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

**T Raja Aadhithan**

**602162021**

**Design Code :**

-- design code:

library ieee;

use ieee.std\_logic\_1164.all;

entity pi\_po is port (

    clk : in std\_logic;

    d1,d2,d3,d4 : in std\_logic;

    q1,q2,q3,q4 : out std\_logic);

end pi\_po;

architecture behavioural of pi\_po is begin

    process(clk) begin

        if(rising\_edge (clk)) then

            q4 <= d4;

            q1 <= d1;

            q2 <= d2;

            q3 <= d3;

        end if;

    end process;

end behavioural;

**Self Checking Testbench :**

-- self checking test bench:

library ieee;

use ieee.std\_logic\_1164.all;

entity tb is end tb;

architecture behaviour of tb is

    component pi\_po is port(clk,d1,d2,d3,d4 : in std\_logic; q1,q2,q3,q4: out std\_logic); end component;

    signal d1,d2,d3,d4,clk : std\_logic := '0';

    signal q1,q2,q3,q4,t1,t2,t3,t4,err : std\_logic;

    signal x : integer := 0;

    begin uut : pi\_po port map(clk => clk, q1=> q1, q2=> q2, q3=> q3, q4=> q4, d1=> d1, d2=> d2, d3=> d3, d4=> d4);

        clk\_process : process begin

            clk <= '0';

            wait for 10 ns;

            clk <= '1';

            wait for 10 ns;

        end process;

        stim\_process : process begin

            d1 <= '1' after 50 ns, '0' after 60 ns, '1' after 450 ns, '0' after 700 ns;

            d2 <= '1' after 100 ns, '0' after 150 ns, '1' after 350 ns, '0' after 600 ns;

            d3 <= '1' after 100 ns, '0' after 200 ns, '1' after 250 ns, '0' after 750 ns;

            d4 <= '1' after 500 ns, '0' after 550 ns, '1' after 650 ns, '0' after 700 ns;

        wait;

        end process;

        check : process

         begin

            t1 <= d1; t2 <= d2; t3 <= d3; t4 <= d4;

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

            err <= (t1 xor q1) and (t2 xor q2) and(t3 xor q3) and(t4 xor q4);

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

        end process;

        final : process begin

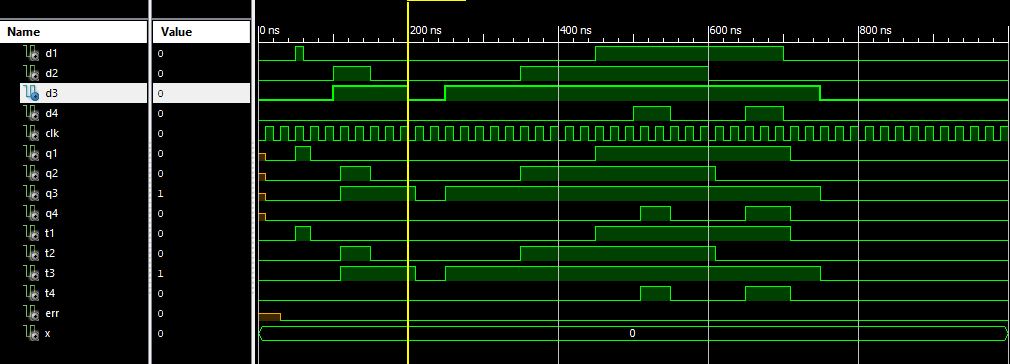
        wait for 800 ns;

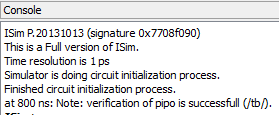
        if (x = 0) then report "verification of pipo is successfull"; end if;

        end process;

end behaviour;

**Simulation result:**

****

****

**RTL Diagram:**

