8-2014

Digital-to-Analog Converter Interface for Computer Assisted Biologically Inspired Systems

Nicholas Conley Poore
University of Tennessee - Knoxville, npoore2@vols.utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Nicholas Conley Poore entitled "Digital-to-Analog Converter Interface for Computer Assisted Biologically Inspired Systems." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Jeremiah Holleman, Major Professor

We have read this thesis and recommend its acceptance:

Benjamin Blalock, John D. Birdwell

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

8-2014

Digital-to-Analog Converter Interface for Computer Assisted Biologically Inspired Systems

Nicholas Conley Poore
University of Tennessee - Knoxville, npoore2@vols.utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Nicholas Conley Poore entitled "Digital-to-Analog Converter Interface for Computer Assisted Biologically Inspired Systems." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Jeremiah Holleman, Major Professor

We have read this thesis and recommend its acceptance:

Benjamin Blalock, John D. Birdwell

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Digital-to-Analog Converter Interface for Computer Assisted Biologically Inspired Systems

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Nicholas Conley Poore

August 2014

In dedication to my father and mother for making me who I am.

Acknowledgements

First and foremost, I would like to thank my adviser, Professor Jeremiah Holleman, for without his signature this document would have never seen the light. On a more serious note, Dr. Holleman has been a great mentor for the past year. His technical advice was of great value to me, as well as his kind encouragement which helped through many difficult times. Professor John D. Birdwell, my co-adviser, has offered his deep expertise in the area of control systems, and has also given me great support during my time at the Univeristy of Tennessee. For that, I owe him great thanks. I am deeply thankful to Professor Ben Blalock for his great lectures on analog circuit design, for his amazingly broad knowledge that he was always willing to share, and for his enthusiasm and optimism. I again would like to thank Professor Holleman and the ISiS group, as their analog signal processing project gave me inspiration to work on low-power DACs and system interfaces.

I would like to thank Mosis for fabricating the chip.

Many thanks to DARPA for funding my research assistantship that allowed me to come to graduate school in the first place.

I would also like to thank our wonderful administrative assistants, Dana Bryson, Julia Elkins, and Melanie Kelley for all their help.

I want to take this opportunity to let my friends and co-workers know how much I appreciate their company and technical advice. Thank you Shahriar"Pollob" Jahan, Kelly Griffin, Tan Yang, Junjie Lu, Terrance Randall, Mohsen Joudi, Peixing Liu, Logan Taylor, and Jake Shelton.

Daniel Tubb, Morgan Helton, Matt Zarb, Ilona Molotoka, Doscher Shewmake, Ali Ghezawi, Chris Zotti, Andy Dagen, Emert Ridener, and many other friends – you are important to me.

To my family, you are the most important thing in my life. Without you my work would be meaningless.

Heather Bradley, I cannot thank you enough for your patience and everything you have done for me.

The words of gratitude in my thesis may not say much and may not be as eloquent as I would like them to be, but I really hope that even without these words, all of you feel very much appreciated. You truly are.

"Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning." - Albert Einstein

Abstract

In today's integrated circuit technology, system interfaces play an important role of enabling fast, reliable data communications. A key feature of this work is the exploration and development of ultra-low power data converters. Data converters are present in some form in almost all mixed-signal systems; in particular, digital-to-analog converters present the opportunity for digitally controlled analog signal sources. Such signal sources are used in a variety of applications such as neuromorphic systems and analog signal processing. Multi-dimensional systems, such as biologically inspired neuromorphic systems, require vectors of analog signals. To use a microprocessor to control these analog systems, we must ultimately convert the digital control signal to an analog control signal and deliver it to the system. Integrating such capabilities of a converter on chip can yield significant power and chip area constraints. Special attention is paid to the power efficiency of the data converter, the data converter design discussed in this thesis yields the lowest power consumption to date. The need for a converter with these properties leads us to the concept of a scalable array of power-efficient digital-to-analog converters; the channels of which are time-domain multiplexed so that chip-area is minimized while preserving performance. To take further advantage of microprocessor capabilities, an analog-to-digital design is proposed to return the analog system's outputs to the microprocessor in a digital form.

A current-steering digital-to-analog converter was chosen as a candidate for the conversion process because of its natural speed and voltage-to-current translation

properties. This choice is nevertheless unusual, because current-steering digital-to-analog converters have a reputation for high performance with high power consumption. A time domain multiplexing scheme is presented such that a digital data set of any size is synchronously multiplexed through a finite array of converters, minimizing the total area and power consumption. I demonstrate the suitability of current-steering digital-to-analog converters for ultra low-power operation with a proof-of-concept design in a widely available 130 nm CMOS technology. In statistical simulation, the proposed digital-to-analog converter was capable of 8-bit, 100 kSps operation while consuming 231 nW of power from a 1 V supply.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Integrated circuit transistor density has been steadily increasing, and with modern VLSI design flow tools, the realization and synthesis of digital circuits is occurring at impressively accelerated rates. But the ubiquitous Complimentary Metal Oxide Semiconductor (CMOS) technology cannot be expected to miniaturize forever. Eventually, CMOS will reach the point where the behavior of the transistor is outside the realm of its expected behavior. In the meantime, digital circuits still benefit from device scaling.

Digital circuits, because of their robustness and ability to be synthesized, are used whenever possible in lieu of analog blocks. The digital hardware can easily be controlled by microprocessors and allow flexibility through programming. So why even bother with analog? Digital processing operations are more complex to implement than analog operations and have a limiting discrete value range with a high cost of precision. It is possibly best said by Bob Widlar, "Digital? Every idiot can count to one... ." In many digital systems, the power and area costs are proportional to the number of bits used in the computation. In such cases, a 12-bit computation, such as an addition, consumes one-half as much area and one-half as much power as does a 24-bit addition if all parameters such as clock frequency, average switching capacitance, and power supply voltage remain fixed. If we do allow

the clock frequency and power supply voltage to scale with the number of bits, as in a bit-serial implementation, then the power costs scales as a polynomial function of the number of bits. Some computations, such as multiplication, have power and area costs that scale with the square of the number of bits. Analog computation can be far more efficient than digital computation due to analog computation's repertoire of rich primitives. For example, the addition of two parallel 8-bit numbers requires a single wire in analog circuits (using Kirchoff's current law), whereas it would require about 240 transistors in static CMOS digital circuits (for a cascade of 8 full adders). (Sarpeshkar, 1998)

Analog computation offers superior scalability in terms of power consumption. Because the number of devices required to perform a computation is significantly greater in digital systems, more wiring and communication overhead is required. The presence of more devices and more communication overhead causes digital circuits to have typically higher area consumption than that of analog circuits. The energy dissapation from switching due to the large number of devices and the communication overhead also causes the power consumption to be higher in digital circuits (Sarpeshkar, 1998). Analog circuits draw power constantly, however, this power is still significantly less than the total average switching power in digital systems. Precision in digital systems is dependant upon the number of bits; each bit requires a wire and additional components. Analog systems have multiple bits of precision represented on a single wire. This leads to the conclusion that analog systems have greater scalability properties for large multi-dimensional designs like biologically inspired neural networks.

These complex data-processing systems can require large numbers of simultaneous inputs. This introduces an issue associated with analog computation— the difficulty of interfacing and managing analog data. Digital data can be easily bussed by a microprocessor and queued in system memory before processing. Unlike digital data, analog signals cannot be bussed by a microprocessor nor does it have the luxury of

nonvolatile memories. Therefore, it is paramount to develop a scalable analog data management solution.

For this thesis, I present a scalable interface architecture that allows analog computational systems to be controlled by a microprocessor. The interface is based on an organized array of Digital-to-Analog Converters (DACs), multiplexers, and analog sample-and-holds. Multi-dimensional data are translated from the digital domain to the analog domain and fed simultaneously into the analog system.

This thesis details the design of the DACs used in the proposed architecture. There are two ultra low-power DAC designs. First, I designed a binary current-steering DAC. While simple in design and with minimum area, the DAC performance was not sufficient. The DAC was redesigned and improved with supplemental thermometer encoding circuitry. This design maintains the same power-consumption and operates faster than the original.

The organization of this thesis is as follows: Chapter 2 spends a significant amount of time discussing basic integrated circuit background and digital-to-analog conversion along with their relevance with the top-level design of the interface. My goal is that someone with no prior knowledge will be able to understand the primary blocks used in the system interface. Chapter 3 primarily proposes a step-by-step design of the proposed microprocessor to analog system interface design. This Thesis descirbes two ultra-low power DAC designs in Chapter 4: one with many design flaws and another that operates successfully. Chapter 4 also compares their simulation results. Discussion of the results and thesis conclusions are presented in Chapter 5.

Chapter 2

Background

This chapter discusses the background information relevant to the overall application and thesis research. In Section 2.1, the topic of digital and analog integrated circuits are reviewed along with CMOS transistors. Section 2.2 will include a discussion of digital and analog signals. For discussion of system I/O, Section 2.3 discusses SISO and MIMO system interfaces. Section 2.4 covers performance metrics that are useful in evaluating and comparing designs of a DAC. Lastly, in Section 2.5, we discuss the basics of digital-to-analog conversion (DAC) techniques and the designs relevant to this research.

2.1 Integrated Circuits

The invention and construction of the integrated circuit is arguably one of the most important inventions of the previous century. The Intel 4004 microprocessor is considered to be the first large-scale integrated circuit (LSI). It was delivered in 1971 and contained about 2300 transistors. Its impact and implications on modern communication and lifestyle has been tremendous. Nowadays (2014), the largest chips contain several billions of transistors (15-Core Xeon Ivy Bridge-EX). If Moore's Law were to continue to hold, the density of transistors on a chip should approximately double every 18th month. In this information technology era,

products such as: wireless terminals (mobile phones), laptop computers, Bluetooth modules, and personal digital assistants (PDAs); require fast, dense, and low power consuming integrated circuits. For high-integration, low-power applications the bipolar technique has been replaced by the CMOS technique. Standard CMOS processes offer significantly smaller transistor sizes, and the capacitive loads in CMOS digital circuits effectively eliminates static power dissipation. However, in instances where applications require high-speed and high-performance, the bipolar technique is still widely used (Rabaey et al., 2002). We will in our case consider the CMOS technology throughout the thesis.

By implementing both analog and digital circuits on the same chip, the off-chip design complexity and the layout of printed circuit board (PCB) are simplified; in addition, the induced disturbance on sensitive interconnection wires is reduced. As digital circuits continue to rapidly develop, the associated power consumption is steadily being reduced due to the decreasing necessary supply voltage. The design of high-efficiency, analog circuits becomes increasingly complicated as the voltage range shrinks. With less voltage headroom, transistors in analog circuits cannot be expected to operate conveniently in the saturation or the triode region in some sub 1.2 V systems. As a result, future design of analog circuits will need to focus on nearthreshold and subthreshold operation.

A mixed-signal circuit is approximately considered to be a subcircuit in which both analog and digital circuits are used. Typically, the interface between the digital and analog domain, such as DACs, ADCs, and phase-locked loops (PLL)— are considered to be mixed-signal circuits.

2.1.1 Digital Circuits

The design of digital circuits can be divided into a number of different classes. The importance of each class is dependent upon the intended application. With decreasing transistor dimensions, the influence of wire lengths, and parasitic capacitance becomes

very important and requires knowledge in pure analog design as well. The accuracy of the circuit can be improved by simply increasing the word length (number of bits used to represent the signals) to a desired level. With carefully evaluated algorithms and long word lengths, the digital noise can be kept at a very low level (Rabaey et al., 2002).

For a digital CMOS circuit, the power dissipation is provided in Equation 2.1, where: α is the circuits switching activity, f is the clock frequency, C_L is the average capacitive load for each gate, n is the number of gates, V_{DD} is the supply voltage, and ΔV is the voltage swing (Rabaey et al., 2002).

$$P \approx \alpha f \cdot C_L \cdot V_{DD} \cdot n \cdot \Delta V \tag{2.1}$$

2.1.2 Analog Circuits

Although, there are now automated tools for use in the layout and design of analog circuits, much of this work is still performed by hand. An experienced designer is needed to implement high-performance analog circuits. Due to short-channel effects, analog circuits do not scale nearly as well as digital circuits. As a result, when the process is changed or updated, the analog circuit requires a significant redesign. However, the smaller process dimensions produces less parasitic capacitance; therefore, the achievable bandwidth can be increased (Wikner, 2000).

For analog designers, one of the major problems with modern CMOS technologies is the decreasing supply voltage. A low supply voltage slows down the circuit, which increases the difficulty of making a current source with high impedance (Razavi, 2002). Another important design challenge is the matching of transistors. Special attention must be paid to the physical layout of circuits; otherwise, process variation will ruin the reliability of analog circuits. Careful layout techniques, such as common-centroid, reduce the effects of process variation and increase reliability.

When analyzing and designing analog circuits, we consider the linearization of the circuit around the operating point. Unlike digital circuits, analog circuits, such as amplifiers, are typically biased to a certain voltage level with a DC bias current. Therefore, the bias current multiplied by the supply voltage produces the power dissipation.

$$P = I_{bias} V_{DD} \tag{2.2}$$

2.1.3 System-on-Chip and Mixed-Signal Circuits

Mixed-signal circuits consist of any combination of analog, digital and radio-frequency circuits. A system-on-chip (SoC) is an integrated circuit (IC) that integrates all components of a computer or other electronic system into a single chip. Various digital signal processing (DSP) cores, memory, analog interfaces, oscillators, plls, and other "blocks" are organized together to create a SoC. SoCs are commonly mixed-signal systems located on a single chip substrate.

2.1.4 Transistor Operation

Analog and digital circuits are commonly built with CMOS transistors. The CMOS transistor can operate in a number of different regions, including the cut-off, subthreshold, linear (triode), and saturation regions. In analog circuits, excellent performance is observed when transistors are biased to operate in their saturation regions. This results in high output impedance; and therefore, a high gain (i.e. the output current is nearly independent of the voltage applied across the drain and source terminals). With lower voltage headroom, transistors may be forced to operate in the triode region. The saturation drain-current relation is shown in Equation 2.4 and the triode drain-current relation is shown in Equation 2.3.

$$I_D \approx \mu_n C_{ox} \frac{W}{L} \left((V_{GS} - V_{th}) V_{DS} - \frac{V_{DS}^2}{2} \right), \quad V_{DS} \leq V_{GS} - V_{TN} \tag{2.3}$$

$$I_D \approx \frac{\mu_n C_{ox}}{2}\frac{W}{L}(V_{GS} - V_{TN})^2(1 + \lambda V_{DS} - V_{DSsat}), \quad V_{DS} \geq V_{GS} - V_{TN} \tag{2.4}$$

For low power operations, subthreshold operation of the transistors is considered. Subthreshold operation is the transistor operating region just before the transistor is turned off. Here the CMOS transistor behaves more like a bipolar transistor, the drain current is described by Equation 2.5, where $n \approx 1.5$ is a process-dependent constant, q is the electron charge, k is the Boltzmann constant, T is the absolute temperature, and I_{D0} is a constant current (Johns and Martin, 2008). The transistor is very slow and sensitive to matching errors in the subthreshold region and is not suitable for high-speed operation. The transistor needs to be large to achieve a high SNR (Wikner, 2000).

$$I_D \approx I_{d0}\frac{W}{L}e^{\frac{q}{kT}\frac{V_{GS}}{n}}, \quad V_{GS} < V_{TN} \tag{2.5}$$

2.2 Analog and Digital Signals

Digital signals are physical signals that represent a set of discrete values (a quantized discrete-time signal). Digital signals have a finite resolution. Analog signals are continuous signals for which the time-varying features (variable) of the signals are a representation of some other varying quantity. In contrast, analog signals have a theoretically infinite resolution. In practice analog signals are subject to electronic noise and distortion introduced by various transmission mediums and signal processing operations, which can progressively degrade the signal-to-noise ratio. Digital signals can be processed or transmitted without introducing additional noise or distortion. In an analog form, the signal is vulnerable to noise and distortion.

Typically, the digital signals have a high switching activity that produces large current and voltage spikes through the supply wires and substrate. In a low-ohmic,

positively doped substrate without the twin-well option, the bulks (the potential of the substrate) of analog and digital NMOS transistors are almost shorted (Razavi, 2002). Through the capacitive coupling the current spikes influence the sensitive analog signals yielding a poor signal-to-noise ratio. To minimize the noise, we should properly guard the sensitive analog circuits and wires in mixed-signal systems through substrate contacts, guard-rings, and isolated ground return paths (Ismail and Fiez, 1994).

Analog signals are very sensitive and therefore require the utmost care when routed. Concerns for digital signal routing is also important, especially when considering particular digital paths, such as high data-rate buses and digital clocks. These buses should be isolated from more sensitive digital circuitry and ever more so from sensitive analog signal paths. Therefore, mixed-signal system interfaces need to effectively isolate digital circuits from analog circuits as much as possible and apply shielding when separation is impossible.

Converting an analog signal to digital form or a digital signal to analog form introduces a constant low-level noise called quantization noise into the signal. The quantization noise determines the minimum noise floor. The hardware that performs these conversions is labeled analog-to-digital converters (ADC) and digital-to-analog converters (DAC). The purpose of these converters is to transform the digital representation of a signal into its corresponding analog representation (or vice versa) without contributing significant noise or distortion.

2.3 System Inputs and Outputs

Regardless of digital and analog circuit combinations, all systems have at least one thing in common— inputs and outputs. In the land of integrated chip design, system inputs and outputs are mapped to physical pins; physical pins provide a medium that allows electromagnetic signals to propagate to and from the system. As the number of inputs and outputs increases, so do the number of physical pins. This

results in increased surface area for additional signal routing and pins. In mixed-signal environments, analog signal routing must be separated from digital routing, which results in a more tedious and time consuming process than that of a solely digital or analog environment.

Generally speaking, an electronic system with n-outputs and p-inputs has n physical pins, and p physical pins, in order to provide the necessary inputs. This does not account for power pins or ground pins. A single-input, single-output (SISO) system is therefore managed with 2 pins for the signals in addition to power pins. A multi-input, multi-output (MIMO) system could easily have 256 or more inputs and outputs— that could quickly result in an I/O routing headache. With chip area at a premium, the total number of physical pins on a given chip size is finite. Without a proper I/O strategy, the system will not be easily scalable. To make the system scalable, a finite set of input and output pins are defined with peripheral interface circuitry to process the inputs and outputs.

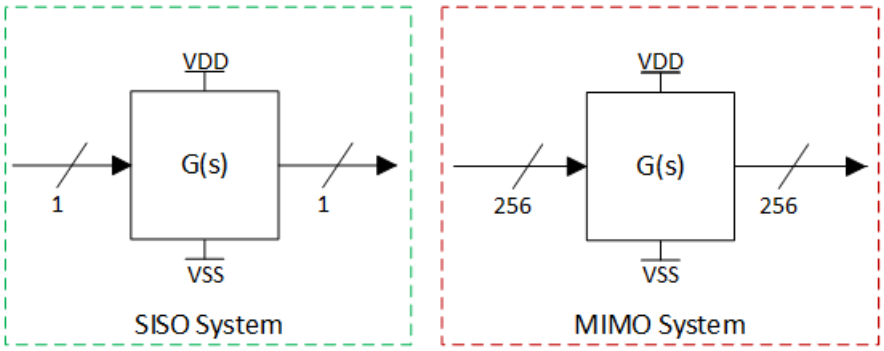

Figure 2.1: SISO and MIMO Systems

2.4 DAC Performance Metrics

This thesis is concerned with the design and development of an analog system interface— more specifically DACs. But before DAC architectures and their design can be discussed, measurements of performance and figures of merit must be properly

defined. The following are some of the many definitions necessary to provide a basis of comparison and discrimination for DACs.

2.4.1 Performance

Resolution

Resolution specifies number of possible output levels the DAC is to reproduce. This property is usually stated as the number of bits it uses. Resolution determines color depth in video applications and audio bit depth.

Max Sampling Rate

Maximum sampling rate is a measurement of the maximum speed at which the DACs circuitry can operate and still produce the correct output. As stated in the Nyquist Sampling Theorem, there exists a defined relationship between the sampling frequency and maximum bandwidth signal that can be reconstructed from knowledge of the sampled signal. The relationship is shown in Equation 2.6, where B is the signal bandwidth, and f_s is the sampling rate.

$$B < \frac{f_s}{2} \tag{2.6}$$

Monotonicity

Monotonicity is the ability of a DAC's analog output to have a positive correlation with the direction of the digital input. If the digital value increases, the output should not decrease. A non-monotonic DAC's output has missing output levels when the digital value increases but the DAC output fails to increase.

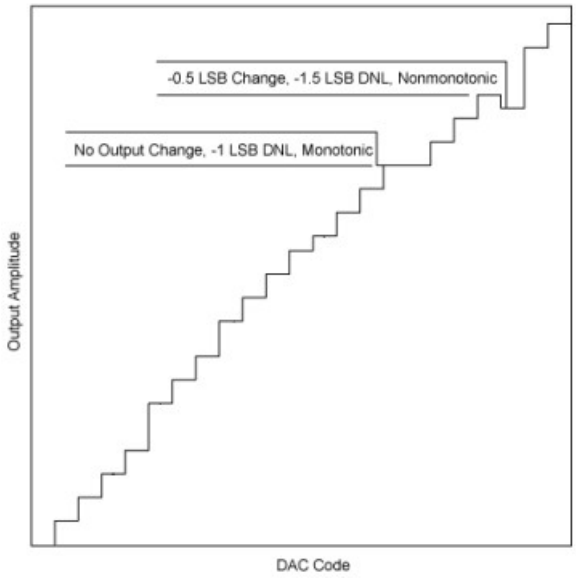

Figure 2.2: Example of Monotonicity from atx7006.com (2014)

2.4.2 Static Figures of Merit

Offset Error and Gain Error

The most commonly specified end-point errors associated with DACs are offset error and gain error. Offset Error quantifies the amount by which the actual characteristic is linearly shifted from its ideal position. It can be measured by applying the all "0"s code to the DAC and measuring the output deviation.

