

UCLA

UCLA Electronic Theses and Dissertations

Title

Multi-Step Charge-Trap Transistor-Based Calibration of High Speed and High Resolution Digital-to-Analog Converters

Permalink

<https://escholarship.org/uc/item/5f99s8m5>

ISBN

9798190937374

Author

Lee, Jun Sang

Publication Date

2026-09-11

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Multi-Step Charge-Trap Transistor-Based Calibration of High Speed and High Resolution
Digital-to-Analog Converters

A thesis submitted in partial satisfaction of the
requirements for the degree

Master of Science in Electrical & Computer Engineering

by

Jun Sang Lee

2026

© Copyright by

Jun Sang Lee

2026

ABSTRACT OF THE THESIS

Multi-Step Charge-Trap Transistor-Based Calibration of High Speed and High Resolution Digital-to-Analog Converters

by

Jun Sang Lee

Master of Science in Electrical & Computer Engineering

University of California, Los Angeles, 2026

Professor Sudhakar Pamarti, Chair

High speed and high resolution digital-to-analog converters (DACs) are highly sought after for modern communication systems. To maintain high speed performance and high resolution accuracy, calibration is used to correct for mismatches. However, conventional schemes interact with the DAC core and signal path, reducing high speed performance. We propose a calibration structure using charge-trap transistors (CTTs) to isolate calibration circuitry from the DAC signal path. To minimize calibration hardware area and complexity, a method of calculating the minimum number of CTTs needed is presented as well. A DAC with the proposed calibration has been designed and simulated with a Nyquist-rate SFDR of 67.3dB and a Nyquist-rate IM3 of 73.4dB.

The thesis of Jun Sang Lee is approved.

Hooman Darabi

Subramanian S. Iyer

Sudhakar Pamarti, Committee Chair

University of California, Los Angeles

2026

Contents

ABSTRACT OF THE THESIS	ii
LIST OF FIGURES	vi
LIST OF TABLES.....	viii
LIST OF ABBREVIATIONS.....	ix
ACKNOWLEDGEMENTS	x
INTRODUCTION	1
CHAPTER 1 Background on Current Steering DACs	2
1.1 Architecture Choice.....	2
1.2 Static Nonlinearities	5
1.2.1 Transistor Mismatch and Sizing	5
1.2.2 Finite Output Resistance.....	6
1.2.3 Nonmonotonicity and Segmentation.....	9
1.3 Dynamic Nonlinearities	12
1.3.1 Switching Glitches.....	13
1.3.2 Randomization.....	14
1.3.3 High Frequency Output Impedance.....	15
CHAPTER 2 Background on DAC Calibration	18
2.1 Conventional Calibration.....	20
2.2 Contribution of this Thesis	23
CHAPTER 3 CTT-based DAC Calibration	24
3.1 CTT Background and Model.....	24
3.2 CTT Calibration Structure	27

3.3 Temperature Tracking	35
CHAPTER 4 Design of a CTT-based 12-bit DAC.....	37
4.1 DAC Core Design.....	37
4.1.1 Current Source Design	38
4.1.2 Switching Pair Design	38
4.1.3 Cascode and Supply Design	39
4.1.4 Segmentation and Keep-Alive Current Choice	40
4.1.5 DAC Core Design Summary	42
4.2 CTT-based Calibration Design	44
4.3 Digital and Calibration Circuitry Design	48
CHAPTER 5 Results and Conclusions	50
5.1 Simulation Results.....	50
5.2 Conclusion and Future Work.....	56
REFERENCES	58

LIST OF FIGURES

Figure 1.1: Single ended current mode DAC cell. (a) Ideal model. (b) Transistor level implementation.....	3
Figure 1.2: Transistor level implementation of a differential current steering DAC cell.	4
Figure 1.3: Representation of current source output resistance in parallel with a resistive load....	6
Figure 1.4: Cascoded current steering DAC cell with a resistive load.	8
Figure 1.5: Nonmonotonicity at the midpoint transition. (a) Numerical representation. (b) Effect on the output ramp.	9
Figure 1.6: Fully thermometer coded DAC circuit.....	10
Figure 1.7: Segmented 6-bit DAC circuit with 4 LSBs of binary weighting and 2 MSBs of thermometer coding.	11
Figure 1.8: DAC output transition. (a) Ideal transition. (b) Transition with switching glitch.	13
Figure 1.9: Output frequency spectrum of a DAC. (a) Without randomization. (b) With randomization.....	15
Figure 1.10: DAC cell with keep-alive currents. (a) ideal current source version. (b) circuit implementation.....	17
Figure 1.11: DAC with one side on, parasitic capacitances being canceled out differentially through keep-alive currents.....	17
Figure 2.1: Comparison of timing mismatch with 2 DAC core sizes.	19
Figure 2.2: Conventional subDAC/CALDAC-based digital calibration loop.	20
Figure 2.3: Conventional subDAC/CALDAC implementation.	21

Figure 3.1: Circuit-based CTT model.	24
Figure 3.2: (a) Family of programmed curves for different v_{th} shifts. (b) INL of programmed current values for a chosen bias point.	25
Figure 3.3: CTT-based DAC design and calibration circuitry in [6].	27
Figure 3.4: CTT source follower driver for a DAC current source.	28
Figure 3.5: Number of CTTs needed per bit slice.	33
Figure 3.6: Multi-step CTT calibration procedure.	34
Figure 3.7: INL measurements of CTT curves at different temperatures.	35
Figure 4.1: Implemented 12-bit DAC design.	37
Figure 4.2: Dynamic performance of the 12-bit DAC design at 20GS/s for different segmentation ratios and active power dissipation.	42
Figure 4.3: DNL spike above 0.5LSBs for initial calibration scheme.	46
Figure 5.1: INL and DNL plots of the DAC design for 200 Monte Carlo points.	51
Figure 5.2: One outlier from 200 point Monte Carlo simulation.	52
Figure 5.3: SAR convergence of calibration for the 23 rd bit.	54
Figure 5.4: Frequency spectrum plots of the DAC design at 20GS/s. a) single tone 9GHz input, SFDR = 67.3dB. b) two tone 9GHz and 9.1GHz input, IM3 = 73.4dB.	55

LIST OF TABLES

Table 4.1: Bias voltage values for $M4$ and $M5$ transistors in Figure 4.1 and corresponding DAC V_{DD} values	40
Table 4.2: Sizes of transistors in Figure 4.1.	43
Table 4.3: Bias values for DAC design in Figure 4.1.	43
Table 4.4: Initial CTT calibration parameters for DAC design in Figure 4.1.	44
Table 4.5: Device statistics and CTT calibration parameters.	44
Table 4.6: DAC design number of CTTs per set for each bit slice.	45
Table 4.7: Revised number of CTTs per set	48
Table 5.1: Calibrated v_{th} shift values for the outlier run	53

LIST OF ABBREVIATIONS

DAC	Digital-to-Analog Converter
LSB	Least Significant Bit
MSB	Most Significant Bit
INL	Integral Nonlinearity
DNL	Differential Nonlinearity
SFDR	Spurious Free Dynamic Range
IM3	3 rd Order Intermodulation
CAL	Calibration
SAR	Successive Approximation Register
CTT	Charge-Trap Transistor
SRAM	Static Random Access Memory

ACKNOWLEDGEMENTS

I would like to give great thanks to my advisor, Professor Sudhakar Pamarti, for his patience in supporting me in my work and providing immense insightful guidance throughout my time at UCLA. Additionally, I would like to thank my committee members Professor Hooman Darabi and Professor Subramanian Iyer for their time. I would also like to thank Vinod Kurian Jacob for his valuable experience in circuit design and helping me understand all the details needed for data conversion design. I am also grateful to Samyak Chakrabarty, Mohammadreza Zeinali, and Samuel Wang for various assistance and feedback in my work.

I would like to thank the Defense Advanced Research Projects Agency (DARPA) for supporting this work through the Optimum Processing Technology Inside Memory Arrays (OPTIMA) Program under Agreement HR0011-24-9-0304.

INTRODUCTION

High speed and high resolution data converters are in high demand for modern wireline and wireless communication systems. Digital-to-analog converters in particular can be used as a driver for high data rate wireline transmitters or wireless power amplifiers. At high resolutions, maintaining analog precision is difficult, and at high speeds, dynamic errors become more problematic. Calibration offers a solution that can potentially handle both issues. Digital subDAC-based calibration has proved successful [3] [4], and other calibration techniques and algorithms have been proposed to guarantee higher precision [10] and retain accuracy in different environments [11]. While proven to be reliable, digital calibration techniques need connections to internal nodes of the DAC, which ultimately slows them down by adding parasitic capacitances, as well as increases their size. Simpler and smaller circuits have less parasitic capacitances, making them more favorable to operate at high speed. The downside of that is smaller and simpler circuits are more prone to high error from process mismatch. This thesis addresses this issue by proposing a calibration structure using charge trap transistors that minimizes the connections to internal nodes. In addition, to reduce calibration time and wiring congestion, a method to minimize the number of CTTs is proposed as well to reduce complexity.

The outline of this thesis is as follows. Chapter 1 will give a background on DAC design challenges and common solutions to mitigate error. Chapter 2 will give a background on DAC calibration and the conventional techniques used. Chapter 3 presents a model of the CTT and proposes a structure using them, as well as analysis of device characteristics to minimize the number of CTTs needed. Chapter 4 goes through the circuit implementation of a 12-bit DAC design with CTT-based calibration in Global Foundries 22nm technology. Chapter 5 presents the simulation results of the calibration and conclusions from this work.

CHAPTER 1

Background on Current Steering DACs

The architecture of choice in prior art for high speed and high resolution DACs is the current steering architecture [8][9][10][12]. This chapter discusses the basics of the current steering DAC architecture and the sources of static and dynamic nonlinearity it faces. Then, we highlight various linearization techniques from prior literature to tackle these issues.

1.1 Architecture Choice

The ideal model of current mode DAC is a static current source in series with an ideal switch, shown in Figure 1.1(a). The digital input controls the state of the switch, and the analog output is either the value of the current source or no current. When replaced with realistic transistors, as shown in Figure 1.1(b), the speed at which the digital input can switch is limited by transistor transit frequency and parasitic capacitances. A constant current means voltages at the output and internal capacitances charge linearly due to a constant slew rate, given by:

$$I = C \frac{dV}{dt} \quad (1.1)$$

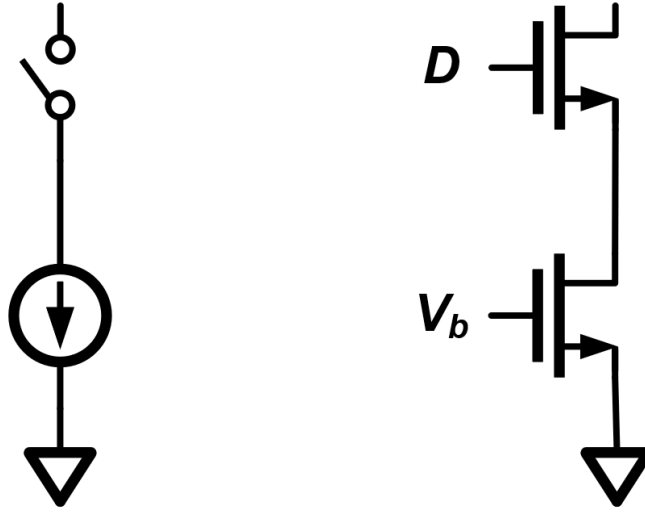


Figure 1.1: Single ended current mode DAC cell. (a) Ideal model. (b) Transistor level implementation.

In contrast, switched capacitor or resistive string architectures settle exponentially based on an RC time constant. Although at first it is faster than linear settling, it slows down as it approaches the target value, slowing down error decay. Higher resolutions demand smaller residual error, so exponential settling can take multiple RC time constants to reach that level of error. Linear settling can reach smaller error values faster since they don't decay.

Although fast, a single-ended design poses a few problems. When the cell switches from off to on, the sudden conduction of current creates a voltage bounce on the supply due to internal inductance of wires. The voltage bounce is described by:

$$V = L \frac{dI}{dt} \quad (1.2)$$

This bounce causes nonlinear distortion on the output signal. There are also many sources of noise that are seen by the output, such as noise from references, substrate, supply ripple, and clock jitter.

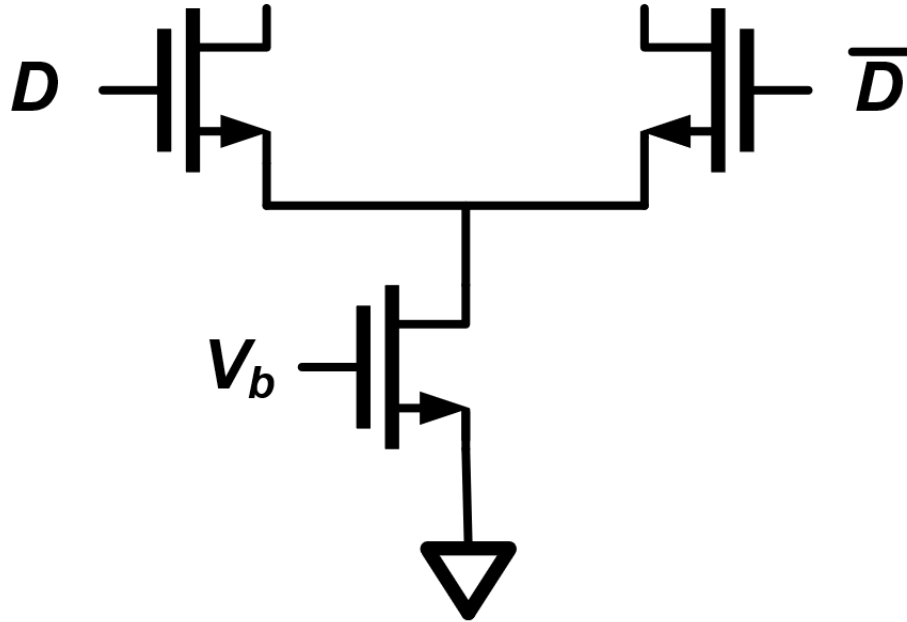


Figure 1.2: Transistor level implementation of a differential current steering DAC cell.

Many issues of the single-ended design are heavily mitigated by using a differential design [13], shown in Figure 1.2. It sees the benefits of common-mode noise suppression, canceling even-order harmonics, as well as maintaining a constant current draw from the supply, since the current source never turns off and is always steered to one side of the switching pair. As such, the design is referred to as the current-steering architecture.

1.2 Static Nonlinearities

When characterizing DACs, the most basic way to verify linearity is to provide a very slow ramp input that increments the digital code from zero to the maximum code the DAC can take. Then, a line is drawn through the endpoints and all DAC outputs for each code are compared to that line. From this, difference in magnitude from the line, called INL, and difference in step size, called DNL, can be determined. The ramp must be slow to allow complete settling of DAC outputs, ideally at DC, hence why nonlinearities relating to the ramp are attributed to static nonlinearities. For a DAC to be considered “sufficiently linear” in terms of static nonlinearity, both INL and DNL must be less than 0.5LSBs. If they are above 0.5LSBs, that means the DAC output at that code has transitioned to the zone of a different code, or that input converts to a wrong output. This section discusses various static nonlinearities that occur in the current steering DAC, and the linearization techniques used to mitigate their effects.

