

INGENIERÍA DE COMPUTADORES 3

Solución al examen de Junio 2026, Primera Semana

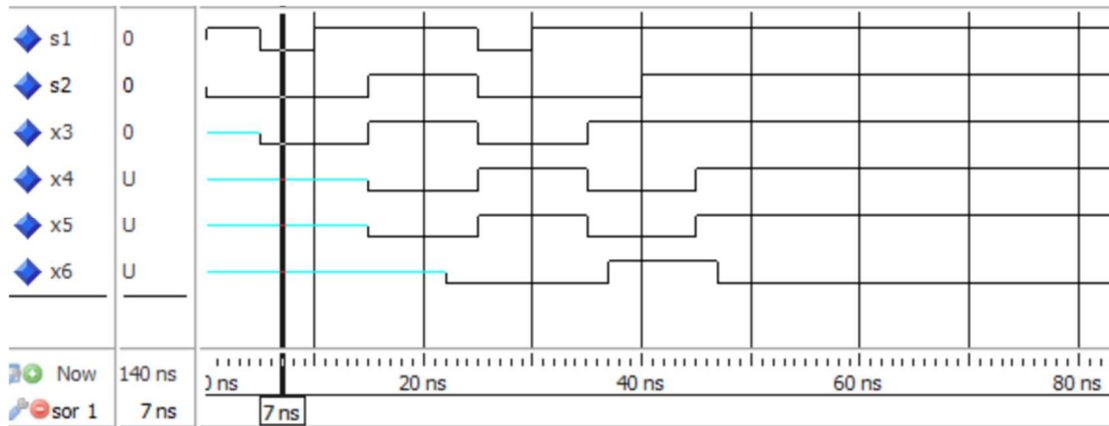
PREGUNTA 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales x3, x4, x5 y x6 entre los instantes 0 y 80 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 cronol is
end entity cronol;
architecture cronol of cronol is
    signal s1, s2, x3, x4, x5, x6 : std_logic;
begin
    s1 <= '1', '0' after 5 ns, '1' after 10 ns,
          '0' after 25 ns, '1' after 30 ns;
    s2 <= '0', '1' after 15 ns, '0' after 25 ns,
          '1' after 40 ns;
    Proc1: process
        variable valor : std_logic;
    begin
        wait for 5 ns;
        for i in 0 to 6 loop
            x3 <= s1 or s2;
            x5 <= x3;
            wait for 10 ns;
            x4 <= x3;
        end loop;
    end process;
    x6 <= s1 xor s2 after 7 ns;
end architecture cronol;
```

Solución a la Pregunta 1

En la siguiente figura se muestra el cronograma de evolución de las señales.

**PREGUNTA 2** (2.5 puntos)

Escriba en VHDL la **entity** y la **architecture** que describe:

- 2.a)** (0.25 puntos) El comportamiento de una puerta NOT de 1 entrada.
- 2.b)** (0.25 puntos) El comportamiento de una puerta AND de 4 entradas.
- 2.c)** (2 puntos) La estructura de un circuito combinacional decodificador 3 a 8 con entrada enable activa a nivel alto. La **architecture** debe describir la estructura del circuito combinacional, instanciando y conectando adecuadamente las puertas lógicas cuyo diseño ha realizado al contestar los dos apartados anteriores.

Solución a la Pregunta 2

El Código VHDL 1.1 y 1.2 muestran la **architecture** de la puerta NOT y AND respectivamente.

```

1 library IEEE; use IEEE.std_logic_1164.all;
2
3 entity not1 is
4   port ( y      : out std_logic;
5         x      : in  std_logic );
6 end entity not1;
7
8 architecture not1 of not1 is
9 begin
10  y <= not x;
11 end architecture not1;

```

Código VHDL 1.1: Puerta NOT.

