

Regulation and Control of City Traffic System using FPGA

A PROJECT REPORT

*Submitted in partial fulfillment of the
requirement for the award of the
Degree of*

**BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION ENGINEERING**

By

**MAYANK PRASAD
(09BEC233)**

**SANGEET SAURABH
(09BEC253)**

Under the Guidance of

Prof. K. SIVASANKARAN



**SCHOOL OF ELECTRONICS ENGINEERING
VIT University
VELLORE. (TN) 632014**

(MAY 2013)

CERTIFICATE

This is to certify that the Project work titled "*Regulation and control of City Traffic Using FPGA*" that is being submitted by "*Mayank Prasad and Sangeet Saurabh*" is in partial fulfillment of the requirements for the award of **Bachelor of Technology**, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Prof. K. SIVASANKARAN
Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner

External Examiner

Approved by

Program Chair

This project is dedicated to all those Indians who are trapped in city traffic, inspiring us to take up this as a challenge and work out a solution for this.

ACKNOWLEDGEMENTS

The project started with the simple idea or rather a discussion on the present scenario of the traffic problem. It would be unfair if we would not be acknowledging the traffic police of the India especially Bangalore Traffic Police which actually inspires a lot out of us. We also acknowledge the contribution of our review panel members and our friends who helped us while debating on the reality of the project whenever needed and their critical suggestion, which helped to make the complex topic into a simpler version. We also thank Mr. Tanoj Kumar for helping us out with the hardware implementation of FPGA. Ultimately, we would like to express our heartfelt thanks to our guide, Prof. K. Sivansankaran for his never ending support and guidance, which led to the completion of this project.

**MAYANK PRASAD
(09BEC233)**

**SANGEET SAURABH
(09BEC253)**

ABSTRACT

The objective of this project is to develop a new and better self-adaptive system to regulate and control the city traffic. The new cost effective system will replace the old traditional traffic control system. This will result in synchronized traffic, and henceforth reduce the number of road accidents.

Traffic light controller establishes a set of rules and instructions that drivers, pilots, train engineers, and ship captains rely on to avoid collisions and other hazards. Traffic Control systems include signs, lights and other devices that communicate specific directions, warnings, or requirements. Traffic light controller (TLC) has been implemented using microcontroller, FPGA, and ASIC design. FPGA has many advantages over microcontroller, some of these advantages are: the speed, number of input/output ports and performance which are all very important in TLC design, at the same time ASIC design is more expensive than FPGA. Most of the TLCs implemented on FPGA are simple ones that have been implemented as examples of FSM. But in this project, our main aim is to synchronize all the TLCs in a particular region of a city. All the TLCs of a particular geographical area will be linked together to a single FPGA, where an algorithm will synchronize them all together. The algorithm will depend upon the real-time traffic intensity.

As of now, the Single Node Model with four road junctions has been taken and a new algorithm has been implemented where the priority of traffic lights is determined by the traffic intensity of the roads.

LIST OF FIGURES

Figure	Title	Page No.
2.1	The DE1 Board	4
2.2	Block Diagram of DE1 Board	6
2.3	Illustrates the JTAG Configuration Setup	10
2.4	Switch Debouncing	11
2.5	Schematic diagram of the pushbutton and toggle switch	12
2.6	Schematic diagram of the LEDs	12
2.7	Schematic diagram of the clock circuit	13
2.8	Schematic diagram of the expansion header	14
2.9	Illustration showing the basic working of SCOOT System	18
2.10	Collection of data by SCOOT System	18
3.1	Single Node Traffic Model	25
3.2	Simplified Block Diagram	27
4.1	Simplified State Diagram	29
4.2	Single Node Model	31
4.3	Self-Soldered PCB for Traffic Light	35
4.4	Proximity IR Sensor	36
4.5	Experimental Setup	36
5.1	Complete Synthesized RTL Netlist (Part 1 of 2)	40
5.2	Complete Synthesized RTL Netlist (Part 2 of 2)	41
5.3	RTL Representation of State and Next State	41
5.4	An instance of comparison of different traffic intensities	42
5.5	Experimental Setup - Top View	43
5.6	Experimental Setup - Side View	43

TABLE OF CONTENTS

Chapter	Title	Page No.
1	Introduction	1
	1.1 Objective and Motivation	1
	1.2 Overview of Traffic Light Controller	1
	1.3 Traffic Lights and Timing Control	3
2	Background Theory	4
	2.1 Altera DE1 Board – Layout and Components	4
	2.2 Block Diagram of DE1 Board	6
	2.3 Using the DE1 Board and Configuring the Components	9
	2.4 SCOOT System	15
	2.5 Sensors for Vehicle Detection	19
3	Project Design	25
	3.1 Single Node Traffic Light Controller	25
	3.2 Project Description	27
	3.3 Design Methodology	28
4	Design Implementation	29
	4.1 State Diagram	29
	4.2 Simplified Algorithm of Single Node Model	33
	4.3 Detailed Functional Description	34
	4.4 Prototype Deployment	35
5	Result and Analysis	37
	5.1 Functional Verification	37
	5.2 Synthesis Outcome	39
	5.3 Experimental Results	43
6	Conclusion	44
	6.1 Conclusion	44
	6.2 Design Features	44
	6.3 Social Outcome	45
7	References	46

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE AND MOTIVATION

The objective of this project is to develop a new and better self-adaptive system to regulate and control the city traffic. The new cost effective system will replace the old traditional traffic control system. This will result in synchronized traffic, and henceforth reduce the number of road accidents.

In the current scenario, the traffic in several cities of India is unorganized, which results in greater number of accidents, unwanted wastage of time, money and resources, frustration, unconventional lifestyle, pollution, etc. So we have decided to tackle this social problem and come up with a cost effective solution to this.

1.2 OVERVIEW OF TRAFFIC LIGHT SYSTEM

Ever since Roman times, society has tried to control traffic. Even the fabled roman road system created a conflict between pedestrian and equine travelers. However, a practical solution was not developed until the mid-nineteenth century. On 10 December 1868, the first traffic lights were installed outside the British Houses of Parliament in London, to control the traffic in Bridge Street, Great George Street and Parliament Street. They were promoted by the railway engineer J. P. Knight and constructed by the railway signal engineers of Saxby & Farmer. The design combined three semaphore arms with red and green gas lamps for night-time use, on a pillar, operated by a police constable. The gas lantern was turned with a lever at its base so that the appropriate light faced traffic.^[9]

Although it was said to be successful at controlling traffic, its operational life was brief. It exploded on 2 January 1869, as a result of a leak in one of the gas lines underneath the pavement, killing the policeman who was operating it. With doubts about its safety, the concept was abandoned until electric signals became available.^[13]

The first electric traffic light was developed in 1912 by Lester Wire, an American policeman of Salt Lake City, Utah, who also used red-green lights. On 5 August 1914, the American Traffic Signal Company installed a traffic signal system on the corner of East 105th Street and Euclid Avenue in Cleveland, Ohio. It had two colors, red and green, and a buzzer, based on the design of James Hoge, to provide a warning for color changes. The design by James Hoge allowed police and fire stations to control the signals in case of emergency. The first four-way, three-color traffic light was created by police officer William Potts in Detroit, Michigan in 1920. In 1922, T.E. Hayes patented his "Combination traffic guide and traffic regulating signal". Ashville, Ohio claims to be the location of the oldest working traffic light in the United States, used at an intersection of public roads until 1982 when it was moved to a local museum.^[13]

The first interconnected traffic signal system was installed in Salt Lake City in 1917, with six connected intersections controlled simultaneously from a manual switch. Automatic control of interconnected traffic lights was introduced March 1922 in Houston, Texas. These basic designs were soon improved. In 1926 the first automatic signals were installed in London; they depended on a timer to activate them. The first automatic experimental traffic lights in England were deployed in Wolverhampton in 1927. In 1923, Garrett Morgan patented his own version. The Morgan traffic signal was a T-shaped pole unit that featured three hand-cranked positions: stop, go, and an all-directional stop position. This third position halted traffic in all directions to give drivers more time to stop before opposing traffic started. It's one "advantage" over others of its type was the ability to operate it from a distance using a mechanical linkage. In the 1930s vehicle-activated lights were created in which cars rolled over half-buried rubber tubes. Air in the tubes was displaced by the weight of the car rolling over them, and the increased pressure operated an electric contact, activating the lights. But these tubes wore out quickly. A better idea was the inductive-loop device: a loop of wire was imbedded in the road itself and connected to a box controlling the lights; a current of electricity passed through the loop, and when the steel body of a car passed overhead, it produced a signal that activated the light. Toronto was the first city to computerize its entire traffic signal system, which it accomplished in 1963.^[13]

1.3 TRAFFIC LIGHTS AND TIMING CONTROL

In 1926 the first automatic signals were installed in London; they depended on a timer to activate them. Since the introduction of the timer there was always a need to improvise the timer effectively based on the observation. With the advent of the new technology though the life of the people was changing very rapidly and so was there life style. Growth in the automobile industries during industrial era increases the need for better and effective system. Traffic problem seriousness was arousing the scientist as well as the common man for better methodology. The larger the city, the more serious and complex a problem we will face. Moreover, the longer we let the problem go unsolved for longer and longer, the problem will become more and more serious.

Part of the traffic problem is congestion at intersections that is caused by various factors. One important factor that impacts on traffic at intersections is the length of each phase in the cycle of the traffic signal. It may not be appropriate and may not be suitable for traffic pattern parameters such as volume of vehicles, queue length, delay, speed and so on. It is a worldwide problem.

One reason for traffic jams is that traffic signal timing is often not suitable for traffic control at the intersection in real time. So the concerned traffic office needs to optimize traffic signal timing to solve the traffic congestion at intersections. Engineers behind the federally funded Traffic Signal System Improvement Program in Denver (Hsiao-Ching & Denver, 1998) have worked over the past 10 years to ease metro-area traffic congestion by coordinating and adjusting the timing of traffic signals on major streets. There are many papers that propose methods to improve traffic signal timing. All of the methods use a similar process, based on observed traffic data input at intersections, such as volume, pattern of traffic, number of cars going straight or turning right, delay, queue length, speed, density, and so on.

CHAPTER 2

BACKGROUND THEORY

2.1 ALTERA DE1 BOARD – LAYOUT AND COMPONENTS

A photograph of the DE1 board is shown in Figure 2.1. It depicts the layout of the board and indicates the connectors and key components.^[4]

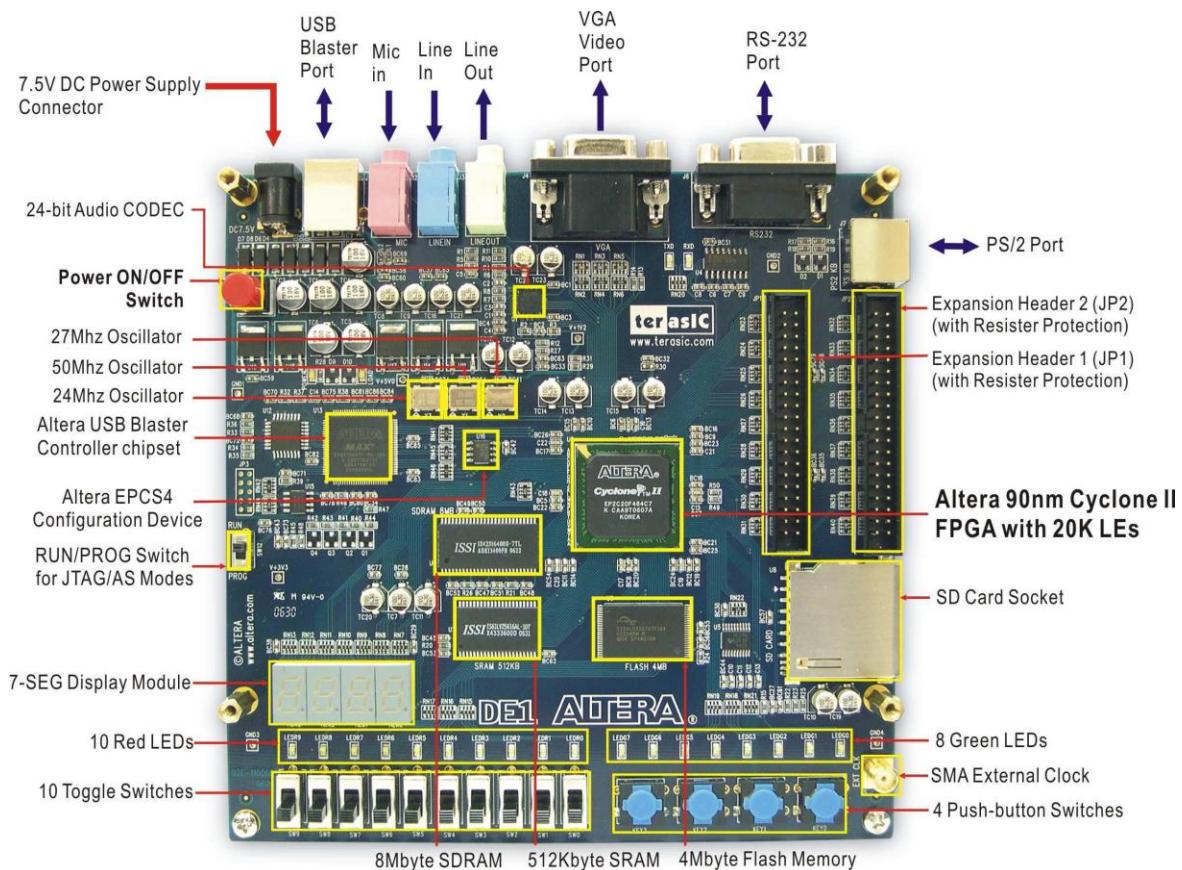


Figure 2.1: The DE1 Board

The DE1 board has many features that allow the user to implement a wide range of designed circuits from simple circuits to various multimedia projects.