The ideal transfer function has a slope defined by drawing a straight line through the two end points. The slope represents the gain of the transfer function. The gain error quantifies the deviation of the slope of the actual staircase from its intended slope (Manganaro, 2012). Gain error percentage is easily determined with Equation 2.7, where: A_{fs} is the full-scale output, A_{os} is the offset error, and $A_{ideal,fs}$ is the ideal full-scale output. Applying the all "1"s code to the DAC and measuring its output, after accounting for offset, determines the full-scale output.

$$GainError(\%) = 100\left(\frac{A_{fs} - A_{os}}{A_{ideal,fs}}\right) \tag{2.7}$$

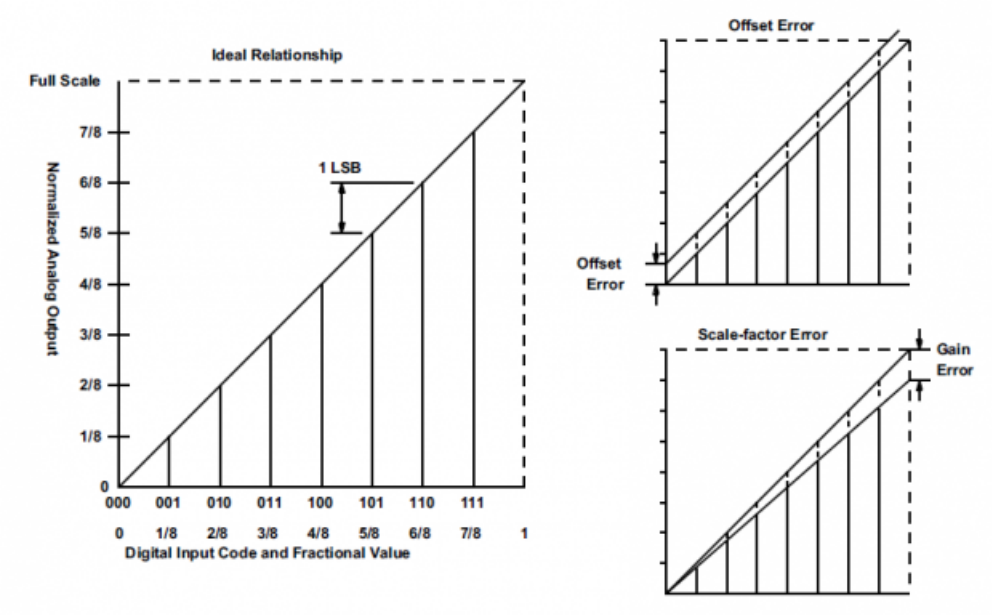

Figure 2.3: Example of Offset and Gain Error from atx7006.com (2014)

Differential Nonlinearity (DNL)

Differential nonlinearity is the maximum deviation of an actual analog output step. This measurement is take after offset and gain errors have been compensated or removed from the DAC transfer function. It is measured as the difference in DAC output levels from the ideal step value of +1 LSB between adjacent input codes. DNL is essentially localized error in the DAC's transfer function. If the differential nonlinearity is more negative than -1 LSB, the DACs transfer function is non-monotonic. Often the DNL metric is provided as worst-case- if the worst-case DNL is greater than +1 LSB or less than -1 LSB, then the DAC is not monotonic. The worst-case DNL errors for a binary weighted DAC occurs when all LSB elements turns-off and a MSB element turns-on (i.e. 0111 $\rightarrow$ 1000) (Manganaro, 2012).

$$LSB' = \frac{A_{fs} - A_{os}}{2^N - 1} \tag{2.8}$$

$$DNL_n = \frac{A_{out}(n+1) - A_{out}(n)}{LSB'} \tag{2.9}$$

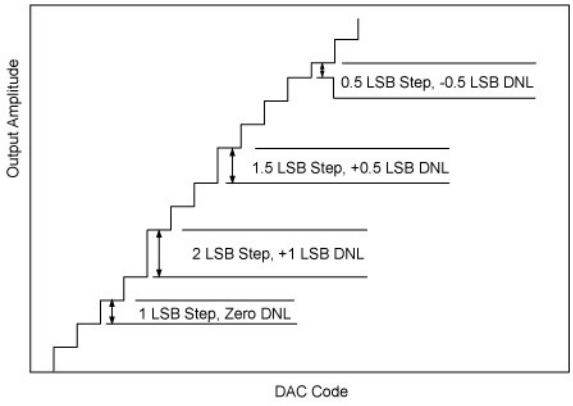

Figure 2.4: Example of DNL Error from atx7006.com (2014)

Integral Nonlinearity (INL)

Integral nonlinearity is the maximum deviation, at any point in the transfer function, of the output amplitude from its ideal value (after offset and gain errors have been removed). The transfer function of a DAC should ideally be a line and the INL measurement depends on the ideal line selected. Two commonly used lines are: the best fit line, which is the line that minimizes the INL result, and the endpoint line, which is a line that passes through the points on the transfer function corresponding to the lowest and highest input code. In all cases, the INL is the maximum distance between the ideal line selected and the actual transfer function (Manganaro, 2012). It is also worth mentioning that the INL of a binary weighted DAC has odd symmetry about the midpoint of the transfer function, i.e., the INL of any particular code is equal and opposite in sign from the INL of the complementary code.

$$INL_n = \sum_{i=0}^{n} DNL_i \tag{2.10}$$

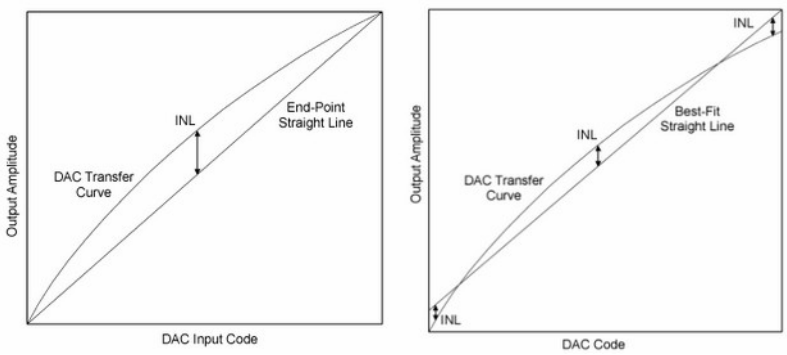

Figure 2.5: Examples of INL Error Detection from atx7006.com (2014)

2.4.3 Dynamic Figures of Merit

Total Harmonic Distortion (THD)

Total Harmonic Distortion of a signal is a measurement of the harmonic distortion present, and is defined as the ratio of the sum of the powers of all harmonic components to the power of the fundamental frequency. Equation 2.11 defines the calculation of THD, where V_1 defines the peak amplitude at the fundamental frequency and V_2 through V_n represent the peak amplitude of the harmonics. Measurements for calculating the THD of a DAC are made at the output. One of the most important factors in obtaining accurate distortion measurements is to ensure that the DAC output frequency, f_0, is not a sub-harmonic of the update rate, f_c. If f_c/f_0 is an integer, then the quantization error is no longer random, but is correlated with the output frequency. This causes the quantization noise energy to be concentrated at harmonics of the fundamental output frequency; thereby, producing distortion, which is an artifact of the sampling process rather than nonlinearity in the DAC (Kester, 2005).

$$THD_F = \frac{\sqrt{V_2^2 + V_3^2 + \ldots + V_n^2}}{V_1} \tag{2.11}$$

Signal-to-Noise Ratio (SNR)

Signal-to-noise ratio is defined as the power ratio between a desired signal and any undesired noise. SNR is mathematically defined in Equation 2.12.

$$SNR_{dB} = 20\log_{10}\left(\frac{A_{signal}}{A_{noise}}\right) \tag{2.12}$$

When a measurement is digitized, the number of bits used to represent the measurement determines the maximum possible signal-to-noise ratio. This is because the minimum possible noise level is set by the quantization of an analog signal. This noise is defined as quantization noise. The theoretical SNR is then mathematically defined by Equation 2.13.

$$SNR_{dB} \approx 20\log_{10}\left(2^n\sqrt{3/2}\right) \approx 6.02 \cdot n + 1.761 \tag{2.13}$$

Signal-to-Noise and Distortion Ratio (SINAD)

SNR can be adjusted to incorporate the power spectral density of harmonic distortion since it is not a desired signal component. This sets the stage for the definition of SINAD in Equation 2.14.

$$SINAD = 10\log_{10}\left(\frac{P_{signal} + P_{noise} + P_{Distortion}}{P_{noise} + P_{Distortion}}\right) \tag{2.14}$$

Effective Number of Bits (ENOB)

The effective number of bits can be calculated from SINAD. The mathematical relation with SINAD is defined in Equation 2.15.

$$ENOB = \frac{SINAD - 1.76}{6.02} \tag{2.15}$$

Spurious Free Dynamic Range (SFDR)

Spurious free dynamic range is the strength ratio of the fundamental signal to the strongest spurious signal in the output. SFDR is presented as a ratio of the RMS value of the carrier frequency (maximum signal component) at the output of DAC, to the RMS value of the second largest noise, or harmonic distortion component, or 'spur' at its output.

2.5 DAC Methods and Architecture

A particular DAC design may be appropriate for one application and completely inappropriate for another. Audio applications where digital data streams are converted into audio signals do not require a high-speed DAC; however, they do require very high resolution. On the contrary, video applications require a high-speed DAC with much less emphasis on resolution. Aside from signal processing applications, DACs have other applications such as digitally controlled voltage or current sources, waveform generators, op-amps offset nulling, oscillators, or even digitally programmable trim elements. Computationally, DACs are also used to perform various arithmetic operations such as: addition, subtraction, multiplication, and division involving analog and digital variables.

2.5.1 DAC Topologies

Voltage-Scaling

Voltage-Scaling architectures convert a reference voltage or current to a set of switched binary-weighted voltages. A switched resistor DAC is an example of this architecture. This style of design contains a parallel resistor network. Individual resistors in the network are enabled or bypassed based on the digital input. Any design implementing passive devices can face large parameter variation resulting in poor matching between the various resistors, which decreases linearity and resolution (non-monotonic) (Allen

et al., 1987). Linearity can then be guaranteed through additional calibration circuitry. Inherent advantages of voltage-scaling are its speed and insensitivity to switch parasitics.

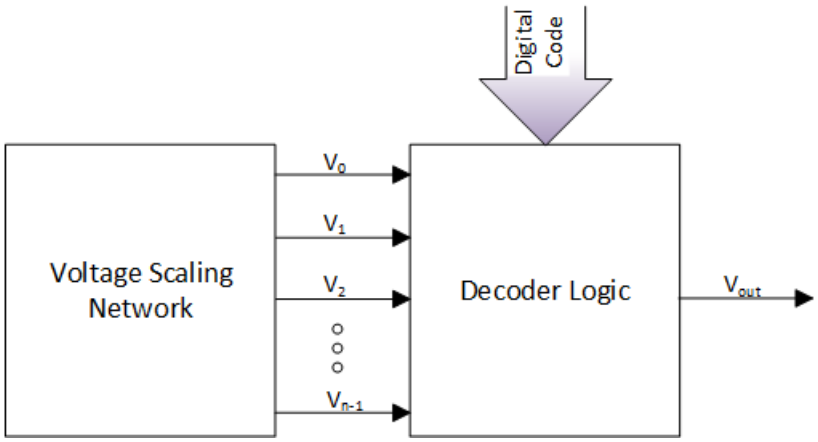

Figure 2.6: Basic Voltage Scaling DAC

Charge-Scaling

Charge-Scaling architectures convert a reference voltage to a set of binary weighted charges. The conversion is made possible by using a switched array of binary weighted capacitances to divide the total charge applied. Charge scaling can easily handle bipolar operation with signed digital words (Allen et al., 1987). Since the conversion is completed with capacitive elements, there is theoretically no DC power consumption, but other supporting electronics still consume DC power. This design is dependent on utilizing suitable capacitive materials; therefore, it requires a fabrication process with acceptable materials. The binary-scaled capacitors can account for large amounts of chip area unless some split array technique is utilized.

Current-Scaling

Current-scaling architectures convert a reference voltage or current to a set of switched binary-weighted currents. These DACs are essentially a set of binary-weighted current

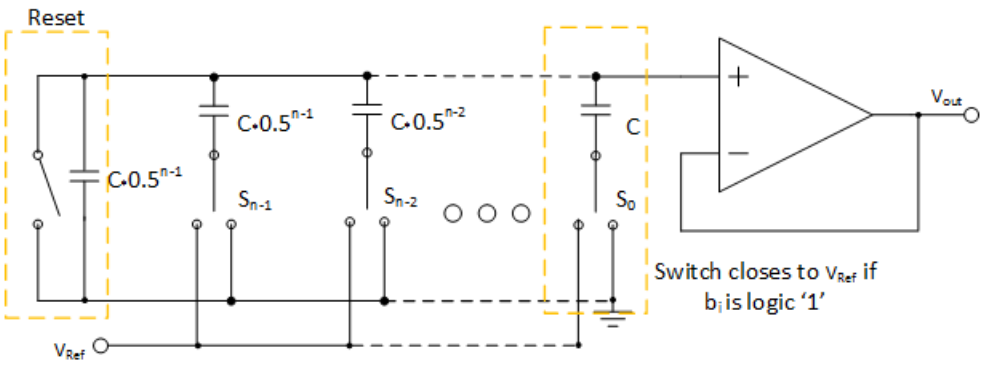

Figure 2.7: Basic charge scaling DAC

sources that are switched based upon an input code. Process variation creates poor matching between the binary-weighted current sources, which in turn decreases linearity and resolution (non-monotonicity) in the DAC (Allen et al., 1987). For improved current source matching and linearity, large area MOSFETS must be used. Section 2.5.2 discusses methods of encoding that promote linearity and guarantee resolution. Also, parasitic capacitances associated with MOS switches and current sources create glitches during switching.

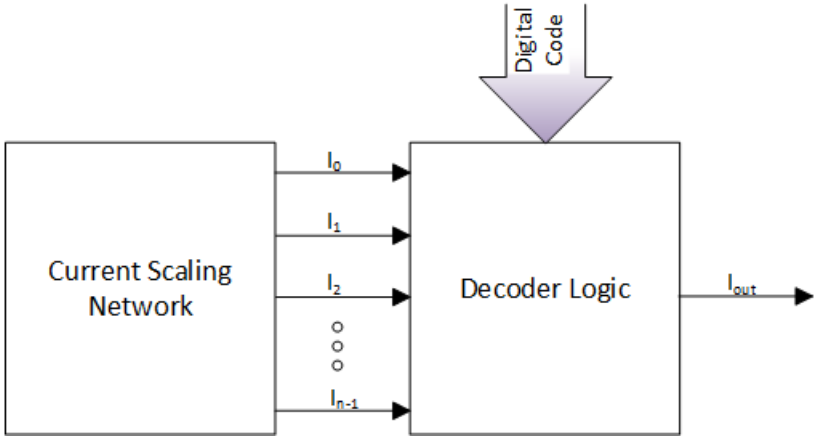

Figure 2.8: Basic Current Scaling DAC

A major advantage of a current steering DAC architecture is its inherently high current drive and high speed. Current-scaling DACs are typically utilized in

applications where there is less emphasis on resolution and where speed is critical. These DACs are great for the interface application at hand because of their readily available current mode output.

2.5.2 Binary Weighted vs. Thermometer Encoding

In binary weighted current steering DACs, monotonicity is only guaranteed if the current sources are closely matched. With parameter variation and systematic errors, a well laid out DAC is still far from monotonic due to large DNL errors. Binary weighted DACs suffer mostly from DNL errors; in fact the worst-case DNL error occurs when a LSB element turns-off and a MSB element turns-on (i.e. 0111 to 1000) Manganaro (2012).

Unary coding or thermometer coding is a method for which the DAC contains an equal current source for every possible output. For example, a fully thermometer coded DAC with n bits has $2^n - 1$ current sources. For every new bit-word, only one current source is switched, this alleviates the large glitch errors from multiple current sources switching (Lee et al., 2009). An example of thermometer encoding is provided in Table 2.1. Binary weighted DACs suffer from greater DNL errors than those of a thermometer DAC, because the current source weighting also provides a weight to mismatch current. For thermometer coding, deviations of actual unary sources from the nominal value will lead to errors in step size, and hence to DNL errors. The accumulation of such errors gives rise to INL errors. Even with thermometer coding, the accumulation of small DNL errors can lead to large INL errors. In statistical terms binary and thermometer architectures have very similar INL performance (Manganaro, 2012).

One major drawback of the thermometer-encoded DAC is area; for every LSB this architecture needs a current source, a switch, and an encoding circuit, as well as an encoder to translate binary codes to unary. To leverage the clear advantages of the thermometer-encoded architecture and to obtain a small area simultaneously, a

Table 2.1: 2-bit Binary to Thermometer Code Conversion Example

Decimal	Binary Code	Thermometer Code
0	00	000
1	01	001
2	10	011
3	11	111

compromise is found by segmentation. Mixed binary-weighted/thermometer-encoded implementations can be conceived that yield the DNL advantages of thermometer-encoded topology with the area efficient advantages of the binary weighted topology. The DAC is divided into two sub-DACs- one for the MSBs and one for the LSBs. Thermometer encoding is used in the MSB segment where the accuracy is needed most. As the percentage of segmentation is increased, the required total area is first dominated by the DNL requirement, then by the INL requirement, and finally by the encoding logic. Often, the system requirements for INL are more relaxed than for DNL (Lin and Bult, 1998).

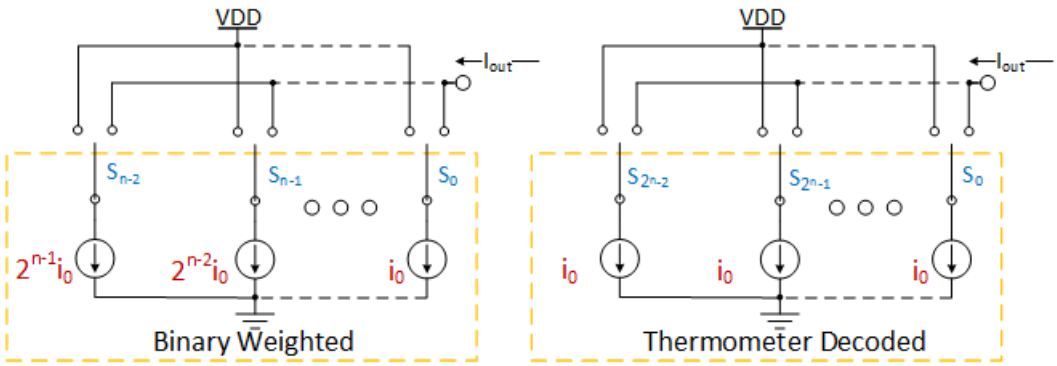

Figure 2.9: Comparison of both ideal binary weighted and thermometer decoded DACs

Chapter 3

Analog System Interface Design

This chapter provides a detailed top-level design of the proposed system interface. In Section 3.1, design of the interface segment responsible for writing data to the analog system is presented. Section 3.2 describes the design of a 'back-end' portion of the interface that is responsible for data to the microprocessor. Overview of the complete interface's operation and conclusion of the chapter is covered in Section 3.3. Timing and time delays are explained in Section 3.3.2. Section 3.3.3 describes the error budget for the system.

3.1 Proposed Front-End Architecture

The application requires the mapping of 8-bit digital vectors to analog vectors; therefore, there is not a need for an overly sophisticated protocol. Figure 3.1 provides illustration of the vector mapping.

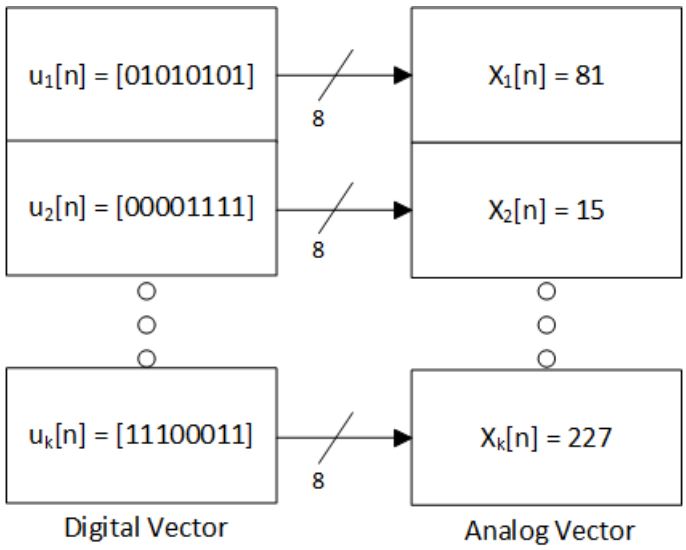

Figure 3.1: Analog to Digital Vector Mapping

There must be a communication protocol for the microprocessor to communicate with the interface. Communication protocols allow systems to use well-defined formats for exchanging messages. Static random access memory (SRAM) is a type of volatile semiconductor memory to store binary logic '1' and '0' bits. SRAMs have three types of lines, or connections: address, data, and control. The address lines are used to connect, or select, a memory location within the memory device. Data are written, or read, over data lines based upon the control signal interpretation. An example of an SRAM interface is provided in Figure 3.2. The typical SRAM connections are readily accessible on microprocessors making an appropriate interface suitable for the application. Since 8-bit digital vectors are written from a microprocessor, the typical interface of an SRAM is justified.

By mimicking the interface of an SRAM, the microprocessor's available connections consist of a pair of 8-bit parallel data and address buses (16 pins), along with read/write control pins (2 pins). For every address in the address space of the SRAM, 8 bits of data is written into an independent DAC. Essentially, the interface acts as an array of addressable DACs to translate digital vectors to continuous vectors. Figure

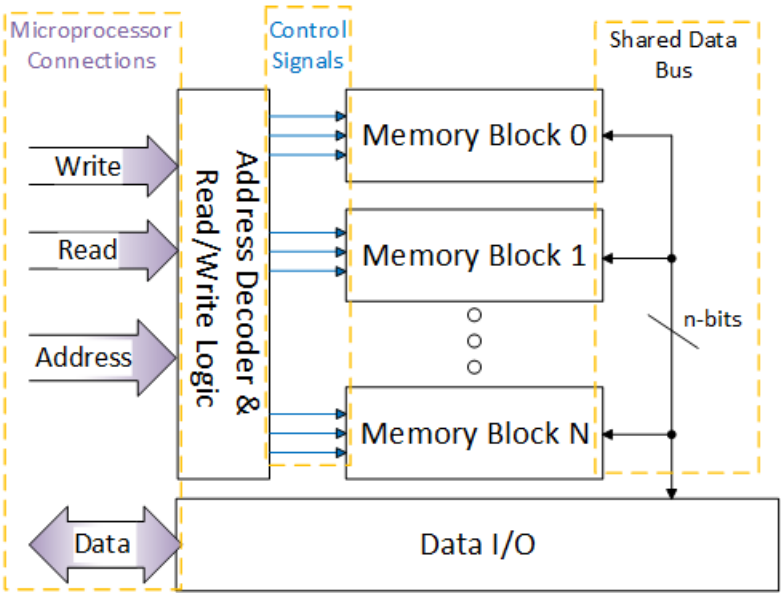

Figure 3.2: SRAM Interface Example

3.3 is provided to illustrate the addressable DAC array. The 8 bits of address space theoretically allow vectors of up to 2^8, or 256 elements, to be translated, which is more than sufficient for immediate applications. Multiplexing the data pins and address pins further provides 16 bits of address space, thus increasing the possible vector length to 2^{16} or 65,536 elements. The cost of utilizing the addressable interface of the SRAM is the additional hardware requirement of an address decoder and the additional system delay. Today's VLSI tools can ease half of the burden by easily synthesizing the address decoder.

Every DAC in the array shares a common 8-bit data bus with a smaller 'DAC interface'. By taking an 8-bit address from the microprocessor, the primary interface (also shown as "address decoder / data routing") in Figure 3.3 was responsible for routing the 8-bit data to the correct DAC interface corresponding to the address. This was accomplished by generating a bus of m '1-hot DAC select' signals to tell one individual DAC interface of the m DACs to latch the 8-bit data. When one of the DAC select signals is logic '1', the corresponding DAC interface latches in the 8-bit data from the bus. The '1-hot' signal was a control signal for an 8-bit register that

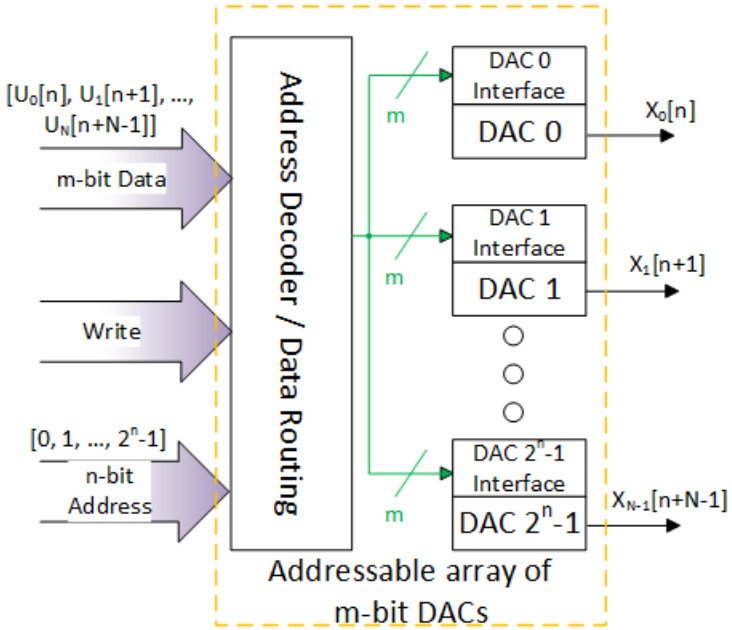

Figure 3.3: Utilizing DACs in an addressable array

exists in the DAC interface. Both the primary interface, and the DAC interfaces were purely digital and synthesized using VLSI tools.

