

Design and Calibration of a 12-Bit Current-Steering DAC Using Data-Interleaving

by

Benjamin Jankunas

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved September 2014 by the
Graduate Supervisory Committee:

Bertan Bakkaloglu, Chair
Sule Ozev
Jennifer Kitchen

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

High speed current-steering DACs with high linearity are needed in today's applications such as wired and wireless communications, instrumentation, radar, and other direct digital synthesis (DDS) applications. However, a trade-off exists between the speed and resolution of Nyquist rate current-steering DACs. As the resolution increases, more transistor area is required to meet matching requirements for optimal linearity and thus, the overall speed of the DAC is limited.

In this thesis work, a 12-bit current-steering DAC was designed with current sources scaled below the required matching size to decrease the area and increase the overall speed of the DAC. By scaling the current sources, however, errors due to random mismatch between current sources will arise and additional calibration hardware is necessary to ensure 12-bit linearity. This work presents how to implement a self-calibration DAC that works to fix amplitude errors while maintaining a lower overall area. Additionally, the DAC designed in this thesis investigates the implementation feasibility of a data-interleaved architecture. Data interleaving can increase the total bandwidth of the DACs by 2 with an increase in SQNR by an additional 3 dB.

The final results show that the calibration method can effectively improve the linearity of the DAC. The DAC is able to run up to 400 MSPS frequencies with a 75 dB SFDR performance and above 87 dB SFDR performance at update rates of 200 MSPS.

ACKNOWLEDGMENTS

I would like to take a moment to thank all of the professors, colleagues and individuals that contributed to my invaluable collegiate experience at Arizona State University. I would like to thank my advisor, Dr. Bertan Bakkaloglu, for his guidance throughout this project. His expertise and insight in analog circuit design helped transform a challenging idea into a realistic project with academic merit. I would also like to thank my committee members, Dr. Ozev and Dr. Kitchen, for all of their support throughout the thesis process. Thank you to Cliff Nixon for teaching me concepts in analog design from an industry perspective. Finally, thank you to Cirrus Logic for funding my graduate education and providing me with phenomenal internship and career opportunities. None of this would possible without the help that I have been given. For that, I am truly grateful.

TABLE OF CONTENTS

	Page
LIST OF TABLES	VI
LIST OF FIGURES	VII
CHAPTER	
1 INTRODUCTION	1
1.1 Aims and Scope of Thesis	1
1.2 Thesis Outline	1
2 DIGITAL TO ANALOG CONVERTER BASICS	2
2.1 Introduction	2
2.2 DAC Sampling Theory	3
2.3 DAC Characterization	5
2.3.1 General DAC Parameters	5
2.3.2 DAC Static Characterization	8
2.3.3 DAC Dynamic Characterization	9
2.4 Current-Steering DAC Architecture	11
3 STATE OF THE ART CALIBRATION METHODS	12
3.1 Introduction	12
3.2 DAC Design Techniques to Reduce Known Error	12
3.2.1 Dynamic Element Matching (DEM)	12
3.2.2 Return-to-Zero (RZ) Output Scheme	13
3.2.3 Differential Quad Switching (DQS)	14
3.3 Self-Measurement and Correction Calibration Methods	15
3.3.1 Digital Pre-Distortion	15
3.3.2 Mapping	16

CHAPTER	Page
3.3.3 Self-Calibration.....	18
4 DATA INTERLEAVING.....	20
4.1 Introduction.....	20
4.2 What is Data Interleaving?	20
4.3 Data Interleaving Analysis: Advantages and Disadvantages	22
5 DESIGN OF A CALIBRATED 12-BIT CS DAC USING DATA INTERLEAVING	25
5.1 Introduction.....	25
5.2 Design of a 12-Bit Data Interleaved Current-Steering DAC	26
5.2.1 Architecture.....	26
5.2.2 Biasing.....	29
5.2.3 Static Design Considerations	30
5.2.4 Dynamic Design Considerations.....	33
5.2.5 Layout Considerations.....	37
5.3 Calibration Theory, Method and Algorithm	38
5.3.1 Current Scaling Theory	38
5.3.2 Method of Calibration	41
5.3.3 Calibration Algorithm	43
5.4 Design of Calibration Circuitry.....	48
5.4.1 Comparator	48
5.4.2 Calibration DAC (CALDAC).....	50
5.4.3 Successive Approximation Register (SAR).....	52
5.4.4 Memory.....	53
5.4.5 Digital Control Circuitry.....	54
5.5 Simulation Results and Analysis.....	54

	Page
REFERENCES.....	58
APPENDIX	
A CALCULATION OF CURRENT SOURCE AREA	60

LIST OF TABLES

Table	Page
1. Unipolar Binary DAC Codes	5
2. Bipolar Binary DAC Codes	5
3. DAC Specifications and Results	26
4. Estimated Area Savings from Current Source Scaling	39
5. Error Arrangements for Single-Ended and Complementary DACs	47
6. SFDR and SNR Results before and after Calibration	56
7. SFDR Results for Various Temperatures at the Nominal Corner	57
8. SFDR Results for Various Corner Simulations with $T=27^{\circ}\text{C}$	57
9. Area of Unit Current Sources for Different Resolutions and Constant Overdrive.....	62

LIST OF FIGURES

Figure		Page
1.	Basic DAC	2
2.	Time Domain Waveforms of a DAC.....	3
3.	The ZOH vs. Ideal Transfer Function	4
4.	Generation of Ideal Filter after Reconstruction	4
5.	Architecture of a CS DAC.....	11
6.	General Mapping Calibration Block Diagram	16
7.	Frequency Spectrum of a Nyquist RZ DAC.....	21
8.	Data-Interleaving System Flow Diagram	22
9.	Spectrums of the DAC Using the 200 MSPS 2x Interleave Mode (Top) and 400 MSPS 2x Interleave Mode (Bottom).....	24
10.	Interleaving DAC Architecture.....	28
11.	Current Switch Network.....	28
12.	Signal Flow of Data before Reaching the Current Switch Network	29
13.	Biasing Circuit Used to Set the Currents in the Main DAC and CALDAC ...	30
14.	Two Stage Op-Amp used in Bias Circuit	30
15.	Output Resistance Relationship vs. Number of Turned on Current Cells ...	31
16.	Simple Model of an All Unary CS DAC	31
17.	Driver Designed with a High Switch Point.....	35
18.	Current Source Distributions for 12-Bit and 6-Bit Matching.....	40
19.	Monte Carlo Current Source Distribution for 6-Bit Intrinsic Matching.....	41
20.	Calibration Loop.....	42
21.	Stages of Calibration Algorithm	43
22.	Calibration State Flow Diagram.....	45

Figure	Page
23. Hardware to Manipulate Error Codes from Unipolar to Bipolar Format ..	48
24. Comparator Schematic.....	50
25. Bias Generator for the Comparator.....	50
26. CALDAC Architecture	51
27. Frequency Spectrum of the CALDAC.....	51
28. Example of a 3-Bit SAR.....	52
29. Memory Black Box.....	53
30. Memory Bit Cell.....	54
31. SFDR Results before and after Calibration for a 200 MSPS Update Rate.	55
32. Comparison of Spectrums (a) before Calibration (Left) and (b) after Calibration (Right)	56

CHAPTER 1

INTRODUCTION

1.1 Aims and Scope of Thesis

Digital to analog converters find wide-spread use in today's mixed signal microelectronics. The most popular DAC architecture for high speed applications uses a current steering approach, commonly referred to as a current-steering DAC. In this thesis, the aim is to present the basic knowledge of current steering DACs and provide a way to design a cutting-edge DAC based on advanced topics like self-calibration and data interleaving architectures. The goal in this thesis was to implement a current-steering DAC that incorporated high linearity (12-bits) while maintaining high update rates (400 MSPS). The design was done in the Cadence Virtuoso Environment using a 0.18 μm process. Results of the design were verified in simulation. DAC layout methods were discussed, but layout of this design was not done and left for future work.

1.2 Thesis Outline

This thesis is divided into five chapters starting with this introductory chapter. Chapter 2 presents the basics of digital-to-analog converters with emphasis on the current-steering DAC architecture. Chapter 3 is a literature review of DAC calibration methods. Chapter 4 discusses data-interleaving and shows results of the DAC performance when such a method is used. Finally, chapter 5 covers design considerations and implementation of the 12-bit current-steering DAC in addition to the theory and design of current source calibration. The chapter finalizes by proving the concepts of calibration through simulation results of the DAC system before and after calibration.

CHAPTER 2

DIGITAL TO ANALOG CONVERTER BASICS

2.1 Introduction

A digital-to-analog converter (DAC) is a system that converts digital input data into an analog output. Figure 1 shows a black box representation of a typical DAC. It consists of a digital input consisting of N-bits, a clock, an analog reference (VREF), a supply (VDD), a ground (VSS) and an analog output. The N digital input bits represent a weighted value that will determine how much of the analog reference is passed to the output. The length of N, will determine the resolution of the DAC, or how many unique levels are represented at the output. The update rate, or the rate that the analog output is sampled and held, is synchronized by the clock to produce a zero-order hold analog output. The details and theory behind the DAC sampling theory will be discussed in the following subsection.

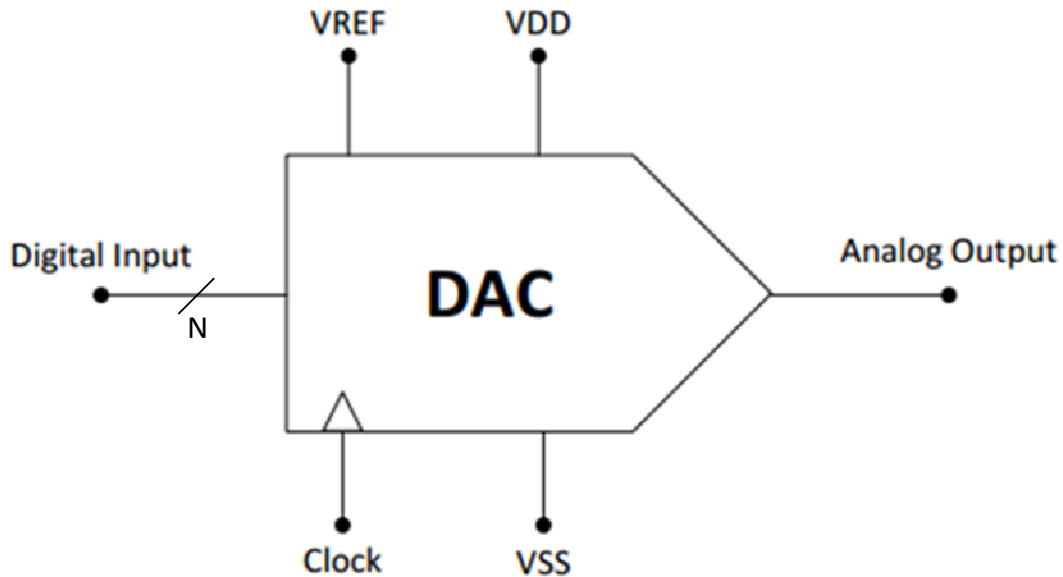


Figure 1. Basic DAC

2.2 DAC Sampling Theory

A DAC's output is held constant for a an time interval, T_s , before it updates to the next analog value. Thus, the update rate of a DAC is defined as $\frac{1}{T_s}$ and is set by the DAC clock (given that the output settling time $< T_s$). Ideally, the update should be infinitely fast since analog waveforms are continuous in nature. However, since DACs are speed limited, they will sample and hold the output. This is known as a zero-order-hold (ZOH) approximation. To gain further intuition about how the ZOH function works, the waveforms of the DAC are modeled in the time domain and then analyzed in the frequency domain.

Figure 2 shows the time domain waveforms of a typical DAC. Here, the output, $y(t)$, of the DAC is modeled by convolving the discrete, quantized data input stream, $x(t)$, with the ZOH function, $h(t)$. The resulting output is shown as a stair-stepped pattern of shifted ZOH signals whose amplitudes are determined by the weight of the digital input bits.

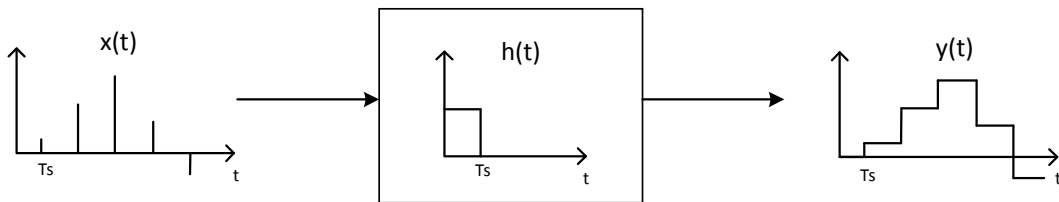


Figure 2. Time Domain Waveforms of a DAC

In the frequency domain, the ZOH transfer function will result in a sinc function with zeroes around $\pm 2\pi$ for $T_s = 1$ (Equation 2.1). Ideally, an ideal brick-wall filter would be wanted since the ZOH function exhibits droop across the frequency spectrum called $\sin x/x$ roll-off. A brick-wall filter would not be practical since the impulse response would require an infinitely long sinc to realize such a filter. Thus, the ZOH transfer function is used for its simplicity despite the $\sin x/x$ roll-off. Figure 3, compares the

brick-wall transfer function to the ZOH transfer function for $T_s = 1$. The droop from the $\sin(x)/x$ roll-off shows that higher input frequencies will suffer from performance degradation.

$$H(j\omega) = e^{-\left(\frac{j\Omega T_s}{2}\right)} \frac{\sin\left(\frac{\Omega T_s}{2}\right)}{\frac{\Omega T_s}{2}} \quad (2.1)$$

To fix the roll-off it is common to implement a reconstruction filter after the DAC. Equation 2.2 models the reconstruction filter in the frequency domain and Figure 4 shows a plot of the reconstruction filter (H_{REC}) and the ZOH transfer function (H_{ZOH}). The dotted line in Figure 4 shows the generation of the ideal brick-wall filter after multiplying H_{REC} with H_{ZOH} . Thus, the reconstruction filter can fix the $\sin x/x$ roll-off inherent in ZOH DACs.

$$H_{rec}(j\omega) = e^{\left(\frac{j\Omega T_s}{2}\right)} \frac{\frac{\Omega T_s}{2}}{\sin\left(\frac{\Omega T_s}{2}\right)} \quad (2.2)$$

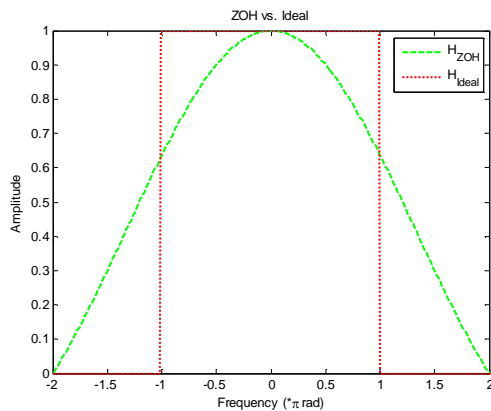


Figure 3. The ZOH vs. Ideal Transfer Function

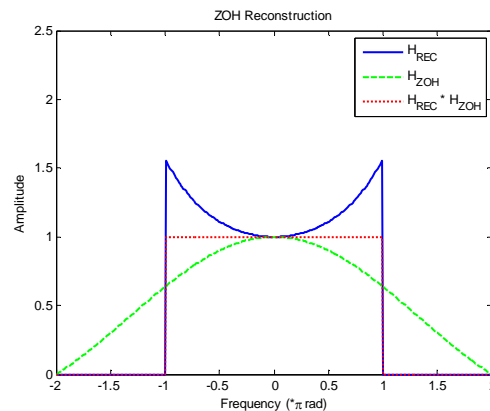


Figure 4. Generation of Ideal Filter after Reconstruction

2.3 DAC Characterization

DACs are characterized by many performance parameters that will be summarized in this section. First, general parameters of the DAC are defined and then the parameters pertaining to DAC static and dynamic performance are defined.

