

Trabajo Práctico N° 1 - ALU

**Profesores:**

Pereyra Martin
Rodriguez Santiago

Alumnos:

Aichino Ignacio Daniel 40673680 (ignacio.aichino@mi.unc.edu.ar)
Vignolo Gabriel Enrique 39080905 (gabriel.vignolo@mi.unc.edu.ar)

Introducción

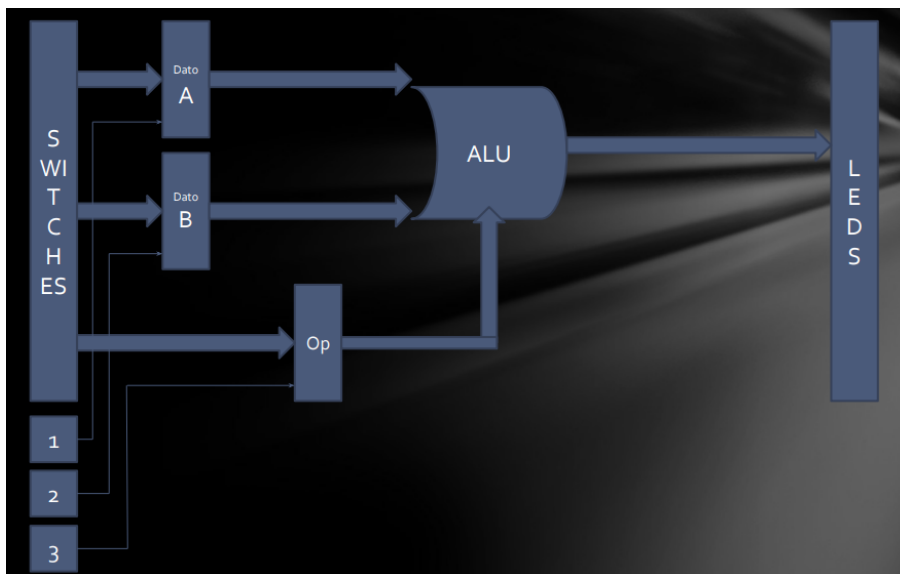
El objetivo de este documento es configurar una FPGA Basys 3 que tendrá implementado el desarrollo de una Unit Arithmetic Logic. Se utiliza el lenguaje Verilog en el entorno de desarrollo Vivado.

Nuestra ALU debe ser parametrizable para poder ser utilizada en futuros proyectos.

Este proyecto cuenta con un módulo que implementa la ALU y un módulo Top que hace uso de la misma. Además se valida el desarrollo por medio de un Test Bench.

Desarrollo

A continuación se ve el diagrama de bloques:



La Unit Arithmetic Logic resuelve las siguientes operaciones:

Operación	Opcode
ADD	100000
SUB	100010
AND	100100
OR	100101
XOR	100110
SRA	000011
SRL	000010
NOR	100111

