1.1 PROCESSORS, INPUT/OUTPUT AND STORAGE · 1.1.2

Types of processor

CISC vs RISC, the **GPU** and its non-graphics uses, and **multicore & parallel** systems. Spec 1.1.2(a)(b)(c).

01 CISC vs RISC

CISC

Complex instruction set. One instruction = several steps. Fewer lines, complex hardware, more power.

RISC

Reduced instruction set. One cycle each. More lines, simple hardware, easy to pipeline, low power.

Mobile Phones use **RISC** (ARM): less heat, longer battery.

02 CPU vs GPU

CPU

A **few** powerful cores. Sequential, branching work. Low latency.

GPU

Thousands of simple cores. Same operation on lots of data (SIMD). High throughput.

Non-graphics ML, simulation, image/video, finance.

03 Multicore & parallel systems

Multicore Several full cores on one chip; each has its own fetch-decode-execute cycle, so instructions run at the same time.

Parallel Many processors together (e.g. supercomputers for weather, climate, DNA).

Catch Full speed-up only if the software is written to run in **parallel**; sequential tasks leave cores idle.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Seven slips that lose marks on "types of processor" questions.

01

CISC = **C**omplex (large set, fewer lines). **R**ISC = **R**educed (small set, one cycle each, more lines).

02

RISC suits mobiles because simple instructions = **low power**, less heat, easy pipelining → longer battery life.

03

A GPU is not "a better CPU". It only wins on **parallel, repetitive** work; CPUs win on sequential/branching tasks.

04

Name a **non-graphics** GPU use when asked: machine learning, simulation, image processing, finance.

05

A GPU has **thousands of simple cores** (SIMD); a CPU has a **few powerful** ones.

06

More cores ≠ proportionally faster — only if the software is **parallelised**.

07

For a "CISC vs RISC" discussion: cover the differences (instruction set, lines of code, power, pipelining) **and** reach a justified conclusion for the given device.