

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 entre los instantes 0 y 50 ns.

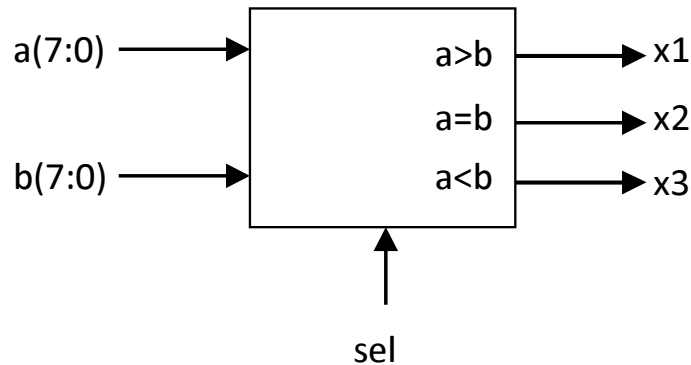
```
library IEEE;
use IEEE.std_logic_1164.all;

entity cronoW is
end entity cronoW;

architecture cronoW of cronoW is
    signal x1, x2, x3, x4, x5 : std_logic;
begin
    x1 <= '1',
        '0' after 10 ns,
        '1' after 20 ns,
        '0' after 30 ns,
        '1' after 40 ns;
    x2 <= '0',
        '1' after 25 ns;
    Procl: process
        variable valor : std_logic;
    begin
        wait for 10 ns;
        x3 <= x1;
        for i in 0 to 3 loop
            valor := x1 or x2;
            x4 <= valor;
            x5 <= x4;
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture cronoW;
```

Pregunta 2 (3 puntos)

Escriba en VHDL la **architecture** que describe el comportamiento del circuito combinacional mostrado en la siguiente figura.



Este circuito realiza la comparación de las señales de 8 bits a y b . Estas señales son interpretadas como números binarios con o sin signo dependiendo del valor de la señal de entrada sel . Si $sel = '1'$, se realiza la comparación entre a y b interpretándolos como números binarios con signo. Por el contrario, si $sel = '0'$ la comparación entre a y b se hace interpretándolos como números binarios sin signo. El circuito tiene tres salidas, $x1$, $x2$ y $x3$ que valen '1' sólo en el caso de que $a > b$, $a = b$ y $a < b$, respectivamente. A continuación se muestra la **entity** del circuito incluyendo la declaración de las únicas librerías que se han de usar en el diseño.

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

entity comparador is port
    ( x3, x2, x1 : out std_logic;
      a, b       : in  std_logic_vector(7 downto 0);
      sel       : in  std_logic );
end entity comparador;
```

Pregunta 3 (2.5 puntos)

Programe en VHDL un banco de pruebas del comparador que ha diseñado al resolver la Pregunta 2. El banco de pruebas debe comprobar el funcionamiento del circuito para los vectores de test siguientes:

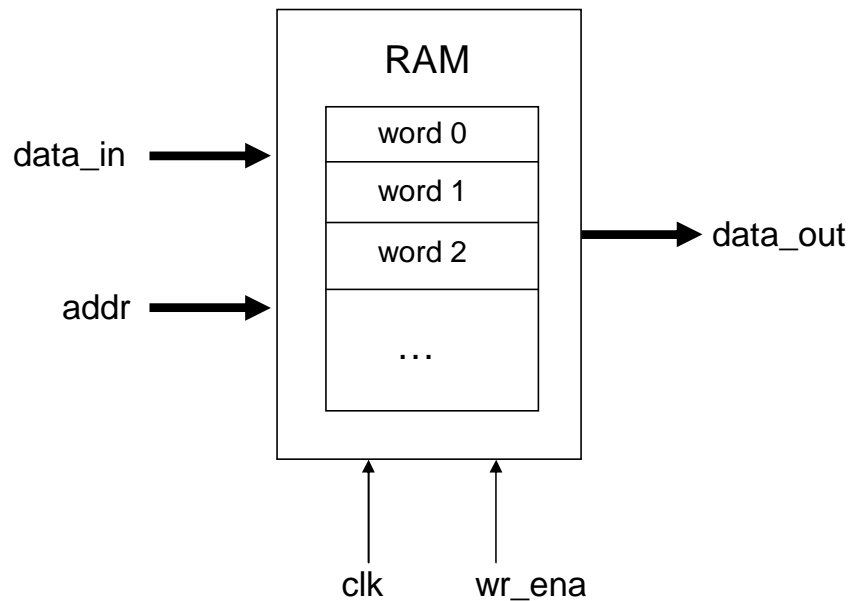
- $a = "11111111"$, $b = "00000000"$, $sel = '1'$
- $a = "11111111"$, $b = "00000000"$, $sel = '0'$
- $a = "00001110"$, $b = "00001111"$, $sel = '1'$

- `a="00001110", b="00001111", sel='0'`
- `a="01111111", b="01111110", sel='1'`
- `a="01111111", b="01111110", sel='0'`
- `a="00011110", b="00011110", sel='1'`
- `a="00011110", b="00011110", sel='0'`

El banco de pruebas debe comprobar para cada vector de test si la salida de la UUT es correcta, mostrar un mensaje cada vez que la salida de la UUT no sea correcta y mostrar un mensaje al finalizar el test en el que se indique el número total de salidas incorrectas.

Pregunta 4 (2.5 puntos)

Diseñe en VHDL una memoria de acceso aleatorio que contenga 16 palabras de 8 bits (RAM 16×8). Tal como se muestra en la siguiente figura, la RAM tiene un bus de entrada de datos (`data_in`), un bus de salida de datos (`data_out`), un bus de direcciones (`addr`), una entrada de reloj (`clk`) y una entrada para habilitar la escritura (`wr_ena`).



La señal `data_in` se almacena en la posición indicada por `addr` cuando la señal `wr_ena` está a '1' y la señal `clk` está en el flanco de subida.

Por otro lado, el bus de salida de datos debe mostrar continuamente los datos seleccionados por `addr`.

Realice el diseño en VHDL de modo que el número de bits de cada palabra (`bits`) y el número de palabras de la memoria (`words`) sean constantes de tipo **generic**.