ARCOS Group

uc3m | Universidad Carlos III de Madrid

L1: Introduction to computers

Computer Structure

Bachelor in Computer Science and Engineering

Bachelor in Applied Mathematics and Computing

Dual Bachelor in Computer Science and Engineering and Business Administration

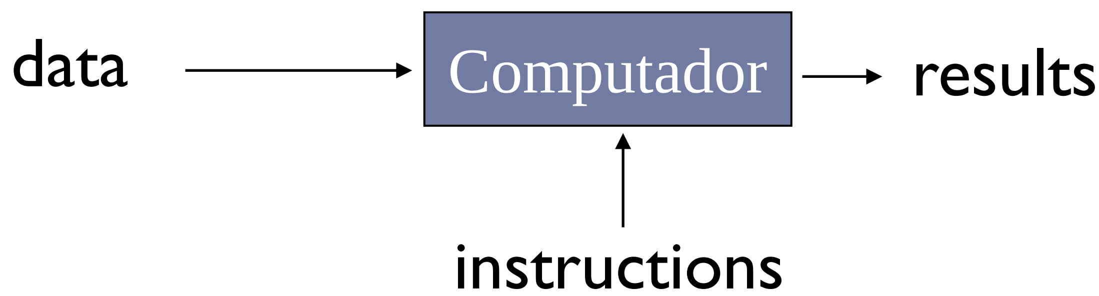
Contents

- Introduction:
  - What is a computer?
  - Building blocks for a computer
  - Concepts of structure and architecture
- Von Neumann computer:
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- Characteristics of a computer and types:
  - Main characteristic parameters of a computer
  - Types of computers
  - Historic evolution

Contents

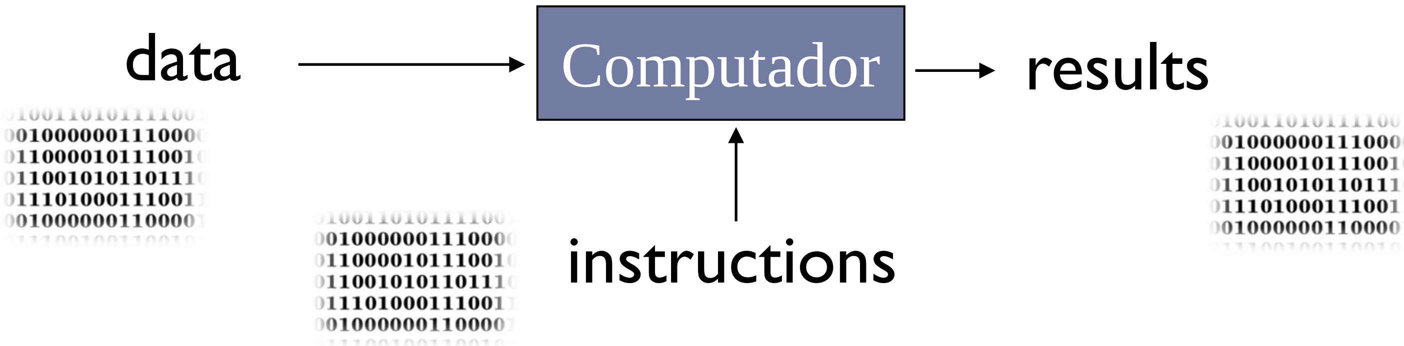
- **Introduction:**
  - **What is a computer?**
  - Building blocks for a computer
  - Concepts of structure and architecture
- Von Neumann computer:
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- Characteristics of a computer and types:
  - Main characteristic parameters of a computer
  - Types of computers
  - Historic evolution

What a computer is?



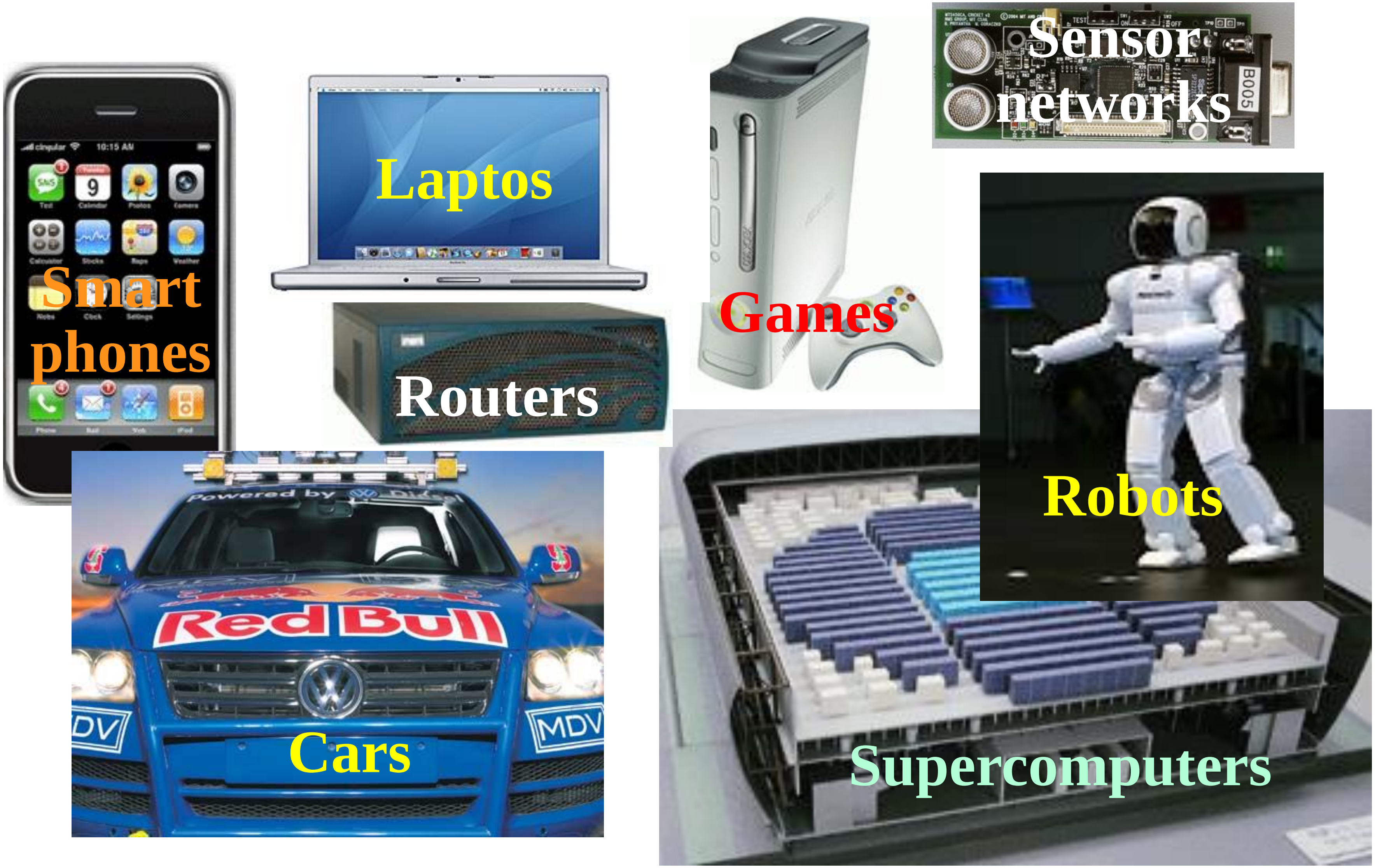
- **Computer: machine designed to process data.**
  - Instructions are applied to data and then results (data/information) are obtained.

What a computer is?

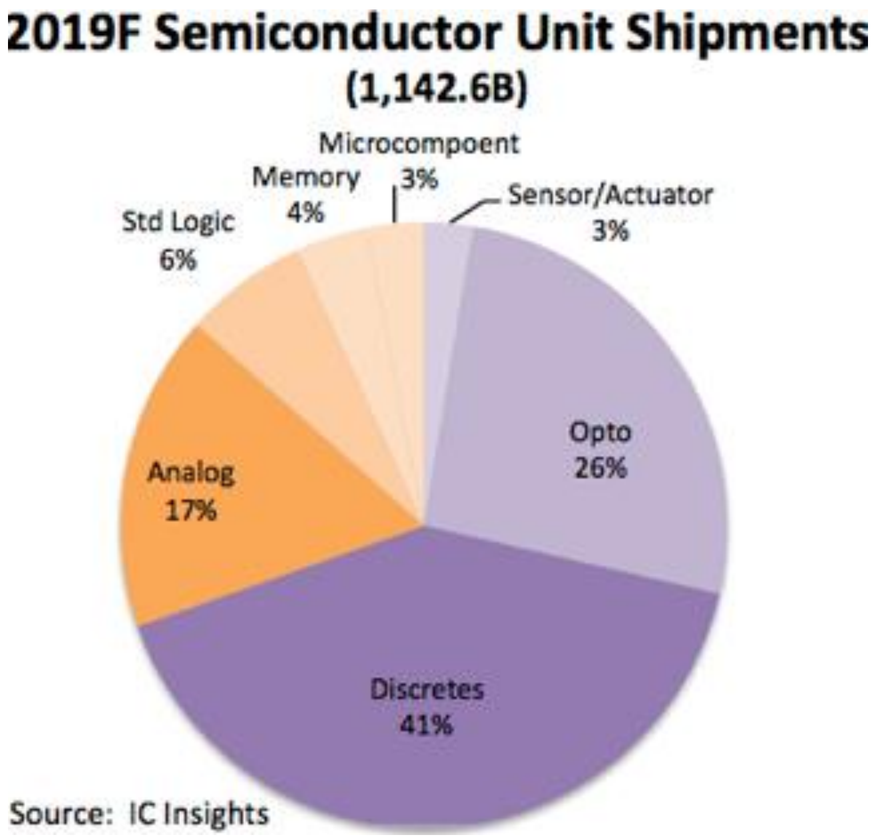


- **Computer: machine designed to process data.**
  - Digital computer: data and instructions in binary format.

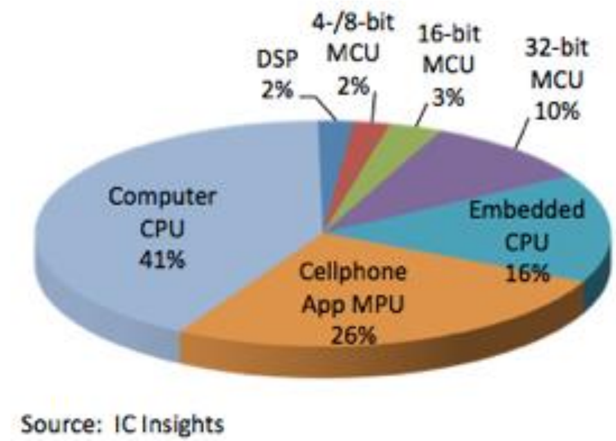
Different types of computers



Semiconductor industry

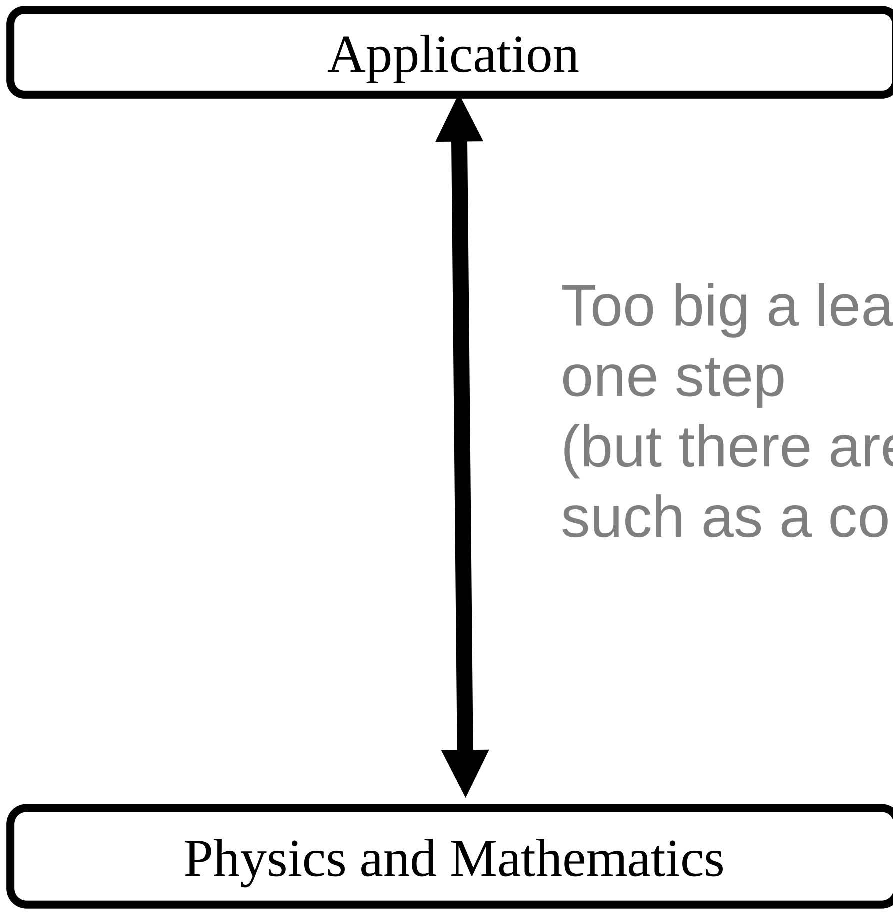


- Processors:
  3% of the industry



Contents

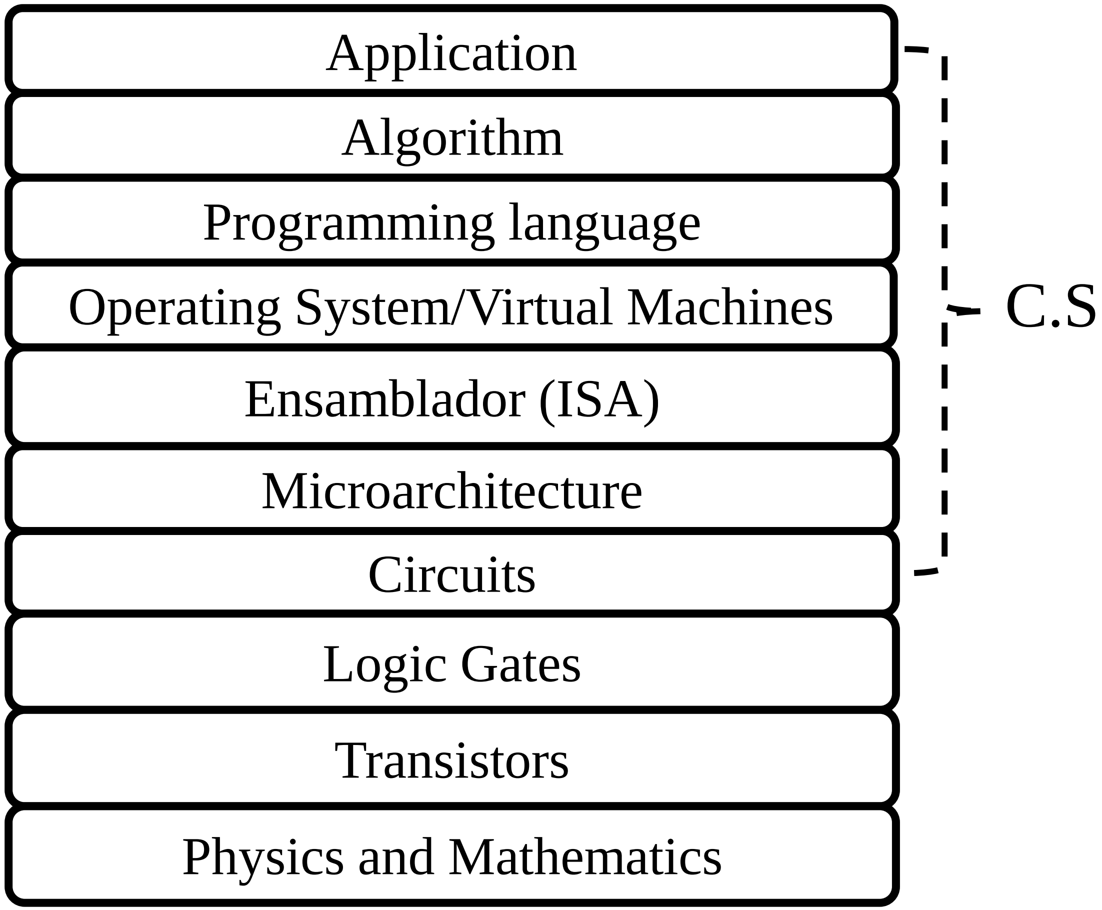
Building blocks



Too big a leap to make in one step
(but there are exceptions such as a compass).

https://cse.buffalo.edu/~stevko/courses/cse490/spring11/schedule.html

Building blocks



https://cse.buffalo.edu/~stevko/courses/cse490/spring11/schedule.html

Félix García-Carballeira, Alejandro Calderón Mateos

Building blocks

Application

Algorithm

Programming language

Operating System/Virtual Machines

Ensamblador (ISA)

Microarchitecture

Circuits

Logic Gates

Transistors

Physics and Mathematics

Binary system based on: 0 y 1

Binary system

- Binary

  X = 1 0 1 0 0 1 0 1   Binary digit d_i

  ... 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 Weight p_i

  - Value = $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \ldots + d_1 \times 2^1 + d_0 \times 2^0$

Binary system

- Binary

$$X = 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1$$

$$\ldots 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

Binary digit d_i

Weight p_i

- Value = $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \ldots + d_1 \times 2^1 + d_0 \times 2^0$

- How many values can be represented with n bits?
- How many bits are necessary to represent m 'values'?

- With n bits, if the values to be represented are numbers and start at 0, what is the maximum representable value?

Binary system

- Binary

  Binary digit d_i

  $X = 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1$

  Weight p_i

  $\ldots 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

  - Value = $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \ldots + d_1 \times 2^1 + d_0 \times 2^0$

  - How many values can be represented with n bits? **2^n**
  - How many bits are necessary to represent m 'values'? **$\log_2(m)$ rounded up**
  - With n bits, if the values to be represented are numbers and start at 0, what is the maximum representable value? **2^n-1**

Félix García-Carballeira, Alejandro Calderón Mateos

Building blocks

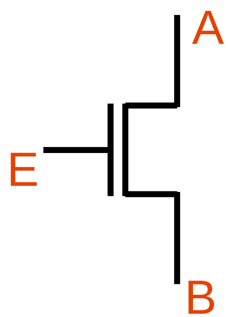
Application
Algorithm
Programming language
Operating System/Virtual Machines
Ensamblador (ISA)
Microarchitecture
Circuits
Logic Gates
Transistors
Physics and Mathematics

Electronic building blocks…

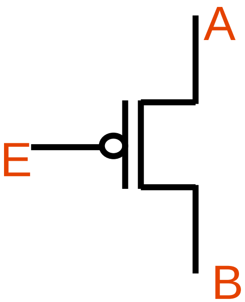
https://cse.buffalo.edu/~stevko/courses/cse490/spring11/schedule.html

Transistor

N-MOS



P-MOS



E	Behavior
1	Connects A to B (open circuit)
0	Does not connect A to B (closed circuit)

E	Behavior
1	Connects A to B (open circuit)
0	Does not connect A to B (closed circuit)

- A transistor acts as a switch
- The p-type and n-type transistors are MOSFET (Metal-Oxide-Semiconductor-Field-Effect Transistor) transistors.
- Origin of CMOS family in the combination of p-type and n-type transistors.

