Electronic Mood Ring

Matthew Redoute, Tyler Glass, Trevor Shankin, Zachary Howard
Electrical and Computer Engineering Department
School of Engineering and Computer Science
emails: mredoute@oakland.edu, tylerglass@oakland.edu, shankin@oakland.edu, zkhoward@oakland.edu

ABSTRACT

The purpose of this project is to create an Electronic Mood Ring that contains a temperature sensor and an accelerometer. The temperature and accelerometer readings will be displayed on the seven segment display of the Nexus DDR4 board (50T). VHDL code will be used to program this Nexus board that will demonstrate the finished project.

I . INTRODUCTION

The motivation for this project is to provide consumers access to a product that is sleek and fun to use. Learn your "mood" by using this tool and watch as the colors change with activity and motion. The idea could be replicated in more forms than just a circuit board with sensors embedded on the inside of a ring. Smartphone applications, actual mood rings, and other devices can implement the same core ideas. Electronic mood rings are fun to use and are a neat display of what electronics are capable of.

Figure 1: Electronic Mood Ring

The project will consist of VHDL code for three finite state machines, seven registers, two counters, PWM, a binary to BCD converter, an RGB controller with an ADXL362 accelerometer, an ADT7420 Temperature Sensor, and a seven segment display serializer on the Nexys A7 FPGA Trainer Board (50T).

II . METHODOLOGY

It was decided that the sensors required to produce an Electric Mood ring would be the temperature sensor and the accelerometer on the Nexys Board. These input sensors are utilized to determine the color represented on the RGB LED, and have the sensors' values outputted onto the seven segment display. The Nexus board is a programmable PCB that has all the input sensors and output displays installed, so all that is needed to be finished for the project is creating a working code and connecting all the components in the top file. An external component in the form of a buzzer was added to the project for audio signals. An optional external LCD screen could be used to provide more readable information provided by the sensors, as seen on figure 5 located in the appendix.

The logic is a state machine taking the inputs from the temperature and speed sensor, comparing them to predetermined values, then outputting high and low values to the RGB in order to change the color.

The current setup is the temperature sensor outputting the values to the 7-Segment display in Celsius. The speed sensor also outputs it's values to the 7-Segment displays on different LEDS.

The output to the RGB LED is digital between the speed and temperature sensor with a switch used to determine the outputting value. The values are always present on the display, but the color controller is toggled by a switch.

PWM was looked into in order to have both sensors controlling the LED simultaneously. Where the Temperature sensor would control the RGB color selection, while the speed sensor would be used to change the brightness of the LED. However PWM was ultimately assigned to a Buzzer that would change volume when certain values on the speed sensor were read.

The current datapaths for the input sensors are located below in the *APPENDIX* section *3A-3B*.

III. EXPERIMENTAL SETUP

1. Inputs

a. Switches

A single switch was utilized on the Nexys A7 50T board to control which component, either temperature or accelerometer sensor, controlled the RGB LED A future expansion of the project is to have another switch made to change the temperature sensor values on the seven segment display to Celisus or Fahrenheit.

b. Accelerometer

An accelerometer is used to detect changes in motion. This component will help the device detect not only this but the orientation of the device. Gravitation is a constant acceleration that will be picked up and will help the device know how it is oriented to the user. This component will interface with the board via the SPI interface. There are 4 wires used including SCLK, MOSI, MISO, and nCS. To properly implement this device an FSM was created to help step through repeatable instructions for carrying out its tasks. A controller block interface was also used to handle the SPI communication and connect the onboard sensor to our circuit. Through trial and error the first thing was to determine where the 3 dimensional axes correspond to which directions. It was found that only the z acceleration was constant which implied this was measuring the acceleration due to gravity. This would help locate which direction is "down" and would work to indicate to the board how it is oriented in terms of tilt and direction.

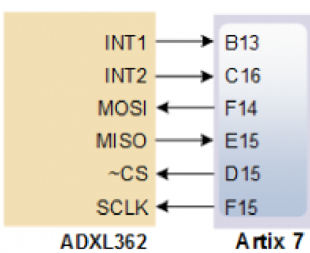


Figure 2: Communication between the accelerometer and the Nexus board

c. Temperature Sensor

The temperature sensor is another major component which will be responsible for reading temperatures in both celsius and fahrenheit depending on user preference. The sensor uses the I^2C interface which has 2 wired synchronous serial interfaces: the SCL and the SDA. The system uses 1 master and many slave devices. These slaves are identified by each obtaining a unique slave address. The temperature sensor acts as the slave device. The project files for the temperature sensor utilize the 16-bit resolution mode in comparison to the 13-bit which is an important distinction to make. The temperature sensor has a range of values it can detect ranging from -40℃ to 150℃ which is a very reasonable range for the intended use of this ring. An FSM was also utilized to deliver repeatable instructions. A controller block interface was also used to handle communication between the sensor and board. The communication involves ACK and NACKs to acknowledge whether or not the data was read or sent by the slave or master.

