# Verifying and Debugging a Traffic Light Controller System

January 2010, Ver. 1.2

Application Note: an_tlcboard

## Objective

To test, verify and debug a Traffic Light Controller system (written in HDL) using CDLogic, referred in rest of the document as **TLC System**.

## Introduction

Following section gives you brief introduction to the traffic light controller system and guides you through the verification process and steps to be followed in order to debug the system on the shelf.

in this document you will learn:

- Overview of the Traffic Light Controller System
- Tools used
- Hardware and software set up
- Verifying the TLC system
- Debugging the system

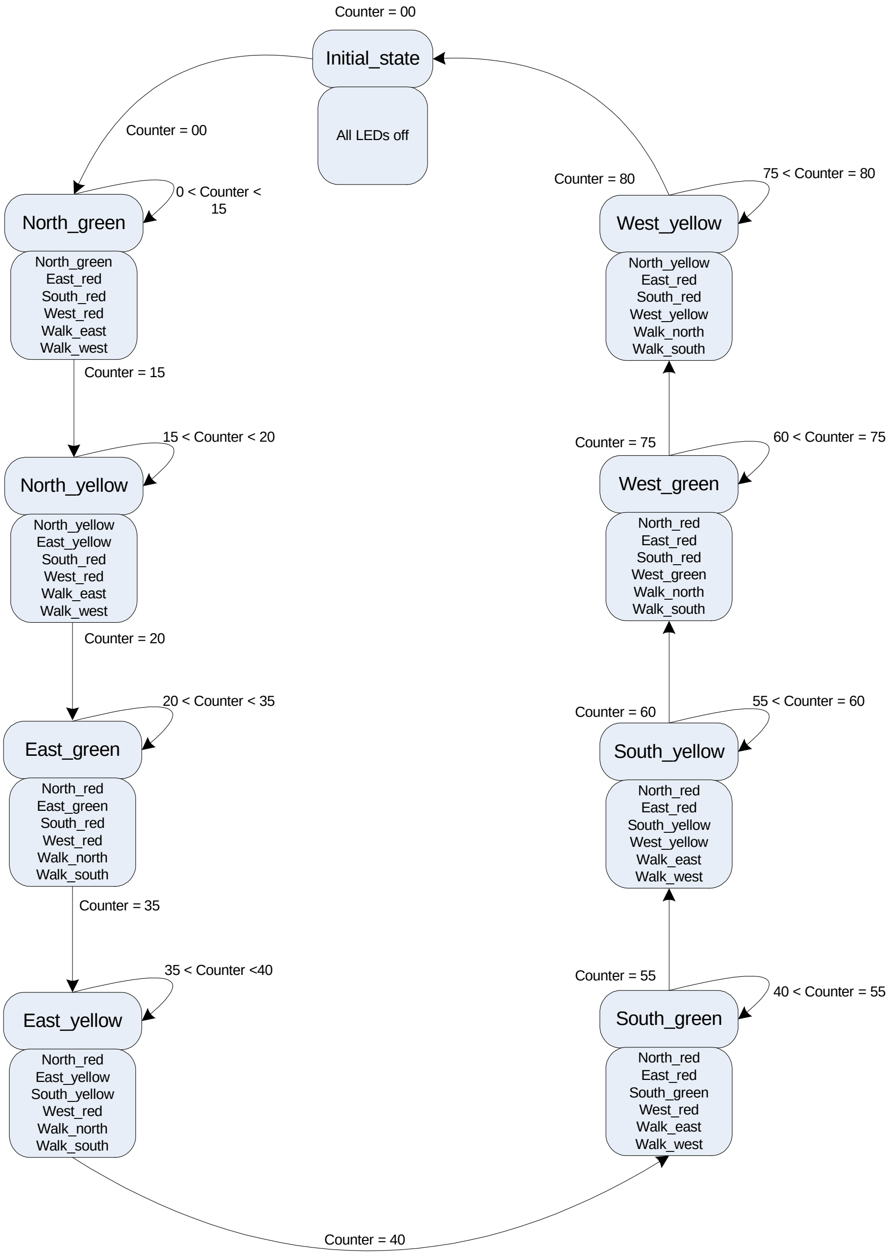
## TLC System Overview

The traffic light controller system is developed for the SLS Traffic Light Controller board.

