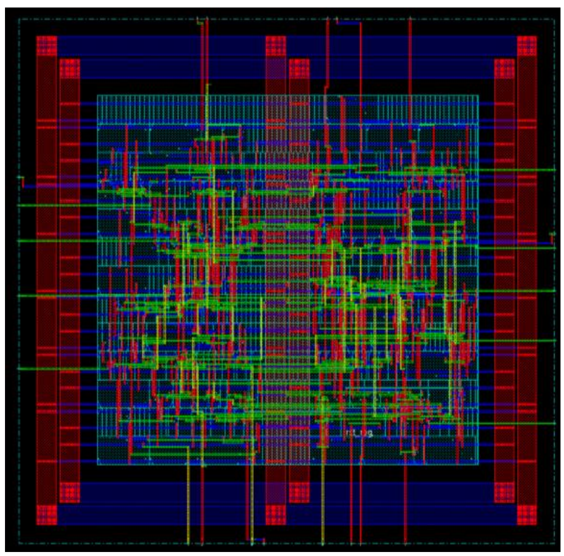
Low Voltage CMOS SAR ADC Design

By

Ryan Hunt



Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

June, 2014

Special Thanks:

To my senior project advisor Dr. Vladimir Prodanov for all of his help with the designs

And

To Dr. Tina Smilkstein for her technical support with Cadence

Table of Contents

Abstract

This project centers on the design of a single ended 10-bit successive approximation register analog to digital converter (SAR ADC for short) that easily interfaces to a micro-controller, such as an Arduino. With micro-controller interfacing in mind, the universal data transfer technique of SPI proved an easy way to communicate between the ADC and the micro-controller. The ADC has a range of 1V (highest code value) to 0V (lowest code value) and operates from a single voltage rail value of 1.8V. Typical SPI clock speeds run on the order of 2MHz [10] and with a 10-bit ADC this means a sampling speed of 200k samples per second, though the design could run at faster speeds. While this design does not provide groundbreaking circuit designs or ideas, it does provide an in-depth learning experience for sub-micron (180nm) circuit design.

Introduction

Several ADC topologies exist. Some of the most popular designs include Δ-Σ ADCs, flash ADCs, and SAR ADCs. By far the most common ADCs are SAR ADCs [13]. The main reason comes down to simplicity and design specifications. SAR ADCs have a decent conversion speed (about 50kHz to 4MHz [13]) and take small overall chip area in comparison to flash ADCs, which are fast but take up a large area. SAR ADC design also flows well with the use of a serial output port due to the nature of the conversion method.

The SAR algorithm works by switching on a large voltage and comparing that to the input voltage. If the switched voltage proves higher than the input voltage then the algorithm turns off that voltage and turns on a voltage half that size and repeats. If the voltage comes up lower than the input voltage it keeps that voltage on and then adds a voltage that represents half the size of the first voltage and repeats. This then corresponds to a series of 1s and 0s. These values are known as bits with the first voltage that is turned on corresponding to the most significant bit (or MSB) and the last voltage corresponding to the least significant bit (or LSB). When the algorithm finishes the result is a binary code that corresponds to the input voltage. This process is referred to as successive approximation. The code produced results in a readable data form for a computer or micro-controller. Figure 1 shows the basics of how the algorithm works where V_{REF} represents the max voltage that the SAR ADC can measure.

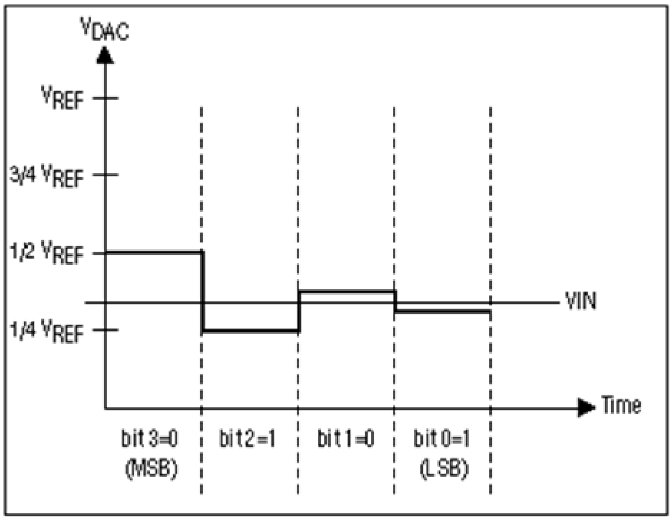


Figure 1: Example of the SAR Algorithm Working [1]

One way of measuring the accuracy of an ADC derives from measuring its differential non-linearity (DNL) or its integral non-linearity (INL). A DNL measurement shows how far off the measured value of each step deviates from the ideal measurement on average and INL represents the measurement of deviation from the ideal line. Figure 2 show an example of DNL error and Figure 3 shows an example of INL error. These terms come up throughout the paper when referring to the accuracy of the ADC.

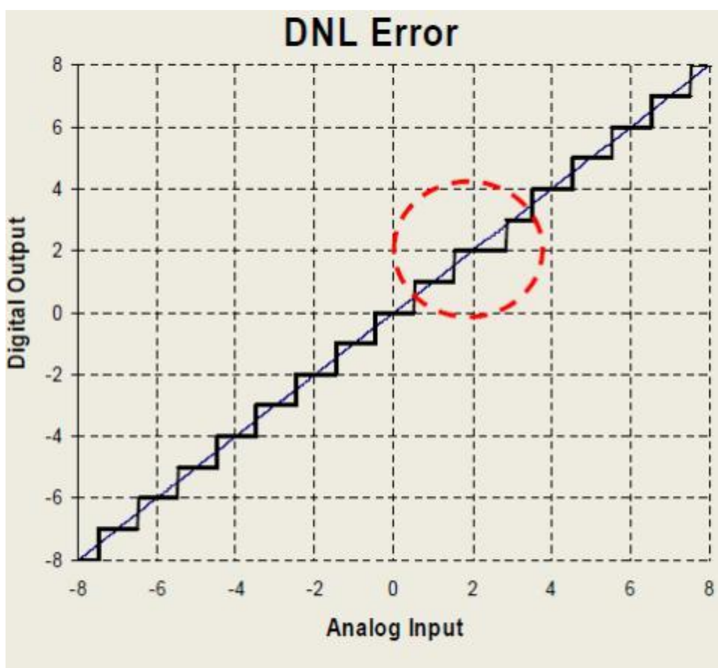


Figure 2: Example of a DNL Error (Blue Line Represents the Ideal Values) [11]

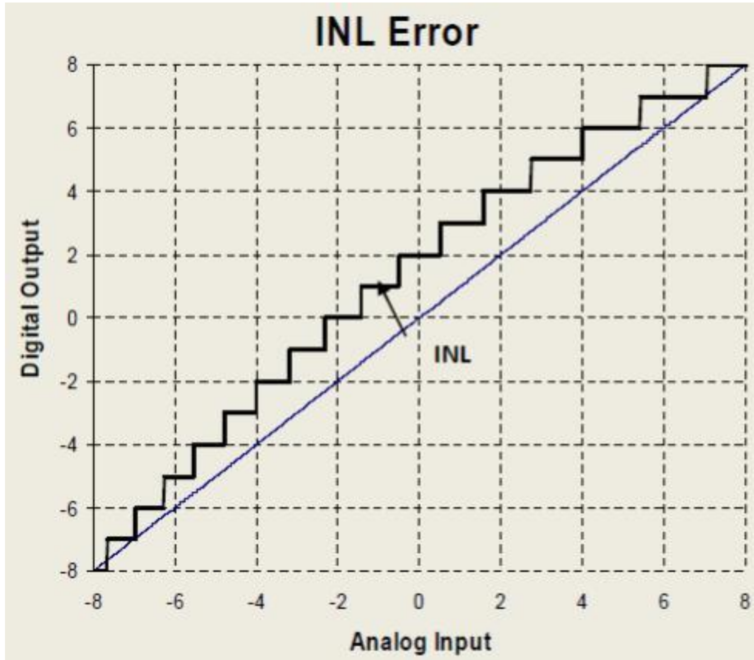


Figure 3: Example of an INL Error (Blue Line Represents the Ideal Values) [11]

Popularity of this design and the relatively straight forward concepts behind it makes it an ideal choice for learning ADC design strategies and interfacing it with other devices.

Requirements and Specifications

An important factor when developing design constraints for the ADC are the needs of the consumer. One, it needs high accuracy over the sampling speed range seeing so the customer receives accurate data. Two, it needs comparable resolution to what comes standard on the development boards currently on the market to provide an effective alternative. Third, due to the constraints already put on the technology that it uses the size needs a specific limitation, which nicely coincides with keeping it compact for the consumer. Last, what type of device the consumer may use with the ADC creates a design constraint; in this case a micro-controller. This constraint puts restrictions on how to pass along data due to the limited amount of inputs on most micro-controllers. Using a serial interface makes sense for this design and SPI has a limit of 10 bits on most micro-controllers [10] so this sets the largest resolution that the ADC can output. Table I addresses the consumer needs and puts them in terms of marketing requirements and engineering specifications.

TABLE I

SAR ADC REQUIREMENTS AND SPECIFICATIONS

Marketing Requirements	Engineering Specifications	Justification
1.	The ADC must sample at 200k samples per second effectively. [10]	Based off of typical SPI speeds (around 2MHz) [10] created by a micro-controller and the resolution of the ADC this gives the 200k samples per second.
2.	The design must fit in an area of 0.9mm x 0.9mm. [4]	This size relates to the type of chips that the school can fabricate so it fits in well with the compact marketing requirement.
3.	It must cost under $25 when put into mass production.	This price allows people on a budget to afford to by the ADC while still leaving room for profit margin.
4.	The ADC must have a 10 bit resolution with a 0V to 1V range. [1, 4, 10]	This allows the use of a SPI output data form at the max allowed. The voltage range gives the design some voltage headroom.
1.	DNL and INL within 1LSB. [2, 11]	This allows for little error and to ensure the accuracy of the ADC.
5.	Serial Data transfer using the standard of SPI. [10]	Due to the limitations of most micro-controllers in terms of pins they have available, the use of serial data transfer is the most effective data transfer type.
6.	The device must run off a 1.8 V rail.	This value comes about due to the restrictions placed on the designs due to the use of 180nm technology that the device design revolves around. [4]
Marketing Requirements 1. High accuracy in sample range. 2. Compact for ease of use. 3. Low cost for use by DIY engineers. 4. High resolution. 5. Serial transfer of data 6. Low voltage rail		

Using Table I the design of the SAR ADC and the project goals are set. The project meets many of the requirements but due to some technical issues such constraints as keeping the DNL and the INL within 1 LSB remain untested. An explanation of the issues is presented later in the paper.

Functional Decomposition

This section breaks the design down into 3 separate levels. Level 0 depicts a basic block diagram showing all of the inputs and outputs of the ADC. Level 1 depicts the basic flow of the signals in the design and how the signal processing works. Level 2 shows all of the connections made in the chip and how everything interacts with each other.

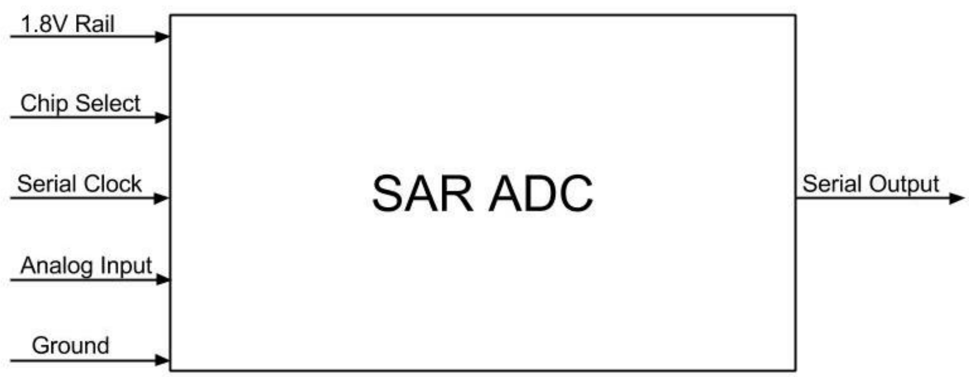


Figure 4: Level 0 Block Diagram

Figure 4 shows the level 0 block diagram. It has 5 inputs and 1 output with 3 of the inputs controllable externally (Chip Select, Serial Clock, and Analog Input). Table II explains what each of these inputs do.

TABLE II
SAR ADC LEVEL ZERO BLOCK DIAGRAM TABLE

	Name	**Description**
Inputs	1.8V Power Supply	DC power supplied from external source. (i.e. micro-controller or battery)
	Analog Input	Input coming from sensor in the analog domain with a 0V to 1V range.
	Chip Select	This control signal tells the ADC that it needs to operate and use the SPI line.
	Serial Clock	External SPI clock taken from the controller to act as the base clock for the ADC
Outputs	Serial Output	Digital code sent out by the ADC representing a voltage value to desired data storage location.
Functionality	The ADC converts analog data into digital data that devices like micro-controllers can understand and use. The ADC waits for the chip select to tell it when to gather and report the data. It uses an external SPI clock to synch with other devices. This design uses a low voltage rail of 1.8V given from the micro-controller to power the ADC.	

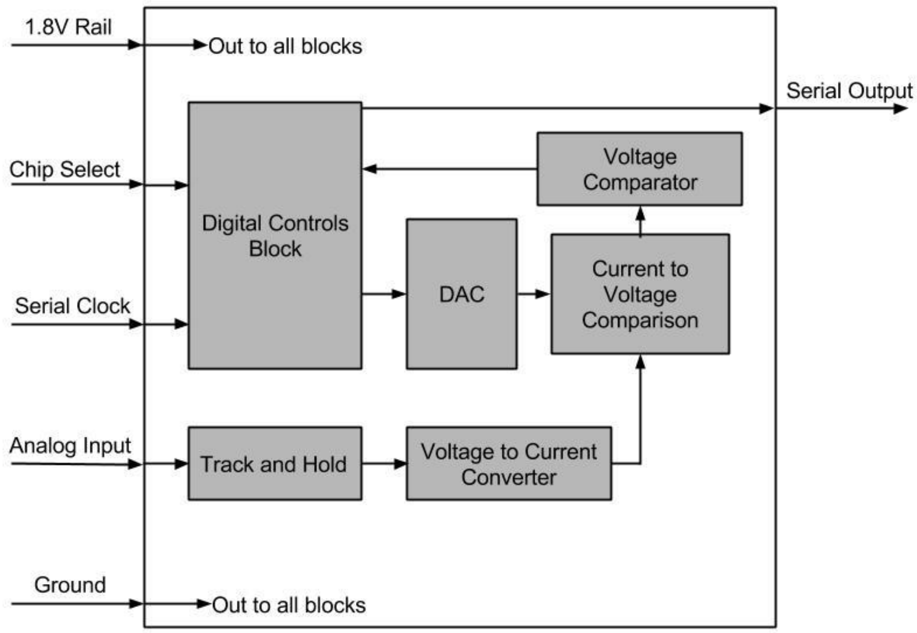


Figure 5: Level 1 Block Diagram

Figure 5 shows the level 1 block diagram. It breaks down into 6 main blocks that deal with the flow of the signal and resulting data. These blocks consist of the track and hold, the voltage to current converter, the current to voltage comparison, the digital to analog converter (or DAC), the voltage comparator, and the digital controls. Table III explains the purpose of each block shown in Figure 5.

TABLE III
SAR ADC LEVEL ONE BLOCK DIAGRAM TABLE

	Name	**Description**
Components	Track And Hold	Samples the analog input and holds that value for the comparator to look at until the full data processing completes
	Voltage to Current Converter	Produces a current that is linearly proportional to the input voltage.
	DAC	This component is a digital to analog converter. It takes data from the digital processing block and then outputs an analog value for the comparator to compare to.
	Voltage Comparator	Compares the current to voltage comparison value to the analog input value coming from the current to voltage comparison.
	Digital Controls	Processes data received form the comparator and the ADC Controls and then controls the SAR DAC
	Current to Voltage Comparison	This block takes the current produced by the DAC and the voltage to current converter and compares them in a way that pulls a voltage higher or lower based on which input has a larger current

Flow based on [1], [5], [13]

One could determine the basic signal flow of the circuit from Figure 5 but further explanation is necessary to get a full understanding. Initially the analog input goes into the track and hold amplifier which tracks the voltage on the input and then holds a voltage specified at a certain time by the digital controls. It holds this voltage until the ADC finishes the SAR algorithm. The voltage to current converter then takes the voltage and creates a proportional current. This current then gets compared to the current that the DAC produces. The voltage produced by the DAC comparison then gets compared to a reference voltage. The comparator then outputs a High or Low voltage based on this comparison that the DAC reads in and then adjusts its current accordingly. At the same time the values from the comparator get outputted as the serial output. These comparisons repeat until a value for all 10-bits gets outputted.

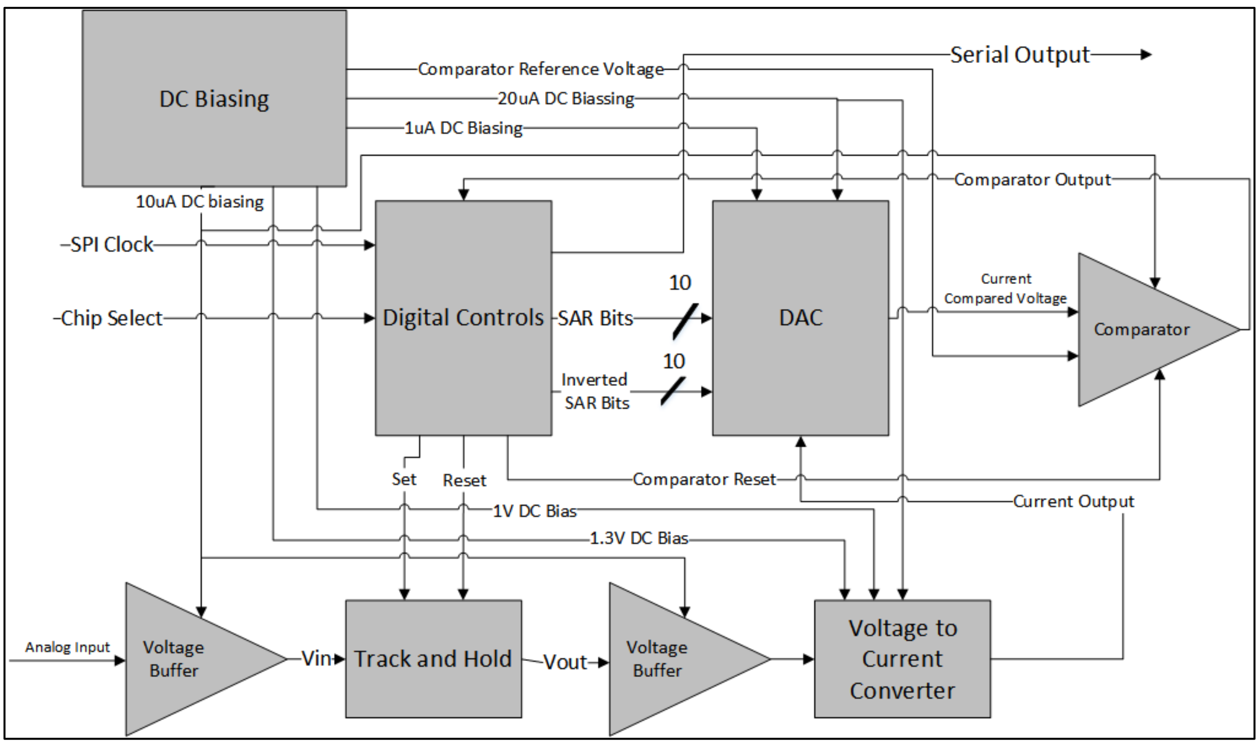


Figure 6: Level 2 Block Diagram

Figure 6 shows the level 2 block diagram. This shows exactly how all of the circuits connect to each other as seen in Appendix A Figure A1. The appendix figure shows the test layout for the circuit. Some main points to note include the disappearance of the current to voltage comparison from the level 1 block diagram to the level 2 block diagram. This component actually lies within the DAC as a part of its design. DC biasing also now appears in this diagram versus appearing in Figure 5 because it does not have a direct effect on the data. Also the addition of the voltage buffers to the diagram help show the type of track and hold that has been implemented. In reality the two voltage buffers are a part of the track and hold system. Note that all blocks except the track and hold block receive the 1.8V rail; all blocks receive ground. Table IV shows the breakdown of what each internal signal represents.

TABLE IV
SAR ADC LEVEL ONE BLOCK DIAGRAM TABLE

	Name	Description
Signal	SPI Clock, Chip Select, Analog Input, Serial Output	External signals explained in Table II.
	Vin	Buffered input voltage
	Vout	Voltage that held by the track and hold.
	Current Output	Current output proportional to the input voltage from the second voltage buffer
	10 µA DC Biasing, 20µA DC biasing, 1µA DC biasing	Biasing currents distributed to various blocks throughout the system.
	1V DC Bias, 1.3V DC Bias, Comparator Reference Voltage	These are the DC voltage biases and references. They remain at a constant voltage to bias various components or act as a reference voltage.
	SAR Bits & Inverted SAR Bits	These two ten bit busses control the switching circuit in the DAC controlling how much current the DAC produces.
	Comparator Reset	This line controls when the Dynamic comparator resets.
	Current Compared Voltage	This line brings the resulting voltage from the current comparison to the comparator
	Set, Reset	These signals tell the track and hold when to set the voltage and when to reset the voltage that it holds. Both use pulses.

SAR Analog to Digital Converter Design

In the most basic sense the project consists of two parts: analog components and digital components. In the case of this design more work and emphasis is placed on the analog components than the digital components mainly because this design does not contain any kind of complicated calibration or correction stages. The digital block only handles the SPI controls, basic ADC controls, and the SAR algorithm.

The design has 6 main analog circuits. These consists of the voltage buffer, the track and hold, the voltage to current converter, the digital to analog converter (DAC), the comparator, and the biasing circuit. How all of the components, including the digital block, connect together is seen in Appendix A, Figure A1. This design follows basic ADC design ideas taken from Dr. Prodanov's EE 409 course notes [13] and from a Maxim Integrated web page on understanding SAR ADCs [1]. The software for designing this project comes from the Cadence design suite using an 180nm IBM processes through Mosis [4].

The following subsections breakdown each of these components and show how and why each design works. Larger images of all designs are seen in Appendix A. All sizing of transistors are collected in tables and provided in Appendix B.

Voltage Buffer Design

The purpose of the voltage revolves around protecting the track and hold and the device providing the input voltage. They buffer both the input and the track and hold amplifier meaning they do not draw a lot of current from their inputs while supplying or sink a substantial amount of current at the same time to their outputs.

