PWM LED Color Control

Through the use temperature sensors, accelerometers, and switches to finely control colors.

Daniyah Alaswad, Joshua Creech, Gurashish Grewal, & Yang Lu
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: dhalaswa@oakland.edu, jmcreech2@oakland.edu, gsgrewal@oakland.edu, & yanglu@oakland.edu

***Abstract*— Special effects lighting is a significant section of the entertainment and advertising industries. This project uses pulse width modulation (PWM) signal to control the brightness and the on and off states of an LED light source. This idea can be employed in several devices to develop several LED lighting effects for different purposes. Moreover, this project represents the functional usage of the Nexys 4 DDR FPGA board using the Vivado VHDL format. Switches, accelerometer, and temperature sensors are all used to control an RGB LED. The switches allow the user to control the color and brightness of the LED. The accelerometer will communicate with the board and turn a white LED color and control its brightness. Finally, the temperature sensors adjust the brightness of the white LED. This project will benefit the user in the case of overheating or device rotation.**

I. Introduction

This final project report describes the final design and implementation of color control using an RGB led with PWM. In addition, it shares our findings related to FPGA development and the process of debugging.

The scope of the project was how to program a Nexys 4 DDR board so that it understands its orientation, surrounding temperature, and which switches are on/off. In this project, we develop a system that can use temperature sensors, accelerometers and switches to control the colors of LEDs. To realize it, we use language VHDL to program the Nexys 4 DDR board. The motivation for this project is to turn on LED's so an object can know its orientation when moving in a three-dimensional space, know when something is getting warm, and use switches to adjust to different colors. A main use of this is when you're using a tablet and are shifting from landscape to portrait. It will shift with the force of gravity on the accelerometer. This project uses a ton of topics covered in this course and the previous Digital Logic Design course. A few are, a shift register, FSM (Finite State Machine), counter, accelerometer, and the seven-segment display. Throughout the two courses we have touched up on all these various topics that are being used in this project. In some cases, there needed to be more research done on a certain topic in order to get a better understanding of it. The project presents details of the methodology used in the design and development of the lighting device. It also demonstrates the experimental setup that led to the creation of the device [4]. Finally, the results of the device are discussed and concluded. It is imperative to mention that this project utilized several topics and ideas covered in class, including Finite State Machines, Digital Logic Design, a shift register, and seven segment display.

II. Methodology

This project utilized the two primary technologies: light emitting diode (LED) and pulse-width modulation (PWM). A pulse-width modulation signal was used to control the light intensity and duration of the LED [5]. Throughout the methodology section, the image below will be referred to. The numbers on the image will retain to a section.

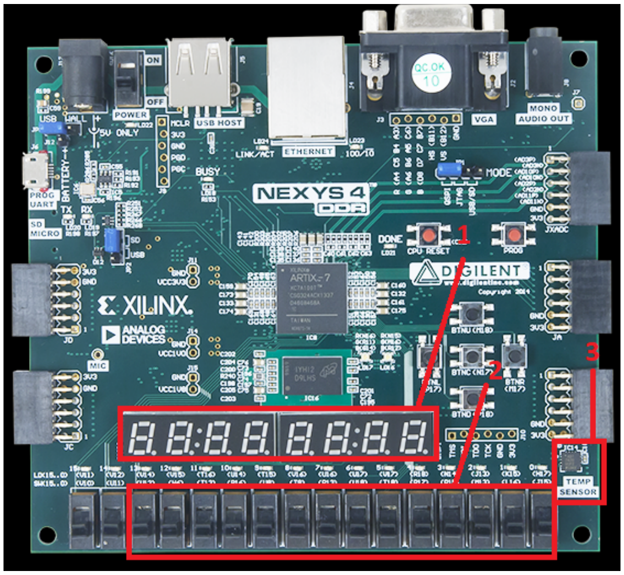

Figure1: Nexys 4 DDR board.

