

UNC

Universidad
Nacional
de Córdoba

Cátedra de Arquitectura de Computadoras

Trabajo Práctico N° I - ALU

Integrantes:

López Gastón
Kleiner Matías

Fecha de entrega:

29 de agosto de 2018

Introducción

En toda implementación de software hay algunos componentes de hardware sumamente necesarios para su funcionamiento. En el siguiente informe se describe el diseño, la implementación y el testing de uno de estos elementos: **ALU**. La ALU (Unidad Aritmética Lógica) es un circuito combinacional digital que calcula operaciones aritméticas así como operaciones lógicas entre argumentos.

Para dicho diseño e implementación se usará una FPGA, la cual es un dispositivo programable que contiene bloques de lógica que, al interconectarlos, genera una funcionalidad. La mencionada interconexión y funcionalidad se configuran mediante un lenguaje de descripción de hardware. Se mostrará a lo largo del informe esquemáticos del diseño, como así también resultados obtenidos del funcionamiento en la FPGA.

Esquema de la consigna

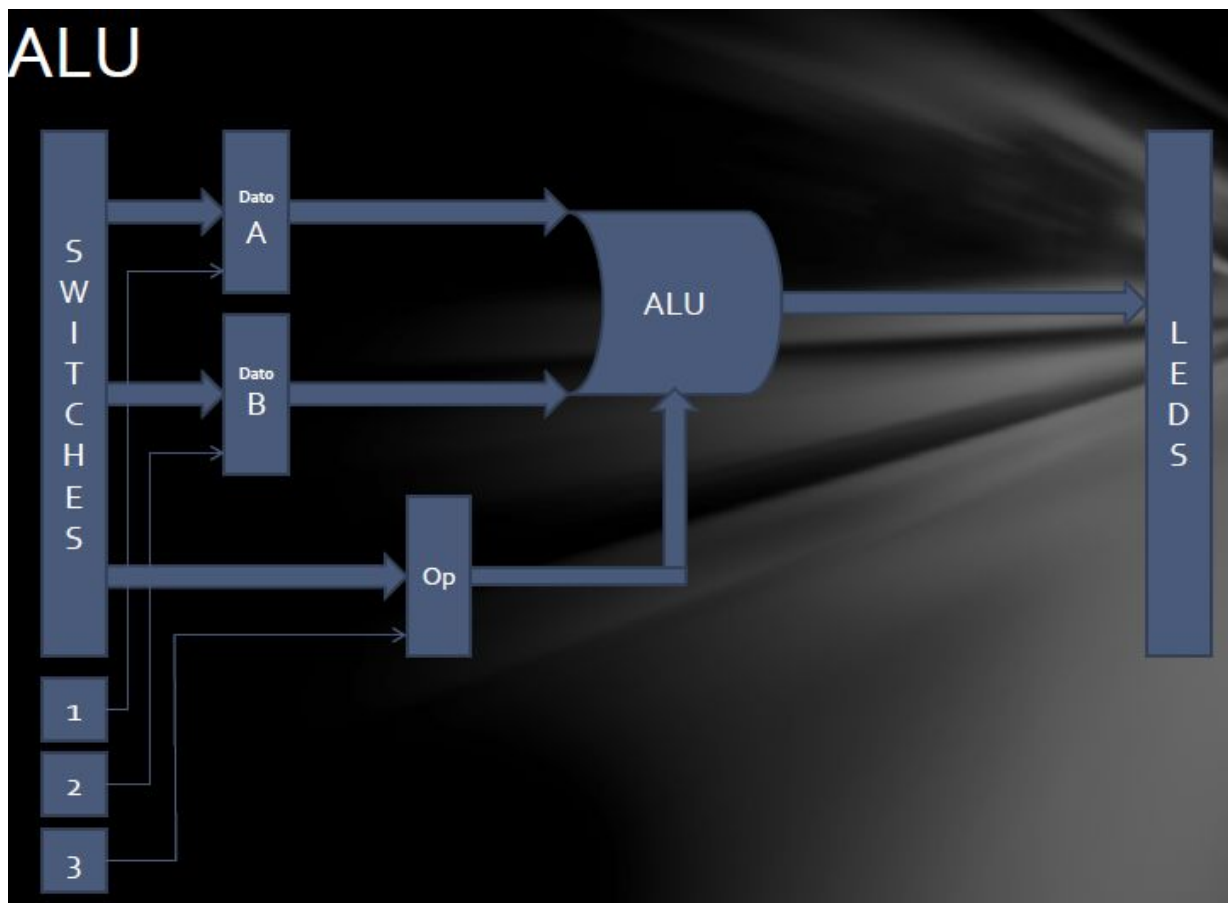


Figura 1: TP1.

Herramientas y elementos utilizados

Entre los dispositivos disponibles para la implementación de la ALU se seleccionó la placa de desarrollo ARTY, que puede verse en la siguiente figura.

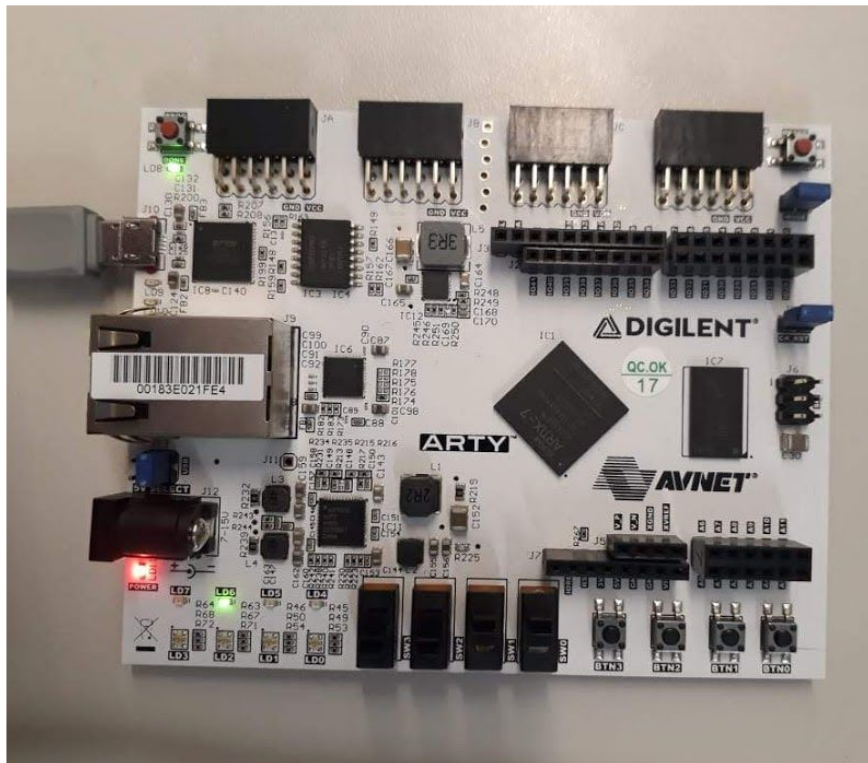


Figura 2: FPGA Utilizada para el desarrollo del proyecto.

La principal razón de la selección fue que esta placa es una de las que soportan la interfaz de desarrollo Vivado, sobre la cual poseemos algo de experiencia previa. Cabe destacar que la FPGA que contiene la placa es una Artix-7 de Xilinx.

Diseño - Módulo ALU

Este módulo contiene el desarrollo de las funcionalidades de la ALU, las cuales consisten en un set de operaciones sobre dos entradas (signadas), y un selector de la operación a realizar sobre dichas entradas. Es un módulo puramente combinacional, razón por la cual no contiene entrada de clock y sus salidas dependen directamente de los cambios en sus entradas.

