|  |
| --- |
|  |

Bottom of Form

# CISC vs RISC: Difference Between Architectures, Instruction Set

## What is CISC?

CISC was developed to make compiler development easier and simpler. The full form of CISC is Complex Instruction Set Computer. They are chips that are easy to program that makes efficient use of memory.

CISC eliminates the need for generating machine instructions to the processor. For example, instead of having to make a compiler, write lengthy machine instructions to calculate a square-root distance, a CISC processor offers a built-in ability to do this.

Many of the early computing machines were programmed in assembly language. Computer memory was slow and expensive. CISC was commonly implemented in such large computers, such as the PDP-11 and the DEC system.

In this tutorial, you will learn:

* [What is CISC?](https://www.guru99.com/risc-vs-cisc-differences.html#1)
* [What is RISC?](https://www.guru99.com/risc-vs-cisc-differences.html#2)
* [Characteristics of CISC](https://www.guru99.com/risc-vs-cisc-differences.html#3)
* [Characteristics of RISC](https://www.guru99.com/risc-vs-cisc-differences.html#4)
* [Difference between CISC and RISC](https://www.guru99.com/risc-vs-cisc-differences.html#5)
* [CISC Advantages](https://www.guru99.com/risc-vs-cisc-differences.html#6)
* [RISC Advantages](https://www.guru99.com/risc-vs-cisc-differences.html#7)
* [CISC Disadvantages](https://www.guru99.com/risc-vs-cisc-differences.html#8)
* [RISC Disadvantages](https://www.guru99.com/risc-vs-cisc-differences.html#9)

## What is RISC?

RISC is designed to perform a smaller number of types of computer instruction. Hence, it can operate at a higher speed. The full form of RISC is Reduced Instruction Set Computers. It is a microprocessor that is designed to perform smaller number of computer instruction so that it can operate at a higher speed.

RISC instruction sets hold less than 100 instructions and use a fixed instruction format. This method uses a few simple addressing modes that use a register-based instruction. In this compiler development mechanism, LOAD/STORE is the only individual instructions for accessing memory.

## KEY DIFFERENCES:

* In CISC, the instruction set is very large that can be used for complex operations while in RISC the instruction set is reduced, and most of these instructions are very primitive.
* CISC computer's execution time is very high whereas RISC computer's execution time is very less.
* In, CISC code expansion is not a problem while in RISC code expansion may create a problem.
* In CISC, decoding of instructions is complex whereas, in RISC, the decoding of instructions is simple.
* CISC requires external memory for calculations, but RISC requires external memory for calculations.
* CISC has only a single register set while RISC has multiple register sets are present.

## Characteristics of CISC

Here, are important characteristics Of CISC

One instruction is needed to support multiple addressing modes.

* A large number of instructions.
* Instruction-decoding logic will be complex.
* Instructions for special tasks used infrequently.
* A large variety of addressing modes
* It offers variable-length instruction formats.
* Instruction 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 operation get performed in memory itself.
* Various CISC designs are set up with two special registers for the stack pointer for managing interrupts

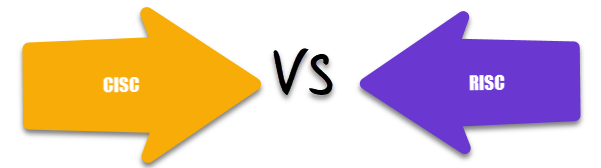
## Characteristics of RISC

Here, are an important characteristic of RICS:

* Simpler instruction decoding
* A number of general-purpose registers.
* Simple Addressing Modes
* Fewer Data types.
* A pipeline can be achieved
* One instruction per cycle
* Register-to-register operations
* Simple instruction format
* Instruction execution would be faster
* Smaller Programs

