Today, many people say that the difference between RISC and CISC computer processors is negligible and that their performance is essentially the same. But is that really true? The reality is that there are still plenty of differences between them and that can make one better suited to the other to certain applications.

We'll take a look at some of the basics of microprocessors before delving into the deeper questions about RISC vs. CISC.

What is a microprocessor?

RISC and CISC are both examples of different designs for computer microprocessors. The microprocessor, known as the central processing unit (CPU), is the brain of the computer. It reads instructions from memory that tells the computer what to do. Microprocessors use those instructions to perform very simple tasks, such as basic logic, controlling and input/output operations that are specified by the makers of the operating system.

RISC, which is an acronym for Reduced Instruction Set Computer, and CISC, short for Complex Instruction Set Computer, refer to the category of the processor, or more accurately, the instruction set architecture (ISA). The ISA is the particular way the processor communicates with the human programmer. The earliest computers had a CISC architecture, but in the 1980s, RISC architecture was developed to overcome the increasing complexity of CISC processors.

The CISC approach

The main idea behind CISC processors is that a single instruction can be used to do all of the loading, evaluating and storing operations. The goal of the CISC approach is to minimize the number of instructions per programme. However, that increases the number of cycles per instruction.

The RISC approach

The primary goal of RISC processors is to make hardware simpler by using an instruction set that's composed of a few basic steps for loading, evaluating and storing operations. It's not necessarily about having fewer instructions, it's more about how those instructions are used. That reduces the number of cycles per instruction but comes at the cost of increasing the number of instructions per program.

What are the differences between RISC and CISC?

The purpose of both RISC and CISC architectures is to increase CPU performance, but they try to achieve that goal in different ways. Generally speaking, RISC is seen by many as an improvement over CISC. The argument for RISC over CISC is that having a less complicated set of instructions makes designing a CPU easier, cheaper and quicker.

The primary difference between RISC and CISC architecture is that RISC-based machines execute one instruction per clock cycle. In a CISC processor, each instruction performs so many actions that it takes several clock cycles to complete. In a RISC processor, every instruction also has a fixed memory size, which makes them easier to decode and execute. In a CISC machine, the instructions can be variable lengths, which increases the processing time.

At a glance, the main differences between RISC and CISC architecture are as follows:

RISC	CISC
Emphasis on software	Emphasis on hardware
Small number of fixed length instructions	Large number of instructions
Simple, standardized instructions	Complex, variable-length instructio
Single clock cycle instructions	Instructions can take several clock cycles
Heavy use of RAM	More efficient use of RAM
Low cycles per second with large code sizes	Small code sizes with high cycles pe second

RISC vs. CISC: Which is better?

The short answer is that no one architecture is better than the others. Each has its benefits and roadblocks, which make them better suited to particular scenarios. The key is to match up those advantages and roadblocks with the intended application to find the most appropriate instruction set architecture for you.