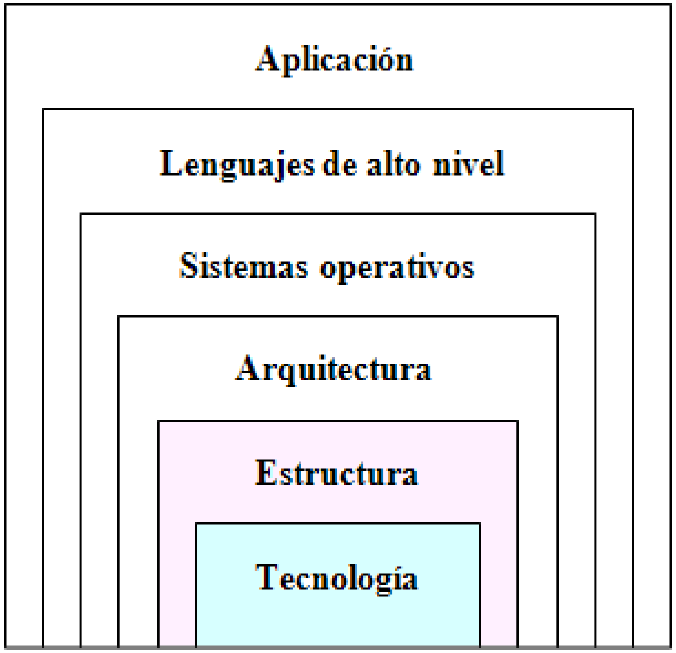
Unit 1:
Introduction to Computers

- Abstraction levels in a computer
- Basic concepts
- Historical Evolution
- Von Neumann Architecture
- Instruction execution cycles
- Programming languages

Abstraction levels in a computer

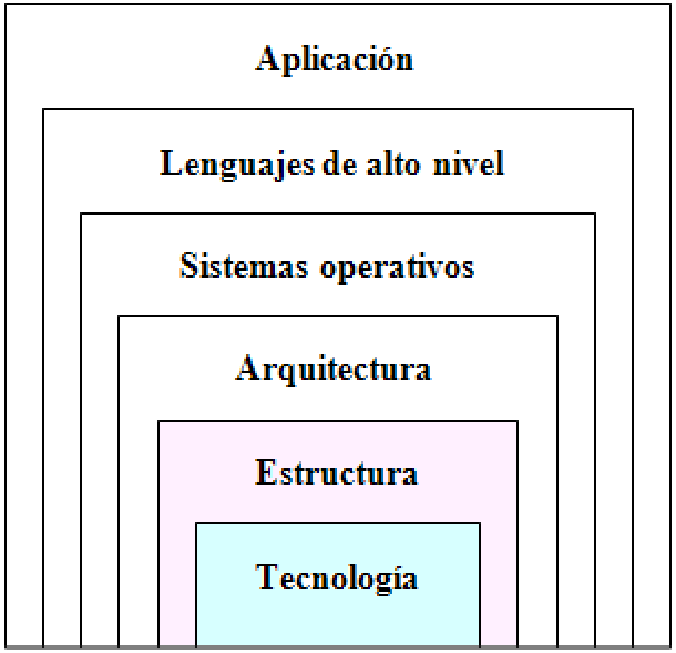
- Grado en Ingeniería Informática



Topic	Subject	Term
Estructura y tecnología de computadores	Fundamentos de tecnología de computadores	1º
	Estructura y organización de computadores	3º
Sistemas Operativos	Sistemas Operativos	2º
	Sistemas Operativos Avanzados	3º
Programación	Fundamentos de programación	1º
	Programación	1º
	Programación Avanzada	4º
	Ampliación de Programación Avanzada	6º
	Procesadores del Lenguaje	5º
Bases de Datos	Bases de Datos	4º
	Bases de Datos Avanzadas	5º

