Martínez Coronel Brayan Yosafat

CÓDIGO VHDL

SUMADOR DE 1 BIT

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;
entity sum_1bit is
    Port ( a, b, cin : in STD_LOGIC;
           s, cout : out STD_LOGIC);
end sum_1bit;

architecture Behavioral of sum_1bit is
begin
s <= a xor b xor cin;
cout <= (a and b) or (a and cin) or (b and cin);
end Behavioral;
```

ALU DE 1 BIT

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity alu_1bit is
    Port ( a, b, s_a, s_b, cin : in STD_LOGIC;
           alu_op : in STD_LOGIC_VECTOR (1 downto 0);
           r, cout : out STD_LOGIC);
end alu_1bit;

architecture Behavioral of alu_1bit is
component sum_1bit is
    Port ( a, b, cin : in STD_LOGIC;
           s,cout : out STD_LOGIC);
end component;

signal aux_b, aux_a, aux_and, aux_or, aux_xor, aux_sum : STD_LOGIC;
begin
    aux_a <= (a and not s_a) or (s_a and not a);
    aux_b <= (b and not s_b) or (s_b and not b);

    aux_and <= aux_a and aux_b;
    aux_or <= aux_a or aux_b;
    aux_xor <= aux_a xor aux_b;
    sum : sum_1bit port map (
        a => aux_a,
        b => aux_b,
        cin => cin,
        s => aux_sum,
        cout => cout
    );

    process (aux_and, aux_or, aux_xor, aux_sum, alu_op)
    begin
        case alu_op is
            when "00" => r <= aux_and;
            when "01" => r <= aux_or;
            when "10" => r <= aux_xor;
            when others => r <= aux_sum;
        end case;
    end process;

end Behavioral;
```

ALU DE 4 BITS

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity alu_4bits is
    Port ( A, B, op : in STD_LOGIC_VECTOR (3 downto 0);
           OV, Z, C, N : out STD_LOGIC;
           R : inout STD_LOGIC_VECTOR (3 downto 0)
);
end alu_4bits;

architecture Behavioral of alu_4bits is
component alu_1bit is
    Port ( a, b, s_a, s_b, cin : in STD_LOGIC;
           alu_op : in STD_LOGIC_VECTOR (1 downto 0);
           r, cout : out STD_LOGIC);
end component;

signal SEL : STD_LOGIC_VECTOR (3 downto 0);
signal carry : STD_LOGIC_VECTOR (4 downto 0);
begin
    carry(0) <= op(2);
    ciclo: for j in 0 to 3 generate
        comp : alu_1bit port map(
            a => A(j),
            b => B(j),
            s_a => op(3),
            s_b => op(2),
            cin =>  carry(j),
            alu_op => op (1 downto 0),
            r => R(j),
            cout => carry(j+1)
        );
    end generate;

    with op select
        OV <= carry(3) xor carry(4) when "0011",
            carry(3) xor carry(4) when "0111",
            '0' when others;

    with op select
        C <= carry(4) when "0011",
            carry(4) when "0111",
            '0' when others;

    N <= R(3);
    Z <= not (R(0) or R(1) or R(2) or R(3));
end Behavioral;
```

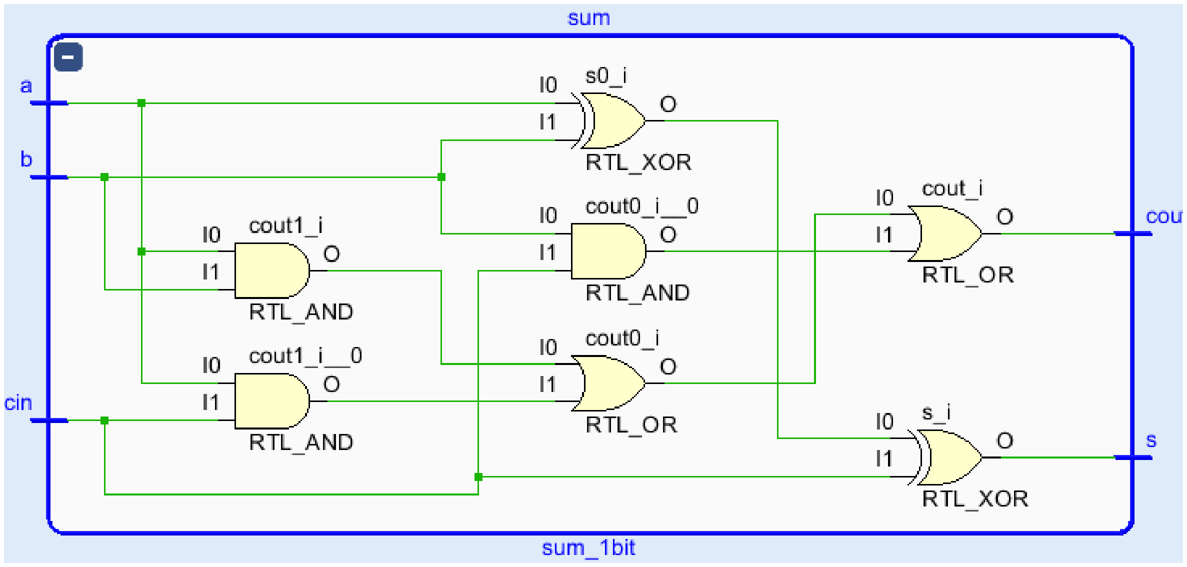
CÓDIGO DEL TEST-BENCH

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity t_alu_4bits is
--  Port ( );
end t_alu_4bits;

architecture Behavioral of t_alu_4bits is
component alu_4bits is
    Port ( A, B, op : in STD_LOGIC_VECTOR (3 downto 0);
           OV, Z, C, N : out STD_LOGIC;
           R : inout STD_LOGIC_VECTOR (3 downto 0)
);
end component;
```

