1.1 PROCESSORS, INPUT/OUTPUT AND STORAGE · 1.1.1

Processor performance & pipelining

The three factors that affect CPU performance, the **cache hierarchy**, and how **pipelining** raises throughput. Spec 1.1.1(c)(d).

01 Performance factors

Clock speed Cycles/sec (GHz). Higher = more instructions/sec; more heat & power.

Cores Each is a full processor. More cores = more at once, **if** software is parallel.

Cache Fast memory near the CPU. More cache = fewer slow RAM trips.

02 Cache hierarchy

L1 cache	fastest · smallest
L2 cache	larger · slower
L3 cache	shared across cores
RAM	gigabytes · much slower

03 Clock speed & cores — the nuance

1 Hz one cycle per second; PCs run at 2–4 GHz.

Limit higher clock = more heat/power, so makers add cores instead.

Cores dual = 2, quad = 4 processors on one chip.

Catch a quad-core is rarely 4× faster — only parallelisable software benefits.

04 Cache hit vs miss

Hit data found in cache. Very fast.

Miss slower trip down toward RAM.

05 Pipelining

A

Without it

Each instruction finishes its full fetch-decode-execute before the next starts. Parts of the CPU sit idle.

B

With it

Stages overlap: while I1 executes, I2 decodes and I3 is fetched. More instructions finish per second.

!

Key point

Pipelining does **not** speed up one instruction. It raises **throughput** by keeping the CPU busy.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Seven slips that lose marks on performance and pipelining questions.

01

"State two factors" needs two of: **clock speed**, **number of cores**, **cache**. "A faster CPU" earns nothing.

02

A quad-core is **not** always 4× faster — only if the software is written for **parallel** processing.

03

Pipelining raises **throughput**; it does not speed up a single instruction.

04

More **cache** can beat a higher clock speed because the CPU waits less on slow RAM.

05

Clock speed is in **hertz** (GHz = billions of cycles/sec). One operation begins per clock pulse.

06

Higher clock speed costs **more heat and power** — a useful point in "discuss" questions.

07

To "define pipelining" for 3 marks: overlapping the **fetch, decode and execute** stages of **consecutive** instructions so they run at the same time, increasing instructions completed per second.