Flexible and self-calibrating current-steering digital-to-analog converters : analysis, classification and design

Flexible and self-calibrating current-steering Digital-to-Analog Converters: analysis, classification and design

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 14 januari 2010 om 16.00 uur

door

Georgi Ivanov Radulov

geboren te Plovdiv, Bulgarije

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. A.H.M. van Roermund

Copromotoren:
dr.ir. J.A. Hegt
en
dr. P.J. Quinn MSc

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-2125-8

To Ivan and Elissaveta

This research work has been financially supported by the Dutch Technical Foundation STW, project ECS.6098.

This research work has been technically supported by Xilinx Ireland, Mixed-Signal Design Group.

This research work has been realized in the Mixed-Signal Microelectronics group at the Electrical Engineering faculty of the Eindhoven University of Technology.

ABSTRACT

This research work proposes new concepts of flexibility and self-correction for current-steering digital-to-analog converters (DACs) which allow the attainment of broad functional and performance specifications, high linearity, and reduced dependence on the fabrication processes.

This work analytically investigates the DAC linearity with respect to the accuracy of the DAC unit elements. The main novelty of the proposed approach is in the application of the Brownian Bridge (BB) process to precisely describe the DAC Integrated-Non-Linearity (INL). The achieved results fill a gap in the general understanding of the most quoted DAC specification - the INL.

Further, this work introduces a classification of the highly diverse current-steering DAC correction methods. The classification automatically points to methods that do not exist yet in the open literature (gaps). Based on the clues of the common properties and identified common techniques in the introduced classification, this work then proposes exemplary solutions to fill in the identified gaps.

Further, this work systematically analyses self-calibration correction methods for the DAC mismatch errors. Their components are analyzed as three building blocks: self-measurement, error processing algorithm and self-correction block. This work systemizes their alternative implementations and the associated trade-offs. The findings are compared to the available solutions in the literature. The efficient calibration of the DAC binary currents is identified as an important missing method. This work proposes a new methodology for correcting the mismatch errors of both the nominally identical unary and the scaled binary DAC currents.

Further, this work proposes a new concept for DAC flexibility. This concept is realized in a new flexible DAC architecture. The architecture is based on a modular design approach that uses parallel sub-DAC units to realize flexible design, flexible functionality and flexible performance. The parallel sub-DAC units form a mixed-signal platform that is capable of many DAC correction methods, including calibration, error mapping, data reshuffling, and harmonic distortion cancellation.

This work presents the implementation and measurement results of three DAC test-chip implementations in 250nm, 180nm, and 40nm standard CMOS IC technologies. The test-chips are used as a tool to practically investigate, validate, and demonstrate two main concepts of this thesis: self-calibration and flexibility. Particularly, the 180nm test-chip is the first reported DAC implementation that calibrates the errors of all its current sources and features flexibility, as suggested in this work. The calibration of all current sources makes the DAC accuracy independent of the tolerances of the manufacturing process. The overall DAC accuracy depends on a single design parameter – the correction step. The third test-chip is the first reported DAC implementation in 40nm CMOS process. A 12 bit DAC core in this test-chip occupies only 0.05mm^2 of silicon area, which is the smallest reported area for a 12 bit current-steering DAC core.

LIST OF ABBREVIATIONS

AC alternating current

ADC analog-to-digital converter

ASIC application-specific integrated circuit

BB Brownian Bridge

BIST built-in-self-test

CALDAC calibrating digital-to-analog converter

CMOS complementary metal-oxide semiconductor

CS current steering

D/A digital-to-analog

DAC digital-to-analog converter

DC direct current

DEM dynamic element matching

DNL differential non-linearity

DQS differential-quad-switching

DR dynamic range

DSP digital signal processor (processing)

ENOB effective number of bits

ETF error transfer function

ETFP error-transfer-function-prevention

ETFC error-transfer-function-correction

FFT fast Fourier transform

FS full scale

FPGA field-programmable gate array

HD harmonic distortion

HF high frequency

I/O input/output

IC integrated circuit

INL integrated non-linearity

IM inter modulation

IMD inter-modulation distortion

LSB least significant bit(s)

LUT look-up table

LVDS low voltage differential signaling

MOS metal-oxide-semiconductor

MOSFET metal-oxide-semiconductor field-effect transistor

MSB most significant bit(s)

NRZ non-return-to-zero

OFDM orthogonal frequency division multiplexing

PA power amplifier

PCB printed circuit board

PDF probability density function

PWM pulse-width modulated

RAM random access memory

RF radio frequency

RTL register transfer level

RZ return-to-zero

S/D source/drain

S/N signal-to-noise

SAR successive approximation register

SC switched-capacitor

SI switched-current

SFDR spurious-free dynamic range

SNR signal-to-noise ratio

SNDR signal-to-noise-and-distortion-ratio

STFC signal-transfer-function-compensation

SPICE simulation program with integrated circuit emphasis

SoC system-on-chip

STF signal transfer function

T/H track and hold

THD total harmonic distortion

UHF ultra-high frequency

VCO voltage-controlled oscillator

VHDL very high speed IC hardware description language

VLSI very-large scale integration

WL area width times length area (of a transistor)

TABLE OF CONTENTS

PART I: INTRODUCTION AND BASICS

This part includes chapters that set the aims, scope, and approach of the thesis. The thesis and its original contributions are outlined. Basic DAC concepts are briefly considered in the framework of modern microelectronics.

1 . C h a p t e r

INTRODUCTION

Microelectronics is an ingredient part in the construction of modern society. The needs of information processing and communication drive the irreversible development of microelectronics. The revolutionary advent and continual evolution of the Complementary Metal Oxide Semiconductor (CMOS) technology and digital electronics established, and continue to create advancements over analog design in such technological categories as computing, communications, and electronic entertainment. Access to these technologies, therefore, is becoming increasingly affordable and realizable.

The digital age, however, has not removed the need for analog circuitry. Consequently, mixed-signal micorelectronics and in particular Analog to Digital Conversion (ADC) and Digital to Analog Conversion (DAC) technologies are very much in demand in order to bridge the gap between the analog and digital domains. ADC and DAC technologies are unavoidably needed, as long as the society relies on digital electronics in areas such as computing, communications, and electronic entertainment.

An illustrative example for these demands is found in communications, both wired and wireless. Society needs faster, cheaper, and more reliable communication than what is presently available. Although digital electronics directly benefit from the advancements of CMOS technologies and can satisfy the functionality needs of the consumer, a critical path still exist between what the consumer wants and what the technology can deliver and that is mixed-signal microelectronics. Mixed-signal microelectronics requires solving fundamental physical constraints, such as noise, matching of components, and fabrication process parasitics.

1.1. MODERN MICRO-ELECTRONICS AND FLEXIBILITY

DAC design in general benefits from CMOS technology scaling. The shrinkage of the transistors allows higher sampling rates and hence larger signal bandwidths. The switching of currents is faster and hence the DAC linearity at higher speeds improves. Figure 1.1-1 compares the CMOS technology development in the years with "the averaged" CMOS technology for published state-of-the-art DACs.

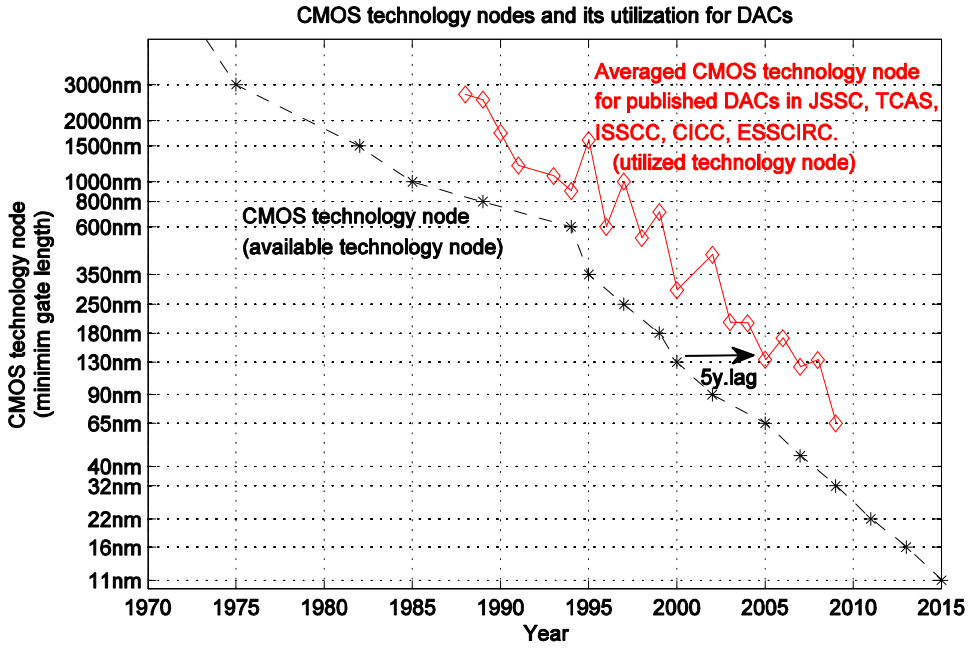


Figure 1.1-1. CMOS technology nodes and its utilization for published DAC designs.

The CMOS technology nodes are shown with stars and a dashed line. The averaged CMOS technology for published DACs is shown with solid line and diamonds. It demonstrates how the CMOS technology advancements are utilized in DAC design. The data is gathered from the majority of published DAC designs in the following leading journals and conference proceedings since 1988: Journal of Solid-State-Circuits (JSSC), Transactions on Circuits and Systems I & II (TCAS), International Solid-State Circuits Conference (ISSCC), Custom Integrated Circuits Conference (CICC), and European Solid-State Circuits Conference (ESSCIRC). The collected data is presented in APPENDIX A.. Each year, different state-of-the-art designs are published in different technologies. To get a representation about the utilization of the CMOS technology per year, an average is taken over the published DAC designs per year. This demonstrates that DAC state-of the art in general follows CMOS technology development. A general trend is that state-of-the art of CMOS technology is utilized in state-of the art DAC designs after a lag of about 5 years. This lag is referred to as "the design gap".

The design gap is the technical time needed by designers and scientists to accommodate new CMOS technologies - to design, to characterize and to publish works concerning new DAC test-chip realizations in new technology nodes. Many DAC designers face common basic challenges in the process of designing in a new technology node, e.g. transistor model characterization, optimizing basic DAC cells, designing for

transistor matching. Thus, designers need to firstly address these basic challenges and then realize the essential functionality. This represents a major overlapping design effort. It reduces the overall efficiency of the utilization of new technologies.

This problem should be considered in the context of modern CMOS technologies. Although the price per wafer increases from CMOS generation to generation, the cost per transistor reduces. Therefore, the cost per function reduces from generation to generation. Thus, attractive products for modern CMOS technologies are large designs, featuring massive co-integration (e.g. System-On-Chips (SoC)), high level of redundancy (e.g. multi-core designs), and self-awareness (e.g. Built-In-Self-Test (BIST)).

To improve the design efficiency, in order to reduce the design gap and to effectively decouple the application DAC design from the silicon design, this thesis considers the concept of flexibility. A flexible DAC platform should provide a set of fundamental building blocks for DAC design that can be used by system designers (i.e. end-customers) to quickly realize their specific required DAC functionality.

1.2. AIMS OF THE THESIS

The two major aims of this thesis are to advance the existing knowledge on efficient and robust high-performance current-steering DACs and to introduce and investigate the concept of DAC flexibility. These two aims directly concern the co-integration of DAC interfaces in modern complex SoC solutions.

The thesis aims to structure, refine and extend the existing knowledge on DAC correction methods. The provided discussion needs to take into account the specifics of high-performance DACs co-integration in a SoC, e.g. high efficiency, robustness, redesign effort, and measurement costs. Note that every SoC co-integrated block needs to be efficient, since its price, i.e. costs, power consumption, extra requirements, price of test, etc., contribute to the overall SoC price, regardless of the fact if the block is actively used or idle in a particular end-user application. Furthermore, this block needs to be robust, since its own failure means a failure for the whole SoC product. Portability to other IC technologies is important, too. It determines how quickly a new SoC can appear on the market. These considerations are derived from the general guidelines of development of the micro-electronics market – towards mainly digital SoC solutions that feature large scale of functional integration. The new market challenges and perspectives require new answers, some of which this thesis aims to address.

The thesis aims to introduce and investigate the concept of DAC flexibility. The traditional approaches consider the DAC as a point-solution – given performance delivered at given cost. Normally, the performance-cost trade-off cannot be altered by the customers, since it is linked to the specific hardware realization. However, such approaches exclude those customers that need either a different point-solution or a number of point-solutions. This thesis aims to introduce DAC flexibility as a potential answer to this limitation. Both performance and price, i.e. the trade-off arguments, need to be flexible and hence controllable by the end-customer. Furthermore, the thesis aims to investigate the cost of flexibility and to propose solutions that fit in the framework of SoC co-integration together with the requirements for efficient and robust high-performance DACs.

1.3. SCOPE OF THE THESIS

The elaborated concepts of this thesis relate to Digital-to-Analog Converters (DACs). Nevertheless, some of the presented concepts can be indeed stretched far beyond this field.

As an example of DAC implementation, the most popular current-steering DAC architecture is considered. Its advantages in speed and accuracy make it a preferred choice for applications in the field of RF and digital communication. However, many of the presented concepts may be translated into other DAC architectures. For example, a proper translation from current as a basic analog entity into charge enables the presented concepts to be applied in switched-capacitor DAC implementations.

The thesis particularly focuses on DAC accuracy and the required resources to guarantee it. DAC correction methods to achieve accuracy and to provide high efficiency are stressed. Particularly, the self-calibration of DAC currents is elaborated. It is an example of a DAC correction method that includes self-measurement, an algorithm, and self-correction – aspects that can easily be translated to other correction methods. Beyond this, the thesis discusses aspects of power consumption, dynamic linearity, ease of implementation, etc.

For the discussion, this thesis assumes FPGAs (Field-Programmable-Gate-Arrays) as an example of a SoC. The thesis argues the mutual advantages of the co-integration between DACs and FPGAs. FPGAs are also chosen because of their digital flexibility, which makes them particularly suitable for the flexibility aim (chapter 1.2.) of this thesis. However, the discussed concepts are applicable to other types of digital SoC.

1.4. SCIENTIFIC APPROACH

To attain the two major aims defined in chapter 1.2. within the scope outlined in chapter 1.3. , this thesis undertakes the following scientific approach. DAC correction methods are considered as an indispensible resource both to achieve accuracy, efficiency and robustness in high-performance DACs and to introduce DAC flexibility. The DAC errors and available DAC correction methods are analyzed and modeled. A classification is proposed that structures the available knowledge. The missing DAC correction methods are identified and new methods to fill in these gaps are proposed. Some of the new methods are elaborated, leading to both specific and general conclusions. Test chip implementations validate in practice the proposed concepts and help investigate new ideas.

DAC correction methods that are available from the open literature are considered. The approach of this thesis is to extract the fundamental mechanisms of error correction from each considered method, rather than evaluating the achieved performance point of the particular published implementation. It is argued that the fundamental mechanisms for error correction can be combined to achieve flexibility.

Analysis of DAC errors is required to understand the working of the correction mechanisms. Then, analysis is also applied to the DAC correction mechanisms to identify common characteristics, advantages and disadvantages.

Modeling of DAC errors is required to further understand the workings of the correction mechanisms. Modeling of DAC errors is used to explain results of published DAC correction methods that the available DAC models cannot explain.

Classification of the DAC correction methods structures the available knowledge. It is based on the first three considered steps in this scientific approach: literature overview of the DAC correction methods, analysis of errors and correction methods, and modeling. It orders the available DAC correction methods, the links between them, their common characteristics, advantages and disadvantages. The classification can put the available knowledge in the context of the modern challenges and trends in the field of mixed-signal design.

The classification also reveals missing DAC correction methods. It is reasoned why these gaps exist, how they might be filled in and how the classification may provide clues to finding new correction methods.

New concepts are proposed and analyzed of the context of the two main aims of the thesis – efficient and robust high performance DACs and flexibility. These concepts include the upgrade of existing knowledge and the introduction of new knowledge.

Some of these new concepts are chosen for rigorous investigation, which reveal both specific and general aspects of the proposed concepts and help clarify the challenges for attaining the two main thesis aims.

Finally, three test chip realizations, in 250nm, 180nm, and 40nm CMOS technology, validate in practice some of the proposed concepts. Practical challenges are revealed and analyzed. The test chip implementations support the proposed new concepts and associated analysis. These implementations cover high-resolutions and high-accuracy of up to 14-16 bits and high-speed up to 650MS/s. They also indicate where further research is needed.

1.5. OUTLINE OF THE THESIS

This thesis contains 21 chapters ordered in 6 groups, forming 6 major parts. The first part is to briefly introduce the reader to the discussed subject. The second part contains chapters that provide an overview of the available from the literature material. The third part contains chapters that propose new modeling, analysis and classification. The fourth part contains chapters that propose new concepts and methods, advancing the state-of-the art. The fifth part contains chapters that present new design examples and measurement results. The sixth part concludes the thesis. Figure 1.5-1 shows a graphical representation of the thesis outline with the relations between the chapters.

PART I contains two chapters. Chapter 1 introduces the reader to the thesis. Chapter 2 introduces the reader to the subject of digital-to-analog converters (DACs).

PART II contains two chapters. Chapter 3 discusses selected state-of-the-art DAC correction methods that do not use exact error information to improve performance. Chapter 4 discusses selected state-of-the-art DAC correction methods that use exact error information to improve performance.

PART III contains four chapters. These cover two areas: modeling and analysis of errors that can be corrected by DAC correction methods and classification and analysis of DAC correction methods. The discussion of both areas follows similar approaches. Firstly, a global discussion is provided. Then, an in-depth analysis for a representative selected area is provided. Chapter 5 provides a broad view on DAC errors and suggests a model to explain how DAC errors cause performance degradation. Chapter 6 selects the DAC current amplitude mismatch errors and analyses in detail the process of generating DAC non-linearity. Chapter 7 introduces a classification of DAC correction methods to counteract the errors. Chapter 8 selects the current calibration DAC correction method for an in-depth analysis.

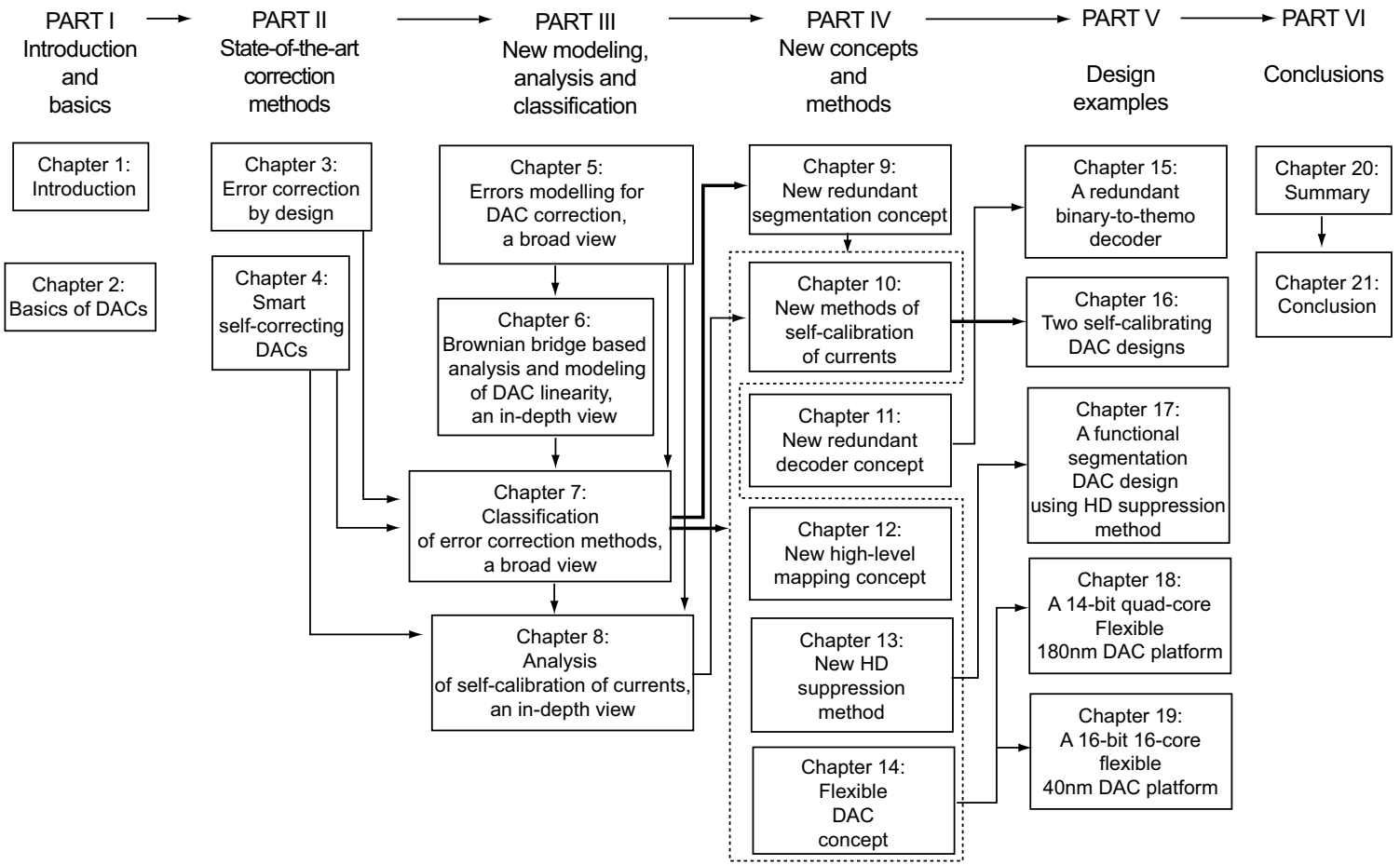


Figure 1.5-1. Outline of the thesis.

PART IV contains 6 chapters. Chapter 9 proposes a new segmentation concept to divide the DAC analog resources. Chapter 10 proposes new methods for self-calibration of current mismatch errors. Chapter 11 proposes a new binary-to-unary decoder with redundant switching sequences. It is an example of a very efficient low-level mapping method for DAC unary currents. Chapter 12 proposes a new high-level mapping method for generic DAC architectures, using the functional segmentation introduced in chapter 9. Chapter 13 proposes a new DAC correction method using parallel DAC functions to suppress harmonic distortion (HD). Chapter 14 proposes the new concept of DAC flexibility.

PART V contains 5 chapters, describing three test chip implementations and reporting measurement test results. Chapter 15 presents an implementation and measurement results for the redundant decoder concept presented in chapter 11. Chapter 16 presents implementation and measurement results for the new calibration methods presented in chapter 10. The presented results are based on two test chip implementations, validating the consistency of the proposed methods. Chapter 17 presents implementation and measurement results for the new method for suppression of HD presented in chapter 13. Chapter 18 presents an implementation and measurement results of a flexible DAC platform based on 4 12 bit sub-DAC cores. Chapter 19 presents an implementation and measurement results of a flexible DAC platform based on 16 12 bit sub-DAC cores.

PART VI concludes the thesis. Chapter 20 provides summary of the thesis. Chapter 21 provides general conclusions.

1.6. ORIGINAL CONTRIBUTIONS

The original contributions of this work are achieved in cooperation with prof. dr. ir. Arthur H.M. van Roermund, dr. ir. Hans A. Hegt, dr. ir. Patrick J. Quinn, prof. dr. Remco W. van der Hofstad, dr. Markus Heydenreich, dr. Olaf Wittich, and ir. Pieter Harpe. These contributions can be summarized in:

1. Unary and binary current-source self-calibration methodology based on:
   - self-correction implemented with small calibrating DACs (CALDACs). For better post-calibration accuracy, the architecture of the CALDACs uses segmentation, whereas the LSB part is implemented in a binary and the MSB part in a unary way. For bi-directional current output, the CALDACs have controllable current-mirror at their outputs.
   - self-measurement implemented with current switches to deviate the current for measurements. A current comparator (1 bit ADC) is used for a 1 bit measurement. Through successive 1 bit measurements and the loop '*comparator – CALDAC - measured current*', high measurement accuracies are achieved. The current comparator is realized as:
     - comparison of two currents;
     - evaluation of the sign of the residue of two subtracted currents.
   - calibration algorithms for both unary and binary currents realized as simple Finite-State-Machines. The algorithms use a technique to control the CALDAC correction quantization error and hence improve the DAC post-calibration accuracy.

This research work led to the following two US patents:

- US7076384, Method and apparatus for calibrating a current-based circuit;
- US7466252, Method and Apparatus for Calibrating a Scaled Current Electronic Circuit.

2. A current-steering DAC architecture for flexibility of DAC design, functionality, and performance that is based on:
   - Parallel sub-DACs;
   - A digital pre-processor;
   - Analog post-processing.

   This research work led to the award for Outstanding Student Paper granted by the IEEE conference APPCAS'08.

3. DAC performance improvement techniques based on parallel sub-DACs:
   - A mapping method: the digital pre-processor distributes the input digital word among the sub-DACs in such a way that the errors of the sub-DACs are mutually compensated;
   - A phase-shifting method: the digital pre-processor generates phase-shifted replicas of the digital input signal and sends these replicas to the sub-DACs. The sub-DACs convert the phase-shifted replicas resulting in analog outputs that are phase-shifted. When the sub-DAC output currents are combined, targeted harmonic distortion components can be suppressed.
   - Power saving method: the power of the unused sub-DACs can be switched off in order to save over-all power.

4. A self-measurement technique based on 1 bit current comparator and parallel sub-DACs.

5. The application of Brownian Bridge process to model the DAC INL and harmonic distortion components as a function of the mismatch of the DAC currents.

6. Three test-chip implementations:
   - A 12 bit self-calibrated current-steering DAC in 250nm CMOS;
   - A flexible 12 to 14 bit DAC based on 4 cores of 12 bit self-calibrated sub-DAC in 180nm CMOS;
   - A flexible 12 to 16 bit DAC based on 16 cores of 12 bit self-calibrated sub-DAC in 40nm CMOS;

7. A methodology for classification and analysis of DAC correction methods.

2. Chapter

BASICS OF DIGITAL-TO-ANALOG CONVERSION

This chapter discusses basic concepts of modern Digital-to-Analog Converters (DACs). The basic generic DAC functionality and specifications are discussed, followed by the algorithmic aspects of D/A conversion. Next, algorithmic segmentation is addressed in the context of utilization of analog resources. Then, architectures and implementation options are discussed. Different DAC implementations are briefly reviewed. Finally, the chapter discusses the most important implementation for this thesis, the current-steering DAC, for which the basic analog entity is the current unit.

2.1. INTRODUCTION

The importance of Digital-to-Analog converters (DACs) is a direct consequence of the utilization of digital electronics. In many applications, digital circuits can only be utilized providing an appropriate translation, i.e. conversion, of their digital output information into the analog world. The function of this translation is realized through DACs. For an in-depth discussion on the basic DAC concepts, the works of [1], [2], and [3] are recommended. This chapter presents the necessary basic DAC concepts for a proper discussion of the presented work further in the thesis.

Depending on the specific applications, there are various specifications that a D/A conversion should satisfy. This chapter discusses the DAC functionality and specifications in section 2.2. These specifications can be realized through different ways, i.e. algorithms. Section 2.3. discusses the algorithmic segmentation of the DAC analog resources. These algorithms can be implemented through various circuit solutions, e.g. current-based circuits, resistor-based circuits, capacitor-based circuits. Section 2.5. briefly reviews the main DAC implementations. Section 2.6. presents the current-steering DAC implementation and section 2.7. important DAC design challenges. Finally, a summary of the chapter is provided in section 2.8.

2.2. FUNCTIONALITY AND SPECIFICATIONS

Digital-to-Analog Converters (DACs) implement a Digital-to-Analog (D/A) conversion function, see Figure 2.2-1. The arguments of this function are digital data, reference clock and reference amplitude (unit). The output of the D/A function is the DAC analog output signal. The input signal is discrete in time and quantized in amplitude, coded in digital bits. The time-reference for the DAC is provided by its input clock signal. In most cases, the DACs are synchronized, requiring a separate clock input. However, there are some cases of asynchronous DACs, which interpret the time-reference through the change of the input digital data. In both cases, the output analog signal is continuous in time with quantized amplitude. The quantization of the output signal amplitude depends on the resolution of the digital input signal and that of the DAC.

The D/A function can also be considered as a translation of the abstract level of digital information to concrete analog entities, such as currents, voltages, power. Such a translation implies an analog characterization of the output of the DAC.

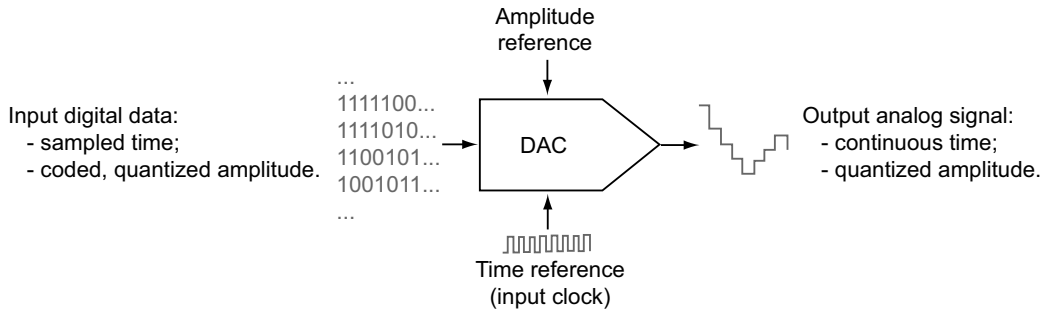


Figure 2.2-1. The DAC as black box: input-to-output transformation.

2.2.1. Static characterization

For a static characterization, the main representation of the D/A function is given by the D/A transfer characteristic. Figure 2.2-2 illustrates a D/A transfer characteristic which is derived from real test-chip measurements with magnified non-linearity by a factor of 150. The plot provides the static relation between the DAC input codes (x axis) and the

DAC output analog value (y axis, representing the DAC differential output voltage). The x axis is discrete and is only defined around the possible digital input codes, represented as bins in a plot. The number of bins is usually determined by the DAC resolution. The example of Figure 2.2-2 shows a 12 bit DAC. The y axis is continuous. The maximal value of the D/A function on the y axis represents the DAC full-scale (FS) range.

The straight line in Figure 2.2-2 is the nominally expected D/A transfer characteristic. It describes the ideal linear relation between the digital input and analog output. Several specifications can be defined, e.g. offset, gain, FS range. The non-linear graph is the actual, e.g. measured, D/A transfer characteristic. For the example of Figure 2.2-2, it is based on real measurement results of a 12 bit DAC with a magnified non-linearity by a factor of 150. The offset, gain and FS specification need to be defined based on the real measurement data. There are a number of ways to define these specifications. These depend on the way the empirical linear equivalent of the actual D/A transfer characteristic is defined. Without loss of generality, in this thesis the line connecting the initial and final points of the actual D/A transfer characteristic is used (dashed line in Figure 2.2-2). Based on this line, the empirical, e.g. measured, DAC offset, gain and FS can be defined, as shown in Figure 2.2-2. Note that there are other ways to define the empirical ideally linear line, e.g. “best-fit” line, see [4].

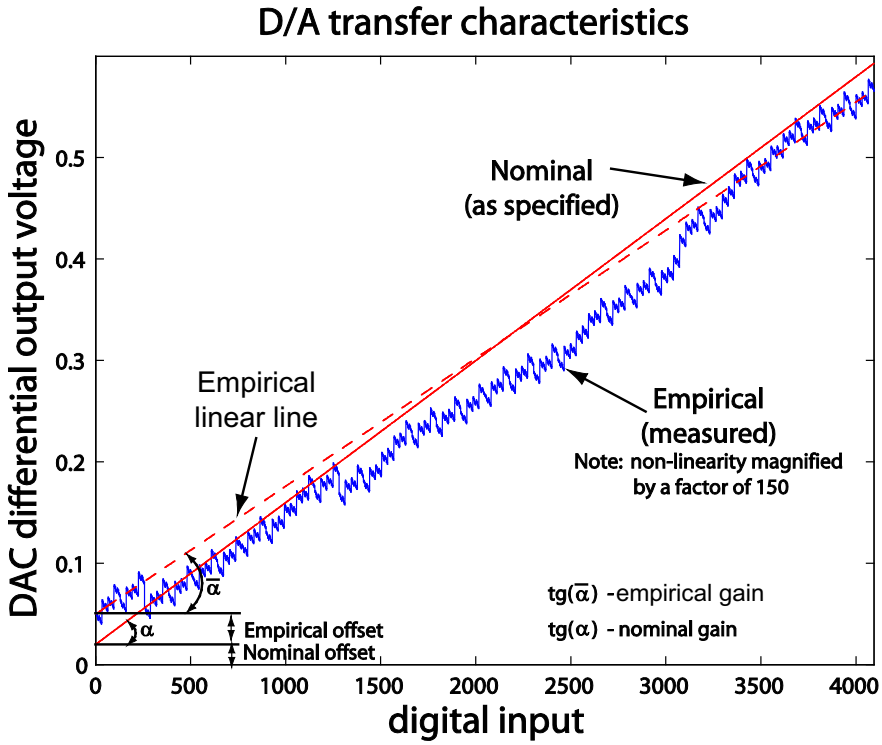


Figure 2.2-2. Three D/A functions for a DAC, as designed (straight line), empirical ideally linear (dashed line), and empirical (non-linear graph, derived from measurements with non-linearity magnified by a factor of 150).

The difference between the empirical ideal linear line and the empirical D/A transfer characteristic shows the DAC non-linearity. For a proper reading of the DAC non-linearity, usually it is normalized to the LSB step of the DAC output. In such a way the DAC INL (Integrated Non-Linearity), shown in Figure 2.2-3, is defined. The evaluation of the INL usually includes two main properties: the global shape of the graph and its deviation from the straight line. The shape of the graph indicates the dominant order of the DAC non-

linearity. For example, the shape shown in Figure 2.2-3 would suggest a strong second-order non-linearity. The deviation from the straight line indicates how strong the non-linearity is and hence how linear the DAC is. For example, the deviation shown in Figure 2.2-3 is about -500 LSB, which suggests a linearity that is 10bit less than the resolution, i.e. 2bit DAC linearity. For accurate analytical definitions of INL, refer to chapter 6.

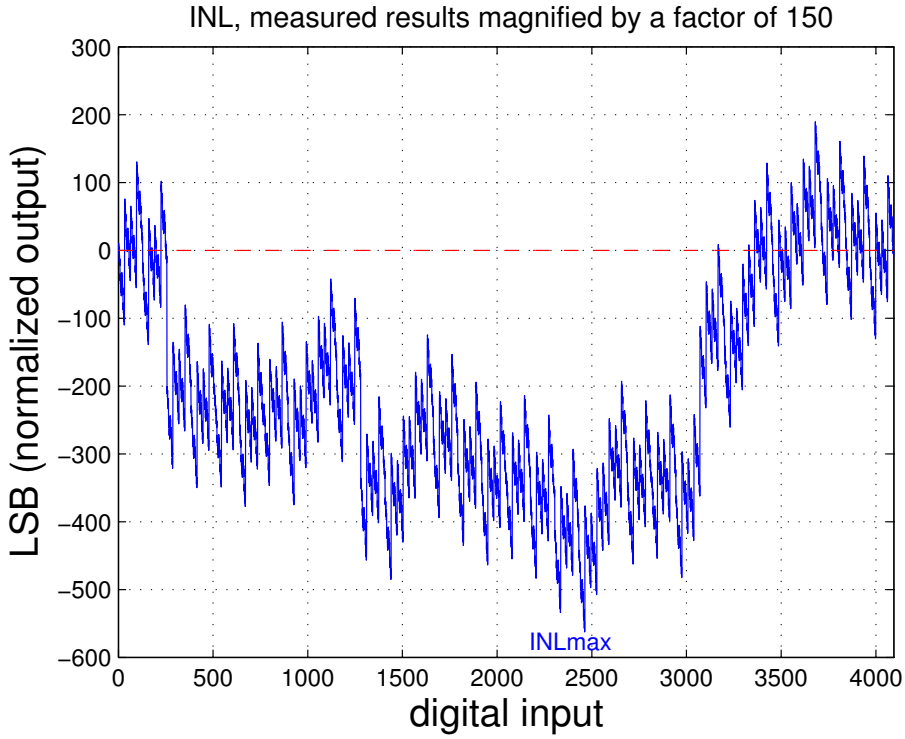


Figure 2.2-3. 12 bit DAC INL.

For many DAC applications, e.g. control and self-calibration as shown further in the thesis, the local behavior of the INL graph is important, i.e. the linearity between successive DAC code transitions. This can be characterized by the DAC DNL (Differential-Non-Linearity). The DNL characterizes the non-linearity for each LSB step. The DNL at code k equals the difference between the two code-consecutive INL values at codes *k+1* and *k*:

Equation 2.2-1

$$DNL_k = INL_{k+1} - INL_k$$

Figure 2.2-4 shows the corresponding DNL characteristic of the D/A transfer characteristic of Figure 2.2-2 and the INL characteristic of Figure 2.2-3. The DAC DNL is usually used to indicate DAC local errors. For example a large deviation for a given DAC analog unit is directly indicated as a spike in the graph. Another commonly used criterion is the DAC monotonicity. A DAC is monotonic if $DNL_k > -1$ for all *k*. The opposite, the non-monotonicity, is a strongly non-linear condition of the D/A transfer characteristic featuring a local gain with opposite sign. That is to say that an input digital code $x_1 > x_2$ is converted to DAC output $y(x_1) < y(x_2)$, while the overall DAC gain is positive.

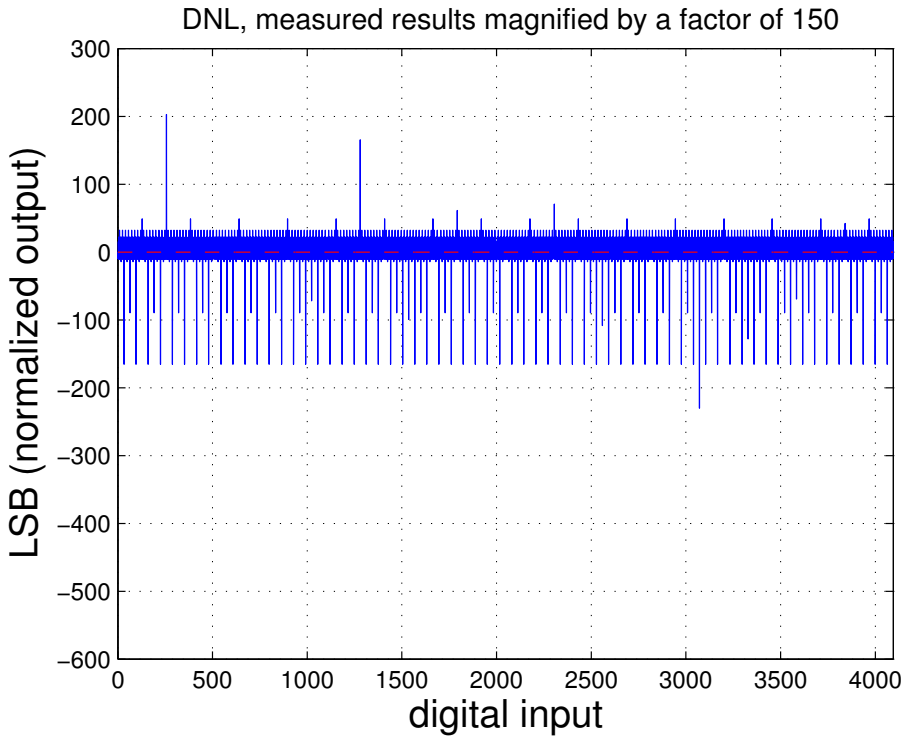


Figure 2.2-4. 12 bit DAC DNL.

2.2.2. Dynamic characterization

For the dynamic DAC characterization, many figures are widely used, depending on the DAC application and its requirements. For example, audio and video applications require strict specifications for glitch energy between the code transitions; radio-frequency (RF) communication applications require strict specifications for DAC dynamic linearity; digital communication applications require strict specifications on FS high-speed specifications eye patterns.

This thesis mainly considers the DAC dynamic linearity group of figures, since they are very important in the RF communication applications. Figure 2.2-5 shows an exemplary spectrum of a DAC sine wave output signal. DAC figures that are important for this thesis are indicated in the spectrum. SFDR (Spurious-Free-Dynamic Range), HD (Harmonic Distortion), and IMD (Inter-Modulation Distortion) are the most important figures that are further used in the thesis. In many places of the thesis, IMD, in case of a single sine wave input, is referred to as “folded” HD due to the similarities (the IMD components in that case are indeed HD components from the image signal bands).

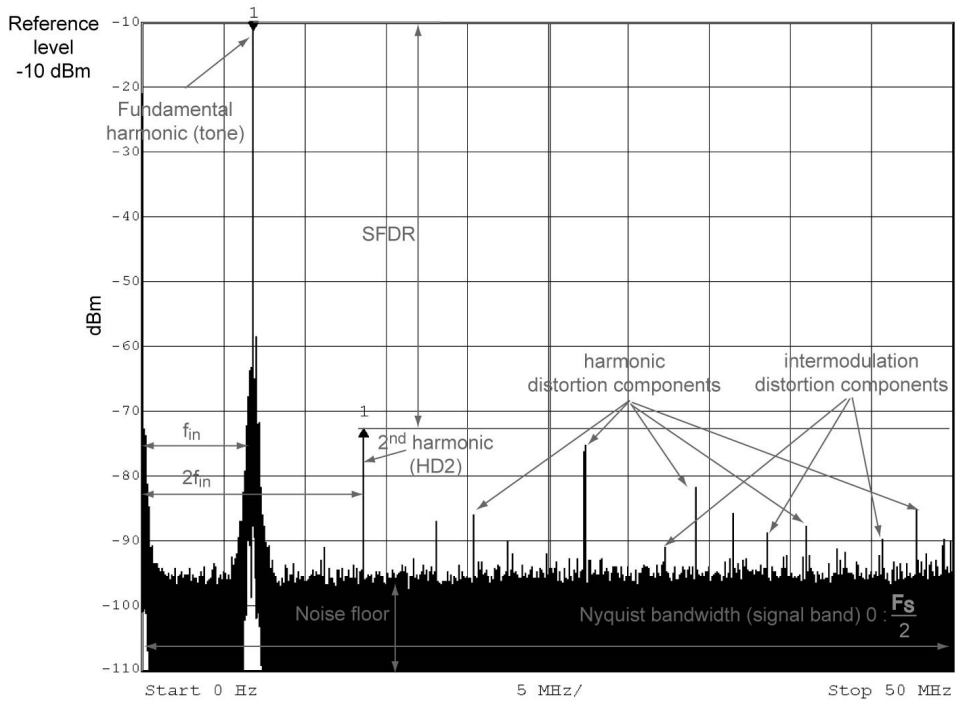


Figure 2.2-5. A DAC output spectrum, converting a sine wave signal.

2.3. DAC RESOURCES

To realize the D/A function, the DAC switches analog entities to the output, according to the digital input. Without loss of generality, the current-steering DACs are considered here as an example. Their analog entity is current and the switching is realized by data switching transistors that redirect this current. The current sources and the switching transistors represent the DAC analog resources that are controlled by the digital data. These data are provided by the DAC data control block, realized with the DAC digital resources that processes, synchronizes and prepares the DAC input digital data for the switching transistors. Figure 2.3-1 shows this mechanism in a simple generalized block diagram, where three main blocks are identified: the *digital data control block*, the *analog resources block*, and the *analog resources support block*.

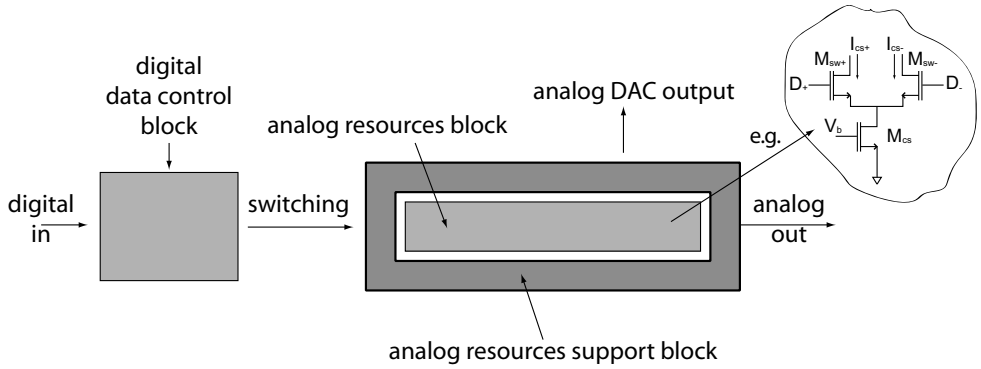


Figure 2.3-1. A generalized block diagram of a DAC, with an example of current-steering implementation.

The digital data control block may include digital input data buffers, digital pre-processing circuits, decoders, delay lines, synchronization latches, clock networks, data buffers, etc. The analog resources blocks may include current source transistors (shown as M_{cs}), cascode transistors, switching transistors (shown as M_{sw}), etc. Often these are referred to as switching current (SI) cells, e.g. [1], [5]. The analog resources support block may include biasing circuits, interconnection network, dummy units for matching purposes in the analog resources block, areas because of tolerance spacing for the circuits in the analog resources block, etc.

Generally, the size of the analog resources block is determined, given certain process technology, by three primary factors: required DAC full-scale (FS) current, required transistor matching, and required output resistance. Note that other factors may influence the size of the analog resources block, but these should be considered as secondary, since they are less important and/or application dependent.

For matching and output resistance, the current source transistors need to be in saturation. An approximation of the DAC FS current is the full current contribution of all SI cells, i.e.

Equation 2.3-1

$$I_{FS} = \sum_i \frac{1}{2} k \left(\frac{W}{L}\right)_i \left(V_{gs} - V_{th}\right)^2 \left(1 + \lambda V_{ds}\right) = \frac{1}{2} k \left(V_{gs} - V_{th}\right)^2 \left(1 + \lambda V_{ds}\right) \sum_i \left(\frac{W}{L}\right)_i ,$$

where I_{FS} is the DAC FS current, $k = \mu_n C_{ox}$ is CMOS process constant (μ_n is the electron mobility, C_{ox} is the oxide capacitance), V_{gs} is the gate-source bias voltage, V_{th} is the transistor threshold voltage, $\lambda = \frac{\Delta L}{L}\frac{1}{V_{ds}}$ is the channel-length modulation due to the drain-source voltage, W is the channel width, and L is the channel length. For fixed bias conditions V_{gs}, the I_{FS} is increased by increasing the ratio $\frac{W}{L}$.

The work of [6] considers the transistor matching, suggesting:

Equation 2.3-2

$$\left(\frac{\sigma_I}{\bar{I}}\right)^2 = \left(A_\beta^2 + \frac{4A_{VT}^2}{\left(V_{gs} - V_{th}\right)^2}\right)\frac{1}{2(WL)},$$

where $\bar{I}$ is the nominal current of a current source, σ_I is its standard deviation due to the tolerances of the CMOS fabrication process, A_β^2 and A_{VT}^2 are constants of the CMOS fabrication process. For fixed bias condition V_{gs}, the current accuracy is improved via increasing the product WL. The relationship between $\frac{\sigma_I}{\bar{I}}$ and DAC linearity is discussed in chapter 6.

The output resistance of the current source transistor is given by:

Equation 2.3-3

$$r_o = \frac{1}{\lambda I_d} = \frac{1}{\Delta L}\frac{V_{ds}}{I_d} L,$$

For fixed bias condition $\frac{V_{ds}}{I_d}$, the transistor output resistance is increased via L. To achieve the required r_o, a minimum L is specified. Note that the output resistance of the current source can be further increased with cascode transistors. For more detailed discussion, refer to [7], [1], [5], and [8].

The output capacitance C_o of the current source transistor is mainly given by the transistor drain-bulk capacitance C_{DB} [7]. It is proportional to W, the bottom-plate junction capacitance Cj, and the sidewall capacitance due to the perimeter of the junction Cjsw, see [7]. To achieve the required C_o, a maximum W is specified. Note that the influence of the c_o can be minimized with cascode transistors. For more detailed discussion, refer to [7], [1], [5], and [8].

Equation 2.3-1, Equation 2.3-2, and Equation 2.3-3 show that the size of the analog resources block depends on the DAC full-scale (FS) current, required transistor matching, and required output resistance. It does not depend on the architecture of the DAC, considered from the point of view of how the digital data control block switches the analog resources to the DAC output. However, the sizes of the digital data control block and the analog resources support blocks are determined by the DAC architecture.

2.4. SEGMENTATION OF DAC ANALOG RESOURCES

The DAC architecture includes the segmentation (division) of the analog resources in groups that are switched either on or off by the digital data control block to generate the DAC analog output. Often, the term “segmentation” is used for a special case of division into binary LSB part and unary MSB part. However, the term “segmentation” is used in its broader meaning as general “division” in this chapter.

As shown in chapter 6, the DAC INL depends on both the accuracy of the DAC current cells and the DAC architecture. However, the differences between the INL figures for DACs using the same current cells but different architectures are not large when high yield figures are considered. That is why a reasonable first-order approximation is that DAC INL depends only on the accuracy of the DAC current units. However, note that the DNL highly depends on the DAC architectures and its maximum is mainly determined by the switching of the largest DAC currents.

This section discusses the conventional DAC segmentation techniques. It provides important background information for the proposed new concepts in chapter 9. Thus, sub-section 2.4.1. discusses the analog resources segmentation in binary DACs. It is the most efficient segmentation. However, it is sensitive to errors related to the switching of currents. The current cells are nominally different and matching is difficult to achieve. This architecture also features no redundancy and hence no correction method can be applied. Sub-section 2.4.2. discusses the analog resources segmentation of sub-binary radix DACs. This segmentation trades off only little of the binary DAC efficiency for significant redundancy, which makes possible the application of DAC correction methods. However, the matching of the current switching is a major limitation for DAC high speed performance. Section 2.4.3. discusses the analog resources segmentation of the unary DACs. This segmentation uses nominally identical analog units to relax the matching requirements for their switching and hence to achieve good DAC high speed performance. The unary segmentation features significant redundancy and hence many

DAC correction methods are available. However, it requires a lot of hardware resources for the digital data control block and the analog resources support block. Section 2.4.4. discusses the LSB-binary and MSB-unary DAC segmentation. This architecture balances between the advantages and disadvantages of the binary and unary DAC segmentation.

Two types of segmentation should be considered and distinguished for the sake of a proper discussion: algorithmic segmentation and hardware segmentation. The algorithmic segmentation is a high-level concept that implies how the digital data control block switches the analog resources. The hardware segmentation is a low-level concept that implies how the analog resources are divided. The properties of the different forms of algorithmic segmentation concern the DAC performance and the sizes of the digital data control block and the analog resources support block. The properties of the hardware segmentation concern the DAC performance and only the size of the analog resources support block. The hardware segmentation needs at least to fully cover the algorithmic segmentation but may further introduce even a finer division of the analog resources. A notable example is the work of [9], which uses binary algorithmic segmentation for an efficient (area and power consumption) digital data control block and further uses LSB-binary and MSB-unary hardware segmentation to achieve good DAC high-speed performance.

2.4.1. Binary algorithmic segmentation

In terms of DAC occupied area and power consumption, the binary algorithmic segmentation is the most efficient segmentation. The binary DAC input word, after possible data synchronization and buffering, directly switches binary scaled groups of currents to the DAC output. That is why the sizes of the digital data control block and the analog resources support block are the smallest possible. For an N bit DAC, the analog resources are grouped in N-groups that are by a factor of 2. Figure 2.4-1 shows a block diagram of the binary segmented DAC.

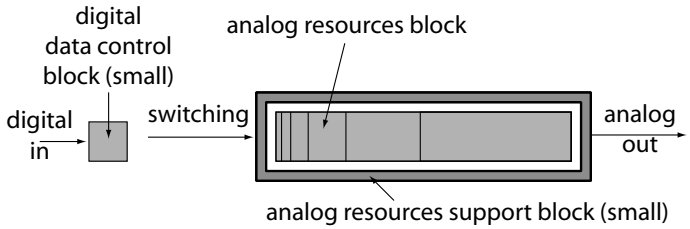


Figure 2.4-1. A generalized block diagram of a binary DAC.

A disadvantage of the binary segmentation is the high matching requirements for the binary scaled currents and their switching. Note that no DAC correction method is available for “only” binary DACs, since they feature no redundancy. If the switching responses of the SI cells are different, a data-dependent error charge that generates HD components is injected at the DAC output during switching. Since the largest errors are generated by the switching of the MSB current, i.e. at half-scale, the odd HD components are particularly dominant. The INL and DNL errors are also statistically expected to be large at half-scale. The work of [10] approximates the DNL error for an N bit binary DAC at half-scale as the standard deviation of the output transition, i.e.:

Equation 2.4-1

$$DNL_{\max} = \sigma(\Delta I) = \sqrt{2^N - 1}\left(\frac{\sigma_u}{I_u}\right),$$

where $\left(\frac{\sigma_u}{I_u}\right)$ is the relative matching of the unit current, and ΔI is the DAC output transition at half-scale [2].

To summarize, the binary segmentation features:

- very high efficiency and hence smallest possible digital data control block and analog resources support block;
- compromised high-speed performance, due to too high matching sensitivity for the MSB current cells;
- no redundancy due to its most efficient architecture and hence only limited correction methods are available through adding extrinsic redundancy.

2.4.2. Sub-binary radix algorithmic segmentation

The sub-binary radix DACs segment the analog resources in scaled groups by (a) factor(s) of less than 2. Looking at the DNL characteristic, many negative DNL errors are intentionally introduced to prevent large positive DNL errors even for very low matching of the currents. By doing so, they introduce enough redundancy in the DAC transfer characteristic for a DAC correction method to be able to improve the DAC linearity to the target level. The negative DNL errors are removed by a pre-processing DAC correction method, as explained in section 4.3.2. The digital data control and the analog resources support blocks are increased compared to the binary DAC architectures but they are still small compared to the other segmentation approaches.

The sub-binary-radix DACs always require correction that depends on self-measurement. However, the matching requirements of the analog resources are highly relaxed, which reduces the required WL area of the current source transistors and results in a small analog resources block. Balancing the extra area of the necessary correction method and the small area of the analog resources block, very small DACs can be designed, which can still achieve high static linearity [11].

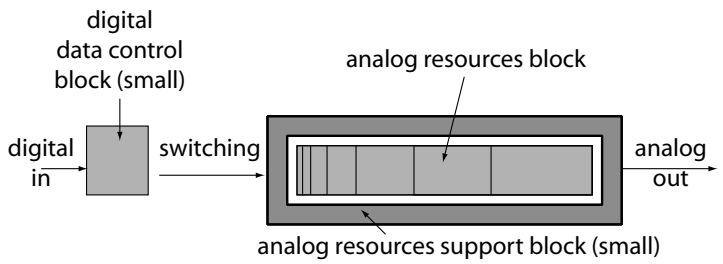


Figure 2.4-2. A generalized block diagram of a sub-binary radix DAC.

A disadvantage of the sub-binary segmentation is the problem of achieving the high matching requirements for the current switching, which may highly compromise the DAC high-speed performance. To realize the sub-binary radix scaling, the DAC design relies on matching of the scaled transistor widths, W. Unit element approach to realize the scaling cannot be used, as e.g. in the binary segmentation, since the scaling factor is not an integer number. Given the fact that the available correction methods for sub-binary DACs improve only the static DAC performance, high-speed performance is compromised. Thus, this segmentation is a good candidate for static applications, such as on-chip measurement and reference generation, e.g. [11].

To summarize, the sub-binary segmentation features:

- high efficiency and hence small digital data control block and analog resources support block;
- very low high-speed performance, due to the intrinsic mismatch sensitivity of the current switching;
- very high redundancy due to intentionally introduced negative DNL errors.

2.4.3. Unary algorithmic segmentation

The unary algorithmic segmentation divides the analog resources block in nominally identical unit elements, as shown in Figure 2.4-3. For an N bit unary DAC, there are 2^N-1 unary elements. Since the digital data control block switches nominally identical unit elements, their matching is relaxed and high DAC performance can be achieved even at very high speeds. However, the digital data control block need to control 2^N-1 unary elements, which for high N, is a much larger number, if compared to the only N elements in the case of the binary segmentation. Therefore, the digital control block is very large. It requires significant area and high power consumption. The analog resources support block is large, too. Since there are 2^N-1 groups in the analog resources block, to support them, the analog resources support block requires a lot of area for interconnections, distances between the transistors, dummy elements for matching, special wiring and placement techniques to compensate for the on-chip gradients, etc. The unary segmentation features very high redundancy and hence many correction methods are possible, e.g. mapping (section 4.3.1.) and DEM (section 3.5.).

To summarize, the unary algorithmic segmentation features:

- large digital data control block and analog resources support block and hence low efficiency;
- high performance at high speeds, due to the relaxed matching sensitivity of the unit current cells and its simple layout implementation though unit elements;
- very high redundancy due to the large numbers of possible combinations to form the DAC output for a given input code.

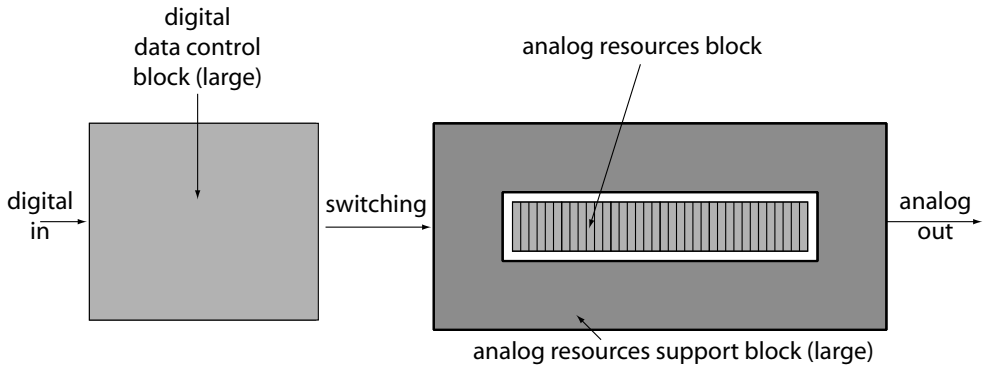


Figure 2.4-3. A generalized block diagram of a unary DAC.

2.4.4. Binary LSB and unary MSB algorithmic segmentation

To balance the advantages and disadvantages of the binary and unary DAC segmentation, the usually applied approach is to implement the DAC LSB part in a binary way and the DAC MSB part in a unary way, shown in Figure 2.4-4. The digital data control block and the analog resources support block are kept within reasonable limits,

while the high-speed performance advantages of the unary segmentation are applied to the most-significant DAC part. The redundancy of this approach is derived from the redundancy of the unary MSB part and hence various correction methods are possible. The work of [12] discusses how to allocate the analog resources in the binary and unary parts in the most optimal way.

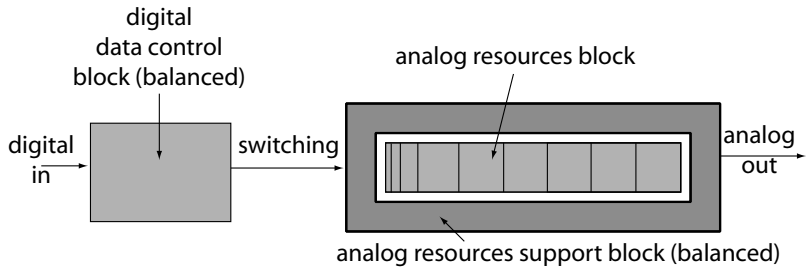


Figure 2.4-4. A generalized block diagram of a segmented binary-unary DAC.

To summarize, the binary LSB and unary MSB DAC segmentation features:

- balanced efficiency and hence balanced digital data control block and analog resources support block;
- balanced to high high-speed performance, due to the relaxed matching sensitivity of the MSB current cells;
- sufficient redundancy is thanks to the different switching sequences for the MSB unary currents.

2.5. DAC IMPLEMENTATIONS

The described functionality in section 2.2. and algorithmic segmentation in section 2.3. can be implemented with different electronic circuit solutions. The most popular implementations include switched-capacitor DACs, resistor-based DACs and current-steering DACs.

The charge-redistribution DAC is a switched-capacitor (SC) circuit, implementing DA conversion in the charge domain [6], [3], [13]. Usually, charges stored on a number of capacitors are used to perform the required conversion. Figure 2.5-1 shows an example of a differential charge-redistribution DAC. Its output signal is generated by an amplifier, the speed and the linearity of which are usually the main performance limitations. Furthermore, the performance of these converters is also constrained in accuracy due to the finite matching of the capacitors.

The R-2R ladder is a simple approach to implement DACs. Its basic principles is outlined in Figure 2.5-2a. When a voltage is applied to node 6 in the circuit, a binary voltage scale builds up along the upper nodes. The same applies to the currents flowing in the vertical resistances 2R [14].

The binary weighted currents flowing through the vertical resistances 2R can be selected and combined in a common node and consequently converted to voltage, for example as shown in Figure 2.5-2 b). Switches, which are controlled by the input digital word, B_n, pass the binary weighted currents to the summing node or redirect them to ground. The summing node in the example, Figure 2.5-2 b), is the virtual ground at the negative input of the amplifier [15]. Today, The R2R ladder approach is rarely encountered in state-of-the-art DACs.

The current-steering DAC is the most popular implementation, mainly because of its high speed, see APPENDIX A. It is the implementation that is considered in this thesis. The main building blocks of a current-steering DACs are discussed in section 2.6.

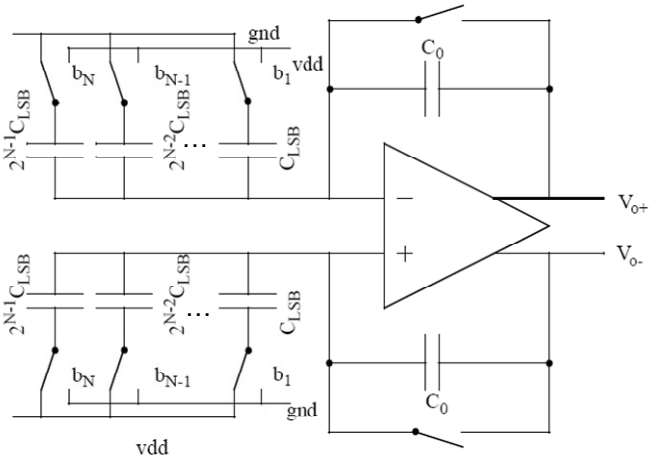


Figure 2.5-1. Exemplary N bit charge redistribution DAC, without reset phase.

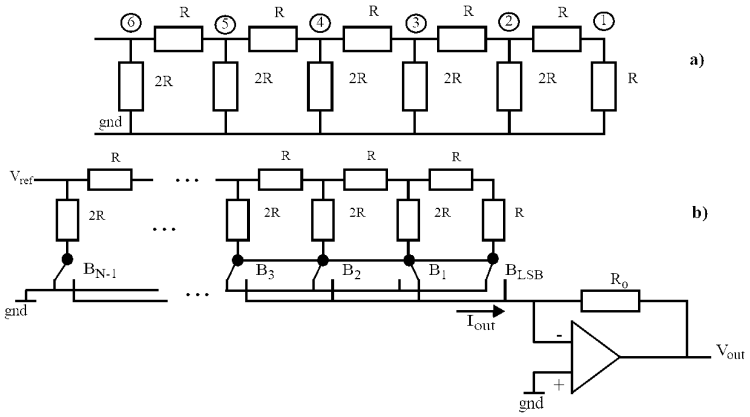


Figure 2.5-2. a) R2R ladder; b) R2R ladder based DAC.

2.6. CURRENT-STEERING DAC ARCHITECTURE

The current-steering DAC implementation provides high-speeds and high-accuracies with respect to the other main implementation alternatives. The basic analog entity is current and hence the DAC output is usually current. The output DAC current can quickly charge parasitic capacitances at the output and hence achieve high speeds. Matching of currents is a well elaborated topic and many techniques exist to achieve higher accuracy and matching of currents than what is possible and reasonable with the other possible alternatives.

