A Case Study on: How CPU has evolved over the years and its Future

Hello!

We are

Sonu Kumar Kushwaha (2K19/CO/383)

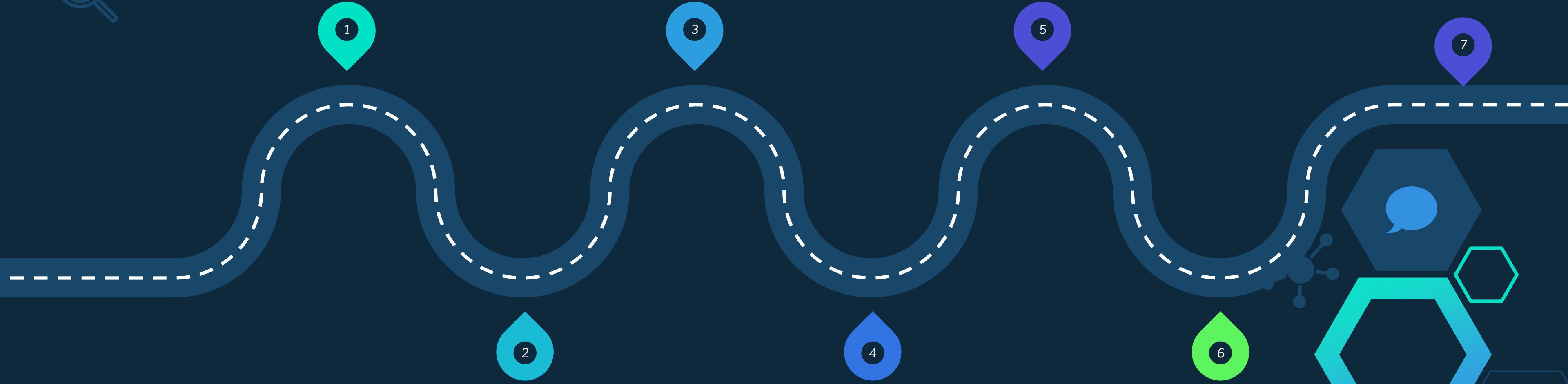
Ayush Karn (2K19/CO/454) Group 1

Abstract

In this project we have done a Case Study on how the CPU/Computers are in the place where they are right now. How has it evolved over the years and what is its future going to look like. It is important to study the history of something in order to better and fully understand the present state of an industry or idea.

Studying the history of computers is a way of paying tribute to those who came before, but more importantly, the previous knowledge in the field is the foundation of what is current. In order to fully understand something, you have to know and understand the foundation. As a computer user, we might not need to understand much about computer history in order to operate a PC; however, I can diagnose problems and solve them better if I understand computer basics. Those basics intersect with the history of computers and how they developed. Those are basics now, but they weren't always basics. Knowing those basics and how they came to be the standardized base knowledge is important, because it creates a deeper and more robust set of knowledge in a user.

Roadmap



1. Working of CPU
2. Intel 4004 and 8008
3. Intel 8080
4. Intel 8086/8087A and IBM PC
5. Intel Pentium & Core 2 Duo
6. Intel i9 and AMD Ryzen 9
7. Moore’s Law

1

Working of a CPU

In order to understand the evolution of a CPU we need to have basic understanding of how a CPU works. So, lets us know how a CPU works

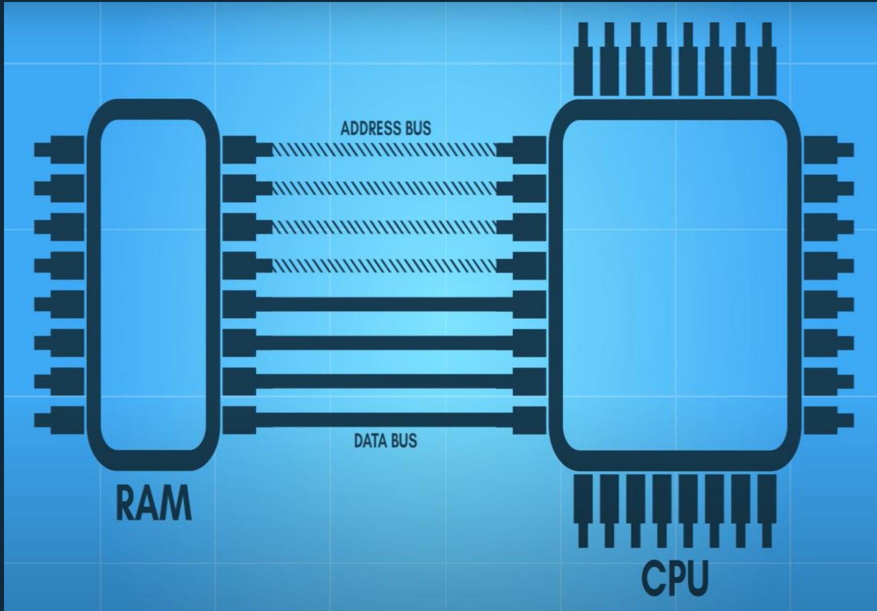
In order to understand how a CPU derives its processing power, let us examine what a CPU actually does and how it interfaces with data.

