Traffic Light Controller

VGA Lights with Crosswalk Systems

Ahmad Abbas, Camille Bilby, Connor Goetz, Mustapha Abbas

Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI

e-mails: ahmadabbas@oakland.edu, mustaphaabbas@oakland.edu, cbilby@oakland.edu, connorgoetz@oakland.edu

***Abstract*—The purpose of this project is to design a traffic light controller for a 2-way intersection road. The digital system implemented seeks to control two stop lights and two crosswalk signals by applying the principles we have learned in digital logic design class. VHDL will be used to within the Vivado design suite to create the circuit. The project will then be implemented on a Nexys Board, utilizing four seven-segment displays for the crosswalk, and a VGA port for the traffic lights.**

I. INTRODUCTION

Traffic light systems and their employment are extremely popular methods to regulate traffic flow within busy intersections. Most intersections consist of two perpendicular roads coming together to form a cross. Thus, in order to prevent accidents, traffic light systems must permit vehicles to proceed from two opposing directions while signaling the two corresponding perpendicular directions to remain stopped. Alongside this, pedestrian walk signals are often utilized to show pedestrians when it is safe to cross, and many modern systems display how many seconds remain for them to cross. These systems can be implemented within a digital logic circuit.

By utilizing a Nexys FPGA board, a circuit can be designed and implemented which will logically control the stoplights as well as the crosswalk signals. This provides an example of how everyday systems are derived, and showcases a cost-effective solution to one such system. The crosswalk "count-downs" are displayed on four of the board's included seven segment displays. By utilizing a computer monitor and the board's VGA port, the stoplights are displayed. These are clearly not the hardware which would be used in a real-world implementation, but the implementation could be adapted to different output devices.

II. METHODOLOGY

As with most real-world problems that can be solved with digital logic, many different possible solutions exist. By using the techniques learned throughout the Digital Logic Design course, an approach and its execution were able to be determined to create a traffic light system. Within the selected approach, four main tools were acknowledged as integral parts of the final circuit. These included the timing system, the crosswalk displays, the FSM, and the VGA represented traffic signals. The following was the resulting circuit diagram:

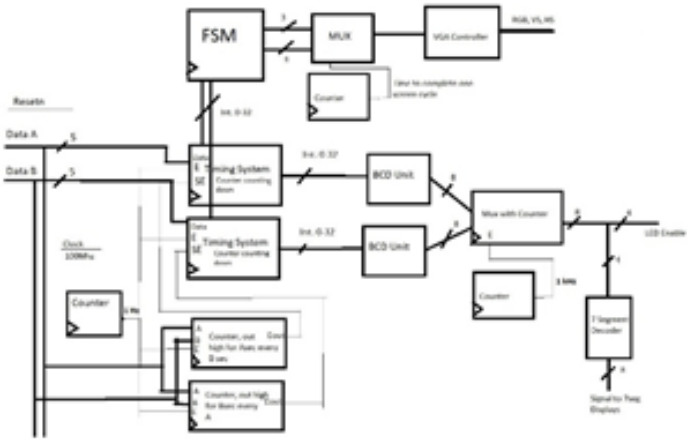

Figure 1: Circuit Diagram

A. Timing System

Arguably the most important part of the traffic system, the timing system is what ensures the traffic lights are displaying the right colors, and that the crosswalk signals are showing accurate and continuous countdowns.

In order to create a system which could be applied universally, it was determined that a user should be able to enter how long each light is allowing cars to pass (green and yellow light length). To give users this ability, ten

switches were assigned as input data, resulting in two five-bit numbers, one for each light length. Both of these data values enter two counting devices. Using some process logic, these counting devices were encoded so that they would emit an enabling signal for one data input's length in seconds, and then stop enabling for the other data input's length in seconds. Hence, one unit sent an enabling signal out for length A seconds, and then disabled for B seconds while the other unit sent an enabling signal out for B seconds while disabling for A seconds. It must also be noted that this second interval signal was created with a counter emitting a 1 Hz signal, thus creating one second pulses.

Finally, these enabling signals were sent to counters which relied on the 1 Hz signal as an enabler, as well as the previous signals to operate. These counters each controlled one crosswalk and its corresponding traffic light by counting down from the user inputted time in seconds. Note that the range of a five-bit number is 0-31, however within the logic of the timing system unit, a one second addition was added to ensure no light would continuously be red incase of a user error. This means that the counting system units emit an integer of range 0-32. This integer signals can then be processed by the other components.

B. Crosswalk Displays

With the integer inputs from the timing systems, all of the necessary inputs are already in place to establish crosswalk signals.

