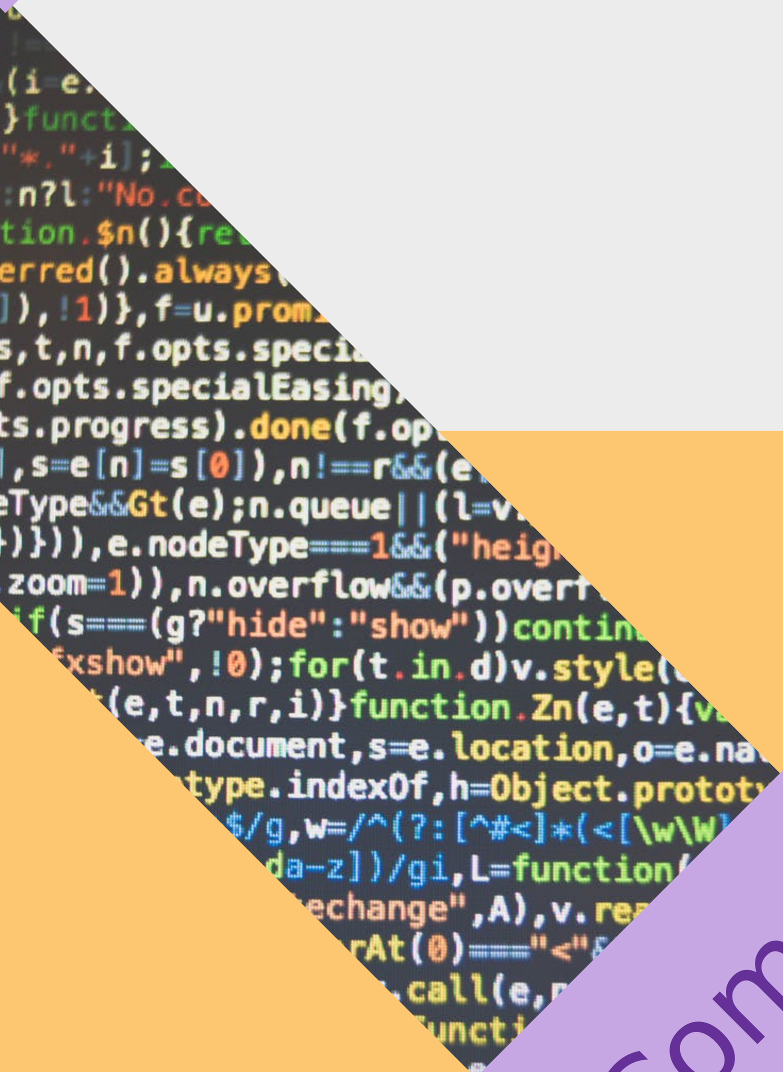
Curs 2020-2021



Computer Systems Functional Units

IES Jaume II El Just
Tavernes de la Valldigna

$12 Introduction bclogocompteur.1$

Contents

1 Introduction

As it is generally known, there have been several generations of computer through history. Thanks to the advances in the field of electronics, computers have evolved to these days. Nowadays, quantum computing is knocking the door but that is a different matter.

The history of computers came from the twentieth century, however there are forerunners as the abacus or some others devices in order to help the man to adding or subtract. One of these devices are the mechanical calculator invented by Blaise Pascal in 1642 but beyond of these machines, Charles Babbage built the first computer which was able to solve polynomials. Also, Babbage invented another programmable device which could be programmed with punched cards.

At the end of the nineteen century, Herman Hollerit invented a mechanical process in order to carry out census of the population in EEUU. The data of each person was introduced in a punched cards. Some years later, Hollerit founded one company which was taken over by other two to establish the International Business Machines Company, better known as IBM.

In the middle of the nineteenth century, electronics appeared and the first computers were built with vacuum tube. Some years later transistor was invented and it meant revolutionary discovery. After that breakthrough, others came as integrated circuits and microprocessor.

Electronics evolved enormously due to the fact that the same size of chip fits more and more transistors, that is to say, the integration rises on according to the time. That is known as Moores law. Moores law is the observation that the number of transistors in a dense integrated circuit doubles about two years. That law has been accomplished during fifty years to nowadays. According to Moore himself, his law is near extinction so Intel and others brands are working in other alternatives such as changing the structure of the chips, others materials different from silicon even quantum computing. It is a matter of time.

