Topic 03: Hardware

IGCSE Computer Science 0478 | SenpaiCorner | Mr. Wei

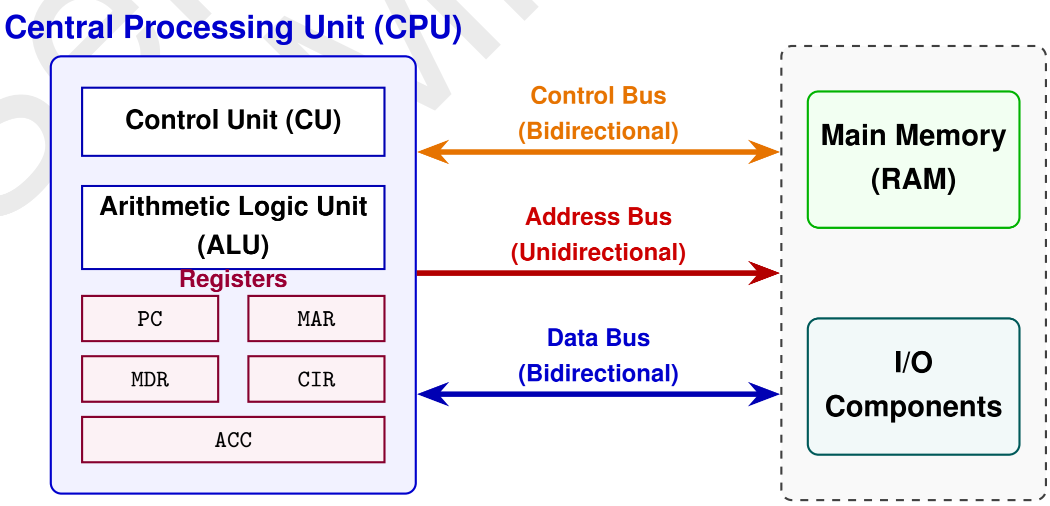
Learning Objectives

By the end of this topic you should be able to:

- Describe the role and components of the CPU under Von Neumann architecture.
- Name and state the purpose of each CPU register and bus.
- Trace and explain the Fetch–Decode–Execute (FDE) cycle step by step.
- Identify the three factors that affect CPU performance and explain their effect.
- Define instruction set and identify the two parts of a machine code instruction.
- Give examples and characteristics of embedded systems.
- Identify and describe input devices, output devices, and sensor types.
- Explain the sensor – ADC – microprocessor – DAC – actuator automated feedback loop.
- Distinguish between primary storage (RAM/ROM), secondary storage, virtual memory, and cloud storage.
- Compare magnetic, optical, and solid-state storage.
- Describe NIC, MAC address, IP address, static vs. dynamic IP, and router.

3.1 Computer Architecture

Role of the CPU and Microprocessor



The **Central Processing Unit (CPU)** is the "brain" of the computer: it executes instructions and processes data by taking input, performing processing, and producing output.

A **microprocessor** is a single integrated circuit (IC) containing a CPU. Microprocessors are commonly used in **embedded systems** (e.g. washing machines, cars).

Von Neumann Architecture

Von Neumann architecture organises the CPU into two main functional units:

- **Control Unit (CU):** Coordinates all CPU operations; decodes instructions and directs other components what to do.
- **Arithmetic Logic Unit (ALU):** Performs mathematical and logical operations (e.g. addition, subtraction, AND, OR).

Registers are small, extremely fast temporary storage locations inside the CPU:

Register	Purpose
Program Counter (PC)	Holds the memory address of the *next* instruction to be fetched.
Memory Address Register (MAR)	Stores the memory address currently being accessed.
Memory Data Register (MDR)	Holds data fetched from or to be written to memory.
Current Instruction Register (CIR)	Stores the instruction currently being decoded or executed.
Accumulator (ACC)	Stores intermediate arithmetic and logic results.

Error Trap — Register Identification

Examiners include distractor names such as "arithmetic register," "binary counter," "control register," or "instruction counter." Only the five registers above (PC, MAR, MDR, CIR, ACC) are valid Von Neumann registers on the 0478 syllabus. A recent mark scheme accepted **any one** of: accumulator, current instruction register, program counter.

