INGENIERÍA DE COMPUTADORES 3

Solución al examen de Junio 2019, Segunda Semana

PREGUNTA 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales `x1`, `x2`, `x3`, `x4` y `x5` entre los instantes 0 y 120 ns.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity cronoW is
end entity cronoW;
architecture cronoW of cronoW is
  signal x1, x2, x3, x4, x5 : std_logic;
begin
  x1 <= '1', '0' after 10 ns,
        '1' after 20 ns,'0' after 25 ns,
        '1' after 40 ns;
  Proc1: process
  begin
    x2 <= '1';
    wait for 10 ns;
    x2 <= '0';
    wait for 15 ns;
    x2 <= '1';
    wait for 20 ns;
    x2 <= '0';
  end process;
   x3 <= x1 after 10 ns;
  Proc2: process
     variable valor  : std_logic;
  begin
    for i in 0 to 3 loop
       valor := x1 xor x2;
       x4 <= valor;
       x5 <= x4;
       wait for 10 ns;
     end loop;
     wait;
  end process;
end architecture cronoW;
```

Solución a la Pregunta 1

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

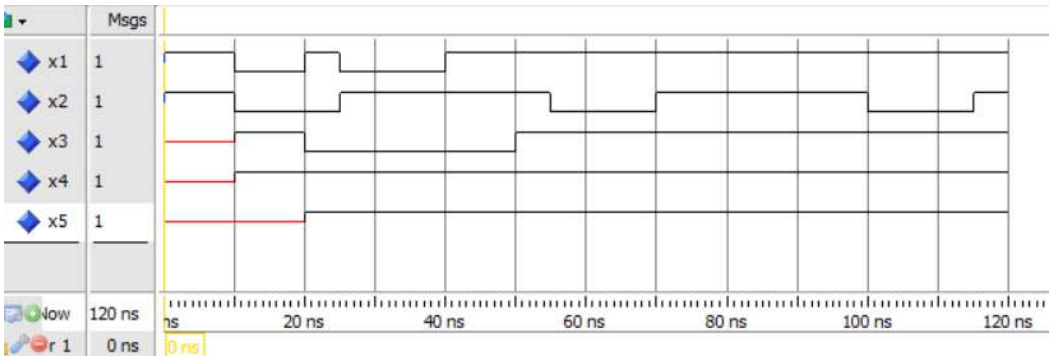


Figura 1.1: Cronograma de evolución de las señales.

PREGUNTA 2 (3 puntos)

Escriba en VHDL, de las cuatro formas que se detallan a continuación, la **architecture** que describe el comportamiento de un circuito combinacional decodificador 3 a 8 con entrada enable activa a nivel alto (señal `En`). La **entity** del circuito es:

```
entity decodificador is
   port ( Yout : out std_logic_vector (7 downto 0);
          Ain : in std_logic_vector (2 downto 0);
          En: in std_logic);
end entity decodificador;
```

2.a) (0.75 puntos) Empleando una sentencia concurrente condicional (**when - else**).

2.b) (0.75 puntos) Empleando una asignación concurrente de selección (**with - select**).

2.c) (0.75 puntos) Empleando un bloque **process** con una sentencia **if**.

2.d) (0.75 puntos) Empleando un bloque **process** con una sentencia **case**.

En aquellos casos en que sea necesario, se podrán emplear otras sentencias a parte de las especificadas.

Solución a la Pregunta 2

El código VHDL del decodificador descrito según las especificaciones de los apartados 2.a, 2.b, 2.c y 2.d se muestra, respectivamente, en Código VHDL 1.1–1.4.

```
architecture decod_when of decodificador is
begin
 Yout <="00000000" when En='0' else
        "00000001" when En='1' and Ain="000" else
        "00000010" when En='1' and Ain="001" else
        "00000100" when En='1' and Ain="010" else
        "00001000" when En='1' and Ain="011" else
        "00010000" when En='1' and Ain="100" else
        "00100000" when En='1' and Ain="101" else
        "01000000" when En='1' and Ain="110" else
        "10000000" when En='1' and Ain="111";
 end decod_when;
```

Código VHDL 1.1: Diseño del decodificador 3 a 8 empleando una sentencia concurrente condicional.

```
architecture decod_with of decodificador is
    signal   temp: std_logic_vector(3 downto 0);
begin
   temp <= En&Ain;
   with (temp) select
     Yout <=
             "00000001" when "1000",
             "00000010" when "1001",
             "00000100" when "1010",
             "00001000" when "1011",
             "00010000" when "1100",
             "00100000" when "1101",
             "01000000" when "1110",
             "10000000" when "1111",
              "00000000" when others;
 end decod_with;
```

Código VHDL 1.2: Diseño del decodificador 3 a 8 empleando una sentencia concurrente de selección.

