A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

3.1 Computers and their components

Name: _____ Class: _____ Date: _____ **Total: 42 marks**

KEY WORDS sensor 传感器 • buffer 缓冲区 • sram 静态 RAM • control system 控制系统 • embedded system 嵌入式系统

Objective

Build the skills to answer exam questions on **computer components** 计算机组件 — input, output, memory and storage; the types of RAM and ROM; embedded systems; and monitoring and control systems.

You must be able to:

- explain the need for **input**, **output**, **primary memory** 主存储器 and **secondary storage** 辅助存储器
- explain the differences between **RAM** and **ROM**, **SRAM** and **DRAM**, and the types of ROM
- describe **embedded systems** 嵌入式系统 and their benefits and drawbacks
- describe **monitoring and control** systems using **sensors** 传感器, **actuators** 执行器 and **feedback** 反馈

1 Worked examples

Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

The four building blocks

Block	Job	Examples
input	get data in	keyboard, mouse, sensor
output	give results out	monitor, speaker, actuator
primary memory	fast store the CPU uses directly	RAM, ROM
secondary storage	large, keeps data when off	hard disk, SSD, USB

RAM, ROM and their types

RAM holds the running program and data and is **volatile** (lost when power is off). **ROM** holds fixed start-up data and is **non-volatile** (kept).

	RAM	ROM
Keeps data without power?	no (volatile)	yes
Can the CPU write to it?	yes	normally no
Used for	running programs	start-up / firmware

- **SRAM vs DRAM:** SRAM is faster and needs no refreshing, so it is used for **cache**; DRAM is cheaper and denser, so it is used for **main memory** (but must be refreshed).
- **ROM types:** **PROM** is written once; **EPROM** is erased by UV light and rewritten; **EEPROM** is erased and rewritten **electrically**, in place.

Processor performance

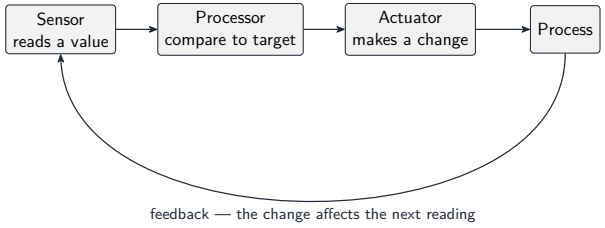
Three things make a CPU run programs faster:

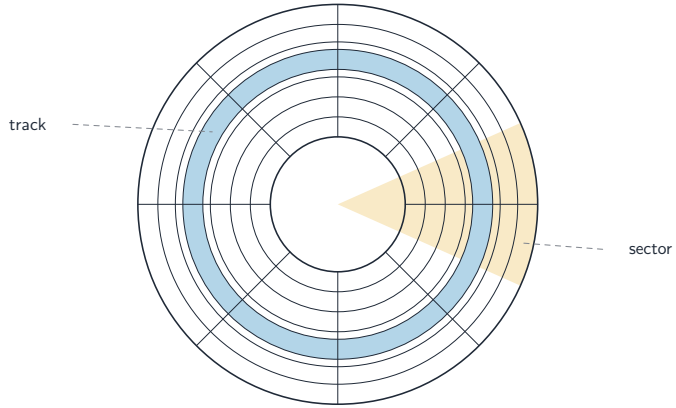
- **clock speed** 时钟速度 — more cycles per second means more instructions per second.
- **number of cores** 核心 — each core can run instructions, so more cores do more work **at the same time** — *but only if the software is written to use them*, and more cores cost more and make more heat.
- **cache** 缓存 — a small, very fast memory near the CPU; more cache means fewer slow trips to main memory.

A **buffer** 缓冲区 is a small temporary store that holds data while a slow device (such as a printer) catches up, so the CPU does not have to wait for it.

Monitoring and control

In **monitoring** the system only reports readings; in **control** it also acts to change the process. A control system uses **feedback** — the change it makes affects the next reading:





A hard disk drive stores data in tracks and sectors

An **embedded system** is a computer built into another device for one fixed job (washing machine, car engine unit). It is small, cheap and reliable, but limited to that task and hard to update.

2 Practice

Now apply the methods above.

