Batch: B-2 Roll No.: 16010122151

Experiment / assignment / tutorial No. 9

TITLE: Study of RISC and CISC Architecture

AIM: Understanding RISC and CISC Architecture

Expected OUTCOME of Experiment: (Mentions the CO/CO's attained)

Books/ Journals/ Websites referred:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, TataMcGraw-Hill.
2. William Stallings, "Computer Organization and Architecture: Designing for Performance", Eighth Edition, Pearson.
3. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization", First Edition, Wiley-India.

Pre Lab/ Prior Concepts:

Reduced Set Instruction Set Architecture (RISC)

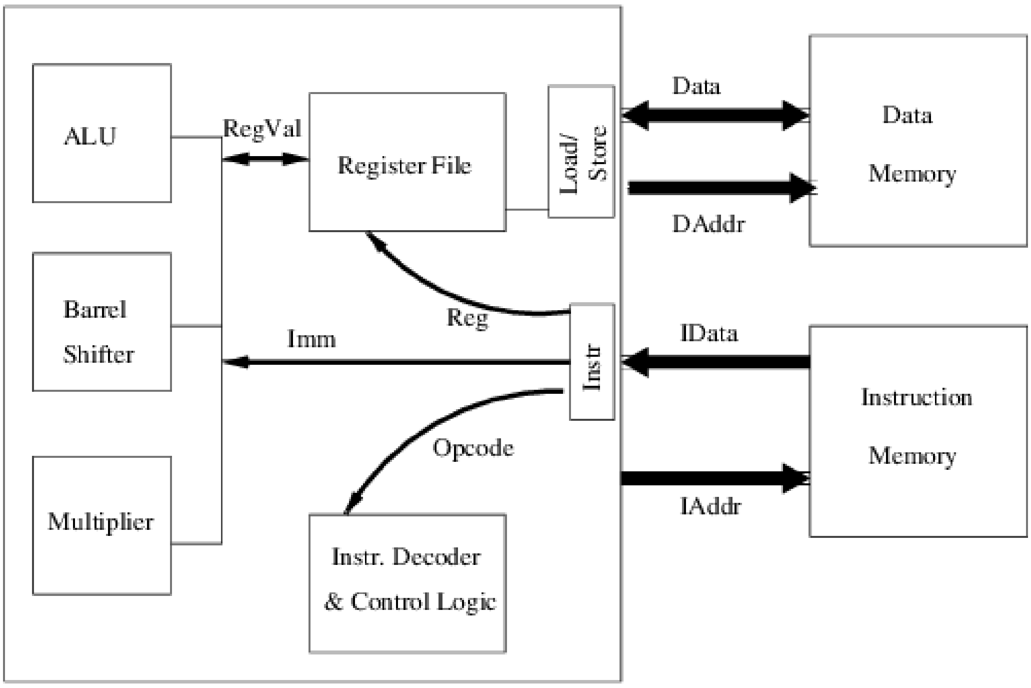
The main idea behind is to make hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating and storing operations just like a load command will load data, store command will store the data.

Complex Instruction Set Architecture (CISC)

The main idea is that a single instruction will do all loading, evaluating and storing operations just like a multiplication command will do stuff like loading data, evaluating and storing it, hence it's complex. Both approaches try to increase the CPU performance.

RISC Architecture

1. Diagram of RISC Architecture:



2. Brief Explanation of each component

RISC stands for Reduced Instruction Set Computing. It is a type of computer architecture that uses a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures. The RISC architecture is characterized by its simplicity, which allows for faster processing speeds and lower power consumption.

Characteristics of RISC include:
- Simpler instructions, hence simple instruction decoding
- More general-purpose registers.
- Simple Addressing Modes.
- Fewer Data types.
- A pipeline can be achieved.