The design for the voltage buffer comes from a Texas Instruments [12] design for an op-amp with a few changes. The op-amp uses a voltage follower mode of operation to create a voltage buffer. Also in the Texas Instruments design they use a resistive and capacitive feedback but that proves unnecessary as this design achieves good loop stability with only a capacitive feedback. The main reason for choosing this design revolves around its relatively simple design and its ability to "pull" to ground with a capacitive load. Figure 7 shows the voltage buffer design used for the ADC.

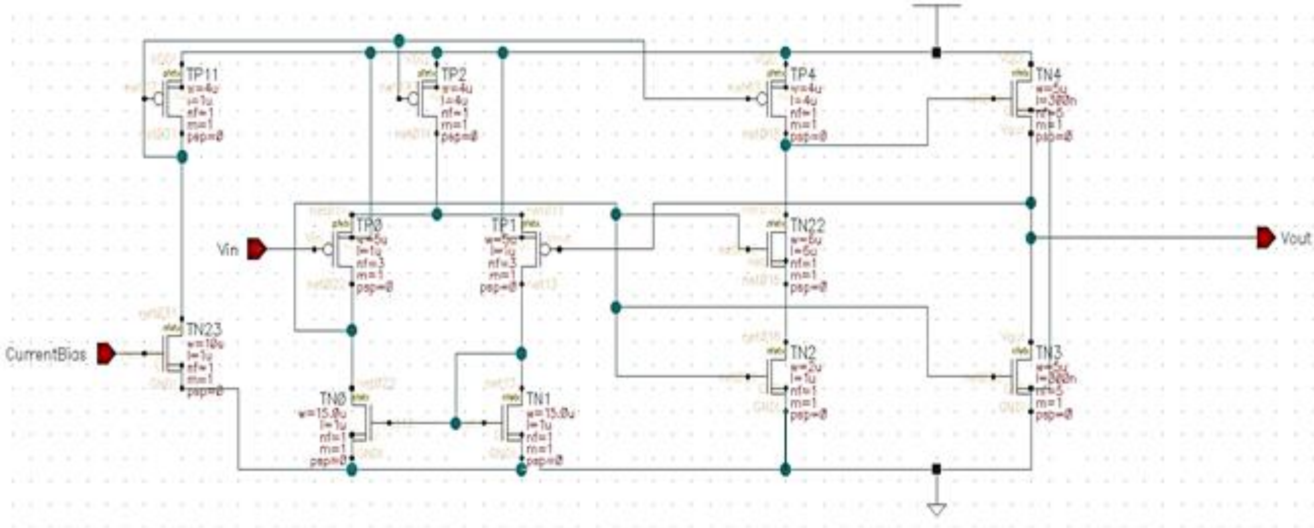


Figure 7: Voltage Buffer Schematic

Input to the voltage buffer comes in on the gate of TP0. This is the positive input of the op-amp while the negative input connects to the output of the op-amp thus placing the op-amp into a voltage follower mode. TP2, TP0, TP1, TN0, and TN1 make up the differential input stage of the op-amp. TP4 and TN2 make up the second stage of the op-amp with TN22 acting as the capacitive feedback. Note that when the drain, body, and source of a CMOS transistor are connected together it acts like a non-linear capacitor with one terminal at the gate and the other at the tied drain and source nodes. This setup tends to create more capacitance per area than a typical metal-to-metal capacitor but they have a non-linear with the applied voltage. TN4 and TN3 provide the push-pull stage for the op-amp allowing it to source and sink current without affecting the rest of the circuit. TP11 and TN23 interface the op-amp to the DC biasing circuit. TP11, TP2, and TP4 make up the current mirrors used in the op-amp. Sizing of transistors' width over length values (or W/L ratio) began initially by following typical choices for ratios for op-amps based on [2]. Once initial values were chosen, various transistors are adjusted to see how they affected the op-amps specs and output. TN0 and TN1 have a direct impact on how closely the buffer can reach ground. The relationship between TN0, TN1, and TN2 would affect various aspects of loop stability such as ringing during the slew rate test. Increasing the W/L of TN3 and TN4 allows for larger loads because it increases the rate that the push-pull stage could sink and source current. Increasing current through the first stage by adjusting its W/L ratio with respect to TP11 causes it to have a faster slew rate but also can cause more ringing. The design with respect to the W/L ratio comes down to a balancing act that depends on both design and the technology used. If one uses the exact same W/L ratios and circuit design but a different technology the result may turn out completely different.

Due to the requirements of the design, it needs two voltage buffers that have different W/L ratios. The two voltage buffers' W/L ratios are collected in Appendix B Tables B1 and B2. Buffer 1's design reflects the need to buffer the output of the track and hold to the input of the voltage to current converter. Buffer 2's design reflects the need to buffer from the input to the chip to input to the track and hold. The main difference between these two voltage buffers involves the type of load they need to supply and sink current to. Buffer 1 connects to a resistive load from the voltage to current converter while buffer 2 connects to the purely capacitive load given by the track and hold circuit. See the Track and Hold and the Voltage to Current Converter sections to better understand the loads. Because the design depends on the load many of their specs differ in several ways. These differences become apparent when comparing their specs seen in Table V and Table VI.

TABLE V
SPECS FOR VOLTAGE BUFFER 1

Specification	Data
Slew Rate	11.86 V/µs
Peaking Range	1MHz to 10Mhz
Open Loop Gain	13k V/V
-3dB Cutoff Frequency	15MHz

Table V shows the data collected from buffer 1. Note that these tests use a capacitive load of 461fF to compare its specifications to buffer 2 on equal terms. 461fF represents the measured capacitance of the track and hold circuit (see Track and Hold section for more information). The test results seen in Table V come from the tests seen in Figures 8 through 10.

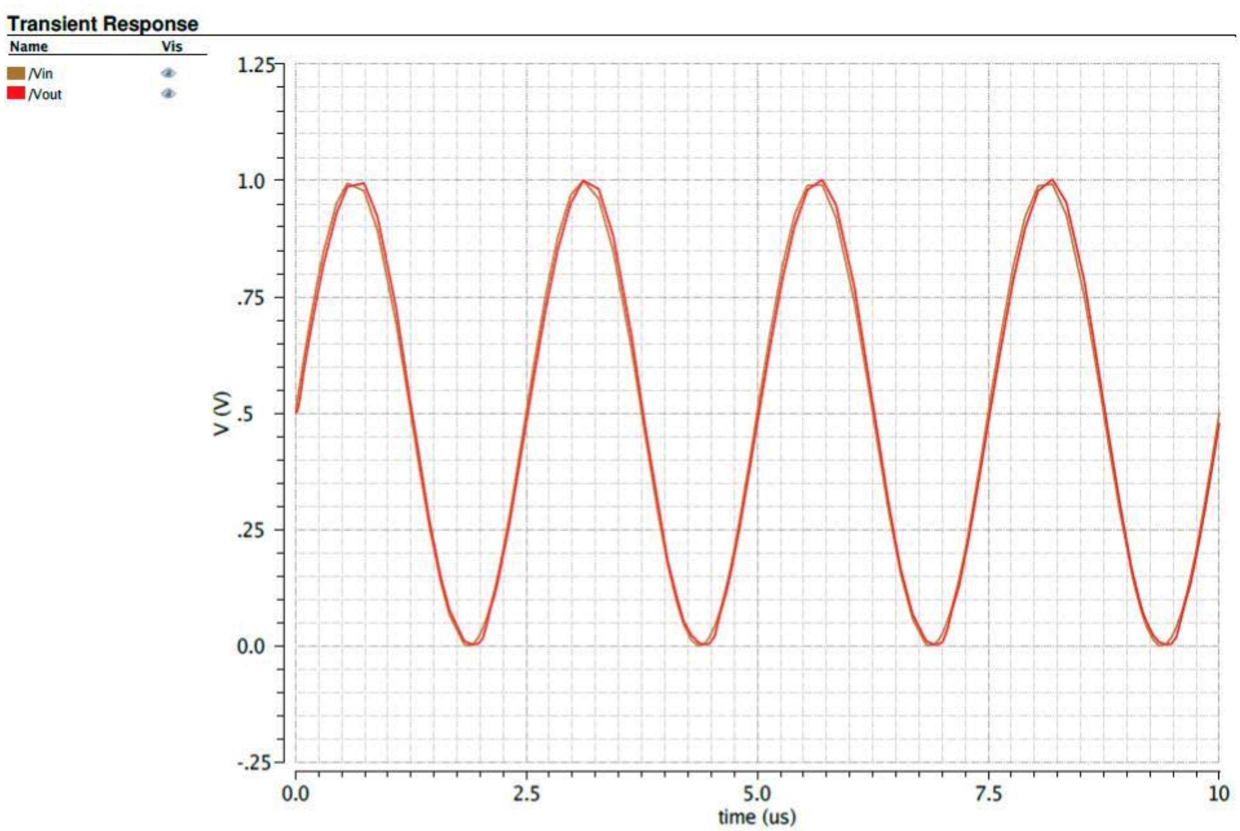


Figure 8: Sinusoidal Input for Voltage Buffer 1 with $1V_{pp}$, 0.5V DC Offset at 400kHz with 461fF Load

Figure 8 shows the voltage buffer operating at 400kHz in its full range of operation. Note some slight distortion at the top and bottom of the sinusoid seen in the red line which represents the output voltage. The distortion at the top has no apparent cause and may be due to an issue with the algorithm that the test program uses to calculate some of the values as they pass through the models of the transistors. The slight distortion on the bottom though derives from the sizing of the transistors. Due to the need to reduce the voltage drop across TN3 to allow it to pull to ground some of the transistors need their sizing adjusted in a way that fits all of the specs in the most effective manner. By adjust the sizing this way it creates the distortion seen in Figure 8. With more time this buffer could be optimized to reduce this distortion though currently it does not prove to cause too large of an issue beyond causing a slight DNL error at those voltages.

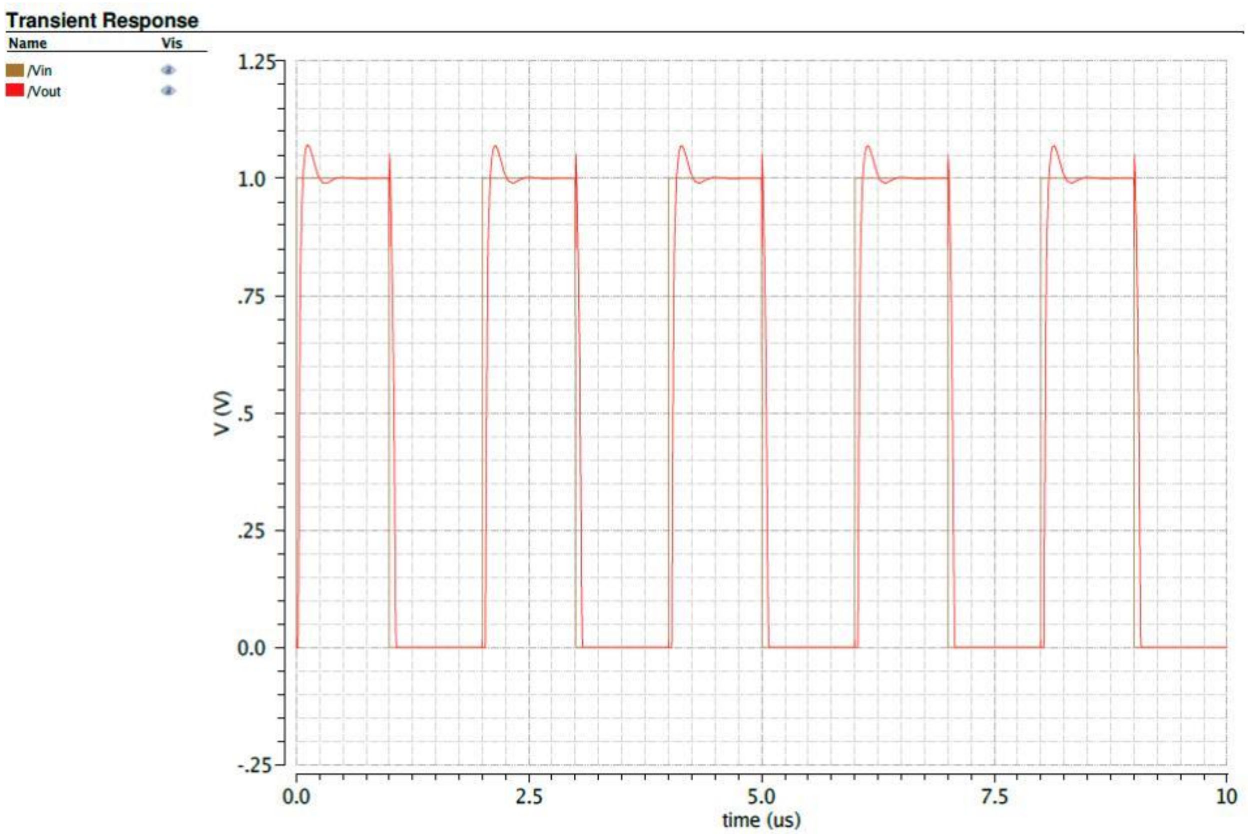


Figure 9: Slew Rate Test for Voltage Buffer 1 with a Square Wav Input of 0 to 1 V at 500kHz with 461fF Load

Figure 9 shows the slew rate test for buffer 1. While relatively fast, the test does show some ringing when using a square input. While not ideal, reducing this proves difficult. If the capacitive feedback increases in capacitance the slew rate goes down drastically and causes a very apparent phase shift when running the test in Figure 8. Once again, with some time optimizing this buffer to work more effectively by reducing the ringing seen in Figure 9 should prove possible.

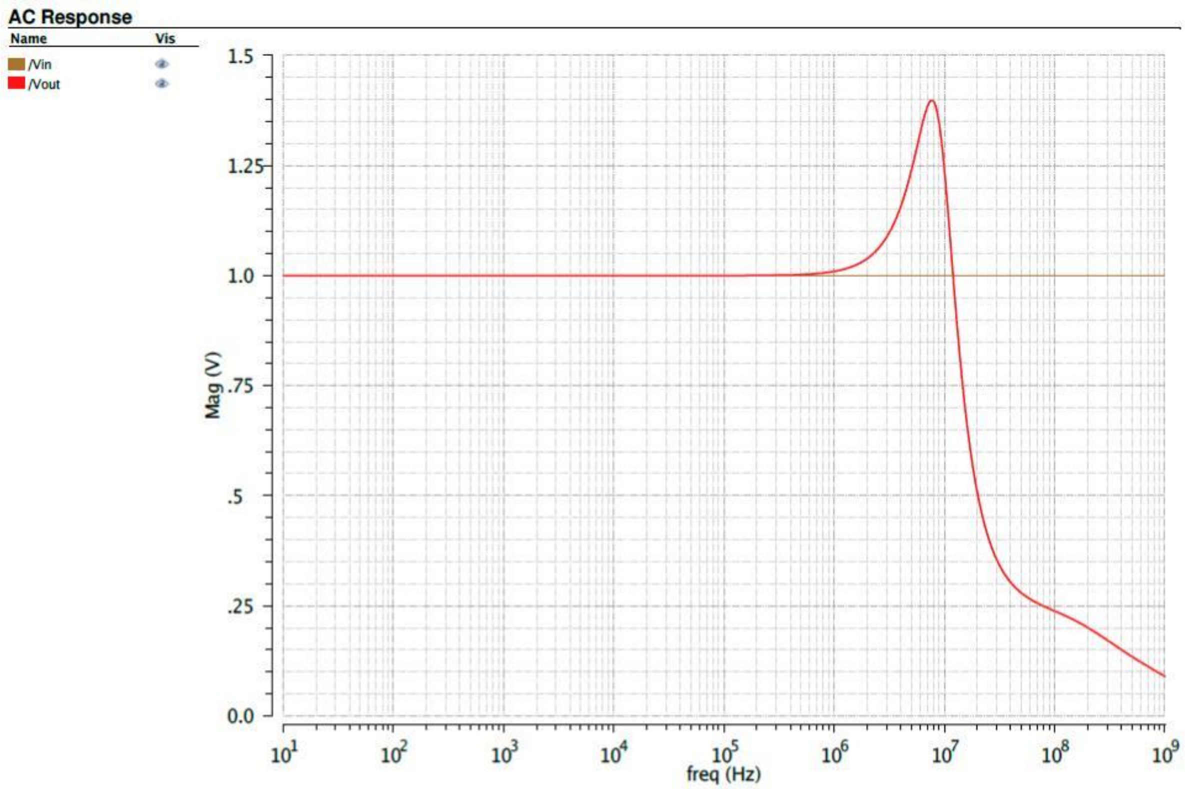


Figure 10: AC Magnitude Response Test for Voltage Buffer 1 Sweeping from 10Hz to 1 GHz with 461 fF Load

Figure 10 shows the AC magnitude response of voltage buffer 1. This graph gives the peaking range, open loop gain, and the -3dB cutoff frequency. The peaking seen reflects what shows up in Figure 9 in the form of ringing. This overshoot does present an apparent issue with the voltage buffer but due to fact that the track and hold can only sample at 200k samples/second meaning the frequencies seen by buffer 1 should not exceed the point where the peaking begins at 1MHz. The open loop gain for the op-amp represents how effective the buffer is at achieving unity gain. The data comes from equation 1 where V^+ is the input voltage and V^- is connected to the output. The larger the number the more effective the buffer at achieving unity gain because this makes the denominator go closer and closer to zero. In the case of buffer 1, it excels in this field with an open loop gain of 13kV/V.

$$Open\ Loop\ Gain = \frac{V_{out}}{V^+ - V^-} \qquad Equation\ 1$$

The -3dB cutoff frequency in this case proves less important than where the peaking begins because if operated beyond this point it would introduce far too large of an error for the ADC to work properly. Still it provides a common spec used to gauge effectiveness of an op-amp.

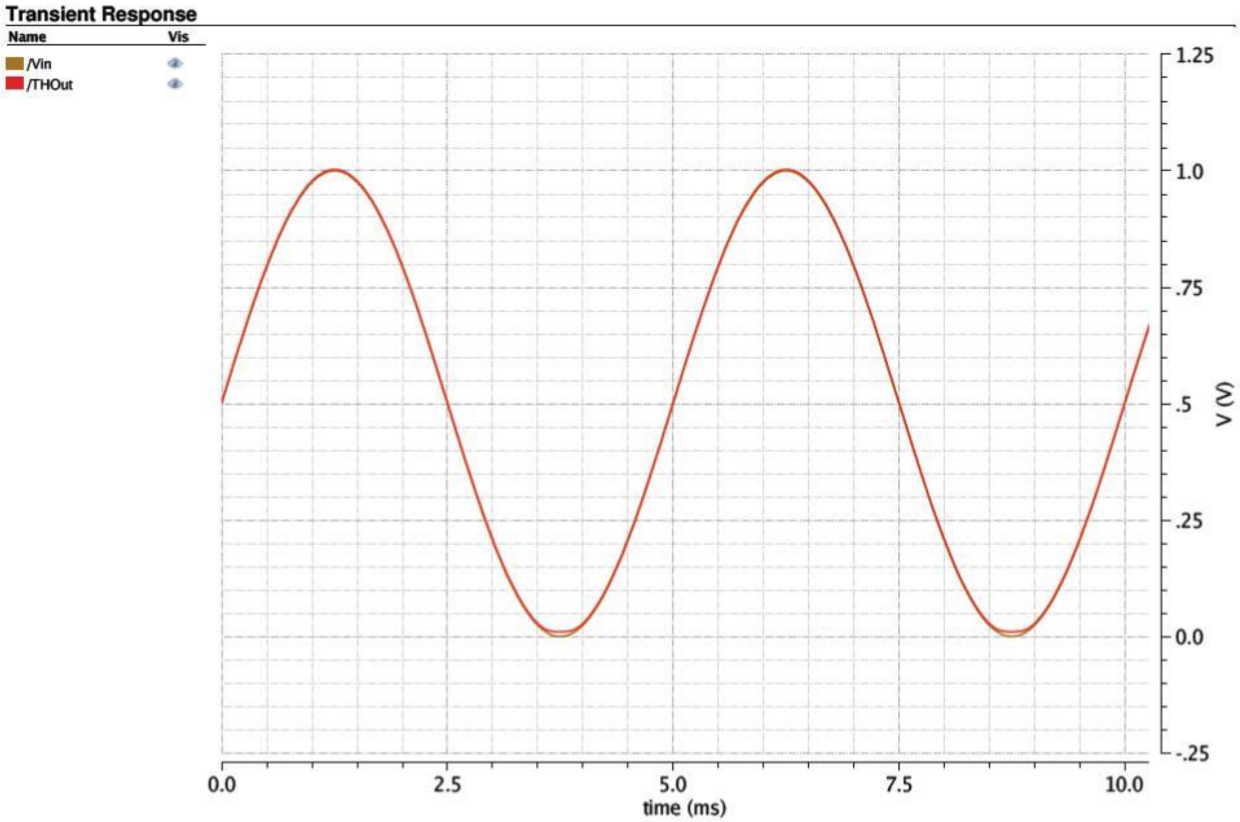


Figure 11: Voltage Buffer Error When Connected to the Voltage to Current Converter (Red Line = Buffer Output)

Figure 11 shows the error that occurs when attached to the voltage to current converter. The error refers to the buffer's inability to "pull" all the way to ground. At its peak it has a deviation of around 9mV, which is about 9.7mV (about 10 LSBs worth of error). In terms of ADCs this presents an issue for the accuracy and the DNL and INL will reflect this. The reason for this revolves around the load, as mentioned before. Due to the fact that the load is resistive this means current must always flow through the sinking transistor due to the nature of the voltage to current converter. This means that a voltage drop must always exist across the current sinking transistor and thus cannot pull all the way to ground.

TABLE VI
SPECS FOR VOLTAGE BUFFER 1

Specification	Data
Slew Rate	13 V/µs
Peaking Range	3MHz to 33Mhz
Open Loop Gain	2.4k V/V
-3dB Cutoff Frequency	35MHz

Table VI shows the data collected for voltage buffer 2. The design for this buffer proves far easier than for buffer 2. The reason it proves easier has to do with using a capacitive load. When this voltage buffer pulls to ground eventually the capacitor drains completely and thus no longer injects current into the push-pull stage unlike the resistive load which will always have current going through it. This allows it to pull to ground within about 33μV.