- In digital electronics everything is represented by the binary “bit” (0/1 - ON/OFF - UP/DOWN - HIGH/LOW)

- In a CPU, a “bit” is physically transmitted as voltage levels. When we combine multiple “bits” together in a group, we can now represent more combinations of discrete states. For example, if we combine eight bits together we form a byte.
- A byte can represent 256 different states and can be used to represent numbers.
- In the case of a byte, any number between 0 and 255 can be expressed
- That same byte, can also represent a number between -128 to 127.

- Other expressions of that byte may be colours or levels of sound.
- When we combine multiple bytes together, we create what's known as a word.
- Words are expressed in their bit capacity. A 32-bit word contains 32-bits. A 64-bit word contains 64 bits and so on.
- The original Intel 4004 processor operated on a 4-bit word. This means data moving through the CPU transmits in chunks of four bits at a time.

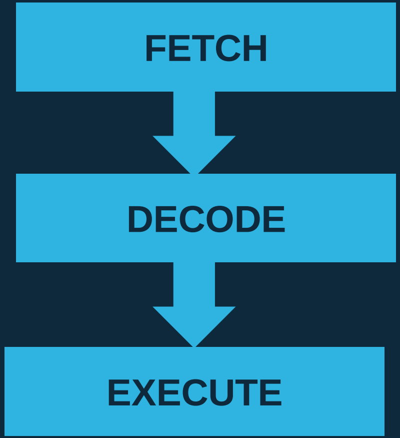
- Modern CPUs are typically 64-bit, however 32-bit processors are still quite common. By making use of larger word sizes we can represent more discrete states and consequently larger numbers.
- A 32-bit word for example, can represent up to 4.2 billion different states.
- Of all the forms data can take inside of a CPU, the most important one is that of an instruction. Instructions are unique bits of data, that are decoded and executed by the CPU as operations.



- *We can think of a CPU as an instruction processing machine. They operate by looping through three basic steps,* ***fetch, decode, and execute.***

In the fetch phase*, the CPU loads the instructions that it will be executing into itself. A CPU can be thought of as existing in an information bubble. It pulls instructions and data from outside of itself, performs operations within its own internal environment, and then returns data back. This data is typically stored in memory external of the CPU called Random Access Memory or [RAM].*

- *When a CPU loads a word of data it does it by requesting the contents of a location in RAM. This is called the data's address.*
- *The amount of data a CPU can address at one time is determined by its address capacity. A 4 bit address for example, can only directly address 16 locations of data.*


FETCH

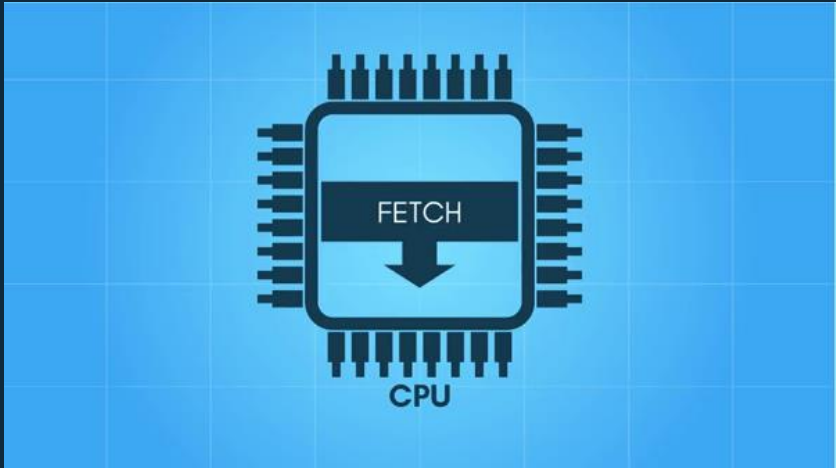
DECODE

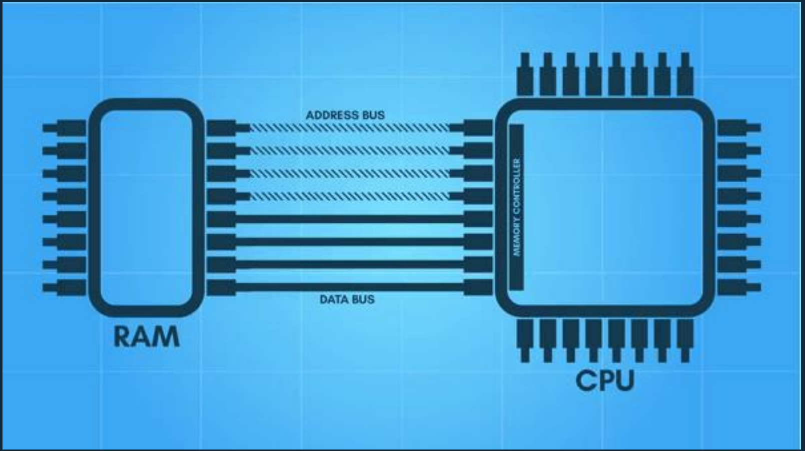
EXECUTE

- *We can think of a CPU as an instruction processing machine. They operate by looping through three basic steps,* **fetch, decode, and execute.**

In the fetch phase*, the CPU loads the instructions that it will be executing into itself. A CPU can be thought of as existing in an information bubble. It pulls instructions and data from outside of itself, performs operations within its own internal environment, and then returns data back. This data is typically stored in memory external of the CPU called Random Access Memory or [RAM].*

- *When a CPU loads a word of data it does it by requesting the contents of a location in RAM. This is called the data's address.*
- *The amount of data a CPU can address at one time is determined by its address capacity. A 4 bit address for example, can only directly address 16 locations of data.*

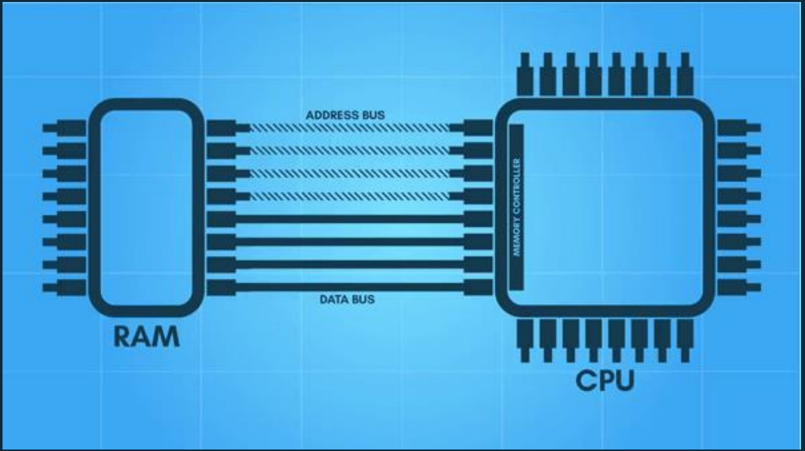




The mechanism by which data moves back and forth to RAM is called a **BUS**. A **BUS** can be thought of as a multi-lane highway between the CPU and RAM. Each bit of data has its own lane. But we also need to transmit the location of the data we're requesting, so a second highway must be added to accommodate both the size of the data word and the address word. These are called the **data bus** and **address bus** respectively. In practice these data and address lines are physical electrical connections between the CPU and RAM and often look exactly like a superhighway on a circuit board.

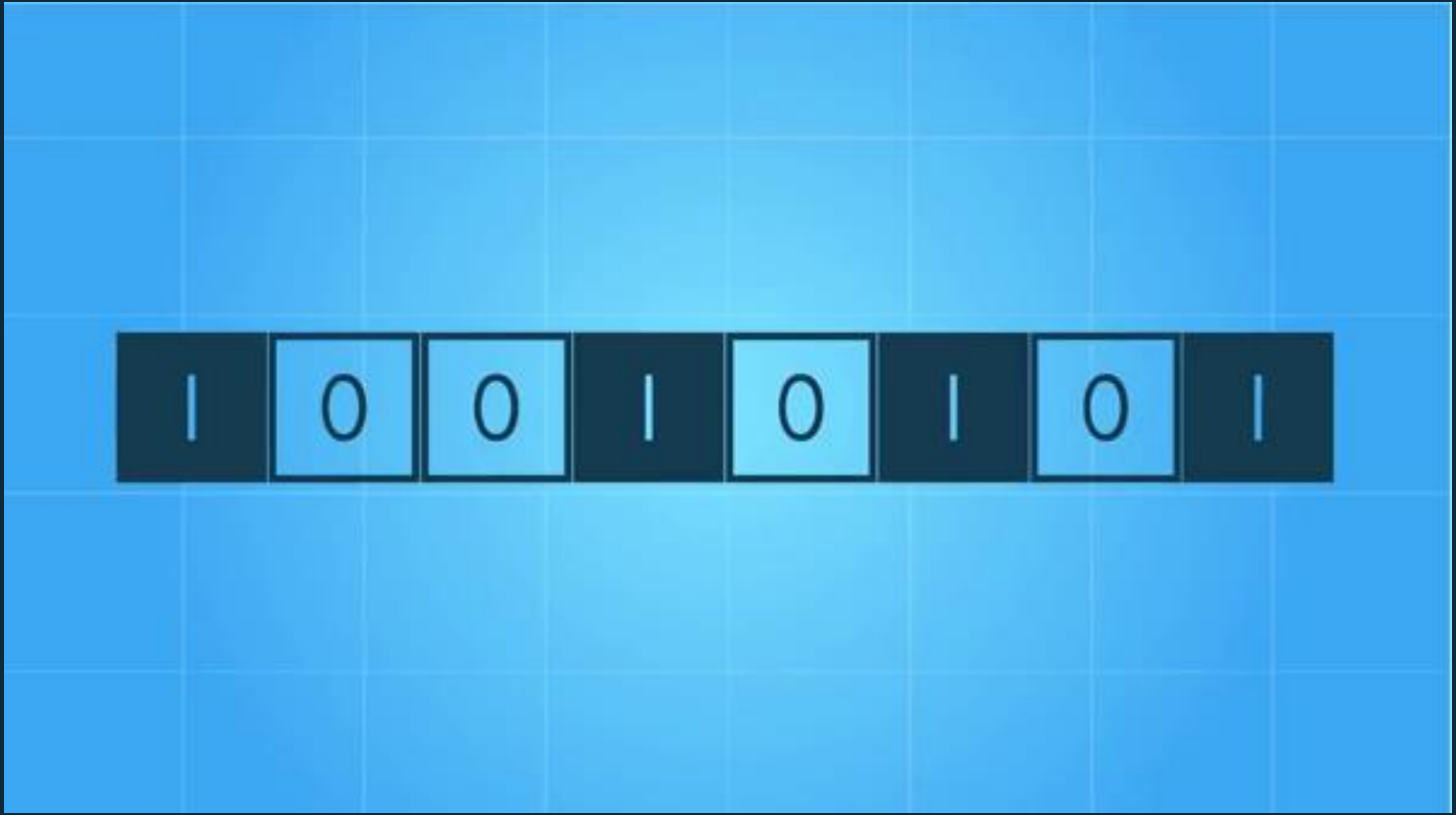
When a CPU makes a request for RAM access, a memory control region of the CPU loads the address bus with the memory word address it wishes to access. It then triggers a control line that signals a memory read request. Upon receiving this request the RAM fills the data bus with the contents of the requested memory location. The CPU now sees this data on the bus. Writing data to RAM works in a similar manner, with CPU posting to the data bus instead. When the RAM receives a "write" signal, the contents of the data bus is written to the RAM location pointed to by the address bus.

The address of the memory location to fetch is stored in the CPU, in a mechanism called a ***register****.*



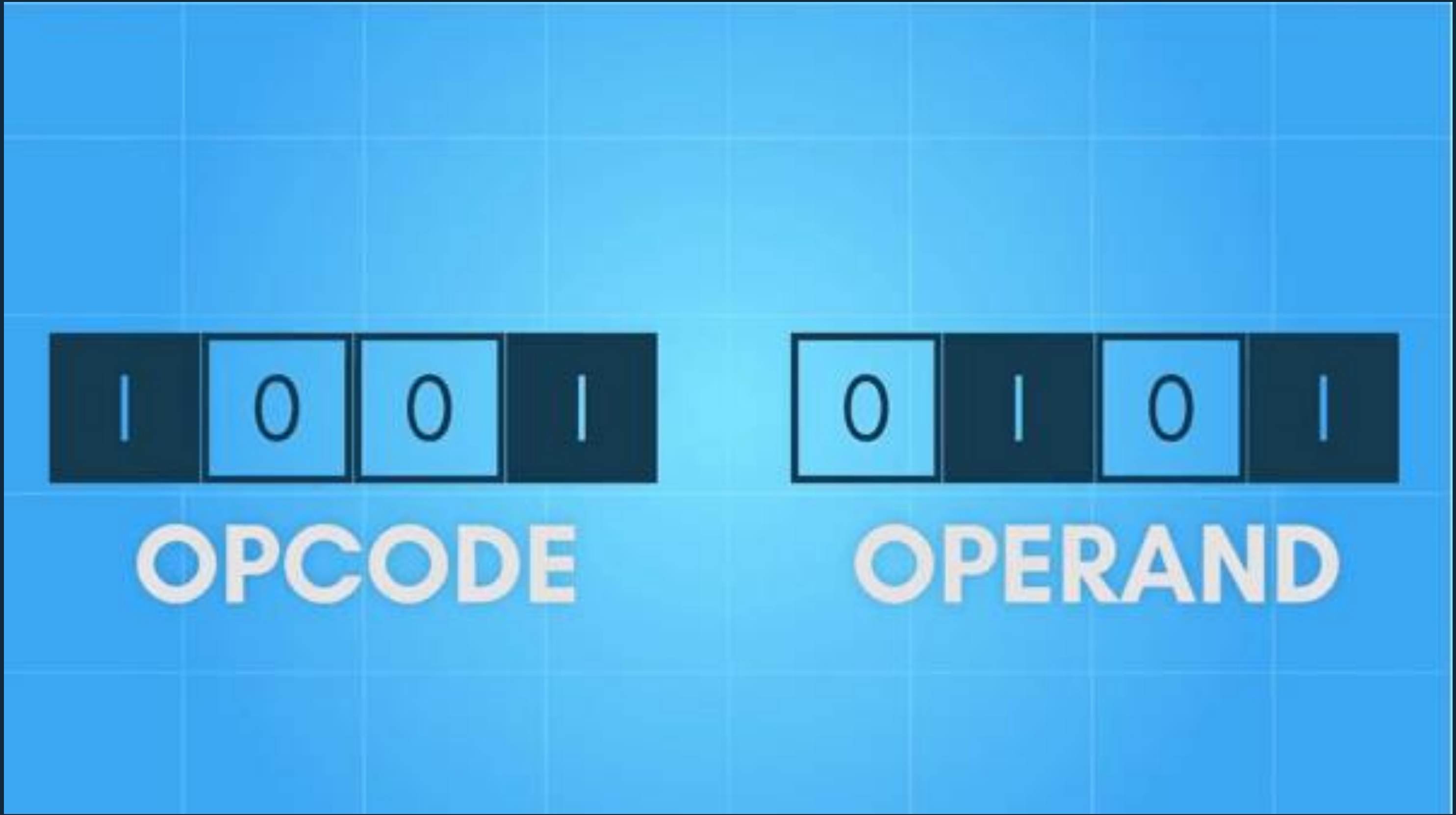
- ❑ *Made inside CPU*
- ❑ *Fastest form of Storage*
- ❑ *Were pretty small sized back in the days*
- ❑ *Generally 4MB 8MB etc are the Size of Registers*

When a instruction is decoded, the word is broken down into two parts known an bit fields that is ***opcode, & operand.***



1 0 0 1 0 1 0 1

Opcodes sometimes requires data to perform its operation on. This part of an instruction is called a ***Operand***

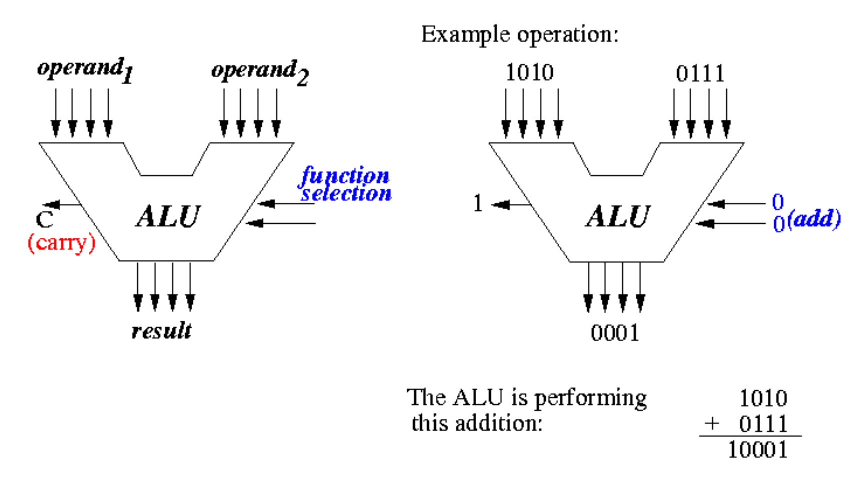


ALU

One of the most commonly used sections of a CPU in execution is the Arithmetic Logic Unit or ALU. This block of circuitry is designed to take in two operands and perform either basic arithmetic or bitwise logical operations on them The result are then outputted along with respective mathematical flags, such as a carry over, an overflow or a zero result. The output of the ALU is then sent to either a register, or a location in memory based on the opcode.