2.3.1 General DAC Parameters

Full Scale (FS): The full scale value is the maximum output value of the DAC.

Equation 2.3 shows the relationship between full scale and the reference voltage.

$$V_{FS} = V_{REF} - 1LSB \quad (2.3)$$

Unipolar and Bipolar DACs: A unipolar DAC will have an output that ranges from zero to FS. A bipolar DAC will have an output that will span from $-V_{REF}$ to $V_{REF} - 1LSB$. Table 1 and Table 2 show the digital to analog binary code translation for unipolar and bipolar DAC formats, respectively.

Table 1. Unipolar Binary DAC Codes

Digital Input	Analog Output
1111	$V_{REF} \left(\frac{15}{16} \right)$
1000	$V_{REF} \left(\frac{8}{16} \right)$
0000	0

Table 2. Bipolar Binary DAC Codes

Digital Input	Analog Output
1111	$V_{REF} \left(\frac{7}{8} \right)$
1000	0
0000	$-V_{REF} \left(\frac{8}{8} \right)$

Resolution: The resolution of the DAC determines the amount of unique levels that the analog output can have. For an N-bit DAC, the number of levels is given by Equation 2.4.

$$L = 2^N - 1 \quad (2.4)$$

The resolution along with the full-scale value will determine the *precision* of the DAC, or the minimum value each step can resolve. This step precision is commonly referred to as the least significant bit (LSB) and is given by Equation 2.5 for single-ended or unipolar DACs. Usually the LSB size is determined by the application of the DAC.

$$LSB = \frac{V_{REF}}{2^N} \quad (2.5)$$

Sampling Rate: The sampling rate is the rate at which the output is updated. Depending on the application, high speed DACs can have sampling rates that range anywhere from mega-samples-per-second (MSPS) to giga-samples-per-second (GSPS).

Monotonic: For a DAC to be monotonic, each increase in digital code should result in an increase in the analog output. If this does not hold, the DAC is no longer monotonic, but non-monotonic.

Binary and Unary Segmentation: There are two main digital weighting schemes that DACs commonly use: unary weighting and binary weighting. Since both schemes have their advantages and disadvantages, DACs can be segmented to incorporate both schemes into the architecture for optimal trade-off depending on DAC application.

Binary weighting is the simplest scheme for DACs. Here, each current source is binary weighted and directly controlled by the digital input bits. For an N-bit DAC, binary weighting only requires N current sources, where each current source is 2 times the size of the previous. Compared to unary switching schemes, binary requires minimal digital and analog resources [7].

There are some drawbacks to binary weighting though. With higher and higher resolution DACs, the size of the LSB current needs to become larger and larger to meet matching requirements. Furthermore, the exponential growth for successive current sources will cause large current source sizes that degrade high speed performance. Binary weighting is also easily susceptible to mid-code glitch due to mismatch in switching moments. Lastly, binary weighting does not offer redundancy in the current sources which is important in most calibration methods [7].

The other weighting scheme known as unary weighting, or thermometer coding, uses identical current sources. In unary weighting, an N-bit DAC requires 2^N-1 unary current source, where each current source is equivalent to one LSB.

The unary scheme adds redundancy to the DAC architecture that the binary weighting does not. This redundancy is vital for correcting DAC errors. Unary switches are also much easier to match and can provide high speed. With the unary scheme, however, digital control and analog support circuitry is much larger since there are 2^N-1 switches rather than N switches. This can complicate layout and routing, which in turn, degrades high speed performance.

By segmenting binary and unary schemes into one DAC, the designer can trade off the advantages and disadvantages of each scheme. Normally binary switches are used for the LSB switches and then unary switches are used for the MSB portion of the DAC. Depending on the requirements of the application, a best case segmentation

scenario can be achieved through segmentation. As it will be seen later in the design, it was chosen to segment a 12-b DAC into 8-bit binary and 4-bit unary segmentation.

2.3.2 DAC Static Characterization

The static performance of a DAC relates to its steady state characteristics. For instance, static parameters do not account for the DAC's dynamic qualities such as timing errors or settling time, but do account for DC qualities such as offset and gain error. For high speed DACs, it is of more interest to design for dynamic performance. However, design of a DAC should ultimately start with the static performance since poor static design can degrade the dynamic performance of a DAC. In this subsection, static DAC parameters including offset error, gain error, differential non-linearity (DNL), and integral non-linearity (INL) will be defined.

Offset Error: Offset error is a constant DC deviation from the ideal output code. It can be measured by setting the input code to all zeros and measuring the output. While offset is an unwanted error, it can easily be compensated by using *a posteriori* data to manipulate the digital code before it reaches the DAC [1].

Gain Error: Gain error is the deviation of the full scale output from the ideal full scale output. When offset has been removed it can be measured by applying an input code of all logic ones (V_{111}). Like offset, gain error can be easily compensated if necessary [1]. Equation 2.6 shows how the gain error can be measured.

$$\text{Gain Error (\%)} = \left[\frac{V_{111}}{V_{FS} - 1LSB} - 1 \right] 100 \quad (2.6)$$

Differential Non-Linearity (DNL): DNL is the difference between sequential output levels measured in LSBs. Ideally the DNL should be 0 LSBs. If the DNL is larger than 1 LSB, the DAC will no longer be monotonic.

$$DNL(n) = \left[\frac{V(n) - V(n+1) - V_{LSB}}{V_{LSB}} \right] \quad (2.7)$$

Integral Non-Linearity (INL): INL is the deviation of the ideal DAC transfer curve from the actual transfer curve measured in LSB's or as a percentage of the full-scale output voltage.

$$INL(n) = \left[\frac{V(n) - V_{ideal}(n)}{V_{LSB}} \right] \quad (2.8)$$

2.3.3 DAC Dynamic Characterization

The dynamic performance of the DAC includes the transient mode of operation. It is commonly characterized by spectral parameters such as the signal-to-noise and distortion ratio (SNDR) and the spurious free dynamic range (SFDR). Errors in the dynamic qualities of a DAC are mainly attributed to the following [8]:

- 1) Code-dependent settling time
- 2) Code-dependent switch feed-through
- 3) Timing skew between current sources
- 4) Major carry glitch
- 5) Current source switching
- 6) On-chip passive analog components
- 7) Mismatch

Design techniques to reduce the dynamic errors will be discussed in Chapter 5.

Signal to Noise and Distortion Ratio (SNDR): For an ideal DAC with noise only due to quantization noise the SNDR is given by Equation 2.9.

$$SNDR = 6.02N + 1.76 \quad (2.9)$$

Spurious Free Dynamic Range (SFDR): SFDR is measured as the difference between the input signal power and the power of next largest spur from 0 to the Nyquist frequency. SFDR will degrade at higher speeds as dynamic effects introduce more harmonic distortion into the output signal.

$$SFDR = P_{signal} - P_{spur} [dB] \quad (2.10)$$

Effective Number of Bits (ENOB): For an ideal DAC, the only noise is due to quantization. However, real DACs will have other sources of noise that degrade the SNR. Ideally, the ENOB should approach the specified bit resolution (N), but any noise degradation other than quantization noise will lower the ENOB. Equation 2.9 can be rearranged to get the ENOB (Equation 2.11).

$$ENOB = \frac{SNDR - 1.76}{6.02} \quad (2.11)$$

Glitch: An under/overshoot of the output waveform during the switching moments of the DAC is called a glitch. Glitches are caused by either capacitive coupling and clock feed-through at the DAC output or mismatches in switch timing.

The worst case glitch occurs at the midscale transition for binary weighted DACs since all of LSBs will turn on or off while the MSB switch will turn off or on. At this

point, any timing error will cause the output to glitch and induce harmonic distortion into the spectrum. Glitch can be reduced by segmenting the architecture into binary and unary switches. If DACs are segmented in an all unary fashion, then only one switch will turn on or off reducing the glitch dramatically. However, as it is discussed later, an all unary switching scheme will call for complex control circuitry that complicates the layout. Complex layouts make it hard to align all signals and larger timing errors will occur and glitch becomes more prominent.

2.4 Current Steering (CS) DAC Architecture

Figure 5 shows a basic 4-bit segmented current steering DAC. Here, the DAC is segmented into 2-bit binary and 2-bit unary current sources that will steer current to either the positive or negative terminals based on the input bits. The 2-to-3 decoder converts the two uppermost input bits into thermometer code, while the delay block delays the two lowermost bits (by the decoder delay) to ensure all switching moments occur simultaneously.

The two output terminals in current steering (CS) DACs ensure that the current is never shut-off. The negative output current can either be steered to ground for unipolar DACs or used as a complementary output for a bipolar DAC, where complementary option is the most common method [1].

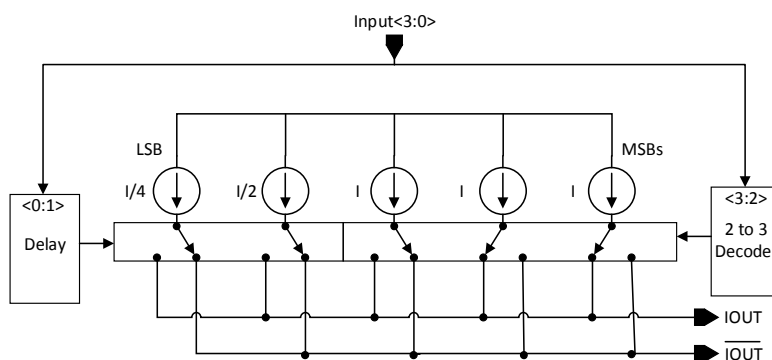


Figure 5. Architecture of a CS DAC

CHAPTER 3

STATE OF THE ART CALIBRATION METHODS

3.1 Introduction

There are many types of calibration methods for current steering DACs that work to calibrate resulting errors from static and dynamic non-linearity. In this chapter, a literature review of various calibration methods is presented. First, DAC design methods to reduce errors without any feedback are reviewed including dynamic element matching (DEM), differential quad switching (DQS), and the return-to-zero (RZ) output scheme. Finally, calibration methods that use a feedback loop to measure and correct DAC errors are investigated including digital pre-distortion, mapping, and self-calibration.

3.2 DAC Design Techniques to Reduce Known Error

Calibration techniques that do not require error measurement circuitry are discussed in this section. These methods rely on known errors of the DAC and work to correct the errors with intrinsic design techniques.

3.2.1 Dynamic Element Matching

Dynamic element matching (DEM) is a calibration technique that uses the redundancy of unary DAC cells to improve the SFDR performance. Essentially, DEM time averages the errors or spreads the harmonic distortion products of the DAC across the frequency spectrum by randomizing the sequence in which current switches are selected. DEM improves SFDR, but not SNDR since distortion is translated into white noise. DEM techniques do not require knowledge of the mismatch inside DACs and are implemented in the digital domain. There are many unique DEM algorithms where each

technique has its own special feature, but the basic idea is always the same; DEM techniques convert spurious noise into white noise.

One DEM method in literature uses a technique to minimize the glitch energy in a current steering DAC in order to reduce SFDR and obtain a lower noise floor [10]. One way of doing this is by finding the most efficient switch coding algorithm that minimizes the current switches turning on or off for each successive input. The thermometer coding scheme meets this requirement but does not randomize the selection of current cells. Thus, [10] proposes random swapping thermometer coding (RSTC). RSTC is an efficient switching algorithm, like thermometer coding, that also randomizes the selection of current cells. RSTC has been shown to exhibit better spectral performance than other DEM algorithms such as data-weighted averaging [10].

DEM techniques can be extended to many circuit applications in which circuit redundancy can be exploited. It can fix unwanted spurious noise due to dynamic glitches and is digital based making it a practical option for future scaling nodes. However, DEM does not improve the overall SNDR since it spreads distortion into white noise. Approaches in literature employ DEM as a standalone calibration or pair DEM with another correction method.

3.2.2 Return-to-Zero Output Scheme

The RZ output scheme works by tracking the output in the first half of the clock period and resetting the output to ground in the second half of the clock period. By setting the output to ground during the switching moments of the DAC, data dependent glitches are blocked from the output and thus, dynamic performance can be enhanced [8]. Transient errors due to the clock switching may still be present at the output, but the frequency of the errors will be at the clock frequency and outside of the signal band.

Therefore, the clock dependency of the RZ output is an advantage over the data dependencies of the NRZ output scheme [9].

Although the dynamic performance of DACs can be improved through the RZ output scheme, RZ usually means slower speeds since the time for the current to settle is reduced by half. RZ schemes also reduce the total signal output power in half. However, the speed and signal power can be fixed by using a data-interleaved output scheme that will be discussed in Chapter 4.

3.2.3 Differential Quad Switching

Differential quad switching (DQS) reduces data dependent errors by making switching transients dependent on the clock frequency which has the effect of pushing the glitch, feed-through and other transient errors out of the signal band. To realize DQS, one of the data switches needs to change state every clock edge. This is not possible when the data remains at the same state during subsequent cycles. Therefore, an additional transistor is placed in parallel with each current switch in the DAC. In this way, if the state of the data remains the same, the original switch and parallel switch can swap states ensuring that the original output remains the same and that a switching transition occurs for each clock transition. The disadvantage of this scheme is that it requires two additional transistors per current cell and increases the dynamic switching power. DQS is also susceptible to data dependent errors due to switching mismatches [19].

3.3 Self-Measurement and Correction Calibration Methods

The following calibration methods measure and correct existing errors through feedback. These calibration methods rely on *a posteriori* information rather than *a priori* information. The methods discussed include digital pre-distortion, mapping, and self-calibration.

3.3.1 Digital Pre-Distortion

Digital pre-distortion is a technique that uses digital compensation to enhance the spectral qualities at the analog output. In [11], this is done by sampling the DAC's output signal and converting it back to a digital format to compare its discrete Fourier transform to an ideal input. The difference between each signal's spectral components is translated into an error vectors. Each error vector is converted to the time domain and subtracted from the digital input to better resemble an ideal output. The algorithm then repeats itself until the error vectors converge within an acceptable range.

The digital pre-distortion method effectively reduces the harmonic distortion and pushes components out of band. However, drawbacks to this algorithm, according to [11], include limitations on the input signal's complexity, the possibility of the algorithm not converging, and the inability to correct for gaps in code due to large DNL errors at the major carry transitions. Even so, if the signal is relatively non-complex and the inherent DAC design can meet DNL requirements, then digital pre-distortion methods can be a practical method for calibration; this is especially true when used in DDS systems since it can calibrate the non-linearity of the whole system.

3.3.2 Mapping

Like DEM, mapping algorithms take advantage of the DAC current cell redundancy for error calibration. However, instead of randomly scrambling the selection of unary current cells, mapping will select a specific line-up or *map* of unary current cells that corresponds to either the best static or dynamic performance. Figure 6 shows the signal flow diagram for a general mapping algorithm. Here, the error measurement of each unary DAC cell is performed in the foreground. Meanwhile, the mapping algorithm finds the best case map based on the error information. The maps are stored in the mapping memory for future use during the normal operation of the DAC. In the figure below, N and S correspond to the digital input and the input to the DAC respectively. For all mapping cases, $S > N$ since the mapping algorithm only operates on weighting that is < 2 (i.e. unary weighting) [7]. For an all unary DAC, S is equal to 2^{N-1} .