A. Understanding of LED's

Several light colours can be generated using an array of green, blue, and red light emitting diodes (LEDs). This effect can be obtained by varying the light intensity of the three coloured light emitting diodes. The light intensity of the LEDs is achieved with pulse width modulation (PWM) [4].

The Nexys 4 DDR board contains two tri-color LEDs. Each tri-color LED has three input signals that drive the cathodes of three smaller internal LEDs: one red, one blue, and one green. Driving the signal corresponding to one of these colors high will illuminate the internal LED. The input signals are driven by the FPGA through a transistor, which inverts the signals. Therefore, to light up the tri-color LED, the corresponding signals need to be driven high. The tri-color LED will emit a color dependent on the combination of internal LEDs that are currently being illuminated.

B. Switches (2)

The switches on the Nexys 4 DDR are used to control the LED's particular color and which special action the board is looking at. The main point in using switches is to allow easy access to the user to set the board to what is needed. The switches allow the user to quickly edit the functions of the board. The switches are set up in a binary manner with certain ones pertaining to actions and other pertaining to the LED's on the board. When certain switches are flipped certain actions will be taken by the board.

In this project, 12 switches will be used to control the color and brightness of the LED, and two switches will be used to enable and disable certain sensors. When the switch inputs are all zero, besides switch 13, the accelerometer will control the brightness of a white LED. When the same is done with switch 12, the temperature sensor will control the brightness of the white LED. Switches 0-3 are used to control the blue LED. Switches 4-7 are used to control the green LED. Finally, switches 8-11 are used to control the red LED. The lower the switch number in its particular color, the lower the brightness of that color will be and vice versa.

The switches allow the user to control the color and brightness of the LED. One of the switches is for the temperature and another one is for the accelerometer. The color and the brightness of LED can be changed by using switches.

C. Accelerometer

The Nexys 4 includes an Analog Device ADXL362 accelerometer. The ADXL362 is a 3-axis MEMS accelerometer that consumes less than 2 µA at a 100 Hz output data rate and 270 nA when in motion triggered wake-up mode. The accelerometer is built into the Nexys 4 DDR board and detects on the x, y, and z axis. This will allow the user to know which way the board is by looking at the color of the LED.

By implementing certain colors to the certain plains, the user will quickly and simply know which orientation the board is. This operation will be very useful for if it was attached to a tablet and one switched from portrait to landscape. This way the user will know it picked up the new orientation. The FPGA can talk with the ADXL362 via SPI interface. While the ADXL362 is in Measurement Mode, it continuously measures and stores acceleration data in the X-data, Y-data, and Z-data registers.

The ADXL362 acts as a slave device using an SPI communication scheme. The recommended SPI clock frequency ranges from 1M Hz to 5MHz. The SPI interface operates in SPI mode 0 with CPOL = 0 and CPHA = 0. All communications with the device must specify a register address and a flag that indicate whether the communication is a read or a write[3].

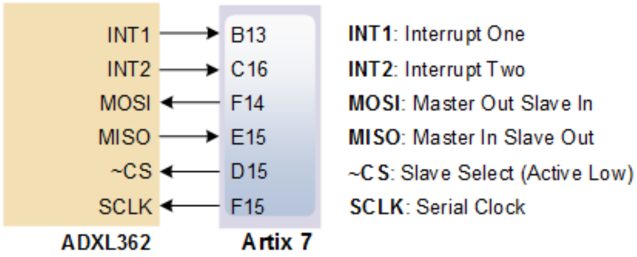

Figure 2: ADXL362

D. Temperature Sensors (3)

The Nexys 4 includes an Analog Device ADT7420 temperature sensor. The sensor always provides a 12-bit output resolution with a typical accuracy better than 0.25 degrees Celsius. The temperature sensors are also built into the Nexys 4 DDR board. When the ADT7420 is powered up, it is in a mode that can be used as a simple temperature sensor without any initial configuration. These sensors will be able to warn the user if the board gets above a certain temperature. The sensor will communicate with the LED's and adjust the brightness of the LED based on the temperature it is at. As soon as the temperature reaches an unsafe digit, the LED's will blink in order to warn the user about the issue. The ADXL362 chip acts as a slave device using the industry standard I2C communication scheme. To communicate with ADT7420 chip, the I2C master must specify a slave address (0x4B) and a flag indicating whether the communication is a read (1) or a write (0). The interface between the FPGA and accelerometer can be seen in the figure below [3].

