MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

IN COLLABORATION WITH MIT MANAGEMENT SLOAN SCHOOL | UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

MASTER MEIM 2022-2023

DIGITAL TECH

High Performance Computing

Lesson 1

Prof. Livia Marcellino

Prof. of High Performance Computing, Università degli Studi di Napoli Parthenope

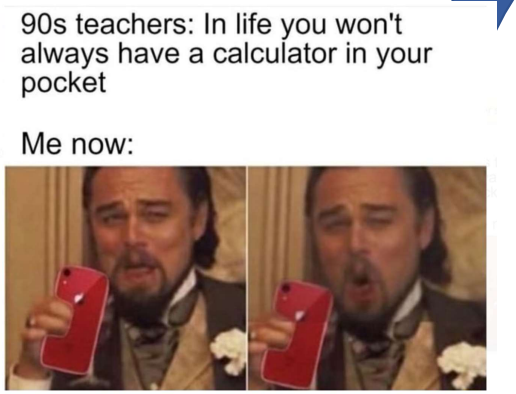
MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

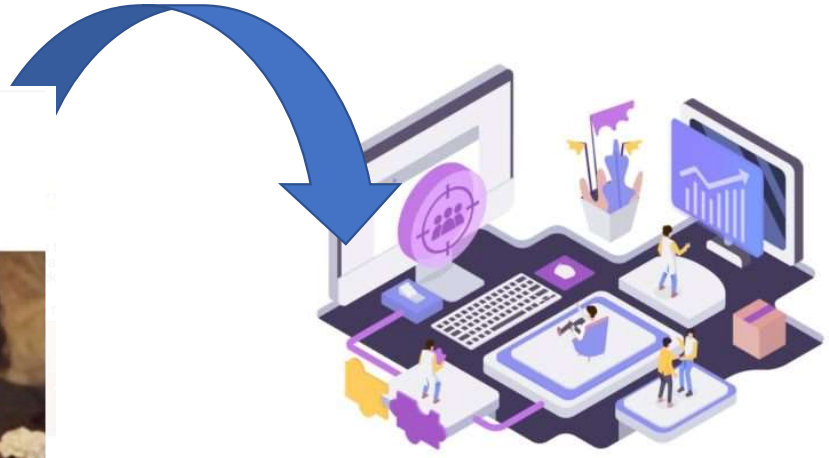
UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

The need for fast computing architectures

Computer

The purpose of using a computer is to use a machine which does computation instead of us





The purpose of a

super computer

is to solve problems

supercomputing :

solve a problem using a supercomputer

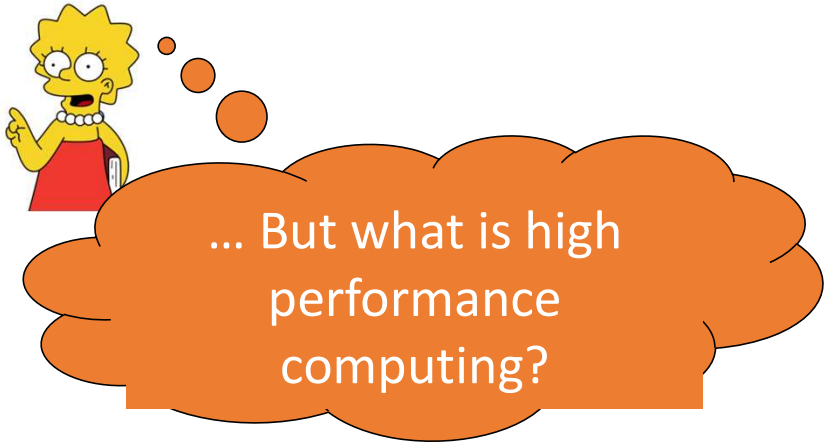
But…
… is it really necessary to do high performance computing?

Computers today aim above all at the speed of applications…

… supercomputers are essential!

High Perfromance Computing

From many years, high-performance computing has helped to improve the quality of life by modeling and predicting a large range of physical properties and phenomena with speed and precision.

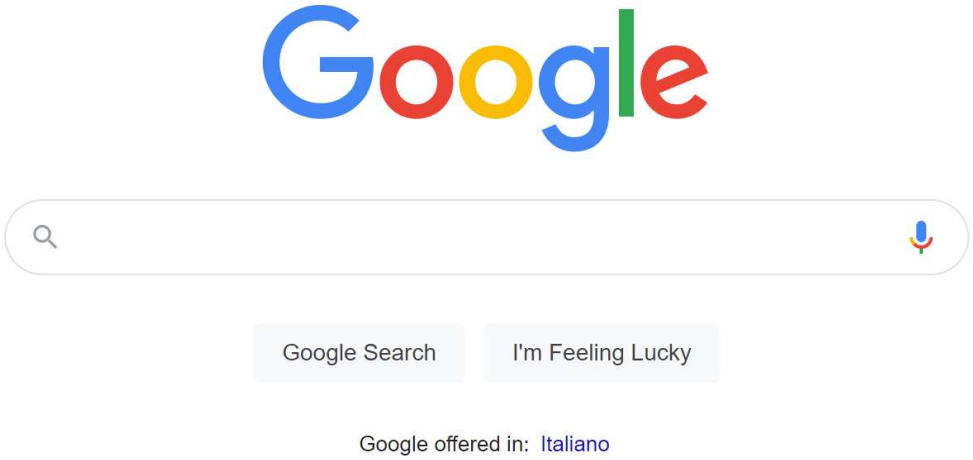


HPC – High Performance Computing is the recent version of supercomputing, i.e. a scientific computing tool that uses algorithms, software and modern hardware (supercomputers) to provide real-time solutions to large-scale problems.

MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Internet searching…

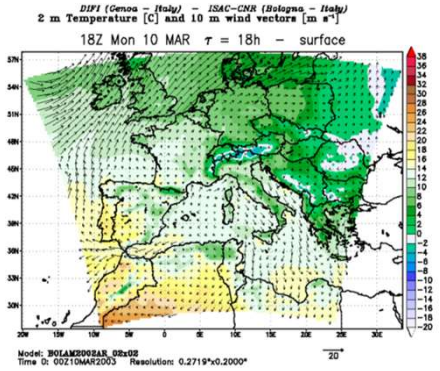


Million of people query on Google every day

MEIM MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

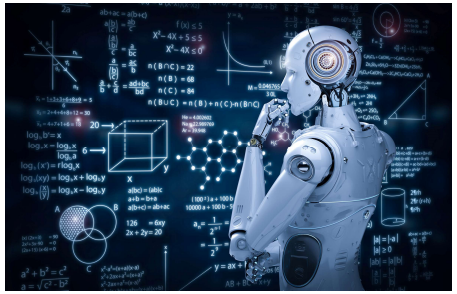
UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Weather forecasts



Financial data mining

Machine Learning



whatever the application… it is big data mining

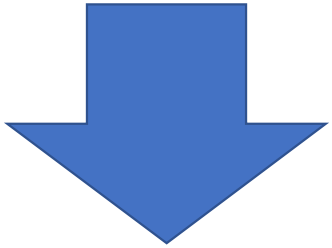
Big Data Problems

- Search on the Internet
- Automatic Planning
- Advertising and Marketing
- Banking and financial services
- Media and Entertainment
- Meteorology
- Health Care
- Cyber Security
- Training

Problems characterized by the need to obtain real-time solution (or just in time!)