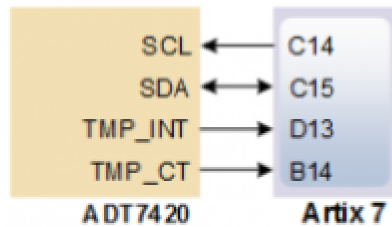


Figure 3: Communication between the temperature sensor and the Nexus board

2. Outputs

a. Seven Segment Display

The seven segment display is one of the key outputs for the operation of this device. The seven segment display will be used to display temperature and the z-axis readings. This data was first fed to a binary coded decimal circuit block so that it could be properly implemented into the display. A serializer was used to provide a means of showing different values. 4 displays are turned on and off to achieve this effect.

For the temperature sensor, AN01 through AN01 were activated and for the temperature, while AN06 and AN07 were used to map the user's location on the z-axis of the accelerometer.

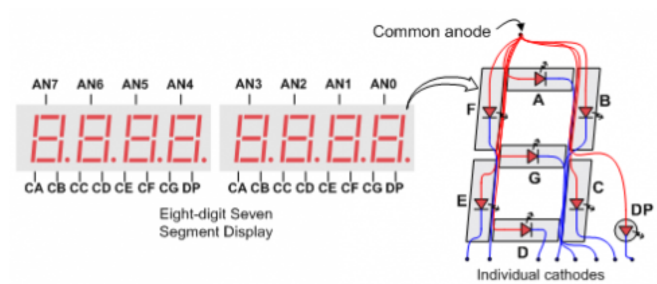


Figure 4: Diagram of the Seven Segment Display

b. RGB LED

The onboard RGB LED was one of the outputs used to express the state of the inputs: *zdata* and *tempdata* coming from the accelerometer and temperature sensors. For example, Table 1 visually shows the colors that are outputting from these input values. The output colors can either be green, purple, blue, cyan, yellow for the accelerometer sensor and blue, yellow, and red for the temperature sensor.

Electronic Mood Ring Input Data	
zdata	tempdata
> x"BF" *and* <= x"CE"	<=30℃
> x"CE" *and* <= x"E4	
> x"E4" *and* <= x"FF"	> 30℃ and <= 40℃
> x"00" *and* <= x"15"	
> x"15" *and* <= x"30"	
> x"30"	> 40℃

Table 1: Input Data

c. Piezo Buzzer

Another output implemented was the use of a piezo buzzer. The setup of this device on the board can be seen in section 2 of the appendix section. The current draw from the I/O pins was too low to drive high sounds from the buzzer, therefore a MOSFET was needed to allow an external battery of 9V to power the speaker. For example, if enough current/voltage comes out of the connected pin, a gate opens within the MOSFET enabling the negative terminal of the battery to connect to ground. Beeping sounds would only occur if the switch was in a HIGH state meaning that the accelerometer was in control of the RGB. If the RGB glowed yellow then a soft beeping sound would be outputted while if the RGB glowed red then a higher, more intense volume sound would occur. The changing volume was made possible through a PWM component in the code files in which different duty cycles permitted the changing volumes.

d. Overall Setup of the Board.

Figure 5 reveals all of the main devices that were used to implement this project while section 1 from the Appendix section shows the real setup. A micro USB supplied power to the board as well as for uploading the program within Vivado. The CPU reset was the only button used on the board, which was helpful to reset the values to zero if any complications arisen with the sensors. From previous mentioned sections, an input switch controlled whether the RGB was controlled through the onboard temperature and accelerometer peripherals. The values from these sensors were displayed on the seven segment displays. The first two displays (on the right) were reserved for the temperature value in degree Celsius while the last two displays (on the left) were reserved for the z-axis data value from the accelerometer. Finally, the other outputs employed were the RGB LED and pin 2 from the JA PMOD port.

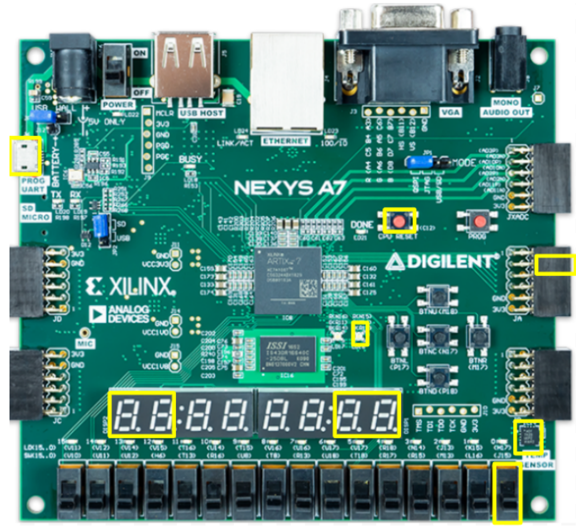


Figure 5: Components used on the FPGA board

IV. RESULTS

Building upon the ideas and concepts learned in class along with the practice in the lab, the Electronic Mood Ring was able to be created.

