Instituto Politécnico Nacional

Escuela Superior de Computo

Práctica 6.

Unidad aritmética / lógica de 4 bits.

Nombre: Flores Castro Luis Antonio.

Arquitectura de Computadoras.

Profesora: Vega García Nayeli.

Código VHDL de Unidad aritmética / lógica de 1 bit.

```
1. library IEEE;
2. use IEEE.STD_LOGIC_1164.ALL;
3.
4. entity ALU1bit is
5.     Port ( A : in STD_LOGIC;
6.            B : in STD_LOGIC;
7.            CIN : in STD_LOGIC;
8.            SELA : in STD_LOGIC;
9.            SELB : in STD_LOGIC;
10.                op : in STD_LOGIC_VECTOR (1 downto 0);
11.                result : out STD_LOGIC;
12.                cout : out STD_LOGIC);
13.     end ALU1bit;
14.
15.     architecture Behavioral of ALU1bit is
16.     signal auxa, auxb, auxand, auxor, auxxor,
  auxsuma: std_logic;
17.     begin
18.
19.         --Multiplexores
20.         auxa<= A when SELA='0' else not A;
21.         auxb<= B xor SELB;
22.
23.         auxand<= auxa and auxb;
24.         auxor<= auxa or auxb;
25.         auxxor<= auxa xor auxb;
26.         auxsuma<= auxa xor auxb xor CIN;
27.
28.         --agregar condicion para activacion en operaciones
  aritmeticas de bandera
29.         cout<= (auxa and CIN) or (auxa and auxb) or (auxb an
  d CIN) when op="11" else '0';
30.
31.         result<= auxand when op ="00" else
32.                  auxor when op ="01" else
33.                  auxxor when op="10" else
34.                  auxsuma;
35.
36.     end Behavioral;
```

Código VHDL de Unidad aritmética / lógica de 4 bits.

```
1. library IEEE;
2. library work;
3. use IEEE.STD_LOGIC_1164.ALL;
4. use work.paqueteALU.all;
5.
6. entity ALU4bits is
7.     Port ( a : in STD_LOGIC_VECTOR (3 downto 0);
8.            b : in STD_LOGIC_VECTOR (3 downto 0);
9.            Aluop : in STD_LOGIC_VECTOR (3 downto 0);
10.                Res : inout STD_LOGIC_VECTOR (3 downto 0);
11.                bandera : out STD_LOGIC_VECTOR(3 downto 0);
12.                Cout : out STD_LOGIC
13.            );
14.     end ALU4bits;
15.
16.     architecture Behavioral of ALU4bits is
17.     signal C : std_logic_vector(4 downto 0); --vector de
   acarreos
18.     begin
19.     C(0)<=Aluop(2);
20.
21.     ciclo: FOR I IN 0 TO 3 GENERATE
22.         ALUcompleta: ALU1bit port map
23.             (
24.                 A => a(i),
25.                 B => b(i),
26.                 CIN => C(i),
27.                 SELA => Aluop(3),
28.                 SELB => Aluop(2),
29.                 op => Aluop(1 downto 0),
30.                 result => Res(i),
31.                 cout => C(i+1)
32.             );
33.     end GENERATE;
34.     Cout<=C(4);
35.
36.     --orden OV / N / Z / C
37.     bandera(0)<=c(4);--acarreo
38.     bandera(2)<=Res(3);--signo
39.     bandera(3)<= (C(4) xor C(3));--ov
40.
41.     --para bandera zero
42.     process(Res, C)
43.     variable z: std_logic;
44.     begin
45.     z:='0';
46.     banderazero: for k in 3 downto 0 loop
47.         z:= z or Res(k);
48.     end loop;
```

```
49.      --zero igual a NOR negar z
50.      bandera(1)<= not z;--zero
51.      end process;
52.
53.      end Behavioral;
```

Código VHDL Test-Bench de Unidad aritmética / lógica de 1 bit.

```
1. library IEEE;
2. use IEEE.STD_LOGIC_1164.ALL;
3.
4. entity testALU1bit is
5. end testALU1bit;
6.
7. architecture Behavioral of testALU1bit is
8. component ALU1bit is
9.     Port ( A : in STD_LOGIC;
10.               B : in STD_LOGIC;
11.               CIN : in STD_LOGIC;
12.               SELA : in STD_LOGIC;
13.               SELB : in STD_LOGIC;
14.               op : in STD_LOGIC_VECTOR (1 downto 0);
15.               result : out STD_LOGIC;
16.               cout : out STD_LOGIC);
17.     end component;
18.
19.     signal A : STD_LOGIC;
20.     signal B : STD_LOGIC;
21.     signal CIN : STD_LOGIC;
22.     signal SELA : STD_LOGIC;
23.     signal SELB : STD_LOGIC;
24.     signal op : STD_LOGIC_VECTOR (1 downto 0);
25.     signal result : STD_LOGIC;
26.     signal cout : STD_LOGIC;
27.
28.     begin
29.
30.     ALU1B: ALU1BIT
31.     Port map( A=>A,
32.               B=>B,
33.               CIN=>CIN,
34.               SELA=>SELA,
35.               SELB=>SELB,
36.               op=>op,
37.               result=>result,
38.               cout=>cout
39.               );
40.
41.     process
42.     begin
43.
44.         A<='1';
45.         B<='0';
46.         CIN<='0';
47.         SELA<='0';
48.         SELB<='0';
49.         op<="00";
```

```
50.         wait for 20 ns;
51.
52.         op<="01";
53.         wait for 20 ns;
54.
55.         op<="10";
56.         wait for 20 ns;
57.
58.         op<="11";
59.         wait for 20 ns;
60.
61.
62.         A<='1';
63.         B<='0';
64.         CIN<='0';
65.         SELA<='0';
66.         SELB<='1';
67.         op<="00";
68.         wait for 20 ns;
69.
70.         op<="01";
71.         wait for 20 ns;
72.
73.         op<="10";
74.         wait for 20 ns;
75.
76.         op<="11";
77.         wait for 20 ns;
78.
79.
80.         A<='1';
81.         B<='0';
82.         CIN<='0';
83.         SELA<='1';
84.         SELB<='0';
85.         op<="00";
86.         wait for 20 ns;
87.
88.         op<="01";
89.         wait for 20 ns;
90.
91.         op<="10";
92.         wait for 20 ns;
93.
94.         op<="11";
95.         wait for 20 ns;
96.
97.
98.         A<='1';
99.         B<='0';
100.        CIN<='0';
101.        SELA<='1';
```

```
102.         SELB<='1';
103.         op<="00";
104.         wait for 20 ns;
105.
106.         op<="01";
107.         wait for 20 ns;
108.
109.         op<="10";
110.         wait for 20 ns;
111.
112.         op<="11";
113.         wait for 20 ns;
114.
115.     end process;
116.
117.     end Behavioral;
```

