Instituto Politécnico Nacional

Escuela Superior de Cómputo

Práctica 3 - Sumador de 8 bits con acarreo anticipado

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

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

entity sum_AnticNbits is
    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 sum_AnticNbits;

architecture Behavioral of sum_AnticNbits is
begin
    process(a, b, cin)
        variable p, g : STD_LOGIC_VECTOR(7 downto 0);
        variable c : STD_LOGIC_VECTOR(8 downto 0);
        variable coPj, gkPm, pm : STD_LOGIC;
    begin
        c(0) := cin;
        for i in 0 to 7 loop
            p(i) := a(i) xor b(i);
            g(i) := a(i) and b(i);
            s(i) <= p(i) xor c(i);

            coPj := c(0);
            for j in 0 to i loop
                coPj := coPj and p(j);
            end loop;

            gkPm := '0';
            for k in 0 to i-1 loop
                pm := '1';
                for m in k+1 to i loop
                    pm := pm and p(m);
                end loop;
                gkPm := gkPm or (g(k) and pm);
            end loop;

            c(i+1) := g(i) or coPj or gkPm;
        end loop;
        cout <= c(8);
    end process;
end Behavioral;
```

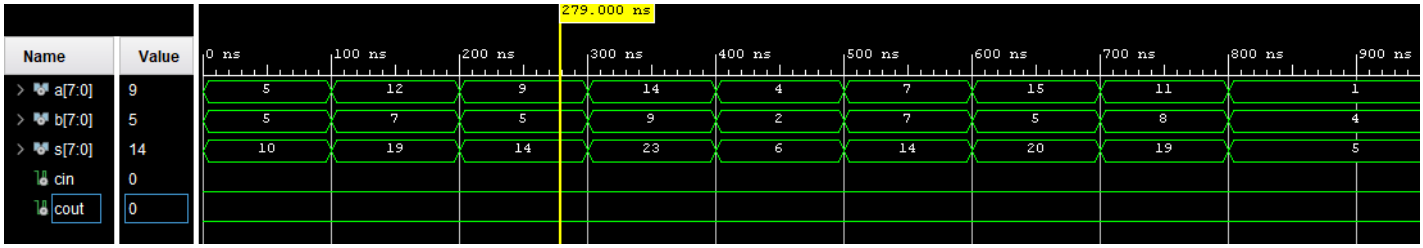
2. Código de simulación

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

entity test_bench is
end test_bench;

architecture Behavioral of test_bench is
    component sum_AnticNbits
        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_Antic : sum_AnticNbits 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;

      cin <= '0';
      a <= "00001100"; -- 12
      b <= "00000111"; -- +7
      wait for 100 ns;

      cin <= '0';
      a <= "00001001"; -- 9
      b <= "00000101"; -- +5
      wait for 100 ns;

      cin <= '0';
      a <= "00001110"; -- 14
```

```
43          b <= "00001001"; -- +9
44          wait for 100 ns;
45
46          cin <= '0';
47          a <= "00000100"; -- 4
48          b <= "00000010"; -- +2
49          wait for 100 ns;
50
51          cin <= '0';
52          a <= "00000111"; -- 7
53          b <= "00000111"; -- +7
54          wait for 100 ns;
55
56          cin <= '0';
57          a <= "00001111"; -- 15
58          b <= "00000101"; -- +5
59          wait for 100 ns;
60
61          cin <= '0';
62          a <= "00001011"; -- 11
63          b <= "00001000"; -- +8
64          wait for 100 ns;
65
66          cin <= '0';
67          a <= "00000001"; -- 1
68          b <= "00000100"; -- +4
69          wait;
70      end process;
71  end Behavioral;
```

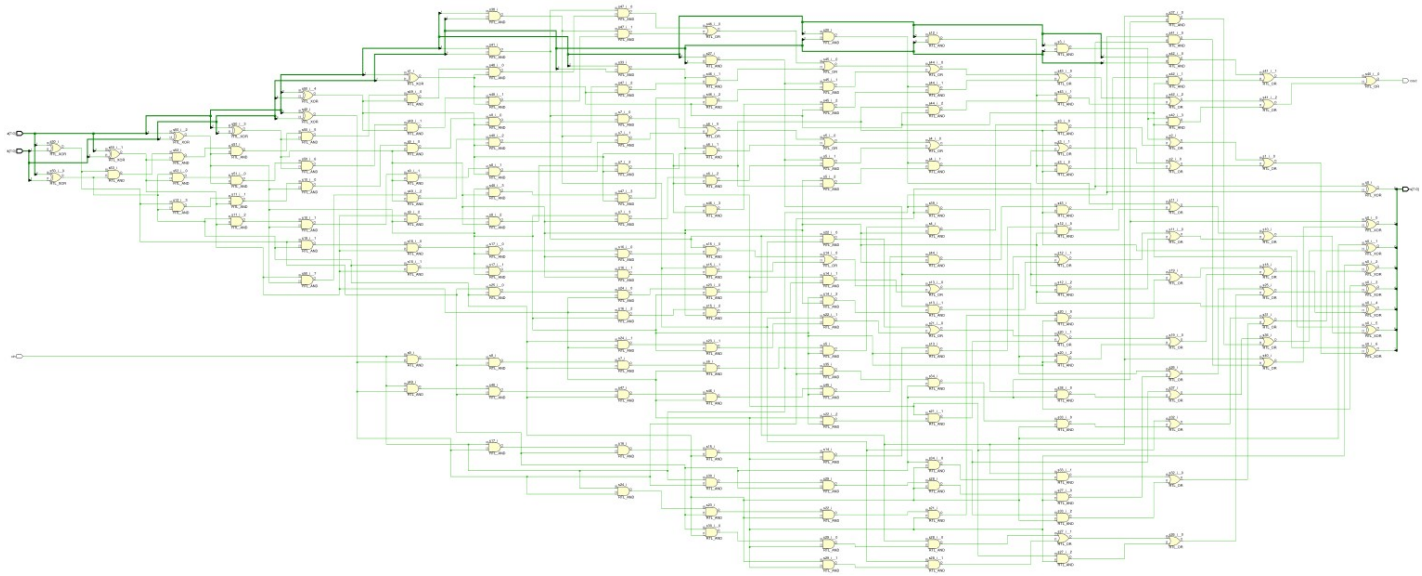
3. Simulación

Operación	A	B	S	Cout
Suma	5	5	10	0
Suma	12	7	19	0
Suma	9	5	14	0
Suma	14	9	23	0
Suma	4	2	6	0
Suma	7	7	14	0
Suma	15	5	20	0
Suma	11	8	19	0
Suma	1	4	5	0



4. Diagramas RTL

4.1. Análisis RTL



4.2. Synthesis