

## Tema 2

### RESUMEN

En el **tema 2** (*Arquitectura Interna de una CPU*) se enseña la arquitectura interna de un microprocesador. Además, se ilustra la relación entre microprocesadores y microcontroladores. Todo el tema se desarrollará de forma genérica, ilustrando los conceptos con el microprocesador (ARM Cortex-M3) y microcontrolador (STM32L152RBT6) utilizado durante el curso.

El tema se encuentra dividido en los siguientes puntos:

En el primer punto se habla sobre los diferentes tipos de arquitecturas, se presenta la familia ARM y se introduce la arquitectura del ARM7. Se recomienda visionar el vídeo suministrado para este tema, para así entender mejor la arquitectura Von Neuman y algunos conceptos adicionales que pueden ser más complejos de seguir sólo con las transparencias. Dicho vídeo no cubre todo lo indicado en las transparencias, pero consideramos que es una ayuda esencial al estudiante.

A continuación, se describe qué es la notación RTL en el mundo de los microcontroladores y se pone un ejemplo muy sencillo.

En el tercer punto se explica qué es la unidad de control y cómo funciona con sus tres fases (*fetch*, *decode* y *execute*), así como el concepto de ciclo máquina y segmentación.

A continuación se muestra la unidad aritmético-lógica con sus rutas de datos, tanto para el ARM7 como para el Cortex-M3.

En el siguiente punto se explican algunos de los registros más importantes del Cortex-M3, imprescindibles para su funcionamiento correcto.

- El Contador de Programa (PC), el cual contiene en cada instante la dirección de la siguiente instrucción a capturar.
- El Registro de Estado (SR), que sirve para almacenar el estado en el que se encuentra la CPU tras la ejecución de una instrucción, y se muestran algunos de sus **flags** más importantes.
- El Puntero de Pila (SP), que es una zona de memoria dedicada a almacenar información en formato LIFO (el último que entra es el primero que sale).

A continuación se muestra la memoria principal, introduciéndose varios conceptos fundamentales relacionados con ella.

- El mapa de memoria, para saber donde se guardan las instrucciones y datos en cada uno de los distintos chips.
- El decodificador de direcciones para saber a qué chip se accede dependiendo del rango al que pertenece la dirección de la memoria.
- Los diferentes bancos de memoria para poder ampliar artificialmente la memoria principal del sistema.

Finalmente se explica cómo funciona el registro de instrucciones, que es el encargado de mantener la instrucción que se va a ejecutar, para que se decodifique y se emprendan las acciones necesarias. Como mínimo, las instrucciones se codifican en dos partes: **opcode**, que es el código que indica la instrucción a ejecutar y su variante (modo de direccionamiento, etc.) y los parámetros, que normalmente determinan uno o varios operandos o la dirección de uno o varios operandos. Se habla también de la diferencia entre diferentes juegos de instrucciones: RISC/CISC y Thumb/Thumb-2, y se muestran ejemplos de una instrucción genérica de transferencia de datos y otra de procesamiento de datos. Esto da pie para entrar, en el próximo tema a hablar de Lenguaje Ensamblador, así como de los diferentes modos de direccionamiento.