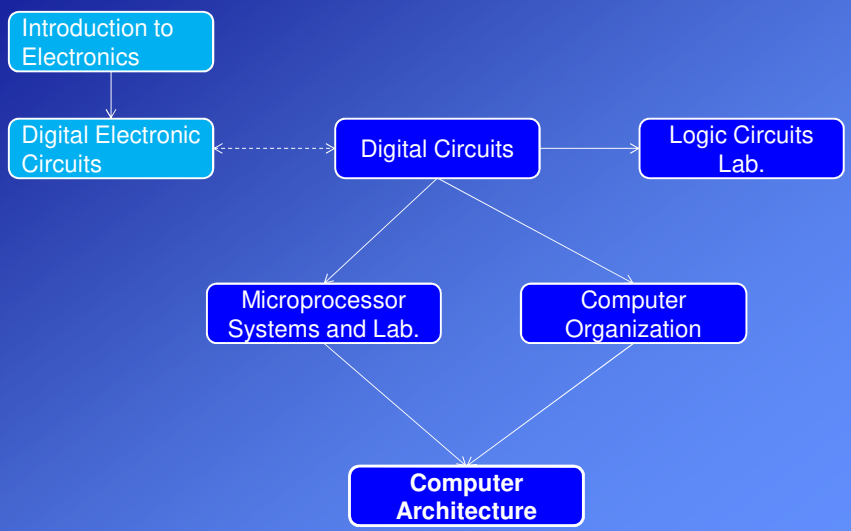
COMPUTER ARCHITECTURE

Feza BUZLUCA, Ph.D
Istanbul Technical University
Computer Engineering Department

http://www.faculty.itu.edu.tr/buzluca
http://www.buzluca.info

1.0 Connections between the hardware-based courses in the İTÜ Computer Engineering Department



1.1. Why to study Computer Architecture?

From: The *IEEE/ACM Computer Curricula 2013*,
prepared by the Joint Task Force on Computing Curricula
of the IEEE (Institute of Electrical and Electronics Engineers) Computer Society
and ACM (Association for Computing Machinery)

- Computing professionals should not regard the computer as just a black box that executes programs by magic.
- Computer architecture is a key component of computer engineering and the practicing computer engineer should have a practical understanding of this topic.
- Students need to understand computer architecture to develop programs that can achieve high performance through a programmer's awareness of parallelism and latency.
- In selecting a system to use, students should to able to understand the tradeoff among various components, such as CPU clock speed, cycles per instruction, memory size, and average memory access time.

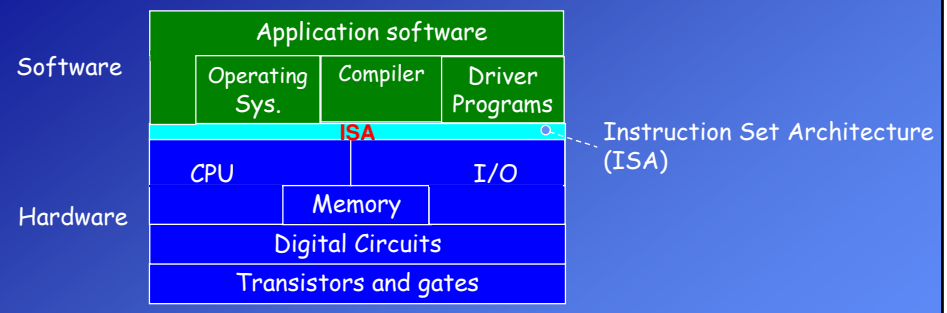
The Purpose of the Course

1. Learning how to design computer systems
2. Learning how to choose an appropriate computer system to solve a problem (to realize a project) considering requirements such as processing and memory capabilities
3. Developing high quality software for big and embedded systems

Topics:

- The Pipeline
  - Instruction Pipeline (Instruction-Level Parallelism)
  - Pipeline hazards and solutions
- Input/Output Organization
  - Handshaking
  - Data transfer between CPU and memory
- Exceptions and Interrupts
  - Vectors, multiple interrupts, priority, nested interrupts
- Direct Memory Access - DMA
- Memory Organization
  - Cache memory
  - Virtual Memory
- RAID: (Redundant Array of Independent/Inexpensive Disks)
- Multiprocessor and multicore systems
- Floating Point Numbers

1.2. The layered logical model of a computer system:



Instruction Set Architecture (ISA) is the interface (the part) of the computer hardware related to the programming.
The ISA includes machine language instructions, registers and native data types.

1.3 The Central Processing Unit – CPU

1.3.1 Categorization of CPUs

CPUs can be categorized in different ways according to their properties.

