Fundamentals

From the architecture point of view, the microprocessor chips can be classified into two categories:

1. Complex Instruction Set Computers (CISC) and
2. Reduce Instruction Set Computers (RISC) .

INTRODUCTION

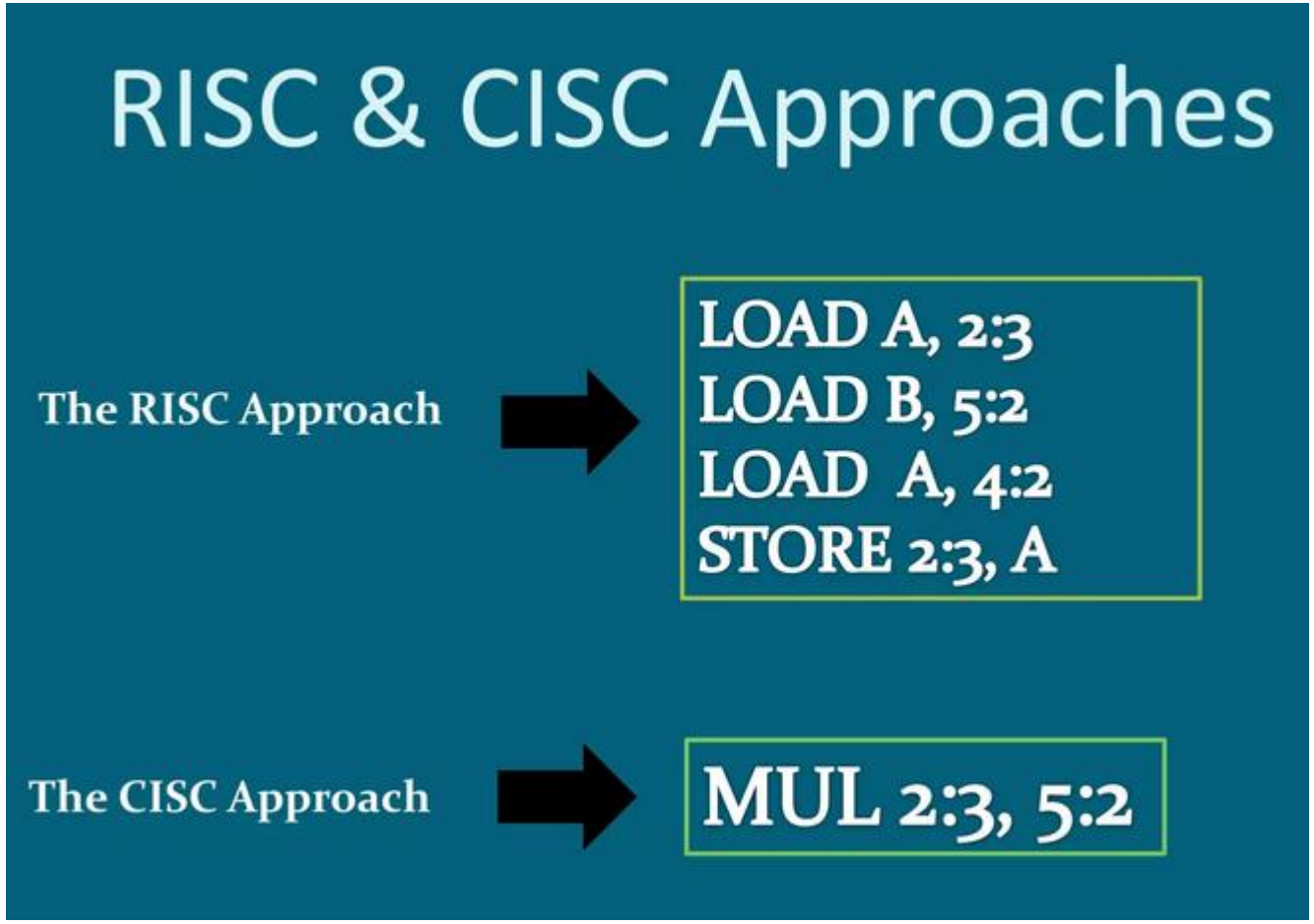
- RISC – Reduced Instruction Set Computer
- RISC is a type of microprocessor architecture that utilizes
  - a small, highly-optimized set of instructions
  - rather than a more specialized set of instructions.
- The main alternative for RISC is CISC ,which stands for complex instruction set computer.
- CISC is the older approach, that came about to maximize performance of earlier computer's. Where instructions were executed sequentially.

