2007

Dynamic calibration of current-steering DAC

Chao Su
Iowa State University

Dynamic calibration of current-steering DAC

by

Chao Su

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical and Computer Engineering

Program of Study Committee:
R.L. Geiger, Major Professor
Degang Chen
Robert Weber
Aleksandar Dogandžić
Mervyn Marasinghe

Iowa State University

Ames, Iowa

2007

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The demand for high-speed communication systems has dramatically increased during the last decades. Working as an interface between the digital and analog world, Digital-to-Analog converters (DACs) are becoming more and more important because they are a key part which limits the accuracy and speed of an overall system. Consequently, the requirements for high-speed and high-accuracy DACs are increasingly demanding. It is well recognized that dynamic performance of the DACs degrades dramatically with increasing input signal frequencies and update rates. The dynamic performance is often characterized by the spurious free dynamic range (SFDR). The SFDR is determined by the spectral harmonics, which are attributable to system nonlinearities.

A new calibration approach is presented in this thesis that compensates for the dynamic errors in performance. In this approach, the nonlinear components of the input dependent and previous input code dependent errors are characterized, and correction codes that can be used to calibrate the DAC for these nonlinearities are stored in a two-dimensional error look-up table. A series of pulses is generated at run time by addressing the error look-up table with the most significant bits of the Boolean input and by using the corresponding output to drive a calibration DAC whose output is summed with the original DAC output. The approach is applied at both the behavioral level and the circuit level in current-steering DAC.

The validity of this approach is verified by simulation. These simulations show that the dynamic nonlinearities can be dramatically reduced with this calibration scheme. The simulation results also show that this calibration approach is robust to errors in both the width and height of calibration pulses.

Experimental measurement results are also provided for a special case of this dynamic calibration algorithm that show that the dynamic performance can be improved through dynamic calibration, provided the mean error values in the table are close to their real values.

CHAPTER 1. INTRODUCTION

1.1 Background

The telecommunication market has experienced unprecedented growth during the last decade. With the decreasing feature size of the transistor, over millions of transistors can be integrated into a single chip. The digital part becomes more important with growth while the analog part remains crucial. As an interface between the digital and analog world, digital-to-analog converters (DACs) have to meet the requirement for both sides[1]. Numerous types of DACs, such as decoder-based, binary-weighted, etc., have been designed to meet the requirement. With the development of submicron CMOS technologies, high-accuracy digital to analog converters are becoming increasingly important as the commonly used building block in communication systems. Furthermore, with increasing of the update rates, like those seen in submicron, CMOS technologies that show an increase in conversion speed up to Giga Hertz range, dynamic performance of such DACs at high frequencies is of particular interest. So, the requirement for high-speed and high-accuracy DACs has become urgent.

The DACs' performance deteriorates dramatically with increases in system clock rate and input signal frequency. To address this, many calibration approaches were reported in the literature to improve DAC performance. But, most of the approaches focused on static nonlinearity calibration, and few papers have reported on how to deal with dynamic calibration at high frequency ranges. In this dissertation, dynamic nonlinearity is analyzed and its effect on the output waveform is discussed. A novel dynamic error calibration approach is presented to compensate for nonlinearities and, therefore, to improve DACs' dynamic performance, especially at high frequency.

DACs convert a signal from a digital code to an analog signal proportional to the digital input. The output signal of the DACs can be voltage, current, or charge. As one of the most common analog interface circuits between the digital domain and analog domain, DACs can be used from low-speed, high-resolution audio application to high-speed, low-resolution video applications. Among these applications, low-sample rate, high-resolution architectures are dominated by oversampling architectures. But, this type of DACs can be only used in medium-to-low speeds. In other ranges the Nyquist-rate DACs dominate. Therefore, we only focus on Nyquist-rate DACs in the following study.

DACs are usually in the form of voltage or current output when they are designed to drive an external load. It is very easy to modify a current DAC into a voltage DAC with a resistive load. Throughout the years, the demand for high-speed and high-resolution has been increasing in communications. Among these DACs, current-steering DAC are often used for high-speed and high-resolution converters in advanced CMOS technologies because of its attractive features. First, current-steering DACs are easy to integrate into standard digital CMOS technologies. Second, current-steering DACs can delivery nearly all power to the output, so the converter is very power efficient.

In this thesis, we study the dynamic nonlinearities of DACs and present an approach that can improve the DACs’ dynamic performance at high-frequency, near-Nyquist rate by adding an extra compensation calibration DAC (CALDAC) output to the main DAC. The output of the CALDAC output is generated according to an error look-up table. It is shown that this approach can significantly improve DACs’ dynamic performance, especially at high frequency.

1.2 Thesis organization

This thesis is organized as six chapters. Chapter 1 is the introduction which gives a brief background of DACs and the thesis organization. Chapter 2 describes varied DAC architecture and concepts of DAC static and dynamic nonlinearies. Some circuits' level methods and calibration approaches are also reviewed in this chapter. In chapter 3, the basic idea of the dynamic calibration is given and the generation of the error used for dynamic nonlinearities is also described. Chapter 4 presents the prototype of a 15-bits, behavioral-mode, current-steering DAC and a 12-bits, transistor-level, current-steering DAC working as a main DAC in the calibration. The simulation results of the dynamic calibration are given in this chapter. The experimental measurement scheme, results, and the discussion about the results are provided in chapter 5. Insight into the shortcomings of this approach and future work that may be necessary are discussed in chapter 6. Finally, the references are cited.

CHAPTER 2. DAC ARCHITECTURE, NONLINEARITIES, AND CALIBRATION REVIEW

In signal processing and telecommunication systems, the digital-to-analog converters (DACs) are used to reconstruct the analog signal from arbitrary digital waveform. DACs are a key part which limits the accuracy and speed of the overall system [2][3]. The DACs can be generally divided into two main types according to the sampling frequency to input signal ratio: Nyquist-rate converters and oversampling converters [4].

1. Nyquist-rate converters:

For Nyquist-rate converters, output values have a one-to-one correspondence with a single input value. Each analog output level is a result of a single K-bit input word. However, this type of converter seldom operates near Nyquist rate for two major reasons. First, the dynamic performance degrades dramatically near the Nyquist rate. Second, the anti-aliasing filter design becomes much harder when DACs operate near the Nyquist rate, which requires the anti-aliasing filter to fall down very sharply. Typically, the Nyquist rate converters' sampling rate is about 1.5 to 10 times the Nyquist rate.

2. Oversampling converters

Oversampling converters can push the quantization noise of the converter out of the signal's bandwidth through oversampling (about 20 to 512 times faster) and filter it out by a following connected filter operation. As a result, the output's signal-to-noise ratio (SNR) can be increased with the larger of the oversampling ratio. The oversampling DAC's have become popular for high resolution, but are only used in medium-to-low speeds [5].

In this thesis, the candidate of study is high-frequency DACs; therefore, we focus on Nyquist-rate DACs in the following chapters.

2.1 Typical DAC circuits implementation

In this work, we focus our study on DACs suitable for high-speed and high-resolution application. There are three basic models of circuit technology to implement the DACs: voltage mode, current model, and charge-redistribution mode. We will give a brief introduction for each mode in the following.

2.1.1 Voltage mode

For the voltage mode, the output signal is given by voltage level. Typically, structure for this mode is the resistor-string. There are several types of architectures for the resistor-sting DAC. The most straightforward approach for realizing the N bits DAC is through decoder-based converters which create 2^N reference signals and pass the appropriate signal to the output. The reference voltage V_{ref} is divided by the resistor-string and form different weights. The switch network is connected in a tree-like decoder. The resistor-string structure is simple, but the delay caused by the switch network severely limits its speed [4]. Though logic can be used for the decoder to make it faster, the improvement is still moderate. When the number of bits becomes large, the resistors and switches number increases significantly, which will occupy a large area. In broadband applications, the Opamp design is becoming difficult. Moreover, resistor matching is also a significant issue.

2.1.2 Charge mode

The charge mode DAC is usually the charge-redistribution DAC implemented with a switched-capacitor (SC) technique. The basic idea is illustrated as Fig. 2.1 [4]. For a N-bit DAC, the most significant bit (MSB) capacitor is 2^{N-1} times of the least significant bit (LSB). For an input word X_i=(b_1, b_2 ,… ,b_N,), where $b_i \in \{0,1\}$, the corresponding switches will be turned on to send different weighted voltage to the output V_{out}, and $V_{out} = -\sum_{i=1}^{N} b_i * \frac{C_i}{C_o} V_{in}$.

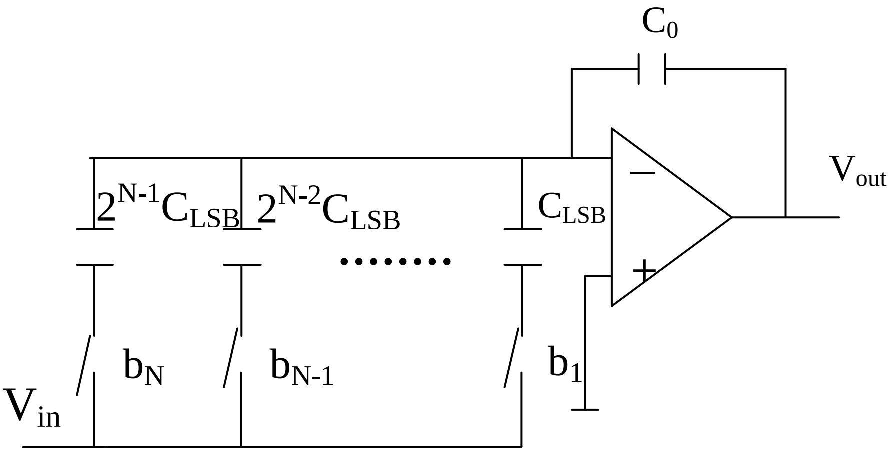


Figure 2. 1 Charge-redistribution DAC

The type of structure is insensitive to Opamp input-offset voltage, 1/f noise, and finite-amplifier gain. But, there are several drawbacks in this structure, such as the capacitors matching, the switch-on resistance, and the finite bandwidth of the amplifier. The Opamp bandwidth limitation limits charge-redistribution, switched-capacitor DAC's in medium-to-low speed application.

2.1.3 Current mode

Current mode DACs are very similar to resistor-based converters but are intended for higher-speed application. The basic idea can be shown as Fig. 2.2 [4]. The switches are controlled by the input word $X_i=(b_1, b_2 ,\ldots ,b_N,)$, where $b_i \in \{0,1\}$, b_N is the MSB, and b_1 is the LSB. Then the output current I_{out} will be given by

$$I_{out} = \sum_{i=1}^{N} b_i * 2^{i-1} * I_{LSB} \qquad (1.1)$$

where I_{LSB} is the unit LSB current.

Current-steering DACs are easy to integrate in standard digital CMOS technologies. They can also deliver nearly all power to the output and are, therefore, power efficient, due to their current characteristic. So, current-steering DACs are often used for high-speed and high-resolution converters in advanced CMOS.

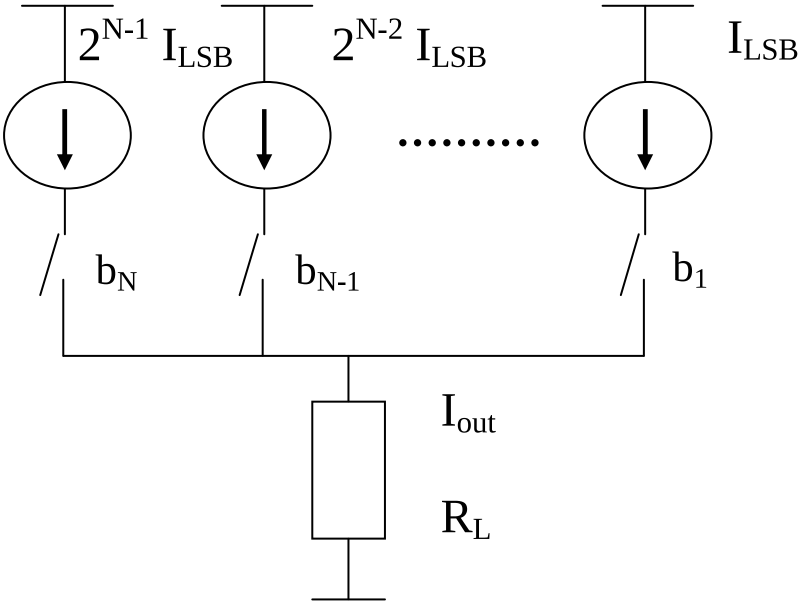


Figure 2. 2 Current-steering DAC

Current-steering DACs can be built very compact, and the area required can be quite small comparing to others architectures. The small area can reduce the gradient error effect. Because they can drive an output resistive load directly without requiring the use of extra buffer, current-steering DACs can also be very fast.

Applications in broad-band communication demand DACs higher than 10 bits linearity and sampling rates up to hundreds of Msamples/s [1][6]. These specifications push the designs to the technological limits of current digital CMOS processes. In this situation, current-steering DACs are often used.

2.2 DAC architectures

There are several possibilities for how a digital-to-analog current-steering converter can be implemented. The approaches differ in complexity, in the control of the switches, and in the weight of the current sources. Some architectures need additional circuitry, like a thermometer decoder. They also differ in static linearity and in dynamic error for the same total current source area. There are three possible architectures for the implementation of the current source array: the binary-weighted architecture, the thermometer-coded architecture, and the segmented architecture. A brief introduction of each will be given in the following.

2.2.1 Binary-weighted architecture

The basic idea of binary-weighted architecture can be shown as Fig. 2.2. In binary-weighted implementation, every switch steers a current to the output that is twice as large as the next least significant bit. The digital input code directly controls these switches. The advantages of this architecture are its simplicity and the small silicon area requirement for digital circuit because no decoding logic is needed. On the other hand, a large differential nonlinearity (DNL) error and an increased dynamic error are intrinsically linked with this architecture. Especially in the medium code, a single current source with the weight 2^{N-1} is switched on or off and N-1 current sources with the total weight 2^{N-1} -1 are switched off or on. This causes a large differential nonlinearity error. A large glitch is normally produced due to timing mismatches, which greatly affects dynamic performance. Moreover, monotonicity cannot be guaranteed by this architecture.

2.2.2 Thermometer-coded architecture

In contrast to binary-weighted architecture, in thermometer-coded architecture every source has a weight of 1 LSB and is addressed individually. The switches are not directly controlled by the digital input code. As shown in Fig. 2.3, the digital input code (B_0, B_1, …, B_{N-1}) is first converted to the thermometer code ($T_0, T_1, \cdots, T_{2^N-1}$), and then the thermometer code are used to control the switches.

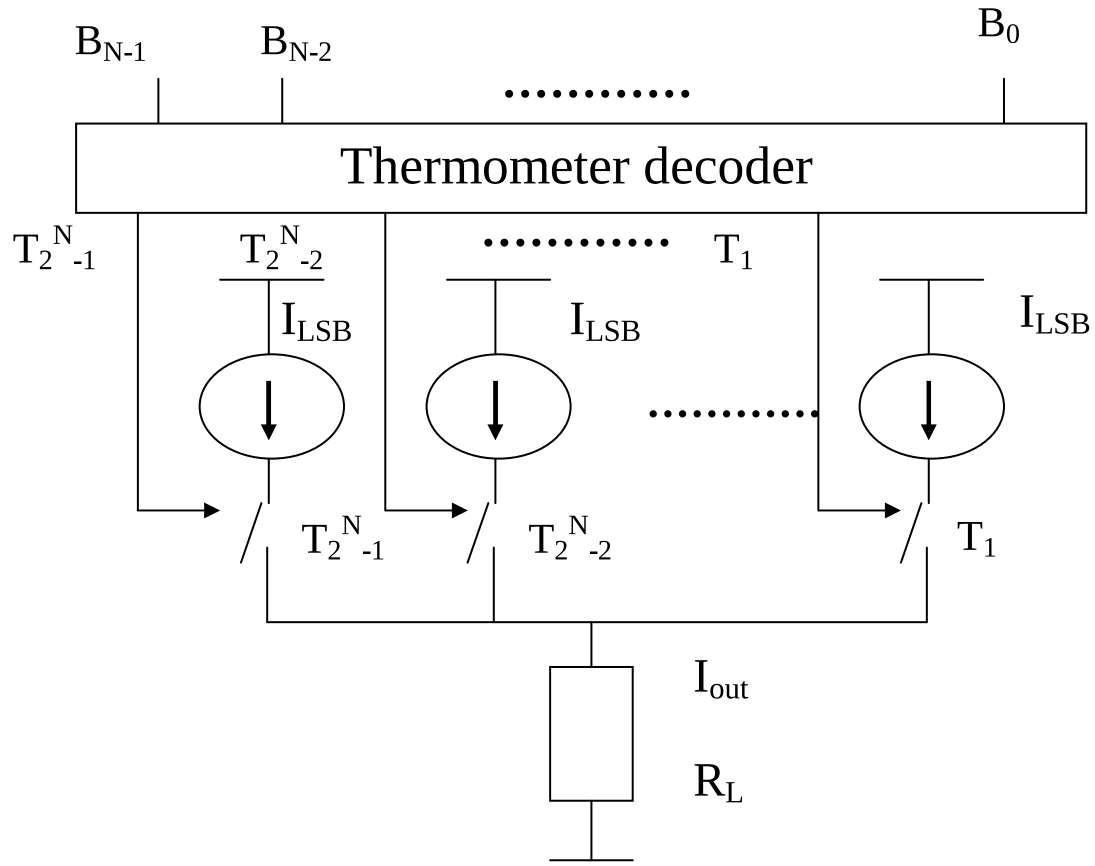


Figure 2. 3 Thermometer coded Architecture

The advantages of this architecture are its good DNL error and the minor dynamic switching errors. Since at every LSB transition only one additional current source has to be switched to one of the outputs, in thermometer-coded architecture, the DAC has a guaranteed monotonic behavior. The major disadvantage of the thermometer-coded architecture is the complexity, the area required, and the power consumption of the thermometer decoder, especially for resolutions beyond 10 bits [7]. An N-bit binary input code is mapped to a 2^N-1 bit thermometer code. For large resolutions, the thermometer decoder will become increasingly complex and large in terms of silicon area.

2.2.3 Segmented architecture

Segmented architecture can combine the advantages of both binary-weighted and thermometer-coded DACs [7]. In this case, the DAC is divided into two sub-DACs: the N_b bits LSB are implemented using a binary-weighted architecture, while the N_t bits MSB are implemented in a thermometer-coded architecture. In this architecture, a balance between good static and dynamic performance at a reasonable decoder area and complexity can be achieved.

2.3 Performance characteristics

In previous sections, we discussed the different architectures and implementation for the DACs where all DACs are assumed to be ideal. This means that the DACs are free of transistor mismatching, all unitary currents are identical and constant, there are no parasitic capacitance and resistance, and the settling time is infinite short. But in reality, DACs rarely run in ideal situations. Their performance may be affected by many factors like temperature variation, device process variation, current source internodes, finite output impedance, etc. [8][9]. All these nonlinearities will add errors to the DACs' output and, therefore, degrade DACs' performance.

The characterization of DACs' performance can be divided into static and dynamic properties [10][11]. The static properties are DACs' performance at DC and low frequencies.

These properties are determined by settled DAC output analog values and affected by static errors like settling errors and finite output resistance. Since static properties are the description of the settled values, they do not describe DACs' behavior in the transient region. They are usually too optimistic for the DACs' measurement, especially at high frequency ranges, but they set the best performance that a DAC can achieve. The static errors can be measured by differential nonliniearity error (DNL), integral nonlinearity error (INL) in time domain, and Spurious-free-dynamic range (SFDR) (at low frequencies) in frequency domain. Dynamic properties present the signal-dependent transition between two states. So, they are affected by not only the static properties but also the successive input codes. In the time domain, DACs' behavior is measured by the settling time, slewing, glitches, and time skew, etc. In the frequency domain, DACs' performance can be measured by SFDR, signal-to-noise ratio (SNR), total harmonic distortion (THD), etc.

Linearity is often an important aspect considered in the current-steering DACs design. The linearity is affected by both the static nonlinear errors and dynamic nonlinear errors [2]. In the following, the major errors are briefly reviewed.

2.3.1 Static performances

The static performances of DACs describe the behavior at DC or low frequencies. The most common static performances are quantization noise, gain error, offset error, INL, and DNL. The most important static measurements are INL and DNL[10].

2.3.1.1 Quantization noise

The quantization noise exists in ADC [4]. As for DACs, they have no quantization noise since the output signals are well defined as long as the resolution of the DAC is not lower than the input signal. But even in this case, we can still compare the DAC output with the "ideal" analog output where the resolution is assumed to be infinite. This can be shown as

Fig 2.4. Assuming the input signal is an uniformly increasing ramp, there is no overloading. In figure 2.4, the dash line is the ideal analog output, and the solid line is the actual output. The DAC output is assumed to be sampled-and-held, meaning the output is staircase shaped. We also assume the transition region is very short compared with the sampling period.

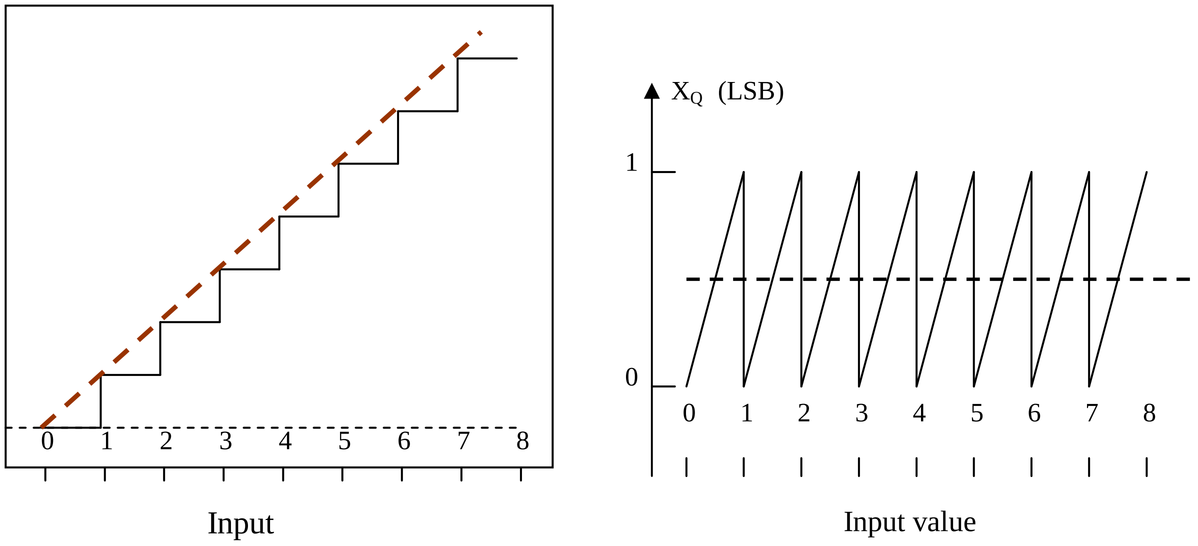


Figure 2. 4 Quantization Noise

Then the actual output can be expressed as:

$$X_0(t) = \sum_{k=0}^{2^N-1} X_i(k).V_{LSB}.S_T(t-kT) \qquad\qquad 0 \le t \le 2^N T \qquad (2.1)$$

where $S_T(t)$ is the square pulse function:

$$S_T(t) = \begin{cases} 1 & 0 \le t \le T \\ 0 & Otherwise \end{cases}$$

The ideal analog output is:

$$X_{id}(t) = V_{LSB}.\frac{t}{T} \qquad\qquad 0 \le t \le 2^N T$$

Taking the difference between these two signals will give the noise signal $X_Q(t)$:

$$X_Q(t)=X_{id}(t)-X_o(t)$$

The quantization signal X_Q is limited to $\pm V_{LSB}/2$, the offset of the quantization signal is $V_{LSB}/2$. However, the root-mean-square (rms) of the noise signal is given by:

$$X_Q^2(rms)=\frac{1}{T}\int_0^T X_Q^2(t)dt-(\frac{V_{LSB}}{2})^2=\frac{V_{LSB}{}^2}{12}$$

So,
$$X_Q(rms)=\frac{V_{LSB}}{\sqrt{12}} \qquad (2.2)$$

For reasonable high-bit resolution, quantization noise can be regarded as white noise. Then, the power spectral density (PSD) of the noise signal will be uniformly distributed over the Nyquist rate range. So, the PSD can be expressed as:

$$S_Q(f)=\frac{V_Q{}^2(rms)}{f_2/2}=\frac{V_{LSB}{}^2}{6.f_s} \qquad (2.3)$$

where f_s is the sampling frequency of the DAC.

For a given input signal waveform, a formula can be derived to give the best possible signal-to-noise ratio (SNR) for a given number bits (N) in an ideal DAC.

The sinusoid signals are the often-used candidate to characterize the DAC. So, we also use a sinusoid signal to derive the SNR. Assuming the V_{in} is a sine waveform between 0 an V_{ref}, the ac power of the signal is $\frac{V_{ref}}{2\sqrt{2}}$. So the SNR will be:

$$\begin{aligned} SNR &= 20\log(\frac{V_{ref}/2\sqrt{2}}{V_Q(rms)})=20\log(\sqrt{\frac{3}{2}}2^N) \\ &= 6.02N+1.76dB \end{aligned} \qquad (2.4)$$

The SNR increased about 6 dB for each additional bit in the DAC.

2.3.1.2 Offset and gain errors

During quantization noise analysis, it is assumed that no errors exist in the DAC, which means that the actual value is equal to the ideal value. But, this equality of values is not true in reality. The output will not be a uniform staircase if there are errors in the DACs.

As we can observe Fig. 2.5, the dashed line is the desired output, and the solid line is the actual output. The two signals will not coincide due to the errors. This error can be specified as offset and gain errors [4]. Gain errors can be divided into linear errors and nonlinear errors. The linear errors will scale the DAC analog output signal magnitude, while the nonlinear errors will introduce distortion to the output waveform.

Gain and offset errors are extracted from a sampled set of digital input code and analog output code. A line can be drawn from the input code and output code. In DACs, the offset error (E_{off}) is defined to be the output that when the DAC input code is $C_{0,}$ which should produce zero output[4]. It can be expressed in the units of LSB:

$$E_{off} = \frac{V_{out}}{V_{LSB}}\Big|_{0\cdots0} \tag{2.5}$$

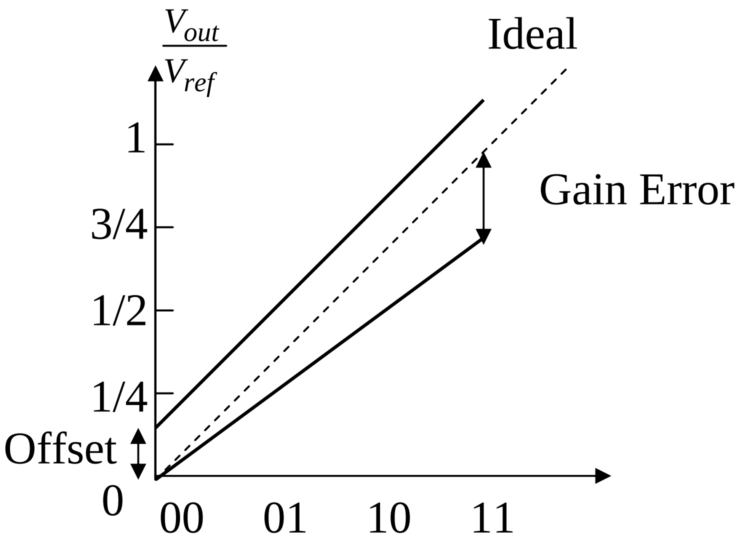


Figure 2. 5 Offset and gain error

The gain error is defined as the difference between the ideal and actual curves at the full-scale value, free of offset error [4]. It can be given as:

$$E_{gain} = \left(\frac{V_{out}}{V_{LSB}}\Big|_{1\cdots1} - \frac{V_{out}}{V_{LSB}}\Big|_{0\cdots0}\right) - (2^{N} - 1) \qquad (2.6)$$

The graphical illustration of the gain and offset errors is shown as Fig.2.5.

2.3.1.3 Integral nonlinearity error

INL error is defined as the deviation of the output characteristic of the DAC from a straight line [4]. A more conservative measure of the INL is to use to endpoints of the converter's transfer response to define the straight line. This is equivalent to gain and offset compensation. An alternative definition is to find the best-bit straight line such that the mean squared error is minimized. INL values are defined for each digital input code, and, thus, the INL can be plotted as a function of the input code. If the general curve of the endpoints line or best line is:

$$y = mx + b$$

where the m is the gain of the DAC and b is the offset. Then the INL can be given in LSB as:

$$INL(i) = \frac{(V(i) - (mX(i) + b))}{V_{LSB}} \qquad (2.7)$$

2.3.1.4 Differential nonlinearity (DNL) error

In an ideal converter, each analog step size is equal to 1 LSB. DNL is defined as the variation in analog step sizes normalized to 1 LSB, and it can be presented in the units of LSB as [4]:

$$DNL_{(i)} = \frac{(V(i+1) - V(i)) - V_{LSB}}{V_{LSB}} \qquad (2.8)$$

Typically once gain and offset errors have been removed an ideal converter has a differential nonlinearity of 1 for all digital input codes; therefore, a converter with maximum DNL of 0.5 LSB has step sizes varying from 0.5 LSB to 1.5 LSB. Once again, as in the case of INL, DNL values are defined for each digital code. Sometimes the term "DNL" is used for the maximum magnitude of the DNL values.

The INL and the DNL error can be used to evaluate the quality of a DAC. Another possibility is to specify the maximum INL and DNL errors and to design the DAC according to maximum errors.

