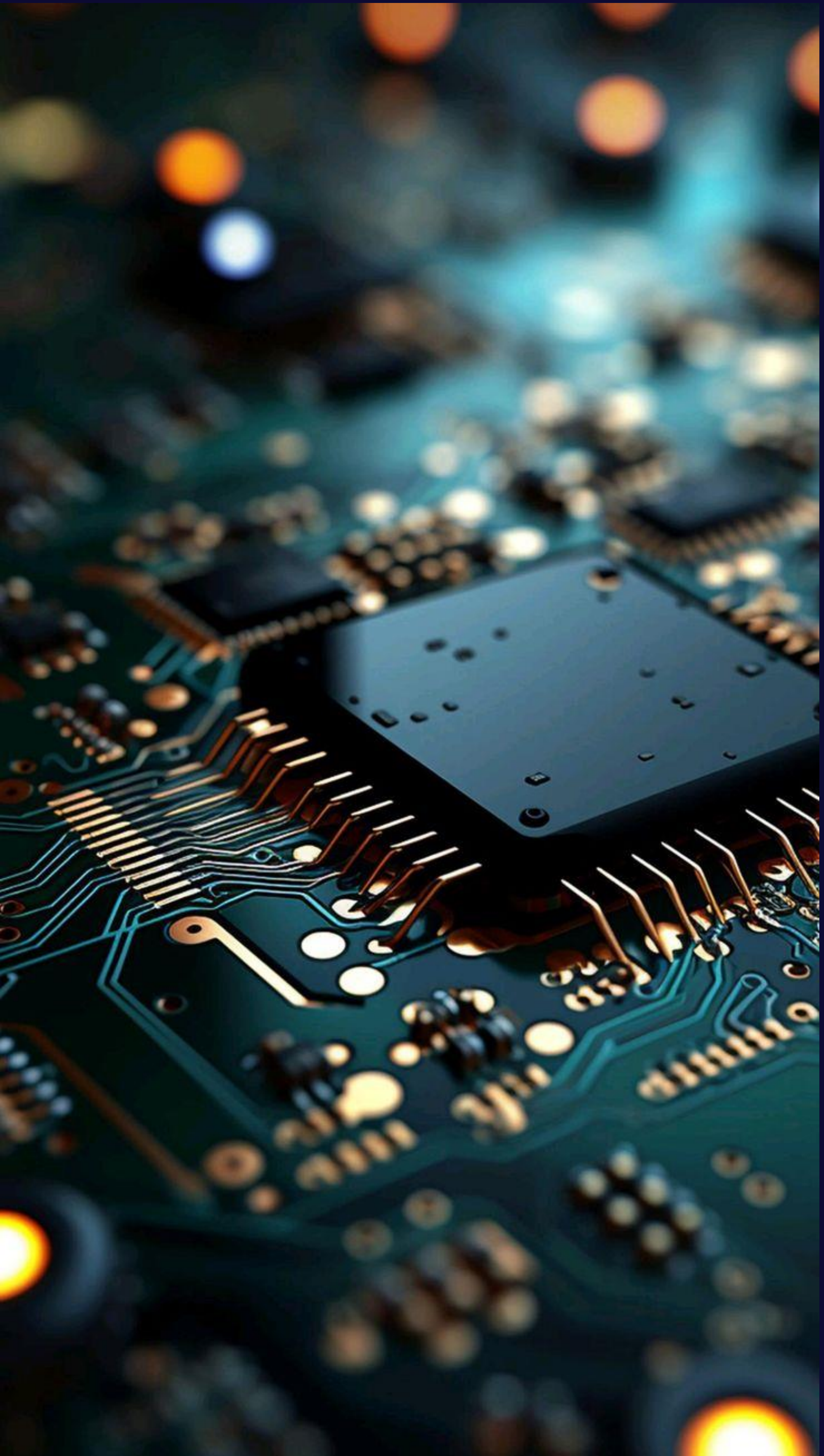


Computer Fundamentals

Course Code: CSE-0611-1101

Md. Zahid Akon
Lecturer
Department of CSE

CLO'S

01 Understand computer systems, components, organization, and data processing.

02 Use operating systems, word processors, spreadsheets, and presentation tools.

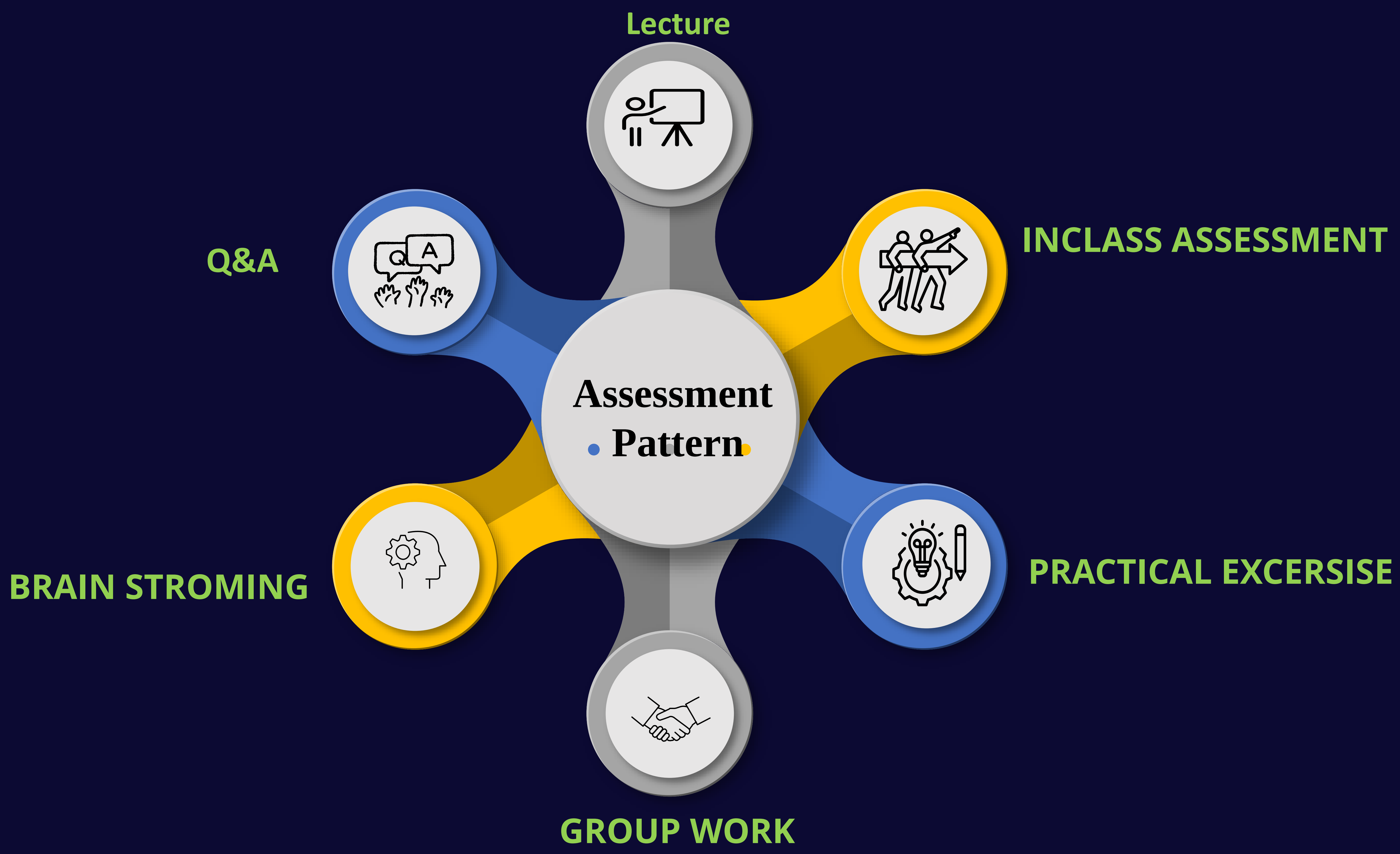
03 Operate computer hardware, peripherals, storage, and networking devices.

04 Evaluate ethical and social issues like privacy, security, and accessibility.

05 Solve computer issues, perform maintenance, and explain tech to non-experts.

Recommended Books

1. "Computer Fundamentals" by P.K. Sinha
2. "Computer Organization and Design" by David A. Patterson and John L. Hennessy
3. "Operating System Concepts" by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin
4. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
5. "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall



Lecture
Q&A
INCLASS ASSESSMENT
Assessment Pattern
BRAIN STROMING
PRACTICAL EXCERSISE
GROUP WORK

Course Plan

Week	Topics	Teaching Learning Strategy(s)	Assessment Strategy(s)	Alignment to CLO
1	Introduction to Computers and Their Components	Lecture, multimedia, discussions	Feedback, Q&A, assessment	CLO1 & CLO2
2	Computer Memory	Lecture, discussions	Q&A, assignments	CLO1
3	Input Devices	Lecture, multimedia	Quiz & assessments	CLO3
4	Output Devices	Lecture, multimedia	Q&A assessments	CLO3
5	Central Processing Unit (CPU)	Lecture, Practical implementation	Feedback, Q&A, assessment	CLO1 & CLO3
6 & 7	Number System	Lecture, Practical implementation	Q&A, assignments	CLO1
7 & 8	Boolean Algebra	Lectures, discussions	Midterm assessments	CLO1 & CLO2
9 & 10	Mid Examination	Mid-term exams	Mid-term assessment	CLO1 – CLO3

11	Logic Gates	Lecture, Practical implementation	Q&A, assignments	CLO1 & CLO3
12	Introduction to Basic Networking Concepts	Lecture, multimedia, discussions	Ethical analysis, Midterm assessments	CLO3, CLO5
13	Operating Systems: Functions, Booting, and Types	Interactive lectures, examples from real-world applications	Final term assessments	CLO5
14	Computer Security: Viruses, Infection Mechanisms, and Antivirus Protection	Lecture, multimedia, discussions	Final term assessments	CLO5
15 & 16	Core Programming Concepts: Algorithms, Flowcharts, and Pseudocode	Interactive lectures, examples from real-world applications	Feedback, Q&A, assessment	CLO4
17	Final Topics Review and Discussion: Integration of Concepts	Revision through Q&A, group activities	Participation, group evaluation	CLO1 - CLO5

Week 1
Introduction to Computers and Their Components

Introduction to Computers

This presentation will provide a comprehensive overview of computers, exploring their components, functions, and impact on modern society.

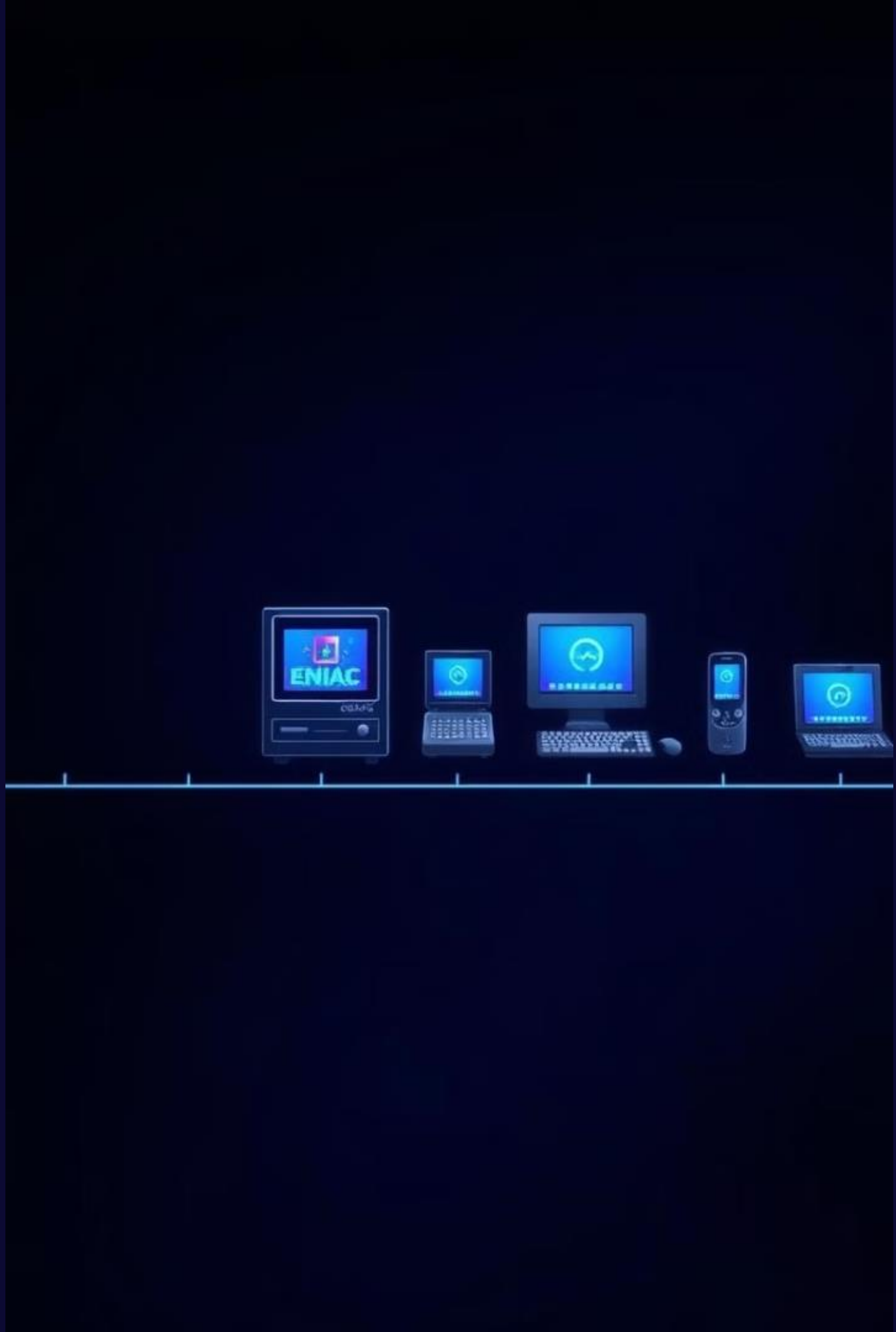
What is a Computer?

A computer is an electronic device that can accept data, process it according to a set of instructions, and produce information.

It's a versatile tool for tasks ranging from basic calculations to complex scientific simulations and creative endeavors.

History and Evolution of Computers

1. The origins of computers can be traced back to mechanical calculators and early electronic devices.
2. Over the decades, computers have undergone rapid advancements in size, processing power, and functionality.
3. From room-sized mainframes to pocket-sized smartphones, computers have become an integral part of our lives.



Hardware Components of a Computer

Input Devices

Devices that allow users to enter data and instructions into the computer.

Output Devices

Devices that display or produce the results of computer processing.

Storage Devices

Devices that store data and instructions for long-term use.

Central Processing Unit (CPU)

The "brain" of the computer, responsible for executing instructions and processing data.

Memory (RAM)

Temporary storage for data and programs that are currently being used by the computer.

Diagram of a Computer System

Input Devices

Keyboard

Used to type text and enter commands.

Mouse

Used to control the cursor and interact with graphical elements.

Touchscreen

Allows direct interaction with the computer by touching the screen.

Microphone

Used to record audio and input voice commands.

Output Devices

Monitor

Displays visual information, such as text, images, and videos.

Printer

Produces hard copies of documents and images.

Speakers

Play audio, such as music, sound effects, and voice recordings.

Projector

Displays images and videos onto a larger screen.

Storage Devices

Hard Drive (HDD)

A traditional storage device that uses spinning platters to store data.

Solid-State Drive (SSD) (SSD)

A faster and more durable storage device that uses flash memory.

USB Drive

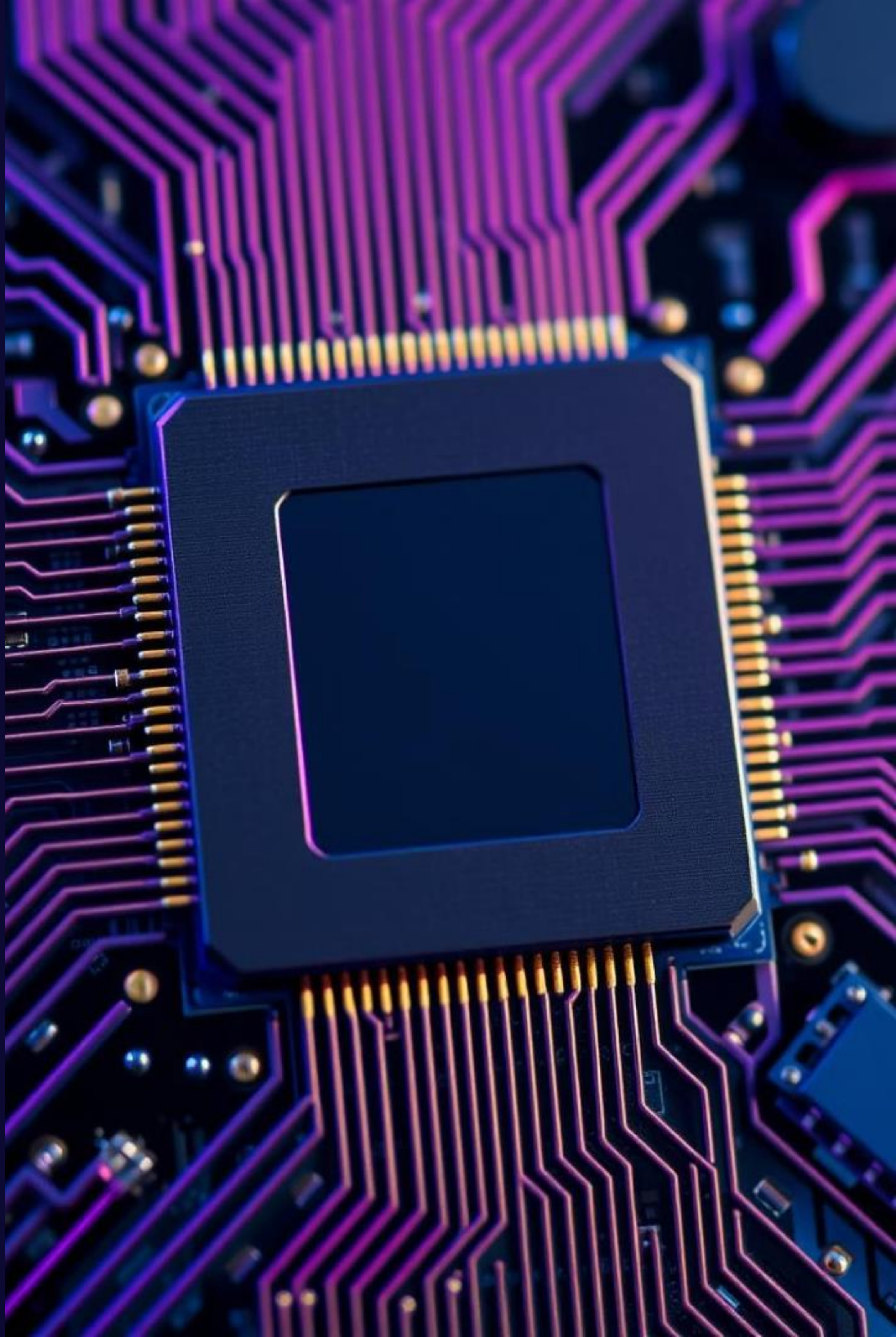
A portable storage device that can be connected to a computer via a USB port.

