A CPE 407 TERM PAPER

TOPIC:

RISC & CISC OF COMPUTER ARCHITECTURE

presented by

GROUP 10

NAME	**MATRIC NUMBER**
Momoh Tomide Tiwalade	CPE/2018/1076
Adesola Akeem Omotayo	CPE/2018/1122
Tayo Adedimeji Bartholomew	CPE/2018/2006
Oyedokun Abdulmalik Oyekunle	CPE/2018/2007
Hamzat Utman Ayomide	CPE/2018/1116
Olatunde Samuel Olayinka	CPE/2018/1088
Adekanbi Michael	CPE/2018/1010
Ajayi Ayomide Timothy	CPE/2018/1023
Kasali Oladimeji Olaoluwa	CPE/2018/1120
Akubor Samuel Toluwani	CPE/2018/1034

INTRODUCTION

What is RISC and CISC of Computer Architecture?

RISC, or Reduced Instruction Set Computing, is a type of computer architecture that uses a smaller set of simple instructions that can be executed quickly and efficiently. CISC, or Complex Instruction Set Computing, is a type of computer architecture that uses a larger set of instructions that can perform more complex operations.

The history of RISC and CISC architecture can be traced back to the early days of computer design, when the first computers were being built. At the time, computers were designed with a focus on maximizing the amount of functionality that could be packed into a single instruction. This resulted in a more complex instruction set, with many different operations that could be performed in a single cycle. Over time, as computer technology advanced and the demands for faster and more efficient computing grew, the limitations of these complex instruction sets became more apparent. This led to the development of the RISC architecture, which was first proposed by John Cocke at IBM in the 1970s. The idea behind RISC was to simplify the instruction set, so that each instruction could be executed in a single cycle, thereby increasing the performance and efficiency of the computer.

In the following years, RISC became increasingly popular, and was widely adopted by many computer manufacturers, including Sun Microsystems, Apple, and MIPS Technologies. The success of RISC architectures led to the development of even more advanced RISC-based systems, such as the ARM processor, which is now used in many of the world's most popular smartphones and mobile devices.

At the same time, CISC architectures continued to evolve, with many manufacturers adding new features and instructions to their instruction sets. However, the trend towards smaller, more efficient, and more cost-effective computing systems has led to a decline in the popularity of CISC, and a growing focus on RISC-based architectures. Today, both RISC and CISC architectures continue to be used, with each having its own strengths and weaknesses, and the choice between them typically depending on the specific requirements of the system being designed.

Overview of RISC

RISC processors also use a simpler pipeline, which allows them to execute instructions faster than CISC processors. A pipeline is a series of stages that instructions go through before they are executed. RISC processors have a shorter pipeline with fewer stages, which reduces the time it takes for an instruction to be executed. This also allows for a higher clock speed, which results in faster instruction execution. Additionally, RISC processors use a technique called register-register operations, which allows for faster data movement between registers. RISC processors have a smaller number of registers compared to CISC processors, but they are designed to be more efficient in terms of data movement. This improves the performance and efficiency of the processor. Overall, RISC processors are designed to be fast and efficient by using a smaller set of simple instructions, a simpler pipeline, and efficient data movement techniques.

CHARACTERISTICS OF RISC

RISC (Reduced Instruction Set Computing) is a computer architecture that is characterized by several key features, including:

- Simple Instruction Set: RISC architecture uses a small, highly optimized instruction set, with a focus on executing instructions in a single cycle. This makes the design of the processor simpler and more efficient, reducing the number of transistors required and reducing power consumption.

- Fixed Instruction Length: RISC instructions are of fixed length, typically 32 or 64 bits, which makes the instruction decoding process more efficient and reduces the complexity of the processor design.

- Load/Store Architecture: RISC architectures use a load/store architecture, where data must be loaded into a register before it can be operated on. This simplifies the instruction set, as most operations are performed on data stored in registers, rather than in memory.

- Register-Oriented Design: RISC architectures are designed around a large number of general-purpose registers, which are used to store data and perform operations. This results in a more efficient design, as the processor can quickly access data stored in registers, rather than having to go to memory to retrieve it.

- Pipelining: RISC architectures typically use a pipelining approach, where instructions are executed in a series of stages. This allows the processor to execute multiple instructions simultaneously, increasing performance.

- High Clock Speed: RISC architectures are designed to run at high clock speeds, typically in the range of hundreds of megahertz to several gigahertz. This allows the processor to execute a large number of instructions per second, resulting in high performance.

- Simple Addressing Modes: RISC architectures typically use simple addressing modes, such as register-direct, register-indirect, and immediate, making the instruction decoding process more efficient and reducing the complexity of the processor design.

These characteristics of RISC architecture help to improve performance, reduce power consumption, and reduce the cost of the processor, making it well-suited for applications where energy efficiency and high performance are key requirements.