Code was created where necessary in order to bring the Mood Ring to life, and code that was borrowed was able to be understood and implemented into the final design. The components being the temperature sensor, speed sensor, and 7-segment display were able to be understood, initiated, and manipulated to fit the design of the project. PWM was researched and incorporated into a buzzer to better see the potential uses of the Mood Ring.

The group was able to accomplish what we had designed, and were able to figure out what additional features to include if the design were to be improved, as well as implement it.

A video of the project functioning is available in the following link: https://www.youtube.com/watch?v=g26nirCt5V0 .

V. CHALLENGES

One challenge was implementing a toggle for the temperature sensor, which would allow the displayed value to be either in Fahrenheit or Celsius. However integrating the conversion into the code presented issues, which would prevent the code from functioning properly. As this unit toggle would only be a refined feature, it was decided to remove it from the project.

The major challenge was determining how to implement PWM control into the project. The reason PWM was difficult to implement was due to the clock cycle. Due to the FPGA board not having a dedicated clock for this type of operation, a digital clock cycle was created through loop and case statements, as well as the time counter available on the board. After establishing a basic clock cycle, PWM would be used to send different audio intensities through a buzzer, only on specific outputs.

Lastly, working together remotely was difficult. It would have been much easier and more efficient to work together in person, especially when construction the design on the Nexus board. The group overcame this challenge and successfully created a working design.

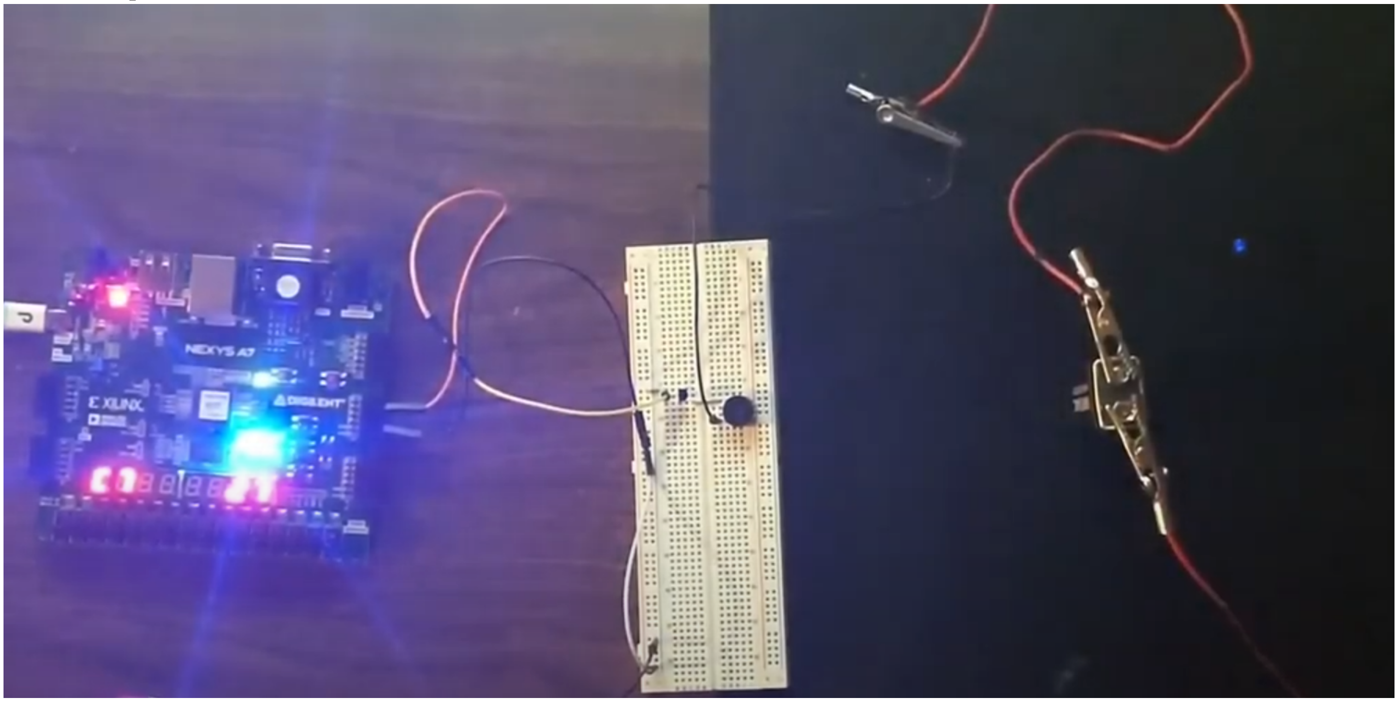
VI. CONCLUSION

By tracking the progress of the Electronic Mood Ring, it became easy to establish what the group wanted to include in the project, and what was feasible within the available time frame.

