

INGENIERÍA DE COMPUTADORES III

Examen semana 2 de convocatoria Junio 2020, en el 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 x1, x2, x3, x4 y x5 en los instantes 0, delta (δ), 5ns, 10ns+delta ($10ns + \delta$), 15 ns, 20 ns, 20 ns + delta ($20ns + \delta$) y 25 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono4 is
end entity crono4;
architecture crono4 of crono4 is
    signal x1, x2, x3, x4, x5 : std_logic;
begin
    x1 <= '1',
        '0' after 5 ns,
        '1' after 15 ns,
        '0' after 20 ns;
    x2 <= '0',
        '1' after 15 ns,
        '0' after 25 ns;
    x3 <= x1 after 10 ns;
    Procl: process
    begin
        for i in 0 to 2 loop
            x4 <= x1 or x2;
            x5 <= x4;
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture crono4;
```

Pregunta 1.2

Dado el siguiente fragmento de código VHDL, indique cuál es el valor de las señales x1, x2, x3, x4 y x5 en los instantes 0, delta (δ), 5ns, 10ns+delta ($10ns + \delta$), 15 ns, 20 ns, 20 ns + delta ($20ns + \delta$) y 25 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono4 is
end entity crono4;
architecture crono4 of crono4 is
    signal x1, x2, x3, x4, x5 : std_logic;
begin
    x1 <= '0',
        '1' after 5 ns,
        '0' after 15 ns,
        '1' after 20 ns;
    x2 <= '0',
        '1' after 15 ns,
        '0' after 25 ns;
    x3 <= x1 after 10 ns;
    Procl: process
    begin
        for i in 0 to 2 loop
            x4 <= x1 xor x2;
            x5 <= x4;
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture crono4;
```

Pregunta 1.3

Dado el siguiente fragmento de código VHDL, indique cuál es el valor de las señales x1, x2, x3, x4 y x5 en los instantes 0, delta (δ), 5ns, 10ns+delta ($10ns + \delta$), 15 ns, 20 ns, 20 ns + delta ($20ns + \delta$), 25 ns y 35 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono4 is
end entity crono4;
architecture crono4 of crono4 is
    signal x1, x2, x3, x4, x5 : std_logic;
begin
    x1 <= '1',
        '0' after 5 ns,
        '1' after 15 ns,
        '0' after 20 ns;
    x2 <= '1',
        '0' after 15 ns,
        '1' after 25 ns;
    x3 <= x1 after 15 ns;
    Procl: process
    begin
        for i in 0 to 2 loop
            x4 <= x1 or x2;
            x5 <= x4;
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture crono4;
```

Pregunta 1.4

Dado el siguiente fragmento de código VHDL, indique cuál es el valor de las señales x1, x2, x3, x4 y x5 en los instantes 0, delta (δ), 5 ns, 10 ns, 10 ns + delta ($10ns + \delta$), 15 ns, 20 ns, 20 ns + delta ($20ns + \delta$) y 25 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono4 is
end entity crono4;
architecture crono4 of crono4 is
    signal x1, x2, x3, x4, x5 : std_logic;
begin
    x1 <= '1',
        '0' after 5 ns,
        '1' after 15 ns,
        '0' after 20 ns;
    x2 <= '1',
        '0' after 15 ns,
        '1' after 25 ns;
    x3 <= x1 after 5 ns;
    Procl: process
    begin
        for i in 0 to 2 loop
            x4 <= x1 xor x2;
            x5 <= x4;
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture crono4;
```

Bloque 2 (3 puntos)

Pregunta 2.1

Escriba en VHDL la **architecture** de una ALU de 8 bits. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity ALU is
  port( y: out std_logic_vector (7 downto 0);
        s: in std_logic_vector (1 downto 0);
        A: in std_logic_vector ( 7 downto 0);
        B: in std_logic_vector ( 7 downto 0));
end entity ALU;
```

s	Operación
"00"	not B
"01"	A+B
"10"	B-A
"11"	B+1

Los números A y B se interpretan como enteros con signo para las operaciones aritméticas.

En el diseño sólo puede emplear las siguientes librerías:

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

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

Pregunta 2.2

Escriba en VHDL la **architecture** de una ALU de 8 bits. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity ALU is
  port( y: out std_logic_vector (7 downto 0);
        s: in std_logic_vector (1 downto 0);
        A: in std_logic_vector ( 7 downto 0);
        B: in std_logic_vector ( 7 downto 0));
end entity ALU;
```

Los números A y B se interpretan como enteros con signo para las operaciones aritméticas.