DACs take up large amounts of silicon area; therefore, making the decision to have a physical DAC for every element in the vector a poor design choice. To reduce the number of DACs, elements of the analog output vector can share a DAC with multiplexing. For example, consider the vector mapping: $[\delta_0, \delta_1, \delta_2]' \rightarrow [\alpha_0, \alpha_1, \alpha_2]'$ where δ_i represents a digital element and α_i represents an analog element. Two DACs are used for this example— DAC0 and DAC1. Element δ_0 of the digital vector is latched to DAC0. Afterwards, δ_1 is latched into DAC1. At this point, DAC0 has converged and the resulting analog signal is sampled and held yielding element α_0 of the analog vector. Element δ_2 of the digital vector is then latched into the newly available DAC0. This process continues until all digital elements are converted into analog elements. A block-level diagram of the DAC multiplexing topology is shown in Figure 3.4. Processor connections in Figure 3.4 are denoted by large arrow signals from the left side of the Figure.

The primary interface now takes the 8-bit address and creates not just a '1-hot DAC select' but also a '1 hot channel select'. Like the example provided in Figure 3.3, the 1-hot DAC select signals tell the corresponding DAC interface to latch 8-bit data, but now the DAC interface also latches the '1-hot channel select' signals. The 'channel select' signals need to be stored in memory for the addressed DAC to provide current output to the appropriate sample-and-hold. To summarize, the primary interface communicates to each DAC interface what data it is responsible for converting and where it is sending its output.

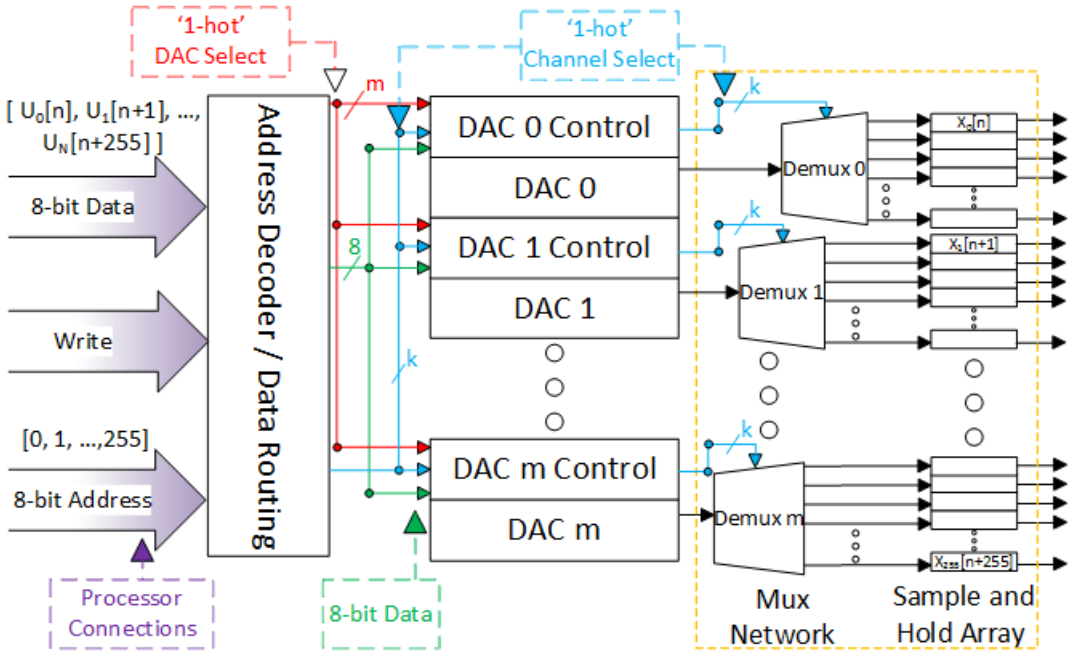

Figure 3.4: Multiplexing DAC outputs to minimize area and power consumption

By multiplexing DAC output channels, the overall chip area is reduced at an expense of increased total conversion time and design complexity. The number of channels per DAC is a decision dependent upon the DAC conversion time and the address decoder conversion time. The objective of this technique is never to have any particular circuit remain idle after a convergence. After all elements of the

digital vector are converted to continuous vector elements, they remain stored in their respective sample-and-holds.

In order to route and store DAC outputs into the different vector elements, or 'channels', a demultiplex operation is required. Demultiplex functionality is integrated into the sample-and-hold array to form a multi-channel sample-and-hold. A schematic of this design is shown in Figure 3.5. Transistors M1 and M2 are diode connected and have a sinking reference current directly from the DAC output. The gate connections of M1 and M2 are shared with 16 sets of 2 'Sample Switches'. These switches are controlled by 16 '1-hot' signals provided by a DAC controller as shown previously in Figure 3.4. Demultiplex functionality is provided by allowing only one set of switches to be on at any given time. When a set of switches is turned on, a current mirror is formed between transistors M1 and M2 and the transistors on the other side of the given switch. The drain current of M1 and M2 is mirrored to the gate connected transistors. For example, consider the pair of switches that provide connections from transistors M1 and M2 to transistors M3 and M4. When these switches are on, transistors M3 and M4 then share the source-gate voltages of transistors M1 and M2 respectively, and form a cascode current mirror.

p-type multi-channel Sample-and-Hold Design

For the DAC multiplexing scheme presented in Figure 3.4 to work, an analog memory must store elements of the analog vector so that the DACs can convert other elements. From the previous example involving Figure 3.5, transistors M1, M2, M3, and M4 created a current mirror when their corresponding switch was switched on. When their switch is switched off, it enters a high-impedance state that severs the current mirror gate connections. Ideally, the source-gate voltage of transistors M3 and M4 is preserved by the trapped charge on their gate-oxide, causing the transistors to continue producing their 'sampled' drain current. Transistors M1 and M2 could then form a current mirror with transistors M5 and M6, while transistors M3 and M4 maintain their sampled current. The sampled gate-oxide charge should be stored

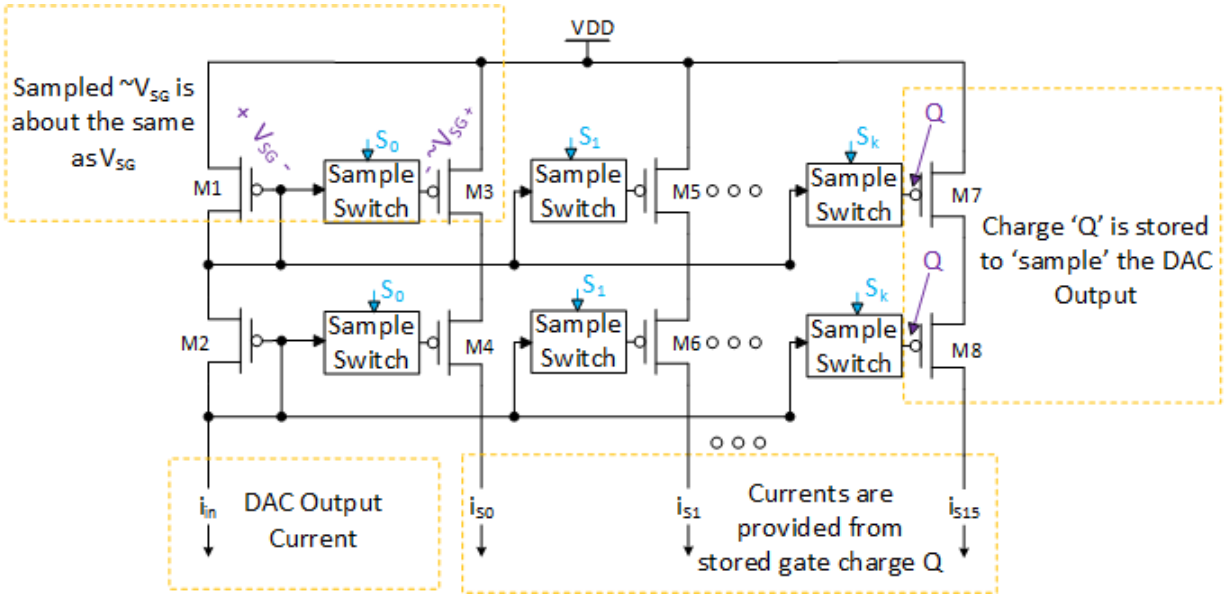

Figure 3.5: Multi-Channel p-Type sample-and-hold design

indefinitely, but in reality there are subthreshold leakage currents through both the transmission gate and gate-oxide that alter the stored charge. Over time, the charge can be altered resulting in an inaccurate sampled current. This leakage current sets limitations for the minimum sampling rate.

pFET Switch Design

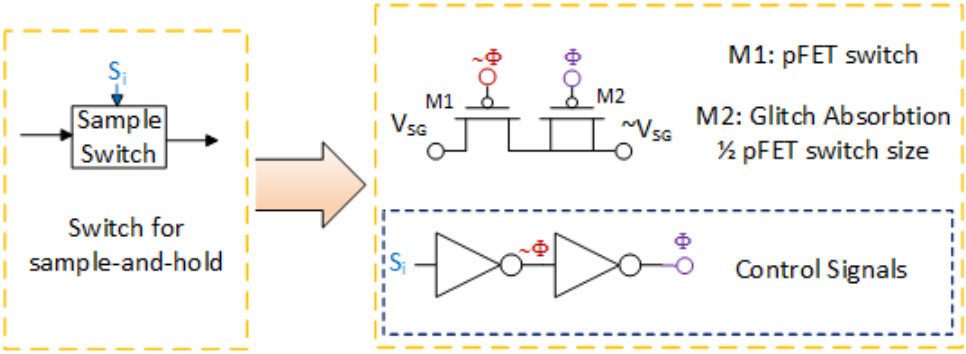

Figure 3.6: p-Type switch with charge injection compensation used in sample-and-hold

A transistor-level schematic of the sample switches is provided in Figure 3.6. Transistor M1 is a pFET switch that turns on when its corresponding '1-hot' signal is logic '1'. When the signal is logic '0' the pFET enters a high-impedance state. A cause of sampled current error can result from control signal feed-through. Switching digital signals couple through the gate-drain and gate-source capacitances of switch FETs. This forces charge injection onto the sensitive charge storing node. To compensate for this, MOS capacitor M2 is placed to absorb the feed-through signal. When the control signal switches, it feeds through M1's gate-oxide capacitance and onto the sensitive charge-storing node producing a 'pedestal' error. MOS capacitor M2 is driven by an inverted system update signal to eliminate the control signal feed-through effect. Theoretically, when the system update signal feeds through, half of the total produced charge is injected out of the drain, and the other half is injected out of the source. The total produced charge is proportional to the size of the transistor (Wegmann et al., 1987). Because of this, M2's geometry is half the size of M1. This might appear to be a cure-all; however, the additional parasitic capacitance increases the total rise time of the sample-and-hold, limiting the maximum sampling rate for a given current input.

The analog signals cannot be sequentially introduced into the system with varying delay; they must all be simultaneously introduced into the analog system. This requires an additional sample-and-hold for every analog element to synchronize the signals. Therefore, a "second line" of sample-and-holds was introduced at each of the DAC outputs after the existing sample-and-holds. An updated block diagram with the additional sample-and-holds is provided in Figure 3.7. For synchronization purposes, an additional control signal 'update' common to all of the second-string sample-and-holds is required from the microprocessor. This signal is driven to logic '1' and held high after the digital vector has been converted. If an additional pin from the microprocessor is physically unavailable, the control signal may be sent internally, with some slight design modification, from an addressable 1-bit register.

The addressable register could then be accessed and written to from the proposed 'SRAM' interface.

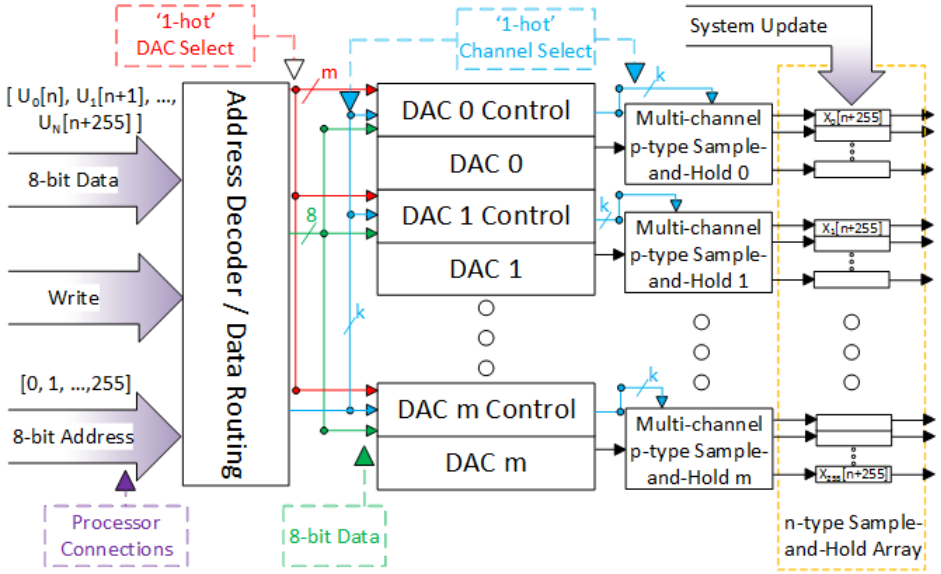

Figure 3.7: Front-end interface topology completed with additional sample-and-hold array for synchronization purposes

The 'first' sample-and-hold in the system is p-type; therefore, it sources an output current to each of its channels. A second cascaded p-type sample-and-hold would not function properly because it requires a sinking input reference current and the first sample-and-hold sources its output current. This would lead to bias issues, and the two blocks would fail to operate. Because of this, an n-type sample-and-hold needed to be designed. A schematic of the n-type sample-and-hold is provided in Figure 3.8. The functionality of this block is very similar to that of the p-type except it sinks current at the input and sources current at the output. Transistors M1 and M2 are diode-connected and their drain current is provided from the first sample-and-hold. M1 and M2 are matched with M3 and M4. When the switches are on (when system update is logic '1'), M1 and M2's gate-source voltage is common to M3 and M4 to force them into mirroring the input drain current. When the switches turn off (when system update is logic '0'), M3 and M4's gate-source voltage is maintained

from charge remaining on their gate-oxide. Ideally, this stored charge allows M3 and M4 to produce the same drain current even if M1 and M2's drain current changes. The drain current from transistor M3 provides the reference current to the diode-connected transistors M5 and M6. Transistors M5 and M6 are matched with M7 and M8 to form a pFET cascode current-mirror. Transistors M7 and M8 mirror the sampled reference current into the analog system.

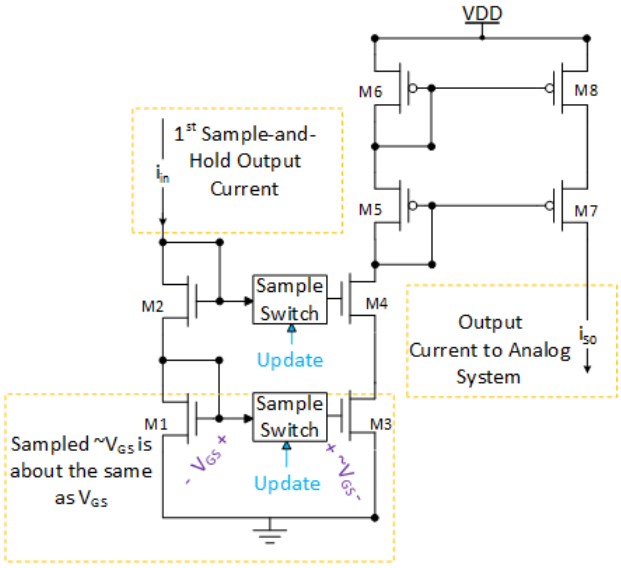

Figure 3.8: n-Type Sample-and-hold

A schematic of the nFET switch used in the n-type sample-and-hold is shown in Figure 3.9. Transistor M1's gate is driven directly by the system update signal. When the signal is logic '1', the switch is turned on to allow a common gate-source voltage. When the signal is logic '0', the switch is turned off to sample the gate-source voltage. When the system update signal switches, it feeds through M1's gate-oxide and onto the sensitive charge-storing node producing a 'pedestal' error. MOS capacitor M2 is driven by an inverted system update signal to eliminate the control signal feed-through effect. When the control signal toggles, theoretically, half of the total charge injection is injected out the drain, and the other half is injected out of the source.

The total produced charge is proportional to the size of the transistor. Because of this, M2's geometry is half the size of M1 to absorb charge feed through.

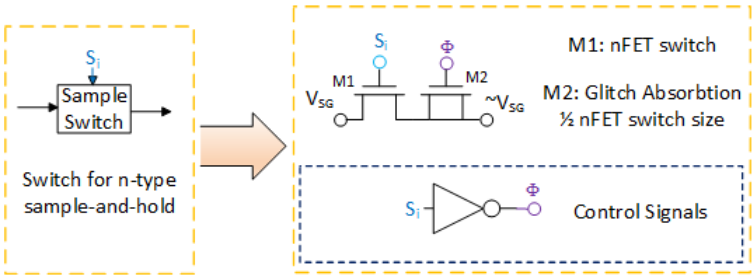

Figure 3.9: nFET switch design

3.2 Proposed Back-end Architecture

To complete the microprocessor interface, output from the analog signal processing system must be obtained. For feedback to be possible, the analog system's output vector must be digitized by ADCs. The back-end architecture is similar to the proposed front-end design discussed in Section 3.1; except, of course, that is opposite in operation. The back-end is designed such that a continuous vector with a number of elements is converted to a digital vector and temporarily stored so that the microprocessor can asynchronously read the digital vector element by element. Like the front-end, the back-end is designed to be easily scalable. The output microprocessor interface is treated as a separate SRAM, requiring the complete interface to take two sets of SRAM connection. This scheme (illustrated in Figure 3.10) allows simultaneous read/write capabilities. With this topology, it is possible to be reading a converted digital vector from the analog system's output while the next digital vector input is converted into a new analog vector and prepared to enter the system.

Similar to demultiplexing DAC outputs, the analog system's outputs are multiplexed among a finite number of SAR ADCs to minimize chip area. A proposed block diagram is shown in Figure 3.11. In this topology, there is no need for a microprocessor

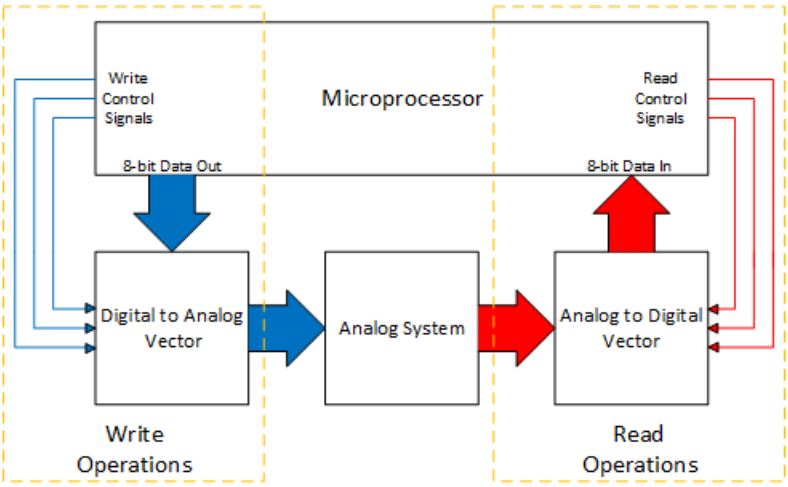

Figure 3.10: Illustration of parallel read/write capability

addressable memory for the ADC outputs. Instead, the ADC outputs are multiplexed to the microprocessor's interface outputs, allowing an ADC's output to be read while the previous ADC processes a new input. Design of the SAR ADC was complete by colleague Liu Peixing. Taking a voltage-mode analog signal, the SAR ADC produces an 8-bit digital representation after some conversion time. The SAR ADC design requires an off-chip clock signal with a period of $50ns$ for their operation.

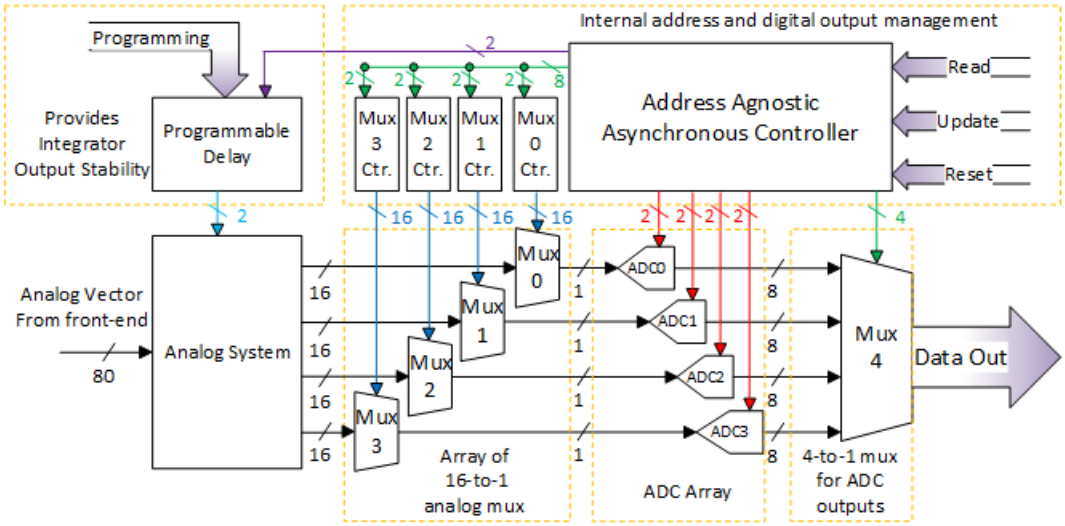

Figure 3.11: ADC Inputs are multiplexed to minimize area and power consumption

To illustrate the desired operation, consider the following example with 2 multiplexed ADCs (ADC0 and ADC1) instead of the 4 shown in Figure 3.11. Assume an output analog vector, consisting of 4 analog elements, $[\beta_0, \beta_1, \beta_2, \beta_3]'$, needs to be mapped to the digital vector of 4 digital elements, $[\delta_0, \delta_1, \delta_2, \delta_3]'$. First, it is assumed that the analog system's outputs converge simultaneously after some conversion time. After conversion, β_0 and β_1 are immediately latched into ADC0 and ADC1 respectively. The ADCs both converge after some time, yielding δ_0 and δ_1. The microprocessor issues its first read command (the first read command after the output has converged from a new vector of inputs). By default, the output multiplexer, shown to the immediate left of the microprocessor in Figure 3.11, is set to route ADC0's output or δ_0 to the microprocessor. With the second read command, the output multiplexer selects ADC1s output, δ_1, to be routed to the microprocessor. During the second read command, ADC0's input was changed from β_0 to β_2 and begins the conversion to δ_2. The pattern is continued until all elements of the digital vector have been read into the microprocessor in order.

The back-end's vector output is adamantly sequential. That is, if the internal address is set to $0x05$ and the microprocessor issues a read command requesting the digital element at address $0xFF$, it will receive the element at address $0x05$ + $0x01$ = $0x06$. Each 1-to-n multiplexer has an independent sub-controller that controls the '1-hot' signals. The sub-controller itself receives two different signals: an increment signal and a reset signal, both provided from the primary interface. When an increment signal is received, the sub controller increments its internal address by one and routes the next sequential channel into the ADC. When an ADC receives a reset signal, the internal address is reset to zero, and the default channel is routed to the ADC input.

The primary controller is interfaced directly with the microprocessor and manages all read operations. It contains counters and logic such that when a microprocessor read signal is received, the ADC responsible for providing the next element is routed to the output bus, and the previous ADC receives a new analog input to convert. The

internal counters maintain correct routing and provide a correct mapping of analog output elements to digital elements. ADC outputs are routed to the output bus by a digital multiplexer whose '1-hot' control signals are sent from the primary interface. A single ADC begins conversion when it receives an 'Enable' signal sent from the primary interface. A timing diagram to illustrate how the blocks operate in harmony is provided in Figure 3.12.

