Resistor-Based Digital-to-Analog Converters

by

Maciej Nawrocki

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 27th November, 2023 at 15:00.

Student number: 5496713
Thesis committee: Prof. dr. Klaas Bult, TU Delft, Ethernovia supervisor
Dr. Masoud Babaie, TU Delft
Dr. Fabio Sebastiano, TU Delft

This thesis is confidential and cannot be made public until 27.11.2025

TU Delft

Acknowledgements

I would like to begin by expressing my deepest gratitude to my supervisor, Professor Klaas Bult. I remember the moment when he first suggested the topic of my project. At first, I thought, ”Resistors and switches, how hard could it be?” However, as I delved deeper into the project, I realized that constructing a DAC was far more challenging than it seemed. Many times, there were things I could not quite grasp, but Klaas was able to explain them in a single sentence. I always looked forward to our weekly meetings, knowing that I would come out with a fresh perspective and new knowledge. Klaas, it has been a pleasure working with you, and I am very excited to continue doing so at Ethernovia.

I am also very thankful to my daily supervisor, Burak, who spent a great deal of time walking me through the fundamentals of data converters. Many times, when I could not progress, he showed me the correct path to take.

I would like to thank the entire Ethernovia team for making me feel welcome and motivated daily. The atmosphere in the office was unparalleled, and our enjoyable lunches and non-technical conversations helped me stay positive and overcome any challenges that came my way. I especially want to thank Els for making moving to Utrecht very smooth and for all her help whenever Dutch proficiency was needed.

In the last two years, I have met amazing people. My dear friends, this paragraph is for you. I want to thank you for all the fantastic moments we had together. We should definitely have more of these in the future.

I also want to thank my parents, Janusz and Marta, for making it possible for me to study in Delft and for their continuous support. If it were not for you, I would never be able to come this far. My success is really your success.

And to my better half, Sabinka, I am very fortunate to have you by my side.

Maciej Nawrocki
Utrecht, November 2023

Contents

List of Figures

1

Introduction

1.1. Motivation

Transmit digital-to-analog converters have become an essential building block for state-of-the-art Ethernet infrastructures. They are also one of the most significant sources of power consumption in an Ethernet physical layer (PHY) transceiver. These devices must maintain high linearity and a well-defined impedance of 100 Ω while operating at a speed of several GS/s. Traditionally, current-steering digital-to-analog converters (DACs) were considered the 'go-to' solutions for high-frequency and high-linearity applications despite having fundamental power efficiency limitations. In recent years, a voltage-mode resistive DAC architecture has gained more popularity. Resistive DACs are inherently more power-efficient thanks to their class-B mode of operation (delivering all available current to the load). Moreover, thanks to their simple design, they do not suffer from headroom limitations and are a natural choice given ongoing technology scaling toward lower supply voltage. This thesis aims to investigate the promising resistive DAC architecture.

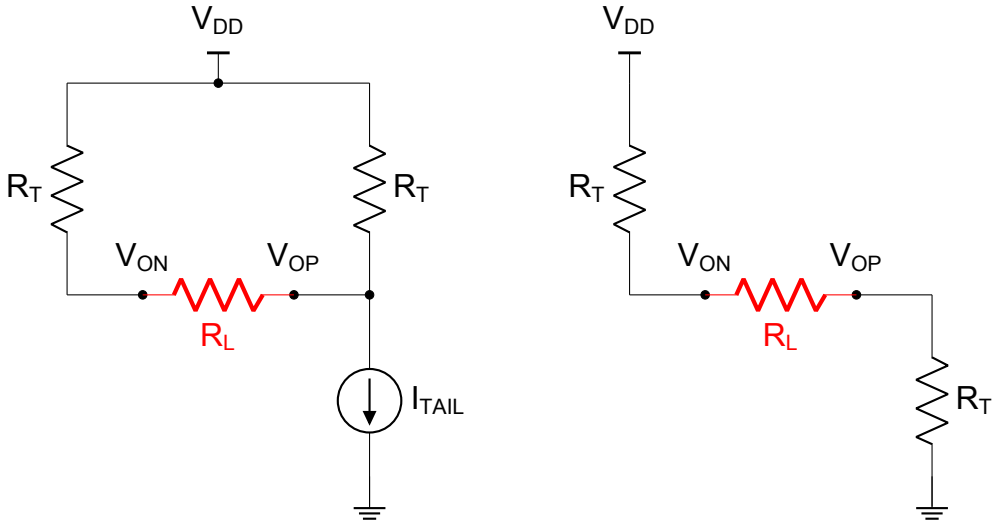
1.2. Thesis structure

The thesis is comprised of six chapters. The first chapter will briefly compare current-steering and resistive DAC converters, with a focus on their power efficiency and design issues. In the second chapter, the basic working principle of the resistive DAC will be explained, along with its primary design challenges. Additionally, the state-of-the-art will be described. The third chapter will cover the step-by-step design process, including the proposed compensation technique, segmentation, and solutions for timing inaccuracy. The fourth chapter will discuss the layout, and in the fifth chapter, simulation results will be presented. Finally, the sixth chapter will provide conclusions and discuss future work.

1.3. Current-Steering vs Resistive DACs comparison

Two main DAC architectures are typically used for high-speed wireline applications: Current-Steering and Resistive DACs. The former is the most widely used, while the latter has gained popularity in recent years [1, 2, 3, 4].

When comparing two implementations, one of the most important factors that can be evaluated is their power efficiency. Figure 1.1 shows current-steering and resistive DACs fully tilted to one side, that is, when they produce maximum output swing.



(a) Current-steering DAC **(b)** Voltage-mode Resistive DAC

Figure 1.1: Circuit equivalents of fully tilted DACs

Efficient power transmission requires that the impedance of the converter matches the impedance of the channel. For example, if the channel impedance R_L is equal to 100Ω, then the value of each termination resistor R_T must be equal to $R_L/2$, which is 50Ω.

For the current steering, when it is tilted to one side, the current that reaches the load is equivalent to 1/4 of the tail current. The maximum theoretical voltage across the channel can also be calculated as $2/3V_{DD}(1-\alpha)$, where α represents the minimum voltage across the tail current source. To express the voltage, current, and power efficiency of the current-steering architecture, the following formulas can be used:

$$\eta_{vol} = \frac{V_{LOAD}}{V_{DD}} = \frac{2}{3}(1-\alpha) \tag{1.1}$$

$$\eta_{cur} = \frac{I_{LOAD}}{I_{TAIL}} = \frac{1}{4} \tag{1.2}$$

$$\eta_{pwr} = \eta_{vol} \cdot \eta_{cur} = \frac{1}{6}(1-\alpha) \tag{1.3}$$

In the second architecture (b), the resistors form a single path from the supply to the ground. This delivers all available current ($V_{DD}/4R_L$) to the load, resulting in 100%

current efficiency. However, only half of the supply voltage can be sensed across the channel. As in the previous case, we calculate the respective efficiencies:

$$\eta_{\text{vol}} = \frac{1}{2} \tag{1.4}$$

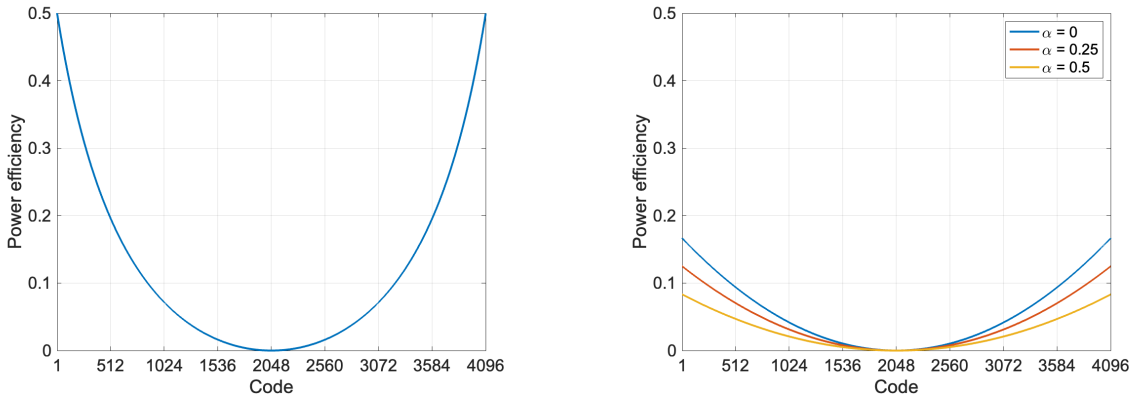
$$\eta_{\text{cur}} = 1 \tag{1.5}$$

$$\eta_{\text{pwr}} = \eta_{\text{vol}} \cdot \eta_{\text{cur}} = \frac{1}{2} \tag{1.6}$$

Relative power efficiency (resistive/current) can be calculated in the following way:

$$\eta_{\text{RDAC}} / \eta_{\text{IDAC}} = \frac{3}{(1-\alpha)} \tag{1.7}$$

In practice, α can be as high or even higher than 0.5, making the resistive DAC six times more power efficient. The current-steering architecture is limited in terms of the amount of voltage headroom required for the tail current source to maintain linearity. The comparison of efficiency presented in this section is basic and is limited to a full-swing case. In reality, the average power efficiency depends on the probability distribution of the input signal. The analysis above can be extended by calculating power efficiency for different inputs. In Figure 1.2, power efficiency is plotted vs input code. For the current-steering architecture, two cases are considered, $\alpha = 0$ and $\alpha = 0.25$.



(a) Resistive DAC **(b)** Current-steering DAC

Figure 1.2: Comparison of power efficiency

Assuming a uniformly distributed input signal, even when $\alpha = 0$, the current-steering DAC is still 2.2 times less power efficient when calculating the average power efficiency across all input codes.

Both architectures can also be compared in terms of design challenges. Some design aspects, such as timing accuracy or segmentation, are common with both architectures, and they are covered in Chapter 3.

As referenced by [5], when designing current-steering DACs, output impedance is chosen much larger than R_L. This is obtained by incorporating cascodes and improves linearity for low frequencies. However, at high frequencies, additional small current sources are needed to minimize the amount of 'switchable' output impedance. In contrast, resistive DACs offer the advantage of a constant output impedance, which is not affected by the input code.

The current drawn from the supply differs between the two architectures. Current-steering DACs draw a constant supply current, whereas resistive DACs have a code-dependent supply current that can cause linearity problems, as discussed in Chapter 2.

Regarding chip area, resistive DACs tend to be smaller because they use fewer components. Their size is mainly determined by resistors (matching) and switch drivers (speed). On the other hand, for current-steering DACs, most of the chip area is occupied by the tail-current sources (matching).

Ultimately, the technology scaling moving toward lower supply voltage will benefit resistive DACs and make the design of current-steering DACs more challenging.

1.4. Specifications

Below are the specifications of the test-chip we are aiming for:

- Technology: 0.18 μm
- Resolution: 12-bit
- Speed: 200 MS/s
- Output impedance: 100 Ω (differential)
- INL and DNL: +-0.5 LSB
- IM3 @ low frequency > 75 dB
- IM3 @ Nyquist frequency > 45 dB

Using 0.18 μm technology makes the design and layout relatively easy compared to more advanced nodes. Additionally, 0.18 μm technology is widely available, with low tape-out costs. The current DAC architecture Ethernovia uses is 12-bit. Therefore, for comparison reasons, we aim for the same resolution. Clock speed is limited to 200 MS/s, comparable with similar resolution DACs in the same technology [6]. Increasing clock frequency beyond that would also make it harder to verify the performance of the resulting chip. Most resistive DACs for Ethernet applications do not achieve 12-bit accuracy, and they are mostly limited to 9 or 10 bits [7, 8]. The main objective of this project is to showcase that we can further improve the performance of the resistive DAC beyond the current state-of-the-art, especially in terms of linearity. Given the project's high complexity and only one student working on it, we have decided to go with the simplest solutions possible and take shortcuts where applicable to achieve our desired results in the shortest design time.

2

Resistive DAC Main Design Challenge and State of the Art

2.1. Resistive DAC working principle

A resistive DAC can be viewed as an array of N resistors in which every resistor shares one common output node, and the other node can be connected to the positive or negative reference depending on the input code, as shown in Figure 2.1.

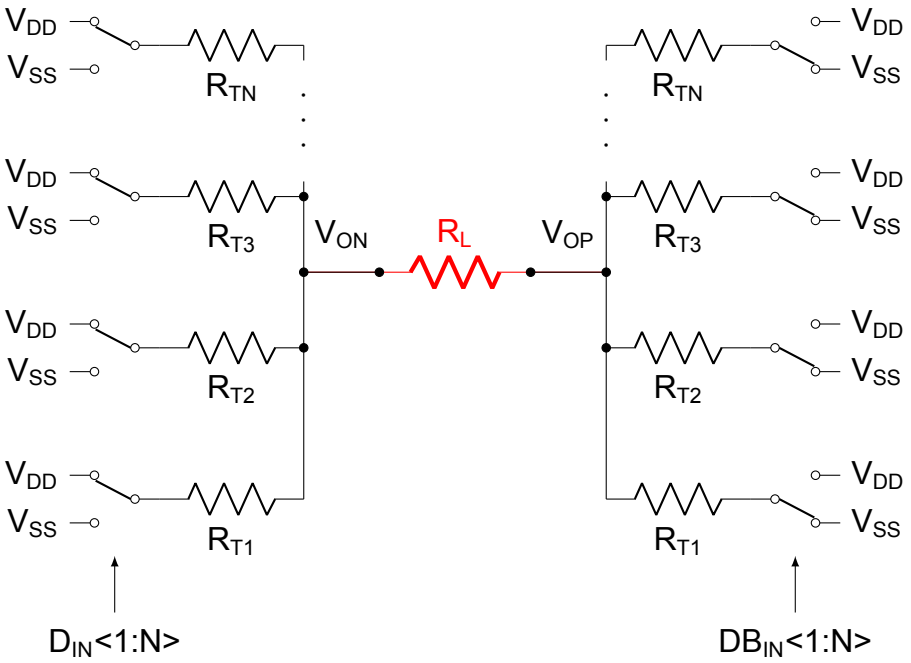


Figure 2.1: Differential implementation of resistive DAC

It is essential to realize that every resistor can be connected either to V_{DD} or V_{SS} but never to both simultaneously. Consequently, for an array of N resistors, each having a value of R_T, the output impedance looking into V_{ON} and V_{OP} nodes remains constant and equal to R_T/N.

2.1.1. Output voltage and supply current derivation

A differential resistive DAC model introduced in Figure 2.2 is used to derive the output voltage and supply current expressions. The number of resistors connected to the positive or negative reference is expressed as equivalent resistance R_A and R_B.

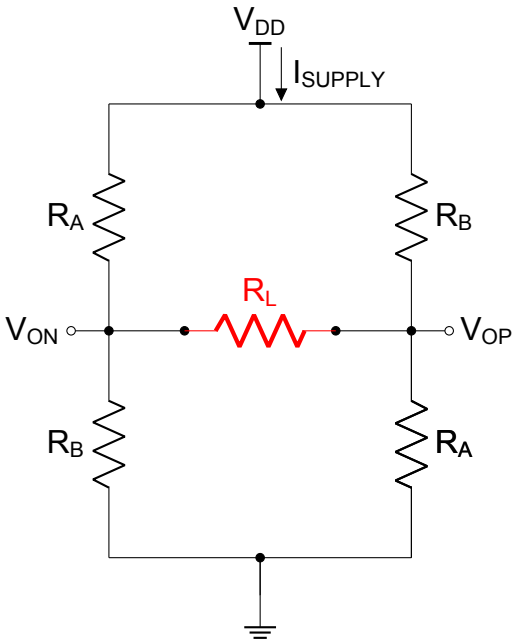


Figure 2.2: Simplified resistive DAC circuit

If an N-bit differential DAC is assumed, the number of units equals $2K = 2^N$. All possible data inputs are D_{IN} = -K,-K+1, ..., K-1, K. Now R_T, R_A and R_B can be expressed in the following way:

$$R_T = R_L K \tag{2.1}$$

$$R_A = \frac{R_T}{K + D_{IN}} \tag{2.2}$$

$$R_B = \frac{R_T}{K - D_{IN}} \tag{2.3}$$

Writing nodal expressions for the output nodes results in a system of two equations:

$$\begin{cases} \frac{V_{ON} - V_{DD}}{R_A} + \frac{V_{ON} - V_{OP}}{R_L} + \frac{V_{ON}}{R_B} = 0 \\ \frac{V_{OP} - V_{DD}}{R_B} + \frac{V_{OP} - V_{ON}}{R_L} + \frac{V_{OP}}{R_A} = 0 \end{cases} \tag{2.4}$$

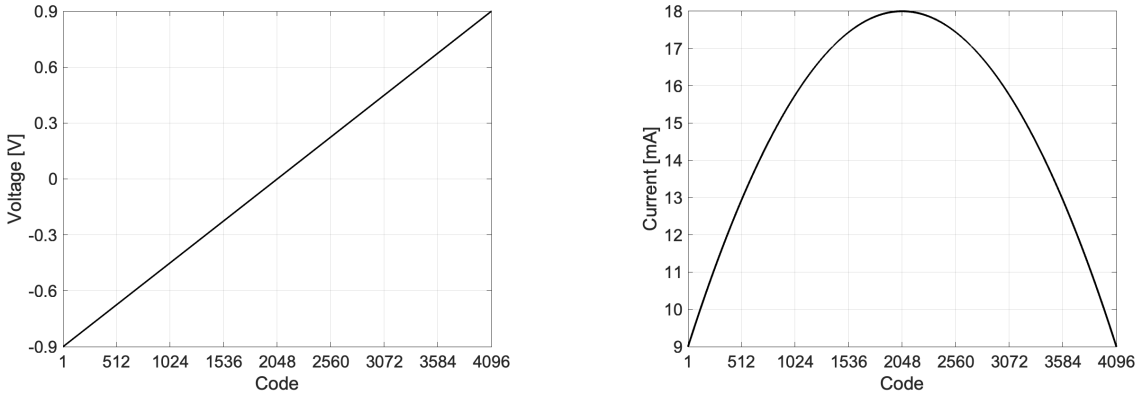
and the expressions of the output voltage and the supply current can be found:

$$\begin{cases} V_{ON} = \dfrac{V_{DD}}{4K}(2k + D_{IN}) \\ V_{OP} = \dfrac{V_{DD}}{4K}(2k - D_{IN}) \end{cases} \tag{2.5}$$

$$V_{OUT} = V_{ON} - V_{OP} = V_{DD}\frac{D_{IN}}{2K} \tag{2.6}$$

$$I_{SUPPLY} = \frac{V_{ON}}{R_B} + \frac{V_{OP}}{R_A} = V_{DD}\frac{-D_{IN}^2 + 2K^2}{2R_L K^2} \Rightarrow I_{SUPPLY} \propto -D_{IN}^2 \tag{2.7}$$

Both the supply current and output voltage for a 12-bit DAC with termination resistors matched to R_L = 100Ω and supply voltage of 1.8 V are plotted in Figure 2.3.



(a) Output voltage **(b)** Supply current

Figure 2.3: Output voltage and supply current across digital input code

The output voltage reaches a maximum (differential) value of 1.8 V, which is equal to V_{DD}. The supply current shows the quadratic dependency on the input code. It is maximum (V_{DD}/R_L) for the middle code and minimum (50% of its maximum value) for the extremes when the DAC is fully tilted to one side. It will soon become clear that the data-dependent supply current is the main problem of the voltage mode architecture.

2.1.2. Finite supply impedance

The analysis from the previous paragraph is repeated in this section with finite supply impedance included. The circuit used to calculate the output voltage is modified by adding two resistors, modeling supply, and ground impedances, as shown in Figure 2.4.

The supply current flowing through R_S will cause a voltage drop, effectively lowering the reference voltage. Moreover, because the supply current is code-dependent, the reference voltage also becomes code-dependent:

$$V_{REF} = REF_+ - REF_- = V_{DD} - 2I_{SUPPLY}R_S \Rightarrow V_{REF} \propto D_{IN}^2 \tag{2.8}$$

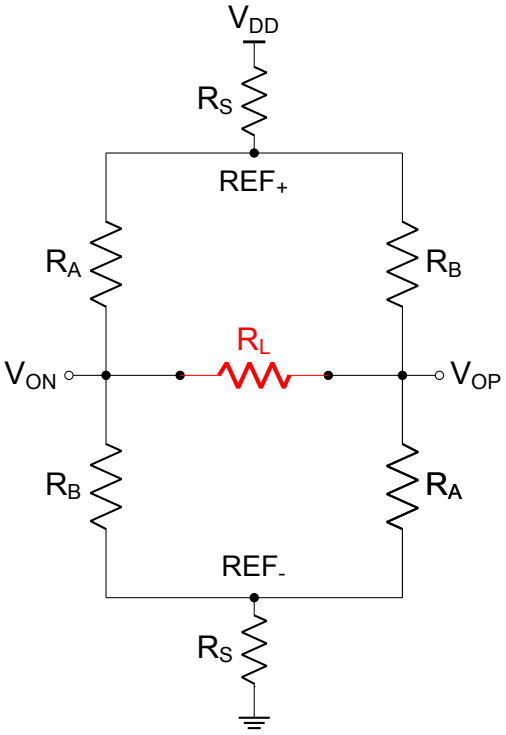


Figure 2.4: Simplified resistive DAC circuit with supply and ground impedance added

This inevitably appears at the output of the DAC and results in distortion. If we assume that the voltage drop across supply impedance is small compared to the output voltage swing, then we can substitute V_{REF} for V_{DD} in Equation 2.6 which gives:

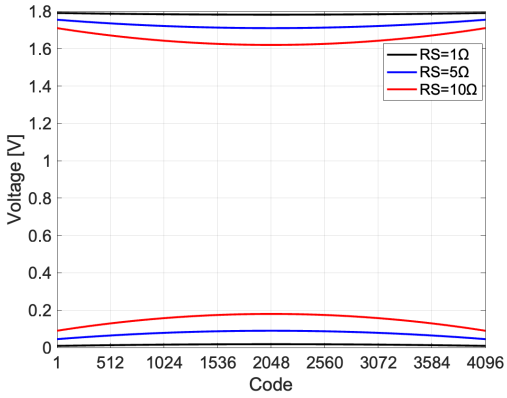
$$V_{OUT} = V_{REF}\frac{D_{IN}}{2K} = D_{IN}V_{DD}\left(\frac{R_L - 2R_S}{2R_LK}\right) + D_{IN}^3V_{DD}\left(\frac{R_S}{2R_LK^3}\right) \tag{2.9}$$

The nonlinearity appears to be purely third-order, and its magnitude can be expressed with the following equation:

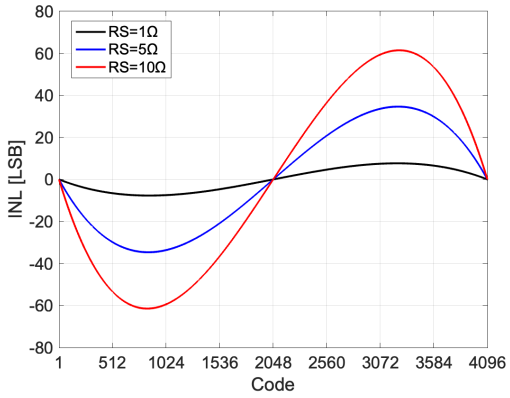
$$HD_3 = \frac{a^2}{4}\frac{R_S}{(R_L - 2R_S)} \approx \frac{a^2}{4}\frac{R_S}{R_L} \text{ assuming } R_S \ll R_L \tag{2.10}$$

where 'a' is the input amplitude ranging from 0 to 1. Noteworthy, the problem becomes more difficult for heavy loads (small R_L) and becomes less troublesome for lighter loads (big R_L). Equation 2.10 assumes that the output impedance and termination are matched perfectly ($R_T = R_L \cdot K$).

