ASIGNATURA	Modelado y Síntesis de Sistemas Electrónicos Digitales	FECHA	Abril 2018
APELLIDOS, NOMBRE			

Cuestión 1

Indicar las líneas de código VHDL que:

a) Definan un tipo de datos de nombre *mi_array* correspondiente un array bidimensional de 32x8 bits (std_logic).

(1 ptos)

b) Definir una señal de nombre *array_signal* del tipo de datos *mi_array.*

(1 ptos)

c) Definir dos señales de nombre *data* y *direction* con el tamaño necesario para que el valor de *data* se le asigne al elemento de *array_signal* seleccionado por *direction.*

(1 ptos)

d) Indicar, con una única sentencia, el código VHDL que permita asignar a todos los elementos de la señal *array_signal* el valor 81_{HEX}.

(1 ptos)

e) Indicar el código VHDL que permita realizar la asignación del aparado c.

(5 ptos)

f) Indicar la declaración de librerías y paquetes necesarios para poder realizar los apartados anteriores.

(1 ptos)

Cuestión 2
Dado los siguientes códigos VHDL

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity cuestion_2 is
  port (
    A   : in  std_logic_vector(2 downto 0);
    S  : out std_logic_vector(7 downto 0));
end cuestion_2;
architecture rtl of cuestion_2 is
begin
 P: process (A)
  begin
    case A is
      when "000"  => s(0) <= '1';
      when "001"  => s(1) <= '1';
      when "010"  => s(2) <= '1';
      when "011"  => s(3) <= '1';
      when "100"  => s(4) <= '1';
      when "101"  => s(5) <= '1';
      when "110"  => s(6) <= '1';
      when "111"  => s(7) <= '1';
      when others => null;
    end case;
  end process;
end rtl;
```

(a)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity cuestion_2_tb is
end cuestion_2_tb;
architecture sim of cuestion_2_tb is
  signal A_i : std_logic_vector(2 downto 0);
  signal S_i : std_logic_vector(7 downto 0);

begin  -- sim

  DUT:entity work.cuestion_2
    port map (
        A => A_i,
        S => S_i);

  process
  begin
    for j in 0 to 7 loop
      A_i<=std_logic_vector(to_unsigned(j,3));
      wait for 50 ns;
    end loop;  -- j
  end process;

end sim;
```

(b)

Dibujar el resultado de la simulación del testbench de la figura b hasta el instante t=500 ns. El código de la figura a modela el funcionamiento de la entidad cuestión_2.

(10 ptos)

Indicar, justificadamente, los problemas que plantea el código VHDL correspondiente a la entidad cuestión_2 si se desea que modele un decodificador de 3:8.

(5 ptos)

Cuestión 3 **(25 ptos)**

Codifique el modelo VHDL sintetizable de la entidad *bcd_counter* correspondiente a un contador BCD con salida *q* y que responda a la siguiente declaración: la señal *clk* es el reloj, es activo en el flanco de subida, *clr* es una señal de inicialización asíncrona, *ce* es la señal de habilitación de cuenta, *u_d* controla el sentido de la cuenta (con un nivel alto la cuenta es ascendente y con un nivel bajo es descendente), *load* es una señal de carga síncrona activa a nivel alto que pone el valor de *din* , simpre que sea corresponda con un dato BCD, en *q*, *tc* es la señal fin de cuenta y *ceo* es la señal de expansión (activa cuando estando habilitado el contador este alcanza el fin de cuenta).

```
entity bcd_counter is
  port (
    clr  : in  std_logic;
    clk  : in  std_logic;
    ce   : in  std_logic;
    u_d  : in  std_logic;
    load : in  std_logic;
    din  : in std_logic_vector(3 downto 0);
    q    : out std_logic_vector(3 downto 0);
    tc   : out std_logic;
    ceo  : out std_logic);
end entity;
```

Cuestión 4 **(30 ptos)**

Se debe crear el código VHDL que permita simular la entidad *bcd_counter* de la cuestión anterior. Los valores que se asignan a las entradas síncronas se realizarán coincidiendo con el flanco de bajada de *clk*.

Cuestión 5 **(20 ptos)**

Se debe crear el código VHDL sintetizable que modele la entidad *prescaler* que genera una salida *pre_out* que tiene un periodo de 1 s, estando a nivel alto un periodo de la señal *clk*. Siendo la frecuencia de la señal *clk* es de 100 MHz. *rst* es una entrada asíncrona de inicialización activa a nivel alto. Siempre que está activada *rst*, *pre_out* permanecerá a nivel bajo.

```
entity prescaler is
  port (
    clk     : in  std_logic;
    rst     : in  std_logic;
    pre_out : out std_logic);
end prescaler;
```