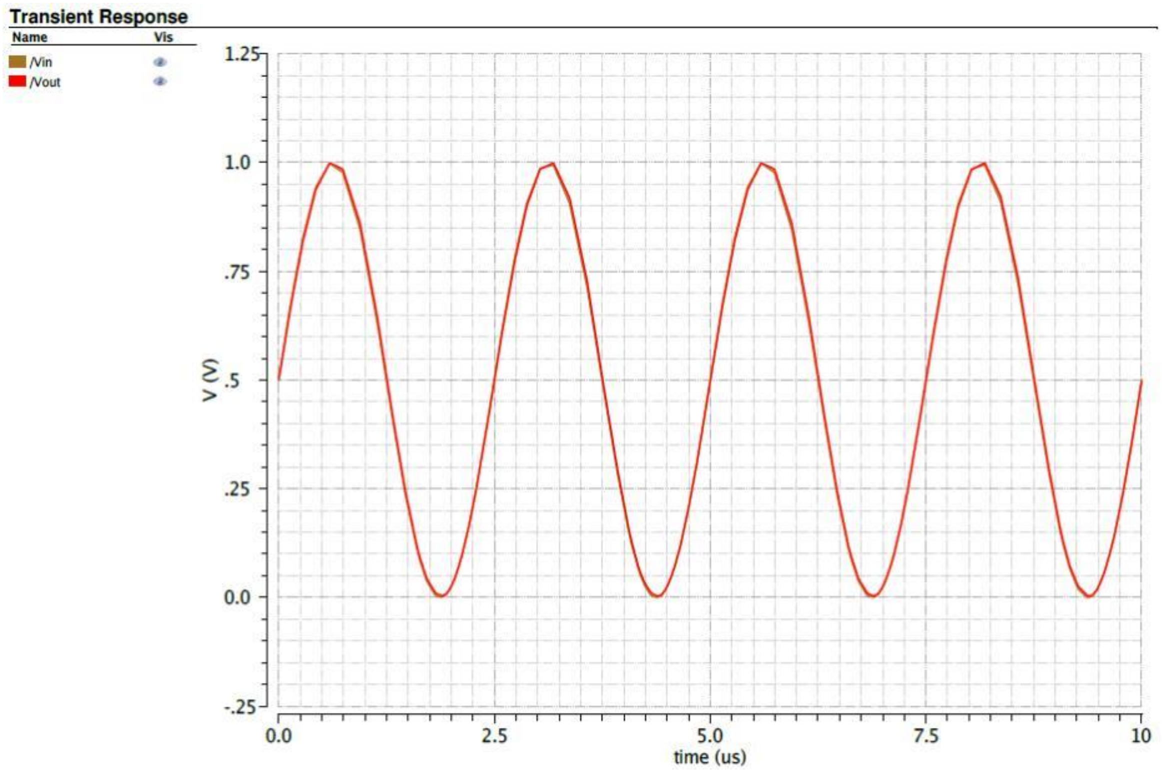


Figure 12: Sinusoidal Input for Voltage Buffer 2 with $1V_{pp}$, 0.5V DC Offset at 400 kHz with 461 fF Load

Figure 12 shows the same test setup as seen in Figure 8 but with buffer 2. One can easily notice the lack of brown representing the input voltage. This means that the buffer follows the input voltage nearly perfectly. None of the distortion seen in Figure 8 presents itself in Figure 12.

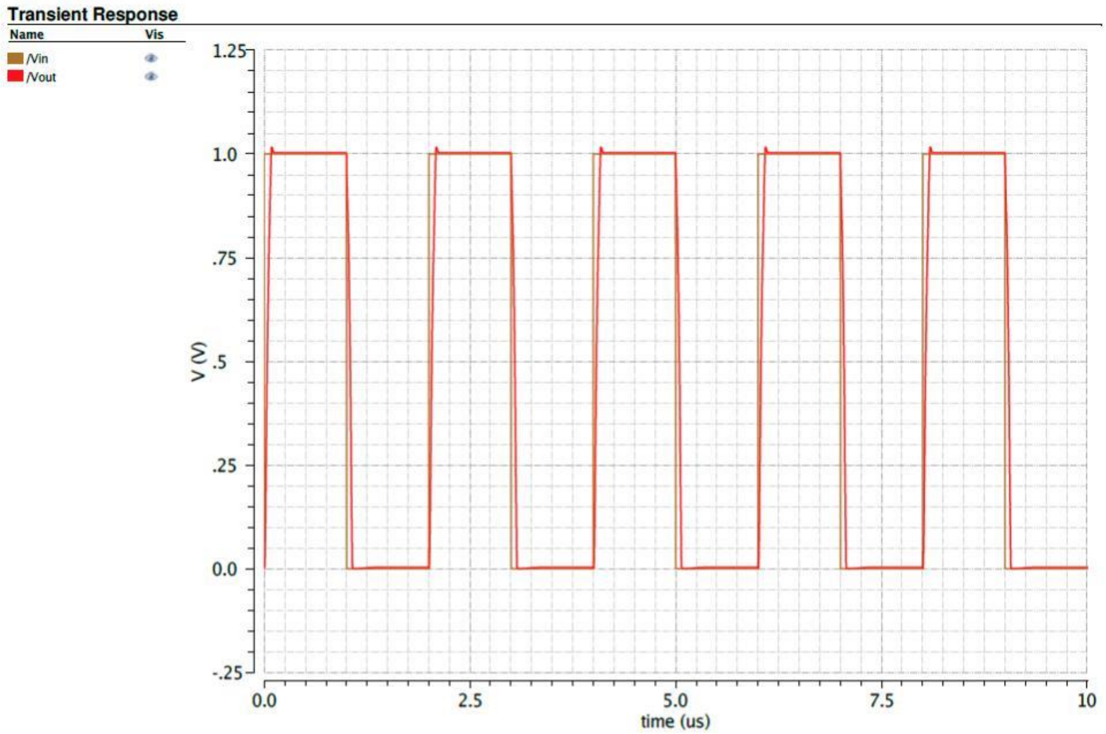


Figure 13: Slew Rate Test for Voltage Buffer 2 with a Square Wav Input of 0 to 1 V at 500 kHz with 461 fF Load

Figure 13 shows the slew rate test for buffer 2. Once again it shows substantial improvement over buffer 1. Not only does it have a faster slew rate as seen in Table VI but virtually no ringing exists in comparison to buffer 1.

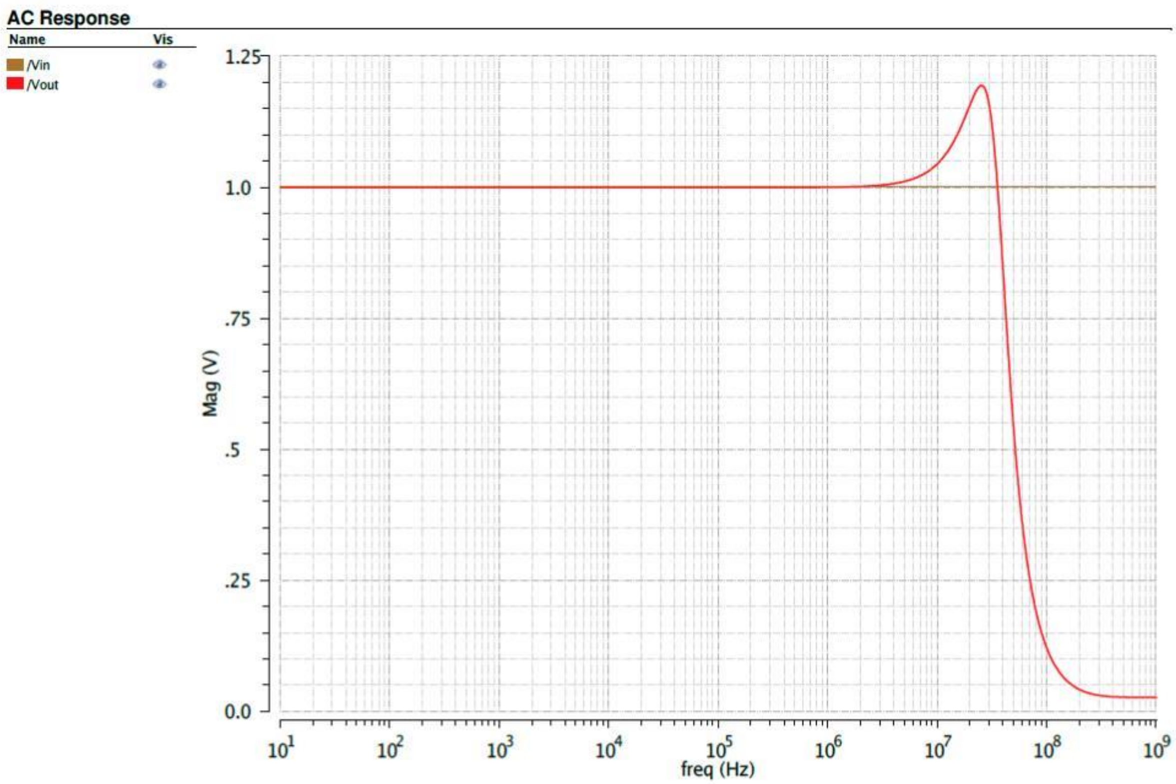


Figure 14: AC Magnitude Response Test for Voltage Buffer 2 Sweeping from 10 Hz to 1 GHz with 461 fF Load

Figure 14 shows the magnitude response of buffer 2. The peaking is nearly half of the value above a magnitude of 1V that it was for buffer 1. The peaking range does triple but so does the point where the peaking range starts. Also the cutoff frequency of buffer 2 more than doubles that of buffer 1. One aspect to note where buffer 2 performs worse than buffer 1 is the open loop gain. Buffer 2's open loop gain is nearly 11k (a drop of 81%) worse than buffer 1. While this value proves substantial the open loop gain of buffer 2 still presents a large enough value that it does not introduce anywhere near a LSB (about 0.98mV) of error thus rendering it unnecessary to increase.

The main conclusion to draw from this data comes from making sure that the design caters to the load. Even though buffer 2 presents far better specs in nearly every aspect it would completely fail when driving the same load that buffer 1 drives. Even with all the advantages that buffer 1 it still has its own issues. It needs some more time put into its design but for the purpose of this project it suffices.

Track and Hold

The track and hold follows a fairly simple design taken from Dr. Prodanov's notes [13]. The design, seen in Figure 15, has 3 inputs. InVolt is the input voltage coming from buffer 2. Set controls when the circuit starts to track the input voltage. It tracks when Set goes high and stops when Set goes low. Reset controls when it resets the voltage on the capacitor. This is important so that the capacitor doesn't have any residual charge on it when a new voltage begins to be tracked. The circuit holds the voltage as long as Set and Reset both remain low. At no time should both Set and Reset go high at the same time.

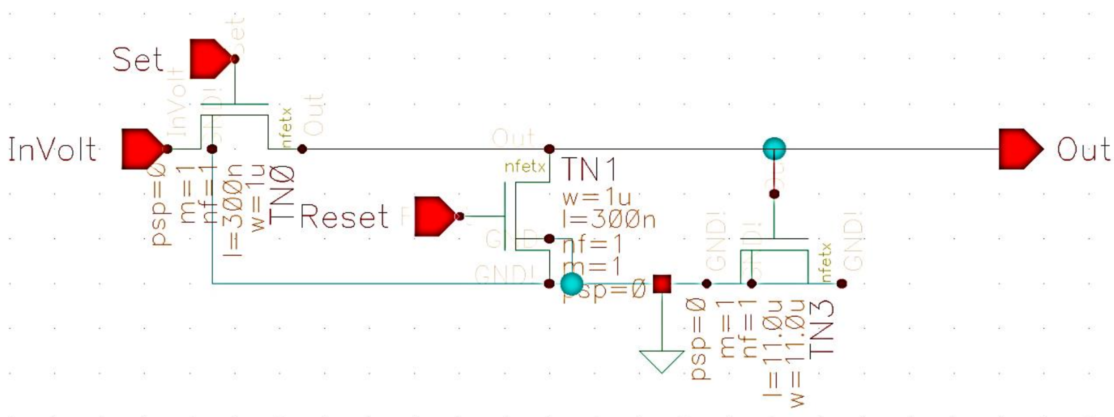


Figure 15: Track and Hold Schematic

Transistor TN0 acts as a switch, only allowing current to flow when Set goes "high" charging up TN3, which acts as a capacitor. TN1 grounds both ends of TN3 and thus discharging it when Reset goes "high." The size of both TN0 and TN1 should allow for enough current to flow that can charge and discharge TN3. A decent sized W/L is important but it does not need to be overdone as it might cause charge injection and pedestal error. The W/L ratio of TN3 derives from the sampling speed of the ADC. It should be large enough that it doesn't dissipate more than an LSB worth of voltage before the next sample is taken. This is important because if it does dissipate more than an LSB this could cause DNL and INL errors due to the voltage changing during the SAR algorithm.

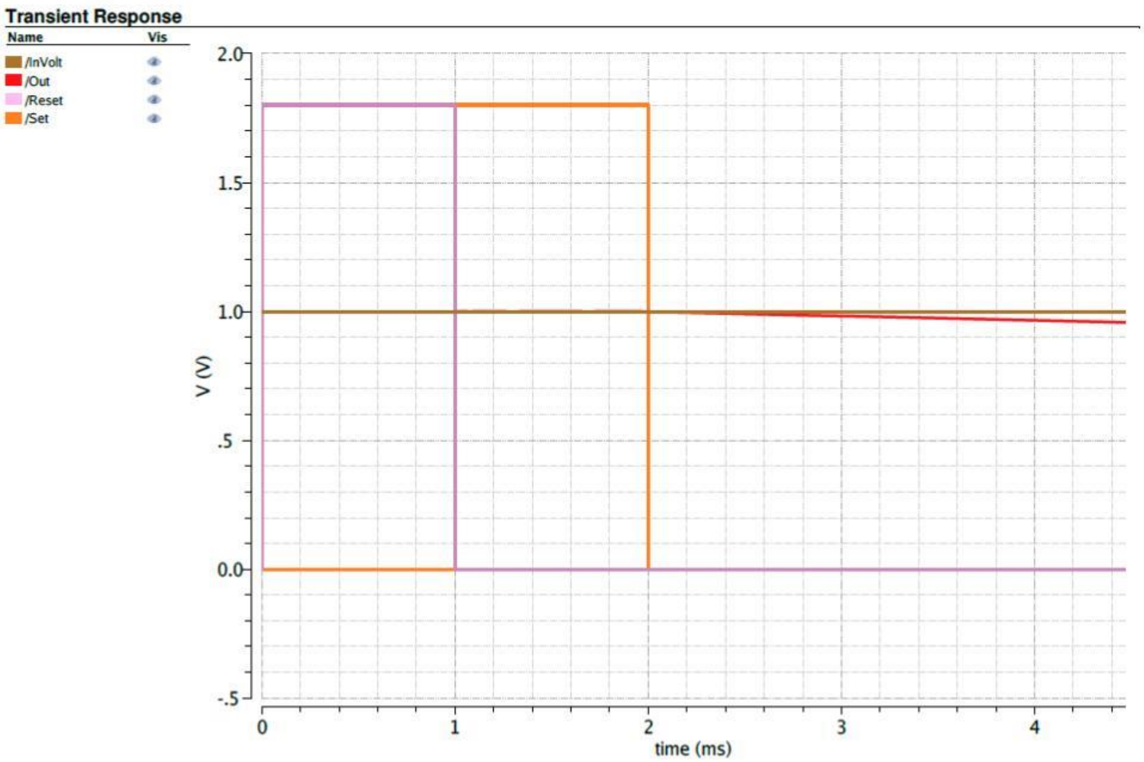


Figure 16: Track and Hold Voltage Dissipation Test

Figure 16 shows the voltage dissipation of the track and hold after a reset and then set cycle. While it looks like it takes a while to noticeably decline in voltage it doesn't take long for it to drop a LSB worth of voltage. With a sample speed of 200k samples/second the minimum time it must hold above a drop of 1 LSB is 5µsec. With this in mind TN3 is adjusted until it can hold that minimum without creating too large of a capacitive load for the voltage buffer. Table VII shows the results with the W/L ratio seen in Appendix B Table B3.

TABLE VII
SPECS FOR TRACK AND HOLD

Specification	Data
Time before the voltage drops 1 LSB	9.895µsec

With this timing spec, the track and hold has no problem keeping the voltage above an LSB in the time frame. While this is nearly double the minimum it ensures that there will be far less than 1 LSB of error introduced to the system.

Voltage to Current Converter

Due to the nature of the DAC a method of comparing the current it produces and the voltage on the input provided an interesting challenge. We resolve this by converting the input voltage to current and then making a direct comparison in the DAC (explained in the DAC section). Figure 17 shows the voltage to current converter design.

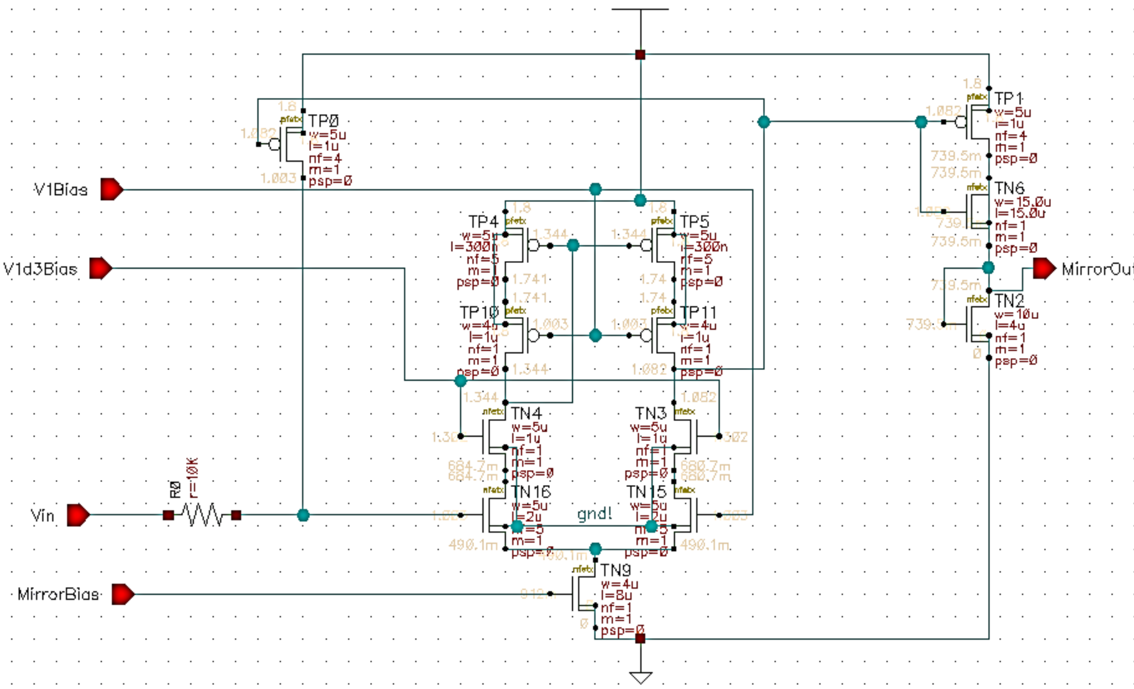


Figure 17: Voltage to Current Converter Schematic

The design utilizes a telescopic op-amp architecture [4] with its positive input connected to a 1V bias, then negative input connected just above the resistor R0, and the output connected to the node between TP0 and TP1. The current mirror in this design acts like a second stage for the telescopic op-amp. Due to this combination acting like a two stage op-amp some capacitive feedback was needed to increase its stability. This comes from TN6 which is setup as a capacitor.

The telescopic op-amp uses cascoded transistors TP10, TP11, TN4, and TN3. These allow for the op-amp to increase its inherent gain. Large gain proves important to this design because the linearity of the output depends on the gain of the telescopic op-amp. The gain of the op-amp directly corresponds to the accuracy of the conversion in its lower range. The larger the gain the more accurate the conversion is. TN4 and TN3 use a 1.3V bias and TP10 and TP11 use a 1V bias. Obtaining these values comes from keeping it within the 1.8V range and adjusting both W/L and the bias until the maximum amount of gain results. The positive input of the op-amp also connects to the 1V bias. Because of how the op-amp works, it places 1V on the node between R0 and TP0. Thus a linear current is produced proportional to input voltage due to the voltage drop across the resistor.

R0 in this design can come from either an internal resistor or an off chip resistor. Whether or not the resistor sits on or off chip depends on the size of the chip that the design resides. A 10kΩ on chip resistor takes up a substantial amount of room but then allows for more control over design variation. The process used for this project would require the resistor to sit off chip.

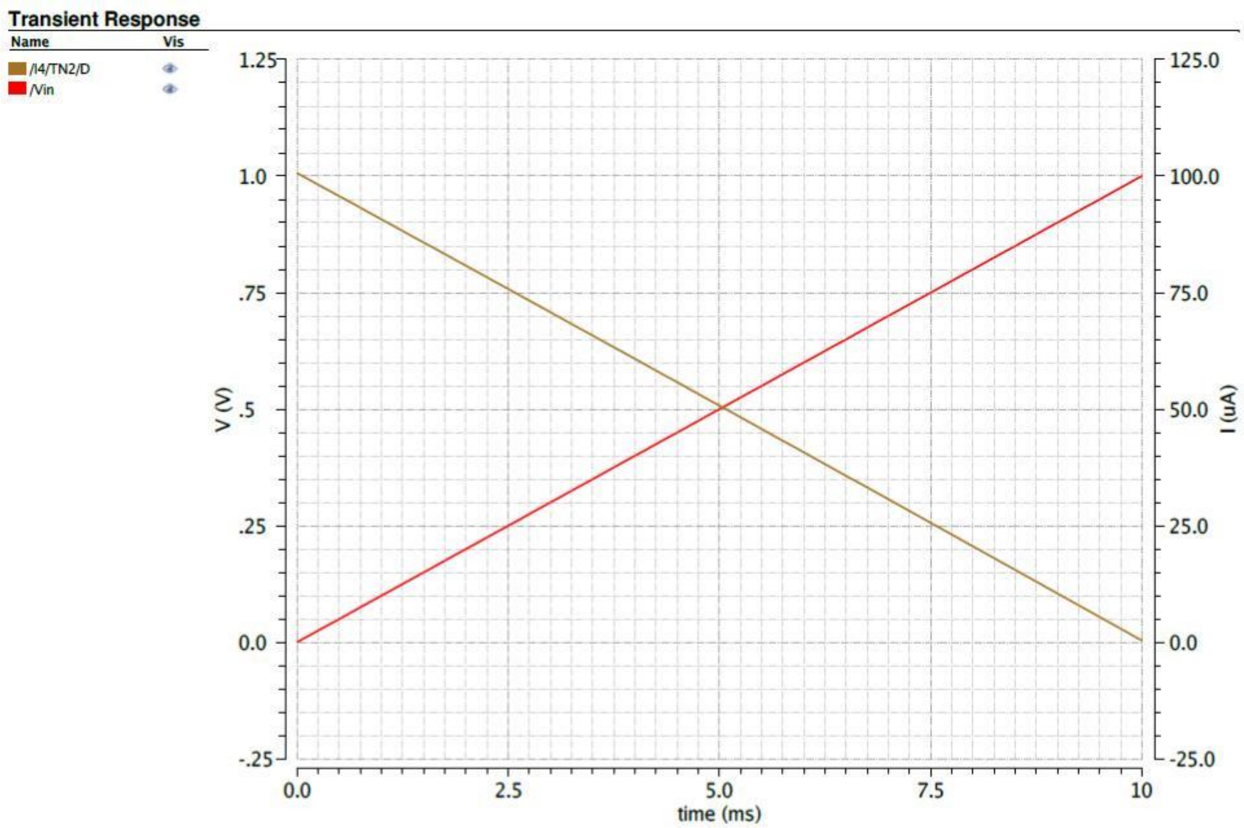


Figure 18: Voltage to Current Linearity Test 0 to 1V Input

Figure 18 shows the relationship between current and voltage. Table VIII shows the slopes of each line. While the slopes do depend on the time scale of the test it does give a general idea of how accurate the conversion is where ideally the current slope would be 10mA/s. Note that this test uses an ideal resistor model for R0.