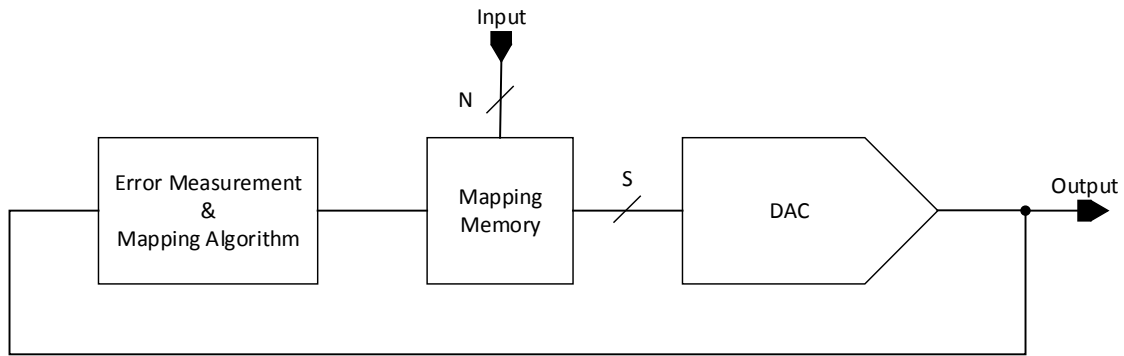


Figure 6. General Mapping Calibration Block Diagram

There are three types of mapping found in literature: static mismatch mapping (SMM), timing error mapping (TEM) and dynamic mismatch mapping (DMM) [14]. Static mismatch mapping is the more general mapping technique that reduces the static error of the DAC. Timing error mapping will work to find a map for the timing errors. Normally, mapping schemes will result in a best case map for good static performance or

good dynamic performance, but not both. However, in the case of DMM, both static and dynamic performance is enhanced. Below, examples of SMM, TEM and DMM are discussed.

An SMM technique in [12] reduces the amplitude errors through a “switching sequence post adjustment calibration.” Basically, a comparator is used to measure the relative amplitudes of each unary current source. Then a multi-step algorithm rearranges the selection of current sources to get the best case INL figure. The rearrangement is based on pairing current sources with equal magnitudes of positive and negative amplitude errors together to mutually compensate each other. This mapping technique shows that it improves INL drastically with the use of an added comparator, digital control circuitry and RAM for the map sequence.

Timing error mapping focuses on the timing errors of the DAC, but do not address the static errors. In timing error mapping a phase detector can be used to measure each current source’s phase relative to a reference waveform’s phase [13]. The measured phase values can then be used to find an optimal map for dynamic performance. For instance a current cell with a negative phase can be mapped with a current cell with positive phase with similar magnitude. It should be noted that another technique found in [13] discusses how a controllable digital delay line can be used for each unary current source to change the timing. This technique is not a mapped based method, but a self-calibration method that is discussed in the next section.

There has been a novel approach to the mapping calibration methods that addresses both static and dynamic errors. This technique is called dynamic mismatch mapping (DMM) [14]. It defines two new dynamic terms: namely, dynamic INL and dynamic DNL. Instead of relating INL/DNL errors to the static transfer curve, the dynamic INL/DNL errors are related to a dynamic transfer curve in the IQ plane that

shows both phase and timing. The dynamic transfer function is found by modulating each current source with a square wave of modulation frequency f_m . The ideal dynamic transfer function consists of a linear line through the IQ plane meaning that the difference between each successive code is equal in magnitude and phase. However, in reality both magnitude and phase of the current source will deviate from the ideal line when a current cell is modulated with a square wave at frequency f_m . In the IQ plane, the deviation from ideal line is labeled as the “dynamic-mismatch error vector” and includes both amplitude and phase errors. A best case map can be found that reduces both static and dynamic errors because the phase and magnitude are both represented in the “dynamic-mismatch error vector.”

3.3.3 Self-Calibration

Self-calibration techniques work to measure the amplitude or timing errors of the DAC and directly correct for it. For instance, a unary current source is modeled by Equation 2.12, where I_N is the amplitude of current source N, I_{NOM} is the nominal unary current source value and I_{ERR} is the error current due to random mismatch. If the error current, I_{ERR} , is measured then it can be subtracted from the unary current value, I_N , to recover the nominal design current, I_{NOM} . Similarly, self calibration methods can correct for timing errors like the DDL method presented in [13]. In Equation 2.13, the time it takes in a DAC for a signal to travel from the output of the data latch to the current switch gate is modeled as t_N . This time, t_N , is equivalent to a nominal time, t_{NOM} , plus or minus some error time, t_{ERR} . Timing errors are measured and corrected for each t_N to improve dynamic performance.

$$I_N = I_{NOM} + I_{ERR} \quad (2.12)$$

$$t_N = t_{NOM} + t_{ERR} \quad (2.13)$$

Self calibration methods will measure, process and eliminate the errors. The measurement is done in the analog domain, the processing is done in either the analog or digital domain and the correction is done in the analog domain. Self-calibration methods are either done with a foreground or background implementation. Foreground techniques only run during a reset phase or upon chip set-up and measure time-invariant errors. The foreground technique has the advantage of being shut-off during the normal operation for better performance of the DAC. However, if time varying errors need to be calibrated, a background calibration that is always running can be implemented. Background calibration could potentially interact with the analog signals in the DAC and negatively impact the performance.

In this thesis, a self-calibration method based on current cell correction was implemented. The details of the design implementation including the measurement, algorithm, and correction can be seen in the Chapter 5.

CHAPTER 4
DATA INTERLEAVING

4.1 Introduction

Data interleaving is a method that can be used to increase the bandwidth in DAC applications by eliminating the unwanted image frequencies and effectively increasing the Nyquist frequency. In this research, the method was considered and implemented in design using two RZ DACs. In the following section, data interleaving is discussed including its limitations and advantages.

4.2 What is Data Interleaving?

Due to the sampling frequency requirement, Nyquist rate DACs can only generate signals at frequencies up to half the sampling frequency without aliasing. A typical frequency spectrum of a Nyquist rate RZ DAC with ZOH sampling is modeled by Equation 2.14 and plotted in Figure 7. Here, the input frequency, f_{in} , is normalized to the sampling frequency or clock frequency, f_s . As it can be seen, unwanted images are generated at $1 \pm f_{in}/f_s$. At higher and higher input frequencies, the image spur will become closer to the band edge ($f_{in}/f_s = 0.5$), requiring higher and higher orders of analog filtering to effectively remove the image spur. If however, the images can be removed, analog filtering requirements will be greatly reduced and input signals can be generated beyond the Nyquist frequency requirement. This can be realized with data interleaving.

$$H_A(f_{in}) = 0.5 \frac{\sin\left(\frac{\pi}{2} \cdot \frac{f_{in}}{f_s}\right)}{\frac{\pi}{2} \cdot \frac{f_{in}}{f_s}} \sum_{n=0}^{\infty} h(f_{in} - n \cdot f_s) \quad (2.14)$$

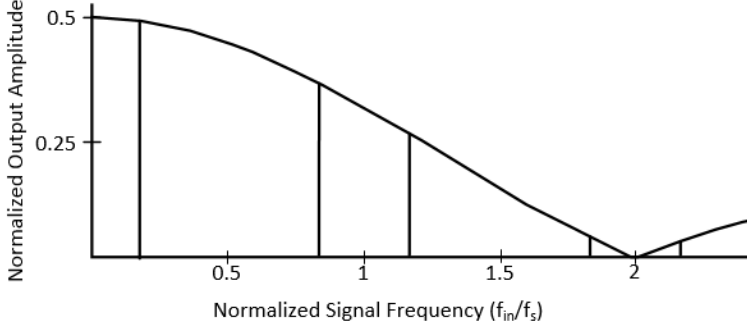


Figure 7. Frequency Spectrum of a Nyquist RZ DAC

Figure 8 shows the signal flow diagram for a data-interleaved DAC based on [20]. This method effectively adds the output of two sub-DACs whose fundamental signal frequencies are equal in magnitude and phase and whose images are equal in magnitude and opposite in phase relative to each other. Thus, by the addition of the two DACs, the images can be removed while retaining the signal frequency. To realize a data-interleaved configuration, there are a few requirements. First, each sub-DAC is given every other sample of the input stream coming in from the “Data Sorter” shown in Figure 8. For instance, DAC-A receives all of the even data samples and DAC-B receives all of the odd data samples. Additionally, the two sub-DACs are clocked 180° out of phase relative to one another. DAC-B can be modeled by implementing a phase delay in the frequency response as shown below. By introducing a phase delay by 180° the fundamental signals between the two DACs are the same phase, while the images are opposite in phase.

$$H_B(f_{in}) = 0.5 \frac{\sin(\frac{\pi}{2} \cdot \frac{f_{in}}{f_s})}{\frac{\pi}{2} \cdot \frac{f_{in}}{f_s}} \sum_{n=0}^{\infty} h(f_{in} - n \cdot f_s) \cdot e^{-jn\pi f_{in}/f_s} \quad (2.15)$$

The total output spectrum is modeled in Equation 2.16 which is the sum of Equation 2.14 and Equation 2.15. Figure 8 shows an example spectrum at the output of each sub-DAC and the resulting spectrum when they are added together.

$$H_{tot}(f_{in}) = H_A(f_{in}) + H_B(f_{in}) \quad (2.16)$$

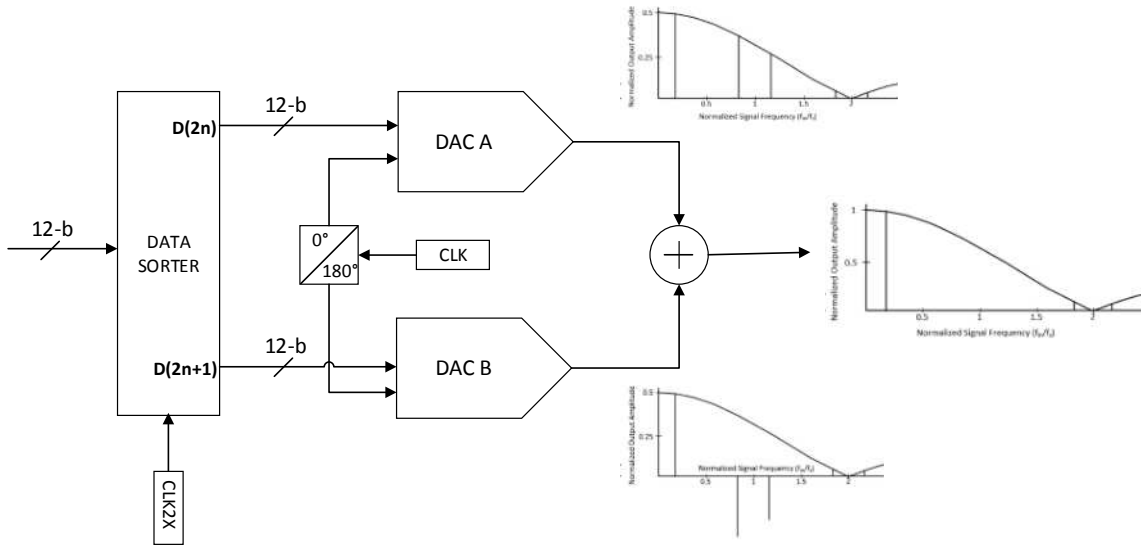


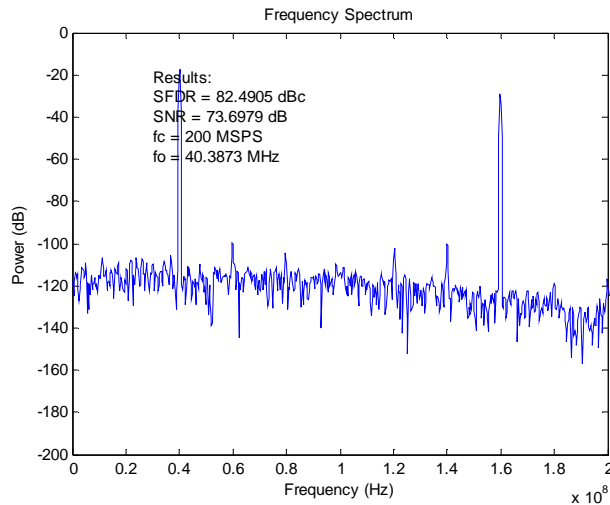
Figure 8. Data-Interleaving System Flow Diagram

4.3 Data Interleaving Analysis: Advantages and Disadvantages

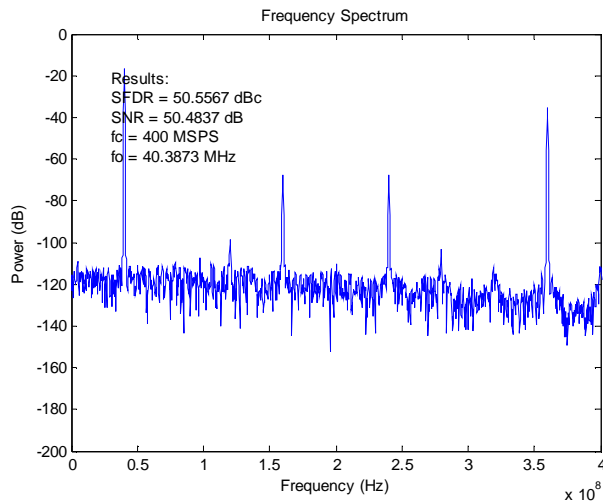
By using data interleaving, some of the drawbacks from the RZ scheme can be eliminated. It is known that RZ schemes are susceptible to decreased settling time and half the output power, but interleaving fixes both of these problems. Settling time is fixed since the bandwidth can be increased by two. This makes the fraction of settling time relative to bandwidth the same as an NRZ DAC. Also, the signal power becomes the same as an NRZ DAC with interleaving due to doubling of the power when the two DACs' outputs are added together; the drawback of this is that two DACs must be used instead of one. Using the RZ scheme with data interleaving is also common due to less $\sin(x/x)$

droop than an NRZ DAC since the zero null in the sinc occurs at twice the sampling frequency rather than the sampling frequency. NRZ DACs can incorporate data interleaving, but their large degradation in SNR due to droop makes it unlikely for NRZ DACs to generate signals above the original Nyquist frequency. Other advantages to data interleaving include better quantization of the noise. The SNR is improved 3 dB; noise power is uncorrelated and increases by 2 while signal power is quadrupled.

Despite the advantages of data interleaving, implementation can be difficult. Interleaving is susceptible to errors due to phase and amplitude alignment. Clock skew, amplitude mismatch or any timing misalignment between the sub-DACs will result in imperfect matching of the image frequencies. If mismatch is present, images will not be eliminated and render the data interleaving method ineffective. At higher frequencies cancellation of the images becomes non-trivial. Additional calibration methods (i.e. phase alignment) are necessary. In this work calibration of the data interleaving alignment was not implemented in design and the degradation of high frequency performance can be seen in the results when using the interleaving mode. Figure 9 shows the resulting spectrums of the DAC designed in this thesis at the 200MSPS 2x interleaved mode and at 400MSPS 2x interleaved mode, respectively. The input frequency for both modes was 40.111MHz. In the 200MSPS 2x interleaved spectrum, each sub-DAC is clocked at 100 MSPS and added together to create a 200MSPS spectrum. Images in this spectrum will occur at $100\text{MHz} \pm 40.111\text{ MHz}$. For the 400MSPS interleaved mode, each sub-DAC is clocked at 200 MSPS causing the images to occur at $200\text{MHz} \pm 40.111\text{MHz}$. As shown, the images in the 200MSPS spectrum are almost completely eliminated while the images in the 400MSPS spectrum are attenuated, but still prominent. This proves that image cancellation is more sensitive to timing mismatch at high frequencies.