In a CPU these 3 phases of operation loop continuously, Gluing this looping machine together is a clock.

A clock is a repeating pulse use to synchronize a CPU's internal mechanics and its interface with external components. CPU clock rate is measured by the number of pulses per second, or Hertz. The Intel 4004 ran at 740 KHz or 740,000 pulses a second. Modern CPUs can touch clock rates approaching 5GHz, or 5 billion pulses a second.



The ALU is performing this addition:

$$\begin{array}{r} 1010 \\ +\ 0111 \\ \hline 10001 \end{array}$$

Evolution

Now that we know how a CPU works it’ll be much easier to understand how they evolved over the years.

Intel 4004 and 8008

Intel 4004

- Year: 1971 AD
- Transistors: 2,300 / 740 kHz clock speed
- 4 bit word CPU / 12-bit addresses.
- Created for: Busicom 141-PF Calculator and then made available to general public.
- Capable of executing 46,250 to 92,500 instructions per second.
- Instruction stored in ROM

Intel 8008

Year: 1972 AD

Transistors: 2,300 / 800kHz

8 bit word CPU / 14 bit address

Capable of executing 45,000 to 100,000 instructions per second.

Indirect Addressing was introduced

Its DNA can be found in today’s CPU as well

Intel 8080

- » 8-bit microprocessor
- » Up to 4 MHz
- » 64 KB RAM
- » Stack in RAM
- » 256 I/O ports

- *The 8080 was designed by Federico Faggin and Masatoshi Shima. Stan Mazor contributed to chip design in year 1974.*
- *The Intel 8080/8080A was not object-code compatible with the 8008, but it was source-code compatible with it. The 8080 CPU had the same interrupt processing logic as the 8008, which made porting of old applications easier. Maximum memory size on the Intel 8080 was increased from 16 KB to 64 KB. The number of I/O ports was increased to 256. In addition to all 8008 instructions and addressing modes the 8080 processor included many new instructions and direct addressing mode. The 8080 also included new Stack Pointer (SP) register. The SP was used to specify position of external stack in CPU memory, and the stack could grow as large as the size of memory. Thus, the CPU was no longer limited to 7-level internal stack, like the 8008 did.*
- *The Intel 8080 microprocessor was very popular and was second-sourced by many manufacturers. Clones of the 8080 processor were made in USSR, Poland, CSSR, Hungary and Romania.*

Intel 8086/ 8087A & IBM PC

- » 16-bit microprocessor
- » 16-bit data bus / CISC
- » Up to 10 MHz
- » 1 MB RAM
- » 64K I/O ports

- *Intel 8086 microprocessor is a first member of x86 family of processors. Advertised as a "source-code compatible" with Intel 8080 and Intel 8085 processors, the 8086 was not object code compatible with them. The 8086 has complete 16-bit architecture - 16-bit internal registers, 16-bit data bus, and 20-bit address bus (1 MB of physical memory). Because the processor has 16-bit index registers and memory pointers, it can effectively address only 64 KB of memory.*
- *Intel 8086 instruction set includes a few very powerful string instructions. When these instructions are prefixed by REP (repeat) instruction, the CPU will perform block operations - move block of data, compare data blocks, set data block to certain value, etc., that is one 8086 string instruction with a REP prefix could do as much as a 4-5 instruction loop on some other processors.*
- *Original Intel 8086 CPU was manufactured using HMOS technology. Later Intel introduced 80C86 and 80C86A - CHMOS versions of the CPU. These microprocessors had much lower power consumption and featured standby mode.*

Intel Pentium and Core 2 Duo

Intel Pentium

- Year: 1993 AD
- Transistors: 3.3 Million
- 64 Bit Data Bus / 32-bit address BUS
- Built-in floating-point and memory-management units, and two 8KB caches.
- 60 megahertz (MHz) to 200 MHz.

Core 2 Duo

- Year: 2006 AD
- 64 bit Processor based on x86
- 291 million transistors
- Dual core technology
- x86 architecture

