UT 1. Introduction to Computer Architecture

Tema 1.1 Concept of Computer Architecture

J. Duato, J. Flich, P. López, V. Lorente,
A. Pérez, S. Petit, J.C. Ruiz, S. Sáez, J. Sahuquillo

Department of Computer Engineering
Universitat Politècnica de València

Contents

Bibliography


John L. Hennessy and David A. Patterson.
Computer Architecture, Sixth Edition: A Quantitative Approach.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 6 edition, 2018.

Contents

Defines the *hardware* of a computer attending to its requirements, as a design trade-off between performance, available technology and cost. It encompasses with the following levels:

Instruction Set Architecture (ISA).

It includes everything that must be known by assembler programmers: description of instructions and their use, logical organization of memory, etc.

Processor organization

It describes the logical elements enabling the execution of instructions: registers, decoders, ALU operators, memory interfaces, etc.

Implementation

It defines the computer implementation: VLSI design, cooling, power supply, packacking technologies, connections, etc.

- Along the classic period (until the 70's), each computer architecture level was designed in isolation by a different specialist.
- Today, computer architecture engineering is considered a transversal discipline, encompassing all the aforementioned three levels. It focuses on the design of programmable machines that must execute a (foreseeable or not) set of programs in a correct and efficient way.
- The three levels are dependent: design decisions at one level may affect the others.

The computer engineer task

- consider the expected requirements
- identify existing technology, energy and cost limitations
- provide the best possible design

$\rightarrow$ quantify performance, cost and other features, compare and select

Contents

In order to design a computer, an architect must consider

- The type of required computer go
- The degree of compatibility with other existing computers

Source code	More flexible design,
	Need of new compilers
Binary	Existing *ISA* (lack of flexibility)
	No new software required

- Operating system requirements

Address space	Limits applications size
Memory management	Paging, segmentation, ...
Protection	
Process Management	Multitasking support

(cont.)

- Market standards

Floating point	IEEE 754
E/S	SATA, SCSI, PCI Express...
Operating systems	Linux, Windows, Mac OSX ...
Networks	Ethernet, InfiniBand ...
Programming languages	C, C++, Java, FORTRAN, ...

Contents

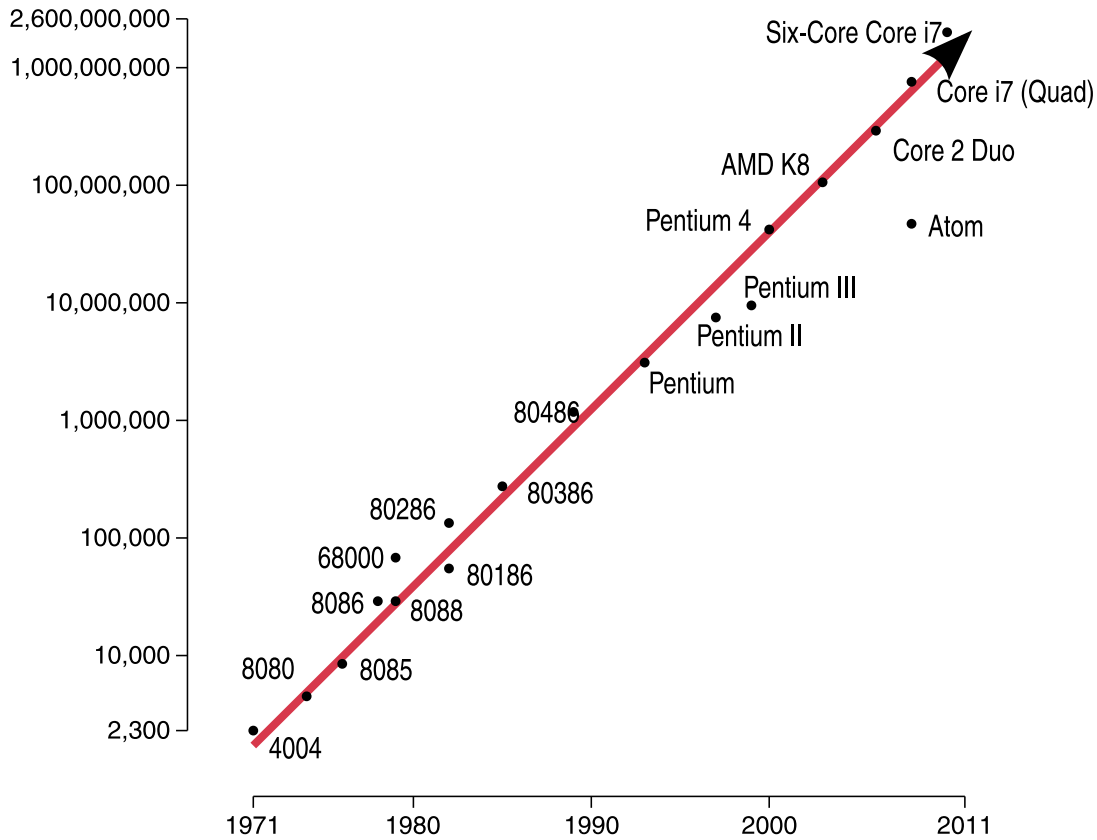
Designers must take into account:

- Available technology
- Power and energy limitations
- Cost

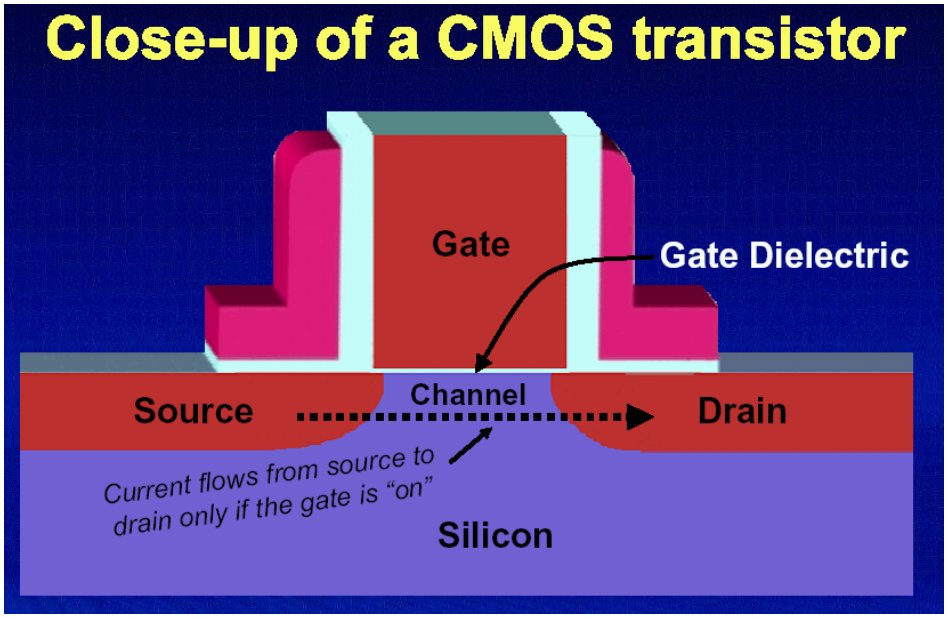
Chip Technology: Moore's Law

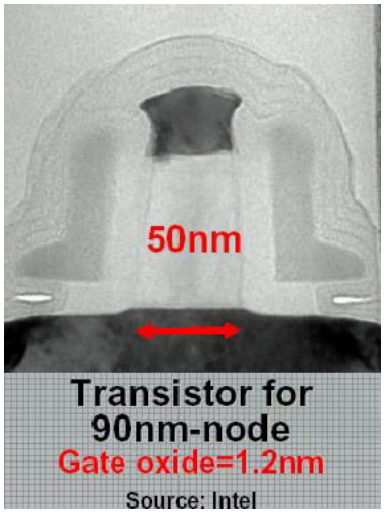
The number of transistors per chip increases 40–55% per year (2X every 18 months / 2 years). Two main causes:

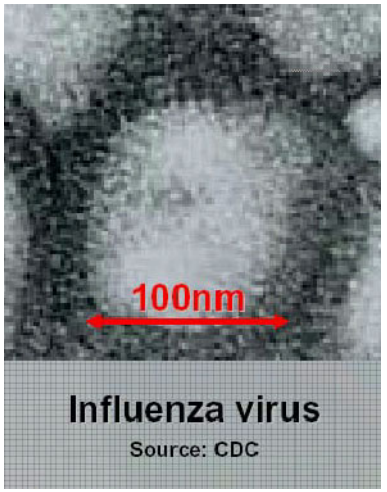
- Density of transistors increases by 35% per year
- Die size increases by 10–20% every year



Feature size (transistor size)



