Week 4: CISC vs RISC architecture-2

The performance of a CPU is highly dependent on its internal architecture. The goal is to do as much 'work' in a given amount of time as possible.

The performance equation

$$\frac{time}{program} = \frac{time}{cycle} \times \frac{cycles}{instruction} \times \frac{instructions}{program}$$

The **CISC** approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction.

RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.

The equation above is commonly used for expressing a computer's performance ability:

There is still considerable controversy among experts about which architecture is better. Some say that RISC is cheaper and faster and therefore the architecture of the future.

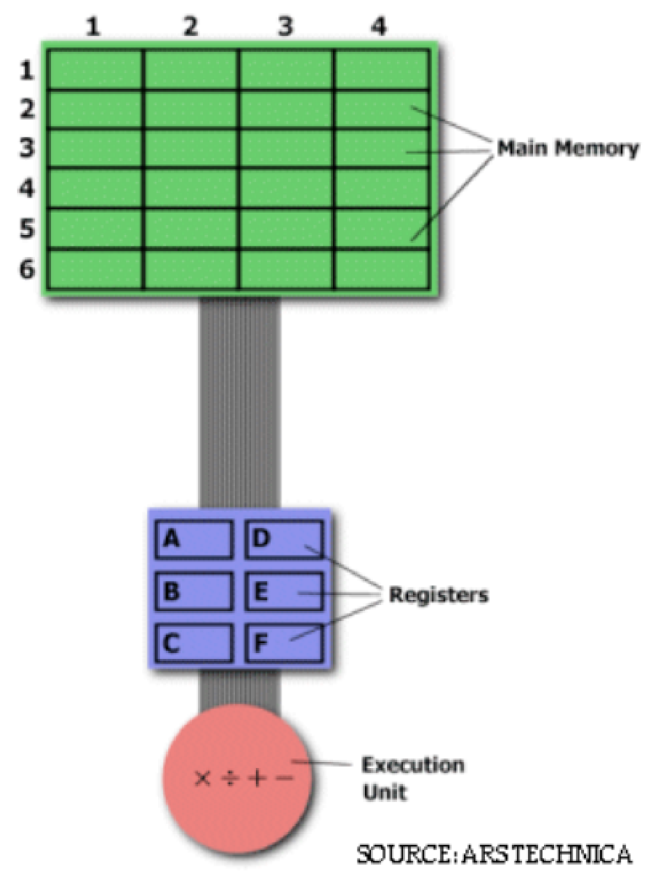
Multiplying two numbers in memory

The main memory is divided into locations numbered from (row) 1: (column) 1 to (row) 6: (column) 4.

The execution unit is responsible for carrying out all computations.

The execution unit can only operate on data that has been loaded into one of the six registers (A, B, C, D, E, or F).

We want to find the product of two numbers – one stored in location 2:3 and another stored in location 5:2 – and then store the product back in the location 2:3.



Complex instruction set computing

- A CISC processor would come prepared with a specific instruction (we'll call it "MULT").
- When executed, this instruction loads the two values into separate registers, multiplies the operands in the execution unit, and then stores the product in the appropriate register.
- The entire task of multiplying two numbers can be completed with one instruction: MULT 2:3, 5:2.
- It operates directly on the computer's memory banks and does not require the programmer to explicitly call any loading or storing functions.
- It closely resembles a command in a higher level language.

Reduced instruction set computing

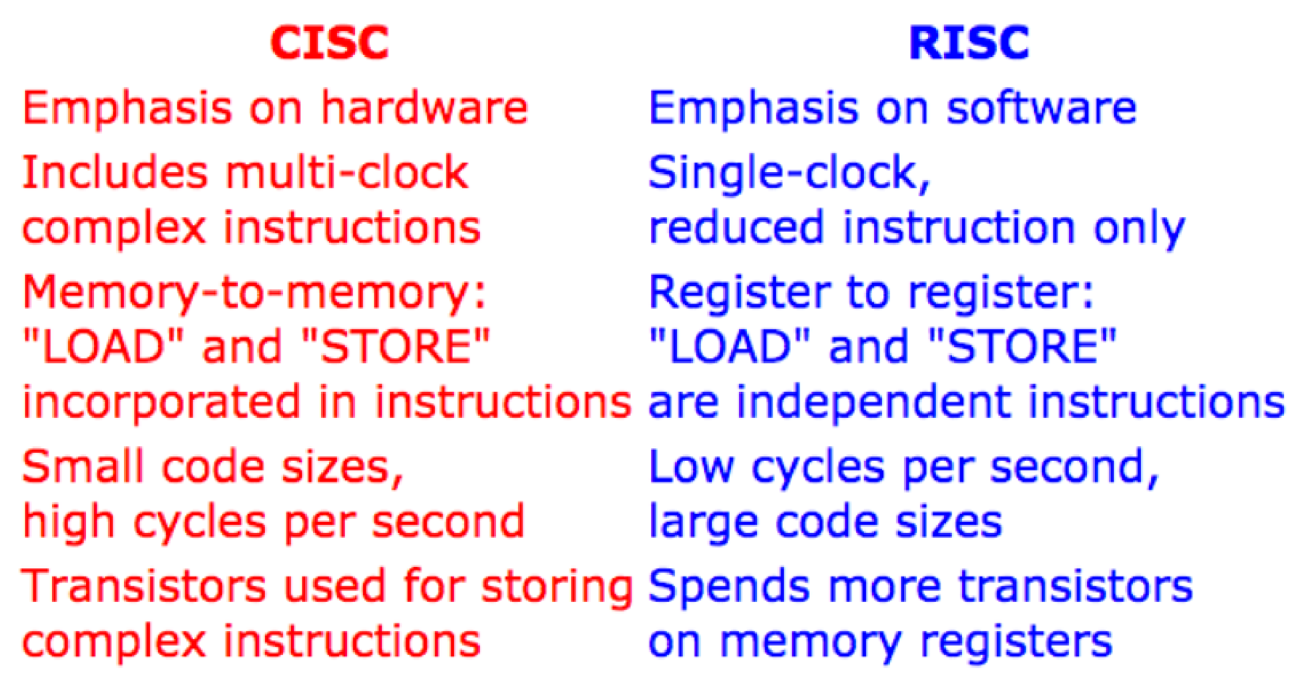
- Only use simple instructions that can be executed within one clock cycle.
- The "MULT" command described above could be divided into three separate commands: "LOAD," which moves data from the memory bank to a register, "PROD," which finds the product of two operands located within the registers, and "STORE," which moves data from a register to the memory banks.
- In order to perform the exact series of steps described in the CISC approach, a programmer would need to code four lines of assembly:

```
LOAD A, 2:3 ;

LOAD B, 5:2 ;

PROD A, B;

STORE 2:3, A
```



CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers