

Implementación de ALU en HDL

Aagaard Martin - Navarro Matias
Universidad Nacional de Córdoba

11 de septiembre de 2019

Índice

1. Introducción	2
2. Implementación del practico	2
2.1. Desarrollo del módulo ALU	2
2.2. Simulaciones del módulo	3
2.3. Implementación en FPGA	4
3. Conclusión	6

1. Introducción

En toda implementación de software hay algunos componentes de hardware sumamente necesarios para su funcionamiento. En el siguiente informe se describirá 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.

2. Implementación del práctico

Entre los dispositivos disponibles para la Implementación de la ALU se seleccionó la placa de desarrollo Basys 3. La principal razón de la selección fue que esta placa es el único modelo disponible y que soporta la interfaz de desarrollo Vivado. La FPGA sobre la que se trabajará es la Artix-7 de Xilinx.

2.1. Desarrollo del módulo ALU

Este módulo contiene el desarrollo de las funcionalidades de la ALU, las cuales consisten en un set de operaciones sobre dos registros, y un selector sobre estas que define la salida del módulo.

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.

La ALU cuenta con tres entradas y una salida, con un ancho de palabra parametrizable y con valores por defecto de 8 bits.

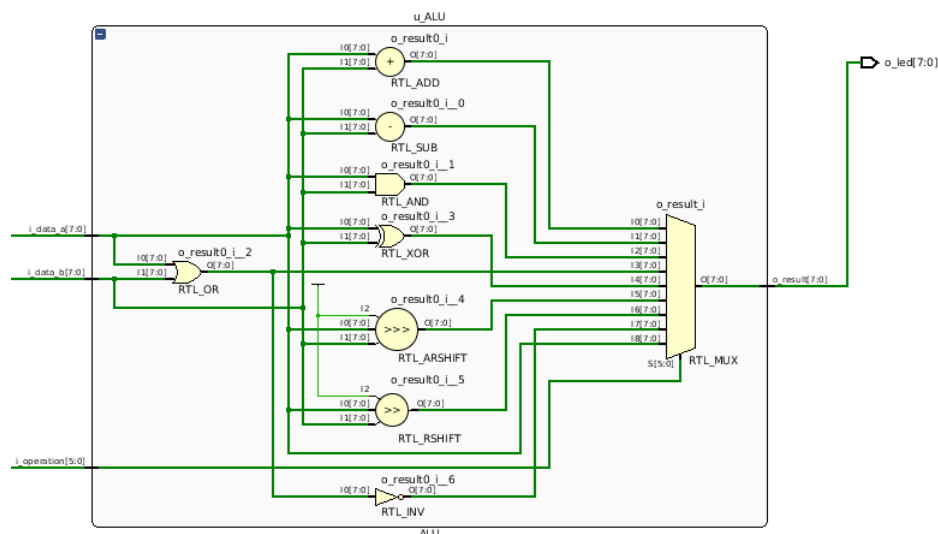


Figura 1: Estructura RTL del módulo ALU obtenida de la herramienta Verilog

2.2. Simulaciones del módulo

Para comprobar el correcto funcionamiento del módulo, se desarrollaron varios test desde la herramienta, donde se puede ver la respuesta de las señales a diferentes estímulos.

- Simulación de la operación de suma: Se simula incrementando los operandos A y B en cada ciclo de clock para comprobar varios resultados.

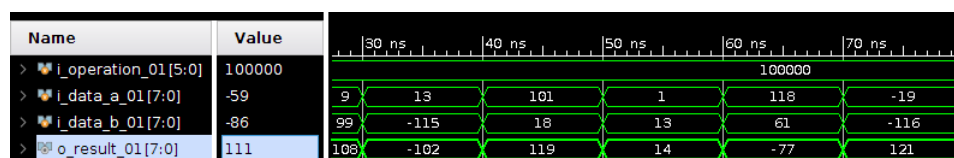


Figura 2: Simulación de la operación de suma

- Simulación de operación AND: Se incrementa el operando A y decrementa B en el test en cada ciclo de clock para obtener diferentes resultados

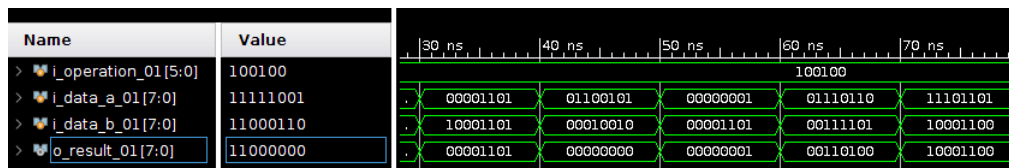


Figura 3: Simulación de la operación AND

2.3. Implementación en FPGA

Para la implementación se instancia el módulo ALU dentro de un módulo top, sobre el cual se declaran como entradas y salidas los pines de la FPGA a utilizar.

Para mapear los pines de la FPGA con las entradas y salidas del módulo, se importo un recurso llamado **constraints**, el cual consiste en un archivo con extensión xdc correspondiente a la placa de desarrollo a utilizar. En el archivo de **constraints** se encuentran todas las líneas comentadas y se le deben sacar los comentarios a las que se van a utilizar como E/S en el módulo top. En nuestro proyecto se utilizaron las siguientes E/S:

- i_clk (Clock 100 MHz)
- i_sw[7:0] para entradas
- i_btn [2:0] para seleccionar las entradas

Para evitar latches en este módulo se trabajó con una señal de clock, la cual es encargada de sincronizar las entradas al módulo ALU y la carga de los registros sobre los cuales esta va operar.

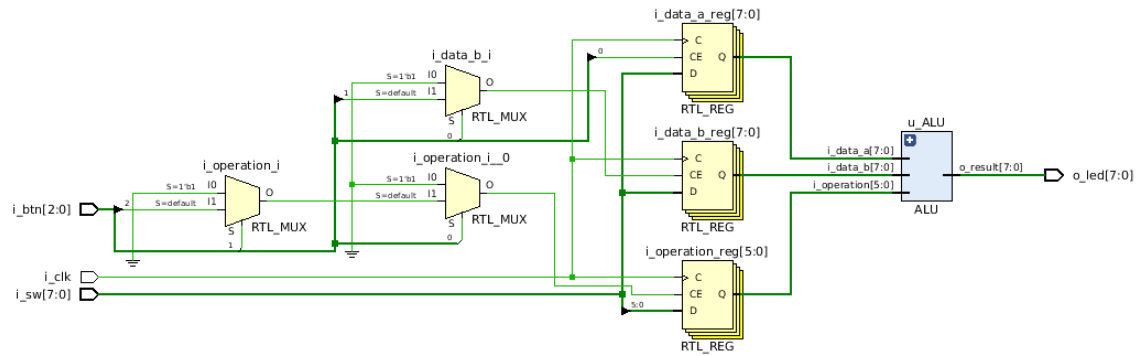


Figura 4: Estructura RTL del módulo top obtenida de la herramienta Verilog

3. 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 del la FPGA en términos de LUT y FF como se introdujo en clase

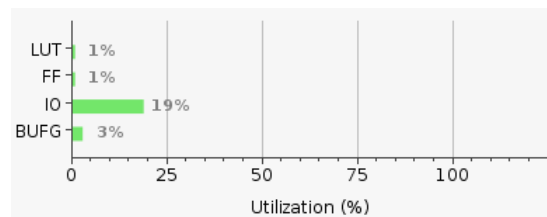


Figura 5: *Utilización de la FPGA*