The following hardware is provided on the DE1 board:^[4]

- Altera Cyclone II EP2C20F484C FPGA device
- Altera Serial Configuration device – EPICS4
- USB Blaster (on board) for programming and user API control; both JTAG and Active Serial (AS) programming modes are supported
- 512-Kbyte SRAM
- 8-Mbyte SDRAM
- 4-Mbyte Flash memory
- SD Card socket
- 4 pushbutton switches
- 10 toggle switches
- 10 red user LEDs
- 8 green user LEDs
- 50-MHz oscillator, 27-MHz oscillator and 24-MHz oscillator for clock sources.
- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
- VGA DAC (4-bit resistor network) with VGA-out connector
- RS-232 transceiver and 9-pin connector
- PS/2 mouse/keyboard connector
- Two 40-pin Expansion Headers with resistor protection
- Powered by either a 7.5V DC adapter or a USB cable.

In addition to these hardware features, the DE1 board has software support for the standard I/O interface and a control panel facility for accessing various components.

2.2 BLOCK DIAGRAM OF THE DE1 BOARD

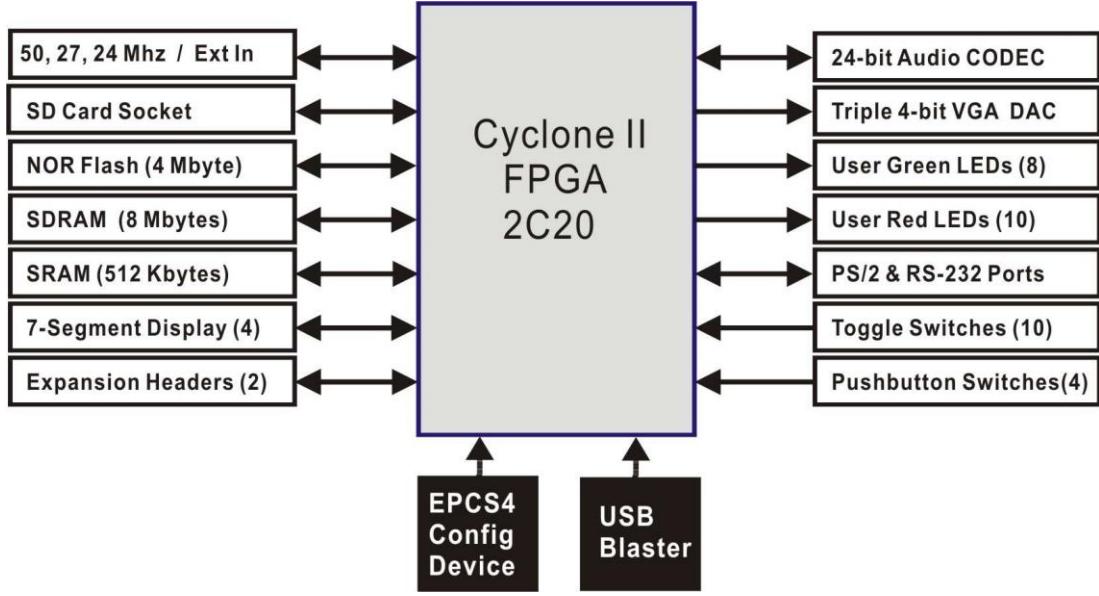


Figure 2.2: Block Diagram of DE1 Board

To provide maximum flexibility for the user , all connections are made through the cyclone II FPGA device. Thus, the user can configure the FPGA to implement any system design.^[4]

Following is more detailed information about the blocks in Figure 2.2:

Cyclone II 2C20 FPGA

- 18,752 LEs
- 52 M4K RAM blocks
- 240K total RAM bits
- 26 embedded multipliers
- 4 PLLs
- 315 user I/O pins
- FineLine BGA 484-pin package

Serial Configuration Device and USB Blaster Circuit

- Altera's EPCS4 Serial Configuration device
- On-board USB Blaster for programming and user API control
- JTAG and AS programming modes are supported.

SRAM

- 512-Kbyte Static RAM memory chip
- Organized as 256K x 16 bits
- Accessible as memory for the Nios II processor and by the DE1 Control Panel.

SDRAM

- 8-Mbyte Single Data Rate Synchronous Dynamic RAM memory chip.
- Organized as 1M x 16 bits x 4 banks.
- Accessible as memory for the Nios II processor and by the DE1 Control Panel.

Flash Memory

- 4-Mbyte NOR Flash memory.
- 8-bit data bus
- Accessible as memory for the Nios II processor and by the DE1 Control Panel

SD Card Socket

- Provides SPI mode for SD Card access
- Accessible as memory for the Nios II processor with the DE1 SD Card Driver

Push button Switches

- 4 pushbutton switches
- Debounced by a Schmitt trigger circuit
- Normally high; generates one active-low pulse when the switch is pressed

Toggle switches

- 10 toggle switches for user inputs
- A switch causes logic 0 when in the DOWN (closest to the edge of the DE1 board) position and logic 1 when in the UP position

Clock inputs

- 50-MHz oscillator
- 27-MHz oscillator
- 24-MHz oscillator
- SMA external clock input

Audio CODEC

- Wolfson WM8731 24-bit sigma-delta audio CODEC
- Line-level input, line-level output, and microphone input jacks
- Sampling frequency: 8 to 96 KHz
- Applications for MP3 players and recorders, PDAs, smart phones, voice recorders, etc.

VGA output

- Uses a 4-bit resistor-network DAC
- With 15-pin high-density D-sub connector
- Supports up to 640x480 at 60-Hz refresh rate
- Can be used with the Cyclone II FPGA to implement a high-performance TV Encoder

Serial ports

- One RS-232 port
- One PS/2 port
- DB-9 serial connector for the RS-232 port
- PS/2 connector for connecting a PS2 mouse or keyboard to the DE1 board

Two 40-pin expansion headers

- 72 Cyclone II I/O pins, as well as 8 power and ground lines, are brought out to two 40-pin expansion connectors
- 40-pin header is designed to accept a standard 40-pin ribbon cable used for IDE hard drives
- Resistor protection is provided.

2.3 USING THE DE1 BOARD AND CONFIGURING THE COMPONENTS

This section gives instructions for using the DE1 board and describes each of I/O devices.

2.3.1 *Configuring the Cyclone II FPGA*

The DE1 board contains a serial EEPROM chip that stores configuration data for the Cyclone II FPGA. This configuration data is automatically loaded from the EEPROM chip into the FPGA each time power is applied to the board. Using the Quartus® II software, it is possible to reprogram the FPGA at any time, and it is also possible to change the non-volatile data that is stored in the serial EEPROM chip. Both types of programming methods are described below.^[4]

2.3.1.1 JTAG Programming: In this method of programming, named after the IEEE standards *Joint Test Action Group*, the configuration bit stream is downloaded directly into the Cyclone II FPGA. The FPGA will retain this configuration as long as power is applied to the board; the configuration is lost when the power is turned off.

2.3.1.2 AS Programming: In this method, called *Active Serial* programming, the configuration bit stream is downloaded into the Altera EPCS16 serial EEPROM chip. It provides non-volatile storage of the bit stream, so that the information is retained even when the power supply to the DE1 board is turned off. When the board's power is turned on, the configuration data in the EPCS16 device is automatically loaded into the Cyclone II FPGA.

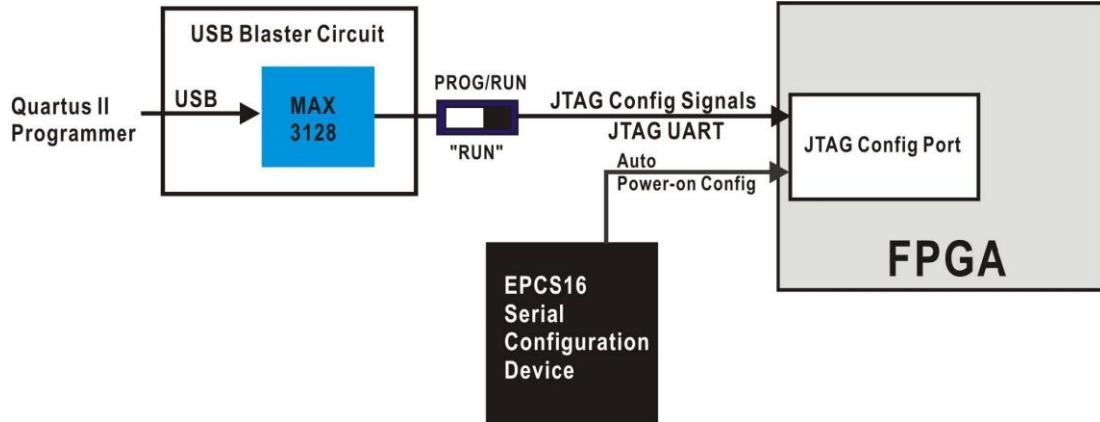


Figure 2.3: Illustrates the JTAG Configuration Setup

To download a configuration bit stream into the Cyclone II FPGA, perform the following steps:^[4]

- Ensure that power is applied to the DE1 board
- Connect the supplied USB cable to the USB Blaster port on the DE1 board (see Figure 2.1)
- Configure the JTAG programming circuit by setting the RUN/PROG switch (on the left side of the board) to the RUN position.
- The FPGA can now be programmed by using the Quartus II Programmer module to select a configuration bit stream file with the .sof filename extension.

2.3.2 Using the LEDs and Switches

The DE1 board provides four pushbutton switches. Each of these switches is debounced using a Schmitt Trigger circuit, as indicated in Figure 2.4. The four outputs called *KEY0*, ..., *KEY3* of the Schmitt Trigger device are connected directly to the Cyclone II FPGA. Each switch provides a high logic level (3.3 volts) when it is not pressed, and provides a low logic level (0 volts) when depressed. Since the pushbutton switches are debounced, they are appropriate for use as clock or reset inputs in a circuit.^[4]

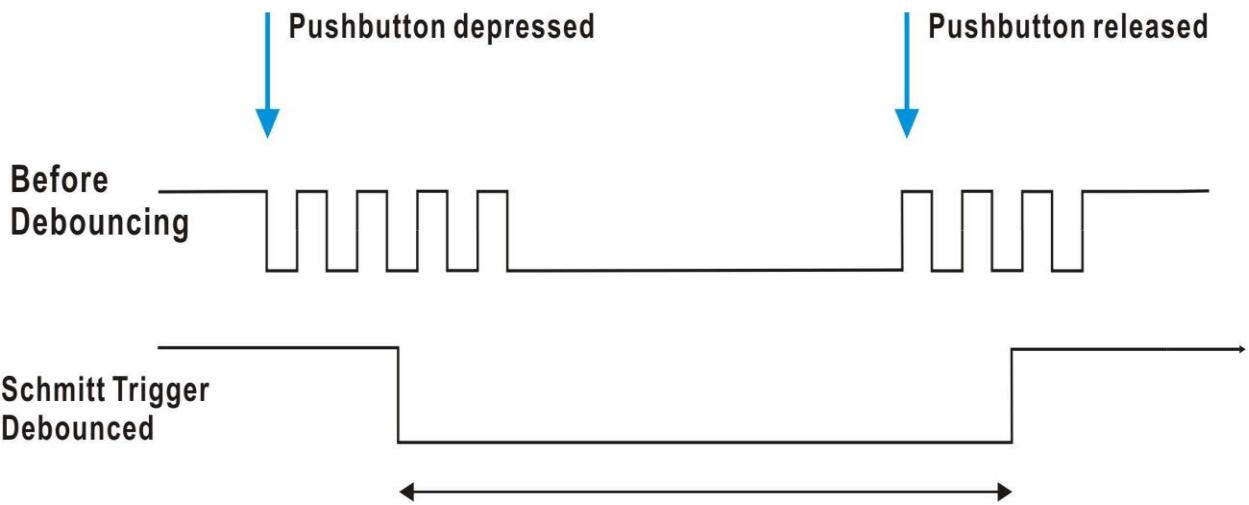


Figure 2.4: Switch Debouncing

There are also 10 toggle switches (sliders) on the DE1 board. These switches are not debounced, and are intended for use as level-sensitive data inputs to a circuit. Each switch is connected directly to a pin on the Cyclone II FPGA. When a switch is in the DOWN position (closest to the edge of the board) it provides a low logic level (0 volts) to the FPGA, and when the switch is in the UP position it provides a high logic level (3.3 volts).