1.2.1 Transistor Mismatch and Sizing

One of the biggest issues with data converters is ensuring precision of analog values. For a current steering DAC, that issue lies in the value of the current source, or the LSB current. Due to fabrication process variation, an exact perfect current source cannot be created reliably. Various parameters are subject variation based on Gaussian distributions, where the ideal value is the mean.

A way to reduce the variation of this error is by increasing the size of the current source transistors. From Pelgrom’s law of transistor matching [1], the normalized variance of V_{TH} is inversely proportional to transistor area, given by:

$$\sigma_{V_{TH}}^2 = \frac{A_{V_{TH}}^2}{2WL} \quad (1.3)$$

Where $A_{V_{TH}}$ is a process related constant. To keep W/L constant, W and L must scale identically. Therefore, if we double the size of W and L to increase the area of the transistor by 4, the effective $\sigma_{V_{TH}}$ is reduced by 2. The current source size can be scaled up as much as needed for the effective variation of the current error is less than 0.5LSBs with whatever yield is designed for, which is often single sided 3σ of variation.

1.2.2 Finite Output Resistance

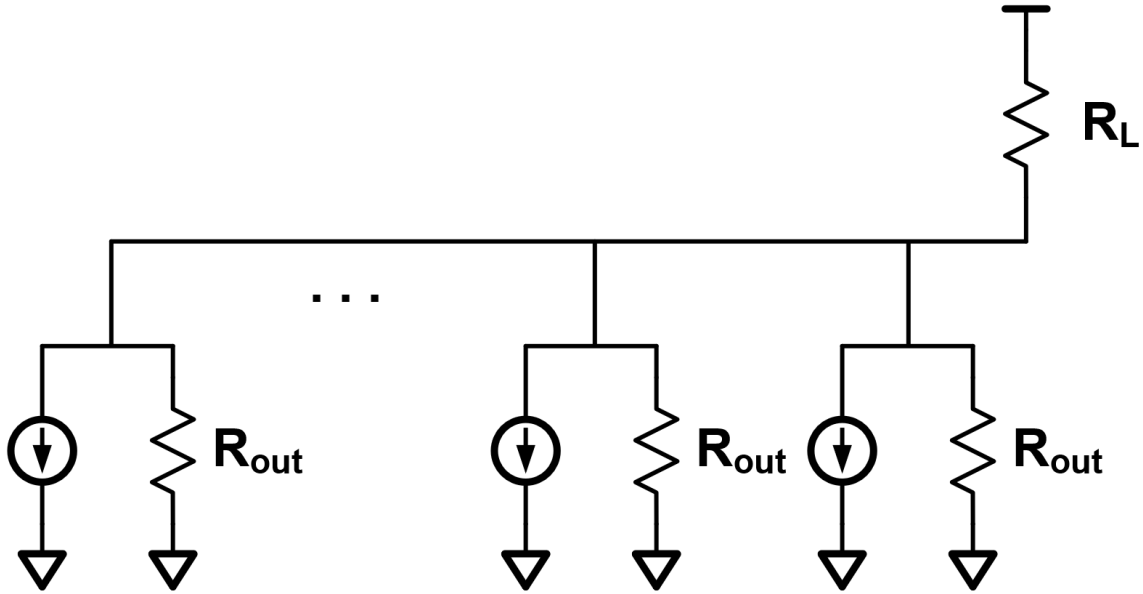


Figure 1.3: Representation of current source output resistance in parallel with a resistive load.

Variation is not the only issue that affects precision of the output. Another source of nonlinearity exists that doesn't directly affect the current but still creates imprecision. Current steering DACs often directly drive some resistive load, often 50Ω , which converts current into voltage to be readable. The current sources in the DAC have finite output resistance, which if there are enough current sources on, all combine in parallel with the resistive load to produce an effective resistive load that creates too much error in the output. The number of current sources in parallel on any side is determined by the digital input code, making this error a "code-dependent" error, as the effective output resistance is a function of the input code. Even if operated in strong inversion, single current sources almost never have enough output resistance to be within acceptable error tolerance at higher resolutions. So, a method of increasing output resistance is required.

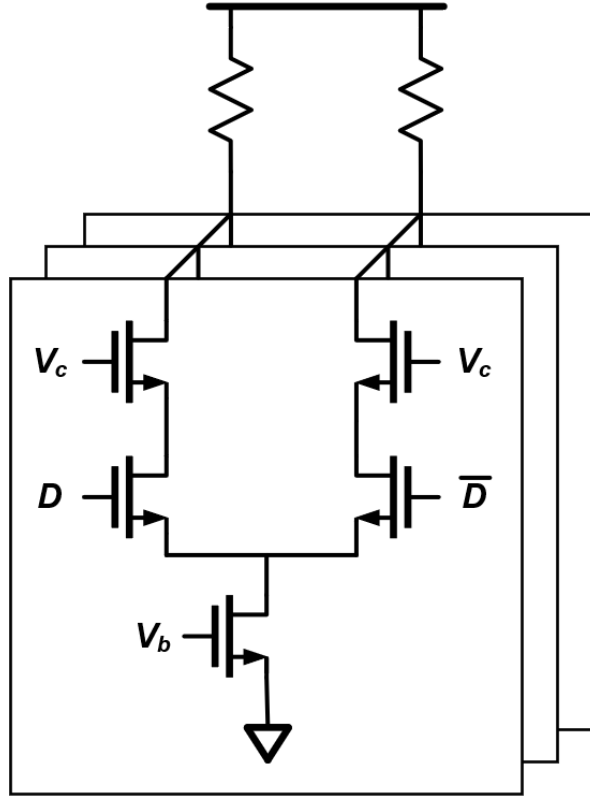


Figure 1.4: Cascoded current steering DAC cell with a resistive load.

The classical way of increasing the output resistance of a current source is to add a cascode device [2]. The switching pair transistors can be repurposed to be cascode transistors by simply biasing them into saturation when it is switched on. This will increase the output resistance of the current source by roughly a factor of $g_m r_0$. For high resolutions, the switching pair may not provide enough output resistance, and another transistor may need to be added above the switching pair, shown in Figure 1.4, to further increase output resistance by another $g_m r_0$. This transistor receives a constant bias, unlike the switching pair transistors below it. If needed, further cascode transistors can be added as well [2][10]. The effectiveness of cascading may decrease at higher frequencies, which will be discussed in a future section.

1.2.3 Nonmonotonicity and Segmentation

Another issue arises when considering the DAC as a whole. While achieving high output resistance is desired, it is not a constant value for all inputs of the DAC. It changes based on the digital input, and therefore the output voltage has an error that is different based on what the input is. Although increasing output resistance high enough mitigates this, it does not solve the issue of current source accuracy. For a fully binary N-bit DAC, the Nth bit, or the MSB, comprises half of all the current sources. There exists a state where every current source except the MSB are on, and the very next digital code corresponds to a state where only the MSB current sources are on. The transition from that code to the next is called the midscale transition, where every single current source in the DAC switches. This midscale transition causes many problems.

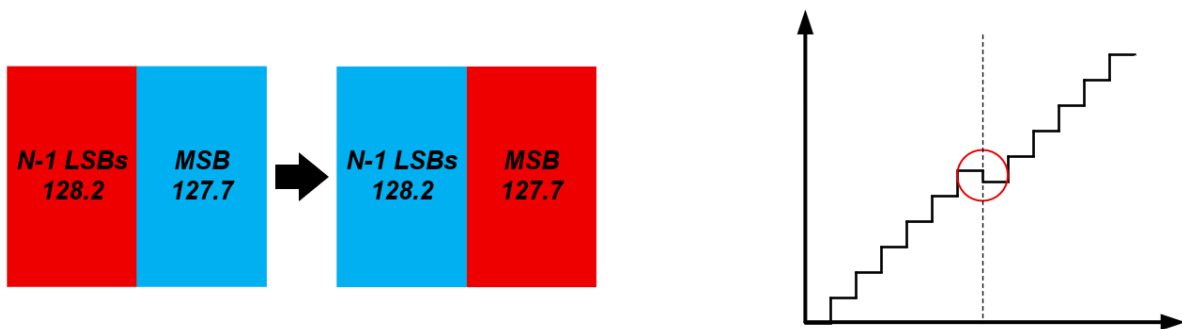


Figure 1.5: Nonmonotonicity at the midpoint transition. (a) Numerical representation. (b) Effect on the output ramp.

Consider an 8-bit DAC with mismatch, where the sum of all non-MSB current sources add up to 128.2LSBs and the MSB current sources add up to 127.7LSBs, resulting from some combination of process variation code dependent output resistance. Switching from digital code

127 to 128 means the current goes from 128.2LSBs to 127.7LSBs. The higher digital code ends up with less current than its previous value, shown in Figure 1.5. This is an example of nonmonotonicity, where increasing digital codes does not result in strictly increasing analog outputs. Nonmonotonicity is highly undesirable for a DAC and essentially requires perfect precision and output resistance to avoid, which is very impractical. Thus, we must find an alternative solution.

One solution is to avoid fully binary-weighted DACs and use a method that is guaranteed monotonic. If you convert the binary input number into a thermometer coded digital number, where every single bit is mapped to an individual current source, shown in Figure 1.6, you can guarantee monotonicity by simply toggling only 1 bit at a time as you increase the digital code. This way, increasing in digital code never switches off any current sources and only switches on one.

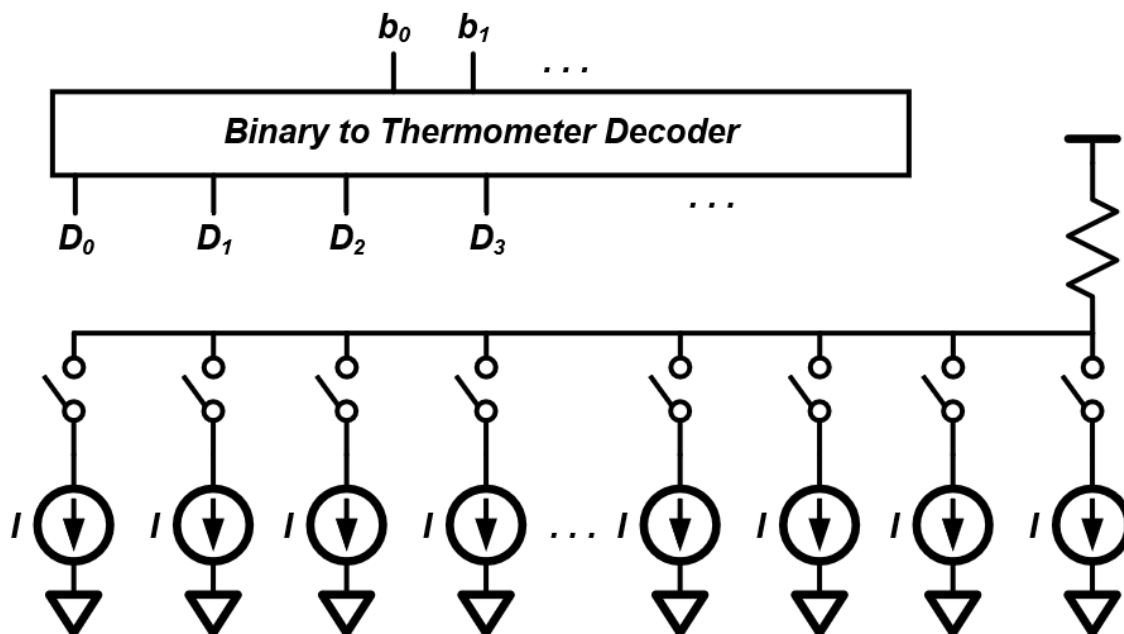


Figure 1.6: Fully thermometer coded DAC circuit.

However, fully thermometric DACs are impractical due to their overhead requirements. To get the thermometer coded number, it must pass through a binary to thermometer converter, a combinational logic circuit that scales exponentially with resolution. A high resolution like 12 bits requires 4095 outputs to be generated and therefore will take up an enormous amount of space, probably more space than the DAC itself. In addition, all 4095 of those inputs need to be routed properly to their respective DAC cells, which causes large routing congestion over the DAC.

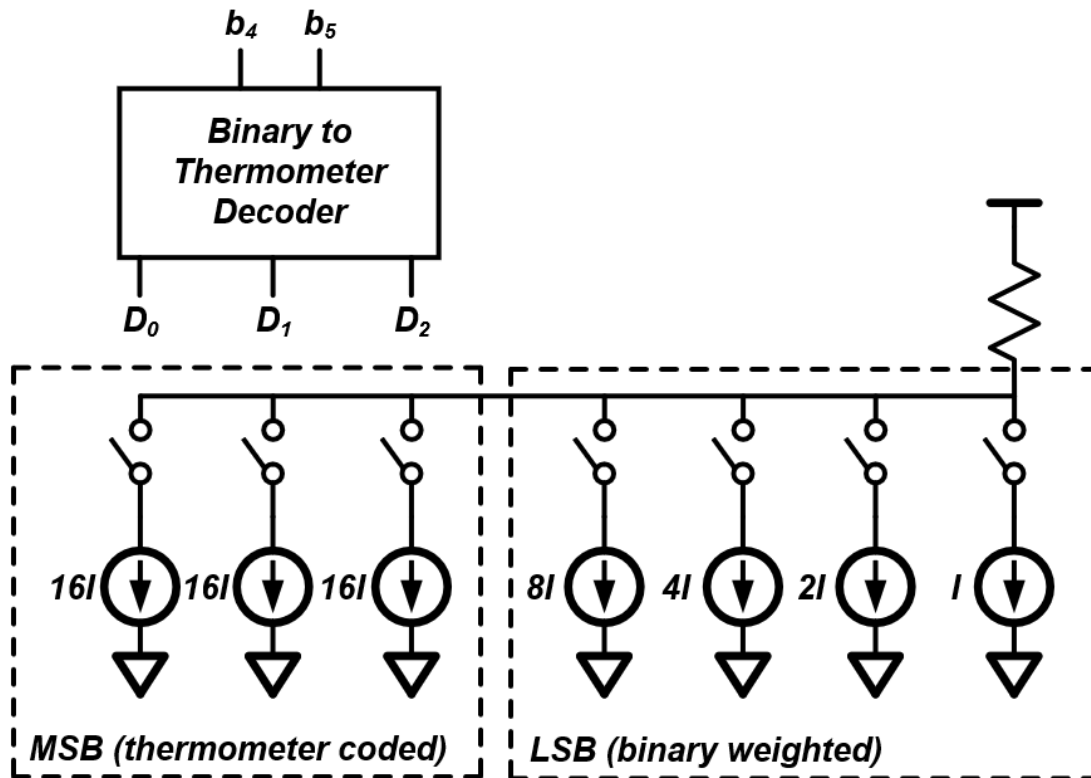


Figure 1.7: Segmented 6-bit DAC circuit with 4 LSBs of binary weighting and 2 MSBs of thermometer coding.

