

EL253 - Sistemas Digitales

Semestre 2022-1

Profesor Kalun José Lau Gan

Sesión de Laboratorio Semana 4

1

Preguntas Previas

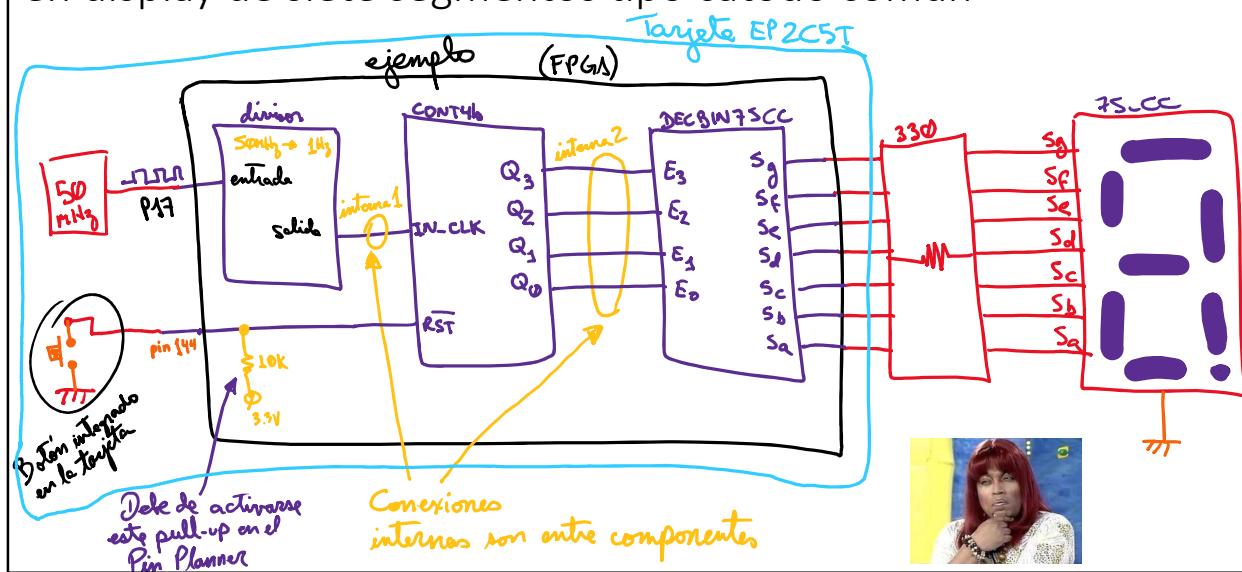
2

Agenda

- Contadores en VHDL
- Divisor de frecuencia
- Estilo de descripción estructural en VHDL (conexión de componentes)
- Implementación de prototipo en el FPGA EP2C5T144

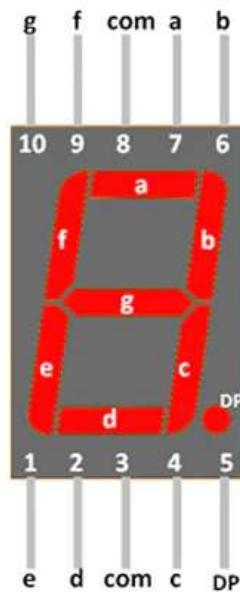
3

Ejemplo: Contador binario de 4 bits autoincremental con periodo de 1 segundo, con RESET y visualización de la cuenta en display de siete segmentos tipo cátodo común



