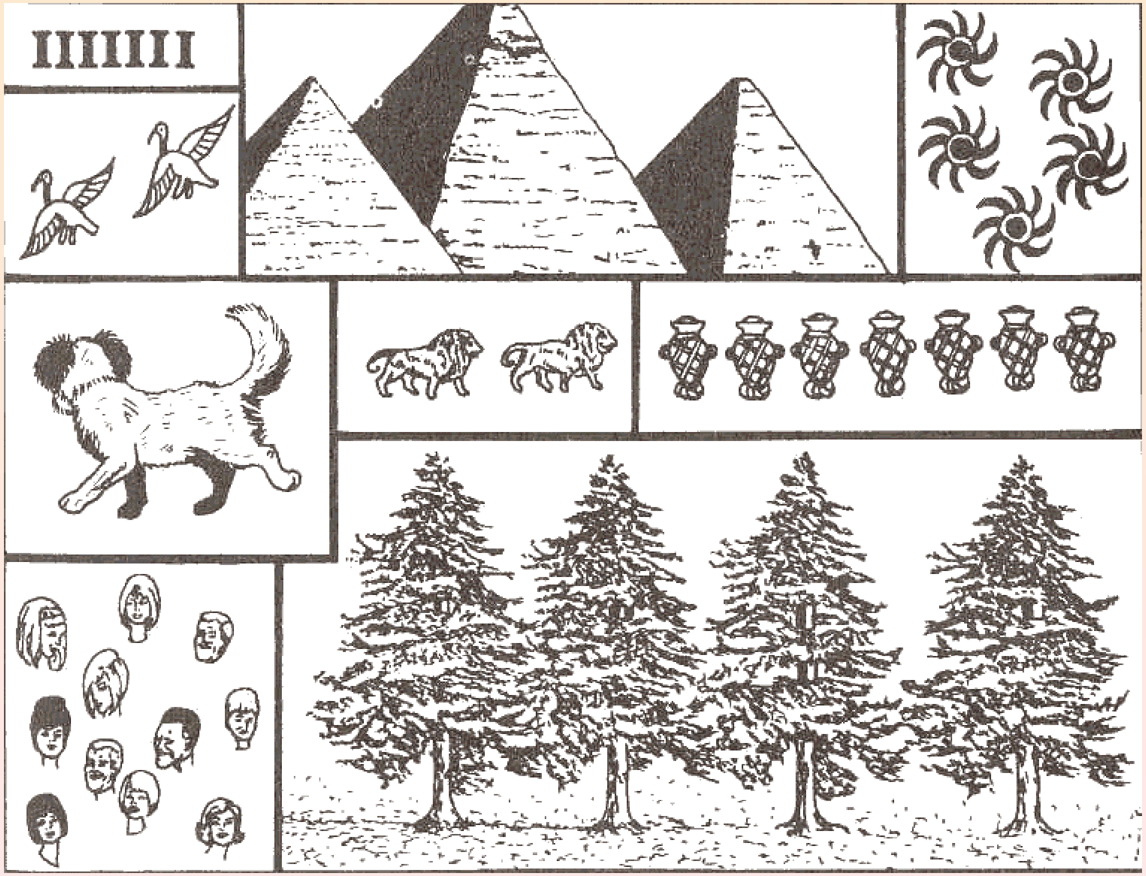
Introduction

A very brief & subjective

History of computation, numbering systems and computing machines

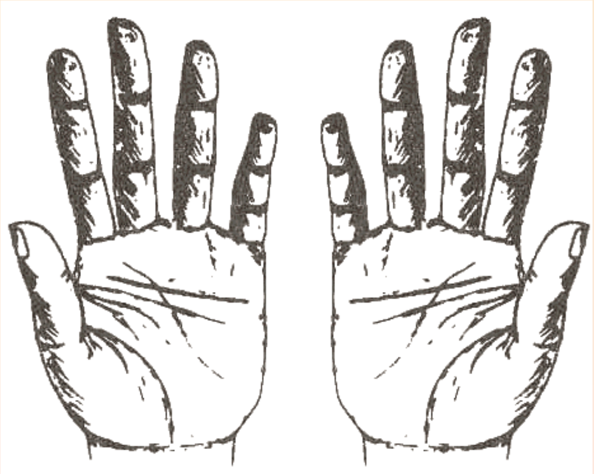
Perception of Quantity

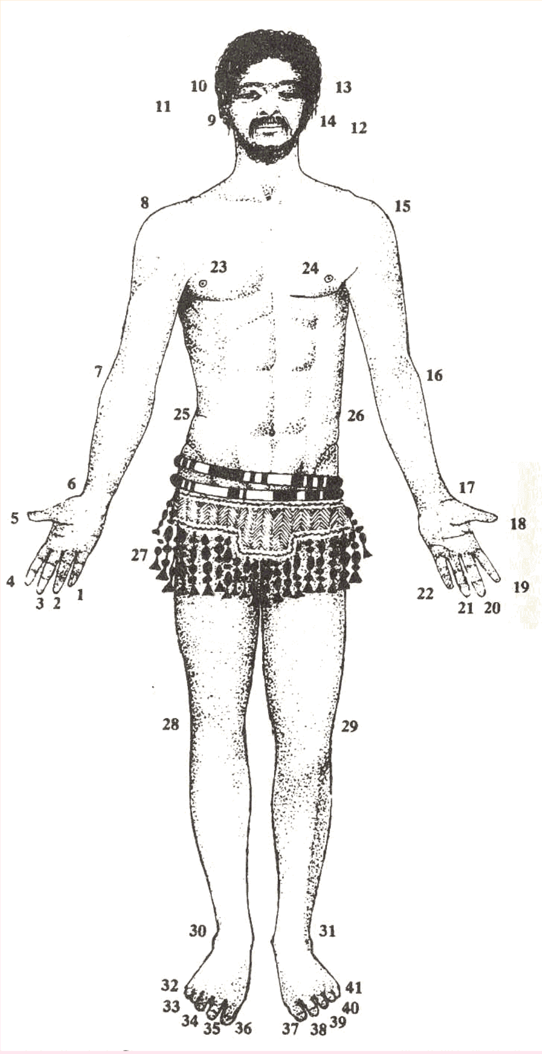


Natural human quantity perception does not exceed 3 or 4

In order to deal with anything more than 4 we need tools: numerals

Counting without numbers



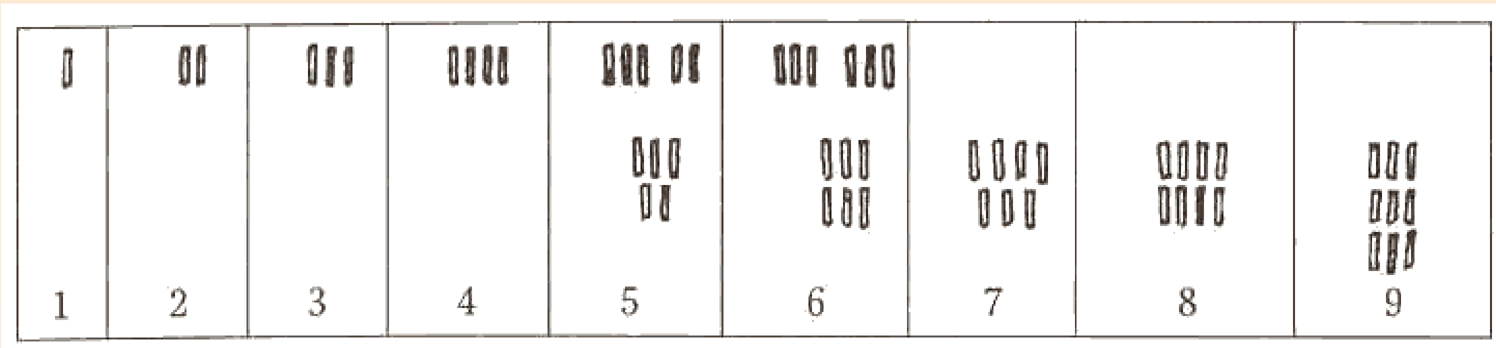


Human body can (and still) provide basic tools for complex counting.

