

SISTEMAS ELECTRÓNICOS DIGITALES. 1º CONTROL, CURSO 18-19, HOJA 2 de 2

NOTAS IMPORTANTES:

- El único resultado válido será la que se indique en la casilla de solución.
- La resolución de las cuestiones sin una mínima explicación escrita de los pasos seguidos para la obtención de los resultados, producirá penalización en la calificación, considerándose no válida la respuesta.
- La resolución de las cuestiones se realizará en el mismo examen.
- La fecha de publicación de las calificaciones y de la revisión estará dentro de los márgenes establecidos según normativa.
- La duración y puntuación del ejercicio se indica en cada parte.

Cuestión 4. (30%)

Los dos fragmentos de código que aparecen más abajo corresponden, respectivamente, a la descripción de determinado dispositivo y a su *testbench*. El código del dispositivo contiene cuatro errores que impedirían su correcta simulación y/o síntesis. El código del *testbench* no contiene errores. Se pide:

1. Localizar los cuatro errores, explicar por qué es incorrecto el código y escribir la versión corregida.
2. Para el dispositivo corregido, dibuje el diagrama de tiempos con las formas de onda de *a*, *b*, *c*, *d*, *e*, y *f* correspondientes a la ejecución del *testbench*. Se sugiere que comience por las señales generadas por el *testbench* y prosiga luego con las salidas del circuito en pruebas.

entidad.vhd:

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

entity e1 is
  port (
    a, b, c: in  std_logic;
    d      : in  std_logic_vector(7 downto 0);
    e, f    : out std_logic
  );
end e1;

architecture behavioral of e1 is
  variable r1: std_logic_vector(9 downto 0);
begin
  process (a)
    variable r2 is integer range r1'range;
  begin
    if a = '0' then
      r1 := (others => '1');
      r2 := 0;
      f <= '1';
    elsif b'event and b = 1 then
      f <= '1';
      if c = '0' then
        r1 := '1' & d & '0';
        r2 := r1'length - 1;
      elsif r2 > 0 then
        r1 := '1' & r1(r1'high downto 1);
        r2 := r2 - 1;
        if r2 = 0 then
          f <= '0';
        end if;
      end if;
    end if;
    e := r1(0);
  end process;
end behavioral;
```

entidad tb.vhd:

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

```
entity e1_tb is
end e1_tb;

architecture behavior of e1_tb is
  component e1
    port(
      a, b, c: in  std_logic;
      d      : in  std_logic_vector(7 downto 0);
      e, f    : out std_logic
    );
  end component;

  signal a, b, c: std_logic;
  signal d      : std_logic_vector(7 downto 0);
  signal e, f    : std_logic;

  constant k: time := 10 ns;

begin
  uut: e1 port map (a, b, c, d, e, f);

  p0: process
  begin
    wait for 0.5 * k;
    b <= '0';
    wait for 0.5 * k;
    b <= '1';
  end process;

  a <= '0' after 0.25 * k, '1' after 0.75 * k;

  p1: process
  begin
    c <= '1';
    wait until a = '1';
    wait until b = '0';
    wait for 0.25 * k;
    d <= x"77";
    c <= '0', '1' after 0.5 * k;
    wait until f = '0';
    wait until f = '1';
    wait for 0.25 * k;
    assert false
      report "[SUCCESS]: simulation finished"
        severity failure;
  end process;
end;
```

Cuestión 5. (15%)

El objetivo del siguiente código VHDL es crear un contador con reset asíncrono, señal de habilitación y salida con un número de bits configurable (inicializarlo a 8), con la siguiente definición de interfaz:

- Entradas:
 - reset: reset del sistema, activo a nivel alto.
 - clk: reloj del sistema.
 - enable: activación (nivel alto) / desactivación (nivel bajo) del contador.
- Salidas:
 - count: salida del contador.

Se proporciona el siguiente código al que le faltan algunas sentencias / comandos / datos.
Se pide completar el código.

