

```
-----
--
-- Purpose:
--   ROM Model (Data is read from a file)
--
-- Discussion:
--
-- Licensing:
--   This code is distributed under the GNU LGPL license.
--
-- Modified:
--   2012.11.07
--
-- 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;

  variable angles : rarray :=
    ( 7.8539816339744830962E-01,  -- pi/4 rad
      4.6364760900080611621E-01,
      2.4497866312686415417E-01,
      1.2435499454676143503E-01,
      6.2418809995957348474E-02,
      3.1239833430268276254E-02,
      1.5623728620476830803E-02,
      7.8123410601011112965E-03,
      3.9062301319669718276E-03,
      1.9531225164788186851E-03,
      9.7656218955931943040E-04,
      4.8828121119489827547E-04,
      2.4414062014936176402E-04,
      1.2207031189367020424E-04,
```

```
6.1035156174208775022E-05,  
3.0517578115526096862E-05,  
1.5258789061315762107E-05,  
7.6293945311019702634E-06,  
3.8146972656064962829E-06,  
1.9073486328101870354E-06,  
9.5367431640596087942E-07,  
4.7683715820308885993E-07,  
2.3841857910155798249E-07,  
1.1920928955078068531E-07,  
5.9604644775390554414E-08,  
2.9802322387695303677E-08,  
1.4901161193847655147E-08,  
7.4505805969238279871E-09,  
3.7252902984619140453E-09,  
1.8626451492309570291E-09,  
9.3132257461547851536E-10,  
4.6566128730773925778E-10,  
2.3283064365386962890E-10,  
1.1641532182693481445E-10,  
5.8207660913467407226E-11,  
2.9103830456733703613E-11,  
1.4551915228366851807E-11,  
7.2759576141834259033E-12,  
3.6379788070917129517E-12,  
1.8189894035458564758E-12,  
9.0949470177292823792E-13,  
4.5474735088646411896E-13,  
2.2737367544323205948E-13,  
1.1368683772161602974E-13,  
5.6843418860808014870E-14,  
2.8421709430404007435E-14,  
1.4210854715202003717E-14,  
7.1054273576010018587E-15,  
3.5527136788005009294E-15,  
1.7763568394002504647E-15,  
8.8817841970012523234E-16,  
4.4408920985006261617E-16,  
2.2204460492503130808E-16,  
1.1102230246251565404E-16,  
5.5511151231257827021E-17,  
2.7755575615628913511E-17,  
1.3877787807814456755E-17,  
6.9388939039072283776E-18,  
3.4694469519536141888E-18,  
1.7347234759768070944E-18,  
1.7347234759768070944E-18,  
1.7347234759768070944E-18,  
1.7347234759768070944E-18 );
```

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

```
procedure ReadData (variable angles : out rarray) is  
  file DataFile : text open read_mode is "angle.data";  
  variable lbuf : line;  
  variable i : integer := 0;  
  variable rdata : real;  
begin  
  while not endfile(DataFile) loop  
    readline(DataFile, lbuf);  
    read(lbuf, rdata);  
    angles(i) := rdata;  
    i := i + 1;  
  end loop;  
end procedure;
```

```
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
        ReadData(rarray);
        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;

```