```
--ESTE PROGRAMA ESTA DISEÑADO PARA CONTAR LAS RPM
--DE UN MOTOR DE CD, REALIZADAS PARA CADA SEGUNDO
--ASI COMO PARA CADA MINUTO, ES DECIR EL CODIGO
--PRESENTE ES PARA UN "TACOMETRO"
--ESTE CÓDIGO FUE PROBADO EN UNA NEXYS2

--DECLARACION DE BIBLIOTECAS
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

--DECLARACION DE ENTRADAS Y SALIDAS
entity ContadorBCD is
   Port ( MHZ50 : in  STD_LOGIC;--RELOJ INTERNO DE LA TARJETA 50MHZ
          SEG7 : out  STD_LOGIC_VECTOR (6 downto 0);--ASIGANCION PARA LOS 7 SEGMENTOS
          SEL7 : out  STD_LOGIC_VECTOR (3 downto 0);--VARIABLE DE ASIGNACION DE SALIDA DEL
MULTIPLEXOR
                              PULSO : IN STD_LOGIC;--AYUDA A LA DETECCIÓN DE PULSOS
--LEDS MUESTRAN LA CUENTA QUE SE LLEVA
                              LEDS : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
                              TM : IN STD_LOGIC;--MUESTRA LAS ACTUALIZACIONESS
                              MUL: IN STD_LOGIC;--MULTIPLICA POR 1 O POR 60
                              R: IN STD_LOGIC;--RESET
          HZ : out  STD_LOGIC);--SALIDA DEL RELOJ DE 1HZ
end ContadorBCD;
```

```
architecture Behavioral of ContadorBCD is

--DECLARACION DE SEÑALES INTERNAS A USARSE
SIGNAL C_SEL:INTEGER RANGE 0 TO 50000:=0;
SIGNAL S_SEL: INTEGER RANGE 0 TO 3:=0;
SIGNAL C_HZ: INTEGER RANGE 0 TO 50000000:=0;
SIGNAL APULSO:STD_LOGIC;
SIGNAL CONTADOR: STD_LOGIC_VECTOR(15 DOWNTO 0) :=X"0000";
SIGNAL P:STD_LOGIC_VECTOR(18 DOWNTO 0);
SIGNAL U,D,C,M:STD_LOGIC_VECTOR (6 DOWNTO 0);
SIGNAL SIM: STD_LOGIC_VECTOR(5 DOWNTO 0);
SIGNAL Conv_BCD: STD_LOGIC_VECTOR(15 DOWNTO 0) :=X"0000";
SIGNAL T_MOSTRAR: INTEGER RANGE 0 TO 50000000:=0;

begin


T_MOSTRAR<=0 WHEN R='1' OR T_MOSTRAR=50000000 ELSE T_MOSTRAR+1 WHEN MHZ50'EVENT AND MHZ50='1';

-- TIEMPO PARA EL DISPLAY
  C_SEL <= 0 WHEN R='1' OR C_SEL=50000 ELSE
                C_SEL+1 WHEN MHZ50'EVENT AND MHZ50='1';

-- SELECCIONADOR
```

```
  S_SEL <= 0 WHEN R='1' OR S_SEL = 4 ELSE

                S_SEL+1 WHEN C_SEL=0 AND MHZ50'EVENT AND MHZ50='1';


-- DEMUX

  SEL7 <= "1110" WHEN S_SEL=0 ELSE

               "1101" WHEN S_SEL=1 ELSE

                              "1011" WHEN S_SEL=2 ELSE

                              "0111" ;


-- TIEMPO PARA GENERAR SEÑAL DE HZ

  C_HZ <= 0 WHEN R='1' OR C_SEL=50000000 ELSE

                C_HZ+1 WHEN MHZ50'EVENT AND MHZ50='1';


-- PULSO DE 1Hz

  HZ<= '0' WHEN R='1' OR C_HZ <25000000 ELSE

             '1';


-- DETECCIÓN FLANCO POSITIVO

APULSO <= PULSO WHEN MHZ50'EVENT AND MHZ50='1';


-- CONTADOR DE PULSOS

CONTADOR<= X"0000"  WHEN R='1' OR CONTADOR = X"2329" or ( T_mostrar=1 and TM='1') ELSE

       CONTADOR+1 WHEN APULSO='0' AND PULSO='1' AND MHZ50'EVENT AND MHZ50='1';



SIM <= "000001" WHEN MUL='0' ELSE "111100";
```

```
Conv_BCD<=CONTADOR * SIM   WHEN  MHZ50'EVENT AND MHZ50='0' AND (T_mostrar=0  OR TM='0');



-- CONVERTIR DE BIN A BCD

BCD1: PROCESS(Conv_BCD)

                          VARIABLE Z:STD_LOGIC_VECTOR(34 DOWNTO 0);

BEGIN