Both types of DACs have their strengths and downsides: binary DACs are simple but can suffer from nonmonotonicity, and thermometer DACs are guaranteed monotonic but have area issues. The common solution is to use a hybrid architecture and segment the DAC [10][13]. Small mismatches don't affect lower bits as much because there are fewer current sources, so nonmonotonicity is almost impossible there. Therefore, it's impractical to use thermometer coding for lower bits. On the other hand, higher bits have lots of current sources per bit, so nonmonotonicity is a real risk. The proper way to implement the split between binary and thermometer coded bits is to do binary-weighting until the $|DNL|$ at a certain code exceeds 0.5LSBs. The DAC should start thermometer coding before the code that triggered the DNL spike occurs. The thermometer coding should also group current sources based on the last binary weighted bit. For example, if a 6-bit DAC's DNL exceeds 0.5LSBs at the midscale transition code of 64, the MSB, which is normally a group of 32LSBs when binary-weighted, should be split into 2 groups of 16LSBs since that is number of LSBs the previous bit has, shown in Figure 1.7. This way, the number of DAC cell inputs is minimized.

1.3 Dynamic Nonlinearities

While static nonlinearities deal with precision issues for static DC conditions, dynamic nonlinearities show up when running a DAC at speed. These nonlinearities are generally characterized with a frequency spectrum plot to view spurious content. Figures of merit involving dynamic nonlinearities are defined for a digitized fullscale sinusoid input, as a sinusoid in an impulse in frequency domain. The most common metric is the difference in magnitude between the carrier sinusoid impulse and the highest nonlinear spur, called spurious free dynamic

range, or SFDR. However, due to the effects of quantization, there is a finite noise floor that is defined by the DAC resolution.

All spurious nonlinearities in DACs are caused by some form of code dependency or periodicity. With a sinusoid that hits every possible digital code, any error that occurs at specific values with a pattern will repeat periodically due to the nature of a sinusoidal wave. Any form of periodicity translates to a pattern frequency, which is then mapped onto the frequency domain as an impulse, like a sinusoid. The magnitude of that impulse is proportional to the energy of the disruption, which we wish to minimize.

1.3.1 Switching Glitches

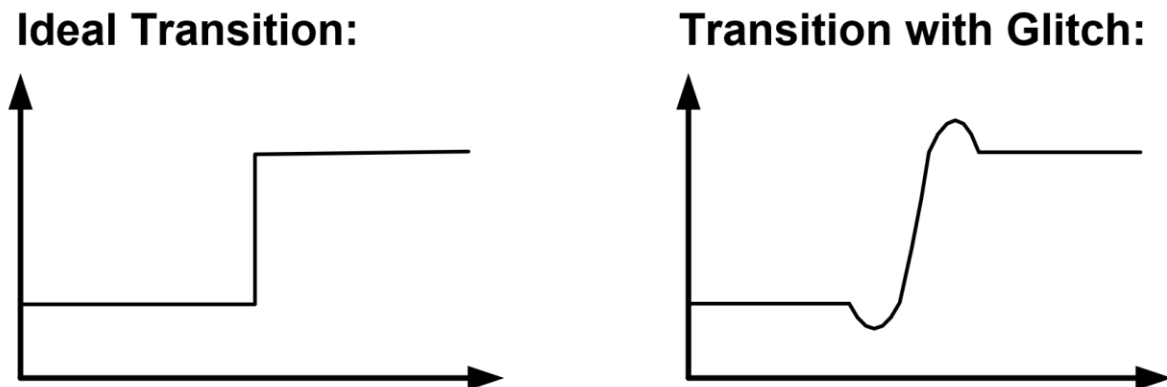


Figure 1.8: DAC output transition. (a) Ideal transition. (b) Transition with switching glitch.

One immediate dynamic nonlinearity that appears especially at higher frequency is switching glitches. When the switching pairs are trying to flip the current from one side to the other, depending on the method of switching, there is a moment where both switches are on or off, which will modulate the drain voltage of the current source and change the current for a

moment. This will change the output voltage by shaping the output waveform, particularly at the transition edges, shown in Figure 1.8(b). Glitches by themselves are not the problem, but when they happen at specific digital codes, they become code-dependent and cause nonlinearity in the form of spurs.

Preventing glitches is difficult, but they can be tolerable if we mitigate how severe they are. The glitch energy of one DAC cell is generally negligible, but if multiple cells are switching, it can become an issue. The primary glitch to be concerned with is the midscale transition, corresponding to the digital code/s in a fully binary weighted DAC where every single cell switches and the glitching energy is the highest.

Segmentation, described in the previous section, prevents high code-dependent glitch energy by limiting the number of switching elements at higher bits. By doing this, the magnitude of switching glitches is reduced at higher codes, where the most number of cells are switching, and spurs are reduced [2].

1.3.2 Randomization

Deterministic strategies to mitigate nonlinearity work well, but the most popular techniques use randomization to make code dependency and periodicity look random. Random elements don't appear as nonlinear spurs in the frequency domain but instead contribute to the noise floor. Therefore, we can mitigate nonlinearity by sacrificing noise floor, as shown in Figure 1.9.

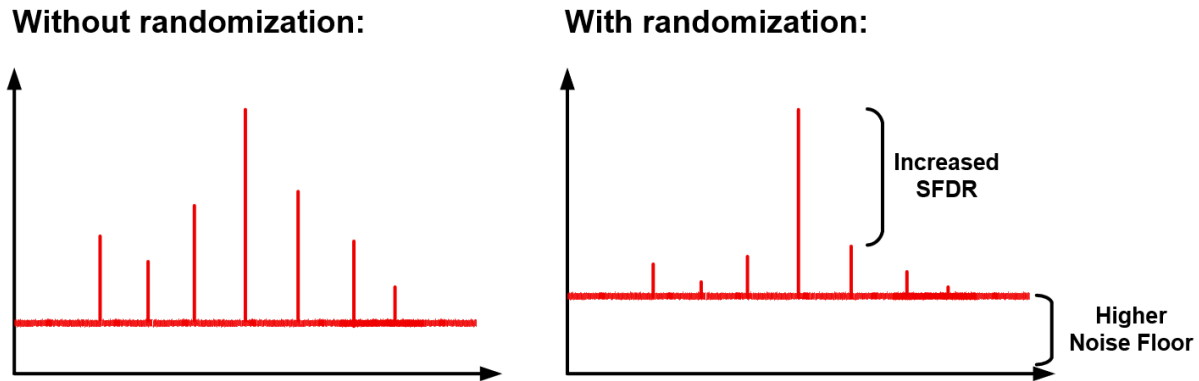


Figure 1.9: Output frequency spectrum of a DAC. (a) Without randomization. (b) With randomization.

One way to introduce randomness is on the digital input before it even enters the DAC. By injecting a small amount of random noise called dither in the input, the digital input occasionally flips its LSB at random, deliberately causing an error. However, this error appears as random noise and does not introduce any nonlinearity. Instead, over a longer period of time, it breaks up the periodicity of code dependent nonlinearities by causing them to not happen sometimes, reducing spur energy and turning it into noise energy instead. Spurious content is often more undesirable than noise content, so this is a tradeoff that is generally desired.

1.3.3 High Frequency Output Impedance

Finite output impedance appears as an issue again at high frequencies, but to a much harsher degree. At higher frequencies, parasitic capacitances have lower impedances, and therefore further deviates the effective load impedance of the DAC. While cascoding was a good solution

at lower frequencies, it does not help enough at higher frequencies due to the lower impedance of the parasitic capacitances [2].

The most obvious solution to this problem is to lower the parasitic capacitances as much as possible, but this solution is limited by the technology and fabrication process. In addition, as discussed in the static nonlinearity section, increasing the size of transistors is required to reduce current variation, but that also increases parasitic capacitances and degrades high frequency dynamic performance. Both reducing current variation and lowering parasitic capacitances are desired for a high speed DAC.

A different solution was proposed in [2], which takes advantage of differential operation and additional cascoding. Shown in Figure 1.10, DC current sources are placed between the switching pair and the additional cascode transistor, ensuring that even if one side is switched off, that side of the DAC cell still draws some amount of static current, essentially keeping it on to some degree. This static current allows the parasitic capacitances from the DAC output to the node above the switching pairs to be seen by the outputs, and are partially canceled differentially, shown in Figure 1.11, reducing the effect of some parasitic capacitances on dynamic performance. The amount of current needed to be burnt statically depends on dynamic parameters, but generally burning more current will improve dynamic performance, although it has diminishing returns.

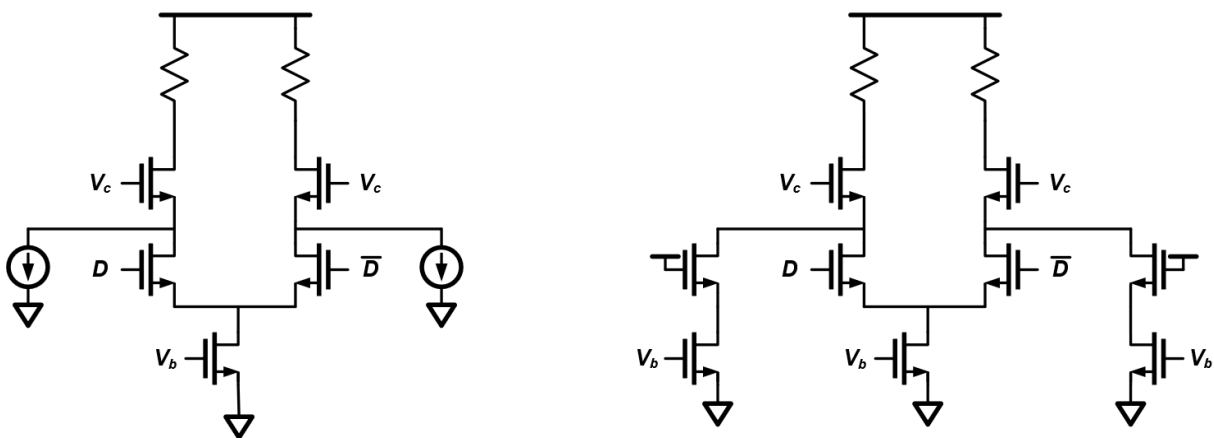


Figure 1.10: DAC cell with keep-alive currents. (a) ideal current source version. (b) circuit implementation.

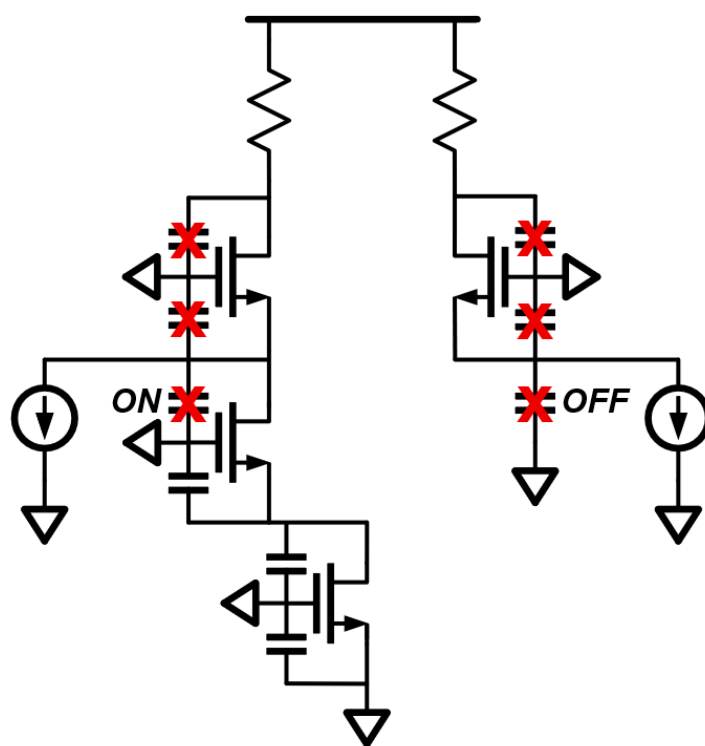


Figure 1.11: DAC with one side on, parasitic capacitances being canceled out differentially through keep-alive currents

CHAPTER 2

Background on DAC Calibration

Current steering DACs have many nonlinearities to consider, and their solutions have tradeoffs with other nonlinearities. For example, sizing up transistors improves static accuracy by reducing variation, but worsens dynamic performance by increasing parasitic capacitances. In the end, both types of nonlinearities are crucial to minimize to get good performance at high speeds. Calibration is a common technique used to improve static accuracy without increasing the size of current sources. It keeps transistor size as small as possible, letting them be subject to large variation, and instead uses area for circuitry that corrects or trims the value of the current sources to their expected value through a negative feedback loop. It is often done at a low speed before the DAC performs any conversion, so there are no overhead effects on the DAC during actual operation. The concept of calibration seems excellent on paper; the challenge lies in the circuit implementation.

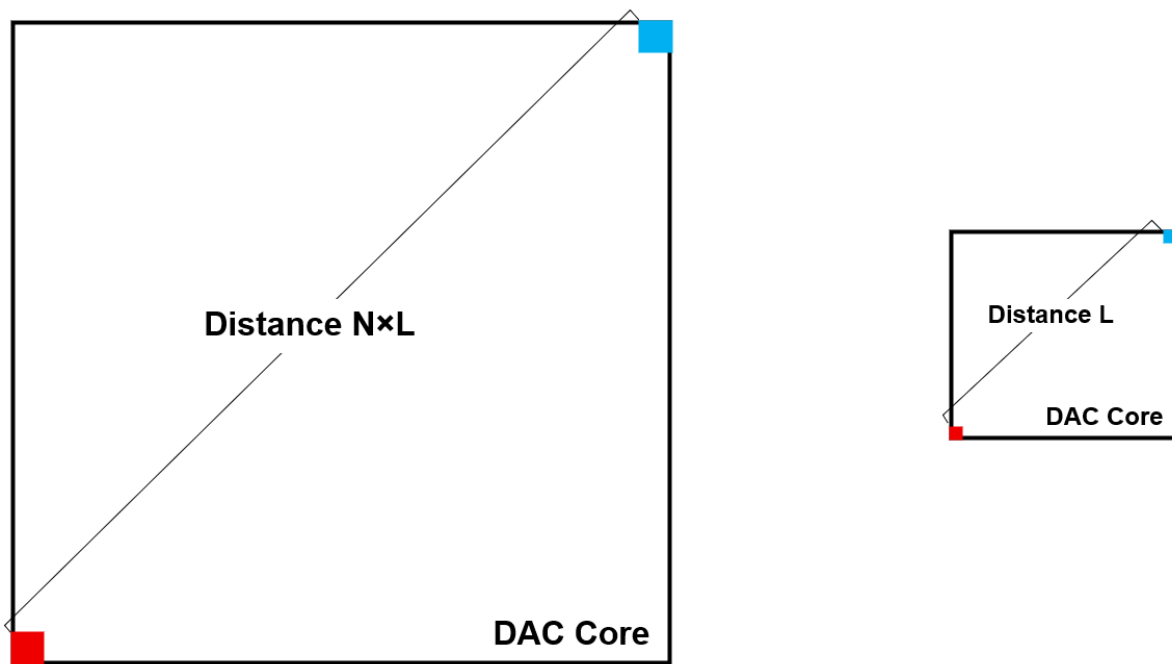


Figure 2.1: Comparison of timing mismatch with 2 DAC core sizes.

