1 Digital Logic

Digital logic is a foundational concept in electronics, serving as the basis for decision-making and control in circuits. Logic gates are the building blocks of digital systems and are used extensively in computation, automation, and complex decision-making processes. Understanding these gates and how they combine enables the creation of systems that respond to specific sets of conditions, which is essential in fields from computing to physics and engineering.

Logic in digital electronics is the study of true and false values represented by 1s and 0s, respectively. This binary system allows complex problems to be broken down into simple "yes" or "no" (1 or 0) decisions. Logic gates are hardware implementations of Boolean functions that allow for manipulation of these binary signals.

Logic serves as the foundation for everything from conditional statements in code, to decision points in experiments, to theoretical modeling. In electronics, logic enables the construction of systems that perform calculations, control responses, and enable communication.

The language of digital systems is "Binary". In binary numbers and states are represented with just two values: 1 and 0. Each logic gate operates based on Boolean algebra (statements of truth), where these two binary states can be combined to achieve specific results. Moreover, counting in binary is crucial for understanding how data is processed within a computer, where each bit represents a power of 2.

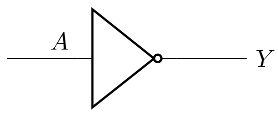
Logic Gates

Logic gates perform fundamental operations that form the basis of digital circuits. Each gate has a unique symbol, truth table, and Boolean expression, representing its behavior in a circuit.

NOT Gate (Inverter)

The NOT gate outputs the inverse of its input. When the input is 1, the output is 0, and vice versa.

$$Y = \overline{A}$$



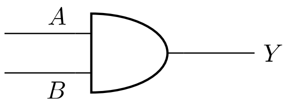
AND Gate

The AND gate outputs 1 only when both inputs are 1.

A	Y
0	1
1	0

Table 1: Truth Table for NOT Gate

$$Y = A \wedge B$$



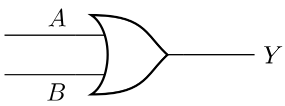
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Table 2: Truth Table for AND Gate

OR Gate

The OR gate outputs 1 if at least one input is 1.

$$Y = A \vee B$$



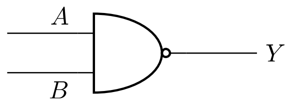
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Table 3: Truth Table for OR Gate

NAND Gate

The NAND gate outputs 1 in all cases except when both inputs are 1.

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



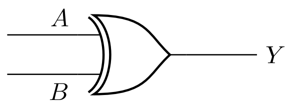
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Table 4: Truth Table for NAND Gate

XOR

The XOR gate outputs 1 if the inputs are different and a 0 if they are the same.

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



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

Table 5: Truth Table for NAND Gate

Combining Logic Gates for Real-World Applications

Example 1: Light Control System

In an energy-saving lighting system, the light turns on only when it is dark and motion is detected. This system combines an **AND gate** to ensure both inputs (darkness and motion) are 1.

Example 2: Alarm System

A building's alarm system uses multiple sensors (doors, windows, motion). An **OR gate** activates the alarm if any input is triggered. Using a **NAND gate** with a master switch can disable the alarm when needed.

Example 3: Voting System in Fault-Tolerant Computing

In systems requiring majority voting (e.g., safety systems), combine inputs with **AND** and **OR gates** to create a circuit that outputs 1 if at least two sensors agree.

Example 4: Password Lock System

For a simple password lock, combine switches using **AND** and **NOT gates** to open the lock only when a specific combination is met, e.g.,

$$Y = A \wedge \overline{B} \wedge C$$

where A = 1, B = 0, and C = 1.

Example 5: Half Adder

The half adder is a fundamental digital circuit that adds two single-bit binary numbers, producing a sum and a carry-out as outputs. It implements basic logical operations using AND and XOR gates to generate these outputs.

1. Inputs and Outputs

   For two input bits A and B: - The sum output S should be 1 if only one of the inputs is 1 (exclusive OR). - The carry output C should be 1 if both inputs are 1 (AND).

2. Logic Equations

   The logic operations for the sum (S) and carry (C) of the half adder can be expressed as:

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

   where:

   - $\oplus$ represents the XOR operation.
   - $\wedge$ represents the AND operation.

3. Truth Table

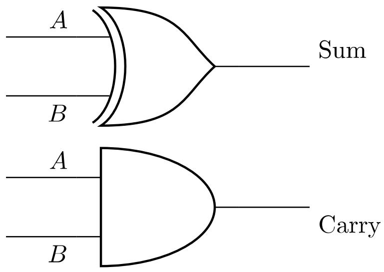
   The truth table for the half adder is as follows:

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

This truth table confirms the half adder logic: the sum is 1 only when A and B are different, and the carry is 1 only when both A and B are 1.

4. Circuit Diagram

   The half adder circuit can be constructed using an XOR gate for the sum and an AND gate for the carry.



5. Explanation of the Logic

   - **XOR Gate for Sum**: The XOR gate outputs 1 only when the inputs are different, giving us the sum output S.
   - **AND Gate for Carry**: The AND gate outputs 1 only when both inputs are 1, giving us the carry output C.

Together, these gates perform the half adder function, making it possible to add two single-bit binary numbers with outputs for both the sum and the carry.

Conclusion

This chapter introduced the basic concepts of digital logic, covering fundamental gates and showing their applications in real-world systems. Logic gates enable complex decision-making and are central to digital circuits, from simple locks to fault-tolerant systems.