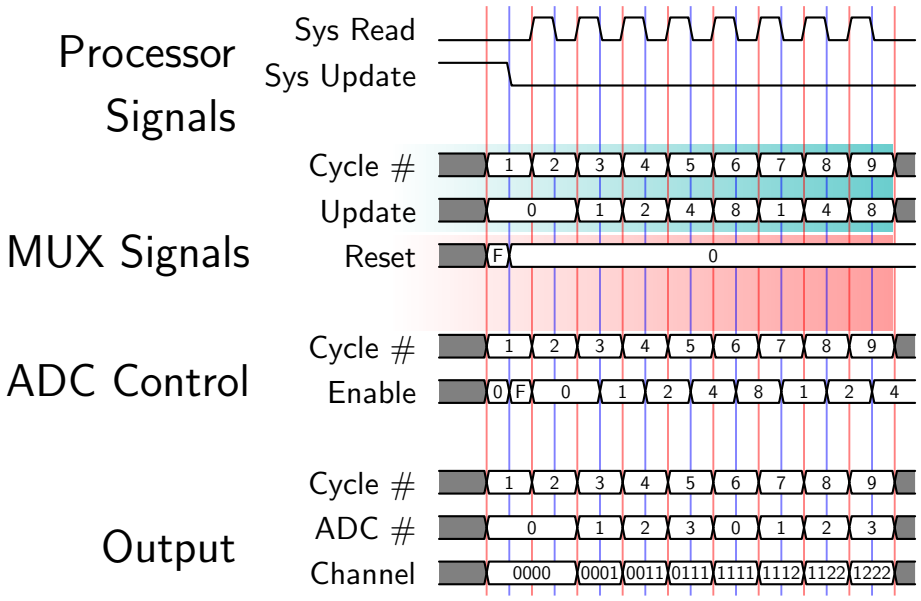

Figure 3.12: Back-End Interface Timing Diagram

The analog system's output is a voltage-mode vector produced from integrators. For output stability, the integrators require control signals for timing. The integrators need two control signals for 'integrator reset' and 'integrate' capabilities. These integrators reset when a new vector input is provided, and they stop integrating the current signal after some finite time. A schematic of the integrator and switches are presented in Figure 3.13. If the integrators are not properly controlled their outputs can lose integrity.

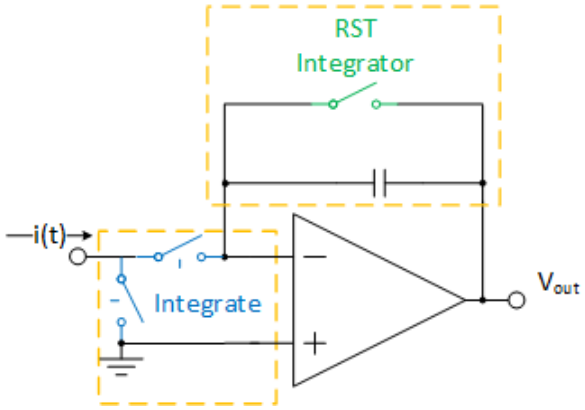

Figure 3.13: Analog System Integrator Output

Integrator control signals are generated by a digital programmable delay that was synthesized using VLSI design flow tools. The programmability offers flexibility during testing. Utilizing the SAR ADC's 20 MHz clock signal as a reference, internal counters increment until the programmed delays have been reached. Once a specified delay has been met, an appropriate control signal is driven to either logic '1' or logic '0'. There are three programmable delays. The first delay pertains to both 'integrate' and 'reset integrate' signals and the time for which when they toggle after system update switches from logic '0' to logic '1'. The second delay specifies how long the 'integrate signal' is held at logic '1' before switching back to logic '0'. The third and final delay specifies the time duration 'reset integrator' is held at logic '0' after the system update switches to logic '0'. The description of these delays are shown in Table 3.1. When the system update signal is driven to logic '1' the whole process starts again. The ADC sample signal is driven to logic '1' on the falling edge of the system update signal, and this is when the ADC's begin to convert their analog input. A timing diagram to illustrate the control signals and their delays is shown in Figure 3.14.

Table 3.1: Programable Delay Signals

Delay	Description
Delay 1	When integration begins after system update rising edge
Delay 2	How long integration lasts
Delay 3	When integrators reset after system update falling edge

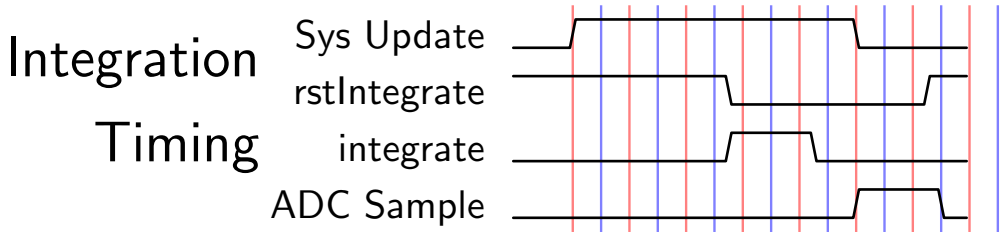

Figure 3.14: Programmable Delay and Integrator Control Timing

An analog multiplex is required for the ADC multiplexing scheme. A schematic of the multiplex construction is provided in Figure 3.15. The design is the same as a demultiplex, except opposite in functionality.

Analog '1-Hot' Mux Design

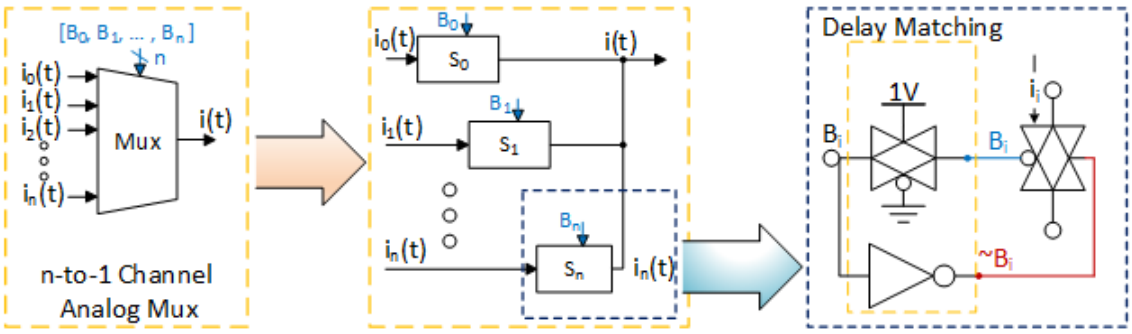

Figure 3.15: Analog '1 hot' mux design

3.3 Analog Interface Operation Summary

A complete block diagram of the interface system including the analog system is provided in Figure 3.16.

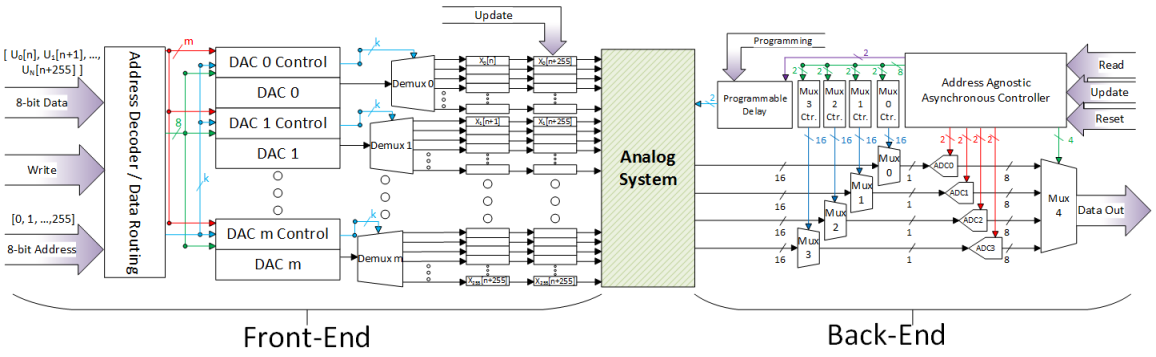

Figure 3.16: Complete schematic of proposed interface

3.3.1 Operation Example

A timing diagram for the functionality of the interface front-end is presented in Figure 3.17. Operation of the interface begins with the microprocessor writing the digital vector. For simplicity, let us assume that there are 5 DACs, and the analog system

requires a vector of 80 elements. This results in a multiplexing scheme of 16 elements per DAC. First, the first 8-bit element is written to address $0x00$ on a write command. The 8 bits are then latched into DAC0 for conversion. Then, the second 8-bit element is written to address $0x01$ on another read command. These new 8 bits are then latched into DAC1 for conversion. This pattern continues until the 5th element is written. When the 5th element is written to address $0x04$, the 8 bits are placed in DAC4. By this time, DAC0 should have converged. Now the 6th element is to be written to address $0x05$. On the rising edge of the write command, DAC0's analog output is sampled and stored to address $0x00$. After the analog output has been stored to address $0x00$, the 6th element's 8 bits are latched into DAC0. The writing continues until the 7th element is written to DAC1. DAC1's analog output is stored at address $0x01$, and shortly afterwards the 7th element's 8 bits are latched in for conversion. After all 80 elements have been converted, the microprocessor issues a rising edge sensitive update command that simultaneously updates the analog system with the new continuous vector of signals. The update command is kept high afterwards.

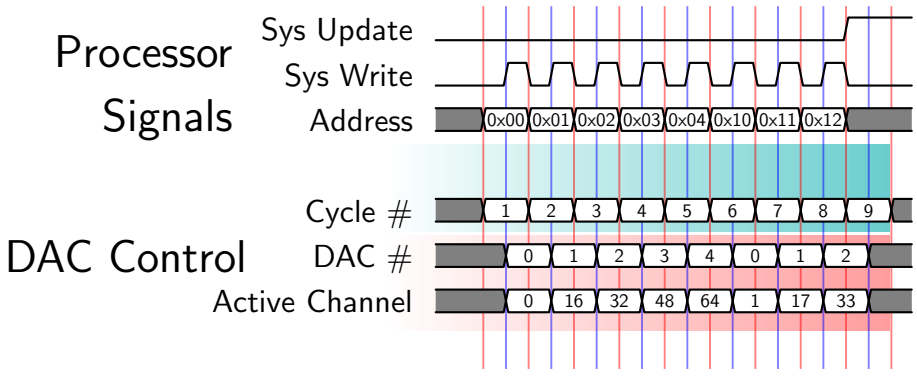

Figure 3.17: Timing of Front-End Interface

The analog system, after being updated with a new input, is expected to converge after some time interval. Another timing diagram of the system back-end is shown in Figure 3.18 for convenience. In the meantime, the next vector of microprocessor

inputs can immediately begin conversion. After the analog system converges, its analog voltage outputs are ready to be converted to digital signals and read into the microprocessor for feedback. For the sake of simplicity again, let us assume the same scheme— there are 5 ADCs with 80 analog signals (16 signals per ADC). This time the process begins with the microprocessor waiting for a finite period of time until the analog system converges. Once the system converges, the system update signal is toggled from logic '1' to logic '0'. On the falling edge of the update command, all ADCs begin converting their first elements of the system's output vector (ADC0 converts the 1st element, ADC1 converts the 2nd element, ...). The microprocessor then issues a read command, which signals the first element is to be read. On the rising edge, ADC0's 8-bit output is latched to the output bus. Then a second read command is issued by the microprocessor, and on its rising edge, ADC1's 8-bit output is latched to the output data bus. In contrast to the first read command, on the falling edge of the 2nd read command ADC0's input is switched to the 6th element of the output vector. And almost exactly like the front end, the pattern continues until after all 80 elements have been converted and read into the microprocessor. To allow this external address agnostic operation, digital blocks with counters are used to manage the vector mapping from the system output to the processor. This scheme reduces the amount of chip area by minimizing the number of ADCs and reducing the need on chip storage. Afterwards, the update command is driven high and on its rising edge the analog system is updated with new inputs. What was not mentioned previously is that, upon that rising edge, the digital counters and control circuitry for the back-end is reset.

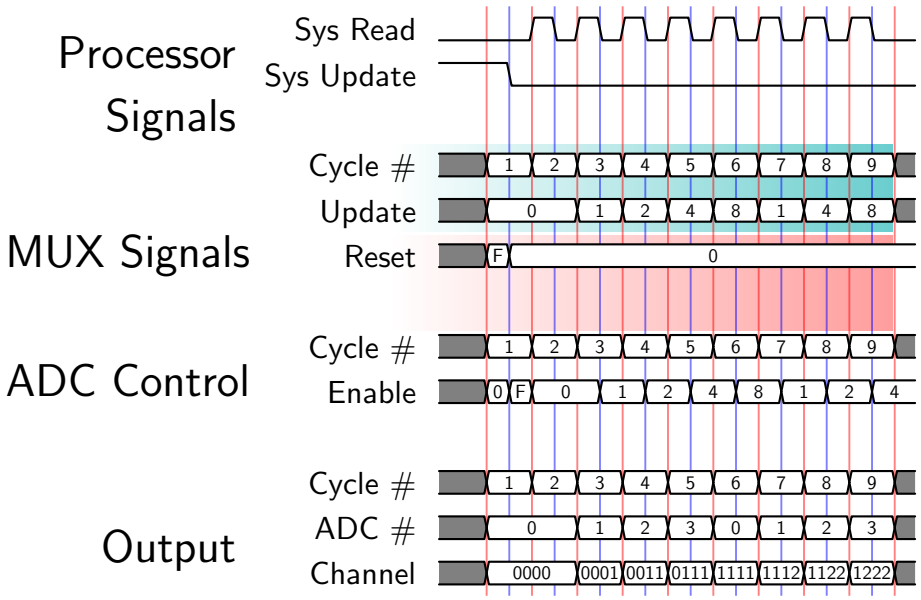

Figure 3.18: Back-End Interface Timing Diagram

3.3.2 Time Delay Definitions

The system operation is broken down into several delays. The first delay represents the time required to prepare a new analog vector with all of the elements stored in the first sample-and-hold. It is defined in Equation 3.2, where Δt_{DAC} is the conversion time of a single DAC, and Δt_{s1} is the convergence time of the first sample-and-hold. For a given element the DAC and first sample-and-hold are cascaded together, so the total delay is calculated as the greater of the two convergence times.

$$\tau_{new} = 16\big(\max(\Delta t_{DAC}, \Delta t_{s1})\big) \tag{3.1}$$

The second delay is defined as the time it takes a newly created analog vector to physically enter the analog system after the update signal toggles to logic '1'. This is mathematically represented by Equation 3.2, where Δt_{s2} is the convergence time of the second sample-and-hold.

$$\tau_{enter} = \Delta t_{s2} \tag{3.2}$$

As previously discussed, following the convergence of the analog system the back-end of the chip will begin its operations. The convergence of the analog system is denoted by the falling edge of the update signal. The third delay of interest will be defined by the period in which the update signal remains at logic '1'. This delay is represented in Equation 3.3, where Δt_{s2} is the convergence time of the second sample-and-hold and Δt_{sys} is the convergence time of the analog system.

$$\tau_{update} = \Delta t_{s2} + \Delta t_{sys} \tag{3.3}$$

After the update signal's falling edge, the analog system's output is to be read by the microprocessor element-by-element. The total delay time for the jth 8-bit element to be presented and read by the microprocessor is defined in Equation 3.4, where Δt_{ADC} is the ADC convergence time, $\Delta t_M UX, i$ is the switch time of any of the first 4 multiplexers, and $\Delta t_{MUX,5}$ is the switch time of the ADC output multiplexer. Recall that on the update signal's falling edge all ADC's sample their default 0th channel input. Therefore, the first element's read time delay is limited by a single ADC conversion time Δt_{ADC}. For the remaining 63 elements, the ADC's update only after their output has been read. This delay has the assumption that the microprocessor read time is negligible.

$$\tau_{out,j} = \begin{cases} \Delta t_{ADC} + \Delta t_{mux5}, & : j = 0 \\ \frac{\Delta t_{ADC} + \Delta t_i}{4} + \Delta t_{mux5}, \quad \forall i \in [0,4] & : j \in [1,63] \end{cases} \tag{3.4}$$

The total time for a single digital vector to be converted, injected into the analog system, and the system's response read back into the microprocessor is defined in Equation 3.5. A more complete mathematical definition for infinite input vectors is provided in Equation 3.6. The first vector is written, converted, stored, and injected into the analog system; during the delays that proceed after the first vector's injection,

a second vector is written and prepared to enter the system. This pattern continues for all consecutive vectors.

$$\tau_{total} = \tau_{new} + \tau_{update} + 64\tau_{out} \tag{3.5}$$

$$\tau_k = \begin{cases} \tau_{new} + \tau_{enter} + \tau_{update} + 64\tau_{out} & : k = 0 \\ \tau_{enter} + \tau_{update} + 64\tau_{out} & : k \in [1, \text{inf}) \end{cases} \tag{3.6}$$

3.3.3 Error Budget Analysis

From the first digital to analog conversion to return of the analog system's response, there is potential for error at every stage in between. Error functions are derived to model the worst case error throughout the system. These function are normalized to 1 LSB. The error function for a single vector element entering the analog system is mathematically defined in Equation 3.7; ϵ_{DAC} is defined as error from the DAC, $\epsilon_{s,1}(t)$ is error introduced from the first sample-and-hold, and $\epsilon_{s,2}(t)$ is error introduced from the second sample-and-hold. The sample-hold-error $\epsilon_{s,2}(t)$ is further defined in Equation 3.8 where α is pedestal error caused by control signal feed-through or charge injection at $t = t_0$ and $\beta(t - t_0)$ is error from subthreshold leakage current at $t > t_0$. The sample-and-hold samples at time t_0.

$$\epsilon_{front}(t) = \epsilon_{DAC} + \epsilon_{s,1}(t - t_0) + \epsilon_{s,2}(t - t_1) \tag{3.7}$$

$$\epsilon_{s,i}(t - t_0) = \alpha + \int_{\tau_0}^{\tau_{stop}} \beta(t - t_0), \quad i \in [1, 2], \quad \beta(t - t_0) = \begin{cases} 0 & : t - t_0 < 0 \\ \beta(t - t_0) & : t - t_0 \geq 0 \end{cases} \tag{3.8}$$

Each element in the analog system's response vector can obtain error before being read into the microprocessor. Assuming that the output of the converged analog system does not change with respect to the analog system input, the back-end error

function is defined in Equation 3.9. The error ϵ_{sys} is defined as any error from analog system and ϵ_{ADC} is any error in the conversion process. This error function does not take into account the time-varying error from the sample-and-holds. The total error introduced into a single element from initial entry into the front-end interface to the final readout from the back-end is defined in Equation 3.10. This error function assumes that the output of the analog system is sensitive to a time-varying input until the integrators stop integration. The time constants τ_0 and τ_1 are defined as the time at which the first sample-and-hold 'samples' and second sample-and-hold 'samples' respectively. τ_{stop} is defined as the time for which the analog system's integrators stop integrating. For accuracy guaranteed to be within 1 LSB, $\epsilon_{element} \leq 1$ must be satisfied.

$$\epsilon_{return} = \epsilon_{sys} + \epsilon_{ADC} \tag{3.9}$$

$$\epsilon_{element} = \epsilon_{DAC} + \left(\alpha_1 + \int_{\tau_0}^{\tau_{stop}} \beta_1(t)\, dt\right) + \left(\alpha_2 + \int_{\tau_1}^{\tau_{stop}} \beta_2(t)\, dt\right) + \epsilon_{sys} + \epsilon_{ADC} \tag{3.10}$$

Chapter 4

DAC Design and Simulation

This chapter describes the design and simulation of two related but separate current steering DAC topologies. In section 4.1, the initial topology, a binary weighted design, is presented. The binary weighted DAC's simulation results are presented in 4.2. Section 4.3 then improves upon the binary weighted design by presenting a segmented design. The simulation results of the segmented DAC are shown in section 4.4. Section 4.5 concludes the chapter with a discussion of the simulation results and how it applies to the application.

4.1 Initial design of DAC

4.1.1 Binary Weighted Current Steering DAC

The design of the current steering DAC consisted of a set of binary weighted current sources and a set of switches; one switch for each bit $b_i, i \in [0, n-1]$ where n is the total number of bits represented by the DAC. The set of current sources is realized by a set of MOSFET current mirrors. Consider a NMOS current mirror where a reference current is provided and each mirror is constructed using NMOS transistors. The MOSFETs in each current mirror have the same geometry and physical characteristics; however, the number of MOSFETS per current mirror is

scaled. The parameter multiplicity is then defined to represent the number of physical MOS transistors in parallel. A MOS transistor with multiplicity m has m copies of itself where the drain, source, and gate of each transistor are commonly connected. A schematic of the connection defining multiplicity is provided in Figure 4.1.

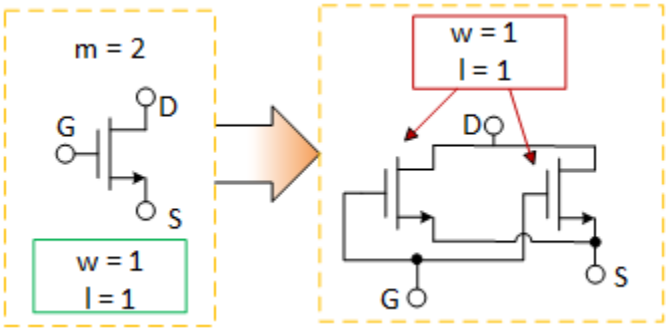

Figure 4.1: Transistor multiplicity

4.1.2 Current Sink Network and Standard Cell

A parallel combination of MOSFETs allows sinking or sourcing currents to be summed at the drain or source node respectively, thereby multiplying the current provided by one transistor by its multiplicity. That is, a binary-weighted current mirror pertaining to the MSB (b_{n-1}) has a multiplicity that is twice that of bit b_{n-2}; where the current source multiplicity function is defined as a geometric progression: $m_n = 2^n, \forall n \in [0, N-1]$. The current sinking network for the binary weighted DAC is shown in Figure 4.2.

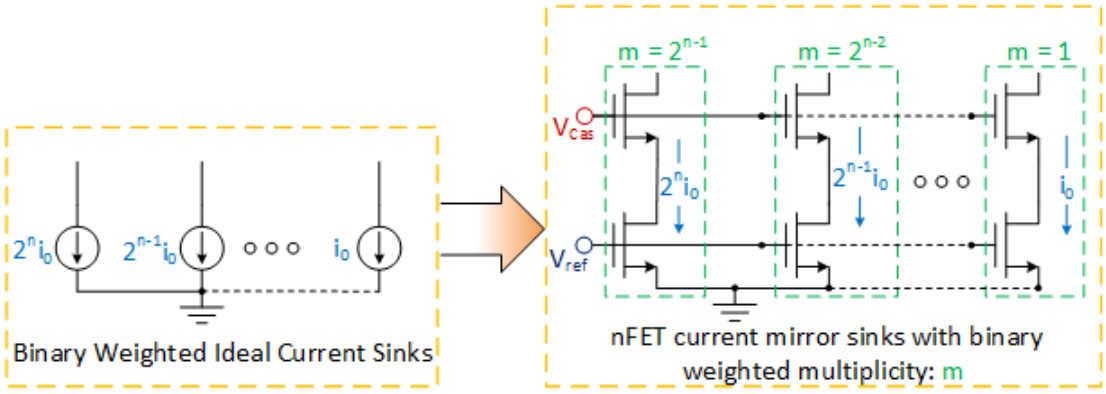

Figure 4.2: Binary weighted current steering DAC

There are 8 scaled current paths, and each path has a current mirror nFET (bottom) and a cascode nFET (topmost). In physical layout, both nFETs are laid out together in a standard cell that is surrounded by a guard ring. The current sinking network is then constructed with an organized square array of these standard cells. Standard cells are then connected into the current branches with various metal layers, based upon the multiplicity of the branch. The current sink standard cell layout is provided in Figure 4.3.