## Difference between CISC and RISC

Here, are important differences between CISC vs. RISC

[](https://www.guru99.com/images/2/022220_0604_CISCvsRISCD1.png)

|  |  |
| --- | --- |
| **CISC** | **RISC** |
| It has a microprogramming unit. | It has a hard-wired unit of programming. |
| The instruction set has various different instructions that can be used for complex operations. | The instruction set is reduced, and most of these instructions are very primitive. |
| Performance is optimized with emphasis on hardware. | Performance is optimized which emphasis on software |
| Only single register set | Multiple register sets are present |
| They are mostly less or not pipelined | This type of processors are highly pipelined |
| Execution time is very high | Execution time is very less |
| Code expansion is not a problem. | Code expansion may create a problem. |
| Decoding of instructions is complex. | The decoding of instructions is simple. |
| It requires external memory for calculations | It doesn't require external memory for calculations |
| Examples of CISC processors are the System/360, VAX, AMD, and Intel x86 CPUs. | Common RISC microprocessors are ARC, Alpha, ARC, ARM, AVR, PA-RISC, and SPARC. |
| Single-cycle for each instruction | Instructions can take several clock cycles |
| Heavy use of RAM (can cause bottlenecks if RAM is limited) | More efficient use of RAM than RISC |
| Simple, standardized instructions | Complex and variable-length instructions |
| A small number of fixed-length instructions | A large number of instructions |
| Limited addressing modes | Compound addressing modes |
| Important applications are Security systems, Home automation. | Important applications are : Smartphones, PDAs. |
| Varying formats (16-64 bits for each instruction). | fixed (32-bit) format |
| Unified cache for instructions and data. | Separate data and instruction cache. |

## CISC Advantages

Here, are pros/benefits of CISC

* In CISC it is easy to add new commands into the chip without need to change the structure of the instruction set
* This architecture allows you to make efficient use of main memory
* The compiler should not be very complicated, as with the case of CISC. The instruction sets can be written to match the structures of high-level languages.

## RISC Advantages

Here, are pros/benefits of RISC

* Complex and efficient machine instructions.
* It offers extensive addressing capabilities for memory management.
* Relatively few registers when compared with RISC processors
* It helps you to reduce the instruction set.
* Offers limited addressing schemes for memory operands

## CISC Disadvantages

Here, are Cons/ Drawbacks of CISC

* Earlier generations of a processor family mostly contained as a subset in every new version. Hence, instruction set & chip hardware becomes complex with each generation of computers.
* The performance of the machine slows down because of clock time taken by different instructions will never be similar.
* They are larger as they require more transistors

## RISC Disadvantages

Here, are Cons/Drawbacks of RISC

* The performance of the RISC processors depends on the programmer or compiler. Compiler plays an important role while converting the CISC code to a RISC code
* RISC processors have large memory caches on the chip itself.
* RISC architecture necessitates on-chip hardware to be continuously reprogrammed.

Top of Form

*vs.* 

Bottom of Form

# SRAM vs. DRAM

**RAM**, or **random access memory**, is a kind of computer memory in which any byte of memory can be accessed without needing to access the previous bytes as well. RAM is a volatile medium for storing digital data, meaning the device needs to be powered on for the RAM to work. DRAM, or Dynamic RAM, is the most widely used RAM that consumers deal with. [DDR3](https://www.diffen.com/difference/DDR2_vs_DDR3) is an example of DRAM.

**SRAM**, or **static RAM**, offers better performance than DRAM because DRAM needs to be refreshed periodically when in use, while SRAM does not. However, SRAM is more expensive and less dense than DRAM, so SRAM sizes are orders of magnitude lower than DRAM.