2.3.1.5 Output impedance

The output impedance and the parasitic impedance of interconnections and switches in the converter will strongly determine the performance. Any nonideal current source has a finite output resistance that can be modeled as Fig. 2.6. When the different current sources are switched to the output, the total output impedance is changed. When only static values are considered, assuming the voltage at output node is $V_{out,}$ then the following equations can be obtained:

$$\begin{cases} I+(V_{dd}-V_{out})g_o=I_{out}\,R_L \\ V_{out}=I_{out}\,R_L \end{cases} \tag{2.9}$$

So, the output current through the load is:

$$I_{out}=\frac{I}{1+R_L g_o}+\frac{g_o.V_{dd}}{1+R_L\,g_o} \tag{2.10}$$

where I is the current source output current, I_{out} is the nominal output current from the DAC, $g_o=1/R_o$ is the output conductance, R_L is the signal-independent load resistance, and V_{dd} is the supply voltage. When $g_o=0$, the output current I_{out} = I. It means all current flows to the output load. For an input signal sinωt, the number of switches S that conduct the current at time t is:

$$S(t)=N(\frac{1+\sin(\omega t)}{2})$$

The total output impedance of the DAC is then determined by the load resistor R_L in parallel with S(t) parallel switched on impedance R_o. So the total impedance is:

$$g_{out} = g_L + g_o N[\frac{\sin(wt)+1}{2}] \qquad (2.11)$$

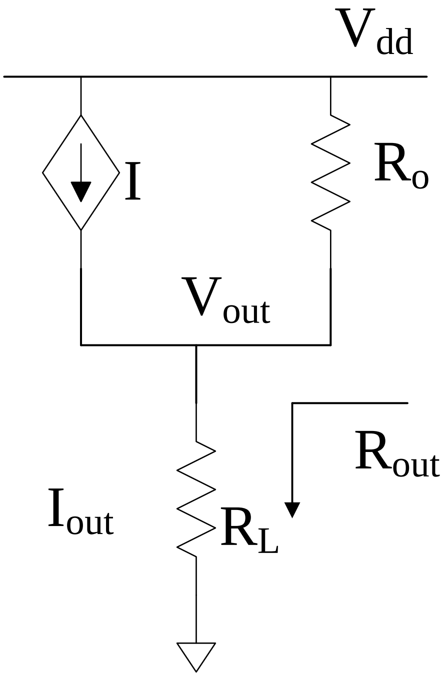


Figure 2. 6 Nonideal output impedance current source

where g_o=1/R_o, g_L=1/R_L. If current for one current source is I, then the total output voltage V_{out} is:

$$V_{out} = S(t)*I/g_{out} = \frac{N(\sin(\omega t)+1)I}{2g_L + g_o N(\sin(\omega t)+1)}$$

If the second harmonic is the biggest spur, then the ratio (R_h) of the coefficients of the second harmonic to the fundamental signal gives the SFDR. Exploit the expression of V_{out}, and the R_h is:

$$R_h \approx \frac{Ng_o}{4g_L} \qquad (2.12)$$

where $R_h = e^{-SFDR/20}$

So the required output impedance to achieve a specified SFDR is:

$$SFDR = -20\log(R_h) = 20\log(\frac{NR_o}{4R_L}) = 20\log(\frac{R_o}{R_L}) - 6.02(N-2) \qquad (2.13)$$

From the equation, it can be seen that when R_L doubles, the SFDR will decrease 6 dB. Increasing the resolution of the DAC for a constant R_o/R_L, the SFDR will degrade.

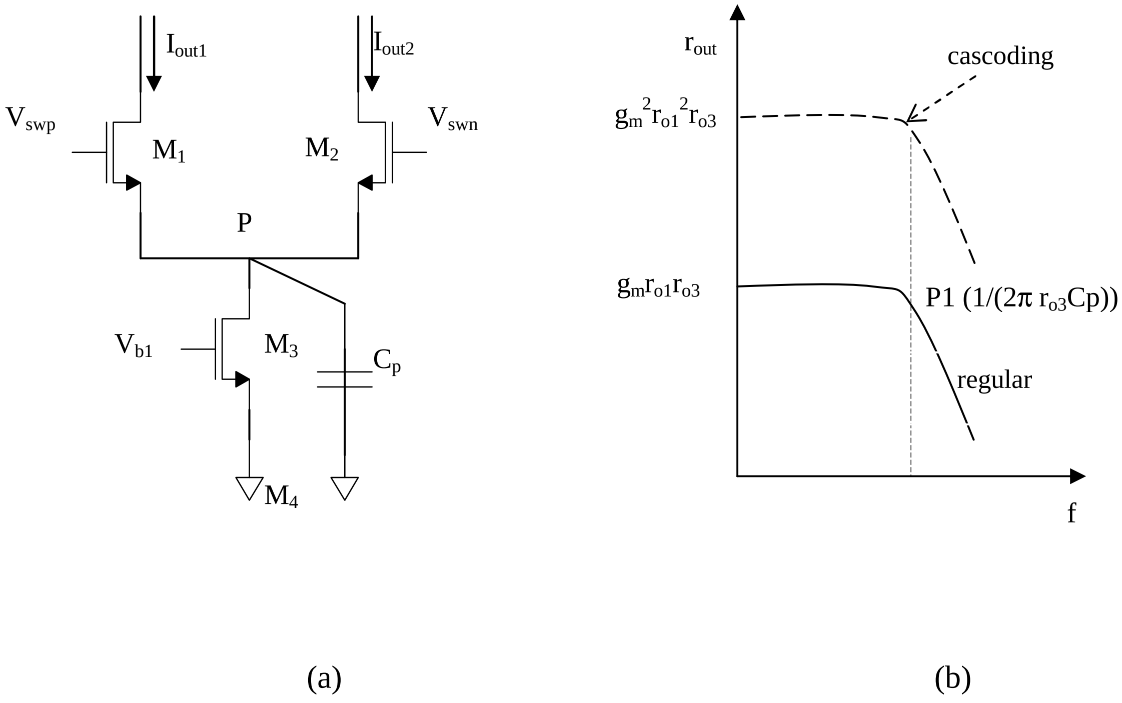


Figure 2. 7 Current source and the output impedance

(a) regular current source (b) output impedance for regular current source and cascade current source

When the operation frequency increases, then the parasitic capacitance needs to be considered [8][9][12]. In DAC design, all the current cells are usually put together and separate from the switches and logic gates. One advantage of this design is that the current sources array will be compact; therefore, it will introduce less gradient error. Another advantage is when the switches are put together and away from the current array, switching noise coupling to the current sources is much reduced. But, this also introduces another issue: larger parasitic capacitance at the current source output node due to the long distance connection to switches and the large current sources size used to reduce random errors. A current cell with switches is shown as Fig. 2.7(a).

Assume the switches M_1 and M_2 are identical and their intrinsic output impedance $r_{o1}=r_{o2}=r_o$, the effective output impedance of the tail current source is r_{o3}, and the parasitic capacitance at node P is C_p. C_p is capacitance look through node P, which is the sum of the capacitance from M1, M2, M3, and the bus connection. The control voltage V_{swp} and V_{swn} turn high or low to switch the current I to either M_1 or M_2. When one switch is ON, for example M1, the output impedance look from output is around:

$$\begin{aligned} r_{out} &\approx g_m * r_{o1} * (r_{o3} // \frac{1}{SC_p}) \\ &= g_m * r_{o1} * \frac{r_{o3}}{1 + r_{o3} * SC_p} \end{aligned} \qquad (2.14)$$

where g_m is the transconductance of transistor M_1 or M_2.

The solid line in Fig. 2.7(b) is the plot of the impedance looking into the drain of switch M_1 from the output node. The equation (2.14) indicates a pole at $1/(r_{o3}C_p)$ exists in the output impedance. This pole will lower the output impedance value when the operation frequency is higher than the pole. From the equation (2.13), a smaller r_{out} will result in SFDR degradation. Considering the parasitic capacitance at output node, there is another pole in the effective output from output node. But since this pole is far away from the original point, we neglect it here.

From equation (2.14), it can be seen that the pole is determined by the interconnection capacitance C_p and current source output impedance r_{o3}. Some layout skills can be applied to optimize C_p. However, this merely shifts the pole to the right side and has no effect on the magnitude of the effective output resistance. Some design-level solutions can be adopted to increase the output impedance value. Usually, this can be done by cascoding another transistor on top of either the switching transistor or the current source transistor. If an identical transistor is cacoded on the top of switching transistor, and only the capacitor C_p is of consideration, the effective output impedance looking from output node into the drain of cascoding transistor is:

$$\begin{aligned} r_{out} &\approx g_m{}^2 * r_{o1}{}^2 * (r_{o3} // \frac{1}{SC_p}) \\ &= g_m{}^2 * r_{o1}{}^2 * \frac{r_{o3}}{1 + r_{o3} * SC_p} \end{aligned} \qquad (2.15)$$

It is about $g_m r_{o1}$ larger than the original one. This is plotted as the slash line in Fig. 2.7(b). From the plot, it can be clearly noticed that the output impedance is much improved at both low frequency and high frequency. Another advantage of this kind of cascode stage is that it can alleviate the feed-through from the control signal. However, it requires two extra transistors for each current source, which will increase the silicon area. For cascode structure, it consumes one overdrive voltage and, therefore, limits the operation output voltage swing.

Another cascode scheme is to put the cascading transistor on the top of the current source. The output impedance can be improved $g_{cs}r_{o3}$ for this kind of structure where g_{cs} is the transconductance of the current source. With the increase of the current source output impedance, the distortion at the DAC output can be reduced due to significant alleviation of voltage variation at node P. As discussion in switching transistor cascode, the current source cascode also needs an extra transistor. It also has voltage-headroom consumption, which will limit the usage of this approach, especially at low power supply voltage design.

2.3.1.6 Current sources mismatch

One of the important static error sources in DACs is current source mismatch. The mismatch errors of the array can be distinguished into random errors and gradient errors.

a. Random errors

Random mismatches are determined by the matching properties of the technology used. The random variations of devices are assumed to be uncorrelated and follow the Gaussian distribution [2].

According to Pelgrom's model[13], for two transistors at the same die and closed to each other, the mismatch-caused variation can be expressed as:

$$\sigma(\Delta V_t)=\frac{A_{VT}}{\sqrt{W.L}}$$

$$\sigma(\Delta\beta/\beta)=\frac{A_\beta}{\sqrt{W.L}} \qquad (2.16)$$

where Vt is the threshold voltage, β is the current factor, W and L are the width and length of the transistor gate.

So, the variation for a single transistor can be expressed as [13]:

$$\sigma^2(\Delta I/I)=\sigma^2(\Delta\beta/\beta)+\frac{4\sigma^2(\Delta_{VT})}{(V_{gs}-V_T)^2}=\frac{1}{W.L}(A_\beta{}^2+\frac{4A_{VT}{}^2}{(V_{gs}-V_T)^2}) \qquad (2.17)$$

where A_β and A_{VT} are the technology constants and the overdrive voltage is V_{GS}-V_T. From the expression, it can be seen that the current variation is inversely proportional to the gate area of the transistor. So, in order to reduce random errors, either the overdrive voltage or the gate area needs to be increased. Since W.L are common in both terms, increasing the active area of each unit current source in the DAC array is the most effective method for a given process technology and for architectures.

b. Gradient errors

Gradient errors can be divided into two categories: linear and quadratic. Linear gradient errors may be caused by the spread of doping and oxide thickness over the wafer or voltage drop along the power line. On the other hand, quadratic gradient errors may be caused by temperature and die stress [2]. The overall gradient error distribution is the superimposition of these error components. From Pelgram's model (2.17), it can be seen that for each extra bit of resolution, in order to obtain the same INL and yield, the active area of the unary array has to be increased by a factor of four. Therefore, for large-bit N, the array area has to be dramatically large to suppress the random errors. However, this large area will cause significant gradient errors due to the long distance between current sources. Matrix configurations are often used, with a square matrix especially preferred. But, even with compact layout, the distances between current sources are still large.

To compensate for symmetrical and graded errors, special switching schemes are implied to implement the current sources [14]. In the design, current source of the unary array is divided into four current sources. In each quadrant, a current source is placed based on a centroid scheme. Dummy rows and columns can be added to avoid edge effects.

The linear gradient error can be expressed as [2]:

$$\varepsilon_\ell(x,y)=g_\ell\cos\theta.x+g_\ell\sin\theta.y \tag{2.18}$$

where θ is the angle of the linear gradient and $\theta \in [0,360]$ while g_1 represents the slope of the gradient, and (x,y) is the current sources location.

The quadratic gradient error can be expressed as:

$$\varepsilon_q(x,y)=g_q(x^2+y^2)-a_0 \tag{2.19}$$

where g_q and a_0 are technological parameters, and (x,y) is the current sources location. In this model, the DAC is assumed to be located at the center of the die, and the quadratic gradient in both the x and y directions are assumed to be independent and equal.

In practice, the errors in a die are the superposition of both the linear gradient error and the quadratic gradient error. So, the joint errors can be expressed as:

$$\varepsilon(x,y) = \varepsilon_\ell(x,y) + \varepsilon_q(x,y) \tag{2.20}$$

2.3.2 Dynamic performance

There are several parameters that affect the dynamic performance of DACs [10][11]:

- Settling or conversion time
- Slew time
- Glitch energy
- Crosstalk
- Timing skew

DACs' output looks like a staircase with unsettled transitions. These parameters are best observed with high bandwidth instruments. High-speed DACs are usually observed with a high-bandwidth oscilloscope on transition from minus full scale to plus full scale and rarely measured in production. Rise time is usually measured at 10%~90% or 20%~80%. Settling time is typically described to within 0.5 LSB. The glitch energy is an integral voltage-time product of the area outside of a 0.5 LSB error band in units of psV.

The influence of the dynamic nonlinearities on the distortion performance of DACs can be described by using measures in both the time and the frequency domain. The most important time and frequency domain specification will be discussed in the following.

2.3.2.1 Settling time

Settling time of a DAC is defined as the time required for the output to experience a full scale transition and to be settled within a specified error band around its final value [4]. Settling time is affected by the slew rate of an output current source and by the amount of output ringing and signal overshoot. With increased input frequency, the clock period is shorter, which may cause insufficient settling time. Therefore, the final output may reach an inaccurate value and introduce nonlinearities.

2.3.2.2 Glitch energy

Glitch energy is defined as the area under two consecutive output codes [11]. This error is mainly caused by timing errors within the DAC and result in a deterioration of the dynamic performance.

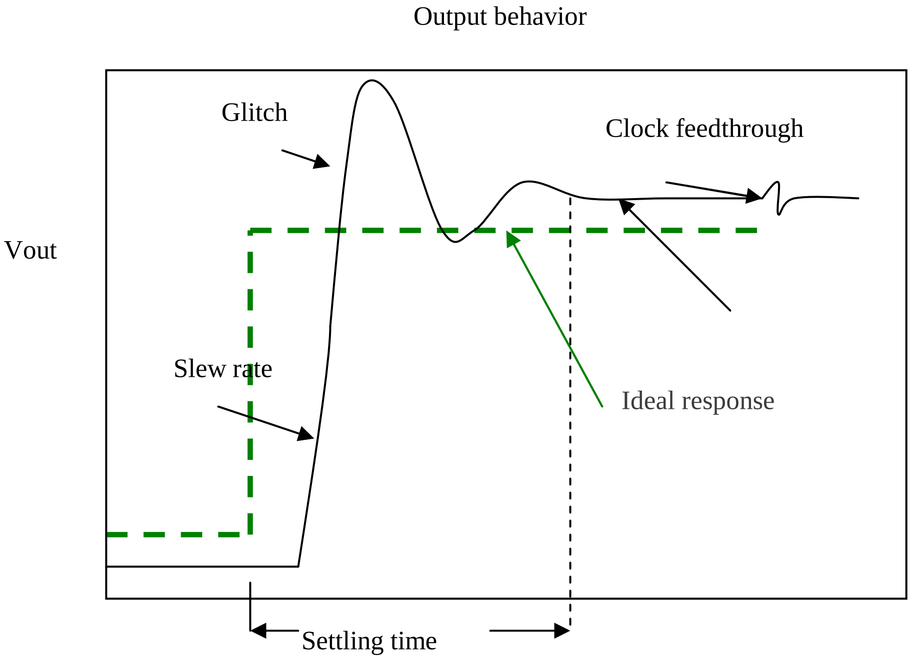


Figure 2. 8 The DAC dynamic specification

Glitches occur when switching time instants of different bits in a DAC are unmatched. This can depend on matching errors in switches and driver circuits, time skew between switching signals, voltage-dependent CMOS switches, etc. For a short period of time, a false code could appear at the output.

Fig.2.8 is an illustration of typical glitch behavior at the DAC output. The dotted line indicates the ideal transfer and the solid line the actual behavior. The glitch is modeled as a pulse as dash in the figure. This pulse has an amplitude of A_g and a time-duration T_g. The glitch energy during the time interval T_g is given by:

$$E_g = A_g{}^2 .T_g \tag{2.21}$$

2.3.2.3 Slew rate

Another phenomenon that can cause nonlinear distortion is slew rate, which is due to an output signal that is too large or the changing is too fast. Slew rate is defined as the maximal rate at which the output of the DAC can change with the varying input [15]. The origin of slew rate is current-source output limitation. If output of the current source is I and the capacitance at the output node is C_o, the maximum rate of charging or discharging the capacitor is I/C_o. When there is a sharp changing at the input and if the output changing is larger than I/C_o, the output will change at rate I/C_o. That means the output fails to follow the input changing. Therefore, distortion is caused to the system.

2.3.2.4 Clock feedthrough

Clock feedthrough is also an important parameter for DACs' dynamic performance [28]. The cause of feedthrough is parasitic capacitive coupling. For a switch implemented with MOS transistor, there is a parasitic capacitance C_{gd} between the digital switching signal and the analog output node. Due to the Miller effect, this capacitance is quite large, and, therefore, the clock, used as switching signal, will affect the analog output, which is very sensitive to noise. This effect can occur at both rising and falling edges of the switching signal [4]. Clock feedthrough is a broad term for any feedthrough from digital data lines or any clock to the DAC output. The effect of the switching of the clock can be directly seen at the output of the DAC. Actually, the clock feedthrough does not introduce any other noise or distortion in the Nyquist base band region because of its code independence. The noise introduced by the clock feedthrough can be removed by just putting a low-pass filter at the output of the DAC. Since the clock feedthrough is related to parasitic capacitance, another way to minimize the clock feedthrough is to reduce the transistor size. However, this will increase the settling time, resulting in a degradation of performance.

2.3.2.5 Timing skew

One of the biggest causes of dynamic distortion in current-steering DACs is error in

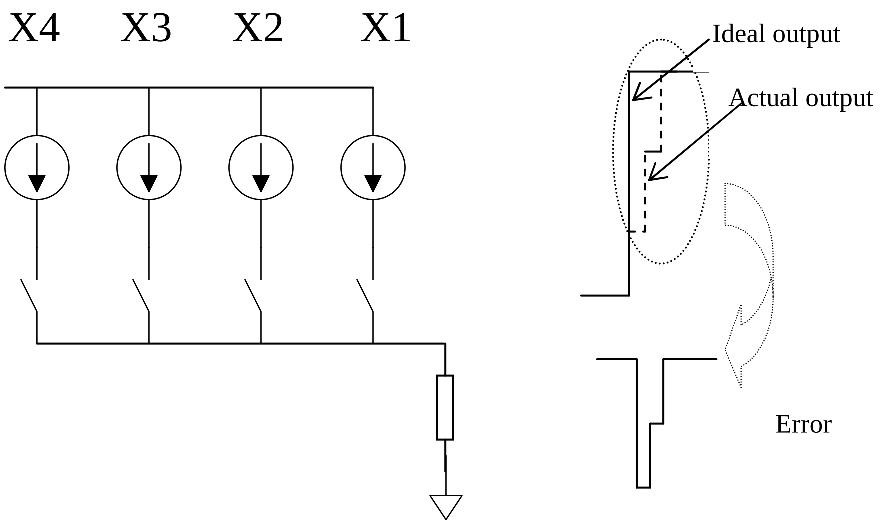


Figure 2. 9 Time skew of DACs

transition instant [9]. When input codes do not switch exactly at the sampling point then a glitch will occur. This can be illustrated as Fig. 2.9. Suppose there are four current sources, X_1, X_2, X_3, X_4, that need to be switched. At first, the X1 switches the unit current, then X2 switches the unit current several ps later, then X3, and X4. Hence, an error waveform will be created. This error signal includes both pulse amplitude and width modulation

2.4 Approaches of nonlinearities compensation

Many approaches were presented in the literature for static nonlinearities and dynamic nonlinearities compensation. Among these approaches, some are from the point of layout, others are from the point of circuit design, and remaining approaches use calibration. The following is a brief introduction of these approaches.

2.4.1 Switching schemes

Different switching schemes are used to minimize random and gradient errors. For thermometer array architecture, a thermometer decoder is used to transfer binary digital word to a decimal value. In order to achieve a high speed, it is typical to implement this strategy with a row-column decoder.

For this commonly used decoder scheme, the spatial gradient are averaged into x and y in two directions. The whole decoder optimization is achieved by optimizing the row and column selection separately. So, the two dimensions can be reduced into one dimension. Three different switching schemes using this decoder have been presented in the literature: symmetrical switching scheme [17], hierarchical symmetrical switching scheme and Q^2 random walk [14].

For the symmetrical switching scheme, graded errors are cancelled at every two increments of the digital inputs, but symmetrical errors will accumulate with increased input codes.

To solve this problem, the hierarchical symmetrical switching scheme is proposed. It is similar to the symmetrical scheme. The difference is that for hierarchical symmetrical switching scheme, if the sequence is divided into four quarters, there are two kinds of switching sequences: type A and type B. For type A, current sources in quarter two and one are turned on first, then three and four. For the type B, current sources in quarter two and three are turned on first, then one and four. Implementing the switching scheme by interweaving type A and type B, the symmetric linear gradient errors are cancelled for every two current source turned on, and the graded errors accumulate then cancel for every four current source turned on. Therefore, it is best to implement the decoder with the hierarchical, symmetrical switching scheme type A since it does not accumulate symmetrical errors and has a small INL error.

For one dimensional gradient error compensation, the row-column switching schemes are good. But, they are inherently insufficient for two-dimensional gradient error compensation. In this case, a two-step hierarchical switching scheme called "Q^2 random walk" switching scheme [14] was presented to compensate for gradient errors. In this approach, for a given N bits DAC, the current matrix is divided into $2^{N/2}$ region, and each region contains $2^{N/2}$ cells. Two steps are used to compensate for errors. First, an optimal switching sequence is used to choose the region to compensate for quadratic errors. Then, an optimal switching sequence is used to choose the cells in each region to compensate for linear errors. A quite good result can be achieved by using this approach at the penalty of complex routing.

2.4.2 Circuit techniques

The switching sequences can reduce random and gradient errors to a certain degree, but for other nonlinearities, like time skew and glitches, it cannot handle them efficiently. So, some other approaches must to be applied to deal with these errors. Some circuit level techniques will be reviewed in the following section.

2.4.2.1 Synchronization block

For the differential control signal, when switches are turned on/off, if the crossing point of the switch control signals are situated exactly at (V_{DD}+V_{SS})/2, then a time interval exists where the overlap voltage at the drain of the current source is above the average value. This will generate a glitch and cause distortion in the output of the DAC and degrade the dynamic performance. A synchronization block can be used immediately in front of the switch transistors to solve this problem.

2.4.2.2 Cascode stage

As mentioned before, output impedance will affect both the static and dynamic performance of DACs. The relationship between the output impedance R_0 and the specified INL is given as [16]:

$$INL = \frac{I_{unit} R_L{}^2 N^2}{4R_0} \tag{2.22}$$

where R_L is the load resistor, I_{unit} is the LSB current, and N is the total number of the unit current sources. From the expression, it can be seen that for a given DAC, in order to achieve a specified INL, the output impedance R_0 should be larger than a required value.

Assuming the input is a sinusoid input and the major spur is from the 2nd harmonic, then the relationship between the output impedance R_0 and the SFDR is:

$$R_0 = \frac{N \cdot R_L \cdot SFDR}{4}$$

In order to achieve a high SFDR, the output impedance has to be sufficiently large. A cascode transistor can be added to the top of a switching transistor or to the top of the current source to increase the output impedance.

But with the transistor feature scaled down, the supply voltage becomes low. It is difficult to implement the circuit with a cascode structure. Moreover, the stacking of the cascode stage reduces the effective voltage headroom of all the transistors in the current cell. It also reduces the effective gate-source voltages of the current sources. The reduction severely deteriorates the matching and noise immunity of the current sources.

2.5 Conventional calibration approaches

From the previous discussion, we know that static and dynamic errors inevitably exist in DACs and these errors degrade DACs' performance. In medium-accuracy DAC designs, nominal element matching without trimming or calibration may satisfy the requirement. But, when higher accuracy is needed, a number of techniques can be applied to correct transistors' mismatches [14]. As mentioned before, since the errors are inverse proportional to transistor area, the most effective method to reduce the random mismatch is to increase the gate area of the transistors. The current sources area has to be increased four times to obtain an additional bit resolution where complex routing is needed [13]. The larger area will result in significant interconnection parasitic capacitance, which severely limits the conversion rate and high-frequency performance and seriously degrades SFDR at high frequency.

A good way to solve this problem is through proper calibration. During calibration, a small amount of extra circuit is used to compensate for nonlinearities in the main DAC. In this way, high linearity can be achieved at the penalty of small area increases. When the total area becomes smaller for a given specification, the gradient errors become small. Therefore, the requirement for switching scheme and layout can be relaxed. The reduction of the complex will result in smaller parasitic of the junction and interconnection. The smaller parasitic capacitance will introduce small nonlinearities and short settling time.

Some conventional calibration approaches have been presented in the literature [14] [17]-[31]. All analog calibration techniques like dynamic element matching or current copying have been implemented to achieve about 16-bit matching for bipolar and CMOS current source elements [31].

Groeneveld [27] proposed a widely-used background calibration scheme. In the approach, every individual current source in the MSB array and the total current of the LSB array are calibrated to equal to reference current. A dummy current source is included in the LSB array to make the total current output is equal to MSB unit current.

This calibration can be used to overcome the static errors in the current sources. However, it may not handle the errors during operation, like clock feedthrough. Another issue that should be paid attention is the spurs during the switching on and off of current sources during calibration.

The previous calibration is the analog signal calibration; another choice is to do digital calibration. In the digital approach, errors are digitized using a slow but accurate analog-to-digital converter (ADC) and stored in a register or RAMs. During conversion, these error messages are read out to either adjust the digital inputs or drive a calibration DAC to correct the analog output. In this approach, there is no need to refresh the calibration often since the error codes are stored in static RAMs. Many methods can be used to implement digital calibration. As a sample, Yonghua’s approach [2] is shown as Fig.2.10. Since most of the errors come from the MSB part, the calibration is only applied to the MSB array. The principle of the calibration is to first set all the MSBs of the DAC to “0” and LSBs to “1”.

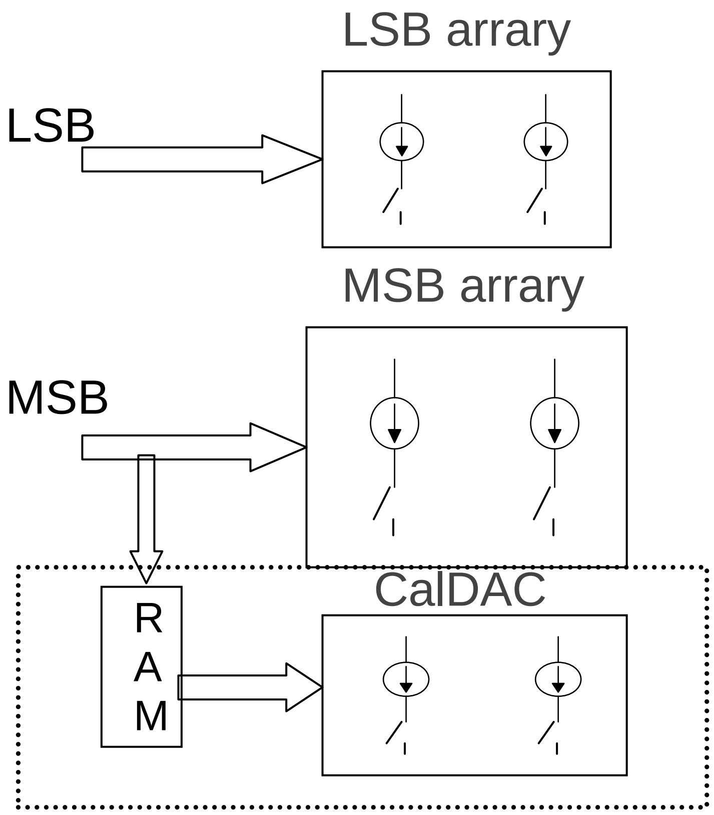


Figure 2. 10 Digital calibration

The overall LSB array outputs, including the dummy current source, are switched to the output and measured by slow but highly accurate ADC. and the result is saved and denoted as D_{LSB}. The following is the MSB array calibration while the LSB array is all set to “0.” The MSB inputs are increased by 1 in each calibration cycle. Then, the output increases by D_{LSB}. Deviation exists due to the existence of errors in the DAC. The deviation is regarded as error information and denoted as e(k) for word k and given by:

$$e(k) = D(k) - k.D_{LSB} \qquad (1 \le k \le 2^{NM} - 1)$$

where NM is the bits number in MSB array.

The coded error is stored in the RAM as word k. The error of each MSB code can be measured and stored in RAM in the same way.

In the conversion mode, the digital inputs drive the main DAC array; meanwhile, the MSB inputs address the according word of RAM and read out the error code. The error code then drives the calibration of the DAC (CALDAC). The CALDAC will generate a correction current and is summed to the main DAC output to provide the overall output current.

Because the calibration is applied to the settled value, and the value is mainly determined by the static current mismatch, this approach can significantly improve the SFDR at low frequency. For high frequency input signals, the improvement of SFDR calibration decreases. This is because the dynamic nonlinearities dominate the static errors, and, therefore, the benefit of calibration becomes less significant.

CHAPTER 3. NEW CALIBRATION SCHEME

Many calibration approaches [2], [14], [24] have been reported in the literature for improving DACs' performance, focusing mainly on reducing the static nonlinearity. In higher frequency ranges, however, the final output may not fully settle due to insufficient settling time, which will deteriorate the DAC's dynamic performance. Little has been presented to lower the dynamic nonlinearities, such as [25]. Some modest improvements in high frequency SFDR have been reported with return-to-zero structures (RTZ) at the expense of sacrificing half of the signal power.

