Laboratorio VHDL

Hoja de respuestas del laboratorio “Arithmetic Circuits and Testbenches”

Asignatura: DSED

Número de grupo: 1

Nombres y apellidos de los miembros del grupo: Álvaro Escribano y Guillermo Pagés

Copia y pega el contenido del fichero lab2_1_1:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity lab2_1_1 is
   Port ( a_in : in STD_LOGIC_VECTOR (3 downto 0);
          b_in : in STD_LOGIC_VECTOR (3 downto 0);
          c_in : in STD_LOGIC;
          c_out : out STD_LOGIC;
          s_out : out STD_LOGIC_VECTOR (3 downto 0));
end lab2_1_1;
architecture Behavioral of lab2_1_1 is
   component fulladder_dataflow
      Port ( a : in STD_LOGIC;
          b : in STD_LOGIC;
          cin : in STD_LOGIC;
          s : out STD_LOGIC;
          cout : out STD_LOGIC);
    end component;
   signal c1, c2, c3 : STD_LOGIC;
begin
   sum0: fulladder_dataflow Port map(a_in(0),b_in(0),c_in,s_out(0),c1);
   sum1: fulladder_dataflow Port map(a_in(1),b_in(1),c1,s_out(1),c2);
   sum2: fulladder_dataflow Port map(a_in(2),b_in(2),c2,s_out(2),c3);
   sum3: fulladder_dataflow Port map(a_in(3),b_in(3),c3,s_out(3),c_out);
end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity fulladder_dataflow is
   Port ( a : in STD_LOGIC;
          b : in STD_LOGIC;
          cin : in STD_LOGIC;
          s : out STD_LOGIC;
          cout : out STD_LOGIC);
end fulladder_dataflow;
architecture Behavioral of fulladder_dataflow is
begin
   s <= a xor b xor cin;
   cout <= (a and b) or (a and cin) or (b and cin);
end Behavioral;
```

Copia y pega el contenido del fichero lab2_1_2:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity lab2_1_2 is
   Port ( a : in STD_LOGIC_VECTOR (3 downto 0);
          b : in STD_LOGIC_VECTOR (3 downto 0);
          c : in STD_LOGIC;
          an_out : out STD_LOGIC_VECTOR (7 downto 0);
          led : out STD_LOGIC;
          seg7 : out STD_LOGIC_VECTOR (0 to 6));
end lab2_1_2;
architecture Behavioral of lab2_1_2 is
   component lab2_1_1
       Port ( a_in : in STD_LOGIC_VECTOR (3 downto 0);
          b_in : in STD_LOGIC_VECTOR (3 downto 0);
          c_in : in STD_LOGIC;
          c_out : out STD_LOGIC;
          s_out : out STD_LOGIC_VECTOR (3 downto 0);
          error_out: out STD_LOGIC);
    end component;
```

```
    component lab1_6_1_partA
        Port ( v : in STD_LOGIC_VECTOR (4 downto 0);
           error : in STD_LOGIC;
           z : out STD_LOGIC;
           m : out STD_LOGIC_VECTOR (3 downto 0));
    end component;
    component bcdto7segment_dataflow
        Port ( x : in STD_LOGIC_VECTOR (3 downto 0);
           an : out STD_LOGIC_VECTOR (7 downto 0);
           seg : out STD_LOGIC_VECTOR (0 to 6));
    end component;
    signal c1, s_error: STD_LOGIC;
    signal s1,s2 : STD_LOGIC_VECTOR (3 downto 0);
    signal aux : STD_LOGIC_VECTOR (4 downto 0);
begin
   suma: lab2_1_1 Port map(a,b,c,c1,s1,s_error);
   aux <= c1&s1;
   reparto: lab1_6_1_partA Port map(aux,s_error,led,s2);
   display: bcdto7segment_dataflow Port map(s2,an_out,seg7);
end Behavioral;
```