Cloud Storage

Data stored on remote servers and accessed over the internet.

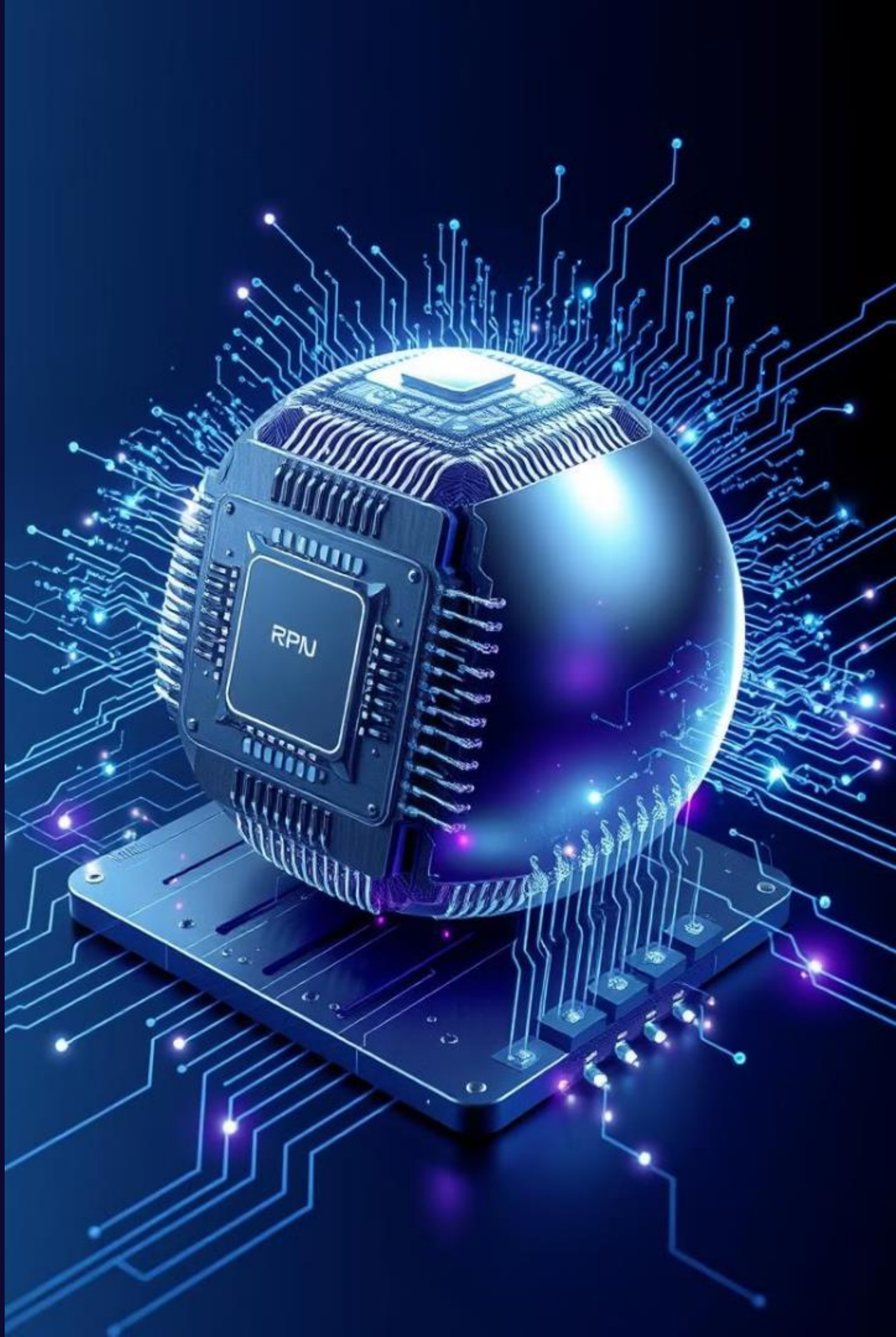
Central Processing Unit (CPU)

The CPU is the brain of the computer, responsible for executing instructions, performing calculations, and controlling the flow of data.



Memory and RAM

RAM is the computer's short-term memory, holding data and instructions that the CPU needs to access quickly for ongoing operations.



Week 2
Computer Memory

Computer Memory: An Overview

This presentation explores the essential components of computer memory, including types, functions, and interactions. We'll delve into the hierarchy of memory, how it's managed, and its impact on system performance.

Types of Computer Memory

RAM

Random access memory (RAM) is temporary storage used by the CPU to hold data and instructions. RAM is volatile, meaning its contents are lost when the power is off.

ROM

Read-only memory (ROM) holds permanent data and instructions that are essential for the computer's boot process. ROM is non-volatile, meaning its contents are retained when the power is off.

Cache Memory

Cache memory is a high-speed temporary storage that stores frequently accessed data and instructions. It speeds up data access by providing faster access to frequently used information.

Secondary Storage Storage

Secondary storage includes devices like magnetic disks (HDDs) and solid state drives (SSDs), which store large amounts of data. Secondary storage is non-volatile and persistent, allowing for long-term data storage.

Random Access Memory (RAM)

1 Fast Access

2 Volatile

3 Temporary Storage

Read-Only Memory (ROM)

1. Non-Volatile
2. Permanent Storage
3. Essential for Boot-up

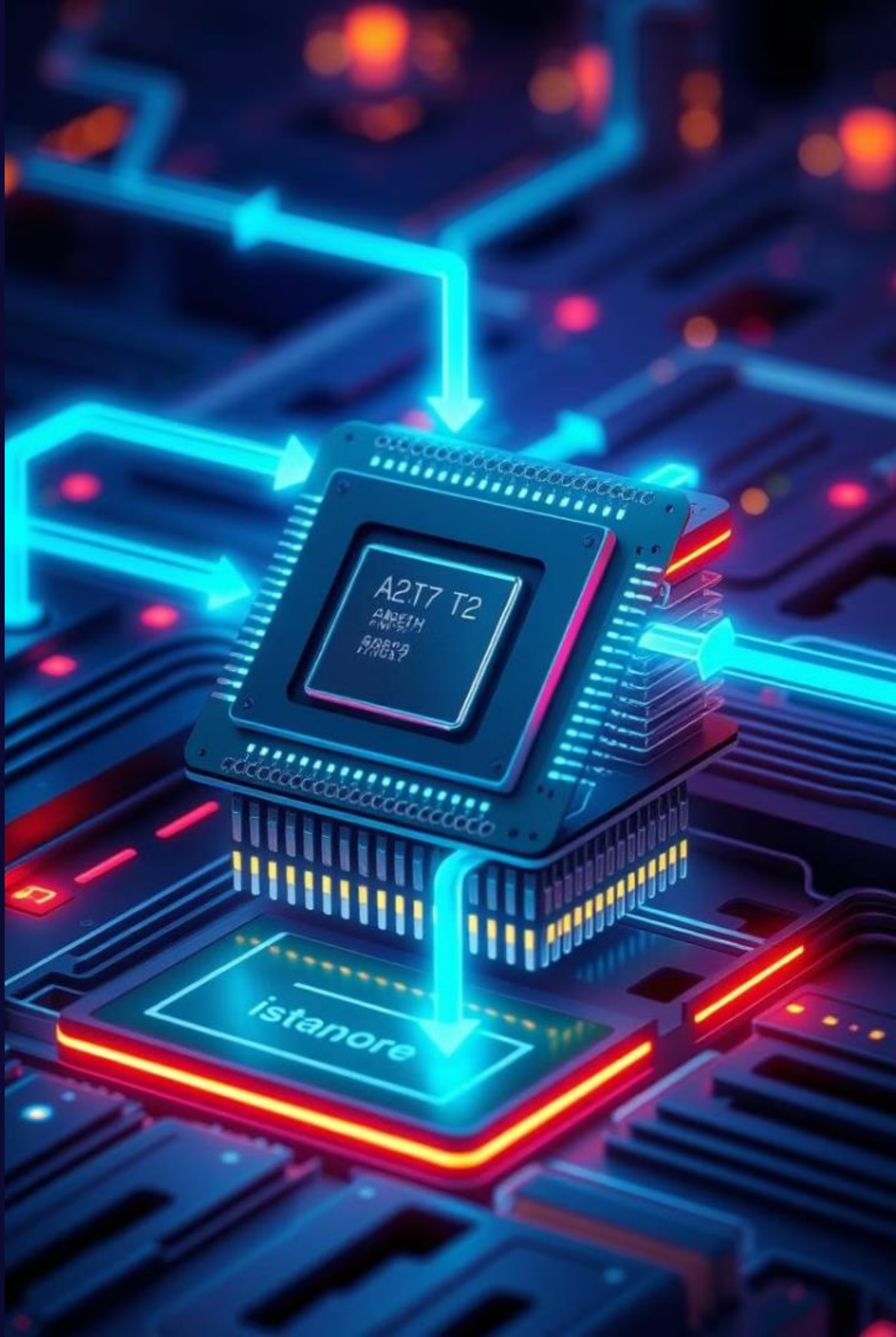
Cache Memory

Speeds up Data Access

Caches store frequently used data, reducing the need to access slower memory.

Levels of Caching

Multiple levels of caches (L1, L2, L3) provide progressively larger and slower storage for data access.



Magnetic Disk Memory

1

Hard Disk Drives (HDDs)

Mechanical storage devices with rotating platters and read/write heads.

2

Non-Volatile

HDDs retain data even when the power is off.

3

Large Capacity

HDDs offer high storage capacities at a lower cost per gigabyte.

Solid State Drives (SSDs)

10x

Faster

More Durable

Durable

Lower Power

Power Efficient

Memory Hierarchy

1. Registers
2. Cache
3. RAM
4. Secondary Storage

Memory Management Techniques

Paging

Dividing memory into equal-sized pages for efficient allocation and management.

Segmentation

Dividing memory into variable-sized segments based on program requirements.

Virtual Memory

Using secondary storage as an extension of RAM to increase available memory.

Conclusion and Key Takeaways

Understanding computer memory is crucial for optimizing system performance. The different types of memory work together in a hierarchy to provide efficient data storage and retrieval. Effective memory management techniques are essential for efficient operation.

Week 3
Input Devices

Input Devices: A Look at How How We Interact With Technology

Keyboard: The Typing Workhorse

Keyboards are essential for data entry, coding, writing, and much more. They come in various layouts and styles, offering comfort and customization. Mechanical keyboards provide tactile feedback and are popular among gamers and writers, while ergonomic keyboards promote comfortable posture.

Mouse: Precise Navigation and Control

1 **Precision and Speed**

Mice have become essential for navigating computer interfaces, selecting files, and manipulating objects. They offer precise control over the cursor.

2 **Ergonomic Comfort**

Many mice are designed with ergonomic considerations, ensuring comfort during extended use.

3 **Variety of Styles**

Mice come in various shapes, sizes, and features, catering to different hand sizes and preferences.

Touchpad: Convenient Navigation on Laptops

Integrated Convenience Convenience

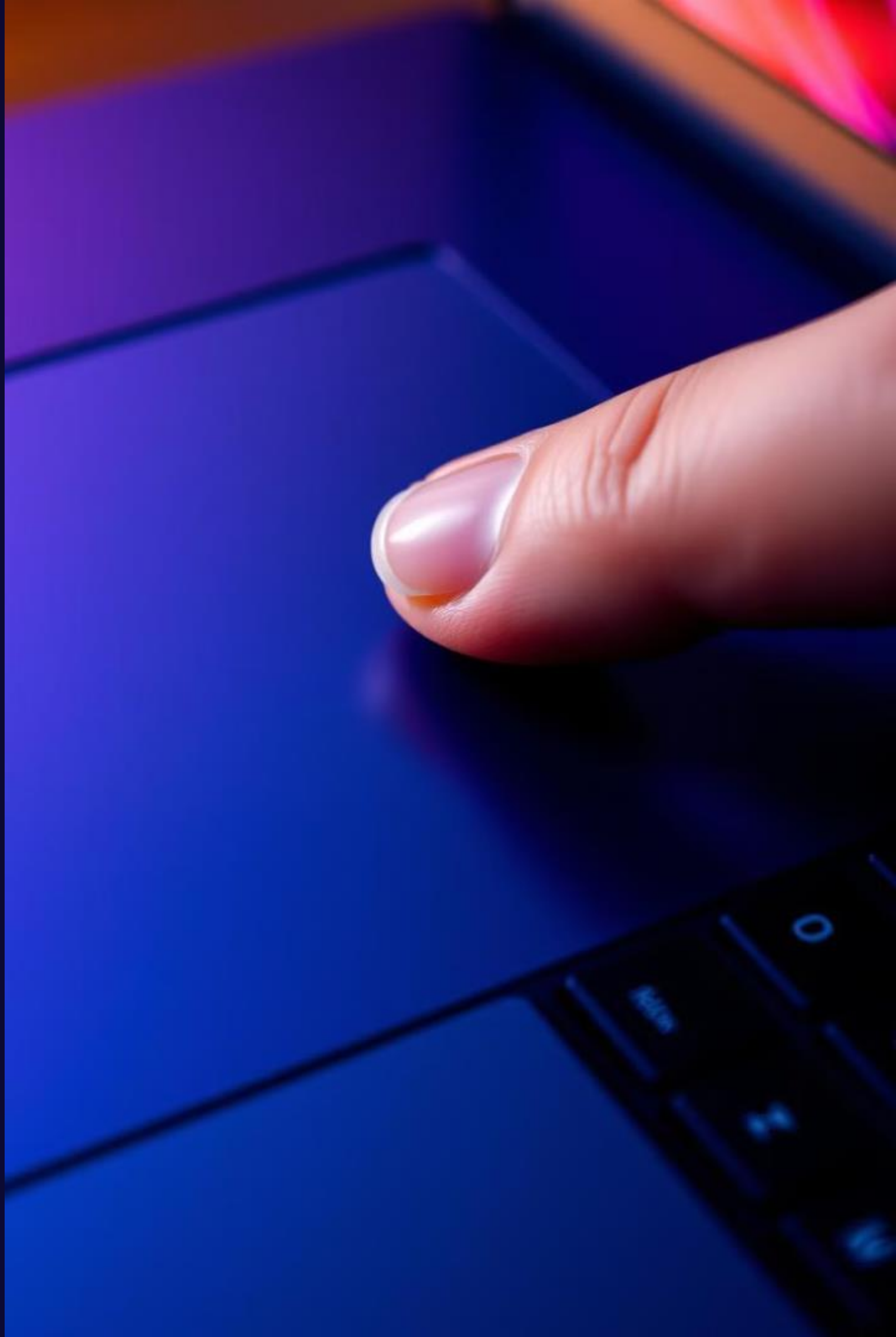
Touchpads are built into most laptops, eliminating the need for an external mouse.

Multi-touch Gestures

Modern touchpads support multi-touch gestures, enabling scrolling, zooming, and other actions with a finger.

Precision Control

Touchpads offer precise control for navigation and cursor movement.



Touchscreen: Interactive and Intuitive

Mobile Devices

Touchscreens have revolutionized mobile devices, allowing users to interact directly with their phones and tablets.

Desktop Computers

Touchscreens are also increasingly found on desktop computers, enhancing interactivity and providing a more intuitive user experience.

Tablets

Tablets are primarily controlled through their touchscreens, offering a seamless and intuitive experience for browsing, gaming, and creativity.

Scanners

Types

Flatbed, sheet-fed, handheld.

Functions

Convert physical documents and images into digital formats.

Joystick: Precise Control for Gaming

Early Gaming

Joysticks were an early form of input for controlling video game characters, offering precise movement and aiming.

1

2

Modern Gaming

Modern joysticks are still used for flight simulators and some racing games, providing immersive control.

Gamepad: Immersive Gaming Gaming Experience

1

Console Gaming

Gamepads are the primary input device for console gaming, offering buttons, joysticks, and triggers for precise control.

2

PC Gaming

Gamepads are increasingly used for PC gaming, providing a more comfortable and intuitive experience for certain games.

Trackball: Ergonomic and Efficient

1 Cursor Control

Trackballs work by rolling a ball with your thumb or fingers, which translates to cursor movement on the screen.

2 Ergonomic Design

Trackballs are known for their ergonomic design, minimizing wrist strain and promoting comfort.

3 Precision and Control

Trackballs offer precise cursor control, making them ideal for tasks that require fine movements.

Digitizer: Precise Input for Artists and Designers

1

Pen Input

Digitizers allow users to draw or write directly on a tablet surface with a stylus, replicating the feel of pen and paper.

2

Pressure Sensitivity

High-quality digitizers feature pressure sensitivity, allowing for varying line widths and brush strokes.

3

Artistic Precision

Digitizers are commonly used by artists, designers, and illustrators for precise and expressive digital creation.

Conclusion: A World of Input Options

1

Variety

The world of input devices offers a diverse range of options to meet different needs.

2

Evolution

Input technology continues to evolve, with new devices and features emerging.

3

Choice

Users have the freedom to choose the input devices that best suit their workflow and preferences.

Week 4
Output Devices

Output Devices: Bringing Information to Life

Output devices are essential tools that allow us to interact with and experience information in various forms.

Printers

Types

Inkjet, laser, 3D.

Functions

Produce hard copies of digital documents and images.

Speakers

Types

Desktop, soundbar, surround sound.

Functions

Reproduce audio signals from computers, phones, and other devices.

Monitors

Types

LCD, LED, OLED.

Functions

Display visual output from computers, laptops, and other devices.

