A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

15.1 Processors, Parallel Processing and Virtual Machines

Name: _____ Class: _____ Date: _____ **Total: 17 marks**

KEY WORDS risc 精简指令集 • cisc 复杂指令集 • virtual machine 虚拟机 • mimd 多指令多数据 • simd 单指令多数据

Objective

Build the skills to answer exam questions on **processors and parallelism** — RISC/CISC, pipelining, Flynn's taxonomy and virtual machines.

You must be able to:

- compare **RISC** and **CISC** processors
- explain **pipelining** and the role of **registers**
- describe **Flynn's taxonomy** and a **virtual machine**

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.

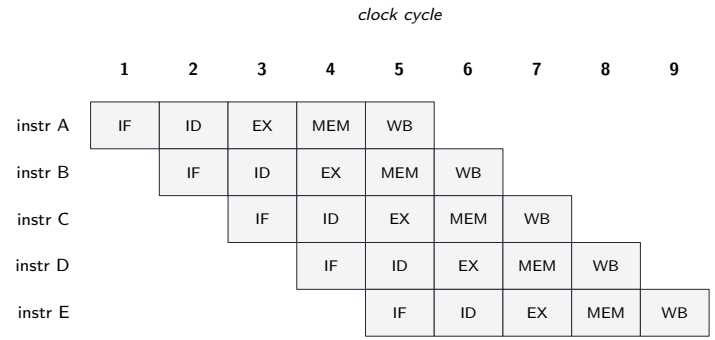
RISC vs CISC

Feature	CISC	RISC
Instruction set	many, complex	few, simple
Instruction length	variable	fixed
Memory access	many instructions	only **load/store**
Pipeline-friendly	harder	naturally

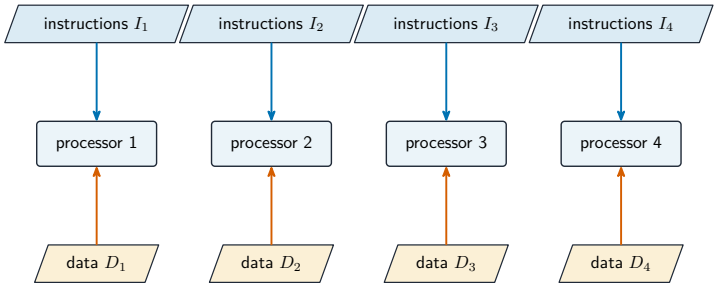
RISC keeps data in **many registers** (fast) and only `load/store` touch memory.

Pipelining

A **pipeline** overlaps the stages of consecutive instructions, like an assembly line, so once full, **one instruction completes per cycle**:



pipeline full — one instruction finishes per cycle



MIMD parallelism runs many instruction streams

RISC's **fixed-length, simple** instructions make each stage take equal time. A pipeline can **stall** on a **hazard** — a data hazard (a needed result isn't ready) or a control hazard (a branch).

Flynn's taxonomy and virtual machines

- **SISD** — one instruction, one data stream (single core).
- **SIMD** — one instruction on **many data items** (GPU).
- **MISD** — rare/theoretical.
- **MIMD** — many processors, different instructions and data (multi-core, clusters).

A **virtual machine** is **software** that behaves like a separate computer, running its own OS on top of a host — used for isolation, testing and running several OSes at once.

2 Practice

Now apply the methods above.

2.1 State **one** difference between a **RISC** and a **CISC** processor. [1]

2.2 State what **pipelining** does. [1]

2.3 State what **SIMD** stands for and give **one** example of SIMD hardware. [2]

2.4 State what a **virtual machine** is. [1]

3 Exam-style questions

3.1 Describe **two** features of a RISC processor that make it well-suited to **pipelining**. [2]

3.2 Explain how **pipelining** improves performance, and name **one** type of hazard that can stall it. [3]

3.3 State the **four** categories of **Flynn's taxonomy**. [2]

3.4 State **one** characteristic of a **massively parallel** computer. [1]

3.5 Identify **four** features of a **RISC** processor. [4]

Solutions

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

2.1

- any one: RISC has **few simple fixed-length** instructions, CISC has **many complex variable-length** ones; in RISC only load/store access memory

2.2

- it **overlaps the stages** of consecutive instructions so more than one is processed at once (one finishes per cycle when full)

2.3

- **Single Instruction, Multiple Data**
- example: a **GPU** / graphics card / CPU vector unit

2.4

- **software** that behaves like a separate (virtual) computer, running its own operating system on a host machine

3.1

- any two: **fixed-length** instructions (each stage takes equal time); **simple** instructions (each ~1 cycle); register-to-register operations (only load/store touch memory), so stages are predictable

3.2

- while one instruction is being executed, the **next is being decoded and another fetched**
- so instructions overlap and throughput rises to ~one per cycle
- a **data** hazard (or control/branch hazard) can stall it

3.3

- **SISD, SIMD, MISD, MIMD**

3.4

- any one: **thousands of processors**; each with its **own (distributed) memory**; they communicate by **message passing**; it is MIMD

3.5

- any **four**: a small set of **simple** instructions; **fixed-length** instructions; only **load/store** instructions access memory; **many general-purpose registers**; a **hardwired** control unit; designed for **pipelining** (about one instruction per cycle)