Ahmed Abuhjar
www.aabuhjar.com

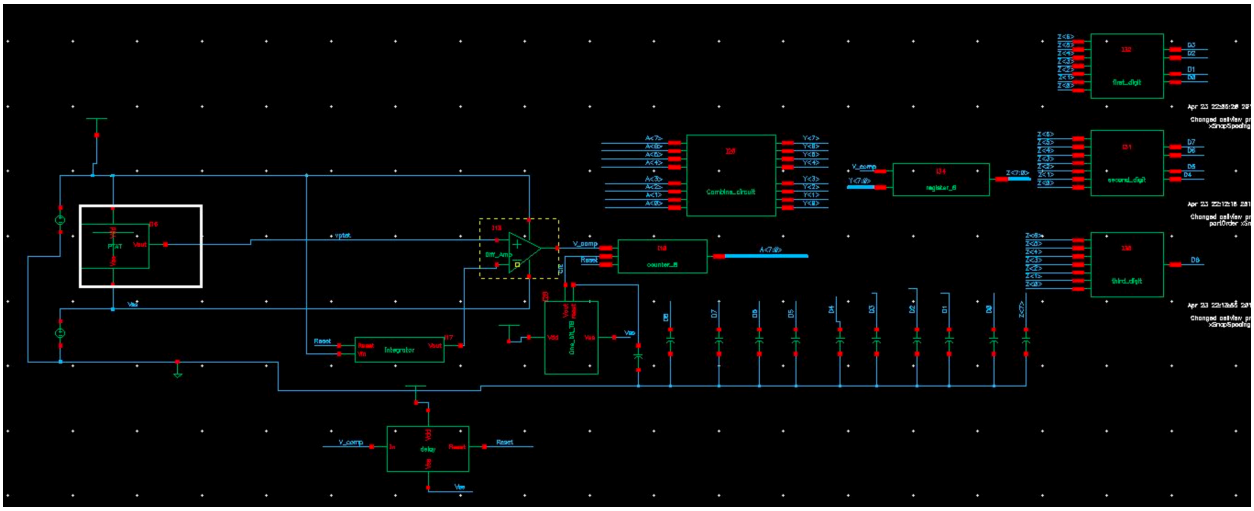
EE 330 Integrated Circuits
Dr. Randy Geiger

May 2018

Temperature Sensor Using Diodes

Introduction:

This temperature sensor design employs temperature-dependent devices, specifically diodes. The circuit diagram depicting the entire configuration is presented below.



Some specs:

The sensor will periodically assess the changing temperature and subsequently exhibit a decimal representation of the measurement in Celsius degrees. The sensor is designed to provide accurate measurements within the temperature range of -20°C to 100°C.

A Boolean signal is generated by the counter based on the voltage difference between two diodes in the PTAT. The counter continuously counts until a predetermined point determined by an integrator circuit. It was essential for the region on the voltage vs. time graph of the integrator to be nearly linear, though not necessarily perfect. This linearity is a crucial aspect of the design to maintain consistent proportionality between the PTAT output and the corresponding Boolean signal from the counter. Otherwise, assigning a Boolean number for each temperature measurement would have been exceedingly complex.
Furthermore, other logic circuits were incorporated into the design, including a combinational circuit that takes inputs from the counter to generate a temperature value in Boolean format. A register was utilized to store the measured values. Subsequently, the Boolean number stored in the register is fed into a BCD converter, which, in turn, drives a 7-segment display device (LF-3011A).

Design:

My design was consisted of only one input, the temperature, which will be measured using highly temperature dependent devices, diodes. Many other ports were used as input and output at the same time. The PTAT output, for example, was used as an input for the comparator as shown in the schematic.

Below is a table that shows the circuits and the corresponding ports used in the design.

Circuit Name Ports Description

Circuit Name	**Ports**	**Description**
PTAT	**Vdd** , **Vss:** Power Sources **Vout** : will be the voltage difference between the two diodes.	The voltage will be a constant value and linearly proportional to the temperature input
Integrator	**Reset** : will be the input to reset the capacitor used in the integrator to the initial condition zero. **Vin** : will be the Vdd input to charge the internal capacitor. **Vout** : will be the capacitor voltage is it charges.	The initial segment of the transient voltage in the internal capacitor will be moderately linear. It will be used to relate the voltage to a time value.
Comparator	**V1 input** : will be the voltage generated from the PTAT **V2 input:** will be the voltage generated from the integrator	The comparator output will remain high as long as the Vptat is higher than that of the integrator. This output will be used to drive the counter.
Clock	**Vdd, Vss:** Power supplies. **Vout** : is the clock output **Reset** : to reset the internal inverter used to design the clock to the zero initial condition.	The clock will have a period determined on how fast or how slow the counter should count in order to have accurate measurements and to avoid any nonlinearities issues.
Counter	**En:** Counter will count only when this output is high. **Clk:** Counter only counts on positive edge of this input. **Reset:** Counter will reset to 0 when this is high. **Y<7:0>:** 8 bit counting output.	Using the integrator, voltage from PTAT, and the comparator I convert the temperature-related-voltage into a time value. 8 bits is one extra bit than what I needed, but I added that for flexibility
Register	**En:** The register will be written into only on the negative edge of this input. **X<7:0>:** 8-bit input. **Y<7:0>:** 8-bit output.	I use the register to pipe the in the binary value in one step instead of having it counting up constantly. I used a negative-edge enable because that fits my purpose better, the comparator going low means my computation from previous steps has ended.

BDC	**A,B,C,D,E,F,G:** 7-bit binary input. **Y<3:0>:** 4-bit BCD output	I will use three BCDs to convert my binary-coded-temperature value to a BCD one. I needed three because I am considering a range that goes up to 100C.
7-Segment Display	**X<3:0>:** BCD input **seg<6:0>:** output used to drive the 7-seg display.	I will use 3 of those to drive 3 different 7-seg displays. A 4th 7-seg will be used but to only display a negative sign when necessarily.

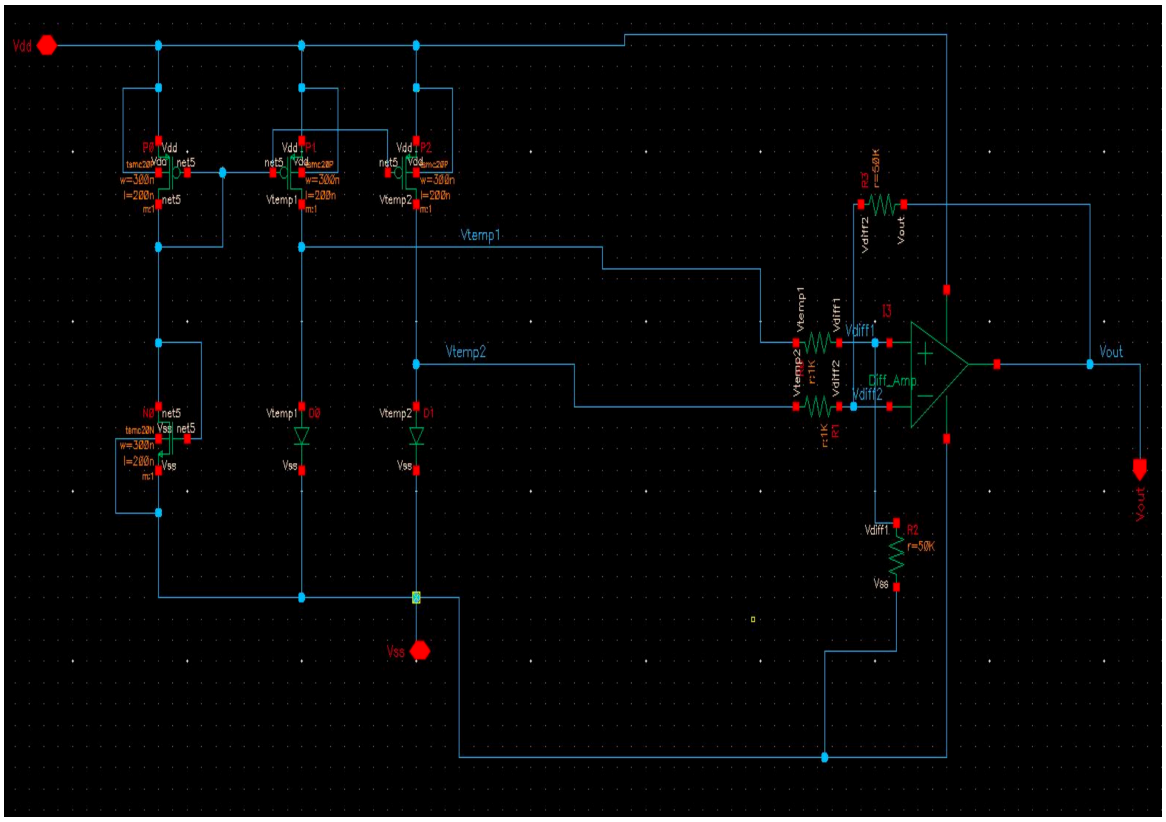
PTAT:

From the exact diode equation, I can recognize that if used the same current through two different diodes, any difference in their voltage would linearly depend on the differences in their area and the temperature, I used this concept to design my PTAT.

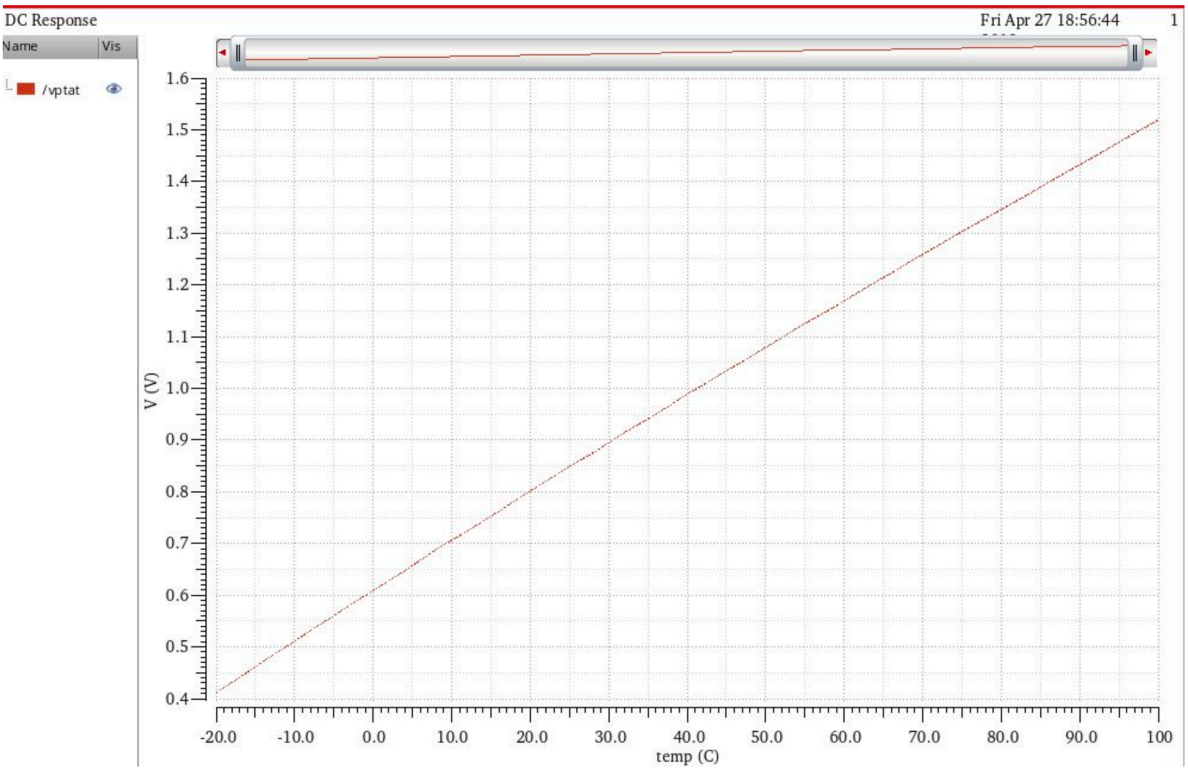
$$I(T) = \left(J_{SX} \left[T^{m} e^{\frac{-V_{G0}}{V_t}} \right] \right) A e^{\frac{V_D}{V_t}}$$

I used a current mirror to create 2 equal currents that run through both diodes, I also set one diode to be 10 times larger than the other (the values for current and diode sizes were based on trial and error methods). Then the difference between the voltages of the 2 diodes is fed into a difference amplifier, which produces a large linear range for me to work with.