Figure 2.6-1 shows a block diagram of a representative current-steering DAC implementation. Usually, the input of the DAC complies with the LVDS (low-voltage-differential signal) standard, which provides high digital input data rate. Thus, the LVDS input buffers are the first circuits that process the input signals. Further, the N bit differential signal is split into B bits LSB and (N-B) bits MSB parts, if the algorithmic

segmentation requires this. The MSB bits are encoded into $2^{(N-B)}$-1 bits of thermometer (unary) code by a binary-to-thermometer decoder. The LSB bits are delayed by a series of buffers with an approximately equal delay to that of the decoder. Equalizing the moments when the information appears at the inputs of the flip-flops is important, since the slowest time difference between the fastest and the slowest bits would limit the maximal possible sampling rate. The decoder usually works asynchronously. Thus, the information needs to be again synchronized after it. The advantage of such an approach is that the implementation of the decoder is decoupled from the implementation of the flip-flops, e.g. the type of logic (CMOS, Differential-NMOS, Current-Steering, etc.). A disadvantage is that an asynchronous decoder always allows slower speeds than a synchronous one.

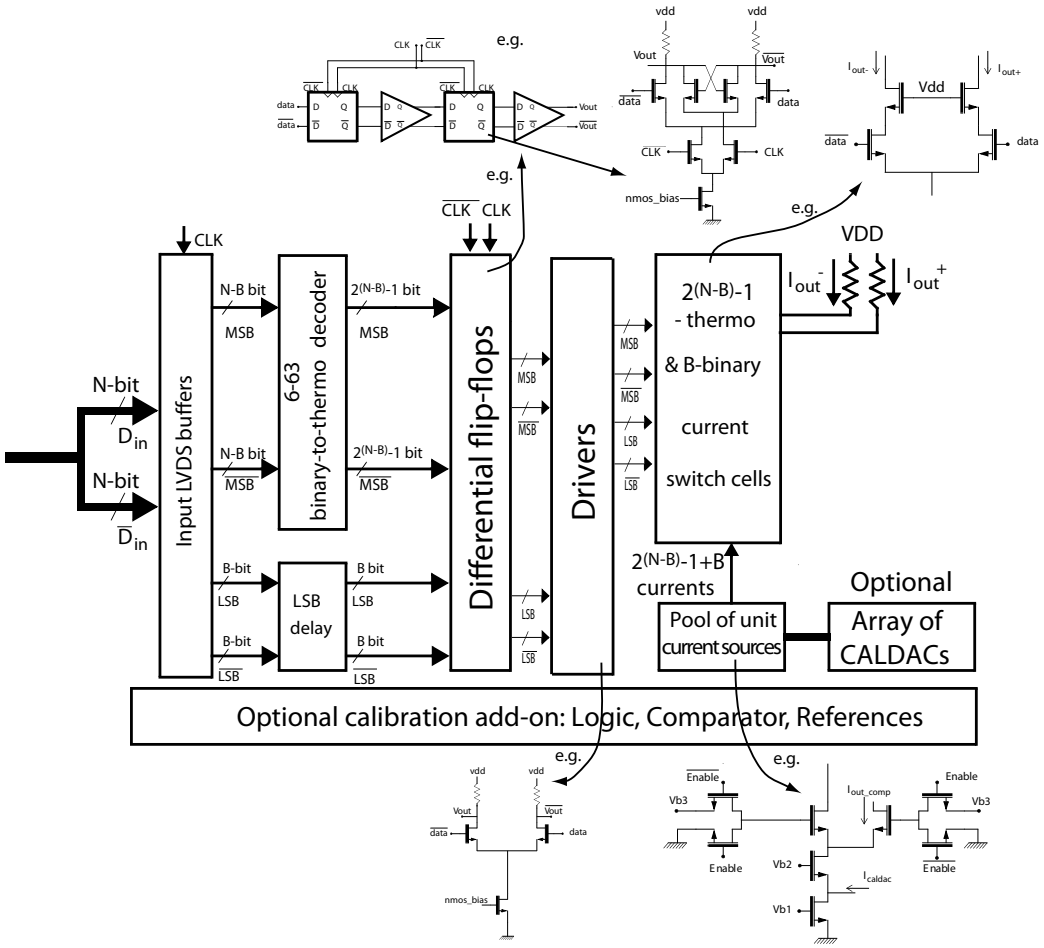


Figure 2.6-1. A block diagram of a Current-Steering DAC, with binary LSB – unary MSB algorithmic segmentation, exemplary circuit implementations, and an optional exemplary calibration engine.

Next, after the binary-to-thermometer decoder, the signals are directed to flip-flops, usually implemented as Master-Slave latches. These flip-flops are referred to as "analog", because of their sensitivity and importance to the quality of the generated converter output. Thus, all physical processes should be considered at a very low, analog level. The electrical signals of the carried digital information are of significant importance. The Master-Slave Latches synchronize the data and shape their physical signals in a proper

way for the next block - the current switching cells. The main task of the current switching cells is to combine the appropriate currents, generating the analog current output for the input digital word being converted. The currents, or the analog elements, which are used to generate the analog output current, are placed in the block "Pool of (unit) current sources". The currents generated by this block may be optionally corrected by currents coming from the Array of CALDACs (calibrating DACs). The calibrated currents are passed to the current switching cells to generate the output of the converter.

Next to the above described core of the DAC, there are optional components, needed for example to perform calibration, like comparators and references. With respect to the operation, the DAC sub-blocks can be classified as dynamic and static blocks. The dynamic circuits are those which process dynamic signals, i.e. data and clock signals. The static circuits are those which operate with static analog values (currents and voltages) during the normal operation of the converter. The dynamic group includes input LVDS buffers, decoder and LSB delay line, flip-flops, and current switches. The static group includes the pool of current sources, the optional array of CALDACs, and biasing circuits (not shown).

2.7. MODERN CURRENT-STEERING DAC CHALLENGES

Modern DACs only partially benefit from the recent developments of the CMOS IC processes. While their speeds, e.g. sampling rate F_s and utilized signal bandwidth, are expected to continue rising, their accuracy, occupied silicon area and production costs are expected to remain problematic design bottlenecks. The DAC sampling rates directly benefit from the scaling of the CMOS technology, because the digital circuits and clock networks mainly depend on factors that are improved with transistor scaling, e.g. switching speeds and reduced parasitic capacitance. However, the DAC analog circuits cannot benefit that much. For example, the transistor matching deteriorates with the reduction of both the available voltage headroom and the transistor sizes. More general details on the consequences of CMOS technology development on both the digital and analog IC design can be found in [16].

For an illustration of the technology development so far, Table 2.7-1 shows an example of how the performance of some selected state-of-the-art CMOS DACs evolves in the years.

Table 2.7-1. Development of the Digital-to-Analog Converters

Reference	Year	CMOS process	Speed	Resolution	Area
[17]	1988	2000nm	27MS/s	8b	3.4mm^2
[18]	1991	1000nm	130MS/s	8b	0.5mm^2
[12]	1998	350nm	500MS/s	10b	1mm^2
[19]	2004	180nm	1.4GS/s	14b	6.2mm^2
[20]	2009	65nm	2.9GS/s	12b	0.3mm^2
Projection	~2020	~10nm	~16GS/s	~14b	~0.1mm^2

The progress of F_s (DAC sampling rate) follows the progress of the CMOS process. The accuracy and the efficiency of the occupied silicon area do not improve at such a straightforward rate. Their improvement is mainly attributed to improved design techniques, new DAC correction methods, and general knowledge of the DAC system over the years. For example, consider the works [12] and [20] (both works belong to the

same people from the same company). For about 11 years and 5 times scaled-down CMOS process, the main DAC improvement is the speed (6 times), while the improvements in the resolution (two more bits, i.e. 4 times increase of the convertible codes) and the occupied area (3 times) follow at a slower rate. Considering these two works the overall progress is thanks to the combination of the CMOS process advancement, new correction techniques, and design experience. Similar observations for the development of the Analog-to-Digital Converters (ADCs) are made in [21] and [22].

A few fundamental limitations of the IC processes hardly allow higher intrinsic accuracy or smaller occupied silicon area than the quoted numbers in Table 2.7-1. The transistor mismatch, parasitic capacitances, and non-linear switching are among the major limitations.

In current steering (CS) DACs, the transistor mismatch limits the accuracy of the signal and bias current sources. These tolerances translate to mismatch among the parallel current cells, causing DAC static and dynamic non-linearity. For good transistor matching, the transistors need to be made big and laid out close to each other. For a current source transistor, Equation 2.3-2 shows the relationship between the transistor area $(WL)_{\min}$ and the relative matching $\sigma_I / \bar{I}$ of the current. Thus, the intrinsic transistor accuracy depends on three main factors: biasing, technology parameters, and transistor size.

To translate the current accuracy into DAC specifications, [23] and chapter 6 show the relationship between the relative deviation of the current and an N bit unary DAC static linearity in the form of the expected $INL_{\max}$:

Equation 2.7-1

$$\mathbb{E}\left(INL_{\max}^{(Thermo)}\right) = \left(\frac{\sigma_I \sqrt{2^N - 1}}{\bar{I}}\right) \cdot \frac{1}{2}\sqrt{2\pi}\ln 2 \approx 0.869 \cdot \left(\frac{\sigma_I \sqrt{2^N - 1}}{\bar{I}}\right)$$

From Equation 2.3-2 and Equation 2.7-1, it can be concluded that to increase the intrinsic CS DAC resolution and accuracy by just one bit, the occupied area of the signal current source transistor needs to be enlarged at least four times. Both requirements for transistor matching, i.e. units that are big and laid out close to each other, are contradictive when the DAC linearity needs to exceed 10-12 bits, i.e. 1023–4095 unit currents. Simply, the transistors are too big and too many to be close to each other. In addition, the long layout distances may increase both the random and systematic mismatch errors. Equation 2.3-2 does not model this phenomenon and hence the chip yield risks deteriorate.

The transistor mismatch is a source of timing errors, too. The DAC synchronization network includes the clock network, the synchronization latches, the data buffers, and the current switches, further discussed in section 2.6. At all these nodes, transistors are used to switch signals. Usually, these transistors need to be fast and hence small. Therefore, these transistors have poor matching and they contribute to the timing mismatches in the switching moments of the DAC current cells. The work of [1] provides an in-depth analysis of the impact of the timing errors on the DAC performance.

Other performance limitations include clock-feedthrough, data-feedthrough, data-dependent disturbances of the substrate and power rails, systematic parasitics due to the layout, output glitches, etc. When these errors are data-dependent, they cause harmonic distortion (HD) of the input signal and hence limit the DAC linear performance.

The need to guarantee the DAC performance further increases the production costs through the cost of test [21]. For the analog DAC output, the cheap and fast digital testers

cannot be used. Analog testing may be too expensive for mass production. Therefore, the DAC accuracy is often massively over-designed but yet not guaranteed. The DAC accuracy still depends on the manufacturing technology. The migration to other technologies remains difficult. The redesign effort is considerable. The production risks are high.

However, various correction methods are available to counteract these performance limitations. These correction methods may support the DAC performance in various ways, e.g. improve overall intrinsic performance, improve chip yield, relax and improve particularly targeted design specifications. Moreover, the evolution of the IC technologies favors the development of sophisticated correction methods, since the chip co-integration price per function becomes low. This argument is particularly plausible for digital correction methods and introduces the trend of *digitally assisted analog performance*, see [24], [25].

2.8. SUMMARY

Basic background knowledge on DAC functionality and specifications is discussed. Important DAC concepts that are relevant for this thesis are presented. An in-depth generic discussion on the algorithmic aspects of the DAC segmentation is provided. It is shown that the binary algorithmic segmentation is the most efficient but provides no redundancy and compromises DAC performance, while the unary algorithmic segmentation provides redundancy and high DAC performance but features low efficiency. The resistive ladder, switched-capacitor and current-steering DAC implementations are briefly reviewed. The basic building blocks of the current-steering DAC architecture are discussed. The analog aspects of modern microelectronics, e.g. matching, are argued as main challenges for modern current-steering DAC design.

PART II: STATE-OF-THE-ART CORRECTION METHODS

This part includes chapters that provide an overview on the state-of-the art DAC correction methods. Chapter 3 discusses DAC correction methods that do not use self-measurement of errors. Chapter 4 discusses DAC correction methods that use self-measurement of errors.

3 . C h a p t e r

ERROR CORRECTION BY DESIGN

This chapter discusses published DAC correction methods that do not use error measurements to improve the DAC performance. It overviews main state-of-the art mechanisms to reduce DAC expected errors. Recent publications are referenced to support the overview.

The discussed methods are: return-to-zero (RZ), differential-quad switching, switching with cascode transistors, and digital data reshuffling methods (DEM). Their main correction targets and associated particular advantages and disadvantages are overviewed. General discussion is provided on the main principles of the discussed methods. These principles are further used in chapter 7 to propose a general classification of the DAC correction methods.

3.1. INTRODUCTION

The open literature provides various DAC correction methods that can improve the DAC performance without relying on information about the exact DAC errors. A main conceptual argument for these methods is avoiding self-measurement circuits that may deteriorate the intrinsic performance of the DAC core. These methods rely on good understanding about the error mechanism to minimize the effects of the errors on the DAC performance.

This chapter discusses a selected set of state-of-the-art DAC correction methods that do not use self-measurements. That is to say that these methods illustrate state-of-the-art principles of improving performance without using exact error information. Some of these principles can be used together or in combination with correction methods that use exact error information (chapter 4). Section 3.2. discusses the return-to-zero (RZ) method. Section 3.3. discusses the differential-quad-switching (DQS) method. Section 3.5. discusses the Input data reshuffling (dynamic element matching) method. A general discussion is provided in section 3.6. . Finally, conclusions are drawn is section 3.7. .

3.2. RETURN-TO-ZERO OUTPUT

To protect the DAC output signal from the data dependent errors that occur during the switching of the current cells, the return-to-zero (RZ) method 'hides' the switching moments from the DAC output signal. Notable examples of DAC designs using RZ are [26], [27], and [28].

During the switching transition time of the current switch transistors, the DAC output is set to zero, i.e. it is in mode 'zero'. Once the data is settled, the DAC output is switched back to normal 'track' mode. The control signals and the DAC output signal are shown in Figure 3.2-1. NRZ (non-return-to-zero) output shows what the DAC output would be if RZ is not applied.

The DAC NRZ output features data-dependent glitches and settling behavior due to the switching transients. During this time, the DAC RZ output is always at zero. The main difference between NRZ and RZ switching errors is that the NRZ transition errors are (predominantly) data-dependent while the RZ transition errors are (predominantly) clock-dependent.

Exemplary implementations of the RZ method are shown in Figure 3.2-2. The work of [26] implements the RZ method with a DAC output stage. This is an example of a high-level RZ method implementation. It is effective against both random and systematic data-switching errors, since the analog signal is reset at the DAC output during the sampling moments. That is to say that during the 'Reset' phase, the signal power is lost.

Let the Track and Zero periods be equal, then half of the DAC output signal power is practically lost, equivalent to about 6dB reduction of the main output signal. To avoid losing signal power, the work of [27], shown in Figure 3.2-2b, suggests to use two parallel RZ sub-DACs A and B, with time-interleaved 'Reset' intervals. Further, the RZ method is implemented at low level, i.e. at the current cell level and while sub-DAC A is in 'Track' mode, sub-DAC B is in 'Zero' mode, and vice-versa. Effectively, using 2 parallel sub-DACs is suggested. These two sub-DACs have half a sampling period shifted Track and Zero phases and hence the lost signal power is compensated. Yet, the power efficiency of the whole system remains the same, since two parallel sub-DACs are used to generate the equivalent signal power of one. This solution is suggested for low-voltage designs, where 6dB of main signal power is more important than the unused power of I_u during the 'Zero' phase.

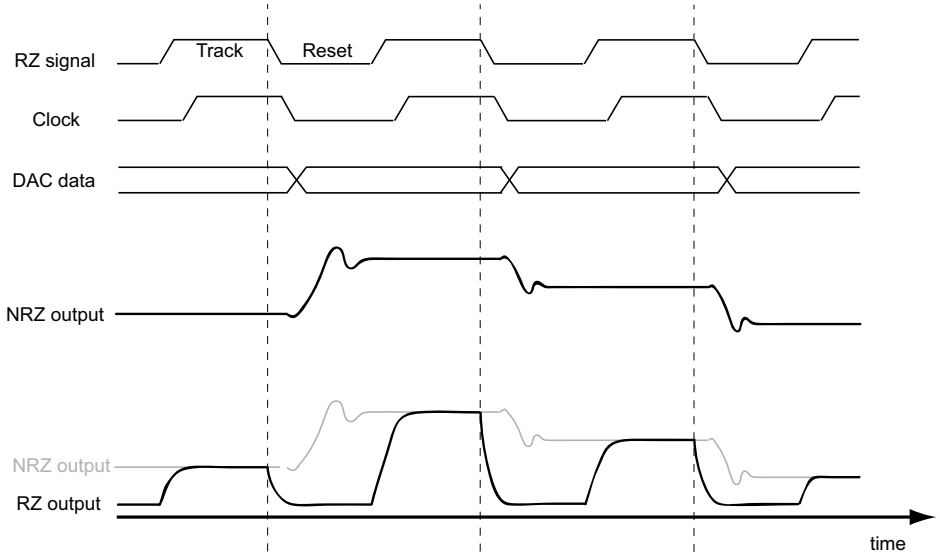
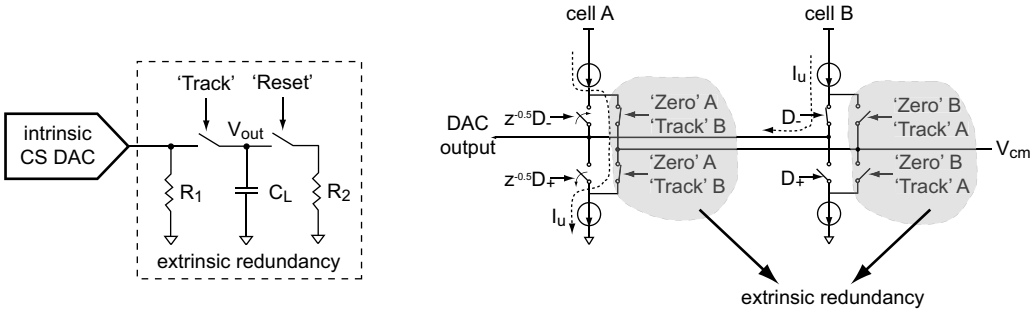


Figure 3.2-1. Control signals and DAC output for the Return-to-Zero (RZ) method, based on [28].

Another essential feature of RZ methods is the equivalent sampling rate. The equivalent sampling rate of the DAC RZ output is $2F_s$, i.e. twice higher than the DAC sampling rate. Thus, the DAC RZ output needs to deal with problems of twice faster speeds than the sampling speeds of the DAC core. At high DAC speeds, when the Track and Zero periods become too short, the DAC output current cannot properly settle. Thus, the 'Reset' effect is lost and the settling is data-dependent.



a) DAC RZ output, see Bugeja, JSSC 1999 b) RZ current cell with DAC NRZ output, see Clara, ESSCIRC 2008

Figure 3.2-2. Exemplary implementations of the RZ method: a) with DAC output stage [26], b) within the current cells [27].

The RZ method corrects for the switching-related data-dependent disturbances of the DAC current cells by forcing the output signal to the 'zero' level. Thus, the RZ method introduces extra information in the system – the DAC output signal is being reset with the clock frequency. The reset circuits represent extrinsic DAC redundancy. Therefore, there is a risk of deteriorating the intrinsic DAC performance. For example, any non-linearity of the reset circuits directly contributes to the overall DAC non-linearity. That is to say that the RZ method is essentially an analog correction. Therefore, the analog accuracy

requirements of the correction need to be considered, e.g. clock jitter effects for $2F_s$, non-linearity and accuracy, layout parasitics effects, etc. The linearity of the reset circuits should at least match the linearity of the intrinsic DAC core. Since the reset circuits directly connect to the DAC output analog signal wires, their errors contribute to the overall DAC errors. For example, the work of [26] employs switching bootstrapping techniques to reduce the non-linearity of the 'Track' and 'Zero' switches, as shown in Figure 3.2-2a.

For the cases when the DAC output signal power is not compensated by a parallel RZ sub-DAC, an inherent advantage of the RZ methods is that they relax the "sin(x)/x" DAC filtering function (that is due to the hold effect of the sampled output analog value). Thus, the DAC attenuation for high input frequencies approaching $F_s/2$ is pushed beyond $F_s/2$, because the effective sampling frequency is twice higher, i.e. 2Fs.

Based on the arguments presented above, the RZ methods are appropriate to DAC architectures that are sensitive to data-dependent switching errors, e.g. binary DACs and DACs using CMOS data drivers. RZ methods can be as well used to efficiently realize a twice higher sampling rate than the sampling rate of the DAC core and hence to reduce the suppression of the DAC inherent "sin(x)/x" filtering function. The RZ methods are not effective for extremely fast DACs for a given IC technology node, since the RZ output cannot properly settle.

3.3. DIFFERENTIAL-QUAD SWITCHING

The current switching disturbs the current cell and hence it behaves differently during the periods of presence and absence of switching. The introduced disturbance is data-dependent and it includes voltage disturbances at the common source node of the current cell, charge feed-through, disturbances on the power supply and bias wires. Particularly, at high speeds, these error mechanisms cause the generation of DAC HD.

The differential-quad switching (DQS) technique replaces the data-dependence characteristic of this disturbance with the dependence on the sampling clock. Examples of this technique can be found in [19] and [29]. The DQS conceptual circuit-level idea is illustrated in Figure 3.3-1. Instead of two differential current-switching transistors, four switches are used M_{13a}, M_{14a}, M_{13b}, and M_{14b}. At every clock sample, two transistors are being switched. That is how the introduced disturbance is repeated at every clock sample, regardless of the digital data.

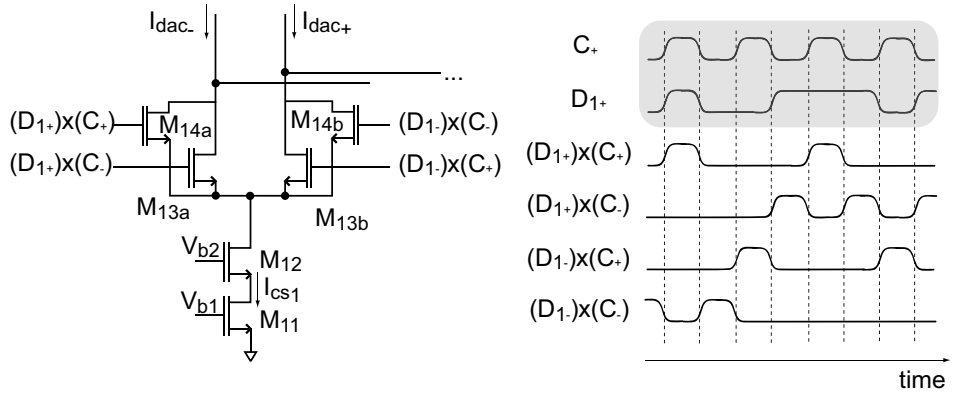


Figure 3.3-1. Differential-quad switching, from [29].

Though the introduced switching repeats at every clock sample, the amount of this disturbance is still data dependent due to some second-order effects. Usually, these

effects involve some relations to other error mechanisms. For example, the modulation of the common source voltage by the output signal in Figure 3.3-1 influences the amount of the injected disturbances, making them data dependent. Another example is the transistor mismatch. The matching between the switch transistors M_{13a}, M_{14a}, M_{13b}, and M_{14b} is important for the matching of the introduced disturbances. The matching requirements concern both the amount of error charge and the timing of the error charge. However, note that these transistors need to be small due to the DAC speed requirements and hence their mismatches would be considerable.

In comparison with the conventional DAC switching, the DQS DAC correction technique introduces two extra current switches, adding implicit redundancy in the design. The DQS DAC correction technique repeats the switching disturbances every clock cycle. Thus, the DAC uses extra information for its switching-related errors during all clock periods when data is not being switched. The DQS DAC correction technique does not need error measurements, which makes it simple to implement. The correction process (actuator) can be considered as compensation in the analog domain, which makes it sensitive to transistor mismatches.

3.4. CASCODE SWITCHES WITH OFFSET CURRENT

To reduce the influence of the DAC output signal (represented as a voltage drop across the load resistors) on the current cells, cascode transistors connected to the drains of the current switching transistors can be used. These cascode-switch transistors are shown in Figure 3.4-1 as M_{cas1}+ and M_{cas1}-. The original current cell is constructed by the current source M_{11}, its cascode transistor M_{21} and the two switches M_{sw1}. An in-depth analysis of the cascode-switches application in DAC design can be found in [5].

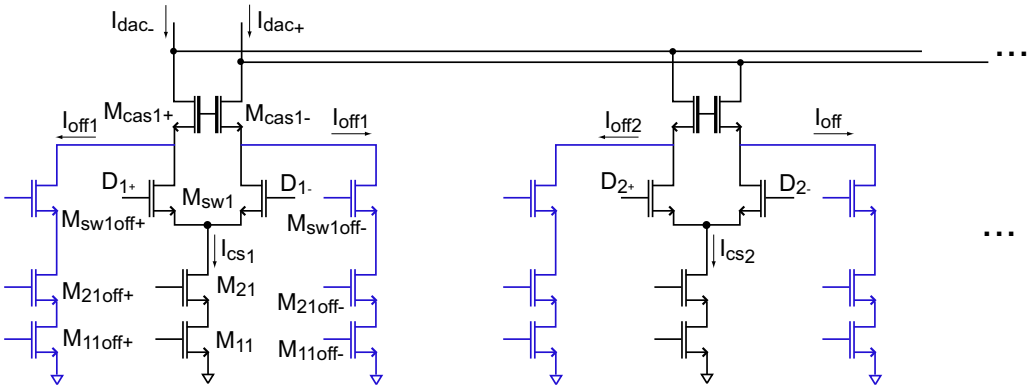


Figure 3.4-1. Current cells with cascode switches and offset currents.

The cascode transistors operate in a common-gate circuit configuration. Thus, these switch currents and do not rely on voltage-to-current switching. Therefore, the switch-cascode transistors should not limit the switching speed of the current cell. The switch-cascode transistors reduce the signal dependent voltage modulation on the drains of the switch transistors. The switch-cascode transistors also improve the output impedance of the whole current cell.

To improve the switching speed of the cascode transistors and to reduce even further the output-signal dependent switching of the current cell, the work of [20] suggest to use offset currents (M_{11off}) that do not allow turning off the cascode switch transistors when the respective switch transistors (Msw1) are off.

This method is very suitable for advanced CMOS technologies where both thin and thick oxide transistors are available. Usually, the thin oxide transistors are allowed to operate only with low voltages (e.g. 1V) and the thick oxide transistors are allowed to operate to somewhat higher than these voltages (e.g. 2.5V). Often, the DAC chip needs to be referenced to high off-chip voltages (e.g. 2.5V) and hence the DAC output needs to be implemented anyway with thick oxide transistors. Thus, the voltage headroom between the low "thin oxide transistor" voltages and the high "thick oxide transistor" voltages (e.g. from 1V to 2.5V) can be used for the cascode transistors M_{cas1}.

3.5. INPUT DATA RESHUFFLING METHODS (DEM)

For many DAC applications, the primary concern is the DAC SFDR (spurious free dynamic range). Particularly, sigma-delta converters require low levels of DAC HD (harmonic distortion) in order to correctly shape the errors to higher frequencies and achieve high dynamic range in the band of interest. However, the transistor mismatches create amplitude and timing errors and hence cause HD components that limit the DAC dynamic range. The input data reshuffling methods, also often referred to as Dynamic Element Matching (DEM) methods, average the mismatch errors of the DAC current cells over time and hence remove their data-dependence. That is why the mismatch errors cause noise-like disturbances and not HD.

The input-data reshuffling methods require specific DAC intrinsic redundancy of a plurality of nominally identical elements. Usually, these elements are the DAC unary current cells. However, parallel sub-DACs can also be used, because at a system level they are also nominally identical. Recently, several publications address the problems of DEM techniques applied to parallel sub-DACs, e.g. the segmented DEM technique in [30], [31], [32], [33]. Such an intrinsic DAC redundancy provides the possibility of generating an analog DAC output for a given digital DAC input using different combinations of the nominally identical elements. The input-data reshuffling methods randomize in time the choice of these combinations.

For example, to convert the input digital word $W[nT]$, w unary elements should be used from a set of total 2^N-1 (i.e. N bits decoded in a unary way). The number of possible combinations to choose these w unary elements is:

Equation 3.5-1

$$ {}_{(2^N-1)}C_w = \begin{pmatrix} (2^N-1) \\ w \end{pmatrix} = \frac{(2^N-1)!}{w!(2^N-1-w)!} $$

For each combination C_j, there is a particular error charge Q_{e_j} which is a sum of the error charges of the chosen unary elements. As j is randomly chosen for the sampling moment nT, the error charge $Q_{e_j}[nT]$ should also be random. Thus, the dependency is reduced between the input data $W[nT]$ and the DAC error charge $Q_{e_j}[nT]$. Note that the error charge $Q_{e_j}[nT]$ can contain both timing and amplitude error charges and hence DEM methods can be effective for both low and high DAC speeds.

Conceptual schemes of low- and high level implementation of DEM methods in DACs is shown in Figure 3.5-1, respectively a) and b). The input data reshuffling method is traditionally used for the DAC MSB part, which is implemented with unary current cells. This approach is conceptually shown in Figure 3.5-1a (top). In the segmented DAC

architectures after the binary-to-thermometer decoder a scrambler block can be placed. It receives digital data coded in a thermometer way. The scrambler block randomly reshuffles this data, while still preserving the number of zeros and ones. In theory, there is no difference for the DAC output when the DAC is modeled with no mismatch errors. In practice, the DAC output differs depending on the particular mismatch errors Q_{e_j} that are associated with the particular chosen map, i.e. combination C_j, for the unary current sources. Providing that C_j is chosen randomly at every clock sample, the error charge averages in time. On one hand, this technique can only be applied to the unary current sources and hence the error of the LSB current cells would still contribute to the DAC output. Thus, the effectiveness of this method depends on the DAC architecture: the more DAC MSB cells are implemented in a unary way, the more effective the data-reshuffling technique is. On the other hand, the larger the DAC unary MSB portion is, the exponentially larger the random reshuffling block is. Note that the random reshuffling block needs to create all possible interconnections between its unary input and its unary output. Thus, its complexity increases "super-exponentially" (an increment within the factorial *((x+1)!)*) for each extra input bit more that is implemented in a unary way. That is why the implementation of this technique is usually limited to low-resolution unary DACs, e.g. sigma-delta DACs.

To overcome this limitation, the work of [30] suggests the segmented DEM technique. This technique introduces redundancy via doubling each current source, i.e. creating parallel sub-DACs. Effectively, [30] uses two binary 13 bit sub-DACs to achieve 14 bit performance with segmented DEM. At system level, the DEM concept can be generalized as shown in Figure 3.5-1b (bottom). The respective reshuffling block randomly selects the sub-DAC input words, observing however that their sum is always equal to the DAC digital input word.

The main concern for the input-data reshuffling methods is the amount of resources that are invested for the realization of the digital reshuffling function. There are two main aspects for this issue: occupied silicon area and power consumption. As shown for the unary currents DEM methods, the complexity increases with each extra binary bit decoded in a unary way. Concerning the segmented DEM techniques, the design requirements are relaxed but still a careful consideration is needed, keeping in mind that the random sub-DAC input data generation block operates at clock frequencies, which suggest increased power consumption.

Another concern is the increase of the noise floor, since the power of the HD components is transformed to noise. However, an improvement of the SNDR (Signal-to-Noise-and-Distortion-Ratio) is still expected even for Nyquist DACs though some rise in the noise floor, as can be seen from the plots of [30].

The input data-reshuffling methods use the DAC intrinsic redundancy. The intrinsic redundancy can be either at low level – unary current cells or at high level – sub-DACs. In both cases, the redundant element can be used for D/A conversion independently from the correction method. The correction of the input data-reshuffling methods is completely in the digital domain. These considerations are further used in the proposed classification of DAC correction methods in chapter 7.

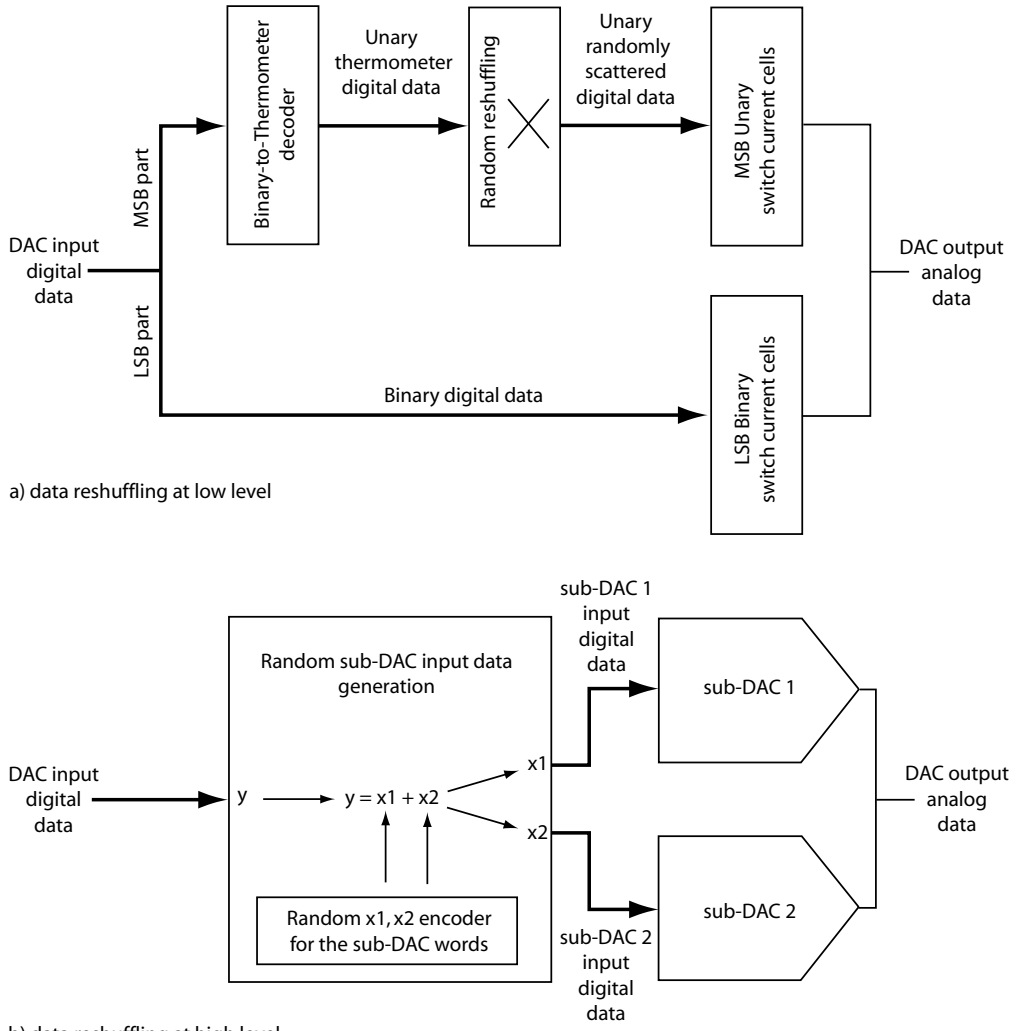


Figure 3.5-1. Conceptual schemes of data reshuffling: a) low level; b) high level;

3.6. DISCUSSION

This chapter demonstrates the high diversity of DAC errors and the DAC correction methods. There is no correction method that can counteract all DAC errors. Rather the various DAC correction methods target a particular set of errors and their effects on the D/A function. Further in this thesis, chapter 7 introduces a classification of the DAC correction methods to systemize them, to identify missing methods, and to derive clues to fill up these gaps.

Table 3.6-1 shows a short summary of the discussed DAC correction methods, ordering them according to four discussed classes: using intrinsic or extrinsic redundancy and implemented on low- or high level. Three notations are used: “Yes” when the method exist within a class; “No” when the method does not exist within a class; and N/A when the method cannot exist within a class (e.g. if it is against its definition). Note that all boxes indicated with “No” may be eventually filled-in with methods that do not exist in the open literature, yet.

Table 3.6-1. Summary of the discussed DAC correction methods.

Correction method	***Classes***			
	Intrinsic redundancy	***Extrinsic redundancy***	***Low-level***	***High-level***
Return-to-zero	*No*	***Yes***	***Yes***	***Yes***
Differential-quad switching	***Yes***	*No*	***Yes***	*No*
Turned ON cascode switches	*No*	***Yes***	***Yes***	*No*
Input data reshuffling (DEM)	***Yes***	*No*	***Yes***	***Yes***

3.7. CONCLUSIONS

An overview of DAC correction methods that do not use self-measurement is presented. The overview is based on state-of-the-art works that are recently published. The main aspects of each correction method are discussed. It is shown that each particular correction method can address a specific set of DAC errors. There is no “universal correction” for all errors. It is also argued that applying a DAC correction method is a matter of a trade-off, since each correction method brings its own particular advantages and disadvantages. Further in the thesis, chapter 7 proposes a classification of DAC correction methods.

4. Chapter

SMART SELF-CORRECTING D/A CONVERTERS

This chapter discusses some of the DAC correction methods that use exact error information acquired from measurements. The discussion concerns the correction only. The measurement and error processing are beyond the scope of this chapter. Three principle correction methods are discussed: self-calibration in section 4.2. , mapping in section 4.3. and digital pre-distortion in section 4.4. . These methods are characterized by the way they utilize DAC intrinsic and extrinsic redundancies. These characteristics are further used in chapter 7 to propose a general classification of the DAC correction methods.

4.1. INTRODUCTION

The open literature provides various DAC correction methods that can improve the DAC performance using exact information about the DAC errors. Broadly, these are methods that are using exact error information to provide exact error correction. This avoids background correction activity that increase power consumption and can potentially deteriorate performance. These methods do not rely on good understanding about the error mechanism and hence they can correct “unexpected” errors, e.g. due to migrating to other technologies or integrating an IP-block within another system.

This chapter discusses a selected set of state-of-the-art DAC correction methods that use self-measurements. That is to say these methods illustrate state-of-the-art principles for improving performance using exact error information. Some of these principles can be used together or in combination with correction methods that do not use exact error information (chapter 3). Section 4.2. discusses the self-calibration of currents method. Section 4.3. discusses the unary code mapping method. Section 4.4. discusses the digital pre-distortion method. A general discussion is provided in section 4.5. . Finally, conclusions are drawn is section 4.6. .

4.2. SELF-CALIBRATION OF DAC CURRENT CELLS

Self-calibration of the switch-current (SI) cell is a direct and powerful method to correct the DAC errors. This method directly targets the errors. It firstly measures the errors, then processes the measurement information, and finally applies correction. In this thesis, self-calibration is defined as follows. “*Self*” means that the correction method is fully autonomous: the error measurement, the algorithm, and the error correction are integrated on-chip and no special activity from the customer is required. Thus, considering the DAC as a black box, there are no functional differences between a self-calibrated DAC and an intrinsic DAC. “*Calibration of the SI cell*” means a correction method that measures and corrects the DAC errors in the same domain (e.g. analog domain for amplitudes and timing errors). This definition is meant to distinguish the self-calibration correction method from other DAC correction methods for the sake of classifying their properties. Figure 4.2-1 shows a generalized block diagram of the self-calibration methods for current-steering DACs. The self-calibration methods always add extrinsic redundancy to the DAC design, i.e. circuits not directly used for the purpose of D/A conversion. The measurement is implemented in the analog domain (including in some cases the time domain). The error processing can be either in the analog or in the digital domain. The correction is always in the analog domain. The self-calibration can correct both amplitude and timing errors.

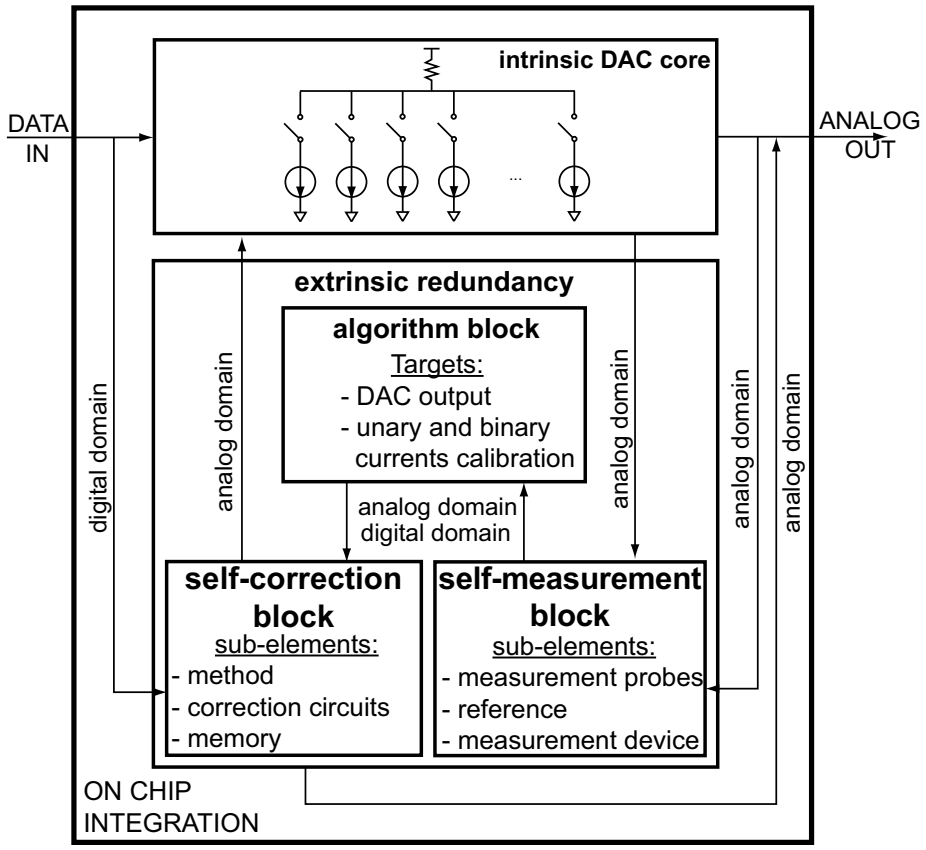


Figure 4.2-1. A generalized block diagram of self-calibrated CS DACs.

4.2.1. Amplitude errors self-calibration

The transistor mismatch errors cause inaccurate current generation in the DAC current cell, which deteriorates both the static and the dynamic DAC performance. Particularly in combination with other error mechanisms, the DAC performance degradation can be significant. Therefore, correcting the amplitude current cell errors not only improves the accuracy of the DAC analog output generation but also reduces errors due to mechanisms that correlate with the current amplitudes, as discussed in chapter 5.

The methods for amplitude error calibration are diverse. Some are completely analog, e.g. [34], [27]. Others use digital error processing [35], [36]. There are background and foreground calibration methods, etc. However, the common properties are that the errors are identified and respective corrections are applied. It is also common to observe improvement in the whole DAC bandwidth range, i.e. for both low and high input signal frequencies.

The common conceptual idea of the self-calibration methods is to acquire accurate enough a-posteriori knowledge about the amplitude errors and to apply the appropriate correction. That is how the dependence on the actual IC fabrication process is minimized and the chip yield is improved. Equation 4.2-1 shows the generation of current I_k that can be seen as either a DAC output current or an individual current source. In either case,

there is a time-invariant error I_{e_k} that accounts for the difference from the nominal design value $\overline{I}$ (time-variant errors are not considered here).

Equation 4.2-1

$$I_k = \overline{I} + I_{e_k}$$

The self-calibration methods apply the correction current I_{cor} that is chosen to be equal to the measured error I_{e_k}, as shown in Equation 4.2-2. That is how the error I_{e_k} is compensated and its effect is practically removed.

Equation 4.2-2

$$I_k = \overline{I} + I_{e_k} - I_{cor} \xrightarrow{I_{e_k} = I_{cor}} \overline{I}$$

Adding extrinsic redundancy to the DAC core creates unavoidable interaction with the intrinsic DAC operation. This is a risk of deterioration of intrinsic performance. This risk is more considerable at high speeds, i.e. at high input signal frequencies and sampling rates. The main aspects of this risk include:

1. Performance degradation due to the self-measurement and self-correction circuits;
2. Reduced correction effectiveness at high speeds;
3. Performance degradation due to interference from calibration activity.

In order to add the self-measurement and self-correction circuits, the optimal intrinsic DAC architecture should be adapted, since both the error measurement and correction are performed in the analog domain. Therefore, the DAC intrinsic behavior unavoidably deviates from its optimal performance. Balancing between the performance improvement and deterioration is an important analysis step in the design of a DAC self-calibration method. The correction is static, in the sense that the targeted errors are static. Furthermore, it is always passive for the foreground self-calibration methods, e.g. [35], and it is always active for the background self-calibration methods, e.g. [27]. At high speeds, other error mechanisms become dominant, e.g. timing errors due to mismatched sampling moment. That is to say the self-calibration methods for amplitude errors are most effective at low speeds. Therefore, it is important to analyze up to what speeds the calibration advantages overcome their disadvantages. Furthermore, for the foreground self-calibration methods the correction is time-invariant. However, there are often time-variant errors, e.g. errors due to temperature changes. A quick solution may be the background calibration methods, in which the errors are continually being measured and corrected. However, these methods may additionally deteriorate the DAC performance due to the actual activity of the calibration method, while the DAC performs D/A conversion.

4.2.2. Timing errors self-calibration

The mismatch errors between the switch-transistor threshold voltages V_{th} add timing errors in the data sampling moments $\overline{t_{on}}$. Since the switching transistors (e.g. of the synchronization latches, clock and data buffers, DAC current cells) need to be small to achieve high speeds, their mismatch errors cannot be reduced by design (big layout units placed close to each other). Therefore, the timing errors can substantially limit the DAC performance at high speeds.

Self-calibration of timing errors is similar to the amplitude errors self-correction. Instead of correcting the current amplitudes, the self-calibration corrects the sampling moments t_{on}. Due to various transistor mismatches in the data chain: "*latches-data drivers-current switches*" and due to various spatial delays and parasitic capacitances, the current cell sampling moments t_{on} are not identical. They deviate from the nominal sampling moment $\overline{t_{on}}$ by an error t_e, see Equation 4.2-3 and Figure 5.3-2.

Equation 4.2-3

$$t_{on} = \overline{t_{on}} + t_e$$

The timing error t_e is a time-invariant error, because it is due to time-invariant error sources and mechanisms. Note that a distinction should be made between the time-invariant error t_e and clock jitter, which is time-variant (it is a similar distinction as between amplitude mismatches and noise). The timing self-calibration method measures t_e and applies an appropriate correction t_{cor}.

Equation 4.2-4

$$t_{on} = \overline{t_{on}} + t_e - t_{cor} \xrightarrow{t_e = t_{cor}} \overline{t_{on}}$$

The work of [37] presents an approach that can measure the timing errors and correct them via DDL-based delays in the data chain.

The major limitation for the timing error self-calibration methods is the measurement of the timing errors t_e. There is no literature source yet that reports timing errors measurement in practice. Note that this correction method is most effective for high speed DAC input signals, when a lot of switching activity is present. Thus, the resolution step of the timing error measurement needs to be very small.

4.2.3. Discussion

The DAC self-calibration methods use extrinsic DAC redundancy. That is why the risks of intrinsic DAC performance deterioration should always be carefully considered. The correction is always in the analog domain and interacts with the intrinsic DAC current generation. However, the self-calibration corrects the errors based on a-posteriori error knowledge and hence reduces the DAC accuracy dependence on the IC fabrication process. As a second-order consideration, the self-calibration methods also rely on a-priori error knowledge in order to be able to appropriately design the operational ranges of their correction circuits. In other words, the correction should be able to cover the errors within the predicted range. Since the mismatch errors are corrected, the intrinsic DAC transistors have relaxed accuracy requirements and hence significant silicon area can be saved.

4.3. MAPPING

The mapping methods use a-posteriori knowledge about the particular mismatch errors between the current cells and apply correction via combining the current cells in such a way that their errors mutually compensate each other. Thus, the mapping methods measure the errors in the analog domain, process the errors in the digital domain and compensate them in the digital domain (applying a map) with an analog effect (using the mutual error compensation). Figure 4.3-1 shows a conceptual diagram of a DAC using a mapping correction method. Note that the intrinsic DAC core is indicated as a DAC system, which may be a single unary or segmented DAC, a sub-binary radix DAC, or multiple parallel sub-DACs.

A map can be defined as a P bit particular representation of the N bit input digital word, with P>N always. From this definition follows a property of these methods: *to apply a mapping technique, the presence of DAC current cells redundancy is necessary*, since P>N is always required. Note that N bit input word is binary and hence features no code redundancy, while P<N loses input information.

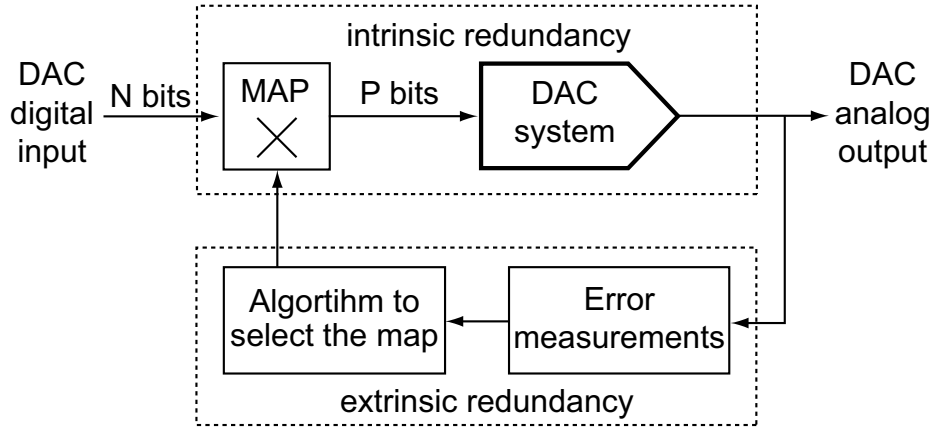


Figure 4.3-1. A conceptual diagram of a DAC using a mapping correction method.

The requirement for the DAC intrinsic redundancy (P bits represent N bit data with P>N) implies in the general case that there are multiple combinations of the DAC resources, i.e. current cells, to represent an input digital word W. Each of these combinations is unique in the analog domain, because of the current source mismatch errors. The coding efficiency is exchanged for redundancy, which is the source of the DAC performance improvement. The mapping algorithms select the optimal combinations.

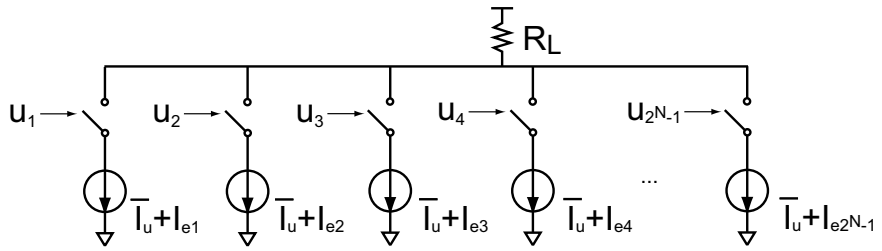
Generally, the MAP block can be defined as a $\left(2^N-1\right)\times P$ memory block. However, certain more efficient solutions may exist for certain applications of the mapping correction method. The contents of the memory block is the particular mapping function, representing the chosen map M_k as a time invariant correspondence between the input N binary bits and the output P bits. The map M_k can be applied to correct either DAC static or dynamic errors but not both since the amplitude and timing errors are different. A compromise between both may also be the target of the algorithm to select the map.

To reduce the size of the MAP block, digital logic instead of memory can be used. Usually, these logic circuits offer only a limited number of available maps. When the correlation between these available maps is low, a much more efficient improvement is achieved using logic circuits than using memory blocks. Some examples of such digital logic are programmable decoders (e.g. proposal in section 11) and matrices based on switches, such as the butterfly switch matrix [38].

4.3.1. Low level maps for DAC unary current cells

The following text briefly presents the mapping method for the unary DACs. It is the simplest form of mapping methods, if compared to the mapping for sub-binary radix DACs and the high-level mapping method, proposed in chapter 12. The text considers the unary DAC intrinsic core, but all examples are also applicable for the unary MSB part of the segmented DAC architectures. Basic definitions of the mapping function are made. The limitations of the method are considered. Finally, discussion is provided.

To illustrate the concept of coding redundancy in DACs, consider the thermometer (unary) digital coding, where intrinsic redundancy is present. For the unary coded DAC, each input bit represents an LSB step. Thus, to convert N binary bits, a unary DAC uses 2^N-1 unary bits that control 2^N-1 unary current cells, as shown in Figure 4.3-2.



Note: u_k is the k-th unary digital bit

Figure 4.3-2. Unary CS DAC controlled by unary digital data.

Let the digital unary bits be $u_k, k = 1,2...2^N - 1$. To convert the input $W[nT]$, w digital unary bits are set to 1 and $2^N - 1 - w$ digital unary bits are set to 0. The thermometer unary coding is defined to be when $u_i = 1, i = 1,2...w$ and $u_j = 0,\ j = w+1,...2^N - 1$. However, the thermometer coding is only one possible combination of the unary bits that represents $W[nT]$. There are a number of other combinations, given by:

Equation 4.3-1

$$ {}_{(2^N-1)}C_w = \begin{pmatrix} (2^N-1) \\ w \end{pmatrix} = \frac{(2^N-1)!}{w!(2^N-1-w)!} $$

In the digital domain there is no difference for which ever combination is chosen for $W[nT]$, since there are no mismatch errors, whilst in the analog domain there are ${}_{(2^N-1)}C_w$ different representations of $W[nT]$, due to the presence of the mismatch errors I_{e_k}. The distribution of the DAC outputs for all ${}_{(2^N-1)}C_w$ has its expected value equal to the sum of the nominal currents, i.e. $w \cdot \overline{I_u}$ and its deviation is defined by the statistical properties of the errors I_{e_k}.

The collection of chosen combinations for all input codes defines the chosen map *M*. Within the set of all possible maps, some specific sub-sets can be pointed out.

The sub-set of thermometer maps includes these maps that feature strong correlation between the chosen combinations for all input codes, like in the thermometer coding. That is to say the unary code combinations for adjacent codes differ by only one added current cell, like in the conventional thermometer coding. Therefore, these maps share the advantages of the thermometer coding – low glitch energy and low power consumption for incremental code transitions. Note that for non-thermometer maps, the two chosen combinations for two adjacent codes may significantly differ for the selected

current sources and hence at the particular code increment event many current switching events occur, which generate glitch energy and consume power.

Various sub-sets can be defined to include such maps which can be implemented using common hardware resources. Since for the general mapping approach at high resolutions large hardware resources are needed, these hardware realizations of the mapping block allow more efficient implementation of the mapping method, e.g. programmable decoders as proposed in section 11 and the butterfly switch matrix (matrices based on switches) [38].

In practice, the choice of the combination C is realized by a 2^N-1 to 2^N-1 decoder block (the MAP) that is placed after the binary-to-thermometer decoder, as shown in Figure 4.3-3.

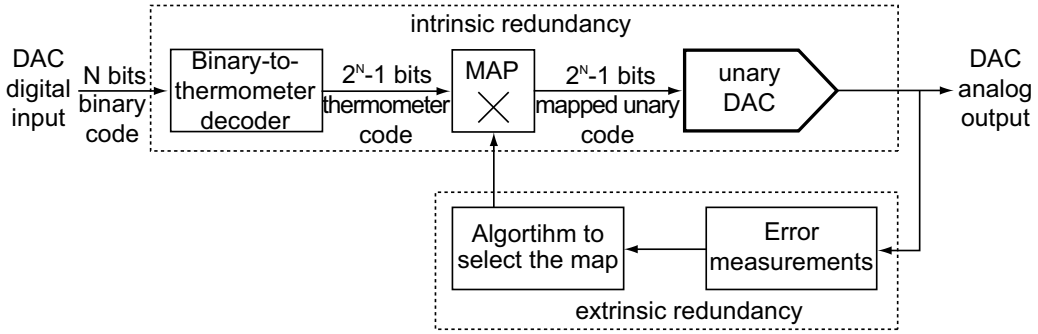


Figure 4.3-3. A mapping correction method for a unary DAC.

A full MAP decoder block can realize all possible permutation of the 2^N-1 codes, i.e. $(2^N\text{-}1)!$ different maps $M_j,\ j \in [1,2,...,(2^N-1)!]$. An illustrative example of the mapping process is shown in Figure 4.3-4. The first matrix shows the nominally identical DAC current cells. A map M_j assigns a switching sequence order to each of the current cells. In the case of thermometer coding, the switching sequence order coincides with the order of the DAC current cells. This map is considered as the first map M_1 (shown at top-right). However, many other switching sequences also exist, e.g. as shown in Figure 4.3-4 (bottom two) for the case of the thermometer maps sub-set. The mapping method knows each I_{e_k} and would choose that map which features the lowest INL_{max}.

Note that the mapping method is discussed here as using a static map, without loss of generality. However, a combination of the mapping method and the DEM method (see chapter 3.5.) can potentially produce better results than these two methods separately. The DEM method can be applied over a set of selected maps.

The main limitation of the low-level unary mapping method is in its coupling between the correction and the DAC architecture. The method can be applied only to the DAC unary current cells and hence in practice it can be applied only to the MSB DAC unary part in the DAC segmented architectures. The effectiveness of the method depends on how many DAC input bits are decoded in a unary way. However, the complexity of the method and its hardware requirements grow exponentially for each extra binary bit decoded in a unary way.

The example above considers the low level maps, when the intrinsic coding redundancy is applied of the level of the current sources, e.g. unary current sources. Note that other low-level mapping techniques also exist, e.g. redundancy in a sub-binary radix coded DACs [11]. There are also high-level mapping techniques, when the redundancy is at the DAC functional level. If there are multiple parallel sub-DACs available to convert

the digital word W, there is intrinsic redundancy in how $W[nT]$ can be split between the different sub-DACs, e.g. $W_{sub-DAC1}[nT]+W_{sub-DAC2}[nT]=W[nT]$.

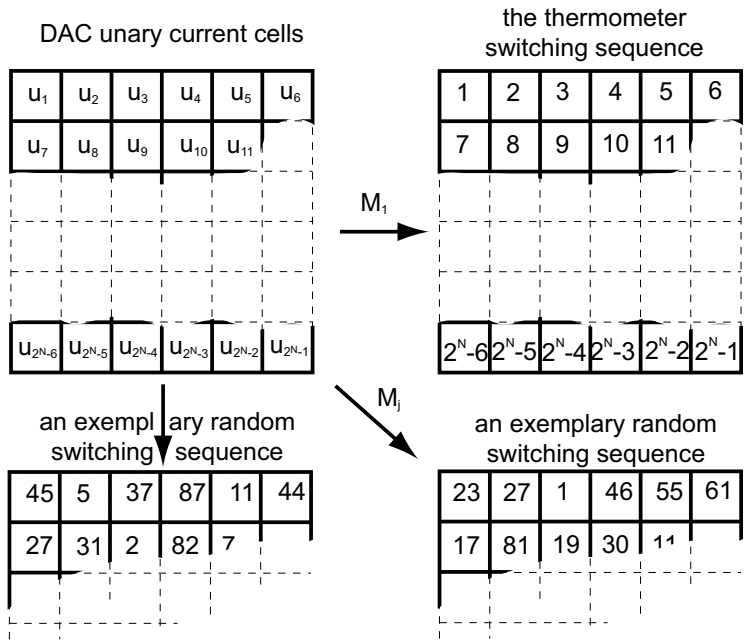


Figure 4.3-4. Mapping of the DAC unary current cells to a switching sequence.

4.3.2. Low-level maps for sub-binary radix DACs

The following text briefly presents the low-level mapping correction techniques for sub-binary radix DACs. The key factors to achieve sufficient intrinsic redundancy for the mapping technique are:

1. DAC currents are designed with scaling coefficients smaller than 2;
2. The number of DAC currents exceeds the number of DAC input bits.

The method measures the errors and selects such a map for the DAC input bits that the DAC output current is sufficiently linear. Since the method can tolerate large amounts of mismatch errors, the current source transistors can be designed small. The method requires much less additional hardware resources than equivalent methods for the unary DAC. Hence, very small DACs with high linearity can be designed.

The digital coding can be represented as a continuum of the scaling coefficients of the individual bit weights of the digital word, as shown in Figure 4.3-5. The binary coding having scaling coefficient of 2 is the most efficient coding to process bit information, since the digital bit itself has two states either 0 or 1. However, the binary coding features no redundancy and hence the low-level mapping techniques cannot be used. By contrast, the unary coding, having scaling coefficient 1, features large amounts of redundancy and hence low-level mapping techniques are possible. However, the unary coding is highly inefficient in terms of data processing and hence the mapping techniques require a lot of resources to implement. A trade-off between these two choices is found in the sub-binary

radix scaling[1]. The transfer characteristic of a sub-binary radix DAC features redundancy which can be combined with a mapping technique to achieve a targeted DAC linearity, see [11], [39], [40], [41].

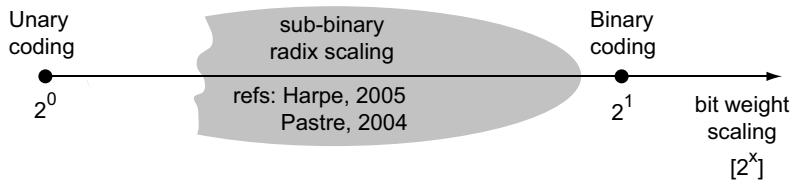


Figure 4.3-5. The digital coding continuum of the bit weight scaling.

The main conceptual idea of the sub-binary radix DACs is that the redundancy in the DAC transfer characteristic guarantees that the current mismatch errors do not cause $DNL_k > 0.5LSB$ for all codes k. All errors of $DNL_k < 0.5LSB$ (including $DNL_k < -0.5LSB$) can be corrected by the mapping technique. Figure 4.3-6 shows the basic conceptual diagram of the sub-binary radix DAC mapping technique.

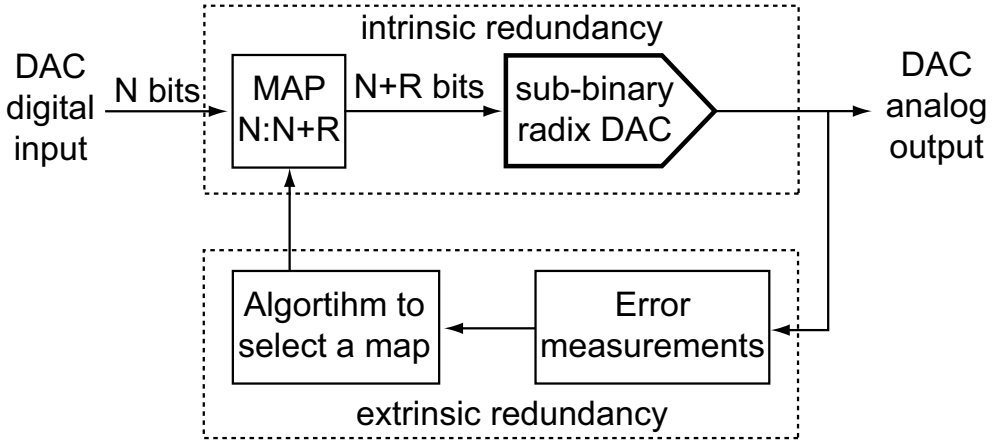


Figure 4.3-6. A conceptual diagram of a sub-binary DAC with pre-correction, e.g. [11], [39], [40].

All N+R current sources of the sub-binary radix DAC are measured. An algorithm selects the appropriate map that delivers linear N bit DAC performance. Note that the size of the map is N:(N+R). The work of [11] reports an implementation of N=12 and R=4. Thus, the size of the map is much smaller than the necessary maps of the unary currents mapping techniques, which would require map sizes of the order of $N:2^N-1$.

The common limitation of the mapping techniques is that usually a single map can be used at a time. Therefore, the map needs to be a compromise between the static, low speed and high speed DAC performance. This trade-off is particularly sharp for the sub-binary radix DAC mapping techniques. The dynamic response of the sub-binary radix DAC current cells can hardly be matched and hence significant timing errors are

[1] Strictly speaking, the unary coding is a sub-binary radix coding. For the sake of simplicity, the sub-binary radix coding is redefined here as a coding with scaling coefficients between 1 and 2.

generated. The challenge to match the current cell responses is made difficult, since no unit elements for the sub-binary radix cells can be used in the layout of the data propagation chain, i.e. latches-data drivers-current switches.

The sub-binary radix DAC mapping techniques measure the DAC errors in the analog domain. The actual correction is in the digital domain, but the correction effect is in the analog domain.

The size of the pre-processor digital circuits is highly reduced in comparison with the unary current cells mapping approaches. However, the matching of the current cell dynamic responses is compromised.