To reduce current mismatch from process variation, the current branches were geometrically connected such that every current branch had the same common-centroid. The common-centroid topology is provided in Figure 4.4. In Figure 4.4, the gates of M1 and M2 for each standard cell are connected by a metal layer to every other standard cell's M1 and M2 gates as previously shown in Figure 4.2. However, the drains of the cascode transistors were connected to drains corresponding to the same current branch to form 8 current summing nodes (i.e. A's drains connect, B's drains connect...). The 8 current summing nodes are then routed to their respective switches. The binary-weighted topology design set an ideal MSB current of $10nA$, and an ideal LSB current of $78pA$.

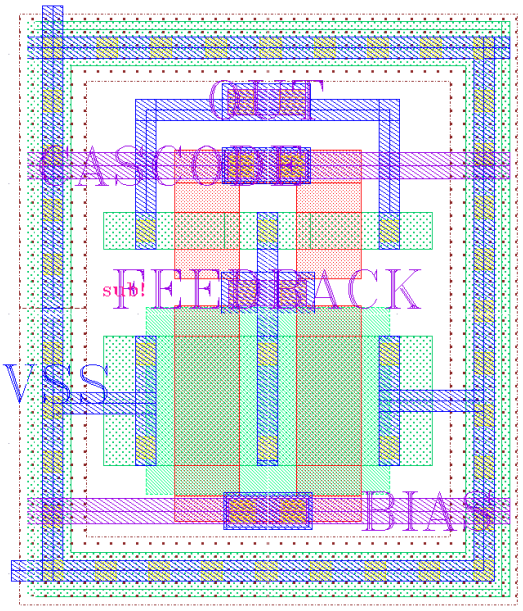

Figure 4.3: Layout of Current Sink Standard Cell

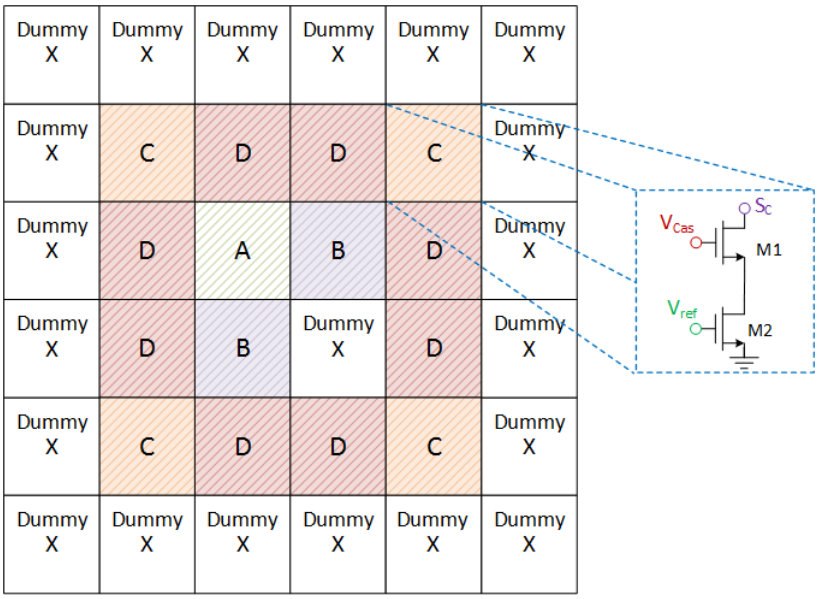

Figure 4.4: Common-Centroid Layout

Low currents and subthreshold operating conditions made current-mirror biasing difficult to match. An operational transconductance amplifier (OTA) is used in a

cascode current sink configuration to maintain proper bias conditions and accuracy. The purpose of the additional OTA minimizes current mismatch in the current mirror by equalizing drain-source voltages. An example of an OTA cascode connection is provided in Figure 4.5. Through negative feedback, the OTA forces transistor M3 to have the same gate-source and drain-source voltage as transistor M1.

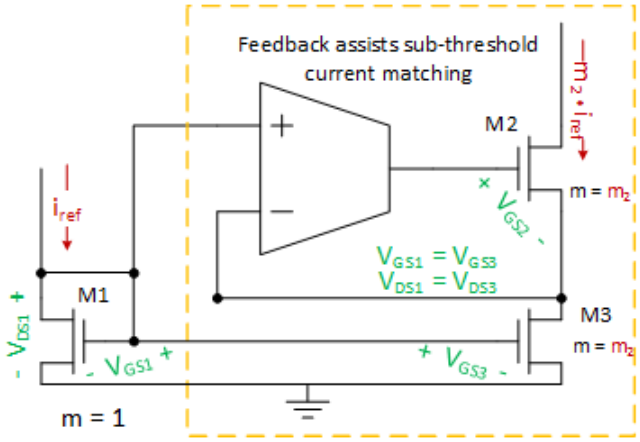

Figure 4.5: Cascode Current Mirror Connection

For the DAC design, negative feedback is obtained from the sources of all parallel MOSFETS responsible for the MSB's current. Output from the OTA is applied to every MOSFET gate in the current mirror network. The reference current for the cascode current mirror is provided from off-chip. Placement of the OTA in the binary weighted DAC topology is provided in Figure 4.6. Feedback was only taken from the MSB current sink node originally because it was the most influential for static performance; and there was an original assumption that the other branches would behave the same. Unfortunately this was not the case as it caused bias mismatch especially for digital input codes where the MSB was off. Section 4.3 discusses a modified design where the OTA is repositioned, such that negative feedback is taken collectively from all of the current branches.

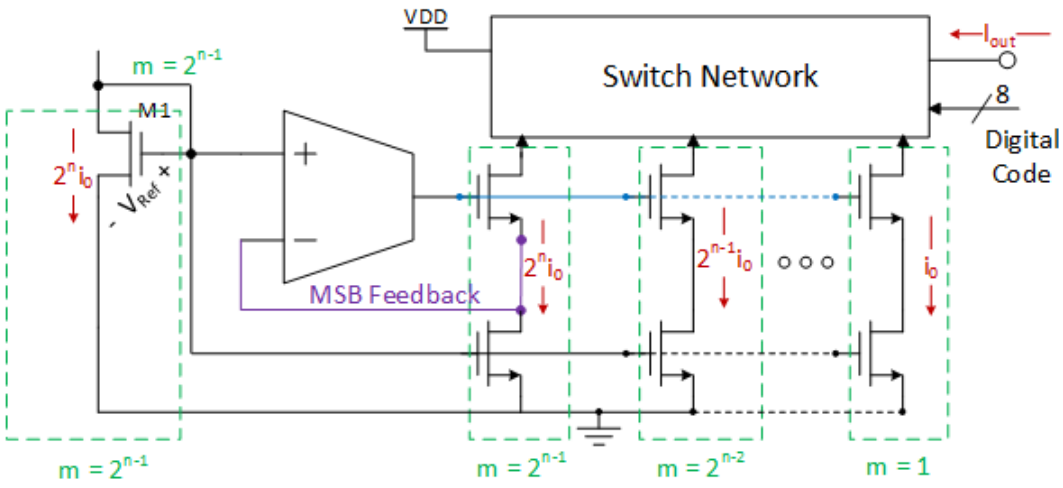

Figure 4.6: OTA placement in DAC topology

A transistor-level schematic of the OTA is provided in Figure 4.7. The OTA was designed for emphasis on power efficiency and the ability to perform on low bias currents. Its design was completed by colleague Tan Yang. The topology of the OTA consisted of a simple differential pair and a current mirror network. The OTA drove an equivalent capacitive load of 110 fF while operating on a low bias current $50nA$ and a $1V$ supply. Dynamic performance of the OTA yields a gain bandwidth product of 2.50 MHz and a phase margin of greater than 85°.

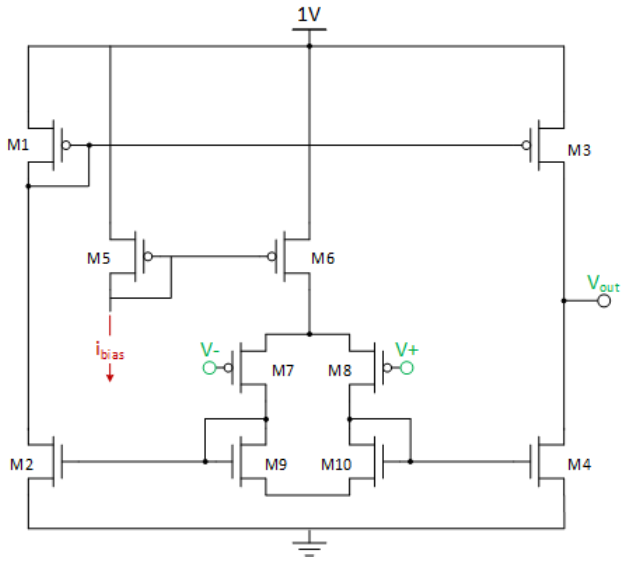

Figure 4.7: Transistor-level schematic of OTA

4.1.3 Switch Design

The switching of sinking currents was accomplished by MOS switches. These switches routed the scaled sinking currents to two separate nodes based upon the applied digital code. A basic switch topology is created from both nMOS and pMOS transistors and is shown in Figure 4.8. An nMOS switch passes the current to the output summing node, and a pMOS switch 'dumps' the current to the 1V rail. The gates of both nMOS and pMOS switches are driven by the same control signal; when it is logic high, the respective current is routed to the output node, and when it is logic low, the current is dumped to the 1V rail. By using the same signal, inverters are not needed to generate complementary signals. However, the dynamics of these simple switches deviate from those of ideal switches. The control signals contain high frequency content that feeds through the gate capacitance and into the sensitive analog signals. Without additional design effort, glitch impulse area will be observed at the output. With an LSB current of $78pA$, the $1V$ digital control signal's feed through contributes

significant glitch energy. This negatively affects the dynamic performance elements of the DAC, such as settling time and overshoots.

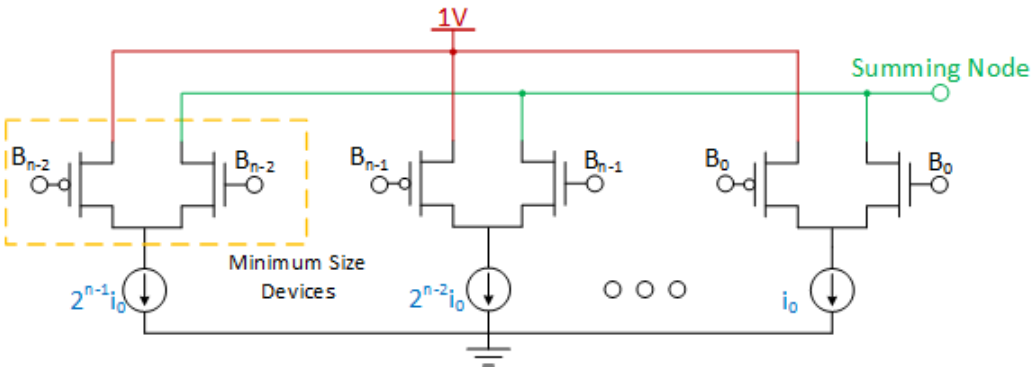

Figure 4.8: nFET and pFET switches sum or dump the current

Layout for the nFET / pFET switch is provided in Figure 4.9.

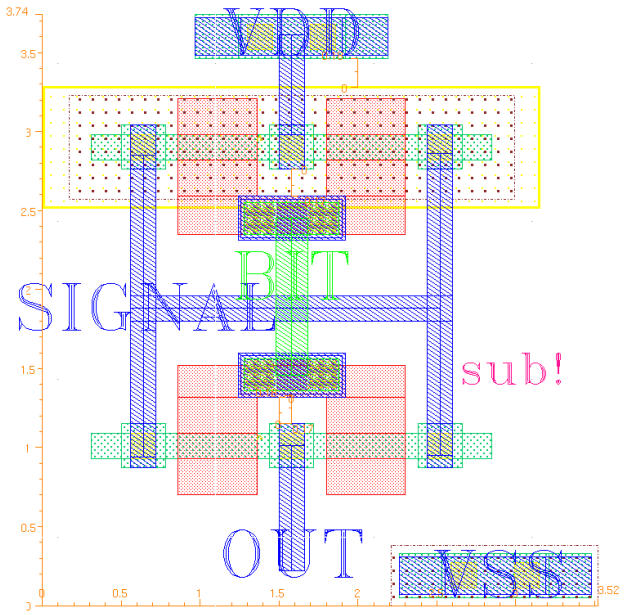

Figure 4.9: Layout of Binary DAC Switch

Maximum sampling rate is dictated by the conversion time from a input step input of code 0x00 to code 0xFF to be within 0.5LSB of the final value. With control

signal feed through, additional settling time is required for the overshooting discharge current to dampen out. Therefore, the geometry of the switch FETS are of minimum feature size, in order to minimize the feed through and parasitic capacitance. The capacitance of the FET switches needs to be minimized to minimize control signal feed through. The optimization is completed by using MOSFETs of minimum feature size geometry. To further decrease settling time, the sinking reference currents are never shut off. Instead, when a bit is switched off one MOS switch routes the sinking current to the supply rail- similar to that of emitter-coupled logic (ECL). Step responses for with and without the additional switch are provided in Figure 4.10.

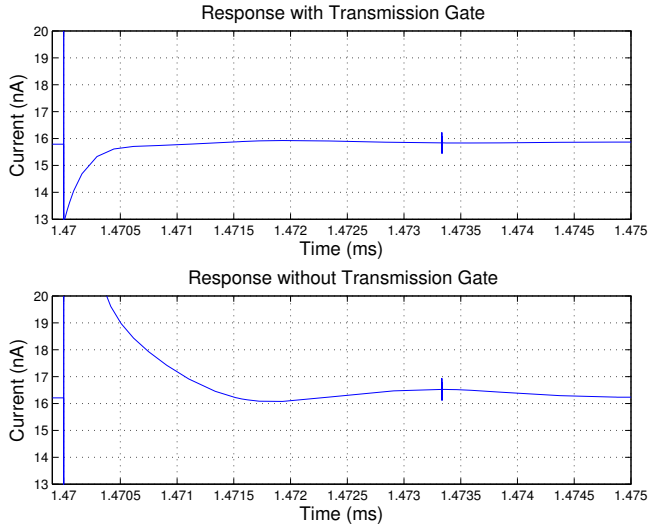

Figure 4.10: Step Response of additional shunting pFET switch

Process variation and mismatch are present in all electronics, including passive elements such as resistors. The matching of devices in a DAC's current mirror is paramount to static performance. Current mirrors can produce a nonuniform scaling of the bit currents that lead to large DNL and INL errors (Pelgrom et al., 1989). The most common mismatch characteristic is from variation in doping. This yields varying threshold voltages in 'matched' MOSFET devices that are higher or lower than the expected value. In many binary-weighted cases the DNL and INL factors

become large, reducing the linearity and eventually the monotonicity of the DAC. The local variation of parameters such as sheet resistance, channel dopant concentration, mobility, and gate oxide thickness have an area dependence shown in Equation 4.1, where: σ_p^2 is the process parameter standard deviation, and W and L is the width and length of the transistor. Qualitatively, local variations decrease as the device size increases since the parameters "average" over a greater distance or area (Drennan and McAndrew, 2003). Mirror nFET devices for the binary weighted topology were increased to 4 times the minimum feature size to reduce the variance of the transistors by half. It would be discovered later that the variance was still too great, and created high static errors.

$$\sigma_p^2 \propto \frac{1}{WL} \tag{4.1}$$

Figure 4.11 presents the complete layout of the binary weighted DAC.

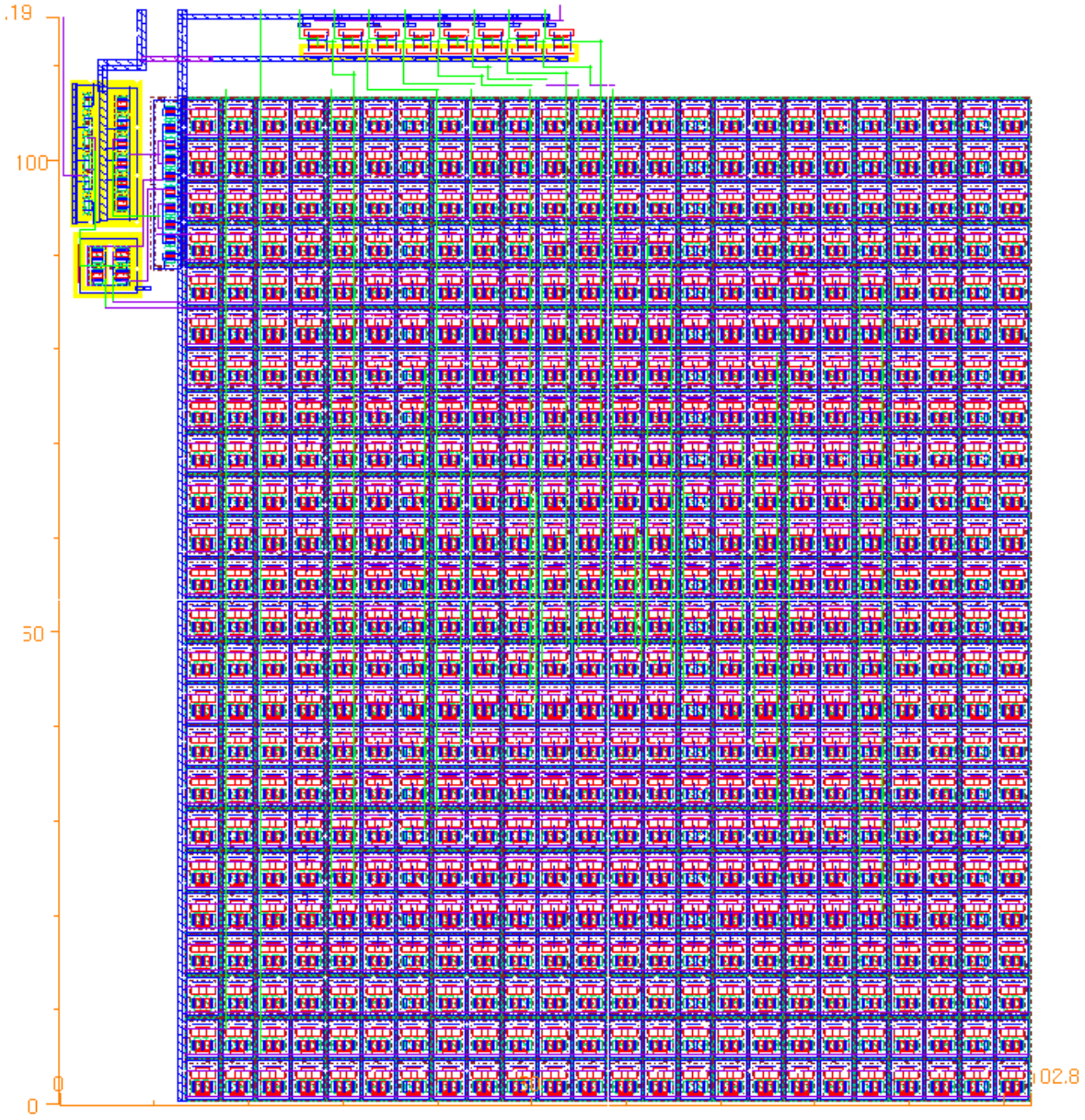

Figure 4.11: Complete Layout of Binary Weighted DAC

4.2 Binary Weighted DAC Simulation Results

4.2.1 Static Performance

The static performance is simulated by applying a digital ramp and performing measurements on the output. A simulated ramp response is used to provided in Figure 4.12.

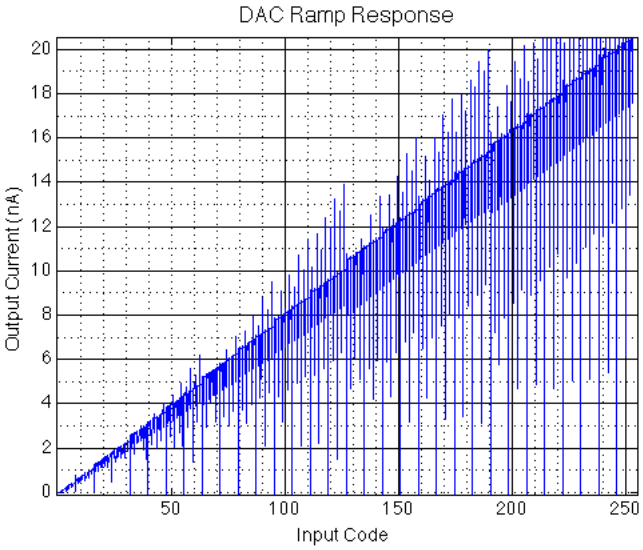

Figure 4.12: Binary Weighted DAC Digital Ramp Response

Offset and Gain Error

The simulated offset error was 17 pA ±1.5 pA with a confidence of 95%. This was found from the results of a 20 iteration Monte Carlo simulation provided in Figure 4.13.

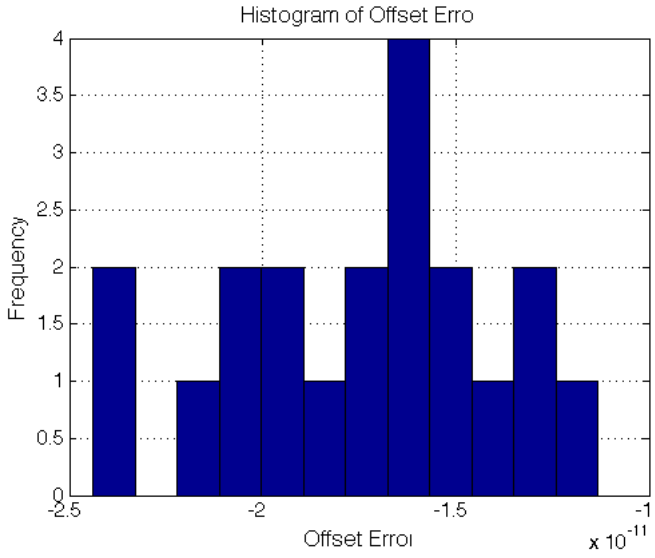

Figure 4.13: DAC Percent Offset Error is 17 pA ±1.5 pA with a confidence of 95%

Gain error percentage was found to be $-0.65\% \pm 3.2\%$ with a confidence of 95%. A 20 iteration Monte Carlo analysis was used to obtain these results. A gain error percent histogram is provided in Figure 4.36.

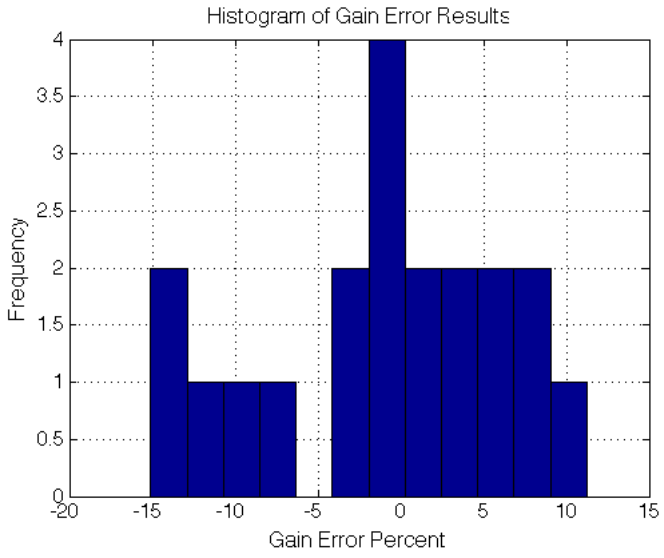

Figure 4.14: DAC Percent Gain Error is $-0.65\% \pm 3.2\%$ with a confidence of 95%

DNL and INL

The DNL and INL simulation results are shown in Figures 4.15 and 4.16 respectively. The histograms were normalized to 1 LSB and obtained from a 20 iteration Monte Carlo simulation. DNL and INL were obtained after offset and gain error artifacts were removed, as discussed in Section 2.4.2. The simulated worst-case DNL is 2.60 LSB ±0.65 with a confidence of 95%. INL is simulated to be 1.5 LSB ±0.65 with a confidence of 95%. These DNL and INL results indicate that the binary weighted DAC is not monotonic and requires modification to reduce mismatch.

