

# EL253 - Sistemas Digitales

Semestre 2020-2

Profesor Kalun José Lau Gan

Sesión de Laboratorio Semana 4

1

## Preguntas Previas

- ¿El weekend lab de implementación?
  - Se dará a conocer hoy por la tarde la hora y el enlace

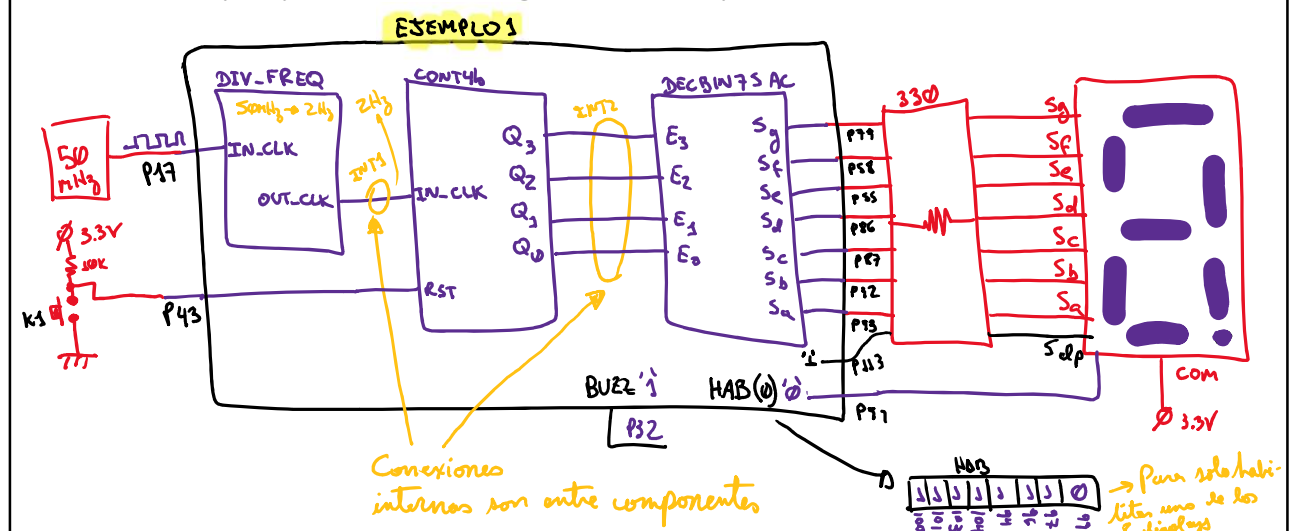
2

## 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 0.5 segundos, con RESET y visualización en display de siete segmentos tipo ánodo común



4

## Desarrollo en el Altera Quartus

- Crear el proyecto.
- Crear los archivos \*.vhd para los circuitos DIV\_FREQ, CONT4B y DECBIN7SAC, 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 internaso: std_logic_vector(25 downto 0);
13   signal salidas: std_logic;
14 begin
15   process(inclk)
16   begin
17     if rising_edge(inclk) then
18       internaso <= internaso + 1;
19       if internaso = 25000000 then
20         internaso <= (others => '0');
21         salidas <= not salidas;
22       end if;
23     end if;
24   end process;
25   outclk <= salidas;
26 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 cont4b is
7   port( inclk, rst: in std_logic;
8         q: out std_logic_vector(3 downto 0));
9 end cont4b;
10
11 architecture flujo of cont4b is
12   signal interno: std_logic_vector(3 downto 0);
13 begin
14   process(inclk, rst)
15   begin
16     if rst = '0' then
17       interno <= (others => '0');
18     else
19       if rising_edge(inclk) then
20         if interno = "1111" then
21           interno <= (others => '0');
22         else
23           interno <= interno + 1;
24         end if;
25       end if;
26     end if;
27   end process;
28   q <= interno;
29 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 <= "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               "1111100" when "1011",
27               "1011000" when "1100",
28               "1011110" when "1101",
29               "1111011" when "1110",
30               "1110001" when "1111",
31               "ZZZZZZZ" when others;
32   sa <= not salida(0);
33   sb <= not salida(1);
34   sc <= not salida(2);
35   sd <= not salida(3);
36   se <= not salida(4);
37   sf <= not salida(5);
38   sg <= not salida(6);
39 end flujo;

```

5

## Desarrollo en el Altera Quartus

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

