Traffic Light System

Lecturer in-charge: Prof. Ali Hayek
Contact email: Ali.Hayek@hshl.de

Nnaemeka Valentine Eze **Onyesi John Abiagam** **Manoj Luitel**

1 Abstract

.

This document presents the design and implementation of a traffic light controller using the Nexys A7 FPGA development board. The project aims to manage traffic at a four-way intersection, incorporating signals for both vehicle and pedestrian traffic. The system employs three LEDs each for North-South (NS) and East-West (EW) vehicle traffic, one LED each for NS and EW pedestrian signals, and two push buttons for pedestrian crossing requests. Additionally, the design includes set/reset, clock, and JTAG interfaces for enhanced control and programming capabilities.

The traffic light controller is developed using KiCad for schematic capture and PCB layout. The FPGA on the Nexys A7 board is programmed using VHDL on Xilinx Vivado to handle the sequencing of traffic signals, ensuring a safe and orderly flow of vehicles and pedestrians. Key design considerations include the efficient placement of components, ensuring minimal signal interference and adherence to design rules. This document outlines the schematic design, PCB layout, and the necessary steps to integrate all components and program the FPGA, providing a comprehensive guide for replicating and understanding the traffic light controller system.

2 Introduction

Hardware engineering, particularly with relation to FPGAs, ASICs, DSPs, and other components, has grown to be an important part of engineering solutions due to the growing requirement for significant system integration and testing. This has necessitated our research in designing a traffic Management System using VHDL, Modelsim, and the Xilinx Nexys A7 board.

2.1 Motivation

Traffic flow was managed by humans, prior to the invention of traffic light systems. This may be devoted staff members who, at their discretion, decide which car should pass, or it could be drivers who determine when it is safe to move. The choice of who gets the passing right

depended on the individual. The total flow of traffic is therefore dependent on humans. As a result, human dependence on the total traffic flow makes it labor-intensive and prone to error.

Automated traffic control systems replaced these. It is a more effective way of managing traffic flow with better precision and reliability. Before traffic system timing is designed, a traffic count is usually done. This involves a feasibility check on the intensity of traffic at different sections of the road, especially at the peak period. This eliminates human error, making it more reliable. The overall requirement and need of a smart traffic system is that of safety[2][3].

2.2 System Overview

3 Concept Draft

The goal of this project is to provide a dependable traffic system that will allow cars and pedestrians to cross intersections safely. It is conceptualised, designed based on Moore finite state machine. The realised FSM is programmed in VHDL simulated and deployed on FPGA. The hardware is realised on PCB using kiCad.

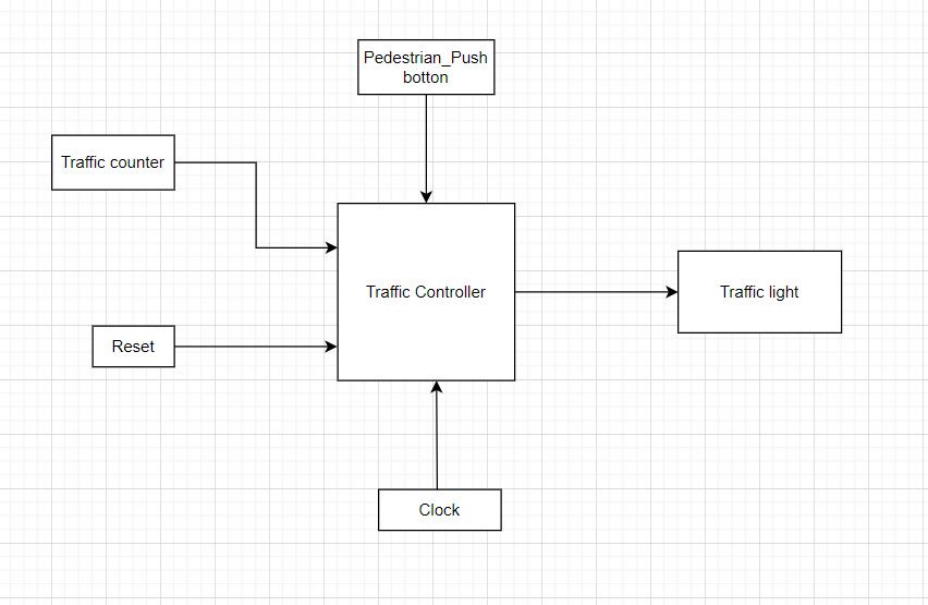

Figure 1: Block Diagram

4 Project/Team management

Agile method became the ideal team methodology. A subset of it Scrum was actively deployed all through the task, where we are all clear on what to do next and collectively work towards realising them, in the coming weeks. Each team member leads in different sprint session, sub-tasks shared. We brainstorm each week on the tasks at hand especially on the initial concept and idea, while the weeks' leader co-ordinate the task.