Buses carry signals between the CPU, memory, and I/O components:

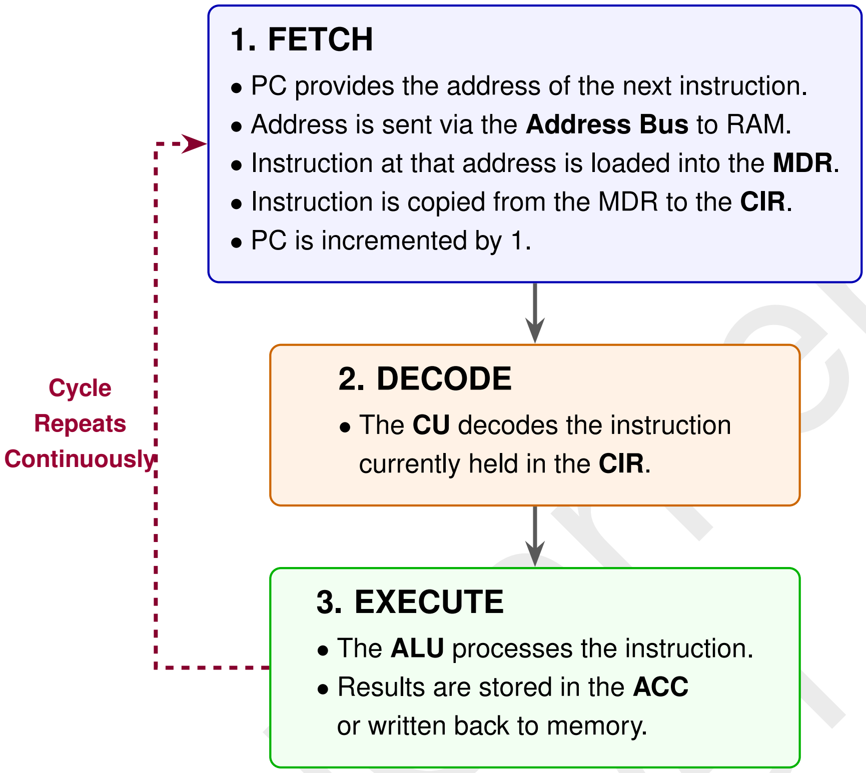
Bus	Function	Direction
Address Bus	Carries memory addresses from the CPU to RAM and I/O components.	Unidirectional
Data Bus	Carries data and instructions between the CPU and memory.	Bidirectional
Control Bus	Carries control and timing signals (e.g. read/write commands and clock pulses) to synchronise components.	Bidirectional

Common Mistake — Bus Direction

The address bus is **unidirectional** — it carries addresses *from* the CPU only. Both the data bus and control bus are **bidirectional**. Do not write "bidirectional" for all three buses.

Fetch–Decode–Execute Cycle

The CPU continuously repeats the FDE cycle to execute a program:



Exam Strategy — Describing the FDE Cycle

Mark schemes award one mark per correct step. Always state *which register* is involved and *which bus* carries the data. Model answer: "The address held in the PC is placed on the Address Bus and sent to RAM; the instruction is fetched into the MDR and copied to the CIR; the PC is incremented by 1; the CU decodes the instruction in the CIR; the ALU executes it and stores the result in the ACC."

Factors Affecting CPU Performance

Factor	Description	Effect on Performance
Cores	Independent processing units within the CPU.	More cores → more tasks executed in parallel.
Cache	Small, very fast memory located close to the CPU.	Larger cache → quicker access to frequently used data.
Clock Speed	Frequency of operations per second, measured in GHz.	Higher speed → more FDE cycles per second.

Exam Scenario — Comparing Three Processors

Three computers: **A** (2.1 GHz quad-core), **B** (2.5 GHz dual-core), **C** (5.2 GHz single-core).

*Which executes the most instructions **simultaneously**?*
Answer: Computer A — with 4 independent cores it can execute 4 instructions in parallel (the most of any option).

Note on clock speed vs cores: Computer C will complete a single task the fastest because its single core runs at a much higher frequency (5.2 GHz). Clock speeds across multiple cores cannot be added together mathematically (e.g. Computer B is a 2.5 GHz processor, not a 5.0 GHz processor).

