

PRÁCTICA 9

Diseño de cronómetros digitales

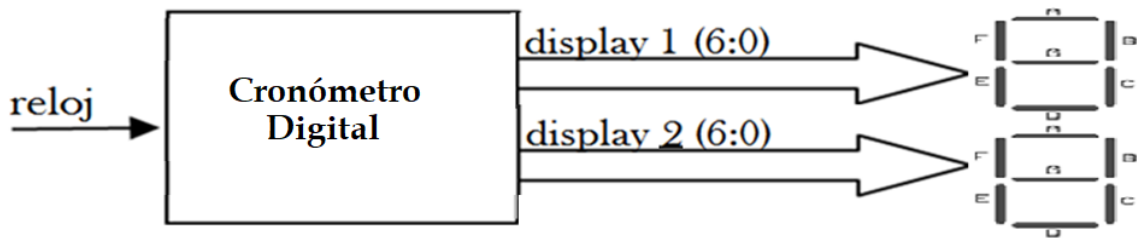
OBJETIVO:

El alumno comprenderá utilizando lenguaje VHDL, el funcionamiento y manejo de contadores, cuando la entrada de reloj sea una cuenta específica.

ESPECIFICACIONES:

Se requiere el diseño y construcción de un cronómetro digital que en 2 displays de 7 segmentos se visualice la cuenta, empezando en 00 y al llegar a 59, se reinicie esta. La siguiente figura muestra el diagrama del bloque de éste sistema.

DIAGRAMA DE BLOQUE:



ANÁLISIS DE LA CUENTA:

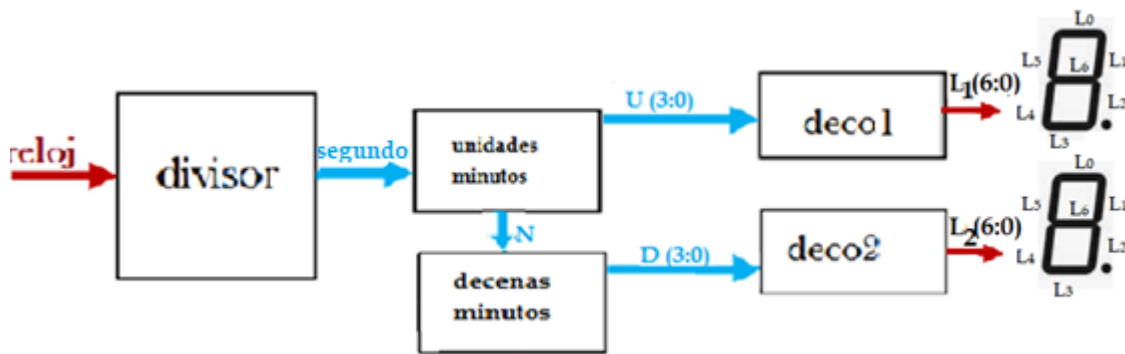
Contador Decenas	Contador Unidades	
0	0	
0	1	
.	.	
0	9	← Se inicia la cuenta y se aumenta un uno a la siguiente columna
1	0	
1	1	
.	.	
1	9	← Se inicia la cuenta y se aumenta un uno a la siguiente columna
2	0	
2	1	
.	.	
.	.	
5	9	← Se inicia la cuenta en ambos contadores

Dentro del cronómetro digital existen varios bloques funcionales, los cuales serán unidos mediante señales. La siguiente figura muestra los bloques funcionales del sistema "cronómetro":

a) **Para tarjetas con los displays de 7 segmentos conectados en forma independiente:**

Dentro del cronómetro digital existen varios bloques funcionales, los cuales serán unidos mediante señales. La siguiente figura muestra los bloques funcionales del sistema "cronómetro":

BLOQUES FUNCIONALES:



CÓDIGO VHDL DEL SISTEMA:

```

library IEEE;
use IEEE.std_logic_1164.ALL;
use IEEE.std_logic_arith.ALL;
use IEEE.std_logic_unsigned.ALL;

entity cronometro is
    Port ( reloj : in std_logic;
          display1, display2: out std_logic_vector (6 downto 0));
end cronometro;

architecture Behavioral of cronometro is
    signal segundo, N : std_logic;
    signal U, D : std_logic_vector(3 downto 0);
begin

```

```

Divi : process (reloj)
variable cuenta: std_logic_vector(27 downto 0) := X"0000000";
begin
    if rising_edge (reloj) then
    if cuenta =X"48009E0" then
    cuenta := X"0000000";
    else
    cuenta := cuenta+1;
    end if; end if;
    segundo <= cuenta (24);
    end process;

```

```

Unidades: process (segundo)
variable cuenta: std_logic_vector( 3 downto 0) := "0000";
begin
    if rising_edge (segundo) then
    if cuenta ="1001" then
    cuenta :="0000";
    N <= '1';
    else
    cuenta := cuenta + 1;
    N <= '0';
    end if; end if;
    U <= cuenta;
    end process;

```

```

Decenas: process (N)
variable cuenta: std_logic_vector( 3 downto 0) := "0000";
begin
    if rising_edge (N) then
    if cuenta ="0101" then
    cuenta :="0000";
    else
    cuenta := cuenta +1;
    end if; end if;
    D <= cuenta;
    end process;

```

with U select

```
display1 <= "1000000" when "0000", --0
           "1111001" when "0001", --1
           "0100100" when "0010", --2
           "0110000" when "0011", --3
           "0011001" when "0100", --4
           "0010010" when "0101", --5
           "0000010" when "0110", --6
           "1111000" when "0111", --7
           "0000000" when "1000", --8
           "0010000" when "1001", --9
           "1000000" when others; --F
```

with D select

```
display2<= "1000000" when "0000", --0
           "1111001" when "0001", --1
           "0100100" when "0010", --2
           "0110000" when "0011", --3
           "0011001" when "0100", --4
           "0010010" when "0101", --5
           "1000000" when others; --F
```

end Behavioral;



<https://www.youtube.com/watch?v=BN9nVSgiJm8>