

INGENIERÍA DE COMPUTADORES 3

Solución al examen de Junio 2025, Primera Semana

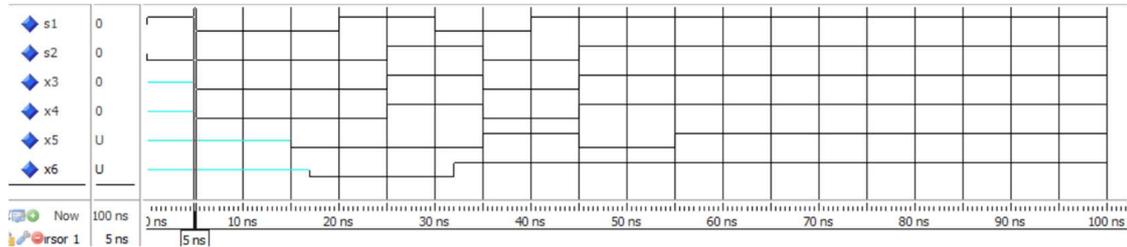
PREGUNTA 1 (2 puntos)

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 20 ns,
          '0' after 30 ns, '1' after 40 ns;
    s2 <= '0', '1' after 25 ns, '0' after 35 ns,
          '1' after 45 ns;
    Proc1: process
        variable valor : std_logic;
    begin
        wait for 5 ns;
        for i in 0 to 6 loop
            x3 <= s1 or s2;
            valor := s1 or s2;
            x4 <= valor;
            x5 <= x3;
            wait for 10 ns;
        end loop;
        wait;
    end process;
    x6 <= s1 or s2 after 12 ns;
end architecture cronol;
```

Solución a la Pregunta 1

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

**PREGUNTA 2** (3 puntos)

Escriba en VHDL la **architecture** que describe el comportamiento de un circuito contador binario de 4 bits que opera de modo síncrono en el flanco de subida de la señal de reloj (clk). El circuito tiene 5 señales de entrada de un bit y una señal de entrada d de 4 bits. El circuito tiene una señal de salida q de 4 bits. A continuación, se muestra la **entity** del circuito.

```
entity contador is
    port ( q : out std_logic_vector(3 downto 0);
          clk, reset : in std_logic;
          sync_clr, en, load : in std_logic;
          d : in std_logic_vector(3 downto 0) );
end entity contador;
```

El orden de prioridad de las señales de entrada del circuito es: reset–sync_clr–load–en, siendo la señal reset la más prioritaria y la señal en la menos prioritaria. Las señales reset, sync_clr, load y en están activas a nivel alto, por lo que producen cambios en la salida del circuito sólo cuando tienen valor '1'. Por ejemplo, para que la señal en produzca un cambio en el circuito tiene que tener un valor '1' y las señales de entrada reset, sync_clr y load tienen que tener valor '0'.

Cuando la señal reset está activa, se produce un reseteo asíncrono del circuito y todos los bits de la señal de salida q se ponen a cero. Cuando la señal sync_clr toma el valor '1' todos los bits de la señal de salida q se ponen a cero en el siguiente flanco de subida de la señal de reloj. Cuando la señal load toma el valor '1', se

carga en la señal de salida q el valor de la señal de entrada d . La señal en habilita la cuenta del circuito. Es decir, cuando tiene el valor '1' se incrementa el valor de la señal de salida q en uno cada ciclo de reloj. Sin embargo, cuando la señal en tiene el valor '0', la señal de salida q no cambia de valor.

Solución a la Pregunta 2

El Código VHDL 1.1 muestra la **architecture** del circuito contador.

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.numeric_std.ALL;
4 architecture contador of contador is
5     signal r_reg : unsigned(3 downto 0);
6 begin
7     process (clk, reset)
8     begin
9         if (reset='1') then
10            r_reg <= (others=>'0');
11        elsif (rising_edge(clk)) then
12            if (sync_clr = '1') then
13                r_reg <= (others=>'0');
14            elsif (load = '1') then
15                r_reg <= unsigned(d);
16            elsif (en = '1') then
17                r_reg <= r_reg +1;
18            else
19                r_reg <= r_reg;
20            end if;
21        end if;
22    end process;
23    q <= std_logic_vector(r_reg);
24 end contador;

```

Código VHDL 1.1: Architecture del circuito contador.

