Reporte de practica 4

González Pardo Adrian

Marzo 2020

1. Código VHDL

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


entity sumador is
    Port ( a,b,cin : in STD_LOGIC;
           s : out STD_LOGIC;
           cout : out STD_LOGIC);
end sumador;

architecture Behavioral of sumador is
begin
    s<=a xor b xor cin;
    cout <= (a and cin)or(b and cin)or(a and b);

end Behavioral;
```

Código fuente de sumador de 1 bit

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

entity alu1 is
    Port ( a, b, sela, selb, cin : in STD_LOGIC;
           op : in STD_LOGIC_VECTOR (1 downto 0);
           res : out STD_LOGIC;
           cout : out STD_LOGIC);
end alu1;

architecture Behavioral of alu1 is

component sumador is
    Port ( a,b,cin : in STD_LOGIC;
           s : out STD_LOGIC;
           cout : out STD_LOGIC);
end component;
signal auxa, auxb, auxs, and1, or1, xor1 ,auxc: std_logic;
```

```vhdl
19 begin
20
21
22 auxa <= a xor sela;
23 auxb <= b xor selb;
24
25 and1 <= auxa and auxb;
26 or1 <= auxa or auxb;
27 xor1 <= auxa xor auxb;
28
29
30 sumador1: sumador
31     Port map(
32     a => auxa,
33     b => auxb,
34     cin => cin,
35     s => auxs,
36     cout => auxc
37     );
38
39      process(and1, or1, xor1, auxs, op)
40      begin
41         case op is
42             when "00" =>res <= and1;
43             cout<='0';
44             when "01" =>res <= or1;
45             cout<='0';
46             when "10" =>res <= xor1;
47             cout<='0';
48             when others =>res <= auxs;
49             cout<=auxc;
50         end case;
51      end process;
52
53
54
55
56
57
58 end Behavioral;
```

Código fuente de ALU de 1 bit

```vhdl
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity AluN is
5 generic(
6     m: integer:= 4);
7     Port ( a, b : in STD_LOGIC_VECTOR (m-1 downto 0);
8             aluop : in STD_LOGIC_VECTOR (3 downto 0);
9             res : out STD_LOGIC_VECTOR (m-1 downto 0);
10            n,z,ov,co : out STD_LOGIC
11            );
```

```vhdl
12 end AluN;
13
14 architecture Behavioral of AluN is
15 component alu1 is
16     Port ( a, b, sela, selb, cin : in STD_LOGIC;
17           op : in STD_LOGIC_VECTOR (1 downto 0);
18           res : out STD_LOGIC;
19           cout : out STD_LOGIC);
20 end component;
21 signal c: std_logic_vector(m downto 0);
22 signal aux: std_logic_vector(m-1 downto 0);
23
24 begin
25
26     c(0) <= aluop(2);
27     ciclo : for i in 0 to m-1 generate
28         bitA : alu1
29         Port  map(
30             a => a(i),
31             b => b(i),
32             sela => aluop(3),
33             selb => aluop(2),
34             cin => c(i),
35             op  => aluop(1 downto 0),
36             res => aux(i),
37             cout => c(i+1)
38         );
39     end generate;
40
41     process(aux,aluop,c)
42     variable auxT:std_logic;
43     begin
44         auxT:='0';
45         for i in 0 to m-1 loop
46             auxT:=auxT or aux(i);
47         end loop;
48         z <= not (auxT);
49         res <=aux;
50         co  <= c(m);
51         n  <= aux(m-1); --signo
52         --z <= '1' when aux = "0000" else '0'; --cero
53         ov <= c(m) xor c(m-1);--overflow
54     end process;
55
56
57 end Behavioral;
```

