Name: ____________________ CM: ____________ Start Date: Tuesday, October 23, 2018

Name: ____________________ CM: ____________ Due Date: Tuesday, October 30, 2018

ECE433-01 Fall 2018 Lab #7 Phase II
Reading Three TMP101 Sensors with Nexys 3 Board

This is a group lab to be performed by groups of two students. Each group will need to submit one report that should include names and CM numbers of both students. All submitted source files must have a header to include names, CM numbers, date and brief description of purpose. All submitted simulation waveforms must be annotated, initialed, and dated by both students of the group.

I. Deliverables Due by Tuesday, October 30, 2018

- *Demonstrate your TMP101-based thermometer on Nexys 3 to display temperature from one of three TMP101 sensors in Fahrenheit and Celsius on four digits 7-segment displays. The temperature is displayed in two decimal digits from 0 to 99 degrees.*
- *The lower 3-bit address of the TMP101 sensor to be displayed is set by three slide switches. There are three addresses: 3'b000, 3'b001, 3'b010. When the 3-bit address changes, its corresponding TMP101 sensor should be sensed and its 2-digit temperature in both Fahrenheit and Celsius should be displayed.*
- *Submit to Lab #7 Phase II Submission Folder on Moodle, a pdf copy of new modules you have created for this lab, including Verilog modules, top-level schematic, simulation test bench files and simulation waveforms for sub-modules if required. All files should have a header to include your names, CM numbers, Date, Brief description of purpose. All simulation waveforms must be signed by all members of the group.*
- *Submit a hard copy of the I2C bus waveforms to send addresses and read TMP101 chips with at least three different addresses and their temperatures captured on an oscilloscope. Show screen shots of complete I2C frame transmissions. Annotate your waveforms to translate temperature values into decimal numbers. The waveforms must be initialed and dated by both students.*
- *Submit a copy of your project folder in rar or zip to Lab #7 Phase II Submission Folder on Moodle. Your project folder should contain all source files.*

II. Objective

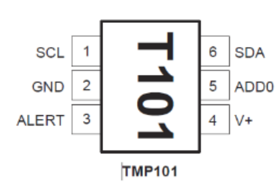
To read the upper 8-bits of the 12-bit temperature registers from TMP101 chips at different addresses, one at a time and display the temperature of the chip currently addressed in Fahrenheit and Celsius in two decimal digits for each TMP101 on 7-segment displays of Nexys 3 board.

III. Available Files

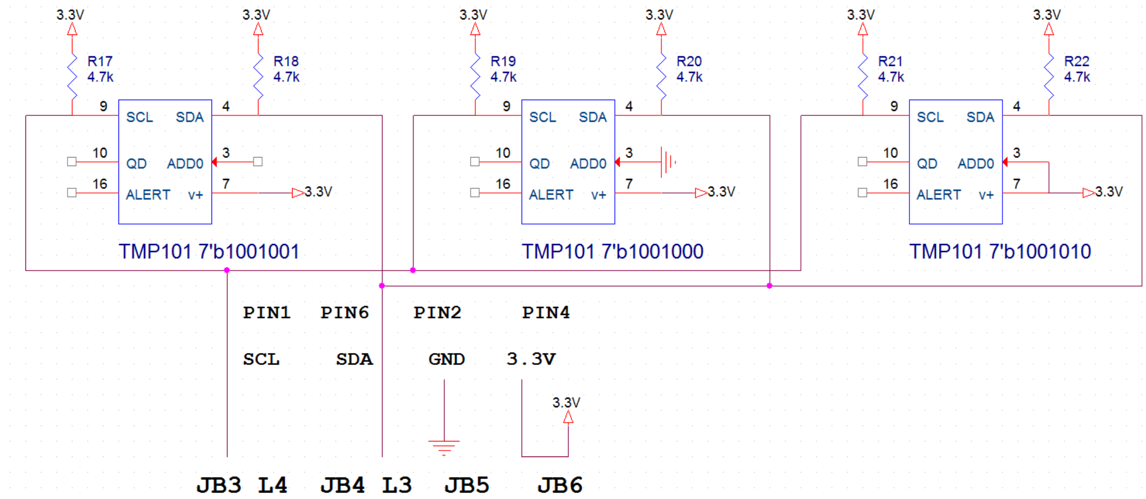
Lab7Phase2I2C2018fallJJS_JJS.v and ControllerReadTempI2C_tb.v are available from Week 9 on Moodle.

IV. TMP101 Interface

TMP101 is a 6-pin SOT23 device. SOT23 stands for small outline transistor 23. It is about 2mm x 3mm. The chip is soldered on to a 6-pin TMP101 PCB board provided in the lab kit. It can be connected to one of Pmod connectors of the Nexys 3 board.



