INGENIERÍA DE COMPUTADORES 3

Solución al examen de Junio 2022, Segunda Semana

PREGUNTA 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales `s1`, `s2`, `s3` y `s4` entre los instantes 0 y 50 ns. En el cronograma se debe indicar para cada una de estas señales el instante de tiempo en que la señal cambia de valor así como su nuevo valor.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity crono2 is
end entity crono2;
architecture crono2 of crono2 is
  signal x1, x2, x3, s1, s2, s3, s4 : std_logic;
begin
  x1 <= '1', '0' after 5 ns, '1' after 15 ns, '0' after 30 ns,
        '1' after 35 ns, '0' after 50 ns;
  x2 <= '0', '1' after 15 ns, '0' after 35 ns;
  x3 <= x1 after 10 ns;
  Proc1: process
     variable valor  : std_logic;
  begin
    for i in 0 to 4 loop
       valor := x1 and x2;
       s1 <= valor;
       s2 <= s1;
       s3 <= x1 and x2;
       s4 <= s2 or s3;
       wait for 10 ns;
     end loop;
     wait;
  end process;
end architecture crono2;
```

Solución a la Pregunta 1

En la Figura 1.1 se muestra el cronograma de evolución de las señales.

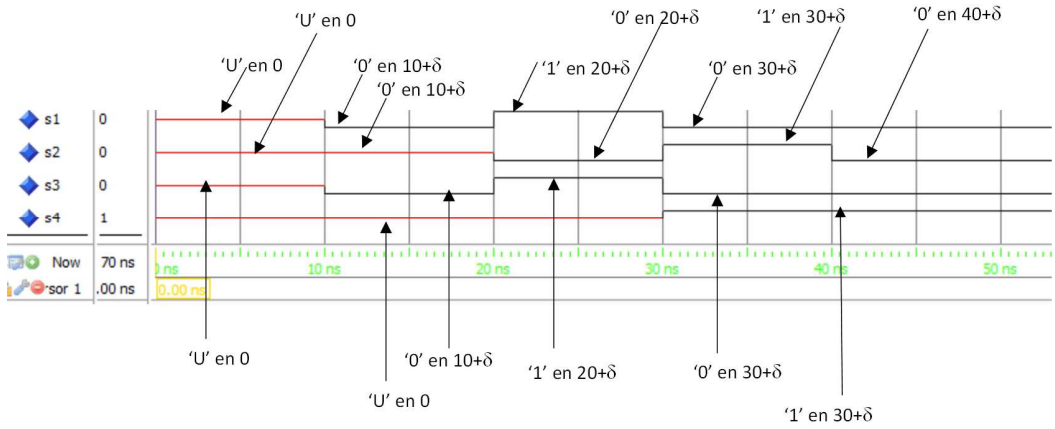

Figura 1.1: Cronograma de evolución de las señales.

PREGUNTA 2 (3 puntos)

Se pretende diseñar un circuito comparador que tiene una señal de salida de un bit llamada `F` y dos señales de entrada: la señal de un bit `E` y la señal de 4 bits `x`. La **entity** del circuito se muestra a continuación.

```
entity comparaXmayor6 is port
 ( F : out std_logic;
   E:  in std_logic;
   x : in  std_logic_vector(3 downto 0) );
end entity comparaXmayor6;
```

El circuito tiene el siguiente comportamiento.

Los cuatro bits de la señal de entrada `x` se interpretan como un número binario sin signo.

La señal de salida `F` tiene valor '1' si y sólo si la señal `E` tiene valor '1' y, además, el valor de `x` es mayor que el número decimal 6.

En cualquier otro caso la señal `F` tiene valor '0'. En particular, si la señal `E` tiene valor '1' y, además, el valor de `x` es menor o igual que el número decimal 6, el valor

de la señal de salida F es '0'. Y si la señal de entrada E tiene valor '0', entonces la señal de salida F tiene valor '0'.

2.a) (1.5 puntos) Escriba la tabla de verdad de la salida (F) en función de las entradas (X y E). Escriba la función lógica (F) en función de X y E obtenida a partir de dicha tabla de verdad. Escriba en VHDL la **architecture** del circuito comparador empleando únicamente sentencias de asignación concurrente y operadores lógicos.

2.b) (0.5 puntos) Dibuje un diagrama circuital del circuito comparador implementado con puertas lógicas. Escriba en VHDL la **entity** y **architecture** de las puertas lógicas que ha incluido en el diagrama circuital.

2.c) (1 puntos) Escriba en VHDL una **architecture** que describa la *estructura* del circuito que ha dibujado en el anterior apartado, instanciando y conectando las puertas lógicas que ha diseñado anteriormente.

Solución a la Pregunta 2

El código VHDL solución del apartado 2.a se muestra en Codigo VHDL 1.1.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
architecture comportamiento OF comparaXmayor6 is
begin
   F <= ((x(2) and x(1) and x(0)) or x(3)) and E;
end comportamiento;
```

Código VHDL 1.1: Architecture del circuito descrito en la Pregunta 2.a.

