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.