High-Resolution ADC Using Delta-Sigma Architectures

sddec18-20
http://sddec18-20.sd.ece.iastate.edu
Advisor: Professor Randall Geiger
Clients: Professor Randall Geiger and Professor Degang Chen

Problem and Intended Use

- Problem
  - Design a Delta-Sigma Analog to Digital Converter that reads an analog signal from an on-chip temperature sensor and converts it into a digital output
- Used in many circuits
  - Physical data is normally characterized as an analog signal, converting it to a digital signal allows the signal to be easily manipulated
  - Normally at the front end of most digital circuits

Analog to Digital Converters

- ADCs allow electronics to interact with the analog world around us
- Nyquist Rate ADCs
  - Samples the input signal at the Nyquist Rate
  - Successive approximation
  - Flash
- Oversampling ADCs
  - Samples the input signal at a frequency significantly higher that the Nyquist rate
  - Delta Sigma

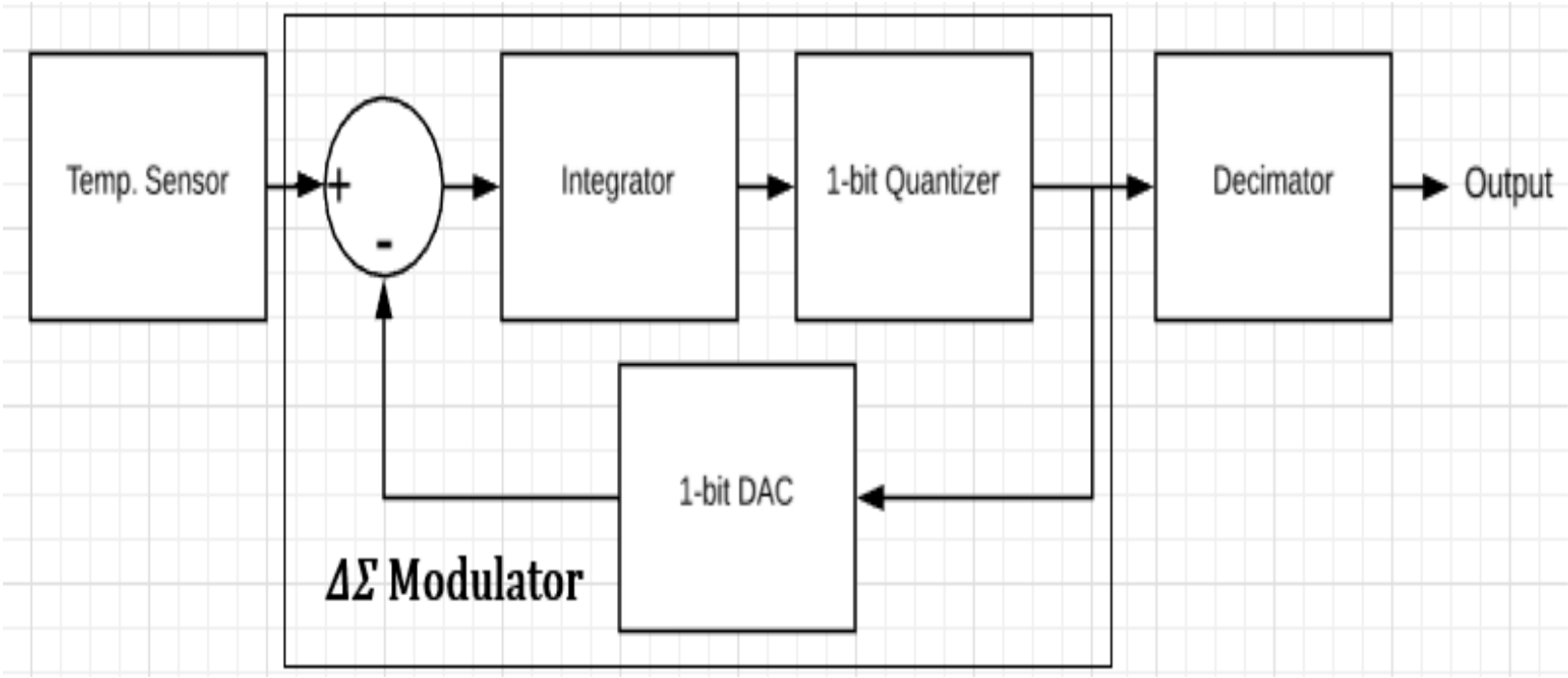
Why Delta Sigma

- Oversampling
  - Higher oversampling ratio leads to higher signal to noise ratio
  - Higher oversampling ratio leads to higher resolution
- Linear System
  - Linearity depends on a 1-bit DAC

Requirements and User Interface

- Requirements
  - Functional
    - Temperature range of sensor from 10°C to 60°C
    - New output code every 10ms
    - 10-bit output
    - Designed in TSMC 180nm process
  - Non-functional
    - Must fit inside a die of 4mmX4mm
  - User Interface
    - IC designers integrate them in IC designs, both in the academic world and in industry

Delta-Sigma ADC and On-Chip Temperature Sensor



Operating Environment and Familiarity with tools

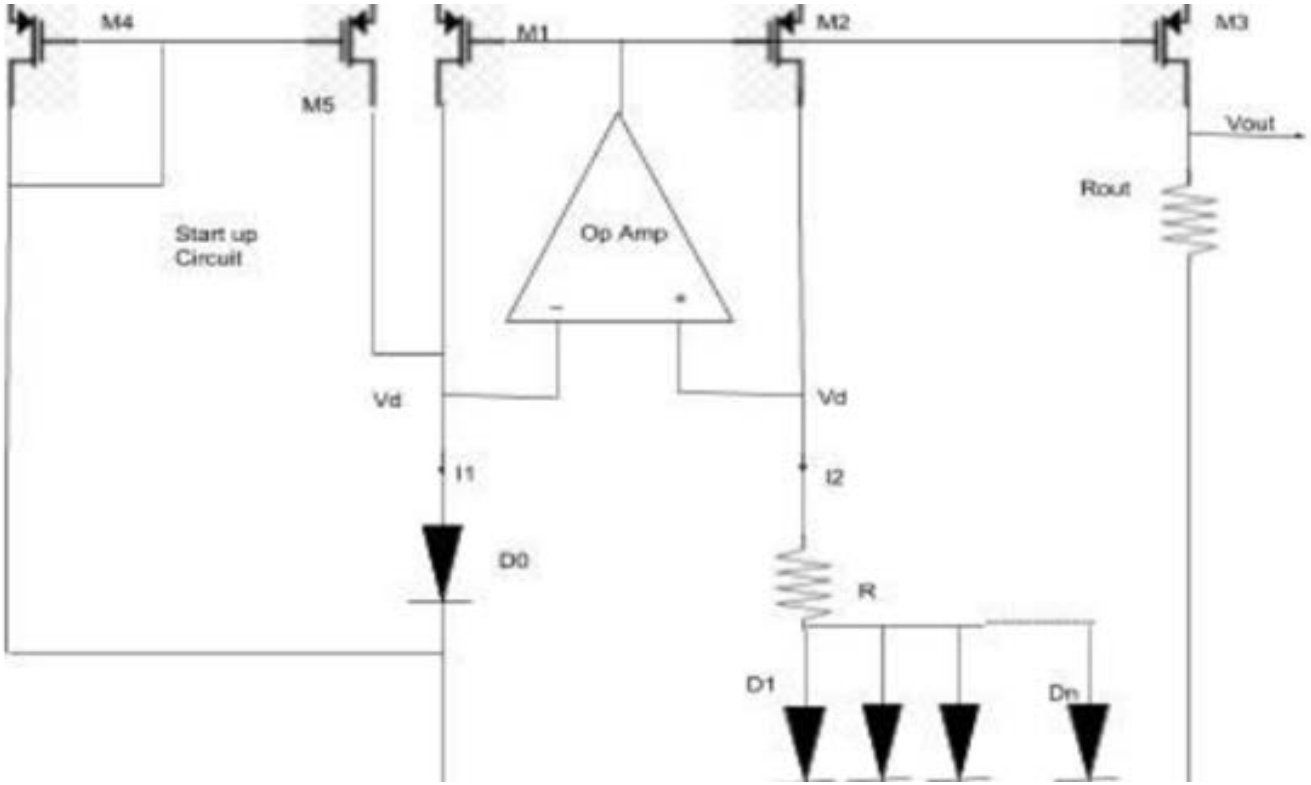
- Operating Environment
  - It can be implemented into any integrated circuit
  - There will be a pin out between the modulator and the temperature sensor, this will allow us to test any analog signal
- Familiarity
  - Simulations performed in Cadence
    - All members of our team have taken classes, and participated in projects, involving integrated circuits and analog VLSI design

Project Management

- Resources
  - Access to:
    - Cadence
    - Fabrication through MOSIS
    - TSMC 180nm Process Design Kit (PDK)
    - Pad Frame compliant with the process
    - Oven
    - Electrolytic bath
    - DAQ (Digital Acquisition) measurement device
  - Work method: divide work into circuits
  - Risks
    - Nobody on team has designed a data converter prior to the project
    - Time and legal constraints involved with fabrication

