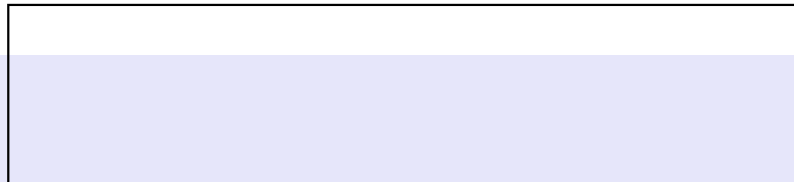


Computers are general  
purpose tools built around a  
micro processor

## How Microprocessors Work

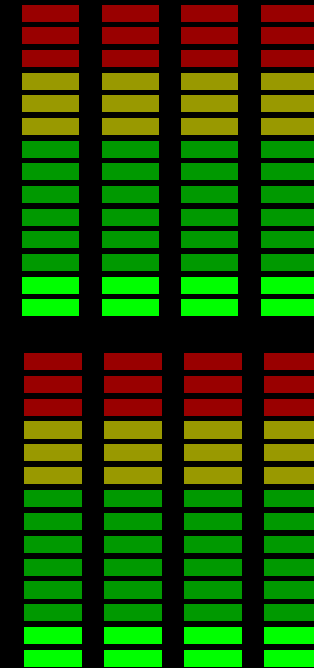
by [Marshall Brain](#)

If you have ever wondered what the microprocessor in your computer is doing, or if you have ever wondered about the differences between types of microprocessors, then read on. In this article, you will learn how fairly simple digital logic techniques allow a computer to do its job, whether its playing a game or spell checking a document!

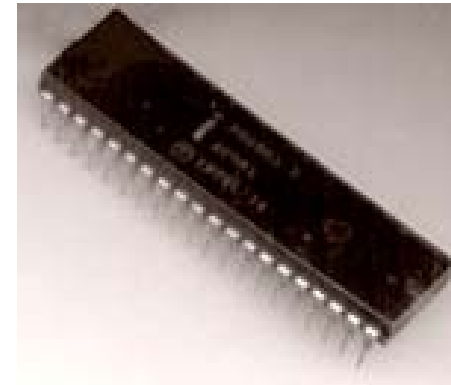


## What's a Chip?

A **chip** is also called an **integrated circuit**. Generally it is a small, thin piece of [silicon](#) onto which the [transistors](#) making up the microprocessor have been etched. A chip might be as large as an inch on a side and can contain tens of millions of transistors. Simpler processors might consist of a few thousand transistors etched onto a chip just a few millimeters square.



The first microprocessor to make it into a home computer was the Intel 8080, a complete 8-bit computer on one chip, introduced in 1974. The first microprocessor to make a real splash in the market was the Intel 8088, introduced in 1979 and incorporated into the IBM PC (which first appeared around 1982). If you are familiar with the PC market and its history, you know that the PC market moved from the 8088 to the 80286 to the 80386 to the 80486 to the Pentium to the Pentium II to the Pentium III to the Pentium 4. All of these microprocessors are made by Intel and all of them are improvements on the basic design of the 8088. The Pentium 4 can execute any piece of code that ran on the original 8088, but it does it about 5,000 times faster!



**Intel 8080**



<b>Name</b>	<b>Date</b>	<b>Transistors</b>	<b>Microns</b>	<b>Clock speed</b>	<b>Data width</b>	<b>MIPS</b>
8080	1974	6,000	6	2 MHz	8 bits	0.64
8088	1979	29,000	3	5 MHz	16 bits 8-bit bus	0.33
80286	1982	134,000	1.5	6 MHz	16 bits	1
80386	1985	275,000	1.5	16 MHz	32 bits	5
80486	1989	1,200,000	1	25 MHz	32 bits	20
Pentium	1993	3,100,000	0.8	60 MHz	32 bits 64-bit bus	100
Pentium II	1997	7,500,000	0.35	233 MHz	32 bits 64-bit bus	~300
Pentium III	1999	9,500,000	0.25	450 MHz	32 bits 64-bit bus	~510
Pentium 4	2000	42,000,000	0.18	1.5 GHz	32 bits 64-bit bus	~1,700
Pentium 4 "Prescott"	2004	125,000,000	0.09	3.6 GHz	32 bits 64-bit bus	~7,000

- **Transistors** is the number of transistors on the chip. You can see that the number of transistors on a single chip has risen steadily over the years.
- **Microns** is the width, in microns, of the smallest wire on the chip. For comparison, a human hair is 100 microns thick. As the feature size on the chip goes down, the number of transistors rises.
- **Clock speed** is the maximum rate that the chip can be clocked at. Clock speed will make more sense in the next section.