Moore's Law

To learn more about the Moore's Law you can watch the following video:

2 What is a computer?

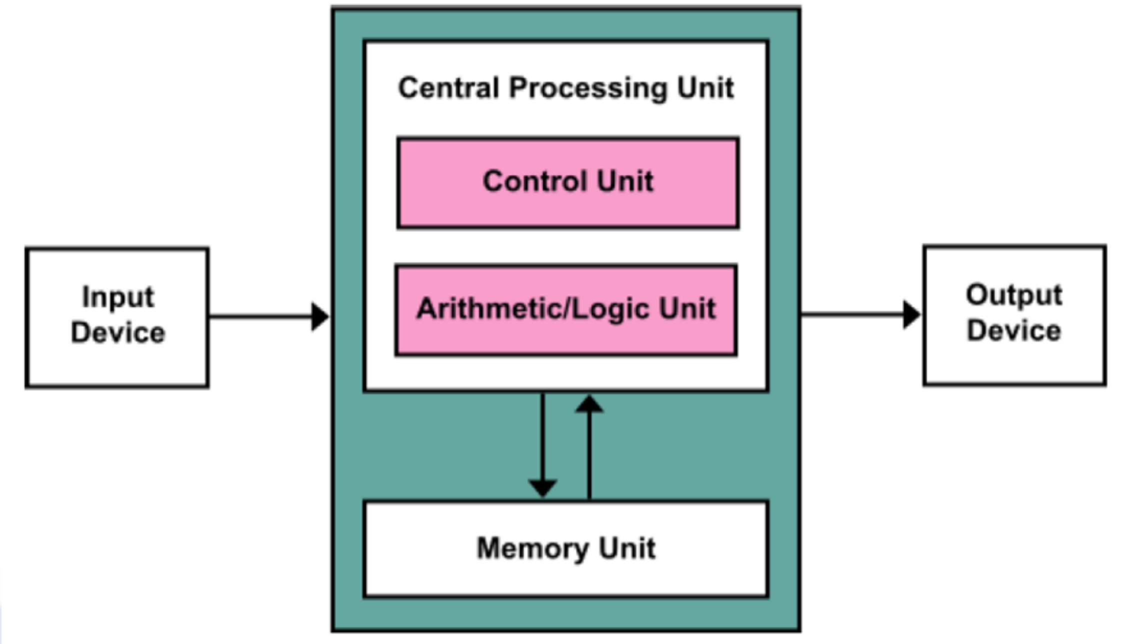
Mankind always have wanted to build devices which help him to solve problems so a computer is a machine for processing information. The process is the different stages and transformations that the information undergoes in order to solve a particular problem.

2.1 Computer architectures

The architecture of a computer defines the functional conduct so the functional units are described, the function of each one and how are related among them.

2.1.1 Von Neumann architecture

The current architecture which is used in computers is Von Neumanns architecture. That architecture was established in 1945.



The main elements of the Von Neumann architecture are:

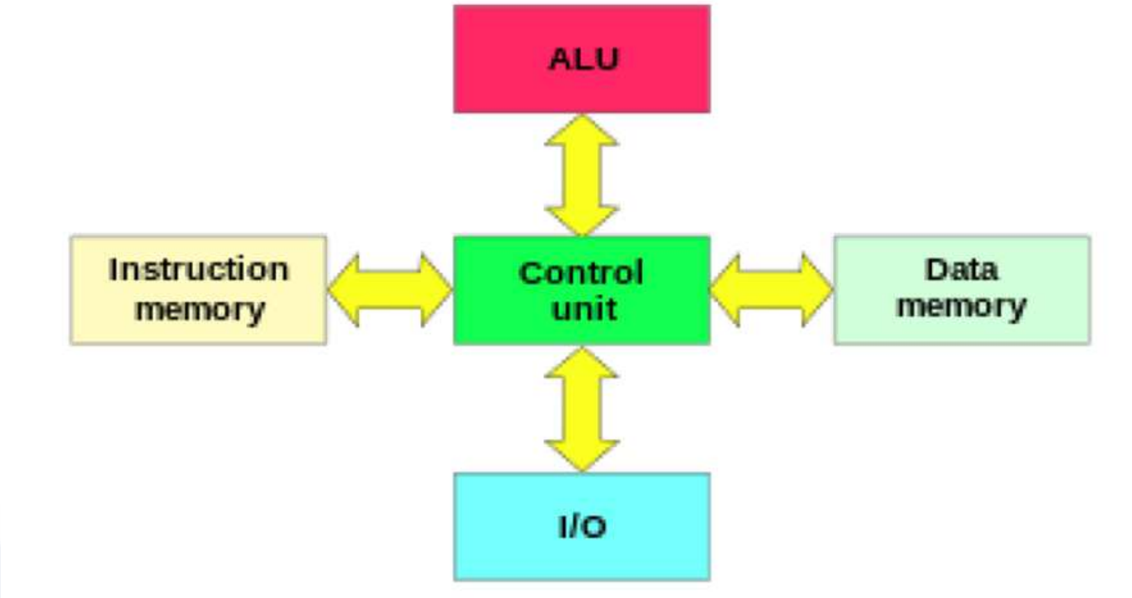
- Memory unit
- CPU
  - Unit Control
  - Arithmetic/Logic Unit
- I/O connexion with external peripherals
- Buses

