

## 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 x1, x2, x3 y x4 entre los instantes 0 y 100 ns.

```
use IEEE.std_logic_1164.all;
entity cronol is
end entity cronol;
architecture cronol of cronol is
    signal x1, x2, x3, x4 : std_logic;
begin
    proc0: process
    begin
        x1 <= TRANSPORT '0' AFTER 10 NS;
        x1 <= TRANSPORT '1' AFTER 20 NS;
        x1 <= TRANSPORT '1' AFTER 15 NS;
        x1 <= TRANSPORT '0' AFTER 30 NS;
        wait;
    end process;
    Proc1: process
    begin
        x2 <= '1'; wait for 10 ns;
        x2 <= '0'; wait for 20 ns;
        x2 <= '1'; wait for 5 ns;
        x2 <= '0';
    end process;
    x3 <= x1 xor x2 after 10 ns;
    Proc2: process
    begin
        for i in 0 to 3 loop
            x4 <= x2; wait for 5 ns;
        end loop;
        wait;
    end process;
end architecture cronol;
```

## Pregunta 2 (2.5 puntos)

Se pretende diseñar un circuito comparador cuya salida ( $F$ ) vale '1' si el valor del número de cuatro bits de entrada ( $x$ ) es mayor que el número decimal 9. Los cuatro bits de entrada se interpretan como un número binario sin signo. La **entity** del circuito se muestra a continuación.

```
entity comparaXmayor9 is port
  ( F : out std_logic;
    x : in  std_logic_vector(3 downto 0) );
end entity comparaXmayor9;
```

- 2.a)** (1 punto) Escriba la tabla de verdad de la salida ( $F$ ) en función de la entrada ( $x$ ). Escriba la función lógica ( $F$ ) en función de ( $x$ ) 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)** (1.5 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 3 (2.5 puntos)

Escriba el código VHDL de la **architecture** que describe el comportamiento de un circuito contador binario de 3 bits cuya salida sigue cíclicamente la secuencia “001”, “100”, “110” y “111”, con señal de reset asíncrona activa a nivel alto. La **entity** del circuito se muestra a continuación.

```
entity contador is
  port ( count      : out std_logic_vector (2 downto 0);
        clk, reset, c : in  std_logic );
end contador;
```

Las entradas al circuito son la señal de reloj (`clk`), la señal de reset asíncrono activo a nivel alto (`reset`) y la señal que habilita la cuenta (`c`). La salida del circuito contador es la señal de 3 bits `count`.

El funcionamiento del circuito debe ser el siguiente:

– *Reset asíncrono activo a nivel alto.*

Mientras `reset` vale '1', el contador tiene el valor “001”.

– *Cuenta síncrona.*

Mientras `reset` vale '0' y la señal `c` que habilita la cuenta vale '1', el contador pasa cíclicamente en cada flanco de subida de la señal de reloj por la secuencia “001”, “100”, “110” y “111”.

– *Cuenta deshabilitada.*

Mientras `reset` vale '0' y la señal `c` que habilita la cuenta vale '0', el contador mantiene el valor de la señal de salida `count`.

Describa el comportamiento del circuito en términos de una máquina de Moore. El estado del contador ha de tener el mismo valor que la señal de salida `count`. En caso de que el circuito esté en un estado distinto a los estados “001”, “100”, “110” y “111”, debe pasar en el flanco de subida de la señal de reloj al estado “001”. Dibuje el diagrama de estados en el que ha basado su diseño.

#### Pregunta 4 (3 puntos)

Diseñe un circuito secuencial síncrono capaz de detectar cuando le llega la secuencia de bits “1101” por su entrada serie de un bit. La **entity** del circuito se muestra a continuación.

```
entity detector is
  port( Y      : out std_logic;
        state : out std_logic_vector(1 downto 0);
        X      : in  std_logic;
        reset  : in  std_logic;
        clk    : in  std_logic);
end entity detector;
```

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 los últimos cuatro bits de su entrada *X* ha llegado la secuencia “1101”. El circuito ha de ser capaz de detectar 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 Mealy. Dibuje el diagrama de estados correspondiente al circuito que ha diseñado, indicando en cada arco de transición los valores de la señal de entrada *X* y de la señal de salida *Y*.