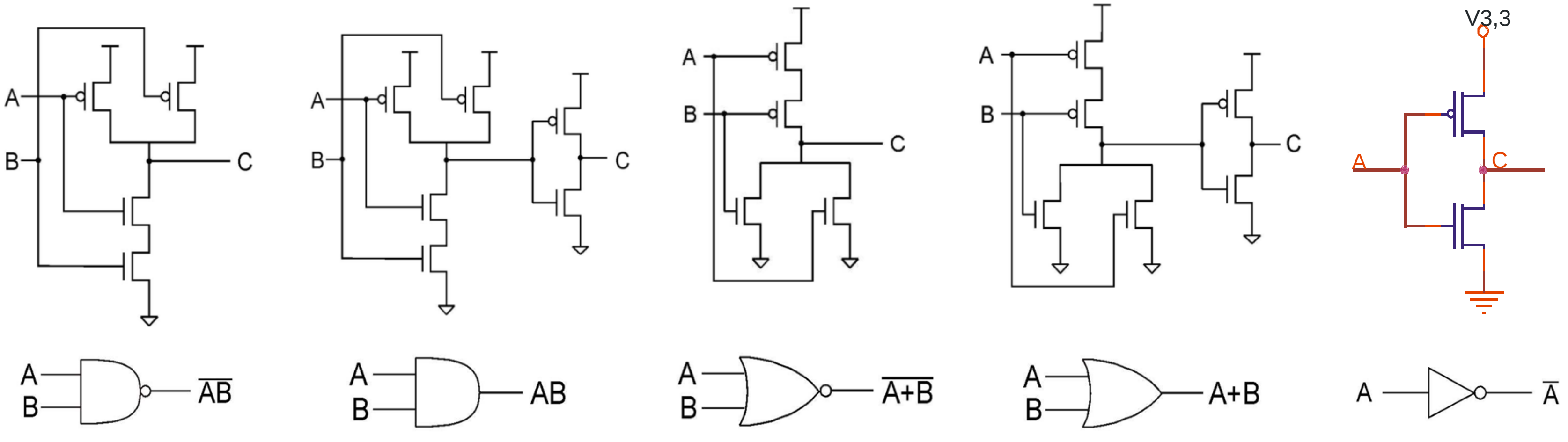
Building blocks

- Application
- Algorithm
- Programming language
- Operating System/Virtual Machines
- Ensamblador (ISA)
- Microarchitecture
- Circuits
- Logic Gates
- Transistors
- Physics and Mathematics

Electronic building blocks…

Logic gates

NAND AND NOR OR NOT



NAND

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

AND

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

NOR

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

OR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

NOT

A	C
1	0
0	1

Félix García-Carballeira, Alejandro Calderón Mateos

Building blocks

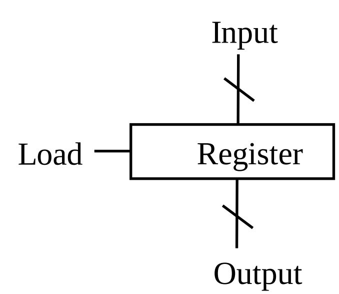
- Application
- Algorithm
- Programming language
- Operating System/Virtual Machines
- Ensamblador (ISA)
- Microarchitecture
- Circuits
- Logic Gates
- Transistors
- Physics and Mathematics

Electronic building blocks…

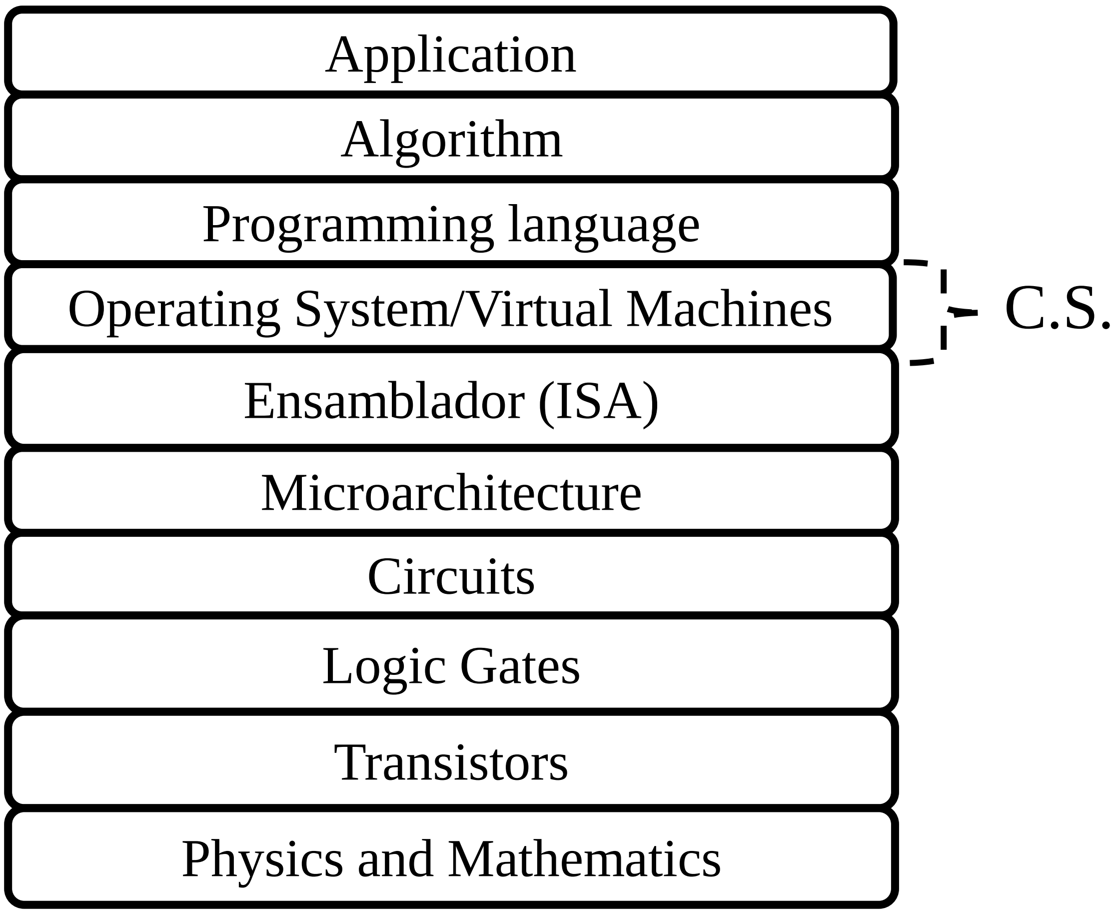
https://cse.buffalo.edu/~stevko/courses/cse490/spring11/schedule.html

Combinational and sequential circuits

- Combinational circuits:
  - the output depends only on the input values
  - Examples:
    - Decoders, Multiplexers, Arithmetic and logical operators, …
- Sequential circuits:
  - Output depends on input and current state.
    It stores information
  - Examples:
    - Flip-flops, registers, …



Building blocks

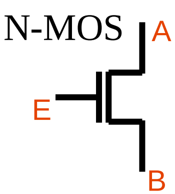


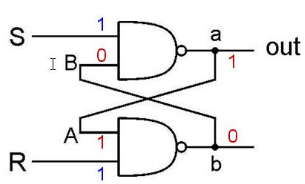
https://cse.buffalo.edu/~stevko/courses/cse490/spring11/schedule.html

Contents

- **Introduction:**
  - What is a computer?
  - Building blocks for a computer
  - **Concepts of structure and architecture**
- Von Neumann computer:
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- Characteristics of a computer and types:
  - Main characteristic parameters of a computer
  - Types of computers
  - Historic evolution

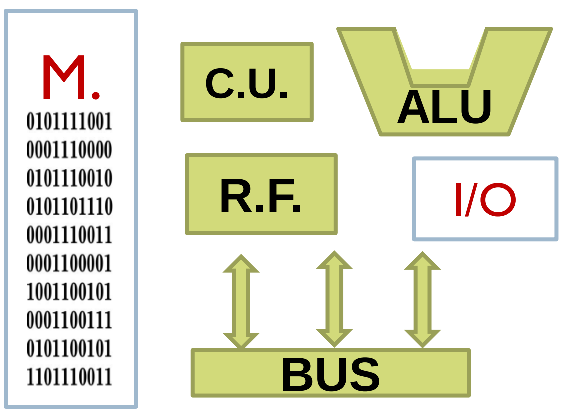
What aspects of a computer do I need to know?

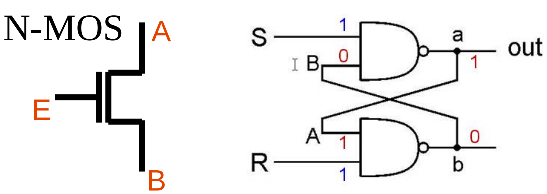




- **Technology:**
  - How components are built

What aspects of a computer do I need to know?

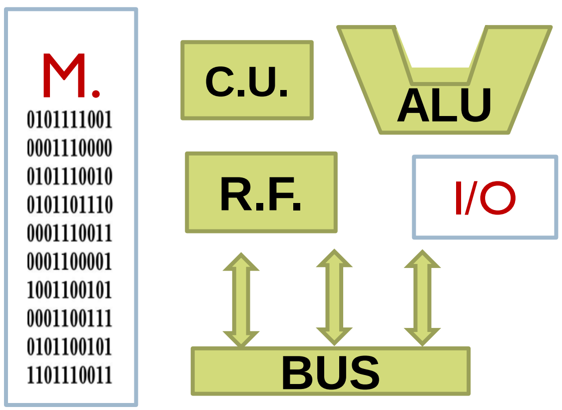




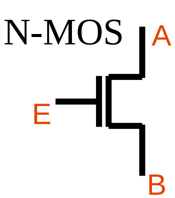
- **Structure:**
  - Components and their organization
- **Technology:**
  - How components are built

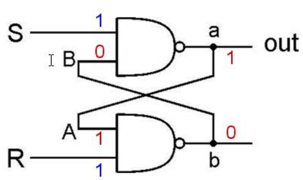
What aspects of a computer do I need to know?

- **Architecture:**
  - Attributes visible to a programmer



- **Structure:**
  - Components and their organization





- **Technology:**
  - How components are built

Structure and Architecture

- Structure
  - Components of a computer
  - Organization of the components
- Architecture: visible attributes for programmers
  - Instruction set offered by the computer (ISA, Instruction Set Architecture)
  - Type and format of data that the computer is capable of using.
  - Number and size of registers
  - Input/Output (I/O) techniques and mechanisms
  - Addressing and memory access techniques
- Technology: how the components are built

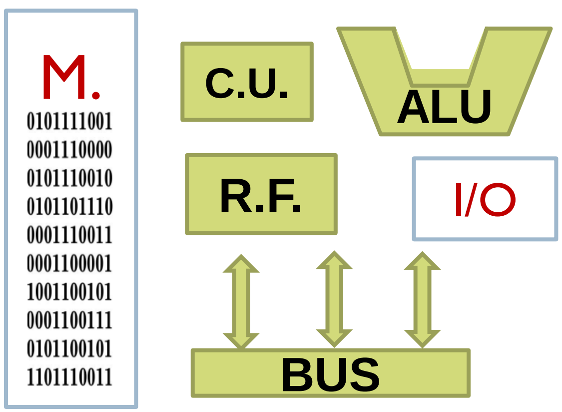
Félix García-Carballeira, Alejandro Calderón Mateos

Contents

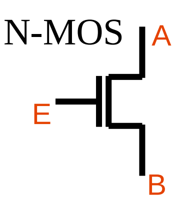
- Introduction:
  - What is a computer?
  - Building blocks for a computer
  - Concepts of structure and architecture
- **Von Neumann computer:**
  - **Von Newmann architecture**
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- Characteristics of a computer and types:
  - Main characteristic parameters of a computer
  - Types of computers
  - Historic evolution

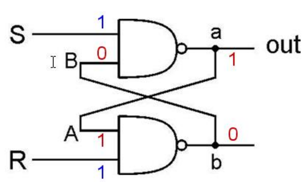
What aspects of a computer do I need to know?

- **Architecture:**
  - Attributes visible to a programmer



- **Structure:**
  - Components and their organization

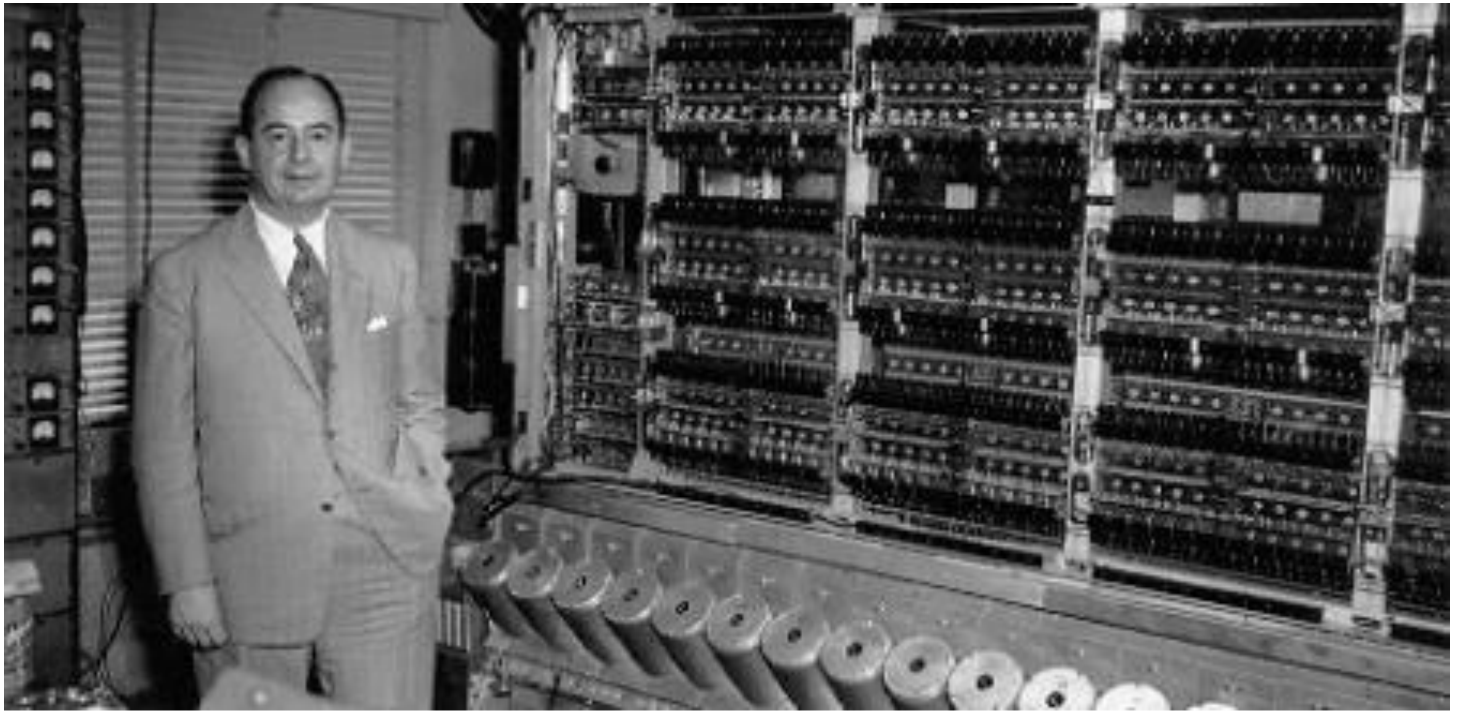




- **Technology:**
  - How components are built

Félix García-Carballeira, Alejandro Calderón Mateos

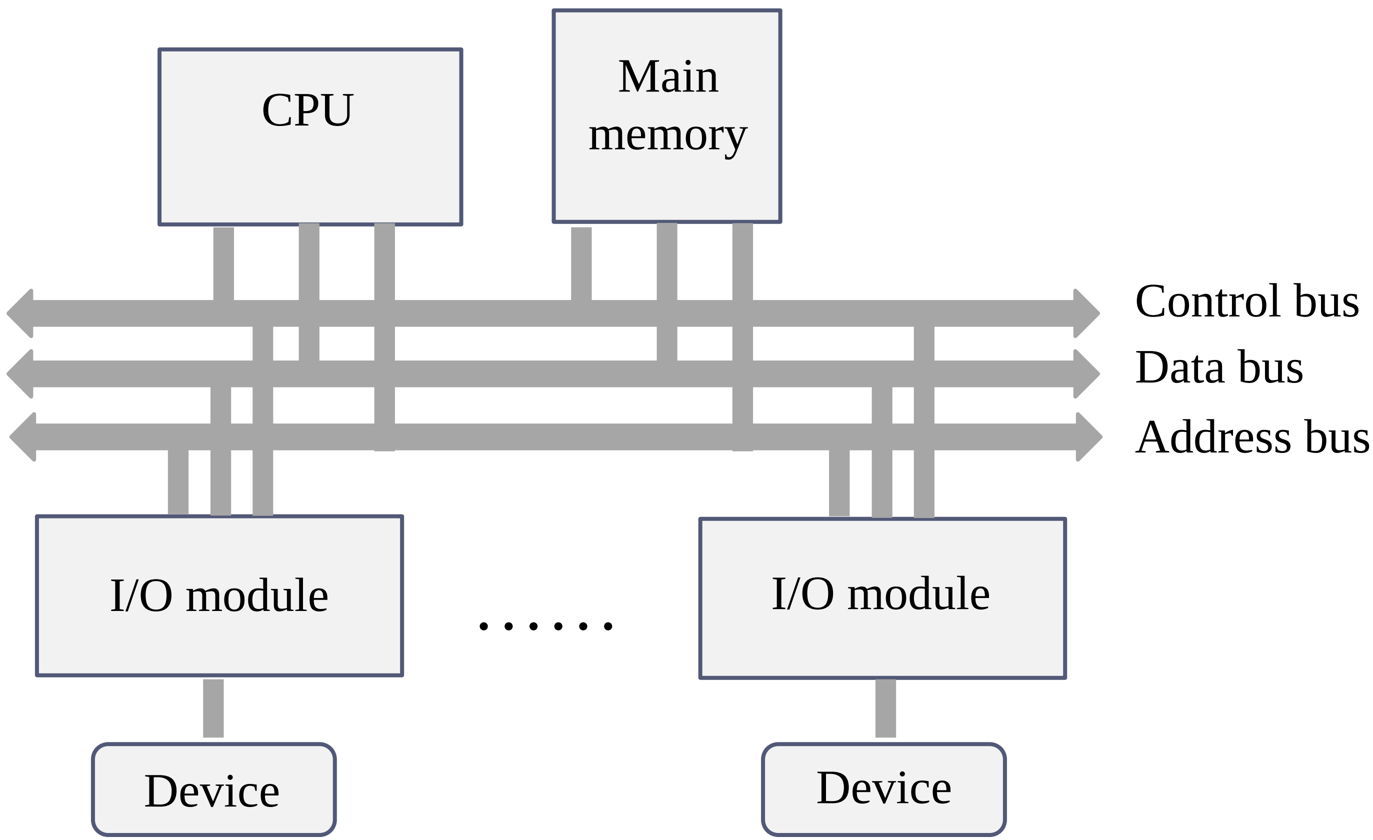
Von Neumann computer



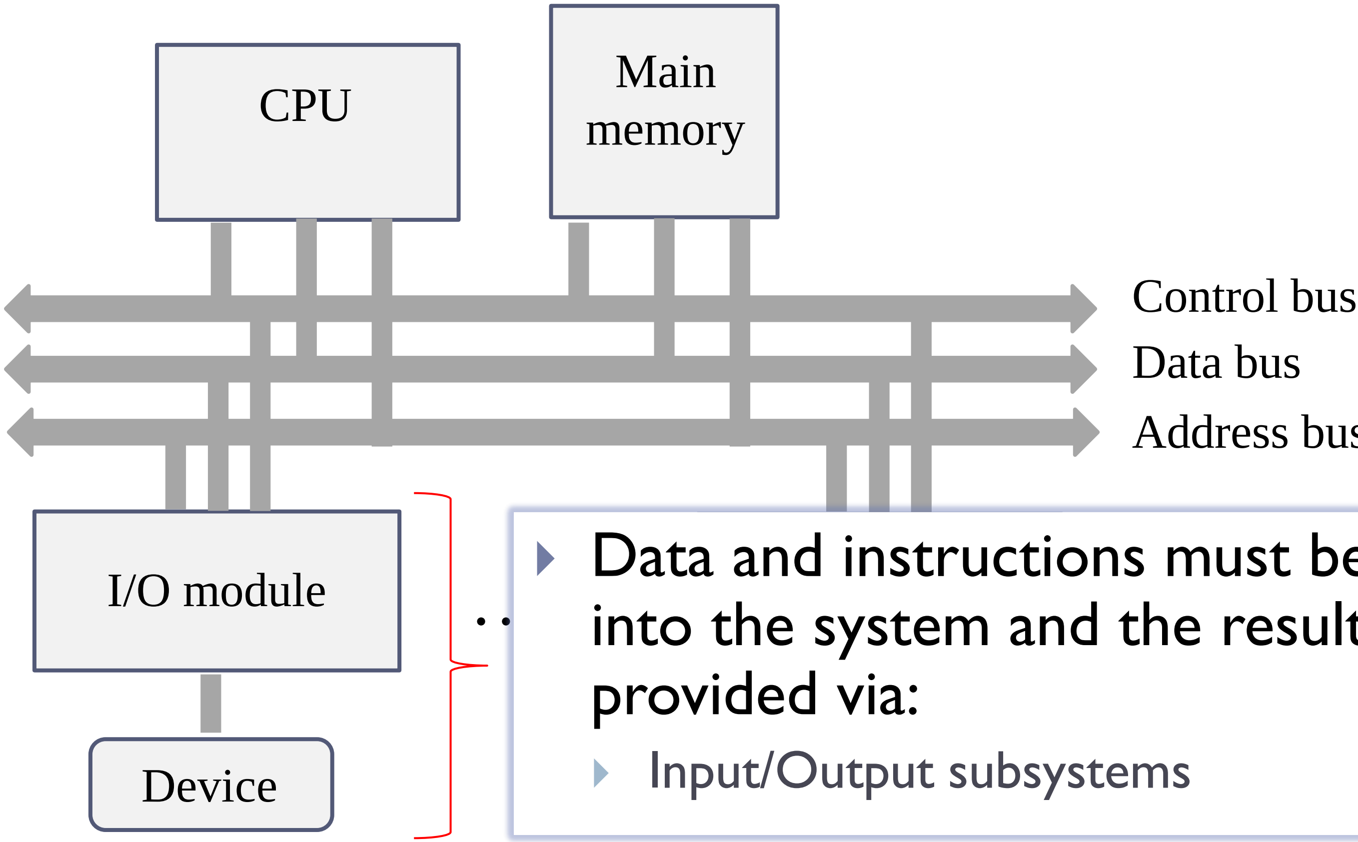
Machine capable of executing a series of elementary instructions (machine instructions) that are stored in memory (read and executed).