The instruction set is the hardware language that tells the processor what to do.

CHARACTERSTICS OF RISC

- **Simplified instructions** , taking 1 clock cycle.
- Large no. of **general purpose registers**.
- **Circuit** is much **simpler**.
- Fast to decode.
- Fast to execute.
- **Pipelining-** fetching of next instruction while previous instruction executes.

RISC & CISC Approaches



INSTRUCTION

CISC

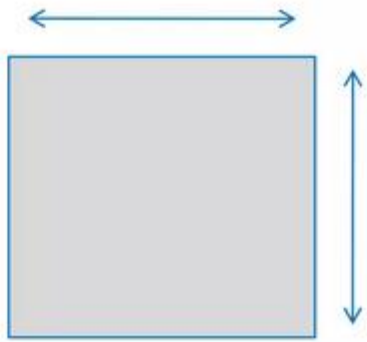
- Complex Instructions.
- ADD AX,[BX + SI + 600H]
- Many operations in single instruction.

RISC

- Simpler or reduced instructions.
- LOAD R1, addresss1
  LOAD R2, address2
  ADD R1, R2
  STORE address1, R1
- One instruction one operation.

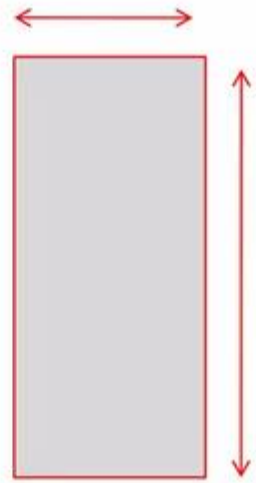
CODE SIZE

CISC



Code size is smaller but complicated.

RISC



Code size is larger but simpler.

REGISTERS AND ADDRESSING MODE

CISC

- Fewer register.
- These registers are designed for special purposes.
- CISC designs provide a large number of addressing modes.

RISC

- Large number of registers.
- Here registers are identical so any register can be used for any purpose.
- RISC designs have single addressing modes.

CISC

- Slower to execute.
- Difficult to decode.
- Instruction size varies in different instructions.
- Complex circuit design.

RISC

- Faster execution.
- Easy to decode.
- Same instruction size in every instructions.
- Circuit design is simpler.

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock, complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Slower since instruction can take more than 1 cycle	Faster since instructions usually take 1 instruction cycle
Main objective is less code.	Main objective is speed.

CISC	RISC
More hardware oriented.	More software oriented since the compiler deals with translations.
Instruction size is mostly varied in size.	Instruction size is always a set size.
Addressing Modes can be complex	Addressing Modes are simple.

RISC and CISC Architecture

Reduced Instruction Set Computer or RISC Architecture

The fundamental goal of RISC is to make hardware simpler by employing an instruction set that consists of only a few basic steps used for evaluating, loading, and storing operations. A load command loads data but a store command stores data.

Characteristics of RISC:

1. It has simpler instructions and thus simple instruction decoding.

2. More general-purpose registers.

3. The instruction takes one clock cycle in order to get executed.

4. The instruction comes under the size of a single word.

5. Pipeline can be easily achieved.

6. Few data types.

7. Simpler addressing modes.

RISC and CISC Architecture

Characteristics of CISC:

1. Instructions are complex, and thus it has complex instruction decoding.
2. The instructions may take more than one clock cycle in order to get executed.
3. The instruction is larger than one-word size.
4. Lesser general-purpose registers since the operations get performed only in the memory.
5. More data types.
6. Complex addressing modes.

Performance Equations

Both CISC and RISC approaches primarily try to increase the performance of a CPU. Here is how both of these work:

1. CISC: This kind of approach tries to minimize the total number of instructions per program, and it does so at the cost of increasing the total number of cycles per instruction.

2. RISC: It reduces the cycles per instruction and does so at the cost of the total number of instructions per program.

$$\text{CPU Time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instructions}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

PERFORMANCE EQUATION

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

The CISC approach attempts to **minimize the number of instructions per program**, sacrificing the number of cycles per instruction.

RISC does the opposite, **reducing the cycles per instruction** at the cost of the number of instructions per program.