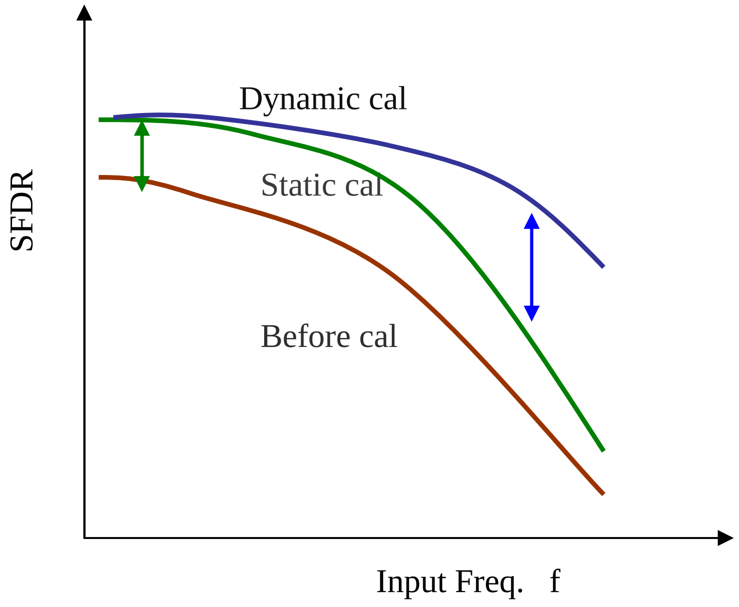


Figure 3. 1 Calibration goal

The goal of the current steering DAC calibration is to improve the DAC's dynamic performance at high frequency without concern for what causes the error performance. A novel dynamic DAC nonlinearity calibration approach is proposed that can improve high frequency SFDR without attenuation of the output signal power by compensating for dynamic errors at the output with extra pulses. The basic idea of this approach and the technique to obtain the error pattern is introduced in this chapter.

3.1 Time domain analysis

In DACs, both static and dynamic nonlinearities exist during transition The influence of nonlinearities on the performance distortion of DACs can be described in both time and frequency domain.

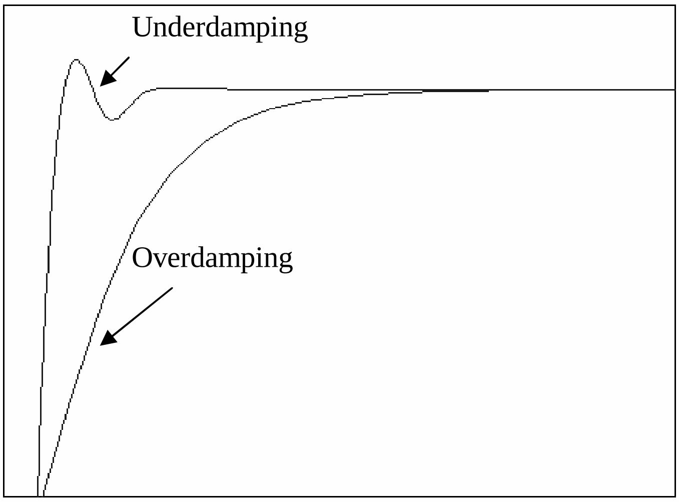
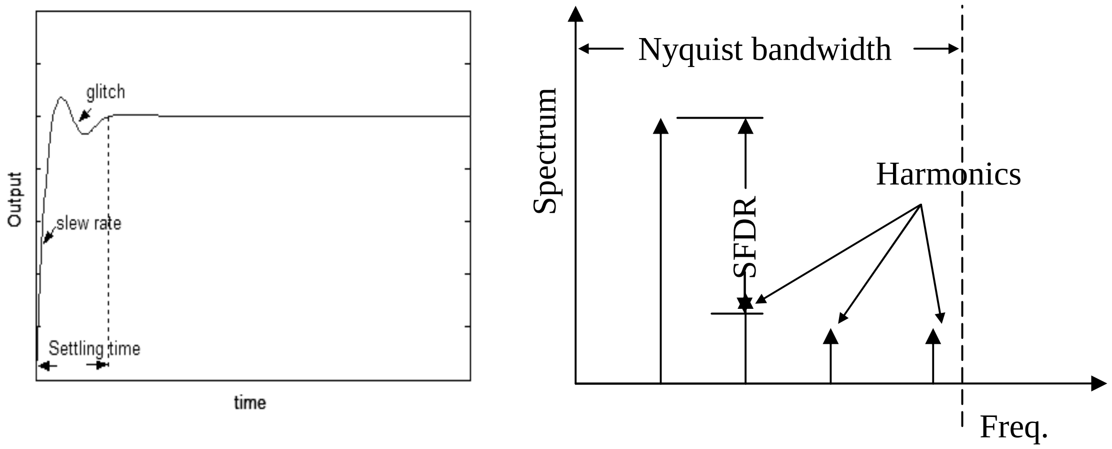


Figure 3. 2 Different output waveforms of step response

The output of a DAC with linear transfer characteristics, normalized by the size of the transition, is depicted in Fig. 3.2 for under-damped and over-damped settling. Although these outputs differ considerably from ideal steps at the leading edge of the transitions, if the magnitude changing is linear related to the input signal and even though their shapes are different, the differences do not create any significant harmonics in the output waveform.



(a) (b)

Figure 3. 3 DAC output nonlinearities

(a) time domain (b) frequency domain

In a real DAC, the settling is not perfectly linear and the value to which the DAC settles may be incorrect. These nonidealities contribute to both static and dynamic nonlinearity, thus causing distortion in the output. Static nonlinearities are dominantly attributable to nonlinear settling artifacts in the output, such as insufficient settling time.

The static errors are the major nonlinear cause at low frequency [9], e.g. the deviation of the final settled value. Other causes like timing skew, slewing, and glitches shown as Fig 3.3(a) can also bring both static and dynamic nonlinearities to DAC output. Fig.3.3(b) is the distortions expressed at frequency domain. The nonlinearities introduce harmonics, therefore degrading the SFDR.

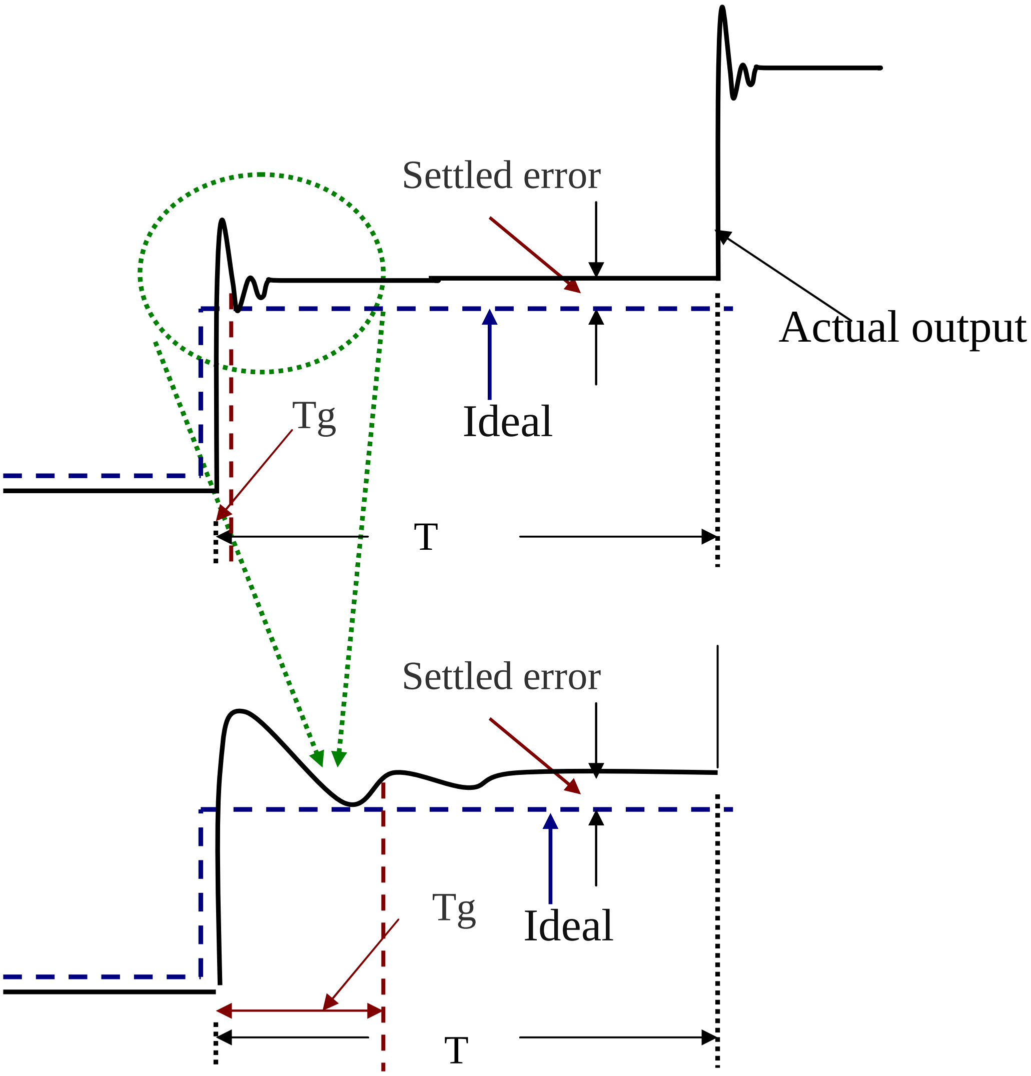


Figure 3. 4 Dynamic error models

Dynamic properties are given by the transition between two consecutive states. The dynamic error model can be shown as Fig. 3.4. Where the dash line is the ideal DAC output, the solid line is the actual DAC output waveform. In reality, the DAC suffers from both static and dynamic nonlinearities. At high frequency, the dynamic nonlinearities are dominant. The dynamic error can be divided into dynamic settled errors and dynamic glitch errors. Both are frequency dependent, and most of these dynamic errors occur at the start of the transition period. As the input signal frequency or the clock update rate increase, the dynamic nonlinearities components become larger. As a result, the linearity degrades and the magnitude of SFDR decreases. In order to improve the SFDR of DACs, the effects of the dynamic nonlinearities must be reduced.

3.2 DAC dynamic calibration scheme

Analog waveform x(t) is first sampled and quantized into a sequence X(n) before it works as an input signal to the DAC. However, the output waveform of the DAC X(t) is not linearly related to the input sequence X(n) due to the existence of the nonlinearity. The DAC produces an output consisting of a desired output waveform and an error waveform.

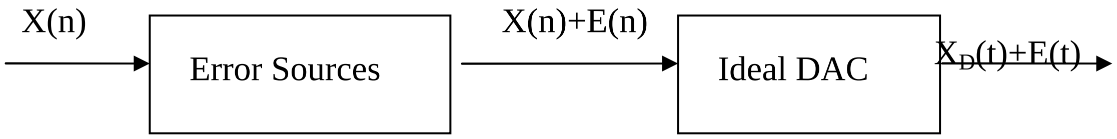


Figure 3. 5 DAC model

So the realistic DAC can be modeled as an ideal DAC with error sources as in Fig. 3.6. And, the output waveform can be expressed as:

$$X_{out}(t) = X_D(t) + E(t) \tag{3.1}$$

where $X_{out}(t)$ is the actual output waveform, $X_D(t)$ is the desired output waveform corresponding to X(n), and E(t) is the error waveform.

The purpose of DAC calibration is to minimize or eliminate the E(t) so that a nearly ideal output can be achieved. The basic idea is that, taking current-steering DAC for example, for any arbitrary input waveform flowing to the main DAC, a small amount of current is generated by a calibration DAC and added to the main DAC output current, so as to get a distortion-free output.

Fig. 3.7 is a brief calibration model. Where an additional calibration DAC (CALDAC) is used to generate a waveform and add to the main DAC output to compensate the nonlinearities error E(t). Actually, for each sample sequence X(n), there is a corresponding error E(n). The E(t) may contain many other frequency components of energy, but those do not appear in the input sequence X(n) due to the distortions in the circuits. The major work of this thesis is to minimize these distortions, especially at high frequency.

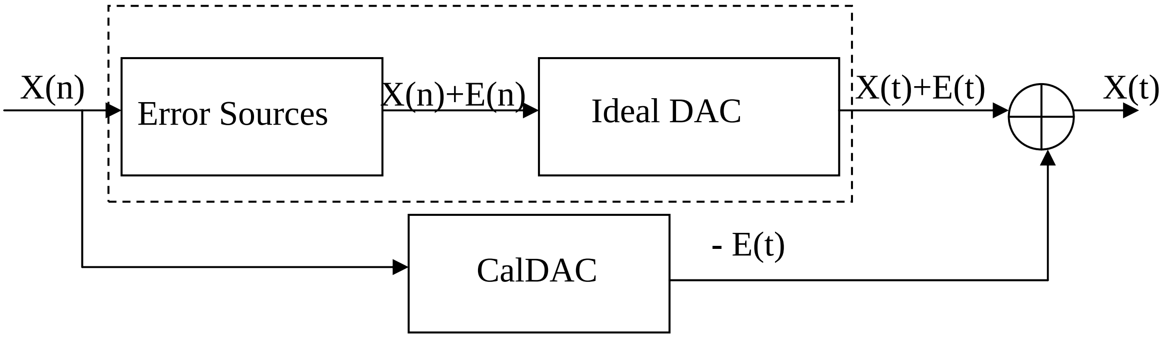
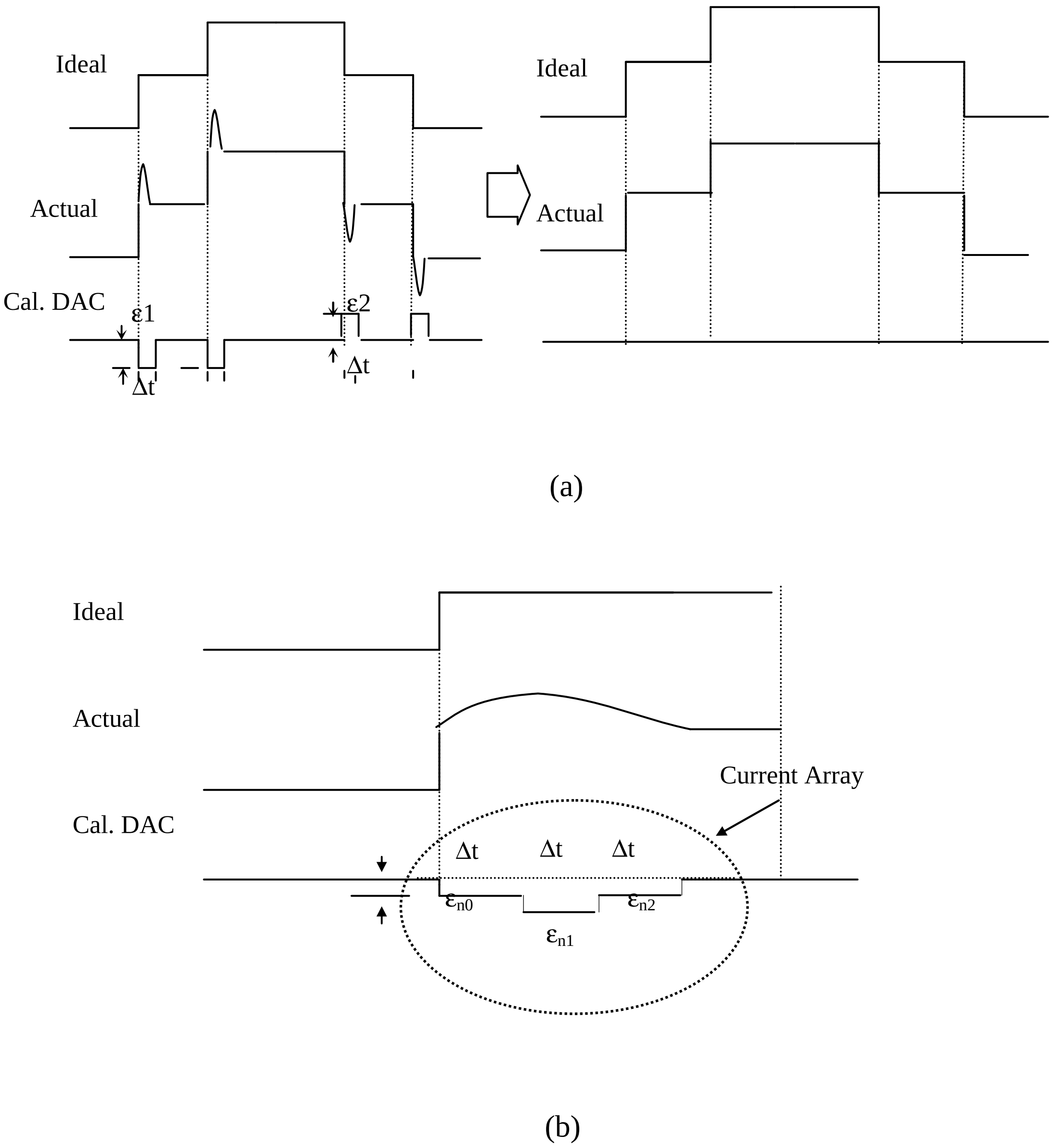


Figure 3. 6 DAC calibration model

3.3 Calibration conception

Cong [2] improved low- and high-frequency SFDR with very low power and small area. The approach is good at low frequency but the SFDR still degrades quickly with increasing frequency. In addition, Bugeja’s structure [25] improved dynamic linearity at high frequency by using an “attenuate and track” approach. But, this improvement is at the price of a factor of 2 in the signal power. The main reason of the improvement of dynamic linearity in

Bugeja’s approach is the suppression of the prior code dependence. So, measures can be adopted to improve the dynamic linearity without loss of signal power.



(a)

(b)

Figure 3. 7 MSB calibration conceptual illustration

Since dynamic errors come from dynamic settled errors and dynamic glitch errors, a pulses array can be applied to compensate for these errors. For the dynamic settled error, the error is the deviation between the desired settled value and the actual settled value. Assuming the clock period is T_{clk}, and the mean offset between the settled value and the desired value is $-\Delta I$, then an amount of current pulse with width T_{clk} and height ΔI can be added to the main DAC output to compensate the dynamic settling error. As for dynamic glitch errors, the majority of them are located in the early period of the transition region, and their width is relatively narrow compared to the clock period $T_{clk.}$ So an array of narrow pulses can be used to compensate the nonlinearities in this region. Just one narrow pulse with width T_p is used in this thesis for simplicity.

Fig. 3.5 is a conceptual illustration of the dynamic glitch calibration. Because of the nonlinearity, glitches exist during the transition period. If a waveform that is a vertical flip of the glitch can be generated and added to the output waveform at each transition period, the glitches can be canceled perfectly. But it is hard to generate such an analog waveform, so a digital pulse is used to compensate the glitch. The width and the magnitude of the pulse can be set as Δt and ε respectively. A pulse array with the same Δt but different magnitudes ε_{ni} (I=0,1,2,…,k) as shown in Fig.3.5(b) can be adopted for each step response to achieve a precise calibration. Each pulse is a Δt delay relative to the previous one. For simplicity, just one pulse is used to compensate for the glitch for every MSB code changing in this thesis. ε, the magnitude of the pulse, is related to the difference between X_n and X_{n-1}. For each step height, there is a corresponding ε value to compensate the glitch. So a look-up-table can be used to save the calibration pulse magnitude values. After the calibration, nonlinearity can be reduced and SFDR can be improved. The same mechanism can be applied to the dynamic settling error calibration but with the pulse width Δt_{st} as wide as the clock period T_{clk}.

3.4 Determine signal error E

The error of the converter as shown in the DAC model in the above section is a nonlinear function of the input sequence. The errors E at the output of the DAC can be expressed as

$$E = f(X_0, X_1, \cdots, X_{n-1}, X_n, \cdots) \tag{3.2}$$

where X_0, X_1, … and X_{n-1}, X_n, … is the input sequence. For most DACs, the output error is primarily attributed by the current input code and the previous input code, i.e. for an input code of X_n, the error will be given by E=f (X_{n-1} , X_n). The error can be divided into two categories: one occurs during the transition region, which can be called a dynamic glitch error caused by the nonlinearities during this region, and the other is the dynamic settling error attributed to the variation of the DAC output settled value compared to the desired one. A single tone sequence is used as a DAC input signal to allow Discrete Fourier Transform (DFT) analysis. The error is obtained by measuring the distortion in the DAC output spectrum through DFT. These distortion terms are then used to get the time domain error waveform through Inverse Discrete Fourier Transform (IDFT). In order to calibrate the output waveform, the error values must be known for every pair of current input code and previous code (X_n, X_{n-1}). The CALDAC will judge the current and previous input code and generate the required calibration current or voltage to the main DAC output according to the error value related to (X_n, X_{n-1}). So, an error look-up table is needed to store the error value with x axis X_n and y axis X_{n-1}. After the calibration, the nonlinearities will reduce and the SFDR will improve.

A diagram which demonstrates the procedure of exacting the error look-up table for single input frequency is shown in Fig. 3.8. First, a single tone sinusoid waveform is sampled and quantized to work as the initial input sequence to the main DAC. Assume, during the simulation interval, that the input signal period number is M, the clock period number is K and frequency is f_{clk}. Then, the input signal frequency f_{sig} is:

$$f_{sig} = \frac{M}{K} \times f_{clk} \tag{3.3}$$

where M and K should be coprime. For each clock period, set the sampled number to L, then the total length of the sequence, N, will be $N = K \times L$. The pattern source cycles through the samples, repeating the sequence every N/f_{clk} second, and the DAC translates these samples into an analog signal. These analog signals usually contain power at frequencies not present in the input sequence X(N), and the distortion power can be measured from the spectrum to determine the error waveform used for calibration later. The DAC output data $X_{sig}(N,t)$, with information of input code and analog output, is exported to a Matlab algorithm program. In this program, FFT is first done to $X_{sig}(N,t)$ to obtain the spectral characteristic in frequency domain. The sequence, X(N), that is driving the DAC is a single-tone signal. So, it has a DFT representation as:

$$X(n) = \sum_{k=0}^{N-1} X_k \, e^{\, i 2\pi k n / N} \tag{3.4}$$

The frequency component X_l is:

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} X(n) \, e^{-i 2\pi k n / N} \tag{3.5}$$

The spectrum of X(N) has N distinct bins for the sequence.

The value of X_k represents the phase and magnitude of each spectrum component. For an ideal DAC, only the fundamental signal components $f_{sig}=f_{clk}*M/N$ and its reflected pair contain energy. As for the practical DAC measurement, any energy appearing in other components will be regarded as error energy.

The frequency domain output $X_{sig}(N,f)$ after DFT will contain both the fundamental frequency component power and the distortions power. The aim is to exact error information used for calibration, so the fundamental frequency component pairs need to be removed.

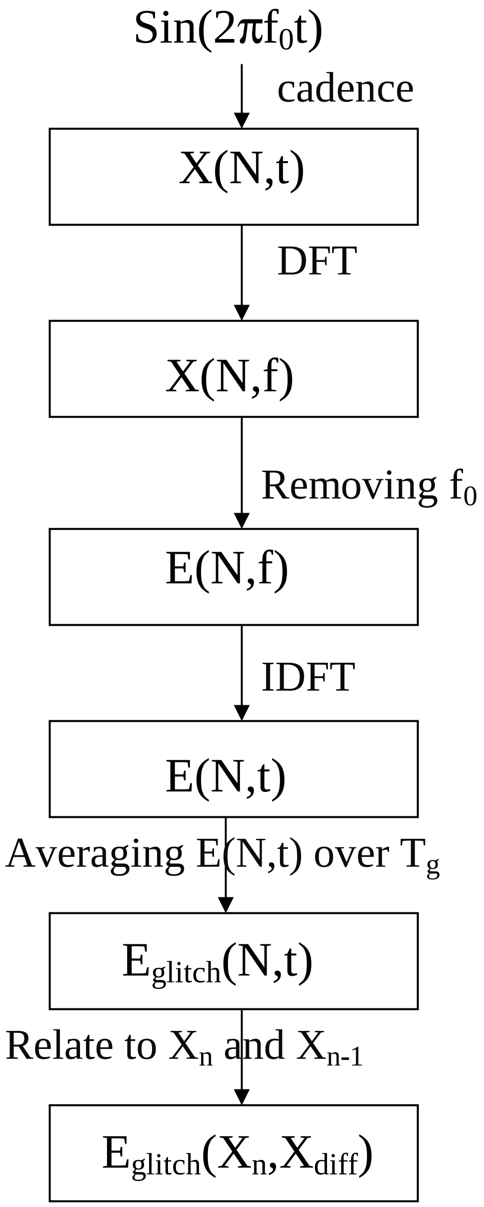


Figure 3. 8 Dynamic glitch error value exaction procedure

After the removal, the left signal should be pure distortion power, E(N,f). Actually, the most significant power components are the first several harmonics. So the effect of the noise floor is negligible. IDFT is done to E(N,f) to get the time domain error sequence related to input sequence X(N). As discussed in the previous section, there are two categories dynamic error: dynamic glitch error occurring during the transition region and dynamic settling error attributed to the variation of the DAC output settled value compared to the desired one. Compared to the dynamic settling error, the dynamic glitch error interval is much shorter. Therefore, a narrow pulse is used to compensate for the dynamic glitch error and a pulse as wide as the clock period is used to compensate for the dynamic settling error. Assuming the error compensation pulses are $E_{glitch}(x_n, x_{diff})$ with pulse width T_g for dynamic glitch error and $E_{settle}(x_n, x_{diff})$ with pulse width T_{clk} for dynamic settling error respectively, the magnitude of the pulse height of $E_{glitch}(x_n, x_{diff})$ is obtained by time averaging the error values in the region of T_g for each transition. Where $X_{diff} = X_n - X_{n-1}$, the step jumping between two consecutive input codes is obtained from X(N,t). $E_{glitch}(x_n, x_{diff})$ is a function of x_n and x_{diff}, so a 3-dimension table can be built to store the compensation error magnitude values related to x_n and x_{diff} for each input frequency. The dynamic settling error $E_{settle}(x_n, x_{diff})$ can be obtained in a similar fashion where the averaging of the error values is carried out over the whole clock period for each transition.

As for a realistic DAC calibration, the look-up-table should be used to calibrate any input frequency signal after it is built. To construct the full-scale error look-up table, the DAC input signals are structured so that the DAC is excited over the entire DAC Nyquist bandwidth with significant distortion terms within the observation band to get small error look-up tables for every input frequency. A big, full-scale look-up table can be built by combining the small error look-up tables together.

To implement the combination, there is a technical issue: how to deal with the overlapping values for the same (X_n, X_{diff})?

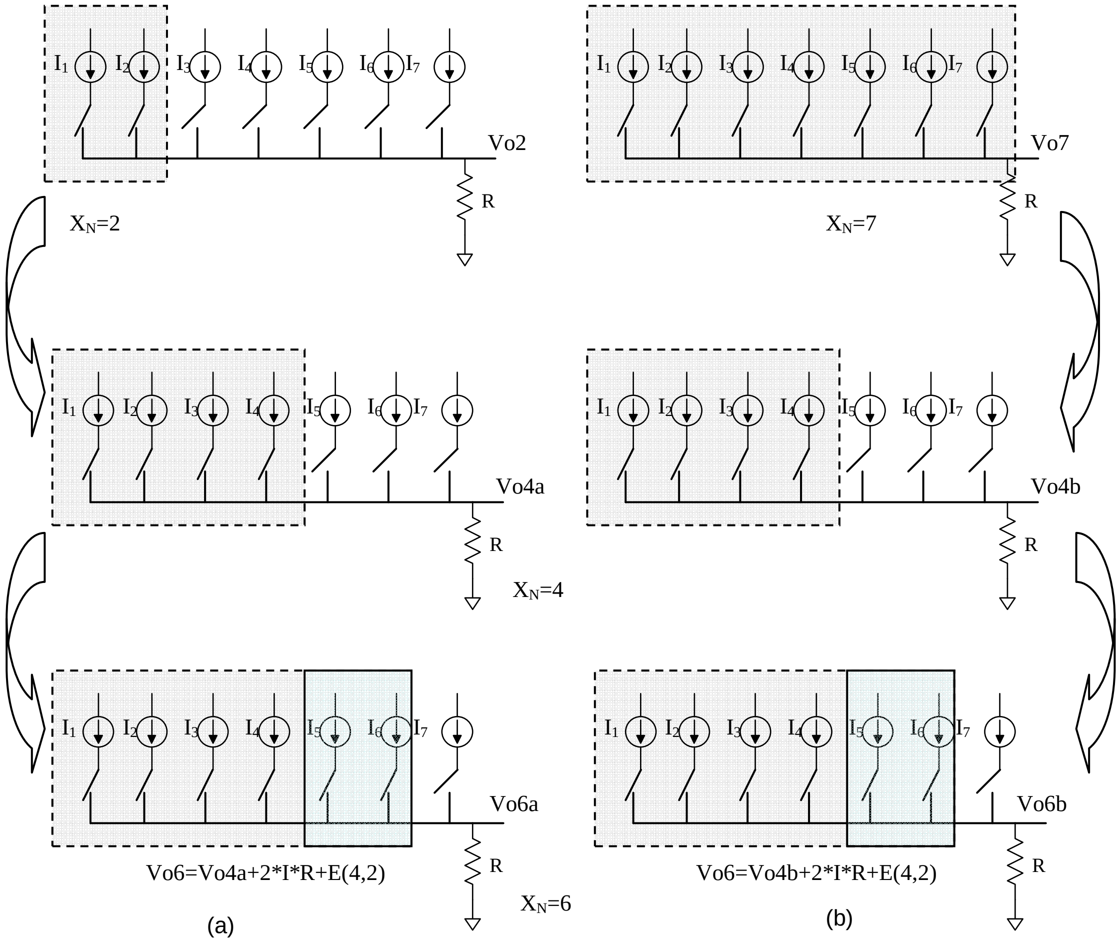


Figure 3. 9 Dynamic errors illustration for a 3 bits current steering DAC

The assumption of this approach is that the error look-up table is mainly the current input code and the difference of the current and previous codes dependent. It means for an input code of X_n, the error will be given by E=f (X_n, X_{diff}), and E is frequency independent or nearly independent. This is easy to understand because no matter what frequency it is, take thermometer current-steering DAC for example, the output current is obtained from certain current sources. For the same current input code, for example, X_n, the current sources that are used to generate the required current should be the same. Thus, no matter what the input signal frequency, if the jumping X_{diff} is equal, the current used to calibrate the nonlinearity should be the same or very close for the same X_n.