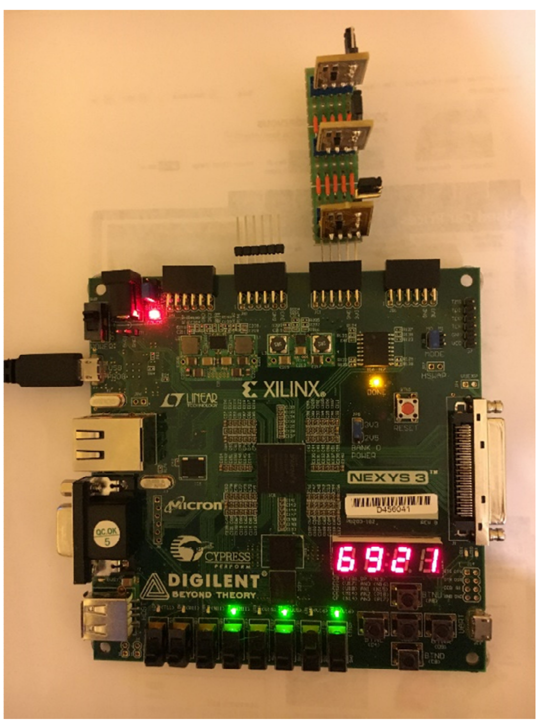
A 4-pin TMP101 board adaptor shown below is made on a perforated board to make it easy to connect the TMP101 board to a Pmod connector of Nexys 3 board. Three possible addresses are possible. The default is address 7'b1001001 when AD0 is floating. The other two addresses are 7'b1001000 when AD0 is grounded and 7'b1001010 when AD0 is at 3.3V. TMP101 can be connected to Nexys 3 as shown in the following schematic. One possible connection is JB3=SCL, JB4=SDA, JB5=GND, and JB6=3.3.

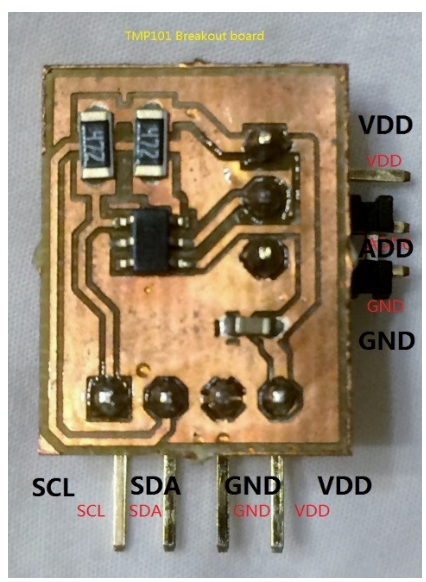


V. TMP101 breakout Boards and I2C waveforms on Nexys 3 Board

V.1 A photo of a single TMP101 sensor on Nexys 3 board

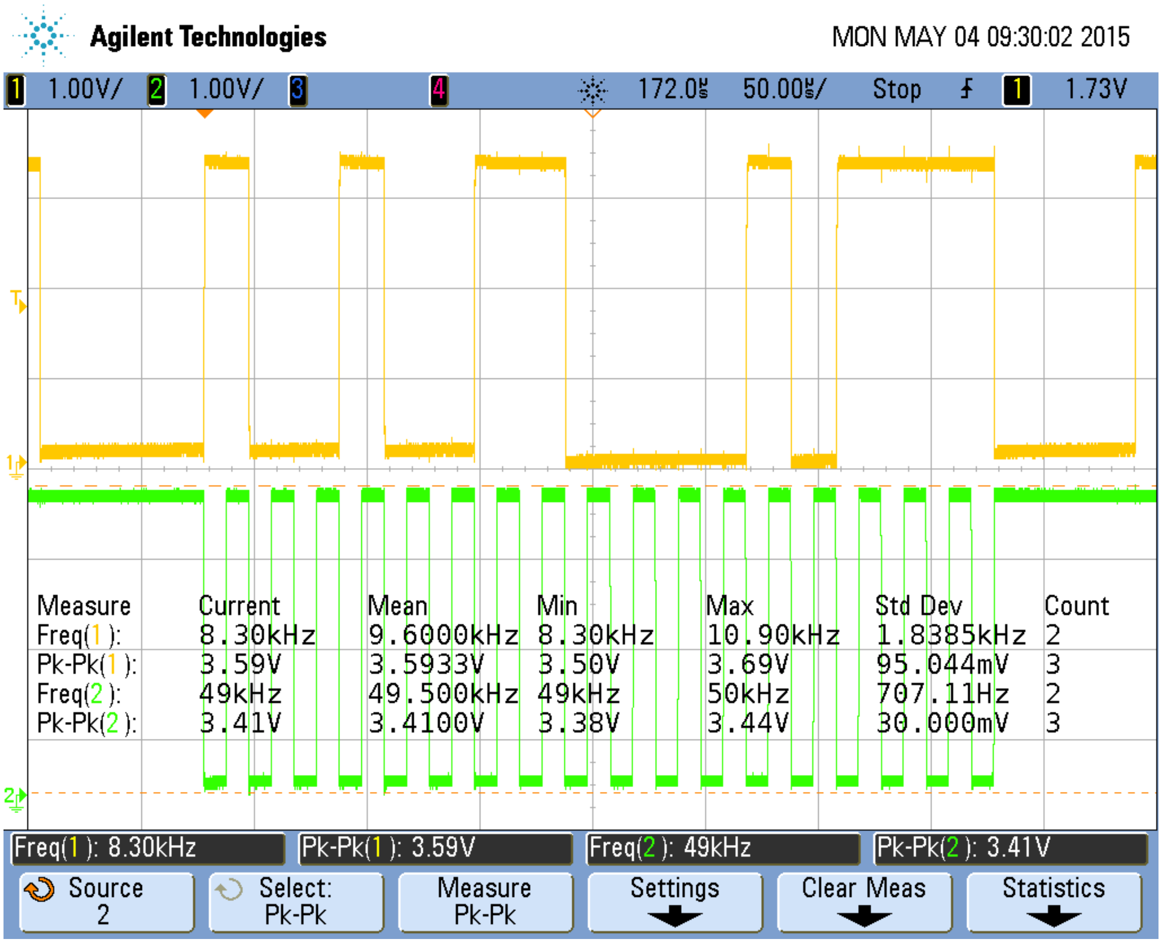
Temperatures 69°F *and* 21°C from one TMP101 breakout board are shown below. The 8-bit temperature is 8'b00010101=8'd21, which is shown on 8-bit LEDs. The measurement is one snapshot only each time a button is pressed to logic high. Notice three TMP101 breakout boards are connected on the same I2C bus line on Pmod JC connector.





V.3 Single TMP101 snap shot waveforms

First byte is 8’b1001001. Second byte is 8’b00010111=23 °C.

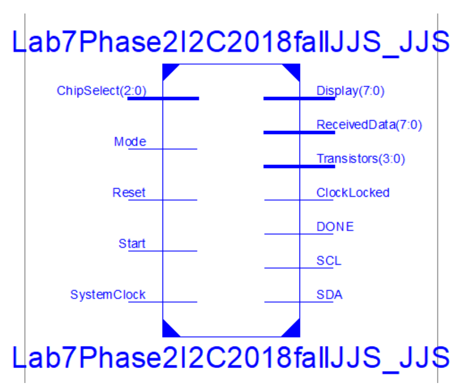


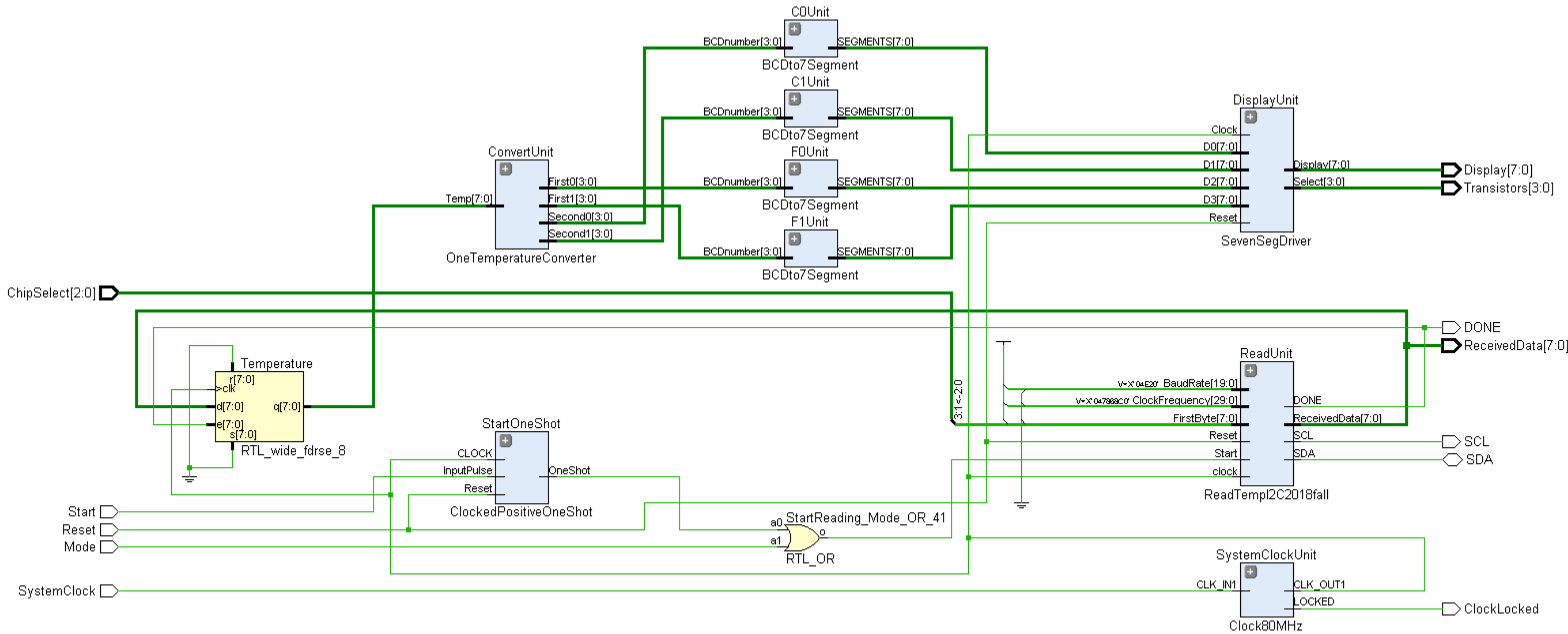
VI. Reading One TMP101 Sensor and Display Temperatures in Fahrenheit and Celsius

Module Lab7Phase2I2C2018fallJJS_JJS.v is the top level module for this lab. ChipSelect is a 3-bit number to select the lower 3-bits of a TMP101 sensor. ChipSelect is connected to three slide switches. The Mode is also connected to a slide switch to choose a single-shot mode or continuous reading mode. On the single shot mode, push button Start is pressed to start a single reading of temperature. One continuous mode, the circuit will read temperature of a TMP101 continuously. ClockLocked and DONE are not debugging pins that can be connected to pins of the Pmod connectors but are not connected for normal operation.