PTAT SCHEMATIC:

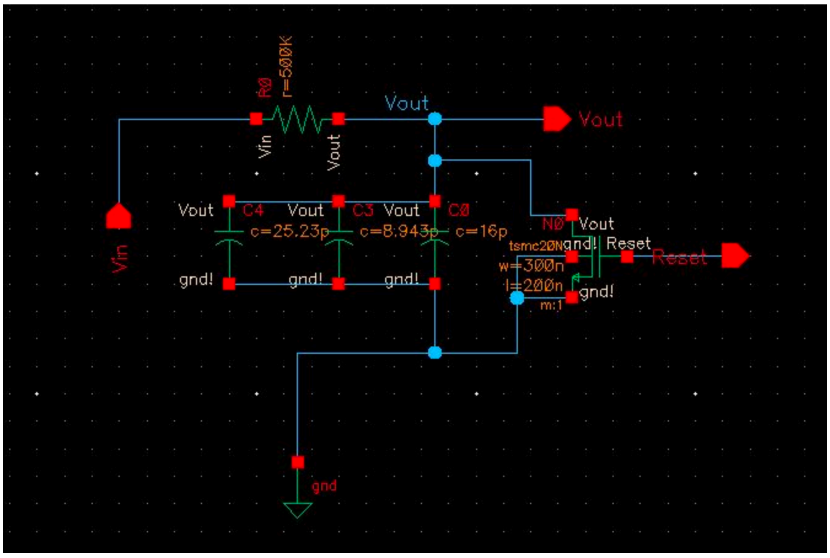


Voltage vs Temperature range:

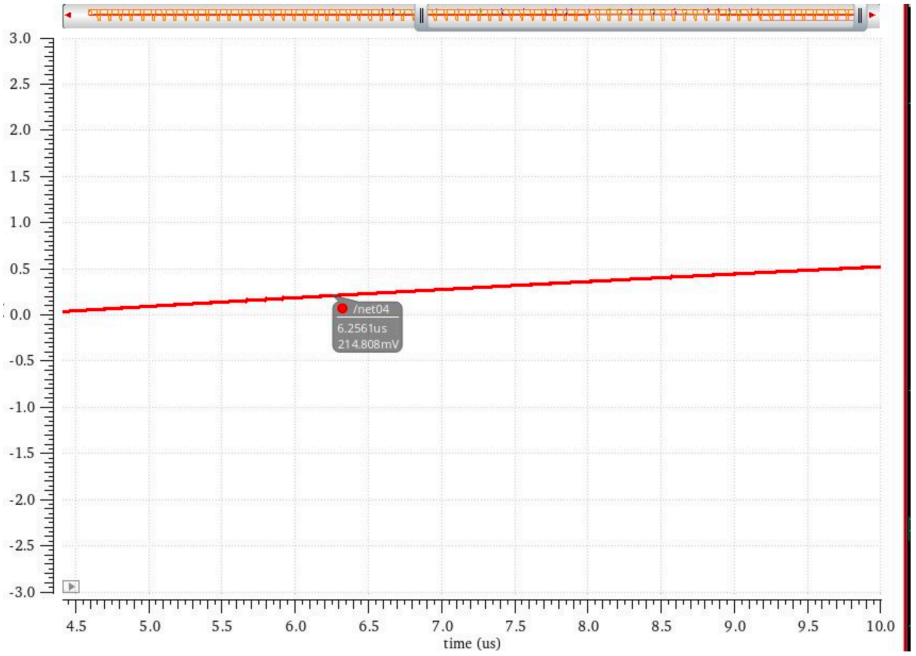


Integrator:

In designing an integrator, I thought of charging a capacitor that will have some sort of delay until it's fully charged. The initial modeeralty linear of the Voltage-time graph will be my interest in this project. The following shows the schematic of the integrator.

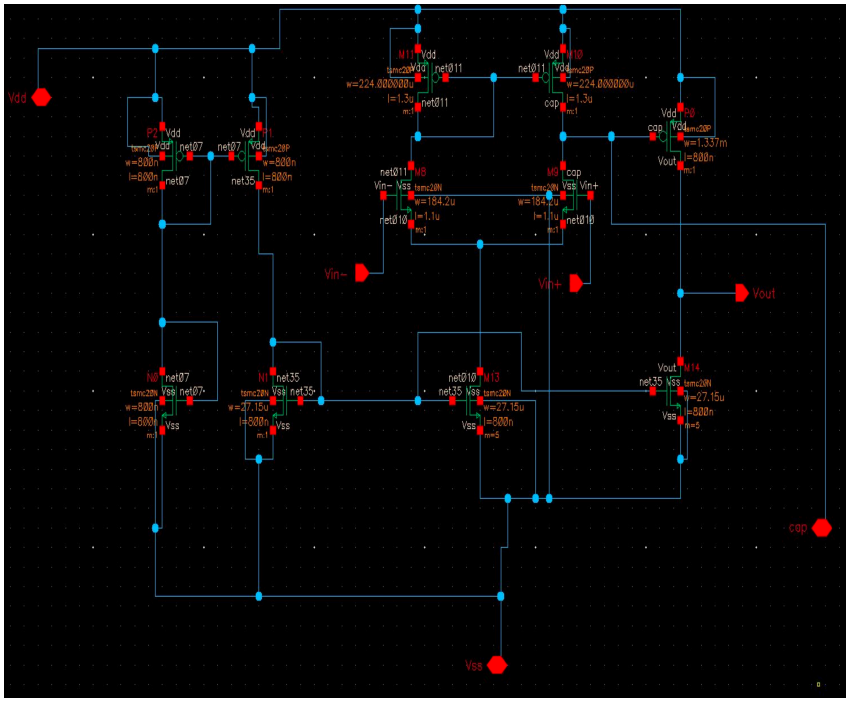


The capacitor was split purposely into three instead of using 50pF which could be quite large for one single capacitor.
A simulation below shows how the integrator works and how the capacitor is charged in a pace (with a slope for the linear segment) I need for the design.



Comparator:

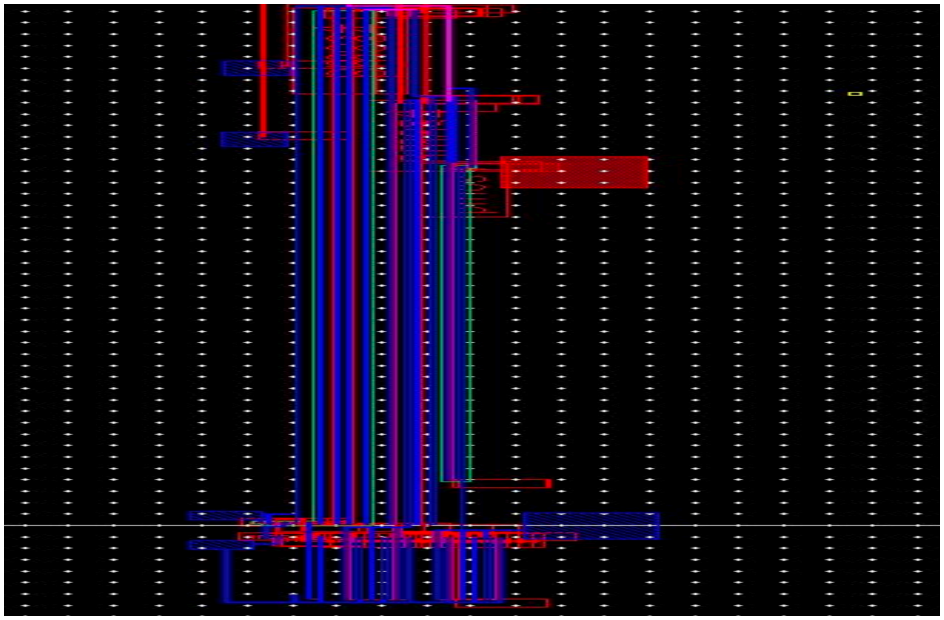
The comparator is made from externally compensated Op Amp described on the EE330 website. I did not measure the exact open loop gain for my Op-Amp but its gain was adequate for my purpose. Shown below is the schematic view of my Op-Amp with the transistors' dimensions shown.



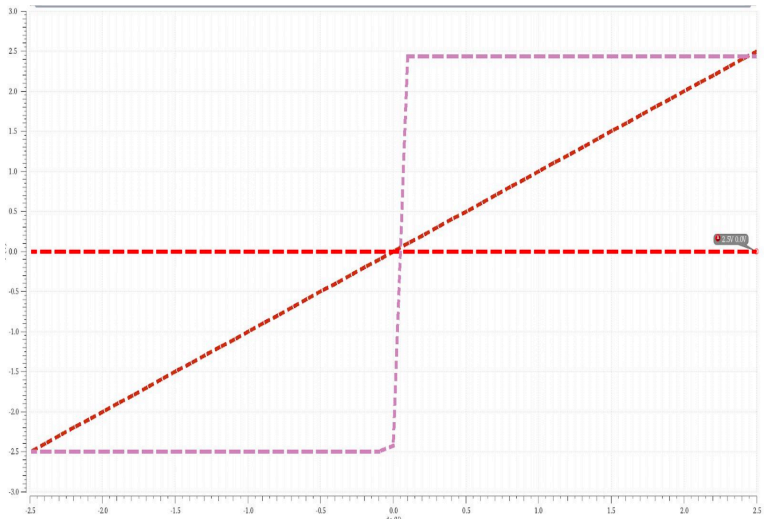
Some of the transistors were 1000 times larger than the others, which is not optimal. I tried to decrease transistor sizes as much as possible, connecting the larger transistors was very tedious.

I used a 10pF external capacitor for my testing. The larger the capacitor, the more stable the Op-Amp is, but I found some time delay which I think it is due to the large external capacitor. After multiple testing, I felt that 10pF capacitor provides a good trade-off between stability and response speed.

Analog-Extracted View:



I tested my Op-Amp by sweeping the positive input terminal and grounding the negative one - the output switched from rail to rail at a difference of 50 µV which was good enough for my purpose.

