ECE 212 – Digital Circuits II

Lab3 – Temperatur Display

Otieno Maurice (Scribe)

Alex Villalba

1. Introduction:

The goal of this lab was to interface the NEXYSA7 100T FPGA with modules to display the temperature inside the ECE 419 digital circuits lab. The NEXYSA7 100T FPGA is equipped with the Analog Devices ADT7420 temperature sensor chip connected to the FPGA by an I^2C two-wire interface. The I^2C interface is controlled by an intellectual property coded in VHDL named TempSensorCtl given by Diligent. In this lab, we developed a module named tdisplay, that takes in the 13-bit fixed point two's complement representation of the temperature and converts it into a form that can be displayed on the seven-segment display on the FPGA.

Specifically, we wanted our seven-segment display to display the temperature in both degrees Celsius and Fahrenheit within a given range of values i.e., -256 to 255 degrees Celsius. Values of the magnitude 60 and above are impractical for room temperature so the only way to know if our temperature sensor would work at those temperatures is by simulation. Consequently, we wrote testbenches for each and every new module created and checked the given results with the expected results. Once we were convinced the design would work, we implemented the design and realized it on the FPGA board.

2. Design:

Design specifications:

1. The design shall interface to the ADT7420 temperature sensor on the Nexy's board using a VHDL IP block provided by Diligent.
2. The design shall convert by the ADT7420 temperature sensor to display the temperature on the in decimal degrees Celsius to an accuracy of one tenth of a degree over an input range of -256 to +255.9375 degrees Celsius.
3. The design shall convert by the ADT7420 temperature sensor to display the temperature in decimal degrees Fahrenheit to an accuracy of one tenth of a degree over an input range of -256 to +255.9375 degrees Celsius.
4. The correct operation of the converter subsystem shall be verified in simulation over the specified input range prior to integration into the larger design.
5. The design shall display the temperature in either degrees Celsius or Fahrenheit over the range of -256 to +255.9375 degrees in a format that includes a dash when the temperature is negative and the magnitude of the temperature to the accuracy of one tenth of a degree.
6. Display of either Celsius or Fahrenheit shall be selected using slide switch SW0 on the Nexy's board.
7. The seven-segment display shall show the temperature as a right-justified number with the letter C or F in the rightmost digit, e.g., 0.0C, 32.0F, 100.0C, 212.0F, -17.8C, 0.0F, 255.9C, 492.7F, -256.0C, -428.8F, etc. Leading zeros shall be blanked as shown.

8. The design shall allow hardware testing of the full range of temperature values by substituting switches SW15-SW3 for the 13-bit value from the temperature sensor. SW1 shall be used to select whether the design displays the temperature sensor (OFF) position, or the switches (ON position).

Design implementation:

This design utilized a top module named lab03_top. This module contains instances of the three submodules sevenseg_display, tdisplay and TempSensorCtl.

Inside lab03_top we select the temperature we want to display by a two-input multiplexer controlled by switch 1. The inputs into this multiplexer are the temperature values from the temperature sensor and user input from the switches. When switch1 is off, the mux selects input from the sensor and when on it selects input from the user.

In module tdisplay, we have four modules each with different functions. The very first module, tconvert, takes in the 13-bit two's complement fixed point value of the temperature and depending on the state of switch0 produces an 18-bit temperature value in either degrees Celsius or degrees Fahrenheit. The temperature from the sensor is in degrees Celsius and has both a fractional part and an integer part. We need to account for both of these parts. Since the double dabble algorithm discussed in DIG 1 won't work for fractional parts, we multiply the temperature by ten to get an 18-bit value. This will not affect the temperature reading because the temperature will be displayed in tenths of degrees Celsius. To display in Fahrenheit, we follow the mathematical formula,

T =9/5×T+32. This is similar to multiplying the temperature by 18 and then adding the value 320 to the integer part. The choice to give out the result in Celsius is done by a two-input multiplexer whose select signal is the switch 0. When zero is driven high, the multiplexer should select Fahrenheit and when low should select Celsius.