https://blogthinkbig.com/john-von-neumann

Von Neumann architecture

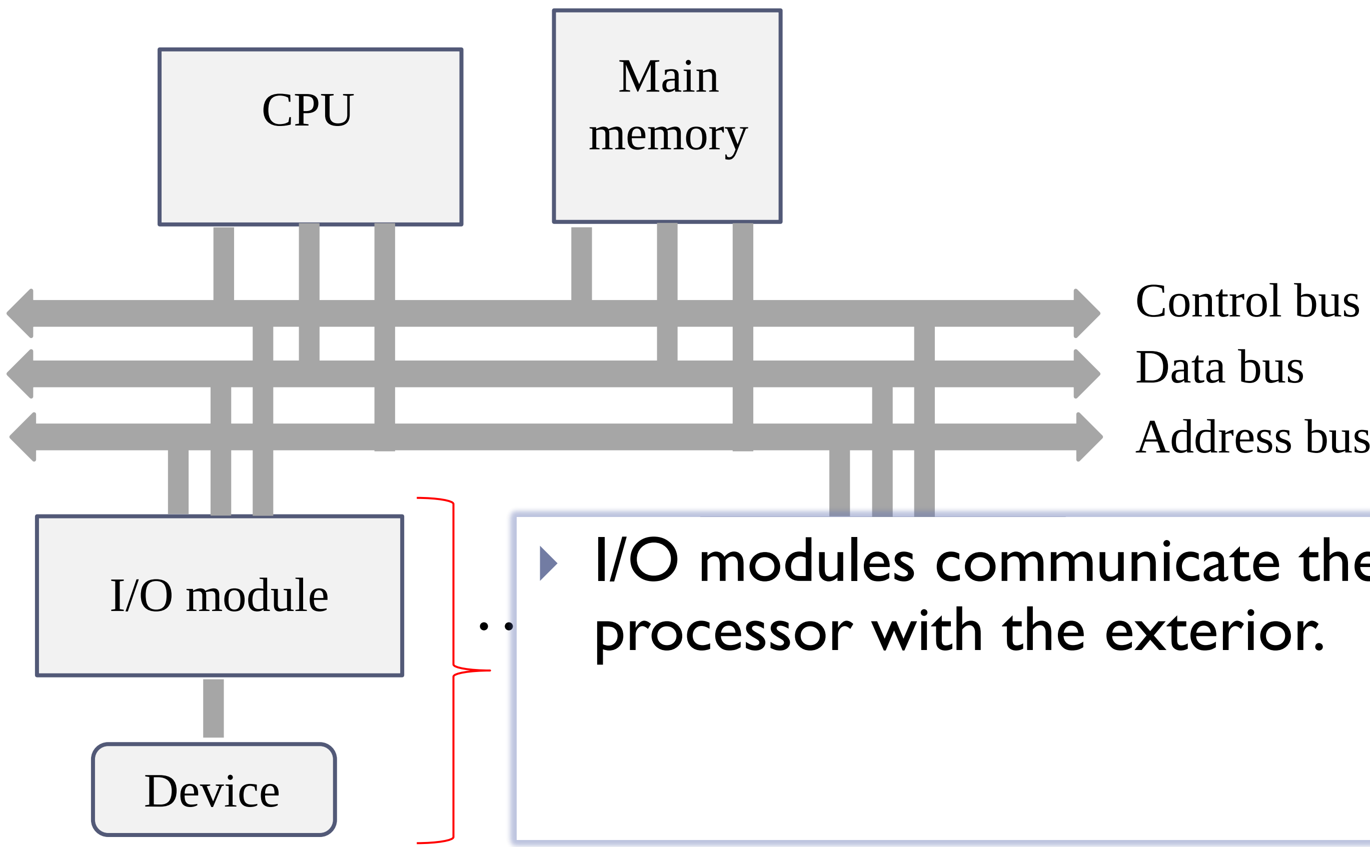


Von Neumann architecture(1/4)



- Data and instructions must be entered into the system and the results are provided via:
  - Input/Output subsystems

Von Neumann architecture (1/4)



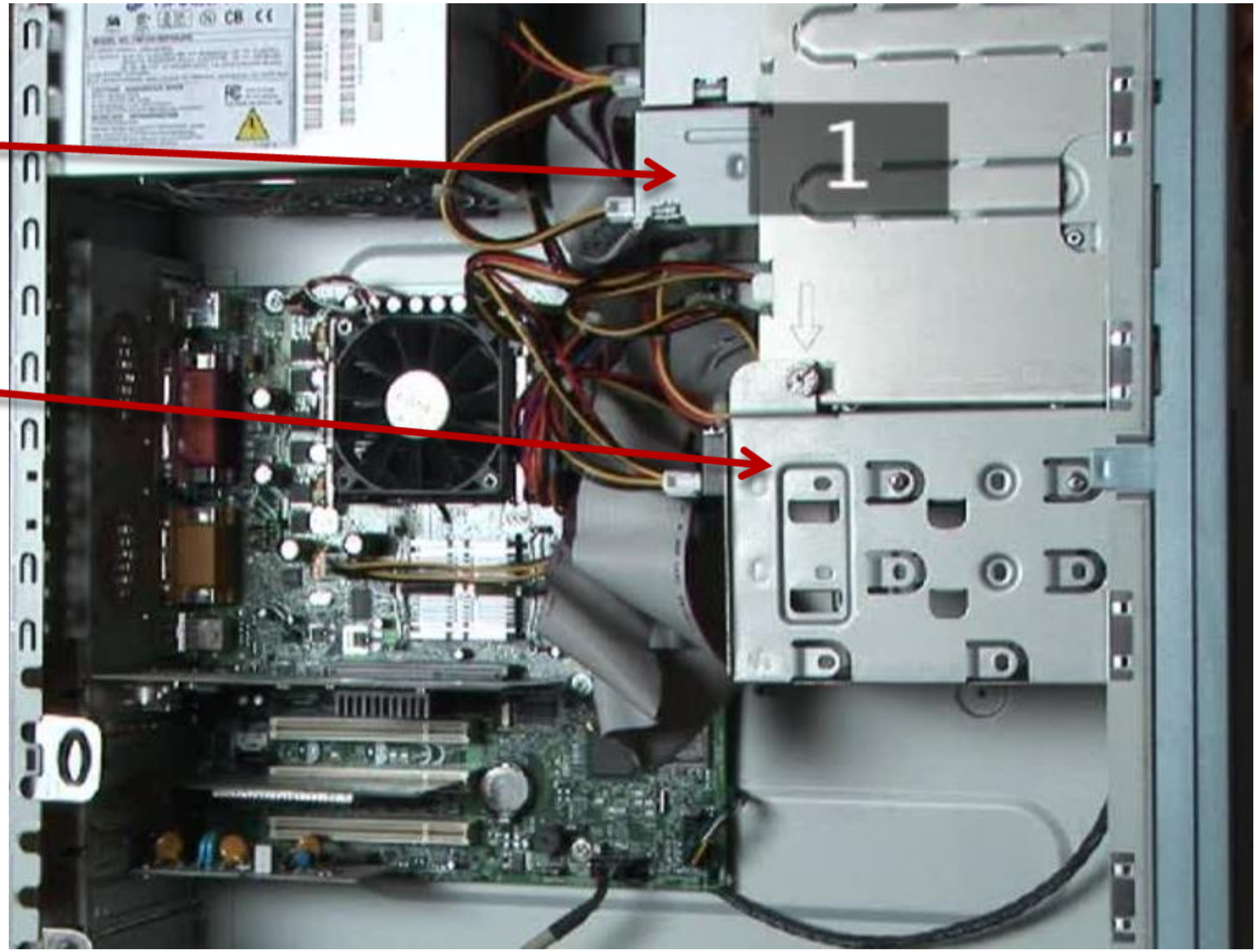
- I/O modules communicate the processor with the exterior.

Félix García-Carballeira, Alejandro Calderón Mateos

Example of I/O module + devices
storage

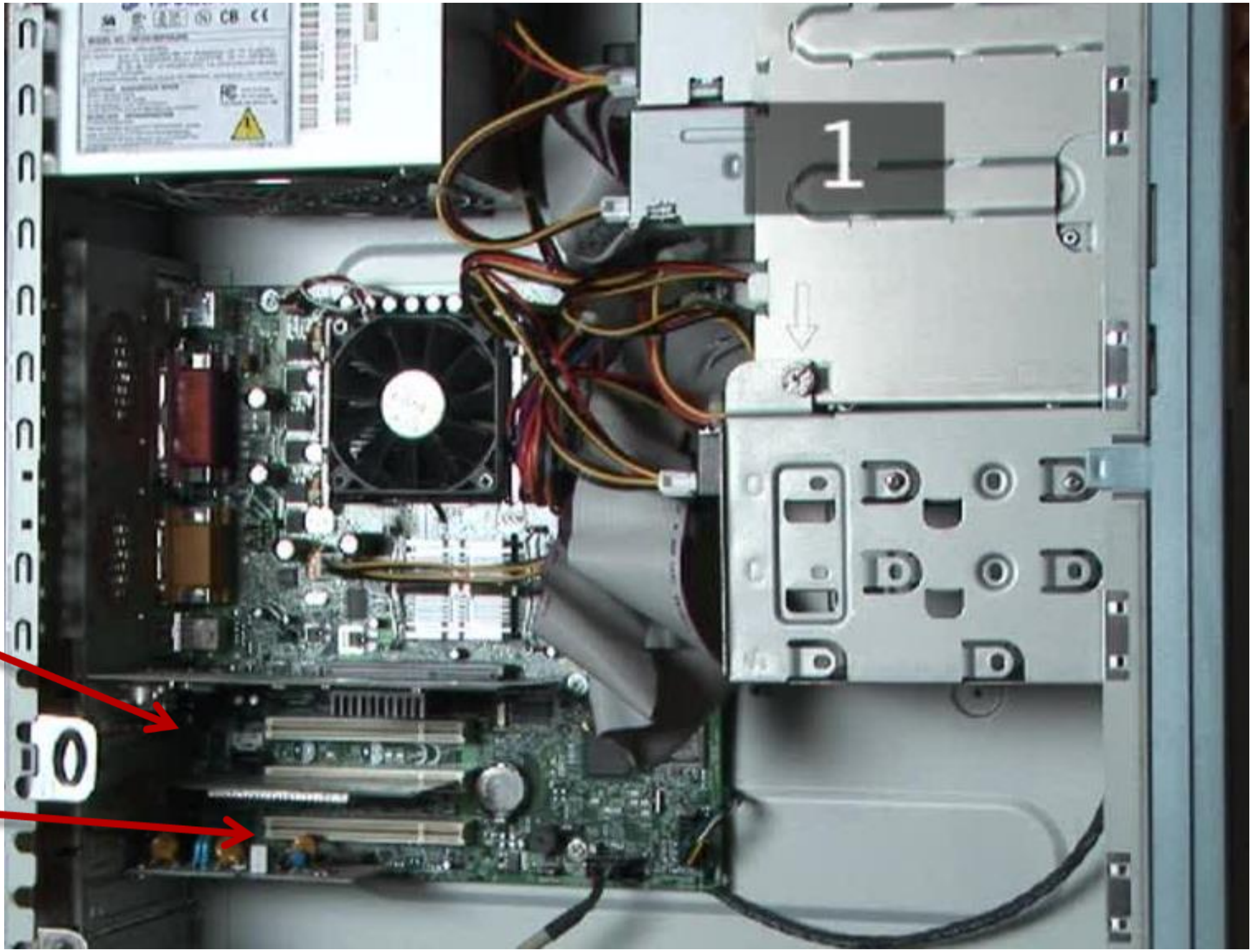
CD-ROM/
DVD-ROM/
BluRay/…

Hard disk



http://www.videojug.com/film/what-components-are-inside-my-computer

Example of I/O module + devices communication

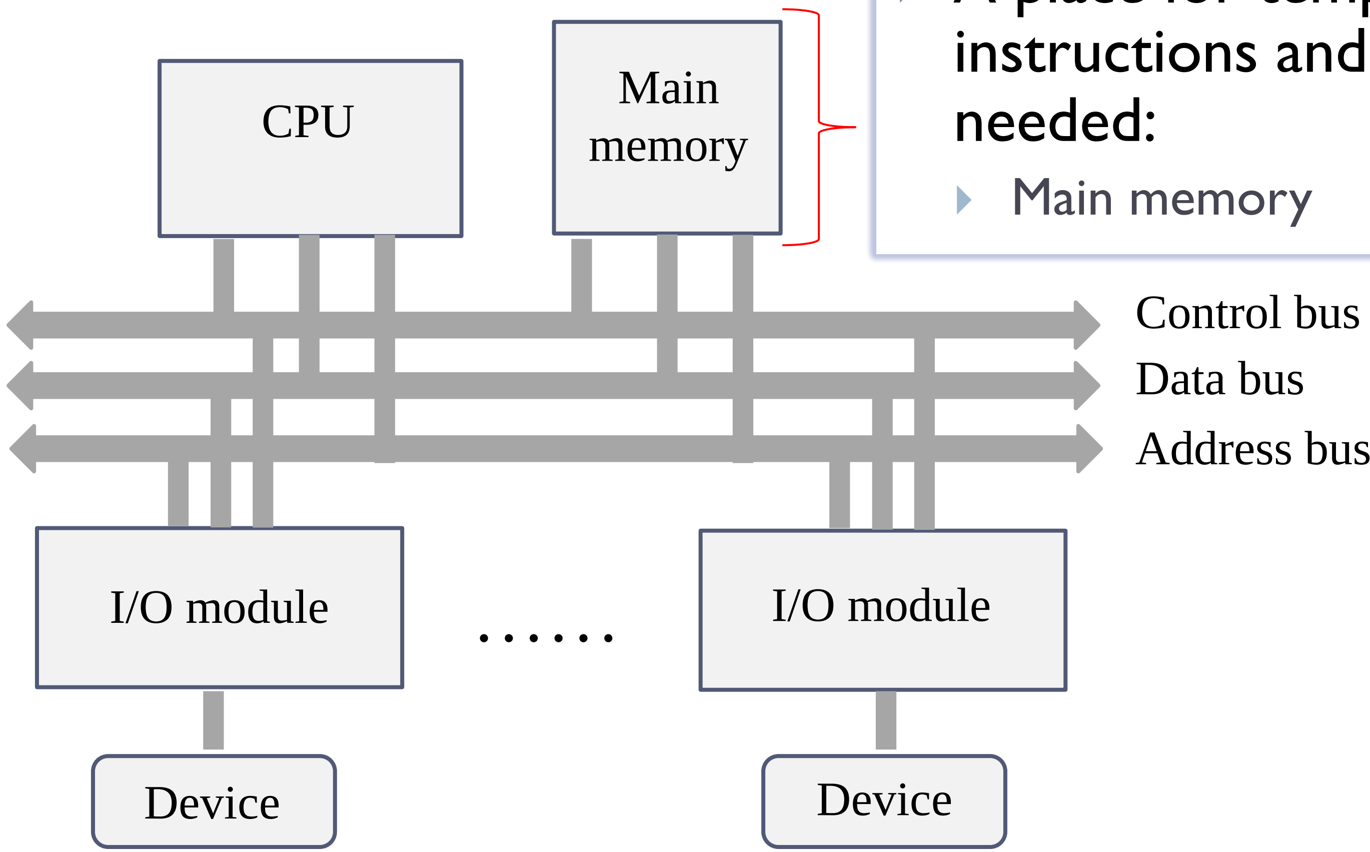


Tarjeta de red

Tarjeta de sonido

http://www.videojug.com/film/what-components-are-inside-my-computer

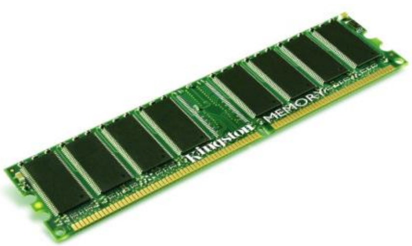
Von Neumann architecture (2/4)