Código fuente de la ALU de m bits generalizada

2. Test-Bench VHDL Código

```vhdl
1 library IEEE;
```

```vhdl
use IEEE.STD_LOGIC_1164.ALL;
entity simAlumN is
    generic(
        m: integer:=4);
-- Port ( );
end simAlumN;

architecture Behavioral of simAlumN is
    component AluN
        Port ( a, b : in STD_LOGIC_VECTOR (m-1 downto 0);
               aluop : in STD_LOGIC_VECTOR (3 downto 0);
               res : out STD_LOGIC_VECTOR (m-1 downto 0);
               n,z,ov,co : out STD_LOGIC
             );
    end component;
    signal a:STD_LOGIC_VECTOR (m-1 downto 0):=x"0";
    signal b:STD_LOGIC_VECTOR (m-1 downto 0):=x"0";
    signal aluop:STD_LOGIC_VECTOR (3 downto 0):=x"0";
    signal res :STD_LOGIC_VECTOR (m-1 downto 0):=x"0";
    signal ov:STD_LOGIC:='0';
    signal n:STD_LOGIC:='0';
    signal z:STD_LOGIC:='0';
    signal co:STD_LOGIC:='0';
begin
    alu:AluN
        Port map(
            a=>a,
            b=>b,
            aluop=>aluop,
            res=>res,
            n=>n,