4.4. DIGITAL PRE-DISTORTION

The digital pre-correction methods measure the errors in the analog domain, process the errors in the digital domain and compensate the errors in the digital data domain with an analog effect.

The DAC pre-distortion correction methods measure the linearity of the DAC, calculate the approximate deviation from the desired linear performance and then add an inverse-linearity function to the DAC digital input. The process may be iterative until the DAC output becomes linear, e.g. [42], [43]. Figure 4.4-1 shows a block diagram of the DAC pre-distortion method.

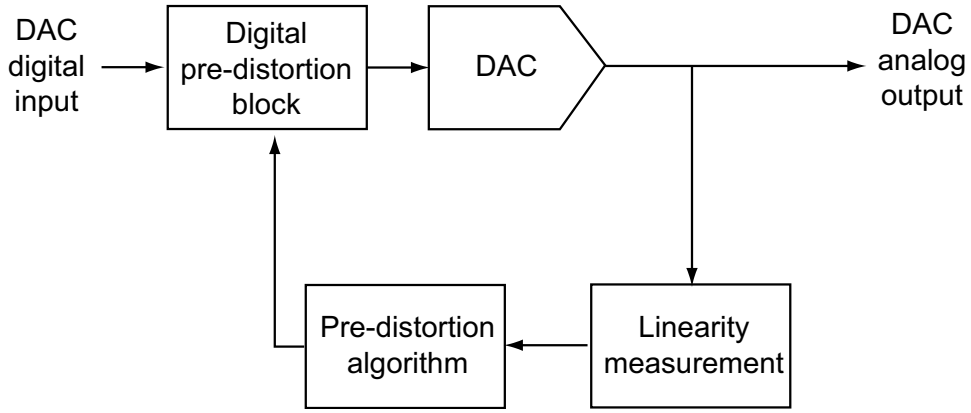


Figure 4.4-1. A block diagram of the DAC pre-distortion method.

A popular approach in the literature is to include in the pre-distortion correction the non-linearity of the whole transmitter front-end, such as shown in Figure 4.4-2, e.g. [44], [45].

The method has difficulties with the ‘gaps’ in the DAC transfer characteristic, according to [42]. These are large positive DNL errors that cause distortion for which it is difficult to find a compensating pre-distortion function. In addition, power supply noise and clock jitter problems cannot be compensated by the pre-distortion methods, according to [43].

The DAC pre-distortion method measures the DAC linearity in the analog domain and corrects for it in the digital domain. This method shares a lot of similarities with the mapping correction methods. However, it does not rely on DAC intrinsic redundancy like the DAC mapping correction methods, but rather on an extrinsic redundancy that adds counter distortion in the system aiming at the cancellation of the overall distortion.

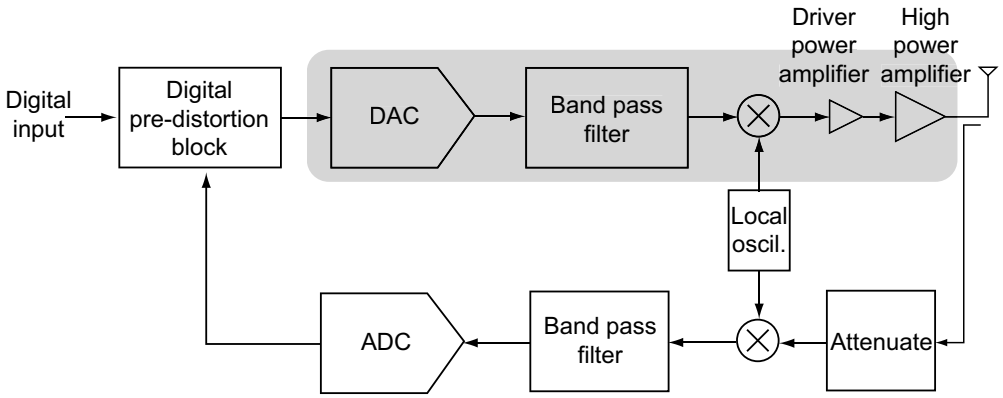


Figure 4.4-2. Pre-distortion correction for the whole transmitter front-end.

4.5. DISCUSSION

This chapter demonstrates different DAC correction methods using exact error information. Further in this thesis, chapter 7 introduces a classification of the DAC correction methods to systemize them, to identify missing methods, and to derive clues to fill up these gaps.

Table 4.5-1 gives a short summary of the discussed DAC correction methods, ordering them according to four discussed classes: using intrinsic or extrinsic redundancy and implemented on low- or high level. Three notations are used: “Yes” when the method exist within a class; “No” when the method does not exist within a class; and N/A when the method cannot exist within a class (e.g. if it is against its definition). Note that all boxes indicated with “No” may be eventually filled-in with methods that do not exist in the open literature yet.

Table 4.5-1. Summary of the discussed DAC correction methods.

Correction method	***Classes***			
	Intrinsic redundancy	***Extrinsic redundancy***	***Low-level***	***High-level***
Self-calibration for timing errors	*N/A*	***Yes***	***Yes***	*No*
Self-calibration for amplitude errors	*N/A*	***Yes***	***Yes***	***Yes***
Mapping for unary currents	***Yes***	*N/A*	***Yes***	*N/A*
Mapping for sub-binary radix DACs	***Yes***	*N/A*	***Yes***	*No*
Digital pre-distortion	*No*	***Yes***	*No*	***Yes***

4.6. CONCLUSIONS

An overview of DAC correction methods that use self-measurement is presented. The overview is based on state-of-the-art works that are recently published. The main aspects of each correction method are discussed. The targeted error mechanisms for the two methods *self-calibration of currents* and *unary code mapping* are similar. However, both methods have their own advantages and disadvantages. The self-calibration is capable of correcting both random and systematic mismatch errors but interacts with the analog DAC signal generation. The mapping method avoids interacting with the analog DAC signal generation, but is not that powerful in correcting systematic mismatch errors. Therefore, it is also argued that applying a DAC correction method is a matter of a trade-off, since each correction method brings its own particular advantages and disadvantages. Further in the thesis, chapter 7 proposes a classification of DAC correction methods.

PART III: NEW MODELING, ANALYSIS, AND CLASSIFICATION

This part includes chapters that analyze and classify DAC error and correction methods. The flow of the presentation follows a pattern of firstly describing a global view on the problems and then concentrating on a representative part of the problem area. Chapter 5 proposes a general modeling of the DAC errors. Chapter 6 concentrates on the analysis and modeling of the amplitude mismatch errors of the DAC currents. Chapter 7 introduces a global classification of the DAC correction methods. Chapter 8 concentrates on the in-depth analysis of the DAC correction method of self-calibration of current mismatch errors

.

5. Chapter

ERROR MODELING FOR DAC CORRECTION, A BROAD VIEW

This chapter reviews some of the DAC error mechanisms that can be counteracted by correction methods. The mismatch related amplitude and sampling errors of the DAC current cells are discussed. A model is proposed that explains the performance deterioration due to the amplitude and timing errors of the current cells. The timing errors are modeled as a composition of sampling and amplitude errors. This model helps explaining how the amplitude errors contribute to the timing errors and hence influence the DAC dynamic performance. Beyond this, other data-dependent error mechanisms are considered: the disturbance at the common source node of the current switches, charge feed-through, and disturbances on the power supply and bias wires. These secondary errors interact with the amplitude and sampling errors and form error mechanisms that cause DAC non-linearity. Since these errors depend on the transistor mismatch, the performance of the overall DAC design is bound to the specified spreads and tolerances of the used IC technology. The chapter defines the Error Transfer Function and its frequency dependence, which is further used for the proposed classification in chapter 7.

5.1. INTRODUCTION

The diversity of DAC errors and error mechanisms is very high. They cause the DAC performance to deteriorate well below the theoretical prediction. For example, certain errors may result in deterioration of the DAC performance in the form of higher non-linearity, gain errors, higher glitch energy, and lower speeds than what is nominally expected.

The DAC application requirements determine the performance specifications, which define the ranges of the allowable DAC errors. For some DAC applications, particular performance parameters are more important than others and hence reducing a selected set of DAC errors and error mechanisms is prioritized over others. For example, some communication applications may require small non-linear errors but tolerate a gain error. However, some control applications, e.g. reference generation, may require small gain errors but tolerate non-linear errors.

Another important consideration is the DAC design and implementation. The specifics of the particular DAC designs make them sensitive to certain errors and error mechanisms. For example, the unary DAC architectures are sensitive to various gradient errors due to the large occupied areas. The low power supply DAC implementations in modern technologies are sensitive to gain errors, because of their limited voltage head room.

A broad overview of a representative majority of the DAC errors can be found in [46], [8], [1], and [5]. These works concentrate on certain error mechanisms and derive the error requirements for the DAC to achieve a targeted performance specification. However, some performance specifications, e.g. SFDR, THD, result from many errors, error mechanisms and their mutual interaction. In the end, this chapter considers a generalized view of the error mechanisms – the Error Transfer Function (ETF), which is further used in this thesis to classify the DAC correction methods (chapter 7).

The high diversity of the DAC errors and their interacting error mechanisms require various DAC correction methods. To provide some concrete examples, this chapter considers in detail only a selected sub-set of the DAC errors: amplitude and timing mismatch-related errors and digital switching caused errors. It is shown how the amplitude errors of the analog elements cause both static and dynamic performance deterioration. Beyond the provided discussion, other data-dependent errors also cause the DAC to deviate from its nominal performance (in particular non-linearity), see [1], and [5]. These include non-linear transistor switching, and finite output impedance of the current sources. Though these errors are not discussed in this chapter, they can still fit in the proposed generalized error description.

To generalize, the D/A (DAC) function can be split into the nominal Signal Transfer Function (STF) and the unwanted Error Transfer Function (ETF):

Equation 5.1-1

$$\underbrace{f\left(Inputs, i_1, i_2, i_3, ..., e_1, e_2, e_3, ...\right)}_{\text{D/A Transfer Function}} = \underbrace{f_S\left(Inputs, i_1, i_2, i_3, ...\right)}_{\text{Nominal Signal Transfer Function (STF)}} + \underbrace{f_E\left(Inputs, e_1, e_2, e_3, ...\right)}_{\text{Error Transfer Function (ETF)}}$$

where *Inputs* represent the DAC inputs, e.g. input data, clock, control signals; i_j represents the nominal analog units, e.g. current cells and their switching; e_j represent the errors of the analog units and inter-modulating error mechanisms. The DAC correction methods aim to minimize the contribution of the ETF, so that the D/A transfer function is as close as possible to the nominal STF.

Section 5.2. proposes a model for the step response of the DAC current cell, based on [1]. Section 5.3. discusses errors that are caused by transistor mismatch: current amplitude errors and timing errors of the current switching. Section 5.4. discusses errors that are caused by digital switching: voltage disturbances at the common source node of the switch-transistors; charge feed-through; disturbances on the power supply wires. Section 5.5. provides a general discussion. Finally, conclusions are drawn in section 5.6. .

5.2. A MODEL OF THE STEP RESPONSE OF A CURRENT CELL

To demonstrate the DAC error mechanisms, a model for the step response of the DAC current cell is presented below. Figure 5.2-1 (top) shows a transistor-level simulation of the differential current step response of a DAC current cell. To model this response, three parameters are used $\overline{t_{on}}$ (turn-on time), t_{slew} (slewing time), and $\overline{I_u}$ (settled current). Below follows an example of how the three parameters can be derived.

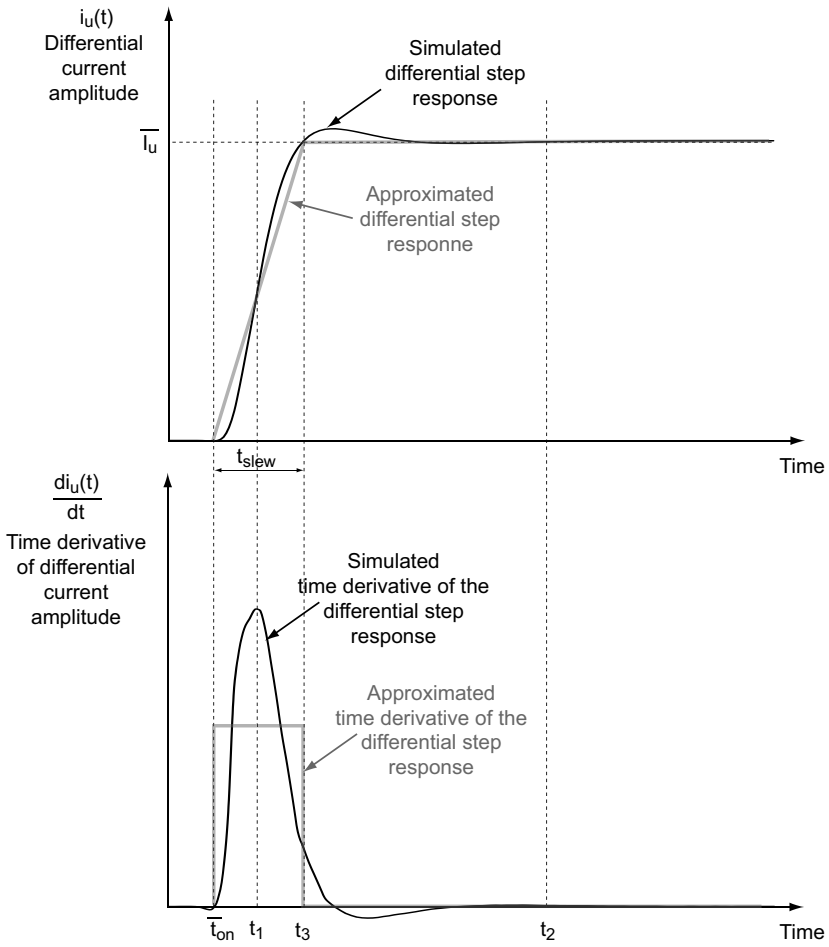


Figure 5.2-1. A simulation and the proposed model for the step response of the DAC current cell.

The current step $i_u(t)$ initiates at a sampling moment, when the digital data changes. This moment can be detected looking at the time derivative $\frac{di_u}{dt}$ of the current amplitude

value, shown in Figure 5.2-1 (bottom), $\frac{di_u}{dt}$ becomes negative and then quickly crosses zero to rise sharply (clarified at the end of section 5.4.). The moment of zero crossing is defined in the proposed model as $\overline{t_{on}}$. If in some DAC implementations, such pattern is not observed, $\overline{t_{on}}$ can be set, for example, at the moment when $\frac{di_u}{dt}$ crosses some threshold value.

The time derivative $\frac{di_u}{dt}$ reaches maximum at t_1 and then declines. Eventually, it settles at zero and the current step settles to its static value $\overline{I_u}$. This moment can be defined as t_2, when the fluctuation of $\frac{di_u}{dt}$ around zero is within certain limits of approximation. Thus, $i_u(t)$ is already settled to $\overline{I_u}$ at t_2. Looking back at the $i_u(t)$ plot, $i_u(t)$ reaches $\overline{I_u}$ for the first time at t_3. The proposed model features two parameters a rising edge interval ($t_3 - \overline{t_{on}}$) and settled value $\overline{I_u}$. The rising edge interval is when the current rises from $\overline{t_{on}}$ until t_3, shown as t_{slew}. From t_3 on, the proposed model assumes that the current has settled to $\overline{I_u}$.

This model is further used to demonstrate DAC errors and error mechanisms and to explain some simple aspects of the DAC performance. Through this model, the amplitude and timing errors of the DAC current cells are illustrated. The model also shows the DAC performance deterioration due to higher input speeds and why by improving the DAC static amplitude errors, the DAC dynamic performance improves, too.

5.3. TRANSISTOR MISMATCH CAUSED ERRORS

There are two main classes of DAC errors that are related to transistor mismatch: the amplitude errors and the timing errors. These errors are due to both transistor random mismatches and various on-chip gradients. Figure 5.3-1 (top) shows simulated step responses of DAC current cells with amplitude and sampling moment offsets, referenced to a nominal response. Figure 5.3-1 (bottom) shows the modeled step responses of DAC current cells with amplitude and sampling moment offsets, referenced to a nominal response.

The amplitude and sampling moment offsets introduce differences between the responses of the DAC current cells. These differences represent both linear and non-linear errors.

Figure 5.3-2 further elaborates the model and shows a first order approximation model of the amplitude and timing errors in the response of a unit current (I_u) switching in one clock period T_s.

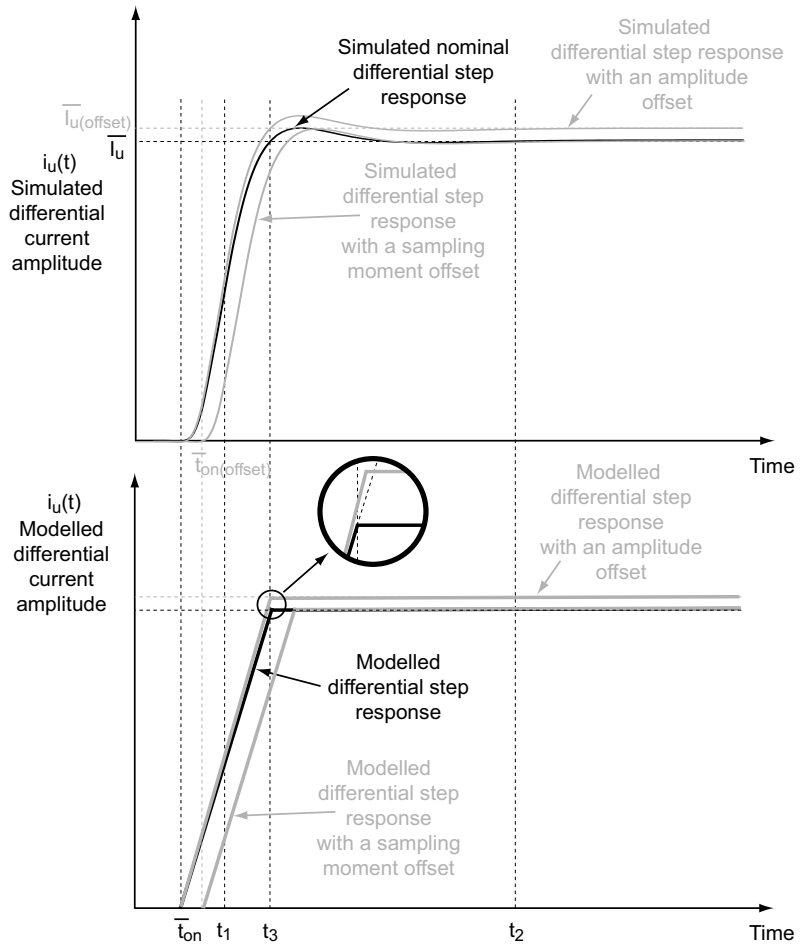
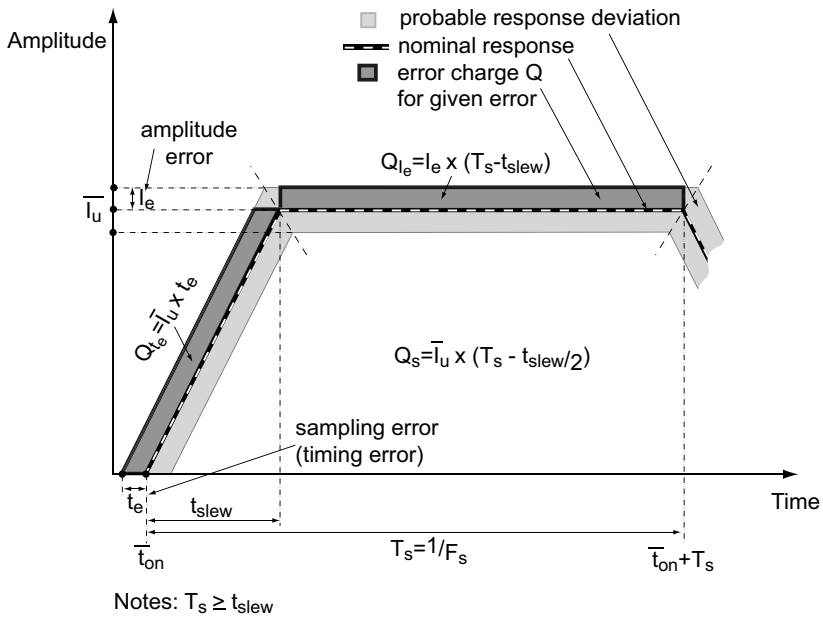


Figure 5.3-1. Simulated and modeled time-offset and amplitude-offset step responses.

After the nominal sampling moment $\overline{t_{on}}$, the static current I_u is switched to the DAC output. Looking from the output, there is an interval of time t_{slew} while the current is rising until it settles to its final static value. As long as the slewing slope is the same for all current cells, it does not generate HD errors at the DAC output. By a first-order approximation, the slewing can be modeled as a common delay of the sampling moments as analyzed in [1]. That is why Figure 5.3-2 denotes the useful signal charge Q_s that is transferred to the DAC output as the charge transferred during the slewing period plus the charge transferred during the settled period. Furthermore, a detailed analysis of the impact of the timing errors on the DAC performance can be found in [1], [5].



Notes: $T_s \geq t_{slew}$

Figure 5.3-2. First order approximation of the amplitude and timing errors of a unit current source response for one clock period. The modeled timing errors are due to errors in the sampling moments.

The actual sampling moment usually has a deviation in time, t_e, from the nominal sampling moment $\overline{t_{on}}$ due to various on-chip mismatches. That is to say t_e models the time-invariant sampling error uncertainty. Note that the sampling error t_e is time-invariant, which makes it different from the clock jitter error. Due to t_e, the current slewing (rising edge of the step) deviates from the nominal slewing and hence creates an error charge Q_{te}. Since t_e is due to some mismatch error mechanisms, it is unique for the current cell. Thus, Q_{te} is both unique and time-invariant and hence contributes a data-dependent error charge to the DAC output, causing HD components. After t_{slew}, I_u settles to its static value, which deviates from its nominal (designed) value $\overline{I_u}$ due to the amplitude errors caused by the transistor mismatch and various on-chip gradients. Figure 5.3-2 models the amplitude error with I_e. Another time-invariant error charge Q_{Ie} is created that is associated with I_u. Note that Q_{te} (timing error charge) and Q_{Ie} (amplitude error charge) have different properties. Q_{te} exist only if I_u is switched. Q_{Ie} exist only if $T_s > t_{slew}$. The error charges can be easily calculated to be:

Equation 5.3-1

$$Q_{t_e} = \overline{I_u} \times t_e$$
$$Q_{I_e} = I_e \times (T_s - t_{slew}),$$

where Q_{t_e} is the timing error charge, $\overline{I_u}$ is the nominal value of the unit current, t_e is the sampling error, Q_{I_e} is the amplitude error charge, I_e is the amplitude current error, T_s is the sampling period, t_{slew} is the rising edge time of the switching step.

The useful signal charge Q_s that is defined by the nominal response of I_u is:

Equation 5.3-2

$$Q_s = \overline{I_u} \times \left(T_s - \frac{t_{slew}}{2} \right)$$

Combining Equation 5.3-1 and Equation 5.3-2, the first order approximation of the relative error charge per T_s is:

Equation 5.3-3

$$\frac{Q_e}{Q_s} = \frac{Q_{t_e} + Q_{I_e}}{Q_s} = \underbrace{\frac{2t_e}{2T_s - t_{slew}}}_{\text{timing error contribution}} + \underbrace{2\left(1 - \frac{T_s}{2T_s - t_{slew}}\right)\frac{I_e}{\overline{I_u}}}_{\text{amplitude error contribution}}$$

where Q_e is the cumulative error charge. For no switching of I_u, $t_{slew} = 0$ and $t_e = 0$, while for very high speeds $T_s \to t_{slew}$. Equation 5.3-3 provides a simple first order approximation for the amplitude and timing errors, represented by the cumulative charge Q_e. Note that this model generally builds on the model of [1] and adds the parameter t_{slew}. To make this representation more accurate, Figure 5.3-3 shows how *the amplitude errors contribute to the timing errors*. This is a second order effect and its error charge is smaller than the error charges of the amplitude and timing errors. There are two main situations: t_{slew} is constant (left side of Figure 5.3-3) and the slew rate is constant (right side of Figure 5.3-3).

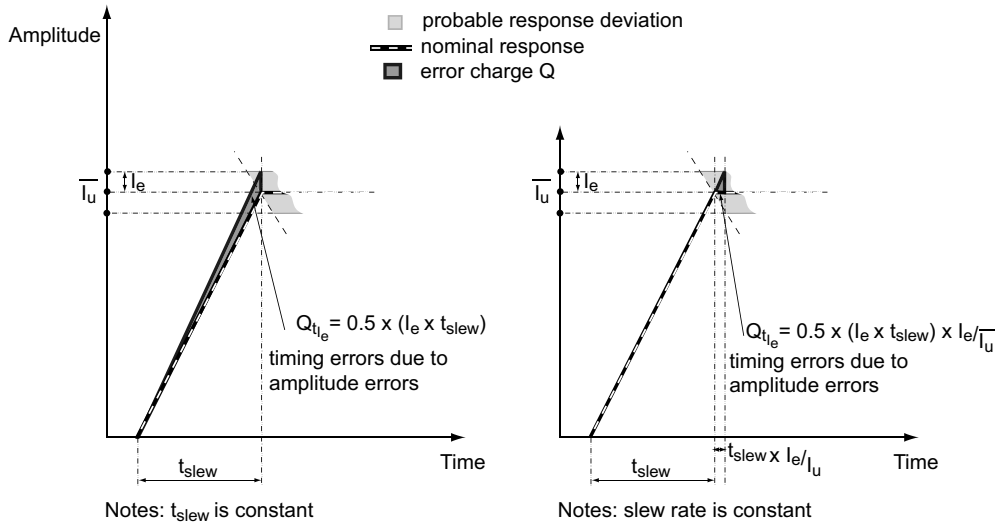


Figure 5.3-3. Modeling of the amplitude errors contribution to the timing errors.

On the one hand, when the current switching needs to charge the output capacitance to a non-nominal value, i.e. to the value $\overline{I_u} + I_e$, an error charge Q_{tle} is created:

Equation 5.3-4

$$Q_{t_{le}} = \frac{1}{2} \times I_e \times t_{slew}$$

On the other hand, due to physical limitations (e.g. the switching transients in the transistors, the connecting wire resistances), the slew rate (in the model) may remain

constant and then the nominal slewing time is changed in order to charge the output capacitance to the non-nominal value $\overline{I_u} + I_e$. The error charge is:

Equation 5.3-5

$$Q_{t_{Ie}} = \frac{1}{2} \times I_e \times t_{slew} \times \frac{I_e}{\overline{I_u}}.$$

Actually, the contribution of the amplitude current errors to timing errors is a combination of both error mechanisms, described by Equation 5.3-4 and Equation 5.3-5 that depends on the particular DAC design, e.g see Figure 5.3-1 for a SPICE (transistor level) simulated response. The important point here is that the amplitude current errors contribute to the DAC timing errors. *The correction of the amplitude errors improves both the static and the dynamic DAC performance*. Indeed, the presented measurement results before and after amplitude errors calibration in [35], [47], [27], and [36] show that improving the amplitude accuracy of the DAC unit currents improves the DAC linearity for both low and high input signal frequencies. Up till now, no explanation is given for this phenomenon in the open literature. In this thesis, this phenomenon is argued to be partially due to reducing Q_{Ie} and Q_{tIe}, but also partially due to reducing the inter-modulation of the amplitude errors with other types of errors, e.g. the digital-switching errors as discussed in the next section.

5.4. DIGITAL-SWITCHING ERRORS

Strong data-dependent errors originate from the processes of the digital data propagation and particularly of the actual D/A conversion. The work of [1] (also [5]) provides an analysis for these errors. The following text briefly considers the origins of those errors that the DAC correction methods can target.

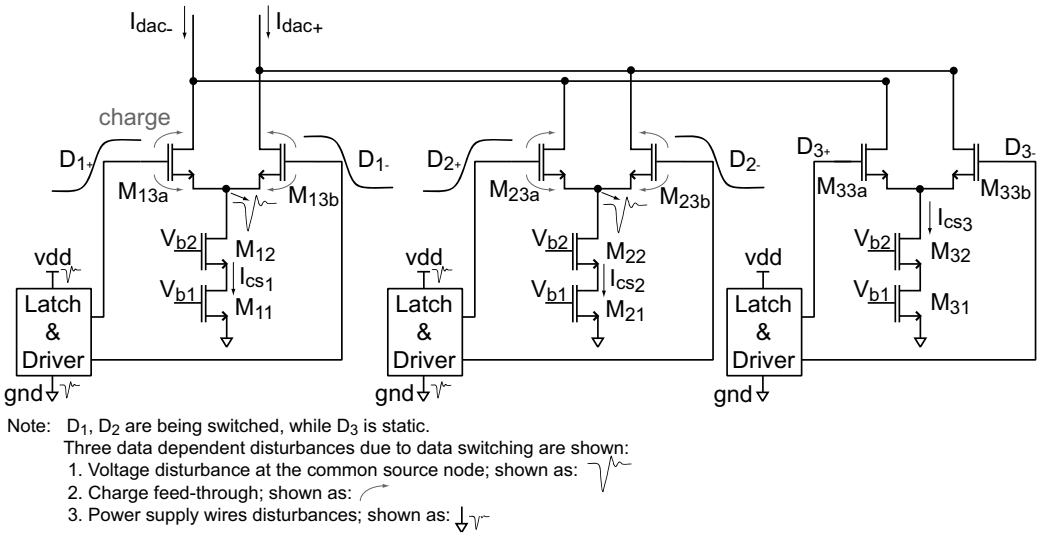
Figure 5.4-1 shows the three main types of data switching related errors:

1. voltage disturbances at the common source node of the differential current switching transistors;
2. charge feed-through via the differential current switching transistors;
3. disturbances on the power supply and bias wires.

Note that the amplitude and timing errors increase the negative effects of these dynamic errors. The amplitude and timing errors introduce mismatches in the current cells and hence their responses to the data-dependent disturbances differ. For example, the voltage disturbance at the common source node may be identical for all current sources if no mismatch errors are assumed. Then, the resulting error charge is the same for all current cells, which is a similar effect to adding the same sampling delay to all current cells. However, the presence of timing and amplitude errors makes each current cell different. Therefore, the responses of the DAC current cells to the disturbances are different. Figure 5.4-2 shows exemplary voltage disturbances at the common source node during the transients of switching. The response to a disturbance of current cell A is compared with the response to a disturbance of current cell B. The voltage disturbances are compensated by the DAC signal current, creating an error charge. The different voltage disturbances cause different data-dependent error charge. Thus, the differences among the responses of the DAC current cells cause non-linear distortion (HD, IMD, etc.).

The example shown in Figure 5.4-2 also illustrates how any data-dependent error due to digital switching combines with the unique mismatch-related amplitude and timing errors of the current cells to generate a different data-dependent error charge. Next, the chapter discusses the digital switching data-dependent errors.

Due to the non-linear nature of the current switching transistors, shown as e.g. M_{13a} and M_{13b} in Figure 5.4-1, the switching of the static current I_{cs} (from Figure 5.4-1) between the differential DAC outputs is not ideal. During the process of switching, e.g. M_{13a} is being switched on and M_{13b} is being switched off, the impedance at the common source node is changed from its static value. For example, if D1- begins decreasing and D1+ begins increasing, M_{13a} cannot immediately start conducting the current that is being reduced via M_{13b}. Thus, the voltage at the common source node is decreased. The voltage drop is compensated by the signal current I_{cs} charging the capacitance at the common source node. Effectively, the charging current represents an amount of error charge, seen from the DAC output. This mechanism also explains why for the current impulse in Figure 5.2-1 before raising the current is expected to drop (decrease) for a very short time interval.



Note: D_1, D_2 are being switched, while D_3 is static.
Three data dependent disturbances due to data switching are shown:
1. Voltage disturbance at the common source node; shown as:
2. Charge feed-through; shown as:
3. Power supply wires disturbances; shown as:

Figure 5.4-1. Data dependent disturbances due to digital data switching.

Due to the parasitic capacitances of the switch transistors, e.g. M_{13a} and M_{13b} in Figure 5.4-1, charge is injected into the current cell. There are two ways for this charge to propagate:

- through the gate-drain capacitance the charge disturbs the DAC output, directly contributing to the DAC output error charge;
- through the gate-source capacitance, the charge disturbs the common source node and hence I_{cs}.

Since the error charge is injected only at the data switching instances, its disturbance is input-data dependent. When this error charge is different for each of the DAC current cells (due to their mismatch), it causes non-linear distortion. The influence of this error mechanism increases at high speeds (both sampling rate and input signal), since the constant amount of delivered error charge per switching is related to short sampling periods.

During switching, the digital circuits, e.g. the synchronization latches and the data drivers, disturb their power supply wires. Particularly noisy are the CMOS circuits which consume instantaneous current at sampling moments. This disturbance is data-dependent and through parasitic capacitances may reach the DAC current cells.

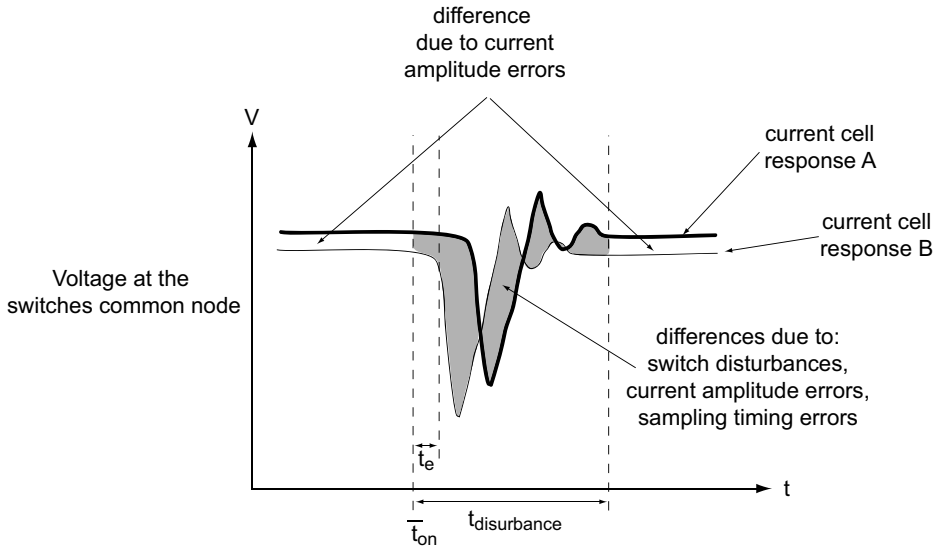


Figure 5.4-2. Voltage disturbance at the switches common node in combination with current amplitude and timing errors.

5.5. DISCUSSION

The DAC performance is limited by its errors and error mechanisms. As indicated in Equation 5.1-1, the DAC errors and error mechanisms cause an unwanted data-dependent ETF that deteriorates the DAC performance. As shown by Equation 5.3-3, the static errors dominate at low input signal frequencies - ETF is static. It depends only little on the speed of the DAC input signal. However, the dominance of the dynamic DAC errors, e.g. timing errors, digital switching related errors, increases at high DAC speeds. The higher the speed, the larger their influence on the ETF is.

Figure 5.5-1 shows exemplary measurements of a self-calibrated 12 bit 250nm DAC test chip implementation, presented further in the thesis (chapter 16.2.). This test-chip clearly demonstrates the main arguments of the presented model, since through activating the calibration engine the current amplitude errors are improved and hence isolated from the time-sampling errors. Figure 5.5-1 shows the DAC SFDR against the frequency of an input sinewave signal f_{in} before calibration (both amplitude and time-sampling errors are present) and after calibration (predominantly time-sampling errors are present). The measurement results are for sampling clock rate of F_s=50MS/s. These measurements show that beyond around 7MHz, for this particular example, the ETF starts increasing and hence the DAC linearity deteriorates. Thus, ETF features a particular frequency response that determines the frequency response of the SFDR for the whole DAC. At high input frequencies, the influence of the timing errors increases, as suggested by Equation 5.3-3. It is also shown that reducing the current amplitude errors improve DAC dynamic performance, because the timing errors are reduced too, as suggested by Figure 5.3-2. This mechanism can only be explained if t_{slew} is taken into account in the model.

To improve the DAC performance, the influence of ETF should be minimized. There are three ways to do this, as discussed further in the thesis in chapter 7: to prevent the errors, to correct the ETF, and to compensate the ETF by modifying the STF.

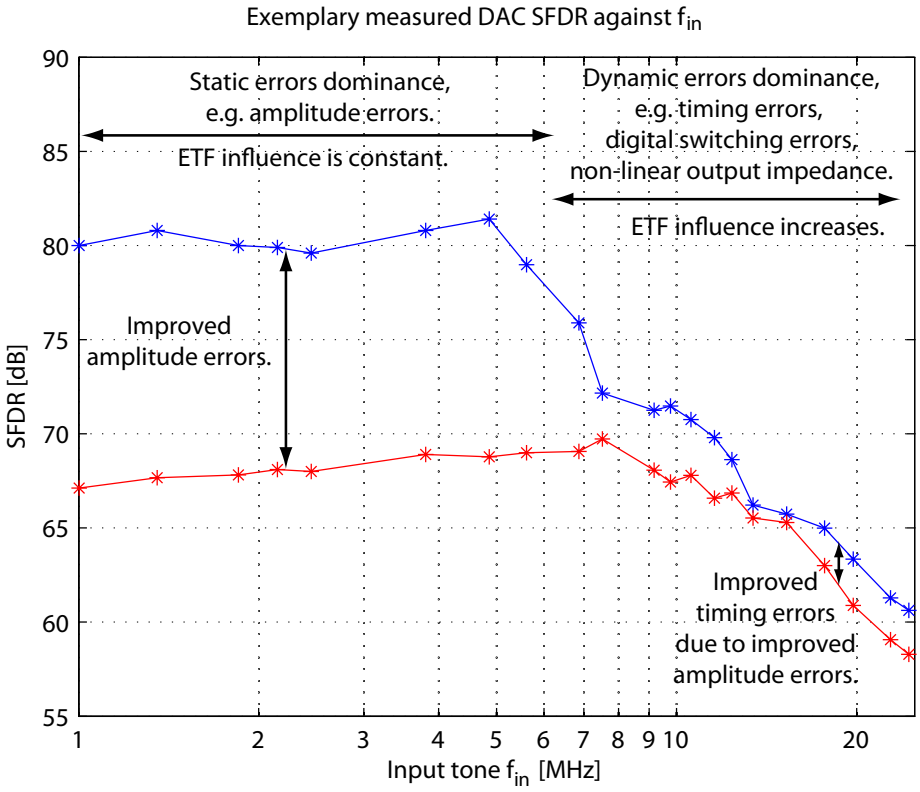


Figure 5.5-1. An exemplary measurement of DAC SFDR against f_{in}, explained by the proposed model.

5.6. CONCLUSIONS

This chapter defines the effect of DAC errors and error mechanisms as an Error Transfer Function (ETF) that is always present next to the nominal, theoretical Signal Transfer Function (STF). Both functions together constitute the D/A function. ETF increases for high DAC speeds and hence the DAC performance deteriorates.

A model for DAC errors is proposed. The model considers the rising edge of the step response of the DAC current cell as an important parameter that can explain the transformation of amplitude errors into timing errors.

To explain why correcting the static DAC current amplitude errors improves also the DAC dynamic performance, this chapter explains that the static amplitude current mismatch errors cause an increase in the ETF at high speeds via at least two error mechanisms. The first mechanism is that during switching, the DAC current units need to settle to different static values and hence their settling transients are different, resulting in input-signal-frequency dependent error charges that particularly increase ETF at high speeds. The second mechanism is that the current switching related errors are modulated by the different static errors in the DAC current cells, resulting again in error charges that increase ETF at high speeds.

To reduce the influence of the ETF, three possible methods are mentioned: to prevent errors, to correct the ETF, and to compensate the ETF through the STF. The proposed classification in chapter 7 is based on this.

6 . C h a p t e r

BROWNIAN BRIDGE BASED ANALYSIS AND MODELING OF DAC LINEARITY, AN IN-DEPTH VIEW

This chapter analytically investigates the DAC static linearity with respect to the accuracy of the DAC unit elements. It continues the discussion of the broad error modeling framework of chapter 5 by concentrating on the DAC unit current amplitude errors. The main novelty of the presented approach is in the application of the Brownian Bridge (BB) process to precisely describe the INL. This method analyzes the thermometer (unary) and binary DAC architectures and is the first to prove that their statistical INL properties are different. The INL of the thermometer DAC is represented as a one-dimensional BB process. For the binary case, the INL is represented as combinations of random variables, the increments of which coincide with a BB process. For both architectures, this chapter derives formulas for the INL main statistical properties, e.g. PDF, mean, deviation, and chip yield. These properties are compared with previous analytical attempts and conclusions are drawn. The results of this chapter fill a gap in the general understanding of the most quoted DAC specification - the INL. In particular, for a high volume chip production, the derived formulas will help engineers to choose the DAC architecture and the allowed mismatch of the DAC unit elements. This chapter is based on the original publication [23]. The detailed mathematical derivations of the presented results are published in [48].

6.1. INTRODUCTION

Linearity of the static transfer characteristic is a simple but important property of Digital-to-Analog Converters (DACs). The DAC Integrated-Non-Linearity (INL) specifies the accuracy of the code conversion and limits the achievable dynamic performance. $INL_{\max}$ gives the largest non-linear error and it is one of the most important DAC specifications for applications like video and telecommunications.

The error sources causing DAC static non-linearity can be systematic and random. Various design techniques exist to minimize the impact of the systematic errors, e.g. [49]. On the other hand, the knowledge on the impact of random errors is currently insufficient. In practice, many designers use Monte Carlo system level simulations, e.g. [12], [50]. Alternatively, inaccurate assumptions are also being applied. However, either approach will not lead to a true estimation of the effects of the mismatch errors. Often the result is over-designing the DAC analog elements with reduced efficiency of the used silicon area, increased systematic errors and hence deteriorated dynamic performance.

In current-steering DACs, the random errors of the current sources, are modeled as independent normally distributed random variables with mean $\bar{I}$ and variance σ^2. The ratio between the current deviation and current mean value $\sigma/\bar{I}$ is referred to as the relative matching and it directly determines the needed silicon area $W \times L \sim \left(\sigma/\bar{I}\right)^{-2}$ of the current source transistors [6], also shown in Equation 2.3-2. However, the relationship between the DAC static linearity expressed by the INL error and the relative matching of its current sources is not precisely understood yet, despite a number of approximation efforts in the literature [51], [52], [46], [10], [53], [54].

The first DAC analytical INL description appears in 1986 [51], [52]. $Yield_{INL}$, defined as the percentage of DAC samples that meet given $INL_{\max}$ specifications, is approximated as the product of the probabilities that the DAC output has smaller than the targeted INL error for every input code. However, due to the disregard of the correlation between the DAC outputs, the suggested specifications lead to a transistor over-design. A simpler INL formula was proposed in [46]. Instead of INL, the absolute DAC output deviation at mid-scale was approximated as $INL_{\max}$. Due to this approximation, i.e. inclusion of gain errors, and due to the disregard that the probability of the largest non-linear error does not necessarily occur at mid-scale, also this formula leads to an over-estimation. Another approximation of $Yield_{INL}$ is proposed in [6], that includes the gain errors, too. However, [10] is one of the first to suggest overcoming the mathematical complexity of the problem through tabulated functions. Later, [53], [54] continued the approach of the tabulated functions and discovered through simulations that the DAC statistical properties depend on the DAC architecture.

Due to the lack of an exact analytical formulation of INL, some important figures for high volume chip production cannot be analytically predicted, e.g. $Yield_{INL}$ and $INL_{\max}$ distribution, deviation and mean. Furthermore, the advantages of many redundancy-based approaches relying on the DAC statistical INL properties [55], [56] cannot be theoretically estimated. Finally, the main DAC architectures, i.e. binary, thermometer, and segmented, cannot be distinguished with respect to their INL properties [46], [10].

To address the need of an accurate INL analytical formulation, this chapter uses the Brownian Bridge process to derive the main statistics of $INL_{\max}$. Section 6.2. gives some basic definitions and continues into two sub-sections. *Sub-section 6.2.1.* analyses the

thermometer (unary) DAC architecture, while *sub-section 6.2.2.* analyses the binary DAC architecture. Further, section 6.3. provides a general discussion on the physical sense of the derived mathematical results. Finally, conclusions are drawn in section 6.4.

6.2. NEW STATISTICAL ANALYSIS OF THE DAC STATIC NON-LINEARITY BASED ON BROWNIAN BRIDGE

The output I_{out_k} of the current steering DAC for the three most popular architectures, the binary, thermometer (unary), and segmented architectures, can be generally expressed as group sums of the DAC unit current sources I_{u_i}. Without loss of generality, the DAC architecture is considered with an S-level algorithmic segmentation, with the $(N-S)$ LSBs being implemented in a binary way and the S MSBs being implemented in a thermometer way:

Equation 6.2-1

$$I_{out_k}^{(Segm)} = \sum_{m=1}^{N-S} B_{k,m} \sum_{i=2^{m-1}}^{2^m-1} I_{u_i} + \sum_{m=1}^{2^S-1} T_{k,m} \sum_{i=2^{N-S}+(m-1)2^{N-S}}^{2^{N-S}+m2^{N-S}-1} I_{u_i} ,$$

where N is the DAC resolution, $i \in [1:2^N-1]$, S are the bits segmented in a thermometer code, while $B_{k,m}$ and $T_{k,m}$ are the switching matrices, respectively for the binary and the thermometer parts. Note that for $S=N$, Equation 6.2-1 is a fully thermometer DAC equation and for $S=0$, Equation 6.2-1 is a fully binary equation. Figure 6.2-1 shows the equivalent high level circuit of Equation 6.2-1. With respect to current mismatch, note that the INL is the same for the single-ended and the differential DAC cases.

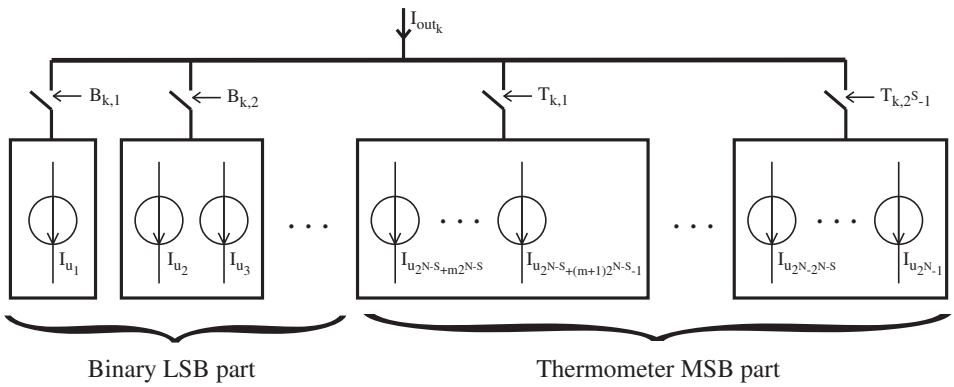


Figure 6.2-1. Segmented N bit DAC architecture based on unit elements.

According to the IEEE definition [57], INL is a measure for the deviation of the actual static transfer characteristic from the ideal one, after the linear errors have been compensated, i.e.:

Equation 6.2-2

$$INL_k = \frac{I_{out_k} - k \cdot I_{lsb}}{I_{lsb}}; \qquad I_{lsb} = \frac{I_{out_{(2^N-1)}}}{\left(2^N-1\right)} .$$

I_{lsb} is the empirical LSB step. Note that for large numbers of unit elements $n = 2^N - 1$, the denominator of INL_k can be approximated with $I_{lsb} \approx \overline{I_u}$ because it only normalizes

to the DAC LSB scale, while the numerator cannot be approximated because it describes the non-linearity.

In the following analysis, this chapter explains how the mismatch of the unit current sources I_{u_i} causes INL errors at the DAC output. A main emphasis is put on $INL_{\max}$, because it determines the chip yield. Figure 6.2-2 shows examples of empirical $INL_{\max}$ distributions for thermometer and binary DACs. Both distributions are compared with the main results of this chapter: an analytical expression of the $INL_{\max}$ PDF for thermometer DAC architectures and a reformulation of the distribution of $INL_{\max}$ in terms of Brownian Bridges for binary DAC architectures.

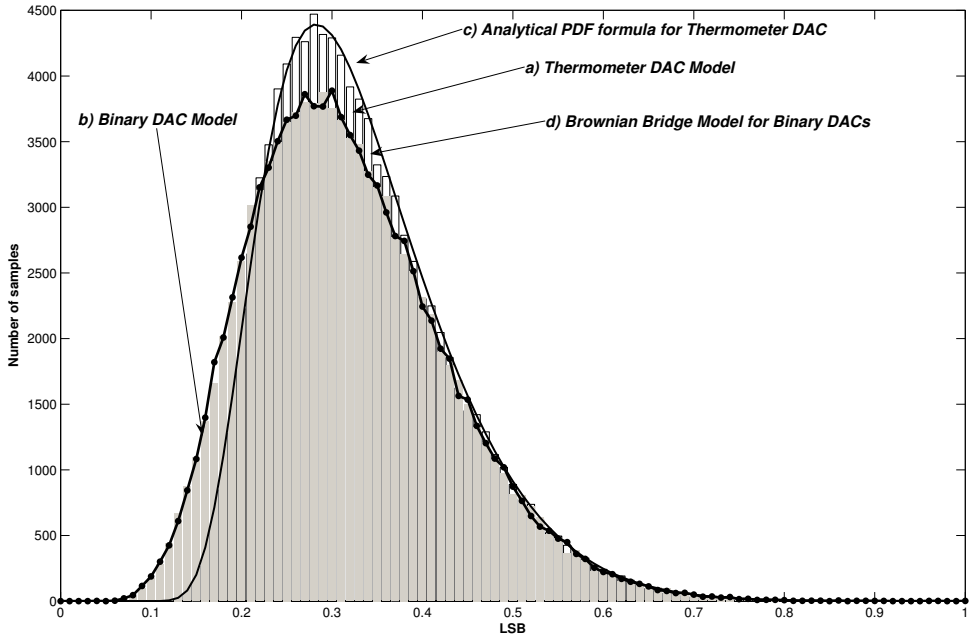


Figure 6.2-2 – Distributions of $INL_{\max}$ for 14 bit DAC models a) thermometer (white bars) and b) binary (gray bars) compared with c) PDF for thermometer DACs (straight line) and d) Brownian Bridge model for binary DACs (line with knots).

6.2.1. Unary DAC

For a unary (thermometer) DAC, i.e. $S = N$, the switching matrix $T_{k,m}$ with $k \in [0 : 2^N - 1]$ and $m \in [1 : 2^N - 1]$ is:

Equation 6.2-3

$$T_{2^N \times 2^N - 1} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix}, \text{ i.e. } T_{k,m} = \begin{cases} 0, & m \geq k \\ 1, & m < k \end{cases}.$$

Then, Equation 6.2-1 can be simplified to a cumulative sum of I_{u_i}:

Equation 6.2-4

$$I_{out_k}^{(Thermo)} = \sum_{i=1}^{k} I_{u_i} .$$

This expression shows the concept of the thermometer DAC. Every output I_{out_k} is constructed by adding an LSB unit current source I_{u_k} to the preceding output $I_{out_{k-1}}$. In Figure 6.2-3, 30 simulations of thermometer DAC samples are shown, with a single DAC sample shown in black.

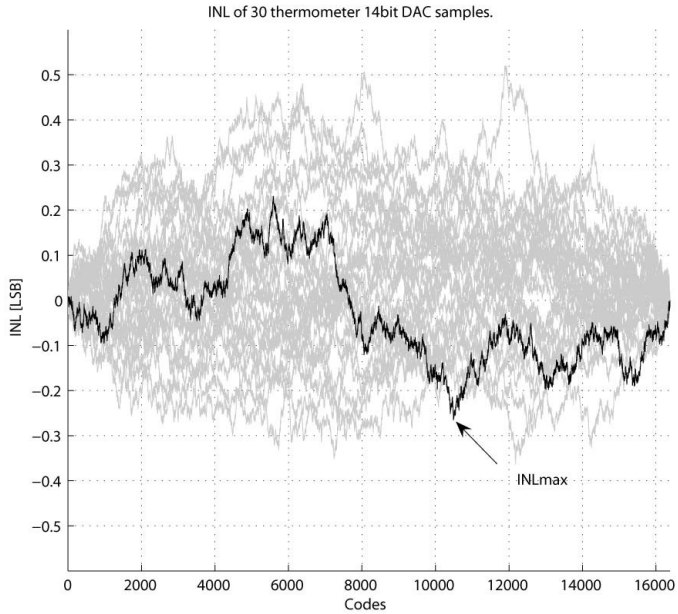


Figure 6.2-3. INL of a 14bit thermometer DAC for 30 Monte Carlo runs.

The cumulative mismatch error, i.e. the deviation from the nominal output for code k, is equal to:

Equation 6.2-5

$$I_{out_k} - k \cdot \overline{I_u} = \sum_{i=1}^{k} \left(I_{u_i} - \overline{I_u} \right),$$

where $\overline{I_u}$ is the nominal or designed unit element current, i.e. the expected value of I_{u_j}. This process is similar to a Wiener process (see e.g. [10]). So, Equation 6.2-5 can be further developed:

Equation 6.2-6

$$I_{out_k} - k \cdot \overline{I_u} = \sum_{i=1}^{k} \left(I_{u_i} - \overline{I_u} \right) = W_k\left(\sigma_u^2\right) = \sigma_u \cdot W_k(1),$$

where W_k is a Wiener process, which is also known as one-dimensional Brownian motion. Note that the Wiener process has a diffusive scaling property [58], i.e.:

Equation 6.2-7

$$W_k\left(\sigma_u^2\right) = \sigma_u \cdot W_k(1) = \sqrt{n} \cdot \sigma_u \cdot W_{k/n}(1),$$

with $n = 2^N - 1$ being the number of unit elements. Therefore, the cumulative mismatch error, Equation 6.2-5, is also diffusive, i.e. *the statistical behavior of a thermometer DAC linearly scales for different resolutions and unit elements mismatch*. Figure 6.2-4 shows the cumulative mismatch error of a 14 bit thermometer DAC model for 100 statistical simulations. The triangular shape demonstrates the diffusive scaling property, predicted by Equation 6.2-7. A single DAC sample simulation is shown with black.

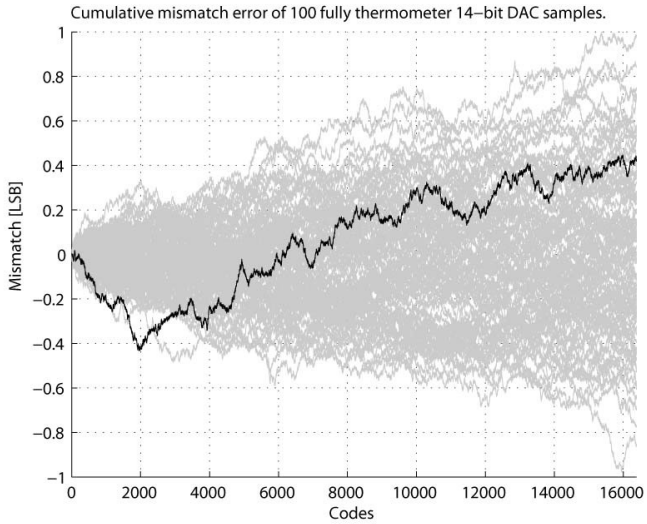


Figure 6.2-4. Cumulative mismatch simulation of 14 bit thermometer DAC (absolute accuracy).

Note that part of the cumulative mismatch can be transferred to a gain error, i.e. a linear error. For the INL, the linear portion of the mismatch error should be removed through recalculating the LSB step from the nominal $\overline{I_u}$ to the empirical I_{lsb} LSB step. The difference between these two is:

Equation 6.2-8

$$I_{lsb} - \overline{I_u} = \frac{1}{n}\sum_{i=1}^{n}\left(I_{u_i} - \overline{I_u}\right) = \frac{\sigma_u}{\sqrt{n}} W_1(1),$$

combining Equation 6.2-5 and Equation 6.2-8, the numerator of INL_k is:

Equation 6.2-9

$$I_{out_k} - k \cdot I_{lsb} = \sigma_u \cdot \sqrt{n} \cdot W_{k/n}(1) - \sigma_u \cdot \frac{1}{\sqrt{n}} \cdot k \cdot W_1(1).$$

From an engineering point of view, Equation 6.2-9 expresses the thermometer DAC non-linearity after the gain is compensated. Interestingly, this expression is also known in statistics as a Brownian Bridge (BB) $\{\mathcal{B}_t\}_{t\in[0;1]}$ [58]. Then, the expression for $INL_{\max}$ is:

Equation 6.2-10

$$\max\left|I_{out_k} - k \cdot I_{lsb}\right| = \sigma_u \cdot \sqrt{n} \cdot \max_{t_k = k/n,\ \ k=1,\ldots n}\left\{\mathcal{B}_{t_k}\right\}.$$

Note that the Wiener process continuously develops in time, while the INL DAC characteristic discretely develops with the codes. For a high number of elements

$n = 2^N - 1$, the discrete codes can be approximated as continuous, so the already developed BB knowledge can be applied for the INL:

Equation 6.2-11

$$\sigma_u \cdot \sqrt{n} \cdot \max_{t_k = k/n,\ k=1,\ldots n} \left\{ \mathcal{B}_{t_k} \right\} \approx \sigma_u \cdot \sqrt{n} \cdot \max_{t \in [0,1]} \left\{ \mathcal{B}_t \right\}.$$

Equation 6.2-9, Equation 6.2-10, Equation 6.2-11 show that the linearity of the thermometer DAC can be scaled to a standard BB function $\mathcal{B}_t$ with a scaling factor $\sigma_u \sqrt{n}$, as also mentioned above for the cumulative mismatch error of Equation 6.2-5 and also discussed in [53]. In the literature [58], the distribution of $\max_{t \in [0,1]} \{\mathcal{B}_t\}$ is known as:

Equation 6.2-12

$$P\left(\max_{0<t<1} \left| \mathcal{B}_t \right| > b \right) = 2 \cdot \sum_{k=1}^{\infty} (-1)^{k-1} e^{-2 \cdot k^2 \cdot b^2}.$$

Then, the probability density function (PDF) of $INL_{\max}$ is:

Equation 6.2-13

$$PDF_{INL_{\max}}^{(Thermo)}(x) = \frac{x}{\left(\frac{\sigma_u}{I_u} \right)^2 \cdot \left(2^N - 1 \right)} \cdot 8 \sum_{k=1}^{\infty} (-1)^{k-1} \cdot k^2 \cdot e^{-2k^2 \frac{x^2}{\left(\frac{\sigma_u}{I_u} \right)^2 \cdot \left(2^N - 1 \right)}}.$$

The $INL_{\max}$ PDF is graphically compared with the $INL_{\max}$ distribution in Figure 6.2-2. The mean and deviation of the $INL_{\max}$ PDF respectively are:

Equation 6.2-14 (a and b)

$$\mathbb{E}\left(INL_{\max}^{(Thermo)} \right) = \left(\frac{\sigma_u \sqrt{n}}{I_u} \right) \cdot \frac{1}{2} \sqrt{2\pi} \ln 2 \approx 0.869 \cdot \left(\frac{\sigma_u \sqrt{n}}{I_u} \right) \quad \text{(a)}$$

$$\sigma\left(INL_{\max}^{(Thermo)} \right) = \left(\frac{\sigma_u \sqrt{n}}{I_u} \right) \cdot \sqrt{\left(\frac{7\pi^4}{720} \right) - \frac{\pi (\ln 2)^2}{2}} \approx 0.2603 \cdot \left(\frac{\sigma_u \sqrt{n}}{I_u} \right) \quad \text{(b)}$$

Furthermore, based on Equation 6.2-13, for $|INL_{\max}| < 0.5LSB$, $Yield_{INL}$ is:

Equation 6.2-15

$$Yield_{INL} = 1 - 2 \sum_{k=1}^{\infty} (-1)^{k-1} e^{-\frac{k^2}{2\left(2^N - 1\right) \cdot \left(\frac{\sigma_u}{I_u} \right)^2}}$$

For the complete derivations of the above facts, the reader is referred to [59], and [48].

6.2.2. Binary DAC

For a binary DAC, i.e. for $S=0$, Equation 6.2-1 can be simplified to the block sum:

Equation 6.2-16

$$I_{out_k}^{(Bin)} = \sum_{m=1}^{N} B_{k,m} \sum_{i=2^{m-1}}^{2^m-1} I_{u_i} \ .$$

The switching matrix $B_{k,m}$, with $k \in [0:n]$, $n = 2^N - 1$, and $m \in [1:N]$, is:

Equation 6.2-17

$$B_{2^N \times N} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix}.$$

In Figure 6.2-5, 30 simulations of binary DAC samples are shown, with a single DAC sample shown in black.

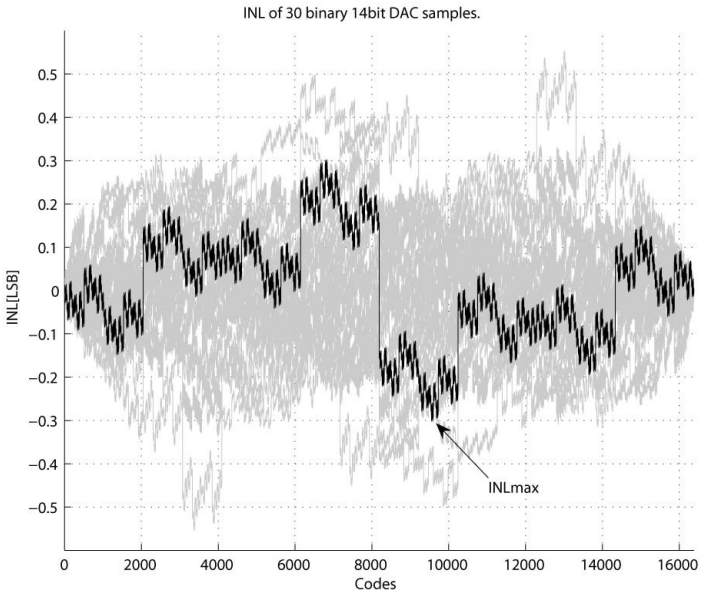


Figure 6.2-5. INL of a 14bit binary DAC for 30 Monte Carlo runs.

At codes $k = 2^{m-1}$, only one binary current source contributes to $I_{out_k}^{(Bin)}$. These points of the binary INL characteristic coincide with a BB process, similarly to Equation 6.2-10:

Equation 6.2-18

$$I_{out_{2^m}}^{(Bin)} - 2^{m-1} I_{lsb} = \sigma_u \sqrt{n} \left(\mathcal{B}_{\frac{2^m-1}{n}} - \mathcal{B}_{\frac{2^{m-1}-1}{n}} \right)$$

The increments of the BB process $\mathcal{B}$ model the mismatch errors of the binary current sources. $INL_{\max}$ is taken over every configuration of the contributing current sources, i.e.:

Equation 6.2-19 (a and b)

$$\max_{k=1,\ldots,n} \left| I_{out_k}^{(Bin)} - k \cdot I_{lsb} \right| = \sigma_u \sqrt{n} \cdot M \qquad \textbf{(a), with}$$

$$M := \max_{L \subset \{1 \ldots N\}} \left| \sum_{m \in L} \left(\mathcal{B}_{\frac{2^m-1}{n}} - \mathcal{B}_{\frac{2^{m-1}-1}{n}} \right) \right| \qquad \textbf{(b)}$$

For high n, the LSB normalization can be approximated as $I_{lsb} \approx \overline{I_u}$ and $INL_{\max}$ can be expressed with the scaling factor $\sigma_u \sqrt{n}$ and the random variable M:

Equation 6.2-20

$$INL_{\max} = \frac{\sigma_u \sqrt{n}}{\overline{I_u}} M \ .$$

To obtain the maximum appearing in M either all positive or all negative increments of $\mathcal{B}$ are combined. The increments $\mathcal{B}_{\frac{2^m-1}{n}} - \mathcal{B}_{\frac{2^{m-1}-1}{n}}$ represent the contributions of the binary elements. Note that their normal sum is 0 because of the INL symmetry, which corresponds to the definition of the BB, i.e. $\mathcal{B}_0 = \mathcal{B}_1 = 0$. Thus, M can be expressed as:

Equation 6.2-21

$$M = \frac{1}{2} \sum_{m=1}^{N} \left| \mathcal{B}_{\frac{2^m-1}{n}} - \mathcal{B}_{\frac{2^{m-1}-1}{n}} \right| .$$

At this point, the equality of $INL_{\max}$ distributions is proven for a binary DAC and Equation 6.2-20 and Equation 6.2-21. Figure 6.2-2 compares the $INL_{\max}$ distributions, generated by Monte Carlo simulations of a 14bit binary DAC and its respective BB model, as given by Equation 6.2-20 and Equation 6.2-21. Note that for the sake of simulation, the proposed model is much simpler. The needed simulation time for Equation 6.2-20 and Equation 6.2-21 is in the order of seconds, while for the high-level binary DAC model, it is in the order of hours! Furthermore, note that Figure 6.2-2 shows examples of binary and thermometer DACs that use the same I_{u_i}. However, both distributions are different as analytically and graphically shown by Equation 6.2-13, Equation 6.2-20, and Equation 6.2-21 and by Figure 6.2-2.