The main features of that architecture are:

- The memory unit provides the instructions and data to the CPU.
- Instructions and data share the same memory.
- The instructions and data come from programs, input, and output devices.
- When CPU receives some instructions through the buses, the control unit decodes the instruction and enable the control in order to execute the instructions.
- The instructions can use the arithmetic/logic unit of the CPU
- The result of the instructions process can be stored in a memory device or sent to external devices through the bus.

2.1.2 Harvard architecture

Harvard architecture is less used than Von Neumann architecture and the main difference between both is that in Harvards architecture instructions and data are stored in two separated memories. One of them for data and other for instructions. That involve having two buses for addresses, one to manage data and other for manage instructions. The final goal is able to manage data and instructions at the same time in order to be more efficient.



Both, Harvard and Von Neumann architectures are functional models and they arent implemented completely.

3 Functional units of a computer

3.1 CPU(Central Processing Unit)

It is called Central Processing Unit to the set of control unit and arithmetic/logic unit and there are some others components:

- **Registers**: they are small memories inside the CPU in order to help control unit and Arithmetic unit to carry out their task. They are faster than the main memory since the speed of the execution of the instructions requires it. The most important registers are:
  - Accumulator register (AX): it is used in arithmetic and logical operations and is always one of the operands of the operation in the arithmetic unit
  - General Purpose registers (BX, CX, DX, etc): in order to contain data in a particular time. The main goal of these is similar to the variables in some language of programming.
  - State register (Flags): there is a register with some indicators in order to point out the result of the last operation. There are indicators of carry, zero or sign. When the result of the last operation has been zero. The indicator of zero will point out it or if the result of the last operation is negative, etc
  - Program counter (PC): it is an incremental counter which stores the memory address of the next instruction.
  - Instruction Register (IR): that register is loaded with the instruction which is being executed
  - Memory address register (MAR): it is used for storing the address of a block of memory for Reading from or writing to.

  - Memory data register (MDR): way register that stores data fetched from memory

- **Arithmetic-logic unit**: it is the unit in charge of do the arithmetic (addition, subtraction,) and logic operations (or, and, xor, not.)

- **Internal Buses**: they are the buses to communicate registers, control unit, arithemtic/-logic unit

- **Control unit**: it is in charge of reading the instructions and actívate the signals in order to execute the instruction. The control unit has 2 main components: decoder and clock:

  - Decoder: it s function is to decode the instruction and actívate the signals to transmit the to the arithmetic/logic unit what operation is going to do.
  - Clock: it is a clock generator which generate pulses to synchorinze the instruction.

3.2 Memory Unit

The memory unit is used for storing instructions and data. It is a unit divided by cells. These cells can be identified by an address. Each cell contains just one bit (0 or 1). Depending on the technology which is used, they will be faster or slower. Besides, there are some of them that need energy in order to maintain data. That one is used in the memory unit in the Von Neumann architecture and is normally called RAM (Random Acces Memory).