One benefit of calibration is that it keeps the circuitry within the signal path of the DAC small in terms of area. Besides minimizing parasitic capacitances, reducing the overall DAC core area mitigates differences in input data propagation between different DAC cells. If the DAC was laid out as a rectangular 2-dimensional array of elements, the cells on opposite corners of the DAC core will have the biggest time difference for when the input data reaches and settles the cell. If the data moves too fast, the input to one of the farther cells could not receive that cycle's data before the next data transition happens, like timing violations in digital circuits. Even if all cells do toggle before the next data transition, due to differences in propagation delay, there could be a moment where the DAC is producing the wrong code because one cell hasn't turned on yet. This brief moment of timing mismatch still shapes the output signal and causes

nonlinearity if this happens at specific codes. High data rate corresponds to less time for all DAC cells to receive input data, so a small DAC area is desired for high speed operation. As shown in Figure 2.1, the less distance the signal has to propagate, the less time it takes for the cell to toggle, mitigating the timing mismatch issue. To have both small area and accurate currents, calibration is a very attractive option.

2.1 Conventional Calibration

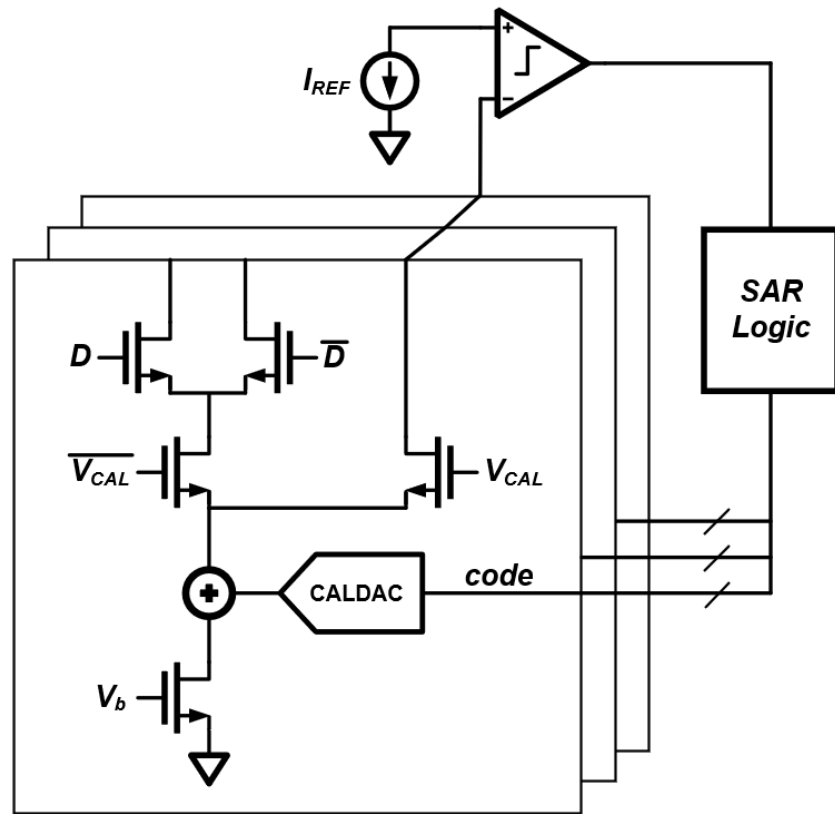


Figure 2.2: Conventional subDAC/CALDAC-based digital calibration loop.

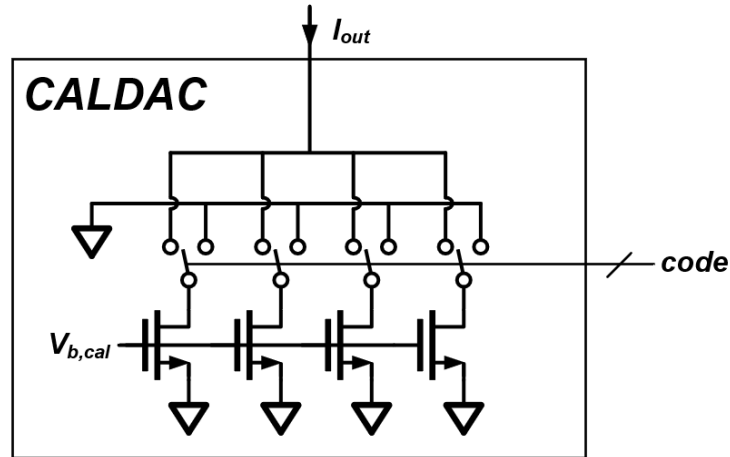


Figure 2.3: Conventional subDAC/CALDAC implementation.

The adjustment of current must be an analog quantity, but using an analog feedback loop with fully analog components is area and power hungry. The conventional calibration loop uses a SAR loop, as shown in Figure 2.2, to compute digital input bits to a sub-DAC, commonly referred to as the CALDAC [3] [4], whose implementation is shown in Figure 2.3. The CALDAC is attached to the current source and provides steps of analog current across input bits, which can be used to provide current correction in only those steps, defining a calibration resolution. Therefore, truly precise current correction is not reliably possible with this architecture, but the current can be close enough such that the error doesn't severely affect performance.

Not every current source needs to be calibrated, in fact it is quite difficult to do so. In general, current steering DACs drive differential 50Ω or 100Ω with 1V or 2V differential swing, corresponding to a fullscale current of 20mA. For high resolutions, such as 12 bits, the LSB current for 20mA fullscale current is around 4.88uA. Especially in finer technology nodes,

4.88uA is not possible for a minimum length transistor to conduct while in saturation; it is usually much higher. Therefore, channel length is often increased to decrease the current to that LSB value. Calibrating the LSB requires a much smaller value of precise current, which becomes impractical in terms of area. So, only higher bits are calibrated typically. The lower bit current sources don't need to be calibrated because the increase in channel length usually reduces the variation enough that their intrinsic matching is sufficiently accurate. Calibration is needed once the intrinsic matching of a bit slice is too inaccurate, and the subDAC LSBs would be a more achievable current value.

2.2 Contribution of this Thesis

While this calibration structure works well, it has some issues that degrade high speed performance. The subDAC itself must be near the DAC core to provide accurate current, and therefore is essentially a part of it, increasing area. In addition, the subDAC output needs a connection to the current source itself in order to correct it. This is often done by connecting it to the drain of the current source, adding its parasitic capacitances to that node and degrading dynamic output impedance. Since it must be attached to the current source itself and not through the full cascode structure above it, its effect on performance is larger.

Improvements can be made on DAC calibration by further isolating the calibration circuitry from the main signal path and reducing the complexity or size of calibration circuitry. This thesis tackles this issue by proposing the following:

- A calibration structure using CTTs that further isolates calibration circuitry connections from the main DAC signal path
- A multi-step calibration technique to cancel out large residual error from the CTTs using additional CTTs
- An analysis and model of CTTs and their programmed current range, with calculations to minimize the number of CTTs needed for a given DAC and its specs

CHAPTER 3

CTT-based DAC Calibration

3.1 CTT Background and Model

The CTT is a device that can trap charge in its high-k gate dielectric due to oxygen vacancy defects. Under proper biasing voltages and conditions, high voltage pulses can be applied to the gate to trap charge in the vacancies, modulating the channel's effective electric field [5][6]. This is similar to floating gate transistors, but they do not require any additional process overhead like floating gates do. This effect has been observed in Global Foundries 22nm technology, which will be the chosen technology for this thesis.

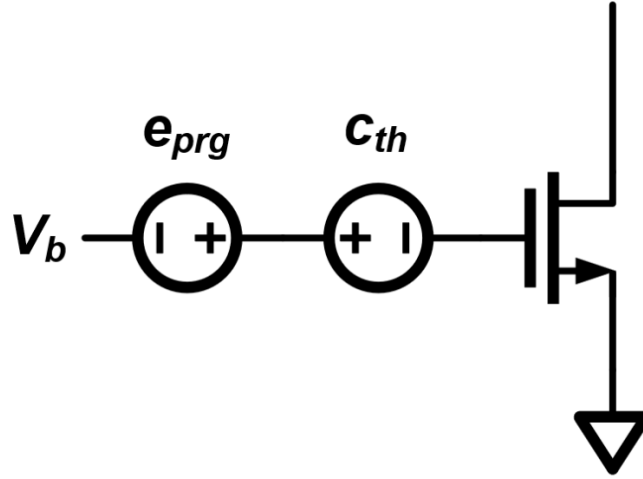


Figure 3.1: Circuit-based CTT model.

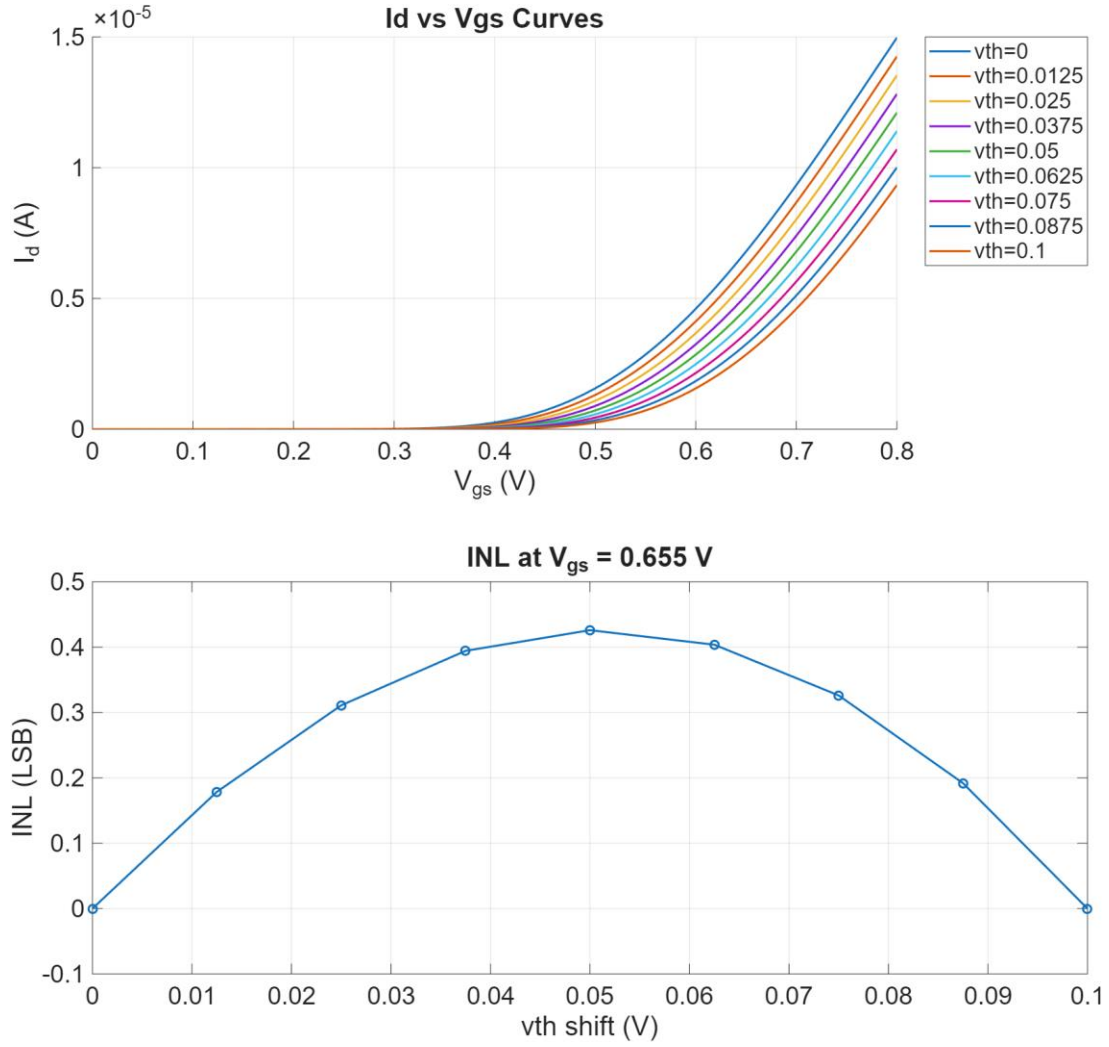


Figure 3.2: (a) Family of programmed curves for different v_{th} shifts. (b) INL of programmed current values for a chosen bias point.

The effect of electrons being trapped in the oxide can be modeled as a threshold voltage shift. The v_{th} shift can be modeled as a voltage source shown in Figure 3.1 as c_{th} . The shift increases as more electrons are trapped, or as a longer high voltage pulse is applied to the gate. Prior art has shown that the maximum v_{th} shift is around 166mV [5].

Programming a CTT to a certain v_{th} shift is not perfectly accurate. There is a programming error similar to quantization error of a data converter, defined by a resolution. For N-bit programming resolution, there are 2^N current curves that describe the quantized currents that the CTT can be programmed to. The error can be modeled as a uniform random variable, described by:

$$e_{\text{prg}} \sim \mathcal{U}(-0.5 \text{ LSB}_{\text{prg}}, +0.5 \text{ LSB}_{\text{prg}}) \quad (3.1)$$

Where LSB_{prg} is the LSB current step between each programmed currents at a chosen bias point, defined as:

$$\text{LSB}_{\text{prg}} = \frac{I_{\text{fs}}}{2^N} \quad (3.2)$$

Where I_{fs} is the total current range of a CTT, which can be obtained by sweeping the v_{th} shift from 0 to the maximum v_{th} shift desired in 2^N steps, shown in Figure 3.2(a). The error can be incorporated into the model by subtracting it from the v_{th} shift, also modeled in Figure 3.1.

The above equations hold only if the current values between quantized shift values at a given voltage bias are equidistant to each other. For this to happen, the CTT must be biased in strong inversion, where the distance between adjacent current values are more equal for equidistant values of v_{th} shift. To determine if a bias point is sufficiently “linear”, the INL of the current steps at different v_{th} shift values at a given bias can be plotted, as shown in Figure 3.2(b). If the INL is greater than 0.5LSBs, the programming resolution should be decreased to drop the INL below 0.5LSBs. Currently, programming resolution of CTTs has been shown up to 6 bits of programmability [5].

3.2 CTT Calibration Structure

A CTT can be viewed as a very compact subDAC, where the digital input and SRAM that stores the calibration bits are encoded in the threshold voltage of the CTT. Since it itself is the current source that it trims, the entire subDAC circuitry is compacted into the current source as a CTT.