- According to the numbers of their operands:
  - Zero operand/address machines (stack machines)
  - One operand/address machines (accumulator machines)
  - Two operand/address machines (Register-register, register-memory, memory-memory)
  - Three operand/address machines
- According to instruction sets and addressing capabilities
  - CISC (Complex Instruction Set Computer)
  - RISC (Reduced Instruction Set)
- According to their instruction and data memories
  - Von Neuman architecture
  - Harvard architecture

1.3.1.1 According to instruction sets and addressing capabilities:

a) CISC (*Complex Instruction Set Computer*)

b) RISC (*Reduced Instruction Set Computer*)

CISC:

Motivation:

- A desire to simplify compilers. The machine language is made close to high level language.
- A desire to improve performance. Shorter programs written with powerful instructions.

Characteristics:

- Large number of instructions (100 -250)
- Complex instructions and complex addressing modes (indirect memory access)
- Instructions that directly operate on memory locations
- Microprogrammed control unit

Consequences:

- Instructions with different lengths. Difficult to decode and prefetch
- Some instructions are used very rarely.
- Complex internal structure

RISC:

Research on computer programs presents following results (motivation):

- There are many assignment operations (A = B) .
- Most of the instructions in a compiled program are the relatively simple ones. Because complex machine instructions are often hard to exploit. The compiler must find those cases that exactly fit the construct.
- Accessed operands are mostly local and simple (not array or vector).
- Function calls (subroutines) have a large overhead: Saving the return address, transfer of parameters, local variables, stack (memory) access
- Most of the subroutines (98%) transfer 6 or less parameters. [1]
- Most of the subroutines (92%) use 6 or less local data.[1]
- Depth of nesting function calls is mostly (99%) less than 8. [2]

According to these results RISC processors with simple instructions which operate only on registers and access memory only for load/store operations are designed.

1. Andrew S. Tanenbaum, Implications of structured programming for machine architecture, Communications of the ACM, Vol.21 , No.3 (1978),pp. 237 - 246
2. Yuval Tamir and Carlo H. Sequin, "Strategies for Managing the Register File in RISC," IEEE Transactions on Computers Vol. C-32(11) pp. 977-989, 1983.

Characteristics of Reduced Instruction Set Architectures

Although there are different types of the reduced instruction set architecture, certain characteristics are common to all of them.

- A small set of (about 30 instructions)
- Simple instructions with simple format (Fixed length, easy to decode)
- Simple addressing modes
- Register to register operations
- Memory access only for load/store instructions (*load-store architecture*).
- One instruction per clock cycle (owing to pipelining)
- *Hardwired* control unit.

Other Characteristics:

Some features are also included by the CISC processor.

Some of the features listed below are not included in all RISC processors

- A large number of registers (128-256) (*Register File*)
- Overlapped register window to transfer of parameters and to save local data
- Instruction pipeline
- Harvard architecture

Examples of CISC and RISC processors:

- RISC:

MIPS, SPARC, Alpha, HP-PA, PowerPC, i860, i960, ARM, Atmel AVR

- CISC:

VAX, PDP-11, Intel x86 until Pentium, Motorola 68K.

- Hybrid (Outer CISC shell with an inner RISC core):

Pentium, AMD Athlon.

There is a growing realization that
RISC designs may benefit from the inclusion of some CISC features
and that CISC designs may benefit from the inclusion of some RISC features.
The result is that the more recent RISC designs, notably the PowerPC, are no longer "pure" RISC and
the more recent CISC designs, notably the Pentium II and later Pentium models, do incorporate some RISC characteristics.

Examples of products where RISC processors are used:

- ARM:
  - Apple iPod , Apple iPhone, iPod Touch, Apple iPad.
  - Palm and PocketPC PDA, smartphone
  - RIM BlackBerry smartphone/email device.
  - Microsoft Windows Mobile
  - Nintendo Game Boy Advance
- MIPS:
  - SGI computers, PlayStation, PlayStation 2
- Power Architecture (IBM, Freescale (previously Motorola SPS)):
  - IBM supercomputers, midrange servers and workstations,
  - Apple PowerPC-bases Macintosh
  - Nintendo Gamecube, Wii
  - Microsoft Xbox 360
  - Sony PlayStation 3
- Atmel AVR:
  - BMW cars as controllers

1.3.1.2. Categorization of CPUs according to their instruction and data memories

a) Von Neumann Architecture:

Instructions and data are stored in the same memory

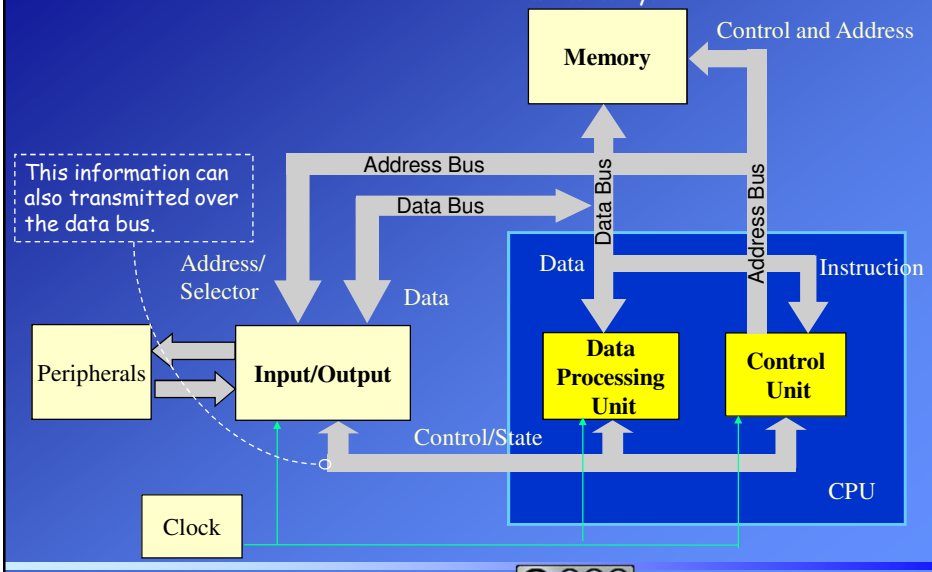
b) Harvard Architecture:

Instructions and data are stored in different memories.

CPU can fetch instructions and data at same time.

Von Neumann Architecture: John von Neumann (1903 – 1957)

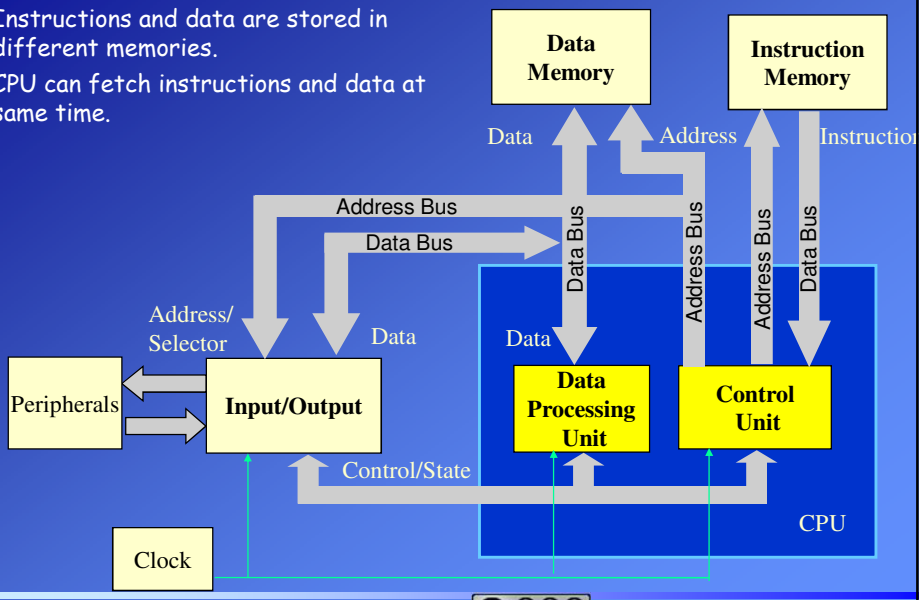
Instructions and data are stored in the same memory.



Harvard Architecture : Harvard Mark I , Harvard University

Instructions and data are stored in different memories.

CPU can fetch instructions and data at same time.



1.3.2 Internal Structure of a CPU

• Data Processing Unit:

Performs data processing and internal data storage functions.

It includes registers, arithmetic-logic unit, floating point unit, data pipeline.

• Control Unit:

It decodes and interprets instructions; provides control signals to the data processing unit.

Actually it controls the operation of the CPU and hence the computer.

The control unit is designed as a finite state machine. Refer to instruction cycles of the CPU for the states (instruction fetch, data fetch, execution etc.)

