Table of Contents

[INTRODUCTION: 2](#_Toc156331826)

[METHODOLOGY: 2](#_Toc156331827)

[RESOURCE UTILIZATION: 4](#_Toc156331828)

[APPLICATIONS: 4](#_Toc156331829)

[VERILOG SOURCE CODE: 5](#_Toc156331830)

[RTL SCHEMATIC: 9](#_Toc156331831)

[SIMULATION RESULTS: 10](#_Toc156331832)

[RESULT ANALYSIS: 10](#_Toc156331833)

[REFERENCES 11](#_Toc156331834)

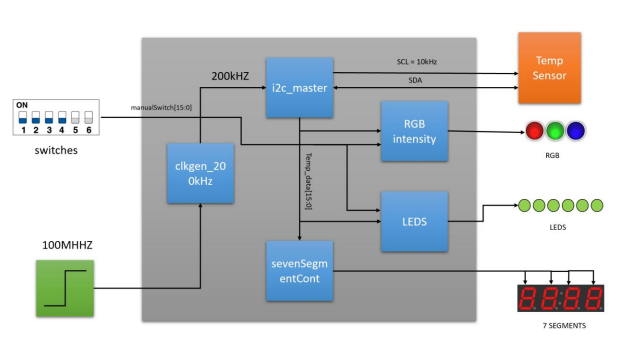
# INTRODUCTION:

The scope of the project is utilizing the ADT7420 Temperature Sensor built into the Nexys 7 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 LEDs on the Nexys 7 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 the 7 Segment Displays built into the board.

# METHODOLOGY:

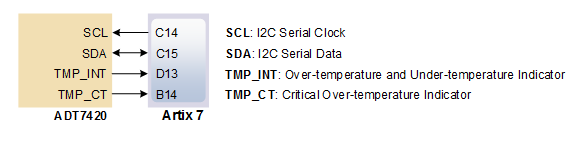
1. *Block Diagram*

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 two major components of the circuit, the RGB intensity controller and the i2c\_master. Finally, the outputs of the circuit are the RGB LEDs and the 7 Segment Displays.



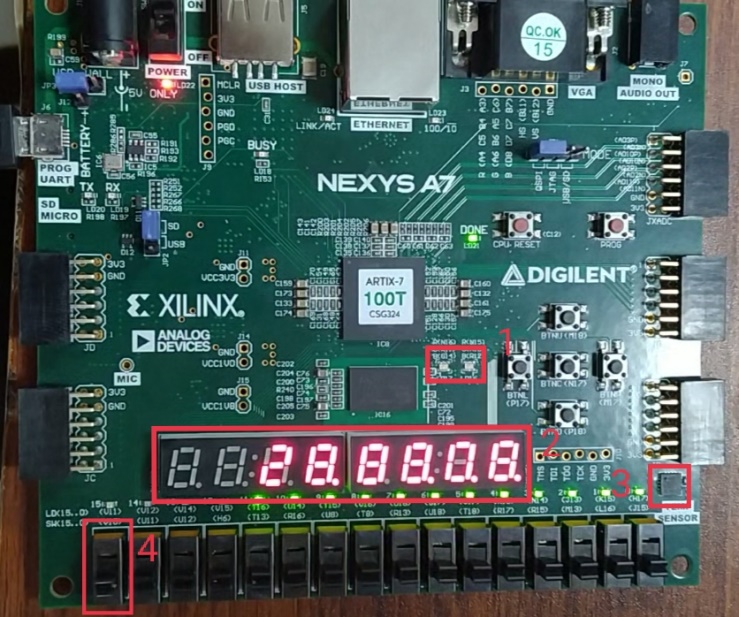
1. *VHDL*

The structure of the circuit created from the block diagram was then translated into a VHDL project file. The project file includes a i2c\_master, clkgen\_200kHz, seg\_Display, tri\_color, pwm and a TopFile. In addition, the constraint file was created. The figure shown below demonstrates how the ADT7420 Temperature Sensor interfaces to the Nexys 7 DDR board.

******

1. *Nexys 7 DDR*

We utilized the Nexys 7 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: 7 segment displays, to display the temperature value. Number 3: ADT7420 Temperature Sensor, used for testing a variety of different temperatures. Finally, Number 4: a single switch used to check output manually.