            z=>z,
            ov=>ov,
            co=>co
        );
    p3:process
    begin
    --Suma
    a<=x"5";
    b<="1110";
    aluop<="0011";
    wait for 10 ns;
    --Resta
    aluop<="0111";
    wait for 10 ns;
    --AND
    aluop<="0000";
    wait for 10 ns;
    --NAND
    aluop<="1101";
    wait for 10 ns;
    --OR
    aluop<="0001";
```
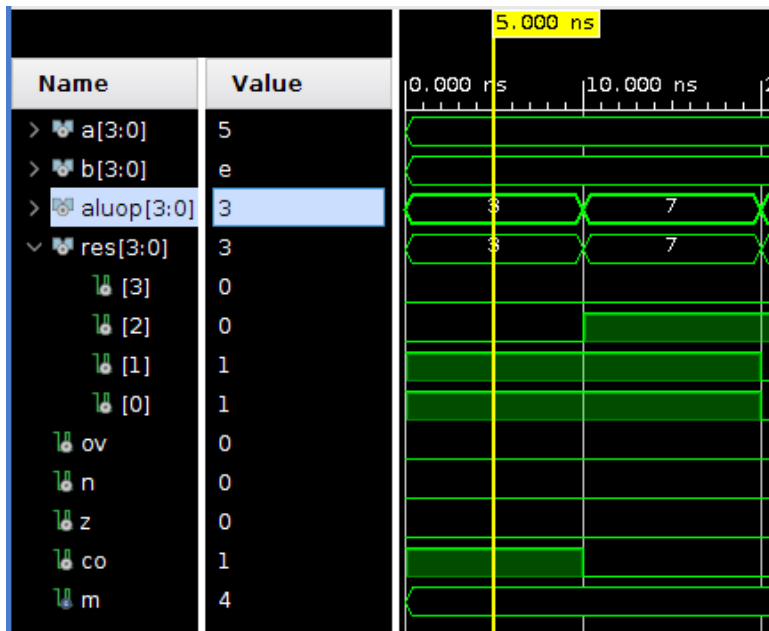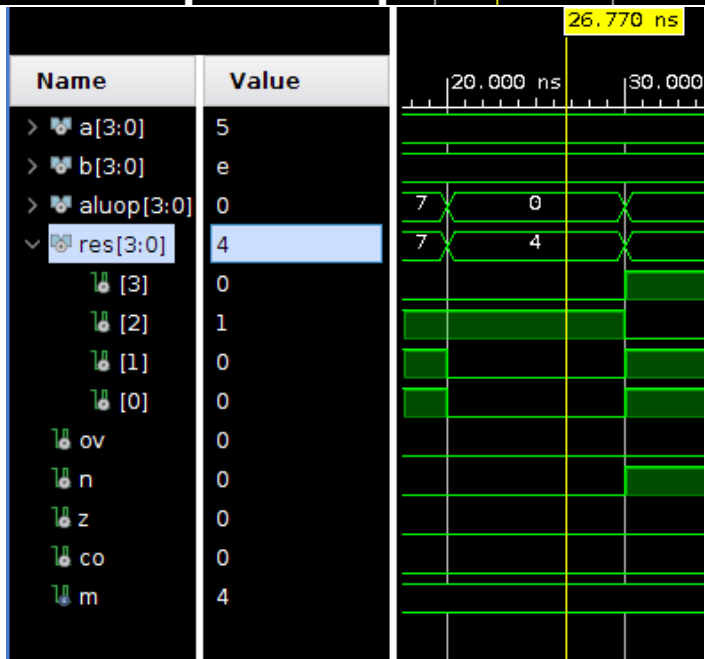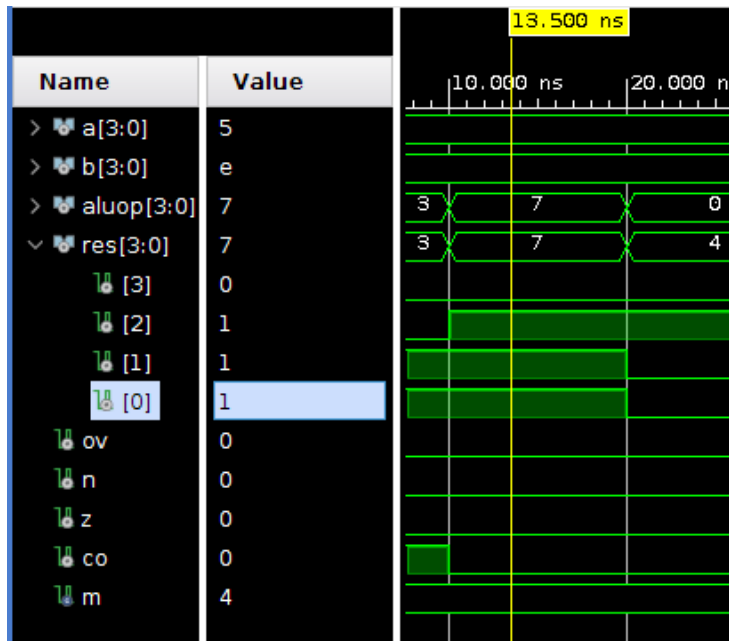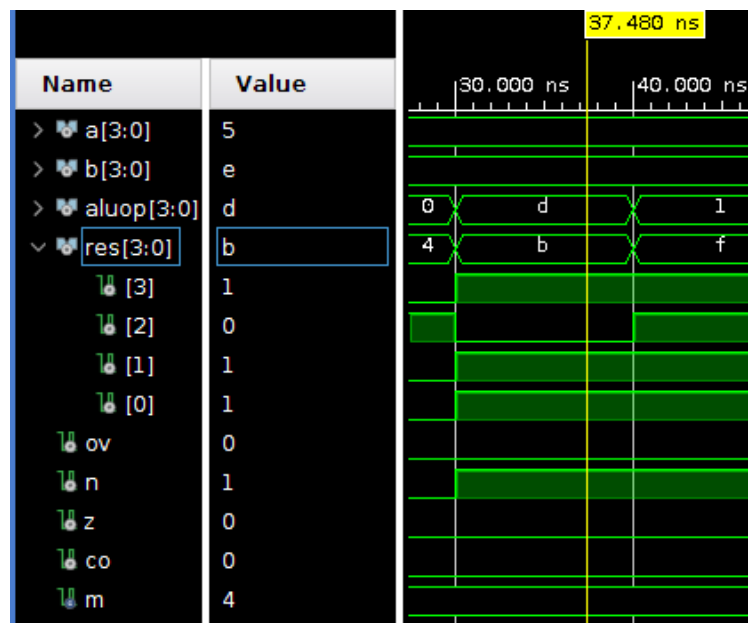
```
55      wait for 10 ns;
56      --NOR
57      aluop<="1100";
58      wait for 10 ns;
59      --XOR
60      aluop<="0010";
61      wait for 10 ns;
62      --XNOR
63      aluop<="1010";
64      wait for 10 ns;
65      --B=7 Suma
66      b<=x"7";
67      aluop<="0011";
68      wait for 10 ns;
69      --Resta
70      b<=x"5";
71      aluop<="0111";
72      wait for 10 ns;
73      --Nand
74      aluop<="1101";
75      wait for 10 ns;
76      wait;
77      end process;
78 end Behavioral;
```

