**Traffic light controller**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

-- Traffic light system for a intersection between highway and farm way

-- There is a snsr on the farm way side, when there are vehicles,

-- Traffic light turns to YELLOW, then GREEN to let the vehicles cross the highway

-- Otherwise, always green light on Highway and Red light on farm way

entity traff\_light\_control is

port ( snsr : in STD\_LOGIC; -- sensor

clk : in STD\_LOGIC; -- clock

rst\_n: in STD\_LOGIC; -- reset active low

light\_highway : out STD\_LOGIC\_VECTOR(2 downto 0); -- light outputs of high way

light\_farm: out STD\_LOGIC\_VECTOR(2 downto 0)-- light outputs of farm way

--RED\_YELLOW\_GREEN

);

end traff\_light\_control ;

architecture traffic\_light of traff\_light\_control is

signal counter\_1s: std\_logic\_vector(27 downto 0):= x"0000000";

signal delat\_ct:std\_logic\_vector(3 downto 0):= x"0";

signal delay\_10s, delay\_3s\_F, delay\_3s\_H, RED\_LIGHT\_ENABLE, YELLOW\_LIGHT1\_ENABLE, YELLOW\_LIGHT2\_ENABLE: std\_logic: ='0';

signal clk\_1s\_enable: std\_logic; -- 1s clock enable

type FSM\_States is (HighGreen\_FarmRed, HighYellow\_FarmRed, HighRed\_FarmGreen, HighRed\_FarmYellow);

-- HighGreen\_FarmRed : Highway green and farmway red

-- HighYellow\_FarmRed : Highway yellow and farmway red

-- HighRed\_FarmGreen : Highway red and farmway green

-- HighRed\_FarmYellow : Highway red and farmway yellow

signal curr\_state, nxt\_state: FSM\_States;

begin

-- next state FSM sequential logic

process(clk,rst\_n)

begin

if(rst\_n='0') then

curr\_state <= HighGreen\_FarmRed;

elsif(rising\_edge(clk)) then

curr\_state <= nxt\_state;

end if;

end process;

-- FSM combinational logic

process(curr\_state, snsr,delay\_3s\_F,delay\_3s\_H,delay\_10s)

begin

case curr\_state is

when HighGreen\_FarmRed => -- When Green light on Highway and Red light on Farm way

RED\_LIGHT\_ENABLE <= '0';-- disable RED light delay counting

YELLOW\_LIGHT1\_ENABLE <= '0';-- disable YELLOW light Highway delay counting

YELLOW\_LIGHT2\_ENABLE <= '0';-- disable YELLOW light Farmway delay counting

light\_highway <= "001"; -- Green light on Highway

light\_farm <= "100"; -- Red light on Farm way

if(snsr = '1') then -- if vehicle is detected on farm way by snsrs

nxt\_state <= HighYellow\_FarmRed;

-- High way turns to Yellow light

nxt\_state <= HighGreen\_FarmRed;

-- Otherwise, remains GREEN ON highway and RED on Farm way

end if;

when HighYellow\_FarmRed => -- When Yellow light on Highway and Red light on Farm way

light\_highway <= "010";-- Yellow light on Highway

light\_farm <= "100";-- Red light on Farm way

RED\_LIGHT\_ENABLE <= '0';-- disable RED light delay counting

YELLOW\_LIGHT1\_ENABLE <= '1';-- enable YELLOW light Highway delay counting

YELLOW\_LIGHT2\_ENABLE <= '0';-- disable YELLOW light Farmway delay counting

if(delay\_3s\_H='1') then

-- if Yellow light delay counts to 3s,

-- turn Highway to RED,

-- Farm way to green light

nxt\_state <= HighRed\_FarmGreen;

else

nxt\_state <= HighYellow\_FarmRed;

-- Remains Yellow on highway and Red on Farm way

-- if Yellow light not yet in 3s

end if;

when HighRed\_FarmGreen =>

light\_highway <= "100";-- RED light on Highway

light\_farm <= "001";-- GREEN light on Farm way

RED\_LIGHT\_ENABLE <= '1';-- enable RED light delay counting

YELLOW\_LIGHT1\_ENABLE <= '0';-- disable YELLOW light Highway delay counting

YELLOW\_LIGHT2\_ENABLE <= '0';-- disable YELLOW light Farmway delay counting

if(delay\_10s='1') then

-- if RED light on highway is 10s, Farm way turns to Yellow

nxt\_state <= HighRed\_FarmYellow;

else

nxt\_state <= HighRed\_FarmGreen;