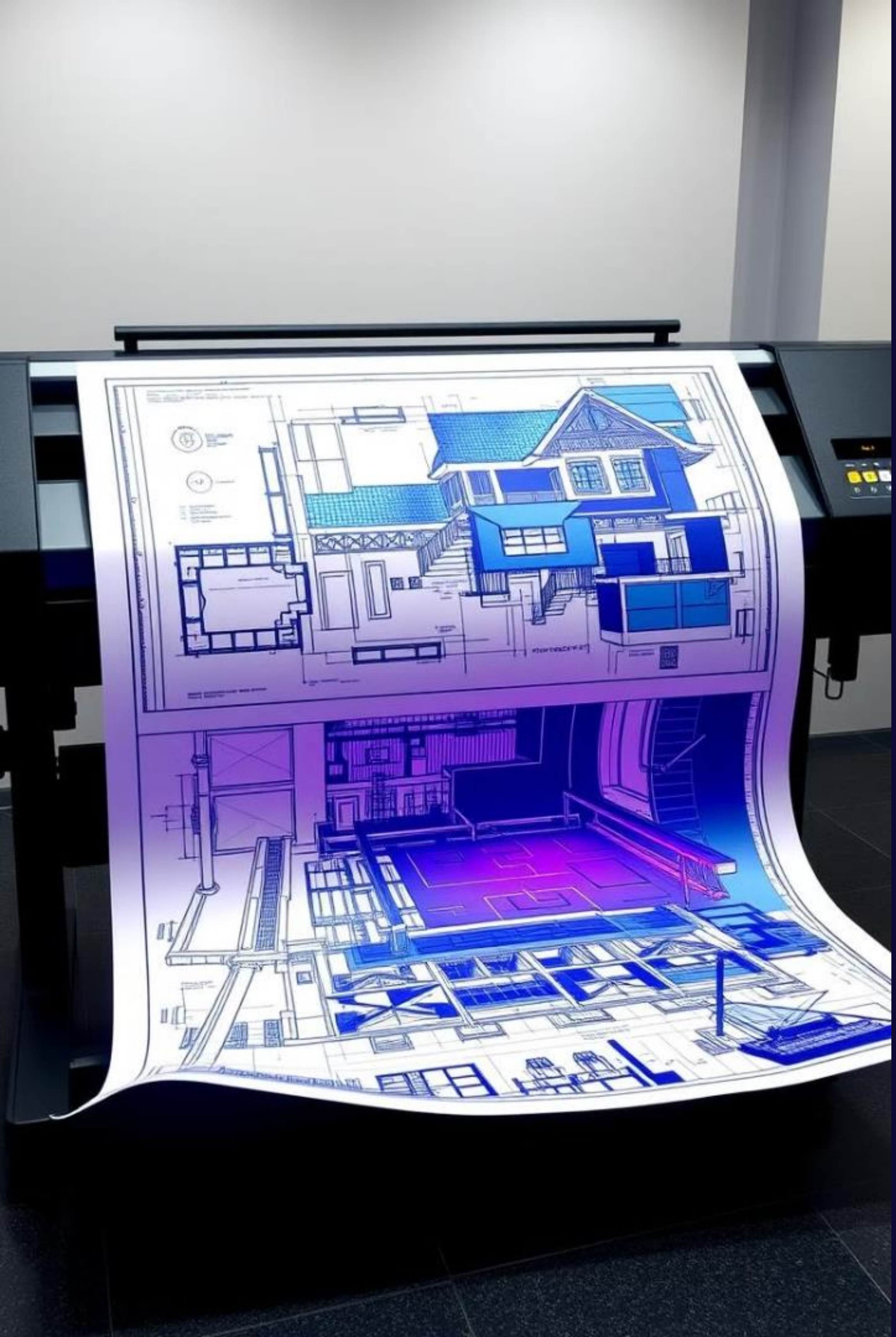
Projectors

Project images and videos onto larger surfaces.

1

2

Commonly used for presentations, home entertainment, and educational purposes.



Plotters

1

Types

Pen plotters, laser plotters, inkjet plotters.

2

Functions

Create high-quality, large-format drawings for engineering, architecture, and design applications.

Headphones

1

Types

Over-ear, on-ear, in-ear.

2

Functions

Provide private audio listening without disturbing others.

Touchscreens

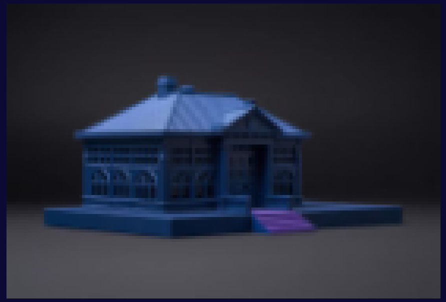
Types

Capacitive, resistive, infrared.

Functions

Allow direct interaction with digital content using touch gestures.

3D Printers



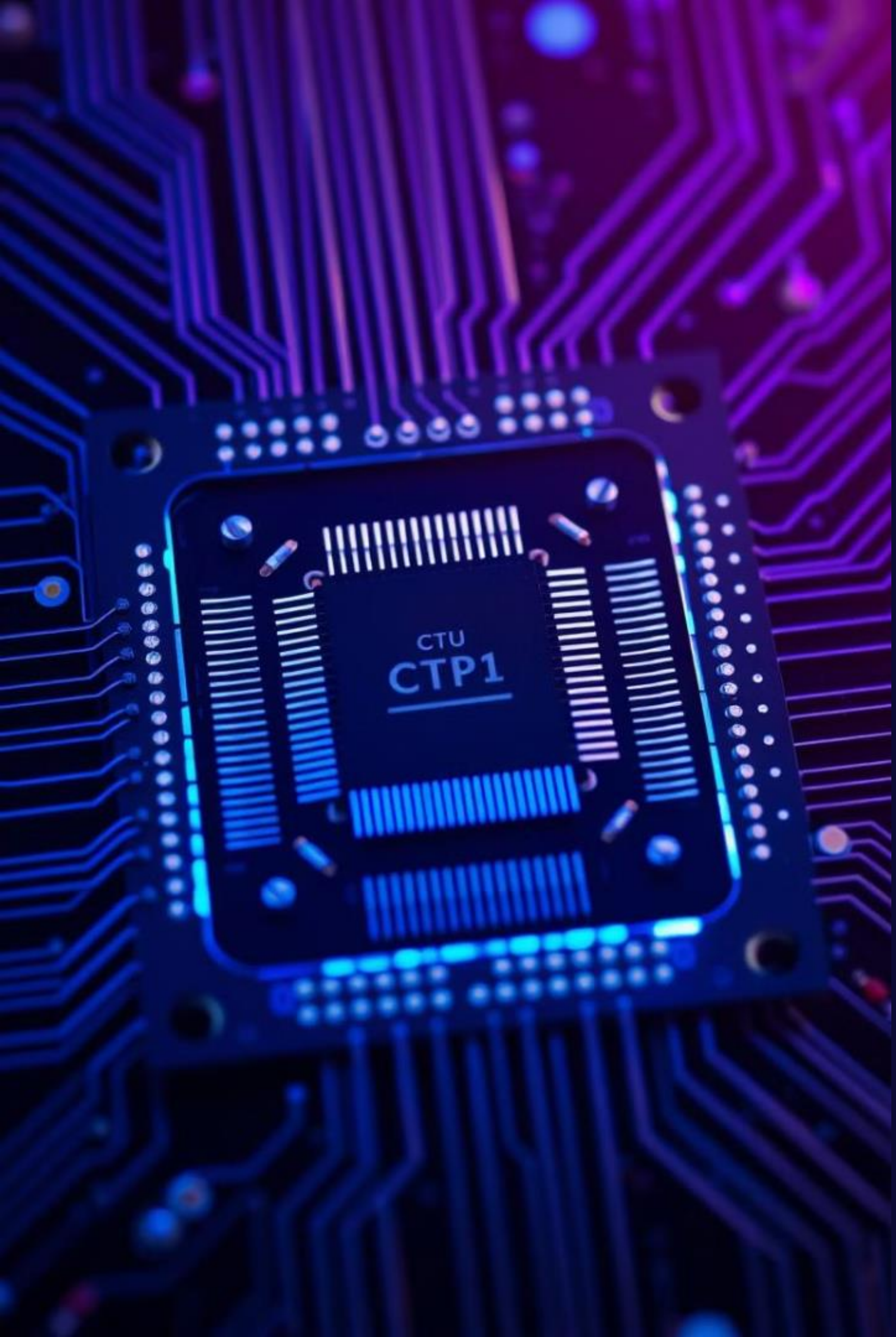
Types

FDM (Fused Deposition Modeling), SLA (Stereolithography), SLS (Selective Laser Sintering), DLP (Digital Light Processing), MJF (Multi Jet Fusion), and metal 3D printers.

Functions

Creates three-dimensional objects layer by layer.

Week 5
Central Processing Unit (CPU)



Introduction to the Central Processing Unit (CPU)

The central processing unit (CPU) is the brain of a computer. It is responsible for executing instructions and processing data. This presentation explores the fascinating world of CPUs, delving into their evolution, architecture, and key components.

Evolution of the CPU

Early Days

The first CPUs were bulky and slow. The development of integrated circuits revolutionized computing, paving the way for smaller, more powerful CPUs.

Microprocessors

The invention of the microprocessor in the 1970s led to the rise of personal computers and the exponential growth of computing power.

Multi-core Era

Today's CPUs feature multiple cores, enhancing performance and enabling parallel processing, revolutionizing tasks like gaming and scientific simulations.

CPU Architecture

1

Control Unit

Directs the flow of instructions and data within the CPU.

2

Arithmetic Logic Unit (ALU)

Performs arithmetic and logical operations on data.

3

Registers

Small, high-speed memory locations used to store temporary data and instructions.

4

Cache Memory

A small, fast memory that stores frequently accessed data for quicker retrieval.

Processor Cores

Single-Core

Traditional CPUs with a single core execute one instruction at a time.

Multi-Core

Modern CPUs with multiple cores can execute multiple instructions simultaneously, boosting performance.

Hyper-Threading

A technology that allows a single core to handle multiple threads, simulating multiple cores.

CPU Clock Speed and Performance

Clock Speed

Measured in gigahertz (GHz), clock speed indicates the number of instructions a CPU can execute per second.

Performance

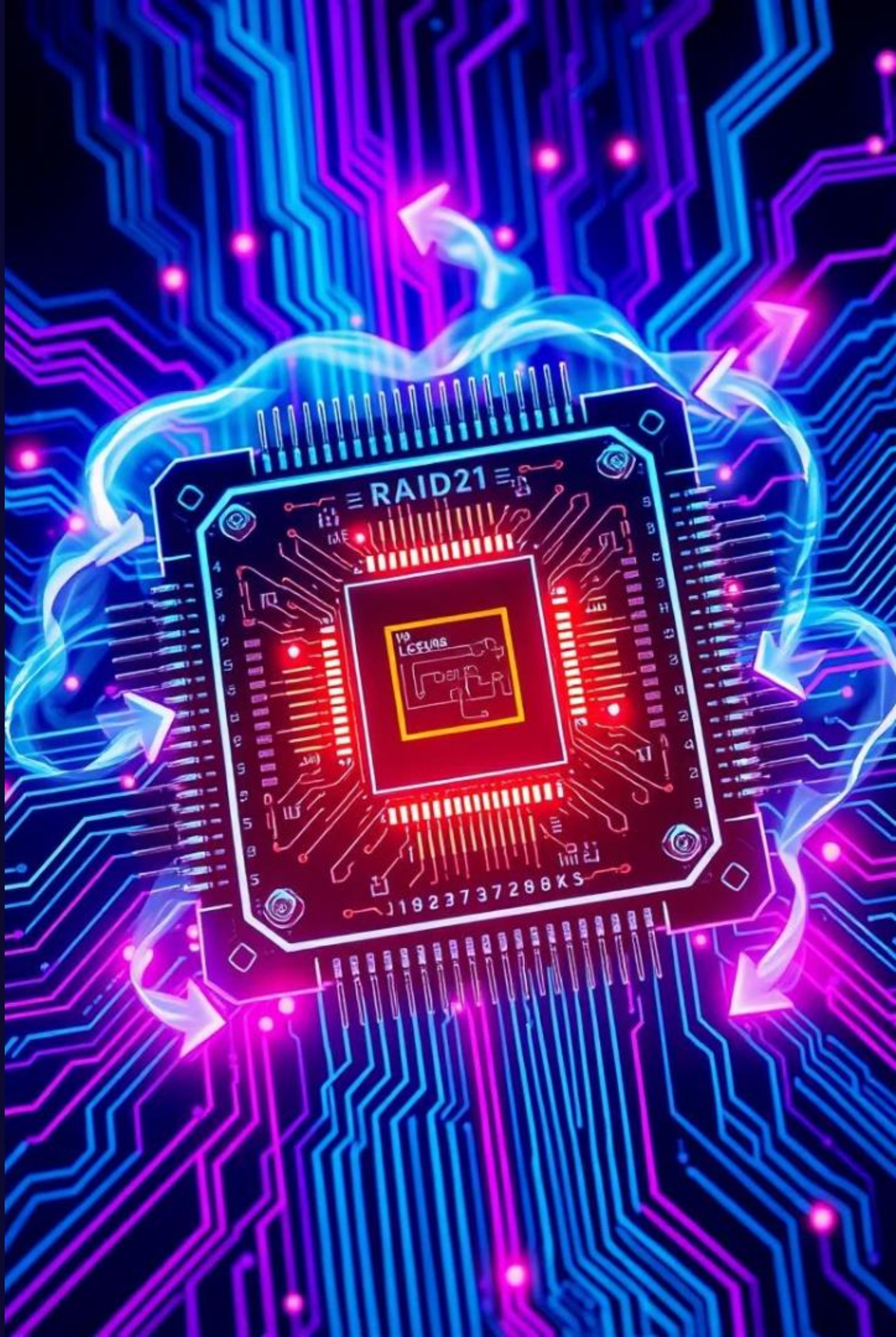
Higher clock speeds generally lead to faster performance, but other factors like cache size and core count also play a crucial role.

CPU Cache Memory

1 L1 Cache: Smallest and fastest cache, storing data most frequently used by the CPU.

2 L2 Cache: Larger and slightly slower than L1, holding less frequently used data.

3 L3 Cache: Largest and slowest, caching data from both L1 and L2, providing a wider range of data.



CPU Instruction Set Architectures Architectures

1

x86

Dominant architecture in PCs, used by Intel and AMD processors.

2

ARM

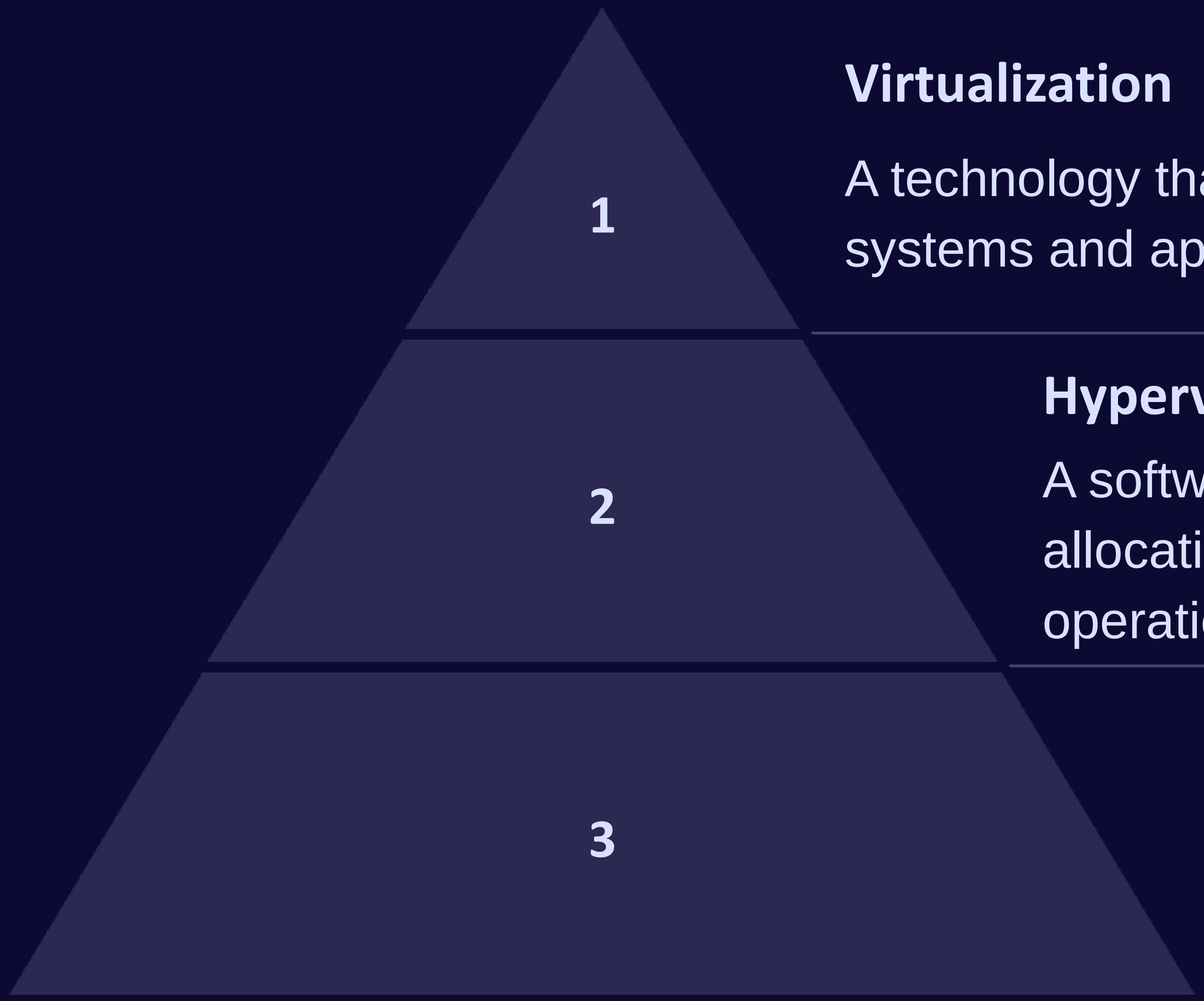
Popular in mobile devices and embedded systems, known for energy efficiency.

3

RISC-V

An open-source architecture gaining popularity for its flexibility and adaptability.

CPU Virtualization



Virtualization

A technology that allows a single CPU to run multiple operating systems and applications simultaneously.

Hypervisor

A software layer that manages the virtual machines, allocating resources and ensuring their smooth operation.

Virtual Machines

Isolated environments that emulate real hardware, allowing multiple applications to run independently.

CPU Power Management

1

Dynamic Frequency Scaling

Adjusts the CPU clock speed based on workload, reducing power consumption when idle.

2

Thermal Throttling

Reduces performance to prevent overheating, protecting the CPU from damage.

3

Power States

Allows the CPU to enter low-power states when not actively used, saving energy.

Future Trends in CPU Technology

1

AI Integration

CPUs are becoming increasingly optimized for AI tasks, enabling faster and more efficient processing of machine learning algorithms.

2

Quantum Computing

Emerging quantum computers could revolutionize computing, offering unprecedented processing power for complex problems.

3

Energy Efficiency

Further advancements in power management technologies will reduce energy consumption and improve the sustainability of computing.

Week 6 & 7
Number System

Number Systems

This presentation will explore the fundamental concepts of number systems, focusing on their types, conversions, and applications.

