Final Project Report: Simple Traffic Light Controller

Author: Aziz Yuldashev

Report Due Date: 03/16/11

Course: ECEGR 401- VHDL

Instructor: Dr. Margarita Takach

Department of Electrical and Computer Engineering

Seattle University
Objective

In this project the main objective was to design a simple traffic light controller and illustrate its operation by programming Altera DE2 Cynclone series education board, which displayed traffic lights on a screen connected though a serial output.

Equipment and Tools

In this project the following equipment and tools were used:

Tools:
- VHDL
- Quartus II compiler designed by Altera

Equipment:
- Altera DE2 Development and Education Board
- Nec computer screen of MultiSync XV15+ series

Equipment Set-Up

The diagram below illustrates the set up of the equipment used in this project:
Design Procedure

The idea was to implement a traffic light controller for an intersection with eight lights. Each street has one lane which then divides into two lanes: one to turn left only and the other to go straight or to turn right, at the intersection. One of the streets is a main street, meaning that the lights to go straight are always green unless there are cars on the intersecting street or a car on the main street appears on the left only lane. The diagram below best illustrates this picture:

As shown above, the street intersecting the main street has two sensors on each direction. On sensor for a left turn and one sensor for a right turn control the lights together with a left turn sensor on the main street for each direction. Thus there are eight traffic lights, four for left turn on each street for each direction and six sensors to control these lights.

Logic of The Traffic Light Controller

The logic of the traffic light controller is very simple. All it has to do is switch lights according to the low or high signal coming from sensors. The initial state of the control is that the straight lights on the main street are always green unless there is a car on the intersecting street trying to straight or turning left. So if there is a car going straight on the intersecting street, then the straight lights on the main street will transition to yellow and then red after
waiting for forty seconds and the straight lights on the intersecting street for each direction will turn green while all other lights remain red. The after waiting for forty seconds the light straight light on the main street will turn green after the straight light on the intersecting street transitions from green to yellow and then to red. So, the same logic applies to other cases.

Before turning this logic into code it was ideal and helpful to have a state diagram describing each cases for the traffic light controller.

The state diagram above shows the general logic of the controller. So at S0 the only lights that are green are West Green Straight (WGST) lights and East Green Straight (EGST) lights. Note that this state diagram only shows the lights that are green, all other lights are red. In addition each state has sub-states that control the timing of the transitions. When a car stops on the intersecting street on a lane going straight, the sensor triggers and the counter starts counting. When the count reaches 40s, the state transfers to S1 and the lights on the main street transition from green to red and the lights going straight on the intersecting street become green thus it is noted as South Green Straight (SGST) and North Green Straight (NGST) in the state diagram. When the light hits green on the intersecting street, the counter refreshes and starts counting from zero. When the count reaches 40s the state transitions to S0. When
there are two cars on the intersecting street, one going straight and one turning left then the new state becomes S3. Here, cars going straight will be passed first and then the cars turning left will go after 40s, thus this phenomenon is noted as South Green Straight (SGST) and North Green Straight (NGST), then wait for 40s (40s) in the state diagram, then turn the south and north straight lights red and turn the south and north left turn lights green thus noted as South Green Left (SGL) and North Green Left (NGL) in the state diagram. So the rest of the logic is similar to what has been explained above.

**VHDL Code**

Two VHDL programs were used in this project. One named VGA_Control. This program is the output program that controls the screen. The second program called make_image is an input program to the VGA_Control program. In this project we have modified the make_image program provided to us by Dr. M. Takach.

Both VHDL codes are attached separately with the synthesis analysis provided by the RTL viewer of Quartus II.

**Control Switches**

Six switches on the Altera board act as sensors for this project. Here are the switch assignments:

Sensor on the west side of main street turning left – PIN_N26 (SW2)
Sensor on the east side of main street turning left – PIN_N25 (SW1)
Sensor on the north side of intersecting street going straight – PIN_V2 (SW15)
Sensor on the north side of intersecting street turning left – PIN_V1 (SW14)
Sensor on the south side of intersecting street going straight – PIN_U4 (SW17)
Sensor on the south side of intersecting street turning left – PIN_U3 (SW16)

Conclusions

In this project we have designed a traffic light controller using VHDL. The process was long and somewhat painful and stressful, however two weeks of work paid off at the end when we were successfully able to implement the state machine and output the results on a screen.