Supercomputing, in order to

solve “large scale” (or big data) problems



solve big data problems in "real time” (or useful time)

MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

How to increase software performance?

How to increase performance?

The time required to run a software depends not only on the number of operations to be done but also on how many operations per second the computer can perform !!!!

The way to write software has been known for hundreds of years. A machine language code executable by the computer i.e. a program that encodes a serial procedure of instructions (algorithm).

the software run time:

$$\tau = k \cdot T(n) \cdot \mu$$

k is a multiplicative constant which can depend on practical and stochastic factors:
memory access, CPU occupation for other applications

depends on the algorithm

T(N) = algorithm computational complexity

depends on the computer

μ = execution time of 1 f.p. operation

How to reduce τ ?

In recent years the biggest challenge has been to reduce this amount!

$$\tau = k \cdot T(n) \cdot \mu$$

by reducing T(n)

or

optimizing the algorithm

To reduce T(n)

It is possible to prove

(complexity theory of algorithms)

that for some classes of problems there are algorithms with minimal computational complexity

(optimal algorithms)

To reduce T(n)

However, in general, it is good…

Numerical Analysis

… always to reorganize the algorithm

in order to obtain the most possible algorithm with

minimum computational complexity

"The fundamental law of computer science: As machines become more powerful, the efficiency of algorithms grows more important, not less."
[N.Trefethen]

How to reduce τ ?

$$\tau = k \cdot T(n) \cdot \mu$$

μ = execution time of 1 f.p. operation

reducing μ

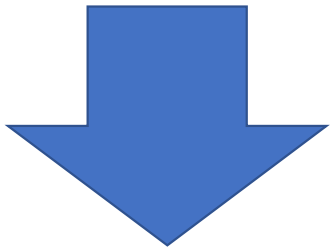
i.e.

improving technology

(miniaturization process)

How to improve the technology?

How did we come to build supercomputers?

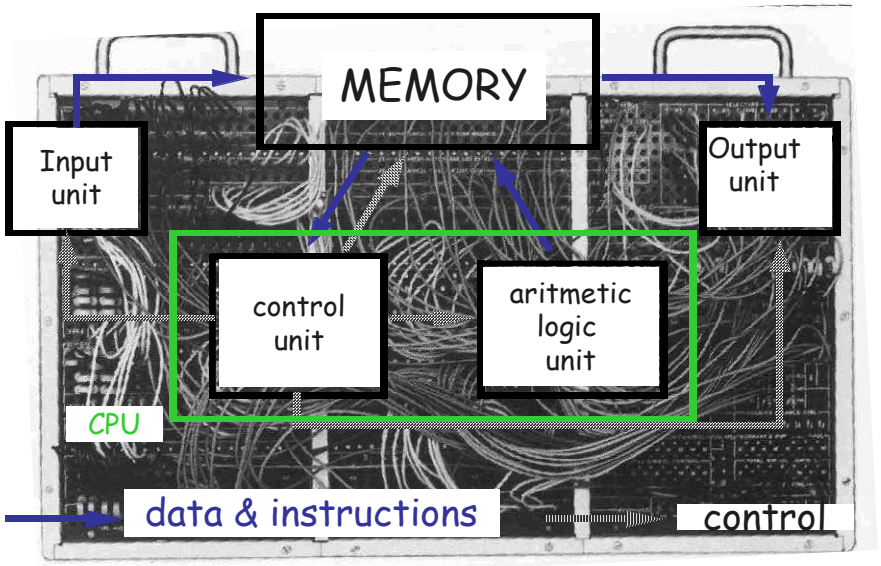


almost 80 years ago, in **1945**,
the first computer built was a supercomputer!

Von Neumann machine

In **1945** John Von Neumann introduced an electronic computer scheme, based on the concept of "memorized program":

the instructions are recorded in the "memory" in numerical form.



And it is still the basic scheme of current computers

MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

Evolution of computer machines

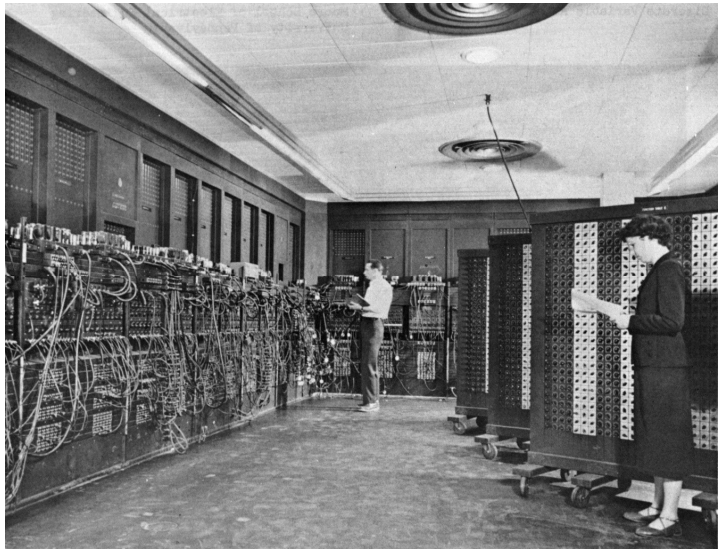
First type of computer

ENIAC (1946)

*(**E**lectronic **N**umerical **I**ntegrator **A**nd **C**omputer)*

uses

Thermoionic valves



It occupied a room of 9x30 square meters
It weighed more than 30 tons

First generation computer

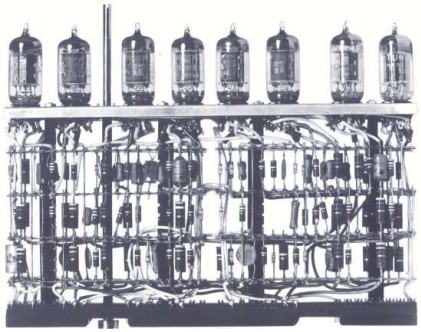
(Thermionic valves: triodes)