TABLE VIII
SLOPES OF VOLTAGE AND CURRENT

Line	Slope
Current	9.9283mA/s
Voltage	100V/s

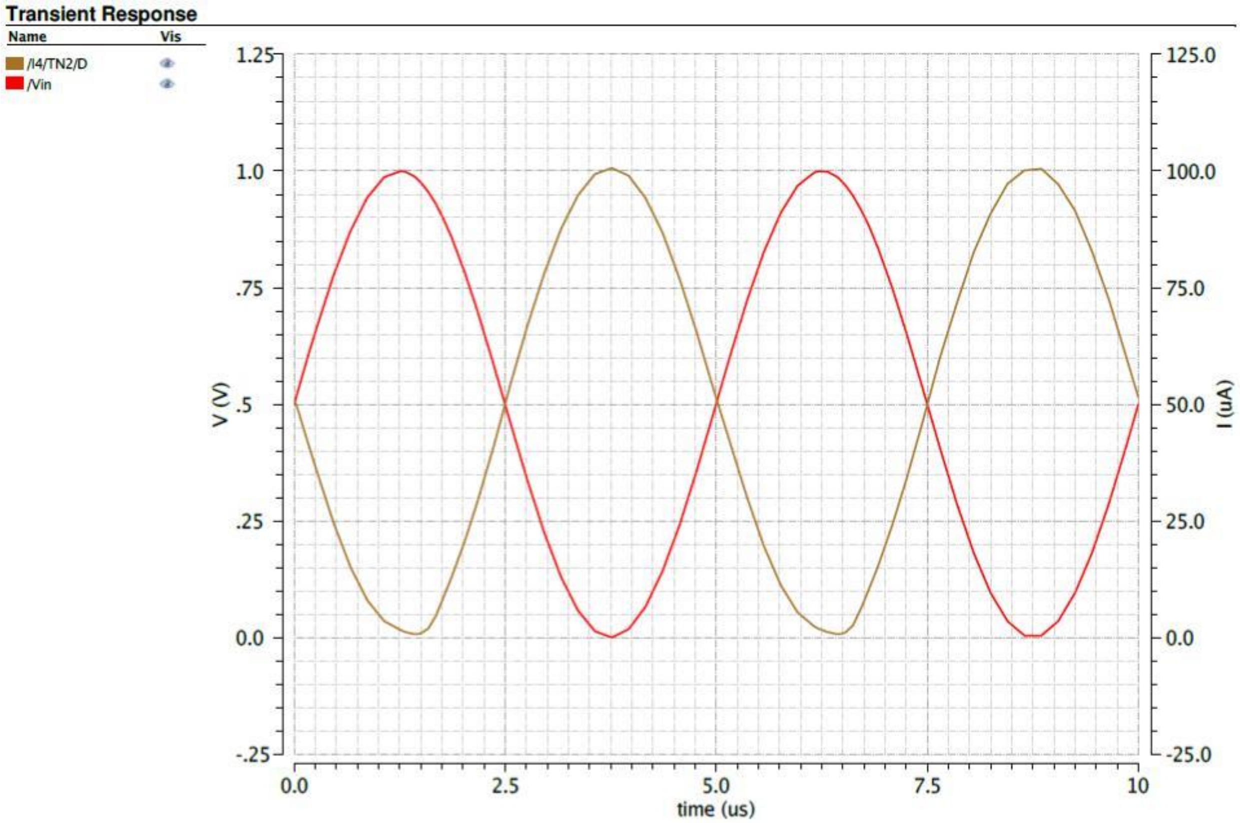


Figure 19: Voltage to Current Distortion Test at 200 kHz (Brown = Current, Red = Input Voltage)

Figure 19 shows the conversion happening at a faster speed. Some obvious distortion near the bottom of the current's wave form shows that the design has some areas for improvement. The main area for improvement revolves around the telescopic op-amp. Possibly using an op-amp with larger gain could create a faster and more accurate design.

Digital to Analog Converter

The design of the digital to analog converter (or DAC) revolves an R-2R style of DAC but instead of using resistors it uses transistors to create binary weighted currents. The idea comes from [6] and [7]. The basic idea is that the DAC uses a ladder of transistors using a unit value for the W/L ratio and alternating between W/L and 2W/L. The values used in this design are seen in Table B5. This creates currents that divide the previous current by 2 in each branch thus creating binary weighted currents.

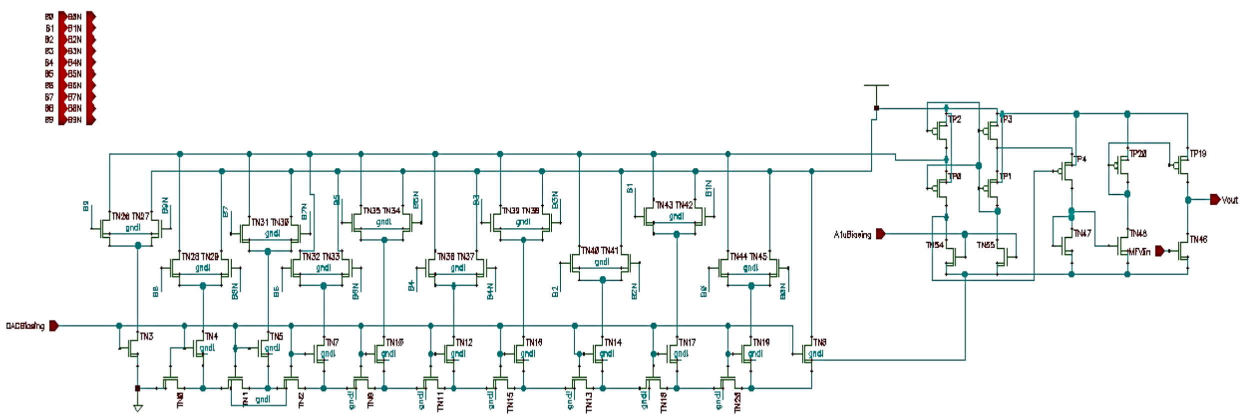


Figure 20: Digital to Analog Converter Schematic

When looking at Figure 20 all components to the left of the large “gap” make up the DAC with the transistors that look like diff-pairs making up the switching circuits. The transistors below the switching circuits make up the transistor ladder that produces the binary weighted currents. The reason for the switching comes from the need to have all of the currents constantly on for the ladder to work correctly. Thus any currents not being used by the DAC get redirected to a “dump” line connected to supply that does not affect the output of the DAC.

When designing the ladder, one of the most important factors to its accuracy is the W/L ratio used as the unit value. This design has a higher accuracy when lowering the W/L ratio by creating very long transistors in comparison to their width. This accuracy issues is noted in [6] and [7]. Though after a certain point of increasing the length of the transistor the increase in accuracy to the size of the transistor diminishes and so increasing the length only helps so much. Also making the transistors larger can help with process variation errors.

All of the transistors to the right of the “gap” make up a current mirror designed to reduce the load on the DAC based on Dr. Podanov’s current mirror design [14]. Due to the stacking of the transistor ladder and the low voltage range very little head room remains for a traditional current mirror. Dr. Prodanov’s current mirror design solves this problem. Without this design for the current mirror the current values that the DAC produces become extremely inaccurate.

Another process that occurs at this time is the current comparison. This comparison is done on the furthest right transistors (TP19 and TN46) seen in Figure 20. When more current is injected by the transistor on the top then the voltage raises at the output node between the two transistors. When the current injected by the bottom transistor (produced by the voltage to current converter) is higher than the current injected by the top transistor then the voltage at the output node decreases. If they are equal the node settles to about 1.14V. Thus this voltage of 1.14V represents the trip point that the comparator looks for to determine whether to increase or decrease the current that the DAC produces.

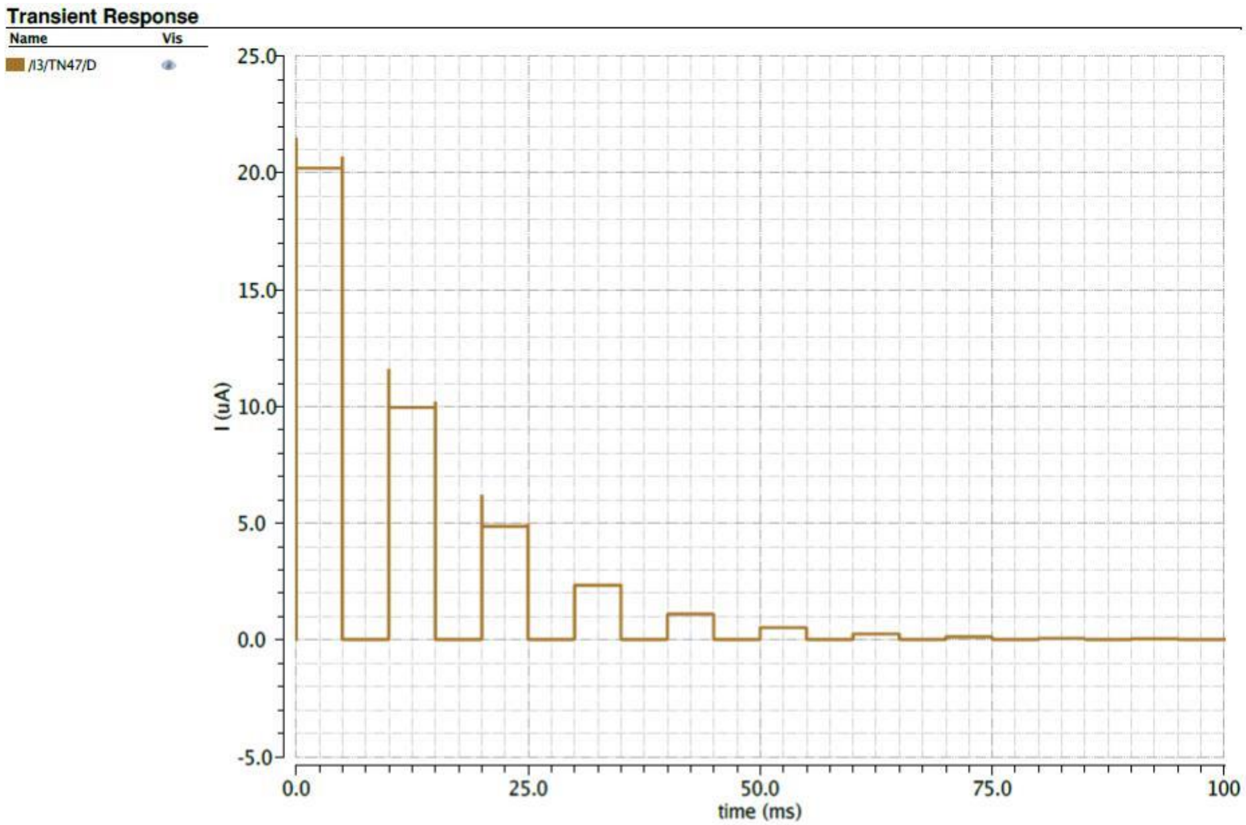


Figure 21: DAC Individual Bit Binary Weighting Test

Figure 21 shows a test run of the DAC switching process where the first value seen is the MSB and the last is the LSB with resets in between each value. Table IX shows all of the measured values from Figure 21. Something to note at this time is that when doing this test initially a large amount of variation was seen when each current value was held. After a substantial amount of time spent trying to resolve the problem I determined that the issue came from the software and not the design. It is extremely important that the simulation is “told” to use a restrictive algorithm rather than a liberal one. When using a liberal algorithm the program has trouble finding the exact current that the DAC produces and thus creates variation.

TABLE IX
DAC CURRENTS AND CORRESPONDING VOLTAGES

Measured Current	Ideal Current	Corresponding Voltage
20.20696µA	20µA	MSB: 500mV
9.959278µA	10µA	250mV
4.865238µA	5µA	125mV
2.340269µA	2.5µA	62.5mV
1.112294µA	1.25µA	31.25mV
0.5295345µA	0.625µA	15.625mV
0.2577114µA	0.3125µA	7.8125mV
0.1321306µA	0.15625µA	3.90625mV
0.07452122µA	0.078125µA	1.9531mV
0.04824886µA	0.039063µA	LSB: 0.97656mV

Table IX compares the measured current to the ideal current values and the corresponding voltages that each current represents. The first value in the table represents the MSB and the last value represents the LSB.

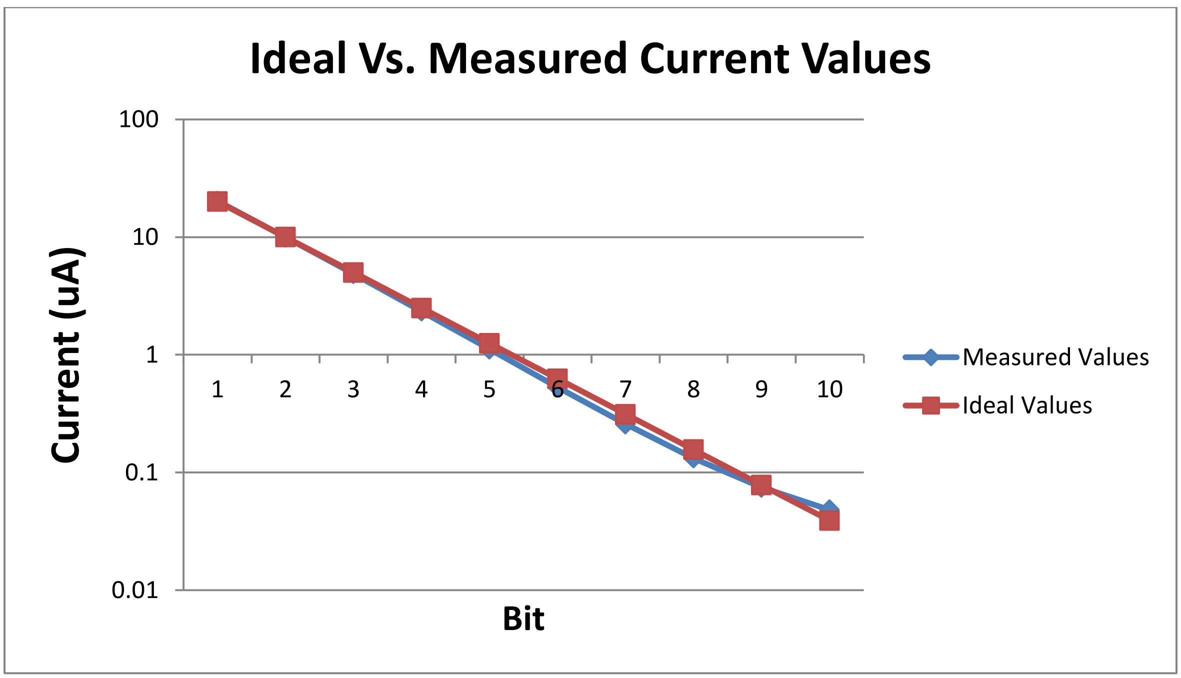


Figure 22: Logarithmic Comparison between Ideal DAC Current Values to Measured Current Values

Figure 22 compares the ideal and measured currents showing some of the error near the bottom of the DAC Values. Note that because this graph uses a logarithmic scale it does not show the severity of the error at each bit but rather how closely the trend follows an ideal set of values.

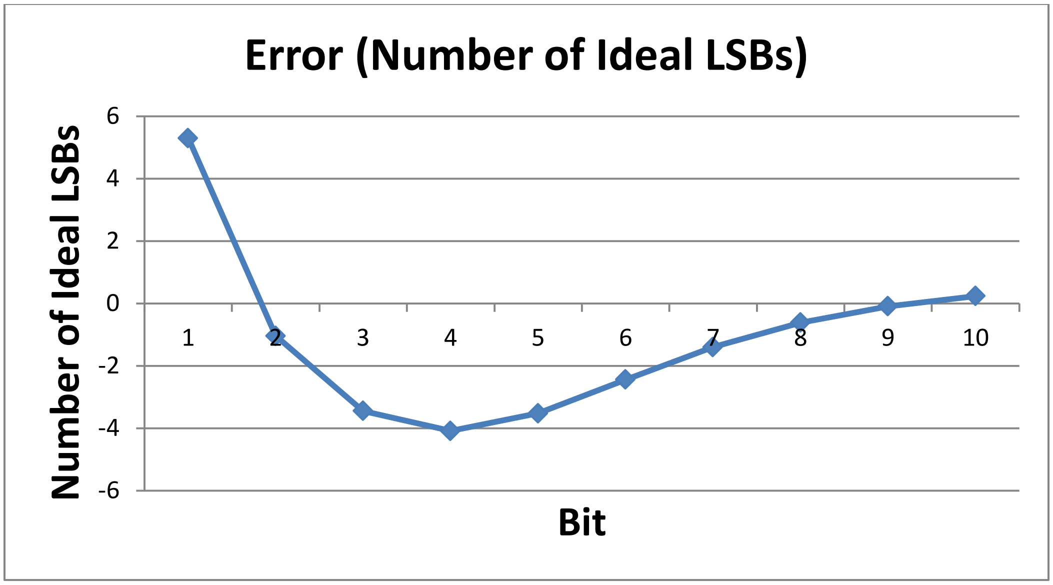


Figure 23: Error of Each Bit in terms of Number of Ideal LSBs.

Figure 23 gives a better representation of the error associated with each bit than Figure 22 gives. It shows the difference between the ideal values and the values measured in the simulation in terms of the number of ideal LSBs worth of difference that the DAC creates. Overall the data tends to sit just under the ideal values; a trend that gets represented in the overall error of the DAC seen in Table X. The DAC has some large non-linearity associated with the spread of the error. This in turn would show up as DNL and INL errors associated with the ADC. The main problem with this is that because the MSB value sits so much larger than the total sum of all the other bits that large code jumps would show up anytime the MSB gets switched and thus causing DNL error.

TABLE X
DAC TOTAL CURRENT COMPARISONS

Measured Current	Ideal Current	Error (Number of Ideal LSBs)
39.526186μA	39.96094μA	-11.1295

Table X shows the total error accumulated over the DAC. While the difference between the ideal total and measured total comes out to just above 0.4μA this comes to a total of 11 LSBs worth of error. While this seems large and in most cases such an error would prove unacceptable, for this design this is fairly good. One has to take into account various factors such as the scale of currents used. The LSB has a value of 48nA and ensuring accuracy on this level proves extremely difficult. One way to fix some of this error is to increase the current being supplied to the DAC and thus increasing the scale but one should use caution when doing this because it could cause the ADC to consume a substantial amount of current. That brings us to one of the main issues with this design. This design always consumes the max amount of current associated with the DAC where other designs such as switched capacitor only uses only uses enough current to charge the capacitor currently in use. One method that could fix the error associated with this design could come in the form of a digital correction method. It would require some alteration to the design but overall alter very little to the functionality of the DAC.

Comparator

This project uses a dynamic comparator as its comparator. A dynamic comparator is a comparator that uses a clocked reset. This means it does not constantly compare the values on its input, rather it compares only after a reset pulse goes from high to low.

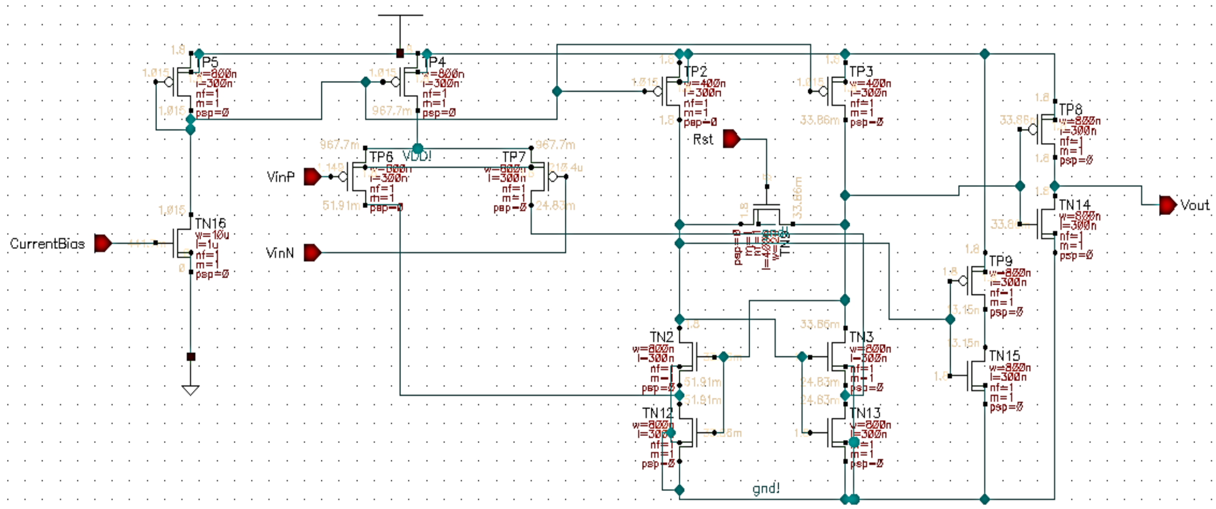


Figure 24: Dynamic Comparator Schematic

Figure 24 shows the design for the dynamic comparator. This design works by creating a difference in current created by the diff-pair (TP5 and TP7) injected into the node between TN2 and TN12 and the node between TN3 and TN13. These four transistors make up a latching circuit and depending on which branch has more current determines whether the comparator latches "high" or "low". The reset works by merging the two nodes where the latching occurs thus removing the latched value. TP8, TN14, TP9, and TN15 make up two inverters. These act as digital buffers for the comparator. These buffers preserve the current differences in the comparator so that the output load doesn't affect the results by consuming current from one side. TP9 and TN15 specifically act as a dummy load to create a balance to the design. Most of these transistors use smaller W/L ratios because they interface with digital circuits and thus don't require as much current. Transistors such as TN9 should be large enough to quickly reset the nodes by merging their currents. This design comes from both [8] and some ideas presented by Dr. Prodanov.

