

Instituto Politécnico Nacional
Escuela Superior de Computo



Práctica 5.

Sumador de 8 bits con acarreo anticipado.

Nombre: Flores Castro Luis Antonio.

Arquitectura de Computadoras.

Profesora: Vega García Nayeli.

Código VHDL del Sumador de 8 bits con acarreo anticipado.

```
1. library IEEE;
2. use IEEE.STD_LOGIC_1164.ALL;
3.
4. entity SumadorAP is
5.     Port ( A: in STD_LOGIC_VECTOR (7 downto 0);
6.           B : in STD_LOGIC_VECTOR (7 downto 0);
7.           sel : in STD_LOGIC;
8.           S : out STD_LOGIC_VECTOR (7 downto 0);
9.           Cout : out STD_LOGIC);
10. end SumadorAP;
11.
12. architecture Behavioral of SumadorAP is
13. begin
14.     process(A, B, sel)
15.         variable P, G : STD_LOGIC_VECTOR(7 downto 0);
16.         variable c : STD_LOGIC_VECTOR(8 downto 0);
17.         variable aux1, aux2, aux3 : STD_LOGIC;
18.     begin
19.         c(0) := sel;
20.         for i in 0 to 7 loop
21.             P(i) := A(i) xor B(i);
22.             G(i) := A(i) and B(i);
23.             S(i) <= P(i) xor c(i);
24.
25.             aux1 := c(0);
26.             for j in 0 to i loop
27.                 aux1 := aux1 and P(j);
28.             end loop;
29.
30.             aux2 := '0';
31.             for k in 0 to i-1 loop
32.                 aux3 := '1';
33.                 for m in k+1 to i loop
34.                     aux3 := aux3 and P(m);
35.                 end loop;
36.                 aux2 := aux2 or (G(k) and aux3);
37.             end loop;
38.
39.             c(i+1) := G(i) or aux1 or aux2;
40.         end loop;
41.         Cout <= c(8);
42.     end process;
43. end Behavioral;
```

Código VHDL Test-Bench del Sumador de 8 bits con acarreo anticipado.

```
1. library IEEE;
2. use IEEE.STD_LOGIC_1164.ALL;
3.
4. entity SumadorAPtb is
5. end SumadorAPtb;
6.
7. architecture Behavioral of SumadorAPtb is
8.     component SumadorAP
9.         Port ( A: in STD_LOGIC_VECTOR (7 downto 0);
10.              B : in STD_LOGIC_VECTOR (7 downto 0);
11.              sel : in STD_LOGIC;
12.              S : out STD_LOGIC_VECTOR (7 downto 0);
13.              Cout : out STD_LOGIC);
14.     end component;
15.     --aquí señales
16.     signal A: STD_LOGIC_VECTOR (7 downto 0);
17.     signal B: STD_LOGIC_VECTOR (7 downto 0);
18.     signal S: STD_LOGIC_VECTOR (7 downto 0);
19.     signal sel: STD_LOGIC;
20.     signal Cout : STD_LOGIC;
21.
22. begin
23.     --proceso e instanciar.
24.     sumadoracarreo : SumadorAP Port map (
25.         A => A,
26.         B => B,
27.         S => S,
28.         sel => sel,
29.         Cout => Cout);
30.
31.     process
32.     begin
33.         sel <= '0';
34.         A <= "00010111";
35.         B <= "01010111";
36.         wait for 20 ns;
37.
38.         sel <= '0';
39.         A <= "00011010";
40.         B <= "01011111";
41.         wait for 20 ns;
42.
43.         sel <= '0';
44.         A <= "00101001";
45.         B <= "01100010";
46.         wait for 20 ns;
47.
48.         sel <= '0';
```

```

49.      A <= "10111010";
50.      B <= "00100100";
51.      wait for 20 ns;
52.
53.      sel <= '0';
54.      A <= "00111111";
55.      B <= "01100010";
56.      wait for 20 ns;
57.
58.      sel <= '0';
59.      A <= "00111100";
60.      B <= "00001111";
61.      wait for 20 ns;
62.
63.      sel <= '0';
64.      A <= "01111000";
65.      B <= "00110000";
66.      wait for 20 ns;
67.
68.      sel <= '0';
69.      A <= "00001011";
70.      B <= "00001000";
71.      wait for 20 ns;
72.
73.      sel <= '0';
74.      A <= "00000001";
75.      B <= "00000100";
76.      wait;
77.  end process;
78. end Behavioral;

```

Diagrama RTL.