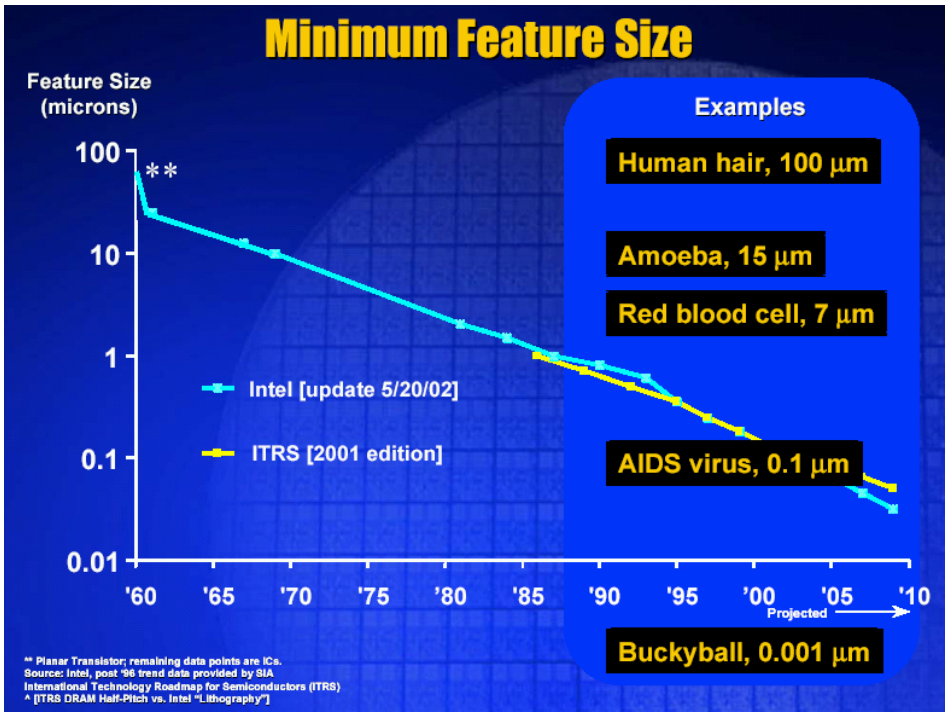


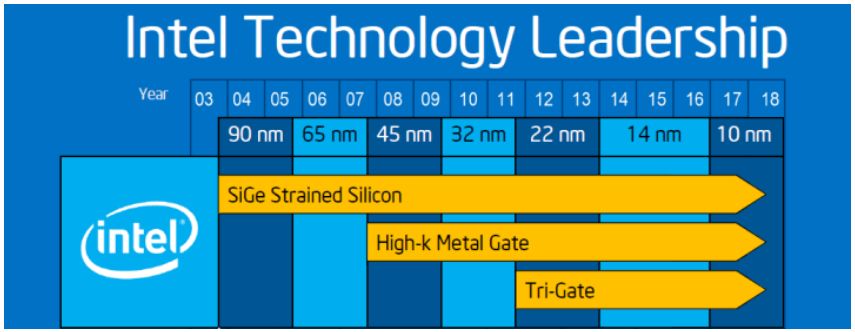


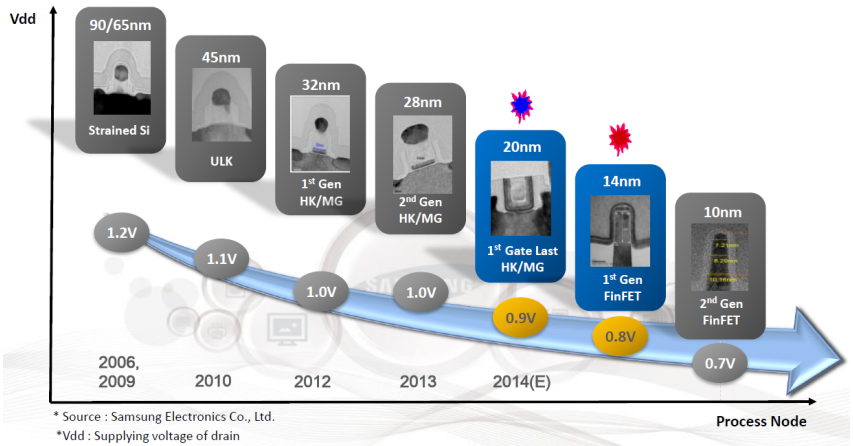
- The number of transistors grows quadratically with the reduction of their feature size
- Transistor speed increases linearly with the reduction of their feature size

Feature size (transistor size) (cont.)

It is measured in *nm* ($10^{-9}m$).



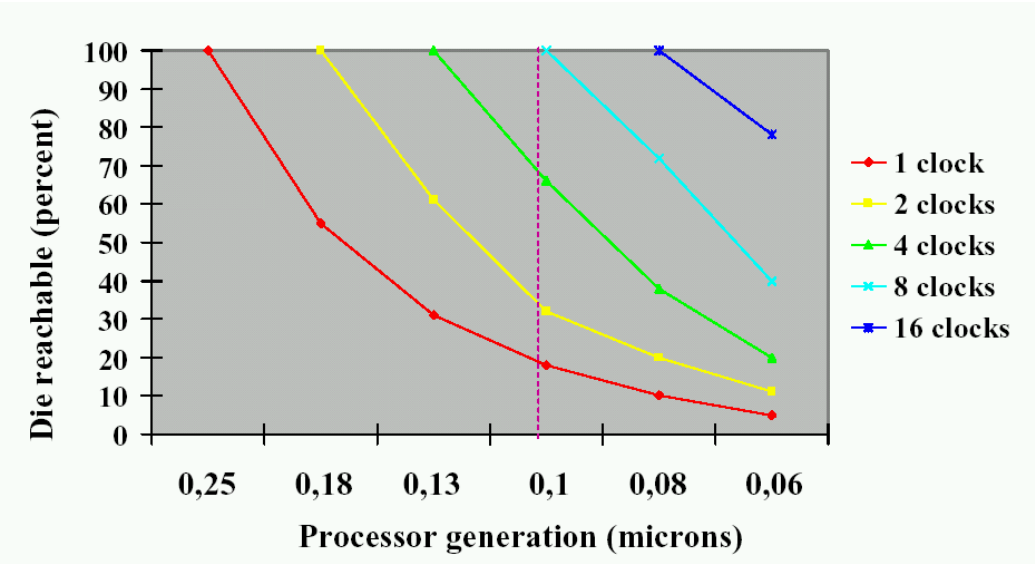




Rise of propagation delay in interconnections

Propagation delay relates to wire resistance and capacitance ($R \cdot C$).