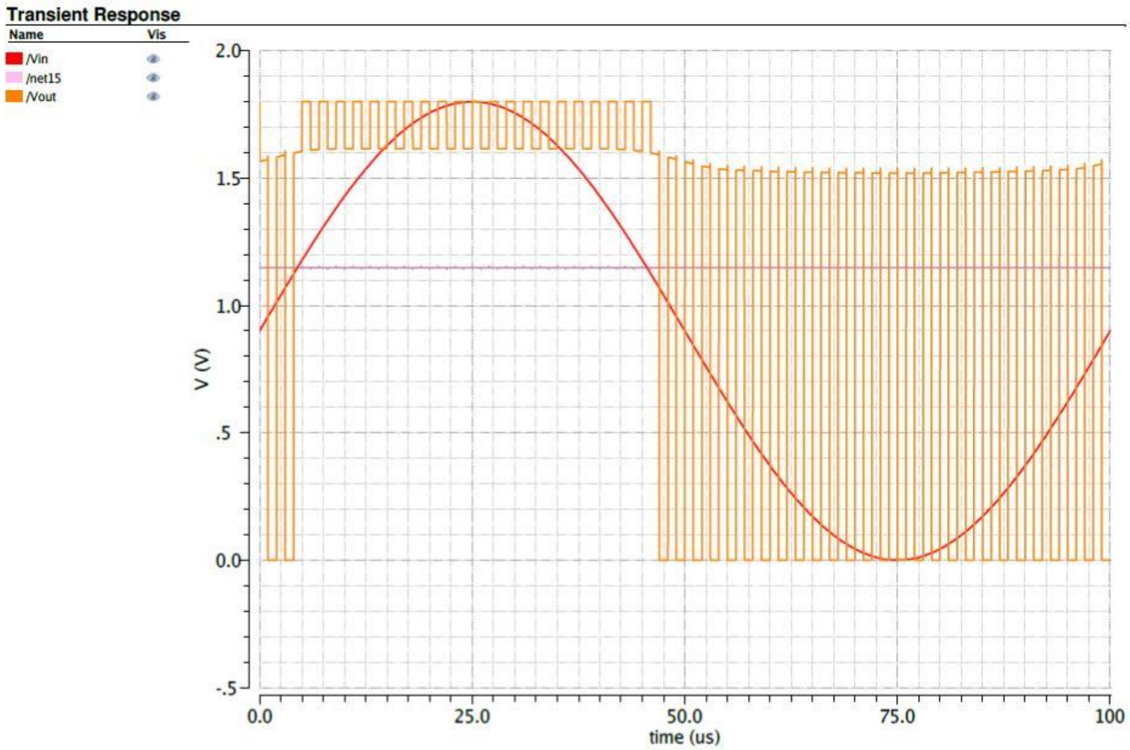


Figure 25: Comparator Functional Test, Input 0 to 1.8V with 200 fF Load

Figure 25 shows the comparator working in the same way it would work in the design. One signal is a constant voltage at about 1.14V while the other value comes from the DAC. One of the major advantages of this design is that it is a rail-to-rail design meaning that it can compare values that have a 0V to 1.8V range. Note that when the comparator outputs about 1.6V the comparator is in a reset cycle. This floating point is inherent to the design and its “high” value is partly due to the inverter stage.

TABLE XI
COMPARATOR SLEW RATE

Test	Results
Slew Rate (200fF load)	100V/μs

Table XI shows the slew rate for the comparator under a 200fF load. 200fF actually may be much larger than a typical load that this comparator would see because it interfaces with the digital controls so it would actually prove much faster than that.

Some sensitivity testing of this circuit also needed to be performed. The important part to test comes from how much current it takes for the latching mechanism to work properly. Figure 26 shows a test performed to test this spec.

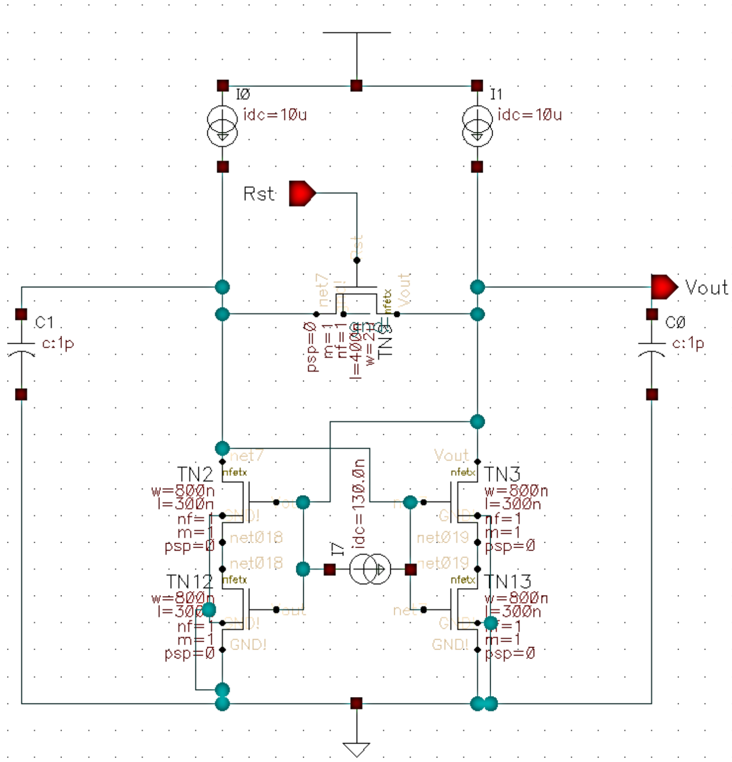


Figure 26: Comparator Second Stage Current Sensitivity Test Schematic

Note that this test just controls the injected current into the latching mechanism and because of using ideal current sources it performs slightly differently than how it does in the full design. The benefit of doing the test this way is that it isolates the latching circuit so that the properties of the latch can be better understood. An example of a waveform produced from this test is seen in Figure 27.

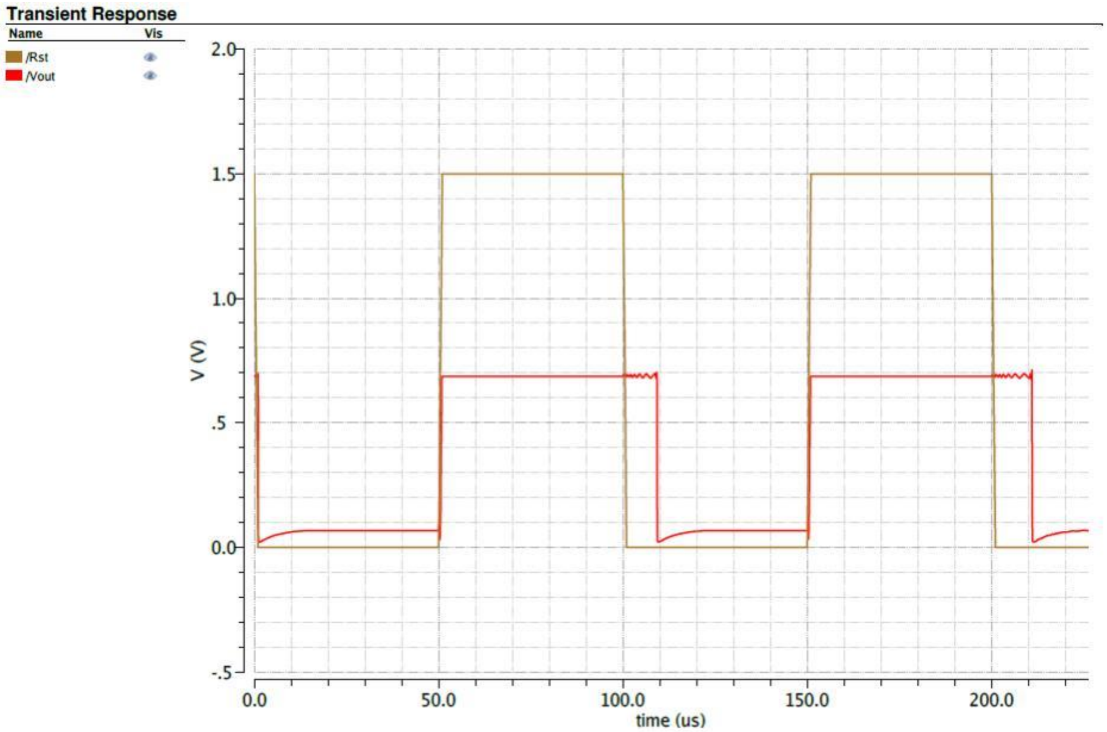


Figure 27: Comparator Second Stage Current Sensitivity Test Results for 130nA Differential

Note that the output of the circuit takes some time to pull low when 130nA is injected into the latch. Also note that this result does not look much like the full circuit results seen in Figure 25. This once again comes from the fact that this uses just the latching circuit and works much better as a whole circuit. By operating this test at several currents Table XII was produced.

TABLE XII
COMPARATOR SENSITIVTY

Current Injected	Settling Time
150nA	< 1µs
140nA	5µs
130nA	10µs
120nA	Tests Fails

As seen in Table XII the comparator works well as long as the difference remains above 150nA. Even when the two inputs have nearly the same voltage on them the difference in current produced between them is well above this value. Note that at 120nA the test fails. This is due to the ideal current sources allowing large voltages to form in the circuit. The test failing has more to do with the setup rather than the circuit itself.

Biasing Circuit

The purpose of the biasing circuit is to create any DC voltages or currents needed throughout the ADC. The ADC for this project uses three different currents and three different voltages for various biases and references. The circuit in Figure 28 creates these biases by using a self-biasing circuit taken from [2]. It uses various cascoded transistors and current mirrors to create a bias of about 1µA. Obtaining this value is a balancing act between the resistor value and the biasing transistors used for the cascodes. Because this circuit self-biases small changes in W/L ratios can create large changes in the biasing current. Also note that all other current mirrors that use this 1µA bias attach to the cascodes and not to the current mirrors used in the self-biasing circuits. Attaching to the current mirrors would produce the wrong current. Figure 28 shows the design of the biasing circuit

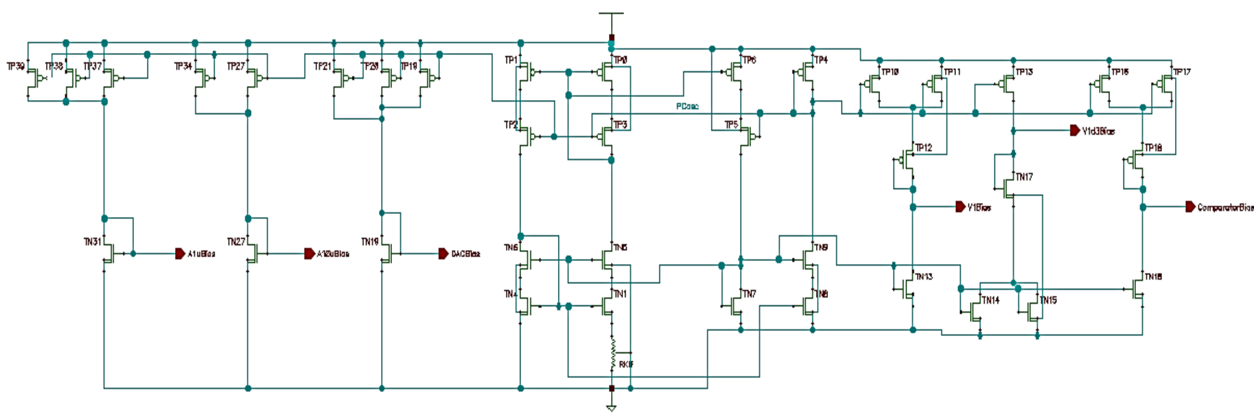


Figure 28: DC Biasing Circuit Schematic

All of the furthest left outputs make up the current biases of 1µA, 10µA, and 20µA (From left to right). All of the outputs furthest to the right make up the voltage biases of 1V, 1.3V, and 1.14V. At the center of theses outputs resides the self-biasing circuit. When creating the current and voltage biases a method of using multiple mirrors adding currents together allows for the creation of far more accurate current and voltage values. Accuracy for the voltages improves by including a diode connected transistor and adjusting its W/L ratio in tandem with adjusting the current mirrors to create accurate bias values.

An important factor to take into account when designing DC biasing curcuits is its temerature dependency. If the circuit cannot maintain relatively constant biases over a large temperature range then the accuracy of the ADC will depend far to much on the temperature that it operates at. Figure 29 shows some temperature tests done on this design.

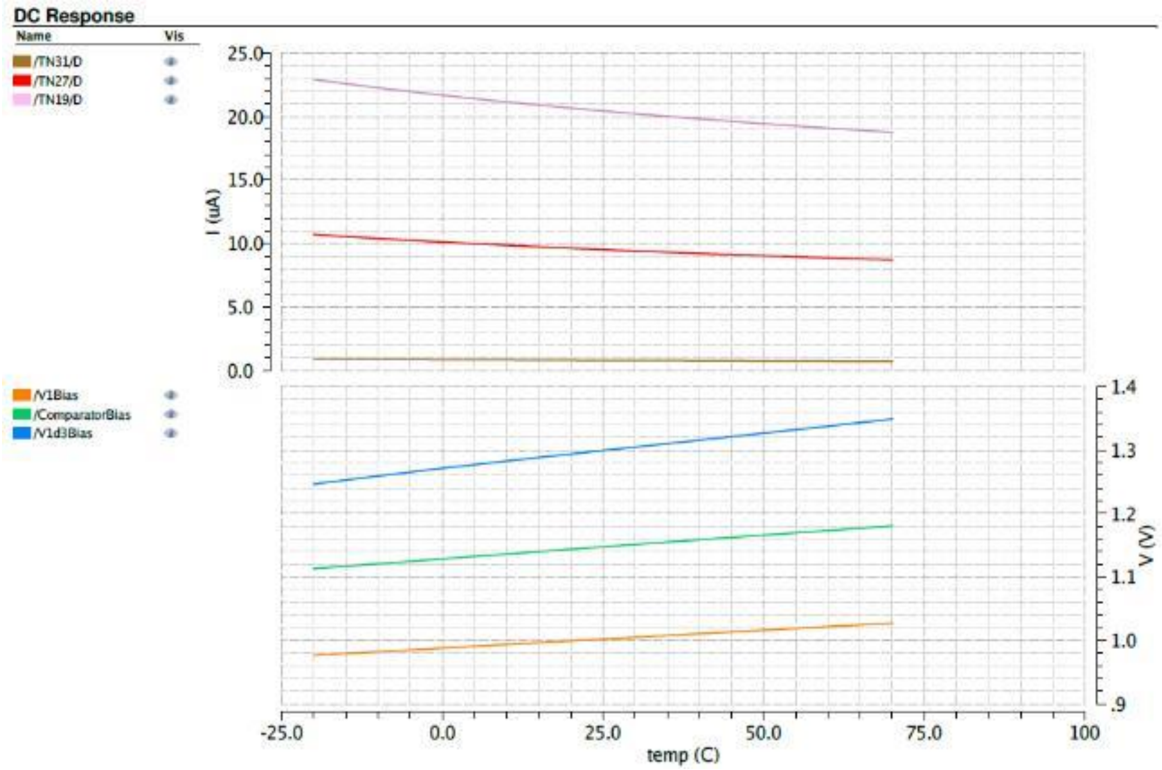


Figure 29: DC Biasing Temperature Sensitivity Test of -20˚C to 70˚C

This DC biasing design is a fairly simple way of creating biases thus it does not focus much on correction for temperature variation. Note that the use of an NPN diode connected transistor has a larger range of variation than a PNP diode connected. This flaw remains to show some possible design changes that could result in smaller variation in values over temperature. Overall the variation in the biases falls outside of what one would consider acceptable ranges of error. In the case of the voltage to current converter huge amounts of error would accumulate due to the sensitivity of the circuit if the chip operates anywhere near the extremes of these temperature variations. Table XIII shows the ranges and the percent change from the desired value that those ranges are associated with.

TABLE XIII
COMPARATOR SENSITIVTY

Bias Current or Voltage	Range (-20°C to 70°C)	Percent Change from Desired
1µA	915.0984nA to 728.1199nA	-8.49% to -27.188%
10µA	10.71914uA to 8.71914uA	7.1914% to -12.808%
20µA	22.95161uA to 18.7743uA	14.758% to -6.1285%
1V	976.3143mV to 1.027227V	-2.369% to 2.723%
1.3V	1.246393V to 1.349055V	-4.124% to 3.773%
1.14V	1.113078V to 1.180465V	-2.362% to 3.55%

There are some possible areas where this design could see improvement but to see a large improvement one should use some more complicated designs than this.

Full Analog Testing

With all of the analog components designed one can conduct some simple tests to see how well the devices function as a system. Often design flaws with individual components arise when performing these tests before introducing the digital components. One such example comes from the issue of loading with the voltage buffers. The issue of the capacitive load versus the resistive load didn't prove an issue until the test was conducted with the system as a whole. Figure 30 shows the test setup for the full analog test.

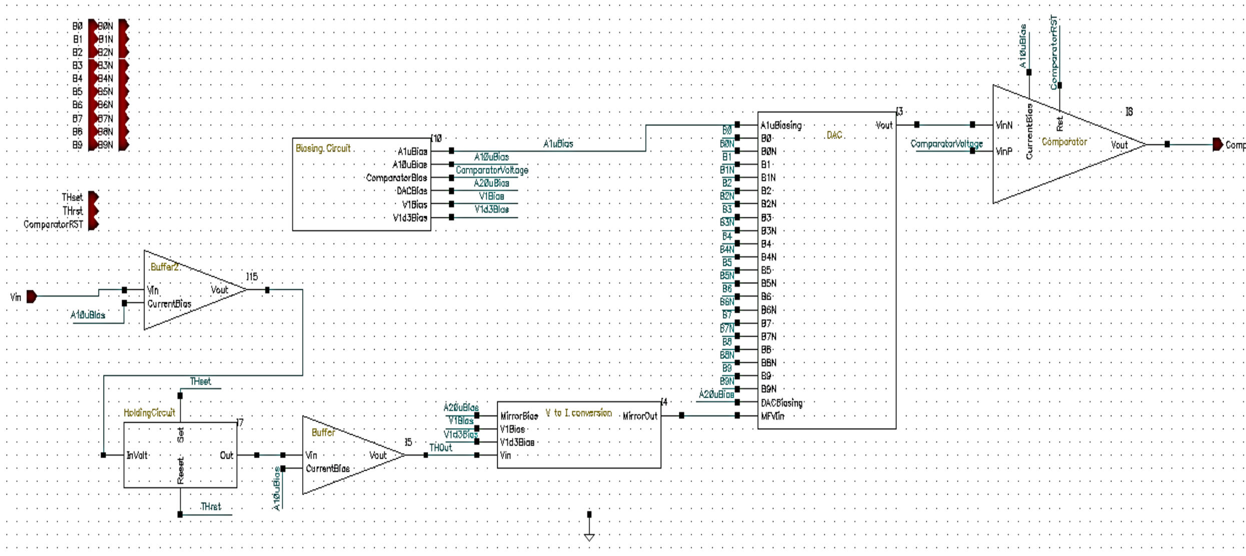


Figure 30: Full Analog Test Setup Block Diagram

Other issues that arose during this testing phase include interfacing the voltage to current converter to the DAC without causing it to load the DAC, some stability issues with the voltage buffers due to using a different load capacitor than the actual design would see, and adjusting the reference voltage of the comparator based on a 0.5V input until it switched at the right time. Results from one of the final tests are seen in Figure 31.

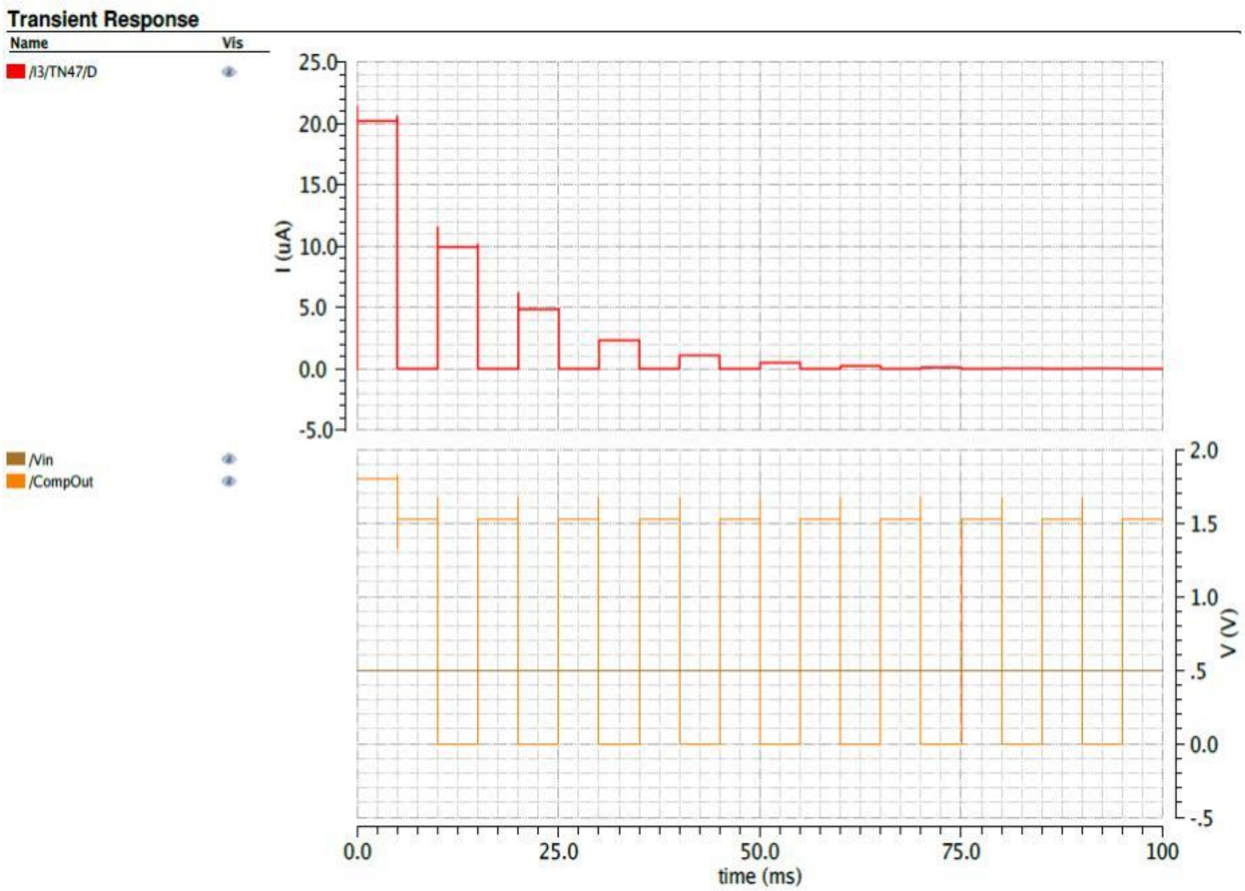


Figure 31: Full Test at a Clock Speed of 200 Hz with 0.5V input (Red = DAC Current, Brown = Voltage Input, Orange = Comparator Output)

Figure 31 shows one of the final tests performed on the analog components. The test shows that with a voltage of 0.5V on the input and when the DAC outputs its MSB that the comparator correctly switches “high.” Once the DAC drops past outputting just the MSB then every proceeding check gives a “low” output from the comparator, as would be expected in this test.

Note that the reason the test has a slow “Clock Speed” mainly has to do with how the test has to be setup. I chose this timing because it was easy to check. When testing without the digital controls every single line controlled by the digital block must be manually controlled as a pulse function in the simulation. Thus all of the timing must be done manually to ensure the test acts as it would with the digital block attached. Of course the test does not use the SAR algorithm and rather just turns on each current value individually.