```

1 library IEEE; use IEEE.std_logic_1164.all;
2
3 entity and4 is
4   port ( y      : out std_logic;
5         x0, x1, x2, x3 : in  std_logic );
6 end entity and4;
7
8 architecture and4 of and4 is
9 begin
10  y <= x0 and x1 and x2 and x3;
11 end architecture and4;

```

Código VHDL 1.2: Puerta AND de 4 entradas.

El Código VHDL 1.3 muestra la **architecture** del circuito decodificador de 3 a 8.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity decodificador_3_a_8 is port
5   ( Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0 : out std_logic;
6     E, A0, A1, A2                  : in  std_logic);
7 end entity decodificador_3_a_8;
8
9 architecture decodificador_3_a_8 of decodificador_3_a_8 is
10  signal not_A2, not_A1, not_A0 : std_logic;
11  -- Declaración de las clases de los componentes
12  component not1 is port
13    ( y      : out std_logic;
14      x      : in  std_logic );
15  end component not1;
16  component and4 is port
17    ( y      : out std_logic;
18      x0, x1, x2, x3 : in  std_logic );
19  end component and4;
20 begin
21  -- Instanciación y conexión de los componentes
22  NOT_0 : component not1 port map (not_A0, A0);
23  NOT_1 : component not1 port map (not_A1, A1);
24  NOT_2 : component not1 port map (not_A2, A2);
25  AND_7 : component and4 port map (Y7, E, A2, A1, A0);

```

```

26 AND_6 : component and4 port map (Y6, E, A2, A1, not_A0);
27 AND_5 : component and4 port map (Y5, E, A2, not_A1, A0);
28 AND_4 : component and4 port map (Y4, E, A2, not_A1, not_A0);
29 AND_3 : component and4 port map (Y3, E, not_A2, A1, A0);
30 AND_2 : component and4 port map (Y2, E, not_A2, A1, not_A0);
31 AND_1 : component and4 port map (Y1, E, not_A2, not_A1, A0);
32 AND_0 : component and4 port map (Y0, E, not_A2, not_A1, not_A0);
33 end architecture decodificador_3_a_8;

```

Código VHDL 1.3: Decodificador de 3 a 8.

PREGUNTA 3 (2.5 puntos)

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

Solución a la Pregunta 3

El código VHDL del banco de pruebas del circuito decodificador se muestra en Código VHDL 1.4.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.numeric_std.all;
4
5 entity bp_decodificador_3_a_8 is
6 end entity bp_decodificador_3_a_8;
7
8 architecture bp_decodificador_3_a_8 of bp_decodificador_3_a_8 is
9     signal Y
10         : std_logic_vector( 7 downto 0); -- Conectar
11         salidas UUT
12     signal E, A0, A1, A2 : std_logic; -- Conectar entradas UUT
13
14     component decodificador_3_a_8 is
15         port ( Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0 : out std_logic;
16             E, A0, A1, A2 : in std_logic);
17     end component decodificador_3_a_8;
18
19 begin
20     -- Instanciar y conectar UUT
21     uut : component decodificador_3_a_8
22         port map( Y(7), Y(6), Y(5), Y(4), Y(3), Y(2), Y(1), Y(0), E, A0,
23             A1, A2 );
24     gen_vec_test : process

```