There are 27 user-controllable LEDs on the DE1 board. Eighteen red LEDs are situated above the 18 toggle switches, and eight green LEDs are found above the pushbutton switches (the 9th green LED is in the middle of the 7-segment displays). Each LED is driven directly by a pin on the Cyclone II FPGA; driving its associated pin to a high logic level turns the LED on, and driving the pin low turns it off. A schematic diagram that shows the pushbutton and toggle switches is given in Figure 2.6. A schematic diagram that shows the LED circuitry appears in Figure 2.7

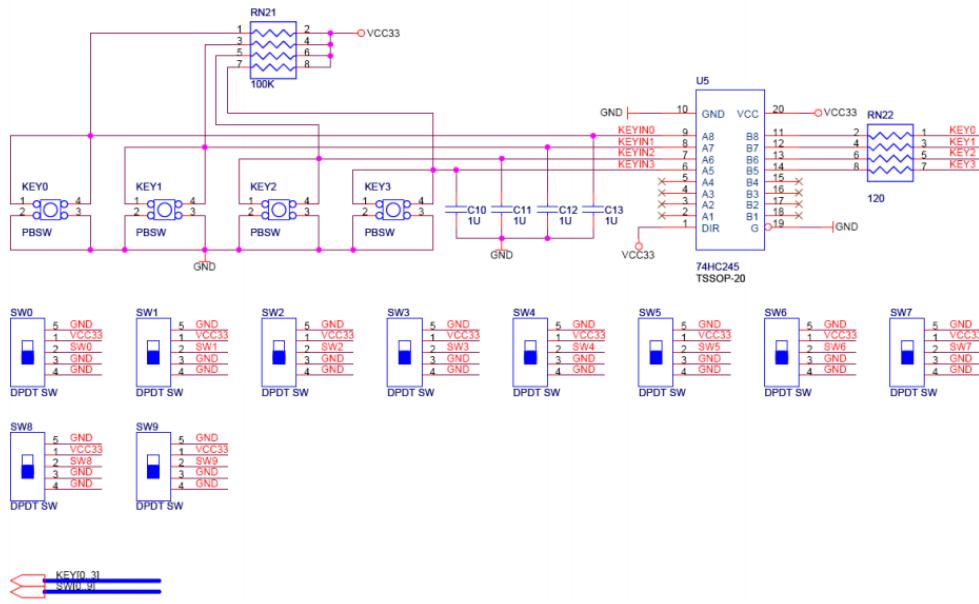


Figure 2.5: Schematic diagram of the pushbutton and toggle switch

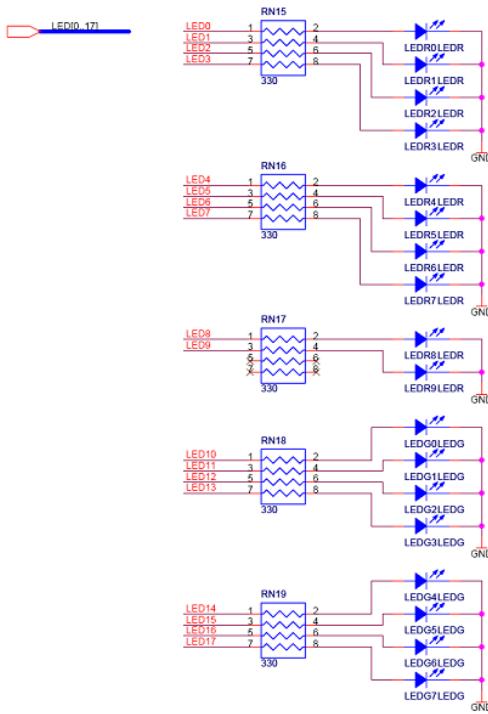


Figure 2.6: Schematic diagram of the LEDs

2.3.3 Clock Inputs

The DE1 board includes three oscillators that produce 27 MHz, 24Mhz, and 50 MHz clock signals. The board also includes an SMA connector which can be used to connect an external clock source to the board. The schematic of the clock circuitry is shown in Figure 2.7. [4]

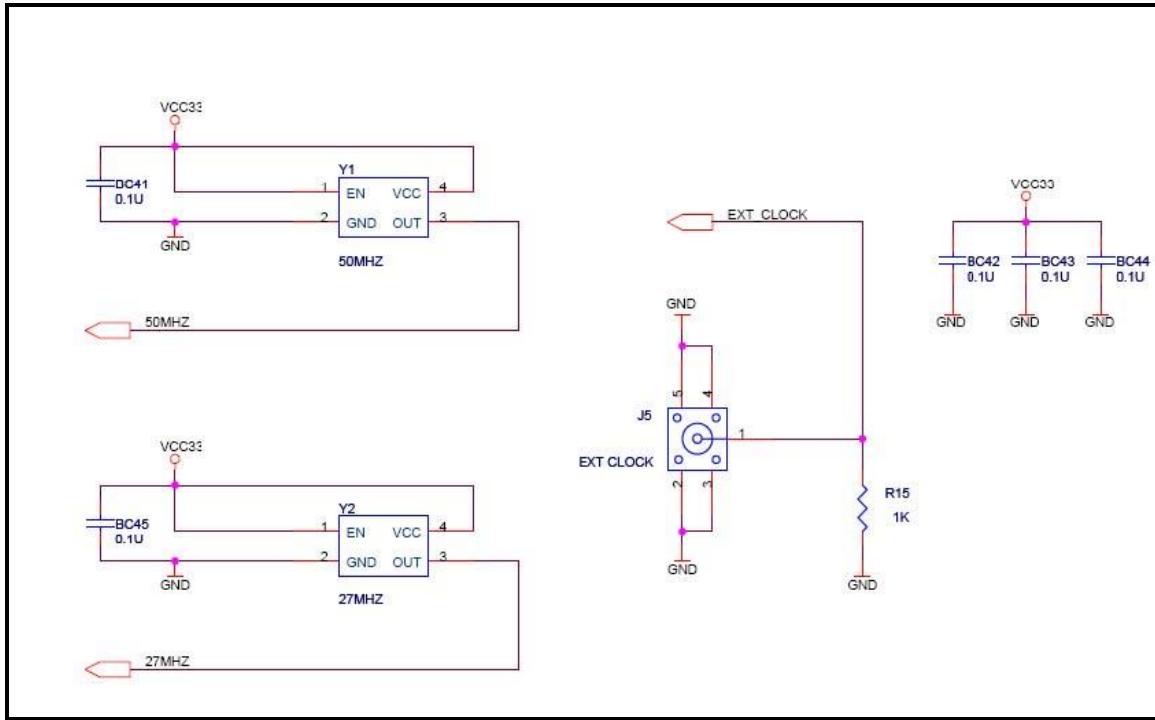


Figure 2.7: Schematic diagram of the clock circuit

2.3.4 Using the Expansion Header

The DE1 Board provides two 40-pin expansion headers. Each header connects directly to 36 pins on the Cyclone II FPGA, and also provides DC +5V (VCC5), DC +3.3V (VCC33), and two GND pins. Figure 2.8 shows the related schematics. Each pin on the expansion headers is connected to a resistor that provide protection from high and low voltages. [4]

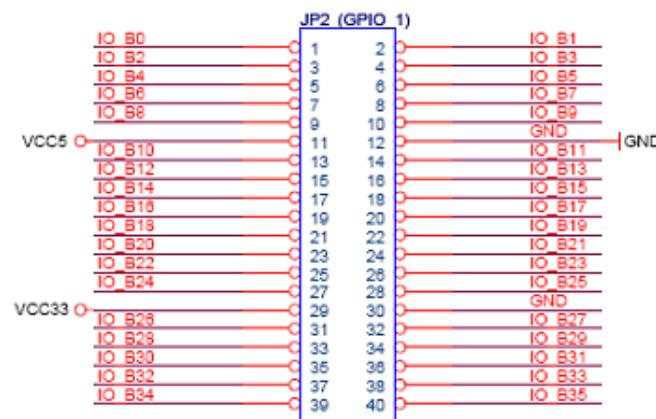
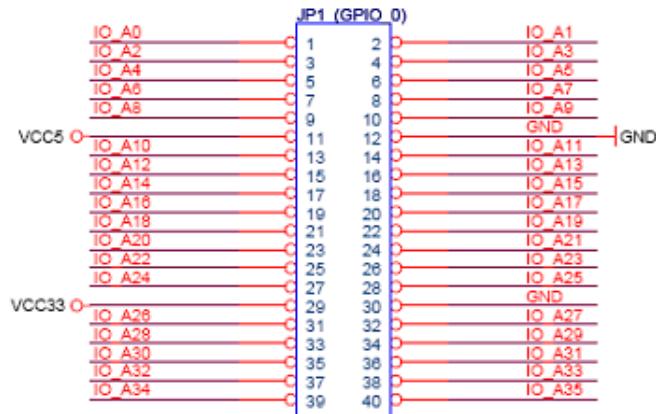


Figure 2.8: Schematic diagram of the expansion header

2.4 SCOOT SYSTEM

The Split Cycle Offset Optimization Technique (SCOOT) urban traffic control system was developed by the Transport Research Laboratory (TRL) in collaboration with the UK traffic systems industry. SCOOT is an adaptive system which responds automatically to traffic fluctuations. It does away with the need for signal plans which are expensive to prepare and keep up to date. SCOOT has proved to be an effective and efficient tool for managing traffic on signalized road networks and is now used in over 130 towns and cities in the UK and overseas. This leaflet is intended to draw the attention of highway authorities (in Scotland, roads authorities) to the advantages of SCOOT. Some authorities may not be aware of the benefits of installing SCOOT. Others which already have SCOOT systems may not be getting the best out of them or appreciate the benefits of extending or updating them. SCOOT has been improved in recent years, and research currently in hand by TRL and the SCOOT suppliers will deliver further improvements in the near future.^[8]

In urban areas where traffic signals are close together, the co-ordination of adjacent signals is important and gives great benefits to road users. Linking traffic signals along a single route so that vehicles get a green signal at each junction in turn is relatively simple. Coordinating signals over a network of conflicting routes is much more difficult. Computer techniques have been developed to calculate optimum signal settings for a signal network. TRANSYT, developed by TRL, is probably the best known example. TRANSYT can be used to compile a series of fixed time signal plans for different times of day or for special recurring traffic conditions. Preparing such signal plans requires traffic data to be collected and analyzed. This is time consuming and expensive, and the resulting plans should be updated regularly as traffic patterns change. To overcome these problems, the concept of a demand responsive UTC system was developed. The first generation of demand responsive systems monitored traffic flows continuously and triggered the most appropriate plan from the library. Second generation systems used current traffic counts to update historical data and produce new plans. However, this often led to frequent plan changing which caused disruption and often sub-optimal plan changes. In the late 1970s, TRL developed a methodology to overcome these problems.

An on line computer continuously monitored traffic flows over the whole network and made a series of frequent small adjustments to signal timings to reduce delays and improve traffic flow. This was the basis of SCOOT. [8]

2.4.1 Benefits of SCOOT

The benefits of SCOOT compared to alternative methods of control have been well documented. Journey time surveys in Worcester and Southampton found that SCOOT control reduced delays. [8]

2.4.2 Working of SCOOT

The kernel software at the heart of a SCOOT system is standard to all installations. The additional software (the "knitting" software) which links the SCOOT kernel to on street equipment and which provides the user interface is specific to the supplier. [8]

SCOOT sends out instructions to the "on street" equipment using dedicated telephone lines. These instructions are interpreted and acted upon by traffic signal equipment at the roadside. The equipment replies to the central computer confirming the acceptance of instruction, or detailing a fault condition. SCOOT obtains information on traffic flows from detectors. As an adaptive system, SCOOT depends on good traffic data so that it can respond to changes in flow. Detectors are normally required on every link. Their location is important and they are usually positioned at the upstream end of the approach link. Inductive loops are normally used, but other methods are being developed. When vehicles pass the detector, SCOOT converts the information into "link profile units" (lpu), a hybrid of link flow and occupancy. This is the unit used by SCOOT in its calculations. "Cyclic flow profiles" of lpu's over time are constructed for each link. [8]

A SCOOT network is divided into "regions", each containing a number of "nodes" (signaled junctions and pedestrian crossings which are all run at the same cycle time to allow co-ordination). Nodes may be "double cycled" (i.e. operate at half of the regional cycle time) at pedestrian crossings of under saturated junctions. Region boundaries are located where links are long enough for lack of Co-ordination not to matter. [8]

SCOOT has three optimization procedures by which it adjusts signal timings - the Split Optimizer, the Offset Optimizer, and the Cycle Time Optimizer. These give SCOOT its name - Split Cycle and Offset Optimization Technique. Each optimizer estimates the effect of a small incremental change in signal timings on the overall performance of the region's traffic signal network. A performance index is used, based on predictions of vehicle delays and stops on each link. The Split Optimizer works at every change of stage by analyzing the current red and green timings to determine whether the stage change time should be advanced retarded or remain the same. The Split Optimizer works in increments of 1 to 4 seconds. The Offset Optimizer works once per cycle for each node. It operates by analyzing the current situation at each node using the cyclic flow profiles predicted for each of the links with upstream or downstream nodes. It then assesses whether the existing action time should be advanced, retarded or remains the same in 4 second increments. The Cycle Time Optimizer operates on a region basis once every five minutes, or every two and a half minutes when cycle times are rising rapidly. It identifies the "critical node" within the region and will attempt to adjust the cycle time to maintain this node with a 90% link saturation on each stage. If it calculates that a change in cycle time is required, it can increase or decrease the cycle time in 4, 8 or 16 second increments.