- As feature size is reduced, wire section (width and hight) also does, thus increasing R
- Although capacitance of conductor surface decreases, capacitance coupling between wires (also named Interwinding capacitance) increases, thus augmenting C



$\rightarrow$ The fraction of the chip reachable in a single cycle is reduced as the feature size does: 100% with 0.25μm to 5% with 0.06μm.

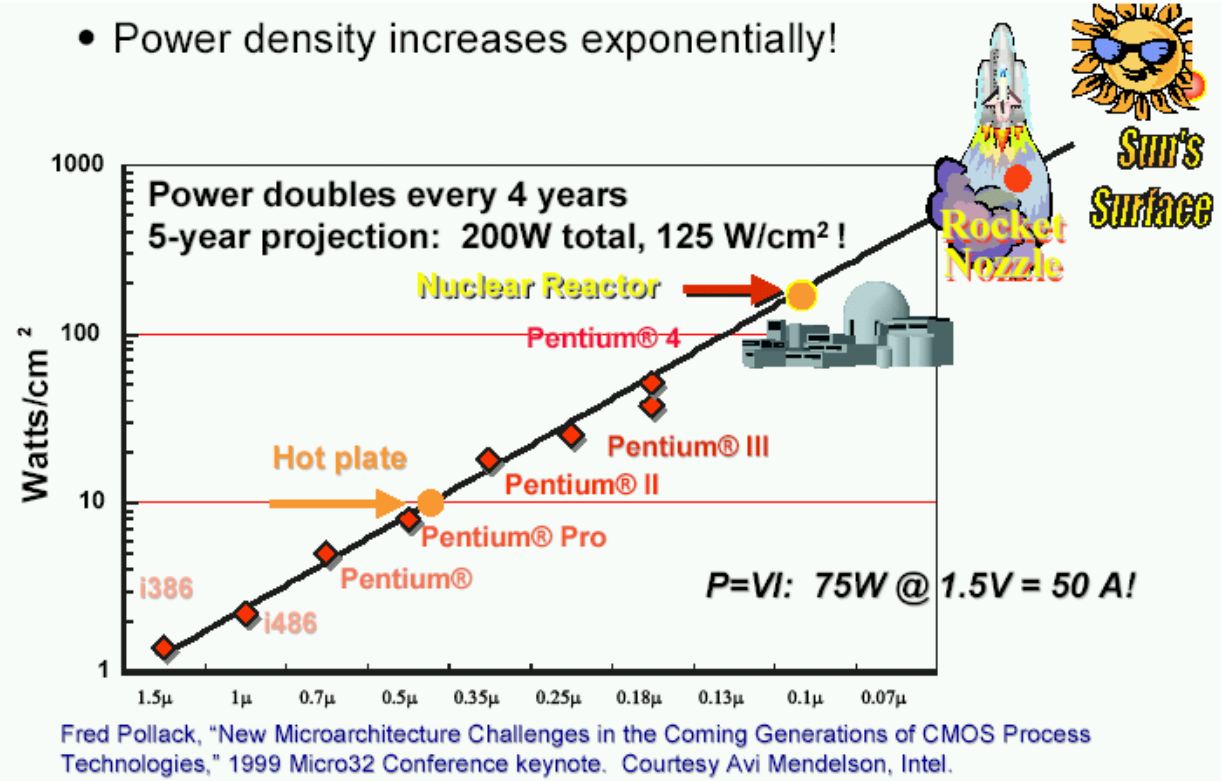
Power consumption and heat dissipation

Power per transistor = $\text{Pw}_{\text{dynamic}}$ + $\text{Pw}_{\text{static}}$

- $\text{Pw}_{\text{dynamic}} = \frac{1}{2} \cdot C \cdot V^2 \cdot f \rightarrow$ proportional to frequency (power switching)
  Solution: reduce supply voltage, but a minimum is required for each frequency
  - During last 24 years, supply voltage has been reduced from 12 to 1.1. V. This reduces $\text{Pw}_{\text{dynamic}}$ by a factor of $\frac{12^2}{1.1^2} = 119X$
  - But there is a minimum voltage value $\rightarrow$ remaining margin for reduction is low (from 1.1 to 0.7 V): $\frac{1.1^2}{0.7^2} = 2.5X$
- $\text{Pw}_{\text{static}} = I_{leakage} \cdot V$
  - As feature size decreases, $\uparrow I_{leakage}$
  - As the number of transistors increases, the $\text{Pw}_{\text{static}}$ also does.

