

### INSTRUCCIONES:

1. Resuelva este ejercicio en las mismas condiciones en que realizará el examen: dos horas de tiempo y sin emplear ningún material.
2. Revise sus contestaciones, empleando para ello el texto y el simulador que esté usando para estudiar la asignatura.
3. Compare sus respuestas revisadas con la solución.

### Pregunta 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales c, x, z1, z2 y z3 entre los instantes 0 y 60 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity crono is
end entity crono;

architecture crono of crono is
    constant PER : time :=10 ns;
    signal c: std_logic:='0';
    signal x: std_logic;
    signal z1: std_logic;
    signal z2: std_logic;
    signal z3: std_logic;
begin
    process(c)
    begin
        if (rising_edge(c)) then
            z1<=x;
            z2<=z1;
            z3<=z2;
        end if;
    end process;
    c<=not c after (PER/2);
    x<= '0', '1' after 5 ns, '0' after 20 ns;
end architecture crono;
```

## Pregunta 2 (3 puntos)

Escriba en VHDL, de las tres formas que se detallan a continuación, la **architecture** que describe el comportamiento de un circuito combinacional conversor de código binario de 3 bits a código Gray. La **entity** del circuito y su tabla de verdad son las siguientes:

```
entity codGray is
  port ( y      : out std_logic_vector(2 downto 0);
        x      : in  std_logic_vector(2 downto 0));
end entity codGray;
```

x	y
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

**2.a)** (1 punto) Empleando un bloque **process** con una sentencia **case**.

**2.b)** (1 punto) Empleando una asignación concurrente de selección (**with - select**).

**2.c)** (1 punto) Realizando una descripción de su estructura empleando para ello dos puertas lógicas XOR de dos entradas. Tenga en cuenta que el bit en la posición  $i$  de una palabra del código Gray es 1 si los bits en las posiciones  $i$  e  $i + 1$  de la correspondiente palabra binaria son diferentes.

La **entity** de la puerta lógica XOR se muestra a continuación.

```
entity xor2 is
  port ( y0      : out std_logic;
        x0, x1  : in  std_logic );
end entity xor2;
```

### Pregunta 3 (3 puntos)

Diseñe un circuito secuencial síncrono capaz de detectar cuando le llega la secuencia “1011” por su entrada. La **entity** del circuito se muestra a continuación. El circuito tiene una señal de reloj (`clk`), un entrada serie de un bit (`X`), una señal de reset asíncrona activa en ‘1’ (`reset`), una señal que indica el estado en que se encuentra el circuito (`state`) y una señal de salida de un bit (`Y`). La señal `Y` se pone a ‘1’ si por la entrada `X` los últimos 4 bits que han llegado se corresponden con la secuencia “1011”. La máquina no vuelve al estado inicial tras haber reconocido la secuencia y detecta secuencias solapadas. La señal `reset` pone el circuito en su estado inicial. Todos los cambios tienen lugar en el flanco de subida de la señal de reloj. Escriba en VHDL la **architecture** que describe el comportamiento del circuito en términos de una máquina de Moore. Dibuje el diagrama de estados correspondiente al circuito que ha diseñado.

```
entity detector is
  port( Y      : out std_logic;
        state : out std_logic_vector(2 downto 0);
        X      : in  std_logic;
        reset  : in  std_logic;
        clk    : in  std_logic);
end entity detector;
```

### Pregunta 4 (2 puntos)

Programa en VHDL un banco de pruebas para el circuito secuencial que ha diseñado al contestar a la Pregunta 3. El programa de test debe primero resetear el circuito y a continuación cargar en el circuito, a través de la entrada (`X`) y por este orden, los siete bits siguientes: ‘1’, ‘0’, ‘1’, ‘1’, ‘0’, ‘1’, ‘1’. Si los valores de la señal de salida de la UUT no coinciden con lo esperado, el programa de test debe mostrar el correspondiente mensaje.