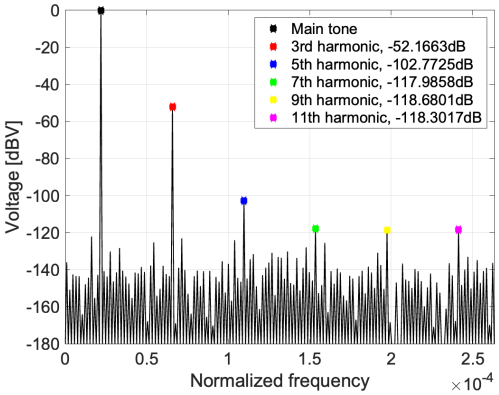
Basic simulations using MATLAB were performed to see the extent of the problem. In Figure 2.5, the right plot shows a 12-bit level integral nonlinearity (INL) for different values of R_S. The left plot illustrates what happens to the positive and negative reference voltages for the same R_S. It is clear that finite supply impedance poses a severe problem because even an R_S as low as 1 Ω results in almost 8 LSB integral nonlinearity error. To keep the INL error between +-0.5 LSB supply impedance needs to be lower than 70 mΩ, which is not possible to realize.



(a) Positive and negative references



(b) Integral nonlinearity



(c) FFT results for RS=1Ω

Figure 2.5: Impact of finite supply impdance

The above analysis assumes relatively small values of R_S relative to R_L. Finite supply impedance results in odd-order distortion components only because the system is symmetrical. The third-order component is by far the most dominant, and analyzing the influence of higher-order components is of limited value. The exact equation for HD_5 and HD_7 are derived in A. The results in this appendix show that $HD_5 = HD_3^2$ which is confirmed by the results in Figure 2.5 (c). As we design for relatively small HD_3 (-75 dB), the assumption that higher harmonics are much smaller is accurate.

2.2. State of the Art solutions

The problem of a code-dependent voltage drop across supply impedance is known and reported in the literature [9]. First and foremost, a careful layout of supply connections must be considered to alleviate this issue. However, it quickly becomes apparent it is challenging to design supply connections with low enough impedance. The standard way to reduce this error is adding buffers for positive and negative references [10]. Placing the buffers as close to the DAC as possible guarantees that the wiring between them does not matter. Nevertheless, designing very low-output impedance buffers is a complex task. If a simple architecture is used, such as a source follower, and channel length modulation is neglected, the output impedance can be approxi-

mated as:

$$R_{OUT} = \frac{1}{g_m} \tag{2.11}$$

where the transconductance is simply:

$$g_m = \frac{2I_D}{V_{GT}} \tag{2.12}$$

Substituting 80mV for V_{GT}, the transistor is placed at the edge of weak inversion [11], which ensures we obtain the best possible gm/I_D ratio. If we choose I_D = 9mA, equal to the DAC's minimum supply current, the resulting output impedance is roughly 4.5Ω, which according to equation 2.10 would result in HD3 of 38dB. The output impedance can be decreased further by implementing more complex buffer architectures such as Super-Source Follower [12]. Let us assume that the circuit in Figure 2.6 is implemented to lower output impedance more.

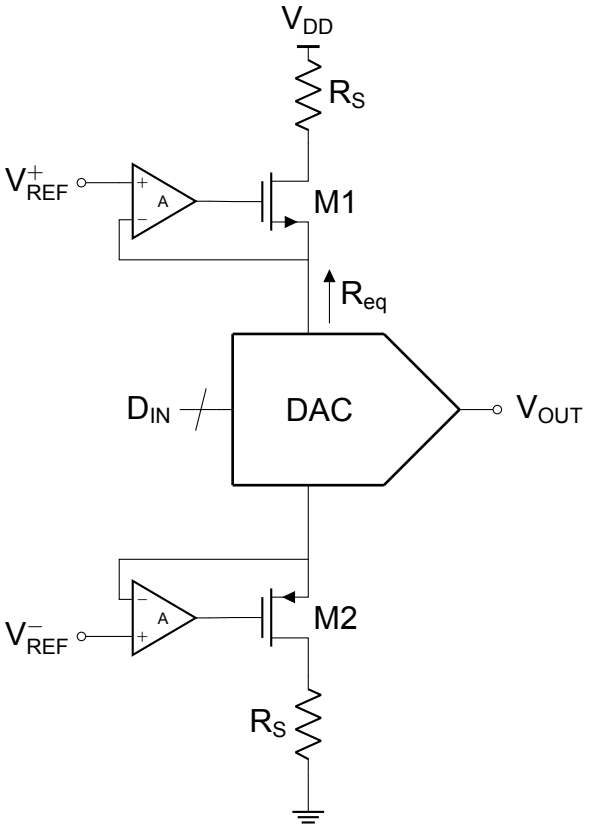


Figure 2.6: DAC circuit with positive and negative reference voltage buffers

Now the equivalent resistance seen at the supply can be calculated as:

$$R_{eq} = \frac{1}{g_m(A+1)} \tag{2.13}$$

It is important to notice that for the buffers to be beneficial, the amplifier has to settle very fast, ideally recovering completely in the DAC's clock period. If the reference buffers can settle back to the required accuracy within one clock period, a system that would sample at the end of the clock period (like a sampling ADC) would obtain results within the specification. However, if the DAC's output is observed continuously (like in the case of Spectrum Analyzer), the reference buffers would have to settle much faster, which would be extremely difficult.

A different solution is proposed in [7] where the fact that switches' on resistance and code-dependent supply current nonlinearities can cancel is used. So far, switches were assumed to be ideal, that is, having zero resistance. In reality, termination resistance, R_T consists of two parts, the resistor and the switch's resistance:

$$R_T = R_{RES} + R_{SW} \tag{2.14}$$

where R_{SW} is equal to:

$$R_{SW} = \frac{L}{\mu C_{ox} W(V_{GS} - V_{TH} - V_{DS}/2)} \tag{2.15}$$

For the extremes, when the DAC is fully tilted to one side, the voltage across the switches (V_{DS}) decreases, and R_{SW} decreases. For the middle code, the voltage across switches is higher, and therefore, the enabled switches experience higher resistance. As the drain to source voltage is output voltage-dependent, it causes distortion. The distortion magnitude is proportional to R_{SW}/R_T. Three cases have been simulated: Ideal switches & finite supply impedance, nonideal switches & zero supply impedance, and nonideal switches & finite supply impedance. The results are shown in Figure 2.7. This figure shows an opposite polarity of the resulting INL for (a) and (b) and a partial cancellation in Figure 2.7 (c).

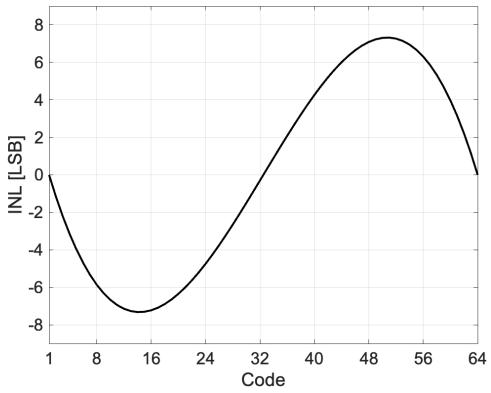
Equation 2.15 can be written as:

$$R_{SW} = \frac{\frac{1}{\beta V_{GT}}}{1 - \frac{V_{DS}}{2V_{GT}}} = \frac{1}{\beta V_{GT}} \cdot \frac{1}{1 - x} \tag{2.16}$$

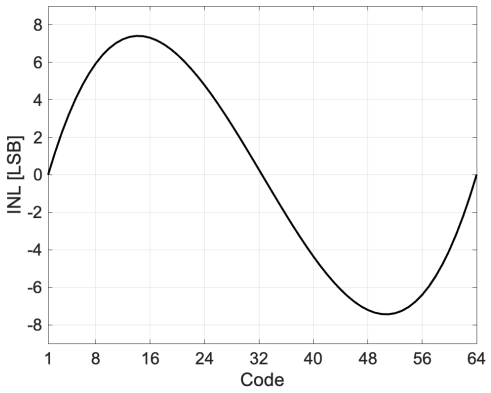
where $\beta = \mu C_{ox} \frac{W}{L}$, $V_{GT} = V_{GS} - V_{TH}$ and $x = \frac{V_{DS}}{2V_{GT}}$. Now, a Taylor series expansion can be applied to 2.16 resulting in:

$$R_{SW} = \frac{1}{4\beta V_{GT}^3} \cdot V_{DS}^2 + \frac{1}{2\beta V_{GT}^2} \cdot V_{DS} + \frac{1}{\beta V_{GT}} \tag{2.17}$$

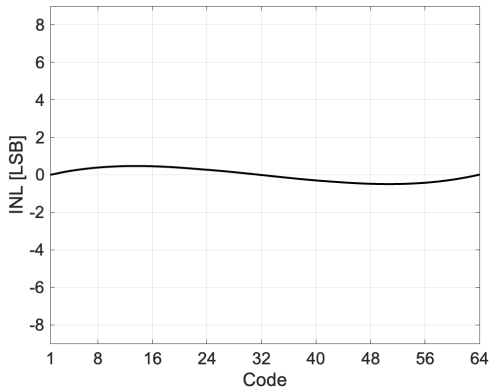
Equation 2.17 shows that the switches' resistance has a positive second-order coefficient vs V_{DS} and consequently input code too. In contrast, the supply current had a negative second-order coefficient versus the input code as derived in Equation 2.7. Therefore, it is possible to carefully size the switches (or vary the supply impedance) to cancel the nonlinearity. Simulations have confirmed that, to cancel the nonlinearity caused by R_S equal to 1 Ω, the switches should be sized so that their resistance is



(a) Ideal switches & finite supply impedance



(b) Zero supply impedance & nonideal switches



(c) Nonideal switches & finite supply impedance

Figure 2.7: INL curves for three cases

approximately $0.25R_T$. This method of nonlinearity cancellation can considerably enhance the converter's speed as smaller switches result in less parasitic capacitance and are easier to drive, reducing dynamic power consumption. However, for perfect cancellation of distortion, the exact value of the impedance of supply and ground networks must be known on forehand. Therefore, this method will work only for one specific value of R_S, and any deviation from it will result in errors. Moreover, the effectiveness of this method depends on the matching of two different physical devices, i.e., resistors and transistors.

This approach will cause the total termination resistance to be dependent on the switch resistance, which is strongly influenced by PVT variations, leading to a termination resistance value that is not well defined. Hence, the method of canceling distortion caused by finite supply impedance by nonlinear switch resistance is prone to rather large distortion spreads due to PVT variations.

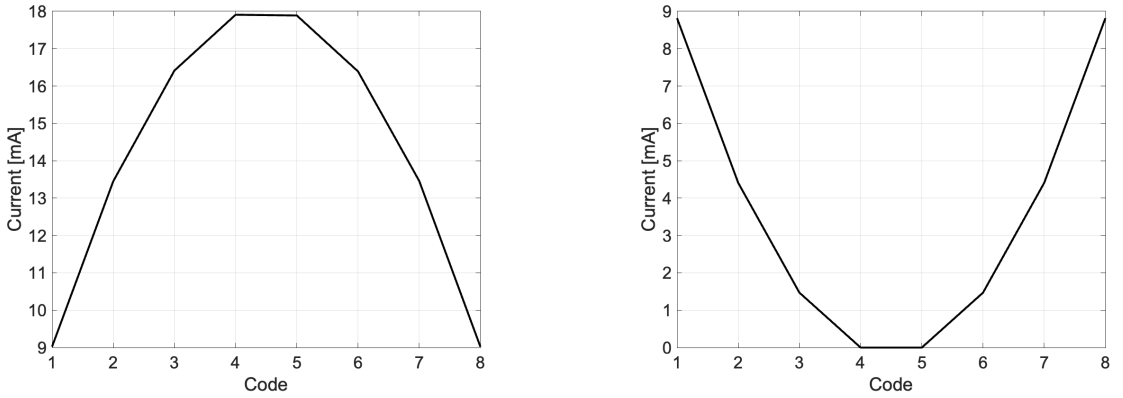
3

Proposed design

3.1. Code dependent supply current compensation

The previous chapter discusses distortion caused by the code-dependent supply current in the case of finite supply impedance. However, if the supply current remains constant for every code, supply impedance will only result in a gain error and not in distortion. In this chapter, we will discuss a method of linearization by pulling an additional current from the supply in order to keep the supply current constant.

In the case of a low-resolution DAC, it is easy to understand the concept. Figure 3.1 (a) shows the supply current of a 3-bit DAC. The graph shows that the current is symmetrical, and it can have four different values. To maintain a constant current, only three levels (Code 1 & 8, 2 & 7, and 3 & 6) require correction. No correction is needed for the two middle codes. If we can add the appropriate additional current (Figure 3.1 (b)), the overall current can be constant. Generally, for an N-bit DAC, $2^{N-1} - 1$ levels must be compensated to achieve constant current.



(a) 3-bit DAC supply current

(b) Compensation current needed to cancel nonlinearity

Figure 3.1: Supply current without compensation and required compensation current

The current needed to linearize the total current from the supply is also quadratic, and it can be produced using an additional component connected in parallel to the

main DAC, which will draw the required current from the supply depending on the input code. The configuration is shown in Figure 3.2.

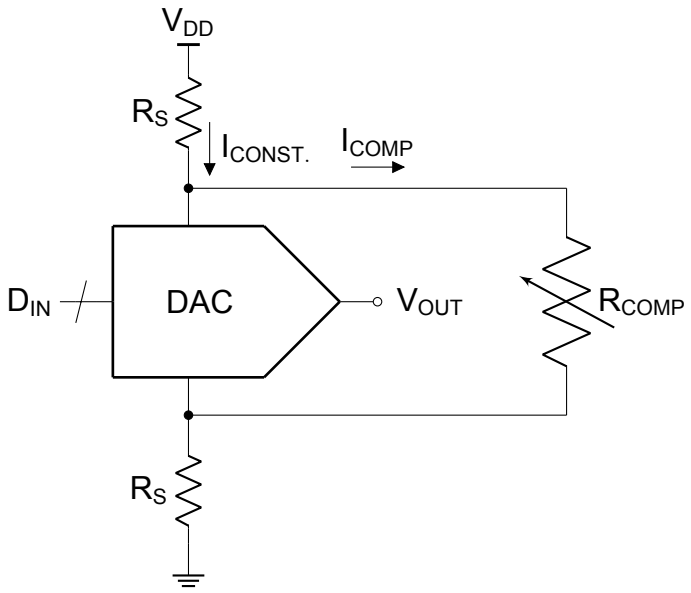


Figure 3.2: Main DAC and Compensation DAC

Although the same idea has already been proposed in [13], [14], and [15], these publications suffer from certain handicaps, respectively: complex digital logic, large resistors not feasible to implement, and power-inefficient compensation. What is more, none of the above addressed layout-related difficulties in implementing the additional compensation element. It may seem like this modification will reduce the power efficiency by half because the current efficiency drops from 100% to 50%. This is only true when we consider extremes. In reality, compensation will result in different power consumption overhead depending on the input signal probability distribution. The simulation results of average power consumption for different input signal distributions are summarized in Table 3.1.

Distribution	Power (comp OFF) [mW]	Power (comp ON) [mW]	Power overhead [%]
Uniform	27.0	32.4	20.0
Sine	24.3	32.4	33.6
Gaussian (6σ)	30.6	32.4	5.7

Table 3.1: Power overhead with compensation enabled

In the following sections, different methods of building the compensation DAC are introduced and compared in terms of digital complexity and size.

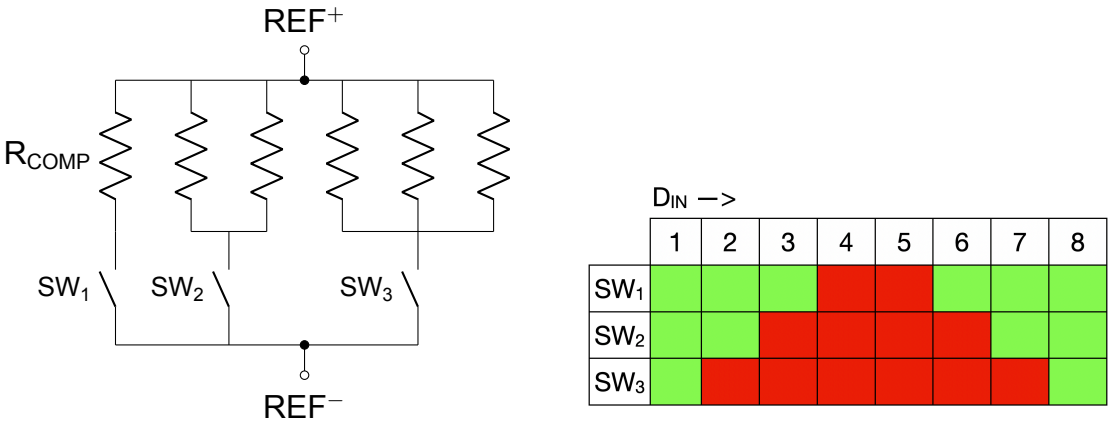
3.1.1. Quadratic compensation

To generate a quadratic current, one can connect a switchable array of resistors in parallel with the main DAC. The value of one unit of the compensation DAC, given certain load impedance R_L and a number of bits of the DAC, can be calculated as follows:

$$R_{COMP} = \frac{R_L}{4} \cdot (2^{Bits} - 1)^2 \tag{3.1}$$

and the number of units required follows:

$$N_{units} = \frac{(2^{Bits-1} - 1) \cdot (2^{Bits-1})}{2} \tag{3.2}$$



(a) 2-bit compensation DAC

(b) Signals for switches (green on, red off)

Figure 3.3: Compensation DAC and control signals for the switches

where Bits is the number of bits of the main DAC. The 2-bit compensation DAC consists of 3 segments. Each consecutive segment has one more resistor connected in parallel, as shown in Figure 3.3.

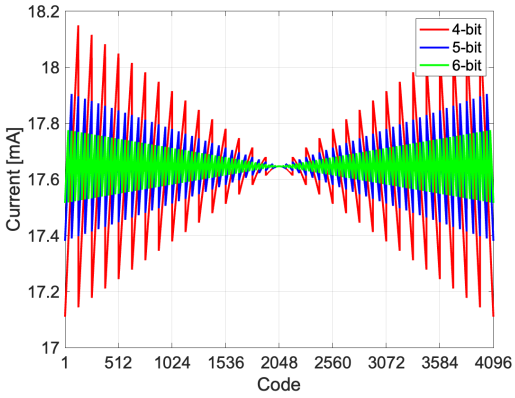
Although this method works well for lower-resolution converters, it is not feasible to use the same approach when dealing with high-resolution converters. The value of R_{COMP} increases very fast with the number of bits, as does the number of required units (for a 12-bit DAC $R_{COMP} \approx 420M\Omega$ and $N_{units} \approx 2mln$). Alternatively, instead of connecting units in parallel to produce the required resistance, each unit can be designed separately, greatly decreasing the area. However, the problem with separately designing each unit is matching. The ideal compensation of a 12-bit DAC is not feasible. Hence, different methods have to be investigated.

3.1.2. Partial compensation

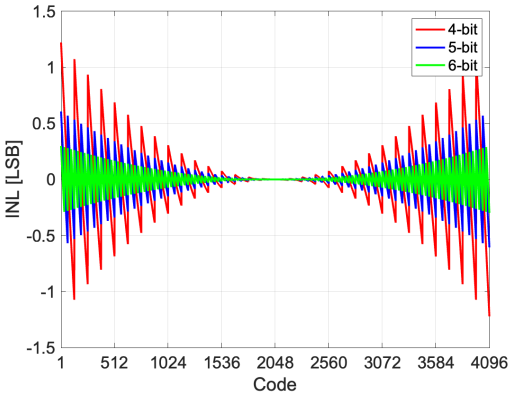
The previous section established that the ideal compensation of a 12-bit DAC is not possible due to the compensation DAC size scaling exponentially with the number of bits. The idea of partial compensation is to see if it is possible to linearize the supply current using a lower-resolution compensation DAC and still obtain 12-bit accuracy.

In Figure 3.4, the performance of lower resolution and, consequently, much smaller DACs for quadratic compensation is presented. In an ideal scenario, the supply current should remain constant, but it deviates from the constant value due to lower resolution compensation. The error is minimal around mid-code because the output current is small, and the parabolic supply current is in its 'flat' part. On the other hand, the

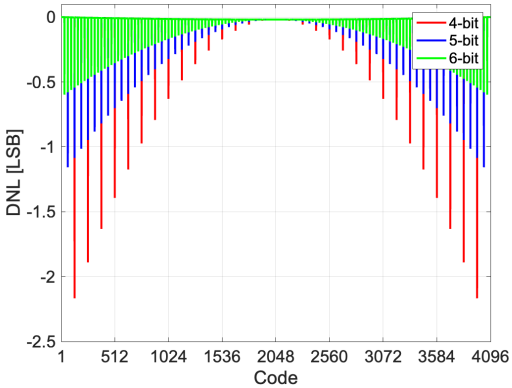
error is larger towards extremes as then the output current is maximal, and the supply current changes fast. The supply current peaks follow a linear contour. Equation 2.8 shows that the quadratic supply current results in a voltage drop across supply impedance that makes the overall reference voltage also quadratic. The obtained reference voltage is then multiplied by $D_{IN}/2K$, resulting in the output voltage that, except the linear term, also has a third-order term as shown in Equation 2.9. If now, the supply current shows linear dependency on the input code, we can conclude that the output voltage will contain a second-order term. Indeed, the linear shape of the supply current resulted in the quadratic behavior of the INL curve. Notably, even with very low-resolution compensation, such as 4 bits, the error in the proximity of the mid-code is minimal and much smaller than what is required. This raises the question of how much accuracy is needed. For compensation resolution, the differential nonlinearity sets the bottom boundary. The DNL error is greater than -1 LSB for less than 6 bits of resolution, translating to non-monotonicity. The resolution of the compensation DAC is directly linked to the magnitude of R_S. A higher compensation resolution is necessary for larger R_S values.