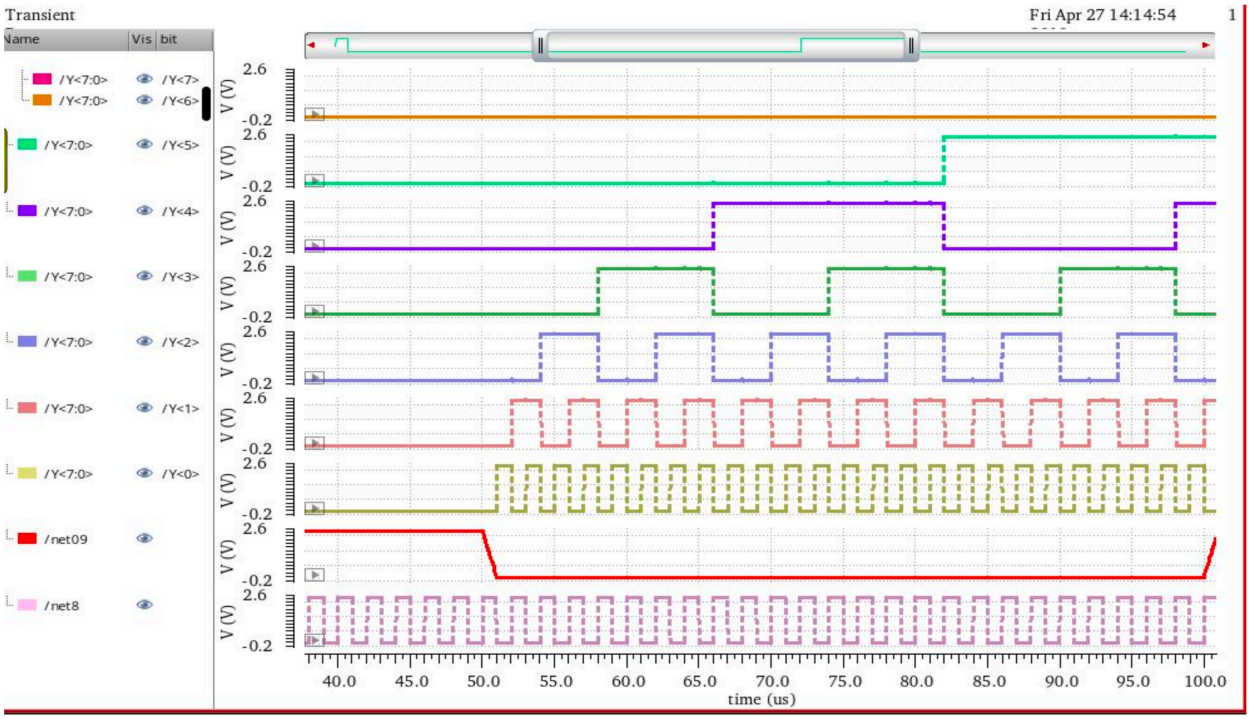


Counter:

The 8 bit was implement using verilog as that was the easiest way. The code for the counter is shown below:

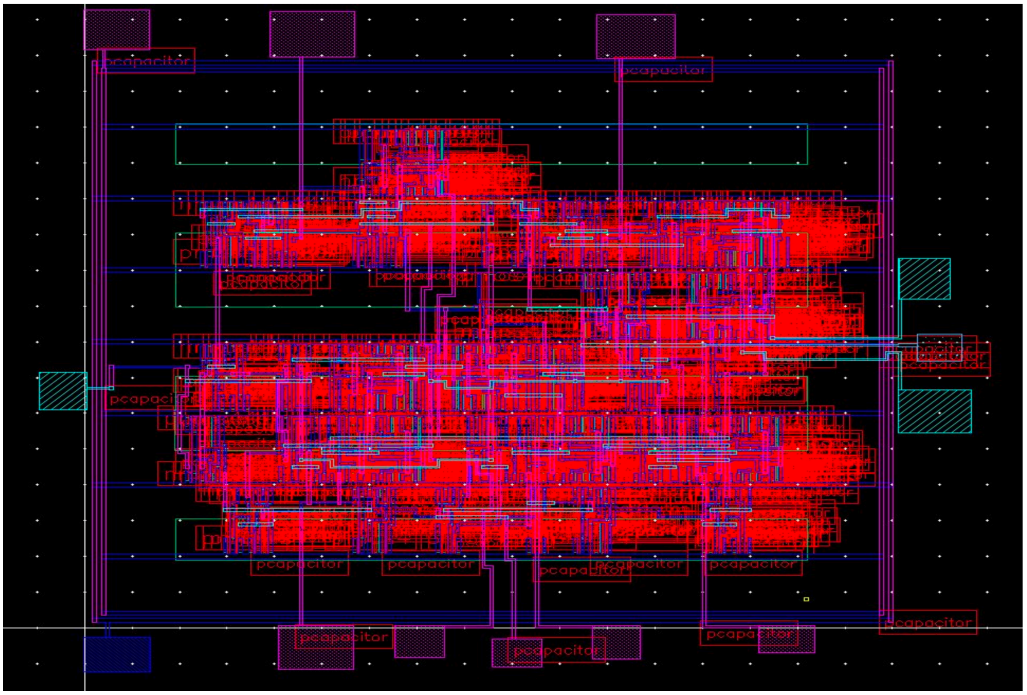
```
module counter_8 (En, Clk,reset,  Y);
        input En,Clk,reset;
        output [7:0] Y;
        reg[7:0] Y = 0;
        always @(posedge Clk)
        if (reset)
                begin
                Y <= 8'b00000000;
                end
        else if (En)
                begin
                Y <= Y+1;
        end
endmodule
```

This is an 8-bit Up Counter with an Enable. The Enable input will come from the comparator, so when the voltage from the PTAT is higher than that of the integrator the counter will count up. Once the integrator voltage is higher than that of the PTAT, comparator output will go low and the counter will stop. The reset will be used to set the counter to 0 in preparation for the next time I have to calculate the temperature. Below is a picture showing my counter working(testing was done post-layout):



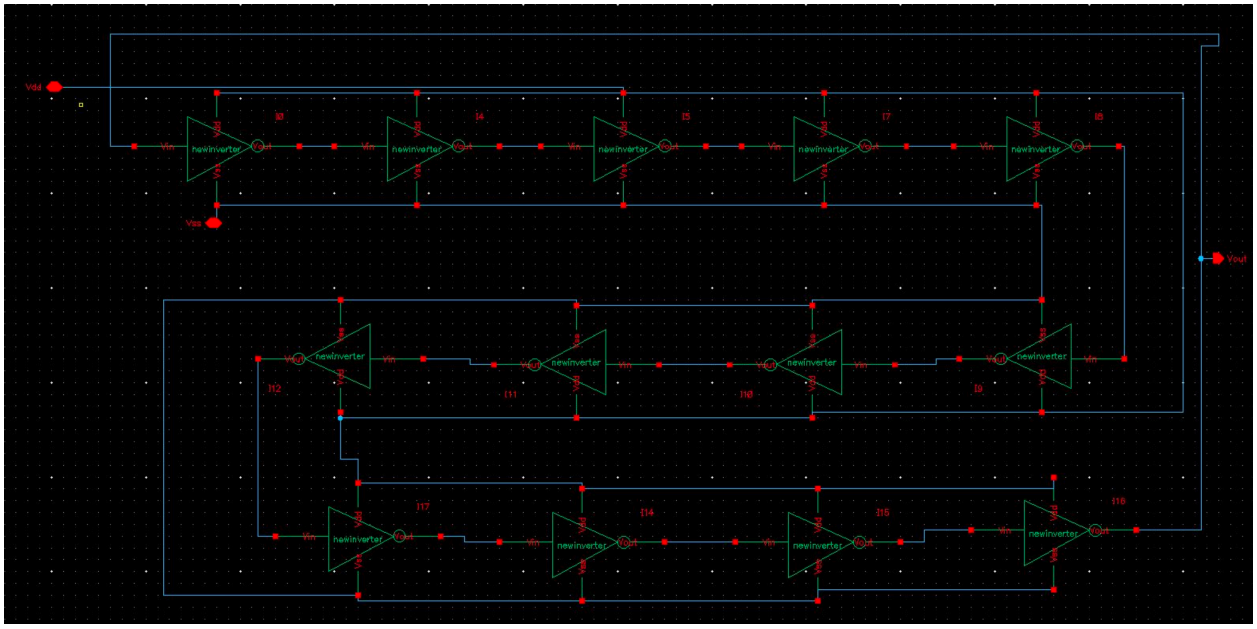
Analog - Extracted View:

This circuit was synthesized using encounter and converted into a layout view. The process was pretty much automatically except for having to connect the input and output pins.

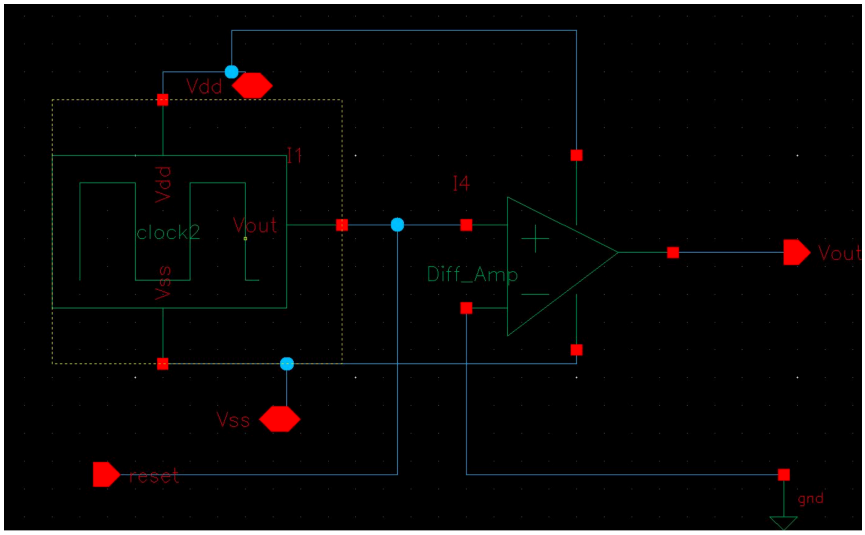


Clock:

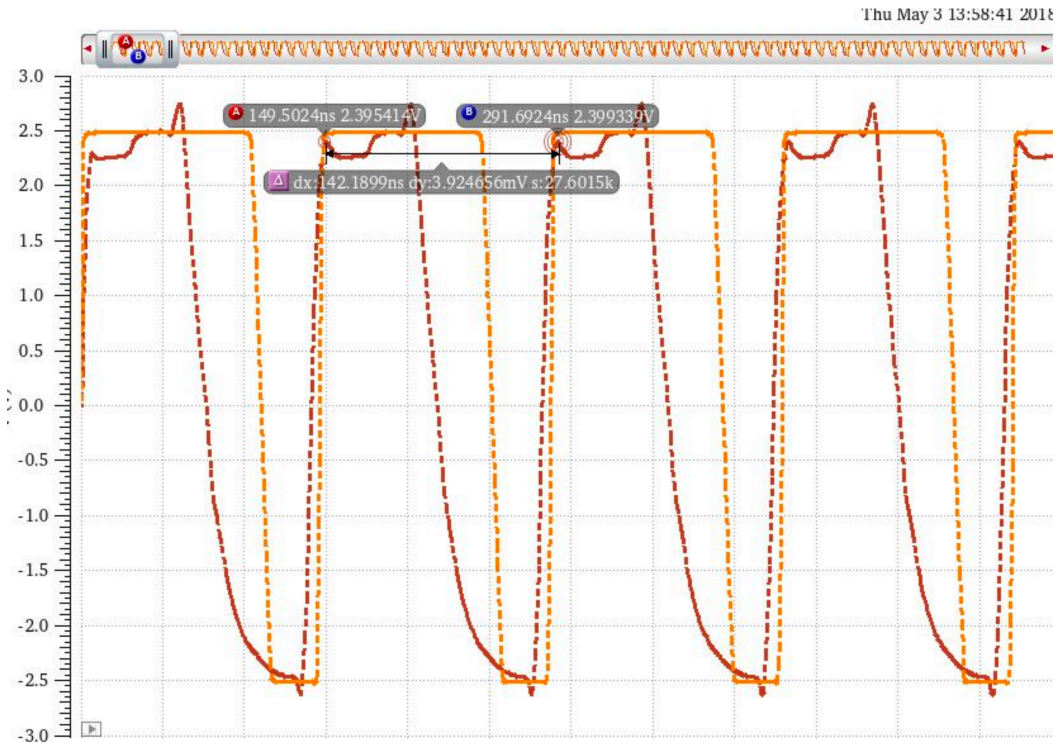
The clock was designed using odd number of inverters. For this project, I needed the clock to have a period of about 140ns. This period was the best I could assign the clock for in order to eventually represent every temperature by more or less one corresponding binary signal, which comes from the counter. The following shows the schematic for the clock designed.



