Jesus Otteson

Hau Fung Moy Kwan

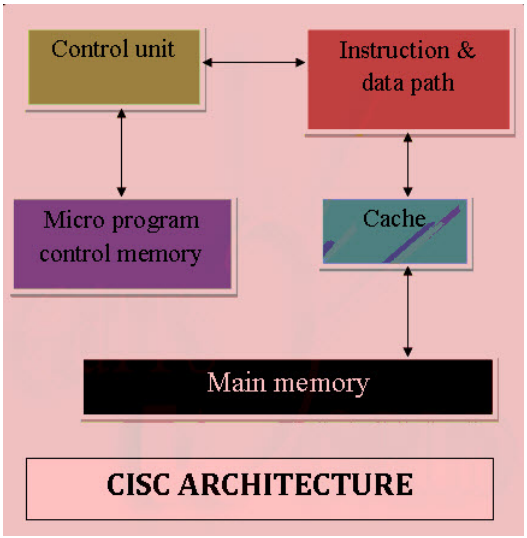
Computer Architecture

April 26, 2019

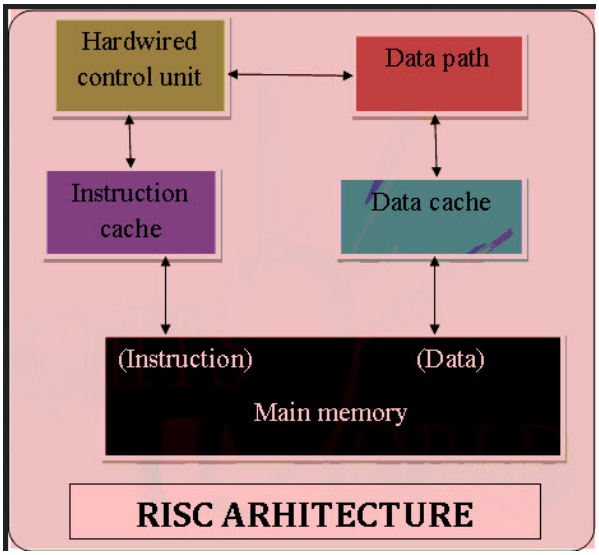
RISC vs CISC, What works best and when?

RISC and CISC are two different types of computer architecture that each would perform well in different conditions. Each method corresponds to how many instructions and cycle a computer would run to complete a set of commands. This research paper will go in-depth into how each works and in what situations we would want to use each. So what is RISC and CISC Architects? There have been numerous technologies and & tools invented to implement architecture type to fulfill the needs of hardware designers. More specifically, in regards to processors, there are two specific types of concepts that work on both the software and hardware side, these are the RISC and CISC.

CISC's approach attempts to use the least amount of instructions per program to reduce the number of cycles per instruction. These types of computers focus on the decease of memory cost, therefore, when large programs need more memory and the cost of memory becomes expensive, a CISC's process will reduce the number of instructions by embedding the number of operations into single instructions making the instructions more complex. Deriving its name **CISC** from **C**omplex **I**nstruction **S**et **C**omputer.



RISC's approach is to simplify the entire process, this is uses in many portable devices as it is power efficient. You can see RISC used in iPods, phones, portable gaming consoles, and so on. RISC reduces the cycles per instruction at the cost of the number of instructions by pipelining the programs. This is done by overlapping the execution of the instructions. RISC helps synthesize complex data types and supports few simple data types. The amount of work it has to do is reduced by separating "Load" and "Store" instructions. However, with RISC more RAM is required to store assembly level instructions. Deriving its name **RISC** from **R**educed **I**nstruction **S**et **C**omputer.



Each have a couple of advantages and disadvantages I would like to cover to further deconstruct the difference between each. The advantages of CISC is the simplicity of implementing microprogramming in assembly, microcoding new instructions allow designers to make CISC machines upward compatible, and as instructions become more accomplished, fewer instructions are needed to implement a specific task. However, with this there is some disadvantages to this, namely there is going to be a huge hit taken to the performance of the machine due to the amount of clock time taken by the different instructions. As the conditional codes are set by the CISC instructions, each instruction takes time for any desired conditions to be set into place since the compiler has to examine each conditional code before anything is set.

As for RISC, the advantages of using this computer architecture system is the compatibility with high-level language compilers so more efficient code can be produced. This system allows the freedom of using the space on microprocessors because of how simple it is. Many RISC processors use the registers for passing arguments and holding the local variables. RISC functions use only a few parameters, and the RISC processors cannot use the call instructions, and therefore, use a fixed length instruction which is easy to pipeline. The speed of the operation can be maximized and the execution time can be minimized. Very less number of instructional formats, a few numbers of instructions and a few addressing modes are needed. Most of the disadvantages of RISC come down to the conversion of CISC into RISC and the first level cache of the RISC processors, in which these processors have large memory caches on the chip itself. For feeding the instructions, they require very fast memory systems.

CISC	RISC
It is prominent on Hardware	It is prominent on the Software
It has high cycles per second	It has low cycles per second
It has transistors used for storing Instructions which are complex	More transistors are used for storing memory
LOAD and STORE memory-to-memory is induced in instructions	LOAD and STORE register-register are independent
It has multi-clock	It has a single - clock

The major difference between these systems that I was able to uncover is that CISC is more hardware based vs RISC which heavily relies on how the software is created. CISC tends to be more outdated in this manor since it requires hardware to im-

prove while still using basic languages like assembly. However, RISC is the newest and most up to date version of computer architecture where a downside use to be that it was unsupported on most devices, now that has changes and almost everything runs on RISC. Because of its ability to run complex languages, simplify instructions to reduce cycles, and use minimal power compared to CISC, this method has become the predominately superior architecture for every day device, meanwhile CISC would be most effective when trying out new techniques in hardware or improving performance on super computers.

In conclusion RISC seems to be the best architecture style as it is the most current and up to date, it is also the most appealing to software developers as the performance of the computer is dependent on how well a developer can code while also being compatible with high level languages that allow more efficient code.

References:

URL: https://www.edgefxkits.com/blog/what-is-risc-and-cisc-architecture/

- Website Title: Edgefx Kits Official Blog
- Article Title: What is RISC and CISC Architecture?
- Date Published: February 19, 2015
- Date Accessed: **April 27, 2019**

URL: https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/

- Website Title: **RISC vs. CISC**
- Date Accessed: **April 27, 2019**

URL: https://vjkadam.wordpress.com/2018/01/17/the-advantages-and-disadvantages-of-risc-and-cisc/

- Website Title: Vinod kadam
- Article Title: The Advantages and Disadvantages of RISC and CISC
- Date Published: **January 17, 2018**
- Date Accessed: **April 27, 2019**