```
1  LIBRARY _____;
2  USE IEEE.STD_LOGIC_1164.all;
3  USE IEEE.NUMERIC_STD.all;
4
5  ENTITY count IS
6      GENERIC (width: _____);
7      PORT (clk: _____;
8            reset: _____;
9            enable: _____;
10           count: _____ std_logic_vector(_____));
11 END count;
12
13 ARCHITECTURE arch1 OF _____ IS
14     SIGNAL cnt: UNSIGNED(width-1 DOWNTO _____);
15 BEGIN
16     pSeq: PROCESS ( _____, _____) IS
17     BEGIN
18         IF reset = _____ THEN
19             cnt <= _____;
20         ELSIF clk' _____ AND _____ THEN
21             IF (enable= _____) THEN
22                 _____;
23             END IF;
24         END IF;
25     END _____;
26     count <= std_logic_vector( _____ );
27 END arch1;
```

Cuestión 6. (15%)

1) Para $x = "110010"$, de tipo BIT_VECTOR (5 DOWNTO 0), determine los valores de las operaciones de cambio enumeradas en la columna de la izquierda a continuación.

2) En la columna de la derecha, escriba una expresión equivalente utilizando el operador de concatenación (el primero ya está hecho).

a) $x \ll 3 =$	a) $x(2 \text{ DOWNTO } 0) \& "000"$
b) $x \ll -2 =$	b)
c) $x \gg 2 =$	c)
d) $x \ll 1 =$	d)
e) $x \gg -3 =$	e)

SISTEMAS ELECTRÓNICOS DIGITALES

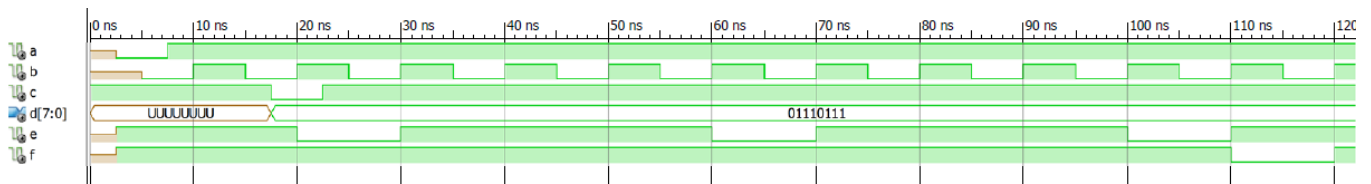
1º EXAMEN PARCIAL - CURSO 18-19 - SOLUCIÓN

Cuestión 4. (30%)

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

entity e1 is
  port (
    a, b, c: in  std_logic;
    d       : in  std_logic_vector(7 downto 0);
    e, f    : out std_logic
  );
end e1;

architecture behavioral of e1 is
  variable r1: std_logic_vector(9 downto 0); (1)
begin
  process (a) (2)
  process (a, b) (2)
    variable r1: std_logic_vector(9 downto 0); (1)
    variable r2 is integer range r1'range; (3)
    subtype r2 t is integer range r1'range; (3)
    variable r2: r2 t; (3)
  begin
    if a = '0' then
      r1 := (others => '1');
      r2 := 0;
      f <= '1';
    elsif b'event and b = 1 then
      f <= '1';
      if c = '0' then
        r1 := '1' & d & '0';
        r2 := r1'length - 1;
      elsif r2 > 0 then
        r1 := '1' & r1(r1'high downto 1);
        r2 := r2 - 1;
        if r2 = 0 then
          f <= '0';
        end if;
      end if;
    end if;
    end if;
    e := r1(0); (4)
    e <= r1(0); (4)
  end process;
end behavioral;
end process;
end;
```



Cuestión 5. (15%)

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.NUMERIC_STD.all;

ENTITY count IS
  GENERIC (width:POSITIVE:=8);
  PORT (clk : IN std_logic;
        reset : IN std_logic;
        enable: IN std_logic;
        count : OUT std_logic_vector(width-1 DOWNT0 0)
  );
END count;

ARCHITECTURE arch1 OF count IS
  SIGNAL cnt : UNSIGNED(width-1 DOWNT0 0);

BEGIN

  pSeq : PROCESS (clk, reset) IS
  BEGIN
    IF reset = '1' THEN
      cnt <= (others => '0');
    ELSIF clk'event AND clk='1' THEN
      IF enable='1' THEN
        cnt <= cnt + 1;
      END IF;
    END IF;
  END PROCESS;

  count <= std_logic_vector(cnt);

END arch1;
```

Cuestión 6. (15%)

a) x SLL 3 = "010000"	a) x(2 DOWNT0 0) & "000";
b) x SLA -2 = "111100"	b) x(5) & x(5) & x(5 DOWNT0 2);
c) x SRA 2 = "111100"	c) x(5) & x(5) & x(5 DOWNT0 2);
d) x ROL 1 = "100101"	d) x(4 DOWNT0 0) & x(5);
e) x ROR -3 = "010110"	e) x(2 DOWNT0 0) & x(5 DOWNT0 3);