RGB LED Control with Temperature Sensor on 7 Segment Display

Kimberly Pawlowicz, Alex Zufelt
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: kapawlowicz@oakland.edu, alexzufelt@oakland.edu

***Abstract*— The purpose of the project was to utilize the ADT7420 Temperature Sensor built into the Nexys 4 DDR board. The output of the Temperature Sensor is in degrees Celsius. Our project gathers this data and displays it on the 7 Segment Displays built into the board. Additionally, we utilized the two RGB LEDs built into the board to correspond with the temperature being read by the sensor. The two RGB's will illuminate Red, Green, or Blue for specifically set temperature ranges in degrees Celsius. This project could be implemented for various applications such as a pool temperature sensing light, Thermocouple cold junction compensation, Industrial control and testing, environmental monitoring and HVAC, or for food transportation and storage.**

I. Introduction

The scope of the project is utilizing the ADT7420 Temperature Sensor built into the Nexys 4 DDR board. The data, in Celsius, is obtained from this sensor and is displayed in two ways. First, the temperature value is indicated using the two RGB LED's on the Nexys 4 DDR board. I.e. Red - High Temperature, Green - Neutral Temperature, and Blue - Cold Temperature. The RGB switches between the three colors depending on the temperature. The second method of displaying the temperature value is utilizing three of the 7 Segment Displays built into the board.

Many topics covered in ECE 2700, Digital Logic Design were utilized to create this project. Concepts and designs from previous labs in the course were implemented for this project. These designs include a Counter, a Finite State Machine (FSM), MUX, Hex to Seven Display, and shift registers. Additional topics that we were required to learn on our own were controlling the RGB LEDs, converting Binary to BCD, and utilizing the ADT7420 Temperature Sensor built into the Nexys 4 DDR board.

This project could potentially be used as a pool temperature sensor. The RGB LEDs would indicate to pool users what temperature their pool is before actually entering it. The Red LEDs would be used for hotter temperatures, typically hot tubs. The Green LEDs would be used for average pool temperatures. Finally, the Blue LEDs would be used for colder temperatures.

II. Methodology

A. Block Diagram

The basic design of the circuit was created from a block diagram. This diagram includes the single input from the ADT7420 Temperature Sensor. It also includes the three major components of the circuit, the RGB Color Controller, Binary to BCD Converter, and the HEX to 7 Segment Display Decoder. Finally, the outputs of the circuit are the RGB LEDs and four of the 7 Segment Displays.

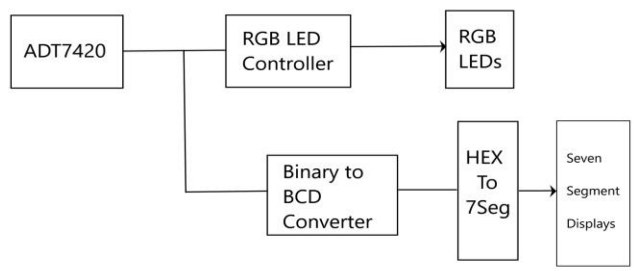

Figure 1: Initial Block Diagram

B. VHDL

The structure of the circuit created from the block diagram was then translated into a VHDL project file. The project file includes a Counter, FSM, Hex to 7 Segment Display, MUX, RGB Controller, dffe, FSM_ack, FSM_scl, Sclr Register, Shift Register, Binary to BCD Converter, ADT7420 Register, and a TopFile. In addition, the constraint file was created.

The figure shown below demonstrates how the ADT7420 Temperature Sensor interfaces to the Nexys 4 DDR board.

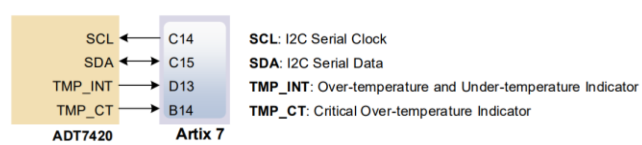

Figure 2: Temperature Sensor Interface

C. Nexys 4 DDR

We utilized the Nexys 4 DDR board to demonstrate our project. Below shows a figure labeling the parts of the board we used. Number 1: two RGB LEDs, scaled to display red, green, or blue depending on the temperature. Number 2: four 7 segment displays, to display the temperature value in degrees Celsius. Number 3: ADT7420 Temperature Sensor, used for testing a variety of different temperatures. Finally, Number 4: a single switch used to output the temperature in degrees Celsius.