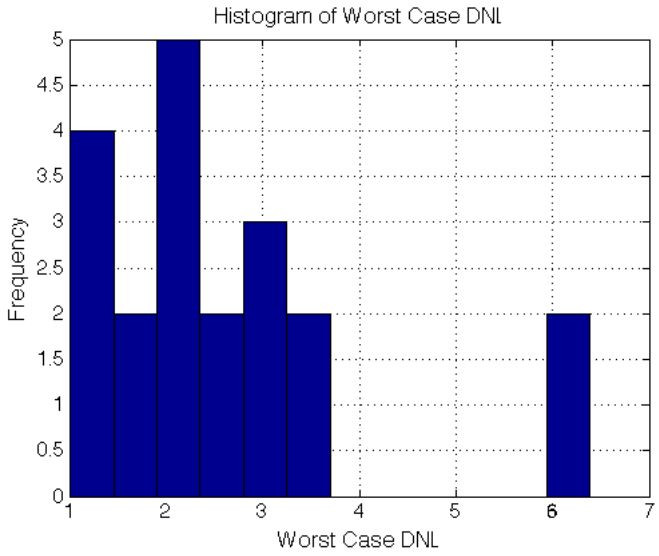

Figure 4.15: DAC Worst Case DNL is 2.60 LSB ±0.65 LSB with a confidence of 95%

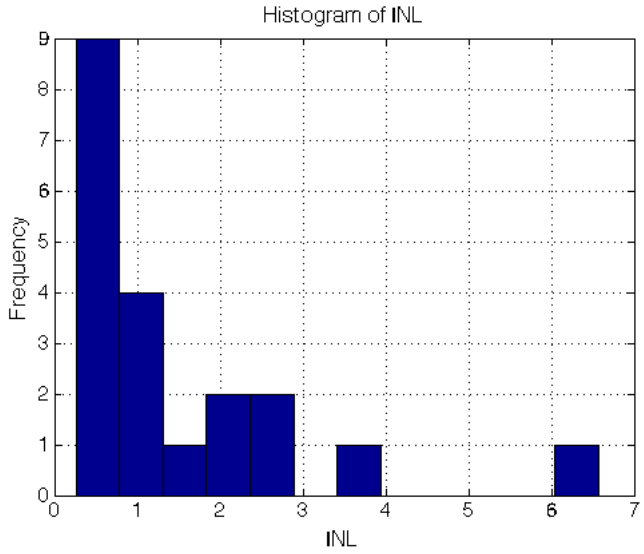

Figure 4.16: DAC INL is 1.5 LSB ±0.65 LSB with a confidence of 95%

4.2.2 Power Consumption

The power consumption results were excellent; simulated average power consumption was $385nW$. The power consumption per each component in the DAC is provided in Table 4.1. The majority of the power consumption was from the OTA.

Table 4.1: Power Consumption by DAC Components

Component	Avg. Power
OTA	$168.7nW$
Current Mirrors	$30nW$
Inverters	$25nW$
Total	$223nW$

The energy-per-step of the DAC was calculated by Equation 4.2, where: P is the total average power of the DAC, f is the DAC frequency, and n is the number of bits. The energy-per-step was found to be $8.71fJ$.

$$E = \frac{P}{f \times 2^n} \tag{4.2}$$

4.2.3 Dynamic Performance

Maximum Sampling Rate

Maximum sampling rate is limited by the switching of 1 LSB from codes $0x00$ to $0xFF$. This step response is provided in Figure 4.17. As shown, the rise-time from 0 to within 0.5 LSB of the final value is about $5.5\mu s$. This leads to a theoretical sampling rate of 181.8kHz.

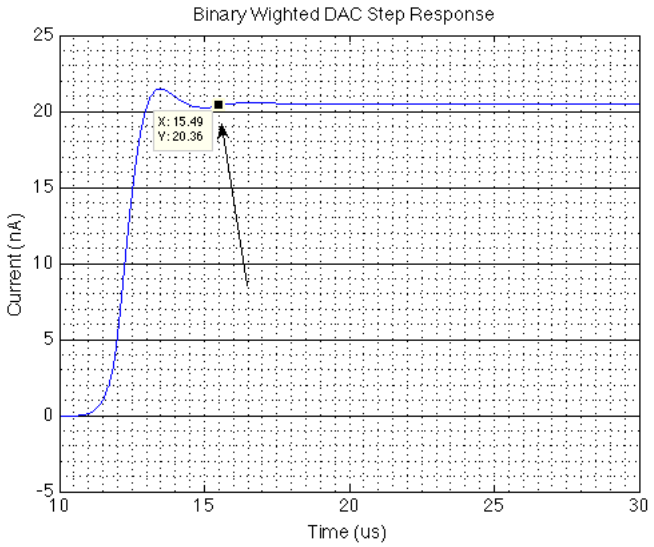

Figure 4.17: Step Response (0x00 to 0xFF)

Noise Analysis

Noise spectral density is presented in Figure 4.18. The output referred noise was $1.1pA$ rms integrated over a bandwidth of 1kHz to 200kHz, and $72.5fA$ rms integrated over a bandwidth of 1Hz to 200KHz.

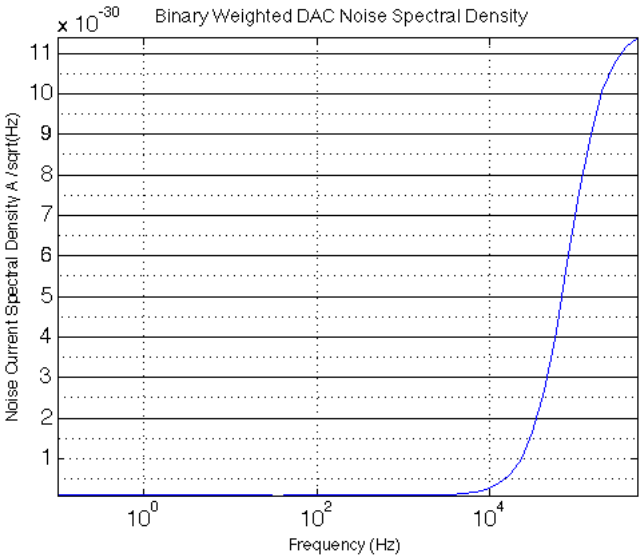

Figure 4.18: Noise Response

Quantization noise was calculated from Equation 4.3 to be $22.51pA$ rms. It can then be safely concluded that the noise floor was limited by only the quantization noise; even if the output-referred noise is integrated over 1Hz to 200kHz. Since additional noise generated from the circuit is significantly less than the quantization noise, it is neglected. The signal to noise and distortion ratio is then limited only by the quantization noise and harmonic distortion.

$$N = \frac{1LSB}{\sqrt{12}} = \frac{78pA}{\sqrt{12}} \tag{4.3}$$

Time-Domain Sine Wave Response

The time-domain response of the DAC to a 7 kHz sinusoidal input is provided in Figure 4.43. From observation, quantization noise is present in the DAC output. Figure 4.44 shows the DAC output that has been reconstructed with a low-pass filter (LPF). The corner frequency of the LPF at the Nyquist frequency (50 kHz).

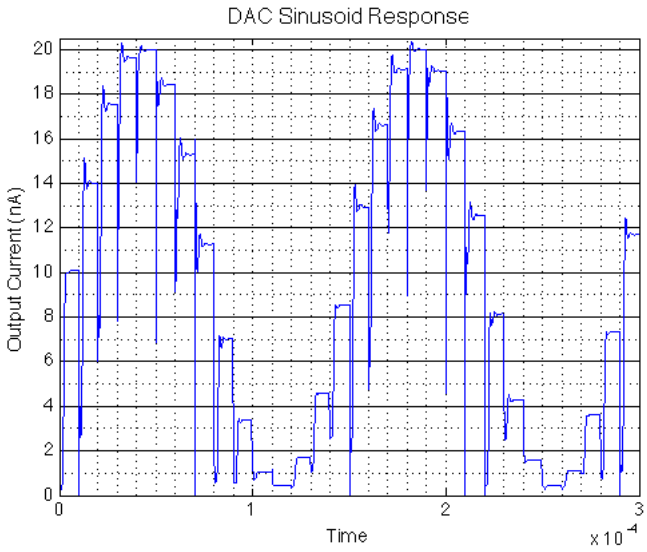

Figure 4.19: 7 kHz Sine Wave Response at 100 kHz Update Rate

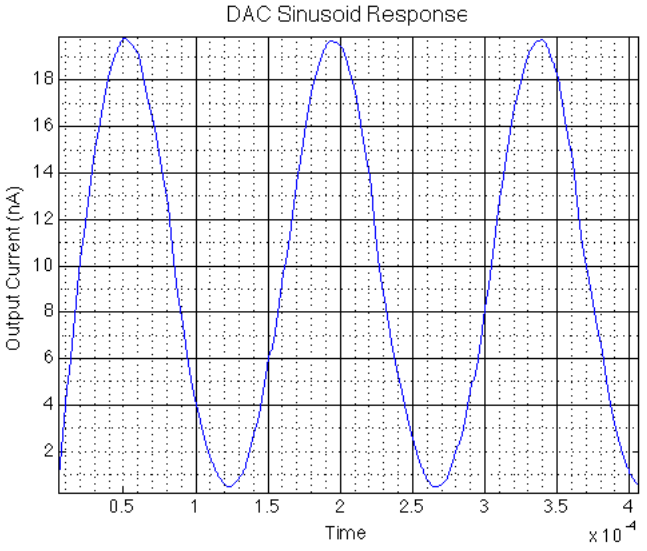

Figure 4.20: 7 kHz Sine Wave Response at 100 kHz Update Rate with a LPF Corner at 50 kHz

THD

Dynamic performance of the thermometer encoded DAC was simulated with a digital sine wave input at a frequency of 7kHz. The sampling rate of the DAC was set

to 100kHz, yielding a non-integer ratio of sampling to input frequency. This kept quantization noise independent of the input signal. Figure 4.21 shows the THD of the DAC output to be -23.5 dB. The THD from a low-pass filter at a corner frequency of 50 kHz is -23.94 dB as shown in Figure 4.22.

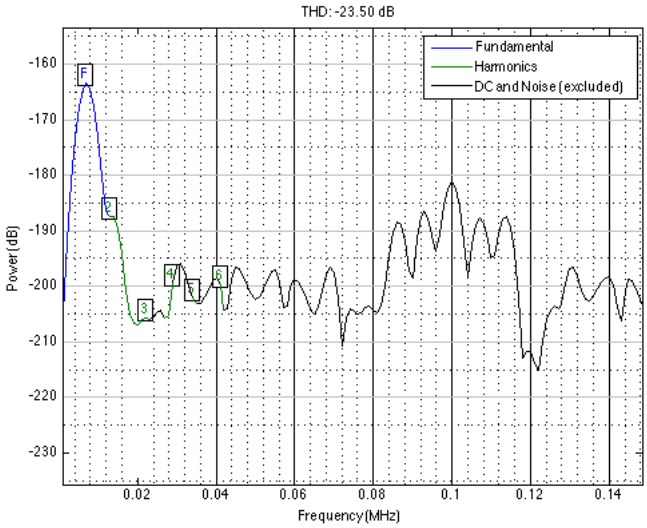

Figure 4.21: DAC THD of -23.5 dB with 7 kHz Fundamental

SFDR

Simulated SFDR for the DAC output is 22.18dB and is shown in Figure 4.23. The 100 kHz spur created from the update control signals is the greatest spur. Figure 4.24 is the SFDR with a LPF filter applied to the DAC output. It is interesting to note that the greatest limiting spur is from the 4th Harmonic.

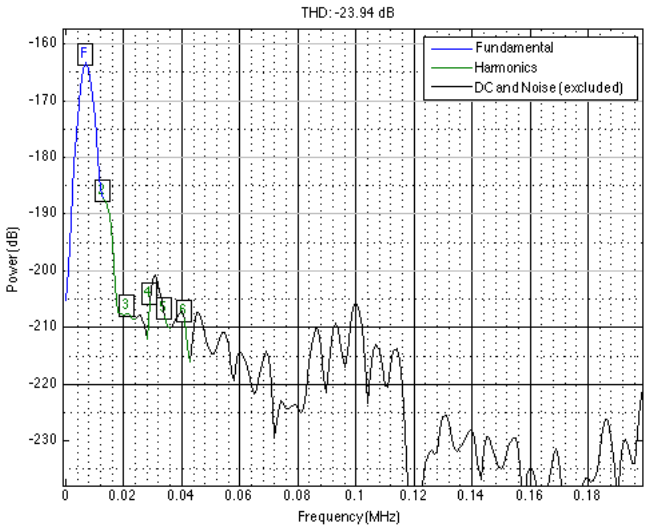

Figure 4.22: DAC THD of -23.94 dB with 7 kHz Fundamental and LPF Corner at 50 kHz

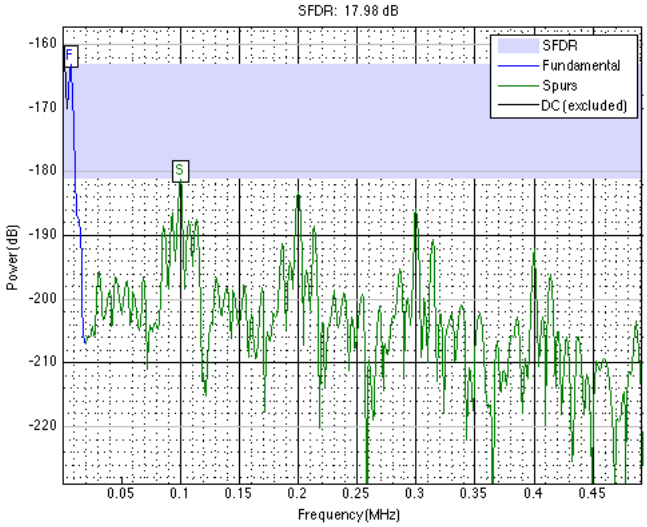

Figure 4.23: DAC SFDR of 17.98 dB with 7 kHz Fundamental

Figure 4.24: DAC SFDR of 37.27 dB with 7 kHz Fundamental LPF Corner at 50 kHz (4th Harmonic is the largest spur)

SINAD and EOB

Simulated SINAD without a reconstructing LPF was 11.43 dB as shown in Figure 4.25. Decreasing the signal bandwidth to 15 kHz with a LPF increases the SINAD to 50.7 dB as shown in Figure 4.26.

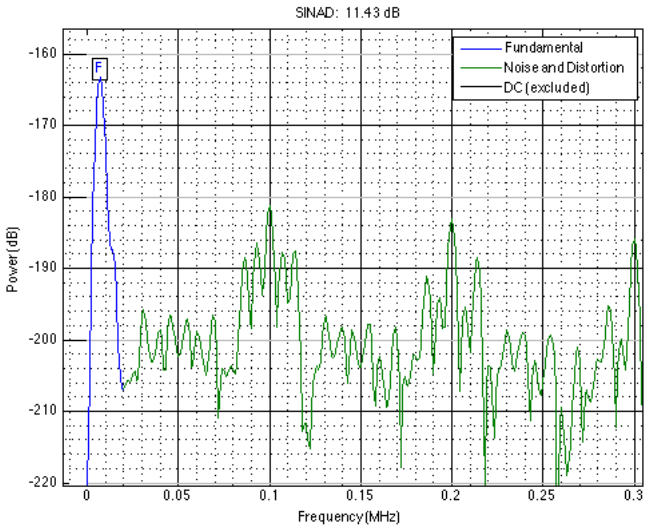

Figure 4.25: Power Spectrum with 7 kHz Fundamental and SINAD of 11.43 dB

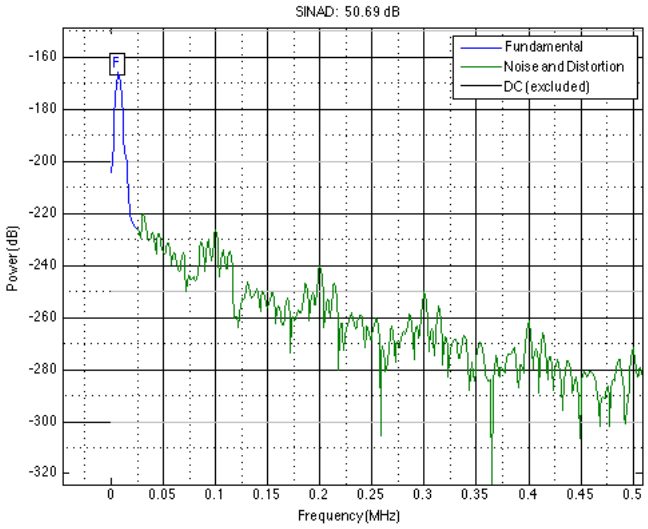

Figure 4.26: Power Spectrum with 7 kHz Fundamental and SINAD of 50.7 dB (LPF Corner at 15 kHz)

Table 4.7 provides the simulated SINAD results and their dependence on bandwidth. The effective number of bits (ENOB) was calculated from SINAD with

Equation 2.15 from Section 2.5. In order to obtain 8 effective bits, bandwidth was reduced by lowering the LPF corner frequency to 15 kHz.

Table 4.2: Low Pass Cutoff Frequency and ENOB

Low Pass Cut-off Frequency (kHz)	-	50	15
SINAD (dB)	11.43	32.85	50.7
ENOB	1.62	5.2	8.0

4.2.4 Summary of Binary Weighted DAC Performance

A summary of the binary weighted DAC's performance and characteristics is provided in Table 4.3. The power efficiency is excellent, however, the DAC fails to meet static performance requirements of INL and DNL because of process variation.

Table 4.3: Performance Overview of DAC

Parameter	Value
Worst Case DNL (LSB)	2.60 ±0.65 (95% Confidence)
INL (LSB)	1.5 ±0.65 (95% Confidence)
Average Power Consumption (nW)	223
Energy per Step (fJ)	8.71
Silicon Area (μm^2)	11822
Theoretical Max Sampling Rate (kHz)	182
Quantization Noise (pA_{rms})	22.5
BW (kHz)	15
THD (dB)	-53.3
SFDR (dB)	54.3
SINAD (dB)	50.7
ENOB	8

4.3 Modified DAC Design

4.3.1 Segmented Current Steering DAC

The previously proposed binary weighted DAC was designed for the specific application of providing ultra low current signal vectors to a system. In terms of area and power efficiency, the simulation results look great. However, high INL and DNL imply that the binary weighted DAC is not monotonic reliable, thus failing static performance requirements. Previously demonstrated in Equation 4.1, increasing transistor area reduces current mismatch variation and theoretically

decreases DNL and INL errors. The dynamic performance, while acceptable for the current application, has room for improvement in future applications. To increase dynamic performance, switches need to become more ideal and their translated glitch energy must be reduced.

To improve both static and dynamic performance, the binary weighted topology was modified to that of a segmented thermometer coding. The segmented thermometer DAC incorporates advantages from both the binary weighted topology and the fully thermometer encoded architecture. The 4 MSBs of the 8 bits were thermometer coded, and the 4 LSBs were left binary weighted. This increased the total number of switches and current branches from 8 to 19. This modification required a digital 4-to-15 thermometer encoder. This digital encoder was synthesized as part the DAC sub-interface (as described in Section 3.1) using VLSI tools. The digital encoder was prototyped with the hardware description language (HDL), Verilog, and then that Verilog code was synthesized into a transistor-level model with synopsis design tools. The physical layout for the encoder was created from the transistor-level model using Cadence's encounter software. The OTA's were designed by colleague Tan Yang.

A top-level schematic of the segmented thermometer encoded DAC is provided in 4.29. Like the previous design, there is a current scaling block along with a switch bank. Unlike the previous design, there is an additional OTA, thermometer encoded, and additional current sources. These differences are fully explained and justified in the following subsections.

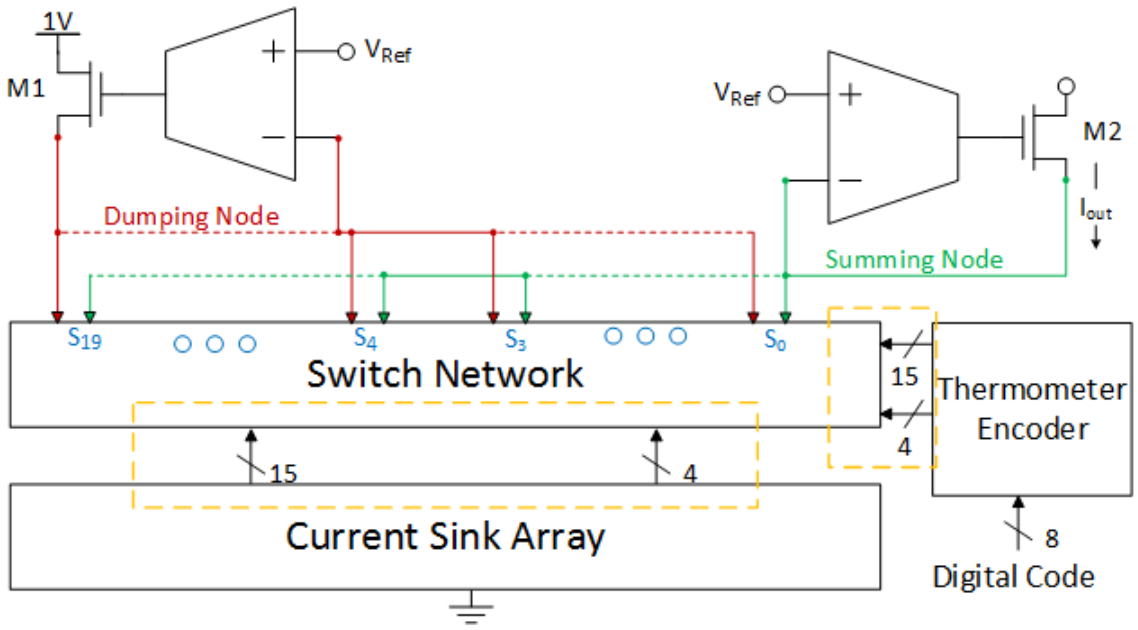

Figure 4.27: Segmented Thermometer Encoded Top Level Schematic

4.3.2 Current Sink Array and Standard Cell

The OTA and the current mirror cascode nFETs were also repositioned such that negative feedback was taken collectively from the 'summing' node, instead of one node pertaining only to the MSB. The cascode nFETs within each of the current mirror standard cells are no longer needed, since they were repositioned with the OTAs; and by removing the cascode nFETs, the current mirror's area is reduced. A modified layout of the standard cell is shown in Figure 4.29.

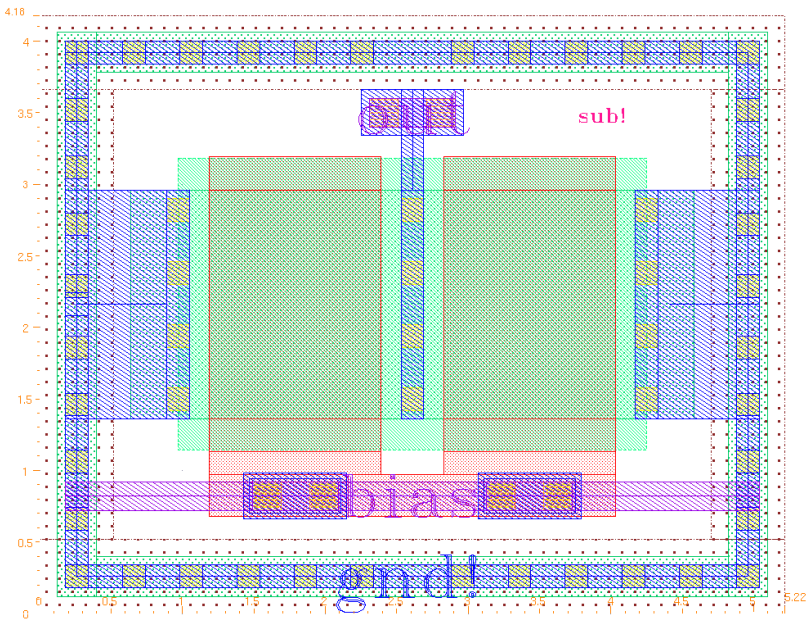

Figure 4.28: Redesigned Current Sink Standard Cell

Figure 4.29 shows a detailed view of the current sinks and how they are segmented. The reference current is provided from off chip and the DAC is designed for the same 80 pA LSB current. The first four LSB current sinks are binary weighted with multiples of the standard cell shown in Figure 4.29. After the 4th LSB, the 5th current sink and onward (or all of the thermometer encoded current sinks) have the same weight as the 4th LSB current (8 x 80 pA). The layout is again completed in a common-centroid topology.