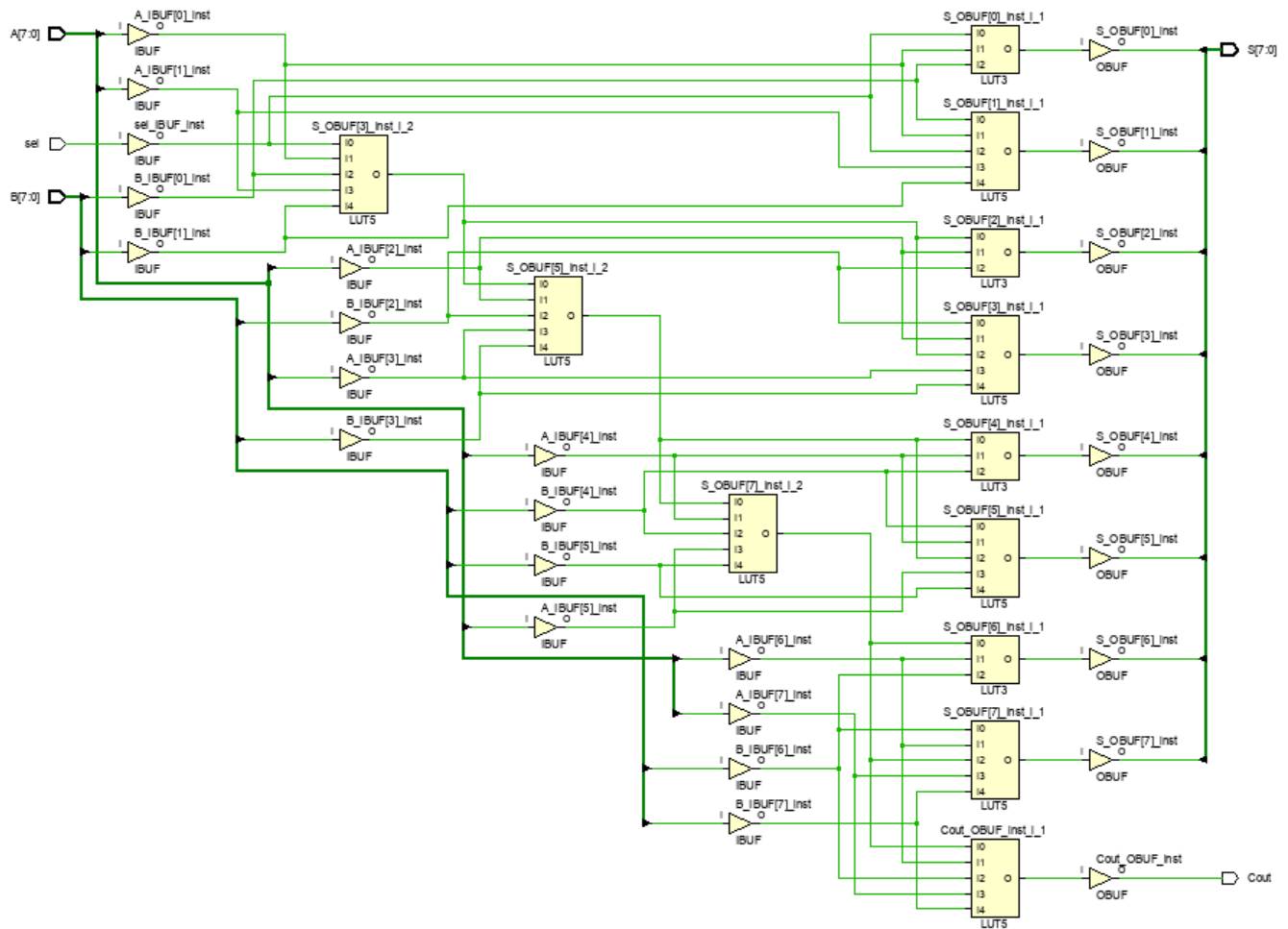


Figura 1. Circuito esquemático del Sumador de 8 bits con acarreo anticipado.

Diagrama RTL.

