RISC & CISC

Thursday, 2 November 2023 5:52 PM

RISC & CISC are two different architectures that differ in how they process instructions.

Reduced Instruction Set Computer (RISC):

- RISC is designed to perform a smaller number of types of computer instructions so that they can operate at a higher speed
- They acheive this by simplifying the processor's ability to execute instructions, using a smaller set of simple and general instructions.
- Each instruction is designed to execute very quickly within the processor.

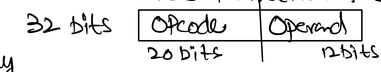
Complex Instruction Set Computer (CISC):

- CISC processors have a larger set of instructions, some of which can perform complex tasks in a single instructions.
- These processors are designed to complete these tasks by using fewer lines of assembly code, potentially leading to more efficient programs.
- The complexity of the instruction can lead to slower execution for each individual instruction compared to RISC.

Differences:

Instruction Size: -RISC instructions are generally the same size, whereas CISC has variable-sized instructions.

RISC Instruction Size



Instruction

Execution Time: -RISC usually has faster instruction execution due to simplicity, while CISC may take longer because some instructions are more complex.

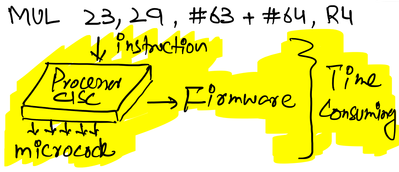
Efficiency: -RISC is often more efficient in terms of cycles per instruction, but CISC is more efficient in terms of program size.

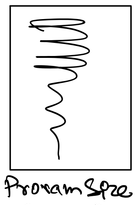
Complexity: -CISC processors are typically more complex in the hardware design, while RISC processors are simpler.

In summary, RISC is like a chef who use afew simple & quick techniques to make a variety of dishes efficiently, while CISC is like a chef who has complex array of tools and techniques to make dishes that might be elaborate in one step.

CISC

MUL 23, 29, #63 + #64, R4

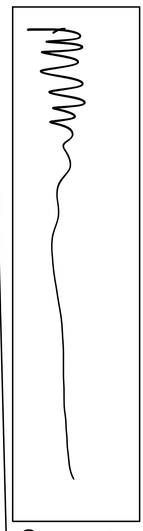




Program Size "Smaller"

- Flexible instruction size
- Short code
- It was made (initially) to shorten the gap b/w Low level & High level languages.
- Programer of LLL & Compiler are at ease. As they have to do less and burdon is taken by hardware.
- Number of instructions is larger.
- Processor does more & generates higher heat.
- Desktop, Laptop etc.
- Require cooling systems
- They work memory to memory
- More clock cycles are required to execute a single instruction
- Expensive as production is complex.
- There are many instructions for memory addressing.
- Pipelining is difficult.

RISC



Program Size "Larger"

```
LDM  #63
ADD  #64
PROD 29
PROD R4
STO  23
END
```

- Fixed instruction size
- Large code
- Programmer of LLL & Compiler of HLL has more to do, as there are simpler instruction and they have to write more & convert more.
- Number of instructions is short.
- Processor has to do less and generally hardware is not heatedup.
- generally castings are enough to dissipate & pacify the heat.
- They work register to register.
- Mostly a single clock cycle is enough for one instruction.
- Inexpensive.
- Only load & store are to acess memory.
- Pipelining is easier.