Abstraction levels in a computer

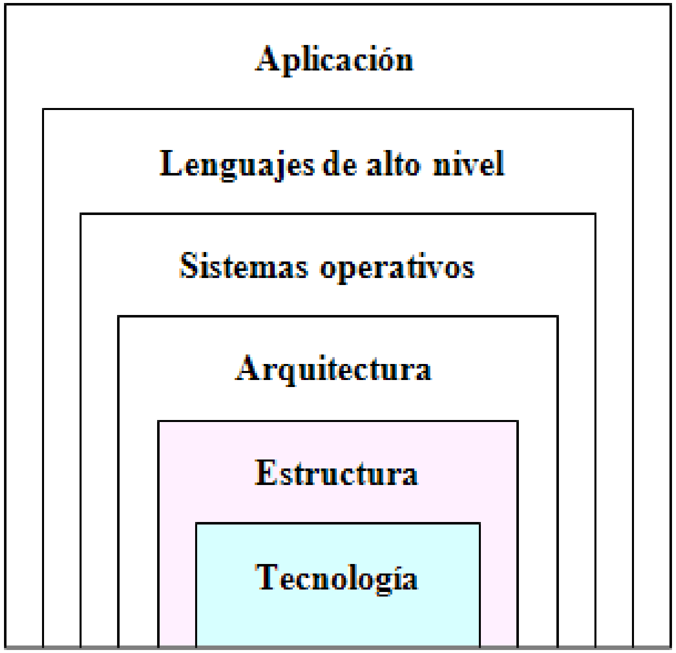
- Grado en Sistemas de la Información



Topic	Subject	Term
Estructura y tecnología de computadores	Fundamentos de tecnología de computadores	1º
Sistemas Operativos	Sistemas Operativos	2º
Programación	Programación y estructuras de datos	1º,2º,3º
Bases de Datos	Bases de Datos	4º,5º

Abstraction levels in a computer

- Grado en Ingeniería de Computadores



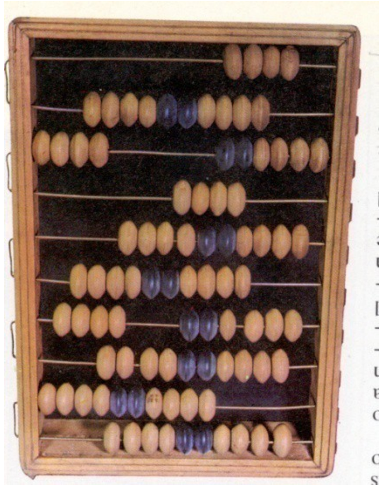
Topic	Subject	Term
Estructura y tecnología de computadores	Fundamentos de Tecnología de Computadores	1º
	Estructura y Organización de Computadores	3º
	Electrónica	5º
Sistemas Operativos	Sistemas Operativos	2º
	Sistemas Operativos Avanzados	3º
Programación	Fundamentos de Programación	1º
	Programación Avanzada	4º
	Procesadores del Lenguaje	6º
Bases de Datos	Bases de Datos	4º
Arquitectura de Computadores	Arquitectura e Ingeniería de Computadores	5º

Basic concepts

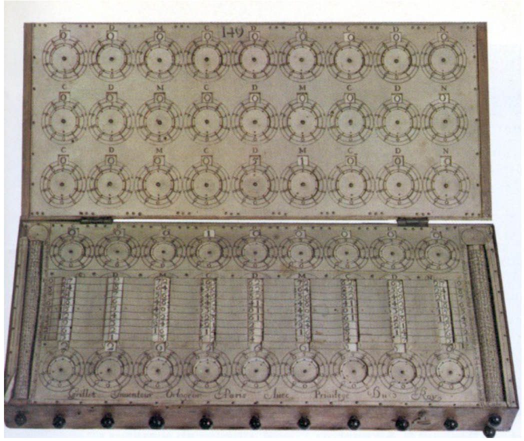
- **Computer**: machine that processes information
- **Information** in a computer:
  - **Bit** ⇨ Binary Digit: Basic information element (‘0’ or ‘1’)
  - **Byte** ⇨ Group of 8 bits (‘01101111’)
- **Word** ⇨ Group of bits with which the computer works usually (8 bits, 16 bits, 32 bits or 64 bits)
  - **Units:** 1 K ⇨ $2^{10} = 1024$
    1 M ⇨ $2^{10} \cdot 2^{10} = 1024$ K
    1 G ⇨ $2^{10} \cdot (2^{10} \cdot 2^{10}) = 1024$ M
- **Instruction**: Operation executed by the computer
- **Data**: Operand or result
- **Program**: Ordered instruction set tha performs a task