By the combination of relatively small changes to traffic signal timings, SCOOT can respond to short term local peaks in traffic demand, as well as following trends over time and maintaining constant co-ordination of the signal network. The main motto behind the Scoot system can be clearly understood by the following figure :

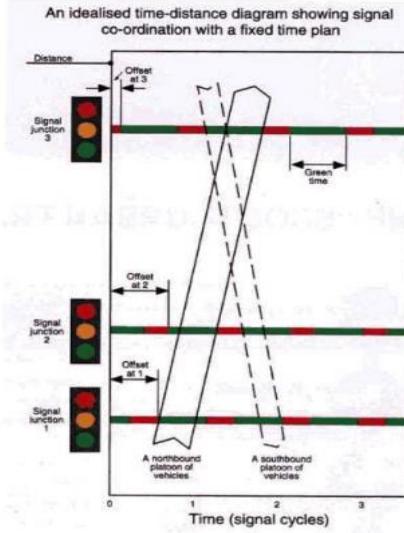


Figure 2.9: Illustration showing the basic working of SCOOT System

From the above figure it's quite certain that the main motive of the SCOOT system was to avail maximum movability of the traffic i.e. movement of the vehicle should be in such a way that probability of availability of the green light should be maximum. [8]

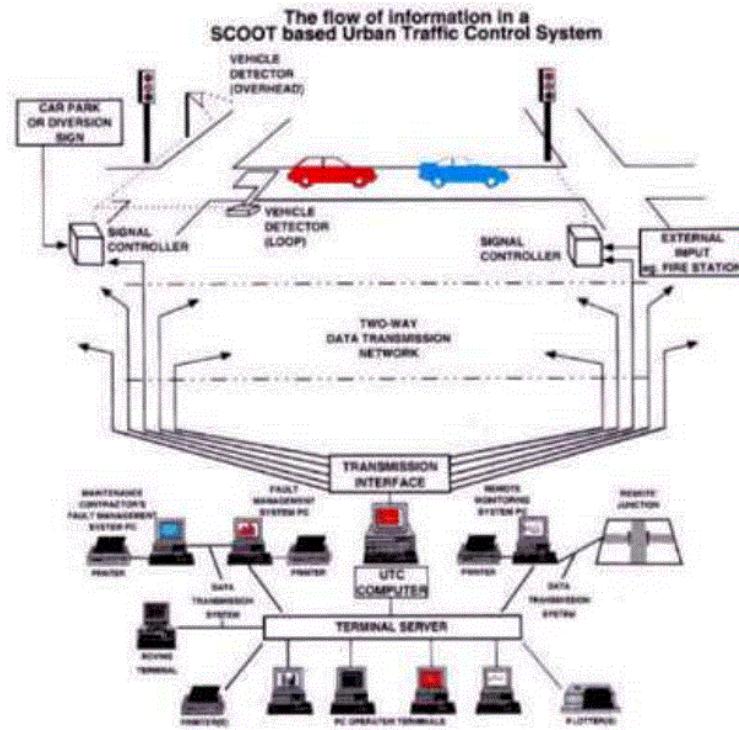


Figure 2.10: Collection of data by SCOOT System
 (Image Source : The "SCOOT" Urban Traffic Control System TRL Laboratory Report 1014)

The above figure emphasizes the real time working scenario of the SCOOT system which can be implemented. The inductive loops are used in the pavement for the intensity of the vehicle or better to say to predict the availability of the vehicle on roads. As the vehicle moves on the roads the availability is detected by the inductive loops. As soon as the availability is detected the signal are synchronously collected and are sent to the main operating main frame computer server which actually redirects the signal to the advance adaptive traffic system which based on the patter of the traffic controls the signaling of the traffic. SCOOT optimizes the signal so as to get the maximum green availability to control the traffic smooth and flexible. The traffic also manages the timing for the pedestrian so that there should not be any problem. The whole system is quite expensive and so the system can't be employed everywhere. So when the system is not being applied the manual traffic control through the signals can be used or when there's a system failure in that case also the manual timing control can be employed.

2.5 SENSORS FOR VEHICLE DETECTION

There is a wide range of sensor technologies available for vehicle detectors. Some of the most common and some developing technologies are described in this section. ^[7]

2.5.1 Video Image Processors

A video image processor (VIP) is a combination of hardware and software which extracts desired information from data provided by an imaging sensor. This imaging sensor can be a conventional TV camera or an infrared camera. A VIP can detect speed, occupancy, count, and presence. Because the VIP produces an image of several lanes, there is potential for a VIP to provide a wealth of traffic information such as vehicle classification and incident detection. A VIP generally operates in the following manner: the operator selects several vehicle detection zones within the field of view (FOV) of the camera. Image processing algorithms are then applied in real time to these zones in order to extract the desired information, such as vehicle speed or occupancy.

Advantages of VIPs are that they are mounted above the road instead of in the road, the placement of vehicle detection zones can be made by the operator, the shape of the

detection zones can be programmed for specific applications, and the system can be used to track vehicles. Disadvantages are the need to overcome detection artifacts caused by shadows, weather, and reflections from the roadway surface. The disadvantages can be overcome through design and installation of the hardware and design of the software algorithms.^[7]

2.5.2 Infrared Detectors

There are two types of infrared (IR) detectors, active and passive. Active infrared sensors operate by transmitting energy from either a light emitting diode (LED) or a laser diode. An LED is used for a non-imaging active IR detector, and a laser diode is used for an imaging active IR detector. In both types of detectors the LED or laser diode illuminates the target, and the reflected energy is focused onto a detector consisting of a pixel or an array of pixels. The measured data is then processed using various signal-processing algorithms to extract the desired information. Active IR detectors provide count, presence, speed, and occupancy data in both night and day operation. The laser diode type can also be used for vehicle classification because it provides vehicle profile and shape data.^[7]

A passive infrared system detects energy emitted by objects in the field of view and may use signal-processing algorithms to extract the desired information. It does not emit any energy of its own for the purposes of detection. Passive infrared systems can detect presence, occupancy, and count.

Some of the advantages of infrared detectors are that they can be operated during both day and night, and they can be mounted in both side and overhead configurations. Disadvantages are that infrared detectors can be sensitive to inclement weather conditions and ambient light. The choice of detector materials and construction of the system, as well as sophisticated signal processing algorithms, can compensate for the disadvantages.

2.5.3 Ultrasonic Detectors

Ultrasonic detectors have not become widely used in the United States, but they are very widely used in Japan. Japan uses ultrasonic detectors in traffic applications as much as the U. S. uses inductive loop detectors in traffic applications. There are two types of ultrasonic sensors available, presence-only and speed measuring. Both types operate by transmitting ultrasonic energy and measuring the energy reflected by the target. These measurements are processed to obtain measurements of vehicle presence, speed, and occupancy.

The advantages of ultrasonic are that they provide all-weather operation, do not need to be approved by the FCC, and provide fixed or portable mounting fixtures above the road. Their disadvantages include their need to be mounted in a down-looking configuration as perpendicular as possible to the target (as opposed to side mounting), a difficulty in identifying lane-straddling vehicles and vehicles traveling side by side, and susceptibility to high wind speeds. Some of these disadvantages may be compensated for through more sophisticated data processing techniques. [7]

2.5.4 Microwave/Millimeter Wave Radar

Microwave detectors have been used extensively in Europe, but not in the United States. They operate by measuring the energy reflected from target vehicles within the field of view. By processing the information received in the reflected energy, the detectors measure speed, occupancy, and presence. Some of the advantages of microwave detectors are that they are a mature technology because of past military applications, they detect velocity directly, and a single detector can cover multiple lanes if it is placed properly and appropriate signal processing techniques are used. In addition, FCC approval is not required if it operates in the X-band or Ku-band, and the output powers are within specified limits. Some of the disadvantages are unwanted vehicle detection based on reception of sidelobe radiation, and false detection due to multipath. Most of these disadvantages can be overcome, in whole or in part, through proper placement of the detectors, signal processing algorithms, and antenna design. [7]

2.5.5 Passive Acoustic Detector Arrays

Another type of vehicle detector is the passive acoustic array. An array of microphones may be used to determine the passage of a vehicle. The signals from the microphones in the array are processed and correlated to obtain information about vehicle passage. The design of the array determines its directionality and field of detection. These types of detectors have not yet been thoroughly investigated, at least in terms of traffic related applications. Video-conferencing companies have been developing sophisticated microphone arrays for their systems, and it is possible that some of their techniques or designs could be adapted to traffic applications.^[7]

2.5.6 Piezoelectric Detectors

Piezoelectric detectors are very accurate vehicle detectors, but they do not detect presence of a stationary vehicle, unless it has stopped with its wheels on the detector. The piezoelectric sensor consists of a long strip of piezoelectric material enclosed in a protective casing. It can be embedded flush with the pavement, and when a car passes over it compressing the piezoelectric material, a voltage is produced. This sets off the controller. The piezoelectric detector has the advantage of indicating exactly when and where a vehicle passed by because it is a line detector perpendicular to the path of the vehicle. A series of two of them may be used to measure vehicle speed. A disadvantage is that for a permanent installation, they must be embedded in the pavement. Every time the roadway is repaved, or if a pothole appears, the sensor would need to be replaced. These types of sensors are currently being tested on the Beltway in Virginia. AMP is a manufacturer of piezoelectric traffic detectors.^[7]

2.5.7 Photoelectric Detectors

Photoelectric devices commonly consist of two components, the light source and the detector. These may both be in the same place, or placed across from each other. When placed across from each other, the detector is activated whenever something obstructs the illumination from the light source. When placed in the together, the detector is activated when light from the light source is reflected from a target and back onto the detector.

There is not enough information on these detectors as applied to vehicle detection. They do not appear to be a competitive technology in the field of vehicle detectors at this time.

2.5.8 Spread-spectrum Wideband Radar

New wideband spread-spectrum radar has recently been developed at Lawrence Livermore Laboratory. It is a significant development because it is very inexpensive and it has extremely accurate range discrimination. It can also penetrate many types of materials, including concrete. It has a range of about 20 feet, so it may be useful as an inexpensive, single-lane vehicle detector. It is predicted that the sensor, when made in production quantities, would cost much less than \$10 per sensor. Because of their accurate range discrimination, they have a very well defined field of detection. They could become a cheap alternative to magnetometer probes. Their ability to detect range provides additional information for future traffic control systems. In addition, Lawrence Livermore has stated that they are developing a broadband transmitter/receiver pair to be used with these sensors. This would eliminate the need for communication lines between the sensor and the controller.^[7]

2.5.9 Inductive Loop Detectors (Recommended)

Loop detectors are the most widely used technology for vehicle detection in the United States. A loop detector consists of one or more loops of wire embedded in the pavement and connected to a control box. The loop may be excited by a signal ranging in frequency from 10 kHz to 200 kHz. This loop forms an inductive element in combination with the control box. When a vehicle passes over or rests on the loop, the inductance of the loop is reduced. This causes a detection to be signaled in the control box.

The advantages of inductive loop detectors are that they are an established technology in the United States, they have a well-defined zone of detection, and they are generally reliable. Disadvantages are that the detectors are very sensitive to the installation process, they can only be installed in good pavement, and they must be reinstalled every time a road is repaved.^[7]

2.5.10 Magnetic Detectors

There are two other types of magnetic detectors, which are used to detect traffic. Both of them are in the form of probes, and they both operate on the principle of a large metal object disturbing a magnetic field, just as inductive loop detectors work. There are both active and passive types. The active type is called a magnetometer. A magnetometer acts in much the same way as an inductive loop detector, except that it consists of a coil of wire wrapped around a magnetic core. It measures the change in the magnetic field caused by the passage of a vehicle. It can be used both for presence and for vehicle passage detection.

The passive type of detector simply measures a change in the flux of the earth's magnetic field caused by the passage of a vehicle. These detectors can only detect moving vehicles, so they cannot be used as presence detectors. They have a fairly large detection range and thus can be used to observe multiple lanes of traffic.

The advantage of both of these types of magnetic detectors is that they can be used where point or small-area location of a vehicle is necessary. For example, on a bridge, inductive loop detectors would be disrupted by the steel struts, and it is necessary to have a point detector. One of their disadvantages is that multiple detectors need to be installed to detect smaller vehicles, such as motorcycles.^[7]

2.5.11 Acceleration Detectors

For the left-turn collision countermeasure system, it is necessary to determine the acceleration of the vehicle, so that it can be determined whether or not the vehicle is slowing to make a left turn. Using Doppler information, the range rate of a vehicle may be determined, but it does not appear that any radar currently being marketed for traffic applications measure the range rate. A simple method is to have three detectors in a linear formation. Measurements from these three detectors will provide an approximation of the acceleration of the vehicle from which the system may determine whether or not to activate the left-turn ahead warning.^[7]

CHAPTER 3

PROJECT DESIGN

3.1 SINGLE NODE TRAFFIC CONTROLLER

The very first way was to analyze the traffic is the single junction. For this purpose a cross junction was taken into consideration. The traffic intensity was monitored with the help of sensors which can predict the presence of vehicle. As of now, we have considered switches to show the intensity and occupancy of the road, which can be easily replaced by any of the previously mentioned digital vehicle detectors. For the convenience of understanding the four way cross junction is shown. The road with double arrow represents the main road or the road with the highest intensity of traffic majority of the time. The other three roads are less occupied as compared with that of the main road. For our own convenience we have given the first priority to the main road. During a cycle time, the main road will be given preference as in traffic of the main road will be cleared for the time being. After the traffic clearance for some particular interval of time the intensity of the rest of the road will be compared and based on the traffic intensity the clearance of that road will be given and hence traffic of all crossing road will be cleared for the particular time interval.