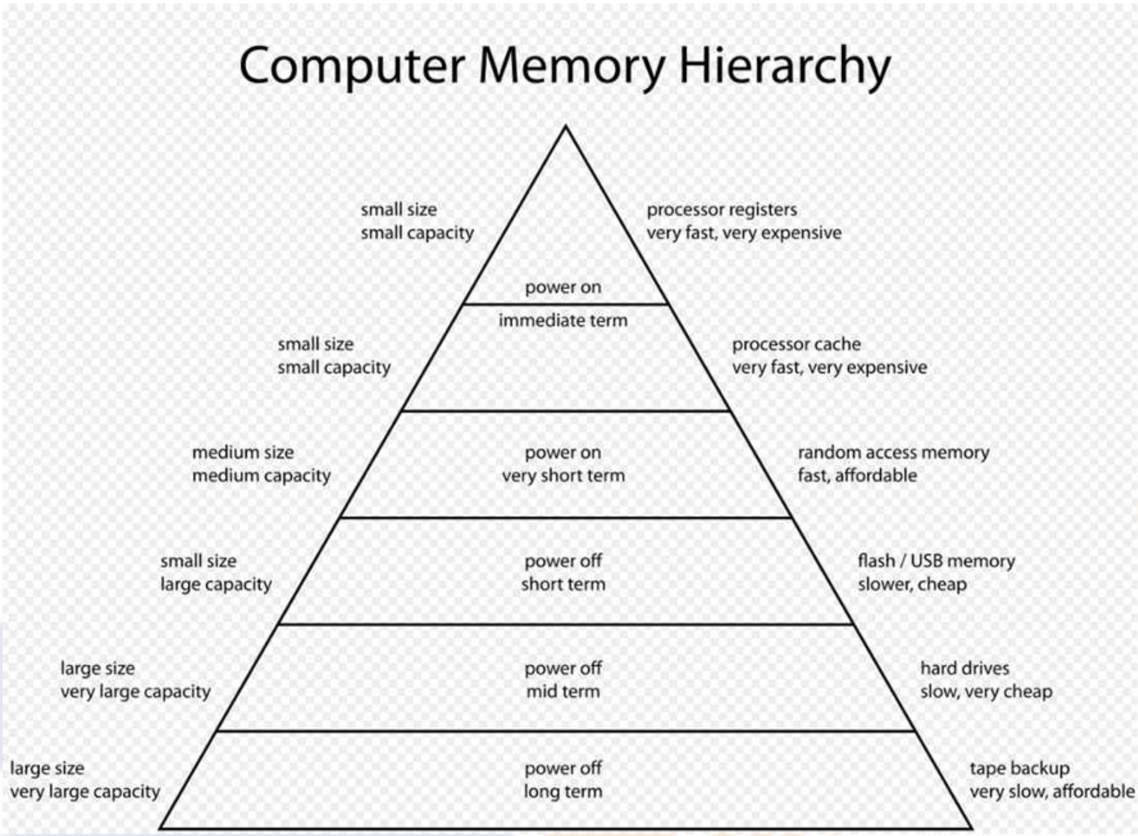
A set of cells is called word and normally computers use words of 32 or 64 bits. The speed of the memory is measured by the numbers of word that is able to read or write in one second. The unit used is Hz.

3.2.1 Kinds of memory

Memory is the element in charge of store data and instructions. The two operations are writing and reading. From the point of view of the utilization, the technology used in order to make the memory has three main properties:

- Cost per byte
- Acces time (latency), how long is required in order to acces the information.
- Capacity

The computer memory hierarchy is shown in the following picture, the faster memory is register processors because they are more expensive and the y have an small size. After registers, it is possible to find cache memory, random access, etc The slower memory is hard drives with larger capacity.



The different kinds of memory in a computer are:

- **ROM (Read Only Memory)**: it is a read only memory and it cant be erased. It is used in old BIOS.
- **EPROM (Erasable Programmable Rea-Only memory)**. It is a type of memory which can be erased with ultraviolet light. It can be written again after of being erased. It is used in BIOS.
- **EEPROM (Electrically Erasable Programmable Read-Only Memory)**. It can be electrically erased and it can be re-written.
- **Flash memory**: is an evolution of EEPROM. It lets to write and read in several memory places at the same time. Usually it is used in USB memories. Nowadays it is the most used technology to store computer BIOS.