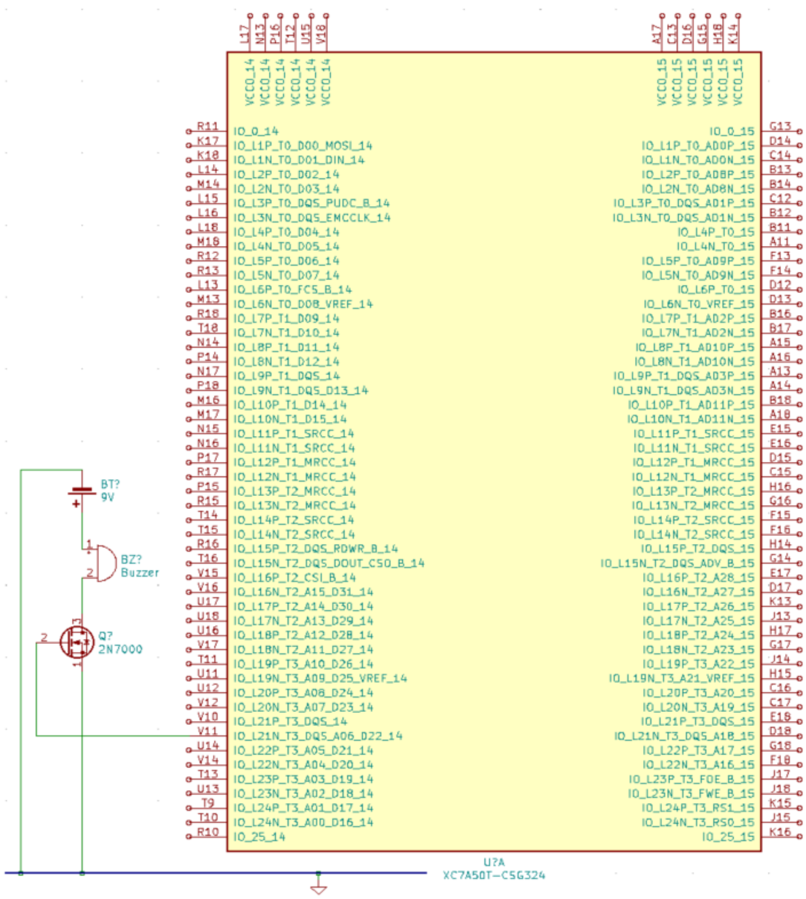
By incorporating the onboard temperature and speed sensor, internal 7-segment display, and the external buzzer; the Electronic Mood Ring was able to be constructed physically, and assembled together electronically through the use of VHDL.

