INFORME DE LABORATORIO 4

Autores: *María Del Mar Arbeláez Sandoval, Julián Mauricio Sánchez Ceballos*

Laboratorio de Electrónica Digital 2
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia

Resumen

Esta práctica consiste en el uso del lenguaje ensamblador para procesadores ARM de 32 bits con el propósito de generar y manipular valores de 64 bits de la serie de Fibonnaci, esto haciendo uso de los conceptos de manejo de pila (Stack), manejo de memoria y funciones. En este informe se presentan algunos conceptos teóricos necesarios para el desarrollo de la practica, los diagramas de flujo en los que se basa el código en ensamblador, explicación del código y por ultimo las conclusiones de la practica.

Palabras clave: ARM, ensamblador, Fibonnaci, Stack, Instrucciones, Registros.

Introducción

En esta practica de laboratorio se implemento mediante el uso del lenguaje ensamblador para procesadores ARM de 32 bits un programa que genera y manipula un conjunto de datos pertenecientes a la serie de Fibonnaci, estos valores son de 64 bits y por tanto se requiere una escritura especial en la memoria, además, se emplea el uso de funciones por lo que el uso del stack es indispensable para guardar valores que deban preservarse para usar una vez terminada la ejecución de las funciones.

Diseño

Para hacer mas comprensible la forma en que fue diseñado el programa en lenguaje ensamblador se presentan los diagramas de flujo que corresponden a una interpretación en más alto nivel de lo logrado en ensamblador, el siguiente es el flujo de todo el programa presentado de forma general:

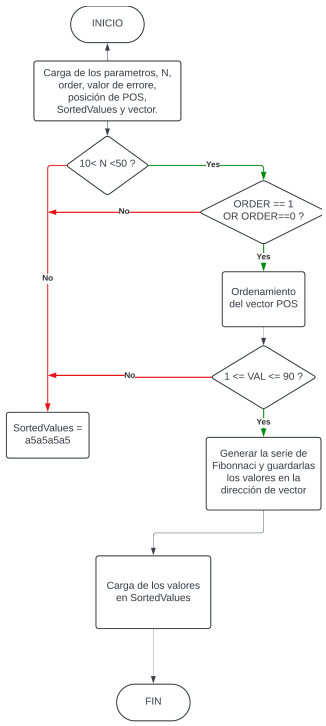

Figura **0-1**: Diagrama de flujo del programa.

La función de Fibonnaci tiene el siguiente diagrama de flujo:

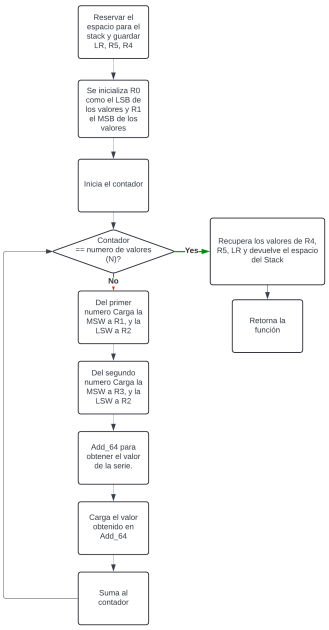

Figura **0-2**: Diagrama de flujo para la función que genera los valores de la serie.

y finalmente el diagrama de flujo del algoritmo de ordenamiento implementado en el código es:

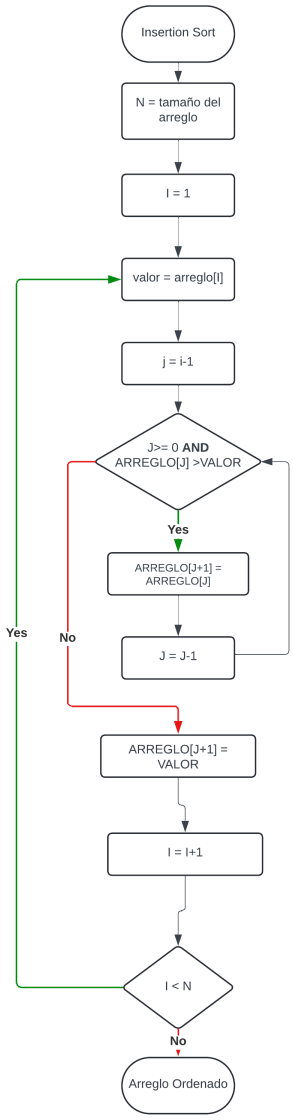

Figura **0-3**: Diagrama de flujo para el algoritmo de ordenamiento.

Simulación

Junto con este informe se entrega el código implementado en ensamblador y debidamente comentado.

Conclusiones

- Como se pudo observar el lenguaje ensamblador implementa una representación simbólica de los códigos de maquina, que, generalmente son escri-

tos en '1' y '0', sin embargo con esta representación simbólica se logra que el programador pueda escribir instrucciones muy cercanas al procesador, lo que confluye en un entendimiento de las diferentes dinámicas que maneja un procesador que implemente una arquitectura ARM de 32 bits como el manejo de memoria, manejo del Stack y la forma en que un procesador hace operaciones aritméticas, conectando los conocimientos previos en otros lenguajes de programación de alto nivel como C y Java, y el modo en que el procesador realmente abstrae estos lenguajes de alto nivel a instrucciones de bajo nivel.

- La cercanía al lenguaje con el procesador y al mismo tiempo la limitación en instrucciones y su funcionamiento hace que escribir códigos en lenguaje ensamblador requiera de seguir convenciones muy especificas y genera en el estudiante una necesidad de aplicar la lógica de solución de problemas de forma óptima, esto no siempre resulta en un tamaño óptimo del código, pues como se pudo observar, estos códigos suelen tener muchas mas lineas para ejecutar programas que en otros lenguajes de más alto nivel tomarían menos lineas.