Código de simulación

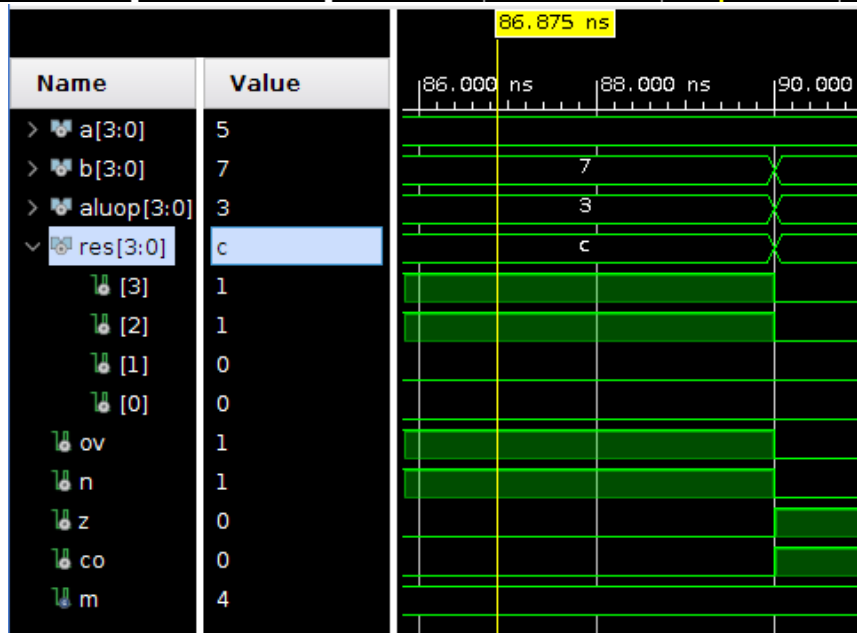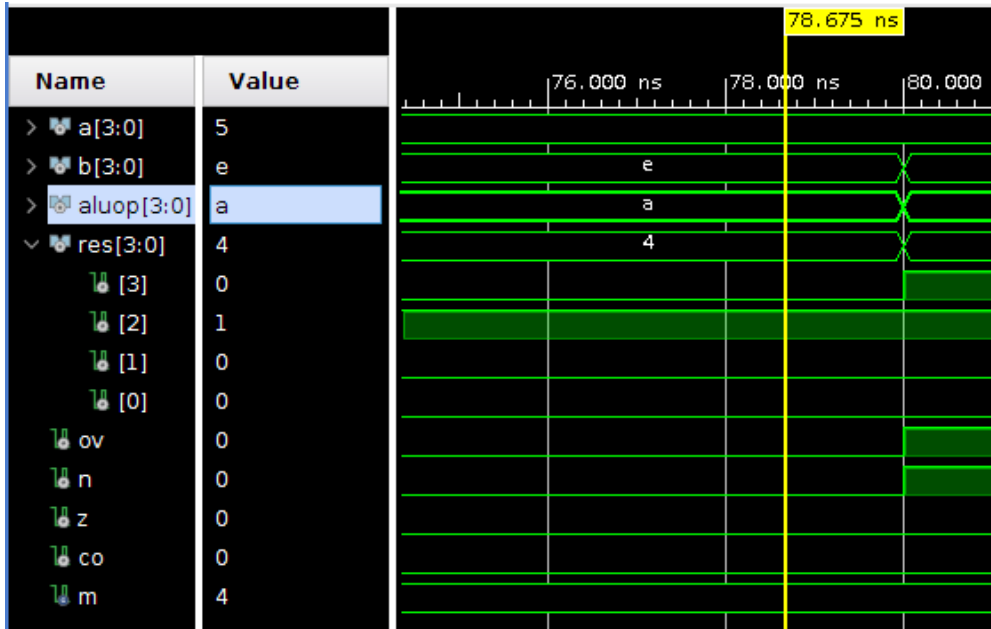