COURSE CODE: IFT 211

COURSE TITLE: Digital Logic Design

COURSE UNITS: 3 Units

MODULE 1

Lesson 1: Introduction to Digital Logic & Number Systems

Introduction:

This lesson provides an introduction to digital logic and number systems, the fundamental building blocks of digital technology. We will cover the basic principles of binary representation and logic gates, providing you with the essential knowledge to understand how digital devices operate.

Learning Outcomes:

By the end of this lesson, students will be able to:

1. **Explain** how data is represented using binary numbers in computers.
2. **Describe** the four main number systems used in digital logic (decimal, binary, octal, hexadecimal).
3. **Perform** conversions between decimal, binary, octal, and hexadecimal number systems.
4. **Illustrate** how ASCII represents characters in binary form.

Data Representation in Computers

Data representation is the process of converting real-world data into a format that computers can understand. This format allows computers to store data, perform calculations, and communicate.

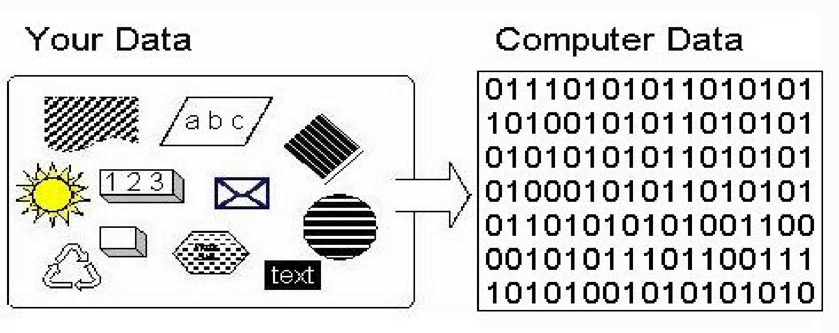


Figure 1: Representing Real-World Data in the Computer

Computers store and process all kinds of data using binary numbers (0s and 1s). Since computers are digital devices, they use transistors, which operate in two states:

- ON (1)
- OFF (0)

All types of information (text, numbers, images, audio, and video) are represented in binary form in a computer. Some common data representations include:

Data Type	**Representation**	**Explanation**
Text (characters)	ASCII, Unicode	Standard encoding systems that assign numerical values to characters.
Numbers	Binary, Octal, Hexadecimal	Different number systems used to represent numerical data in digital form.
Images	Bitmap (Pixels)	Images are represented as a grid of pixels, each with a binary value for color.
Audio	Digital Sound Wave	Sound is sampled and converted into binary data representing the wave's amplitude.
Video	Frames of Images	Video is a sequence of still images (frames), each represented digitally.

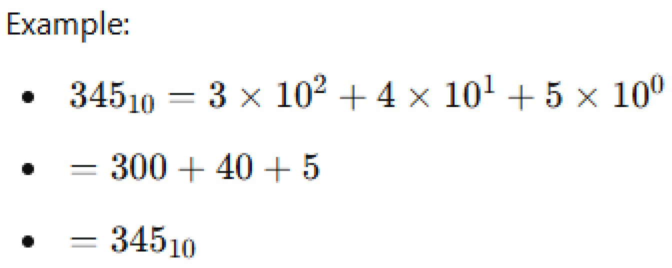
Example: ASCII Representation

- The letter 'A' is represented as 01000001 in 8-bit ASCII.
- The number 65 (decimal) corresponds to A in ASCII.

Number Systems

A number system defines how numbers are represented using different symbols or digits. The most commonly used number systems in digital logic are:

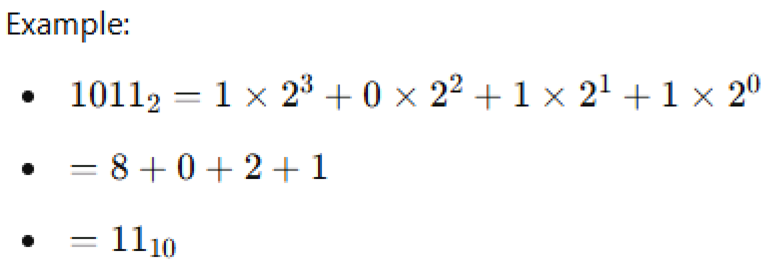
1. **Decimal Number System (Base-10)**
   - Uses digits 0-9
   - Each digit represents powers of 10



Example:

- $345_{10} = 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$
- $= 300 + 40 + 5$
- $= 345_{10}$

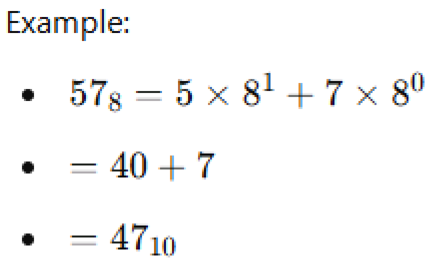
2. **Binary Number System (Base-2)**
   - Uses only two digits: 0 and 1
   - Each digit represents powers of 2



Example:

- $1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
- $= 8 + 0 + 2 + 1$
- $= 11_{10}$

