Reporte de practica 2

González Pardo Adrian

Febrero 2020

1. Código VHDL

Código 1: Descripción del modulo sumador de 1 solo bit

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
4 entity sumadorNBits is
5 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);
11 end sumadorNBits;
13 architecture Behavioral of sumadorNBits is
  component sumador is
14
      Port ( a,b,cin : in STD_LOGIC;
             s : out STD_LOGIC;
16
             cout : out STD_LOGIC);
```

```
18 end component;
signal c:std_logic_vector(n downto 0);
20 signal eb:std_logic_vector(n-1 downto 0);
21 begin
22 c(0) <= cin;
      ciclo: for i in 0 to n-1 generate
           eb(i) \le b(i) \times c(0);
           bit1: sumador Port map(
                a=> a(i),
26
               b = > eb(i),
                cin=>c(i),
                s=>s(i),
29
                cout = > c(i+1)
30
           );
      end generate;
      cout <= c(n);
36 end Behavioral;
```

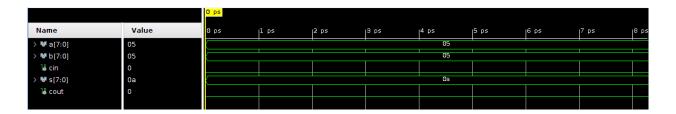
Código 2: A través de uso de componentes de código para la descripción del sumador/restador en cascada de 8 bits

2. Test-Bench VHDL Código

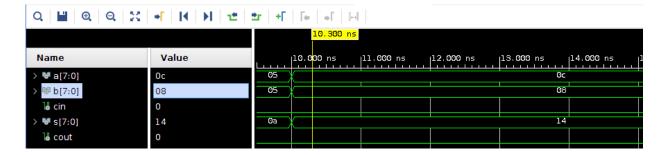
```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity simuSumador is
    Port ();
5 end simuSumador;
  architecture Behavioral of simuSumador is
  component sumadorNBits is
      Port ( a,b : in STD_LOGIC_VECTOR (7 downto 0);
             cin : in STD_LOGIC;
             s : out STD_LOGIC_VECTOR (7 downto 0);
11
             cout : out STD_LOGIC);
13 end component;
signal a :STD_LOGIC_VECTOR (7 downto 0) := x"00";
signal b :STD_LOGIC_VECTOR (7 downto 0) := x"00";
signal cin : STD_LOGIC := '0';
signal s : STD_LOGIC_VECTOR (7 downto 0);
18 signal cout : STD_LOGIC;
19 begin
20 sumado : sumadorNBits
      Port map (
      a => a
      b \Rightarrow b,
23
      cin => cin,
24
      s => s,
25
      cout => cout
      );
```