Instruction Set and Machine Code

An **instruction set** is the complete list of all machine code commands a particular CPU can execute. All instructions are written in **machine code** (binary). Different CPUs may have different instruction sets.

Each machine code instruction has two distinct parts:

- **Opcode:** Specifies the type of operation to be carried out (e.g. Load, Store, Add, Jump).
- **Operand:** Contains the actual data item, or the memory address where the data resides.

Embedded Systems

An **embedded system** is a dedicated-purpose computer built into another device; it is not general-purpose. Common examples: washing machines, microwave ovens, cars, CCTV cameras, traffic lights, smart lighting systems.

Feature	General Purpose Computer	Embedded System
Function	Can perform multiple different tasks.	Performs one specific dedicated task.
Programmability	Flexible — user can install any software.	Fixed firmware; cannot be easily reprogrammed.
Example	Laptop, desktop PC.	Digital watch, ATM, washing machine.

Exam Scenario — ATM as an Embedded System

An ATM is an embedded system because:

- It performs a dedicated function (cash withdrawal, balance enquiry) — not general purpose.
- Its hardware is dedicated to ATM functions only.
- It cannot be easily reprogrammed.
- It uses firmware and contains a microprocessor rather than a general CPU.

3.2 Input and Output Devices

Input Devices

An **input device** is a hardware component that sends data into a computer system, allowing a user or the environment to interact with or provide instructions to the computer.

Device	Description	Typical Use
Keyboard	Manual text entry.	Typing documents, data entry.
Barcode Scanner	Reads barcodes using a laser or camera.	Retail checkouts, inventory control.
QR Code Scanner	Reads QR codes.	Mobile payments, linking to websites.
Digital Camera	Captures images digitally.	Photography, document scanning.
Microphone	Converts sound into digital signals.	Voice input, video calls.
Optical Mouse	Detects movement using LED light.	Pointing and clicking.
Touch Screen	Detects touch input from a finger or stylus.	Smartphones, tablets, kiosks.

Touch screen technologies:

Type	How It Works	Features
Resistive	Two conductive layers make contact when pressed.	Works with finger or stylus; cheaper to manufacture.
Capacitive	Detects the electrical charge from a human finger.	Works only with a bare finger; better display clarity.
Infra-red	A grid of IR sensors and light beams detect interruption.	Very durable; used on public-facing screens.

Output Devices

An **output device** is a hardware component that receives processed data from a computer and converts it into a human-perceivable form (visual, audio, or physical action).

