© 2014 IJIRT | Volume 1 Issue 6 | ISSN : 2349-6002

RISC AND CISC ARCHITECTURE

Kirti Tokas, Dhruv Sharma, Lokesh Yadav

Abstract- **Performance comparison between RISC and CISC has been quite a well-known topic of research for many years. This paper gives an architectural comparison between the two, i.e., RISC and CISC, their architectural advantages, development and also a new trend in architecture i.e., CRISC (Complex-Reduced Instruction Set Computer).**

I. INTRODUCTION

The microprocessor chips, from the architecture point of view can be classified into two categories: Complex Instruction Set Computers (CISC) and Reduce Instruction Set Computers (RISC). In both, the main purpose is to ameliorate system performance. The research on these two architectures is very compelling, demanding, and sometimes complicated.

CISC (Complex Instruction Set Computer) computers have a complex

instruction set in which microcodes are used to execute several instructions. To make a complex instruction set more adaptable, microcodes are used. Through these hardware designs can be changed while maintaining the backward compatibility with instructions. Although microcodes used in CISC enhances the versatility of hardware but they also slows down the performance of microprocessor as number of instruction increases to execute every CISC instruction. Instruction set of CISC incorporates many instructions with varying sizes and execution cycles, which makes them difficult to pipeline. CISC microprocessors became prevalent from 1960's, each consecutive processor with further complex hardware and instruction sets. This trend started from Intel 80486, Pentium MMX to Pentium III.

RISC (Reduced Instruction Set Computer) chips came into existence around the mid-1970 as a response to CISC chips. At IBM's T.J Watson Research Center, John Cocke gave the basic concepts of RISC; the idea came from the IBM 801

Minicomputer which is used as a fast controller in a very large telephone switching system. RISC chip should have: few instructions, fix-sized instructions in a fixed format, execution on a single cycle of a processor and a Load / Store architecture. RISC promised higher performance, less cost and faster design time. The simple load/store computers such as MIPS 2 are commonly called RISC architectures. David A. Patterson coined the term RISC, after that John L. Hennessy invented the MIPS architecture to represent RISC.

The second section of this paper is comprised of the comparison between CISC and RISC. Section 3 presents the CRISC technology. Finally, the last section will be the conclusion of this work.

II. RISC VERSUS CISC

New generation processors are designed to give improved performance. For this we consider three main factors and they are:

•How fast you can crank up the clock.

•How much work you can do per cycle.

•How many instructions you need to perform a task.

This section will deal with the major differences between the two architectures based on the above factors.

A CISC processor has the following properties:

•Richer instruction set, some simple, some very complex

•Instructions generally take more than 1 clock to execute

•Instructions are variable in size

•Instructions interface with memory in multiple mechanisms with complex
addressing modes

•No pipelining

•Upward compatibility within a family

•Microcode control

•Work well with simpler compiler

As time passed, one of the non-RISC architecture with large market was the Intel x86 family, it has some specific characteristics that became associated with CISC:

•Segmented memory model

•Few registers

•Crappy floating point performance

CISC chips constitute of large and complex instructions. A powerful instruction set is always needed as it is said that hardware is always faster than software. A CISC chip uses fewer instructions than RISC but is relatively slow per instructions if compared to RISC.

RISC processor constitute of the following properties:

- Simple primitive instructions and addressing modes
- Instructions execute in one clock cycle
- Uniformed length instructions and fixed instruction format
- Instructions interface with memory via fixed mechanisms (load/store)
- Pipelining
- Instruction set is orthogonal (little overlapping of instruction functionality)
- Hardwired control
- Complexity pushed to the compiler.

Further RISC and CISC can be compared based on the following properties as:

RISC	CISC
Design is more cost-effective	Design is costlier compared to RISC
Superscaling is possible	Superscaling is not possible
Large number of registers are used	Few registers are used
Fixed format instruction	Variable format instruction
One clock cycle	Multiple cycle
Few addressing modes	Many addressing modes
Easy compiler design	Complex compiler design

Nowadays, the difference between RISC and CISC is no longer limited to instruction sets, but the whole chip architecture and system. What a programmer requires now is not the number of instructions but how fast a chip can execute instructions it is given and how compatible it is with the current software.

RISC was employed to limit the number of instructions and to execute these instructions in one cycle.

The table below shows some examples of CISC and RISC processors:

CISC Processors	RISC Processors
IBM 370/168	MIPS R2000
VAX 11/780	SUN SPARC
Microvax II	INTEL i860
INTEL 80386	MOTOROLA 8800
INTEL 80286	POWERPC 601
Sun-3/75	IBM RS/6000

CISC and RISC architectures, can be compared in many ways

- Based on instruction set- The instruction set chosen for a particular processor determines the way that machine language programs are constructed.
- Based on addressing modes- Addressing modes gives us an image about the memory or register referencing, which is one of the essential factors for comparison in performances.
- Based on the integer and floating point units.
- Based on instruction pipelining- More the pipelining stages the processor has, faster will be the execution of instructions.
- Based on the memory- The performance of the processor is also affected by the cache and the main memory.
- Based on the number of transistors-more the number of transistors, more compact will be the design, resulting in a faster and improved performance.

III. COMPLEX-REDUCE INSTRUCTION SET COMPUTERS (CRISC)

CRISC (Complex-Reduce Instruction Set Computers) is the hybrid between the two architectures. It came into existence by the release of Intel i486 processor and it was said that it has a RISC integer unit. With the continued evolution of both RISC and CISC designs, it has become less meaningful. The first pipelined "CISC" CPUs, such as 486s [19] from Intel, AMD, Cyrix, and IBM, supported every instruction that their predecessors did, but achieved high efficiency only on a fairly simple x86 subset (resembling a non-load/store "RISC" instruction set).

An example of CRISC is the Intel Pentium-Pro, which is an interesting combination of the two architectures. It still executes the CISC instruction set, but the internal implementation is a high performance "Post RISC" CPU.

CRISC is the future trend of CISC and RISC.

IV. CONCLUSION

The RISC design had advantages that result to a machine's excellent performance and has been adopted for commercial products, as the industry is designing more and more RISC-based processors. The RISC versus CISC debate was most certainly a good thing as it allowed the industry to explore and come up with solutions that have significantly raised the machine's performance we use today. Due to this exploration, both architectures have continuously developed and processors today have a little bit of something from both the architecture and also a new processor called CRISC is employed.

REFERENCES

1. Yi Gao, Shilang Tang, Zhangli Ding, "Comparison between CISC and RISC"
2. Marie Elma Domingo, "A presentation of the RISC and CISC instruction set architecture"