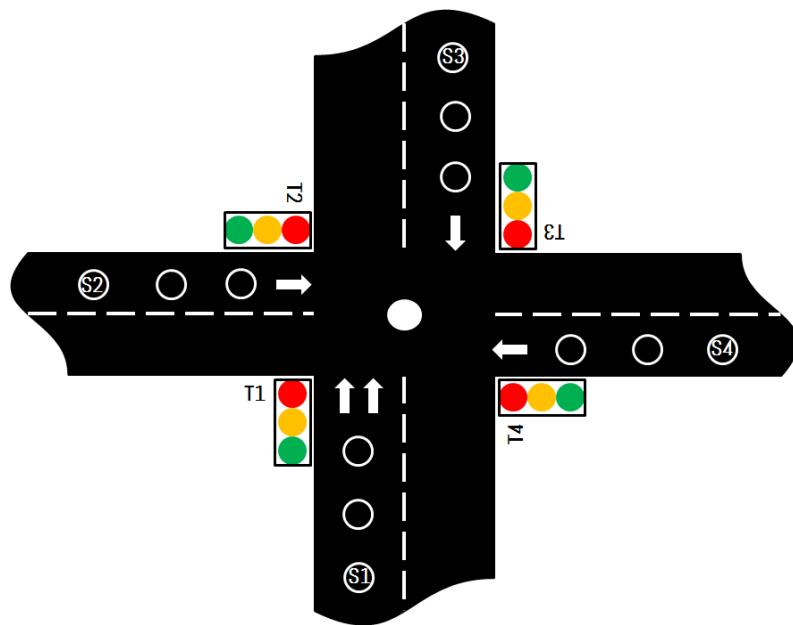


Figure 3.1: Single Node Traffic Model

3.1.1 Traffic Lights

In the Single Node Model as shown in Figure 3.1, there are four traffic lights, denoted as T1, T2, T3 and T4. Each of these traffic lights has three colours – RED (100), YELLOW (010) and GREEN (001). In the thesis, the notations T1,..., T4 have been used exchangably for traffic lights as well as street number, which will be clear from the context being discussed. Working is explained in section 4.1.

3.1.2 Sensors

Each of these circles on the roads are simple representation of inductive loop sensors as discussed in section 2.5.9. For our convenience, we have assumed that each of these lanes has three sensors to measure the traffic intensities. These sensors are digital in nature i.e. they give a HIGH output when the vehicle is present, and a LOW output in the absence of any vehicle. Since inductive loop sensors get activated only when some metallic object (like a car) comes into its vicinity, it is the most suitable form of sensor to be used. In addition to that, there is no human interference with these sensors since humans have very negligible or almost nil metal content.

These sensors are denoted as S1, S2, S3 and S4 for each of the four lanes as shown in Figure 3.1. Thus, they send in a digital array of data into the FPGA. Since it sends three bits of data, there are eight conditions, out of which, only the following are taken into consideration and others as don't cares.

- 000: Road is empty
- 001: Road is less crowded
- 011: Road is moderately crowded
- 111: Road is fully crowded

Thus, these sensors, when used as an array, can send real time traffic information to the FPGA for processing. More details regarding how the data is interpreted and processed can be found in section 4.1.

3.1.3 Delay Implementation

In any conventional traffic light controller, there is a concept of delay. There is a delay when the traffic light goes from GREEN to YELLOW, YELLOW to RED, etc. The same concept of delay has been used in this project with some modifications to make it more dynamic and real-time. There are two types of delays used here – fixed and variable.

In **fixed delay implementation**, the delay between two states of the traffic lights is fixed and is determined beforehand. This implementation is not dynamic in nature and does not take into consideration the real-time traffic intensities sent by the sensors. This is the conventional method of providing delays in traffic light controllers.

In **variable delay implementation**, the delay between two states of the traffic lights is variable. This is dynamic in nature, and changes in accordance with the traffic intensities of S1, S2, S3 and S4. For instance, if the road is less crowded, the GREEN time is lesser as compared to if the road would have been more crowded. This is implemented in order to effectively optimize and regulate the traffic.

3.2 PROJECT DESCRIPTION

The overall project is based on a simple concept – take the real-time data from sensors, process them to determine the optimum traffic efficiency, and give the output in the form of manipulated traffic light signals. A simplified block diagram is shown below:



Figure 3.2: Simplified Block Diagram

3.3 DESIGN METHODOLOGY

This section describes our design methodology and how we implemented the project.

- **Project Analysis** – We analyzed the current situation of city traffic with the help of Bangalore Traffic Control Center. We also learnt about the current technologies they use to control the city's traffic.
- **Algorithm Development** – After analyzing the current scenario and the current technologies, we planned our own algorithm, which has an edge over the existing traffic light controllers.
- **Software Design and Simulation** – Then the algorithm was implemented in ModelSim® 10.1b using Verilog HDL and then simulated in order to validate the algorithm.
- **Hardware Design** – Then the algorithm was implemented in Quartus® II v9.1 software using Verilog HDL in order synthesize the design.
- **Functional Verification** – The algorithm's validity after synthesis was once again verified using ModelSim® 10.1b software.
- **Hardware Interfacing** – Then the synthesized code was implemented in hardware FPGA by means of USB Blaster.
- **Timing Analysis** – Then timing analysis of the system was done using the in-system debugging feature of the SignalTap® II Embedded Logic Analyzer of the Quartus® II software.
- **Testing** – Finally, the entire system was tested under various circumstances.

CHAPTER 4

DESIGN IMPLEMENTATION

4.1 STATE DIAGRAM

Before proceeding with the software implementation of the project, we have designed and implemented a state diagram for the entire algorithm. A simplified version of the State Diagram is given below, which will be explained just after that.

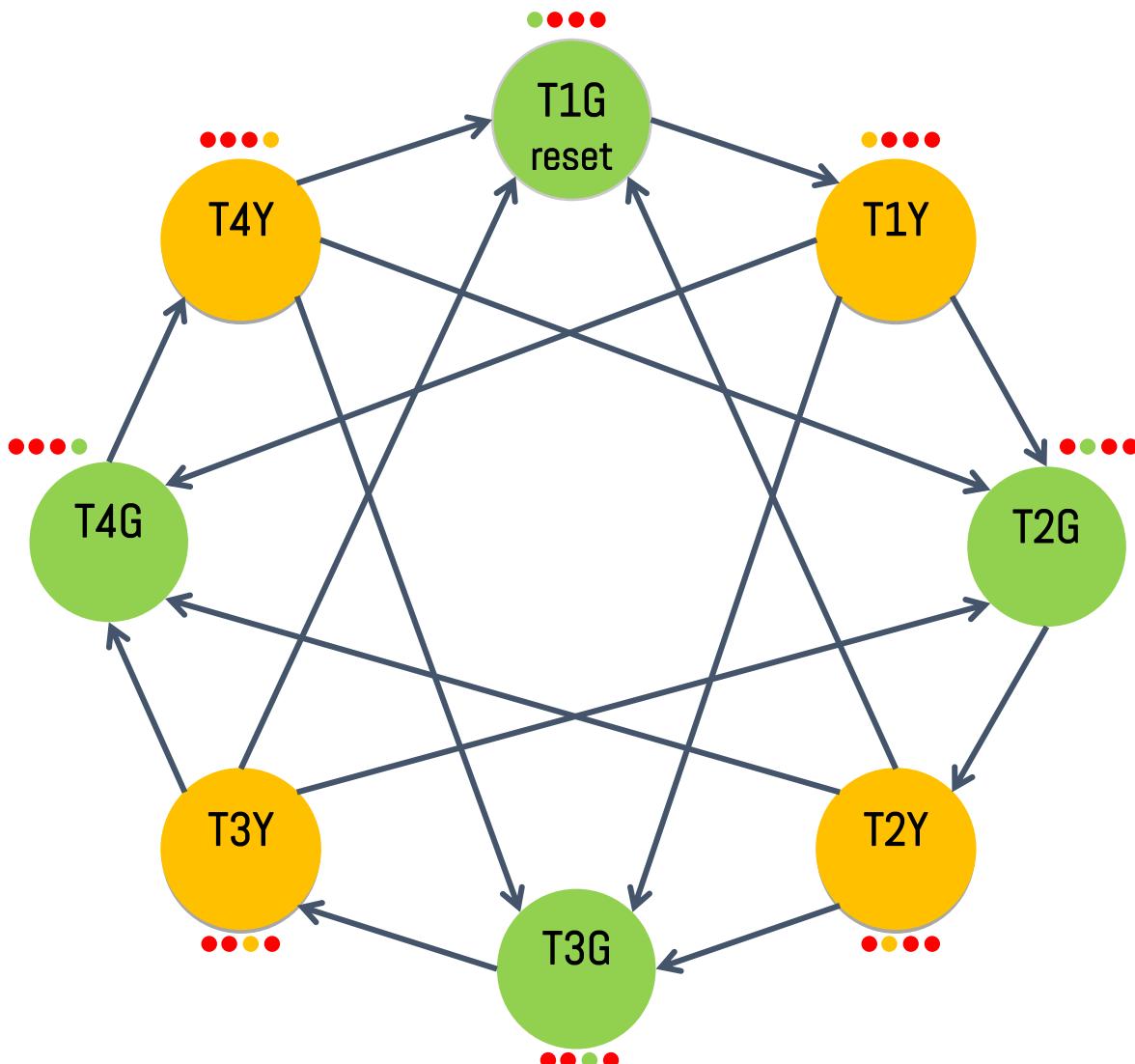


Figure 4.1: Simplified State Diagram

The state diagram is designed in such a way that the traffic lights toggle among themselves in a cyclic manner. Each “cycle” has four “rounds”, each round consists of a particular traffic light being GREEN followed by YELLOW. Thus, in one cycle, all the four traffic lights become GREEN and YELLOW only once.

4.1.1 State Diagram Explained

In the state diagram in Figure 4.1, there are only eight states, which are designated as follows:

- a) T1G: Only traffic light T1 is **GREEN**, rest all are **RED** (reset state)
- b) T1Y: Only traffic light T1 is **YELLOW**, rest all are **RED**
- c) T2G: Only traffic light T2 is **GREEN**, rest all are **RED**
- d) T2Y: Only traffic light T2 is **YELLOW**, rest all are **RED**
- e) T3G: Only traffic light T3 is **GREEN**, rest all are **RED**
- f) T3Y: Only traffic light T3 is **YELLOW**, rest all are **RED**
- g) T4G: Only traffic light T4 is **GREEN**, rest all are **RED**
- h) T4Y: Only traffic light T4 is **YELLOW**, rest all are **RED**

Out of these 8 states, state T1G is the initial/reset state. When the system starts, initially it goes to this state and traffic light T1 is made GREEN whereas rest all (T2, T3 and T4) are made RED.

The system remains at this initial state T1G for a stipulated amount of time, which is calculated based upon the current traffic intensity. Once the required delay is over, the system moves to another state, T1Y. In this transition from T1G to T1Y, only the traffic light T1 is changed from GREEN to YELLOW. The system stays in this state, again for some stipulated amount of time, after which it can either go to any of these states: T2G, T3G or T4G. This depends upon the traffic intensities of the three roads.

Referring to Figure 4.2, the Single Node Model is a junction of four roads, and as discussed in Chapter 3, their traffic intensities are designated as S1, S2, S3 and S4 respectively. After first round, only the traffic intensities S2, S3 and S4 are compared.

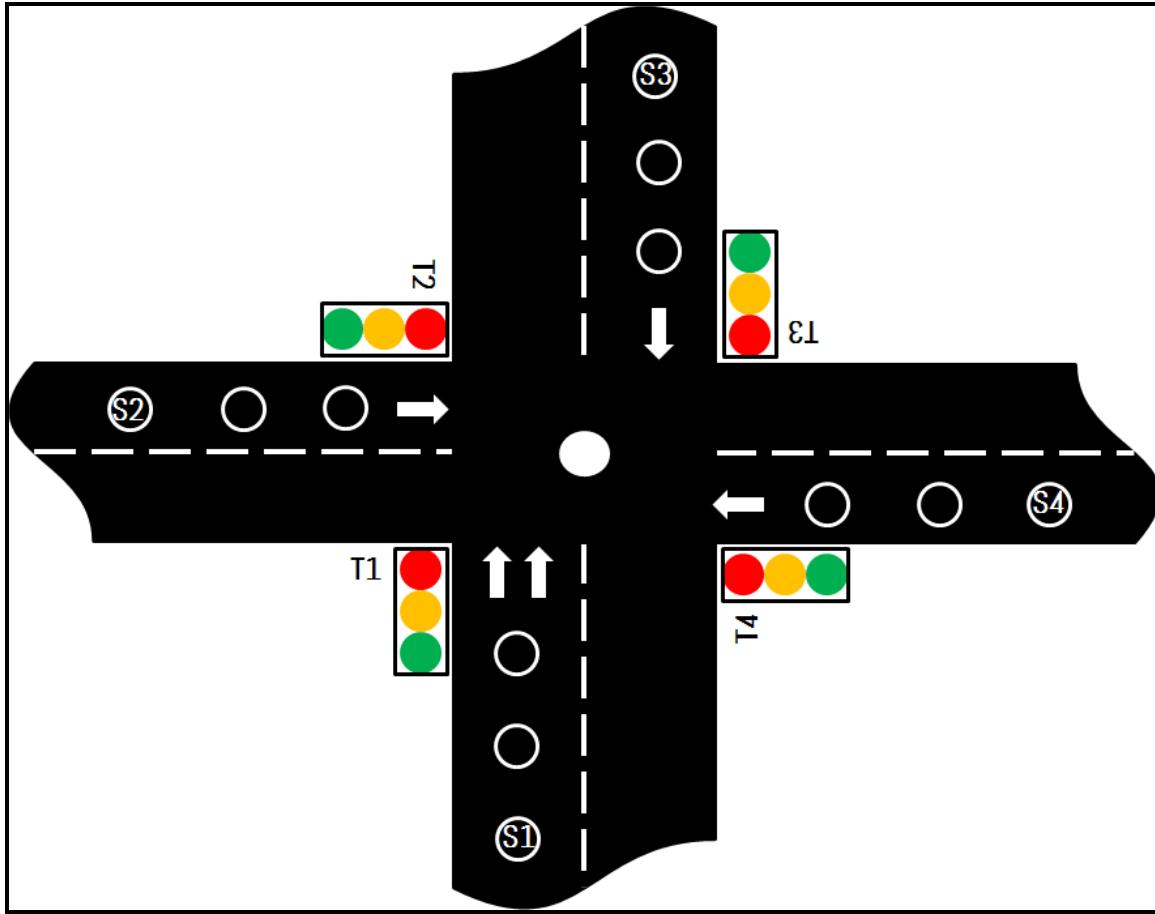


Figure 4.2: Single Node Model

This is because the traffic light T1 has already been GREEN in the first round, and it will be GREEN again only in the next cycle. Hence, for the current cycle, only the remaining traffic lights (T2, T3 and T4) need to be switched on, thus comparing only S2, S3 and S4. Thus, for the next (second) round, the system will go into the state which has the maximum traffic intensity. For example, if out of S2, S3 and S4, S3 has the maximum traffic, traffic light T3 will be made GREEN, and thus, the system will go from state T1Y to T3G.