The main components of a RISC architecture include the following:

1. **Registers**: A RISC processor has a large number of registers that are used to store data and instructions. These registers are used to hold the operands for arithmetic and logical operations, as well as the results of these operations.
2. **Load/Store Architecture**: In a RISC architecture, memory access is limited to load and store instructions. This means that data must be loaded from memory into registers before it can be processed, and the results must be stored back into memory after processing.
3. **Pipeline**: A RISC processor uses a pipeline to execute instructions. The pipeline is divided into stages, with each stage performing a specific operation on the instruction. This allows multiple instructions to be executed simultaneously, which increases processing speed.
4. **Fixed Instruction Length**: In a RISC architecture, all instructions are of the same length. This makes it easier to decode and execute instructions, which again increases processing speed.
5. **Simple Addressing Modes**: A RISC processor uses simple addressing modes to access memory. This means that the processor can only access memory using a limited set of addressing modes, which again simplifies the decoding and execution of instructions.

Advantages of RISC:

- Simpler instructions: RISC processors use a smaller set of simple instructions, which makes them easier to decode and execute quickly. This results in faster processing times.
- Lower power consumption: RISC processors consume less power than CISC processors, making them ideal for portable devices.

Disadvantages of RISC:

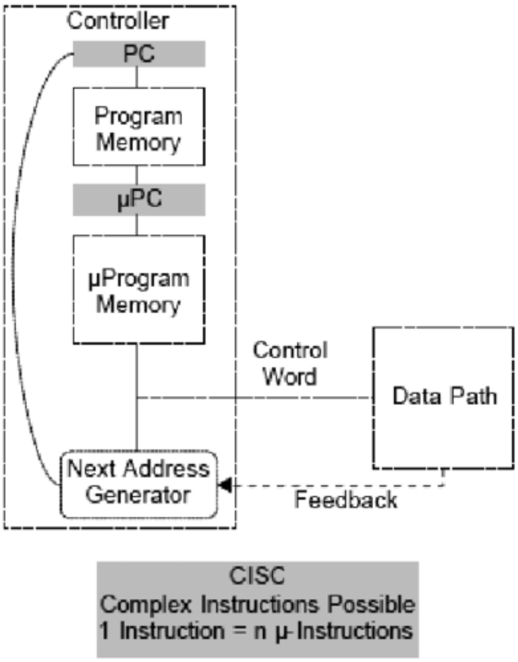
- More instructions required: RISC processors require more instructions to perform complex tasks than CISC processors.
- Increased memory usage: RISC processors require more memory to store the additional instructions needed to perform complex tasks.
- Higher cost: Developing and manufacturing RISC processors can be more expensive than CISC processors.

3. RISC Processor Instruction Set Examples with explanation (Any 2)

1. **LOAD**: This instruction loads data from memory into a register. For example, the instruction LOAD R1, 0(R2) would load data from memory location 0 plus the contents of register R2 into register R1.
2. **STORE**: This instruction stores data from a register into memory. For example, the instruction STORE R1, 0(R2) would store the contents of register R1 into memory location 0 plus the contents of register R2.

CISC Architecture

1. Diagram of CISC Architecture:



2. Brief Explanation of each component

CISC stands for Complex Instruction Set Computing. It is a type of computer architecture that uses a large, complex set of instructions, rather than a more specialized set of instructions often found in other types of architectures. The CISC architecture is characterized by its ability to perform complex operations in a single instruction, which can reduce the number of instructions required to perform a task.

Characteristics of CISC include:
- Complex instruction, hence complex instruction decoding.
- Instructions are larger than one-word size.
- Instruction may take more than a single clock cycle to get executed.
- Less number of general-purpose registers as operations get performed in memory itself.
- Complex Addressing Modes.
- More Data types.

