```
--------------------------------------------------------------------------------
--
--  Purpose:
--
--    ROM Model
--
--  Discussion:
--
--
--  Licensing:
--
--    This code is distributed under the GNU LGPL license.
--
--  Modified:
--
--    2012.03.30
--
--  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;

use WORK.cordic_pkg.all;



entity rom is
  generic (
    WD     : in natural := 32;
    SH     : in natural :=  5;
    PWR    : in natural := 64);

  port (
    addr   : in   std_logic_vector (SH-1 downto 0) := (others=>'0');
    cs     : in   std_logic := '0';
    data   : out  std_logic_vector (WD-1 downto 0) := (others=>'0') );

end rom;

architecture rtl of rom is
   type rarray is array (natural range <>) of real;

   constant angles : rarray :=
                     (  7.853981633974483096E-01,  -- pi/4 rad
                        4.636476090008061162E-01,
                        2.449786631268641541E-01,
                        1.243549945467614350E-01,
                        6.241880999595734847E-02,
                        3.123983343026827625E-02,
                        1.562372862047683080E-02,
                        7.812341060101111296E-03,
                        3.906230131966971827E-03,
                        1.953122516478818685E-03,
                        9.765621895593194304E-04,
                        4.882812111948982754E-04,
                        2.441406201493617640E-04,
                        1.220703118936702042E-04,
```

```
                        6.103515617420877502E-05,
                        3.051757811552609686E-05,
                        1.525878906131576210E-05,
                        7.629394531101970263E-06,
                        3.814697265606496283E-06,
                        1.907348632810187035E-06,
                        9.536743164059608794E-07,
                        4.768371582030888599E-07,
                        2.384185791015579825E-07,
                        1.192092895507806853E-07,
                        5.960464477539055441E-08,
                        2.980232238769530368E-08,
                        1.490116119384765515E-08,
                        7.450580596923827987E-09,
                        3.725290298461914045E-09,
                        1.862645149230957029E-09,
                        9.313225746154785154E-10,
                        4.656612873077392578E-10,
                        2.328306436538696289E-10,
                        1.164153218269348145E-10,
                        5.820766091346740722E-11,
                        2.910383045673370361E-11,
                        1.455191522836685181E-11,
                        7.275957614183425903E-12,
                        3.637978807091712952E-12,
                        1.818989403545856476E-12,
                        9.094947017729282379E-13,
                        4.547473508864641190E-13,
                        2.273736754432320595E-13,
                        1.136868377216160297E-13,
                        5.684341886080801486E-14,
                        2.842170943040400744E-14,
                        1.421085471520200372E-14,
                        7.105427357601001859E-15,
                        3.552713678800500929E-15,
                        1.776356839400250465E-15,
                        8.881784197001252323E-16,
                        4.440892098500626162E-16,
                        2.220446049250313081E-16,
                        1.110223024625156540E-16,
                        5.551115123125782702E-17,
                        2.775557561562891351E-17,
                        1.387778780781445676E-17,
                        6.938893903907228378E-18,
                        3.469446951953614189E-18,
                        1.734723475976807094E-18  );


  signal dinInc : std_logic_vector (SH-1 downto 0);

begin

  ROM: process (addr, cs)
    type darray is array (0 to PWR) of std_logic_vector (WD-1 downto 0);
    variable romData : darray;
    variable initRom : boolean := false;
    variable cntInt : integer := 0;
    variable angleMin : std_logic_vector (WD-1 downto 0);
  begin  -- process Reg
    if (initRom=false) then
      for i in 0 to PWR-1 loop
        romData(i) := Conv2fixedPt(angles(i), WD);
      end loop;  -- i
      initRom := true;
    end if;

    if cs = '1' then                    -- asynchronous reset (active low)
      data <= romData(to_integer(unsigned(addr)));
    else
      data <= (others=>'1');
```

```
    end if;
  end process ROM;

end rtl;
```