Copia y pega el contenido del fichero **lab2_1_3**:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity lab2_1_3 is
    Port ( a : in STD_LOGIC_VECTOR (3 downto 0);
       b : in STD_LOGIC_VECTOR (3 downto 0);
       c : in STD_LOGIC;
       an_out : out STD_LOGIC_VECTOR (7 downto 0);
       led : out STD_LOGIC;
       seg7 : out STD_LOGIC_VECTOR (0 to 6));
end lab2_1_3;
architecture Behavioral of lab2_1_3 is
    component lab2_1_1
        Port ( a_in : in STD_LOGIC_VECTOR (3 downto 0);
```

```
            b_in : in STD_LOGIC_VECTOR (3 downto 0);
            c_in : in STD_LOGIC;
            s_out : out STD_LOGIC_VECTOR (4 downto 0);
            error_out: out STD_LOGIC);
    end component;
    component lab1_6_1_partA
        Port ( v : in STD_LOGIC_VECTOR (4 downto 0);
             error : in STD_LOGIC;
             z : out STD_LOGIC;
             m : out STD_LOGIC_VECTOR (3 downto 0));
    end component;
    component bcdto7segment_dataflow
        Port ( x : in STD_LOGIC_VECTOR (3 downto 0);
            an : out STD_LOGIC_VECTOR (7 downto 0);
            seg : out STD_LOGIC_VECTOR (0 to 6));
    end component;
    signal c1, s_error: STD_LOGIC;
    signal s2 : STD_LOGIC_VECTOR (3 downto 0);
    signal s1 : STD_LOGIC_VECTOR (4 downto 0);
begin
    suma: lab2_1_1 Port map(a,b,c,s1,s_error);
    reparto: lab1_6_1_partA Port map(s1,s_error,led,s2);
    display: bcdto7segment_dataflow Port map(s2,an_out,seg7);
end Behavioral;
```

Copia y pega el contenido del fichero lab2_3_1:

```
entity lab2_3_1 is
    Port ( a : in STD_LOGIC_VECTOR (1 downto 0);
         b : in STD_LOGIC_VECTOR (1 downto 0);
         ROM_data : out STD_LOGIC_VECTOR (3 downto 0));
end lab2_3_1;
architecture Behavioral of lab2_3_1 is
type rom is array (0 to 2**4-1) of std_logic_vector(3 downto 0);
constant MY_ROM : rom :=(
    0=>"0000",
    1=>"0001",
```

```
   2=>"0010",
   3=>"0011",
   4=>"0100",
   5=>"0101",
   6=>"0110",
   7=>"0111",
   8=>"1000",
   9=>"1001",
   10=>"0000",
   11=>"0000",
   12=>"0000",
   13=>"0000",
   14=>"0000",
   15=>"0000"
 );
 signal mult: std_logic_vector(3 downto 0);
begin
process(a,b)
   begin
      mult<=a*b;
      case mult is
         when "0000" => ROM_data <= MY_ROM(0);
         when "0001" => ROM_data <= MY_ROM(1);
         when "0010" => ROM_data <= MY_ROM(2);
         when "0011" => ROM_data <= MY_ROM(3);
         when "0100" => ROM_data <= MY_ROM(4);
         when "0101" => ROM_data <= MY_ROM(5);
         when "0110" => ROM_data <= MY_ROM(6);
         when "0111" => ROM_data <= MY_ROM(7);
         when "1000" => ROM_data <= MY_ROM(8);
         when "1001" => ROM_data <= MY_ROM(9);
         when others => ROM_data <= "0000";
       end case;
   end process;
end Behavioral;
```

Copia y pega el contenido del fichero lab2_4_2 1 así como una captura de pantalla del resultado del testbench:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity lab2_4_2 is
end lab2_4_2;
architecture Behavioral of lab2_4_2 is
   signal A, G1 : std_logic := '0';
   signal G2 : std_logic := '1';
begin
process
begin
   wait for 40 ns; A <= '1';
   wait for 20 ns; G1 <= '1';
   wait for 20 ns; G2 <= '0';
   wait for 20 ns; A <= '0';
   wait for 20 ns; G1 <= '0';
   wait for 20 ns; G2 <= '1';
   wait for 100 ns;
end process;
end Behavioral;
```