

3.1 Computer architecture

Name: _____ Class: _____ Date: _____ **Total: 28 marks**

KEY WORDS clock speed 时钟速度 • instructions 指令 • embedded system 嵌入式系统
• current instruction register 当前指令寄存器 • cache 高速缓存

Objective

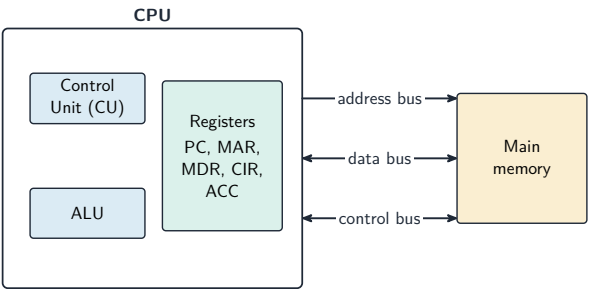
Build the skills to answer exam questions on **computer architecture** 计算机体系结构 — the CPU parts, the **fetch–execute cycle** 取指执行周期, **cache** 高速缓存, and CPU performance.

You must be able to:

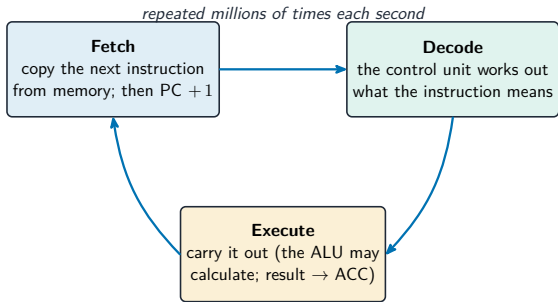
- name the CPU parts (ALU, CU, registers) and the buses
- describe the fetch–decode–execute cycle
- explain what affects CPU performance and what an embedded system is

1 Worked examples

■ The CPU and the cycle



The CPU holds the CU, ALU and registers; the address, data and control buses link it to memory



The CPU repeats: fetch the instruction, decode it, then execute it

- performance ↑ with more **cores**, a larger **cache**, and a higher **clock speed**.

■ Naming the parts, and describing the cycle in order

The CPU parts and what each one does. The **control unit (CU)** 控制单元 decodes instructions and sends control signals; the **arithmetic logic unit (ALU)** 算术逻辑单元 does the calculations and logic; **registers** 寄存器 hold data during processing. The named registers you must know:

Register	What it holds
program counter (PC)	the address of the next instruction
memory address register (MAR)	the address currently being read from or written to
memory data register (MDR)	the data or instruction just fetched, or about to be written
current instruction register (CIR)	the instruction being decoded and executed
accumulator (ACC)	the result of the ALU's work

The three buses: the **address bus** 地址总线 carries addresses (one direction only, CPU to memory); the **data bus** 数据总线 carries data and instructions (both directions); the **control bus** 控制总线 carries control signals such as read and write (both directions).

The fetch–decode–execute cycle, in the order the marks are awarded:

1. The address in the **PC** is copied to the **MAR**.
2. The PC is **incremented** by one.
3. The instruction at that address is fetched along the **data bus** into the **MDR**.
4. The instruction is copied from the MDR to the **CIR**.
5. The CU **decodes** the instruction.
6. The instruction is **executed**, and any result is stored in the **ACC**.

Performance. More **cores** let the CPU work on more instructions at once; a larger **cache** 高速缓存 keeps frequently used instructions closer to the CPU, so fewer slow fetches from RAM are needed; a higher **clock speed** 时钟速度 means more cycles each second. An **embedded system** 嵌入式系统 is a computer built into a larger device to do one dedicated job, such as the controller in a washing machine or a set of traffic lights.

2 Practice

2.1 State the function of the ALU. [1]

2.2 Name the three stages of the fetch–execute cycle. [3]

2.3 Name the register that holds the address of the next instruction. [1]

2.4 Identify the bus that carries data in both directions, and the bus that carries addresses in one direction only. [2]

2.5 Explain how a larger cache improves the performance of a computer. [2]

3 Exam-style questions

3.1 Cache memory speeds up the CPU because it: [1]

- | | |
|---|---|
| A | B |
| C | D |
- A stores data permanently
 - B is very fast memory holding frequently used data
 - C has no power needs
 - D is part of secondary storage

3.2 A computer is upgraded.

(a) State two changes that would improve CPU performance. [2]

(b) State what is meant by an embedded system, with one example. [2]

3.3 A computer has a Von Neumann architecture with a single processor.

(a) Complete the table by giving the full name of each register and what it stores. [4]

Abbreviation	Full name	What it stores
PC		
MAR		
MDR		
ACC		

(b) Describe the fetch stage of the fetch–execute cycle, using the register names from part (a). [4]

(c) Identify the bus that would be used to send a “write” signal to memory, and explain why this bus is bidirectional. [2]

(d) The computer is upgraded from two cores to four. Explain why this does not always make a program run twice as fast. [2]

(e) Give one example of an embedded system and describe what makes it different from a general-purpose computer. [2]