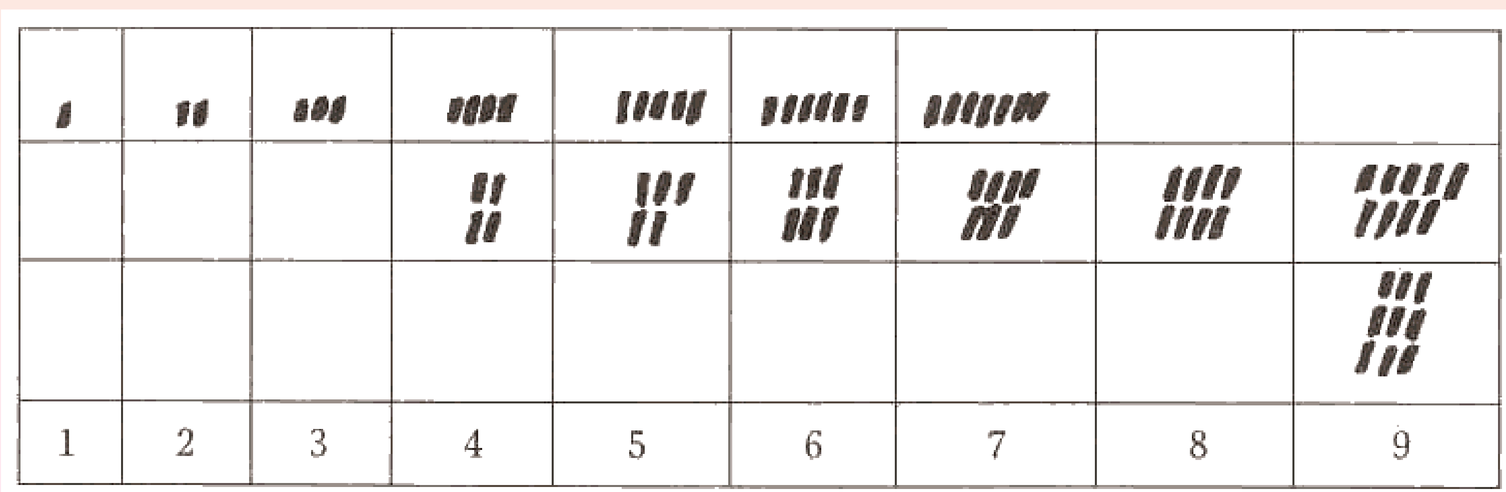
Many names of numerals in modern languages originate from parts of human body.

Development of notation

Egipt 3000-1000 BC

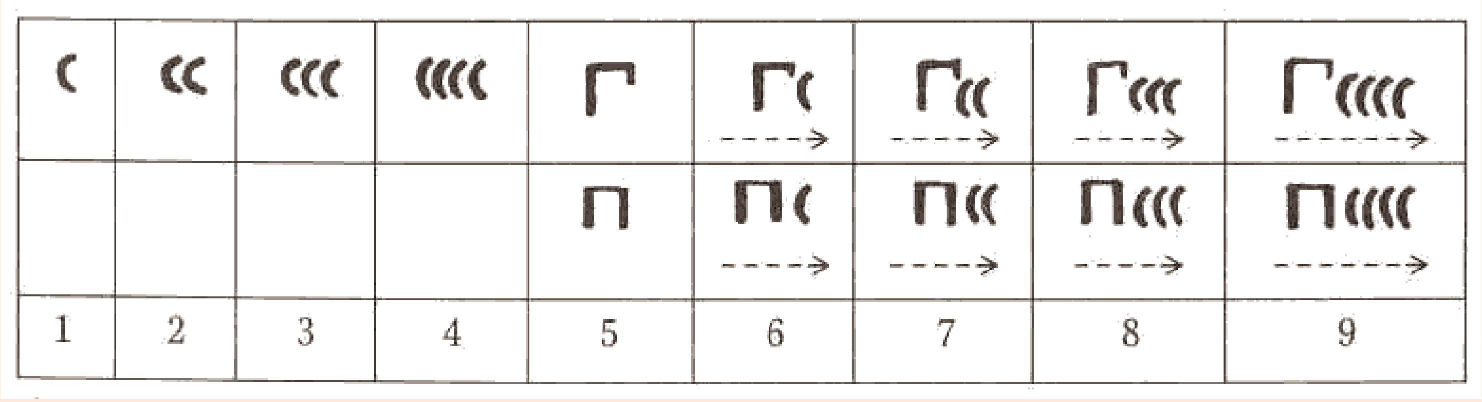


India 2500-1700 BC

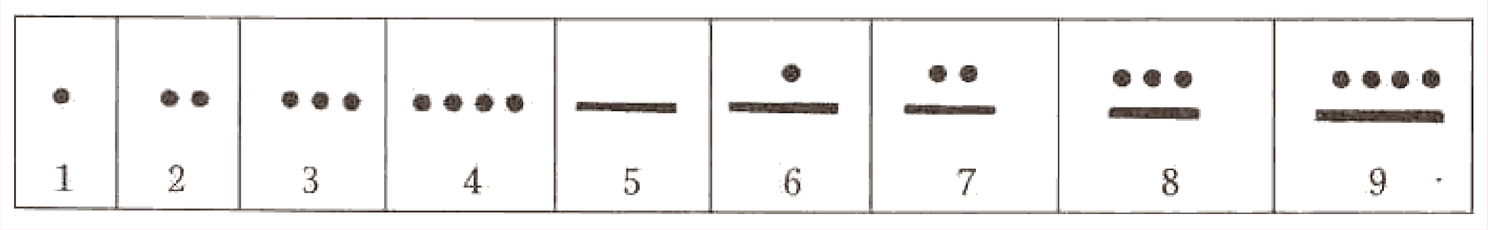


Development of notation

Greece 500-200 BC

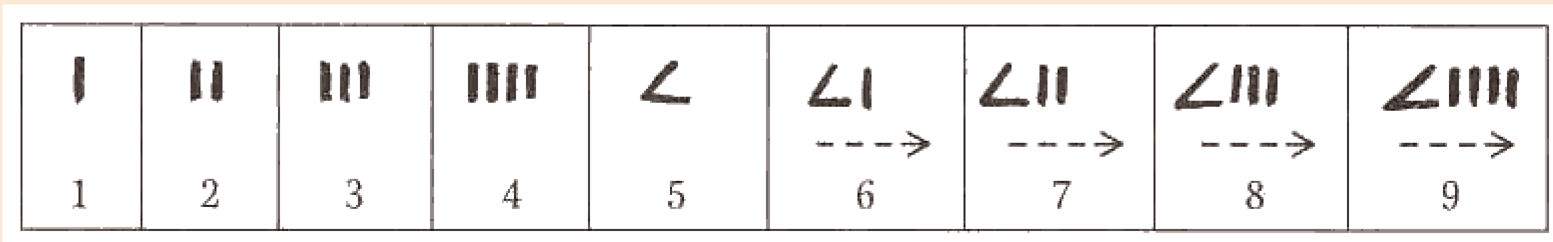


Mayan Civ. 300-1400 AD

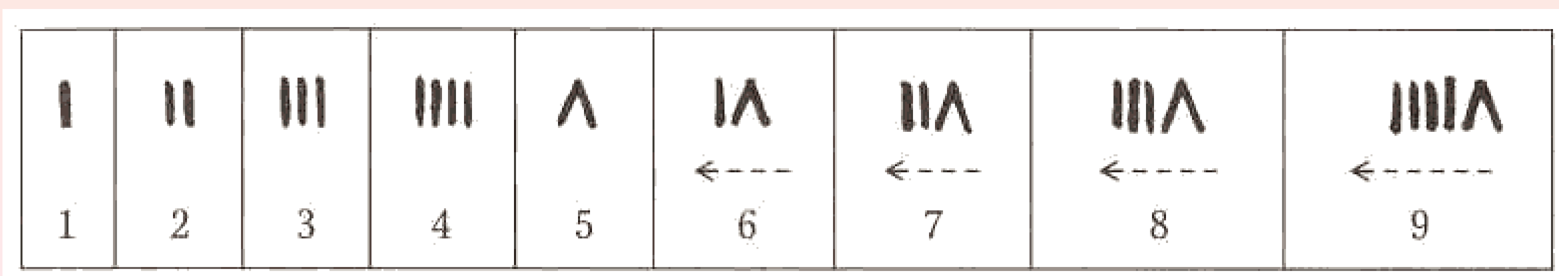


Development of notation

Lycia (Asia Minor) 500-0 p.Chr.



Etruskan Civ. 600-400 p.Chr.



