***History of Processors***

* In the early **1950s**, each computer design was unique. There were no upward-compatible machines or computer architectures with multiple, differing implementations. Programs written for one machine would run on no other kind, even other kinds from the same company. This was not a major drawback then because no large body of software had been developed to run on computers, so starting programming from scratch was not seen as a large barrier.

* By the end of the 1950s, commercial builders had developed factory-constructed, truck-deliverable computers. The most widely installed computer was the IBM 650, which used drum memory onto which programs were loaded using either paper punched tape or punched cards. Some very high-end machines also included core memory which provided higher speeds. Hard disks were also starting to grow popular.

Picture from WIkipedia

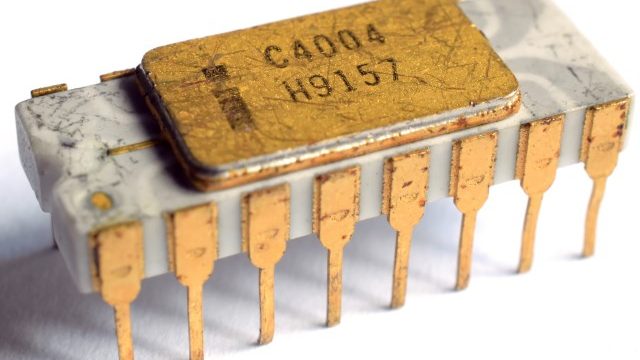
* In 1962, IBM tried a new approach to designing computers. The plan was to make a family of computers that could all run the same software, but with different performances, and at different prices. As users' needs grew, they could move up to larger computers, and still keep all of their investment in programs, data and storage media.

Picture from WIkipedia

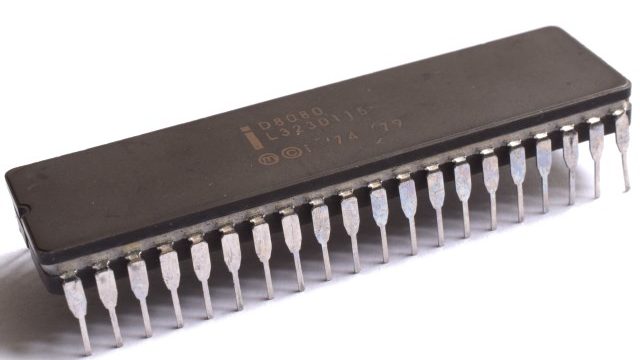
* To do this, they designed one reference computer named System/360. This was a virtual computer, a reference instruction set, and abilities that all machines in the family would support. To provide different classes of machines, each computer in the family would use more or less hardware emulation, and more or less micro program emulation, to create a machine able to run the full S/360 instruction set. IBM chose consciously to make the reference [instruction set](https://en.wikipedia.org/wiki/Instruction_set) quite complex, and very capable. Even though the computer was complex, its [control store](https://en.wikipedia.org/wiki/Control_store) holding the [micro program](https://en.wikipedia.org/wiki/Microprogram) would stay relatively small, and could be made with very fast memory.
* In the 1960s, the development of electronic [calculators](https://en.wikipedia.org/wiki/Calculator), electronic [clocks](https://en.wikipedia.org/wiki/Clock), the [Apollo guidance computer](https://en.wikipedia.org/wiki/Apollo_guidance_computer), and [Minuteman missile](https://en.wikipedia.org/wiki/Minuteman_missile), helped make [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuit) economical and practical. In the late 1960s, the first calculator and clock chips began to show that very small computers might be possible with [large-scale integration](https://en.wikipedia.org/wiki/Large-scale_integration) (LSI). This culminated in the invention of the [microprocessor](https://en.wikipedia.org/wiki/Microprocessor), a single-chip CPU.

Picture from ExtremeTech.com

Picture from ExtremeTech.com



* The [Intel 4004](https://en.wikipedia.org/wiki/Intel_4004), released in 1971, was the first commercial microprocessor. It used [binary coded decimal](https://en.wikipedia.org/wiki/Binary_coded_decimal) (BCD), and was released by [Busicom](https://en.wikipedia.org/wiki/Busicom) and Intel. In March 1972, Intel introduced a microprocessor with an [8-bit](https://en.wikipedia.org/wiki/8-bit) architecture, the [8008](https://en.wikipedia.org/wiki/8008), an integrated [PMOS](https://en.wikipedia.org/wiki/PMOS_logic) re-implementation of the [transistor–transistor logic](https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor_logic) (TTL) based [Datapoint 2200](https://en.wikipedia.org/wiki/Datapoint_2200) CPU.
*  The 8080 was the basis for the [8086](https://en.wikipedia.org/wiki/8086), which is a direct ancestor to today's ubiquitous [x86](https://en.wikipedia.org/wiki/X86) family.