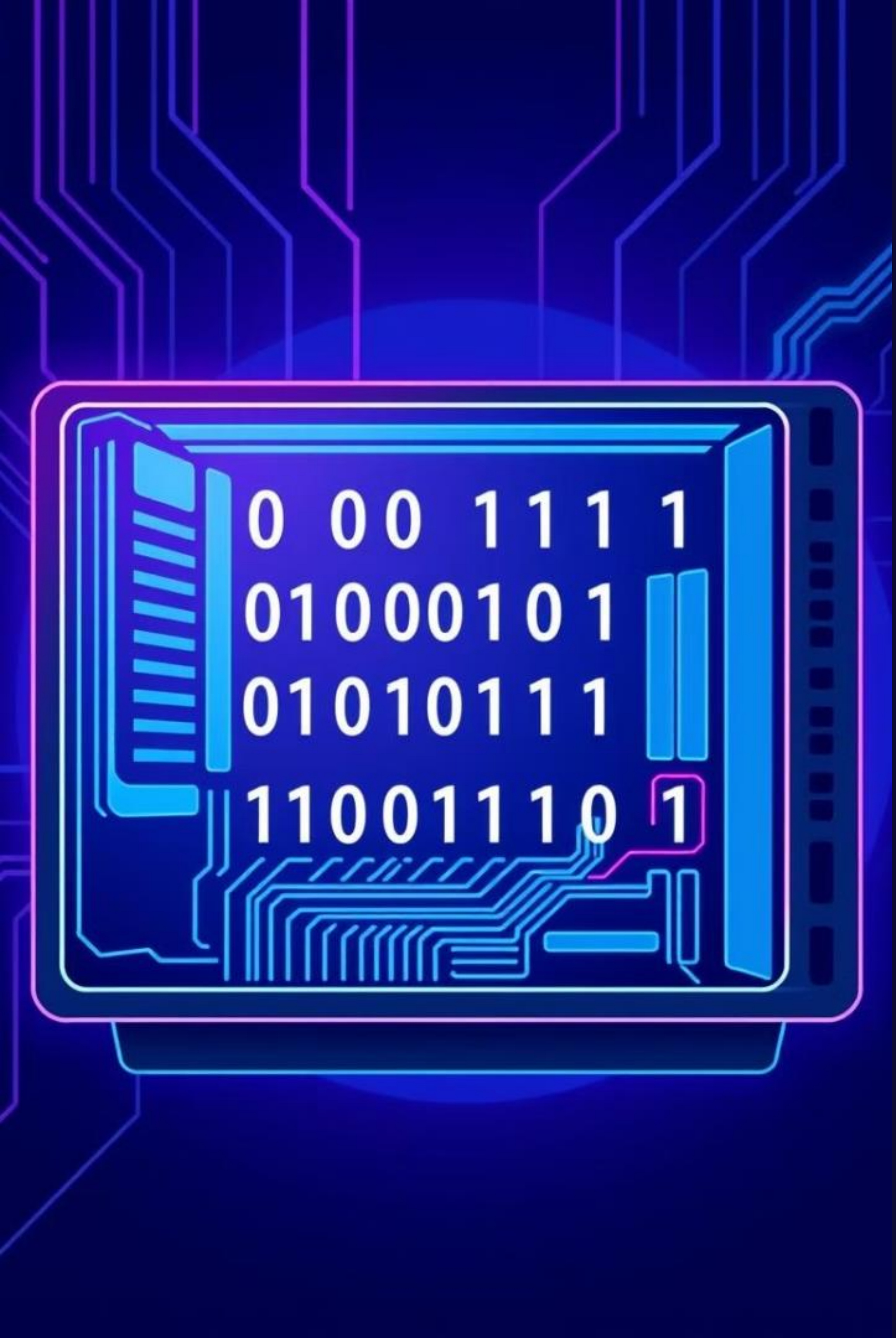
Decimal Number System

Base-10 System

The decimal number system uses ten unique digits, from 0 to 9. It is the most commonly used system in everyday life.

Positional Notation

Each digit's position in a decimal number determines its value. The rightmost digit represents the units place, while the next digit represents the tens place, and so on.



Binary Number System

1 **Base-2 System**

The binary number system only uses two digits, 0 and 1, which are called bits.

2 **Digital Computers**

Binary is the foundation of modern computers, as they can easily represent data and perform computations using only two states.

Octal Number System

Base-8 System

The octal system uses eight digits, from 0 to 7.

Concise Representation

Octal provides a more compact way to represent large binary numbers.

Hexadecimal Number System

Base-16 System

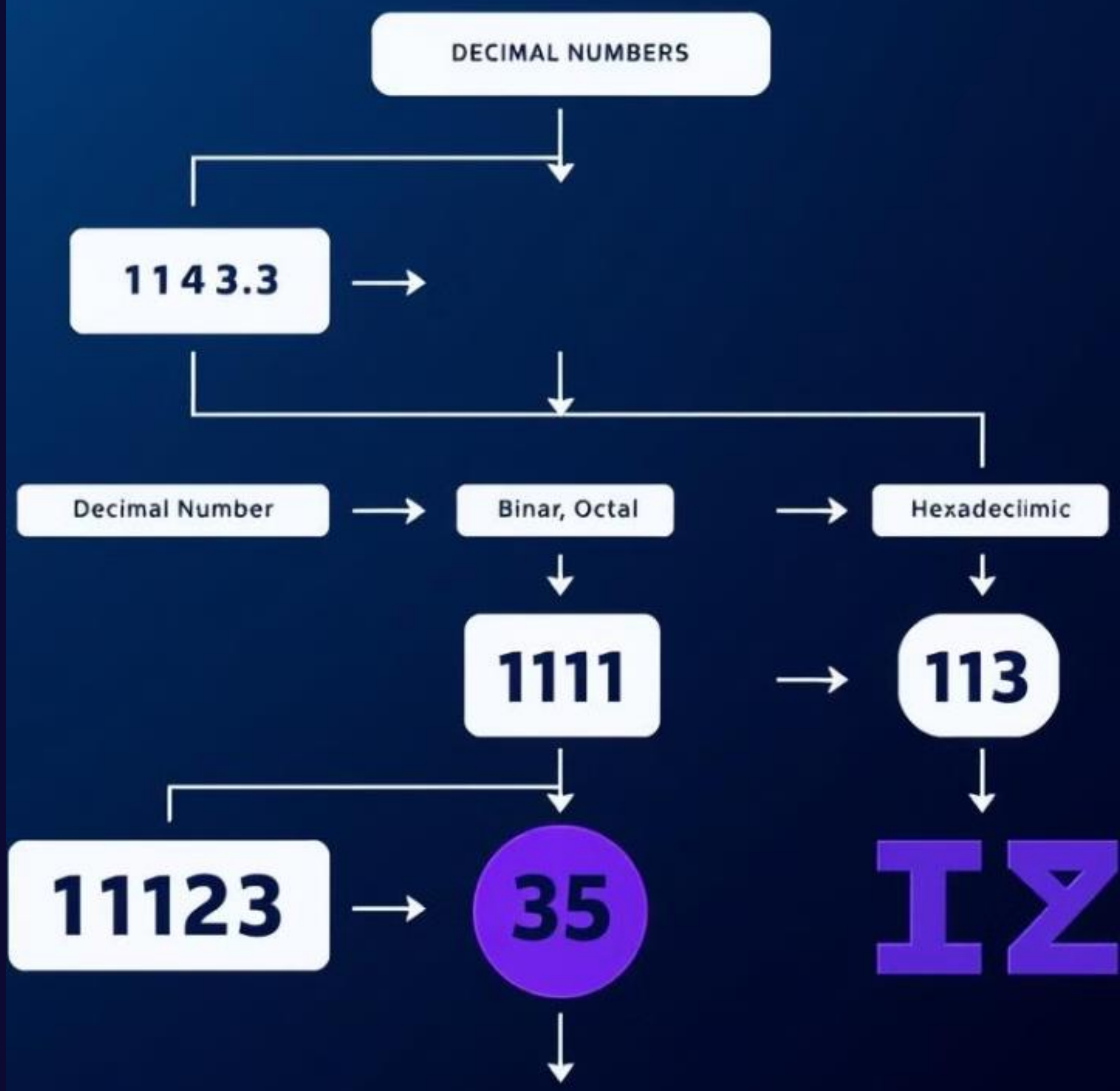
The hexadecimal system uses 16 digits, from 0 to 9 and A to F, representing values 10 to 15.

Color Codes

Hexadecimal is widely used in web development for defining colors.

Conversion between Number Number Systems

1. Decimal to Binary: Divide the decimal number by 2 repeatedly and collect the remainders in reverse order.
2. Binary to Decimal: Multiply each digit by its corresponding power of 2 and sum the results.
3. Conversion between other bases: Use decimal as an intermediary, converting the number to decimal and then to the desired base.



Addition and Subtraction in Binary Binary

1

Binary Addition

Follow the same principles as decimal addition, but with only two digits.

2

Carry-Over

When the sum of two bits exceeds 1, a carry-over is generated to the next position.

3

Binary Subtraction

Similar to decimal subtraction, with borrowing from the next position when necessary.

Multiplication and Division in Binary

1

Binary Multiplication

Similar to decimal multiplication, but with simpler multiplication rules.

2

Shifting and Adding

Multiplication involves shifting and adding based on the multiplicand's value.

3

Binary Division

Consists of repeatedly subtracting the divisor from the dividend, similar to decimal division.

Applications of Number Systems

1

Computers

Binary forms the foundation for data representation and processing in modern computers.

2

Networks

Binary is used in network protocols for data transmission and routing.

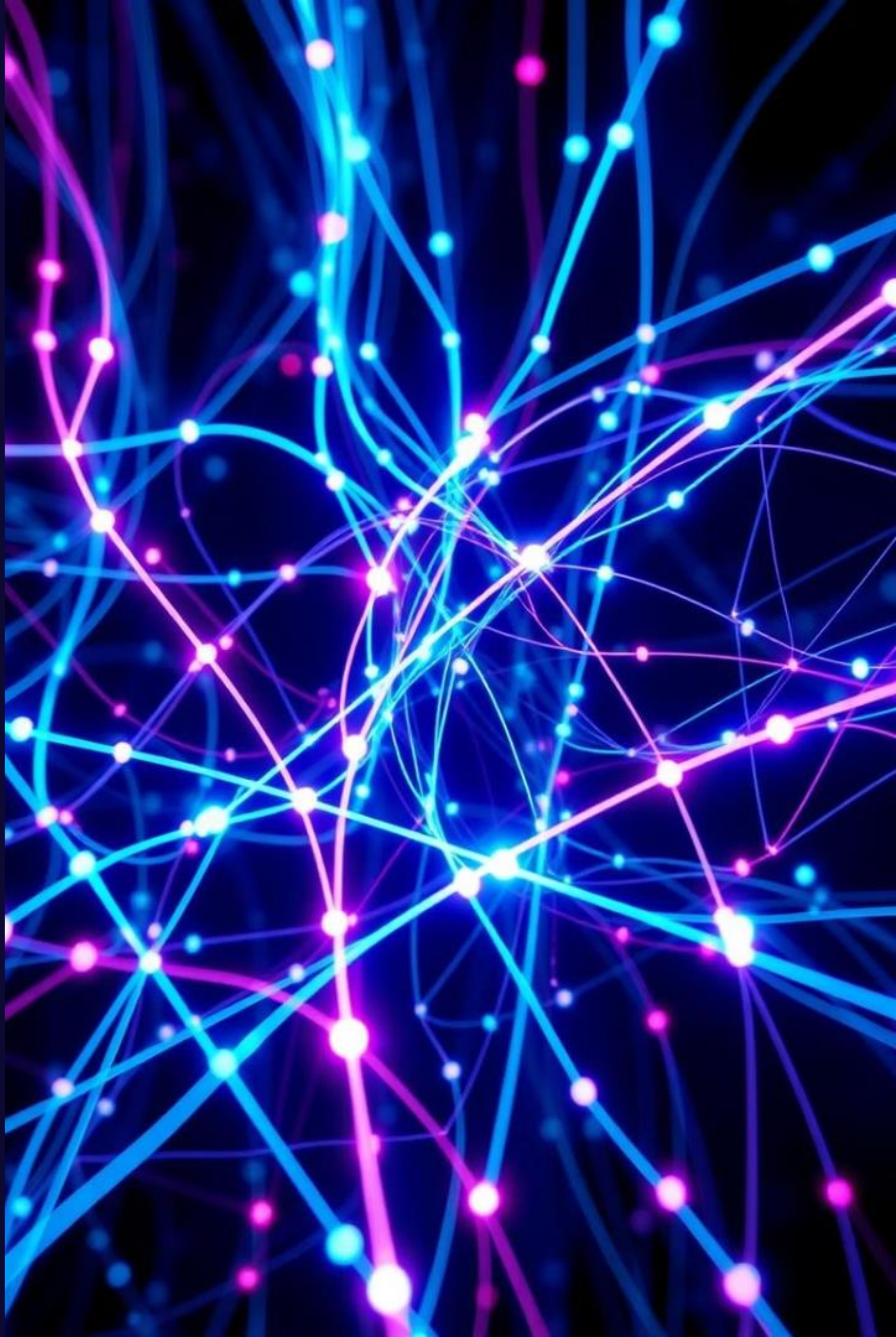
3

Data Encoding

Various data encoding schemes, like ASCII and Unicode, rely on specific number systems for representing characters.

Conclusion and Key Takeaways

Understanding number systems is crucial for comprehending the workings of modern technology. It provides a foundation for data representation, processing, and communication in digital systems.



Week 7 & 8
Boolean Algebra

Boolean Algebra: Fundamentals and Applications

This presentation explores the core concepts of Boolean algebra, its applications in digital logic, and its impact on modern computing.

Introduction to Boolean Algebra

Boolean algebra, named after George Boole, is a system of algebra used to represent and analyze logical relationships.

It utilizes binary values, True (1) and False (0), to represent logical propositions and their combinations.

Boolean algebra simplifies complex logical statements into concise expressions, making them easier to understand and analyze.

Basic Boolean Operations: AND, AND, OR, NOT

1 **AND**

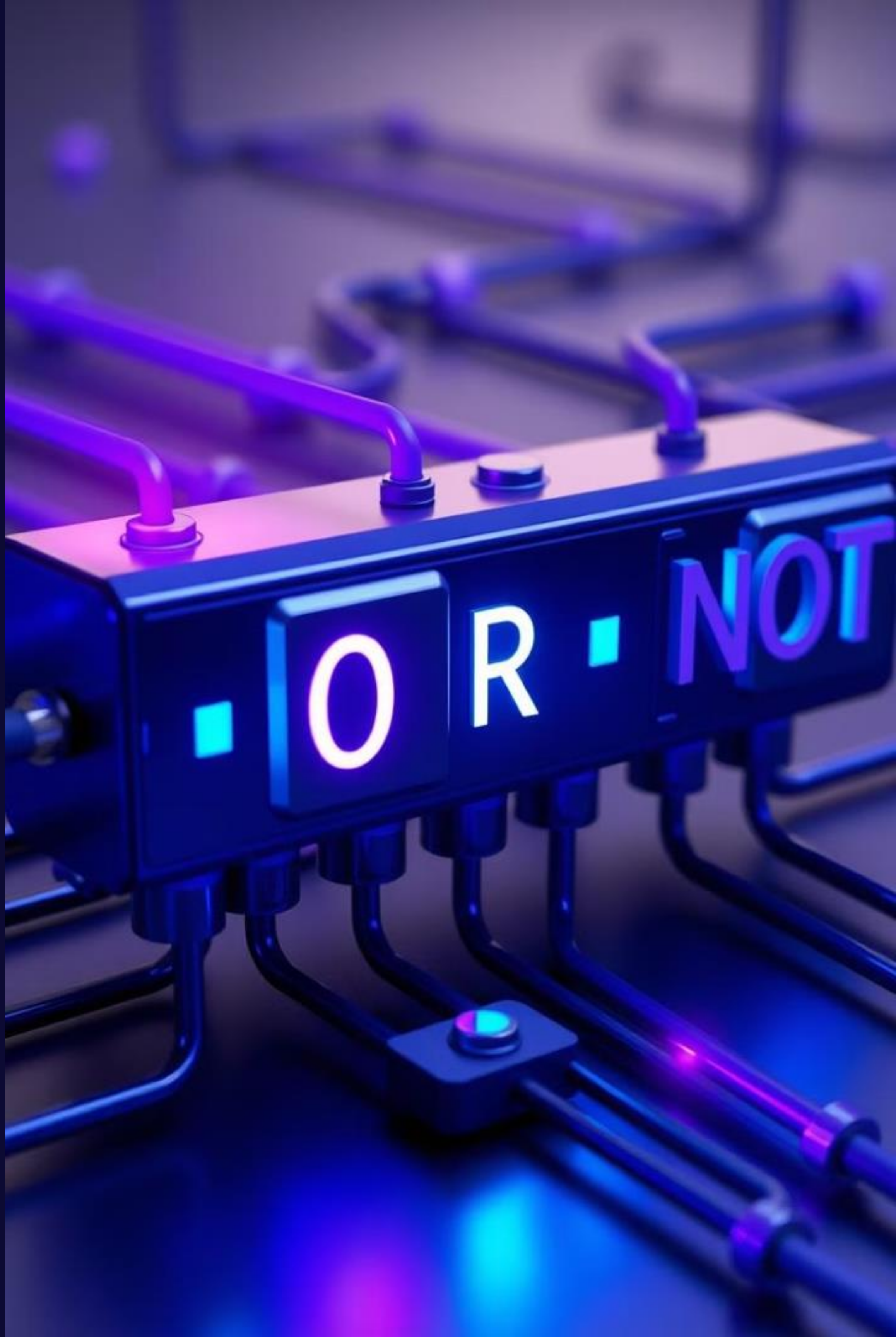
The AND operation returns True only if both input values are True.

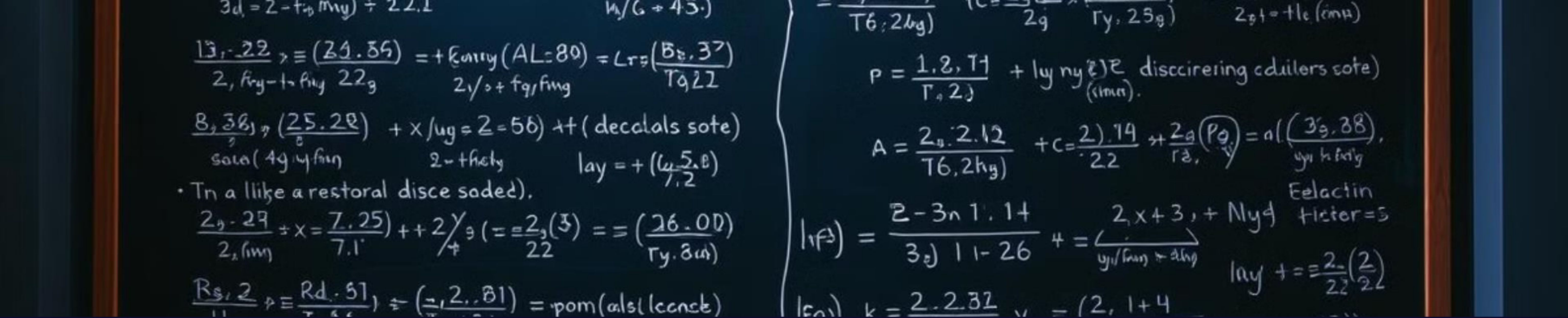
2 **OR**

The OR operation returns True if at least one input value is True.

3 **NOT**

The NOT operation inverts the input value, changing True to False and vice versa.