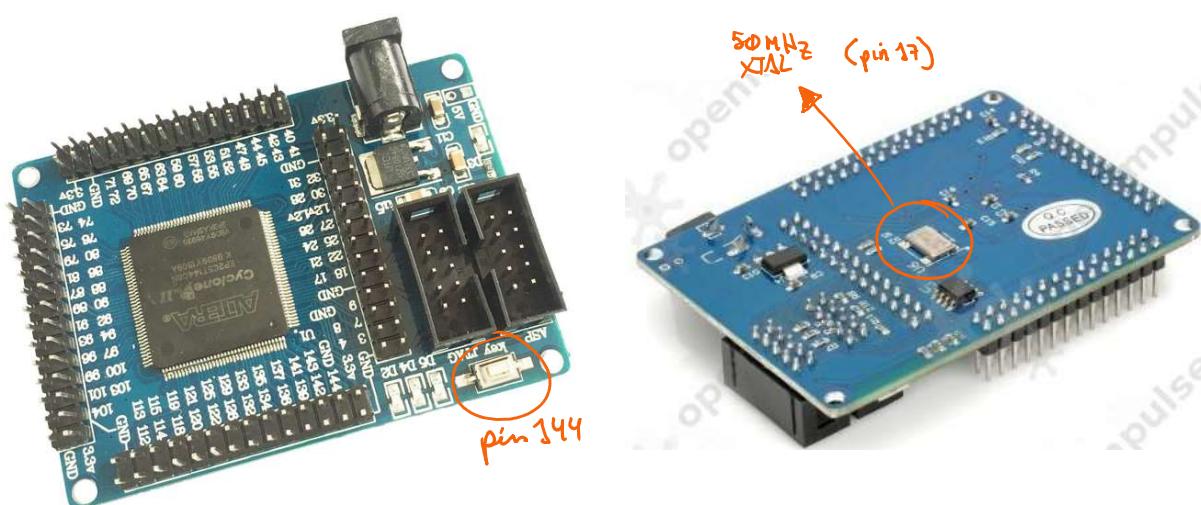
4

Diagrama de pines del display de siete segmentos:



5

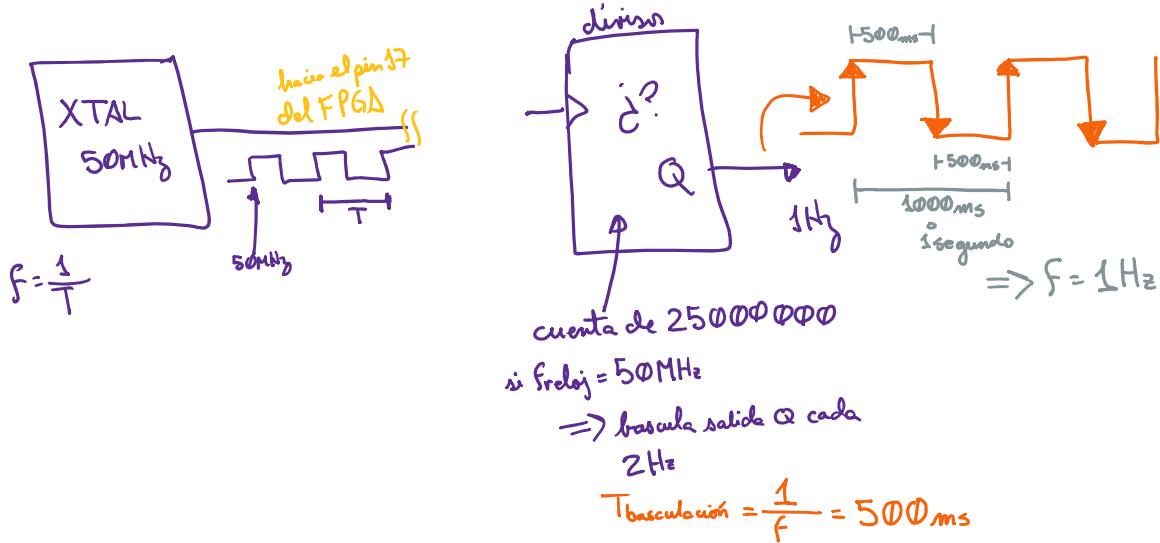
La tarjeta EP2C5T144



http://land-boards.com/blwiki/index.php?title=Cyclone_II_EP2C5_Mini_Dev_Board

6

Funcionamiento del divisor de frecuencia



7

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

- Crear el proyecto con nombre ejemplo.
- Crear los archivos *.vhd para los circuitos divisor, cont4b y decbin7scc, tener en cuenta que el nombre del archivo debe de coincidir con el nombre de la entidad.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity divisor is
7   port( entrada: in std_logic;
8         salida: out std_logic);
9 end divisor;
10
11 architecture funcionamiento of divisor is
12 signal cuenta_interna: std_logic_vector(24 downto 0);
13 signal basculacion: std_logic;
14 begin
15   process(entrada)
16   begin
17     if rising_edge(entrada) then
18       cuenta_interna <= cuenta_interna + 1;
19     if cuenta_interna = 2500000 then
20       basculacion <= not basculacion;
21       cuenta_interna <= (others => '0');
22     end if;
23   end process;
24   salida <= basculacion;
25 end funcionamiento;
26

```