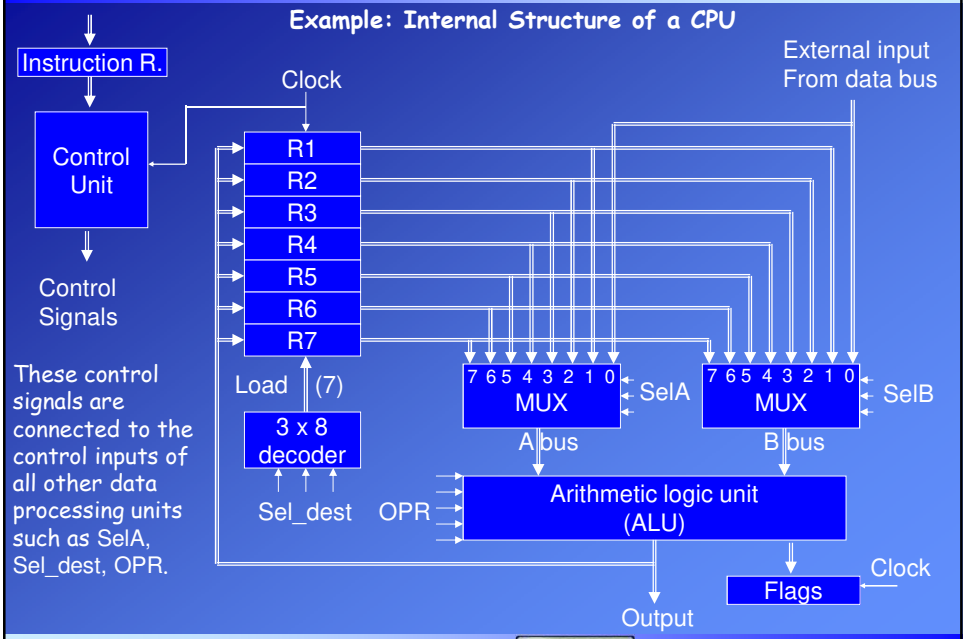
It can be implemented as a synchronous sequential circuit (*hardwired*)

or as a *microprogrammed* machine.

Remember each instruction in the machine language of the processor is translated into a sequence of lower-level control unit instructions which are called microinstructions.

The internal structure of an exemplary CPU is shown in 1.17.

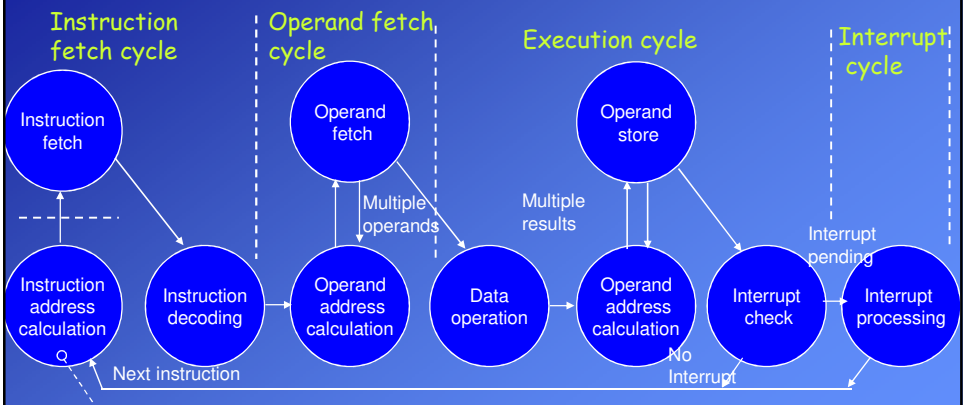
Example: Internal Structure of a CPU



Instruction Cycle of a CPU:

The control unit of a CPU designed as finite state machine can be in one of the 4 states (cycles):

1. Instruction fetch cycle, 2. Operand fetch c., 3. Execution c., 4. Interrupt cycl.



The address of the next instruction is calculated in execution or interrupt cycle.

1.4 The evolution of computers

Characteristics of evolution:

Increase in processor speed, increase in level of integration of circuits, decrease in component size, increase in memory size, and increase in I/O capacity and speed.

Reasons for the increase in processor speed:

- **Achievements in material:**

  Shrinking size of microprocessor components; this reduces the distance between components and hence increases speed.

- **Organizational improvements:**

  Heavy use of pipelining and parallel execution techniques, multiple ALUs multicore designs

  Cache memories

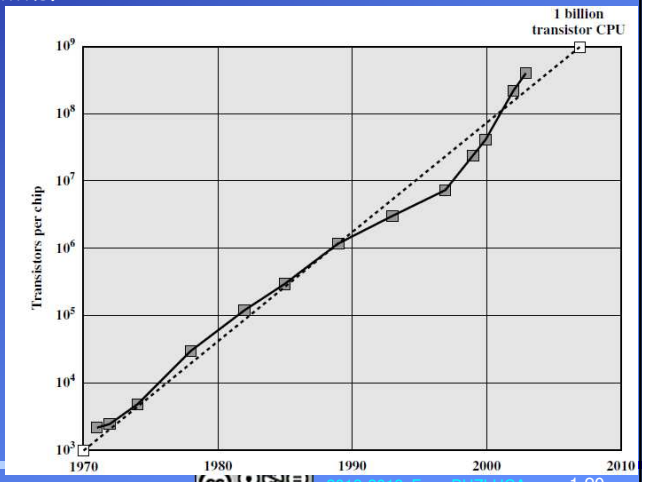
In this course we will discuss the organizational improvements.

