

# Diseño: Procesamiento de texto.

Carlos Adrián Araya Ramírez, Michael Shakime Richards Sparks  
adrian.araya@estudiantec.cr, jeykimers.johnson@estudiantec.cr

CE3201 - Taller de Diseño Digital  
Área Académica Ingeniería en Computadores  
Instituto Tecnológico de Costa Rica

15 de junio de 2022

## Índice

<b>1. Introducción.....</b>	<b>2</b>
<b>2. Requerimientos del Sistema .....</b>	<b>2</b>
2.1. Diseño del microprocesador ARMv4 .....	2
2.2. Desarrollo en HDL del microprocesador ARMv4.....	3
2.3. Desarrollo del código en ARMv4 .....	4
2.4. Conexión del código ARMv4 con en el procesador .....	5
<b>3. Referencias.....</b>	<b>6</b>

# 1. Introducción

El diseño en los proyectos de ingeniería es tan importante como la implementación de este. En este proyecto se dedicó gran parte del tiempo para el desarrollo de una propuesta de diseño capaz de resolver el problema de forma conceptual. Para lograr este diseño inicial se plantearon los diferentes requerimientos del sistema a desarrollar, donde se explican los problemas a resolver y una posible solución planteada por el grupo de trabajo.

## 2. Requerimientos del Sistema

El sistema por desarrollar corresponde a un microprocesador uniciclo que sea capaz de ejecutar algunas de las instrucciones de la arquitectura ARMv4, con este procesador se espera desarrollar un programa en código ensamblador que permita encontrar palabras con al menos 3 vocales en un texto previamente cargado en una memoria ROM y las reemplace por algún símbolo de entrada. El texto inicial antes de ser modificado y después de la modificación debe mostrarse en un Monitor VGA y las palabras con más de tres vocales deben ser coloreadas con un color distinto. A continuación, se muestra una lista detallada de los requerimientos del sistema a desarrollar.

### 2.1. Diseño del microprocesador ARMv4

El diseño de un procesador es similar para todas las arquitecturas, sin embargo, para el equipo de trabajo es el primer acercamiento al diseño de procesadores por lo que se decidió realizar un diseño basado en la teoría y diseño de un procesador uniciclo que se presenta en el libro del curso. Las partes involucradas en el desarrollo del procesador uniciclo se detallan a continuación.

1. Módulo del registro PC: corresponde a un registro que almacena el valor actual del *program counter*.
2. Módulo de banco de registros: en este módulo se almacena el valor de cada uno de los 16 registros de la arquitectura ARMv4.
3. ALU: este módulo es el encargado de realizar las operaciones lógicas y aritméticas necesarias en la mayoría de las instrucciones de la arquitectura.
4. Módulo extensor: este módulo se encarga de realizar todas las extensiones de signo y de ceros necesarias en cada una de las instrucciones.
5. Multiplexor: en la implementación del procesador es necesario un módulo multiplexor ya que es necesario instanciar varios de ellos para seleccionar los datos que ingresan a cada uno de los componentes dependiendo de cada instrucción.
6. Sumador: es necesario un módulo capaz de realizar la suma de dos valores debido a que al tratarse de un procesador uniciclo no se puede utilizar la ALU más de una vez por instrucción y el incremento del PC y las instrucciones de salto necesitan un sumador.
7. Memoria de instrucciones: esta memoria es externa al procesador, pero es necesario implementarla para poder almacenar y leer el programa que se busca ejecutar.

8. Memoria de datos: esta memoria también es externa al procesador, pero es necesaria su implementación para poder leer y guardar los datos relacionados al texto.
9. Unidad de control: este módulo se encarga de controlar el flujo completo de las instrucciones mediante el cambio de estado en las banderas y señales de selección a partir de la decodificación de la instrucción. Este módulo es el más importante debido a que controla lo que deben realizar todos los módulos que pertenecen al procesador.

Otros aspectos sobre este requerimiento:

- Estado del arte: el diseño de la arquitectura ARM inicio en 1983 y ha sido muy popular debido a que su relativa simplicidad los hace ideales para aplicaciones de baja potencia. Roger Wilson y Steve Furber, fueron los pioneros en el desarrollo de esta tecnología en 1985 cuando presentaron el primer chip ARM1 pero ARM2 fue el primero en ser comercial y su principal característica es que contenía un bus de 32 bits. El ARM2 era muy sencillo, contenía solo 30,000 transistores. En 2005, alrededor del 98% de los más de mil millones de teléfonos celulares vendidos utilizan al menos un procesador ARM. Desde 2009 los procesadores ARM son aproximadamente el 90% de los procesadores RISC de 32 bits integrados [1].
- Estándares: el diseño deberá ser lo más modular posible con el objetivo de aumentar la legibilidad, mantenibilidad y la escalabilidad del código.
- Normas: los nombres de los componentes y las entradas y salidas de estos deben tener nombres sencillos y representativos, en los diseños, los nombres de las señales deberán utilizar formato PascalCase.