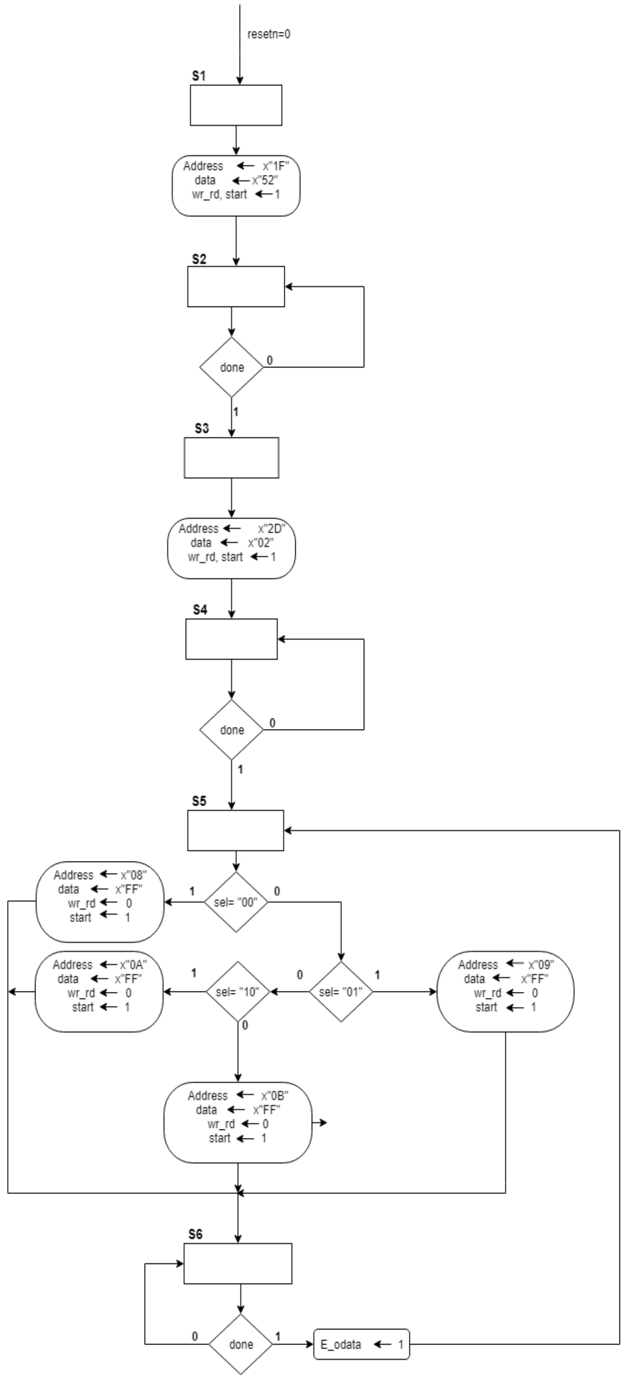

Figure 3: Accelerometer FSM

The basic function of the above stated Accelerometer FSm is to store values at address stated in the FSM diagram. At intial stage, the program communicates with the SPI and when the process is done, it commands to configure 2 ADXL362 registers with initial address sated in the FSM. The next step is to check the switches. The program stores and displays the 8-bit X, Y & Z-axis values on the seven-segment display. In the figure above, address x “08” is for X-axis, x “09” is for Y-axis, x “0A” is for Z-axis and x “0B” for the status register. This 3-axis device operates as a SPI slave device. We read/write data via a register-based interface: we can write/read a byte or many bytes per bus transaction.