Integration of circuits

Moore's law (by Gordon Moore, cofounder of Intel): "The number of transistors that could be put on a single chip is doubling every year and this pace will continue into the near future". (1965)

After 1970s the number of transistors on integrated circuits doubles approximately every 18 months.

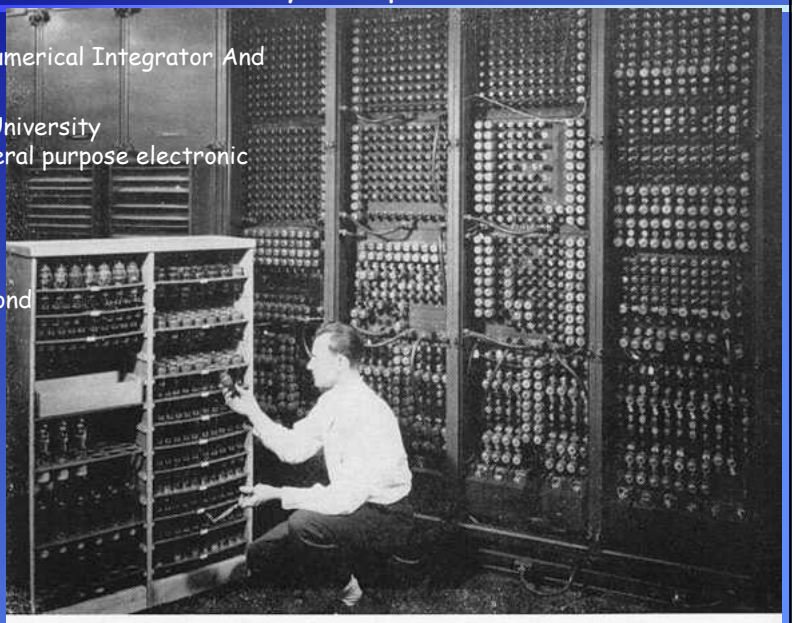
Source:
William Stallings, Computer Organization and Architecture, 8/e, Prentice Hall, 2009



A brief history of computers

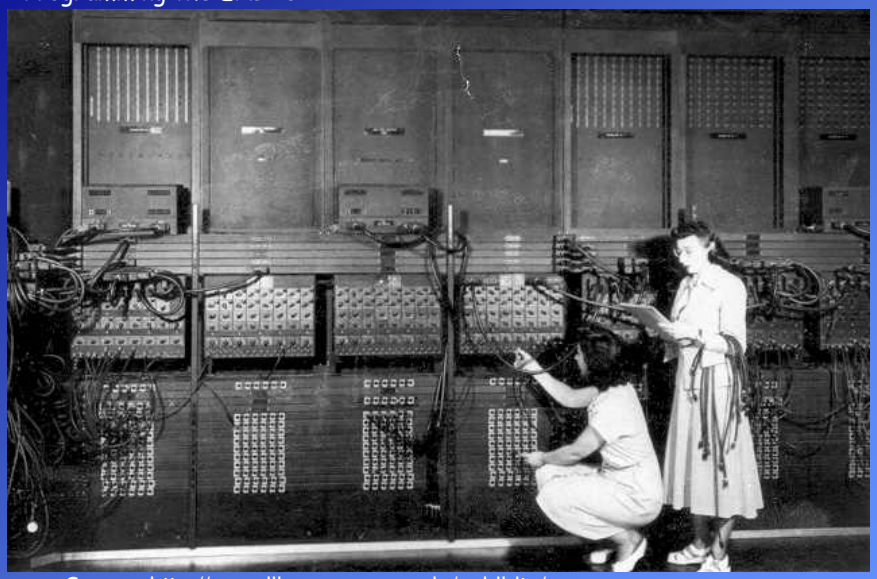
ENIAC 1946
(Electronic Numerical Integrator And Computer),

Pennsylvania University
The first general purpose electronic computer
30 tons
140 kW
5000 additions/second



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

Programming the ENIAC



Source http://www.library.upenn.edu/exhibits/

Z3 (1941):

Konrad Zuse, (1910-1995)

The first general purpose computer.