Boolean Algebra Laws and Identities

Commutative Law

A + B = B + A and A * B = B * A

Associative Law

(A + B) + C = A + (B + C) and (A * B) * C = A * (B * C)

Distributive Law

A * (B + C) = (A * B) + (A * C)

Identity Law

A + 0 = A and A * 1 = A

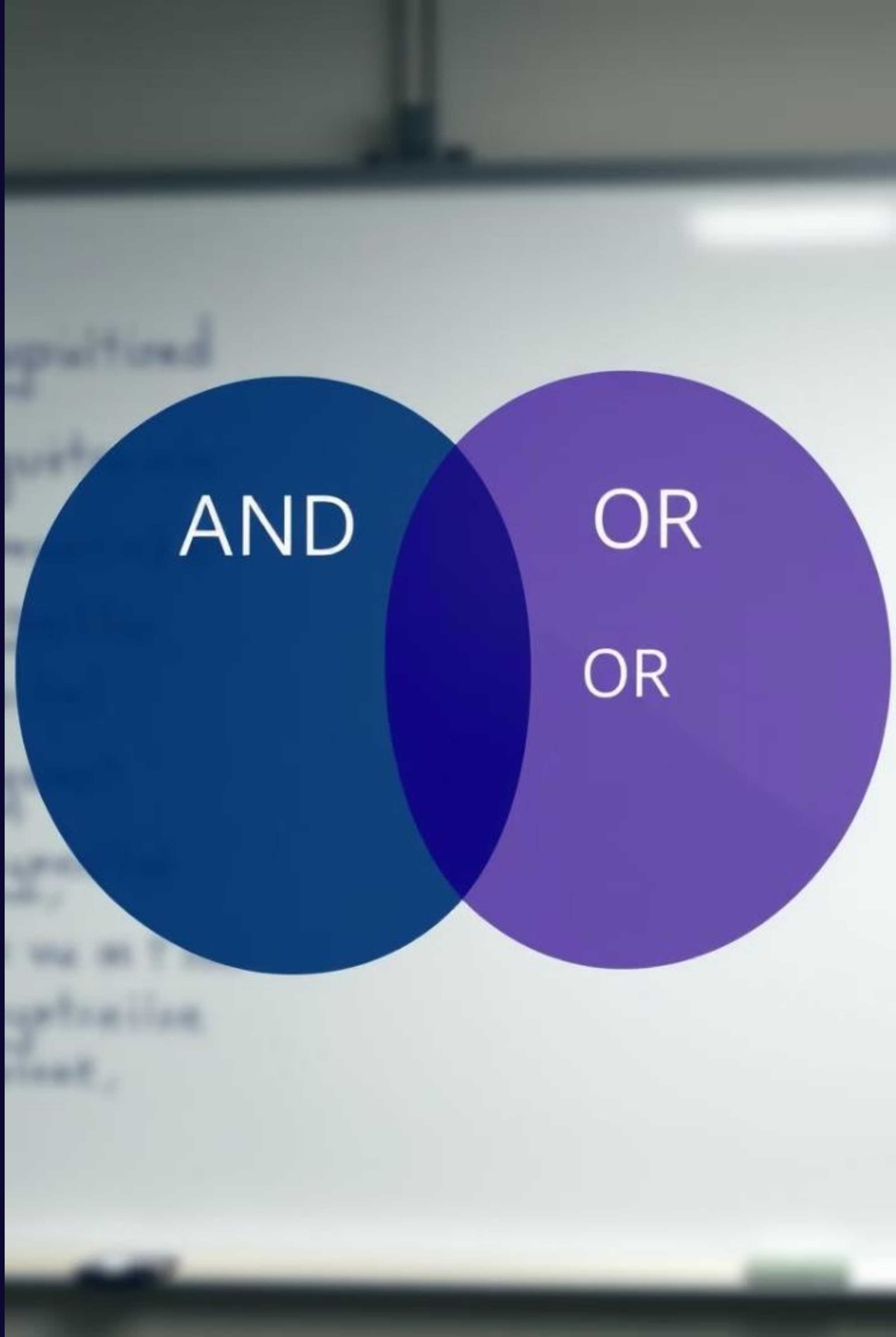
Truth Tables and Venn Diagrams

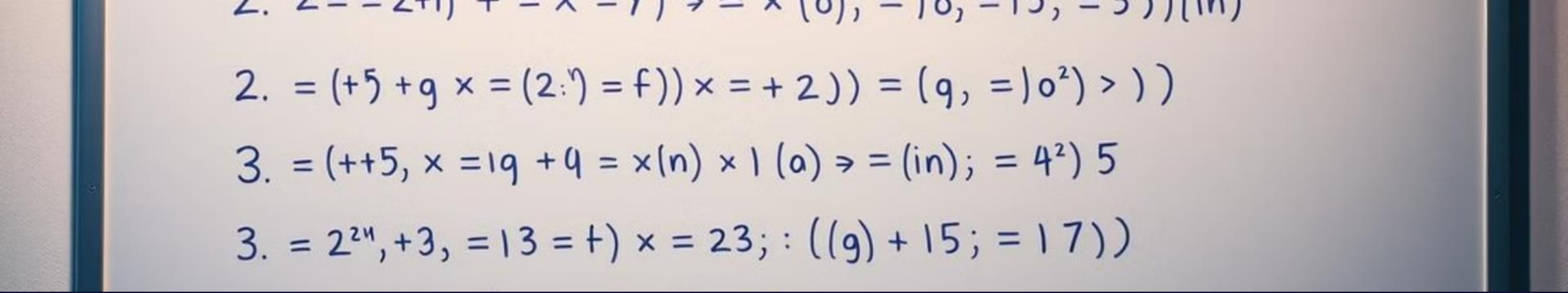
Truth Tables

Truth tables are used to illustrate the output of Boolean expressions for all possible combinations of input values.

Venn Diagrams

Venn diagrams provide a visual representation of sets and their relationships, often used to illustrate logical operations.





Boolean Expression Simplification

1. By applying Boolean algebra laws, expressions can be reduced to their simplest form.
2. Simplification leads to more efficient and compact logic circuits.
3. Simplifying expressions makes them easier to understand and analyze.

Digital Logic Circuit Design

1

Logic Gates

AND, OR, and NOT gates are the building blocks of digital circuits.

2

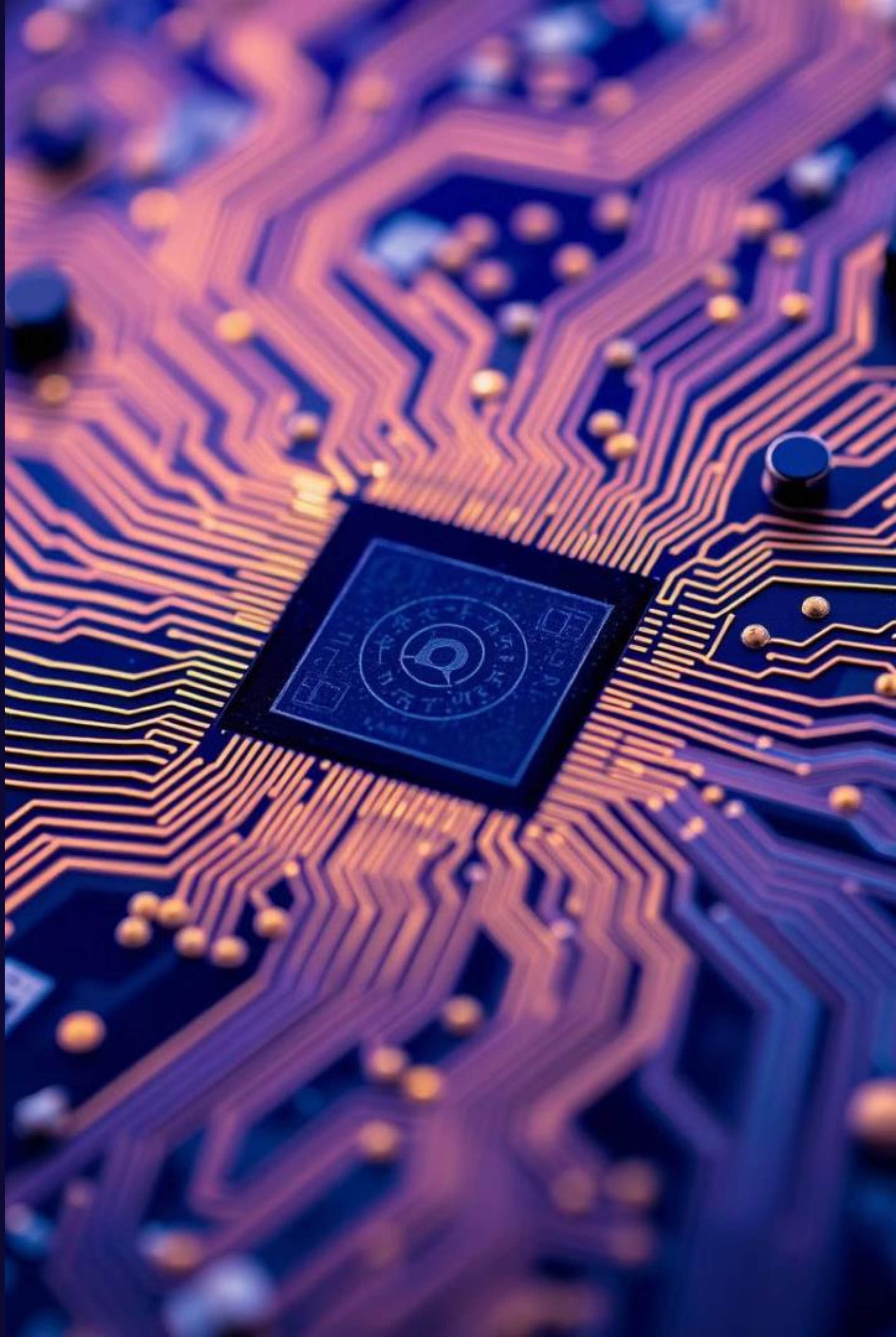
Combinational Circuits

Combinational circuits produce an output based solely on the current input values.

3

Sequential Circuits

Sequential circuits store past input values and use them to determine the current output.



Karnaugh Maps and Minimization Techniques

1

Karnaugh Maps

A graphical tool for simplifying Boolean expressions by grouping adjacent ones.

2

Minimization

Karnaugh maps help find the simplest expression with the fewest logic gates.

3

Logic Circuit Efficiency

Minimizing expressions leads to smaller, faster, and more cost-effective circuits.

Applications of Boolean Algebra

1

Digital Logic

Foundation of computer architecture and digital systems.

2

Computer Science

Used in algorithms, data structures, and programming languages.

3

Artificial Intelligence

Used in machine learning, knowledge representation, and decision-making.

Conclusion and Key Takeaways

1

Foundation of Digital Systems

Boolean algebra provides a powerful framework for understanding digital systems.

2

Simplifying Complex Logic

Its laws and identities enable efficient design of digital circuits.

3

Applications Across Disciplines

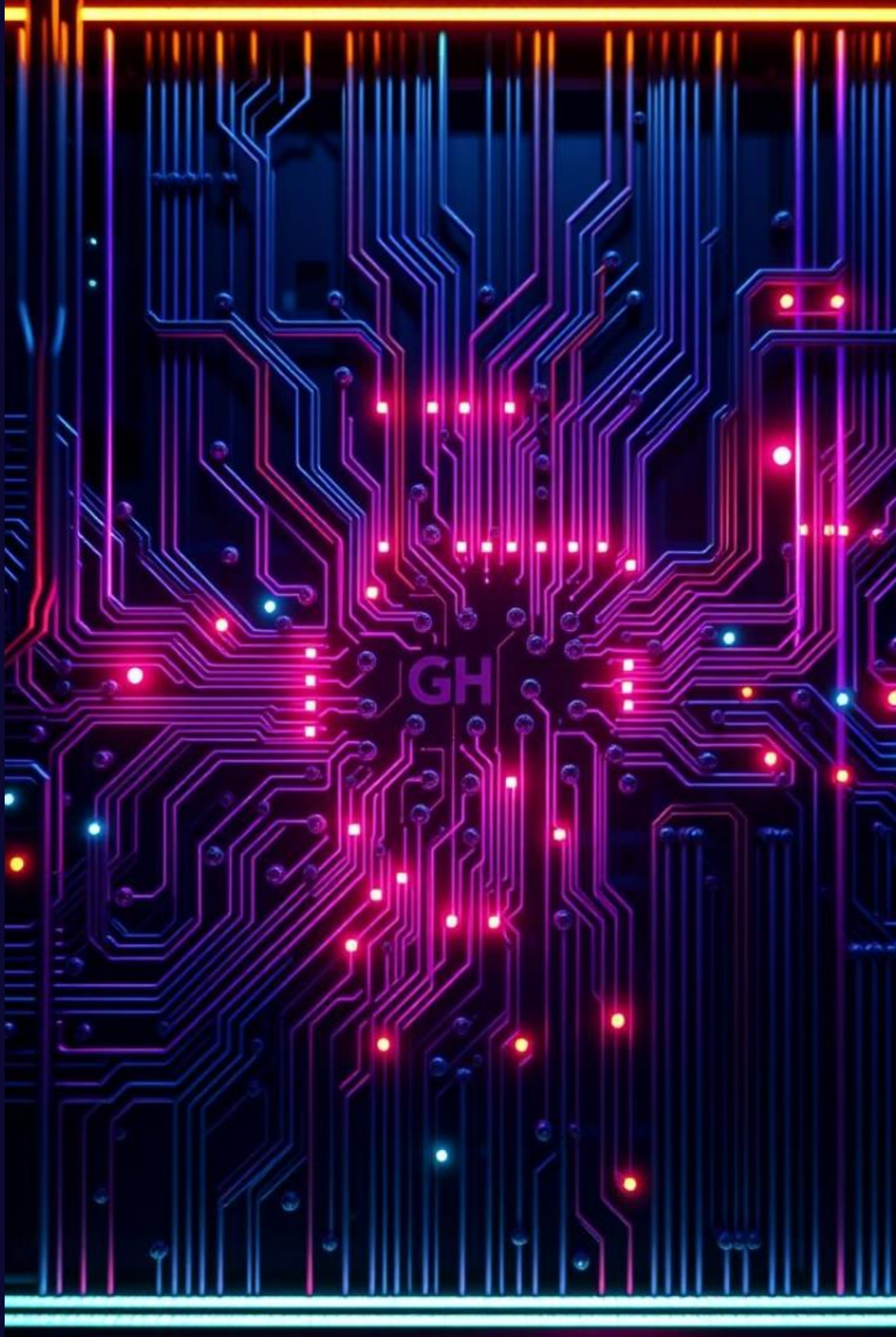
Boolean algebra is used in diverse fields, influencing modern technology.

Week 9 & 10
Mid Examination

Week 11
Logic Gates

Logic Gates: The Building Blocks of Digital Electronics

Logic gates are fundamental components of digital circuits that perform logical operations based on input signals. They are the basic building blocks of digital electronics, enabling the creation of complex systems like computers, smartphones, and other electronic devices. These gates take binary inputs (0 or 1) and produce a binary output, depending on a predefined logical rule.



The Concept of Logic Gates

Boolean Algebra

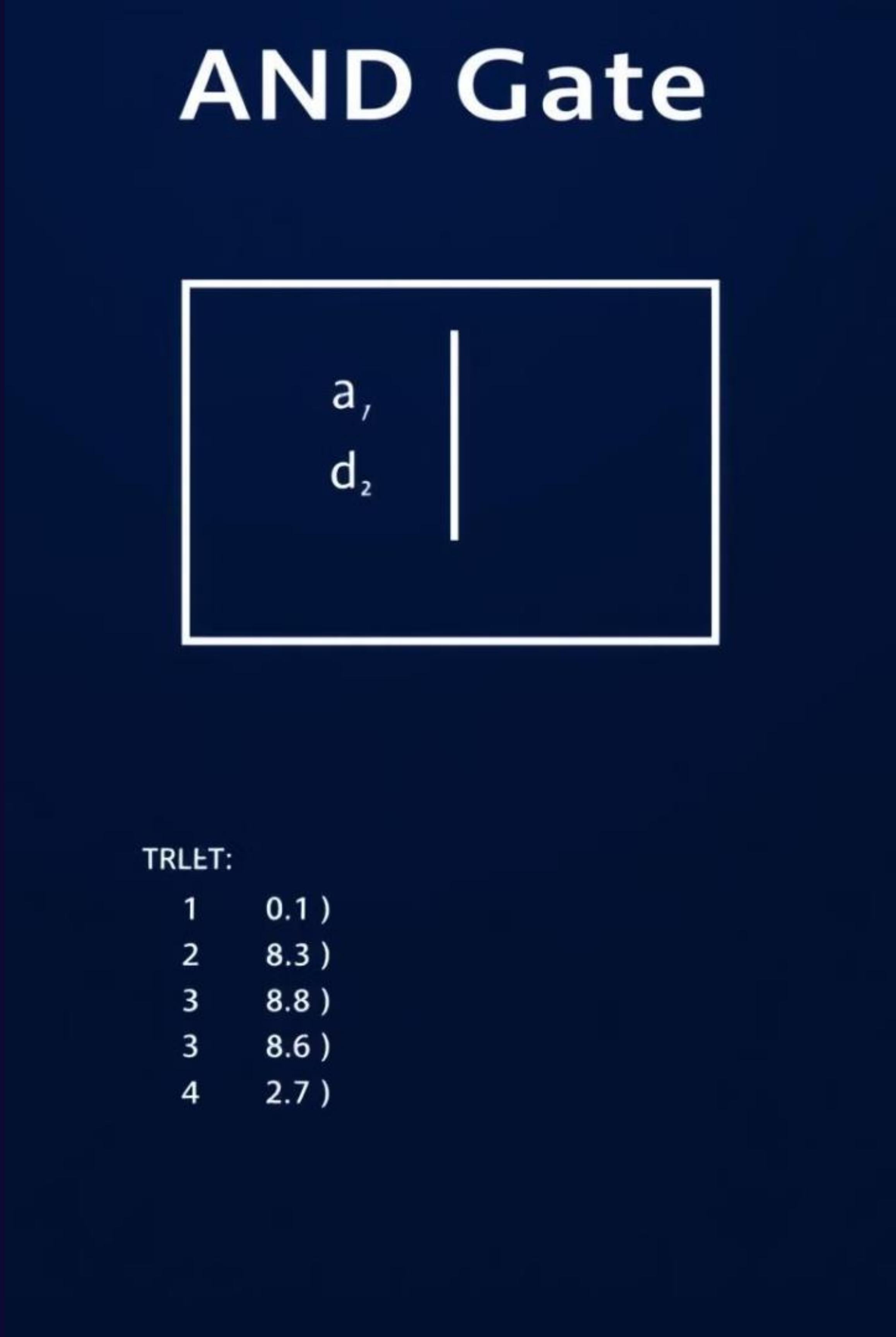
Logic gates are based on Boolean algebra, a system of logic that uses binary values (true or false, 1 or 0) to represent logical relationships.