The Intel 8008 traded some clock speed -- it ran at 500 KHz, as opposed to the 4004's 720 KHz -- for enhanced capabilities. While somewhat slower, it could operate on eight bits of data at a time, rather than the four-bit limitation of the 4004. Initially commissioned before the 4004, the chip was late to market and the company that initially ordered it, Computer Terminal Corporation, later decided not to use the design. The company agreed to allow Intel to market the chip independently, and the 8008 became a commercial success.

Picture from ExtremeTech.com

The Intel 8080 was a 1974 design that followed the 8008 and drastically increased its clock speed, from 500 KHz to 2MHz (later chips at up to 3.125MHz would be released). The 8080 was the first chip Faggin designed from scratch, and it used the n-channel MOS process (the 4004 and 8008 had used p-channel MOS), with fewer support chips required and 6000 transistors in the base design. Intel, back then, was known more for its memory than its CPUs, but the 8080 was a huge success in the budding microcomputer market.

Picture from ExtremeTech.com

* **The x86 processors**
* In the early 1980s, researchers at [UC Berkeley](https://en.wikipedia.org/wiki/UC_Berkeley) and [IBM](https://en.wikipedia.org/wiki/IBM) both discovered that most computer language compilers and interpreters used only a small subset of the instructions of [complex instruction set computing](https://en.wikipedia.org/wiki/Complex_instruction_set_computing) (CISC). Much of the power of the CPU was being ignored in real-world use. They realized that by making the computer simpler and less orthogonal, they could make it faster and less costly at the same time, (including [Pentium](https://en.wikipedia.org/wiki/Pentium) and [Core i7](https://en.wikipedia.org/wiki/Intel_Core#Core_i7)). Every instruction of the 8080 has a direct equivalent in the large x86 instruction set, although the op code values are different in the latter. At the same time, CPU calculation became faster in relation to the time for needed memory accesses. A common variant on the RISC design employs the [Harvard architecture](https://en.wikipedia.org/wiki/Harvard_architecture), versus [Von Neumann architecture](https://en.wikipedia.org/wiki/Von_Neumann_architecture) or stored program architecture common to most other designs. In a Harvard Architecture machine, the program and data occupy separate memory devices and can be accessed simultaneously. In Von Neumann machines, the data and programs are mixed in one memory device, requiring sequential accessing which produces the so-called [Von Neumann bottleneck](https://en.wikipedia.org/wiki/Von_Neumann_architecture#Von_Neumann_bottleneck).

### Intel 80286

* ****The Intel 80286 was the successor to the 8086 (the 80186 was mostly used in embedded markets). It drove IBM's then-new PC/AT platform and was available in 4MHz, 6MHz, 8MHz, and 12.5MHz varieties. The 80286 packed 134,000 transistors and was substantially faster than the 8086 clock-for-clock thanks to de-multiplexed address and data buses, a dedicated adder, and a hardware-based multiplier. In some cases, the 80286 was up to twice as fast as the older 8086

Picture from ExtremeTech.com

### Intel 80386C:\Users\Ashutosh Chauhan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Intel-80386-640x360.jpg

The 80386 was a substantial step forward for Intel's CPU designs. Built on 1.5-micron and 1-micron technology, it was Intel's first 32-bit x86 processor and the first Intel processor that could theoretically address up to 4GB of memory. It expanded and extended the 80286's Protected Mode and included a new virtual 8086 mode that allowed the 386 to emulate multiple 8086 chips simultaneously. This allowed the 386 to execute programs that could only run in Real Mode while using a Protected Mode operating system. The 80386 also supported a flat memory model while in Protected Mode that allowed programs to treat memory as a single, contiguous address space, even though the CPU itself actually used a segmented memory model.

Picture from ExtremeTech.com

### C:\Users\Ashutosh Chauhan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Intel-486-DX2-640x360.jpgIntel 80486

* The 80486 launched in 1989, three years after the initial debut of the 80386. It was the first Intel CPU to contain over a million transistors, the first Intel x86 chip with an on-die L1 cache, and the first tightly-pipelined x86 core (a tight pipeline is one in which each stage performs its operations within the same time slot). The 80486 was the first Intel chip to implement a clock multiplier, in which the CPU runs at multiples of the base bus speed, and the first Intel chip that could regularly complete simple instructions within a single clock cycle.

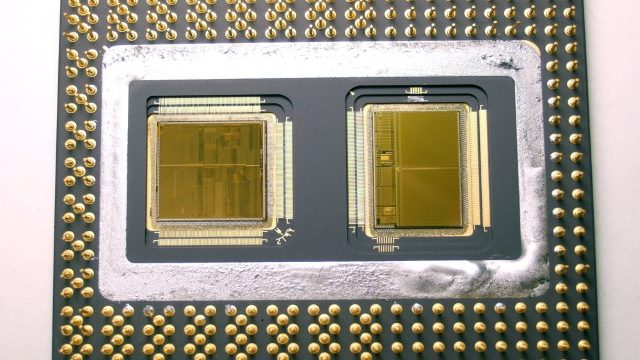
Picture from ExtremeTech.com

### C:\Users\Ashutosh Chauhan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pentium-60-640x360.jpgIntel Pentium and Pentium MMX

* The Pentium debuted in 1993 as Intel's first non-numerical x86 processor brand. The new CPU included dual integer pipelines, meaning it could sometimes complete up to two instructions per clock cycle (this is referred to as a superscalar architecture). It used a 64-bit external bus instead of the 486's 32-bit bus and it offered separate instruction and data caches to boost performance. FPU performance was also dramatically improved. Intel would later introduce the Pentium MMX, with support for new multimedia instructions (MMX technology), larger caches, and higher overall performance.