(a) Supply current after compensation



(b) 12-bit level INL after compensation



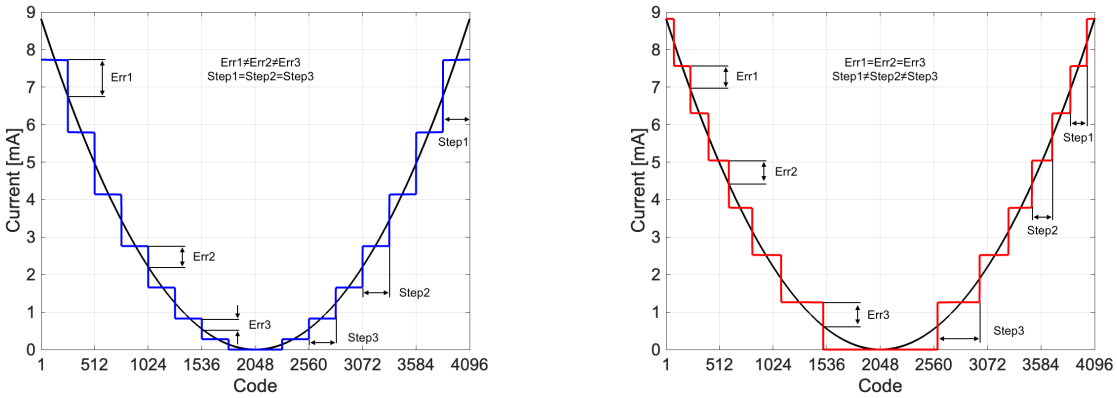
(c) 12-bit level DNL after compensation

Figure 3.4: Partial quadratic compensation for R_S = 1 Ω

The quadratic compensation method has a significant limitation regarding its accuracy towards the extremes when the DAC is fully tilted to one side.

3.1.3. Best fit compensation

We showed in the previous section that quadratic compensation suffers from limited accuracy at the extremes. At the same time, it is exceedingly accurate around mid-code. That good accuracy is not required, and it seems that trading accuracy in the middle for accuracy at the extremes could be beneficial. The difference between quadratic and best-fit compensation can be seen in Figure 3.4, plotting the compensation current required to be drawn from the supply to linearize the total supply current. The quadratic method takes equal steps on the x-axis, but its error on the y-axis is inconsistent and worsens further from the middle code. It results from fixing the step size to be equal for the whole range, while the quadratic function does not change with a constant rate. On the other hand, best-fit compensation takes unequal steps on the x-axis but maintains a steady error on the y-axis. To determine optimal switching points, the ideal compensation current can be divided with equally spaced horizontal lines as shown in Figure 3.6. Then, the intersection points will point to a code on the x-axis.



(a) Quadratic compensation

(b) Best-fit compensation

Figure 3.5: Quadratic and Best-fit compensation comparison

The supply current and INL simulation is repeated for the second method and plotted in Figure 3.7. The supply current peaks have the same value, and the INL has a linear shape. For this method, 5 bits are already sufficient to achieve a monotonic transfer function (DNL better than -1 LSB). Therefore, one bit is saved compared with the previous method. A drawback of this method is that to determine the exact switching points, all data inputs of the main converter have to be used.

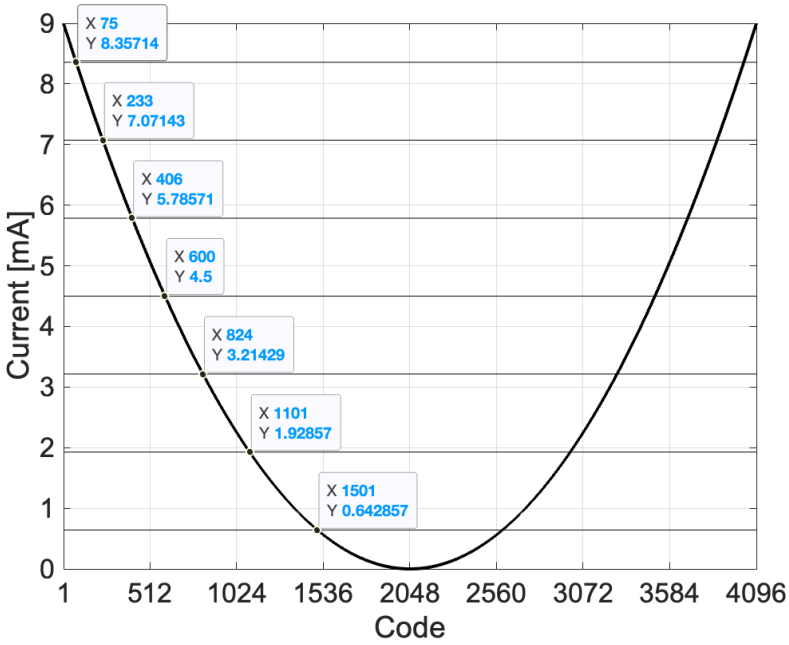
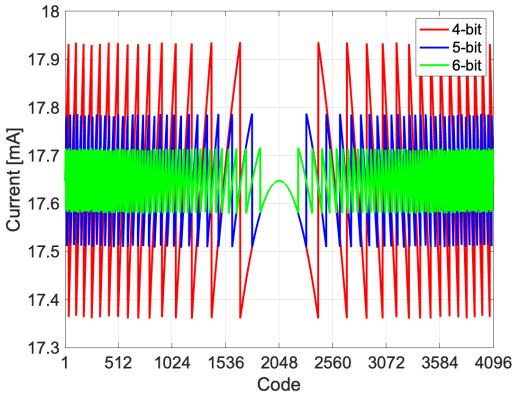
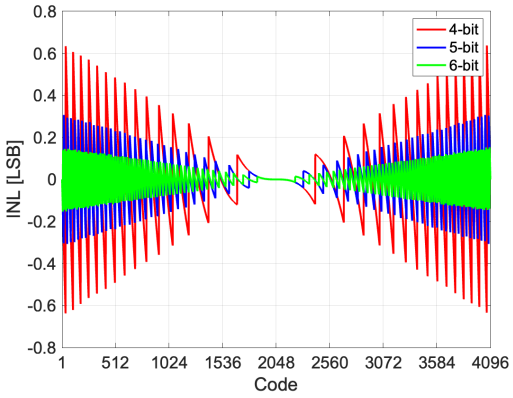


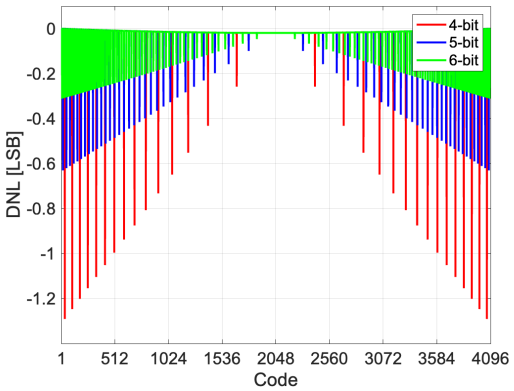
Figure 3.6: Obtaining switching indexes



(a) Supply current after compensation



(b) 12-bit level INL after compensaiton



(c) 12-bit level DNL after compensaiton

Figure 3.7: Partial best-fit compensation for R_S = 1 Ω

Similar to the previous methods, an array of resistors can be connected in parallel with the main data converter. The number of resistor units (NofU) in the compensation DAC can be arbitrary, and the value of R_{COMP} is simply:

$$R_{COMP} = 2R_L \cdot NofU \tag{3.3}$$

and the resulting circuit would look as shown in Figure 3.8

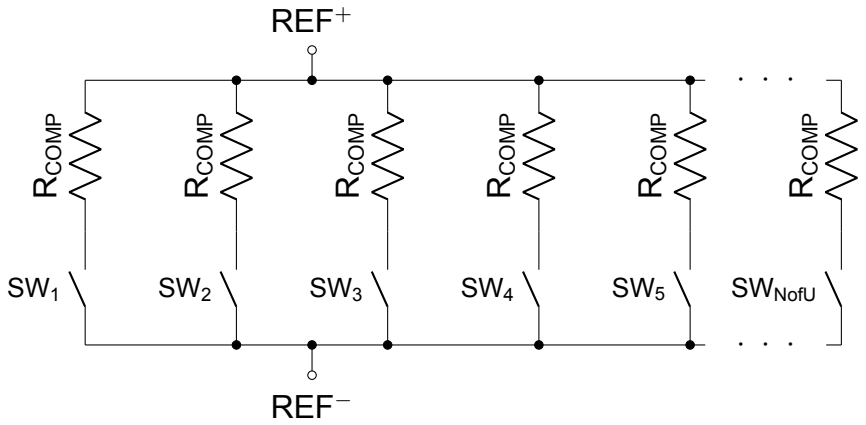


Figure 3.8: Best-fit DAC implementation

Two immediate advantages of this method are visible. Firstly, the R_{COMP} unit value is much smaller than in the previous method. For 6-bits, the unit value is approximately 400kΩ for the quadratic method, but only 12.8kΩ for the best-fit method. Secondly, the best fit uses identical resistors, and their number is significantly reduced. The only drawback of the best-fit method is its complexity. For a 12-bit DAC, it is not straightforward to detect switching points using digital logic. Before designing the complete compensation DAC, we will discuss segmentation. Segmentation will reduce the complexity of the main DAC, making the design of the compensation DAC easier.

3.2. Segmentation

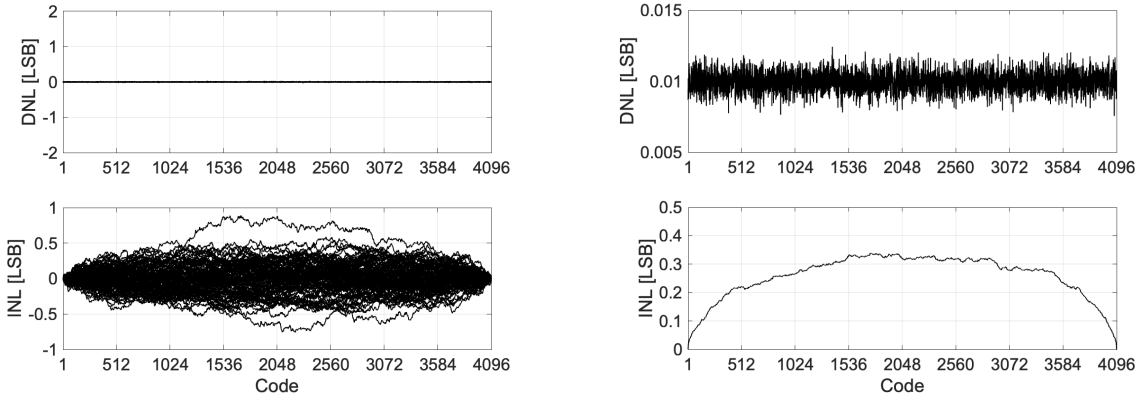
3.2.1. Unary digital-to-analog converter

The unary architecture is a straightforward method of constructing a digital-to-analog converter. Using this architecture has significant advantages. It is intrinsically monotonic, and to ensure a DNL error of +-0.5 LSB, only 50% matching is required. What is more, glitches that appear during switching in a unary DAC do not contribute to distortion because their magnitude is proportional to the number of units that are switching. However, these benefits come with a great cost as $2^{N-1} - 1$ unit elements are required to build an N-bit unary DAC. The equations from [16] can be used to predict the theoretical value of INL and DNL, given a certain spread (σ) between elements:

$$INL = 0.5\sqrt{2^N}\sigma \tag{3.4}$$

$$DNL = \sigma \tag{3.5}$$

For a 12-bit DAC with $\sigma = 1\%$ of mismatch between elements, the peak INL and DNL values are predicted to be 0.32 and 0.01 LSB, respectively. To verify these results, a Monte Carlo simulation can be performed. An array of resistors 4095, each having a mean of 1 LSB and (σ) of 1% is generated. Now, the INL and DNL of the resulting DAC is calculated in 100 random simulations. In Figure 3.9 (a), the results of all 100 simulations are plotted, and in (b), the root mean square values of the above-mentioned runs are calculated. The simulated results agree with the equations above.



(a) 100 runs with sigma = 0.01

(b) RMS value of all runs

Figure 3.9: Unary architecture

3.2.2. Binary digital-to-analog converter

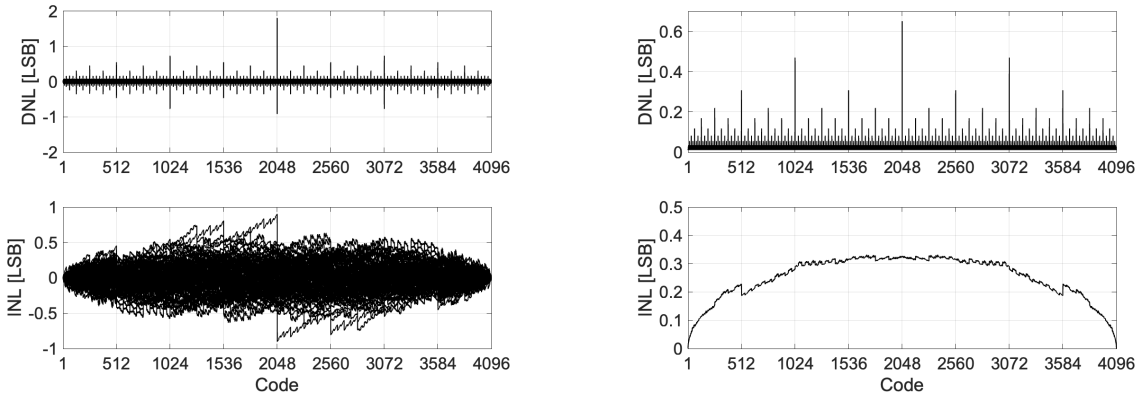
An exponentially growing decoding complexity and wiring in the case of fully unary architectures makes it very hard to implement unary high-resolution converters: A 12-bit DAC can be implemented using binary architecture, eliminating the need for 4095 separate data inputs and a complicated decoding block. Given the array of unary 4095 units, they can be grouped in the following fashion: the first unit is left unchanged (Least significant bit), the following two units are connected in parallel (LSB+1), then the next four units are merged (LSB+2), and it continues until the last group of 2048 units is combined (Most significant bit). The resulting converter has only 12 units, scaled in a power of 2 fashion. The area of the converter remains exactly the same as in the previous example. The INL and DNL can be calculated using the equations:

$$\mathrm{INL} = 0.5\sqrt{2^N}\sigma \tag{3.6}$$

$$\mathrm{DNL} = \sqrt{2^N}\sigma \tag{3.7}$$

Similarly, the theoretical values are checked with a simulation. The simulation results are plotted in Figure 3.10. Two crucial observations can be made: both architectures introduced so far result in the same INL error. The DNL error of the binary architecture is $\sqrt{2^N}$ time larger and is the main limitation of this architecture. During the mid-code transition, all bits have to switch, which results in a large peak in DNL. In the case of a unary architecture, for the same mid-code transition, only a single unit was switching. It is also worth mentioning that the non-linear relation between

the number of segments switched on and the output value in binary architectures may give rise to dynamic errors [16].



(a) 100 runs with sigma = 0.01

(b) RMS value of all runs

Figure 3.10: Binary architecture

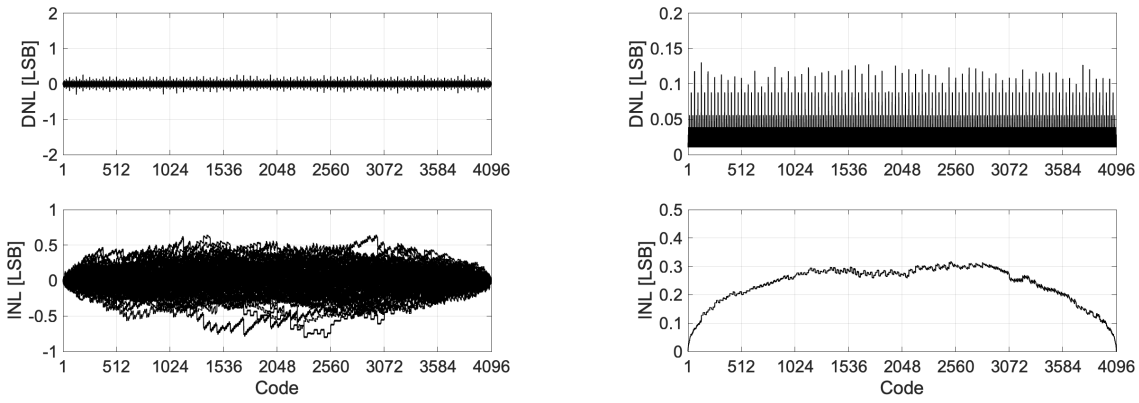
3.2.3. Combining unary and binary segments

In the design of high-resolution converters, using either unary or binary architectures alone is rare. Instead, a combination of both architectures is often implemented. Using unary architecture for the MSB DAC is preferred because its matching and dynamic performance benefits can be exploited. On the other hand, the LSB DAC can be implemented either as a binary or unary DAC. At the beginning of the design process, one must decide how many unary and binary segments to use. In [16], an extensive analysis of segmentation has been performed to investigate how to minimize the overall chip area while optimizing dynamic performance. The analysis reveals that as the architecture moves from fully binary to fully unary, the area is first dominated by the DNL requirements, then by the INL requirements, and finally, for a primarily unary architecture, by the decoding logic. Regarding area minimization, the optimum tends to be shallow, and the same (minimum) overall area can be achieved using different segmentation. In that case, it is preferred to choose more unary segmentation, as it shows better high-frequency performance. For completeness, the Monte Carlo simulation of a 6+6 segmentation is also performed, assuming an equal area. As can be seen in Figure 3.11, the INL remains the same, but the DNL error is somewhat larger than for the unary architecture but smaller than in the case of the binary architecture. The DNL and INL errors can be calculated with the equations:

$$\mathrm{INL} = 0.5\sqrt{2^{\mathrm{N}}}\sigma \tag{3.8}$$

$$\mathrm{DNL} = \sqrt{2^{\mathrm{N-U+1}}}\sigma \tag{3.9}$$

When 6 is substituted for 'U' in Equation 3.9, the resulting DNL is equal to 0.11 LSB, which corresponds well with the result in Figure 3.11 (b).



(a) 100 runs with sigma = 0.01

(b) RMS value of all runs

Figure 3.11: 6+6 segmentation

As mentioned previously, to build a higher resolution DAC it is possible to use two unary DACs. Due to project time constraints, this method is preferred because it allows for identical sub-DACs for both the MSBs and LSBs, which significantly reduces design and layout time. The implementation of a 12-bit DAC using two equal unary DACs is shown in Figure 3.12.

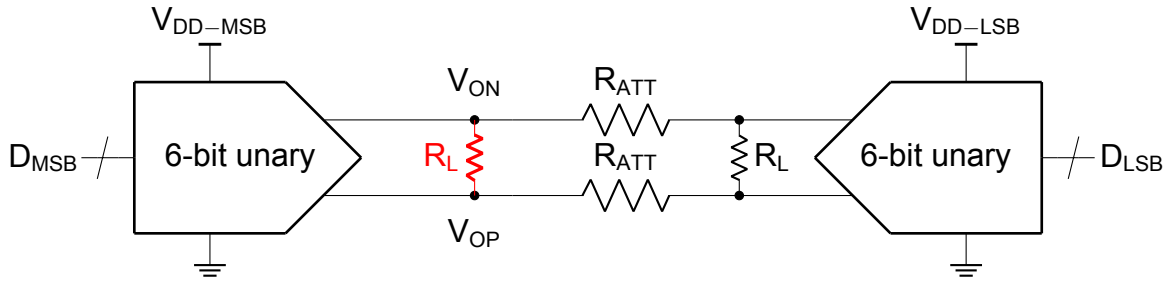


Figure 3.12: A 12-bit implementation using two equal unary DACs

It is important to notice that if the LSB DAC is an exact copy of the MSB DAC, its strength is also the same. For that reason, we need to attenuate the LSB side by a factor of 2^6. In Figure 3.13, a single-ended representation of the circuit from Figure 3.12 is shown.

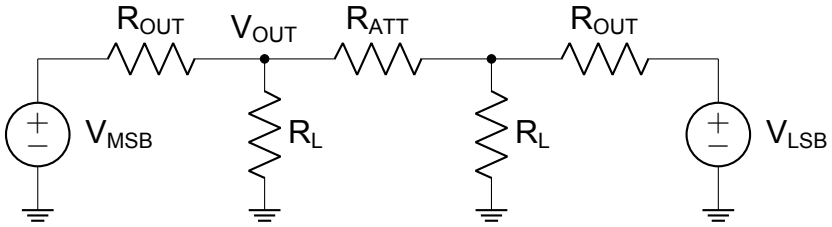


Figure 3.13: Circuit for calculating Ratt

In this circuit, R_{OUT} is the output impedance of each DAC, and it matches the load R_L (R_{OUT}=R_L). Now, the transfer function from V_{LSB} to V_{OUT} can be calculated as:

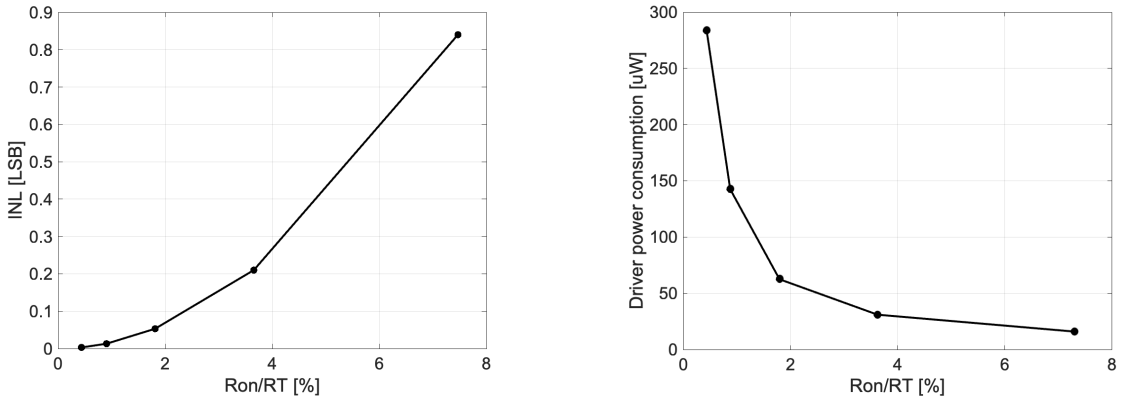
$$\frac{V_{OUT}}{V_{LSB}} = \frac{R_L/2}{R_{ATT} + R_L} \tag{3.10}$$

Solving for $V_{OUT}/V_{LSB} = 2^{-6}$ yields $R_{ATT} = R_L(2^5 - 1)$, which is the required resistance value to achieve adequate attenuation of the LSB DAC.