Power consumption and heat dissipation (cont.)

The increase in the number of transistors and frequency prevails over the reduction in voltage supply and capacitance $\rightarrow$ from 0.01 W in first microprocessors to around 130 W in an Itanium2 processor.

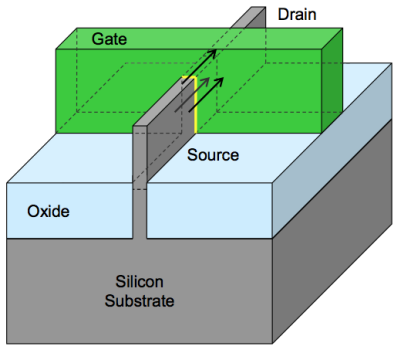


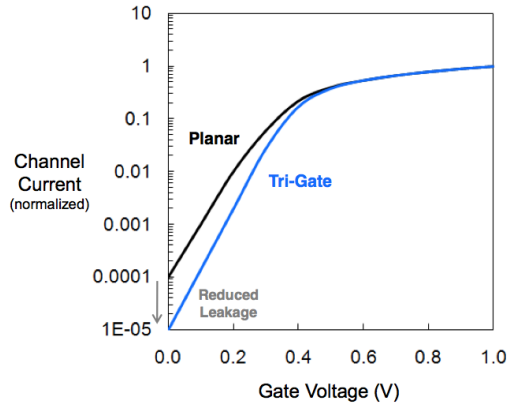
Power consumption and heat dissipation (cont.)

Implications:

- Microprocessor power (current) distribution. Modern microprocessors integrate hundreds of power supply pins.
- Heat dissipation and cooling
- Development of new materials to reduce the leakage current. Dielectric improvement (*high-k* or *Tri-Gate* from Intel).

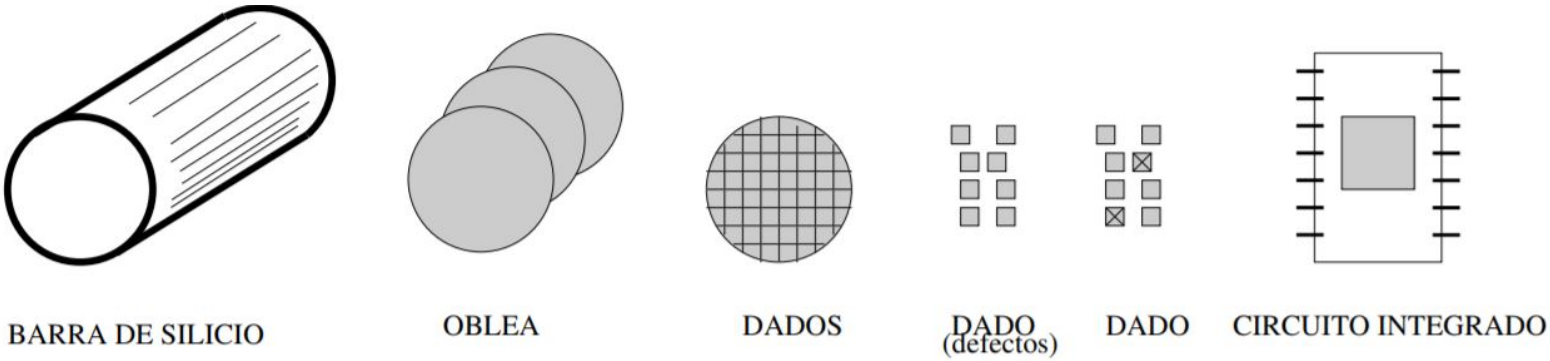
Tri-Gate Transistor:





Cost

- Cost of an integrated circuit.



Final cost $\approx$ f (die surface^4)
Die surface = f (design complexity)

Cost (cont.)

