Traffic Light Controller

List of Authors (Joseph Hill, Aldo Gega, John Brooks, Nashwan Marcos)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: josephhill@oakland.edu, agega@oakland.edu, johnbrooks@oakland.edu, nashwanmarcos@oakland.edu

***Abstract*—The purpose of this project is to design traffic lights for a 4-way intersection using the principles of digital logic design. VHDL will be used to describe digital signals and implement the state and control into Nexys FPGA board.**

I. Introduction

The intention of most school related project is to challenge students with real world applications intended to further their knowledge and enhance their skills. The intentions for this project are to explore the design and implementation of a 4-way traffic light controller. This involved the decision of a project idea, planning and capturing the thought processes from what we have learned and what we were able to bring to the table from our individual experience. The most challenging part in the design process is the control of the different states using counters, LUTs and the logic behind it. Our design should be a basic prototype of a working system with room for improvements. We can't avoid traffic jams and busy intersections, but we can always add smarter logic systems that will aid in traffic flow making it smarter, safer and more adaptable to growing infrastructure.

II. Methodology

A. State Table.

The very first step in the design approach was to create the state table to have a better understanding of the different stages and the cycle time for each stage. This approach gives a solid idea about the type of counters and the amount of wait time needed. This can also be expanded for more states and adopted to different cycle times if necessary.

North / South			East / West			Time (s)
R	Y	G	R	Y	G	
0	0	1	1	0	0	10
0	1	0	1	0	0	3
1	0	0	1	0	0	1
1	0	0	0	0	1	10
1	0	0	0	1	0	3
1	0	0	1	0	0	1

Table 1: Simplified State Table for the different stages

B. Top Level Overview.

Our VHDL approach was to create one main top file with three sub modules. The first module brings the clock down from 100 MHz to 1 Hz. The second module counts the time for each cycle. The third module has a rolling counter to loop through the different stages. LUTs are used to control the outputs and the time for next stage.

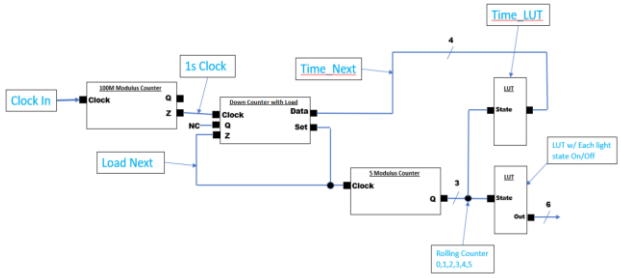

Figure 1: Top Level overview diagram

III. Experimental Setup

The prototype design was written in VHDL using *Vivado*. The *GitHub* code repository was used during development to allow for version control and each person could update the design from their own remote location.

The design was implemented on a Nexys 4 FPGA and tested on both the *XC7A100T* and *XC7A50T* models. An external board with LED soldered to it was used to demonstrate the behavior of the design and provide a visual representation of the intended concept.

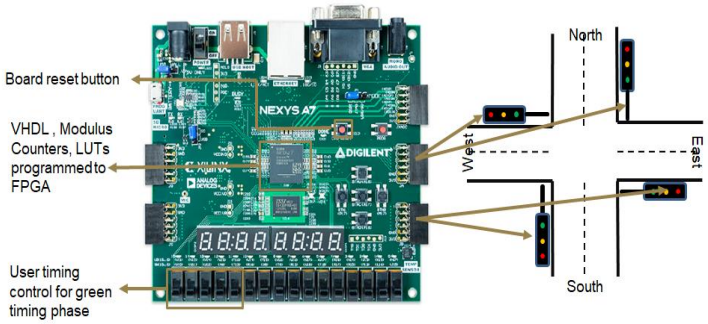

Figure 2: Design and Implementation Overview

IV. RESULTS

The implemented design was observed while running on the FPGA to verify that the system was operating as intended. It was immediately noted that the system was failing to achieve the designated traffic light hold pattern. The original settings called for a ten-second red/green light, two-second yellow light, and a one-second overlap of red lights during transition. The original behavior of the system, with run times falling short of the desired values, can be seen at:

https://www.youtube.com/watch?v=OjH88N4XZc0&feature=youtu.be

After re-synthesizing the code several times on both models of the Nexys 4 available to the team, even more erratic behavior was observed. In all cases the system failed to keep proper time, but in some cases the shorter-duration yellow light would be skipped entirely in a cycle.

Finally, the bug in the system was resolved. It was discovered that the Z-signal of the counter was wired like a mealy state machine so at every time that the registers of the counter are equal, the final value will send out a pulse. As registers increment, a glitch is captured which triggers an error pulse which appears prior to the intended pulse thus causing timing to be completely of from the time set value. This issue was corrected by moving the output z signal into the synchronous logic within the clock event process.

CONCLUSIONS

The original design concept was proven to be functional after minor modifications. The design was physically implemented and validated for all programmed time settings. It is expected our system will hold in operation for as long as required within the hardware's life expectancy.

Based on the reliable functionality of the final product, it is reasonable to consider adding further improvements. The most obvious next step for our system would be implementing time-specific operating modes, including a "night mode" with flashing yellow and red lights to optimize late-night traffic flow. More advanced improvements which may complement the current design include smart detection of traffic flow via machine vision, and closed-form feedback to monitor functionality.

REFERENCES

[1] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, 3rd ed., New York, NY: McGraw-Hill, 2009.

[2] http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html

[3] https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/reference-manual