library IEEE;

use IEEE.STD\_LOGIC\_1164.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 primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity register1 is

Port ( d : in STD\_LOGIC;

clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

q : out STD\_LOGIC);

end register1;

architecture Behavioral of register1 is

begin

process(clk,rst)

variable internal:std\_logic\_vector(3 downto 0);

begin

if(rst='1')then

internal:=(others=>'0');

elsif(clk'event and clk='1')then

internal:=d&internal(3 downto 1);

end if;

q<=internal(0);

end process;

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--USE ieee.numeric\_std.ALL;

ENTITY registertest IS

END registertest;

ARCHITECTURE behavior OF registertest IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT register1

PORT(

d : IN std\_logic;

clk : IN std\_logic;

rst : IN std\_logic;

q : OUT std\_logic

);

END COMPONENT;

--Inputs

signal d : std\_logic := '0';

signal clk : std\_logic := '0';

signal rst : std\_logic := '0';

--Outputs

signal q : std\_logic;

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: register1 PORT MAP (

d => d,

clk => clk,

rst => rst,

q => q

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

rst<='1';

wait for 100 ns;

rst<='0';

wait for 100 ns;

wait for clk\_period\*10;

-- insert stimulus here

wait;

end process;

END;