Remember, the FPGA has no way of directly displaying a negative number. In lab1, we designed the seven-segment controller to display a minus sign to represent negative. We used this minus sign in this lab. The next step is to generate a signal to display the minus sign when the temperature value is negative. To do this we look at the most significant bit of the temperature. If it is a one, we invert the value of the temperature and then produce a high signal named sign. If it is a zero, WE PRESERVE the value of the temperature and the sig signal is driven low. This is done in a module named Conv_sgnmag. The temperature output of this module is therefore a seventeen-bit value.

Next step is to round of the temperature so that it is accurate to a tenth of a degree. Inside the module round, we check if the fourth bit is high. If it is, we truncate out the last four bits and add one to the resulting thirteen-bit number If the fourth bit is low, we just truncate out the last four bits. Finally, we feed this value into the double dabble algorithm to produce its binary coded decimal equivalent for display.

The value from display along with its sign bit is then fed into the seven-segment controller to determine what to display.

The top-level diagram for this design is given below.

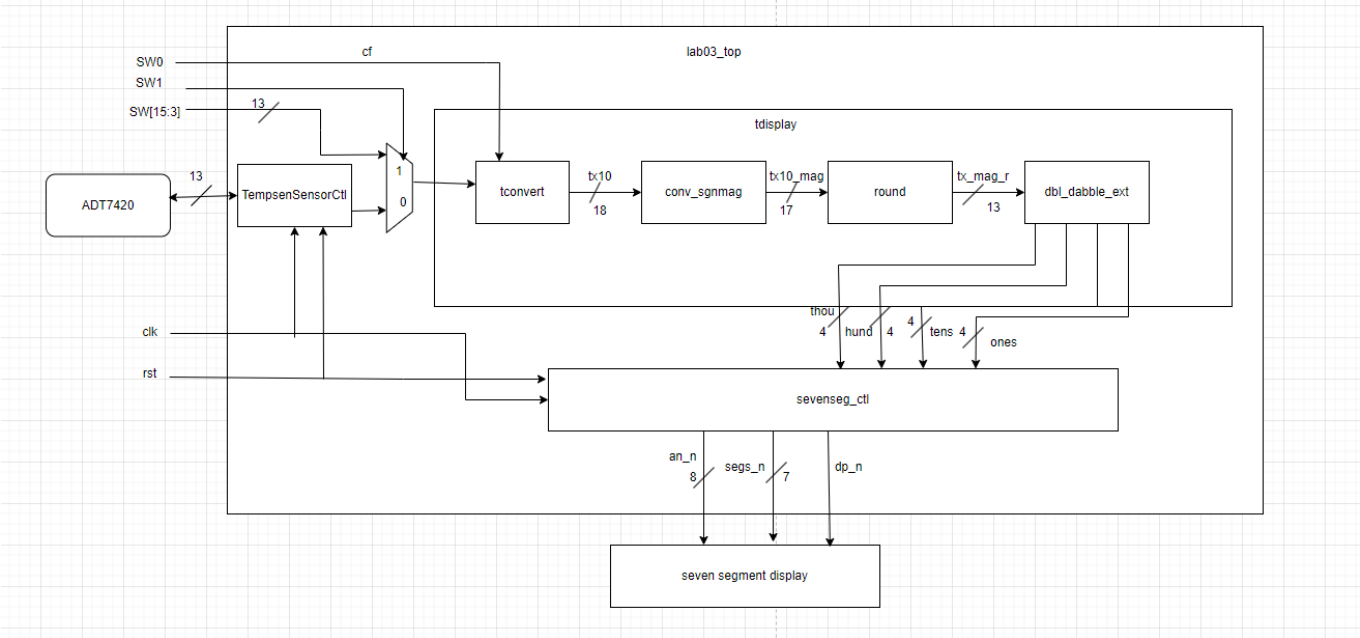

Figure 1top level diagram showing the temperature display module

We described the behavior of this modules in system Verilog and used vivado's CAD tools to simulate the design. The results are given in the results section below.

Design realization:

To realize this design on our FPGA, we specified the constraints in an xdc file and generated write bitstream. We then programmed our device and observed the behavior of the seven-segment display for given test-values by the user and for values from the temperature sensor.