******

1. *ADT7420 Temperature Sensor*

The ADT7420 Temperature Sensor is located on the bottom right corner of the Nexys 7 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.

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

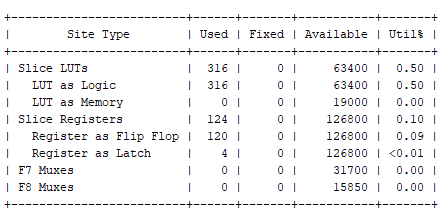
1. *RGB LEDs*

The Nexys 7 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 26+ degrees Celsius to illuminate the LEDs to Red. Additionally, the temperature range 10 - 26 degrees Celsius was set to illuminate the color green. Finally, the temperature range of 10 to -30 degrees Celsius was set to illuminate the RGB LEDs to Blue.

1. *Seven Segment Displays*

To provide a visual, we utilized all of the 7 segment displays to show the temperature of the ADT7420 Temperature Sensor in degrees Celsius

# RESOURCE UTILIZATION:



# APPLICATIONS:

This project could be implemented for various applications such as a pool temperature sensing light, Motors, Home appliances, Industrial control and testing, environmental monitoring and HVAC, or for food transportation and storage.

# VERILOG SOURCE CODE:

module tri\_color(

input clk\_100MHz, // Nexys A7 clock

input [15:0] temp\_data, // 8 bits of temperature data

output reg R, //for Tricolor LED

output reg G, //for Tricolor LED

output reg B //for Tricolor LED

);

//8 bit pwm counter

wire [7:0] pwm\_count;

//3 bit RGB\_Color

reg [2:0] RGB\_Color;

//reg [12:0] temperature;

reg signed [12:0] temperature;

// Instantiate PWM module

pwm p1(clk\_100MHz,pwm\_count);

always @(posedge clk\_100MHz)

begin

if (temp\_data[15]==1) //For negative temperature

begin

temperature<=(((temp\_data[15:3])-8192)/16);

end

else //For positive temperature

begin

temperature<=(temp\_data[15:3])/16;

end

//For Color Coding and variation in Intensity

if(-13'sd30<=temperature && temperature<=13'sd10) //For range of temperature [-30,10] Cold temperature with Blue color

begin

RGB\_Color <= 3'b001; // blue

if(-13'sd1<temperature && temperature<=13'sd10) //Low intensity for temperature(-1,10]

begin

if (pwm\_count<10) //10% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else if (-13'sd20<temperature && temperature<=-13'sd1) //Medium intensity for temperature(-20,-1]

begin

if (pwm\_count< 50) //50% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else if(-13'sd30<=temperature && temperature<=-13'sd20) //High intensity for temperature [-30,-20]

begin

if (pwm\_count<90) //90% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else

begin

end

end

else if(10<temperature && temperature<=26) //For range of temperature (10,26] Normal temperature with Green color

begin

RGB\_Color <= 3'b010; // Green

if(10<temperature && temperature<=15) //Low intensity for temperature (10,15]

begin

if (pwm\_count<10) //10% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else if (15<temperature && temperature<=20) //Medium intensity for temperature (15,20]

begin

if (pwm\_count<50) //50% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else if(20<temperature && temperature<=26)//High intensity for temperature (20,26]

begin

if (pwm\_count<90) //90% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else

begin

end

end

else if(26<temperature && temperature<=55) //For range of temperature (26,55] Hot temperature with Red color

begin

RGB\_Color <= 3'b100; // Red

if(26<temperature && temperature<=28) //Low intensity for temperature (26,28]

begin

if (pwm\_count<10) //10% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else if (28<temperature && temperature<=46) //Medium intensity for temperature (28,46]

begin

if (pwm\_count<50) //50% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else if (46<temperature && temperature<=55) //High intensity for temperature (46,55]

begin

if (pwm\_count<90)//90% duty cycle

begin

R<= RGB\_Color[2];

G<= RGB\_Color[1];

B<= RGB\_Color[0];

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

else

begin

end

end

else

begin

R<=0;

G<=0;