Numbering systems

Additive:

- **simplest and oldest**
- **value is the sum of symbols**
- **easy addition and subtraction**
- **difficult multiplicatin and division**

Positional:

- **currently in use**
- **value is the position-weighted sum**
- **concept of null symbol – zero**
- **easy arithmetics: +,-,*,/ and other**

Additive System

X=VV=IIIIIIIIII, V=IIIII, I=one

Digit (symbol) has constant value

number **X X X I I I**

Digit position 6 5 4 3 2 1

$$X_6 = X_5 = X_4 = \text{ten}$$

Position does not matter

X X X I I I = X+X+X+I+I+I = ten

Value is the sum of digits

(with some minor modifications)

Positional System

Digit position does matter
Invention of Zero (lack of digit at position)

$$3\,0\,4 \neq 4\,0\,3$$

$$3 \neq 3$$

$$341 = 3 \cdot 10^2 + 4 \cdot 10^1 + 1 \cdot 10^0$$

Value is the weighted sum of digits.
The weights are successive powers of the base

Positional Systems in Use

$$a_m\, a_{m-1} \ldots a_2\, a_1\, a_0$$

b – base
m...0 – position

$$a_m \cdot b^m + a_{m-1} \cdot b^{m-1} + \ldots + a_1 \cdot b^1 + a_0 \cdot b^0$$

e.g.

Decimal system: 0 1 2 3 4 5 6 7 8 9
1234 = (→dec) $1*10^3 + 2*10^2 + 3*10^1 + 4*10^0$

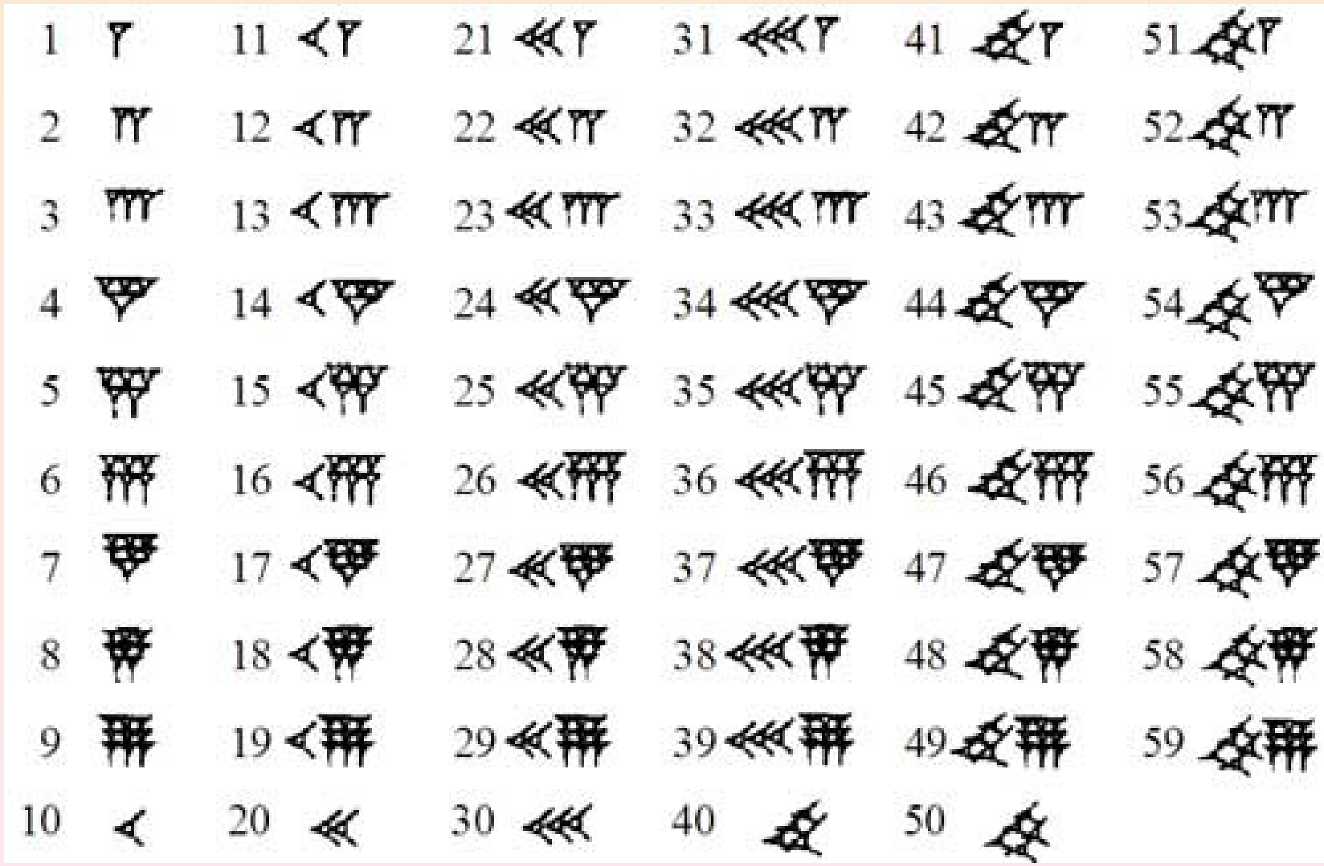
Binary system: 0 1, base=2
1101 = (→dec) $1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$

Octal system: 0 1 2 3 4 5 6 7, base 8
1207 = (→dec) $1*8^3 + 2*8^2 + 0*8^1 + 7*8^0$

Hexadecimal system: 0 1 2 3 4 5 6 7 8 9 A B C D E F, base 16
a2f5 = (→dec) $10*16^3 + 2*16^2 + 15*16^1 + 5*16^0$

base-12 & base-60: Babylonian numerals

1900 BC



Sexagesimal positional system (base-60)

Binary system

The simplest of all: digits 0 i 1

Ideal to make use of any two-state phenomena:

- **sound: short/long: Morse code**
- **color: black/white: stripe codes**
- **voltage: high/low: bits in computer systems**

Two-stated coding system is the fundation of reliability of digital technique
– only two opposite state are allowed,
– no intermediate levels.

Natural Binary Code – NBC

Binary digit (0 or 1) – bit (***bi**nary digi**t***)

$$a_{n-1} \ldots a_1 a_0,$$

$$a_{n-1} \cdot 2^{n-1} + \ldots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

NBC – only non-negative integers

Length of Binary Numbers

Binary numbers can be of arbitrary length, but the dominating practise is to use the following standards:

8-bits –*Byte*, B
16-bits –*Word*, W
32-bits –*Long (Double) Word*, L
64-bits –*Quad Word*, Q

Ranges (for NBC)

Byte:	**0 ... 2^8-1**	**= 0 ... 255**
Word:	**0 ... 2^{16}-1**	**= 0 ... 65535**
L.Word:	**0 ... 2^{32}-1**	**= 0 ... 2147483647**
Q.Word:	**0 ... 2^{64}-1**	

Kilo, Mega, Giga … in Computer Science

The International System of Units (SI):

1 k = 1000, 1 M = 1000 k, 1 G = 1000 M

Bit (b) is a basic unit of information
Byte (1B = 8b) is unit of memory capacity

The bits and bytes are interchangable, but typically:
- memory systems are measured in bytes (B)
- data transfer rates are measured in bits (b/s)

Multiples of bits and bytes are not SI:

kb (SI) → 1000b … but we really mean 1024b
kB (SI) → 1000B … but we really mean 1024B

Kilo, Mega, Giga … in Computer Science

IEC 60027 (2005): Letter symbols to be used in electrical technology

SI Metric Prefix	Symbol	Decimal Multiplication Factor	Exponent
deka	da	10	10^{1}
hecto	h	100	10^{2}
kilo	k	1 000	10^{3}
mega	M	1 000 000	10^{6}
giga	G	1 000 000 000	10^{9}
tera	T	1 000 000 000 000	10^{12}
peta	P	1 000 000 000 000 000	10^{15}
exa	E	1 000 000 000 000 000 000	10^{18}
zeta	Z	1 000 000 000 000 000 000 000	10^{21}
yotta	Y	1 000 000 000 000 000 000 000 000	10^{24}

IEC Binary Prefix	Symbol	Binary Multiplication Factor	Exponent	COMMENTS
kibi	Ki	1 024	2^{10}	Kilobinary
mebi	Mi	1 048 576	2^{20}	Megabinary
gibi	Gi	1 073 741 824	2^{30}	Gigabinary
tebi	Ti	1 099 511 627 776	2^{40}	Terabinary
pebi	Pi	1 125 899 906 842 624	2^{50}	Petabinary
exbi	Ei	1 152 921 504 606 846 976	2^{60}	Exabinary

