# 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;