## **2.2. Desarrollo en HDL del microprocesador ARMv4**

Basados en el diseño anteriormente presentado se debe programar por medio de un lenguaje de descripción de hardware el microprocesador planteado. Para este punto se espera utilizar las herramientas aprendidas en el curso. Las partes involucradas en este requerimiento se detallan a continuación.

1. Desarrollo de módulos combinacionales: la mayoría de los módulos del procesador son combinacionales debido a que se trata de un procesador unificado, la ALU, el extensor, los MUX y los sumadores son módulos que dependen únicamente de sus entradas por lo que se deben implementar correctamente para un funcionamiento adecuado.
2. Desarrollo de módulos secuenciales: en este procesador los módulos dependientes del tiempo son el registro del PC, el registro de las banderas para los condicionales, las memorias externas y el banco de registros.
3. Desarrollo de una unidad de control: según el planteamiento mencionado anteriormente, este módulo debe implementarse de tal forma que sea capaz de controlar el funcionamiento de los demás componentes a partir de las banderas (enable) y señales de selección.

4. Conexión de las memorias externas con el procesador: se debe implementar un módulo que realice la conexión entre el procesador y las memorias externas.

Otros aspectos sobre este requerimiento:

- Estado del arte:
- Estándares: los nombres de los módulos se escribirán en mayúsculas y el nombre de las entradas y salidas se escribirán en PascalCase.
- Normas: todos los módulos deberán ser probados, compilados y simulados para garantizar el correcto funcionamiento de estos y evitar complicaciones al realizar la conexión. Todos los módulos deben estar debidamente documentados sin sobrecargarlos para evitar problemas de legibilidad.

## 2.3. Desarrollo del código en ARMv4

El procesador por sí solo no realizará la tarea de manejar el texto, sino que será capaz de ejecutar una serie de instrucciones en lenguaje ensamblador para la arquitectura ARMv4.

Es por esto por lo que uno de los requerimientos es el desarrollo de este código en ensamblador. Para este se deben considerar las siguientes partes:

1. Estudio de las instrucciones disponibles en la arquitectura ARMv4: para la implementación del algoritmo que procesará el texto es necesario conocer las instrucciones disponibles en ARMv4 para así determinar cuáles instrucciones deberá ser capaz de ejecutar el procesador e implementar solo las que son necesarias.
2. Desarrollo del algoritmo de procesamiento de texto: para programar la solución del problema mediante lenguaje ensamblador se debe investigar la documentación y sintaxis del lenguaje hasta el punto de ser capaces de implementar el algoritmo más eficiente que cumpla con los requisitos.

Otros aspectos sobre este requerimiento:

- Estado del arte: Stan Poley escribió el programa de Ensamblaje Óptimo Simbólico o lenguaje ensamblador SOAP para la computadora IBM 650 en 1955. Los lenguajes ensambladores comenzaron a usarse ampliamente, ya que liberaban a los programadores de tareas tediosas como recordar códigos numéricos. Sin embargo, su uso se redujo sustancialmente en la década de 1980 debido a la introducción de los lenguajes de alto nivel [2].
- Estándares: el código se escribirá con todas las instrucciones en mayúsculas y se deben utilizar la mínima cantidad de instrucciones. Los registros deben ser utilizados en orden, es decir, empezando desde el r0 e ir usando los demás en orden de necesidad.
- Normas: los registros deben ser reutilizados siempre que sea posible y el programa debe ser probado las veces que sean necesarias para garantizar que funciona perfectamente antes de extraer el binario.

## **2.4. Conexión del código ARMv4 con en el procesador**

Para lograr la conexión del código ARM con el procesador se deben cargar las instrucciones del programa compilado y codificado en hexadecimal a la memoria de instrucciones. Los pasos por considerar en este aspecto son los siguientes:

1. Obtener el código binario del programa: para esta tarea se debe utilizar una herramienta capaz de compilar y generar el código binario equivalente al programa, el cuál además permite depurar en el proceso, esta herramienta se detalla en el documento de herramientas de ingeniería.
2. Cargar el binario del programa en la memoria de instrucciones: se debe buscar algún método para extraer el código binario y cargarlo al archivo de inicialización de la memoria ya sea mediante un script o alguna herramienta.

Otros aspectos sobre este requerimiento:

- Estándares: el código del programa se debe guardar en un archivo de inicialización de memoria codificado en hexadecimal.
- Normas: todos los archivos relacionados a la implementación del código en ARM deben estar organizados y localizados en un directorio específico donde se tenga como mínimo el archivo correspondiente al código en ensamblador y su equivalente en binario.

### 3. Referencias

[1] M. Molina, "Historia de los procesadores ARM", *PasionMovil*, 2012. [Online]. Available: <https://www.pasionmovil.com/investigacion-y-desarrollo/que-es-la-arquitectura-arm/>. [Accessed: 16- Jun- 2022].

[2] R. IT, "Reclu IT", *Recluit.com*, 2020. [Online]. Available: <https://recluit.com/que-es-el-lenguaje-assembly/>. [Accessed: 16- Jun- 2022].