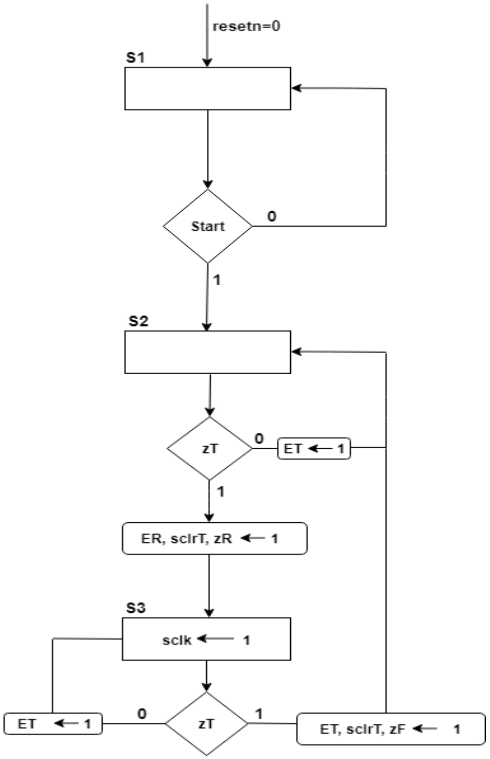

Figure 4: Synchronous clock

Synchronous clock counter who’s output is used in as a clock input in the read-write register FSM to control the clock.

E. Seven Segment Display (1)

The Nexys 4 DDR board contains two four-digit common anode seven-segment LED displays, configured to behave like a single eight-digit display. In this project, the coordinates of the results in the x, y, and z coordinate system and the value of the temperature will be shown on the seven-segment display.

F. Pulse Width Modulation (PWM)

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on and off by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of the on time is called the pulse width. To get varying analog values, you change the pulse width. If you repeat this on-off pattern fast enough with an LED, the result is as if the signal is a steady voltage controlling the brightness of the LED [4].

In the image below, you are able to see four different duty cycles. The duty cycle translates to the cycle of operation of the voltage being applied. The smaller the duty cycle, the dimmer the LED. The larger the duty cycle, the brighter the LED. The 100% duty cycle relates to the highest switch for each color, whereas the 25% duty cycle relates to the lowest switch for each color.