1kB = 1000B, **1KiB = 1024B**

1MB = 10^6B, **1MiB = 2^{20}B**

Don't be confused when you discover that your 1TB HD has actually 931GiB

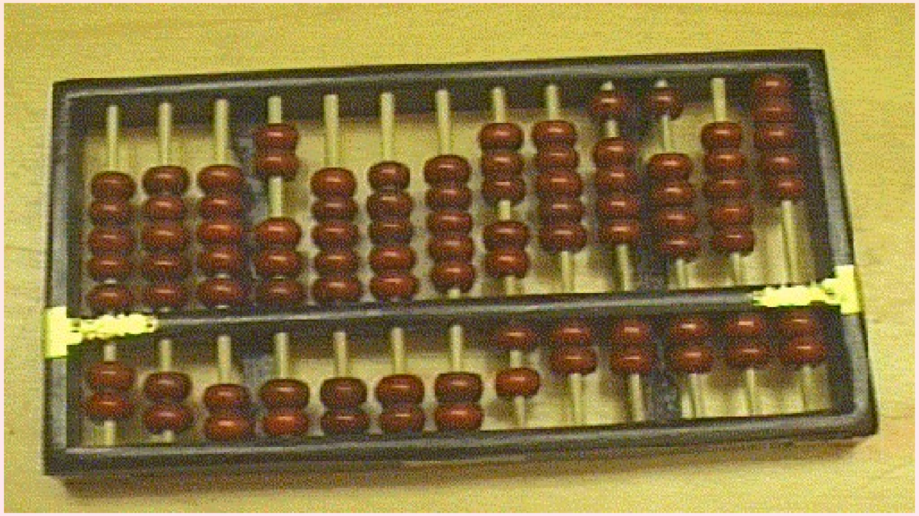
History of computation

Prehistory

- numerals: one, two, ... many
- tools: stones, bones, sticks

3000-500 BC.

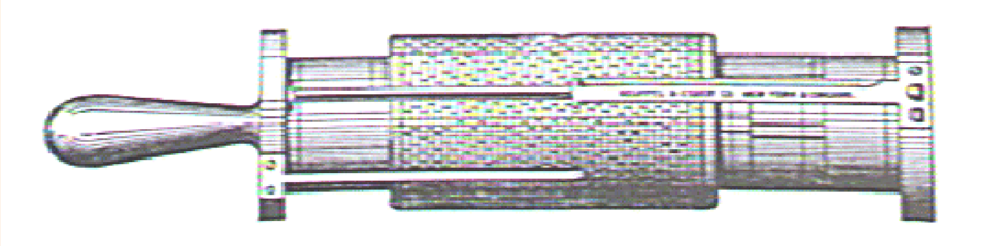
- abacus



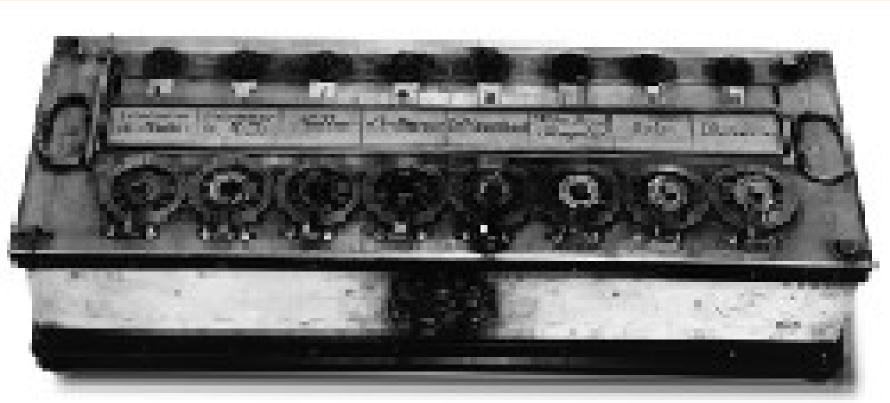
History ...

17th century

1622 - slide rule,
William Oughtred,
England



1642 - adding machine
Blaise Pascal

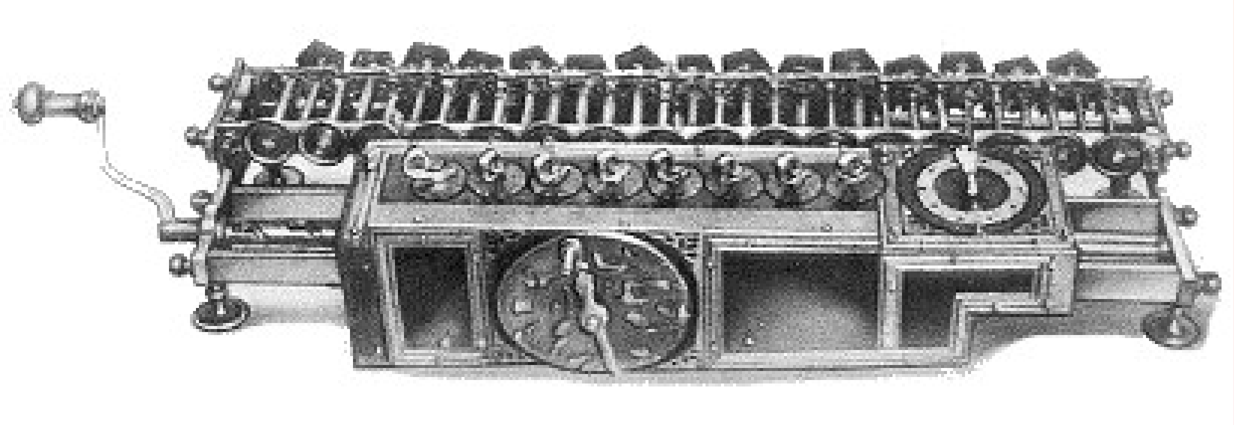


History ...

17th century

Gottfried Leibniz (1646-1716), Germany

1673 - four-operation calculation machine

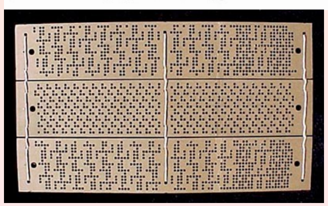


Discovery of binary system (Ying-Yang influence)

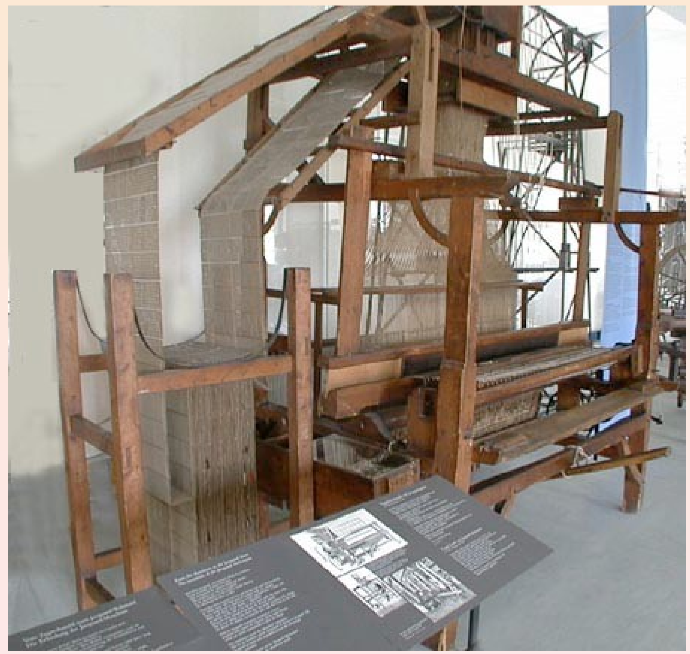
History ...

19th century Jacquard's loom — first programmable machine

[Jacquard-card Making.]



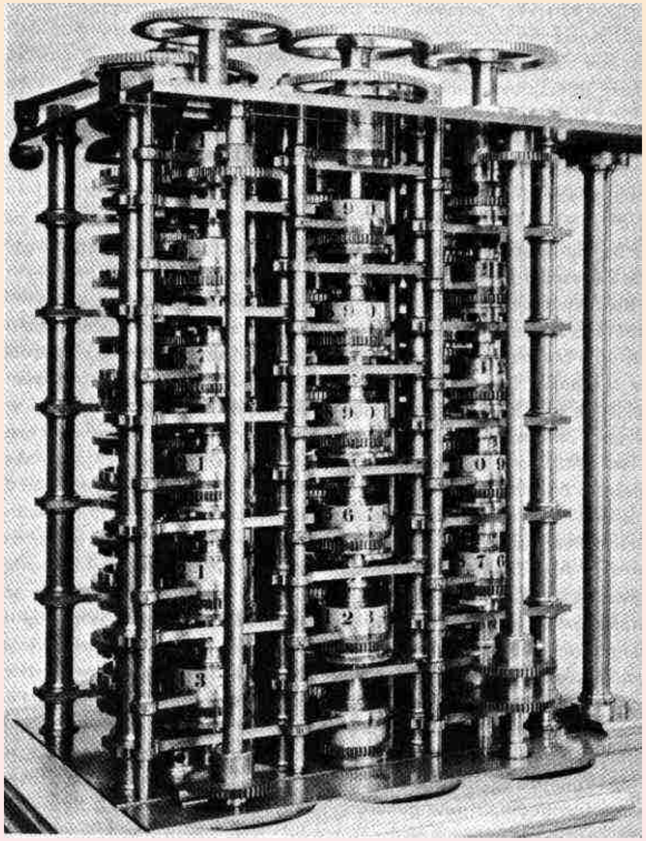
The Program (input)



The Machine (processing)

The Output (Woven Fabrics)

History ...