Cabe destacar que la ALU cuenta con tres entradas y una salida, todas de ancho de palabra parametrizable y con valores por defecto de 4 bits. Las operaciones de la misma están basadas en el set de instrucciones del MIPS IV y son:

- **ADD.** (1000).
- **SUB.** (1010).
- **AND.** (1100).
- **OR.** (1101).
- **XOR.** (1110).
- **SRA** (Shift Right Arithmetic). (0011).
- **SRL** (Shift Right Logic). (0010).
- **NOR.** (1111).

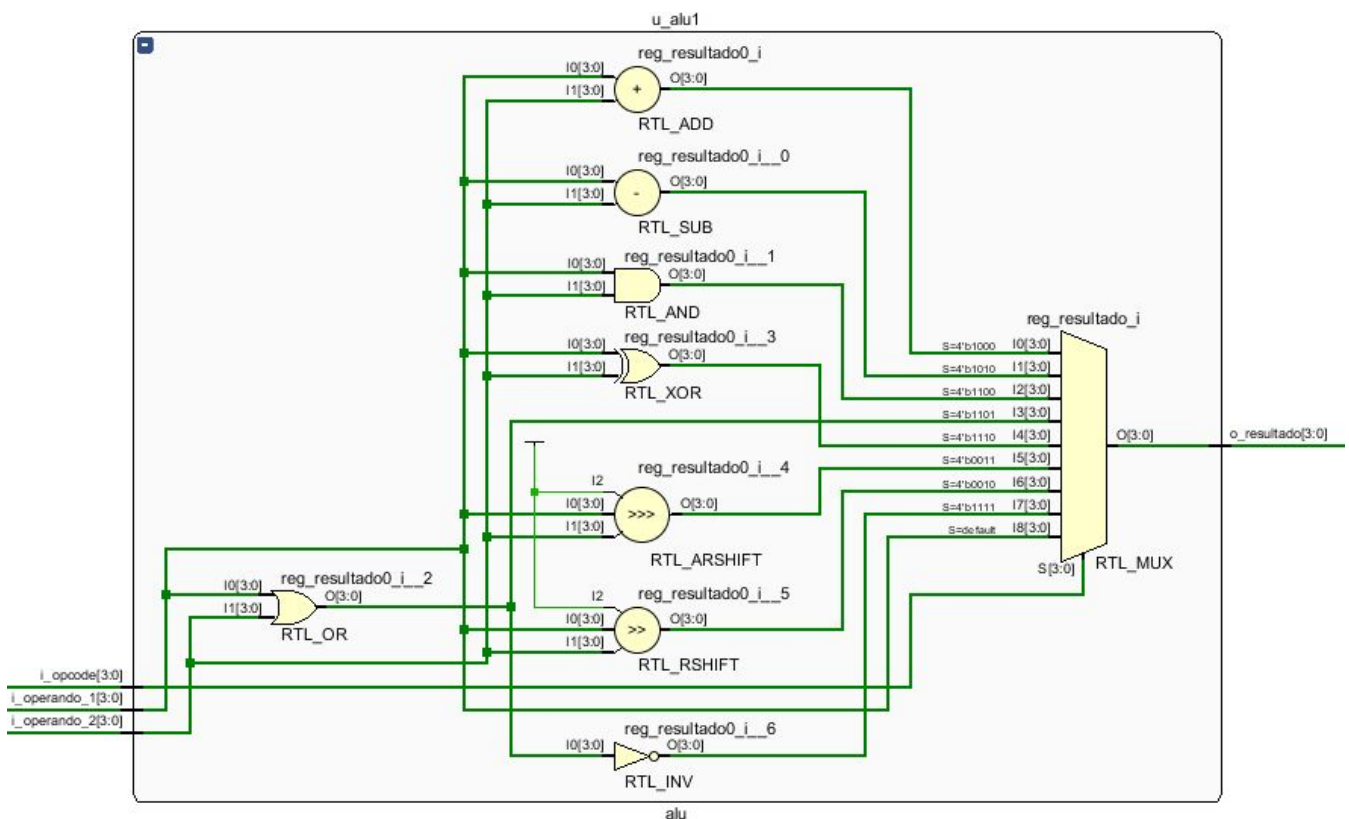


Figura 3: Esquemático RTL del módulo ALU.