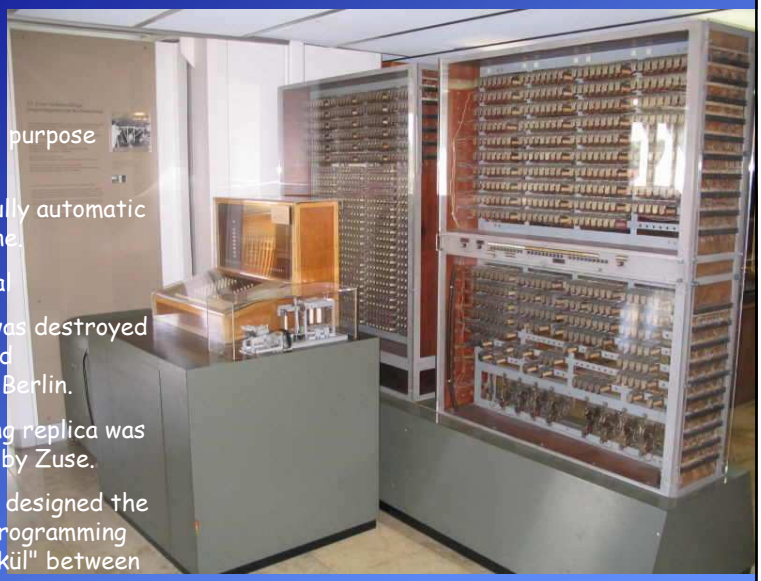
Programmable, fully automatic computing machine.

Electromechanical

The original Z3 was destroyed in 1943 during and bombardment of Berlin.

A fully functioning replica was built in the 1962 by Zuse.

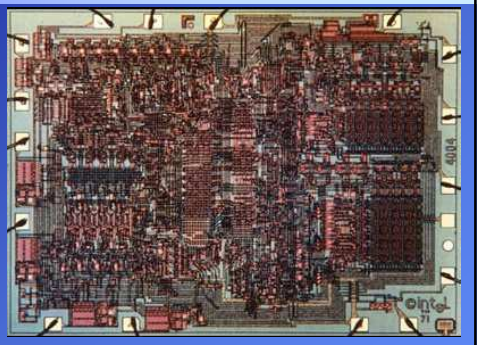
Konrad Zuse also designed the first high-level programming language "Plankalkül" between 1942 and 1946.

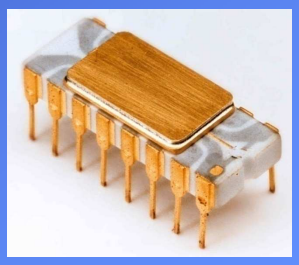


First microprocessor:

Intel 4004

- 1971
- 4-bit data
- 2300 transistors
- 740 KHz
- Addressable Memory: 640 Bytes
- 12 V

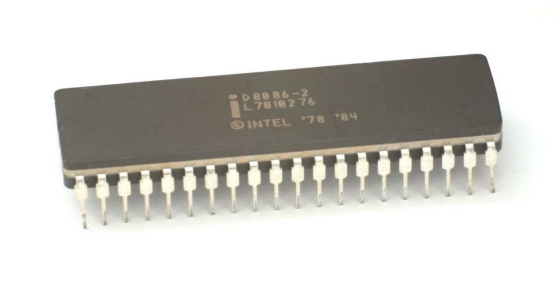




Source http://www.intel.com

First member of the x86 Family:
Intel 8086

- 1978
- 16-bit data
- 29000 transistors
- 3-10 MHz
- Addressable Memory: 1 MBytes
- 5 V

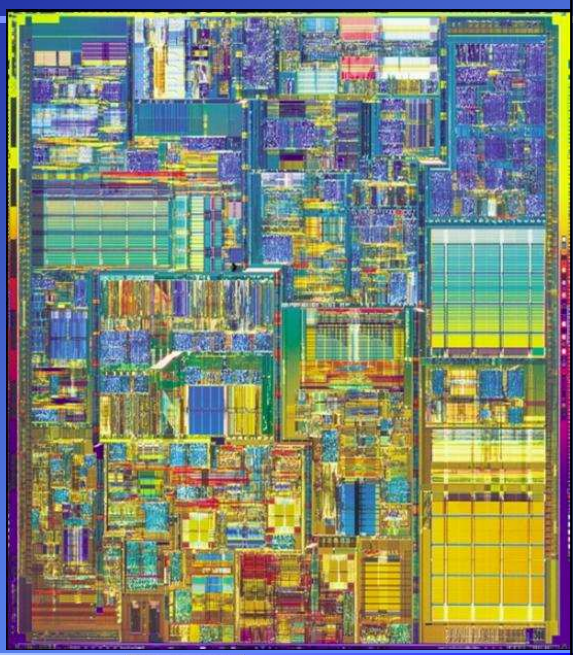


This picture is taken from Wikipedia.