Digital Controls

The digital controls make up the brain of the ADC. It tells all the different analog components when to turn on and off based on a set flow. In this case that flow comes from the Successive Approximation algorithm. Figure 32 shows a simple flow diagram showing the basics of how this design's code works. All diamond shapes make up decisions, all filled squares make up processes, and the unfilled square consists of the SAR algorithm sub-process. Appendix C contains all of the actual code used for this design.

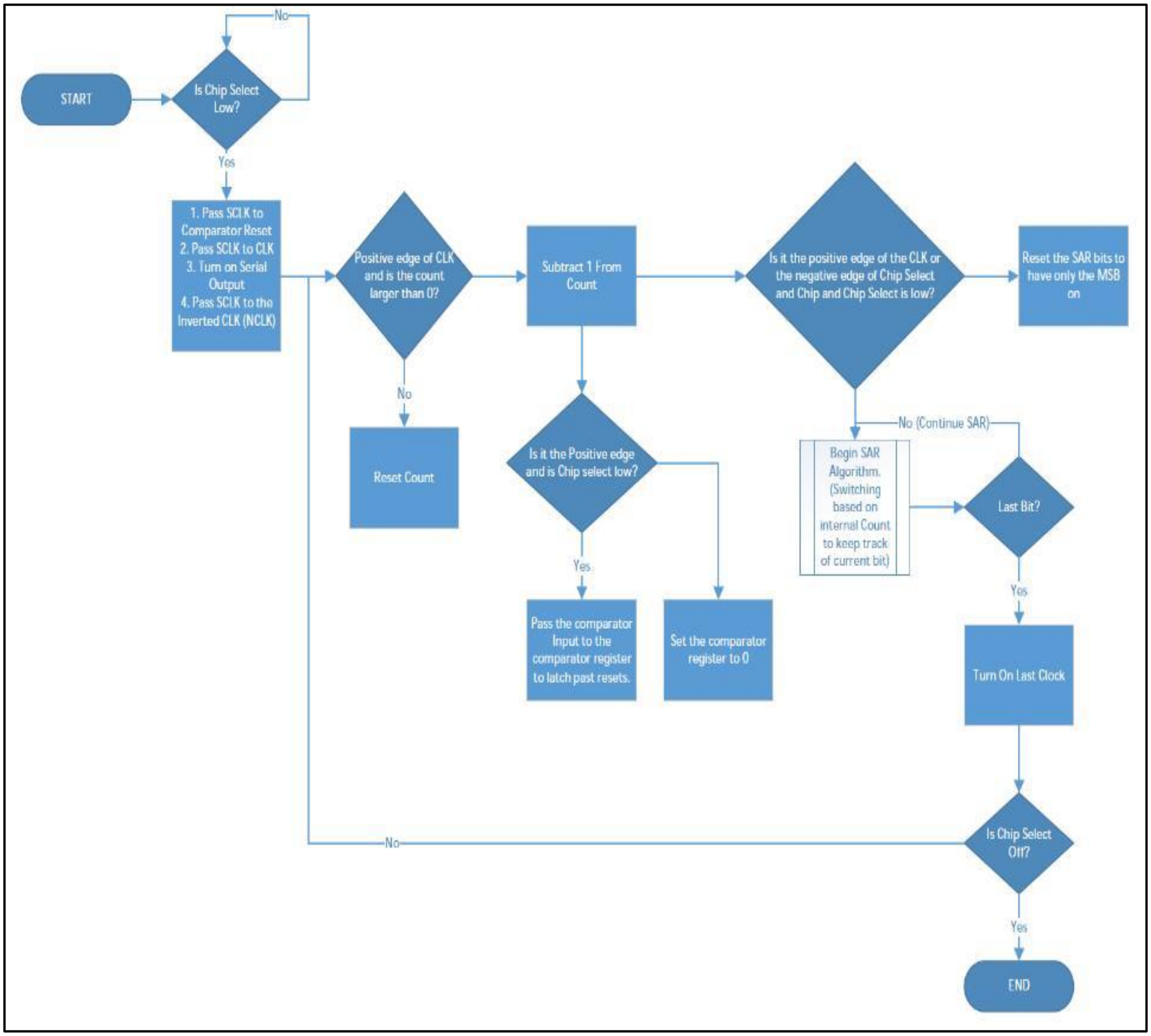


Figure 32: Digital Controls Flow Diagram

The flow diagram helps understand how the code decides which signals to turn on and off at certain times but the timing diagram seen in Figure 33 shows exactly when all the signals turn on and off.

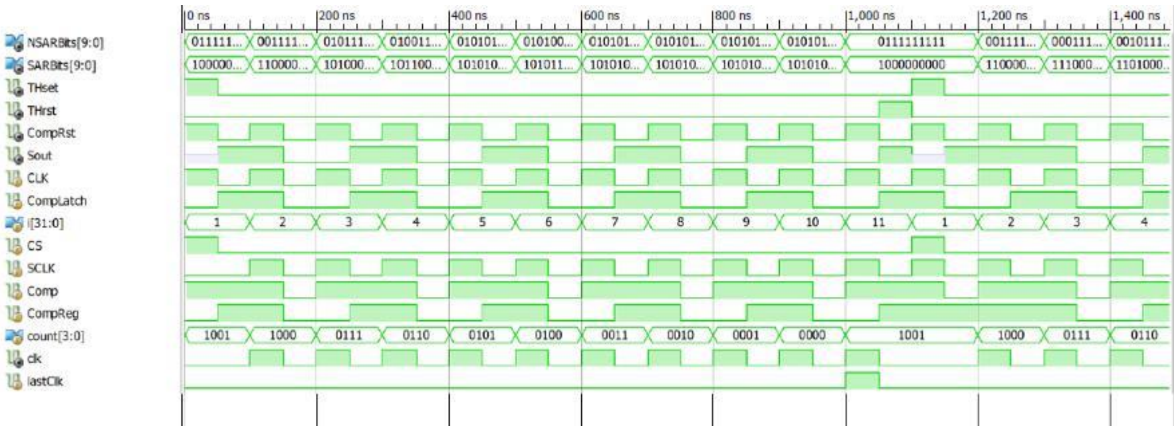


Figure 33: Timing Diagram for the ADC Digital Controls

The test performed uses a set of test bench code (seen in Appendix C) that responds to the digital controls and gives it feedback similar to how the analog components would give it feedback based on the input voltage. The timing diagram shows that the SAR algorithm correctly responds to the input from the comparator and turns on all the signals at the correct time. Table XIV shows what each signal in Figure 33 represents. Note that both Figure 33 and Table XIV have both test bench and ADC control signals.

TABLE XIV
COMPARATOR SENSITIVTY

Signal	Description
NSARBits[9:0]	This 10 bit bus controls whether or not the current in each DAC bit goes to the current dump line. Always the inverse of SARBits.
SARBits[9:0]	This 10 bit bus controls whether or not the current in each DAC bit goes to the output line.
THset	This signal controls the set in the track and hold.
THrst	This signal controls the reset in the track and hold.
CompRst	This signal controls the reset for the comparator.
Sout	This signal outputs the serial data.
CLK	This signal is the test bench clock.
CompLatch	This signal creates the random comparator outputs.
i[31:0]	This is the internal count for the test bench code.
CS	This is the Chip select signal controlled by the test bench code.
SCLK	This is the clock input to the ADC.
Comp	This is the comparator input signal to the ADC (includes resets).
CompReg	This latches the comparator output to avoid resets.
Count[3:0]	Internal Bit count for the SAR Algorithm.
clk	Internal Clock passes from SCLK.
lastClk	Gives another edge at the end of the algorithm to ensure SPI functionality.

One interesting concept about designing using SPI is the lack of external signals to trigger off of. A total of two external signals (Serial Clock and Chip Select) make up the signals that the ADC has to work with. This makes creating a large control system difficult to program to ensure all the timing of signals works correctly. One way that this design gets around that is by creating its own internal trigger signals such as NCLK and lastCLK which add edges to the system that can trigger certain events to happen. In some cases certain methods of coding seen in Appendix C can seem redundant but in some cases it proves better to stay on the safe side rather than have issues come up that were not accounted for during the design process. Also if the code proves truly unnecessary then the synthesis tool that turns the code into gates will optimize it out. This brings the design to the next step which involves taking the code seen in Appendix C and putting it through the synthesis tool. The results of the synthesis tool are provided in Figure 34.

Figure 34: Gate Diagram Produced by Synthesis

The synthesis tool attaches the technology library to the code producing gates and flip-flops that represent the functionality of the code that the synthesis tool receives. At this point many issues that may not arise during the initial testing may come up. One such issue associated with this design comes from the SPI design. Normally one would use a tri-state buffer on the output of the ADC so that it does not affect the data line when not in use. Unfortunately the technology library used for this project does not contain a tri-state buffer so in this case the design does not follow a true SPI format and the chip would not work with other devices connected to the data line.

After the synthesis process comes the place and route process using Encounter to turn the gates into a silicon layout (part of the Cadence design suite). This process uses premade silicon layouts for the gates and flip-flops to produce a compact design. By setting some parameters associated with the size of the design, power delivery layout, and clock timing the place and route produces an analog design for the digital code. For the most part this process involves just letting the computer do its own thing much like what happened during the synthesis stage. The results of this process can be seen in Figure 35.

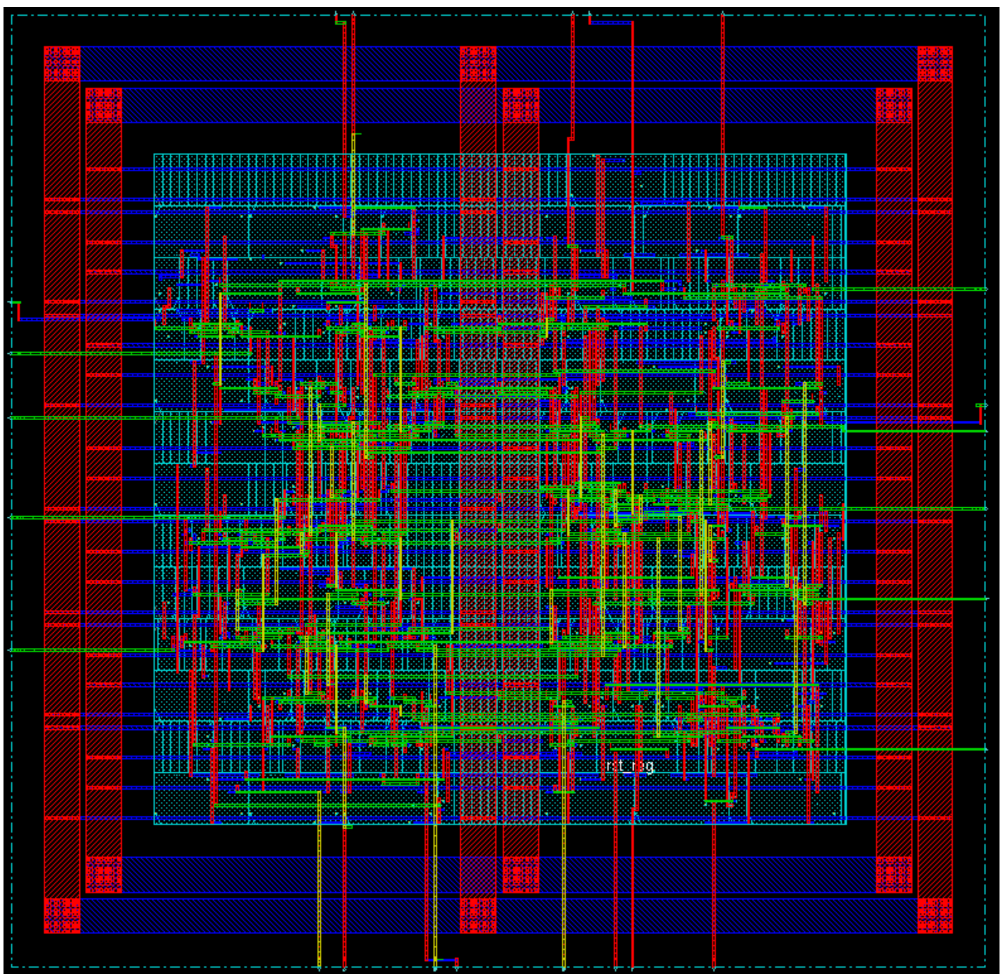


Figure 35: Transistor Layout of the ADC Digital Controls

At this point the project hit a road block. The issue was getting the silicon layout into the analog simulator so it could interact with the rest of the design to test timing and ensure that no functionality got lost from the code to the silicon layout. Unfortunately the programs had issues transferring the correct files and the problem could not be resolved in a timely manner thus ending the project prematurely.

Conclusion

As both a learning experience and an overall project I consider this effort a success. Unfortunately my project did not reach full functionality but that does not mean that beneficial results were not obtained. While several steps still remain including performing a test with both the digital and analog components together most of the designs are proven to work to a reasonable level with some minor design flaws. The next steps in this project would consist of finishing the full-system testing and completing the layout for all of the analog components. Subsequently Mosis could take the designs and manufacture a test chip that could be run through some DNL and INL testing to see how well this design could work as an ADC.

Through the process of this project I acquired a substantial amount of knowledge about transistor level design, especially its limitations and capabilities. I successfully produced components that in the end could meet all of the specification I gave save maybe some of the DNL and INL specifications with most of the error coming in on the low end associated with the voltage buffer. While the design does need a lot of work, overall the system performs function as intended.

References

1. Maxim Integrated. (2013, 19 October). *Understanding SAR ADCs: Their Architecture and Comparison with Other ADCs* [Online]. Available: http://www.maximintegrated.com/app-notes/index.mvp/id/1080
2. David A. Johns, Ken Martin, *Analog Integrated Circuit Design*, John Wiley &Sons Inc., 1997.
3. IEEE, (2013, 4 November) “IEEE Code of Ethics” [Online], Available: http://www.ieee.org/about/corporate/governance/p7-8.html
4. Mosis,(2013, 10 October) *The Mosis Service*, 2013 [Online], Available: http://www.mosis.com/
5. Digel, Johannes; Groezing, Markus; Berroth, Manfred, "A 9 bit 34 MS/s SAR Analog-to-Digital Converter in 130 nm SiGe BiCMOS," Ph.D. Research in Microelectronics and Electronics (PRIME), 2012 8th Conference on , vol., no., pp.1,4, 12-15 June 2012, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6226179&isnumber=6225736
6. Hammerschmied, C.M.; Qiuting Huang, "Design and implementation of an untrimmed MOSFET-only 10-bit A/D converter with -79-dB THD," *Solid-State Circuits, IEEE Journal of*, vol.33, no.8, pp.1148,1157, Aug 1998 doi: 10.1109/4.705353 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=705353&isnumber=15241
7. Linares-Barranco, B.; Serrano-Gotarredona, T.; Serrano-Gotarredona, R., "Compact low-power calibration mini-DACs for neural arrays with programmable weights," Neural Networks, IEEE Transactions on , vol.14, no.5, pp.1207,1216, Sept. 2003 doi: 10.1109/TNN.2003.816370 ,URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1243722&isnumber=27868
8. S. Hoyos. (2014, February 15). *CMOS Comparators*. Fall 2009. [Online], Available: http://amesp02.tamu.edu/~jsilva/610/Lecture%20notes/Comparators.pdf
9. Wikipedia, (2014, 16 April). “Serial Peripheral Interface Bus.” [Online], Available: http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus#Mode_Numbers
10. Arduino. (2013, 16 December). “SPI Library.” [Online], Available: http://arduino.cc/en/Reference/SPI
11. Cypress. (2014, May 23). “Definitions of INL and DNL in an ADC,” *Support & Community: Knowledge Base Article*. [Online], Available: http://www.cypress.com/?id=4&rID=32117
12. Texas Instruments. (2014, January 12) “TLC27M4, TLC27M4A, TLC27M4B, TLC27M4Y, TLC27M9 LinCMOS PRECISION QUAD OPERATIONAL AMPLIFIERS.” [Revised October 2012] Available: http://www.ti.com/lit/ds/symlink/tlc27m9.pdf
13. Prodanov, Vladimir. EE409 Lecture 22 through 24. Fall 2013.
14. Prodanov, Vladimir and Green, M. M. (2014, January 15) “CMOS current mirrors with reduced input and output voltage requirements.” Available: http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1136&context=eeng_fac
15. Smilkstein, Tina. “Computer-Aided Design of VLSI Devices.” EE431 Cal Poly Course. Fall 2013.
16. J. Holden, (2013, 3 November) “The Environmental Impact of the Manufacturing of Semiconductors” [Online], Available: http://cnx.org/content/m14503/latest/