## Comparison chart

| Dynamic random-access memory versus Static random-access memory comparison chart | | |
| --- | --- | --- |
| **[Edit this comparison chart](https://www.diffen.com/difference/Special:EditTable?diffenVal1=Dynamic+random-access+memory&diffenVal2=Static+random-access+memory)** | **Dynamic random-access memory** | **Static random-access memory** |
| **Introduction (from Wikipedia)** | Dynamic random-access memory is a type of random-access memory that stores each bit of data in a separate capacitor within an integrated circuit. | Static random-access memory is a type of semiconductor memory that uses bistable latching circuitry to store each bit. The term static differentiates it from dynamic RAM (DRAM) which must be periodically refreshed. |
| **Typical applications** | Main memory in a computer (e.g. DDR3). Not for long-term storage. | L2 and L3 cache in a CPU |
| **Typical sizes** | 1GB to 2GB in smartphones and tablets; 4GB to 16GB in laptops | 1MB to 16MB |
| **Place Where Present** | Present on motherboard. | Present on Processors or between Processor and Main Memory. |

## Different Kinds of Memory Explained

The following video explains the different types of memory used in a computer — DRAM, SRAM (such as used in a processor's L2 cache) and NAND flash (e.g. used in an SSD).

## Structure and Function

The structures of both types of RAM are responsible for their main characteristics as well as their respective pros and cons. For a technical, in-depth explanation of how DRAM and SRAM work, see this [engineering lecture from the University of Virginia](https://www.youtube.com/watch?v=SO4i3rKkLIE).

### Dynamic RAM (DRAM)

Each memory cell in a DRAM chip holds one bit of data and is composed of a transistor and a capacitor. The transistor functions as a switch that allows the control circuitry on the memory chip to read the capacitor or change its state, while the capacitor is responsible for holding the bit of data in the form of a 1 or 0.

In terms of function, a capacitor is like a container that stores electrons. When this container is full, it designates a 1, while a container empty of electrons designates a 0. However, capacitors have a leakage that causes them to lose this charge, and as a result, the “container” becomes empty after just a few milliseconds.

Thus, in order for a DRAM chip to work, the CPU or memory controller must recharge the capacitors that are filled with electrons (and therefore indicate a 1) before they discharge in order to retain the data. To do this, the memory controller reads the data and then rewrites it. This is called refreshing and occurs thousands of times per second in a DRAM chip. This is also where the “Dynamic” in Dynamic RAM originates, since it refers to the refreshing necessary to retain the data.

Because of the need to constantly refresh data, which takes time, DRAM is slower.

### Static RAM (SRAM)

Static RAM, on the other hand, uses [flip-flops](https://en.wikipedia.org/wiki/Flip-flop_%28electronics%29), which can be in one of two stable states that the support circuitry can read as either a 1 or a 0. A flip-flop, while requiring six transistors, has the advantage of not needing to be refreshed. The lack of a need to constantly refresh makes SRAM faster than DRAM; however, because SRAM needs more parts and wiring, an SRAM cell takes up more space on a chip than a DRAM cell does. Thus, SRAM is more expensive, not only because there is less memory per chip (less dense) but also because they are harder to manufacture.

[DNA vs. RNA](https://www.diffen.com/difference/DNA_vs_RNA)

### 

### Speed

Because SRAM does not need to refresh, it is typically faster. The average access time of DRAM is about 60 nanoseconds, while SRAM can give access times as low as 10 nanoseconds.

## Capacity and Density

Because of its structure, SRAM needs more transistors than DRAM to store a certain amount of data. While a DRAM module only requires one transistor and one capacitor to store every bit of data, SRAM needs 6 transistors. Since the number of transistors in a memory module determines its capacity, for a similar number of transistors, a DRAM module can have up to 6 times more capacity than an SRAM module.

## Power Consumption

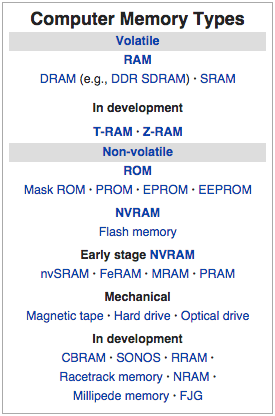
Typically, an SRAM module consumes less power than a DRAM module. This is because SRAM only requires a small steady current while DRAM requires bursts of power every few milliseconds to refresh. This refresh current is several orders of magnitude greater than the low SRAM standby current. Thus, SRAM is used in most portable and battery-operated equipment.