```
architecture decod_if of decodificador is
begin
   process (Ain, En)
   begin
      if (En='0') then
          Yout <= (others => '0');
      elsif (Ain = "000") then
          Yout <= "00000001";
      elsif (Ain = "001") then
          Yout <= "00000010";
      elsif (Ain = "010") then
          Yout <= "00000100";
      elsif (Ain = "011") then
          Yout <= "00001000";
      elsif (Ain = "100") then
          Yout <= "00010000";
      elsif (Ain = "101") then
          Yout <= "00100000";
      elsif (Ain = "110") then
          Yout <= "01000000";
      elsif (Ain = "111") then
          Yout <= "10000000";
     else
          Yout <= "00000000";
     end if;
   end process;
end decod_if;
```

Código VHDL 1.3: Diseño del decodificador 3 a 8 empleando un bloque **process** con una sentencia **if**.

```
architecture decod_case of decodificador is
begin
   process (Ain, En)
   begin
      if (En='0') then
          Yout <= (others => '0');
      else
          case Ain is
             when "000" => Yout <= "00000001";
             when "001" => Yout <= "00000010";
             when "010" => Yout <= "00000100";
             when "011" => Yout <= "00001000";
             when "100" => Yout <= "00010000";
             when "101" => Yout <= "00100000";
             when "110" => Yout <= "01000000";
             when "111" => Yout <= "10000000";
             when others => Yout <= "00000000";
          end case;
      end if;
   end process;
end decod_case;
```

Código VHDL 1.4: Diseño del decodificador 3 a 8 empleando un bloque **process** con una sentencia **case**.

PREGUNTA 3 (3 puntos)

Diseñe usando VHDL el siguiente circuito secuencial síncrono. El circuito opera en el flanco de subida de la señal de reloj (`clk`) y tiene como entradas la señal de reloj `clk`, una señal `reset` asíncrona activa a nivel alto y una señal de un bit `X`. El circuito tiene una señal de salida de un bit `Z`.

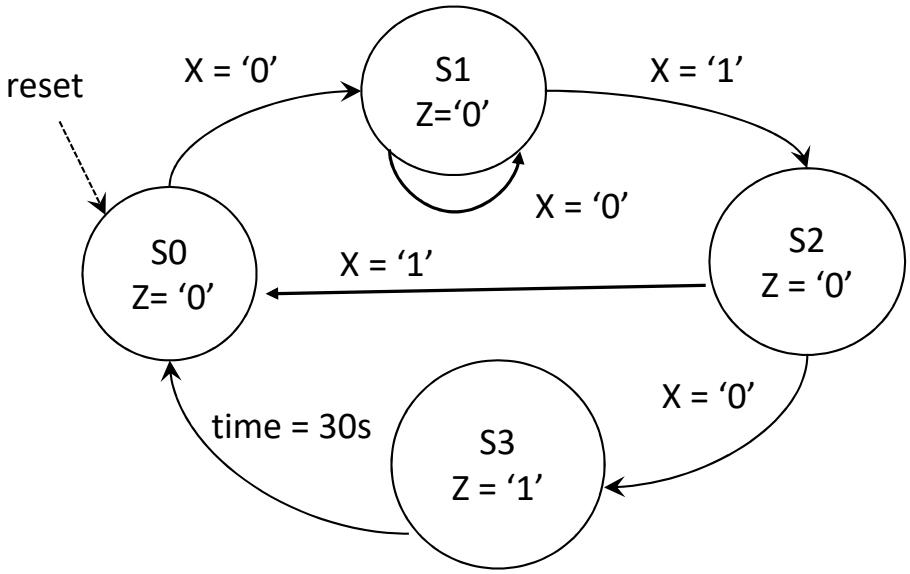
El circuito realiza cíclicamente los siguientes pasos:

1. No le ha llegado ningún bit al circuito. La señal de salida `Z` tiene el valor '0'.
2. El circuito recibe bits consecutivamente, y trabaja para reconocer la secuencia no solapada "010"de los tres últimos bits de la entrada `X`. La señal de salida `Z` mantiene el valor '0'. En caso de reconocer la secuencia, se pasa al paso tres, donde permanece 30 segundos.
3. La señal de salida `Z` tiene el valor '1'. No se reconoce ninguna secuencia de entrada.

La señal `reset` inicializa el circuito poniendo el circuito en el paso 1. Diseñe el circuito como una máquina de Moore. Dibuje un diagrama con los estados del circuito. Escriba el código VHDL de la **entity** y **architecture** que describe el comportamiento del circuito siguiendo el diagrama de estados descrito anteriormente.

Solución a la Pregunta 3

El diagrama de estados correspondiente al circuito detector se muestra en la siguiente figura.



El código VHDL correspondiente al circuito se muestra en Código VHDL 1.5–1.6. Se ha realizado la suposición de que la señal de reloj es de 1 Hz.

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity circSec is
    port (   z : out std_logic;
             x, clk, reset    : in   std_logic );
end circSec;
architecture circSec of circSec is
     signal   internal_state: std_logic_vector(1 downto 0);
     constant timeS3   : integer := 29;
begin
proximo_estado: process (clk, reset)
    variable count: integer range 0 to timeS3;
begin
    if (reset = '1') then
         internal_state <= "00";--INICIAL
         count := 0;
    elsif (rising_edge(clk)) then
         case internal_state is
              when "00" => --INICIAL
                if (x='0') then
                   internal_state <= "01";
                   count := 0;
                end if;
              when "01" =>   --RECONOCIDO 0
                if (x='1') then
                   internal_state <= "10";
                end if;
                count := 0;
              when "10" =>   -- RECONOCIDO 01
                if (x='0') then
                   internal_state <= "11";
                else
                   Internal_state <="00";
                end if;
                count := 0;
              when others => --RECONOCIDO 010
                if (count >= timeS3) then
                    internal_state <= "00";
                    count := 0;
                else
                     count := count + 1;
                end if;
        end case;
    end if;
end process proximo_estado;
```