Código VHDL Test-Bench de Unidad aritmética / lógica de 4 bits.

```
1. library IEEE;
2. use IEEE.STD_LOGIC_1164.ALL;
3.
4. entity testALU4bits is
5. end testALU4bits;
6.
7. architecture Behavioral of testALU4bits is
8. component ALU4bits is
9.     Port ( a : in STD_LOGIC_VECTOR (3 downto 0);
10.               b : in STD_LOGIC_VECTOR (3 downto 0);
11.               Aluop : in STD_LOGIC_VECTOR (3 downto 0);
12.               Res : inout STD_LOGIC_VECTOR (3 downto 0);
13.               bandera : out STD_LOGIC_VECTOR(3 downto 0);
14.               Cout : out STD_LOGIC
15.             );
16.     end component;
17.
18.
19.     signal a : STD_LOGIC_VECTOR (3 downto 0);
20.     signal b : STD_LOGIC_VECTOR (3 downto 0);
21.     signal Aluop : STD_LOGIC_VECTOR (3 downto 0);
22.     signal Res : STD_LOGIC_VECTOR (3 downto 0);
23.     signal bandera : STD_LOGIC_VECTOR(3 downto 0);
24.     signal Cout : STD_LOGIC;
25.
26.     begin
27.     --mapear
28.
29.     ALUFULL: ALU4bits
30.         Port map ( a =>a,
31.                b =>b,
32.                Aluop =>Aluop,
33.                Res =>Res,
34.                bandera =>bandera,
35.                Cout =>Cout
36.              );
37.
38.     process
39.     begin
40.     --stimulators
41.         a<="0101";
42.         b<="1110";
43.         Aluop<="0011";
44.         wait for 20ns;
45.
46.         Aluop<="0111";
47.         wait for 20ns;
48.
49.         Aluop<="0000";
```

```
50.         wait for 20ns;
51.
52.         Aluop<="1101";
53.         wait for 20ns;
54.
55.         Aluop<="0001";
56.         wait for 20ns;
57.
58.         Aluop<="1100";
59.         wait for 20ns;
60.
61.         Aluop<="0010";
62.         wait for 20ns;
63.
64.         Aluop<="1010";
65.         wait for 20ns;
66.
67.         ------
68.         a<="0101";
69.         b<="0111";
70.         Aluop<="0011";
71.         wait for 20ns;
72.
73.         ---
74.         a<="0101";
75.         b<="0101";
76.         Aluop<="0111";
77.         wait for 20ns;
78.
79.         Aluop<="1101";
80.         wait for 20ns;
81.
82.         wait;
83.     end process;
84.
85.     end Behavioral;
```

Diagrama RTL.

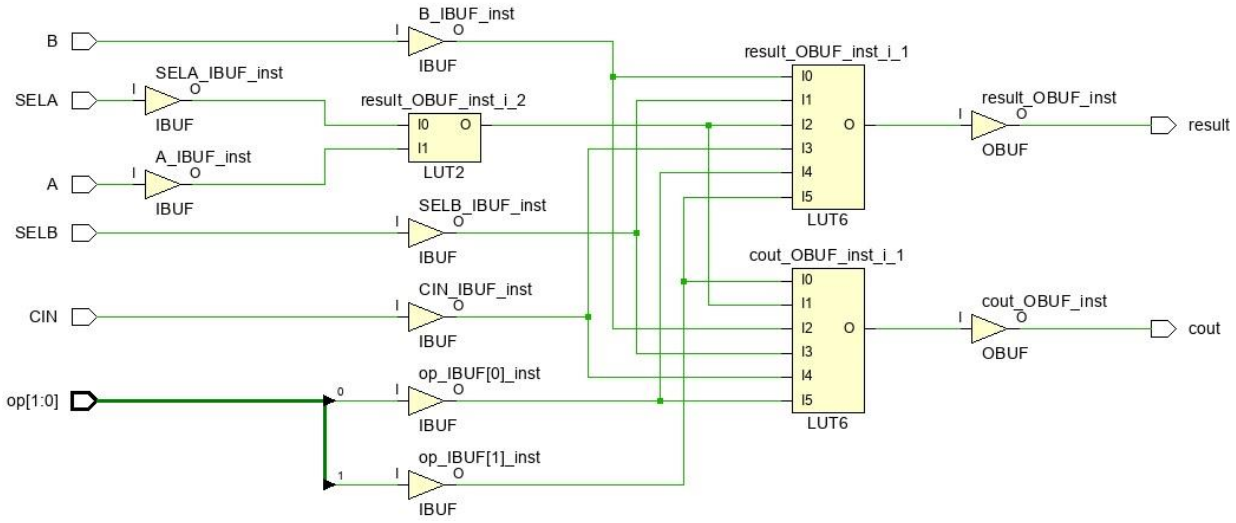

Figura 1. Circuito esquemático de la Unidad aritmética / lógica de 1 bit.

Diagrama RTL.