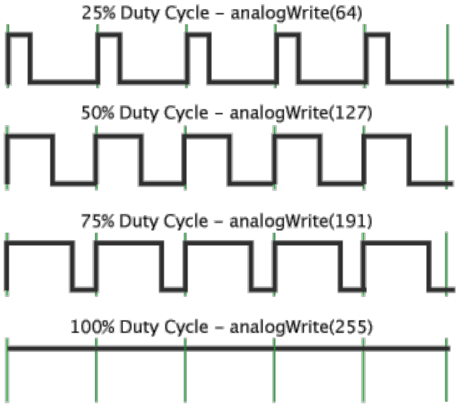

Figure 5: PWM Duty cycle

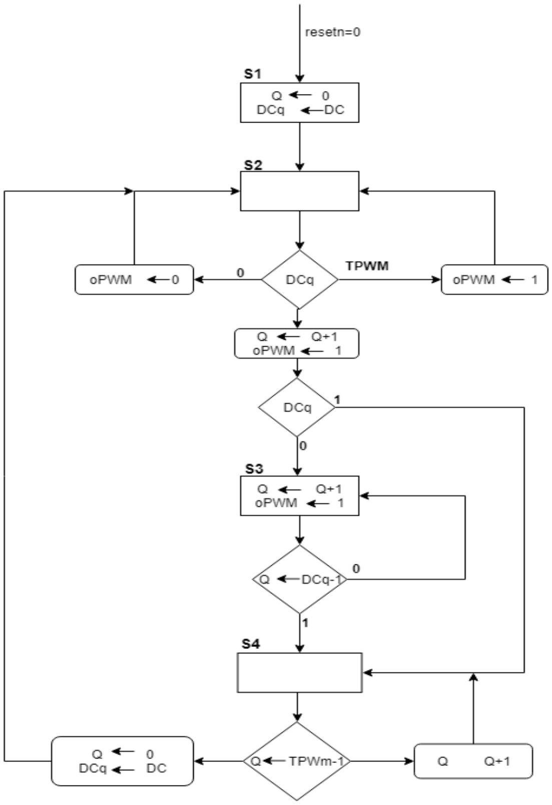

Figure 6: PWM FSM

The FSM shown above is Pulse width modulation control unit. DCq is a number between Time Pulse width modulation and, which converted from TPWM to DC in order to get a high or low signal. If the DCq conditions are true, the code follows as per the FSM. PWM control is a simple SPI, which interfaces data asynchronously and transmits the data frame a specific baud rate. In the FSM above, PWM input conditions are checked which varies between 0-255 and are ceiled between 0-100% of the actual. The conditions in the code are then compared to the input conditions, from where the output "oPWM" is controlled. We generate a square wave where we control the Duty Cycle. Duty Cycle is specified as a percentage: from 0 to 100%. PWM can be used to vary the average voltage on an output pin. This can be useful to control the brightness of an LED, speed of a DC motor, volume of a tone in a speaker, etc.

III. *Experimental Setup*

This project used the following equipment to develop the RGB LEDS: Switches, Accelerometer, and Temperature Sensor on the Nexys 4 DDR. The Nexys 4 DDR board was programmed to control the RGB LEDs using pulse width modulation signal. Every constituent of the project was simulated to ensure they were operating properly. The code was tested before it was implemented into the Nexys 4 DDR. After the uploading was done, testing and observation ensured.

In more details, Vivado 2017 was used to create, synthesize and implement the project design to the Nexys 4 DDR board. The setup and testing followed what was done in lab procedure. Every component of the project was designed and simulated to ensure it was in working order. As soon as the timing and behavioral simulations functioned as anticipated, the code was then implemented into the overall structure. The overall testing was done in chunks as there were multiple different kinds of files. This was done in order to troubleshoot problems instantly as they may appear. Using the test bench, several different operations were established as modules, and from there, the top file was established to combine the operations. From the timing and behavioral diagrams, it was possible to see if any mistakes were in the code or if the logic made sense. In addition, some of the code used and ideas were implemented from previous labs and examples presented by Dr. Llamocca and help from the TA's.

The last setup is the operations we do on the board. We setup the switches for inputs. We will also setup two switches to enable and disable the sensors. The projected results were that tilting the board, flipping switches, and changing the temperature are all factors in certain LED brightness and colors on the Nexys 4 DDR board.

