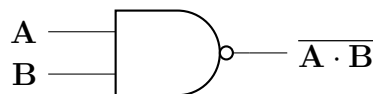

Figure 5: Representation of NAND Gate

2.6   NOR gate:

Similar to NAND gate, NOR gate is also composition of two gates. These gates are NOT gate and AND gate. NOR gate returns the negation of OR output. For inputs **A** and **B**, the output is denoted as $\overline{\mathbf{A} + \mathbf{B}}$
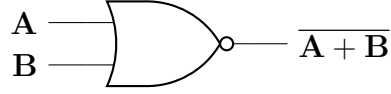

Figure 6: Representation of NOR Gate

A	**B**	**A** $\cdot$ **B**	**A** + **B**	$\overline{\mathbf{AB}}$	$\overline{\mathbf{A} + \mathbf{B}}$	**A** $\oplus$ **B**
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

Table 1: Truth Table for the Logic Gates

This logic gates are implemented in circuits using different integrated circuits (IC). Different ICs have pin-configurations for different logic gates. The ICs are:

- IC7408 is used for AND gate.

- IC7432 is used for OR gate.

- IC7404 is used for NOT gate.

- IC7400 is used for NAND gate.

- IC7402 is used for NOR gate.

- IC7486 is used for XOR gate.

3   Verification of Boolean Algebra:

To verify the rules of Boolean algebra we created three circuits in bread board using ICs as the logic gates. From the circuit we experimentally computed the truth table. For each of them we also computed the truth table just using boolean algebra. And then we verified if the results are consistent or not. The circuits are shown below.
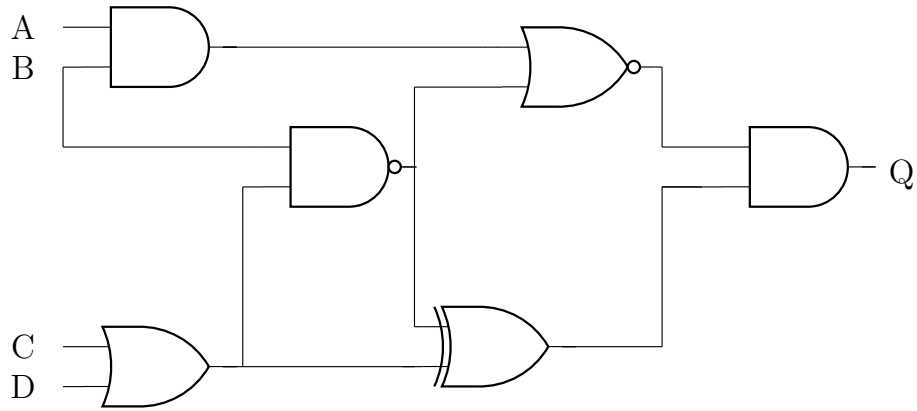
3.1 Circuit 1:

Figure 7: Circuit 1

Using a LED, we calculated the result of different inputs for this circuit. 1 corresponds to LED being on and 0 corresponds to LED being off. Truth table of the circuit is shown below. This truth table matches with the truth table computed using Boolean algebra.

A	B	C	D	Q
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	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	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Table 2: Truth table for Circuit 1
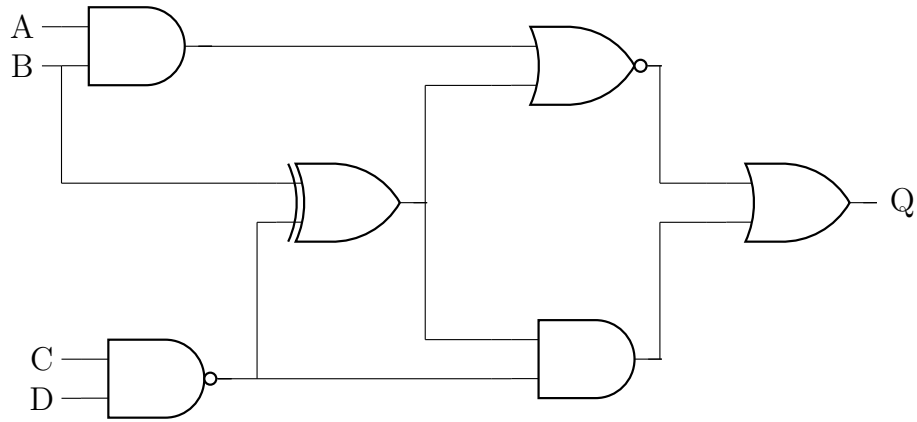
3.2 Circuit 2:

Figure 8: Circuit 2

Using different ICs, this circuit was made on a breadboard. The output was measured using a LED. The truth table for this circuit is captured in the following table. This matches our expected results.

A	B	C	D	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
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	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Table 3: Truth table for Circuit 2
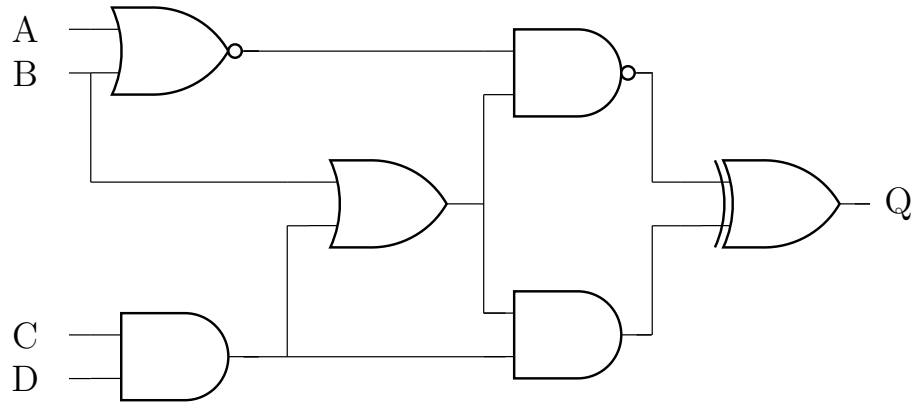
3.3 Circuit 3:

Figure 9: Circuit 3

We created this circuit in breadboard using different ICs. Using a LED we checked the output of the circuit for different inputs. If the LED is on then the output is 1 and if the LED is off, then the output was 0. The truth table from the circuit is shown below. This truth table matched with the truth table from the theoretical calculation.

A	B	C	D	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
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	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

4   Discussion:

During this experiment, we encountered a faulty IC which returned wrong outputs. Also initially we did not add any resistance along with the LED. The output current was more than the maximum LED current. This burnt the LED. After the resistance was added the circuit worked.

5   Conclusion:

In this experiment, we studied Boolean algebra and verified the the rules for three different circuits. The output of the circuit matched the theoretical calculations for the circuits.