

INGENIERÍA DE COMPUTADORES 3

Solución al examen de Junio 2023, Primera Semana

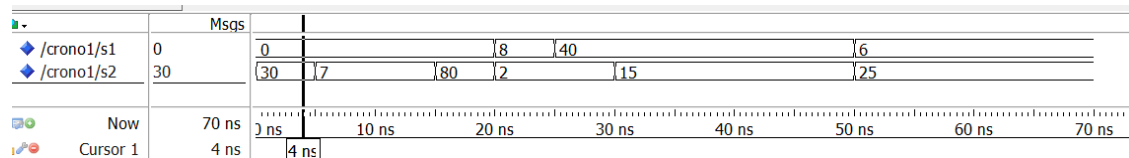
PREGUNTA 1 (2 puntos)

Tomando como base el siguiente código VHDL, dibuje el cronograma de evolución de las señales $s1$ y $s2$ entre los instantes 0 y 70 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, considerando el retardo delta.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity cronol is
end entity cronol;
architecture cronol of cronol is
    signal s1, s2 : natural;
begin
    process
    begin
        s1 <= 4 after 2 ns,
            1 after 10 ns,
            5 after 15 ns,
            10 after 24 ns,
            20 after 28 ns,
            50 after 70 ns;
        s1 <= 8 after 20 ns,
            40 after 25 ns,
            6 after 50 ns;
        s2 <= transport 30 after 0 ns,
            7 after 5 ns,
            80 after 15 ns,
            24 after 20 ns,
            20 after 25 ns,
            40 after 70 ns;
        s2 <= transport 2 after 20 ns,
            15 after 30 ns,
            25 after 50 ns;
        wait;
    end process;
end architecture cronol;
```

Solución a la Pregunta 1

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

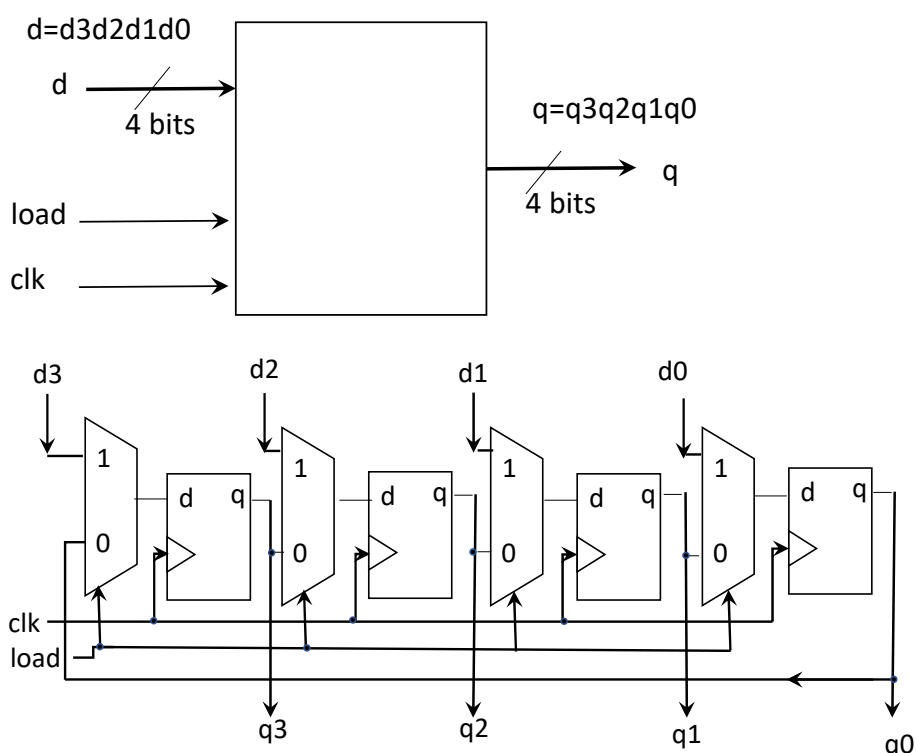


En la siguiente figura se muestra el valor de las señales teniendo en cuenta el retardo delta.

ps	delta	/crono1/s1	/crono1/s2
0	+0	0	0
0	+1	0	30
5000	+0	0	7
15000	+0	0	80
20000	+0	8	2
25000	+0	40	2
30000	+0	40	15
50000	+0	6	25

PREGUNTA 2 (3 puntos)

Se quiere programar usando VHDL un circuito secuencial que es un registro de desplazamiento circular que opera en el flanco de subida de la señal de reloj. En la parte superior de la figura se muestra el circuito con sus entradas y salidas. El diagrama de dicho circuito, compuesto empleando biestables tipo D y multiplexores 2 a 1, se muestra en la parte inferior de dicha figura.



Se observa que cuando la señal de un bit `load` vale '1', se carga en los biestables la señal de entrada `d`. Por el contrario, cuando la señal `load` vale '0', se carga en los biestables el resultado de realizar la rotación de la salida de los biestables una posición.

El circuito ha de tener la **entity** siguiente:

```
entity Desplazador is
  port( q      : out std_logic_vector (3 downto 0);
        d      : in  std_logic_vector (3 downto 0);
        clk    : in  std_logic;
        load   : in  std_logic );
end entity Desplazador;
```

- 2.a)** (0.75 puntos) Escriba en VHDL la **architecture** que describe el comportamiento del biestable D que opera en el flanco de subida de la señal de reloj empleando sólo un bloque **process** y una sentencia **if**. El biestable D ha de tener la **entity** siguiente:

```
entity flipflopD is
  port( q    : out std_logic;
        d    : in  std_logic;
        clk  : in  std_logic );
end entity flipflopD;
```

- 2.b)** (0.75 puntos) Escriba en VHDL la **architecture** de un multiplexor 2 a 1 de un bit que describa su comportamiento empleando una sentencia **with-select**. El multiplexor ha de tener la **entity** siguiente:

```
entity mux2a1 is
  port(   y      : out std_logic;
         sel     : in  std_logic;
         x1, x0 : in  std_logic);
end entity mux2a1;
```

- 2.c)** (1.5 puntos) Escriba en VHDL una **architecture** que describa la *estructura* del circuito desplazador circular, instanciando y conectando adecuadamente los multiplexores 2 a 1 y los biestables d que ha diseñado anteriormente. Emplee en el diseño la sentencia **generate** para realizar la instanciación y conexión de todos los componentes.

Solución a la Pregunta 2

El Código VHDL 1.1 es la solución del Apartado 2.a. La **architecture** del multiplexor 2 a 1 es el Código VHDL 1.2. El Código VHDL 1.3 es la solución del Apartado 2.c.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 architecture flipflopD of flipflopD is
4 begin
5     process (clk) is
6         begin
7             if rising_edge (clk) then
8                 q <= d;
9             end if;
10        end process;
11 end architecture flipflopD;

```

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

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 architecture arch_selec of mux2a1 is
4 begin
5     with sel select
6         y <= x0 when '0',
7         x1  when others;
8 end architecture arch_selec;

```

Código VHDL 1.2: Architecture del multiplexor 2 a 1.

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 architecture estructual of desplazador is
4     signal u,i: std_logic_vector(3 downto 0);
5     component mux2a1 is
6         port(
7             y: out std_logic;
8             sel: in std_logic;
9             x1, x0: in std_logic);
10    end component mux2a1;
11    component flipflopD is
12        port(
13            q: out std_logic;
14            d: in std_logic;
15            clk: in std_logic);
16    end component flipflopD;
17 begin
18     q <= u;
19     genConexion: for k in 3 downto 0 generate
20         inicial: if k = 3 generate
21             mux3: mux2a1 port map( i(3), load, d(3), u(0));
22         end generate inicial;
23         intermedio: if k/=3 generate
24             muxk: mux2a1 port map ( i(k), load, d(k), u(k+1));
25         end generate intermedio;
26         ffk: flipflopD port map (u(k), i(k), clk);
27     end generate genConexion;
28 end architecture;

```

Código VHDL 1.3: Architecture del circuito descrito en la Pregunta 2.c.

PREGUNTA 3 (2 puntos)

Programa el banco de pruebas del circuito secuencial que ha diseñado en la Pregunta 2.c. La señal de reloj debe tener un periodo de 10 ns. El banco de pruebas debe realizar por orden las siguientes comprobaciones:

- Primero, cargar la señal de entrada cuyo valor es $d = "0001"$
- Realizar la rotación durante 8 ciclos de reloj
- Finalmente, cargar la señal de entrada cuyo valor es $d = "0000"$

El banco de pruebas debe permitir comprobar mediante inspección visual que los valores obtenidos de la UUT coinciden con los esperados. Al final del test, debe mostrarse un mensaje indicando que el test ha finalizado. Dibuje el cronograma que se debería obtener al simular el banco de pruebas mostrando la evolución temporal de las señales del circuito bajo test.