```

1 library IEEE;
2 use IEEE.std_logic_arith.all;
3 use IEEE.std_logic_1164.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity decbin7scc is
7   port( E: in std_logic_vector(3 downto 0);
8         Sa, Sb, Sc, Sd, Se, Sf, Sg:  out std_logic);
9 end decbin7scc;
10
11 architecture combinacion of decbin7scc is
12 signal interno:  std_logic_vector(6 downto 0);
13 begin
14   with E select
15     interno <= "0111111" when "0000",
16                  "0000110" when "0001",
17                  "1011011" when "0010",
18                  "1001111" when "0011",
19                  "1100110" when "0100",
20                  "1101101" when "0101",
21                  "1111101" when "0110",
22                  "0000111" when "0111",
23                  "1111111" when "1000",
24                  "1100111" when "1001",
25                  "1110111" when "1010",
26                  "1111000" when "1011",
27                  "0111001" when "1100",
28                  "1011110" when "1101",
29                  "1111001" when "1110",
30                  "1100001" when "1111",
31                  "ZZZZZZZ" when others;
32   Sa <= interno(0);  --conexiones a las salidas
33   Sb <= interno(1);
34   Sc <= interno(2);
35   Sd <= interno(3);
36   Se <= interno(4);
37   Sf <= interno(5);
38   Sg <= interno(6);

```

8

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

- Crear el archivo ejemplo.vhd que llamará a DIVISOR, CONT4B y DECBIN7SCC como componentes y establecerá las conexiones entre ellos.