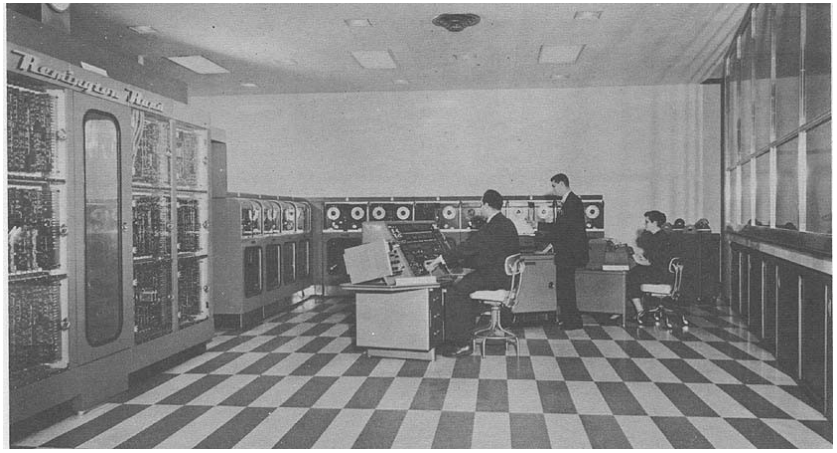
UNIVAC-I (1950)

*(**UNIV**ersal **A**utomatic **C**omputer I)*

First commercial computer

based on thermionics valve technology





It weighs 5 tons. The central unit is more than 5 meters long and 2.5 meters high.

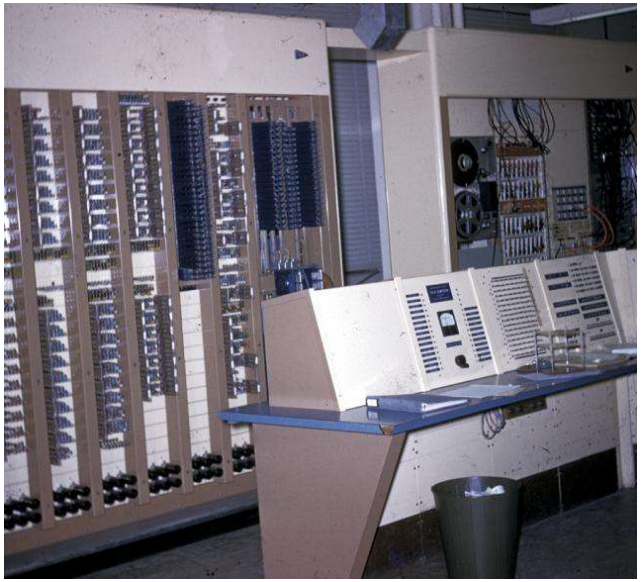
It was used in **1952** for the first exit polls of the Americans presidential elections. It correctly anticipated Eisenhower's winnings.

Second type of computer

TX-0 (1956)

*(**T**ransistorized e**X**perimental computer **zero**)*

Experimental computer which uses **Transistor**

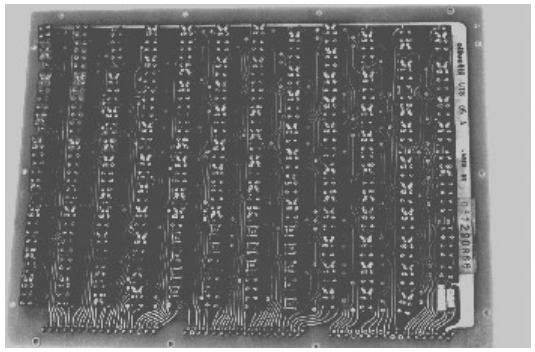


Second generation computer
(transistor)

Siemens 2000 (**1957**)

First commercial computer

based on transistor technology.



Integrated Circuits - Chip

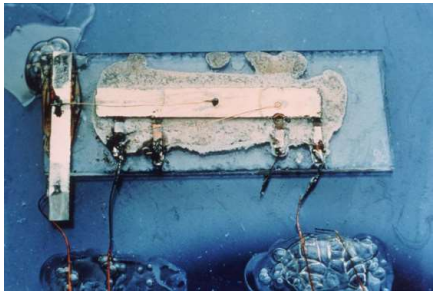
… idea:

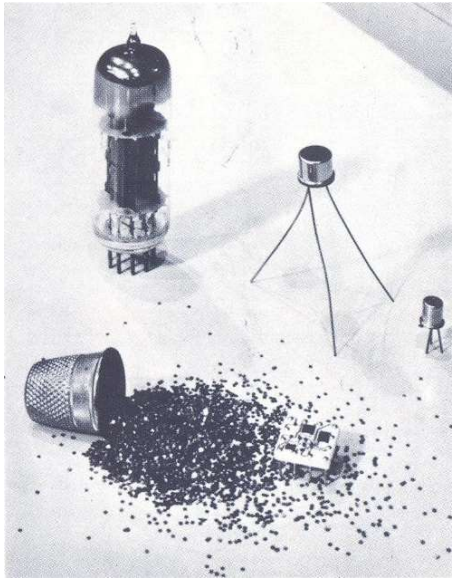
put more transistors on

a plate of germanium

as big as

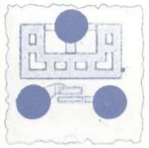
a postage stamp (**1958**)





50000 transistor of IBM360 compared to old genration

They can switch in $\mathbf{1/10^9}$ seconds

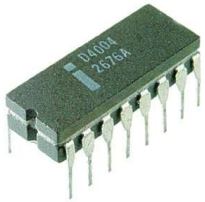


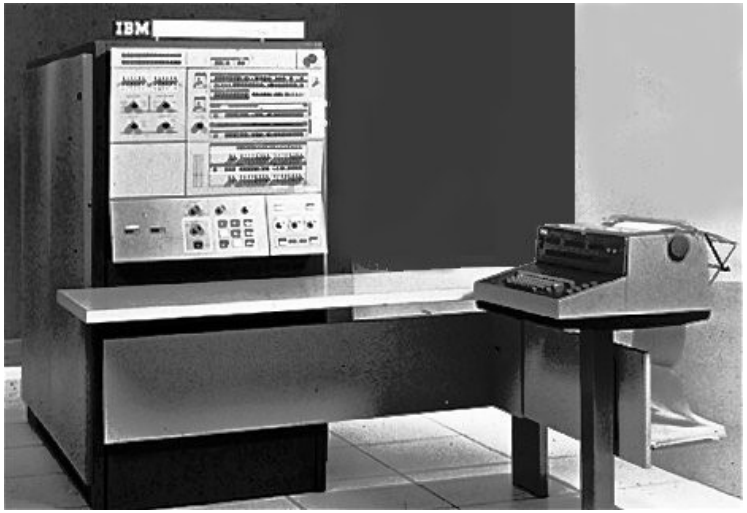
Third type of computer

IBM 360 (1964)

First computer with

Integrated circuits (Chip)