Diseño - Módulo Configurador

Este módulo contiene tres registros como salida: dato A, dato B y OPCODE. Se trata de un circuito secuencial, y es por ello que se tiene como entrada la señal de clock. A su vez, otras entradas a dicho módulo son la señal de reset (para poner en un estado inicial los valores de los registros), los botones y los switches que conforman el bus de entrada. La mencionada señal de reset proviene del botón de reset de la placa (hard reset). Por último, cabe destacar que la función de este

circuito es la de setear el valor binario que conforman los switches en los registros correspondientes de acuerdo a la combinación de los botones. Es decir, si los botones forman el valor binario:

- 001 = El valor de los switches se carga en dato A.
- 010 = El valor de los switches se carga en OP CODE (código de operación).
- 100 = El valor de los switches se carga en dato B.
- Otro caso = Sigue todo igual.

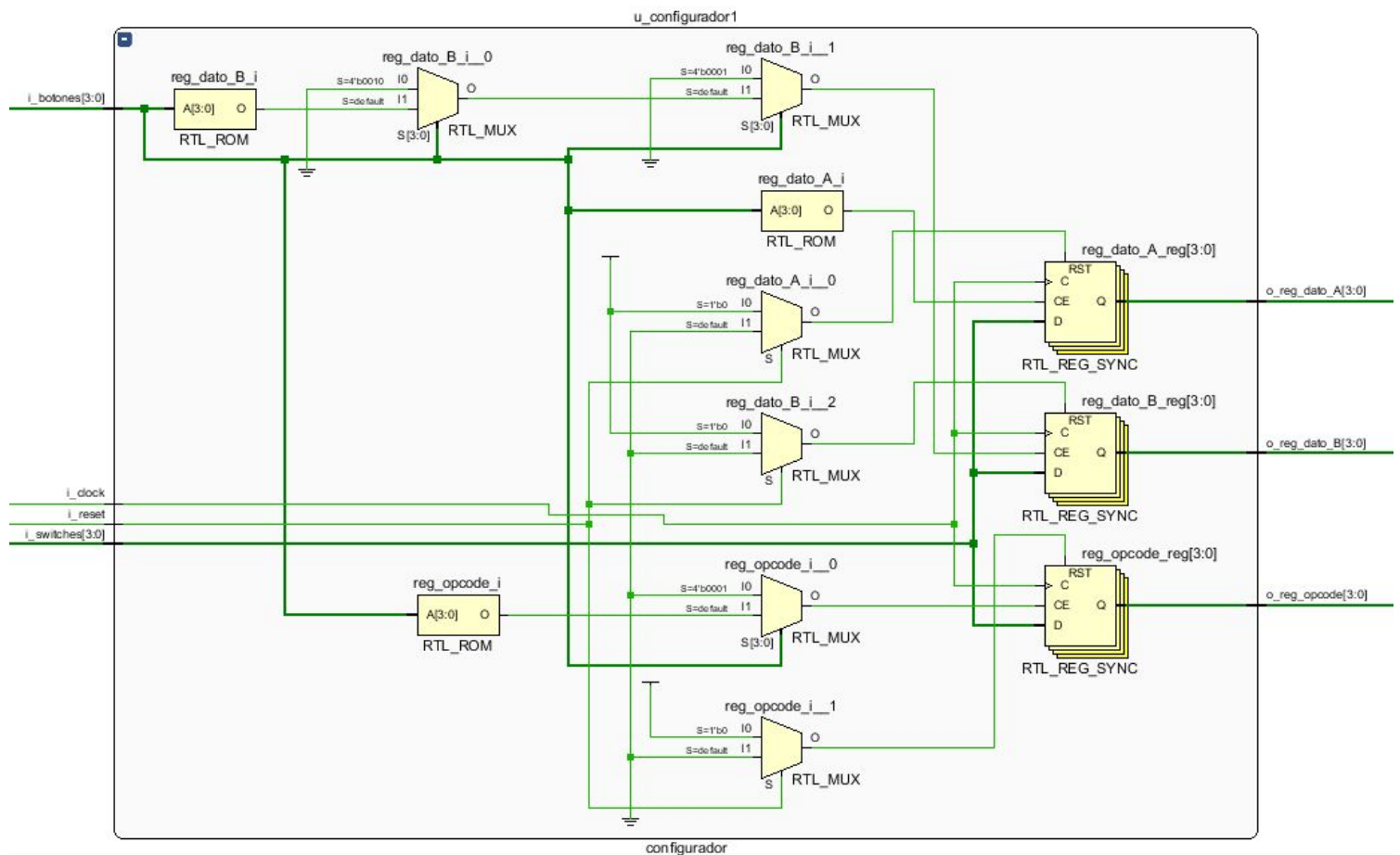


Figura 4: Esquemático RTL del módulo Configurador.

Diseño - Módulo TOP