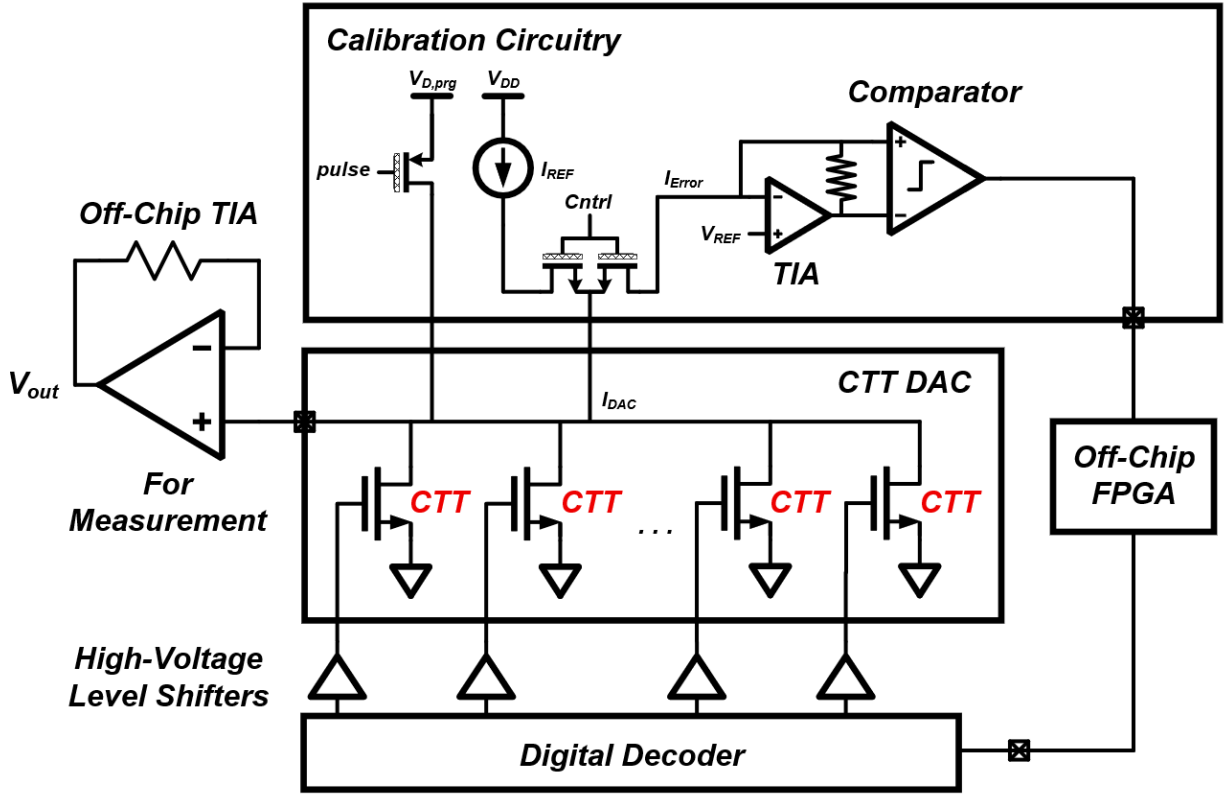


Figure 3.3: CTT-based DAC design and calibration circuitry in [6].

CTT-based DAC calibration has been demonstrated in the past [6] with successful results, shown in Figure 3.3. However, the calibration demonstrated in [6] has issues that prevent it from working with a differential current-steering design at high speeds. One issue that arises is the structure used for programming CTTs. The calibration circuit in [6] compares the DAC

output current with a reference, and routes the error current to a TIA, which then goes into a comparator to determine whether the CTT current is higher or lower than the reference. The comparator's decision is processed through an FPGA and goes to a digital decoder that selects a particular CTT for calibration. The decoder then triggers a high-voltage level shifter to apply a high-voltage pulse to the gate of the selected CTT, applying a positive voltage to lower the current or negative voltage to raise it. In addition, a high voltage is also forced onto the CTT drain node during programming to apply a proper biasing condition. Since the current source is the CTT, this implies that circuitry must be attached to the drain node of the current source to force it to a certain value, which adds parasitic capacitance to the DAC core and slows it down. High voltage is also very dangerous in a differential current-steering design, as the transistors above the current source may be affected by it.

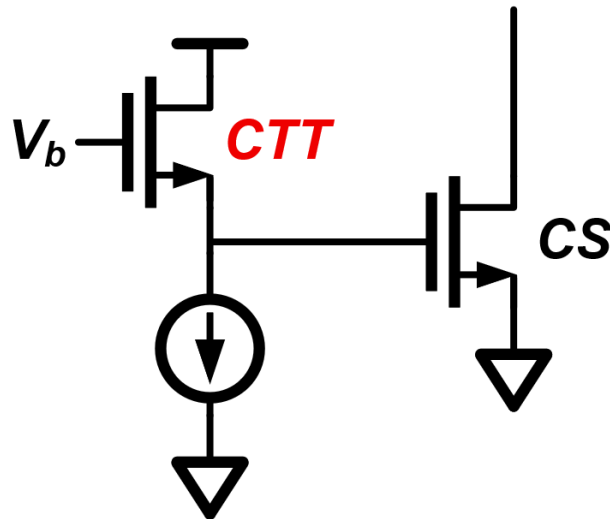


Figure 3.4: CTT source follower driver for a DAC current source.

Instead of replacing the current source with a CTT, we move the CTT to the input transistor of a source follower whose output drives the gate of the current source, shown in Figure 3.4. The source follower output is simply a copy of the CTT input, with some v_{th} offset that can be accounted for, and will change accordingly with CTT programming. In this configuration, programming the CTT to a certain v_{th} shift also shifts the source follower output by that amount. Through this, we can set V_b to half the v_{th} shift range above the nominal current source bias voltage, accounting for any v_{th} offsets, then program the CTT to the half the v_{th} shift range to apply the nominal bias voltage to the current source gate. At this midscale v_{th} shift, there can be a positive or negative v_{th} change with respect to the nominal bias voltage through programming, allowing current source mismatch to be calibrated for both positive and negative errors. The current source gate voltage can effectively be programmed to a value within V_b , the highest voltage possible, and the maximum v_{th} shift below it, the lowest voltage possible. By comparing the current source current with a target reference current, its gate voltage can be tuned until it matches the target, calibrating the current source.

This arrangement greatly isolates the CTT from the DAC signal path, which not only reduces the parasitic capacitances on the current source drain node compared to the arrangement in [6], but also protects the current source drain node from the high voltages needed to program the CTT. We can still use the same circuitry and drivers in [6] but apply them in to the new configuration.

Another issue in [6] is that the DAC is fully thermometric, and every single current source is a CTT. Although compact, CTT programming is very complicated and requires a long process to properly calibrate each device [6]. For a high resolution DAC, programming every single current source would take too long, and there would be heavy routing congestion for the

thermometric DAC inputs. So, we want to adopt a segmented DAC to reduce the number of DAC inputs, and we want the number of CTTs should be minimized to reduce how many CTTs we need to program. We can calculate the minimum number of CTTs needed in each bit slice to correct the DAC for a given statistical yield.

First, we review device matching statistics to determine the magnitude of mismatch to correct for. For every bit slice of a DAC, there are a specific number of unit cells corresponding to their binary (or thermometric) weight, which we can define as M . Each unit cell contains a current source that draws current equal to LSB_{DAC} . Every unit cell contains a current source that exhibits error modeled as a Gaussian distribution:

$$\sim \mathcal{N}(0, \sigma_{\Delta I}) \quad (3.3)$$

Where $\sigma_{\Delta I}$ is the standard deviation current error of a transistor. There are M independent and noncorrelated sources of $\sigma_{\Delta I}$, which add in squares. So, the total σ of the entire bit slice is:

$$\sigma_{total} = \sqrt{M} \cdot \sigma_{\Delta I} \quad (3.4)$$

For single sided 3σ variation, or double sided 6σ variation, the error entire bit slice error is:

$$e_{\Delta I} = \sqrt{M} 6\sigma_{\Delta I} \quad (3.5)$$

To be considered sufficiently linear, the error must fulfill this accuracy condition:

$$\sqrt{M} 6\sigma_{\Delta I} < 0.5 LSB_{DAC} \quad (3.6)$$

This condition must be met by a DAC to produce accurate analog outputs. If calibration were not involved, the transistor would be sized up to reduce $\sigma_{\Delta I}$ to match this condition.

With calibration, instead of reducing $\sigma_{\Delta I}$, we seek to cancel it with a correction term. We can define a single CTT's current correction range as c_I , which is the difference in current of a CTT between 0mV and maximum v_{th} shift. We can calculate the number of CTTs needed to correct the entire error range of a bit slice with this condition:

$$\sqrt{M} 6\sigma_{\Delta I} \leq K_1 c_I \quad (3.7)$$

Where K_1 is the number of CTTs needed to correct the mismatch every current source in the bit slice. After rearranging, we obtain the minimum number of CTTs needed as:

$$K_1 \geq \frac{\sqrt{M} 6\sigma_{\Delta I}}{c_I} \quad (3.8)$$

If the K_1 is a non-integer number, we can simply round up to the nearest integer, as there can only be an integer number of CTTs. If K_1 is larger than M , that means it requires more CTTs than number of current sources to correct the mismatch of the bit slice, which is undesirable.

This alone is not enough to sufficiently calibrate the DAC. We must now account for the residual programming error that the CTTs will produce. The total error range of K_1 CTTs is:

$$e_{K_1} = K_1 \left(\frac{1}{2} \text{LSB}_{prg} - \left(-\frac{1}{2} \text{LSB}_{prg} \right) \right) = K_1 \text{LSB}_{prg} \quad (3.9)$$

If this is greater than 0.5LSB_{DAC} , we must do a second stage of correction using more CTTs to further correct it. To correct the error range of the CTTs, we fulfill this condition:

$$K_1 \text{ LSB}_{prg} \leq K_2 c_I \quad (3.10)$$

Which simplifies to:

$$K_2 \geq \frac{K_1 \text{ LSB}_{prg}}{c_I} \quad (3.11)$$

If K_2 CTTs still produce too much error, then this process can be repeated as many times as needed. The magnitude of error after calibration can be modeled just by the last K -th set of CTTs, and the maximum number of CTTs that the last K set can have is:

$$K > \frac{\text{LSB}_{DAC}}{\text{LSB}_{prg}} \quad (3.12)$$

Therefore, we need $K_1 + K_2 + \dots + K_i$ CTTs in a bit slice to sufficiently correct its error to an acceptable range. From the total M current sources, we pick $K_1 + K_2 + \dots + K_i$ to be CTT calibrated, and the $K_0 = M - (K_1 + K_2 + \dots + K_i)$ remaining current sources just have a constant midscale bias and do not need to be calibrated since the correction from K_1 accounts for their mismatch as well.

K_1 should be minimized to ensure minimum number of CTTs, as K_1 also influences subsequent sets of CTTs. To do this, the number of current source mismatches that a CTT can correct for should be maximized so that fewer CTTs are needed per slice, especially at the larger bit slices. In other words, c_I should be maximized while keeping the current steps sufficiently “linear” based on their INL, as shown in Figure 3.2(b).

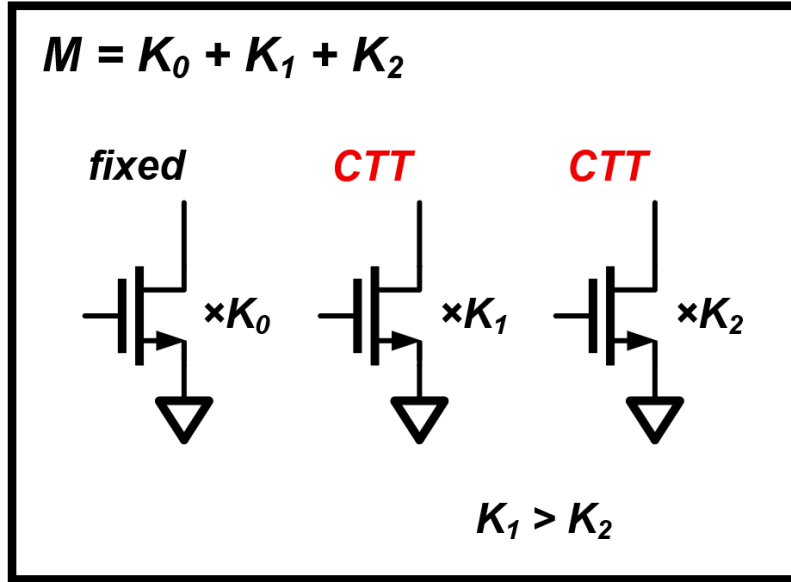


Figure 3.5: Number of CTTs needed per bit slice.

An example breakdown of the number of CTTs needed per bit are shown in Figure 3.5 with just 2 sets of CTTs. The statement $K_1 > K_2$ assumes that one CTT can correct for multiple CTTs' error ranges. If this condition is not met, the error produced by set K_2 is greater than or equal to the error of set K_1 , and therefore is not useful as the purpose of K_2 is to effectively reduce the error of set K_1 . Similarly, one CTT should be able to correct for the error of multiple current source mismatches. If a CTT can only correct less than one current source mismatch error range, There would need to be more CTTs than number of current sources in the DAC, and the calibration scheme must be changed.

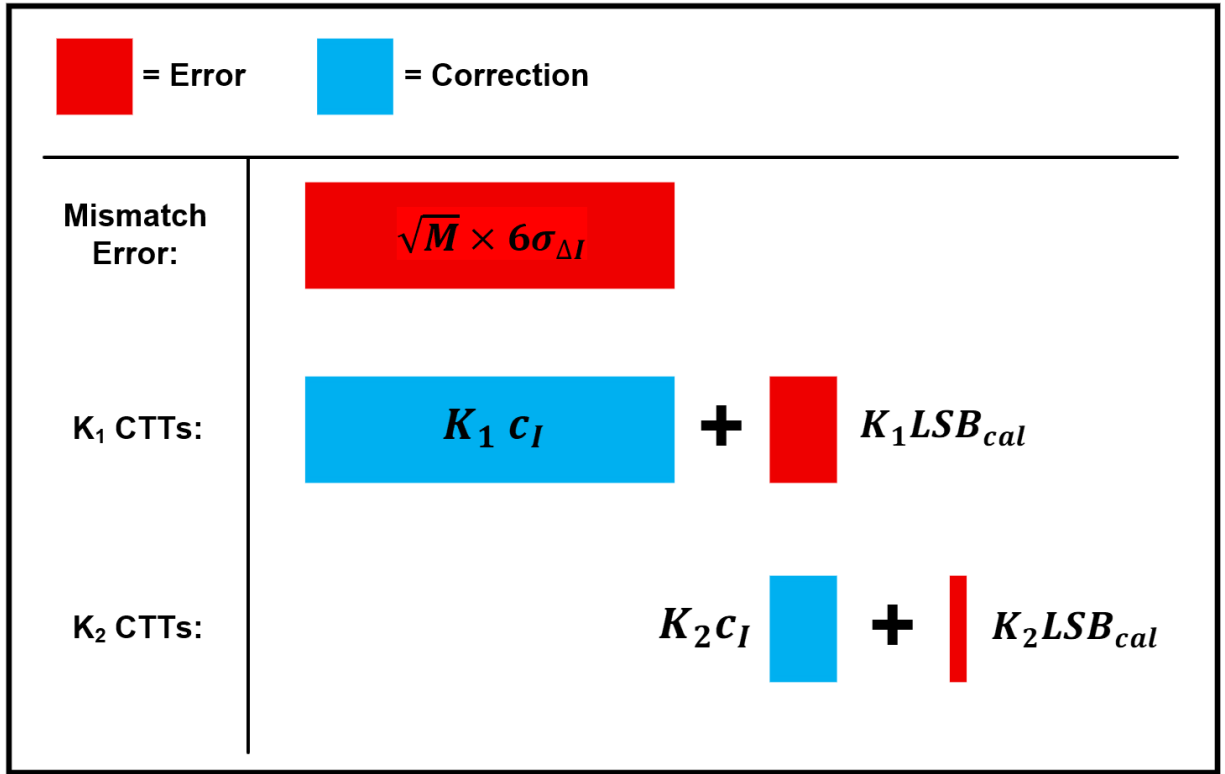


Figure 3.6: Multi-step CTT calibration procedure.

