

INGENIERÍA DE COMPUTADORES III

Examen semana 1 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 a, b, c, d, s1, s2, s3 en los instantes 0, delta (δ), 2 delta (2δ), 3 delta (3δ) y 4 delta (4δ).

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono4 is
end entity crono4;
architecture crono4 of crono4 is
    signal a,b,c,d,s1,s2,s3: std_logic;
begin
    procl: process
    begin
        a <= '0';
        b <= '1';
        c <= '0';
        wait;
    end process;
    s1<= a or b;
    s2 <= b and c;
    s3 <= s1 xor s2;
    d<= not s3;
end architecture crono4;
```

Pregunta 1.2

Dado el siguiente fragmento de código VHDL, indique cuál es el valor de las señales a, b, c, d, s1, s2, s3 en los instantes 0, delta (δ), 2 delta (2δ) y 3 delta (3δ).

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono5 is
end entity crono5;
architecture crono5 of crono5 is
    signal a,b,c,d,s1,s2,s3: std_logic;
begin
    procl: process
    begin
        a <= '1';
        b <= '0';
        c <= '1';
        wait;
    end process;
    s1<= a xor b;
    s2 <= b and c;
    s3 <= s1 xor s2;
    d<= not s2;
end architecture crono5;
```

Pregunta 1.3

Dado el siguiente fragmento de código VHDL, indique cuál es el valor de las señales a, b, c, d, s1, s2, s3 en los instantes 0, delta (δ), 2 delta (2δ), 10+ delta ($10 + \delta$), 10+2 delta ($10 + 2\delta$), 10+3 delta ($10 + 3\delta$) y 10+4delta ($10 + 4\delta$).

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono4 is
end entity crono4;
architecture crono4 of crono4 is
    signal a,b,c,d,s1,s2,s3: std_logic;
begin
    procl: process
    begin
        a <= '0';
        b <= '1';
        wait for 10 ns;
        c <= '0';
        wait;
    end process;
    s1<= a or b;
    s2 <= b and c;
    s3 <= s1 xor s2;
    d<= not s3;
end architecture crono4;
```

Pregunta 1.4

Dado el siguiente fragmento de código VHDL, indique cuál es el valor de las señales a, b, c, d, s1, s2, s3 en los instantes 0, delta (δ), 2 delta (2δ), 10+ delta ($10 + \delta$), 10+2 delta ($10 + 2\delta$) y 10+3 delta ($10 + 3\delta$).

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono5 is
end entity crono5;
architecture crono5 of crono5 is
    signal a,b,c,d,s1,s2,s3: std_logic;
begin
    procl: process
    begin
        a <= '1';
        wait for 10 ns;
        b <= '0';
        c <= '1';
        wait;
    end process;
    s1<= a and b;
    s2 <= b and c;
    s3 <= s1 or s2;
    d<= a or s2;
end architecture crono5;
```

Bloque 2 (3 puntos)

Pregunta 2.1

Escriba en VHDL la **architecture** de un circuito desplazador de 8 bits. 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 (1 downto 0);
        x: in std_logic_vector ( 7 downto 0));
end entity Desplazador;
```

s	Operación
"00"	Pasa el dato
"01"	Desplaza a la izquierda una posición rellenando con ceros
"10"	Desplaza a la derecha una posición rellenando con unos
"11"	Desplaza a la derecha dos posiciones rellenando con ceros

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

Pregunta 2.2

Escriba en VHDL la **architecture** de un circuito desplazador de 8 bits. 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 (1 downto 0);
        x: in std_logic_vector ( 7 downto 0));
end entity Desplazador;
```

s	Operación
"00"	Pasa el dato
"01"	Desplaza a la izquierda una posición rellenando con ceros
"10"	Desplaza a la derecha una posición rellenando con unos
"11"	Desplaza a la derecha dos posiciones rellenando con ceros

