

15 RISC and CISC – Part 1

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English

Write it 3 times (English)

CISC

RISC

registers

Recall

There are two styles of CPU design, and they make opposite trade-offs.

New ideas

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

■ 1 CISC vs RISC

CISC

many, complex instructions
variable length
does more per instruction

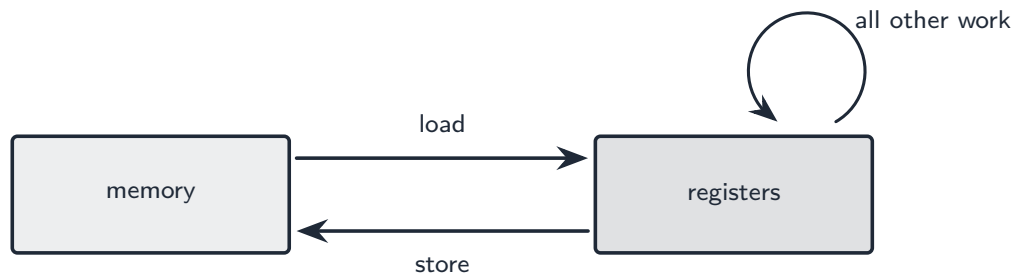
RISC

few, simple instructions
fixed length
each instruction is fast

- **CISC** has many complex instructions of variable length – does more per instruction (Intel x86);
- **RISC** has few simple instructions of fixed length – each is fast (ARM).

Worked example. To add two numbers held in memory, a CISC CPU might use a *single* instruction that reads memory, adds and writes back. A RISC CPU splits the same job into simple steps: load each value, add the registers, store the result.

■ 2 Load and store



In RISC, only **load** and **store** touch memory; everything else works between **registers**, which are very fast.

Worked example.

```
LOAD  R1, x      // copy x from memory into register R1
LOAD  R2, y      // copy y into R2
ADD   R3, R1, R2  // R3 ← R1 + R2 (registers only)
STORE R3, z      // copy the result back to memory z
```

Watch out: RISC having simpler instructions does *not* make its programs slower. Each instruction is fast and fits a pipeline neatly; CISC does more per instruction but each one is slower and varies in length.

Practice

Try these in order.

15.1 Which design has *many complex* instructions? [1]

15.2 Which design uses *fixed-length* instructions? [1]

15.3 Give one example of a RISC processor. [1]

15.4 In RISC, which two instructions touch memory? [1]

15.5 Why does RISC keep data in *registers*?

[1]

15.6 A phone uses an ARM (RISC) chip rather than a CISC one. Suggest one reason RISC suits a battery-powered device.

[2]

15.7 Challenge. A task needs two numbers in memory (**a** and **b**) added, and the result stored in **c**. Write the RISC steps (load, add, store) it would use.

[2]