4.1 Central Processing Unit (CPU) Architecture

Name: ____________ Class: ________ Date: __________ **Total: 25 marks**

Objective

Build the skills to answer exam questions on **CPU architecture** —the Von Neumann model, the registers, the buses, and the fetch–execute cycle.

You must be able to:

- explain the **Von Neumann** 冯 · 诺依曼 model and the **stored-program** 存储程序 concept
- state the role of the **registers**, the **ALU** 算术逻辑单元, the **control unit** 控制单元 and the clock
- explain the **address**, **data** and **control buses**
- describe the stages of the **fetch–execute cycle** and the purpose of **interrupts** 中断

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.

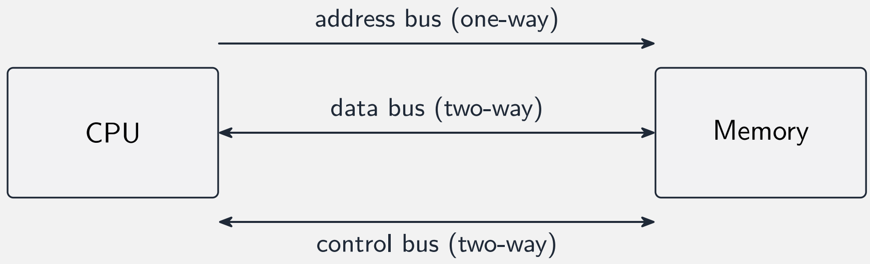
■ Von Neumann model and registers

In the **Von Neumann** model one memory holds **both program and data** (the **stored-program** idea), and the CPU runs the instructions one at a time, in order, unless a branch changes the flow. **Registers** 寄存器 are tiny, very fast stores inside the CPU. Each special-purpose register has a fixed job:

Register	Holds
PC (program counter)	address of the **next** instruction
MAR (memory address register)	the address being read/written
MDR (memory data register)	the data going to/from memory
CIR (current instruction register)	the instruction being decoded
ACC (accumulator)	the value the ALU is working on
IX (index register)	a value added to a base address (indexed addressing)

The **ALU** does arithmetic and logic; the **control unit** decodes instructions and sends control signals; the **clock** keeps everything in step.

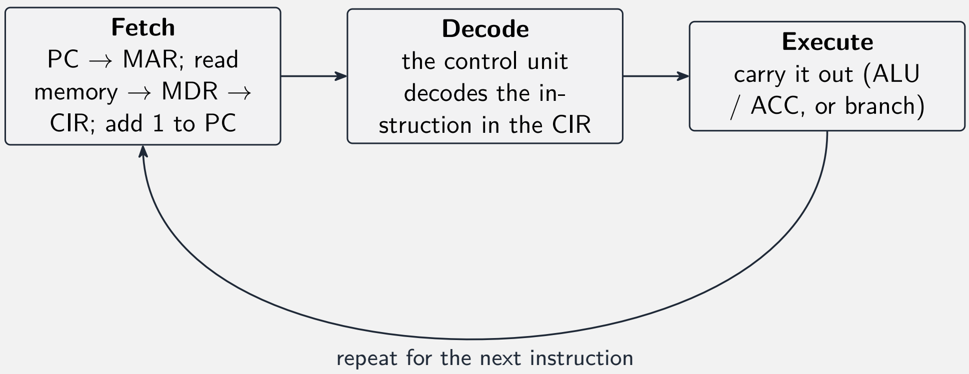
■ The three buses



The **address bus** is **one-way** (the CPU only ever *sends* an address to memory); the **data** and **control** buses are **two-way**. The **width** of a bus matters: a wider **data bus** carries more bits per transfer (faster), and a wider **address bus** can address **more** memory locations (each extra line doubles the number of addresses).

■ The fetch–execute cycle

Every instruction goes through the same three stages, then the cycle repeats:



An **interrupt** is a signal that makes the CPU pause the current program, run a service routine, then return —so urgent events are handled quickly.

2 Practice

Now apply the methods above.

2.1 State what the **stored-program concept** means. [1]

2.2 State the role of the **PC** and the **MAR**. [2]

2.3 State the difference between a **general-purpose** and a **special-purpose** register. [2]

2.4 State which bus is **one-way**, and explain why. [2]

3 Exam-style questions

3.1 Describe the stages of the **fetch–execute cycle**. [4]

3.2 (a) State the role of the **ALU** and the **control unit**. [2]

(b) State **two** factors that affect the performance of a CPU. [2]

3.3 Explain the purpose of an **interrupt**. [2]

3.4 During the fetch stage, describe the roles of the **MAR** and the **MDR**. [2]

3.5 A computer is redesigned with a **wider address bus**. Explain the effect this has on the computer. [2]

4 Go further

You are now ready for the real exam questions on this subtopic. Open the **4.1 CPU Architecture** past-paper sheet in the Library, or try these in **Practice mode**:

- 9618/11 N24 —Q3 (registers and the fetch–execute cycle)
- 9618/12 N24 —Q3 (CPU architecture and buses)
- 9618/11 J25 —Q6 (registers, buses and performance)
- 9618/13 N25 —Q6 (the fetch–execute cycle)

Solutions

2.1 Both the **program (instructions) and the data** are held together in the **same memory**.

2.2 PC holds the address of the **next** instruction; MAR holds the address currently being read from or written to memory.

2.3 A special-purpose register has a **fixed role** in the cycle (e.g. PC, MAR); a general-purpose register holds **temporary values** chosen by the program.

2.4 The **address bus** is one-way; the CPU only ever **sends** an address to memory — memory never sends an address back.

3.1 **Fetch:** the address in the PC is copied to the MAR; the instruction is read from memory into the MDR, then the CIR; the PC is increased by 1. **Decode:** the control unit decodes the instruction in the CIR. **Execute:** the instruction is carried out (ALU/ACC, or a branch).

3.2 (a) The ALU performs **arithmetic and logic** operations; the control unit **decodes instructions and sends control signals**.
(b) Any two of: clock speed; number of cores; cache size; bus width.

3.3 An interrupt is a signal that makes the CPU **pause** the current program, run a routine to handle an urgent event, then **return** to where it left off.

3.4 The MAR holds the **address** of the instruction to fetch; the MDR holds the **instruction/data** read back from that address.

3.5 A wider address bus has more lines, so it can carry **more distinct addresses**, letting the CPU directly address **more memory** (each extra line doubles the addressable memory).