An example of 3-bit current steering DAC is shown in Fig. 3.9. The DAC consists of seven identical current sources I1~I7, where $I_1=I_2=\ldots=I_7=I$. For input code K, there will be k current sources I_1~I_k turned on, the output voltage V_{ok} then is k*I*R + E_k , where E_k is the output error compared to ideal output. In case (a), the input code sequence is: 2→4→6. So the number of closed current sources for each stage is 2, 4 and 6, and the output voltages are V_{o2}, $V_{o4a,}$ and V_{o6a} respectively, where $V_{o6a}=V_{o4a}+2*i*R+E(4,2)$. For case (b), the input code sequence is: 7→4→6, the output voltage values are V_{o7}, $V_{o4b,}$ and V_{o6b}, where $V_{o6b}=V_{o4b}+2*I*R+E(4,2)$. V_{o6a} may not equal V_{o6b} since V_{o4a} and V_{o4b} may not be equal. But the difference between V_{o6a} and V_{o4a} and between V_{o6b} and V_{o4b} are both E(4,2). So, the error for the transition from 4→6 will be reduce or eliminate if E(4,2) is compensated for from the actual DAC output. Therefore, dynamic linearity improvement will be achieved after calibration.

3.5 Current-steering DAC calibration scheme

The current-steering DAC model contains additive static and dynamic errors. The major static error sources are due to mismatch between current sources and their finite output impedance. The current source mismatch is mainly due to the random variations that can be attributed to local random variations and gradient effects [2]. The output current of the current sources may vary with the output voltage because the output impedance of the current sources is not infinite. These error sources cause inaccurately settled values for DAC outputs. Moreover, the slewing rate limit, insufficient settling time, glitches etc. also introduce nonlinearity to the circuit. Both static and dynamic nonlinearities contribute to undesired harmonics in the output.

For static errors, which are main contributors to inaccurate output settled values, are regarded as only a function of input code X_N. They can be calibrated efficiently by Cong's approach. In this thesis, the main DAC is after static error calibration.

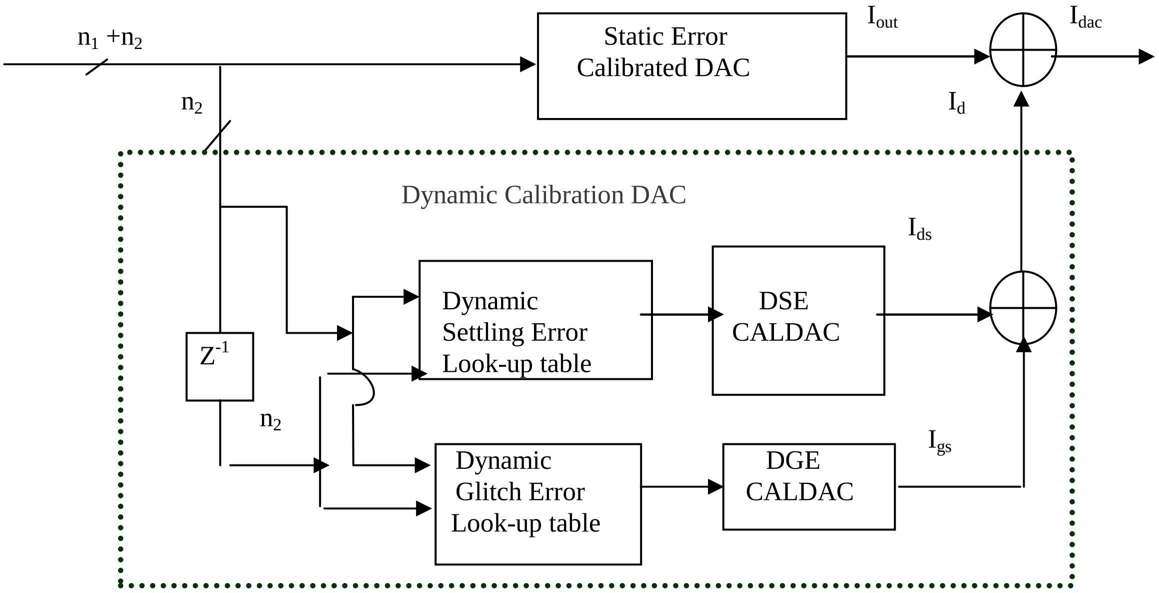


Figure 3. 10 DAC calibration scheme

Fig. 3.10 shows the architecture of a DAC with its dynamic calibration. It consists of a static-error-calibrated DAC as used by Cong in [2], a Dynamic Calibration DAC, and a delay block. The Dynamic Calibration DAC block is divided into dynamic settling block and dynamic glitch block. The dynamic settling block includes a Dynamic Settling Error (DSE) look-up table and a Dynamic Settling Error Calibration DAC (DSE CALDAC), the dynamic glitch block includes a Dynamic Glitch Error (DGE) look-up table and a Dynamic Glitch Error Calibration DAC (DGE CALDAC) block. The delay block is used to get the previous code, $X_{n-1,}$ for dynamic calibration. Based on X_n and X_{diff}, the dynamic glitch error can be obtained from DGE and the current pulses I_{ds} will be generated from DGE CALDAC. Similarly, I_{gs} for dynamic settling error compensation will be generated. These pulses will be then added to the original static-calibrated DAC output to obtain the desired DAC output. The raw DAC is segmented into a most significant bit (MSB) part (n_2-bit) and a least significant bit (LSB) part (n_1-bit). The thermometer decoding is used in the MSB part to reduce the output glitches. The LSB part is implemented with a binary structure.

CHAPTER 4. SIMULATION RESULTS AND DISCUSSION

In this section, we will present work on behavior mode and circuit implementations of current-steering DACs. Throughout the discussion in the previous chapter, we identified current-steering DACs as a suitable candidate for high-speed and high-resolution communication applications. This architecture does not need any output buffer compared to switched-capacitor DACs. It will, however, become sensitive to finite output impedance. Furthermore, current-steering DACs can be implemented with MOS-only components and still reach rather high accuracy. Resistor-string or R–2R ladders are also very fast, but they require high-accuracy, on-chip resistors. We focus on the pure current-steering versions where a number of weighted current sources are used to form the conversion function.

Chapter 3 outlined the dynamic errors in DACs and a procedure to determine input-code-dependent harmonic errors in a DAC. To further evaluate the validity of this approach, we present the results from implementation of a 15-bit AHDL behavioral-mode, current-steering DAC and a 12-bit transistor-level, current-steering DAC in section 4.1 and 4.2 respectively. We show design tradeoffs and ideas for how to implement the required circuit elements and calibration procedure. Simulation results from the two DACs are also presented and discussed in this chapter. Output spectrum and SFDR improvement from single-input frequency error look-up tables and full-scale error look-up tables are also shown. We have found that DACs' dynamic performance can be dramatically improved through this novel approach. Furthermore, results show that SFDR improvement is mainly from the dynamic glitch calibration after free-of-transistors mismatching.

4.1 15-bit current-steering DAC behavior mode prototype

4.1.1 DAC calibration scheme

A 15-bit DAC was built to verify the validity of the approach. Fig. 4.1 is the whole-circuits, top-level illustration of the test setup. It consists of an ideal 15-bit analog-to-digital converter, binary-to-thermometer decoder, buffer, and a 15-bit raw DAC where the ADC implemented in AHDL code is used to generate the input codes to the DAC. As in discussion in chapter 2, segmentation is applied to current-steering DAC to obtain quite good linearity at reasonable area. The raw DAC is segmented into a 8-bit binary LSB section and 7-bit thermometer MSB section. Since most of the nonlinearity errors come from the MSB part, only the 3-bit upper MSB current sources array are implemented with transistors level to reflect the nonlinearities in the circuit, while the 4-bit low MSB and the 8-bit LSB DAC is a

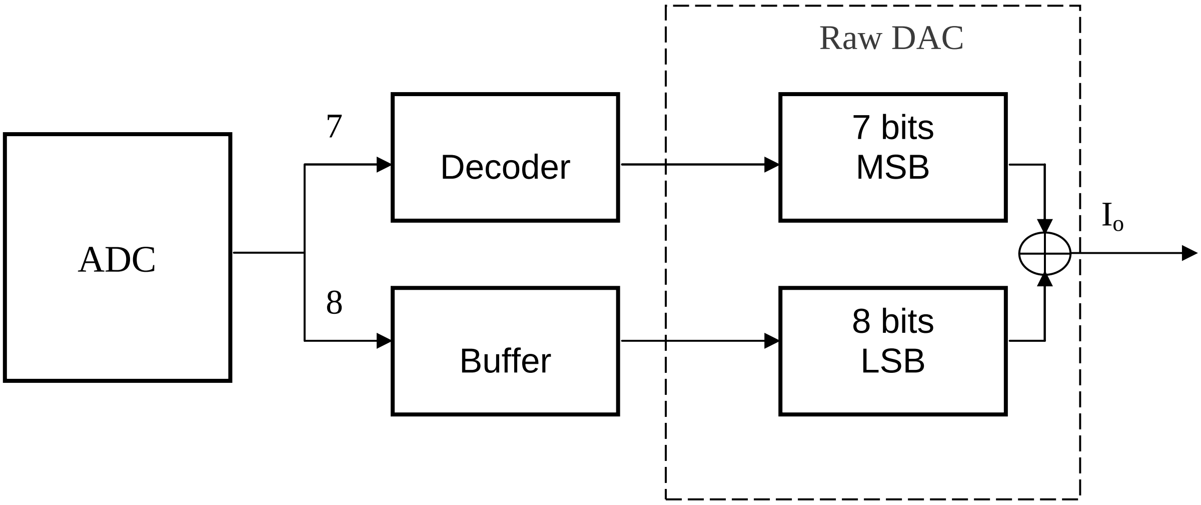


Figure 4. 1 DAC test diagram

AHDL behavior model. The decoder is used to transfer the ADC 7-bit MSB outputs from binary code into thermometer codes which serve as input signals to the raw DAC MSB. A buffer is inserted between the ADC and the DAC LSB parts to simulate the latency time of decoder. All the decoder and buffer are also implemented with an AHDL behavior mode. Rise time and fall time are set in the behavioral mode to reflect the real circuits' propagation time.

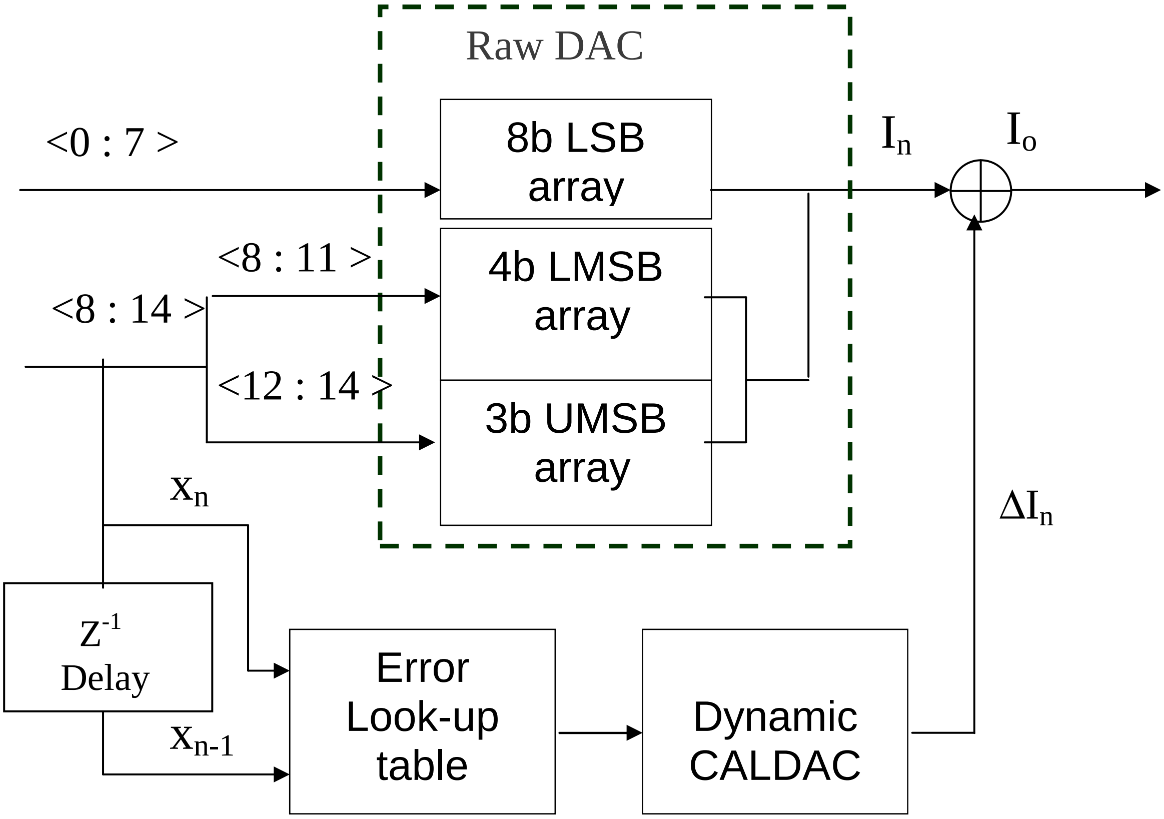


Figure 4. 2 DAC calibration scheme

Fig. 4.2 is illustration of the calibration scheme. The dynamic calibration is applied only to the 7-bit MSB part. Among the 7-bit MSB part, the upper 3 bits (UMSB) are implemented with transistor level while the lower 4 bits (LMSB) are built with behavior mode. A delay block which is also behavior mode is used to generate a previous input code x_{n-1} according to the current input code x_n. The error look-up table will judge the current input code x_n and previous code x_{n-1} obtained from delay block then visit the corresponding cell (x_n, x_{n-1}) to read the compensation value $E(x_n, x_{n-1})$. Then, the $E(x_n, x_{n-1})$ will drive the calibration DAC “Dynamic CALDAC” to generate a current ΔI_n which is summed to the raw DAC output to compensate for the nonlinearities.

4.1.2 Simulation results

The calibration is done for three cases:

a) Dynamic settling error calibration

b) Dynamic glitch error calibration

c) Both dynamic settling and glitch errors calibration

As discussion in chapter 3, the dynamic settling error calibration is done by compensating the nonlinearities using only error wave E_{settle} with unit pulse as wide as the clock period, while for the dynamic glitch error calibration, the unit-pulse width of compensation error wave E_{glitch} is much narrower than the clock period. In the case of calibration for both dynamic settling and glitch errors calibration, both E_{settle} and E_{glitch} are used to compensate for nonlinearities.

In order to compensate for the dynamic error, it is better to calibrate as many bits as possible. But, the complexity will increase dramatically as the number of calibrated bits increases. However, the improvement of SFDR may be not obvious if the calibration bits number is too small resulting in insufficient information in the look-up table. So a tradeoff exists between the calibration bits. In this thesis, the 7-bit MSB (3-bit thermometer transistor level for upper MSB and 4-bit binary behavior model for lower MSB) is under calibration. Current sources with cascode structure are used in 3-bit upper MSB to increase the output impedance and reduce the current source drain voltage variation. The clock frequency f_{clk} is 200MHz. The dynamic glitch compensation pulse width is 400ps.

The calibration results are shown in Fig. 4.3. From the plot, it can be seen that the SFDR values before and after calibration are close at low frequency. But, SFDR values after dynamic calibration remain nearly constant until the input signal frequency reaches 11MHz where the SFDR improvement is about 25dB. Although the SFDR values after dynamic calibration decrease as the input signal frequency increases, the improvement is still over 20dB. This results show that the SFDR can be greatly improved by this approach.

The SFDR improvement between dynamic glitch calibration and both dynamic calibrations are very close, while the SFDR values of dynamic settling calibration and before calibration are nearly the same. The results show that the SFDR improvement is mainly from the dynamic glitch calibration after free-of-transistors mismatching.

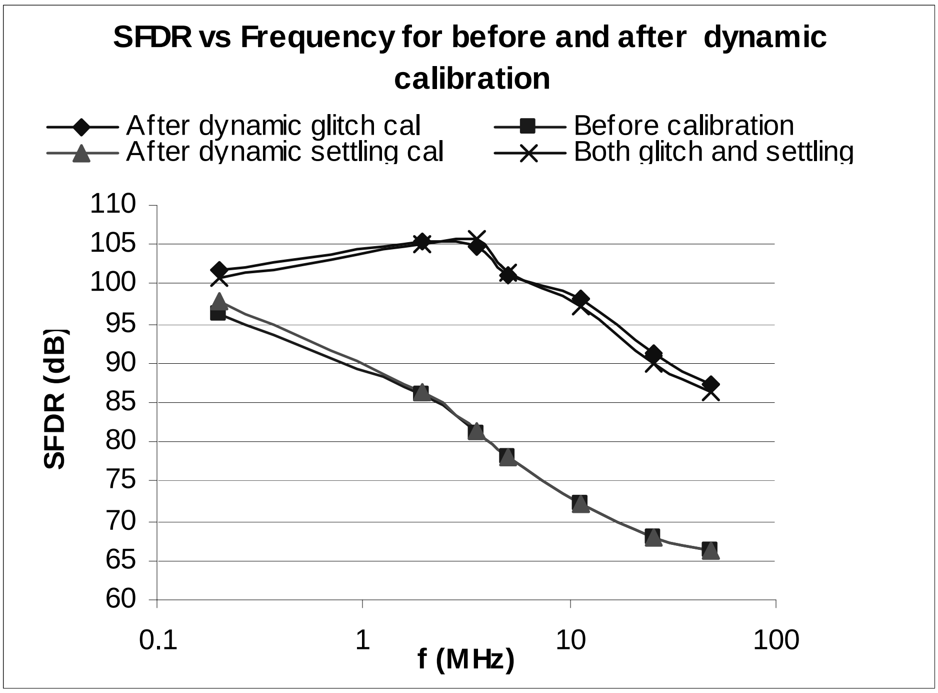


Figure 4. 3 SFDR vs. input signal frequency between before and after dynamic calibration

4.1.3 Robustness of the calibration approach

The effectiveness of this approach is proved in the simulation. But, as a calibration approach, robustness is also critical. In order to evaluate the robustness of this approach, the time interval, Δt, and the delay, Δt_d, of the pulse used for dynamic glitch compensation are varied to observe the SFDR of the signal after dynamic glitch calibration. The calibration pulse shapes are also studied to verify the approach.

1) SFDR vs. compensation pulse width

The center value of the compensation current pulse width is set as 0.4ns, the input signal frequency, $f_{o,}$ is about 2MHz. The simulation results are shown in Fig. 4.4. When the width changes ±0.1ns(±25%), the SFDR value change is about 4dB (4%). It is within the

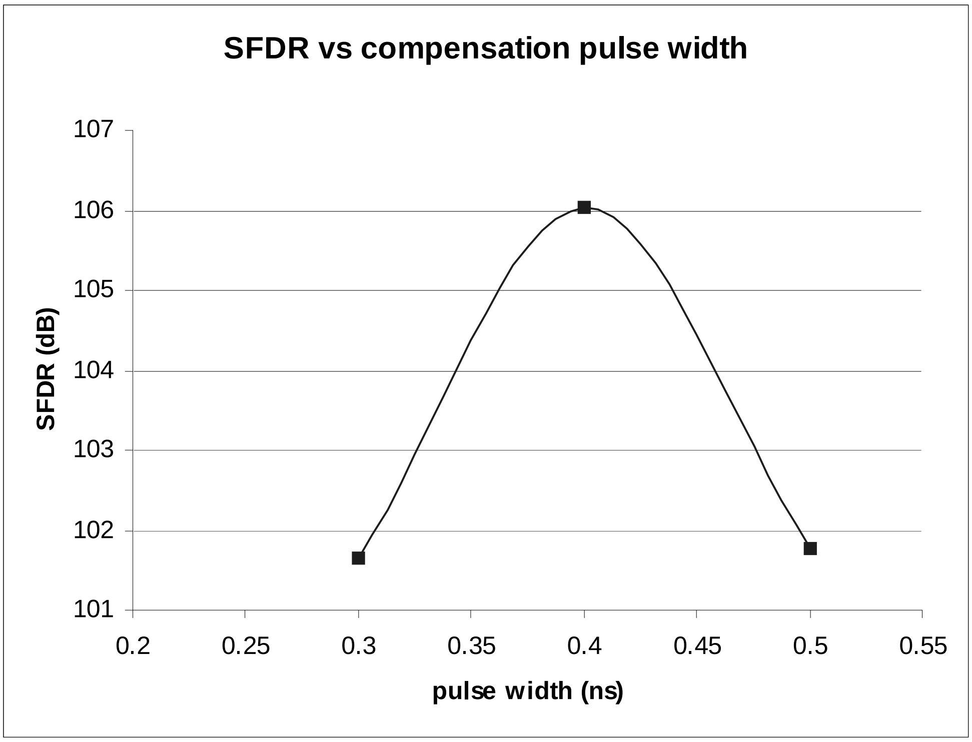


Figure 4. 4 SFDR vs. compensation pulse width

tolerate range. Because it is not difficult to adjust the normalized width within 5%, this result indicates that the compensation pulse width variation has little effect on the SFDR improvement.

2) SFDR vs. compensation pulse delay

In the above calibration, it is assumed that the compensation pulse is aligned with the raw DAC output. But that may not be the case in reality. A time interval between the raw DAC output and the calibration pulse, Δt_d , may exist. This delay will have an impact on the SFDR. The effect of different Δt_d to SFDR is shown in Fig. 4.5. In the simulation, the compensation pulse width is 400ps and the input signal frequency is about 11.3MHz. The magnitude of SFDR decreases from about 98dB to about 87dB when the delay width is from

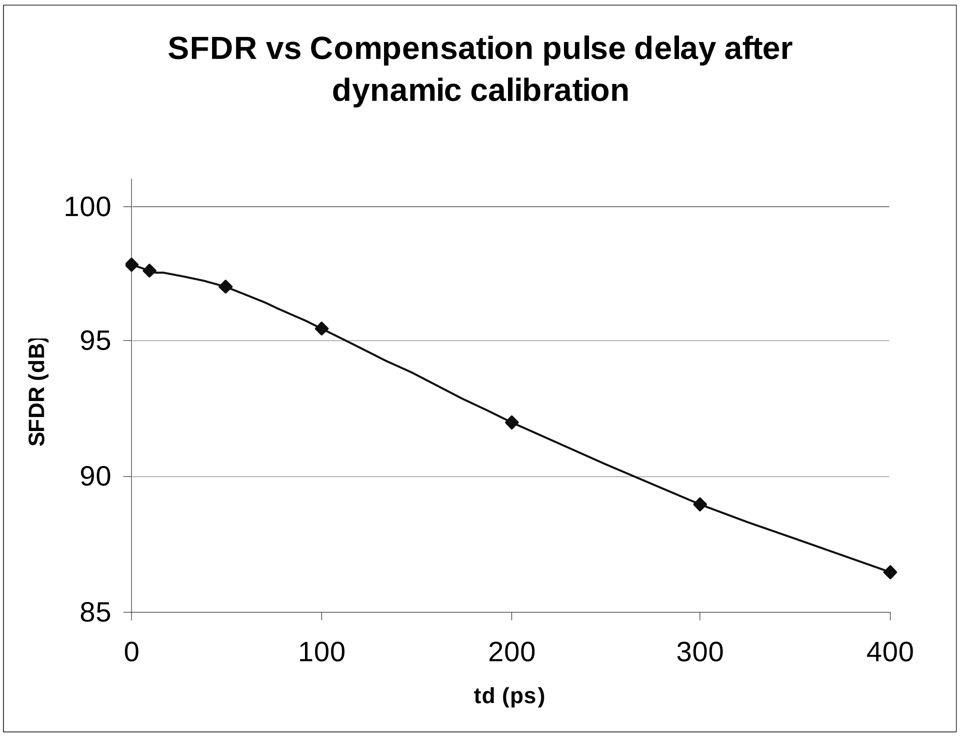


Figure 4. 5 SFDR vs. compensation pulse delay

0 to 400ps. If the delay can be controlled under 200ps, the degradation of the SFDR is less than about 6%. The requirement of this control is not difficult, so it is not a significant concern in the design. Both results of 1) and 2) show that this calibration approach is robust.

3) SFDR vs. shape of compensation pulse

In the calibration process, not only the robustness of the approach but also the implementation possibilities must be considered. In the previous calibration, the pulses used in calibration are rectangle waveforms. In reality, it is a challenge to get rectangle waveforms with widths of half a nanosecond. It is much easier to generate a triangle or near triangle waveform. If the SFDR improvement using triangle waveforms for compensation can achieve the same or just below that of rectangle waveforms, it will much relax the realization of the calibration. Two kinds of triangle waveforms (a) and (b), as shown in Fig. 4.6, are used to replace the rectangle waveform where the height is normalized to the height of the rectangle waveform. The SFDR improvement for different compensation pulse shapes is shown in Fig. 4.7. The results show that the SFDR curves are within the 3dB range. That means the shapes of the triangle have little effect on SFDR improvement given the same compensation pulse energy area.

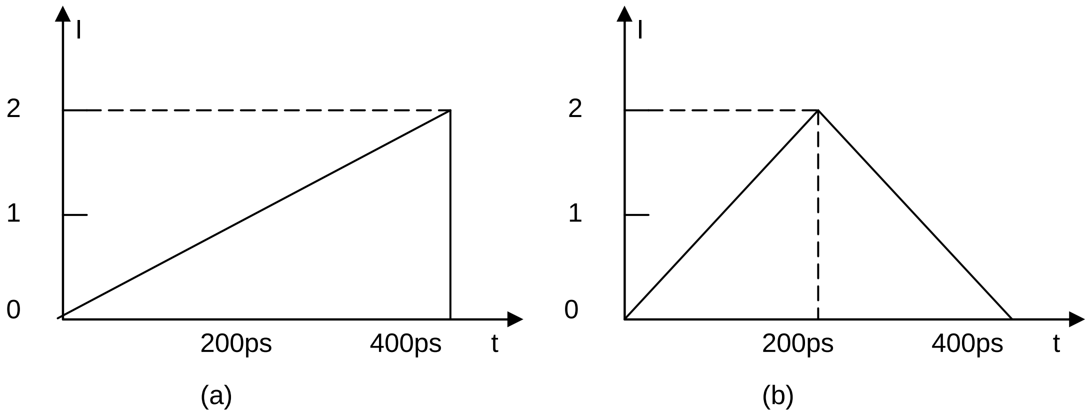


Figure 4. 6   Two calibration triangle waveforms

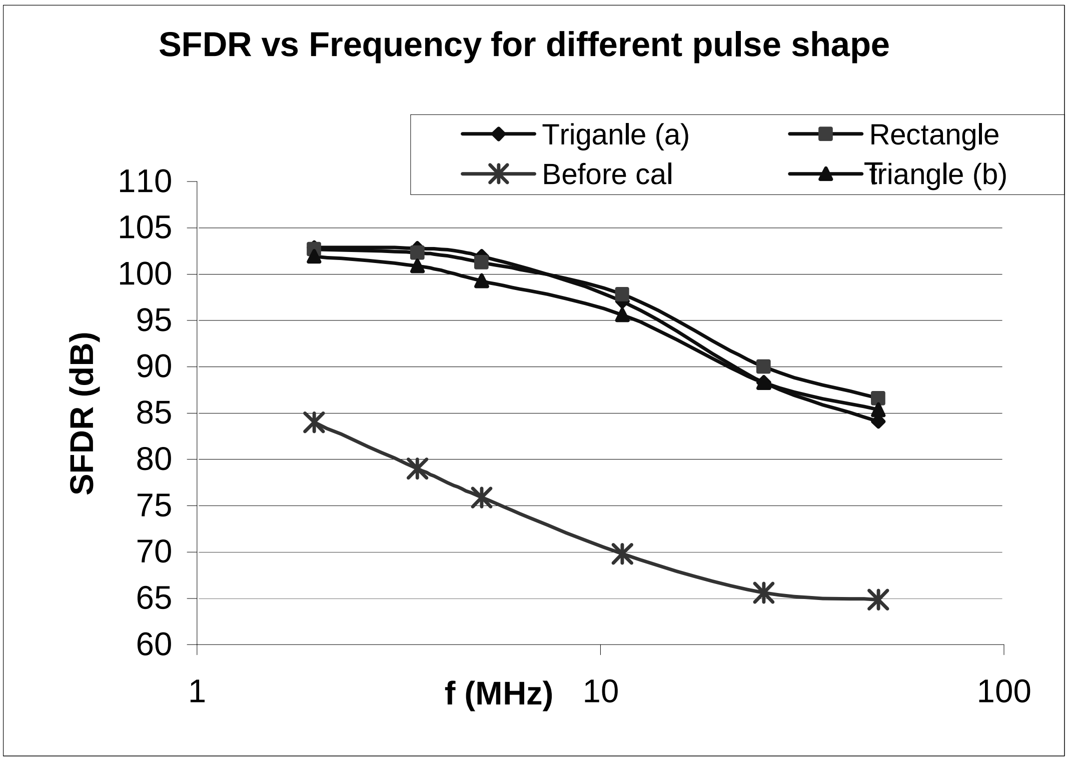


Figure 4. 7 SFDR vs. input signal Frequency for different compensation pulse shapes

4.1.4 Conclusions

A technique for DAC dynamic nonlinearity calibration has been presented. The dynamic errors were assumed to be a function of the two successive input digital codes. A 15-bit current steering DAC was designed to verify the validity of this approach. The simulation results showed that generating an appropriate amount of current pulse and adding it to the raw DAC output current at each transition period can dramatically attenuate the input code dependence. Therefore, the SFDR can be significantly improved through this approach.