B<=0;

end

end

endmodule

module pwm(

input clk\_100MHz,

output reg [7:0] pwm\_count

);

reg [7:0] pwm\_count=0;

always @(posedge clk\_100MHz )

begin

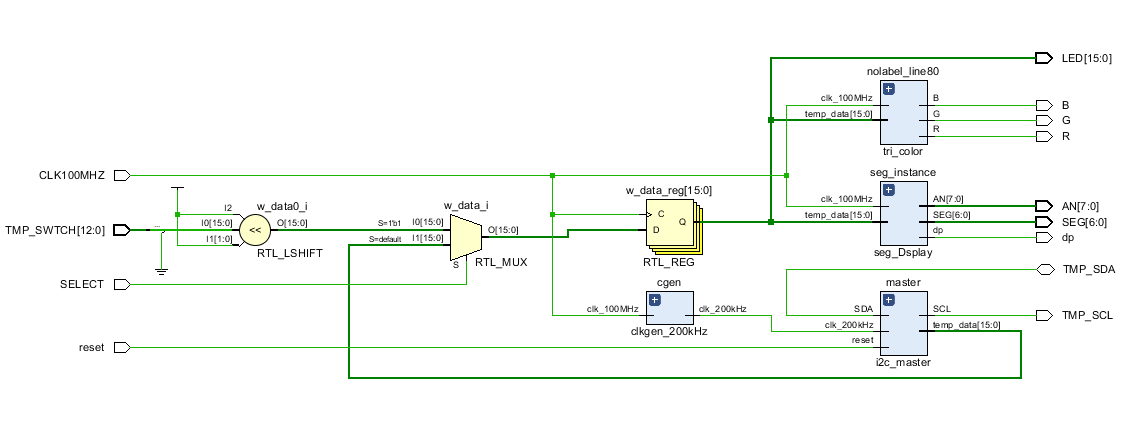
if (pwm\_count < 100) pwm\_count<=pwm\_count+1; //count until 100

else pwm\_count <=0; //reset counter

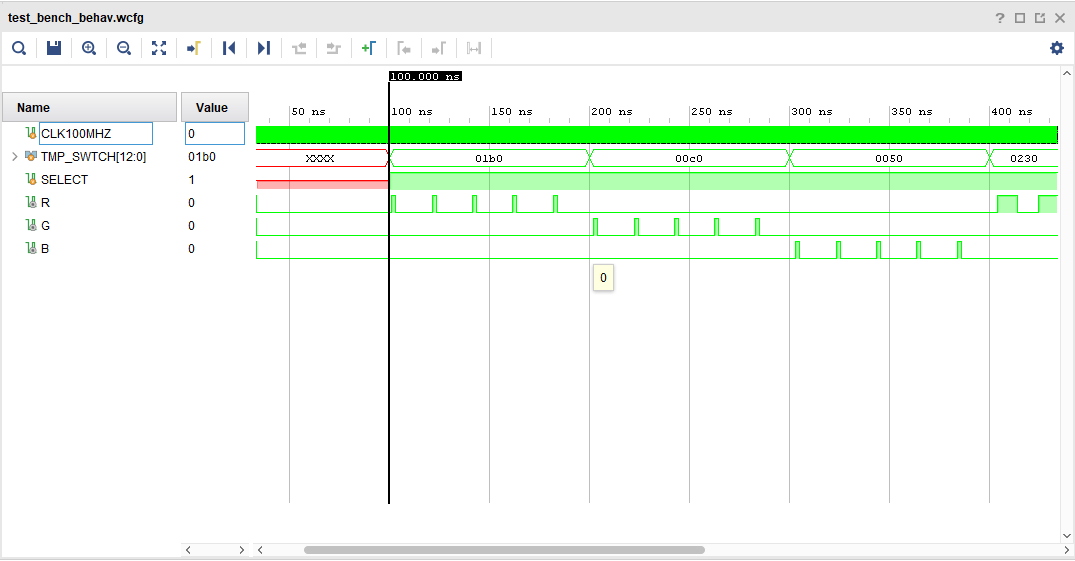
end

endmodule

# RTL SCHEMATIC:



# SIMULATION RESULTS:



# RESULT ANALYSIS:

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

# REFERENCES

1. Jan-2024. [Online]. Available

<https://digilent.com/reference/_media/reference/programmable-logic/nexys-a7/nexys-a7_rm.pdf>

1. Jan-2024. [Online]. Available

<https://www.analog.com/media/en/technical-documentation/data-sheets/adt7420.pdf>