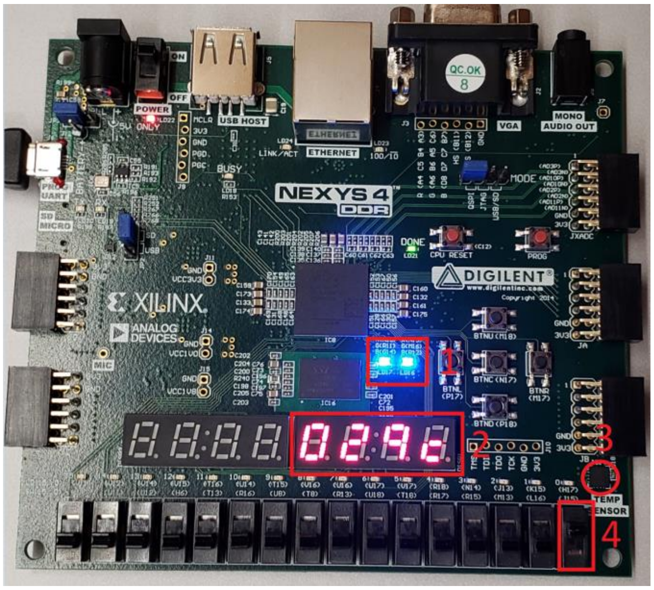

Figure 3: Nexys 4 DDR Board

D. ADT7420 Temperature Sensor

The ADT7420 Temperature Sensor is located on the bottom right corner of the Nexys 4 DDR board. This high accuracy digital temperature sensor operates as an I2C slave device. Data can be read or written via a register-based interface. The amount of data that one could read/write is one byte or two bytes per bus transaction. The resolution of this sensor can be selected by the user. We chose to implement the 16 bit resolution with an FX signed representation. We configured the appropriate ADT7420 registers and then proceeded to read two 8-bit registers. These registers included 0x00 Temp_H and 0x01 Temp_L. Together, these two 8 bits of data equalled the entire temperature.

The basic controller is depicted below. The block wr_reg_adt7420 handles the I2C communication based on address, data and write/read decision. Asserting the □□□□□ signal initiates a transaction. We read data, which appears on □□□□□. Two registers were utilized to gather the complete stream of odata. The temperature data was divided over odata_H (0x00) and odata_L (0x01), together we received a 16 bit binary temperature. A new transaction can be started on the next cycle after □□□□ = 1. [3]

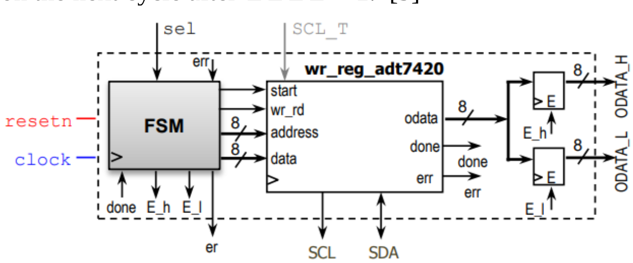

Figure 4: Temperature Sensor Controller [3]

We included a Finite State Machine that issues commands to configure one ADT7420 register (CONFIG) and then read (cyclically) from two of four 8- bit ADT7420 registers (selected by the sel input). Data is fetched on the output registers. We use the registers to read TEMP_H and TEMP_L (0x01, 0x00). [3]

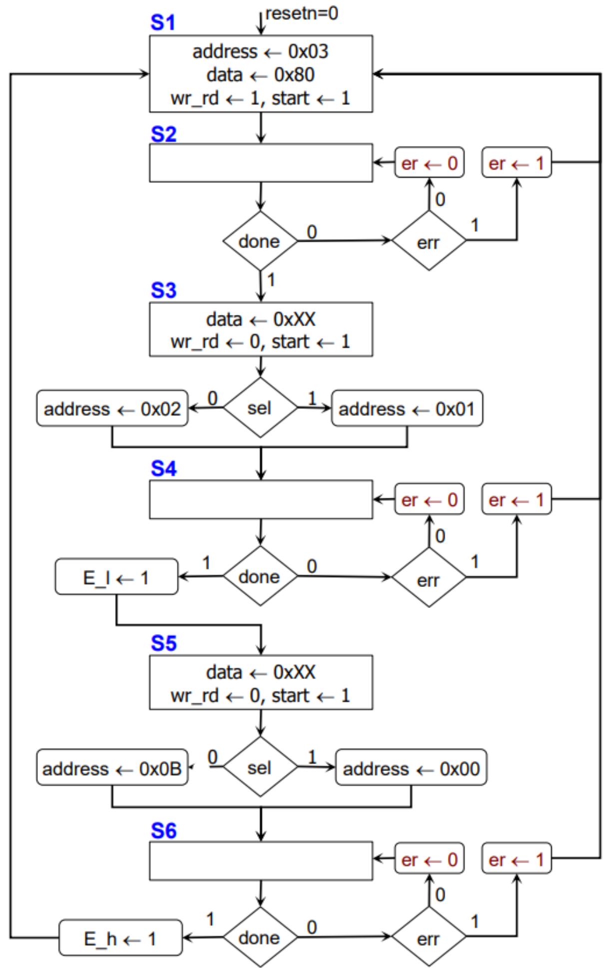

Figure 5: Finite State Machine for ADT7420 Temperature Sensor [3]