The main components of a CISC architecture include the following:

1. **Registers**: A CISC processor has a smaller number of registers than a RISC processor. These registers are used to hold the operands for arithmetic and logical operations, as well as the results of these operations.
2. **Memory Access**: In a CISC architecture, memory access is not limited to load and store instructions. This means that data can be accessed directly from memory without having to load it into registers first.
3. **Microcode**: A CISC processor uses microcode to execute instructions. Microcode is a low-level code that is used to translate complex instructions into simpler operations that can be executed by the processor.
4. **Variable Instruction Length**: In a CISC architecture, instructions can be of variable length. This allows for more complex instructions to be executed in a single instruction.
5. **Complex Addressing Modes**: A CISC processor uses complex addressing modes to access memory. This means that the processor can access memory using a wide range of addressing modes, which allows for more flexibility in programming.

Advantages of CISC:
- Reduced code size: CISC processors use complex instructions that can perform multiple operations, reducing the amount of code needed to perform a task.
- More memory efficient: Because CISC instructions are more complex, they require fewer instructions to perform complex tasks, which can result in more memory-efficient code.
- Widely used: CISC processors have been in use for a longer time than RISC processors, so they have a larger user base and more available software.

Disadvantages of CISC:
- Slower execution: CISC processors take longer to execute instructions because they have more complex instructions and need more time to decode them.

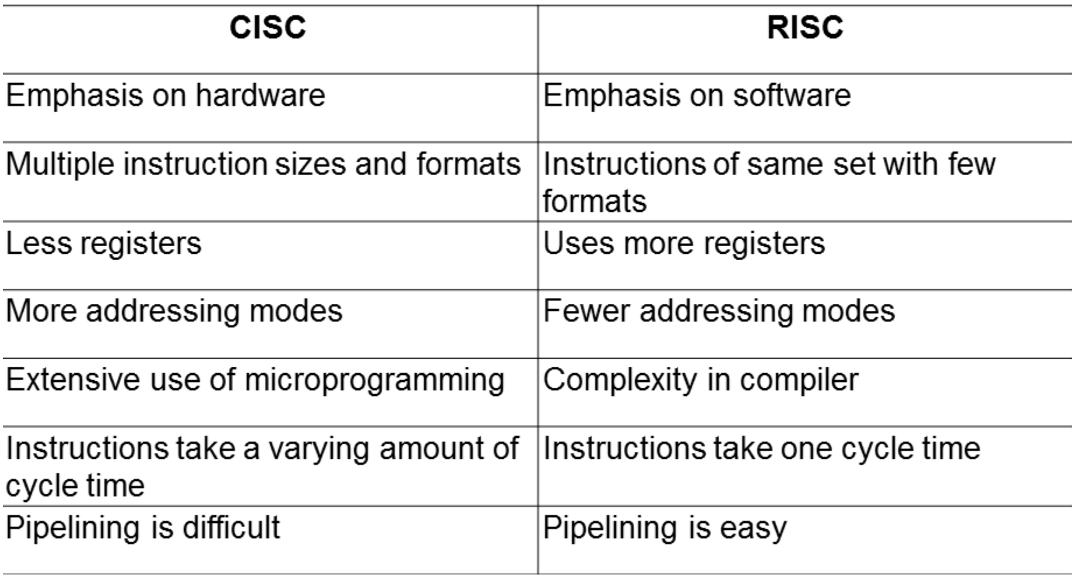
- More complex design: CISC processors have more complex instruction sets, which makes them more difficult to design and manufacture.

3. CISC Processor Instruction Set Examples with explanation (Any 2)

1. **MUL**: This instruction multiplies two operands together and stores the result in a register. For example, the instruction MUL R1, R2 would multiply the contents of registers R1 and R2 together and store the result in register R1.
2. **DIV**: This instruction divides one operand by another and stores the result in a register. For example, the instruction DIV R1, R2 would divide the contents of register R1 by the contents of register R2 and store the result in register R1.

Post Lab Descriptive Questions

Write a tabular comparative analysis of CISC v/s RISC



CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instructions of same set with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

Conclusion:

Date: _09/10/2023_ **Signature of faculty in-charge**