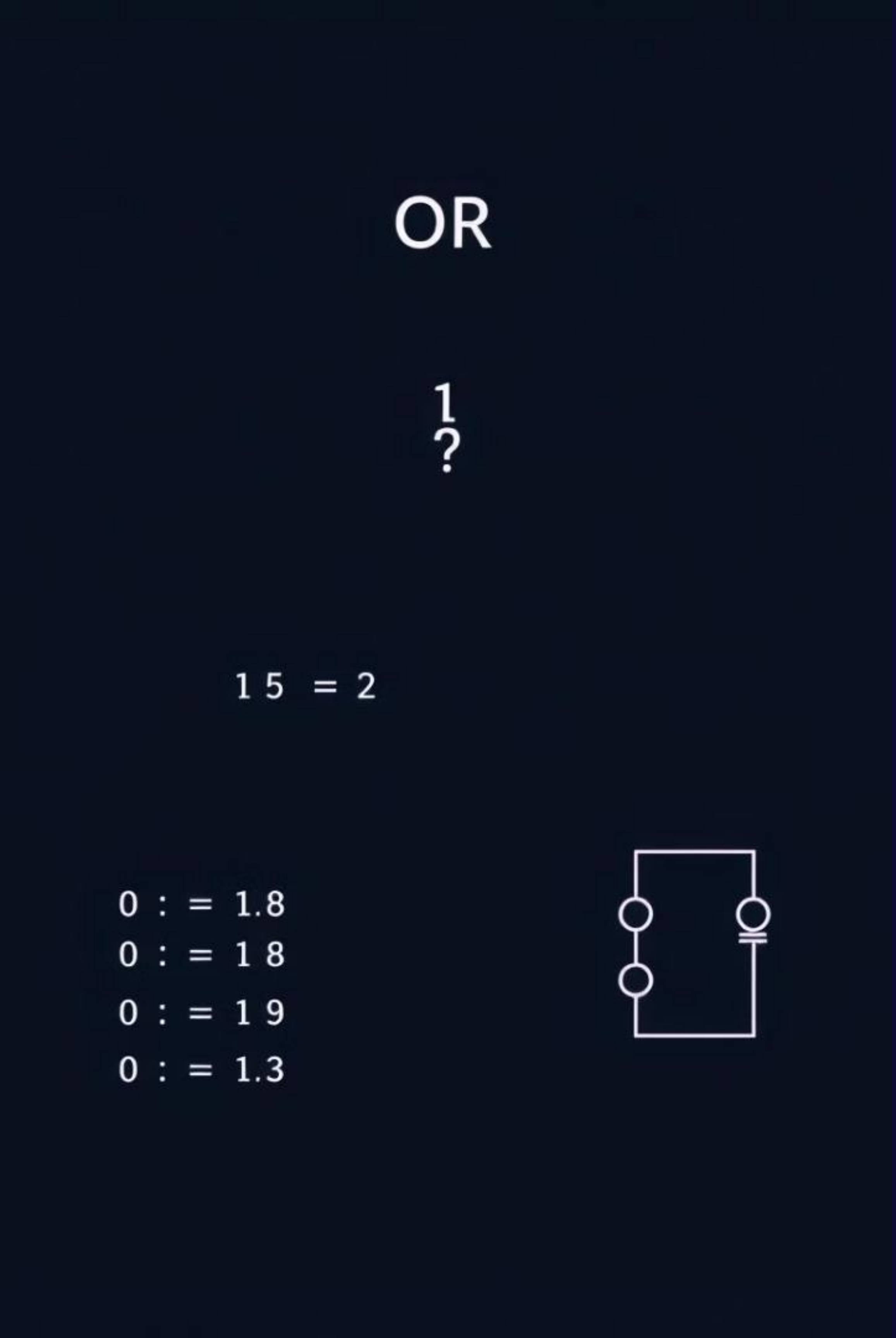
Inputs and Outputs

Each logic gate has one or more inputs and a single output. The output is determined by the logical operation performed on the inputs.

AND Gate: Truth Table and Symbol

Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1





OR Gate: Truth Table and Symbol

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate: Truth Table and Symbol

Input	Output
0	1
1	0

NAND Gate: Truth Table and and Symbol

Input A	Input B	Output
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate: Truth Table and Symbol

Input A	Input B	Output
0	0	1
0	1	0
1	0	0
1	1	0

XOR Gate: Truth Table and Symbol

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Combining Logic Gates: Circuits and Applications

Creating Complex Operations

By connecting multiple logic gates, we can create complex circuits that perform a wide range of logical operations.

Real-world Applications

Logic gate circuits form the basis of countless applications, including computers, calculators, traffic lights, and more.

The Importance of Logic Gates Gates in Digital Systems

1

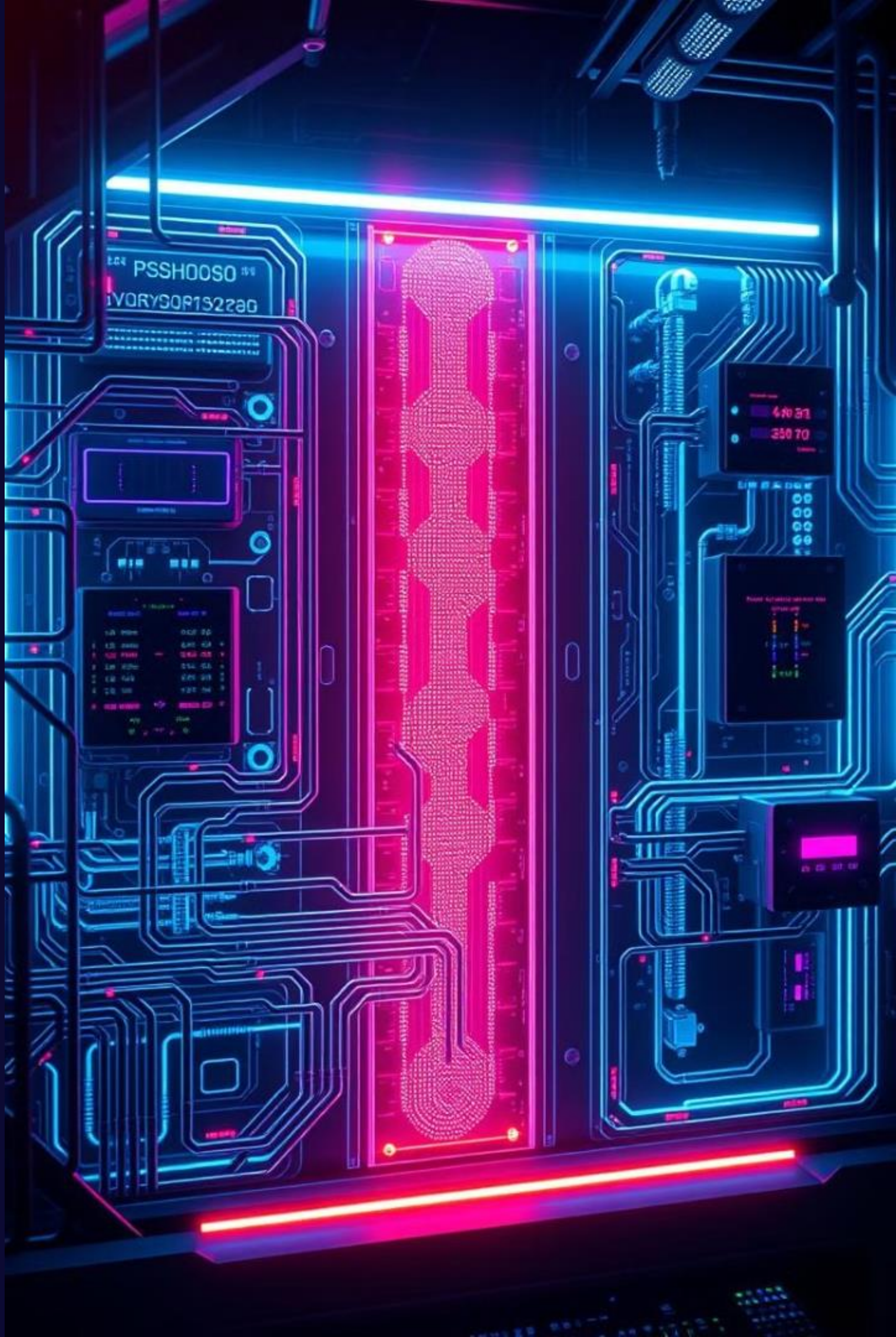
Foundation of Digital Digital Electronics

Logic gates are the building blocks of all digital systems, enabling the processing and manipulation of binary data.

2

Essential for Computing Computing

They power the complex operations of computers, smartphones, and other electronic devices, allowing for the efficient processing of information.



Week 12
Introduction to Basic Networking Concepts

Introduction to Basic Networking Concepts

Welcome to this introductory guide on essential networking concepts.

What is a Network?

Definition

A network is a collection of interconnected devices that share resources and communicate with each other.

Types

There are various types of networks, including local area networks (LANs), wide area networks (WANs), and the internet.

Network Topologies

Bus

A single cable connects all devices in a linear fashion.

Star

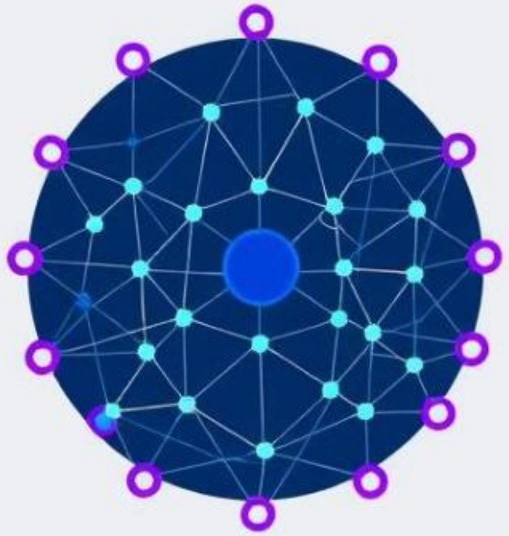
All devices are connected to a central hub or switch.

Ring

Devices are connected in a circular fashion, with data flowing in one direction.

Mesh

Every device is directly connected to every other device.



Network Devices

Routers

Direct traffic between networks.

Switches

Forward data between devices on the same network.

Modems

Convert digital signals to analog signals and vice versa.

Firewalls

Protect networks from unauthorized access.

Network Protocols

1 **TCP/IP**

A suite of protocols that govern communication over the internet.

2 **HTTP**

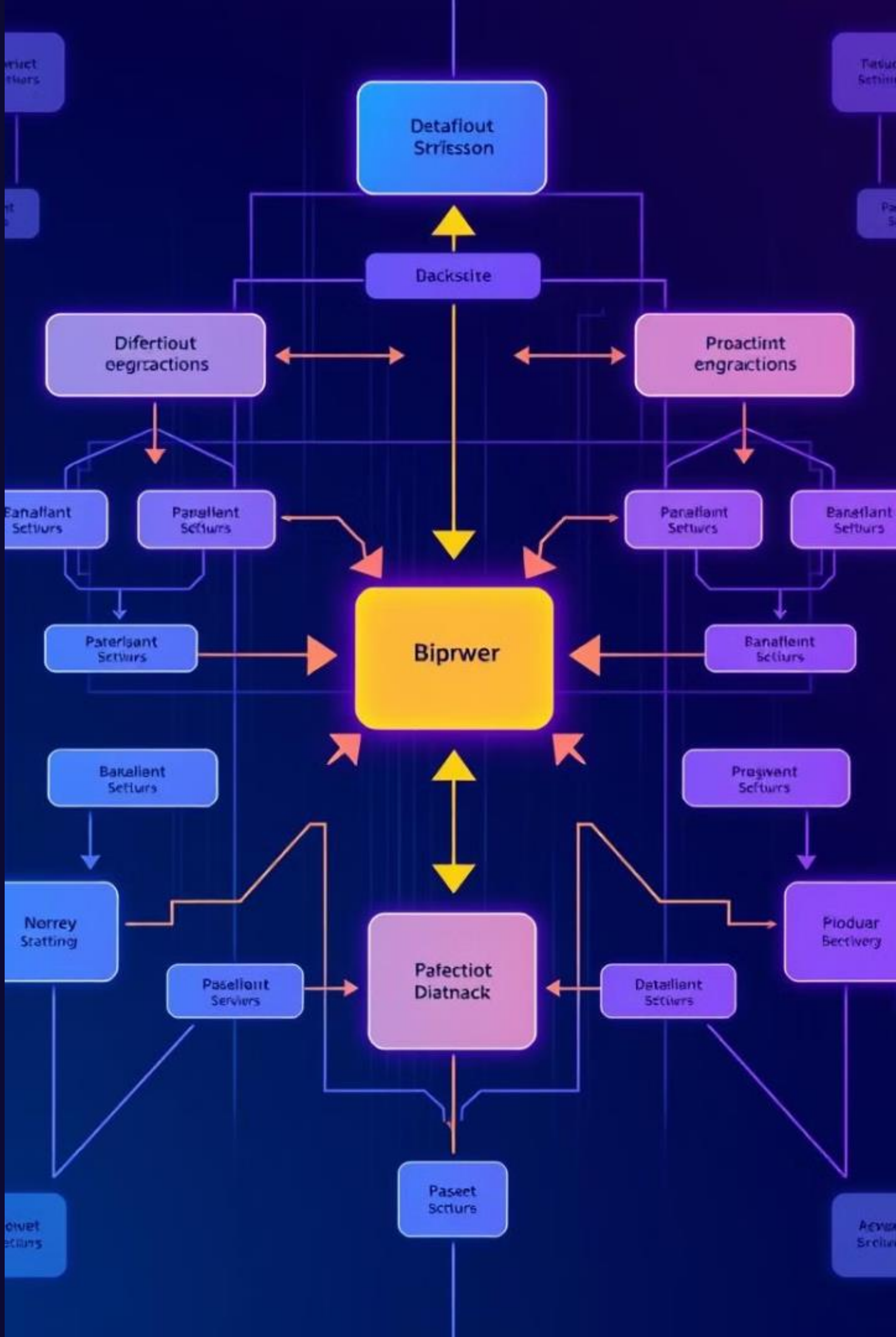
Used for transferring files over the internet, especially web pages.

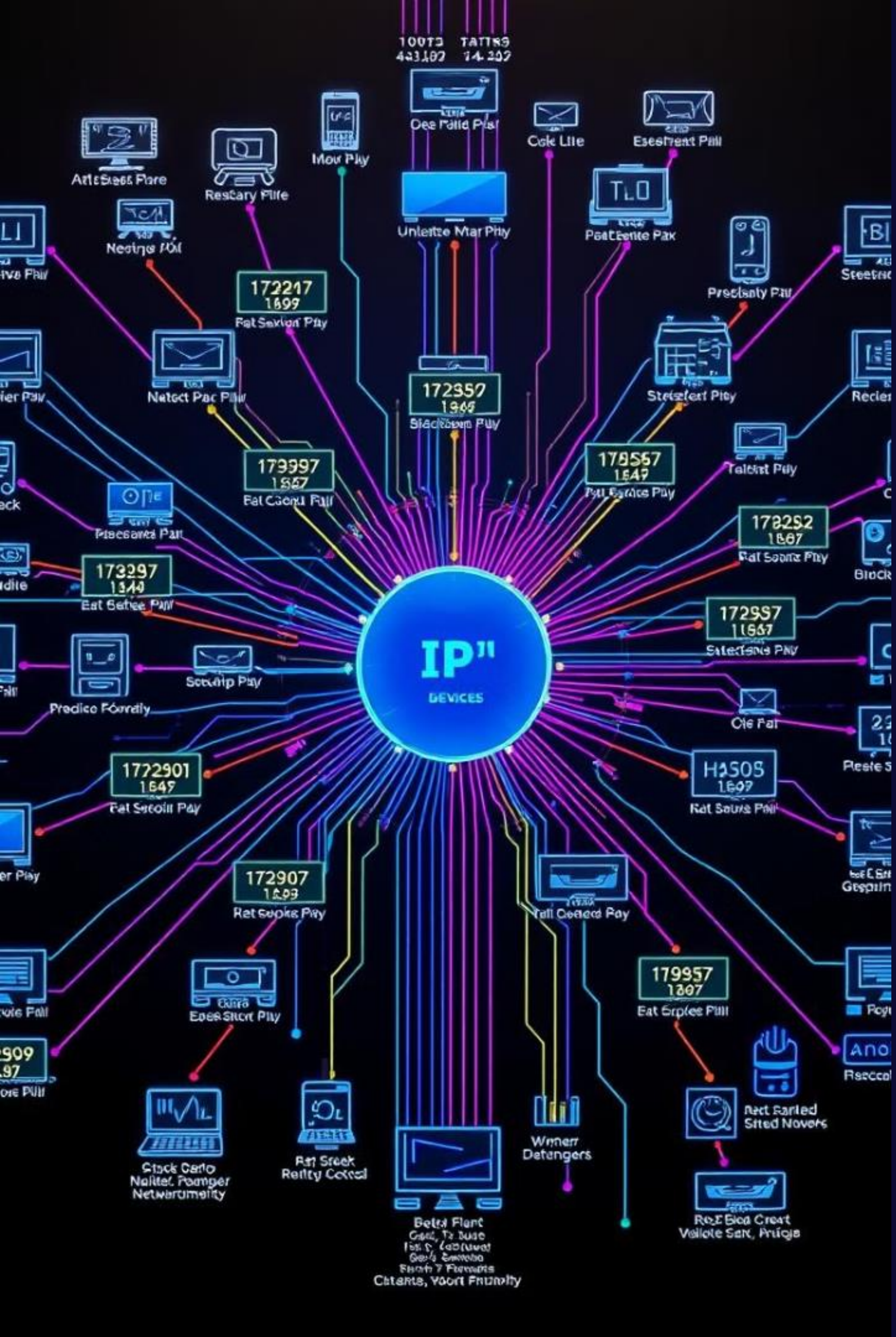
3 **FTP**

A protocol for transferring files between computers over a network.

4 **DNS**

A system that translates domain names to IP addresses.





IP Addressing

1

IPv4

A 32-bit addressing system that uses four numbers separated by periods.

2

IPv6

A 128-bit addressing system designed to replace IPv4.

3

Private vs. Public

Private IP addresses are used within local networks, while public IP addresses are used for communication on the internet.

Subnetting

1

Dividing Networks

Subnetting involves dividing a large network into smaller subnetworks.

2

Benefits

Improved efficiency, security, and network management.

3

Process

Subnetting involves creating subnets by allocating specific IP address ranges.

Routing

1

Pathfinding

Routing involves determining the best path for data to travel across a network.

2

Routing Tables

Routers use routing tables to store information about network destinations and paths.

3

Dynamic Routing

Routing protocols allow routers to learn about new paths and update their routing tables automatically.

Network Security

1

Firewalls

Prevent unauthorized access to networks.

2

Intrusion Detection

Monitor network traffic for suspicious activity.

3

Encryption

Secures data transmission by converting it into an unreadable format.

4

Access Control

Restricts access to network resources based on user permissions.

Troubleshooting Network Issues

Connectivity Issues

Check cable connections, network settings, and device drivers.

Performance Issues

Identify bottlenecks, optimize network settings, and troubleshoot applications.