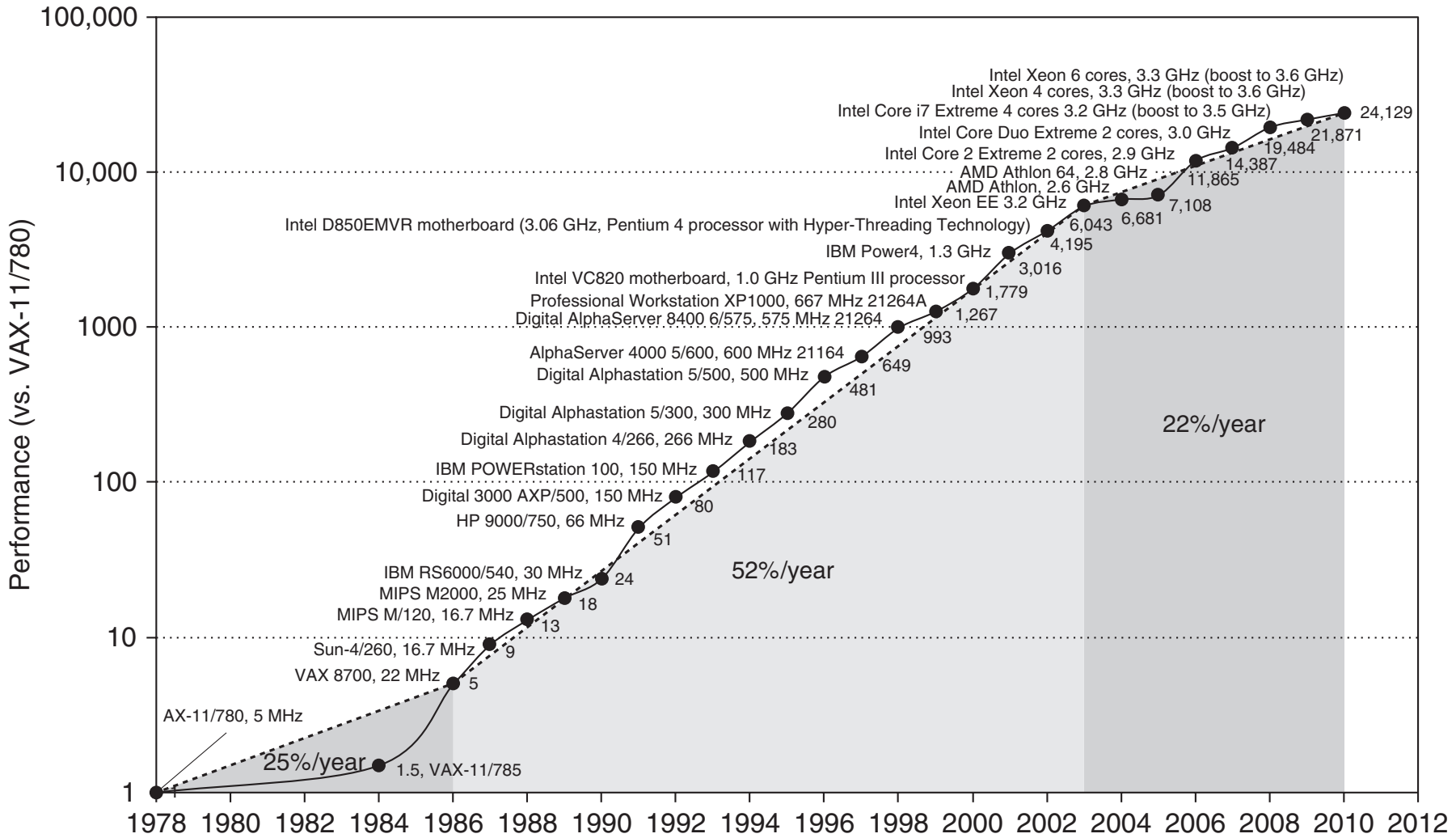
- Factors decreasing the cost of components:

  Learning curve: The cost of a component decreases over time, since throughput increases (faulty component production rate decreases)

  Sales volume: Doubling the sales volume decreases costs by 10%

- Design costs
  - Factory costs are inversely proportional to *feature sizes*.
  - Size of a design team is inversely proportional to *feature size*: from 3 people in the Intel 4004 design team to more than 300 in the ones designing modern processors.

Contents


Performance (vs. VAX-11/780)
100,000
10,000
1000
100
10
1
1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012
Intel Xeon 6 cores, 3.3 GHz (boost to 3.6 GHz)
Intel Xeon 4 cores, 3.3 GHz (boost to 3.6 GHz)
Intel Core i7 Extreme 4 cores 3.2 GHz (boost to 3.5 GHz)
Intel Core Duo Extreme 2 cores, 3.0 GHz
Intel Core 2 Extreme 2 cores, 2.9 GHz
AMD Athlon 64, 2.8 GHz
AMD Athlon, 2.6 GHz
Intel Xeon EE 3.2 GHz
Intel D850EMVR motherboard (3.06 GHz, Pentium 4 processor with Hyper-Threading Technology)
IBM Power4, 1.3 GHz
Intel VC820 motherboard, 1.0 GHz Pentium III processor
Professional Workstation XP1000, 667 MHz 21264A
Digital AlphaServer 8400 6/575, 575 MHz 21264
AlphaServer 4000 5/600, 600 MHz 21164
Digital Alphastation 5/500, 500 MHz
Digital Alphastation 5/300, 300 MHz
Digital Alphastation 4/266, 266 MHz
IBM POWERstation 100, 150 MHz
Digital 3000 AXP/500, 150 MHz
HP 9000/750, 66 MHz
IBM RS6000/540, 30 MHz
MIPS M2000, 25 MHz
MIPS M/120, 16.7 MHz
Sun-4/260, 16.7 MHz
VAX 8700, 22 MHz
AX-11/780, 5 MHz
1.5, VAX-11/785
24,129
21,871
19,484
14,387
11,865
7,108
6,681
6,043
4,195
3,016
1,779
1,267
993
649
481
280
183
117
80
51
24
18
13
9
5
1
22%/year
52%/year
25%/year

Period I (1978–1986). Performance grows at a rate of 25% per year, mainly due to technology enhancements.