VI.1 Module Lab7Phase2I2C2018fallJJS_JJS.v

Top-level module Lab7Phase2I2C2018fallJJS_JJS.v is shown on the right hand side. ChipSelect is a 3-bit input to set the lower 3-bit address of a TMP101. Start is a one-bit one shot signal to start one conversion each time it goes from low to high. Mode is “0” for single shot mode and Mode is “1” for continuous reading mode. Two digit temperature is displayed in both Fahrenheit and Celsius. ReceivedData is the received byte to be displayed on 8 LEDs. ClockLocked is not connected to a peripheral device.



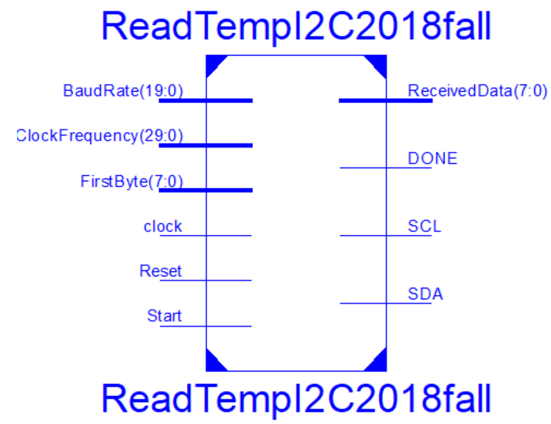

C0Unit
BCDnumber[3:0]
SEGMENTS[7:0]
BCDto7Segment
C1Unit
BCDnumber[3:0]
SEGMENTS[7:0]
BCDto7Segment
F0Unit
BCDnumber[3:0]
SEGMENTS[7:0]
BCDto7Segment
F1Unit
BCDnumber[3:0]
SEGMENTS[7:0]
BCDto7Segment
ConvertUnit
Temp[7:0]
First0[3:0]
First1[3:0]
Second0[3:0]
Second1[3:0]
OneTemperatureConverter
DisplayUnit
Clock
D0[7:0]
D1[7:0]
D2[7:0]
D3[7:0]
Reset
Display[7:0]
Select[3:0]
SevenSegDriver
Display[7:0]
Transistors[3:0]
ChipSelect[2:0]
DONE
ReceivedData[7:0]
Temperature
r[7:0]
clk
d[7:0]
q[7:0]
e[7:0]
s[7:0]
RTL_wide_fdrse_8
ReadUnit
BaudRate[19:0]
ClockFrequency[29:0]
FirstByte[7:0]
Reset
Start
clock
DONE
ReceivedData[7:0]
SCL
SDA
ReadTempI2C2018fall
SCL
SDA
StartOneShot
CLOCK
InputPulse
Reset
OneShot
ClockedPositiveOneShot
Start
Reset
Mode
StartReading_Mode_OR_41
a0
a1
o
RTL_OR
SystemClockUnit
CLK_IN1
CLK_OUT1
LOCKED
Clock80MHz
SystemClock
ClockLocked

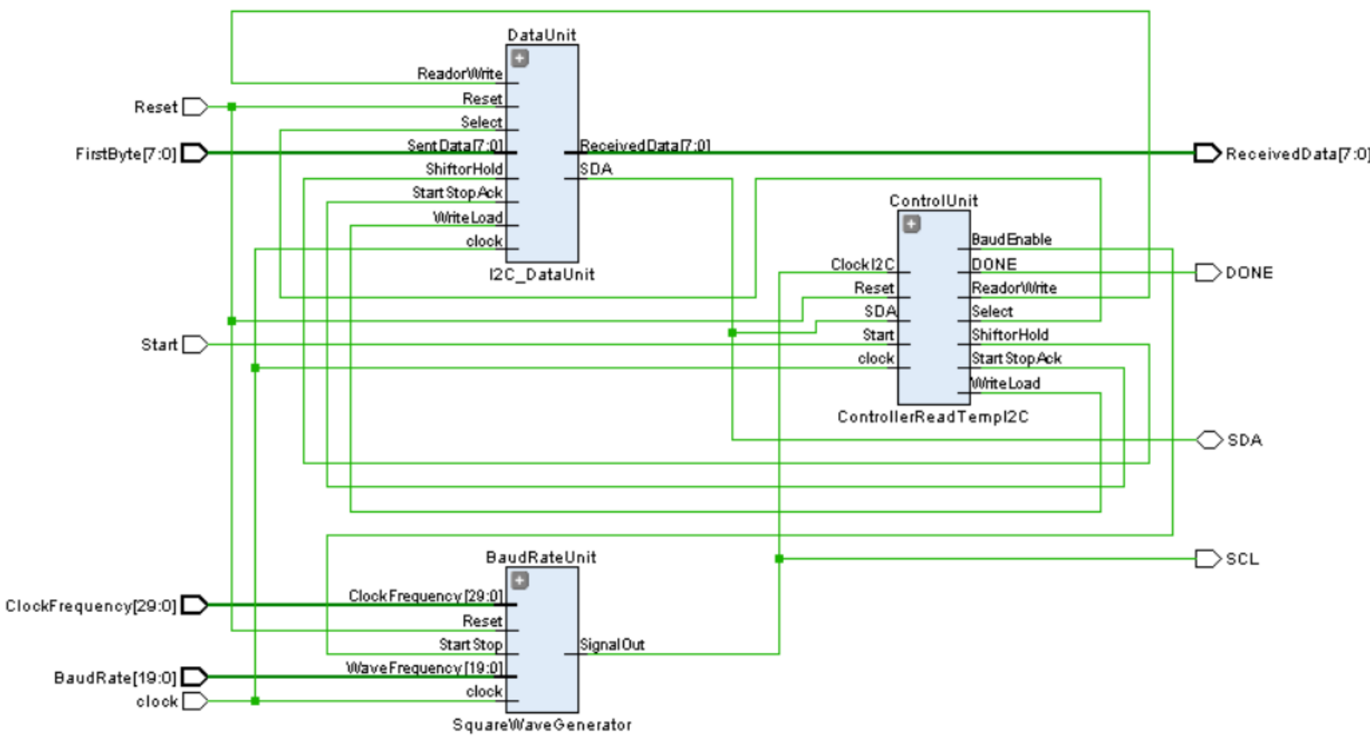
DONE==1 when the reading is complete, which is not connected to an external display.