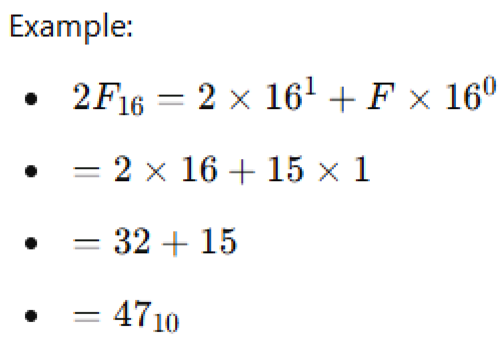
3. **Octal Number System (Base-8)**
   - Uses digits 0-7
   - Each digit represents powers of 8



Example:

- $57_8 = 5 \times 8^1 + 7 \times 8^0$
- $= 40 + 7$
- $= 47_{10}$

4. **Hexadecimal Number System (Base-16)**
    - Uses digits 0-9 and letters A-F
    - Each digit represents powers of 16



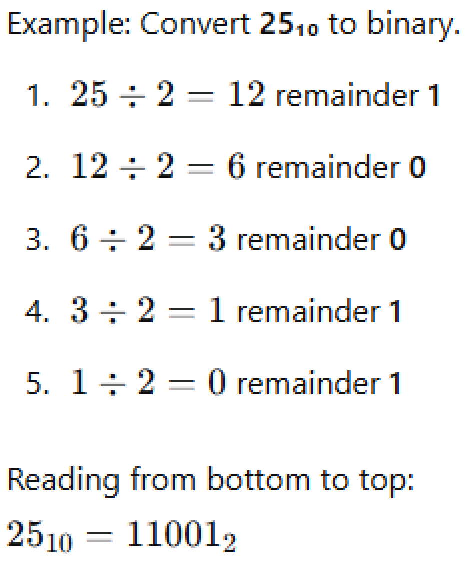
Example:

- $2F_{16} = 2 \times 16^1 + F \times 16^0$
- $= 2 \times 16 + 15 \times 1$
- $= 32 + 15$
- $= 47_{10}$

Conversion between Number Systems

Converting Decimal to Binary

Use successive division by 2 method.



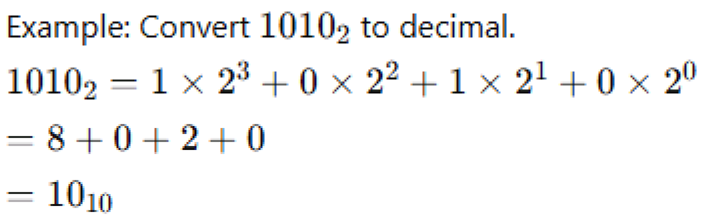
Example: Convert **25_{10}** to binary.

1. $25 \div 2 = 12$ remainder **1**
2. $12 \div 2 = 6$ remainder **0**
3. $6 \div 2 = 3$ remainder **0**
4. $3 \div 2 = 1$ remainder **1**
5. $1 \div 2 = 0$ remainder **1**

Reading from bottom to top:
$25_{10} = 11001_2$

Converting Binary to Decimal

Multiply each bit by increasing powers of 2.

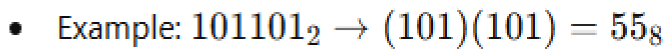


Example: Convert 1010_2 to decimal.

$$1010_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$
$$= 8 + 0 + 2 + 0$$
$$= 10_{10}$$

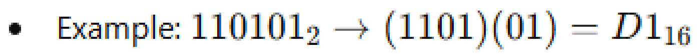
Converting Binary to Octal & Hexadecimal

Binary to Octal: Group into three bits from right.



- Example: $101101_2 \rightarrow (101)(101) = 55_8$

Binary to Hexadecimal: Group into four bits from right.



- Example: $110101_2 \rightarrow (1101)(01) = D1_{16}$

Summary

Computers represent data using binary numbers (0s and 1s) since they operate digitally with transistors switching between ON (1) and OFF (0) states. Different types of data, such as text and images, are stored in binary form, with common representations like ASCII encoding characters into binary. Number systems define how numbers are represented, with the most common ones being decimal (Base-10), binary (Base-2), octal (Base-8), and hexadecimal (Base-16). Converting between these number systems involves specific methods, such as successive division for decimal-to-binary conversion and grouping bits for binary-to-octal or binary-to-hexadecimal conversion. Understanding these concepts is essential for working with digital logic and computer systems.

Evaluation Questions:

1. Convert the decimal number 25 to binary.
2. Convert the binary number 110110 to decimal.

3. Convert the binary number 10111010 to hexadecimal.
4. Explain how the transistor on/off states relate to binary 1 and 0.
5. What are some real world applications of binary, and hexadecimal number systems?

Suggested Answers:

1. **Convert the decimal number 25 to binary.**
    - Answer: 11001
2. **Convert the binary number 110110 to decimal.**
    - Answer: 54
3. **Convert the binary number 10111010 to hexadecimal.**
    - Answer: BA
4. **Explain how the transistor on/off states relate to binary 1 and 0.**
    - Answer: In a computer, transistors act as switches. An "ON" state corresponds to a binary 1, and an "OFF" state corresponds to a binary 0. This allows computers to represent and process information using binary code.
5. **What are some real world applications of binary, and hexadecimal number systems?**
    - Answer:
        - Binary: Used in all digital devices, including computers, smartphones, and digital circuits, for data storage and processing.
        - Hexadecimal: Used for representing memory addresses, color codes in web design, and for debugging in programming. It's used because it is a more compact way to represent binary data.