

Laboratorio de Electrónica Digital II

Práctica No. 5: Procesador ARM con Ejecución de Instrucciones en un Solo Ciclo

Profesores

María Katherine Plazas O. (maria.plazas@udea.edu.co)

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

Abril 30, 2024



Fecha de entrega: del 14 al 16 de mayo de 2024

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

Sustentación: Horario de laboratorio

Valor Práctica: 10% 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, junto con un programa que pueda ejecutarse en el mencionado procesador, haciendo uso de los periféricos del sistema. La descripción del hardware será realizada mediante el uso del lenguaje de descripción de hardware SystemVerilog, mientras que el programa será escrito en lenguaje ensamblador para ARM, 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, mediante el uso del lenguaje de descripción de hardware SystemVerilog, que pueda ejecutar programas desarrollados en lenguaje ensamblador para ARM en el sistema de desarrollo DE10-Lite.

3 Procedimiento

Para el correcto diseño e implementación de esta práctica, se sugiere realizar el siguiente procedimiento:

- a. Lea esta guía varias veces para comprender el trabajo a realizar, identificando así los requerimientos que deberán cumplirse el día de la sustentación.
- b. Descargue el código del procesador ARM con ejecución de instrucciones de un solo ciclo, disponible en la plataforma del curso Ingeni@ ('05-ARM-SingleCycle-students').
- c. Estudie la sección 7.3 del texto guía del curso, *Digital Design And Computer Architecture - ARMEdition*, 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.
- d. Analice el código de las unidades de *Datapath* y Control para entender cómo se han implementado las siguientes instrucciones:
 - (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.
- e. Repita el procedimiento de síntesis, simulación y despliegue en la FPGA realizado en el tutorial desarrollado en el laboratorio, para tener una mejor comprensión del procesador ARM con ejecución de instrucciones de un solo ciclo.
- f. Lleve a cabo el desarrollo propuesto en la siguiente sección, realizando las simulaciones y correcciones pertinentes, antes de llevar a cabo el despliegue final a la FPGA.
- g. Envíe el código fuente en SystemVerilog del procesador, junto con el programa hecho en lenguaje ensamblador para el procesador ARM y un breve reporte, antes de la fecha límite. Consideraciones sobre el reporte serán dadas al finalizar esta guía.

- h. Sustente el desarrollo en el horario de laboratorio correspondiente.

4 Especificaciones

Para el desarrollo de esta práctica, siga los siguientes pasos:

- a. Cree una copia al código del procesador ARM con ejecución de instrucciones de un solo ciclo que descargó de la plataforma de Ingeni@ ('05-ARM-SingleCycle-students') y llámela '05-ARM-SingleCycle-mod'. A continuación, realice las modificaciones correspondientes en este nuevo código.
- b. Modifique las unidades de *Datapath* y Control para que el procesador soporte las siguientes instrucciones adicionales a las ya existentes:
 - (a) Procesamiento de datos (1): LSL, LSR, ASR, ROR, usando solamente valores inmediatos. Para la realización, tenga en cuenta el ejemplo 7.3 del texto guía.
 - (b) Procesamiento de datos (2): MOV, usando registros y valor inmediato sin rotación. ¿Qué habría que hacer para permitir la rotación?
 - (c) Procesamiento de datos (3): CMP.
- c. Agregue un módulo en SystemVerilog llamado 'displays.sv' para tener un nuevo periférico llamados Displays. Este periférico deberá manipular cinco (5) displays de 7 segmentos del sistema DE10-Lite: los cuatro menos significativos se emplearán para mostrar un valor en el rango -128 a 127 (decimal), mientras que el quinto se empleará para mostrar una de las letras 'A', 'b' y 'r'. El periférico Displays deberá ser mapeado en dos direcciones de memoria: 0xC000_0008 para pasar el valor a mostrar en los displays en el rango -128 a 127, y 0xC000_000C para indicar la letra a mostrar en el quinto display. Es responsabilidad del grupo de trabajo escoger el código que identifica cada una de las tres letras posibles. Recuerde modificar el módulo 'dmem.sv' para mapear correctamente el periférico. Es también responsabilidad del grupo de trabajo establecer la interfaz entre los módulos 'dmem.sv' y 'displays.sv'.
- d. Modifique el módulo 'dmem.sv' para mapear uno de los pulsadores del sistema DE10-Lite en la memoria del sistema. El grupo de trabajo será responsable de asignar la dirección de memoria.
- e. Realice un programa en lenguaje ensamblador para ejecutarse en la versión modificada del procesador ARM, que realice la operación $R = A + B$ dados los siguientes requerimientos:
 - (a) Los operados A y B son valores de 8-bits con signo, los cuales se introducen empleando ocho (8) de los suiches y uno (1) de los pulsadores disponibles en el sistema DE10-Lite. Establezca una estrategia para poder entrar tanto A como B, sabiendo que solo podrá emplear 8 suiches. Tenga en cuenta que mientras se ingresa el operando A o B, los displays deberán mostrar el valor actualmente disponible en los suiches, así como la letra del operando que se está ingresando.