This module is composed of a number of submodules: ReadTempI2C2017fall, Clock80MHz, OneTemperatureConverster, etc. ReadTempI2C2017fall.v is the main submodule that needs to be created and debugged.

VI.2 Module ReadTempI2C2018fall.v

This module will start reading a TMP101 from address FirstByte when input Start=1 and display 8-bit temperature as ReceviedData and set Done =1 to indicate completion of temperature reading. I2C baud rate and system clock frequency are also to inputs to this module.





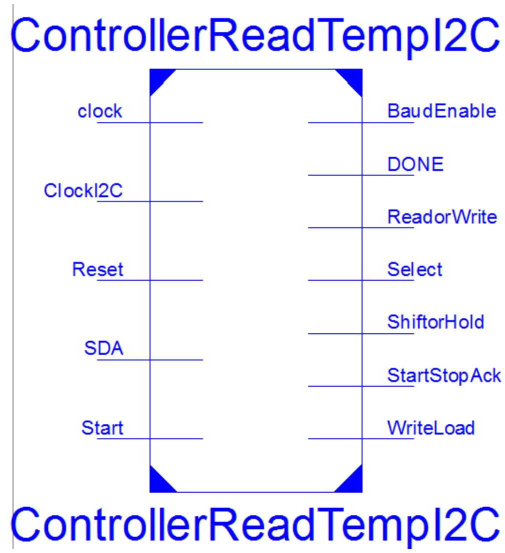
This module is composed of three submodules: I2C_DataUnit.v and SquareWaveGenerator.v and ControllerReadTempI2C.v. The first two modules are created from Phase I of this lab. ControllerReadTempI2C.v will need to be created.

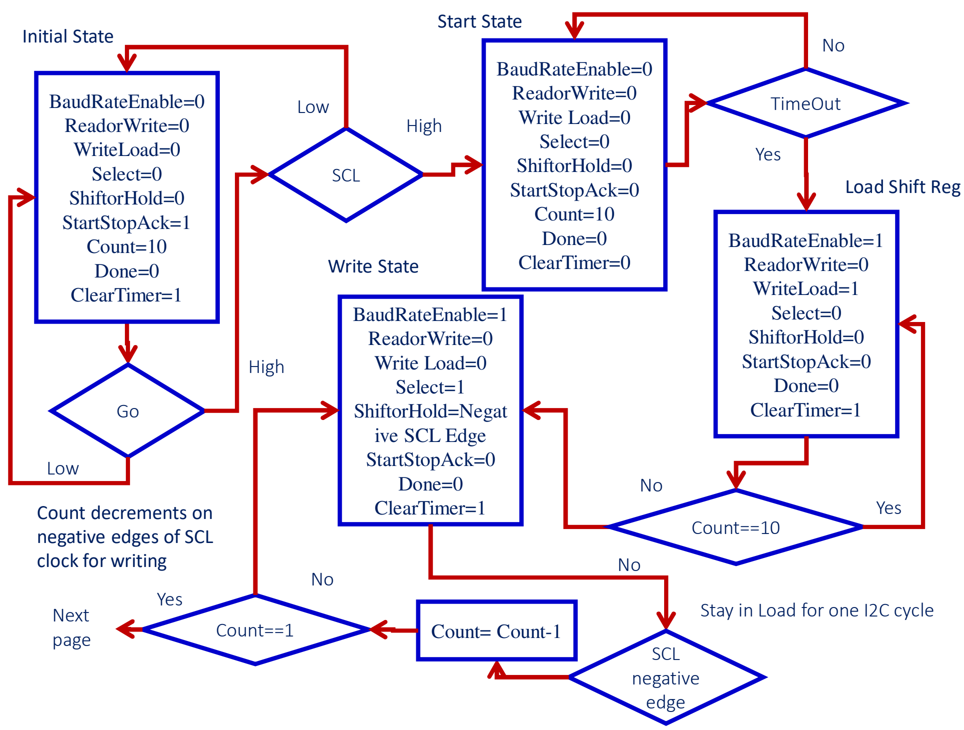
```
//File Name: Lab7Phase2I2C2018fallJJS_JJS.v
//Author: Jianjian Song
//Date: Oct. 2018
//ECE433 Fall 2018
//ChipSelect is the lower 3-bit address of TM101 on SW7-5
//Read one of TMP101 temperature sensor
//send first byte to I2C bus with slave address
//Receive first byte from I2C bus as temperature
//Display the 8-bit temperature in Celsius on LEDs LD7-0
//Display 2-digit temperature in Fahrenheit and 2-digit in Celsius
//on 4-digit 7-segment display

module Lab7Phase2I2C2018fallJJS_JJS(ChipSelect, Start, Mode, ReceivedData, SCL, SDA, Reset,
   SystemClock, ClockLocked, DONE, Display,Transistors);

input Reset, Mode, SystemClock, Start;
input [2:0] ChipSelect;
output SCL, ClockLocked, DONE;
inout SDA;
output [7:0] ReceivedData;
output [3:0] Transistors;
output [7:0] Display;
parameter BaudRate=20'd30000, ClockFrequency=30'd60000000;
wire clock;
//simulation parameter
//parameter BaudRate=2, ClockFrequency=12;

Clk60MHz SystemClockUnit(SystemClock,clock,ClockLocked);

ClockedPositiveOneShot StartOneShot(Start, StartReading, Reset, clock) ;

wire [7:0] Chip;
wire WriteLoad, ReadorWrite, ShiftorHold, Select, BaudEnable, StartStopAck;

//module ReadTempI2C2018fall(Start, Address, ReceivedData, Done, SCL, SDA,
   //BaudRate, ClockFrequency, Reset, clock);
assign Chip = {4'b1001,ChipSelect,1'b1};
ReadTempI2C2018fall ReadUnit(StartReading||Mode, Chip, ReceivedData, DONE, SCL, SDA,
   BaudRate, ClockFrequency, Reset, clock);

wire [3:0] First1, First0, Second1, Second0;
wire [7:0] F1code,F0code,S1code,S0code;
reg [7:0] Temperature;
always@(posedge clock)
   if(DONE==1) Temperature<=ReceivedData;
   else Temperature<=Temperature;
OneTemperatureConverter ConvertUnit(Temperature, First1, First0, Second1, Second0);
BCDto7Segment F1Unit(First1,F1code);
BCDto7Segment F0Unit(First0,F0code);
BCDto7Segment C1Unit(Second1, S1code);
BCDto7Segment C0Unit(Second0,S0code);
//7-segment display
//module SevenSegDriver(D3, D2, D1, D0, Display, Reset, Clock, Select);
SevenSegDriver DisplayUnit(F1code,F0code,S1code,S0code, Display,
   Reset, clock, Transistors);
endmodule
```