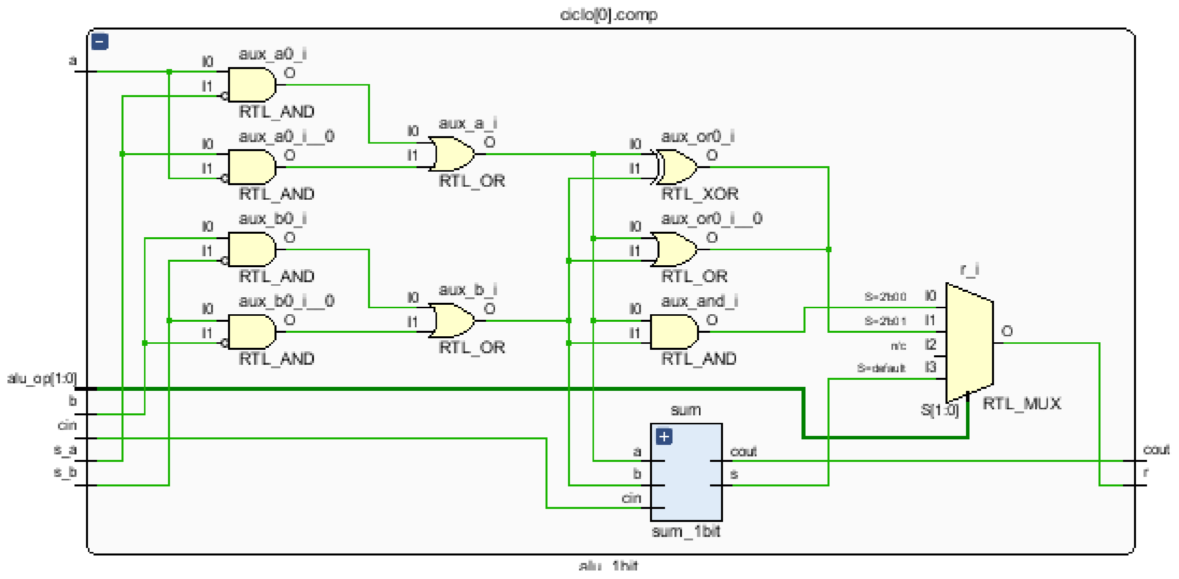
```
signal A, B, op : STD_LOGIC_VECTOR (3 downto 0);
signal OV, Z, C, N : STD_LOGIC;
signal R : STD_LOGIC_VECTOR (3 downto 0);
begin

    test : alu_4bits port map (
        A => A,
        B => B,
        op => op,
        OV => OV,
        Z => Z,
        C => C,
        N => N,
        R => R
    );

    process
    begin
    A <= "0101"; -- 5
    B <= "1110"; -- -2

    op <= "0011"; -- +
    wait for 50 ns;

    op <= "0111"; -- -
    wait for 50 ns;

    op <= "0000"; -- and
    wait for 50 ns;

    op <= "1101"; -- nand
    wait for 50 ns;

    op <= "0001"; -- or
    wait for 50 ns;

    op <= "1100"; -- nor
    wait for 50 ns;

    op <= "0010"; -- xor
    wait for 50 ns;
    op <= "1010"; --xnor
    wait for 50 ns;

    B <= "0111";
    op <= "0011"; -- +
    wait for 50 ns;

    B <= "0101";
    op <= "0111"; -- -
    wait for 50 ns;

    op <= "1101"; -- nand
    wait;
    end process;

end Behavioral;
```