The system controls the traffic with the help of a timer that calculates the time for the traffic to wait or go in each direction. The input frequency to the system is 48MHz. The total time for which the timer runs is 80 seconds, from which 20 seconds are used to display in each direction for timer display. Four switches are used to view the counter for the respective direction. When timer displays 20 on any direction display, the green light for the direction glows and the traffic is allowed to go till the timer reaches 15 seconds. At 15 seconds, the yellow LED glows and at 0, the red. Now the green LED for that direction will glow after 80 seconds.

The state machine diagram for the design is given in Figure 1.

*Figure 1. State Machine Diagram for the TLC system*



Refer the ***TrafficLightController.v*** file for TLC System code, and **ref_design** folder for Quartus II project in *<Traffic Light Controller Board Support Package Installation Path>*/**TLC_Reference_Designs/app_up3** folder.

☞ The two seven-segment displays on the board have shared data lines. Each of the segment is provided with an enable signal and depending on that the corresponding display takes the value on the data line.

The TLC code has error. In the next section, we will debug the code logic in the hardware using CDLogic.

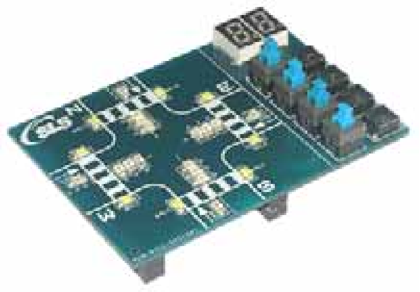
## Tools Used

The application note uses following tools in order to create, test and debug the traffic light controller system.

- SLS Traffic Light Controller Board

  The SLS Traffic Light Controller is a SC Nova board that is developed specifically for implementing traffic light control state machines. Figure 2 shows the SLS Traffic Light Controller Board.

*Figure 2. The SLS Traffic Light Controller Board*

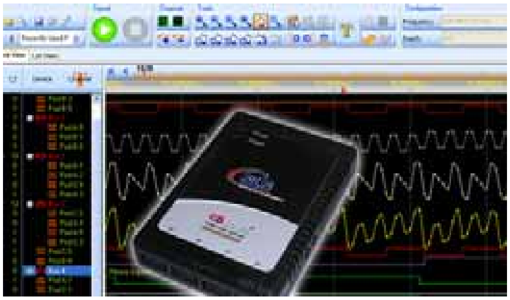


☞ For information about Traffic Light Controller Board, refer to http://www.slscorp.com/pages/sls_trafficlight_controller.php.

- CDLogic - The SLS Logic Analyzer

  CDLogic is a system-level debugging tool that captures and displays the signals in any digital circuit. The CDLogic is controlled with easy to use software that supports Windows. Figure 3 shows the SLS CDLogic, Logic Analyzer.

*Figure 3. CDLogic - The SLS Logic Analyzer*

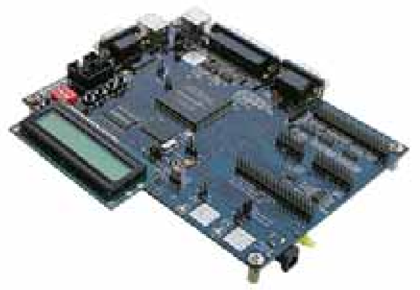


☞ For more information about CDLogic refer to http://www.slscorp.com/pages/cdlogicsls.php.

- UP3 Education Kit

  The UP3 Education kit provides a solution for prototyping and rapid development of the products. The board has a Altera Cyclone FPGA. Figure 4 shows the UP3 Education cum development board.

*Figure 4. The SLS UP3 Board*



☞ For more information about UP3 educational development board, refer to http://www.slscorp.com/pages/esdksls.php.

