

INGENIERÍA DE COMPUTADORES III

INSTRUCCIONES

Por favor, entregue esta primera hoja de enunciado junto con el examen.

Dispone de 2 horas para realizar el examen.

MATERIAL PERMITIDO: Ninguno.

Pregunta 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales `clk`, `x`, `y1` e `y2` entre los instantes 0 y 1000 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono2 is
end entity crono2;
architecture crono2 of crono2 is
    signal x, y1, y2 : std_logic;
    signal clk : std_logic:='0';
begin
    process (clk)
        variable a, b, c: std_logic;
    begin
        if ( rising_edge(clk) ) then
            a := x;
            b := a;
            c := b;
            y1 <= c;
            y2 <= y1;
        end if;
    end process;
    clk <= not clk after 100 ns;
    process is
    begin
        x<='0'; wait until falling_edge(clk);
        x<='1'; wait until falling_edge(clk);
        x<='1'; wait until falling_edge(clk);
        x<='0'; wait until falling_edge(clk);
        wait;
    end process;
end architecture crono2;
```

Pregunta 2 (3 puntos)

- 2.a)** (0.75 puntos) Escriba en VHDL la **architecture** de un multiplexor 4:1 empleando una sentencia concurrente condicional (**when - else**). La **entity** del multiplexor 4:1 se muestra a continuación.

```
entity mux4_1 is
  port (  y: out std_logic;
         s: in std_logic_vector(1 downto 0);
         x: in std_logic_vector(3 downto 0)      );
end entity mux4_1;
```

- 2.b)** (0.25 puntos) Dibuje el diagrama circuital de un circuito desplazador de barril de 4 bits empleando para ello únicamente multiplexores 4:1 como el descrito en el apartado 2.a. Este circuito desplazador tiene una señal de entrada *s* de 2 bits para seleccionar la operación, una señal de entrada *x* de 4 bits y una señal de salida *y* de 4 bits. La tabla de operaciones del circuito se muestra a continuación.

<i>s</i>	Operación	<i>y</i>
"00"	Ninguna	$x(3)x(2)x(1)x(0)$
"01"	Rota 1 bit a la izquierda	$x(2)x(1)x(0)x(3)$
"10"	Rota 2 bits a la izquierda	$x(1)x(0)x(3)x(2)$
"11"	Rota 3 bits a la izquierda	$x(0)x(3)x(2)x(1)$

- 2.c)** (2 puntos) Escriba en VHDL la **architecture** que describe la estructura del circuito desplazador siguiendo el diagrama dibujado en el apartado anterior y empleando el multiplexor cuyo diseño ha realizado al contestar el primer apartado. La **entity** del circuito se muestra a continuación.

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

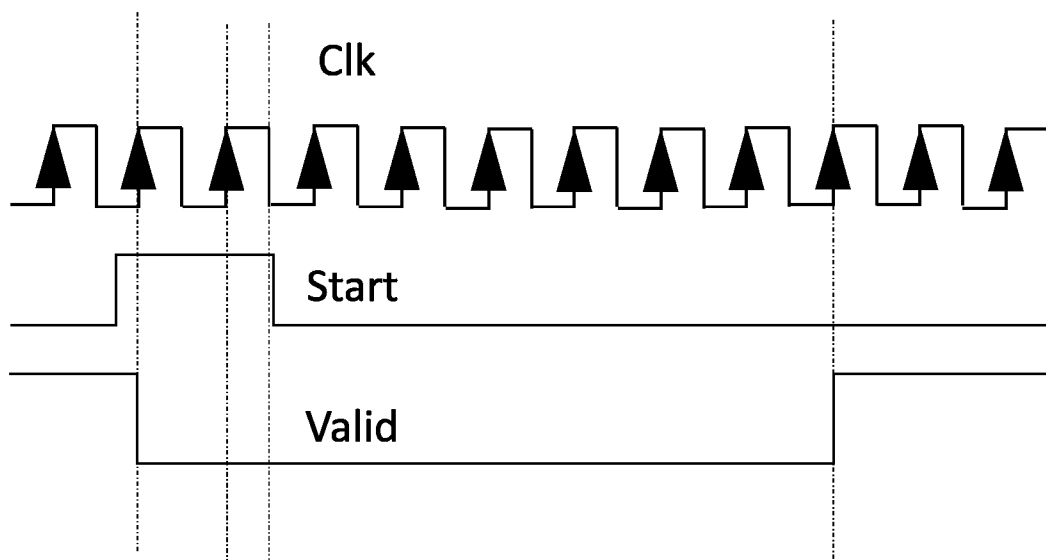
Pregunta 3 (3 puntos)

Diseñe usando VHDL un circuito secuencial conversor serie-a-paralelo de 8 bits que opera en el flanco de subida de la señal de reloj y tiene reset síncrono. El circuito tiene la siguiente **entity**:

```
entity conversor is
  port( DataOut:  out std_logic_vector( 7 downto 0);
        Valid: out std_logic;
        Clk: in std_logic;
        Reset: in std_logic;
        Start: in std_logic;
        DataIn: in std_logic );
end entity conversor;
```

A continuación se describe el funcionamiento del circuito conversor.

La señal de salida *Valid* se pone a '0' cuando la señal de entrada *Start* vale '1' en el flanco de subida de la señal de reloj. Si a partir de ese momento se producen 7 flancos de subida de la señal de reloj en los que la señal *Start* valga '0', la señal *Valid* pasa a tomar el valor '1' en el séptimo flanco de subida de la señal de reloj. El valor de la señal *Valid* se mantiene a '1' siempre que la señal *Start* valga '0' en los flancos de subida de la señal de reloj. En el siguiente cronograma pueden verse los valores que toma la señal de salida *Valid* en función de los valores que tienen la señal de reloj *Clk* y la señal de entrada *Start*.



Los bits recibidos por la entrada serie *DataIn* se almacenan en un registro interno del circuito llamado *Content* tal como se describe a continuación. En el flanco de subida de la señal de reloj, si la señal *Valid* vale '0' y la señal *Reset* vale '0', se introduce el valor de la señal de

entrada serie `DataIn` en la posición más significativa del registro `Content` desplazando el contenido del registro a la derecha.

El contenido del registro `Content` es reseteado al valor “00000000” en el flanco de subida de la señal de reloj cuando la señal `Reset` vale ‘1’.

Si la señal `Reset` vale ‘0’ y la señal `Valid` vale ‘1’, se mantiene el valor del registro `Content`.

El contenido del registro `Content` se carga en la salida paralelo `DataOut` siempre que la señal `Valid` tenga el valor ‘1’ y la señal `Reset` valga ‘0’. Si la señal `Valid` tiene el valor ‘0’ y la señal `Reset` vale ‘0’, la señal `DataOut` mantiene su valor. La salida `DataOut` se resetea al valor “00000000” cuando la señal `Reset` vale ‘1’.

El diseño del registro en VHDL debe realizarse describiendo el comportamiento del circuito.

Pregunta 4 (2 puntos)

Programa en VHDL un banco de pruebas para el circuito que ha diseñado al contestar a la Pregunta 3. La señal de reloj (`Clk`) debe tener un periodo de 20 ns e inicialmente valer ‘0’. El primer flanco de subida de la señal de reloj se ha de producir en el instante 10 ns. El programa de test debe realizar consecutivamente las acciones siguientes:

1. *Reset.* La señal de reset ha de tener el valor ‘1’ durante los primeros 15 ns.
2. *Cargar en la señal de salida paralelo `DataOut` el valor “11111100”.* La señal `Start` ha de tener el valor ‘1’ únicamente entre los instantes 40 ns y 60 ns, debiendo tener el valor ‘0’ el resto del tiempo. La señal de entrada `DataIn` ha de valer inicialmente ‘0’ y cambiar al valor ‘1’ en el instante 80 ns. En consecuencia, la señal `DataIn` toma en ciclos de reloj consecutivos los valores ‘0’, ‘0’, ‘1’, ‘1’, ‘1’, ‘1’, ‘1’ y ‘1’. Comprobar que las señales de salida `DataOut` y `Valid` toman los valores correctos. En caso contrario, el banco de pruebas debe mostrar un mensaje indicándolo.