The simulation results also showed that SFDR improvement is mainly achieved from the dynamic glitch calibration, which means that most of the nonlinearities come from the early period of the transition region. The robustness of the approach was also proven in the simulation. The pulses with different shapes were applied to the dynamic glitch calibration, and the simulation showed that given the same compensation pulse energy area, the shapes have little effect on improvement of the DACs' dynamic performance. Therefore, this approach was shown feasible.

4.2 12-bit current-steering DAC transistor level prototype

4.2.1 DAC structure

In section 4.1, we proved that the DACs' SFDR can be significant improved by a new approach through a 15-bit DAC behavior mode. To further illustrate the validity of this approach, we replace the behavior mode blocks with transistor level to reflect a more realistic situation in the circuits. A 12-bit DAC as shown Fig 4.8 is used as a prototype in this thesis. A segmented architecture is employed in the DAC design. The current-steering DAC consists of a 6-bit thermometer MSB sub-DAC and a 6-bit binary LSB sub-DAC to achieve good performance at reasonable area. Since most of the nonlinearity errors come from the MSB part, only the MSB part is calibrated. The 6-bit MSB sub-DAC array consists of 63 MSB current sources. The digital binary input codes of the 6-bit MSB part transfer to 63 bit thermometer codes by a 3x3 row-column decoder, followed by latches for signal synchronization. The digital input codes to the 6-bit LSB part pass through buffer and latches to achieve the same delay and, therefore, synchronize with the input codes in the MSB part. Cascode structure is used in all current sources to obtain high-output impedance in order to reduce the current source drain-voltage variation. An ideal ADC is used to generate the 12-bit input codes.

An overview of different blocks designs is given first in this section. Some practical issues are also discussed during the introduction.

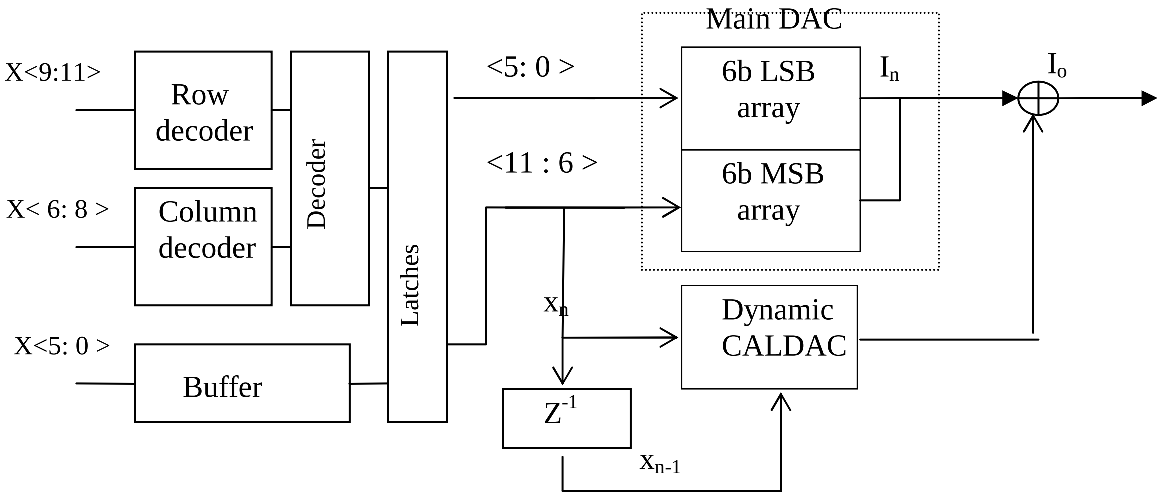


Figure 4. 8 Current steering DAC structure

4.2.2 Architecture

In DAC design, not only the current sources introduce error but also the peripheral blocks such as decoders and switches contribute significant nonlinearities. In the behavioral mode, these errors cannot be fully included. So, transistor level blocks are used to replace the behavioral mode to reflect these nonlinearities. The ideas and issues of practical design of decoders, latches and, switches are discussed in the following section. The current sources design is also introduced.

4.2.2.1 Row-column decoder

As discussed in the section above, segmentation can offer benefits in terms of DNL and performance of a DAC. However, for high resolution DAC design, the number of MSBs becomes larger in the segmentation in order to achieve a low DNL and to minimize current source area, while maintaining a desired yield level. In this case, the complexity in the decoding logic becomes a major drawback in this type of architecture. The digital decoder results in longer delays in the digital decoder and affects the speed. Furthermore, it occupies a large amount of silicon area for the logic circuit. Therefore, the decoder for the MSB thermometer-coded structure is divided into a row and a column decoder so that the logic in the decoder is greatly reduced.

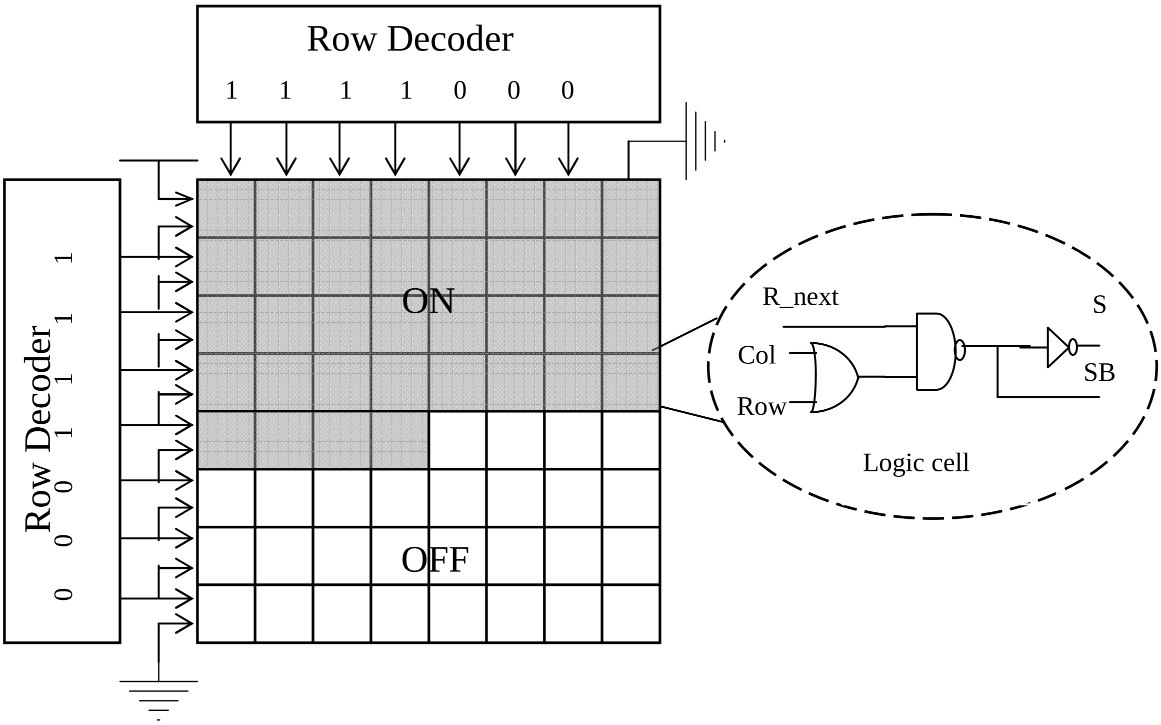


Figure 4. 9 Two-step decoding

After generation of the digital signals, the 63 outputs of the thermometer decoder are obtained by transferring the 6 most significant input bits from binary to thermometer. For a given input code, a simple logic circuit at each logic cell, as shown in Fig. 4.9 [26], will decide whether to turn on or off the corresponding current cell. Then, a region like the shadowed portion in Fig. 4.9 will generate the control signal that will select and turn on the current sources to which it connects. The above described decoding logic has been implemented with NAND and NOR logic as shown in logic cell in Fig. 4.9.

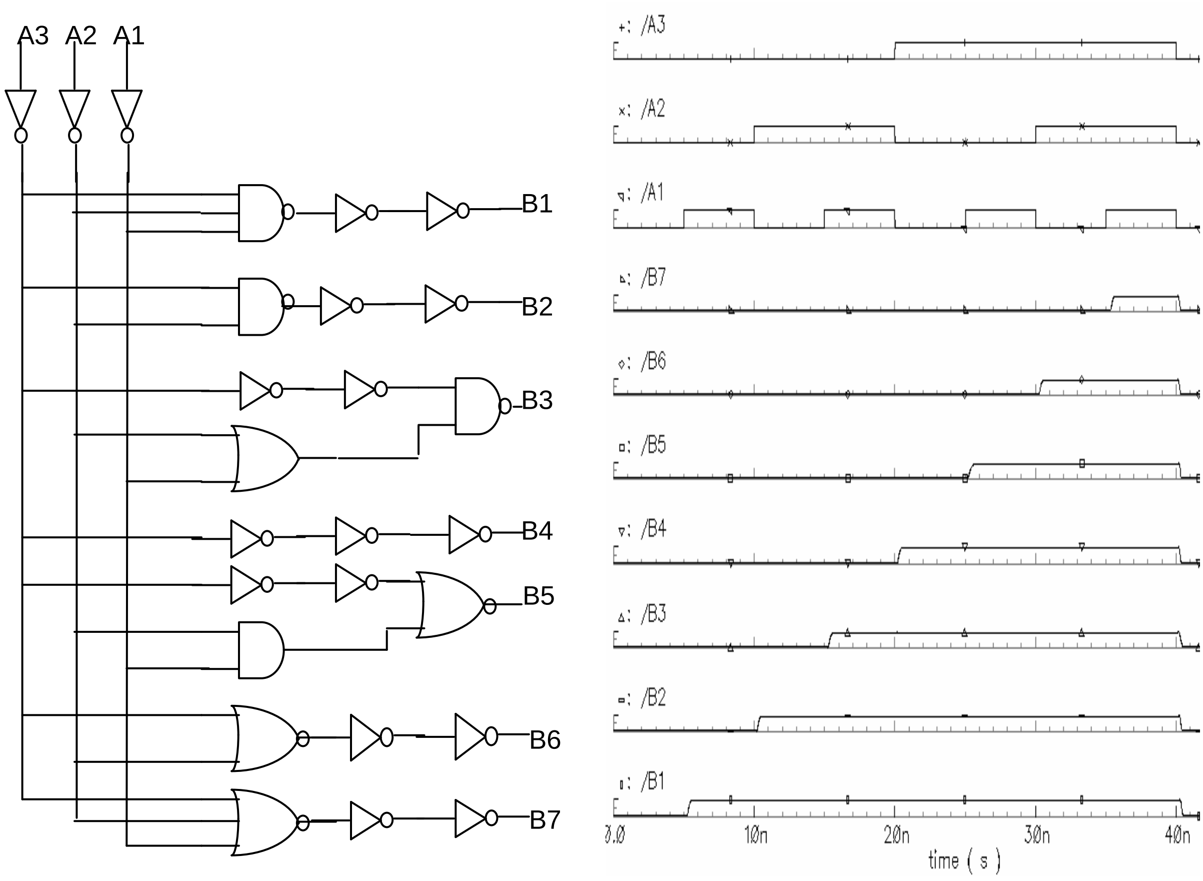


Figure 4. 10   A 3-to-8 row decoder circuit and the outputs for a ramp

According to the input codes, the decoder matrix consists of three types of rows. They are rows in which all of the current cells are turned on; rows in which all of the current cells are turned off; and a certain row in which current cells are turned on depending upon the column decoder signal.

In consideration of these three types of rows, a two-step decoding logic has been developed. The details of the decoding are as follows. In the first step, digital inputs are decoded in the row decoder and column decoder. The number of flags in the columns corresponds to the input value of the column decoder. The number of flags in the rows corresponds to the input value of the row decoder plus one. In the next step, each logic gate in the current cell identifies the row type described above by comparing one row signal with the one next to it. If both of the row signals are at a high level, then the current source is turned on regardless of column signal. If the two row signals are different, then the current source is turned on depending upon the column signal. This operation can be achieved by using peripheral decoders and NOR and NAND gates in the logic gate cells in two logic stages. The inverters inside the cells are used for buffering and complementary signals generation. One local latch has been inserted between the cells and the corresponding current source to synchronize the output switch signal and to suppress the glitch.

The actual row and column decoder circuits are shown as Fig. 4.10(a). NAND and NOR logic gates are used to implement the 3 to 8 decoding. The inverters at the input node and between the logic gates are used for the generation of the complementary signals and for buffering to achieve a time delay. The fan-out and fan-in in each logic gate are optimized to enhance the decoding speed. For the 3-bit binary input codes, A_1~A_3, the thermometer outputs are B_1~B_7. Fig. 4.10(b) is the simulated output for the decoder. It can be seen that the output increases one by one with a 3-bit input ramp.

4.2.2.2 Latch

Latches have the function of storing digital signal values between two consecutive clock cycles and operating the switches only at the active-clock edge. For the correct operation of a DAC, it is very important that one switch is always turned on within the current cell so that the current can flow into either one of outputs. while the switches are controlled by the switching-control signals produced by the latches.

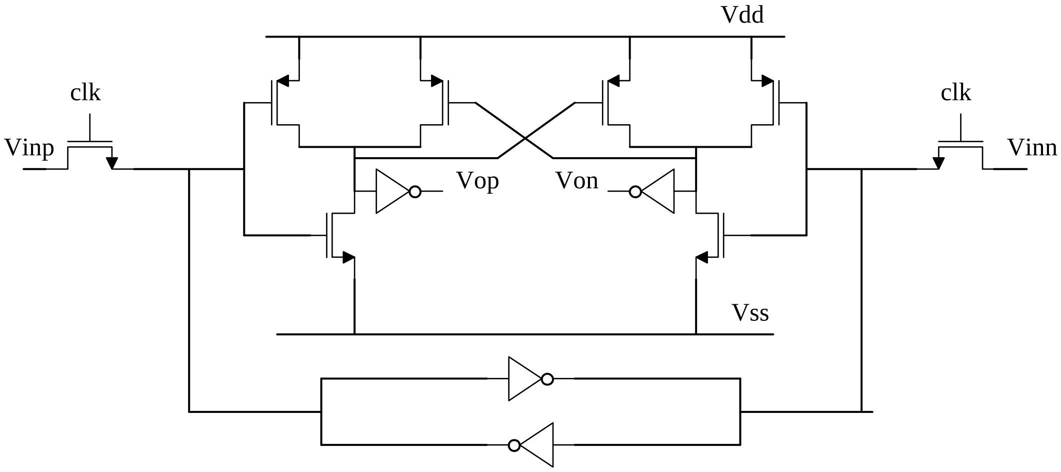


Figure 4. 11 Latches structure

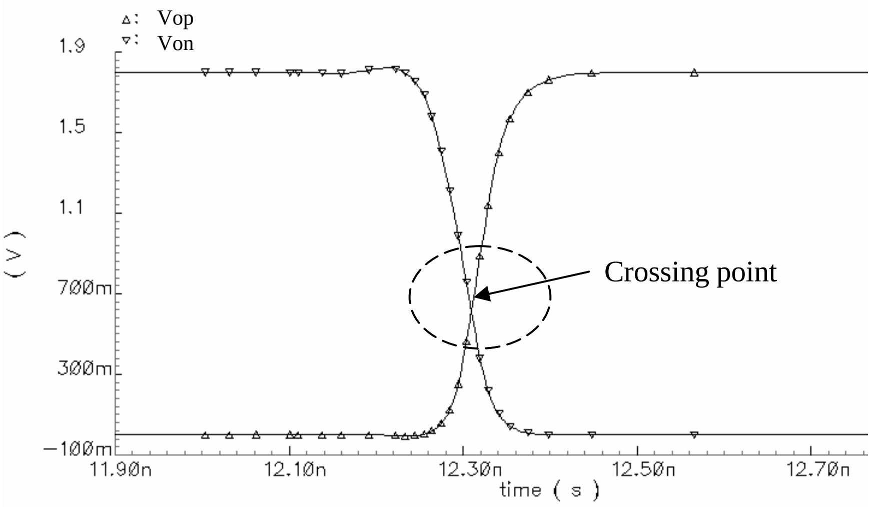


Figure 4. 12 Latch output signal crossing point

It is widely known that both inaccurate settled values and nonlinear switching transient contribute to spectral harmonics in DAC output. These harmonics are major factors limiting the spurious free dynamic range (SFDR). The inaccuracy of the settled values is mainly due to static error, while the nonlinearity of switching transient is primarily due to parasitic effects in the current source cells and the nonsynchronous control signals. There are some important issues that have been identified that cause dynamic limitations by switching [23]: imperfect synchronization of control signals at the switches; drain-voltage variation of the current-source transistors; and coupling of the control signals through the switches to the output.

A well-designed synchronized driver is used to minimize these three effects. In the case of a traditional-switch driver, both switches can be turned off simultaneously for a short period of time. The drain capacitor of the current source transistor will be charged or discharged during this time interval, which will introduce a significant glitch and deteriorate the dynamic performance of the DAC. This phenomenon can be attenuated through a traditional-switch driver by shifting the crossing point of the switch transistors' differential control signals. This method prevents these transistors from being simultaneously in the off state. The difference in delay between the different digital decoder outputs can also be minimized by placing the driver in front of the switches and through careful design, therefore, the final synchronization is performed. Furthermore, the dynamic error caused by the parasitic gate–drain feedthrough capacitance is significantly lowered by the use of a reduced voltage swing at the input of the switches. This reduced voltage swing is achieved by lowering the power supply of the digital driver.

A high-speed rise/fall-time-based circuit has been suggested [23] to work as a driver. The driver is based on a simple latch. An extra PMOS input circuit is placed in parallel with each of the cross-coupled PMOS transistors situated at the top of the circuit. The input signals are then connected to the PMOS through a NMOS switch controlled by a clock. As a result, the output nodes can be instantaneously charged to high voltage level when the input falls low, while it reaches low voltage level when the input is high.

In this structure, the intrinsic delay is eliminated from the circuit's operation as charging and discharging starts at the same moment. The crossing point can be controlled by the scaling of the gate width of the PMOS and NMOS transistors. In this circuit, the PMOS positive feedback loop results in a rise time that is much faster than the fall time of the driver circuit. Therefore, a high crossing point of the differential outputs is available at the output of the latch. The outputs can be used directly for NMOS DAC implementation. In PMOS implementation, the low crossing point of the differential output can be realized by scaling the NMOS gate width up and the PMOS gate width down. An inverter can also be placed, as shown in Fig. 4.11 and Fig. 4.12, in the circuit after the outputs of the driver to convert the high crossing point to a low crossing point.

An additional latch formed by the small invertors can be inserted between the two inputs of the latch to suppress the clock feedthrough by the pass transistors and stabilize the synchronized inputs.

4.2.2.3 Current source design

For high-speed, high-accuracy DACs, a segmented current steering topology is usually chosen, as it is intrinsically faster and more linear than competing architectures. The conceptual block diagram of this type DAC is depicted in Fig. 4.13: the l least significant bits are implemented in binary, while the m most significant bits steer a unary current source array.

The general specification for a current-steering DAC can be divided into static, dynamic, environmental, and optimization specifications. In the case of a DAC, the static parameters include static accuracy, integral nonlinearity (INL), differential nonlinearity (DNL), and yield. The dynamic parameters include the settling time, glitch energy, spurious-free dynamic range (SFDR), and sample frequency. The environmental parameters include the power supply, digital levels, output load, and input/output range. The power consumption and area are the optimization targets and need to be minimized for a given technology. In this paper, we focus on DACs' dynamic property.

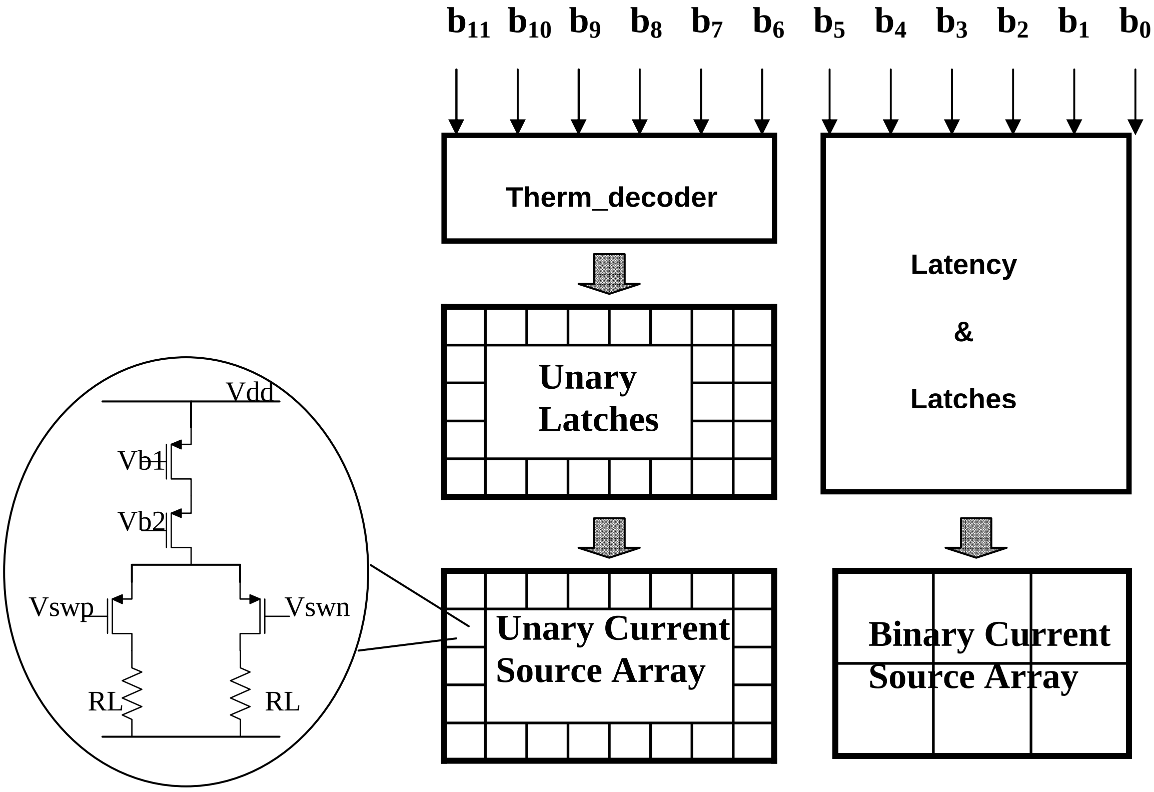

b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Therm_decoder
Latency
&
Latches
Unary Latches
Unary Current Source Array
Binary Current Source Array
Vdd
Vb1
Vb2
Vswp
Vswn
RL
RL

Figure 4. 13 Conceptual block diagram of a DAC

The conceptual block diagram is implemented by the proposed segmented architecture as shown in Fig. 4.13. The current source is implemented by a cascode structure.

The current generated by the current sources is switched to either side of the two differential output nodes by switch transistors Vswp and Vswn. These signals are synchronized by a latch in front of the switches. The steering signals are generated from the binary-to-thermometer decoder, for the unary latches out of the digital input word and a latency equalizer block for binary structure. This latency equalizer block ensures correct timing for the steering signals of the binary signals. One of the important architectural choices is how many bits are implemented using binary weighted current sources and how many using unary weighted sources. The thermometer architecture allows for minimum glitch energy and DNL at all code transitions because each LSB current source is turned on or off individually, but it requires a large number of switches and decoding logic, which greatly increases the die area and the digital decoding delay. The binary approach allows for the minimum number of switches and decoding logic because each input bit controls a binary weighted number of current cells, so no decoding is required. However, this results in all current cells switching at the major code transition, which maximizes the glitch energy and provides worst-case DNL.

The design approach taken here is to have the most thermometer code in the DAC architecture that does not increase the die area significantly over the fully binary approach. Thus, this design uses an architecture that is half thermometer and half binary because it results in an acceptable penalty in die area over the fully binary approach. The basic floor plan of the proposed architecture is shown in Fig. 4.13. The switches and the cascode transistors are placed in an array, which is separated from the latches array, to make the whole current source array area small, and the connection between switches and current sources, therefore, alleviate the gradient error effect and parasitic capacitance at the drain of current sources.

4.2.2.4 Switches

The dynamic performance of the current-steering DAC is highly dependent on the current switches. The switches are a simple differential pair whose tail current is switched completely to either of the two output branches. The maximum operation speed of the switch is ultimately limited by the process parameters. At high-signal frequencies the output spectrum is degraded due to glitches. The key issues to minimize the glitches are [23]

a. to minimize the capacitive coupling from the digital control signal to the analog output and to the current sources;

b. to avoid timing differences between switch controls;

c. to minimize the voltage variation in the common source node of the differential current switch during the switching; and

d. to minimize the stray capacitance from the current source output to the ground.

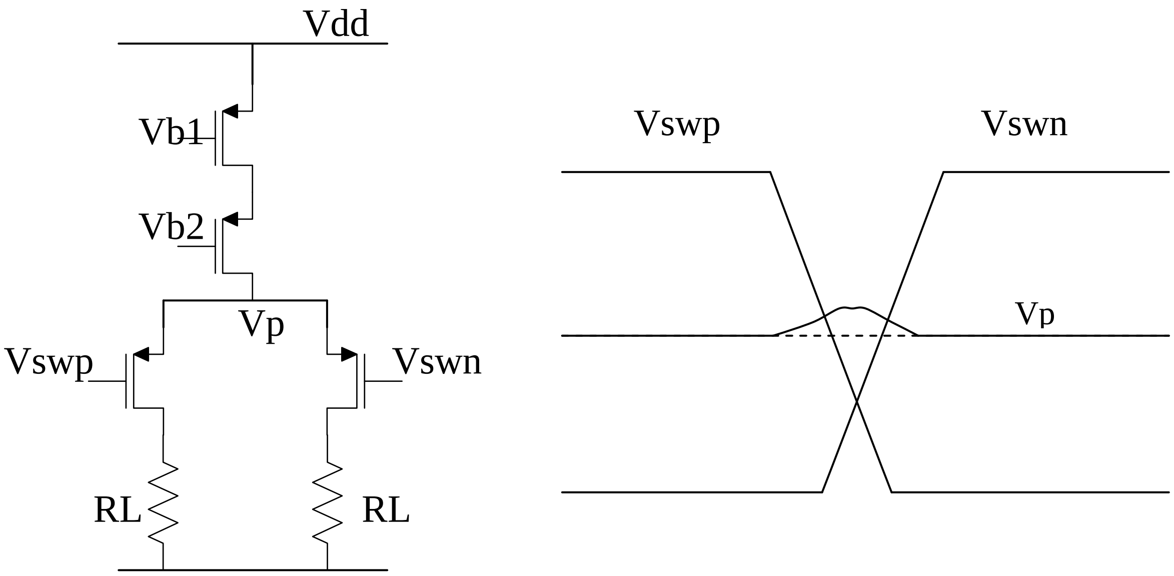


Figure 4. 14 Current source unit cell and the voltage variation at point P

The unsynchronized digital input is fed in from the left and the cascode current source and the current switch are shown in Fig. 4.14. The capacitive coupling to the analog output can be minimized by limiting the amplitudes of the control signals just high enough to switch the tail current completely to the desired output branch of the differential pair. In addition, the switch transistors are kept relatively small in order to avoid large parasitic capacitances. The digital input is synchronized with a latch. To ensure equal operation speed of different switches, the current densities in the switch transistors has to be the same, which is obtained by scaling the width of the transistors. The scaling is done only for the switches corresponding to the 6 MSBs to avoid impractically large transistor sizes. It is important to keep the voltage in the common source node of the differential pair as constant as possible during the switching. The voltage variation in this node causes the stray capacitance to be charged and discharged, which, in turn, slows down the settling of the output current. The voltage variation is minimized by overlapping the control signals in such a way that their across point lies slightly above the minimum voltage level. The DAC is organized as an array of PMOS current sources driving a 50Ω.

4.2.3 Simulation results

As discussed in section 4.1, in order to compensate for dynamic errors, it is better to calibrate as many bits as possible. But, the complexity of the look-up-table will increase dramatically with increased calibrated bits, while improvement of SFDR may not be proportional to the calibrated bit number. So a tradeoff exists between the number of calibration bits and the complexity. In this paper, calibration is done to the 6-bit MSB current array. The clock frequency f_{clk} is 200MHz. A sine waveform is used as the input signal. The dynamic glitch compensation pulse width is 0.8ns, while the dynamic settling compensation pulse width is 5ns.