Código VHDL 1.5: Circuito solución de la Pregunta 3 (1/2).

```
salidas: process(internal_state)
begin
     case internal_state is
          when "11" =>
               z <= '1';
          when others =>
               z <='0';
     end case;
end process salidas;
end architecture circSec;
```

Código VHDL 1.6: Circuito solución de la Pregunta 3 (2/2).

PREGUNTA 4 (2 puntos)

Programe en VHDL el banco de pruebas del decodificador que ha diseñado al resolver la Pregunta 2. El banco de pruebas debe comprobar el funcionamiento del circuito de forma exhaustiva. El banco de pruebas debe generar un conjunto de vectores de test, comprobar si la salida de la UUT es correcta, mostrar un mensaje cada vez que la salida de la UUT no sea correcta y mostrar un mensaje al finalizar el test en el que se indique el número total de salidas incorrectas.

Solución a la Pregunta 4

El código VHDL del banco de pruebas se muestra en Código VHDL 1.7–1.8.

```
--------------------------------------------------
-- Banco de pruebas del decodificador 8:3
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
entity bp_Decodificador is
    constant DELAY     : time      := 10 ns; -- Retardo usado en el test
end entity bp_Decodificador;
architecture bp_Decodificador of bp_Decodificador is
    signal Yout   : std_logic_vector(7 downto 0);
    signal Ain    : std_logic_vector(2 downto 0);
    signal En     : std_logic;
    component decodificador is
        port (   Yout : out std_logic_vector (7 downto 0);
                 Ain : in std_logic_vector (2 downto 0);
                  En : in std_logic);
    end component decodificador;
    procedure comprueba_salidas
        ( actual_yout            :       in std_logic_vector (7 downto 0);
          Ain                    :       in     std_logic_vector (2 downto 0);
          En                     :       in     std_logic;
          error_count            : inout integer) is
        variable esperado_yout: std_logic_vector(7 downto 0);
    begin
     if (En='0') then
         esperado_yout := (others => '0');
     else
         case Ain is
            when "000" => esperado_yout := "00000001";
            when "001" => esperado_yout := "00000010";
            when "010" => esperado_yout := "00000100";
            when "011" => esperado_yout := "00001000";
            when "100" => esperado_yout := "00010000";
            when "101" => esperado_yout := "00100000";
            when "110" => esperado_yout := "01000000";
            when "111" => esperado_yout := "10000000";
            when others => esperado_yout := "00000000";
         end case;
     end if;
         if ( esperado_yout /= actual_yout ) then
             report "ERROR: Estado esperado ("      &
                       std_logic'image(esperado_yout(7))     &
                       std_logic'image(esperado_yout(6))     &
                      std_logic'image(esperado_yout(5))     &
                      std_logic'image(esperado_yout(4))     &
                      std_logic'image(esperado_yout(3))     &
                      std_logic'image(esperado_yout(2))     &
                      std_logic'image(esperado_yout(1))     &
                      std_logic'image(esperado_yout(0))     &
                      "), estado actual ("           &
                      std_logic'image(actual_yout(7))     &
                       std_logic'image(actual_yout(6))     &
                      std_logic'image(actual_yout(5))     &
                      std_logic'image(actual_yout(4))     &
                      std_logic'image(actual_yout(3))     &
                     std_logic'image(actual_yout(2))     &
                      std_logic'image(actual_yout(1))     &
                      std_logic'image(actual_yout(0))     &
                      "), instante: "                &
                      time'image(now);
             error_count := error_count + 1;
         end if;
     end procedure comprueba_salidas;
```

Código VHDL 1.7: Diseño del banco de pruebas del decodificador 3 a 8 (1/2).

```
begin
   UUT : component decodificador port map
          (Yout, Ain, En);
vec_test : process is
    variable temp  : unsigned (3 downto 0);
    variable error_count : integer := 0; -- Núm. errores
   begin
      report "Comienza la simulación";
      -- Generar todos los posibles valores de entrada
      for i in 0 to 15 loop
         temp := TO_UNSIGNED(i,4);
         En <= std_logic(temp(3));
         Ain(2) <= std_logic(temp(2));
         Ain(1) <= std_logic(temp(1));
         Ain(0) <= std_logic(temp(0));
         wait for DELAY;
         comprueba_salidas(Yout, Ain, En, error_count);
       end loop;
      report "Simulación finalizada";
      wait;      -- Final de la simulación
   end process vec_test;
end architecture bp_Decodificador;
```

Código VHDL 1.8: Continuación del banco de pruebas del decodificador 3 a 8 (2/2).