Historical Evolution of Computers (I) Precedents

- Abacus: the first known invention to compute:
  - Origin: centuries III- IV BC.
- Pascal arithmetical machine
  - Developed by Blas Pascal (1642)
  - Set of gear wheels numbered from 0 to 9. When passing from 9 to 0, one wheel pushed the next wheel one decimal.
  - A memory system was included to store results



Abacus



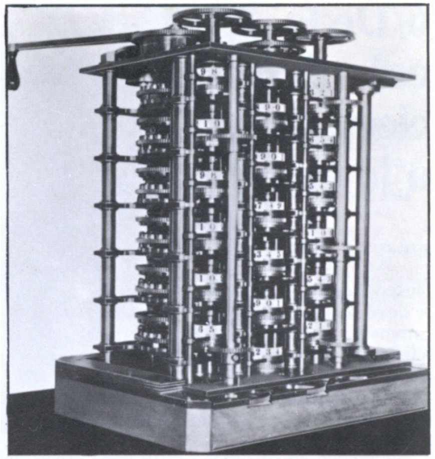
Arithmetical machine

Historical Evolution of Computers (II) Precedents

- Leibnitz machine (1671)
  - It performed four arithmetical operations.

- Charles Babbage: Difference engine (1823) and analytical engine (1831):
  - Executes any operation without human intervention
  - It had a memory, arithmetic unit, gear system to transfer data, an an input/output device
  - It used punched cards to be programmed
  - It was never builted

Difference engine



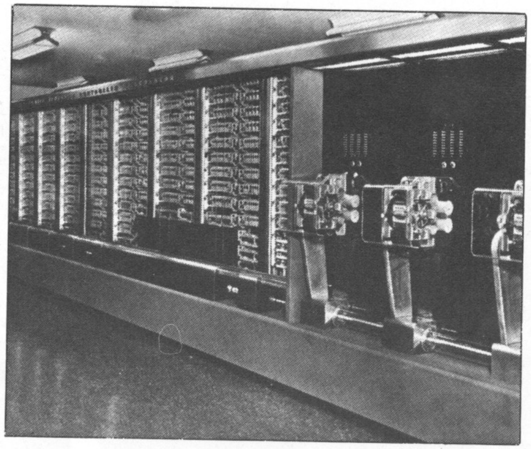
Historical Evolution of Computers (III)
1ª generation

- Experimental machines built with vacuum tubes
- Relay calculators: H. Aiken built MARK series
- 1941: ENIAC - Electronic Numerical Integrator and Calculator. Eckert y Mauchly

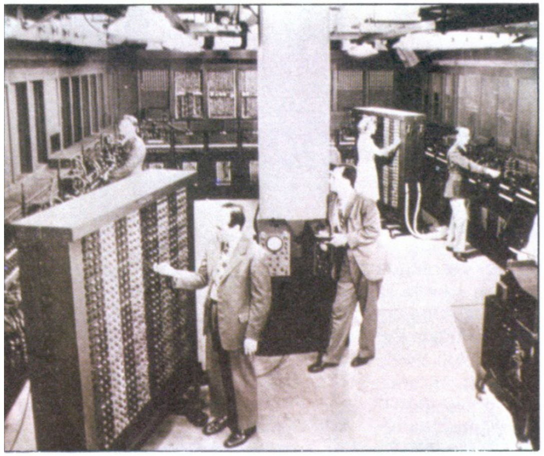
  General purpose computer with wired program

- 1945: *First Draft of Report on the* EDVAC - *Electronic Discrete Variable Automatic Computer.* Von Neumann.