```
28 p1 : process
29
  begin
       cin <= '0';
       a \le x"05";
31
       b \le x"05";
32
       wait
             for 10 ns;
33
         <= x"0C";
34
       b \le x"08";
       wait for 10 ns;
36
         <= x"09";
37
       b \le x"05";
       wait for 10 ns;
39
       cin <= '1';
40
         \leq x"0A";
41
       b \le x"09";
42
       wait for 10 ns;
43
       cin <= '0';
44
       a \le x"04";
45
       b \le x"02";
       wait for 10 ns;
47
       cin <= '1';
48
       a \le x"07";
49
       b \le x"09";
50
       wait for 10 ns;
         <= x"0F";
         <= x"0F";
53
       wait for 10 ns;
         <= x"0B";
         <= x"08";
56
       wait for 10 ns;
         <= x"0A";
58
       b \le x"09";
59
       wait
             for 10 ns;
60
         <= x"01";
61
       b \le x''04'';
       wait;
64 end process;
  end Behavioral;
```

Código de simulación Anexo de fotos de la simulación de impulsos



Primer parte con valores hexadecimales equivales a valores decimales: $a = 5_{10} \& b = 5_{10}$ con salida $s = A_{16} = 10_{10}$



Segunda parte con valores hexadecimales equivales a valores decimales: $a=C_{16}=12_{10}$ & $b=8_{10}$ con salida $s=14_{16}=20_{10}$

				20.980 ns			
Name	Value		20.000 ns	21.000 ns	22.000 ns	23. 000 ns	24.000 ns
> 🚳 a[7:0]	09	0с	X			09	
> ₩ b[7:0]	05	08	*			05	
¹⊌ cin	0						
> W s[7:0]	0e	14	X			0e	
¹⊌ cout	0						

Tercer parte con valores hexadecimales equivales a valores decimales: $a = 9_{10} \& b = 5_{10}$ con salida $s = E_{16} = 14_{10}$

				31.002 ns			
Name	Value	29.000 ns	30.000 ns	31.000 ns	32.000 ns	33.000 ns	34.000 ns
> ₩ a[7:0]	0a	09	X			0a	
> 😽 b[7:0]	09	05	X			09	
¹⊌ cin	1						
> 💖 s[7:0]	01	0e	X			01	
l⊌ cout	1						

Cuarta parte con valores hexadecimales equivales a valores decimales: $a=A_{16}=10_{10}$, $b=9_{10}$ & $cin=1_{10}$ con salida $s=1_{10}$ y un $cout=1_{10}$

				40.982 ns			
Name	Value		40.000 ns	41.000 ns	42.000 ns	43. 000 ns	44.000 ns
> 🚳 a[7:0]	04	0a	Х			04	
> ₩ b[7:0]	02	09	Х			02	
l⊌ cin	0						
> W s[7:0]	06	01	X			06	
¹⊌ cout	0						

Quinta parte con valores hexadecimales equivales a valores decimales: $a=4_{10}$, $b=2_{10}$ & $cin=0_{10}$ con salida $s=6_{10}$

				51.002 ns			
Name	Value		50.000 ns	51.000 ns	52.000 ns	53. 000 ns	54.000 ns
> ™ a[7:0]	07	04	X			07	
> W b[7:0]	09	02	X			09	
¹⊌ cin	1						
> W s[7:0]	fe	06	X			fe	
¹⊌ cout	0						

Sexta parte con valores hexadecimales equivales a valores decimales: $a=7_{10}$, $b=9_{10}$ & $cin=1_{10}$ con salida $s=FE_{16}=254_{10,C2}=-2_{10}$ y un cout $=0_{10}$

				61.012 ns			
Name	Value		60.000 ns	61. 000 ns	62.000 ns	63. 000 ns	64.000 ns
> W a[7:0]	Of	07	X			Of	
> W b[7:0]	Of	09	X			0f	
⅓ cin	1						
> W s[7:0]	00	fe	X			00	
¼ cout	1						

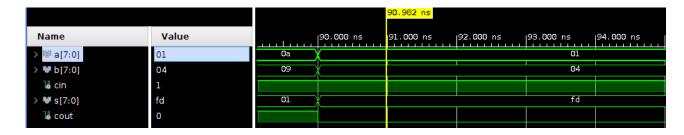
Septima parte con valores hexadecimales equivales a valores decimales: $a=F_{16}=15_{10}$, $b=F_{16}=15_{10}$ & $cin=1_{10}$ con salida $s=0_{10}$ y un $cout=1_{10}$

		71.012 ns	
Name	Value	70.000 ns 71.000 ns 72.000 ns 73.000 ns 7	4.000 ns
> W a[7:0]	Ob	Of Ob	
> W b[7:0]	08	0f 08	
¹⊌ cin	1		
> 😽 s[7:0]	03	00 03	
¹⊌ cout	1		

Octava parte con valores hexadecimales equivales a valores decimales: $a=B_{16}=11_{10}$, $b=8_{10}$ & $cin=1_{10}$ con salida $s=3_{10}$ y un $cout=1_{10}$

				80.992 ns			
Name	Value		80.000 ns	81. 000 ns	82. 000 ns	83. 000 ns	84.000 ns
> ₩ a[7:0]	0a	ОР	K			Oa	
> W b[7:0]	09	08	k			09	
¹⊌ cin	1						
> 🕷 s[7:0]	01	03	(01	
1₫ cout	1						

Novena parte con valores hexadecimales equivales a valores decimales: $a=A_{16}=10_{10}$, $b=9_{10}$ & $cin=1_{10}$ con salida $s=1_{10}$ y un $cout=1_{10}$



Decima parte con valores hexadecimales equivales a valores decimales: $a=1_{10}$, $b=4_{10}$ & $cin=1_{10}$ con salida $s=FD_{16}=253_{10.C2}=-3_{10}$ y un cout $=0_{10}$

3. Tabla de resultados

Operación	A	В	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 -C2 = -2	0
Resta	15	15	0	1
Resta	11	8	3	1
Resta	10	9	1	1
Resta	1	4	253 -C2 = -3	0

4. Diagrama RTL

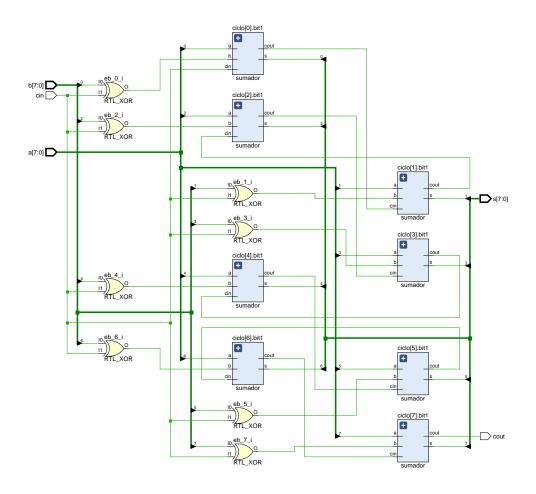


Diagrama RTL del archivo VHDL del sumador/restador en cascada de 8 bits

5. Diagrama de Síntesis

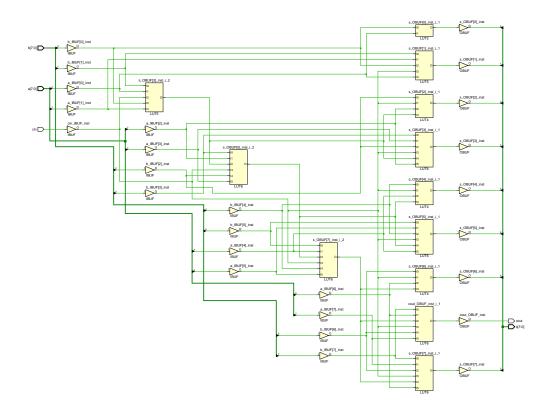


Diagrama de Sintesis del archivo VHDL del sumador/restador en cascada de 8 bits