

INGENIERÍA DE COMPUTADORES III

INSTRUCCIONES

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 x1, x2, x3, x4 y x5 entre los instantes 0 y 100 ns. Indique el valor inicial de cada una de las señales, así como el instante en que cambian de valor y el nuevo valor.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity cronol is
end entity cronol;
architecture cronol of cronol is
    signal s1, x1, x2, x3, x4 : std_logic;
    signal x5 : std_logic := '0';
begin
    process (s1, x5)
        variable temp : std_logic;
    begin
        temp := x5;
        x1 <= s1;
        x2 <= x1;
        x3 <= temp;
    end process;
    s1 <= '0', '1' after 10 ns, '0' after 30 ns,
        '1' after 40 ns, '0' after 60 ns;
    x4 <= transport s1 after 15 ns;
    x5 <= s1 after 15 ns;
end architecture cronol;
```

Pregunta 2 (3 puntos)

Diseñe un circuito secuencial síncrono que permita controlar el funcionamiento de una máquina expendedora sencilla. El usuario de la máquina puede introducir únicamente monedas de 5 céntimos y 10 céntimos. Cuando se introducen exactamente 15 céntimos, la máquina expendedora saca un chicle. Este sistema de control debe emitir una señal de apertura (señal `abrir`) cuando ha detectado que se han introducido 15 céntimos (una moneda de 5 céntimos y una moneda de 10 céntimos, o tres monedas de 5 céntimos). No hace falta que considere el caso en el cual el importe introducido sea superior a 15 céntimos. El sistema de control recibe dos señales de entrada: `cinco` y `diez`. La señal `cinco` es una señal de entrada del circuito que tiene valor '1' durante un tiempo sólo si el usuario introduce una moneda de 5 céntimos. La señal `diez` es una señal de entrada del circuito que tiene valor '1' durante un tiempo sólo si el usuario introduce una moneda de 10 céntimos.

La **entity** del circuito se muestra a continuación.

```
entity maquina is
    port( abrir : out std_logic;
          cinco : in std_logic;
          diez  : in std_logic;
          reset  : in std_logic;
          clk    : in std_logic);
end entity maquina;
```

El circuito tiene una señal de reloj (`clk`), dos entradas de un bit (`cinco` y `diez`), una señal de reset asíncrona activa en '1' (`reset`) y una señal de salida de un bit (`abrir`).

La señal `reset` es la única señal asíncrona del circuito y pone el circuito en su estado inicial. Este estado inicial indica que no ha recibido ninguna moneda.

La señal `abrir` se pone a '1' sólo si la máquina ha recibido exactamente 15 céntimos. Esta señal ha de permanecer con el valor '1' sólo durante un periodo de la señal de reloj `clk`.

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.

Pregunta 3 (2 puntos)

Escriba en VHDL la **architecture** que describe el comportamiento de un circuito combinacional que tiene como señal de salida el número de ceros que contiene la señal de entrada de izquierda a derecha hasta que se encuentra el primer uno. La **entity** del circuito se muestra a continuación.

```
entity numCeros is
    port( Y      : out std_logic_vector(3 downto 0);
          X      : in  std_logic_vector(7 downto 0) );
end entity numCeros;
```

El circuito tiene una señal de entrada de ocho bits (X) y una señal de salida de cuatro bits (Y).

La señal Y indica en binario sin signo el número de bits que tiene la entrada X con valor '0' hasta que se encuentra el primer bit con valor '1', empezando a revisar los bits de la señal X de izquierda a derecha. Por ejemplo, cuando la señal X tiene el valor "00100101", la señal Y tiene el valor "0010" que se corresponde con el valor decimal 2.

Pregunta 4 (3 puntos)

Programe en VHDL un banco de pruebas que compruebe, para todos los posibles valores de la señal de entrada, si el circuito que ha diseñado al contestar a la Pregunta 3 se comporta correctamente.

Explique detalladamente cómo el programa de test comprueba exhaustivamente el valor del UUT para todos los posibles valores de la entrada.

El banco de pruebas debe comprobar que los valores obtenidos del 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.