

# INGENIERÍA DE COMPUTADORES III

Examen de convocatoria Septiembre 2021, 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:
  - 1 pregunta del bloque 1, 1 pregunta del bloque 2 y 1 pregunta del bloque 3.
  - La puntuación de la pregunta del bloques 1 es de 3 puntos.
  - La puntuación de la pregunta del bloque 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 en los instantes 0,  $\delta$ , 5ns,  $5ns + \delta$ , 10 ns,  $10ns + \delta$ , 11 ns, 20 ns,  $20ns + \delta$ , 21ns, 30ns y 35 ns. Para cada señal, solo es necesario dar su valor en los instantes de tiempo donde se produce un cambio de valor de dicha señal

```
library IEEE;
use IEEE.std_logic_1164.all;
entity codigo is
end entity codigo;
architecture codigo of codigo is
    signal x1, a, b, c, d: std_logic;
begin
    a <= '0',
        '1' after 10 ns;
    a <= '0',
        '1' after 5 ns;
    x1 <= '0',
        '1' after 10 ns,
        '0' after 20 ns,
        '1' after 25 ns;
    b <= x1 after 10 ns;
    c <= transport x1 after 10 ns;
    d <= b or c after 1 ns;
end architecture codigo;
```

## Pregunta 1.2

Dado el siguiente fragmento de código VHDL indique cuál es el valor de las señales a, b, c y d en los instantes 0, delta ( $\delta$ ), 2 ns, 5 ns, 7 ns, 8 ns, 10 ns, 12 ns, 15 ns, 18 ns, 20 ns y 23 ns. Para cada señal, solo es necesario dar su valor en los instantes de tiempo donde se produce un cambio de valor de dicha señal.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity codigo is
end entity codigo;
architecture codigo of codigo is
  signal x1, a, b, c, d: std_logic;
begin
  a <= '0',
      '1' after 2 ns;
  a <= '0',
      '1' after 5 ns;
  x1 <= '0',
      '1' after 2 ns,
      '0' after 10 ns,
      '1' after 15 ns;
  b <= x1 after 5 ns;
  c <= transport x1 after 8 ns;
  d <= b or c after 5 ns;
end architecture codigo;
```

## 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  $s$  de dos bits y una salida  $y$  de 4 bits. La señal de salida es el resultado de rotar la señal  $x$  el número de bits determinado por la señal de selección  $s$ . Emplee en el diseño un bloque **process** con una sentencia **case**. En la descripción del circuito no se pueden emplear ni operadores ni funciones de desplazamiento lógico. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity Desplazador is
  port( y: out std_logic_vector (3 downto 0);
        s: in std_logic_vector (1 downto 0);
        x: in std_logic_vector ( 3 downto 0));
end entity Desplazador;
```

s	Operación realizada sobre x
00	Pasa sin desplazar
01	Rota a la derecha una posición
10	Rota a la derecha dos posiciones
11	Rota a la derecha tres posiciones

### 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  $s$  de dos bits y una salida  $y$  de 4 bits. La señal de salida es el resultado de rotar la señal  $x$  el número de bits determinado por la señal de selección  $s$ . Emplee en el diseño una sentencia concurrente **when-else**. En la descripción del circuito no se pueden emplear ni operadores ni funciones de desplazamiento lógico. La **entity** y la tabla de operaciones de este circuito se muestran a continuación.

```
entity Desplazador is
  port( y: out std_logic_vector (3 downto 0);
        s: in std_logic_vector (1 downto 0);
        x: in std_logic_vector ( 3 downto 0));
end entity Desplazador;
```

s	Operación realizada sobre x
00	Pasa sin desplazar
01	Rota a la derecha una posición
10	Rota a la derecha dos posiciones
11	Rota a la derecha tres posiciones

## 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 dos señales de entrada de 1 bit: la señal de reloj `clk` y la señal de reset asíncrona activa a nivel 1 `rst`. El circuito tiene una señal de salida de un bit llamada `salida`.

El circuito tiene los dos estados siguientes:

- Estado CERO: la señal salida tiene valor '0'. Se permanece en este estado 10 segundos siempre que no se resetee el circuito. Pasados los 10 segundos, se pasa al estado UNO. Cuando la señal `rst` se pone a '1', se vuelve a iniciar la cuenta de 10 segundos.
- Estado UNO: la señal salida tiene el valor '1'. Se permanece en este estado 4 segundos siempre que no se resetee el circuito. Pasados los 4 segundos, se pasa al estado CERO. Cuando la señal `rst` se pone a '1', el circuito pasa al estado CERO.

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

```
entity control is
  port ( salida          : out std_logic;
        clk, rst       : in  std_logic );
end entity control;
```

Para el diseño del circuito emplee únicamente las siguientes librerías:

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

## Pregunta 3.2

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 dos señales de entrada de 1 bit: la señal de reloj `clk` y la señal de reset asíncrona activa a nivel 1 `rst`. El circuito tiene una señal de salida de dos bits llamada `salida`.

El circuito tiene los dos estados siguientes:

- Estado CERO: la señal salida tiene valor "00". Se permanece en este estado 6 segundos siempre que no se resetee el circuito. Pasados los 6 segundos, se pasa al estado UNO. Cuando la señal `rst` se pone a '1', se vuelve a iniciar la cuenta de 6 segundos.
- Estado UNO: la señal salida tiene el valor "11". Se permanece en este estado 2 segundos siempre que no se resetee el circuito. Pasados los 2 segundos, se pasa al estado CERO. Cuando la señal `rst` se pone a '1', el circuito pasa al estado CERO.

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

```
entity control is
  port ( salida          : out std_logic_vector(1 downto 0);
        clk, rst       : in  std_logic );
end entity control;
```

Para el diseño del circuito emplee únicamente las siguientes librerías:

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