I first ran the simulation for one single inverter to determine the propagation delay for the inverter and then I calculated how many inverters I need for the clock to have about 140ns. In this case, I used 13 inverters. After running the simulation, I observed that the edges of the clock are as perfectly sharp as needed. Hence I used a comparator to sharpen the edges. The schematic is shown below



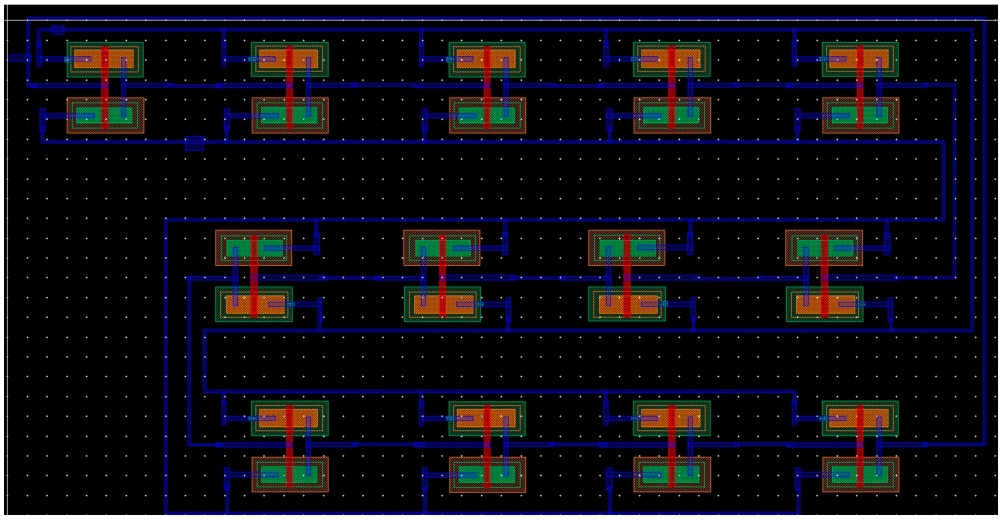
A simulation of both the clock before and after adding the comparator.



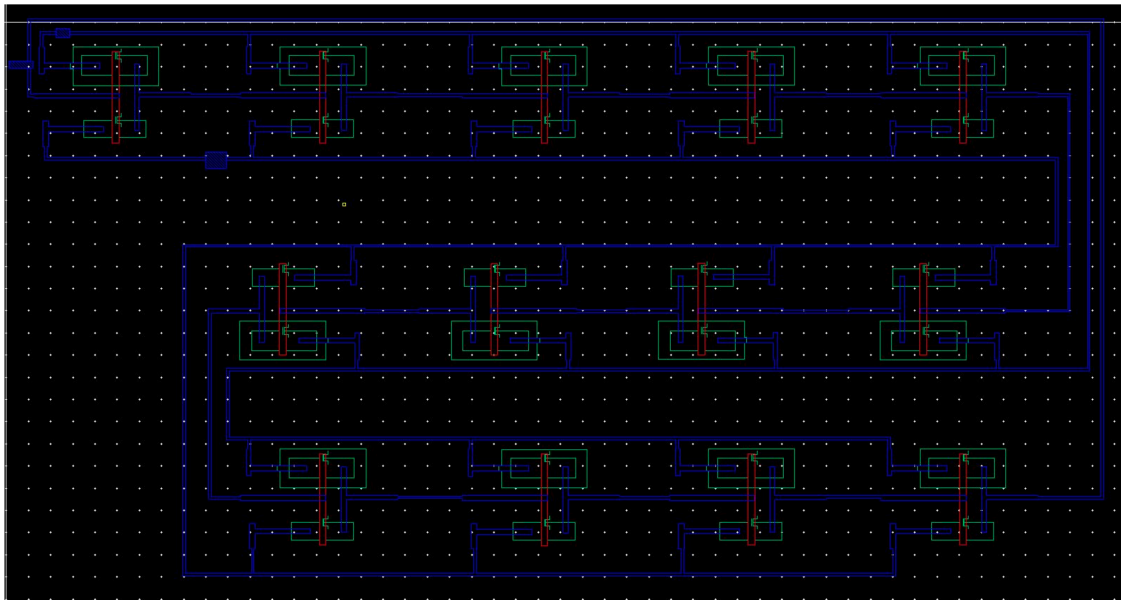
As the graph shows above, the clock has a period of about 142 ns, and since I will only use the clock to trigger the counter, the positive edge of the clock is the only thing that matters in this case.
I faced some unexpected issues during designing the clock and that's it was noticed that the clock period remains to be around 140ns when the temp is between -20 and around 40C but beyond that temperature range, the clock somehow starts to cycle fast and the time period keeps decreasing till it gets to around 126ns at 100C. This behavior could affect how fast the counter goes, and hence it causes an uncertainty of about 4-10 degrees for very high temperatures 65-100C.

In order to fix this, I adjusted the clock time period to be on an average so that the uncertainty for the output is not too big.

THE LAYOUT

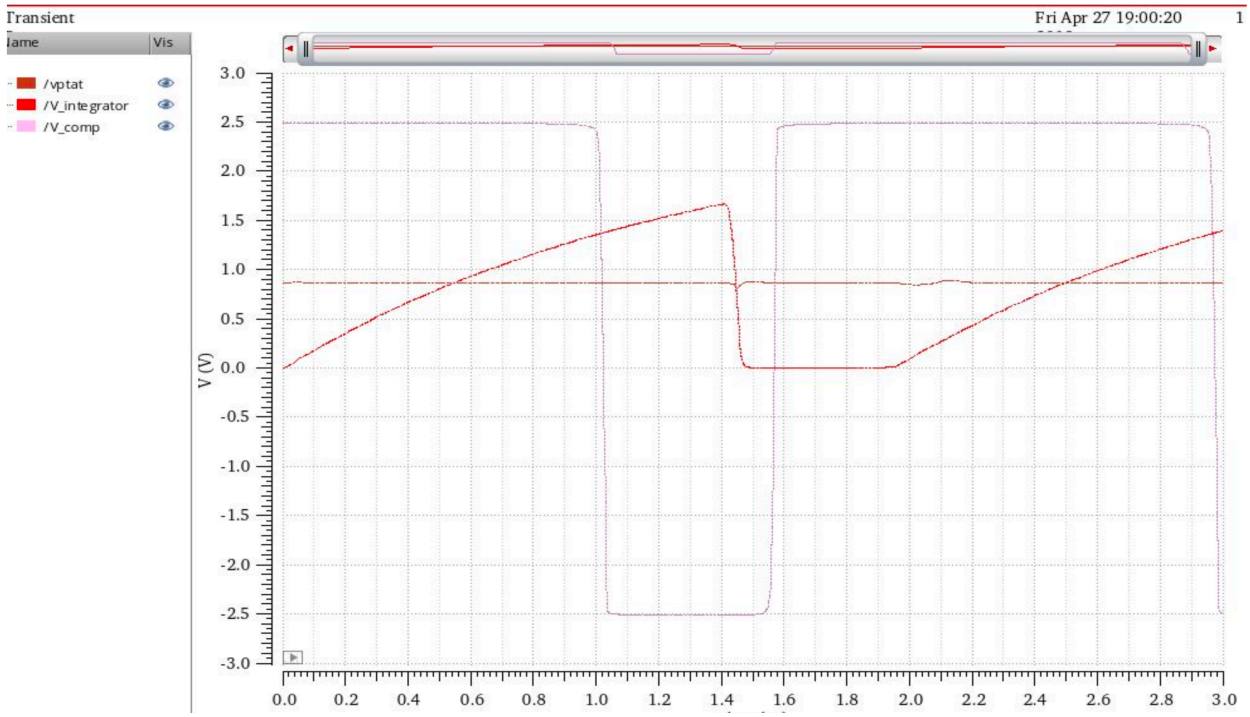


THE EXTRACTED LAYOUT



ADC full test:

When the entire ADC circuit (Integrator, Comparator, and Reset) was put together it showed the behavior presented below.



As you can see, when the integrator and Vptat voltage intersect, there is a slight time delay and then the comparator switched to LOW. After a delay of about 0.5 µs (which is about 5 clock cycles) the integrator's capacitor is discharged and the whole process repeats again. It is worth noting that the comparator switched to HIGH slightly earlier than the capacitor starts recharging which can induce some error on additional runs after the first one.

Combinational circuit:

The combinational circuit is designed to take the Boolean signal from the counter and convert it into a value that represents the corresponding temperature in binary number. I designed a verilog code for this circuit and eventually synthesized the circuit.