- Effective Sampling Rate = 200 MSPS
- Sub-DAC Sampling Rate = 100 MSPS
- Input frequency = 40.111 MHz
- SNDR = 73.6 dB
- SFDR = 82.5 dB
- ENOB = 11.9 bits



- Effective Sampling Rate = 400 MSPS
- Sub-DAC Sampling Rate = 200 MSPS
- Input Frequency = 40.111 MHz
- SNDR = 50.48 dB
- SFDR = 50.56 dB
- ENOB = 8.1 bits

Figure 9. Spectrums of the DAC Using the 200 MSPS 2x Interleave Mode (Top) and 400 MSPS 2x Interleave Mode (Bottom)

Data Interleaving can be an effective way to increase the bandwidth in DACs, but methods to calibrate the mismatch between the images are necessary for very high frequency applications. In this design, calibration would be necessary above the 200 MSPS 2x mode to effectively use data interleaving beyond this update rate. Future work in the calibration of data-interleaved image mismatch could be a unique and rewarding research opportunity.

CHAPTER 5

DESIGN OF A CALIBRATED 12-BIT CS DAC USING DATA INTERLEAVING

5.1 Introduction

In this chapter the calibration method and design procedure for a 12-bit data interleaved current steering DAC is discussed. Section 5.2 starts with the architecture and biasing of the DAC and then moves on to discuss common design considerations for optimal static and dynamic performance of a DAC. The section wraps up with some layout considerations. In section 5.3, the calibration method and algorithm are reviewed. The effects of scaling current sources are analyzed and the specific design details needed for the self-calibration circuitry are included to explain the calibration method and algorithm. In section 5.4, the design of the extrinsic calibration circuitry is reviewed and showcased. The calibration components include a comparator, SAR, CALDAC and a memory bank. Finally, section 5.5 presents the final simulation results of the DAC. Here, results show a sweep of the SFDR before and after calibration and show that calibration improves the performance. The specifications and results of this DAC are summed up in Table 3.

Table 3. DAC Specifications and Results

Specification	Result
Resolution	12 bits
Update Rate	Up to 400 MSPS
Full Scale Current	20.078 mA
Load	25 Ω
Technology	0.18 μm CMOS
Power Supply (Digital and Analog)	1.8 V
Average Power Consumption (Normal Mode)	127 mW
SFDR	See Results Section 5.4

5.2 Design of a 12-Bit Data Interleaved Current Steering DAC

The intrinsic design of a 12-bit data interleaved current steering DAC is discussed in this section. This section begins with the DAC architecture, then continues on to design considerations for static and dynamic performance and ends with layout considerations.

5.2.1 Architecture

The core DAC architecture designed in this thesis is shown in Figure 10. The DAC consists of a total of 23 equivalent current sources for each of the two sub-DACs (DAC-A and DAC-B). Each DAC can be separated into an 8-bit binary LSB and 4-bit unary MSB segmentation. The LSB portion is composed of eight unary current sources that are steered to an R2R network and the MSB portion is composed of 15 unary currents that are directly connected to the output. An additional current is added to be

used as a reference for calibration. The current switches will steer current to either the positive or negative output nodes depending on the data inputs and the clock polarity.

A transistor level view of the current switch and current source is shown in Figure 11. Here, M1 and M2 combine to form the cascode current source; M3 and M4 are the current switches that are controlled by the complementary data inputs; and M5 through M8 are the return-to-zero switches controlled by the complementary clock inputs. The transistors M9-M11 are used to route the current to the calibration measurement circuitry during the foreground calibration mode.

Figure 12 shows the signal flow path of the digital inputs before reaching the current switches of the DAC. This path consists of binary-to-thermometer encoders, delay blocks, data drivers, and clock drivers. Driver design techniques such as lowering the signal swing and using a high switch point were used for optimal dynamic operation.

It should be noted that the all unary architecture used in this design has the advantage of being able to calibrate the amplitude errors of all MSB and LSB current sources if one chooses. This is unlike most DACs since they will only be able to calibrate the unary MSB section and not the binary LSB section. If the LSB calibration is necessary, the LSB errors can be measured by bypassing the R2R network through the “current path for calibration measurement” (see Figure 10). A simple binary scaling could then be applied in the digital domain to get the correct error code for each LSB. LSB error correction was not done in design since it also requires additional memory and digital control. However, if it were necessary, this architecture can easily incorporate LSB error correction.

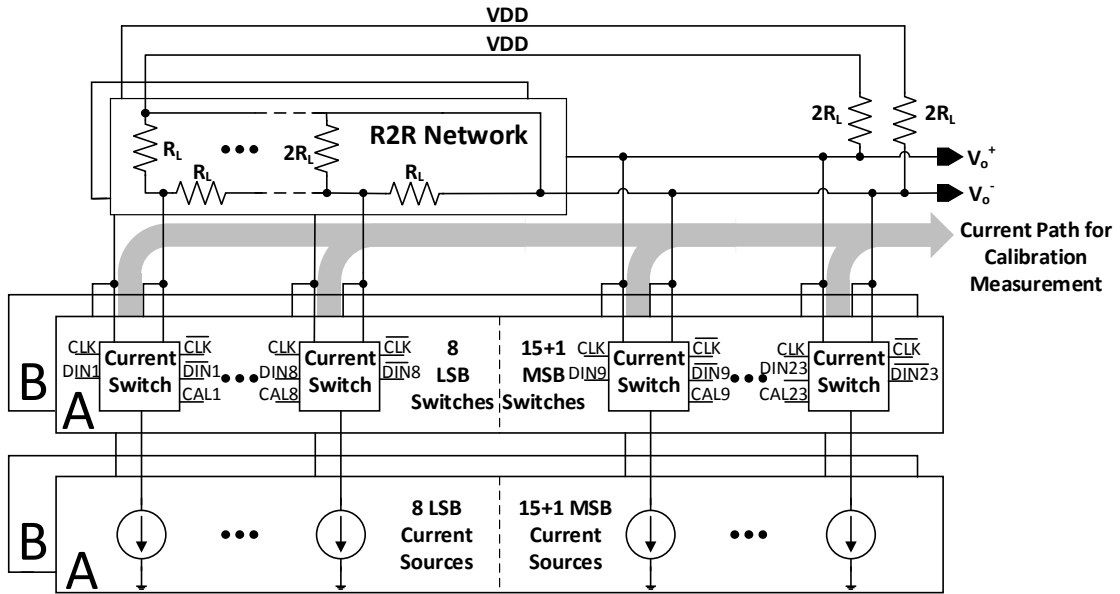


Figure 10. Interleaving DAC Architecture

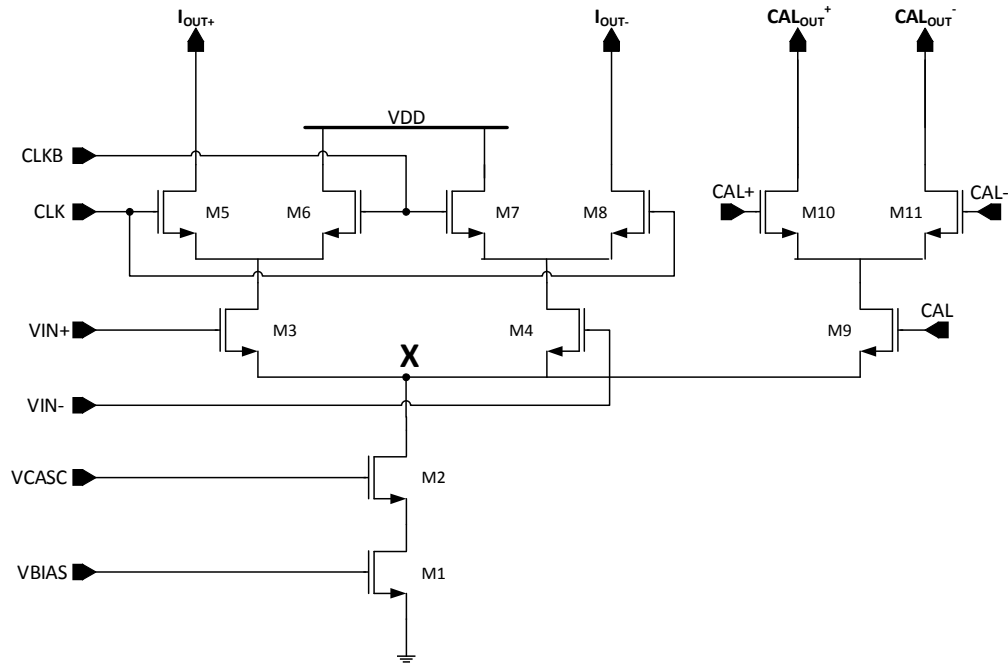


Figure 11. Current Switch Network

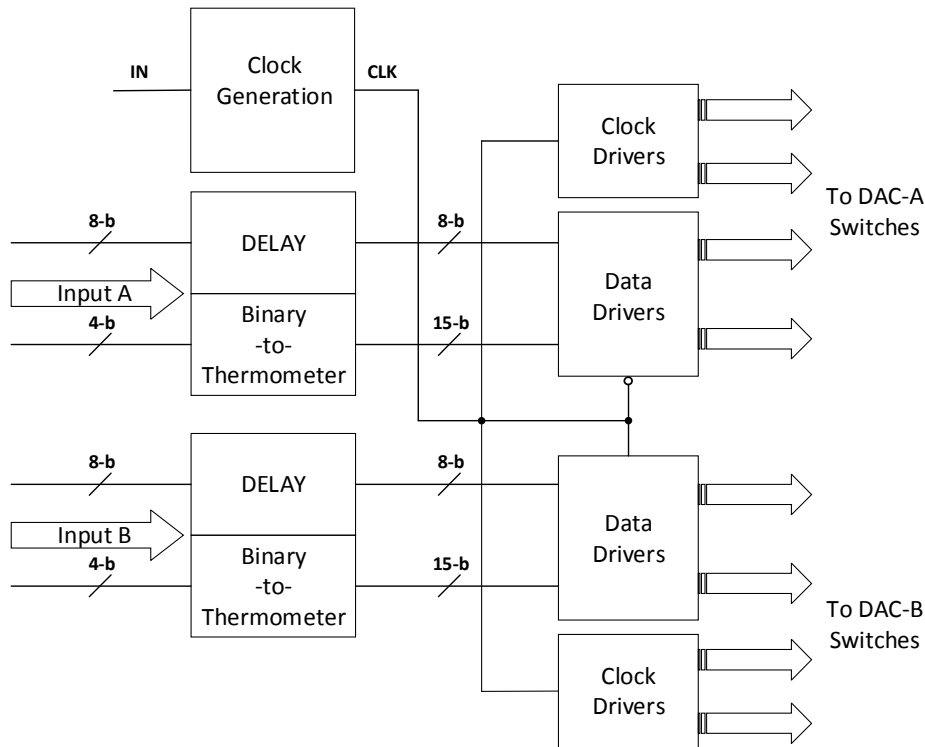


Figure 12. Signal Flow of Data before Reaching the Current Switch Networks

5.2.2 Biasing

The bias circuit of this DAC uses a two stage op-amp (Figure 14) to set the current in each of the PMOS transistors shown in Figure 14. Four, 1.25 mA currents are generated for the main DAC and four, 39.0625 μ A currents are generated for the CALDAC. Each current is sent to a wide swing NMOS cascode current mirror that sets bias points for all the current source transistors, M1 and M2 in the current switch network (Figure 11). Note that in Figure 13, the resistor was chosen to be 500 Ω making VREF 625 mV.

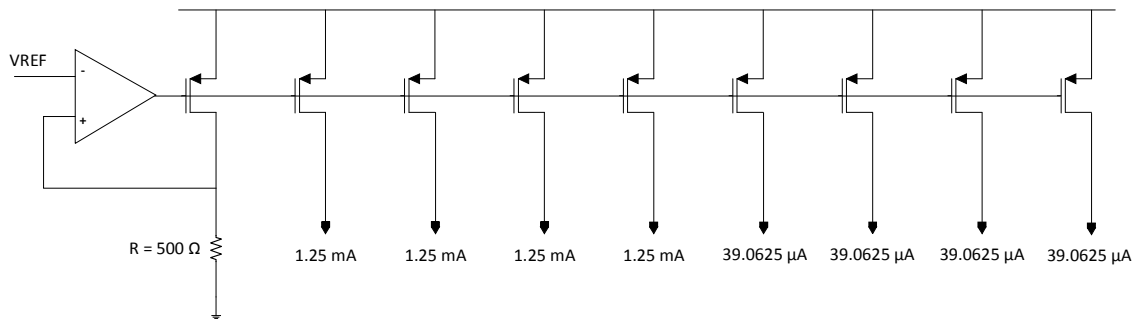


Figure 13. Biasing Circuit Used to Set the Currents in the Main DAC and CALDAC

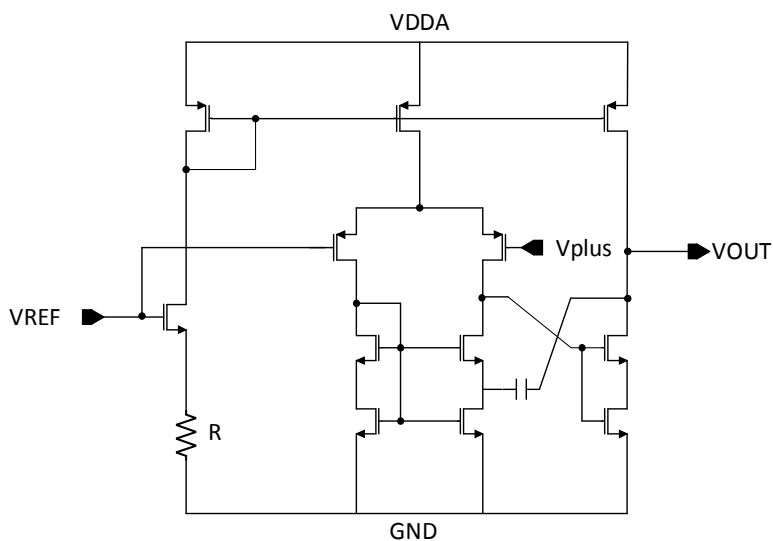


Figure 14. Two Stage Op-Amp Used in Bias Circuit

5.2.3 Static Design Considerations

In this section, static design methods are discussed in order to meet INL and DNL requirements for a current steering DAC. Two qualities in a current-steering DAC that ensure good static performance include a high output impedance and a good match between current source transistors. The following sections will discuss why these characteristics are needed and how they can be achieved.

Output Impedance

The output impedance of a DAC is a function of the number of unit current cells turned on at one time. The plot in Figure 15 shows that the output impedance linearly decreases for an increase in the number of the unary current sources switched on and Figure 16 shows a simple model of an all unary DAC from which the output impedance trend can be derived. In this figure, the variable k represents the number of unit current sources that are on; R_L represents the load resistance; and r_o represents the resistance of a unit current source.