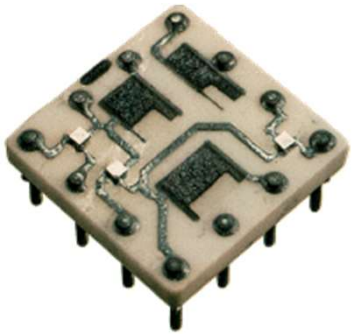


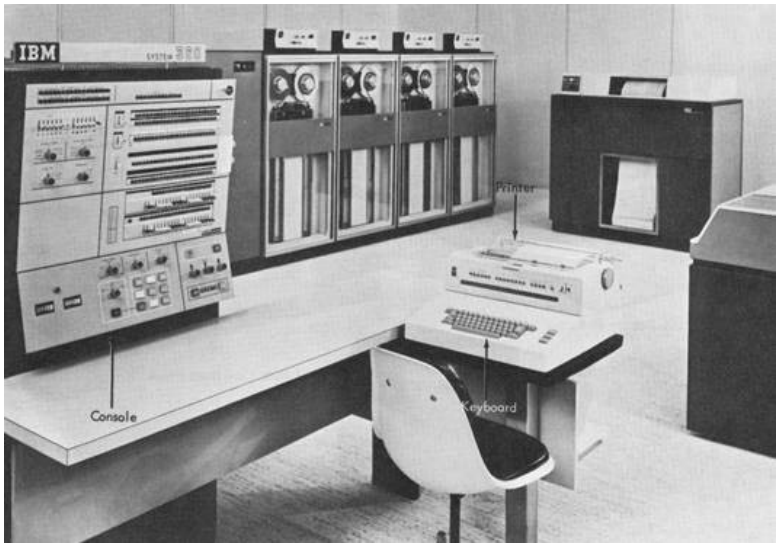


Third generation computer
(chip)

IBM System/360 (1965-1978)

First commercial computer based on Chip technology.





integrated circuits (Chip)

more transistors on

a **plate of germanium**

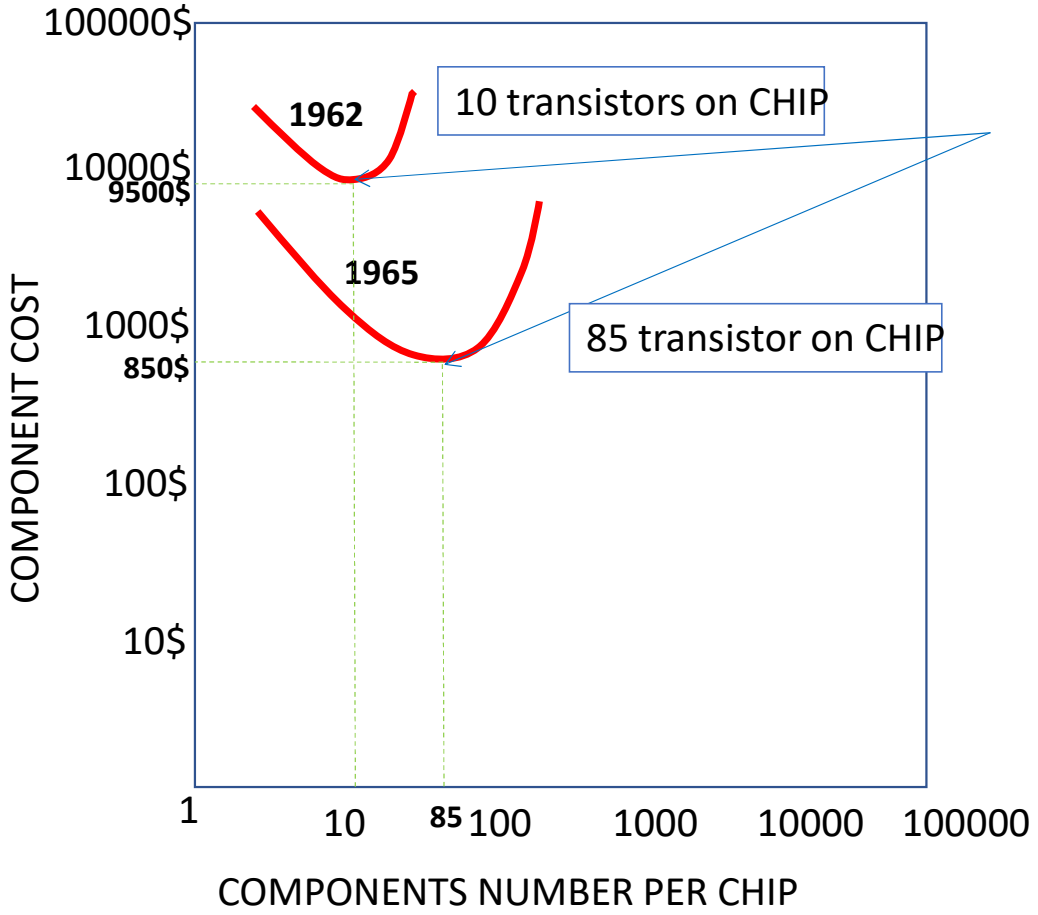
as big as

a postage stamp (**1958**)

How many transistors on each chip?

At this point…

… OBSERVATION OF DATA (1965):