Finally, the following is proposed: instead of designing the main 12-bit DAC and compensation DAC separately, a 6-bit main DAC can be designed together with its compensation DAC, which is a much easier task. Then, two precisely the same circuits can be combined, resulting in a 12-bit DAC. The next sections will cover the circuit-level design of the main DAC and compensation DAC.

3.3. Main DAC unit implementation

In the previous section, the choice of the architecture has been made. A 12-bit DAC will be built using two equal 6-bit sub-DACs. Therefore, it is only required to design a 6-bit converter. The output resistance requirement of 50Ω readily tells what the unit resistance value needs to be: $R_T = 50 \cdot (2^6 - 1) = 3150\Omega$. However, changing the termination resistance to 3200Ω will enable us to use the same resistors and transistor to design the main DAC and the compensation DAC (if R_T = 3200Ω then R_{COMP} from Equation 3.3 can be made = 6400Ω which exactly twice the termination value). As discussed in section 2.2, termination resistance is made up of the switch's resistance and the resistor itself. Different nonlinearity arises depending on the ratio of these two, which is illustrated in Figure 3.14 (a).



(a) R_{ON} nonlinearity

(b) Driver power consumption

Figure 3.14: Switch resistance impact on linearity and driver power consumption

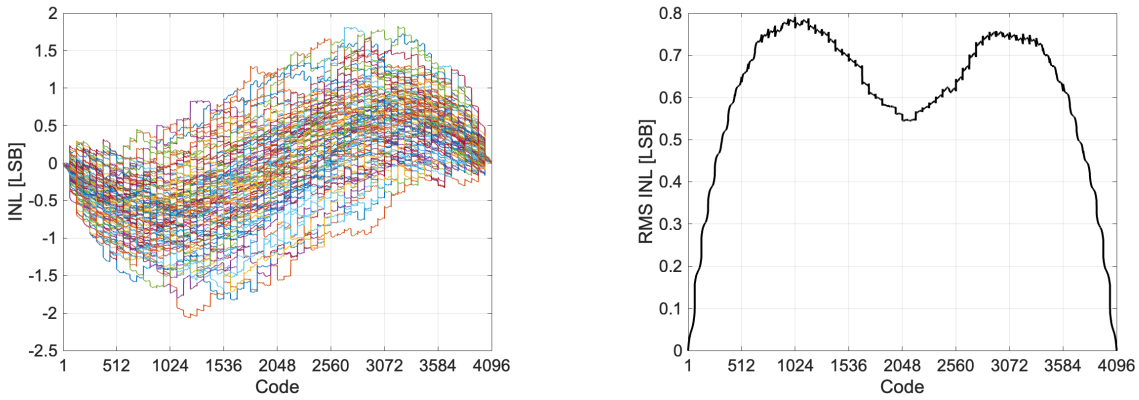
For higher values of the R_{ON} to R_T ratio, the operating point of the switches moves towards saturation. Then the transistors become more susceptible to V_{DS} variations. To keep the INL of this effect smaller than 0.1 LSB, a ratio of 2% was chosen. At this point, switch nonlinearity becomes negligible, and decreasing its resistance more yields no performance improvement. Moreover, making switches larger also increases capacitance, and big switches are harder to drive. In Figure 3.14 (b), the power consumption of a single driver switching at Nyquist frequency for different R_{ON} to R_T ratio

is plotted. For this simulation, single inverters are used, and they are sized to maintain the same rising time for all points. At higher frequencies, the dynamic power consumed by the drivers preceding the switches can easily be higher than the static power consumption of the DAC itself. Hence, there exists a trade-off between speed, static linearity, and power consumption.

The NMOS and PMOS transistors are designed to have the same resistance for mid-code and room temperature. To achieve that, the PMOS transistor is designed with a larger width than the NMOS.

The choice of resistors is also important. High sheet resistance helps in reducing the overall area of the design. What is more, good matching properties are important. For that, P-poly resistors are used [17]. They also feature a very low (negative) temperature coefficient, meaning they are stable across a wide range of temperatures.

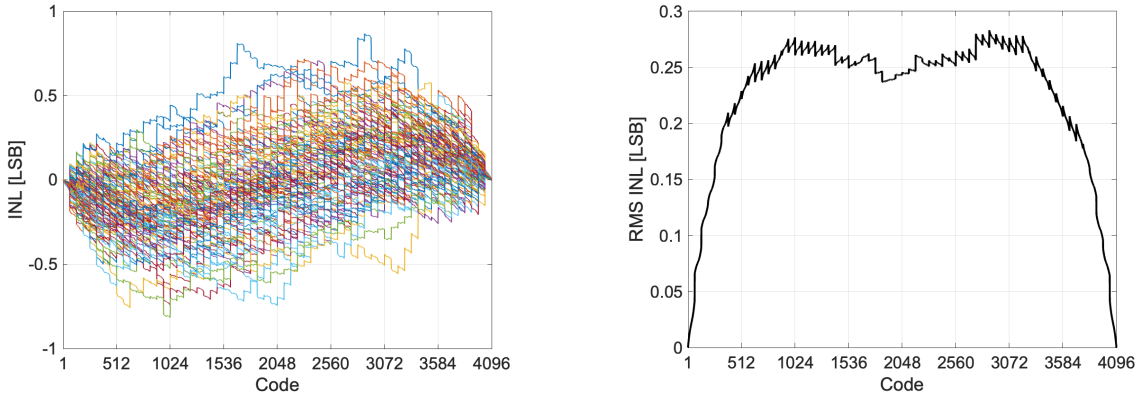
To verify the mismatch of transistors and resistors, 100 Monte Carlo runs were simulated, and the results of these simulations were plotted in Figures 3.15 and 3.16. A higher yield (INL between +-0.5 LSB) can be achieved by further increasing the resistor's area [18]. It is worth noting that transistors do not contribute significantly to the mismatch, as they account for only 2% of the termination.



(a) 100 Monte Carlo runs

(b) RMS value of all runs

Figure 3.15: Monte Carlo simulation, resistor width = 1um



(a) 100 Monte Carlo runs

(b) RMS value of all runs

Figure 3.16: Monte Carlo simulation, resistor width = 2um

P-poly resistors also show voltage dependency, as illustrated in Figure 3.17. It explains why nonlinearity is visible in Figure 3.15 (b). To obtain good matching and linearity, a width of 2um is chosen for resistors.

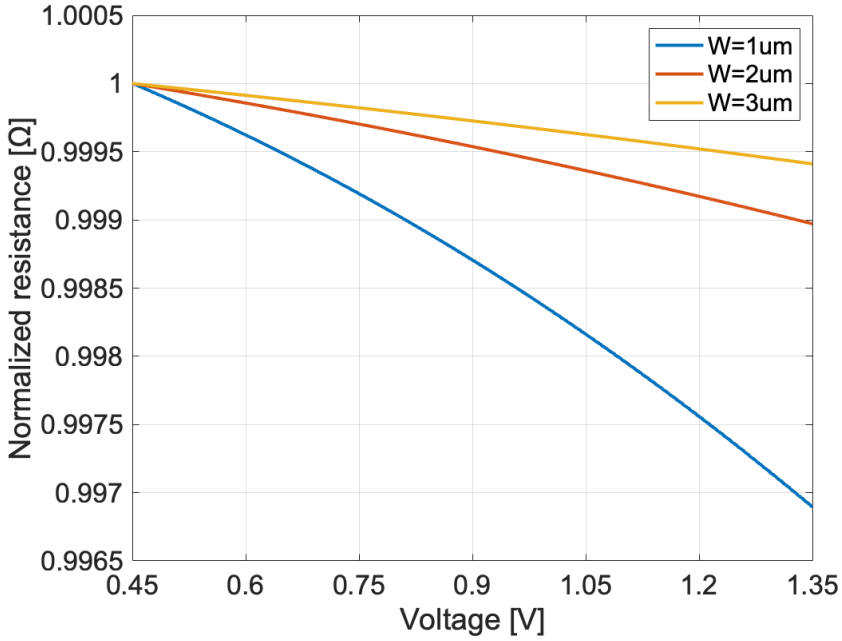


Figure 3.17: P-poly resistors voltage dependency

A single differential DAC unit can be built with four transistors and two resistors, as shown in the schematic in Figure 3.18.

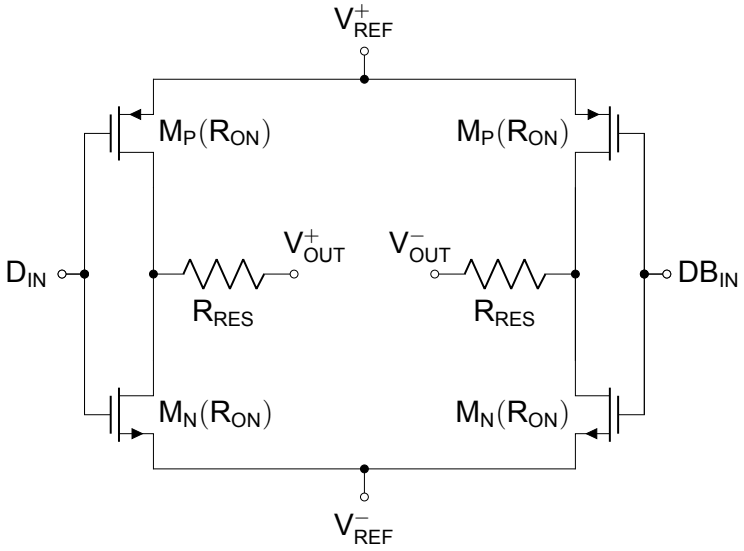


Figure 3.18: Differential DAC unit

When the transistors are switching, both PMOS and NMOS are conducting momentarily, and they pull a large current from the supply (crowbar current). This sudden

supply current change will cause a voltage drop across the supply impedance, changing the reference voltage. Because the output voltage is proportional to the reference voltage, as shown in Equation 2.9, the effect of crowbar current can also be observed in the output voltage of the DAC. This problem has been addressed in [19], where separate termination resistors are introduced in the push-up and pull-down paths as shown in Figure 3.19. By doing it, the crowbar current can be significantly reduced. However, twice the number of resistors needs to be used. Moreover, the bandwidth of the converter will be reduced. A comparison between the two methods is presented in Figure 3.20.

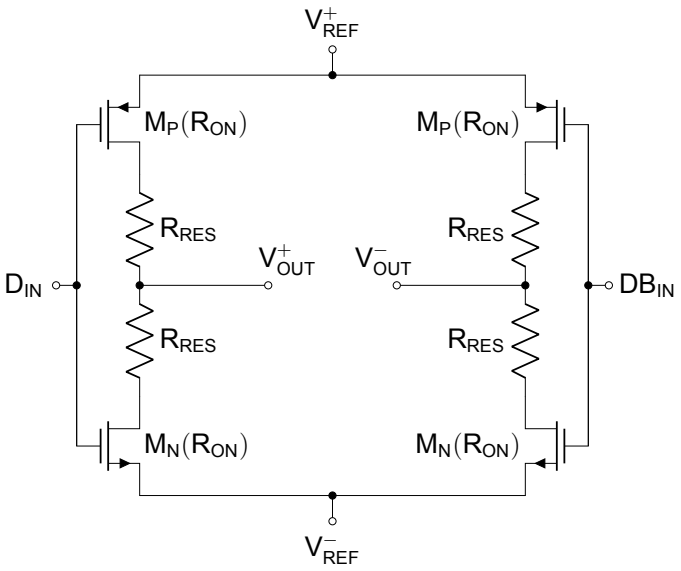


Figure 3.19: Differential DAC unit

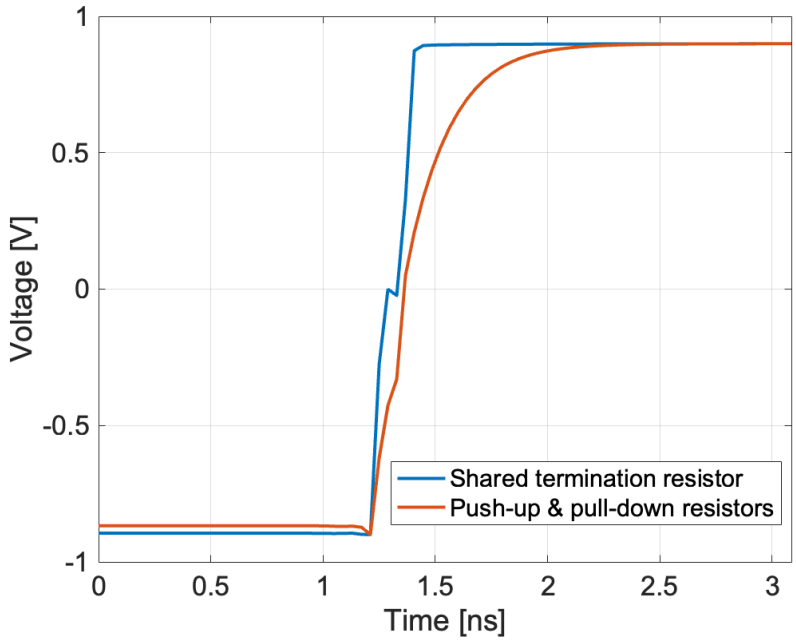
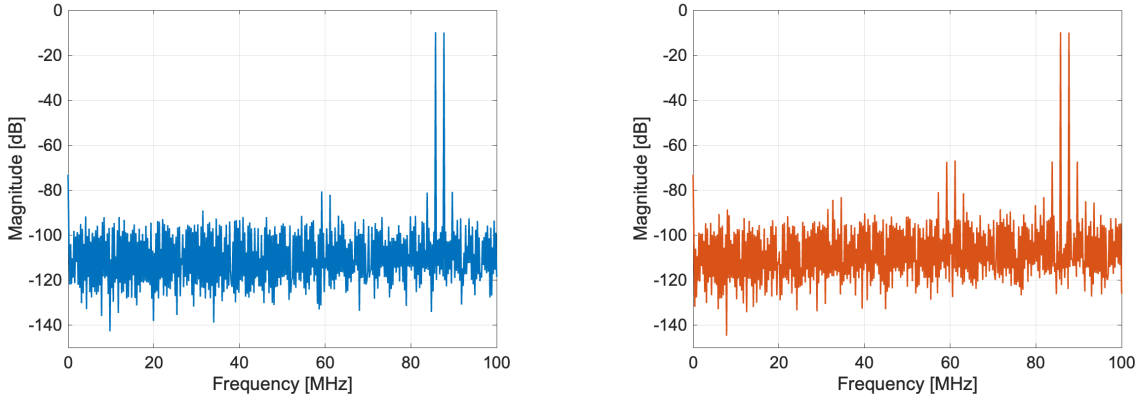


Figure 3.20: Settling comparison

A full-scale input step is applied to the DAC's input for both circuits, and the output voltage is plotted. Using only a single, shared termination resistor results in much faster settling, but the effect of the crowbar current manifests itself as a sharp wiggle in output voltage while transitioning from low to high level. On the other hand, when two resistors are used, settling is smoother.

A two-tone FFT simulation was performed for a high-frequency input to verify which method results in better linearity. The results of this simulation is presented in Figure 3.21.



(a) FFT results for one resistor

(b) FFT results for two resistors

Figure 3.21: Comparison of termination methods, FFT simulation

Using only a single resistor obtains much better results (SNDR = 64.3 dB) compared to using push-up and pull-down resistors (SNDR = 53.8 dB). This is because nonlinear slewing at the output is much shorter when only a single resistor is used.

- M_P: W/L = 32.00um/0.18um; $R_{ON} = 64\Omega$
- M_N: W/L = 9.32um/0.18um; $R_{ON} = 64\Omega$
- R_{RES}: W/L = 2.00um/18.78um; $R = 3136\Omega$

It is also important to realize that across corners and for different temperatures, the value of R_T can vary significantly (+-20%), which will cause the DAC's output impedance to not match the channel impedance. Different techniques can be used to compensate for temperature and process variation. Some of them include body-biasing of the switches and adding extra parallel branches, [20] or tuning the size of MOS devices [21]. For simplicity reasons, no techniques to correct for PVT variations were implemented. It is important to add that for two of the above-mentioned methods (body-biasing and MOS tuning), the switch resistance needs to be a significant portion of R_T, which is not the case in our design.

3.4. Need for adjustable LSB strength

In Section 3.2.3, adjusting the strength of the LSB side is done using attenuation resistors R_{ATT}, which needed a value of $R_{ATT} = R_L(2^5 - 1) \approx R_T/2$. In the previous section, we explained that R_T consists of both a resistor and a transistor. To obtain

the value of $R_T/2$ needed for the attenuation resistor, one can connect two R_T units in parallel. However, this solution is not feasible because transistors cannot be placed in the signal path. That is because high voltage swings across R_{ATT} for different input codes. Instead, transistors are neglected, and only resistors are used to produce R_{ATT}. The deviation from an ideal value of R_{ATT} will cause DNL errors. To adjust the LSB DAC strength, its load resistor can be varied. However, it is a differential resistor, and making it programmable is not feasible. A solution to this problem is proposed in Figure 3.22.

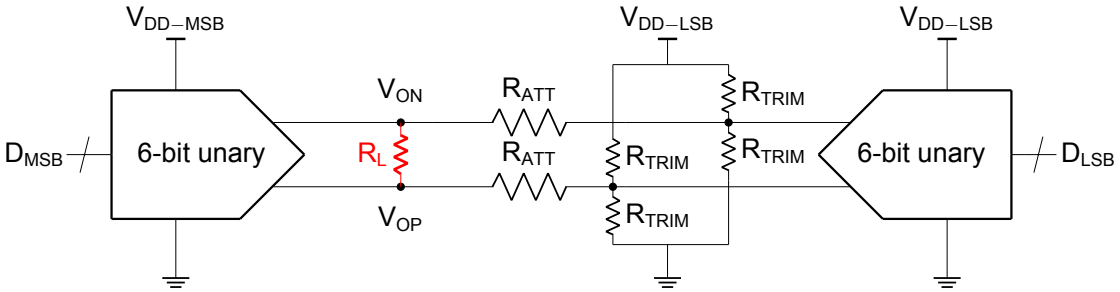


Figure 3.22: 12-bit DAC with extra trimming for LSB strength

The load resistor of the LSB DAC can be divided into four separate resistors. Now, because these resistors are connected to V_{DD} or ground, it is possible to make them programmable using transistors.

To obtain a load of 100Ω for the LSB DAC, each of four R_{TRIM} resistors ideally needs a value of 100Ω. We can use the same switches and resistors (R_{SW} = 64Ω and R_{RES} 3136Ω) as in the main DAC to construct R_{TRIM}. The implementation is shown in Figure 3.23. For readability, transistors are replaced with ideal switches.

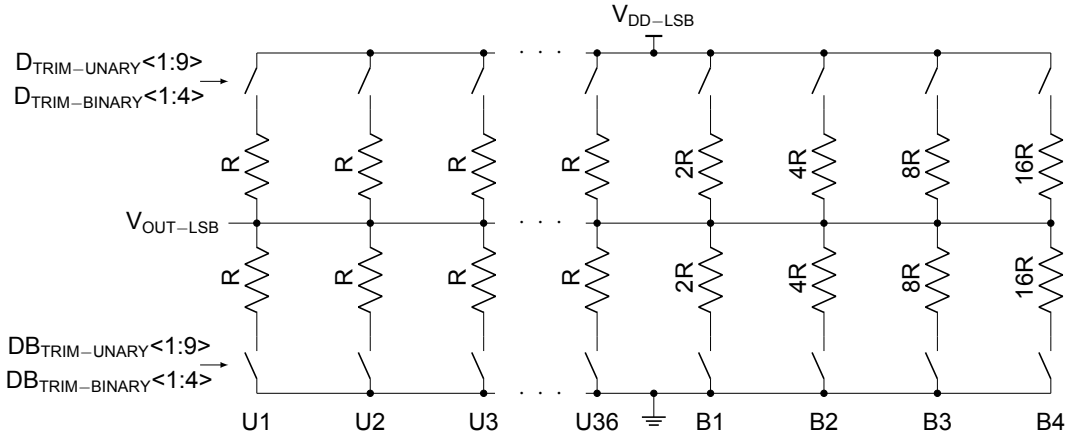


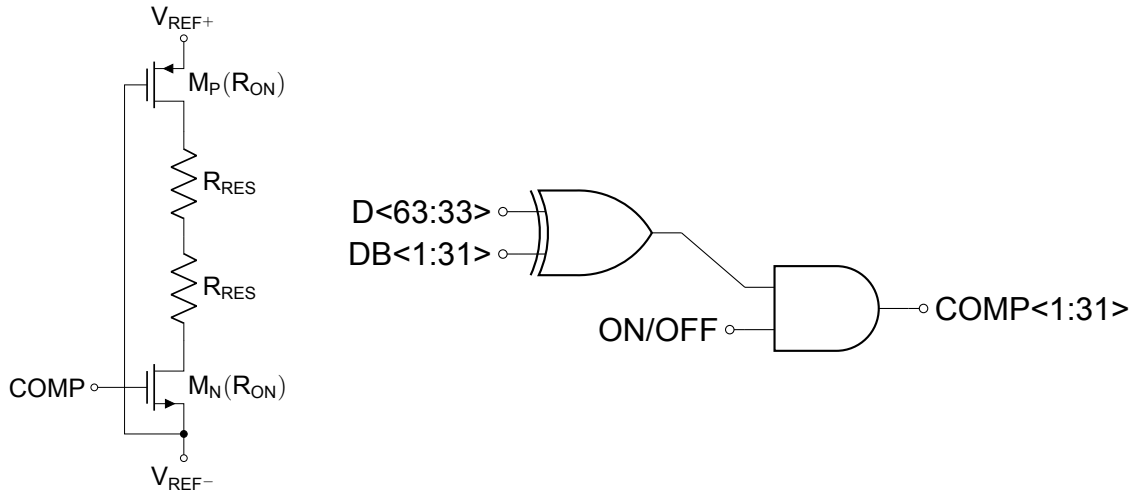
Figure 3.23: Implementation of two R_{TRIM} resistors sharing one output

Two programmable resistors are built using 36 unary units (U1-U36) and 4 binary units (B1, B2, B3 and B4) connected in parallel and sharing a single output node. Worth noting, each unary unit is built precisely as one side of the differential DAC unit from Figure 3.19, the only difference being that PMOS and NMOS are driven with separate signals so that they can be both turned on or off at the same time. To

build binary units, more resistors are added in series to obtain 2R, 4R, 8R, and 16R. Switches are not scaled because their contribution is negligible. 27 unary units are fixed, meaning their switches are always on, and the remaining 9 unary units and 4 binary units can be controlled. For nominal resistance of 100Ω (looking up or down from $V_{OUT-LSB}$), 32 unary units have to be connected. With the configuration mentioned above, changing the value of R_{TRIM} by +- 13% with a minimum step of 0.1 Ω is possible, which can easily correct for R_{ATT} deviation and get us the required LSB strength.

