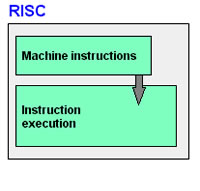
**Difference between RISC and CISC**

**Key difference: The main difference between RISC and CISC is in the number of computing cycles each of their instructions take. The difference the number of cycles is based on the complexity and the goal of their instructions.**

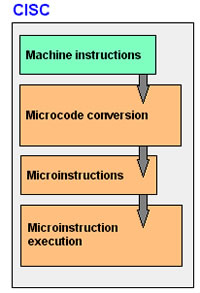
**The term RISC stands for ‘Reduced Instruction Set Computer’. It is a CPU design strategy based on simple instructions and fast performance.**

**RISC is small or reduced set of instructions. Here, each instruction is meant to achieve very small tasks. In a RISC machine, the instruction sets are simple and basic, which help in composing more complex instructions. Each instruction is of the same length; the instructions are strung together to get complex tasks done in a single operation. Most instructions are completed in one machine cycle. This pipelining is a key technique used to speed up RISC machines.**

**RISC is a microprocessor that is designed to carry out few instructions at the same time. Based on small instructions, these chips require fewer transistors, which make the transistors cheaper to design and produce. Some other features of RISC include:**

* **Less decoding demand**
* **Uniform instruction set**
* **Identical general purpose register**
* **Simple addressing nodes**
* **Few data types in hardware**

**Also, while writing codes, RISC makes it easier by allowing the programmer to remove unnecessary codes and prevents wasting of cycles.**

****

**The term CISC stands for ‘Complex Instruction Set Computer’. It is a CPU design strategy based on single instructions, which are capable of performing multi-step operations.**

**CISC computers have shorted programs. It has a large number of complex instructions, which takes long time to execute. Here, a single set of instruction is covered in multiple steps; each instruction set has more than three hundred separate instructions. Most instructions are completed in two to ten machine cycles. In CISC, instruction pipelining is not easily implemented.**

**The CISC machines have good performances, based on the simplification of program compilers; as the range of advanced instructions are easily available in one instruction set. They design complex instructions in one simple set of instructions. They perform low level operations such as an arithmetic operation, or a load from memory and memory store. CISC makes it easier to have large addressing nodes and more data types in the machine hardware. However, CISC is considered less efficient than RISC, because of it inefficiency to remove codes which leads to wasting of cycles. Also, microprocessor chips are difficult to understand and program for, because of the complexity of the hardware.**

**Comparison between RISC and CISC:**

|  |  |  |
| --- | --- | --- |
|  | **RISC** | **CISC** |
| **Acronym** | **It stands for ‘Reduced Instruction Set Computer’.** | **It stands for ‘Complex Instruction Set Computer’.** |
| **Definition** | **The RISC processors have a smaller set of instructions with few addressing nodes.** | **The CISC processors have a larger set of instructions with many addressing nodes.** |
| **Memory unit** | **It has no memory unit and uses a separate hardware to implement instructions.** | **It has a memory unit to implement complex instructions.** |
| **Program** | **It has a hard-wired unit of programming.** | **It has a micro-programming unit.** |
| **Design** | **It is a complex complier design.** | **It is an easy complier design.** |
| **Calculations** | **The calculations are faster and precise.** | **The calculations are slow and precise.** |
| **Decoding** | **Decoding of instructions is simple.** | **Decoding of instructions is complex.** |
| **Time** | **Execution time is very less.** | **Execution time is very high.** |
| **External memory** | **It does not require external memory for calculations.** | **It requires external memory for calculations.** |
| **Pipelining** | **Pipelining does function correctly.** | **Pipelining does not function correctly.** |
| **Code expansion** | **Code expansion can be a problem.** | **Code expansion is not a problem.** |
| **Applications** | **Used in high end applications such as video processing, telecommunications and image processing.** | **Used in low end applications such as security systems, home automations, etc.** |