MEIM | MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

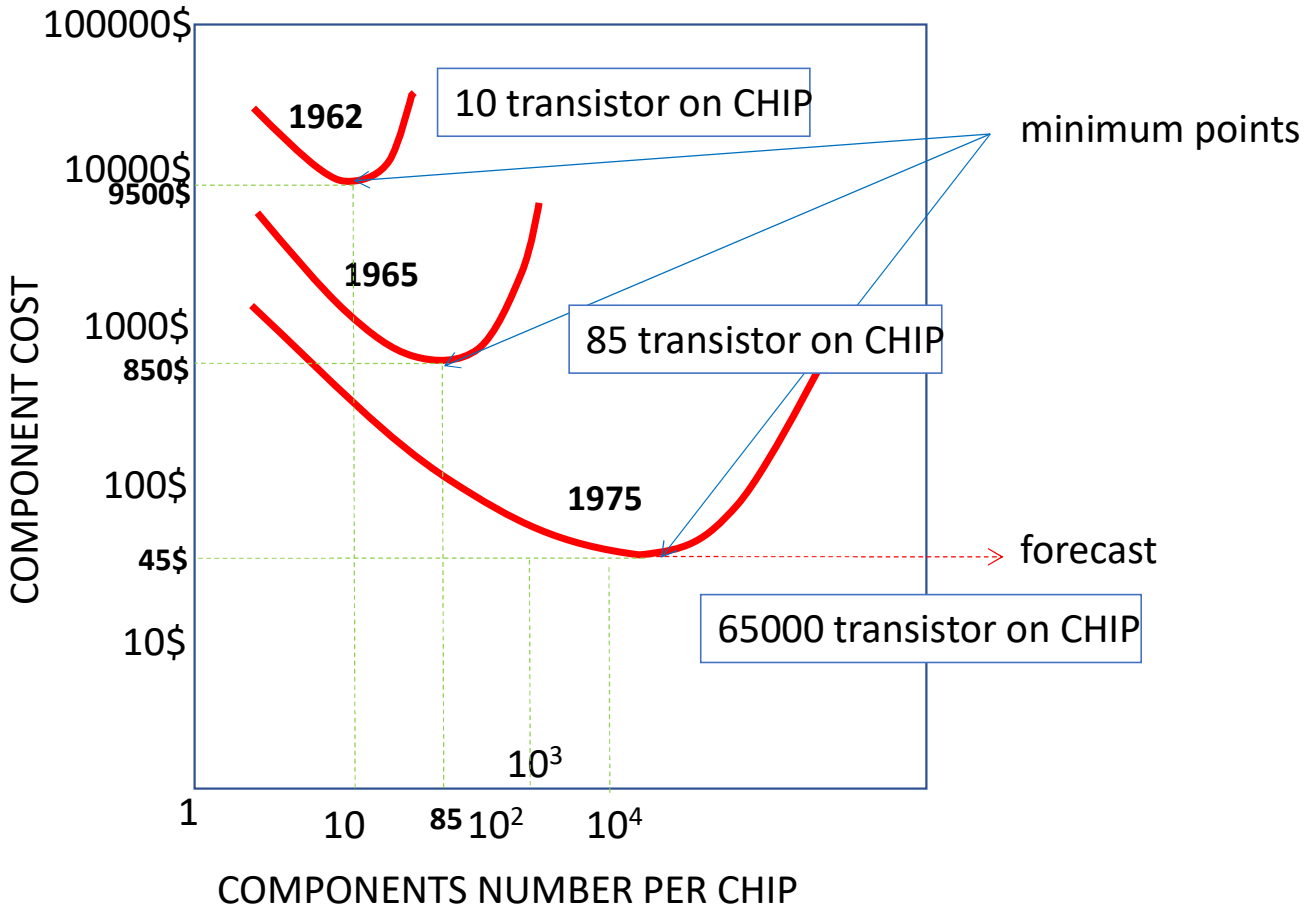
UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

The Moore's Law

Moore's observation (1965)

"(…) The number of transistors on CHIP has doubled from year to year.(…)"

MOORE forecast (1965):



Moore's prediction (1965)

"(…) in 1975 the number of transistors on a chip will be 65,000. (…)"

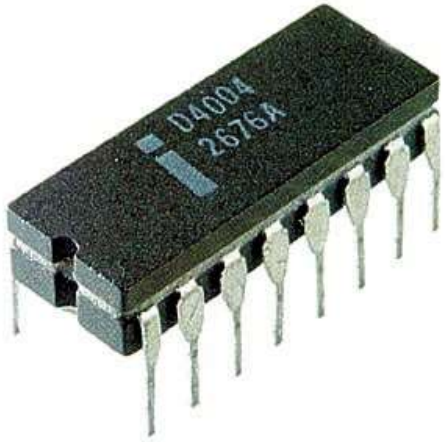
"The number of transistors will double every two years."

The first microprocessor

Confident in his prediction, in **1968**, Gordon Moore founds Integrated Electronics Inc (better known as Intel).

In **1971** the Intel created the first **microprocessor**, made entirely on a single chip.

Among the fathers of the microprocessor there is the Italian **Federico Faggin** who is responsible for its design.

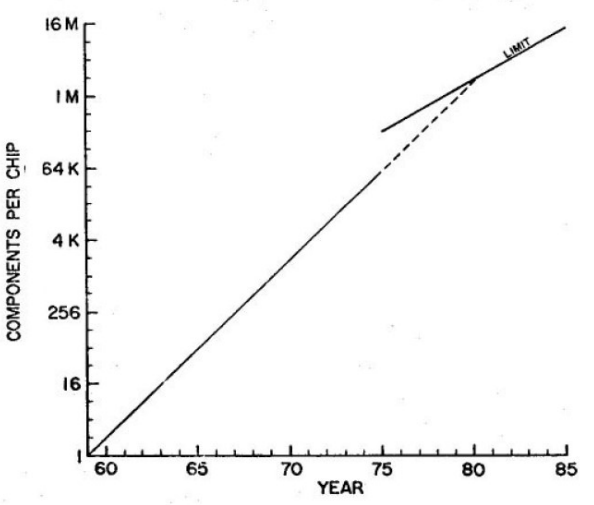


Intel 4004

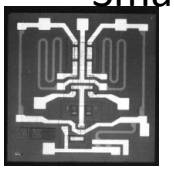
Moore's Law

Analyzing the technological evolution of the following years, in **1975** Moore slightly changed his prediction:

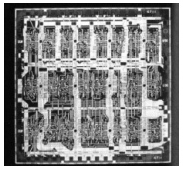
"… The computing power of microprocessors will double every 18 months … "



Moore's Law



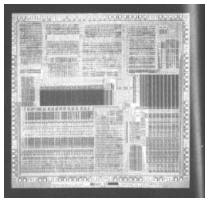
Small Scale Integration
(SSI)
5 transistor
(1964)



Medium Scale Integration
(MSI)
180 transistor
(1968)

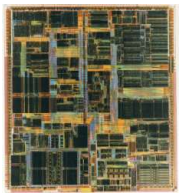
Large Scale Integration
(LSI)
10000
transistor
(1976)

This law has been reflected in practice for over 40 years and has established a **virtuous cycle**, pushing technological advances towards better and cheaper products, which in turn push the creation of new applications, which again encourage technological advancement, and so on.



Very Large Scale Integration
(VLSI)
132000
transistor
(~1983)

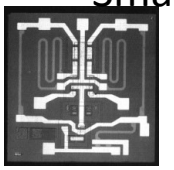
Born of PCs



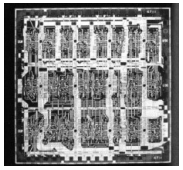
Ultra Large Scale Integration
(ULSI)
7500000
transistor
(1997)

Let's take a break

Moore's Law



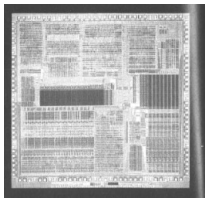
Small Scale Integration
(SSI)
5 transistor
(1964)



Medium Scale Integration
(MSI)
180 transistor
(1968)

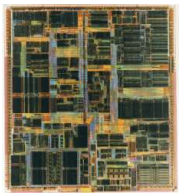
Large Scale Integration
(LSI)
10000
transistor
(1976)

This law has been reflected in practice for over 40 years and has established a **virtuous cycle**, pushing technological advances towards better and cheaper products, which in turn push the creation of new applications, which again encourage technological advancement, and so on.