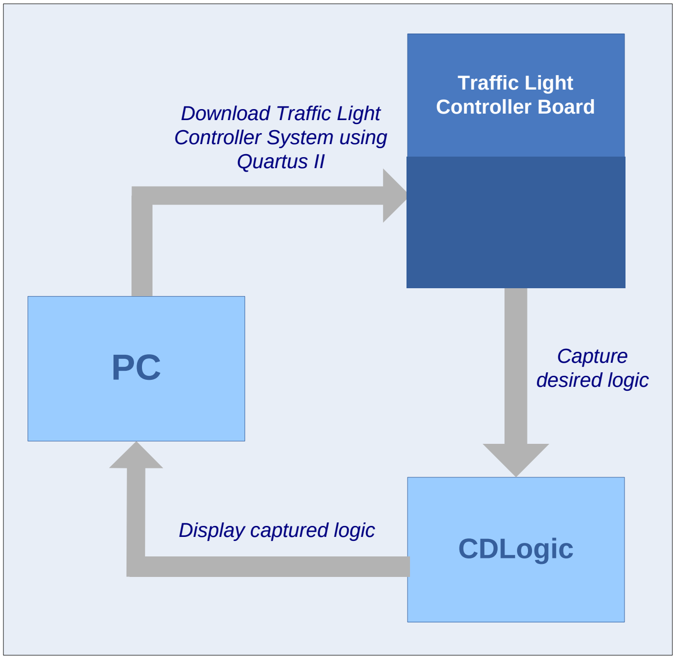
# Verification Set up

The section below explain the hardware and software set up for verification of the Traffic Light Controller system on the hardware.

## Hardware Set up

We have used the UP3 board to verify the functionality of our Traffic Light Controller system. We will bring out the signals to debug on the expansion headers of the UP3 board and will use it for debugging. The Figure 5 below shows the overview of the hardware set up.

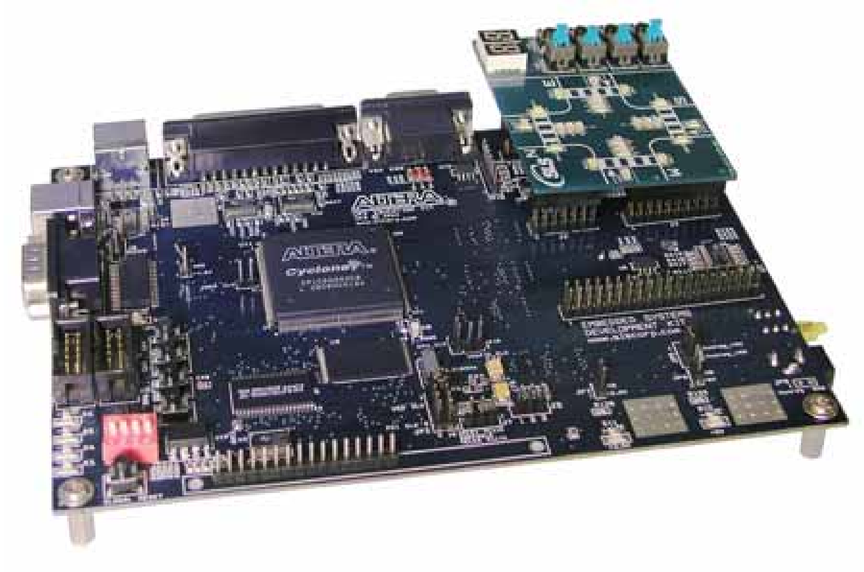
*Figure 5. Hardware Set up Overview*


Traffic Light Controller Board
Download Traffic Light Controller System using Quartus II
PC
Capture desired logic
Display captured logic
CDLogic

Please follow the steps mentioned below to set up the hardware for debugging the Traffic Light Controller System using UP3 and CDLogic.

1. Connect the traffic light controller board with the expansion headers on the UP3 board as shown in Figure 6.

*Figure 6. Connection of Traffic Light Controller Board with UP3 Board*



2. Connect the CDLogic cable assembly to the POD terminals on the CDLogic hardware. We require maximum 18 channels to debug; therefore, we shall connect only 3 PoD terminals.

3. As per the pin assignments on the expansion header, connect the following signals to CDLogic channels.
   - Seg_Data [6:0]: Data displayed on seven segment display
   - En_Disp [1:0]: Seven segment display selection signals
   - Pb_north: Push button switch for north
   - Pb_east: Push button switch for east
   - Pb_south: Push button switch for south
   - Pb_west: Push button switch for west
   - Disp_1_Data [7:0]: Internal counter signals
   - reset_n: Reset signal