Figure 3.6 shows an example breakdown of error calibration for 2 sets of CTTs as well. In hardware, this implies that calibration must be done in multiple steps, as the error of the set K_1 only becomes apparent after they have been fully calibrated. Therefore, there must be as many calibration steps as there are number of CTT sets. During the calibration of set K_1 , all succeeding sets have their gate bias tied to the midscale bias to act as fixed current sources until it is their turn to be calibrated.

3.3 Temperature Tracking

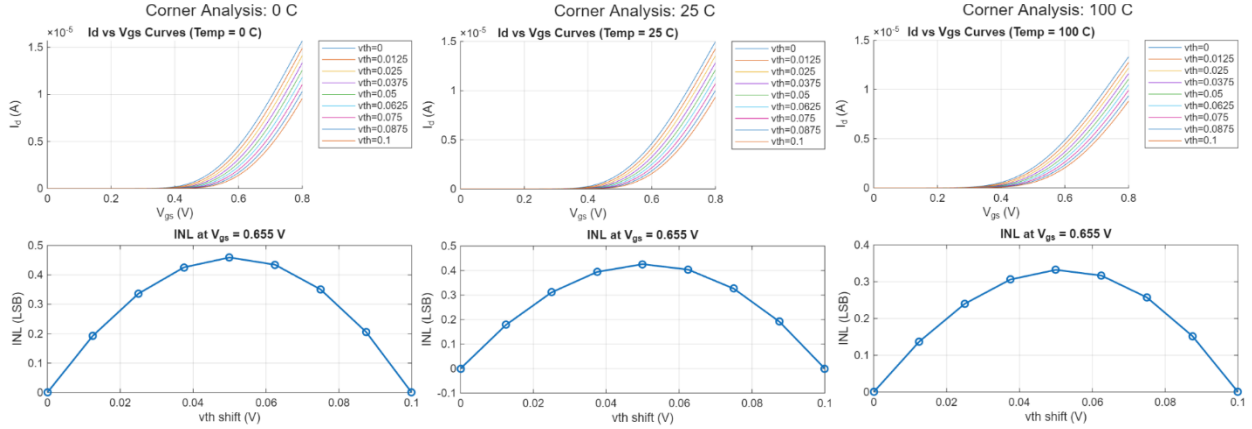


Figure 3.7: INL measurements of CTT curves at different temperatures.

One concern about DAC calibration is whether their correction will hold over different temperatures since they are being calibrated at only one temperature. As temperature changes, both current and transconductance g_m change in different ways [7]. The CTT-based calibration has some method of tracking temperature that relies on the equidistant current steps at the bias point. Across temperature, the I_d/V_{gs} curve of a transistor changes shape. If biased in strong inversion, the family of programmed curves all change in shape and the distance between the currents change, but they remain equidistant. As long as the currents remain equidistant, the current steps simply scale with the LSB current at a given temperature, and the calibration still holds. The DAC in [6] had current sources biased in sub-threshold, where the current steps between programmed curves are not equidistant to begin with, and so current steps would change differently between curves across temperature. Biasing in strong inversion not only shows the benefit of higher output impedance that we already need, but also more robustness to temperature variation.

Realistically, the linearity of the current steps may change based on temperature. Through simulation, it has been observed that the INL of the current steps do change with temperature, but very little and not enough cause significant error in the equidistant current steps, as shown in Figure 3.6. For these small deviations in INL, a temperature tracking circuit can be used to offset the bias voltage and slightly correct the INL.

CHAPTER 4

Design of a CTT-based 12-bit DAC

4.1 DAC Core Design

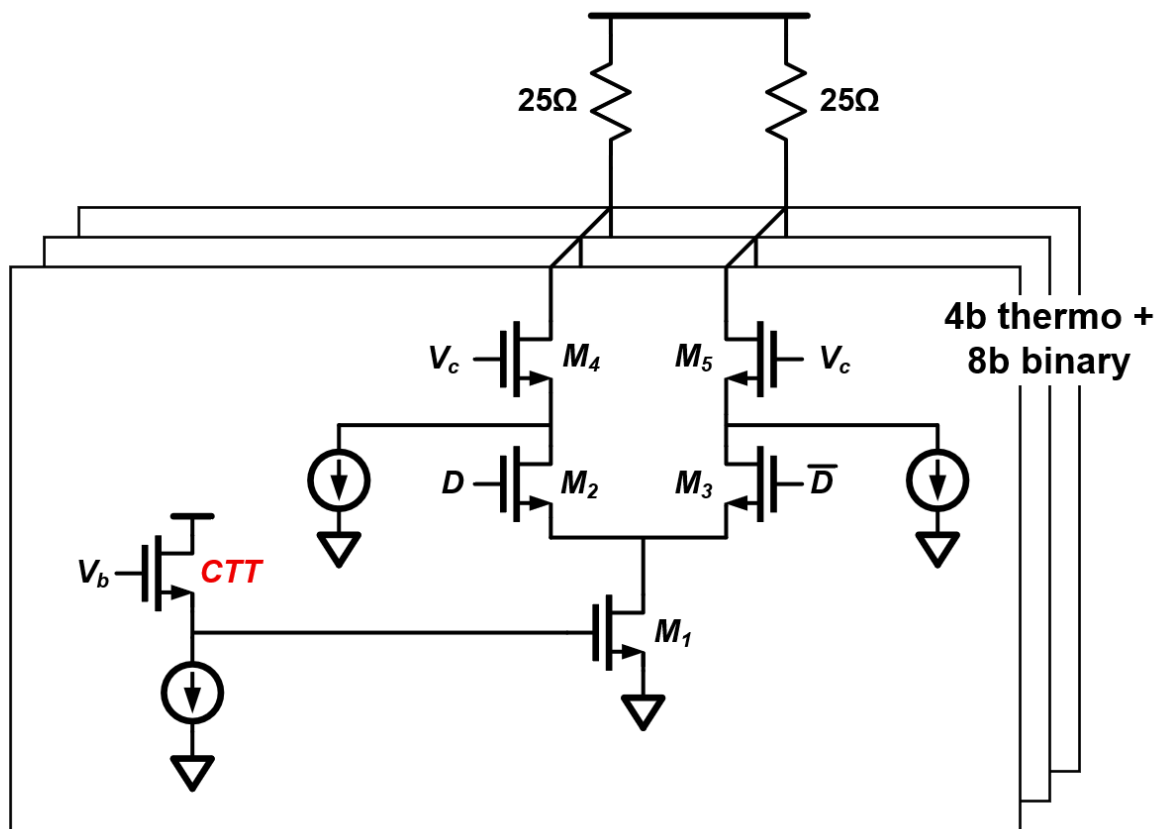


Figure 4.1: Implemented 12-bit DAC design.

Figure 4.1 shows the overall DAC design that will be described in this chapter. A 12-bit resolution was chosen as a reasonable resolution to represent higher resolution DACs. As for

speed, the target sample frequency was 20GS/s, a comparable speed to high speed high resolution DACs in prior art [8][9]. A fullscale current of 20mA was chosen into 25Ω resistive loads on each side, creating a differential resistive load of 50Ω , to create a differential voltage swing of 1V.

4.1.1 Current Source Design

To achieve 20mA fullscale current with a 12-bit DAC, the LSB current must be $20/4096 = 4.88\mu\text{A}$. This current must be achieved at a certain midscale bias point determined by the CTT's family of programmed current curves. We wish for the transistor to remain in strong inversion for the entire range of v_{th} shifts, which requires a high drain-source voltage. To get the family of curves, a size and v_{ds} must be chosen initially and verified of the other conditions. After a recursive verification loop of different sizes and biases, the size was settled to be 80nm/215nm with a midscale bias of 0.605V and a v_{ds} of 0.35V, resulting in a current of $4.82\mu\text{A}$, which is close to $4.88\mu\text{A}$. Figure 3.2(a) shows the family of curves for 3-bit CTT programming resolution, with Figure 3.2(b) showing the INL at the high bias of 0.655V, meaning a 100mV v_{th} shift range in order to keep 0.605V at the midscale with 50mV v_{th} shift. These CTT parameters were chosen as higher v_{th} shift ranges resulted in INL exceeding 0.5LSBs, and the same reason for programming resolution.

4.1.2 Switching Pair Design

The switching transistors size was set to be 80nm/28nm, unlike the current source size. The current source size is very large, and it is not desired for the entire DAC to be of the same

size, as long as they can still support the current that the current sources draw. The switching transistors need to be biased in saturation to carry $4.82\mu\text{A}$ with a source voltage of 0.35V . A gate voltage of 0.8V results in a v_{gs} of 0.45V , and so a proper v_{ds} is needed to be in saturation. A v_{ds} of 0.3V was chosen, which results in a v_{th} of 0.355V . The overdrive voltage $v_{gs} - v_{th} = 0.095\text{V}$, which is much less than the v_{ds} of 0.3V , and therefore the switching transistors are in saturation.

4.1.3 Cascode and Supply Design

As the DAC is a higher resolution, the extra cascode transistor above the switching pairs were added. Their size was also set to $80\text{nm}/28\text{nm}$ to be consistent with the switching pairs, but their biases and headroom must account for the “keep-alive” current they draw. Instead of having to support just $4.82\mu\text{A}$, they must also support the additional DC current that will be drawn at all times. The amount of keep-alive current should be determined by dynamic performance, as the technique is intended to improve that. So, 5 different values of current I_{KA} were designed for and tested: $4.82\mu\text{A}$, $2.41\mu\text{A}$, $1.205\mu\text{A}$, 602.5nA , and 301.25nA . These values correspond to the amount of current burned in both sides of the DAC, so actual current draw is twice the value listed. From these values, the body voltage v_b and v_g can be found for the cascode transistors, as well as the overall supply voltage V_{DD} of the DAC. The values of the bias voltages for the cascode transistors and the DAC V_{DD} are shown in Table 4.1. The dynamic performance of the tested keep-alive current values are shown in Figure 4.2, which will be discussed later.

I_{KA}	v_b (V)	v_g (V)	V_{DD} (V)
4.82 μ A	1	1.07	1.94
2.41 μ A	1	1.05	1.69
1.205 μ A	1	1.03	1.566
602.5nA	1	1.02	1.505
301.25nA	1	1.02	1.474

Table 4.1: Bias voltage values for M_4 and M_5 transistors in Figure 4.1 and corresponding DAC V_{DD} values

From simulation, it was immediately clear that a fully binary 12-bit DAC would not meet the static nonlinearity requirements, and therefore segmentation was necessary. Although the static nonlinearity requirement of INL and DNL >0.5 LSB was met with just 9 LSBs of binary weighting and 3 MSBs of thermometer coding, 3 more segmentation ratios were tested for dynamic performance benefits, shown in Figure 4.2.

4.1.4 Segmentation and Keep-Alive Current Choice

From the data in Figure 4.2, the segmentation ratio and keep-alive current value was chosen together. Dynamic performance data for this plot was taken for a design with no mismatch or calibration, and the input was a dithered fullscale sinusoid. SFDR performance at a low input frequency of 100MHz is generally the same for all segmentation ratios and keep-alive

current values, so it is not useful for selecting design parameters. Looking at SFDR and IM3 at a frequency closer to Nyquist of 9GHz reveals more information. From the IM3, we see that the 9bit binary 3bit thermo segmentation sees the worst performance at current values of 1.205uA and lower, while lower ratios of segmentation perform similarly. From the Nyquist SFDR, 9bit binary 3bit thermo and 8bit binary 4bit thermo segmentation ratios perform worse than the remaining 2 ratios at current values 1.205μA and lower. Between the 2 plots, 8bit binary 4bit thermo segmentation was chosen as a middle ground between the 2 ratios around it. As for the keep-alive current value, 1.205μA was chosen as a compromise between the diminishing returns seen in the IM3 plot and saving power, as 1.205μA keep-alive current already means that it accounts for half the DAC's fullscale current already.