However, the power consumption of SRAM does depend on the frequency at which it is accessed. When SRAM is used at a slower pace, it draws nearly negligible power while idled. On the other hand, at higher frequencies, SRAM can consume as much power as DRAM.

## Price

SRAM is much more expensive than DRAM. A gigabyte of SRAM cache costs around $5000, while a gigabyte of DRAM costs $20-$75. Since SRAM uses flip-flops, which can be made of up to 6 transistors, SRAM needs more transistors to store 1 bit than DRAM does, which only uses a single transistor and capacitor. Thus, for the same amount of memory, SRAM requires a higher number of transistors, which increases the production cost.

## Applications

[](https://www.diffen.com/difference/Image:Computer-Memory-types.png)

[magnify](https://www.diffen.com/difference/Image:Computer-Memory-types.png)

Computer memory types

Like all RAM, DRAM and SRAM are volatile and therefore cannot be used to store "permanent" data such as operating systems or data files like pictures and spreadsheets.

The most common application of SRAM is to serve as cache for the processor (CPU). In processor specifications, this is listed as L2 cache or L3 cache. SRAM performance is really fast but SRAM is expensive, so typical values of L2 and L3 cache are 1MB to 8MB.

The most common application of DRAM — such as DDR3 — is volatile storage for computers. While not as fast as SRAM, DRAM is still very fast and can connect directly to the CPU bus. Typical sizes of DRAM are about 1 to 2GB in smartphones and tablets, and 4 to 16GB in laptops.

## Introduction to Instruction Cycle

The instruction cycle is related to the CPU (Central processing unit), a list of steps that follows when the computer system starts until the computer system is shut down. The other name of the instruction cycle is the fetch-decode-execute cycle. There are basically four phases in the instruction cycle: fetching an instruction from memory, decoding of the fetched instruction, reading of the address from memory, and the last phase include the instruction execution. The computer processor executes it.

### What is the Instruction Cycle?

It is defined as the basic cycle carried out in a computer system in which the computer system fetched the instruction from memory, decodes the instruction and then executes the instruction. It is also known as Fetch-Execute-Cycle. In the computer system, all the instructions are executed in the RAM of the computer system. The CPU is responsible for executing the instruction. The CPU system first fetches the data and instruction from the main memory and store in the temporary memory, which is known as registers. This phase is known as the fetch cycle. After fetching an instruction from memory, the next step taken by the CPU is decoding of fetched instruction. This phase is known as the decode phase.

The CPU contains an instruction set which contain all the predefined list of instructions. In the last phase of the instruction cycle, data processing takes place. The instruction executes in this phase, and the result is stored in another register. After the fetch-decode-execute cycle, the CPU reset itself for another instruction cycle. The CPU system is considered the basic operation cycle, which is carried out in RAM of the central processing unit and executes the instruction. It is repeatedly continuously in the CPU when the computer system boots and then shut down. It is executed in a sequential manner means when one instruction cycle is completed, then only the next instruction cycle begins. But in the modern era central processing unit, the instructions can be executed in a parallel manner.

There are basically five stages of the instruction cycle which are described below:

#### 1. Initiating Cycle

In this phase, the computer system boots up, and the operating system loads in the main memory (read-only memory) of the central processing unit. It immediately begins when the computer system starts.

#### 2. Fetching of Instruction

The fetching of instruction is the first phase. The fetch instruction is common for each instruction executes in a central processing unit. In this phase, the central processing unit sends the PC to MAR and then sends the READ command into a control bus. After sending a read command on the data bus, the memory returns the instruction, which is stored at that particular address in the memory. Then, the CPU copies data from the data bus into MBR and then copies the data from MBR to registers. After all this, the pointer is incremented to the next memory location so that the next instruction can be fetched from memory.

