

Tema 3

RESUMEN

En el **tema 3** (*Lenguaje Ensamblador*) se enseña el lenguaje de ensamblador de un microcontrolador y microprocesador, particularizando los conceptos en el utilizado en el curso (STM32L152RBT6 de ARM), divididos en varios puntos:

En el primer punto se habla del modelo de programador, el cual establece el conjunto de elementos que son necesarios conocer de la arquitectura interna de la CPU para realizar programas. En particular se explica lo siguiente.

- El modelo de memoria utilizado en el ARM7, el cual direcciona bytes, con codificación little-endian, y en el que los datos se pueden almacenar de forma alineada o no alineada.
- Los tipos de datos utilizados, que pueden ser los siguientes: “word”, “halfword”, “byte”, “con signo” y “sin signo”.
- Los modos de operación, que pueden ser: “thread mode” para ejecutar aplicaciones software y “handler mode” para gestionar las excepciones.
- Los niveles de privilegio, que pueden ser: “unprivileged”, en el que el software tiene acceso limitado a algunas instrucciones y “privileged”, en el que el software tiene acceso a todas las instrucciones y todos los recursos.

A continuación se describen los diferentes mecanismos de programación utilizados, es decir, el desarrollo de un programa se realiza mediante la concatenación de instrucciones que se ejecutarán siguiendo un flujo lineal, pero este flujo lineal se puede alterar mediante.

- Saltos: El programa continúa la ejecución en otro punto de la memoria, normalmente en función de una condición que se debe cumplir.
- Subrutinas, al igual que las llamadas a función en C, en las que se realiza un salto temporal, volviendo posteriormente a la instrucción siguiente a la del salto a subrutina.
- Interrupciones, por el cual se permite que la CPU realice determinadas operaciones, cuando haya ocurrido un determinado evento externo. Cuando ocurre ese evento se pasa a ejecutar una subrutina, conocida como “Rutina de Atención a la Interrupción” o “Rutina de Servicio”.

En el tercer punto se explica el juego de instrucciones utilizado en el ARM7, pero con una serie de simplificaciones para el curso. Se muestran todas las instrucciones divididas en las siguientes.

- Transferencia de Datos (entre memoria y CPU, entre registros)

- Aritméticas y Lógicas
- Control de Flujo

Una vez visto cómo son las instrucciones en un microprocesador, se pasa a explicar cómo poder especificar los distintos operandos. Para ello se explicarán con detalle los diferentes modos de direccionamiento, es decir la forma de indicar la ubicación del operando. De forma genérica, existen los siguientes:

- Inherente o Implícito
- Inmediato o Literal
- Directo o Absoluto
- Indirecto
- Indirecto con Desplazamiento
- Indexado
- Relativo a PC

Una vez vistos los modos de direccionamiento genérico, se pasará a hablar de los diferentes modos de direccionamiento del ARM7 y se explican en detalle dos ejemplos para el “grupo 1 <op2>” (para especificar el segundo operando en las instrucciones de procesamiento de datos) y “grupo 2 <am2>” (operando en instrucciones de transferencia de datos de tamaño “word” o “unsigned byte”).

Finalmente se explica la diferencia entre la programación en ensamblador y la programación utilizando compiladores, mostrando un ejemplo práctico con el entorno de desarrollo:

- Programando en lenguaje ensamblador se puede considerar que cada instrucción corresponde a una instrucción en código máquina, de manera que se trabaja muy cercano al microcontrolador y no hay lugar a interpretaciones.
- Por el contrario, los compiladores de lenguajes de medio y alto nivel, tienen que interpretar el código escrito, y buscarle una traducción a lenguaje ensamblador. Esas interpretaciones ayudan por un lado al programador a desarrollar su solución, pero le hacen perder algo el control de la CPU.