Verilog code is shown on the right.

```
`timescale 1ns/1ps

module Combine_circuit(A,B,C,D,E,F,G,H,Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0);

  input A,B,C,D,E,F,G,H;
  output Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0;

assign Y7 = ~A&~B&~D | ~A&~B&~E&~F;
assign Y6 = A | B&C&D&E&H | B&C&D&E&G | B&C&D&E&F;
assign Y5 = B&C&~D | B&C&~E | A&C&E | ~A&~C&D&F | ~A&~C&D&E | A&C&F&H | A&C&F&G | B&C&~F&~G&~H;
assign Y4 = A&D | B&~D&E | ~A&~C&~D&F | ~C&D&~E&~F | B&~D&F&G | B&C&D&~E | A&E&F&G | A&C&~E&~F | ~A&~B&~D&~E&~F |
~A&~B&~D&~E&~G | ~A&~B&~D&~E&~H | A&C&~E&~G&~H | B&C&E&~F&~G&~H;
assign Y3 = A&D | ~B&~C&H | ~B&~C&G | ~B&~C&E | B&~E&~F | ~C&~D&E&F | ~B&E&F&~G | ~C&E&F&H | ~C&E&F&G | ~B&D&E&F | B&C&~E&~G
| B&C&D&~E | A&F&~G&~H | A&C&~E&~F | ~A&~B&~D&E&~F | B&~D&E&F&G&H | B&C&D&~F&~G&~H | ~A&~B&~D&~E&F&G&H;

assign Y1 = ~A&~B&E&G | ~A&~C&~D&G | ~A&~C&G&H | B&~C&~F&G | D&~E&F&G | D&~E&G&~H | A&C&E&~G | ~C&~E&F&G | A&C&E&~H |
~A&~B&G&H | A&C&F&~G&~H | A&E&F&~G&~H | ~B&D&F&G&~H | B&C&~D&E&~G | C&D&E&G&H | A&D&E&F&~G | A&D&E&~G&~H | A&D&~E&~F&~G |
A&C&~E&~F&G | C&~D&E&F&G&~H | B&~D&~E&~F&G&H | ~C&~D&E&~F&G&H | ~B&~C&~D&~E&F&H | ~B&~C&~E&~F&~G&~H | ~A&C&~D&~E&~G&~H |
B&C&E&F&~G&~H | B&C&D&~E&~F&~G&H;
assign Y2 = ~A&~C&~F | ~A&~B&~F&~G | ~C&~D&E&~F | B&~D&~F&G | A&D&~E&F | A&C&~E&~F | ~C&~D&~F&~G&~H | ~B&~C&~D&E&~G |
~A&~B&D&E&~F | D&E&~F&~G&~H | B&~D&~E&~F&H | B&C&~D&F&~G | B&C&~E&F&~G | B&C&F&~G&H | A&D&~E&G&H | A&D&F&G&H |
A&C&~E&~G&~H | A&C&~F&G&H | ~A&~B&~D&~E&~F&~H | ~A&~B&~D&F&G&H | C&~D&E&F&G&~H;
assign Y0 = ~A&~C&~F&H | ~C&~D&F&H | ~C&~E&F&H | B&C&D&~H | A&C&F&~H | ~A&~B&~D&~E&H | ~B&~C&~D&F&~G | ~A&~B&~E&F&H |
~B&~C&F&~G&H | ~D&~E&F&G&H | ~A&~B&D&E&H | C&D&~E&~F&~H | B&~D&E&~F&H | B&~D&E&~G&H | B&C&~E&~F&~H | B&C&~E&~G&~H |
A&~D&~E&~G&~H | A&E&~F&G&~H | A&D&~F&G&~H | A&D&E&~F&~H | A&D&E&~F&G | A&C&~E&~F&H | ~A&~B&~D&E&G&~H | ~A&~B&~D&E&F&~H |
~C&D&E&F&G&~H;
endmodule
```

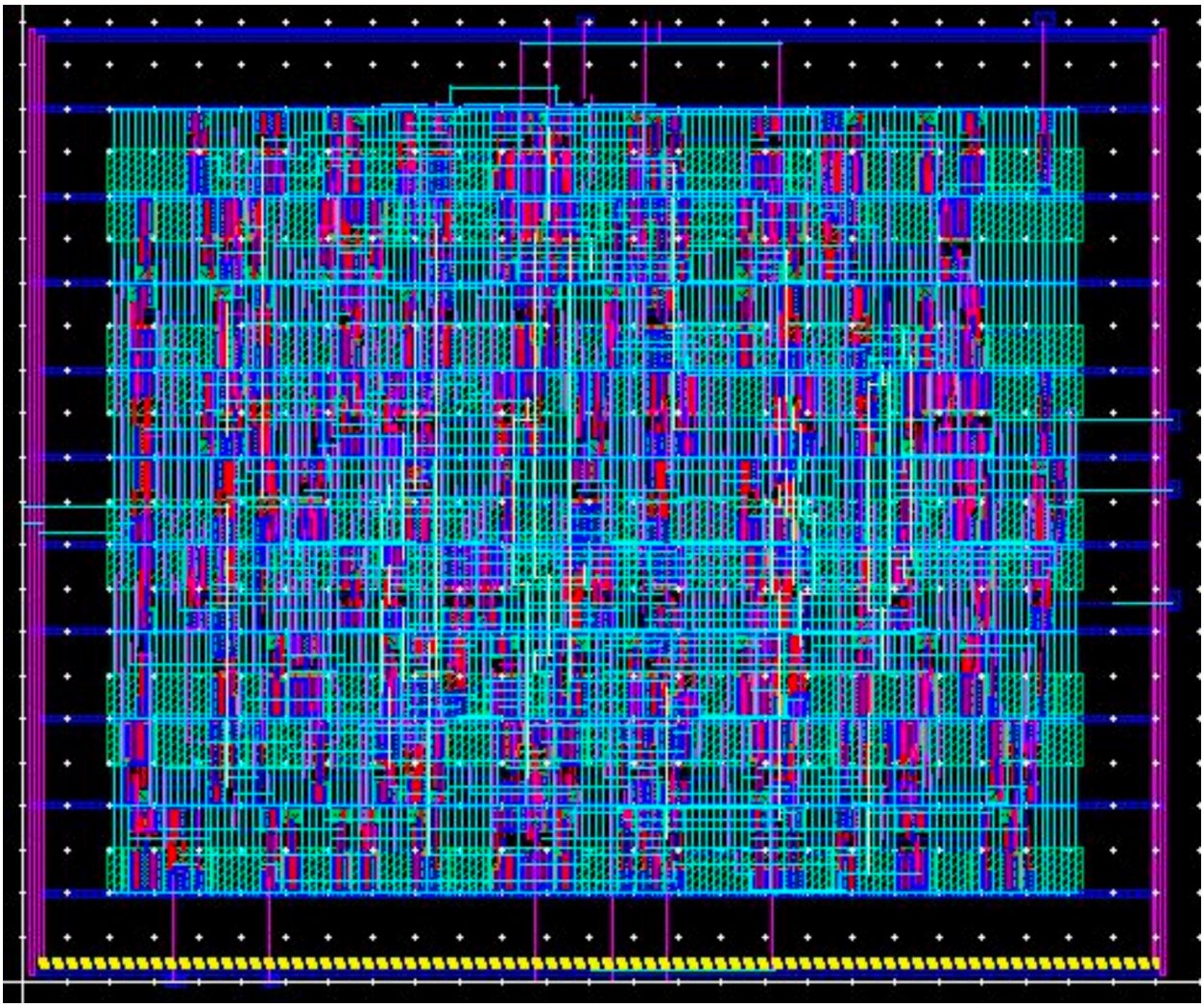
Verilog Synthesis with RTL Compilation

With the Verilog code written and verified for correct output, I next had to design my circuit layout using RTL. To begin, I set up all of the files synthesized by using the terminal. The following topics were the basic flow of the design:

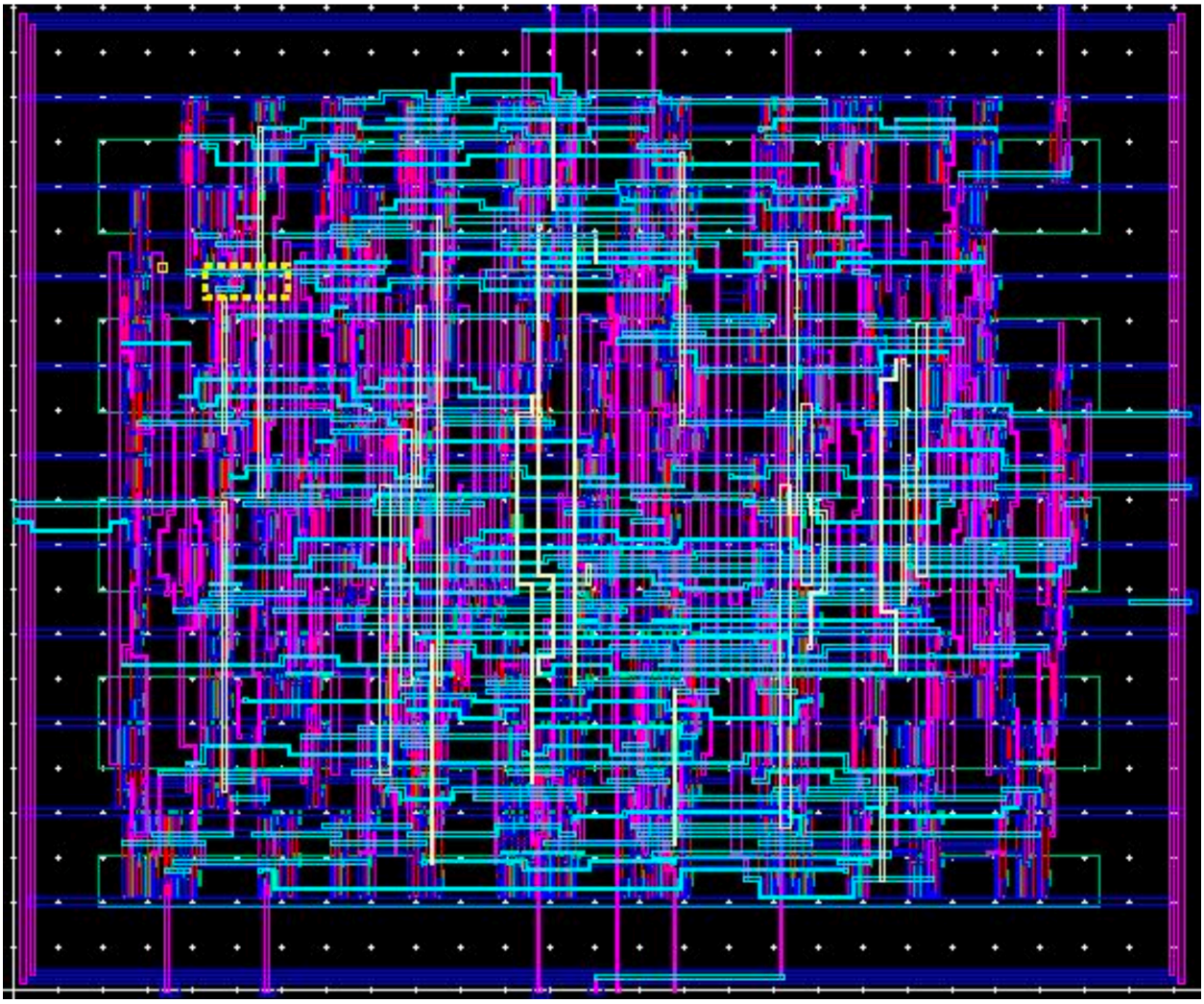
1. Design in HDL (Verilog file)
2. RTL Compiler (Verilog file --> Synthesized Verilog file)
3. Encounter (Synthesized Verilog file --> Layout)
4. Cadence
    a. Layout Import (Encounter --> CIW Import Stream)
    b. Netlist Import (Synthesized Verilog file --> Verilog Import)
5. LVS Verification

I loaded the Verilog file for the Combinational Circuit into the RTL Compiler, and eventually synthesized the circuit. The following shows the layout uploaded into cadence.