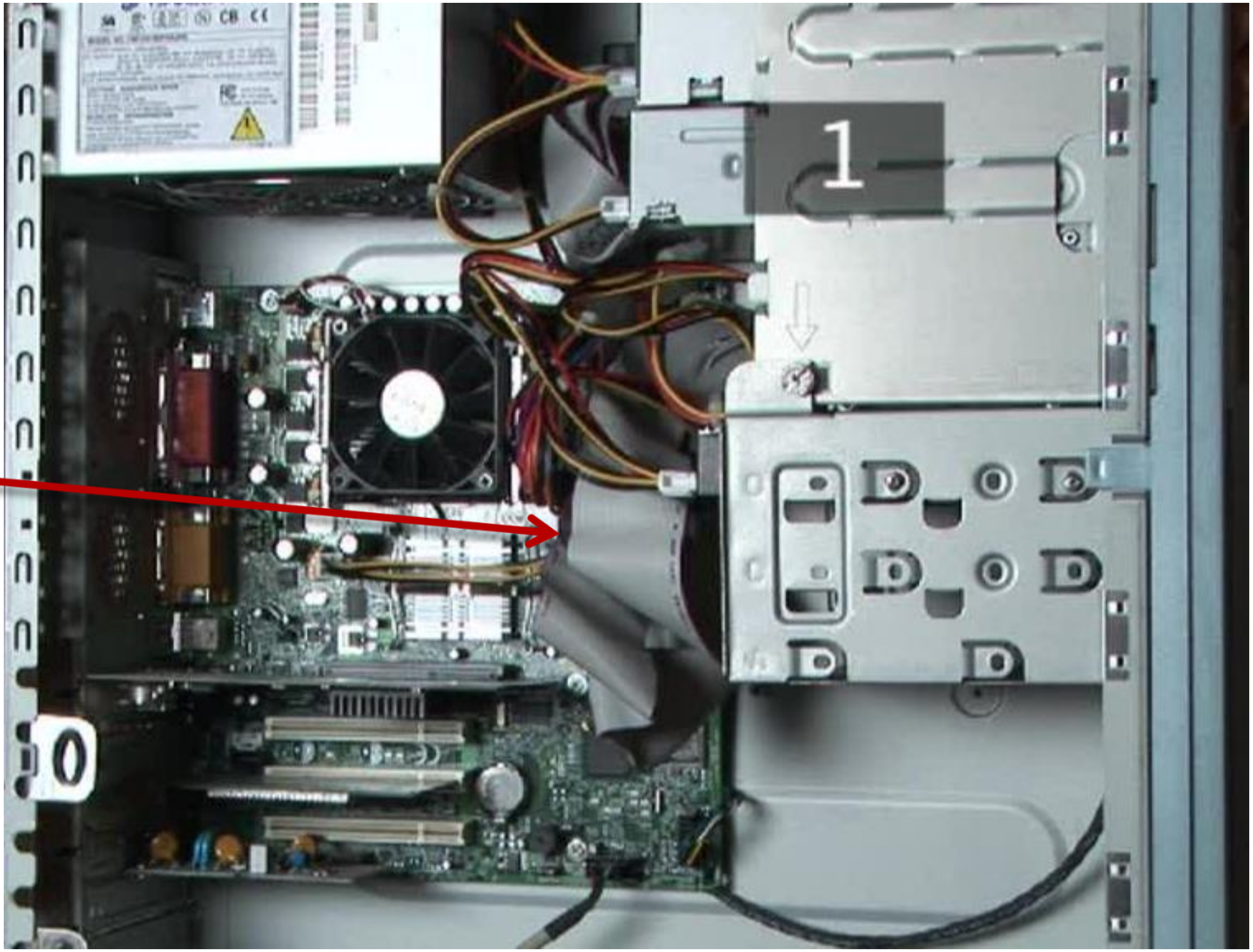


- A place for temporary store instructions and data is needed:
  - Main memory

Example: main memory

Main memory

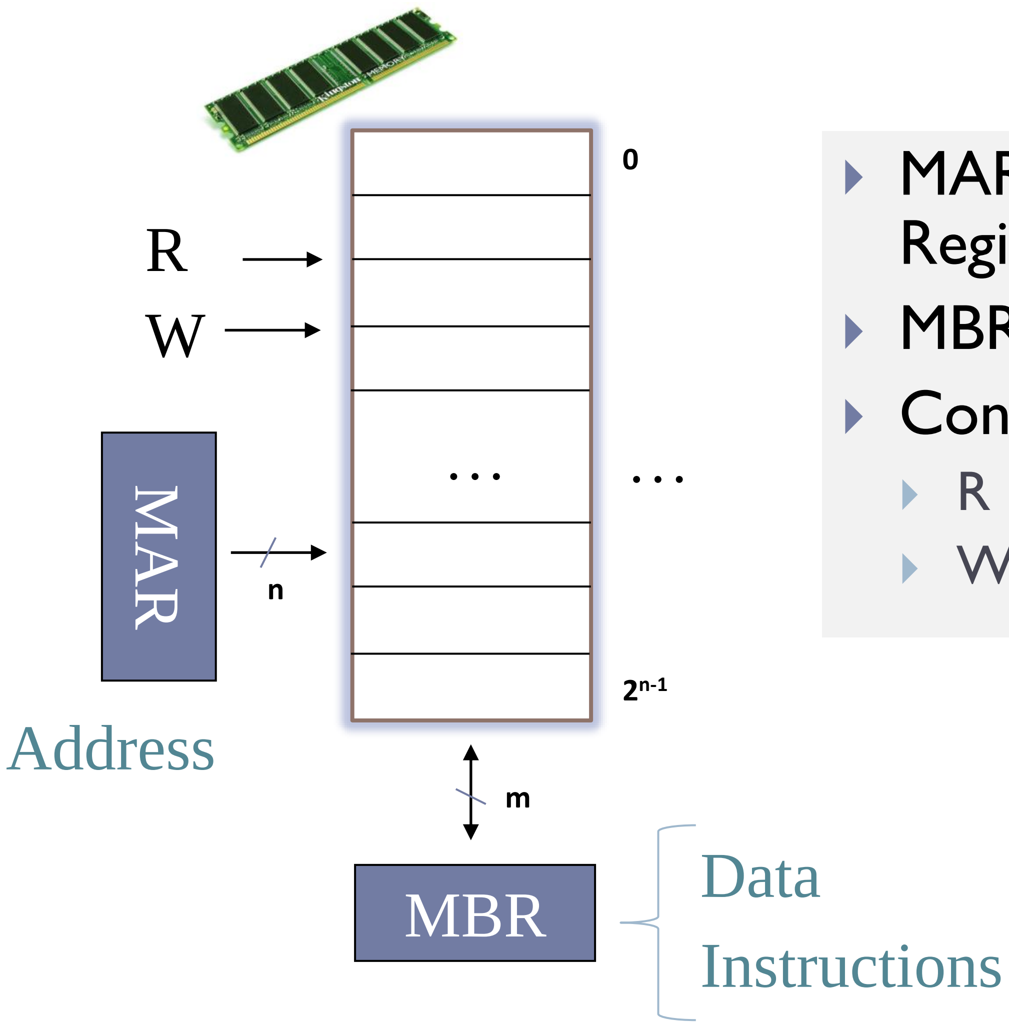




http://www.videojug.com/film/what-components-are-inside-my-computer

Félix García-Carballeira, Alejandro Calderón Mateos

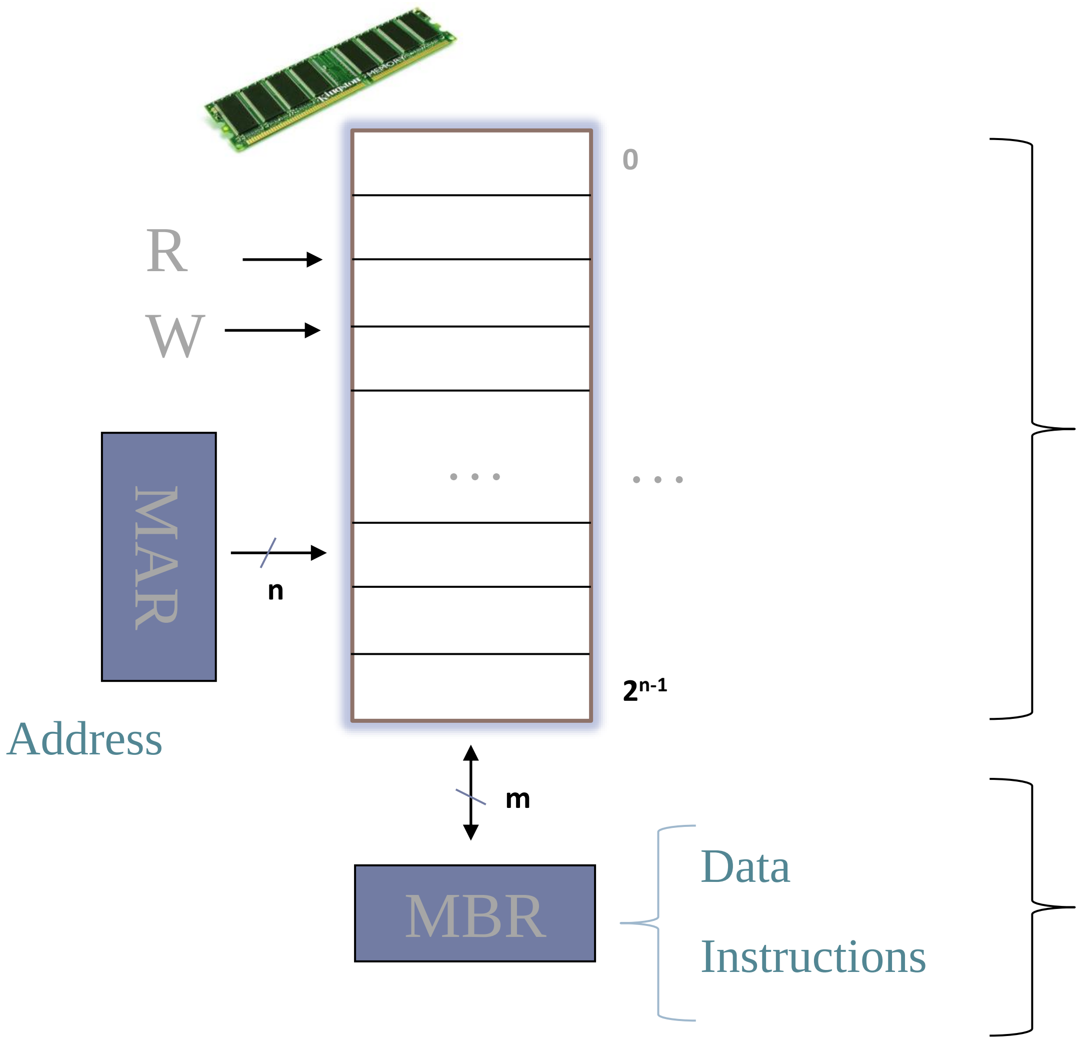
Main memory elements



- MAR: Memory Address Register
- MBR: Memory Buffer Register
- Control signals
  - R – Read from memory
  - W – Write to memory

Félix García-Carballeira, Alejandro Calderón Mateos

Main memory elements



Address space:

Number of memory locations

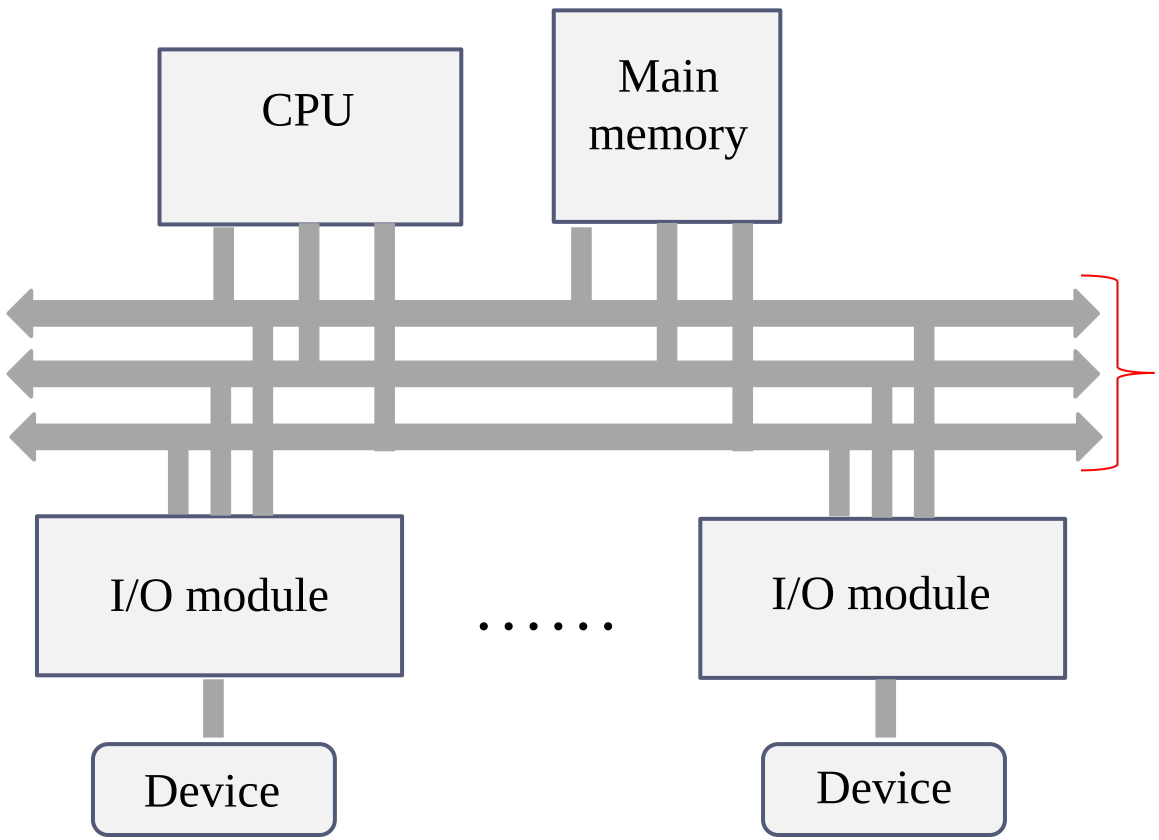
2^n locations

Size of each location:

Number of bits per location

2^m values

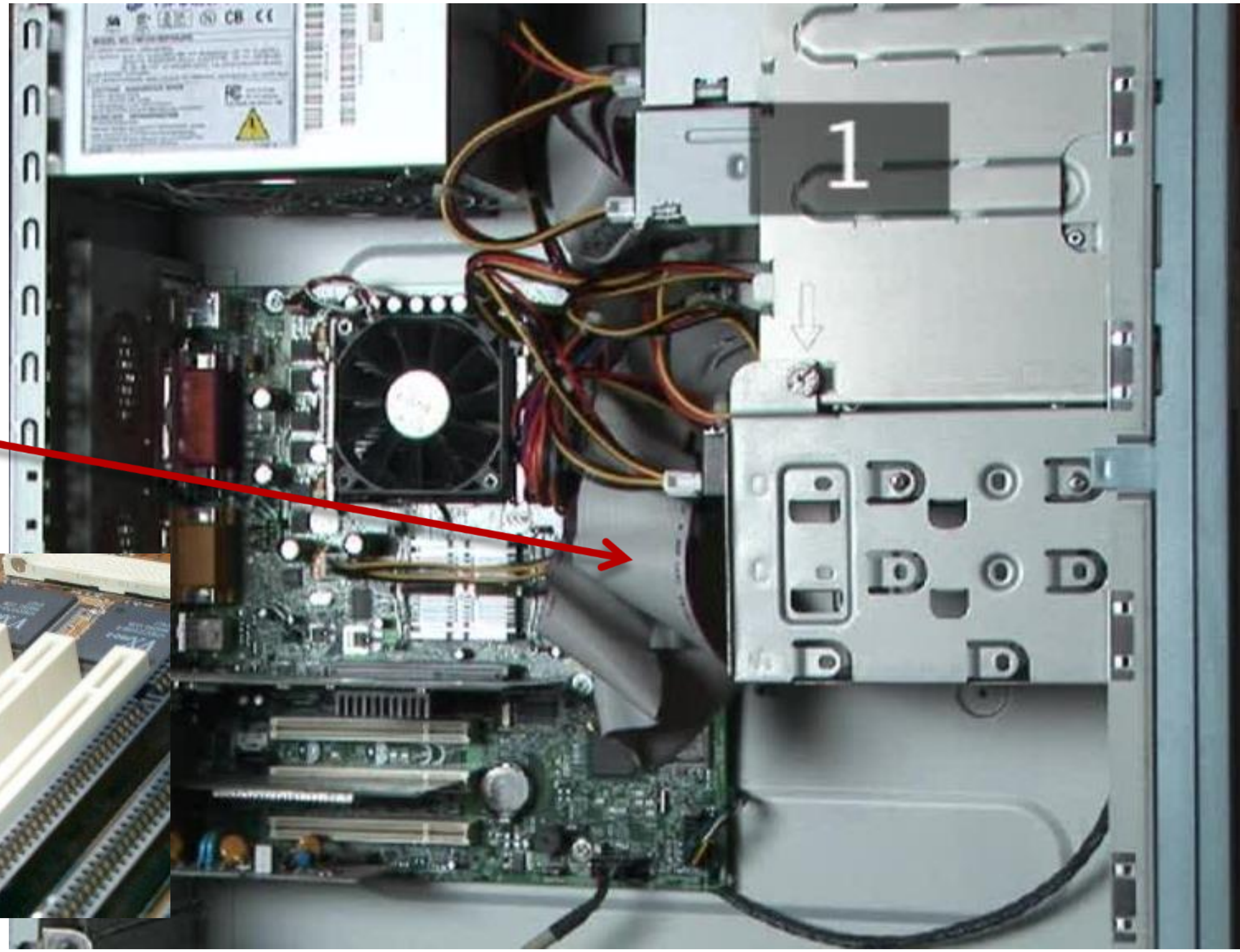
Von Neumann architecture (3/4)



- The different parts of the computer must communicate each other:
  - Buses

Example of buses

Buses



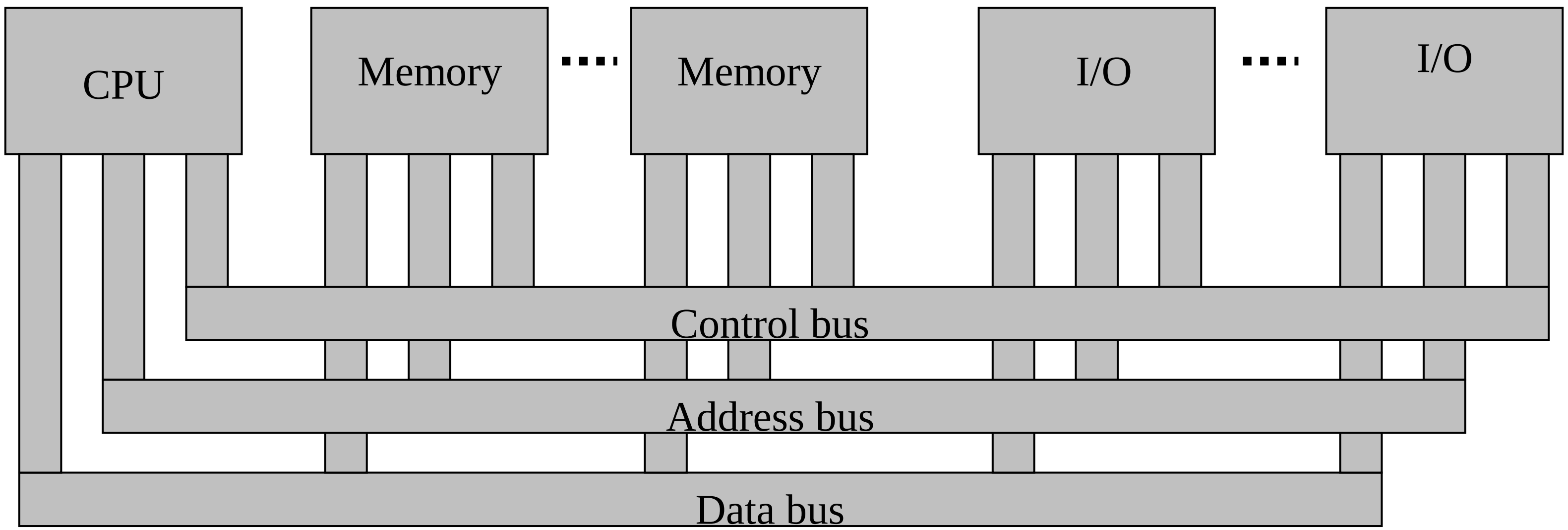
http://www.videojug.com/film/what-components-are-inside-my-computer

Félix García-Carballeira, Alejandro Calderón Mateos

Buses

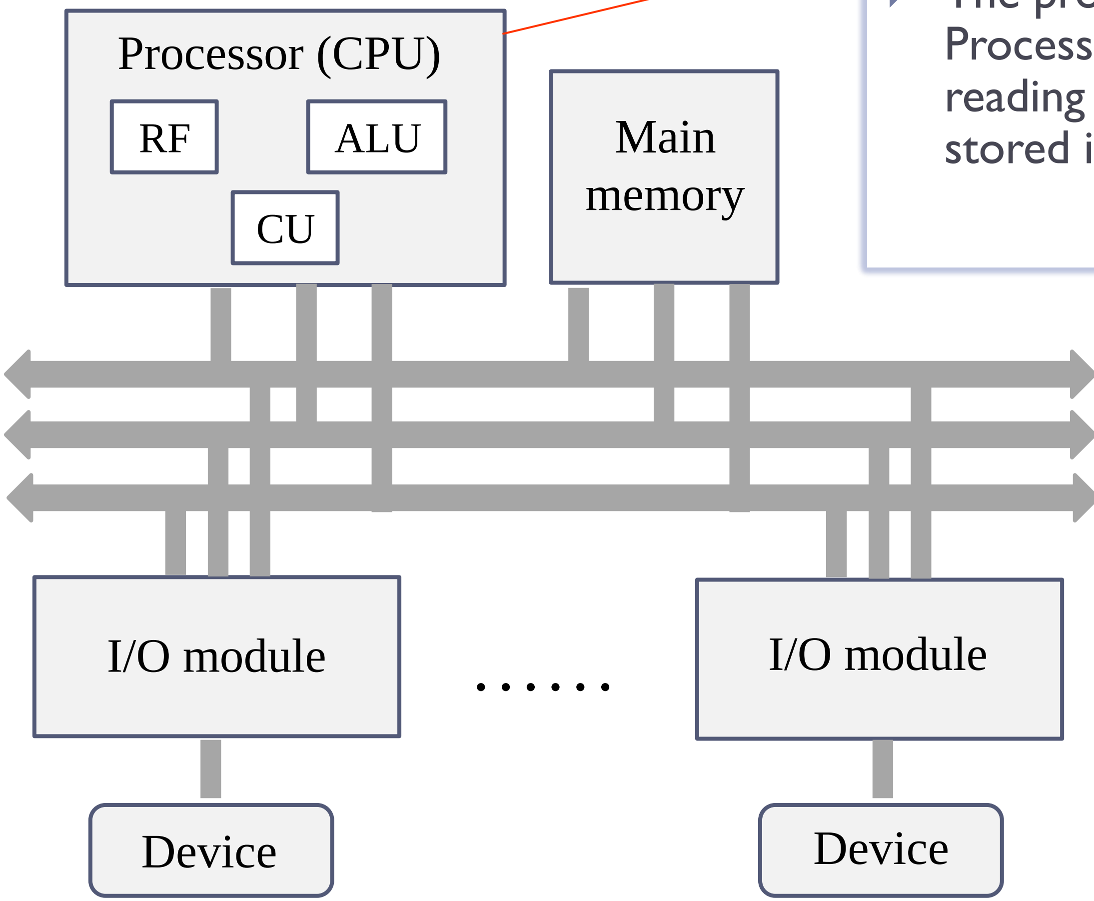
- A bus is a communication path between two or more elements (CPU, memory, ...) for the transmission of information between them.
- A bus usually consists of several communication lines, each transmitting one bit.
  - The width of the bus represents the size at which the computer works (example: a 32-bit computer has 32 buses).
- Three main types: data, address and control.

Bus interconnection diagram



- **Control bus**: control and timing signals.
- **Address bus**: designates the source or destination of a data.
  - Its width determines the maximum memory capacity of the system.
- **Data bus**: data movement between components.

