

INSTRUCCIONES:

1. Resuelva este ejercicio en las mismas condiciones en que realizará el examen: dos horas de tiempo y sin emplear ningún material.
2. Revise sus contestaciones, empleando para ello el texto y el simulador que esté usando para estudiar la asignatura.
3. Compare sus respuestas revisadas con la solución.

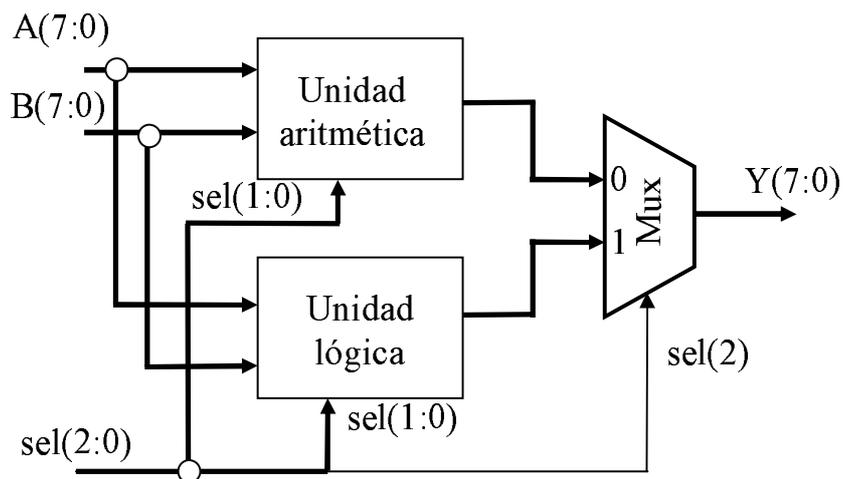
Pregunta 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales in1, in2, s1, s2, s3 y s4 entre los instantes 0 y 60 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity codigo is
end entity codigo;
architecture codigo of codigo is
    signal s1, s2, s3, s4 : std_logic;
    signal in1, in2      : std_logic;
begin
    s1 <= in1 nand in2 after 5 ns;
    process
        variable var1, var2 : std_logic;
    begin
        for i in 0 to 4 loop
            var1 := in1 nand in2;
            var2 := var1;
            s2   <= in1 nand in2;
            s3   <= s2;
            s4 <= var2;
            wait for 10 ns;
        end loop;
        wait;
    end process;
    in1 <= '1';
    in2 <= '0',
        '1' after 10 ns, '0' after 11 ns,
        '1' after 20 ns, '0' after 30 ns,
        '1' after 40 ns, '0' after 50 ns;
end architecture codigo;
```

Pregunta 2 (3 puntos)

A continuación, se muestra el circuito, la tabla de operaciones y la **entity** de una ALU. La ALU realiza operaciones sobre dos operandos de 8 bits, denominados A y B. La salida de la ALU se selecciona mediante el bit más significativo de la señal *sel*, mientras que la operación que realiza se especifica mediante los otros dos bits de esta señal.



Unidad	sel	Operación
Aritmética	000	A
	001	A+B
	010	A+1
	011	A-1
Lógica	100	not A
	101	A and B
	110	A or B
	111	A xor B

```
entity ALU is
  port( Y      : out std_logic_vector ( 7 downto 0 );
        A, B   : in  std_logic_vector ( 7 downto 0 );
        sel    : in  std_logic_vector ( 2 downto 0 ) );
end entity ALU;
```

Escriba en VHDL la **architecture** que describe el comportamiento de la ALU, empleando para ello tres sentencias de asignación concurrente de selección.

Puede realizar las hipótesis de diseño adicionales que estime convenientes, siempre y cuando las explique detalladamente y no estén en contradicción con las especificaciones anteriores.

Pregunta 3 (3 puntos)

Diseñe un circuito secuencial síncrono del tipo *máquina de Moore*, con una entrada serie de un bit, que detecte cuando recibe por dicha entrada tres o mas bits consecutivos de valor '1'. La **entity** del circuito se muestra a continuación.

```
entity detector is
  port( Y      : out std_logic;
        state : out std_logic_vector(1 downto 0);
        X      : in  std_logic;
        reset  : in  std_logic;
        clk    : in  std_logic);
end entity detector;
```

Como puede verse en la definición anterior de la **entity**, el circuito tiene una señal de reloj (`clk`), un entrada serie de un bit (`X`), una señal de reset asíncrona activa en '1' (`reset`), una señal que indica el estado en el que se encuentra el circuito (`state`) y una señal de salida de un bit (`Y`).

El funcionamiento del circuito es el siguiente. La señal `Y` se pone a '1' cuando por la entrada `X` se reciben tres o más bits consecutivos de valor '1'. Todos los cambios tienen lugar en el flanco de subida de la señal de reloj. Por otra parte, la señal `reset` pone asíncronamente el circuito en su estado inicial.

Escriba en VHDL la **architecture** que describe el comportamiento del circuito en términos de una máquina de Moore. Dibuje el diagrama de estados correspondiente al circuito que ha diseñado.

Puede realizar las hipótesis de diseño adicionales que estime convenientes, siempre y cuando las explique detalladamente y no estén en contradicción con las especificaciones anteriores.

Pregunta 4 (2 puntos)

Programe en VHDL el banco de pruebas del circuito secuencial que ha diseñado al contestar a la Pregunta 3. Explique detalladamente cómo el programa de test comprueba de manera sistemática el funcionamiento del circuito. El banco de pruebas debe comprobar que los valores obtenidos de la UUT coinciden con los esperados, mostrando el correspondiente mensaje en caso de que no coincidan. Al final del test, debe mostrarse un mensaje indicando el número total de errores.