Charles Babbage machines:

"Difference Engine":
tables of polynomials

"Analytical Engine":
basic idea of a universal computing machine

History ...

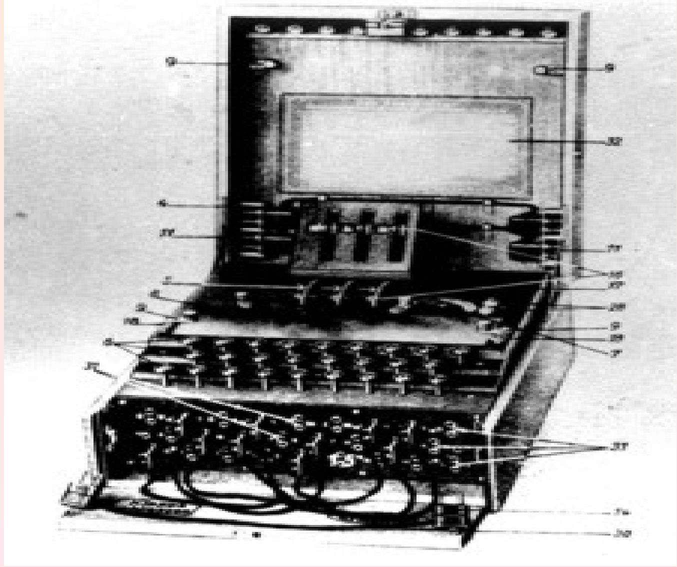
19th century

mechanical arithmometers (number crunchers) using punched cards (perforated cards)

20th century

Complex mathematical problems mainly for military purposes:

- ballistics,
- ciphering (Enigma),
- nuclear weapon research



History ...

Alan Turing

- study on automated mathematical problem solving
- Idea of universal machine executing algorithm based on input data (symbols) and its internal state - Universal Turing Machine (UTM)
- conception of personality as a convoluted algorithm: "Turing Test" starts discussion/research about artificial intelligence.

HUMAN: "I don't have anything against my **father**. He was okay ..."
ELIZA: "Tell me more about your **family**."

John von Neumann

- concept of modern computer that consists of: central processing unit, memory, peripherals.
- Computer processes program stored in memory.

History ...

1944 — Mark I : first electromechanical computer

- length 120m , height 2.5m, weight 5t, wires length 700km
- 760.000 components, mostly electromechanical relays
- use: military ballistic calculations
- output device: electrical typewriter
- speed: multiplication ca. 3s

History ...

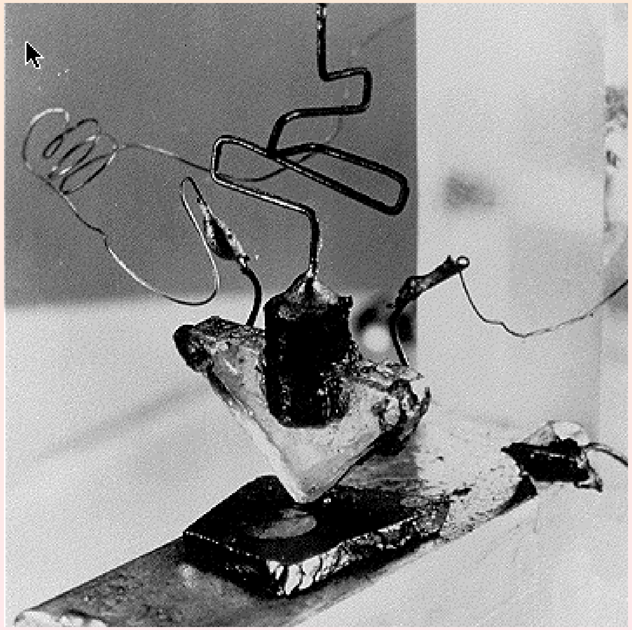
1944 — ENIAC (Electrical Numerical Integrator And Calculator): vacuum tube technology: 1st generation computers (1944-55)

17.468 tubes, 70.000 resistors, 10.000 capacitors,
1.500 relays, 6.000 manual switches and 5 million solder joints
area 167 m^2, weight 30 tons, power 160 kilowatts
(during turning on caused light dimming in nearby Philadelphia)

Within 1s ENIAC (1000 times faster than any computer built till this time) performed: 5,000 summations, 357 multiplications or 38 divisions.
Programming was based on electrical connecting modules that were responsible for certain computations (hardwired programming) and used to take weeks.
Machine needed frequent and wearing maintenance.

History ...

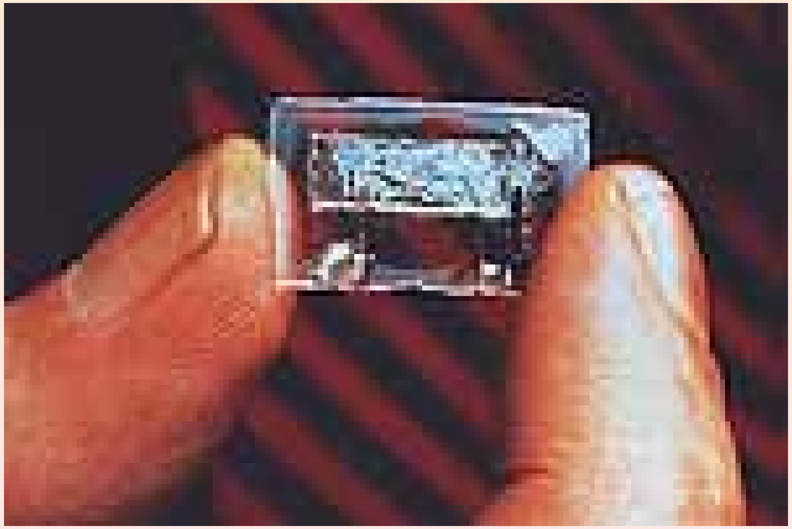
1947 — transistor
semiconductor electronics: 2^{nd} generation computers (1956-1963)

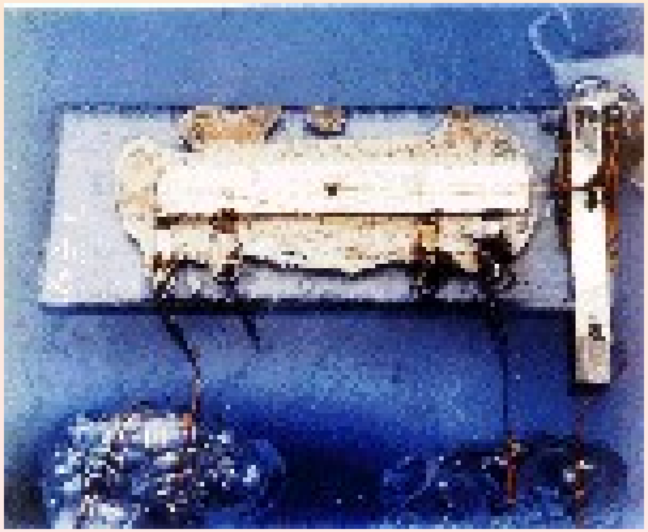


- **low voltage,**
- **low power consumption,**
- **small dimensions,**
- **high reliability.**

History ...

