PH3204: Electronics Lab

Study of Boolean Algebra using Logic Gates

Sabarno Saha

22MS037

Grp. B-10

Indian Institute of Science Education and Research, Kolkata,
Mohanpur, West Bengal, 741246, India.

Contents

1 Aim

In this experiment, we aim to study the basic laws of Boolean algebra using logic gates. The Logic gates we will be using are AND, OR, NOT, NAND, NOR, and XOR gates. We will also study the properties of these gates and their applications in digital circuits. Then we will build a few circuits using these gates and verify their truth tables.

2 Theory

Boolean algebra is just the study of functions that map binary strings to either 0 or 1. The basic operations in Boolean algebra are

AND: $Q = A \cdot B$ OR: $Q = A + B$ NOT: $Q = \overline{A}$
We will be studying the properties of these gates and their applications in digital circuits. We will also study the NAND, NOR, and XOR gates.

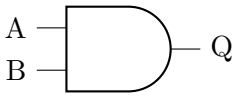
2.1 Gates

2.1.1 AND Gate

The AND gate is a digital logic gate that implements logical multiplication. We represent the AND Gate using $Q = A \cdot B = AB$. It behaves according to the trieeestd uth table shown below:

A	B	Q=AB
0	0	0
0	1	0
1	0	0
1	1	1

(a) Truth Table of AND Gate



(b) Symbol of AND Gate

Figure 1: Truth Table and Symbol of AND Gate

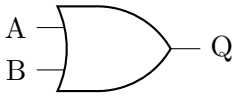
We use the IC 7408 to implement the AND gate.

2.1.2 OR Gate

The OR Gate is a digital logic gate that implements logical addition. We represent the OR Gate using $Q = A + B$. The truth table for the OR gate is shown below:

A	B	Q=A+B
0	0	0
0	1	1
1	0	1
1	1	1

(a) Truth Table of OR Gate



(b) Symbol of OR Gate

Figure 2: Truth Table and Symbol of OR Gate

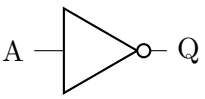
We use the IC 7432 to implement the OR gate.

2.1.3 NOT Gate

The NOT Gate is a digital logic gate that implements logical negation. We represent the NOT Gate using $Q = \overline{A}$. The truth table for the NOT gate is shown below:

A	Q=$\overline{A}$
0	1
1	0

(a) Truth Table of NOT Gate



(b) Symbol of NOT Gate

Figure 3: Truth Table and Symbol of NOT Gate

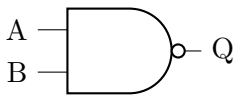
We use the IC 7404 to implement the NOT gate.

2.1.4 NAND Gate

The NAND Gate is a digital logic gate that implements the logical negation of the logical multiplication. We represent the NAND Gate using $Q = \overline{AB}$. This gate just inverts the output of the AND gate. The truth table for the NAND gate is shown below:

A	B	Q=$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

(a) Truth Table of NAND Gate



(b) Symbol of NAND Gate

Figure 4: Truth Table and Symbol of NAND Gate

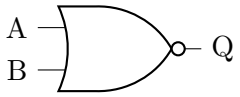
We use the IC 7400 to implement the NAND gate.

2.1.5 NOR Gate

The NOR Gate is a digital logic gate that implements the logical negation of the logical addition. We represent the NOR Gate using $Q = \overline{A + B}$. This gate just inverts the output of the OR gate. The truth table for the NOR gate is shown below:

A	B	Q=$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

(a) Truth Table of NOR Gate



(b) Symbol of NOR Gate

Figure 5: Truth Table and Symbol of NOR Gate

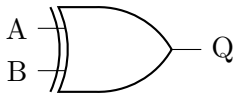
We use the IC 7402 to implement the NOR gate.

2.1.6 XOR Gate

The XOR Gate is a digital logic gate that implements the logical addition modulo 2. We represent the XOR Gate using $Q = A \oplus B$. The boolean expression for the XOR gate is $Q = A\overline{B} + \overline{A}B$. The truth table for the XOR gate is shown below:

A	B	Q=$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(a) Truth Table of XOR Gate



(b) Symbol of XOR Gate

Figure 6: Truth Table and Symbol of XOR Gate

We use the IC 7486 to implement the XOR gate.

