INGENIERÍA DE COMPUTADORES III

Examen de convocatoria extraordinaria Septiembre 2020 y examen de convocatoria ordinaria de Junio en Septiembre, en Aula virtual de Examen UNED (AvEx)

INSTRUCCIONES

- El examen debe realizarse de manera individual.
- No está permitido el uso de ningún material.
- El examen se compone de tres preguntas de desarrollo.

  La puntuación de cada pregunta de los bloques 1 y 2 es de 3 puntos.

  La puntuación de la pregunta del bloque 3 es de 4 puntos.

  La aplicación mostrará aleatoriamente una pregunta de cada uno de los bloques, de modo que el examen a desarrollar tiene 3 preguntas.
- Para aprobar el examen debe obtener una puntuación igual a superior a 5 puntos.
- Dispone de 1 hora para realizar el examen.

Preguntas Bloque 1 (3 puntos)

Pregunta 1.1

Dado el siguiente fragmento de código VHDL indique cuál es el valor de las señales a, b, c en los instantes 0ns, delta (δ), 2 delta (2δ), 5ns, 5ns + delta (5ns+δ), 15 ns, 15 ns+delta (15ns+δ), 25 ns y 25 ns+delta (25ns+δ).

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono is
end entity crono;
architecture crono of crono is
  signal x1, x2, x3, a, b : std_logic;
  signal c  : std_logic := '0';
begin
  x1 <= '1',
        '0' after 5 ns,
        '1' after 15 ns;
  x2 <= '0',
        '1' after 25 ns;
  x3 <= '0',
        '1' after 10 ns,
        '0' after 20 ns;
  c <= x1 after 10 ns;
  Proc1: process (x1, x2)
  begin
    a <= x1 and x2;
    b <= a or x3;
  end process;
end architecture crono;
```

Pregunta 1.2

Dado el siguiente fragmento de código VHDL indique cuál es el valor de las señales d, e, f en los instantes 0ns, delta (δ), 2 delta (2δ), 5ns, 5ns + delta (5ns+δ), 10 ns, 15 ns, 15 ns+delta (15ns+δ), 20 ns y 25 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono is
end entity crono;
architecture crono of crono is
  signal x1, x2, x3, d, e, f : std_logic;
begin
  x1 <= '0',
        '1' after 5 ns,
        '0' after 15 ns;
  x2 <= '1',
        '0' after 25 ns;
  x3 <= '0',
        '1' after 10 ns,
        '0' after 20 ns;
  d <= x1 after 5 ns;
  Proc1: process (x1, x2)
  begin
    e <= x1 and x2;
    f <= d or x3;
  end process;
end architecture crono;
```

Bloque 2 (3 puntos)

Pregunta 2.1

Escriba en VHDL la **architecture** de un circuito combinacional que tiene una señal de entrada de 4 bits `x`, una señal de selección `s` y una salida `y` de 8 bits. La señal de salida es el resultado de desplazar la concatenación de `"0000"` y la señal `x` el número de bits determinado por la señal de selección `s`. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity Desplazador is
   port(  y: out std_logic_vector (7 downto 0);
            s: in std_logic_vector (2 downto 0);
            x: in std_logic_vector ( 3 downto 0));
end entity Desplazador;
```

s	Operación realizada sobre "0000"&x
"000"	Pasa sin desplazar
"001"	Rota a la izquierda 1 posición
"010"	Rota a la izquierda 2 posiciones
"011"	Rota a la izquierda 3 posiciones
"100" o "101" o "110" o "111"	Rota a la izquierda 4 posiciones

Emplee en el diseño un bloque **process** con una sentencia **case**.

Pregunta 2.2

Escriba en VHDL la **architecture** de un circuito combinacional que tiene una señal de entrada de 4 bits `x`, una señal de selección `s` y una salida `y` de 8 bits. La señal de salida es el resultado de desplazar la concatenación de `"0000"` y la señal `x` el número de bits determinado por la señal de selección `s`. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity Desplazador is
   port(  y: out std_logic_vector (7 downto 0);
            s: in std_logic_vector (2 downto 0);
            x: in std_logic_vector ( 3 downto 0));
end entity Desplazador;
```

s	Operación realizada sobre "0000"&x
"000"	Pasa sin desplazar
"001"	Rota a la izquierda 1 posición
"010"	Rota a la izquierda 2 posiciones
"011"	Rota a la izquierda 3 posiciones
"100" o "101" o "110" o "111"	Rota a la izquierda 4 posiciones

Emplee en el diseño una sentencia concurrente **when**–**else**.

Bloque 3 (4 puntos)

Pregunta 3.1

Escriba en VHDL la **architecture** de un circuito secuencial síncrono que opera en el flanco de subida de la señal de reloj. Se supone que la señal de reloj de entrada tiene un periodo de 1 segundo. Este circuito tiene tres señales de entrada de 1 bit: la señal de reloj `clk`, la señal de habilitación `ena` y la señal de reset asíncrona activa a nivel 1 `rst`. El circuito tiene una señal de salida de un bit llamada `luz`. El circuito tiene los tres estados siguientes:

- Estado STOP: la señal `luz` tiene valor '0'. Se realiza la transición de este estado al estado ON cuando la señal `ena` pasa a valer '1'. Se pasa a este estado de modo asíncrono cuando se resetea el circuito. También se pasa a este estado (desde cualquiera de los otros dos) en el primer flanco de subida de la señal de reloj después de que la señal ena toma el valor '0'.
- Estado ON: la señal `luz` tiene valor '1'. Se permanece en este estado 15 segundos siempre que la señal `ena` tenga el valor '1' y no se resetee el circuito. Pasados los 15 segundos, se pasa al estado OFF.
- Estado OFF: la señal `luz` tiene el valor '0'. Se permanece en este estado 10 segundos siempre que la señal `ena` tenga el valor '1' y no se resetee el circuito. Pasados los 10 segundos, se pasa al estado ON.

La **entity** de este circuito se muestra a continuación.

```
entity controlLED is
  port ( luz          : out std_logic;
         clk, ena, rst : in  std_logic );
end entity controlLED;
```

Pregunta 3.2

Escriba en VHDL la **architecture** de un circuito secuencial que opera en el flanco de subida de la señal de reloj. El circuito tiene una señal de salida `out1` y dos señales de entrada, la señal de reloj `clk` y la señal de reset asíncrona activa a nivel alto `rst`. La señal de salida `out1` tiene el valor '1' mientras la señal `rst` tiene el valor '1'. Cuando la señal `rst` pasa a tener valor '0', la señal `out1` mantiene el valor '1' durante tres periodos de la señal de reloj `clk` y va cambiando de valor de modo que su periodo sea seis veces el periodo de la señal `clk`. La **entity** de este circuito se muestra a continuación.

```
entity circMultPeriodo  is
  port ( out1: out std_logic;
         clk, rst : in  std_logic );
end entity circMultPeriodo;
```