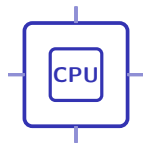


Hardware

IGCSE Computer Science ■ Topic 3

IGCSE Computer Science



What we will cover

- 1 The CPU & buses
- 2 The fetch–execute cycle
- 3 Input & output devices
- 4 RAM, ROM & storage
- 5 Virtual memory
- 6 Network hardware

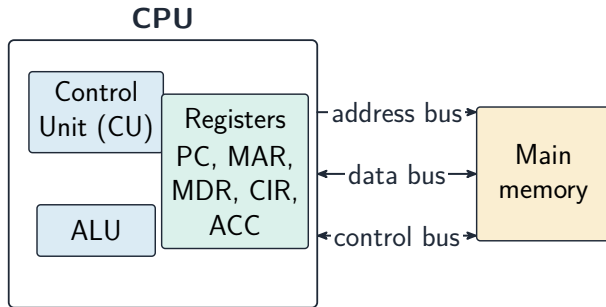




Inside the CPU

Von Neumann : **one** memory holds data *and* instructions, reached over shared buses.

- **ALU** – arithmetic and logic
- **control unit** – decodes and directs
- registers, stepped by the clock



Three buses link the CPU to main memory

Inside the CPU: the units and the three buses

control unit

decodes each instruction and sends the signals that carry it out

arithmetic logic unit

the ALU – does calculations (add, subtract) and **logical** operations (compare)

address bus

carries memory addresses – **one direction only**, CPU to memory

data bus

carries data and instructions – **both** directions

control bus

carries control signals, such as read and write

The width of the **address** bus fixes how much memory can be addressed; the width of the **data** bus fixes how much moves per transfer. That asymmetry is why only one is bidirectional.

The special registers

PC – address of
the *next* instruction

MAR – the address to read/write

MDR – the data just fetched

CIR – the instruc-
tion now running

ACC – the ALU's result

The registers, and what each holds

MAR

The **memory address register** holds the *address* being read from or written to.

MDR

The **memory data register** holds the *data* that just came from, or is going to, that address.

CIR

The **current instruction register** holds the instruction being decoded and executed right now.

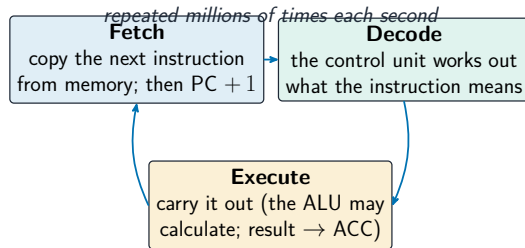
PC

The program counter holds the address of the *next* instruction – which is what a jump changes.

Address, data, instruction, next: four **registers** inside the **central processing unit** , and the **fetch–execute cycle** is just them passing values to each other.

The fetch–execute cycle

- **Fetch:** PC $\rightarrow$ MAR, memory $\rightarrow$ MDR, copy to CIR, PC +1
- **Decode** the opcode
- **Execute** (ALU, load, store, jump)
– then repeat



The processor does nothing else, billions of times a second

What makes a CPU faster

cores

each runs instructions on its own – more do more at once

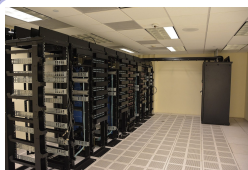
cache size

more data kept ready, so the CPU waits less

clock speed

cycles per second
– more instructions per second

Faster CPU: more **cores**, more **cache**, higher **clock speed**. An **embedded system** is a small computer built into a device to do **one** fixed job.



Cloud servers

Worked example – which computer is faster?

A: 3.0 GHz single-core, 2 MB cache. B: 2.4 GHz quad-core, 8 MB cache. Which is faster?

- it depends on the task
- one program that cannot be split: A wins on **clock speed**
- several programs, or one written for many **cores** : B wins – each core runs instructions on its own
- and B's larger **cache** means fewer slow trips to RAM

“Higher GHz is faster” is only true **holding cores and cache equal**. The exam sets this comparison precisely to see whether you say so.



Input devices



keyboard



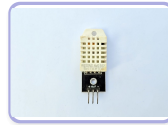
mouse



touch screen



microphone



sensor



barcode scanner

An **input** device sends data *into* the computer. A **sensor** is the one that needs no person: it measures a physical quantity and sends it as data.

The three kinds of touch screen

touch screen

detects **where** your finger touches the screen

resistive

two layers are **pressed together** where you touch – cheap, works with a glove or stylus, but less durable and no multi-touch

capacitive

senses the tiny **electric charge** of your finger – bright, durable, multi-touch, but bare skin only

infrared

your finger **breaks a grid** of light beams – works with any object and at large sizes, but sunlight interferes

Match the type to the use: capacitive for a phone, resistive for a factory where operators wear gloves, infrared for a large interactive whiteboard.