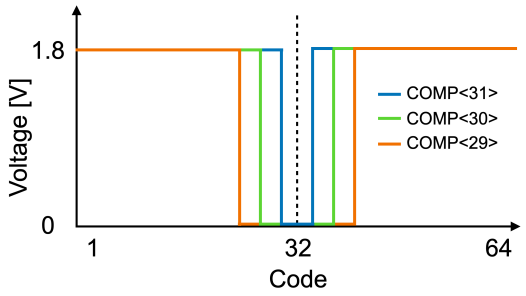
3.5. Compensation DAC circuit and logic implementation

The compensation DAC is built using 32 units connected in parallel, which means it is half the size of the main DAC. The logic is realized using XOR gates cascaded with AND gates. XOR gates are necessary to produce symmetrical signals, which drive compensation DAC switches. For the extremes, all compensation units are conducting. Moving towards mid-code, fewer and fewer compensation units are enabled. An overview of compensation DAC circuit implementation and logic is shown in Figure 3.24.



(a) Compensation DAC unit

(b) Compensation logic



(c) Compensation logic signals

Figure 3.24: Compensation DAC overview

Although 32 units are used, the resulting DAC has 25 levels because of the way it is wired. The reason for this is explained in Figure 3.25.

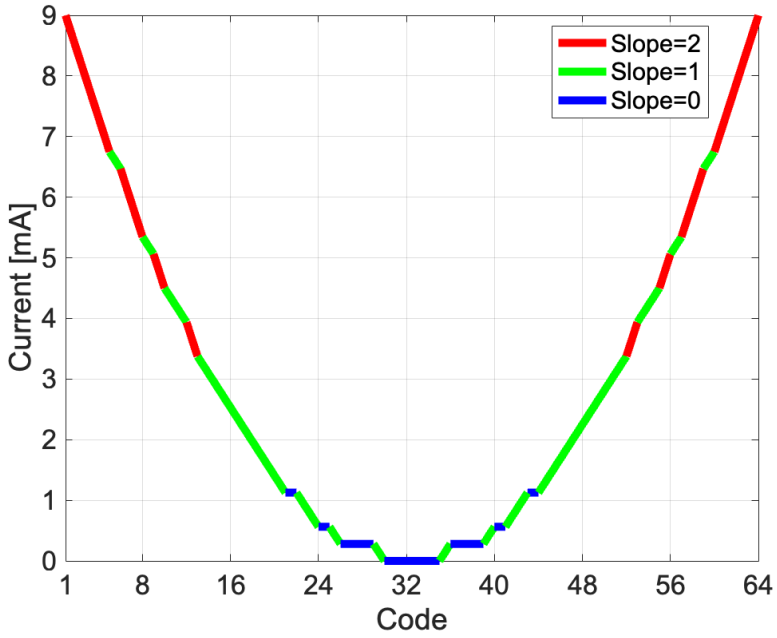
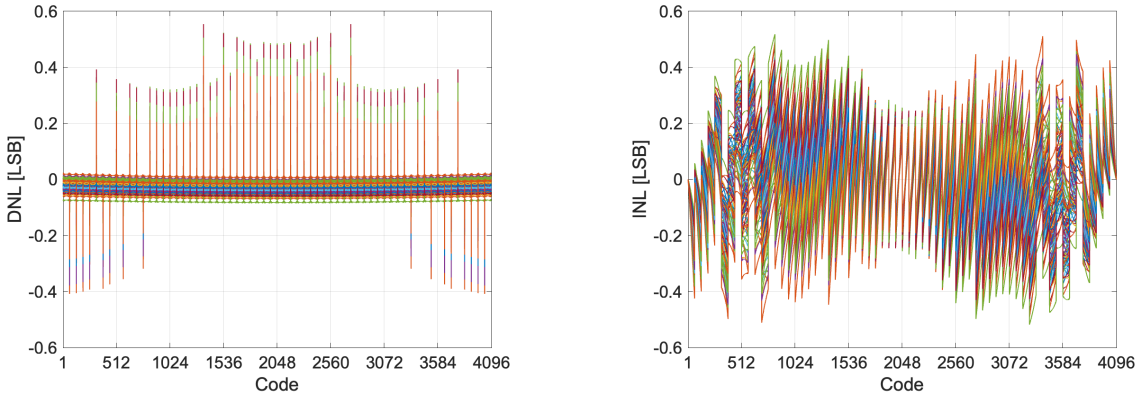


Figure 3.25: Compensation current with switching indicated

The parabola is constructed using three different slopes to better approximate the nonlinear current required for compensation. To make up for the slow-changing compensation current required close to mid-code for some data inputs, the compensation DAC remains static (blue color), and similarly, when the current has to change faster towards the extremes, two compensation units are turned at once (red color). In between, if the main DAC switches one unit on or off, the compensation DAC also enables or disables a single unit (green color).

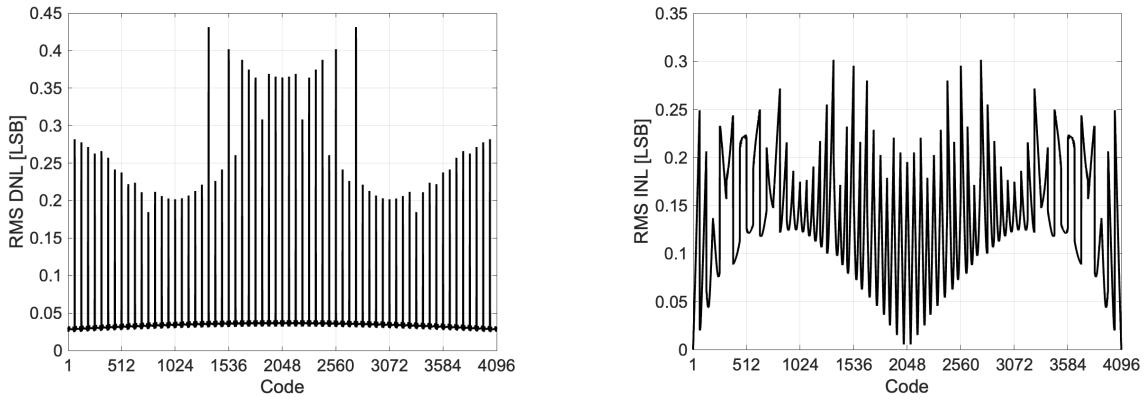
To verify performance, 100 Monte Carlo runs (typical corner) were simulated with 1Ω supply impedance added to both supply and ground connection. The DNL and INL results and their RMS values are shown in Figures 3.26 and 3.27, respectively.



(a) All DNL curves

(b) All INL curves

Figure 3.26: 100 Monte Carlo runs



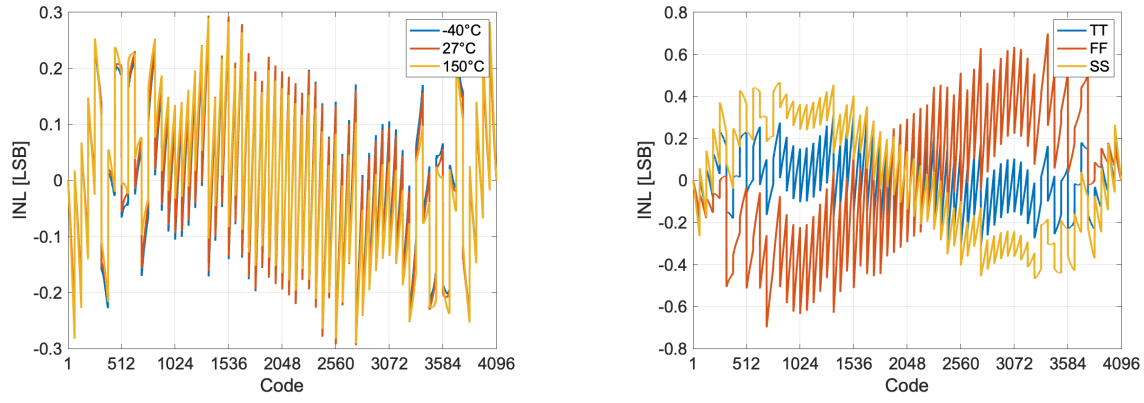
(a) RMS of all DNL curves

(b) RMS of all INL curves

Figure 3.27: RMS values of all Monte Carlo runs

The current drawn from the supply depends on both R_T and R_L. R_L is an external impedance. Any mismatch between on-chip and external impedance reduces the effectiveness of the compensation. The output impedance of the DAC can be affected by both temperature and process variation. Thanks to the very low-temperature coefficient of the resistors, negligible switches' resistance, and the fact that the same components are used for the main DAC and the compensation DAC, the static performance remains good for a wide temperature range. The only problem is that for extreme temperatures (-40°C or 150°C), the LSB strength has to be adjusted with the trimming DAC introduced in Section 3.4.

For different process corners, the value of output impedance can vary up to 20%, resulting in worse compensation performance. The fast corner gives the worst results, which can be explained by the fact that with the lower output impedance than the nominal 50Ω, the supply current variation is more than that shown in Figure 2.3. On the other hand, when the output impedance is too high, supply current variation is less, and 'overcompensation' takes place, which also explains different distortion polarities for FF and SS corners. The results are plotted in Figure 3.28



(a) INL across temperature

(b) INL for different corners

Figure 3.28: Process and temperature variation influence on INL

3.6. Binary-to-thermometer decoder

Given that both the MSB and LSB DACs are unary, they both need a binary-to-thermometer decoder. A standard decoder would result in 63 output wires that have to be routed to every switch. For two DACs, that number is doubled. In [22, 16], Row and Column decoding is used to reduce the wiring complexity.

The decoder is divided into two parts: a core and local decoding. Figures 3.29 and 3.30 show the core and local decoding circuits, respectively. The core circuits produce row and column signals, while local decoding, common to every DAC cell, generates the thermometer code. In Figure 3.31, a map is used to visualize how to convert Row and Column signals into the thermometer code using the local decoding circuit. The left-most column contains signals that need to be applied to Row_{up} and Row_{down} inputs, while the top row shows signals that need to be applied to Col input. This technique significantly simplifies wiring requirements, as only 14 output wires are necessary instead of the 63 wires that would be needed without it.

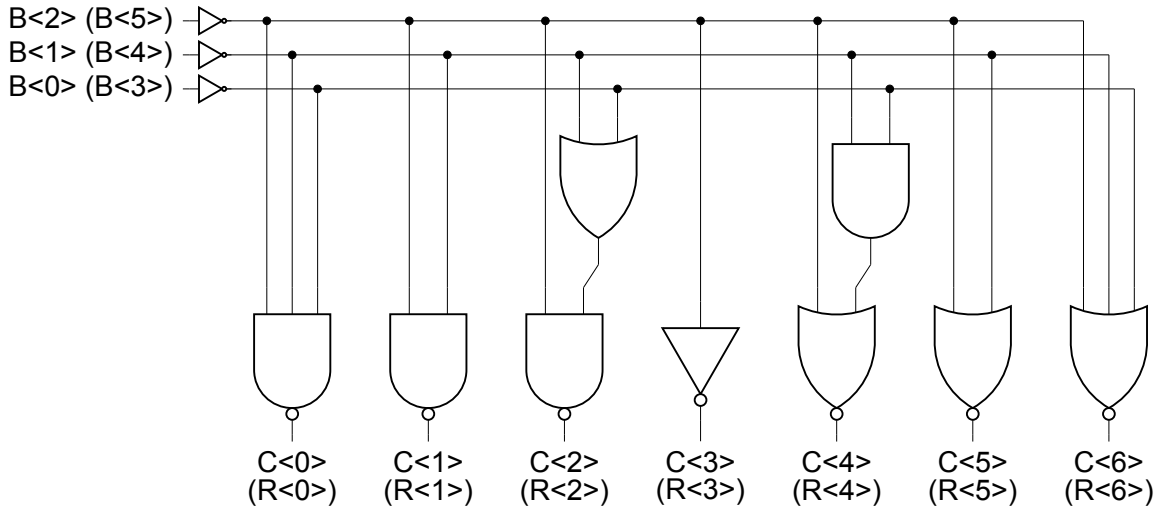


Figure 3.29: Column (Row) decoding

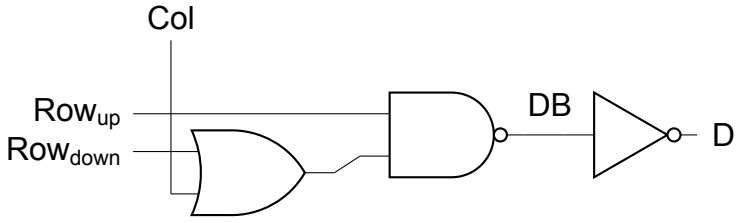


Figure 3.30: Local decoding

	C<0>	C<1>	C<2>	C<3>	C<4>	C<5>	C<6>	VSS
VDD R<0>	0	1	2	3	4	5	6	7
R<0> R<1>	8	9	10	11	12	13	14	15
R<1> R<2>	16	17	18	19	20	21	22	23
R<2> R<3>	24	25	26	27	28	29	30	31
R<3> R<4>	32	33	34	35	36	37	38	39
R<4> R<5>	40	41	42	43	44	45	46	47
R<5> R<6>	48	49	50	51	52	53	54	55
R<6> VSS	56	57	58	59	60	61	62	63

Figure 3.31: Row & Column to thermometer map

3.7. Latches and timing accuracy

It is key to the performance of a DAC that the change of output signal happens exactly on the grid of the clock. Timing errors in DAC circuits are of equal importance to amplitude errors. Signal (code) dependent delays of logic gates can cause distortion if fed directly to the switching transistors. To provide an example, the delays caused by the proposed row-column decoding architecture in signals arriving at the switches are plotted in Figure 3.32.

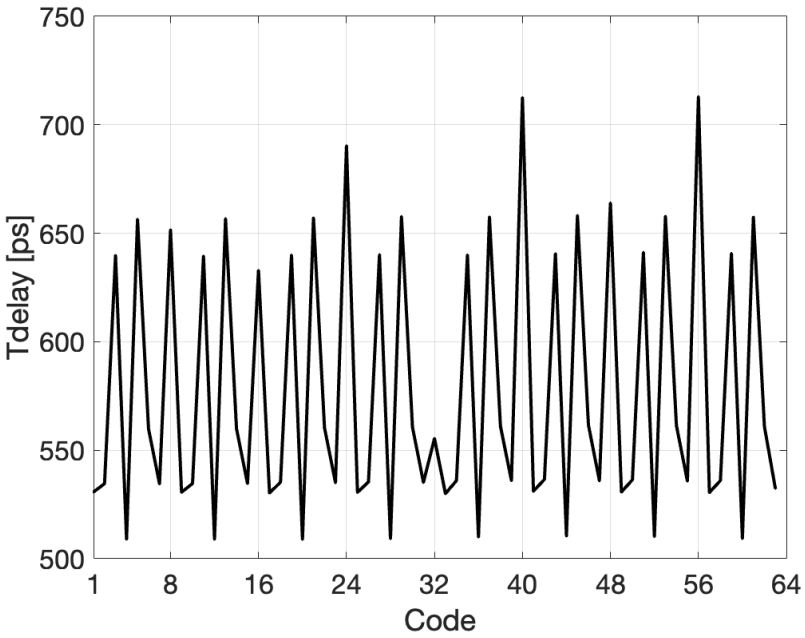


Figure 3.32: Code-dependent delays caused by decoding (extracted simulation)

For certain codes (24, 40, and 56), it takes much longer to make it to the switches. To alleviate this problem, it is necessary to re-time signals driving the switches. To do that, latches are used. Figure 3.33 illustrates how re-timing is achieved.

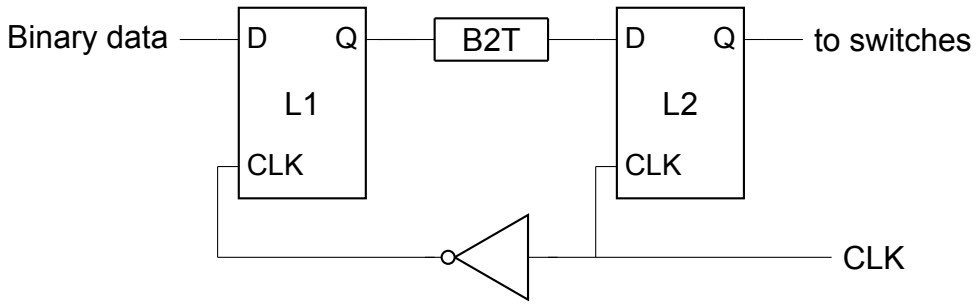


Figure 3.33: Re-timing input signals

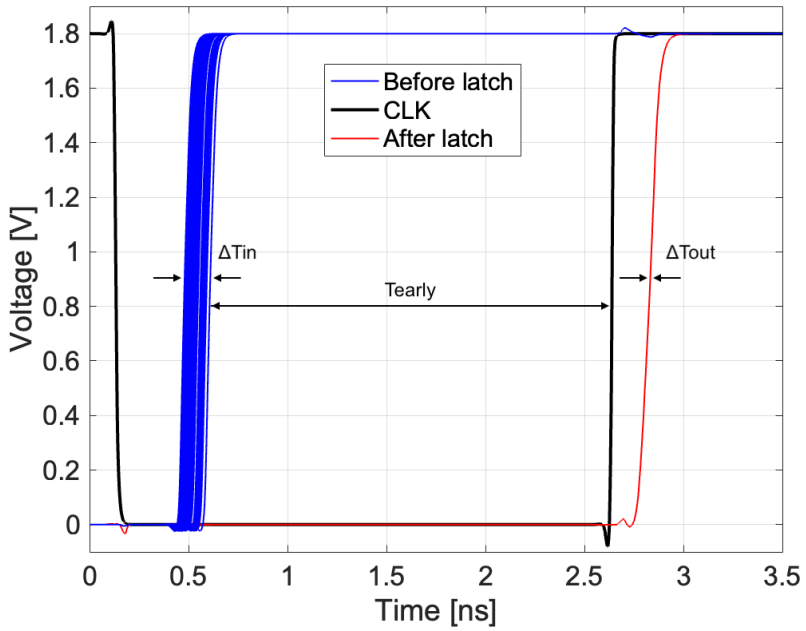


Figure 3.34: Timing spread improvement after using a latch

On the falling edge of the clock signal, the first latch (L1) transmits the binary data to the binary-to-thermometer decoder (B2T). The decoder's output is then passed through the second latch (L2) on the rising edge of the clock. Figure 3.34 illustrates the input signals of the second latch (blue curves), output signals of the second latch (red curves), and the clock signal (black curve). As seen in the plot, the data arrives at the input of the second latch with considerable spread well before the rising edge of the clock, indicated by 'Tearly.' After the rising edge of the clock, the data is passed through the second latch and shows a significant improvement in its timing spread. We can define a latch gain as the improvement in timing uncertainty between the output and input signals of the latch: $\Delta T_{in}/\Delta T_{out}$. To attain a 12-bit timing accuracy in the signals that drive the switches, ΔT_{out} must be restricted to $T_{CLK}/2^{12}$, which roughly

translates to 1 ps. Given the input spread, which was simulated to be roughly 200 ps, a latch gain of 200 is needed. In [23] latch gain is calculated as:

$$A_{LATCH} = e^{\frac{-t_{early}}{\tau_{driver}}} \tag{3.11}$$

In order to maximize the latch gain, two factors need to be taken into account. Firstly, it is important to maximize Tearly. This is because the gain is exponentially dependent on how early the signals reach the input of the latch. Secondly, the timing constant of the latch's driver should be minimized. The influence of these two factors is illustrated in Figure 3.35. The driver's time constant is changed by varying the latch's input capacitance. Obtaining accurate latch gain simulation results is

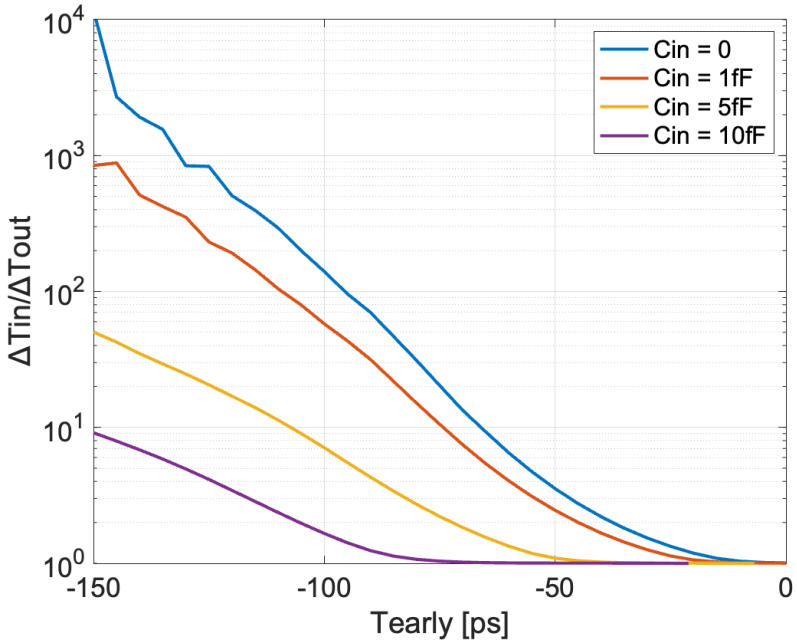


Figure 3.35: Latch gain for different latch's input capacitance

not straightforward and is limited by the simulator's accuracy, which is visible for the blue and red curves and manifests in sharp changes. Any parasitics at the input of the latch will change the driver's time constant and can significantly affect the performance. When higher speed converters are considered, combining shorter available Tearly and layout-induced parasitics, the timing spread reduction may not be sufficient. For that reason, [5] proposes using multiple latches. The circuit from 3.33 can be modified by adding the second latch, as shown in Figure 3.36. This modification dramatically improves timing spread. Moreover, the signal path is never transparent, which helps in robustness. If necessary, another latch can be added. Although one latch already gives good results (12-bit timing accuracy is achievable), to account for layout and other nonidealities, a second latch was also added.