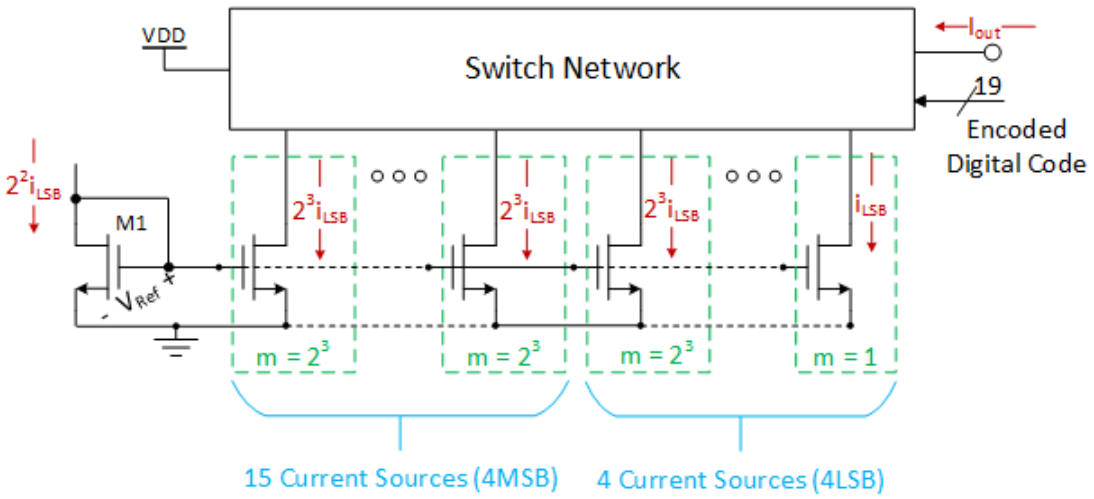

Figure 4.29: Top Level Schematic Showing Segmented Current Sink Circuitry

With the reposition of the original OTA, another OTA was added in a similar configuration with negative feedback taken from 'dumping' node to maintain proper DC bias conditions independent of the digital input code. Both OTAs drive new cascode nFETs. One cascode nFET's source is connected at the summing node, routing current to the output; the other cascode nFET's source is connected at the dumping node, dumping the current at the rail voltage. The reposition and connection of the OTAs along with cascode nFETs, M1 and M2, are shown in Figure 4.30. To further reduce DNL from process variation, the current mirror nFETs' geometry was set to 16 times the minimum feature size. Following Equation 4.1, this yields a 4x decrease in the variance. The increase in the nFET mirror area was almost equally paid for by the reduction in area from removing the cascode nFETs.

4.3.3 Switch Design

The switches were made more ideal with the use of complementary differential switches to reduce the feed-through by increasing off impedance (Luh et al., 2000). To improve the switching performance, at the slight expense of area, all single transistor switches were replaced with differential switches such as those described for the

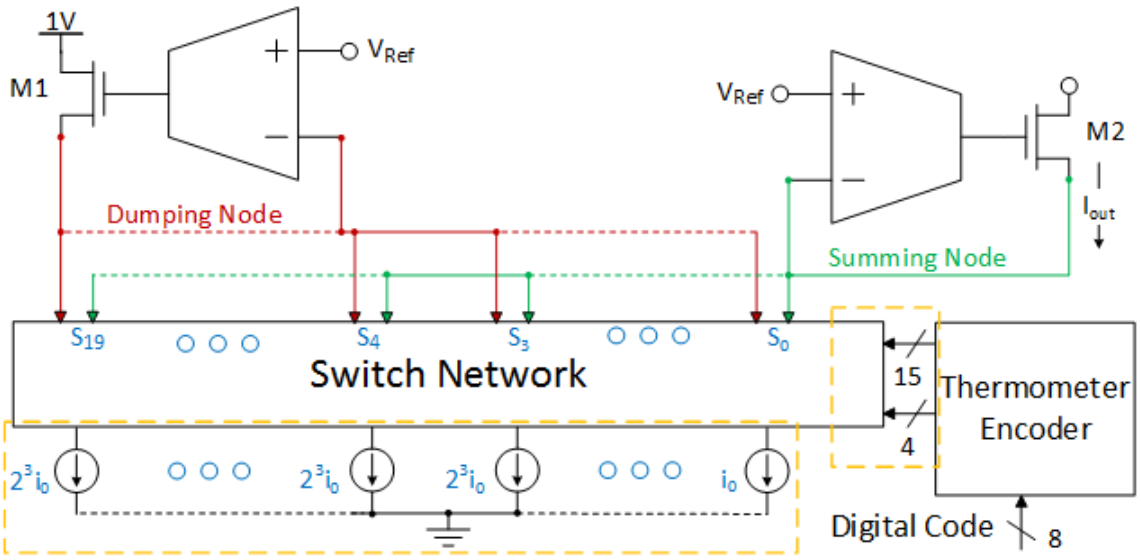

Figure 4.30: OTAs repositioned for increased performance

proposed demultiplex in Section 3.1. A schematic view of the new switch is provided in Figure 4.31. The control signals for the dummy switch are taken directly from the rail voltage and ground. The result of the dummy switch is a minimization of settling time (Przyborowski and Idzik, 2009).

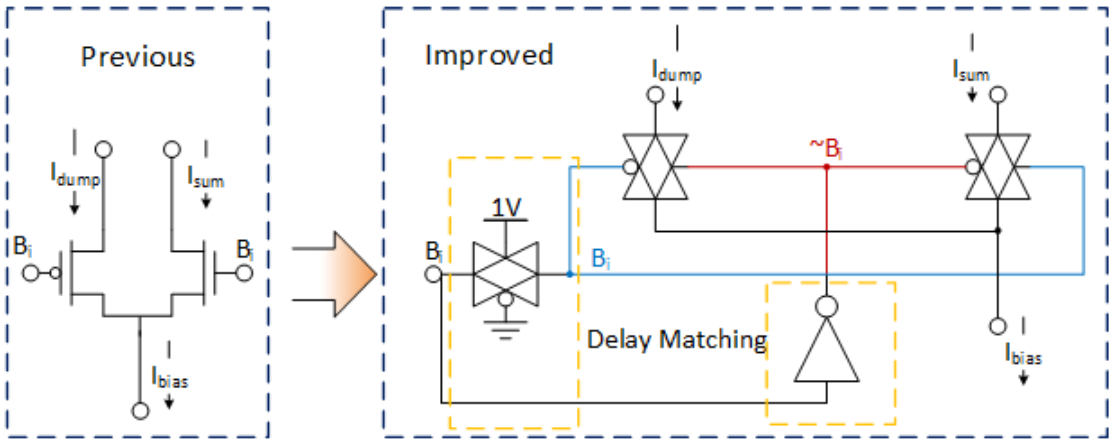

Figure 4.31: Complimentary transmission gates and delay matching

Layout for the new switch is shown in Figure 4.32. Unlike the previous switches, digital wire traces are almost completely separated from any analog wire traces. The

only exception is inside the center of the layout. When analog traces have to cross digital traces, a grounded metal plane is placed between the two traces to 'shield' them. Analog signals enter and leave through the bottom of the layout, and digital control signals enter through the top of the layout. This was done to simplify top-level routing.

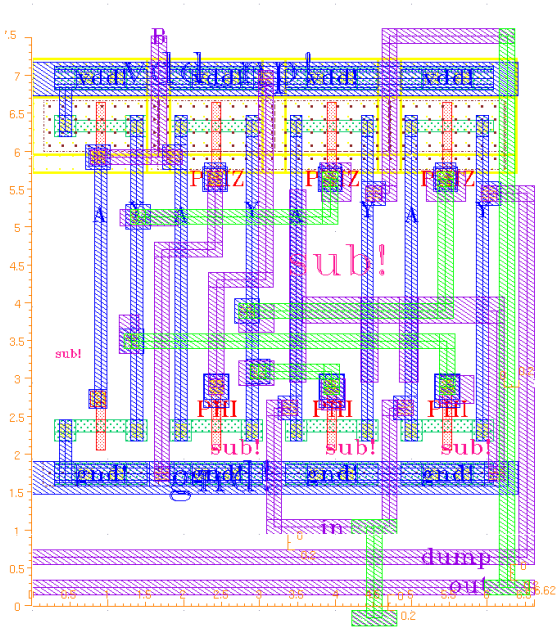

Figure 4.32: Layout of redesigned complimentary transmission gates

4.3.4 Complete Segmented Current Steering DAC Layout

Figure 4.33 presents the complete layout of the thermometer encoded DAC. On the far right, the 4-to-15 thermometer encoder can be observed. Special attention was paid to the isolation of the digital bus from any analog routing. The digital bus travels up the right side of the current sink array and each signal enters the top of their corresponding switch. The analog signals provided by the current sink array enter the switches from the botton away from the digital signals. The switchs' analog output signals exit through the bottom of the switch to either the summing output

node or the 'dumping' node. These nodes run up the left side of the layout toward their designated OTAs at the very top.

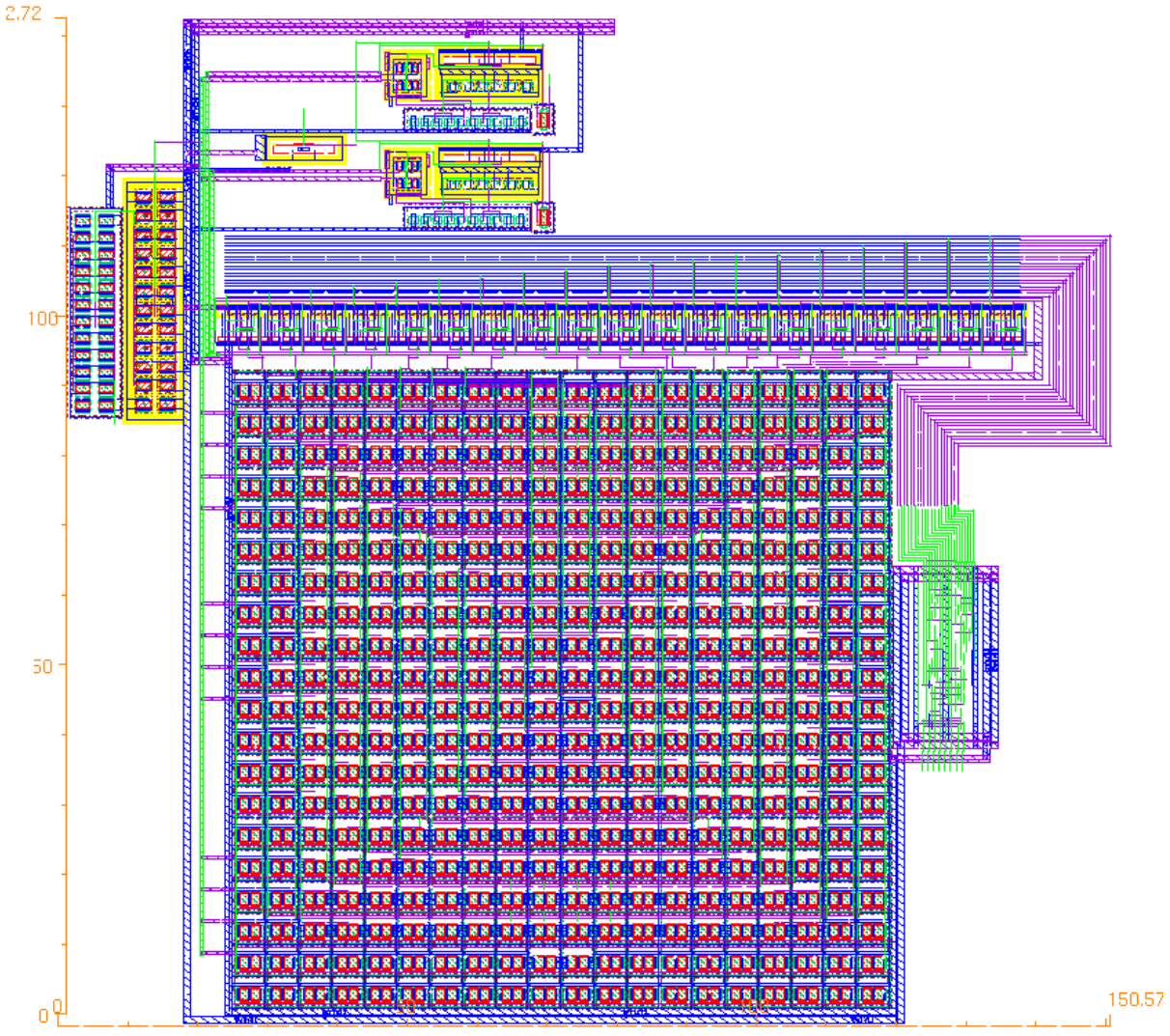

Figure 4.33: Complete Layout of Thermometer Encoded DAC

4.4 Modified DAC Simulation Results

4.4.1 Static Performance

The static performance is simulated by applying a digital ramp and performing measurements on the output. A simulated ramp response is used to provided in Figure 4.34.

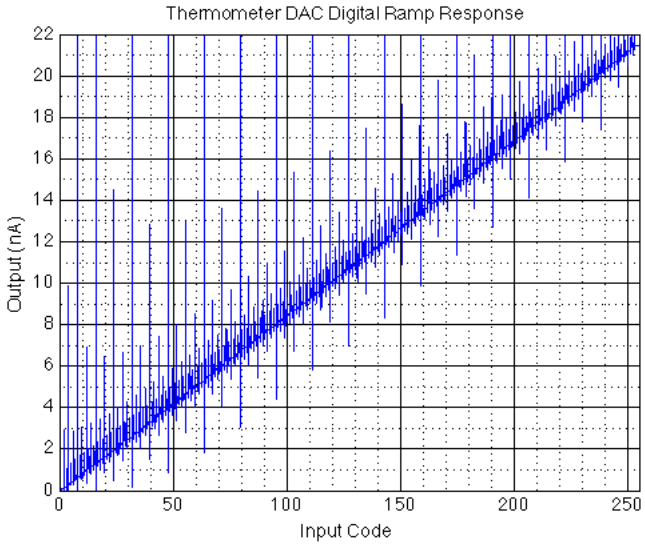

Figure 4.34: Montecarlo Thermometer DAC Digital Ramp Response

Offset and Gain Error

The simulated offset error was nearly deterministic at 4.2 pA ±0.5 fA with a confidence of 95%. This was found from the results of a 100 iteration Monte Carlo simulation provided in 4.35.

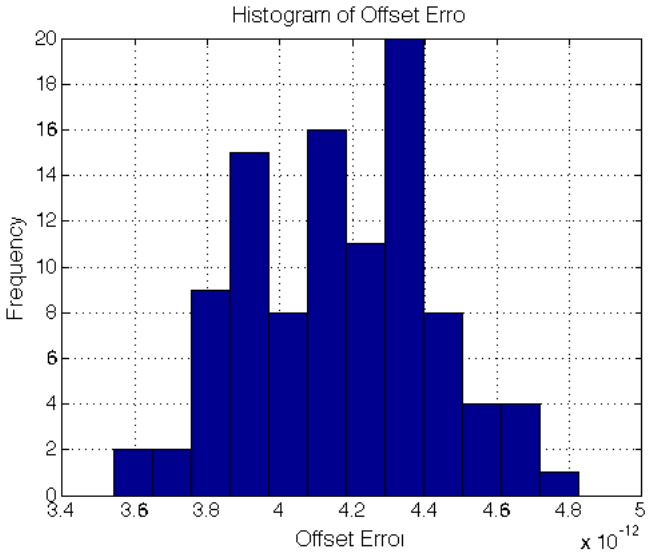

Figure 4.35: DAC Percent Offset Error is 4.2 pA ±0.5 fA with a confidence of 95%

Gain error percentage was found to be $-2.6\% \pm 1.4\%$ with a confidence of 95%. Monte Carlo analysis was used to create a gain error percent histogram in Figure 4.36.

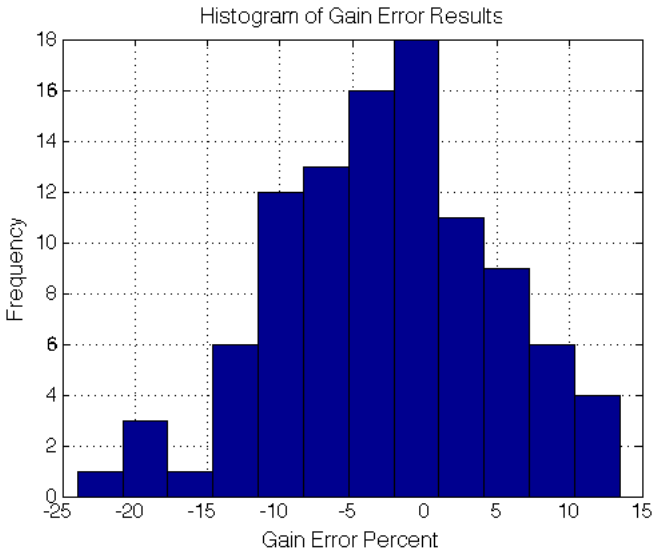

Figure 4.36: DAC Percent Gain Error is $-2.6\% \pm 1.4\%$ with a confidence of 95%

INL and DNL

The plots have been normalized to 1 LSB over the complete range of digital input codes. DNL and INL were obtained after offset and gain error artifacts were removed from the transfer function as discussed in Section 2.4.2. Monte Carlo analysis with 100 iterations yields the worst case DNL and INL histograms results in Figures 4.37 and 4.38 respectively. The worst case DNL was found to be 0.41 LSB ±0.04 LSB with a confidence of 95%. The INL was found to be 0.66 LSB ±0.06 LSB with a confidence of 95%.

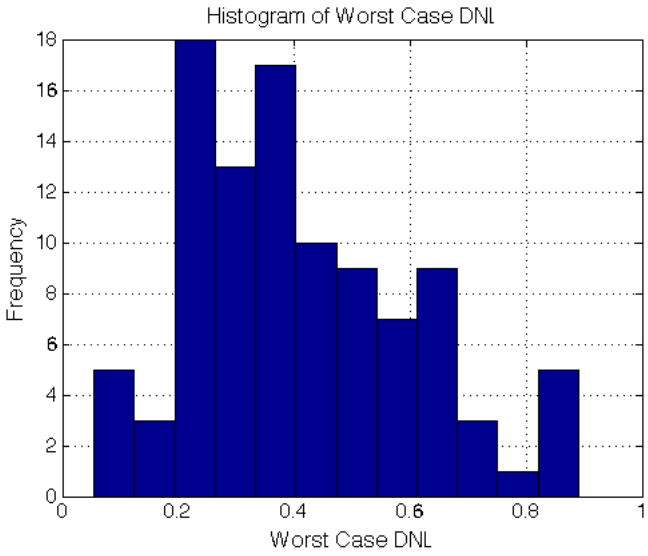

Figure 4.37: DAC Worst Case DNL is 0.41 LSB ±0.04 LSB with a confidence of 95%

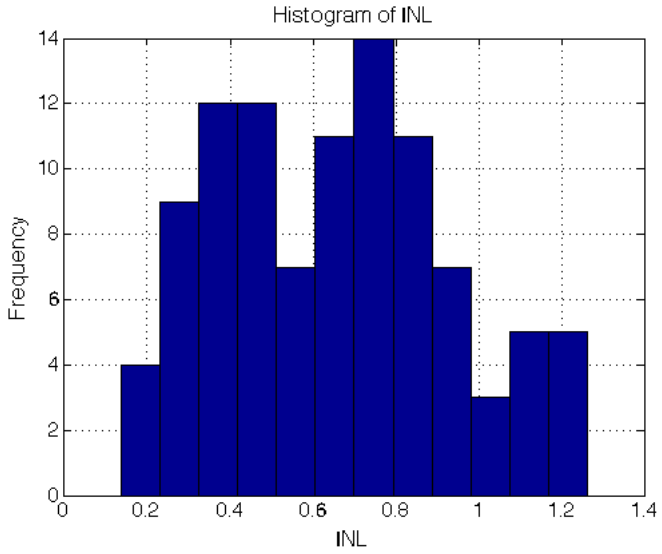

Figure 4.38: DAC INL is 0.66 LSB ±0.06 LSB with a confidence of 95%

These DNL and INL results indicate that the thermometer DAC is indeed monotonic. Figure 4.39 shows a single Monte Carlo iteration with a worst-case simulated DNL of 0.54 LSB and INL bounded within ±1 LSB.

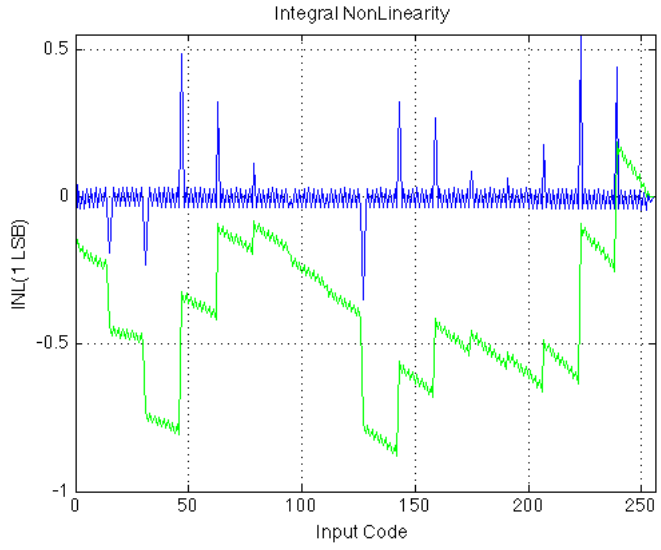

Figure 4.39: Single Monte Carlo Iteration INL and DNL

4.4.2 Power Consumption

The simulated average power consumption from the thermometer encoded DAC was 231.9 nW with a 231.9 nA current draw from a $1V$ rail voltage.

Table 4.4: Power Consumption by Thermometer DAC Components

Component	Avg. Power
OTA	89 nW
Current Mirrors	32 nW
Digital Circuitry	34 nW
Total (x2 OTA)	232 nW

The energy-per-step of the DAC was calculated by the previous Equation 4.2. The energy-per-step was found to be 9.05 fJ.

4.4.3 Dynamic Performance

Maximum Sampling Rate

Maximum sampling rate is limited by the full-range step from codes $0x00$ to $0xFF$ or $0xFF$ to $0x00$; Whichever is the greatest settling time. In this case, the max sampling rate is proportional to the power consumption of the OTA. Figure 4.40 shows a step response with an OTA bias current of 50 nA (168.7 nW per OTA). The worst 0.5 LSB convergence time of the two step responses is 2.49 μs leading to a maximum sampling rate of 403 kHz. The target sampling rate is 100 kHz so we can reduce the OTA power consumption. A reduced power step response is provided in Figure 4.41 with an OTA bias current of 20 nA (88.7 nW per OTA). As shown, the convergence time from 0 to within 0.5 LSB of the final value is about 5.96 μs. This leads to a max sampling rate of 167 kHz with a 67 kHz margin for error.