Solución a la Pregunta 3

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

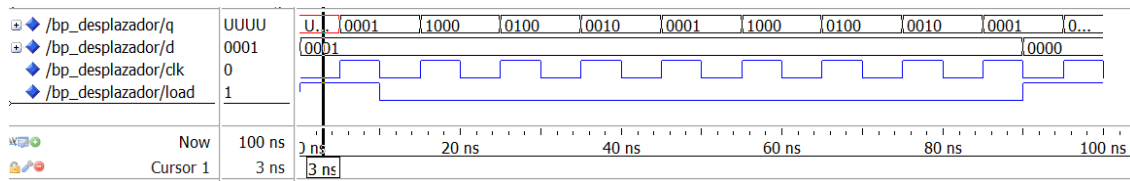
```

1  -- Banco de pruebas
2  -- fichero: bp_desplazador.vhd
3  library IEEE;
4  use IEEE.std_logic_1164.all;
5  use IEEE.numeric_std.all;
6
7  entity bp_desplazador is
8      constant DELAY      : time      := 10 ns;
9  end entity bp_desplazador;
10
11 architecture bp_desplazador of bp_desplazador is
12     signal q,d : std_logic_vector (3 downto 0);
13     signal clk  : std_logic:= '0';
14     signal load : std_logic;
15     component Desplazador is
16     port( q   : out std_logic_vector ( 3 downto 0);
17          d   : in  std_logic_vector (3 downto 0);
18          clk : in  std_logic;
19          load: in  std_logic);
20     end component Desplazador;
21 begin
22     UUT : component Desplazador port map (q, d, clk, load);
23     clk <= not clk after DELAY/2;
24     -- bloque process para generar los vectores de test
25     main : process is
26         variable error_count : integer := 0;
27     begin
28         report "Comienza la simulación";
29         d<="0001";
30         load <= '1';
31         wait for DELAY;
32         load <= '0';
33         for i in 0 to 7 loop
34             wait for DELAY;
35         end loop;
36         d<="0000";
37         load <= '1';
38         wait for DELAY;
39         report "Finaliza la simulación: ";
40         wait; -- Termina la simulación
41     end process main;
42 end architecture bp_desplazador;

```

Código VHDL 1.4: Banco de pruebas del desplazador.

En la siguiente figura se muestra el cronograma obtenido al simular el banco de pruebas durante 100 ns.



PREGUNTA 4 (3 puntos)

Escriba en VHDL la **architecture** de un circuito secuencial síncrono que opera en el flanco de subida de la señal de reloj. Se supone que la señal de reloj de entrada tiene un periodo de 1 segundo. Este circuito tiene dos señales de entrada de 1 bit: la señal de reloj `clk` y la señal de reset síncrona activa a nivel 0 `rst`. El circuito tiene una señal de salida de un bit llamada `salida`.

El circuito tiene los dos estados siguientes:

- Estado A: la señal `salida` tiene valor '0'. Se permanece en este estado 8 segundos siempre que no se resetee el circuito. Pasados los 8 segundos, se pasa al estado B. Cuando la señal `rst` se pone a '0', se vuelve a iniciar la cuenta de 8 segundos en el primer flanco de subida de la señal de reloj.
- Estado B: la señal `salida` tiene el valor '1'. Se permanece en este estado 4 segundos siempre que no se resetee el circuito. Pasados los 4 segundos, se pasa al estado A. Cuando la señal `rst` se pone a '0', el circuito pasa al estado A en el primer flanco de subida de la señal de reloj.

La **entity** del circuito se muestra a continuación.

```
entity control is
  port ( salida    : out std_logic;
        clk, rst  : in  std_logic );
end entity control;
```

Para el diseño del circuito emplee únicamente las siguientes librerías:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.ALL;
```


Solución a la Pregunta 4

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

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.numeric_std.ALL;
4 architecture control of control is
5     constant A: std_logic := '0';
6     constant B: std_logic := '1';
7     signal internal_state: std_logic;
8 begin
9     process (clk)
10        variable tempA : integer range 0 to 7:=0;
11        variable tempB : integer range 0 to 3:=0;
12    begin
13        if (rising_edge(clk) ) then
14            if (rst = '0') then
15                internal_state <= A;
16                tempA := 0;
17                tempB := 0;
18            else
19                case internal_state is
20                    when A =>
21                        if (tempA<7) then
22                            tempA := tempA+1;
23                        else
24                            tempA := 0;
25                            internal_state <= B;
26                        end if;
27                    when B =>
28                        if (tempB<3) then
29                            tempB := tempB+1;
30                        else
31                            tempB := 0;
32                            internal_state <= A;
33                        end if;
34                    when others =>
35                        internal_state <= A;
36                end case;
37            end if;
38        end if;
39    end process;
40    salida <= '1' when (internal_state = B) else '0';
41 end control;

```

Código VHDL 1.5: Circuito solución de la Pregunta 4.