Instituto Politécnico Nacional

Escuela Superior de Cómputo

Práctica 2 - Sumador/restador de 8 bits con acarreo en cascada

Unidad de aprendizaje: Arquitectura de Computadoras

Grupo: 3CV1

Alumno(a):
Ramos Diaz Enrique

Profesor(a):
Vega García Nayeli

23 de febrero 2020

Índice

1. Código de implementación

1.1. Sumador de 1 bit

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

entity sum1bit is
    Port ( a, b, cin : in STD_LOGIC;
           s, cout : out STD_LOGIC);
end sum1bit;

architecture Behavioral of sum1bit is
begin
    s <= a xor b xor cin;
    cout <= (a and cin) or (a and b) or (b and cin);
end Behavioral;
```

1.2. Sumador/Restador de N = 8 bits

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

entity sum_resNbits is
    generic ( n : integer := 8);
    Port ( a,b : in STD_LOGIC_VECTOR (n-1 downto 0);
           cin : in STD_LOGIC;
           s : out STD_LOGIC_VECTOR (n-1 downto 0);
           cout : out STD_LOGIC);
end sum_resNbits;

architecture Behavioral of sum_resNbits is
    component sum1bit is
        Port ( a, b, cin : in STD_LOGIC;
               s, cout : out STD_LOGIC);
    end component;
    signal c : STD_LOGIC_VECTOR (n downto 0);
    signal eb : STD_LOGIC_VECTOR (n-1 downto 0);

begin
    c(0) <= cin;
    ciclo : for i in 0 to n-1 generate
        eb(i) <= b(i) xor c(0);
```

```
        bit0 : sum1bit Port map (
            a => a(i),
            b => eb(i),
            s => s(i),
            cin => c(i),
            cout => c(i+1)
        );
    end generate;
    cout <= c(n);
end Behavioral;
```

2. Código de simulación

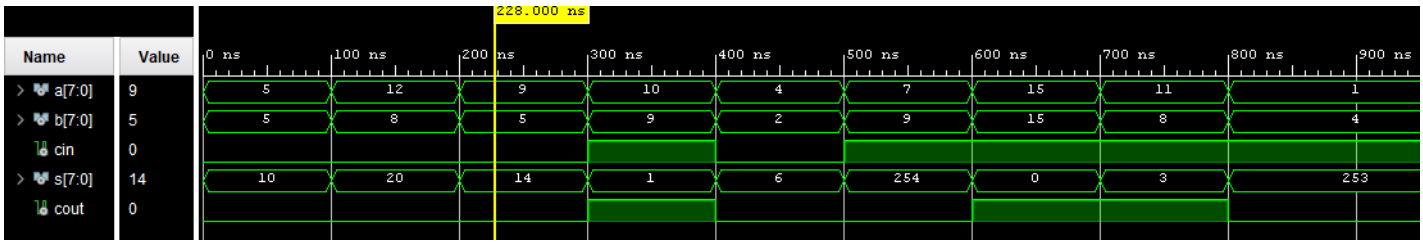
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_bench is
end test_bench;

architecture Behavioral of test_bench is
    component sum_resNbits
        Port ( a, b : in STD_LOGIC_VECTOR (7 downto 0);
               cin : in STD_LOGIC;
               s : out STD_LOGIC_VECTOR (7 downto 0);
               cout : out STD_LOGIC);
    end component;
    signal a, b, s : STD_LOGIC_VECTOR (7 downto 0);
    signal cin, cout : STD_LOGIC;

begin
    sum_res : sum_resNbits Port map (
        a => a,
        b => b,
        s => s,
        cin => cin,
        cout => cout
    );

    process begin
        cin <= '0';
      a <= "00000101"; -- 5
      b <= "00000101"; -- +5
      wait for 100 ns;
```