Intel Core i9 & AMD Ryzen 9

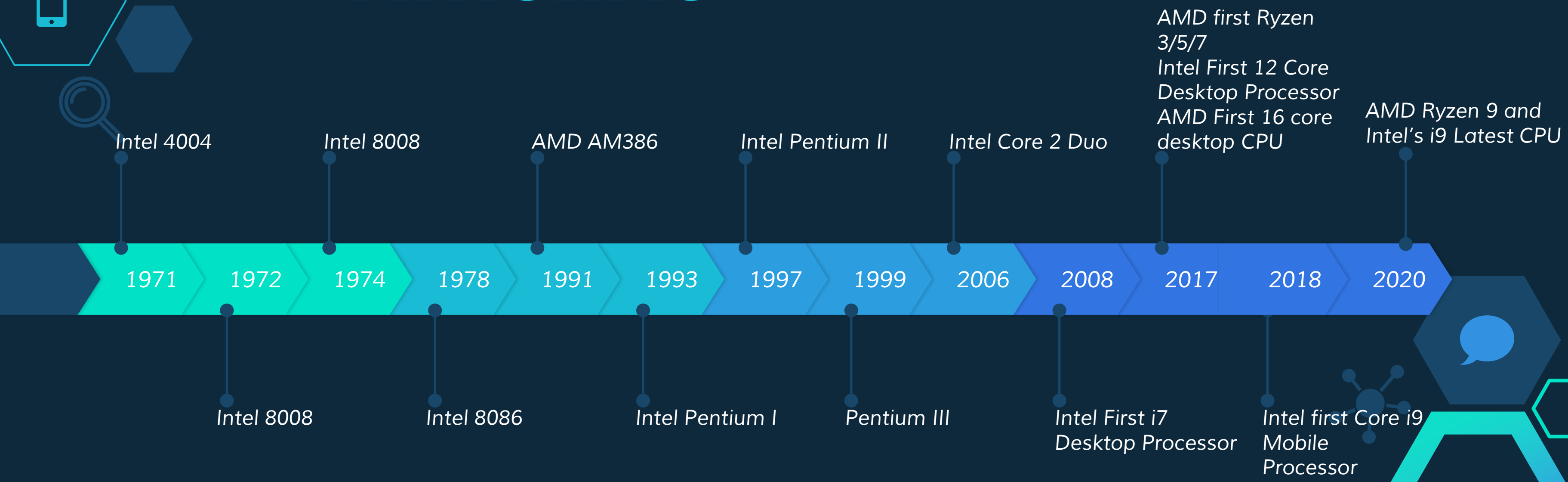
Intel Core i9 (Latest)

- Year: 2021 AD
- Max Frequency 5 GHz
- Cache 16MB
- Bus Speed is 8GT/s
- 64 bit Processor
- 14nm Technology
- 8 Core 16 Thread

AMD Ryzen 9 (Latest)

- Year: 2021 AD
- Max Frequency 4.9GHz
- L2 Cache 8MB and L3 Cache 64MB
- Base Clock 3.4 GHz
- 64 bit Processor
- 7nm Technology
- 16 Core 32 Thread

Timeline



1971	Intel 4004
1972	Intel 8008
1974	Intel 8008
1978	Intel 8086
1991	AMD AM386
1993	Intel Pentium I
1997	Intel Pentium II
1999	Pentium III
2006	Intel Core 2 Duo
2008	Intel First i7 Desktop Processor
2017	AMD first Ryzen 3/5/7 Intel First 12 Core Desktop Processor AMD First 16 core desktop CPU
2018	Intel first Core i9 Mobile Processor
2020	AMD Ryzen 9 and Intel's i9 Latest CPU

The future of processors: Graphene Computers

- *Graphene is a thin layer of carbon whose atoms are arranged in a hexagonal configuration. They are a scientific marvel due to their atypical but remarkable electrical and mechanical properties.*

- *Discovered in 2004, graphene gave rise to a new wave of research in electronics. This super-effective material possesses a couple of features which will allow it to become the future of computing.*

- *Firstly, it is capable of conducting heat and electricity faster than any other conductor used in electronics, including copper.*

- *It act as a super-conductor of heat and electricity at higher temperature than any other material, few degrees above 0.*

Advantages over Silicon

- *At room temperature, graphene is capable of conducting electricity 250 times better than silicon, a rate faster than any other known substance.*

- *The top clock speed silicon-based chips can work at reaches 3-5 GHz. Scientists managed to achieve a speed which was a thousand times higher than that of silicon chips.*

- *Graphene-based CPUs turned out to consume a hundred times less energy than their silicon counterparts.*

- *Graphene also allows for smaller size and greater functionality of the devices having it.*

- *The processors built based on Graphene technology will be Hundredth of time faster than current silicon based processors.*

Disadvantages

Silicon serves as a good semiconductor that is able not only to carry electricity but also to retain it. Graphene, on the other hand, is a 'superconductor' that carries electricity at a super-high speed but cannot retain the charge.

Researchers have tried various methods to introduce artificial bandgaps into graphene, from patterning it into nanoscale ribbons to doping its surface with chemicals. However, these methods are typically complex and expensive, making it difficult to adapt them for large-scale use in industry.