Furthermore, the mean and the standard deviation of $INL_{\max}^{(Bin)}$ distribution for binary DACs can be calculated with approximated M for $N \to \infty$ as:

Equation 6.2-22 (a and b)

$$\mathbb{E}\left(INL_{\max}^{(Bin)}\right)=\left(\frac{\sigma_u\sqrt{n}}{\overline{I_u}}\right)\cdot\sqrt{\frac{1}{2\pi}}\sum_{k=1}^{+\infty}\sqrt{\left(2^{-k}-2^{-2k}\right)}\approx 0.840\cdot\left(\frac{\sigma_u\sqrt{n}}{\overline{I_u}}\right), \qquad \textbf{(a)}$$

$$\sigma\left(INL_{\max}^{(Bin)}\right)\approx 0.2830\cdot\left(\frac{\sigma_u\sqrt{n}}{\overline{I_u}}\right) \qquad \textbf{(b)}$$

Note that the approximation of M for $N\to\infty$ is a similar approximation as the one used in Equation 6.2-11 for the thermometer DAC architecture, where the continuous BB process was approximated for the discrete case of the INL. It can be shown that with respect to N, the $PDF(M)$ converges to the case of $N\to\infty$, which is demonstrated in Figure 6.2-6 for N=1,2…8 (numerical simulations).

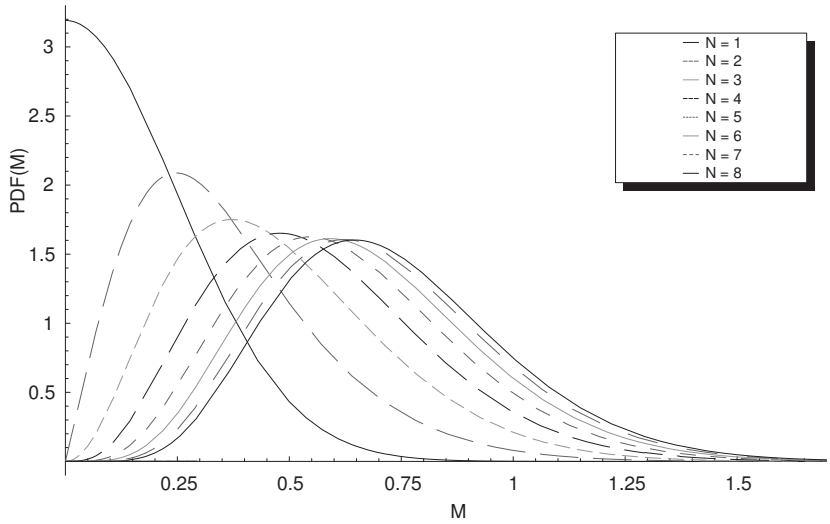


Figure 6.2-6. Convergence of the PDF of M, as N increases.

For the PDF and the derivations of the above facts, the reader is referred to [59] and [48].

6.3. DISCUSSION

As shown, the absolute mismatch of the thermometer DAC is independent for every code, while for the binary DAC, it is independent only when a single current source contributes to the output. Nevertheless, in both cases, to analyze the INL, a BB process can be used, since in both cases the INL characteristic starts and ends at 0. For both cases, the shape of the $INL_{\max}$ distribution depends on the resolution N. For the thermometer case, this dependence is neglected through the approximation of the continuous BB process into the discrete case of INL, i.e. $N\to\infty$. On the other hand, the shape of the $INL_{\max}^{(Bin)}$ distribution exponentially converges and for higher resolutions the dependence on the resolution N can also be neglected. Note also that an increment of the binary resolution by 1 bit is adding one LSB, exponentially less influencing the DAC output.

As shown in Figure 6.2-2 and by Equation 6.2-14a and Equation 6.2-22a, the mean of the binary $INL_{\max}^{(Bin)}$ distribution is smaller than the mean of the thermometer $INL_{\max}^{(Thermo)}$. *Therefore on average, binary DACs are expected to have smaller INL errors than thermometer DACs, for the same resolutions and for the same accuracy of their unit elements*. However, the spread in the binary $INL_{\max}^{(Bin)}$ distribution is larger and therefore its advantages diminish when higher yield levels are considered, e.g. $Yield_{INL_{\max}} > 99\%$. Thus, the reported $Yield_{INL}^{(Thermo)}$ formula (Equation 6.2-15) may also be used for binary and segmented DACs for high $Yield_{INL}$ values.

Figure 6.3-1 summarizes some important 14bit DAC $Yield_{INL}$ analytical results from the literature [51], [52], [10] and compares them to Equation 6.2-15 and to Monte Carlo simulations for thermometer and binary DACs.

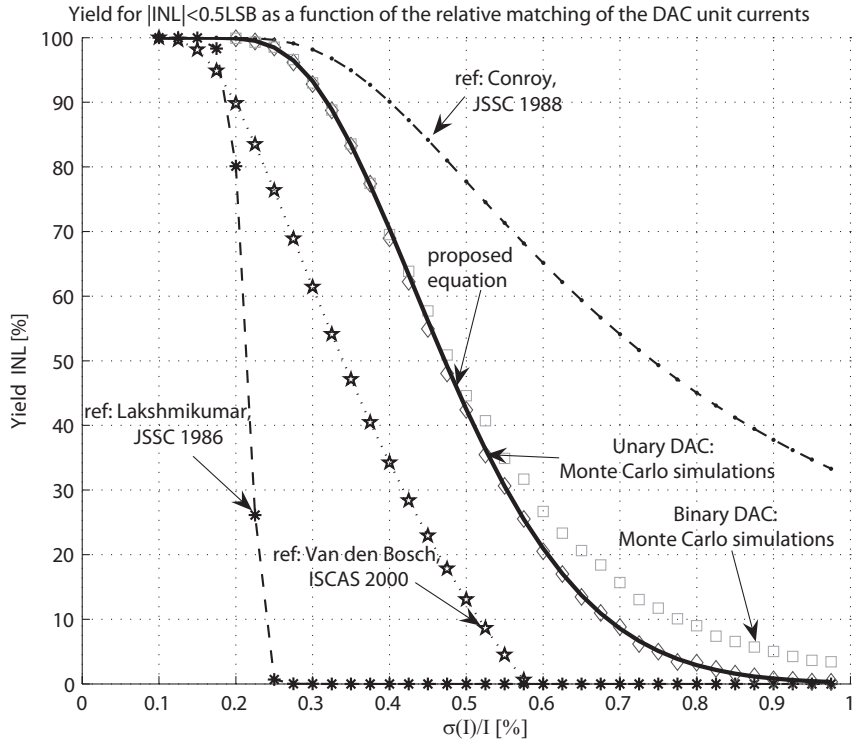


Figure 6.3-1 - 14bit DAC INL yield against the relative unit current mismatch: a comparison between references [51], [52], [10], this work (first 300 terms of Equation 6.2-15, shown as "proposed equation") and binary and unary DAC Monte Carlo simulations.

As commented, the approximation errors in [51], [52], [10] are bigger than in this work. Indeed, there are only two small approximations in this work. Firstly, the discrete INL characteristic is approximated as continuous in Equation 6.2-11. In practice, for $N > 7$ bits, this approximation error is negligible. Secondly, for higher $INL_{\max}$ barriers, the binary and segmented DAC $Yield_{INL}$ is approximated to the thermometer $Yield_{INL}^{(Thermo)}$. As shown in Figure 6.3-1, for $Yield_{INL}$ levels higher than 40%, the difference between the binary and thermometer cases become significantly smaller. The complete derivations of

$Yield_{INL}$ for binary and thermometer DACs are not included in this chapter as they go beyond its scope. See [59] and [48] for them and also for the derivations of all formulas presented in this chapter.

The presented work in this chapter is done in close collaboration with prof.dr. Remco W. van der Hofstad and dr. Markus Heydenreich from the department of mathematics and computer science of Eindhoven University of Technology.

6.4. CONCLUSIONS

To analytically approach the DAC INL, a new method based on a Brownian Bridge (BB) process is shown. For the first time, it is analytically proven that the INL properties of binary and thermometer DACs are not the same but rather they are different due to the different switching sequences of their analog elements. This chapter statistically quantifies this difference. On average, the binary DAC architecture features a smaller INL but this advantage is lost if a higher chip $Yield_{INL}$ is targeted. This chapter also shows that the INL distribution of thermometer and binary DACs are diffusive, scaling with the resolution and the mismatch of the analog elements. Furthermore, important relationships linking the DAC element mismatch with the DAC linearity are also derived. The results of this work fill a gap in the general knowledge on the DAC INL. They will help engineers to properly understand the error mechanisms beneath element mismatch and how they affect the DAC linearity. The analytical relations showing the links between the process spread and the DAC specifications (e.g. in terms of INL) allow a single synthesis design step, instead of a long-lasting iterative design procedure with time consuming Monte Carlo simulations.

7. Chapter

CLASSIFICATION OF ERROR CORRECTION METHODS, A BROAD VIEW

This chapter introduces a classification of DAC correction methods. A set of representative DAC correction methods is selected and three fundamental groups of methods are defined: Error-Transfer-Function-Prevention (ETFP), Error-Transfer-Function-Correction (ETFC), and Signal-Transfer-Function-Compensation (STFC). Then, three categories with dichotomous classes are defined along three different angles of incidence: error measurement, redundancy, and system level. The presented DAC correction methods are ordered and the missing methods are identified. Common properties of the methods that belong to the same classes are discussed. Clues are provided for the development of new correction methods that can fill-in the gaps. A discussion about the relevance of the introduced classification is provided and potential extensions of the classification are outlined.

7.1. INTRODUCTION

Chapter 5.1. and Equation 5.1-1 define the DAC transfer function as the sum of the nominal DAC signal transfer function (STF) and the unwanted DAC error transfer function (ETF). Figure 7.1-1 illustrates this simplification. There are various available correction methods that can counteract the unwanted ETF. The self-calibration correction methods can adjust the DAC current cells and hence correct for the mismatch-related amplitude and timing errors, e.g. [35] and [37]. The mapping errors can adjust the distribution (order) of the mismatch errors, creating mutual cancellation and effectively correcting for them, e.g. [1]. The input data reshuffling method, or Dynamic Element Matching (DEM), can adjust the time-average of the errors by randomly cycling through the DAC elements, [30]. The Return-to-Zero (RZ) method 'hides' the periods of switching from the DAC output, i.e. prevents (i.e. removes) a large portion of the data-dependent errors, and hence corrects for the switching disturbances [26]. The Differential-Quad Switching (DQS) method injects switching-related disturbances at every clock edge and hence removes a large portion of the data-dependent errors, [29]. The digital pre-distortion methods add extra non-linear error information in the DAC input signal to cancel the DAC non-linearity, [42].

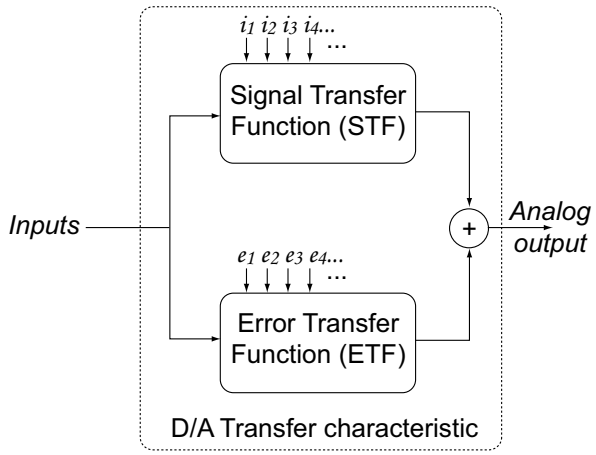


Figure 7.1-1. A reduction of the DAC transfer function into STF and ETF.

To order the already available knowledge on DAC correction methods, this chapter introduces a classification. The classification derives common properties of DAC correction methods, identifies missing methods, and provides clues for developing new correction methods. Chapter 7.2. presents the selected set of DAC correction methods and makes some basic definitions. It defines three main categories: error measurement, redundancy, and system level. Chapter 7.3. presents the error measurement category. Chapter 7.4. presents the redundancy category. Chapter 7.5. presents the system level category. Further, chapter 7.6. provides a general discussion on the relevance of the presented classification. Finally, conclusions are drawn in chapter 7.7.

7.2. SELECTED SET OF DAC CORRECTION METHODS AND DEFINITIONS

The following text discusses a selected set of DAC correction methods that can counteract some of the considered DAC errors in chapter 5. The relations among these diverse DAC correction methods are considered further in the text. The "gaps" in the available knowledge from the literature are identified and new methods to fill these "gaps" are proposed further in the thesis. The selected DAC correction methods are ordered in Table 7.2-1.

Table 7.2-1. Discussed DAC correction methods.

Group	*Correction method*	Main targets	Main challenges	Chap-ter
ETFP	*Return-to-zero*	Data-switching errors	Signal power, speed	3.2.
	Differential-quad switching	Data-switching errors	Error matching	3.3.
	Cascode switches with offset current	Data-dependent settling	Voltage headroom	3.4.
ETFC	*Self-calibration for timing errors*	Sampling moment errors	Timing errors measurement	4.2.2.
	Self-calibration for amplitude errors[1]	Current mismatch errors	Dynamic performance	4.2.1.
	Mapping for unary currents[1]	Current mismatch errors, Sampling moment errors	Occupied silicon area	4.3.1.
	Mapping for sub-DACs[1]	Current mismatch errors	Occupied silicon area	12
	Mapping for sub-binary radix DACs	Current mismatch errors	Dynamic performance	4.3.2.
	Input data reshuffling (DEM)	Current mismatch errors, Sampling moment errors	Occupied silicon area	3.5.
STFC	*Digital pre-distortion*	Linearity, DAC HD	Linearity measurement, occupied area	4.4.
	Suppression of HD[1]	DAC HD	Occupied silicon area	13

Three groups of methods are identified. The first group *prevents* the errors (e_j) from having any influence on the DAC output and hence prevents the ETF (for those errors). This group is named Error Transfer Function Prevention (*ETFP*). Examples include Return-to-zero (RZ) and Differential-Quad Switching (DQS). The second group *corrects* the errors and hence corrects the ETF (Error Transfer Function Correction (*ETFC*)). Examples include self-calibration, errors mapping and DEM). The third group *compensates* (modifies) the STF for the effect of the ETF (Signal Transfer Function Compensation (STFC)). Examples include digital pre-distortion and suppression of HD (introduced in this chapter).

A common requirement of all correction methods is that they cause a change in the core (intrinsic) D/A function – preventing errors, correcting errors, or modifying the input data to compensate for errors. Thus, the DAC correction methods change in some way the error effects to improve the DAC intrinsic performance. The change can be based on knowledge about the errors, e.g. recreating the error information/mechanism and using it in a different way, with exact knowledge of it. Alternatively, the change can be based on

[1] (With) an original contribution proposed in this thesis.

copying the error information/mechanism and using it in a different way, without exact knowledge of it. The change can be based as well on erasing the information, without again exact knowledge of it, etc. However, there is always a change in the error mechanism. The change is defined in this thesis as: preventing errors, correcting errors, or compensating the inputs (i.e. modifying the input data to compensate for errors).

Manipulating the error mechanism and hence improving the intrinsic DAC performance can occur *with or without exact knowledge on the DAC errors*. For example, self-calibration methods acquire a-posteriori knowledge of the errors, compute the needed correction and so adjust the error mechanism. However, it is indeed possible to adjust the error mechanism without such an exact knowledge. For example, the DEM methods average in time the error information and so adjust it to a time-averaged value per element.

Manipulating the error mechanism and hence improving the intrinsic DAC performance implies using suitable DAC redundancy. No correction method can exist without DAC redundancy. For example, an intrinsic single binary DAC core cannot be used in a correction method, because it features no intrinsic redundancy that the correction method can exploit to improve performance. Redundancy can be either *intrinsic* or *extrinsic*. The intrinsic redundancy is found in elements that are intrinsic for the DAC functionality but that still contain hidden and unused resources. For example, the DAC unary current cells represent intrinsic redundancy, since they are used for the DAC functionality but they still contain the hidden resources of multiple possible switching sequences. The extrinsic redundancy is found in elements that are not intrinsic for the DAC functionality but still used to correct its performance. To illustrate, these may even be called an “outside help”. For example, the correcting circuits in a self-calibration method are extrinsic elements for the DAC function, e.g. the CALDACs in Figure 8.5-4.

Finally, manipulating the error mechanism and hence improving the DAC intrinsic performance can occur at *low-* or *high system level*. The correction method can work with the DAC current cells or with the DAC as a block. For example, a self-calibration correction method can adjust the errors per every current cell, i.e. low-level correction. A return-to-zero output stage can reset the DAC output for every half a clock period to remove the errors during switching, i.e. high-level correction.

Figure 7.2-1 depicts the DAC correction methods as a process to change the effect of the DAC ETF on the D/A function to achieve performance improvement.

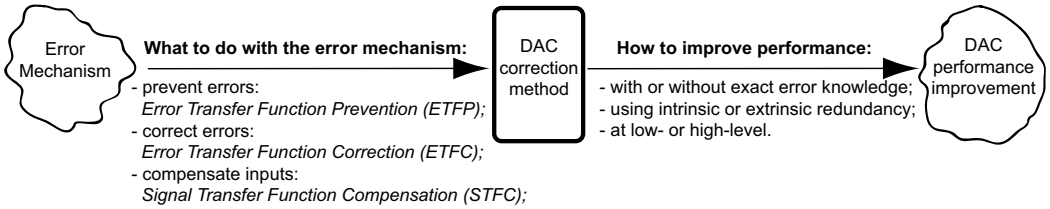


Figure 7.2-1. The process of changing the effects of the DAC ETF resulting in performance improvement.

Thus, *three groups* of correction methods are defined, depending on the approach against the errors. *Three categories* are defined along three different angles of incidence: *error measurement*, *redundancy*, and *system level*. Each category can be further split into dichotomous classes: with or without error measurements; intrinsic or extrinsic redundancy; low- or high-level. Most combinations among the three groups, the three categories, and two classes should be possible. Below, the text considers in details these combinations.

7.3. ERROR MEASUREMENT CATEGORY

The DAC correction methods can either use exact knowledge of the errors acquired through measurements or use the errors without having such exact knowledge. Accordingly, two classes of DAC correction methods are distinguished, see Table 7.3-1. Examples of using errors without knowing them are: discarding or duplicating unknown errors during a known period (RZ and the DQS methods remove the data-dependent errors created from the switching disturbances); and averaging unknown errors over known elements (DEM methods).

Table 7.3-1 orders the methods in these two classes and identifies examples from the literature for the three defined groups of correction methods – ETFP, ETFC, and STFC. Logically, there are no methods in the ETFP group that use exact error knowledge: by definition, ETFP prevents the errors, so they do not exist and cannot be measured.

Table 7.3-1. Error measurement category, with its two classes.

Group	*With error measurements*	*Without error measurements*
ETFP	*By definition N/A*	*Return-to-zero: [26], [27], Differential-quad switching: [29], Cascode switches with offset current: [20]*
ETFC	*Self-calibration: [35], [36] Mapping: [1], [11],* ***this work*** *Mapping for sub-binary radix DACs: [11]*	*Input data reshuffling (DEM):* [38] *Segmented DEM (parallel sub-DAC level): [60]*
STFC	*Digital pre-distortion: [42], [45]*	*Suppression of HD:* ***this work***
	Common characteristics: Analog error measurement (exact a-posteriori error knowledge, risks of performance deterioration due to the measurement circuits; results rely on the accuracy of the measurement; extra hardware resources; possible complications for IC process migration; etc.). Background and Foreground correction modes.	Common characteristics: No exact knowledge on the errors (e.g. discarding unknown error information, discarding unknown signal and error information during known period, averaging known information over known elements, duplicating unknown information from known source; etc.). Background mode only[1] (risks of performance deterioration due to background activity; increased power consumption due to background activity, increased noise).

The mapping and DEM correction methods are similar in how they implement the correction and different in the way they use the error information. They are in the same ETFC groups but belong to the different classes "*with error measurements*" and "*without error measurements*". However, for the digital pre-distortion methods from the class *"with*

[1] Note that the discussion here is not about the advantages and disadvantages of the background and foreground correction modes but about the fact that the correction methods that do not use error measurements cannot use foreground correction modes.

error measurements", there is no such corresponding method in the class "*without error measurements*". In the literature, no examples are found for the correction methods of the STFC group without exact error knowledge. However, it is possible to synthesize such a method based on the available methods from the same group but from the other class. The digital pre-distortion methods, e.g. [42], [45], subtract specific distortion information from the DAC input signal. The subtracted information is based on the exact knowledge of the DAC non-linearity. That results in two different harmonic error mechanisms: one injected from the outside and the other intrinsic to the DAC. Both error mechanisms subtract from each other and hence the DAC linearity is improved. It is indeed possible to apply the same technique but without having the exact information about the linearity. For example, the error information can be duplicated from a nominally identical source. A similar method to generate the inverse distortion function is already applied and validated in the RF field; it is reported in an independent research work [61]. This method can be applied to DACs, shown in the table as Suppression of HD and presented in chapter 13 of this thesis.

The correction methods in the "*with error measurement*" class share the common properties of analog error measurement and the possibility of both background and foreground operating modes. The background operating mode implies that the measurements and/or corrections are active while the DAC core performs the D/A conversion of the main signal[1]. The foreground operating mode implies that the measurements are done while the DAC core is not performing the D/A conversion of the main signal. Note that the correction is still active at background; the type of foreground correction, e.g. laser trimming, is beyond the scope of this thesis.

The analog error measurements provide accurate knowledge of the DAC errors. However, they also create the risk of DAC intrinsic performance deterioration, since the measurement circuits need to connect to the sensitive DAC analog circuits. Furthermore, the analog measurements need to be sufficiently accurate, which poses extra accuracy requirements on the design of the overall system. The portability of the design to other IC technologies should also consider this. The advantages of the foreground correction modes are that, once the errors are known, the correction can be static, so that there are no correction activities that can interfere with the normal operation of the DAC core. Static correction is also power efficient.

The correction methods in the "*without error measurements*" class share the common properties of having no exact error knowledge and using only background correction modes. Having no exact error knowledge is not important at all in some cases, e.g. RZ methods discard the switching periods anyway. However, the DEM methods that average the errors over time need to take into account the consequences of not knowing the errors. For example, the errors are transformed to noise. Moreover, since the errors are averaged in time, the DEM DACs have a mismatch dependent gain, which may be a problem for some low-voltage power-supply applications or for applications where gain is important. The other common property is that all methods that do not have the exact error knowledge need to be implemented always in background mode, which creates risks for the DAC intrinsic performance, since there are always correction activities during the normal operation of the DAC. The background correction methods consume more power than the foreground methods, because their correction circuits cannot be static and hence need power. However, their active correction can take care of errors that are slowly changing due to e.g. temperature changes.

[1] The terminology "main signal" is used here to distinguish from a test signal.

7.4. REDUNDANCY CATEGORY

The DAC correction methods can use either intrinsic or extrinsic redundancy for the correction. Accordingly, two category classes are defined: the *intrinsic redundancy* class and the *extrinsic redundancy* class, see Table 7.4-1. Here, intrinsic redundancy is defined as redundancy in the circuits that are native to the DAC core and directly used in its D/A function. For example, the DAC unary currents contain intrinsic redundancy: they are directly used in the conversion, but still have the potential of different switching sequences. Extrinsic redundancy is defined here as redundancy in circuits which are outside the DAC core and are only indirectly used in its D/A function, but which help to improve the performance. For example, the CALDACs in Figure 16.2-1 represent extrinsic redundancy. Although D/A function can exist without them, they can correct the errors of the DAC current sources and hence improve the performance. Note that the redundancy category applies only to the correction part (redundancy used for error measurement is always implemented with extrinsic redundancy).

Table 7.4-1. Redundancy category, with its two classes.

Group	*Intrinsic redundancy based*	*Extrinsic redundancy based*
ETFP	*Differential-quad switching: [29]*	*Return-to-zero: [26], [27]* *Cascode switches with offset current: [20]*
ETFC	*Mapping: [1], [11],* ***this work,*** *Input data reshuffling (DEM):* [38], [60]	*Self-calibration: [35], [36]*
STFC	*Suppression of HD:* ***this work***	*Digital pre-distortion: [42], [45]*
	Common characteristics: Risks of insufficient own resources for correction	Common characteristics: Risks of deterioration of intrinsic DAC performance Increased complexity

Two methods use the intrinsic redundancy in the unary currents : the mapping technique, which chooses the most appropriate sequence and the DEM technique, which randomly varies in time the switching sequence. Both techniques belong to the ETFC group. Examples of the “extrinsic redundancy” ETFC group are found in the self-calibration methods. Extrinsic correction circuits adjust the mismatch errors according to the error information acquired from measurements. Examples of the ETFP group extrinsic redundancy are found in the RZ methods and the “Cascode switches with offset current” methods, where extra circuits remove the switching-related errors from the DAC output. For the ETFP group, examples of the intrinsic redundancy class are found in the DQS methods, where all transistors are used for the correction and the actual D/A conversion process. Examples of the STFC group extrinsic redundancy are found in the DAC digital pre-distortion correction methods: based on the measured non-linearity information, an inverse non-linearity function is generated in the digital domain. This inverse error information is extrinsic for the DAC and it is added to the input signal to cancel the DAC own intrinsic non-linearity errors.

For one box of Table 7.4-1, the STFC group using intrinsic redundancy, examples are not found in the literature. However, it is possible indeed to synthesize such a method based on its counterpart from the extrinsic redundancy class – the digital pre-distortion method. While the added error correction in the digital pre-distortion method is extrinsic

for the DAC core, it is intrinsic for the method of suppression of HD, developed in this work. The identified other similar correction methods are the DQS and the RZ methods; Mapping/DEM and self-calibration; suppression of HD (presented in this paper) and digital pre-distortion.

The DAC correction methods from the intrinsic redundancy class share the common characteristic that the intrinsic resources may give insufficient error correction. For example, for the mapping methods it is possible that there are no available maps that can sufficiently adjust the errors for a particular DAC code. Furthermore, no single map may be found that sufficiently improves the DAC performance at both low and high speeds, because the static and dynamic errors for the individual current cells are different. Concerning the DEM, particular mismatch errors through the process of averaging are transformed to a DAC gain error. Though the DAC gain error does not generate harmonic distortion, the gain may be a problem for low voltage supply applications or in general for applications that are sensitive to gain errors. In addition, the mismatch errors are transformed to a noise floor that may still be a problem for some applications.

The DAC correction methods from the extrinsic redundancy class share the following common characteristics: risks of deterioration of the intrinsic DAC performance and increased complexity. Since the extrinsic redundancy implies adding extra circuits to the DAC intrinsic core, it is always possible that these extrinsic circuits deteriorate in some way the DAC performance. This argument is even stronger at high speeds, where the parasitic capacitances and resistances strongly influence the DAC dynamic performance. For example, the RZ methods at high speeds are not effective because the currents have insufficient time to properly charge all parasitic capacitances. Self-calibration methods are also not that efficient at high speeds mainly due to the parasitics of their correction circuits.

7.5. SYSTEM LEVEL CATEGORY

The DAC correction methods can be applied to either *low- or high system level*. Accordingly, dichotomous classes within the system level category are defined, see Table 7.5-1. Low level corrections are those correction methods that work with the DAC elements. High level corrections are those correction methods that work with the DAC as a whole.

The RZ method is represented in both classes; an example of low-level is found in [27], and of high level in [26]. The self-calibration methods too: low-level in [35], while high-level in [36]. In the literature, the mapping technique is found only applied at low level. This thesis proposes a high-level mapping technique which uses multiple parallel sub-DACs, realized via ane appropriate distribution of the DAC input digital word among the parallel sub-DACs. The DEM method is represented in the literature in both low- and high level variants: the work of [38] applies DEM to the DAC unary current cells, while the segmented DEM of [60] applies DEM to two parallel sub-DACs.

For the STFC group, examples are found for the high-level class - the DAC pre-distortion methods. This thesis proposes another high-level method in the STFC group – the method of suppression of HD. Low level DAC correction methods from the STFC group are not found in the literature. However, in [62]an example is suggested, though this work is not related to the DAC field: intrinsic error information is added in a V-I converter through degeneration switch resistors to improve its linearity.

Table 7.5-1. System-level category, with its two classes.

Group	*Low-level correction method*	*High-level correction method*
ETFP	*Return-to-zero current cell: [27], Differential-quad switching: [29], Cascode switches with offset current: [20]*	*Return-to-zero output stage, e.g. [26]*
ETFC	*Self-calibration, e.g. [35], Mapping for unary currents, e.g. [1], Mapping for sub-binary radix DACs, e.g. [11], Input data reshuffling (DEM), e.g.* [38]	*Self-calibration, e.g. [36], Mapping for sub-DACs, e.g.* ***this work,*** *Segmented DEM (parallel sub-DAC level), e.g. [60]*
STFC	*Possibly, V-I converter linearization (beyond DAC field), like in [62]*	*Digital pre-distortion, e.g. [42], [45], Suppression of HD, e.g.* ***this work***
	Common characteristics: Increased hardware resources (generally); Complexity (generally);	Common characteristics: Reduced dependence on DAC architecture (The intrinsic DAC core can be designed independently from the correction method);

The methods from the low-level class share the following characteristics: increased hardware resources and increased system complexity. For example, if the RZ method is implemented at low level, each current cell needs to have extra switch transistors and the switching control circuitry becomes complex, e.g. [27].

The methods from the high-level class feature only limited dependence on the DAC intrinsic architecture. They consider the DAC as a function that needs to be corrected. That is why they can be independent of this function. For example, the RZ output stage of [26] can be applied to different DACs. Because the correction method and the DAC architecture are independent, both can be separately optimized for different targets, e.g. linearity, area, power consumption, reliability, or flexibility as proposed in this thesis in chapter 12. Moreover, the DAC correction circuits can be reused with different intrinsic DAC cores.

7.6. DISCUSSION

The presented classification introduces an order in the knowledge on the DAC correction methods. It helps to quickly evaluate a new DAC correction method and identify its advantages and disadvantages by judging the common characteristics. For example, the low-level mapping methods generally feature increased complexity, as suggested by Table 7.5-1. Thus, a new low level mapping method is developed and presented in chapter 11 that specifically targets reducing complexity compared to its peers and so exchanges improvement potential for simplicity.

The presented classification also indentifies missing correction methods. However, it also provides clues for developing new methods that may fill in the gaps. For example, the DEM methods are identified to be close to the mapping methods. Thus, based on the clues of the high-level segmented DEM methods [60], a new high-level mapping method is proposed in chapter 12. Another example is the suppression of HD method, presented

in chapter 13, which is suggested by the “digital pre-distortion” method in Table 7.3-1 and Table 7.4-1.

A classification as used above based on the three defined groups ETFP, ETFC, and STFC can be used beyond the field of DACs, since Figure 7.1-1 can be applied to other fields, e.g. ADC. Further, the three defined categories: error measurement, redundancy and system level can be appended with other categories to refine the classification or to apply it to other fields. For example, a possible extension may include static versus dynamic error corrections for the field of DACs; a new category for the field of ADCs may include analog versus digital methods.

7.7. CONCLUSION

This chapter introduces a classification of the DAC correction methods. It considers a selected set of correction methods, which are ordered in three fundamental groups: Error-Transfer-Function-Prevention (ETFP); Error-Transfer-Function-Correction (ETFC); Signal-Transfer-Function-Compensation (STFC). Then, three categories with dichotomous classes are defined along three different angles of incidence: error measurement, redundancy, and system level. The classification derives common properties of the correction methods that belong to the different classes. The classification identifies missing methods and provides clues for new correction methods that can fill-in the identified gaps. Examples are provided with references to this thesis of new methods, the synthesis of which is assisted by the presented classification. It is argued that the presented classification will help engineers to quickly evaluate new correction methods even beyond the field of DACs.

8. Chapter

ANALYSIS OF SELF-CALIBRATION OF CURRENTS, AN IN-DEPTH VIEW

This chapter systematically analyses one member of the correction methods that are presented in chapter 4 and 7: the self-calibration of the current mismatch errors in Digital-to-Analog Converters (DACs). The self-calibration components are classified in three groups: self-measurement, error processing algorithm and self-correction (where the addition “self” distinguishes self-correction from the more general term “correction”).

The discussed self-measurement elements are measurement probe, references, and measurement devices. The discussed algorithms cover unary-current calibration and binary-current calibration. The discussed self-correction elements are the self-correction method, self-correction circuits, and self-correction memory.

The chapter systemizes possible and known alternative implementations and their associated trade-offs. The findings are compared to the available solutions in the literature and missing techniques are identified. Calibration of DAC binary currents is pointed to as an important missing technique and a solution is proposed in chapter 10.

It is argued that self-calibration of currents is the only DAC correction method that can correct both random and systematic current mismatch errors that cause both non-linear and linear DAC errors. That is why it is chosen for an in-depth analysis in this chapter. The chapter provides a broad overview, in a systematic way, of the DAC current self-calibration methods that will help engineers to develop an approach for evaluating new DAC correction methods, in general, and DAC current calibration methods, in particular.

8.1. INTRODUCTION

Transistor mismatch of the DAC current cells cause both amplitude and timing errors, as discussed in chapter 5, in relation to Figure 5.3-2 and Figure 5.3-3. It causes general mismatch among the DAC current cells, which results in different error responses to nominally identical error sources. For example, the disturbance at the common source node of the current switch transistors is different when the current cells conduct different currents due to the current mismatch, as discussed in relation to Figure 5.4-1. Thus, the current mismatch of the DAC current cells deteriorates both the DAC static accuracy and the DAC dynamic performance. On the one hand, these errors deteriorate both the DAC performance and the chip yield. On the other hand, their mismatch origin makes the DACs as a whole, and particularly the DAC accuracy, sensitive to the fabrication IC process. The transition to other IC processes is not guaranteed and the redesign requires considerable effort.

To solve these problems, the DAC accuracy needs to be robust. It needs to be technology independent and guaranteed by design. For example, the DAC accuracy should not depend on a manufacturing process parameter, e.g. A_β and A_{VT}, see Equation 2.3-2, but rather *the DAC accuracy should be a design parameter* (at least for the first order approximation).

Self-calibration provides a solution. The post-calibration errors are small and controllable. Their statistical distributions are functions of design parameters and hence not functions of the tolerances of the CMOS process A_β and A_{VT}. Self-calibration of currents, or simply self-calibration, distinguishes itself from the other correction methods discussed in chapter 4 in the way the errors are corrected. Firstly, the errors are measured and then the exact correction is generated to adjust for these errors. That is why this is the correction method that can theoretically lead to the highest degree of technology and design independence when compared to the other correction methods, e.g. mapping.

The evolution of the CMOS IC technologies encourages self-calibration. Though smaller transistors feature greater mismatch errors, the feedback mechanism of the self-calibration can correct these errors. The on-chip integration of additional functions, such as a calibration engine, is relatively inexpensive. Unfortunately, most of the self-calibration methods that are available in the literature cannot achieve full technology independence, do not use efficient and robust circuits, and do not maintain the calibration improvements at high speeds.

Section 8.2. proposes a classification of the DAC self-calibration methods. Section 8.3. considers self-measurement aspects, section 8.4. algorithm aspects, section 8.5. self-correction aspects. Finally, conclusions are drawn in section 8.6.

8.2. DAC CURRENTS SELF-CALIBRATION CLASSIFICATION

Self-calibration is a general approach to improve a performance parameter of a system by measuring and correcting some of its errors. The three defining elements of DAC currents self-calibration are self-measurement, integrated algorithm (error processing), and self-correction. Figure 4.2-1 and Figure 8.2-1 show conceptual diagrams of generalized self-calibrating current-steering DAC. The three self-calibration elements are shown in boxes together with some of their main sub-elements (in some cases properties). They distinguish the self-calibration approach from other approaches that can improve performance, e.g. dynamic-element-matching (DEM) [63] and input data-reshuffling [30], harmonic cancellation [64], [61], and error mapping [65]. The self-

calibration corrects the origin of the error, while the other smart techniques work on its effects. Therefore, only the self-calibration can achieve technology and design independence. It is equally effective against both random and systematic mismatch errors.

In relation to the DAC correction methods classification presented in chapter 7, the self-calibration is a method that uses error measurement, extrinsic redundancy, and either high- or low level error measurement and correction. The self-calibration method is close to the mapping methods (chapter 4.3.) in terms of using a-posteriori knowledge of the mismatch errors. Both methods can also be applied to low and high levels. However, the self-calibration methods use extrinsic DAC redundancy, while the mapping methods use intrinsic DAC redundancy. Therefore, the error correction is guaranteed by design for the self-calibration methods, while it is statistically guaranteed for the mapping methods. For example, the self-calibration method is much more effective against systematic (design) errors than the mapping methods. However, noting that the extrinsic redundancy poses risks for the intrinsic DAC performance (particularly at high speeds), as discussed in chapter 7.4. , the design of every self-calibration engine needs to be carefully considered.

See Figure 8.2-1. The possible calibration targets are the DAC signal current sources, biasing currents, switching moment instances, data drivers, and DAC output itself. The self-measurement block extracts the error information from either the DAC core or the DAC output. The algorithm block processes the error information and controls the whole operation. The self-correction block can correct either the DAC core or the DAC output. The relations between the three calibration elements and the DAC core can be background, foreground, or both. A background relation is when the blocks work together, while the DAC core performs the D/A conversion of the input data signal. In a foreground relation, the blocks work together while the DAC core does not perform D/A conversion. As shown in Figure 8.2-1, some of the relations are only possible in a single mode, while others in either background or foreground mode.

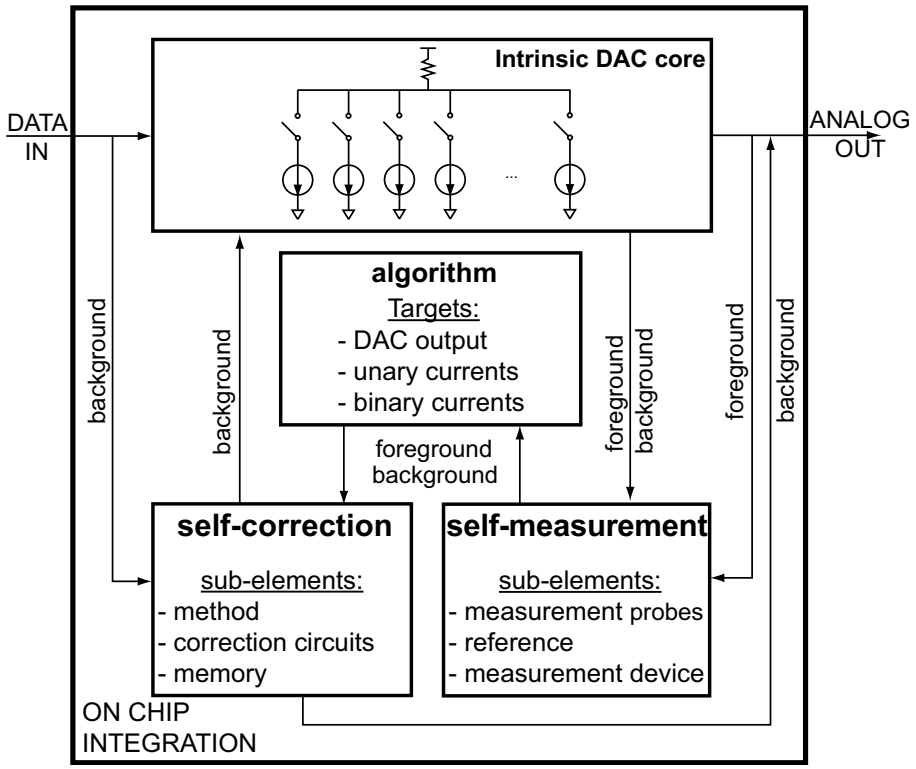


Figure 8.2-1. A conceptual block diagram of a generalized self-calibrated CS DAC.

This chapter concentrates on the static self-calibration of the DAC currents. The three prime elements of self-measurement, algorithm, and self-correction are examined. The self-measurement prime element may be considered as common for all DAC correction methods that rely on error measurements (chapter 4). The algorithm prime element is always specific to the particular DAC self-calibration method. The self-correction prime element represents extrinsic DAC redundancy and is the property that distinguishes the self-calibration from the mapping methods.

8.3. SELF-MEASUREMENT

The calibration self-measurement block focuses on the extraction of the actual errors. The measured errors are the deviations of the examined electrical quantities from their expected values. The self-measurement has three sub-elements: measurement probes, measurement device, and the reference, see Figure 8.2-1.

8.3.1. Measurement probes

The measurement probes are the circuitry that makes possible the actual error measurement by the measurement device. There are four main requirements for the measurement probes: mutual suitability with the other calibration components, minimum degradation of the DAC performance, minimum alteration of the measured quantity, and minimum silicon area.

The measurement probes can be implemented in the upper, middle, or lower range of the available voltage headroom. Figure 8.3-1 (upper voltage headroom range), Figure

8.3-2 (lower voltage headroom range), and Figure 8.3-3 (middle voltage headroom range) show examples of DAC current cells with self-measurement probes, where the measurement probes are marked in gray rectangular boxes. The measurement probes in the upper and lower ranges of the voltage headroom leave only little voltage headroom for the measurement device. They allow only measurements in the voltage domain. The measurements need a voltage-input ADC, e.g. a sigma-delta ADC [36], [66], [67]. The measurement probes in the middle range of the voltage headroom leave more voltage headroom for the measurement circuits. They allow measurements by means of current based circuits, which in general are more efficient, because they do not implement the conversion from current to voltage.

8.3.1.1. Measurements at the upper voltage headroom

Figure 8.3-1a shows a straightforward way of measuring the DAC static error: the error information is directly measured at the DAC output. An example of this approach is demonstrated in [36]. This self-measurement is only possible in a foreground mode. The measurement device adds a capacitive load to the DAC output that deteriorates the DAC dynamic performance. To avoid the extra capacitive load, the current can be redirected to on-chip measurement termination resistors, as shown in Figure 8.3-1b, based on [68] and [65].

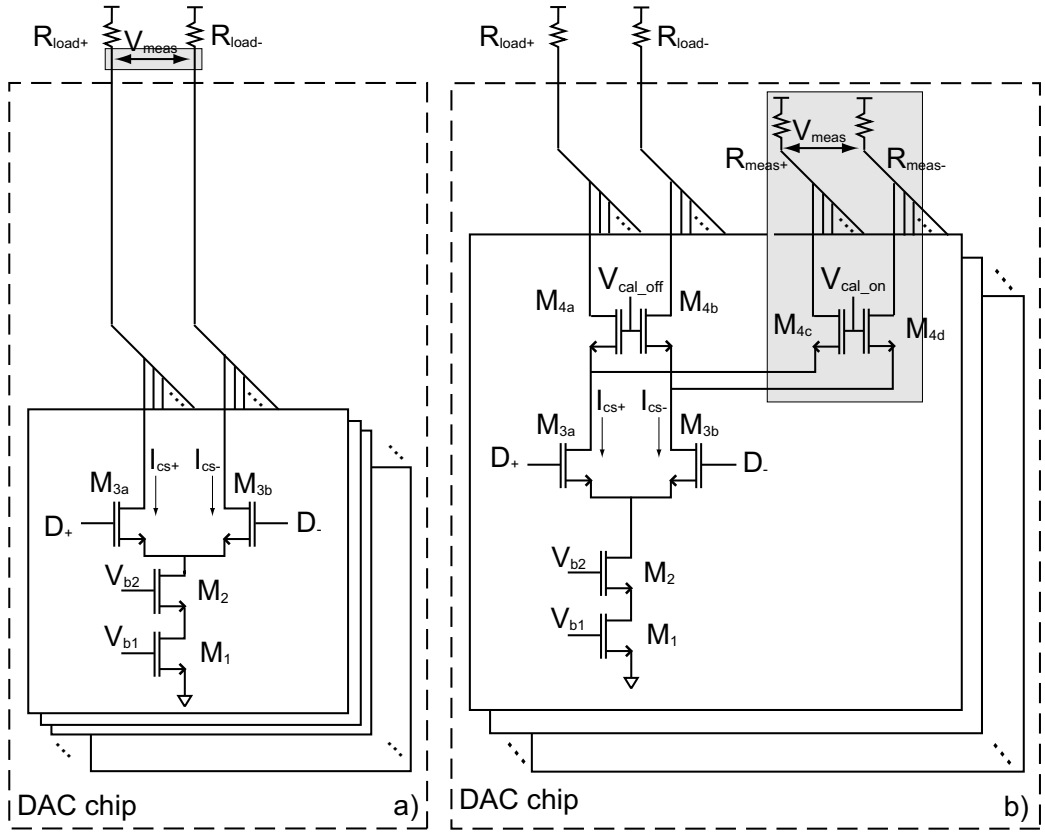


Figure 8.3-1. Examples of DAC current cells with self-measurement probes in the upper range of the voltage headroom: a) measurement of the DAC output, see [36]; b) measurement of a dummy DAC output, based on [68], [65];

8.3.1.2. Measurements at the lower voltage headroom

Figure 8.3-2 shows an example of extracting the measurement current in the lower range of the voltage headroom. Examples of this approach are demonstrated in [66], [67]. The extraction is realized in a true background mode. The current source transistor M_{bp1} generates a bias current. The regulated cascode M_{bn1}- M_{n1} controls the voltage of the source of M_{p1} and hence regulates its V_{gs}. Through the drain of M_{n1}, the so generated current flows to the data switches M_{2a} and M_{2b} for the actual D/A conversion. Through M_{p1}, the so generated current flows to the measurement switches and the measurement resistors. Thus, the current can be simultaneously used for measurements and a D/A conversion.

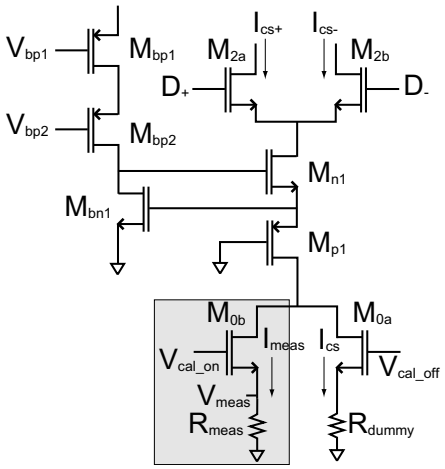


Figure 8.3-2. A DAC current cell with self-measurement probes in the lower range of the voltage headroom, from [66].

8.3.1.3. Measurements at the middle voltage headroom

Figure 8.3-3 shows ways to deviate the current from the main D/A conversion flow in the middle range of the voltage headroom, facilitating the use of a current-input comparator. These examples are more appropriate for foreground self-measurement, though [69] demonstrates an approach to use this type of architectures for background measurements through combining them with Return-To-Zero output DAC circuitry. When M_{2a} is turned on and M_{2b} is turned off, the M_1 current can be measured. When M_{2a} is turned off and M_{2b} is turned on, the current is used for D/A conversion. Note that the mismatch between M_{2a} and M_{2b} transforms into mismatch between the measured current I_{meas} and the current for D/A conversion I_{cs}.

To reduce the effect of the mismatch between the measurement switches, the architecture of Figure 8.3-3b uses a common cascode transistor M_2 between the current source transistor M_1 and the calibration switches M_{3a} and M_{3b}. An example of this approach is demonstrated in [35] and further in this thesis (chapters 10 and 16). The example of Figure 8.3-3c reduces the number of transistors in series, while using a common cascode transistor for both I_{meas} and I_{cs} by placing the measurement switch in parallel to the data switches. An example of this approach is reported in [69]. To measure I_{meas}, the data D/A switches M_{3a} and M_{3b} need to be simultaneously switched off.

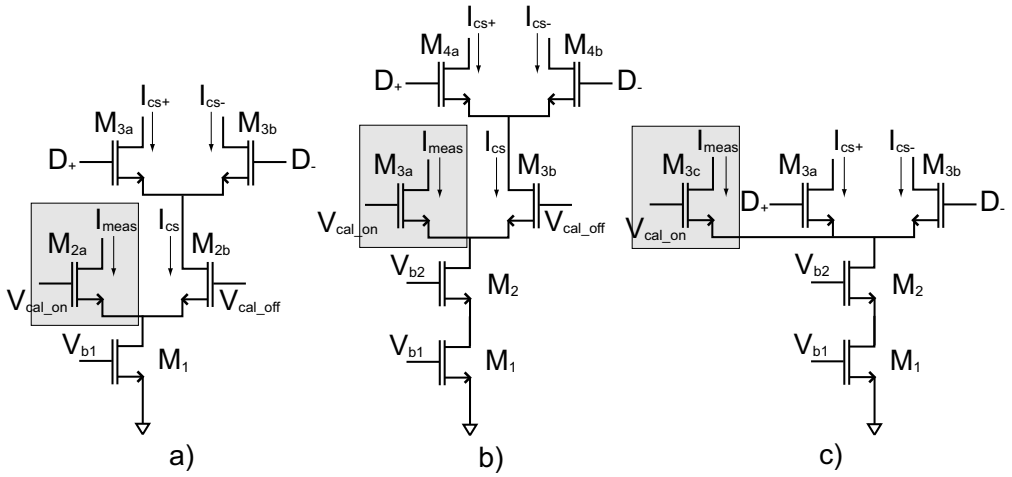


Figure 8.3-3. Examples of DAC current cells with self-measurement probes in the middle range of the voltage headroom: a) current deviation at the first cascode; b) current deviation at the second cascode; c)current deviation at the current D/A switches;

8.3.1.4. Deterioration of the intrinsic DAC performance, discussion and comparison

The addition of the measurement probes in the DAC current cell unavoidably causes some deterioration of the DAC core performance. The deterioration depends on the architecture of the measurement probes, the architecture of the DAC, the CMOS process, the available voltage headroom, and the particular realization.

The measurement probes can deteriorate the DAC performance through two dominant error mechanisms: the decreased impedance of the current cell and the increased charge feed-through due to the current switching. Too low output impedance of the current cell makes the generation of the static current unstable and hence vulnerable to external disturbances. Too low output impedance also causes rise of non-linear HD, deteriorating the DAC dynamic performance. According to [20] (see also [2] for HD2 analysis):

Equation 8.3-1

$$HD_3 = \left[\frac{nR_L}{4|Z_0|} \right]^2,$$

where Z_0 is the DAC output impedance per unit cell, R_L is the DAC load, n is the number of DAC unit cells.

Too large charge feed-through due to the data switch transistors is another problem. It causes two error mechanisms: direct disturbance of the DAC output and spikes at the common switch node. Since the charge feed-through is data dependent, these error mechanisms cause harmonic distortion of the DAC dynamic performance.

Table II comparatively summarizes the general advantages and disadvantages of the selected measurement architectures. The presented examples have specific advantages and disadvantages, briefly discussed below. However, only the example of Figure 8.3-3b

balances between the selected criteria. This is the only solution that does not have strong disadvantages, while it relaxes the requirements for the measurement device.

Table 8.3-1. Comparison of measurement probes and their relations to the DAC performance

Archi-tecture	Impe-dance	Charge feed-through	Meas. current change	Occupied area	Voltage headroom	Notes
Figure 8.3-1a	--	++	++	++	++	DAC output loaded with the capacitance of the measurement ADC.
Figure 8.3-1b	++	+	++	-	+	Little voltage headroom is left for the measurement device.
Figure 8.3-2	-	++	++	--	--	Large voltage headroom is required and resistors are used.
Figure 8.3-3a	++	++	--	++	++	Errors, $I_{meas} \neq I_{cs}$, due to the mismatch in the measurement switches.
Figure 8.3-3b	+	+	+	+	+	The advantages and disadvantages are balanced.
Figure 8.3-3c	++	-	+	--	++	Extra area is required for circuitry to shut down the data switches.

8.3.1.4.1. Current source impedance

The impedance of the measurement architecture of Figure 8.3-1a is significantly affected by the output capacitive load of the measurement device. The small circuit adjustment of Figure 8.3-1b removes the extra output capacitive load. The example of Figure 8.3-2 has low impedance, too. It requires large voltage headroom that makes it difficult to use transistor cascodes. Moreover, it needs both large nmos and pmos transistors to generate the current. The architecture group of Figure 8.3-3 features relatively good impedance levels. The example of Figure 8.3-3b uses 2 cascode transistors and hence for the same voltage headroom it needs to use bigger data switching transistors M_{4a} and M_{4b} than the examples of Figure 8.3-3a and Figure 8.3-4c. Therefore, at higher frequencies the impedance of Figure 8.3-3b is smaller than the impedance of the similar architectures of Figure 8.3-3a and Figure 8.3-3c.

8.3.1.4.2. Charge feed-through

Concerning the charge feed-through, only the architectures that add extra capacitive load around the data switching transistors may affect the overall DAC dynamic performance. The example of Figure 8.3-1b doubles the capacitive load at the drain of the data switches. The example of Figure 8.3-3b does not add extra transistors around the data switches but needs bigger transistors to implement the data switches, because of the two cascode transistors on top of the current source. The example of Figure 8.3-3c adds more capacitive load at the sources of the data switches. Any charge feed-through modulates the voltage at the common node and hence consumes signal current to compensate it, unavoidably causing data dependent errors.

8.3.1.4.3. Alteration of the measured current

Furthermore, the measurement probes should truly represent the measured quantity. In most of the cases, it unavoidably introduces errors in the measurement process. The measured quantity differs from the actual quantity that is used in the D/A conversion process. This difference limits the achievable post-calibration accuracy. Considering the discussed circuits, only the circuit of Figure 8.3-1a does not introduce changes in the measured current, because the current is directly measured at the output of the DAC. All the other circuits introduce some measurement error, i.e. $I_{meas} \neq I_{cs}$. They redirect the current from the main D/A conversion circuitry to the measurement device and hence they measure the errors of the measurement switches, too. Based on simulations for 10% mismatch of the V_{th} of the measurement switches, the example of Figure 8.3-2 shows the best results – only 0.003% difference between the actual current and the measured current, which should guarantee more than 28 bit of measurement accuracy. The example of Figure 8.3-3a guarantees only 15 bit of measurement accuracy. The examples of Figure 8.3-3b and Figure 8.3-3c guarantee approximately the same measurement accuracy – about 23 bits.

8.3.1.4.4. Occupied silicon area

Concerning the occupied silicon area, the example of Figure 8.3-1a does not consume extra area for the measurement probes. The examples of Figure 8.3-3a and Figure 8.3-3b consume only one extra switch transistor to realize the measurement probes. The example of Figure 8.3-3c consumes one extra measurement switch and dedicated circuitry to shut down the data D/A conversion switches, while measuring. The example of Figure 8.3-2 requires both nmos and pmos transistors to realize the floating current source architecture. It also requires one extra measurement switch and a resistor. From the considered examples, its measurement probes consume the most resources.

8.3.1.4.5. Voltage headroom range

Concerning, the voltage headroom, the example of Figure 8.3-2 requires significantly more voltage headroom than the others. The examples of Figure 8.3-1b and Figure 8.3-3b need to stack 4 transistors in series, while the examples of Figure 8.3-3a and Figure 8.3-3c need to stack only 3 transistors in series.

8.3.2. Reference

The measured quantity is evaluated against a reference, e.g. reference current I_{ref}. When only a portion of the DAC is calibrated, then the mismatch between the calibrated DAC portion (e.g. MSB unary currents) and the non-calibrated DAC portion (e.g. LSB binary currents) is a major source of a DAC post-calibration error.

When the DAC MSB unary currents should be only calibrated, then the mismatch between I_{ref} and the non-calibrated binary currents can be removed by constructing the reference I_{ref} as the sum of all binary currents plus one dummy LSB, such as illustrated later in Figure 8.3-6b and indicated in Equation 8.3-2. The sum nominally equals the calibration target, i.e. the MSB unary current.

Equation 8.3-2

$$I_{ref} = \underbrace{\sum_{j=1}^{B} I_{b,j}}_{\substack{all\ binary \\ currents}} + \underbrace{I_{b,1}}_{\substack{dummy\ LSB \\ current}}$$

This approach transfers the cumulative mismatch errors of the binary currents into I_{ref}. Through calibration against I_{ref}, these mismatch errors are transferred to a DAC gain error.

For applications that are sensitive to gain errors, an alternative approach [69] is to calibrate all binary currents plus the dummy LSB, as a group, to I_{ref} :

Equation 8.3-3

$$\underbrace{\left(\sum_{j=1}^{B} I_{b,j} + I_{b,1} \right)}_{adjust\ as\ a\ group} + \overline{I_q} \approx I_{ref} ,$$

where $\sum_{j=1}^{B} I_{b,j}$ is the sum of all B binary currents, $I_{b,1}$ is one dummy LSB current, and $\overline{I_q}$ is the expected post-correction error (can be either negative or positive).

However, this approach assumes the risk that the errors within the binary calibrated group are not proportionately distributed (i.e. equally with respect to the binary weights). Thus, a repetitive error would appear in the INL DAC characteristic.

8.3.3. Measurement device

The measurement device quantifies the measured error with respect to the reference. The error information can be produced in either the analog or digital domain. In the analog domain, the measurement is instantaneous and usually simple to make. The measurement device can be based on a passive switch or an amplifier in a feedback loop that adjusts the calibrated current [34], [69]. However, the error information processing in the analog domain is difficult. The measurement process is sensitive to process variations, various disturbances, noise, etc. Therefore, the error processing in the digital domain is often preferred [35], [70], [66]. It is more robust and suitable for advanced error processing. The measurement device is an ADC (analog-to-digital converter). It quantifies the error in the digital domain. For the calibration of CS DACs, the measurement ADC can be either current or voltage based. Further, it can be either a high resolution (multi bit) ADC or a comparator (single bit ADC). The requirements of the measurement device are small measurement error, small area, and high power efficiency.

8.3.3.1. Fully analog measurements

One of the first presented calibration approaches is [34], see Figure 8.3-4a. It is based on the switch SW_1 and the gate-source capacitor C_{gsM1}. This approach calibrates I_{cs} to I_{ref} with an offset measurement error due to accuracy limitations of the architecture. One of the main error origins is the different impedance levels at the drain of M_1 during the calibration phase (SW_1 closed) and the normal operation phase (SW_1 opened). The work

of [69] suggests an improvement by adding more feedback gain: the measurement device appends the switch SW_1 with an amplifier and a feedback loop, see Figure 8.3-4b.

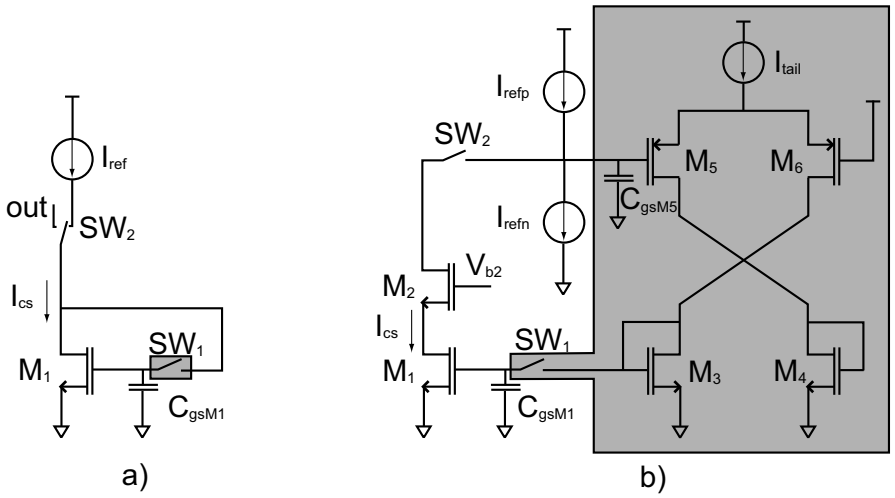


Figure 8.3-4. Fully analog measurements based on a feedback loop: a)with a simple passive switch from [34]; b)with an active feedback gain, from [27].

The two most important advantages of the analog measurement techniques are speed and simplicity. The measurement results are instantaneously ready. The background calibration approaches may need short time measurements to avoid major disturbances both from and to the D/A conversion process, since the analog measurements are usually used in a background mode. The support and control circuitry is much smaller than those used by the ADC based measurements. However, the analog measurement/correction technique has some major drawbacks when compared to the digital techniques, as discussed further in the chapter, see Figure 8.5-2. Among these drawbacks, the major examples are noise, limited loop gain, input offset of the measurement device, sensitivity to IC process parameters, limited error processing capabilities (binary currents cannot be calibrated).

8.3.3.2. ADC-based measurements

An alternative to the fully analog measurements are the ADC-based measurements. Their main advantage is that the error information is available in the digital domain, where advanced error data processing can be applied.

8.3.3.2.1. Multi bit voltage ADC-based measurements

Figure 8.3-5 shows multi bit voltage measurements. The measurement results are available in short time. However, the multi bit measurement ADC devices require significant design resources and effort. Their measurement error should be both small and guaranteed. They need to be linear and power efficient, too.

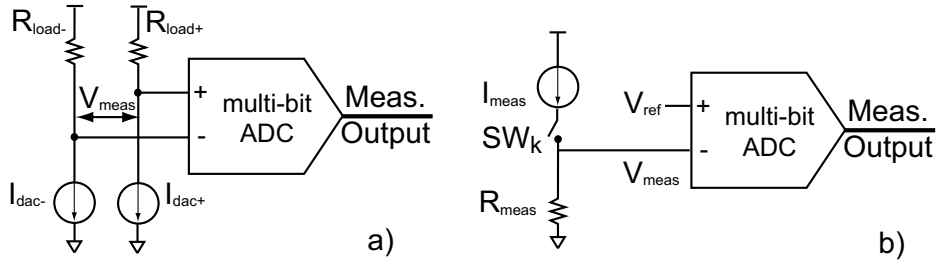


Figure 8.3-5. Multi bit voltage domain measurement ADC: a) measurement of the DAC output; b) measurement of current.

Figure 8.3-5a shows an example of voltage measurements at the output of the DAC, e.g. demonstrated in [36]. The ADC directly measures the DAC transfer characteristic. The measurement acquires a true representation of the DAC static performance. Note that the input offset error of the ADC is only a linear error, for it is a common error of all measured points. However, the measurement can only be executed in foreground and hence the advantage of the short measurement time window is lost.

Figure 8.3-5b shows a way to measure a current source, instead of the DAC transfer characteristic [68]. Thus, indirect information about the DAC transfer characteristic can be acquired. The measured current source is terminated with a load measurement resistor R_{meas}. A voltage drop V_{meas} is produced and the ADC measures it against a reference voltage V_{ref}. SW_k is the switch that connects the k-th measured current to the ADC. This method is demonstrated in [66], [67]. It is suitable for background measurements, because the measurement results are available in short time. However, the requirements for the ADC may be difficult to meet. The input offset can be a problem if only some selected currents are measured and calibrated, while the rest remain uncalibrated. Furthermore, the problems influence all the ADC requirements, e.g. sufficient linearity, small area and power efficiency.

8.3.3.2.2. Single bit voltage ADC-based measurements

To relax the requirements on the measurement ADC, a single bit ADC (a comparator) can be used. A single bit ADC in combination with a feedback to the measured quantity, control logic and multiple single bit measurements can achieve the high resolution of a multi bit measurement ADC. These techniques are similar to the SAR (successive approximation) and algorithmic ADCs.

Figure 8.3-6 shows a way to measure I_{meas} against a reference current I_{ref} by means of a single voltage comparator. This technique is demonstrated in [71], [72]. The diode connected nmos M_{1b} terminates the reference current I_{ref}. The current mirror $M_{1b} \rightarrow M_{1a}$ copies I_{ref}. The drain of M_{1a} is a high impedance point and its potential is very sensitive to the difference between the copied I_{ref} and the measured I_{meas} current. I_{meas} is one of the MSB unary currents I_u. The switch SW_k connects the k-th unary MSB current $I_{u,k}$ to M_{1a}. After a measurement, I_{meas} is adjusted by a single step, with respect to the direction given by the 1 bit ADC measurement information. The corrected I_{meas} is again measured against I_{ref}. The procedure repeats until the decision of the comparator changes, which means that the drain voltage of M_{1a} is about equal to the

drain voltage of M_{1b}. This is the criterion that either the measured current I_{meas} is already adjusted (in the case of calibration) or its difference with I_{ref} is in a certain relation to both the number of measurement cycles and the correction step (as in the case of BIST – built-in-self-test).