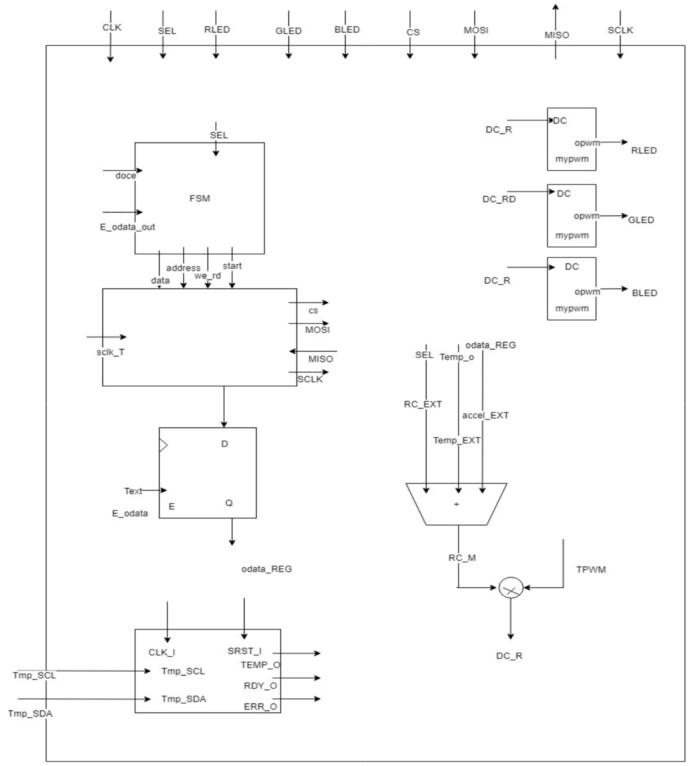

Figure 7: Data path

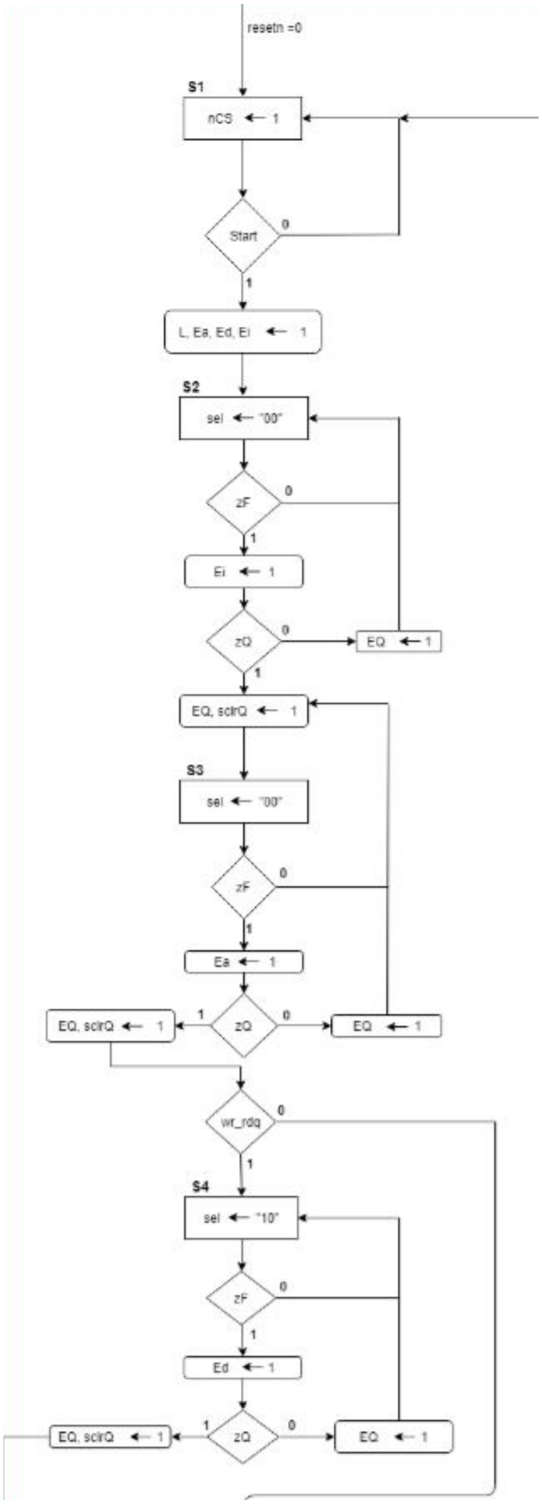

Figure 8a: read-write register FSM

Data path shown above is complete working and circuitry of the program. There are 4 FSMs in the whole program, which includes read-write register, synchronous clock, PWM inputs and an Accelerometer.

