

# 4 Processor Fundamentals – Part 1

Name: \_\_\_\_\_ Class: \_\_\_\_\_ Date: \_\_\_\_\_

## Vocabulary 词汇

Write each word three times. 把每个单词抄写三遍。

| English                      | 中文       | Write it 3 times (English) |       |       |
|------------------------------|----------|----------------------------|-------|-------|
| processor                    | 处理器      | _____                      | _____ | _____ |
| stored program               | 存储程序     | _____                      | _____ | _____ |
| arithmetic and logic unit    | 算术逻辑单元   | _____                      | _____ | _____ |
| control unit                 | 控制单元     | _____                      | _____ | _____ |
| register                     | 寄存器      | _____                      | _____ | _____ |
| Program Counter              | 程序计数器    | _____                      | _____ | _____ |
| Accumulator                  | 累加器      | _____                      | _____ | _____ |
| memory address register      | 存储器地址寄存器 | _____                      | _____ | _____ |
| memory data register         | 存储器数据寄存器 | _____                      | _____ | _____ |
| current instruction register | 当前指令寄存器  | _____                      | _____ | _____ |
| fetch-execute cycle          | 取指执行周期   | _____                      | _____ | _____ |
| data bus                     | 数据总线     | _____                      | _____ | _____ |
| address bus                  | 地址总线     | _____                      | _____ | _____ |

## Recall

At IGCSE you know the CPU runs programs. Bring this with you:

- The **processor** 处理器 fetches and carries out instructions.
- Programs and data are stored in memory.

At AS we look inside the processor and at how it runs a single instruction — the cycle every computer repeats billions of times a second.

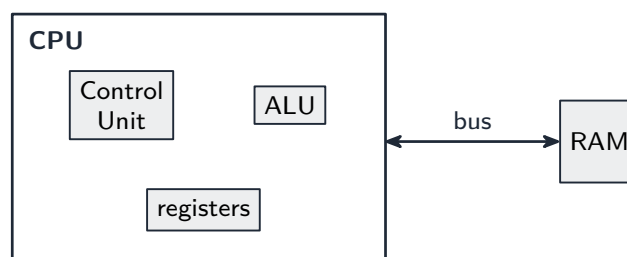
## New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

### ■ 1 Inside the processor

A computer keeps its instructions and data together in memory — the **stored program** 存储程序 idea. The processor contains:

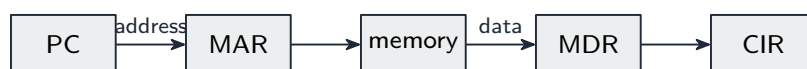
- an **arithmetic and logic unit** 算术逻辑单元 that does the maths and logic;
- a **control unit** 控制单元 that directs everything;
- small, very fast stores called **register** 寄存器 cells.



*Worked example.* To work out  $3 + 4$ : the control unit fetches and decodes the "add" instruction, the ALU adds the two numbers, and a register holds the answer 7.

### ■ 2 Two important registers

- the **Program Counter** 程序计数器 holds the address of the next instruction to run;
- the **Accumulator** 累加器 holds the result of a calculation.



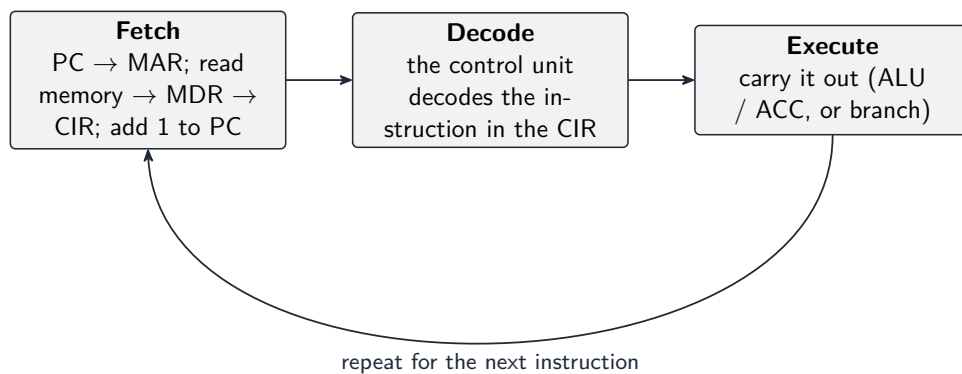
*Worked example.* Say the Program Counter holds 30. The CPU copies 30 into the **memory address register** 存储器地址寄存器 (MAR) and reads memory location 30; the instruction arrives in the **memory data register** 存储器数据寄存器 (MDR), then moves to the **current instruction register** 当前指令寄存器 (CIR), ready to be run.

**Watch out:** the Program Counter holds the *address* of the next instruction, not the instruction itself.

### ■ 3 The fetch–execute cycle

To run a program the processor repeats the **fetch–execute cycle** 取指执行周期:

1. **fetch** the next instruction from memory,
2. **decode** it (work out what it means),
3. **execute** it (carry it out).



Inside, data travels on the **data bus** 数据总线 and memory addresses travel on the **address bus** 地址总线.

*Worked example.* For "add the number at address 50": *fetch* the instruction from memory, *decode* it as an add, then *execute* it — the ALU adds the value stored at address 50 into the accumulator.

## Practice

---

**4.1** State what the Program Counter holds. [1]

---

**4.2** State what travels on the address bus. [1]

---

**4.3** Name the two main units inside the processor and say what each does. [2]

---

---

**4.4** State, in order, the three steps of the fetch–execute cycle. [3]

---

---

---

**4.5** Explain *why* the Program Counter must be increased during each fetch–execute cycle. [2]

---

---

**4.6 Challenge.** The Program Counter holds 30. Describe how the registers are used to fetch the instruction stored there. [3]

---

---

---