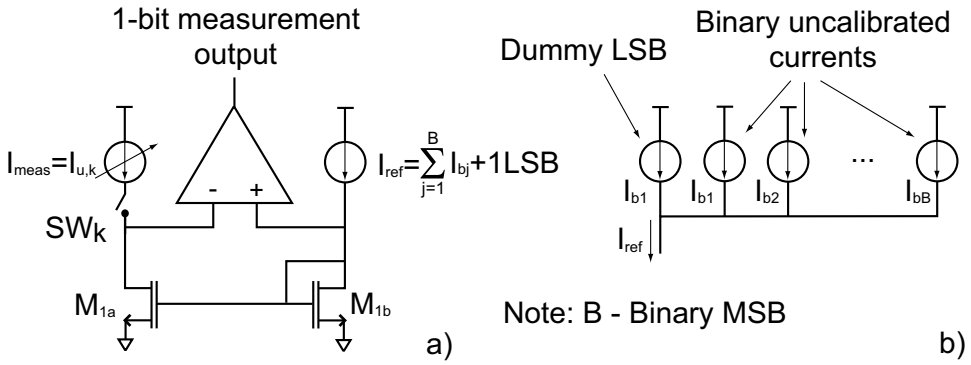


Figure 8.3-6. A 1 bit measurement ADC from [71]: a) measurement with a voltage comparator; b) construction of the reference as the sum of all uncalibrated binary currents plus 1 dummy LSB current.

This technique is sensitive to the unavoidable offset error of the measurement. All measured currents are calibrated with a common error that is due to the mismatch between:

1) the input transistors of the comparator;
2) M_{1a} and M_{1b};
3) the loads of the M_{1a} drain and M_{1b} drain;
4) I_{ref} and the non-calibrated binary currents.

8.3.3.2.3. Single bit current ADC-based measurements

To reduce the offset measurement error, the current measurement can be executed in the current domain by means of a current comparator, for example as shown in Figure 8.3-7, Figure 8.3-8 and Figure 8.3-9. Current-based techniques can be used to cancel the offset measurement error. For example, Figure 8.3-7 and Figure 8.3-8 show two different realizations of a way to calibrate I_{meas} to I_{ref} in two steps (ϕA and ϕB) via temporary current adjustment. The final result is a complete cancellation of the measurement error.

8.3.3.2.3.1. ANALOG COMPENSATION OF MEASUREMENT ERRORS

Figure 8.3-7 shows a measurement technique that is derived from elements of [40]. During the first step ϕA, the switch SW_k is connected to I_{ref}. I_{temp} is modified until the inverter INV1 changes state. This is the criterion that $I_{bias} = I_{diff} = I_{ref} - I_{temp} + I_{offset}$. I_{offset} includes the error offset current, due to the mismatch between the nmos current mirror transistors M_{1a} and M_{1b}, the pmos current mirror M_{2a} and M_{2b}, and the mismatch of the threshold voltage of the inverter INV1. During the second step ϕB, the switch SW_k disconnects I_{ref} and connects I_{meas}. The current of I_{meas} is either incremented or decremented until the output of INV1 changes its state. This is the criterion that

$I_{bias} = I_{diff} = I_{u,k} - I_{temp} + I_{offset}$ and hence $I_{u,k} \approx I_{ref}$ within the post-correction errors of the two correction steps. The calibration equations are shown in Equation 8.3-4.

Equation 8.3-4

$$\phi A : I_{temp} \approx I_{ref} - I_{bias} + I_{offset} + \tilde{I}_{qA}$$

$$\phi B : I_{u,k} \approx I_{temp} + I_{bias} - I_{offset} + \tilde{I}_{qB} \approx I_{ref} + 2\tilde{I}_{q}\text{'}$$

where $\tilde{I}_{qA} \in \left[-I_{caldacLSB}, +I_{caldacLSB}\right]$, $\tilde{I}_{qB} \in \left[-I_{caldacLSB}, +I_{caldacLSB}\right]$, $\tilde{I}_{q} \in \left[-I_{caldacLSB}, +I_{caldacLSB}\right]$ are post-correction errors, $I_{caldacLSB}$ is the correction LSB step, and the other currents are indicated in Figure 8.3-7. The two calibration steps double the range of the post-correction error $\tilde{I}_{q}$.

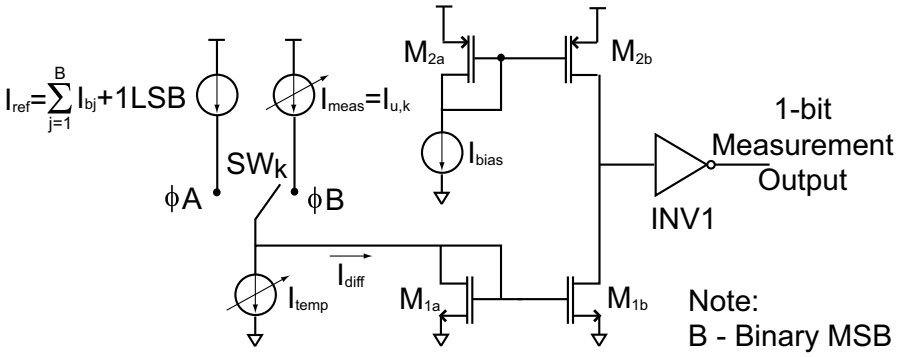


Figure 8.3-7. A 1 bit current comparator with offset error compensation, e.g. [40].

The original work presented in [35], [73], [74], [75] realizes the cancellation of the measurement offset in a simpler way than [40]. Figure 8.3-8 shows the measurement device. By controlling the sign of the calibration quantization error $\tilde{I}_{q}$ during ϕA and ϕB, the final quantization error is optimized.

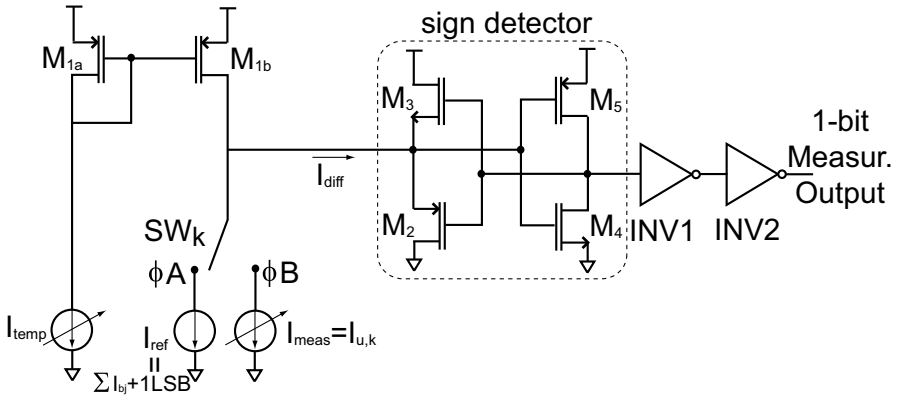


Figure 8.3-8. A 1 bit current comparator, with offset error compensation, from [35] and chapter 10.2. .

During ϕA, I_{temp} is adjusted to I_{ref}. I_{temp} is copied via the current mirror $M_{1a} \rightarrow M_{1b}$. The drain of M_{1b} is a low impedance node and the current difference $I_{diff} = I_{temp} - I_{ref}$

flows in or out of the current sign detector M_2, M_3, M_4, and M_5. I_{temp} is adjusted until the sign detector changes its state, but always with a negative correction error, i.e. $I_q < 0$. During ϕB, the switch SW_k disconnects I_{ref} and connects I_{meas}. I_{meas} is adjusted now until the output of the sign detector changes its state, but always with a positive correction error, i.e. $I_q > 0$. Both the measurement error and the mean of the correction error are cancelled. The calibration equations are shown in Equation 8.3-5.

Equation 8.3-5

$$\phi A : I_{temp} \approx I_{ref} + I_{offset} - \tilde{I}_{qA}$$
$$\phi B : I_{u,k} \approx I_{temp} - I_{offset} + \tilde{I}_{qB} \approx I_{ref} + \tilde{I}_q,$$

where $\tilde{I}_{qA} \in [0, I_{caldacLSB}]$ is the correction quantization error of ϕA, $\tilde{I}_{qB} \in [0, I_{caldacLSB}]$ is the calibration quantization error of ϕB. $\tilde{I}_q \in [-I_{caldacLSB}, +I_{caldacLSB}]$ is the final correction error. The spread of $I_{u,k}$ is two times smaller than the one in Equation 8.3-4.

8.3.3.2.3.2. DIGITAL COMPENSATION OF MEASUREMENT ERRORS

Figure 8.3-9 shows a measurement approach that uses the average of two measurements to cancel the measurement offset error. This measurement technique is reported in [70].

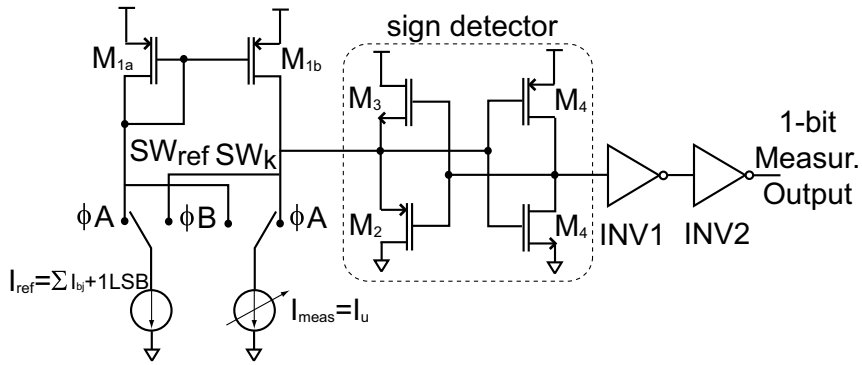


Figure 8.3-9. 1 bit current comparator, with offset error compensation via the averaging 2 exchange-input measurements, see [70].

The measurement uses two separate measurements in two steps ϕA and ϕB. During ϕA, the switch SW_{ref} connects I_{ref} to M_{1a} and the switch SW_k connects I_{meas} to M_{1b}. A comparison is done. I_{meas} is adjusted repeatedly until the output of the comparator changes. The result is stored. During ϕB, the currents I_{ref} and I_{meas} are swapped and I_{meas} is reset. The switch SW_{ref} connects I_{ref} to M_{1b} and the switch SW_k connects I_{meas} to M_{1a}. The measurement repeats and the final result is again stored. In the digital domain the average of both measurements is calculated and it is accepted as the final result. A disadvantage of this measurement technique is the linearity requirements of the analog adjustment: while the average of both measurements is correct in the digital domain, it can only be correct in the analog domain if the conversion between both domains is sufficiently linear.

8.3.3.2.3.3. MEASUREMENT-ERROR INSENSITIVE APPROACH

To solve the problems due to the measurement offset error without increasing the correction error, while simplifying further the measurement circuits, this thesis proposes a new calibration approach (in chapter 10.4.). It calibrates both the binary and unary DAC currents and hence significantly relaxes the design requirements of the measurement device. The measurement error is completely assumed and tolerated, because all currents, both binary and unary, are calibrated and hence the measurement error is transformed into a linear gain error of the whole DAC. The DAC gain can be simply ignored, for it is linear and does not generate harmonic distortion components. If the applications do not allow ignoring the gain error, it can easily be additionally calibrated, for it is a common error of all currents.

Figure 8.3-10 shows the simplified measurement approach. For more in-depth description see chapter 10.4.

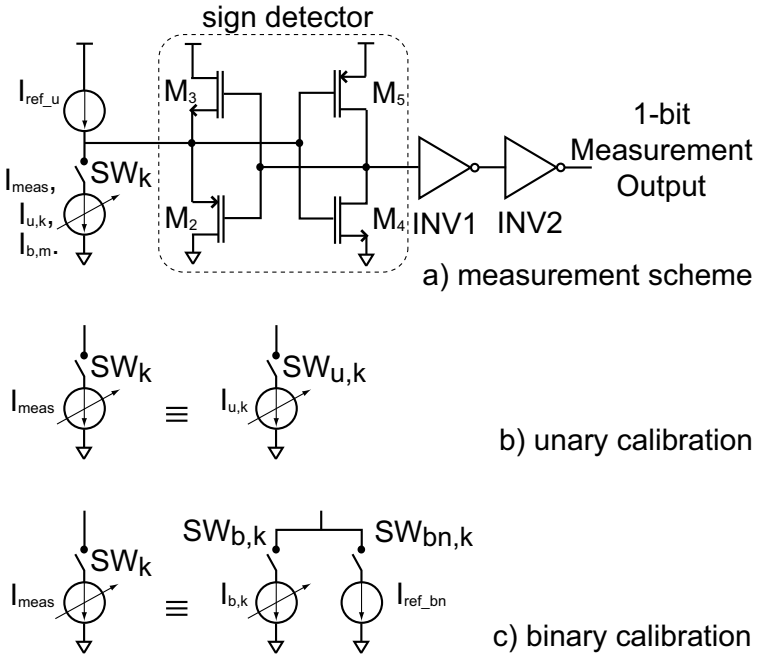


Figure 8.3-10. a) Straightforward current comparison by means of a current sign detector; b) calibration of the MSB unary currents; c) calibration of the LSB binary currents.

The reference contains two parts: a positive one and a negative one, i.e. $I_{ref} = I_{ref\_u} - I_{ref\_bn}$. For calibration of binary currents, I_{ref_bn} is used to adjust the nominal value of the reference. More details follow below. Nominally, I_{ref_u} equals the unary MSB current. When calibrating the unary MSB current, only I_{ref_u} is used, as shown in Figure 8.3-10a and Figure 8.3-10b. I_{ref_u} is a pmos current that can be either on-chip or off-chip generated. It is connected to the low impedance node of the input of the current sign detector (the sources of M_2 and M_3). Its nominal value is equal to the nominal value of the MSB DAC current. The MSB DAC current can be either a unary current or a binary current, respectively in the cases of either segmented/unary or binary DAC architectures. The switch SW_k represents either a single switch or a cluster of

switches. When a unary current is calibrated, SW_k is the single switch of the selected unary current, as in Figure 8.3-10b. To meet the measurement/correction criterion, the calibrated current is tuned until the 1 bit measurement output changes its value from either 1 to 0 or 0 to 1.

When a binary current is calibrated SW_k is a cluster of switches that connects the binary current for calibration and the negative nmos current reference I_{ref_bn}. I_{ref_bn} is generated internally by connecting some of the binary currents in parallel, so that nominally Equation 8.3-6 is satisfied.

Equation 8.3-6

$$\bar{I}_{b,k} = \bar{I}_{ref\_u} - \bar{I}_{ref\_bn},$$

where $\bar{I}_{b,k}$ is the expected value of the k-th binary current, $\bar{I}_{ref\_u}$ is the expected value of the positive pmos reference, and $\bar{I}_{ref\_bn}$ is the expected value of the negative nmos reference. To meet the measurement/correction criterion, the selected binary current is tuned, until the 1 bit measurement output changes its value from either 1 to 0 or 0 to 1. More details of the binary currents calibration are addressed in the algorithm description of the method further in the text.

8.3.3.3. Discussion

From the foregoing discussion, it is clear that the accuracy problems of the measurement device and the reference can be solved at three different levels: circuit, algorithmic, and conceptual level. For example, at circuit level, the current-based measurement devices guarantee smaller errors than the voltage-based measurement devices. At algorithmic level, current-based techniques can cancel the measurement error for the price of increased post-correction error. However, the best results can be achieved when the measurement errors are taken into account at conceptual level. For example, the measurement errors are transformed to a linear gain error when all DAC currents are corrected.

8.4. ALGORITHM

The algorithm block is responsible for implementing the sequence of operations that make possible the error measurement and its consequent correction. As shown in Figure 8.2-1, the calibration algorithm can be executed in either background or foreground. The foreground algorithms are usually executed at start-up and they are not active during the normal D/A conversion operations. The background algorithms form a loop that is being executed simultaneously with the normal D/A conversion operations. To avoid frequency spurs at the output of the DAC, the background algorithms can be executed at randomized clock steps, as in [69].

The two main types of DAC signal current sources are the unary and binary currents. The majority of the calibration methods in the literature correct only the unary currents. There is no practically realized method that corrects all binary current sources. This chapter proposes a new original method to calibrate both the binary and unary current sources. Due to the inbuilt error information processing, the algorithms are designed in the digital domain. Therefore in practice, the algorithm descriptions assume both an ADC error acquisition and quantized current correction. However, there is no fundamental reason why these algorithms should not be used with analog error measurements.

The text below describes new original concepts that are only partially overlapping with chapter 10. These concepts are included here to make the analysis of the self-calibration method complete.

8.4.1. Unary-currents calibration

The primary task of the unary currents calibration is to correct every unary current $I_{u,k}$ to the value of the reference current I_{ref}. If the method corrects only the unary currents, a secondary task needs to be to compensate all measurement errors. Note that if the measurement error is not compensated, it will appear in the INL and DNL DAC characteristics at the transitions in which the unary currents are switched. Furthermore, the goals of the calibration algorithms may include high post-calibration accuracy, implementation efficiency, and scalability. The post-calibration accuracy can be probabilistically specified in terms of the correction error. The implementation efficiency requires that the algorithm is realized as a simple circuit and occupies reasonable on-chip silicon area. The algorithm needs to be scalable, i.e. the algorithm should be fully parameterized and it should be able to correct different numbers of currents without major modifications.

Figure 8.4-1 shows the self-calibration algorithm of [35], [73], [74], [75], designed for the measurement setup of Figure 8.3-8. This algorithm is superior to the algorithms designed for the measurement setups of Figure 8.3-6, Figure 8.3-7 and Figure 8.3-9. The algorithm corrects for the measurement offset error and minimizes the correction error. The algorithm is realized as a simple 8-state Finite-State-Machine (FSM) and it is fully parameterizable in terms of the number of unary currents that need to be calibrated to the common reference I_{ref}.

In Figure 8.4-1, S0 is the initial reset state. The two main registers X1 and X2 are reset to 0. X1 is used to store the digital calibration word for I_{temp}. X2 is used to store the digital calibration word for the unary current that is being calibrated. When the calibration of the unary current is ready, X2 is copied to the local register of the current and X2 is reset to 0. At S0, the algorithm expects the instruction to start the calibration routine. The first calibration step ϕA includes the states S1, S2, and S3. ϕA adjusts I_{temp} to I_{ref}. At S1, I_{ref} is connected to the common node of M_{1b}, M_2, and M_3 of Figure 8.3-8. Consequently, I_{ref} and the adjustable I_{temp} are compared. The inverted comparison result is stored in the polarity register P. At S3, X1 is incremented, which corresponds to either increase or decrease of the current I_{temp} by one correction step, depending on the value of the polarity bit P. That is to say that the difference between the I_{temp} and I_{ref} is reduced by one correction step. The algorithm repeats S3 until the difference between I_{temp} and I_{ref} changes its sign. Before exiting S3, the X3 is decremented if P=0, so that I_{temp} is always less than I_{ref} within one correction step I_q, according to Equation 8.3-3.

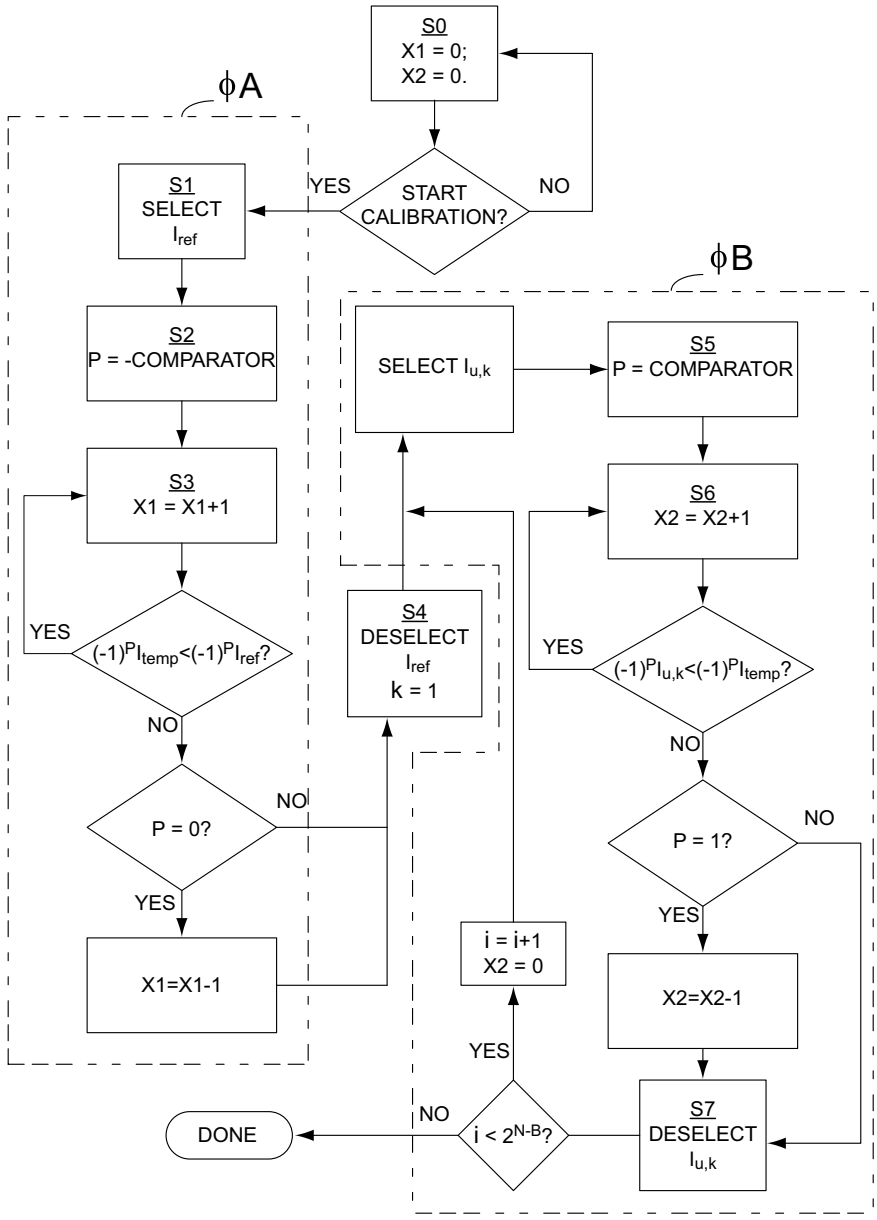


Figure 8.4-1. FSM chart of the unary currents self-calibration algorithm, based on [35] and [76].

S4 is an intermediate state between the two main calibration steps ϕA and ϕB. S4 disconnects I_{ref} from the sign detector input, sets the counter k=1 and connects the first unary current $I_{u,1}$ to the input of the sign detector. The second calibration phase ϕB includes the states S5, S6, and S7. ϕB corrects all unary currents to I_{temp} always within a positive correction error, as in Equation 8.3-3. At S5, the comparison result is stored in the polarity bit P. At S6, the register of the correction for the calibrated unary current is incremented, so that the difference between $I_{u,k}$ and I_{temp} is minimized. When the sign

detector changes its sign, the value of the unary current correction is adjusted, so that the correction error is always positive. At S7, $I_{u,k}$ is disconnected, the algorithm checks if there are more currents to be calibrated. If more unary currents need to be corrected ϕB is repeated until all unary currents are calibrated.

The final result of the unary currents calibration is that the post-correction accuracy of every unary current is within one correction step $\overline{I}_q$. Therefore, the accuracy of the unary currents does not depend on the CMOS process parameters A_β and A_{VT} of Equation 2.3-2, but rather on the design parameter $\overline{I}_q$. However, the post-calibration accuracy of the whole DAC still depends on A_β and A_{VT}, because the binary currents are not calibrated. Therefore, the architecture of the DAC in terms of segmentation is correlated with the calibration, since only the MSB unary currents are calibrated. To achieve both full design independence of A_β and A_{VT} and independence of any relationship between DAC architecture and calibration, the binary currents must be calibrated, too.

8.4.2. New binary-currents calibration in a unary way

To reduce risks in the DAC product due to the non-calibrated binary part, the work of [77] (related to this thesis) suggests a unary approach to the design of the most-significant binary bits. This solution is a particular case of a more general method that separates the unit of calibration from the DAC architecture. This separation is conceptualized in chapter 9 as calibration segmentation. A calibrated unit element (CUE) is introduced as an additional abstraction level to the DAC current cells. For the needs of the calibration, the CUE can be chosen in the binary part, at the unary current level, or in the unary MSB part of the segmented DAC, respectively as shown in the examples of Figure 8.4-2a, b, and c.

Thus, the calibration is still unary, but it can be applied in practice to the most-significant binary currents. A modification of this approach is demonstrated in [69]. The DAC architecture contains a binary LSB part, a unary Middle-SB part, and a unary MSB part. The CUE is chosen to be nominally equal to the Middle-SB unary currents. All binary bit currents plus one dummy LSB current are packed in a group and calibrated together, similarly to the unary MSB part in Figure 8.4-2c. The Middle-SB unary currents are calibrated directly to the reference. The MSB unary currents are calibrated through their ingredient CUE parts, as in Figure 8.4-2a.

Although this approach is able to calibrate some of the DAC binary currents, it does not guarantee full technology and design independence. In practice, not all binary currents can be calibrated. In addition, the occupied silicon area of the calibration resources exponentially increases for each additional calibrated binary bit. Therefore, this calibration approach has practical limitations and hence the DAC accuracy still depends, on the CMOS process parameters A_β and A_{VT}, although to a lesser extent.

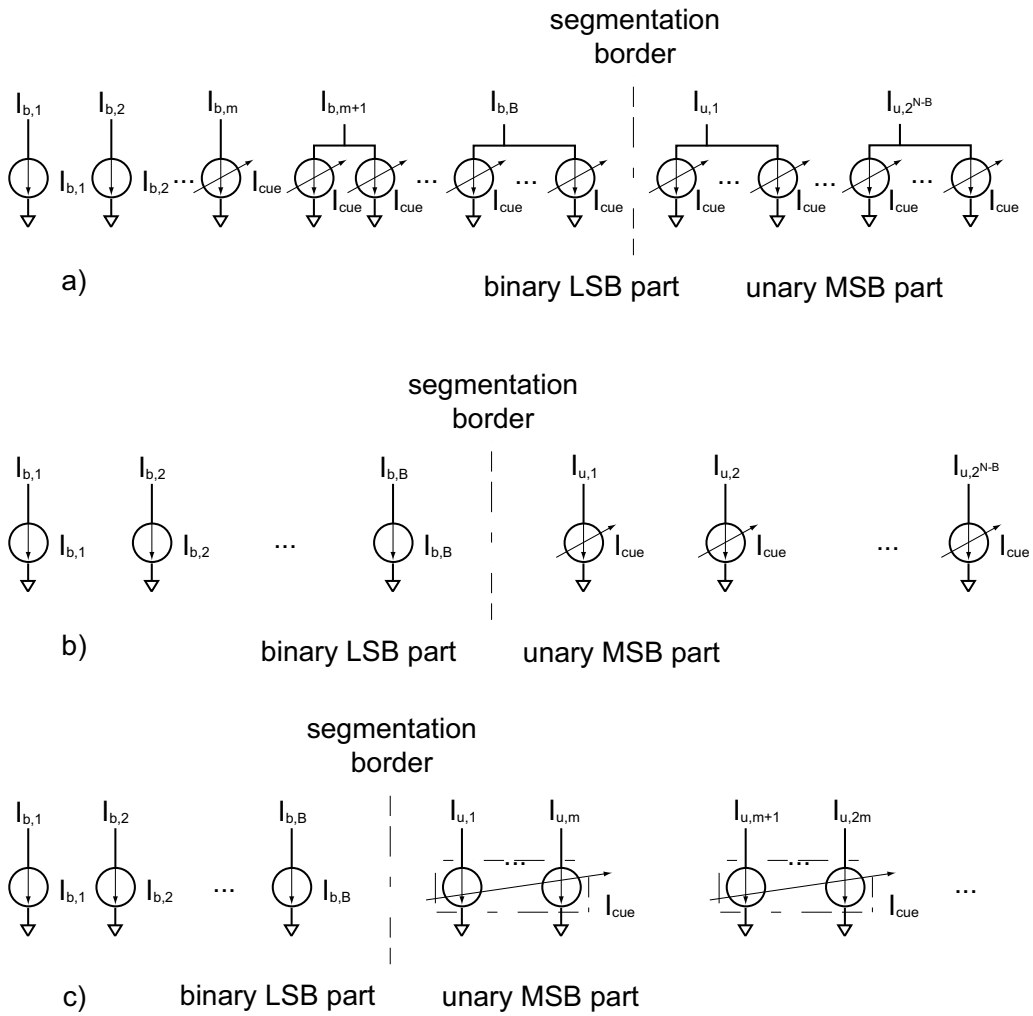


Figure 8.4-2. Three different options to choose the bit of the CUE in a segmented DAC, based on [77]: a) CUE is chosen in the LSB binary part; b) CUE is chosen at the unary level; c) CUE is chosen in the MSB unary part.

8.4.3. New true binary-currents calibration

True binary currents calibration is needed to make the DAC accuracy depend only on design parameters. As the calibration guarantees accuracy, the DAC architecture can be tailored to optimally achieve different targets, e.g. segmentation, occupied area, etc. In the literature, a practically realized self-calibration algorithm for the 4 MSB binary currents of a 14 bit DAC is reported in [70]. The reported method is based on the original theoretical work of [78], where two algorithms are proposed. To execute the calibration algorithms, multiple independent references to generate all binary currents are needed. These proposals consume too many resources in practice to justify binary currents calibration.

To address the need of an efficient calibration of all binary current sources, this chapter proposes a methodology for calibration of both unary and binary currents to a single common reference. The only special requirement for the algorithm is the need for two sets of binary currents. That is to say each binary current needs to have a nominally

identical replica. The possible circuit solutions include doubling the binary bit currents, splitting the binary bit currents, using multi-core DAC systems [30], etc. Chapter 9 analyzes and discusses in details the segmentation possibilities and particularly the hardware and calibration segmentations that can be used to create nominally identical binary current replicas. For illustrative purposes, the example of splitting the binary currents to create two sets of binary sub-currents is shown in Figure 8.4-3. This architecture of the binary current sources is a special case of the CUE concept, discussed in the previous section. Note that the DAC current cells remain unchanged as a current cell entity and only one pair of data switches is used. A broader description of the methodology can be also found in the patent [79].

The calibration scheme is shown in Figure 8.3-10. The pmos reference I_{ref_u} is connected to the current comparator, i.e. 1b ADC, current sign detector. It is nominally equal to the MSB binary current I_{B+1}. That is to say I_{ref_u} is nominally twice as large as each of the MSB binary sub-currents I(B)(1) and I(B)(2). I_{ref_u} can be either externally or internally generated. The other part of the reference current is the nmos I_{ref_bn}. The currents of the first binary set construct I_{ref_bn} in a configuration that depends on the particular binary current that is being calibrated.

There are two ways to select a binary current. The binary current can be either selected only for connection or selected for both connection and correction. When the binary current is selected only for connection, it is actually used as a part to construct the variable reference I_{ref_bn}. When the binary current is selected for both connection and correction, it is a current source that needs to be adjusted.

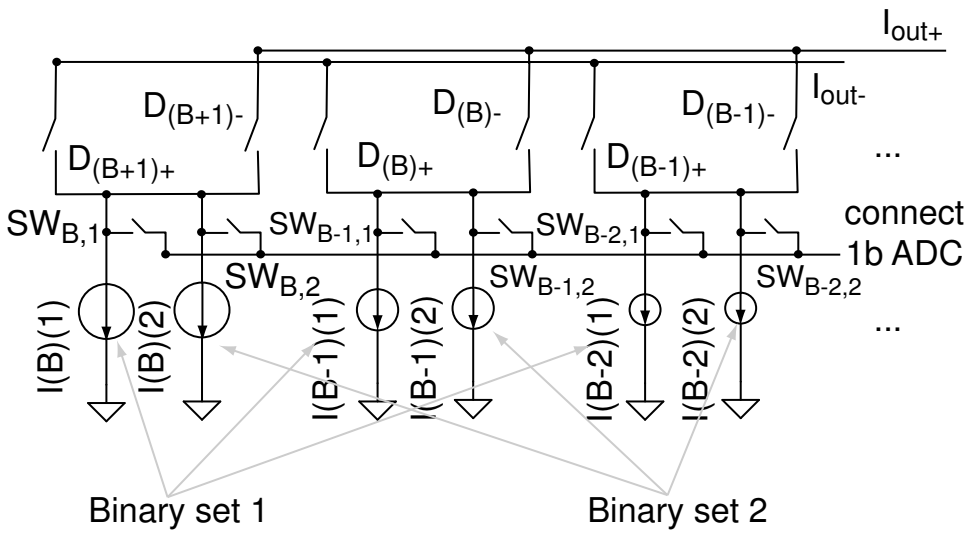


Figure 8.4-3. Splitting of the DAC binary currents to create two sets of binary sub-currents, i.e. implementing hardware segmentation, based on [80].

The calibration algorithm can be described with the calibration Equation 8.4-1 and Equation 8.4-2. Three calibration loop steps are needed to correct the MSB binary sub-currents $I(B)(1)$ and $I(B)(2)$. In the calibration equations, the sink currents are indicated on the left side of the equations, while the source current I_{ref_u} is indicated on the right.

Each equation includes an adjustable gray term and constant black terms. Steps 1 through 3 adjust the B bit binary sub-currents of binary sets 1 and 2 to $\frac{I_{ref\_u}}{2}$. After step 2, $I(B)(1)$ and $I(B)(2)$ are equal. Step 3 adjusts $I(B)(1)$ and $I(B)(2)$ at the same time, so that they are both made equal to $\frac{I_{ref\_u}}{2}$.

Equation 8.4-1

$$1: I_{bin}(B)(1) + \underbrace{\sum_{i=1}^{B-1} I_{bin}(i)(1) + 1LSB}_{I_{ref\_bn}} \overleftarrow{:=} I_{ref\_u}$$

$$2: I_{bin}(B)(2) + \underbrace{\sum_{i=1}^{B-1} I_{bin}(i)(1) + 1LSB}_{I_{ref\_bn}} \overleftarrow{:=} I_{ref\_u},$$

$$3: I_{bin}(B)(1) + I_{bin}(B)(2) \overleftarrow{:=} I_{ref\_u}$$

In like manner, the calibration flow continues with the $B-1$ currents $I(B-1)(1)$ and $I(B-1)(2)$. Their calibration is shown in Equation 8.4-2. Note that the left side of step 6 equation includes the already adjusted $I(B)(1) = \frac{I_{ref\_n}}{2}$. The sum $I(B-1)(1) + I(B-1)(2)$ is calibrated to $\frac{I_{ref\_n}}{2}$, and hence $I(B-1)(1) = I(B-1)(2) = \frac{I_{ref\_n}}{4}$. The rest of the binary currents are similarly calibrated.

Equation 8.4-2

$$4: I_{bin}(B-1)(1) + \underbrace{\sum_{i=1}^{B-2} I_{bin}(i)(1) + \sum_{i=B}^{B} I_{bin}(i)(1) + 1LSB}_{I_{ref\_bn}} \overleftarrow{:=} I_{ref\_u}$$

$$5: I_{bin}(B-1)(2) + \underbrace{\sum_{i=1}^{B-2} I_{bin}(i)(1) + \sum_{i=B}^{B} I_{bin}(i)(1) + 1LSB}_{I_{ref\_bn}} \overleftarrow{:=} I_{ref\_u},$$

$$6: I_{bin}(B-1)(1) + I_{bin}(B-1)(2) + \underbrace{I_{bin}(B)(1)}_{I_{ref\_bn}} \overleftarrow{:=} I_{ref\_u}$$

Equation 8.4-1 and Equation 8.4-2 indicate that the binary current calibration algorithm requires a simple 3-state FSM (plus one reset state). The FSM is shown in Figure 8.4-4. For simplicity, several notation differences are introduced compared to the unary case of Figure 8.4-1. In this FSM, k indicates the bit current that is being calibrated, m indicates the addressed binary set, X is the digital calibration word. Here, X is assumed inherently bipolar, i.e. the polarity bit P from Figure 8.4-1 is included in X. Therefore, both increment and decrement operations are possible for X. The index all means that an operation is applied to all similar objects. The index $all < k$ means that an operation is applied to all objects having index from 1 to k-1.

In the FSM chart of Figure 8.4-4, S0 is the initial reset state. All registers are set to 0. The bit pointer k is assigned to the MSB binary bit B. The set pointer m is assigned to the first binary set. When the calibration is initiated, the execution moves on the main loop states S1, S2, and S3. These states operate in two modes. The first mode $m \le 2$ implements the first two lines of the calibration equations Equation 8.4-1 and Equation 8.4-2. The second mode $m > 2$ implements the third line of Equation 8.4-1 and Equation 8.4-2.

In state S1, first all binary currents of the first set are selected. If $m \le 2$, the k -th binary current is deselected and replaced with the k -th binary current of the m -th set. Of course, if $m = 1$, nothing needs to be done. In state S2, the result of the sign detector is recorded as *Sign*, see Figure 8.3-10 for the schematics. In state S3, the register of the k -th binary current (being calibrated) of the m -th set is adjusted until the comparator output differs from its initial result, recorded as *Sign*. This check-point realizes both the first line of equations Equation 8.4-1 and Equation 8.4-2, when $m = 1$, and then the second line of Equation 8.4-1 and Equation 8.4-2, when $m = 2$. Further, m is either incremented or set to 1. If $m = 3$, the loop implements the third line of Equation 8.4-1 and Equation 8.4-2. S1 selects the k -th binary currents of both sets and the binary currents of the first set so that all selected currents are nominally equal to the reference, see Figure 8.3-10. Then, S3 simultaneously adjusts the correction currents of the k -th binary currents of both sets.

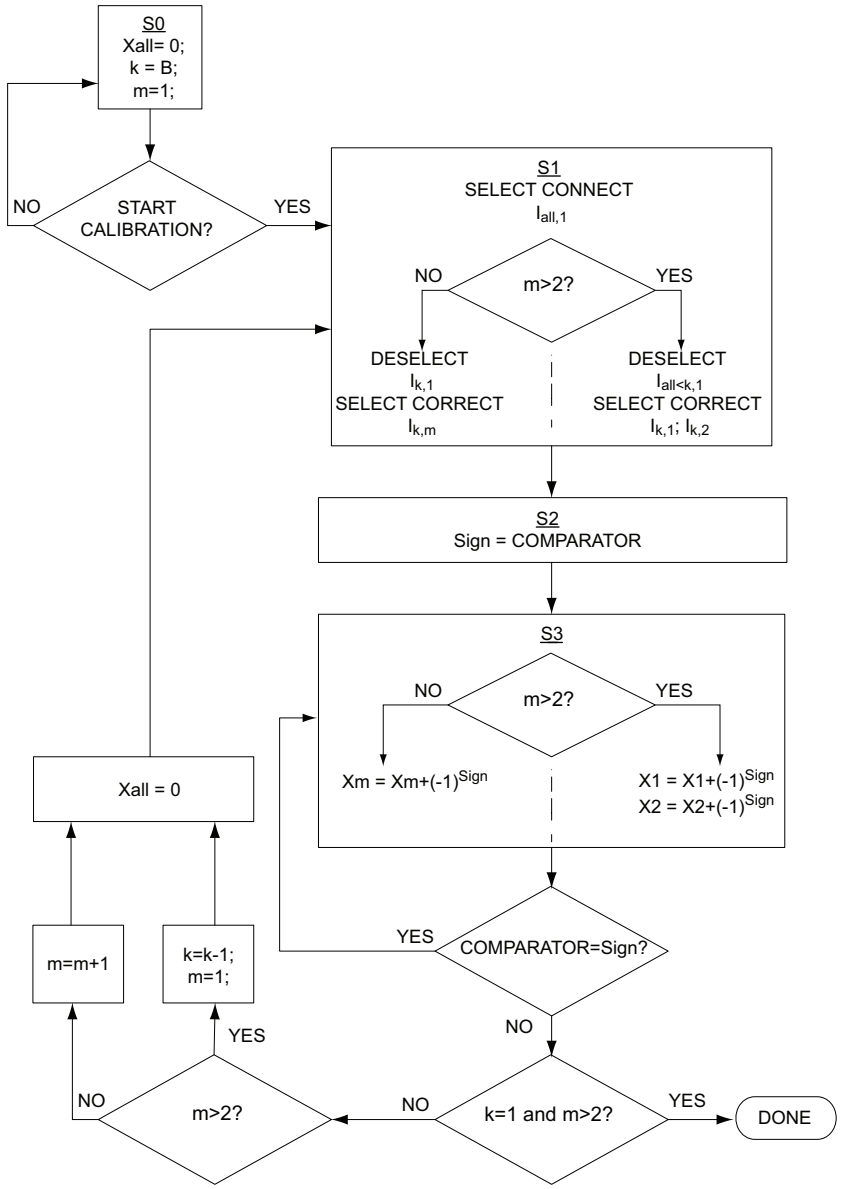


Figure 8.4-4. FSM chart of the binary currents self-calibration algorithm, based on [80] and [47].

This is the first algorithm implementation that is capable to correct non-unary currents, i.e. scaled binary currents. The implementation is simple and it can fit in a FSM with only 4-states.

8.5. SELF-CORRECTION

The calibration self-correction element concerns the means for compensating the measured errors. It has three sub-elements: the method, the correction circuit, and the correction memory.

8.5.1. Self-correction method

The self-correction method realizes the correction of the measured error. To classify the self-correction methods, this chapter uses three categories. The first category specifies at what level the correction is applied: either at high (DAC system) or low (DAC elements). The second category specifies how the correction is applied: either the correction quantity is injected or the main quantity is regulated. The third category specifies what the amplitude nature of the correction quantity is: either discrete or continuous. Figure 8.5-1 shows a classification diagram of the DAC self-correction methods with respect to the first two categories, i.e. at what level and how the correction is applied. The diagram shows in grey the correction means.

8.5.1.1. High level correction

If the correction is applied at high level, i.e. the DAC output is directly compensated: a correction DAC can be used in parallel with the main DAC. This correction method is demonstrated in [36]. The correction DAC generates the correction current in accordance with the input code, so that the accuracy of the combined output is improved. This correction solution is simple and very efficient. Since the DAC is corrected at its output, many types of errors are simultaneously corrected. These include current mismatch, finite output resistance, load resistors mismatch, etc. The overall DAC accuracy depends only on a design parameter: the LSB step of the correction DAC. However, the dynamic responses of the main DAC and the correction DAC are different, because both DACs are intrinsically different. Therefore, the combined output deteriorates at higher speeds.

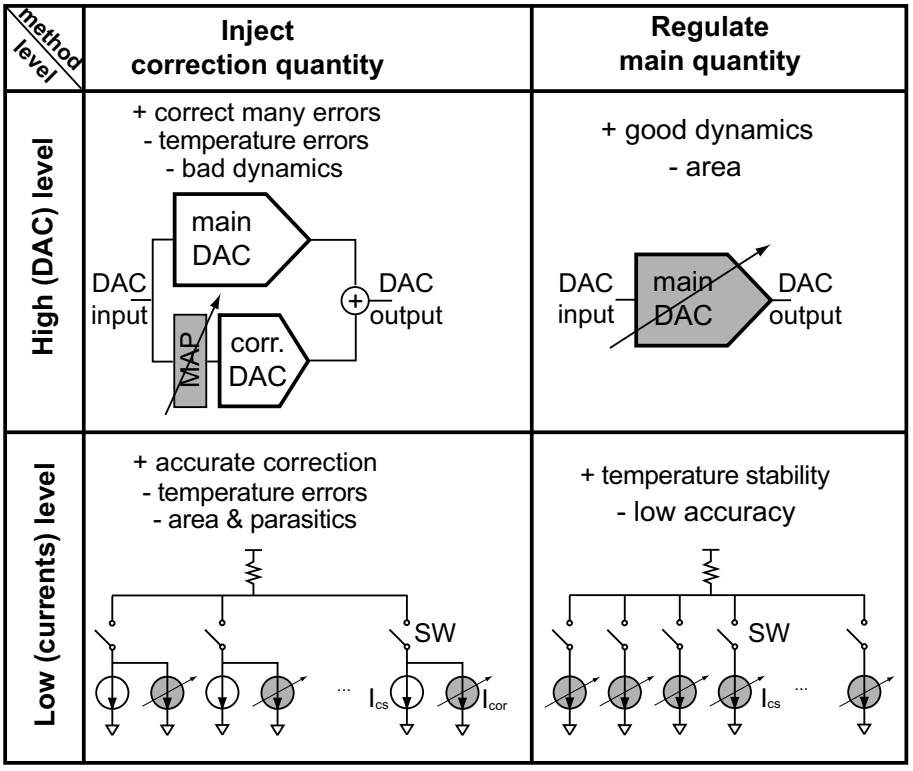


Figure 8.5-1. Classification of the DAC self-calibration methods with respect to either injecting correction quantity or regulating the main quantity.

8.5.1.2. Low level correction, inject or regulate

Alternatively, the main DAC can be internally regulated, see Figure 8.5-1, so that the errors in its output are removed. The correction needs to be applied then at low level, i.e. the errors of the DAC currents need to be compensated. The DAC currents can be either regulated via adjusting their values or compensated via injecting correction current. Injecting correction current requires more area than regulating the main current. It also adds parasitic wiring capacitance at the connection point between the main current and the correction current. In foreground calibration methods, injecting correction current has the comparative disadvantage that the main current and the correction current have different temperature coefficients and hence the error compensation deteriorates when the temperature changes after the calibration. However, injecting correction current can achieve more accurate error compensation than regulating the main current, because the nominal value of the correction current is much smaller than the nominal value of the main current.

8.5.1.3. Low level correction, continuous or discrete

Low level correction can be implemented in either a continuous or a discrete way, which is the third category of the proposed classification see Figure 8.5-2. An example of a continuous correction can be found in [69]. An example of a discrete correction is described further in the chapter.

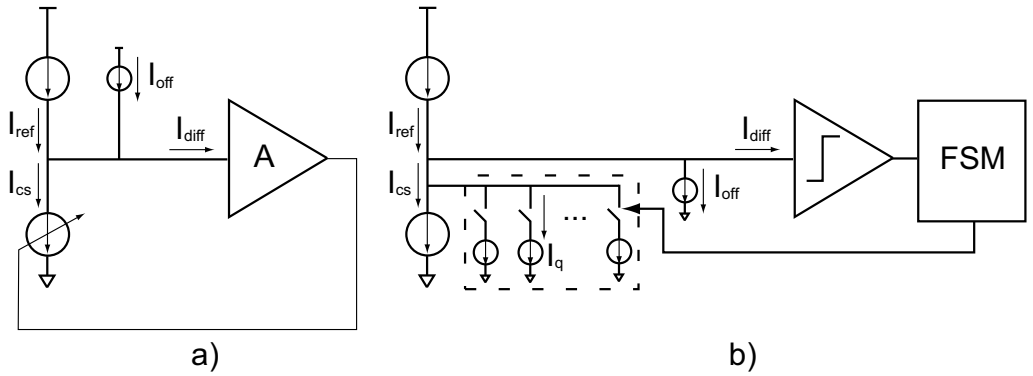


Figure 8.5-2. a) Continuous current correction; b) discrete current correction.

When a continuous correction is applied to the current, the corrected current, I_{cs}, becomes equal to the reference only within a certain error that includes the limited loop gain error, input offset error of the gain device, and the intrinsic and extrinsic noise, i.e.:

Equation 8.5-1

$$I_{cs} \approx \frac{A\left(I_{ref} + I_{off} + I_{n\_in}\right)}{1+A} + I_{n\_ex},$$

where I_{cs} is the current for calibration, A is the loop gain, I_{ref} is the reference, I_{off} is the offset error of the measurement device (can be either positive or negative), I_{n_in} (not shown in Figure 8.5-2) is the amplifier input referred noise (e.g. noise of the amplifier), I_{n_ex} (not shown in Figure 8.5-2) is the amplifier output referred noise (e.g. disturbances due to the DAC normal operation).

When a discrete correction is applied to the current, the corrected current, I_{cs}, becomes equal to the reference only within the quantization error of the correcting step, i.e. the LSB step of the correction. Thus, the correcting LSB step determines the post-correction accuracy of the calibrated current.

Equation 8.5-2

$$I_{cs} \approx I_{ref} + I_q,$$

where I_{ref} is the reference, and I_q is the quantization error (can be either positive or negative) due to the discrete nature of the correction quantity. Note that digital techniques can be used to cancel I_{off}, as shown earlier in this text. Equation 8.5-1 and Equation 8.5-2 indicate that the discrete correction approach offers a greater level of control than the continuous one. It features more relaxed requirements and is more robust against disturbances.

8.5.2. Correction circuits

The discussed category criterion "*Inject correction or regulate main quantity*" can be applied to the correction circuits, too. The current correction circuits can be divided in two major groups: gate-source voltage (V_{gs}) regulating circuits and correction current injection circuits. Note that this is a division on how the correction current is generated only and not on the actual method. For example, the correction current can be generated by regulating its V_{gs} and then injected in parallel to the main current source.

8.5.2.1. Gate-source voltage regulating circuits

This group of circuits regulates the bias voltages of the current source transistors. They feature increased complexity with respect to those that inject the correction current. The calibrated current sources need to have their gate nets controlled by the calibration algorithm block. The regulation is highly non-linear with respect to the generated correction current I_{cor}. These circuits directly connect to the DAC signal generation path, increasing the risks for DAC performance deterioration.

The work of [70] suggests a circuit that can regulate a current by means of regulating a voltage drop on its bias net. The circuit can easily be modified to generate either a continuous or a discrete correction current, as shown in Figure 8.5-3. Further, the circuit can be used either to inject a correction current or to regulate the main current, respectively should M_{1b} generate either the main current or only the correction current.

As the error correction is implemented within the current source, the standard DAC performance is not compromised. There are no extra correction circuits that interfere with the normal DAC signal current generation. In other words, the correction remains "behind the curtains". However, the complexity of the correction circuitry is high. Moreover, this added complexity is in a very sensitive area – the bias net of the DAC signal current sources. Therefore, the whole solution is very sensitive to external disturbances, particularly at higher DAC speeds. In addition, the correction is highly non-linear, because it uses the transformation $V_{gs} \rightarrow I_{cs}$. Thus, the advantages of the advanced error processing techniques, e.g. binary currents calibration from Figure 8.4-4, can be seriously compromised.

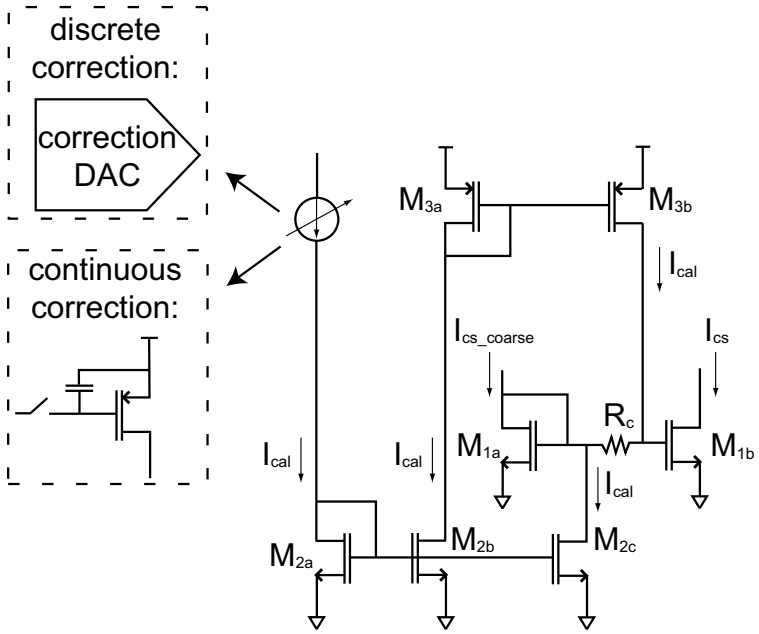


Figure 8.5-3. Regulation of V_{gs} by means of voltage drop on the bias net. The circuit can be modified to comply with continuous or discrete current, and to injecting correction or regulating the main current.

8.5.2.2. Correction current injecting circuits

To achieve more accurate correction, a separate device can generate the correction current. The correction current I_{cor} is added to the main current by injecting I_{cor} at the drain of the main course current source transistor. As the main coarse current I_{cs_main} generation circuit and the correction current I_{cor} generation circuit operate at different conditions (mainly different V_{gs} and transistor W/L), I_{cor} can be easily adjusted with very high resolution relatively to I_{cs_main}. The circuit can be easily modified to generate either quantized or continuous correction current, see Figure 8.5-4. A correction DAC (CALDAC) can generate the discrete current correction. Its advantages over the continuous current correction include linear correction and error information in the digital domain.

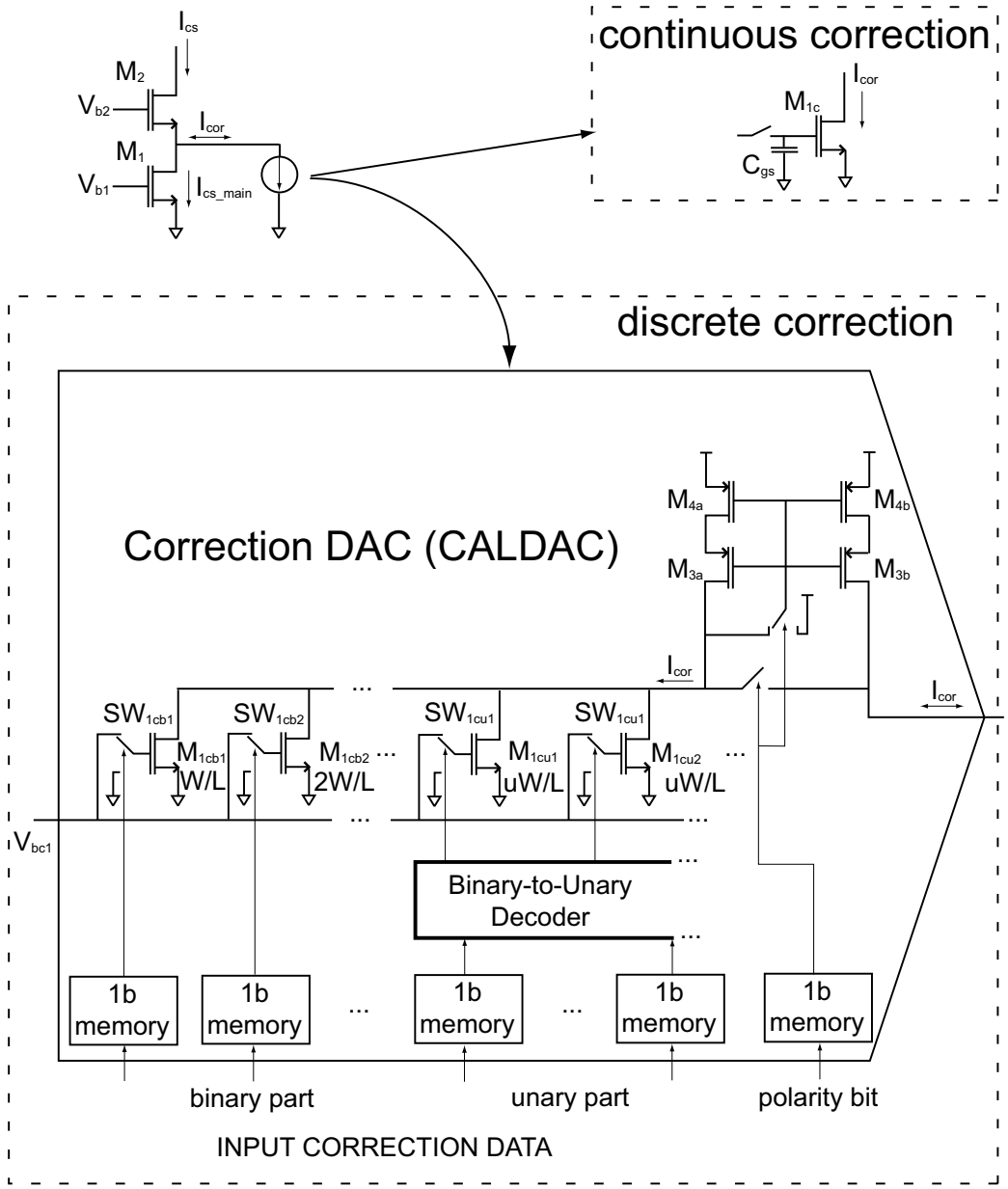


Figure 8.5-4. Injection of correction current with continuous and quantized current correction.

8.5.3. Correction memory

The correction memory stores the error information. This is an important block for the whole integrated self-calibration apparatus, because it occupies considerable silicon area. Respective to the type of the correction current, either continuous or discrete, the memory can be either analog or digital.

The analog memory is normally gate-source capacitance. Due to its leakage current, this type of memory needs to be regularly updated. The bigger the capacitor, the more leakage current can be tolerated but the more area it occupies. The digital memory keeps

the error information in the digital domain. It is normally a CMOS latch. To save area, dynamic digital memory can also be used. A capacitor that is regularly updated can keep digital information.

8.6. CONCLUSIONS

An analysis of the DAC current mismatch self-calibration methods has been presented. The analysis is based on three prime elements: self-measurement, algorithm, and self-correction. Each of these elements is discussed in detail and examples from the open literature are provided. Possible design alternatives are compared and a framework for evaluating their advantages and disadvantages is proposed. Missing techniques and alternatives are identified.

The efficient calibration of binary currents is argued as an important missing technique, which can bring a number of advantages, including DAC independence from technology and design errors, relaxation of self-measurement requirements, and simplifications of the measurement transistor circuits. Conceptual solutions for binary currents calibration are proposed.

The presented analysis will help engineers to evaluate current self-calibration methods and their building blocks. The analysis of the self-measurement element can be used in other smart DAC correction techniques, e.g. mapping. The presented analysis is used further in chapter 10 to develop new current self-calibration methods and in chapter 16 to design self-calibrating DAC test-chips.

PART IV: NEW CONCEPTS AND METHODS

This part includes chapters that propose new concepts and methods. Chapter 9 proposes new concepts for DAC segmentation. Chapter 10 proposes new methods for self-calibration of currents. Chapter 11 proposes a new concept for binary-to-thermometer decoder with built-in redundancy. Chapter 12 proposes a new high-level mapping method for mutual compensation of DAC errors. Chapter 13 proposes a new method for suppression of HD without error measurement. Chapter 14 proposes a new concept for DAC flexibility.

9. Chapter

NEW REDUNDANT SEGMENTATION CONCEPT

This chapter extends the segmentation concepts for dividing the DAC analog resources. Four main levels are discussed: functional, algorithmic, hardware, and correction. In the literature, various examples of algorithmic and hardware segmentation can be found. This chapter introduces the functional and correction levels of segmentation which are new for the field and create a lot of new possibilities for innovation.

The chapter concentrates on the functional segmentation and elaborates the new concept of functional redundancy, which suggests dividing the DAC analog resources into multiple D/A functions, called sub-DACs.

A new redundant segmentation is proposed which uses the unary approach for functional segmentation and mixed binary LSB/unary MSB approach for algorithmic segmentation. That is to say the new segmentation approach uses multiple nominally identical D/A functions and each of these D/A functions can be further segmented at algorithmic level to either binary, unary or mixed implementation.

9.1. INTRODUCTION

To generate the DAC analog output, the DAC digital data signals need to switch analog segments (e.g. currents) of the DAC analog resources on or off. Thus, segmentation refers to the way the DAC analog resources are divided in blocks to be controlled by the digital data. The two main segmentation approaches are binary and unary. The binary segmentation divides the DAC analog resources for an N bit resolution into N scaled blocks (segments). The smallest segment is the LSB segment and the size of the rest is scaled up with a factor of 2^j with $j \in [1; N-1]$ (defined here as binary scaling). The unary segmentation divides the DAC analog resources for an N bit resolution in 2^N-1 nominally identical blocks (segments), each having the size of the LSB block. The binary and unary are the most important segmentations, because they represent the corners of an important design trade-off "*resources versus performance*" (see discussions e.g. in [12], [81]). The binary segmentation is the most resources-efficient, while the unary segmentation delivers the best performance. For example, area and power consumption can be considered as DAC hardware resources, while DAC linearity and glitch energy can be considered as DAC performance parameters. Solving this trade-off determines the optimal DAC segmentation.

Beyond the trade-off "*resources versus performance*", there is the concept of redundancy. As shown in chapter 7, DAC correction methods can transform redundancy into performance improvement (see Table 7.4-1). Considering the binary segmentation, it offers no redundancy, since it is the most efficient segmentation. All segmentation approaches that divide the analog resources in finer segments than the binary segments feature redundancy. For example, the unary segmentation is an extreme case, featuring a significant amount of redundancy. The concept of generating redundancy to improve performance can be used by other, not that popular, segmentation approaches, e.g. sub-binary [11], or mixed (each segment is defined with different scaling factor).

To balance between *Resources*, *Performance*, and *Redundancy,* segmentation can be applied at different levels of abstraction. In the literature, the hardware and algorithmic levels are the most popular.

At hardware level, the transistor implementation is divided in segments. The hardware level is the lowest level that is considered in the following analysis. It concerns the transistor/layout level designs. For example, the implementation of a transistor can be split in four transistors, each having one quarter of the targeted overall W/L size. This represents unary segmentation at transistor level. Further, each of these unit transistors has a specific layout implementation that can be further split into units. Such techniques can improve transistor matching, compensate for various on-chip gradients, etc.

At algorithmic level, the D/A function is also divided in segments. For example, the LSB bits are implemented with binary segmentation and the MSB bits are implemented with unary segmentation. In this way, the advantages of both the binary and unary segmentations are used.

There is a hierarchical relationship between the algorithmic and hardware segmentation. The hardware segmentation should fully implement the algorithmic segmentation. However, the hardware segmentation can introduce further sub-segmentation (i.e. finer) within a segment defined at algorithmic level. For example, the work of [81] uses binary algorithmic segmentation but at hardware level the MSB binary segments are implemented in a unary way. That is to say unary units are implemented at schematic and layout but these are connected in parallel to realize binary currents.

In the literature, there is no sufficient discussion and understanding about different levels of segmentation. Usually, only the algorithmic level is discussed. That is why segmentation is not always optimal for the application goals. For example, DAC correction approaches are adapted to the chosen algorithmic calibration. Such an approach couples the DAC correction methods with the DAC architecture, i.e. the segmentation approach. For example, the DAC correction method calibrating the DAC unary currents ([35], chapter 16.2.) would deliver better overall DAC INL, when more DAC MSB bits are implemented in a unary way. Moreover, this correction method is by definition not possible with binary DACs. Similar examples include the mapping methods (chapter 4.3.), the DEM methods (chapter 3.5.), etc. Such DAC correction methods use the intrinsic DAC redundancy of the unary analog segments which results from the algorithmic segmentation. However, that creates dependence between the DAC algorithmic segmentation and the DAC correction methods. This dependence features a major contradiction that fundamentally limits the effectiveness of the correction methods. It can be briefly summarized as follows.

A goal of the DAC correction methods is to reduce the area of the DAC analog resources. For example, the DAC calibration can improve the accuracy of the unary current sources. Hence, the unary current sources can be designed with smaller transistors, since mismatch is tolerated. The more calibrated the DAC unary currents, the more accurate the overall DAC. However, using more unary currents requires a larger unary algorithmic segmentation part, which requires more area for the digital data control block and for the analog resources support block, as shown in Figure 2.4-3.

To avoid the dependence between the DAC segmentation and the DAC correction methods and to introduce further the concepts of DAC flexibility, a new DAC segmentation approach is proposed in this chapter. Two new levels of segmentation are introduced: functional segmentation and correction segmentation.

At functional level, the D/A function is divided in sub-functions (i.e. sub-DACs). Therefore, the functional segmentation can create functional level redundancy, which can be used for performance improvement.