Popular Course in this category

[Software Testing Training (9 Courses, 2 Projects)9 Online Courses | 2 Hands-on Projects | 60+ Hours | Verifiable Certificate of Completion | Lifetime Access   
4.5 (6,057 ratings)](https://www.educba.com/software-development/courses/software-testing-course/?btnz=edu-blg-inline-banner1" \t "_blank)

[Course Price  
$29](https://www.educba.com/software-development/courses/software-testing-course/?btnz=edu-blg-inline-banner1" \t "_blank) ~~[$399](https://www.educba.com/software-development/courses/software-testing-course/?btnz=edu-blg-inline-banner1" \t "_blank)~~ [View Course](https://www.educba.com/software-development/courses/software-testing-course/?btnz=edu-blg-inline-banner1" \t "_blank)

Related Courses

[Penetration Testing Training Program (2 Courses)](https://www.educba.com/software-development/courses/penetration-testing-certification/?btnz=edu-blg-inline-banner1)[TestNG Training (4 Courses, 2 Project)](https://www.educba.com/software-development/courses/testng-training/?btnz=edu-blg-inline-banner1)

#### 3. Decoding of **Instruction**

The decoding of instruction is the second phase. In this phase, the CPU determines which instruction is fetched from the instruction and what action needs to be performed on the instruction. The opcode for the instruction is also fetched from memory and decodes the related operation which needs to be performed for the related instruction.

#### 4. Read of an Effective Address

The reading of an effective address is the third phase. This phase deals with the decision of the operation. The operation can be of any type of memory type non-memory type operation. Memory instruction can be categorized into two categories direct memory instruction and indirect memory instruction.

#### 5. Executing of Instruction

The executing of instruction is the last phase. In this stage, the instruction is finally executed. The instruction is executed, and the result of the instruction is stored in the register. After the execution of an instruction, the CPU prepares itself for the execution of the next instruction. For every instruction, the execution time is calculated, which is used to tell the processing speed of the processor.

### Why we Need an Instruction Cycle?

* There is a need for an instruction cycle for the computer system so that proper understanding can be done to understand the flow of instructions and execution of an instruction in a computer processor.
* It deals with the complete flow of instruction when the computer system boots up until the computer system is shut down. By the instruction cycle, the internal flow of the central processing unit can be better understand so that if there is an issue, it can be easily resolved.
* It deals with basic operations of computer processor there is a need for proper understanding of various stages involved in it.
* The fetch-decode-execute cycle is common for all types of instructions for the computer processor system.

### Importance of Instruction Cycle

* It is important for the processor system for the central processing unit as the instructions are the basic operations that are performed in the main memory of the central processing unit.
* It is a set of steps that help to understand the flow of instruction. By the instruction cycle, the end to end the flow of instructions can be visualized in the computer processor.
* It is common for all instruction set it needs to properly understand so that all the operations can be performed easily.
* By the instruction cycle, the processing time of the program can be easily calculated, which helps to determine the speed of the processor.
* As the speed of the processor tells how many instructions can be simultaneously executed in the central processing unit.

### Conclusion

The instruction cycle is the basic operation cycle related to the computer system, which deals with the basic operation in the central processing unit. The other name of the instruction cycle is fetched decode execute cycle. It is continuously repeated until the system goes into the shutdown phase. It is common for all types of instructions defined for computer processor system.

### Recommended Articles

This is a guide to What is Instruction Cycle? Here we discuss the introduction; why we need an instruction cycle? And importance. You may also have a look at the following articles to learn more –

1. [Cache Memory Types](https://www.educba.com/cache-memory-types/)
2. [Types of Primary Memory](https://www.educba.com/thread-life-cycle-in-java/)
3. [What is CPU Register](https://www.educba.com/what-is-cpu-register/)