By using BCD units which accept integers as an input, both crosswalk display numbers can be represented by eight bits, four for each singular decimal number. These four numbers can then be multiplexed onto four seven segment displays. By using a multiplexer that cycles through each input based on a 1 kHz signal generated by a counter, the four displays can be assigned so quickly that the human eye cannot perceive that only one display is truly on at any given time. These provide the crosswalk signals which countdown, showing pedestrians how much time they have remaining to cross.

C. FSM Unit

In order to correctly assign a red, green, or yellow light, a FSM was utilized as it can provide a fine-tuned control of the emitted signals to the VGA display component. The state machine implemented has five total states as shown in the diagram below:

Figure 2: FSM Flow

Each state provided necessary 3-bit RGB values that were eventually implemented through the display.

D. VGA Traffic Signals

As noted with the FSM sub-section, two 3-bit RGB are produced by the FSM in any given state. These values are then passed into a multiplexer. The multiplexer select line is controlled by a timer which divides the amount of time necessary for an entire scan of the screen through a VGA output into an area to represent the first light and another area to represent the second light. This VGA output circuit came from source [1], and originally allowed for the background to be entirely one 3-bit color. In this way, both lights can be observed easily in order to determine whether the circuit is operating correctly.

III. EXPERIMENTAL SETUP

Thus far, testing has been done through both simulation and hardware implementation. A test bench was created to verify that the crosswalk signal countdown circuit was working correctly:

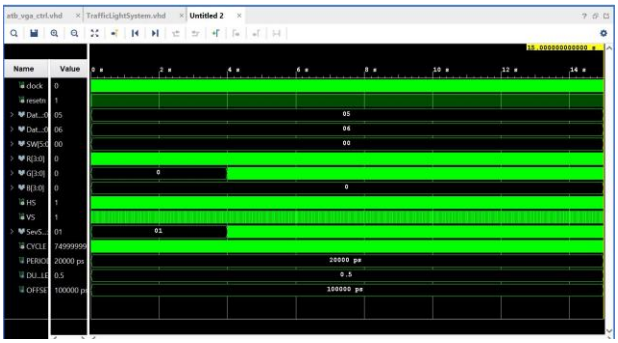

Figure 3: Full Simulation Wave

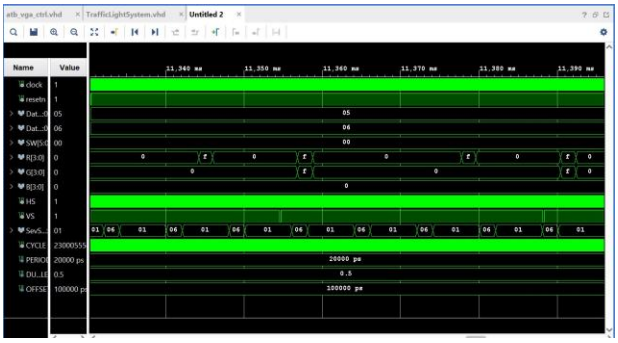

Figure 4:"Zoomed" in Simulation Wave

However, it must be noted that the timers had to be accelerated to a faster pace, namely from one second to a microsecond pace so that the simulation could be completed in a timely manner. It was repeatedly found that, when loaded onto the board itself, the timers would behave unexpectedly. As an example, the counts would sometimes stop at one, rather than progressing to zero. After simulation, the designed file was implemented onto a Nexys board where the seven segment displays showed that the circuit was operating as expected after some trial and error.

The core experimental setup of the project was using the Nexys A7-50T: FPGA Trainer Board in conjunction with the Vivado 2018.3 HL WebPACK program by Xilinx. The crosswalk function was the first one created, which was then tested with a test bench to verify that the crosswalk signal countdown circuit was working correctly. The final result of the system counted down in seconds, however this pace was accelerated in the testing phase to microseconds so that the test bench simulation could be completed in a timely manner for more efficient debugging. This turned out to be fruitful, as there were a few problems encountered, with the timers behaving unexpectedly and having to be redesigned and tweaked. As an example, the countdowns would begin to reduce in size with each repeating cycle they went through, if one was originally set to count down from 12 seconds, on the next cycle it would count down from 11, then on the next from 10, and onward. The issues being faced turned out to be in a different file than expected, in the timing system itself rather than the crosswalks' counter modules, and having a swift test bench to debug with vastly helped the process of fixing this issue. When testing on the physical model the clock was set to 1 MHz, while the testbench clock was set to 100 MHz to manage Vivado's simulation time [2]. The final results were satisfactory, and the countdown clocks were precise and functioned as expected. This was the most important part of the system ensure accuracy with, as this timing system was the backbone of the entire project.