1958 — integrated circuit: 3rd generation computers (1964-1972)

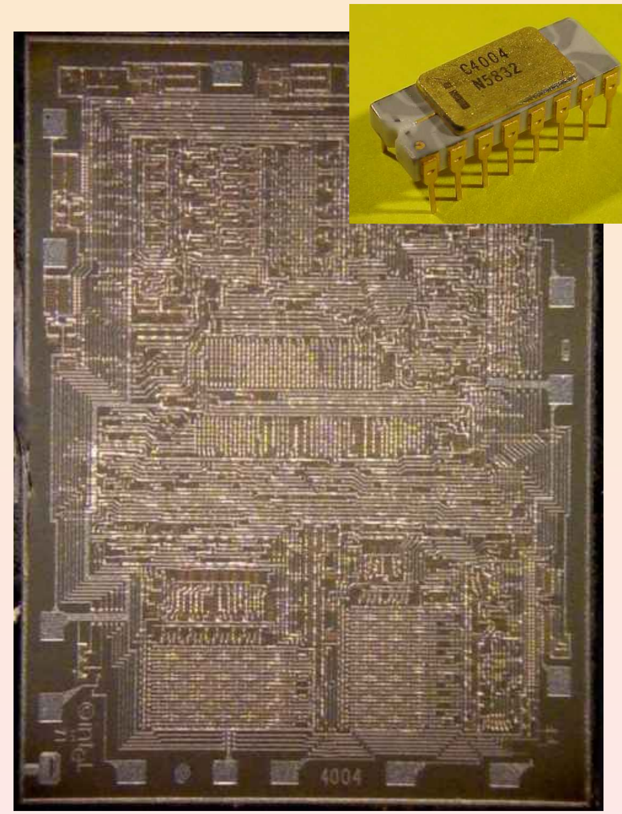




Complicated electronic device build on silicon plate surface (transistors, diodes, resistors and connections)
fabricated in a single process,
high density of components and low power consumption.

History ...

1971 — microprocessor Intel 4004
4th generation computers (1973-1977)



Intel 4004 - dimensions: 3mm x 4mm
consisted of 2300 transistors,
computing power similar to ENIAC
(18.000 tubes).
In 1972r. space probe Pioneer 10
already used Intel 4004

Combine in a single integrated circuit: control unit, general purpose and arithmetic registers.
Next generations of microprocessors mostly differ only by components density (number of integrated elements). General idea remains unchanged.

History ...

Personal computers (PC)

1976 — first microcomputer Apple I

Steven Wozniak,
Steve Jobs

Microprocessor:	MOS 6502, 1 MHz
RAM memory:	8 KB
ROM memory:	256 bytes

History ...

1977 - Apple II

Microprocessor: MOS 6502, 1 MHz
RAM memory: 4 KB (do 64KB)
ROM memory: 12 kB
Built-in BASIC interpreter
Produced in different versions till 1993.

1980 - Apple III

Microprocessor: MOS 6502A, 2 MHz
RAM memory: 128 KB (up to 256KB)
ROM memory: 16 kB

History ...

1981 — Personal Computer (PC) - IBM PC

- First 16-bit microprocessor Intel 8088, clock 4.77 MHz.
- 16kB memory, expandable to 256kB,
- floppy disk drive, capacity 160kB,
- color monitor,
- price 1565$ (equivalent to present 4000$),
- BASIC interpreter built-in ROM memory
- DOS operating system (a little later)

"Open architecture" concept — the reason of great popularity until today

History ...

1983 - IBM PC/XT (5160)

Microprocessor:	Intel 8088, 4.77MHz
RAM memory:	64-640kB (depending on model)
Video card:	CGA (320x200 / 640x200)
Operating system:	MS-DOS 2.0

FDD 5.25”, HDD 10,20MB

1983 - IBM PC Junior

Microprocessor:	Intel 8088, 4.77MHz
RAM memory:	64 KB (do 640kB)
Video card:	CGA (320x200 / 640x200)
Operating system:	MS-DOS 2.0

Mass storage.: tape, option: FDD 5.25”

1984 - IBM PC/AT

Microprocessor:	Intel 80286, 6MHz
RAM memory:	512 KB (do 3MB)
Video card:	EGA (640x350)
Operating system:	MS-DOS 3.0

HDD,FDD

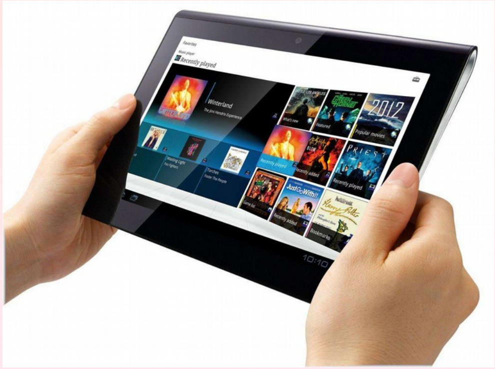
1984 - IBM PC Portable (5155)

Microprocessor:	Intel 8088, 4.77MHz
RAM memory:	256 KB (do 640kB)
Video card:	CGA (320x200 / 640x200)
Operating system:	PC-DOS 2.1

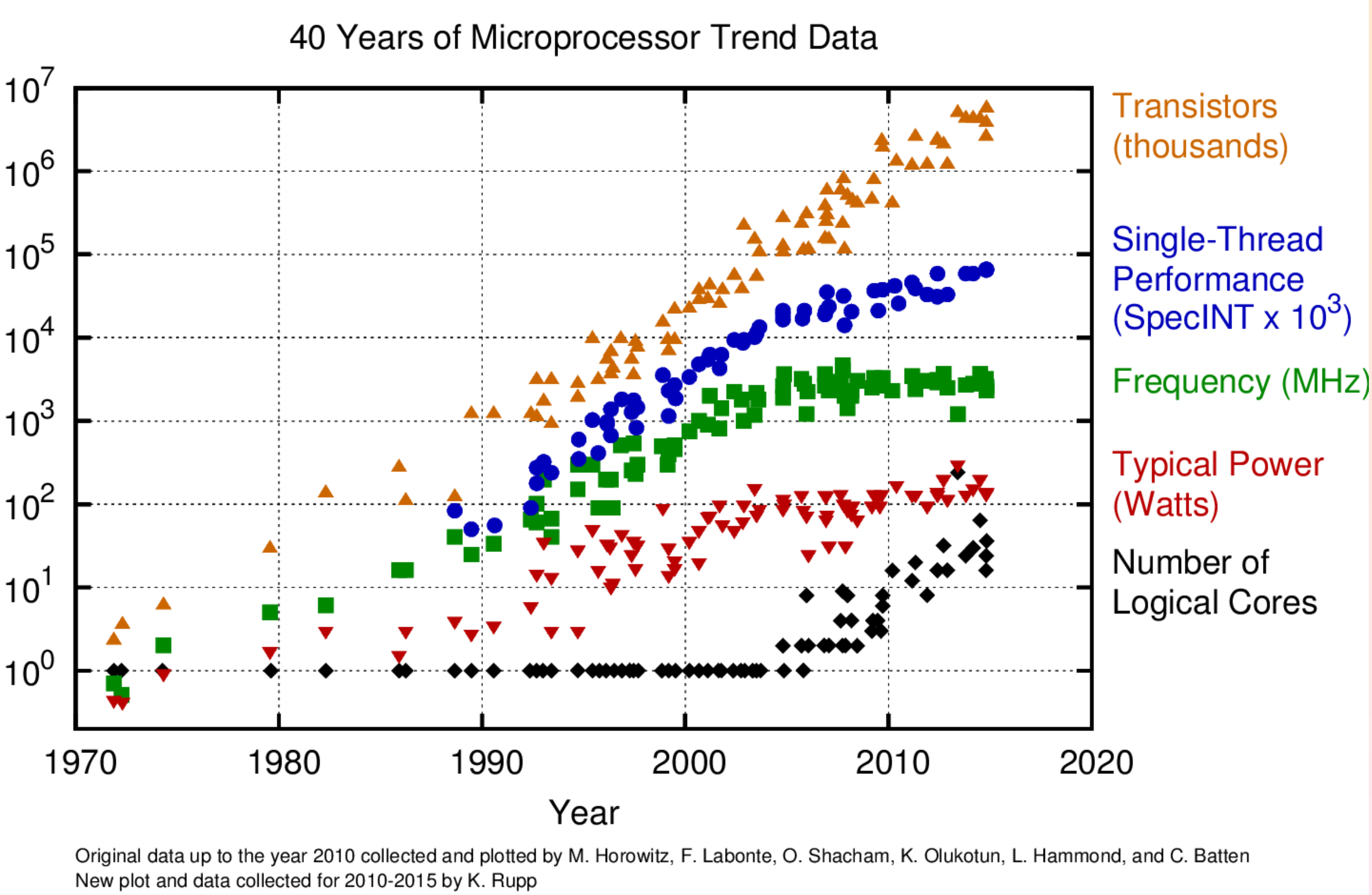
1x lub 2x FDD 5.25”

PC today ?

