College of Sciences
Intelligent Medical System Department

Lecture 1:

Introduction of Logic Design

Subject: Logic Design
Level: First
Semester: second
Lecturer: Asst. Lect. Ali Saleem Haleem

Google Class Room

Introduction

Logic design is the foundational subject that explains how digital systems represent, process, and control information. Before studying logic gates, Boolean algebra, combinational circuits, and sequential circuits, students must first understand the basic language of digital electronics: signals, bits, logic levels, number systems, and simple logical operations.

According to the uploaded syllabus, the first week introduces six core ideas: digital and analog quantities, binary digits, logic levels and digital waveforms, basic logic operations, decimal numbers, binary numbers, and decimal-to-binary conversion. These topics are essential because they connect physical electrical signals with symbolic logical values inside digital hardware.

This lecture therefore serves as the conceptual entry point to the course. It builds the academic background required for later topics such as logic gates, truth tables, simplification methods, Karnaugh maps, and circuit implementation.

Learning Outcomes

• Define analog quantity and digital quantity accurately.
• Explain the meaning of a binary digit (bit), a logic level, and a digital waveform.
• Identify the role of the basic logic operations used in digital systems.
• Distinguish clearly between decimal and binary number systems.
• Convert positive decimal integers into binary form using the repeated division-by-2 method.

Digital and Analog Quantities

A quantity is any measurable physical value such as voltage, temperature, pressure, speed, or sound intensity. In electronics and computing, information is usually represented in one of two forms: ***analog*** or ***digital***.

An analog quantity changes continuously and may assume any value within a specific range. Examples include body temperature, microphone output, light intensity, and the level of water in a tank. Because analog change is smooth, it resembles many real-world phenomena directly.

A digital quantity changes in discrete steps and is restricted to a limited set of values. In most digital systems, the two principal states are 0 and 1. This representation is preferred in computing because it is easier to store, transmit, copy, and process with high reliability and better noise immunity.

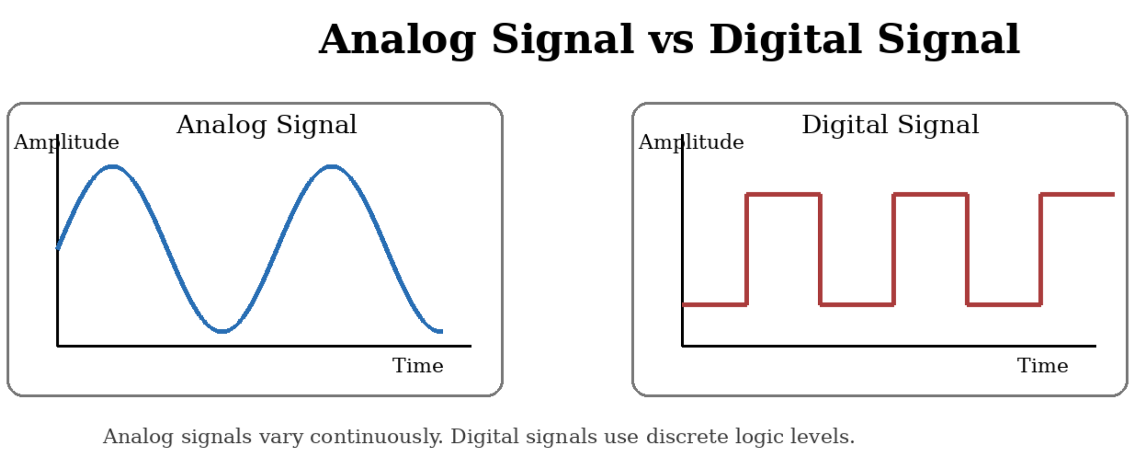

Figure 1. Continuous analog variation compared with discrete digital switching.

Comparison between Analog and Digital Quantities

Aspect	Analog Quantity	Digital Quantity
Nature of values	Continuous values	Discrete values
Typical states	Infinite values within a range	Usually two logic states: 0 and 1
Examples	Temperature, sound, and voltage level	Binary data in computers and controllers
Noise sensitivity	More sensitive to noise and distortion	More robust within valid thresholds

Binary Digits, Logic Levels, and Digital Waveforms

The smallest unit of information in a digital system is the binary digit, or bit. A bit has only two possible values: 0 or 1. All digital data, whether it represents a number, a character, an image, or a control signal, is ultimately stored and processed as combinations of bits.

A logic level is the electrical interpretation assigned to a binary value. In practice, logic 0 does not always equal exactly 0 V, and logic 1 does not always equal exactly 5 V or 3.3 V. Instead, digital families define acceptable voltage ranges for LOW and HIGH levels. A circuit operates correctly when the input voltage falls inside the valid range.

A digital waveform is a time-based representation of how a digital signal changes between LOW and HIGH states. It is used to analyze timing, transitions, pulse width, and signal behavior in digital circuits.

