

INGENIERÍA DE COMPUTADORES III

INSTRUCCIONES

Por favor, entregue todas las hojas 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, x4, x5, x6 entre los instantes 0 y 50 ns.

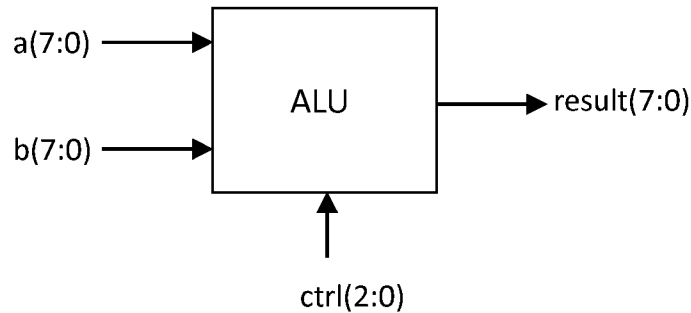
```
library IEEE;
use IEEE.std_logic_1164.all;

entity crono is
end entity crono;

architecture crono of crono is
    signal x1, x3, x4, x5, x6: std_logic;
    signal x2 : std_logic := '0';
begin
    x1 <= '1',
        '0' after 10 ns,
        '1' after 20 ns,
        '0' after 30 ns,
        '1' after 40 ns;
    x2 <= '0',
        '1' after 35 ns;
    Proc1: process (x1, x2, x3)
        variable valor : std_logic;
    begin
        x4 <= x1 or x2;
        valor := x1 or x2;
        x5 <= x4;
        x6 <= valor;
        x3 <= x6;
    end process;
end architecture crono;
```

Pregunta 2 (3 puntos)

En la siguiente figura se muestran las señales de entrada y salida de una Unidad Aritmético Lógica (ALU).



La **entity** de dicho circuito es la siguiente:

```
entity alu is
    port( result : out std_logic_vector (7 downto 0);
          ctrl   : in  std_logic_vector (2 downto 0);
          a , b  : in  std_logic_vector (7 downto 0) );
end entity alu;
```

Las señales *a* y *b*, de 8 bits, son los operandos. La señal *ctrl*, de 3 bits, determina la operación realizada por el circuito, según se indica en la tabla siguiente. Se ha empleado el símbolo – para indicar que el valor puede ser 0 ó 1. El circuito realiza las operaciones aritméticas suma y resta interpretando los operandos como números binarios con signo.

ctrl	Operación
"0--"	$a + 1$
"100"	$a + b$
"101"	$a - b$
"110"	$a \text{ and } b$
"111"	$a \text{ or } b$

Diseñe en VHDL la **architecture** que describe el comportamiento del circuito, usando únicamente asignaciones concurrentes simples y asignaciones concurrentes condicionales (**when - else**), y empleando únicamente los dos siguientes paquetes de la librería IEEE:

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

Puede emplear en el diseño tantas señales auxiliares como considere necesario.

Pregunta 3 (2.5 puntos)

A continuación se muestra la **entity** de un circuito contador binario ascendente y descendente de 3 bits.

```
entity contador is
  port ( cuenta : out std_logic_vector(2 downto 0);
        clk      : in  std_logic;
        reset    : in  std_logic;
        up       : in  std_logic;
        down     : in  std_logic );
end contador;
```

El circuito tiene una señal de 3 bits (*cuenta*) de salida y cuatro señales de entrada: la señal de reloj (*clk*), la señal de reset asíncrono activo a nivel alto (*reset*), la señal de cuenta ascendente (*up*) y la señal de cuenta descendente (*down*).

El funcionamiento del circuito debe ser el siguiente:

- *Reset asíncrono activo a nivel alto.* Cuando la señal *reset* pasa a valer 1, el contador se pone al valor “000”.
- *Cuenta ascendente.* La señal *up* tiene prioridad sobre la señal *down*. Mientras *reset* vale '0' y *up* vale '1', sea cual sea el valor de la señal *down*, en cada flanco de subida de la señal de reloj el contador se incrementa en uno, hasta llegar al valor “111”. La cuenta ascendente se detiene en “111”, dado que el contador no tiene rebosamiento.
- *Cuenta descendente.* Mientras *reset* vale '0', *down* vale '1' y *up* vale '0', en cada flanco de subida de la señal de reloj el contador se decrementa en uno hasta llegar al valor “000”. La cuenta descendente se detiene en “000”.

Diseña en VHDL la **architecture** que describe el comportamiento del circuito, empleando para ello un único bloque **process** y utilizando únicamente los dos siguientes paquetes de la librería IEEE:

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

Pregunta 4 (2.5 puntos)

Programe en VHDL un banco de pruebas para el contador binario que ha diseñado en la Pregunta 3.

La señal de reloj (`clk`) debe tener un periodo de 10 ns e inicialmente valer '0'.

El programa de test debe realizar las acciones siguientes:

1. Resetear el contador. La señal (`reset`) debe valer '1' únicamente entre los instantes 2 ns y 7 ns.
2. Incrementar el valor del contador hasta "111".
3. Mantener el valor "111" durante un ciclo de reloj.
4. Decrementar la cuenta hasta alcanzar el valor "000".

Si el valor de la señal de salida de la UUT no coincide con lo esperado, el programa de test debe mostrar el correspondiente mensaje.

Además, dibuje el cronograma de evolución que han de seguir las señales de entrada y salida de la UUT.