THE LAYOUT



EXTRACTED LAYOUT



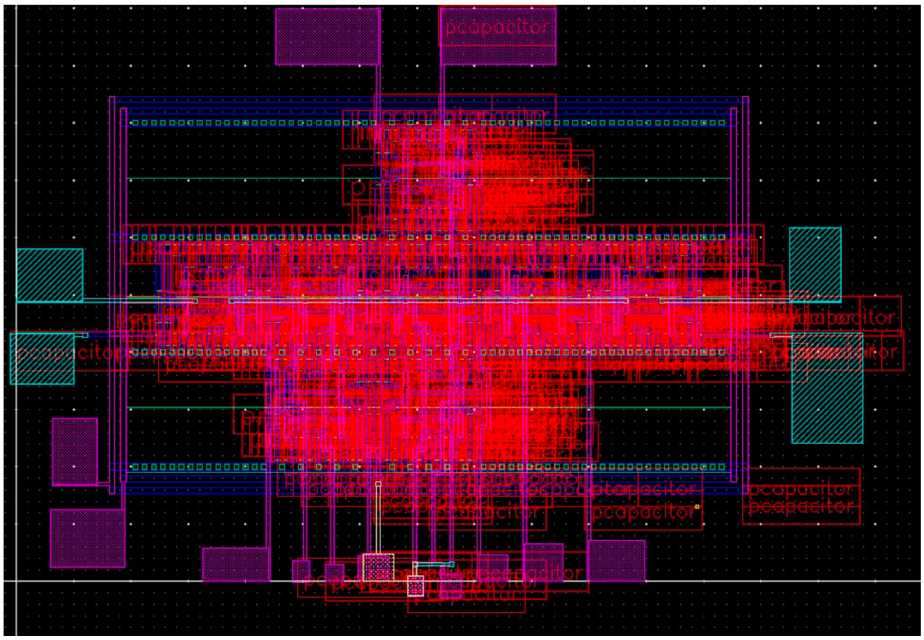
Register:

I created a digital 8-bit register to store in the binary-coded temperature value. This way the 7-segment doesn't have to count up with the counter, but I will pipe the entire value to the register so the display looks smoother. I designed the Enable to be trigger on the negative edge of the input, I will use my comparator output as a 2nd clock for the register. I felt that this will be the smarter way to implement this as my comparator going low indicates the end of my ADC computation.

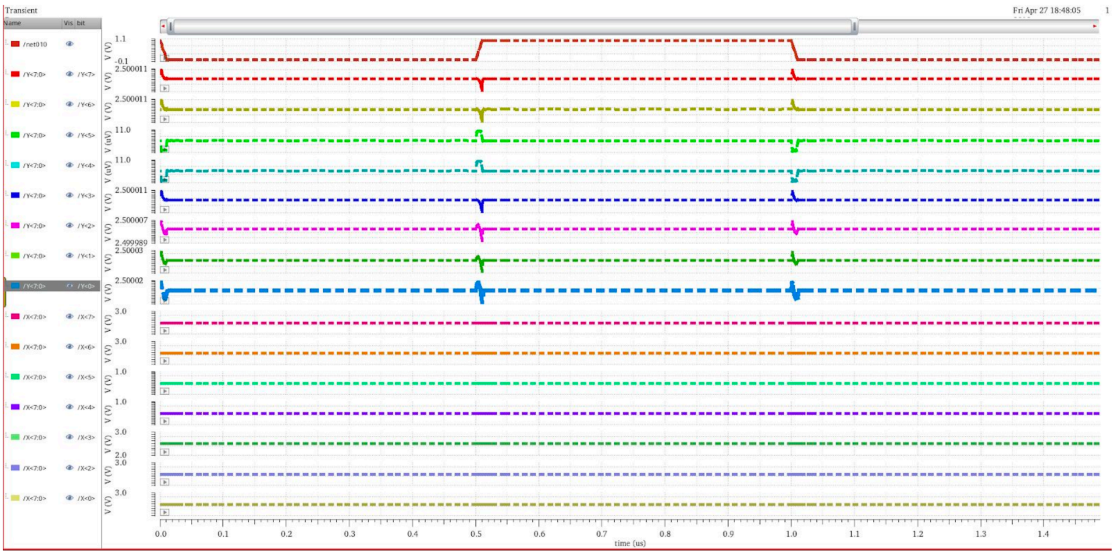
CODE:

```
module register_8(En,X,Y);
  input En;
  input [7:0] X;
  output [7:0] Y;
  reg[7:0] Y = 8'b00000000;

  always @ (negedge En)
    begin
    Y <= X ;
     end

endmodule
```

ANALOG-EXTRACTED:



POST-LAYOUT TEST:



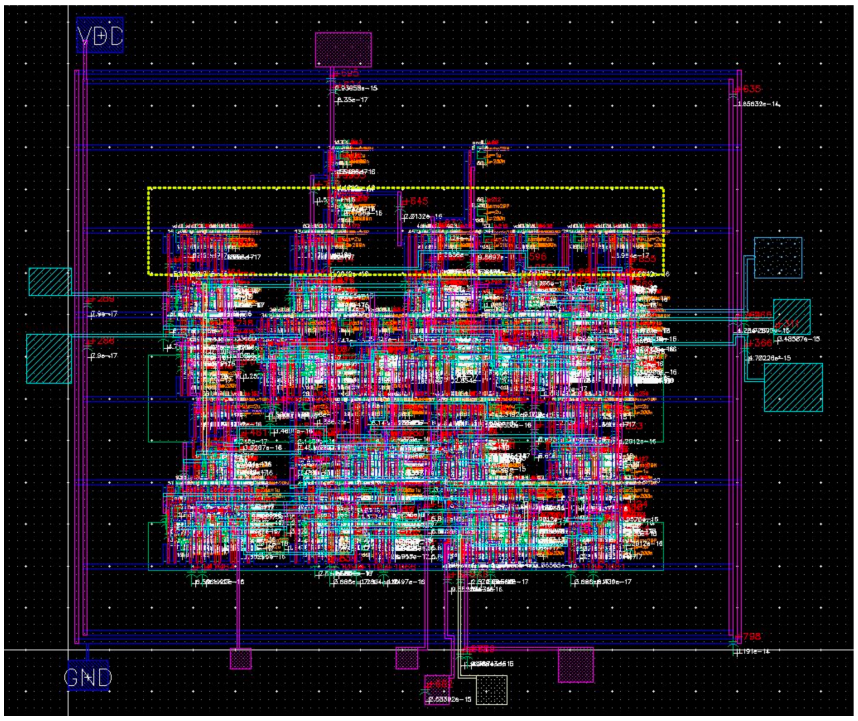
BDC:

I used 3 BCD converters - one for each digits - to convert binary number into BCD number. This circuit was realized digitally; I did not go with double dapper method as my friends advised me that implementing that digitally would be difficult.

CODE and LAYOUT:

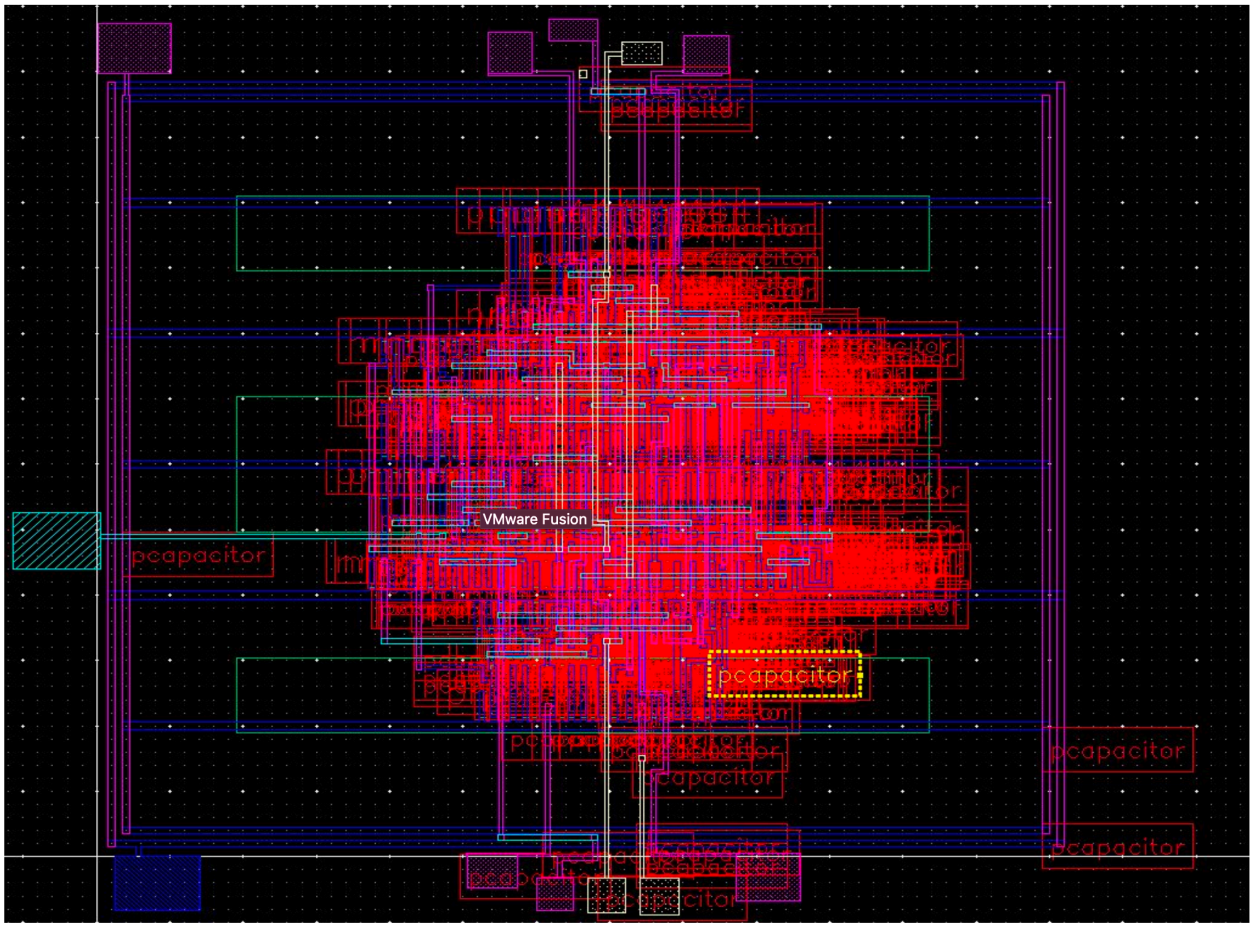
First digit:

```
module first_digit(A,B,C,D,E,F,G,Y3,Y2,Y1,Y0);
  input A,B,C,D,E,F,G;
  output Y3,Y2,Y1,Y0;


  assign Y0 = G | ~A & ~B & ~C & ~D & E & F | ~A & ~B & C & ~D & ~E & ~F | ~A & ~B & C & D & E &
~F | ~A & B & ~C & ~D & E & ~F | ~A & B & ~C & D & E & F | ~A & B & C & D & ~E & ~F | A & ~B & ~C
& ~D & ~E & F | A & ~B & ~C & D & E & ~F | A & ~B & C & ~D & E & F | A & B & ~C & ~D & ~E & ~F | A
& B & ~C & D & ~E & F | A & B & C & ~D & E & ~F | A & B & C & D & E & F;
  assign Y1 = ~A&~B&~C&~D&~E&F | ~A&~B&~C&~D&F&G | ~B&~C&~D&~E&F&G | ~A&~B&~C&D&E&~F |
~B&~C&D&E&~F&G | ~A&~B&C&~D&E&F | ~B&C&~D&E&F&G | ~A&B&~C&~D&~E&~F | ~A&B&~C&~D&~F&G |
B&~C&~D&~E&~F&G | ~A&B&~C&D&~E&F | ~A&B&~C&D&F&G | B&~C&D&~E&F&G | ~A&B&C&~D&E&~F | B&C&~D&E&~F&G
| ~A&B&C&D&E&F | B&C&D&E&F&G | A&~B&~C&D&~E&~F | A&~B&C&~D&~E&F | A&~B&C&D&E&~F | A&B&~C&~D&E&F |
A&B&C&~D&~E&~F | A&B&C&D&~E&F | ~A&~B&C&~D&~E&~F&G | ~A&~B&C&D&~E&F&G | ~A&B&C&D&~E&~F&G;
  assign Y2 = ~A&~B&~C&~D&E | ~A&~C&D&E&F | ~A&~B&C&~E&~F | ~A&~B&C&D&~E | ~A&C&D&~E&~F |
~A&B&~C&E&~F | A&~B&~C&~D&~E | A&~C&D&~E&F | A&~B&C&~D&E | A&C&D&E&F | A&B&~C&~E&~F | A&B&C&E&~F |
~A&B&~C&~D&~E&F | ~A&B&C&~D&E&F | A&~B&~C&D&E&~F | A&B&C&~D&~E&F;
  assign Y3 = ~A&~B&~C&D&~E&~F | ~A&~B&C&~D&~E&F | ~A&~B&C&D&E&~F | ~A&B&~C&~D&E&F |
~A&B&C&~D&~E&~F | ~A&B&C&D&~E&F | A&~B&~C&~D&E&~F | A&~B&~C&D&E&F | A&~B&C&D&~E&~F |
A&B&~C&~D&~E&F | A&B&~C&D&E&~F | A&B&C&~D&E&F;

endmodule
```



