```vhdl
-------------------------------------------------------------------------------
--
--  Purpose:
--
--    some test pieces for data type conversion
--    real, integer, std_logic_vector
--
--  Discussion:
--
--
--  Licensing:
--
--    This code is distributed under the GNU LGPL license.
--
--  Modified:
--
--    2012.03.09
--
--  Author:
--
--    Young W. Lim
--
--  Parameters:
--
--    Input:
--
--    Output:
--------------------------------------------------------------------------------




library STD;
use STD.textio.all;

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity test_conv is
end test_conv;

architecture test of test_conv is
  constant y : real := 0.5;
  constant x : real := 7.85398163397448309621E-01;
  constant shft : std_logic_vector (31 downto 0) := X"2000_0000";

  function Conv2fixedPt (x : real; n : integer) return std_logic_vector is
    constant shft : std_logic_vector (n-1 downto 0) := X"2000_0000";
    variable s : std_logic_vector (n-1 downto 0) ;
    variable z : real := 0.0;
  begin
      -- shft = 2^29 = 536870912
      -- bit 31 : msb - sign bit
      -- bit 30,29 : integer part
      -- bit 28 ~ 0 : fractional part
      -- for the value of 0.5
      -- first 4 msb bits [0, 0, 0, 1] --> X"1000_0000"
      --
      -- To obtain binary number representation of x,
      -- where the implicit decimal point between bit 29 and bit 28,
      -- multiply "integer converted shft"
      --
      z := x * real(to_integer(unsigned(shft)));

      s := std_logic_vector(to_signed(integer(z), n));

      return s;

  end Conv2fixedPt;

  function Conv2real (s : std_logic_vector (31 downto 0) ) return real is
    constant shft : std_logic_vector (31 downto 0) := X"2000_0000";
    variable z : real := 0.0;
  begin
```

```vhdl
      z := real(to_integer(signed(s))) / real(to_integer(unsigned(shft)));

    return z;

  end Conv2real;


begin
  process
    variable l : line;
    variable m,n : integer := 0;
    variable s : std_logic_vector (31 downto 0);
    variable z : real := 0.0;
  begin


      write(l, String'("test "));

      s := Conv2fixedPt(x, 32);

      z := Conv2real(s);

      write(l, real'(z));
      writeline(output, l);

      write(l, String'("--------------------------------------- "));
      writeline(output, l);

      z := x * real(to_integer(unsigned(shft)));
      write(l, real'(z));
      writeline(output, l);

      s := std_logic_vector(to_signed(integer(z), 32));

      z := real(to_integer(signed(s))) / real(to_integer(unsigned(shft)));
      write(l, real'(z));
      writeline(output, l);


      write(l, String'("--------------------------------------- "));
      writeline(output, l);

      -- shft = 2^29 = 536870912
      -- bit 31 : msb - sign bit
      -- bit 30,29 : integer part
      -- bit 28 ~ 0 : fractional part
      -- for the value of 0.5
      -- first 4 msb bits [0, 0, 0, 1] --> X"1000_0000"
      --
      -- std_logic_vector --> integer
      --             shft -->       m
      m := to_integer(unsigned(shft));
      write(l, integer'(m));
      write(l, ' ');

      -- To obtain binary number representation of x,
      -- where the implicit decimal point between bit 29 and bit 28,
      -- multiply m (integer converted shft)
      z := x * real(m);
      write(l, real'(z));
      writeline(output, l);


      -- truncate the multiplication result into n
      n := integer(z);
      write(l, integer'(n));
      write(l, ' ');

      -- convert integer n into std_logic_vector
      s := std_logic_vector(to_signed(n, 32));

      -- to verify the result
      m := to_integer(signed(s));
      write(l, integer'(m));
      writeline(output, l);
      z := real(m) / real(to_integer(unsigned(shft)));
      write(l, real'(z));
      writeline(output, l);



    wait;
  end process;


end test;
```