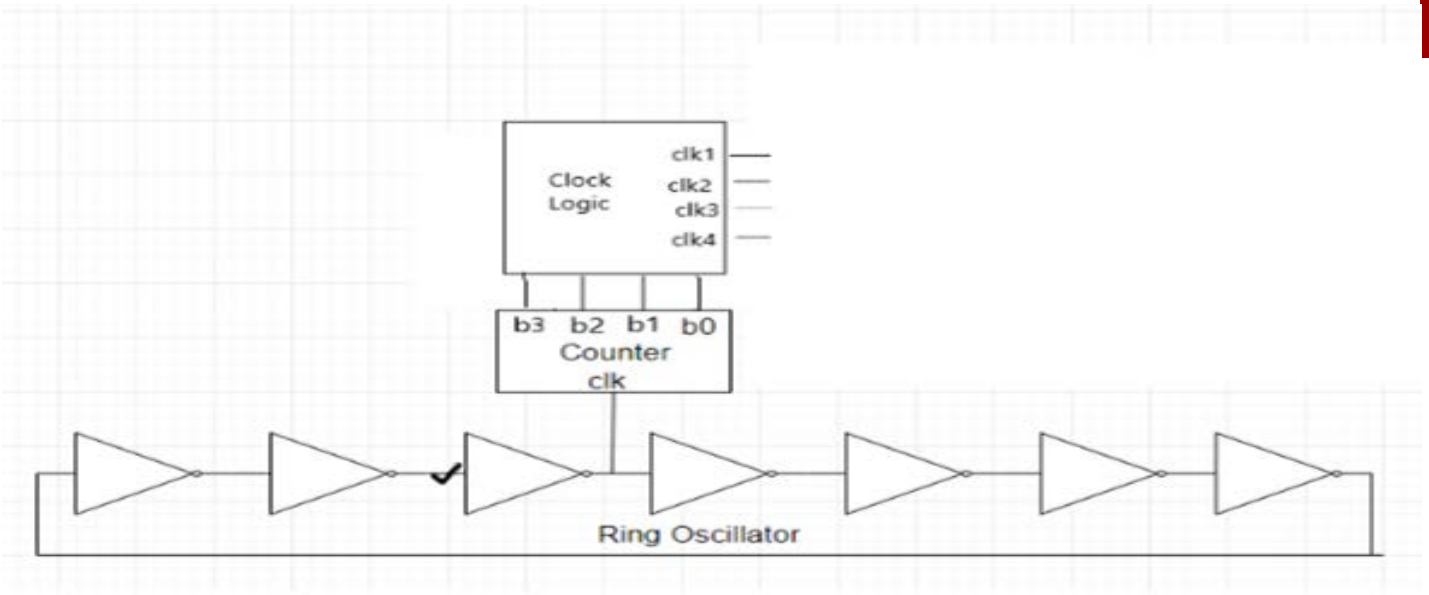
System Design Blocks

Temperature Sensor



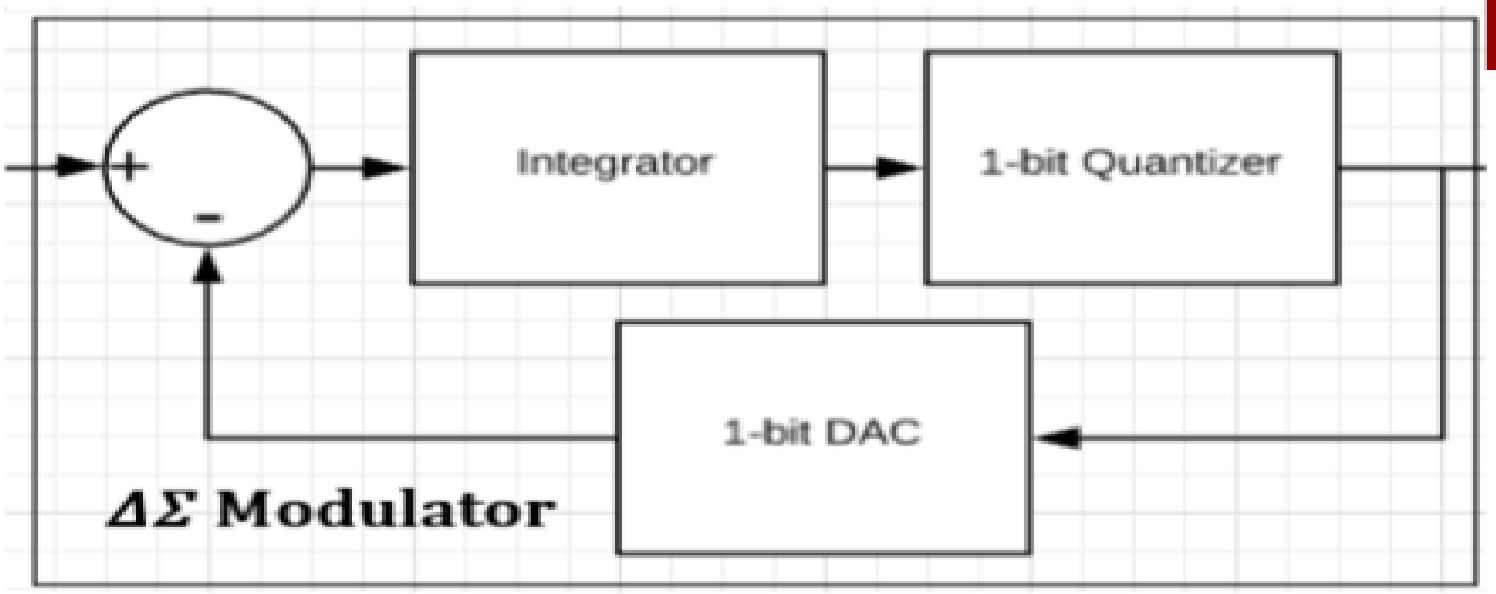
- A diode has a current that is dependent on its temperature
- The sensor takes advantage of this to produce a voltage proportional to the chip's temperature

Clock



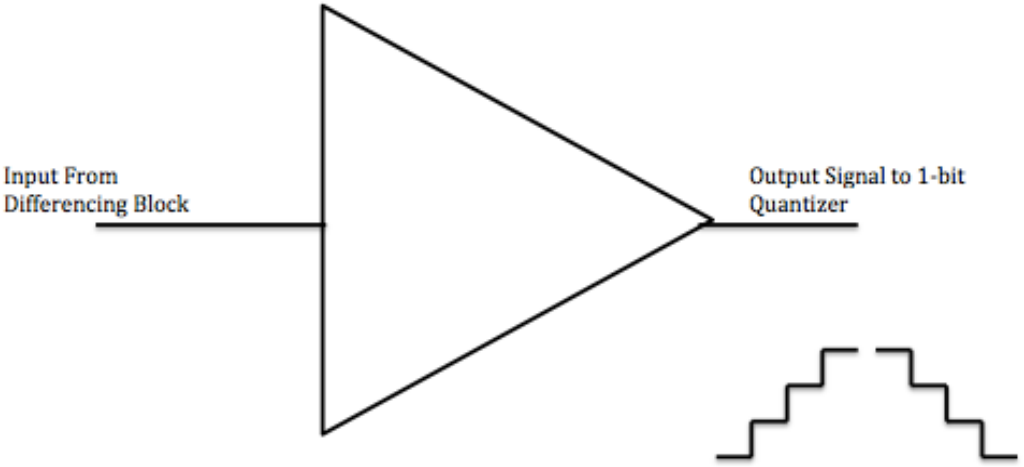
- System clock is produced by a ring oscillator
- This is used to produce 4 clock signals with various phases and duty cycles (nominally 100 KHz)

Modulator



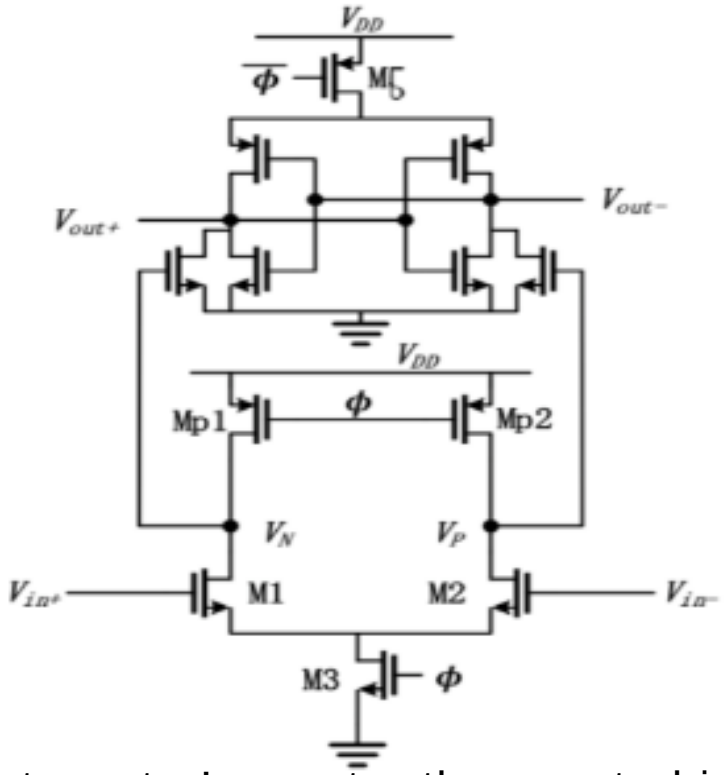
- Converts an input voltage to a binary bit stream
- Bit stream's ratio of ones to zeros is proportional to the input voltage

Integrator



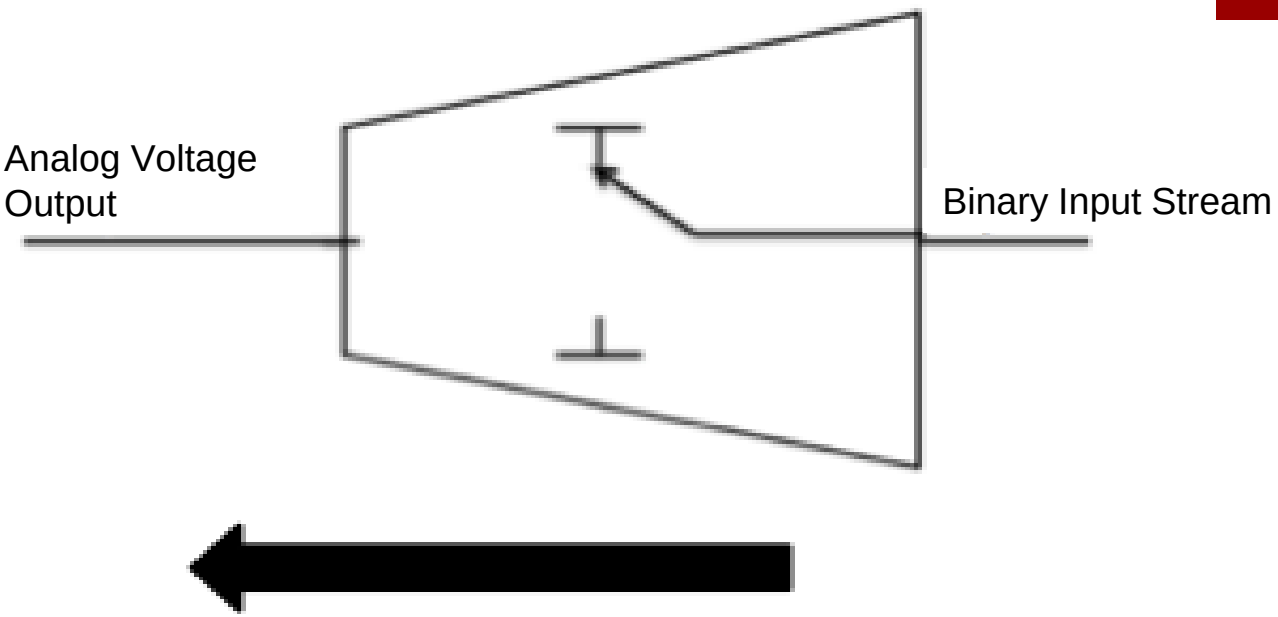
- Generates a saw-tooth wave that progresses in a positive or negative direction depending on the sign of the input voltage
- The slope of this progression depends on the magnitude of the input voltage