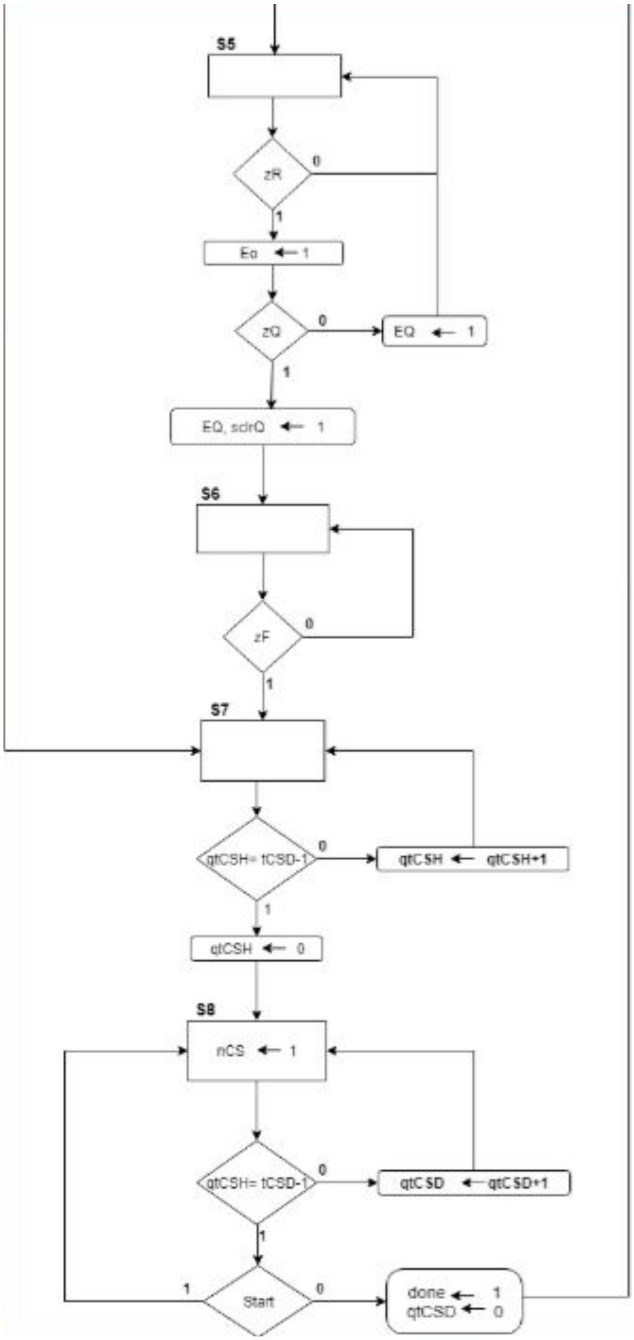

Figure 8b: read-write register FSM (continued)

This circuit handles the SPI communication with the ADXL362. The user provides address, data, and read/write. Then, a read/write SPI transaction is executed. At every transaction, we write or retrieve 8 bits of data. This FSM includes a major part of the outputs coming from the other FSMs created in this project. The main use of this FSM is to read and write location address to memory. wr_reg is the most important it handles the I2C communication based on address, data and write/read decision. Asserting the start signal initiates a transaction. When the operation is completed, the signal done is asserted for one clock cycle. If reading data, it appears on qtCSD, a new transaction can be started on the next cycle after done 1.

III. Results

The results obtained from this project were how to create an advanced vivado project and implement it on the Nexys 4 DDR board. Using the switches to simply control the LED color to using the accelerometer and temperature sensors to adjust the brightness of the LED, this project has gone well.

The testbench was used in order to check if certain parts of the code were working as they should. Every time a section of code was updated, a test was run in order to check if it is correct or not. If it was not correct, one would go back to the code and adjust it until it was correct.

Throughout the entire project the group encountered errors that were able to resolve by other members. These errors were either with mistaken variables or simple wrong doing of the code. The biggest problem the group faced was the 7 Segment Display. This display gave problems once one tried to use it. The failure was overcome once the group realized all one has to do is turn it on and then add the values that were stored in certain registers.

The results of the switches being used are in the image below. This specific image uses switch 0 (dim blue) and 8 (dim red), thus creating the purple color of the LED. This result is exactly as expected when you combine two of the colors together.