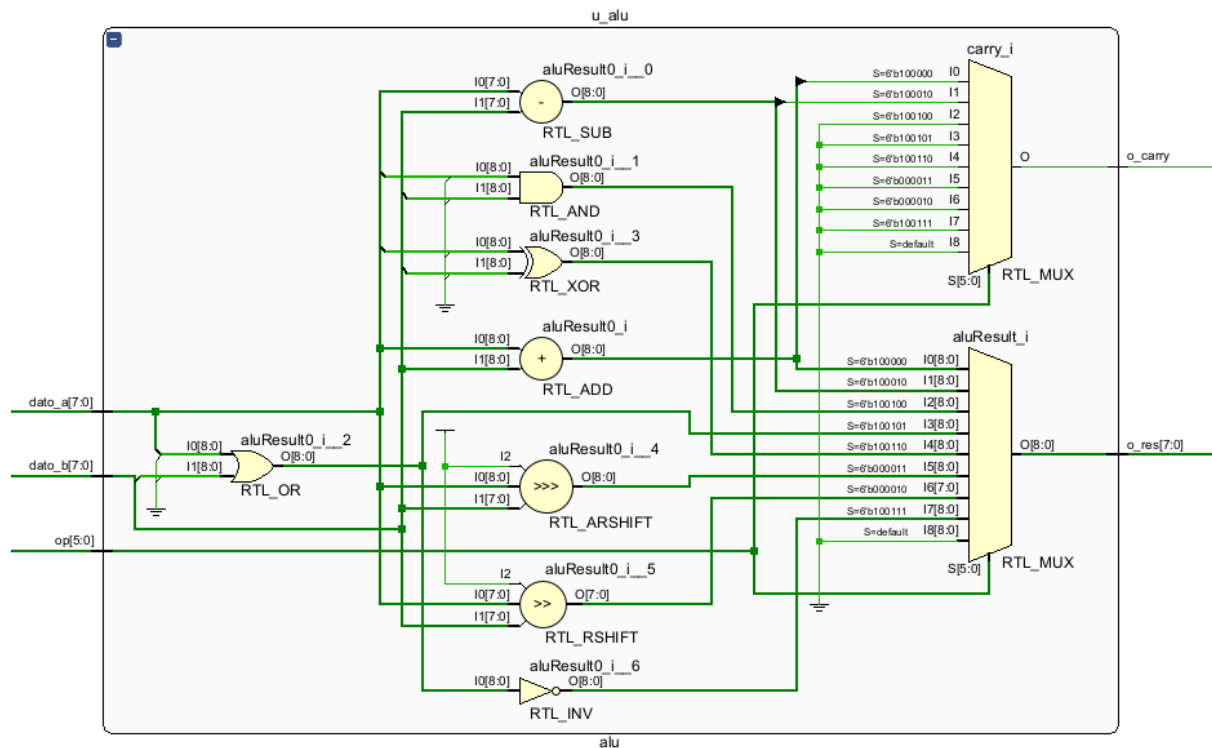
Módulo ALU

Este módulo utiliza una lógica combinacional con un bloque Always y un switch case para ejecutar el código de operación apropiado. Se dispone de tres datos de entradas parametrizables, y estos son según el diagrama de bloques el **datoA**, **datoB** y **Op**. A su vez contamos con dos salidas **o_res** para mostrar el resultado en los leds y **o_carry** para mostrar el carry.

Tendremos un registro del tamaño de un byte para el resultado de un bit extra que será el carry.

El resultado se presenta por el uso de los Led's por lo que para ello usamos el byte del resultado y para el carry usaremos otro Led tomando el bit más significativo del registro mencionado.

En la siguiente imagen se puede observar el diagrama esquemático del módulo ALU:



Módulo TOP

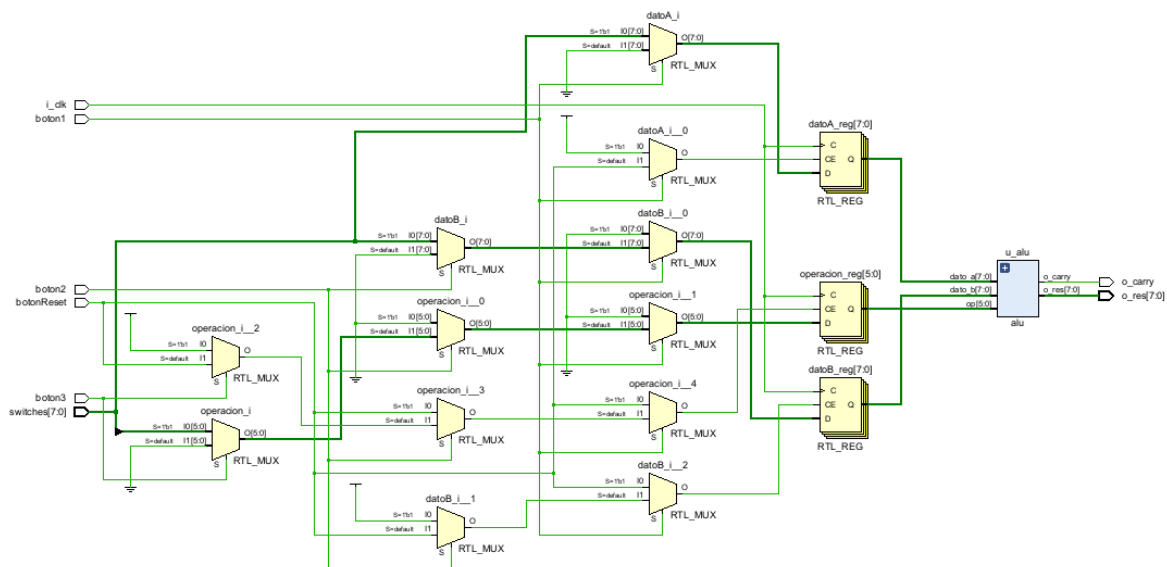
Este módulo se utiliza para instanciar al módulo de la ALU y realizar toda la lógica del pulsado de los botones para cargar los datos y para cargar la operación a realizar.

El cargado de los datos A y B y de la operación se realiza mediante un always secuencial al que va a ingresar en el flanco de subida del clock.

Se implementan las siguientes entradas:

1. **switches:** Es el valor que se carga en los switches de la FPGA que luego será utilizado para cargar en el dato A, dato B y operación.
2. **boton1:** Es el botón utilizado para cargar el valor de los switches en el dato A.
3. **boton2:** Es el botón utilizado para cargar el valor de los switches en el dato B.
4. **boton3:** Es el botón utilizado para cargar el valor de los switches en la operación.
5. **botonReset:** Es un botón utilizado para limpiar los registros.
6. **i_clk:** Es el clock utilizado en el always secuencial para cargar el dato A, dato B y operación.

A continuación se puede ver el diagrama esquemático del módulo TOP:



Running Simulacion

A continuación se presenta una imagen con la ejecución del modulo ALU. Se ejecutó la simulación Post Synthesis Timing con una operación de suma con carry.

Para este ejemplo de ejecución se utiliza los siguientes datos:

- Dato A = FF
- Dato B= FF
- Operacion = ADD (20)

Se puede observar que el resultado es el esperado (**FE**) y además podemos visualizar el bit en alto para el **carry**.

Para realizar las pruebas con todas las operaciones, este proyecto cuenta con el TB que valida las mismas.