Dynamic Comparator



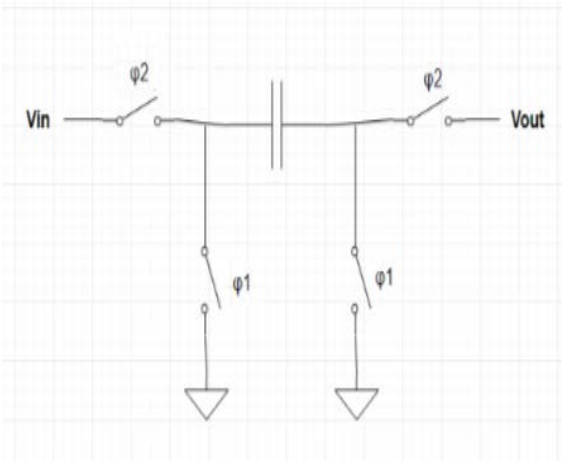
- Converts integrator's saw-tooth wave to binary bit stream by comparing its magnitude with a reference voltage at a clock edge

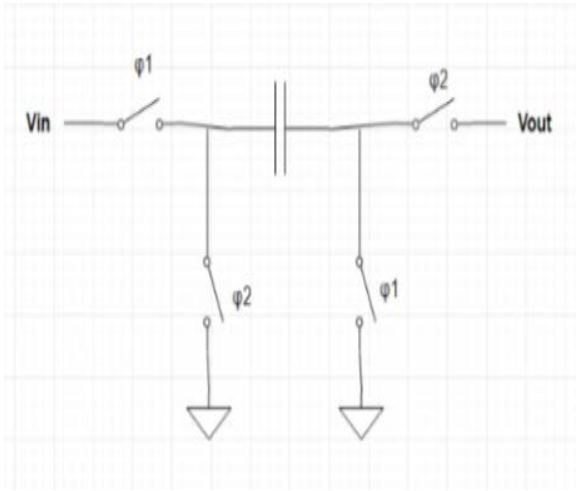
DAC

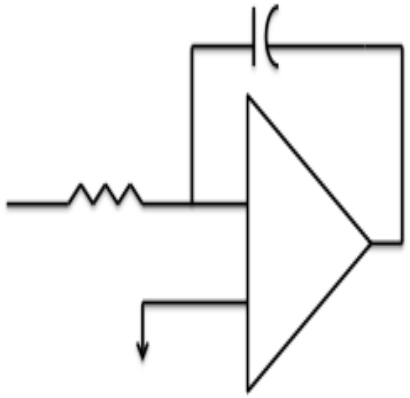


- Converts output bit stream to analog signal applied to integrator to control the direction of the saw-tooth wave's progression

Switched Capacitors





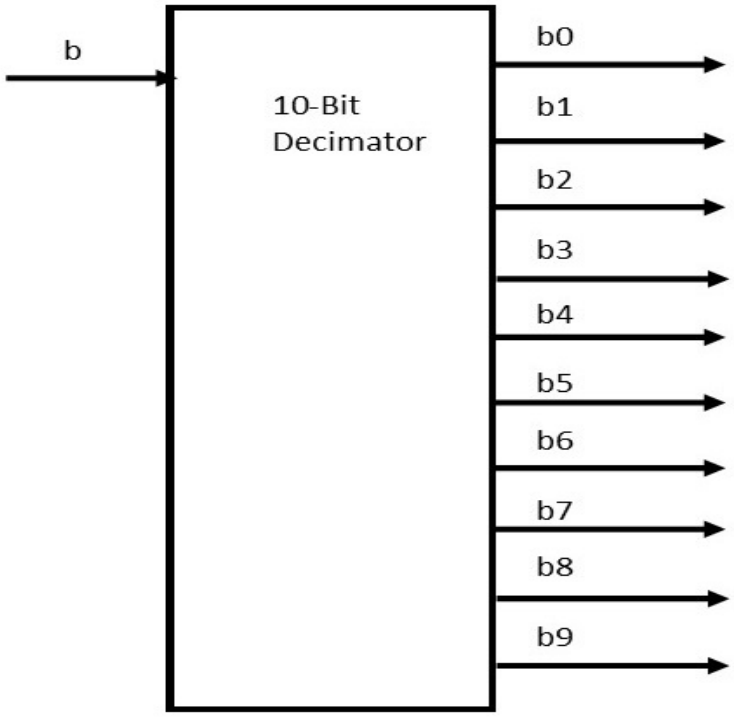


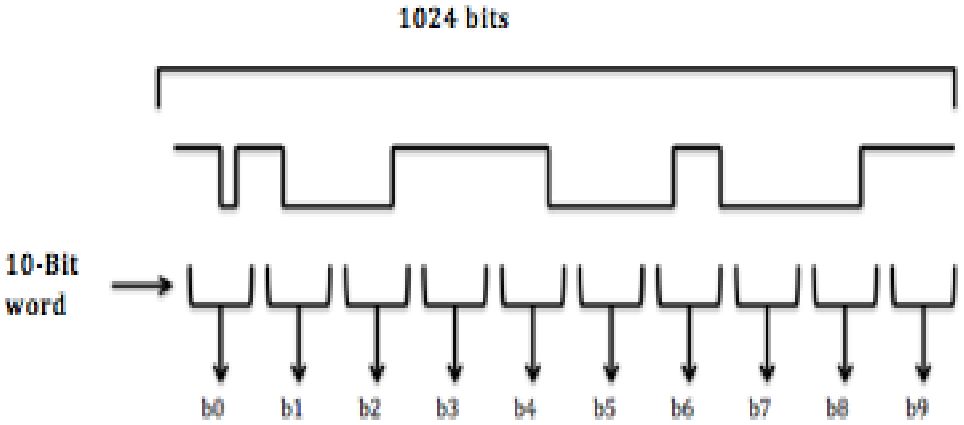
Non-Inverting

Inverting

- Controls current by incrementally transferring charge using a capacitor and clock
- Allows implementation of a large resistor using a small capacitor, saving circuit area

Decimator





- Converts bit stream from modulator's output to 10-bit parallel output codes by summing every 1024 bits

180nm Test Results

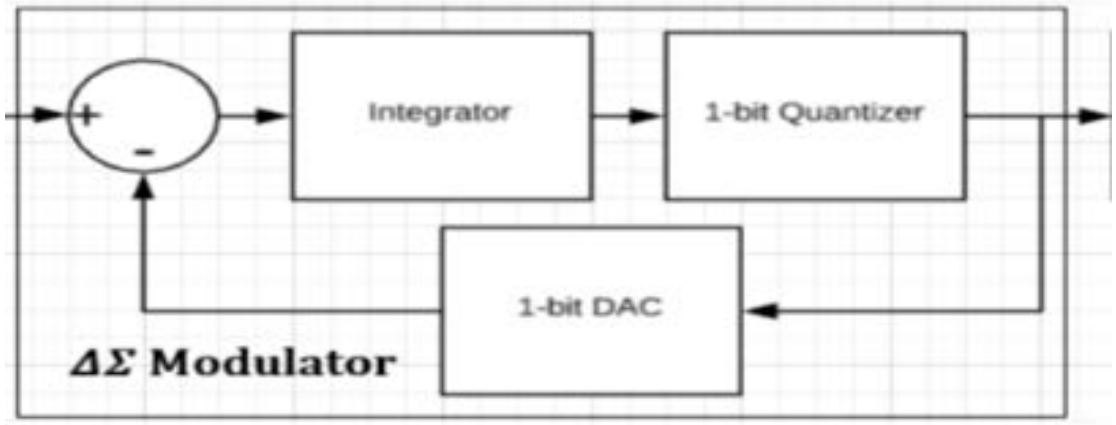
Overall Results

Input Voltage (mV)	First Code	Second Code	Expected Code
730.068359375	3	2	1
752.5	257	256	256
765V	512	512	512
782.5	767	767	768
799.931640625	1023	1023	1023

