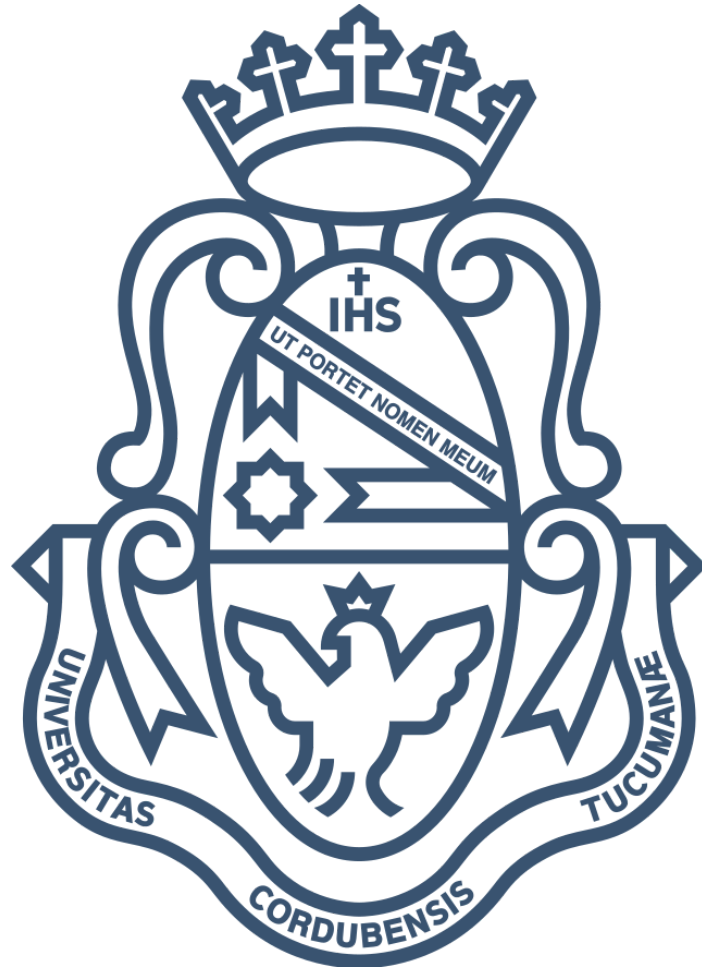


Universidad Nacional de Córdoba

Facultad de Ciencias Exactas Físicas y Naturales



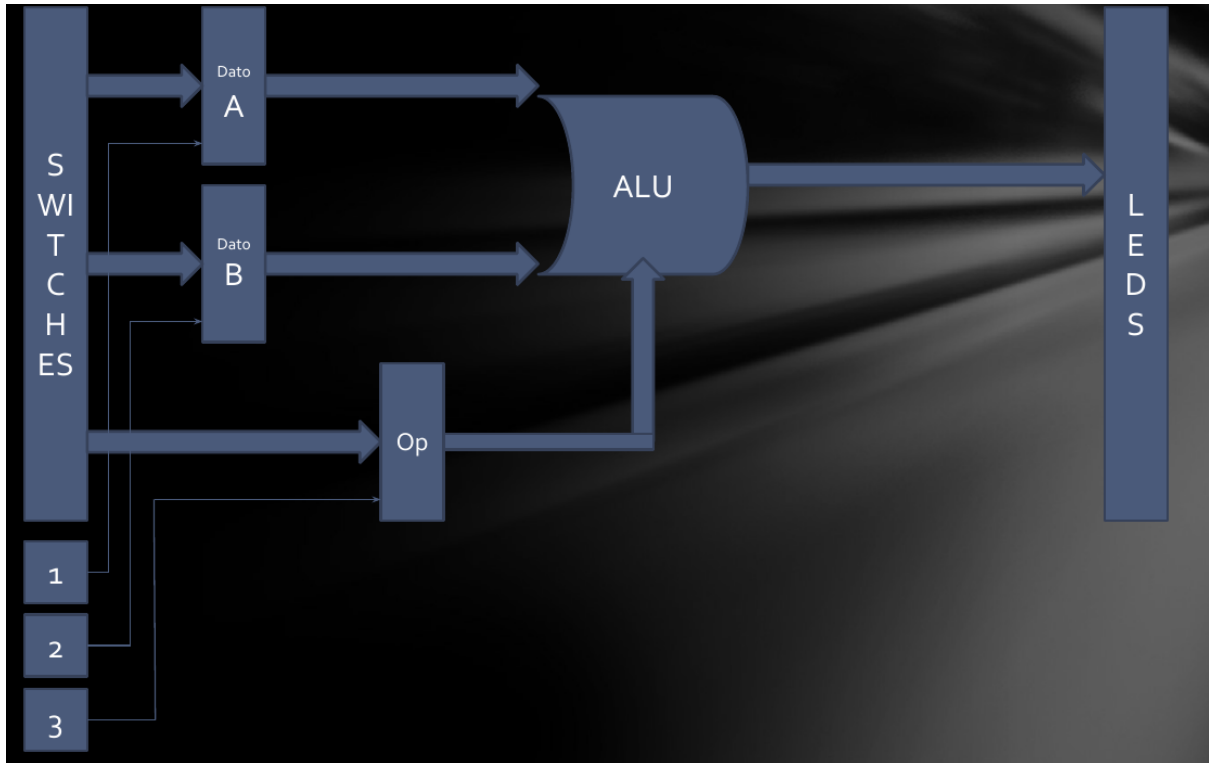
Arquitectura de Computadoras - Trabajo Práctico 1