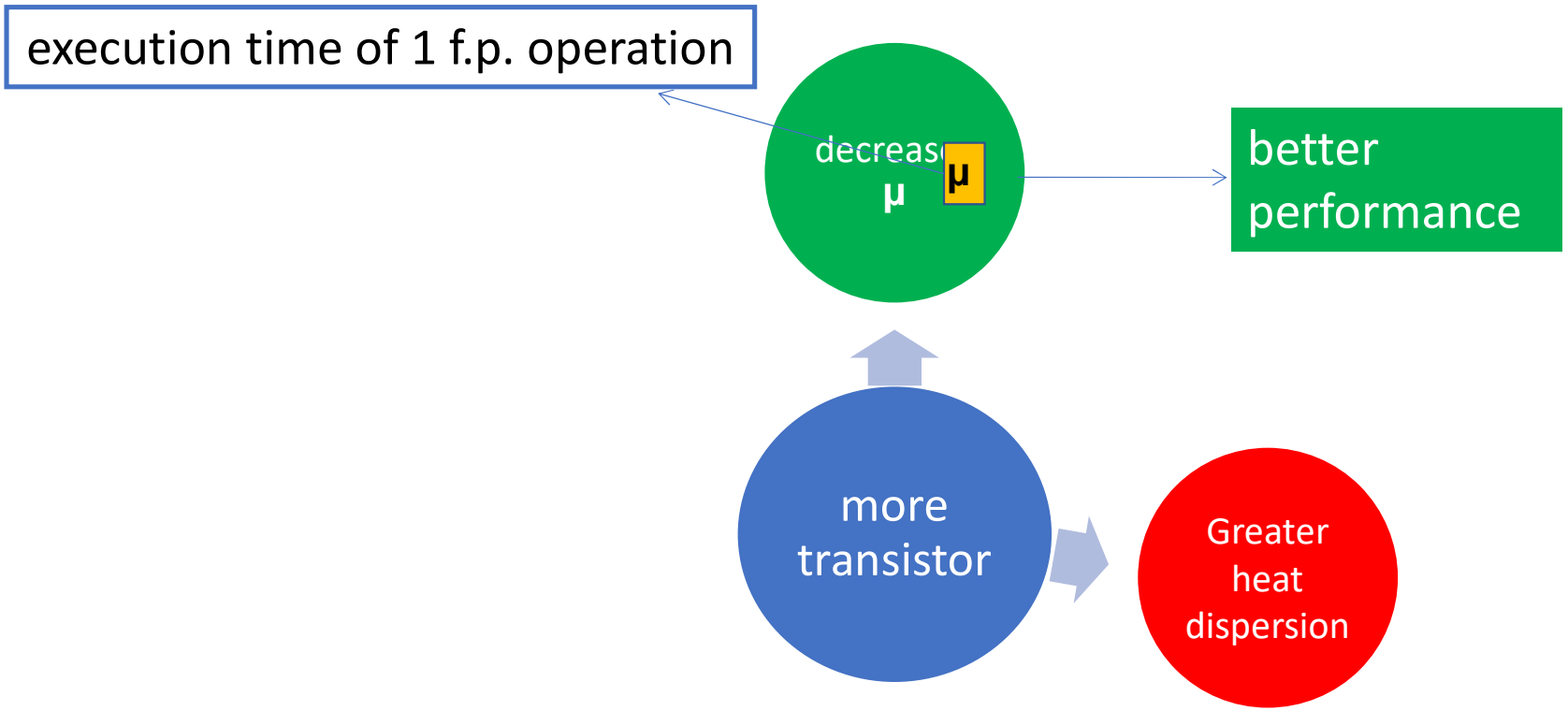
Very Large Scale Integration
(VLSI)
132000
transistor
(~1983)

Born of PCs



Ultra Large Scale Integration
(ULSI)
7500000
transistor
(1997)

However… more transistors

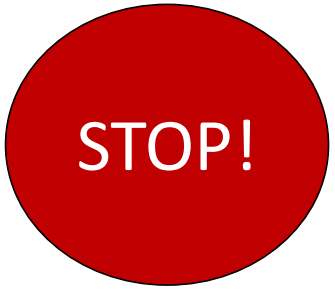


To reduce μ: minimize distances

problems of packaging, cooling and dispersion

The miniaturization process cannot proceed indefinitely!

TECHNOLOGICAL LIMITS



MEIM MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

Quantum computers

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

computer with a revised memory (first idea 1980)...

different data storing (different numerical representation), faster access to them.

Quantum algorithms, designed to run on **quantum computers**.

In February 2019, first commercial quantum computer.

Despite the advantages obtained, these need important conceptual changes on standard algorithms.

This still represents a limitation today!!!

How to reduce τ ?

$$\tau = k \cdot T(n) \cdot \mu$$

μ = execution time of 1 f.p. operation

C:
Parallel Computing

$$\tau = k \cdot T_{np}(N) \cdot \mu$$

A: reducing T(n) to optimize the algorithm

STOP!

B: reducing μ to improve technology

STOP!

MEIM MASTER IN ENTREPRENEURSHIP INNOVATION MANAGEMENT IN COLLABORATION WITH MIT SLOAN

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

The parallel paradigm:
temporal parallelism

Parallel Computing

Parallel computing is an evolution of serial computing that attempts to emulate what often happens in the natural world: multiple complex and interrelated events happening at the same time.

The idea of parallel computing is based on the simultaneous use of multiple computing resources to solve a single problem, breaking it into discrete parts that can be processed simultaneously, i.e. that can be performed serially on different CPUs.

