# Laboratorio VHDL

## Hoja de respuestas del laboratorio “Timing Constraints, Architectural Wizard, IP Catalog and Real-Time Clock”

Asignatura: DSED

Número de grupo: 1

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

**2-1: Haz una demo del comportamiento de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

entity lab4\_1\_1 is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

en : in STD\_LOGIC;

sig : out STD\_LOGIC;

MMCM : out STD\_LOGIC);

end lab4\_1\_1;

architecture Behavioral of lab4\_1\_1 is

Signal clk5 : STD\_LOGIC;

Signal counter : unsigned(21 downto 0) := (others=>'0');

Signal salida : STD\_LOGIC := '0';

component clk\_wiz\_0

Port( clk\_out1 : out STD\_LOGIC;

reset: in STD\_LOGIC;

locked: out STD\_LOGIC;

clk\_in : in STD\_LOGIC);

end component;

begin

Cach: clk\_wiz\_0 Port map(clk5,rst,MMCM,clk);

process(clk5)

begin

if rst = '1' then

counter <= (others=>'0');

salida <= '0';

elsif rising\_edge(clk5) and en = '1' then

counter <= counter+1;

if counter >= 2499999 then

salida <= not salida;

counter <= (others=>'0');

end if;

end if;

end process;

sig <= salida;

end Behavioral;

**2-2: Haz una demo del comportamiento de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

entity bcdto7segment\_dataflow is

Port ( clk : in STD\_LOGIC;

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 bcdto7segment\_dataflow;

architecture Behavioral of bcdto7segment\_dataflow is

signal counter : unsigned(12 downto 0) := (others=>'0');

signal an\_s : STD\_LOGIC\_VECTOR (7 downto 0) := "11111101";

begin

process(clk)

begin

if rising\_edge(clk) then

counter <= counter+1;

if counter = 4999 and an\_s = "11111110" then

an\_s <= "11111101";

counter <= (others=>'0');

elsif counter = 4999 and an\_s = "11111101" then

an\_s <= "11111110";

counter <= (others=>'0');

end if;

end if;

end process;

an <= an\_s;

process(x)

begin

if an\_s = "11111101" then

if x > "1001" then

seg <= "1001111";

else

seg <= "0000001";

end if;

else

if(x="0000") then

seg<="0000001";

elsif(x="0001") then

seg<="1001111";

elsif(x="0010") then

seg<="0010010";

elsif(x="0011") then

seg<="0000110";

elsif(x="0100") then

seg<="1001100";

elsif(x="0101") then

seg<="0100100";

elsif(x="0110") then

seg<="0100000";

elsif(x="0111") then

seg<="0001111";

elsif(x="1000") then

seg<="0000000";

elsif(x="1001") then

seg<="0000100";

elsif(x="1010") then

seg<="0000001";

elsif(x="1011") then

seg<="1001111";

elsif(x="1100") then

seg<="0010010";

elsif(x="1101") then

seg<="0000110";

elsif(x="1110") then

seg<="1001100";

elsif(x="1111") then

seg<="0100100";

end if;

end if;

end process;

end Behavioral;

entity lab1\_6\_1 is

Port ( clk : in STD\_LOGIC;

v\_i : in STD\_LOGIC\_VECTOR (3 downto 0);

seg\_o : out STD\_LOGIC\_VECTOR (0 to 6);

an\_o: out STD\_LOGIC\_VECTOR (7 downto 0));

end lab1\_6\_1;

architecture Behavioral of lab1\_6\_1 is

component bcdto7segment\_dataflow is

Port ( clk : in STD\_LOGIC;

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;

component clk\_wiz\_0 is

Port (clk\_out1 : out STD\_LOGIC;

locked : out STD\_LOGIC;

clk\_in : in STD\_LOGIC);

end component;

signal clk5 : STD\_LOGIC;

signal lol: STD\_LOGIC := '1';

begin

reloj : clk\_wiz\_0 Port map(clk5,lol,clk);

bcd7seg : bcdto7segment\_dataflow Port map(clk5,v\_i,an\_o,seg\_o);

end Behavioral;

**3-1: Haz una demo del comportamiento de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

entity lab4\_3\_1 is

Port ( clk : in STD\_LOGIC;

en : in STD\_LOGIC;

rst: in STD\_LOGIC;

q\_out : out STD\_LOGIC\_VECTOR (7 downto 0));

end lab4\_3\_1;

architecture Behavioral of lab4\_3\_1 is

component clk\_wiz\_0 is

Port (clk\_5MHz : out STD\_LOGIC;

clk\_in : in STD\_LOGIC);

end component;

component c\_counter\_binary\_0 is

Port (clk : in STD\_LOGIC;

ce : in STD\_LOGIC;

sclr : in STD\_LOGIC;

thresh0 : out STD\_LOGIC;

q : out STD\_LOGIC\_VECTOR(3 downto 0));

end component;

component c\_counter\_binary\_1 is

Port (clk : in STD\_LOGIC;

ce : in STD\_LOGIC;

sclr : in STD\_LOGIC;

q : out STD\_LOGIC\_VECTOR(3 downto 0));

end component;

signal clk5 : STD\_LOGIC;

signal clk1 : STD\_LOGIC := '0';

signal clk10 : STD\_LOGIC := '0';

signal counteru : unsigned(21 downto 0) := (others=>'0');

signal qoutu: STD\_LOGIC\_VECTOR(3 downto 0);

signal qoutd: STD\_LOGIC\_VECTOR(3 downto 0);

signal threshold: STD\_LOGIC;

signal en2: STD\_LOGIC;

begin

process(clk5)

begin

if rising\_edge(clk5) then

if rst = '1' then

counteru <= (others=>'0');

clk1 <= '0';

end if;

counteru <= counteru+1;

if counteru = 2499999 then

clk1 <= not(clk1);

counteru <= (others=>'0');

end if;

end if;

end process;

reloj : clk\_wiz\_0 Port map(clk5,clk);

counteruds : c\_counter\_binary\_0 Port map(clk1,en,rst,threshold,qoutu);

en2 <= en and threshold;

counterds : c\_counter\_binary\_1 Port map(clk1,en2,rst,qoutd);

q\_out<= qoutd & qoutu;

end Behavioral;

**4-1: Muestra la simulación de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

**5-1: Obtén la siguiente información del “Project Summary Tab”:**

Number of BUFG/BUFGCTRL: 2

Number of slices used: 13

Number of registers used: 32

Number of DSP48A1 slices used: 0

Number of IOs used: 12

**5-2: Obtén la siguiente información del “Project Summary Tab”:**

Number of BUFG/BUFGCTRL:2

Number of slices used:5

Number of registers used:1

Number of DSP48A1 slices used:2

Number of IOs used:12

**5-3: Obtén la siguiente información del “Project Summary Tab”:**

Number of BUFG/BUFGCTRL:2

Number of slices used:12

Number of registers used:32

Number of DSP48A1 slices used:0

Number of IOs used:12

**5-4: Obtén la siguiente información del “Project Summary Tab”:**

Number of BUFG/BUFGCTRL:2

Number of slices used:6

Number of registers used:9

Number of DSP48A1 slices used:1

Number of IOs used:12

**6-1: Muestra la simulación de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

Sube este fichero buzón del Moodle.