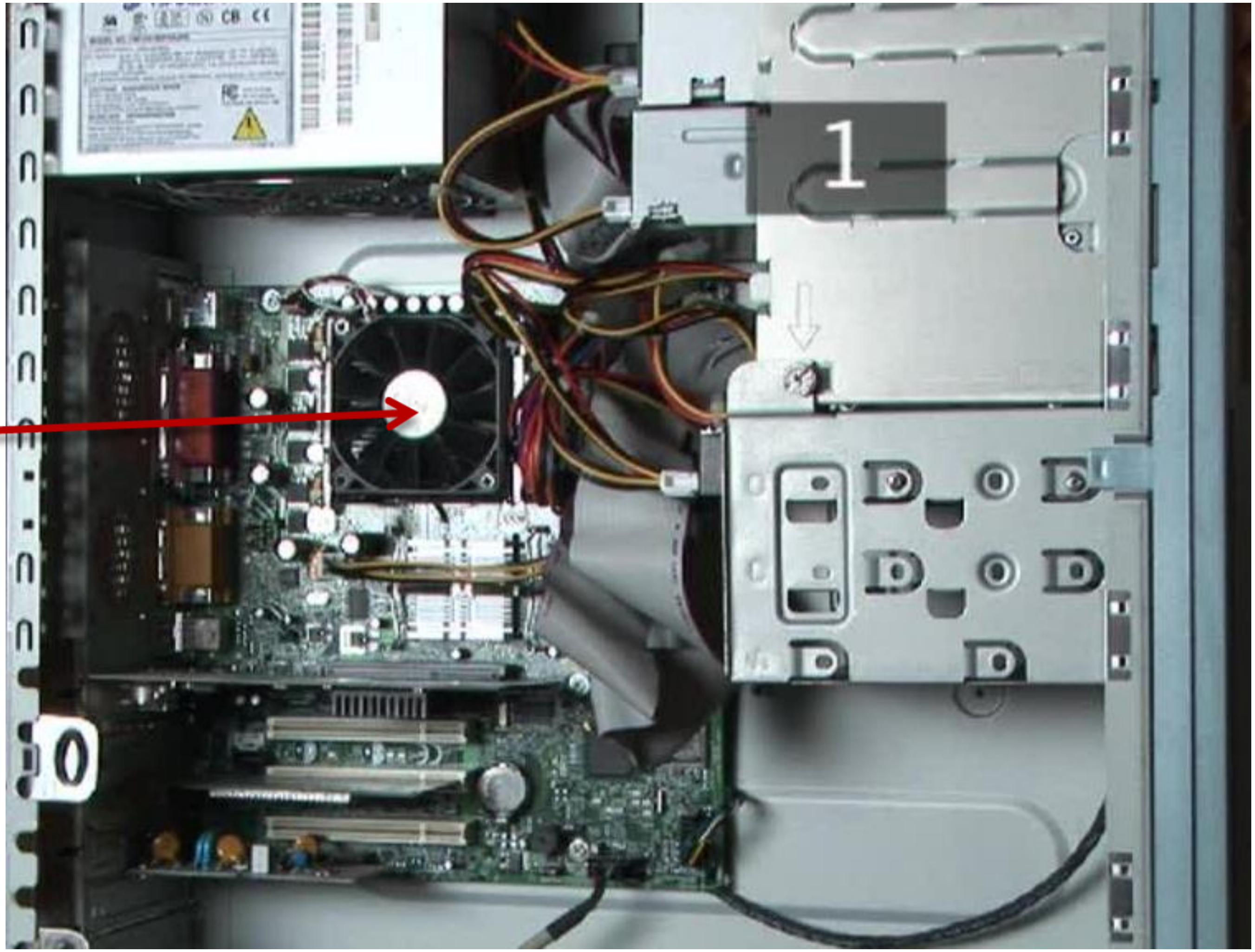
Von Neumann architecture (4/4)



- The processor or CPU (Central Processing Unit) is responsible for reading and executing the instructions stored in the main memory.

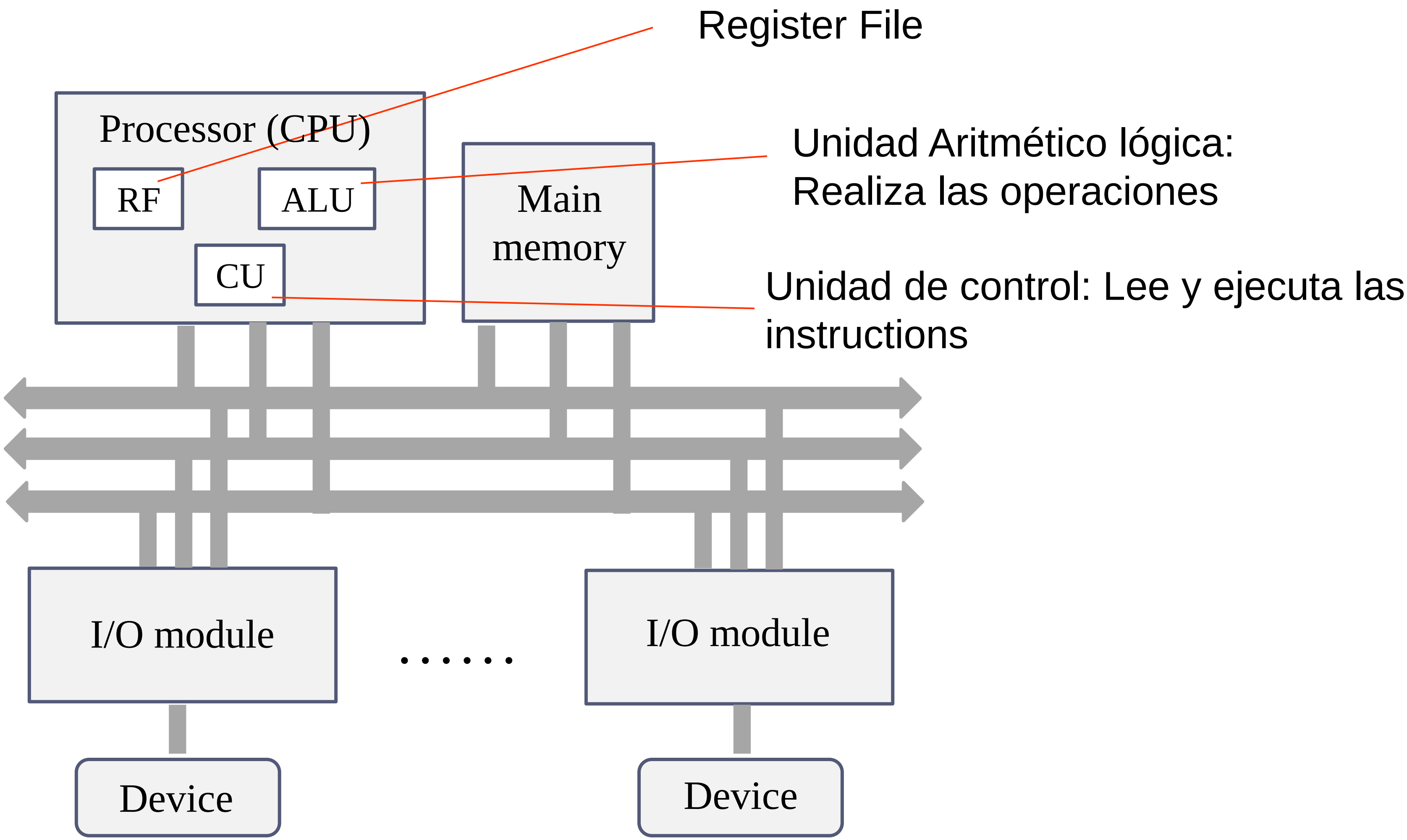
Example of CPU

CPU

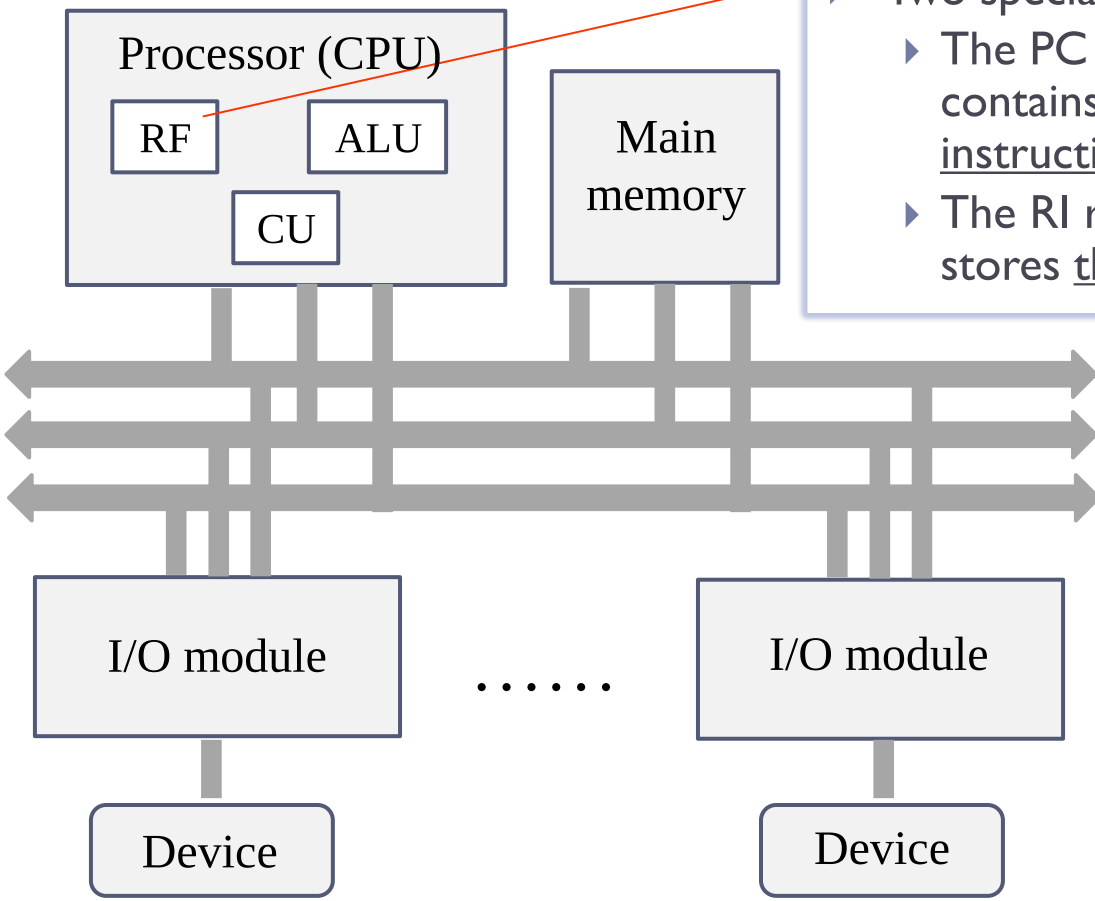


http://www.videojug.com/film/what-components-are-inside-my-computer

Von Neumann architecture(4/4)

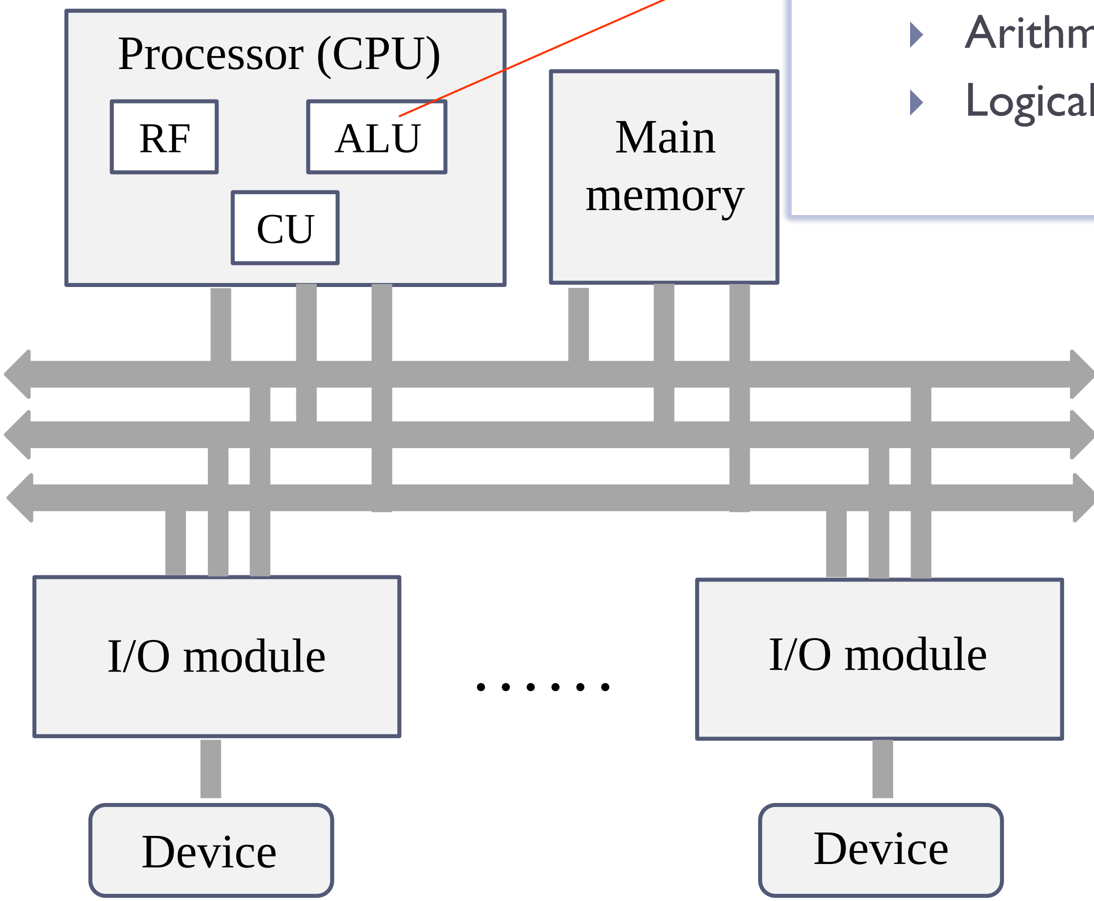


Processor: registros



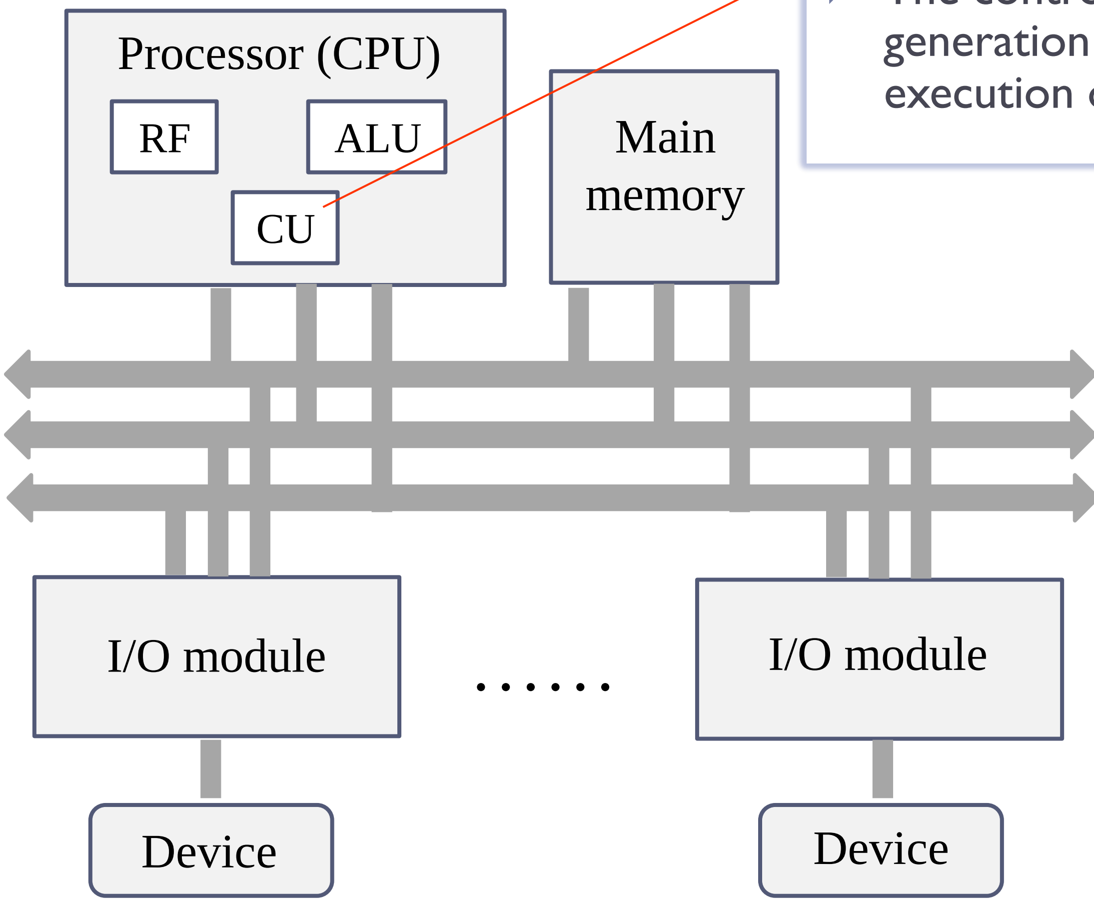
- Registers: store a sequence of bits.
- Two special registers:
  - The PC register (program counter) contains the address of the next instruction to be executed.
  - The RI register (instruction register) stores the instruction being executed.

Processor: Unidad aritmético lógica ALU



- Perform elementary operations on data:
  - Arithmetic
  - Logical

Processor: Unidad de control, UC



- The control unit is responsible for generation of control signals for the execution of instructions.

Contents

Program

- Consecutive sequence of machine instructions

```
00001001110001101010101111101011000
10101111010110000000100111000110
11000110101010111101011000000001001
01011000000001001110001101010101111
```

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

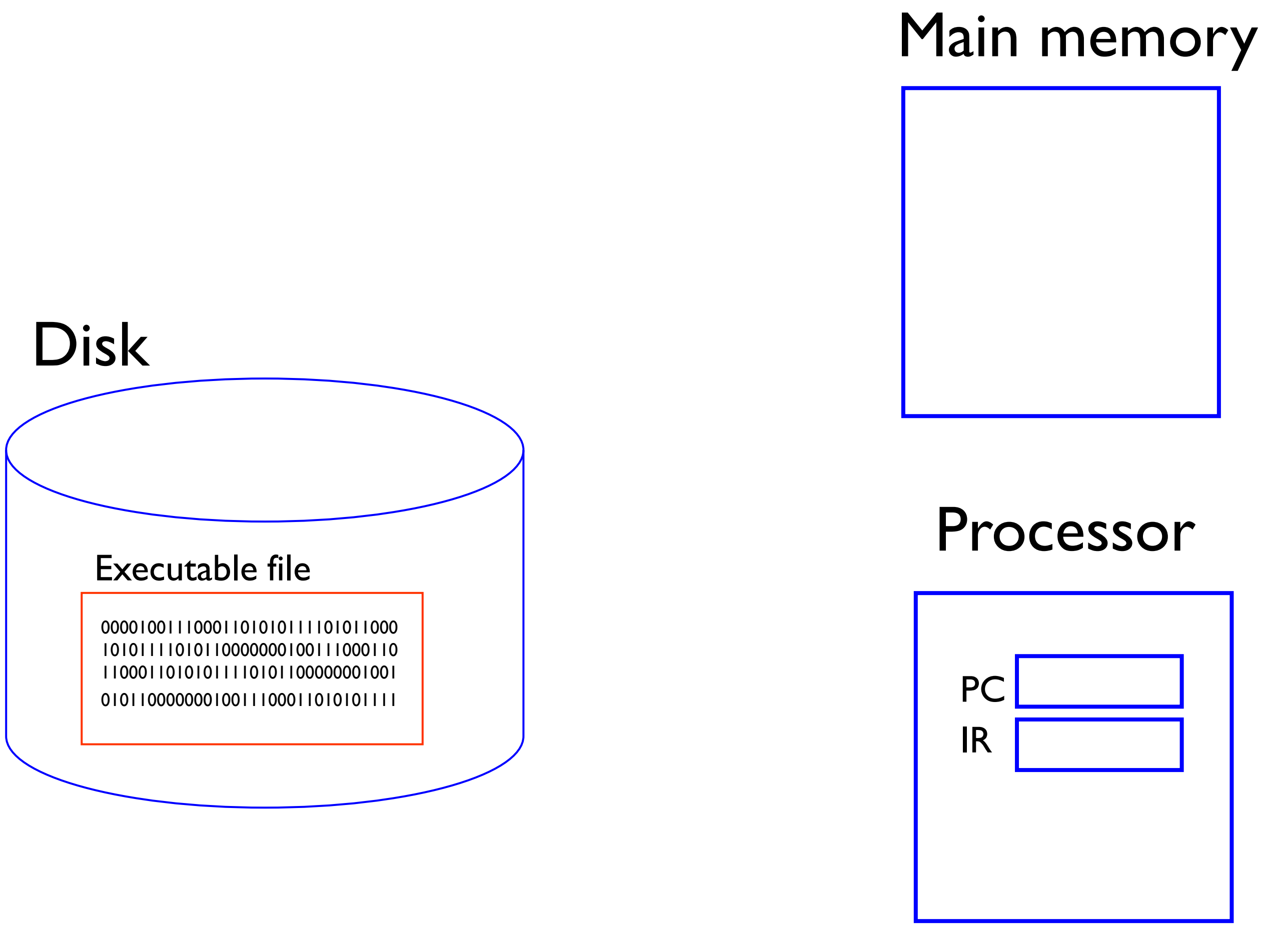
Program

- Consecutive sequence of machine instructions
- Machine instruction: elementary operation that can be executed directly by a processor.
  - Binary coding