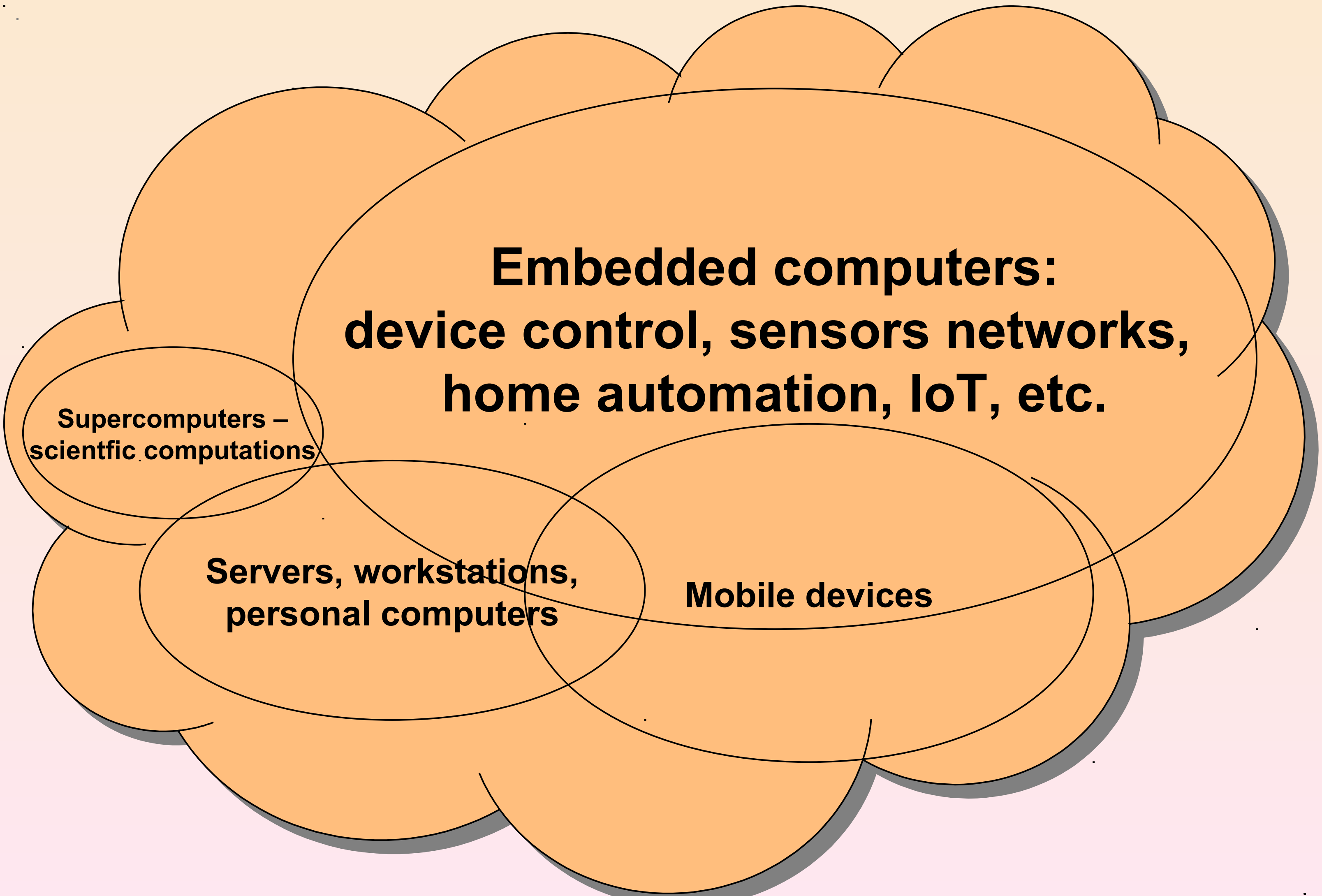
Desktop
Laptop
Tablet/PDA
Smartphone
...



Moore's Law

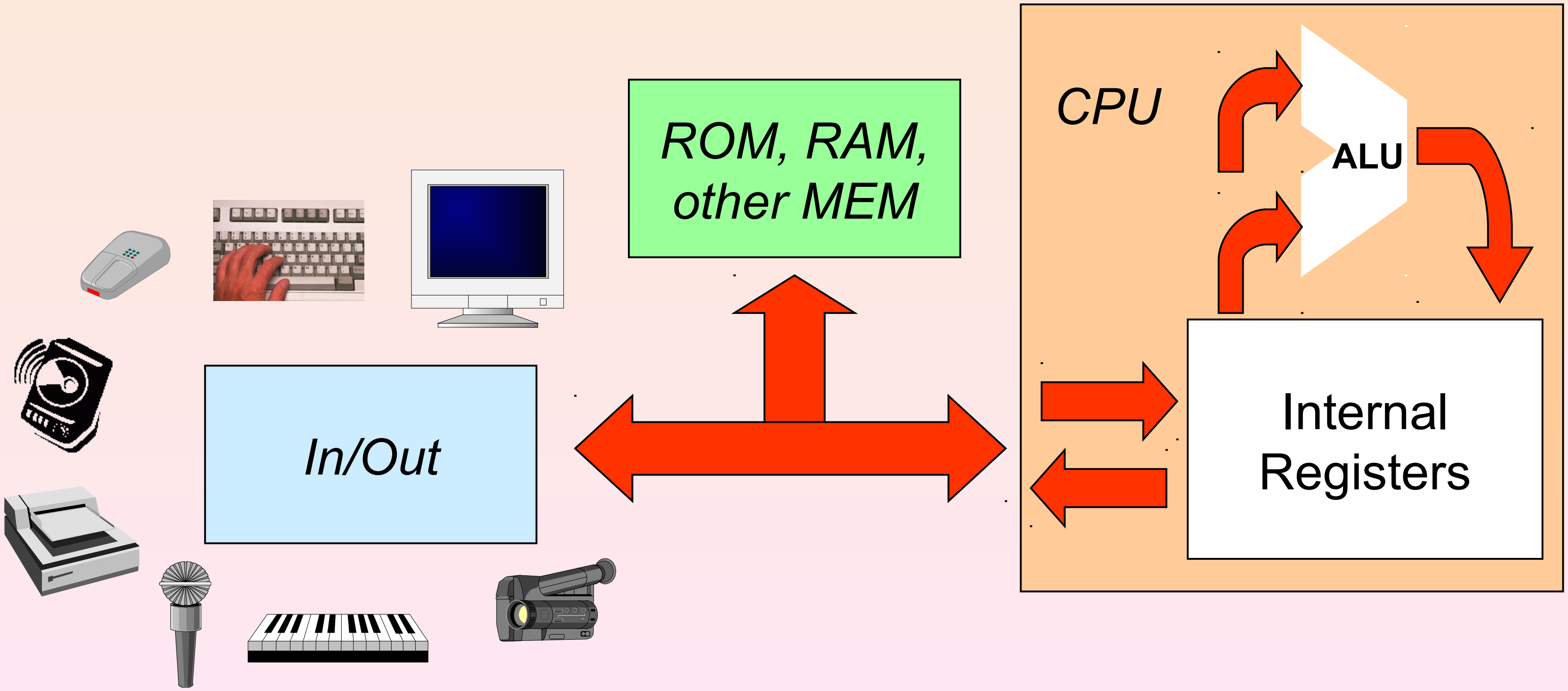

40 Years of Microprocessor Trend Data
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1970
1980
1990
2000
2010
2020
Year
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Fields of computer applications