En el diseño sólo puede emplear las siguientes librerías:

s	Operación
"00"	not B
"01"	A+B
"10"	B-A
"11"	B+1

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

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

Pregunta 2.3

Escriba en VHDL la **architecture** de una ALU de 8 bits. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity ALU is
    port( y: out std_logic_vector (7 downto 0);
          s: in std_logic_vector (1 downto 0);
          A: in std_logic_vector ( 7 downto 0);
          B: in std_logic_vector ( 7 downto 0));
end entity ALU;
```

s	Operación
"00"	not B
"01"	A+B
"10"	B-A
"11"	B+1

Los números A y B se interpretan como enteros con signo para las operaciones aritméticas.

En el diseño sólo puede emplear las siguientes librerías:

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

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

Pregunta 2.4

Escriba en VHDL la **architecture** de una ALU de 8 bits. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity ALU is
  port( y: out std_logic_vector (7 downto 0);
        s: in std_logic_vector (1 downto 0);
        A: in std_logic_vector ( 7 downto 0);
        B: in std_logic_vector ( 7 downto 0));
end entity ALU;
```

s	Operación
"00 "	not B
"01 "	A+B
"10 "	B-A
"11 "	B+1

Los números A y B se interpretan como enteros con signo para las operaciones aritméticas.

En el diseño sólo puede emplear las siguientes librerías:

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

Emplee en el diseño una sentencia concurrente **with-select**.

Bloque 3 (4 puntos)

Pregunta 3.1

Escriba en VHDL la architecture de un circuito secuencial capaz de detectar cuando le llega la secuencia "011" por su entrada X. El circuito tiene una señal de reloj (`clk`), una entrada serie de un bit (`X`), una señal de reset asíncrona activa en 0 (`reset`) y una señal de salida de un bit (`Y`). La señal `Y` vale '1' cuando se detecta la secuencia. En cualquier otro caso, la señal `Y` tiene valor '0'. La señal `reset` pone el circuito en su estado inicial (no se ha reconocido ningún elemento de la secuencia). Todos los cambios, salvo el reseteo del circuito, tienen lugar en el flanco de subida de la señal de reloj.

La **entity** del circuito se muestra a continuación.

```
entity detector is
  port ( Y      : out std_logic;
        reset   : in  std_logic;
        clk     : in  std_logic;
        X       : in  std_logic );
end entity detector;
```

Escriba en VHDL la **architecture** que describe el comportamiento del circuito en términos de una máquina de Moore.

Pregunta 3.2

Escriba en VHDL la architecture de un circuito secuencial capaz de detectar cuando le llega la secuencia "011" por su entrada X. El circuito tiene una señal de reloj (`clk`), una entrada serie de un bit (`X`), una señal de reset asíncrona activa en 0 (`reset`) y una señal de salida de un bit (`Y`). La señal `Y` vale '1' cuando se detecta la secuencia. En cualquier otro caso, la señal `Y` tiene valor '0'. La señal `reset` pone el circuito en su estado inicial (no se ha reconocido ningún elemento de la secuencia). Todos los cambios, salvo el reseteo del circuito, tienen lugar en el flanco de subida de la señal de reloj.

La **entity** del circuito se muestra a continuación.

```
entity detector is
  port ( Y      : out std_logic;
        reset   : in  std_logic;
        clk     : in  std_logic;
        X       : in  std_logic );
end entity detector;
```

Escriba en VHDL la **architecture** que describe el comportamiento del circuito en términos de una máquina de Mealy.

Pregunta 3.3

Escriba en VHDL la architecture de un circuito secuencial capaz de detectar cuando le llega la secuencia "011" por su entrada X. El circuito tiene una señal de reloj (clk), una entrada serie de un bit (X), una señal de reset síncrona activa en 1 (reset) y una señal de salida de un bit (Y). La señal Y vale '1' cuando se detecta la secuencia. En cualquier otro caso, la señal Y tiene valor '0'. La señal reset pone el circuito en su estado inicial (no se ha reconocido ningún elemento de la secuencia). Todos los cambios tienen lugar en el flanco de subida de la señal de reloj.

La **entity** del circuito se muestra a continuación.

```
entity detector is
  port ( Y      : out std_logic;
        reset   : in  std_logic;
        clk     : in  std_logic;
        X       : in  std_logic );
end entity detector;
```

Escriba en VHDL la **architecture** que describe el comportamiento del circuito en términos de una máquina de Moore.

Pregunta 3.4

Escriba en VHDL la architecture de un circuito secuencial capaz de detectar cuando le llega la secuencia "011" por su entrada X. El circuito tiene una señal de reloj (clk), una entrada serie de un bit (X), una señal de reset síncrona activa en 1 (reset) y una señal de salida de un bit (Y). La señal Y vale '1' cuando se detecta la secuencia. En cualquier otro caso, la señal Y tiene valor '0'. La señal reset pone el circuito en su estado inicial (no se ha reconocido ningún elemento de la secuencia). Todos los cambios tienen lugar en el flanco de subida de la señal de reloj.

La **entity** del circuito se muestra a continuación.

```
entity detector is
  port ( Y      : out std_logic;
        reset   : in  std_logic;
        clk     : in  std_logic;
        X       : in  std_logic );
end entity detector;
```

Escriba en VHDL la **architecture** que describe el comportamiento del circuito en términos de una máquina de Mealy.