```
-------------------------------------------------------------------------
---------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 11/04/2020 04:21:10 PM
-- Design Name: 
-- Module Name: inmultireBooth_tb - Behavioral
-- Project Name: 
-- Target Devices: 
-- Tool Versions: 
-- Description: 
-- 
-- Dependencies: 
-- 
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-- 
-------------------------------------------------------------------------
---------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_arith.all;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity inmultireBooth_tb is
--  Port ( );
end inmultireBooth_tb;

architecture Behavioral of inmultireBooth_tb is
constant CLK_PERIOD: time := 20 ns;
signal Clk: std_logic;
signal Rst, Start: std_logic:='0';
signal X, Y: std_logic_vector(7 downto 0):= (others => '0');
signal A, Q: std_logic_vector(7 downto 0);
signal iesire: std_logic_vector(15 downto 0);
signal Term: std_logic;

begin

CLK_process :process
begin
    Clk <= '0';
    wait for CLK_PERIOD/2;
      Clk <= '1';
      wait for CLK_PERIOD/2;
end process;
```

```
DUT: entity WORK.inmultireBooth
            port map(Clk=>Clk, Rst=>Rst, Start=>Start, X=>X, Y=>Y, A=>A,
Q=>Q, Term=>Term);

--proc2: process
--begin
    --wait for 20 ns;
    --Rst<='1';
    --Start<='1';
    --wait for 20 ns;
    --Rst<='0';
    --wait for 20 ns;
    --X<=x"11";
    --wait for 20 ns;
    --Y<=x"10";
    --wait for 20 ns;
    --Start<='1';
    --wait for 20 ns;
    --wait;
--end process;

process
variable sign: integer:= 1;
begin

    wait for 20 ns;
    Rst<='1';
    wait for 20 ns;
    Rst<='0';

    --punctul 6
    for i in -2 to -2 loop
        for j in -2 to -2 loop
            X <= conv_std_logic_vector(i, 8);
            Y <= conv_std_logic_vector(j, 8);
            Start <= '1';
            wait for 300 ns;
            iesire <= A&Q;
        end loop;
    end loop;

    --punctul 7
    for i in 125 to 127 loop
        for j in 125 to 127 loop
            if i = 127 and j = 127 then sign:=-1;
            X <= conv_std_logic_vector(i*sign, 8);
            Y <= conv_std_logic_vector(j*sign, 8);
            end if;
            Start <= '1';
            wait for 300 ns;
            iesire <= A&Q;
        end loop;
    end loop;

end process;
end Behavioral;
```