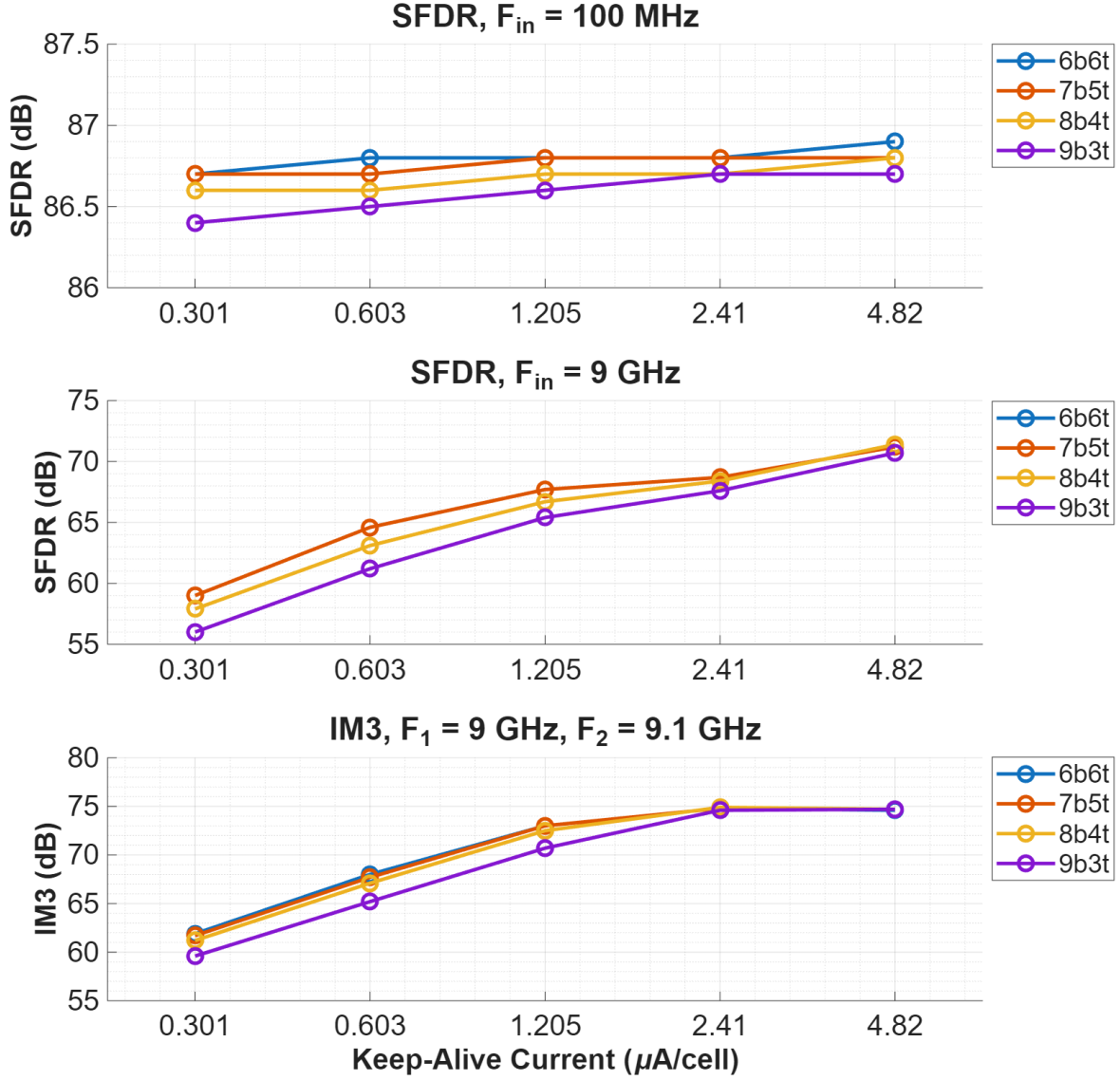


Figure 4.2: Dynamic performance of the 12-bit DAC design at 20GS/s for different segmentation ratios and active power dissipation.

4.1.5 DAC Core Design Summary

All sizes and bias values of the DAC design shown in Figure 4.1 are listed on Table 4.2 and 4.3 respectively. Initial CTT calibration parameters that were determined during DAC core design are listed on Table 4.4.

Device	Size
M1	80nm/215nm
M2-M5	80nm/28nm

Table 4.2: Sizes of transistors in Figure 4.1.

Bias	Value
M1 gate bias	605mV
$D/Dbar$ (M2-M3 gate bias)	800mV/200mV
V_c (M4-M5 gate bias)	1.03V
M4-M5 body bias v_b	1V
V_{DD}	1.56V
I_{ka} (total)	10 mA

Table 4.3: Bias values for DAC design in Figure 4.1.

Parameter	Value
Programming resolution	3
v_{th} shift range	100mV
Maximum/high bias	655mV
Midscale bias	605mV

Table 4.4: Initial CTT calibration parameters for DAC design in Figure 4.1.

4.2 CTT-based Calibration Design

To determine the number of CTTs needed in each bit slice, device matching statistics and concrete CTT data is needed, which is shown in Table 4.4. By following the math in Section 3.2, the number of CTTs needed in each bit slice can be calculated and shown in Table 4.5.

Parameter	Value
$\sigma_{\Delta I}$	379nA
c_I	4.01 μ A
LSB_{prg}	502nA

Table 4.5: Device statistics and CTT calibration parameters.

Bit number	Total cells	K_0	K_1	K_2	Total CTTs
1	1	0	1	0	1
2	2	1	1	0	1
3	4	2	2	0	2
4	8	6	2	0	2
5	16	13	3	0	3
6	32	28	4	0	4
7	64	59	5	0	5
8	128	121	7	0	7
9-12 (thermo-coded, 15 slices)	256	244	10	2	12
Total	4095				205

Table 4.6: DAC design number of CTTs per set for each bit slice.

One issue arose with the calibration that caused unacceptable error, shown in Figure 4.3. When calibrating the DAC with the given CTT numbers in Table 4.4, there exists large DNL spikes 15 times periodically in the ramp. These codes directly translate to the input codes where a new thermometer slice is turned on, of which there are 15 of. These spikes are very high, and some even exceeding 0.5LSBs.

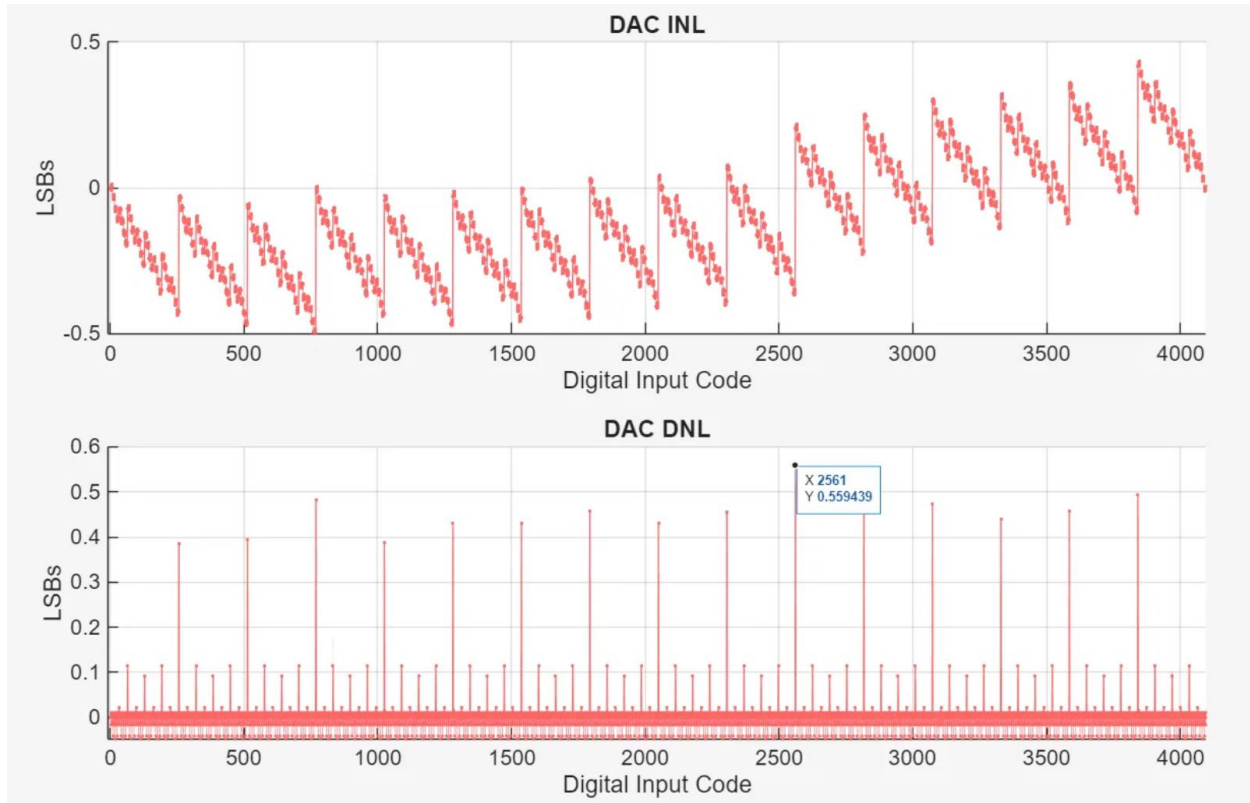


Figure 4.3: DNL spike above 0.5LSBs for initial calibration scheme.

After some investigation, the issue was found to be in the 7th and 8th bits, the last 2 binary-weighted bits before thermometer coding happens. As seen from Table 4.4, the 7th and 8th bit terminate in 5 and 7 CTTs respectively, which by the calculations in Section 3.2, should have a lower error than 0.5LSB per slice. However, this accuracy requirement doesn't consider the combined total error every terminating set of CTTs in every single bit slice. Because of this, although the 7th and 8th bit slice do not contribute more than 0.5LSB of error, they have the potential to contribute around 0.1-0.2LSBs of error because their weighting is much higher than the other binary slices, and they also terminate in the greatest number of CTTs in the design.

This causes the total DAC error to jump above 0.5LSBs of error when adding the error of all other bit slices.

One solution is to simply tighten the accuracy condition per bit cell, so that each cell contributes even less error. If the accuracy requirement is halved from 0.5LSBs to 0.25LSBs, the number of CTTs that can terminate the last set reduces, meaning less error in affected slices. For this design, the only 2 cells that were affected by this accuracy requirement change were the 7th and 8th bit, which now have 1 CTT in their K_2 set and subtract 1 device from their K_0 set to compensate. Thus, those bit slices now terminate in just a single CTT, greatly reducing their error. Table 4.6 shows the new number of CTTs.

Bit number	Total cells	K_0	K_1	K_2	Total CTTs
1	1	0	1	0	1
2	2	1	1	0	1
3	4	2	2	0	2
4	8	6	2	0	2
5	16	13	3	0	3
6	32	28	4	0	4
7	64	58	5	1	6
8	128	120	7	1	8
9-12 (thermo-coded, 15 slices)	256	244	10	2	12
Total	4095				207

Table 4.7: Revised number of CTTs per set

4.3 Digital and Calibration Circuitry Design

All the design of level shifters and digital circuitry supporting the DAC core was implemented with ideal components and in Verilog-A. The calibration circuitry was implemented as a SAR loop in Verilog-A to converge within $1\mu\text{V}$ of the ideal calibration value.

In a chip, the high-voltage level shifters and calibration circuitry required to program the CTTs take up a majority of the $75\mu\text{m} \times 300\mu\text{m}$ area of the DAC design in [6], with the level shifters taking up around half the area themselves of around 0.01mm^2 . However, as described in [6], the level shifters took up significant area due to the design being fully thermometric, and a

segmented DAC design would significantly reduce the number of level shifters needed. Each set of CTTs can be programmed as a group to reduce the number of level shifters, meaning that for the designed DAC with CTT numbers described in Table 4.7, 41 level shifters are required. If the same level shifter circuitry is used, this design would use 16% of the area for level shifters than [6], resulting in around $1600\mu\text{m}^2$.

The DAC core area is estimated to be around $4600\mu\text{m}^2$, using the $16\mu\text{m}^2$ reported area of the CTT DAC core in [6] as a scaling reference. While the area of level shifters was reduced, the total area of the calibration circuitry would still be well over 2 times the estimated DAC core area. However, the calibration circuitry can be shared amongst multiple DAC cores. So, if an array of DACs are calibrated using the same circuitry, the cost of the calibration hardware can be amortized across the array.

Uniform random numbers were generated per instance of a run and are applied the gate bias of the CTTs after calibration to model the programming quantization error.

CHAPTER 5

Results and Conclusions

5.1 Simulation Results

Figure 5.1 shows the INL and DNL results of the DAC design for a 200 point Monte Carlo simulation. In each point, the transistors exhibit mismatch for single sided 3σ variation, or double sided 6σ variation, and are calibrated. A ramp was applied to both a calibrated and non-calibrated DAC to compare the results. Non-calibrated data appears as gray and calibrated data appears as red in Figure 5.1. The black dotted line shows the $\pm 0.5\text{LSB}$ boundary that defines the accuracy condition.

Although not apparent from Figure 5.1 due to the density of data, there is exactly one outlier in the data, where the INL dips below -0.5LSBs . The INL of that data point is shown in Figure 5.2, which reveals that between digital codes 3500 and 4000, the INL drops slightly below -0.5LSBs .

To analyze the cause of this error we can look at the calibrated v_{th} shift values that were determined for that run, which are displayed in Table 5.1. Bits 1-6 don't have any set K_2 CTTs, so they don't have a value for K_2 calibrated values. A negative INL means that the actual DAC output is lower than the straight line through the endpoints. There is a sharp jump from the negative INL to a value near 0LSBs , implying that there's a big jump in output voltage at that code. The cause of this big jump is from a properly calibrated set of bit slices switching to a set

of bit slices whose current is not calibrated enough, or the current is too high and causing too much voltage difference from the last code. In this specific case, the switch to the 23rd thermo bit is causing this because that bit slice's calibrated v_{th} shift is too low and needs to be higher.

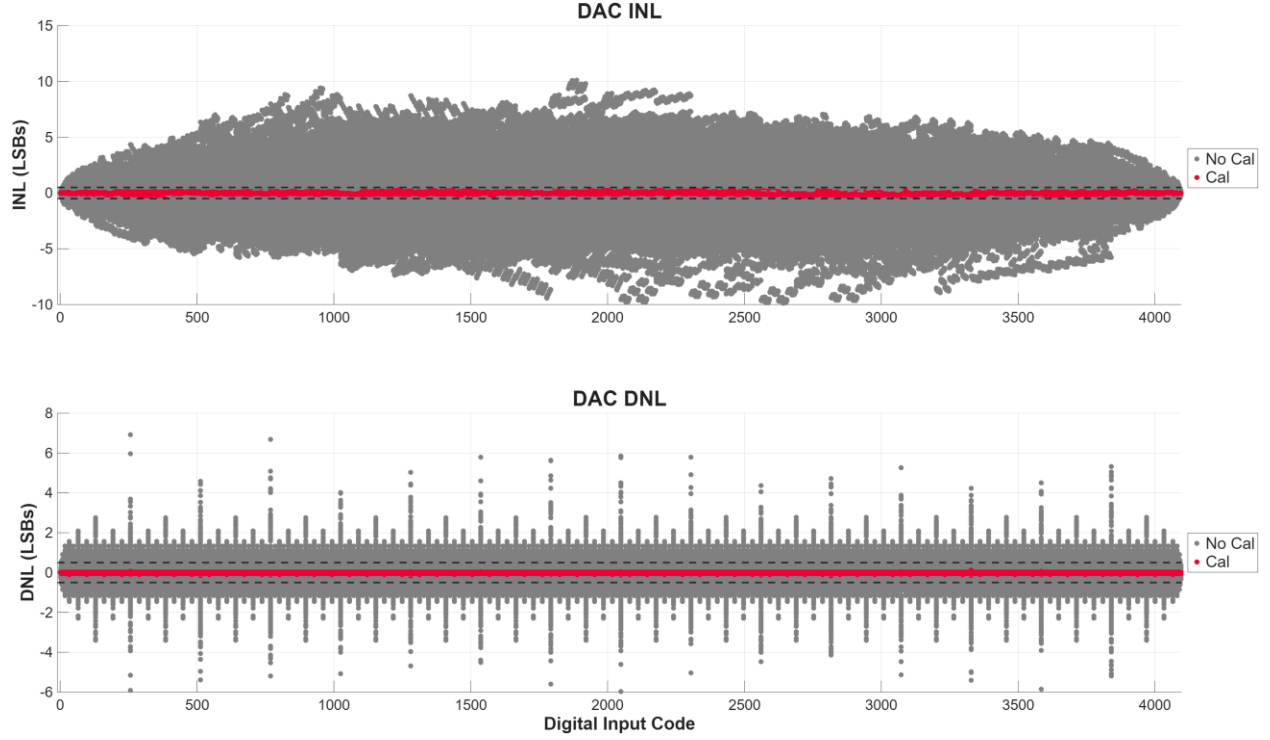


Figure 5.1: INL and DNL plots of the DAC design for 200 Monte Carlo points.