Period II (1986–2003). Performance grows at a rate of 52% per year, due to technology enhancements and **architectural improvements:** RISC *(Reduced Instruction Set Computer)* architecture, pipelining, instruction level paralellism (ILP), caches, etc.

Period III (2003–2011). Performance grows at a rate of 23% per year. Limits in ILP, Memory latency, power consumption, and heat dissipation $\rightarrow$ performance improvement can only be achieved through parallelism.

Period IV (2011–2015). Performance grows at a rate of 12% per year due to technological and parallelism limitations (Amdahl's law).

Period V (2015–2018). Negligible performance growth (6.5% per year)

- End of Moore's law
- Technology does not improve energy efficiency
- Reaching the limits of parallelism

Current situation:

- Performance and energy efficiency through specialization $\rightarrow$ use of domain-specific accelerators, such as Google Tensor Processing Units (TPU) multiplying per 80 the performance/watt wrt a CPU in the context of neural networks-based inference.
- Research and Development of new technologies, as in: https://bit.ly/2lU7DN9.

Architectural improvements

Some examples:

RISC architecture: Simple instructions that execute very fast. Simple hardware.

Pipelining: Splitting the instruction cycle into phases that are executed concurrently.

ILP exploitation: Execution of instructions out of program order.

Parallelism:

- DLP *Data-Level Paralellism*. Performing an operation on multiple data.
- TLP *Thread-Level Paralelism*. Performing several tasks in parallel.

Many of these improvements are only possible when the number of available transistors is high enough.

Some architectural ideas are motivated by the technology available

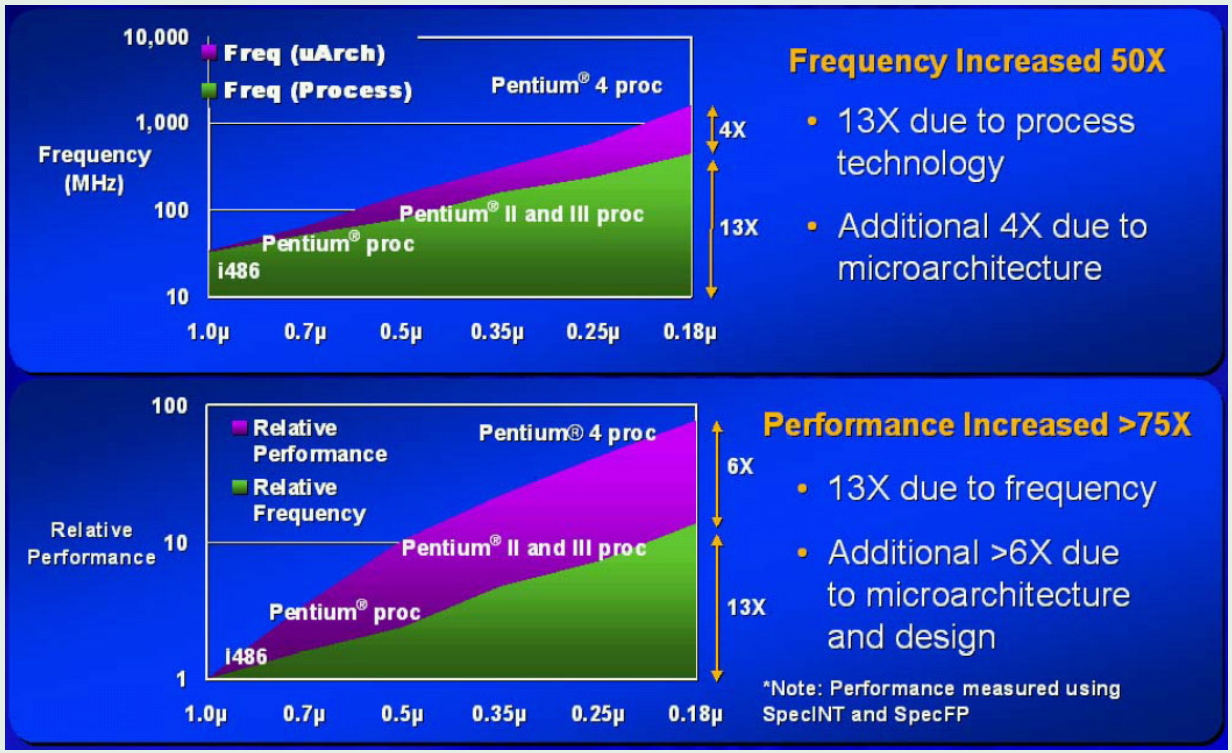
Some examples:

Caches The speed difference between the processor and the memory motivated the introduction of the *cache* memory between the processor and memory. The continuous growth of this difference has motivated the use of various levels (L1, L2, L3) of *cache* in the designs.