APPENDIX

1. Real Setup

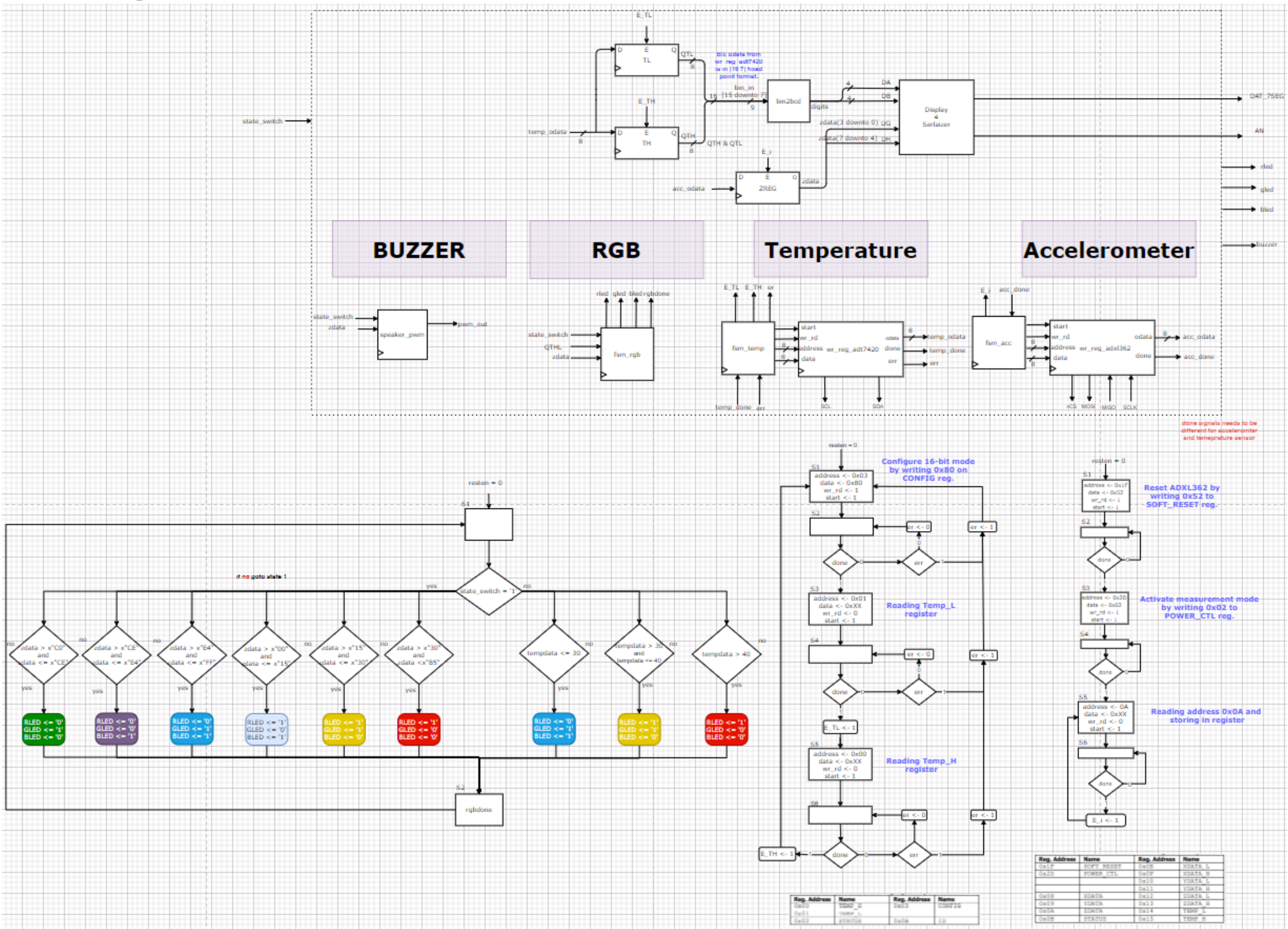


2. KiCad Schematic of Buzzer

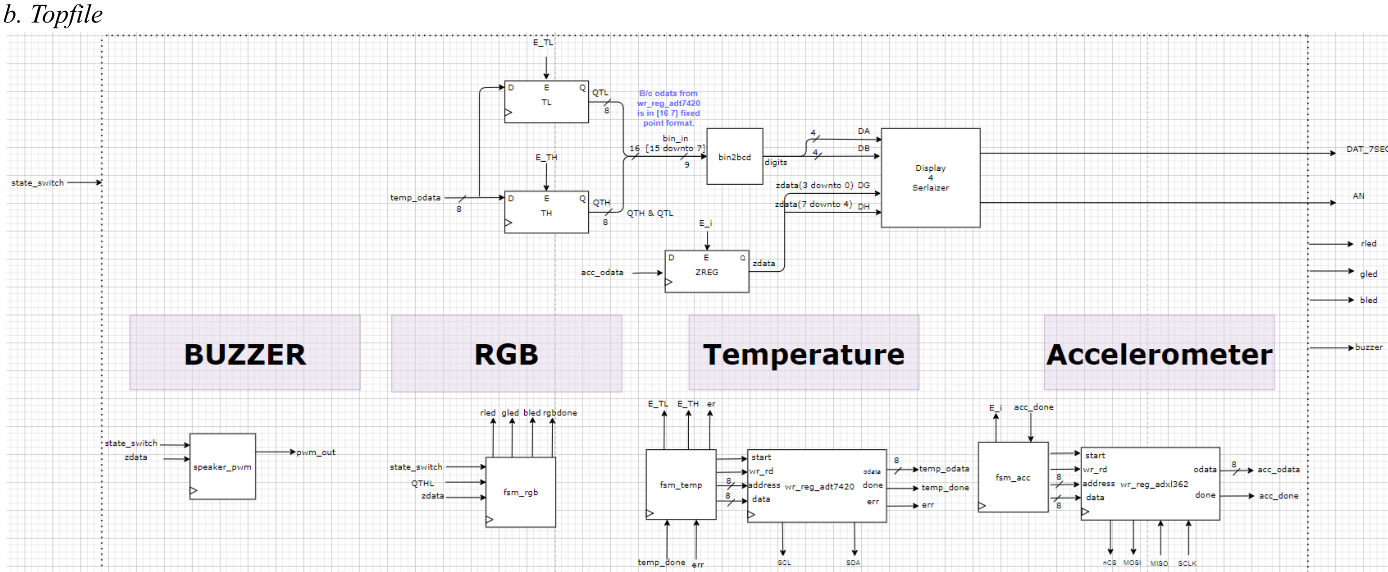


3. Datapath

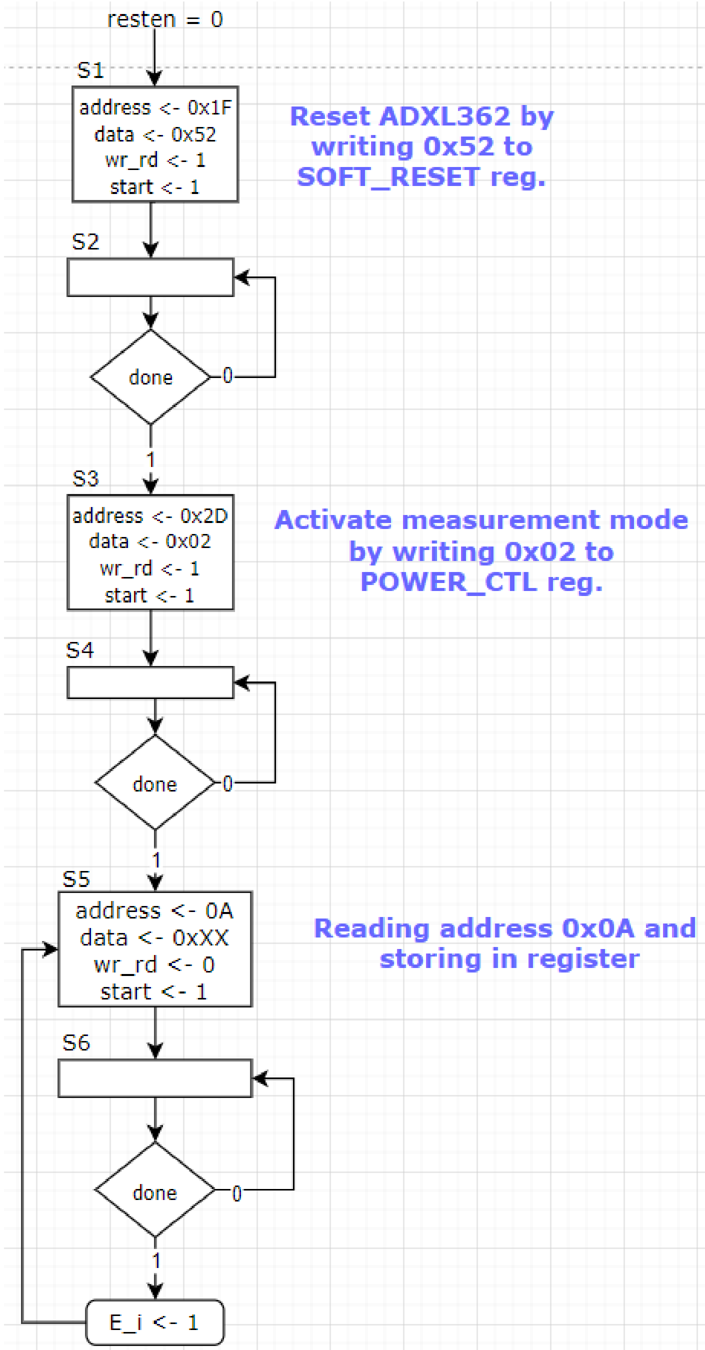
a. Entire Datapath



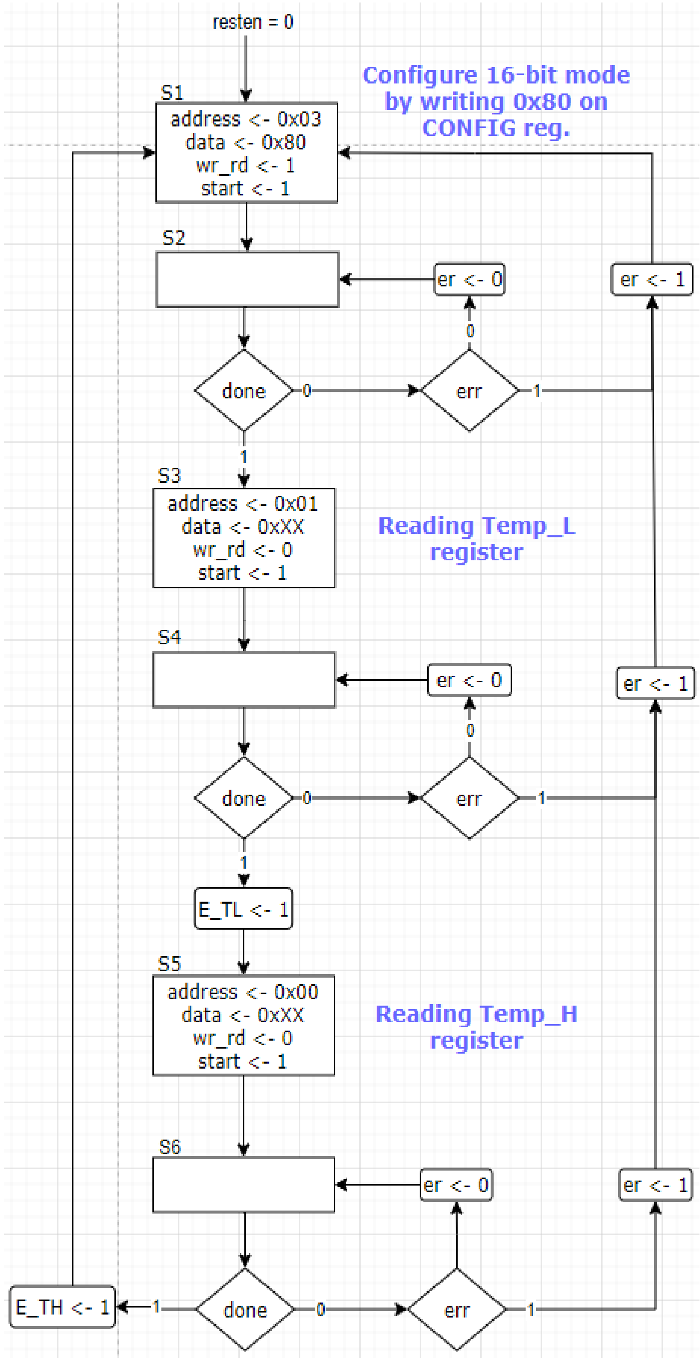
b. Topfile