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

Pregunta 2.3

Escriba en VHDL la **architecture** de un circuito desplazador de 8 bits. 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 (1 downto 0);
        x: in std_logic_vector ( 7 downto 0));
end entity Desplazador;
```

s	Operación
"00"	Pasa el dato
"01"	Desplaza a la izquierda una posición rellenando con ceros
"10"	Desplaza a la derecha una posición rellenando con unos
"11"	Desplaza a la derecha dos posiciones rellenando con ceros

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

Pregunta 2.4

Escriba en VHDL la **architecture** de un circuito desplazador de 8 bits. 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 (1 downto 0);
        x: in std_logic_vector ( 7 downto 0));
end entity Desplazador;
```

s	Operación
"00"	Pasa el dato
"01"	Desplaza a la izquierda una posición rellenando con ceros
"10"	Desplaza a la derecha una posición rellenando con unos
"11"	Desplaza a la derecha dos posiciones rellenando con ceros

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

Bloque 3 (4 puntos)

Pregunta 3.1

Escriba en VHDL la **architecture** de un circuito contador de 4 bits con señal de reset asíncrona activa a nivel alto. El circuito se comporta de la forma siguiente. Cuando la señal de `reset` se pone a 1, la salida del contador se pone al valor "0001". Mientras la señal de `reset` está a 0, el contador funciona de manera síncrona, rotando circularmente hacia la derecha el patrón de bits: 0001, 1000, 0100, 0010, 0001, 1000, 0100, y así sucesivamente. La operación de desplazamiento se produce en el flanco de subida de la señal de reloj. La **entity** de este circuito se muestra a continuación.

```
entity counter is
  port ( q          : out std_logic_vector(3 downto 0);
        clk, reset : in  std_logic );
end entity counter;
```

Pregunta 3.2

Escriba en VHDL la **architecture** de un circuito contador de 4 bits con señal de reset asíncrona activa a nivel bajo. El circuito se comporta de la forma siguiente. Cuando la señal de `reset` se pone a 0, la salida del contador se pone al valor "0001". Mientras la señal de `reset` está a 1, el contador funciona de manera síncrona, rotando circularmente hacia la derecha el patrón de bits: 0001, 1000, 0100, 0010, 0001, 1000, 0100, y así sucesivamente. La operación de desplazamiento se produce en el flanco de subida de la señal de reloj. La **entity** de este circuito se muestra a continuación.

```
entity counter is
  port ( q          : out std_logic_vector(3 downto 0);
        clk, reset : in  std_logic );
end entity counter;
```

Pregunta 3.3

Escriba en VHDL la **architecture** de un circuito contador de 4 bits con señal de reset síncrona activa a nivel alto. El circuito se comporta de la forma siguiente. Cuando la señal de `reset` se pone a 1 y se está en el flanco de subida de la señal de reloj, la salida del contador se pone al valor "0001". Mientras la señal de `reset` está a 0, el contador funciona de manera síncrona, rotando circularmente hacia la derecha el patrón de bits: 0001, 1000, 0100, 0010, 0001, 1000, 0100, y así sucesivamente. La operación de `reset` y de desplazamiento se producen en el flanco de subida de la señal de reloj. La **entity** de este circuito se muestra a continuación.

```
entity counter is
  port ( q          : out std_logic_vector(3 downto 0);
        clk, reset : in  std_logic );
end entity counter;
```


Pregunta 3.4

Escriba en VHDL la architecture de un circuito contador de 4 bits con señal de reset síncrona activa a nivel bajo. El circuito se comporta de la forma siguiente. Cuando la señal de `reset` se pone a 0 y se está en el flanco de subida de la señal de reloj, la salida del contador se pone al valor "0001". Mientras la señal de `reset` está a 1, el contador funciona de manera síncrona, rotando circularmente hacia la derecha el patrón de bits: 0001, 1000, 0100, 0010, 0001, 1000, 0100, y así sucesivamente. La operación de reset y de desplazamiento se producen en el flanco de subida de la señal de reloj.

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

```
entity counter is
  port ( q          : out std_logic_vector(3 downto 0);
        clk, reset : in  std_logic );
end entity counter;
```