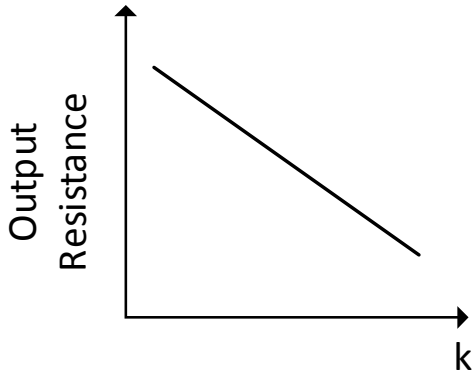


Figure 15. Output Resistance Relationship vs. Number of Turned on Current Cells

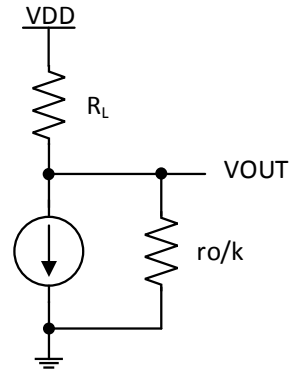


Figure 16. Simple Model of an All Unary CS DAC

If the DAC design has too low of an output impedance there will be implications for both static and dynamic performance. Ideally, the output impedance should be equivalent to the load resistance, R_L , but as more currents are switched to the output the value r_o/k decreases and the equivalent output impedance may no longer be sufficient to meet static INL and DNL requirements. Therefore, before design of a DAC, the requirement for the output resistance of the current switch network should be considered. For an all unary DAC implementation the resistance requirement is given in Equation 2.17 for a single ended design [15]. Here the value I is the unit current; N is the number of current sources; and $INL_{SE}(k)$ is the INL for the k^{th} current source in a

unipolar DAC. The requirement for the output impedance for a fully-differential DAC is even less than the single-ended requirement. This is because the second order harmonics are cancelled [16].

$$r_o = \frac{I \cdot R_L \cdot k(k - N)}{INL_{SE}(k)} \quad (2.17)$$

In this DAC, output impedance requirements were met easily with a cascode current source and INL/DNL results were all under $1/2$ an LSB.

Matching Requirements

Matching of current sources is one of the main concerns of a DAC design. Any random mismatch of the current sources will cause amplitude errors in the DAC output and degrade performance.

In order to achieve an inherent DAC resolution, the area of the current sources must meet the requirement given in Equation 2.18. Here, A_β and A_{V_T} are process parameters; $\frac{\sigma_{\Delta I_D}^2}{I^2}$ is the relative variation in drain current of a unit current source and can be derived by Equation 2.15; $(V_{GS} - V_T)^2$ is commonly known as the overdrive voltage; and WL is the area of one unit current source transistor. By substituting Equation 2.19 into Equation 2.18, it is shown that the area of the current sources needs to be doubled for an increase in resolution by 1-bit. Therefore, there is a trade-off between resolution and speed, since larger and larger current sources will reduce the overall speed of the DAC.

$$WL = \frac{1}{\frac{\sigma_{\Delta I_D}^2}{I^2}} \left[A_\beta + \frac{4A_{VT}}{(V_{GS} - V_T)^2} \right] \quad (2.18)$$

$$\frac{\sigma_{\Delta I_D}}{I} = \frac{1}{2\sqrt{2^N - 1}} \quad (2.19)$$

For an inherent 12-bit resolution DAC with a 6-sigma matching confidence and an overdrive voltage of 300 mV, an area of about 60 μm^2 is required for the unit current source. This means that the MSB current source for an all binary DAC would be required to be $2^{11} \times 60 \mu\text{m}^2$ or about 0.123 mm^2 . The impractically large area of the current sources shows that an all binary, high-resolution DAC would be infeasible for high speed applications. To reduce the area of the largest MSB current sources in a binary DAC, unary and binary segmentation is commonly done. In this design, as it will be seen, the DAC was segmented with an 8-bit LSB binary and 4-bit MSB unary segmentation. After segmentation, further reduction in current source size is done to drop the area even more. This has the advantage of increasing the speed of the DAC by lowering internal capacitance seen at the current source node as well. However, further reduction will cause amplitude mismatch in the current cells. Calibration must be used to fix the amplitude errors due to scaling current source transistors. In this thesis, scaling past the intrinsic matching requirement was done and a self-calibration method to correct the resulting errors was used. The details of this will be discussed in section 5.3.

5.2.4 Dynamic Design Considerations

In this section, design methods to improve the dynamic performance of a current-steering DAC are considered. The main components affecting dynamic performance are the switches and switch drivers. Therefore, careful design of these

components needs to be done. In literature, there are many techniques for switch and switch driver design to improve the dynamic performance of DACs by reducing the signal feed-through, current source node settling, and local and global timing errors. The following will discuss the methods used in this DAC to reduce these errors.

Signal Feed-through

When there is a signal transition at the current switch's gate, a portion of the signal is sent to the output node through the parasitic capacitance. The signal that is fed on to the output is called signal feed-through. The amount of signal feed-through seen at the output of the DAC is dependent on the magnitude of the channel and overlap capacitances of the switch transistors. It also depends on the voltage swing and rise/fall times of the signal seen at the switch transistor's gate [17].

In this DAC, signal feed-through from the data switch transistors, M3 and M4, is minimized since it will be buffered by the clock switch transistors, M5 and M8 (Figure 11). It is also true that there will be clock feed-through from the clock signal at transistors M5 and M8. However, since the clock feed-through is independent of the input data, it should show-up outside of the signal band. Nevertheless, since second order effects will cause clock feed-through to be dependent on the data all current switch transistors, M4-M8, were designed to reduce the total feed-through. One way of doing this was to reduce the total swing of the signals. To do this both clock and data drivers supplied a 0.6V swing rather than a 1.8 V swing to the switch transistors. Additionally, the switches were sized as small as possible to reduce the parasitic capacitance. The limitations on sizing and swing depended on the possible headroom available to keep all transistors in saturation.

Current Source Node Disturbances

During the switching moments of the DAC, a voltage deviation occurs at the current source node X seen in Figure 11. This disturbance is dependent on the change in voltage and the capacitance at these nodes. The voltage spike will cause data-dependent errors to occur at the output since it is modulated by the input. To reduce this effect, it is common to employ a high switching point between complementary signals provided by the signal drivers. This technique ensures that both switch transistors remain conducting throughout the switching moment. If both transistors were to be off at the same time, a large transient would occur due to the rapid voltage change at this node and a current glitch can be seen at the output.

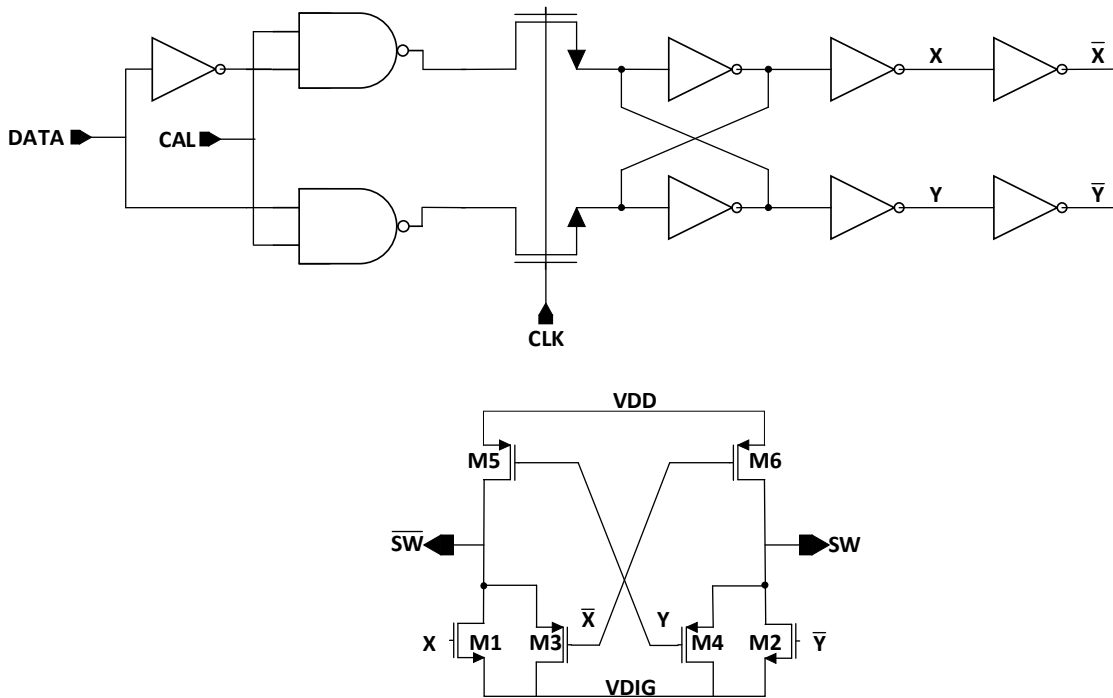


Figure 17. Driver Designed with a High Switch Point

The driver employing a high switch point in this thesis is shown in Figure 17 above and is based on the work in [18]. To get a high switch point the rise and fall times

of the signals SW and \overline{SW} are unequal. For example, the rise time of the each signal is always fast and the fall time is slew limited. In this way, a high switch point can be achieved. This uneven rise and fall times can be explained with help from the circuit above. During the transition of node X from low to high, the transistors M1, M3 and M6 are on. The signal, SW , will have a very fast rise time, while its complementary signal, \overline{SW} , will have a slower fall time since V_{GS3} eventually shuts off. Thus, a high switch point is achieved.

The amount of current source node disturbance translated to the output also depends on the buffering capabilities between this node and the output. In this DAC, buffering is done with the RZ output scheme. As it was discussed in chapter 3, the switching moments of the data occur when the clock transistor is off, effectively buffering the switch transients seen at the current source node. Therefore, the source node error caused from these switches is reduced by the RZ scheme.

The amplitude of the current source node disturbances is directly related to the capacitance as well. Capacitance is reduced by scaling the current sources and using a smaller cascode transistor to buffer the large capacitance from the current source transistor. However, due to the addition of calibration transistors, the capacitance at this node is approximately the same as before.

Timing Errors

The timing of each data signal's switching moment should line up perfectly to get optimal dynamic performance. Timing errors between switches cause glitches to occur at the output. The worst case glitch will occur at the mid-code transition for an all binary DAC since all switches change state at this point. Timing mismatch at this point can cause the output to potentially rise or fall to the output signal's full-scale value. Unary or

thermometer segmentation can reduce the timing error glitch, but first, the timing of each data signal should be considered.

In this DAC, timing error is minimized through the use of the same sized switch drivers and switches for each current switch network. Therefore, each driver sees the same capacitive load and dynamic timing is the same for each switch in the DAC. For DACs with binary weighted current sources, this is not the case since the current switches are also binary weighted resulting in different capacitive loads.

Re-sampling the data inputs with a latch placed right before the data current switch is a ubiquitous technique used in DAC design. Figure 17 shows the complementary data inputs are re-sampled by latches enabled by the global clock. The two cross coupled inverters ensure equal switching times on both lines.

Another potential timing error can arise from the delay of the thermometer decoding block. To correct this problem, a delay block equivalent to the delay of the thermometer decoder is used for the binary bits to ensure all bits reach the re-sampling latches at the correct sampling period. Figure 12 shows the arrangement of the delay block and the thermometer blocks.

5.2.5 Layout Considerations

DAC layout is an important aspect of design and careful consideration must be taken to reduce layout induced errors. The main layout errors that occur within a DAC are random local errors and deterministic errors such as gradient effects. Local errors can be diminished by the large current sources; deterministic errors can be reduced by using techniques such as geometrical mapping, division of thermometer sources into sub-units and local biasing for each sub-unit [17]. However, with calibration, layout complexity can be simplified since the calibration will work to fix local and deterministic

errors. Thus, there is no need to have a complex geometrical mapping of current sources and local biasing of each unary source subdivision. In other words, this design is layout independent due to the self-correcting nature of the design.

Layout should be implemented to minimize the timing errors, however. Dynamic performance will be closely related to the timing equalization of all data and clock signals. The following layout guidelines should be used for best timing:

- Routing delays of the signals from the sampling latch to the current switch should be minimal and the same for all signals.
- Symmetric clock trees and buffering need to be used with equal loading between branches to minimize clock skew and keep transient waveforms equivalent.

5.3 Calibration Theory, Method, and Algorithm

To reduce the overall area of this DAC and increase its speed, the current sources were scaled down in size to less than the intrinsic matching requirement. By doing this, though, the variation of current between sources will increase and cause amplitude errors that degrade performance. Since this is unacceptable, self-calibration was used to measure and correct the resulting amplitude errors. As it will be shown, the area of the DAC can be drastically decreased with the proposed calibration method even with additional extrinsic calibration circuitry. In this section, the theory and method of this calibration approach will be discussed in detail.

5.3.1 Current Scaling Theory

Matching requirements call for large area current sources, especially for higher and higher resolutions. For the R2R architecture used in this DAC, the size of one unary

current source needs to be sized to roughly $15360 \mu\text{m}^2$ for 12-bit intrinsic accuracy (see Appendix A for derivation). In this DAC, the current sources are scaled down to be 6 bit-intrinsic, adjusting the current source transistors to be $240 \mu\text{m}^2$. This results in a total reduction in theoretical reduction of 64 times. However, since calibration circuitry is necessary, the reduction in area will be less than 64 times. Table 4 shows the total area of the DAC before and after the current scaling reductions are applied. Even with the additional calibration circuitry, which constitutes an estimated $9324 \mu\text{m}^2$ of additional area, the total area savings is still reduced by 24 times.

Table 4. Estimated Area Savings from Current Source Scaling

Accuracy of Current Sources	Total Area of Current Source Transistors	Area Savings
12-bit intrinsic accuracy	0.36864 mm^2	0
6-bit intrinsic accuracy	$5760 \mu\text{m}^2$	64x area reduction
6-bit intrinsic accuracy with additional calibration hardware	$5760 \mu\text{m}^2$ + $9324 \mu\text{m}^2$	24x area reduction

By sizing the current sources down, the relative standard deviation of the current ($\sigma_{\Delta I_D}/I$) will increase. The variable, $\sigma_{\Delta I_D}/I$, can be found based on matching and INL requirements. Equation 2.20 shows the relationship between the INL and $\sigma_{\Delta I_D}/I$. By setting the maximum allowable INL to half of an LSB, the standard deviation can be found by rearranging Equation 2.20 into Equation 2.21.

$$INL_{max} = \sqrt{2^N - 1} \left(\frac{\sigma_{\Delta I_D}}{I} \right) I_{LSB} = \frac{1}{2} I_{LSB} \quad (2.20)$$

$$\frac{\sigma_{\Delta I_D}}{I} = \frac{1}{2\sqrt{2^N - 1}} \quad (2.21)$$

Below is the ratio of the standard deviations due to 6-bit matching and 12-bit matching. It shows that the standard deviation for 6-bit matching will have 8 times the standard deviation of the 12-bit matching case. INL will increase by 8 as well since it is proportional to the standard deviation. Figure 18 shows the normalized distributions for each case.

$$\frac{\sigma_{6\Delta I_D}}{\sigma_{12\Delta I_D}} = \frac{2\sqrt{2^{12} - 1}}{2\sqrt{2^6 - 1}} \approx 8$$

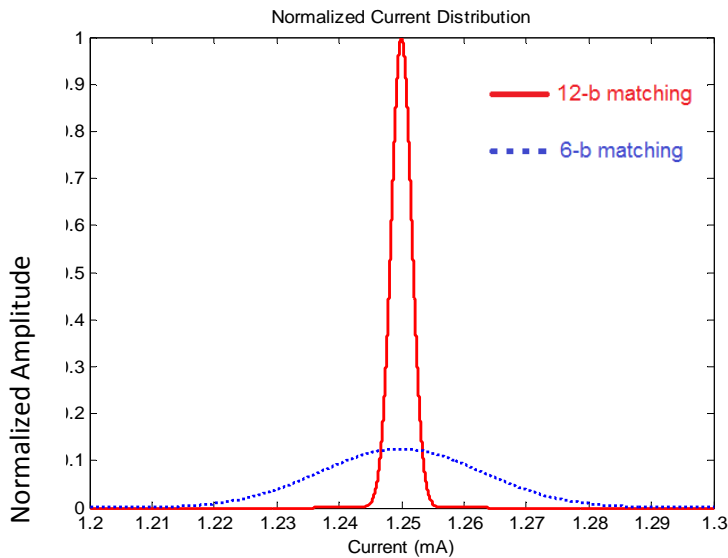


Figure 18. Current Source Distributions for 12-Bit and 6-Bit Matching

To validate the theoretical calculations, Monte Carlo simulations were also performed to get a histogram of the deviations caused by local mismatch and global process variations. In Figure 19, the Monte Carlo simulations show a standard deviation that of 4.57 μA .