In Figure 3.37, a transistor-level implementation of the latch circuit is shown. The latch circuit uses two back-to-back connected inverters (M1-M2 and M3-M4) and 4 NMOS transistors (M5-M8) driven by data inputs and the clock. Correct sizing is essential to ensure proper functionality of the circuit. Symmetrical PMOS and NMOS is

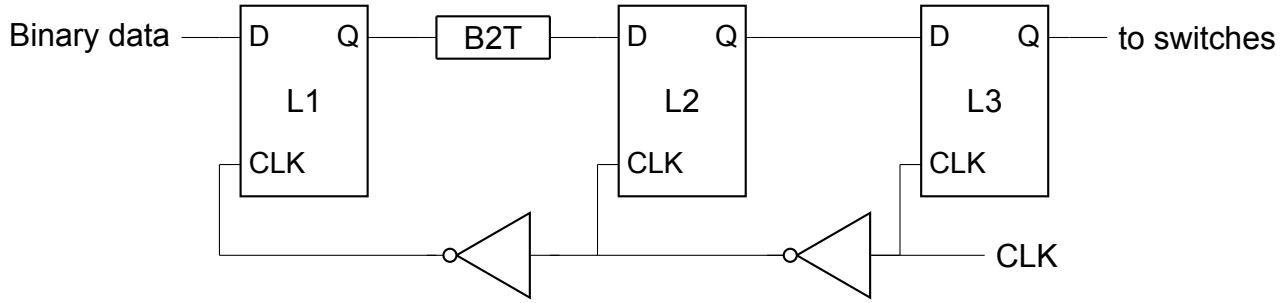


Figure 3.36: Adding another latch to improve timing spread reduction

less critical than fast switching; hence, PMOS transistors M2 and M4 are not sized to have the same strength as M1 and M3 transistors. The reason is that for the latch to switch, transistors M5 and M6 (M7 and M8) need to be stronger than M2 (M4). M5 and M6 (M7 and M8) transistors can be viewed as a single NMOS transistor with double the length. It has been verified that using the same size for all transistors results in a good performance even for the worst corner case (fast PMOS, slow NMOS). M1-M8 transistors have W/L of 1.62um/0.18um.

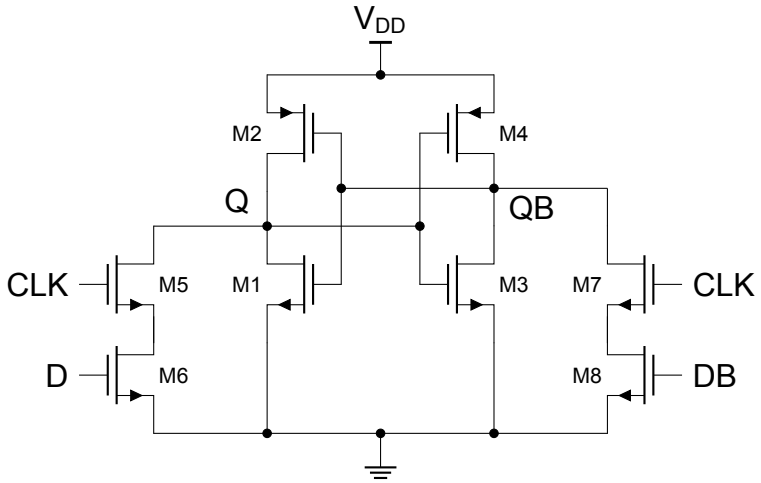


Figure 3.37: Transistor level latch implementation

The simulation of code-dependent delay was repeated after adding latches. The results in Figure 3.38 only show the spread reduction after the first latch. After the second latch, simulator accuracy was insufficient to show clear results. The delay is referred to the same clock edge as in Figure 3.32. The obtained latch gain is roughly equal to 400, which makes signal-feedthrough negligible.

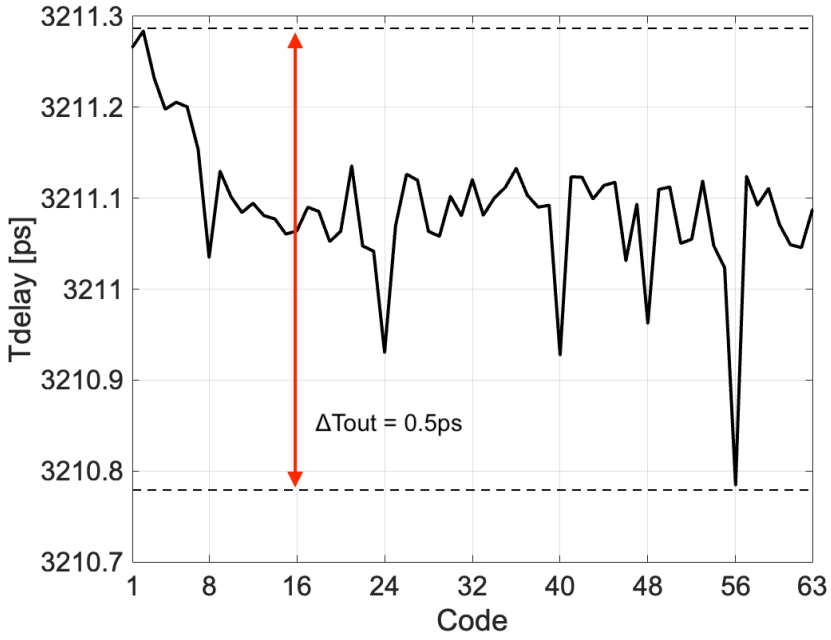


Figure 3.38: Code-dependent delays reduced by latch (extracted simulation)

3.8. Complete 6-bit DAC with compensation

Figure 3.39 illustrates a complete DAC unit with latches and local decoding. Extra drivers are required after the second latch to ensure quick switching of the DAC switches. Each driver is a tapered chain of inverters. The first inverter is 4um/2um, and the second is 8um/4um. Two separate supplies are used: one for the DAC core circuit (AVDD) and the other for the rest of the circuitry (logic, latches, and switch drivers).

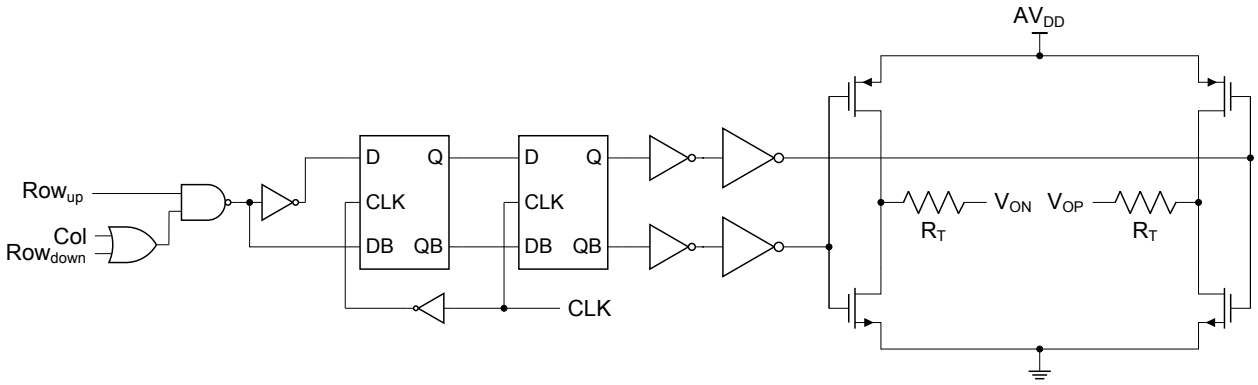


Figure 3.39: Complete DAC unit

The compensation DAC unit in Figure 3.40 uses exactly the same latches and switch drivers. The only difference is that it includes two local decoding units (omitted for readability) that produce IN_1 and IN_2 signals (see Figure 3.24 (b)) for the digital logic. Because for simplicity, we want to preserve the same design, and there is only

one transistor to drive; one output of the switch drivers is not connected to anything. Having only one switch to drive will also impact switching speed. For that reason, a dummy PMOS transistor (not shown in Figure 3.40) is added at the out of the driver to match the timing between compensation and the main DAC.

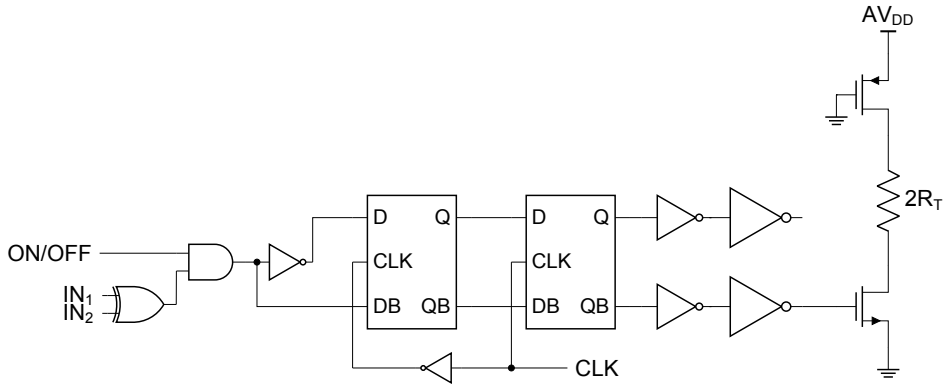


Figure 3.40: Complete compensation DAC unit

In the end, each of the two 6-bit sub-DACs is built using 63 DAC units connected in parallel as shown in Figure 3.41. Similarly, the compensation DAC is built by merging compensation DAC units. The resulting circuit is shown in Figure 3.42.

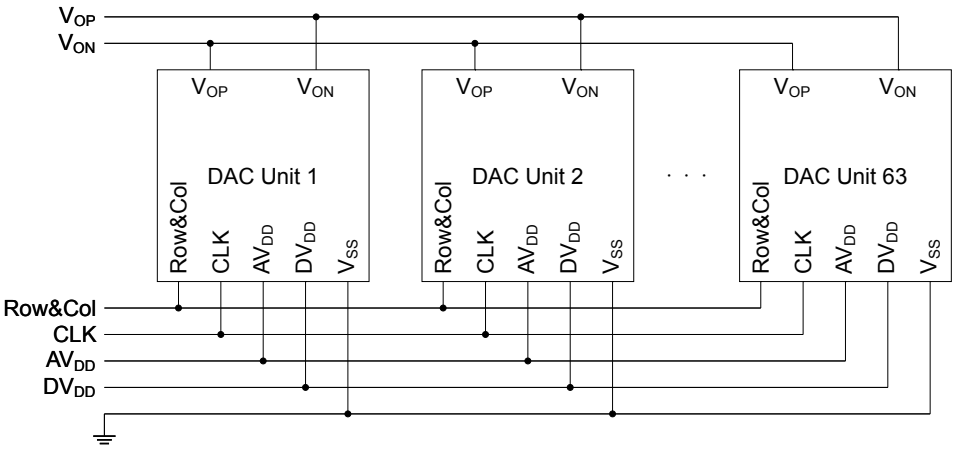


Figure 3.41: Complete 6-bit main DAC

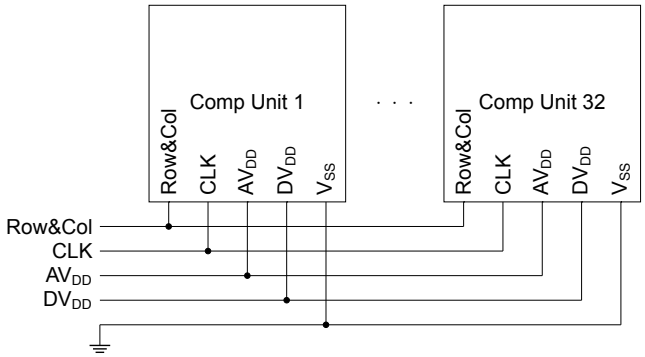


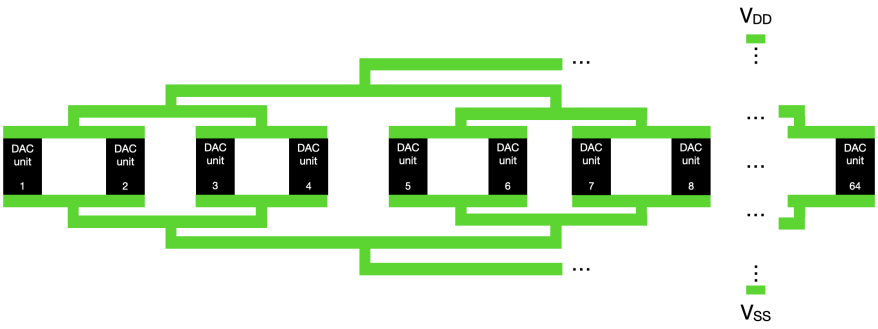
Figure 3.42: Complete compensation DAC

4

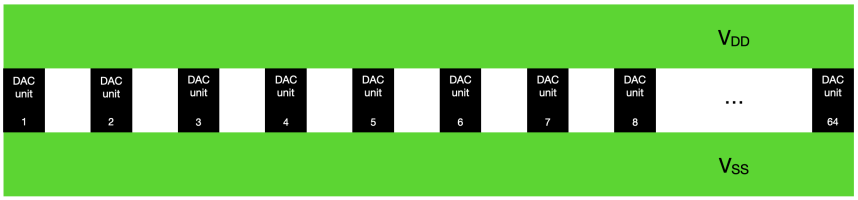
Layout considerations

4.1. Analog supply routing

So far, we have considered the effect of finite supply impedance as a single node in the schematic like in Figure 3.41. However, each of the cells in the DAC has to be wired up to the supply, introducing additional parasitic impedance between the units. This additional supply impedance is not compensated for by the compensation DAC. Hence, careful layout is required to reduce the impact of parasitics. Two approaches presented in Figure 4.1 were considered: a binary tree structure and a wide metal line (20um width). For the sake of comparison, the overall area of the supply wiring (green color) of the two methods is kept equal.



(a) Tree supply structure



(b) Wide wire

Figure 4.1: Analog supply routing

In the 180nm process, six layers of metal are available. M1 to M5 layers have the same properties and sheet resistance of (70mΩ). M6 is a special metal layer with increased thickness and a significantly reduced sheet resistance (7mΩ). For that reason, it is used for supply routing.

The simulation results in Figure 4.2 show that for the same area, using a wide metal line yields better results than a tree connection, which is a standard way to route supply and ground connections in current-steering DACs [23]. This can be explained by the fact that in current-steering DACs, all units pull current from the supply, and this current is only steered one way or another. Such configuration results in uniform voltage drop across all tree branches. In resistive DACs, only enabled units draw current from the supply, which means that different voltage drops occur along tree branches depending on the input code. Supply and ground connections were implemented as a 20um wide metal connection. It was verified using extraction tools that tapping in the middle or on the edge of the supply line makes no difference.

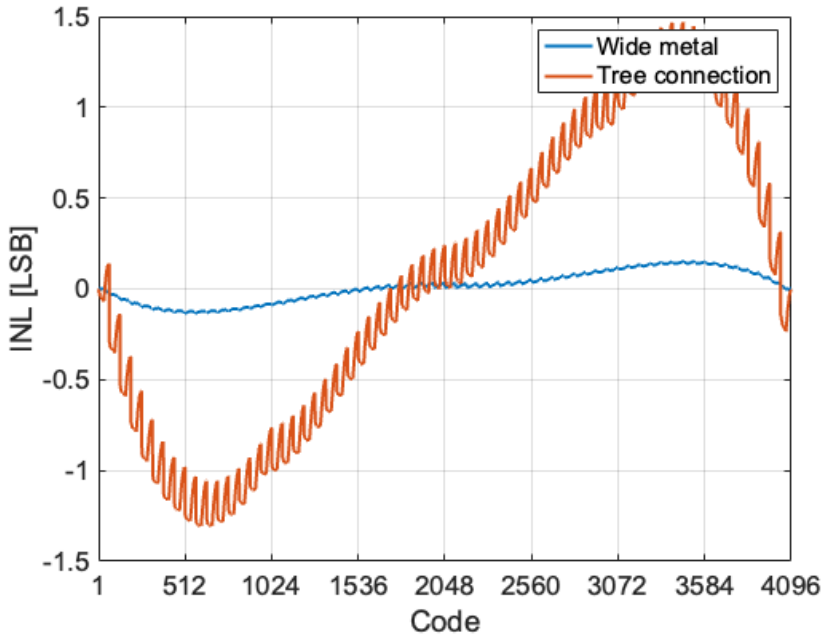


Figure 4.2: Supply wiring influence on INL

4.2. Digital supply routing

Assuming that latches reduce the timing spread to zero, it is important to note that the parasitic resistance of the digital supply can still impact the switching time of different units. Figure 4.3 illustrates the delay in switching (rising edge at the output of switch drivers) depending on the connection type for 20mΩ between each unit. When only a single connection on edge is used, the drivers close to it experience no degradation, but units further away have a smaller effective supply voltage available and switch slower, which can negatively affect performance. The resulting SDR can be calculated using the equation from [24]:

$$SDR = 3.01(N - 1) - 10\log_{10}(\sigma^2 f_{in} f_{clk}) - 9.03 \quad (4.1)$$

Let us assume that only MSB DAC contributes to timing accuracy and that the input signal is at Nyquist frequency. The RMS value of the timing spread for the edge connection in Figure 4.3 equals $\sigma = 9.7$ps, using Equation 4.1 we obtain SDR of 63dB, which is sufficient. If a higher linearity is needed, connecting in the middle lowers the maximum delay by a factor of 4. The same can be achieved by connecting from both sides. To minimize delays, supply resistance should be lowered by using a low-resistivity metal layer, stacking multiple metals, or using multiple taps along the supply rails.

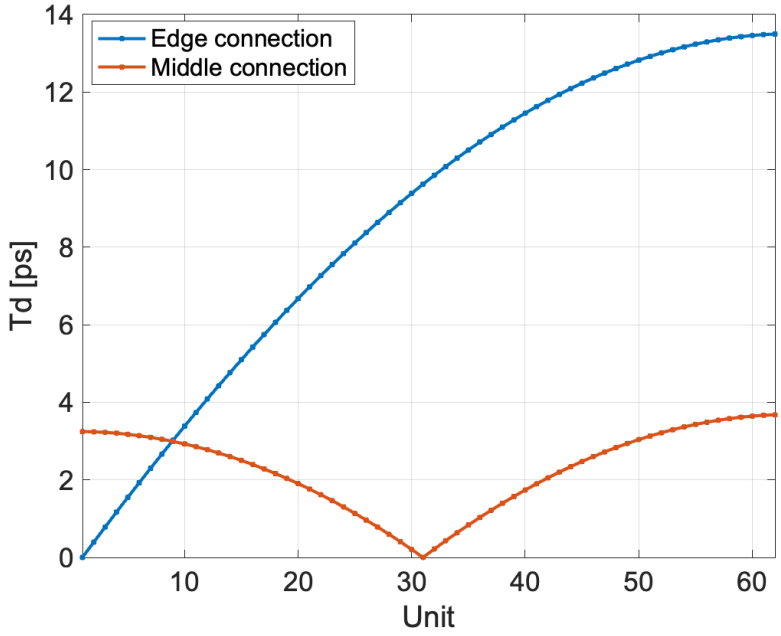


Figure 4.3: Digital supply parasitics influence

4.3. Output tree

Output trees are necessary to ensure equal weights and arrival times of the outputs of separate DAC units to the common output node. As the positive and negative outputs both need to be routed simultaneously, and their lines cross, it is required to use two metal layers for differential outputs, which already introduces imbalance: one output will be wired with a metal closer to the substrate, resulting in larger parasitic capacitance. Besides parasitic capacitors from each output node to the ground, the differential capacitor between two nodes also matters and should be minimized. To do that, we can increase the vertical distance between two outputs; one metal line is skipped between positive and negative outputs, effectively reducing the capacitance. Parasitic capacitance at the output node can also cause a timing mismatch between LSB and MSB DACs. Any capacitor at the output nodes of the LSB DAC will produce an RC time constant with R_{ATT} resistors. The excessive delay between MSB and LSB

outputs will cause a noise floor to rise at higher frequencies. This can be intuitively explained: for a brief moment, the 12-bit DAC is actually only 6-bit (because LSBs are lagging behind). This phenomenon is illustrated in Figure 4.4. The effect becomes significant for parasitics value of 10 pF, and below 1 pF is negligible. Keeping the parasitic capacitance smaller than 1pF is achievable in layout, but lowering it more is not straightforward.

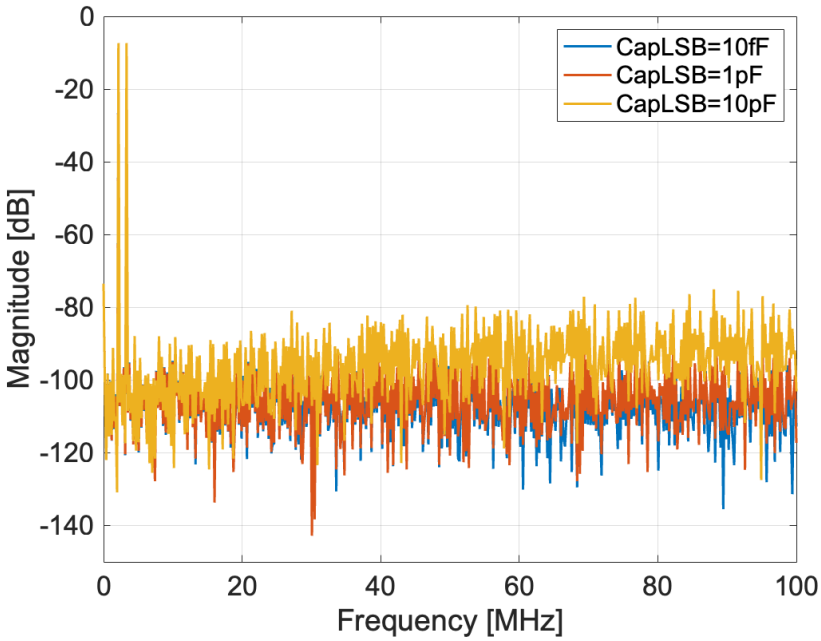


Figure 4.4: MSB - LSB timing mismatch

The influence of supply routing and output tree is verified in Figure 4.5. With compensation enabled, linearity of +- 0.38 LSB is achieved. This result clearly shows the effectiveness of our approach.