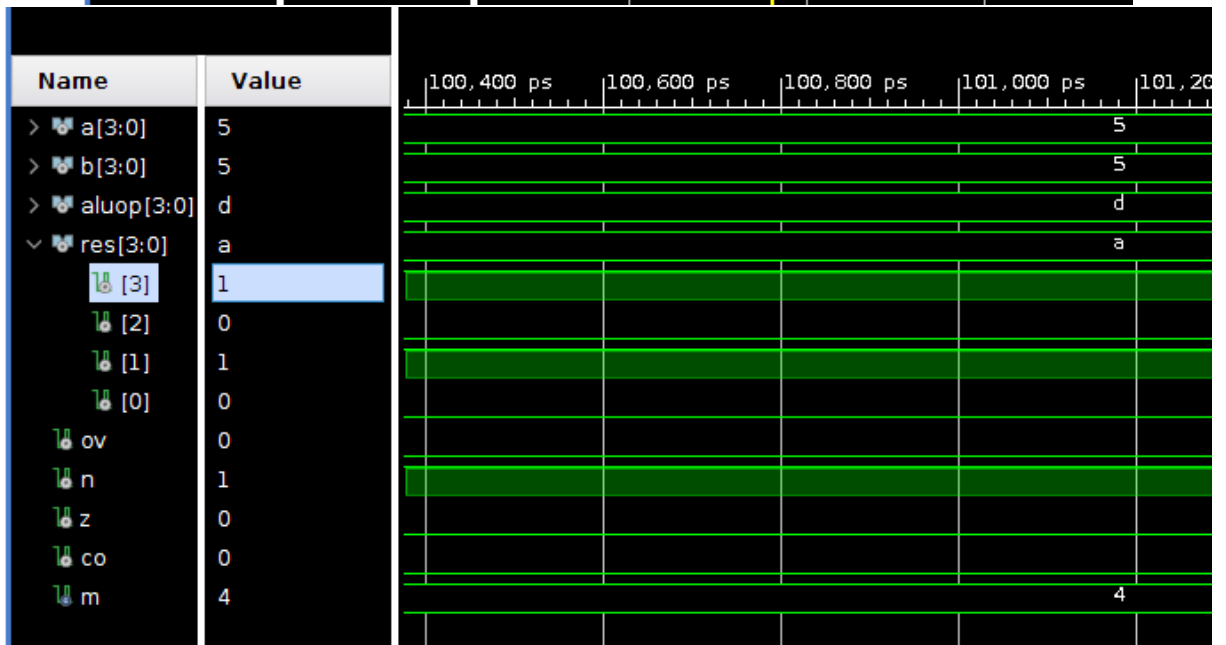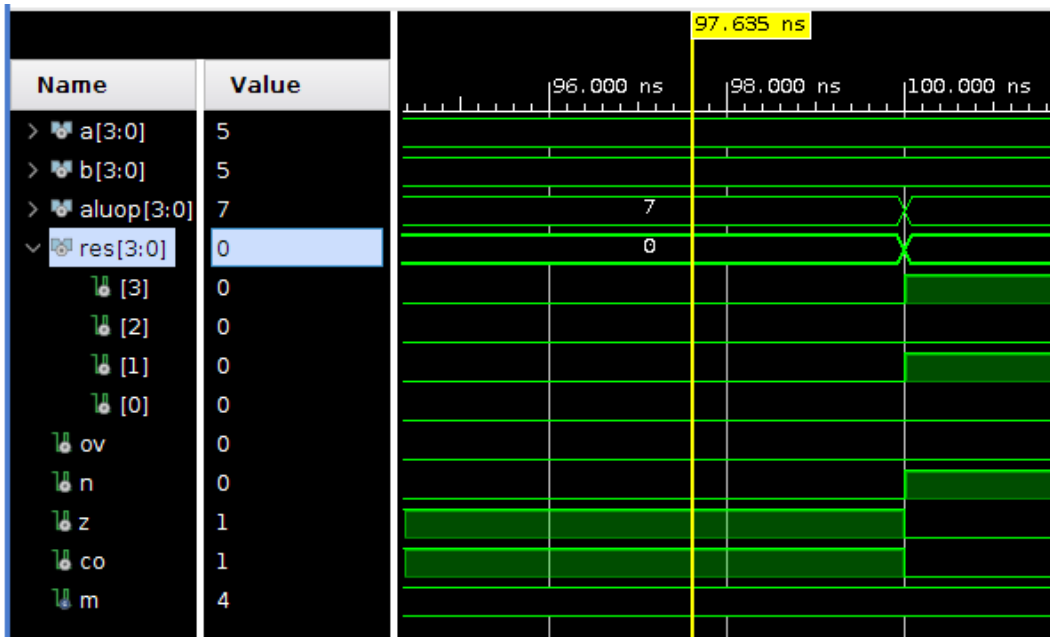**Anexo de fotos de la simulación a impulsos**