E. Binary to BCD Converter

The 16 bit temperature obtained from the two registers was required to be divided by 128 according to the ADT7420 datasheet. [2] Dividing by 128 correlates to shifting to the right by 7 bits. Instead of utilizing a shift register, we stored the 16 bit temperature into a vector. We grabbed the necessary bits to convert to BCD. The necessary bits include the 7 Least Significant Bits (LSB) of odata_H and the Most Significant Bit (MSB) of odata_L. The reason for ignoring the MSB of odata_H is because it is the sign bit. Since we were unable to get a temperature in negative degrees Celsius with our testing, we decided to ignore this bit. In our conversion, we received three digits. The amount of digits is three since the greatest number one can receive from 8 bits of data is 256. We stored each of these digits, 4 bits each, in a

16 bit vector. The last 4 bits of the vector were hard coded to the value of “C”, indicating the unit of temperature.

F. RGB LEDs

The Nexys 4 DDR board contains two RGB LEDs. We utilized these RGB LEDs to serve as an indicator of what temperature the temperature sensor was reading. The temperature value threshold was set to 50+ degrees Celsius to illuminate the LEDs to Red. Additionally, the temperature range 30 - 49 degrees Celsius was set to illuminate the color green. Finally, the temperature range of 0 to 15 degrees degrees Celsius was set to illuminate the RGB LEDs to Blue.

The RGB LED Controller utilizing the entire 16 bit temperature gathered from the odata_H and odata_L register data. We set ranges for each of the three colors of LEDs in binary. Since the data coming directly from the temperature sensor is required to be divided by 128, we multiplied the binary ranges for the LEDs by 128.

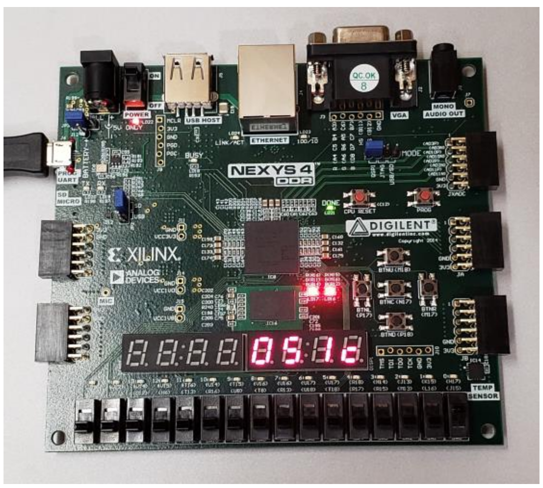

Figure 6: Example of Red LEDs illuminating for 51 Degrees Celsius

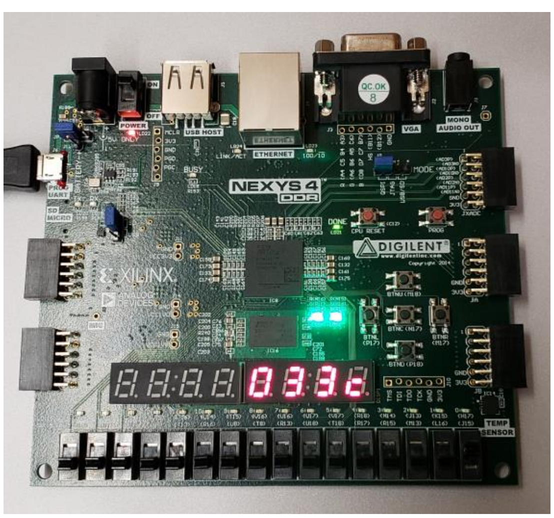

Figure 7: Example of Green LEDs illuminating for 33 Degrees Celsius

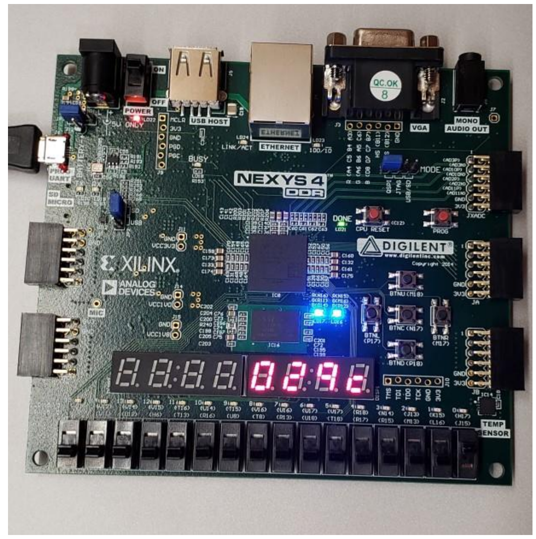

Figure 8: Example of Blue LEDs illuminating for 29 Degrees Celsius

G. Seven Segment Displays (4)

To provide a visual, we utilized four of the 7 segment displays to show the temperature of the ADT7420 Temperature Sensor in degrees Celsius. We took the vector output from the Binary to BCD converter and utilized a Hex to Seven Segment Display mapping we formed in a previous lab. The first 7 segment display was used to show the value “c”, indicating the unit of temperature for the value being shown. Below is a figure demonstrating the finite state machine we implemented for switching between the four 7 segment displays.

