

Difference between ARM and Intel x86-64

With ARM chips now ubiquitous in high-performance laptops and edge servers, it's reasonable to ask whether these new devices are worth buying.

It's time to understand the technical differences between ARM processors and their x86-64 predecessors.

What does x86-64 mean?

Intel processors found in desktops and laptops are sometimes called 'x86' processors. The 64-bit architecture that all modern computers use today is now on the x86 model, now known as 'x86-64'.

This all sounds a bit confusing but it helps explain why there are two different **Program Files folders on Windows**.

Name	Date modified	Type
\$SysReset	10/14/2023 2:33 AM	File folder
\$WinREAgent	7/1/2024 9:25 PM	File folder
OneDriveTemp	5/3/2024 5:12 AM	File folder
PerfLogs	5/7/2022 8:24 AM	File folder
Program Files	6/23/2024 8:38 PM	File folder
Program Files (x86)	6/23/2024 8:43 PM	File folder

All 32-bit applications will be in the x86 folder and 64-bit applications will be in another folder. This has been the convention since Microsoft released the 64-bit version of Windows XP in 2001.

To simplify things a bit, we'll call 'x86-64' 'x86' because there are enough similarities between 32-bit and 64-bit chips to group them together.

The 'x86' label comes from the grandfather of the modern processor industry, a microprocessor created by Intel called the 8086. This 16-bit chip operates on an instruction set that is still used by modern processors. Today it is mostly used to perform advanced computing tasks.

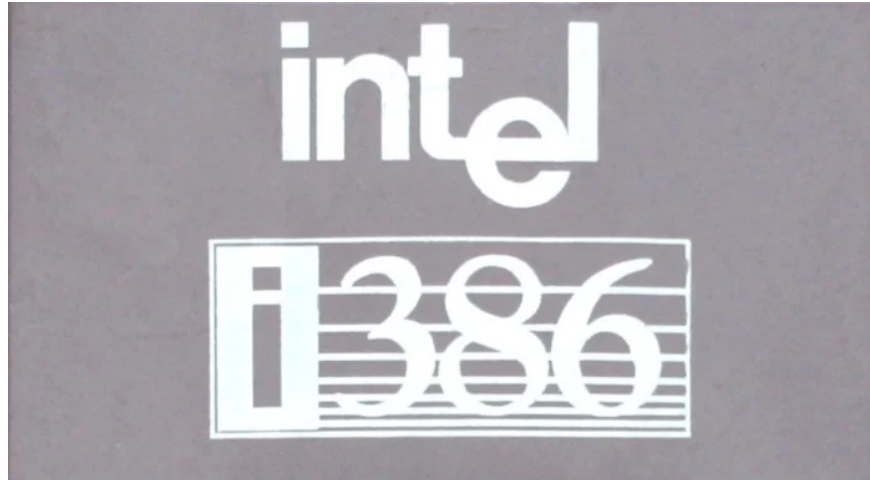
Everything from text editors to advanced video rendering software is still capable of operating under the architecture that dates back to 1978!

Since the release of the 8086 microprocessor, Intel continued to use the naming convention for future models such as the 80286 and 80386 until breaking with tradition by releasing the Pentium line. However, we still pay homage to those old chips by classifying all subsequent processors that use its instruction set as 'x86'.

Even the more modern 64-bit processors that AMD first released in the Opteron line in the 2000s are similarly named, as are the majority of 64-bit processors in desktop and server systems. continues to be a mainstay today.

Scripts

All processor functions are based on what is called an instruction set. This is a hard-wired set of rudimentary operations that a chip can perform. Each command tells the chip to do things like basic math and move data around. At the end of every piece of code in every program you run on your machine is a layer that communicates with the processor. This layer also uses its own code called assembly language, which translates one step deeper down to the basic instructions often called machine code or machine language.



This is where we realize the core difference between ARM and x86 chips.

x86 chips use CISC (Complex Instruction Set Computing), while ARM chips use RISC (Reduced Instruction Set Computing). While CISC chips try to put more 'work' into a single instruction while sacrificing clock cycles to complete this task, RISC chips have a much smaller instruction set that breaks things down into steps. simpler, can be executed in a single clock cycle.

CISC vs. RISC

If a CISC chip needs to multiply two numbers, it can run a single instruction: `MULT 2, 3`. This single instruction will consist of loading numbers from memory, multiplying them together and saving the result in the correct memory location. A RISC chip performing the same operation would require more steps. First, the `LOAD` command to move numbers from the register unit to the execution unit. Then a `PROD` instruction to multiply the numbers. Finally, the `STOR` command to place the result in the correct register.



Although the CISC chip may seem more efficient because its commands seem simple, keep in mind some important differences:

1. First, CISC chips execute instructions in multiple clock cycles, while each instruction in RISC chips is executed in one clock cycle. Therefore, a hypothetical `MULT` instruction can run in the same number of cycles as a set of RISC instructions that perform the same task.
2. Second, all CISC commands and logic need to be stored in the transistor. RISC chips can use fewer transistors because they need to store fewer instructions.
3. Third, the lower number of transistors required by RISC allows for less power usage.

There are several advantages to CISC. First, the computer doesn't need to do much work to convert from a human-level programming language like `C` to a processor-level assembly language. In fact, the `MULT` command mentioned above is very similar to the `C` command `foo = foo * bar`.

RISC chips require the compiler to do more work to convert the code to assembly language. CISC chips can also perform their operations directly on the system memory while RISC chips have to get data from memory into processor registers before working with it.

In a head-to-head performance competition between the two, for consumers, there is no clear winning option. However, RISC has one more trump card to really gain an advantage over CISC.

Energy consumption

Between CISC and RISC, there is absolutely no competition in terms of power usage. RISC will definitely gain the advantage here. The reduced instruction set reduces wafer area, making the chip smaller and therefore less power-hungry.



Small chips with fewer microscopic components also have less resistance to handle and don't require as much power to run. The best case study for this is the proven technology of the smartphone, a 'light computing' device that delivers tremendous performance despite its relatively small size but can still sometimes survive for more than 24 hours on a relatively small battery.

That being said, this doesn't mean that all computers will start using ARM chips to reduce power consumption. In the desktop market, where systems often rely on a direct connection to a wall outlet instead of a battery, power efficiency is not such a big priority. CISC chips still perform admirably in these situations, and there's really no sign of this changing any time soon.

Should I buy an ARM or x86 system?

With Microsoft also releasing Windows 11 for ARM architecture and Apple switching Macs to using ARM chips, you may also want to buy an ARM PC. However, it breaks things down to a very simple question: Are you trying to squeeze every bit of performance out of every watt-hour of power provided by the battery, or are you inclined to push as much raw performance through device and harm battery life?

The answer depends on each use case. Finally, just make sure you don't forget to bring your charging cable with you!

You finished reading the article "**Difference between ARM and Intel x86-64**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.