

# INGENIERÍA DE COMPUTADORES III

## INSTRUCCIONES

Por favor, entregue esta primera hoja de enunciado junto con el examen.

Dispone de 2 horas para realizar el examen.

MATERIAL PERMITIDO: Ninguno.

## Pregunta 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales  $s_1$ ,  $s_2$ ,  $s_3$  y  $s_4$  entre los instantes 0 y 50 ns. En el cronograma se debe indicar para cada una de estas señales el instante de tiempo en que la señal cambia de valor así como su nuevo valor.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono2 is
end entity crono2;
architecture crono2 of crono2 is
    signal x1, x2, x3: std_logic;
    signal s1, s2, s3, s4 : std_logic;

begin
    x1 <= '1', '0' after 5 ns,
        '1' after 15 ns, '0' after 30 ns,
        '1' after 35 ns, '0' after 50 ns;
    x2 <= '0', '1' after 15 ns,
        '0' after 35 ns;
    x3 <= x1 after 10 ns;
    Procl: process
        variable valor : std_logic;
    begin
        for i in 0 to 4 loop
            valor := x1 and x2;
            s1 <= valor;
            s2 <= s1;
            s3 <= x1 and x2;
            s4 <= s2 or s3;
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture crono2;
```

## Pregunta 2 (3 puntos)

Se pretende diseñar un circuito comparador que tiene una señal de salida de un bit llamada  $F$  y dos señales de entrada: la señal de un bit  $E$  y la señal de 4 bits  $x$ . La **entity** del circuito se muestra a continuación.

```
entity comparaXmayor6 is port
  ( F : out std_logic;
    E:  in std_logic;
    x : in  std_logic_vector(3 downto 0) );
end entity comparaXmayor6;
```

El circuito tiene el siguiente comportamiento.

Los cuatro bits de la señal de entrada  $x$  se interpretan como un número binario sin signo.

La señal de salida  $F$  tiene valor '1' si y sólo si la señal  $E$  tiene valor '1' y, además, el valor de  $x$  es mayor que el número decimal 6.

En cualquier otro caso la señal  $F$  tiene valor '0'. En particular, si la señal  $E$  tiene valor '1' y, además, el valor de  $x$  es menor o igual que el número decimal 6, el valor de la señal de salida  $F$  es '0'. Y si la señal de entrada  $E$  tiene valor '0', entonces la señal de salida  $F$  tiene valor '0'.

- 2.a) (1.5 puntos) Escriba la tabla de verdad de la salida ( $F$ ) en función de las entradas ( $x$  y  $E$ ). Escriba la función lógica ( $F$ ) en función de  $x$  y  $E$  obtenida a partir de dicha tabla de verdad. Escriba en VHDL la **architecture** del circuito comparador empleando únicamente sentencias de asignación concurrente y operadores lógicos.
- 2.b) (0.5 puntos) Dibuje un diagrama circuital del circuito comparador implementado con puertas lógicas. Escriba en VHDL la **entity** y **architecture** de las puertas lógicas que ha incluido en el diagrama circuital.
- 2.c) (1 puntos) Escriba en VHDL una **architecture** que describa la *estructura* del circuito que ha dibujado en el anterior apartado, instanciando y conectando las puertas lógicas que ha diseñado anteriormente.

## Pregunta 3 (2 puntos)

Programe el banco de pruebas del circuito combinacional que ha diseñado en la Pregunta 2.a. Explique detalladamente cómo el programa de test comprueba de manera sistemática el funcionamiento del circuito. El banco de pruebas debe comprobar que los valores obtenidos de la UUT coinciden con los esperados, mostrando el correspondiente mensaje en caso de que no coincidan. Al final del test, debe mostrarse un mensaje indicando el número total de errores.

#### **Pregunta 4** (3 puntos)

Programe en VHDL un circuito contador módulo 10 como una máquina de estado tipo Moore sensible al flanco de subida de la señal de reloj. En el diseño únicamente pueden emplearse los dos siguientes paquetes de la librería IEEE:

```
IEEE.std_logic_1164  
IEEE.numeric_std
```

La **entity** del circuito contador se muestra a continuación:

```
entity dec_counter is port  
  ( q : out std_logic_vector(3 downto 0);  
    pulse : out std_logic;  
    en: in std_logic;  
    clk, reset: in std_logic);  
end entity dec_counter;
```

Las señales de entrada del circuito son las siguientes:

- Una señal `reset` asíncrona activa a nivel alto. Cuando esta señal está activa, la cuenta se pone a cero.
- La señal `en` permite habilitar o deshabilitar la cuenta. Si el valor de la señal `en` es '1' la cuenta está habilitada. En caso contrario, se para la cuenta.
- La señal de reloj `clk`.

Las señales de salida del circuito son las siguientes:

- La señal `q` muestra el valor de la cuenta en número binario. En consecuencia, esta señal toma cíclicamente los valores "0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111", "1000", "1001".
- La señal `pulse` toma el valor '1' sólo cuando el valor de la cuenta es "1001".