```

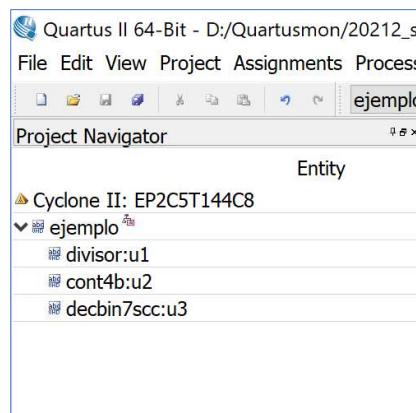
1 library IEEE;
2 use IEEE.std_logic_arith.all;
3 use IEEE.std_logic_1164.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity ejemplo is
7   port( xtal_50M, nRST:  in std_logic;
8         Sa, Sb, Sc, Sd, Se, Sf, Sg:  out std_logic);
9 end ejemplo;
10
11 architecture funcionamiento of ejemplo is
12   signal interno1:  std_logic;
13   signal interno2: std_logic_vector(3 downto 0);
14
15 component divisor is
16   port( entrada: in std_logic;
17         salida: out std_logic);
18 end component divisor;
19
20 component cont4b is
21   port( in_clk, nRST:  in std_logic;
22         Q: out std_logic_vector(3 downto 0));
23 end component cont4b;
24
25 component decbin7scc is
26   port( E: in std_logic_vector(3 downto 0);
27         Sa, Sb, Sc, Sd, Se, Sf, Sg:  out std_logic);
28 end component decbin7scc;
29
30 begin
31   u1:  divisor PORT MAP (xtal_50M, interno1);
32   u2:  cont4b PORT MAP (interno1, nRST, interno2);
33   u3:  decbin7scc PORT MAP (interno2, Sa, Sb, Sc, Sd, Se, Sf, Sg);
34 end funcionamiento;

```

9

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

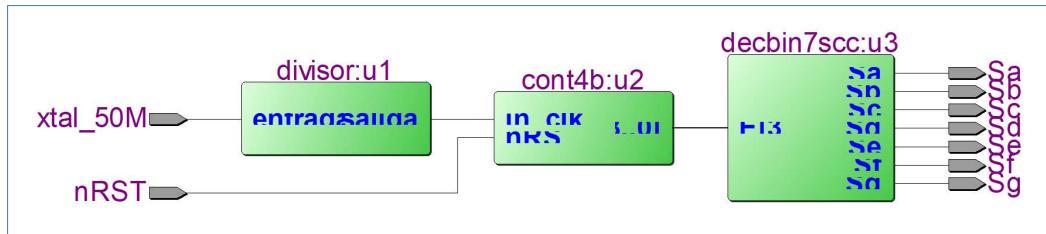
- En un primer proceso de compilado satisfactorio el proyecto organizará los archivos de acuerdo a la jerarquía descrita en EJEMPLO:



10

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

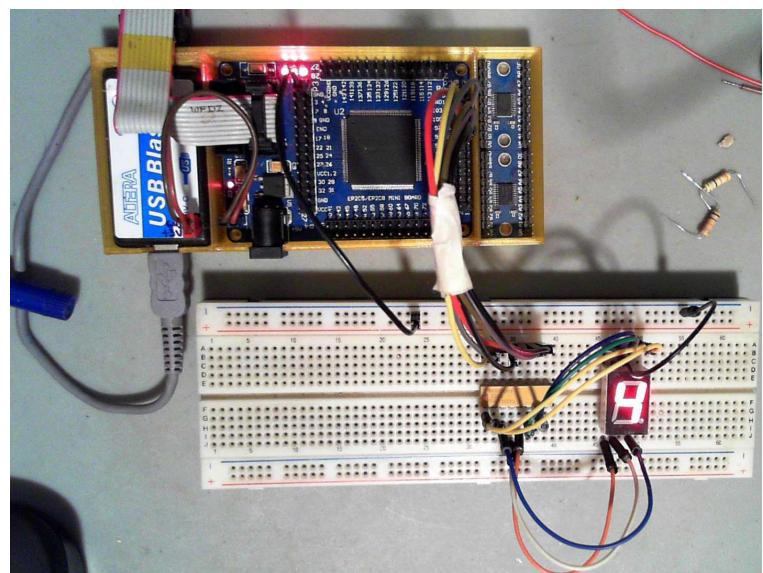
- Se puede apreciar que lo mostrado en el RTL Viewer tiene semejanza con lo planteado inicialmente en el ejemplo:



11

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

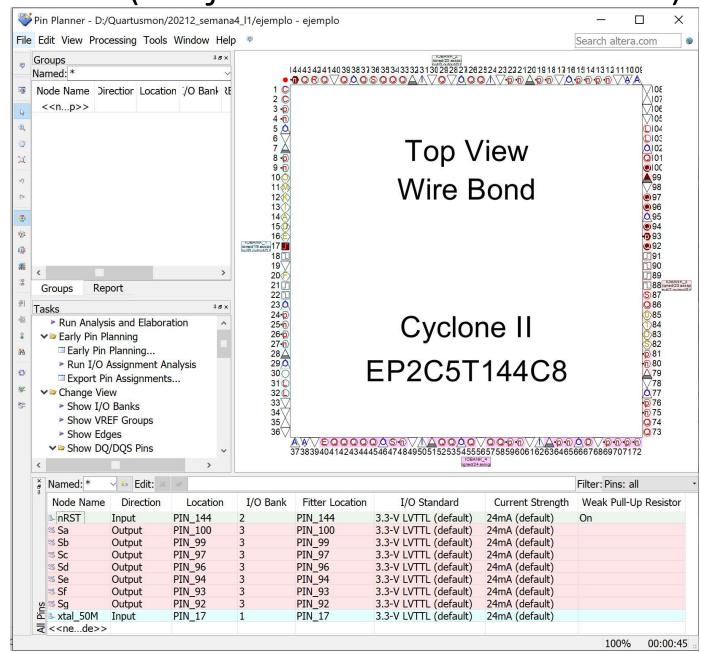
- Se procede al armado del circuito empleando un protoboard para instalar el display de siete segmentos y las resistencias de 330Ω , luego se emplearán cables jumper para conectar la tarjeta EP2C5.
- Se están empleando los pines 100, 99, 97, 96, 94, 93, 92 para los segmentos a, b, c, d, e, f, g respectivamente.
- Se emplearán el botón integrado en la tarjeta (pin 144) para la función de reset activo en bajo y el cristal de 50MHz (pin 17)



12

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

- Se abre el Pin Planner en el Quartus II para proceder a asignar los pines usados en la implementación con las señales del VHDL TOP-LEVEL.
- La asignación se hace en la columna “Location”
- Una vez asignado se retorna al Quartus II para hacer una segunda compilación para que se actualicen los cambios realizados en el Pin Planner



13

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

- Luego de la segunda compilación satisfactoria se procede a abrir el programador (Ventana de tareas/procesos, la última opción que es Open Programmer) y se procede a grabar el dispositivo
- Tener en cuenta que el archivo necesario para la grabación del FPGA tiene una extensión *.sof



14

Desarrollo en el Altera Quartus (Tarjeta EP2C5 Miniboard)

Notas finales:

- Verificar el funcionamiento del ejemplo en el circuito implementado
- Al presionar el botón ubicado en la tarjeta la cuenta visualizada en el display pasará a cero.
- Si se requiere que el cambio sea mas rápido solo se necesitará cambiar la cantidad de cuentas que hace el módulo divisor de frecuencia

15

Desarrollo en el Altera Quartus (Tarjeta A-C2FB)

- Crear el proyecto.
- Crear los archivos *.vhd para los circuitos divisor, cont4b y decbin7scc, tener en cuenta que el nombre del archivo debe de coincidir con el nombre de la entidad.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity div_freq is
7   port( inclk : in std_logic;
8         outclk : out std_logic);
9 end div_freq;
10
11 architecture flujo of div_freq is
12   signal internoso: std_logic_vector(25 downto 0);
13   signal salidasda: std_logic;
14   begin
15     process(inclk)
16     begin
17       if rising_edge(inclk) then
18         internoso <= internoso + 1;
19         if internoso = 2500000 then
20           internoso <= (others => '0');
21           salidasda <= not salidasda;
22         end if;
23       end process;
24     end if;
25   end process;
26   outclk <= salidasda;
27 end flujo;