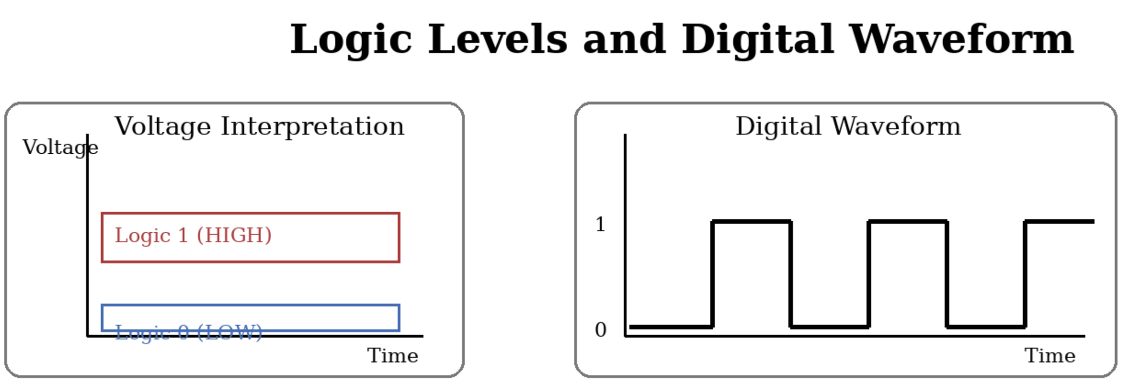

Figure 2. Logic levels are interpreted from voltage ranges and observed as digital waveforms over time.

Key Terms

- Bit: the smallest unit of digital information; it can be either 0 or 1.
- Logic 0: the LOW state in a digital circuit.
- Logic 1: the HIGH state in a digital circuit.
- Digital waveform: a graphical description of a signal changing with time between valid logic levels.

Basic Logic Operations

Logic operations form the basis of decision-making in digital systems. A logic operation accepts one or more binary inputs and produces a binary output. In this introductory lecture, the most important operations are NOT, AND, OR, and XOR.

- **NOT operation:** A unary operation because it uses only one input. It produces the complement of the input. If A = 1, then NOT A = 0. If A = 0, then NOT A = 1.
- **AND operation:** The output is 1 only when all inputs are 1. For a two-input AND gate, if either input is 0, the output becomes 0.
- **OR operation:** The output is 1 when at least one input is 1. The output is 0 only when all inputs are 0.
- **XOR operation:** The output is 1 when the inputs are different, and 0 when the inputs are the same.

Truth Table for Basic Operations

A	B	NOT A	A AND B	A OR B	A XOR B
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Decimal Numbers

The decimal number system is the number system used in everyday life. It is called a base-10 system because it uses ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Each position in a decimal number has a weight that is a power of 10.

For example, the decimal number 572 can be expanded according to place value as follows:

$$572_{10} = (5 \times 10^2) + (7 \times 10^1) + (2 \times 10^0)$$

This expansion shows that the value of a decimal number depends on both the digit itself and its position in the number.

Binary Numbers

The binary number system is the fundamental number system used in digital electronics and computing. It is called a base-2 system because it uses only two digits: 0 and 1. Each binary position has a weight that is a power of 2.

For example, the binary number 101101_2 can be written as:

$$101101_2 = (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 45_{10}$$

Binary numbers are extremely important because digital hardware internally represents data using only two stable states. Therefore, the binary number system matches naturally with logic 0 and logic 1.

Decimal to Binary Conversion

To convert a positive decimal integer into binary, the standard method is repeated division by 2. At each step, we record the remainder. The binary result is obtained by reading the remainders from bottom to top.

General Steps

1. Divide the decimal number by 2.
2. Record the remainder (0 or 1).
3. Divide the quotient again by 2.
4. Continue until the quotient becomes 0.
5. Read the remainders from the last one to the first one.

Example 1: Convert 25_{10} to Binary

Step	Dividend	Quotient	Remainder
1	25	12	1
2	12	6	0
3	6	3	0
4	3	1	1
5	1	0	1

Therefore, $25_{10} = 11001_2$

Example 2: Convert 43_{10} to Binary

Step	Dividend	Quotient	Remainder
1	43	21	1
2	21	10	1
3	10	5	0
4	5	2	1
5	2	1	0
6	1	0	1

Therefore, $43_{10} = 101011_2$

Homework

1. Define analog quantity and digital quantity, and write two examples for each one.
2. Explain the difference between a bit, a logic level, and a digital waveform.
3. Write the truth table for NOT, AND, OR, and XOR operations.
4. Convert the following decimal numbers to binary: 13, 19, 32, and 57.
5. Convert the following binary numbers to decimal: 1010_2, 1111_2, and 100101_2.