

Laboratorio de Electrónica Digital II

Práctica No. 6: Procesador ARM con Ejecución de Instrucciones en un Solo Ciclo

Profesores

Luis Fernando Castaño L. (luis.castanol@udea.edu.co)

Luis Germán García M. (german.garcia@udea.edu.co)

Octubre 17, 2023



Fecha de entrega: Del 31 de Octubre al 3 de Noviembre de 2023

Medio de entrega: <https://virtualingenieriaudea.co/>

Sustentación: Horario de Laboratorio

Valor Práctica: 12% del curso

1 Introducción

En esta práctica de laboratorio, el grupo de trabajo (máximo dos integrantes) llevará a cabo la implementación de una versión simplificada del procesador ARM con ejecución de instrucciones en un solo ciclo. El desarrollo será llevado a cabo en dos partes o etapas. La primera parte se realizará durante la primera semana y consistirá en estudiar el código del procesador dado e implementarlo en el sistema de desarrollo DE10-Lite. Para la prueba, será dado un programa escrito en ensamblador, el cual deberá ser cargado en las memorias del sistema antes de descargar

el desarrollo en la FPGA. La segunda parte se realizará durante la segunda semana y consistirá en agregar al procesador ARM implementado previamente, un conjunto de nuevas instrucciones. Además, el grupo de trabajo deberá crear un programa en lenguaje ensamblador para ser ejecutado sobre la versión modificada del procesador. La descripción del hardware será realizada mediante el uso del lenguaje de descripción de hardware SystemVerilog, mientras que los programas de prueba serán escritos en ensamblador, con la ayuda de CPULATOR. Se deberán realizar pruebas tanto en simulación como con el sistema de desarrollo DE10-Lite.

2 Objetivo de la Práctica

Implementar una versión simplificada del procesador ARM con ejecución de instrucciones en un solo ciclo en el sistema de desarrollo DE10-Lite, mediante el uso del lenguaje de descripción de hardware SystemVerilog, con el propósito de ejecutar un conjunto de programas escritos en lenguaje ensamblador que haga uso de los periféricos del sistema.

3 Parte 1: estudio e implementación del procesador ARM

Para el desarrollo de esta primera parte, siga los siguientes pasos:

- a. Descargue el código del procesador ARM con ejecución de instrucciones de un solo ciclo, disponible en la plataforma del curso Ingeni@.
- b. Estudie la sección 7.3 del texto guía del curso, *Digital Design And Computer Architecture - ARM Edition*, para comprender en qué consiste y cómo funciona el procesador ARM con ejecución de instrucciones en un solo ciclo. La microarquitectura del mencionado procesador fue también explicada en la parte teórica del curso. Observe y comprenda el esquemático del procesador ARM disponible en la figura 7.13 del texto guía. A continuación, revise el código que ha descargado y comprenda, principalmente, el funcionamiento y construcción de las unidades *Datapath* y Control, observando detalladamente como cada unidad está construida a partir de otros módulos.
- c. Analice el código de las unidades de *Datapath* y Control para entender cómo se han implementado las instrucciones siguientes:
 - (a) Procesamiento de datos: ADD, SUB, AND, ORR, usando registros y valor inmediato sin desplazamientos.
 - (b) Memoria: LDR y STR, con desplazamiento inmediato positivo.
 - (c) Salto: Bxx.
- d. Sintetice el código del procesador junto con las memorias de instrucciones y datos ('top.sv') y simule, con la ayuda del profesor del laboratorio, el testbench dado en la sección 7.6 del texto guía del curso, incluido en el código que descargó (archivos 'testbench_book.sv' y 'code_tb_from_book.txt'), para verificar el correcto funcionamiento del procesador ARM. Los

valores en código de máquina tanto para las instrucciones como para los datos han sido escritos en los archivos 'imem_tb_from_book.dat' y 'dmem_tb_from_book.dat', respectivamente. Observe que el archivo para la memoria de datos está vacío ya que el programa no requiere datos iniciales desde la memoria. Antes de ejecutar la simulación, agregue/retire los correspondientes comentarios de los archivos 'imem.sv' y 'dmem.sv' para que se carguen las instrucciones y datos correspondientes. Una vez completada la simulación, responda ¿Qué valor y en qué dirección de memoria escribe la instrucción STR R2, [R0, #100] al finalizar el código? ¿Cómo comprobar que el código ha sido correctamente ejecutado por su procesador ARM?