Stages to propagate signals The increasing wire delay motivated the inclusion in the Pentium 4 of two pipeline stages intended only to propagate signals through the wires.

Chip multiprocessors A sophisticated processor that works at very high frequency consumes a lot of power and dissipates a lot of heat. Instead, we can put several simpler processors that operate at lower frequency and lower voltage.

Example: Intel processors



→ The architectural improvements delivered a speedup of 7X over the one that would have been achieved with only technological improvements.

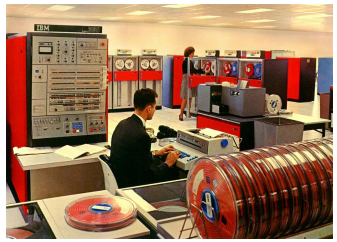
Contents

Types of computers

Computer market evolves as available technology and society consumption habits do.
Computer industry adapts its offer to this evolution by offering different types of devices.
For instance, during the 70's, as first microprocessors were under construction, two types of computers were commercially available:

Mainframes: very big and expensive computers only affordable to big corporations.

Minicomputers: medium-size computers, largely used at universities.



Types of computers (cont.)

Since 2012, thanks to advances in microprocessor design, communications and human–machine interfaces, the application domain of computers has became wider:

- Processors and memories have become small enough to be integrated in the design of mass-market electronic devices, some of them portable.
- A computer can integrate a large number of processors with memory that collaborate to improve the overall system performance.

Currently, most relevant computers are: Mobile devices, Embedded Systems, Personal computers, Servers, Clusters, Supercomputers

Personal Mobile devices

Include (but not limited to): smartphones, tablets, PDAs, etc.
Main features:

- Very limited energy consumption
  - Battery powered
  - No forced cooling
- *Responsive* and *predictable* design
  - Need of multimedia GUIs
  - Video frames and audio block must be processed on time (*real-time* requirements).
- Reduced main memory capacity
  - Optimized code (small footprint)
- Secondary memory of *flash* type

Embedded systems

Include: automotive and aerospace embedded systems, appliances, (air, marine, space) navigation, etc. ...
Mobile devices are a concrete case of embedded system.
Differences wrt mobile devices:

- Wide spectrum of design and performance requirements
- Specific software developed by manufacturers (no third-party code)

Personal computers

Examples: Desktop, Laptops, Netbooks.
Main features:

- Well-balanced numerical and graphical computing power.
- Optimal relation between performance and cost
- Wide spectrum of configurations

Servers

Computer providing services (data, email, printing, . . .) in a network.
Use to be part of the computer infrastructure of a corporation.
Main features:

- Availability is critical
- Scalable design
- Design for *responsiveness*
- Throughput is a key attribute

Cluster

It is a collection of computers, each one with its own operating system, interconnected through a network. Is is externally seen as a single computer.
Main Features:

- Significative investment in power supply and cooling
- Scalability: The system can be easily adapted to changes (typically peaks) in workloads

Large Scale Clusters

Provide support to the big internet services: social networks, search engines, e-commerce and file sharing platforms, etc.
Main features:

- Need of a huge Internet bandwidth and secondary storage
- Good balance between performance and cost.
- Dependability-oriented design: Availability is critical, but low-cost components are used $\rightarrow$ need of redundancy

Supercomputer

Machines designed for high performance, despite their cost.
Main features:

- Execution of large scientific distributed applications with limited user interaction (weather forecasting, protein folding, . . .)
- Very high floating point arithmetic throughput
- Cluster-based supercomputers are becoming more and more common

Contents

¿Interesado en Aprender Más?

- Título oficial, 1 año de duración (60 ECTS).
  - Más información en http://mic.disca.upv.es.
- Asignaturas relacionadas:
  - Arquitectura y Tecnología de los Procesadores Multinúcleo
    - Claves de las prestaciones de los procesadores multinúcleo.
    - Diseño de los procesadoes multinúcleo actuales.
    - Conceptos avanzados sobre caches compartidas.
    - Memoria principal con controladores de memoria compartidos.
  - Redes en Chip
    - Diseño avanzado y construcción de redes en chip.
    - Prestaciones de las redes en chip.
    - Encaminamiento, reconfiguración, topologías y conmutación.
    - Tecnologías futuras y ejemplos actuales de redes en chip.

¿Interesado en Aprender Más? (cont.)

- Arquitectura de Redes de Altas Prestaciones
  - Diseño y construcción de redes de altas prestaciones.
  - Técnicas de control de flujo, tráfico, tolerancia a fallos, etc.
  - Reducción del consumo en redes de altas prestaciones.
  - Diseño de routers.
- Configuración, Administración y Utilización de Clusters
  - Diseño y configuración de clusters.
  - Sistemas de almacenamiento.
  - Equilibrado de carga.
  - Clusters de alta disponibilidad.