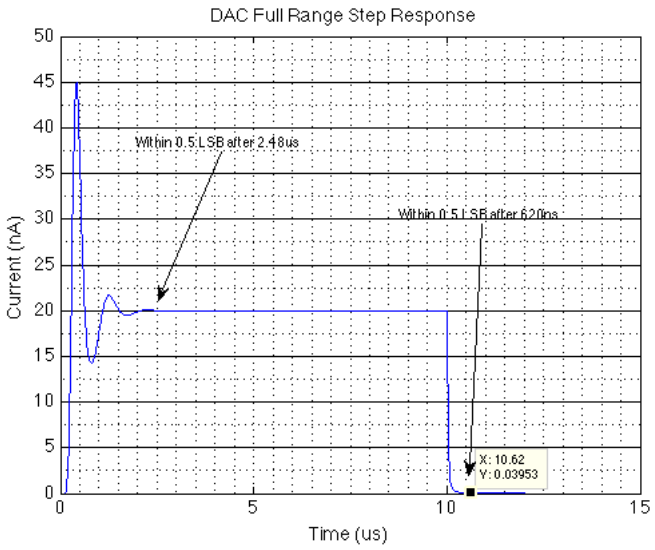

Figure 4.40: Positive and Negative Step Response (50nA OTA Bias Current)

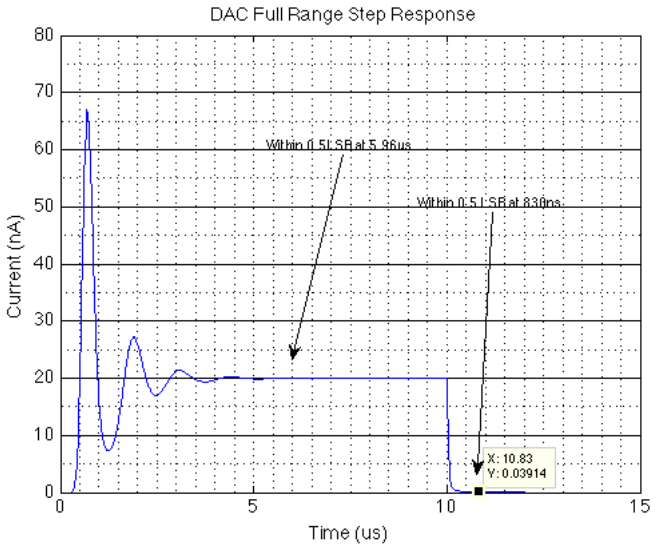

Figure 4.41: Positive and Negative Step Response (20nA OTA Bias Current)

Noise Analysis

Thermometer DAC noise simulation results conclude that the DAC is not limited by other forms of noise (thermal or flicker). Noise spectral density is presented in Figure

4.42. The output-referred noise was $72.4fA$ rms integrated over a bandwidth of 1kHz to 200kHz, and $72.6fA$ rms integrated over a bandwidth of 1Hz to 200KHz.

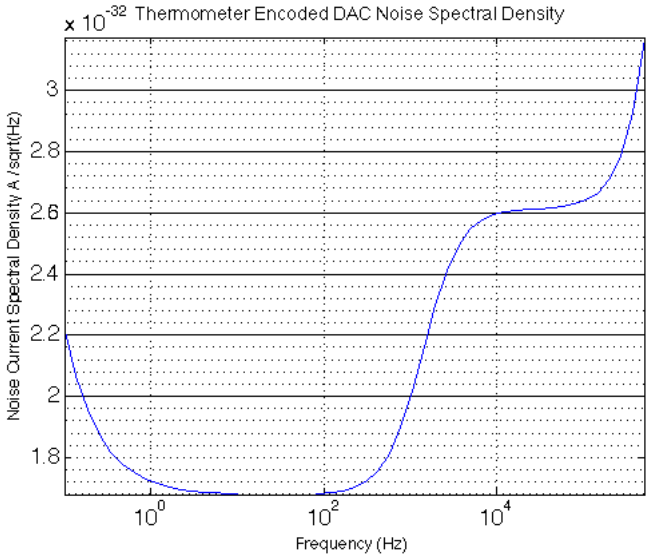

Figure 4.42: Output Referred Spectral Noise Density

It was then safely concluded that the noise floor was limited by only the quantization noise. Quantization noise remains the same from the previous Equation (4.3) to be $22.51pA$ rms. The noise floor was safely concluded that the noise floor was limited by only the quantization noise even if the output-referred noise is integrated over 1Hz to 200kHz. The signal to noise and distortion ratio is then limited only by the quantization noise and harmonic distortion.

Time-Domain Sine Wave Response

The time-domain response of the DAC to a 7 kHz sinusoidal input is provided in Figure 4.43. From observation, quantization noise is present in the DAC output. Figure 4.44 shows the DAC output that has been low-pass filtered with a corner at the Nyquist frequency (50 kHz).

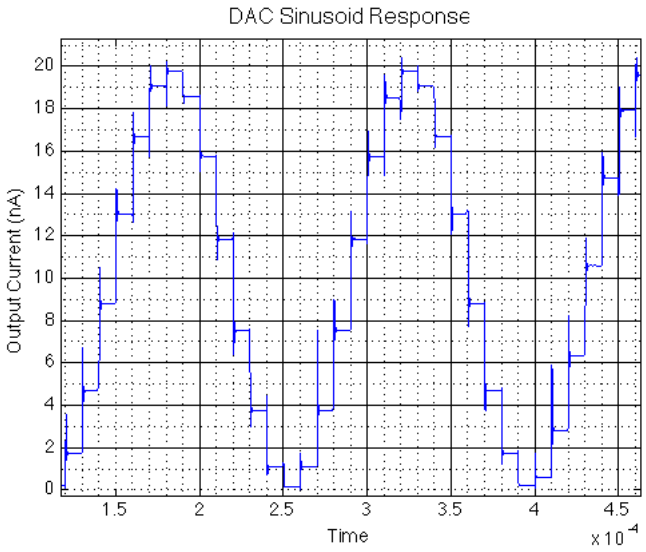

Figure 4.43: 7 kHz Sine Wave Response at 100 kHz Update Rate

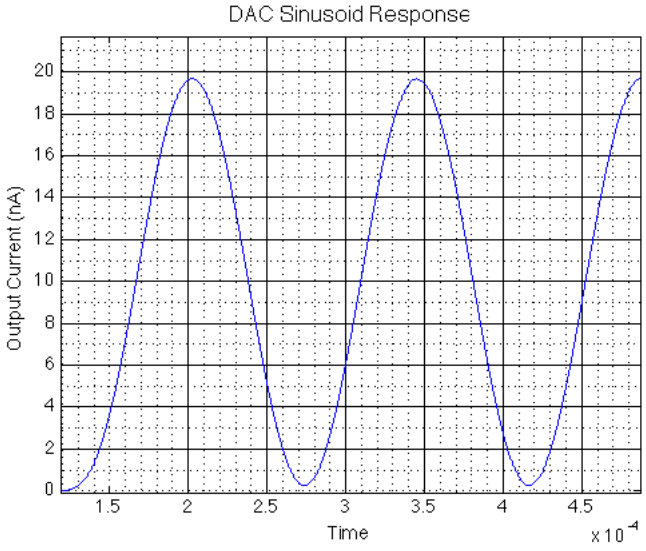

Figure 4.44: 7 kHz Sine Wave Response at 100 kHz Update Rate with a LPF Corner at 50 kHz

THD

Dynamic performance of the thermometer encoded DAC was simulated with a digital sine wave input at a frequency of 7kHz. The sampling rate of the DAC was set

to 100kHz, yielding a non-integer ratio of sampling to input frequency. This kept quantization noise independent of the input signal.

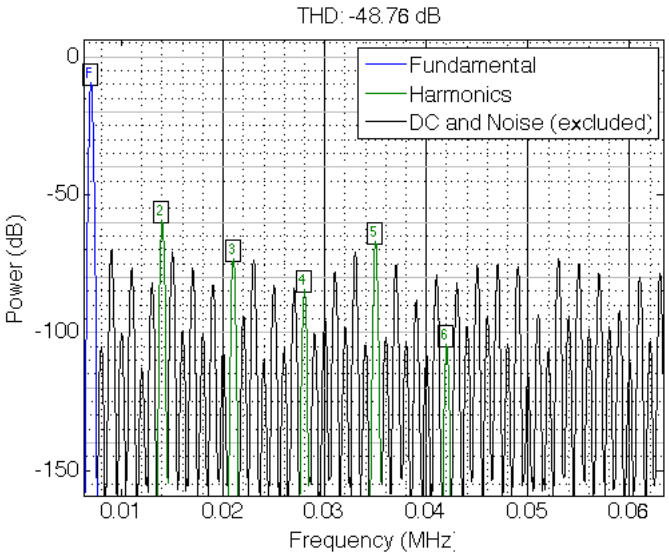

Figure 4.45: DAC THD of -48.76 dB with 7 kHz Fundamental

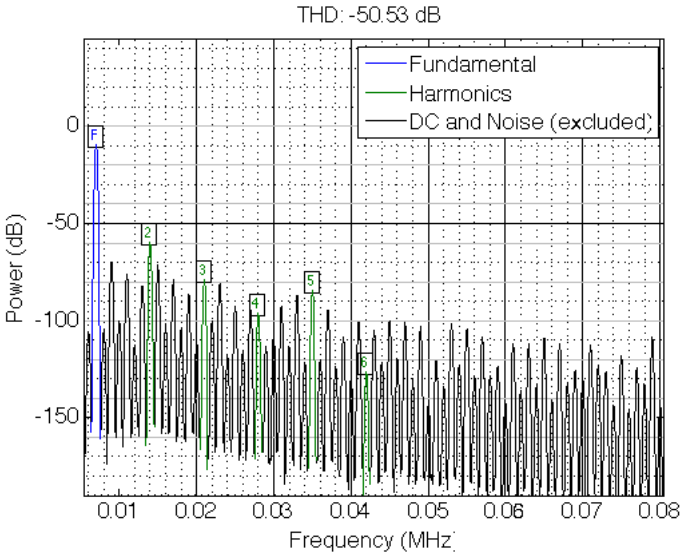

Figure 4.46: DAC THD of -50.53 dB with 7 kHz Fundamental and LPF Corner at 50 kHz

SFDR

Simulated SFDR for the DAC output is 22.18dB and is shown in Figure 4.47. The 100 kHz spur created from the update control signals is the greatest spur. Figure 4.48 is the SFDR with a LPF filter applied to the DAC output. It is important to note that the greatest limiting spur is from the 2nd Harmonic.

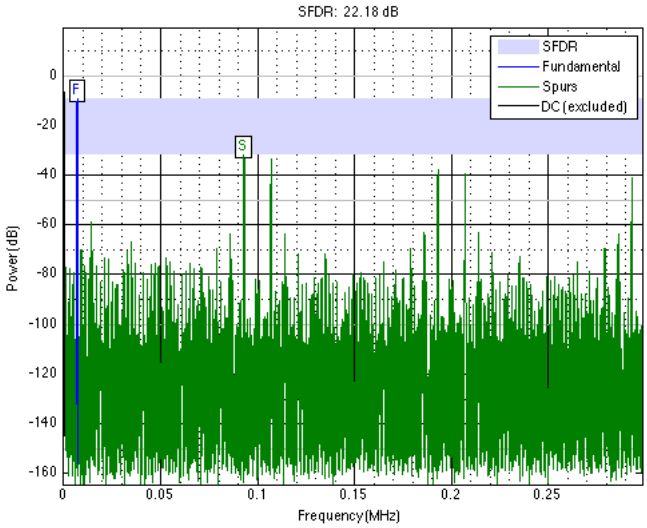

Figure 4.47: DAC SFDR of 22.18 dB with 7 kHz Fundamental

SINAD and EOB

Simulated SINAD without a reconstructing LPF was 17.5 dB as shown in Figure 4.49. Decreasing the signal bandwidth to 25 kHz with a LPF increases the SINAD to 50 dB as shown in Figure 4.50.

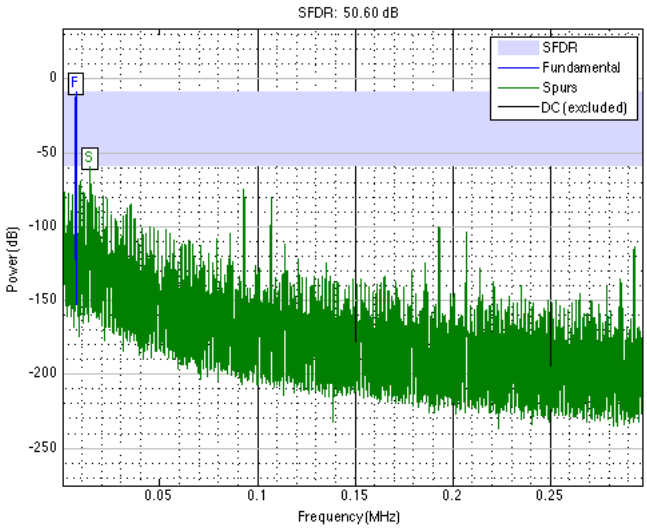

Figure 4.48: DAC SFDR of 50.6 dB with 7 kHz Fundamental LPF Corner at 50 kHz (2nd Harmonic is the largest spur)

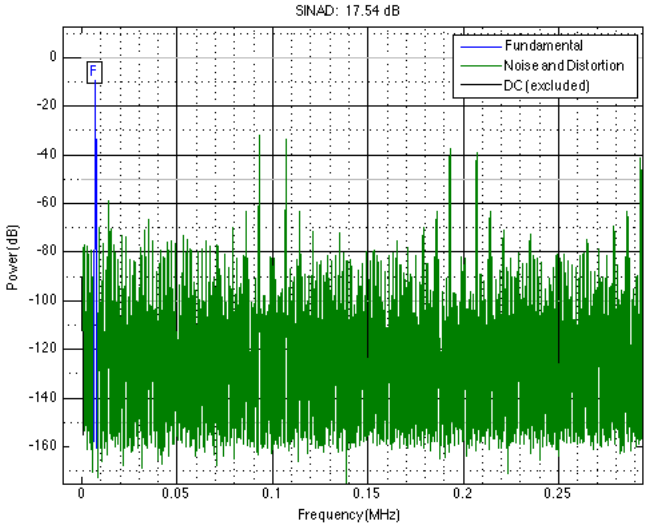

Figure 4.49: Power Spectrum with 7 kHz Fundamental and SINAD of 17.54 dB

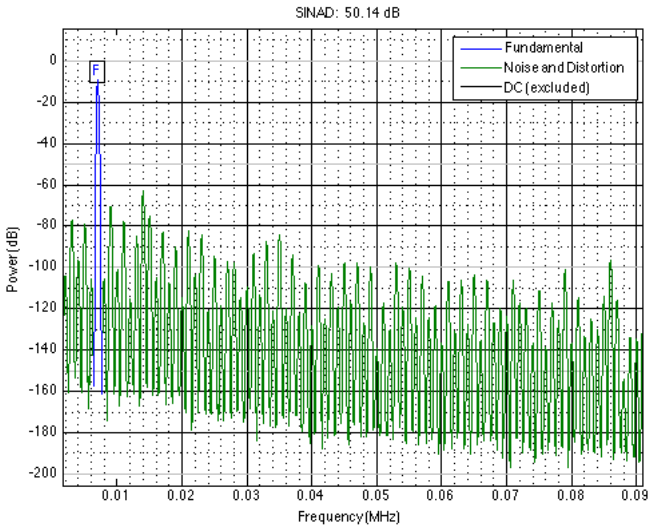

Figure 4.50: Power Spectrum with 7 kHz Fundamental and SINAD of 50.14 dB (LPF Corner at 25 kHz)

Table 4.7 provides the simulated SINAD results and their dependence on bandwidth. The effective number of bits (ENOB) was calculated from SINAD with Equation 2.15 from Section 2.5. In order to obtain 8 effective bits, bandwidth was reduced by lowering the LPF corner frequency to 25 kHz.

Table 4.5: Low Pass Cutoff Frequency and ENOB

Low Pass Cut-off Frequency (kHz)	-	50	25
SINAD (dB)	17.54	42.33	50.14
ENOB	2.62	6.73	8.04

4.4.4 Summary of Segmented DAC Performance

A summary of the binary weighted DAC's performance and characteristics is provided in Table 4.6. The power efficiency is excellent. Static performance meets requirements, as does dynamic performance.

Table 4.6: Performance Overview of DAC

Parameter	Value
Worst Case DNL (LSB)	0.41 ±0.04 (95% Confidence)
INL (LSB)	0.66 ±0.06 (95% Confidence)
Average Power Consumption (nW)	231.9
Energy per Step (fJ)	9.05
Silicon Area (μm^2)	18000
Theoretical Max Sampling Rate (kHz)	167.8
Quantization Noise (pA_{rms})	22.5
BW (kHz)	25
THD (dB)	-53.3
SFDR (dB)	53.3
SINAD (dB)	50.14
ENOB	8

4.5 Discussion

4.5.1 Comparison of Designs

A direct comparison of the two designs is presented in Table 4.7. Granted the bare-bones binary weighted DAC has a smaller floor plan and lower power consumption than those of the segmented DAC, the binary weighted DAC has failing static performance. In some Monte Carlo iterations the binary weighted DAC is not monotonic. Output referred noise integrated over 1Hz to 200kHz remained about the same. The static performance of the segmented DAC meets requirements. Aside from

a slight increase in power consumption and area set-back, the dynamic performance characteristics of the thermometer encoded DAC exceed those of the binary weighted DAC with a larger bandwidth.

Table 4.7: Comparison of DAC Designs

Characteristic	Binary Weighted	Segmented
Worst Case DNL (LSB with 95% Confidence)	2.60 ±0.65	0.41 ±0.04
INL (LSB with 95% Confidence)	1.5 ±0.65	0.66 ±0.06
Average Power Consumption (nW)	223	232
Energy per Step (fJ)	8.7	9.1
Silicon Area (μm^2)	11822	18000
Theoretical Max Sampling Rate (kHz)	182	168
Quantization Noise (pA_{rms})	22.5	22.5
BW (kHz)	15	25
THD (dB)	-54.3	-53.3
SFDR (dB)	54.3	53.3
SINAD (dB)	50.7	50.1
ENOB	8	8

4.5.2 Analog System Interface Application

For the analog system interface application, a LPF is not need because the DAC's raw quantized current signal is used. The DAC's worst case INL is 0.72 LSB requiring the two cascaded sample-and-holds to contribute a combined error of no more than a 0.28 LSBs. A single DAC's convergence time is 10 μs. Since the DAC's output is

demultiplexed to 16 channels, the total convergence time is 160 μs. With the proposed interface, a new analog vector can be created in a time span of $T = 160\ \mu\text{s} + \Delta t_1$, where Δt_1 is the convergence time of the first sample-and-hold. The total delay for a new analog vector entering the analog system is then calculated to be $T = 160\ \mu\text{s} + \Delta t_1 + \Delta t_2$, where Δt_2 is the delay of the second 'synchronizing' sample-and-hold.

Chapter 5

Conclusion

This thesis presents a on-chip system interface that conveniently enables analog vector signal generation from a microprocessor to an analog system. Multidimensional convolution is but one of many multidimensional analog operations being investigated in academia and industry that require such signals. The proposed system utilizes a microprocessor to write data to these systems and read the system outputs. The primary concern of such an interface is scalability. Scalability is enhanced through the minimization of chip area and power consumption. First, area efficiency is addressed by minimizing the number of larger blocks, like the DAC, through the multiplexing of signals and use of smaller sample-and-holds. Second, the minimization of power is then made on a block-to-block basis by limiting reference currents and such.

A variety of components are necessary for the realization of the interface. Digital logic blocks such as controllers were easily synthesized from VLSI tools. Analog blocks such as the sample-and-hold and data converters must be laid out manually. A current steering DAC topology is selected because of its natural speed and current output properties. Two different versions of the DAC were designed to demonstrate their performance. The first design was a simple binary weighted DAC whose static performance failed to meet accuracy requirements. The second design was a thermometer encoded design that offered much needed improvement in

static performance, and a significant dynamic performance boost. This performance increase was made at the expense of a slight increase in power consumption- primarily due to the addition of a second OTA. Statistical simulation results provide evidence that the static performance of the current-steering DAC is indeed reliable.

Bibliography

Allen, P. E., Holberg, D. R., Allen, P. E., and Allen, P. (1987). *CMOS analog circuit design*. Holt, Rinehart and Winston New York. 17, 18, 19

atx7006.com (2014). Parameter calculations d/a converter. http://www.atx7006.com/articles/static_analysis/dac. Accessed: 2014-06-30. xiii, 12, 13, 14, 15

Drennan, P. G. and McAndrew, C. C. (2003). Understanding mosfet mismatch for analog design. *Solid-State Circuits, IEEE Journal of*, 38(3):450–456. 54

Ismail, M. and Fiez, T. (1994). *Analog VLSI: signal and information processing*, volume 166. McGraw-Hill New York. 9

Johns, D. A. and Martin, K. (2008). *Analog integrated circuit design*. John Wiley & Sons. 8

Kester, W. A. (2005). *Data conversion handbook*. Newnes. 15

Lee, D.-H., Kuo, T.-H., and Wen, K.-L. (2009). Low-cost 14-bit current-steering dac with a randomized thermometer-coding method. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 56(2):137–141. 20

Lin, C.-H. and Bult, K. (1998). A 10-b, 500-msample/s cmos dac in 0.6 mm 2. *Solid-State Circuits, IEEE Journal of*, 33(12):1948–1958. 21

Luh, L., Choma Jr, J., and Draper, J. (2000). A high-speed fully differential current switch. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 47(4):358–363. 72

Manganaro, G. (2012). *Advanced data converters*. Cambridge University Press. 12, 13, 14, 20

Pelgrom, M. J., Duinmaijer, A. C., and Welbers, A. P. (1989). Matching properties of mos transistors. *Solid-State Circuits, IEEE Journal of*, 24(5):1433–1439. 53

Przyborowski, D. and Idzik, M. (2009). Development of a general purpose low-power small-area 10 bit current steering cmos dac. In *Mixed Design of Integrated Circuits & Systems, 2009. MIXDES'09. MIXDES-16th International Conference*, pages 256–261. IEEE. 73

Rabaey, J. M., Chandrakasan, A. P., and Nikolic, B. (2002). *Digital integrated circuits*, volume 2. Prentice hall Englewood Cliffs. 5, 6

Razavi, B. (2002). *Design of analog CMOS integrated circuits*. Tata McGraw-Hill Education. 6, 9

Sarpeshkar, R. (1998). Analog versus digital: extrapolating from electronics to neurobiology. *Neural computation*, 10(7):1601–1638. 2

Wegmann, G., Vittoz, E. A., and Rahali, F. (1987). Charge injection in analog mos switches. *Solid-State Circuits, IEEE Journal of*, 22(6):1091–1097. 29

Wikner, J. (2000). Studies on cmos digital-to-analog converters. 6, 8

Vita

Nicholas Conley Poore was born in Louisville, KY, on 19 June 1990, the son of Joesph Conley Poore and Cathy Poore. After graduating from Powell High School, he went on to begin his studies into Electrical Engineering at Pellissippi State Community College. After two years at Pellissippi, Nicholas began his academic career at the University of Tennessee, Knoxville, where he continued studying Electrical Engineering. Nicholas obtained the office of Vice President of Eta Kappa Nu Honor Society in 2011 and held the office until the completion of his degree in 2013. He also held the honor of recruiting chair for the Institute of Electrical and Electronics Engineers between 2011 and 2013.

During the summer of 2012, he held an internship position with L-3 Communications Mission Integration Division in Greenville, Texas. While at L-3, he supported software group design and development efforts on ELINT systems to improve energy efficiency, modified mission-critical designs to increase speed and performance by ten percent and verified that new software produced in-house met customer requirements.

Nicholas graduated from the University of Tennessee in May of 2013 with honors, and became an NCEES Certified Engineer in Training. Throughout his academic career Nicholas was granted multiple scholarships including: Tennessee Academic Competitive Scholarship (2009-2013), Gonzalez Family IEEE Scholarship (2009-2013), and the UT Engineering Scholarship (2012-2013). Nicholas entered the graduate school at the University of Tennessee in the fall of 2013, where he held a graduate research assistantship. He made many contributions to his research

group including: designing hardware to leverage power efficiency and significant power savings, developing tools to automatically generate digital circuits, and several mixed-signal system designs. Nicholas plans on completing his Masters in Electrical Engineering in August of 2014, and will then continue on to Exxon Mobile in Beaumont, TX where he has accepted a position as an Instrumentation Engineer.