3.3 I/O Unit

It carries out the communication with external devices. The external devices are called peripherals and it lets load data, programs and sends results. Each unit of I/O is used to connect one peripheral and it is called interface. There are input only peripherals (keyboard, mouse) or output (printer) but there are input and output devices (hard disk, graphic card, etc)

3.4 Buses

A bus is the way where the information travels from the different units of the computer. The number of bits which can be sent in one operation is the boundary of the bus. Normally it is 32, 64 or 128 bits. The speed of the bus is measured by the number of operations which are performed in a time of unit. One of the units of the bus is in Hz. The buses can be:

- **Serial**: all the information is sent through a wire, bit per bit always following some protocol. Example of serial buses is USB or SATA.
- **Parallel**: The information is sent through several wires. Examples of those buses are PATA (IDE) or parallel port which is not used. The performance of the Serial ports is better than parallel port.

Talking about Von Neumann architecture, it is possible to distinguish between:

- **External**: they are buses in order to connect the different units of the architecture.
- **Internal**: they connect the different internal components of the CPU (Control Unit, Arithmetic/logic unit, registers).

It is possible to find three specialised buses in Von Neumann architecture:

- Control bus
- Address bus
- Data bus

4 Machine code

The machine code is formed by a set of simple instructions simplify the electronic design of it. Those instructions are those which the computer can execute in a direct way.

The set of instructions establishes the capacity of the computer. The simpler set of instructions the more instructions will be needed in order to do some operation. Besides, the number of bits of the instructions is the number of bits of the CPU (16, 32, 64 bits)

Each instruction in machine code is represented in a binary code but in order to make easier and understable the machine code, it exists an equivalent representation called assembly language which is closer to human language. Example of machine code and assembly language:

$$10111000\ 00000101\ 00000000 \Rightarrow \text{MOV AX, 5d}$$

The possible instructions can be:

- Arithmetic and logic operations.
- Control flow operations. That kind of instructions can change the normal flow of the program modifying and causing jumps.
- Data handling and memory operations.

4.1 Instruction cycle

As it is known, instructions are codified in code machine and the information what can be pulled out of them is what kind of operation is going to execute and the data needed in order to execute the instruction. But execution of the instruction is divided in several parts. Those parts are:

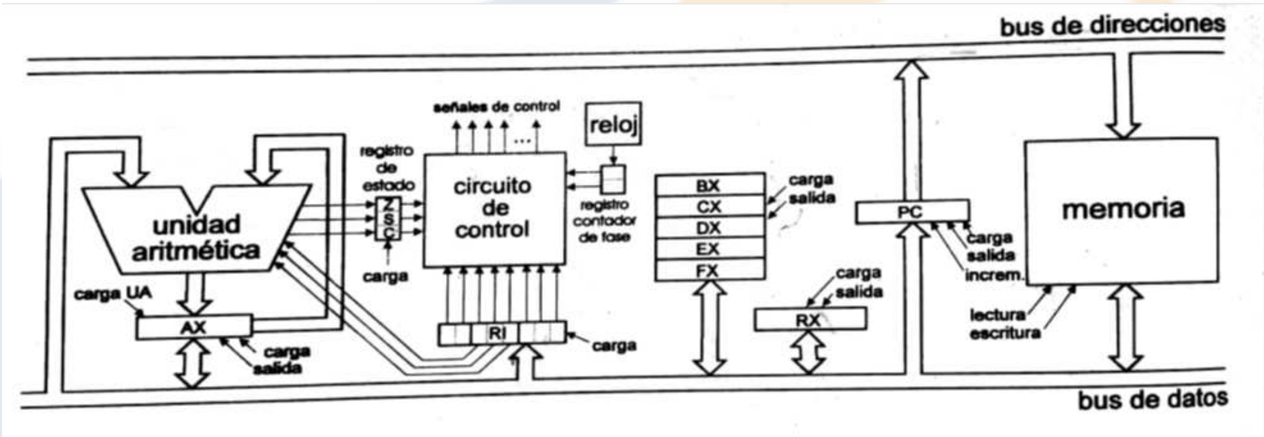
1. **Fetch the instruction**: in that part, instruction is read from memory and is stored in IR register. The address where the instruction is stored is in Program Counter register (PC).

2. **Decode the instruction**: the decoder of the control unit is in charge of decoding the instruction.

3. **Execute the instruction**: the control unit enables the control signals in order to execute the instructions. For example if it is necessary to load some data from memory to some register or execute some arithmetic or logic instruction.