Integrantes:

- Moyano Seminare, Carlos Alejo DNI: 40.802.646
- Zanotti, Emiliano DNI: 40.678.754

Introducción

En este trabajo práctico se desarrolló una ALU en Verilog, utilizando una placa Basys 3. La misma cuenta con las operaciones ADD, SUB, AND, OR, XOR, SRA, SRL y NOR.



Esquema de diseño para el TP1

Se declararon 3 botones que corresponden al dato A, B y al operando. Además se utilizan switches para mostrar las entradas, además del bit de carry y cero, y LEDs que muestran el resultado obtenido.

Funcionamiento:

Al presionar uno de los botones, el valor que contienen los switches se guardará en un operando o en la operación dependiendo del botón que se presionó. Luego de ingresar el OP se mostrará en los leds el resultado, como así también el bit de carry y cero.

Diagrama:

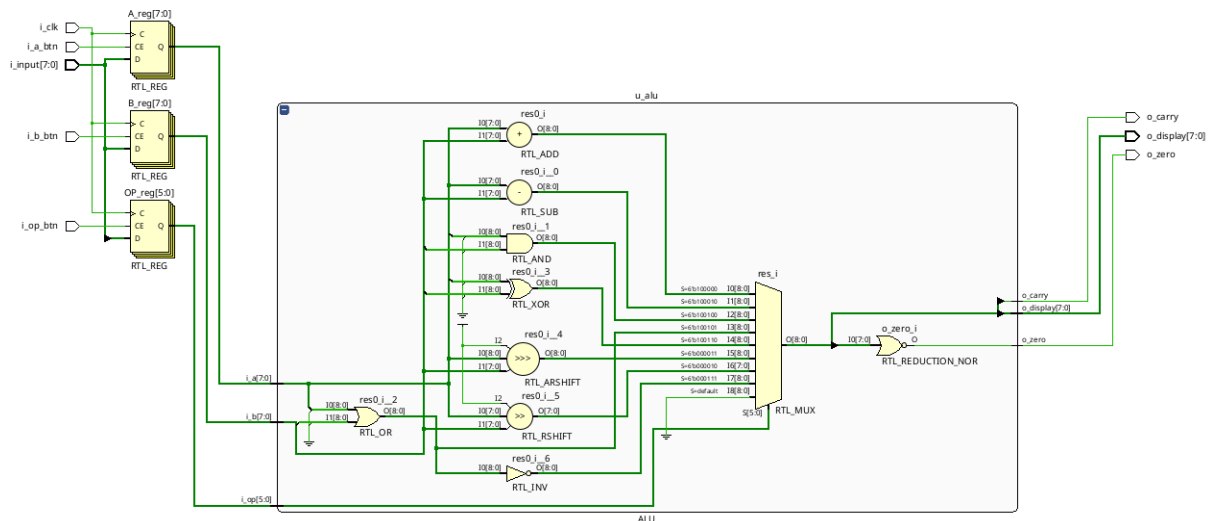


Figura con los componentes del sistema, así como sus salidas y entradas.

Testbench:

Se adjunta el testbench utilizado para probar todas las operaciones soportadas por la ALU, así como el funcionamiento del bit carry y del bit cero.

```
`timescale 1ns / 1ps

module testbench;

    localparam SIZE_OP = 6;
    localparam SIZE_BUS = 8;

    reg [SIZE_BUS - 1:0] i_input;
    reg i_a_btn, i_b_btn, i_op_btn, i_clk = 0;
    wire [SIZE_BUS - 1:0] o_display;
    wire o_zero, o_carry;

    localparam ADD = 6'b100000;
    localparam SUB = 6'b100010;
    localparam AND = 6'b100100;
    localparam OR = 6'b100101;
    localparam XOR = 6'b100110;
    localparam SRA = 6'b000011;
    localparam SRL = 6'b000010;
    localparam NOR = 6'b000111;

    top # (
        .SIZE_OP(SIZE_OP),
```

```

        .SIZE_BUS(SIZE_BUS)
    ) u_top
    (
        .i_input(i_input),
        .i_a_btn(i_a_btn),
        .i_b_btn(i_b_btn),
        .i_op_btn(i_op_btn),
        .i_clk(i_clk),
        .o_display(o_display),
        .o_zero(o_zero),
        .o_carry(o_carry)
    );

initial begin
    i_input = 8'h05; //Primer operando: 05
    #5
    i_a_btn = 1;
    #5
    i_a_btn = 0;
    #5
    i_input = 8'h07; //Segundo operando: 07
    #5
    i_b_btn = 1;
    #5
    i_b_btn = 0;
    #5
    i_input = ADD; //Suma. Valor esperado: 0C
    #5
    i_op_btn = 1;
    #5
    i_op_btn = 0;
    #5
    i_input = SUB; //Resta. Valor esperado: FE ; Carry = 1
    #5
    i_op_btn = 1;
    #5
    i_op_btn = 0;
    #5
    i_input = AND; //Valor esperado: 05
    #5
    i_op_btn = 1;
    #5
    i_op_btn = 0;
    #5
    i_input = OR; //Valor esperado: 07
    #5