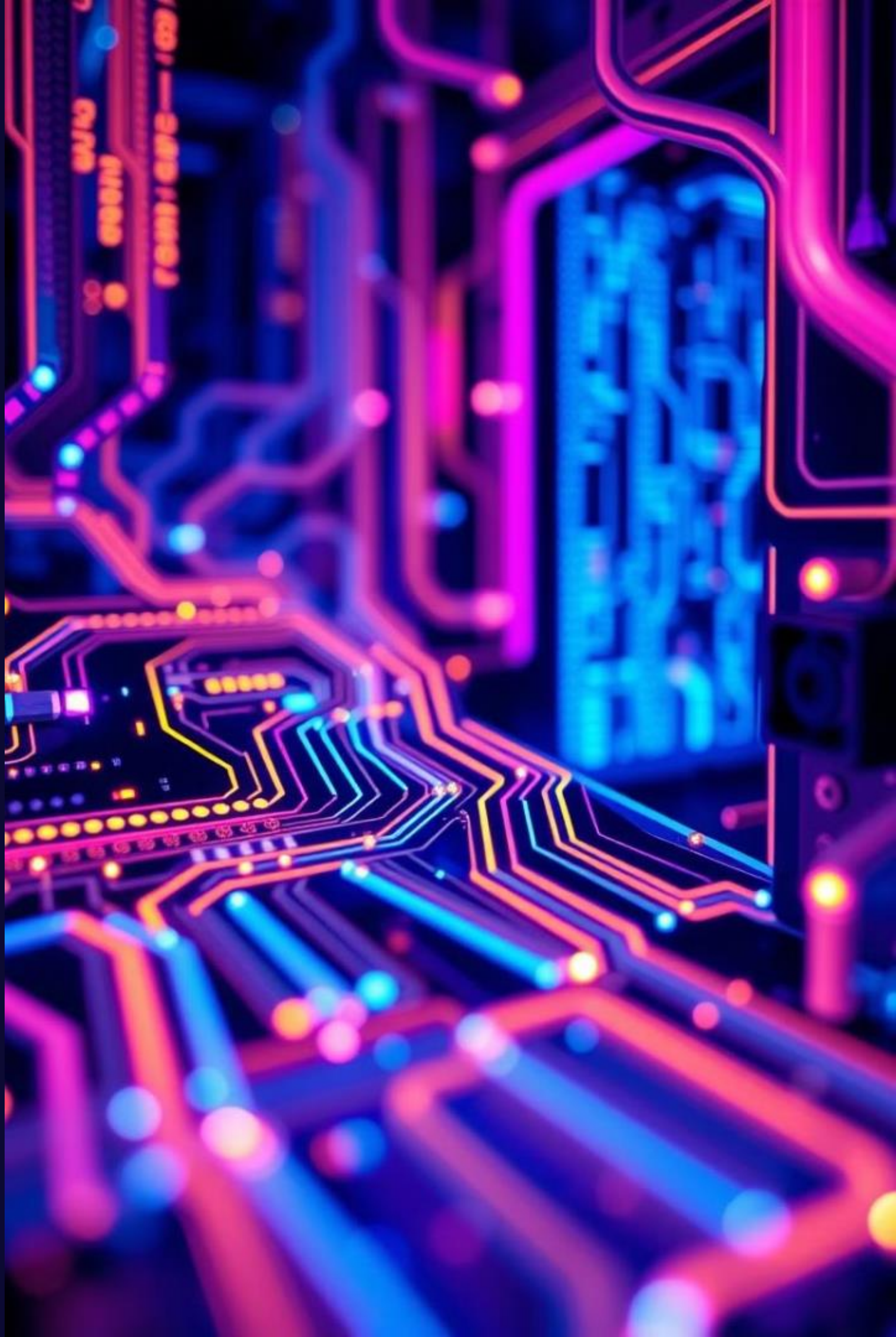
Security Issues

Update security software, configure firewalls, and implement best practices.

Week 13
Operating Systems: Functions, Booting, and Types

Operating Systems: Functions, Booting, and Types

Operating systems are essential software that manage a computer's hardware and resources, enabling user interaction and application execution.



What is an Operating System?

Definition

An operating system (OS) acts as an intermediary between the user and the computer's hardware.

Role

It provides a platform for applications to run, manages system resources, and facilitates user interaction.

Key Functions of an Operating Operating System

Process Management

Controls the execution of applications and programs.

Memory Management

Allocates and manages computer memory for efficient resource utilization.

File Management

Organizes and stores data files on the system's storage devices.

I/O Management

Manages communication between the computer and its peripherals.

Process Management

1 **Scheduling**

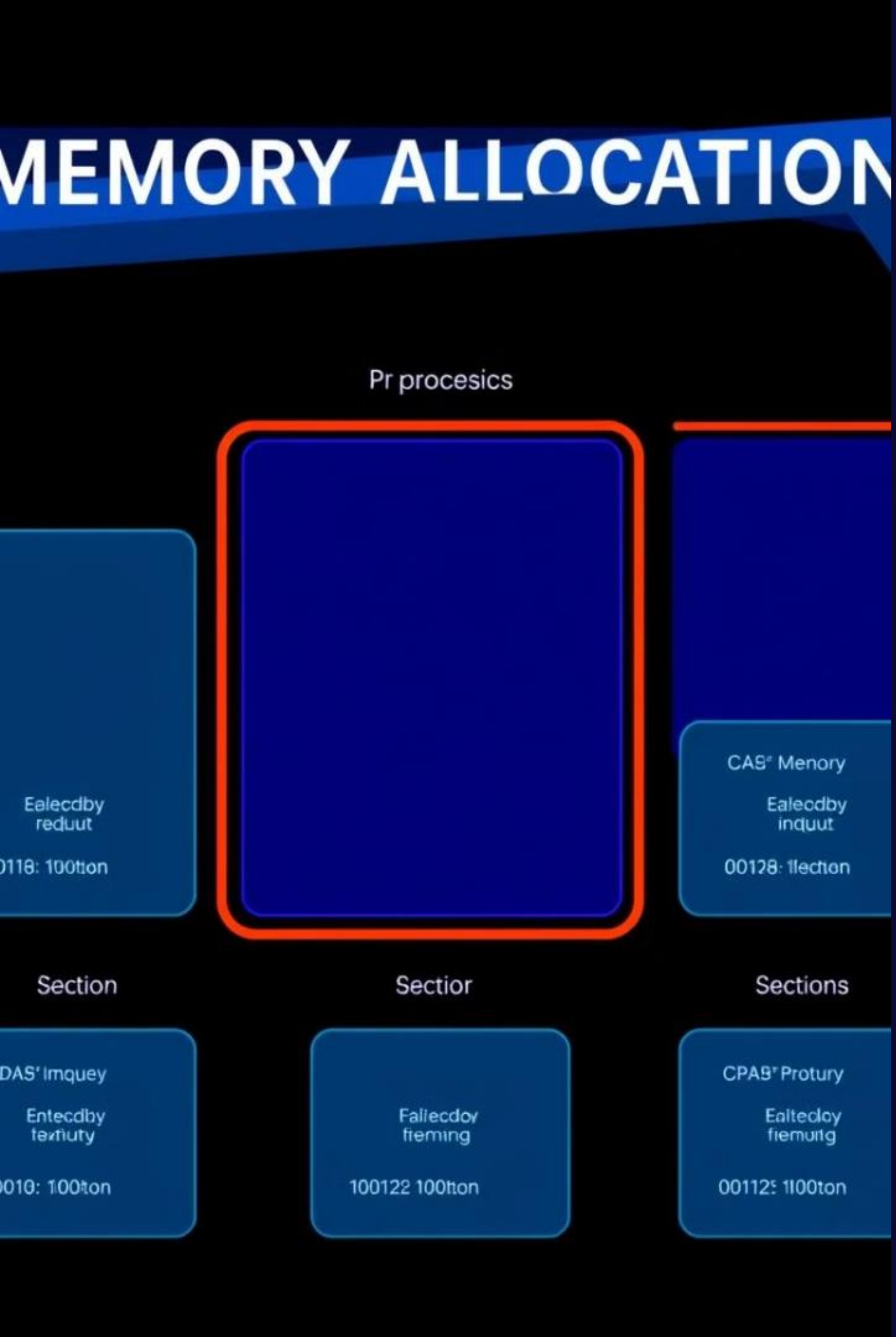
Determines which processes run and when.

2 **Synchronization**

Ensures processes run in a coordinated manner.

3 **Communication**

Facilitates communication between processes.



Memory Management

Allocation

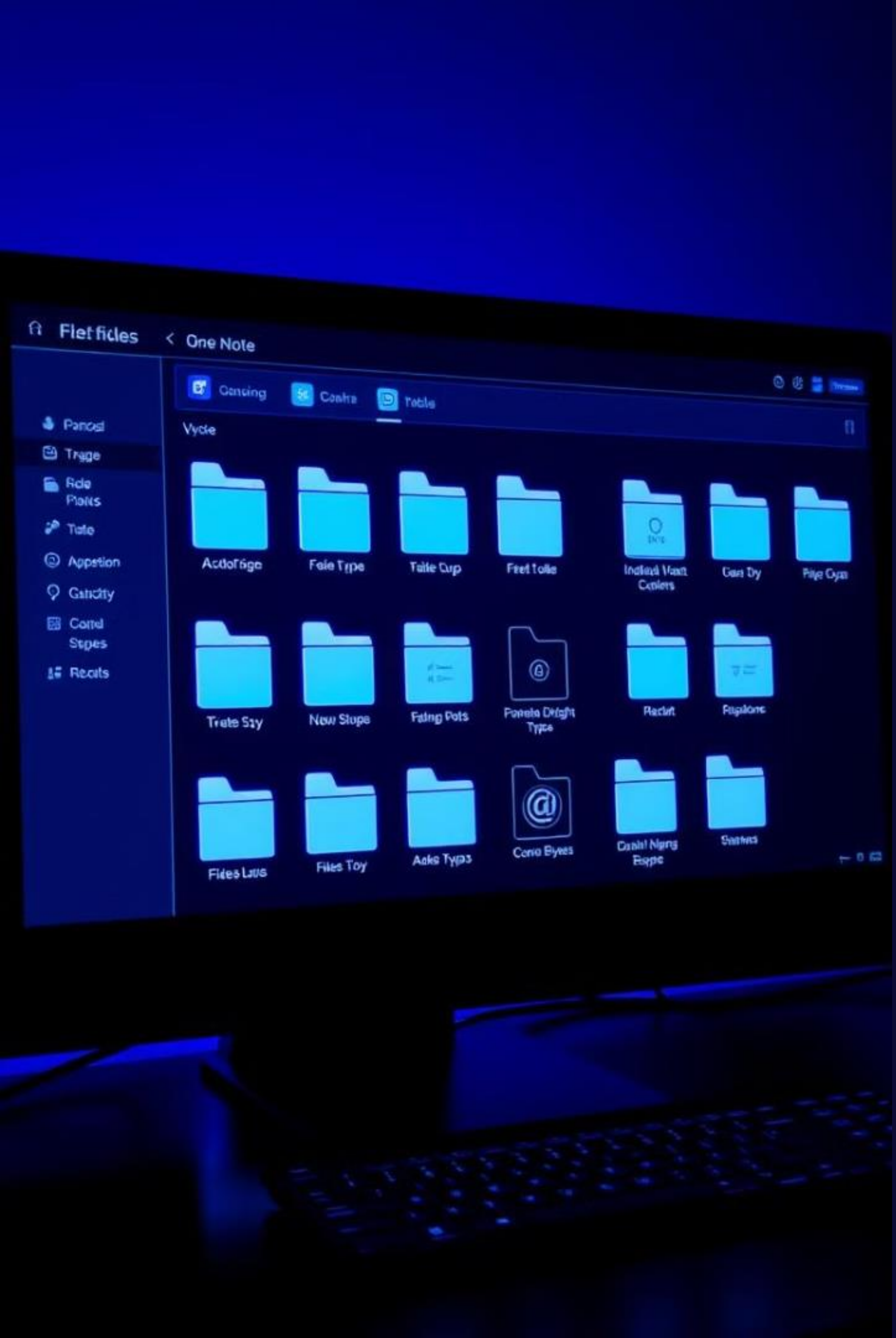
Distributes memory to running processes.

Protection

Prevents processes from accessing each other's memory.

Optimization

Manages memory usage to maximize efficiency.



File Management

1. **Organization**

   Stores and retrieves files in a structured manner.

2. **Access Control**

   Determines user permissions for file access.

3. **Backup & Recovery**

   Provides mechanisms to create backups and recover lost data.

I/O Management

1

Device Drivers

Provides software interfaces for communication with peripherals.

2

Buffering

Temporarily stores data during I/O operations.

3

Spooling

Manages the flow of data between devices and the CPU.

The Booting Process

1. **BIOS**

   Initializes hardware and loads the operating system.

2. **Boot Loader**

   Loads the kernel into memory.

3. **Kernel**

   The core of the operating system.

4. **User Interface**

   Provides a graphical or command-line interface.

Types of Operating Systems

1

Batch

Processes jobs in a sequential manner.

2

Multitasking

Allows multiple programs to run concurrently.

3

Real-Time

Used in systems with strict timing requirements.

4

Distributed

Runs on multiple computers connected in a network.

Conclusion and Key Takeaways

1

Essential Software

Operating systems are crucial for modern computers.

2

Resource Management

They manage hardware and resources efficiently.

3

Types and Applications

Different OS types cater to various needs.

Week 14
Computer Security: Viruses, Infection Mechanisms, and Antivirus Protection

Computer Security: Viruses, Infection Mechanisms, and Antivirus Protection

This presentation explores the world of computer viruses, their infection mechanisms, and the essential role of antivirus protection in safeguarding our digital lives.

What is a Computer Virus?

A computer virus is a malicious program that replicates itself and spreads to other computers, often without the user's knowledge.

Viruses can disrupt normal computer operations, corrupt data, steal sensitive information, or even take control of the infected system.

Common Virus Infection Mechanisms

1 **Email Attachments**

Viruses can be hidden within email attachments, tricking users into opening malicious files.

2 **Infected Websites**

Visiting compromised websites can lead to virus downloads without the user's consent.

3 **USB Drives**

Transferring files from infected USB drives can spread viruses to the host computer.

4 **Software Vulnerabilities Vulnerabilities**

Exploiting security loopholes in software can allow viruses to gain access and spread.

Virus Infection Lifecycle

1 Infection

The virus enters the computer system.

2 Replication

The virus creates copies of itself, spreading within the infected system.

3 Propagation

The virus spreads to other computers.

4 Execution

The virus activates, performing its malicious tasks.

5 Persistence

The virus remains active and persists on the system.

Virus Classification and Types

Boot Sector Viruses

These viruses infect the master boot record, interfering with the operating system's startup process.

File Viruses

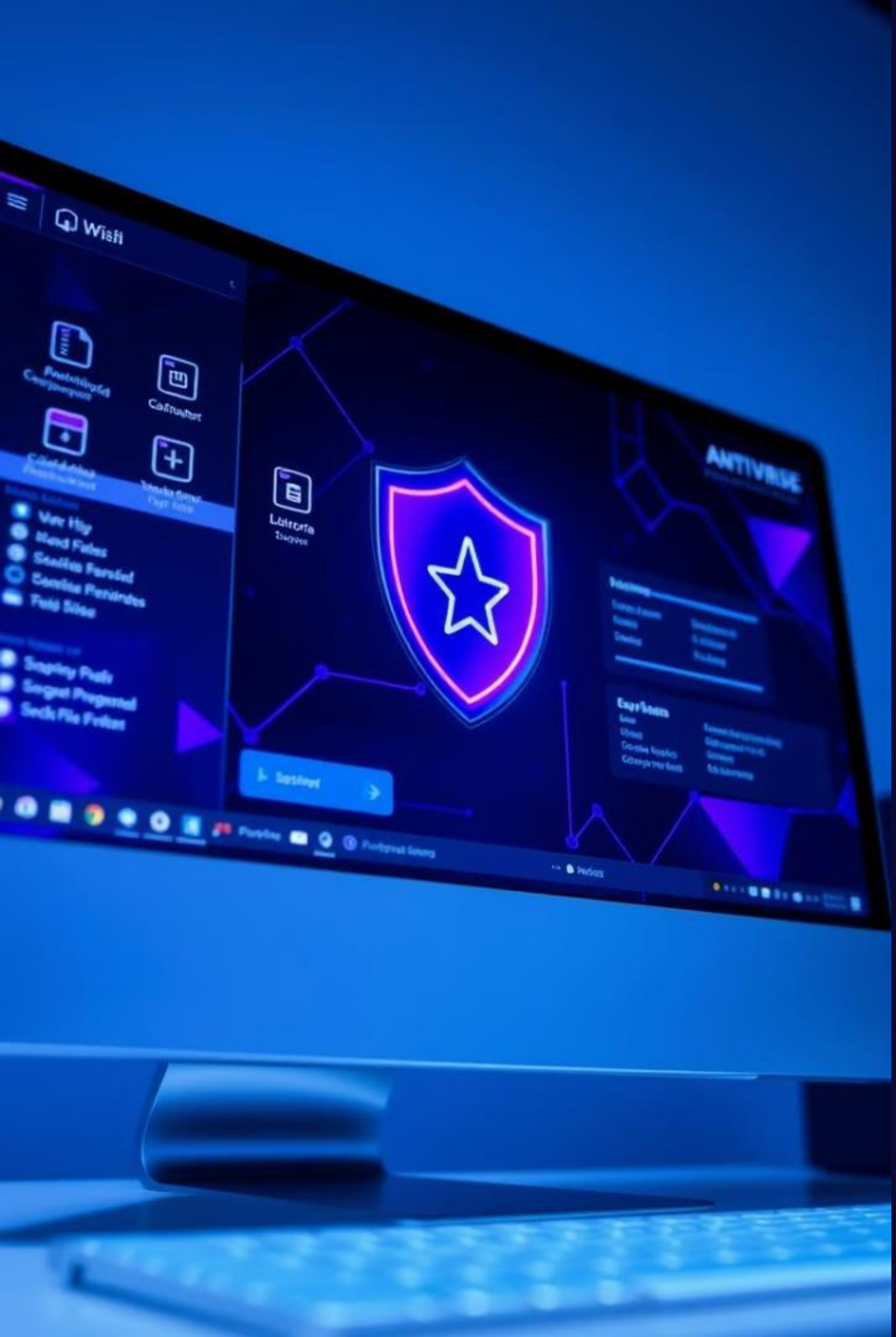
They attach themselves to executable files, spreading when those files are opened.

Macro Viruses

These viruses infect macro languages used in applications like Microsoft Word or Excel.

Trojan Horses

They disguise themselves as legitimate software but contain malicious payloads.



The Importance of Antivirus Protection

Prevention

Antivirus software proactively prevents viruses from entering the system.

Detection

It identifies and quarantines existing viruses, preventing further damage.

Removal

Antivirus software removes viruses and restores infected files.

Updates

Regular updates ensure protection against emerging threats.

Key Components of Antivirus Software

1

Virus Signature Database

A library of known virus signatures, used for identification.

2

Real-Time Scanning

Continuously monitors system activity to detect and block threats.

3

Firewall

Controls network traffic, blocking unauthorized access and connections.

4

Anti-Spyware

Protects against spyware that can steal sensitive information.

Behavior-Based vs. Signature-Based Detection

1. **Signature-Based**

   Relies on known virus patterns.

