**VHDL code to implement UP/DOWN Counter using Behavioral modeling**

**Design Code:**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity count is port (clk,u\_d,rst : in std\_logic; q : out std\_logic\_vector(3 downto 0)); end count;

architecture behaviour of count is

    signal temp : std\_logic\_vector(3 downto 0);

    begin process(clk) -- synchronous reset

        begin

        if (clk = '1')then

        if (rst = '1') then temp <= "0000";

        elsif (rst = '0') then

            if (u\_d = '0') then temp <= temp + 1; -- up if 0

            elsif (u\_d = '1') then temp <= temp - 1; -- down if 1

            else temp <= temp;

            end if;

        end if;

        end if;

        q <= temp;

    end process;

end behaviour;

**Test Bench Code:**

library ieee;

use ieee.std\_logic\_1164.all;

entity tb is end tb;

architecture test of tb is

    component count is port(clk,u\_d,rst : in std\_logic; q: out std\_logic\_vector(3 downto 0));  end component;

    signal clk,rst,u\_d : std\_logic := '0';

    signal q : std\_logic\_vector(3 downto 0);

    begin uut : count port map(clk => clk, rst => rst, u\_d => u\_d, q =>q);

        clock : process begin

            clk <= '1';

            wait for 10 ns;

            clk <= '0';

            wait for 10 ns;

        end process;

        stim : process begin

            rst <= '1' after 50 ns,'0' after 100 ns;

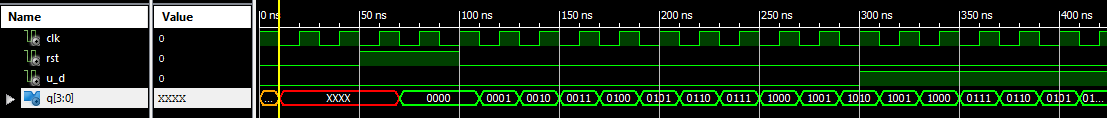
            u\_d <= '1' after 300 ns, '0' after 500 ns;

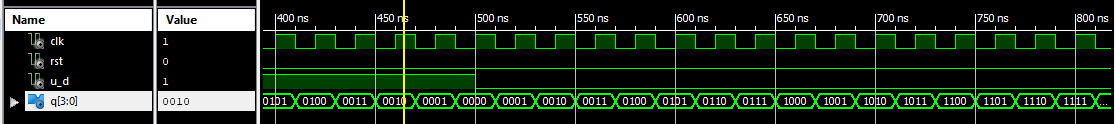
            wait;

        end process;

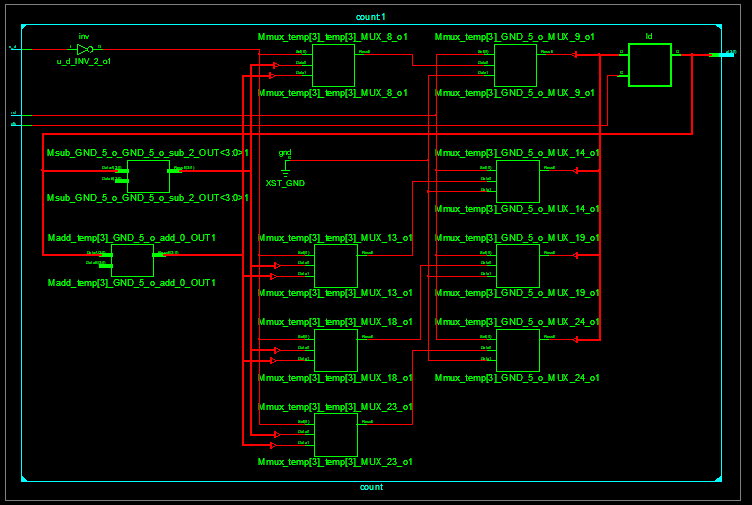
end test;

**Simulation Result:**

****

****

**RTL Diagram:**

****