3. Results

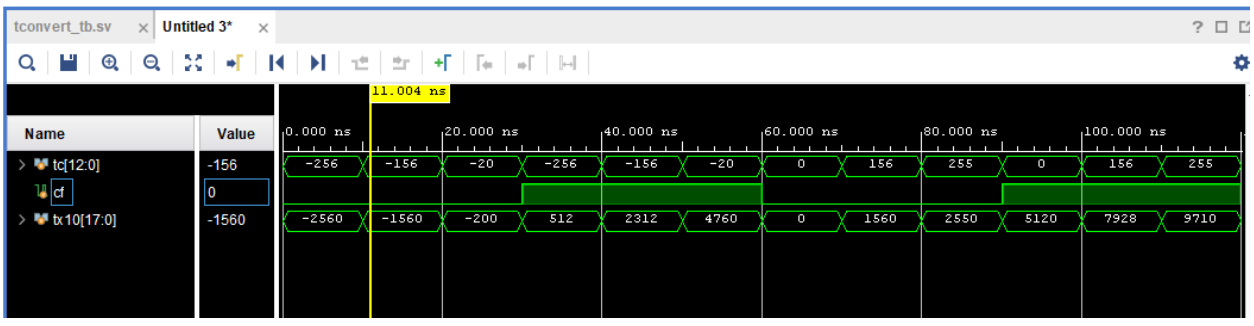

Figure 2tconvert simulation scope screenshot

The tconvert module was simulated for specific values within the required range.

