INGENIERÍA DE COMPUTADORES III

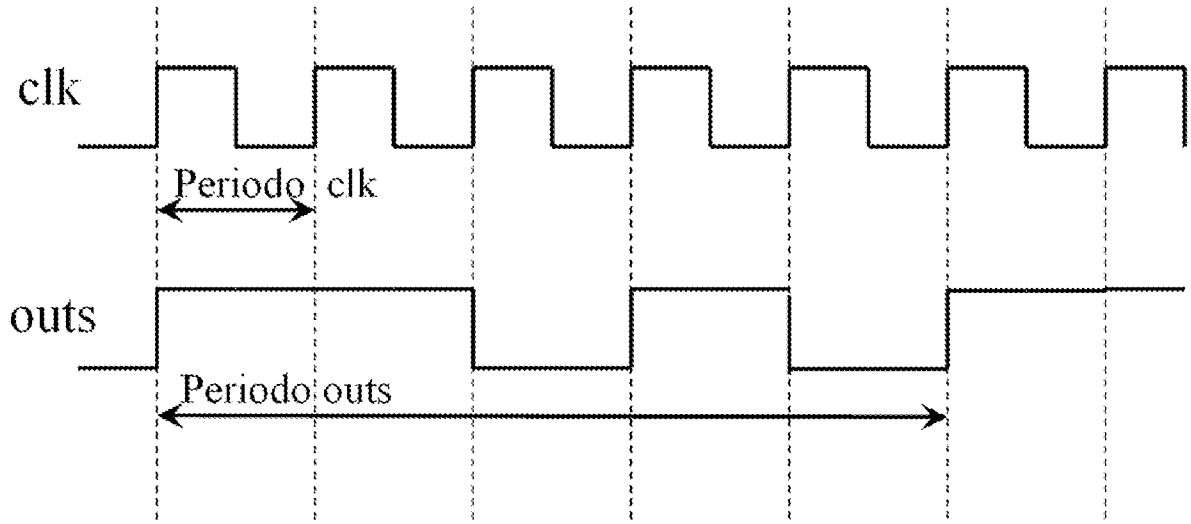
INSTRUCCIONES
Por favor, entregue esta primera hoja de enunciado junto con el examen.
Dispone de 2 horas para realizar el examen.
MATERIAL PERMITIDO: Ninguno.

Pregunta 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales `x1`, `x2`, `x3`, `x4` y `x5` entre los instantes 0 y 80 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 5 ns,
        '1' after 15 ns,'0' after 20 ns,
        '1' after 40 ns;
  Proc1: process
  begin
    x2 <= '1';
    wait for 5 ns;
    x2 <= '0';
    wait for 15 ns;
    x2 <= '1';
    wait for 10 ns;
    x2 <= '0';
  end process;
   x3 <= x1 after 10 ns;
  Proc2: process
     variable valor  : std_logic;
  begin
    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 (2.5 puntos) Diseñe un generador de señales que obtenga la forma de onda mostrada en la parte inferior de la figura siguiente (señal `outs`) a partir de una señal de reloj `clk`. Describa su comportamiento como una máquina de estado finito de Moore.



Pregunta 3 (3.5 puntos)

Diseñe usando VHDL un desplazador de nBits bits que opere en el flanco de subida de la señal de reloj, con carga de datos síncrona, reset síncrono activo a nivel alto y con una señal para seleccionar desplazamiento a la izquierda o a la derecha. El desplazador tiene la siguiente **entity**.

```
entity desplazador is
   generic (nBits: integer:=8);
   port (  dout : out std_logic_vector( nBits-1 downto 0);
           data: in std_logic_vector( nBits-1 downto 0);
           clk, load, izqda, reset  : in  std_logic );
end desplazador;
```

Las entradas al circuito son la señal de reloj (`clk`), la señal de carga (`load`), la señal de reset (`reset`), la señal que selecciona el tipo de desplazamiento (`izqda`) y la señal `data`. El circuito tiene una única señal de salida llamada `dout` y una constante *generic* que indica el número de bits de las señales de entrada y salida de datos.

La señal de carga tiene prioridad sobre la señal de reset, y la señal de reset tiene prioridad sobre la operación de desplazamiento. Cuando las señales `load` y `reset` valen '0', se realiza en el flanco de subida de la señal de reloj el desplazamiento de un bit a la izquierda o a la derecha dependiendo del valor de la señal `izqda`. Cuando la señal `izqda` vale '0', se realiza el desplazamiento de un bit a la derecha introduciendo un '0' a la izquierda y cuando vale '1' se realiza el desplazamiento de un bit a la izquierda introduciendo un '0' a la derecha.

Pregunta 4 (2 puntos)

Programe en VHDL un banco de pruebas para el circuito que ha diseñado al contestar a la Pregunta 3 para un valor de la constante `nBits` igual a 8. La señal de reloj (`clk`) debe tener un periodo de 20 ns e inicialmente valer '0'. El programa de test debe realizar consecutivamente las acciones siguientes:

1. *Reset*. La señal `reset` ha de tener el valor '1' durante 15 ns.
2. *Cargar el valor "11111100"*.
3. *Realizar 4 desplazamientos a la izquierda*. Comprobar que el circuito pasa consecutivamente por los valores "11111000", "11110000", "11100000" y "11000000".
4. *Cargar el valor "10111100"*.
5. *Realizar 4 desplazamientos a la derecha*. Comprobar que el circuito pasa por los valores "01011110", "00101111", "00010111" y "00001011".

El banco de pruebas debe comprobar que los valores de las señales de salida 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.