```
31
32          cin <= '0';
33          a <= "00001100"; -- 12
34          b <= "00001000"; -- +8
35          wait for 100 ns;
36
37          cin <= '0';
38          a <= "00001001"; -- 9
39          b <= "00000101"; -- +5
40          wait for 100 ns;
41
42          cin <= '1';
43          a <= "00001010"; -- 10
44          b <= "00001001"; -- -9
45          wait for 100 ns;
46
47          cin <= '0';
48          a <= "00000100"; -- 4
49          b <= "00000010"; -- +2
50          wait for 100 ns;
51
52          cin <= '1';
53          a <= "00000111"; -- 7
54          b <= "00001001"; -- -9
55          wait for 100 ns;
56
57          cin <= '1';
58          a <= "00001111"; -- 15
59          b <= "00001111"; -- -15
60          wait for 100 ns;
61
62          cin <= '1';
63          a <= "00001011"; -- 11
64          b <= "00001000"; -- -8
65          wait for 100 ns;
66
67          cin <= '1';
68          a <= "00000001"; -- 1
69          b <= "00000100"; -- -4
70          wait;
71      end process;
72  end Behavioral;
```

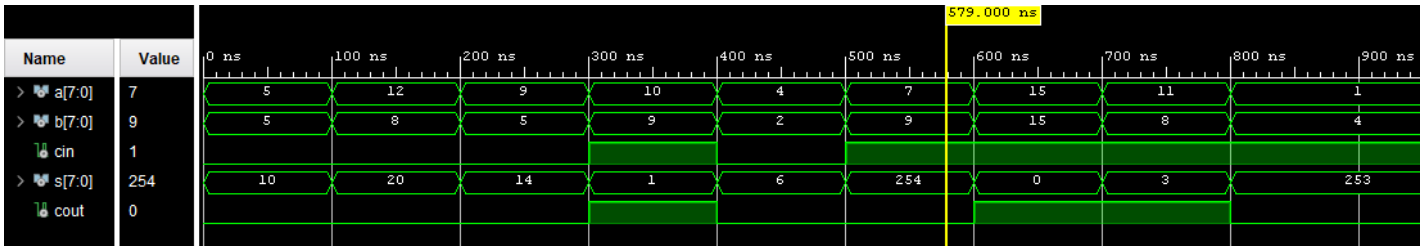
3. Simulación

Operación	A	B	S	Cout
Suma	5	5	10	0
Suma	12	8	20	0
Suma	9	5	14	0
Resta	10	9	1	1
Suma	4	2	6	0
Resta	7	9	254	0
Resta	15	15	0	1
Resta	11	8	3	1
Resta	1	4	253	0

3.1. Suma



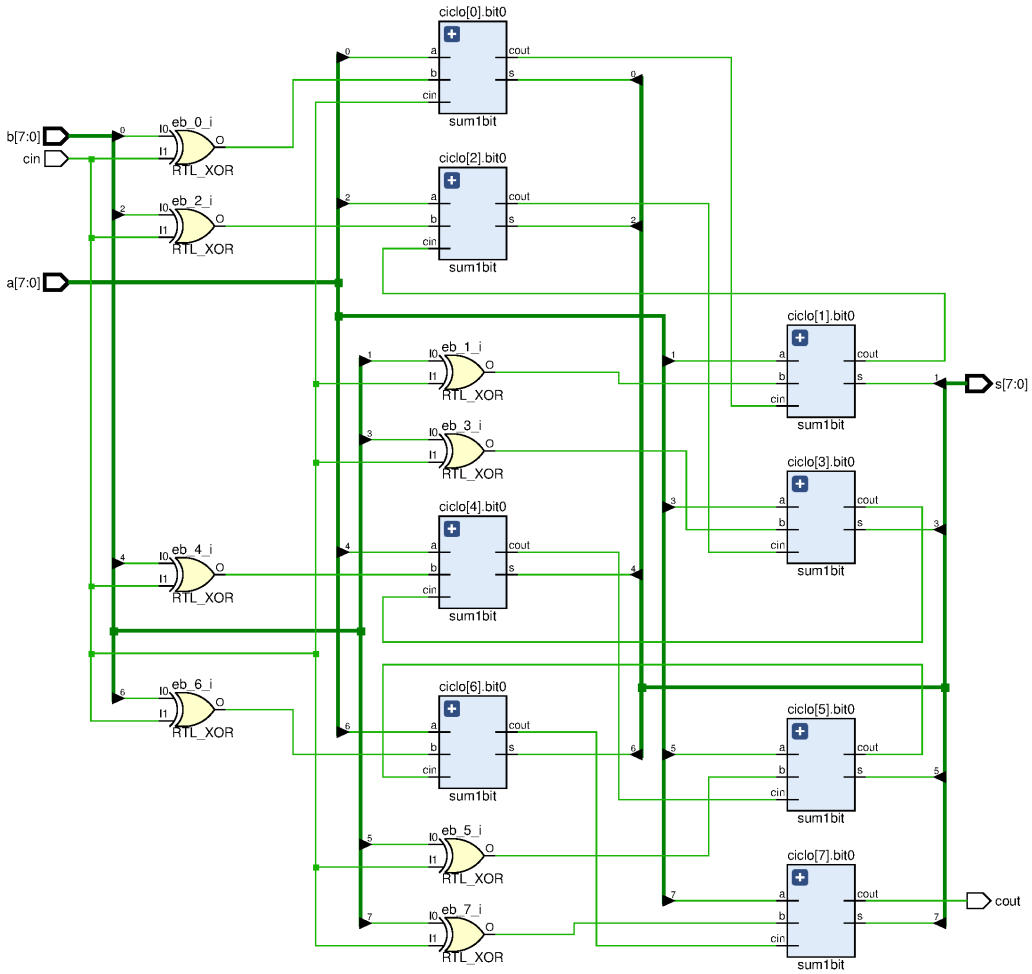
3.2. Resta



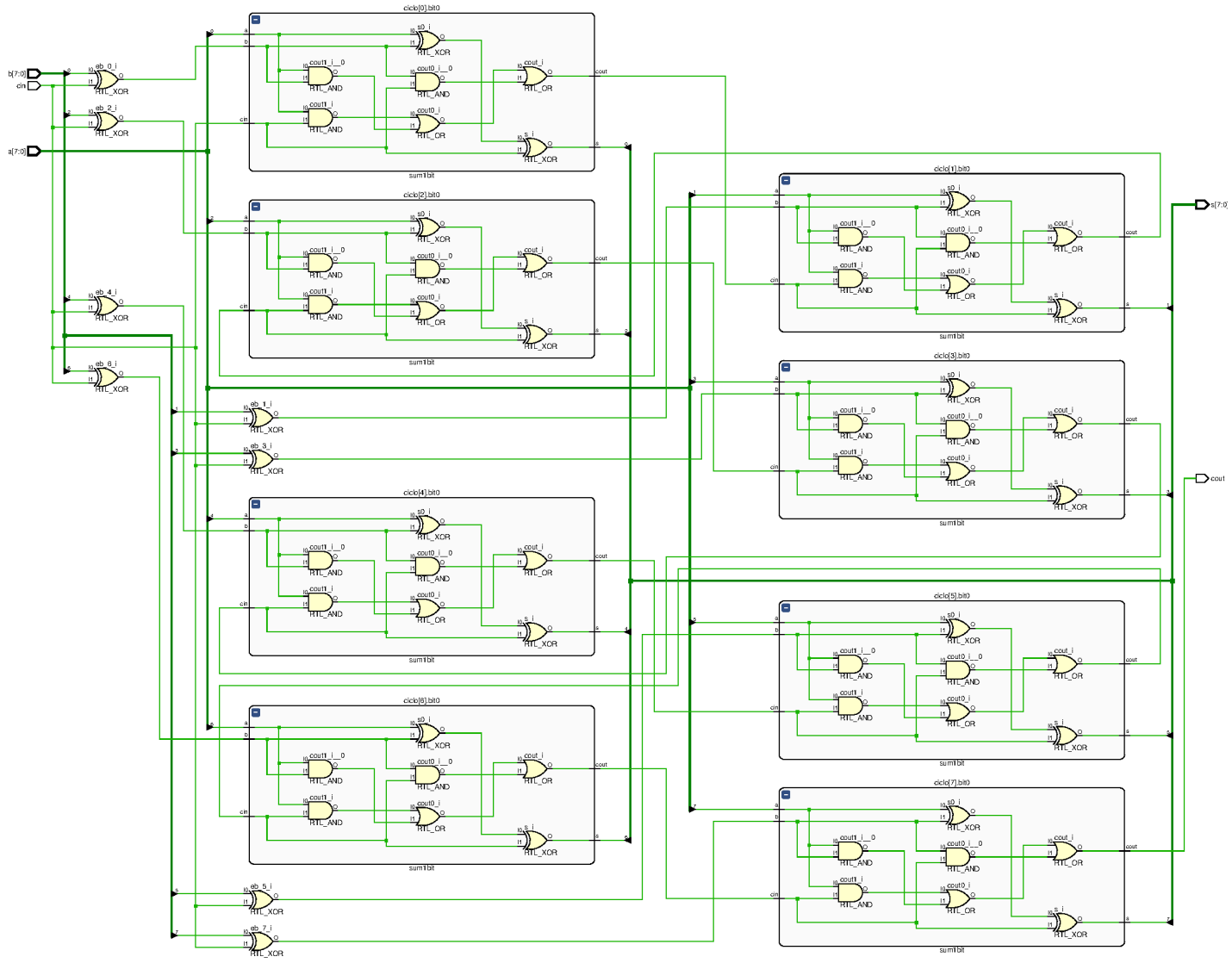
4. Diagramas RTL

4.1. Análisis RTL

4.1.1. Diagrama comprimido



4.1.2. Diagrama expandido



4.2. Synthesis