Experiment 04: Study of Boolean Algebra

Sagnik Seth - 22MS026
(Subgroup - A7)

1 Aim

To study the basic laws of Boolean Algebra and verify them using implementation of logic gates using Integrated Circuits.

2 Theory

Boolean Algebra deals with logical operations of binary variables. A Boolean function f is generally defined as $f : \{0,1\}^k \to \{0,1\}$, that is, it takes input from k cartesian products of the set $\{0,1\}$ and returns a single output which can be 0 or 1. The basic boolean functions are:

- **NOT Gate**

  It is a unary operator which takes a single input and returns the negation of the input. It is represented by the symbol $\overline{\mathrm{A}}$. If input is 1, it return 0 and if input is 0, it returns 1.

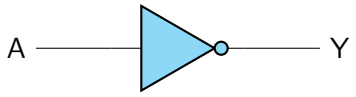

Figure 1: Representation of NOT Gate

- **AND Gate**

  It is a binary operator which takes two inputs and returns the logical AND of the inputs. It is represented by the symbol $\mathrm{A} \cdot \mathrm{B}$. It returns 1 only if both inputs are 1, otherwise it returns 0.

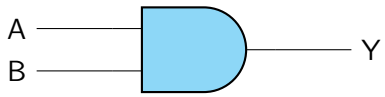

Figure 2: Representation of AND Gate

- **OR Gate**

  It is a binary operator which takes two inputs and returns the logical OR of the inputs. It is represented by the symbol $\mathrm{A} + \mathrm{B}$. It returns 1 if any of the inputs is 1, otherwise it returns 0.

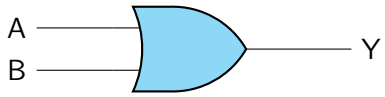

Figure 3: Representation of OR Gate

- **XOR Gate**

  It is a binary operator which takes two inputs and returns the logical XOR of the inputs. It is represented by the symbol $\mathrm{A} \oplus \mathrm{B}$. It can be showed that the expression for XOR is equivalent to $\overline{\mathrm{A}}\mathrm{B} + \mathrm{A}\overline{\mathrm{B}}$. It returns 1 if the inputs are different, otherwise it returns 0.

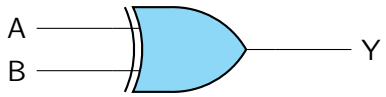

Figure 4: Representation of XOR Gate

- **NAND Gate**

  It is a binary operator which takes two inputs and returns the logical NAND of the inputs. It is represented by the symbol $\overline{\mathrm{AB}}$. Thus, it is a composition of AND and NOT operations. From de Morgan's law, it can be shown that NAND is equivalent to $\overline{\mathrm{A}} + \overline{\mathrm{B}}$.

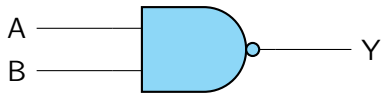

Figure 5: Representation of NAND Gate

- **NOR Gate** It is a binary operator which takes two inputs and returns the logical NOR of the inputs. It is represented by the symbol $\overline{\mathrm{A} + \mathrm{B}}$. Thus, it is a composition of OR and NOT operations. From de Morgan's law, it can be shown that NOR is equivalent to $\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}$.

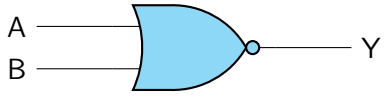

Figure 6: Representation of NOR Gate

A	B	AB	$A+B$	$\overline{AB}$	$\overline{A+B}$	$A \oplus B$
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 above logical operations

The logic gates can be implemented using Integrated Circuit (IC). They generally have 14 pins and multiple gates of the same kind are implemented on the same IC chip. The ICs used in this experiment are:

- IC 7408 (AND Gate)
- IC 7432 (OR Gate)
- IC 7404 (NOT Gate)
- IC 7400 (NAND Gate)
- IC 7402 (NOR Gate)

- IC 7486 (XOR Gate)

Each of these chips have a V_{CC} and Ground terminal which connects it to the main circuit.

3 Verification of Truth Tables and Observations

Example 1:

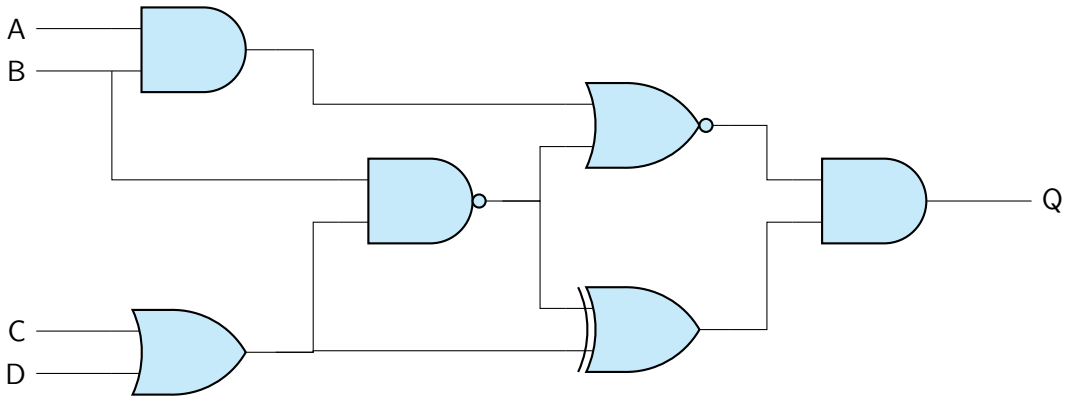

Figure 7: Logic Gate circuit diagram for for Example 1

The analytic expression for the above circuit was found out to be:

$$\mathrm{Q} = \overline{\mathrm{A}}\mathrm{B}(\mathrm{C} + \mathrm{D})$$

Table 2: Truth Table for $\mathrm{Q} = \overline{\mathrm{A}}\mathrm{B}(\mathrm{C} + \mathrm{D})$

A	B	C	D	Y
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

We created the circuit on the breadboard using the integrated circuits and connected all the components. Then, using an LED Bulb, we verified the above truth table for the circuit shown. For all configurations of the 4 inputs, the output in the circuit matched with the theoretical truth table, that is, the LED bulb glowed when the output was 1 and it did not glow when the output was 0.