```

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity decbin7sac is
7   port( e: in std_logic_vector(3 downto 0);
8         sa, sb, sc, sd, se, sf, sg: out std_logic);
9 end decbin7sac;
10
11 architecture flujo of decbin7sac is
12   signal salida: std_logic_vector(6 downto 0);
13   begin
14     with e select
15       salida <=
16         "011111" when "0000",
17         "000010" when "0001",
18         "1011011" when "0010",
19         "1001111" when "0011",
20         "1100110" when "0100",
21         "1101101" when "0101",
22         "1111011" when "0110",
23         "0000111" when "0111",
24         "1111111" when "1000",
25         "1100111" when "1001",
26         "1110111" when "1010",
27         "1111100" when "1011",
28         "1011000" when "1100",
29         "1011110" when "1101",
30         "1111011" when "1110",
31         "1110001" when "1111",
32         "ZZZZZZZ" when others;
33       sa <= not salida(0);
34       sb <= not salida(1);
35       sc <= not salida(2);
36       sd <= not salida(3);
37       se <= not salida(4);
38       sf <= not salida(5);
39       sg <= not salida(6);

```

16

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- Crear el archivo ejemplo1.vhd que llamará a DIV_FREQ, CONT4B y DECBIN7SAC como componentes y establecerá las conexiones entre ellos.