Now once again, the round is repeated. Traffic light T3 will be GREEN for some time, and then it will be YELLOW for some time. In other words, the system moves from state T3G to state T3Y. Again, after some delay, the system changes state. But this time, for the next (third) round, only the remaining two traffic intensities will be compared. In this scenario, only S2 and S4 will be compared. If traffic of S2 is greater than S4, then traffic

light T2 will be made GREEN and YELLOW. That is, from state T3Y, it goes to T2G, and then to T2Y. After that, for the next (fourth and last) round, there is no need of comparing traffic intensities, since for the current cycle, only one traffic light remains, T4. Thus, it straightaway goes to state T4G, followed by T4Y.

When all the four rounds of a cycle are complete, the system again goes into its reset state, marking the beginning of another cycle. Since we have assumed that the road with intensity S1 is the main road, it is always given a preference in every cycle to start with. Thus, this method takes in the real time inputs from sensors to generate traffic intensity values, and this information is used to prioritize the different traffic lights.

As far as delay is concerned, it is the same as discussed in Chapter 3. It can be of two types – Fixed Delay and Variable Delay. In Fixed Delay Implementation, the delay for GREEN light will be fixed for all the rounds of all the cycles. In Variable Delay Implementation (recommended), the time for GREEN signal depends upon the dynamic traffic intensity of that road. If the traffic is less on that road, the GREEN time will be less, and if the road is crowded, then the GREEN time will be more.

4.1.2 Condition of Equality and Priority Order

The general operation of the state diagram is explained in the previous section 4.1.1. In this section, we will include the condition of equality and priority order into the algorithm. To understand about the condition of equality, let us assume that all roads are equally crowded. In this case, the priority order is determined as: T1 > T2 > T3 > T4. T1 is given the highest priority since it is assumed to be the main road. This priority is taken into account in all the rounds of a cycle. For instance, for the third round, we are comparing S2 and S3. If both the roads have same traffic i.e. S2 = S3, then as per the priority criteria, T2 will be chosen over T3 for that round.

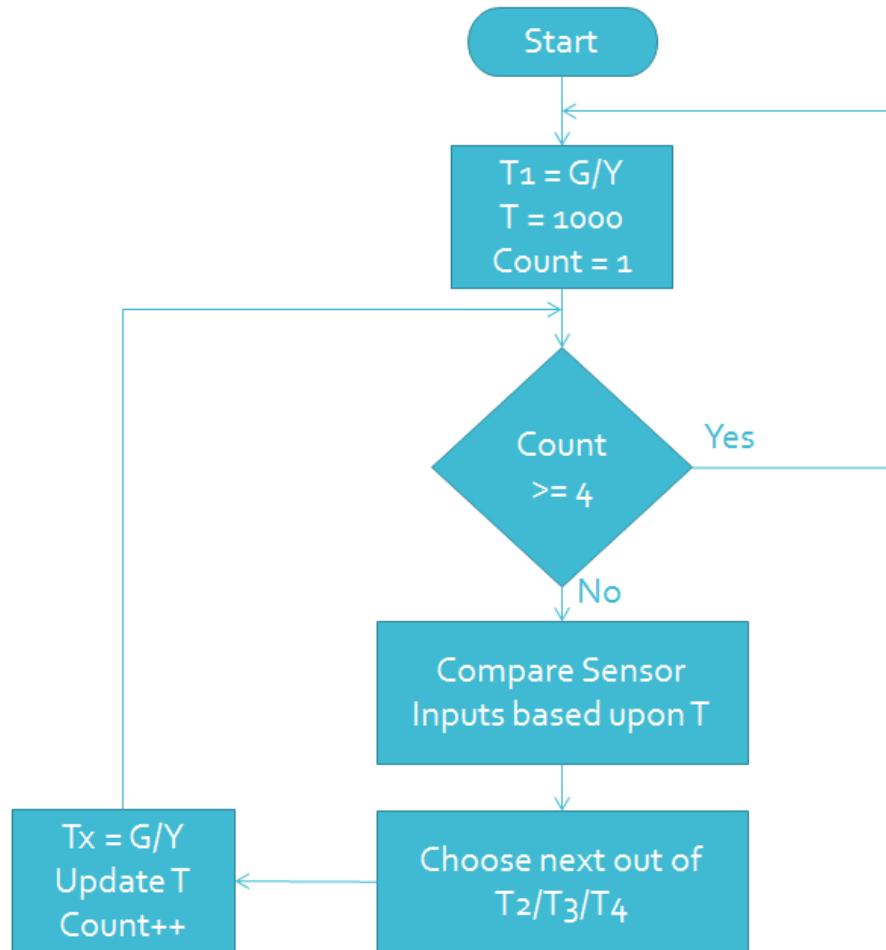
As a special case to this, if all the roads are empty, even then the same priority will be taken into account. If all the roads are empty, it will be useless to have RED signal for all of them, and we don't have a separate state for that as well. Thus, in case of completely

empty roads, the traffic lights will still toggle among themselves as per their priority order.

4.1.3 Empty Roads

If out of all of the roads, if any of the roads in empty (without any traffic), then that road will not be given a GREEN signal unless it has some traffic intensity. For instance, if $S_2 = 0$, then T_2 will not be GREEN in that cycle. However, this condition is not valid for T_1 since it is assumed to be the main road. An exception to this is already covered in the previous section 4.1.2 when all the roads are empty. This will be treated under the equality criteria.

4.2 SIMPLIFIED ALGORITHM OF SINGLE NODE MODEL



4.3 DETAILED FUNCTIONAL DESCRIPTION

We used the Altera® Quartus® II software version 9.1 to design the entire system for the Altera® Cyclone II FPGA of the DE1 board. We created this using Verilog HDL. Our design has the following functionality:

- Sensors detect the presence of a vehicle on the road and sends signals to the FPGA.
- The FPGA decodes the data from the sensors as traffic intensities
- Clock division method was used to generate 1 Hz pulse (for a delay of 1 second) from the 24 MHz on-board clock source.
- At every rising edge of the 1 Hz generated clock, depending upon the delay counter, either the state is changed or the system remains in the same state.
- Each state is assigned a particular colour code (like Red-Green-Red-Red, etc.)
- Depending upon the state, necessary delay is given by means of a delay counter. After that delay is over, the state is changed.
- Depending upon the present state and the traffic intensity information from the sensors, the next state is predicted dynamically.
- In one cycle, all the traffic lights are toggled only once, thus there are four rounds in a cycle, after which the cycle repeats.

4.4 PROTOTYPE DEPLOYMENT

A sample prototype model was made for this project by means of stationery items such as thermocols, chart papers, acrylic colours, IR proximity sensors and self-soldered PCBs for traffic lights. The snapshots are shown below.

4.4.1 *Traffic Light*

Traffic light is designed as a small self-soldered PCB as shown in the picture.

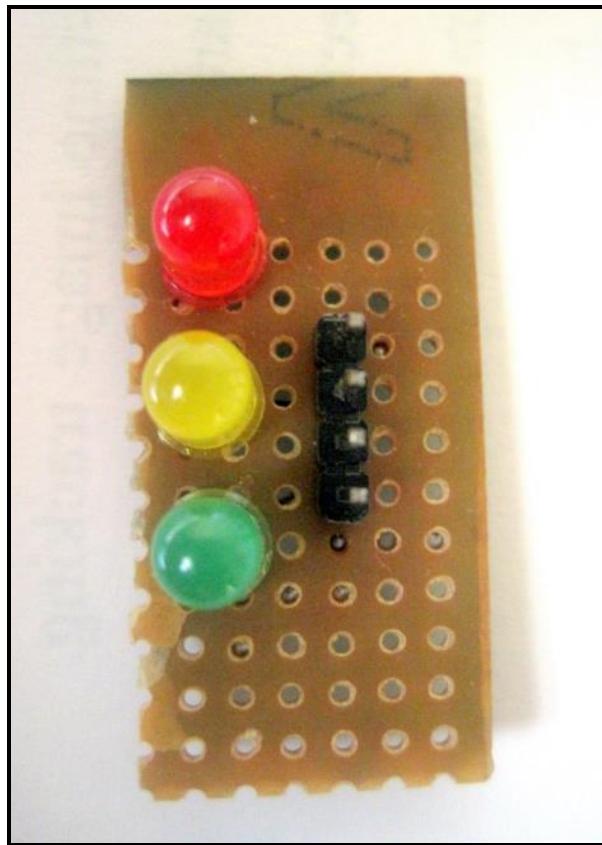


Figure 4.3: Self-Soldered PCB for Traffic Light

4.4.2 *Vehicle Detector*

For prototyping purposes, we used simple proximity IR sensors as vehicle detectors. These sensors give digital output – HIGH (1) when a vehicle is present and LOW (0) in the absence of any vehicle. One sample is shown in the picture below.

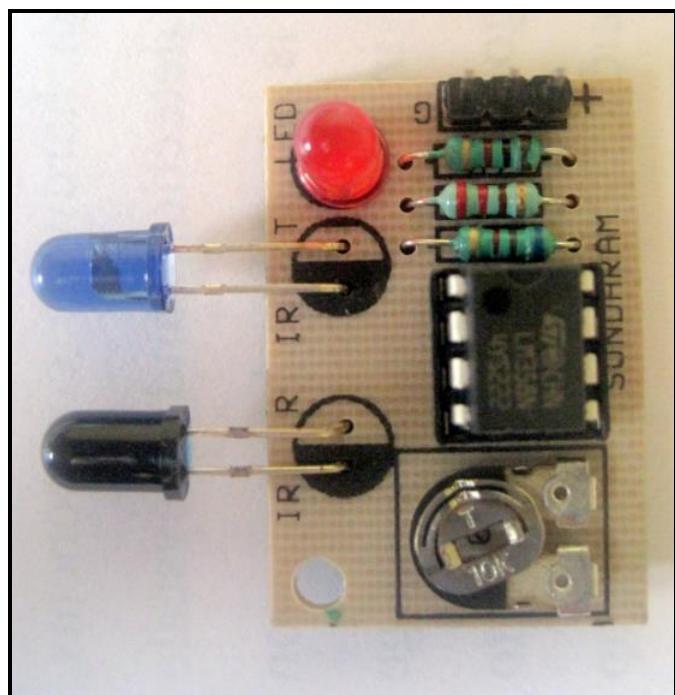


Figure 4.4: Proximity IR Sensor

4.4.3 Experimental Setup

The following setup was used for prototyping purposes.

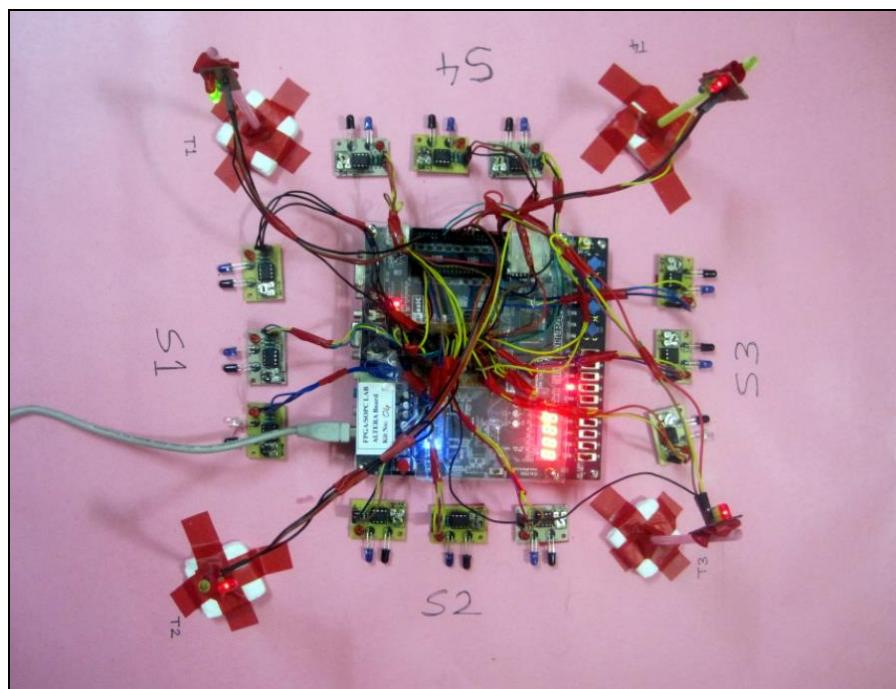


Figure 4.5: Experimental Setup

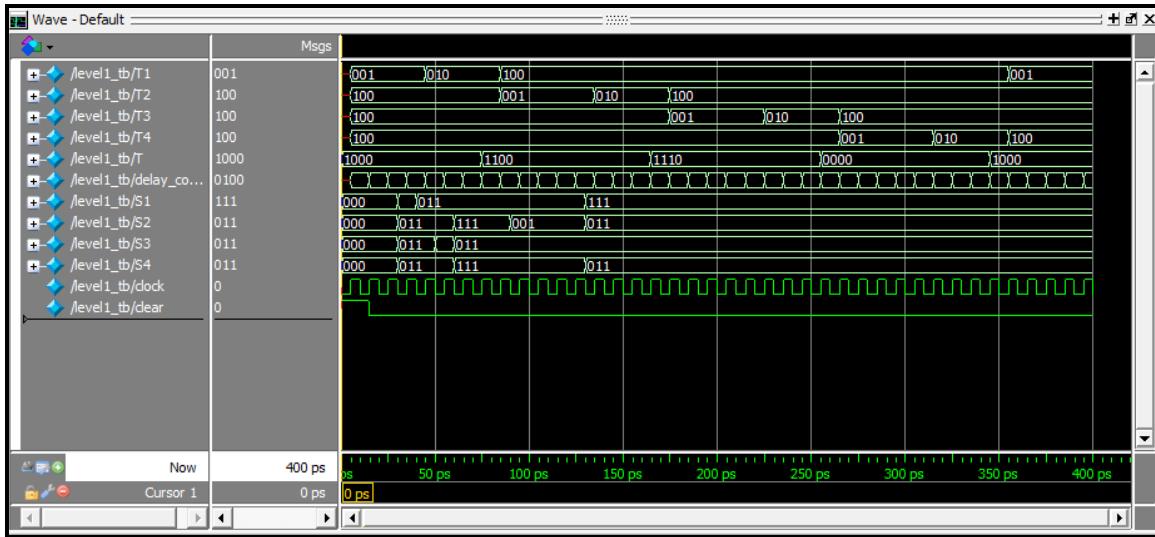
CHAPTER 5

RESULT AND ANALYSIS

5.1 FUNCTIONAL VERIFICATION

After the code was written, compiled and synthesized in Quartus® II v9.1 software, it was simulated in Mentor Graphics® ModelSim® ALTERA Starter Edition 10.1b. The simulation done to achieve the following:

- to check the algorithm implemented
- to perform timing analysis of the algorithm
- to perform functional verification of the entire system



The following can be clearly seen in the waveform:

- The traffic lights (T1, T2, T3 and T4) toggle once in a cycle.
- Each traffic light becomes GREEN (001) from RED (100), then YELLOW (010) after the specified delay interval, and then again back to RED (100) after some delay.
- The values from the sensors i.e. traffic intensities (S1, S2, S3 and S4) change as per the triggers provided.
- The delay counter increments at every rising edge of the clock pulse. When the required delay is achieved, the value of the traffic lights change.

ModelSim Simulation Monitor Window Output

```

0 T = 1000 T1 = xxx T2 = xxx T3 = xxx T4 = xxx S1 = 000 S2 = 000 S3 = 000 S4 = 000 Delay = x
5 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 000 S2 = 000 S3 = 000 S4 = 000 Delay = 1
15 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 000 S2 = 000 S3 = 000 S4 = 000 Delay = 2
25 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 000 S2 = 000 S3 = 000 S4 = 000 Delay = 3
30 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
35 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 4
40 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 011 S3 = 011 S4 = 011 Delay = 4
45 T = 1000 T1 = 010 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 011 S3 = 011 S4 = 011 Delay = 0
50 T = 1000 T1 = 010 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 011 S3 = 111 S4 = 011 Delay = 0
55 T = 1000 T1 = 010 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 011 S3 = 111 S4 = 011 Delay = 1
60 T = 1000 T1 = 010 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 111 S3 = 011 S4 = 111 Delay = 1
65 T = 1000 T1 = 010 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 111 S3 = 011 S4 = 111 Delay = 2
75 T = 1100 T1 = 010 T2 = 100 T3 = 100 T4 = 100 S1 = 011 S2 = 111 S3 = 011 S4 = 111 Delay = 3
85 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 011 S2 = 111 S3 = 011 S4 = 111 Delay = 0
90 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 011 S2 = 001 S3 = 011 S4 = 111 Delay = 0
95 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 011 S2 = 001 S3 = 011 S4 = 111 Delay = 1
105 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 011 S2 = 001 S3 = 011 S4 = 111 Delay = 2
115 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 011 S2 = 001 S3 = 011 S4 = 111 Delay = 3
125 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 011 S2 = 001 S3 = 011 S4 = 111 Delay = 4
130 T = 1100 T1 = 100 T2 = 001 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 4
135 T = 1100 T1 = 100 T2 = 010 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 0
145 T = 1100 T1 = 100 T2 = 010 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 1
155 T = 1100 T1 = 100 T2 = 010 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 2
165 T = 1110 T1 = 100 T2 = 010 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
175 T = 1110 T1 = 100 T2 = 100 T3 = 001 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 0
185 T = 1110 T1 = 100 T2 = 100 T3 = 001 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 1
195 T = 1110 T1 = 100 T2 = 100 T3 = 001 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 2
205 T = 1110 T1 = 100 T2 = 100 T3 = 001 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
215 T = 1110 T1 = 100 T2 = 100 T3 = 001 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 4
225 T = 1110 T1 = 100 T2 = 100 T3 = 010 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 0
235 T = 1110 T1 = 100 T2 = 100 T3 = 010 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 1
245 T = 1110 T1 = 100 T2 = 100 T3 = 010 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 2
255 T = 0000 T1 = 100 T2 = 100 T3 = 010 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
265 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 001 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 0
275 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 001 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 1
285 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 001 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 2
295 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 001 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
305 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 001 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 4
315 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 010 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 0
325 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 010 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 1
335 T = 0000 T1 = 100 T2 = 100 T3 = 100 T4 = 010 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 2
345 T = 1000 T1 = 100 T2 = 100 T3 = 100 T4 = 010 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
355 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 0
365 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 1
375 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 2
385 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 3
395 T = 1000 T1 = 001 T2 = 100 T3 = 100 T4 = 100 S1 = 111 S2 = 011 S3 = 011 S4 = 011 Delay = 4

```

- All the signals reset when clear is HIGH.
- The variable “T” is used in a logic which maintains the cycle of traffic lights.
- The output of the monitor window is available above.

5.2 SYNTHESIS OUTCOME

The design has been compiled and synthesized successfully using the Quartus® II v9.1 software. The Compilation Report – Flow Summary is as follows:

Flow Status	Successful - Mon May 06 11:42:58 2013
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	level1
Top-level Entity Name	level1
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	No
Total logic elements	1,002 / 18,752 (5 %)
Total combinational functions	585 / 18,752 (3 %)
Dedicated logic registers	769 / 18,752 (4 %)
Total registers	769
Total pins	34 / 315 (11 %)
Total virtual pins	0
Total memory bits	4,736 / 239,616 (2 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

5.2.1 RTL Netlist

Some of the major hardware components as a result of synthesis are shown below. All the following snapshots are taken from the RTL Viewer of Quartus® II v9.1 software.

The complete synthesized netlist of the entire system are shown in Figures 4.3 and 4.4. The RTL representation of state and next state is shown in Figure 4.5. An instance of comparison of different traffic intensities is shown in Figure 4.6. To implement various delays in the program, a delay counter is used, which is shown in Figure 4.7.

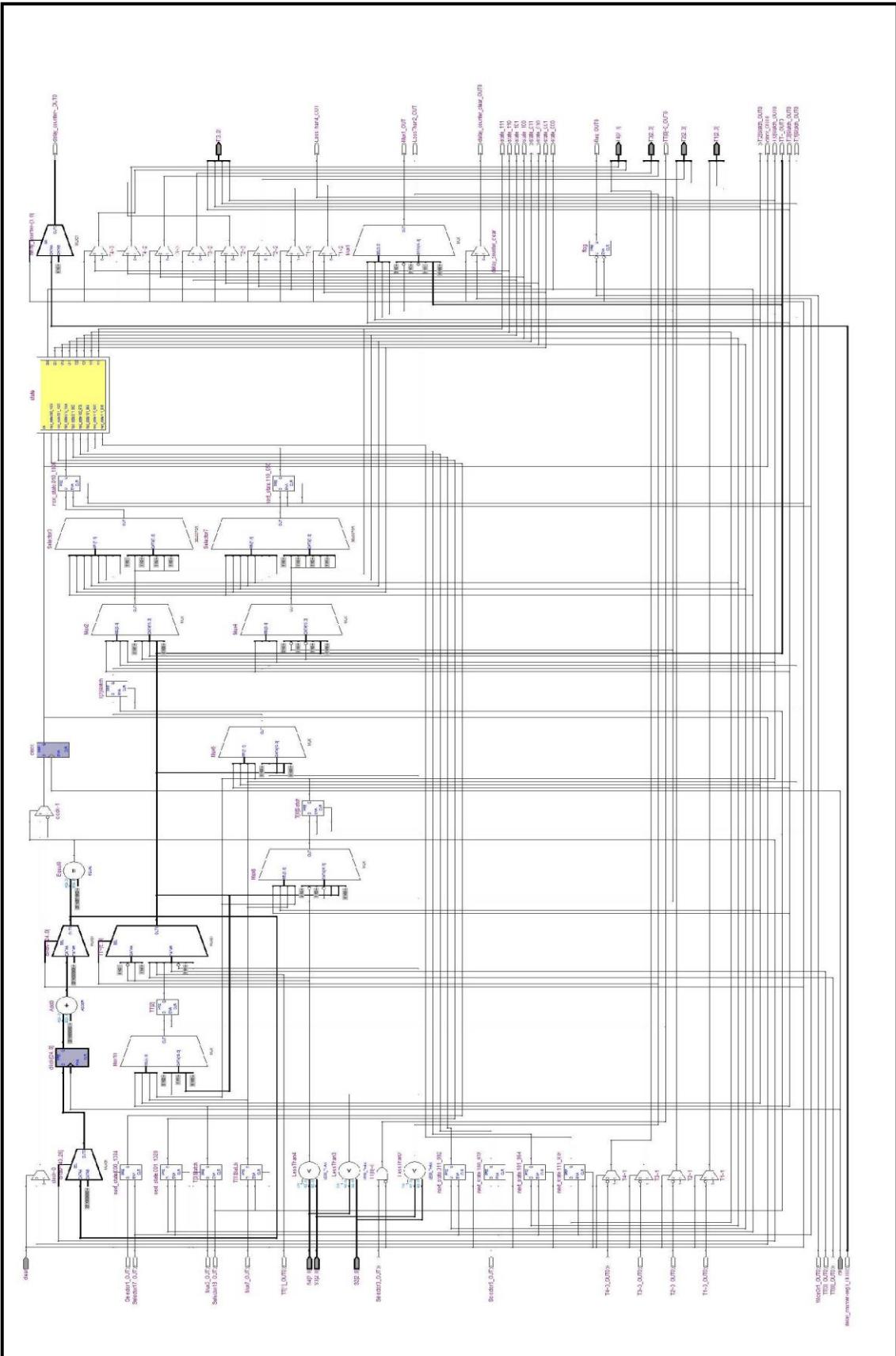


Figure 5.1: Complete Synthesized RTL Netlist (Part 1 of 2)

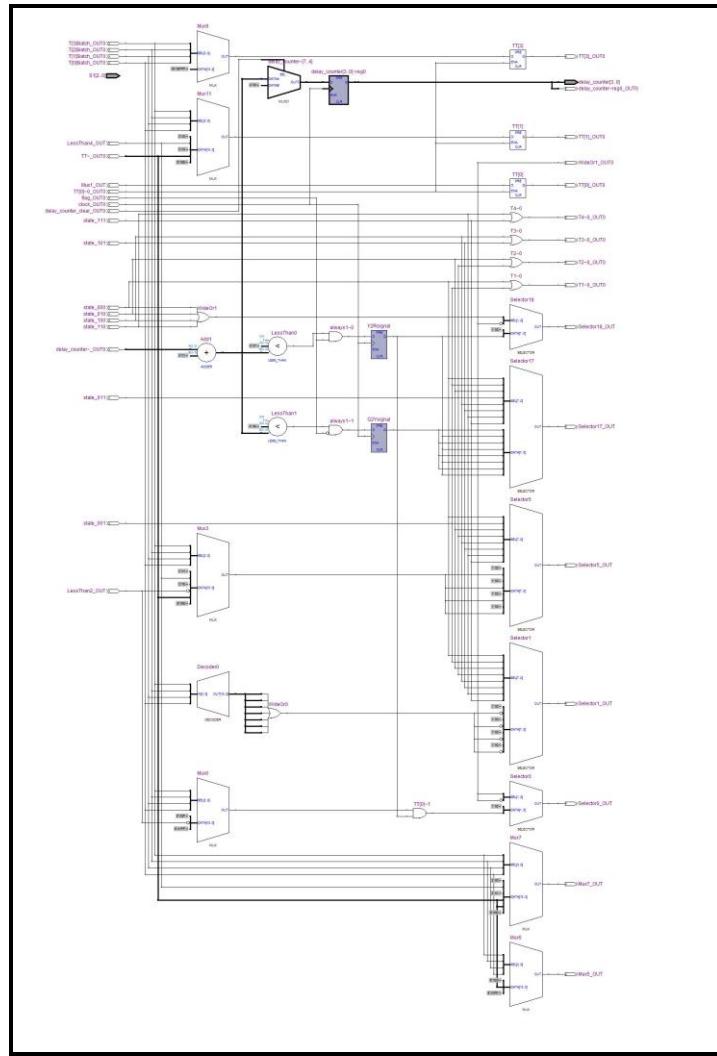


Figure 5.2: Complete Synthesized RTL Netlist (Part 2 of 2)