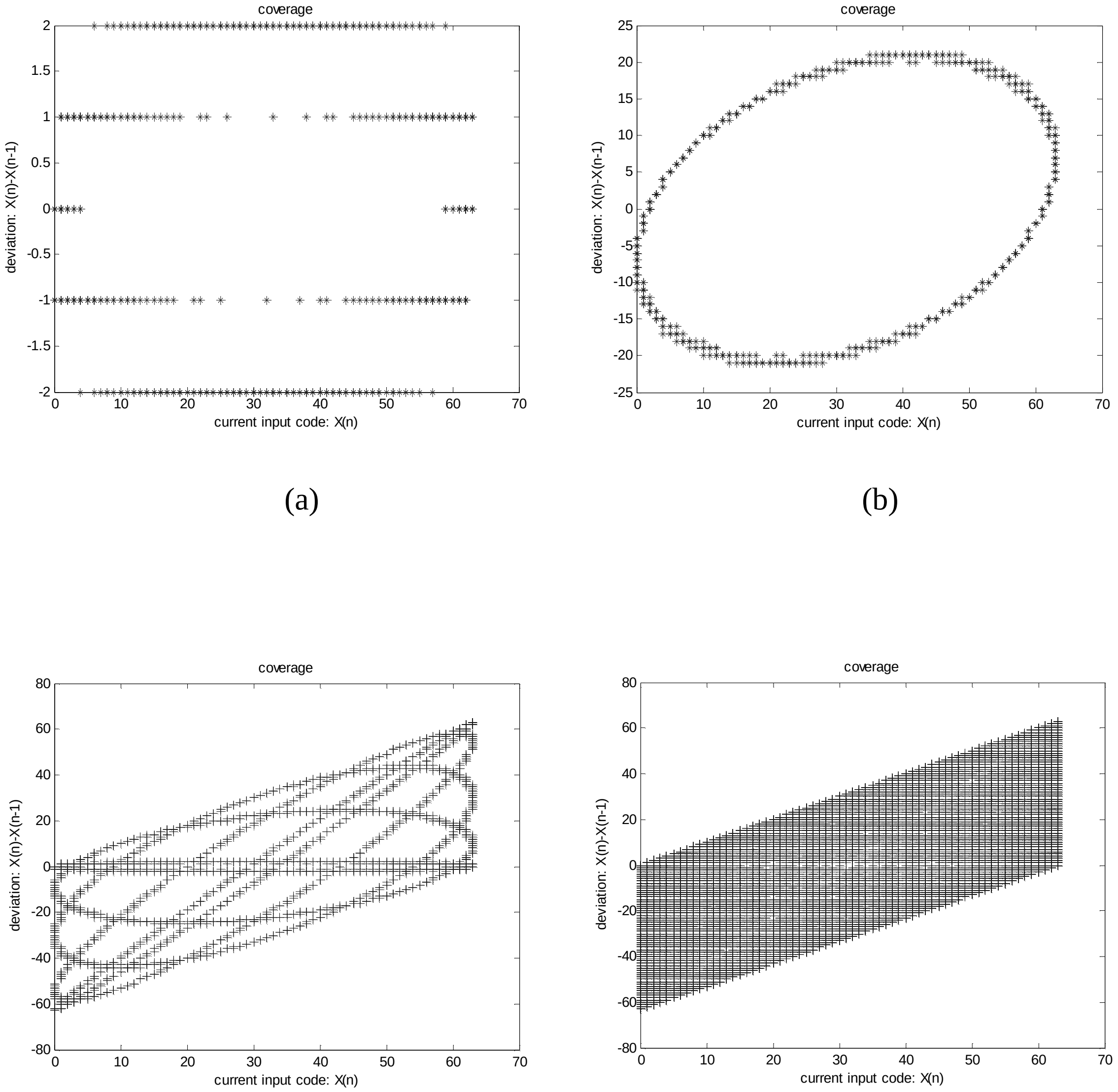
The results in the previous 15-bit DAC calibration showed that the SFDR improvement is mainly from the dynamic glitch calibration. Therefore, only dynamic glitch calibration is applied to the transistor-level implemented circuits.

Dynamic calibration is done for two cases: single input frequency error look-up-table calibration and combined error look-up-table calibration.

4.2.3.1 Error look-up table coverage

As we discussed in chapter 3, distortions in DACs are a function of the input code and its step-jumping from the previous input code. To calibrate distortion, a three-dimension error look-up table is built where one error value is stored for a given input code x_n and step-jumping x_{step}.

As for realistic DAC calibrations, a full-scale look-up-table should be used to calibrate any input frequency signal. To construct the full-scale error look-up table, DAC input signals are structured so that the DAC is excited over the entire DAC Nyquist bandwidth with significant distortion terms within the observation band to get small error look-up tables for every input frequency first. A full-scale look-up table can then be built by combining the small error look-up tables together. The calibration is applied to 6-bit MSB in this design. As a result, the look-up table is of a size of 64x127 (input code: 0~63, step jumping-63~63). There is no need to fill in the whole table because for a given input code the step-jumping is not from -63~63. For example, when the input code is x_n =8, the step-jumping is from -55~8. In practice, step-jumping is between –(63-x_n) ~ x_n which is half of the whole jumping range. So, only half of the error look-up table needs to be filled. A small table can be constructed at a single input frequency. By carefully choosing input frequencies to activate the full-scale input codes and step-jumping, one can then combine all the small tables together to obtain a full-scale table. Fig. 4.15 is an illustration of the small look-up table at different input frequencies and the combined full-scale table. For sampling frequency f_{clk} with the input signal frequency $f_{sig} = \frac{M}{512} \times f_{clk}$, the resulting look-up table is illustrated in Fig. 4.15(a), (b) and (c). The horizontal axis is input codes, while the vertical axis is the step-jumping. It can be seen from the plots that for a single input frequency, only a part of the table is covered.



(a) (b)

(c) (d)

Figure 4. 15 Table coverage for different input signal frequencies

(a) M=5 (b) M=55 (c) M=[5;65;125;195;245] (d) full scale

The higher the input signal frequency, the larger the step-jumping. Many small tables have to be built and put together to construct the full scale look-up table as shown in (d).

4.2.3.2 Single-input frequency error look-up table calibration

From the foregoing description in chapter 3, a single-input frequency error look-up-table E_{sn} (n=1,2,…,N) can be obtained for each input signal frequency. For a 6-bit MSB subDAC, the input code X_n varies between 0 and 63, while D_{step} is between -63 and 63. At a signal-input frequency, the corresponding small error table E_{sn} contains only a fraction of the full-scale table. In this calibration, the compensation pulse is generated from the corresponding small error look-up table for each input signal frequency. The dynamic performance of the DAC is shown in Fig. 4.17. From the plot, it can be seen that a significant increase in the SFDR performance, at least 20dB, can be noticed at high frequencies when dynamic glitch calibration is taken into account. The plots in Fig. 4.16 are the spectrum characteristics before and after calibration. There is obvious harmonic distortion in the plot before calibration. However, it can be seen from the plot after calibration that the distortion can be significantly suppressed to as low as the noise floor through dynamic calibration even at high frequency close to Nyquist rate (100MHz). The results show that the main dynamic distortion is contributed to by the nonlinearities among the transient region. SFDR can be dramatically improved by reducing input-codes and jumping-steps dependent nonlinearities in the transient region through dynamic glitch calibration.

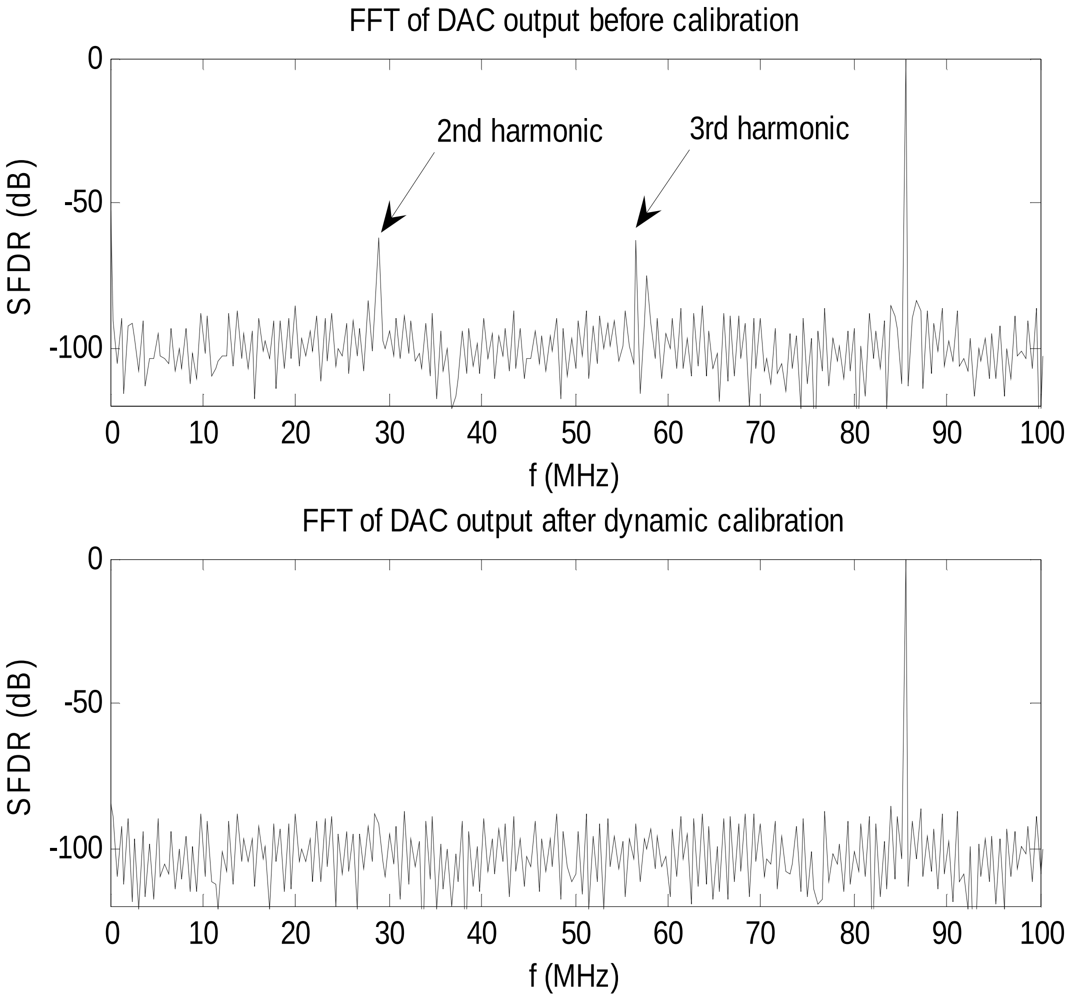


Figure 4. 16 FFT of DAC output before and after dynamic glitch calibration with single frequency error look-up table

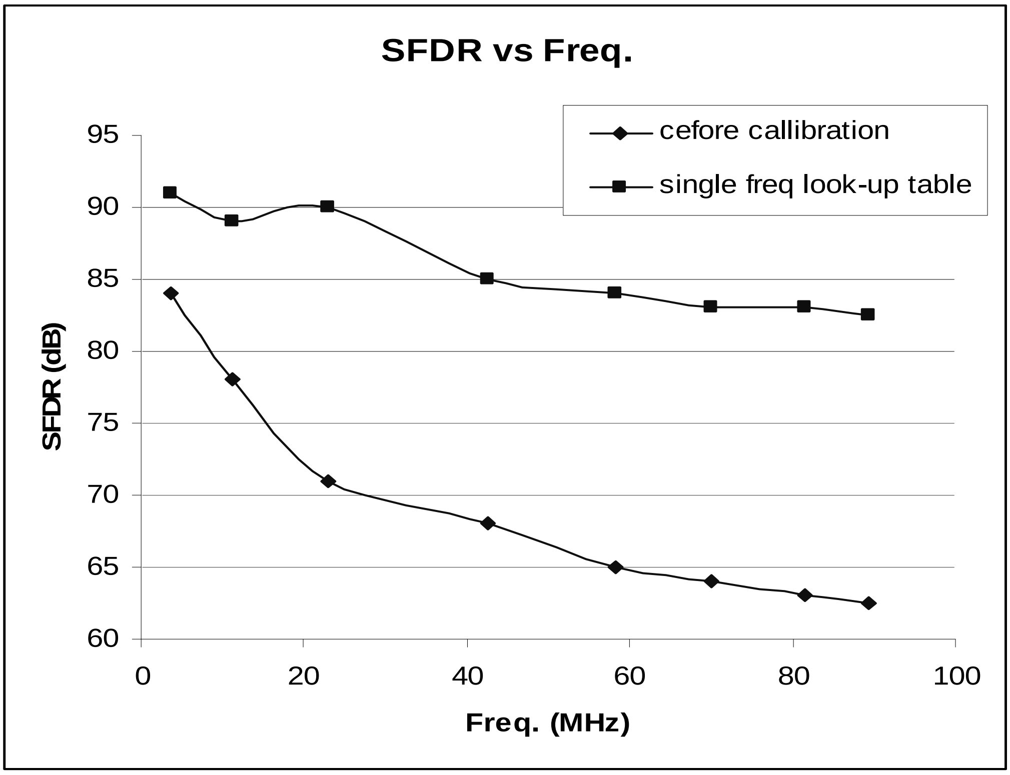


Figure 4. 17 SFDR vs. input signal frequency for signal frequency error look-up table calibration

4.2.3.3 Combined error look-up-table calibration

As for a realistic DAC calibration, the look-up-table should be used to calibrate any input frequency signal after it is built. The assumption of this approach is that the correction table is mainly the current-input code and the difference of the current and previous codes dependent. It means for an input code of A_n, the error will be given by E=f (X_n , X_{diff}), and E is frequency independent or nearly independent. This is easy to understand because no matter what frequency it is, the output current is obtained from certain current sources. For the same current input code, for example, X_n, the current sources that are used to generate the required current should be the same. Thus, no matter what the input signal frequency, if the jumping

X_{diff} is equal, the current used to calibrate the nonlinearity should be the same or very close for the same X_n. It is found to be true after comparing the overlapping error value among different small error look-up-tables. Fig. 4.18 is the error look-up-table of full scale input codes and jumping steps.

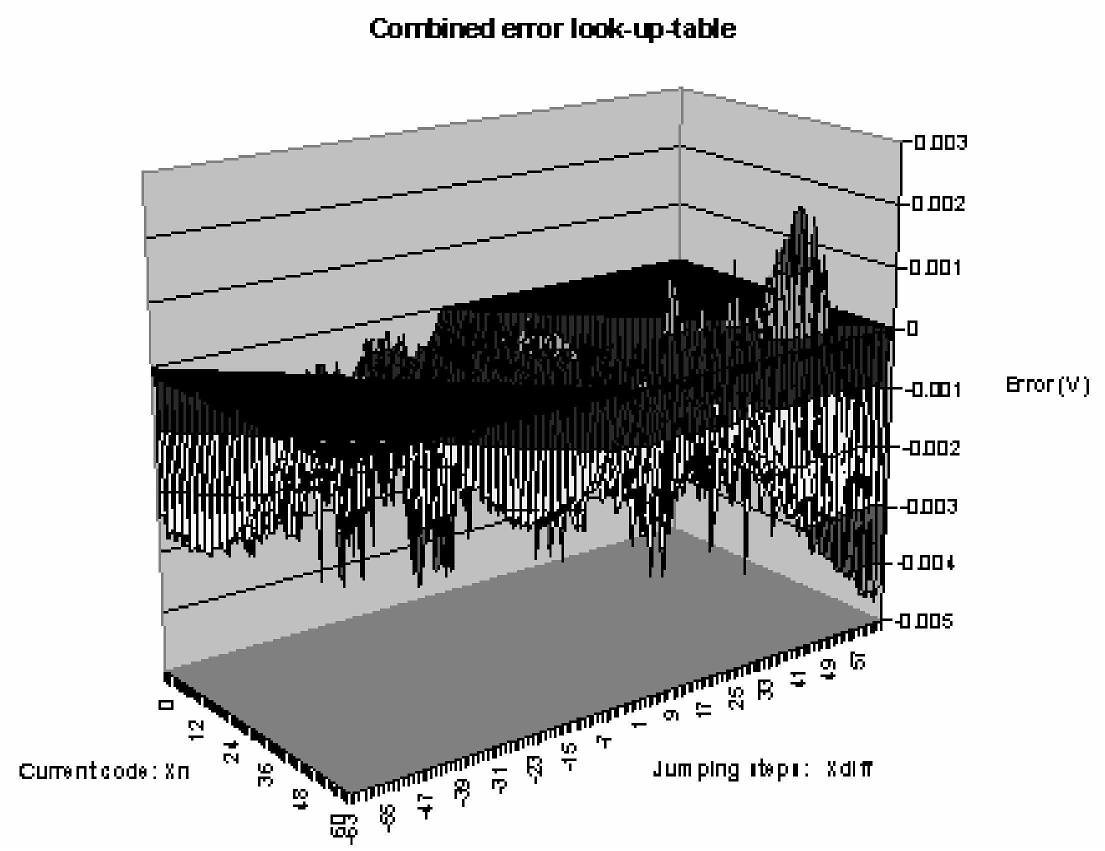


Figure 4. 18 Combined error look-up table

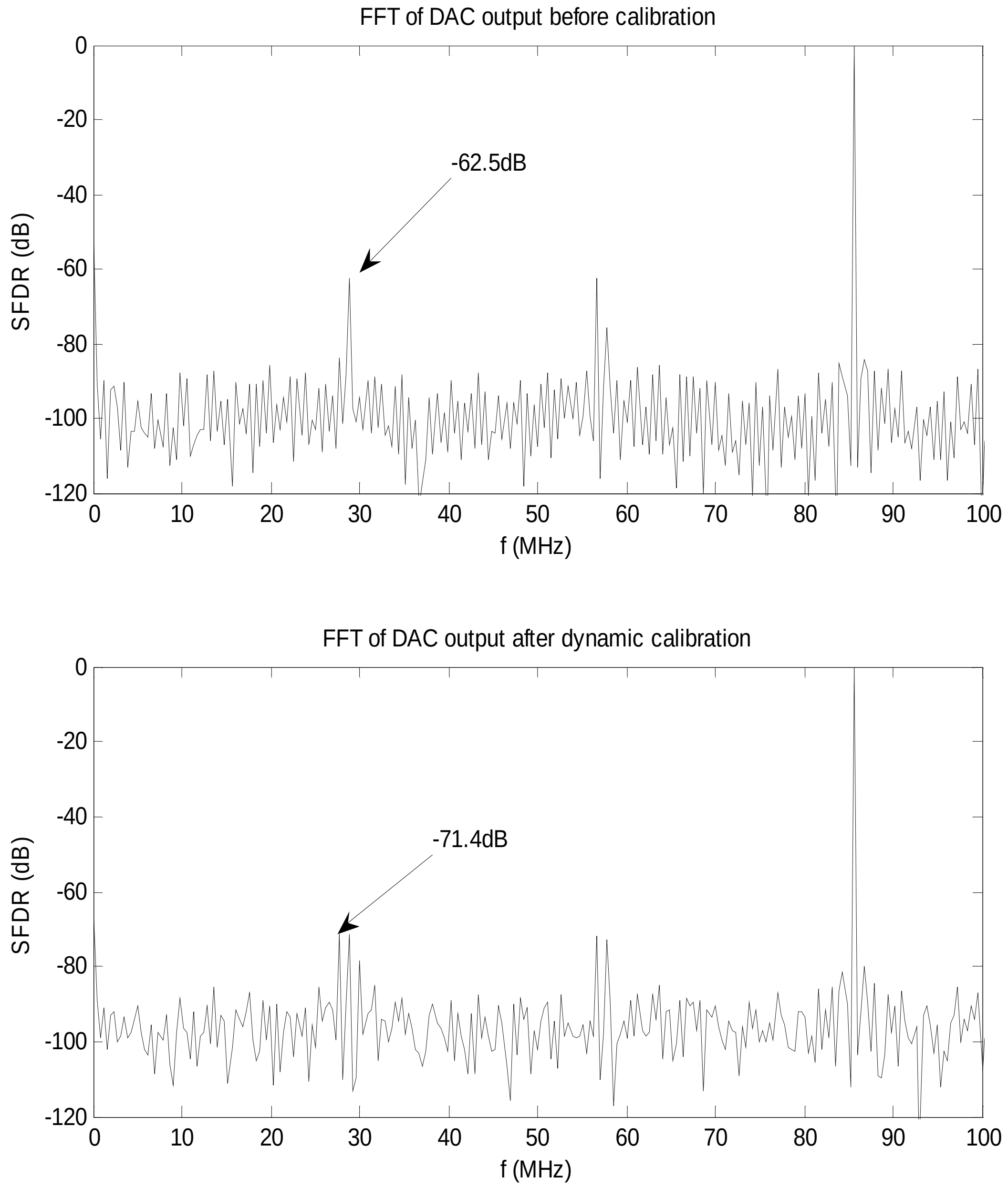


Figure 4. 19 FFT of DAC output before and after dynamic glitch calibration with full scale error look-up table

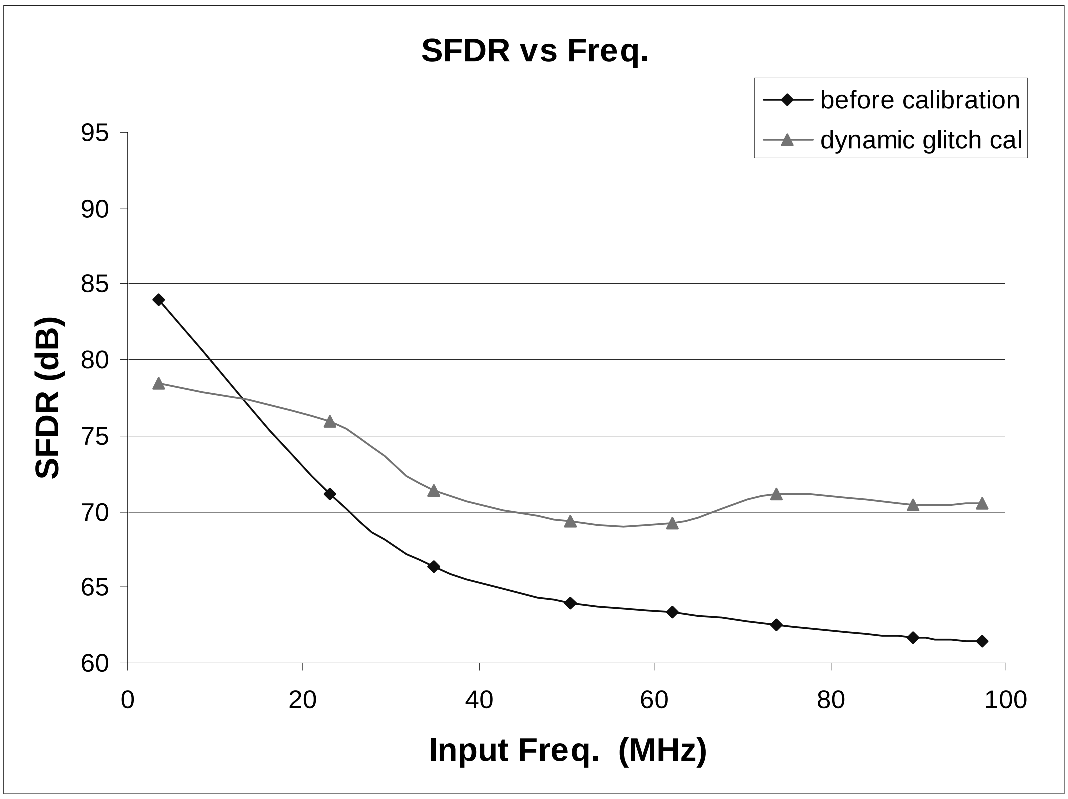


Figure 4. 20 SFDR vs input signal frequency before and after dynamic calibration

The plots in Fig. 4.19 are the spectrum characteristics before and after calibration. From the plot it can be clearly observed that the harmonic is suppressed from -62.5dB to -71.4dB. About 9dB improvement is achieved at near Nyquist rate.

Fig. 4.20 shows the DAC's dynamic performance before and after calibration. Although the SFDR may degrade a little at low frequency, the improvement becomes more significant with increasing input-signal frequency. More important is that the SFDR remains nearly constant until Nyquist rate, where the SFDR improvement is significant.

The improvement of SFDR from the combined error look-up-table is not as large as that from single-input frequency error look-up-tables. One reason may be that, in this approach, it is assumed that the error table should be frequency independent, but it is not completely unrelated to input signal frequency in reality.

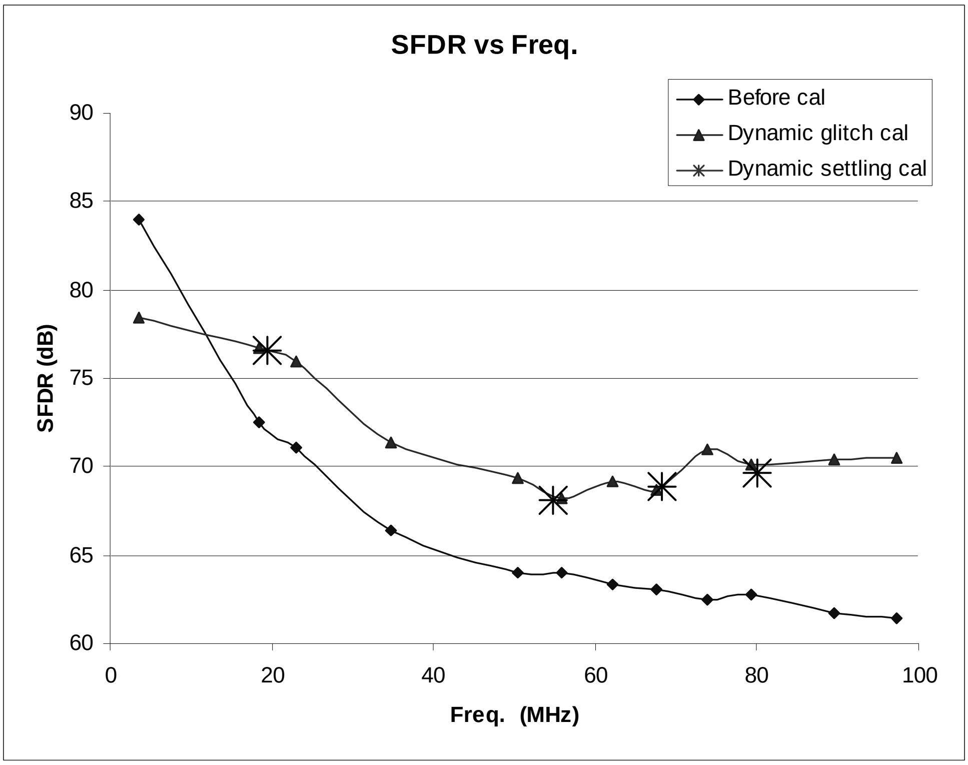


Figure 4. 21 SFDR vs input signal frequency before and after dynamic calibration

Frequencies plotted in Fig. 4.20 are all from those used to build the full scale error look-up table. To verify the robustness of this calibration, frequencies not used in table construction need to be under calibration to observe how the SFDR will change after dynamic calibration. Fig. 4.21 shows the DAC’s dynamic performance before and after calibration, where the “*” on the plot marks the tested frequencies not used in table construction. The plot shows that all the frequencies follow the trend of SFDR vs input signal frequencies although the SFDR improvement at the tested frequencies are 1~2 dB smaller than that from those frequencies used in error look-up-table construction. There may be some frequencies where the SFDR improvement is dramatically lower that the trend. However, a lot of frequencies are used to build the error look-up-table, and the calibration results from frequencies that are not used in the error look-up-table construction also follow the trend, the possibility of SFDR after dynamic calibration with significant deviation from the trend is quite small.

In above calibration, the full scale error look-up-table is built by averaging errors over whole Nyquist bandwidth. In this way, the full scale table can be used to do broadband calibration. However, if only a narrow frequencies band is of interest, the full scale table can be constructed in a different way. For example, if signals at low frequencies are critical, the errors obtained at low frequencies are weighted more in the error look-up-table construction. Fig 4.22 is the DAC's dynamic performance after calibration with full scale error look-up-table built by replacing error values of small jumping steps with the errors extracted at low frequencies. The plot shows that significant improvement is achieved at low frequencies although DAC's performance degrades moderately in the medium frequencies range.

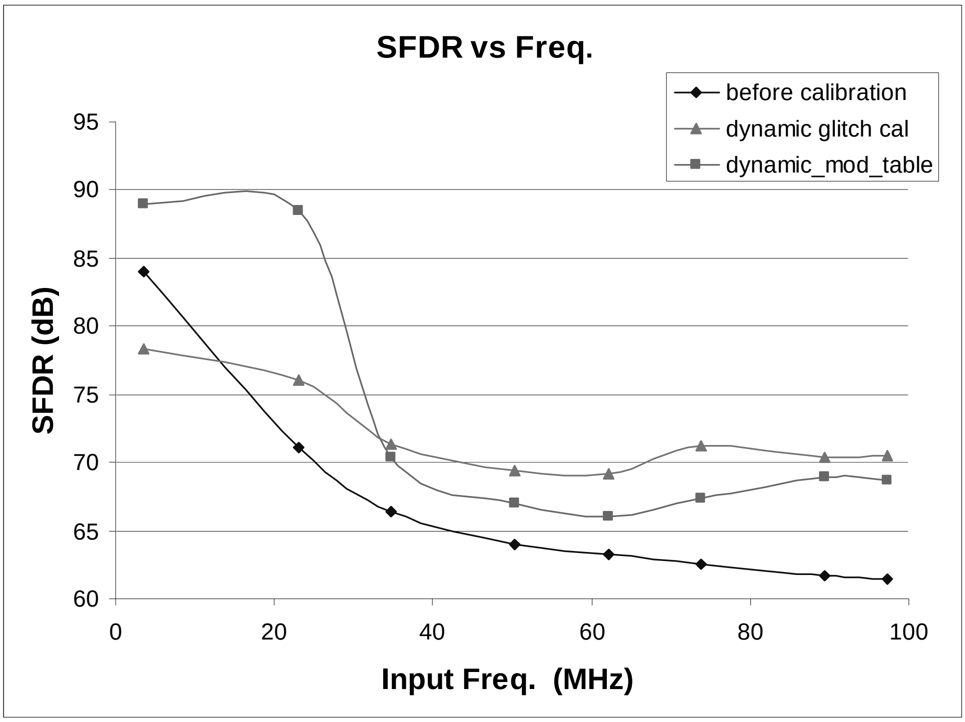


Figure 4. 22 SFDR vs input signal frequency with modified error look-up-table (low frequencies)

The same mechanism can be applied to high frequencies signal calibration. Fig. 4.23 is the plot of SFDR vs input signal frequencies after replacing the corresponding error values at the full scale error look-up-table with errors from high frequencies small tables. The results show that significant improvement is achieved at high frequencies although DAC's performance degrades moderately in the medium and low frequencies.