Es el módulo encargado de instanciar los dos módulos descritos anteriormente y conectarlos entre sí y con la pines de la placa correspondientes. Es importante destacar que la ALU usará los registros del módulo configurador para realizar la operación solicitada.

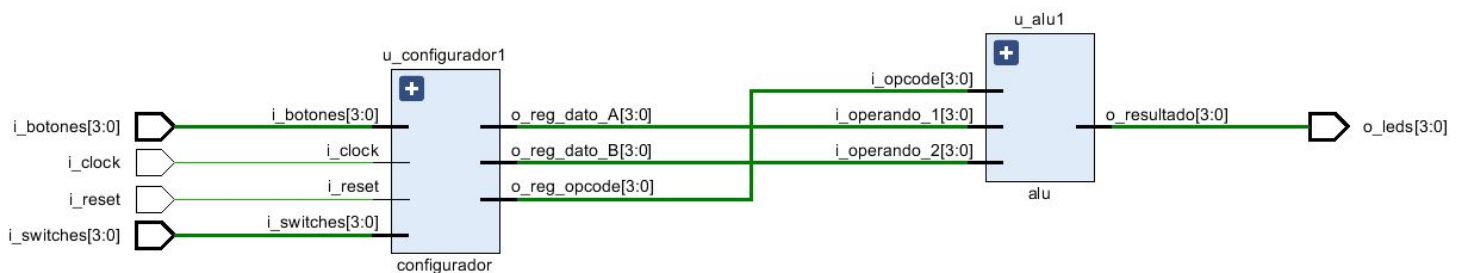


Figura 5: Esquemático RTL del módulo TOP.

Simulaciones de los módulos

Para comprobar el correcto funcionamiento de los dos primeros módulos mencionados, se desarrollaron tests benches desde la herramienta, donde se comprobaron los siguientes casos:

- Correcto funcionamiento del reset. (Módulo configurador).
- Correcto cargado de los valores en los registros según la combinación de los botones. (Módulo configurador).
- Resultados correctos al setear operandos y códigos de operación. Se comprobó para cada una de las operaciones e inclusive para un valor de opcode que no coincida con ninguno de los detallados anteriormente (Módulo ALU).

Implementación en FPGA

Para la implementación, se instanció el módulo ALU junto al módulo configurador dentro del módulo top, sobre el cual se declaran como entradas y salidas puertos que se mapean con pines de la FPGA a utilizar en un archivo denominado constraints. El mismo consiste de un archivo con extensión .xdc.