4. Table 1 below for pin assignments and connect the CDLogic channels on the J1 header on UP3 board accordingly.

*Table 1. Pin Assignments for CDLogic Debug Signals*

| TLC System Signal Name | UP3 Header Pin Number | UP3 FPGA Pin Number | CDLogic Channel Number |
|---|---|---|---|
| reset_n | J1.3 | PIN_138 | D0_POD_A1 |
| Disp_1_Data [0] | J1.4 | PIN_139 | D0_POD_A2 |
| Disp_1_Data [1] | J1.5 | PIN_140 | D0_POD_A3 |
| Disp_1_Data [2] | J1.6 | PIN_141 | D0_POD_A4 |
| Disp_1_Data [3] | J1.7 | PIN_143 | D0_POD_A5 |
| Disp_1_Data [4] | J1.8 | PIN_156 | D0_POD_A6 |
| Disp_1_Data [5] | J1.9 | PIN_158 | D0_POD_A7 |
| Disp_1_Data [6] | J1.10 | PIN_159 | D0_POD_A8 |
| Disp_1_Data [7] | J1.11 | PIN_160 | D0_POD_A9 |
| En_Disp [0] | J1.12 | PIN_161 | D0_POD_B1 |
| En_Disp [1] | J1.15 | PIN_162 | D0_POD_B2 |
| Pb_north | J1.16 | PIN_163 | D0_POD_B3 |
| Pb_east | J1.17 | PIN_164 | D0_POD_B4 |
| Pb_south | J1.18 | PIN_165 | D0_POD_B5 |
| Pb_west | J1.19 | PIN_166 | D0_POD_B6 |
| Seg_Data [0] | J1.20 | PIN_167 | D0_POD_B7 |
| Seg_Data [1] | J1.21 | PIN_168 | D0_POD_B8 |
| Seg_Data [2] | J1.22 | PIN_169 | D0_POD_B9 |
| Seg_Data [3] | J1.23 | PIN_180 | D0_POD_C1 |
| Seg_Data [4] | J1.24 | PIN_181 | D0_POD_C2 |
| Seg_Data [5] | J1.25 | PIN_175 | D0_POD_C3 |
| Seg_Data [6] | J1.26 | PIN_177 | D0_POD_C4 |

☞ Make sure that you have connected proper ground channels.

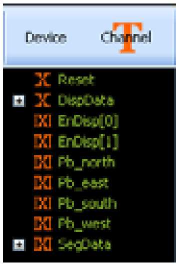
Now, we have completed the hardware set up and we will now set up the CDLogic software for test.

## Software Set up

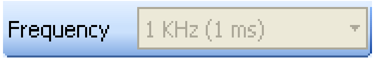
Please follow the steps mentioned below for setting up the CDLogic software to debug this application.

1. Open **CDLogic** software.
2. We will rename the channels as per the connections in the Table 1. Refer to the **setup.smp** file provided for the software set up. We can hide the remaining unused channels to get a proper set up.
3. Select **Edit->Channel->Hide** to hide the selected channels, alternatively you can hide the channel by pressing **Ctrl+h**. To view them back, select **Edit->Channel->Unhide channels** or press **Ctrl+A** to show all the channels. See Figure 7.

*Figure 7. setup.smp file*



4. We need to capture at least 100 or 90 seconds of data (80 seconds for one timer cycle and other 20 or 10 seconds for viewing the next cycle beginning). If we set the sampling rate as **100 MHz** (default), then our memory **128Kbits** will get filled up in **0.128** seconds only. Therefore,



Frequency 1 KHz (1 ms)

we shall set the sampling rate to **1KHz** so that we can capture the data for entire **128** seconds.

5. Now set the **trigger** on channel **reset**.
6. Define bus for **DispData[7:0]** and **SegData[6:0]**. Set the representation format for DispData as **Hexadecimal** and that for SegData as **Mnemonic**. Load the mnemonic file **svnseg.mne** to load the appropriate mnemonics.

This completes our CDLogic software set up and we are now all set to start the verification and debug.

## Capture and Verify the data

For verification, we will give different input conditions and observe the results.

### Input Condition 1

Capture the data after giving **reset** and without pressing any of the **direction switches** for 128 seconds. See ***capture1.smp*** file.

#### *Result*

Observe the **LED states** change on the snap on board while the capture is ongoing. The seven segment should display **00**. In the captured data, you will be able to see the reset pulse, while all other channels are blank.

