

INGENIERÍA DE COMPUTADORES III

INSTRUCCIONES

Por favor, entregue todas las hojas de enunciado junto con el examen.

Dispone de 2 horas para realizar el examen.

MATERIAL PERMITIDO: Ninguno.

Pregunta 1 (2 puntos)

1.a) (1 punto) Dibuje el diagrama conceptual correspondiente al fragmento de código *Fragmento 1*.

1.b) (1 punto) Dibuje el diagrama conceptual correspondiente al fragmento de código *Fragmento 2*.

```
---- Fragmento 1-----  
signal a, b, r : unsigned (7 downto 0);  
signal x, y    : unsigned (3 downto 0);  
...  
r <=  a+b when x+y>0 else  
      a-b when x>y and y>0 else  
      a;
```

```
---- Fragmento 2-----  
signal s: std_logic_vector (1 downto 0);  
signal a, b : std_logic;  
...  
process(s)  
begin  
  case s is  
    when "00" =>  
      a <= '1';  
      b <= '0';  
    when "01" =>  
      a <= '1';  
      b <= '1';  
    when others =>  
      a <= '0';  
      b <= '0';  
  end case;  
end process;
```

Pregunta 2 (3 puntos)

Escriba el código VHDL de la **architecture** que describe el comportamiento de un circuito contador binario ascendente de 4 bits, con señal de reset asíncrona activa a nivel bajo y con carga paralelo síncrona. La **entity** del circuito se muestra a continuación.

```
entity contador is
  port ( count          : out std_logic_vector (3 downto 0);
        clk, reset, load : in  std_logic;
        data            : in  std_logic_vector (3 downto 0) );
end contador;
```

Las entradas al circuito son la señal de reloj (`clk`), la señal de reset asíncrono activo a nivel bajo (`reset`) y las señales de carga (`load`, `data`). La salida del circuito contador es la señal de 4 bits `count`.

El funcionamiento del circuito debe ser el siguiente:

- *Reset asíncrono activo a nivel bajo.* Cuando `reset` pasa a valer '0', el contador se pone al valor "0000".
- *Cuenta síncrona ascendente.* Mientras `reset` vale '1' y `load` vale '0', el contador se incrementa en uno en cada flanco de subida de la señal de reloj.
- *Carga paralelo síncrona.* Mientras `reset` vale '1' y `load` vale '1', en cada flanco de subida de la señal de reloj se carga en el contador el valor de la señal de entrada de 4 bits `data`.

El diseño debe realizarse describiendo el comportamiento del circuito. Para ello, deben emplearse tantas señales auxiliares y asignaciones concurrentes simples como sean necesarias, así como un único bloque **process**. Asimismo, en el diseño únicamente pueden emplearse los dos siguientes paquetes de la librería IEEE:

```
IEEE.std_logic_1164
IEEE.numeric_std
```

Pregunta 3 (2 puntos)

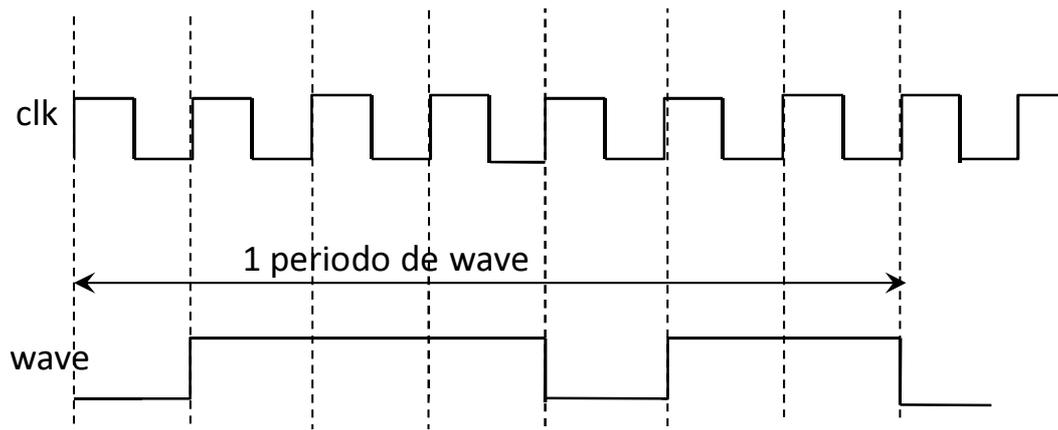
Programa en VHDL un banco de pruebas para el contador binario que ha diseñado al contestar a la Pregunta 2. La señal de reloj (`clk`) debe tener un periodo de 10 ns e inicialmente valer '0'. El programa de test debe realizar las acciones siguientes:

1. *Reset*. Resetear el contador.
2. *Cuenta síncrona ascendente*. Incrementar el valor del contador hasta "0011".
3. *Carga paralelo síncrona*. Cargar el valor "1101".
4. *Cuenta síncrona ascendente*. Incrementar el valor del contador hasta "1111".
5. *Reset*. Resetear el contador.

El correcto funcionamiento del circuito debe comprobarse mediante inspección visual. No es necesario que el banco de prueba compruebe que las salidas de la UUT son las esperadas. Dibuje el cronograma de evolución que han de seguir las señales de entrada y salida de la UUT.

Pregunta 4 (3 puntos)

Diseñe un generador de señales que obtiene la forma de onda mostrada a continuación (señal *wave*) a partir de una señal de reloj (`clk`). Describa su comportamiento como una máquina de estado finito de Moore cuyas transiciones se producen en el flanco de subida de la señal de reloj.



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

```
entity generador is
  port( wave : out std_logic;
        clk  : in  std_logic );
end entity generador;
```