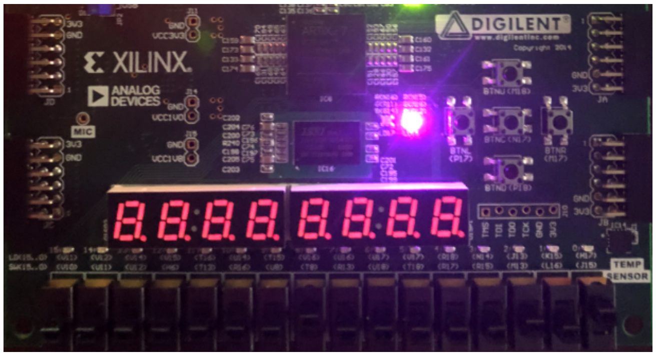

Figure 9: Color of two LEDs combined using switches

The results of the accelerometer were unable to be captured on camera. The results were as expected, and the LED brightness was adjusted when the device was tilted.

The results of the temperature sensor are in the image below. The green LED's are displaying the temperature. The white LED in the center is the LED with PWM that the temperature is able to adjust. This is on a medium brightness as the temperature was moderately warm when the picture was taken. Using an air dryer or ice will adjust the brightness of this LED as demonstrated in the presentation.

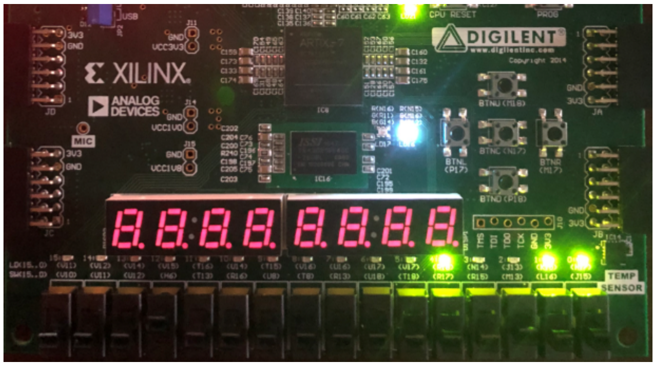

Figure 10: Temperature sensor with blue LED

Overall, the device was able to regulate the light intensity of the connected LEDs using pulse width modulation (PWM). Different lighting effects, including various colours and brightness intensity, were produced.

IV. Conclusions

In conclusion, the group has come to a better understanding of a lot of functions on the Nexys 4 DDR board. The design was successful in that we were able to color control using an RGB LED with PWM. The mode the LED is controlled by the switches. The accelerometer controls and turn certain LED colors stronger. Also, the temperature sensors adjust the brightness of the LED. We implement SPI and I2C communication protocol to communicate with accelerometer and temperature sensor. It realized when moving in a three-dimensional space, the LED's will be turned on, so an object can know its orientation. This project will benefit the user in the case of overheating or device rotation. Besides this, we also well learned about how the SPI and I2C work to communicate with the accelerometer and the temperature sensor. We successfully applied what we learned in class to practice in the real life. Deepen understanding of the use of accelerators and FSM. In this semester, we were focusing on the studying of FSM (Finite State Machine), counter, accelerometer, and we successfully applied these to our design. We did research on those topics and discussed with our professor and TA, which helped us a lot to solve the problems that we met in the process of implementing our design. All experiences and knowledge that we learned will continually be used in the future study and work.

The control of LEDs using pulse width modulation (PWM) generates various interesting effects, which can be used in various applications, especially in advertising and entertainment. This project utilized various concepts learnt from class to generate a lighting device.

References

[1] DeHaan, Mike, et al. "Matrix Mathematics: How Do We Use Matrices In Day-to-Day Life?"*Decoded Science*, 2 Mar. 2018, https://reference.digilentinc.com/learn/programmable-logic/tutorials/start.

[2] Nexys 4 DDR Keyboard Demo. Digilent Inc., n.d. Web. 15 Mar. 2017

[3] Nexys 4 DDR FPGA board reference manual. 11 Sep. 2014

[4] K, Dušan, M. Taraba, J. Adamec, and M. Danko. "Design of a system for the brightness

[5] M, John. "Lighting system with power factor correction control data determined from a phase modulated signal." U.S. Patent 8,963,449, issued February 24, 2015.

[6] Llamocca, Daniel. "VHDL Coding for FPGA's." Reconfigurable Computing Research Laboratory. N.p., n.d. Web. 8 Mar 2018.