```

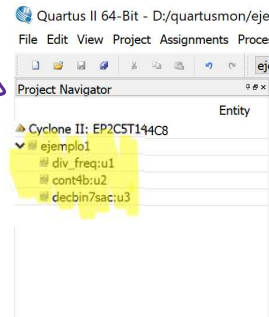
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity ejemplo1 is
7   port( inclk, rst: in std_logic;
8         buzz, sa, sb, sc, sd, se, sf, sg, sdp: out std_logic;
9         hab: out std_logic_vector(7 downto 0));
10 end ejemplo1;
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;

```

6

## Desarrollo en el Altera Quartus

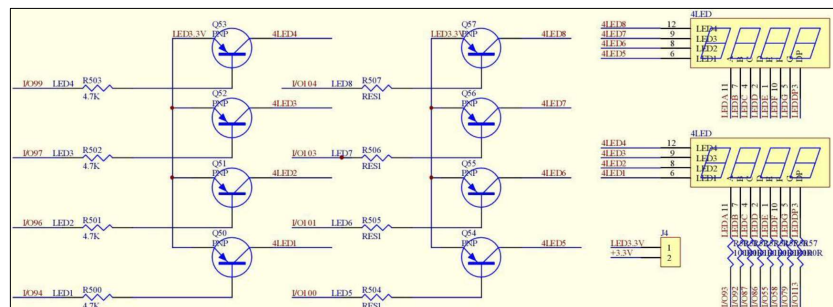
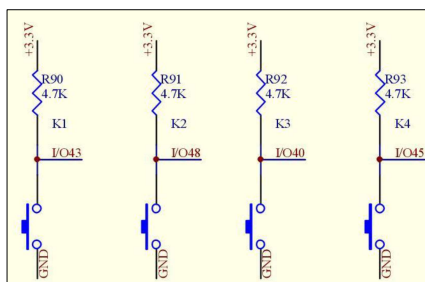
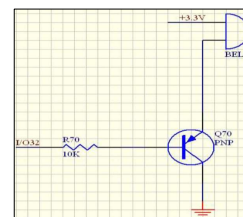
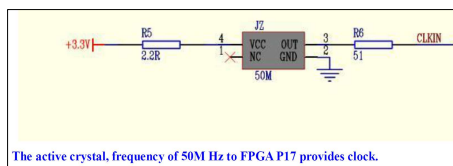
- Al realizar una primera compilación se actualizará la jerarquía de archivos en el proyecto
- Tener en cuenta que debido al DIV\_FREQ no se podrá obtener una simulación satisfactoria debido al tiempo prolongado que tendría dicha simulación.
- Para verificar el diseño se empleará la tarjeta de desarrollo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



7

## Desarrollo en el Altera Quartus

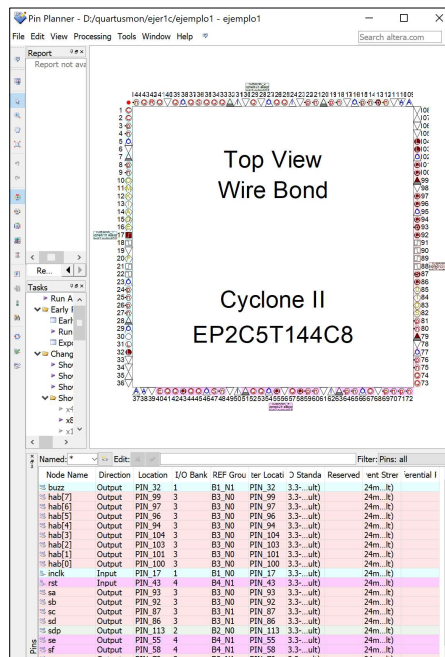
- Según la hoja técnica de la tarjeta A-C2FB, se tienen las siguientes configuraciones de periféricos:



8

## Desarrollo en el Altera Quartus

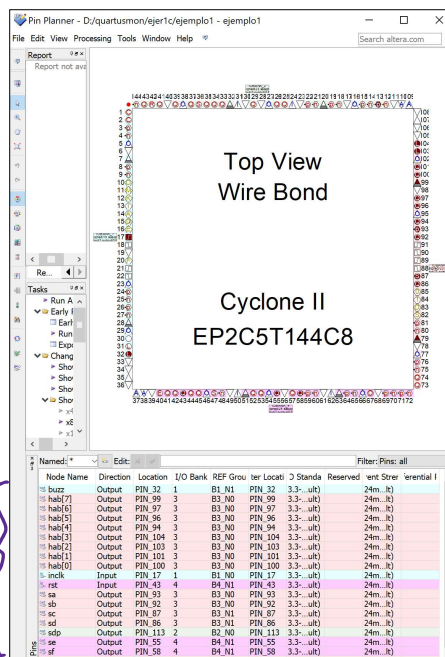
- Teniendo en cuenta lo anterior asignaremos los pines del FPGA con el ejemplo1 empleando el PinPlanner:
- Para que aparezcan los pines en el PinPlanner se debió hacer una primera compilación de ejemplo1