2.2 Pin diagrams of ICs

The ICs used here are the 74LSXX series, where LS stands for Low Power Schottky. The pin diagrams of the ICs are shown below:

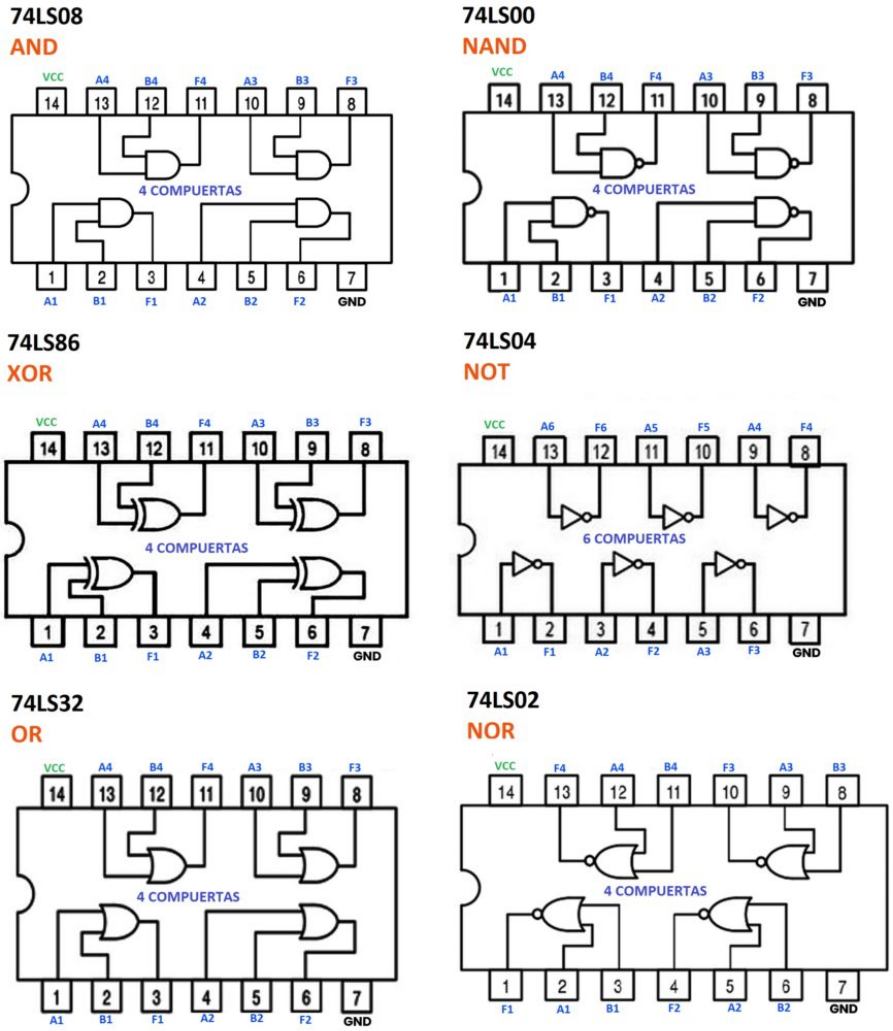


Figure 7: Pin Diagram of IC 7400. (Source : The Internet)

3 Experiment

3.1 Example 1

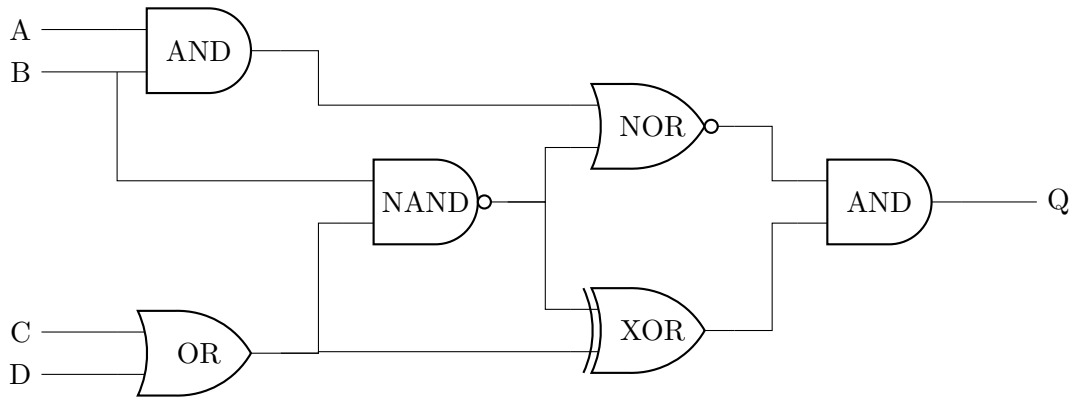


Figure 8: Circuit Diagram of the given circuit

The Simplified boolean expression for the circuit is $Q = \overline{A}B(C + D)$. The truth table for the circuit is shown below:

Table 1: Truth Table for $Q = \overline{A}B(C + D)$

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

3.2 Example 2

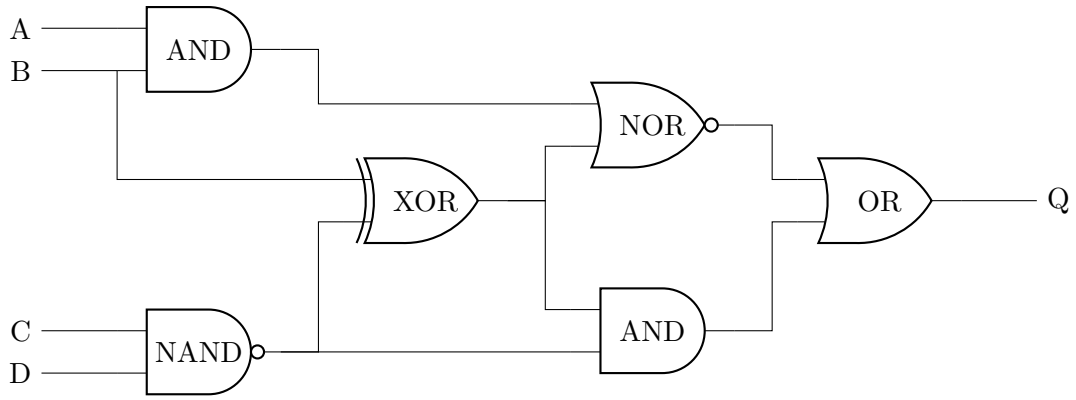


Figure 9: Circuit Diagram of the given circuit

The Simplified boolean expression for the above circuit was found out to be $Q = \overline{B} + \overline{A}B(\overline{C} + \overline{D})$

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

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

3.3 Example 3

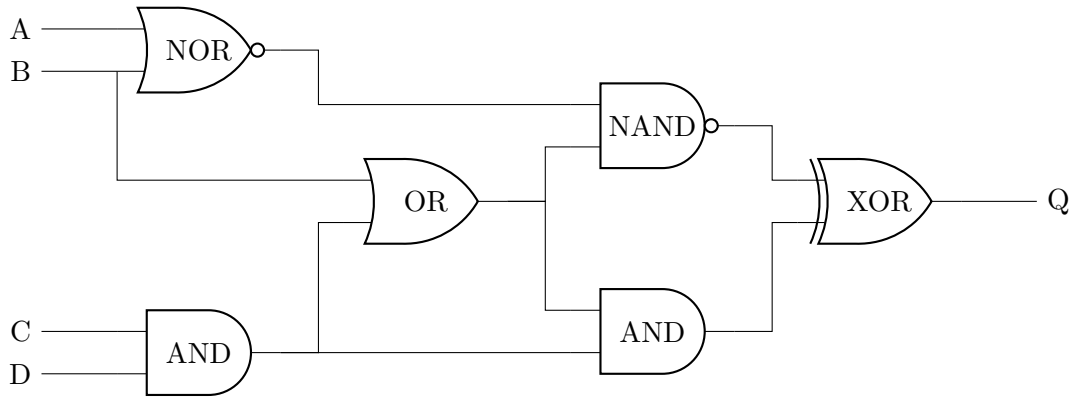


Figure 10: Circuit Diagram of the given circuit

The simplified boolean expression for the above circuit was found out to be $Q = \overline{A}\ \overline{B} + \overline{C} + \overline{D}$.

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

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

4 Results

The truth tables for the individual gates were verified and the simplified boolean expressions for the circuits were found out. Then the truth tables were also experimentally verified.

5 Conclusion

We conclude by verifying the truth tables of the individual gates and the circuits.