

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, x5 y x6 entre los instantes 0 y 90 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, x4, x5, x6 : std_logic;
begin
    x1 <= '1', '0' after 10 ns, '1' after 20 ns,
        '0' after 25 ns, '1' after 40 ns;
    Proc1: process
    begin
        x2 <= '1'; wait for 5 ns;
        x2 <= '0'; wait for 10 ns;
        x2 <= '1'; wait for 20 ns;
        x2 <= '0';
    end process;
    x3 <= x1 after 10 ns;
    x4 <= transport x1 after 10 ns;
    Proc2: process
        variable v : std_logic;
    begin
        for i in 0 to 3 loop
            v := x1 xor x2; x5 <= v;
            x6 <= x5; wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture crono2;
```

Pregunta 2 (3 puntos)

Programe en VHDL un sumador de dos números BCD de 8 bits con 1 bit de acarreo de entrada cuya salida es otro número BCD de 8 bits y 1 bit de acarreo de salida. El circuito tiene como entradas las señales `a` y `b`, que representan los dos números BCD de 8 bits, y una señal de acarreo de 1 bit llamada `cin`. Tiene como salidas la señal `sum` de 8 bits, que contiene el resultado de la suma, y la señal `cout` de 1 bit con el valor del acarreo de salida. Por ejemplo, si `a` es 00010001 entonces esta señal representa al número decimal 11. Si `b` es 00110001 entonces esta señal representa al número decimal 31. El resultado de la suma si la señal de acarreo de entrada es 0 será el número decimal 42, cuyo código BCD es 01000010. Por tanto, la señal `sum` tendrá el valor 01000010 y la señal `cout` tendrá el valor 0.

Se debe emplear el siguiente paquete:

```
package BCD_CONSTANTS is
    constant WIDTH      : integer := 8;
                        -- Núm. bits de los operandos
end package BCD_CONSTANTS;
```

En el diseño únicamente pueden emplearse los dos siguientes paquetes de la librería IEEE:

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

El circuito sumador ha de tener la siguiente **entity**:

```
entity BCDSum is
    port(
        sum   : out std_logic_vector (WIDTH-1 downto 0);
        cout  : out std_logic;
        a, b  : in  std_logic_vector (WIDTH-1 downto 0);
        cin   : in  std_logic );
end entity BCDSum;
```

Pregunta 3 (3 puntos)

Programe un banco de pruebas del circuito combinacional que ha diseñado en la Pregunta 2. El banco de pruebas va a comprobar únicamente que el circuito genera correctamente los valores de las señales de salida en los siguiente cuatro casos:

- a = "00010001", b = "00010001", cin = '0'
- a = "10011001", b = "00000001", cin = '0'
- a = "10011001", b = "00000001", cin = '1'
- a = "00110101", b = "10001000", cin = '1'

El banco de pruebas debe permitir comprobar que los valores de las señales de salida del circuito combinacional a probar coinciden con los esperados, mostrando un mensaje de error en caso contrario. Al final del test, debe mostrarse un mensaje indicando que el test ha finalizado y el número total de errores detectados.

Pregunta 4 (2 puntos)

Programe en VHDL un registro desplazamiento de 8 bits con entrada y salida serie. El registro realiza una operación de desplazamiento de 1 bit en cada flanco de subida de la señal de reloj (clk).

La **entity** del registro de desplazamiento se muestra a continuación:

```
entity registro8 is port
  ( Serial_out : out std_logic;
    Serial_in   : in  std_logic;
    clk        : in  std_logic );
end entity registro8;
```