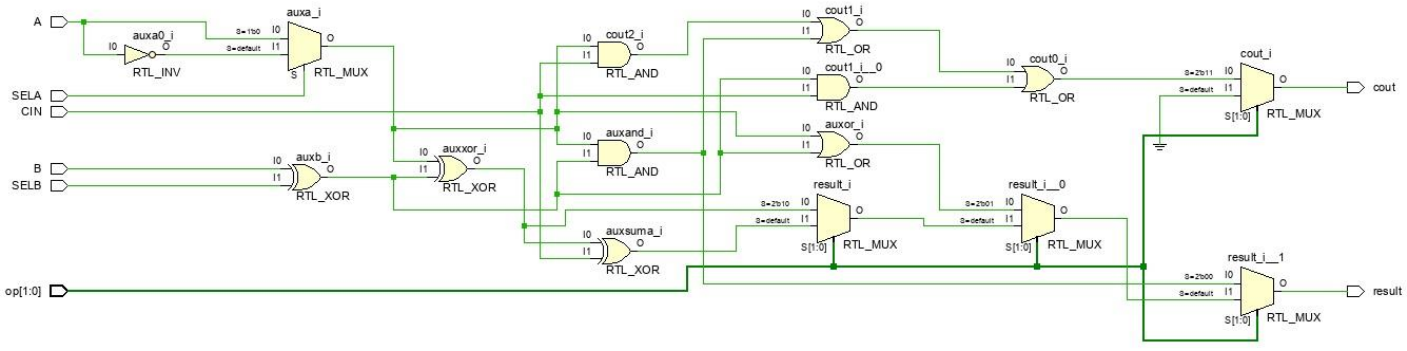

Figura 1.1. Circuito lógico de la Unidad aritmética / lógica de 1 bit.

Diagrama de Onda de la Unidad aritmética / lógica de 1 bit.

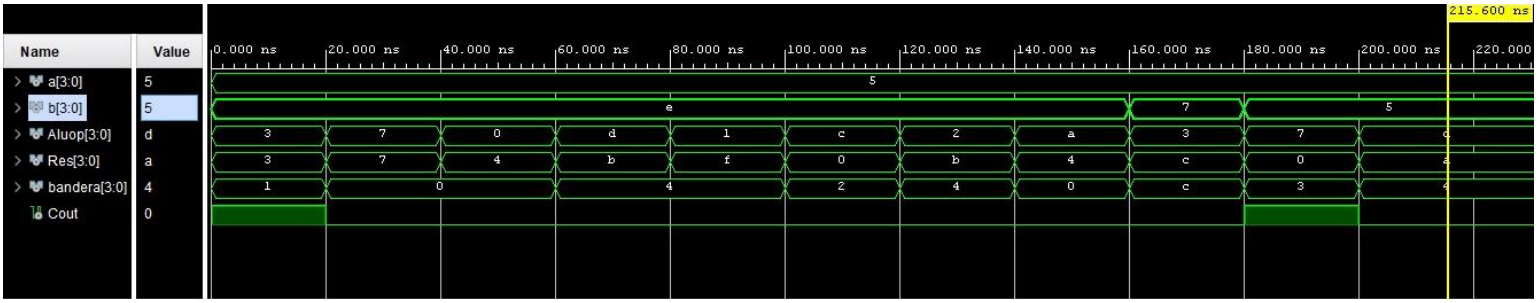

Figura 1.2 Simulación del Test-Bench de la Unidad aritmética / lógica de 1 bit.

Diagrama RTL.