  General purpose computer with **stored** program



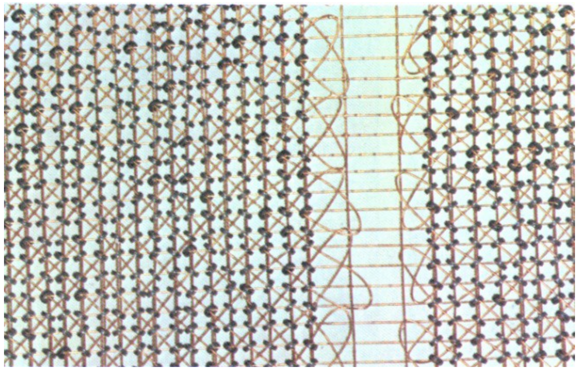
MARK I

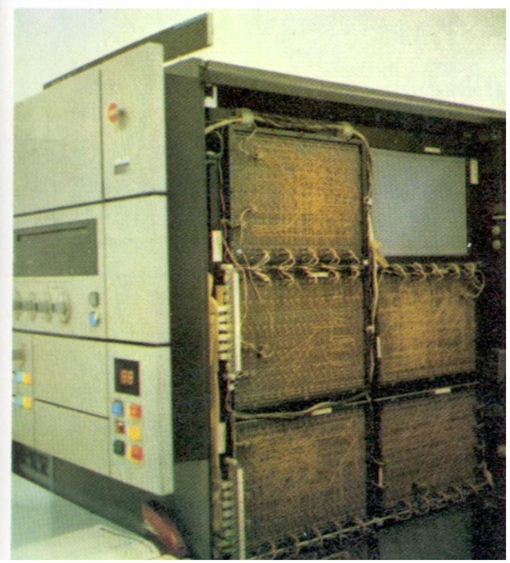


ENIAC

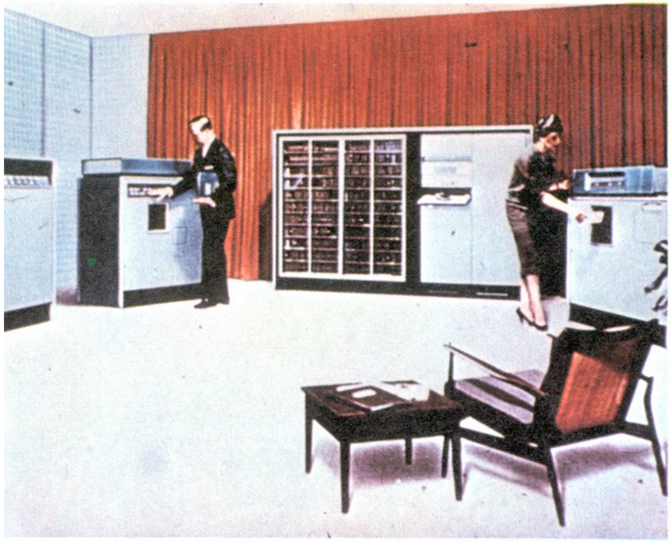
Historical Evolution of Computers (IV)
2nd generation

- Comercial computers
- Built with transistors ⇨ smaller, less heat dissipation, greater reliability
- ferrite memories





Ferrite memories

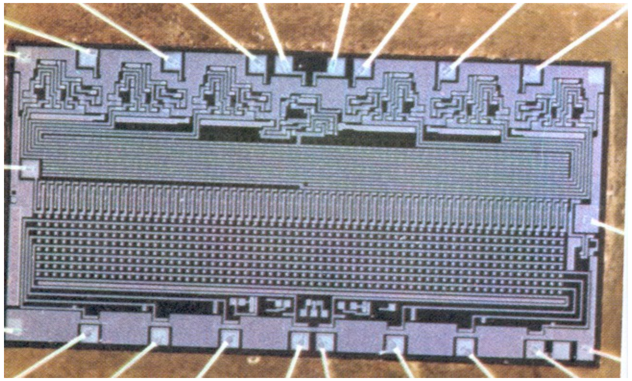


UNIVAC (2ª gen.)

