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 vectoring mode CORDIC:

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

--2d vec mode cordic
entity vec_cordic is
port(
  x0: in std_logic_vector(15 downto 0);
  y0: in std_logic_vector(15 downto 0);
  r:  out std_logic_vector(15 downto 0);
  a:  out std_logic_vector(15 downto 0)
  );
end vec_cordic;

architecture rtl of vec_cordic is
begin
     process (l)
     variable xi: std_logic_vector(15 downto 0);
     variable yi: std_logic_vector(15 downto 0);
     variable sig: std_logic_vector(15 downto 0) := (others => '0');
    begin
     a <= "0000000000000000";
     for I in 0 to 15 loop
          if (y0 > '0') then
             --cw rotation
             xi := x0 + right_shift(y0, I);
             yi := y0 - right_shift(x0, I);
           else
             --ccw rotation
             xi := x0 - right_shift(y0, I);
             yi := y0 + right_shift(x0, I);
             sig(I) := '1';
           x0 <= xi;
           y0 <= yi;
           --add or sub angle acc to cw or ccw rot
           if (sig(I) = '0') then
```

```
                    a <= a + right_shift(unsigned(45), I);
                else
                      a <= a - right_shift(unsigned(45), I);
                end if;
            end loop;
            --after rotating to x axis x ccord is norm
            r <= x0;
        end process;
end rtl;
```