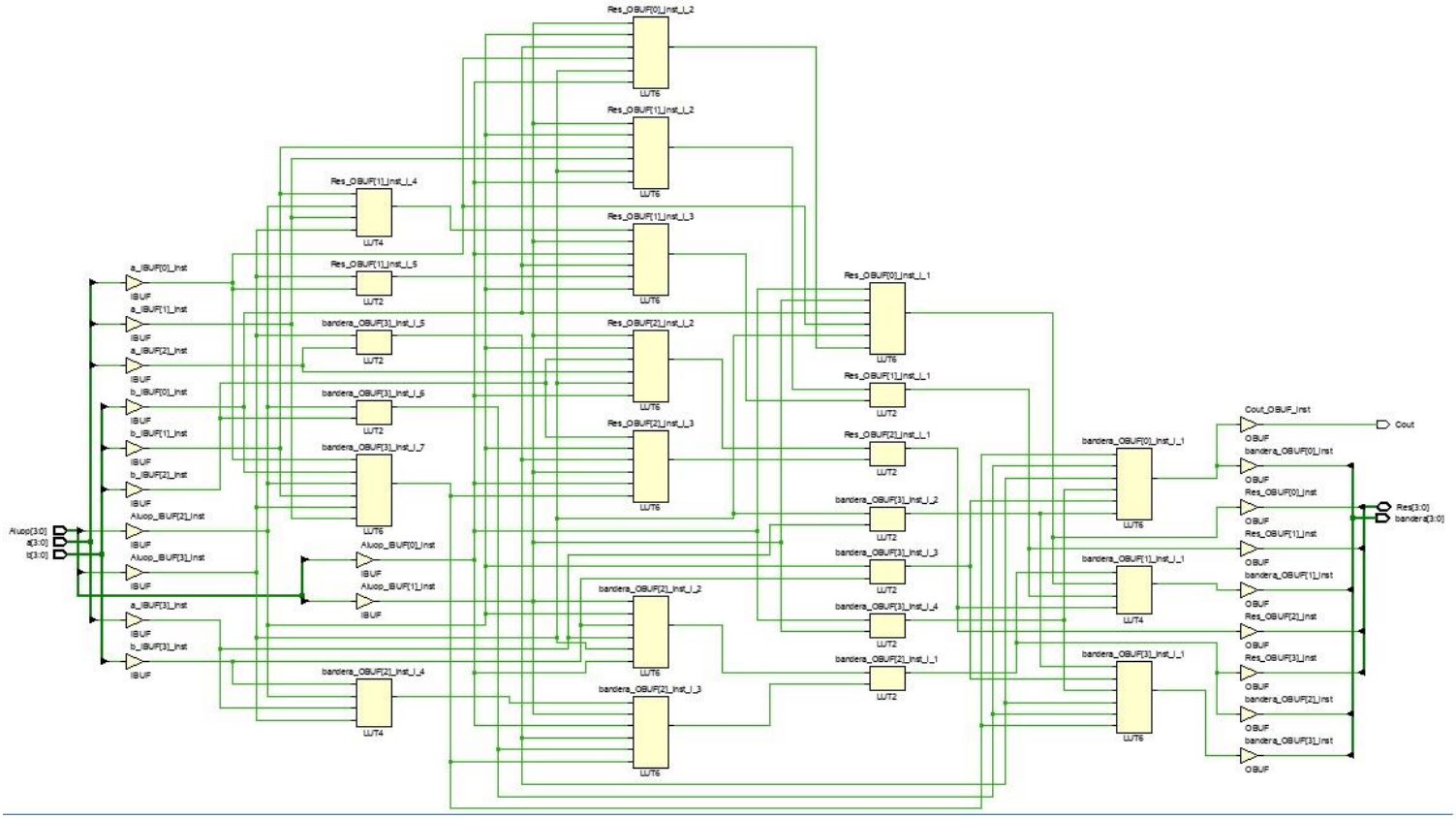

Figura 2. Circuito esquemático de la Unidad aritmética / lógica de 4 bits.

Diagrama RTL.