### Input Condition 2

Capture the data for **128 seconds** after giving reset and pressing **North** push button switch simultaneously. Make sure that you release the **North** push button switch after leaving the reset switch. See ***capture_north.smp*** file.

#### *Result*

Observe the **LED states** and the **timer** on the seven segment display on the snap on board while the capture is ongoing. Scroll and verify the counter and seven segment truth table. You will be able to see the counter data (starting from 14) and timer data (starting from 20) in sync. Verify the counter in hex and converted data in decimal visually.

### Input Condition 3

Capture the data for **128 seconds** after giving the reset and pressing **East** push button switch simultaneously. Make sure that you release the **East** push button switch after leaving the reset switch. See ***capture_east.smp*** file.

#### *Result*

Observe the **LED states** and the **timer** on the seven segment display on the snap on board while the capture is ongoing. Scroll and verify the counter and seven segment truth table. You will be able to see the counter data (starting from 50) and timer data (starting fro 80) in sync. Verify the counter in hex and converted data in decimal visually.

## Input Condition 4

Capture the data for **128 seconds** after giving the reset and pressing **South** push button switch simultaneously. Make sure that you release the **South** push button switch after leaving the reset switch. See ***capture_south.smp*** file.

### *Result*

Observe the **LED states** and the **timer** on the seven segment display on the snap on board while the capture is ongoing. Scroll and verify the counter and seven segment truth table. You will be able to see the counter data (starting from 3C) and timer data (starting fro 60) in sync. Verify the counter in hex and converted data in decimal visually.

## Input Condition 5

Capture the data for **128 seconds** after giving the reset and pressing **West** push button switch simultaneously. Make sure that you release the **West** push button switch after leaving the reset switch. See ***capture_west.smp*** file.

### *Result*

Observe the LED states and the timer on the seven segment display on the snap on board while the capture is ongoing. Scroll and verify the counter and seven segment truth table. You will be able to see the counter data (starting from 28) and timer data (starting fro 40) in sync. Verify the counter in hex and converted data in decimal visually.

☞ Rrefer *<Traffic Light Controller Board Support Package>/* **TLC_Reference_Designs/app_up3/cdlogic_files** folder for captured file of all direction by CDLogic.

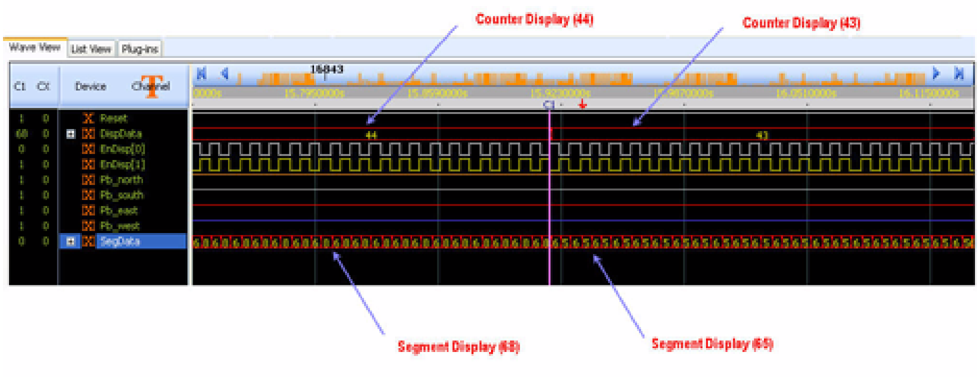
# Debug with CDLogic

## Example Problem 1

During verification of the data for all the input conditions, we have observed that in all the three sample files, the sequence of display after 68 remains as 68, 65, 66, 66, 64, 63, 62. instead of 68, 67, 66, 65, 64, 63, 62 for all the direction switches i.e. in all the four files.

Figure 8 below shows the error in ***capture_east.smp*** file.

*Figure 8. Error display in capture_east.smp file*



To debug the problem, perform following steps:

1. Check the counter for the above mentioned values, whether it is changing values or not.
2. If there is any error in the counter, means there is something wrong in the program logic.
3. If the counter is alright, then there is some error in hex to decimal conversion look up table. Check the table for the specific values i.e. 67. and 65.