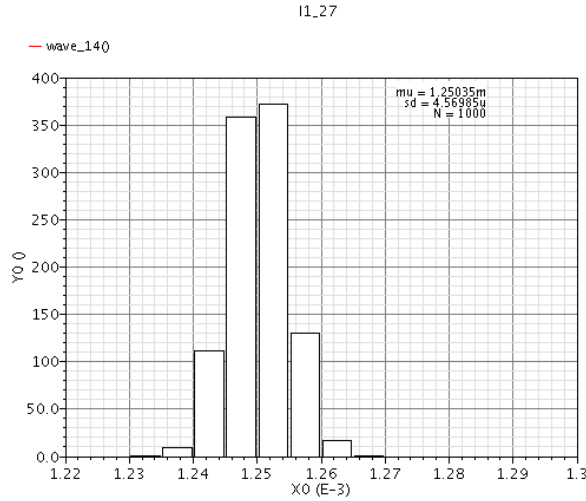


Figure 19. Monte Carlo Current Source Distribution for 6-Bit Intrinsic Matching

The deviation in the current sources will need to be corrected by adding or subtracting it from the nominal current value set by a current reference. The amount of deviation found in Monte Carlo analysis will determine the parameters of calibration circuitry which is discussed in the following section.

5.3.2 Method of Calibration

To correct for the deviations in current, a self-correcting calibration method using similar calibration circuitry in [4] was used. As discussed in chapter 3, the self-correction technique measures and corrects the resulting errors rather than relying on statistical information like DEM techniques. In this way, self-correction techniques guarantee that a correction solution exists. This was the main deciding factor with going with the self-correction calibration approach.

The calibration circuitry shown in Figure 20 consists of a feedback loop that consists of a comparator, a 7-bit successive approximation register (SAR), a 7-bit calibration DAC (CALDAC), and static memory. A digital controller is used to control the calibration circuitry. The proposed calibration method uses a foreground measurement scheme to

measure the amplitude errors of each of the 46 current sources individually (23 current sources in A-DAC and 23 current sources in B-DAC). Essentially, one current is sent to the comparator input one at a time to be measured against a reference current. The difference of these currents, which is the error current, is measured and converted to a digital code through a multi-cycle binary search algorithm that closely resembles a SAR ADC. The generated error codes are then stored and organized in memory. After the calibration of the current sources has completed, the DAC operates in its normal mode. Here, the digital input to the DAC will address the memory to access the proper error code to be loaded into the CALDAC. The CALDAC will then add or subtract the necessary current at the output of the DAC to eliminate the current amplitude errors due to random mismatch.

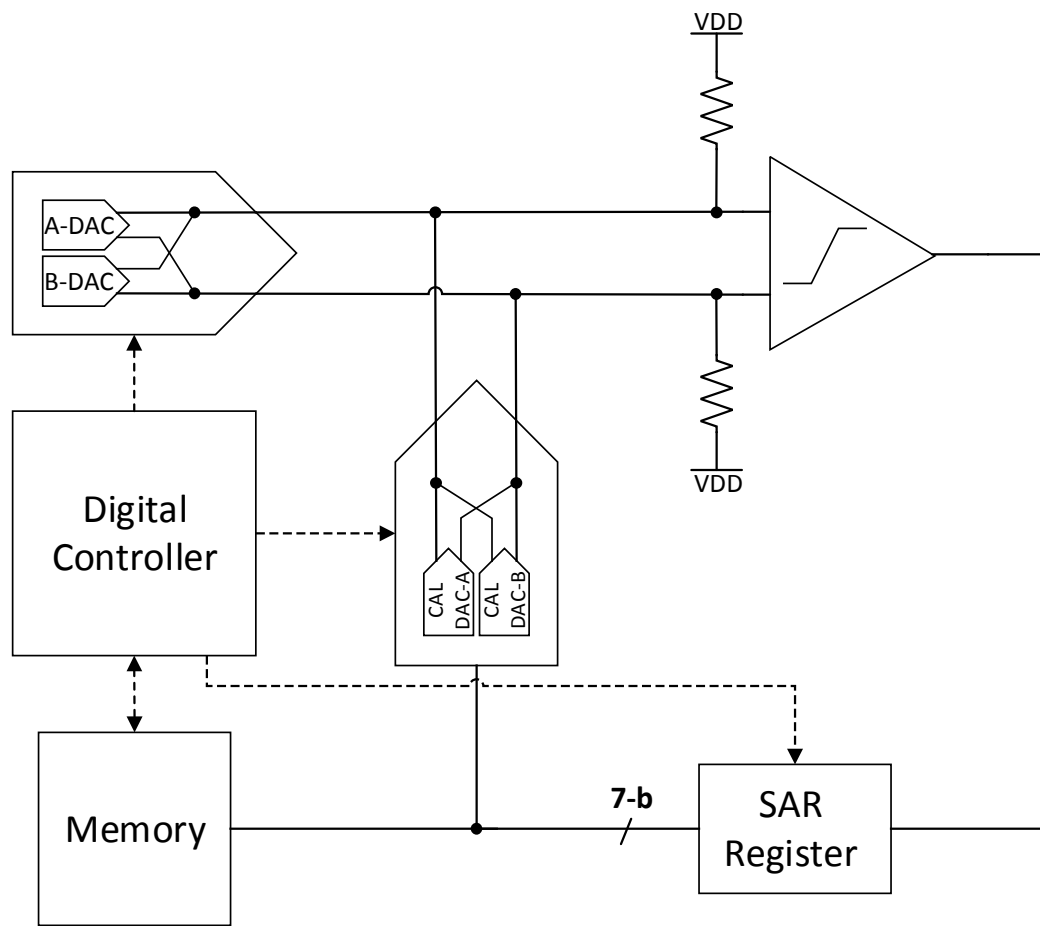


Figure 20. Calibration Loop

The full scale current and resolution of the CALDAC can be determined by the standard deviation of the current source and the precision needed in the correction currents. Monte Carlo simulations showed that the standard deviation of a current source was approximately $4.27\mu\text{A}$ (Figure 19). To ensure that the CALDAC covers a 6σ range, it must have a swing of $\pm 6 \cdot 4.27\mu\text{A}$ or $\pm 25.62\mu\text{A}$. The LSB current of the CALDAC was chosen to be 0.125 LSB of the main DAC making the required resolution of the CALDAC 7-bits.

5.3.3 Calibration Algorithm

The calibration algorithm can be broken up into 3 different stages: calibration of the A-DAC current sources, calibration of the B-DAC current sources, and normal operation mode. The state diagram in Figure 21 shows the transition of the algorithm through each of these three stages. During the foreground calibration phases, the DC error of each current source in A-DAC and B-DAC is measured one-by-one, organized and stored in memory. Then the DAC will transition into its normal operation mode. Note that during calibration of the A-DAC only one current source is sent to the comparator to be measured against the reference current. All of the remaining current sources are steered away that are not being measured at that moment.

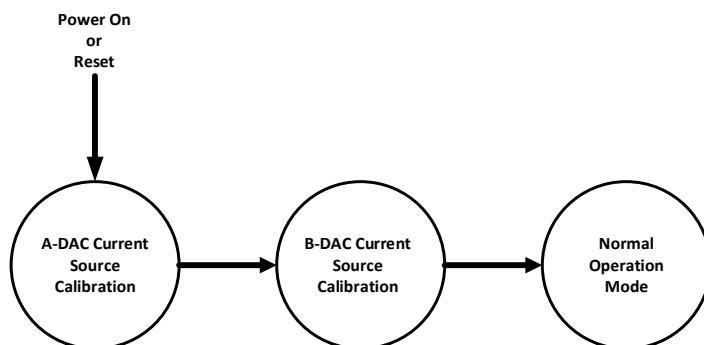


Figure 21. Stages of Calibration Algorithm

A more detailed state flow diagram of the calibration algorithm can be seen in Figure 22. In this state flow diagram, $I_{k,i}$ denotes the k^{th} current source in the i^{th} DAC, where $i=1$ corresponds to A-DAC and $i=2$ corresponds to B-DAC. The calibration algorithm begins when the circuit is initially powered on or reset and a signal to start calibration is asserted. Once this occurs, the calibration algorithm selects the 1st current source from A-DAC ($k=1, i=1$) and sends it to be compared to the reference current source. The difference between the two sources is digitized through a 7-cycle binary search iteration. The cumulative error of all current sources (i.e. $k=1$ through $k=16$) is stored in memory at address k in the manner shown in Equation 2.22.

$$e(k) = e(k) + e(k - 1) \quad (2.22)$$

When all current sources have been measured and stored appropriately in memory (i.e. when $k=16$), the error codes are manipulated once more in order to be used in a complementary DAC format. The variable i is then incremented and the same calibration procedure continues for B-DAC. When B-DAC is finished calibrating, i increments to 3 and normal operation ensues. At this point the DAC is finished with the foreground calibration and emulates the normal function of a DAC.

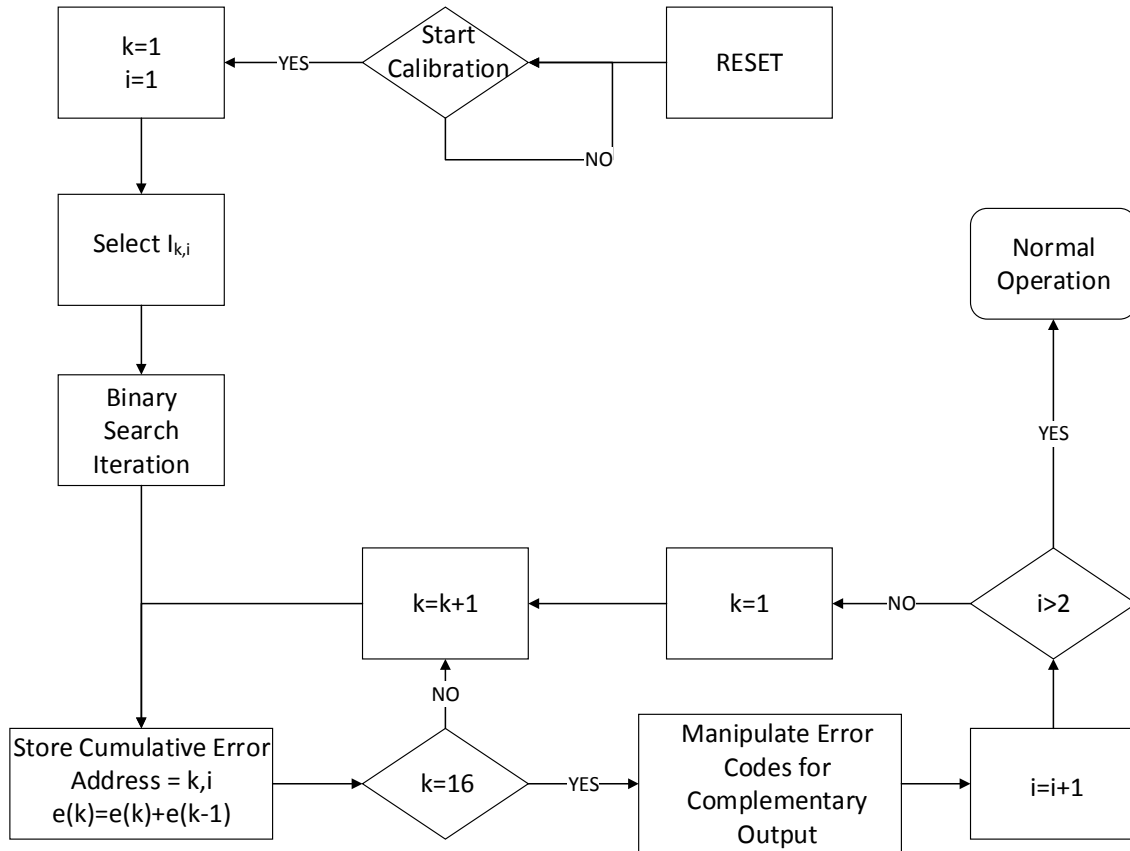


Figure 22. Calibration State Flow Diagram

The binary search iteration is similar to a SAR ADC. The iteration involves a feedback loop that includes the comparator, SAR, and CALDAC. Initially, when the algorithm selects a new current source to be compared to the reference, the input code for the CALDAC is set at mid-code (i.e. 1000000). At mid-code the amount of current from the CALDAC on the positive and negative output lines is equivalent; this means that the differential output is zero. While set to mid-code, the current source $I_{k,i}$ is compared to I_{REF} . If $I_{k,i}$ is greater than I_{REF} the code is changed to 01111111, and the CALDAC subtracts current from the output. If $I_{k,i}$ was less than I_{REF} the code would remain at 1000000. The binary search algorithm continues for 7-cycles and when

finished a 7-bit error code that resembles the analog difference between the $I_{k,i}$ and I_{REF} is found.

An example of how the digital error code is represented is explained below. In this example we assume that the calibration current is less than the reference current by 4 LSBs. Since the LSB of the CALDAC is an eighth of an LSB, the code in offset binary is represented as 1100000.

$$1100000 \rightarrow 32 \cdot \frac{LSB}{8} = 4 \text{ LSB}$$

The formula above shows that the digital representation represents a positive 4 LSBs, which is the necessary amount of current to be added to the current source to be equivalent to the reference current source.

Initially, the error codes are arranged in the memory to be used in a single-ended DAC. However, the error codes stored in memory will need to be manipulated to accommodate a complementary output as mentioned previously. The reasoning behind this can be explained through another example. Suppose during normal operation of the DAC, that one of the MSB currents is on and the other 14 are off. For a complementary DAC, the current of one of the MSB currents is supplied to the positive output line and the other 14 MSB currents are supplied to the negative output line. Based on this fact, the correct error code for a complementary output will have to include the error of the MSB current going to the positive output subtracted by the cumulative error of the 14 MSB currents going to the negative output. Table 5 shows how the error codes would be rearranged for a 2-bit thermometer encoded DAC for both single-ended and complementary outputs.

Table 5. Error Arrangements for Single-Ended and Complementary DACs

Address (k)	Error Arrangement, $e(k)$, for Single-Ended DAC	Error Arrangement, $e(k)$, for Complementary DAC
1	$e(1)$	$e(1)-e(2)-e(3)$
2	$e(1)+e(2)$	$e(1)+e(2)-e(3)$
3	$e(1)+e(2)+e(3)$	$e(1)+e(2)+e(3)$

The translation of the error codes from single-ended to complementary output modes is done through the additional hardware seen in Figure 23. Taking the 2-bit thermometer example, the memory will initially store error codes in a digital format that corresponds to the second column in Table 5 and the “cumulative error” register stores the contents of address 3. First, the digital controller reads the memory starting at address 3. The code gets translated to a complementary format with a couple of additions and is written back to address 3 during the next cycle. Next, the digital controller reads the memory at address 2, and the same process is done as before until all addresses have been read and written to. Figure 23 shows the error codes at each node during the read and write phases when the address is equal to 2.