Historical Evolution of Computers (V)
3rd generation

Families of computers: Minicomputers and supercomputers

Built with integrated circuits ⇨ smaller, cheaper, less heat dissipation



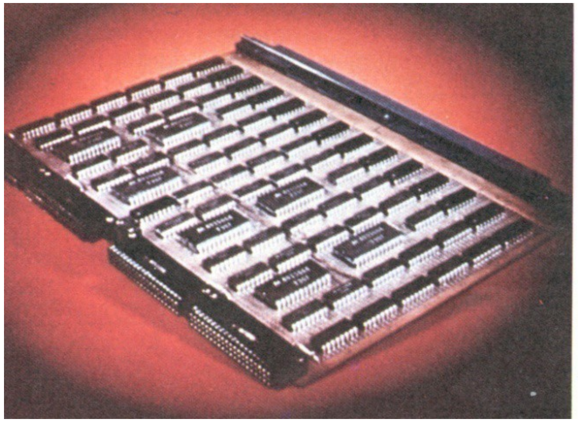
Integrated circuit

IBM serie 370 (3ª gen.)

Historical Evolution of Computers (VI)
4th generation

- Personal Computers (PC) and work stations.
- Other applications: household appliances, music and video..., etc.
- Built with microprocessors and semicondutor memories.
  1971: 1er microprocesador, INTEL 4004
- 80's ➪ information processing
- 90's-00's ➪ Communications (nets)

PC (4th gen.)



Semiconductor memory

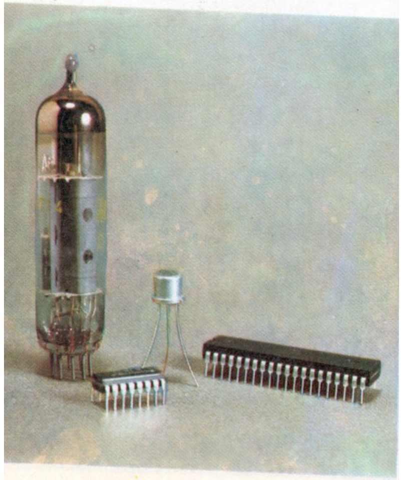
Historical Evolution of Computers (VII)
5th generation

- Microprocesador as basic element
- Massive parallelism
- Cummunication and connections among computers.
- Internet, WWW, email...

- **6th Generation**?
  - Miniaturization
  - Parallelism
  - Clusters
  - Smart phones

Historical Evolution of Computers (VIII)
Summary

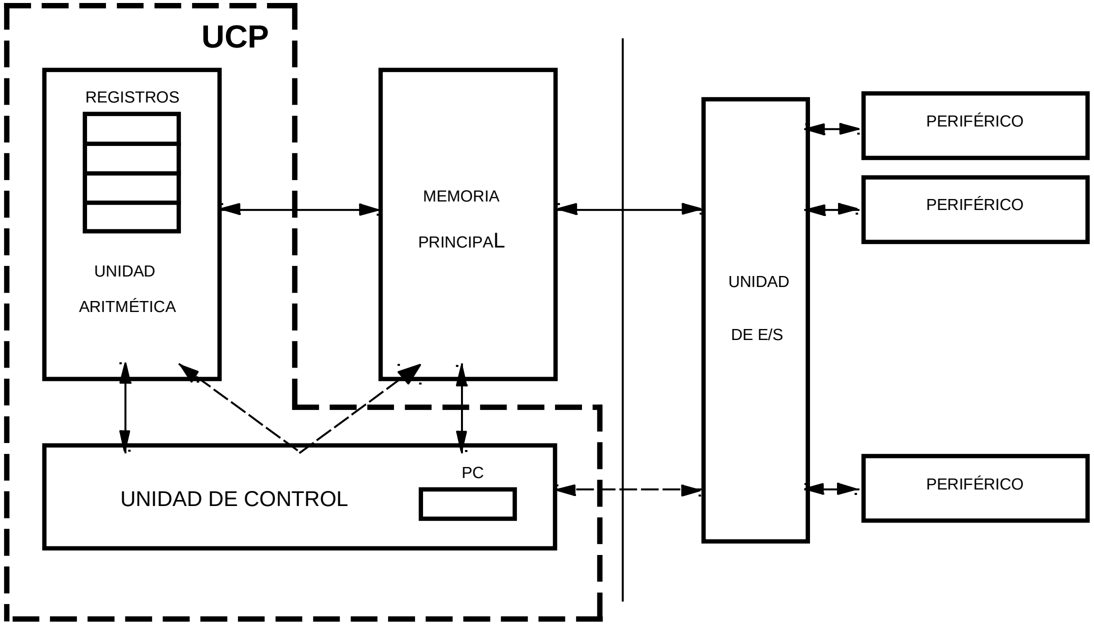
Generation Characteristics	1st	2nd	3rd	4th	5th
Decades	1950 - 1960	1960 - 1970	1970 - 1980	1980 - 1990	1990 -200?
Technology	Vacuum tubes	Transistors	I.C. (SSI-MMI)	I.C (LSI)	I.C. (VLSI)
Machines	IBM 701	CDC 6600	PDP-8, PDP-11	Fujitsu M382 Cray X-MP	Alpha 21164 Pentium
Memory	Williams tubes magnetic tapes	ferrites	Integrated circuits, caches	Virtual memories	Multiple level cache
Languages	Machine	FORTRAM, COBOL, ALGOL, PL1	BASIC, PASCAL	High leevl	C, C++, Java
Product	Computer	comercial computer	Minicomputer	Microcomputer	Multiprocessor