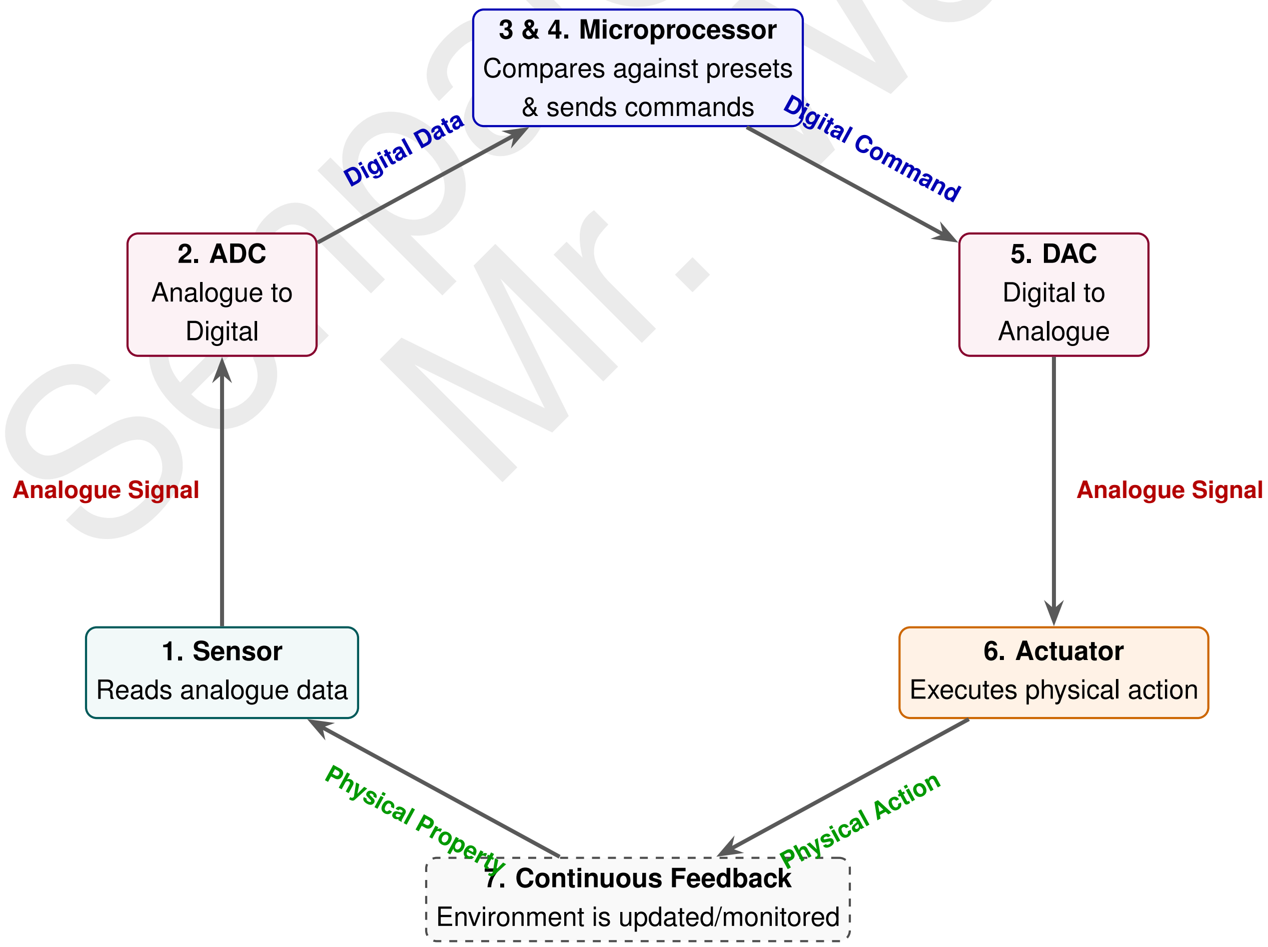
Device	Description	Output Type
Speaker	Converts digital audio signals to sound.	Sound.
LED Screen	Bright, energy-efficient display.	Visual.
LCD Screen	Common display used in monitors and laptops.	Visual.

Device	Description	Output Type
LCD Projector	Projects a computer display onto a large surface.	Visual (large scale).
DLP Projector	Uses mirrors and a colour wheel to project images.	High-quality visual.
Inkjet Printer	Sprays tiny ink droplets onto paper.	Physical document.
Laser Printer	Uses toner and heat for fast, high-volume printing.	Physical document.
3D Printer	Builds a physical object layer by layer from material.	Physical model.
Actuator	Converts an electrical signal into mechanical movement.	Physical action.

Sensors and the Automated Feedback Loop

A **sensor** is a hardware device that detects and measures a physical property (e.g. light, temperature, pressure) and sends digital data to a microprocessor. Sensors are used in automated/embedded systems to collect data from the environment without human intervention.

The sensor–microprocessor automated feedback loop:



Exam Strategy — Describing the Sensor Feedback Loop

Always name **ADC** when moving from sensor to microprocessor and **DAC** when moving from microprocessor to actuator — these are awarded as separate marking points. Also state that the microprocessor *compares* the sensor data against *pre-stored* values before making a decision.

Sensor types and uses:

Sensor	Detects	Example Use
Acoustic	Sound levels.	Noise monitoring, voice detection.
Accelerometer	Motion or tilt.	Smartphones, gaming, vehicle systems.
Flow	Movement of liquids or gases.	Water pipelines, HVAC systems.
Gas	Gas concentration (e.g. CO_2, methane).	Industrial safety, air quality monitoring.
Humidity	Moisture content in air.	Weather stations, greenhouses.
Infra-red	Heat or infrared radiation.	Remote controls, motion sensors.
Level	Fluid level.	Fuel tanks, water reservoirs.
Light	Ambient light level.	Automatic screen brightness adjustment.
Magnetic Field	Magnetic fields.	Compass apps, metal detection.
Moisture	Soil or water content.	Agriculture, automated gardening.
pH	Acidity or alkalinity.	Chemical labs, water treatment.
Pressure	Force or pressure.	Car tyres, industrial automation.
Proximity	Nearby objects.	Phones, automatic doors, ATM detection.
Temperature	Heat level.	Thermostats, computer cooling fans.