Appendix A: Analog Circuit Schematics

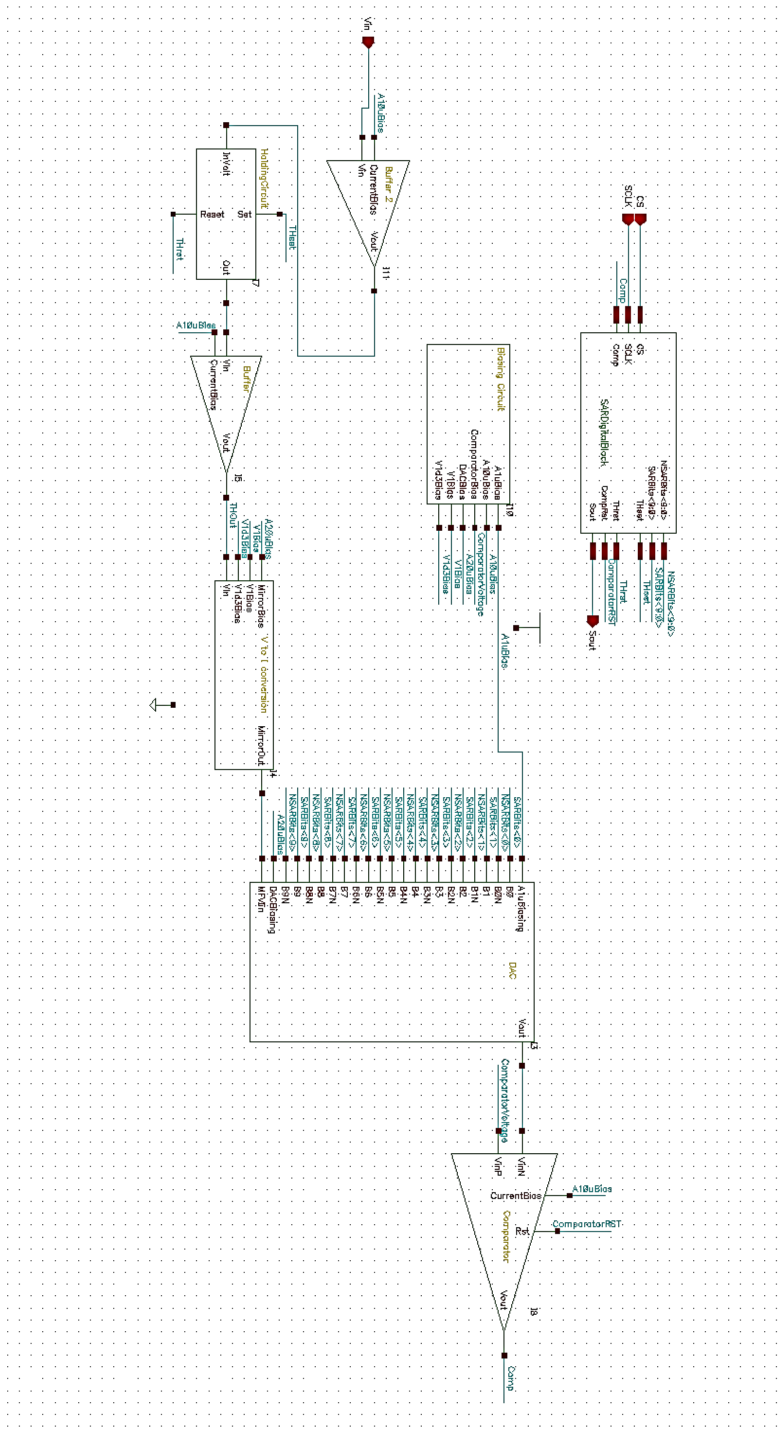


Figure A1: Full ADC Block Diagram Connections

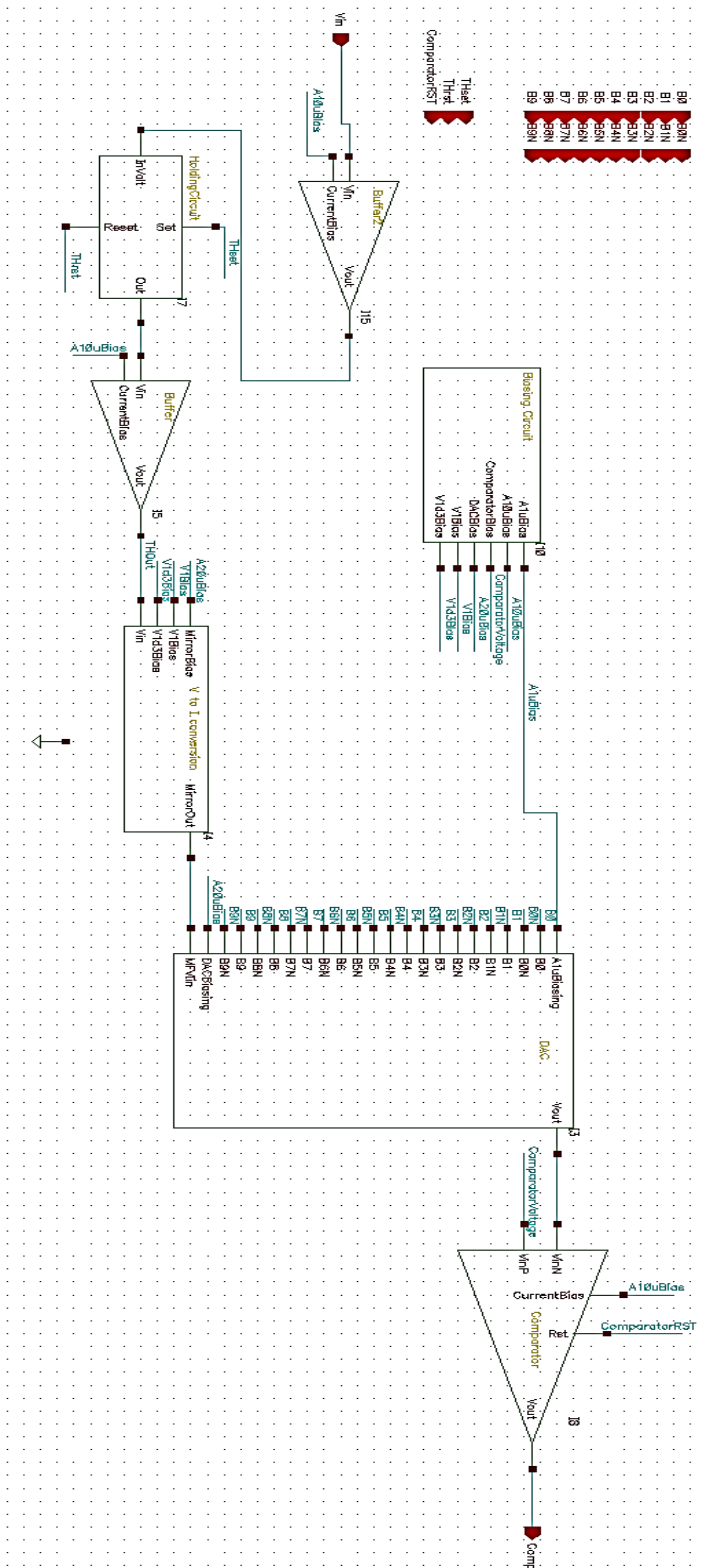


Figure A2: Full Analog Test Block Diagram

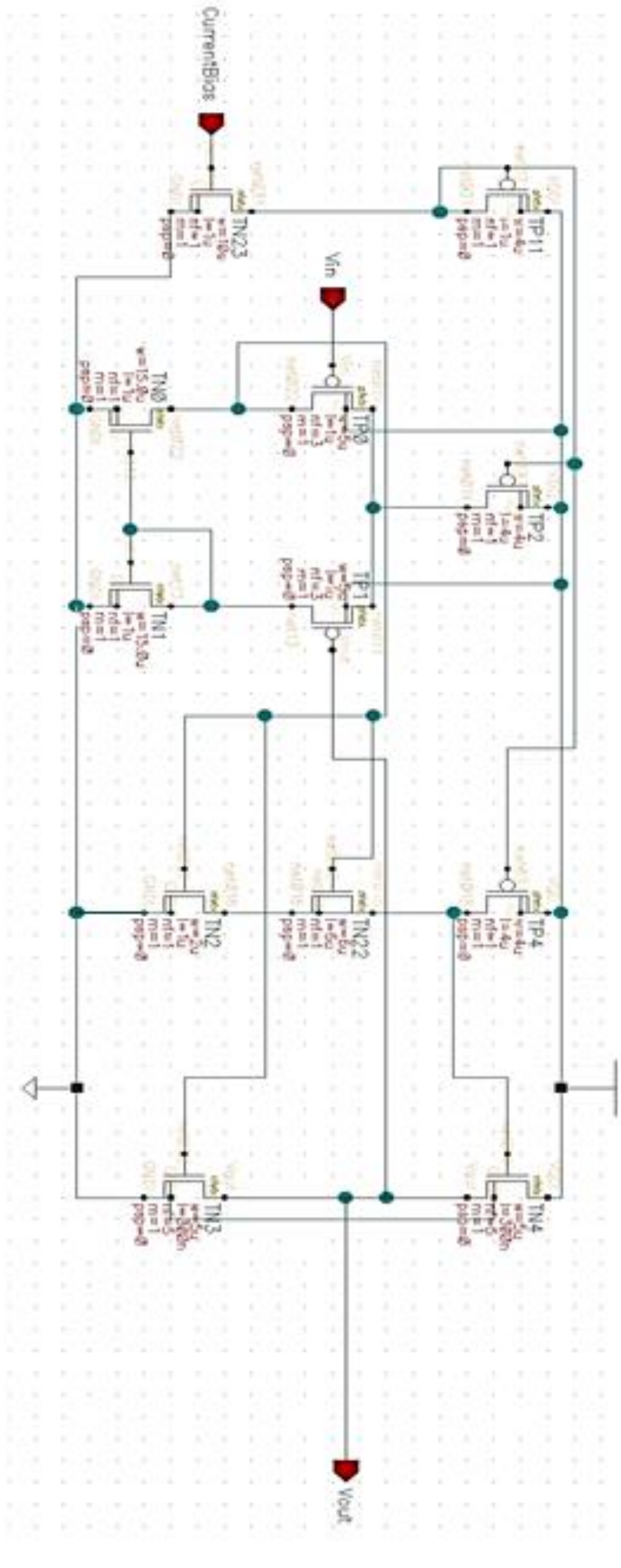


Figure A3: Voltage Buffer Circuit Schematic

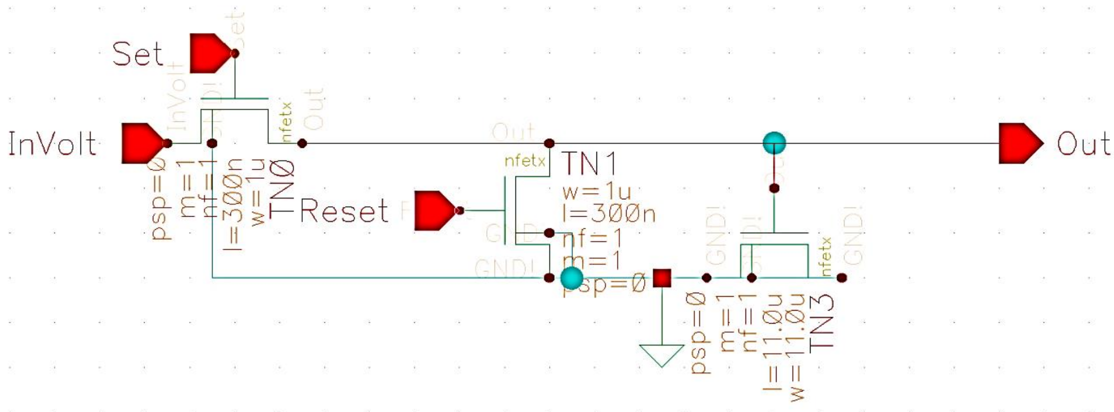


Figure A4: Track and Hold Circuit Schematic

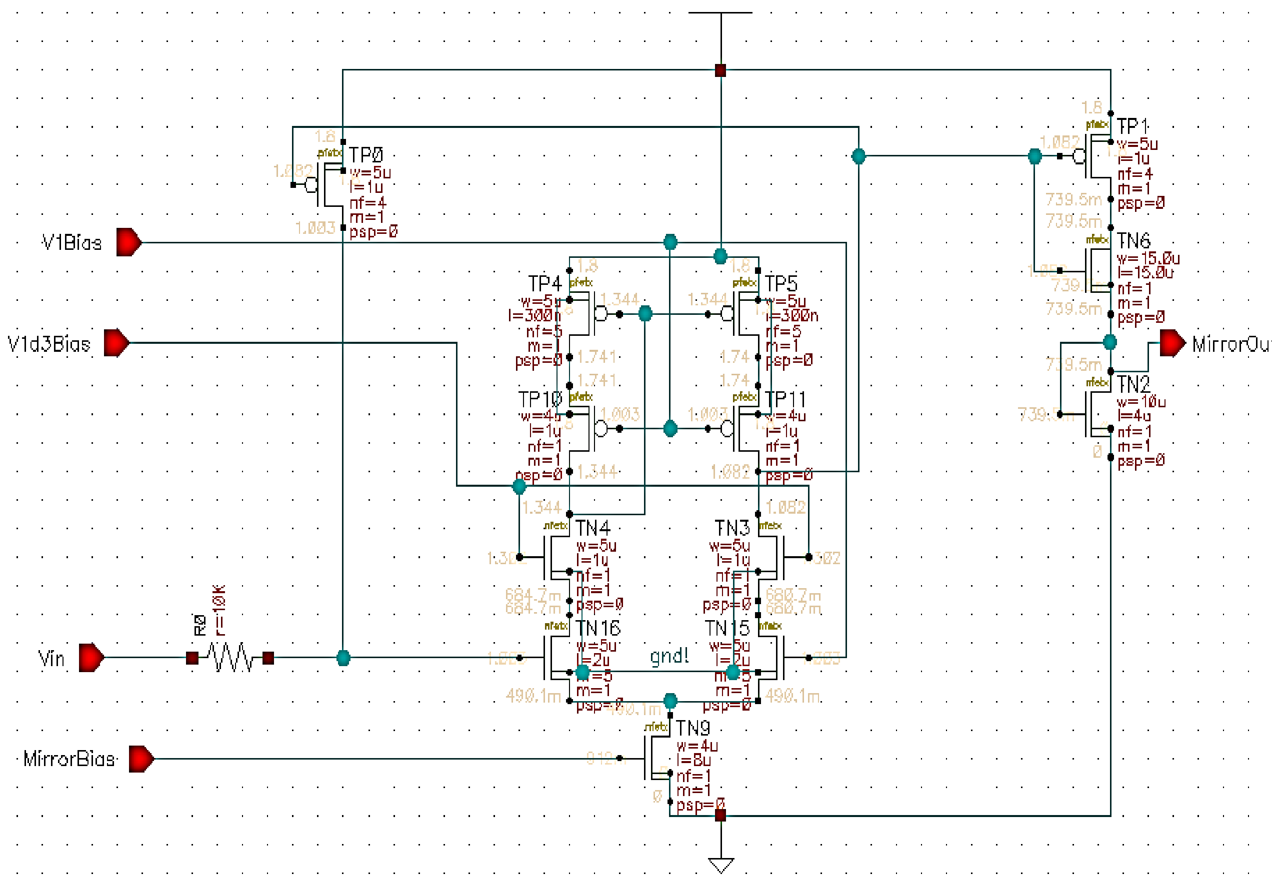


Figure A5: Voltage to Current Converter Circuit Schematic

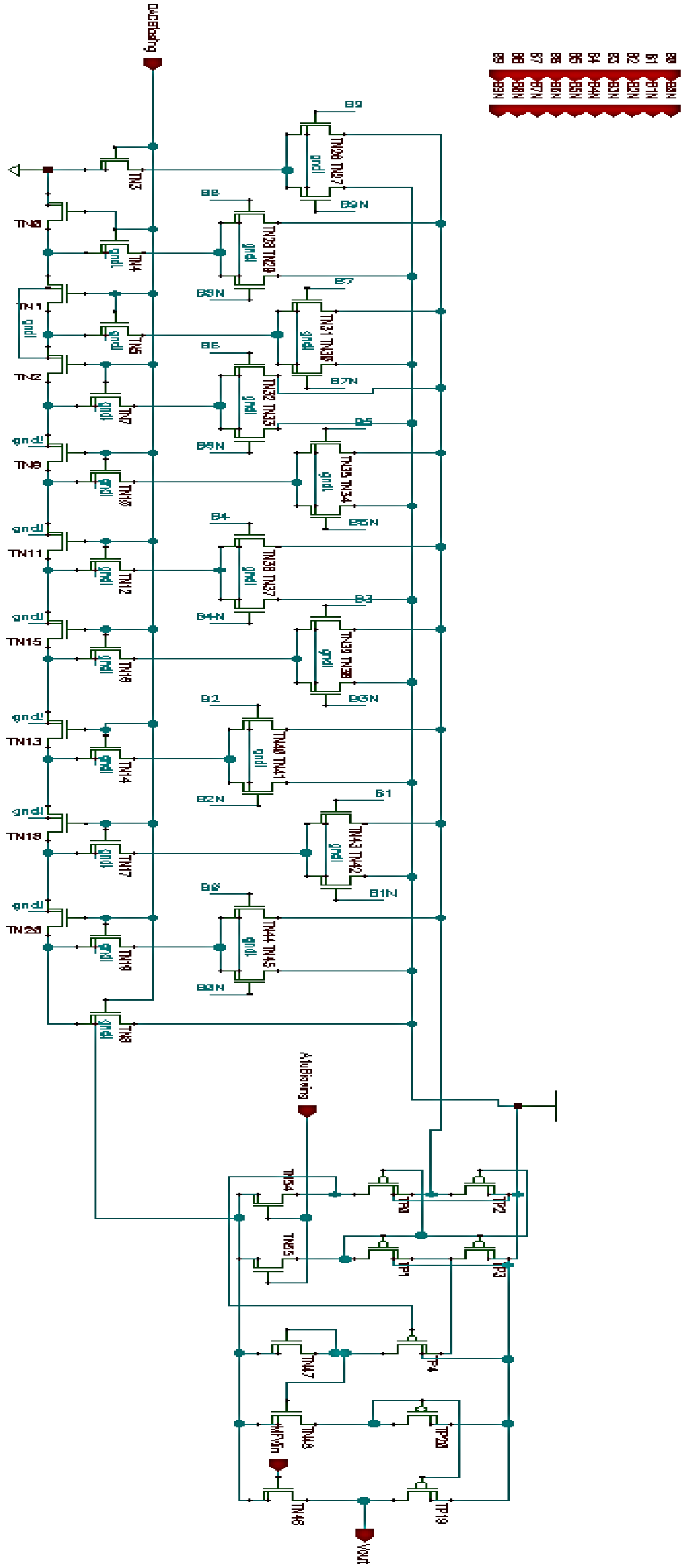


Figure A6: DAC Circuit Schematic

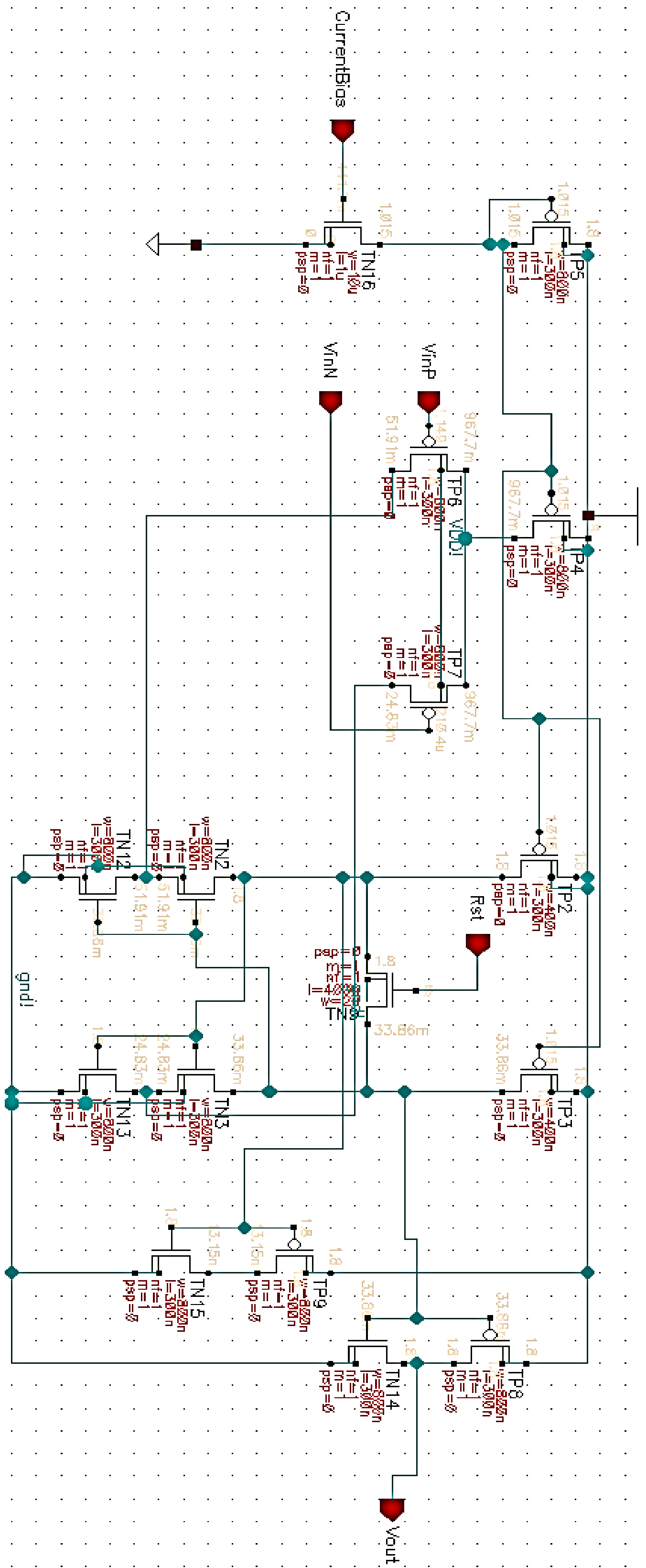


Figure A7: Dynamic Comparator Circuit Schematic

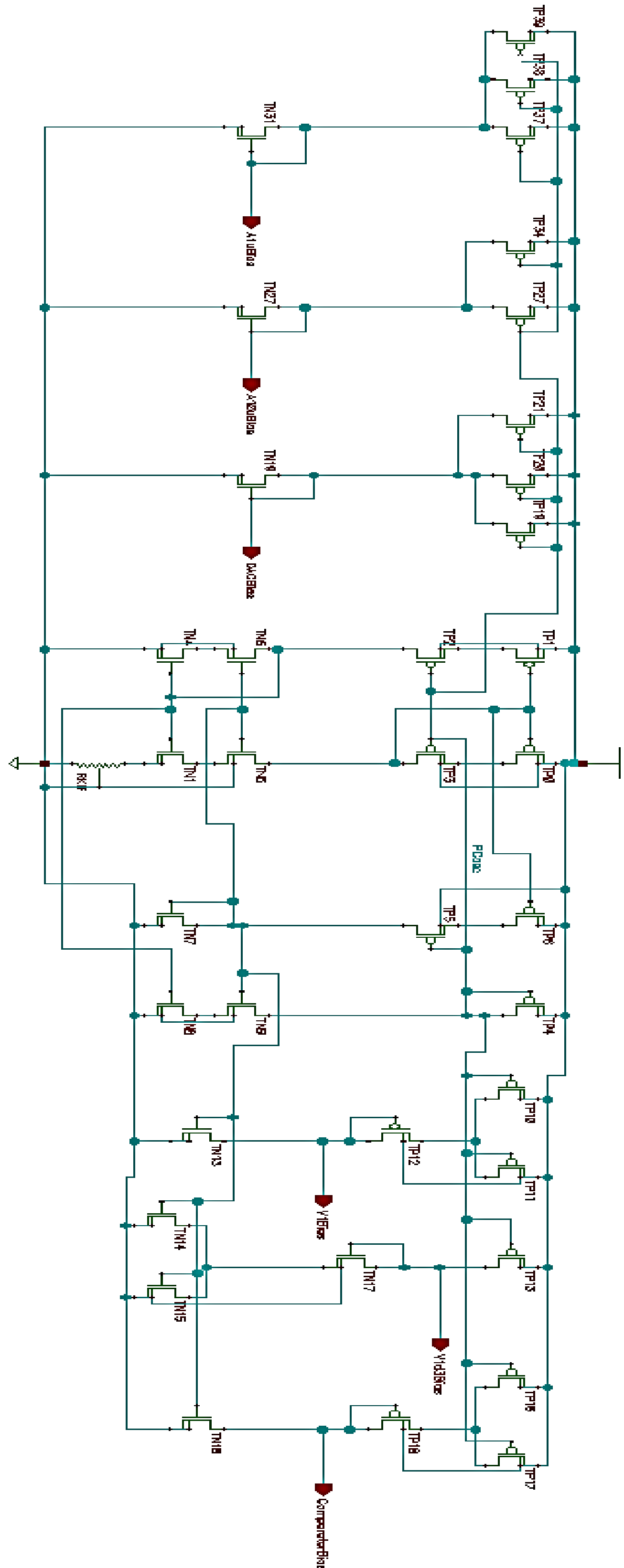


Figure A8: DC Biasing Circuit Schematic

Appendix B: Tables of Transistor W/L Ratios

***Note: TN stands for Transistor NMOS and TP stands for Transistor PMOS**

Table B1: Voltage Buffer 2 Transistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TP11	4/1	1
TN23	10/1	1
TP2	4/4	1
TP0	5/1	3
TP1	5/1	3
TN0	15/1	1
TN1	15/1	1
TP4	4/4	1
TN22	6/6	1
TN2	2/1	1
TN4	5/0.3	5
TN3	5/0.3	5

Table B2: Voltage Buffer 1 Transistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TP11	4/1	1
TN23	10/1	1
TP2	2/1	1
TP0	5/10	2
TP1	5/10	2
TN0	10/10	1
TN1	10/10	1
TP4	8/1	1
TN22	10/10	1
TN2	6/1	1
TN4	5/0.3	4
TN3	6/0.3	7

Table B3: Track and Hold Transistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TN0	1/0.3	1
TN1	1/0.3	1
TN3	11/11	1

Table B4: Voltage to Current Converter Transistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TP0	5/1	4
TP4	5/0.3	5
TP5	5/0.3	5
TP10	4/1	1
TP11	4/1	1
TN4	5/1	1
TN3	5/1	1
TN16	5/2	5
TN15	5/2	5
TN9	4/8	1
TP1	5/1	4
TN6	15/15	1
TN2	10/4	1

Table B5: DAC Transistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TN26-TN45	2/2	4
TN3	4/8	1
TN0	4/8	2
TN4	4/8	1
TN1	4/8	2
TN5	4/8	1
TN2	4/8	2
TN7	4/8	1
TN9	4/8	2
TN10	4/8	1
TN11	4/8	2
TN12	4/8	1
TN15	4/8	2
TN16	4/8	1
TN13	4/8	2
TN14	4/8	1
TN18	4/8	2
TN17	4/8	1
TN20	4/8	2
TN19	4/8	1
TN8	4/8	1
TP2	8/0.5	2
TP3	8/0.5	2
TP0	8/0.5	2
TP1	8/0.5	2
TN54	2/2	1
TN55	2/2	1
TP4	8/8	2
TN47	8/8	2
TP20	7/4	2
TN48	8/8	2
TP19	7/4	2
TN46	4/4	1

Table B6: Comparator Transistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TN16	10/1	1
TP5	0.8/0.3	1
TP4	0.8/0.3	1
TP6	0.8/0.3	1
TP7	0.8/0.3	1
TP2	0.4/0.3	1
TP3	0.4/0.3	1
TN9	2/0.4	1
TN2	0.8/0.3	1
TN3	0.8/0.3	1
TN12	0.8/0.3	1
TN13	0.8/0.3	1
TP9	0.8/0.3	1
TN15	0.8/0.3	1
TP8	0.8/0.3	1
TN14	0.8/0.3	1

Table B7: DC Biasing Transistor and Resistor W/L Ratios

Transistor	Width/Length (µm/µm)	Number of Fingers
TP39	1/5	1
TP38	1/5	1
TP37	1/4	1
TN31	2/2	1
TP34	8/2	1
TP27	8/2	1
TN27	10/1	1
TP21	18/3	1
TP20	17/3	1
TP19	17/3	1
TN19	4/8	1
TP1	4/2	1
TP0	4/2	1
TP2	2/2	1
TP3	2/2	1
TN6	2/2	1
TN5	2/2	1
TN4	2/2	1
TN1	4/2	1
RK0	5/3.44	N/A
TP6	4/2	1
TP5	4/4	1
TN7	2/2	1
TP4	2/2	1
TN9	4/4	1
TN8	4/2	1
TP10	3/1	1
TP11	3/2	1
TP12	8/5	1
TN13	6/2	1
TP13	3/2	1
TN17	8/0.3	1
TN14	10/8	1
TN15	10/8	1
TP16	5/2	1
TP17	3/1	1
TP18	9/2	1
TN18	6/2	1