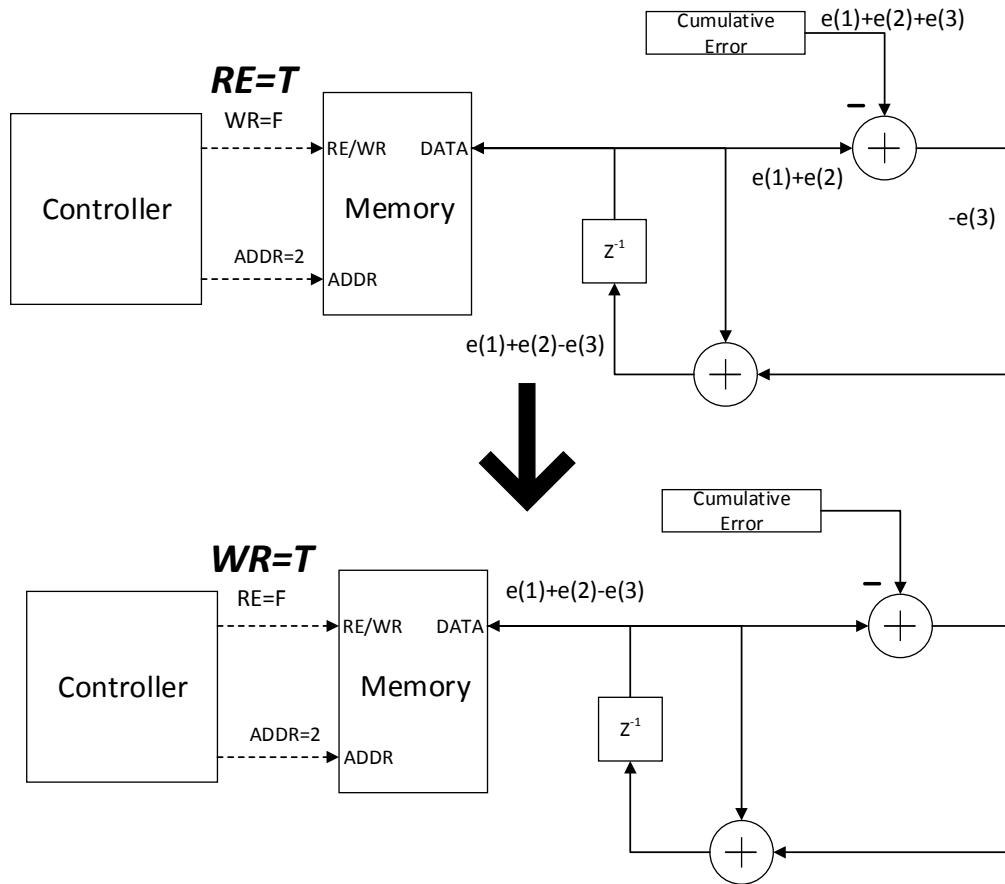


Figure 23. Hardware to Manipulate Error Codes from Unipolar to Bipolar Format

5.4 Design of Calibration Circuitry

The calibration circuitry includes the comparator, CALDAC, SAR, memory and digital control block. In the following section, each component's design is described in more detail.

5.4.1 Comparator

The comparator works to measure the difference between the DAC thermometer and reference currents. Design of this comparator needs have high sensitivity, low offset and a high input common mode range to ensure that proper error codes are generated

during the calibration phase. The speed requirement of the comparator is less critical to the design assuming that foreground calibration can be clocked at slow enough speeds.

For the high sensitivity requirement, the comparator needs to be able to sense a minimum signal difference equivalent to the step size of the CALDAC. The CALDAC's LSB current is equivalent to one-eighth of an LSB of the main DAC making the minimum step size of the CALDAC equivalent to one-fourth of an LSB or $30.52 \mu\text{V}$. The offset should also be as small as possible to generate the correct codes. If the offset is too large, it will directly translate into the error codes and the output current of the DAC. The comparator also needs to work properly for an input common mode range of 1.3V to 1.8V.

The voltage comparator used in this thesis is shown in Figure 24. It uses a 3-stage configuration consisting of a pre-amplifier, decision circuit, and output buffer, respectively. The output buffer is followed by a series of 3 inverters to boost the signal to either the supply or ground rails. The pre-amplification stage uses a NMOS differential amplifier with PMOS active loads and can detect a high input common mode range up to the supply voltage. The decision stage uses a positive feedback configuration that can be designed to add hysteresis to the comparator. However, in this design zero hysteresis was added. The output buffer was designed as a PMOS differential amplifier. This stage increases the total gain and shifts the common mode output to a level acceptable for the chain of inverters to properly distinguish. The bias of the pre-amplifier and output buffer were generated by the bias circuit seen in Figure 25.

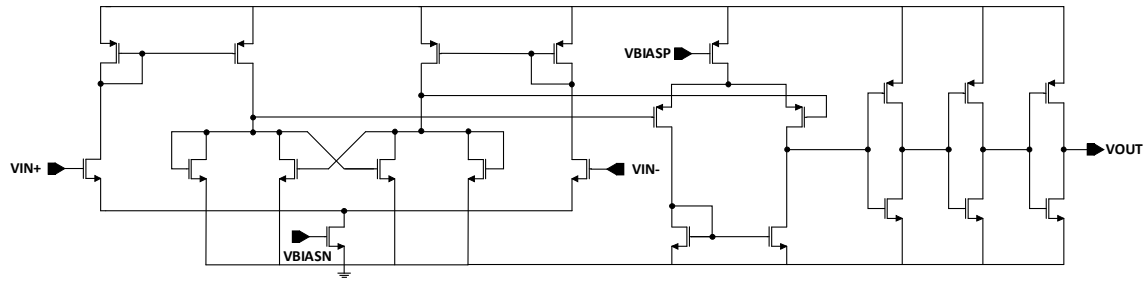


Figure 24. Comparator Schematic

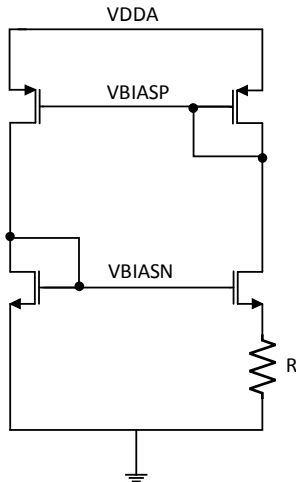


Figure 25. Bias Generator for the Comparator

5.4.2 Calibration DAC (CALDAC)

The CALDAC converts the error codes into current to be added to the main DAC's output during normal operation. It consists of two, 7-bit binary current steering DAC's as shown in Figure 26 below. The current switch design network is the same design as in Figure 11 except for the binary weighting of the transistor sizes. The LSB unit current is equal to 610.35 nA.

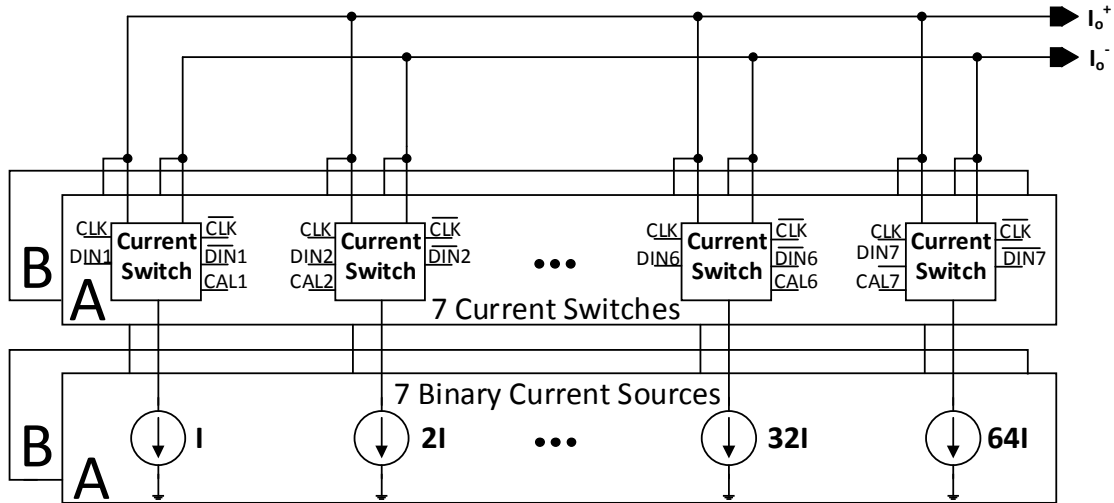


Figure 26. CALDAC Architecture

The performance of the whole system depends on the performance of the DAC and CALDAC working properly together. To ensure that the intrinsic DAC performance is not degraded by the CALDAC, 7-bit linearity is necessary. The CALDAC needs to work at high update rates that will be used for the main DAC. A sample spectrum of the CALDAC is shown in Figure 27. The spectrum shows approximately 7-bit linearity for an input frequency of 29.99 MHz and an update rate of 400MSPS.

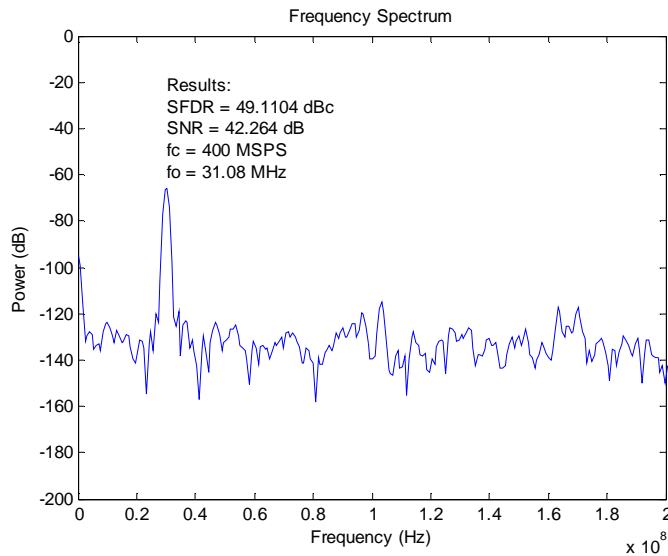


Figure 27. Frequency Spectrum of the CALDAC

The CALDAC's outputs need to line up exactly with the main DAC's outputs otherwise distortion will occur. Data drivers similar to the ones used in the main DAC are used to ensure that the outputs are aligned. The alignment was also verified in simulations.

5.4.3 Successive Approximation Register (SAR)

The SAR is a critical component in the calibration loop during the binary search iteration. Its main function is to update the code to the CALDAC during each cycle of the binary search iteration based on the digital output from the comparator.

Figure 28 shows an example of a 3-bit SAR. The inputs of the SAR include the clock (CLK), the comparator output (COMP) and reset (RST); the outputs include the 3-bits, B2, B1, and B0. During the first cycle of the clock, the SAR is reset and the output updates to mid-code or 100. The comparator then compares the inputs at its terminals and sends the output to the SAR. During the rising edge of the clock (beginning of the second cycle), the SAR will decide how to update its code based on the output from the comparator (COMP). For example, if the comparator output was a digital 1, the SAR would update to 110. If the comparator output was a 0, the SAR would update to 010. The SAR continues to work in this fashion for N-cycles, where N is the number of output bits. In the thesis design, a 7-bit SAR was implemented.

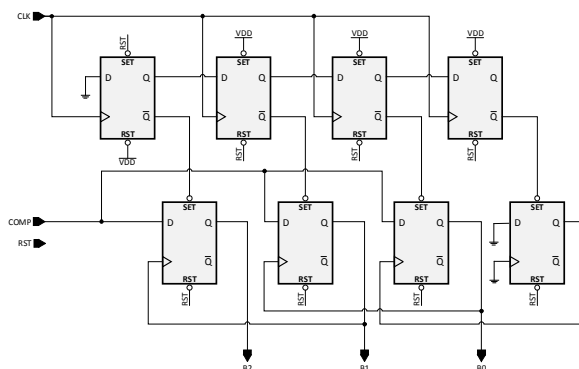


Figure 28. Example of a 3-Bit SAR

5.4.4 Memory

The memory is responsible for storing the error codes generated during the calibration phase. The size of the memory is dependent on the amount of thermometer current sources in the DAC. For this design, only the MSB current sources were calibrated. Since each DAC has 15 MSB current sources the memory needs to store $(15+1) \times 2$ or 32 error codes that are 7-bits long. Thus, there needs to be 32 addresses each able to hold 7-bits of information. In this design, the memory was divided into two memory banks, one for the A-DAC and one for the B-DAC. It should be noted that if LSB thermometer sources were also calibrated each of the memory banks would need to include 8 more addresses.

Figure 29 shows one memory bank black box. The memory has a read/write enable port and a 3-bit address port for its inputs. The data port is used to import and export data into and out of memory. Inside the memory bank black box there will be an array of 16×7 unit cells. The unit cell structure can be seen in Figure 30. Design of the unit cell consists of two tri-state inverters, an inverter and a transmission gate for a total of 12 transistors per unit cell. Since memory design was not a priority in this design the 12-transistor unit cell was chosen for its ease of implementation and robustness.

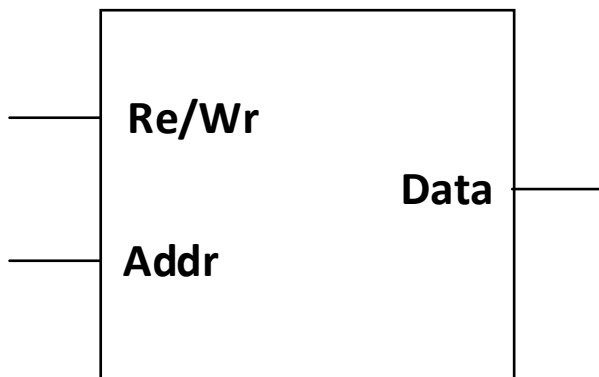


Figure 29. Memory Black Box

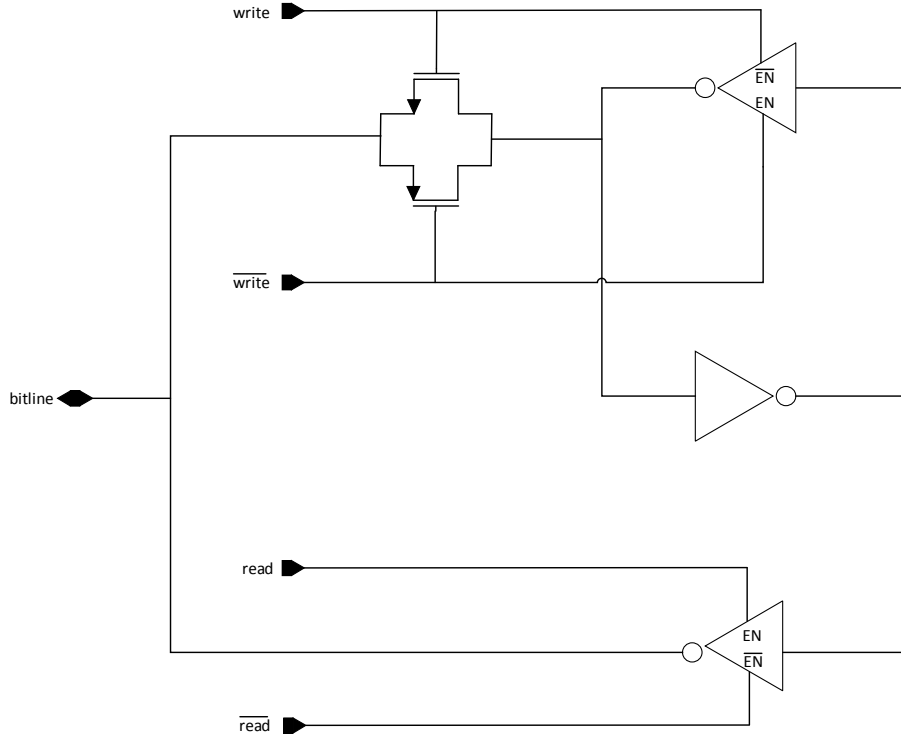


Figure 30. Memory Bit Cell

5.4.5 Digital Control Circuitry

The digital control block was the main processor that controlled the flow of the calibration. The block was implemented with DFFs and various logic gates using Mealy and Moore methods. The algorithm/ signal flow diagram can be seen in Figure 22.