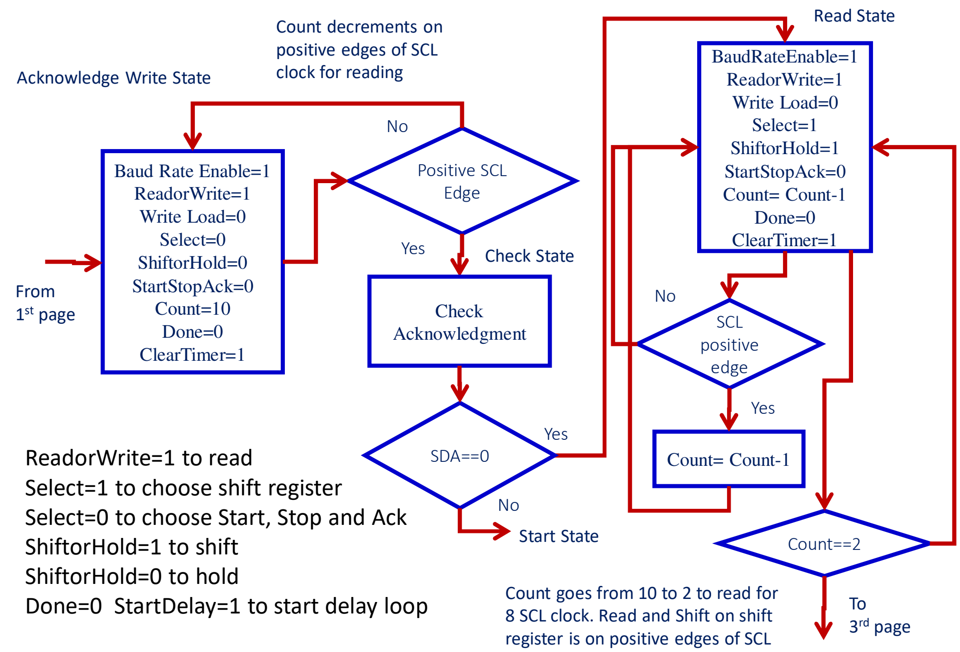
VI.3 ControllerReadTempI2C.v

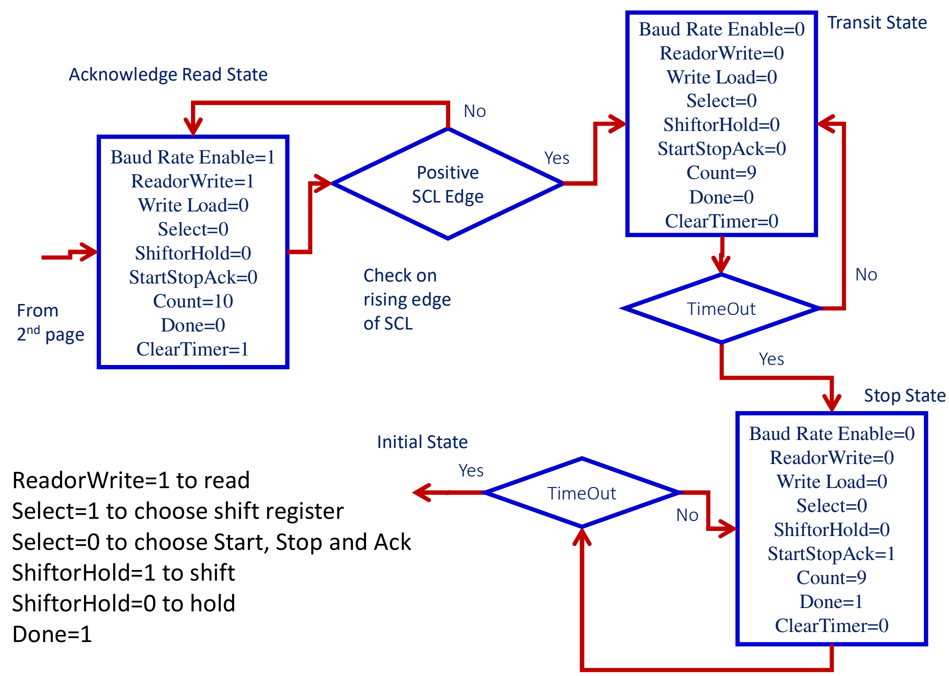
The controller uses one shots from both positive and negative edges of SCL signal to synchronize the state changes and counter updates. The example circuit statements for them are given below as well as a recommended ASM chart for the controller.

```
wire OneShotNegative, OneShotPositive;
ClockedNegativeOneShot OneShotNegativeUnit(ClockI2C, OneShotNegative, Reset, clock);
ClockedPositiveOneShot OneShotPositiveUnit(ClockI2C, OneShotPositive, Reset, clock);
reg ACKbit;
always@(posedge clock)
   if(Reset==1) begin State<=InitialState; ACKbit<=1; end
   else begin State<=NextState;
                if(OneShotPositive==1) ACKbit<=SDA; else ACKbit<=ACKbit; end
//count update
always@(posedge clock)
if(Reset==1) begin DataCounter<=4'd10; end
else
case (State)
