Course Name:	**COA**	**Semester:**	**III**
Date of Performance:	**28-08-2024**	**Batch No:**	**D Division**
Faculty Name:	**Prof. Snehal R. Shinde**	**Roll No:**	**16010123294**
Faculty Sign & Date:		**Grade/Marks:**	**/25**

EXERCISE

Title: **RISC & CISC Work Exercise**

Aim and Objective of the Experiment:

To understand the fundamental differences between RISC (Reduced Instruction Set Computer) and CISC (Complex Instruction Set Computer) architectures, and how these differences influence processor design and performance.

Implementation details:

Definition:

- RISC: A computer architecture that uses a small, highly optimized set of instructions.
- CISC: A computer architecture that uses a large set of instructions, with some instructions performing complex tasks

Characteristics:

- **RISC:**

1. Simple instructions that are executed in a single clock cycle.
2. Large number of general-purpose registers.
3. Load/store architecture (only load and store instructions can access memory).
4. Fixed instruction length.
5. Pipelining is used to increase instruction throughput.

- **CISC:**

1. Complex instructions that can execute multiple low-level operations.
2. Fewer general-purpose registers.
3. Instructions can access memory directly.
4. Variable instruction length.
5. Less emphasis on pipelining.

Examples: Identify two processors that use RISC architecture and two that use CISC architecture. Explain briefly how these processors implement the characteristics of their respective architectures.

- **RISC Processors:**

1. **ARM Cortex-A Series:** Utilizes RISC architecture with a focus on energy efficiency.
2. **MIPS Processors:** Implements a straightforward and efficient instruction set.

- **CISC Processors:**

1. **Intel x86:** Known for its extensive instruction set and backward compatibility.
2. **AMD Ryzen:** Follows CISC architecture, focusing on performance with complex instructions.

Case Study Analysis:

- **Instruction Execution Speed:**

  - **RISC (ARM Cortex-A):** Typically faster due to simple instructions that can be executed in one clock cycle.
  - **CISC (Intel x86):** Slower per instruction, as complex instructions might require multiple cycles.

- **Power Consumption:**

  - **RISC (ARM Cortex-A):** More power-efficient, making it ideal for mobile devices.
  - **CISC (Intel x86):** Generally consumes more power due to the complexity of its instructions.

- **Complexity:**

  - **RISC:** Simpler design with fewer instructions, leading to easier optimization.
  - **CISC:** More complex design with many instructions, making it harder to optimize.

Practical Exercise:

RISC:

```
lw $t0, num1 # Load num1 into register $t0

lw $t1, num2 # Load num2 into register $t1

add $t2, $t0, $t1 # Add $t0 and $t1, store result in $t2

 sw $t2, sum # Store the result in memory
```

CISC:

```
mov ax, [num1] ; Load num1 into register AX

add ax, [num2] ; Add num2 to AX

 mov [sum], ax ; Store the result in memory
```

Explaination:

RISC: Manipulations using registers.

CISC: Manipulations direct in the memory

Critical Thinking Questions:

Which architecture would you prefer for designing a new processor for mobile devices, and why?

Ans: RISC Due to its energy efficiency and simpler design, it's more suitable for mobile devices.

How might advancements in technology affect the future of RISC and CISC architectures?

Ans: Technological advancements may continue to blur the lines between RISC and CISC, with hybrid approaches becoming more common to leverage the benefits of both.

Conclusion:

Understanding the differences between RISC and CISC is crucial for designing processors suited to specific applications, especially in terms of performance, power efficiency, and complexity.

Submission Deadline: 28/08/2024