Appendix C: Verilog Code

ADC Digital Controls Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: Cal Poly
// Engineer: Ryan Hunt
//
// Create Date:          10:33:08 04/04/2014
// Design Name:     SAR ADC
// Module Name:       Main
// Project Name: SAR ADC Design
// Target Devices: CADENCE
// Tool versions:
// Description: Verilog code for ADC Digital Controls
//
// Dependencies:
//
// Revision: 0
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module SARDigitalBlock(
        NSARBits,
        SARBits,
        CS,     //Chip Select
        SCLK,           //Serial Clock
        Comp,           // Comparator input
        THset,   //Track and Hold Set
        THrst,   // Track and Hold reset
        CompRst, // Comparator Reset
        Sout    // Serial Output
        );
        output [9:0] NSARBits;
        output [9:0] SARBits;
        input CS;       //Chip Select
        input SCLK;   //Serial Clock
        input Comp;   // Comparator input
        output THset;   //Track and Hold Set
        output THrst;   // Track and Hold reset
        output CompRst; // Comparator Reset
        output Sout;   // Serial Output
  // out holds the bit value of the SAR Data
  reg  CompReg = 0;
  reg rst = 0;
  wire clk;
```

```
   wire Nclk;
   reg [9:0] NSARBitsReg = 10'b0111111111;
   reg [9:0] SARBitsReg = 10'b1000000000;
   reg lastClk = 0; // Gives the last clock cycle
   reg [3:0] count = 4'b1001;

   assign NSARBits = NSARBitsReg;
   assign SARBits = SARBitsReg;
   assign THrst = rst;
   assign CompRst = CS? 1 : SCLK;
   assign THset = CS;
   assign Nclk = CS? 1 : !clk;
   assign clk = CS? 0 : SCLK;          // gives the clock to the device if CS is low
   assign Sout = CS? 0 : CompReg; // pushes it out to the Serial output
                                          // If CS is not low then it outputs high Z
   always @ (clk) begin
     if (count == 0) begin
       lastClk <= clk;
     end
     else begin
        lastClk <= 0;
     end
   end

   always @ (posedge clk) begin
      if (count > 0) begin
       count = count - 4'b0001; // reduces count at start
     end
     else begin
       count = 4'b1001;
     end
   end

   always @ (posedge Nclk or negedge CS) begin
     if (CS==0) begin
       CompReg = Comp;   // latches in comparator output so resets don't effect the output
line
     end
     else begin
       CompReg = 0;
     end
   end

   always @ (posedge clk or negedge CS) begin //SAR Algorithm
         if (CS == 0) begin
         case (count)
           4'b1001 :   begin
             SARBitsReg = 10'b1000000000;  //turns on the bit
             NSARBitsReg = 10'b0111111111; // turns off the not bit
```

```
            end
            4'b1000 :   begin
              if(CompReg == 1'b1) begin
                SARBitsReg = SARBitsReg | 10'b0100000000;
                NSARBitsReg = NSARBitsReg & 10'b1011111111;
             end
             else begin
                SARBitsReg = SARBitsReg ~^ 10'b0011111111; // turns off the previous bit
and
                                                                //turns on the current bit
                NSARBitsReg = NSARBitsReg ^ 10'b1100000000; // turns on previous not bit
and
                                                                      //turns off the current not
bit
              end
            end
            4'b0111 :   begin
               if(CompReg == 1'b1) begin
                   SARBitsReg = SARBitsReg | 10'b0010000000;
                   NSARBitsReg = NSARBitsReg & 10'b1101111111;
              end
              else begin
                   SARBitsReg = {SARBitsReg[9],SARBitsReg[8:0] ~^ 9'b001111111};
                   NSARBitsReg = {NSARBitsReg[9],NSARBitsReg[8:0] ^ 9'b110000000};
              end
            end
            4'b0110 :   begin
               if(CompReg == 1'b1) begin
                   SARBitsReg = SARBitsReg | 10'b0001000000;
                   NSARBitsReg = NSARBitsReg & 10'b1110111111;
               end
               else begin
                   SARBitsReg = {SARBitsReg[9:8],SARBitsReg[7:0] ~^ 8'b00111111};
                   NSARBitsReg = {NSARBitsReg[9:8],NSARBitsReg[7:0] ^ 8'b11000000};
               end
             end
             4'b0101 :   begin
                if(CompReg == 1'b1) begin
                   SARBitsReg = SARBitsReg | 10'b0000100000;
                   NSARBitsReg = NSARBitsReg & 10'b1111011111;
               end
              else begin
                   SARBitsReg = {SARBitsReg[9:7],SARBitsReg[6:0] ~^ 7'b0011111};
                   NSARBitsReg = {NSARBitsReg[9:7],NSARBitsReg[6:0] ^ 7'b1100000};
              end
           end
           4'b0100 :   begin
             if(CompReg == 1'b1) begin
                   SARBitsReg = SARBitsReg | 10'b0000010000;
```

```
          NSARBitsReg = NSARBitsReg & 10'b1111101111;
    end
   else begin
          SARBitsReg = {SARBitsReg[9:6],SARBitsReg[5:0] ~^ 6'b001111};
          NSARBitsReg = {NSARBitsReg[9:6],NSARBitsReg[5:0] ^ 6'b110000};
   end
end
4'b0011 :   begin
    if(CompReg == 1'b1) begin
          SARBitsReg = SARBitsReg | 10'b0000001000;
          NSARBitsReg = NSARBitsReg & 10'b1111110111;
    end
    else begin
          SARBitsReg = {SARBitsReg[9:5],SARBitsReg[4:0] ~^ 5'b00111};
          NSARBitsReg = {NSARBitsReg[9:5],NSARBitsReg[4:0] ^ 5'b11000};
    end
 end
 4'b0010 :   begin
     if(CompReg == 1'b1) begin
          SARBitsReg = SARBitsReg | 10'b0000000100;
          NSARBitsReg = NSARBitsReg & 10'b1111111011;
     end
     else begin
          SARBitsReg = {SARBitsReg[9:4],SARBitsReg[3:0] ~^ 4'b0011};
          NSARBitsReg = {NSARBitsReg[9:4],NSARBitsReg[3:0] ^ 4'b1100};
     end
  end
  4'b0001 :   begin
      if(CompReg == 1'b1) begin
          SARBitsReg = SARBitsReg | 10'b0000000010;
          NSARBitsReg = NSARBitsReg & 10'b1111111101;
      end
      else begin
          SARBitsReg = {SARBitsReg[9:3],SARBitsReg[2:0] ~^ 3'b001};
          NSARBitsReg = {NSARBitsReg[9:3],NSARBitsReg[2:0] ^ 3'b110};
     end
  end
  4'b0000 :   begin
      if(CompReg == 1'b1) begin
          SARBitsReg = SARBitsReg | 10'b0000000001;
          NSARBitsReg = NSARBitsReg & 10'b1111111110;
      end
     else begin
          SARBitsReg = {SARBitsReg[9:2],SARBitsReg[1:0] ~^ 2'b00};
          NSARBitsReg = {NSARBitsReg[9:2],NSARBitsReg[1:0] ^ 2'b11};
     end
 end
 default : begin
     SARBitsReg = SARBitsReg | 10'b0010000000;
```

```
              NSARBitsReg = NSARBitsReg & 10'b1101111111;
           end
          endcase
         end
         else begin
            SARBitsReg = 10'b1000000000;  //turns on the bit
            NSARBitsReg = 10'b0111111111; // turns off the not bit
         end
  end

  always @ (negedge lastClk or posedge CS)begin  //Track and hold reset
     if (CS == 1) begin
        rst = 0;
    end
    else begin
        rst =!rst;
    end
  end

endmodule
```

Test Bench Code:

```
`timescale 10ns / 1ns

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:   10:24:40 04/18/2014
// Design Name:   Main
// Module Name:   E:/Xilinx Verilog Files/ADC_SAR_BLOCK/ADC_Testbench.v
// Project Name:  ADC_SAR_BLOCK
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: Main
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////
```

```
module ADC_Testbench;
	//Internal
	reg CLK;
	reg CompLatch;
	integer i;

	// Inputs
	reg CS;
	reg SCLK;
	reg Comp;
	// Outputs
	wire [9:0] NSARBits;
	wire [9:0] SARBits;
	wire THset;
	wire THrst;
	wire CompRst;
	wire Sout;

	// Instantiate the Unit Under Test (UUT)
	Main uut (
		.NSARBits(NSARBits),
		.SARBits(SARBits),
		.CS(CS),
		.SCLK(SCLK),
		.Comp(Comp),
		.THset(THset),
		.THrst(THrst),
		.CompRst(CompRst),
		.Sout(Sout)
	);
	initial begin
		// Initialize Inputs
		CS = 1;
		i = 0;
		SCLK = 0;
		Comp = 0;
		CLK = 1;
		CompLatch = 0;
		#10;
	end

	always begin //Creates Fake clock
		CLK = 1;
		#5;
		CLK = 0;
		#5;
	end
```

```
	always @(negedge CLK) begin // Always keeps CS low after first clock
			CS = 0;		// Can reset for on a positive edge of CLK
		end

	always @ (CLK) begin
			if(CS == 0) begin // Passes CLK to SCLK after CS goes low
				SCLK = CLK;
			end
			else begin
				SCLK = 0;
			end
	end

	always @(posedge CompRst) begin
			Comp = 1; // Creates the reset state when CompRst goes high
			end

	always @(negedge CompRst) begin
			CompLatch = !CompLatch;
			Comp = CompLatch;	// Creates an alternating input
	end


	always @(posedge CLK) begin
			 i = i + 1;
				if (i == 12) begin // Counts when to set CS high
					CS = 1;
					i = 1;
				end
			end
endmodule
```

Appendix D: ABET Senior Project Analysis

Project Title: SAR ADC

Student: Ryan Hunt

Advisor: Dr. Vladimir Prodonov **Initials:** **Date:**

1. **Summary of Functional Requirments**

   The project centers on the design of a successive approximation register (SAR) ADC architecture that easily interfaces to a microcontroller (i.e. an Arduino). The ADC has a 10 bit resolution with reasonable DNL and INL (hopefully within 1 LSB). It has a range of 1V (largest code) to about 0V (smallest Code). The speed depends on the microprocessor's capabilities and the output to the microprocessor though this design uses the standard of a 2MHz clock creating a sample rate of 200k samples/second. This design allows for a cost effective alternative to the on board ADC that most development boards have as well as an effective choice for more custom designs with microcontrollers that don't have a built in ADC.

2. **Primary Constraints**

   There are several difficulties associated with the success of the project. I have to learn how to effectively use a very complicated software program for the design progress called Cadence. My ability to learn the software effectively will determine how fast the project reaches completion while also determining how well the project turns out. Another issue relates to the difficulty of the work itself. We get very little exposure to CMOS design so I have a lot more learning and research that I must accomplish to create an effective design.

   Another issue deals with testability of the finished product. Many of the subsystems of the ADC are inherently inaccessible. Without designing the chip with testing in mind may lead to difficulties with testing the chip so the chip's I/Os will have to have access to some of the intermediate steps in the ADC.

3. **Economic**

   If the design were to get funded by a large company for manufacturing many possible economic impacts could arise. For one, the device would take up space in the company's foundries. This means that the company has to put forward a large initial investment in terms of human resources to make the chip and financial capital to pay for resources to create the chip before they would make any profit. The creation of the chip also provides people working in the foundries with another product. If the product goes into high demand the foundries might have to invest in more real-estate on the foundry floor and thus hire more people to produce the product.

   When talking about costs and benefits throughout the lifespan of the project, if a company decided to bring it to full scale manufacturing it would take some time before the company could obtain a profit. Initially the only costs involve paying me for my time designing the chip but after that stage the chip needs to go out to manufacturing for testing which can cost a substantial amount (around $500,000 total).

At the end of senior project the chip will need to go to manufacturing which will take around a month. After if comes back one would need to perform testing on the chip to ensure that it works. This could take another few months. If it doesn't work it would have to go back to the design stage to get fixed and then retested (another 5 months or so). If it worked as intended the first round then the chip could go into a full scale manufacturing after the testing finished. The Gantt chart for my development process is seen below.

Key	Color
Work In Progress	
Due	
Advisor Feedback	
Break, No work	
Major Divisions	

Figure D1: Gantt Chart Color Key

Task	Start	End	Duration (Days)	Fall 2013											
				Week 1 23-Sep	Week 2 30-Sep	Week 3 7-Oct	Week 4 14-Oct	Week 5 21-Oct	Week 6 28-Oct	Week 7 4-Nov	Week 8 11-Nov	Week 9 18-Nov	Week 10 25-Nov	Week 11 2-Dec	Week 12 9-Dec
Project Plan (EE460 Requirements)															
Abstract V1	Summer	23-Sep	varies												
Requirements and Specs. V1	25-Sep	7-Oct	12												
Block Diagram	9-Oct	14-Oct	5												
Literature Search	25-Sep	21-Oct	26												
Gantt Chart	23-Oct	28-Oct	5												
Cost Estimates	23-Oct	28-Oct	5												
ABET Sr. Project Analyisi	25-Oct	4-Nov	9												
Requirements and Specs. V2	6-Nov	11-Nov	5												
Report V1	23-Sep	15-Nov	52												
Advisor Feed Back	N/A	N/A	N/A												
Report V2	22-Nov	9-Dec	17												
Folded-Cascode Op-Amp Design															
LTSpice Layout	24-Oct	31-Oct	7												
Simulation Test Design	24-Oct	31-Oct	7												
Move To Cadence	31-Oct	3-Nov	3												
Layout Design	1-Nov	8-Nov	7												
Simulation Test Layout Design	2-Nov	9-Nov	7												
Design V2 Revisions	8-Nov	12-Nov	4												
Comparator Design															
LTSpice Layout	11-Nov	22-Nov	11												
Simulation Test Design	11-Nov	22-Nov	11												
Move To Cadence	22-Nov	24-Nov	2												
Layout Design	25-Nov	9-Dec	14												
Simulation Test Layout Design	25-Nov	9-Dec	14												
Design V2 Revisions	9-Dec	12-Dec	3												

Figure D2: Fall 2013 Gantt Chart

Task	Start	End	Duration (Days)	Winter 2014											Break
				Week 1 6-Jan	Week 2 13-Jan	Week 3 20-Jan	Week 4 27-Jan	Week 5 3-Feb	Week 6 10-Feb	Week 7 17-Feb	Week 8 24-Feb	Week 9 3-Mar	Week 10 10-Mar	Week 11 17-Mar	Week 12 24-Mar
Report (EE463 Requirements)															
Simulation Testing Documentation	6-Jan	12-Mar	66												
Report Revisions (EE463 Report)	N/A	N/A	N/A						10-Feb					12-Mar	
Faculty Design Review	10-Feb	20-Feb	10												
Overall Design															
Level 1 Diagram	6-Jan	16-Jan	10												
Level 2 Diagram	16-Jan	27-Jan	11												
System Flow Overview	6-Jan	10-Feb	34												
Digital Flow Overview	27-Jan	17-Feb	20												
Sample And Hold Design															
LTSpice Layout	13-Jan	27-Jan	14												
Simulation Test Design	14-Jan	28-Jan	14												
Move To Cadence	28-Jan	30-Jan	2												
Layout Design	30-Jan	14-Feb	14												
Simulation Test Layout Design	31-Jan	15-Feb	15												
Design V2 Revisions	15-Feb	22-Feb	7												
Analog Side of DAC Design															
LTSpice Layout	10-Feb	24-Feb	14												
Simulation Test Design	11-Feb	25-Feb	14												
Move To Cadence	25-Feb	27-Feb	2												
Layout Design	27-Feb	12-Mar	15												
Simulation Test Layout Design	27-Feb	12-Mar	15												
Design V2 Revisions	12-Mar	17-Mar	5												

Figure D3: Winter Gantt Chart

Task	Start	End	Duration (Days)	Spring 2014											Break
				Week 1 31-Mar	Week 2 7-Apr	Week 3 14-Apr	Week 4 21-Apr	Week 5 28-Apr	Week 6 5-May	Week 7 12-May	Week 8 19-May	Week 9 26-May	Week 10 2-Jun	Week 11 9-Jun	Week 12
Report (EE464 Requirements)															
Simulation Testing Documentation	31-Mar	19-May	49												
EE464 Report (Revision Dates)	N/A	N/A	N/A												
Senior Project Expo	19-May	29-May	10												
Upload to Digital Commons	N/A	N/A	N/A										2-Jun		
Additional Designs															
LTSpice Layout	31-Mar	7-Apr	7												
Simulation Test Design	1-Apr	7-Apr	6												
Move To Cadence	7-Apr	9-Apr	2												
Layout Design	9-Apr	19-Apr	10												
Simulation Test Layout Design	9-Apr	19-Apr	10												
Design V2 Revisions	19-Apr	23-Apr	4												
Analog Sytstem Integration															
Combine Cadence Cells	23-Apr	4-May	11												
Combine Layout Cells	23-Apr	4-May	11												
System Testing	23-Apr	4-May	11												
Design V2 Revisions	4-May	9-May	5												
Verilog Digital Design															
Initial Design	4-May	12-May	8												
Integration With Analog	11-May	15-May	4												
Simulation Testing	15-May	17-May	2												
Design V2 Revisions	17-May	29-May	12												

Figure D4: Spring Gantt Chart

4. If Manufactured on a Commercial Basis

After the chip goes into full mass production it costs far less per chip (around $5 to $8). at this point profit starts to accrue. Priced at about $25 each chip turns a profit of about $20. Say that with initial costs of my payment (around $4.5k) plus the testing costs ($500k) the total cost of initial investment comes out to $504.5k. If around 5000 (this is a fairly conservative number for worldwide sales) of these chips are sold each year this would yield a profit of $100k. This means it would take about 5 years to break even. The user shouldn't have to pay for anything other than the cost of the chip (unless they damage it) and for the wires required to attach it to the microcontroller.

Small production could be run through Mosis [4] to get the chip in working order. This would cut down on the price of creating the test chips seeing as for this project it comes free to the school for research purposes.

5. Environmental

Many environmental issues coincide with the manufacturing of ICs. Directly the process uses a substantial amount of silicon. It also uses rare metals such as gallium and arsenide. The process also uses many various chemicals such as acetone, arsenic, arsine, benzene, cadmium, hydrochloric acid, lead, methyl chloroform, toluene, and trichloroethylene [16]. These chemicals take up resources and require disposal, which in some countries where the manufacturing is done means just throwing it out in a way that may damage the environment. Manufacturing also indirectly uses electricity as a resource which is created through various means including nuclear, coal, and thermal energy. Each of these sources also has their own environmental impact. The main way that this project would affect other species ties directly to pollution coming from the plants where the chips get manufactured.

6. Manufacturability

This product must be funded by a large company to make it to the manufacturing stage. Without a large company, finding the financial capital to support large scale production would prove difficult. Another reason for getting a large company behind the manufacturing process pertains to the ability to find high quality foundries. If the foundry proves not to create a high quality product yield from each wafer could be severely diminished and thus costing the company more to produce the product. Various other aspects can affect the production of the product such as natural disasters putting the foundry out of commission. This loss of production time could lead to a loss in profits.

7. Sustainability

Once completed the chip has minimal maintenance requirements. Ensuring that the device does not get destroyed by giving it too much voltage on the rails or the input ensures that the device doesn't get damaged. Also making sure that the leads remain corrosion free ensures that the device's connectivity remains high. Keeping the chip in a place where the leads won't bend ensures that they do not break off or get damaged in any way that would prevent the chip from working in the intended fashion.

The device does require some rare metals and various chemicals to manufacture. In some cases these chemicals won't last forever but under the current usage no problems should arise in maintaining manufacturing for some time.

Over time the device will age which means that the wires will deteriorate. The device should maintain operational use for at least 20 years of consistent use. The chip should also be designed in a way that the components can get recycled easily so as to lessen its overall environmental footprint and allow for continued manufacturing.

Once the design ships out upgrades don't prove as an effective way to improve the design seeing as any improvements would involve design changes on the IC level. If upgrades became necessary then they would have to go through the whole process of design and testing all over again to ensure that the changes to the design worked. At that point just releasing a new chip would probably prove more effective.

8. Ethical

IEEE Code of Ethics [3]:

When it comes to the IEE code of Ethics several ethical problem could arise from the manufacturing and completion of the project. A major ethical problem derives from the statement "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others." If in the testing stage we found that for all but one of the desired aspects of the chip worked perfectly it would be considered unethical to just write off this small problem and ok the chip to go to manufacturing. This slight error could then manifest into a serious issue that prevents the chip from working perfectly and thus could make the product a failure. Not only would this affect the company's profit but it would also adversely affect the consumer because they now have a faulty product which can in turn affect the product they are developing. This in turn also comes up in the statement "to avoid real or perceived conflicts of interest whenever possible and to disclose them to affected parties when they do exist." under the same situation the consumer would most likely not know that the device has faulty data on it. This becomes a conflict of interest for the company developing the chip because they might not want the release of the information on the faulty chip to become public. Though to avoid this conflict of interest the companies that have purchased the chip would need to get the information pertaining to the issues that slipped past testing.

Utilitarianism:

The utilitarianism framework of the greatest good for the greatest number affects how I go about my design itself. The idea of the project revolves around creating a universal ADC that works in many different projects and has a relatively low cost. When obtaining a lower cost I may need to send manufacturing overseas where prices reflect the general economic standpoint of the country. This means taking away jobs from people in the U.S. but in return the consumer of the product will have a cheaper product.

9. Health and Safety

The completed device has very little health or safety issues associated with it but the manufacturing process does have quite a few. The chip uses low power technology so the chances of an electrical shock remain low but if the chip were to get thermal damage it could lead to a small fire or the release of fumes that could prove hazardous to one's health. On the manufacturing side many toxic chemicals (such as lead) get used to create the chip which can put the employees of the plant at risk.

10. Social And Political

Revisiting what I addressed when talking about ethics from a utilitarianism stand point where the chip gets manufactured has large social and political impacts. By not choosing to manufacture in the U.S. some of the cost gets reduced but it also takes away jobs from U.S. manufactures. Also depending on where the chip gets manufactured the quality of life the facility's staff may not live up to an ethical standard that we hold here in the U.S. This can impact the community that the chip gets manufactured in as well because it can bring in outside income to help sustain the local and federal government associated with that community.

The stakeholders in the project range from the people directly creating the chip to the people delivering it. The chip brings business to all of the people involved directly in getting the chip to the consumer as well as the people who use the device that the consumer themselves create. If the chip doesn't work or doesn't sell well then all of the stakeholders get adversely affected.

11. Development

During the development process I had to learn how to use Cadence. This program handles all the simulation and layout of the chip. Cadence gets used across the board in industry for IC design so dealing with this program proved to benefit me greatly. I also had to learn the technology behind manufacturing the chips to find my limitations in the designs that I can make.