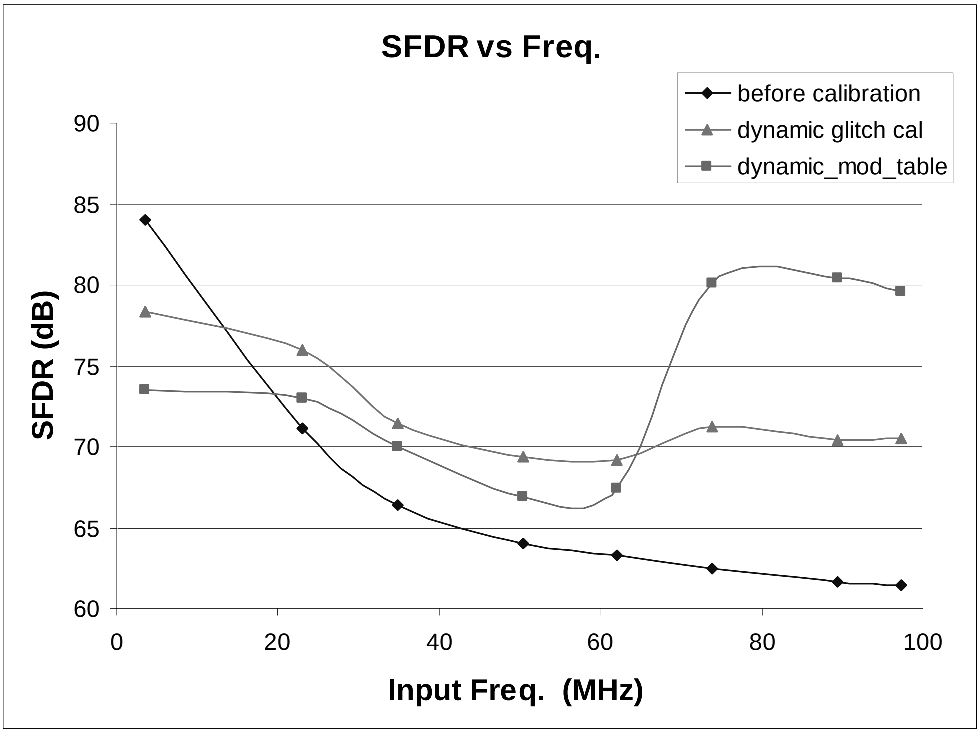


Figure 4. 23 SFDR vs input signal frequency with modified error look-up-table (high frequencies)

The simulation results show that more significant improvement of the DAC's dynamic performance can be achieved if the calibration is focused on a narrow frequencies range compared to broadband calibration.

1) SFDR vs. compensation pulse delay

In the above calibration, it is assumed that the compensation pulse is aligned with the raw DAC output. But, that may be not the case in the reality. A time interval between the raw DAC output and the calibration, Δt_d, may exist. This delay will have an impact on SFDR. The effect of different Δt_d to SFDR is shown in Fig. 4.24. In the simulation, the compensation pulse width is 800ps and the input signal frequency is about 73.8MHz. The magnitude of SFDR decreases from about 70dB to about 65dB when the delay width changes from 0 to 400ps. If the delay can be controlled under 200ps, the degradation of the SFDR is less than about 4%. The requirement of this control is not hard, so it is not a significant concern in the design. This result shows that this approach is robust.

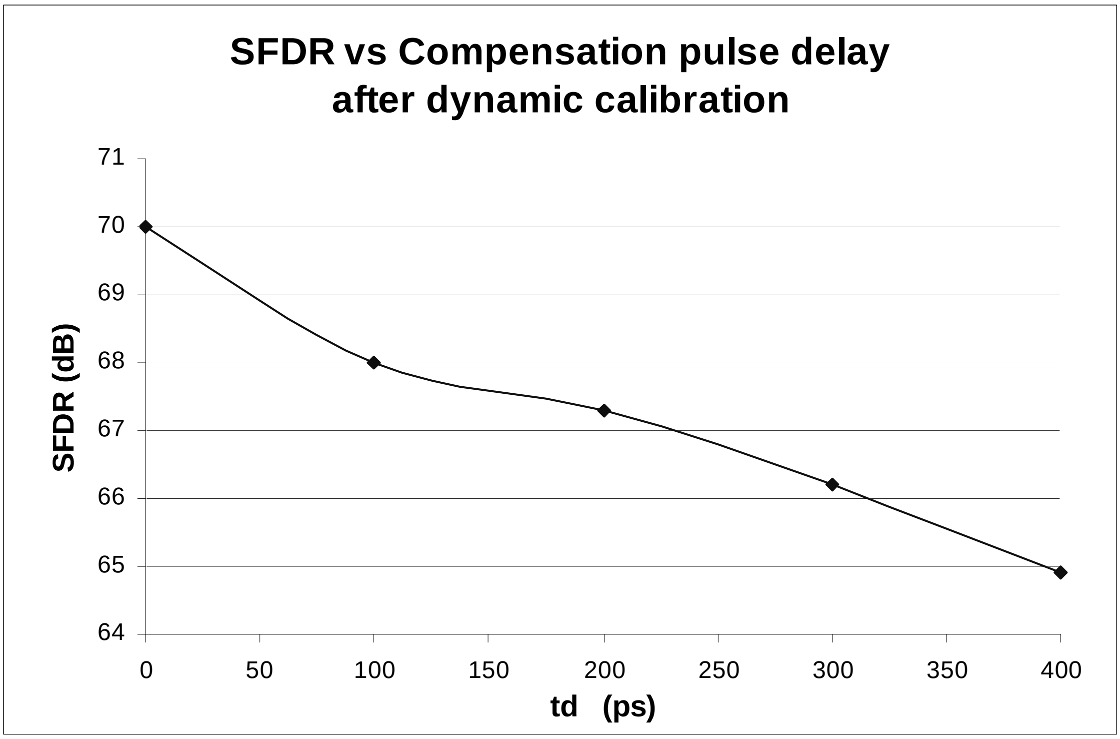
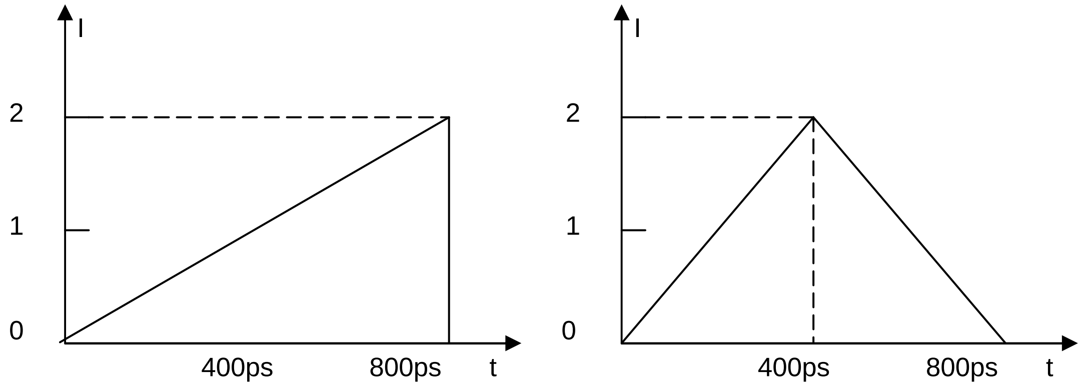


Figure 4. 24 SFDR vs. compensation pulse delay of DAC after dynamic calibration

2) SFDR vs. shape of compensation pulse

As in the previous discussion of 15-bit DAC calibration, not only the robustness of the approach but also implementation possibilities need to be considered. In the previous calibration, the pulses used in calibration are rectangle waveforms. In reality, it is a challenge to get rectangle waveform with width of half a nanosecond. However, it is easy to generate a triangle or near triangle waveform. If the SFDR improvement using triangle waveforms for compensation can achieve the same or just below that of rectangle waveforms, it will relax the realization of the calibration. Two kinds of triangle waveforms, (a) and (b) as shown in Fig. 4.25, are used to replace the rectangle waveforms. Both of the triangle waveforms have a width of 800ps. The height of the triangle waveforms is normalized to the height of the rectangle waveforms as 2 to achieve equal compensation waveform area. SFDR improvement for different compensation pulse shapes is shown in Fig. 4.26. The SFDR vs. input signal frequency curves of the dynamic calibration are close to each other, especially the calibration through rectangle waveforms and shape 2 triangle waveforms. The SFDR after the shape 1 triangle waveform calibration has just less than 1 dB deviation. The results show that the shapes of the triangle have little effect on the SFDR improvement given the same compensation pulse energy area.



(a) (b)

Figure 4. 25 Two calibration triangle waveforms

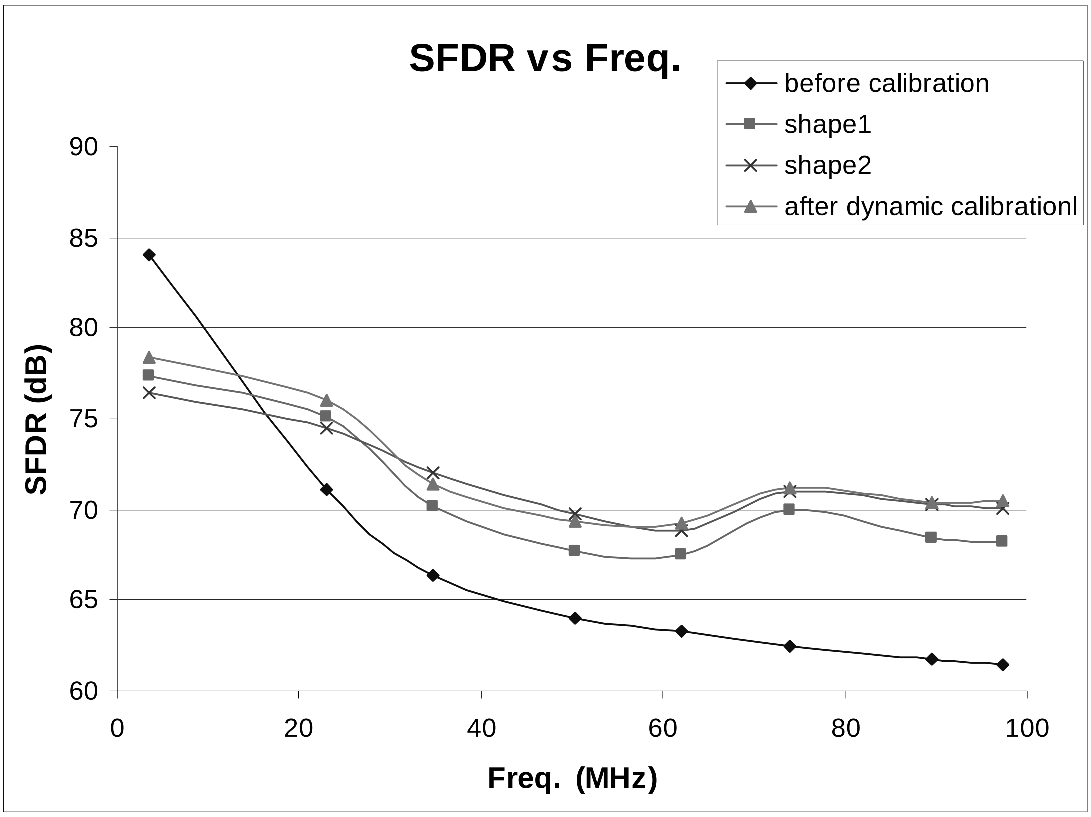


Figure 4. 26 SFDR vs. input-signal frequency for different compensation pulse shapes

4.2.4 Conclusions

The simulation results of the 12-bit transistor level current steering DAC using dynamic glitch calibration applied with both single-input frequency error look-up tables and a combined error look-up table showed that distortions can be efficiently reduced and the SFDR can be correspondingly significantly improved through this calibration scheme by applying narrow pulses to compensate for nonlinearities. The simulation also showed that this approach is robust. Given the same compensation pulse energy area, the pulse sharps have little effect to the DAC dynamic performance improvement. Therefore, this approach is feasible.

CHAPTER 5. EXPERIMENTNAL RESULTS

The behavior- and transistor-level calibration procedures were presented in chapter 4. In this chapter, we will present and discuss the experimental results of the DAC dynamic calibration. First is an introduction to the experiment measurement procedures and principles. Following that is the DAC experiment testing setup. Finally, the measurement results and the discussion of the results are given.

5.1 Introduction

The goal of the calibration of a DAC is to find an error look-up table which drives the CALDAC to generate an extra current and add it to the raw DAC to produce a spectrally-pure desired output waveform. In the previous chapter, error information was obtained by removing the fundamental components from the spectrum output and applying IFFT to the residual frequency domain series. It is easy to realize the procedure in simulation. However, there are difficulties in determining error values in experimental measurement. First, output of the DAC is analog, so error data of a DAC needs to be quantized so that it may be manipulated with a computer. To obtain the error data, a possible scheme is to sample the output of the DAC using a higher-performance (high resolution, same speed) ADC. However, an ADC with equal or better performance to the DAC is generally not available. In practice, ADC errors usually dominate the DAC errors for devices at the same sample frequencies. A system with a ADC with higher performance that operates at slower clock rate is a possible solution [29]. In this approach, the system clocks the ADC at the DAC rate divided by a ratio R. A low pass filter is added after the output of the DAC to remove frequencies outside the Nyquist band of the ADC. This allows the user to implement a slower ADC with linearity superior to the DAC. This ADC may measure DAC output errors that appear at frequencies within the filter pass band. In order to perform an accurate calibration, the effects of the calibration system on the system output that are not related to DAC error, such as the phase and amplitude changes introduced by the low pass filter and the gain error and delay caused by the ADC, must be removed. Amplitude is not an issue, but the phase reference between the DAC input samples and the measured samples from the ADC used to calibrate the DAC must be determined. The coherent measurement system requires substantial time and effort to determine the phase references. This limits the use of this approach.

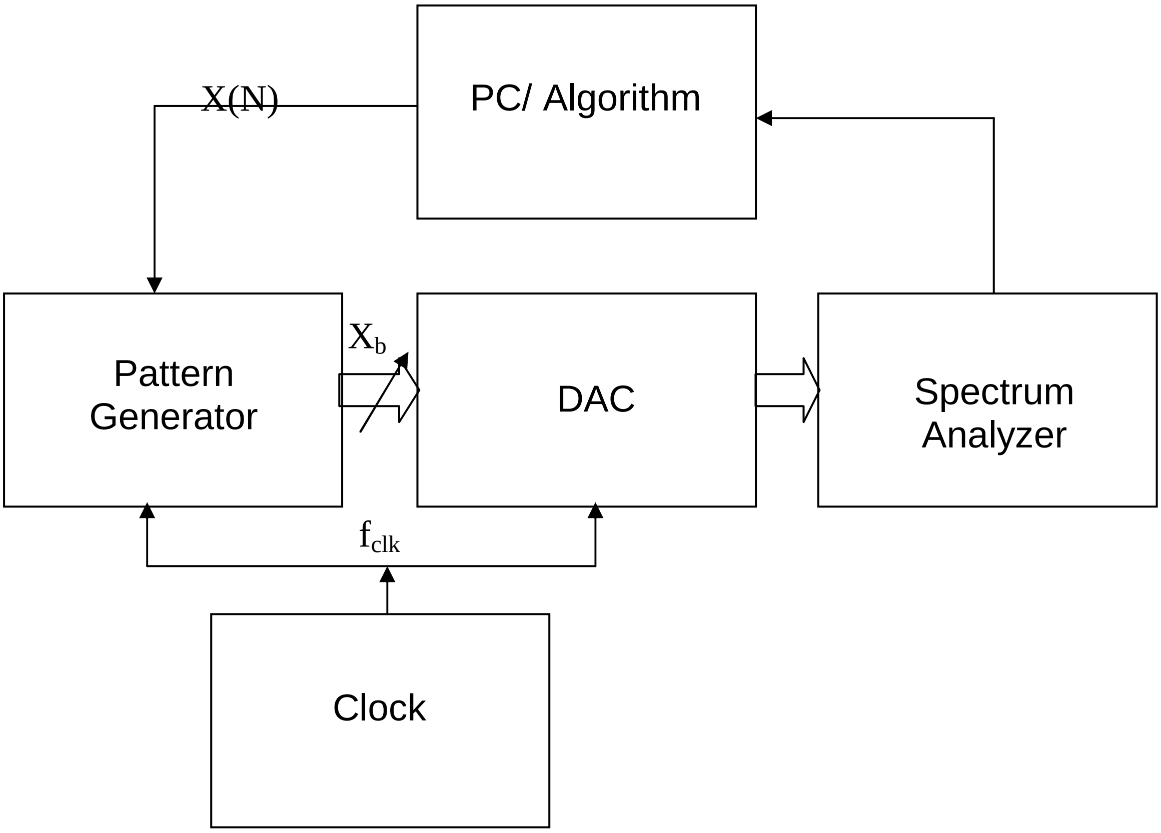


Figure 5. 1 DAC test setting

Another approach [30] to determine DAC error without knowledge of timing has been introduced. The procedure is to measure the power of the DAC output at every input frequency to extract the error waveform by varying the input signals. Then, the error waveform is used to modify the input sequence to obtain the desired output. This is a precode approach which is not the same as ours, but we can draw inspiration from its idea. A measurement setup is illustrated in Fig. 5.1. At first, a program is used to generate a length N periodic sequence with period number M and deliver the sequence to the signal generator. The pattern source then cycles through the samples, repeating the sequence every T=N/f_{clk} second and the DAD translates these samples to an analog signal. This analog signal containing power at input frequencies and the distortion power at other frequencies is measured by a spectrum analyzer. Based on the measured values, the error is extracted. The sequence is then modified according to the errors and the process is repeated until the error is removed to a tolerated level.

5.1.1 The error determination procedure

The sequence sent to pattern generator is a periodic signal with length N. The discrete-time Fourier series (DTFS) is:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \\ x(n) &= \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi kn}{N}} \end{aligned} \tag{5.1}$$

The spectrum of x(n) has N distinct bins for k=0,1,2,…,N-1. The values of x(k) represents the phase and magnitude of each spectral component. The output of an ideal DAC would contain energy only at those frequencies related to N bins at frequencies $f = \frac{kf_{clk}}{N}$. To measure error, a DAC is driven with a sequence that only has energy in known bin locations.

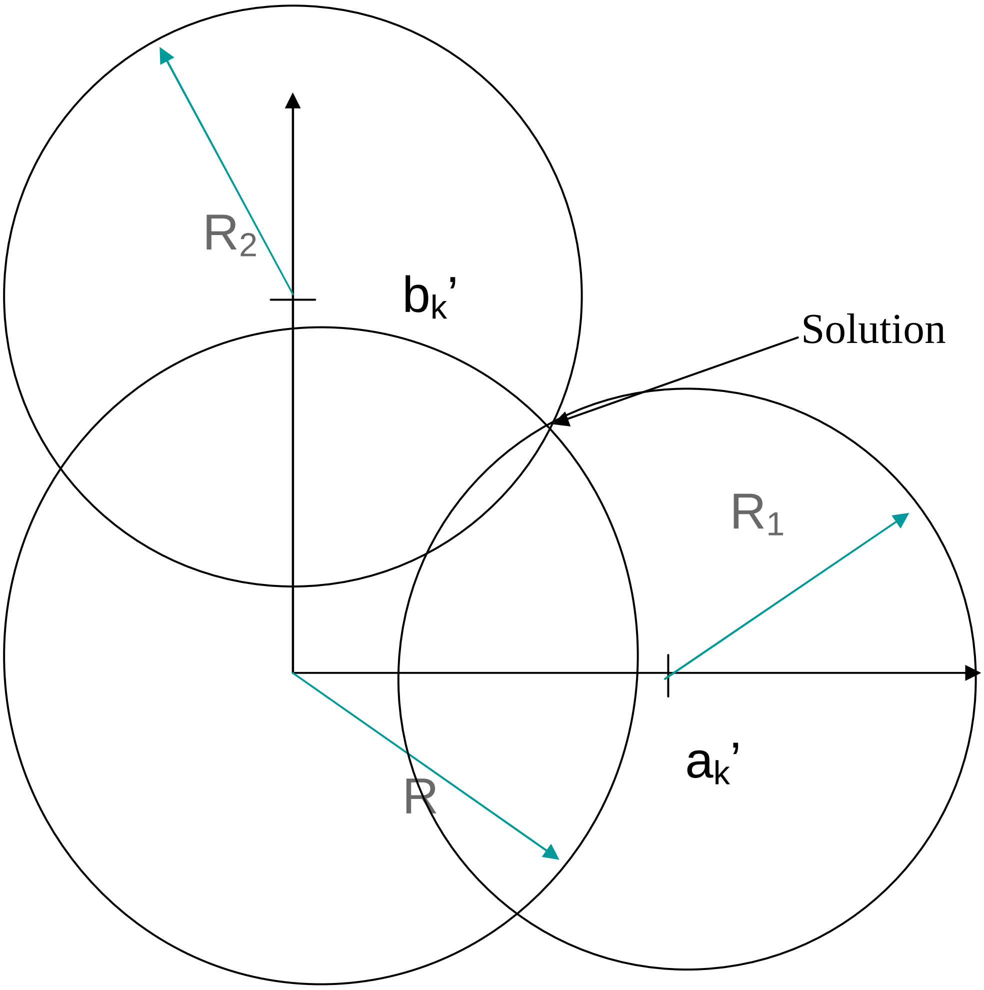


Figure 5. 2 a$_k$ and b$_k$ determination through three power spectrum measures

Any energy appearing in bins other than those predetermined would be considered error energy. By measuring error energy in these bins, an error sequence is determined. The error waveform experiences the same devices as the signal, and, therefore it should have the same expression with x(n):

$$\begin{aligned} e(n) &= \frac{1}{N}\sum_{k=0}^{N-1} C_k e^{j\frac{2\pi kn}{N}} \\ &= \frac{1}{N}[\frac{a_0}{2} + \sum_{k=1}^{N/2-1} (a_k \cos(\frac{2\pi kn}{N}) + b_k \sin(\frac{2\pi kn}{N}))] \end{aligned} \tag{5.2}$$

where $C_k = \frac{a_k - ib_k}{2}$ $\quad | C_k |^2 = \frac{a_k{}^2 + b_k{}^2}{4}$

In order to determine the error sequence, measurements of a_k and b_k must be made. A power spectrum analyzer provides measurements of the signal power P_k at the k_{th} bin. The measured power component is:

$$P_k = \gamma(\frac{a_k{}^2 + b_k{}^2}{4}) \tag{5.3}$$

where γ is a gain constant between the samples and the analog power spectrum measurement. To get the input error waveform, a_k and b_k need to be known. However, the power measured by spectrum analyzer is proportional to a_k^2+ b_k^2, not a_k and b_k. a_k^2+ b_k^2 defines a circle on which the value a_k and b_k must lie. From the measurement, it is only known that the values of a_k and b_k lie on the circle with radius:

$$R = | C_k |^2 = (\frac{a_k{}^2 + b_k{}^2}{4}) = \frac{P_0}{\gamma} \tag{5.5}$$

But their location can’t be determined yet. Since there are two variable, two more measurements are needed to solve the values of a_k and b_k.

Modifying the DAC input sequence by adding a cosine with amplitude $a_k^{'}$ at that bin results in a change of the measured power and, thus, results in a new circle with radius R_1:

$$R_1 = (\frac{(a_k - a_k')^2 + b_k{}^2}{4}) = \frac{P_1}{\gamma} \tag{5.6}$$

By adding this second circle, the a_k value has been identified by the intersection of the two circles while the b_k has only two possible values. The value of b_k can be found by adding a sine with amplitude $b_k^{'}$. The third circle has equation given by

$$R_2 = \left(\frac{a_k{}^2 + (b_k - b_k')^2}{4}\right) = \frac{P_2}{\gamma} \tag{5.7}$$

From the equations (5.5), (5.6), and (5.7), the values of a_k and b_k can be determined:

$$a_k = -\frac{4R_1{}^2 - 4R^2 - a_k'^2}{2a_k'}$$

$$b_k = -\frac{4R_2{}^2 - 4R^2 - b_k'^2}{2b_k'} \tag{5.8}$$

Then the k_{th} harmonic component of the error sequence is identified. The same procedure can be applied to other harmonics.

The procedure can be summarized as follows:

1. Generate a signal with known amplitude and input to the DAC, and then measure the output to determine the scale factor γ.
2. Determine bins $\vec{x} = \{x_1, x_2, x_3, \cdots; x_m\}$ (m: desired signal number) and frequencies f_x on which the desired signal energy is located. For sampling frequency f_{clk}, period length N, if the location of the desired signal bin is located at M, then the signal frequency $f_{sig} = \frac{M}{N} \times f_{clk}$.
3. Quantize the desired waveform into a sequence working as input signal to the DAC.
4. Measure the DAC output power spectrum (P_0) for $f = \frac{kf_{clk}}{N}$, (k=0,1,…,N/2-1). In practice, the dominant distortion is from the 2nd and 3rd harmonics; as a result, only bins at which the 2nd and 3rd harmonics locate need to be measured.
5. Modify the original sequence by adding $a_k' \sin(\frac{2\pi kn}{Ns})$ (k is the location bin of 2nd and 3rd harmonics within Nyquist band).
6. Measure the power spectrum (P_1) for the desired signal: 2nd and 3rd harmonics within the Nyquist band.
7. Modify the original sequence by adding $b_k' \cos(\frac{2\pi kn}{Ns})$ (k is the location bin of 2nd and 3rd harmonics within Nyquist band).

8. Measure the power spectrum (P_2) for the desired signal: 2nd and 3rd harmonics within the Nyquist band.

9. Using $a_k = -\frac{4R_1^{\,2} - 4R^2 - a_k'^2}{2a_k'}$, $b_k = -\frac{4R_2^{\,2} - 4R^2 - b_k'^2}{2b_k'}$, where $R = \frac{P_0}{\gamma}, R_1 = \frac{P_1}{\gamma}$ and $R_2 = \frac{P_2}{\gamma}$, to determine the values of a_k and b_k. Then the calculated error is $e(n) = (a_{k2}\sin(\frac{2\pi nk_2}{N}) + b_{k2}\sin(\frac{2\pi nk_2}{N})) + (a_{k3}\sin(\frac{2\pi nk_3}{N}) + b_{k3}\sin(\frac{2\pi nk_3}{N}))$, where k_2 and k_3 are the bin location of 2nd and 3rd harmonics in the first Nyquist rate. A desired output can be obtained by subtracting the error wave from the original signal.

5.2 Experimental setup

According to the procedure description in section 5.1, we will evaluate this procedure and use it to get the error information on experimental DAC.

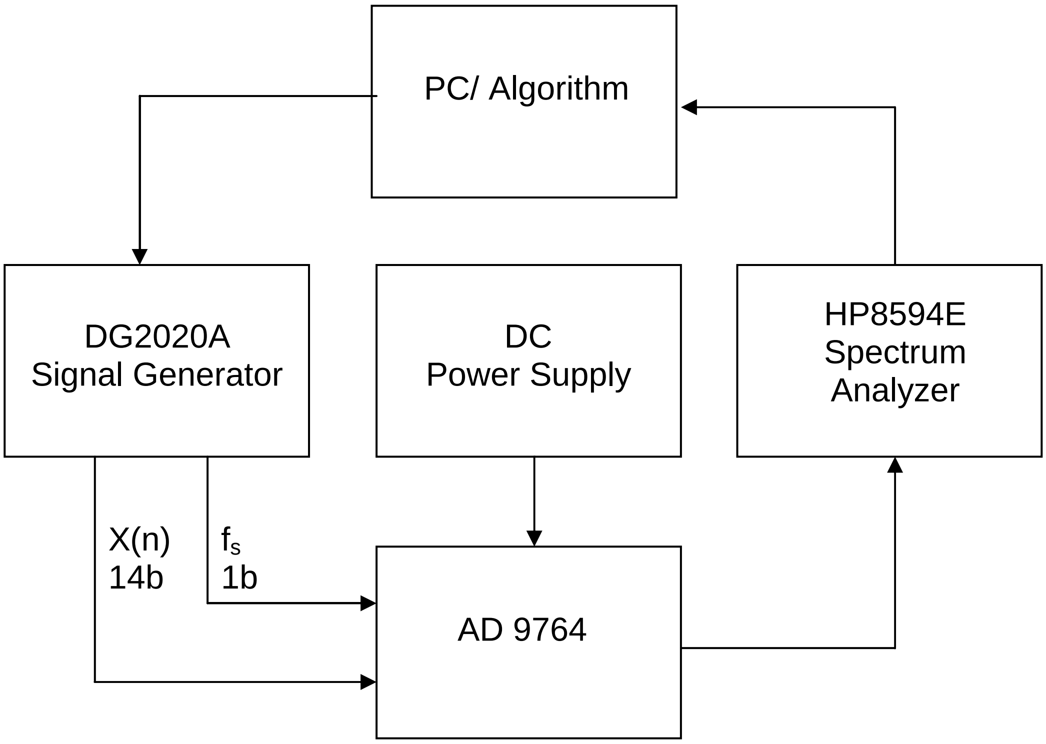


Figure 5. 3 DAC experimental measurement setup

The procedure was verified on an Analog Devices, Inc. 14-bit 125 MSPS TxDAC (AD9764). The experimental setting is shown in Fig. 5.3. To get precise values of a_k and b_k, it is very important to make accurate power measurements from the spectrum analyzer. In the measurement, it is critical to make sure that the input signal magnitude into the spectrum analyzer is not beyond the input range of the device or distortion will occur which will, causing inaccurate results. In this experimental measurement, a 20dB attenuate is applied to the input signal when it enters the spectrum analyzer. To obtain an accurate measurement result, the span and bandwidth must also be considered. The center frequency of the analyzer is set to the desired frequency. The span is set to 1.5KHz and the resolution bandwidth is set to 100Hz.

The following are the measurement equipment:

Devices	Model	Specifications
DAC	AD9764	Current steering, 14 bits, 125MSPS
Signal Generator	DG2020A	Data rate : 200 MHz, Output channels: 12,24,36
DC Power Supply		Vdd=3V, Vss=0V
Spectrum Analyzer	HP8594E	Dynamic range: +30dBm~-127dBm Frequency range: 9kHz~2.9GHz Input impedance: 50 Ohms

The input sequence length N is set as 4096, the sampling frequency f_s = 50MHz.

5.3 Experimental results

The calibration was implemented for two cases: calibration through precode and calibration through error look-up table.