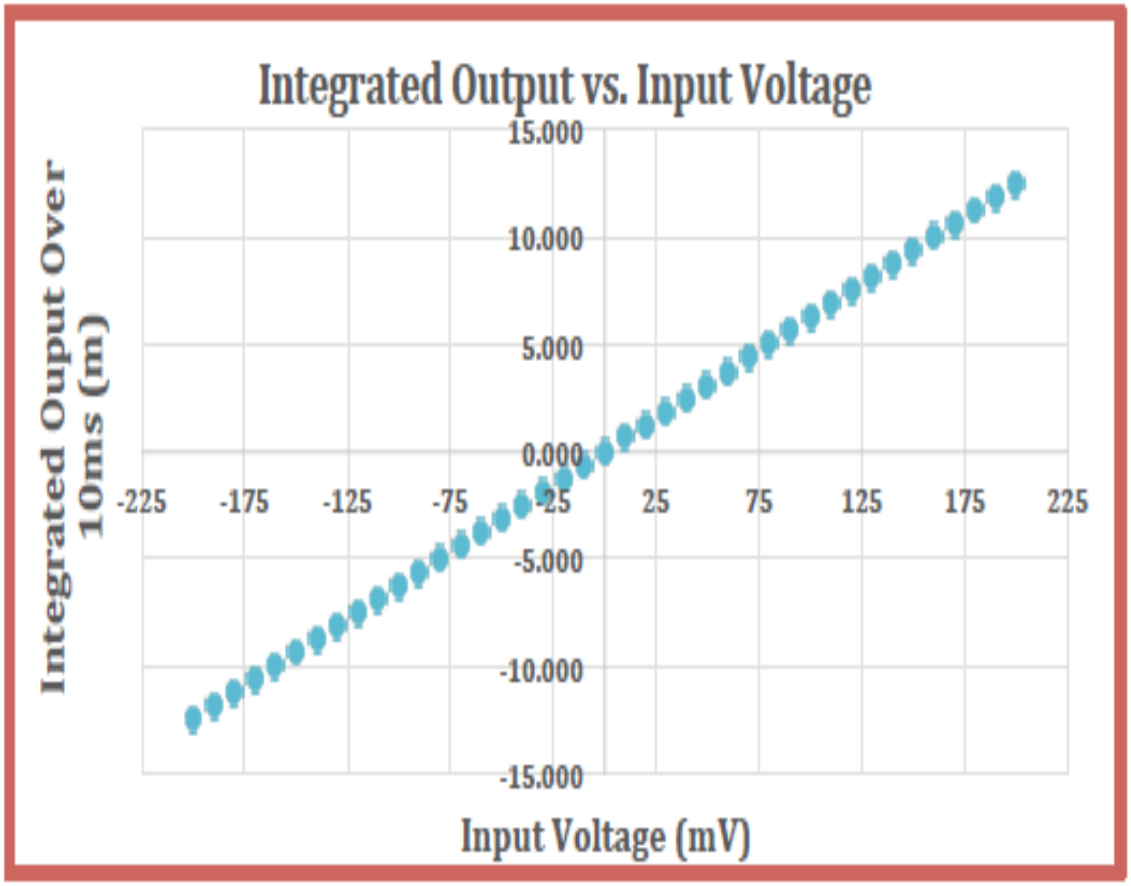
The desired output is a 10-bit code on 10 parallel output pins, whose magnitude is proportional to the proportion of the input values that are high in a 10ms timespan. A new code should be output every 10ms.

