

Laboratorio de Electrónica Digital II

Práctica 2: Circuitos Aritméticos para Números Enteros

Unidad Aritmético Lógica

Profesores

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

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

Agosto 29, 2023



Fecha de entrega: Del 5 al 8 de Septiembre de 2023

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

Sustentación: Horario de Laboratorio

Valor Práctica: 6% del curso

1 Introducción

En esta práctica de laboratorio, el grupo de estudiantes implementará un sistema electrónico digital para la realización de operaciones aritméticas y lógicas básicas sobre números enteros. El corazón de la práctica será la Unidad Aritmético Lógica (ALU), la cual se encargará de realizar las operaciones requeridas. La descripción de la ALU junto con la lógica requerida para ingresar los datos y mostrar los resultados será realizada empleando el lenguaje de descripción de hardware SystemVerilog. Se deberá emplear la herramienta de simulación ModelSIM para llevar a cabo una

serie de pruebas que permitan verificar el correcto funcionamiento de la ALU. Finalmente, el grupo de trabajo implementará el diseño en el sistema de desarrollo DE10-Lite, donde se deberá verificar el correcto funcionamiento de todo el sistema.

2 Objetivo de la Práctica

Diseñar e implementar un sistema electrónico digital para la realización de operaciones aritméticas y lógicas básicas con números de 5-bits, empleando el lenguaje de descripción de hardware SystemVerilog y herramientas de desarrollo para FPGAs.

3 Procedimiento

Para el correcto diseño e implementación de la práctica, es necesario leer completamente esta guía. Se sugiere seguir el procedimiento indicado a continuación:

- a. Estudiar la sección 5.2.4 Unidad Aritmético Lógica de 32-bits (ALU), del texto guía *Digital Design And Computer Architecture - ARMEdition*.
- b. Elaborar un diagrama de bloques de alto nivel (*diagrama de bloques principal*) para representar el problema principal junto con las entradas y las salidas del sistema. Este diagrama incluye el bloque que contiene la ALU junto con la lógica adicional requerida para el ingreso de datos y visualización de resultados.
- c. Describir el diseño de la ALU y la lógica adicional usando SystemVerilog y emplear la herramienta de simulación ModelSIM para verificar el correcto funcionamiento del mismo. Llevar a cabo las correcciones pertinentes, si fuese el caso.
- d. Implementar el diseño en la FPGA disponible en el sistema de desarrollo DE10-Lite (disponible en el laboratorio LED) y verificar el correcto funcionamiento del mismo. Llevar a cabo las correcciones pertinentes, si fuese el caso.
- e. Escribir un corto reporte sobre el diseño e implementación de la práctica y subirlo a la plataforma de Ingeni@ junto con los archivos principales de su diseño en Quartus Prime.
- f. Sustentar el desarrollo en el horario de laboratorio correspondiente.

4 Especificaciones

A continuación se indican las especificaciones que el grupo de trabajo deberá considerar para el diseño e implementación del circuito digital, representado en la Fig. 1:

- a. El módulo de la ALU a describir en SystemVerilog debe poder realizar las operaciones de suma, resta, AND y OR entre operandos de N-bits, donde N será un parámetro del módulo.