4.4. Clock tree

To ensure that the global clock signal reaches every DAC unit at the same time, a clock network must be designed. The most straightforward way to accomplish this is by using balanced, binary trees [23]. To minimize RC delay, a clock tree with 256 outputs was designed using M6 (top metal, low resistivity, and low capacitance to substrate). Each of the two 6-bit sub-DACs requires 96 clock inputs, of which 64 are for the main DAC, and 32 are for the compensation DAC. Therefore, a total of 192 clock outputs are necessary. The remaining 64 outputs, which are loaded with dummy latches, are used to construct a balanced tree. Every output of the tree is loaded with one latch and one minimum-size inverter, which, together with the tree itself, is a large load to drive. It is possible to alleviate this problem by adding intermediate clock buffers along the tree. However, adding buffers also inherently introduces mismatches in different clock paths. For that reason, only a single big buffer was used to drive the whole clock

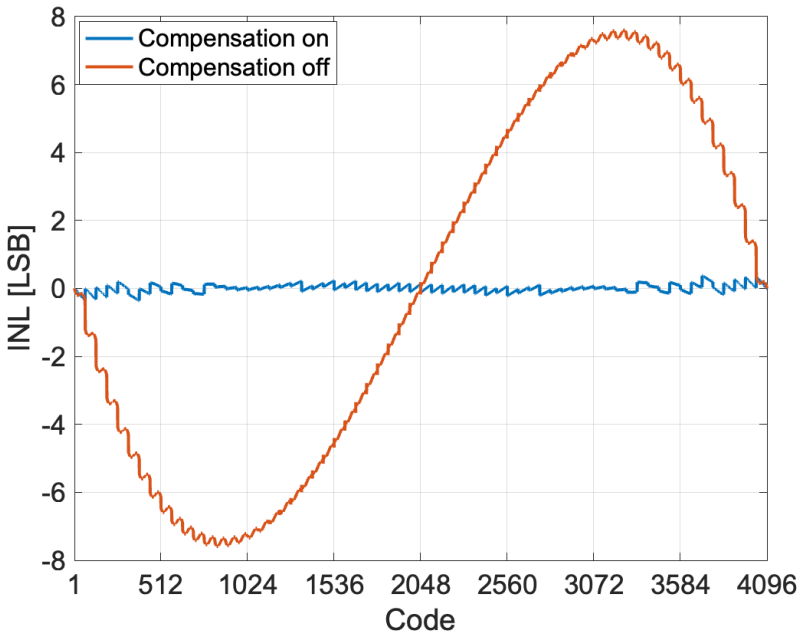


Figure 4.5: Static linearity with output tree and supply routing included

network. Delay in clock signals of MSB and LSB DACs will also result in a rising noise floor as in Figure 4.4.

4.5. Layout gallery

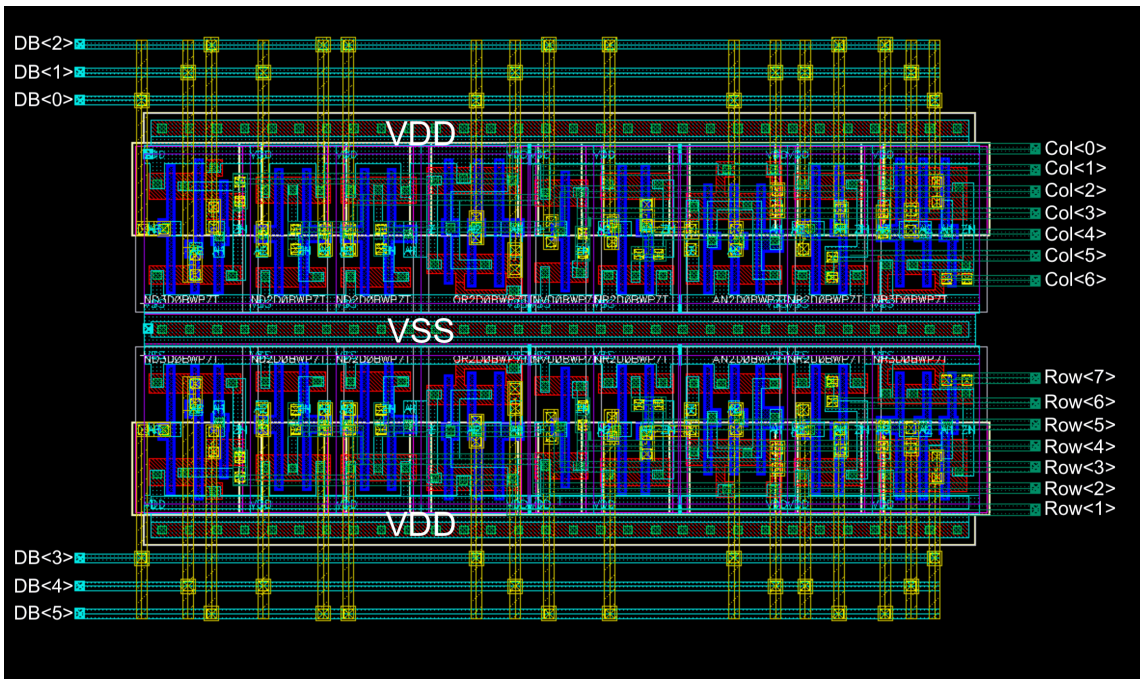


Figure 4.6: B2T core circuit

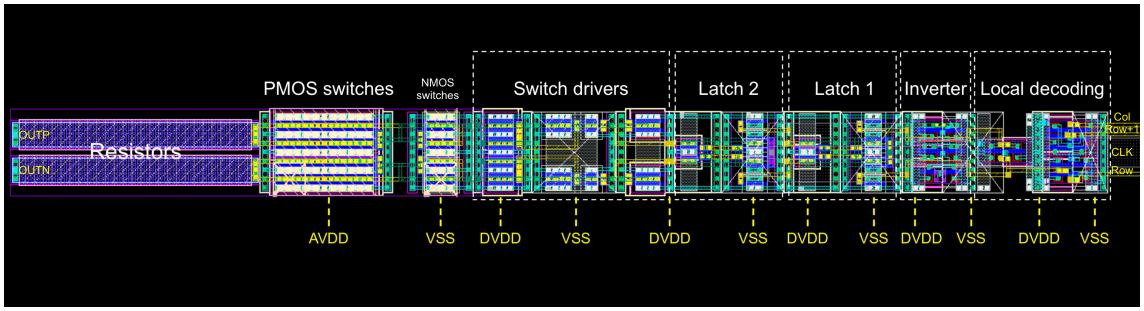


Figure 4.7: Complete DAC unit

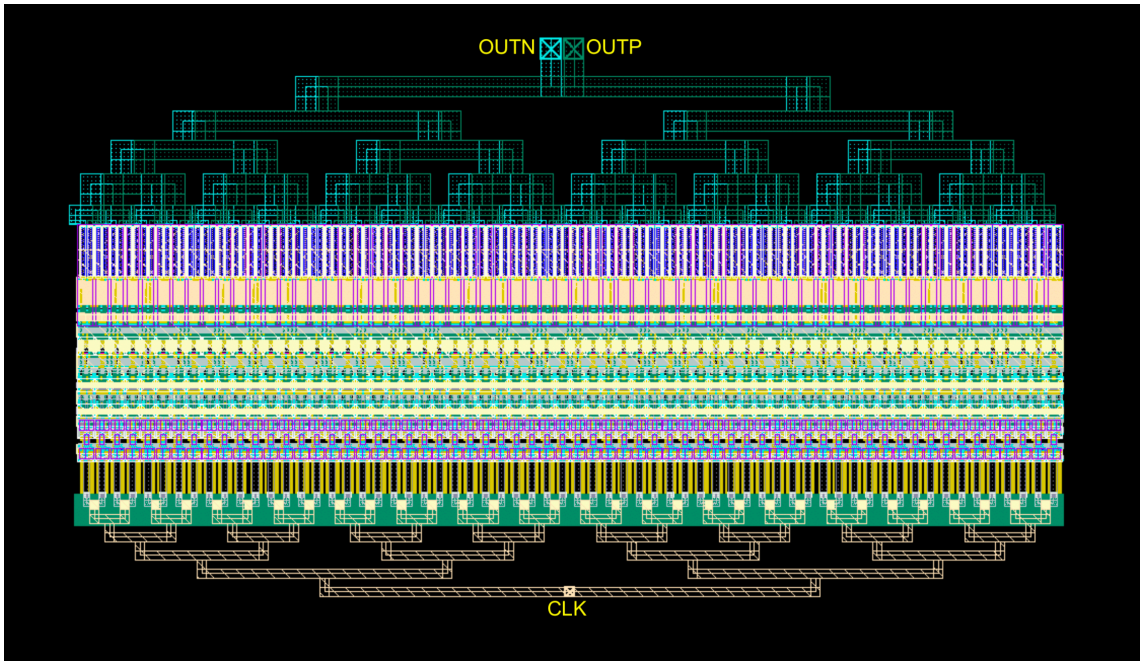


Figure 4.8: 6-bit subDAC

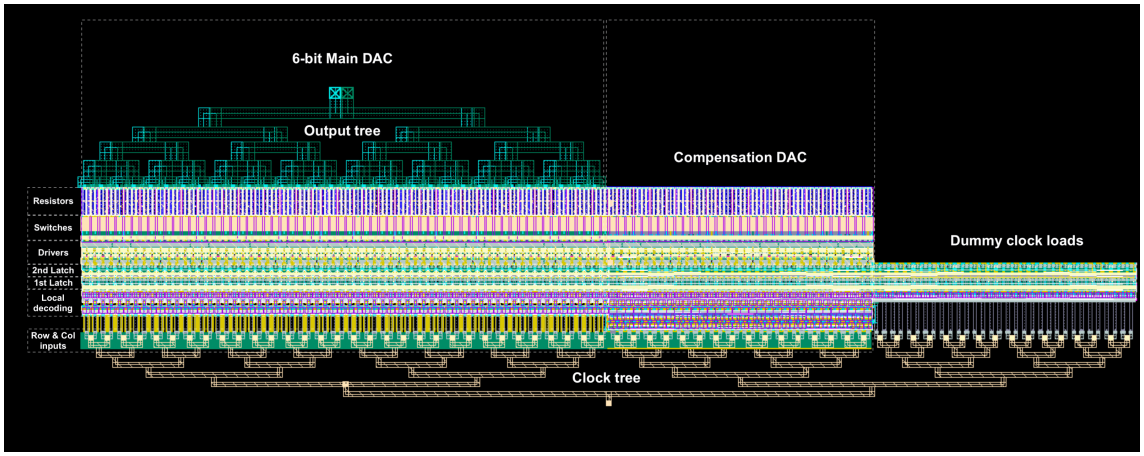


Figure 4.9: 6-bit subDAC including compensation

The complete 6-bit DAC, including the compensation DAC next to it (Figure 4.9), has dimensions of 230um x 750um.

5

Simulation results

5.1. Supply current linearization

The compensation technique introduced in Chapter 3 aimed to linearize the total supply current to improve linearity. In Figures 5.1 (low-frequency input) and 5.2 (high-frequency input), the supply current of the complete extracted MSB DAC is plotted. The supply current shows a significant reduction in its variation. The improvement is especially significant at low frequencies. For high frequencies, shown in Figure 5.2, switching of many DAC units at the same time results in large current spikes.

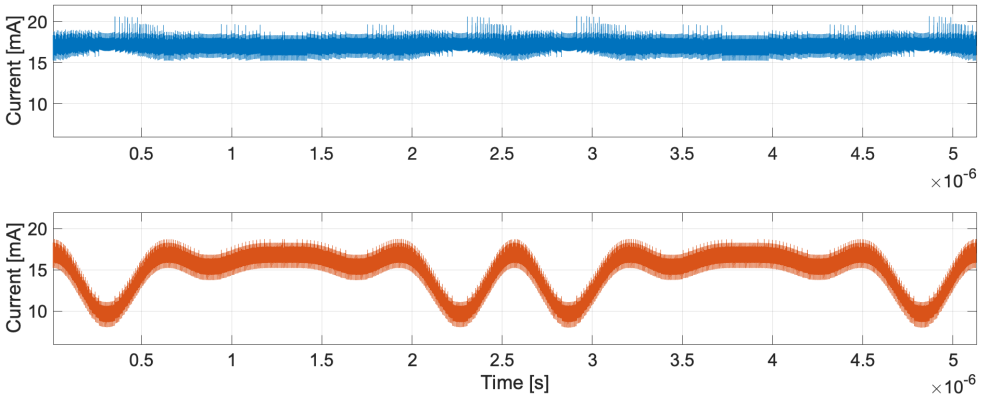


Figure 5.1: MSB supply current low frequency input (blue compensated, red uncompensated) - extracted

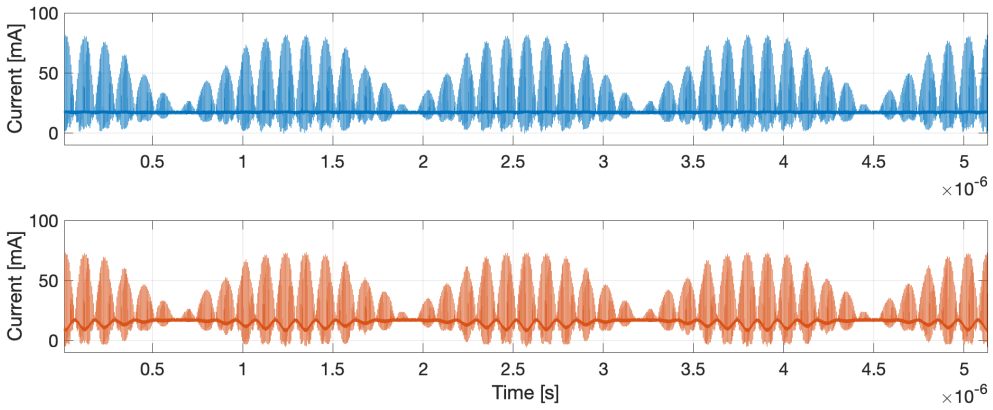


Figure 5.2: MSB supply current Nyquist frequency input (blue compensated, red uncompensated) - extracted

5.2. Dynamic performance

To verify dynamic performance, a two-tone FFT simulation was performed. An oversampling ratio of 128 was chosen to capture settling nonidealities and glitches in the output signal. The relative tolerance of the simulator was set to 0.0001 to increase simulation accuracy. Schematic level and extracted simulation results are plotted in Figures 5.3 and 5.4, respectively.

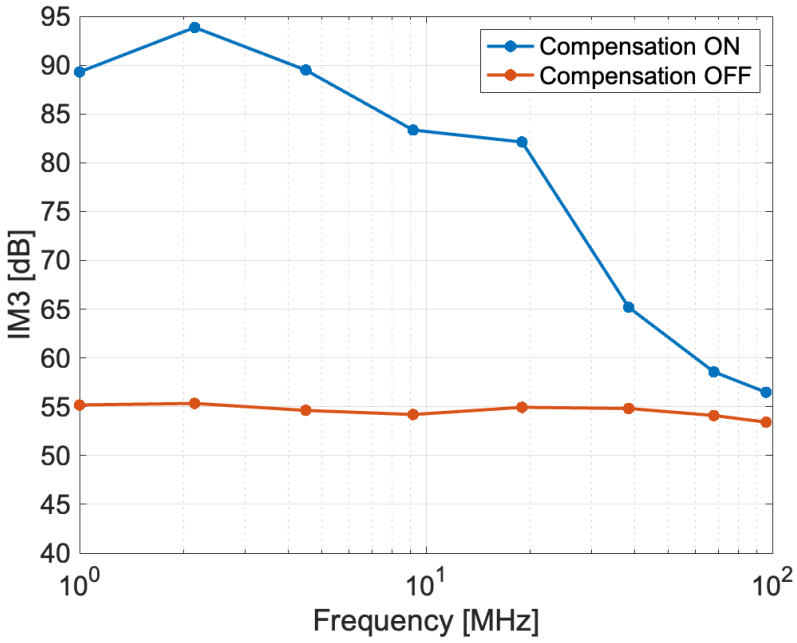


Figure 5.3: IM3 across frequency - schematic

Compensation significantly improves linearity for lower frequencies (below 10 MHz), and IM3 better than 75 dB is achievable in schematic and extracted simulations. Beyond 20 MHz, a degradation in performance is observed. Interestingly, compensation

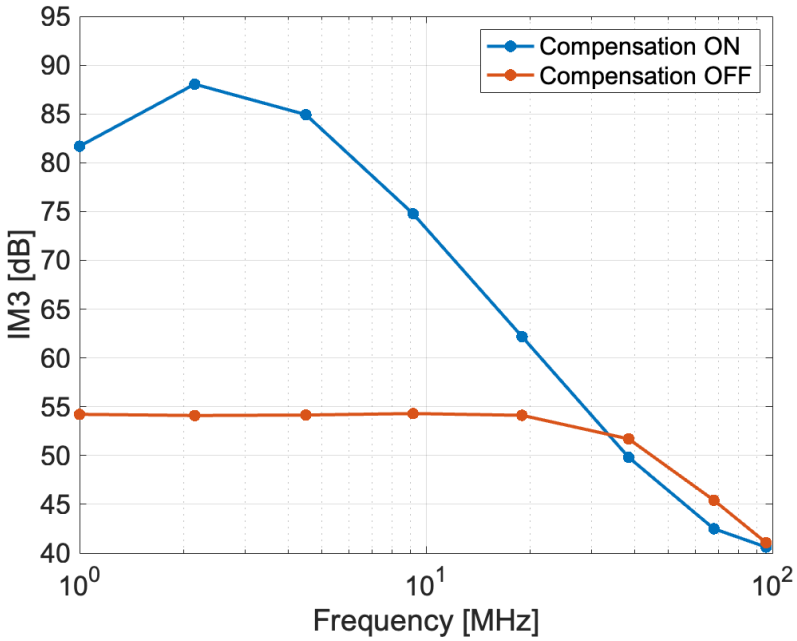
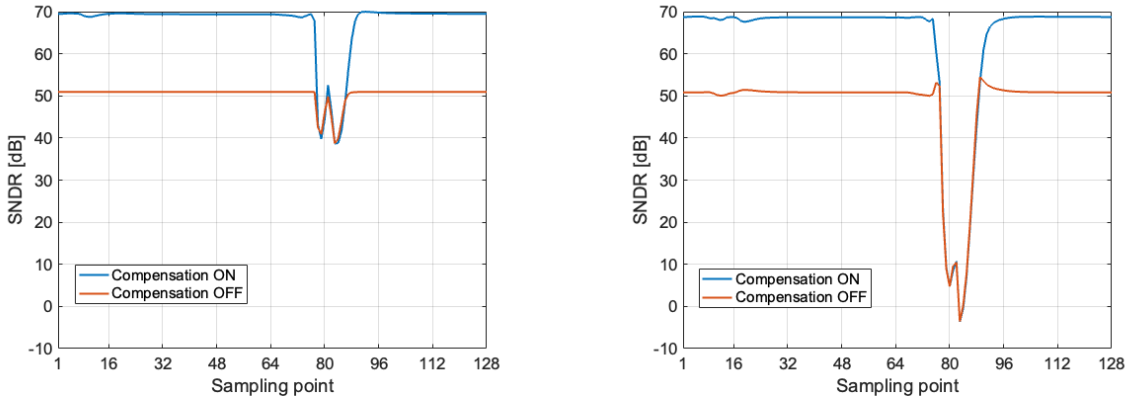


Figure 5.4: IM3 across frequency - extracted

makes linearity worse in extracted simulation at higher frequencies, likely due to increased switching activity.

128 samples per DAC period were taken, and SNDR can be calculated separately for each sample. This way, it is possible to calculate worst-case SNDR, for example, if a sample is taken each period only when the DAC is settling. Low-frequency and high-frequency inputs are compared in Figures 5.5.



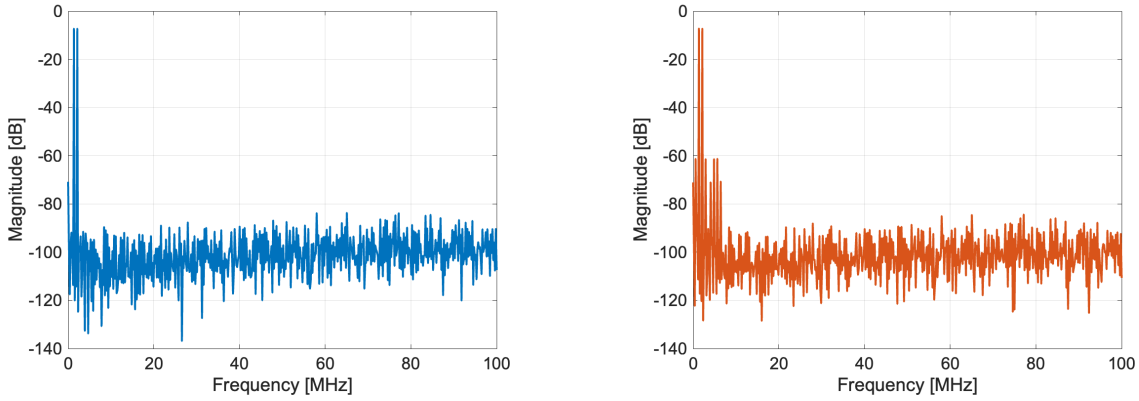
(a) Low frequency input signal

(b) Nyquist rate input signal

Figure 5.5: SNDR for different sampling moments - extracted

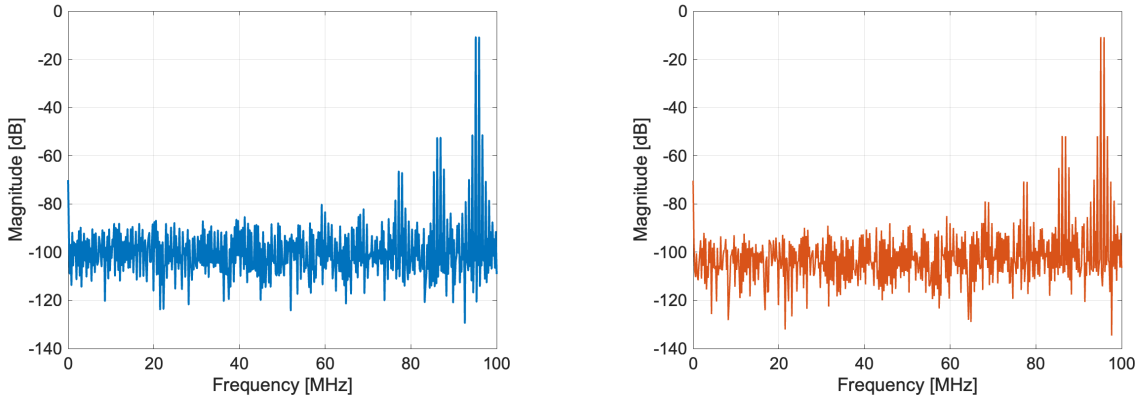
In both cases, compensation improves SNDR by roughly 20 dB. However, the performance significantly drops for points 70-90. At that moment, the output of the DAC is changing. The output waveform of the DAC is a continuous signal. Therefore, linearity is determined by aspects such as nonlinear settling time (slewing). Although excellent low-frequency linearity is reported, high-frequency results could be improved.