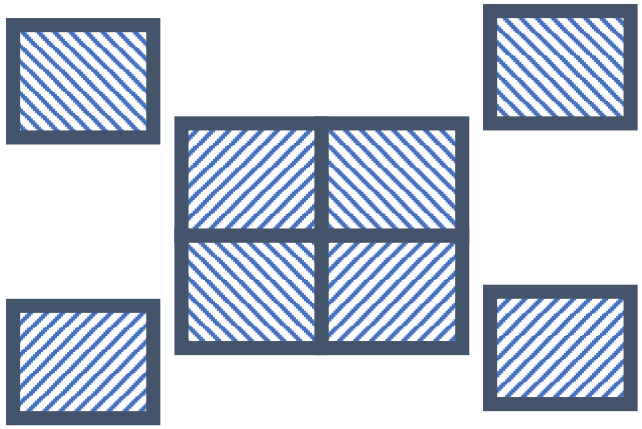
PARALLEL COMPUTING

Decompose a problem

in more subproblems

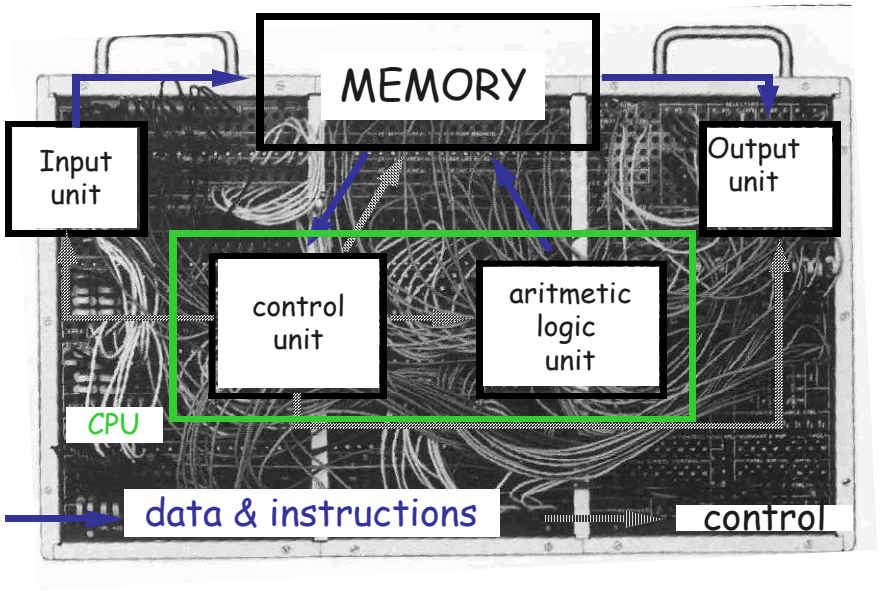
and solve them at the same time

with more processing units!



Need to create machines that can distribute the work among them
hardware development

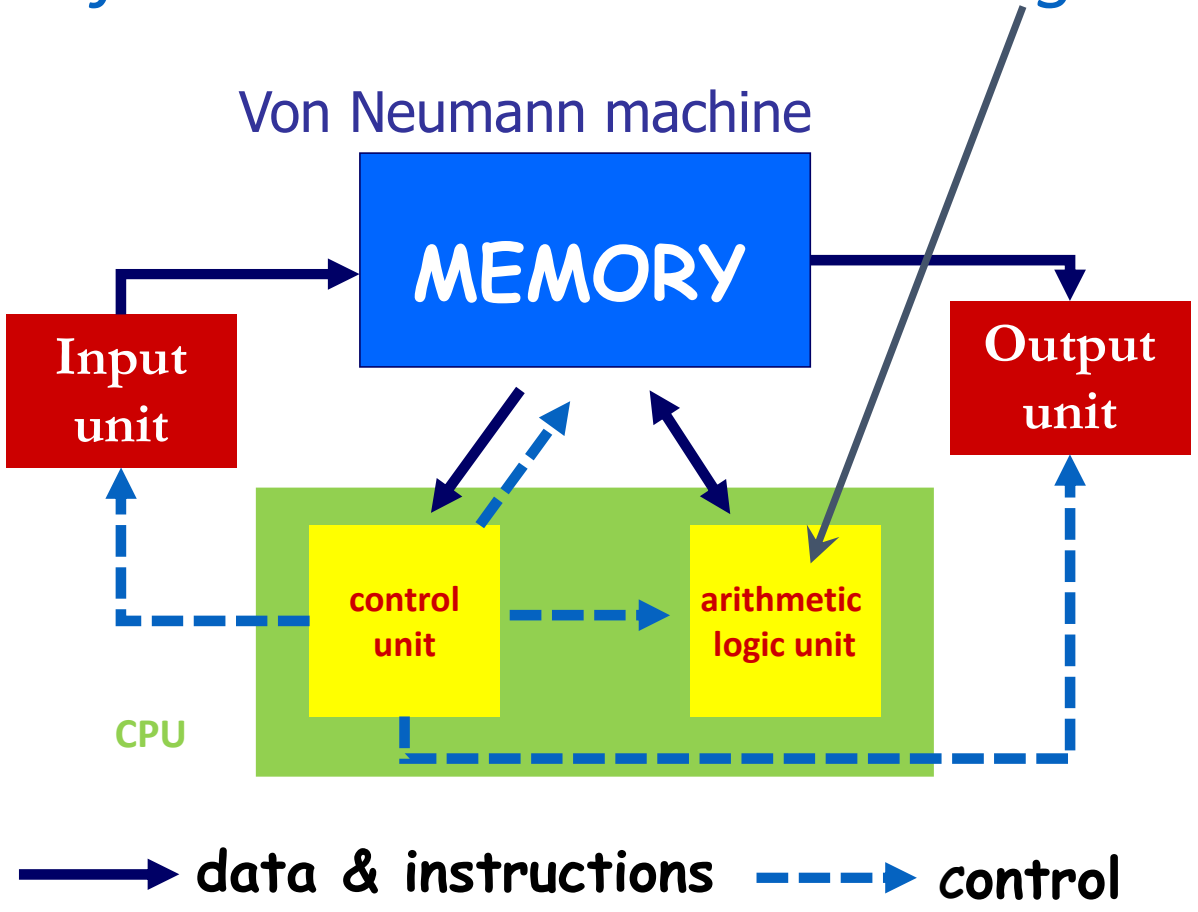
Von Neumann machine



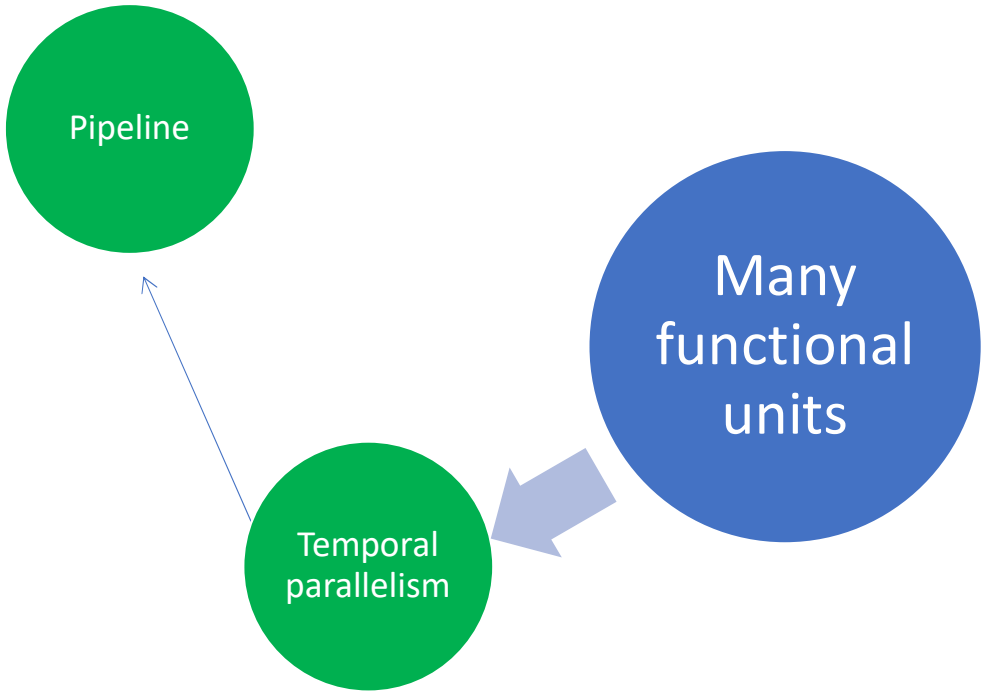
How the parallelism has been implemented on machines over the years?