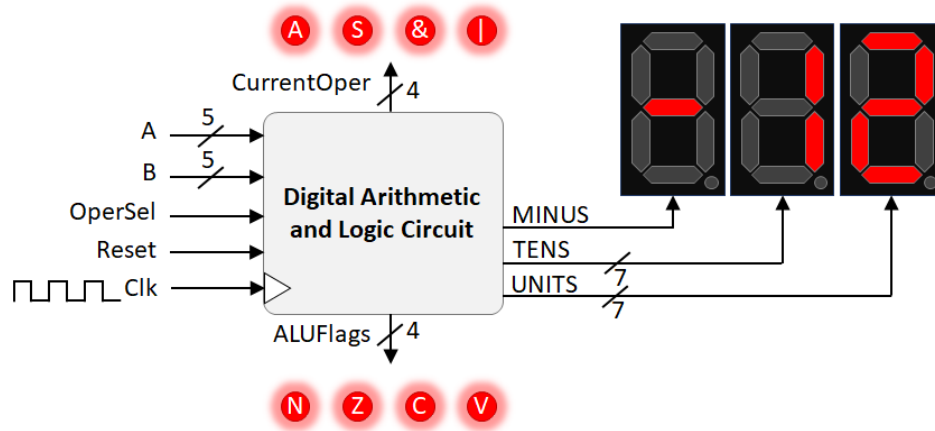


Fig. 1: Diagrama de bloques para el sistema

Para la simulación e implementación se empleará el valor $N = 5$. Tenga en cuenta que la descripción de la ALU solo podrá contar con un sólo sumador para realizar tanto la suma como la resta.

- b. Las entradas **A** y **B** corresponden a los operandos de la operación, de 5-bits cada uno, los cuales serán cargados a través de los suiches de la board DE10-Lite.
- c. La entrada **OperSel** permite seleccionar entre 4 operaciones posibles a realizar: suma, resta, AND y OR. Esta entrada proviene de uno de los pulsadores de la board DE10-Lite. Cada que se presiona el pulsador, el sistema deberá cambiar de operación a realizar: suma \rightarrow resta \rightarrow AND \rightarrow OR \rightarrow suma \rightarrow etc. La salida **CurrentOper** de 4-bits se emplea para indicar cuál es la operación actualmente seleccionada. Estas líneas deberán ser conectadas a cuatro LEDs de la board DE10-Lite para indicar la operación seleccionada actualmente: suma (A: add), resta (S: subtract), AND (&) y OR (|).
- d. Las salidas **TENS** y **UNITS** corresponden cada una a 7 líneas para manejar dos displays de 7 segmentos en la board DE10-Lite, correspondientes a las decenas y unidades del resultado, respectivamente. Una tercera salida llamada **MINUS** deberá ser conectada al segmento **g** de un tercer display para representar el símbolo menos '-' cuando un resultado sea negativo. Tenga en cuenta que para operaciones aritméticas el resultado a mostrar en los displays corresponde a un valor con signo, mientras que para operaciones lógicas, el resultado a mostrar corresponde a un valor sin signo.
- e. La salida **ALUFlags** corresponde a 4-líneas que representan los códigos de condición generados por la ALU: N (Negativo), Z (Cero), C (Acarreo) y V (Overflow). Estas líneas serán conectadas a cuatro LEDs de la board DE10-Lite para la visualización de los *Flags* de la ALU.
- f. Finalmente, las entradas **Clk** y **Reset** son requeridas por el circuito interno que mantiene la operación que actualmente realiza el sistema.

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. **Diseño y Simulación:** diagrama de bloques del sistema y una o varias imágenes que muestren el resultado de la simulación realizada. Tenga en cuenta que la simulación deberá mostrar las 4 operaciones para varios valores de entrada.
- c. **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 proyecto en Quartus Prime como se describe a continuación:

- a. **Reporte:** archivo con extensión .pdf
- b. **Archivos proyecto Quartus:** archivos con extensión .sv, .qpf y .qsf.

El nombre del archivo comprimido deberá tener el siguiente formato:

p2_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: *p2_castano_garcia_m9-12.zip*.

6 Evaluación

La evaluación de la práctica se divide en tres partes: funcionamiento (50%), sustentación (40%) y reporte (10%). 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. 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 implementación y 7 minutos para preguntas. Es importante tener abierto el proyecto y programada la FPGA para cuando el profesor llegue a su puesto de trabajo. No habrá tiempo para hacer correcciones de último momento.

7 Referencias

- a. Quartus Prime Lite Edition
<https://www.intel.com/content/www/us/en/products/details/fpga/development-tools/quartus-prime.html>

- b. Sistema de desarrollo DE10-Lite
<http://de10-lite.terasic.com/>
- c. Tutorial de SystemVerilog
<https://verilogguide.readthedocs.io/en/latest/>