UMC 65nm Design

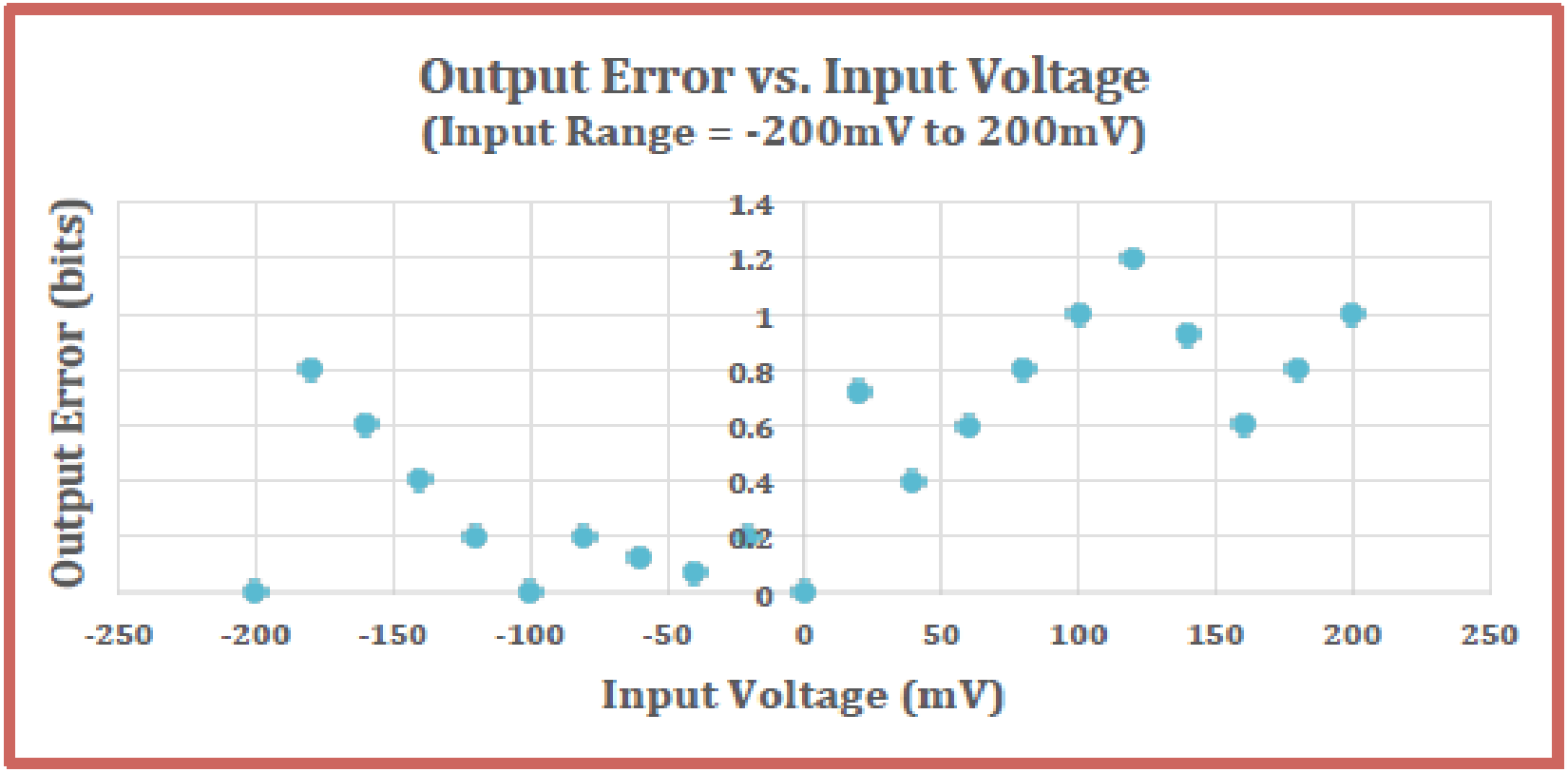
Delta-Sigma ADC 65nm Design



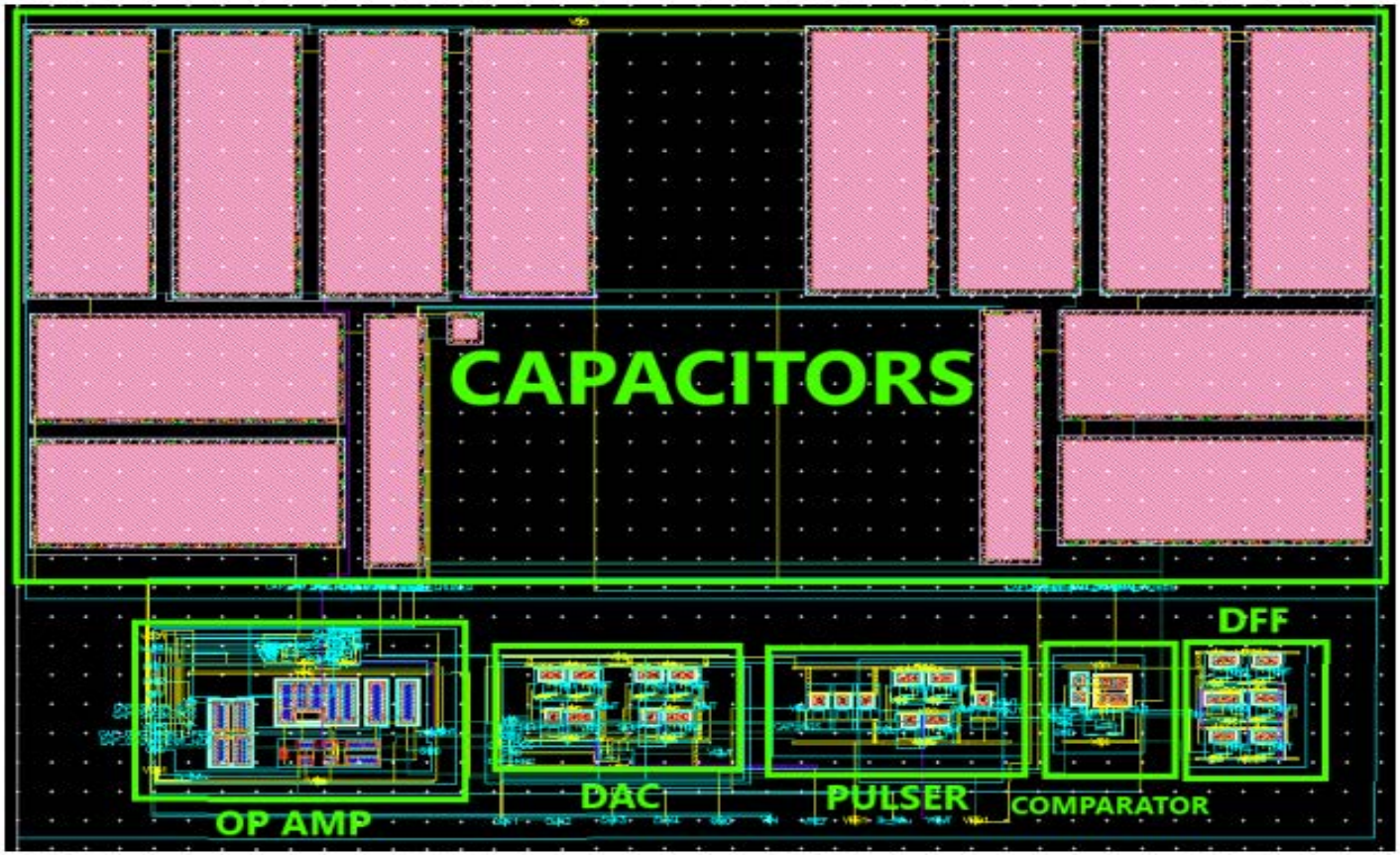
65nm Modulator Results



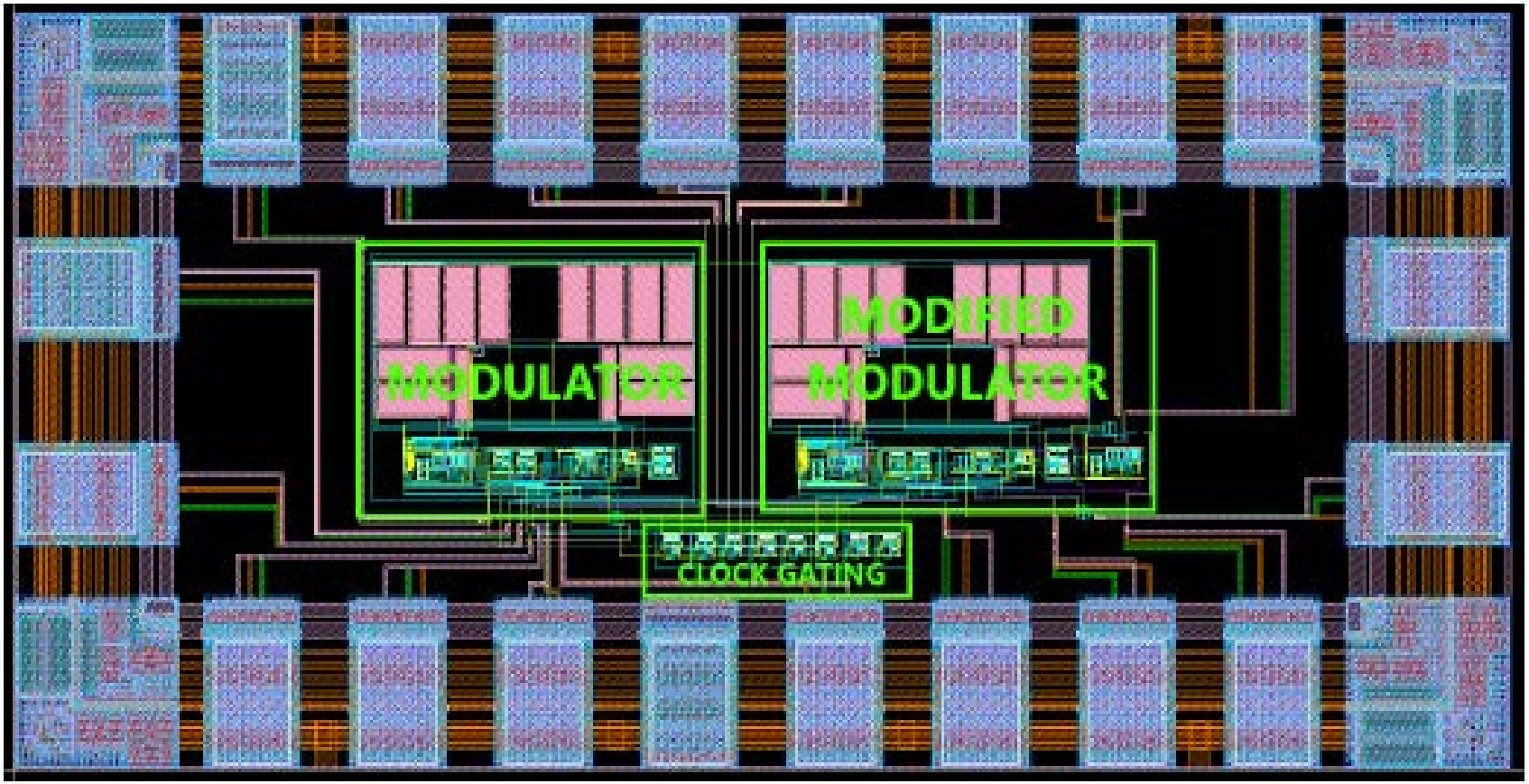
Final Results



65nm Modulator Layout



Final Layout of 65nm Design



Post-Fabrication Testing

- Post-fabrication testing will define the performance and characteristics parameters of the ADC
- Design of a PCB (Printed Circuit Board) has been implemented to take data off-chip and create test points along the modulator
- A DAQ (Data Acquisition) Device from MCC will provide us the ability to acquire data from the ADC and obtain characteristic data

Characteristic Testing

- The SNR (Signal-to-Noise Ratio), SNDR (Signal-to-Noise and Distortion Ratio), and ENOB (Effective Number of Bits) characterizes the performance of the ADC
- These will provide accuracy and characteristic information about our device
- To test these values we will use a Matlab script to post-process in conjunction with our DAQ to acquire data
- The Matlab script also allows for further characterization parameters if need be

Behavioral Testing

- Measurements that exhibit the behavior of the ADC:
  - The time it takes for an output code to arrive given a particular input
  - The rate at which the design is capable of operating
  - Power consumption of the ADC
- These values help us ensure the design meets original requirements

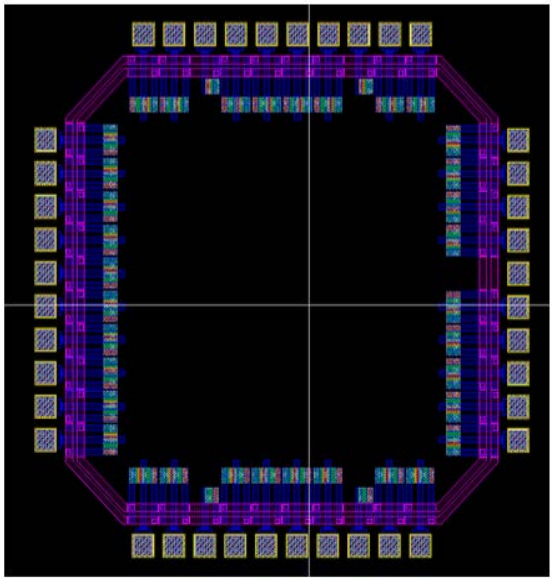
Next Semester and Lessons Learned

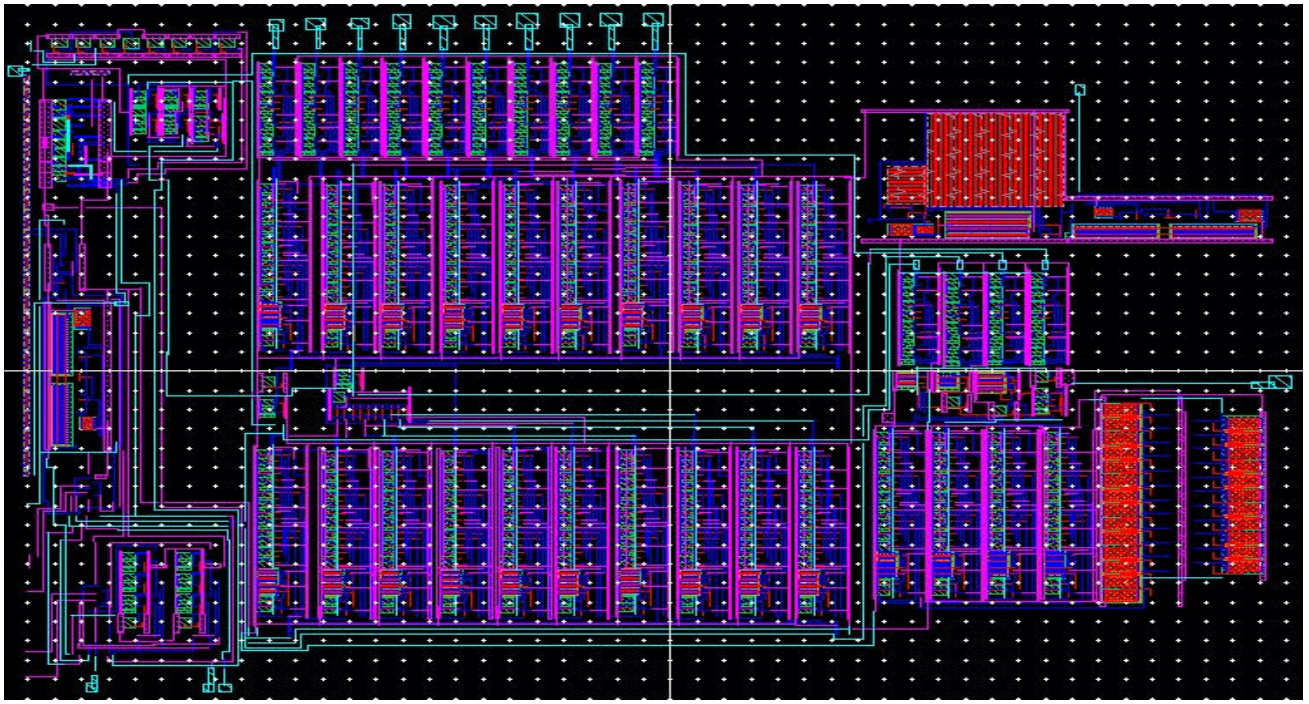
- Two members of the team who will be here next semester will carry out testing of the fabricated chip
- Once the device is returned from fabrication will need to be packaged, currently looking at a ZIF (zero insertion force) package
- A PCB will be used for testing the device with direct test points and proper input signals.
- Transitioning from one process to another is a valuable skill to use in industry

Thank you!