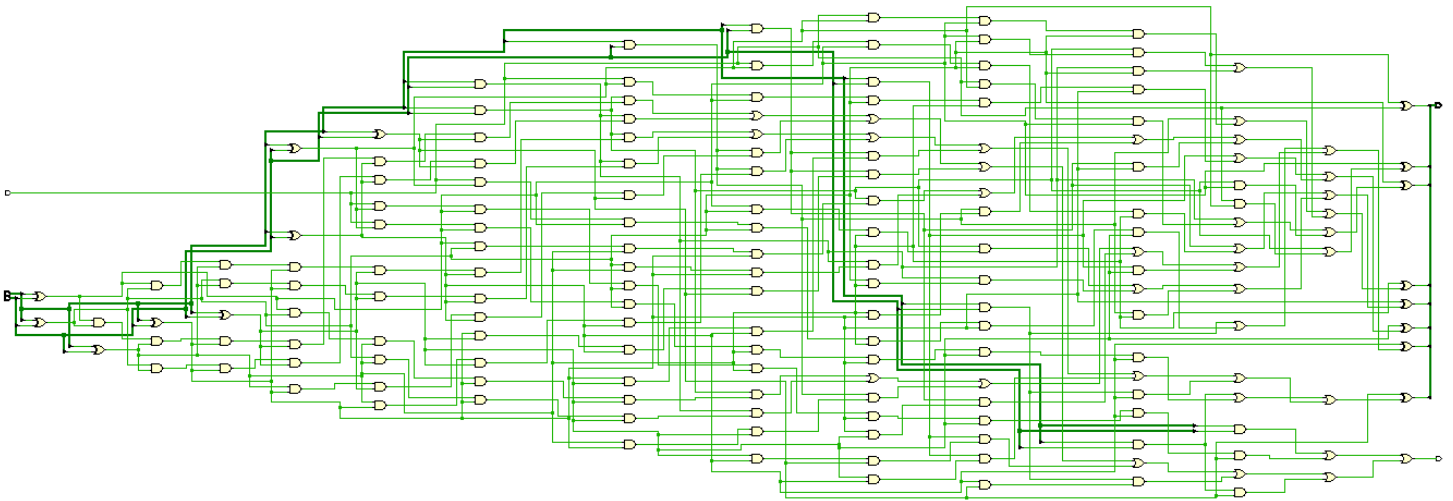


Figura 1.1. Circuito lógico del Sumador de 8 bits con acarreo anticipado.

Diagrama de Onda del Sumador de 8 bits con acarreo anticipado.

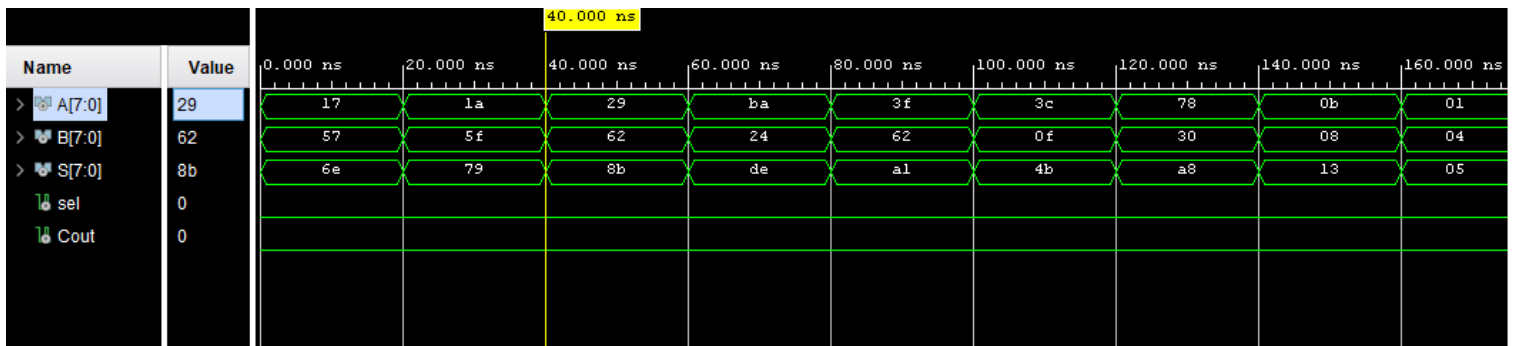


Figura 2. Simulación del Test-Bench del Sumador de 8 bits con acarreo anticipado.

Tabla de Resultados.

Operación	A	B	S	Cout
Suma	23	87	110	0
Suma	26	95	121	0
Suma	41	98	139	0
Suma	186	36	222	0
Suma	63	98	161	0
Suma	60	15	75	0
Suma	120	48	168	0
Suma	11	8	19	0
Suma	1	4	5	0

Tabla 1. Resultados de la simulación del Sumador de 8 bits con acarreo anticipado.

Tabla de Resultados en hexadecimal originada por el programa originalmente.

Operación	A	B	S	Cout
Suma	17	57	6E	0
Suma	1A	5F	79	0
Suma	29	62	8B	0
Suma	BA	24	DE	0
Suma	3F	62	A1	0
Suma	3C	0F	4B	0
Suma	78	30	A8	0
Suma	0B	08	13	0
Suma	01	04	05	0

Tabla 1.1 Resultados de la simulación del Sumador de 8 bits con acarreo anticipado en base hexadecimal.

Se anexa la tabla 1.1 ya que Vivado no me dejó modificar los valores a decimal, los valores de la tabla 1.1 son los generados originalmente por la simulación de onda y los de la tabla 1, es la conversión de estos valores a base decimal.