

CRUNIVERSIDAD NACIONAL AUTÓNAMA DE
MÉXICO



FACULTAD DE INGENIERÍA



REPORTE DE PRÁCTICA 11:

DISEÑO Y CONSTRUCCIÓN DE UN CRONÓMETRO
DIGITAL

ALUMNO:

JAVIER SOLÍS MARTÍNEZ

PROFESORA:

M.I. NORMA ELVA CHÁVEZ

FECHA DE ENTREGA:

24/NOVIEMBRE/2013

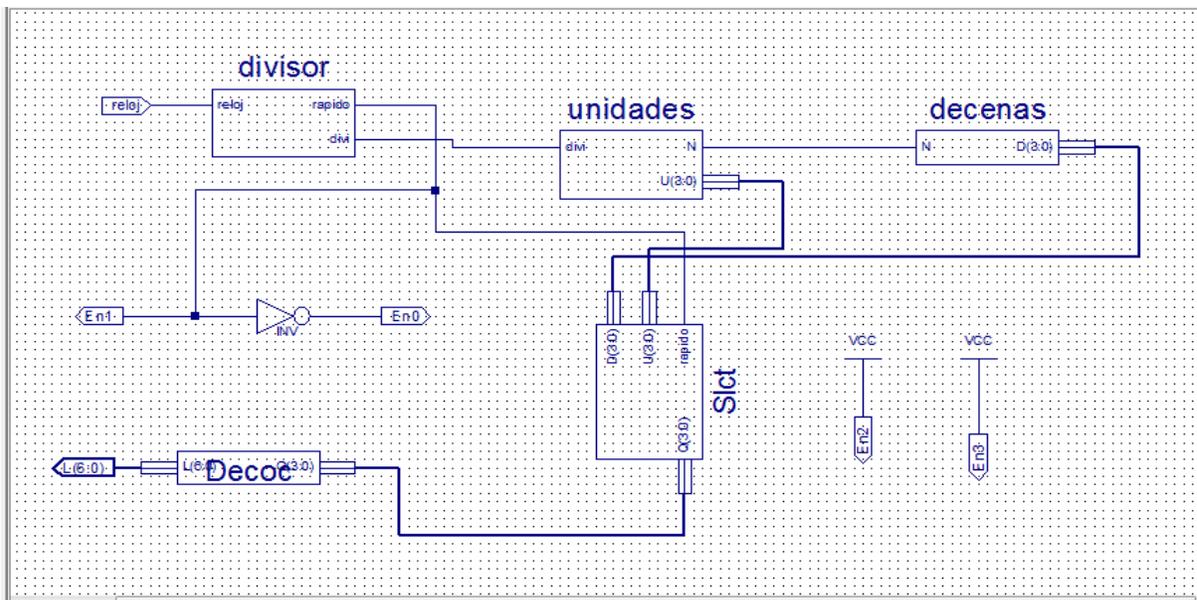
OBJETIVO

El alumno aprenderá a diseñar utilizando lenguaje VHD, un cronómetro utilizando contadores con corte de cuenta y dos displays en paralelo, que muestren en forma visual diferente información en cada uno de ellos.

ESPECIFICACIONES

Diseñar un cronómetro digital utilizando dos de los cuatro display's que tienen las tarjetas Spartan3 en paralelo, se construirá un cronómetro digital el cual empiece la cuenta en 00 y la termine en 59 para volver empezar.

DIAGRAMA DE BLOQUES:

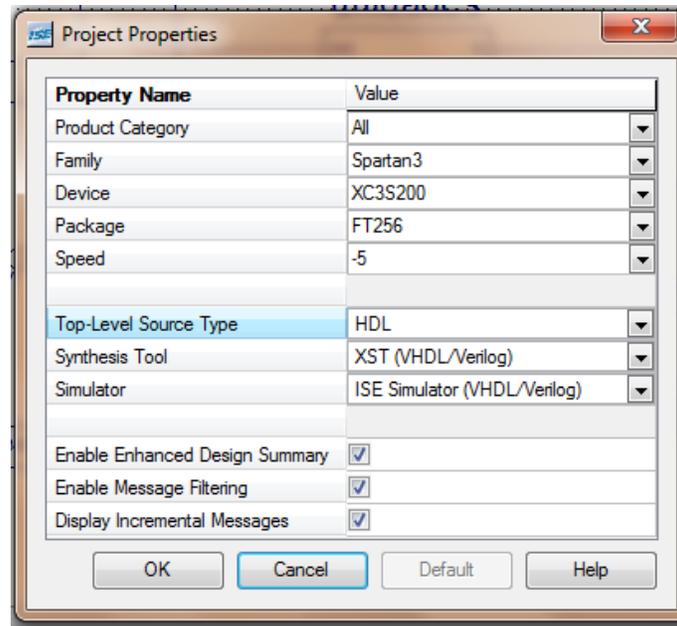


PROCEDIMIENTO:

Para este proyecto haremos uso de la plataforma ISE de Xilinx y la tarjeta Spartan3. Crearemos una carpeta llamada "Cronometro".

Abrimos la plataforma ISE, creamos un nuevo proyecto lo guardaremos como "Cronometro" en la misma carpeta que se creó previamente.

Para el funcionamiento con la tarjeta Spartan3 pondremos la siguiente configuración en la plataforma:



CÓDIGO:

- 1) Primero crearemos el divisor con entra el reloj de un bit, salidas rápido y divi de un bit.

```
entity divisor is
  Port ( reloj : in  STD_LOGIC;
        rapido : out STD_LOGIC;
        divi : out  STD_LOGIC);
end divisor;

architecture Behavioral of divisor is
begin

  Process(reloj)
  variable cuenta: std_logic_vector(27 downto 0) := X"00000000";
  begin
    if rising_edge (reloj) then
      if cuenta= X"48009E0" then -- Tiempo de 1.51s
        cuenta:=X"00000000";
      else
        cuenta:= cuenta+1;
      end if;
    end if;
    divi <= cuenta(25);
    rapido <= cuenta(10);
  end Process;

end Behavioral;
```

Después de tener el código se revisa sintaxis y se crea el símbolo esquemático.