Using a VGA port, the next part of the system was tested and implemented. The timing of the screen's color switching was tested at first only using one color at a time, as the code it was adapted from was created to do. Then the code was modified to split into two separate colors in a split-screen fashion, with the top half of the screen corresponding to the North-South crosswalk, whose timing was dictated by data A, and the bottom half of the screen corresponding to the East-West crosswalk, whose timing was dictated by data B. There were some issues encountered in this particular set of tests. Due to the nature of the VGA screen and the timing of the signals we were sending it, when the colors were split they would bleed into one another upon changing. This defeated the purpose of the screen, as the colors were no longer split into two distinct "lights." This was solved by altering the VHDL code so that there was a gap between the colors, with about the top and bottom quarters of the screen being taken up with the "lights" and the rest of the screen being filled with black. This solved the bleeding issue and made it such that the colors were distinct.

IV. RESULTS

The project functions as we expected, the crosswalk signals where demonstrated using four of the seven segment displays(two displays for each crosswalk) on the Nexys A7 50T board,which counts down from the user inputted times, and are not counting down at the same time (one display is always displaying "do not walk" while the other informs pedestrians that it is safe to walk).we have also used the VGA port to demonstrate the traffic lights on the screen which gave a clear and accurate representation rather than connecting extra components and LED's to our system, which saved time to remain focused on the main idea of our project.

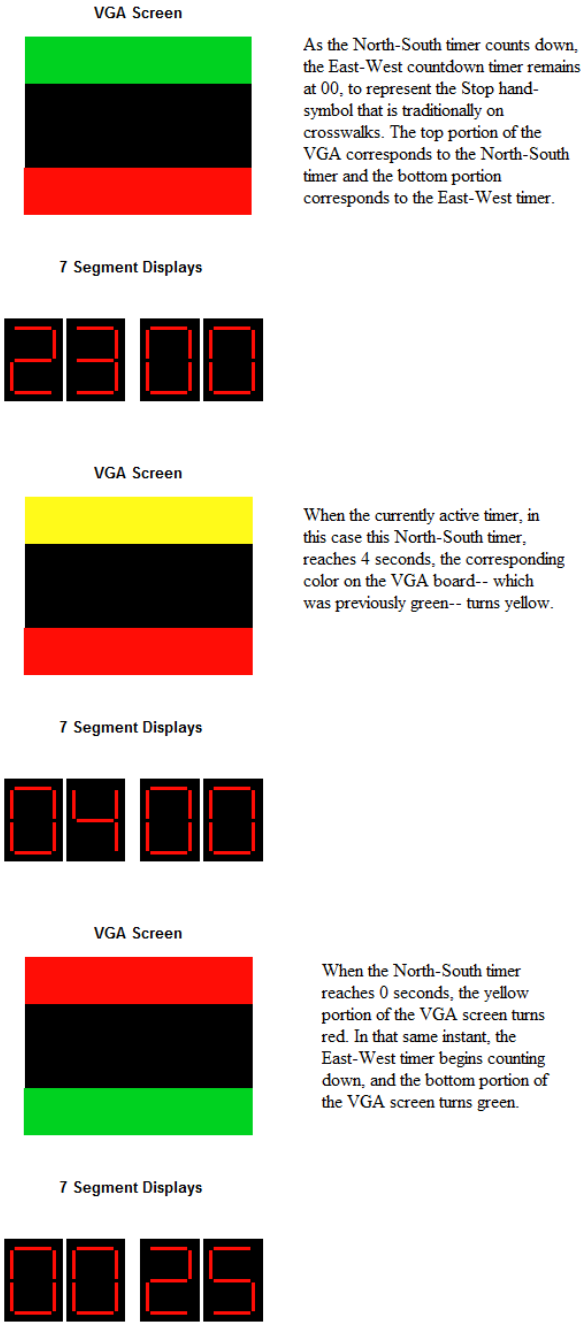

Figure 5: Demonstrated Results

V. CONCLUSIONS

The main takeaway of this project is the ability to combine multiple circuits together and being able to test it and implement it referring to what we have learned in our lab for the digital logic design class. We have learned that just the smallest thing can have a drastic effect. For example, the timers for the crosswalk had to be increased to a faster pace from seconds to microseconds, to complete the simulation in a timely manner. The improvement that we suggest for the system is the ability to detect the volume of traffic, which will be achieved by adding sensors and some components into our system. This would minimize the time people have to wait at a traffic light or when crossing.

REFERENCES

[1] Llamocca, Daniel. "VGA Display Control." VHDL Coding for FPGAs. http://www.secs.oakland.edu/~llamocca/Tutorials/VHDLFPGA/Viva do/Unit_7/vga_ctrl.zi