CISC vs RISC

Complex Instruction Set Computing (CISC) vs Reduced Instruction Set Computing (RISC)

They are both types of microprocessor architectures.

CISC	RISC
Programmer Oriented	Machine Oriented
Longer Instructions, Shorter Code	Simpler Instructions, Longer Code
Emphasis on Hardware since code is short but complex	Emphasis on Software since code is long but simple
1 line of assembly may take multiple clock cycles	1 line of assembly = 1 Clock Cycle
Hard to Pipeline	Easy to Pipeline

Multiply two numbers

CISC	RISC
MULT X, X	Load X, X Load X, X Prod X, X Store X

RISC Architecture is used in:

- PowerPC processors used by MacBook from 1998 to 2005,
- ARM Processors (iPhones, Servers),
- Qualcomm Processors (Android Phones) -> Power Efficiency for mobile devices

Intel uses microcode, a lower level machine code to take advantage of the RISC pipeline efficiencies –> x86
The x86 architecture design can be classified as a CISC design:

- Desktop PC

CISC architecture was initially used.

RAM used to run at around the same speed as the CPU. RAM was expensive in the 1970s. Utilizing less RAM was preferred, by designing complicated processor, a single line of code could translate to multiple instructions, reducing the amount of RAM usage. However, multiple clock cycles resulted in heavy power usage. Programming in assembly language was common.

The RISC architecture seemed like a viable solution to this, with more registers within the CPU, thus reducing the use of RAM. The SPARC Processor is an example of this early success of the RISC architecture.

However, CISC processors started implementing RISC-like features, in their processors: Pentium.