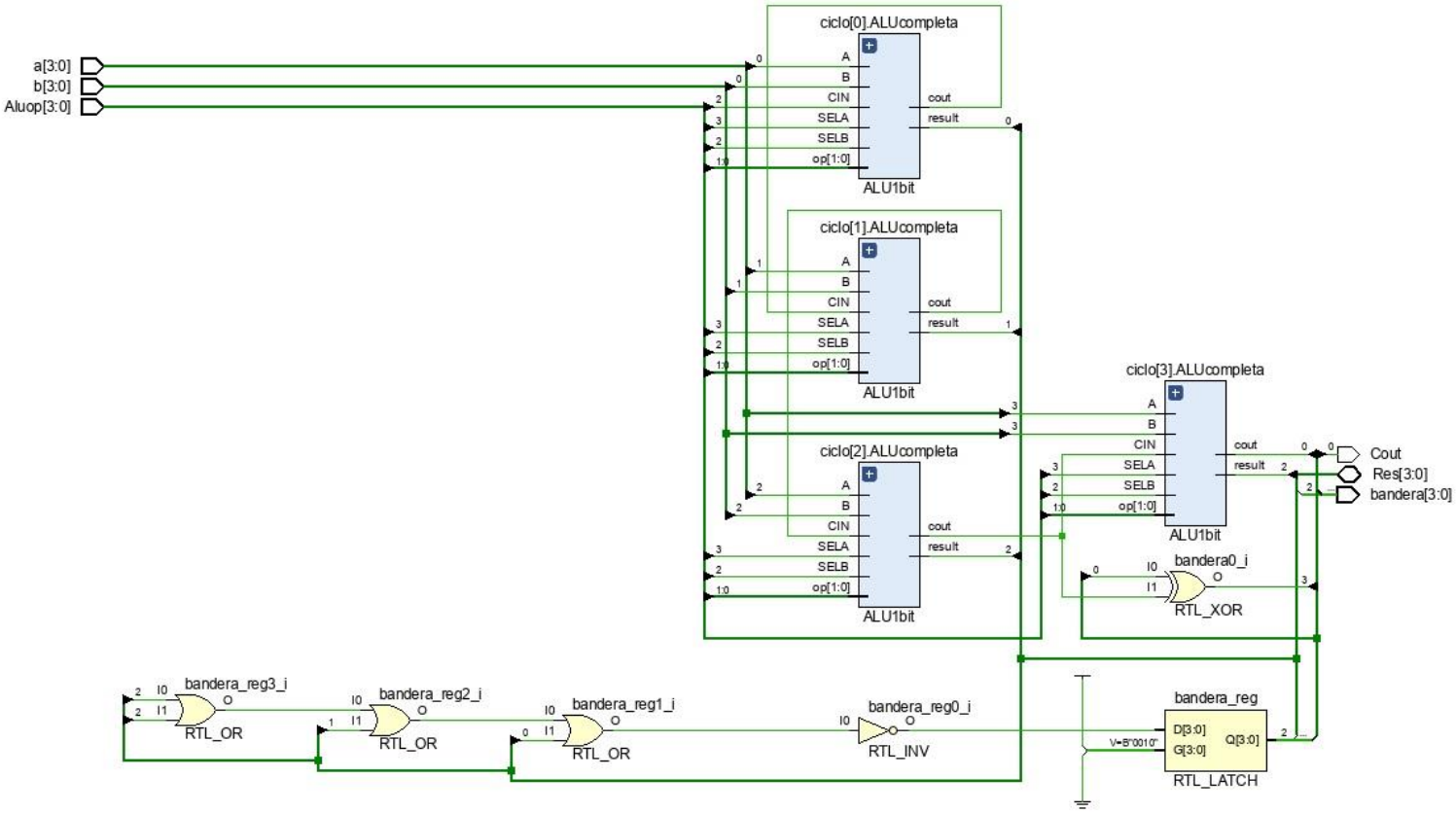

Figura 2.1 Circuito esquemático de la Unidad aritmética / lógica de 4 bits.

Diagrama RTL.