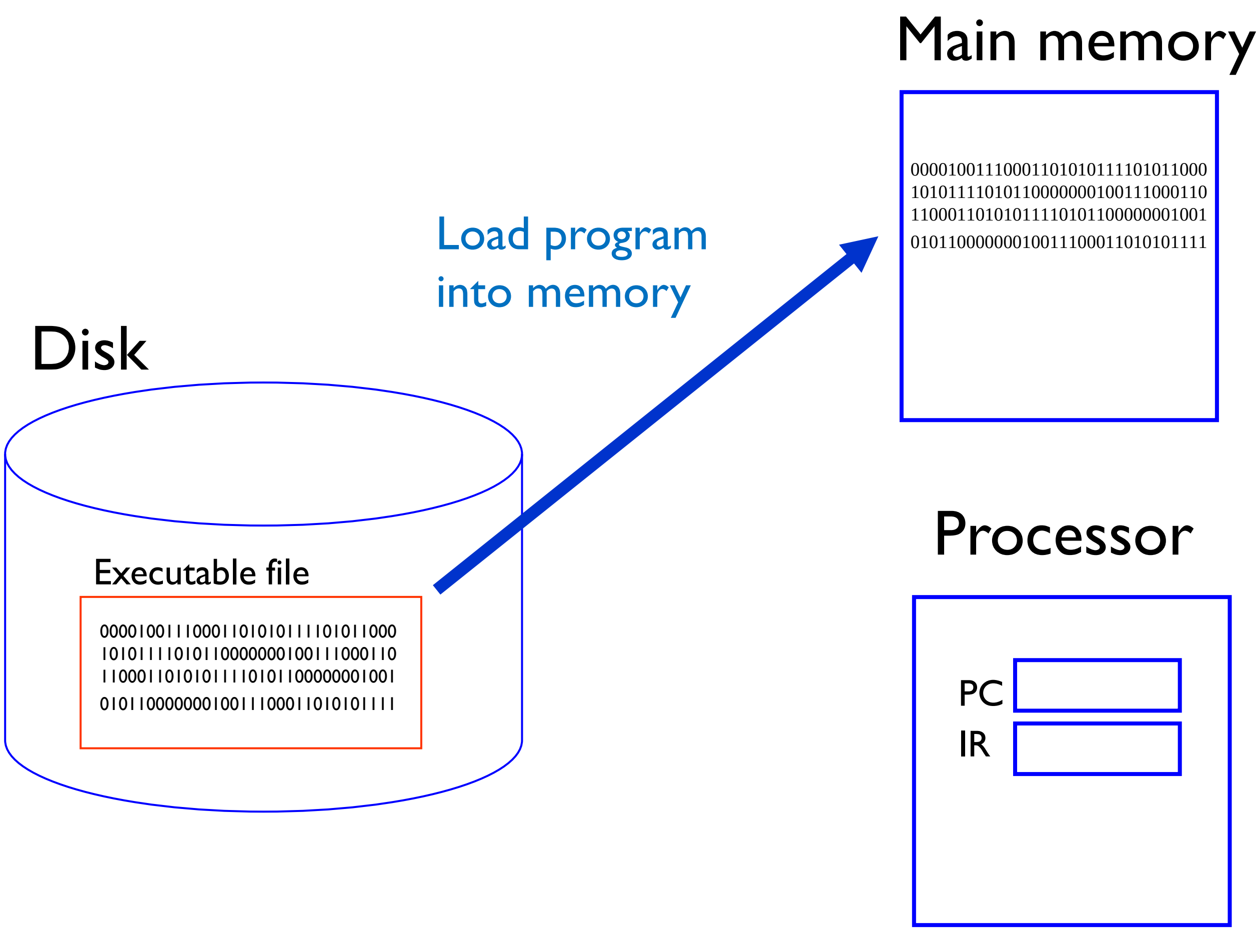
```
00001001110001101010111101011000
10101111010110000000100111000110
11000110101011110101100000001001
01011000000010011100011010101111
```

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

Program execution

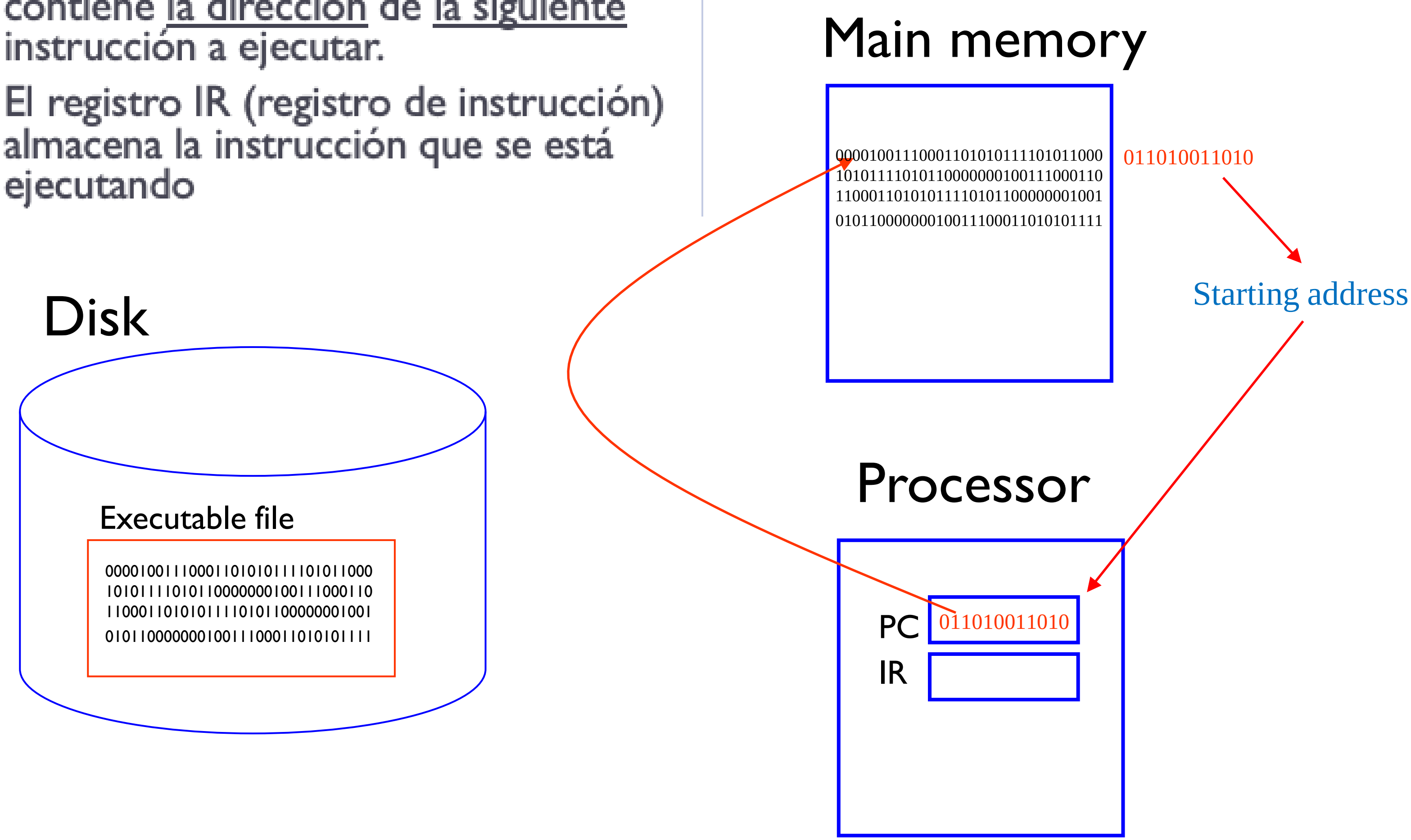


Program execution



Program execution

- El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.
- El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando

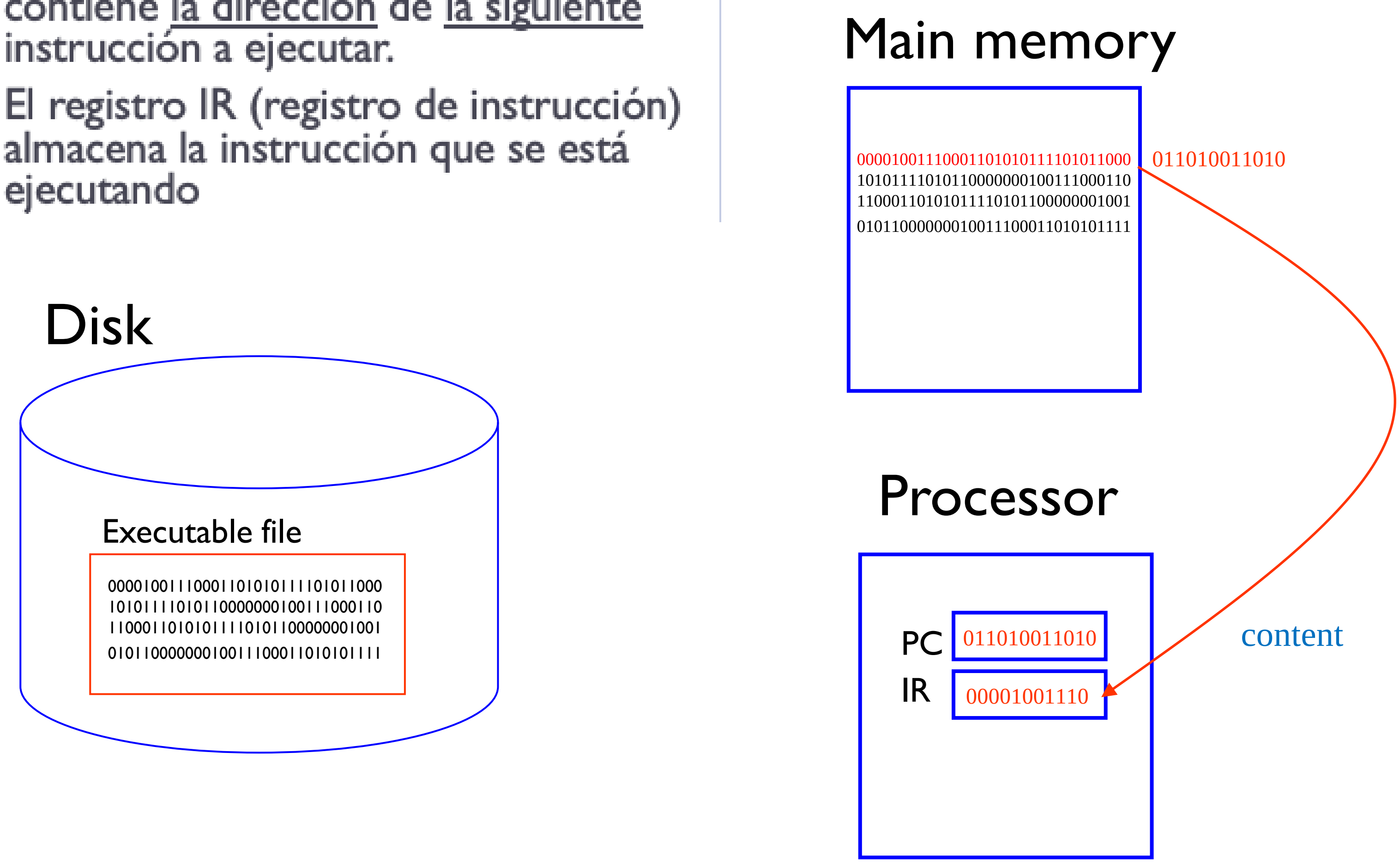


Félix García-Carballeira, Alejandro Calderón Mateos

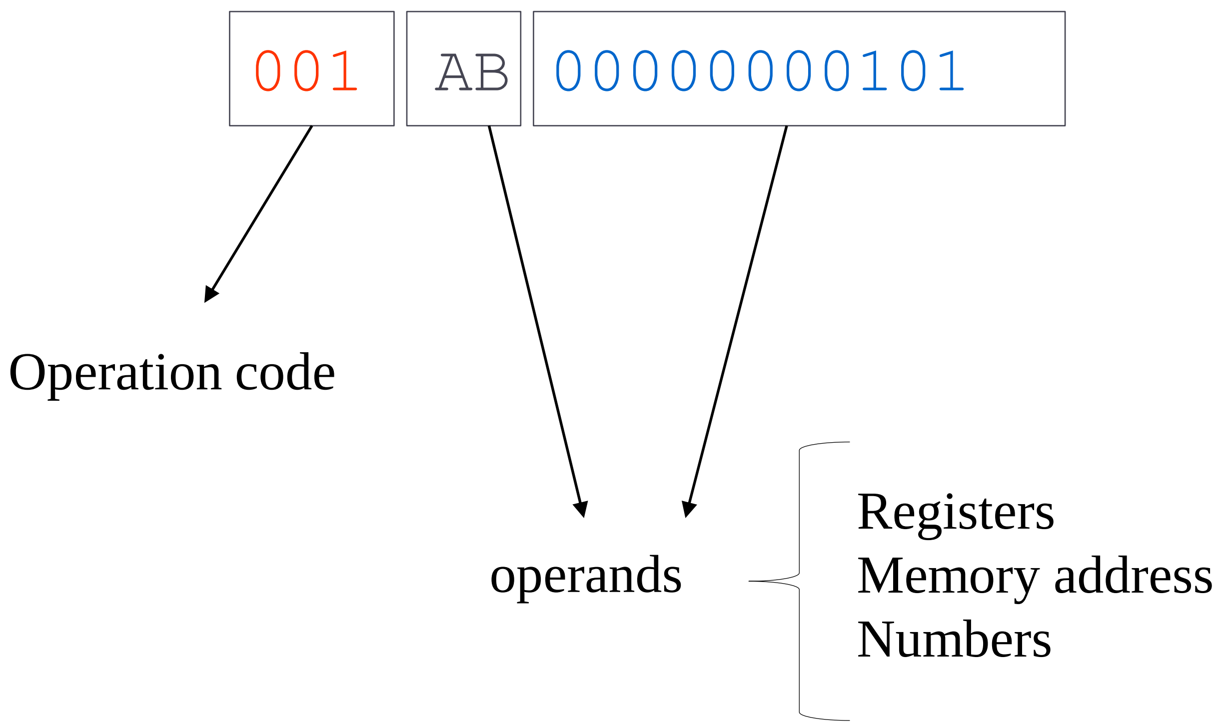
Program execution

El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.

El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando



Format of a machine instruction



Program generation and loading

```
i=0;
s = 0;
while (i < 4)
{
   s = s + 1;
   i = i + 1;
}
```

```
       li R0, 0
       li R1, 4
       li R2, 1
       li R3, 0
loop:  beq R0, R1, end
       add R3, R3, R2
       add R0, R0, R2
       b loop
end:   sw R3, 100000
```

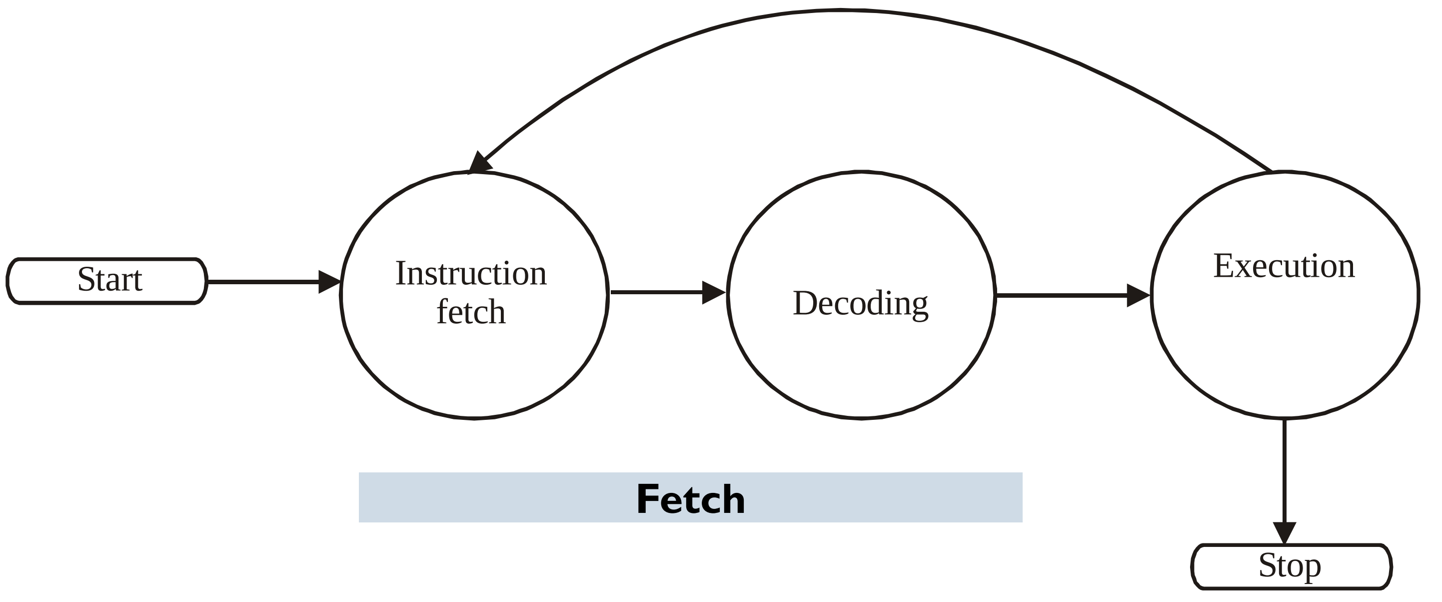
Main memory

000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

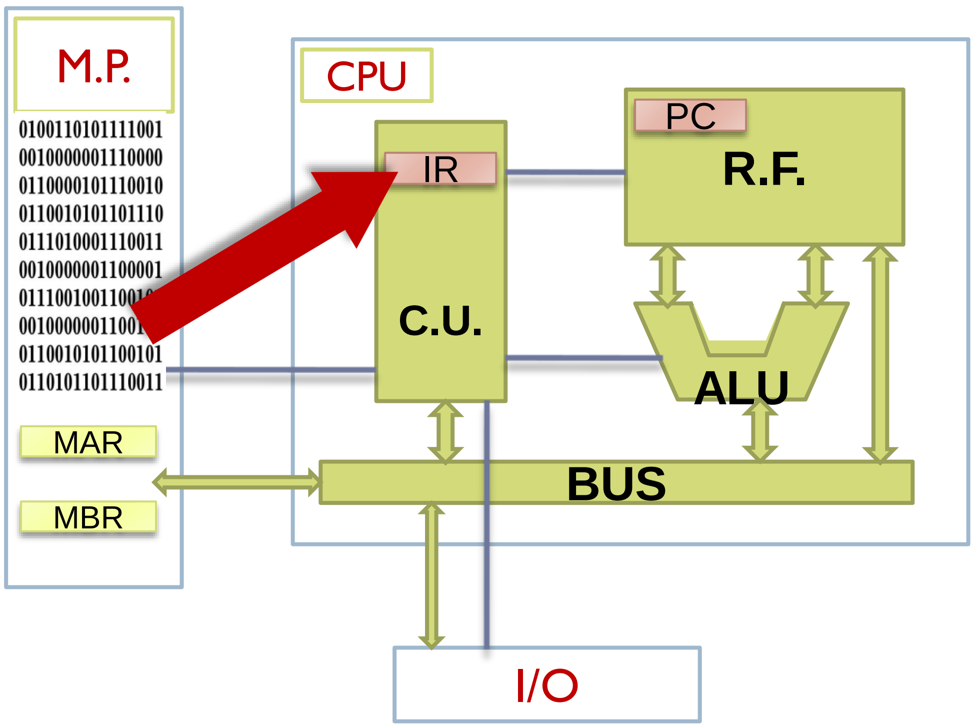
Contents

- Introduction:
  - What is a computer?
  - Building blocks for a computer
  - Concepts of structure and architecture
- **Von Neumann computer:**
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - **Phases in the execution of an instruction**
- Characteristics of a computer and types:
  - Main characteristic parameters of a computer
  - Types of computers
  - Historic evolution

Phases in the execution of an instruction

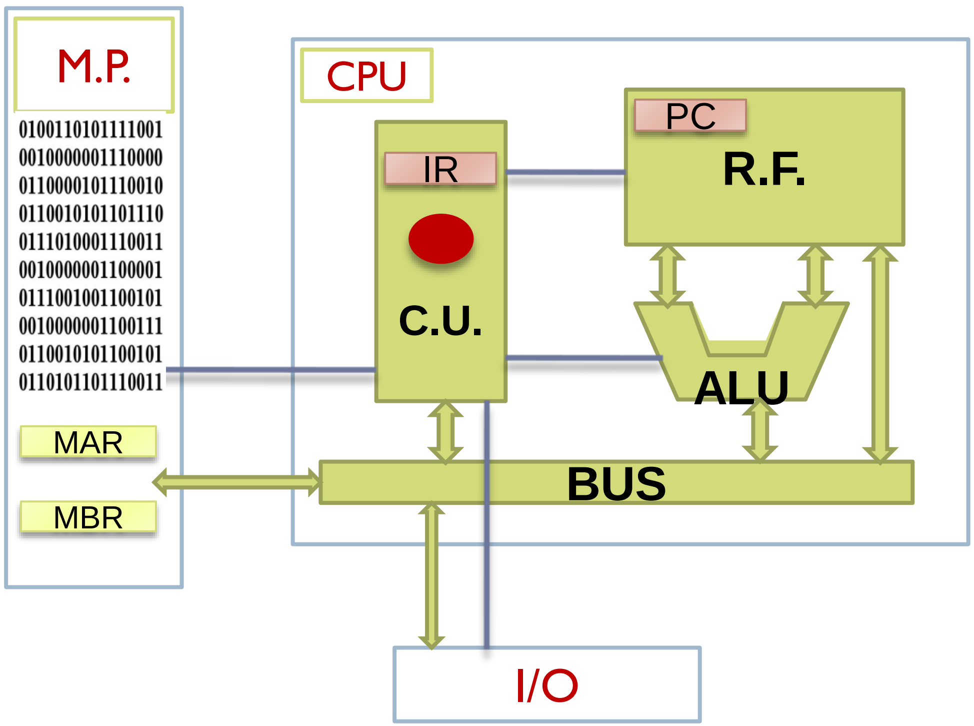


Phases: Instruction fetch



- Read from Main memory the instruction pointed by the PC
  - The PC contains the memory address where the instruction to be executed is stored.
  - The instruction read from M.M. is stored in IR.
- Increment PC
  - Increment the address stored in the PC so that it points to the next instruction
- Decode instruction
- Execute instruction