```

22  variable test_in      : unsigned (3 downto 0); -- Vector de test
23  variable esperado_Y   : std_logic_vector(7 downto 0);
24  variable error_count  : integer    := 0;
25  begin
26  test_in := B"0000";
27  report "Comienza la simulación";
28  for count in 0 to 15 loop
29      E <= test_in(3);
30      A2 <= test_in(2);
31      A1 <= test_in(1);
32      A0 <= test_in(0);
33      if (count<8) then esperado_Y := "00000000";
34      elsif (count=8) then esperado_Y := "00000001";
35      elsif (count=9) then esperado_Y := "00000010";
36      elsif (count=10) then esperado_Y := "00000100";
37      elsif (count=11) then esperado_Y := "00001000";
38      elsif (count=12) then esperado_Y := "00010000";
39      elsif (count=13) then esperado_Y := "00100000";
40      elsif (count=14) then esperado_Y := "01000000";
41      else esperado_Y := "10000000";
42      end if;
43      wait for 10 ns;
44      test_in := test_in + 1;
45      if ( esperado_Y /= Y ) then
46          report "ERROR en la salida valida. Valor esperado: " &
47              std_logic'image(esperado_Y(7)) &
48              std_logic'image(esperado_Y(6)) &
49              std_logic'image(esperado_Y(5)) &
50              std_logic'image(esperado_Y(4)) &
51              std_logic'image(esperado_Y(3)) &
52              std_logic'image(esperado_Y(2)) &
53              std_logic'image(esperado_Y(1)) &
54              std_logic'image(esperado_Y(0)) &
55              ", valor actual: " &
56              std_logic'image(Y(7)) &
57              std_logic'image(Y(6)) &
58              std_logic'image(Y(5)) &
59              std_logic'image(Y(4)) &
60              std_logic'image(Y(3)) &
61              std_logic'image(Y(2)) &
62              std_logic'image(Y(1)) &
63              std_logic'image(Y(0)) &
64              " en el instante: " &
65              time'image(now);
66          error_count := error_count + 1;
67      end if;
68  end loop;
69  report "ERROR: Hay " &
70      integer'image(error_count) & " errores.";
71  wait;
72  end process gen_vec_test;
73 end architecture bp_decodificador_3_a_8;

```

Código VHDL 1.4: Banco de pruebas del circuito decodificador.

PREGUNTA 4 (3 puntos)

Escriba en VHDL la **architecture** que describe el comportamiento de un contador síncrono módulo 10 de 4 bits, con señal de reset síncrona activa a nivel bajo y que opera en el flanco de subida de la señal de reloj. El circuito tiene como entradas una señal de reloj (`clk`), una señal de 4 bits `D`, una señal de control `ctrl` de dos bits y una señal de reseteo llamada `reset`. El circuito tiene una señal de salida de 4 bits `Q`. Cuando la señal `reset` tiene el valor '0' en el flanco de subida de la señal de reloj pone la salida `Q` del circuito al valor "0000", con independencia del valor del resto de las señales de entrada del circuito. El circuito opera de la siguiente manera, según el valor de la señal `ctrl` (siempre que la señal de reset tenga valor '1'):

- Si `ctrl` tiene el valor "00", la señal de salida `Q` mantiene su valor.
- Si `ctrl` tiene el valor "01", la señal de salida `Q` toma el siguiente valor de la cuenta.
- Si `ctrl` tiene el valor "10", la señal de salida `Q` toma el valor de la señal de entrada `D`.
- Si `ctrl` tiene el valor "11", la señal de salida `Q` toma el valor "0000".

El código VHDL de la **entity** del contador módulo 10 se muestra a continuación.

```
entity contadorMod10 is
    port(
        clk : in  std_logic;
        reset : in  std_logic;
        ctrl : in std_logic_vector(1 downto 0);
        D : in std_logic_vector (3 downto 0);
        Q : out std_logic_vector (3 downto 0));
end contadorMod10;
```

Solución a la Pregunta 4

El Código VHDL 1.5 es el diseño del circuito secuencial solución de la Pregunta 4.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.numeric_std.all;
4
5 architecture contadorMod10 of contadorMod10 is
6     signal processQ: unsigned (3 downto 0);
7 begin
8     process (clk)
9     begin
10         if (rising_edge(clk)) then
11             if (reset = '0') then
12                 processQ <= (others => '0');
13             elsif ((processQ < 9) and (ctrl = "01")) then
14                 processQ <= processQ + 1;
15             elsif ((processQ = 9) and (ctrl = "01")) then
16                 processQ <= (others => '0');
17             elsif (ctrl = "10") then
18                 processQ <= unsigned(D);
19             elsif (ctrl = "11") then
20                 processQ <= (others => '0');
21             end if;
22         end if;
23     end process;
24     Q <= std_logic_vector(processQ);
25 end contadorMod10;

```

Código VHDL 1.5: Circuito contador.