Vacuum tube, transistor, integrated circuit

Von Neumann Architecture

- Developed 1945 by John von Neumann
- Executes machine instructions from a program stored in the memory



- Blocks:
  - Memory
  - Arithmetic-Logic Unit and registers
  - Control Unit
  - Input/Output unit

- **Buses** connect different parts:

data bus, address bus, control bus

Instruction execution cycles

1. **Fetch**
   CU generates signals to read a instruction from memory in the direction pointed by the program counter PC
2. **Decode:**
   CU receives instruction at IR and decodes it
3. **Fetch operands:**
   CU reads, if neccessary, operands from memory or registers
4. **Executions and store results**
   CU generates signals to execute instruction, and stores result in memory or registers
5. **Program Counter update.** CU updates the program counter to point to the next instruction to be executed.
   - sequential working
   - sequence modification ➪ PC modification ➪ bifurcation or jump

Programming languages (I)

High level language:
-Set of instructions and syntaxis (PASCAL, C)
-portable (same code compiles in different machines)

Low level language

- **Machine language**:
  Instructions are written in binary
  - Difficult and a lot of mistakes ➪ Solution: high level language and compile

- **Assembly language**:
  Instructions are represented with symbolic names or mnemonics
  - Each instruction correspond with a machine instruction

Programming languages (II)

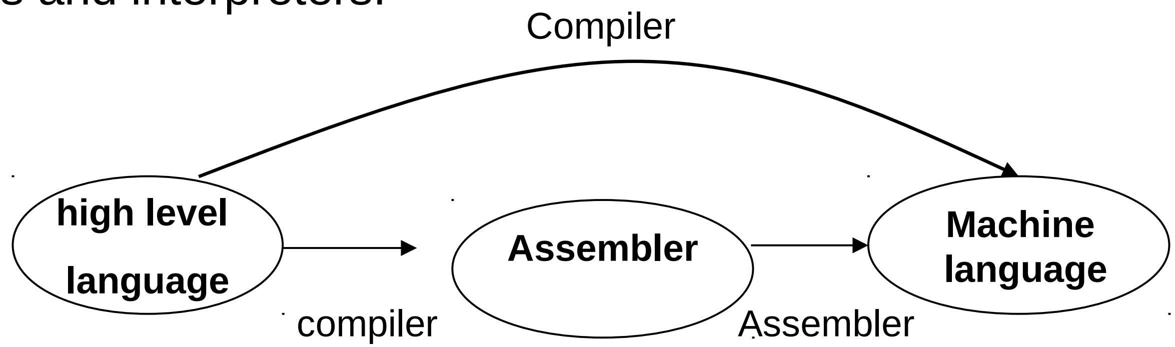
- High level language (example: PASCAL)

```
BEGIN
      Resta:= Minuendo - Sustraendo
 END.
```

- Machine language and assembler (Example: i80x86)

```
A10000          MOV AX, Minuendo
2B060200        SUB AX, Sustrayendo
A30400          MOV Resta, AX
```

- Traduction from high level program to machine language are carried on by compilers and interpreters:


Compiler
high level language
Assembler
Machine language
compiler
Assembler