```

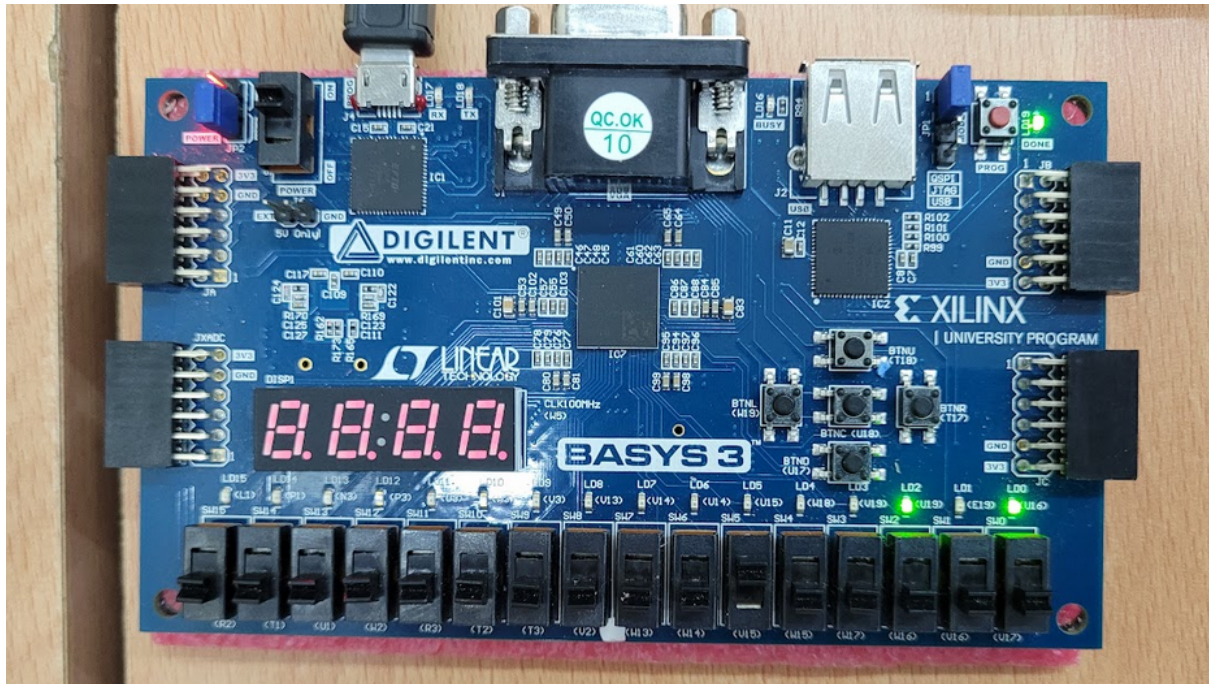
```
i_op_btn = 1;
#5
i_op_btn = 0;
#5
i_input = XOR; //Valor esperado: 02
#5
i_op_btn = 1;
#5
i_op_btn = 0;
#5
i_input = NOR; //Valor esperado: FE
#5
i_op_btn = 1;
#5
i_op_btn = 0;

//OPERACIONES SHIFT
i_input = -8'h0F; //Primer operando: F1
#5
i_a_btn = 1;
#5
i_a_btn = 0;
#5
i_input = 8'h02; //Segundo operando: 02
#5
i_b_btn = 1;
#5
i_b_btn = 0;
#5
i_input = SRA; //Valor esperado: FC
#5
i_op_btn = 1;
#5
i_op_btn = 0;
#5
i_input = SRL; //Valor esperado: 3C
#5
i_op_btn = 1;
#5
i_op_btn = 0;

//PRUEBA CERO
i_input = -8'h05; //Primer operando: FB
#5
i_a_btn = 1;
```


Prueba en hardware

Una vez testeado el código en simulación, se puede flashear en la placa y validar su funcionamiento real en el hardware.



Basys 3 mostrando el resultado de la suma entre 3 y 2

Conclusiones

Se logró implementar una ALU en Verilog, capaz de realizar operaciones aritméticas básicas y se logró probar su funcionamiento en hardware real.

Anexo

Repositorio del proyecto: <https://github.com/alejomoyano/alu>