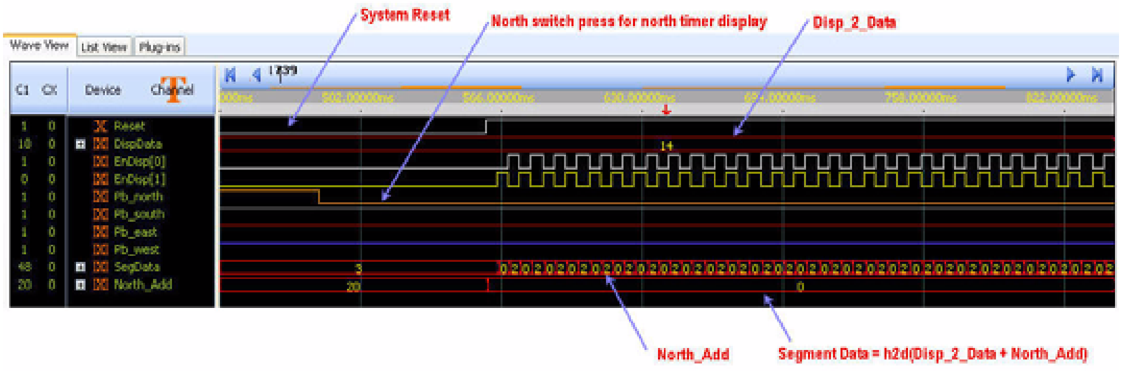
## Example Problem 2

Second problem, we have observed is that the counter and timer began from 40 instead of 60 for south whereas the counter and the timer began from 60 instead of 40 for west.

To debug this problem, follow the steps mentioned below:

1. The counter signal is made up of **Disp_2_Data** and respective **direction variables.** We will bring out the signals i.e. **Disp_2_Data** on counter channels and direction variables i.e. **North_Add**, **East_Add**, **South_Add** and **West_Add** (one by one) on additional 8 channels and name them as **DirData**. See ***North_Add.smp*** file. The Figure 9 below shows the set up.

*Figure 9. Debug set up in CDLogic (showing North_Add.smp)*



2. From the **Quartus II project**, make the changes accordingly in the code so that we get those channels on **J1** expansion header and **connect** the CDLogic cable assembly at respective pin.

3. Assign **North_Add** signals and capture the data by pressing the **north push button switch**. Record the signal changes at every 20 seconds.

4. Similarly repeat the steps 2 and 3 for **East_Add**, **South_Add** and **West_Add**. See files ***North_Add.smp***, ***East_Add.smp***, ***South_Add.smp*** and ***West_Add.smp***.

5. The recorded data is listed in the Table 2 below:

*Table 2. Recorded Data*

| North_Add | East_Add | South_Add | West_Add |
|---|---|---|---|
| 0 | 20 | 60 | 40 |
| 60 | 0 | 20 | 40 |
| 40 | 60 | 0 | 20 |
| 20 | 40 | 60 | 0 |
| 0 | 20 | 60 | 40 |
| 60 | 0 | 20 | 40 |
| 40 | 60 | 0 | 20 |

The time difference should remain **'20'** horizontally and vertically. The places where the rule is violated are marked in **Red** color. Now you can go through the code and spot the error at once.

Refer ***TrafficLightController_modified.v*** file for errors corrected at *<Traffic Light Controller Board Support Package Installation Path>/* **TLC_Reference_Designs/app_up3/hdl** folder.

## Conclusion

Using CDLogic, it's easy to debug the Traffic Light Controller counter by viewing and re-viewing the display data without wasting time. Internal signals can be brought out any time for analyzing design logic. Measurement and error marking is now convenient with bookmarks and cursors in CDLogic software.

## Further Information

- The CDLogic - The Logic Analyzer software is available at http://www.slscorp.com/pages/demodownload.php.
- The documents and Quartus II projects for traffic light controller is available at http://www.slscorp.com/pages/snap_on_board_packages.php.

## Revision History

The table below shows the revision history.

| Version | Date | Description |
|---|---|---|
| 1.0 | September 2007 | Initial Release |
| 1.1 | September 2007 | Added CDLogic Screen shot, State Machine figure, objective and conclusion |
| 1.2 | January 2010 | Update the link of the pages in Further Information section |

**System Level Solutions, Inc.**
**(USA) 14100 Murphy  Avenue**
**San Martin, CA 95046**
**(408) 852 - 0067**

**http://www.slscorp.com**

an_tlcboard_1.2