**Appendix A: VHDL Code Implementing State Machines to Control Trains**

-- ORGATE.VHD (VHDL)

-- This code produces a negative-logic OR circuit

-- Vikram Jain

-- ECE2031 L09

-- 10/10/2013

-- State machine to control trains

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.all;

USE IEEE.STD\_LOGIC\_ARITH.all;

USE IEEE.STD\_LOGIC\_UNSIGNED.all;

-- Sensors and switches are specified beforehand

ENTITY Tcontrol IS

PORT(reset, clock, sensor1, sensor2 : IN std\_logic;

sensor3, sensor4, sensor5, sensor6 : IN std\_logic;

switch1, switch2, switch3, switch4 : OUT std\_logic;

dirA, dirB : OUT std\_logic\_vector(1 DOWNTO 0));

END Tcontrol;

-- Sensors and sensor combinations I use specified

-- State machine names specified

ARCHITECTURE a OF Tcontrol IS

TYPE STATE\_TYPE IS (Initial, Athrough, Athrough2, Bthrough, Bthrough2,

Ain, Ain2, Bpass, Apass, Apass2);

SIGNAL state : STATE\_TYPE;

SIGNAL sensor12, sensor13, sensor24 : std\_logic\_vector(1 DOWNTO 0);

SIGNAL sensor34, sensor26, sensor52 : std\_logic\_vector(1 DOWNTO 0);

SIGNAL sensor43 : std\_logic\_vector(1 DOWNTO 0);

-- Lists the state machines and what happens at each part

BEGIN

PROCESS (clock, reset)

BEGIN

-- Reset to this state

IF reset = '1' THEN

state <= Initial;

ELSIF clock'EVENT AND clock = '1' THEN

-- Case statement to determine next state

CASE state IS

WHEN Initial =>

CASE Sensor13 IS

WHEN "00" => state <= Initial;

WHEN "01" => state <= Bthrough;

WHEN "10" => state <= Athrough;

WHEN "11" => state <= Athrough2;

WHEN OTHERS => state <= Initial;

END CASE;

WHEN Athrough =>

CASE Sensor43 IS

WHEN "00" => state <= Athrough;

WHEN "01" => state <= Athrough2;

WHEN "10" => state <= Initial;

WHEN "11" => state <= Ain;

WHEN OTHERS => state <= Athrough;

END CASE;

WHEN Athrough2 =>

IF Sensor4 = '1' THEN

state <= Ain;

ELSE

state <= Athrough2;

END IF;

WHEN Bthrough2 =>

IF Sensor2 = '1' THEN

state <= Bthrough;

ELSE

state <= Bthrough2;

END IF;

WHEN Bthrough =>

CASE Sensor12 IS

WHEN "00" => state <= Bthrough;

WHEN "01" => state <= Initial;

WHEN "10" => state <= Bthrough2;

WHEN "11" => state <= Athrough;

WHEN OTHERS => state <= Bthrough2;

END CASE;

WHEN Ain =>

CASE Sensor52 IS

WHEN "00" => state <= Ain;

WHEN "01" => state <= Ain2;

WHEN "10" => state <= Apass;

WHEN "11" => state <= Apass;

WHEN OTHERS => state <= Ain;

END CASE;

WHEN Ain2 =>

IF Sensor5 = '1' THEN

state <= Bpass;

ELSE

state <= Ain2;

END IF;

WHEN Bpass =>

IF Sensor3 = '1' THEN

state <= Apass;

ELSE

state <= Bpass;

END IF;

WHEN Apass =>

CASE Sensor26 IS

WHEN "00" => state <= Apass;

WHEN "01" => state <= Bthrough;

WHEN "10" => state <= Apass2;

WHEN "11" => state <= Initial;

WHEN OTHERS => state <= Apass;

END CASE;

WHEN Apass2 =>

IF Sensor6 = '1' THEN

state <= Initial;

ELSE

state <= Apass2;

END IF;

END CASE;

END IF;

END PROCESS;

-- lists what the sensor combinations mean

-- combine bits for case statements above

-- "&" operator combines bits

sensor12 <= sensor1 & sensor2;

sensor13 <= sensor1 & sensor3;

sensor24 <= sensor2 & sensor4;

sensor34 <= sensor3 & sensor4;

sensor26 <= sensor2 & sensor6;

sensor52 <= sensor5 & sensor2;

sensor43 <= sensor4 & sensor3;

-- These outputs do not depend on the state

-- Outputs that depend on state

-- gives outputs of each switch depending on the state

WITH state SELECT

Switch1 <= '0' WHEN Athrough,

'0' WHEN Athrough2,

'1' WHEN Bthrough,

'1' WHEN Bthrough2,

'1' WHEN Ain,

'0' WHEN Initial,

'1' WHEN Bpass,

'1' WHEN Apass2,

'1' WHEN Ain2,

'1' WHEN Apass;

WITH state SELECT

Switch2 <= '0' WHEN Athrough,

'0' WHEN Athrough2,

'1' WHEN Bthrough,

'1' WHEN Bthrough2,

'1' WHEN Ain,

'0' WHEN Initial,

'1' WHEN Bpass,

'1' WHEN Apass2,

'1' WHEN Ain2,

'1' WHEN Apass;

WITH state SELECT

Switch3 <= '0' WHEN Initial,

'0' WHEN Bthrough,

'1' WHEN Ain,

'1' WHEN Apass,

'1' WHEN Ain2,

'0' WHEN Athrough,

'0' WHEN Athrough2,

'1' WHEN Bthrough2,

'1' WHEN Bpass,

'1' WHEN Apass2;

WITH state SELECT

Switch4 <= '0' WHEN Initial,

'0' WHEN Bthrough,

'1' WHEN Ain,

'1' WHEN Apass,

'1' WHEN Ain2,

'0' WHEN Athrough,

'0' WHEN Athrough2,

'1' WHEN Bthrough2,

'1' WHEN Bpass,

'1' WHEN Apass2;

WITH state SELECT

DirA <= "01" WHEN Initial,

"01" WHEN Athrough,

"01" WHEN Athrough2,

"00" WHEN Bthrough2,

"01" WHEN Bthrough,

"01" WHEN Ain,

"01" WHEN Ain2,

"00" WHEN Bpass,

"01" WHEN Apass,

"01" WHEN Apass2;

WITH state SELECT

DirB <= "10" WHEN Initial,

"10" WHEN Athrough,

"00" WHEN Athrough2,

"10" WHEN Bthrough2,

"10" WHEN Bthrough,

"10" WHEN Ain,

"00" WHEN Ain2,

"10" WHEN Bpass,

"10" WHEN Apass,

"00" WHEN Apass2;

END a;