5.3.1 Precode

In this case, a sequence x(n) at each frequency was generated to drive the DAC. The 2nd and 3rd harmonics parameters a_2, b_2, $a_{3,}$ and b_3 were obtained through the procedure described in section 5.1. Therefore, the error waveform e(t) for this input frequency was known. Subtracting the error waveform from the original signal waveform, a desired signal x(t)' after calibration was obtained.

$$X(t)' = X(t) - e(t) \tag{5.8}$$

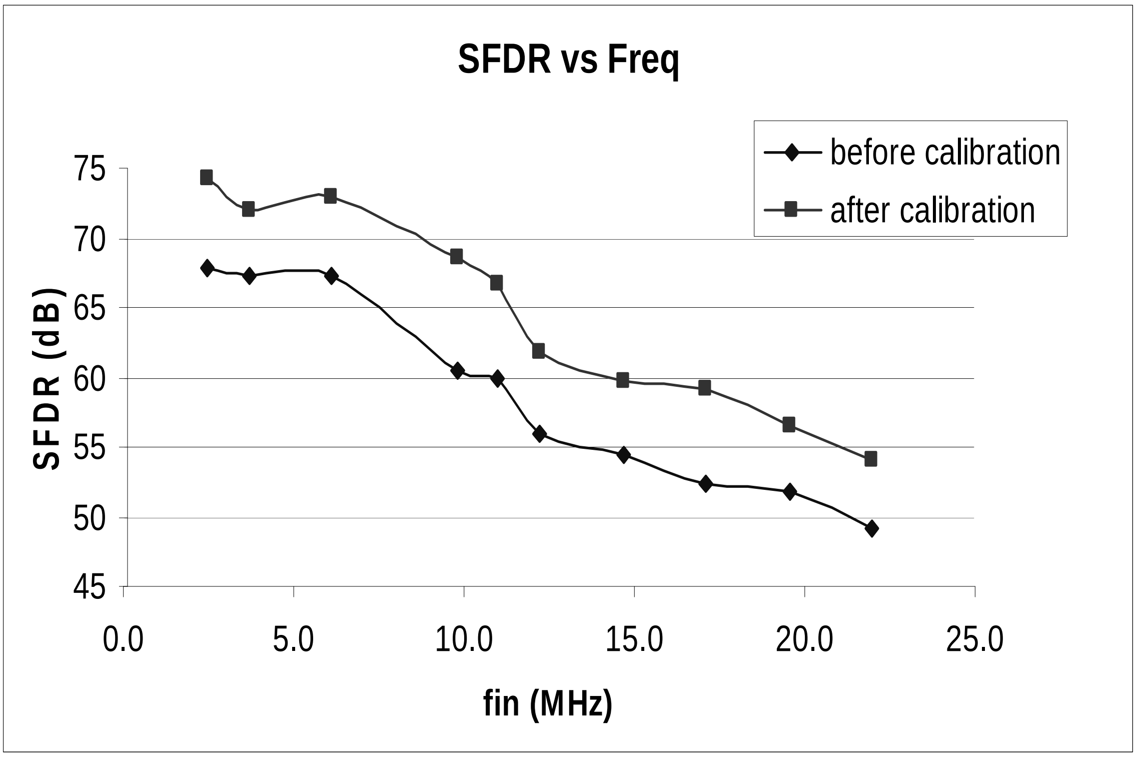


Figure 5. 4 SFDR vs. input signal frequencies for precode calibration

After sampling and quantizing the desired signal, a new sequence x(n)' is structured and used to drive the DAC. The same procedure is applied to signals at different input frequencies. Both x(n) and x(n)' are measured with spectrum analyzer, the SFDR vs input signal frequencies is plotted as Fig. 5.4. From the figure, it can be seen that around 5 dB improvement is achieved through the precode calibration.

5.3.2 Look-up table calibration

At the precode calibration, the desired waveform was obtained by subtracting the error waveform from the original signal. In practice, it was impossible to compensate the error for variable input frequency signals by subtracting the harmonics. A table was needed to store

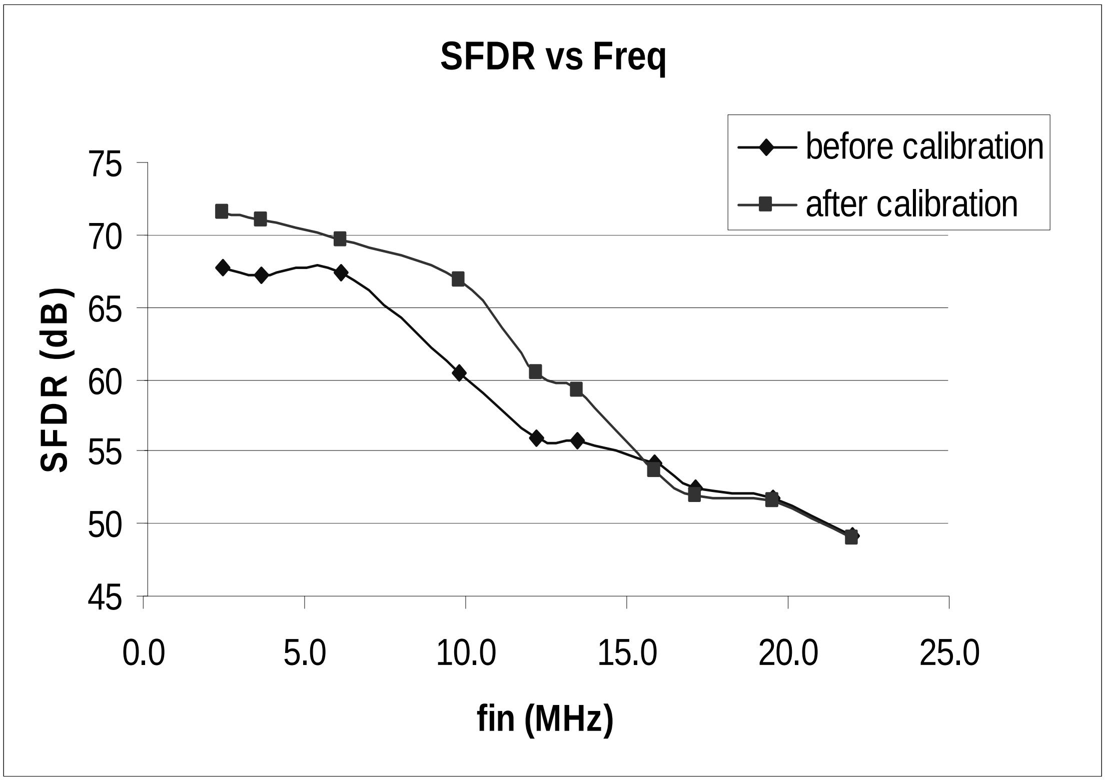


Figure 5. 5 SFDR vs. input signal frequencies for error look-up table calibration

the error values for different input codes and step jumping heights. The error look-up table was built by averaging the error values for the same input code and step-jump height. The calibration results are shown as Fig. 5.5. It can be seen from the plot that there is about 5 dB improvement of SFDR at low frequency, and the improvement decreases with the increasing of the input frequencies. At high frequency, the improvement is negligible .

5.4 Discussion

The results in section 5.3 show that the DAC's dynamic performance can be much improved through the precode method. It also shows that SFDR improvement can be achieved at low frequency through an error look-up table although the improvement is much less at high frequency. What is the cause of the improvement degradation with the increasing of the input signal frequency? Since the calibration is implemented through error waveform, it is worthy to observe the error waveform and look-up table.

Set the sequence length N=4096, where the input signal period number during the sequence is M, the clock frequency of the signal generator is f_s=50MHz because it takes length of two data to form a clock period, and the actual output frequency of the sequence from the data generator is:

$$f_{sig} = \frac{M \times f_{clk}}{N/2} = \frac{M}{2048} \times 50MHz \qquad (5.9)$$

Table 5.1 contains the error values for a given input code x_n and jumping step height x_{step} , where $x_{step} = x_{n}- x_{n-1}$. Observing the error values, we can notice that the error values are nearly the same at small M (low-input frequency). When the M is larger than a certain number, the error values will deviate and converge to two different values. Moreover, the deviation becomes more obvious when M grows. The two values become nearly symmetric around zero when the input frequency reaches Nyquist rate. The method applied to deal with the error values for the same (x_x, x_{step}) is to average these values to get a effective value $e_{avg}(n)$ used for compensation. So, $e_{avg}(n)$ is the mean of the group of error values for (x_x, x_{step}). At low frequency, all error values converge to one value; therefore, their mean is close to the real errors. With the increasing of the input frequency, the errors deviate to two different values with nearly the same magnitude but opposite signs. As a result, their means come to zero. This means that nearly no compensation is applied to the raw DAC, which results in no improvement in SFDR when the input signal frequency is close to Nyquist rate.

Observing the error values and M more carefully, we find that the error values deviate when M is lager than a certain value. Actually, the phenomenon occurs when the 3rd harmonic bin is beyond the Nyquist band. This can be explained easily. In section 5.1, we derived the error waveform expression:

$$e(n) = a_{k2}\sin(2\omega_0 t) + b_{k2}\cos(2\omega_0 t) + a_{k3}\sin(3\omega_0 t) + b_{k3}\cos(3\omega_0 t) \qquad (5.10)$$

This expression is valid when the 3rd harmonic bin is located within the Nyquist band.

In the calibration, we only compensate for harmonics below the Nyquist rate. If the harmonic bins are outside of the Nyquist band, the harmonics bins reflected into Nyquist band are of consideration. For example, for N=2048, M=401, the 3rd harmonic bin is 1204, which is outside the Nyquist band (1024). So the reflected bin located at (2048-3x401)=846 will be chosen to compensate for 3rd harmonic distortion at this input frequency signal.

For input signal frequency f_0, the actual error wave is:

$$e(n) = a_{k2}\sin(2\pi f_2 t) + b_{k2}\cos(2\pi f_2 t) + a_{k3}\sin(2\pi f_3 t) + b_{k3}\cos(2\pi f_3 t) \qquad (5.11)$$

And

$$\begin{cases} f_2 = 2f_0 \\ f_3 = 3f_0 & when \quad f_0 < \dfrac{f_s}{6} \\ \\ f_2 = 2f_0 \\ f_3 = | f_s - 3f_0 | & when \quad \dfrac{f_s}{6} < f_0 < \dfrac{f_s}{4} \\ \\ f_2 = | f_s - 3f_0 | \\ f_3 = | f_s - 3f_0 | & when \quad \dfrac{f_s}{4} < f_0 < \dfrac{f_s}{2} \end{cases} \qquad (5.12)$$

where f_s is the clock frequency.

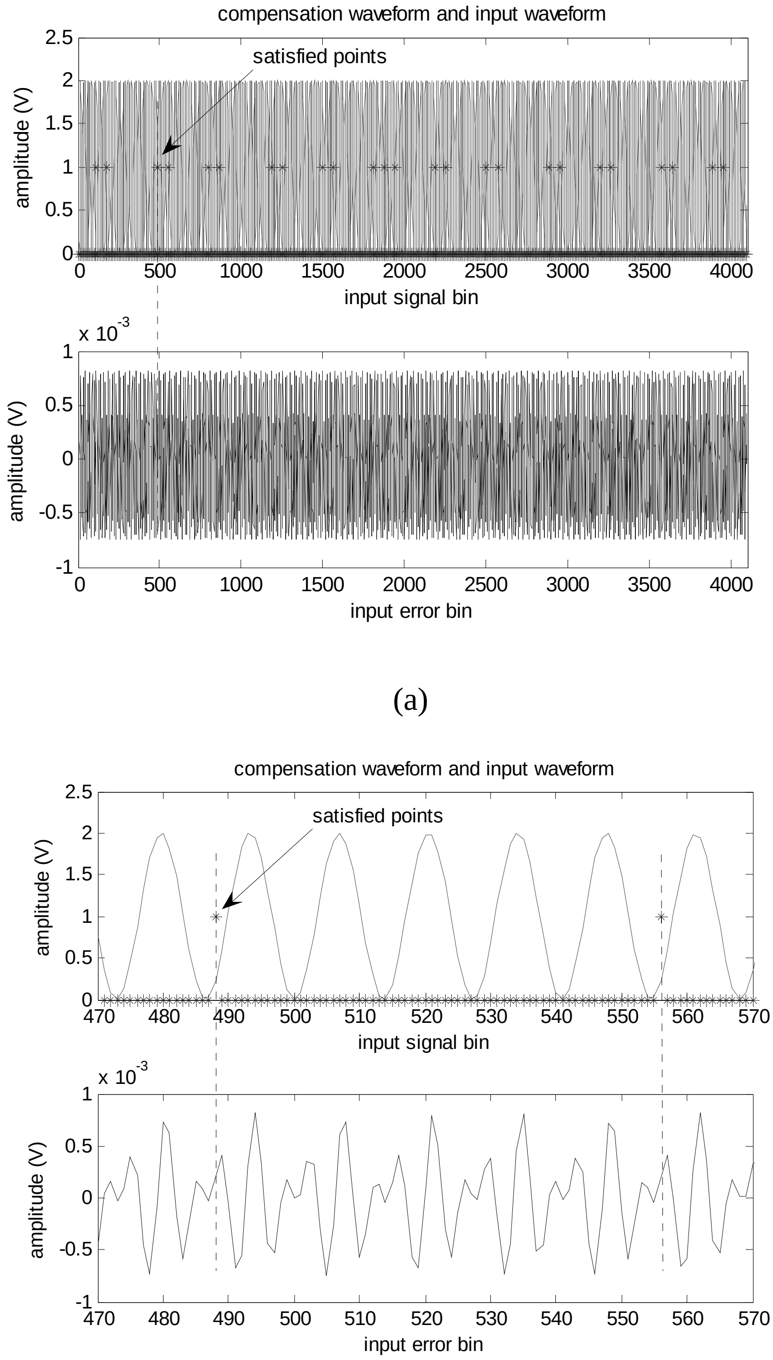
Fig. 5.6 and Fig. 5.7 show error waveforms at different frequencies. For a pair of (x_x,x_{step}), ‘*’ marks the satisfied points. Fig. 5.6 is the plot for low frequency ($3f_0<f_s/2$). Fig. 5.7 is the plot for high frequency (f_0 is close to $f_s/2$). At low frequencies, the error waveform is periodic to the satisfied points, so the error values hit by these points are of close values. While at high frequencies, there are two satisfied points in a period, and each of them hits a different value. Therefore, two values occur at high frequencies.

The errors for given (x_x,x_{step}) and their extracted value determined by averaging these values are shown in table 5.1 at different input frequencies. In the table, the extracted values are close to their real errors at low frequencies, while the extracted values deviate from their real errors with increasing input frequency. Values become near zero at frequencies close to the Nyquist rate.

The previous discussion revealed that SFDR improvement decreases due to deviation of compensation error values for a given (x_x,x_{step}). This observation is taken to the error with a width equal to a clock period. How about the error codes in the dynamic glitch calibration, since those pulses have a much narrower width?

Table 5. 1 Compensation error values for different input frequencies at different input code and step-jumping

(x_n, x_{step})	(6,5)	(7,7)	(7,7)	(7,7)	(8,7)	(10,7)
M	201	301	401	551	701	851
	-0.00047	0.00022	0.0010	0.00094	0.0012	-0.0025
	-0.00047	0.00021	0.0004	-0.00092	-0.001	0.0025
	-0.00048	0.00022	0.0010	-0.00095	0.0012	-0.0025
	-0.00045	0.00020	0.0004	0.00093	-0.001	0.0025
	-0.00046	0.00022	0.0010	-0.00094	0.0012	0.0025
	-0.00046	0.00021	0.0004	0.00093	-0.0011	-0.0025
E	-0.00047	0.00021	0.0010	0.00095	0.0012	0.0025
R	-0.00047	0.00020	0.0004	-0.00094	-0.0011	0.0025
R	-0.00045	0.00022	0.0010	0.00092	0.0013	-0.0025
O	-0.00046	0.00021	0.0009	0.00094	-0.0011	0.0025
R	-0.00046	0.00023	0.0004	-0.00093	0.0013	0.0025
	-0.00047	0.00021	0.0004	-0.00095	-0.0011	-0.0025
V	-0.00047	0.00020	0.0010	0.00094	0.001	0.0025
A	-0.00045	0.00022	0.0009	-0.00092	0.0013	0.0025
L	-0.00045	0.00020	0.0004	-0.00095	-0.0012	-0.0025
U	-0.00046	0.00022	0.0004	0.00093	0.001	0.0025
E	-0.00046	0.00021	0.0009	0.00095	-0.0012	-0.0025
S	-0.00047	0.00022	0.0004	-0.00094	0.001	-0.0025
	-0.00048	0.00020	0.0010	0.00092	-0.0012	0.0025
(V)	-0.00045	0.00022	0.0004	0.00095	0.0011	-0.0025
	-0.00046	0.00021	0.0010	-0.00093	-0.0012	-0.0025
	-0.00046	0.00021	0.0004	-0.00095	0.0011	0.0025
	-0.00047	0.00020	0.0010	0.00094	-0.0013	-0.0025
	-0.00047	0.00022	0.0004	-0.00092	0.0011	-0.0025
	-0.00045	0.00020	0.0010	-0.00095	-0.0013	0.0025
Mean (V)	-0.00046	0.00021	0.00069	-0.000001	0.000003	-0.00009

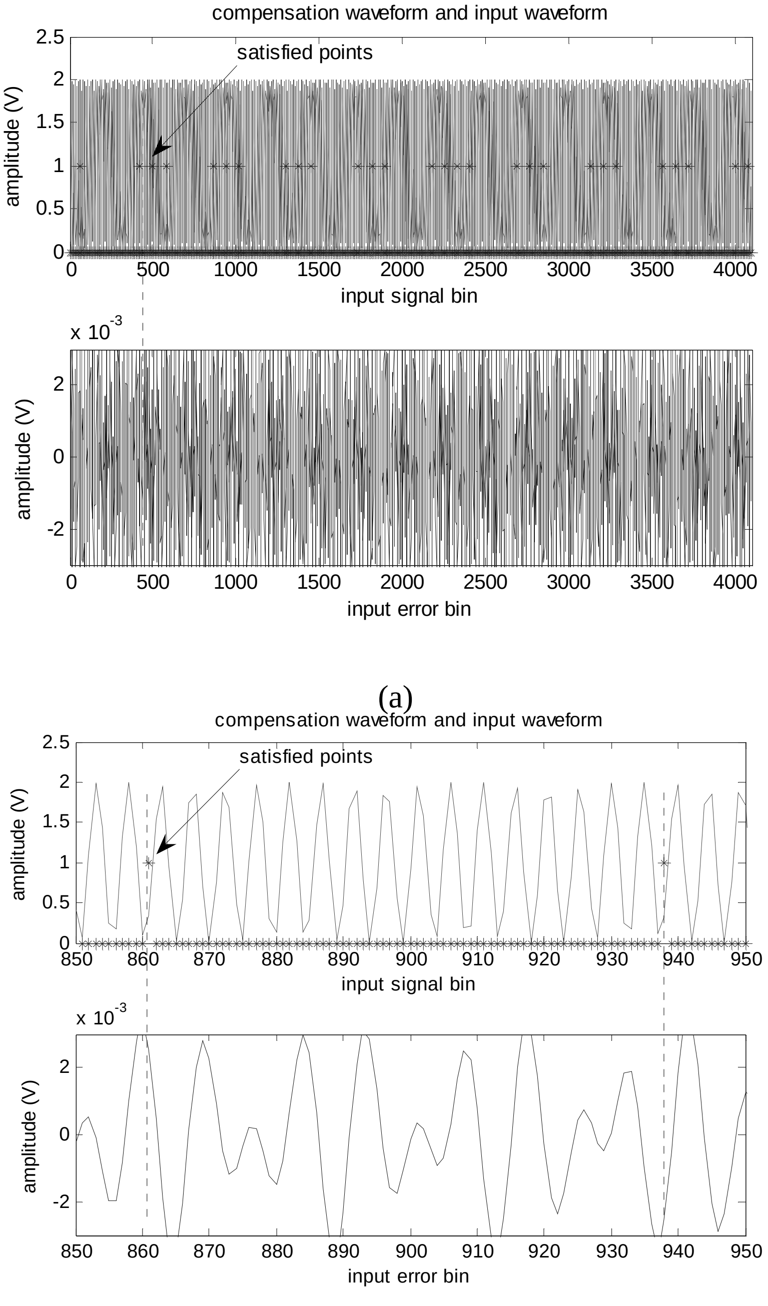


(a)

(b)

Figure 5. 6 Input signal and error waveform for (x_n, x_{step})=(7,7) @ f_{sig}=7.34MHz, f_s =50mHz

(a) whole window view (b) zoom in view between bin 470 and 570



(a)

(b)

Figure 5. 7 Input signal and error waveform for (x_n, x_{step})=(10,7) @ f_{sig}=20.8MHz, f_s =50mHz

(a) whole window view (b) zoom in view between bin 850 and 950

Table 5.2 is the error pulse amplitude for a given (x_x,x_{step}) at each input frequency in the 12-bit transistor-level dynamic-glitch calibration. The data in the table show that the error values are very close to each other at both low- and high-input frequencies. This indicates that the mean used to compensate for nonlinear distortions is very close to the real error. Therefore, much significant improvement of SFDR can be achieved through this approach.

Table 5. 2 Dynamic glitch error values for different input frequencies at different input code and step-jumping

(x_n, x_{step})	(8,7)	(23,22)	(38,34)
M	35	81	189
Error (V)	-0.00016926 -0.00020762 -0.00029773	-0.0022182 -0.0011162 -0.0020118	-0.0009618 -0.0015234 -0.0006088
Mean (V)	-0.000224869	-0.001782067	-0.001031343

- 12 bits current steering DAC
- Sampling frequency f_s=200MHz
- Input signal frequency $f_{sig} = \frac{M}{512} \times f_s$

Table 5.3 is the error values for dynamic settling calibration at different input frequencies. The table shows that the values vary widely for a given (x_x,x_{step}) at an input frequency. Therefore, it is no surprise to expect large difference between the mean and the real error and little SFDR improvement through dynamic settling calibration.

Table 5. 3 Dynamic settling error values for different input frequencies at different input code and step-jumping

(x_n, x_{step})	(8,7)	(23,22)	(38,34)
M	35	81	189
Error (V)	4.77363E-05 9.14835E-05 0.000103844 -8.74185E-05	-0.0001025 0.00016688 0.00010294	-8.538E-05 -0.0001575 -0.0002532
Mean (V)	3.89112E-05	5.57698E-05	-0.000165367

- 12 bits current steering DAC
- Sampling frequency f_s=200MHz
- Input signal frequency $f_{sig} = \frac{M}{512} \times f_s$

CONCLUSIONS

In this dissertation we have addressed the issue of dynamic nonlinearities in the digital-to-analog converters. We have explored a novel approach to calibrate current-steering DACs. This approach assumes that the nonlinearities come mainly from the early period of the transition region and a series of pulses can be generated according to the error waveform and added to the output of the raw DAC to compensate for nonlinearities. The errors are regarded as a function of current input signal x_n and the step jumping height x_{step}. A small error look-up table was built for each input signal frequency. The full-scale look-up table consisting of many small tables can be constructed to calibrate arbitrary input signals. The validity of this approach was demonstrated with a 12-bit current steering DAC. The simulation results showed that the main nonlinearities were from the early period of the input codes transition, called dynamic glitch errors in this dissertation. The dynamic glitch error calibration approach used by applying narrow pulses to compensate the nonlinearities did efficiently reduce the distortions, and the SFDR improved significantly improved using this calibration scheme. The robustness of this approach was also proved in the simulation. The calibration was then applied to a commercial current-steering DAC. Experimental measurement results are also provided for a special case of this dynamic calibration algorithm that show that the dynamic performance can be improved through dynamic calibration, provided the mean error values in the table are close to their real values.

REFERENCES

[1]. J. Sevenhans, Z. Chang, “A/D and D/A conversion for telecommunication”, IEEE citcuits and Dev. Mag., Vol. 14, pp. 32-42, Jan. 1998.

[2]. Yonghua Cong; Geiger, R.L., “A 1.5-V 14-bit 100-MS/s self-calibrated DAC”, IEEE J. Solid-State Circuits, Vol. 38 , Issue: 12 , Pages:2051 – 2060, Dec. 2003

[3]. I. Mehr, M. Prabir, D. Paterson, “A 12-bit integrated analog front end for broadband wireline networks”, IEEE J. Solid-State Circuits, Vol. 37, Pages: 302-309, May 2001.

[4]. David A. Johns Ken Martin, *Analog integrated circuit design*, by John Wiley & Sons, Inc. New York, 1997.

[5]. Falakshahi, K.; Yang, C.-K.K., Wooley, B.A., “A 14-bit, 10-Msamples/s D/A converter using multibit /spl Sigma//spl Delta/ modulation”, IEEE J. Solid-State Circuits, Vol. 34, Issue 5, Pages: 607 – 615, May 1999.

[6]. P. Hendriks, “Specifying communications DACs”, IEEE Spectrum, Vol.34, pp. 58-69, July 1997.

[7]. Chi-Hung Lin; Bult, K.,” A 10-b, 500-MSample/s CMOS DAC in 0.6 mm^2” , IEEE J. Solid-State Circuits, Vol. 33, Issue 12, Page:1948 – 1958, Dec. 1998.

[8]. Van den Bosch, A.; Steyaert, M.; Sansen, W., “SFDR-bandwidth limitations for high speed high resolution current steering CMOS D/A converters” Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on, Vol.3, 5-8, Sept. 1999.

[9]. Wikner, J.J.; Nianxiong Tan, “Modeling of CMOS digital-to-analog converters for telecommunication”**,** Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, Vol.46, Issue: 5, Pages:489 – 499, 1999.

[10]. Bosch, Anne van den, *Static and dynamic performance limitations for high speed D/A converters*, Boston: Kluwer Academic Publishers, c2004.

[11]. Rudy van de Plassche, CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, Norwell, MA, Kluwer, 2003.

[12]. Luschas, S.; Lee, H.-S., “Output impedance requirements for DACs”, ISCAS '03. Proceedings of the 2003 International Symposiumon, Vol. 1, 25-28, Pages: I-861 - I-864, May 2003.

[13]. Pelgrom, M.J.M., etc., “Matching properties of MOS transistors”, IEEE J. Solid-State Circuits, Vol. 24, Issue 5, Pages: 1433-1439, Oct 1989.

[14]. Van Der Plas, G.A.M., etc., “A 14-bit intrinsic accuracy Q^2 random walk CMOS DAC”, IEEE J. Solid-State Circuits, Vol. 34, Issue: 12, Pages: 1708 – 1718, Dec. 1999.

[15]. Adel S. Sedra, Kenneth C. Smith, *Microelectronic circuits, -4th ed*. Oxford University Press, Inc. 1998.

[16]. B.Razavi, Principles of Data Conversion System Design, Piscataway, NJ: IEEE Press, 1995.

[17]. Bastos, J.; Marques, A.M.; Steyaert, M.S.J.; Sansen, W., “A 12-bit intrinsic accuracy high-speed CMOS DAC”, IEEE J. Solid-State Circuits, Vol. 33, Issue: 12, Pages: 1959 – 1969, Dec. 1998.

[18]. Bugeja, A.R.; et al. “A 14-b, 100-MS/s CMOS DAC designed for spectral performance”, IEEE J. Solid-State Circuits, Vol. 34, Issue: 12, Pages: 1719 – 1732, Dec. 1999.

[19]. Tesch, B.J.; Garcia, J.C., “A low glitch 14-b 100-MHz D/A converter”, IEEE J. Solid-State Circuits, Vol. 32, Issue: 9, Pages: 1465 – 1469, Sept. 1997 .

[20]. Qiuting Huang; Francese, P.A.; Martelli, C.; Nielsen, J., “A 200MS/s 14b 97mW DAC in 0.18/spl mu/m CMOS”, Solid-State Circuits Conference, Pages: 364 - 532 Vol.1, 2004.

[21]. Mercer, D., “A 16-b D/A converter with increased spurious free dynamic range”, IEEE J. Solid-State Circuits, Vol. 29, Issue: 10, Pages: 1180 – 1185, Oct. 1994 .

[22]. Pelgrom, M.J.M., Roorda, M., “An algorithmic 15-bit CMOS digital-to-analog converter”, IEEE J. Solid-State Circuits, Vol. 23, Issue: 6, Pages: 1402 – 1405, Dec. 1988.

[23]. Van Den Bosch, A, et al, “A 12 b 500 MSample/s current-steering CMOS D/A converter”, Solid-State Circuits Conference, Pages: 366 - 367, 466, 2001.

[24]. Tang, A.T.K.; Toumazou, C., “Self-calibration for high-speed, high-resolution D/A converters”, Advanced A-D and D-A Conversion Techniques and their Applications, 1994. Second International Conference on 6-8 July 1994, Pages: 142 – 147, July 1994.

[25]. Bugeja, A.R.; Bang-Sup Song; “A self-trimming 14-b 100-MS/s CMOS DAC”, IEEE J. of Solid-State Circuits, Vol. 35, Issue: 12, Pages: 1841 - 1852, Dec. 2000.

[26]. T.; Nakamura, Y.; Nakaya, M.; Asai, S.; Akasaka, Y.; Horiba, Y., “An 80-MHz 8-bit CMOS D/A converter Miki”, IEEE J. Solid-State Circuits, Vol. 21, Issue 6, Pages: 983 – 988, Dec 1986.

[27]. D. Groeneveld et al., “A self-calibration technique for Monolithic high-resolution D/A converters”, IEEE J. Solid-State Circuits, Vol. 24, Pages: 1517-1522, Dec. 1989.

[28]. Robert Mckinstry Robinson Neff, “Automatic synthesis of CMOS Digital/Analog Converters”, thesis, University of California at Berkeley, 1987.

[29]. Riley, K.J., et al., “Dynamic compensation of digital to analog converters”, Instrumentation and Measurement Technology Conference, 1999. Vol. 2, 24-26 Pages: 1310 - 1315, May 1999.

[30]. Gregory A. Friel, *Measurement and Compensation of Digital to Analog Converter Nonlinearity*, thesis, University of Maine, 2000.

[31]. Schouwenaars, H.J.; Dijkmans, E.C.; Kup, B.M.J.; Van Tuijl, E.J.M., “A Monolithic Dual 16-Bit D/A Converter”, IEEE J. Solid-State Circuits, Vol. 21, Issue 3, Page: 424 – 429, Jun 1986.