

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 ($5ns+\delta$), 15 ns, 15 ns+delta ($15ns+\delta$), 25 ns y 25 ns+delta ($25ns+\delta$).

```
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;
    Procl: process (x1, x2)
    begin
        a <= x1 and x2;
        b <= a or x3;
    end process;
end architecture crono;
```

Solución

- Señal a: 'U' en 0ns, 'U' en delta, '0' en 2delta, '0' en 5ns, '0' en 5ns+delta, '0' en 15 ns, '0' en 15ns+delta, '0' en 25 ns y '1' en 25ns+delta.
- Señal b: 'U' en 0ns, 'U' en delta, 'U' en 2delta, 'U' en 5ns, '0' en 5ns+delta, '0' en 15 ns, '1' en 15ns+delta, '1' en 25 ns y '0' en 25ns+delta.
- Señal c: '0' en 0ns, '0' en delta, '0' en 2delta, '0' en 5ns, '0' en 5ns+delta, '0' en 15 ns, '0' en 15ns+delta, '1' en 25 ns y '1' en 25ns+delta.

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 ($5ns+\delta$), 10 ns, 15 ns, 15 ns+delta ($15ns+\delta$), 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;
```

Señal f: 'U' en 0ns, 'U' en delta, 'U' en 2delta, 'U' en 5ns, '0' en 5n+delta, '0' en 10 ns, '0' en 15 ns, '1' en 15 ns+delta, '1' en 20ns y '1' en 25ns.

Solución

- Señal d: 'U' en 0ns, 'U' en delta, 'U' en 2delta, '0' en 5ns, '0' en 5ns+delta, '1' en 10 ns, '1' en 15 ns, '1' en 15ns+delta, '0' en 20 ns y '0' en 25ns.
- Señal e: 'U' en 0ns, 'U' en delta, '0' en 2delta, '0' en 5ns, '1' en 5ns+delta, '1' en 10 ns, '1' en 15 ns, '0' en 15ns+delta, '0' en 20 ns y '0' en 25ns.
- Señal f: 'U' en 0ns, 'U' en delta, 'U' en 2delta, 'U' en 5ns, '0' en 5ns+delta, '0' en 10 ns, '0' en 15 ns, '1' en 15ns+delta, '1' en 20 ns y '1' en 25ns.

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**.

Solución

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
architecture DespCase of Desplazador is
    signal x0: std_logic_vector(7 downto 0);
begin
    x0 <= "0000" &x;
    process(s,x)
    begin
        case s is
            when "000" =>
                y <= x0;
            when "001" =>
                y <= x0(6 downto 0) & x0(7);
            when "010" =>
                y <= x0(5 downto 0)&x0(7 downto 6);
            when "011" =>
                y <=x0(4 downto 0)&x0(7 downto 5);
            when others =>
                y <=x0(3 downto 0)&x0(7 downto 4);
        end case;
    end process;
end DespCase;
```

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**.

Solución

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
architecture DespWhen of Desplazador is
  signal x0: std_logic_vector(7 downto 0);
begin
  x0 <= "0000"&x;
  y <= x0 when (s = "000") else
    x0(6 downto 0) & x0(7) when (s="001") else
    x0(5 downto 0) & x0(7 downto 6) when (s="010") else
    x0(4 downto 0)&x0(7 downto 5) when (s="011") else
    x0(3 downto 0)&x0(7 downto 4);
end DespWhen;
```

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;
```

Solución

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.ALL;

architecture controlled of controlled is
constant s_STOP: std_logic_vector(1 downto 0) := "00"; -- Estados
constant s_ON: std_logic_vector(1 downto 0) := "01";
constant s_OFF: std_logic_vector(1 downto 0) := "10";
signal internal_state: std_logic_vector(1 downto 0);
begin
process(clk, rst)
variable tempON : integer range 0 to 15;
variable tempOFF : integer range 0 to 10;
begin
if (rst = '1') then
    internal_state <= s_STOP;
elsif (rising_edge(clk))then
    case internal_state is
        when s_STOP =>
            if (ena = '1') then
                internal_state <= s_ON;
                tempON := 0;
                tempOFF := 0;
            end if;
        when s_ON =>
            if (ena = '0') then
                internal_state <= s_STOP;
            elsif (tempON < 14) then
                tempON := tempON + 1;
            else
                tempON := 0;
                internal_state <= s_OFF;
            end if;
        when s_OFF =>
            if (ena = '0') then
                internal_state <= s_STOP;
            elsif (tempOFF < 9) then
                tempOFF := tempOFF + 1;
            else
                tempOFF := 0;
                internal_state <= s_ON;
            end if;
        when others =>
            internal_state <= s_STOP;
    end case;
end if;
end process;
    luz <= '1' when (internal_state = s_ON) else '0';
end 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;
```

Solución

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.ALL;

architecture circMultPeriodo of circMultPeriodo is
  signal count1: integer range 0 to 6:=0;
  signal aux: std_logic:='1';
begin
  process(clk, rst)
  begin
    if (rst = '1') then
      aux <= '1';
    elsif (rising_edge(clk)) then
      count1 <= count1 + 1;
      if (count1 = 2) then
        aux <= not aux;
        count1 <= 0;
      end if;
    end if;
  end process;
  out1 <= aux;
end circMultPeriodo;
```