Picture from ExtremeTech.com

### Intel Pentium Pro

* The Intel Pentium Pro is the great-great-grandfather of virtually every consumer CPU Intel has built over the past 21 years. It was the first Intel chip to translate x86 instructions into internal micro-ops and to reorder those translated operations for optimal execution within the CPU. Today, this is a common ****technique known as out-of-order execution, but it was radical at the time and cost Intel a significant amount of die space and power consumption compared with the Pentium.

Picture from ExtremeTech.com

* **[](https://www.elprocus.com/wp-content/uploads/2014/03/513.jpg)Current Generation Processors (64 bit Processors)**

### Xeon

* Xeon processor is a 400 MHz Pentium processor from the Intel for use in workstations and enterprise servers. This processor is designed for multimedia applications, engineering graphics, Internet and large data base servers.

Picture from Elprocus.com

* **Features of current generation Processors**

### Multi-core

Multi-core CPUs are typically multiple CPU cores on the same die, connected to each other via a shared L2 or L3 cache, an on-die [bus](https://en.wikipedia.org/wiki/Computer_bus), or an on-die [crossbar switch](https://en.wikipedia.org/wiki/Crossbar_switch). All the CPU cores on the die share interconnect components with which to interface to other processors and the rest of the system. These components may include a [front side bus](https://en.wikipedia.org/wiki/Front_side_bus) interface, a [memory controller](https://en.wikipedia.org/wiki/Memory_controller) to interface with [dynamic random access memory](https://en.wikipedia.org/wiki/Dynamic_random_access_memory) (DRAM), a [cache coherent](https://en.wikipedia.org/wiki/Cache_coherency) [link](https://en.wikipedia.org/wiki/Computer_bus) to other processors, and a non-coherent link to the [Southbridge](https://en.wikipedia.org/wiki/Southbridge_(computing)) and I/O devices. The terms [multi-core](https://en.wikipedia.org/wiki/Multi-core) and [microprocessor](https://en.wikipedia.org/wiki/Microprocessor) unit (MPU) have come into general use for one die having multiple CPU cores.

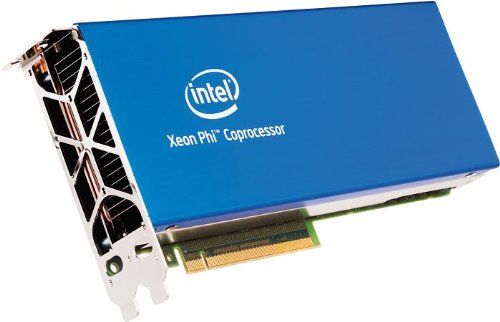
### Multi-threading

Current designs work best when the computer is running only one program. However, nearly all modern systems allow running multiple programs together. For the CPU to change over and do work on another program needs costly [context switching](https://en.wikipedia.org/wiki/Context_switching). In contrast, multi-threaded CPUs can handle instructions from multiple programs at once.

To do this, such CPUs include several sets of registers. When a context switch occurs, the contents of the working registers are simply copied into one of a set of registers for this purpose.

Such designs often include thousands of registers instead of hundreds as in a typical design. On the downside, registers tend to be somewhat costly in chip space needed to implement them. This chip space might be used otherwise for some other purpose.

Intel calls this technology "hyper-threading" and offers two threads per core in its current Core i3, Core i7 and Core i9 Desktop lineup (as well as in its Core i3, Core i5 and Core i7 Mobile lineup), as well as offering up to four threads per core in high-end Xeon Phi processors.

* **Xeon Phi**

Xeon Phi is a series of [x86](https://en.wikipedia.org/wiki/X86) [manycore processors](https://en.wikipedia.org/wiki/Manycore_processors) designed and made entirely by [Intel](https://en.wikipedia.org/wiki/Intel). They are intended for use in supercomputers, servers, and high-end workstations. Its architecture allows use of standard programming languages and APIs such as [OpenMP](https://en.wikipedia.org/wiki/OpenMP).

Since it was originally based on an earlier GPU design by Intel, it shares application areas with GPUs. The main difference between Xeon Phi and a [GPGPU](https://en.wikipedia.org/wiki/GPGPU) like [Nvidia Tesla](https://en.wikipedia.org/wiki/Nvidia_Tesla) is that Xeon Phi, with an x86-compatible core, can, with less modification, run software that was originally targeted at a standard x86 CPU.

Picture from Intel