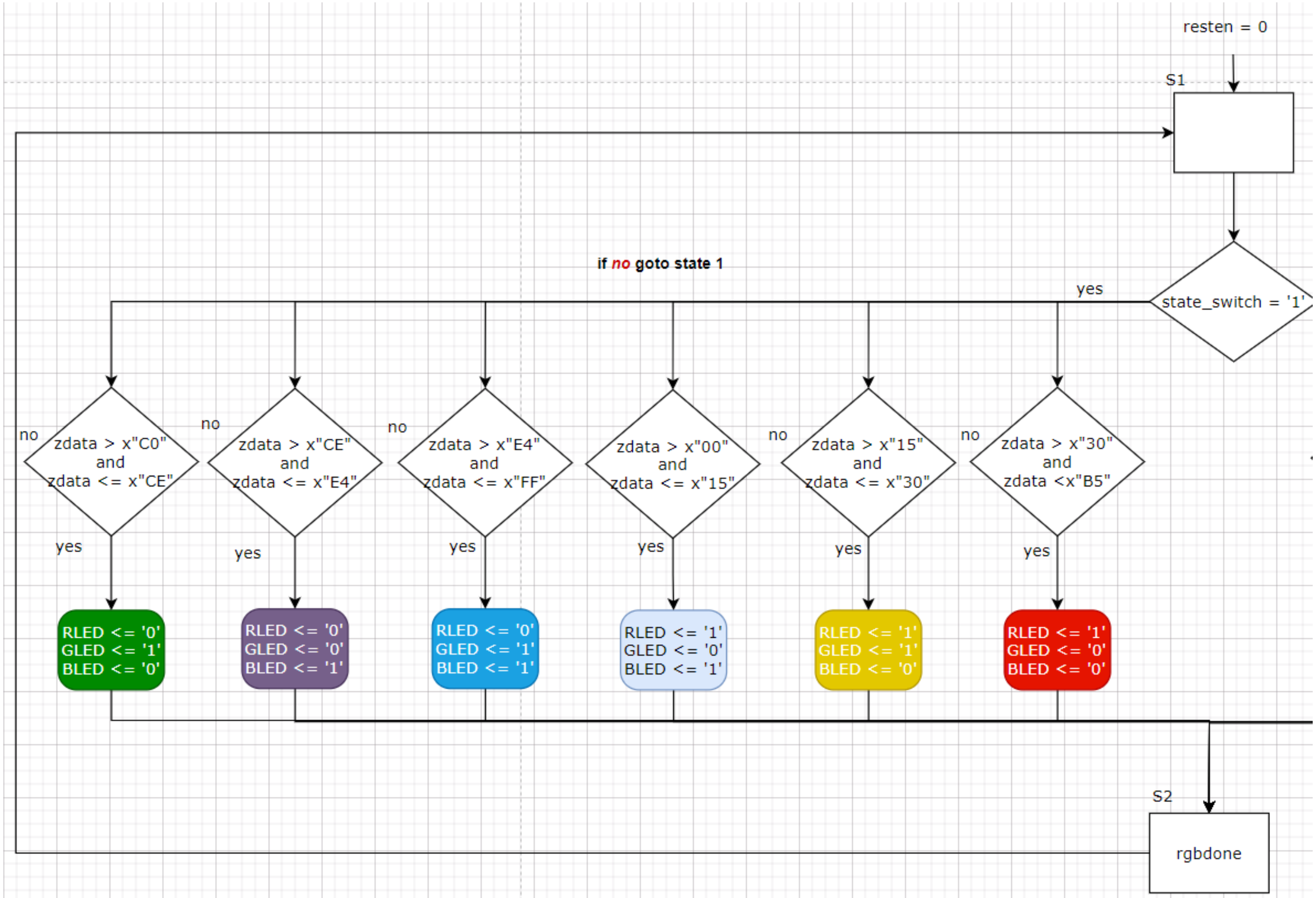
c. FSM Accelerometer

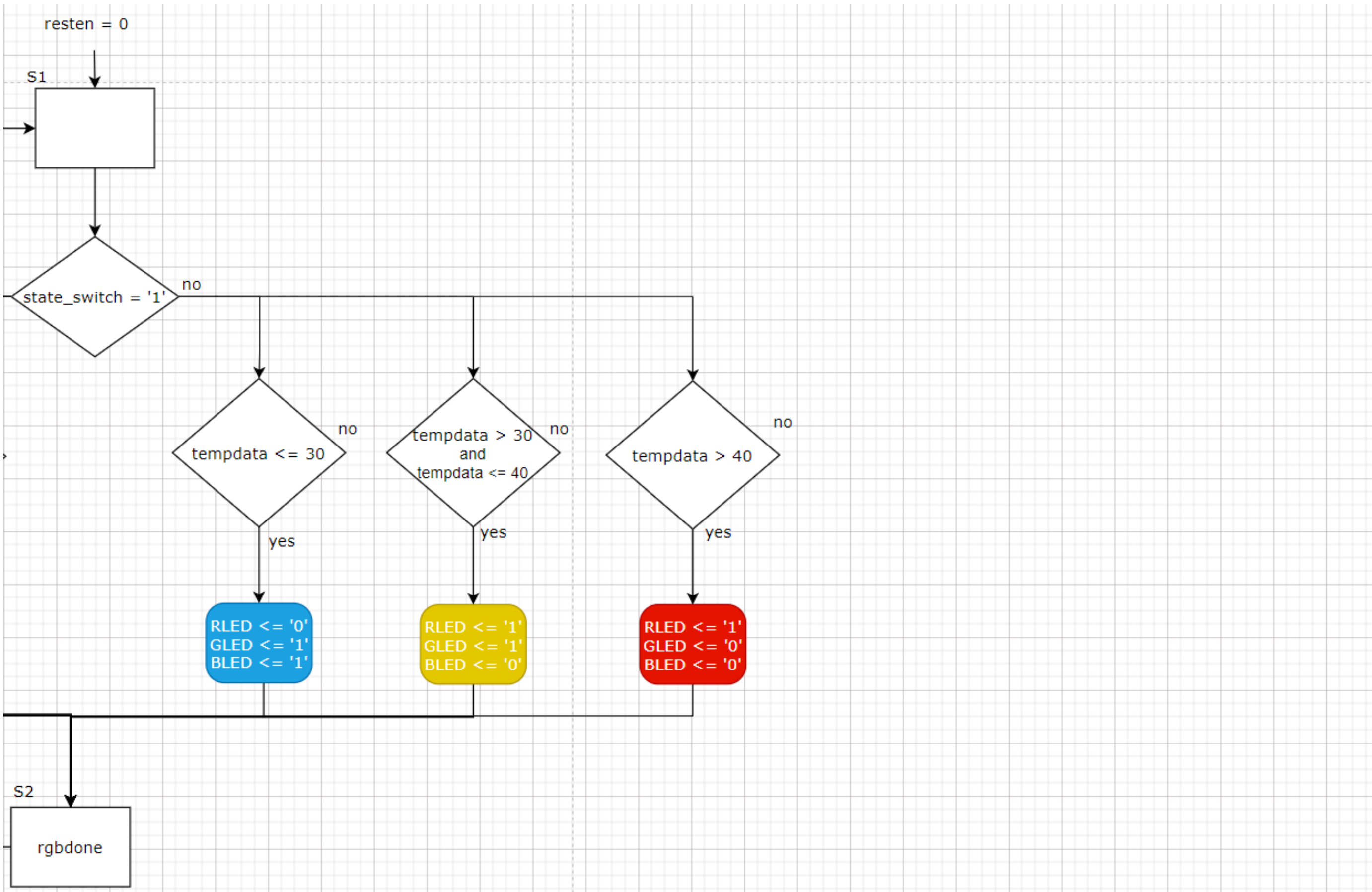


d. FSM Temperature



e. FSM RGB


resten = 0
S1
if no goto state 1
yes
state_switch = '1'
no
zdata > x"C0" and zdata <= x"CE"
no
zdata > x"CE" and zdata <= x"E4"
no
zdata > x"E4" and zdata <= x"FF"
zdata > x"00" and zdata <= x"15"
no
zdata > x"15" and zdata <= x"30"
no
zdata > x"30" and zdata <x"B5"
yes
RLED <= '0' GLED <= '1' BLED <= '0'
RLED <= '0' GLED <= '0' BLED <= '1'
RLED <= '0' GLED <= '1' BLED <= '1'
RLED <= '1' GLED <= '0' BLED <= '1'
RLED <= '1' GLED <= '1' BLED <= '0'
RLED <= '1' GLED <= '0' BLED <= '0'
S2
rgbdone


resten = 0
S1
state_switch = '1'
no
tempdata <= 30
no
tempdata > 30
and
tempdata <= 40
no
tempdata > 40
no
yes
yes
yes
RLED <= '0'
GLED <= '1'
BLED <= '1'
RLED <= '1'
GLED <= '1'
BLED <= '0'
RLED <= '1'
GLED <= '0'
BLED <= '0'
S2
rgbdone