- **i_clock.** (CLK100MHz).
- **ck_rst.** (Hard reset).
- **i_switch[3:0].** Bus de entrada.
- **i_btn [3:0].** (Para seleccionar en cual registro se mapea el valor de los switches).
- **o_leds[3:0].** (Resultado de la operación de la ALU).

Explicación de warnings.

En el desarrollo del práctico, apareció el siguiente warning al generar el bitstream que se debe cargar en la FPGA correspondiente:

“[DRC CFGBVS-1] Missing CFGBVS and CONFIG_VOLTAGE Design Properties: Neither the CFGBVS nor CONFIG_VOLTAGE voltage property is set in the current_design. Configuration bank voltage select (CFGBVS) must be set to VCCO or GND, and CONFIG_VOLTAGE must be set to the correct configuration voltage, in order to determine the I/O voltage support for the pins in bank 0. It is suggested to specify these either using the 'Edit Device Properties' function in the GUI or directly in the XDC file using the following syntax:

```
set_property CFGBVS value1 [current_design]  
#where value1 is either VCCO or GND
```

```
set_property CONFIG_VOLTAGE value2 [current_design]  
#where value2 is the voltage provided to configuration bank 0
```

Refer to the device configuration user guide for more information.”

Se buscó información sobre este warning, dando como resultado un conocimiento acerca del origen del mismo. Dicho warning ocurre porque se recomienda especificar el nivel de tensión de los pines I/O del bank 0 de la FPGA en el constraint (ver links al final de esta sección). Dicho nivel de tensión por defecto es de 3.3 V. Además cabe destacar que para poder asociar la placa ARTY con el proyecto en Vivado se debió descargar un paquete de software aparte. Un nivel de tensión erróneo puede causar daños en la placa. Teniendo en cuenta lo mencionado, se agregaron las líneas correspondientes en el constraints, lo que arrojó warnings debido a una incompatibilidad de dichas líneas con el paquete de software asociado a la ARTY, que ya posee los niveles de tensiones por defecto en caso de que no se especifique nada. Es por esto que no se ha corregido dicho warning debido a que no afecta el funcionamiento del proyecto, además de que las únicas input/output son las integradas a la placa (switches, botones y leds).

Links consultados:

- <https://reference.digilentinc.com/reference/programmable-logic/arty/reference-manual?redirect=1>
- http://www.iamElectronic.com/products/T0005_Artix-7_FPGA_Module/datasheet/

Conclusión

Mediante un desarrollo sencillo de una ALU, se pudo comprender el principio de funcionamiento de una FPGA. Una vez terminado el práctico, a través de la herramienta Verilog se obtuvieron las especificaciones de la utilización de la FPGA en términos de LUT y FF como se introdujo en clase.

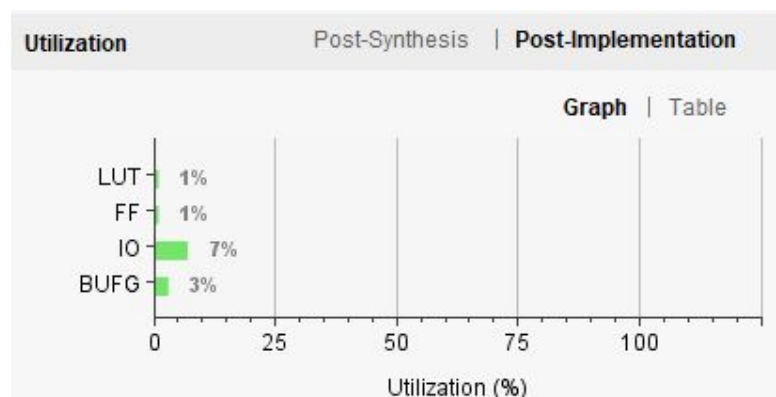


Figura 6: Recursos utilizados para el desarrollo del proyecto.