Quadrant II – Notes

Programme **:** Bachelor of Science (Second Year)
Subject : Computer Science
Semester : IV
Paper Code : CSC104
Paper Title : Computer Organization and Operating Systems
Title of the Unit **:** Basic Computer Organization and Design
Module Name **:** RISC, CISC Architectures
Module Number **:** 14
Name of the Presenter: Mrs. Avani Anil Kharde

Before we discuss the differences between the RISC and CISC architecture let us know about the concepts of RISC and CISC

RISC and CISC Processors

- What is RISC?
- RISC Architecture
- Characteristic of RISC
- Advantages of RISC Architecture
- Disadvantages of RISC Architecture
- Examples of RISC processors

What is RISC?

A reduced instruction set computer is a computer that only uses simple commands that can be divided into several instructions that achieve low-level operation within a single CLK cycle, as its name proposes “Reduced Instruction Set”.

The RISC is a Reduced Instruction Set Computer microprocessor, and its architecture includes a set of instructions that are highly customized.

The main function of this is to reduce the time of instruction execution by limiting as well as optimizing the number of commands. So each command cycle uses a single clock cycle where every clock cycle includes three parameters namely fetch, decode & execute.

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.

The kind of processor is mainly used to execute several difficult commands by merging them into simpler ones. RISC processor needs a number of transistors to design and it reduces the instruction time for execution.

The best examples of RISC processors include PowerPC, SUN’s SPARC, RISC-V, Microchip PIC processors, etc.

RISC Architecture
The term RISC stands for ‘’Reduced Instruction Set Computer’’. It is a CPU design plan based on simple orders and acts fast.

This is a small or reduced set of instructions. Here, every instruction is expected to attain very small jobs. In this machine, the instruction sets are modest and simple, which help in comprising more complex commands. Each instruction is of a similar length; these are wound together to get compound tasks done in a single operation. Most commands are completed in one machine cycle. This pipelining is a crucial technique used to speed up RISC machines.

RISC Processor Architecture (Block diagram)

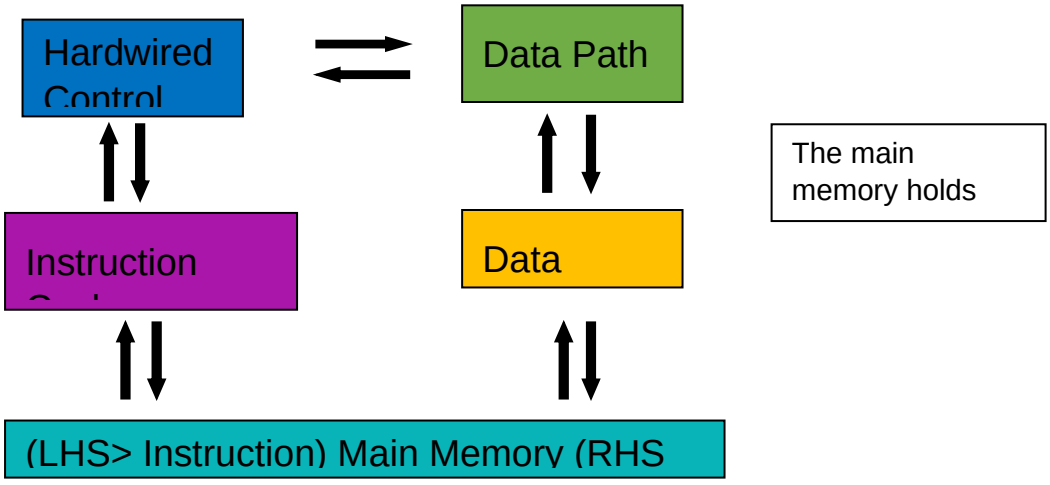
RISC processor is implemented using the hardwired control unit**.** The hardwired control unit produces **control signals** which regulate the working of processors hardware. RISC architecture emphasizes on using the **registers** rather than memory.

