# Presentation guide for unit 4
# Processor

The aim of this unit is to present the structure and operation of an elementary processor. This is done by describing how instructions are executed in a processor and how the control unit of a processor works and is designed. In this topic the following contents are presented:

1. Elements and organization of a processor
2. Execution of instructions
3. Execution modes
4. Control unit design
5. performance and parallelism

## 1. Elements and organization of a processor

This section describes all the elements that are part of a computer with Von Neumann architecture: memory, processor, buses and input/output modules, with special attention to the processor. All the elements of the processor are described: registers, arithmetic-logical unit and control unit. It is also defined the register transfer level language, the concept of elementary operations and the concept of control signals, as a previous step to the execution of instructions and control unit design.

The unit is illustrated with the WepWIM simulator (https://wepsim.github.io/wepsim/) developed by the authors of this course.

## 2. Execution of instructions

Once the elements of a processor are described, the execution phases of an instruction are described in detail:

- Reading the instruction (fetch)
- Decoding
- Execution

Several examples of instructions are presented and for each of them their compartment is defined by elementary operations and control signals.

## 3. Execution modes

The execution mode of a processor determines what type of program is being executed and what machine instructions it can and cannot execute. Basically, a processor includes at least a user mode and a kernel mode. The user mode is intended for the execution of user programs and in it, programs cannot execute privileged instructions. Examples of privileged instructions are input/output instructions. The kernel mode is reserved for the execution of the operating system and in it, the program that is running on the processor has full access to the entire instruction set of the processor. This section also describes the steps that a computer performs during the boot process.

## 4. Interrupts

An Interrupt is a signal that reaches the control unit and breaks the normal sequence of program execution (a division by zero, access to an illegal memory position, etc.). When an interrupt is generated, the program currently running in the processor is stopped and the execution is transferred to another program that attends the interrupt. The program in charge of attending and processing the interruption is part of the operating system. This section describes the different types of interrupts that can occur and the steps that the control unit performs when an interruption occurs. It also describes the mechanism of vectorized interrupts, which is mostly used to determine the directions where they are stored routines that should treat and process the various interruptions that occur in a processor.

## 5. Control unit design

The control unit is the element of the processor that is responsible for the execution of instructions. It is responsible, therefore, for generating all the control signals needed to execute the various machine instructions that are part of the programs. The control unit takes as input the instruction register, which stores the instruction currently being executed in the processor, the status register that stores information on the status of the program being executed, the signal that indicates whether there have been interrupts or not and the clock signal. With all these elements, is responsible for generating all necessary control signals. In this section of the topic, illustrates how the steps are to design a control unit and describes two types of control units:

- Wired control unit
- Microprogrammed control unit

The unit dedicates special attention to the microprogrammed control units, which have a control memory where all the control signals to be activated by the control unit in the process of instruction execution are stored. The concept of microprogramming is illustrated with the WepSIM simulator.

## 6. Parallelism and performance

The unit ends with a short introduction to computers that use parallelism at the instruction level with the aim of improving performance. Segmented processors, which use segmentation or pipeline techniques to process several instructions simultaneously, are briefly described. Superscalar processors, which are segmented processors that can execute several machine instructions in parallel, one of them in a different segmented unit. Also, is presented the concept of the multi-core processor, which combines two or more independent processors in a single chip or package.

## Material

As material associated with this unit is included the theory material and a collection of exercises proposed and solved on the aspects covered in the unit. Access to the WepSIM simulator is also provided. WepSIM is a web simulator, which simulates the operation of an elementary processor with a microprogrammed control unit and allows the definition of different types of instruction sets. We propose a practice, which uses the WepSIM simulator, to understand how through microprogramming the instruction set of a processor can be designed. Other resources provide a

link to various simulators of other processors and simulators that illustrate the operation of the pipeline.

**Lecturas recomendadas**

- "Problemas resueltos de estructuras de computadores" (GARCIA CARBALLEIRA, Félix et al.).
- "Computer organization and design. The hardware/software interface" (PATTERSON, David, et al).
- "Computer Organization and Architecture" (STALLINGS, William).