En la Figura 1.2 se muestra el diagrama circuital del comparador. En este diagrama se emplean una puerta AND de 2 entradas, una puerta AND de 3 entradas y una puerta OR de dos entradas. El código VHDL de estas puertas se muestra respectivamente en Código VHDL 1.2, 1.3 y 1.4.

El código VHDL solución del apartado 2.c se muestra en Codigo VHDL 1.5.

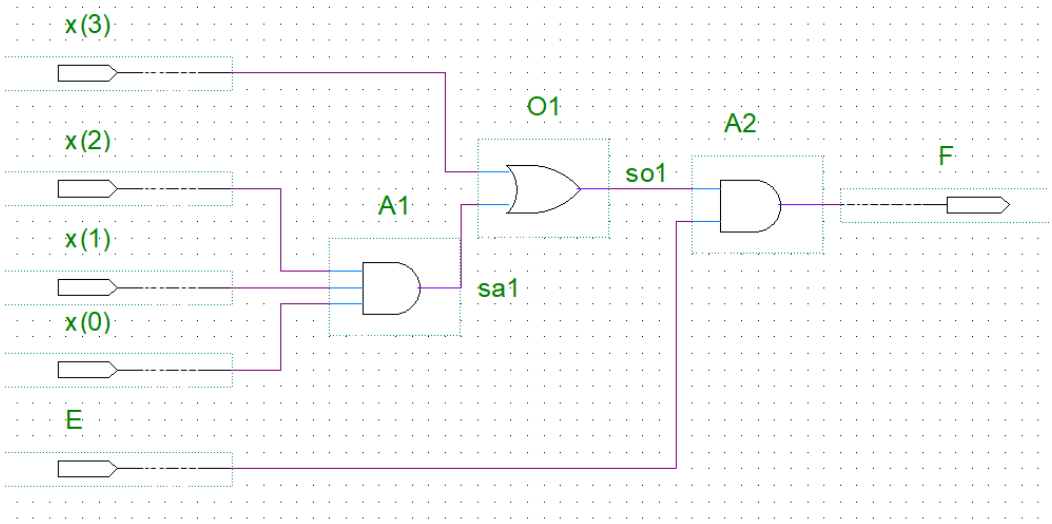

Figura 1.2: Diagrama circuital del comparador.

```
----------------------------------
-- AND de 2 entradas
library IEEE; use IEEE.std_logic_1164.all;

entity and2 is
   port ( y0        : out std_logic;
          x0, x1 : in    std_logic );
end entity and2;

architecture and2 of and2 is
begin
   y0 <= ( x0 and x1 );
end architecture and2;
----------------------------------
```

Código VHDL 1.2: Puerta lógica AND de dos entradas.

```
----------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity and3 is
   port ( y0 : out std_logic;
        x0, x1, x2 : in std_logic );
end entity and3;

architecture and3 of and3 is
begin
   y0 <= x0 and x1 and x2;
end architecture and3;
----------------------------------
```

Código VHDL 1.3: Puerta lógica AND de tres entradas.

```
-------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;

entity or2 is
   port ( y0 : out std_logic;
        x0, x1 : in std_logic );
end entity or2;

architecture or2 of or2 is
begin
   y0 <= x0 or x1;
end architecture or2;
-------------------------------------
```

Código VHDL 1.4: Puerta lógica OR de dos entradas.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
architecture Estructura OF comparaXmayor6 is
   signal sa1, so1: std_logic;
-- Declaración de las clases de los componentes
   component and3 is
   port (          y0 : out std_logic ;
           x0, x1, x2 : in std_logic);
   end component and3;
   component and2 is
   port (    y0 : out std_logic ;
           x0, x1 : in std_logic);
   end component and2;
   component or2 is
      port ( y0 : out std_logic;
           x0, x1 : in std_logic );
   end component or2;
begin
  -- Instanciación y conexión de los componentes

   A1 : component and3 port map (sa1, x(2), x(1), x(0));
   O1 : component or2 port map (so1, sa1, x(3) );
   A2 : component and2 port map (F, so1, E);
end Estructura;
```

Código VHDL 1.5: Architecture del comparador describiendo su estructura.

PREGUNTA 3 (2 puntos)

Programe el banco de pruebas del circuito combinacional que ha diseñado en la Pregunta 2.a. 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.

Solución a la Pregunta 3

El código VHDL del banco de pruebas se muestra en Código VHDL 1.6.