Alternatively to segmenting the DAC analog resources in order to realize a D/A function in an optimal way, the DAC analog resources can be segmented in order to calibrate the analog resources in an optimal way. Therefore, the whole hierarchical chain functional-algorithmic-hardware segmentation (as shown in Figure 9.2-1) can be realized as a correction level segmentation. For the sake of simplicity, the whole hierarchical chain of segmentation for correction is simply named and used here as “correction level segmentation”. That is to say that the correction level segmentation is orthogonal to the hierarchical chain functional-algorithmic-hardware segmentation (both are shown in parallel in Figure 9.2-1). The goal of the correction level segmentation is to divide the DAC analog resources for the sake of correcting them (as opposed to dividing the DAC analog resources for the sake of generating the D/A function). At correction level, the segmentation can create correction level redundancy (the calibrated unit element (CUE) from Figure 8.4-2).

The two new segmentation levels, functional and correction levels, are independent of both the algorithmic and the hardware level redundancy. The algorithmic and hardware level redundancy concern the DAC intrinsic performance and the invested resources in the DAC core, while the functional and correction level redundancy concern the resources for the DAC correction method only and the performance improvement of the DAC core. In addition, the functional level redundancy concerns DAC flexibility as shown further in the text.

This chapter further analyses the abstraction levels of segmentation, introduces the two new levels, and proposes a new redundant segmentation. Section 9.2. discusses the various abstraction levels of the DAC segmentation. In section 9.3. , a concrete new segmentation approach is proposed. Section 9.4. provides a general discussion. Finally, conclusions are drawn in section 9.5.

9.2. ABSTRACTION LEVELS OF SEGMENTATION

Segmentation of the DAC analog resources can be implemented at various abstraction levels for various goals. Figure 9.2-1 shows a chart of the possible DAC segmentation levels and their relationships. At the left side, the discussed levels of abstraction are shown. At the right side, some exemplary schematics are shown to illustrate the concepts behind the discussed levels of abstraction. The boxes that are tagged with "**New**" represent original contributions.

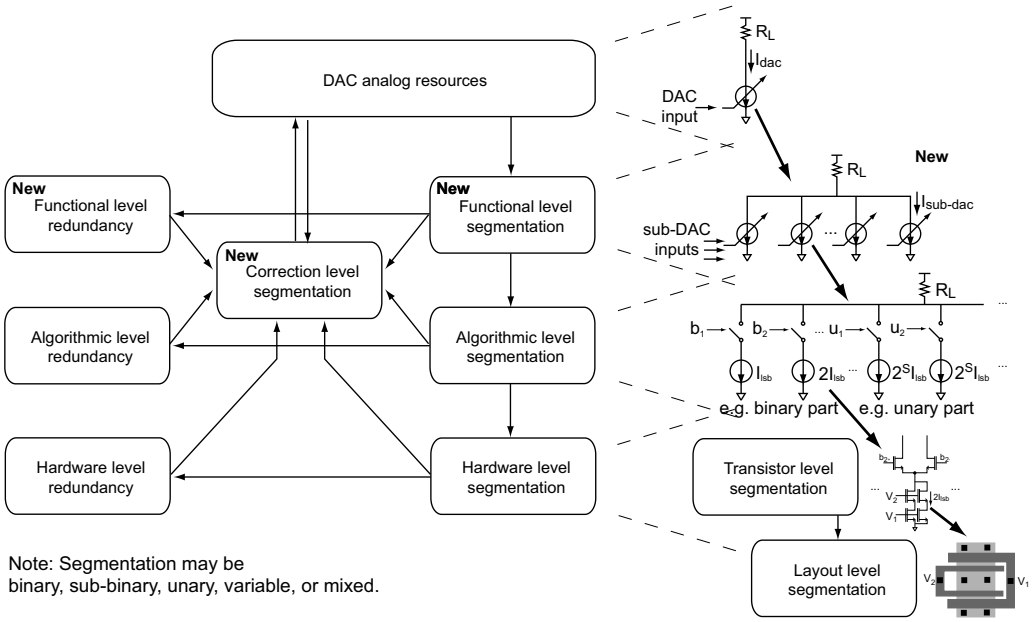


Figure 9.2-1. Analysis of the abstraction levels for DAC segmentation.

Three main abstraction levels for the DAC segmentation are considered: *functional*, *algorithmic*, and *hardware* level. At functional level, segmentation is applied to the system function, defining multiple D/A functions. This is a new level of segmentation that is introduced in this thesis. At algorithmic level, segmentation is applied to the algorithmic construction of a given D/A function defined at functional level. At hardware level, each of the algorithmic segments is implemented through hardware units. Note that the segmentation at each of these levels depends on the segmentation of the above levels. At lower levels, the segmentation can either coincide with the segmentation at the higher levels or be further refined. That is to say that the analog segments at the algorithmic level cannot be coarser (i.e. bigger) than the analog segments at functional level; and the analog segments at hardware level cannot be coarser than the analog segments at algorithmic level.

Along with the segmentation of the D/A function, there is the *correction* level segmentation. This represents the division of the analog resources in groups *for the sake of correction* (as opposed to the division of the analog resources in groups for the sake of DAC output signal generation). This is a new abstraction level of

segmentation introduced in this thesis in chapter 8 via the definition of the CUE (calibrated unit element). Note that the concept of CUE can be also applied to many DAC correction methods, e.g. mapping, current calibration. It was originally published in [77] and successfully validated in practice by [27].

The targets of segmentation can be DAC efficiency, DAC errors reduction through design, D/A functions definition, DAC errors reduction through correction (redundancy). In theory, these targets can be independently tailored at each abstraction level. In practice however, there is a correlation between the choices at the different abstraction levels. In general, the available analog resources can be segmented for high efficiency through binary segmentation or for small errors through unary segmentation.

The available analog resources can be used to define one or multiple self-sufficient D/A functions in a fixed or flexible way. This segmentation can be implemented at the functional level. When the D/A functions are nominally identical, this is unary segmentation at functional level. However, the D/A functions may be scaled with respect to each other in similar way to binary scaling, sub- and super-binary radix scaling, etc. These different scaling possibilities of the D/A functions are beyond the scope of the presented work and are not further explored in this thesis.

The available analog resources can be segmented in order to create redundancy that can reduce the errors of the analog resources. Being able to reduce the errors in the analog resources leads in general to reduction of the size of the analog current source transistors implementing the DAC analog resources. Segmentation at functional level can create functional level redundancy, which can be used by DAC correction methods working at high system level, e.g. as shown in Table 7.5-1 (high-level correction methods class). Segmentation at algorithmic level creates algorithmic level redundancy, e.g. the intrinsic redundancy of the unary current sources which can be used in many DAC correction methods as shown in Table 7.5-1 (low-level correction methods class). Segmentation at hardware level can create hardware level redundancy, e.g. unit element approach for layout [81], current source transistor placement strategies to reduce errors, e.g. [49].

Thus, three new concepts related to the segmentation of DAC analog resources are proposed and elaborated further in this thesis. The new concept of segmentation at functional level is used to define DAC flexibility in chapter 14. The functional level redundancy can be used for correction, as exemplified by many new correction methods proposed in this thesis, e.g. the high-level mapping (chapter 12) and the harmonic distortion suppression method (chapter 13). The segmentation at correction level is used to calibrate binary currents with a calibration algorithm for unary currents as shown in chapter 8.4.2. (analysis point of view), chapter 10.3. (implementation of new correction methods) and published in the related to this thesis paper [77] but also in an independent research work [69].

9.3. NEW REDUNDANT SEGMENTATION

A high functional level redundancy can be introduced in the DAC by defining more D/A functions than strictly necessary. To achieve this, the available analog resources are divided in M nominally identical D/A functions. Each of the D/A functions can be considered as an independent DAC being segmented at algorithmic level, including but not limited to a binary and a binary/unary segmentation. Such a D/A function is defined here as a sub-DAC. To describe this segmentation, Equation 9.3-1 expresses the M D/A transfer characteristics defined over an array of unit currents I_u. For the sake of simplicity, assume N bit sub-DACs with a conventional algorithmic segmentation of B bits binary LSB part and S bits unary MSB part. Thus, the N bit sub-DAC transfer characteristics,

expressed as series of the input code k, assuming segmentation with B=N-S binary LSB bits:

Equation 9.3-1

$$I_{DAC}(k)=\sum_{j=1}^{M} I_k^{(j)}, \quad where:$$

$$\text{Functional level segmentation, unary part}\begin{cases} I_k^{(1)}=\sum_{m=1}^{N-S} B_{k,m}^{(1)} \sum_{i=2^{m-1}}^{2^m-1} I_{u_i} + \sum_{m=1}^{2^S-1} T_{k,m}^{(1)} \sum_{i=2^{N-S}+(m-1)2^{N-S}}^{2^{N-S}+m2^{N-S}-1} I_{u_i} \\ I_k^{(2)}=\sum_{m=1}^{N-S} B_{k,m}^{(2)} \sum_{i=2^{m-1}+(2^N-1)}^{2^m-1+(2^N-1)} I_{u_i} + \sum_{m=1}^{2^S-1} T_{k,m}^{(2)} \sum_{i=2^{N-S}+(m-1)2^{N-S}+(2^N-1)}^{2^{N-S}+m2^{N-S}-1+(2^N-1)} I_{u_i} \\ \vdots \\ I_k^{(M)}=\underbrace{\sum_{m=1}^{N-S} B_{k,m}^{(M)} \sum_{i=2^{m-1}+(M-1)(2^N-1)}^{2^m-1+(M-1)(2^N-1)} I_{u_i}}_{\text{Binary part}} + \underbrace{\sum_{m=1}^{2^S-1} T_{k,m}^{(M)} \sum_{i=2^{N-S}+(m-1)2^{N-S}+(M-1)(2^N-1)}^{2^{N-S}+m2^{N-S}-1+(M-1)(2^N-1)} I_{u_i}}_{\text{Unary part}} \end{cases}$$

$$\text{Algorithmic level segmentation}$$

,

where $I_k^{(j)}$ is the output of the j-th N bit sub-DAC for (overall) input code k, with $j \in [1:M]$, $i \in \left[1:M\left(2^N-1\right)\right]$, S are the bits segmented in a unary way, while $B_{k,m}^{(j)}$ and $T_{k,m}^{(j)}$ are the switching matrices, respectively for the binary and the thermometer parts of the j-th sub-DAC. The switching matrices determine which analog segments are used to construct the DAC analog output signal at a given input code k. Examples of analog matrices for binary and unary algorithmic segmentations are shown in Equation 6.2-3 and Equation 6.2-17. Note that for $S=N$, the equations represent fully thermometer sub-DAC transfer characteristics and for $S=0$, the equations represent a fully binary sub-DAC transfer characteristic. Figure 9.3-1 shows the equivalent high level circuit of Equation 9.3-1.

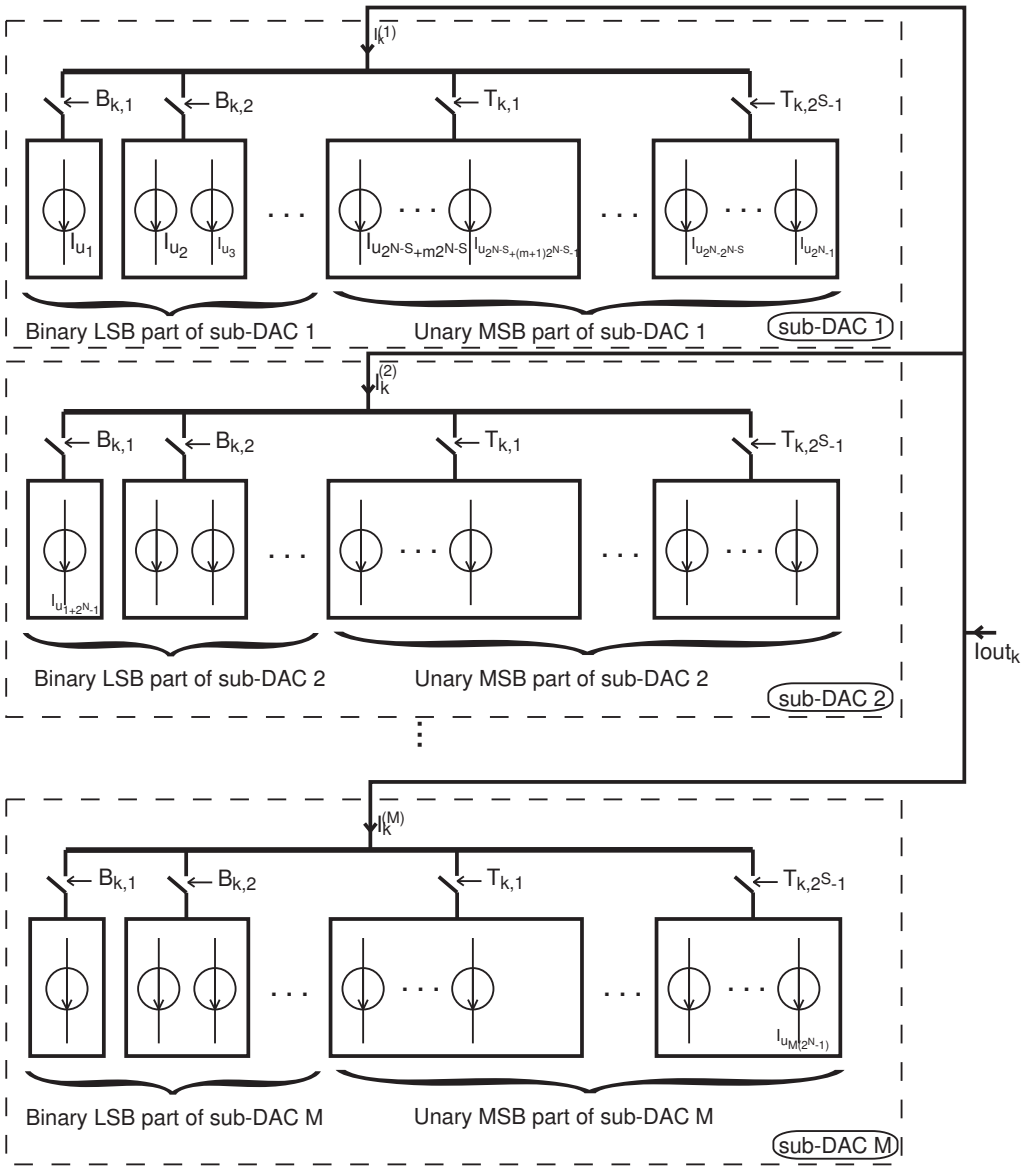


Figure 9.3-1. DAC unary functional segmentation with binary/unary algorithmic segmentation.

Alternatively to the conventional DAC approaches, the proposed new functional level unary segmentation features multiple nominally identical D/A functions $I_k^{(j)}$, i.e. redundancy at the high functional level. If the functional level is flattened, i.e. only the algorithmic level is considered, then the new segmentation features multiple binary sets of currents. Thus, the new segmentation features redundancy of the binary sets of currents, if compared to the conventional approaches. Instead of only one set of binary currents (analog segments) like in the conventional approaches, the proposed segmentation allocates multiple binary sets. That is to say that each binary current has at least one nominally identical replica. Thus, the DAC binary currents are described as $I(b)(j)$, with b being the binary bit and j being one of the multiple binary sets. The main

advantage of this segmentation approach is its high- and low-level intrinsic redundancy. At high level, multiple D/A functions are defined around the multiple binary sets of currents. At low level, each binary current has a redundant replica with respect to the realization of a single D/A function. This intrinsic redundancy can be used either in various DAC correction methods or for DAC flexibility.

Note that the new segmentation only concerns the division of the existing DAC analog resources and hence it does not require more analog resources. The area of the DAC analog resources is determined by the size of the DAC current source transistors, cascode transistors, current switches, and optionally cascode switches, as suggested in chapter 2.3. Figure 9.3-2 shows a block diagram of a DAC having an exemplary segmentation as suggested here.

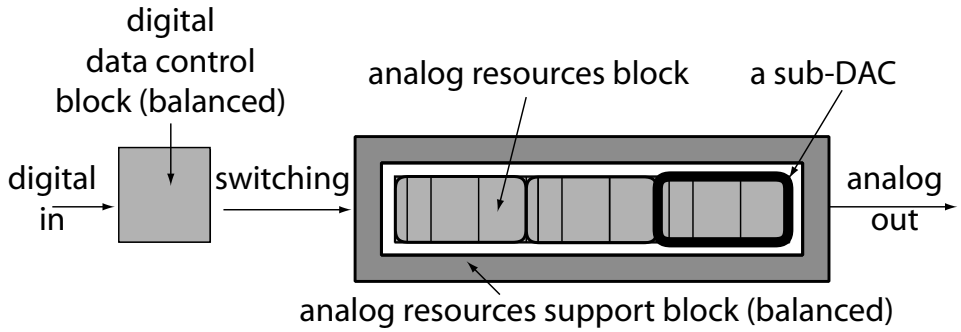


Figure 9.3-2. A generalized block diagram a DAC based on parallel sub-DACs.

As shown, multiple sets of binary units are defined within the analog resources block. The size of the analog resources block remains unchanged when compared to the conventional binary segmentation (Figure 2.4-1), unary segmentation (Figure 2.4-3), and mixed segmentation (Figure 2.4-4). The sizes of the digital data control block and the analog resources support block mainly depend on the algorithmic segmentation and not that much on the functional level segmentation. That is to say that DACs that are highly area and power efficient but feature redundancy for correction (small errors) can be designed via segmentation at functional level to two unary D/A functions (two sub-DACs), while each sub-DAC features full binary segmentation (i.e. is efficient) at algorithmic level. The binary algorithmic segmentation guarantees efficiency, while the DAC errors are corrected via DAC correction methods, thanks to the redundancy at functional level. Note that two sub-DACs is a minimum number to create a useful overall functional redundancy. A number of examples can be pointed out, from both independent research projects and this thesis: segmented DEM (via two binary sub-DACs) [60]; calibration of binary currents [47], chapter 16.3. ; high-level mapping (chapter 12); harmonic distortion suppression (chapter 13); etc. It is important to note that the new DAC segmentation provides possibilities for flexibility, too. Since multiple D/A functions are defined, these can be used for flexible design, functionality and performance of the DAC platform, as further discussed in chapter 14.

9.4. DISCUSSION

The proposed functional level segmentation is new for the field and hence it cannot be compared to the existing conventional segmentation approaches. That is why a comparison is done here at algorithmic level only. Table 9.4-1 compares the proposed new redundant segmentation with the conventional segmentation approaches.

Table 9.4-1. Summary of the DAC segmentation approaches and their features

Target	DAC segmentation	Consequences
High *Efficiency*	Binary (algorithmic level)	No *Redundancy*; compromised high-speed *Performance*
Efficiency, *Redundancy*	Sub-binary radix (algorithmic level)	Highly compromised high-speed *Performance*
Performance	Unary (algorithmic level)	High *Redundancy*, low *Efficiency*
Balance between *Efficiency* and *Performance*	Mixed binary LSB/unary MSB (algorithmic level)	Balanced consequences between efficiency/performance/algorithmic redundancy
Redundancy, ***Flexibility***, balance between *Efficiency* and *Performance*	**Parallel sub-DACs** (new unary functional segmentation; multiple binary sets)	Balanced consequences between efficiency/performance/ algorithmic redundancy/ functional redundancy/flexibility

For the highest efficiency, binary algorithmic segmentation should be chosen. However, it features high design requirements for the errors of the DAC analog resources, while providing no redundancy for calibration. Sub-binary radix segmentation introduces redundancy but compromises the DAC dynamic performance due to complications in the hardware implementation. The unary segmentation features both low design requirements (and hence small errors) and redundancy for error correction. However, it requires a lot of design resources, e.g. area and power. The popular mixed algorithmic segmentation, binary LSB/unary MSB, balances the advantages of binary and unary segmentation approaches to provide both efficiency and performance. At algorithmic level, it features a single binary set of analog units (e.g. currents) and multiple unary analog units. The new redundant segmentation introduces functional level redundancy, independent of the choice of the algorithmic segmentation. At algorithmic level only, the new redundant segmentation features multiple binary sets of analog units and optionally multiple unary analog units. Therefore, it is still a balanced segmentation approach as the mixed binary LSB/unary MSB approach, while new advantages are added: flexibility, low-level redundancy of the binary segments, and high-level functional redundancy (independent of the optional lower level algorithmic redundancy).

9.5. CONCLUSION

Two new levels of segmentation are proposed: the functional and the correction level. These levels of segmentation can be used to introduce functional and correction redundancy. The functional redundancy introduces multiple D/A functions, i.e. redundancy in the D/A functions. The correction redundancy is considered as the calibrated unit element (CUE) discussed in details in chapter 8.4.2. The functional and correction level redundancies are independent of the conventional algorithmic and hardware redundancies. That enables a variety of new high level correction methods that use functional and correction level redundancies, while the DAC core is independently optimized at algorithmic and hardware level.

A new redundant segmentation is proposed. The segmentation uses unary functional redundancy, while the choice of the algorithmic redundancy remains free. Multiple nominally identical D/A functions are defined over the DAC analog resources. The D/A

functions can be further segmented at algorithmic level in binary/unary/mixed way. The new redundant segmentation adds three main advantages to the conventional mixed binary LSB/unary MSB algorithmic segmentation. These advantages are flexibility, low-level redundancy of the binary segments, and high-level functional redundancy. Further, DAC correction methods can transform the created intrinsic redundancy into performance improvement.

Discussion and comparison is provided between the conventional algorithmic level segmentations and the new redundant segmentation. To compare these, the new redundant segmentation is flattened down to algorithmic level as segmentation featuring multiple binary sets of analog segments (binary currents). It is shown that the new redundant segmentation balances the advantages of the binary and unary algorithmic segmentations and adds the new advantages of flexibility, low-level redundancy of the binary segments and high-level functional redundancy, while requiring similar hardware resources as the mixed binary LSB/unary MSB algorithmic segmentation.

10. Chapter

NEW METHODS FOR SELF-CALIBRATION OF CURRENTS

This chapter presents a new self-calibration method for all currents in current-steering DACs. The new method is based on simple circuits that do not require any special high design specifications. For self-measurements, a current comparator is used. For self-correction, small calibrating DACs (CALDACs) are attached to every current for calibration. The algorithms that control the calibration process are simple and can be implemented as finite-state-machines (FSMs).

Self-calibration automatically detects and corrects production deviations (technology tolerances). To save area, the intrinsic accuracy of a DAC, and hence its static linearity, expressed in bits, is designed lower than its resolution. Before or during chip usage, smart built-in circuits and automated operations improve the accuracy of the DAC. This process also corrects any production deviations and hence improves the chip yield. Generally, these operations build on two sub-processes: self-test and self-correction. Firstly, information is acquired about the deviation of the actual targeted value from the reference. Secondly, correction activities minimize that deviation. Both self-test and self-correction are often combined and simply referred to as self-calibration.

This chapter presents three main novelties with respect to the available current calibration methods in the open literature. The first novelty is that for the unary currents self-calibration, the proposed method can compensate within the calibration algorithm for the errors of the self-measurement process. The second novelty is the use of correction level segmentation for calibration of DAC scaled binary currents. The third novelty is the direct self-calibration of binary currents, using multiple sets of binary currents, which can be implemented with either functional or hardware segmentation.

10.1. INTRODUCTION

The linearity of Digital-to-Analog Converters (DACs) is a key characteristic for their performance. It determines the accuracy of the static code conversion and influences the quality of the dynamic analog output. The static figures of merit INL and DNL are used to characterize the linearity of the DAC transfer characteristic. While the DNL can be relaxed by applying either unary or mixed binary LSB/unary MSB algorithmic segmentation, the INL is fundamentally coupled with the amplitude errors of the analog elements that generate the output, see chapter 6 (Equation 6.2-14 and Equation 6.2-22), [73], [46]. The DAC dynamic performance also partly depends on the accuracy of the DAC currents. As shown in chapter 5, all dynamic error mechanisms can be modulated by the amplitude errors of the current cells, creating different error charges that deteriorate the DAC linearity, e.g. as shown in Figure 5.4-2. To limit these errors two main approaches are available.

Those DAC designs without calibration limit these amplitude errors, using large enough analog elements [46], [6]. Alternatively, smart DAC designs use additional logic and operations, to improve the linearity, [66], [36], [70], [82], [71].

Self-calibration of AD/DA converters has gained momentum in the past few years. The increased interest is largely due to the prospective of the CMOS development: positive signs for digital logic (e.g. reduced area, increased speed) and negative signs for analog circuits (e.g. poor matching, lower supply voltages), see [56].

Section 10.2. presents a self-calibration method for nominally identical unary current sources. Section 10.3. extends the calibration of nominally identical currents to calibration of scaled currents by using a correction level segmentation (calibrated unit element) that is different from the DAC algorithmic segmentation. The section discusses how an optimal DAC, in terms of area, can be designed. Section 10.4. presents a self-measurement method for scaled binary currents. Section 10.5. provides a general discussion and finally conclusions are drawn in section 10.6.

10.2. SELF-CALIBRATION OF UNARY CURRENTS

This section describes a method, which via extra components and operations, improves the DAC linearity through improving the accuracy of the DAC MSB unary currents. This allows relaxed design requirements for the DAC intrinsic core, in terms of matching accuracy, greater design margins, and smaller area. In practice, the presented calibration algorithm targets area efficiency without deteriorating dynamic performance. As long as the saved silicon area is greater than the area occupied by the calibration circuitry, the application of calibration is area efficient.

10.2.1. New calibration method

The presented algorithm improves the linearity of current-steering DACs with unary or mixed binary LSB/unary MSB segmentation (algorithmic, hardware, or correction level) via calibrating nominally identical current sources. It regulates the current of the unary current sources via embedded calibrating DACs (CALDACs, e.g. as shown in Figure 8.5-4) in the current cells, similarly to [71]. The calibration method also uses a current comparator and a single additional temporary current source. The input offset of the comparator is compensated within the calibration algorithm. Every unary current source is individually calibrated to a common reference, through successively increasing the current from its bi-directional CALDAC, until the comparator indicates that the value of the reference is reached. The input offset of the comparator is compensated via the additional temporary current source. At system level, the main types of errors are

corrected. The actual implementation of the algorithm is an 8-state Finite-State-Machine (FSM).

The concept of the algorithm is shown in Figure 10.2-1. The reference current source I_{ref} is composed of the sum of all binary current sources from the LSB segmented part, plus one LSB current source. The algorithm adjusts the unary current sources to this reference using a current comparator. To compensate for its unavoidable input offset I_{offset} (shown in gray) an extra temporary current source I_{temp} is used. The FSM, the comparator, and I_{temp} are common for the calibration of all currents. The CALDACs are different instances of the same design. These CALDACs are embedded within each unary DAC current cell.

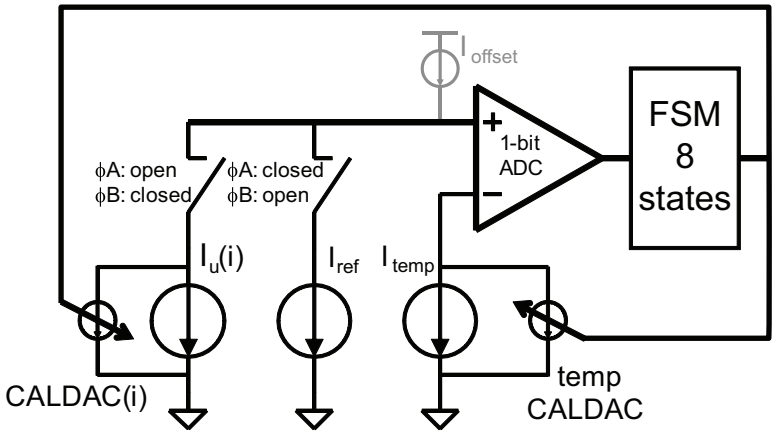


Figure 10.2-1. Calibration scheme for unary currents.

The algorithm is based on two phases: ϕA and ϕB . During ϕA , I_{temp} is calibrated to I_{ref}. The comparator offset is inherently also recorded. Thus, the current of I_{temp} at the end of ϕA is:

Equation 10.2-1

$$\phi A: \quad I_{temp} \approx I_{ref} + I_{offset}$$

The result is memorized in the CALDAC of I_{temp}. During ϕB , instead of I_{ref}, the i^{-th} unary current source, $I_u(i)$, is connected to the input of the comparator. The $I_u(i)$ is calibrated to I_{temp}. Due to the exchanged positions of the currents being calibrated at the comparator inputs, the recorded I_{offset} in ϕB is the opposite of the one recorded in ϕB . Therefore, $I_u(i)$ at the end of ϕB is:

Equation 10.2-2

$$\phi B: \quad I_u(i) \approx I_{temp} - I_{offset}$$

Substituting Equation 10.2-1 in Equation 10.2-2, I_{offset} is cancelled and $I_u(i)$ is calibrated to I_{ref}, free of offset.

Equation 10.2-3

$$I_u(i) \approx I_{ref}$$

The result is stored in the i^{-th} CALDAC and the operation is repeated for the next unary current source. The process continues by calibrating all unary current sources.

The actual current adjustment, i.e. satisfying Equation 10.2-1 and Equation 10.2-2, is a process of successive approximation that can be described as follows. First, both currents from the two opposite sides of the calibration equations are compared and a decision for the polarity of the calibration is taken, i.e. either the calibrating current has to be added or it has to be subtracted. To approach the reference current, the CALDAC word is increased by a 1 CALDAC LSB and the output of the current comparator is evaluated. Thus, the calibrating current at moment *nT* is:

Equation 10.2-4

$$I_{cal}\left(nT\right)=\left(-1\right)^{P}\cdot\left(X\left(nT\right)\cdot I_{LSBCALDAC}\right),$$

where $P\in 0,1$ is the calibration polarity, *X(nT)* is the CALDAC word at moment *nT*, and $I_{LSBCALDAC}$ is the 1 LSB step of the CALDAC.

The currents are considered calibrated when the output of the current comparator changes compared to the preceding iteration. Therefore, a quantization error I_q must be added to Equation 10.2-1 and Equation 10.2-2., where I_q is the difference between the calibrated current source and the reference. The sign of I_q can be controlled via the condition of the calibration process completion. When the comparator generates a different result at the moment *mT* than the result at moment *(m-1)T*, this is an indication that the targeted reference is reached. Thus, the calibration result can be either $X\left(mT\right)$ or $X\left(\left(m-1\right)T\right)$. The latter is in fact $X\left(\left(m+1\right)T\right)$, since the algorithm firstly detects the crossing of the threshold and then returns one step back, i.e. $X\left(\left(m+1\right)T\right)=X\left(\left(m\right)T\right)-1=X\left(\left(m-1\right)T\right)$. In the case where the result is $X\left(mT\right)$, the currents that are smaller than I_{ref} are calibrated with a positive I_q and the those greater currents with a negative I_q. In the case where the result is $X\left(\left(m-1\right)T\right)$, it is the other way around. Thus, if both cases are combined, the sign of I_q for either currents can be predicted. An example for ϕA is provided in Figure 10.2-2 and an example for ϕB is provided in Figure 10.2-3.

In Figure 10.2-2 (ϕA calibration Probability-Density-Function (PDF)), the statistical pre-calibration distribution of I_{temp} currents is shown in light gray. All actual samples smaller than the reference I_{ref} are calibrated with a CALDAC word, $X\left(\left(m+1\right)T\right)$, which is 1 CALDAC LSB smaller than $X\left(mT\right)$ and all the samples larger than the reference are calibrated with $X\left(mT\right)$, with *mT* the period when the comparator changes its state. In Figure 10.2-3 (ϕB calibration PDF), the statistical pre-calibration distribution of I_u currents is shown in light gray. All actual samples greater than the reference I_{temp} are calibrated with a CALDAC word $X\left(\left(m+1\right)T\right)$ which is 1 CALDAC LSB smaller than $X\left(mT\right)$ and all the samples smaller than the reference are calibrated with $X\left(mT\right)$, with *mT* the period when the comparator changes its state.

For a quantized type of calibration, the resulting post-calibration distribution has the minimum possible width of $I_{LSBCALDAC}$ per a phase of calibration, i.e. per either ϕA or

ϕB. Providing a sufficiently small calibration step, the calibration quantization error can be approximated as uniformly distributed. This post-calibration distribution has an offset $\frac{1}{2} I_{LSBCALDAC}$ in its mean value, with regard to the reference: $\overline{I_{q\phi A}} = -\frac{1}{2} I_{LSBCALDAC}$ for ϕA and $\overline{I_{q\phi B}} = \frac{1}{2} I_{LSBCALDAC}$ for ϕB .

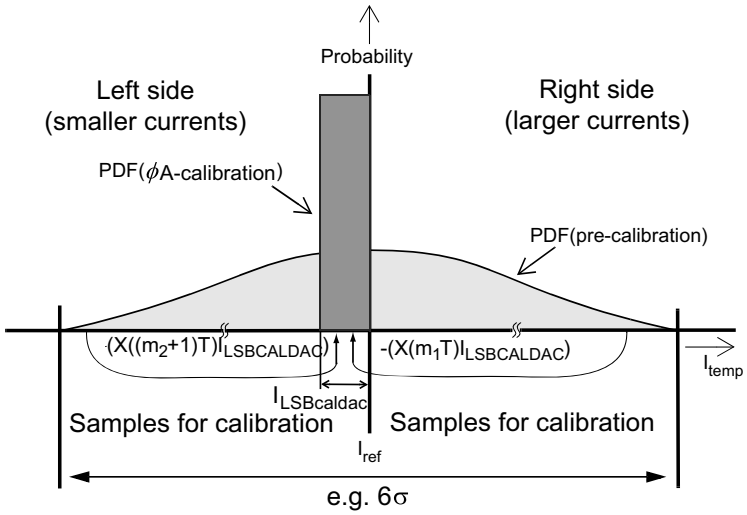


Figure 10.2-2. Pre-calibration and post-ϕA-calibration PDFs of I_{temp}.

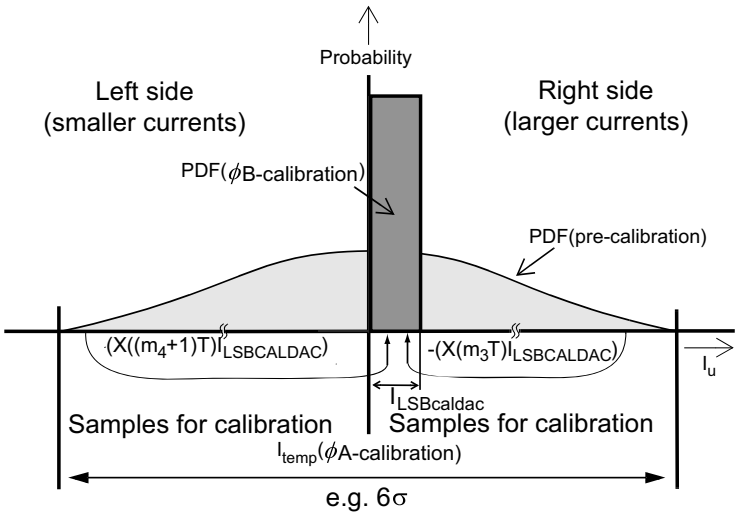


Figure 10.2-3. Pre-calibration and post-ϕB-calibration PDFs of I_u (m_1, m_2, m_3, and m_4 indicate different parameters of equality moment m)

The proposed algorithm inherently compensates for this error by controlling the signs of the mean quantization calibration errors $\overline{I_{q\phi A}}$ and $\overline{I_{q\phi B}}$, during ϕA and ϕB, respectively:

Equation 10.2-5

$$\phi A: \quad \overline{I_q} = \overline{I_{q\phi A}}; \qquad \phi B: \quad \overline{I_q} = \overline{I_{q\phi A}} - \overline{I_{q\phi B}}$$

providing that $\overline{I_{q\phi A}} = \overline{I_{q\phi B}}$,

Equation 10.2-6

$$\phi B: \quad \overline{I_q} = 0$$

Therefore, the resulting post-calibration PDF, shown in Figure 10.2-4, is centered around I_{ref}. The post-calibration PDF is a result of the two independent uniform PDFs, applied in both phases. Providing a sufficiently small calibration step size, the resulting post-calibration PDF can be approximated with a triangular shape, as shown in Figure 10.2-4. The error due to this approximation comes from the fact that at the corners of the post-calibrated PDF ($\pm I_{LSBCALDAC}$), the probability is not zero, but the product of the corner probabilities of ϕA and ϕB .

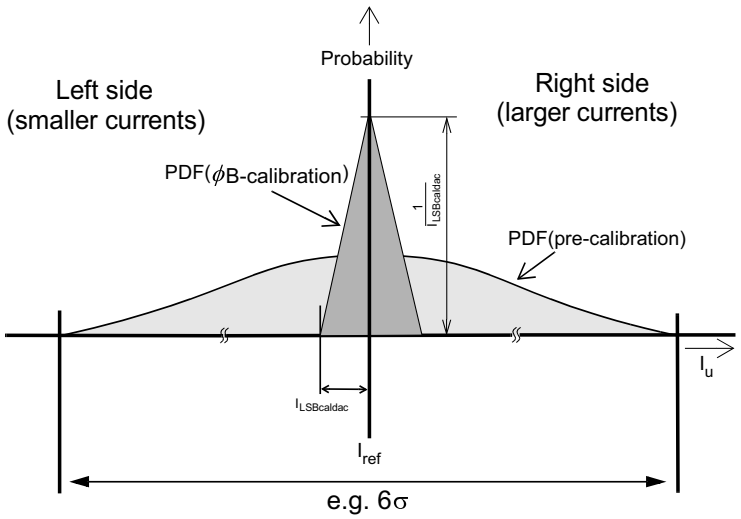


Figure 10.2-4. Pre-calibration and post-calibration PDFs of I_u.

For the triangular PDF, the post-calibration standard deviation of the thermometer current sources is derived, [83]:

Equation 10.2-7

$$\sigma_{t_{PC}} = \frac{I_{LSBcaldac}}{\sqrt{6}}$$

Further in the thesis, this post-calibration PDF is empirically proven in Figure 16.2-5.

The calibration logic is organized as a FSM. Its state diagram is shown in Figure 10.2-5. After reset, the FSM is set to state 0 and it waits for a control signal *Start_cal*. The loop at *state* 3 implements ϕA of the algorithm. The loop at state 6 implements ϕB . The WHILE loops and the IF conditions in these two states realize the successive quantized approach method for calibration, described above. The first WHILE loop implements the incrementing of the CALDAC word, *X(nT)*, until the moment *mT*, when the comparator changes its output. Then, based on the polarity of calibration, a decision is taken to

accept either *X(mT)* or *X((m-1)T* as a final result of the calibration. Finally, the loop from *state* 5 to *state* 7, containing ϕB , is repeated for all unary current source. The algorithm concludes, when all the thermometer current sources are calibrated.

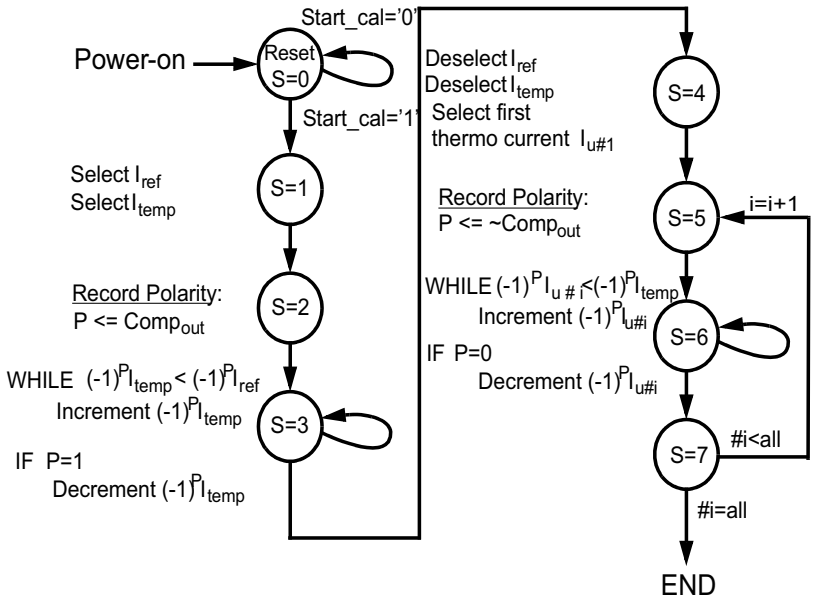


Figure 10.2-5. State diagram of the calibration algorithm.

10.2.2. Conclusions

A new method for calibration of nominally identical currents is proposed. The method makes use of components with relaxed matching requirements, e.g. a current comparator, multiple calibrating DACs (CALDACs), digital logic. The proposed calibration algorithm compensates for the comparator input offset and optimizes the width of the post-calibration current distribution. The realization of the calibration algorithm is simple and efficient – an 8 state finite-state-machine (FSM). This self-calibration method is suitable for the MSB unary currents of a current steering DAC. A chip implementation based on this algorithm is realized and further explained in this thesis in chapter 16.2.

10.3. A CALIBRATION METHOD FOR GENERIC CURRENT-STEERING D/A CONVERTERS WITH OPTIMAL AREA SOLUTION

Current-steering (CS) DACs may achieve high resolutions and high static linearity but at a price that is paid for in large power consumption and silicon areas. Two independent approaches may save area: binary segmentation [81] and self-calibration. The binary DAC architectures do not spend any extra design resources for the DAC digital control block, as opposed to the unary architectures. The self-calibration corrects for and hence tolerates intrinsic inaccuracies, demanding smaller silicon areas. Section 10.2. presents a current calibration method for unary current. However, this method corrects only nominally identical currents and thus is directly applicable only to unary segmented DAC currents. Since the algorithmic segmentation may be optimized for dynamic performance, the calibration efficiency may be low. Furthermore, the method is not directly applicable to binary DACs, because their current sources are not nominally identical and hence the reuse of the calibration apparatus is difficult.

This section presents a new start-up calibration method for current-steering D/A converters, based on a 1 bit ADC. The section demonstrates how the correction level segmentation can be used to calibrate scaled currents. It proposes a new current cell that allows calibration of non-identical current sources by way of a shared calibration apparatus. The current cell uses parallel self-calibrated unit elements. Each of them is calibrated individually and when all assembled back, the accuracy of the current sources is improved. The correction level segmentation is independent of the algorithmic level segmentation. Therefore, a minimal area solution through optimizing the calibration strength exists, since the method is not limited within only the nominally identical unary currents, like most of the existing current calibration approaches. A general discussion on the new calibration method is offered and conclusions are drawn.

Thus, a simple calibration scheme is presented that relies on a new architecture for the DAC current cells to reuse the calibration apparatus for different currents. This solution, defined in chapter 9 as calibration level segmentation, has several advantages. It leads to small high-resolution self-calibrated DACs, an algorithmic segmentation (e.g. chosen for dynamic performance) that is independent of the area optimization through calibration. This adds a new degree of freedom in the trade-offs of the DAC design.

10.3.1. New self-calibrating current cell for a generic DAC architecture

The method described in section 10.2. is applied to nominally identical current sources. These can be the unary current sources that result from the algorithmic unary segmentation. However, nominally identical current sources can be defined independently of the algorithmic segmentation. For the sake of calibration, these current sources can be defined as calibrated unit elements (CUE), e.g. generating I_{cue}, see Figure 10.3-1. The CUE cells can be used as unary current cells for calibration e.g. in the calibration scheme shown in Figure 10.2-1. Calibration of nominally identical current sources allows a shared calibration circuitry, in terms of the measurement device, the digital control logic, the temporary current source I_{temp}, and instances of identical CALDACs. These calibrated unit elements (CUEs) can construct the non-identical scaled current sources. For example, in a binary DAC, let the M^{-th} bit be constructed by 2^L identical parallel CUEs. Then, the $(M+1)^{-th}$–bit is constructed by 2^{L+1} parallel CUEs, etc. Thus, every signal current source to be calibrated is composed of CUE(s), as shown in Figure 10.3-1. The start-up calibration apparatus individually corrects every I_{cue}. Consequently, the accuracy of the (binary-scaled) signal current sources is improved and the target static accuracy of the DAC system is achieved.

In Figure 10.3-1, NMOST M1 generates the coarse current $I_{cue}(i)$ and the correcting fine current is generated by the CALDAC connected to the drain of M1. The cascode M2 shields the generation of $I_{cue}(i)$ from the calibration and operational circuits and increases the output impedance of the self-calibrated current source $I_{cue}(i)$. The switches M3a and M3b redirect the current to be used either by the D/A converter or by the calibration circuit. To construct the scaled binary current, several CUEs are connected in parallel, but each of them is calibrated separately. This would increase the actual total area with respect to the conventional compact current cells, though both approaches have the same active (gate) area. However, with the evolution of the CMOS technology, the extra area of the proposed method will be getting smaller, due to the general shrinkage of the passive area.

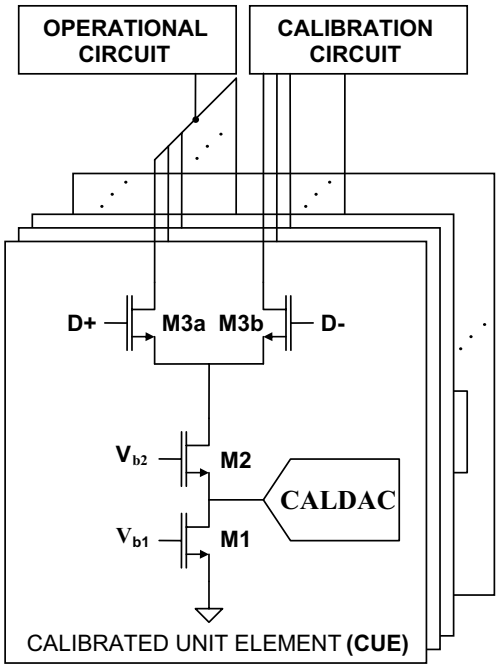


Figure 10.3-1. A self-calibrated scaled current cell composed of parallel CUEs.

Furthermore, the non-calibrated binary current sources and one additional LSB current source are summed in parallel during ϕA to construct I_{ref}, see Figure 10.2-1. This approach minimizes the DNL error in the DAC transfer characteristic at the transition points between non-calibrated and calibrated bits, as discussed in section Figure 8.3-2. The non-calibrated current sources and the remaining post-calibration mismatch in the calibrated current sources set the post-calibration static DAC accuracy. An optimal DAC design, in terms of silicon area, has a balance between the calibrated and non-calibrated current sources, i.e. between the required intrinsic accuracy and the devoted resources for calibration. This trade-off is discussed in details in the following section.

10.3.2. Area driven optimum of the level of calibration

Two design specifications set the calibration strength: the post-calibration static DAC accuracy set by the required INL and the intrinsic DAC accuracy set by the initial (intrinsic) INL_0. While the first design specification is often set by the targeted application, e.g. INL<0.5LSB, the second is a trade-off. It influences the area of the converter. This specification also sets the required calibration strength, being from *weak* to *aggressive*. On the one hand, an *aggressive* calibration takes place when the DAC accuracy is intrinsically very poor and improved to the specified level. In this case, the majority of current sources are calibrated and the CALDACs have large full-scales, whereas the most aggressive calibration would calibrate all current sources but the CALDACs would have extremely high full-scales. On the other hand, a *weak* calibration is when the intrinsic DAC accuracy is close to the specified level. In this case, a minority of current sources is calibrated and the CALDACs have small full-scales, whereas the weakest calibration would mean little or no need for calibration – the intrinsic accuracy of the DAC is equal to the specified level. Table 10.3-1 summarizes these area trade-offs.

Table 10.3-1. A conceptual comparison between weak and aggressive calibration.

Weak calibration →	**Optimum**	← **Aggressive** calibration
Large area of the coarse current sources	⇨	**Small** area of the coarse current sources
Small area of the CALDACs	⇦	**Large** area of the CALDACs
Constant area of self-test and control circuitry		

Thus, the area of the coarse current sources and the area of the CALDACs balance an area trade-off for self-calibrating D/A Converters. Figure 10.3-2 conceptually shows the area trade-off between the area of the coarse current source transistors and the CALDACs for a calibrated DAC to achieve a given target accuracy.

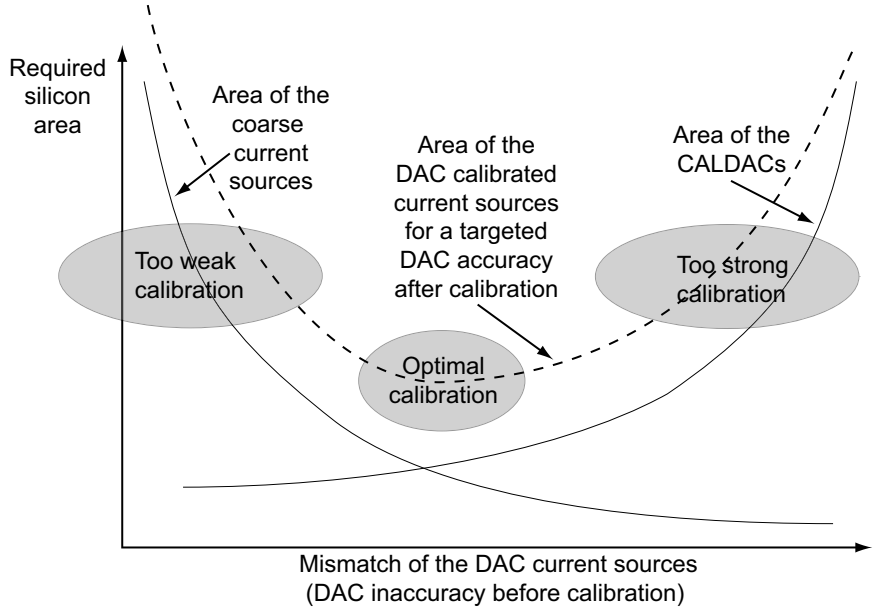


Figure 10.3-2. Area trade-off between coarse current sources and CALDACs of a calibrated DAC.

For weak calibration, only a few currents are calibrated but accurate coarse current sources are required. For strong calibration, in accurate current sources are tolerated but a lot of currents are calibrated. Thus, on one hand, as shown in [6], the active silicon area of the coarse current sources is inversely proportional to the square of their mismatch. On the other hand, the active silicon area required for the array of CALDACs depends, firstly, on the area of each individual CALDAC (dependence on the DAC intrinsic accuracy), secondly, on the number of used CALDACs. This number can be derived from the INL results of section 6, considering that both the calibrated and the non-calibrated current sources contribute to the DAC post-calibration INL.

The analysis graphs of Figure 10.3-2 is a higher level abstraction and it considers only active areas, so a great number of details are discarded.

10.3.3. Discussion

The presented method for calibration of scaled current sources can be applied to all DAC architectures, i.e. binary, unary, and mixed, because it does not rely on the algorithmic segmentation. Depending on the optimal calibration strength with respect to area optimization, the calibration may appear in three cases, as shown in Figure 10.3-3. The first option is for the calibration to include a portion of the binary bits. The second option is for the calibration to include only all the unary bits. The third option is for the calibration to exclude a portion of the unary bits.

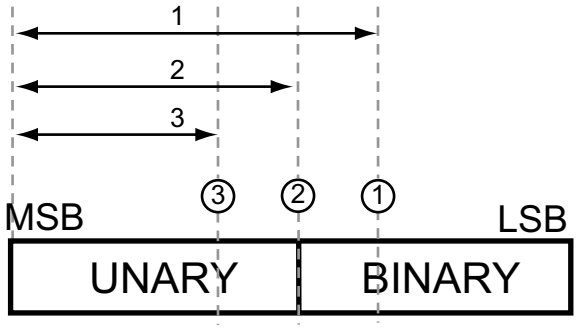


Figure 10.3-3. Correction level segmentation with CUEs with respect to a mixed binary/unary algorithmic segmentation.

In the first case, when the calibration includes a portion of the binary bits, the calibrated binary and unary bits are constructed with parallel CUEs. For example, if the 7^{th} bit is the first unary bit, but the optimal calibration should begin from the 5^{th} bit, then during the normal operation of the DAC, the 5^{th} bit is a CUE, the 6^{th} bit is 2 CUEs, and the unary current sources are 4 CUEs. During the calibration phase, every CUE is calibrated separately. The second case is the conventional DAC design where the binary bits are not calibrated and only the unary bits are calibrated, e.g. [35], section 16.2. . Then, every unary current cell is a single CUE. In the third case, the optimal calibration excludes a portion of the unary bits. For example, if the 7^{th} bit is the first unary bit, but the optimal calibration begins from the 9^{th} bit, then 4 unary current sources construct 1 CUE. That is to say, the unary current sources are calibrated in packs of 4, i.e. a single CALDAC is shared among 4 thermo currents.

The presented method advances the state-of-the-art techniques with the freedom of setting the calibration strength independently of the DAC segmentation. This will lead to smaller and more compact DAC chips. Smaller DACs will reduce the direct manufacturers' costs. More compact DACs will also reduce the parasitic capacitances and resistances, improving the performance. Furthermore, the presented analysis gives a picture of the trade-off on the calibration strength, in terms of the intrinsic DAC accuracy. Practically, the presented analysis can be concluded to a simple rule of thumb: *"The silicon layout area of the calibration circuitry, i.e. CALDACs, measurement devices, digital logic, etc., should be equal to (or slightly less than) the area of the array of the coarse current sources."* Such an approach should be close to the optimal area, as suggested by Figure 10.3-2. Thus, the presented analysis will help designers to quickly estimate a minimal area for a self-calibrated DAC. This will reduce the circuit design cycle allowing a faster market-release of the self-calibrated DAC chip or the system that includes it.

10.3.4. Conclusions

The proposed new calibration method is applicable to D/A converters with generic architectures. The segmentation for calibration is different from the DAC algorithmic segmentation for data conversion. The unary segmentation at correction level is defined as the nominally identical CUEs. Further, the method is based on a 1 bit ADC appended with digital logic. Most of the calibration circuitry is reused. The sources of calibration errors are compensated at system level. The method makes the optimization of the calibration strength independent of the choice of the converter algorithmic segmentation and hence allows calibration of binary converters. The presented analysis shows that for an area-optimal calibration, the array of the coarse current sources and the calibrating circuitry should occupy approximately equal areas.

10.4. A CALIBRATION METHOD FOR BINARY SIGNAL CURRENT SOURCES

This section presents a new start-up calibration method for binary signal current sources in a D/A Converter. The method uses a 1 bit ADC (sign detector) for self-measurement and small calibrating DACs (CALDACs) for self-correction. An NMOS-based current for calibration is subtracted from a PMOS-based reference current. Their residue is minimized by adjusting the current of the NMOS-based current source. The nominal value of the two opposite currents is equal to two times the MSB current of the DAC (e.g. unary current in a mixed binary/unary algorithmic segmentation). To calibrate the LSB binary current sources, a few extra calibration steps guarantee an accurate division of the PMOS-based current. The calibration of both unary and binary current sources to a common PMOS-based reference makes the method universal, i.e. independent of the DAC architecture (algorithmic segmentation). This is the first method that uses the calibration routine to accurately divide the reference current and hence to calibrate for nominally different currents, leading to the following two important properties. Firstly, the post-calibration current distribution and hence DAC INL entirely depends on a single design parameter – the CALDAC calibration step. Secondly, the intrinsic accuracy, i.e. the size of the current source transistors, can be fairly traded for the resolution of the CALDACs, i.e. their physical area. These two properties improve DAC yield and minimize silicon area.

In sections 10.2. and 10.3. , start-up methods are proposed to calibrate nominally identical current sources. The start-up methods execute the calibration routine with chip power-up. The results are memorized and later used in the D/A conversion process. The isolation of the D/A conversion process from the calibration activities is a main advantage of this method with respect to others, e.g. background calibration [66] and output compensation [36].

These methods can achieve excellent results for the calibration of the MSB unary current sources or CUE. If only the MSB unary currents ($I_u(i)$) are calibrated, the overall static DAC performance is determined by the non-calibrated binary current sources. A simple reason may be given for the calibration of only the MSB $I_u(i)$: they are nominally identical, so a common reference can be used. Therefore, the calibration results are closely linked to the algorithmic segmentation of the DAC. A solution to break this dependence is proposed in section 10.3. . The improved method can calibrate a reasonable part of the binary LSB currents, by introducing a level of abstraction: calibrated unit element (CUE), $I_{cue}(i)$. This concept allows a common reference, because the calibration is executed on nominally identical currents, i.e. $I_{cue}(i)$. At a higher abstraction level, the calibrated, i.e. already accurate, CUEs are used to construct the signal current sources. The work presented in section 10.3. concentrates on the optimum area solution, balancing between the area needed for the main, coarse, current sources

(i.e. their intrinsic accuracy) and the area needed for calibration, so that the non-calibrated and calibrated currents deliver the targeted static DAC linearity.

Alternatively to the CUE approach, this chapter proposes a new method to calibrate for all binary scaled current sources. The common reference problem is solved by accurately dividing the reference within the calibration routine. The new method does not leave any non-calibrated current sources. Therefore, the DAC post-calibration static linearity depends only on a single design parameter: the CALDAC calibration step.

10.4.1. Calibration of scaled currents

The new calibration method features two important improvements. Firstly, new system and transistor level schemes are proposed, which use less hardware resources. Secondly, a new algorithm is proposed, which accurately divides the reference in order to calibrate the binary currents. Note that both improvements are independent and therefore they can be always used partially in combination with the previous approaches.

The new calibration scheme, shown in Figure 10.4-1, features not only a new calibration algorithm (coded in FSM2) but also two important improvements. Firstly, all signal currents, i.e. binary and unary, are calibrated to a PMOS-based reference, I_{temp}. The current for calibration is subtracted from I_{temp} and the sign of the residue is evaluated by FSM2. Secondly, the functionality of FSM2 is further extended. FSM2 not only connects NMOS-based currents to the 1 bit ADC, i.e. to the sign detector, but also enables writing in the CALDAC registers. The latter extension is essential for the calibration of the binary currents.

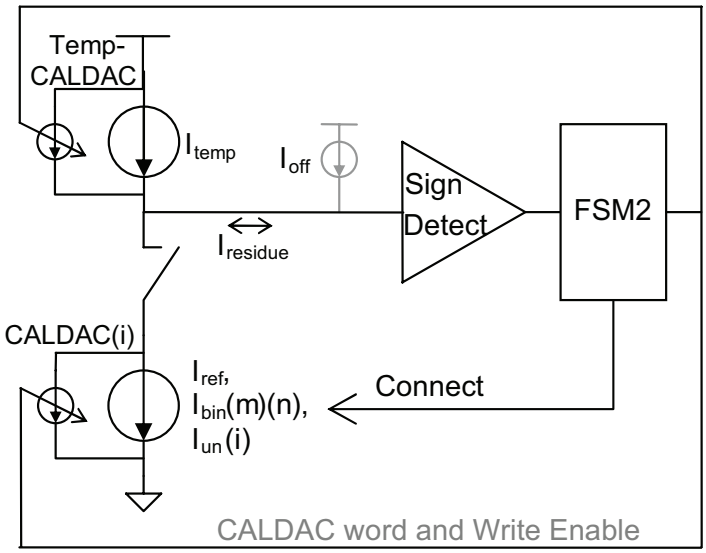


Figure 10.4-1. Proposed new self-calibration scheme.

The method accurately divides the PMOS-based reference, in order to calibrate for the mismatch errors in the nominally different binary currents. This division process requires at least one extra set of all binary currents, shown as $I_{bin}(m)(n)$ in Figure 10.4-1, with *m* being the set of binary currents (at least 2 should exist) and *n* is the binary bit. Extra sets of binary bits can be realized in either of the two following ways. Either through the correction level of segmentation (via setting two CUEs for each binary current) or through the functional level of segmentation (via defining two parallel D/A functions), see Figure 9.2-1 for the abstraction levels of DAC segmentation.

The proposed new calibration algorithm builds on the advantages of the algorithm presented in section 10.2. These include system level cancellation of the unavoidable input offset I_{off} of the 1bit ADC and optimization of the post-calibration current spread through control of the calibration quantization errors. The new algorithm adds one important feature: calibration of the binary currents. For an exemplary DAC based on two sets of binary currents with bit 6 being the MSB binary bit, the calibration equations of the new algorithm are:

Equation 10.4-1

$$1: I_b(1)(6) + \sum_{i=1}^{5} I_b(1)(i) + 1LSB := I_{temp}$$

$$2: I_b(2)(6) + \sum_{i=1}^{5} I_b(1)(i) + 1LSB := I_{temp}$$

$$3: I_b(1)(6) + I_b(2)(6) := I_{temp}$$

$$4: I_b(1)(5) + \sum_{i=1}^{4} I_b(1)(i) + \sum_{i=6}^{6} I_b(1)(i) + 1LSB := I_{temp}$$

$$5: I_b(2)(5) + \sum_{i=1}^{4} I_b(1)(i) + \sum_{i=6}^{6} I_b(1)(i) + 1LSB := I_{temp}$$

$$6: I_b(1)(5) + I_b(2)(5) + I_b(1)(6) := I_{temp}$$

$$7: \cdots$$

The NMOS currents are given always on the left side of the equations and the PMOS I_{temp} - always on the right side. The currents being adjusted are given in grey. Note that all the equations need nominally equal currents on both sides. Step 1 adjusts one of the two 6th binary bit currents (the MSB binary current), so that the equation is satisfied. Step 2 does the same for the other 6th binary bit. Thus, after step 2, both 6th bit currents, $I_b(1)(6)$ and $I_b(2)(6)$, have the same values. Step 3 simultaneously adjusts both 6th bit currents, so that their value is exactly equal to the half of I_{temp}. Steps 4 through 6 realize a similar routine to calibrate the 5th binary bit currents. Note that the left side of the step 6 equation includes the already adjusted $I_b(1)(6)=0.5I_{temp}$ to balance the left and the right equation sides. Furthermore, the rest of the binary currents are similarly calibrated. Any I_{off} would be transformed to a gain error because all binary current sources are calibrated to a common reference.

The role of I_{temp} is to compensate for I_{off} as shown in chapter 10.2. and to use a reference current I_{ref} that has the same polarity as the calibrated currents. However, it is indeed feasible that I_{ref} substitutes I_{temp}, providing that the range of calibration (full-scale) for all CALDACs includes all possible into account I_{off}.

10.4.2. Conclusions

A new start-up calibration method is described. The method is applicable for binary scaled currents. It is based on a 1bit ADC, i.e. a sign detector, and small calibrating CALDACs attached to every signal current source. All current sources are calibrated to a common PMOS-based reference. To calibrate for the nominally different binary currents, the reference is accurately divided. Ultimately, the post-calibration accuracy depends only on the calibration step. That is why the DAC post- and pre-calibration accuracy and the

resolution and LSB step of the CALDACs form simple linear trade-off relationships. This new method will lead to much smaller physical sizes and higher resolution D/A Converters.

10.5. DISCUSSION

According to the classification of chapter 7, this is a method that: belongs to the ETFC (error-transfer function correction) group; uses self-measurement; uses extrinsic redundancy, and is a low-level method. Therefore, analog error measurement must be implemented. It provides exact a-posteriori error knowledge, but creates risks of performance deterioration due to the measurements circuits. Furthermore, the results rely on the accuracy of the measurement; extra hardware resources are required; possible complications for IC process migration exist; etc. However, the correction method is available in both background and foreground correction modes, which is an important design freedom (see Table 7.3-1). The presented method uses extrinsic resources for correction, which may in some case jeopardize the intrinsic DAC performance, especially at high speeds. The extrinsic resources increase the overall complexity of the system, too. However, extrinsic design resources guarantee that correction is always found, since it is extrinsically generated (see Table 7.4-1). Furthermore, the presented method does not feature high dependence on the DAC architecture, as other low-level methods (see Table 7.5-1), since its independence is guaranteed by the approach presented in section 10.3. and the algorithm presented in section 10.4. .

10.6. CONCLUSIONS

New methods for current self-calibration are presented. They are based on simple circuits that do not require any special accuracy requirements. For self-measurements, a 1 bit ADC (current comparator or current sign detector) is used. For self-correction, small CALDACs (calibrating DACs) are attached to every current source for calibration. Algorithms are presented that can calibrate both unary and binary currents. The algorithms for the self-calibration method are implemented with simple FSMs. They compensate for the errors of the self-measurement circuits and optimize the DAC post-calibration accuracy. In section 10.2. a simple 8-state algorithm is presented for calibration of unary currents (nominally identical currents). In section 10.3. , a method is presented to calibrate binary currents using an algorithm for nominally identical currents. In section 10.4. an algorithm is presented for a direct calibration of binary currents. Being able to calibrate all currents decouples the correction method from the DAC architecture. The overall DAC static accuracy depends on a design parameter, the size of the calibration step; it does not depend on the parameters of the CMOS process, which paves the way for technology independent designs.