5.4 Simulation Results and Analysis

In this thesis, the DAC's analog output data was acquired through simulations in Cadence's Virtuoso environment and then sent to MATLAB for FFT analysis to obtain SNR and SFDR results. Simulations were done with and without calibration over multiple input frequencies and update rates to prove that the calibration fixes the distortion due to random mismatch errors in the current sources. Because the results were not obtained from a physical chip, the mismatch of current sources was modeled in

the design. The mismatch of the current sources or error currents were modeled by individual ideal current sources placed in parallel to each of the 46 current source transistors in the main DAC. The error current amplitudes were chosen based on a zero mean normal Gaussian distribution based on the Monte Carlo plot in Figure 19.

The plot in Figure 31 shows the SFDR results before and after calibration for multiple input frequencies at an update rate of 200 MSPS. The results show that SFDR is improved 19-21 dB across the frequency range with the added calibration. Without calibration, this DAC's performance would underperform and would not be useful for 12-bit linearity. Thus, it is necessary to include calibration when the current sources are scaled down below the size of matching requirements. The data points that were used in the plot below can be seen in Table 6. The table also shows measurements taken at update rates of 100 MSPS and 400 MSPS for various input frequencies. Fields with a dash through them signify that the simulation was not performed.

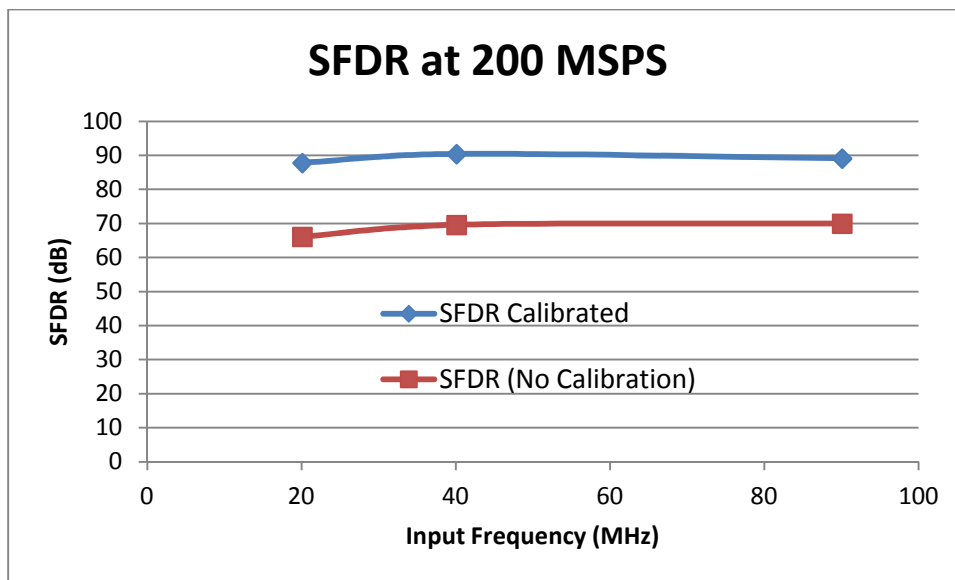


Figure 31. SFDR Results before and after Calibration for a 200 MSPS Update Rate

Table 6. SFDR and SNR Results before and after Calibration

		Before Calibration		After Calibration	
FOUT (MSPS)	FIN (MHz)	SFDR	SNR	SFDR	SNR
100	20.111	69.3	65.7	90.9	77.07
	40.111	-	-	-	-
200	20.111	66.1	61.7	87.8	73.9
	40.111	69.6	67.8	90.4	75.6
	90.111	69.98	64.2	89.1	76.3
400	20.111	-	-	-	-
	40.111	-	-	75.9	69.5
	120.111	-	-	-	-
	190.111	-	-	-	-

The frequency spectrums of the DAC offer some insight into what is happening before and after calibration. In Figure 32, a comparison of two DAC output spectrums before and after calibration with a 20MHz input and 100 MSPS update rate is shown. The spectrum on the left side is clearly the un-calibrated spectrum and exhibits a noisy spectrum with spurious tones throughout. When the DAC is calibrated, most of the harmonics are suppressed below the noise floor which drastically improves the SFDR and SNR.

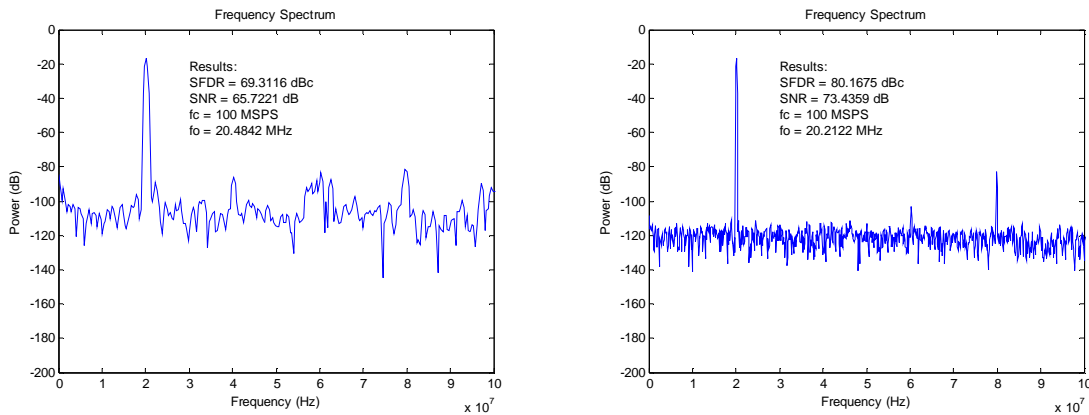


Figure 32. Comparison of Spectrums (a) before Calibration (Left) and (b) after Calibration (Right)

Temperature and corner simulations were done to verify the DAC works over a broad range. For these measurements, a 90 MHz input frequency with a 200 MSPS update rate was used. Table 7 tabulates the SFDR results for temperatures ranging from -40°C to 125°C at the nominal corner and Table 8 shows the corner results of this DAC for nominal, fast, and slow corners at a temperature of 27°C. The SFDR results remain over 80 dB for all of these measurements except for the fast corner which shows a SFDR of 76.7 dB. Even though the fast corner is lower compared to the rest of the SFDR numbers in this DAC, its performance is still comparable to 12-bit DACs in literature which commonly showcase DACs that have 71 dB or more of SFDR [4].

Table 7. SFDR Results for Various Temperatures at the Nominal Corner

Temperature	SFDR
-40	80.8
27	89.1
125	87.1

Table 8. SFDR Results for Various Corner Simulations with T=27°C

Corner	SFDR
Nominal	89.1
Fast	76.7
Slow	90.3

This section showed that the DAC designed in this thesis is capable of running at high speeds with high linearity. We saw that update rates of 400 MSPS were possible with an SFDR of 75.9 dB. The results in this DAC are comparable to state-of-the-art DACs documented in literature today. Similar DACs show maximum SFDR numbers of 71 dB on up for sampling rates in a similar range [4].

REFERENCES

- [1] K. Kavanagh. (1995-2014). *Open-Loop Calibration Techniques for Digital-to-Analog Converters*. [Online]. Available: <http://www.analog.com>
- [2] W. Kester, D. Sheingold, "Testing Data Converters," in *Data Conversion Handbook*, 1st ed. Analog Devices, 2004, ch. 5, sec. 5.1, pp. 5.1-5.25.
- [3] Y. Cong and R. Geiger, "A 1.5-V 14-Bit 100-MS/s Self-Calibrated DAC," *IEEE J. Solid State Circuits*, vol. 38, no. 12, pp. 2051-2060, Dec. 2003.
- [4] M.-H. Shen and P.-C. Huang, "A Low Cost Calibrated DAC for High Resolution Video Display System," *IEEE Trans. VLSI Syst.*, vol. 20, no. 9, Sept. 2012.
- [5] W. Kester, D. Sheingold, J. Bryant, "Fundamentals of Sampled Data Systems," in *Data Conversion Handbook*, 1st ed. Analog Devices, 2004, ch. 3.
- [6] H. Zumbahlen, "Converters," in *Linear Circuit Design Handbook*, 1st ed. Analog Devices, 2008, ch. 6, sec. 6.1.
- [7] G.I. Radulov, "Flexible and Self-Calibrating Current Steering Digital-to-Analog Converters: Analysis, Classification and Design," Ph.D. dissertation, Dept. Elec. Eng., Eindhoven University of Technology, Eindhoven, Netherlands, 2010.
- [8] A. R. Bugeja, B. S. Song, P. L. Rakers, and S. F. Gillig, "A 14-b, 100-MS/s CMOS DAC Designed for Spectral Performance," *Solid-State Circuits, IEEE Journal of*, vol. 34, pp. 1719-1732, 1999.
- [9] C. Myung-Jun, B. Kwang-Hyun, and M. Teshome, "A 1.6-GS/s 12 Bit Return-to-Zero GaAs RF DAC for Multiple Nyquist Operation," *Solid-State Circuits, IEEE Journal of*, vol. 40, pp. 2456-2468, 2005.
- [10] M.-H. Shen, J.-H. Tsai, and P.-C. Huang, "Random Swapping Dynamic Element Matching Technique for Glitch Energy Minimization in Current-Steering DAC," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 5, pp. 369-373, May 2010.
- [11] G. R. Spalding and R. L. Geiger, "Digital Correction for Improved Spectral Response in Signal Generation Systems," *Circuits and Systems, 1993, ISCAS '93, 1993 IEEE International Symposium on*, 1993, pp. 132-135.
- [12] T. Chen and G. G. E Gielen, "A 14-bit 200-MHz current-steering DAC with switching-sequence post-adjustment calibration," *IEEE J. Solid-State Circuits*, vol.42, no. 11, pp. 2386-2349, Nov. 2007.

- [13] Y. Tang, H. Hegt, A. van Roermund, "DDL-based calibration techniques for timing errors in current-steering DACs," *ISCAS 2006*, pp. 101-104.
- [14] Y. Tang, J. Briaire, K. Doris, R. van Veldhoven, P. van Beek, H. Hegt, A. van Roermund, "A 14-bit 200 MS/s DAC with SFDR >78 dBc, IM₃ < -83 dBc and NSD < -163 dBm/Hz across the whole Nyquist band enabled by dynamic-mismatch mapping," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1371-1381, June 2011.
- [15] S. Luschas and H.S. Lee, "Output impedance requirements for DAC's," in *Proc. IEEE 2003 ISCAS*, May 2003, pp. I-861-I-864.
- [16] J. Wikner, N. Tan, "Modeling of CMOS Digital-to-Analog Converters for Telecommunication," *IEEE Transactions on Circuits and Systems*, vol. 46, No. 5 May 1999.
- [17] K. Doris, A. van Roermund, D. Leenaerts, *Wide-Bandwidth High Dynamic Range D/A Converters*, Springer, 2006.
- [18] Luo Fangjie, Yin Yongsheng, Liang Shangquan, Gao Minglun, "Current switch driver and current source designs for high-speed current-steering DAC," *IEEE.ASID2008.2rd International conference*, 364-367, 20-23 Aug. 2008.
- [19] P. Sungkyung, K. Gyudong, P. Sin-Chong, and K. Wonchan, "A digital-to-analog converter based on differential-quad switching," *Solid-State Circuits, IEEE Journal of*, vol. 37, pp. 1335-1338, 2002.
- [20] J. Deveugele, P. Palmers, and M. S. J. Steyaert, "Parallel-path digital-to-analog converters for Nyquist signal generation," *IEEE J. Solid-State Circuits*, vol. 39, no. 7, pp. 1073-1082, Jul. 2004.

APPENDIX A
CALCULATION OF CURRENT SOURCE AREA

The area of the unit current source transistor in the DAC can be modeled using the equation below. Here, A_β and A_{V_T} are process parameters; $\frac{\sigma_{\Delta I_D}^2}{I^2}$ is the relative variation in drain current of a unit current source; $(V_{GS} - V_T)^2$ is commonly known as the overdrive voltage; and WL is the area of one unit current source transistor.

$$WL = \frac{1}{\frac{\sigma_{\Delta I_D}^2}{I^2}} \left[A_\beta + \frac{4A_{V_T}}{(V_{GS} - V_T)^2} \right] \quad (2.23)$$

The area can be found by substituting the parameters into the equation above. The parameters A_β and A_{V_T} are given by the process, the overdrive voltage is set by the designer, and the relative standard deviation can be found based on the resolution (N) and INL or DNL requirement of the DAC. Equation 2.24 shows the relation between INL, the resolution and the relative standard deviation. If the maximum INL is set to half an LSB as shown then Equation 2.24 can be rearranged to that of Equation 2.25.

$$INL_{max} = \sqrt{2^N} \left(\frac{\sigma_{\Delta I_D}}{I} \right) I_{LSB} = \frac{1}{2} I_{LSB} \quad (2.24)$$

$$\frac{\sigma_{\Delta I_D}}{I} = \frac{1}{2\sqrt{2^N - 1}} \quad (2.25)$$

Now that all parameters are defined, they can be plugged into Equation 2.23 to find the required area of each unit current source. Table 7 shows how the size requirement for the unit current source varies for a constant overdrive voltage of 300 mV at different resolutions. Note that the relative standard deviation in current was calculated using a 6 sigma confidence rather than 1 σ confidence. This is because 1 σ

deviation only accounts for 68.27% of the population while 6σ includes almost 100% of the population. To represent this, Equation 2.25 is changed to the Equation 2.26 below:

$$\frac{\sigma_{\Delta I_D}}{I} = \frac{1}{6 \times 2\sqrt{2^N - 1}} \quad (2.26)$$

Table 9. Area of Unit Current Sources for Different Resolutions and Constant Overdrive

Resolution	Relative standard deviation of drain current with 6 sigma confidence (%)	Area of unit current source (μm^2)
6	1.042	0.928
8	0.5208	3.75
12	0.1302	60

By using the information from Table 7, the sizing of the current sources can be found. In the subsequent paragraph, the method on how to size the MSB current sources for 6 and 12-bit matching is explained.

For the architecture in this 12-bit DAC, the MSB current sources represent the uppermost 4-bits of the DAC. Since they are unary weighted, the weighting of all 15 of the MSB current sources will be 2^9 or 256 times the LSB current. This means that the area of each MSB source will also need to be 256 times the unit current source area shown in Table 7. Therefore, the area of each MSB needs to be greater than $0.928\mu\text{m}^2 \times 256$ or $237.568 \mu\text{m}^2$ for 6-bit matching. In this DAC an area of $240 \mu\text{m}^2$ was used. If the designer chose to keep the area of each MSB current source 12-bit accurate then the size of the MSB would be $256 \times 60\mu\text{m}^2$ or $15360\mu\text{m}^2$.