Name: Priyansh Agrahari
Roll Number: CS20B1014

(The following design does not compile, it is only to show the planned implementation of the CORDIC algorithm as we were taught in class)
(The biggest problem is the use of whole numbers)
VHDL Design Module for 2D rotation mode CORDIC:

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

--2d rot mode cordic
entity rot_cordic is
port(
  x0: in std_logic_vector(15 downto 0);
  y0: in std_logic_vector(15 downto 0);
  a:  in std_logic_vector(15 downto 0);
  xf: out std_logic_vector(15 downto 0);
  yf: out std_logic_vector(15 downto 0)
  );
end rot_cordic;

architecture rtl of rot_cordic is
begin
     process (l)
     variable xi: std_logic_vector(15 downto 0);
     variable yi: std_logic_vector(15 downto 0);
     variable d: std_logic_vector(15 downto 0) := (others => '0');
    begin
     for I in 0 to 15 loop
           d := a - right_shift(unsigned(45), I);
            if (d > '0') then
              --cw rotation
              xi := x0 + right_shift(y0, I);
              yi := y0 - right_shift(x0, I);
              a <= a - right_shift(unsigned(45), I);
            else
              --ccw rotation
              xi := x0 - right_shift(y0, I);
              yi := y0 + right_shift(x0, I);
              a <= a + right_shift(unsigned(45), I);
            x0 <= xi;
            y0 <= yi;
```

```
        end loop;
        xf <= x0;
        yf <= y0;
    end process;
end rtl;
```