4. **Store**: the result of the instruction will be stored in the proper place: register, memory, device and eventually, should be updated with the address of the following instructions to be executed and the cycle will start again.

It is possible to watch some example of executing instruction in the following CPU:



As it can be seen, CPU has the arithmetic/logic unit, control unit and Registers. Register AX is the accumulator register and BX, CX, DX, EX and FX are the register for general purpose. On the other hand, it is possible to watch IR register below of the control unit and flags on the right. On the left of the control unit, it can be seen the PC register in order to take control of the following instruction to execute. On the right of the picture it is possible to find the memory which is considered out of the CPU.

On the other hand, in order to execute the instructions, CPU needs codify them. For example, in that CPU, the instruction MOVE for data handling is used for moving data from the memory, register or someplace to other place, register, memory address etc And in that CPU is codified as:

00 destination source

The destination refers to some register or addresses memory and is codified with 3 bits and the same with the source:

- 000 direct value and the value will be placed in the following memory adress
- 001 memory address and the adress will be placed in the following memory adress
- 010 AX register
- 011 BX register
- 100 CX register
- 101 DX register

- 110 EX register
- 111 FX register

For example, the instruction:

MOVE AX, FX

It would be used in order to move the value of the register FX to register AX and the code machine would be:

00 010 111

In order to execute the instruction, CPU would divide the execution of the instruction in some parts as it has seen before:

1. **Fetch the instruction**: where the PC register value would travel through address bus to the memory in order to find the instruction. The instruction would be read and load in IR register
2. **Decode the instruction**: once the instruction in the IR register, the control unit would decode the instruction in order to execute in the following stage.
3. **Execute the instruction**: as the data is in the FX register, the data would be loaded in the data bus to RX register. That register is in charge of the loading of every value.
4. **Store**: eventually, the value from RX register is loaded in AX and PC register is updated with the address of the following instruction to run.

Other example would be the ADD instruction in order to add some value from register, memory or direct value to register AX.

The instruction could be:

ADD 2

So, the machine code for arithmetic and logic instruction would be:

01 operation source

And the machine code for the instruction would be:

01 000 000 00 000 010

The first binary code shows the instruction and the following binary code show the value which is going to be added, in that case 2 in decimal code.

The source would be pointed out with same way of the source and destination of the instruction MOVE explained before and the operation would answer to the following code:

- 000 ADD
- 001 SUB
- 010 MUL
- 011 DIV
- 100 AND

- 101 OR
- 110 NOT
- 111 XOR

In order to execute the instruction, CPU would divide the execution of the instruction in some parts as it has seen before:

1. **Fetch the instruction**: where the PC register value would travel through address bus to the memory in order to find the instruction. The instruction would be read and loaded in IR register
2. **Decode the instruction**: once the instruction is in the IR register, the control unit would decode the instruction in order to execute in the following stage.
3. **Execute the instruction**: PC register would be increased and PC value would travel through the address bus in order to find the value (2 in that case) to be added, the value would be read in memory and loaded to the RX register. Once the operand is in RX register, it can be loaded in the arithmetic unit and added to AX register.
4. **Store**: eventually, the value of AX register is loaded with the new value and the flags register updated because an arithmetic operation has been executed. Besides, PC register is updated with the address of the following instruction to run.

5 RISC and CISC CPUs

CISC (Complex Instruction Set Computer) is an architecture designed in order to minimize the number of instruction, sacrificing the number of cycles per instruction. It means that the instructions are complex but it spends several cycles of clock to execute the instruction. The main features are:

- High set of instructions
- Variable size instructions.
- The number of instruction per program can be reduced by doing several operations in one instruction.

RISC (Reduced Instruction Set Computer) is an architecture which was introduced in the eighties because the CISC architecture was becoming more complex. The main characteristics of that architecture are:

- Reduced set of instructions.
- Fixed size of instructions
- Only load/store instructions access to memory.
- Each instruction can be run in just one clock cycle.
- CPU can work faster but the programs can become complex.

Nowadays CPUs use a mix of both architectures, it means that they are a CISC design but each instruction is divided in smaller instructions (RISC) because it should bear in mind:

- RISC architecture gets better performance.
- Programs of CISC architecture require less instructions and compilers are simpler.