3.3 Data Storage

Primary Storage

Primary storage is memory the CPU can access *directly and very quickly*. It stores data and programs currently in use and is typically volatile.

Type	Description	Volatility	Role
RAM (Random Access Memory)	Stores data and instructions for programs currently being executed; can be read from and written to.	Volatile	Temporary workspace for the CPU during processing.
ROM (Read Only Memory)	Stores permanent firmware/BIOS needed to start the computer; cannot be easily altered.	Non-volatile	Provides boot instructions to load the OS into RAM.

Common Mistake — RAM vs ROM Volatility

RAM is **volatile** — its contents are lost when power is removed. ROM is **non-volatile** — it retains its data without power. RAM is used for programs currently running; ROM stores the firmware needed to boot the computer.

Secondary Storage

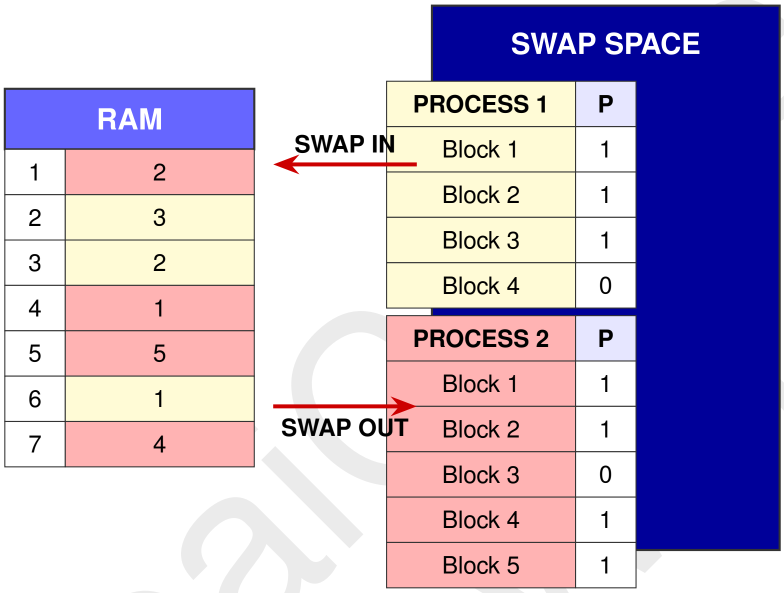
Secondary storage is memory that is *not* directly accessed by the CPU. It provides **long-term, non-volatile** storage of files, documents, software, and the operating system. Data persists when the computer is turned off.

Type	How It Works	Examples	Advantages	Disadvantages
Magnetic	Spinning platters divided into tracks and sectors; electromagnetic heads read/write data.	HDD	Cheap; high capacity.	Slower; mechanical parts can fail.
Optical	A laser reads/writes data by detecting pits and lands on the disc surface.	CD, DVD, Blu-ray	Portable; low cost.	Slower access; lower capacity.
Solid-State	NAND/NOR flash memory stores data in floating-gate transistors — no moving parts.	SSD, USB, SD card	Fast; durable.	More expensive per GB.

Virtual Memory

Virtual memory is a section of the hard drive that the OS uses as an extension of RAM when physical RAM is full.

- When RAM fills up, the OS moves *inactive* data to a **paging file** (swap space) on the hard disk.
- When that data is needed again it is moved back into RAM.
- Prevents system crashes when RAM runs out.
- **Trade-off:** Hard disks are slower than RAM, so performance may drop noticeably.



Cloud Storage

Cloud storage allows users to store data on remote servers accessed via the internet, rather than on a local device. Examples: Google Drive, Dropbox, OneDrive. Files are uploaded to a remote data centre and can be accessed from any internet-connected device.