This is because the registers are the 'fastest' available memory source. The registers are physically small and are placed on the same chip where the ALU and the control unit are placed on the processor. The RISC instructions **operate** on the operands present in **processor's registers**.

Observe one thing here, we **don't** have the "microprogram control store" or the "control memory" like we have seen in the CISC architecture in our previous content.

It is just because all instructions in RISC are simple and execute one instruction per cycle. So, here the **instructions are hardwired** and there is no need for control store. For each operation, we will have as defined hardwire. Making an instruction hardwired is making a function or operation in instruction **permanent** using connected circuits.

Below we have the block diagram for the RISC architecture.



Characteristic of RISC –

1. Simpler instruction, hence simple instruction decoding.
2. Instruction come under size of one word.
3. Instruction take single clock cycle to get executed.
4. More number of general purpose register.
5. Simple Addressing Modes.
6. Less Data types.
7. Pipeline can be achieved.

Advantages of RISC Architecture

- The performance of RISC processors is often two to four times than that of CISC processors because of simplified instruction set.
- This architecture uses less chip space due to reduced instruction set. This makes to place extra functions like floating point arithmetic units or memory management units on the same chip.
- The per-chip cost is reduced by this architecture that uses smaller chips consisting of more components on a single silicon wafer.
- RISC processors can be designed more quickly than CISC processors due to its simple architecture.
- The execution of instructions in RISC processors is high due to the use of many registers for holding and passing the instructions as compared to CISC processors.

Disadvantages of RISC Architecture

- The performance of a RISC processor depends on the code that is being executed. The processor spends much time waiting for first instruction result before it proceeds with next subsequent instruction, when a compiler makes a poor job of scheduling instruction execution.

    - RISC processors require very fast memory systems to feed various instructions. Typically, a large memory cache is provided on the chip in most RISC based systems.
- RISC instruction size is reduced but more instructions are required to perform an operation when compared with CISC. So, we can say that the length of the program is increased.
- The machine instructions are **hardwired** in RISC so, it would **cost** if any instruction needs modification.
- It finds is **difficulty** in processing **complex instruction and complex addressing mode**.
- RISC instructions do not allow **direct** memory to memory transfer, it requires Load and Store instructions to do so.
    -

Examples of RISC processors

MIPS, SPARC, IBM POWER instruction set, Alpha, RISC-V, ARM architecture.

What is CISC?
It was developed by the Intel Corporation and it is Complex Instruction Set Computer. This processor includes a huge collection of simple to complex instructions. These instructions are specified in the level of assembly language level and the execution of these instructions takes more time.

A complex instruction set computer is a computer where single instructions can perform numerous low-level operations like a load from memory, an arithmetic operation, and a memory store or are accomplished by multi-step processes or addressing modes in single instructions, as its name proposes “Complex Instruction Set ”.

So, this processor moves to decrease the number of instructions on every program & ignore the number of cycles for each instruction. It highlights to assemble complex instructions openly within the hardware as the hardware is always as compared with software. However, CISC chips are relatively slower as compared to RISC chips but

utilize small instruction as compare with RISC. The best examples of the CISC processor include AMD, VAX, System/360 & Intel x86.

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.

CISC Architecture
The term CISC stands for ''Complex Instruction Set Computer''. It is a CPU design plan based on single commands, which are skilled in executing multi-step operations.

CISC computers have small programs. It has a huge number of compound instructions, which takes a long time to perform. Here, a single set of instructions is protected in several steps; each instruction set has additional than 300 separate instructions. Maximum instructions are finished in two to ten machine cycles. In CISC, instruction pipelining is not easily implemented.

CISC Processors Architecture

As we have studied above the main objective of a CISC processor is to **minimize** the program size by **reducing** the **number of instructions** in a program. This is done by 'embedding some of the low-level instructions in a single complex instruction'. Later when decoded this instruction generates several microinstructions to execute.

