MOD N COUNTER ( MAINCODE):

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_arith.ALL;

use IEEE.STD\_LOGIC\_unsigned.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 COUNTER is

Port ( DIR : in STD\_LOGIC;

-- PRT : in STD\_LOGIC;

RST : in STD\_LOGIC;

PR : in STD\_LOGIC;

CLK : in STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR(3 downto 0));

end COUNTER;

architecture Behavioral of COUNTER is

signal tmp :std\_logic\_vector(3 downto 0):= "1111";

begin

Process (DIR,CLK,RST)

begin

if (RST='1')then tmp <= "0000";

elsif (Pr ='1')then tmp <= "1111";

elsif rising\_edge (CLK) then

if (DIR = '1') then

tmp <= tmp+1;

else

tmp <= tmp-1;

end if;

end if;

end Process;

Q<=TMP;

end Behavioral;

TESTBENCH:

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 COUNTER\_00 IS

END COUNTER\_00;

ARCHITECTURE behavior OF COUNTER\_00 IS

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

COMPONENT COUNTER

PORT(

DIR : IN std\_logic;

RST : IN std\_logic;

PR : IN std\_logic;

CLK : IN std\_logic;

Q : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

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

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

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

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

--Outputs

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

-- Clock period definitions

constant CLK\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: COUNTER PORT MAP (

DIR => DIR,

RST => RST,

PR => PR,

CLK => CLK,

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.

wait for 100 ns;

wait for CLK\_period\*10;

-- insert stimulus here

wait;

end process;

END;