```

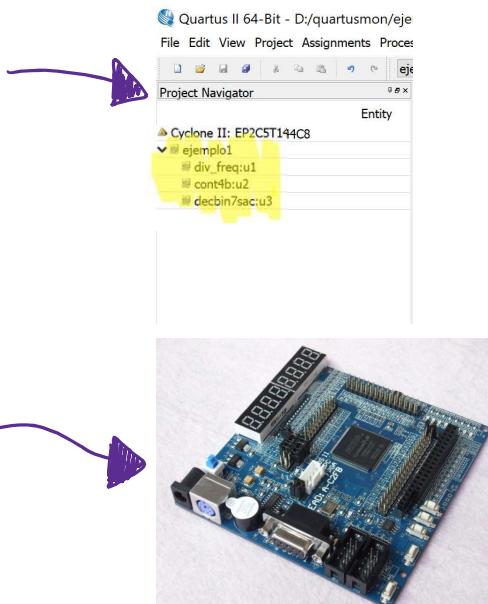
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity ejemplo1 is
7   port (inclk, rst: in std_logic;
8         buzz, sa, sb, sc, sd, se, sf, sg, sdp: out std_logic;
9         hab: out std_logic_vector(7 downto 0));
10 end ejemplo1;
11
12 architecture flujomon of ejemplo1 is
13 |
14 component div_freq
15   port( inclk : in std_logic;
16         outclk : out std_logic);
17 end component;
18
19 component cont4b
20   port( inclk, rst: in std_logic;
21         q: out std_logic_vector(3 downto 0));
22 end component;
23
24 component decbin7sac
25   port( e: in std_logic_vector(3 downto 0);
26         sa, sb, sc, sd, se, sf, sg: out std_logic);
27 end component;
28
29 signal int1: std_logic;
30 signal int2: std_logic_vector(3 downto 0);
31
32 begin
33   u1: div_freq PORT MAP(inclk, int1);
34   u2: cont4b PORT MAP (int1, rst, int2);
35   u3: decbin7sac PORT MAP (int2, sa, sb, sc, sd, se, sf, sg);
36   buzz <= '1';
37   sdp <= '1';
38   hab <= "11111110";
39 end flujomon;

```

17

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

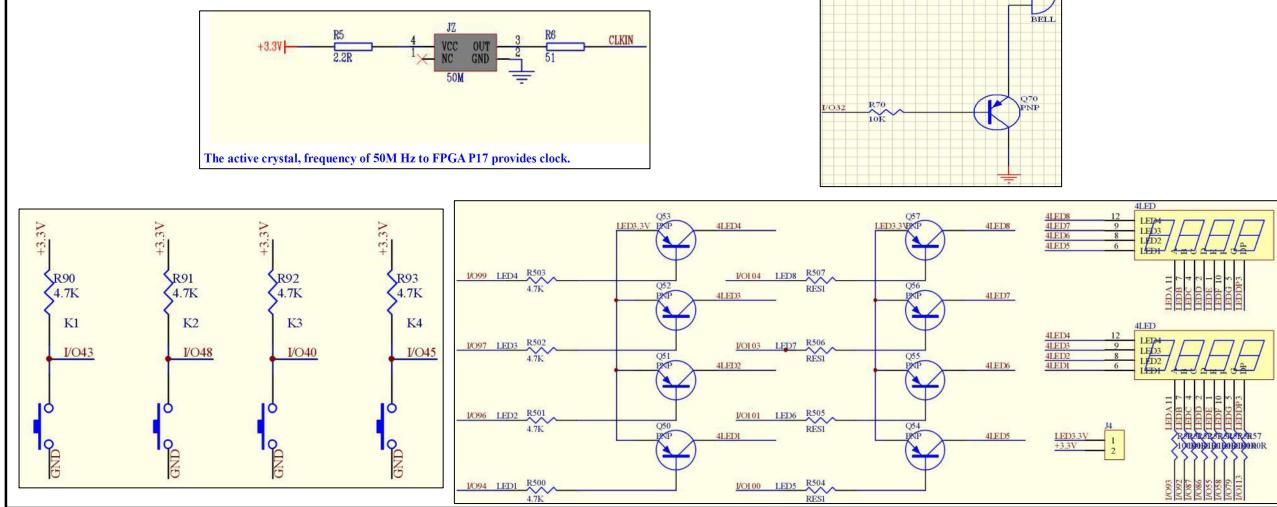
- Al realizar una primera compilación se actualizará la jerarquía de archivos en el proyecto
- Tener en cuenta que debido al DIV_FREQ no se podrá obtener una simulación satisfactoria debido al tiempo prolongado que tendría dicha simulación.
- Para verificar el diseño se empleará la tarjeta de desarrollo A-C2FB que posee un FPGA Altera Cyclone II EP2C5T144C8N y diversos periféricos.



18

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- Según la hoja técnica de la tarjeta A-C2FB, se tienen las siguientes configuraciones de periféricos:

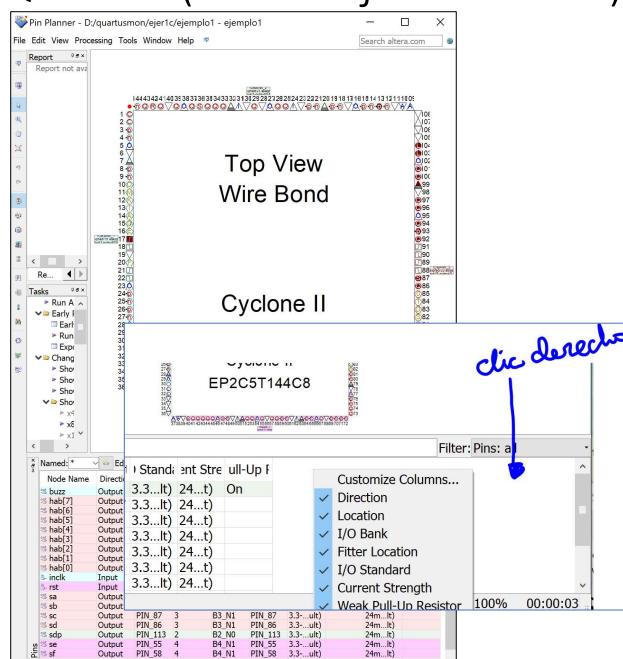


19

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- Teniendo en cuenta lo anterior asignaremos los pines del FPGA con el ejemplo1 empleando el PinPlanner:

- Para que aparezcan los pines en el PinPlanner se debió hacer una primera compilación de ejemplo1



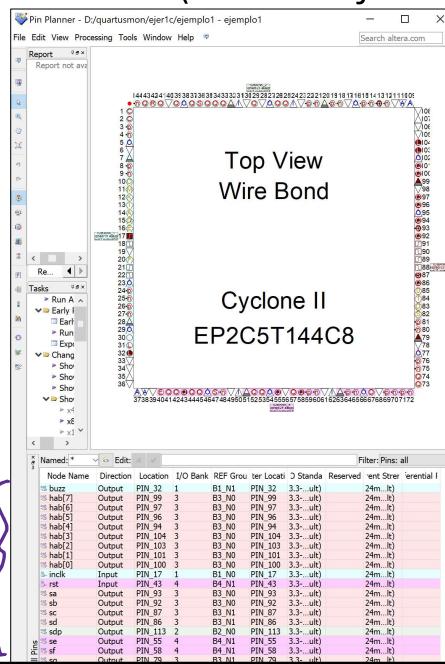
20

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- Teniendo en cuenta lo anterior asignaremos los pines del FPGA con el ejemplo1 empleando el PinPlanner:

- Para que aparezcan los pines en el PinPlanner se debió hacer una primera compilación de ejemplo1

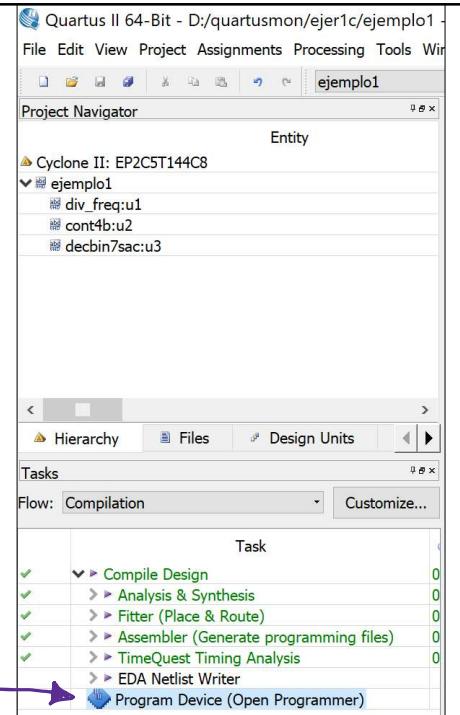
- Una vez asignados los pines del FPGA para con el ejemplo1 se procederá a realizar nuevamente una compilación



21

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- Luego de la segunda compilación (con los pines asignados) se procederá a realizar la implementación de ejemplo1 en la tarjeta A-C2FB.
- En la sección tareas abrir el programador:

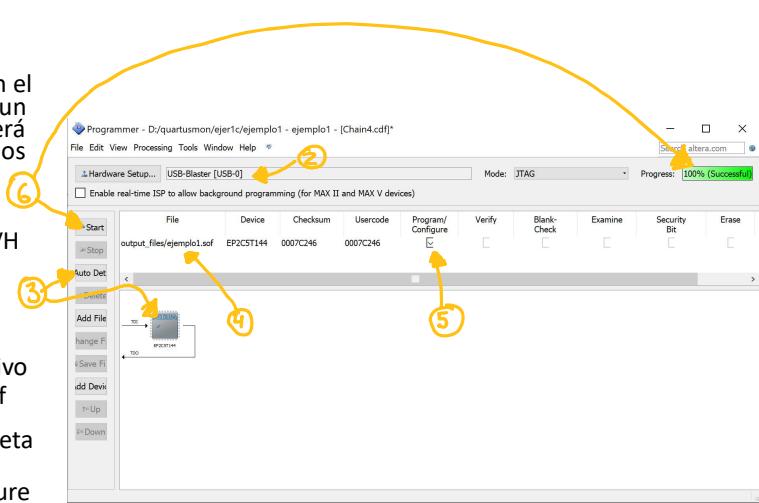


22

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- En el programador hacer lo siguiente:

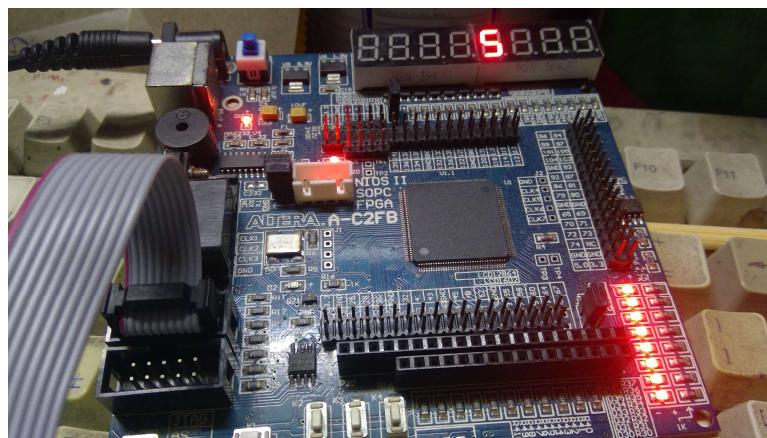
1. Conectar la tarjeta A-C2FB con el USB Blaster (puerto JTAG) y a un puerto US?B de la PC. Se deberá haber previamente instalado los drivers del USB Blaster de manera manual (revisar procedimiento en <https://github.com/tocache/VHDL-QuartusII/>)
2. Verificar si se encuentra habilitado el USB-Blaster.
3. Presionar autodetectar y verificar si aparece el dispositivo
4. Añadir el archivo ejemplo1.sof (ubicado en la carpeta output_files dentro de la carpeta del proyecto)