Example 2:

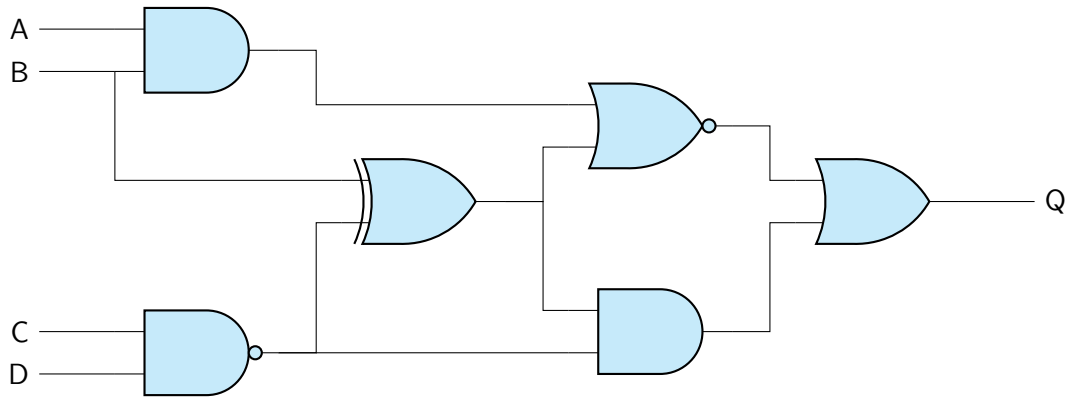

Figure 8: Logic Gate circuit diagram for for Example 2

The analytic expression for the above circuit was found out to be:

$$\mathrm{Q} = \overline{\mathrm{A}}\,\overline{\mathrm{C}} + \overline{\mathrm{A}}\,\overline{\mathrm{D}} + \overline{\mathrm{B}}$$

Table 3: Truth Table for $\mathrm{Q} = \overline{\mathrm{A}}\,\overline{\mathrm{C}} + \overline{\mathrm{A}}\,\overline{\mathrm{D}} + \overline{\mathrm{B}}$

A	B	C	D	Y
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

We created the circuit on the breadboard using the integrated circuits and connected all the components. Then, using an LED Bulb, we verified the above truth table for the circuit shown. For all configurations of the 4 inputs, the output in the circuit matched with the theoretical truth table, that is, the LED bulb glowed when the output was 1 and it did not glow when the output was 0.

Example 3:

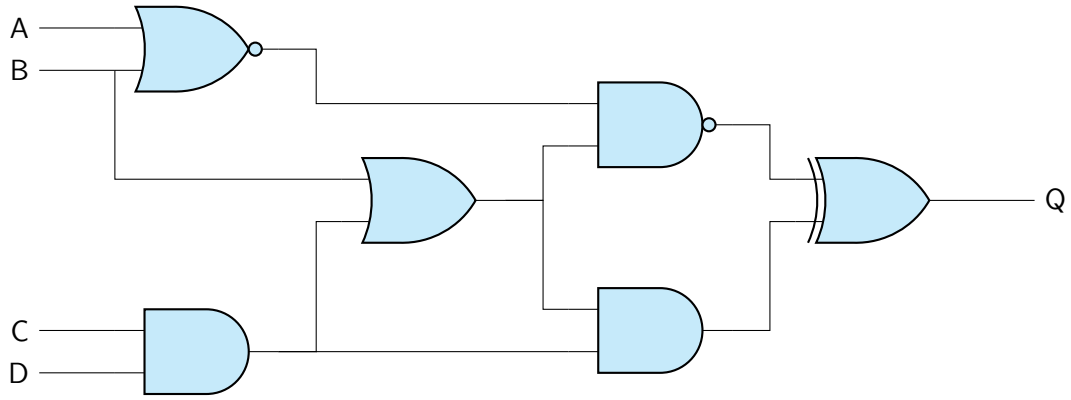

Figure 9: Logic Gate circuit diagram for for Example 3

The analytic expression for the above circuit was found out to be:

$$\mathrm{Q} = \overline{\mathrm{A}}\,\overline{\mathrm{B}} + \overline{\mathrm{C}} + \overline{\mathrm{D}}$$

Table 4: Truth Table for $\mathrm{Q} = \overline{\mathrm{A}}\,\overline{\mathrm{B}} + \overline{\mathrm{D}} + \overline{\mathrm{C}}$

A	B	C	D	Y
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

We created the circuit on the breadboard using the integrated circuits and connected all the components. Then, using an LED Bulb, we verified the above truth table for the circuit shown. For all configurations of the 4 inputs, the output in the circuit matched with the theoretical truth table, that is, the LED bulb glowed when the output was 1 and it did not glow when the output was 0.

4 Sources of Error

5 Discussion and Conclusion