A digital camera: a lens focuses light



A flatbed scanner: it shines light across a page to copy it into the computer

A digital camera: a lens focuses light onto a sensor that records the pixels

Output devices



monitor



laser printer



inkjet printer



3D printer



speaker



actuator

An **output** device sends data *out*. An **actuator** turns the signal into **movement** – which is how a computer acts on the world.

The output devices, by how each makes its image

LCD/LED screen

shows images using a grid of tiny coloured dots

projector

shines an **enlarged** image onto a wall or screen

inkjet printer

sprays tiny drops of ink onto paper – cheap to buy, good photos, slow for long runs

laser printer

a laser and powder (**toner**) fixed onto the paper by **heat** – fast and cheap per page in bulk

3D printer

builds a **solid object** layer by layer

speaker and actuator

turn a signal into sound, or into movement in a control system

Choose the printer by **volume**: inkjet for a few high-quality pages, laser for many. That is the comparison the exam asks for.

A pair of blue over-ear headphones



Headphones turn an electrical signal into sound for one listener

RAM vs ROM

	RAM	ROM
access	read & write	read only
power off	loses data (volatile)	keeps data (non-volatile)
holds	programs in use	start-up code

RAM is volatile working memory;
ROM keeps the start-up code with
the power off.

RAM, ROM, and the two kinds of RAM

RAM

Volatile: it loses its contents when the power goes. It holds what-ever is running now.

Read only memory

Read only memory (ROM) is non-volatile and holds the start-up instructions – the BIOS – that the machine needs before any disk is read.

Dynamic RAM

Dynamic RAM stores each bit in a capacitor that leaks, so it must be refreshed thousands of times a second. Cheap and dense – this is main memory.

Static RAM

Static RAM holds each bit in a flip-flop, needs no refresh and is much faster – and much dearer, so it is used for cache.

Volatile or not is the first question; refreshed or not is what separates the cache from the main memory.

Primary storage: RAM and ROM

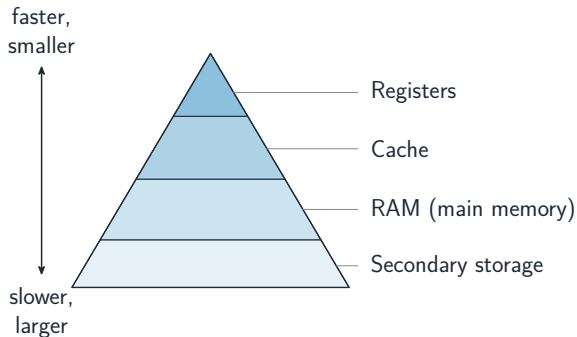
RAM	ROM
volatile can change running programs	non-volatile cannot change startup (BIOS)



A stick of RAM (random access memory): it holds the programs and data in use now

RAM is volatile and holds running programs; ROM is non-volatile and read-only

The memory hierarchy



- Registers and **cache** : fastest, smallest, closest to the ALU.
- **RAM** : the programs and data in use now.
- Secondary storage: slowest but by far the largest, and **non-volatile**.

Faster memory costs more per byte, so a computer has a little of the fast kind and a lot of the slow kind.

Secondary storage

magnetic

magnetised spots
on spinning disks
HDD, tape

optical

marks read by a laser
CD, DVD, Blu-ray

solid state

flash memory,
no moving parts
SSD, USB drive

Secondary storage is **non-volatile** – it keeps data with the power off, unlike RAM.



*Inside an HDD: the disk spins, the arm
reads*

Storage media, in the flesh



hard disk
(magnetic)



tape (magnetic)



CD



SSD (solid state)



USB drive
(solid state)

Magnetic and optical media have **moving parts**; solid state has none, so it is faster and survives being dropped.

Three kinds of secondary storage

Magnetic storage

Magnetic storage : a spinning platter magnetised in tiny regions. Cheap per gigabyte, large, and it has moving parts that fail.

Optical storage

Optical storage : pits burned into a disc surface and read by a laser – CD, DVD, Blu-ray. Cheap and slow.

Solid state storage

Solid state storage : no moving parts, flash memory holding charge in cells. Fast, silent, robust, dearer per gigabyte.

