

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERU
FACULTAD DE ESTUDIOS GENERALES CIENCIAS



Solución a las experiencias del Laboratorio 6

EXPERIENCIA 1 DEL LABORATORIO 6

ALUMNO:

MATEO GUERRERO ISUIZA

CÓDIGO:

20191867

HORARIO:

0441

PROFESOR:

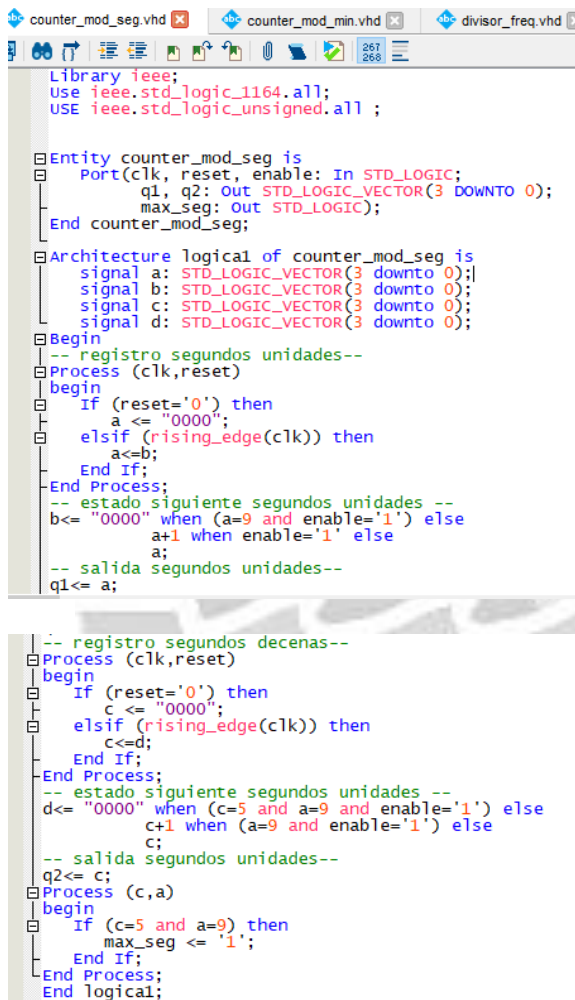
Flores Espinoza Donato Andrés

2020-2

Lima, Octubre, 2020

1. EXPERIENCIA 1

- i. Realizar los circuitos: \neg counter_mod_seg (archivo counter_mod_seg.vhd), que es el contador módulo 60. (1 punto) y el counter_mod_min (archivo counter_mod_min.vhd), que es el contador módulo 24. (3 puntos)



```

counter_mod_seg.vhd
counter_mod_min.vhd
divisor_freq.vhd

Library ieee;
use ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all ;

Entity counter_mod_seg is
Port(clk, reset, enable: In STD_LOGIC;
q1, q2: Out STD_LOGIC_VECTOR(3 DOWNTO 0);
max_seg: Out STD_LOGIC);
End counter_mod_seg;

Architecture logical of counter_mod_seg is
signal a: STD_LOGIC_VECTOR(3 downto 0);
signal b: STD_LOGIC_VECTOR(3 downto 0);
signal c: STD_LOGIC_VECTOR(3 downto 0);
signal d: STD_LOGIC_VECTOR(3 downto 0);
Begin
-- registro segundos unidades--
Process (clk,reset)
begin
If (reset='0') then
a <= "0000";
elsif (rising_edge(clk)) then
a<=b;
End If;
End Process;
-- estado siguiente segundos unidades --
b<= "0000" when (a=9 and enable='1') else
a+1 when enable='1' else
a;
-- salida segundos unidades--
q1<= a;

-- registro segundos decenas--
Process (clk,reset)
begin
If (reset='0') then
c <= "0000";
elsif (rising_edge(clk)) then
c<=d;
End If;
End Process;
-- estado siguiente segundos unidades --
d<= "0000" when (c=5 and a=9 and enable='1') else
c+1 when (a=9 and enable='1') else
c;
-- salida segundos unidades--
q2<= c;
Process (c,a)
begin
If (c=5 and a=9) then
max_seg <= '1';
End If;
End Process;
End logical;

```

```

counter_mod_min.vhd  divisor_freq.vhd  Compilation Report - reloj
Library ieee;
Use ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all ;

Entity counter_mod_min is
Port(clk, reset, enable: In STD_LOGIC;
      q1, q2: Out STD_LOGIC_VECTOR(3 DOWNT0 0);
      max_min: Out STD_LOGIC);
End counter_mod_min;

Architecture logica2 of counter_mod_min is
Signal a: STD_LOGIC_VECTOR(3 downto 0);
Signal b: STD_LOGIC_VECTOR(3 downto 0);
Signal c: STD_LOGIC_VECTOR(3 downto 0);
Signal d: STD_LOGIC_VECTOR(3 downto 0);
Begin
-- registro minutos unidades--
Process (clk,reset)
begin
If (reset='0') then
a <= "0000";
elsif (rising_edge(clk)) then
a <= b;
End If;
--End Process;
-- estado siguiente segundos unidades --
b <= "0000" when (a=9 and enable='1') else
"0000" when (a=3 and c=2 and enable='1') else
a+1 when enable='1' else
a;
-- salida segundos unidades--
q1 <= std_logic_vector(a);

-- registro segundos decenas--
Process (clk,reset)
begin
If (reset='0') then
c <= "0000";
elsif (rising_edge(clk)) then
c <= d;
End If;
--End Process;
-- estado siguiente segundos unidades --
d <= "0000" when (c=2 and a=3 and enable='1') else
c+1 when (c=9 and enable='1') else
c;
-- salida segundos unidades--
q2 <= std_logic_vector(c);
Process (c, a)
begin
If (a=3 and c=2) then
max_min <= '1';
End If;
End Process;
End logica2;