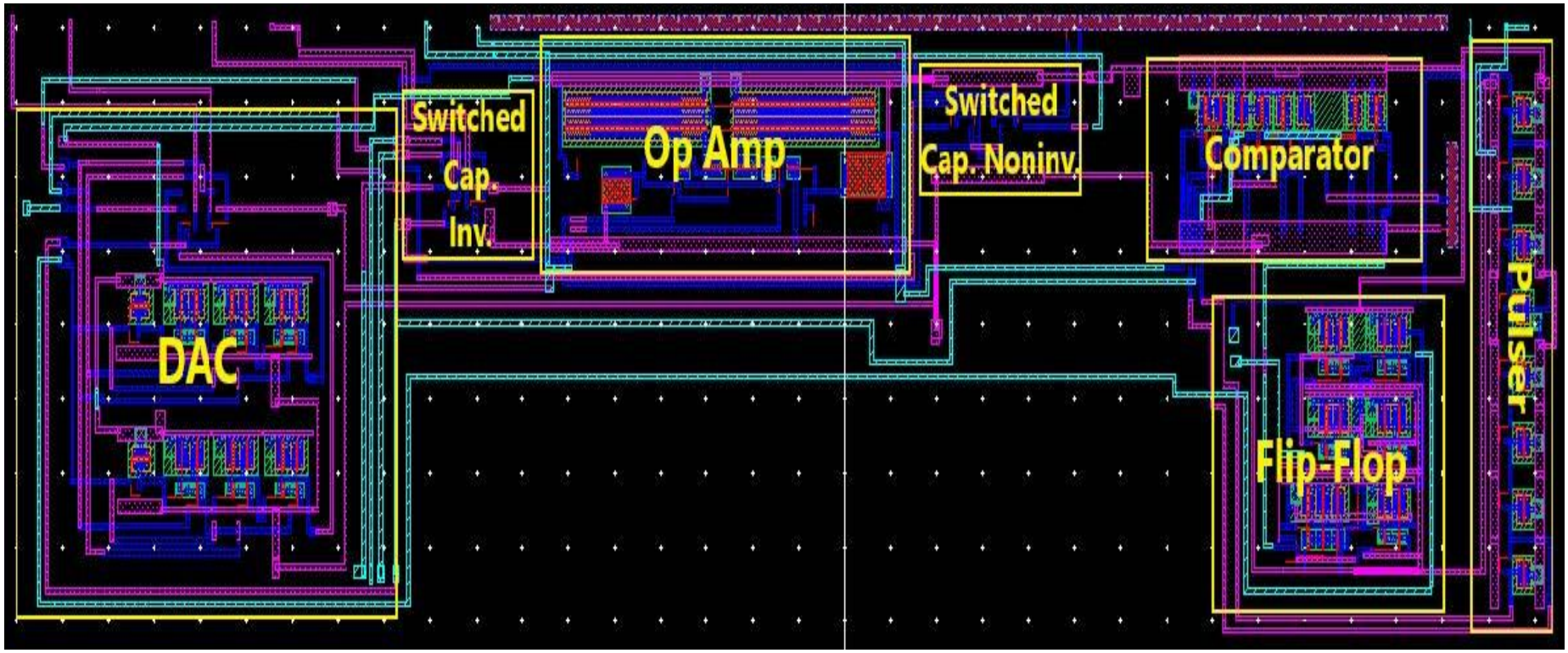
Caroline Alva
Tyler Archer
Caleb Davidson
Mahmoud Gshash
Josh Rolles