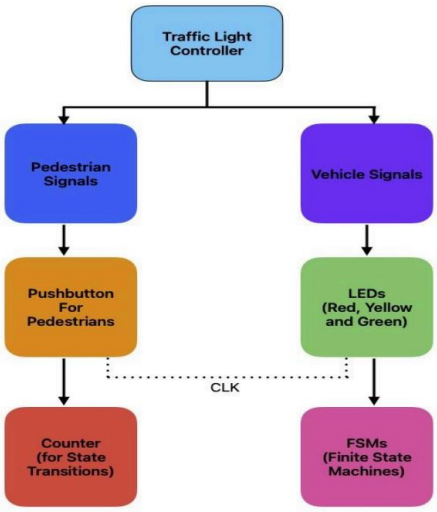

Figure 2: Concept Draft

5 Technologies

VHSIC Hardware Description Language and Testbench

5.1 FPGA

FPGA board Nexys ARTIX-7 100T CSG324 The Xilinx FPGA board Nexys ARTIX-7 100T is a development board designed to facilitate the implementation and testing of digital designs using Field Programmable Gate Array (FPGA) technology[1]. It features the Artix-7 FPGA, which is a high-performance and low-power FPGA family. The Nexys ARTIX-7 board offers a range of features and peripherals that make it suitable for a wide range of applications. It includes a powerful FPGA chip with 101,440 logic cells, 15,850 slices, and 240 DSP slices, allowing for complex digital designs to be implemented. The board provides various I/O interfaces, including USB, Ethernet, VGA, HDMI, audio, and several PMOD connectors, which enable easy integration with external devices and communication interfaces. Additionally, it is equipped with a 128 MB DDR3 memory module for data storage and processing. The Nexys ARTIX-7 board supports programming and configuration through Xilinx's Vivado Design Suite, a comprehensive development environment for FPGA design. It allows users to write, simulate, synthesize, implement, and generate bitstreams to program the FPGA chip on the board.

Timing Analysis

6 Implementation

6.1 Modelsim

In Hardware Description Language(HDL), Modelsim is used in debugging and verification of design logic and features. Its role in the development of Finite State Machine(FSM) can not

be over emphasised. This is because it is used to ensure the correctness, and reliability of the design before it is implemented on the actual hardware.

In this project, the VHDL codes were first written and simulated on Modelsim. This helped us correct some dependencies on various sub-models, before deploying on FPGA board. The various testbenches; Counter, Register, Finite State were all simulated as shown below. The various models of the system were simulated and verified as follows 3

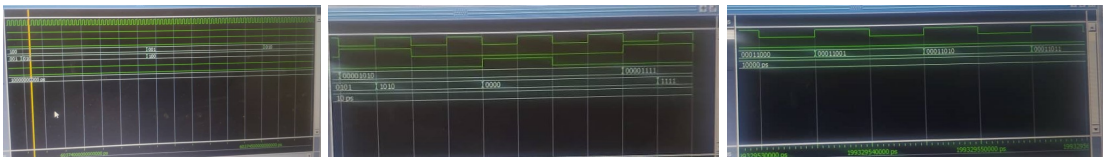

Figure 3: VHDL Testbench

6.2 RTL Synthesis

In digital design, RTL synthesis is used to transform an Register Transfer Level description of a digital circuit into a gate-level representation. This could be realised on Synthesis tool like Xylinx FPGA board. RTL is a HDL of higher abstraction level, and as such it is easier to design and easily read by human. The synthesis tool further analyzes the RTL description and performs optimizations in other to generate an optimized gate-level netlist, focusing on constraints like area, power, and timing. It also maps the RTL code to the target technology library, resulting in a gate-level netlist representing the circuit in terms of gates, flip-flops, and interconnections. In addition, it creates a gate-level netlist which reflects the circuit as a set of gates, flip-flops, and interconnections by mapping the RTL code to the target technology library as shown below;

