```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;

package meu_pacote_led is

    component ccto is
            generic (N : integer);
            port(   mclk, reset: in std_logic;
                    clk: out std_logic      );
   end component;

    component ccto2 is
            generic (N : integer);
            port(   mclk, reset: in std_logic;
                    clk: out std_logic      );
   end component;


    component mux2x1 is
            port(   seletor: in std_logic;
                    entrada1: in integer range 0 to 15;
                    entrada2: in integer range 0 to 15;
                    saida: out integer range 0 to 15            );
    end component;

    component dem2x4 is
            port(   aux3 : in std_logic_vector(1 downto 0);
                    aux4 : out std_logic_vector(3 downto 0)         );
    end component;

    component decod_7_seg is

         GENERIC (N : integer := 3);
         port(
         chaves: in std_logic_vector(3 downto 0);
         chaves_display: in std_logic_vector(3 downto 0);
         anodos: out std_logic_vector(3 downto 0);
         catodos: out std_logic_vector(0 to 7)          );

    end component;

    component somachaves is

        port(   soma: inout std_logic_vector(11 downto 0);
                chaves: in std_logic_vector(7 downto 0);
                saida3: out integer range 0 to 15;
                saida2: out integer range 0 to 15;
                saida1: out integer range 0 to 15                );
    end component;

    function to_bcd (bin : std_logic_vector(7 downto 0) ) return std_logic_vector;

end package;

package body meu_pacote_led is      -- double dabble

    function to_bcd ( bin : std_logic_vector(7 downto 0) ) return std_logic_vector is
        variable i : integer:=0;
        variable bcd : std_logic_vector(11 downto 0) := (others => '0');
        variable bint : std_logic_vector(7 downto 0) := bin;

        begin
            for i in 0 to 7 loop -- repeating 8 times.
```

```
                bcd(11 downto 1) := bcd(10 downto 0); --shifting the bits.
                bcd(0) := bint(7);
                bint(7 downto 1) := bint(6 downto 0);
                bint(0) :='0';

                if(i < 7 and bcd(3 downto 0) > "0100") then --add 3 if BCD digit is greater than 4.
                    bcd(3 downto 0) := bcd(3 downto 0) + "0011";
                end if;

                if(i < 7 and bcd(7 downto 4) > "0100") then --add 3 if BCD digit is greater than 4.
                    bcd(7 downto 4) := bcd(7 downto 4) + "0011";
                end if;

                if(i < 7 and bcd(11 downto 8) > "0100") then --add 3 if BCD digit is        greater than 4.
                    bcd(11 downto 8) := bcd(11 downto 8) + "0011";
                end if;

            end loop;
        return bcd;
    end to_bcd;

end meu_pacote_led;
```