Overview of CISC

The basic principle behind CISC is to provide the processor with a wide range of instructions that can perform a variety of tasks, such as memory access, arithmetic, and logic operations, with a single instruction. This allows for more efficient use of memory and processing resources, as well as more flexibility in programming. However, CISC processors typically have more complex instruction decoders and require more transistors, which can lead to higher power consumption and heat dissipation.

Characteristics of CISC

CISC (Complex Instruction Set Computing) is a computer architecture that is characterized by several key features, including:

- Large Instruction Set: CISC architecture uses a large, complex instruction set, with a focus on providing more functionality in a single instruction. This results in a more powerful architecture, but also with a higher power consumption and longer instruction cycle times.

- Variable Instruction Length: CISC instructions can be of variable length, ranging from a few bytes to several kilobytes. This makes the instruction decoding process more complex, but also allows for more functionality to be included in a single instruction.

- Memory-Memory Operations: CISC architectures allow for memory-to-memory operations, where data is transferred directly between memory locations, without the need to first load the data into a register. This makes the instruction set more powerful, but also increases the complexity of the processor design.

- Microcode: CISC architectures often use microcode to implement complex instructions, where a single instruction is executed as a series of microinstructions. This allows for a greater level of functionality, but also increases the complexity of the processor design.

- Complex Addressing Modes: CISC architectures often use complex addressing modes, such as segmented, indexed, and indirect, which provide a greater level of functionality but also make the instruction decoding process more complex.

- Large Register Files: CISC architectures typically use large register files, which allow for more data to be stored and manipulated in a single instruction. This results in a more powerful architecture, but also increases the complexity of the processor design.

- Legacy Compatibility: CISC architectures are often favored in situations where compatibility with legacy software and hardware is important, as they can often execute legacy CISC instructions more efficiently than RISC architectures.

These characteristics of CISC architecture help to provide a more powerful and flexible architecture, with a greater level of functionality. However, they also result in a more complex design, with a higher power consumption and longer instruction cycle times, making CISC less well-suited for applications where energy efficiency and high performance are key requirements.

Advantages of RISC

RISC (Reduced Instruction Set Computing) processors have several advantages over CISC (Complex Instruction Set Computing) processors:

- Simplicity: RISC processors have a simpler instruction set, which makes them easier to design, manufacture and program. This simplicity also makes them more reliable and less prone to errors.

- Speed: RISC processors have a regular instruction format, which makes them easier to decode and execute. This allows them to have higher clock speeds and greater performance per watt.

- Power efficiency: RISC processors have a simpler and regular instruction set, which makes them more power-efficient. They can perform more operations per watt, which is important for mobile devices and other devices that rely on battery power.

- Code density: RISC processors can handle more instructions per clock cycle which makes the code density higher and more memory efficient.

- Cost-effective: RISC processors are generally less expensive to manufacture than CISC processors, due to their simpler design and fewer transistors required.

- Easy to program: RISC processors are easier to program than CISC processors, as they have a smaller and simpler instruction set. This makes it easier to write efficient and optimized code.

Disadvantages of RISC

RISC (Reduced Instruction Set Computing) processors also have some disadvantages compared to CISC (Complex Instruction Set Computing) processors:

- Complex tasks: RISC processors have a simpler instruction set, which can make it more difficult to perform complex tasks. CISC processors, on the other hand, have specialized instructions for specific tasks, which can make them more efficient for certain types of workloads.

- Code size: RISC processors may require more instructions to perform the same task as a CISC processor, resulting in larger code size and more memory usage.

- Compatibility: RISC processors may not be fully compatible with existing CISC software, requiring new software to be written or existing software to be ported.

- Less efficient: RISC processors may not be as efficient as CISC processors in certain situations, such as when the workload requires a large number of complex operations.

- More hardware is needed: RISC processors may require additional hardware, such as memory management units, to perform certain tasks that are handled by the processor in CISC.

- Limited market: RISC processors have a smaller market share compared to CISC processors, which can make it more difficult to find compatible software and hardware.

Advantages of CISC

- CISC (Complex Instruction Set Computing) processors have several advantages over RISC (Reduced Instruction Set Computing) processors:

- Complex tasks: CISC processors have a more complex instruction set, which allows them to perform complex tasks more efficiently. They have specialized instructions for specific tasks, which can make them more efficient for certain types of workloads.

- Code size: CISC processors may require fewer instructions to perform the same task as a RISC processor, resulting in smaller code size and less memory usage.

- Compatibility: CISC processors are generally more compatible with existing software, as they have a larger and more complex instruction set. This can make it easier to run legacy software on a CISC processor.
- More efficient: CISC processors can be more efficient than RISC processors in certain situations, such as when the workload requires a large number of complex operations.

- Less hardware is needed: CISC processors may require less additional hardware, such as memory management units, to perform certain tasks that are handled by the processor in RISC.

