CORRECTED APRIL 22 !!

For this assignment, you should write SystemC code for the square root problem below. You should put the following 5 files in your Courses/ECE3485 directory before the due date. Create a PA2 subdirectory in your Courses directory. Name your files main.cpp, sqrt_tb.h, sqrt_tb.cpp, sqrt.h and sqrt.cpp.

To complete this assignment you should write behavioral SystemC and simulate it using the scirocco VHDL simulator. This is similar to the example in ~libhls/systemc/distance.

Write a SystemC behavioral description of a circuit whose output Y is the square root of its input Xin. Both Xin and Y are 16 bit, signed integers. Your circuit should implement the following pseudo-code:
Wait until Init = 1;
  DiffX := XIn;
  YShift := 0;
  B := 16384;
  Done := 0;
  loop until Done = 1;
    if DiffX = 0 then
      DiffX := DiffX;
      YShift := YShift div 2;
      B := B div 4;
    else
      if DiffX > 0 then
        DiffX := DiffX - YShift - B;
        YShift := (YShift + B * 2) div 2;
        B := B div 4;
      else
        DiffX := DiffX + YShift - B;
        YShift := (YShift - B * 2) div 2;
        B := B div 4;
      end loop;
    end if;
  end if;
  Y := YShift;

Your circuit should wait until Init is 1 and then read the input. When Done is 1 it should write the output to Y.

This algorithm calculates the exact square root when \( x \) is a square. Otherwise the answer might be less than or greater than the exact square root, but it will always be odd. Here are some examples on how the algorithm runs on different inputs. Here we show \( y \) and \( x - y^2 \). The algorithm works on a shifted version of \( y \) called YShift and a shifted version of \( x - y^2 \) called DiffX:

\[
\begin{align*}
x = 18_{10} &= 10100_2 & \text{iterate 1} & y = 100 & x - y^2 = + \\
\text{iterate 2} & y = 110 & x - y^2 = - \\
\text{iterate 3} & y = 101 \\
x = 15_{10} &= 01111_2 & \text{iterate 1} & y = 100 & x - y^2 = - \\
\text{iterate 2} & y = 010 & x - y^2 = + \\
\text{iterate 3} & y = 011 \\
x = 16_{10} &= 10000_2 & \text{iterate 1} & y = 100 & x - y^2 = 0
\end{align*}
\]

The pseudo-code describes the algorithm. It is not necessarily the best way to code your SystemC! Think about your coding style before you write SystemC for the behavior.

Be sure to use behavioral SystemC and only synthesizable data types.