**VHDL code to implement JK Flip Flop using Behavioral modeling**

**T Raja Aadhithan**

**602162021**

Design Code:

library ieee;

use ieee.std\_logic\_1164.all;

entity jk\_ff is port (j,k,clk,rst: in std\_logic; q,qb: out std\_logic); end jk\_ff;

architecture ff of jk\_ff is

    begin

    process(j,k,clk,rst)

        variable temp : std\_logic := '0';

        begin

            if (rst = '1') then temp := '0';

            elsif (rising\_edge (clk)) then

                if (j /= k) then temp := j;

                elsif (j = '1' and k = '1') then temp := not temp;

                end if;

            end if;

        q <= temp;

        qb <= not temp;

    end process;

end ff;

Test bench Code:

library ieee;

use ieee.std\_logic\_1164.all;

entity tb is end tb;

architecture behaviour of tb is

    component jk\_ff is port (j,k,clk,rst : in std\_logic; q,qb :out std\_logic); end component;

    signal j,k,clk,rst : std\_logic := '0';

    signal q,qb :std\_logic;

    begin uut : jk\_ff port map(j => j, k => k, clk => clk, rst => rst, q => q, qb => qb);

        clk\_process : process begin

            clk <= '0';

            wait for 10 ns;

            clk <= '1';

            wait for 10 ns;

        end process;

        stim\_process : process begin

         rst <= '1', '0' after 50 ns;

         j <= '0','1' after 150 ns, '0' after 250 ns, '1' after 350 ns;

         k <= '0','1' after 100 ns, '0' after 150 ns,'1' after 200 ns,

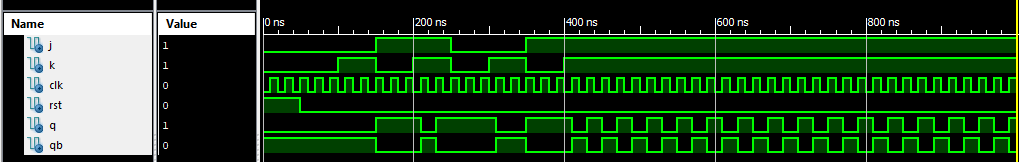
  '0' after 250 ns,'1' after 300 ns, '0' after 350 ns,'1' after 400 ns;

         wait;

        end process;

end behaviour;

Simulation result:



RTL Diagram: 