Phases: Decode

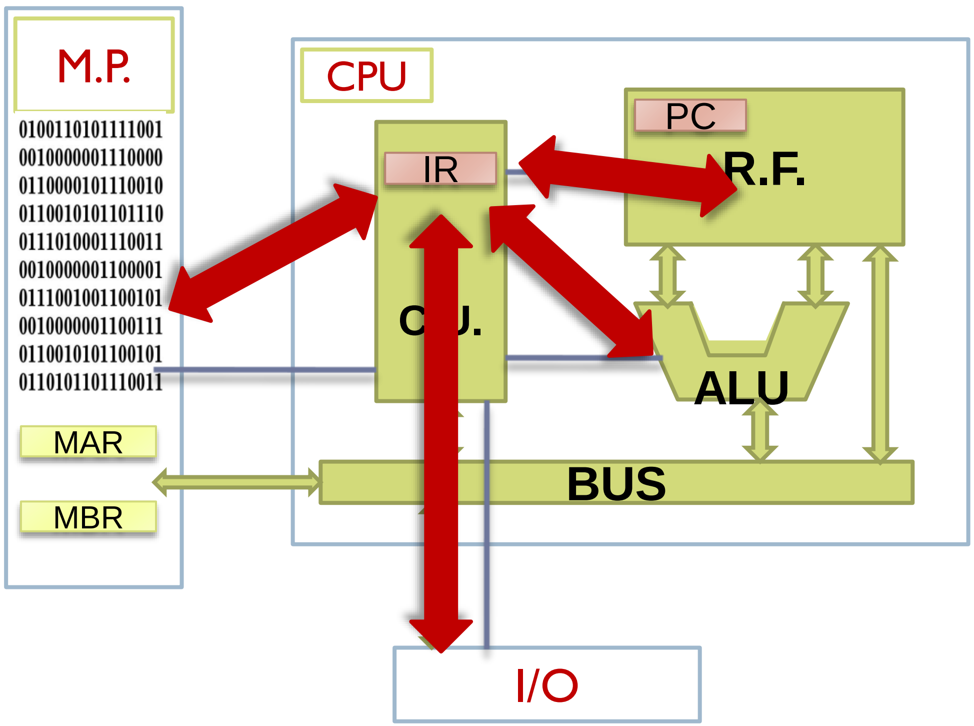


El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.

El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando

- Read from Main memory the instruction pointed by the PC
  - The PC contains the memory address where the instruction to be executed is stored.
  - The instruction read from M.M. is stored in IR.
- Increment PC
  - Increment the address stored in the PC so that it points to the next instruction
- **Decode instruction**
- Execute instruction

Phases: Execution



- Read from Main memory the instruction pointed by the PC
  - The PC contains the memory address where the instruction to be executed is stored.
  - The instruction read from M.M. is stored in IR.
- Increment PC
  - Increment the address stored in the PC so that it points to the next instruction
- Decode instruction
- **Execute instruction**

Example of program execution

Processor

PC	000100
RI	?
00	?
01	?
10	?
11	?

- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111000000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Example of program execution

Processor

PC	000100
RI	0010000000000000
00	?
01	?
10	?
11	?

- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Félix García-Carballeira, Alejandro Calderón Mateos

Example of program execution

Processor

PC	000101
RI	0010000000000000
00	?
01	?
10	?
11	?

- Instruction fetch
- Point to the next instruction
  - PC ⟵ PC + “1”
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Example of program execution

Processor

PC	000101
RI	0010000000000000
00	?
01	?
10	?
11	?

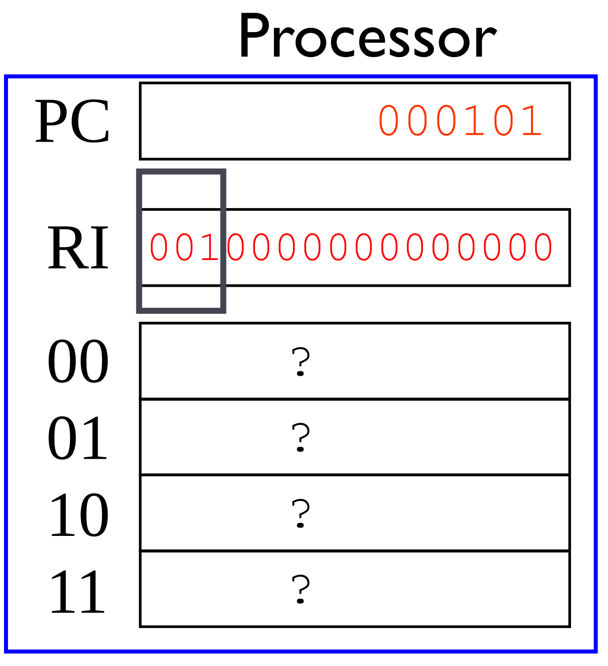
- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

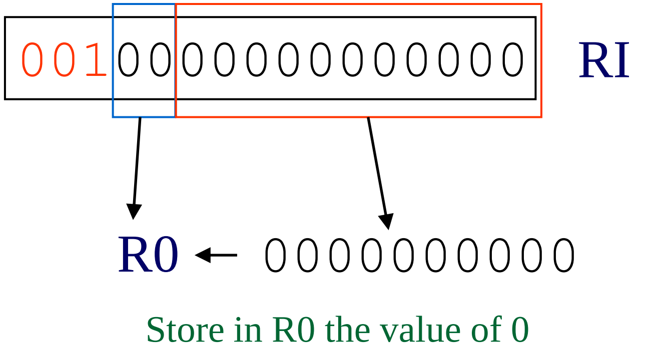
Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Félix García-Carballeira, Alejandro Calderón Mateos

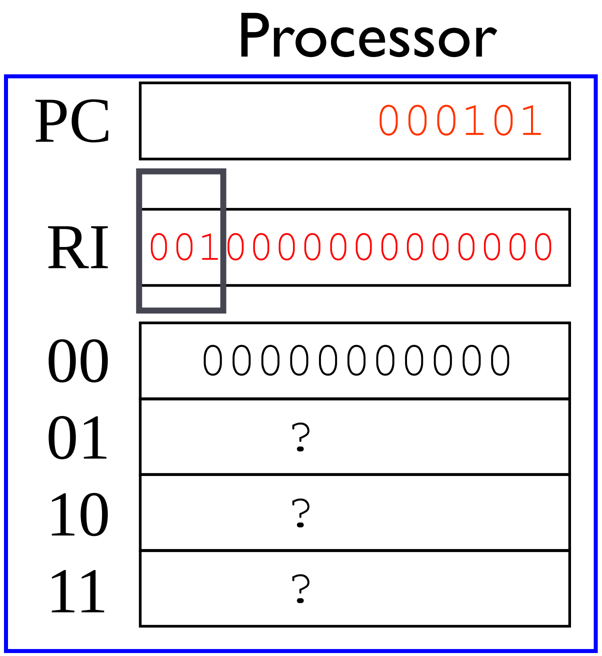
Example of program execution

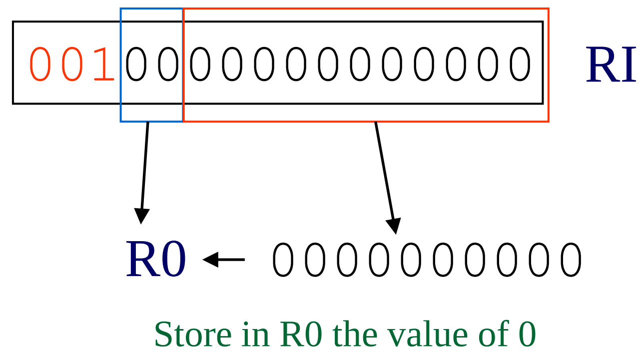




- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Example of program execution





- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Example of program execution

Processor

PC	000101
RI	0010000000000000
00	00000000000
01	?
10	?
11	?

- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Félix García-Carballeira, Alejandro Calderón Mateos

Example of program execution

Processor

PC	000101
RI	0010100000000100
00	000000000000
01	?
10	?
11	?

- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Example of program execution

Processor

PC	000110
RI	0010100000000100
00	00000000000
01	?
10	?
11	?

- Instruction fetch
- Point to the next instruction
  - PC ⟵ PC + "1"
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Example of program execution

Processor

PC	000110
RI	0010100000000100
00	00000000000
01	?
10	?
11	?

Main memory

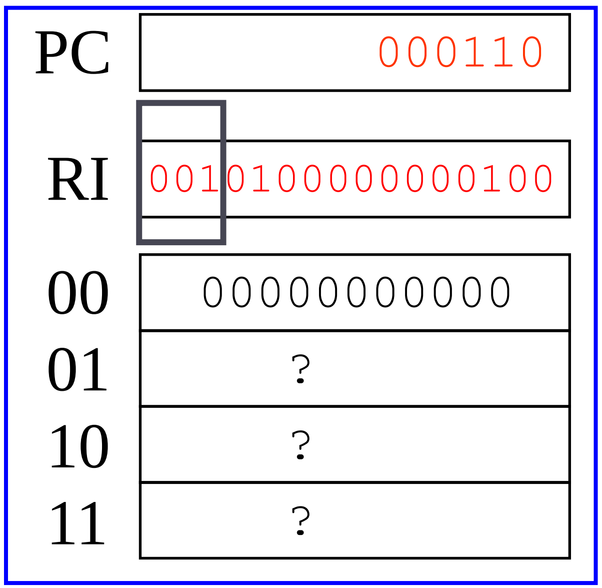
Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

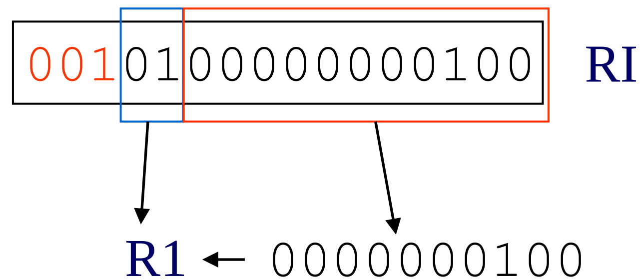
- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Félix García-Carballeira, Alejandro Calderón Mateos

Example of program execution

Processor



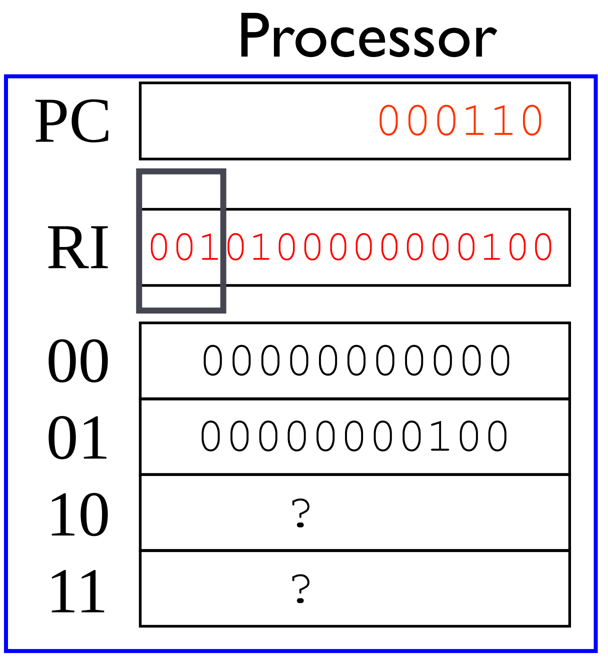


Store in R1 the value of 4

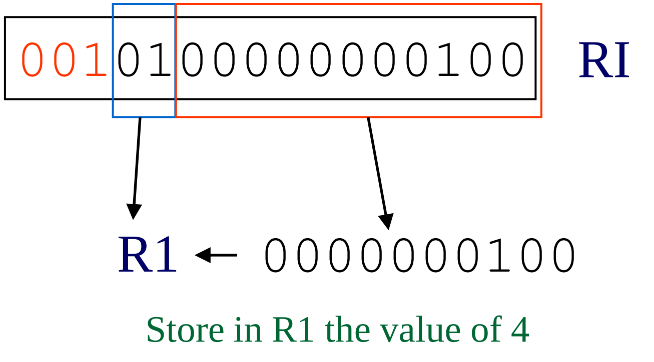
- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Félix García-Carballeira, Alejandro Calderón Mateos

Example of program execution



- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch



Example of program execution

Processor

PC	000110
RI	0010100000000100
00	00000000000
01	00000000100
10	?
11	?

- Instruction fetch
- Point to the next instruction
- Instruction decoding
- Instruction execution
- Jump to fetch

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Félix García-Carballeira, Alejandro Calderón Mateos

Example of program execution

Processor

PC	000110
RI	0010100000000100
00	00000000000
01	00000000100
10	?
11	?

Main memory

Address	Content
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

- And so on…

Contents

- Introduction:
  - What is a computer?
  - Building blocks for a computer
  - Concepts of structure and architecture
- Von Neumann computer:
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- **Characteristics of a computer and types:**
  - **Main characteristic parameters of a computer**
  - Types of computers
  - Historic evolution

Main characteristic parameters of a computer

- Regarding its architecture
  - Word and word size
- Storage
  - Size
  - Storage units
- Communications
  - Bandwidth
  - Latency
- Computer power
  - MIPS
  - MFLOPS

Word Width

- Number of bits handled in parallel inside the computer.
  - Influences the size of the registers (BR).
  - Therefore, also in the ALU
    - Two 32-bit sums are not the same as one 64-bit sum.
  - Therefore, also on the width of the buses.
    - A 32-bit address bus 'only' addresses 4 GB
- A computer with a word width of n bits:
  - n-bit memory addresses
  - Registers store n bits
  - n-bit integers
- Typical sizes ➔ 32 bits, 64 bits

Exercise

- Consider a hypothetical computer with a word width of 20 bits with 60 registers that addresses memory by bytes.

  Please answer the following questions:

  a) How many bits are used for memory addresses?

  b) What is the size of the registers?

  c) How many bits are stored in each memory location?

  d) How many memory locations can be addressed? Express the result in KB.

  e) How many bits are needed to identify the registers?

Privileged sizes

- Word
  - Information handled in parallel inside the processor.
  - Typically, 32/64 bits
- Half word
- Double word

- Octet, character or byte
  - Representation of a character
  - Typically, 8 bits

Unidades para tamaño

- Normalmente se expresa en octetos o bytes:

Name	Binary prefix	IS prefix
Kilo	2^{10} = 1,024	10^{3} = 1,000
Mega	2^{20} = 1,048,576	10^{6} = 1,000,000
Giga	2^{30} = 1,073,741,824	10^{9} = 1,000,000,000
Tera	2^{40} = 1,099,511,627,776	10^{12} = 1,000,000,000,000
Peta	2^{50} = 1,125,899,906,842,624	10^{15} = 1,000,000,000,000,000
Exa	2^{60} = 1,152,921,504,606,846,976	10^{18} = 1,000,000,000,000,000,000
Zetta	2^{70} = 1,180,591,620,717,411,303,424	10^{21} = 1,000,000,000,000,000,000,000
Yotta	2^{80} = 1,208,925,819,614,629,174,706,176	10^{24} = 1,000,000,000,000,000,000,000,000

Name	Binary prefix	IS prefix
Kilo	kibibyte (KiB)	kilobyte (kB)
Mega	mebibyte (MiB)	megabyte (MB)
Giga	gibibyte (GiB)	gigabyte (GB)
Tera	tebibyte (TiB)	terabyte (TB)
Peta	pebibyte (PiB)	perabyte (PB)
Exa	exbibyte (EiB)	exabyte (EB)
Zetta	zebibyte (ZiB)	zettabyte (ZB)
Yotta	yobibyte (YiB)	yottabyte (YB)

https://es.wikipedia.org/wiki/Prefijos_del_Sistema_Internacional

Units for size

- In communication, powers of 10 are used:
  - 1 K**b** = 1000 **bits**
  - 1 K**B** = 1000 **bytes**
- In storage, some manufacturers do not use powers of two, but powers of 10:
  - kilobyte 1 KB = 1.000 bytes 10^3 bytes
  - megabyte 1 MB = 1.000 KB 10^6 bytes
  - gigabyte 1 GB = 1.000 MB 10^9 bytes
  - terabyte 1 TB = 1.000 GB 10^{12} bytes
  - …..

Félix García-Carballeira, Alejandro Calderón Mateos

Exercise

- How many bytes does a 200 GB hard disk have?
- How many bytes per second does my 20 Mb ADSL transmit?

Exercise (solution)

- How many bytes does a 200 GB hard disk have?
  - 200 GB = 200 * 10^9 bytes = 186.26 Gigabytes

- How many bytes per second does my 20 Mb ADSL transmit?
  - B $\longrightarrow$ Byte
  - b $\longrightarrow$ bit
  - 20 Mb/s = 20 * 10^6 bits/s = (20 * 10^6) / 8 bytes/s
    = 2.38 MegaBytes per second

Félix García-Carballeira, Alejandro Calderón Mateos

Bandwidth

- Several interpretations:
  - Information throughput transmitted by a bus.
  - Information throughput transmitted by an I/O unit.
  - Information throughput that can be processed by a unit.
  - Number of bits transferred per unit of time.
- Unit:
  - Kb/s (Kilobits per second,   not to be confused with KB/s)
  - Mb/s (Megabits per second, <u>not</u> megabytes per second)

Félix García-Carballeira,   Alejandro Calderón Mateos

Latency

- Various interpretations:
  - Elapsed time in issuing a request in a reliable messaging system.
  - Elapsed time between the issuance of a request and the performance of the associated action.
  - Elapsed time between the issuance of a request and the receipt of the response.
- Unit:
  - s. (seconds)

Computing power

- Measurement of computing power.
- Factors involved:
  - Instruction set.
  - CPU clock (1 GHz vs 2 GHz vs 4 GHz...)
  - Number of 'cores' (quadcore vs dualcore vs...)
  - Word width (32 bits vs 64 bits vs...)
- Typical ways of expressing computational power:
  - MIPS
  - MFLOPS
  - …

MIPS

- Millions of Instructions Per Second.
- Typical range: 10-100 MIPS
- Not all instructions take the same amount of time to execute Depends on which instructions are executed.
- Not 100% reliable as a measure of performance.

Félix García-Carballeira, Alejandro Calderón Mateos

MFLOPS

- Millions of Floating-Point Operations per Second.
- Scientific computing power.
- MFLOPS < MIPS
  - Floating operation more complex than normal operation
- Vector Computers: MFLOPS > MIPS
- Example: Itanium 2 ➔ 3,5 GFLOPS

Vectors per second

- Computing power in graphics generation.
- Applicable to graphics processors.
- Can be measured in:
  - 2D vectors.
  - 3D vectors.
- Example: ATI Radeon 8500 → 3 Million.