Multithreading
(Hyper-threading)

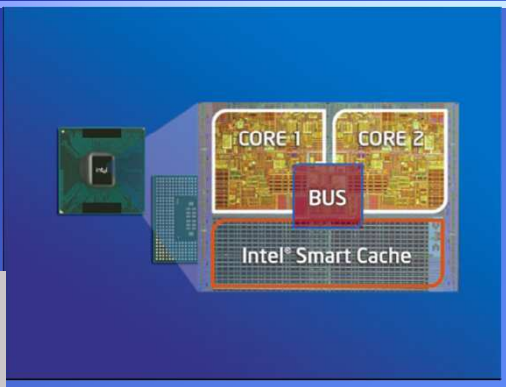
Intel Pentium4 + HT

- 2003
- 32/64-bit data
- 55 million transistors
- 3.4 GHz
- Addressable Memory: 64 GBytes
- 1.2 V



Multicore Processors:
Intel® Core™ Duo

- 2006
- 64-bit data
- 100 million transistors
- 2.66 GHz
- 1.5 V



Intel® Core™ i7 EE - 4960X

- 2013
- 4 GHz
- 6 cores
- 64-bit data
- 64 GB memory addressing capacity
- 1,5 MB L2, 15 MB L3 cache memory

Improvements in Computer Organization and Architecture

There are three approaches for achieving increased processor speed:

1. Increasing the hardware speed of the processor (Clock speed) (But !)
2. Increasing the size and speed of cache memories
3. Making changes to the processor organization and architecture that increase the effective speed of instruction execution

As clock speed and logic density increase, a number of problems arises (#1 above).

Power: It is difficult to dissipate the heat generated on high-density, high-speed chips (The power wall, see slide 1.29).

RC delay: The speed at which electrons can flow on a chip between transistors is limited by the resistance and capacitance of the metal wires connecting them.

The wire interconnects become thinner, increasing resistance. Also, the wires are closer together, increasing capacitance.

Therefore it is not possible to increase the clock speed.

Memory latency: In addition, speed of memories does not increase as much as the speed of processors.

Memory speeds lag processor speeds, as previously discussed.

The Power Wall *:

Dynamic power per transistor (P) is proportional to frequency of operation (f) times the square of the operating voltage (V) ($\mathbf{P \sim V^2 f}$).

To reduce the power increase the operating voltage can be decreased, but it is limited by the transistors' operating threshold voltages.

The total dynamic power dissipated by an entire IC can be expressed as $\mathbf{P \sim Nf}$, where N represents the total number of transistors operating simultaneously.

Increasing the number of transistors at Moore's law pace and increasing the operating frequency is bound to reach a thermal dissipation limit—the power wall.

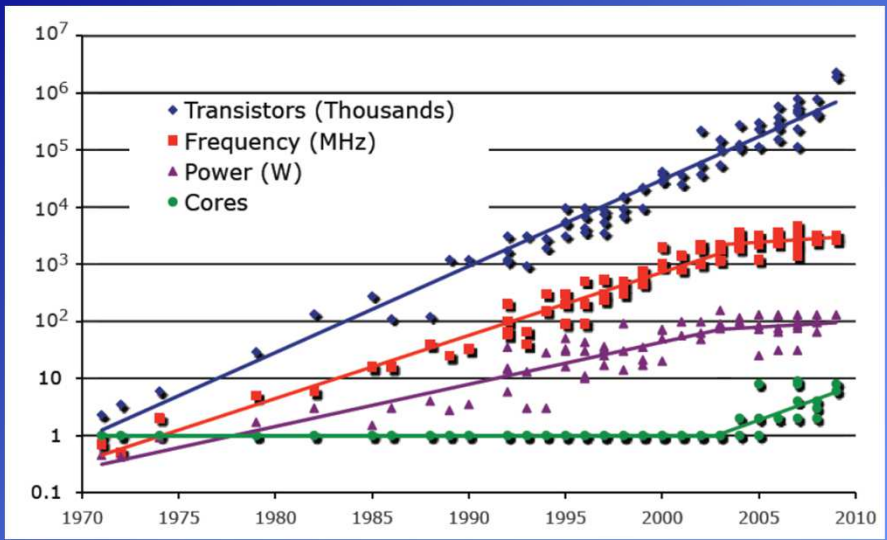
By 2003, processors exceeded 200 W per chip. This milestone marked the crossing of a power threshold that requires far more expensive cooling technologies, which were outside the system-cost envelope of PC hardware at that time.

The industry had to choose which to slow down: the growth of the microprocessor's transistor number from one generation to the next, or the operational frequency rate.

They decided on the second option, which maintained Moore's law but sacrificed frequency growth.

*Source: T. M. Conte, E. P. DeBenedictis, P. A. Gargini, and E. Track, "Rebooting Computing: The Road Ahead," Computer, vol. 50, no. 1, pp. 20–29, Jan. 2017.