5. Dar check en Program/Configure
6. Presionar botón Start y esperar a que la barra de progreso llegue a 100%



23

Desarrollo en el Altera Quartus (Para tarjeta A-C2FB)

- Verificar el funcionamiento de ejemplo1 en la tarjeta A-C2FB



24

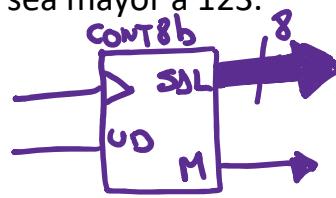
Troubleshooting:

- Conecto el USB Blaster a la PC y no aparece en el programador:
 - No se ha instalado los drivers del USB Blaster, proceder manualmente a instalarlo según lo explicado.
- Compilación y grabación satisfactoria pero no funciona el circuito:
 - Verifica el circuito implementado.
 - Verifica continuidad en los cables jumper.
 - Verifica en el protoboard que los buses de alimentación no estén partidos en la mitad.
 - Verifica si el tipo de display es el correcto (debe de ser cátodo común para que funcione con el ejemplo de esta sesión).
 - Verifica si el pin común (COM) del display esta conectado a la línea de GND.
 - Verifica si has conectado el GND de la tarjeta con el bus de GND del protoboard.
 - Verifica si los pines empleados en la implementación coincidan con lo estipulado en el Pin Planner y que se haya hecho un proceso de compilación luego de haber modificado en el Pin Planner antes de grabar el FPGA.
 - Revisa si la tarjeta tenga la alimentación de 5V conectada (LED 3)

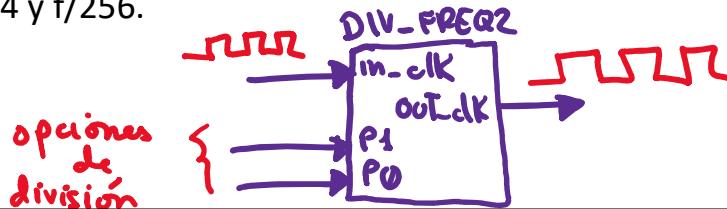
25

Cuestionario:

- Desarrollar un contador binario de 8 bits con opción de cuenta ascendente/descente a través de un pin UD (U=0/D=1). Tener en cuenta una señal de salida adicional de nombre 'M' para que sea '1' cuando la cuenta sea mayor a 123.



- Desarrollar un divisor de frecuencia con opciones de división f/2, f/8, f/64 y f/256.



26

Fin de la sesión