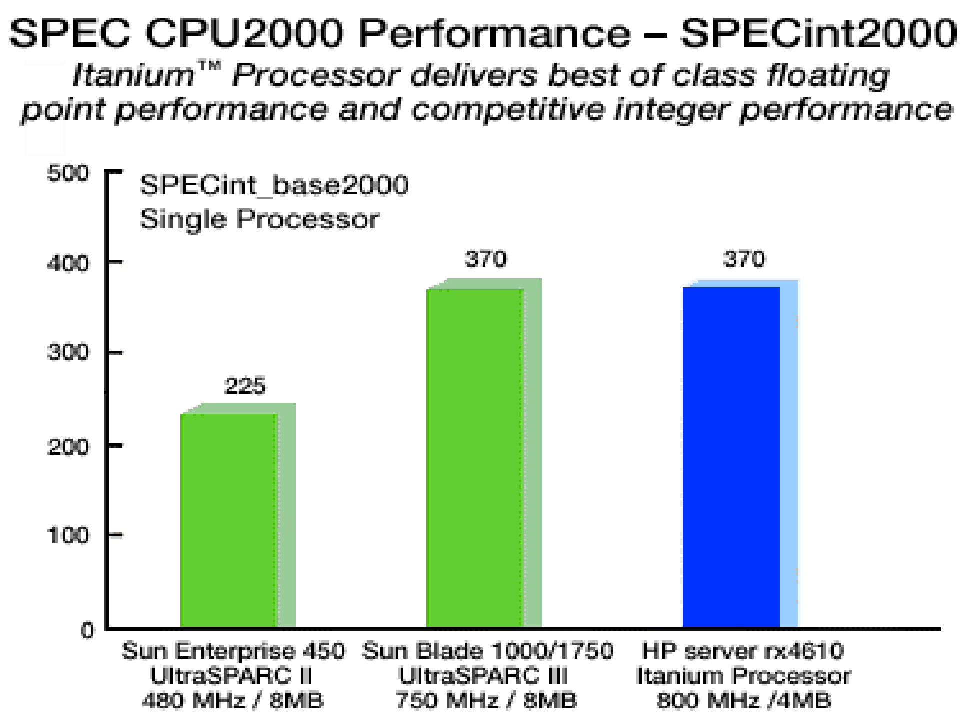
Synthetic tests

- **MIPS** and **MFLOPS** not valid for comparing different machines.
- Tests based on running the same program on different machines to compare them.
- They measure effectiveness Compiler + CPU

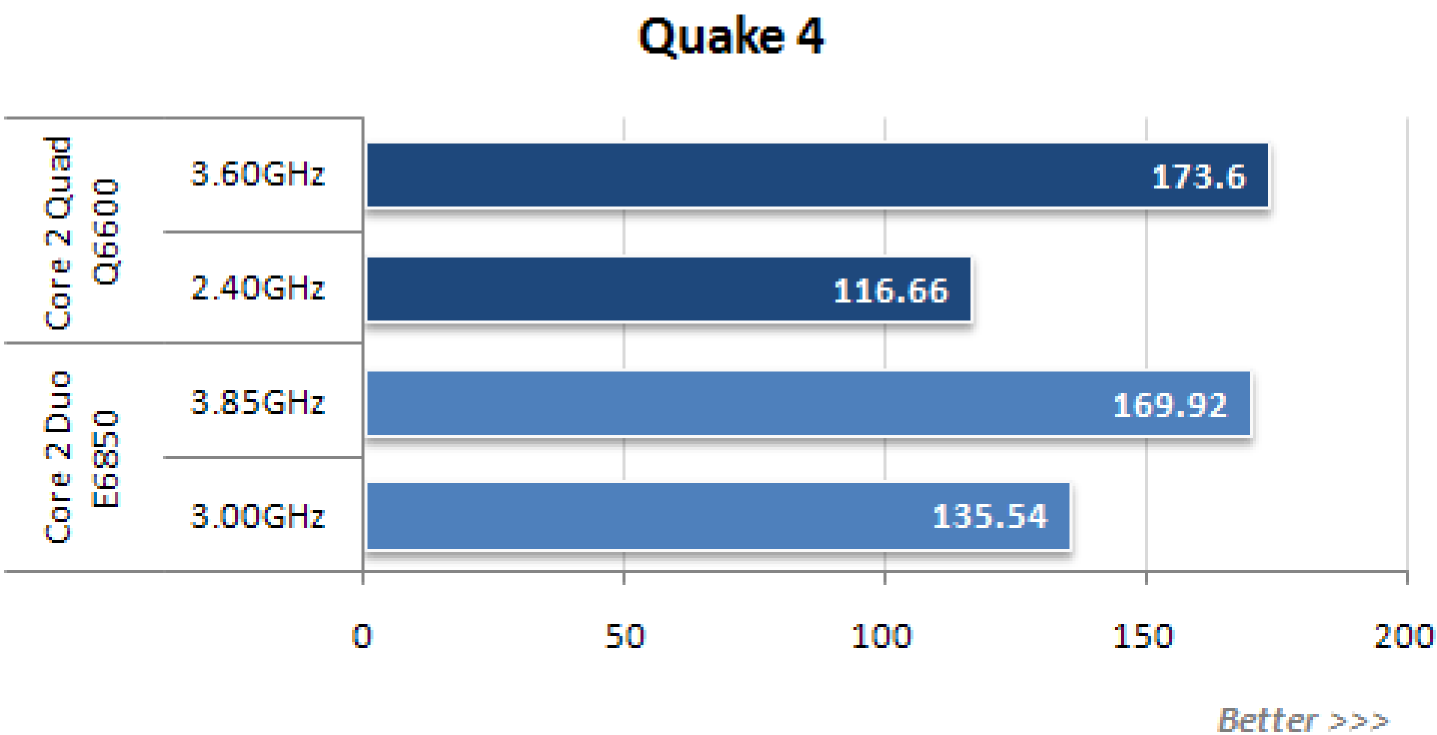
- Standardized ("official") synthetic tests seek to compare the power of two computers.
- It is possible to use "unofficial" synthetic tests to get an idea of the improvement with daily workload.

“Official" synthetic tests

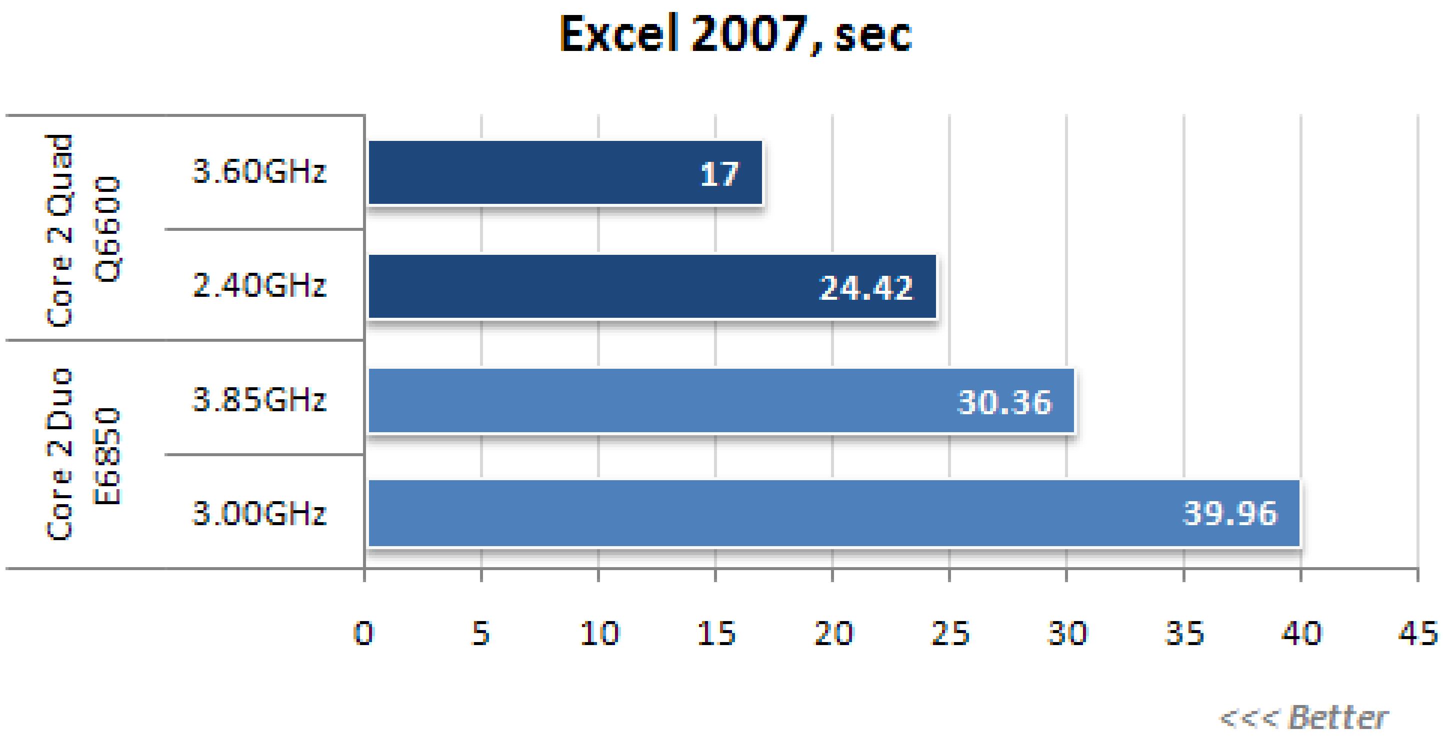
- Most frequently used tests:
  - Linpack.
  - SPEC.



“Non-official" synthetic tests



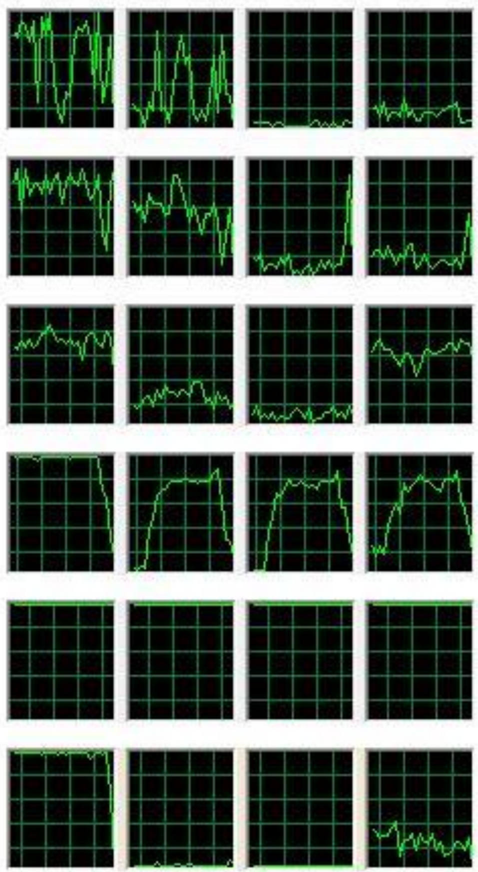
“Non-official" synthetic tests



http://www.xbitlabs.com/articles/cpu/display/core2quad-q6600_11.html

“Non-official" synthetic tests

Task Manager CPU Graph



	improvement 2 to 4 cores
The Elder Scrolls IV: Oblivion	none
Rainbow 6: Vegas	none
Supreme Commander	none
Valve Source engine particle simulation	1.8 x
Valve VRAD map compilation	1.9 x
3DMark06: Return to Proxycon	none

http://www.codinghorror.com/blog/archives/000942.html

Contents

- Introduction:
  - What is a computer?
  - Building blocks for a computer
  - Concepts of structure and architecture
- Von Neumann computer:
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- **Characteristics of a computer and types:**
  - Main characteristic parameters of a computer
  - **Types of computers**
  - Historic evolution

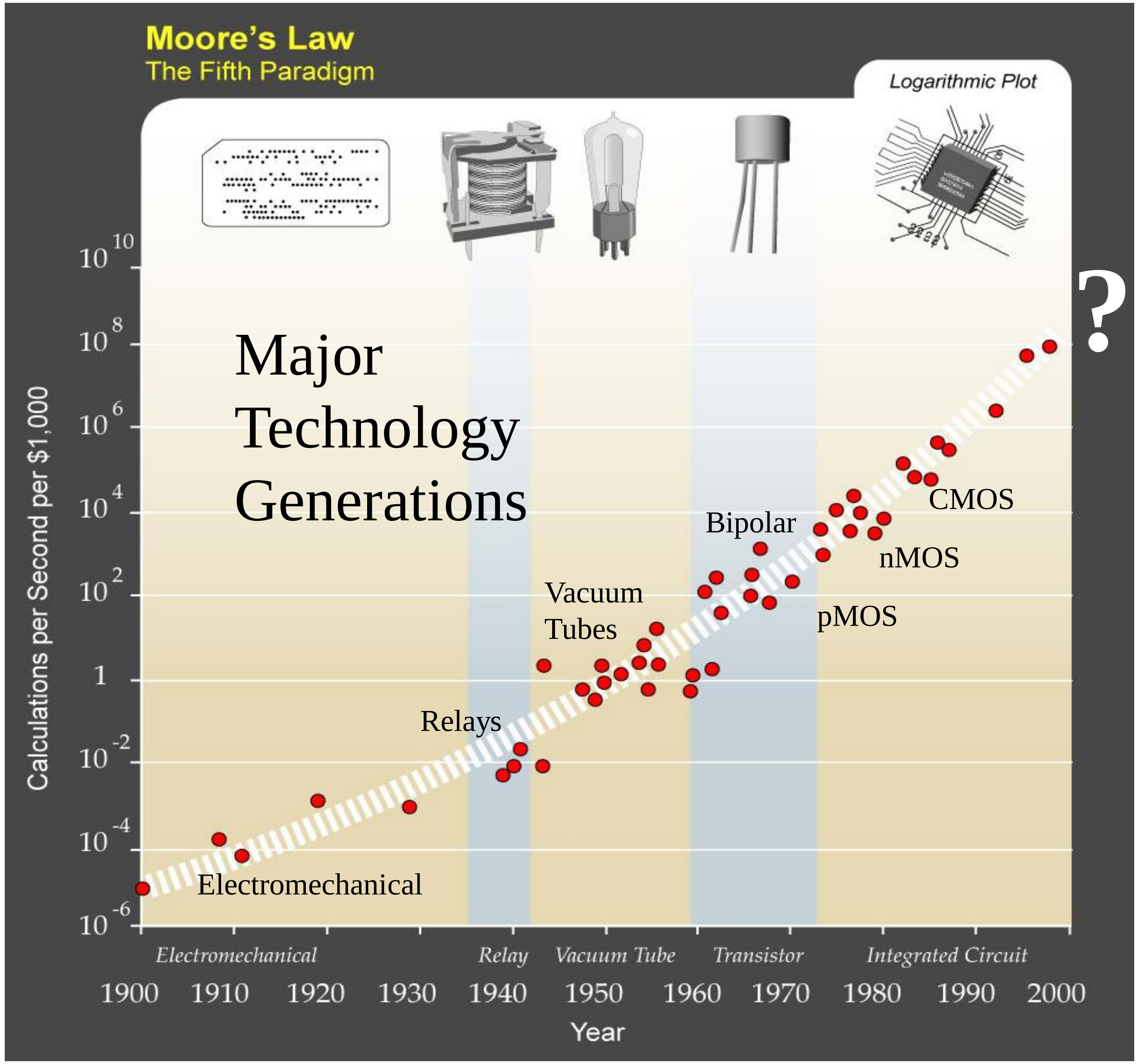
Types of computers

Name	Goals	Examples	Design aspects
Desktop	Designed to deliver good performance to users	Currently, most of them are portable	• Price-performance ratio • Power • Graphics performance
Personal mobile devices	Wireless devices with multimedia user interface	Smartphones, tablets,…	• Price • Energy • Performance • Response time
Servers	Used to run high performance or scale applications	Serve multiple users simultaneously	• Throughput (Processing rate) • Availability • Reliability • Energy • Scalability
Clusters	A set of computers connected by a network that acts as a single, higher performance computer	Used in supercomputers and large data centers	• Price-performance • Throughput (Processing rate) • Availability • Reliability • Energy • Scalability
Embedded	Computer inside another system to control its operation	Lavadoras, TV, MP3, consolas de videojuegos, etc.	• Price • Energy • Application specific performance

Contents

- Introduction:
  - What is a computer?
  - Building blocks for a computer
  - Concepts of structure and architecture
- Von Neumann computer:
  - Von Newmann architecture
  - Machine instructions and assembly programming
  - Phases in the execution of an instruction
- **Characteristics of a computer and types:**
  - Main characteristic parameters of a computer
  - Types of computers
  - **Historic evolution**

Main technological generations



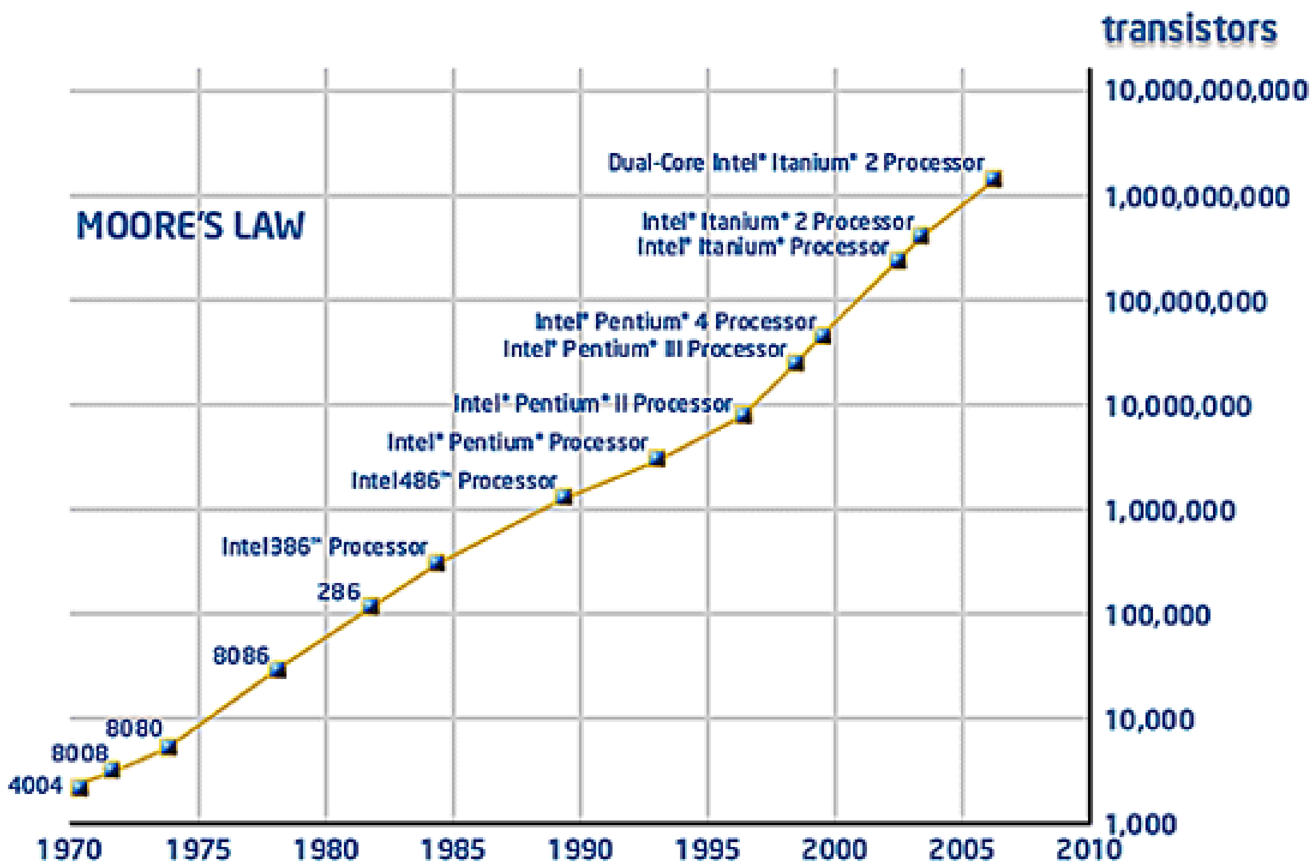
[from Kurzweil]

https://cse.buffalo.edu/~stevko/courses/cse490/spring11/schedule.html

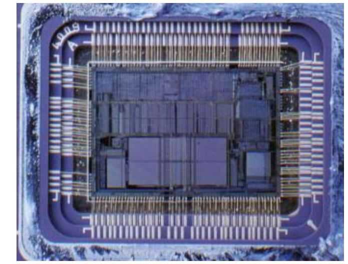
Microprocessor

- A **microprocessor** incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit

Moore’s law


transistors
10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
MOORE'S LAW
Dual-Core Intel® Itanium® 2 Processor
Intel® Itanium® 2 Processor
Intel® Itanium® Processor
Intel® Pentium® 4 Processor
Intel® Pentium® III Processor
Intel® Pentium® II Processor
Intel® Pentium® Processor
Intel486™ Processor
Intel386™ Processor
286
8086
8080
8008
4004
1970
1975
1980
1985
1990
1995
2000
2005
2010

Moore’s law



- Double the density implies to reduce the dimensions the 30%
- In 1971 the 4004 Intel had 2300 transistors of 10 micrometers
- Nowadays there are microprocessors with less than 30 nanometers
- Moore’s law need technology with a price that double every 4.4 years

Félix García-Carballeira,   Alejandro Calderón Mateos

Technology improvements

- Memory
  - DRAM capacity: 2x / 2 years (since 96);
    64x in the last decade.
- Processor
  - Speed: 2x / 1.5 years (since 85);
    100X in the last decade.
- Disks
  - Capacity: 2x / 1 year (since 97)
    250X in the last decade.

Historic evolution: bibliography

- http://history.sandiego.edu/GEN/recording/computer1.html
- http://www.computerhope.com/history/
- http://www.computerhistory.org/
- http://www.computersciencelab.com/ComputerHistory/History.htm
- Museos de informática
- Search in Google: Computer history

ARCOS Group

uc3m | Universidad Carlos III de Madrid

L1: Introduction to computers

Computer Structure

Bachelor in Computer Science and Engineering

Bachelor in Applied Mathematics and Computing

Dual Bachelor in Computer Science and Engineering and Business Administration