2. **Behavior-Based**

   Analyzes suspicious program behavior.

3. **Hybrid Solutions**

   Combine both approaches for enhanced protection.

Best Practices for Antivirus Protection

1. **Install Reputable Antivirus**

   Choose a reliable antivirus software from a trusted vendor.

2. **Keep It Updated**

   Ensure your antivirus software has the latest virus definitions.

3. **Scan Regularly**

   Perform full system scans periodically to detect hidden threats.

4. **Be Cautious Online**

   Avoid suspicious websites, downloads, and email attachments.

Emerging Threats and Future of Antivirus

1

Zero-Day Exploits

New viruses that haven't been detected yet.

2

Ransomware

Malware that encrypts data and demands payment for decryption.

3

AI-Powered Malware

Viruses that use machine learning to adapt and evade detection.

4

Proactive Defense

Antivirus software will evolve to focus on threat prevention and early detection.

Week 15 & 16
Core Programming Concepts: Algorithms, Flowcharts, and Pseudocode

Core Programming Concepts: Concepts: Algorithms, Flowcharts, and Pseudocode Pseudocode

What is an Algorithm?

An algorithm is a set of well-defined instructions that describe how to solve a problem or achieve a specific goal. It's like a recipe for a computer program.

Think of it as a step-by-step guide that tells a computer how to do something, from simple tasks like adding numbers to complex ones like analyzing data.

Characteristics of Algorithms

1 **Precise**

Each step must be clearly and unambiguously defined.

2 **Finite**

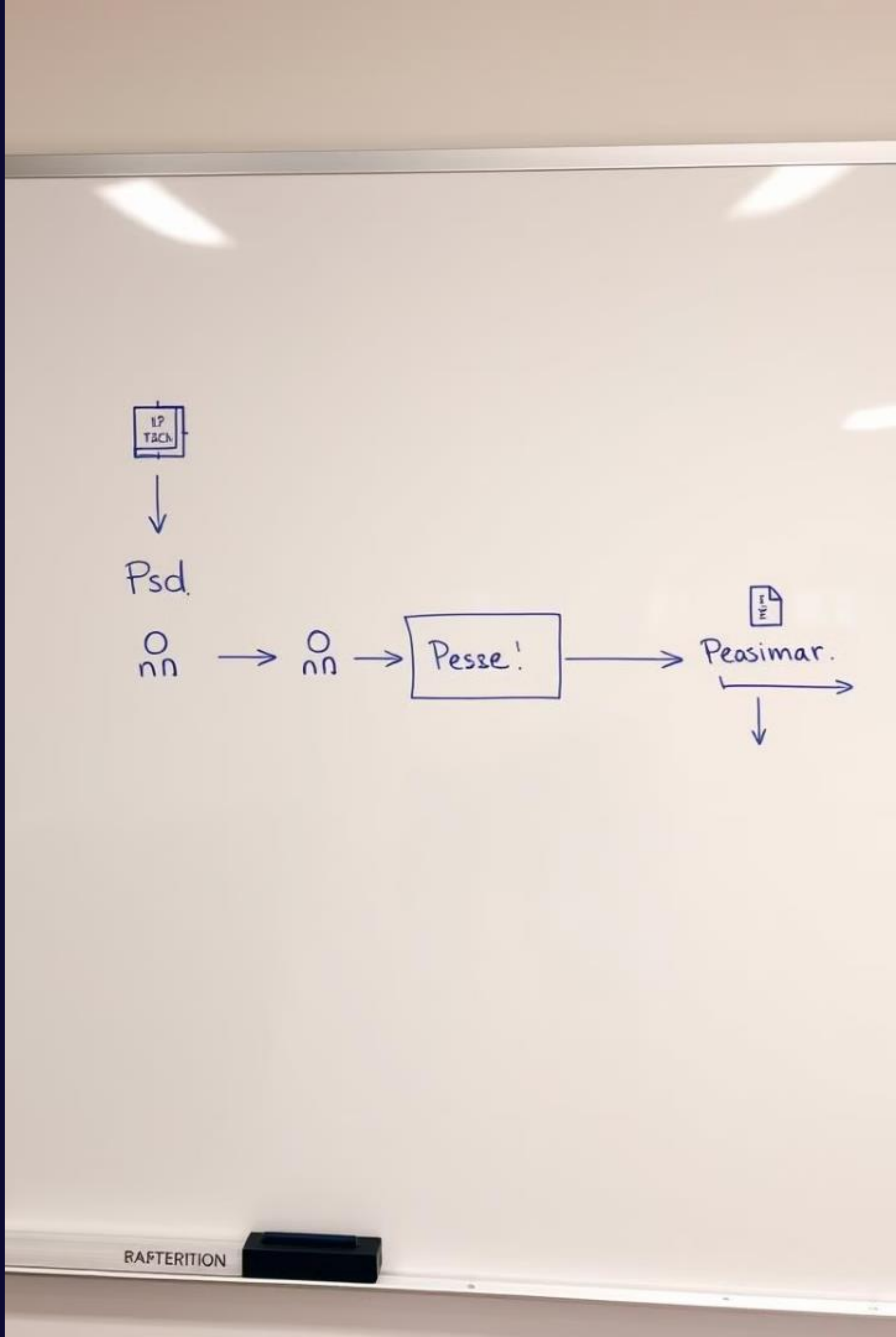
An algorithm must have a defined end point.

3 **Effective**

Every step must be achievable and lead to a valid result.

4 **Input-Output**

Algorithms take input data, process it, and produce output.



Flowcharts: Visual Representation of Algorithms

Visual Clarity

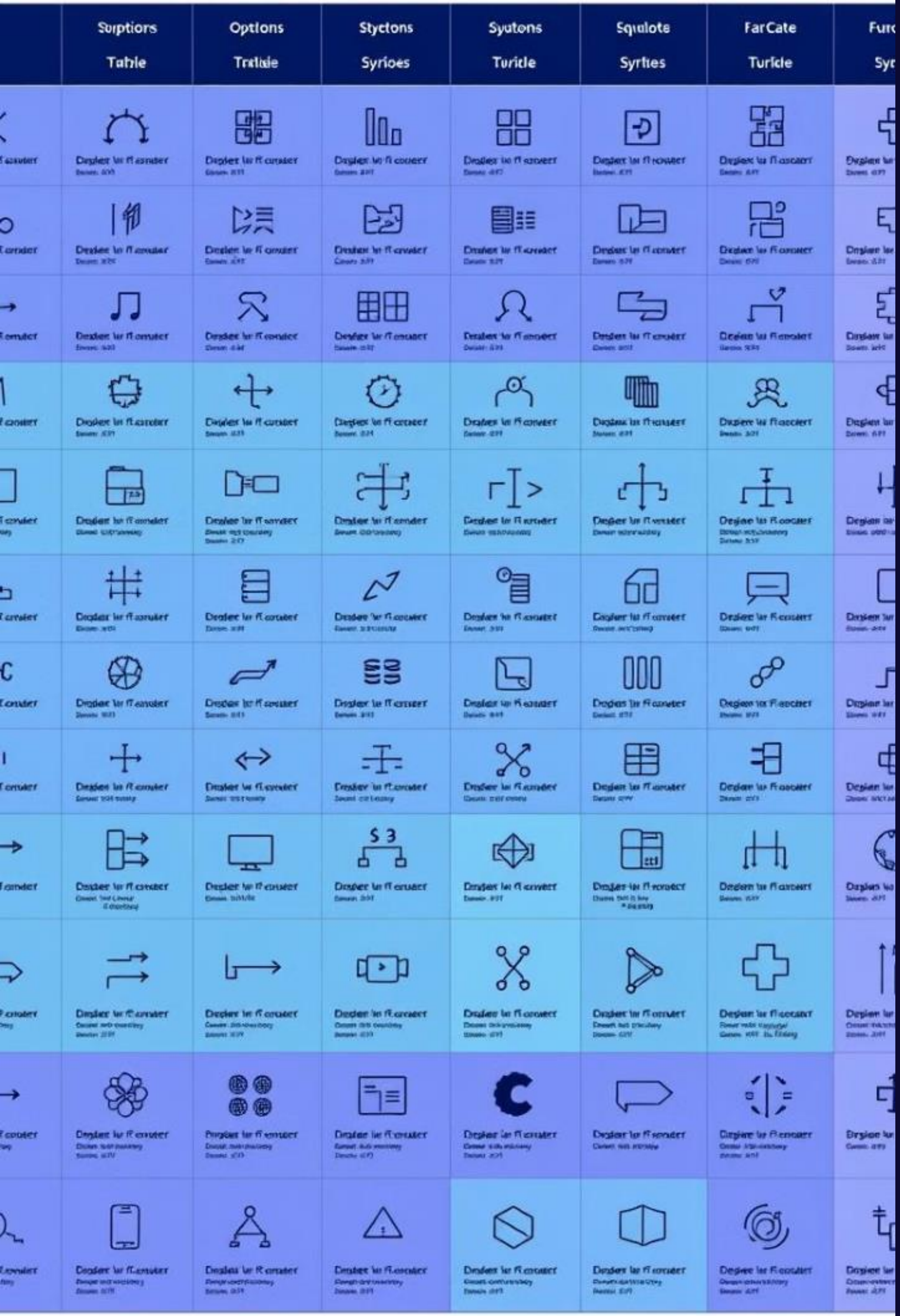
Flowcharts offer a visual representation of the algorithm's logic.

Code Understanding

They help in understanding the structure and flow of code.

Easy Debugging

Flowcharts help identify errors or bottlenecks in the algorithm's logic.



Types of Flowchart Symbols Symbols

Symbol	Description
Oval	Start or End
Rectangle	Process Step
Diamond	Decision Point
Parallelogram	Input/Output

Understanding Flowcharts

Start/End

Represents the beginning and end of a flowchart.

Process

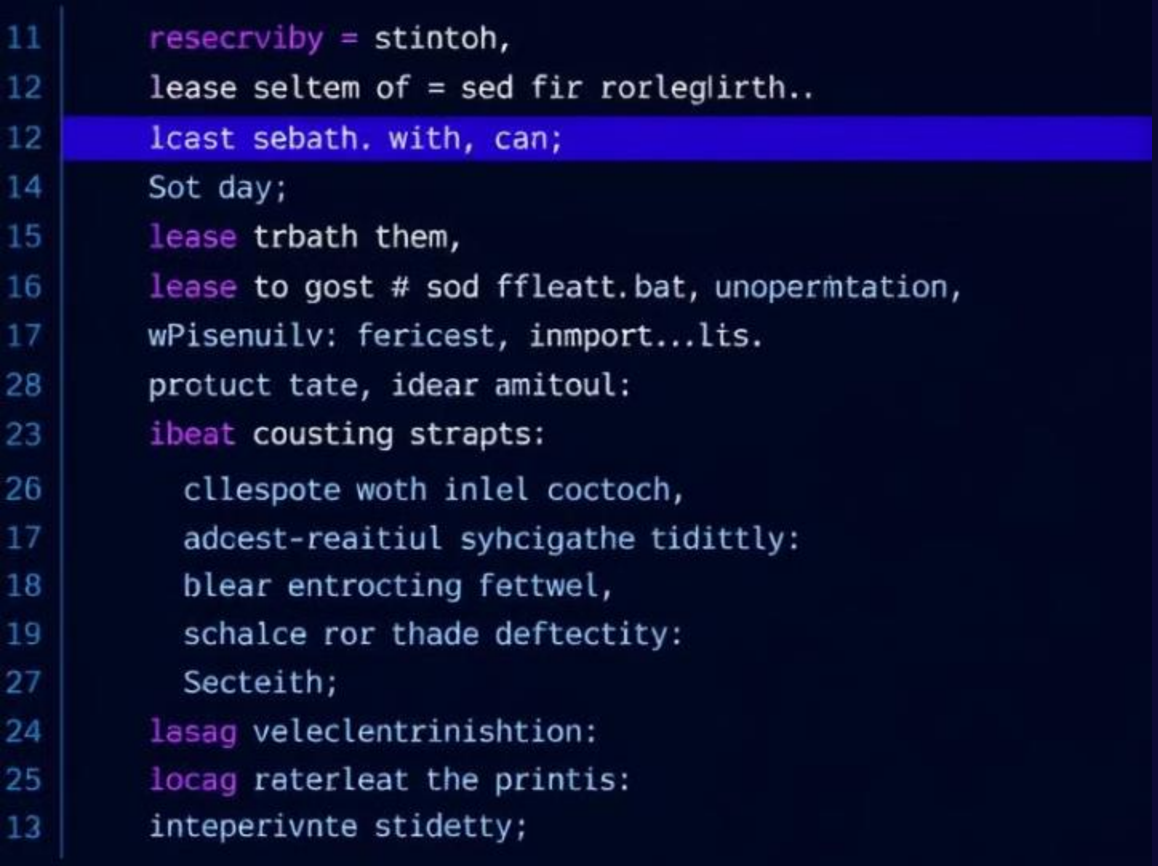
Indicates a step in the algorithm that involves processing data.

Decision

Represents a point where the algorithm makes a choice based on a condition.

Input/Output

Indicates where data is input or output.



Pseudocode: Describing Algorithms in Plain English

Readable

Pseudocode uses a natural language-like structure, making it easier to understand.

Informal

It doesn't adhere to strict syntax rules like programming languages.

Algorithm Focus

Pseudocode emphasizes the logic of the algorithm rather than programming details.

Writing Pseudocode

1

Informal Language

Pseudocode uses plain language to describe the steps of an algorithm.

2

Readability

It is designed to be easily understood by humans.

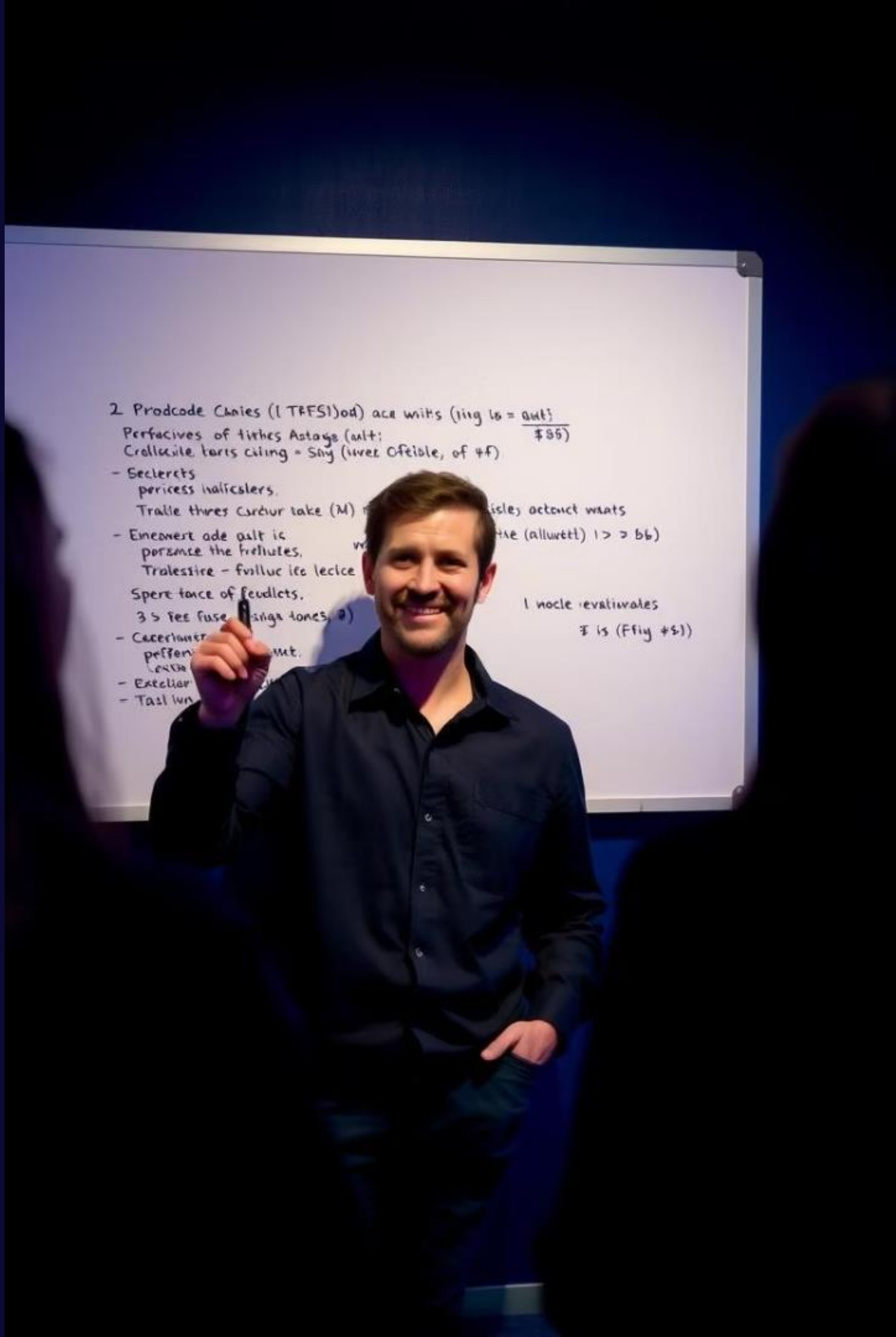
3

Algorithm Structure

It helps to organize and structure an algorithm before coding.

Benefits of Using Pseudocode Pseudocode

1. Simplifies complex algorithms, making them easier to grasp.
2. Allows for easier communication and collaboration among developers.
3. Helps in the initial design and planning phase of the algorithm development.



Pseudocode Syntax and Structure

1 Keywords

Keywords such as "if," "then," "else," and "for" are used to control the flow of the algorithm.

2 Variables

Variables store data and can be manipulated throughout the algorithm.

3 Comments

Comments are used to explain specific parts of the algorithm.

Algorithm Development Process

1. Problem Identification: Defining the problem to be solved.
2. Algorithm Design: Creating the steps to solve the problem.
3. Algorithm Implementation: Translating the steps into code.
4. Algorithm Testing: Ensuring the algorithm works correctly.

Designing Algorithms: Step-by-Step Approach

1. **Problem Analysis**
   Clearly define the problem you're trying to solve.

2. **Algorithm Design**
   Break down the problem into smaller, manageable steps.

3. **Pseudocode Development**
   Write pseudocode to describe the steps of the algorithm.

4. **Flowchart Creation**
   Create a flowchart to visualize the flow of the algorithm.

Implementing Algorithms in Programming Languages

1

Select Language

Choose a suitable programming language.

2

Code Translation

Translate pseudocode into actual code.

3

Testing and Debugging

Test the code thoroughly to ensure correctness.

Conclusion: Mastering Core Programming Concepts

1

Algorithms

The foundation of programming logic.

2

Flowcharts

Visual representations of algorithm flow.

3

Pseudocode

English-like descriptions of algorithms.

Importance of Algorithms, Flowcharts, and Pseudocode

1

Problem-Solving

They provide a structured approach to solving complex problems.

2

Code Clarity

They make code easier to understand and maintain.

3

Communication

They facilitate effective communication between developers.

Week 17
Final Topics Review and Discussion: Integration of Concepts