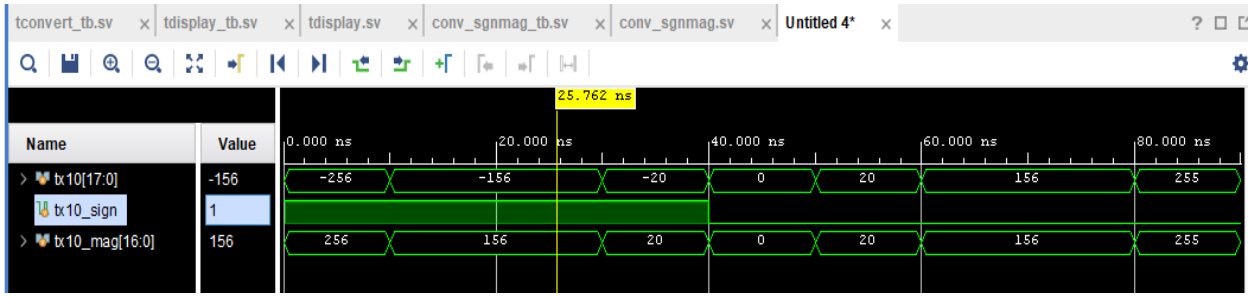

Figure 3 simulation scope for conv_sgnmag

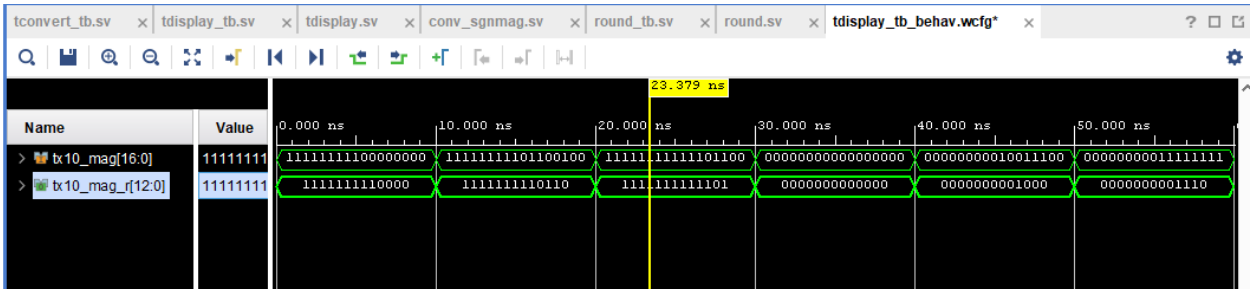

Figure 4round_tb simulation scope screenshot

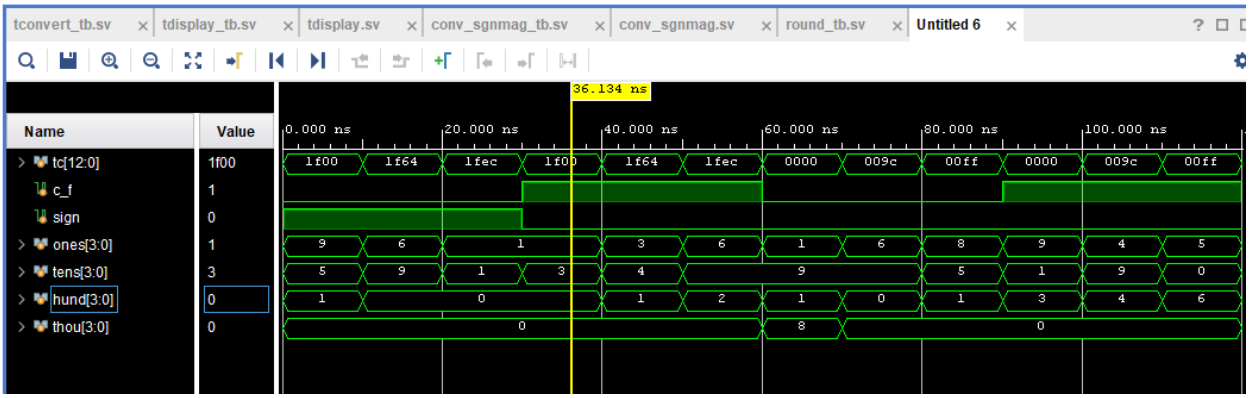

Figure 5 tdisplay_tb simulation scope

The room temperature at the time of running tests on the design was found out to be 22 degrees Celsius

In Fahrenheit, the seven-segment display displayed 71.6 degrees. These were reasonable values and the design was therefore a success.

Elaborated Design is out-of-date. Design sources were modified. details Reload

Project Summary | Schematic | tconvert_tb.sv | tdisplay_tb.sv | tdisplay.sv | conv_sgnma

C:/Users/otienom/Documents/ece212_alex_maurice/lab3/project file/lab3_0/lab3_0.runs/synth_1/lab03_top.vds

```
151         3 Input   18 Bit       Adders := 1
152         2 Input   17 Bit       Adders := 1
153         2 Input   13 Bit       Adders := 1
154         2 Input    8 Bit       Adders := 2
155         2 Input    4 Bit       Adders := 23
156         2 Input    3 Bit       Adders := 2
157         2 Input    2 Bit       Adders := 3
158 +---XORs :
159         2 Input      1 Bit         XORs := 1
160 +---Registers :
161                     16 Bit    Registers := 1
162                      8 Bit    Registers := 5
163                      4 Bit    Registers := 1
164                      3 Bit    Registers := 4
165                      2 Bit    Registers := 4
166                      1 Bit    Registers := 7
167 +---Muxes :
168         2 Input   18 Bit        Muxes := 1
169         4 Input   17 Bit        Muxes := 1
170         2 Input   17 Bit        Muxes := 1
171         2 Input   13 Bit        Muxes := 1
172         2 Input    8 Bit        Muxes := 4
173         8 Input    8 Bit        Muxes := 3
174        17 Input    7 Bit        Muxes := 1
175         2 Input    7 Bit        Muxes := 3
176         9 Input    4 Bit        Muxes := 1
177        27 Input    4 Bit        Muxes := 1
178         2 Input    4 Bit        Muxes := 22
179         8 Input    3 Bit        Muxes := 1
180         4 Input    3 Bit        Muxes := 1
181         2 Input    3 Bit        Muxes := 4
182         3 Input    2 Bit        Muxes := 1
183         2 Input    2 Bit        Muxes := 1
184         2 Input    1 Bit        Muxes := 31
185         4 Input    1 Bit        Muxes := 20
186         8 Input    1 Bit        Muxes := 7
187         3 Input    1 Bit        Muxes := 5
188         9 Input    1 Bit        Muxes := 1
189 ---------------------------------------------------------------------------------
190 Finished RTL Component Statistics
191 ---------------------------------------------------------------------------------
192 ---------------------------------------------------------------------------------
```

4. **Conclusion**

   In this lab we learnt how to manipulate fixed point numbers and to include Intellectual properties in our design. We also learnt a little about how I^2C interfacing works and how the connection can be done in system Verilog.

5. **Time spent on the lab**

   3 hrs.