Advantages	**Disadvantages**
Accessible from anywhere with an internet connection.	Cannot access files without internet.
Good for backups and disaster recovery.	Data may be less secure or private.
Frees up local disk space.	Ongoing subscription costs possible.
Easy to share files and collaborate.	Dependent on the cloud provider's reliability.

3.4 Network Hardware

Network Interface Card

A **Network Interface Card (NIC)** is a hardware component inside a computer that enables it to connect to a network. It may support wired (Ethernet) or wireless (Wi-Fi) communication. Every NIC is assigned a unique MAC address; without a NIC a device cannot access or communicate on a network.

Purposes of a NIC:

- Transmit and receive data to and from the network or other devices.
- Allow a physical connection to the network.
- Convert network data so it can be understood by the computer, and vice versa.
- Enable a device to be assigned an IP address.

Media Access Control Address

A **MAC address** is a unique identifier assigned to a NIC by the manufacturer, used to identify devices within a LAN.

- Written in hexadecimal, e.g. `00:1A:C2:7B:00:47`.
- Six groups of two-digit hexadecimal numbers — 48 bits (12 hex digits) in total.
- First 3 pairs: Manufacturer ID; last 3 pairs: unique device serial number.
- **Static** — cannot be changed.

Exam Strategy — MAC Address Marking Points

A 3-mark MAC address question typically awards one mark each for: *written in hexadecimal*; *six groups separated by colons or hyphens*; *48 bits/12 hex digits total*; *contains manufacturer ID and device serial number*; *static — cannot be changed*. Learn all five points.

Internet Protocol Address

An **IP address** identifies a device on a network, acting like a mailing address for sending and receiving data.

Static vs. Dynamic IP:

Type	Description
Static IP	Fixed address that does not change; usually assigned manually. Useful for servers that must always be reachable at the same address.
Dynamic IP	Changes each time the device connects to the network; assigned automatically by a router using DHCP. Common for home users.

IPv4 vs. IPv6:

Feature	IPv4	IPv6
Length	32 bits.	128 bits.
Format	4 decimal numbers, e.g. `192.168.1.1`.	Hexadecimal groups, e.g. `2001:0db8:85a3::7334`.
Address Limit	≈4 billion addresses.	Trillions — designed for future internet growth.

Common Mistake — Valid IPv4 Format

A valid IPv4 address has exactly **4** decimal groups separated by full stops, with each group between 0 and 255. Watch for: more than 4 groups, hexadecimal digits in the address, or any group exceeding 255. Example: `3.162.74.3` is valid; `8.0.257.6.8` is not.

Router

A **router** connects different networks together — for example, a home network to the internet.

- Routes incoming and outgoing data between local devices and the internet using IP addresses to identify the correct destination.
- Assigns IP addresses to devices on the local network using **DHCP** (Dynamic Host Configuration Protocol).
- Can provide firewall protection and manage network traffic flow.

Switches and Hubs

A **switch** is an intelligent hardware device that provides a central connection point for all devices within a Local Area Network (LAN).

- It stores the **MAC addresses** of all connected devices in a routing table.
- When it receives a data packet, it examines the destination MAC address and sends the data *only* to the correct specific port/device.
- This targeted transmission reduces network traffic, improves security, and prevents data collisions.

A **hub** is an older, less intelligent alternative to a switch.

- When a hub receives a data packet, it **broadcasts** (copies) it to *all* connected devices on the network.
- This creates unnecessary network traffic, increases data collisions, and poses a security risk because every device receives every data packet.

Common Mistake — Router vs Switch

Do not confuse a router with a switch. A **switch** connects devices *within* the same network using **MAC addresses**. A **router** connects *multiple networks* together (e.g. a LAN to the internet) using **IP addresses**.

Wireless Access Point (WAP)

A **Wireless Access Point (WAP)** is a piece of hardware that allows wireless devices to connect to a wired network using Wi-Fi.

- It broadcasts and receives radio frequencies to communicate with wireless NICs built into devices like laptops and smartphones.
- It is often built into home routers, but can also be installed as a standalone device in large buildings to extend the wireless range of the LAN.