**VHDL code to implement SISO using Behavioral modeling**

**T Raja Aadhithan**

**602162021**

**Design Code :**

-- design code

library ieee;

use ieee.std\_logic\_1164.all;

entity si\_so is port (clk,i : in std\_logic; y: out std\_logic); end si\_so;

architecture behavioral of si\_so is signal q1,q2,q3 : std\_logic; begin

        process(clk) begin

            if(rising\_edge (clk))then

                q1 <= i;

                q2 <= q1;

                q3 <= q2;

                y <= q3;

            end if;

        end process;

end behavioral;

**Self Checking Test bench:**

-- self checking test bench

library ieee;

use ieee.std\_logic\_1164.all;

entity tb is end tb;

architecture behaviour of tb is

    component si\_so is port(clk,i : in std\_logic; y: out std\_logic); end component;

    signal i,clk : std\_logic := '0';

    signal y,temp,err : std\_logic;

    signal x : integer := 0;

    begin uut : si\_so port map(clk => clk, i=> i, y =>y);

        clk\_process : process begin

            clk <= '0';

            wait for 10 ns;

            clk <= '1';

            wait for 10 ns;

        end process;

        stim\_process : process begin

            i <= '0','1' after 50 ns,'0' after 100 ns, '1' after 150 ns,'0' after 200 ns, '1' after 250 ns,'0' after 300 ns,'1' after 350 ns, '1' after 400 ns,'0' after 450 ns, '0' after 500 ns,'1' after 550 ns, '1' after 600 ns,'0' after 650 ns, '1' after 700 ns,'1' after 750 ns;

        wait;

        end process;

        check : process

         begin

            temp <= i;

            wait until clk'event and clk = '1';

            wait until clk'event and clk = '1';

            wait until clk'event and clk = '1';

            wait until clk'event and clk = '1';

            err <= temp xor y;

            if(err = '1')then x <= x+1; end if;

        end process;

        final : process begin

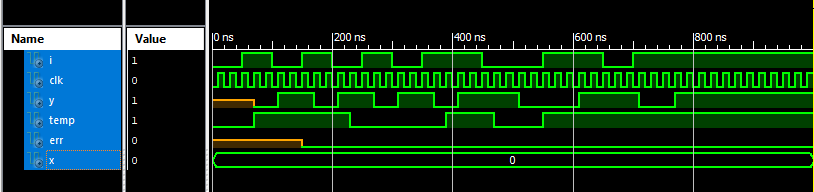
        wait for 800 ns;

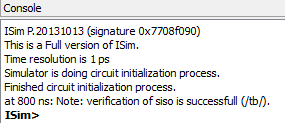
        if (x = 0) then report "verification of siso is successfull"; end if;

        end process;

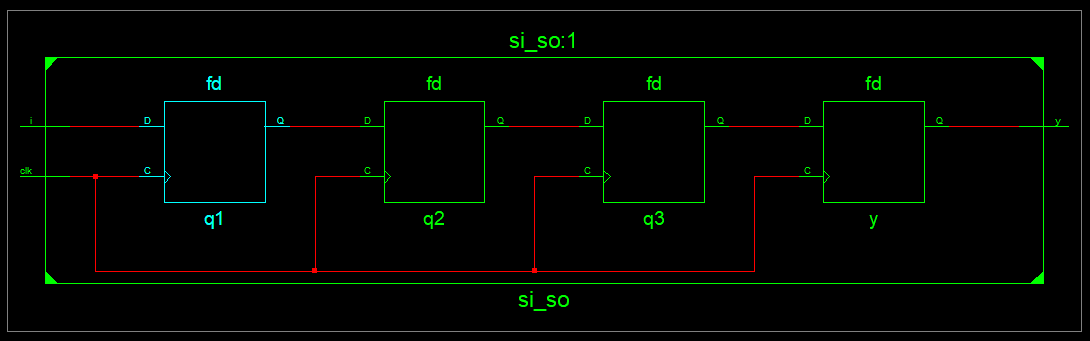
end behaviour;

**Simulation Output:**





**RTL Diagram :**

****