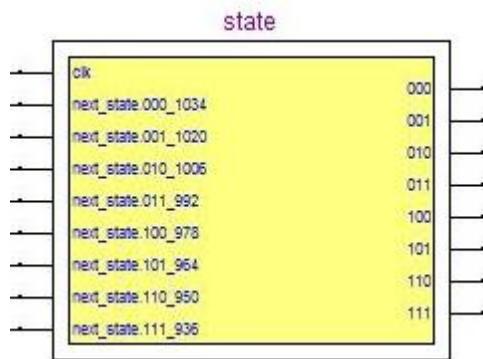


Figure 5.3: RTL Representation of State and Next State

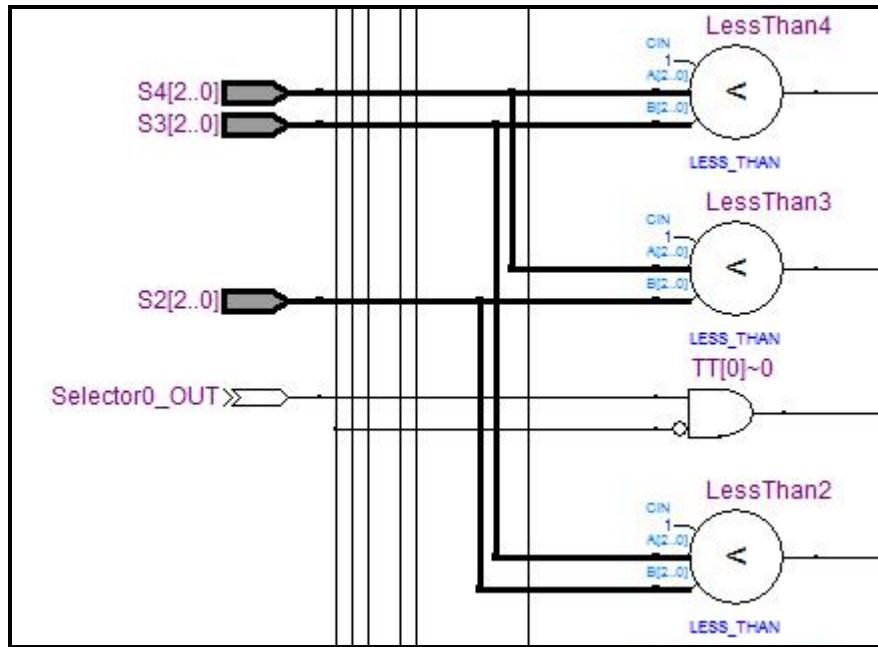


Figure 5.4: An instance of comparison of different traffic intensities

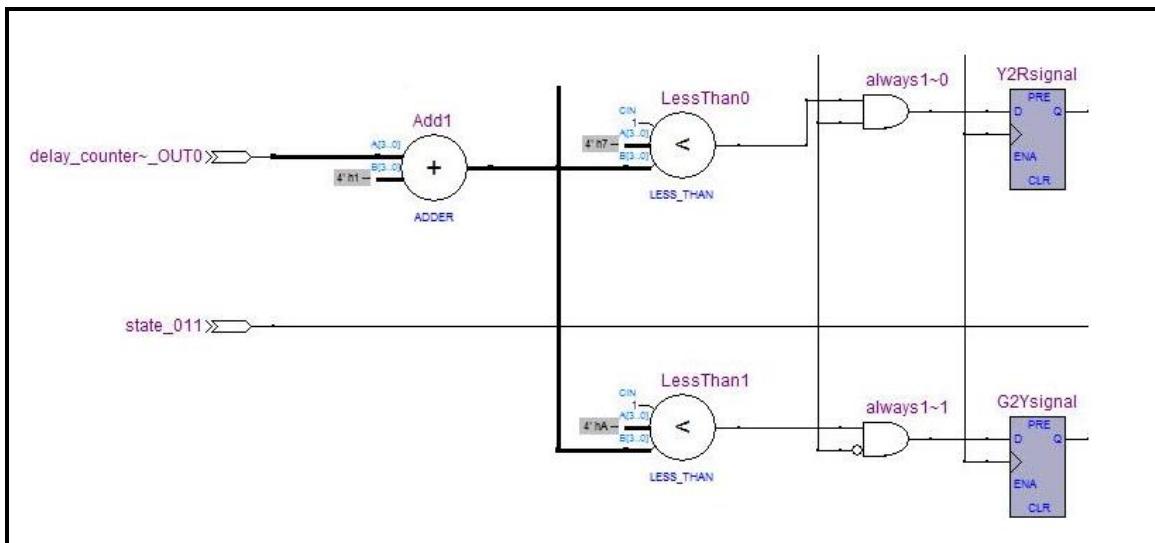


Figure 5.5: Implementation of Delay Counter

5.3 EXPERIMENTAL RESULT

Finally the synthesized code was dumped into the FPGA and the desired results were obtained.

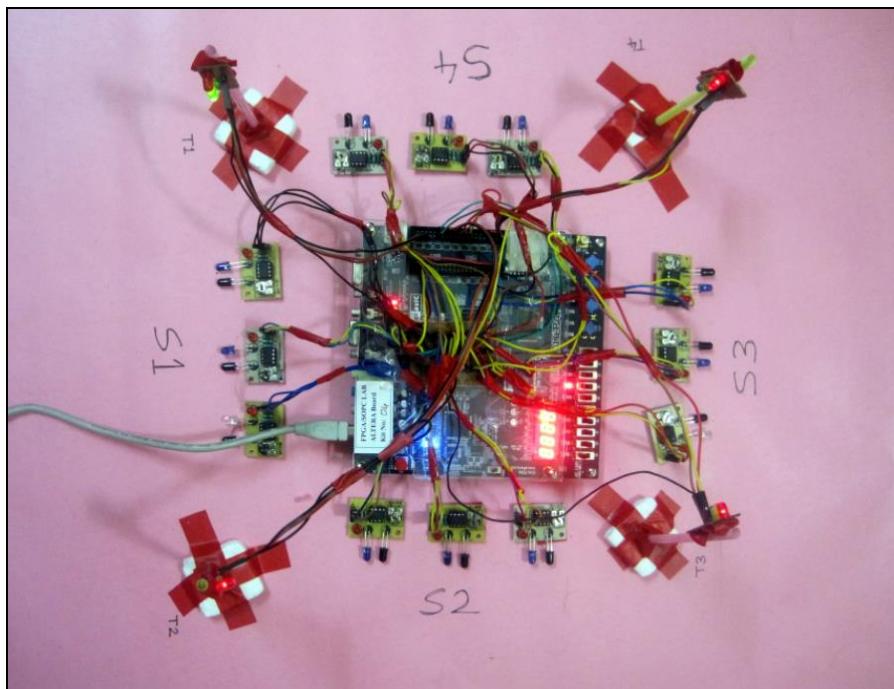


Figure 5.6: Experimental Setup - Top View

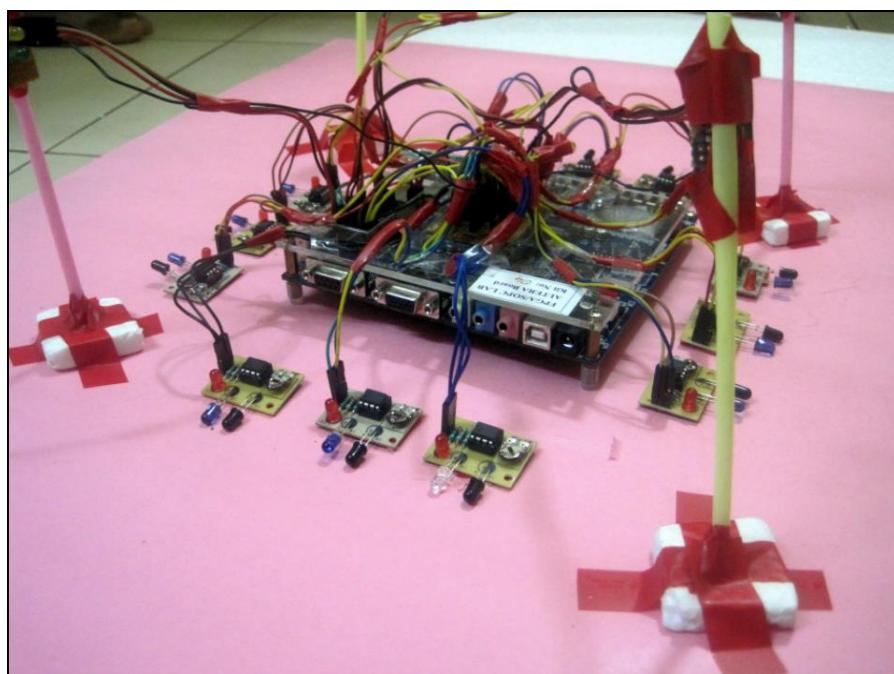


Figure 5.7: Experimental Setup - Side View

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The design has been compiled and synthesized successfully using the Quartus® II v9.1. We divided our work into architectural development including development of the new algorithm, and hardware development which are described as follows:

Architecture Development

We implemented the entire system in Quartus II 9.1 and successfully generated the necessary architecture required for regulation and control of traffic system. We designed a new algorithm which can detect the intensity of traffic in real-time and based on that, the traffic can be controlled and regulated accordingly.

Hardware Development

One of the most important part through out the project work was the implementation of the project in hardware. We used Altera® DE1 Development and Education Board for this purpose. Besides this, infrared proximity sensors were used for the detection of prototype vehicles which acts as inputs to the FPGA. The FPGA processes the data from the sensors and gives the necessary output using Red, Yellow and Green LEDs.

6.2 DESIGN FEATURES

Our design has following features:

100% Hardware Implementation

We used the entire hardware system to control and regulate the traffic light and used the proximity IR sensors to fetch the input and were connected to the GPIO I/O pins of the Altera® DE1 Board. For the output Red, Yellow, Green LEDs were used to show the traffic control of the road junction.

A New and Improved Traffic Regulation Algorithm

We implemented the regulation and control of traffic system based on the intensity of the traffic intensity on the prototype model. Detection of the vehicle, default state of the traffic with various cases were also considered which includes non presence of the traffic for a particular road, equal intensity of the traffic and priority given in that situation, preference being given to the main road with high traffic intensity most of time and many other cases have also been considered. New algorithm using the hardware interfacing will not only increases the liquidity of the traffic but will also be real time based and the updated traffic can be easily controlled.

Use of Optimum FPGA Resources

This Design use the Altera® Cyclone® II EP2C20F484C FPGA high capacity device. The whole system uses a total 153 logic elements and 40 dedicated logic registers. In all, 34 pins were used to interface the entire hardware with the FPGA.

6.3 SOCIAL OUTCOME

This new algorithm has a positive social impact as well. If implemented, this new model is expected to help regulate the city traffic a lot. People will spend less time travelling, thus saving a lot of money, time and frustration as well. This will also result in lesser noise and air pollution.

In sum, this project is not only advanced technically, but has a good social outcome as well.

REFERENCES

- [1] *FPGA-Based Advanced Real Traffic Light Controller System Design*, WM El-Medany 1, MR Hussain, IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 6-8 September 2007, Dortmund, Germany
- [2] *FPGA Implementation of Advanced Traffic Light Controller using Verilog HDL*, B. Dilip, Y. Alekhy, P. Divya Bharathi, International Journal of Advanced Research in Computer Engineering and Technology, Volume 1, Issue 7, September 2012
- [3] *Wireless Visual Sensor Networks for Urban Traffic Management*, NIST Technology Innovation Program, Professor Al Bovik, The University of Texas at Austin
- [4] Cyclone® II Device Handbook, Volume 1 by Altera® Corporation
- [5] DE1 Development and Education Board User Manual, Version 1.1, 2006 by Altera® Corporation
- [6] Using ModelSim Guide, EECS150 Fall 2004, University of California at Berkeley
- [7] Vehicle Detector Guide by T Kon, 1998, <http://scholar.lib.vt.edu/theses/available/etd-43098-201311/unrestricted/APPENDIX-B2.PDF>
- [8] The "SCOOT" Urban Traffic Control System by Driver Information and Traffic Management Division, Great Minster House
- [9] Development of A Traffic Light Control System Using Programmable Logic Controller by Azrulnor Bin Ahmad
- [10] Adaptive Traffic Signals, Comparison and Case Studies by Kevin Fehon.
- [11] SCOOT - Split, Cycle & Offset Optimization Technique by Ian Day (SiemensAG, Traffic Control Systems Division).
- [12] Traffic Advisory Leaflet 7/99 April 1999 by Traffic Management Division, Department for Transport, England
- [13] Wikipedia, http://en.wikipedia.org/wiki/Traffic_light
- [14] *Verilog HDL* by Samir Palnitkar, SunSoft Press, 1996

Brief Bio Data

Mayank Prasad (09BEC233) is fondly known as the creator of maxEmbedded.com. He is pursuing his final year of B.Tech in Electronics and Communication Engineering and minor in Industrial Automation from VIT University, Vellore. He has been the President of roboVITics - the Robotics Club of VIT and the Founder/Vice-President of TEC - The Electronics Club of VIT. His interest lies in the fields of analog/mixed-signal processing, digital systems, real-time processors, computer architecture, computer vision and VLSI. He likes to write tutorials for newbies and loves to teach others. In his spare time, you will find him with his iPod.



Sangeet Saurabh (09BEC253) is currently pursuing his B.Tech in Electronics and Communication from VIT University, Vellore. He has served as the President of Instrument Society Of India (ISOI) – VIT Chapter. His area of interest lies in Analog Circuit

Design and Mobile Communication. Apart from spending a lot of time Googling he spend his day interacting with students and reading novels.