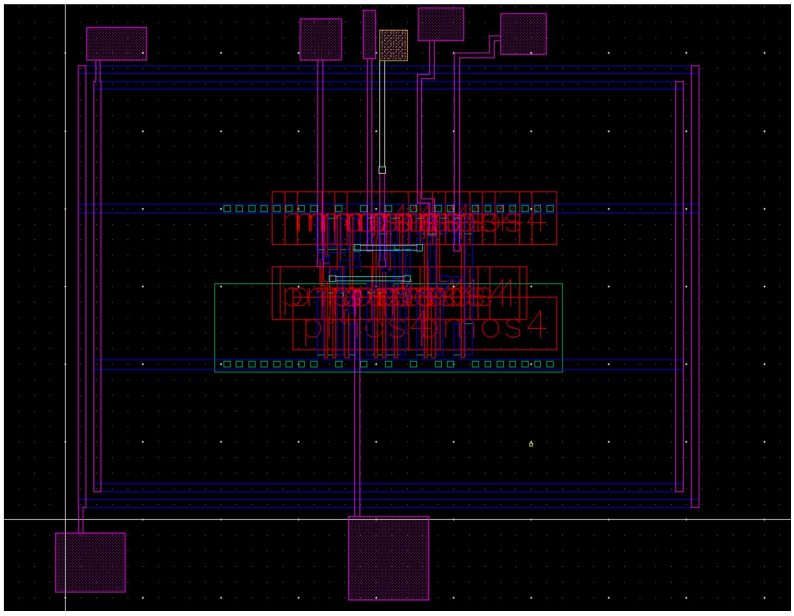
Second digit:

```
module second_digit(A,B,C,D,E,F,G,Y7,Y6,Y5,Y4);
  input A,B,C,D,E,F,G;
  output Y7,Y6,Y5,Y4;

  assign Y4 = ~B&~C&D&F | ~B&~C&D&E | ~B&D&E&F | ~A&B&~C&~D | ~A&B&~D&F | B&C&~D&E | A&~B&~C&D |
A&~B&D&F | A&~B&D&E | A&B&~D&~E | ~A&~B&C&~D&~E | ~A&B&C&D&~E | A&~B&~C&E&F | A&~C&D&E&F;
  assign Y5 = A&~B&~C | ~A&~B&C&E | ~A&~B&C&D | ~A&C&D&E | ~A&B&~C&~D | A&B&C&D;
  assign Y6 = ~A&B&D | ~A&B&C | A&~B&~C;
  assign Y7 = A&~B&C | A&B&~C&~D&~E;
endmodule
```



Third Digit:

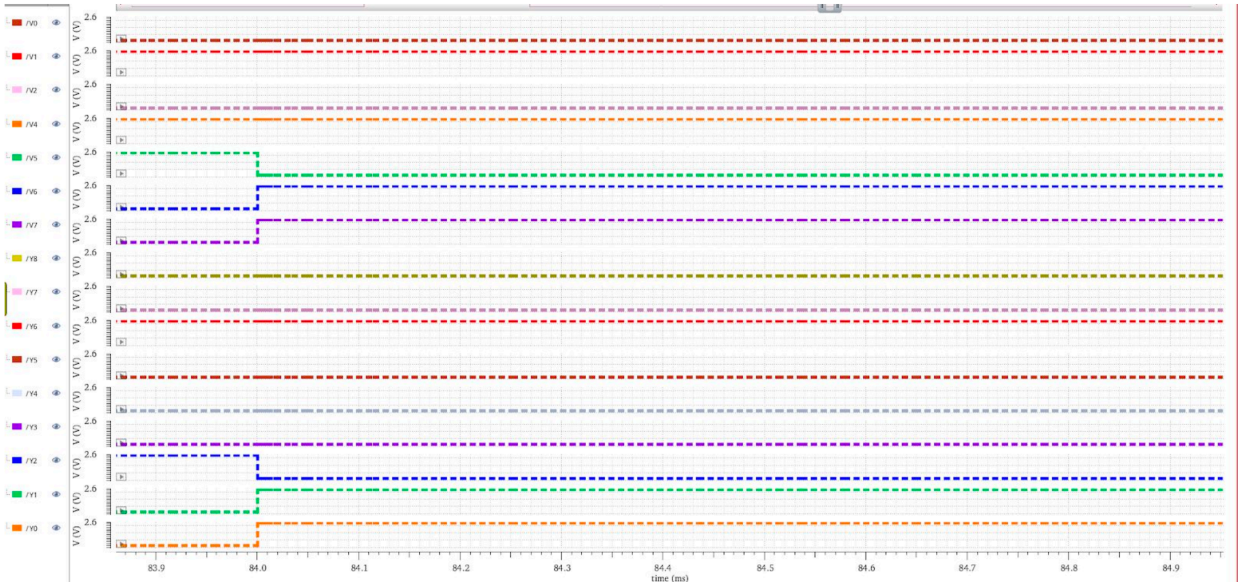
```
module third_digit(A,B,C,D,E,F,G, Y8);
  input A,B,C,D,E,F,G;
  output Y8;

  assign Y8 = A&B&E | A&B&D | A&B&C;
endmodule
```



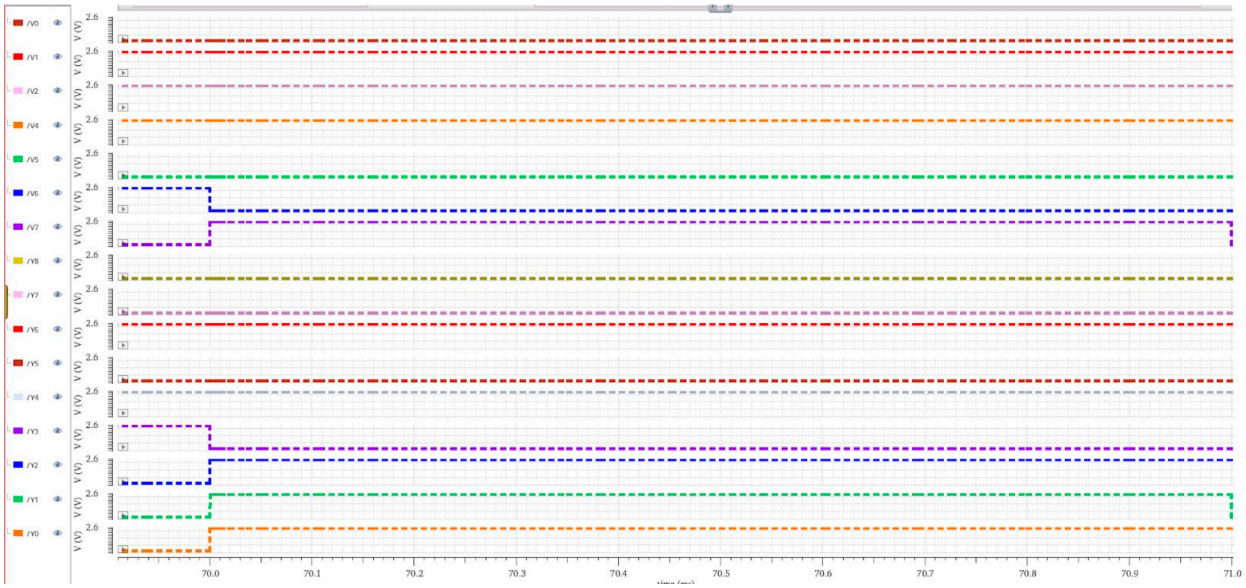
Test:

Below I tested my BCD circuits with 2 different numbers: 43 and 57. It works as expected

43:



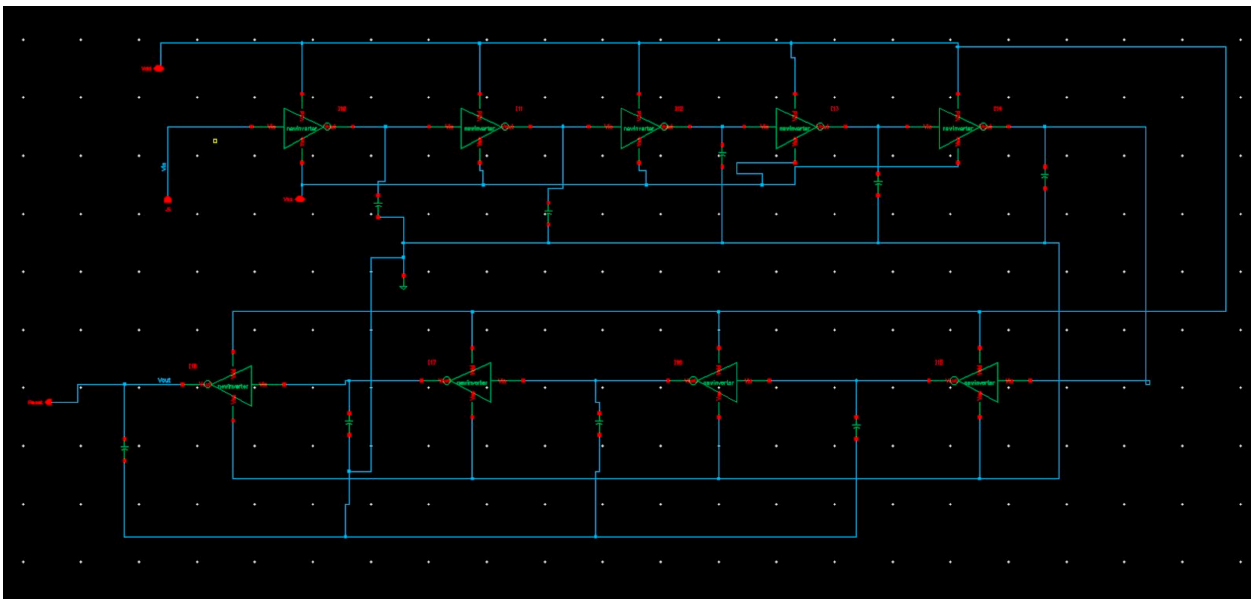
57:



Delay:

The delay is used to reset all the circuits to initial value of zero. This will include the counter and the integrator for this design.The delay will be made using multiple of inverter. For my purpose a 2ns of delay or more to reset all the circuits will be enough.