Computer Architecture

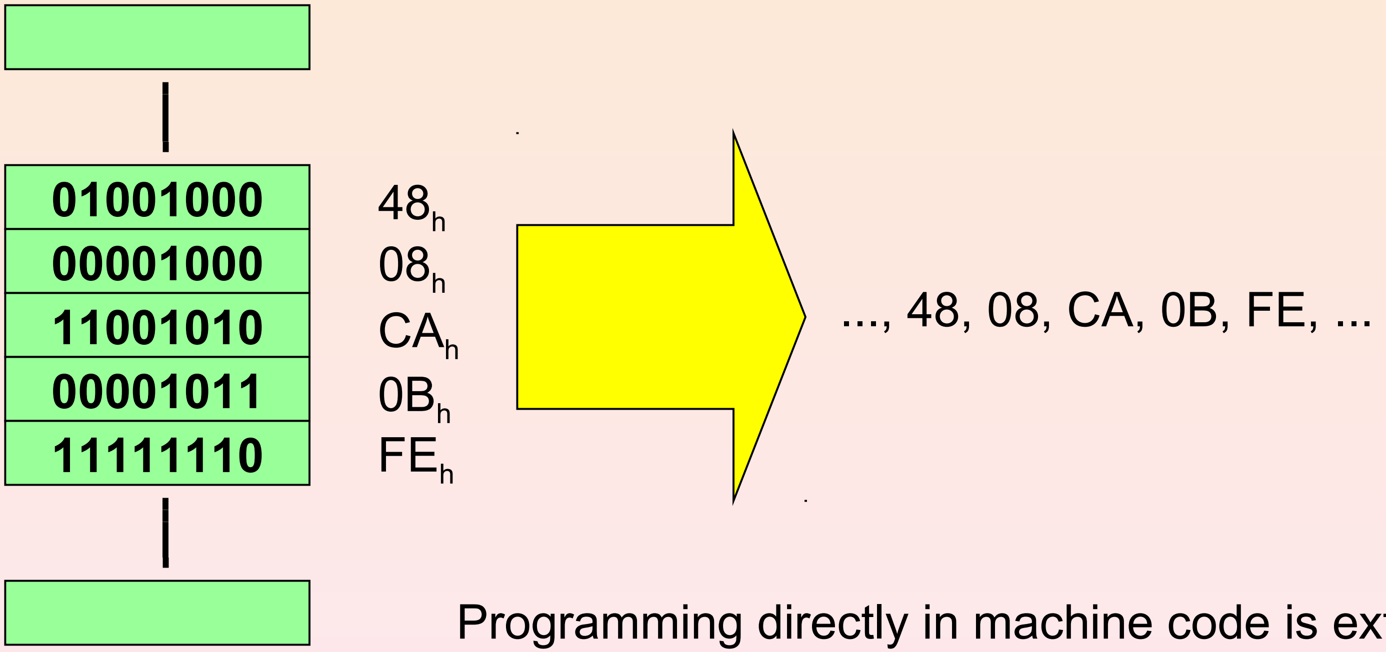
- Computer is composed of processing unit (CPU), memory and I/O
- Computer is executing commands stored in the memory.
- Commands controll very simple operations with ALU and internal registers and memory transfers.
- Human-computer interaction is carried out with Input/Output (I/O) peripheral devices.



Machine code

- Computer instructions (program code) in memory are stored as binary numbers
- Machine code is a sequence of binary numbers, that can be directly "understood" by the CPU

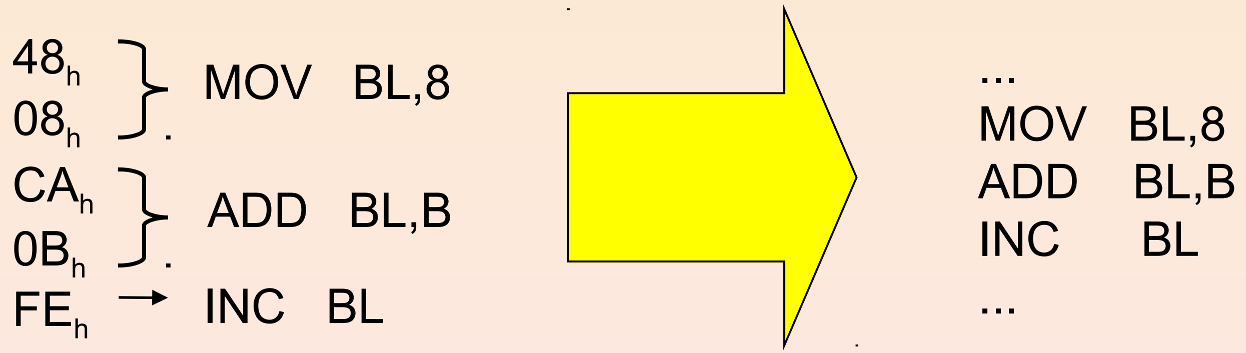
Memory space
ROM + RAM



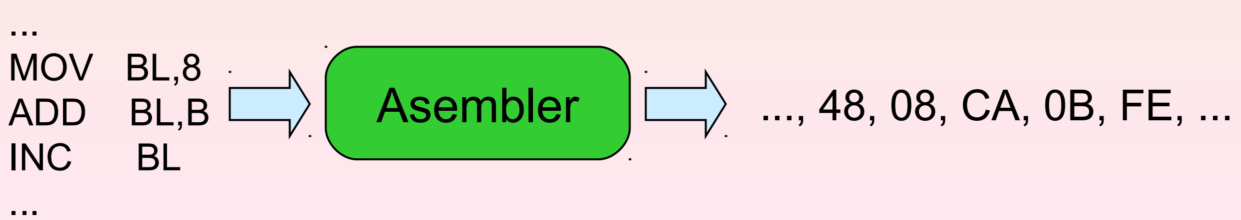
Programming directly in machine code is extremely difficult, but it was a necessity for I and II generation of computers.

Assembly language

- Machine command can be assigned an easy to memorize "mnemonic"
- Programs with mnemonics is assembly language (or assembler)



- Program in assembly language cannot be executed directly by CPU and requires a translation program called "assembler".
- Due to the direct connection between machine command and mnemonics,
- assemblers are relatively simple programs.



High-level languages

- Data abstraction - operation on variable without direct operations on internal registers or memory locations.
- Complex data types - tables, stacks, queues, trees, sets, etc
- Advanced and intuitive control instructions: IF-THEN-ELSE, etc.
- Freedom in program composition - according to human preferences
- Possibility to manage the complex algorithms

- High-level programs need very complex translators - compilers.
- Process of compilation is takes time
- Generated machine code is NOT guaranteed to be optimal (size, speed)
- If something goes wrong, tracing back the cause is extremely difficult

Evolution of user interface

	Users	*Application*	*Type of user interface*
1945-1955	**experts**	**calculator**	**none**
1955-1965	**scientists**	**computations**	**textual – simple commands**
1965-1985	**trained personnel**	**data processing**	**textual – complex command, semi-graphics**
1985-1995	**employes, hobbyists**	**work and leisure**	**graphical – simple menus and forms**
1995-2005	**many people**	**work, communication, leisure**	**multimedia, contextual, task oriented**
2005-2020(?)	**most of people**	**personal assistant**	**intuitive, elements of AR and VR**
2020-	**everyone (?)**	**can't live without**	**bio-iterfaces (?)**

Future of human-computer interaction

Dealing with local computer data (TB) and cloud (PB) resources of computer networks is getting beyond human perception.

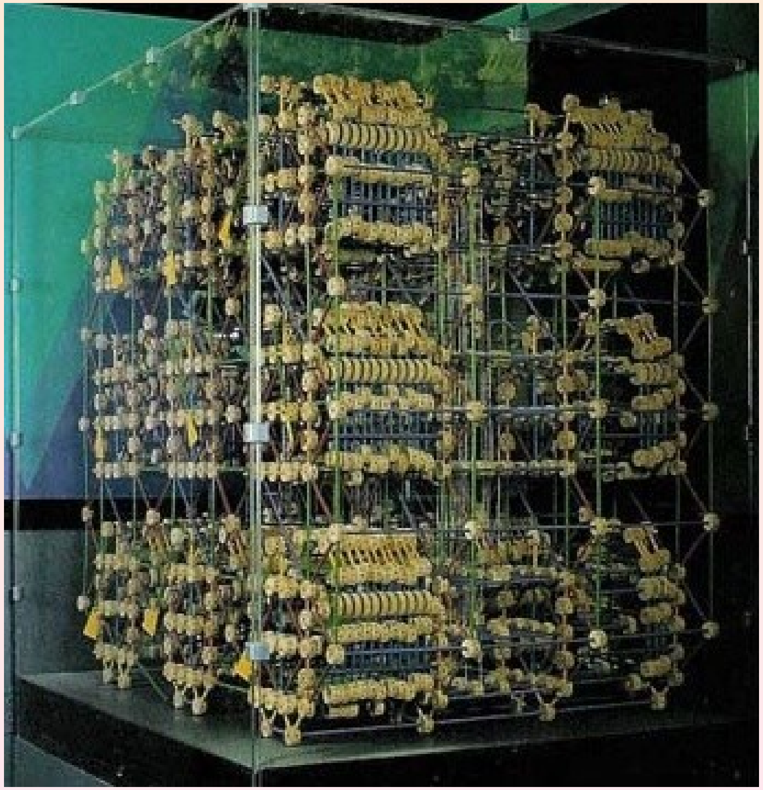
- **Interpretation/guessing the user needs**
- **Informal/casual syntax of command**
- **Very high abstraction of system tasks**
- **Programming by demonstration**
- **Divers and disperse I/O's**
- **User bio-interfaces**

It is more that just a machine ...

- **Physical computer:**
  processor, memory, main board, extension cards, monitor, etc...
- **Logical computer:**
  programs, documents, catalogs, data files, etc...
- **Virtual computer:**
  user interface, *look&feel*, WWW, cloud services

... but it is still the same simple concept

The Tinkertoy Computer



A computer can be built using perhaps any physical phenomena:
- **electrical**
- **optical**
- **mechanical**
- **(bio)chemical**
- **pneumatic**
- **...**