The primary reasons why graphene is currently not being used include that it is still relatively new and in early stages of development — and there's also cost to take into account. Right now the cost of making a singular gram of graphene is approximately $100 USD.

Moore’s Law

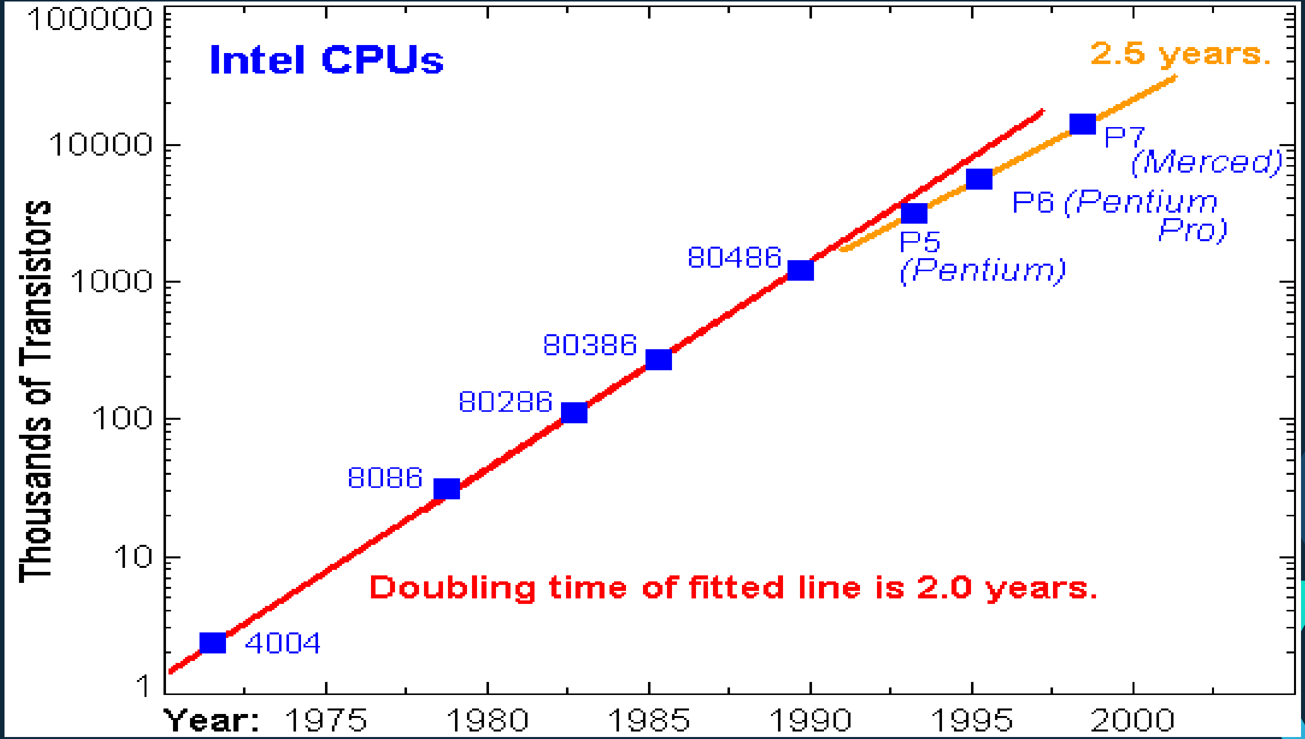
Moore's Law refers to Moore's perception that the number of transistors on a microchip doubles every two years, though the cost of computers is halved. Moore's Law states that we can expect the speed and capability of our computers to increase every couple of years, and we will pay less for them. Another tenet of Moore's Law asserts that this growth is exponential.

Moore's Law in Action: You and I

Maybe you have experienced (as I have) the need to purchase a new computer or phone more often than you wanted to—say every two-to-four years—either because it was too slow, would not run a new application, or for other reasons. This is a phenomenon of Moore's Law that we all know quite well.

Is Moore’s Law coming to an end?

According to expert opinion, Moore’s Law is estimated to end sometime in the 2020s. What this means is that computers are projected to reach their limits because transistors will be unable to operate within smaller circuits at increasingly higher temperatures. This is due to the fact that cooling the transistors will require more energy than the energy that passes through the transistor itself.


Intel CPUs
2.5 years.
Thousands of Transistors
100000
10000
1000
100
10
1
4004
8086
80286
80386
80486
P5
(Pentium)
P6 (Pentium
Pro)
P7
(Merced)
Doubling time of fitted line is 2.0 years.
Year: 1975
1980
1985
1990
1995
2000

Moore's Law

Thanks!