LoadState: if(OneShotNegative==0) DataCounter<=DataCounter-1'b1;
else DataCounter<=DataCounter;
WriteState: if(OneShotNegative==0) DataCounter<=DataCounter-1'b1;
else DataCounter<=DataCounter;
ReadState: if(OneShotPositive==1) DataCounter<=DataCounter-1'b1;
else DataCounter<=DataCounter;
default: DataCounter<=4'd10;
endcase
```





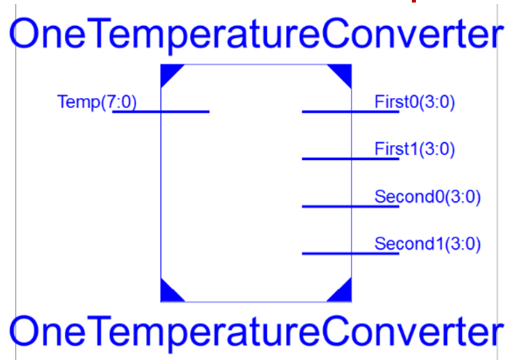




I. Temperature Conversion

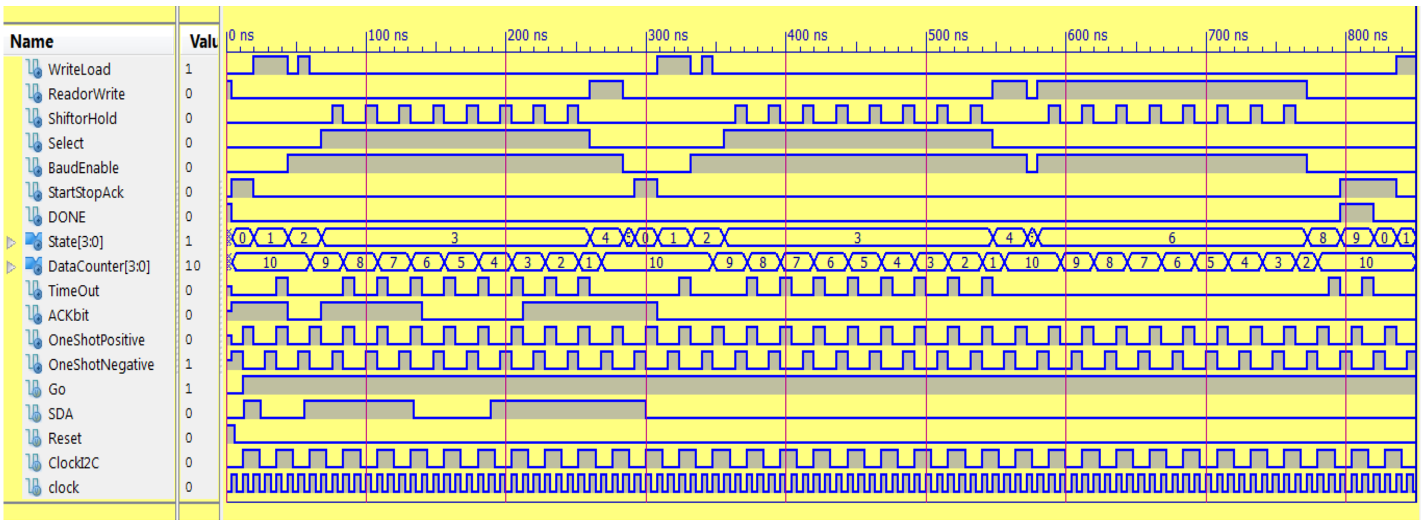
The temperature conversion circuits are given below. The first circuit converts one temperature to both Fahrenheit and Celsius and the second circuit converts two temperatures to Fahrenheit.

```
module OneTemperatureConverter(Temp, First1, First0, Second1, Second0);
input [7:0] Temp;
output [3:0] First1, First0, Second1, Second0;

assign First1 = (Temp*18/10+32)/10;
assign First0 = (Temp*18/10+32)%10;

assign Second1 = Temp/10;
assign Second0 = Temp%10;

endmodule
```