- e. Revise el código en SystemVerilog de la memoria de datos 'dmem.sv' y comprenda como fueron mapeados en memoria los suiches y los leds del sistema de desarrollo DE10-Lite. ¿Cuál dirección de memoria corresponde a los suiches y cuál a los leds? ¿Cuál dispositivo es de lectura y cuál es de escritura? ¿Cómo podría leer el valor de los 10 suiches empleando instrucciones en ensamblador? ¿Cómo podría encender y apagar los 10 leds empleando instrucciones en ensamblador?
- f. Comprenda el código del programa escrito en lenguaje ensamblador, dado en el archivo 'code_to_test_peripherals.txt'. Para este programa tenga en cuenta que cuando se realicen lecturas a las direcciones de memoria 0x0000_0000 y 0x0000_0004, la memoria de datos retornará los valores 0xC000_0004 y 0x002F_0000, respectivamente, debido a los datos preestablecidos en el archivo 'dmem_to_test_peripherals.dat'.
- g. Agregue/retire los correspondientes comentarios en los archivos 'imem.sv' y 'dmem.sv' para que se carguen, tanto en la memoria de instrucciones como la de datos, las correspondientes instrucciones ('imem_to_test_peripherals.dat') y datos ('dmem_to_test_peripherals.dat') del programa de prueba dado en el archivo 'code_to_test_peripherals.txt'.
- h. Compile el código del procesador junto con las memorias de instrucciones y datos ('top.sv') y programe la FPGA del sistema DE10-Lite. A continuación, ponga todos los suiches del sistema a '0' (OFF) y comience a mover algunos de ellos a la posición de '1' (ON). ¿Cuál es el comportamiento de los leds ante las variaciones en los suiches? ¿Corresponde con el programa que usted analizó ('code_to_test_peripherals.txt')? ¿Cómo se podría cambiar el tiempo entre cambios en los leds sin modificar las instrucciones del programa ni el hardware del procesador?
- i. Agregue al reporte definitivo las respuestas a todas las preguntas realizadas previamente.

4 Parte 2: agregando instrucciones al procesador ARM y creando un nuevo programa de prueba

Para el desarrollo de esta segunda parte, siga los siguientes pasos:

- a. Cree una copia al código del procesador ARM con ejecución de instrucciones de un solo ciclo, que empleó en la primera parte de esta práctica y llámela 'ARM-SingleCycle-mod'. A continuación, realice sus modificaciones al nuevo código.
- b. Modifique las unidades de *Datapath* y Control para que soporte las siguientes instrucciones adicionales a las ya existentes:
 - (a) Procesamiento de datos (1): LSL, LSR, ASR, ROR, usando solamente valores inmediatos.
 - (b) Procesamiento de datos (2): MOV, usando registros y valor inmediato sin rotación.
 - (c) Salto: BL.
- c. Realice un programa en lenguaje ensamblador empleando las instrucciones soportadas por la nueva versión de su procesador ARM, que cumpla con los siguientes requerimientos:
 - (a) Generar dos secuencias de luces empleando los leds del sistema de desarrollo DE10-Lite. La secuencia a mostrarse en determinado momento se escoge mediante uno de los suiches del sistema DE10-Lite.
 - (b) Cada secuencia deberá emplear los 10 leds, además de repetirse de manera indefinida.
 - (c) El programa deberá hacer uso de al menos tres de las nuevas instrucciones de procesamiento de datos: LSL, LSR, ASR, ROR o MOV.
 - (d) El programa deberá hacer uso de la instrucción BL, es decir, implementar al menos una función y realizar el respectivo llamado.
- d. Convierta a lenguaje de máquina el programa escrito en lenguaje ensamblador y guárdelo en los archivos 'imem_made_by_students.dat' (instrucciones) y 'dmem_made_by_students.dat' (datos). ¿Cómo hacerlo de manera rápida? Realice los respectivos cambios en los archivos 'imem.sv' y 'dmem.sv' para cargar las nuevas instrucciones y datos.
- e. Cree un testbench para su procesador ARM modificado, junto con el programa realizado, y verifique el funcionamiento de todo el sistema. ¿Cómo hacerlo de manera efectiva? Consulte con el profesor del laboratorio si tiene dificultades.
- f. Compile el código del procesador junto con las memorias de instrucciones y datos ('top.sv') y programe la FPGA del sistema DE10-Lite. A continuación, verifique que ambas secuencias se ejecuten de manera correcta.
- g. Sustente el trabajo al profesor del laboratorio.
- h. Agregue al reporte las modificaciones realizadas al hardware del procesador para soportar las nuevas instrucciones (empleando esquemáticos) y los códigos escritos en lenguaje ensamblador.