9

## Desarrollo en el Altera Quartus

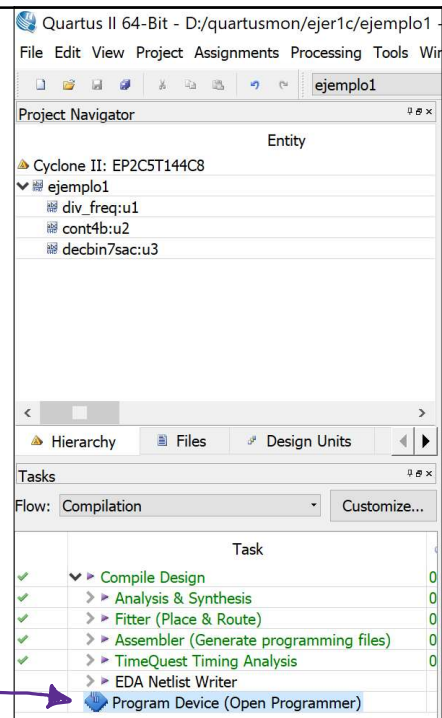
- Teniendo en cuenta lo anterior asignaremos los pines del FPGA con el ejemplo1 empleando el PinPlanner:
- Para que aparezcan los pines en el PinPlanner se debió hacer una primera compilación de ejemplo1
- Una vez asignados los pines del FPGA para con el ejemplo1 se procederá a realizar nuevamente una compilación



10

## Desarrollo en el Altera Quartus

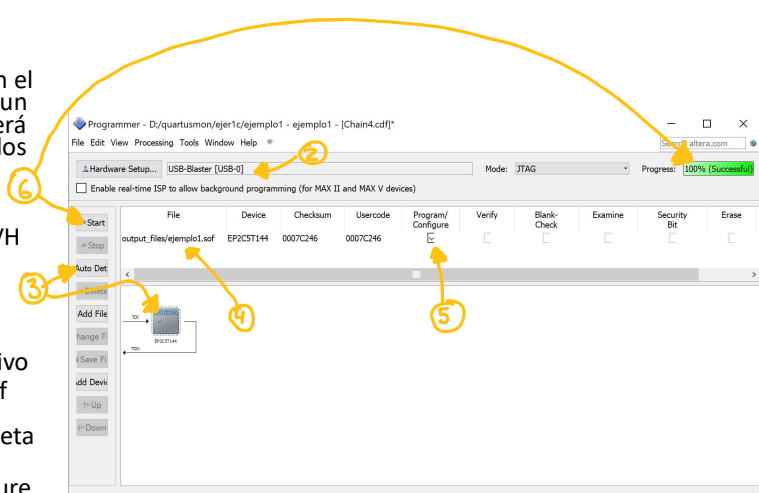
- Luego de la segunda compilación (con los pines asignados) se procederá a realizar la implementación de ejemplo1 en la tarjeta A-C2FB.
- En la sección tareas abrir el programador:



11

## Desarrollo en el Altera Quartus

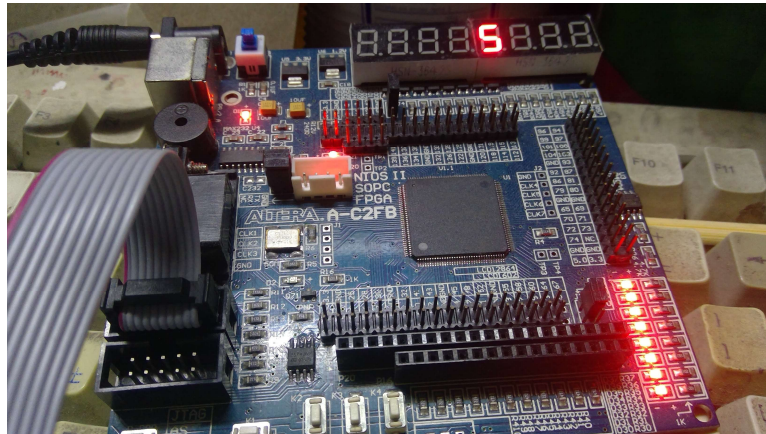
- En el programador hacer lo siguiente:
  1. Conectar la tarjeta A-C2FB con el USB Blaster (puerto JTAG) y a un puerto USB de la PC. Se deberá haber previamente instalado los drivers del USB Blaster de manera manual (revisar procedimiento en <https://github.com/tocache/VH-DL-QuartusII/>)
  2. Verificar 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%



12

## Desarrollo en el Altera Quartus

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



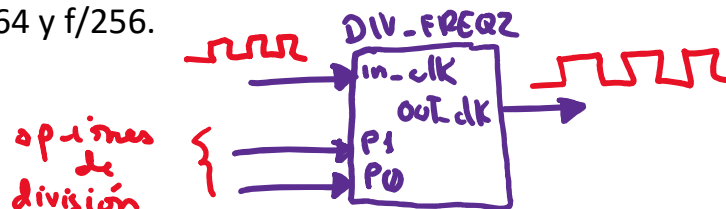
13

## 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  $f/256$ .



14

Fin de la sesión