```
-------------------------------------
-- Banco de pruebas comparador X>6
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
entity bp_comparador is
end entity bp_comparador;

architecture bp_comparador of bp_comparador is
   signal F : std_logic;    -- Conectar salidas UUT
   signal x : std_logic_vector(3 downto 0);    -- Conectar entradas UUT
   signal E : std_logic;    -- Conectar entradas UUT
   component comparaXmayor6 is port
    ( F : out std_logic;
      E:   in std_logic;
      x : in   std_logic_vector(3 downto 0) );
   end component comparaXmayor6;
begin
   -- Instanciar y conectar UUT
   uut : component comparaXmayor6 port map
         ( F => F, E=>E, x => x );
   gen_vec_test : process
      variable test_in      : unsigned (3 downto 0):= B"0000"; -- Vector de test
      variable num_errores   : integer := 0;              -- Numero de errores
      variable testE : unsigned(0 downto 0);
   begin
      for countE in 1 downto 0 loop
         testE := to_unsigned(countE,1);
         E <= std_logic(testE(0));
         for count in 0 to 2**4-1 loop
            x(3)  <= test_in(3);
            x(2)  <= test_in(2);
            x(1)  <= test_in(1);
            x(0)  <= test_in(0);
            wait for 10 ns;
          -- Comprueba resultado
          if (F='0' and test_in>6 and E = '1') or
              (F='1' and test_in<=6) or (F='1' and E='0') then
                 report("Error. Valor "&integer'image(to_integer(test_in)));
                 num_errores := num_errores + 1;
           end if;
           test_in := test_in + 1;
         end loop;
      end loop;
      report "Test completo. Hay "       &
          integer'image(num_errores) &
          " errores.";
      wait; --Final simulación
   end process gen_vec_test;
end architecture bp_comparador;
-------------------------------------
```

Código VHDL 1.6: Diseño del banco de pruebas del comparador.

PREGUNTA 4 (3 puntos)

Programe en VHDL un circuito contador módulo 10 como una máquina de estado tipo Moore sensible al flanco de subida de la señal de reloj. En el diseño únicamente pueden emplearse los dos siguientes paquetes de la librería IEEE:

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

La **entity** del circuito contador se muestra a continuación:

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

Las señales de entrada del circuito son las siguientes:

- Una señal `reset` asíncrona activa a nivel alto. Cuando esta señal está activa, la cuenta se pone a cero.
- La señal `en` permite habilitar o deshabilitar la cuenta. Si el valor de la señal `en` es '1' la cuenta está habilitada. En caso contrario, se para la cuenta.
- La señal de reloj `clk`.

Las señales de salida del circuito son la siguientes:

- La señal `q` muestra el valor de la cuenta en número binario. En consecuencia, esta señal toma cíclicamente los valores "0000", "0001", "0010", "0011", "0100", "0101", "0110", "0111", "1000", "1001".
- La señal `pulse` toma el valor '1' sólo cuando el valor de la cuenta es "1001".

Solución a la Pregunta 4

El código VHDL correspondiente a la **architecture** del circuito contador se muestra en Código VHDL 1.7.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
architecture dec_counter of dec_counter is
   signal r_reg: unsigned(3 downto 0);
   signal r_rext: unsigned(3 downto 0);
   constant TEN: integer := 10;
begin
   process(clk, reset)
   begin
       if (reset = '1') then
            r_reg <= (others=>'0');
       elsif (rising_edge(clk)) then
            if (en = '1') then
                 if r_reg = (TEN-1) then
                     r_reg <= (others=>'0');
                  else
                     r_reg <= r_reg+1;
                  end if;
            end if;
       end if;
   end process;
   q <= std_logic_vector(r_reg);
   pulse <= '1' when r_reg=(TEN-1) else '0';
end dec_counter;
```

Código VHDL 1.7: Circuito contador módulo 10.