- Wider market: CISC processors have a larger market share compared to RISC processors, which can make it easier to find compatible software and hardware.

Disadvantages of CISC

- CISC (Complex Instruction Set Computing) processors also have some disadvantages compared to RISC (Reduced Instruction Set Computing) processors:

- Complexity: CISC processors have a more complex instruction set, which makes them more difficult to design, manufacture, and program. This complexity also makes them less reliable and more prone to errors.

- Speed: CISC processors often have more complex instruction formats, which can make them slower to decode and execute. This can result in lower clock speeds and lower performance per watt.
- Power efficiency: CISC processors can be less power-efficient than RISC processors, due to their more complex instruction set and larger number of transistors required.
-
- Less Code density: CISC processors can handle fewer instructions per clock cycle, which makes the code density lower and less memory efficient.
-
- Cost: CISC processors are generally more expensive to manufacture than RISC processors, due to their more complex design and larger number of transistors required.
-

- Difficult to program: CISC processors are more difficult to program than RISC processors, as they have a larger and more complex instruction set. This makes it more difficult to write efficient and optimized code.

Applications of RISC & CISC

RISC is used in:

- Embedded systems and microcontrollers, where low power consumption and cost are important factors.
- Mobile devices, where energy efficiency is crucial.
- Workstations and servers, where high performance is necessary.

CISC is mostly used in:

- Mainframes, where complex instructions are necessary to support legacy applications and handle large amounts of data.
- Personal computers, where compatibility with a wide range of software and hardware is important.

RISC- CISC Processors

It's important to note that modern processors are not pure RISC or CISC but a combination of both, these are called as RISC-CISC Hybrid processors which overcomes some of the disadvantages of RISC and CISC processors.RISC-CISC Hybrid processors are a combination of both RISC (Reduced Instruction Set Computing) and CISC (Complex Instruction Set Computing) processors. They are designed to overcome some of the disadvantages of both RISC and CISC processors.

- Performance: RISC-CISC Hybrid processors can offer high performance by combining the efficiency of RISC processors with the functionality of CISC processors. They can handle a large number of complex operations more efficiently than RISC processors alone.

- Compatibility: RISC-CISC Hybrid processors can run both RISC and CISC code, which makes them more compatible with existing software and hardware. This allows them to run legacy software and new software more efficiently.

- Power Efficiency: RISC-CISC Hybrid processors can be more power-efficient than CISC processors, as they can handle more operations per clock cycle. This can result in lower power consumption and longer battery life.

- Cost: RISC-CISC Hybrid processors can be less expensive to manufacture than CISC processors, as they require fewer transistors and a simpler design.

- Code density: RISC-CISC Hybrid processors can handle more instructions per clock cycle, which makes the code density higher and more memory efficient.

- Easier to program: RISC-CISC Hybrid processors are easier to program than CISC processors, as they have a smaller and simpler instruction set. This makes it easier to write efficient and optimized code.

It's important to note that RISC-CISC hybrid processors are not one-size-fits-all solutions and the design of the processor varies from vendor to vendor. They are optimized for specific workloads and are used in a wide range of devices such as smartphones, laptops, servers, and supercomputers.

Implications for future Computer Architecture

The choice between RISC and CISC architecture has implications for the future of computer architecture. Here are a few key points:

- Energy Efficiency: As the demand for energy-efficient computing devices continues to grow, RISC architecture is likely to become increasingly popular due to its simple, efficient design and low power consumption.

- High Performance: For applications that require high performance, RISC architecture will continue to be a popular choice due to its simple instruction set, pipelining, and high clock speeds.

- Embedded Devices: For embedded devices, where low power consumption and a small physical footprint are important, RISC architecture will likely remain the preferred choice.

- Legacy Compatibility: For situations where compatibility with legacy software and hardware is important, CISC architecture may continue to be used, especially in the short term, as it provides a more powerful and flexible architecture.

- Increased Use of GPUs: As the use of GPUs (graphics processing units) for general-purpose computing continues to grow, the line between RISC and CISC architecture may become blurred, as GPUs often use a mixture of both RISC and CISC-style instruction sets.

- Machine Learning: As the field of machine learning continues to grow, new computer architectures that are specifically optimized for machine learning may emerge, using a combination of RISC and CISC-style instruction sets.

CONCLUSION

In conclusion, RISC and CISC are two different types of computer architecture that have their own advantages and disadvantages. RISC architecture simplifies the process of instruction execution, which allows for faster processing speeds, while CISC architecture allows for more complex instructions to be executed, which can result in more efficient code. Both architectures are used in different types of computers and applications, and the choice of architecture depends on the specific needs of the application.

REFERENCE

- https://en.wikipedia.org/wiki/Computer_architecture
- https://developer.arm.com/architectures
- https://riscv.org/