11. Chapter

NEW REDUNDANT DECODER CONCEPT

The following text describes a new architecture for binary-to-thermometer decoders used in segmented D/A converters to decode the input binary bits into unary bits. To efficiently improve basic DAC characteristics, the architecture features redundant output unary codes. The main concept offers two modes of operation. Each mode generates a different unary output, i.e. a different switching sequence for the DAC (MSB) unary analog elements. The correlation between both switching sequences is low due to the “geometrical” properties of the decoder. This results in two different transfer characteristics of the whole DAC for the same mismatch errors of its elements. After on-chip or off-chip measurements, the more linear transfer characteristic can be selected. Alternatively, the two switching sequences can be dynamically averaged, similarly to DEM correction methods. In either way, chip yield is improved and the design requirements can be relaxed. Ultimately, the advantages introduced by the proposed decoder will lead to cheaper and smaller D/A Converters.

11.1. INTRODUCTION

Yield is an important concern for the chip industry. Low yield makes the manufactured chips more expensive, because the production costs for the "bad" devices are calculated in the price of the "good" devices. Production tolerances, which can result in random mismatch and gradient errors, are among the main reasons for lower yields in analog and mixed-signal devices [46], [1], [10]. For DACs, yield can be defined for their static performance, which is the percentage of devices that meet defined INL_{max} and DNL_{max} specifications. The random mismatch of the analog elements is a main error contributor to DAC INL and DNL, providing that any systematic and gradient errors are minimized through careful layout [49]. Thus, once a chip sample is manufactured, its analog elements have mismatch errors that are randomly distributed and time invariant. The DNL errors can be relaxed at system level by applying thermometer coding for the MSBs. However, the INL errors firmly depend on both the element mismatch errors and their distribution.

Self-calibration techniques can correct for element mismatch, paying with additional resources and affecting in some way the D/A conversion process [56], [71], [36], [35]. An alternative approach to improve yield is to change the distribution of the errors in such a way to improve linearity. Dynamic element matching (DEM) techniques are examples of such an approach [84], [85]. However, these techniques require significant digital resources in terms of both area and power. A static solution that alters the distribution of the mismatch errors is suggested in [1], [86]. Large memory blocks can program (re-map) the thermometer switching sequence but at a price of doubling silicon area of the DAC.

This chapter shows that redundancy in the binary-to-thermometer decoder can also be achieved with little additional resources. Although this little investment may not offer the freedom of [1], [86], statistically it pays back with substantial yield improvement for the whole D/A Converter thanks to the low correlation of the available redundancy. The latter makes the proposed solution much more area and power efficient. The proposed solution improves chip yield and DAC static linearity. It does not deteriorate in any way the dynamic DAC performance.

Based on the background information of section 11.2. , section 11.3. presents the proposed decoder architecture with 2 built-in switching sequences. Section 11.4. presents simulation results for different levels of resolution and segmentation. Finally, conclusions are drawn.

11.2. CONVENTIONAL ROW-COLUMN DECODER

In 1985, Miki proposed the Row-Column Decoder [87], see Figure 11.2-1. Later, a few variations followed, e.g. [88]. Today, it is already a well accepted solution for decoding the MSB binary bits into thermometer (unary) bits in the segmented DACs, e.g. [89], [35], [46].

The binary-to-thermometer decoding is realized in two decoding steps. Firstly, after being split into two groups, the binary bits are decoded into two intermediate sequential thermometer (thermo) codes: the row thermo codes (MSBs, R_j) and the column thermo codes (LSBs, C_i). Secondly, both intermediate thermo codes are applied to a decoding array to generate the final thermo code.

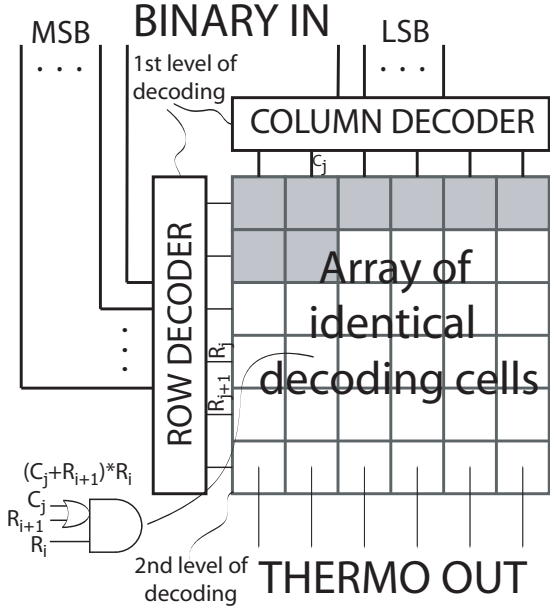


Figure 11.2-1. Conventional Row-Column decoder, from [87].

In the decoding array, three types of rows may be defined. The first row type has all its cells turned "ON" (Row 1 (top) in Figure 11.2-1). The second row type has some of its cells turned "ON" and the others turned "OFF" (Row 2). The third row type has all its cells turned "OFF" (Rows 3,4,5,6). A decoding cell T_{ij} of the i-th row and the j-th column realizes either of the following functions:

Equation 11.2-1

$$T_{ij} = (C_j \cdot R_i) + R_{i+1}, \quad \text{or}$$

Equation 11.2-2

$$T_{ij} = (C_j + R_{i+1}) \cdot R_i$$

Simple ANDOR or ORAND gates can implement respectively Equation 11.2-1 as in [88] or Equation 11.2-2 as in [87]. This decoder architecture is simple and area efficient.

11.3. NEW DECODER WITH REDUNDANCY

Though highly structured, the Row-Column decoder is not symmetrical. Although the first level decoders use identical hardware, they connect differently to the second level decoding ORAND cell. The ORAND cell gets two bits of information from the Rows (MSB part) but only one bit of information from the Columns (LSB). Therefore, the rows must always be used with the MSBs and the columns with the LSBs. That is why the standard Row-Column decoder can only generate one thermometer output. However, hardware symmetry can be implemented efficiently by way of two small alterations, shown in Figure 11.3-1.

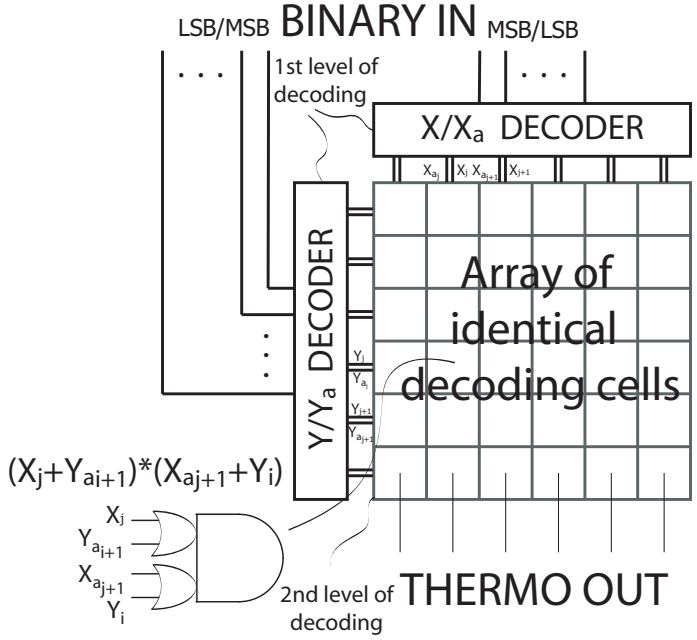


Figure 11.3-1. New decoder with two built-in switching sequences.

Firstly, an ORORAND decoding cell is used, instead of ORAND gate. This ORORAND gate has inputs that are balanced between the rows and the columns. Symmetrical decoding cells allow two operation modes, i.e. two switching sequences instead of only one. Secondly, extra control signals need to be embedded in the first decoding level to set the operation mode. As the symmetry makes the Row and Column decoders equivalent, it is preferred to refer to them as X and Y decoders. The logic equation of the second level decoding cell is changed from Equation 11.2-1 to:

Equation 11.3-1

$$T_{ij} = (X_j + Ya_{i+1}) \cdot (Xa_{j+1} + Y_i),$$

$$with \begin{cases} Mode\ \ 1: Ya = Y, \quad Xa = 0; \\ Mode\ \ 2: Ya = 0, \quad Xa = X. \end{cases}$$

The information for the mode of operation is provided by the Xa and Ya decoders. In Mode 1, all thermo Ya signals are equal to Y signals, and all thermo Xa signals are zero. The symmetrical decoder operates as a Row-Column Decoder with X_j being C_j and Y_i being R_i. In Mode 2, all thermo Ya signals are zero and all thermo Xa signals are equal to X. The symmetrical decoder operates as a Column-Row Decoder with X_j being R_i and Y_i being C_j. Thus, the symmetrical ORORAND gate is transformed to an ORAND gate with different connectivity in the two modes of operation, see Figure 11.3-2.

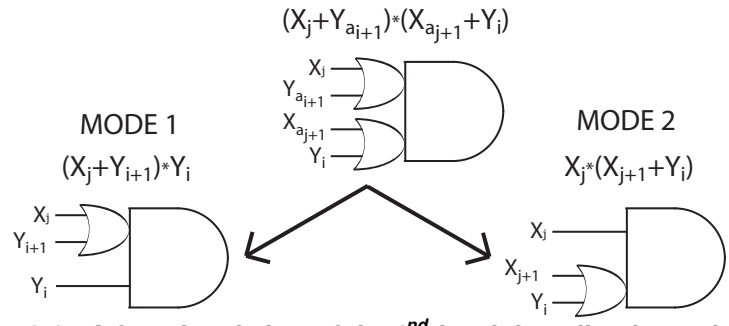


Figure 11.3-2. A functional view of the 2nd level decoding in mode 1 and 2.

Mode 1 and Mode 2 generate different unary switching sequences. Figure 11.3-3 shows a geometrical example of the two switching sequences for a 6-to-63 binary-to-unary decoder. This decoder can be referred to as redundant, because it offers two output switching sequences and a choice between these is possible. In mode 1 the unary code develops horizontally, while in mode 2 the unary code develops vertically. According to the geometrical illustration of Figure 11.3-3, both switching sequences are orthogonal. The correlation between them is low and hence the potential improvement is high.

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

a)

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

b)

Figure 11.3-3. Two different switching sequences of a 6-to-63 redundant binary-to-unary decoder.

The two different switching sequences determine two different distributions of the mismatch errors of the DAC MSB unary elements. For one set of mismatch errors, Figure 11.3-4 shows an example of two DAC INL characteristics for a typical 12b DAC using a 6-to-63 decoder with built-in redundancy. The repetition of the binary errors remains unchanged, while the effect of the dominant thermometer errors is changed.

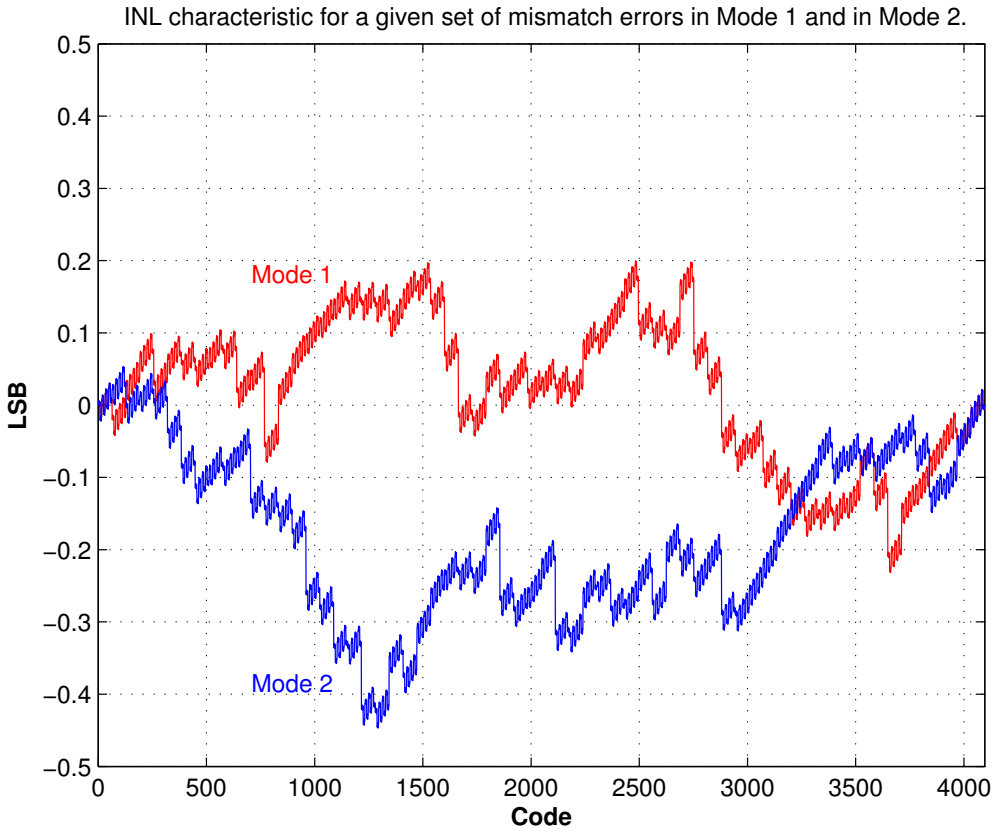


Figure 11.3-4. Simulated INL characteristic of a 12 bit 6-6 segmented DAC for both switching sequences of the redundant decoder.

11.4. SIMULATION RESULTS

To verify, quantify, and demonstrate the improvement in DAC performance when the binary-to-thermometer decoder has redundancy, a high-level static DAC model is simulated. The random mismatch of the DAC unit elements is modeled as independent random Gaussian functions. The resulting maximal INL (INL_{max})error is investigated as a function of the standard deviation of the unit elements.

Figure 11.4-1, Figure 11.4-2 and Figure 11.4-3 shows the distribution of INL_{max} for a 12 bit DAC that is segmented into 6 binary LSBs and 63 thermometer MSBs. This distribution resulted from a Monte Carlo simulation with 1000 statistical runs. The relative mismatch of the unit elements is $\sigma_u / I_u = 0.6\%$, (with I_u - the unit element and σ_u - its standard deviation). The distributions for Mode 1 and Mode 2 are statistically identical in which 7% of the samples do not meet the standard $\pm 1/2\,LSB$ specification for INL_{max}. However, if the built-in redundancy is considered, i.e. the better operation mode of the two Mode 1 and 2 is selected for one DAC sample, then the “bad” devices are reduced to less than 1%!

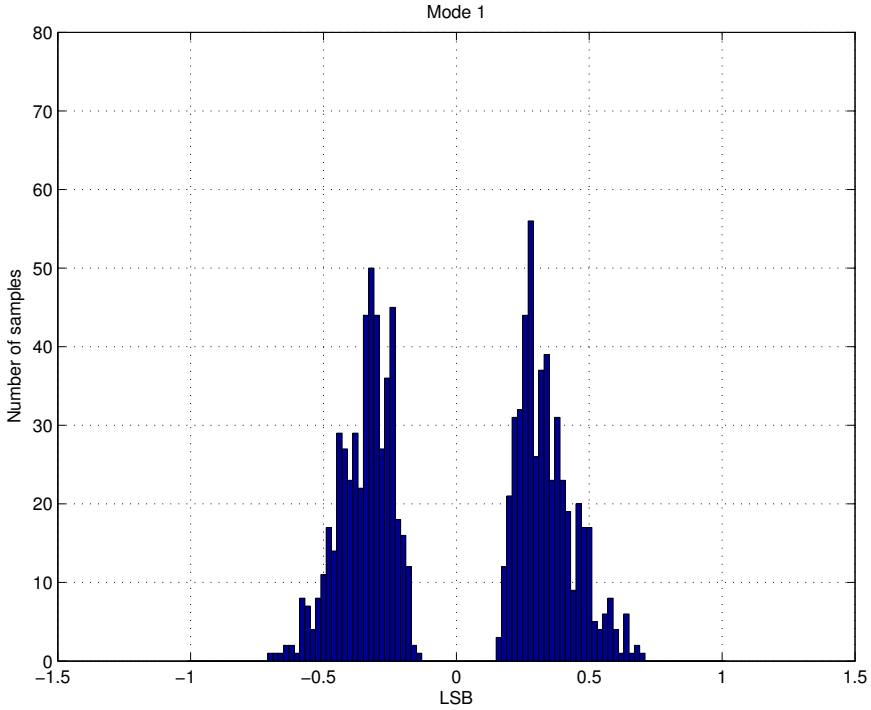


Figure 11.4-1. Simulated distribution of INL_{max} for the redundant decoder, Mode 1.

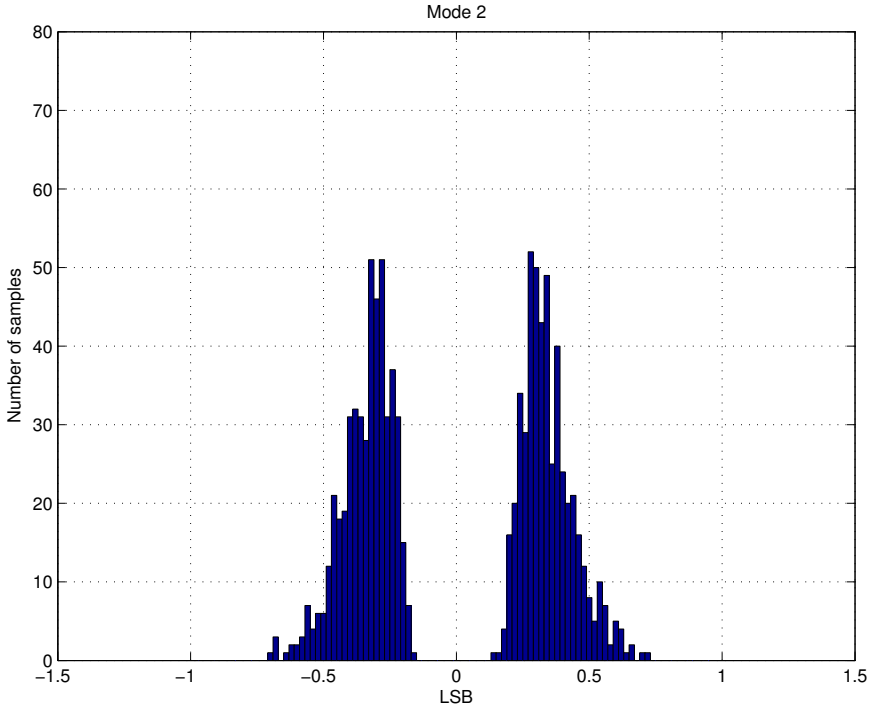


Figure 11.4-2. Simulated distribution of INL_{max} for the redundant decoder, Mode 2.

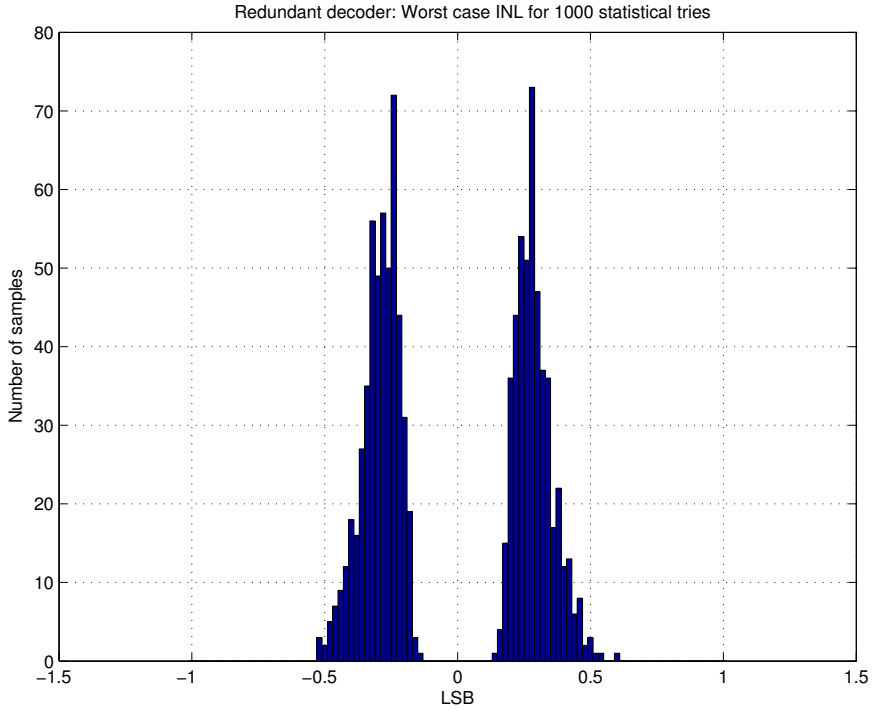


Figure 11.4-3. Simulated distribution of INL_{max} for the redundant decoder, the best of the two Modes.

For $INL < 1/2LSB$ specification, Figure 11.4-4 shows simulation results of DAC yield as a function of the relative mismatch σ/I of the unit elements. The built-in redundancy improves the yield by about 15% on average. This improvement can be also traded for relaxed DAC design requirements. For example, for a yield target of 99.7%, the DAC with a redundant decoder requires $\sigma_u / I_u = 0.56\%$, while the conventional approach requires $\sigma_u / I_u = 0.36\%$. *Thus, the design requirements for the unit elements are relaxed by about 30%.* Therefore, the DACs with a redundant decoder may have up to 2.5 times smaller unit current source transistors, according to the Pelgrom's accuracy-area relationship [6].

The advantages of the redundant decoder can be considered from another point view – the improvement of the yield. When a target yield is specified, the redundant decoder allows using a DAC design featuring lower yield. The difference between the target yield and the intrinsic yield represents the yield improvement that the redundant decoder can provide. For 1 million Monte Carlo simulations, Table 11.4-1 and Table 11.4-2 show the percentage improvement in the yield for a range of resolutions and segmentations. The yield barrier (target yield for |INLmax|<0.5LSB) for Table 11.4-1 is 95% and for Table 11.4-2 is 97%. The shaded boxes can be correspond with the simulation results of Figure 11.4-4. The yield improvement is greater for a larger number of thermometer bits and hardly depends on the number of the binary LSBs. The unaffected binary bits have less influence on the output of the DAC. Note that for higher yield barriers, the advantages of the decoder with built-in redundancy decrease.

Table 11.4-1. DAC yield simulated improvement in [%] for 95% |INLmax|<0.5LSB.

Total bits	Bits of segmentation								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
4	4.9	8.8	13.4						
5	4.8	8.5	13.1	15.0					
6	4.8	8.3	12.8	14.7	15.9				
7	4.5	8.6	12.7	14.8	16.0	16.9			
8	4.6	8.6	13.0	14.5	16.5	16.2	17.2		
9	4.8	8.4	12.7	14.0	16.2	16.1	17.0	16.8	
10	4.6	8.5	12.9	14.4	15.8	16.3	16.4	16.9	17.1
11	4.7	8.3	12.6	15.1	15.9	16.9	16.3	16.9	17.3
12	4.4	8.1	12.7	14.7	16.2	15.9	17.0	17.3	18.0
13	4.9	8.1	12.8	14.5	15.7	15.7	17.0	17.0	17.0
14	4.7	8.8	12.5	14.7	16.0	17.1	17.2	16.8	16.8

Table 11.4-2. DAC yield simulated improvement in [%] for 97% |INLmax|<0.5LSB.

Total bits	Bits of segmentation								
	2	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
4	3.5	6.5	11.6						
5	3.6	6.6	10.6	11.8					
6	3.4	6.5	10.1	11.8	13.1				
7	3.4	6.4	9.9	11.2	13.2	13.7			
8	3.3	6.2	9.9	12.1	12.8	13.6	14.2		
9	3.4	6.7	10.5	12.0	13.2	13.5	13.5	13.6	
10	3.3	6.4	10.1	11.1	12.9	13.8	13.7	14.0	14.6
11	3.3	6.3	10.1	12.1	12.6	13.6	13.7	14.4	14.0
12	3.4	6.0	10.1	11.7	12.1	14.0	13.7	13.7	14.1
13	3.4	6.5	9.8	11.8	12.5	13.9	13.6	13.8	13.9
14	3.6	6.5	9.9	12.0	13.1	13.5	14.0	14.5	14.1

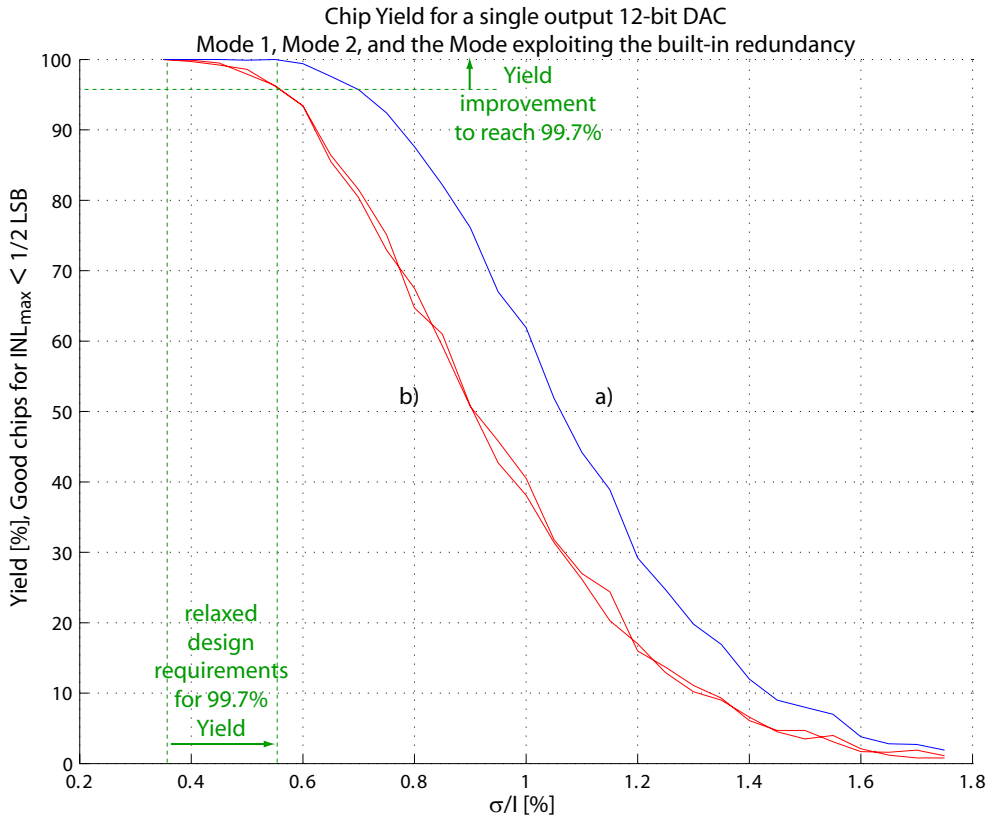


Figure 11.4-4. Yield as a function of the accuracy of the unit elements: a) using the built-in redundancy; b) Mode1 and Mode 2.

11.5. DISCUSSION

According to the classification of chapter 7, this is a method that belongs to the ETFC (error-transfer function correction) group; uses self-measurement; uses intrinsic redundancy, and is a low-level method. Therefore, analog error measurement must be implemented. It provides exact a-posteriori error knowledge, but creates risks of performance deterioration due to the measurements circuits. Furthermore, the results rely on the accuracy of the measurement, either on-chip or off-chip. In the case of on-chip measurements, extra hardware resources are required and possible complications for IC process migration exist. In the case of off-chip measurements, there is a potential increase in the overall test costs.

As a correction method, the decoder can be used both in background methods (in DEM) and in foreground methods (in mapping), which is an important design freedom (see Table 7.3-1). The presented method uses DAC core own resources for correction, which may be insufficient in some cases (Table 7.4-1), especially when they are limited, like in the presented case. Finally, the presented method features increased dependence on DAC architecture, i.e. the intrinsic DAC core cannot be designed independently of the correction method, (see Table 7.5-1).

11.6. CONCLUSIONS

A new binary-to-thermometer decoder with built-in redundancy for segmented and unary DACs is presented. Instead of the conventional single unary (thermometer)

switching sequence, the proposed decoder can generate two different switching sequences with very low correlation. This built-in redundancy affects the whole DAC by allowing two different transfer characteristics. Simulations show that the consequent improvement of the chip yield can reach up to 18% and the design requirements for the accuracy of the DAC currents can be relaxed by about 30%. The improvement is achieved for little redundant hardware resources, which makes the proposed decoder architecture very area and power efficient.

Taking into account the classification of chapter 7, it can be suggested that the proposed decoder can be used either with a mapping method (with self-measurement and static map) or with a DEM method (without self-measurement and dynamically alternating maps). Ultimately, the proposed decoder can lead to improved chip yield and relaxed design requirements. Therefore, the proposed redundant decoder can reduce the overall price of segmented DACs of all types: resistor-based, switched capacitors, or current-steering type, using segmentation at two levels: functional and algorithmic, since it is a generic concept.

12. Chapter

NEW HIGH-LEVEL MAPPING CONCEPT

This chapter proposes a high-level mapping correction method. This high-level mapping correction method uses a functional level DAC intrinsic redundancy, as defined in Figure 9.2-1., to decouple the error correction from the DAC architecture (at algorithmic level). When multiple parallel sub-DACs implement a common D/A function, there is an intrinsic redundancy hidden in the way how each of the sub-DACs contributes to the common output. The presented method uses such functional redundancy to reduce the DAC errors.

It is shown that a highly linear DAC performance can be achieved, while the intrinsic DAC cores are designed with highly relaxed matching specifications. A simulated linearity of almost 16bit is demonstrated, while the initial data from measurements of the intrinsic performance of four parallel sub-DACs shows only 9 bit linearity.

12.1. INTRODUCTION

So far in the literature, the mapping technique is studied only for the unary MSB part of the segmented DAC, because the unary coding inherently features redundancy at algorithmic level [89], [86], [90]. Since the binary coding features no redundancy at algorithmic level, the mapping techniques cannot be used. However, the proposed DAC functional segmentation (chapter 9) introduces functional redundancy at a higher functional sub-DAC level. Therefore, the mapping technique can be used independently of the architecture (algorithmic segmentation) of the sub-DAC unit. It can be effective even when the sub-DACs are implemented with fully binary architectures, yielding a very area-efficient design.

Chapter 7 shows that the mapping correction methods are similar to the DEM correction methods. Both methods use similar circuits for correction. However, the former methods use exact a-posteriori error information to select which errors can compensate each other, while the latter methods use randomization to average the errors in time. In [60], a DEM using two sub-DACs is proposed. The classification of chapter 7 suggests that a similar high-level mapping method should exist. Instead of randomization however, an a-posteriori error information can be used.

Thus, the error correction method proposed in this chapter is as follows. Firstly, the errors of the sub-DACs are measured. Then, an appropriate combination between the contributions of the sub-DACs to the common DAC output is found. That is to say that the sub-DAC errors substantially compensate each other for every code of the common D/A transfer function.

Since the DAC correction is independent of the sub-DACs algorithmic segmentation, both can be optimized independently. For example, this method can be applied to DACs with binary algorithmic segmentation, leading to very efficient and linear designs that inherently feature reduced dependence on the manufacturing IC technology (see Table 7.5-1.), as in the segmented DEM of [60]. However, the proposed high level method does not increase the noise floor, since the errors are mutually compensated and not randomized. In addition, the proposed high level method uses less power and hardware than the segmented DEM, since no dynamic randomization is needed.

12.2. CONCEPTUAL IDEA

The DAC system from Figure 4.3-1 can be constructed using multiple parallel sub-DACs, as shown in Figure 12.2-1 for M parallel current-steering (CS) sub-DACs. The sub-DACs are defined as independent D/A sub-functions using functional segmentation of the main system D/A function, as described in chapter 9. Each of the D/A sub-functions can be considered as a separate DAC entity but when they are used together, functional-level redundancy is added to the overall DAC system. The digital input word $w(nT)$ can be converted to an analog signal I_{out} (not shown) just as in a conventional DAC. However, there is an extra degree of freedom in choosing the sub-DAC input words $w_i(nT),\ i=1,2,\cdots M$, due to the functional intrinsic redundancy of the architecture.

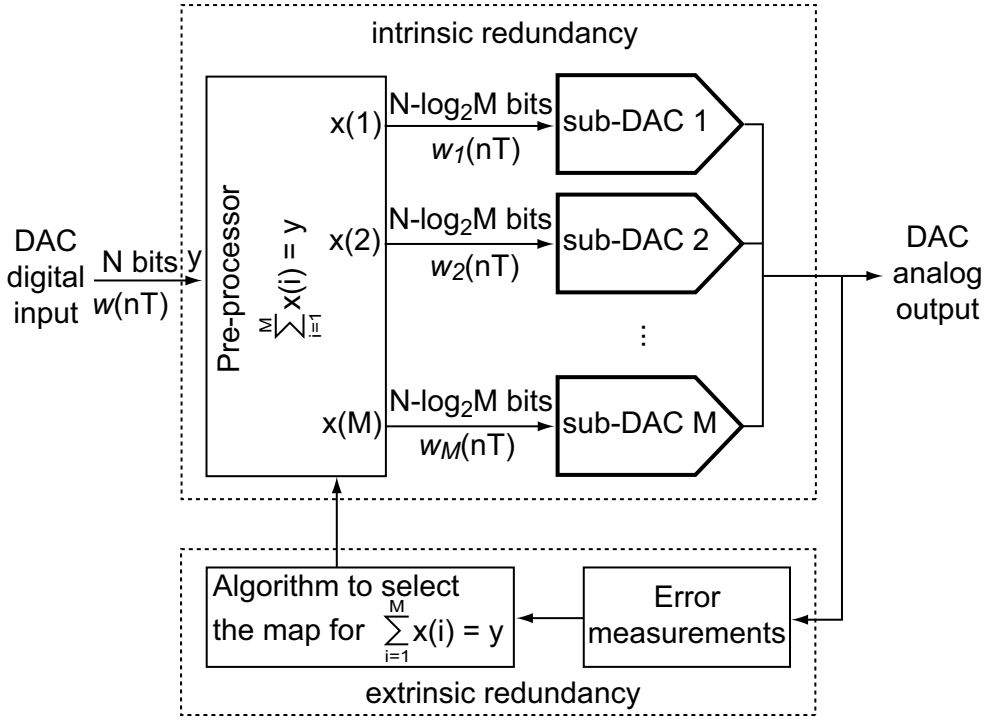


Figure 12.2-1. A conceptual diagram of the high-level mapping method with M parallel CS sub-DACs.

The digital pre-processor distributes $w(nT)$ among the sub-DACs. The sum of all sub-DAC words $w_i(nT)$ should equal $w(nT)$:

Equation 12.2-1

$$w(nT) = w_1(nT) + w_2(nT) + \cdots + w_M(nT)$$

The output signal current is a sum of the sub-DAC output currents. The static and dynamic errors of these currents $e_i(t), i=1,2,\ldots M$ also sum up and contribute to the output. Therefore, the output of the DAC can be considered as having an ideal part $i_{out}(t)$ (contributing to STF, as defined in chapter 7) and an error part $e(t)$ (contributing to ETF, as defined in chapter 7). Then, the DAC output can be written as:

Equation 12.2-2

$$i_{out}(t)+e(t)=i_{out1}(t)+e_1(t)+\cdots+i_{outM}(t)+e_M(t)$$

The mapping correction method minimizes $e(t)$ by means of mutual error compensation. Once the information of the particular current errors is available, the digital pre-processor can be programmed to distribute $w(nT)$ in such a way that all $e_i(t)$ sum to minimum. Thus, the mapping criterion is:

Equation 12.2-3

$$\min[e(nT)]=\min[e_1(nT)+e_2(nT)+\cdots+e_M(nT)]$$

The number of available combinations is very large. These correspond to the number of possible ways C_k to spread the input word $w(nT)$ among the sub-DACs via $w_i(nT)$. Figure 12.2-2 shows the number of possible combinations C_k for the 512 codes full-scale (FS) range of $w(nT)$ in the cases of 2, 3, and 4 sub-DACs.

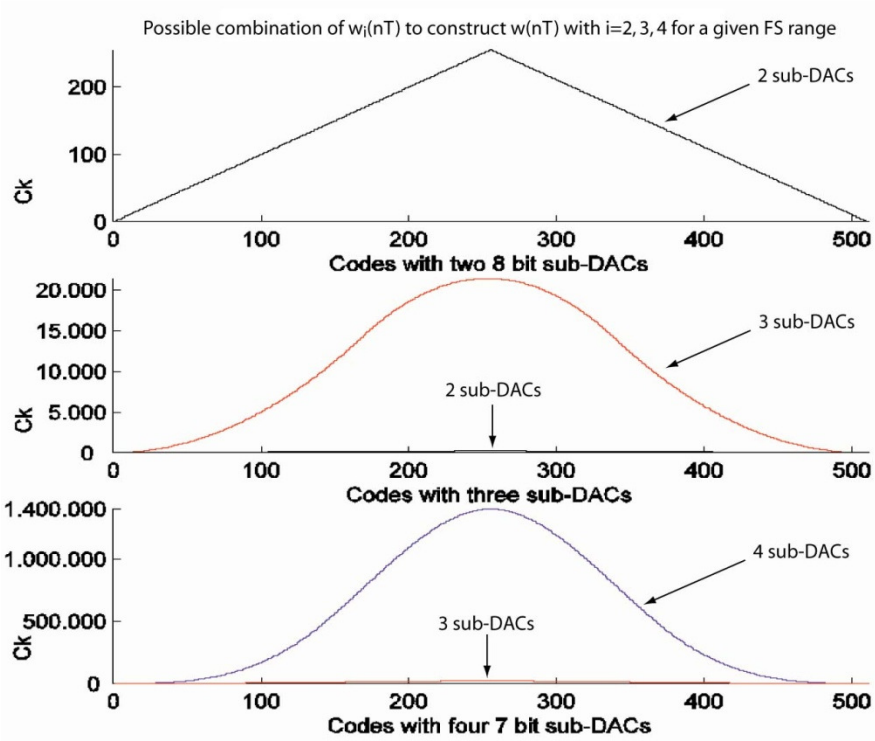


Figure 12.2-2. Redundancy in generating w(nT), shown as combinations of $w_i(nT)$ for 512 codes.

Practically, except for the initial and last codes of the DAC transfer characteristic, the number of possible combinations between $w_i(nT)$ to construct $w(nT)$ is very large. That is to say that the intrinsic redundancy in the system is very large. This redundancy can be used for performance improvement.

Note from Table 7.5-1 that a high-level DEM method already exists from an independent research work [60] that validates with measurements the performance improvement. In this work [60], two parallel sub-DACs are used and the different

combinations of $w_i(nT)$ to generate $w(nT)=\sum_{i=1}^{2} w_i(nT)$ is randomized. However, the proposed method of high-level mapping uses exact knowledge about the sub-DAC errors (amplitude and timing) to realize Equation 12.2-3 and hence chooses one static combination to generate $w(nT)=\sum_{i=1}^{M} w_i(nT)$ that features the smallest errors. A compromise should be made between selecting a map to compensate for either amplitude or timing errors. Thus, selecting the optimal map is frequency dependent.

12.3. ILLUSTRATIVE MEASUREMENT AND SIMULATION RESULTS FOR AMPLITUDE ERRORS MAPPING

To validate in practice the high-level mapping technique, a DAC system with four parallel 12 bit sub-DACs, sub-DAC A, B, C and D, is designed and measured. Figure 12.3-1 shows the layout of the DAC system, occupying area of about 0.8mm^2, which is in the order of the other published DAC designs. Thus, using multiple parallel sub-DACs does not necessarily require an increase of the DAC overall area. The parallel sub-DACs can be considered as a way to functionally segment the available analog resources, similarly to the algorithmic binary, unary, and segmented DAC architectures, discussed in chapter 9. More about this test-chip implementation is provided further in the thesis, chapters 15, 16, 17 and 18.

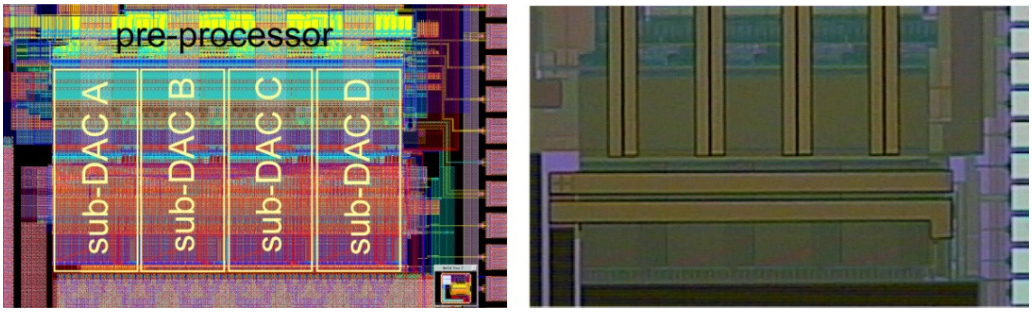


Figure 12.3-1. Layout and micrograph of the 180nm CMOS test chip DAC with four 12 bit sub-DACs.

The sub-DACs are designed for 9 bit static accuracy. The measured intrinsic INL of the sub-DACs is shown in Figure 12.3-2. The INL_{max} of the sub-DACs is just exceeding 2 LSB at 12 bit level. The sub-DACs are combined and the mapping method is applied to form a 14bit DAC system.

The algorithm searching for an optimal map is implemented in MATLAB. For practical convergence reasons, it searches only within a limited set of maps, so that its convergence is within about a minute. Figure 12.3-3 shows the simulated DAC INL and DNL after applying a high-level map to distribute the sub-DAC input words, starting from the measured results of the sub-DAC intrinsic linearity. The achieved INL_{max} does not exceed 0.5LSB at 14bit level, while the output accuracy for a great majority of the codes is within 0.125LSB (16b accuracy). As it can be seen from Figure 12.3-3, the glitch of 0.5LSB is at the very beginning of the INL characteristic, where not as many combinations are available as in the middle of the INL characteristic. Note that most of the errors in the middle scale are due to the applied map searching algorithm, which searches only within a limited set of maps. A much better than 16bit linearity can be

achieved for the mid-scale codes with an exhaustive search algorithm, which is beyond the scope of this thesis.

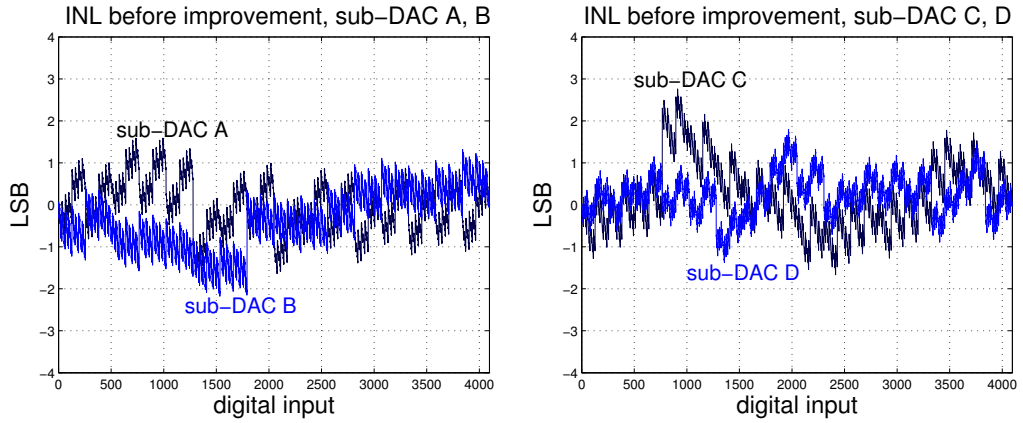


Figure 12.3-2. Measured intrinsic linearity: 12 bit INL of sub-DACs A, B, C, and D.

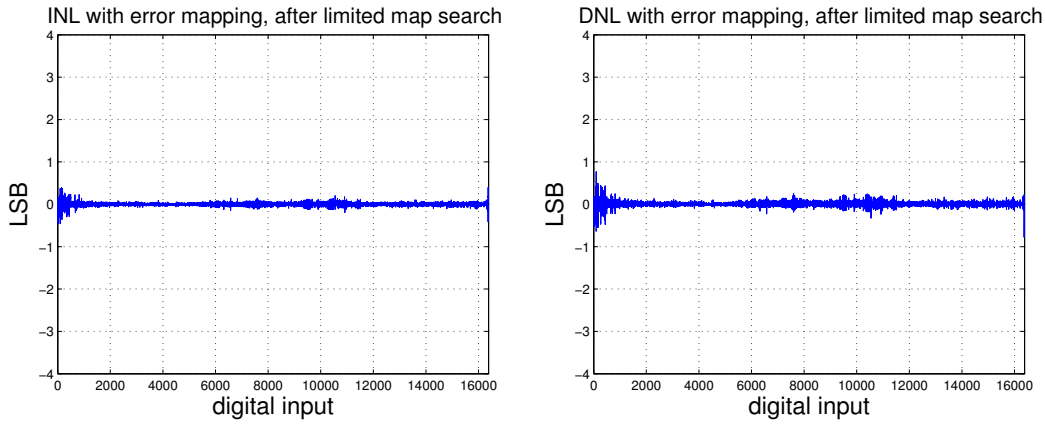


Figure 12.3-3. Simulated 14 bit op-mode DAC INL (left) and DNL (right) after errors compensation by means of mapping the distribution of the sub-DAC words, based on initial data from Figure 12.3-2.

12.4. LIMITATIONS AND DISCUSSION

Similarly to the other mapping methods, only one map can be used at a time. The choice of this map cannot cover the optimal solutions for both static and dynamic DAC performance, respectively optimizing the amplitude and the timing errors. Therefore, a compromise should be made.

Another limitation is the needed digital resources. To implement in hardware the mapping memory for the measurement example, a memory block of $\left(2^N-1\right)\times P=\left(2^{14}-1\right)\times\left(4\times12\right)\approx768Kb$ (from Figure 4.3-1 and the general discussion thereafter), which can be optimized (e.g. one of the sub-DAC input words can be mathematically derived from the others) to about 500Kb. Potentially, more efficient implementations can be found if the number of available maps is limited.

The high-level mapping technique can achieve very high linearity. However, there are two main challenges that should be addressed. These are the size of the MAP block and algorithm to choose the best map. Both challenges are due to the very high number of possible ways to distribute the input word $w(nT)$ among the sub-DAC words $w_i(nT), \quad with \quad i \in [1,2,...,M]$. Possible solutions include using only a limited set of MAPs, which however may not feature sufficient mutual error compensation. Another solution is the DAC co-integration within a digital system that can provide the needed memory for the mapping block and implement sophisticated map searching algorithms. Such a digital platform may be for example the FPGA.

According to the classification of chapter 7, this is a method that belongs to the ETFC (error-transfer function correction) group; uses self-measurement; uses intrinsic redundancy, and is a high-level method. Therefore, analog error measurement must be implemented. It provides exact a-posteriori error knowledge, but creates risks of performance deterioration due to the measurements circuits. Furthermore, the results rely on the accuracy of the measurement; extra hardware resources are required; possible complications for IC process migration exist. However, the correction method is available in both background and foreground correction modes, which is an important design freedom (see Table 7.3-1). The presented method uses DAC core own resources for correction, which may turn insufficient in some cases. However, the DAC core intrinsic performance is preserved, since no extrinsic circuits can jeopardize it (Table 7.4-1). Finally, the presented method features reduced dependence on DAC architecture, i.e. the intrinsic DAC core can be designed independently of the correction method, (see Table 7.5-1).

12.5. CONCLUSIONS

A DAC high-level mapping correction method is proposed. This method uses functional segmentation to define multiple sub-DACs. Then, the sub-DACs are used to compensate each others' errors. This method is independent of the algorithmic segmentation of the sub-DACs. Therefore, it can be used with binary sub-DACs, leading to very efficient but linear design. The correction is applied to the whole DAC transfer characteristic, as opposed to the conventional low-level unary approaches that apply correction only to the DAC MSB unary part.

The synthesis of this method is an example how the analysis of chapter 7 can be used to help developing new DAC correction methods. Based on the clues of Table 7.5-1, the method for segmented DEM of [60], and the low-level mapping of unary currents, the proposed method can be synthesized.

Simulation results based on measured initial data show that higher than 16 bit linearity can be achieved. Providing sufficient digital resources to realize the map searching algorithms and the actual map, highly linear DAC designs can be implemented.

13. Chapter

NEW HARMONIC-DISTORTION-SUPPRESSION METHOD

This section presents a method to improve the linearity of current steering (CS) digital-to-analog converters (DACs). The method uses unary functional segmentation to create redundant D/A functions, defined around nominally identical sub-DAC entities. Further, the method uses a digital pre-processor to create phase shifted replicas of the input signal that are fed to the sub-DACs. These represent the input arguments for the redundant D/A functions. The parallel sub-DAC branches convert the input signal replicas to currents. The output currents are combined and their harmonic distortion components mutually suppress each other. In this way, the overall DAC spurious-free-dynamic-range (SFDR) and total-harmonic-distortion (THD) are improved. The method can be applied with either a single conventional CS DAC, based on emulated virtual parallel sub-DACs, or a CS DAC with unary functional segmentation, i.e. based on physically present parallel sub-DACs. The method is particularly suitable for OFDM communication systems and mixed-signal systems-on-a-chip with DACs, e.g. FPGAs with integrated DACs. Recently, a similar method has been developed for the RF field, in an independent research project [61], which further advocates in favor for applying this approach in the field of DACs.

13.1. INTRODUCTION

Linearity is among the most important specifications of the contemporary current-steering digital-to-analog converters (CS DACs). For example, communication applications need linear circuits in order to distinguish between the wanted signals and the unwanted interferers. A few different approaches for achieving and guaranteeing the specified DAC linear performance are popular in the literature, e.g. intrinsic design [89], current sources calibration [35], switching time adjustment [90], etc (see Table 7.2-1). A common characteristic of the existing approaches is that they target a specific error mechanism. These design approaches are customized to particular DAC applications, specifications, and manufacturing IC (integrated circuit) technology. That is why the redesign and migration to other IC technologies may involve considerable effort, time, and risks. To relax such drawbacks and more importantly to improve further the DAC performance, this section proposes a flexible method to suppress the DAC harmonic distortion (HD) spurs via parallel phase-shifted signal processing. The method relies on matching the error mechanisms to mutually suppress their effects. It is particularly suitable for OFDM (orthogonal frequency-division multiplexing) communication systems, where the realization of phase-shifted signal replicas is simple. The signal pre-processing of the method is robust, predictable and technology independent, because it is done completely in the digital domain. In addition, the method is flexible to counteract a large variety of errors, e.g. due to digital switching. These two main characteristics make the method particularly suitable for System-On-a-Chip (SoC) integrated DACs (e.g. in FPGAs (field programmable gate arrays)).

Section 13.2. describes the theoretical background for the proposed method. Section 13.3. discusses important aspects of the implementation of the method. Finally, conclusions are drawn in section 13.5. .

13.2. THEORETICAL BACKGROUND

The DAC HD spurs for an input sinewave are a product of the mix between the DAC non-linear transfer characteristic, the input signal frequency f_{in} and the sampling frequency F_s. If F_s and f_{in} are known, the frequencies of the DAC HD components can be predicted:

Equation 13.2-1

$$f_{HD_k^p} = kf_{in} \pm pF_s ,$$

where *p* is the *p-th* image band, and *k* is the *k-th* harmonic component. Let a DAC system use unary functional segmentation to define multiple D/A function (sub-DACs). Let the multiple sub-DACs work in parallel, as shown in Figure 13.2-1. Let the pre-processor add particular phase delays $\varphi_{in}^{(m)}$, with $m \in [1,...,M]$, to the digital input signal of all sub-DACs. Then, for an input frequency $\omega_{in} = 2\pi f_{in}$ and sampling frequency $\omega_s = 2\pi F_s$, these sub-DAC outputs are:

Equation 13.2-2

$$I_m(j\omega) = \sum_{p=-\infty}^{\infty} \sum_{k=1}^{\infty} A_{k,p}^{(m)} e^{j\left((k\omega_{in}+p\omega_s)t+p\varphi_s^{(m)}\right)} e^{jk\varphi_{in}^{(m)}} ,$$

where $I_m(j\omega)$ is the output of the m^{-th} sub-DAC in the complex domain, $A_{k,p}^{(m)}$ is the amplitude of the k^{-th} harmonic component from the p^{-th} image band of the m^{-th} sub-DAC, $\varphi_s^{(m)}$ is the phase of the sampling signal of the m^{-th} sub-DAC . When M sub-DACs operate in parallel, their combined output can be approximated as the superposition of their individual outputs. Let the M sub-DACs be synchronized, i.e. $\varphi_s^{(m)} = 0$ for all m, and let the M sub-DACs be identical, i.e. $A_{k,p}^{(m)} = A_{k,p}$ for all m. The combined output $I_{out}(j\omega)$ of the M parallel sub-DACs is:

Equation 13.2-3

$$I_{out}(j\omega) = \sum_{m=1}^{M} I_m(j\omega) = \sum_{p=-\infty}^{\infty} \sum_{k=1}^{\infty} A_{k,p} e^{j(k\omega_{in}+p\omega_s)t} \sum_{m=1}^{M} e^{jk\varphi_{in}^{(m)}},$$

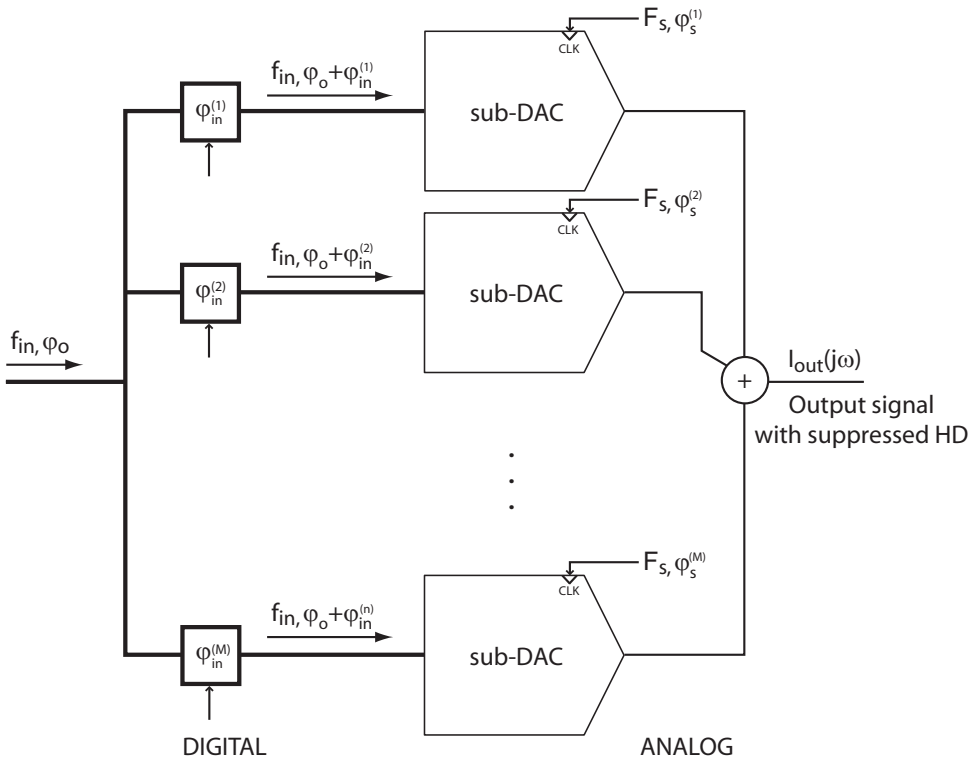


Figure 13.2-1. A sub-DAC-based system, processing phase-shifted replicas of the input signals.

A proper combination of $k\varphi_{in}^{(m)}$ can minimize the factor $\sum_{m=1}^{M} e^{jk\varphi_{in}^{(m)}}$ and hence suppress the overall (i.e. at the output) $A_{k,p}^{(1,2,\ldots,m)}$ for all p. That is to say that for a particular combination between the given sub-DAC non-linearity, i.e. the amplitudes $A_{k,p}^{(m)}$ and the

input-signal phase shift, i.e. $\varphi_{in}^{(m)}$, the DAC output HD spurs can be suppressed through mutual compensation.

For example, suppression of HD_3 can be achieved with two nominally identical parallel DACs, i.e. $A_{k,p}^{(1)} = A_{k,p}^{(2)} = A_{k,p}$, converting phase shifted input signals by $\pi/3$, i.e. $\varphi_{in}^{(1)} = \varphi_{in}^{(2)} - \pi/3$, and combining their outputs to reduce the superposition of the 3rd harmonic amplitude $A_{3,p}^{(1,2)}$ for all image bands *p*:

Equation 13.2-4

$$A_{3,p}^{(1,2)} = A_{3,p}\left| e^{j(0\cdot\pi)} + e^{j\left(3\cdot\frac{\pi}{3}\right)} \right| = 0$$

The cancellation is valid for all *p*, i.e. for all image bands. Figure 13.2-2 visualizes the process by means of phazor diagrams:

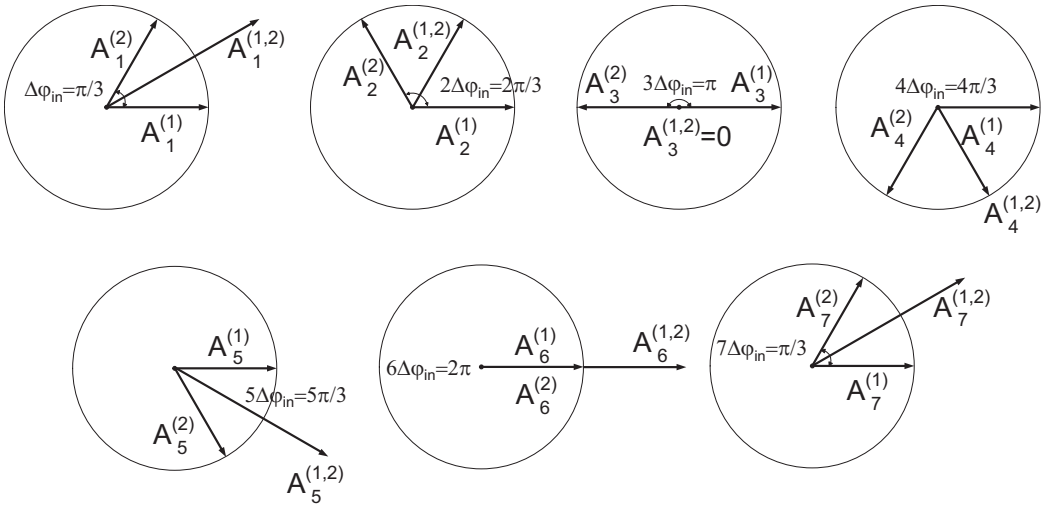


Figure 13.2-2. Phazor diagrams for the main signal (A0) and the amplitudes of the 2nd (A2) and 3rd (A3) harmonic tones for two sub-DACs processing phase-shifted input signals.

In theory, $A_{3,p}^{(1,2)}$ is completely cancelled, $A_{1,p}^{(1,2)}$, $A_{5,p}^{(1,2)}$, $A_{6,p}^{(1,2)}$, $A_{7,p}^{(1,2)}$ are increased, and $A_{2,p}^{(1,2)}$, $A_{4,p}^{(1,2)}$ remain the same. Relatively to the change of the main single $A_1^{(1,2)}$, the HD components can be considered. $HD_{3,p}^{(1,2)}$ is completely cancelled, $HD_{2,p}^{(1,2)}$ and $HD_{4,p}^{(1,2)}$ are reduced, $HD_{5,p}^{(1,2)}$ and $HD_{7,p}^{(1,2)}$ remain the same, and $HD_{6,p}^{(1,2)}$ is increased. Concerning $HD_{6,p}^{(1,2)}$, if its intrinsic power is low, the increase may not be significant.

In practice, due to the sub-DAC differences $A_{k,p}^{(1)} \neq A_{k,p}^{(2)}$ and error mechanisms located beyond the summation point, $HD_{3,p}^{(1,2)}$ is only reduced. Examples of these error mechanisms include the mismatch errors and the finite output impedance of the current switching cells.

13.3. APPLICATION AREA

In the broader electronics literature, the principles of phase different parallel signal processing are applied in different fields, e.g. RF circuits [91], [92]. However, from theory to applications at least three important aspects should be carefully considered. These are the realization of the phase-shifted replicas, the implementation of the parallel sub-DACs, and the choice of a target candidate application.

13.3.1. Phase-shifters

A major common difficulty in the reported methods is the realization of the input signal phase delays. For example, in the RF electronics domain [91], [92], extra modulation and demodulation operations are used in order to create the phase-different signal replicas. Despite the added complexity and linearity requirements, a reported harmonic reduction of up to 50dB for a targeted harmonic spur is achieved. However, the steps of modulation in the digital domain and demodulation in the analog domain (using the φ_s factor from Equation 13.2-2) would pose too heavy burden in the DAC design requirements to become practical.

However, accurate phase-shift filtering of the input DAC digital signals is practically feasible, alleviating the need of modulation and demodulation. The main possible implementations include signal phase-shifting through filtering for only real input signals and through complex multiplication for complex input signals. A real domain phase-shift filter should realize the following transfer function:

Equation 13.3-1

$$H(f) = e^{j\varphi_{in} sign(f)}$$

The transfer function can be directly synthesized into a practical FIR filter. The size of the filter determines the quality of the phase-shift. Figure 13.3-1 compares the angles of phase-shift for 16, 64, and 256 taps FIR filters. Smaller FIR filters use less digital resources but limit the useful bandwidth and add fluctuations in both the gain and the phase-shift. Bigger FIR filters deliver better performance but consume more resources.

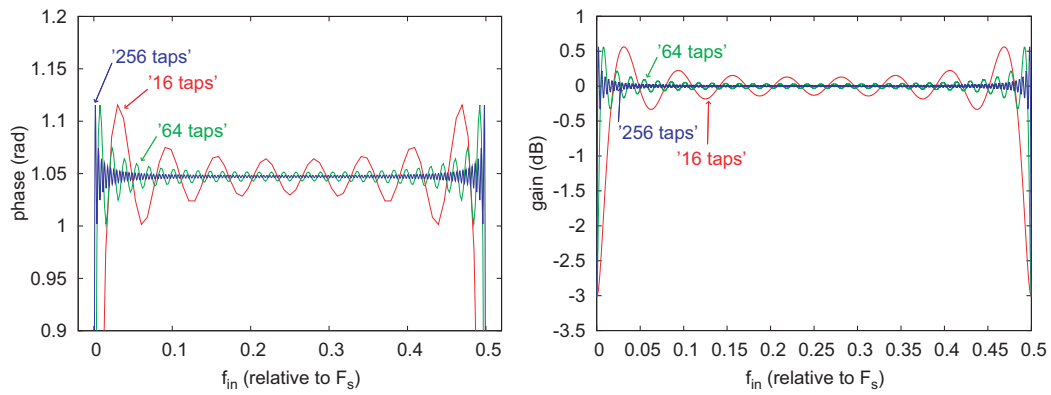


Figure 13.3-1. Phase and amplitude response of a digital phase-shifting FIR filter.

Note that two FIR filter taps require one digital multiplier and two memory steps (e.g. flip-flops). That is why a FIR filter based implementation is useful for systems with sufficient digital resources, e.g. FPGAs in the case of co-integrated DACs.

A more efficient implementation of the digital phase-shift can be achieved in the complex domain through a single complex multiplication of the input signal by the factor

$e^{j\phi}$, when the input signal is available in a complex form. Some of the exemplary applications are OFDM communication systems, signal generators, since these applications have the input signal available in a complex form. A single digital complex multiplication requires 4 (real) multiplications and 2 (real) additions blocks, which can easily be transformed to 3 multiplications and 5 additions blocks. For the sake of the discussed method, only the real part of the result can be used and so the operations can be reduced to the efficient 2 multiplications and 3 additions.

13.3.2. Parallel sub-DACs

The method requires parallel DACs for mutual non-linearity suppression. The parallel DACs can be either virtually emulated or physically implemented.

Standard DACs can emulate two or more virtual parallel sub-DACs by trading their signal bandwidth. For example, to emulate two input-signal phase-shifted sub-DACs, the input digital signal can be composed by time-interleaving the codes of two phase-shifted signals. The resulting DAC output contains the analog representation of the two digital signals in a time interleaved way, i.e. $\varphi_s^{(1)} = \varphi_s^{(2)} + \pi$ (φ_s is the phase of the clock signal for both sub-DACs).

To avoid the reduction of the useful bandwidth, the overall DAC architecture can be specifically based on parallel sub-DAC units, via e.g. unary functional segmentation (see section 9). The parallel sub-DACs are physically present and synchronized, e.g. $\varphi_s^{(1)} = \varphi_s^{(2)}$. Thus, the method can be implemented without loss of signal bandwidth. However, the parallel sub-DACs should be matched as much as practically possible, i.e. $A_{k,p}^{(i)} \approx A_{k,p}^{(j)}$, so that their non-linearity mutually cancel as much as possible. A practical parallel sub-DACs based design is proposed in [64] and this thesis.