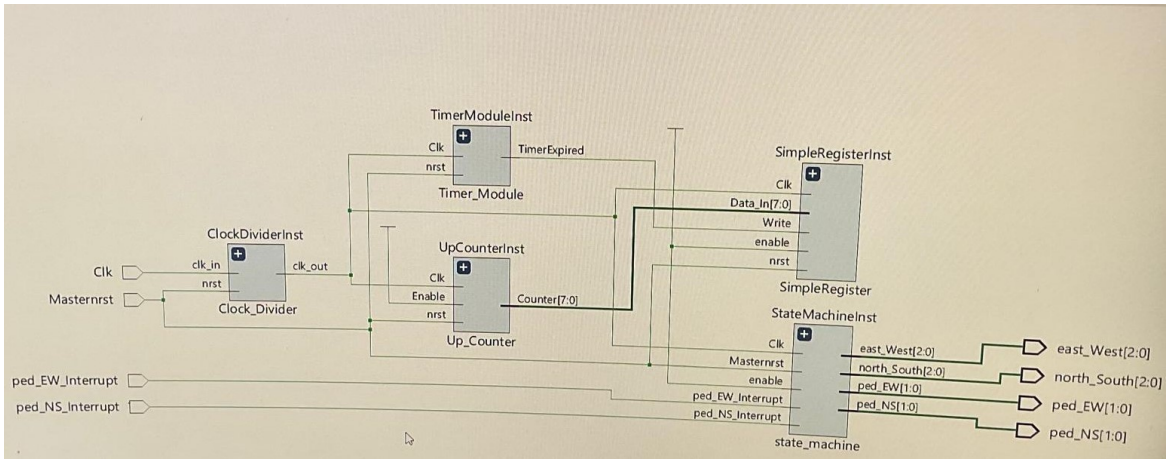

Figure 4: RTL Schematics

6.3 Xilinx - Vivado

Vivado is a synthesizing environment used to implement, design, and generate the bit stream of VHDL code that is uploaded to the FPGA board. This code is needed to carry out the intended systems logic as shown below;

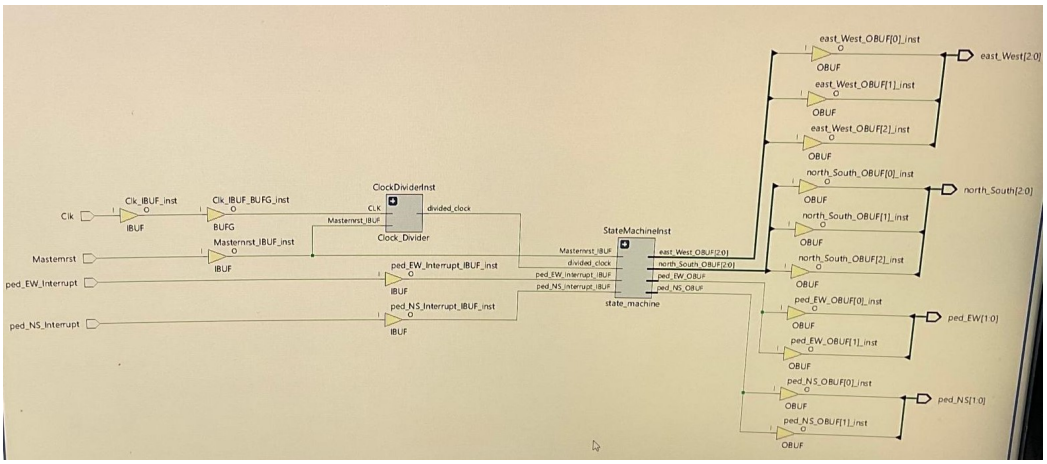

Figure 5: RTL Schematics2

6.4 PCB Schematic and Layout

KiCad

LEDs and push buttons are placed in accessible and logical positions Traces are kept as short as possible to reduce noise and interference Appropriate trace widths for signal and power lines are used. Proper grounding and power distribution are ensured. Component Placement are as follows: LEDs:

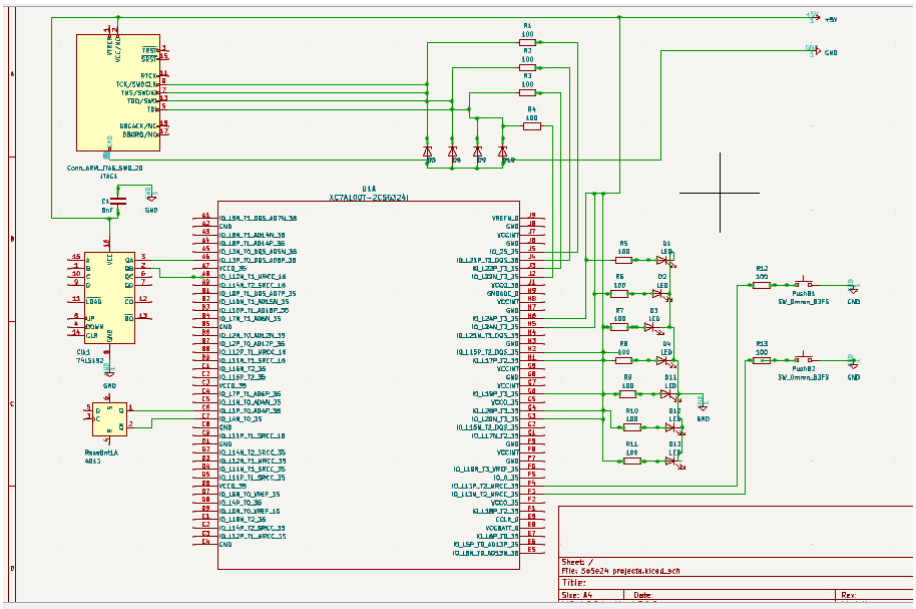

Figure 6: PCB Schematic

Group NS vehicle LEDs together near the edge of the board for visibility Group EW vehicle LEDs similarly on the opposite side Pedestrian LEDs are near their corresponding vehicle LEDs Push Buttons: NS and EW pedestrian request buttons near the edges of the board for easy access Interfaces: Set/reset pins, clock interface, and JTAG connector in easily accessible locations Routing Power (VCC) and ground (GND) lines first routed. Signal lines from FPGA to LEDs and buttons, ensuring minimal crossing of traces are routed Vias to route signals

between layers if using a multi-layer board are used. The latest version of Layout and the corresponding 3D view are displayed below:

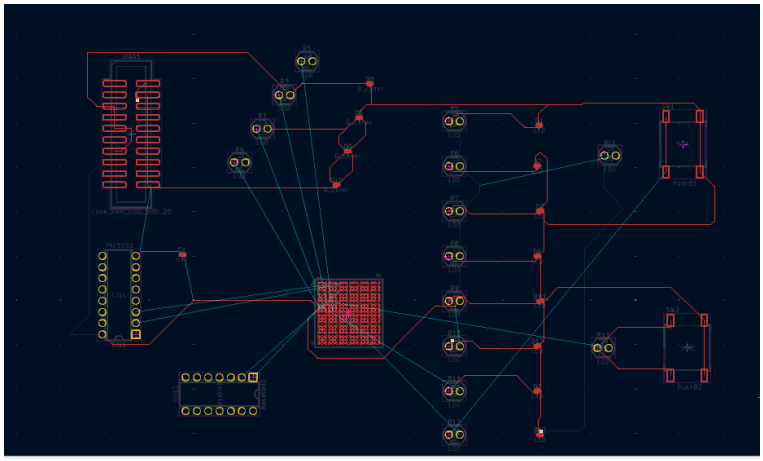

Figure 7: PCB Layout

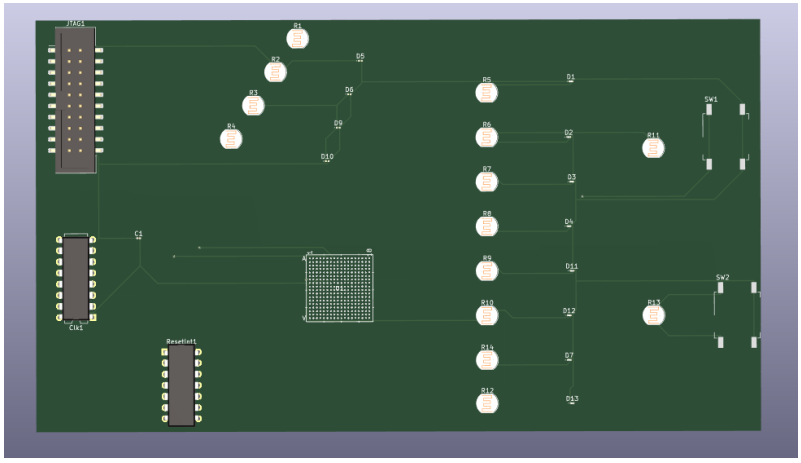

Figure 8: 3D View

7 Use Case

The overall system was first tested in modular forms, and finally the holistic system aswell. This involves testbench as above 3 and on FPGA board. The outcome were satisfactorily realised, but could be improved.

The few limitations were recorded by the timing constraints. The transitions could be better improved by clock divider when added to the top module. Overall, the project realisation is suited for the intended task of reliable and safe traffic management with minimal error; and a call for deeper research into real-time traffic management with multiple intersection points.

References

[1] Digilent – start smart, build brilliant.

[2] Luís Conde Bento, Ricardo Parafita, and Urbano Nunes. Intelligent traffic management at intersections supported by v2v and v2i communications. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1495–1502. ISSN: 2153-0017.

[3] Ahmed S. Salama, Bahaa K. Saleh, and Mohamad M. Eassa. Intelligent cross road traffic management system (ICRTMS). In *2010 2nd International Conference on Computer Technology and Development*, pages 27–31.