PREGUNTA 3 (2 puntos)

Escriba en VHDL un banco de pruebas que permita comprobar mediante inspección visual el comportamiento del circuito contador de la Pregunta 2. El banco de pruebas debe generar una señal de reloj de periodo 100 ns. Además, debe realizar consecutivamente los siguientes pasos:

- Primero, realiza un reseteo asíncrono del circuito entre los instantes 0 ns y 125 ns.
- Después, carga en la señal de salida el valor "0010".
- Finalmente, realiza la cuenta hasta el valor "0100" y mantiene este valor.

Dibuja el cronograma de las señales de entrada y salida del circuito que se generará tras simular el banco de pruebas que ha escrito.

Solución a la Pregunta 3

El código VHDL del banco de pruebas del circuito contador se muestra en Código VHDL 1.2.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.numeric_std.all;
4
5 entity bp_contador is
6     constant PERIODO: time := 100 ns;
7 end entity bp_contador;
8
9 architecture bp_contador of bp_contador is
10     signal reset : std_logic;
11     signal clk: std_logic := '0';
12     signal sync_clr, en, load: std_logic;
13     signal q, d : std_logic_vector(3 downto 0);
14     component contador is
15         port ( q
16             : out std_logic_vector(3 downto 0);
17             clk, reset
18             : in std_logic;
19             sync_clr, en, load
20             : in std_logic;
21             d
22             : in std_logic_vector(3 downto 0));
23     end component contador;
24
25 begin
26     UUT:component contador port map (q, clk, reset, sync_clr, en, load, d
27 );

```

```

24 reset <= '1',
25         '0' after (PERIODO+PERIODO/4);
26 clk <= not clk after (PERIODO/2);
27
28 main: process is
29 begin
30     d <= "0010";
31     load <= '1';
32     sync_clr <= '0';
33     en <= '0';
34     wait for 2*PERIODO+PERIODO/4;
35     load <= '0';
36     en <= '1';
37     wait for 2*PERIODO;
38     en <= '0';
39     wait;
40 end process main;
41 end architecture bp_contador;

```

Código VHDL 1.2: Banco de pruebas del circuito contador.

PREGUNTA 4 (3 puntos)

Escriba en VHDL la **architecture** que describe el comportamiento de un divisor de frecuencias por 6, con señal de reset asíncrona activa a nivel bajo.

Es decir, la señal de salida del circuito (`clk6`) tiene una frecuencia que es la sexta parte de la frecuencia de la señal de reloj de entrada (`clk`). Cuando la señal de `resetsn` tiene valor '0', la señal de salida `clk6` se pone a valor cero de forma asíncrona.

El código VHDL de la **entity** del divisor de frecuencias se muestra a continuación.

```

entity divisor_frecuencia_6 is port (
    clk6          : out std_logic;
    clk, resetsn : in  std_logic );
end entity divisor_frecuencia_6;

```

Solución a la Pregunta 4

El Código VHDL 1.3 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 divisor_frecuencia_6 of divisor_frecuencia_6 is
6     signal countr: integer := 1;
7     signal tmp: std_logic;
8 begin
9     process (clk, resetn)
10    begin
11        if resetn = '0' then
12            countr <= 1;
13            tmp <= '0';
14        elsif (rising_edge(clk)) then
15            countr <= countr+1;
16            if countr = 3 then
17                tmp <= not tmp;
18                countr <= 1;
19            end if;
20        end if;
21    end process;
22    clk6 <= tmp;
23 end;
```

Código VHDL 1.3: Circuito divisor de frecuencia.