- (b) Una vez ingresados los valores de A y B, se deberá mostrar el resultado de la operación $R = A + B$ en los displays, junto con la letra 'r' indicando que se está visualizando el resultado. Ej: r-120, indica que el resultado de la operación fue -120.
 - (c) Tanto los operandos A y B como el resultado deberán visualizarse en ocho (8) de los leds del sistema DE10-Lite. Lo anterior les permitirá detectar fallos que se pueden presentar en el programa en ensamblador o en el hardware que gestiona los displays de 7 segmentos.
 - (d) El programa deberá hacer uso de al menos dos de las nuevas instrucciones de desplazamiento y rotación: LSL, LSR, ASR y ROR.
 - (e) El programa deberá hacer uso de las instrucciones MOV y CMP.
 - (f) La ejecución del programa deberá ser cíclica, es decir, una vez realizada la operación $R = A + B$, deberá quedar listo para recibir nuevamente los operandos A y B para proceder a realizar otra suma más.
- f. 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) de acuerdo con los lineamientos que recibió durante el tutorial. Realice los respectivos cambios en los archivos 'imem.sv' y 'dmem.sv' para cargar las nuevas instrucciones y datos.
 - g. Cree un testbench para su procesador ARM modificado, junto con el programa realizado, y verifique el funcionamiento de todo el sistema. Consulte con el profesor del laboratorio si tiene dificultades.
 - 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, verifique que la operación $R = A + B$ se ejecuta y visualiza de manera correcta.
 - i. Sustente el trabajo al profesor del laboratorio.

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. **Esquemas de HW:** esquemáticos del hardware del procesador ARM modificado para soportar las nuevas instrucciones y periféricos.
- c. **Simulación:** simulación del programa en ensamblador que realiza la operación $R = A + B$.
- d. **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 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:

p5_primerapellidointegrante1_primerapellidointegrante2_horariolaboratorio.zip.

Ejemplo: si el primer apellido de ambos integrantes es **Plazas** y **Garcia**, respectivamente, y el laboratorio es el Martes 9-12, entonces el archivo debe ser nombrado: *p5_plazas_garcia_m9-12.zip*.

6 Evaluación

La evaluación de la práctica se divide en tres partes: funcionamiento con cumplimiento de requerimientos (40%), sustentación (40%) y código fuente / reporte (20%). La nota del funcionamiento se asigna 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 tres maneras posibles:

- a. Solicitud de cambios al código por parte del profesor de laboratorio.
- b. Un par de preguntas orales que pongan a prueba los conocimientos del estudiante sobre el desarrollo de la práctica.
- c. 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. Finalmente, en caso no se realice entrega del reporte y los códigos, la nota asignada a la práctica de laboratorio será de 0.0 así se haya sustentado.

Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 15 minutos, 8 minutos para revisar la implementación y 7 minutos para preguntas. Es importante tener abierto el Quartus y CPUlator 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/>