II. ControllerReadTempI2C.v Simulation

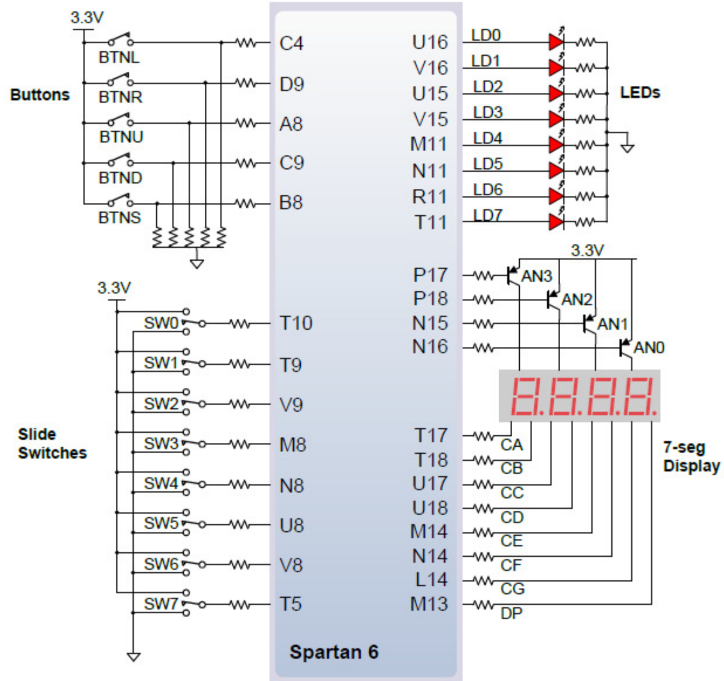
Here is one simulation outcome of the controller.

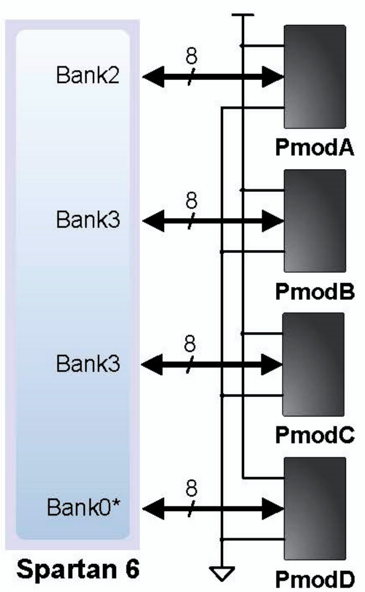


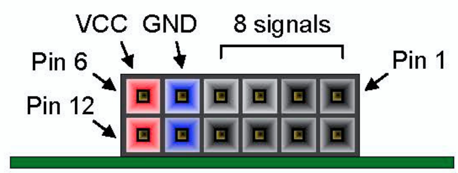
This is the test bench that generates above waveforms.

```
`timescale 1ns / 1ps
module ControllerReadTempI2C_tb;
   reg Go, SDA, Reset, ClockI2C, clock;
   wire WriteLoad, ReadorWrite, ShiftorHold, Select, BaudEnable, StartStopAck,DONE;
   wire [3:0] State=uut.State;
   wire [3:0] DataCounter=uut.DataCounter;
   wire TimeOut=uut.TimeOut, ACKbit=uut.ACKbit;
   wire OneShotPositive=uut.OneShotPositive, OneShotNegative=uut.OneShotNegative;
ControllerReadTempI2C uut (Go, SDA, WriteLoad, ReadorWrite, ShiftorHold, Select,
   BaudEnable, StartStopAck, DONE, Reset, ClockI2C, clock);

   initial begin  Go = 0;  Reset = 0;  ClockI2C = 0;  clock = 0; end
   always #4 clock=~clock;
   always #12 ClockI2C=~ClockI2C;
   initial fork
   #0 Go = 0;  #12 Go = 1;
   #0 Reset = 1;  #6 Reset = 0;
   #0 SDA=0;  #13 SDA=1; #25 SDA=0; #56 SDA=1; #134 SDA=0; #189 SDA=1; #300 SDA=0;
   #850 $stop;
   join
endmodule
```

III. Peripherals and Pmod Connectors on Nexys 3 Board







Pmod Connectors – front view as loaded on PCB

Pmod Pinouts

JA1: T12	JB1: K2	JC1: H3	JD1: G11
JA2: V12	JB2: K1	JC2: L7	JD2: F10
JA3: N10	JB3: L4	JC3: K6	JD3: F11
JA4: P11	JB4: L3	JC4: G3	JD4: E11
JA7: M10	JB7: J3	JC7: G1	JD7: D12
JA8: N9	JB8: J1	JC8: J7	JD8: C12
JA9: U11	JB9: K3	JC9: J6	JD9: F12
JA10: V11	JB10: K5	JC10: F2	JD10: E12

IV. Setting up a scope at triggering mode for MSO7012B from Agilent

- *Press Trigger button to choose Trigger on Edge and select the source and slope.*
- *Press Mode Coupling button to choose Normal Mode*
- *Turn the Trigger Level knob to select voltage level to be within the signal voltage range*
- *Press Single button to set up the scope to wait for the right edge on the input to trigger the storage*
- *Start your signals so that they can be captured on the scope.*