From the values, we see that the calibrated value of the 23rd bit, the last bit, calibrated to a value of 100.1mV, which is essentially the maximum v_{th} shift that the calibration was designed for. The SAR calibration loop was designed for 100mV of calibration range, with the first step of the SAR algorithm being 50mV and halving each succeeding cycle until reset. This means that the calibration can basically only increase the v_{th} to the designed maximum range of 100mV, even if the desired v_{th} is higher. So, as shown in Figure 5.3, the calibration value continuously tries to increase in value, but can't because the step size halves every cycle. The negative INL

magnitude isn't that much higher than 0.5LSBs, so we can infer that the 23rd bit probably just needed a little more v_{th} shift to be calibrated properly, maybe 1-2mV.

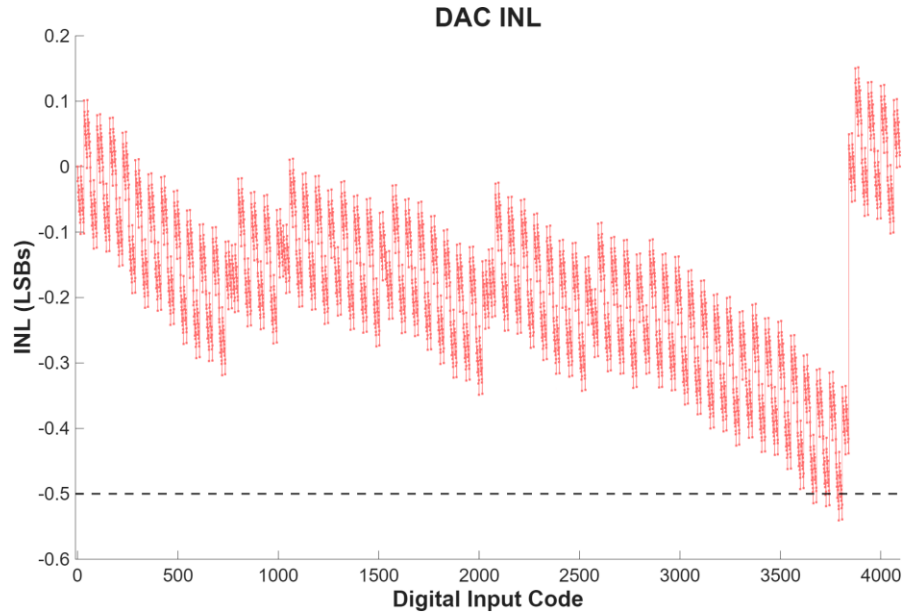


Figure 5.2: One outlier from 200 point Monte Carlo simulation.

Bit number	K_1 values (mV)	K_2 values (mV)
1	41.81	-
2	39.95	-
3	35.64	-
4	47.94	-
5	44.32	-
6	55.16	-
7	53.58	64.72
8	53.58	59.94

9 (thermo)	37.72	40.39
10 (thermo)	40.17	42.92
11 (thermo)	64.11	47.29
12 (thermo)	34.75	44.72
13 (thermo)	62.21	63.77
14 (thermo)	25.7	64.43
15 (thermo)	23.98	44.95
16 (thermo)	17.25	54.26
17 (thermo)	17.69	36.96
18 (thermo)	29.04	40.5
19 (thermo)	35.41	66.02
20 (thermo)	32.91	62.75
21 (thermo)	19.91	53.5
22 (thermo)	51.2	49.79
23 (thermo)	100.1	100.1

Table 5.1: Calibrated v_{th} shift values for the outlier run

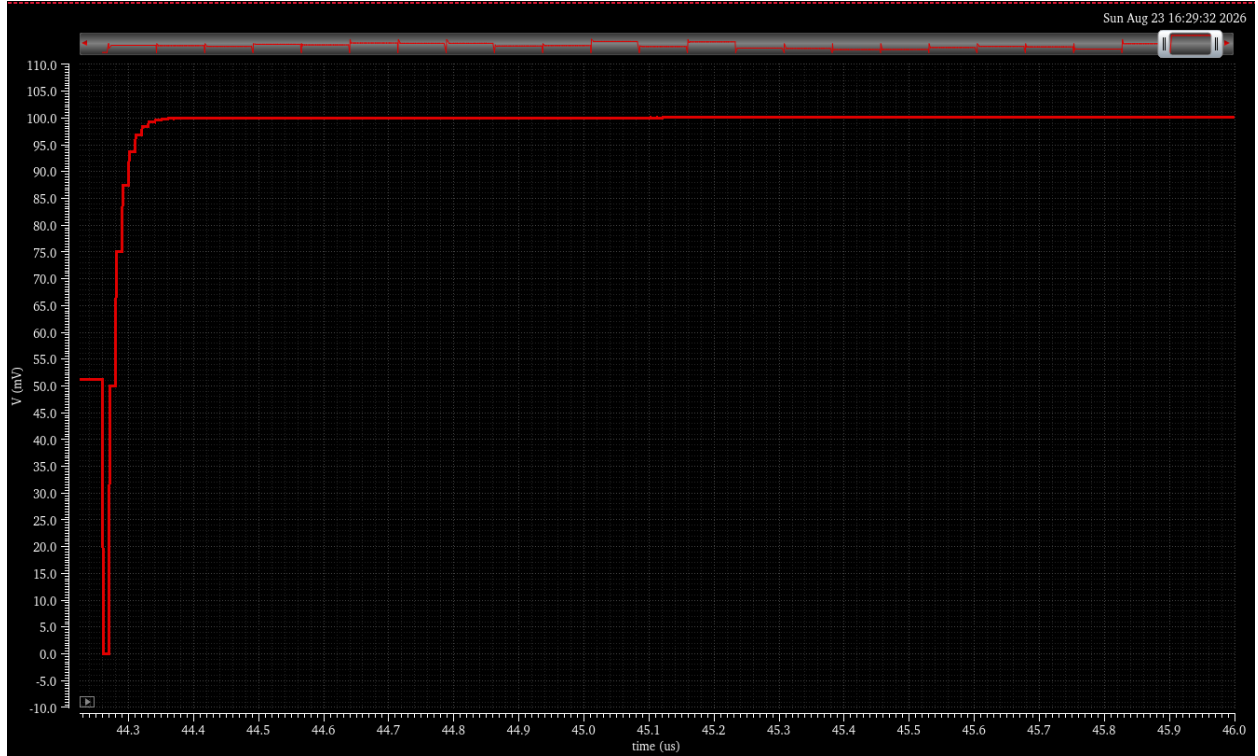


Figure 5.3: SAR convergence of calibration for the 23rd bit.

This issue arises from the fact that despite being in strong inversion, the midscale current curve in Figure 3.2(a) is not perfectly centered between the endpoint current curves, which is also implied by the valley-shaped INL curve in Figure 3.2(b). As a result, for the same v_{th} shift in either the positive or negative direction from the midscale does not result in the same magnitude shift in current because of changes in inversion level. Specifically, a positive v_{th} shift from midscale lowers the current and a negative v_{th} shift from midscale increases current, and the magnitude of current shift from a positive v_{th} shift is lower than a negative v_{th} shift. So, a larger positive v_{th} shift is required to achieve an equal magnitude of current shift from a negative v_{th} shift.

Since mismatch statistics are equal on both sides of the mean value, there is a small section of the positive side that is not accounted for due to the non-centered midscale current of CTT programming. However, this is not a huge problem as this exclusively affects positive v_{th} shifts. A little extra positive v_{th} shift is always possible unlike a negative v_{th} shift from 0mV, which is impossible. Therefore, this problem can be accounted for by allowing the calibration loop to be able to program a CTT to a v_{th} shift slightly above the maximum range designed for. For future endeavors, this error should be analyzed and accounted for to see exactly how much extra v_{th} shift range should be allowed. For this dataset, since the calibration was designed to never exceed 100.1mV, it is an outlier that doesn't meet the accuracy requirement we set. A few other runs exhibit the same issue of being bounded by the 100.1mV limit on different bit slices, but only this one outlier shows error that violates the accuracy requirement.

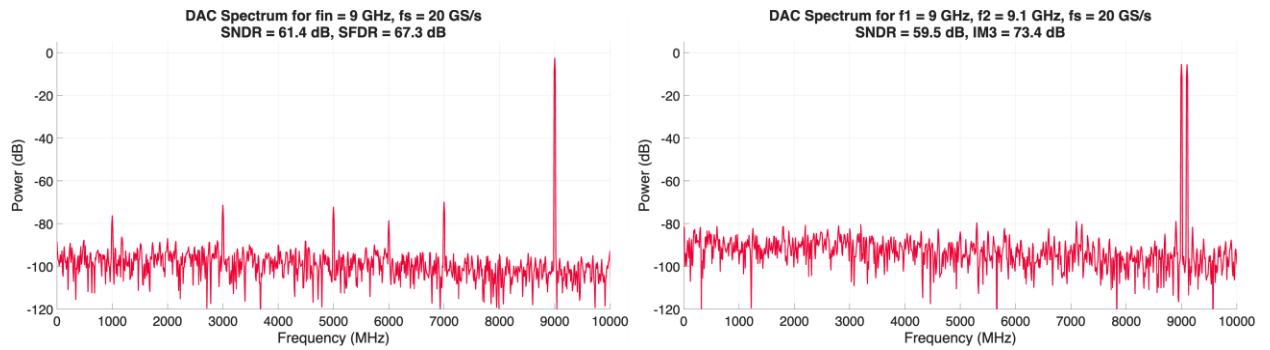


Figure 5.4: Frequency spectrum plots of the DAC design at 20GS/s. a) single tone 9GHz input, SFDR = 67.3dB. b) two tone 9GHz and 9.1GHz input, IM3 = 73.4dB

Figure 5.4 shows the simulated frequency spectrum plots of the calibrated DAC output for 2 input cases. Figure 5.4(a) shows the output spectrum for an input of a single tone digitized sinusoid at 9GHz with 1LSB dither. Figure 5.4(b) shows the output spectrum for an input of a

two tone digitized sinusoid at 9GHz and 9.1GHz with 1LSB dither. Each output spectrum was taken from 100ns of output data.

Figure 5.4(a) shows the harmonics of the input frequency folding back into the Nyquist band: the 3rd harmonic of 27GHz folding back to 7GHz, the 5th harmonic of 45GHz folding back to 5GHz, and the 7th harmonic of 63GHz folding back to 3GHz. Since the DAC design is differential, even order harmonics are suppressed. There are 2 lower magnitude spurs at 1GHz and 6GHz, which do not correspond to any significant odd-order harmonics, that may be a result of other imperfections such as imperfect calibration. From Figure 5.4(b), we see that the IM3 of the two tone input is at the same level as the SNR of a quantized sinusoid with an effective number of bits of 12, which is around 72dB.

5.2 Conclusion and Future Work

This thesis presents a calibration structure using CTTs to minimize on-path calibration circuitry involvement and subDAC size, as well as a method of minimizing the number of CTTs needed in multi-step calibration. The system was modeled and designed in Global Foundries 22nm, and simulated nonlinearity results are shown.

Future work can initially focus on accounting for the error of all terminating CTT errors combined to meet the accuracy requirement, as well as the extra v_{th} shift range that may be needed. Since the calibration technique itself was very simple and based off conventional calibration algorithms, adopting different calibration techniques to the same general structure can be done to see if CTTs work well with a variety of algorithms, such as INL-based calibration in [10] and the 2-parameter temperature tracking calibration in [11]. Since the CTT calibration

structure was proposed to improve dynamic performance and timing mismatch, area comparisons should be made with calibration structures from prior art, and the dynamic performance of calibration techniques and structures from prior art should be tested on silicon with CTT calibration to truly verify dynamic performance benefits in a realistic setting rather than just simulation. In particular, multiple DACs should be placed in an array and calibrated using the same circuitry to determine if it is worth the large area cost of the calibration circuitry. Lastly, CTT charge trapping retention and drift over time should be explored further to determine its effects on DAC performance, and its potential solutions.

REFERENCES

- [1] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching properties of MOS transistors," in *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1439, Oct. 1989.
- [2] C. -H. Lin *et al.*, "A 12 bit 2.9 GS/s DAC With IM3 $\ll$ -60 dBc Beyond 1 GHz in 65 nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3285-3293, Dec. 2009.
- [3] Yonghua Cong and R. L. Geiger, "A 1.5-V 14-bit 100-MS/s self-calibrated DAC," in *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2051-2060, Dec. 2003.
- [4] D. A. Mercer, "Low-Power Approaches to High-Speed Current-Steering Digital-to-Analog Converters in 0.18- μ m CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 42, no. 8, pp. 1688-1698, Aug. 2007.
- [5] S. Qiao, S. Wang, J. Zhu, Z. Guo, M. Zeinali, V. K. Jacob, S. Chakrabarty, B. Vaisband, S. S. Iyer, and S. Pamarti, "Back-Gate Mediated VTH Mismatch Reduction to Enable 6-bit Precision in FD-SOI Charge-Trap Transistors," in *Proc. Government Microcircuit Applications & Critical Technology Conf. (GOMACTech)*, Pasadena, CA, USA, Mar. 2025.
- [6] M. Zeinali and S. Pamarti, "On-Chip Charge-Trap-Transistor-Based Mismatch Calibration of an 8-Bit Thermometer Current-Source DAC," in *IEEE Solid-State Circuits Letters*, vol. 9, pp. 53-56, 2026.
- [7] P. Andricciola and H. P. Tuinhout, "The Temperature Dependence of Mismatch in Deep-Submicrometer Bulk MOSFETs," in *IEEE Electron Device Letters*, vol. 30, no. 6, pp. 690-692, June 2009.

- [8] W. -H. Tseng *et al.*, "17.3 A 14b 16GS/s Time-Interleaving Direct-RF Synthesis DAC with T-DEM Achieving -70dBc IM3 up to 7.8GHz in 7nm," *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2023.
- [9] H. Luo *et al.*, "11.8 A 14b 20GS/s RF-Sampling DAC Achieving 70.4dBc IMD3 Up to 8.9GHz," *2026 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2026.
- [10] C. -H. Lin *et al.*, "A 16b 6GS/S Nyquist DAC with IMD <-90dBc up to 1.9GHz in 16nm CMOS," *2018 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2018.
- [11] H. Zhu, W. Yang, G. Engel and Y. -B. Kim, "A Two-Parameter Calibration Technique Tracking Temperature Variations for Current Source Mismatch," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 387-391, April 2017.
- [12] T. C. Carusone, D. A. Johns, and K. W. Martin, *Analog Integrated Circuit Design*, 2nd ed. New York, NY, USA: Wiley, 2012.
- [13] B. Razavi, *Analysis and Design of Data Converters*. Cambridge: Cambridge University Press, 2025.