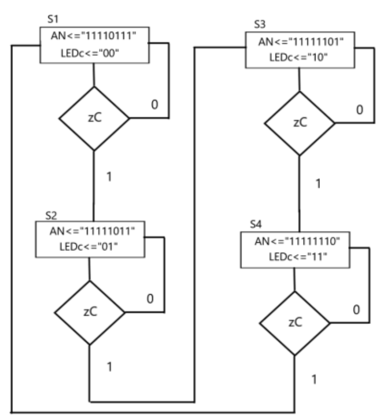

Figure 9: FSM for 7 Segment Display

III. EXPERIMENTAL SETUP

The required hardware for the project includes a Nexys 4 DDR board. On the board, the ADT7420 Temperature Sensor is used as the input for the temperature value. Additionally, the RGB LEDs and one switch on the board are used for this project. The software required is Vivado 2018.2. To test multiple variations in temperature values, a hair dryer will be utilized.

The expected results include the RGB LED's appropriately changing color according to the temperature value. It is expected that a temperature value of 122+ degrees Fahrenheit (50+ degrees Celsius) will illuminate the color red. Additionally, it is expected that a temperature value of 86 to 121 degrees Fahrenheit (30 - 49 degrees Celsius) will illuminate the color green. A temperature range of 32 to 85 degrees Fahrenheit (0 to 29 degrees Celsius) will illuminate the color blue.

To test for temperatures in the red range, 50+ degrees Celsius, the hairdryer was set at its highest temperature setting of 55 Celcius. The green range of 30 to 49 degrees Celsius was tested by using the hairdryer on its medium temperature setting. Finally the blue temperature range of 0 to 29 degrees Celcius was tested by using the hairdryer at its cooling setting.

I. Results

The results obtained from this project were as expected. The 7 Segment Display functions as intended as it shows the value of temperature in degrees Celsius. Initially, we wanted to include a switch to change the temperature output to degrees Fahrenheit. Unfortunately, the group ran out of time to complete this task. This is why there is an extra 7 Segment Display being illuminated.

Additionally, the RGB LEDs illuminate as intended. The LEDs illuminate Red, Green, or Blue for the proper range of values.

The ADT7420 Temperature Sensor gave our group the most issues. This is mainly due to the fact that no one in our group had worked with this sensor before. We initially believed the data we grabbed from the sensor was being outputted in 2's complement. However, upon digging deeper into the data sheet and consulting the professor, we discovered this data was outputted in 16 bit FX format. FX format is binary data that is signed. Once we discovered this, we were able to grab the necessary bits of data to convert to BCD and then display on the 7 Segment Displays.

Conclusions

Overall, the material covered throughout the semester was extremely beneficial in aiding the success of our project. The labs involving a Counter, FSM, Hex to 7 Segment Display, and MUX were applied to our project design. We implemented the Binary to BCD converter, RGB LED Controller, and the Temperature Sensor exclusively for this project.

We gained more knowledge in the functionality capabilities of the Nexys 4 DDR board. Additionally, we learned how to interface with the ADT7420 Temperature Sensor. Successfully, we implemented reading data from the temperature sensor, displaying it in degrees Celsius on the 7 Segment Displays, and mapping the temperature values to corresponding colors on the RGB LEDs built into the board.

Future improvements include adding a switch on the Nexys 4 DDR board to switch the unit of temperature being displayed to degrees Fahrenheit so it is more user friendly in America. Additional improvements included mapping the RGB LED values to gradually fade into another color.

References

[1] Apr-2016. [Online]. Available: "https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-4-ddr/nexys4ddr_rm.pdf." [Accessed: 08-Dec-2018].

[2] Oct-2018. [Online]. Available: http://www.secs.oakland.edu/~llamocca/Tutorials/VHDLFPGA/Vivado/Unit_7/Tempsensor_ADT7420.pdf. [Accessed: 08-Dec-2018].

[3] Oct-2018. [Online]. Available: http://www.secs.oakland.edu/~llamocca/Courses/W18_ECE3710/Notes - Unit 3.pdf. [Accessed: 08-Dec-2018].