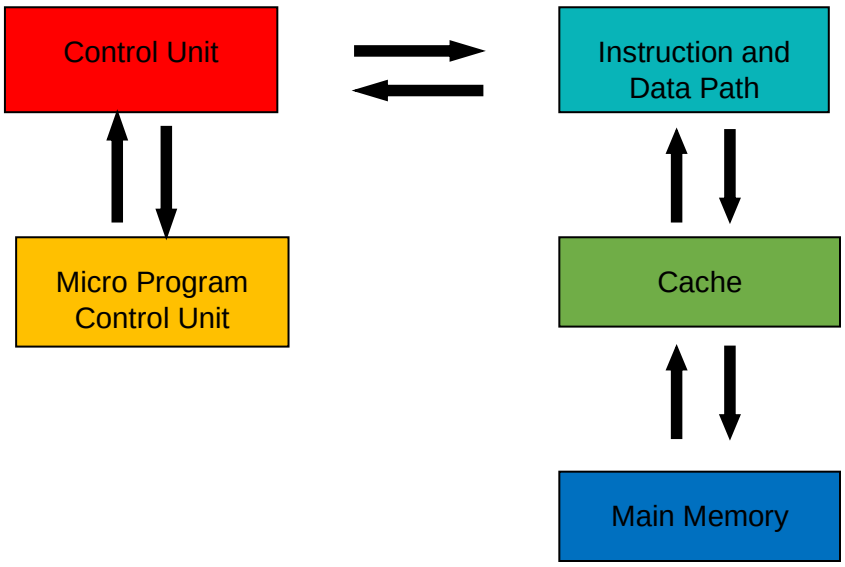
Now if a program/software is getting simplified then the hardware has to get on work and must be able to perform the complex tasks. That's why a CISC processor has complex hardware. The block diagram of CISC architecture is given below:

Here, you have a special **microprogram control unit** that uses a series of microinstructions of the microprogram stored in the "control memory" of the *microprogram control unit* and generate the **control signals**.

The **control units** access the **control signals** produced by the *microprogram control unit* & operate the functioning of processors hardware.

Instruction and data path fetches the opcode and operands of the instructions from the memory.

Cache and **main memory** is the location where the program instructions and operands are stored.



Examples of CISC processor

1. IBM 370/168
2. VAX 11/780
3. Intel 386, 486, Pentium, Pentium Pro, Pentium II, Pentium III
4. Motorola's 68000, 68020, 68040, etc.

Characteristic of CISC –

1. Complex instruction, hence complex instruction decoding.
2. Instruction are larger than one word size.
3. Instruction may take more than single clock cycle to get executed.
4. Less number of general purpose register normally from 5 to 20 as operation get performed in memory itself.
5. Complex Addressing Modes.
6. More Data types.

7. The code length is very short, so it needs extremely small RAM.
8. It highlights the instruction on hardware while designing as it is faster to design than the software.
9. It gives simple programming within assembly language.

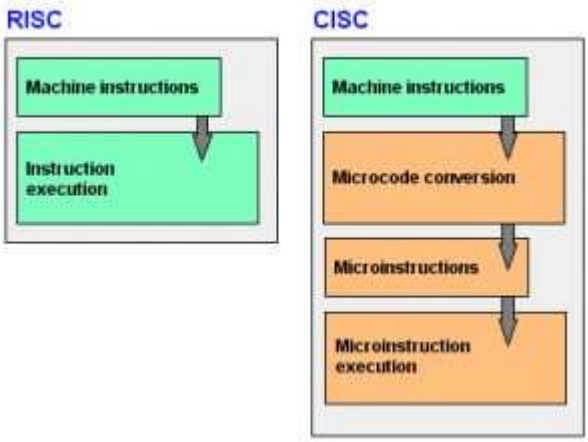
Advantages of CISC Architecture

- Microprogramming is easy to implement and much less expensive than hard wiring a control unit.
- It is easy to add new commands into the chip without changing the structure of the instruction set as the architecture uses general-purpose hardware to carry out commands.
- This architecture makes the efficient use of main memory since the complexity (or more capability) of instruction allows to use less number of instructions to achieve a given task.
- The compiler need not be very complicated, as the micro program instruction sets can be written to match the constructs of high level languages.