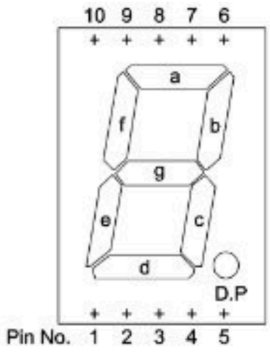
SCHEMATIC



7 Segment Display:

For my Output I will use 4 7 Seg displays to display the temperature. 3 of them will be connected to the BCDs and the 4th 7-seg will only one connected that will turn on the g-section to display a negative sign when necessarily. I went with the LF-3011A which is Common Cathode so I to illuminate the individual segments by applying a HIGH logica to it. The 7-seg works with a voltage range that include my 2.5V for HIGH logic.

●Pin assignments



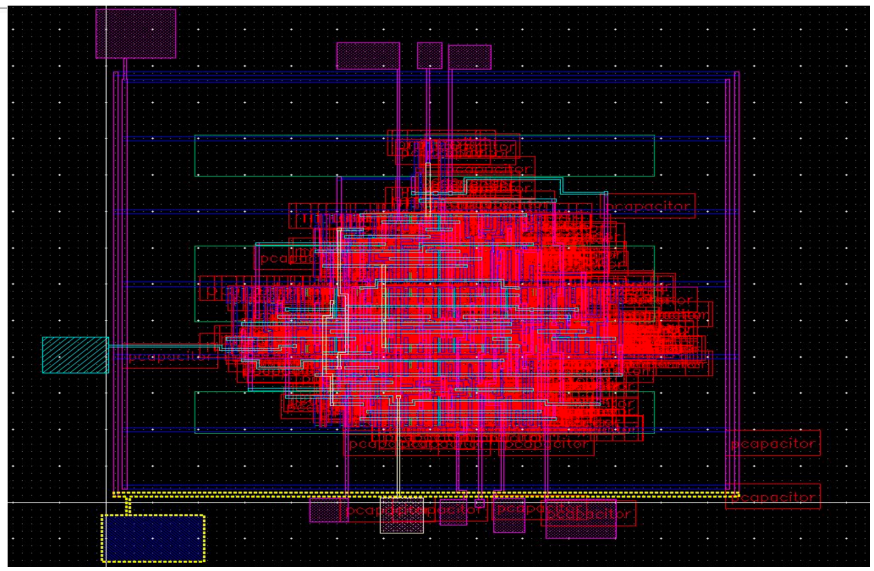
●**Electrical and optical characteristics** (T_a = 25℃)

Parameter	Symbol	Conditions	Red		Green		Unit
			Typ.	Max.	Typ.	Max.	
Forward voltage	V_F	I_F=10mA	2.0	2.8	2.1	2.8	V
Reverse current	I_R	V_R=3V	-	100	-	100	μA
Peak wavelength	λ_p	I_F=10mA	650	-	563	-	nm
Spectral line halfwidth	$\Delta\lambda$	I_F=10mA	40	-	40	-	nm

※ The products are not radiations resistant

Code and Layout:

```
module seven_seg(X3,X2,X1,X0,seg);
  input X3,X2,X1,X0;
  output [6:0] seg;
  reg [6:0] seg;

  always @ (X3 or X2 or X1 or X0)
  begin
      case ({X3, X2, X1, X0})
        4'b0000: seg = 7'b1111110;
        4'b0001: seg = 7'b0110000;
        4'b0010: seg = 7'b1101101;
        4'b0011: seg = 7'b1111001;
        4'b0100: seg = 7'b0110011;
        4'b0101: seg = 7'b1011011;
        4'b0110: seg = 7'b1011111;
        4'b0111: seg = 7'b1110000;
        4'b1000: seg = 7'b1111111;
        4'b1001: seg = 7'b1111011;
      endcase
    end
  endmodule
```



Seg<6> will be connected to pin A, Seg<5> will be connected to pin B, and so on.

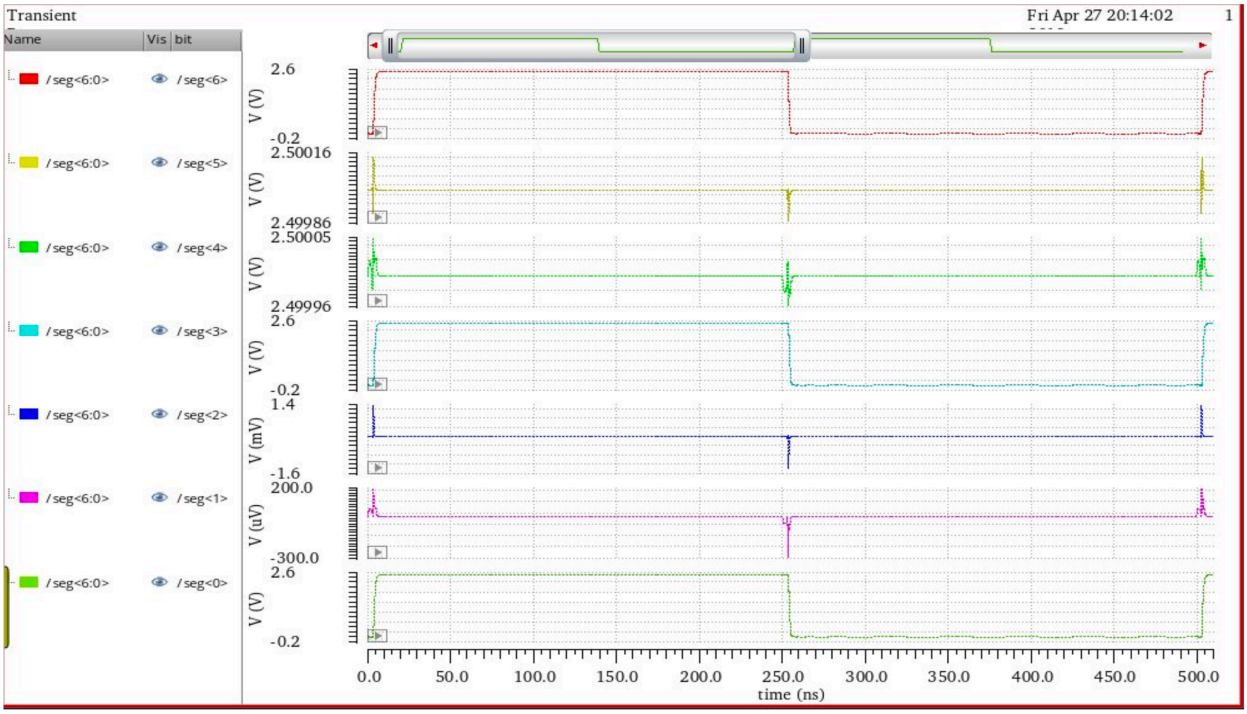
Ahmed Abuhjar
www.aabuhjar.com

EE 330 Integrated Circuits
Dr. Randy Geiger

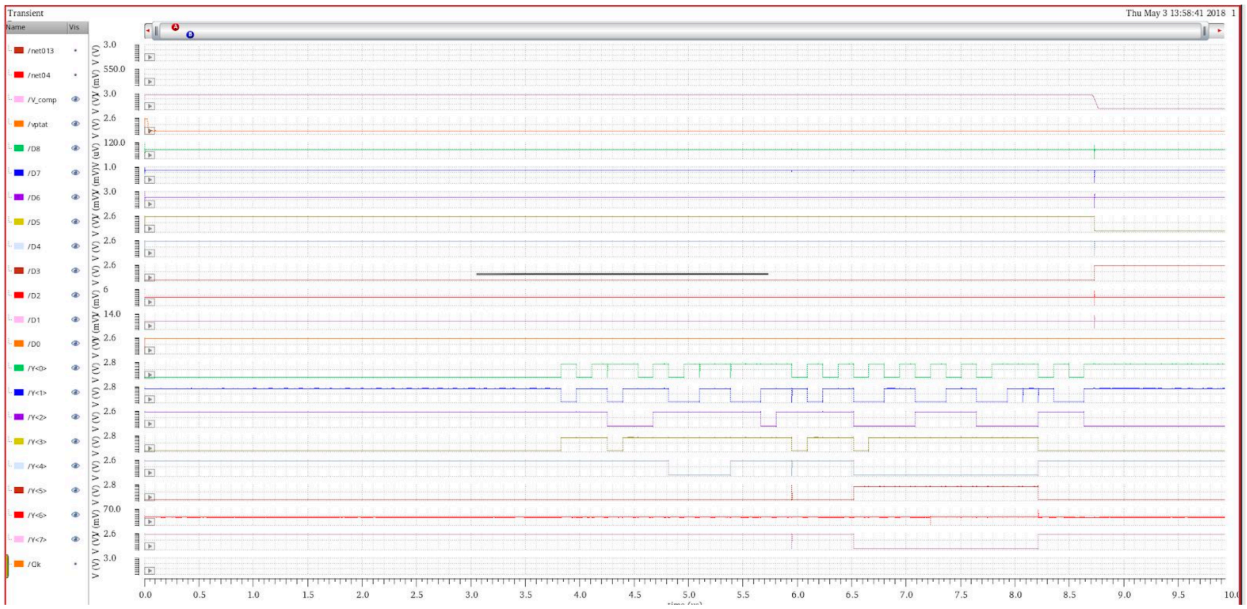
May 2018

POST-LAYOUT TEST:

I tested my circuit with 2 inputs : 1 and 3. To display 3, seg<6,5,4,3,0> must be HIGH to illuminate pins a,b,c,d, and g to display 3. Likewise, seg<5,4> must be HIGH to illuminate pins b and c to display a 1; this can be seen in my test results:



A demonstration for the entire design was tested for -20C and is shown below. The output was successfully determined to be satisfied.



Conclusion:

Engaging in this project was a highly enriching experience as it allowed me to conceptualize and design a circuit with versatile applications across various industries. The journey presented numerous challenges, each requiring dedicated effort to identify and implement effective solutions. Overcoming these obstacles not only resolved immediate issues but also equipped me with valuable problem-solving skills for future endeavors.

This project served as a comprehensive learning opportunity, providing insights into synthesized circuits and Hardware Description Languages within a practical, working context. While delving into the analysis phase, I encountered several ambiguities that were not explicitly addressed in the introduction. To fulfill the assigned task, I developed my own interpretations and strategies to tackle these uncertainties. Ultimately, the design successfully fulfilled the stipulated functions to a significant extent.

However, certain uncertainties persisted in the output, leading to temperature discrepancies of 1-4 degrees. This was primarily attributed to the non-100% linearity of the integrator's (voltage vs. time) graph, introducing a level of uncertainty. Additionally, the dependence of the clock time period on temperature emerged as a significant factor contributing to these uncertainties.

In the realm of synthesized circuits, the Verilog simulation closely mirrored the theoretical results, and the corresponding layout design passed the Design Rule Check (DRC). While some circuit layouts proved intricate and time constraints prevented their inclusion in this report, the overall satisfaction derived from the project was profound.

In conclusion, this project has been immensely enjoyable, offering valuable insights into the practical aspects of electrical engineering. It has provided a glimpse into the complexities and problem-solving demands that characterize the field, contributing significantly to my understanding and appreciation of the discipline.