parallelism (on-chip)

(multiple functional units within a single ALU)

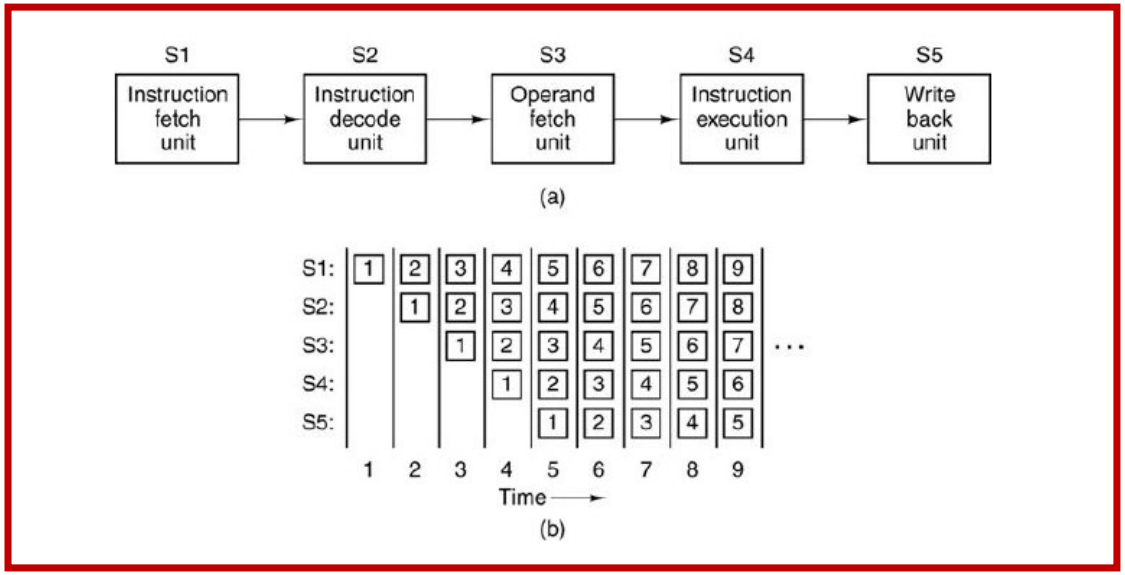


Temporal parallelism



First type of parallelism

Assembly line technique (pipeline)



Temporal parallelism

pipelining

Assuming that the different phases of the operation are separable, they are assigned to different functional units, so that when the first unit finishes our work for a phase of operation, it can be dedicated to the next one.

Architectures with *pipelined* functional units

The use of pipelined functional units is also the basis of:

array processor

capable to operate efficiently

on data structured in array format

Architectures with *pipelined* functional units

The first pipelined system was an **IBM System 360/91** (**1966**):

thanks to the pipeline it obtained an increasing of **33%** in performance!

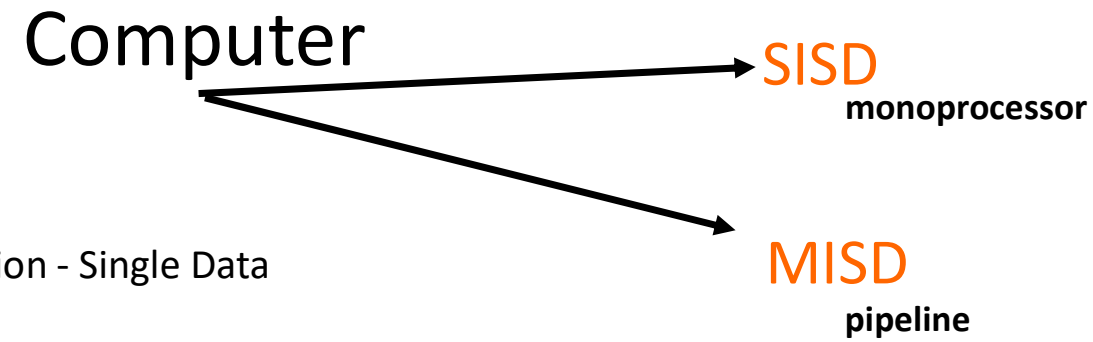
The first microprocessor to use a pipeline:

MOS Technology 6502 (**1975**)

Currently
all microprocessors use a pipeline
structure to improve their performance.

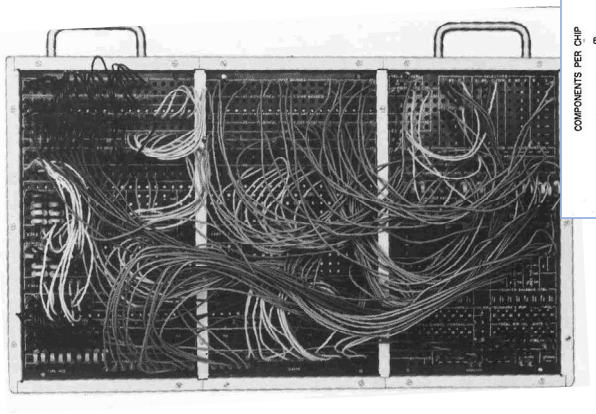
Flynn's taxonomy (since 1966)

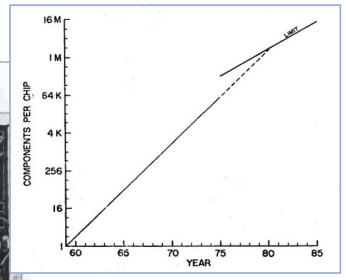
Michael J. Flynn begins to classify computers...

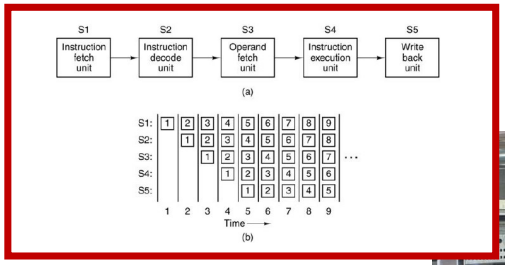


SISD: Single Instruction - Single Data

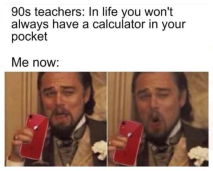
MISD: Multiple Instruction - Single Data







That's all for today!



$$\tau = k \cdot T_{np}(N) \cdot \mu$$