- The **date** is the year that the processor was first introduced. Many processors are re-introduced at higher clock speeds for many years after the original release date.
- **Transistors** is the number of transistors on the chip. You can see that the number of transistors on a single chip has risen steadily over the years.
- **Microns** is the width, in microns, of the smallest wire on the chip. For comparison, a human hair is 100 microns thick. As the feature size on the chip goes down, the number of transistors rises.
- **Clock speed** is the maximum rate that the chip can be clocked at. Clock speed will make more sense in the next section.

**Data Width** is the width of the ALU. An 8-bit ALU can add/subtract/multiply/etc. two 8-bit numbers, while a 32-bit ALU can manipulate 32-bit numbers. An 8-bit ALU would have to execute four instructions to add two 32-bit numbers, while a 32-bit ALU can do it in one instruction. In many cases, the external data bus is the same width as the ALU, but not always. The 8088 had a 16-bit ALU and an 8-bit bus, while the modern Pentiums fetch data 64 bits at a time for their 32-bit ALUs.

**MIPS** stands for "millions of instructions per second" and is a rough measure of the performance of a CPU. Modern CPUs can do so many different things that MIPS ratings lose a lot of their meaning, but you can get a general sense of the relative power of the CPUs from this column.

## Inside a Microprocessor

To understand how a microprocessor works, it is helpful to look inside and learn about the logic used to create one. In the process you can also learn about **assembly language** -- the native language of a microprocessor -- and many of the things that engineers can do to boost the speed of a processor.

A microprocessor executes a collection of machine instructions that tell the processor what to do. Based on the instructions, a microprocessor does three basic things:

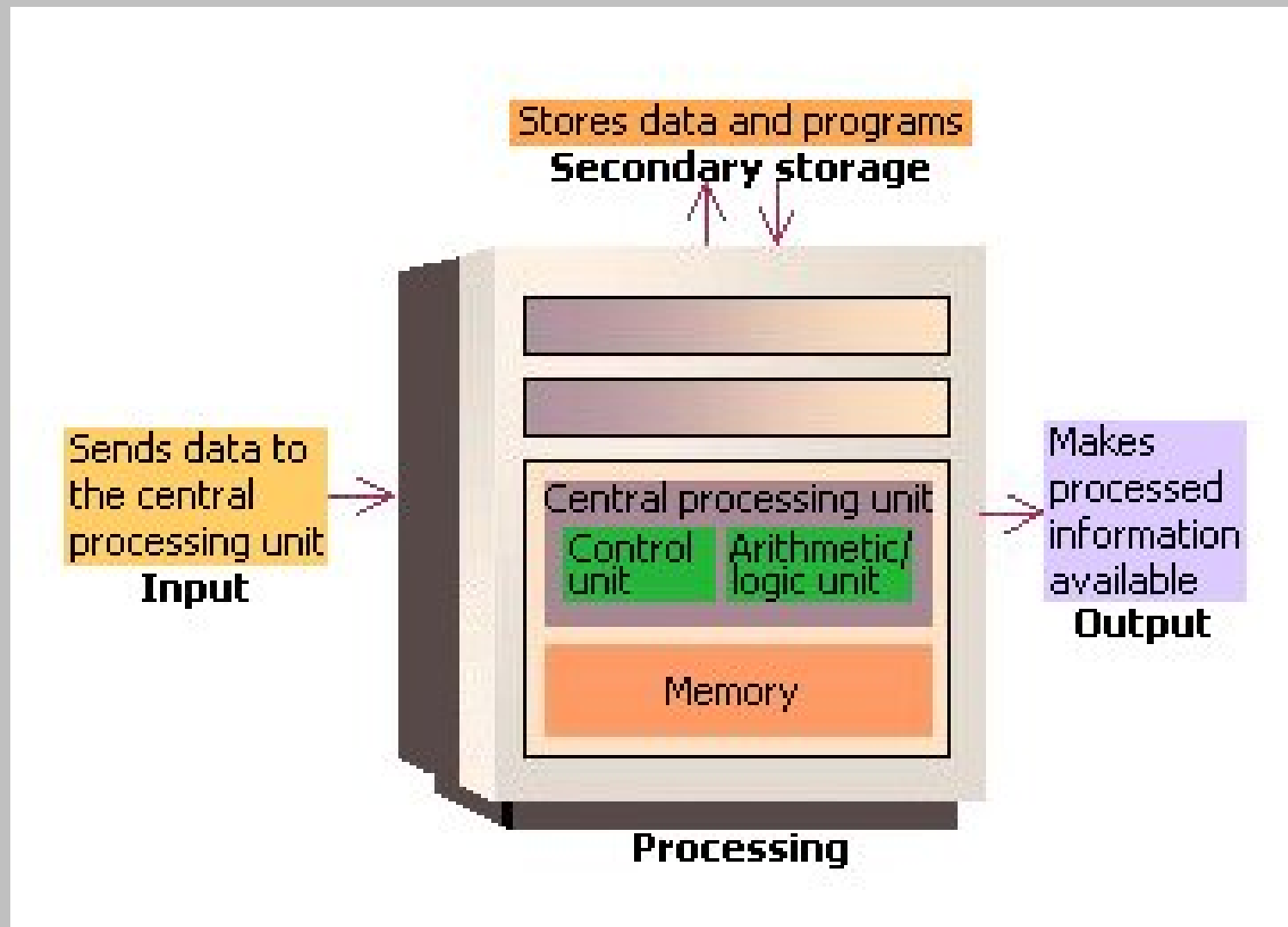


[Microprocessor](#)  
**Intel Pentium 4 processor**

- Using its ALU (Arithmetic/Logic Unit), a microprocessor can perform mathematical operations like addition, subtraction, multiplication and division. Modern microprocessors contain complete floating point processors that can perform extremely sophisticated operations on large floating point numbers.
- A microprocessor can move data from one [memory](#) location to another.
- A microprocessor can make decisions and jump to a new set of instructions based on those decisions.

There may be very sophisticated things that a microprocessor does, but those are its three basic activities. The following diagram shows an extremely simple microprocessor capable of doing those three things:





## **The Arithmetic/Logic Unit**

The arithmetic/logic unit (ALU) contains the electronic circuitry that executes all arithmetic and logical operations.

The arithmetic/logic unit can perform four kinds of arithmetic operations, or mathematical calculations: addition, subtraction, multiplication, and division. As its name implies, the arithmetic/logic unit also performs logical operations. A logical operation is usually a comparison. The unit can compare numbers, letters, or special characters. The computer can then take action based on the result of the comparison. This is a very important capability. It is by comparing that a computer is able to tell, for instance, whether there are unfilled seats on airplanes, whether charge- card customers have exceeded their credit limits, and whether one candidate for Congress has more votes than another.

Logical operations can test for three conditions:

**Equal-to condition.** In a test for this condition, the arithmetic/logic unit compares two values to determine if they are equal. For example: If the number of tickets sold equals the number of seats in the auditorium, then the concert is declared sold out.

**Less-than condition.** To test for this condition, the computer compares values to determine if one is less than another. For example: If the number of speeding tickets on a driver's record is less than three, then insurance rates are \$425; otherwise, the rates are \$500.

**Greater-than condition.** In this type of comparison, the computer determines if one value is greater than another. For example: If the hours a person worked this week are greater than 40, then multiply every extra hour by 1.5 times the usual hourly wage to compute overtime pay.

## **Registers: Temporary Storage Areas**

Registers are temporary storage areas for instructions or data. They are not a part of memory; rather they are special additional storage locations that offer the advantage of speed. Registers work under the direction of the control unit to accept, hold, and transfer instructions or data and perform arithmetic or logical comparisons at high speed. The control unit uses a data storage register the way a store owner uses a cash register-as a temporary, convenient place to store what is used in transactions.

. Computers usually assign special roles to certain registers, including these registers:

- **An accumulator**, which collects the result of computations.
- **An address register**, which keeps track of where a given instruction or piece of data is stored in memory. Each storage location in memory is identified by an address, just as each house on a street has an address.
- **A storage register**, which temporarily holds data taken from or about to be sent to memory.

A general-purpose **register**, which is used for several functions