Disadvantages of CISC Architecture

- A new or succeeding versions of CISC processors consists early generation processors in their subsets (succeeding version). Therefore, chip hardware and instruction set became complex with each generation of the processor.
- The overall performance of the machine is reduced because of slower clock speed.
- The pipeline execution within the CISC will make it difficult to use.
- The performance of this processor may change based on the executed code because the next commands may depend on the earlier instruction for their implementation within a cycle.
- These processors need very quick memory to keep different instructions that use a huge collection of cache memory to react to the command within less time.

Difference between RISC and CISC Architecture



Difference between RISC and CISC

RISC	CISC
1. RISC stands for Reduced Instruction Set Computer.	1. CISC stands for Complex Instruction Set Computer.
2. RISC processors have simple instructions taking about one clock cycle. The average clock cycle per instruction (CPI) is 1.5	2. CSIC processor has complex instructions that take up multiple clocks for execution. The average clock cycle per instruction (CPI) is in the range of 2 and 15.
3. Performance is optimized with more focus on software	3. Performance is optimized with more focus on hardware.
4. It has no memory unit and uses separate hardware to implement instructions..	4. It has a memory unit to implement complex instructions.
5. It has a hard-wired unit of programming.	5. It has a microprogramming unit.
6. The instruction set is reduced i.e. it has only a few instructions in the instruction set. Many of these instructions are very primitive.	**6.** The instruction set has a variety of different instructions that can be used for complex operations.
7. The instruction set has a variety of different instructions that can be used for complex operations.	**7.** CISC has many different addressing modes and can thus be used to represent higher-level programming language statements more efficiently.
8. Complex addressing modes are synthesized using the software.	8. CISC already supports complex addressing modes
9. Multiple register sets are present	9. Only has a single register set
10. RISC processors are highly pipelined	10. They are normally not pipelined or less pipelined

11. The complexity of RISC lies with the compiler that executes the program	11. The complexity lies in the microprogram
12. Execution time is very less	12. Execution time is very high
13. Code expansion can be a problem	13. Code expansion is not a problem
14. The decoding of instructions is simple.	14. Decoding of instructions is complex
15. It does not require external memory for calculations	15. It requires external memory for calculations
16. The most common RISC microprocessors are Alpha, ARC, ARM, AVR, MIPS, PA-RISC, PIC, Power Architecture, and SPARC.	16. Examples of CISC processors are the System/360, VAX, PDP-11, Motorola 68000 family, AMD, and Intel x86 CPUs.
17. RISC architecture is used in high-end applications such as video processing, telecommunications, and image processing.	17. CISC architecture is used in low-end applications such as security systems, home automation, etc.

Conclusion:

This is all about the RISC processor and its instruction set architecture. RISC architecture is now used worldwide in cellular telephones, computer tables and even supercomputers.

This session discusses the concepts of RISC, CISC, and differences. When the first microprocessors, as well as microcontrollers, were introduced, there is no better and suitable architecture. Once these processors were implemented, the CISC architecture is used mostly due to the lack of software support in the RISC processor. This is mainly doing to build all their hardware as well as software back

well-suited through their first 8086 processors. We hope that you have got a better understanding of this concept.

There is no better architecture.

When microprocessors and microcontroller were first being introduced, they were mostly CISC mainly because of the lack of software support for RISC. Later a few companies started delving into the RISC architecture such as Apple.After a few decades, the CISC architecture was becoming difficult to improve and develop. Intel however had a lot of resources and were able to overcome most of the major roadblocks. They were doing this mainly to make all their hardware and software back compatible with their initial 8086 processors.

Currently, the boundary between RISC and CISC architectures are very blurred as both hardware and software support for RISC and CISC are readily available.

****