2.1 State the difference between **primary memory** and **secondary storage**. [2]

2.2 State **two** differences between **RAM** and **ROM**. [2]

2.3 Give one **benefit** and one **drawback** of an embedded system. [2]

2.4 Name the component in a control system that **makes a physical change** to the process. [1]

3 Exam-style questions

3.1 (a) Explain the difference between RAM and ROM. [2]

(b) Give one use of ROM in a computer. [1]

3.2 Explain why **SRAM** is used for cache memory while **DRAM** is used for main memory. [3]

3.3 State the difference between **PROM**, **EPROM** and **EEPROM**. [3]

3.4 A greenhouse control system keeps the temperature near 25 °C.

(a) Name a suitable **sensor** for this system. [1]

(b) Describe how the system uses **feedback** to keep the temperature steady. [3]

3.5 Describe the principal operation of **either** a laser printer **or** a 3D printer. [3]

3.6 A computer is upgraded from a single-core to a quad-core processor.

(a) Explain how having more **cores** can improve performance. [2]

(b) State **one** reason the performance may **not** improve as much as expected. [1]

3.7 A computer sends documents to a printer, which prints much more slowly than the computer can send. State why the printer needs a **buffer**. [2]

3.8 A greenhouse uses a computer system to look after tomato plants. Sensors measure temperature, humidity and soil moisture. A separate 3D printer in the same building makes replacement clips for the plant supports.

(a) **Describe** the difference between a **monitoring** system and a **control** system, using this greenhouse as your example. [3]

(b) The greenhouse system opens a vent when the temperature exceeds 28 °C. **Explain** the role of the **actuator** and of the **ADC** in this system. [3]

(c) **Explain** why the system takes readings **continuously** in a loop rather than once an hour. [2]

(d) **Explain** how the 3D printer produces a clip from a design file. [3]

(e) **State** why the 3D printer needs a **buffer**, and **explain** what would happen without one. [3]

Solutions

Each answer shows the steps, then the **result**.

2.1

- primary memory is fast and reached **directly** by the CPU (holds the running program)
- secondary storage is larger, slower, and **keeps data when the power is off**

2.2

- any two of: RAM is volatile / ROM is non-volatile; the CPU can write to RAM / normally not to ROM; RAM holds running programs / ROM holds start-up data

2.3

- benefit e.g. small / cheap / low-power / reliable
- drawback e.g. limited to one task / hard to update / hard to repair

2.4

- **actuator**

3.1

(a) RAM is **volatile** and the CPU can read **and write** it (running programs)

- ROM is **non-volatile** and normally only read

(b) e.g. it stores the **boot/start-up program** (BIOS)

3.2

- SRAM is **faster** and needs no refreshing, so it suits the small, speed-critical cache
- DRAM is **cheaper and denser** per bit, so it suits the large main memory
- even though DRAM must be refreshed

3.3

- **PROM** can be written **once**
- **EPROM** can be erased (by UV light) and rewritten
- **EEPROM** can be erased and rewritten **electrically**, in place

3.4

(a) a **temperature sensor**

(b) the sensor reads the temperature

- the processor compares it with 25 °C and switches a heater/cooler (actuator) on or off
- this changes the temperature, which the sensor reads again — the feedback keeps it near the target

3.5

- **laser printer:** a laser draws the page image as charge on a photosensitive drum
- toner sticks to the charged areas and transfers to the paper
- a fuser melts the toner onto the paper
- *(or a correct layer-by-layer description of a 3D printer)*

3.6

(a) each core can execute instructions, so a quad-core CPU can run up to four tasks (or threads) at the same time

- that means more work per second

(b) the software (or task) must be written to split across cores; a single-threaded program still uses only one core

3.7

- the printer is much slower than the computer; the buffer holds the data waiting to be printed
- the computer (CPU) is freed to do other work instead of waiting for the printer

3.8

(a) a **monitoring** system reads the sensors and reports or records the values — it displays the greenhouse temperature and logs it, but changes nothing

- a **control** system reads the same sensors and **acts** on the result, opening the vent or switching on the irrigation
- so a control system contains a monitoring system plus the output devices that change the conditions

(b) the **ADC** converts the sensor's continuously varying analogue voltage into the binary values the processor can work with

- the **actuator** is the device that carries out the processor's decision — here the motor that drives the vent open

(c) conditions change quickly in a greenhouse, and a plant can be damaged within minutes of overheating

- reading in a continuous loop means the system responds as soon as the threshold is crossed rather than up to an hour later

(d) the design file is sliced into many thin horizontal **layers**

- the printer deposits or solidifies material one layer at a time
- each layer bonds to the one beneath, building the clip from the bottom up

(e) the printer works far more slowly than the computer can send data

- the buffer stores the data waiting to be printed
- without it the computer would have to wait for the printer, unable to do anything else, and data arriving faster than it was consumed would be lost