DIAGRAMAS RTL

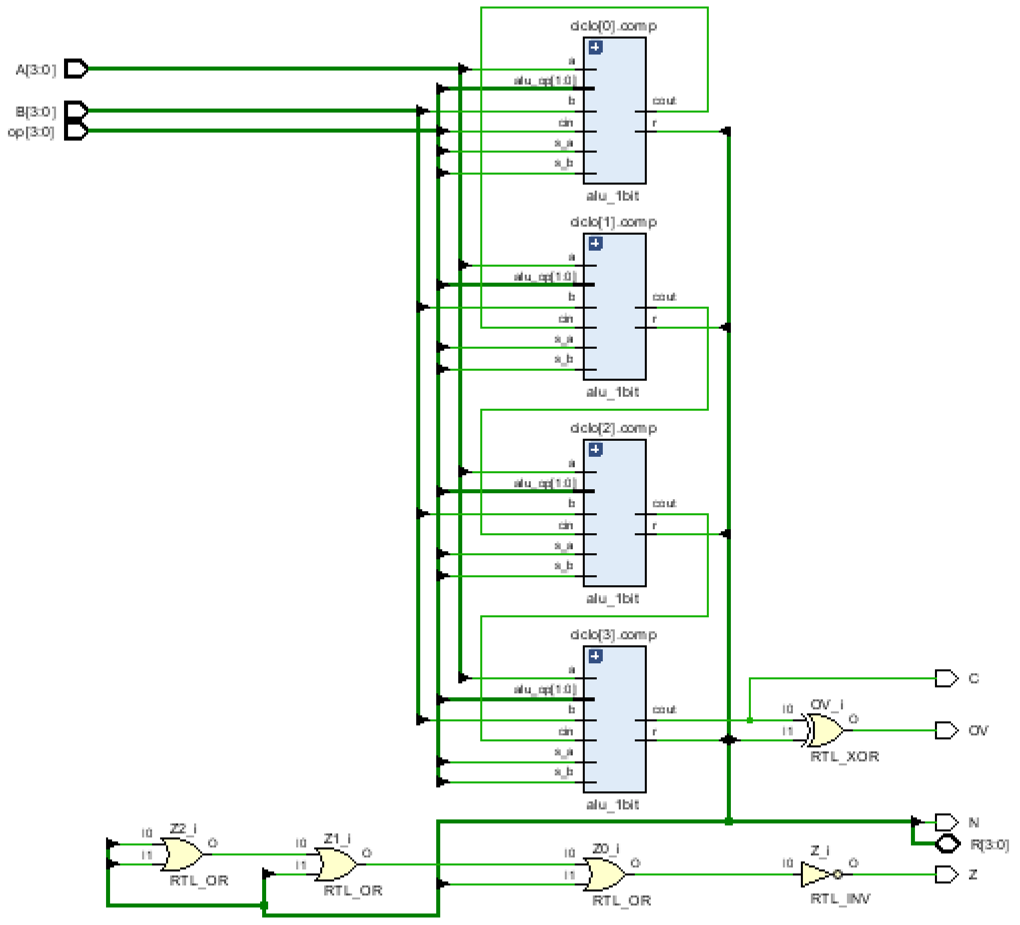
SUMADOR DE 1 BIT



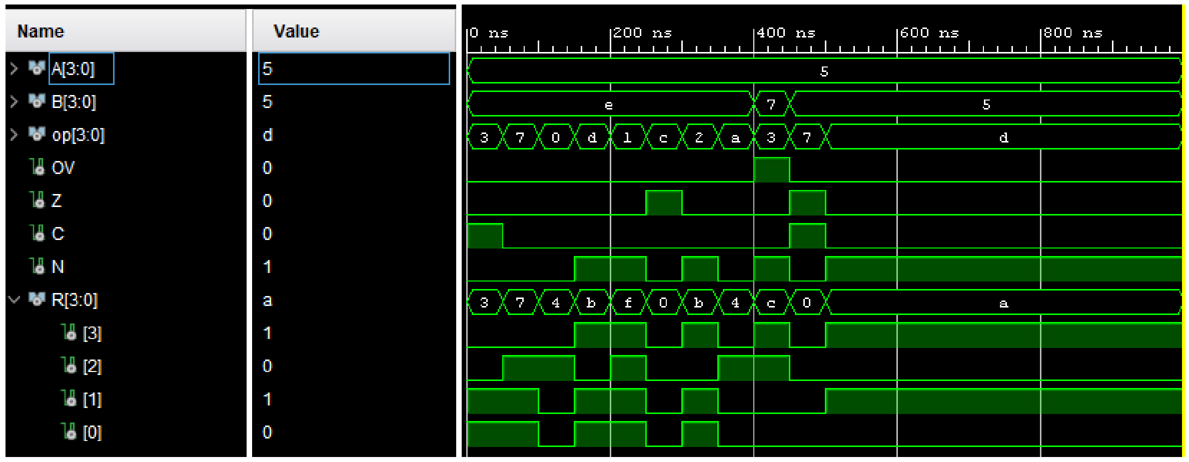
ALU DE 1 BIT



ALU 4 BITS



SIMULACIÓN


Name
Value
A[3:0]
5
B[3:0]
5
op[3:0]
d
OV
0
Z
0
C
0
N
1
R[3:0]
a
[3]
1
[2]
0
[1]
1
[0]
0
0 ns
200 ns
400 ns
600 ns
800 ns