5 Entrega

El grupo de trabajo deberá escribir un breve reporte en formato IEEE que contenga los siguientes elementos (ver guía para reportes en la página del curso):

- a. **Abstract:** resumen del diseño e implementación de la práctica.
- b. **Respuestas a preguntas:** respuestas a las preguntas de la primera parte del desarrollo de la práctica.
- c. **Esquemas de HW:** esquemáticos del hardware del procesador ARM modificado para soportar las nuevas instrucciones.
- d. **Código ensamblador:** código del programa en ensamblador realizado para la segunda parte del desarrollo de la práctica.
- e. **Simulación:** simulación de la segunda parte del desarrollo de la práctica.
- f. **Conclusiones:** dos o tres conclusiones sobre el trabajo realizado por el grupo de trabajo. Indicar el tiempo que les tomó realizar la práctica en las conclusiones.

Crear un archivo comprimido que incluya el reporte y los archivos importantes de su código en ensamblador como se describe a continuación:

- a. **Reporte:** archivo con extensión .pdf
- b. **Archivos esenciales:** proyecto en QuartusPrime y códigos en ensamblador.

El nombre del archivo comprimido deberá tener el siguiente formato:

p6_primerapellidointegrante1_primerapellidointegrante2_horariolaboratorio.zip.

Ejemplo: si el primer apellido de ambos integrantes es **Castano** y **Garcia**, respectivamente, y el laboratorio es el Martes 9-12, entonces el archivo debe ser nombrado: *p6_castano_garcia_m9-12.zip*.

6 Evaluación

La evaluación de la práctica se divide en tres partes: funcionamiento (40%), sustentación (50%) y reporte (10%). Las notas de funcionamiento y reporte se asignan por igual a todos los integrantes del grupo de trabajo (máximo dos personas por equipo), mientras que la nota de sustentación es individual. La sustentación podrá realizarse de cuatro maneras posibles:

- a. Un examen corto al finalizar la sesión de laboratorio.
- b. Solicitud de cambios al código por parte del profesor de laboratorio.
- c. Un par de preguntas orales que pongan a prueba los conocimientos del estudiante sobre el desarrollo de la práctica.

- d. Una presentación corta a todo el grupo de laboratorio.

En caso un estudiante obtenga una nota inferior a 3.0 en la sustentación, la nota final de la práctica para el estudiante en mención será la que obtuvo en la sustentación, es decir, no se tendrá en cuenta el funcionamiento en el cálculo.

Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 15 minutos, 8 minutos para revisar la funcionalidad y 7 minutos para la sustentación en cualquiera de las cuatro modalidades presentadas. Es importante tener abierto el CPULATOR y el Quartus Prime con el programa listo para cuando el profesor llegue a su puesto de trabajo. No habrá tiempo para hacer correcciones de último momento.

7 Referencias

- a. Harris, Sarah and Harris, David. Digital Design and Computer Architecture: ARM Edition. Morgan Kaufmann Publishers Inc. 2015. ISBN: 0128000562
- b. CPULATOR Computer System Simulator
<https://cpulator.01xz.net/?sys=arm>
- c. Quartus Prime Lite Edition
<https://www.intel.com/content/www/us/en/products/details/fpga/development-tools/quartus-prime.html>
- d. Sistema de desarrollo DE10-Lite
<http://de10-lite.terasic.com/>
- e. Tutorial de SystemVerilog
<https://verilogguide.readthedocs.io/en/latest/>