  FOR I IN 0 TO 34 LOOP

    Z(I):='0';

  END LOOP;


  Z(18 DOWNTO 3):=Conv_BCD;

  FOR I IN 0 TO 12 LOOP

     IF Z(19 DOWNTO 16) > 4 THEN

        Z(19 DOWNTO 16):= Z(19 DOWNTO 16)+3;

     END IF;


     IF Z(23 DOWNTO 20) > 4 THEN

        Z(23 DOWNTO 20):= Z(23 DOWNTO 20)+3;

     END IF;


     IF Z(27 DOWNTO 24) > 4 THEN

        Z(27 DOWNTO 24):= Z(27 DOWNTO 24)+3;

     END IF;
    IF Z(31 DOWNTO 28) > 4 THEN

      Z(31 DOWNTO 28):= Z(31 DOWNTO 28)+3;

    END IF;


            Z(34 DOWNTO 1):= Z(33 DOWNTO 0);


 END LOOP;


  P<=Z(34 DOWNTO 16);

 END PROCESS;




-- DECODIFICADORES 7 SEGMENTOS PARA CADA UNO DE LOS DISPLAYS

-- ESTOS DISPLAYS SE REALIZAN DE DERECHA A IZQUIERDA




--UNIDADES

U <= "0000001" WHEN P(3 DOWNTO 0)="0000" ELSE

    "1001111" WHEN P(3 DOWNTO 0)="0001" ELSE

            "0010010" WHEN P(3 DOWNTO 0)="0010" ELSE

            "0000110" WHEN P(3 DOWNTO 0)="0011" ELSE

            "1001100" WHEN P(3 DOWNTO 0)="0100" ELSE

            "0100100" WHEN P(3 DOWNTO 0)="0101" ELSE

            "0100000" WHEN P(3 DOWNTO 0)="0110" ELSE

            "0001111" WHEN P(3 DOWNTO 0)="0111" ELSE
```

```
          "0000000" WHEN P(3 DOWNTO 0)="1000" ELSE
          "0001100" ;

--DECENAS
D <= "0000001" WHEN P(7 DOWNTO 4)="0000" ELSE
   "1001111" WHEN P(7 DOWNTO 4)="0001" ELSE
          "0010010" WHEN P(7 DOWNTO 4)="0010" ELSE
          "0000110" WHEN P(7 DOWNTO 4)="0011" ELSE
          "1001100" WHEN P(7 DOWNTO 4)="0100" ELSE
          "0100100" WHEN P(7 DOWNTO 4)="0101" ELSE
          "0100000" WHEN P(7 DOWNTO 4)="0110" ELSE
          "0001111" WHEN P(7 DOWNTO 4)="0111" ELSE
          "0000000" WHEN P(7 DOWNTO 4)="1000" ELSE
          "0001100" ;

--CENTENAS
C <= "0000001" WHEN P(11 DOWNTO 8)="0000" ELSE
   "1001111" WHEN P(11 DOWNTO 8)="0001" ELSE
          "0010010" WHEN P(11 DOWNTO 8)="0010" ELSE
          "0000110" WHEN P(11 DOWNTO 8)="0011" ELSE
          "1001100" WHEN P(11 DOWNTO 8)="0100" ELSE
          "0100100" WHEN P(11 DOWNTO 8)="0101" ELSE
          "0100000" WHEN P(11 DOWNTO 8)="0110" ELSE
          "0001111" WHEN P(11 DOWNTO 8)="0111" ELSE
          "0000000" WHEN P(11 DOWNTO 8)="1000" ELSE
          "0001100" ;
```

```
--MILLARES
M <= "0000001" WHEN P(15 DOWNTO 12)="0000" ELSE
   "1001111" WHEN P(15 DOWNTO 12)="0001" ELSE
          "0010010" WHEN P(15 DOWNTO 12)="0010" ELSE
          "0000110" WHEN P(15 DOWNTO 12)="0011" ELSE
          "1001100" WHEN P(15 DOWNTO 12)="0100" ELSE
          "0100100" WHEN P(15 DOWNTO 12)="0101" ELSE
          "0100000" WHEN P(15 DOWNTO 12)="0110" ELSE
          "0001111" WHEN P(15 DOWNTO 12)="0111" ELSE
          "0000000" WHEN P(15 DOWNTO 12)="1000" ELSE
          "0001100" ;



--SE MUESTRA EN DISPLAYS CADA UNA DE LAS CUENTAS
 SEG7 <= U WHEN S_SEL=0 ELSE
     D WHEN S_SEL=1 ELSE
                         C WHEN S_SEL=2 ELSE
                         M ;


--SE MUESTRA EN LOS LEDS LA CUENTA QUE SE LLEVA DE LAS RPM
LEDS<= CONTADOR( 7 DOWNTO 0);


end Behavioral;
```