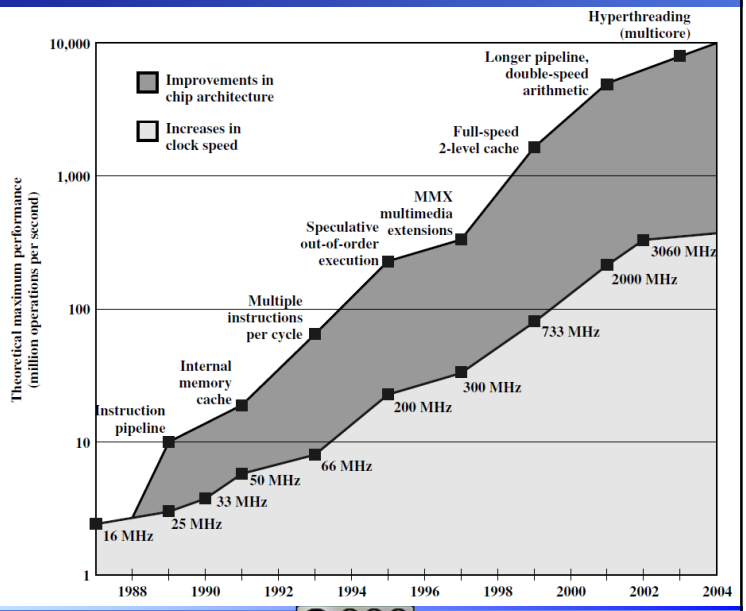
Processor Trends:



Source:
William Stallings, Computer Organization and Architecture, 10/e, Prentice Hall, 2016

Intel Microprocessor Performance:

Source:
William Stallings, Computer Organization and Architecture, 8/e, Prentice Hall, 2009



Performance balance between processor and main memory

In computer system design it is critical to balance the performance of different components so that gain in performance in one element is not handicapped by a lag in another element.

Processor speed has increased more rapidly than the speed of the main memory (memory access time).

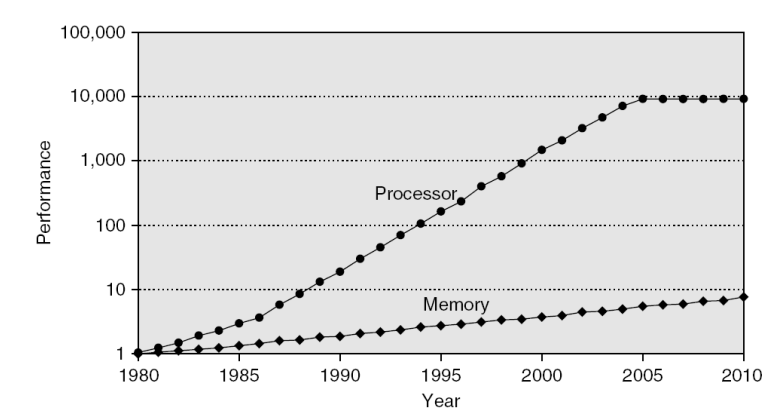
For example; the Intel Core i7 with four cores and a 3.2 GHz can demand for data and instructions a total peak bandwidth of 409.6 GB/sec.

In contrast, the peak bandwidth to DRAM main memory is only 6% of this (25 GB/sec).

Different techniques are used to compensate for this mismatch, including caches, wider data paths.

The performance gap between the CPU and the main memory is shown in the next slide.

Performance gap between the CPU and the main memory



Single processor

The processor line: Increase in memory requests per second on average (i.e., the inverse of the latency between memory references) (single processor)

The memory line: Increase in DRAM accesses per second (i.e., the inverse of the DRAM access latency).

Source: Hennessy and Patterson, "Computer Architecture A Quantitative Approach", 5/e, MK, 2012.

Benchmarks

MIPS (Millions of instructions per second) and MFLOPS (floating-point operations per second) are inadequate to evaluating the performance of processors.

Because of differences in instruction sets, the instruction execution rate is not a valid means of comparing the performance of different architectures.

SPEC Benchmarks: (http://www.spec.org/)

The best known collection of benchmark suites is defined and maintained by the System Performance Evaluation Corporation (SPEC), an industry consortium.

A benchmark suite is a collection of programs, defined in a high-level language, that together can used to test a computer in a particular application or system programming area.

The best known of the SPEC benchmark suites is SPEC CPU2006.This is the industry standard suite for processor-intensive applications.

Other SPEC suites:

- **SPECjvm98:** For Java Virtual Machine (JVM) clients
- **SPECjbb2000 (Java Business Benchmark):** A benchmark for evaluating server-side Java-based electronic commerce applications
- **SPECweb99:** Evaluates the performance of World Wide Web (WWW) servers
- **SPECmail2001:** Measures a system's performance acting as a mail server.