Cloud storage

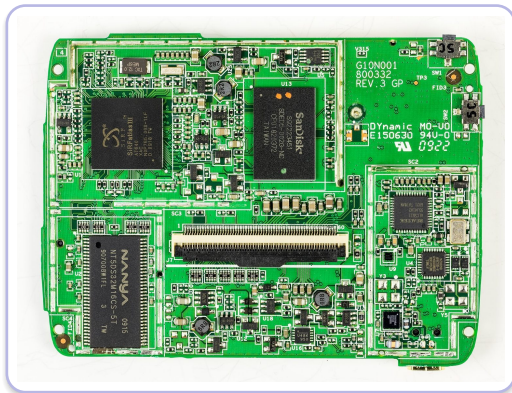
Cloud storage is someone else's secondary storage, reached over a network – so it needs a connection, and it raises the security question.

Compare on the same four axes: capacity, speed, durability and cost per gigabyte. That is what the exam question is really asking for.

Embedded systems

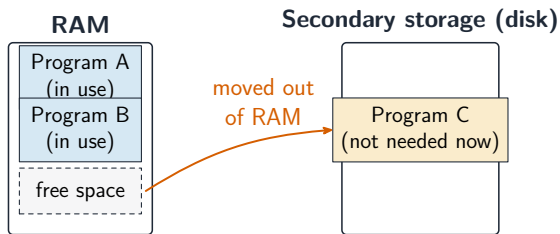
An **embedded system** is a small computer built *inside* a larger device to do *one fixed job*.

- dedicated to that task, small and low-cost
- no general operating system
- a washing machine, a microwave, a set-top box



The mainboard from inside a satnav

Virtual memory



This frees RAM, so more programs can run than RAM alone could hold — but it is slower, because the disk is much slower than RAM.

- **Advantages:** reach it from anywhere on any device; no local storage hardware to buy or maintain; easy to share files and to back up automatically.
- **Disadvantages:** needs a working internet connection; an ongoing subscription cost; security and privacy depend on the provider; you rely on that provider staying available and in business.

Virtual memory, and the two kinds of address

virtual memory

when RAM fills, the computer uses part of **secondary storage** as extra, pretend RAM

the cost

secondary storage is far slower, so heavy use makes the machine crawl

MAC address MAC

a **permanent** hardware address, set by the manufacturer, unique to the network card

IP address IP

assigned by the network, and it **changes** when the device moves

IPv4

32-bit, four denary numbers 0–255, for example 192.168.0.1

IPv6

128-bit, eight groups of hexadecimal – far more addresses, because IPv4 ran out

MAC identifies the **device**; IP identifies **where it is on the network**. A letter needs both the person and the address.



Network hardware

- **NIC** : the card that joins a device to a network
- **MAC address** : identifies one physical device – *fixed*
- **IP address** : identifies a device on a network – *depends on the network*
- **router** : connects networks (home → internet)



A Wi-Fi router forwards packets between networks

Two more devices, and where they sit

Network interface card

A **network interface card** (NIC) gives a device its physical connection and its MAC address – wired or wireless.

Sensors

Sensors are input devices that read a physical quantity continuously – temperature, light, pressure, motion – and feed an embedded system.

QR code scanner

A **QR code scanner** photographs the pattern and decodes the data stored in it – a camera plus software, not a new kind of hardware.

Sort any device by what it does to data: brings it in, sends it out, stores it, or moves it between machines.

Exam tips

- Learn the **fetch–decode–execute** cycle and the special registers: the **program counter** holds the address of the *next* instruction, the MAR/MDR carry the address/data, and the **accumulator** holds the ALU's result.
- CPU performance improves with more **cores**, a larger **cache**, and a higher **clock speed**.
- **RAM** is volatile (it loses its data with no power) and holds programs in use; **ROM** is non-volatile and read-only (the start-up instructions).
- Match the secondary-storage types: **magnetic** (HDD, tape), **optical** (CD/DVD, read by a laser), **solid state** (SSD, USB, flash – no moving parts).
- A **MAC address** identifies the physical device and does not normally change; an **IP address** is given by the network and can change.



Topic 3 – key takeaways

1 CPU

ALU + CU + registers, joined by buses

2 Cycle

fetch → decode → execute, PC +1

3 Memory

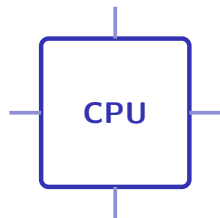
RAM volatile, ROM non-volatile; disk is secondary

4 Network

MAC fixed to device; IP set by the network

Check yourself

- 1 Which register holds the next instruction's address?
- 2 Is RAM volatile or non-volatile?
- 3 Which memory holds the start-up instructions?
- 4 Which address is fixed to the physical device?



Answers 1. the PC 2. volatile 3. ROM 4. the MAC address