-- Remains if delay counts for RED light on highway not enough 10s

end if;

when HighRed\_FarmYellow =>

light\_highway <= "100";-- RED light on Highway

light\_farm <= "010";-- Yellow light on Farm way

RED\_LIGHT\_ENABLE <= '0'; -- disable RED light delay counting

YELLOW\_LIGHT1\_ENABLE <= '0';-- disable YELLOW light Highway delay counting

YELLOW\_LIGHT2\_ENABLE <= '1';-- enable YELLOW light Farmway delay counting

if(delay\_3s\_F='1') then

-- if delay for Yellow light is 3s,

-- turn highway to GREEN light

-- Farm way to RED Light

nxt\_state <= HighGreen\_FarmRed;

else

nxt\_state <= HighRed\_FarmYellow;

-- if not enough 3s, remain the same state

end if;

when others => nxt\_state <= HighGreen\_FarmRed; -- Green on highway, red on farm way

end case;

end process;

-- Delay counts for Yellow and RED light

process(clk)

begin

if(rising\_edge(clk)) then

if(clk\_1s\_enable='1') then

if(RED\_LIGHT\_ENABLE='1' or YELLOW\_LIGHT1\_ENABLE='1' or YELLOW\_LIGHT2\_ENABLE='1') then

delat\_ct <= delat\_ct + x"1";

if((delat\_ct = x"9") and RED\_LIGHT\_ENABLE ='1') then

delay\_10s <= '1';

delay\_3s\_H <= '0';

delay\_3s\_F <= '0';

delat\_ct <= x"0";

elsif((delat\_ct = x"2") and YELLOW\_LIGHT1\_ENABLE= '1') then

delay\_10s <= '0';

delay\_3s\_H <= '1';

delay\_3s\_F <= '0';

delat\_ct <= x"0";

elsif((delat\_ct = x"2") and YELLOW\_LIGHT2\_ENABLE= '1') then

delay\_10s <= '0';

delay\_3s\_H <= '0';

delay\_3s\_F <= '1';

delat\_ct <= x"0";

else

delay\_10s <= '0';

delay\_3s\_H <= '0';

delay\_3s\_F <= '0';

end if;

end if;

end if;

end if;

end process;

-- create delay 1s

process(clk)

begin

if(rising\_edge(clk)) then

counter\_1s <= counter\_1s + x"0000001";

if(counter\_1s >= x"0000003") then -- x"0004" is for simulation

-- change to x"2FAF080" for 50 MHz clock running real FPGA

counter\_1s <= x"0000000";

end if;

end if;

end process;

clk\_1s\_enable <= '1' when counter\_1s = x"0003" else '0'; -- x"0002" is for simulation

-- x"2FAF080" for 50Mhz clock on FPGA

end traffic\_light;

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

-- Testbench VHDL code for traffic light controller

ENTITY tb\_ traff\_light\_control IS

END tb\_ traff\_light\_control ;

ARCHITECTURE behavior OF tb\_ traff\_light\_control IS

-- Component Declaration for the traffic light controller

COMPONENT traff\_light\_control

PORT(

snsr : IN std\_logic;

clk : IN std\_logic;

rst\_n : IN std\_logic;

light\_highway : OUT std\_logic\_vector(2 downto 0);

light\_farm : OUT std\_logic\_vector(2 downto 0)

);

END COMPONENT;

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

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

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

--Outputs

signal light\_highway : std\_logic\_vector(2 downto 0);

signal light\_farm : std\_logic\_vector(2 downto 0);

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the traffic light controller

trafficlightcontroller : traff\_light\_control PORT MAP (

snsr => snsr,

clk => clk,

rst\_n => rst\_n,

light\_highway => light\_highway,

light\_farm => light\_farm

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

stim\_proc: process

begin

rst\_n <= '0';

snsr <= '0';

wait for clk\_period\*10;

rst\_n <= '1';

wait for clk\_period\*20;

snsr <= '1';

wait for clk\_period\*100;

snsr <= '0';

wait;

end process;

END;