Final Layout of 180nm Design

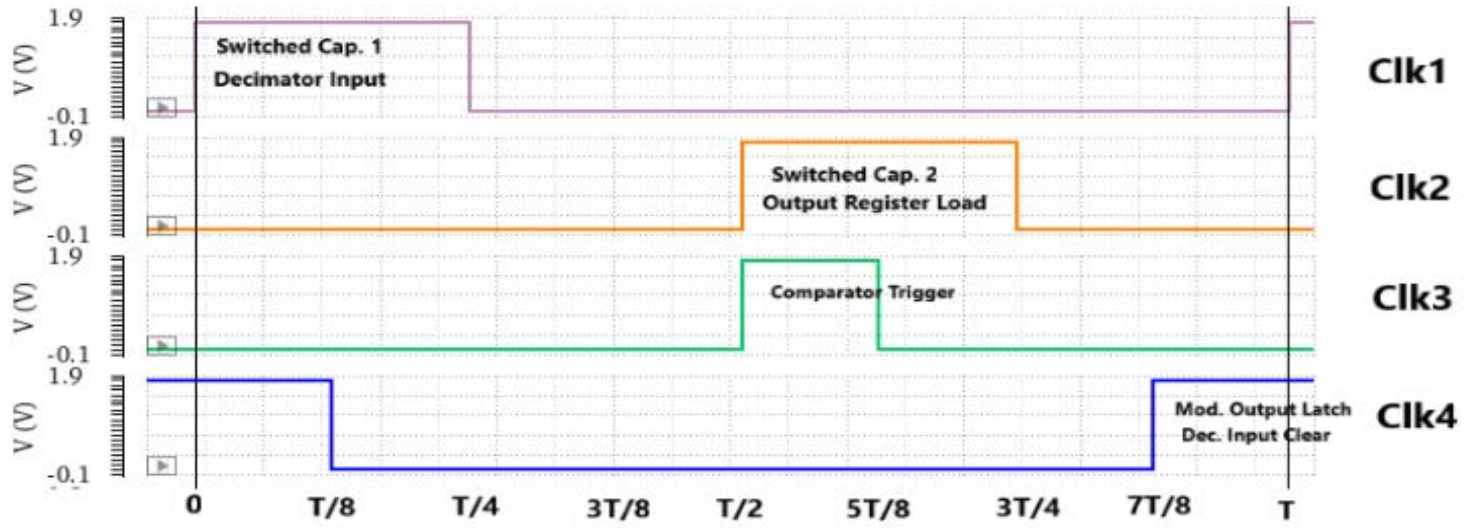




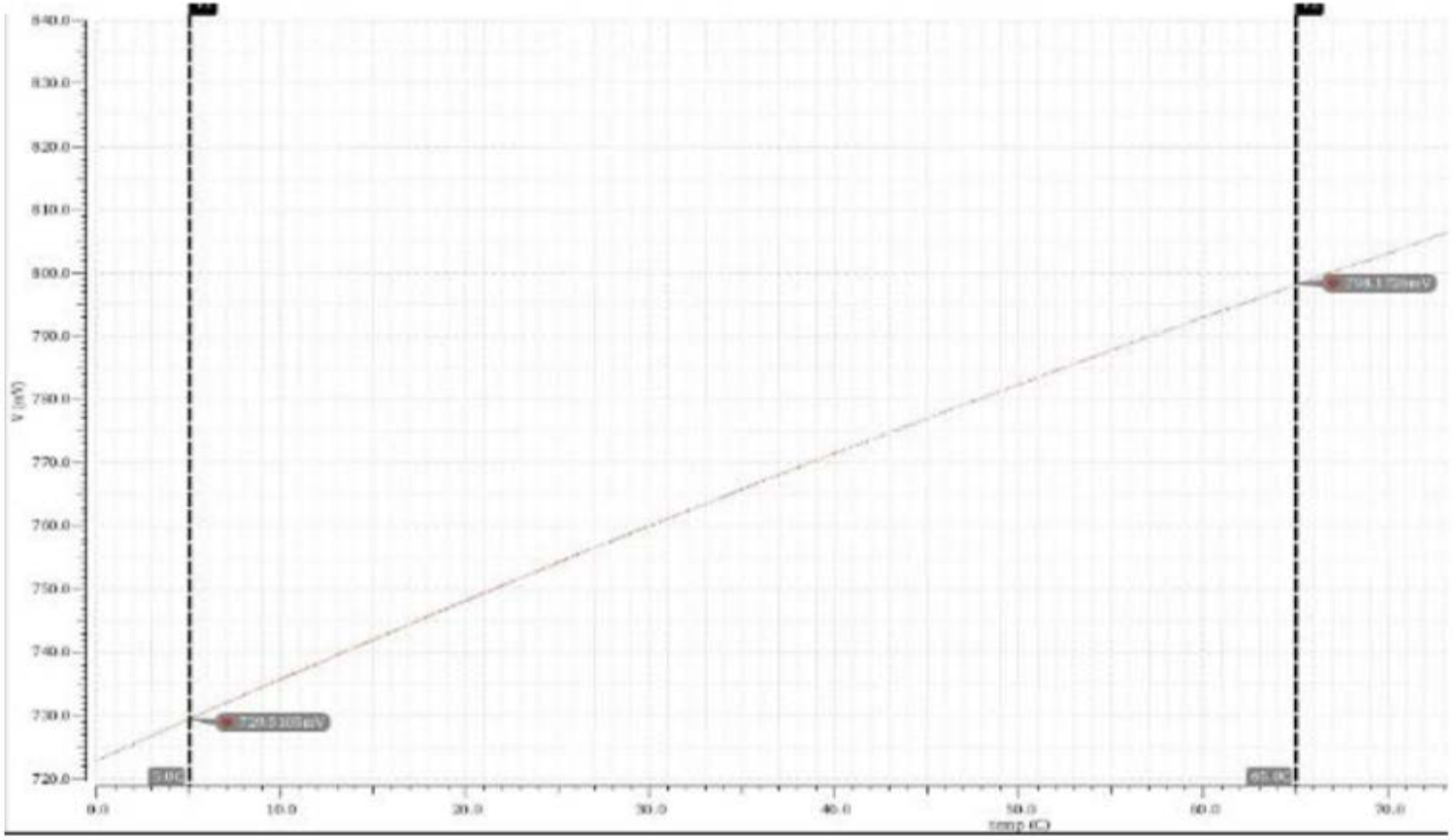
180nm Modulator Layout



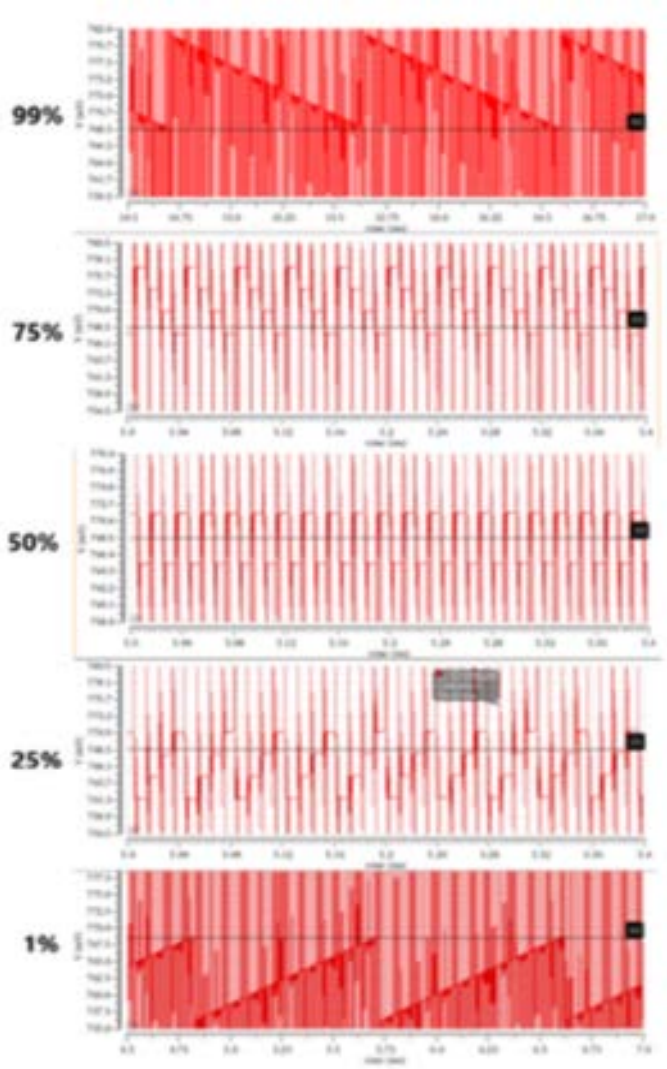
Clock outputs



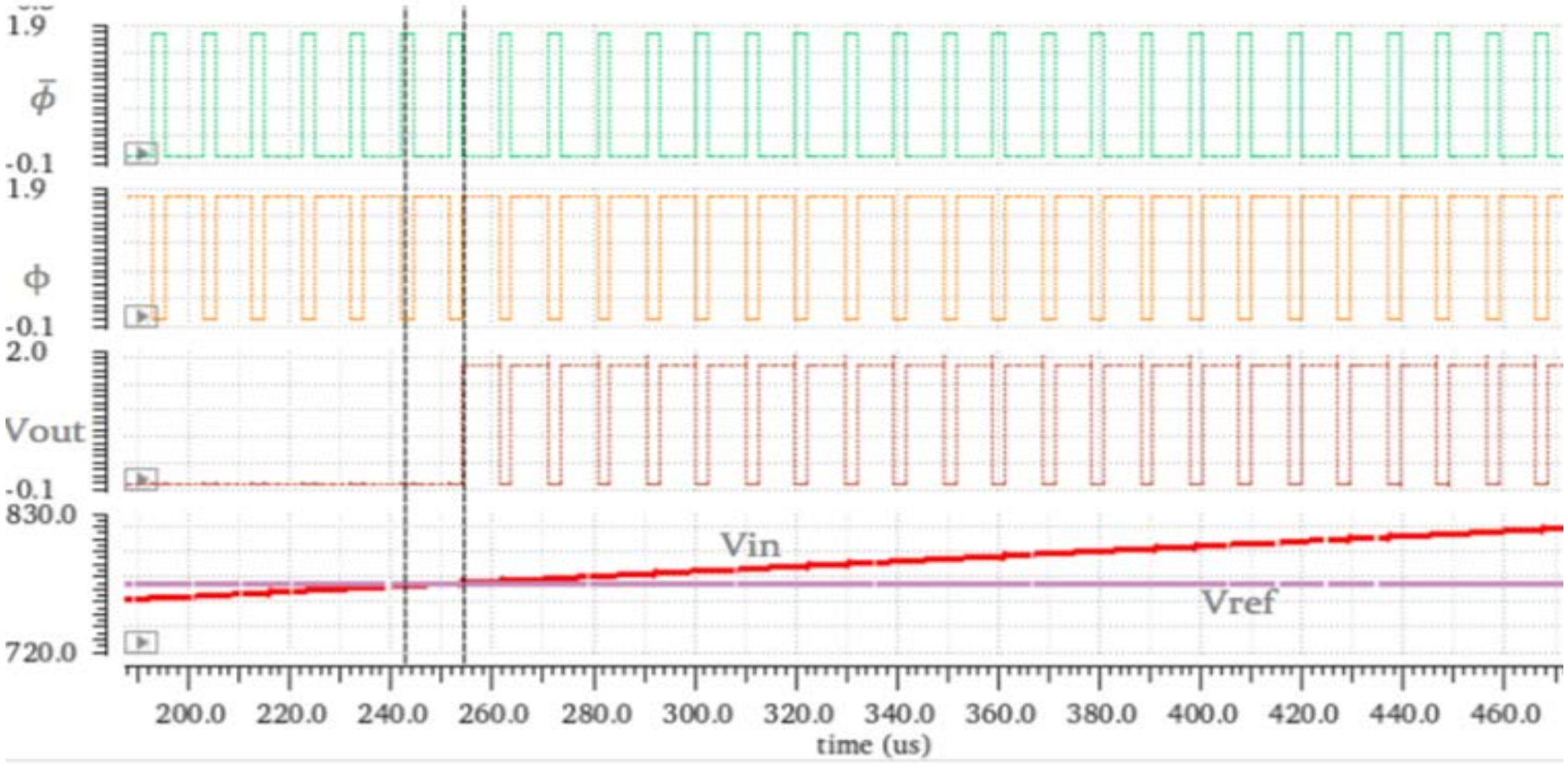
Temperature Sensor Output



Integrator Output



Dynamic Comparator

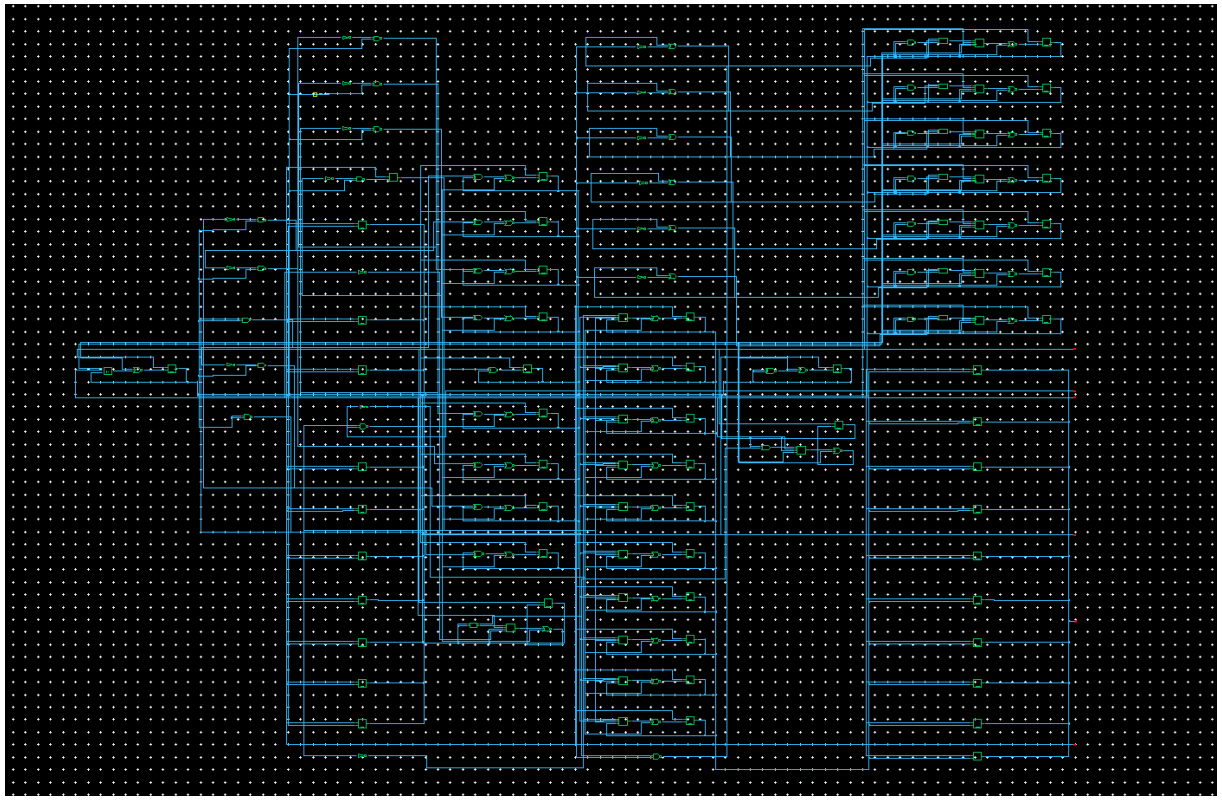


Counter-based Decimator Verilog

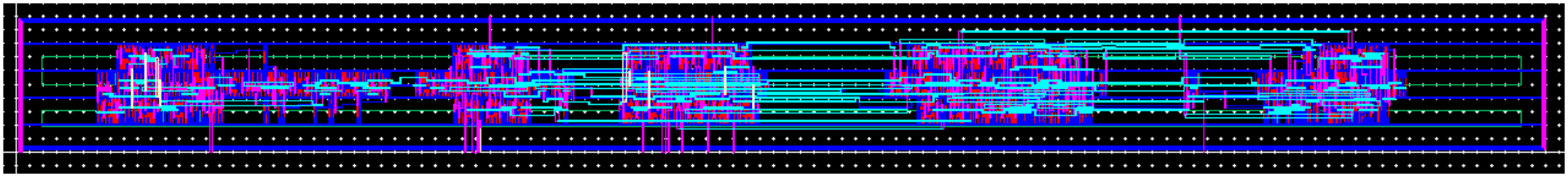
```
58
59    //start the actual connection of the various components for the main_top module
60
61    Register reg1(.reset(r), .resCnt(~resCnt), .clk(clk3), .indata(regin), .outdata(regout));
62
63    updowncount bookcount1(.Clock(clk1), .L(countr), .E(Vin), .up_down(1'b1), .Q(bookout1));
64
65    updowncount bookcount2(.Clock(clk2), .L(countr), .E(1'b1), .up_down(1'b1), .Q(bookout2));
66
67    always @(*) begin
68        countr <= resCnt & r;
69    end
70    always @(posedge clk3) begin
71        if(bookout2 == 10'b1111111111) begin
72            resCnt <= 1'b0;
73        end
74        else begin
75            resCnt <= 1'b1;
76        end
77        regin <= bookout1;
78        parout <= regout;
79    end
80    endmodule
81
```

```
1   // Author: Caleb Davidson
2   // Date: 03/22/2018
3   // Description: Main file for sddec18-20
4   // Delta-Sigma ADC filtration modules
5
6   //main module that make the connects between all the other modules
7   //`timescale 1ns/1ps
8   module main(Vin,r,clk1,clk2,clk3,parout);
9       parameter resVal = 1023;
10      input Vin;
11      input r;
12      input clk1,clk2,clk3;
13      output reg [9:0] parout;
14      reg resCnt;
15      reg [9:0] regin;
16      reg countr;
17      wire [9:0] regout, bookout1, bookout2;
18
19
20  //File: updowncount.v
21  //modified from 281 textbook
22
23  module updowncount (Clock, L, E, up_down, Q);
24    //parameter n = 8;
25    //input [9:0] R;
26    input Clock, L, E, up_down;
27    output [9:0] Q;
28    reg [9:0] Q;
29    integer direction;
30    always @(posedge Clock)
31    begin
32      if (up_down)
33        direction <= 1;
34      else
35        direction <= -1;
36      if (!L)
37        //Q <= R;
38        Q <= 10'b0000000000;
39      else if (E)
40        Q <= Q + direction;
41    end
42  endmodule
43
44  // File: Register.v
45  module Register(reset, resCnt, clk, indata, outdata);
46    input reset, clk, resCnt;
47    input [9:0] indata;
48    output reg [9:0] outdata;
