Práctica 4

Código de implementación:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity sumador4b is
    Port ( a,b : in STD_LOGIC_VECTOR (3 downto 0);
           sel : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (3 downto 0);
           cout : out STD_LOGIC);
end sumador4b;
architecture Behavioral of sumador4b is
signal eb: std_logic_vector(3 downto 0);
signal c: std_logic_vector(4 downto 0);
begin
    c(0)  <= sel;
    cicle : for i in 0 to 3 generate
        eb(i) <= b(i) xor c(0);
        S(i) <= a(i) xor eb(i) xor c(i);
        c(i+1) <= (a(i) and eb(i)) or (a(i) and c(i)) or (eb(i) and c(i));
    end generate;
    cout <= c(4);
end Behavioral;
```

Código test-brench:

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

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

architecture Behavioral of sumadorTB is
component sumador4b is
    Port ( a,b : in STD_LOGIC_VECTOR (3 downto 0);
           sel : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (3 downto 0);
           cout : out STD_LOGIC);
end component;

signal a,b :  STD_LOGIC_VECTOR (3 downto 0);
signal sel :  STD_LOGIC;
signal S :  STD_LOGIC_VECTOR (3 downto 0);
signal cout :  STD_LOGIC;

begin
unidad1:  sumador4b
    Port map(
        a => a,
        b => b,
            sel => sel,
            S => S,
            cout => cout
    );
process
begin
  a <= "valor inicial";
  b <= "valor inicial";
  sel <= 'operador (0=+ / 1=-)';
  wait for 20 ns; --espera este tiempo antes de ir
    --a otra operación
  sel <= 'operador (0=+ / 1=-)';
  a <= "valor nuevo";
  b <= "valor nuevo";
  wait;
end process;
end Behavioral;
```

Diagrama RTL:

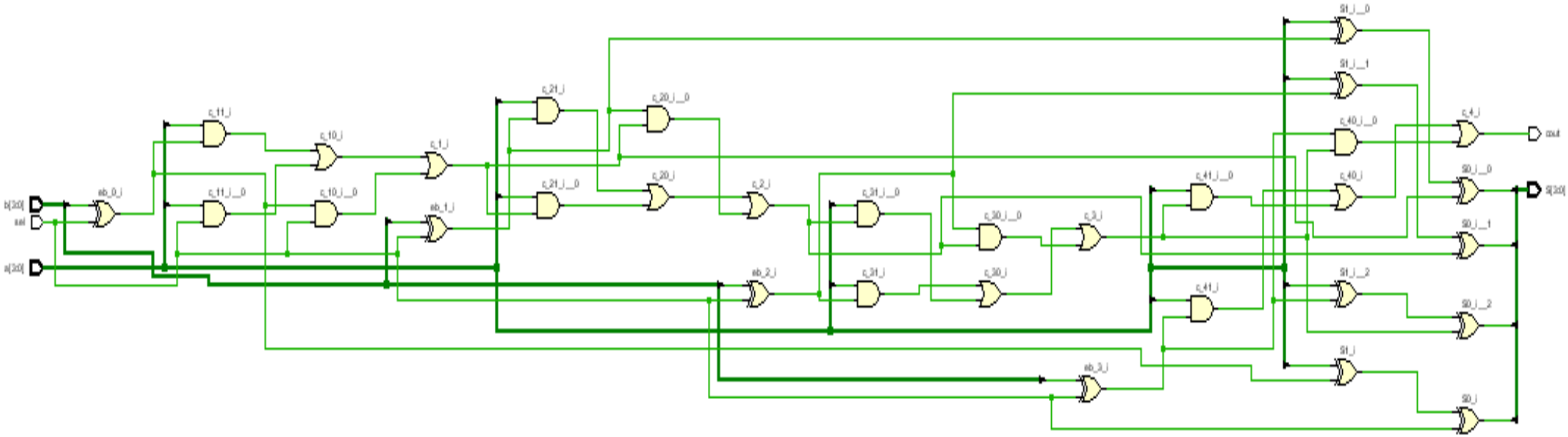
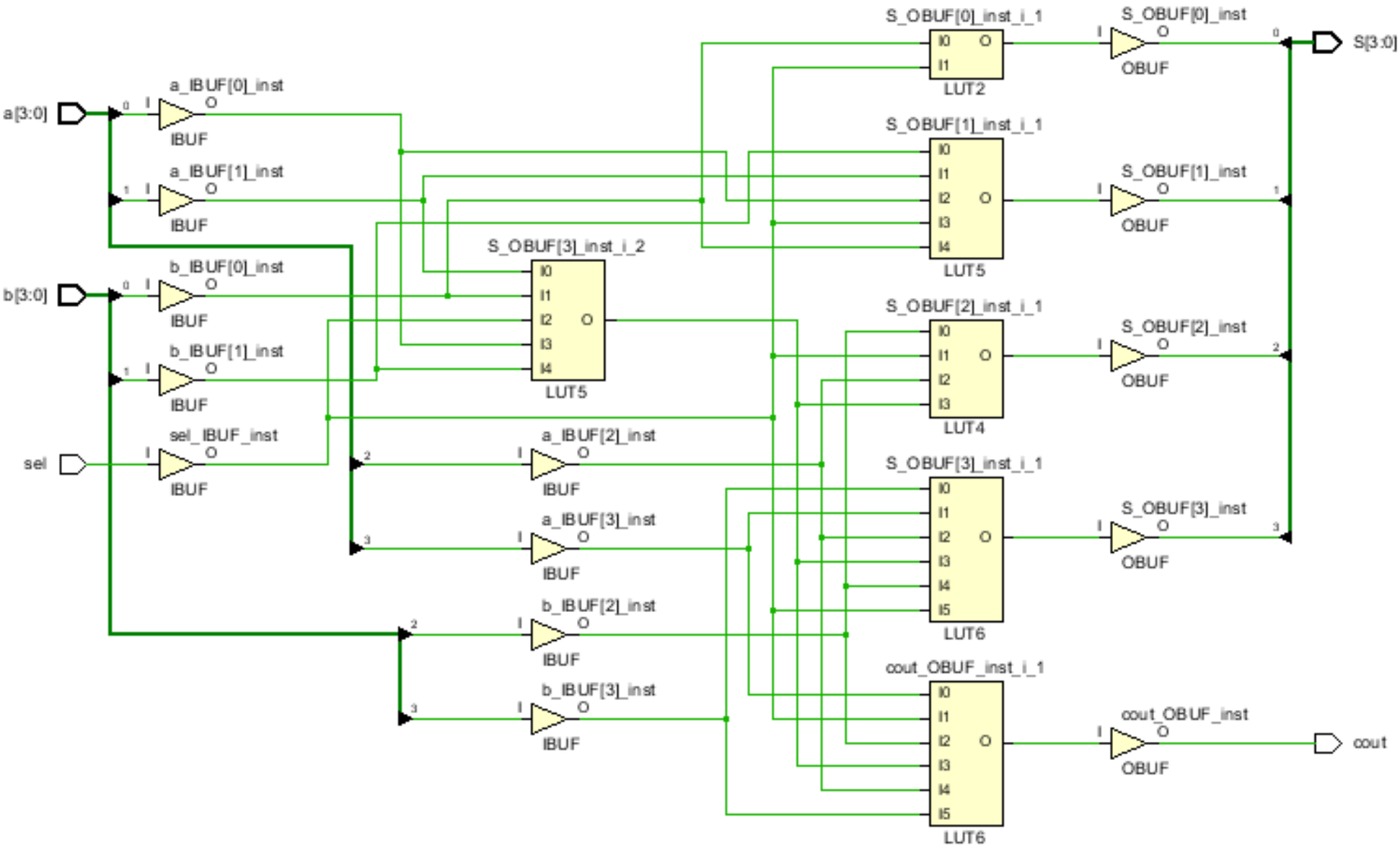


Diagrama Lógico:



Forma de onda:

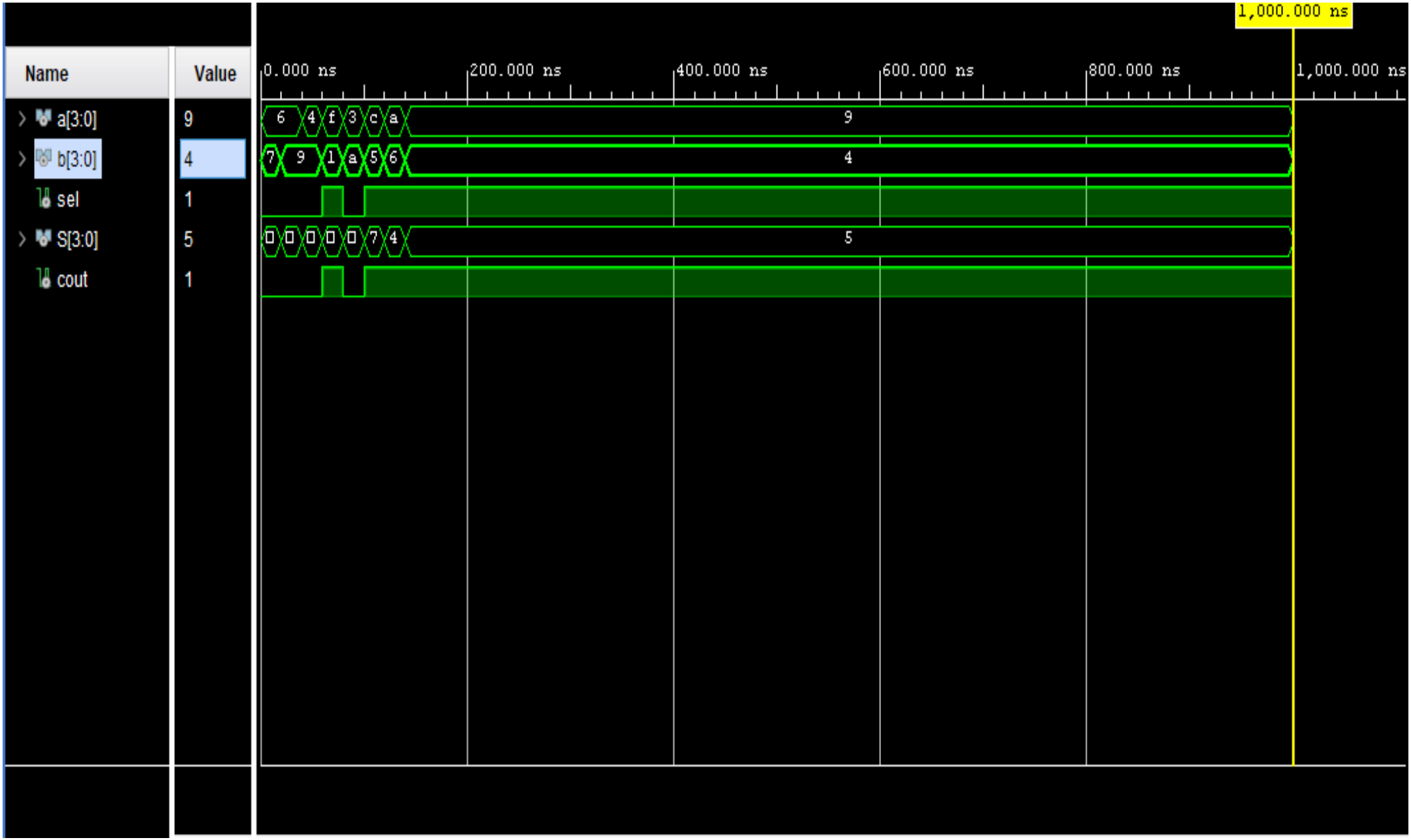


Tabla de resultados:

Operación	A	B	S	Cout
Suma	6	7	13	0
Suma	6	9	15	0
Suma	4	9	13	0
Resta	15	1	14	1
Suma	3	10	13	0
Resta	12	5	7	1
Resta	14	8	6	1
Resta	10	6	4	1
Resta	9	4	5	1