```

- ii. Realizar el package my_components (my_components.vhd), el cual debe contener las definiciones de los circuitos de inciso anterior junto con las del circuito divisor_freq. (1 punto)

```

reloj.vhd  my_components.vhd  counter_mod_seg.vhd  counter_mod_min.vhd
1 Library ieee;
2 Use ieee.std_logic_1164.all;
3
4 Package my_components is
5 Component counter_mod_seg
6 Port(clk, reset, enable: In STD_LOGIC;
7       q1, q2: Out STD_LOGIC_VECTOR(3 DOWNT0 0);
8       max_seg: Out STD_LOGIC);
9 End component;
10 Component counter_mod_min
11 Port(clk, reset, enable: In STD_LOGIC;
12       q1, q2: Out STD_LOGIC_VECTOR(3 DOWNT0 0);
13       max_min: Out STD_LOGIC);
14 End component;
15 Component divisor_freq
16 Port (signal reset_n : in std_logic;
17       signal clk : in std_logic;
18       signal clk_o : out std_logic);
19 End component;
20 End my_components;

```

- iii. Realizar la descripción del circuito reloj (archivo reloj.vhd). (6 puntos)

```

Library ieee ;
Use ieee.std_logic_1164.all ;
Use work.my_components.all ;

Entity reloj is
  Port (clk, reset_n, ini_pausa, borrar: in std_logic ;
        max_value: out std_logic ;
        segundos_decenas : out std_logic_vector (1 to 7);
        segundos_unidades : out std_logic_vector (1 to 7);
        minutos_decenas: out std_logic_vector (1 to 7);
        minutos_unidades: out std_logic_vector (1 to 7));
End reloj;

Architecture logica of reloj is
  signal en_seg: std_logic;
  signal en_min: std_logic;
  signal clk_en: std_logic;
  signal g_reset_n: std_logic;
  signal q1: std_logic_vector (3 DOWNTO 0);
  signal q2: std_logic_vector (3 DOWNTO 0);
  signal max_seg: std_logic;
  signal max_min: std_logic;
  signal p1: std_logic_vector (3 DOWNTO 0);
  signal p2: std_logic_vector (3 DOWNTO 0);
Begin
  contadorseg: counter_mod_seg PORT MAP (clk,g_reset_n,en_seg
  contadormin: counter_mod_min PORT MAP (clk,g_reset_n,en_min
  divisorfre : divisor_freq PORT MAP (g_reset_n,clk,clk_en
  -- señales internas --
  Process (clk_en, ini_pausa, en_seg, max_seg, borrar, reset_n)

begin
  g_reset_n <= (not(borrar) or ini_pausa) and reset_n;
  en_seg <= clk_en and ini_pausa;
  en_min <= en_seg and max_seg;
End Process;

-- salida --
Process (max_seg, max_min, q1)
begin
  max_value <= max_min and max_seg;
  case (q1) is
    when "0000" => segundos_unidades<="1111110"; --abcde
    when "0001" => segundos_unidades<="0110000";
    when "0010" => segundos_unidades<="1101101";
    when "0011" => segundos_unidades<="1111001";
    when "0100" => segundos_unidades<="0110011";
    when "0101" => segundos_unidades<="1011011";
    when "0110" => segundos_unidades<="1011111";
    when "0111" => segundos_unidades<="1110000";
    when "1000" => segundos_unidades<="1111111";
    when "1001" => segundos_unidades<="1111011";
    when OTHERS => segundos_unidades<="-----";
  end case;
  case (q2) is
    when "0000" => segundos_decenas<="1111110"; --abcdef
    when "0001" => segundos_decenas<="0110000";
    when "0010" => segundos_decenas<="1101101";
    when "0011" => segundos_decenas<="1111001";
    when "0100" => segundos_decenas<="0110011";
    when OTHERS => segundos_decenas<="-----";
  end case;

  case (p1) is
    when "0000" => minutos_unidades<="1111110"; --abcde
    when "0001" => minutos_unidades<="0110000";
    when "0010" => minutos_unidades<="1101101";
    when "0011" => minutos_unidades<="1111001";
    when "0100" => minutos_unidades<="0110011";
    when "0101" => minutos_unidades<="1011011";
    when "0110" => minutos_unidades<="1011111";
    when "0111" => minutos_unidades<="1110000";
    when "1000" => minutos_unidades<="1111111";
    when "1001" => minutos_unidades<="1111011";
    when OTHERS => minutos_unidades<="-----";
  end case;
  case (p2) is
    when "0000" => minutos_decenas<="1111110"; --abcde
    when "0001" => minutos_decenas<="0110000";
    when "0010" => minutos_decenas<="1101101";
    when OTHERS => minutos_decenas<="-----";
  end case;
End Process;
End logica;

```

- iv. Aclaración: Antes de compilar (realizar la síntesis) entre en: Project -> add/remove files in Project. Asegúrese de que el orden de los archivos sea: counter_mod_seg.vhd (parte superior), counter_mod_min.vhd, divisor_freq.vhd y reloj.vhd (parte inferior), (cualquier cosa emplee los botones de Up o Down).



-

- 5

