Exercise walk-through

Computer architecture ▪ 3.1

eXercise

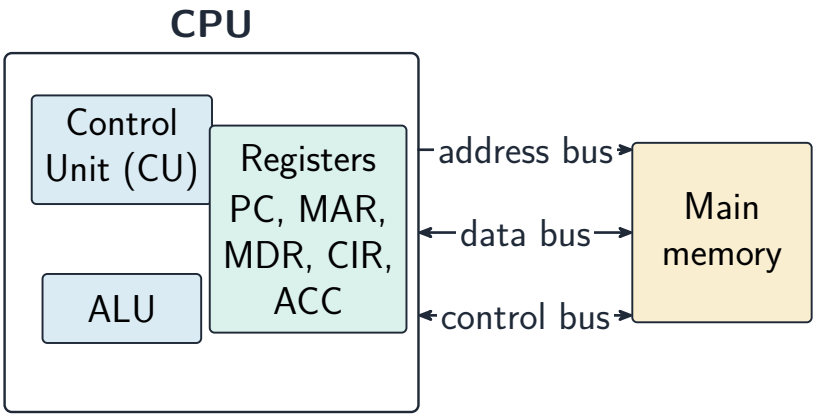
You must be able to

- 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

Method — The CPU and the cycle

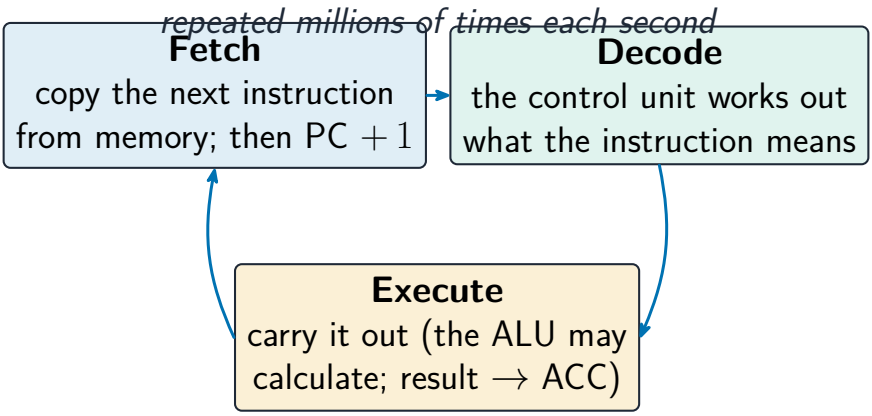
- performance ↑ with more **cores**, a larger **cache**, and a higher **clock speed**.

Method — The CPU and the cycle



A CPU block diagram with CU, ALU, registers and buses

Method — The CPU and the cycle



Fetch → decode → execute loop

Method — 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

Method — Naming the parts, and describing the cycle in order

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**.

Method — Naming the parts, and describing the cycle in order

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.

Practice 2.1 ▪ 1 mark

State the function of the ALU.

Solution 2.1

Worked steps

it carries out calculations and logical operations B1.

Practice 2.2 ▪ 3 marks

Name the three stages of the fetch–execute cycle.

Solution 2.2

Method: Naming the parts, and describing the cycle in order

Worked steps

fetch; decode; execute B1 B1 B1.

Practice 2.3 ▪ 1 mark

Name the register that holds the address of the next instruction.

Solution 2.3

Method: Naming the parts, and describing the cycle in order

Worked steps

The program counter (PC) B1.

Practice 2.4 ▪ 2 marks

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

Solution 2.4

Method: Naming the parts, and describing the cycle in order

Worked steps

Data bus, both directions B1; address bus, one direction only, from CPU to memory B1.

Practice 2.5 ▪ 2 marks

Explain how a larger cache improves the performance of a computer.

Solution 2.5

Worked steps

Cache is very fast memory that holds frequently used instructions and data close to the CPU B1; the CPU then makes fewer slow fetches from RAM, so it spends less time waiting B1.

Exam-style 3.1 ▪ 1 mark

Cache memory speeds up the CPU because it:

A stores data permanently

B is very fast memory holding frequently used data

C has no power needs

D is part of secondary storage

Solution 3.1

A stores data permanently

B is very fast memory holding frequently used data

C has no power needs

D is part of secondary storage

Worked steps

B. Cache is fast memory holding frequently used data.

Exam-style 3.2 ▪ 2 marks

A computer is upgraded.
(a) State two changes that would improve CPU performance.
(b) State what is meant by an embedded system, with one example.

Solution 3.2

Method: Naming the parts, and describing the cycle in order

Worked steps

(a) any two of: more cores; larger cache; higher clock speed B1 B1.
(b) a small computer built into a device to do one fixed job M1; e.g. a washing machine or microwave A1.

Exam-style 3.3 ▪ 4 marks

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.

Abbreviation	Full name	What it stores
PC		
MAR		
MDR		
ACC		

Exam-style 3.3 ▪ 4 marks

(b) Describe the fetch stage of the fetch–execute cycle, using the register names from part (a).
(c) Identify the bus that would be used to send a “write” signal to memory, and explain why this bus is bidirectional.
(d) The computer is upgraded from two cores to four. Explain why this does not always make a program run twice as fast.
(e) Give one example of an embedded system and describe what makes it different from a general-purpose computer.

Solution 3.3

Method: Naming the parts, and describing the cycle in order

Worked steps

(a) PC: program counter, holds the address of the next instruction B1. MAR: memory address register, holds the address being read from or written to B1. MDR: memory data register, holds the data or instruction fetched from that address B1. ACC: accumulator, holds the result of a calculation B1.
(b) The address in the PC is copied to the MAR B1; the PC is incremented B1; the instruction at that address is fetched into the MDR along the data bus B1; it is copied from the MDR into the CIR ready for decoding B1.
(c) The control bus B1; it must carry signals both ways, because the CPU sends

Solution 3.3

commands such as read and write while devices send back signals such as an interrupt request B1.
(d) Not all programs can be split into parts that run at the same time B1, and cores must still share memory and wait for each other, so the extra cores are not always fully used B1.
(e) Any one of: a washing machine controller, traffic lights, a microwave, a car engine management unit B1. It is built into a larger device and runs one dedicated program that the user cannot change, rather than running many different applications B1.