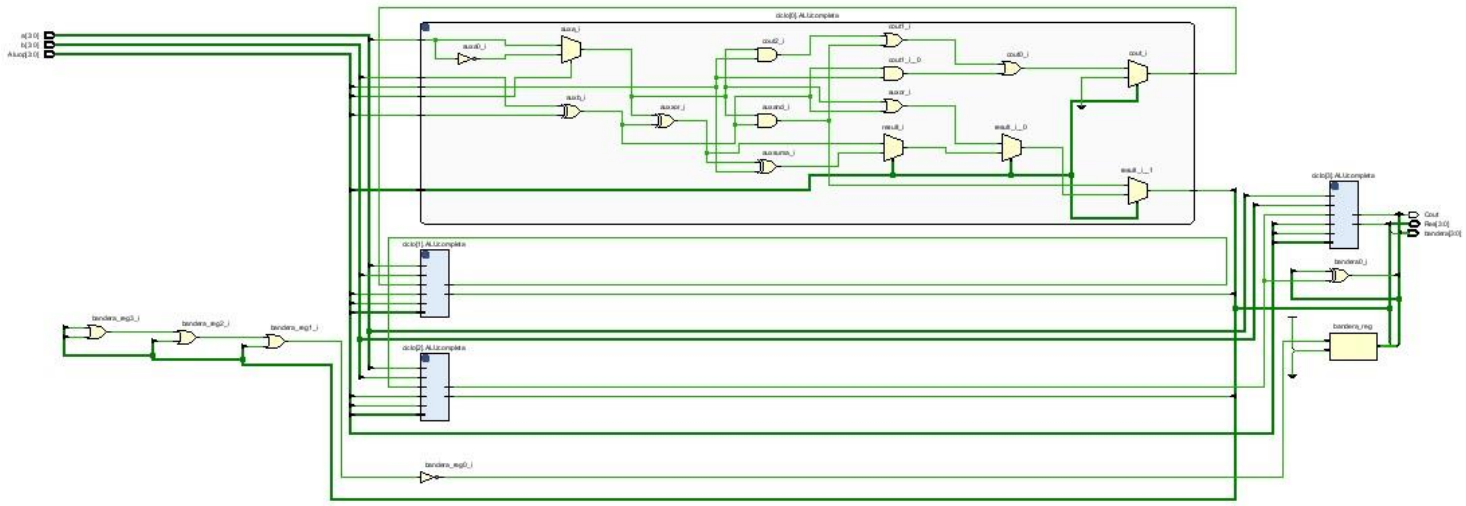

Figura 2.2 Circuito esquemático de la Unidad aritmética / lógica de 4 bits, vista de 1 ALU 1 bit.

Diagrama de Onda de la Unidad aritmética / lógica de 4 bits.

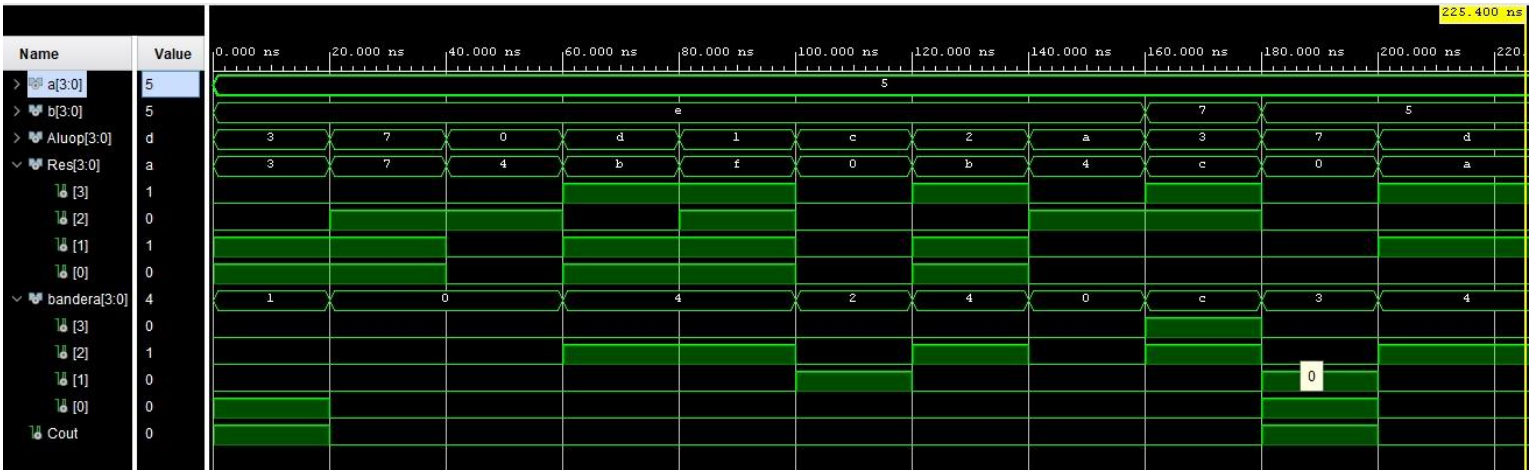

Figura 2.3 Simulación del Test-Bench de la Unidad aritmética / lógica de 4 bits.