

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 `clk`, `x`, `y1` e `y2` entre los instantes 0 y 1000 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono2 is
end entity crono2;
architecture crono2 of crono2 is
    signal x, y1, y2 : std_logic;
    signal clk : std_logic:='0';
begin
    process (clk)
        variable a, b, c: std_logic;
    begin
        if ( rising_edge(clk) ) then
            a := x;
            b := a;
            c := b;
            y1 <= c;
            y2 <= y1;
        end if;
    end process;
    clk <= not clk after 100 ns;
    process is
    begin
        x<='0'; wait until falling_edge(clk);
        x<='1'; wait until falling_edge(clk);
        x<='1'; wait until falling_edge(clk);
        x<='0'; wait until falling_edge(clk);
        wait;
    end process;
end architecture crono2;
```

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). 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` 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 rebido 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 (3 puntos)

Diseñe usando VHDL un contador módulo 16 ascendente/ descendente con carga paralela y señal reset asíncrona activa a nivel alto. Realice el diseño describiendo el comportamiento del circuito. El circuito opera en el flanco de subida de la señal de reloj. Tiene la siguiente **entity**:

```
entity contador is
    port( Q:  out std_logic_vector( 3 downto 0);
          D:  in  std_logic_vector( 3 downto 0);
          Clk: in std_logic;
          Load: in std_logic;
          Reset: in std_logic;
          Count: in std_logic;
          Down: in std_logic );
end entity contador;
```

A continuación se describe el funcionamiento del circuito contador.

– *Reset asíncrono activo a nivel alto.*

Mientras Reset vale ‘1’, el contador tiene el valor “0000”.

– *Carga síncrona.*

Mientras Reset vale ‘0’ y la señal Load tiene valor ‘1’, se carga el valor de la señal de entrada D en el registro en el flanco de subida de la señal de reloj.

– *Cuenta síncrona ascendente.*

Mientras Reset vale ‘0’, la señal Load tiene valor ‘0’, la señal Count que habilita la cuenta vale ‘1’ y la señal Down vale ‘0’, se realiza la cuenta síncrona ascendente. Es decir, el contador pasa cíclicamente en cada flanco de subida de la señal de reloj por la secuencia “0000”, “0001”, “0010”, “0011”, “0100”, “0101”, “0110”, “0111”, “1000”, “1001”, “1010”, “1011”, “1100”, “1101”, “1110” y “1111”.

– *Cuenta síncrona descendente.*

Mientras Reset vale ‘0’, la señal Load tiene valor ‘0’, la señal Count que habilita la cuenta vale ‘1’ y la señal Down vale ‘1’, se realiza la cuenta síncrona descendente. Es decir, el contador pasa cíclicamente en cada flanco de subida de la señal de reloj por la secuencia “1111”, “1110”, “1101”, “1100”, “1011”, “1010”, “1001”, “1000”, “0111”, “0110”, “0101”, “0100”, “0011”, “0010”, “0001” y “0000”.

– *Cuenta deshabilitada.*

Mientras Reset vale ‘0’, la señal Load tiene valor ‘0’ y la señal Count que habilita la cuenta vale ‘0’, el contador mantiene el valor de la señal de salida Count.

Pregunta 4 (2 puntos)

Programe en VHDL un banco de pruebas para el circuito que ha diseñado al contestar a la Pregunta 3. La señal de reloj (`Clk`) debe tener un periodo de 20 ns e inicialmente valer '0'. El primer flanco de subida de la señal de reloj se ha de producir en el instante 10 ns. El programa de test debe realizar consecutivamente las acciones siguientes:

1. *Reset.* La señal de reset ha de tener el valor '1' durante los primeros 15 ns.
2. *Realizar durante tres periodos de la señal de reloj la cuenta ascendente.*
3. *Cargar en la señal de salida `Count` el valor "1111".*
4. *Realizar durante cuatro periodos de la señal de reloj la cuenta descendente.*
5. *Deshabilitar la cuenta.*