- 2) Ahora crearemos el bloque de unidades con entrada divi y salidas N y U (3:0).

```

entity unidades is
  Port ( divi : in  STD_LOGIC;
        N : out  STD_LOGIC;
        U : out  STD_LOGIC_VECTOR (3 downto 0));
end unidades;

architecture Behavioral of unidades is

begin

  Process(divi)
  variable cuenta: std logic vector(3 downto 0):="0000";
  begin
    if rising edge(divi) then
      if cuenta="1001" then
        cuenta:="0000";
        N<='1';
      else
        cuenta:=cuenta+1;
        N<='0';
      end if;
    end if;
    U<=cuenta;
  end Process;
end Behavioral;

```

Se revisa la sintaxis y se crea su símbolo.

- 3) Ahora crearemos el bloque de decenas con entrada N y salida D (3:0):

```

entity decenas is
  Port ( N : in  STD_LOGIC;
        D : out  STD_LOGIC_VECTOR (3 downto 0));
end decenas;

architecture Behavioral of decenas is

begin

  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;
end Behavioral;

```

Se revisa la sintaxis y se crea su símbolo.

- 4) Ahora crearemos el bloque de selector con entradas U(3:0), D(3:0) y rápido de un bit y salida Q(3:0):

```

entity Slct is
  Port ( U : in  STD_LOGIC_VECTOR (3 downto 0);
        D : in  STD_LOGIC_VECTOR (3 downto 0);
        rapido : in  STD_LOGIC;
        Q : out  STD_LOGIC_VECTOR (3 downto 0));
end Slct;

architecture Behavioral of Slct is
begin
  Process(rapido)
  begin
    case rapido is
      when '0' =>
        Q <= U;
      when others =>
        Q <= D;
    end case;
  end Process;
end Slct;

```

Se revisa la sintaxis y se crea su símbolo.

5) Por ultimo crearemos el deco con entrada Q (3:0) y salida L (6:0):

```

entity Decoc is
  Port ( Q : in  STD_LOGIC_VECTOR (3 downto 0);
        L : out  STD_LOGIC_VECTOR (6 downto 0));
end Decoc;

architecture Behavioral of Decoc is
begin
  Process(Q)
  begin
    case Q is
      when "0000" =>
        L <= "1000000";
      when "0001" =>
        L <= "1111001";
      when "0010" =>
        L <= "0100100";
      when "0101" =>
        L <= "0011001";
      when "0110" =>
        L <= "0000011";
      when "0111" =>
        L <= "1111000";
      when "1000" =>
        L <= "0000000";
      when others =>
        L <= "0011000";
    end case;
  end Process;
end Decoc;

```

Se revisa sintaxis y se crea su símbolo.

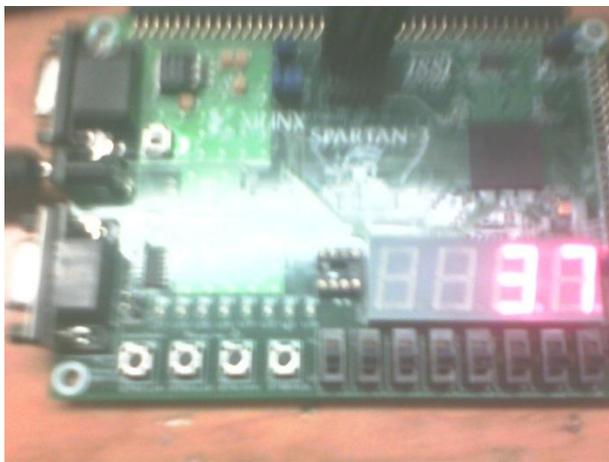
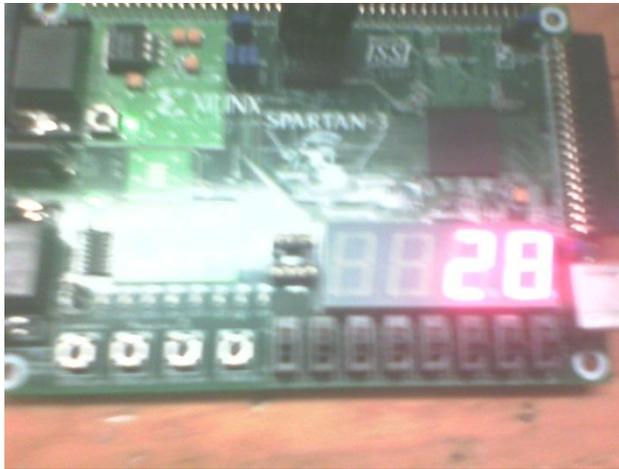
Una vez teniendo cada uno de los bloques, creamos una nueva fuente que la llamaremos TOP y que sea esquemático interconectamos cada uno de los bloques.

Sintetizamos el Top, y ahora crearemos el área de construcción asignando los pines a cada salida y entrada.

Creamos el archivo con extensión .bit, abrimos el programa "Adept" y cargamos nuestro Top a la tarjeta Spartan3 .

Resultados





CONCLUSIÓN:

EN esta práctica se observó y se aprendió a utilizar los contadores con corte de cuenta para poder hacer un cronometro de 00 al 59 utilizando la herramienta ISE de Xilinx y la Spartan 3E