6

3. Tabla de resultados

Estado de Banderas				Operación				
				1	0	0	0	Cn
				0	1	0	1	A = 5
OV	N	Z	C	1	1	1	0	B = -2
0	0	0	1	0	0	1	1	A+B
0	0	0	0	0	1	1	1	A-B
0	0	0	0	0	1	0	0	AND
0	1	0	0	1	0	1	1	NAND
0	1	0	0	1	1	1	1	OR
0	0	1	0	0	0	0	0	NOR
0	1	0	0	1	0	1	1	XOR
0	0	0	0	0	1	0	0	XNOR
				1	1	1	0	Cn
OV	N	Z	C	0	1	0	1	A = 5
				0	1	1	1	B = 7
1	1	0	0	1	1	0	0	A+B
				1	1	1	0	Cn
OV	N	Z	C	0	1	0	1	A
				0	1	0	1	B
0	0	1	1	0	0	0	0	A-B
0	1	0	0	1	0	1	0	NAND (NOT)

Figura: Tabla que concuerda con los resultados obtenidos en la simulación
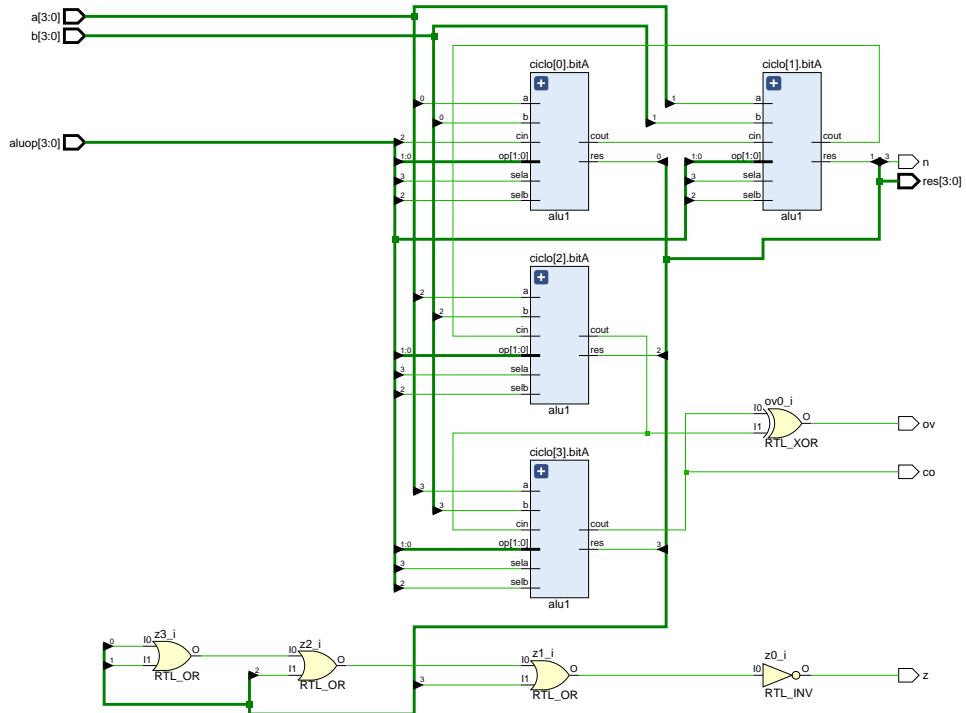
4. Diagrama RTL
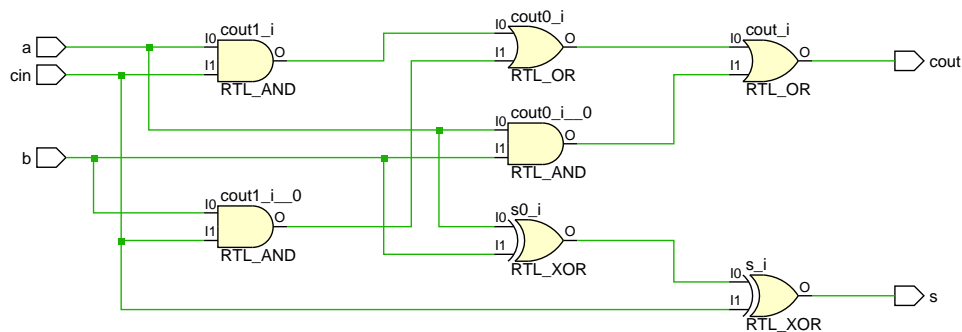
Diagrama RTL del archivo VHDL del sumador de 1 bit
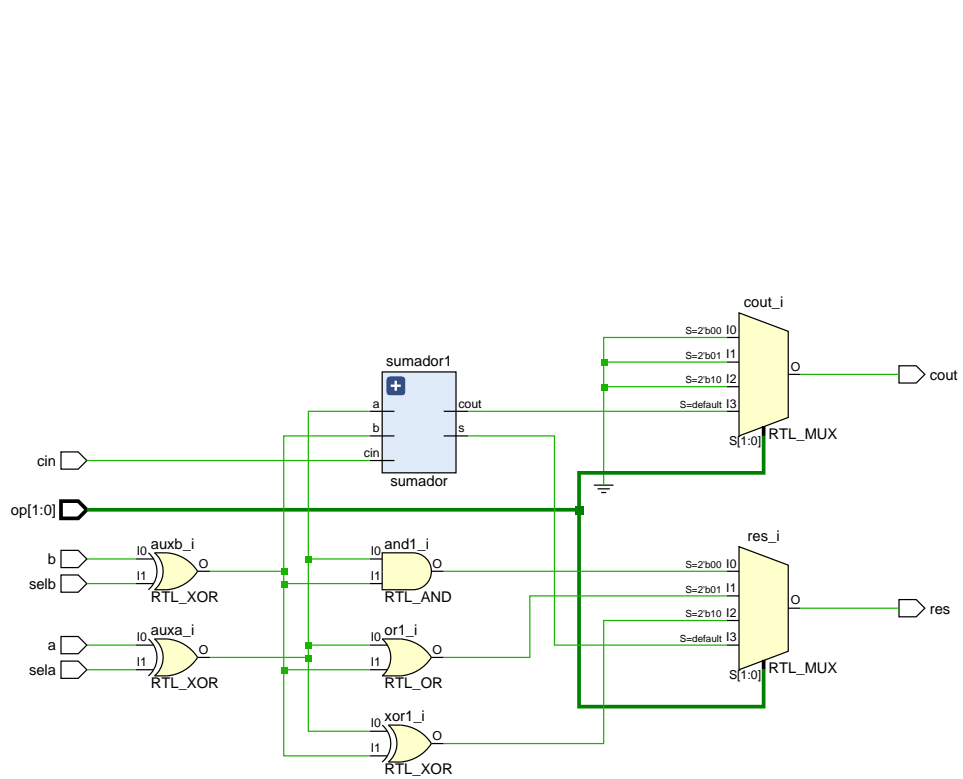
Diagrama RTL del archivo VHDL de la ALU de 1 bit

Diagrama RTL del archivo VHDL de la ALU de m bits