49
50    always @(posedge clk) begin
51      if(!reset)
52        outdata <= 10'b0000000000;
53      else if (resCnt)
54        outdata <= indata;
55    end
56
57  endmodule
```

Counter-based Decimator Schematic



Counter-based Decimator Verilog Layout



Initial Decimator Design Strategies

- Researched the theory and FIR filter structure that generally constructs the low pass and decimation filtration.
- Initial thought process was to build a design centered around summating impulse responses of the input signal, but this was soon found to be not an effective or efficient path to take.
- The Sinc filter of orders 1, 3, and 5 were considered as their frequency response fits the characteristics of the theory behind the decimator filter. Sinc filter of order 1 was considered due to its lessened difficulty to implement where we did not need the added complexity of orders 3 or 5.
- Even further the complexity of a Sinc order 1 filter proved to be more than needed in our output, and in order to design efficiently we chose a simpler counter-based structure for decimation. This provides less noise shaping effects than other structures.

Decimator Unit Testing

- Tested output when 50%, 75%, and 25% duty cycles. All outputs matched the input from an ideal input voltage, and for multiple output codes given the same input cycle.
- Tested high-frequency input with a 50% on, 50% off system, but where on is for one clock cycle and off is for one clock cycle.
- Modulator integration was tested with a full modulator input data. This led to accurate results with each of the above test cases for multiple output codes.

How the Decimator Integrates into the Design

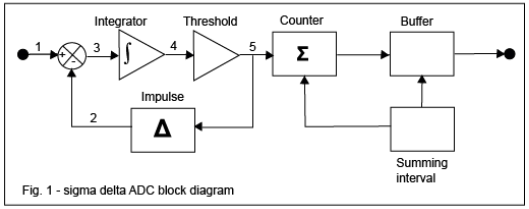
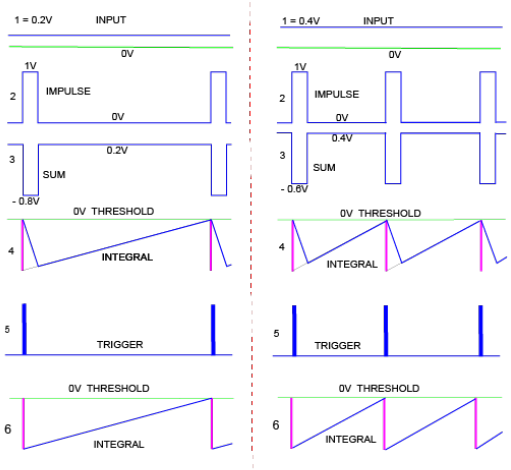


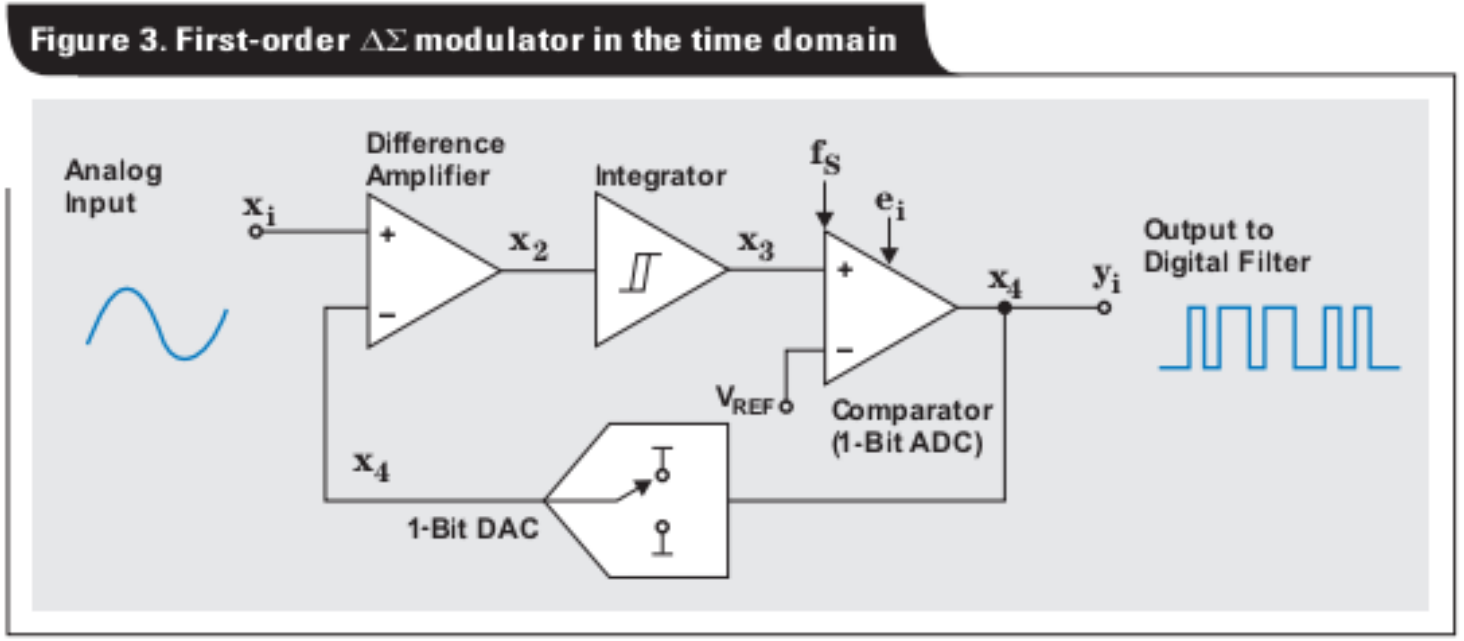
Fig. 1 - sigma delta ADC block diagram



Typical waveforms

- The Sigma component of the delta-sigma architecture is our 10-bit counter reading values from the modulator.
- Another counter acts as our summing interval.
- The register acts as a buffer holding one output code per 1024 inputs from the modulator.

Modulator Diagram



Overview Level Diagram with Averaging Filter Design

Figure 1. Block diagram of ΔΣ ADC

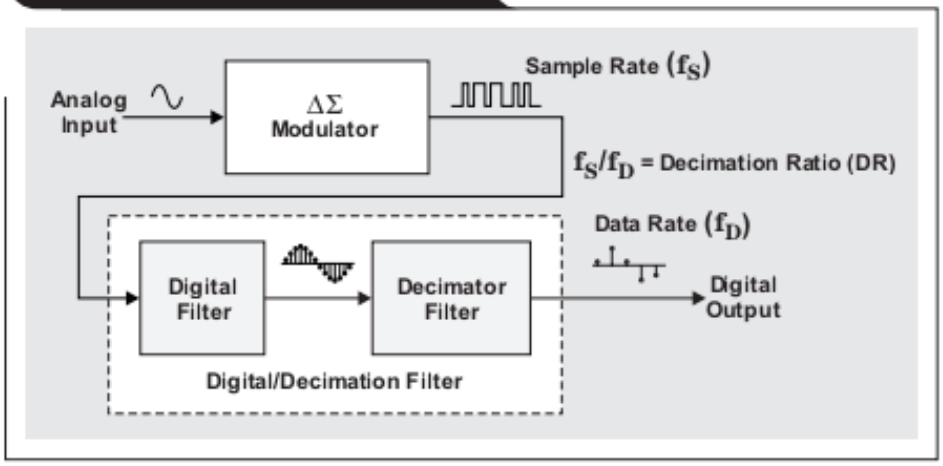
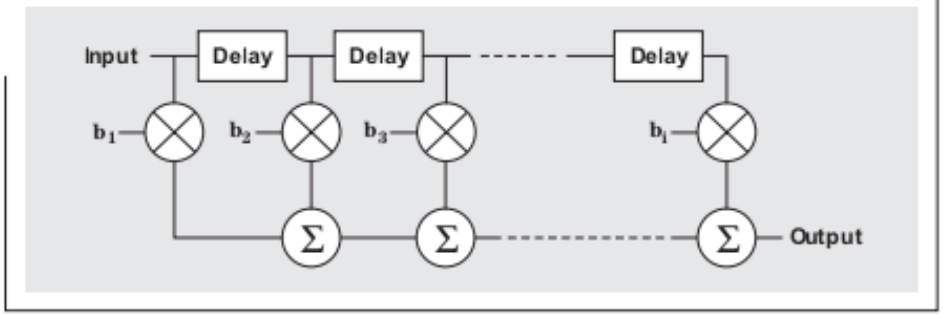


Figure 2. First-order, low-pass averaging filter



Sinc 3rd Order Example Filter Design