To improve high-frequency behavior, switching speed should be maximized because nonlinear slewing time is limiting the performance. Switches should be as small as possible, and big drivers should be used. This, however, leads to contradictions because in Section 3.3, switches were sized big to improve static linearity. It becomes clear that dynamic effects take over at high frequency and start dominating over static effects. Finally, in Figures 5.6 and 5.7, output spectra of the DAC output voltage are plotted for low and high frequencies.



(a) Compensation ON **(b)** Compensation OFF

Figure 5.6: Output spectrum low frequency input - extracted



(a) Compensation ON **(b)** Compensation OFF

Figure 5.7: Output spectrum high frequency input - extracted

Enabling compensation improves the SNDR from 50.7 dB to 64.0 dB at low frequencies. The noise floor rises at higher frequencies, which points out issues with matching the MSBs and LSBs timing. At higher frequencies, the performance is worse, regardless of whether compensation is enabled or disabled. In both cases, an SNDR of 37.9 dB is simulated.

5.3. Power consumption simulation

Lastly, the average power consumption of the extracted circuit is simulated for a Nyquist frequency input signal. The results are shown in Table 5.1 below:

Compensation	Analog power [mW]	Digital power [mW]
ON	97.00	17.73
OFF	88.40	15.84

Table 5.1: Power consumption summary

Analog power accounts for MSB and LSB DAC cores, and digital power accounts for all digital logic, latches, switch drivers, and the clock buffer. Enabling compensation results in only 10% higher total power consumption.

6

Conclusions and future work

6.1. Thesis outcome and findings

The purpose of this thesis was to explore the design of resistive DACs for Ethernet applications. The study concluded that resistive DACs are significantly more power-efficient compared to current-steering DACs. As technology continues to scale towards lower supply voltages, resistive DACs are expected to become even more advantageous.

In Chapter 2, nonlinearity mechanisms have been analyzed. It has been proven that the magnitude of distortion caused by the code-dependent supply current is proportional to the supply and load impedance ratio. The 3rd order distortion component is by far the most dominant one.

In Chapter 3, the circuit implementation of the complete converter was discussed. Firstly, a compensation technique to linearize the supply current was proposed. Using an extra component connected in parallel to the main DAC, it is possible to make the supply current constant, which results only in gain error and not distortion. It is worth noting that the increase of power consumption due to supply current linearization depends on the input signal probability distribution. Complete compensation is not feasible due to the complexity and size of the resulting compensation DAC. The 12-bit DAC is segmented into two 6-bit unary DACs, each having its own compensation DAC. To have control over MSB and LSB matching, a way to change LSB DAC's strength is proposed. The row and column method for the binary-to-thermometer decoder was used to simplify wiring. Lastly, latches are added to every DAC unit to improve the timing accuracy of the system. Although using a single latch yields sufficient results, an additional latch is added for robustness.

Chapter 4 briefly explained the layout and concentrated on analog and digital supply, output, and clock trees. Implications of layout on static and dynamic linearity are discussed. We found that, as opposed to current-steering DACs, using a single wide metal line for the supply routing yields better linearity than a binary tree. Additionally, pictures of the most important circuit blocks are shown.

Finally, the simulation results were presented in Chapter 5 of the report. The compensation technique used in the simulation demonstrated excellent performance at

low frequencies, with IM3 higher than 85 dB. On the other hand, there is still room for improvement regarding high-frequency performance.

6.2. Comparison with literature

Finally, our design is compared with recent resistive DAC papers in Table 6.1. We achieve the best low-frequency linearity and the largest output swing (compared to V_{DD} used).

Specifications	This work	[1]	[2]
Num. of bits	12	9	8
Supply voltage [V]	1.8	1.2	0.95
Clock speed [GS/s]	0.2	5.6[1]	72
Technology	180nm	22nm	4nm
IM3 low frequency [dB]	88	58[2]	57[2]
IM3 high frequency [dB]	41	50[2]	-
Output swing [Vdpp]	1.76	1.01	0.92
Load impedance [Ω]	100	100	100

Table 6.1: Comparison with different resistive DACs
[1] Up to 11.2, but no linearity results reported for that speed.
[2] Single tone test.

It is worth noting that the difference in output swing between our design and [1] works in our favor as $(1.76/1.01)^2 = 3.04$ which is equivalent to 9.65 dB in IM3.

6.3. Future work

6.3.1. Verification in silicon

Although most of the layout is finished, the top-level layout is still lacking. The first steps toward the tape-out have already been made. A complete list of pins is ready, and the package has already been chosen. An external FPGA board will feed data into the chip using LVDS (Low-Voltage Differential Signaling).

6.3.2. Possible improvements

There are 3 straightforward improvements. Firstly, power consumption can be lowered. Because programmable LSB strength was needed, the current consumption of the LSB DAC is twice that of the MSB DAC. The total (DC with compensation enabled) supply current drawn by the converter equals 54 mA. This value can be lowered three times to 18 mA if the LSBs are scaled down and not attenuated. Scaling the LSBs down may not be completely possible as, ideally, 64x larger resistors would be needed.

Secondly, more emphasis should be put on the dynamic performance. Switches could be made almost three times smaller without sacrificing much static linearity. At

the same time, switch drivers could be made larger to decrease the nonlinear slewing time significantly.

Thirdly, means to control the output impedance of the DAC should be implemented.

6.3.3. Tri-level DAC

To lower power consumption even more, a different DAC architecture can be implemented which pushes the DAC towards a class AB behavior. Figure 6.1 shows a Tri-level resistive DAC. Below are also equations describing how R_A, R_B, and R_{MID} change across input code. Depending on the input code, resistors are also connected between differential outputs via a common middle node (MID). This node is a floating node that, by connecting its left and right components, ends up exactly in the middle ($V_{DD}/2$). For completeness, Figure 6.2 shows the output voltage and supply current.

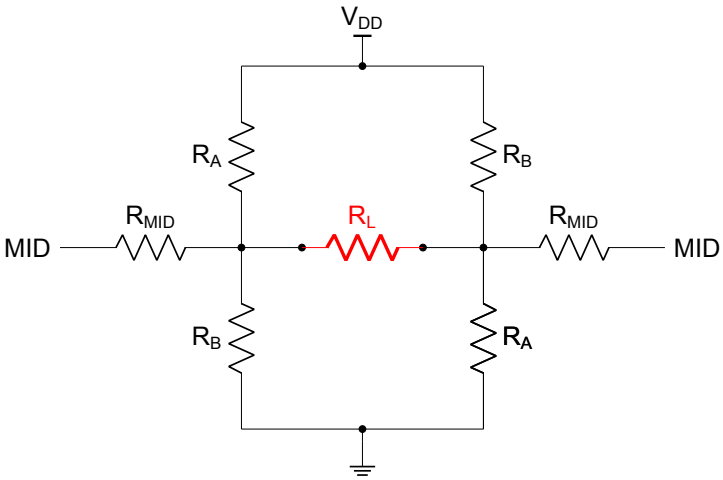


Figure 6.1: Tri-level DAC

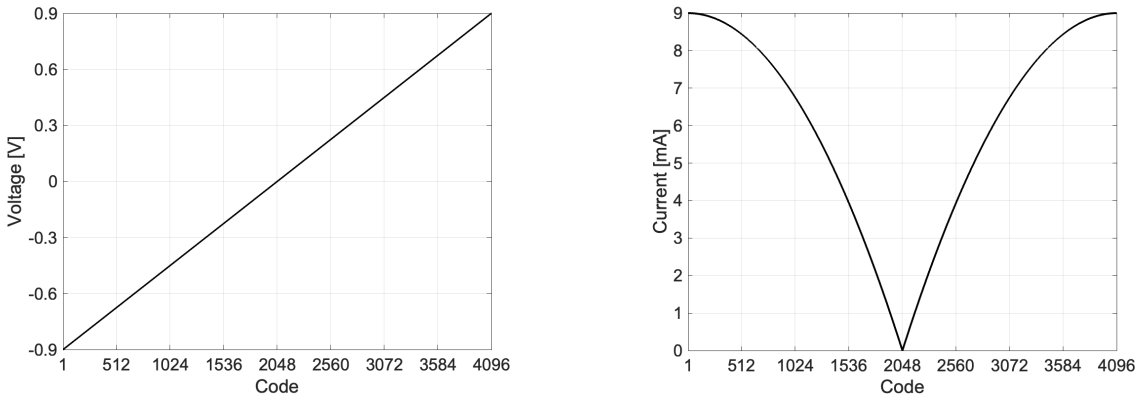
$$D_{IN} = \text{-K,-K+1, ..., K-1, K} \tag{6.1}$$

$$R_T = \frac{R_L K}{2} \tag{6.2}$$

$$R_A = \frac{R_T}{\max(0, D_{IN})} \tag{6.3}$$

$$R_B = \frac{R_T}{\max(0, -D_{IN})} \tag{6.4}$$

$$R_{MID} = \frac{R_T}{K - \text{abs}(D_{IN})} \tag{6.5}$$

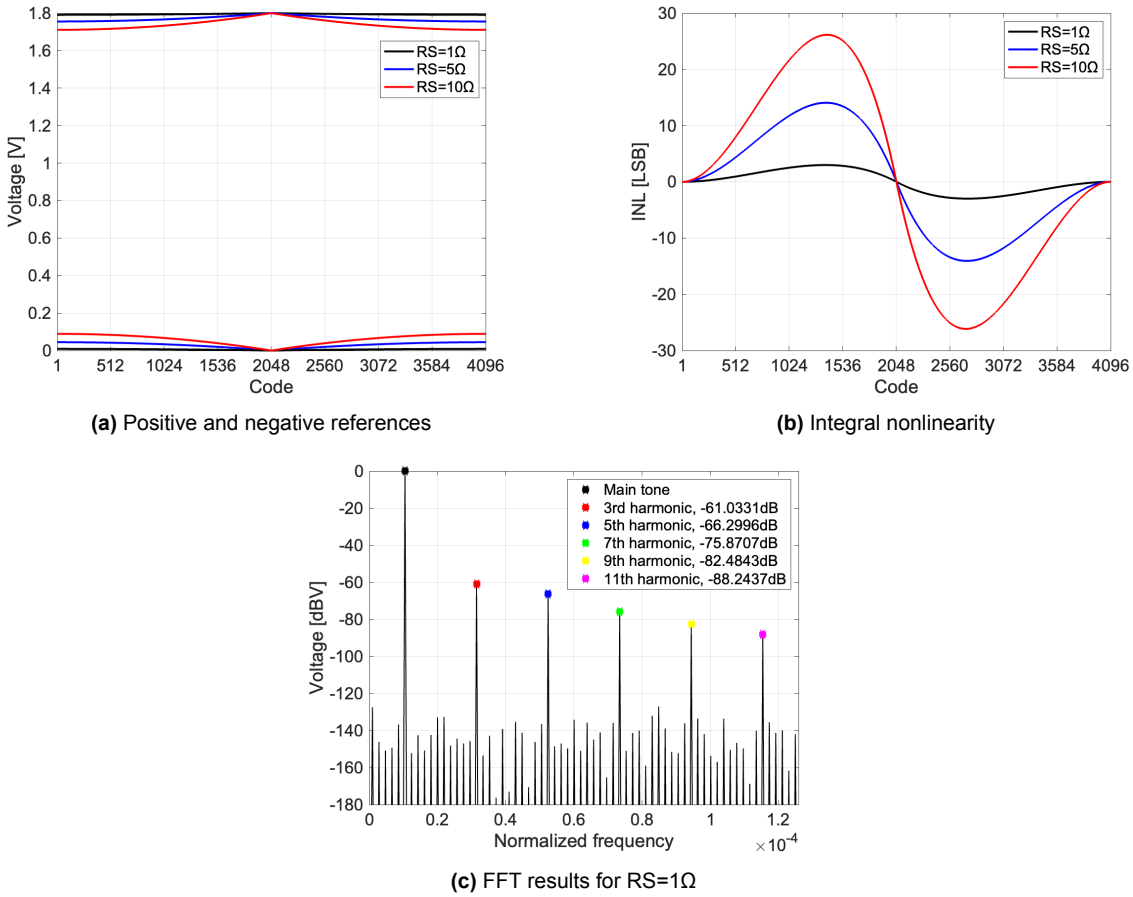


(a) Output voltage

(b) Supply current

Figure 6.2: Output voltage and supply current across digital input code for Tri-level DAC

In Figure 6.3, the impact of finite supply impedance on performance of Tri-level DAC can be observed. When compared with Figure 2.5, linearity is 2.5 times better for the same supply impedance.



(a) Positive and negative references

(b) Integral nonlinearity

(c) FFT results for RS=1Ω

Figure 6.3: Impact of finite supply impedance fro Tri-level DAC

This type of DAC can also be compensated using the same method as one introduced in Section 3.1. Doing that would make the current drawn from the supply a

constant 9 mA, reducing the power dissipation of the DAC 2x compared to the design described in this thesis. The challenge of this design, however, is a proper design of the switches at mid-level.

References

[1] Hossein Ghafarian et al. “A 9-bit, 45 mW, 0.05 mm2 Source-Series-Terminated DAC Driver With Echo Canceller in 22-nm CMOS for In-Vehicle Communication”. In: *IEEE Solid-State Circuits Letters* 4 (2021), pp. 10–13. DOI: 10.1109/LSSC.2020.3048752.

[2] Timothy Dickson et al. “A 72GS/s, 8-bit DAC-based Wireline Transmitter in 4nm FinFET CMOS for 200+Gb/s Serial Links”. In: *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 2022, pp. 28–29. DOI: 10.1109/VLSITechnologyandCir46769.2022.9830421.

[3] Marcel A. Kossel et al. “8.3 An 8b DAC-Based SST TX Using Metal Gate Resistors with 1.4pJ/b Efficiency at 112Gb/s PAM-4 and 8-Tap FFE in 7nm CMOS”. In: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 64. 2021, pp. 130–132. DOI: 10.1109/ISSCC42613.2021.9365784.

[4] Christian Menolfi et al. “A 112Gb/S 2.6pJ/b 8-Tap FFE PAM-4 SST TX in 14nm CMOS”. In: *2018 IEEE International Solid-State Circuits Conference - (ISSCC)*. 2018, pp. 104–106. DOI: 10.1109/ISSCC.2018.8310205.

[5] Chi-Hung 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* 44.12 (2009), pp. 3285–3293. DOI: 10.1109/JSSC.2009.2032624.

[6] Douglas A. Mercer. “Low-Power Approaches to High-Speed Current-Steering Digital-to-Analog Converters in 0.18-μm CMOS”. In: *IEEE Journal of Solid-State Circuits* 42.8 (2007), pp. 1688–1698. DOI: 10.1109/JSSC.2007.900279.

[7] Hossein Ghafarian and Friedel Gerfers. “Analysis and compensation technique canceling non-linear switch and package impedance effects of a 3.2GS/S TX-DAC”. In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017, pp. 1172–1175. DOI: 10.1109/MWSCAS.2017.8053137.

[8] Helia Ordouei, Hossein Ghafarian, and Friedel Gerfers. “Analysis of Package Impedance Effects on the Linearity of Source Series Terminated DACs”. In: *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2018, pp. 250–253. DOI: 10.1109/MWSCAS.2018.8623920.

[9] David Marche and Yvon Savaria. “Modeling $R-2R$ Segmented-Ladder DACs”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 57.1 (2010), pp. 31–43. DOI: 10.1109/TCSI.2009.2019396.

[10] Waleed Khalil Xuewen Jiang Bobby Nikjou and Syed Nagvi. *Differential digital-to-analog converter*. U.S. Patent US20040257254A1, Dec. 2004.

[11] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. 1st ed. USA: McGraw-Hill, Inc., 2000. ISBN: 0072380322.

[12] Carmine Cozzolino. *Super source follower output impedance enhancement*. U.S. Patent US7944303B2, May 2011.

[13] Wenjuan Guo et al. “An Area- and Power-Efficient I_{ref} Compensation Technique for Voltage-Mode $R-2R$ DACs”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.7 (2015), pp. 656–660. DOI: 10.1109/TCSII.2015.2406351.

[14] Didier Davino Thierry Soude Michel Cuenca. *Current compensation for digital-to-analog converter*. U.S. Patent US20090153380A1, Dec. 1991.

[15] Mohammad Sadeghifar. *Studies on Selected Topics in Radio Frequency Digital-to-Analog Converters*. Nov. 2019. ISBN: 9789176850305. DOI: 10.3384/diss.diva-160893.

[16] Chi-Hung Lin and K. Bult. “A 10-b, 500-MSample/s CMOS DAC in 0.6 mm^2”. In: *IEEE Journal of Solid-State Circuits* 33.12 (1998), pp. 1948–1958. DOI: 10.1109/4.735535.

[17] Sining Pan. “Resistor-based Temperature Sensors in CMOS Technology”. PhD thesis. Delft University of Technology, 2021.

[18] 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* 24.5 (1989), pp. 1433–1439. DOI: 10.1109/JSSC.1989.572629.

[19] Hossein Ghafarian and Friedel Gerfers. “A digital calibration technique canceling non-linear switch and package impedance effects of a 1.6GS/s TX-DAC in 28 nm CMOS”. In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017, pp. 1–4. DOI: 10.1109/ISCAS.2017.8050689.

[20] Hossein Ghafarian, Helia Ordouei, and Friedel Gerfers. “Impedance Calibration Technique Canceling Process and Temperature Variation in Source Terminated DAC Drivers in 22nm FDSOI”. In: *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2019, pp. 113–116. DOI: 10.1109/MWSCAS.2019.8884867.

[21] Khawaja Qasim Maqbool et al. “EMI common-mode (CM) noise suppression from self-calibration of high-speed SST driver using on-chip process monitoring circuit”. In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017, pp. 1–4. DOI: 10.1109/ISCAS.2017.8050879.

[22] T. Miki et al. “An 80-MHz 8-bit CMOS D/A converter”. In: *IEEE Journal of Solid-State Circuits* 21.6 (1986), pp. 983–988. DOI: 10.1109/JSSC.1986.1052639.

[23] Klaas Bult. *High Speed CMOS DACs, MEAD course*. 2019.

[24] K. Doris, A.H.M. Roermund, van, and D.M.W. Leenaerts. *Wide-bandwidth high-dynamic range D/A converters*. English. The Kluwer international series in engineering and computer science. SECS. Germany: Springer, 2006. ISBN: 978-0-387-30415-1. DOI: 10.1007/0-387-30416-9.

A

Detailed Output Voltage and Distortion Derivation

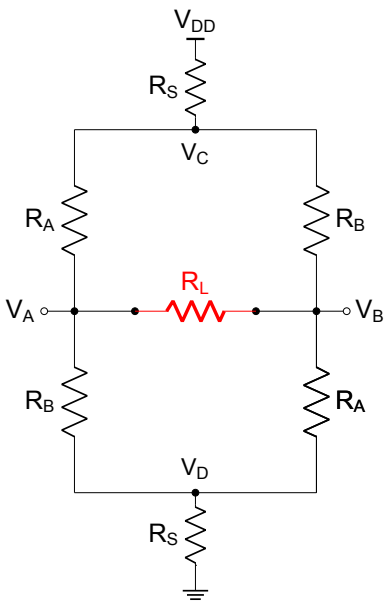


Figure A.1: Simplified resistive DAC circuit with supply and ground impedance added

$$D_{IN} = \text{-K,-K+1, ..., K-1, K} \tag{A.1}$$

$$R_T = R_L K \tag{A.2}$$

$$R_A = \frac{R_T}{K + D_{IN}} \tag{A.3}$$

$$R_B = \frac{R_T}{K - D_{IN}} \tag{A.4}$$

$$\begin{cases} \frac{V_C - V_{DD}}{R_S} + \frac{V_C - V_A}{R_A} + \frac{V_C - V_B}{R_B} = 0 \\ \frac{V_D}{R_S} + \frac{V_D - V_A}{R_B} + \frac{V_D - V_B}{R_A} = 0 \\ \frac{V_A - V_C}{R_A} + \frac{V_A - V_D}{R_B} + \frac{V_A - V_B}{R_L} = 0 \\ \frac{V_B - V_C}{R_B} + \frac{V_B - V_D}{R_A} + \frac{V_B - V_A}{R_L} = 0 \end{cases} \tag{A.5}$$

$$V_{OUT} = V_A - V_B = V_{DD} \frac{D_{IN} R_L K}{-D_{in}^2 + 2K^2(R_L + 2R_S)} \tag{A.6}$$

Applying a Taylor series expansion to Equation A.6 we have:

$$\begin{aligned} V_{OUT} = {} & D_{IN} \frac{V_{DD} R_L K}{2K^2(R_L + 2R_S)} \\ & + D_{IN}^3 \frac{V_{DD} R_L R_S K}{2K^4(R_L^2 + 4R_L R_S + 4R_S^2)} \\ & + D_{IN}^5 \frac{V_{DD} R_L R_S^2 K}{2K^6(R_L^3 + 6RL_2 R_S + 12R_L R_S^2 + 8R_S^3)} \\ & + D_{IN}^7 \frac{V_{DD} R_L R_S^3 K}{16K^8(R_L^4 + 8R_L^3 R_S + 24R_L^2 R_S^2 + 32R_L R_S^3 + 16R_S^4)} \end{aligned} \tag{A.7}$$

Now HD_3, HD_5 and HD_7 can be calculated in the following way:

$$HD_3 = \frac{R_S\, a^2}{4\ (RL + 2\, R_S)} \tag{A.8}$$

$$HD_5 = \frac{{R_S}^2\, a^4}{16\, (R_L + 2\, R_S)^2} \tag{A.9}$$

$$HD_7 = \frac{{R_S}^3\, a^6}{64\, (R_L + 2\, RS)^3} \tag{A.10}$$

where 'a' is the input range that can take values from 0 to 1.