13.3.3. Candidate applications

This method is a good candidate for signal generation applications or band-limited applications. The efficiency of the suggested method is increased when it is used in suitable mixed-signal applications. Notable examples include co-integration with FPGA systems, signal generators, OFDM communication systems. Modern FPGA systems offer a large amount of digital resources that can be used for pre-processing. Furthermore, convenient places for DAC integration are in the I/O (input/output) FPGA blocks. Thus, the individual DAC would practically be in parallel and hence the presented method can be easily implemented. The signal generation applications require pure and clean signals, while the target signals are a priori known. Using this knowledge, the DAC SFDR can be boosted via a proper choice of the parameters of Equation 13.2-3. In OFDM communication systems, the information is coded in the frequency domain. The frequency bins are generated in the time domain by the transmitter system. In the process of generation, the time domain signals are available in a complex form. The generation of the phase-shifted replicas is relaxed because only a complex multiplication is required.

13.4. LIMITATIONS AND DISCUSSION

The method for suppression of HD in the form presented in Figure 13.2-1 with real parallel sub-DACs can cancel HD components that are intrinsic to the sub-DACs. However, HD components that are intrinsic to the whole system cannot be canceled. For example, the effects of the finite output impedance of the DAC current cells generate HD. The associated error mechanism originates from the voltage drop of the DAC output

signal which modulates the currents of the DAC current cells, due to their unavoidably limited output impedance. Since the DAC output signal is the super position of the contributions of the sub-DAC outputs, the generated HD components are common for the parallel sub-DACs and they are always in-phase.

A possible solution may be considered in RZ (e.g. [27]) techniques for the parallel sub-DACs. They would decouple the parallel sub-DACs in time and hence the modulation due to the DAC output voltage variations for the parallel sub-DACs would be decoupled in time, too. However, the parallel sub-DACs would have phase-shifted sampling moments that would create problems with the phase-shifts in the odd image bands, as discussed later in section 17.3. .

According to the classification of chapter 7, this is a method that: belongs to the STFC (signal-transfer function compensation) group; uses no self-measurement; uses intrinsic redundancy, and is a high-level method. Therefore, no exact knowledge on the errors is used and hence the method relies on matching of the errors. The method is available in background mode only, which may increase the power-consumption at high speeds (see Table 7.3-1). It uses DAC core own resources for correction, which may turn insufficient in some cases. However, the DAC core intrinsic performance is preserved, since no extrinsic circuits can jeopardize it (Table 7.4-1). Finally, the presented method features reduced dependence on DAC architecture, i.e. the intrinsic DAC core can be designed independently of the correction method, (see Table 7.5-1).

13.5. CONCLUSIONS

A new method is presented that improves the linearity of CS DACs. The method is based on parallel sub-DACs, which can be either emulated or created via unary functional segmentation of the DAC analog resources. Further, the method uses digital pre-processing to create phase-shifted replicas of the digital input signal. After the transformation of these replicas into analog signals by the sub-DACs, the intrinsic DAC HD is mutually suppressed and the overall DAC dynamic range is improved. The presented method is particularly suitable for OFDM communication applications, where the realization of the input signal phase delays is simple. The method features digital pre-processing and mutual non-linearity self-suppression, which are suitable for future CMOS technologies and relax the general technology migration.

14. Chapter

FLEXIBLE DIGITAL-TO-ANALOG CONVERTERS CONCEPT

This chapter proposes a new concept for flexibility of Digital-to-Analog Converters (DACs). This concept is realized in a new flexible current-steering DAC architecture. The architecture is based on a modular design approach that uses unary functional segmentation (i.e. parallel sub-DAC units) to realize flexible design, functionality and performance. The architecture includes circuit redundancy, benefiting from the reduced cost per transistor that comes with CMOS technology scaling. The parallel sub-DAC units form a mixed-signal platform that is capable of a number of DAC correction methods, including calibration, error mapping, data reshuffling, and harmonic distortion cancellation. The proposed new DAC flexibility concept is expected to assist FPGA digital platforms in their continual expansion into the mixed-signal domain.

14.1. INTRODUCTION

Flexibility is a concept introduced to adapt electronic systems to their ever changing environment which includes the new market challenges, the application requirements and the IC technologies. Nowadays, the market for electronics is mature and highly competitive, where only the first, the best, and the most innovative and efficient companies survive. At the same time, the life cycle of new application standards is getting short and the engineers must run long races in short time to meet the new targets. The advantages of new IC technologies can help them only after having successfully addressed the various disadvantages. In particular for analog design, these include problems with the reduced design voltage headroom and hence deteriorated transistor matching; problems with the required silicon area for the analog blocks when compared to the scaled-down area of the digital blocks. A viable concept to address the challenges of the ever changing environment of electronics is *flexibility*.

Flexibility is already a very successful concept in the area of digital electronics. The programmable FPGA market is continually expanding and naturally it should cross into the mixed-signal domain. The SoC (System-On-Chip) co-integration between FPGAs and mixed-signal interfaces will introduce the programmability advantages into the mixed-signal domain, e.g. decoupling the application from the silicon design, reducing time-to-market, and bringing application tailored performance at minimum cost. The flexible DAC/ADC extends the functionality of the FPGA into the mixed signal domain, while the FPGA provides a lot of digital assistance for their performance. This is a synergy between the FPGA and the flexible DAC/ADC interfaces that encourages their on-chip co-integration.

This chapter proposes a new flexible architecture for current-steering DACs. Its main principle suggests introducing DAC redundancy at system and functional levels. In practice, this principle is translated into using parallel low-resolution sub-DAC units as building blocks of a high resolution flexible DAC platform. A digital pre-processor distributes the digital input among the sub-DACs, in a way controlled by the operation mode (op-mode) of the DAC platform. The sub-DAC outputs are combined in the analog post-processor block, which in its simplest form is just a summation of currents.

Further, section 14.2. presents the flexible DAC platform concept. Section 14.3. makes some basic definitions for flexibility with respect to the proposed DAC platform. Section 14.4. discusses various operation modes (op-modes) that illustrate the flexibility of the DAC platform. The op-modes are considered in two groups: general flexibility concepts (section 14.4.1.) and special op-modes that are enhanced with DAC correction methods (section 14.4.2.). Section 14.5. considers an inherent problem of the DAC platform and its possible solutions. Finally, conclusions are drawn in section 14.6.

14.2. FLEXIBLE DAC PLATFORM

Figure 14.2-1 shows a diagram of the flexible DAC concept. The main features of this DAC platform include *flexible design*, *flexible functionality*, and *flexible performance*. At design level, engineers have the flexibility of a modular design approach. At functional level, customers can configure the platform to operate as either a single or multiple independent DACs. At performance level, customers can set the DAC resolution, power consumption, and realize trade-offs between DAC speed and output linearity.

The basic building block of the flexible DAC platform is an N bit sub-DAC. If M sub-DACs are used in parallel, then the overall resolution can range from N to $N+\log_2 M$ bits. Furthermore, this architecture facilitates a number of DAC correction methods,

thanks to its inherent functional segmentation. These correction methods include: complete current sources calibration (see chapter 10), error mapping (see chapters 11 and 12), data reshuffling (see chapters 3.5. and 11) and harmonic distortion cancellation (see chapter 13), etc.

When some sub-DACs are not used for a selected op-mode, their power supply can be switched off, reducing the power consumption for the whole DAC system. This possibility is indicated with the power supply management block.

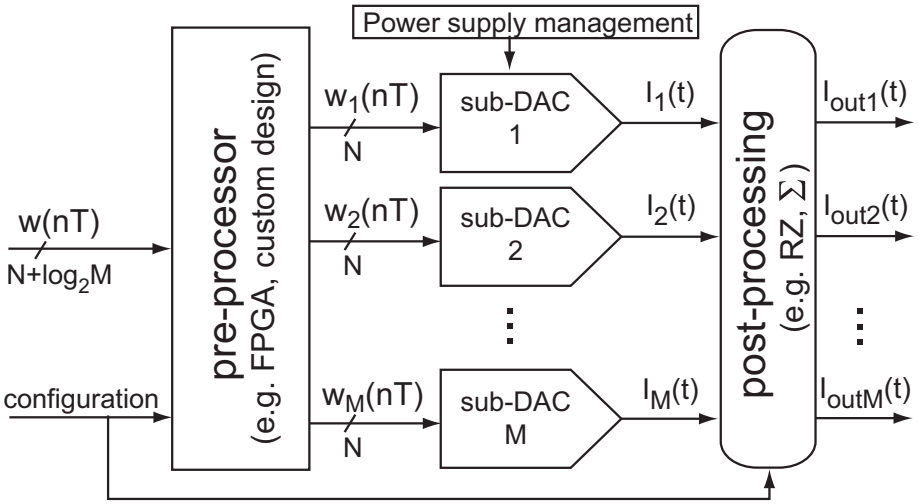


Figure 14.2-1. A flexible DAC architecture based on parallel sub-DAC units.

The flexible DAC architecture includes a data pre-processor in the digital domain. Its main function is to distribute the main DAC digital input word w(nT) among the sub-DAC words $w_1(nT)$, $w_2(nT)$, etc., according to the targeted op-mode. When the flexible DAC architecture is co-integrated in an FPGA, the FPGA flexible digital resources can be used for the pre-processor. Depending on the selected op-mode, the complexity and hence the size of the digital resources are different. When the flexible DAC architecture is not co-integrated in an FPGA, the digital pre-processor is implemented for a selected set of op-modes with a customized design. For example, it may be a simple de-multiplexer that redirects the input digital word to its sub-DAC destination.

The flexible DAC architecture includes an analog post-processor. Its main function is rerouting (e.g. de-multiplexing) of the sub-DAC analog outputs, according to the programmed op-mode. The implementation complexity can vary from a simple summation via a common node wire connection for the current-steering DACs to advanced summation methods, like Return-to-Zero methods for interleaved sub-DACs, e.g. [27].

14.3. DEFINITIONS OF FLEXIBILITY

This section makes basic definitions of the flexibility concept. It demonstrates various aspects of the proposed flexible DAC platform. Section 14.3.1. defines the concepts of hardware flexibility (also called configurability). Different hardware configurations of the DAC platform are presented and discussed. Section 14.3.2. defines the concepts of software flexibility (also called programmability). It is shown how the chosen hardware configurations can be further programmed to customize the DAC performance.

14.3.1. Hardware flexibility: configurability

The hardware resources of the flexible DAC platform can be configured in different ways depending on the chosen operation mode (op-mode). Figure 14.3-1 illustrates an example of how the hardware resources may be configured, starting from the general architecture of the flexible DAC platform.

In the digital domain, the pre-processor is configured to distribute the sub-DAC input words $w_i(nT)$, according to the chosen op-mode. In the mixed-signal domain, some of the sub-DACs are turned on and others are turned off. In the analog domain, the post-processor connects some of the sub-DACs together and activates some of the available outputs.

For the example of Figure 14.3-1 at the left, two parallel DAC functions are realized, each of the two DAC functions is N+1 bits in resolution; a special correction method is applied – the high level mapping method (see chapter 12). Thus, the pre-processor distributes $w_1(nT)$ and $w_2(nT)$, according to the chosen MAP for the first DAC function $I_{out1}(t)$ and $w_3(nT)$ and $w_4(nT)$, according to the chosen MAP for the second DAC function $I_{out2}(t)$. Only four sub-DACs are used in this configuration, so the rest, e.g. sub-DAC 4, 5, …, M, can be switched off to save power. The analog post-processor connects sub-DAC 1 and 2 together to form $I_{out1}(t)$ and sub-DAC 3 and 4 together to form $I_{out2}(t)$.

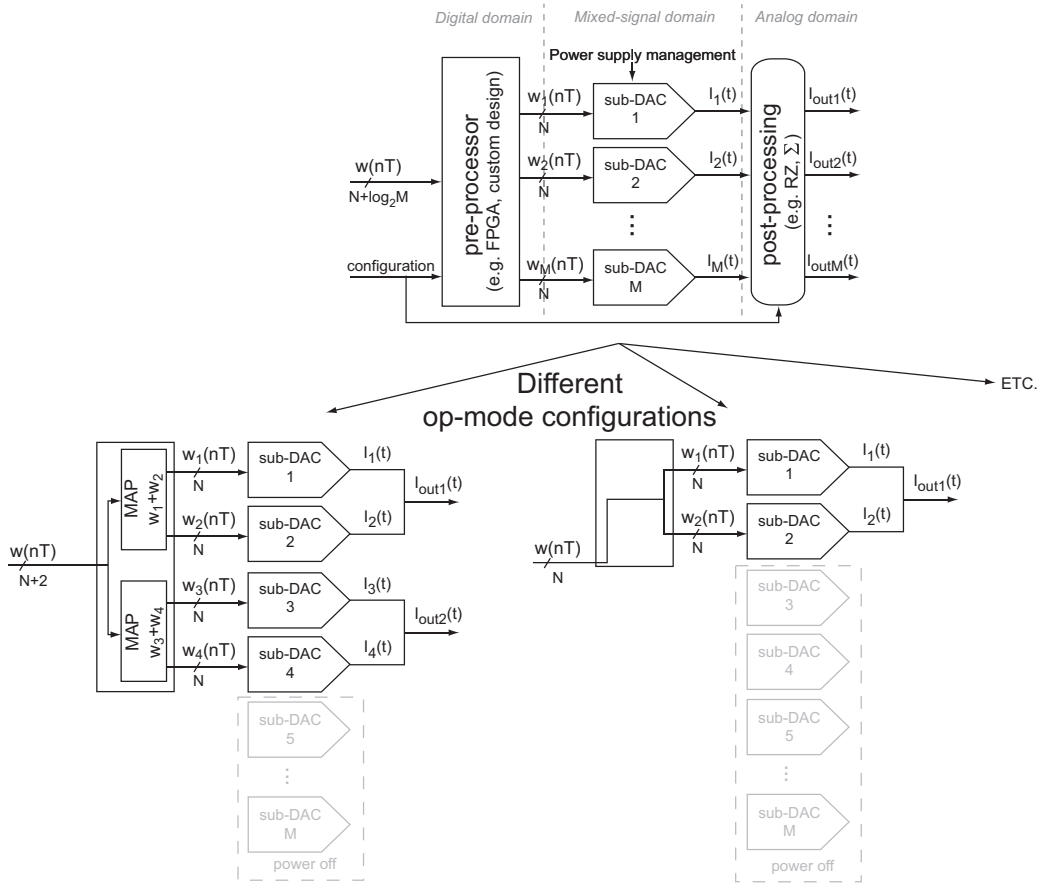


Figure 14.3-1. Examples of different hardware configurations of the FlexDAC op-modes.

For the example of Figure 14.3-1 at the lower right, one DAC function is realized with a resolution of N bits and increased gain and output power by increasing twice the LSB step with respect to the sub-DAC. The pre-processor copies the DAC input word to both $w_1(nT)$ and $w_2(nT)$. Only two sub-DACs are used in this configuration, so the rest, e.g. sub-DAC 4, 5, …, M, can be switched off to save power. The analog post-processor connects sub-DAC 1 and 2 together to form $I_{out1}(t)$.

Table 14.3-1 shows different configuration options of the three main building blocks: the digital pre-processor, the sub-DACs, and the analog post-processor. The first column shows which block is being configured. The second column shows how the selected block can be configured. Finally, the third column shows what different possibilities exist for further customizing a selected configuration.

Table 14.3-1 – Exemplary hardware flexibility via different hardware configurations

Configuration	***Flexibility***	***Possible options***
Pre-processor, *digital domain*	Decoder	Choose the switching sequence of the sub-DACs in “stacked” mode
		Code-interleaved mode
	Map (section 14.4.2.3.)	Map for static performance
		Map for dynamic performance
		An optimized map, for performance within a selected frequency band
	Copy	The input digital word is copied to all sub-DACs for higher gain.
	De-multiplexing	The input digital words contain the digital input words of the sub-DACs, interleaved in time.
	DSP: Phase shifts (section 14.4.2.5.)	Selects a set of phase-shifts for optimal HD suppression
	DSP: Pre-distortion (chapter 4.4.)	Programming particular pre-distortion function to cancel the DAC own distortion.
Sub-DACs, *mixed signal domain*	Number of powered ON and powered OFF sub-DACs (flexible power consumption) (section 14.4.1.4.)	Switching on/off bias currents; Externally switching on/off power supplies; Switching off clock network for the CMOS digital logic
Post-processor, *analog domain*	Number of independent DAC outputs (*available D/A functions*) (section 14.4.1.1.)	Connected sub-DAC output nets; Time interleaved DAC outputs via RZ techniques; Transformer based summation; Switched-capacitor techniques, etc.
	Gain per D/A function (section 14.4.1.2.)	
	Full-scale range per D/A function	

14.3.2. Flexible op-modes (software flexibility): programmability

Once the hardware of the DAC platform is configured for some op-modes, it can be further programmed. This represents another level of flexibility – software flexibility. Figure 14.3-2 shows an example of such programmability for two op-modes with DAC correction methods: high-level mapping and suppression of HD.

For the high-level mapping correction method, the selected maps can be programmed. The customers can select a map for static or dynamic performance (including an optimized performance for a frequency band of interest). For the suppression of HD, the customers can select the phase shifts of the parallel DAC branches, depending on which HD components, or combination of them, are targeted for suppression.

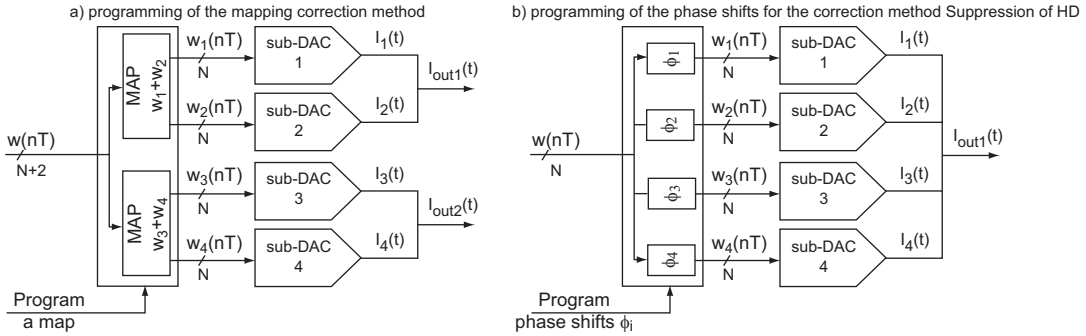


Figure 14.3-2. Examples of programming for two particular hardware configurations.

Table 14.3-2 lists possible programmable aspects for some of the hardware op-modes. The first column shows the hardware block that can be programmed. For example, the digital pre-processor, in an FPGA co-integration, can be configured as a decoder. The sub-DACs are used in “stacked” mode, as illustrated further in Table 14.4-1. In this case, the switching sequence of the sub-DACs can be controlled (switching sequence of the unary cells of the functional segmentation). Other hardware blocks that can be further programmed include high-level mapper (a block the implements the map), various DSP functions, DEM block, calibration engine, etc. Table 14.3-2 shows in the second column which aspects are programmable of the hardware blocks (column 1). The third column shows the targets that can be achieved. For example, the redundant decoder can be used to improve yield and relax design requirements, as described in chapter 11. However, the redundant decoder is used here at functional level. Nevertheless, the same arguments can be used to show that it can still improve performance and reduce design requirements.

Table 14.3-2 – Software flexibility (programmability) for a given hardware configuration

Hardware configuration	***Flexibility/Programmability***	***Options***
Decoder	Switching sequence of the sub-DACs (e.g. redundant decoder, chapter 11)	Select switching sequence (improve yield; relax design)
MAP	Select a map (high level mapping, chapter 12)	Mapping for low speeds, Mapping for high speeds; Mapping for a frequency band.
DSP	Phase shifts (chapter 13), Pre-distortion	Select HD components to suppress
DEM	Algorithm	Randomization; Cyclic/Periodic.
Calibration engine	Calibration algorithm	Unary currents (section 10.2.); Binary currents (section 10.4.); CUE (section 10.3.)

14.4. OPERATION MODES

The parallel sub-DACs, the digital pre-processor, and the analog post-processor (shown in Figure 14.2-1) form a platform that features two classes of original contributions: flexibility and smartness. Some initial aspects of these concepts are discussed in [64], [48] and chapters 9, 10, 11, 12, and 13. The flexibility concepts cover three main DAC aspects: design, functionality and performance. The suitable smartness concepts include DAC correction methods that are enabled by the new functional segmentation (chapters 9), i.e. parallel sub-DACs. This text concentrates on the digital pre-processor and the mixed-signal core that is based on parallel sub-DACs. For the sake of simplicity, the analog post-processing is assumed in its simplest form – a common node connection of the sub-DAC output currents. The realization of sophisticated analog post-processing techniques, such as RZ, time interleaving, is beyond the scope of this thesis.

This chapter considers some general aspects of flexibility in section 14.4.1. These include a flexible number of D/A functions, flexible gain, flexible resolution and static linearity, flexible power consumption, flexible dynamic linearity, etc. Further, the chapter considers some special aspects of flexibility in section 14.4.2. These include self-measurement op-modes, low-level current calibration, high-level mapping, DEM (high-level data reshuffling), harmonic distortion suppression. In section 14.5. a particular characteristic of the DAC platform is discussed, called the "*missing code problem*". Finally, conclusions are drawn in section 14.6.

14.4.1. General flexibility

The flexible architecture can adapt DACs to the new market challenges as discussed in section 14.1. via: design flexibility, D/A function flexibility, and performance flexibility.

The main design target is the low-resolution sub-DAC unit. Once the N bit sub-DAC block is available in the design library, the construction of higher $N+\log_2 M$ resolution DAC platforms by means of M parallel sub-DACs is quick and easy. This design approach is particularly suitable for a co-integration with SoC, where many product families exist with different size divisions and hence different DAC resolutions would be needed. The FPGA platforms are considered as such SoC. This modular design approach represents the design flexibility of the proposed DAC architecture.

The M transfer characteristics of the N bit sub-DACs, expressed as series of the sub-DAC input code k and assuming segmentation with B binary LSB bits, are:

Equation 14.4-1

$$subDAC\ 1: \quad I_{1_k}^{(1)} = \sum_{m=1}^{N-S} B_{k,m} \sum_{i=2^{m-1}}^{2^m-1} I_{u_i}^{(1)} + \sum_{m=1}^{2^S-1} T_{k,m} \sum_{i=2^{N-S}+(m-1)2^{N-S}}^{2^{N-S}+m2^{N-S}-1} I_{u_i}^{(1)},$$

$$subDAC\ 2: \quad I_{k}^{(2)} = \sum_{m=1}^{N-S} B_{k,m} \sum_{i=2^{m-1}}^{2^m-1} I_{u_i}^{(2)} + \sum_{m=1}^{2^S-1} T_{k,m} \sum_{i=2^{N-S}+(m-1)2^{N-S}}^{2^{N-S}+m2^{N-S}-1} I_{u_i}^{(2)},$$

...

$$subDAC\ M: \quad I_{k}^{(M)} = \underbrace{\sum_{m=1}^{N-S} B_{k,m} \sum_{i=2^{m-1}}^{2^m-1} I_{u_i}^{(M)}}_{\text{Binary part}} + \underbrace{\sum_{m=1}^{2^S-1} T_{k,m} \sum_{i=2^{N-S}+(m-1)2^{N-S}}^{2^{N-S}+m2^{N-S}-1} I_{u_i}^{(M)}}_{\text{Unary part}},$$

where $I_k^{(j)}$ is the output of the j-th N bit sub-DAC with $j \in [1:M]$, $i \in [1:2^N-1]$, S are the bits segmented in a unary code at algorithmic level, while $B_{k,m}$ and $T_{k,m}$ are the switching matrices, respectively for the binary and the unary (thermometer) parts. Note that for $S=N$, the equations represent fully unary, at algorithmic level, sub-DAC transfer characteristics and for $S=0$, the equations represents a fully binary, at algorithmic level, sub-DAC transfer characteristic.

With respect to defining the sub-DACs, the unary functional segmentation is only one way of defining the sub-DACs. Thus, the main difference between Equation 9.3-1and Equation 14.4-1 is that the latter represents a general case. For example, the sub-DACs of Equation 14.4-1 do not need to share the same DAC analog resources, as assumed in Equation 9.3-1. DACs, co-integrated in FPGAs, can find their places in the I/O banks of the FPGA, which are around the whole chip. In this case, the sub-DACs are defined physically apart, i.e. they cannot share the same array for the unit currents I_u. Therefore, Equation 14.4-1 assumes an independent array of $I_{u_i}^{(j)}$ for each sub-DAC. Figure 14.4-1 shows the equivalent high level circuit of Equation 14.4-1.

Thus, the functionality of the DAC platform is flexible. It broadens the market niche with respect to conventional DAC (point solutions) architectures, covering more possible applications. The customers can select a particular system level D/A function among the following options:

- Select a single D/A function or multiple independent D/A functions (single or multiple sub-DACs);

- Select the gain of the D/A function(s);
- Select the resolution of the D/A function(s).

The following sections discuss these different possibilities.

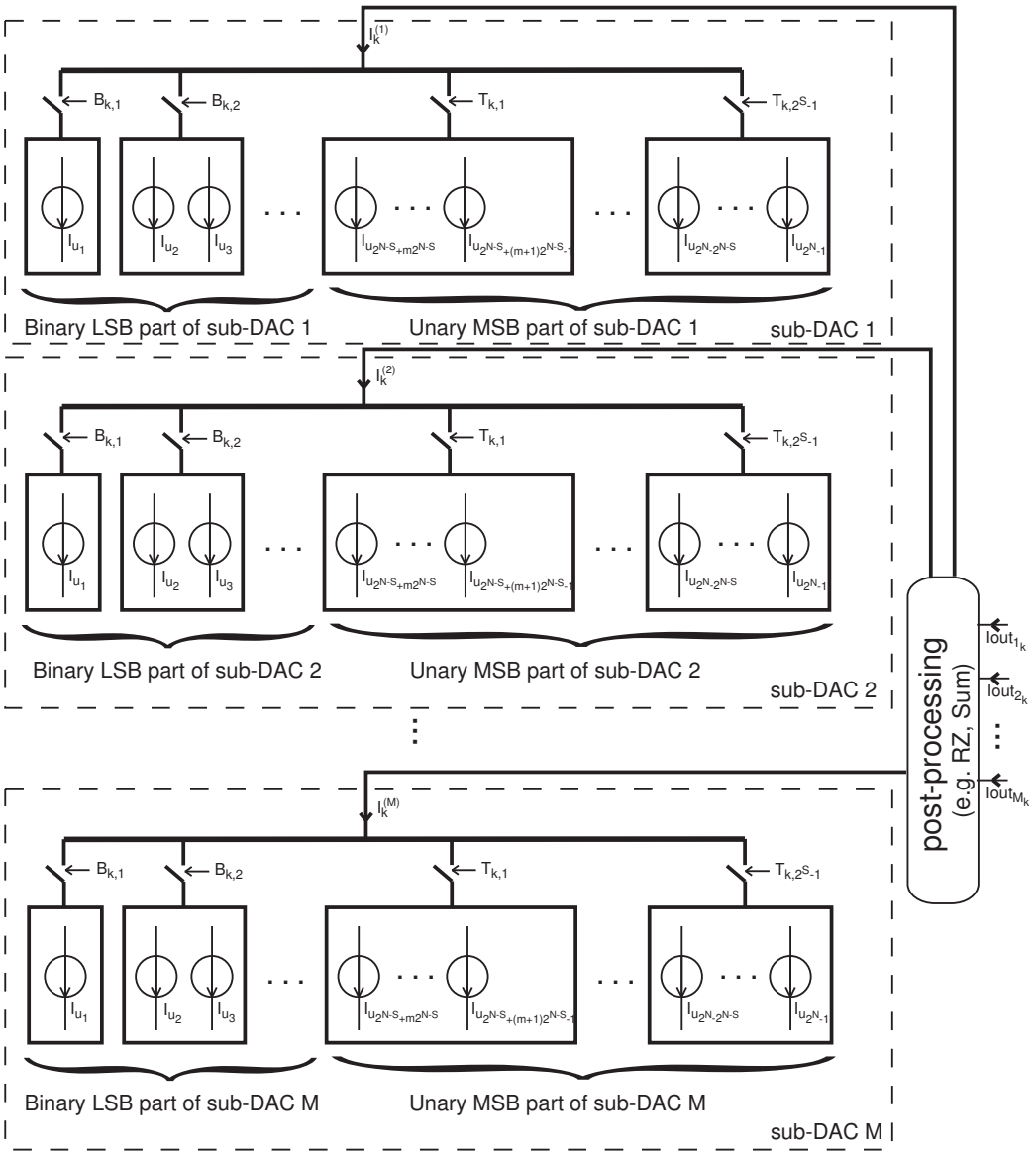


Figure 14.4-1. A flexible CS DAC platform.

14.4.1.1. Single and multiple DACs

The flexible DAC architecture can operate as a single DAC or multiple DACs. For the single DAC op-mode at system level, only one sub-DAC can be selected or multiple sub-DACs can be used in parallel, at functional level. The latter is the opposite process of the functional segmentation, i.e. *functional integration* - multiple D/A functions are combined (merged) to create a single common D/A function. That is to say that the output

of the DAC system/platform in its simplest form is the output of the selected j[-th] sub-DAC $I_{out} = I_j$. However, it is possible to construct the D/A function (, i.e. the "one DAC operation") with multiple parallel sub-DACs, too. The various single-DAC op-modes at system level are shown in Table 14.4-1.

Table 14.4-1. Single DAC op-modes with N bit sub-DACs

Single DAC op-mode	***Distribution of the DAC input word*** $w[nT]$ ***to sub-DAC words*** $w_i[nT]$
N bit single DAC mapped to one N bit sub-DAC: Only one sub-DAC (*j[th]*) is used for the D/A conversion. (The redundancy of multiple sub-DACs can be used to choose the one featuring the best linearity).	$w_j[nT] = w[nT]$, $w_i[nT] =$ N/A, $i \neq j$
N bit single DAC with high gain (g): The parallel sub-DACs convert the same sub-DAC digital input words w_j[nT].	$w_j[nT] = w[nT]$, e.g.: $j = 1, 2, \ldots g$;
N+$\log_2$(m) bit single DAC with large full-scale range and high resolution (option 1, decoder-based: "stacked sub-DACs" op-mode): Sub-DACs convert the DAC input digital codes in stacked (series) way. The sub-DAC outputs are summed to create the DAC output. The digital bits of the DAC input $w[nT]$ are divided into N bits LSB part and $\log_2$(m) bits MSB part. If the input code w[nT]<2^N-1, indicated by the MSB part of w[nT] i.e. $w[nT]\left[N+1:\log_2(m)\right]$, sub-DAC 1 does the conversion of the N bit LSB part i.e. $w[nT][1:N]$, while the other sub-DACs are set to 0. If the input code is 2^N<w[nT]<2(2^N-1), then sub-DAC 1 is set to full-scale, sub-DAC 2 does the conversion and the rest of the sub-DACs are set to 0. ETC. Note that for the sake of simplicity, the "missing code problem" should be taken into account, as assumed in the description above. Chapter 14.5. provides detailed discussion on this problem.	$w_{i<j}[nT] = Full\ Scale$, $w_j[nT] = w[nT][1:N]$, $w_{j>i>m}[nT] = 0$, for $w[nT] > (j-1)(2^N-1)$, $w[nT] < j(2^N-1)$

Single DAC op-mode	***Distribution of the DAC input word*** $w[nT]$ ***to sub-DAC words*** $w_i[nT]$
<u>N+log$_2$(m) bit single DAC with large full-scale range and high resolution (option 2, bits shift based: "code interleaved" sub-DACs op-mode):</u> Sub-DACs convert the DAC input digital codes in interleaved (parallel) way. This method is convenient for number of sub-DACs equal to a power of 2, since the hardware implementation is simple. The input digital word is divided by the number of parallel sub-DACs (division by two is shifting by one the digital bits of the input word). The sub-DAC input words are the division quotient of k/m plus an optional increment from the remainder k/m.	For $w[nT][LSB:MSB]$, e.g. m=2, $w_1[nT][1:N] =$ $= w[nT][2:N+1] +$ $+ w[nT][1];$ $w_2[nT][1:N] =$ $= w[nT][2:N+1];$
<u>Single DAC with a DAC correction method:</u> Sub-DACs convert the sub-DAC digital input words w_1[nT], … w_j[nT], according to a high-level DAC correction method. For example, the high-level DAC correction method can be high-level mapping (chapter 12), high-level input data reshuffling method (DEM) (chapter 3.5. , or method for suppression of HD (chapter 13). The indexes l, m, …, j are set by the applied DAC correction method.	$w[nT] = w_1[nT] + ...$ $... + w_j[nT]$

For the multiple independent DAC outputs, the platform activates the independent outputs I_{out1}, I_{out2}… I_{outM}, shown in Figure 14.4-1. Note that each individual output can be treated as a single sub-DAC and the op-modes shown in Table 14.4-1 can be applied.

14.4.1.2. Flexible DAC gain

The flexible DAC architecture can regulate the gain of the D/A conversion function. There are two main ways to regulate the DAC gain (i.e. to change the LSB step) via:

- increasing the unit current I_u;
- using parallel sub-DACs that convert the same input digital word.

The DAC LSB step amplitude can be regulated via changing the V_{gs} bias voltage of the current source transistors. However, the matching properties of the current source cell change, too, as indicated by Equation 2.3-2. Reducing V_{gs} deteriorates the current matching, while increasing V_{gs} increases $V_{on}= V_{gs}-V_{th}$ of the transistor, which may put it out of the saturation region $V_{ds}< V_{on}$ and reduce the output impedance of the current cell. Therefore, regulating the DAC gain via regulating the amplitude of I_u should be avoided, if possible.

Alternatively, the DAC LSB step amplitude can be increased if multiple LSB currents from multiple sub-DACs work in parallel and join currents. This approach does not change the biasing conditions of the DAC current cells and hence their matching

properties remain unchanged. This approach is indicated in the second row of Table 14.4-1.

14.4.1.3. Flexible DAC resolution and static linearity

The resolution of the whole platform can be changed from *N* bits (when only one sub-DAC is used) to $N+\log_2 M$ (when all sub-DACs are used). There are two ways to change the DAC resolution. These are via:

- changing the DAC full-scale range and keeping the DAC LSB step the same;
- changing the DAC LSB step and keeping the DAC full-scale range the same;

Changing the DAC LSB step should be avoided if possible due to matching problems, as discussed in chapter 14.4.1.2. Thus, increasing the resolution from *N* bits to $N+\log_2 M$ implies either "stacking" (the third row of Table 14.4-1) or interleaving (the fourth row of Table 14.4-1) sub-DACs together. In either case, the full-scale range of the whole DAC system is increased from 2^N-1 codes to $M\times(2^N-1)$ (for sub-DACs with 2^N-1 full-scale ranges). The LSB step of the whole DAC system remains the LSB step of the sub-DAC, i.e. the unit current I_u is not changed. Figure 14.4-2, Figure 14.4-3, Figure 14.4-4, and Figure 14.4-5 illustrate with INL and DNL DAC plots how the resolution can be increased in the "stacking" sub-DACs mode.

Note that increasing the resolution of the DAC system does not imply increasing its static linearity by the same factor. Per 1 bit increase of the resolution, the static absolute linearity improves by a factor of $\sqrt{2}$. The absolute linearity cannot increase at the same rate as the increase of the resolution bits, due to the different gains between the sub-DACs. The sub-DACs feature different gains due to the amplitude mismatches of their currents. Note that the INL per sub-DAC does not take into account the gain error for it is a linear error and hence it does not cause non-linearity (and HD components). However, when two sub-DACs are "stacked" together, the difference of their linear errors (gain) become non-linear errors for the whole DAC system. The gain errors cause large INL errors of the overall DAC system.

At Full Scale, i.e. at code k=2^N-1, the expected error due to the DAC gain errors and the standard deviation of this error is the same for binary and unary DACs, since this error is determined by all DAC currents (only the random mismatch errors are considered). The expected full-scale current and its error, in terms of its standard deviation (σ_u is the standard deviation of the unit element) are:

Equation 14.4-2 (a and b)

$$\mathbb{E}\left(I_{out}\left(2^N-1\right)\right)=\left(2^N-1\right)\overline{I_u} \qquad \text{(a)}$$

$$\sigma\left(I_{out}\left(2^N-1\right)\right)=\sigma_u\times\sqrt{2^N-1} \qquad \text{(b)}$$

This error is transformed to a non-linear error for the whole DAC system, regardless of the op-mode being either "stacking" or interleaving the sub-DACs. To illustrate this argument, consider the example of using 2 unary sub-DACs. Since both sub-DACs are unary, then the whole DAC system is unary. From Equation 6.2-14, it follows that the expected INL_{max} is:

Equation 14.4-3

$$\mathbb{E}\left(INL_{\max}^{(2\,sub-DACs)}\right)=\mathbb{E}\left(INL_{\max}^{(sub-DAC)}\right)\sqrt{2}\approx 0.869\cdot\left(\frac{\sigma_u\sqrt{2\left(2^N-1\right)}}{\overline{I_u}}\right),$$

with N being the sub-DAC resolution, $\left(\frac{\sigma_u}{\overline{I_u}}\right)$ is the relative mismatch of the unit elements, $\mathbb{E}\left(INL_{\max}^{(2\,sub-DACs)}\right)$ is the expected INL_{max} for the two sub-DACs configuration given at (N+1) bit LSB level, $\mathbb{E}\left(INL_{\max}^{(sub-DAC)}\right)$ is the expected INL_{max} for one sub-DAC given at N bit LSB level.

Note that $\mathbb{E}\left(INL_{\max}^{(2\,sub-DACs)}\right)$ is defined to 1 bit higher resolution than $\mathbb{E}\left(INL_{\max}^{(sub-DAC)}\right)$. Thus, the 2 sub-DACs used together yield $\sqrt{2}$ better static absolute linearity than when used separately. Considering the DNL errors, they remain the same $\mathbb{E}\left(DNL_{\max}^{(2\,sub-DACs)}\right)=\mathbb{E}\left(DNL_{\max}^{(sub-DAC)}\right)$, since in unary DACs the DNL errors depend only on the relative matching $\left(\frac{\sigma_u}{\overline{I_u}}\right)$ of the unit elements. Thus, the absolute DNL error improves at the same rate as the increase of the resolution.

To verify and illustrate this reasoning with an actual high-level model of a flexible DAC platform, four 12 bit sub-DACs are statistically simulated. Each sub-DAC features segmentation of 8 LSB binary bits and 4 MSB thermometer bits (15 unary currents). The mismatch of the unit element is set to $\left(\frac{\sigma_u}{\overline{I_u}}\right)=0.018$. Figure 14.4-2 shows 30 simulated INL characteristics of the four 12 bit sub-DACs A, B, C, and D. Note that they are shown relatively to the 12 bit level. The simulated static linearity of the sub-DACs is about 10 bit. Next, the DAC platform is configured to two high-resolution op-modes, “stacked” sub-DACs and “interleaved” sub-DACs, to form a 14 bit D/A function, as described in Table 14.4-1. Figure 14.4-4 and Figure 14.4-6 show the resulting INL characteristics respectively for the “stacked” and “code interleaved op-modes”. Note that they are shown relatively to the 14 bit level. The static linearity does not statistically differ between the two high resolution op-modes. For the whole system the static linearity is about 11 bit. That is to say that the static linearity is improved by 1 bit, while the resolution is increased by 2 bits. Figure 14.4-3, Figure 14.4-5 and Figure 14.4-7 show the respective DNL characteristics. The DNL is improved by 2 bits, i.e. by a factor equal to the increase in the resolution.

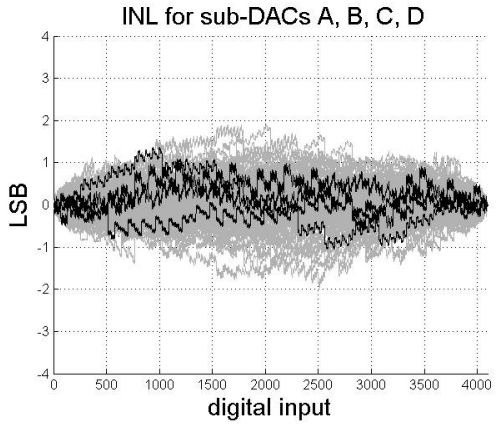


Figure 14.4-2. Simulated 30 INL characteristics for sub-DACs A, B, C, D at 12 bit LSB level.

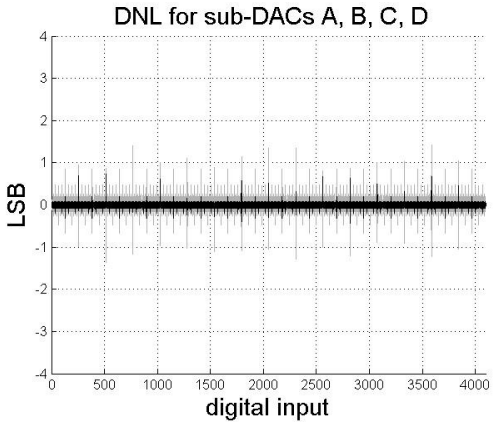


Figure 14.4-3. Simulated 30 DNL characteristics for sub-DACs A, B, C, D at 12 bit LSB level.

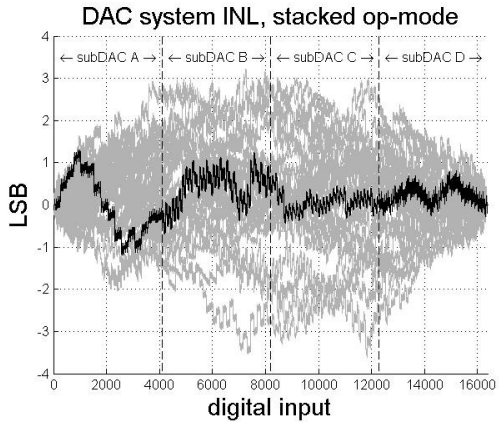


Figure 14.4-4 Simulated 30 INL characteristics of a DAC system at 14b LSB level based on four "stacked" 12b sub-DACs.

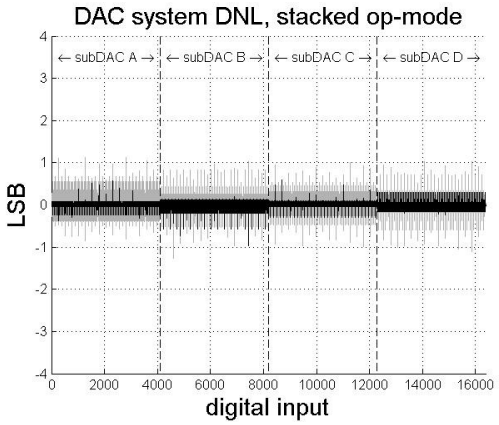


Figure 14.4-5 Simulated 30 DNL characteristics of a DAC system at 14b LSB level based on four "stacked" 12b sub-DACs.

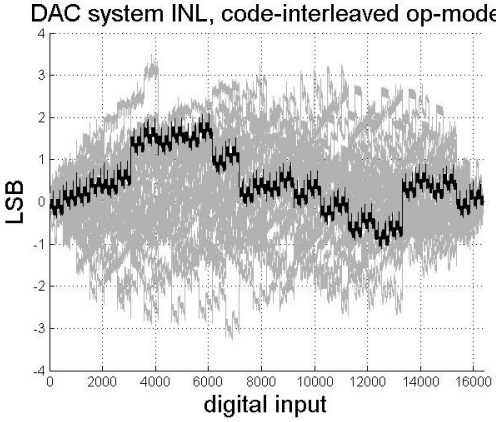


Figure 14.4-6 Simulated 30 INL characteristics of a DAC system at 14b LSB level based on four code-interleaved 12 bit sub-DACs.

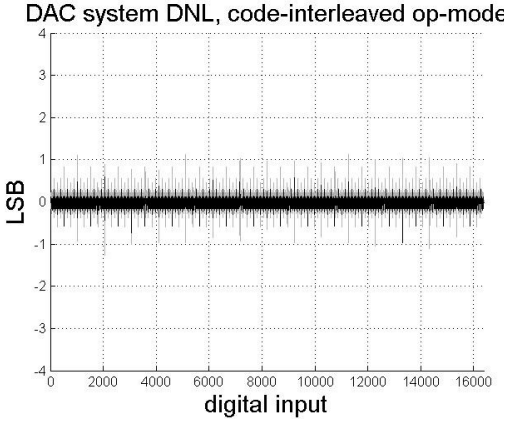


Figure 14.4-7 Simulated 30 DNL characteristics of a DAC system at 14b LSB level based on four code-interleaved 12 bit sub-DACs.

14.4.1.4. Flexible power consumption

When a sub-DAC is not used, it can be switched off and hence power can be saved. The over-all power consumption of the flexible DAC platform is:

Equation 14.4-4

$$P_{all} = mP_{subDAC} + P_{common},$$

where $m \in [1; M]$ are the sub-DACs in use (m depends on the chosen op-mode), P_{subDAC} is the power consumption of a sub-DAC, P_{common} is the power consumption of the shared circuits, e.g. digital pre-processor, analog post-processor, bias circuits, etc. Equation 14.4-4 makes a distinctive difference between the onventional DAC architectures (using no functional segmentation) and the proposed flexible DAC architecture (optionally using functional segmentation). The flexible DAC can switch off sections of its own and still preserve the D/A functionality, while this is not possible without functional redundancy, like in the normal DACs.

P_{subDAC} is defined over the sub-blocks that consume power in a sub-DAC and includes: signal current generation circuits (may coincide with the DAC current cells but also may include e.g. current calibrating circuits, etc.); bias circuits for the DAC current cell, data drivers, synchronization latches; binary-to-thermometer decoders and equivalent delay circuits, local clock drivers, etc. Note that all these circuits are local for the sub-DAC unit and their switching off does not influence the performance of the other sub-DACs and their D/A conversion functions. These local circuits can be either current-steering or CMOS based. The current-steering based circuits are the bias circuits, CML (current-mode logic) gates, the DAC current cells, data drivers, etc. These circuits can be switched off through reducing to zero their bias currents. The CMOS based circuits are CMOS logic gates. They consume power only when being switched. Thus, no special circuits are needed to switch them off, if not used.

14.4.1.5. Flexible dynamic linearity

The error mechanisms at low and high DAC speeds are different. At lower speeds, the main limitations of the DAC dynamic range are the quantization noise and the amplitude mismatch of the DAC signal current sources. At high speeds, the main limitations include the switching synchronization of the DAC current cells, the data-dependent disturbances, the non-linear DAC output capacitance, various spatially related electrical (e.g. voltage drops due to flowing current via parasitic resistances) and material (e.g. variations of V_{th}) on-chip gradients, etc.

The quantization noise directly depends on the resolution of the DAC. For a single tone-type N bit DAC input signal, the signal-to-(quantization)-noise ratio (SNR), according to the well-known formula e.g. from [3], is:

Equation 14.4-5

$$SNR = 6.02N + 1.76 \quad [dB]$$

The amplitude errors related to the DAC signal current matching $\left(\frac{\sigma_u}{I_u}\right)$ are discussed in chapter 5.1. It is also shown via Equation 6.2-14 (for unary DACs) and Equation 6.2-22 (for binary DACs) that increasing the DAC resolution from N bit to N+1 bit (i.e. doubling the number of unit current cells) relaxes the (absolute) matching of the unit current cells

$\left(\frac{\sigma_u}{I_u}\right)$ by a factor of $\sqrt{2}$ for an N bit linearity target. This phenomenon is due to averaging effects. Thus, high DAC resolutions favor the DAC accuracy and linearity at low speeds through two main mechanisms: reducing the quantization noise and averaging the current amplitude errors.

However, the synchronization of the switching moments of many current cells is difficult. Each current cell features some timing errors that become dominant at high speeds, as discussed in relation to Figure 5.3-2 in chapter 5.1. Furthermore, the data-dependent switching errors (discussed in chapter 5.4.) become also dominant at high speeds. Many current cells that connect to the DAC output contribute to the non-linear output capacitance introducing linear and non-linear errors, as discussed in [1]. Therefore, switching only a few current cells should be preferred. Thus, low resolution DACs (having less current cells) are preferable over high resolution DACs at high speeds. Note that this discussion concerns the unary and the segmented DAC implementations. The binary DACs feature area-efficiency but bad dynamic performance due to the difficulties of matching the switching responses of the nominally different current cells. There are some high speed design techniques for binary DACs, e.g. [9]. However, these build the DAC architecture on unary current cells that are grouped in a binary way and hence feature no area-efficiency.

Arguments related to various parasitic RC delays, electrical and material on-chip gradients further contribute to the disadvantages of the high resolution DACs at high speeds. Usually, high-resolution DACs occupy large silicon areas. Note also from the open literature that unary DAC architectures, or segmented DAC architectures with large MSB unary parts are preferred for high speed DACs, e.g. [89], [12]. However, using unary DAC current cells requires silicon area and hence unavoidably causes large on-chip distances among the DAC elements.

Figure 14.4-8 illustrates the various gradients-related problems that occur in large area-occupying DACs. The connections that require matching for both static and dynamic response include the DAC output metal wires, the clock network, the ground and the power supply wires. These connections are sensitive to the electrical on-chip gradients. The length of the path that a signal needs to propagate determines the spatial gradient, since the propagating metal wire unavoidably features some parasitic resistance and capacitance, shown as R_{par} and C_{par}. When the nominally identical current cells are spatially located at different places on the chip, the resulting propagation time delays, slopes of the signals, even filtering of the signals, etc. are unique for each current cell. That is why a data-dependent error is added to the response of the current cells. Indeed, there are techniques to minimize these effects, such as the “star” connections (shown in Figure 14.4-8), spatially locating the current cells depending on their place in the DAC transfer characteristic switching sequence (e.g. [9]), etc. These techniques can only minimize the spatial effects but not eliminate them. In addition to the DAC output network, the clock network features particularly high design requirements for the timing sampling errors. Using “star” connections increases the length of the clock path and due to the parasitic RC constants the targeted high DAC speeds cannot be reached while spending practically reasonable power. Adding clock buffers increases both the transistor mismatch sampling errors and the clock jitter. The power supply and ground networks should also by matched, particularly in the case of CMOS circuits that draw instantaneous currents. Therefore, synchronizing fewer over more current cells is beneficial in general at very high DAC speeds.

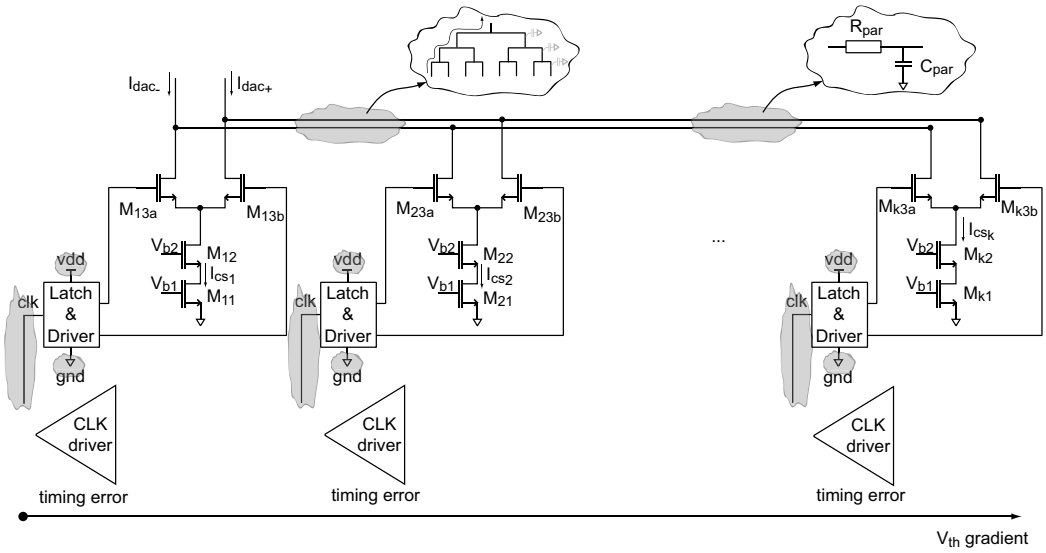


Figure 14.4-8. Various gradients causing dynamic problems in large area-occupying DACs

14.4.2. Special op-modes with DAC correction methods

This text discusses further a high level method for self-measurement (based on [93], [94]; it is independent from what is discussed in chapter 8.3.) and four examples of DAC correction methods that are only possible with parallel sub-DACs:

- an all unary and binary currents calibration [79],discussed in chapter 8.4.3. ;
- an all currents error compensation by mapping, discussed in chapter 12;
- an all currents averaging by data reshuffling, discussed in chapter 3.5. ;
- harmonic distortion cancellation, [64], discussed in chapter 13.

The first two techniques require knowledge about the actual current mismatch errors. A self-measurement op-mode can provide this knowledge. The flexible DAC platform can use its sub-DACs together with a simple sign detector (1b ADC) to measure the errors of a selected sub-DAC under test.

14.4.2.1. Self-measurement op-modes

The flexible DAC platform can be configured to self-measure the errors of its own sub-DACs. The errors are quantified in the digital domain, where they can be used for example by some algorithms of self-correction methods or built-in-self-test (BIST) units. The following text considers the measurement of the static sub-DAC output errors. These errors are mainly caused by the amplitude mismatch of the sub-DAC currents but also can be caused by on-chip systematic errors, gradients, finite output resistance of the current sources, etc. Self-measuring the static DAC errors demonstrates the ability of the DAC platform to be configured into a self-measurement op-mode. Self-measurement of other DAC errors, e.g. dynamic errors, temperature errors, etc., is not discussed, because it is beyond the scope of this work. However, this does not limit the presented DAC platform exclusively to static errors self-measurements.

The self-measurement op-mode hardware configurations are based on concepts of the algorithmic ADCs, e.g. successive-approximation (SAR) ADCs. One of their basic ideas is that an internal analog source is adjusted through digital means to be equal to the ADC input. The digital value of the adjustment eventually represents the measured ADC

input in the digital domain. Figure 14.4-9 shows a basic conceptual diagram of the algorithmic ADC.

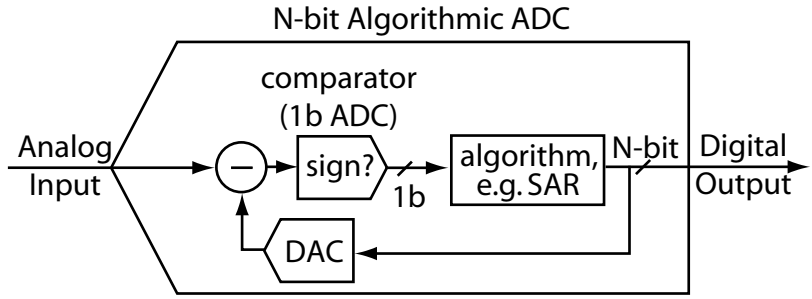


Figure 14.4-9. A basic concept of the algorithmic ADCs.

To realize the A/D conversion an ADC block and a DAC block are used. The ADC is usually only 1 bit, a comparator, and hence it is linear by definition. Its digital output is used by the algorithm block that generates an input code word for the DAC block. The output of the DAC is subtracted from the analog input. The goal of the algorithm block is to find such a DAC input word, so that the DAC output is equal to the analog input and hence the DAC input word is the digital representation of the measured analog input. The input of the DAC (with a possible correction) can be used as an output of the whole N bit algorithmic ADC. Thus, the DAC represents a reference for the analog input to be measured against. Therefore, the linearity of the whole algorithmic ADC is limited by the linearity of the DAC. Note that similar basic algorithmic ADC concepts are used for the self-measurement of the current source amplitude errors in chapter 10 and demonstrated in practice in chapter 16. However, these concepts are applied there to current sources, i.e. at low level, while here these concepts are applied to sub-DACs, i.e. at high system level.

These concepts are suitable for the DAC flexible platform, since it contains multiple parallel sub-DACs, and a digital pre-processor that can be designed/programmed to perform as the algorithm block. The so configured algorithmic ADC can be used for self-measurement of the sub-DAC errors. To measure the errors of some sub-DACs under test, other sub-DACs are configured as reference DACs. Their output current is scaled down to reduce the resolution step, so that they can quantize the nominal LSB step.

The conceptual schematic of a possible self-measurement op-mode is shown in Figure 14.4-10. To measure the errors in one sub-DAC, the self-measurement op-mode requires a current sign detector (1 bit ADC comparator) and three sub-DACs. Effectively, the algorithm acquires information for the sub-DAC DNL in the digital domain, which enables current mismatch errors to be retrieved.

For each input code for sub-DAC 1, the current is first compensated by an output current of sub-DAC 2, which is given the same input code, plus a fine current from a down-scaled sub-DAC 3. Next, the code of sub-DAC 1 is set one LSB higher; this is compensated by increasing the code at sub-DAC 3. The differences in codes for sub-DAC 3 indicate now the DNL error. For example, to measure the DNL_k error of sub-DAC 1 at digital code *k*, the following operation steps are executed. The digital code X_k is placed at the inputs $X1$ and $X2$ of sub-DACs 1 and 2. Their outputs are subtracted, giving $I_1 - I_2$. The scaled down output of sub-DAC 3 is added and the sign of the combined current is evaluated. The input of sub-DAC 3 $X3$ is successively changed in a step-wise manner until the output of the sign-detector is changed. This is an indication that $I_1 - I_2 + I_3$ approaches zero for digital word $X3_1$, i.e. $I_2(k)-I_1(k)=I_3$, where the index 1

indicates the first part of the measurement procedure. The next part begins with incrementing the digital input of sub-DAC 1, i.e. $X1 = X_{k+1}$. The output of sub-DAC 1 increases by 1 LSB. The input $X3$ is adjusted until $I_1 - I_2 + I_3$ again approaches zero, i.e. $I_2(k)-I_1(k+1)=I_3$. The value of $X3$ is stored as $X3_2$. The digital difference $X3_2 - X3_1 - X3_{LSB}$ is an indication of the DNL_k error of sub-DAC 1, where $X3_{LSB} = \frac{1}{2^N - 2} \sum_{k=1}^{2^N-2} X3_{k+1} - X3_k$ is the average LSB step. Once the error information is available, a correction can follow.

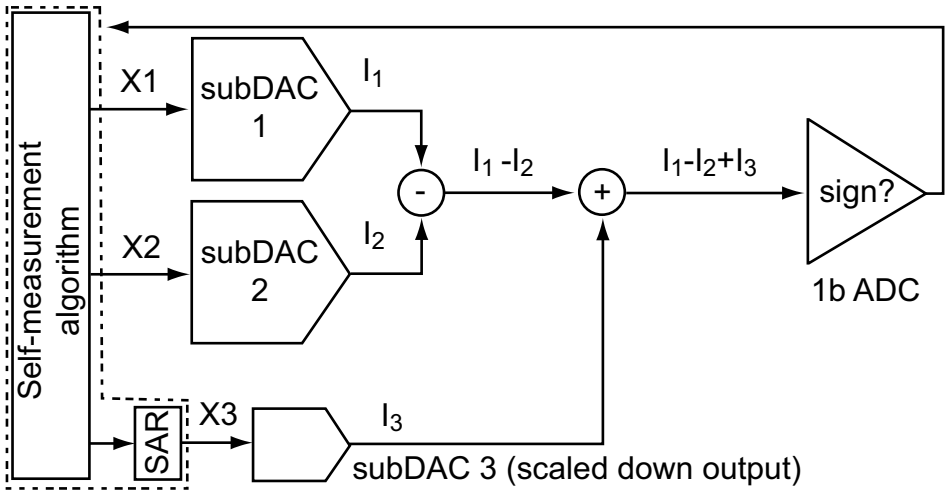


Figure 14.4-10. Self-measurement op-mode.

Scaling down the output of sub-DAC 3 requires reducing the amplitude of its LSB step via decreasing the V_{gs} voltage of the DAC current source transistors. Thus, the LSB current from $\overline{I_u}$ becomes $\overline{I'_u}$, with $\overline{I_u} > \overline{I'_u}$. As already discussed, it follows from Equation 2.3-2 that this approach increases the mismatch errors in sub-DAC3, i.e. $\left(\frac{\sigma_u}{\overline{I_u}}\right) < \left(\frac{\sigma'_u}{\overline{I'_u}}\right)$. However, the absolute matching is improved because the current is scaled down, i.e. $\sigma_u > \sigma'_u$. Therefore, with appropriate scaling for the sake only of measurement accuracy, the required relative matching to I_u before scaling $\left(\frac{\sigma'_u}{\overline{I_u}}\right)$ can be achieved.

14.4.2.2. Unary and binary currents calibration

A novel calibration algorithm uses redundant sets of binary currents to calibrate all of them to a common reference. This is published in [79] and discussed in chapters 8.4.3. and 10.4. The calibration method is suitable for integration in the presented DAC platform. A chip implementation co-integrated with the presented flexible DAC platform (chapter 16) is the first published DAC chip that can calibrate *all* of its current sources, both unary and binary. In this way, it is possible to calibrate the complete DAC system, independent of the algorithmic constitution of binary, unary or segmented current sources. Therefore, the post-calibration DAC accuracy does not depend on the manufacturing tolerances but only on the design parameters of the correction circuits (see e.g. Figure 16.3-5). As presented in [79], the calibration method can autonomously measure the current errors but the presented self-measurement op-mode in the previous chapter is also suitable.

Providing that all currents are calibrated, scaling down the sub-DAC LSB step through reducing V_{gs} of the unit current I_u does not increase the mismatch errors, since the self-calibration engine corrects for all amplitude errors of the current cells.

However, the calibration advantages are often lost at very high speeds. An alternative correction solution is mapping, which can be used at both low and high speeds.

14.4.2.3. Mapping

The mapping technique that is previously described for low level unary current cells is expanded to high level sub-DACs. This is discussed in chapter 12. The main concept is to combine the parallel sub-DACs in such a way that their errors mutually compensate each other. The technique is available for both amplitude and timing errors. The method requires a-posteriori knowledge of the errors, which can be acquired through self-measurement. Possible self-measurement techniques for amplitude errors are described in chapter 8.3. and chapter 14.4.2.1. Note that the mapping technique should take into account the "missing code problem" of using parallel sub-DACs that is discussed in chapter 14.5.

14.4.2.4. Data reshuffling

As discussed in the previous sub-chapter, the mapping technique requires knowledge of the actual mismatch errors. When either this knowledge is not available or the chosen map is compromised between the static and the dynamic DAC errors, random averaging of the mismatch errors can be used in order to reduce the harmonic distortion of the DAC. The randomization transforms the power of the non-linear distortion into "white noise". For this technique, the digital pre-processor needs to randomize the distribution of the digital sub-words $w_i(nT)$ while always respecting $w(nT) = \sum_{j}^{k} w_i(nT)$, for parallel sub-DACs from the j-th to the k-th sub-DAC. The parameters j and k are meant to indicate only (k-j) number of sub-DAC, i.e. illustrating multiple sub-DACs. This argument is valid for any number of multiple sub-DACs, saying that as long as the total sum of the sub-DACs input words $w_i(nT)$ equals the input DAC word $w(nT)$, the particular sub-DAC input words $w_i(nT)$ can be randomized. The errors are averaged by continually randomizing the output contribution of each sub-DAC. A similar technique based on two parallel binary DACs is demonstrated in an independent research work [32]. Note that the high-level data reshuffling method should take into account the "missing code problem" of using parallel sub-DACs that is discussed in chapter 14.5.

14.4.2.5. Harmonic distortion cancellation

All discussed special op-modes in the previous sections focus on the mismatch errors between the parallel current cells. Another important error concern is the class of systematic errors. These errors cause deviation from the ideal behavior that are partly identical for each identical unit current source, and hence partly identical for each sub-DAC. Examples of these errors are the non-linear switching of the transistors, systematic parasitics due to the layout, output glitches, clock-feedthrough, data-feedthrough, data-dependent disturbances of the substrate and power rails, etc. These errors result in harmonic distortion (HD) of the input signal. A method for the suppression of harmonic distortion is proposed in [64]. This method is able to counteract the effect of the DAC systematic error mechanisms by implementing the D/A conversion through parallel branches, i.e. sub-DACs. Each of the sub-DACs converts a digital input signal $w_i(nT)$

that has phase shifted information relatively to the main signal $w(nT)$. In this way, a HD component targeted by the customer for removal can be cancelled, while the power of the main signal is only slightly decreased. For more detailed information on the method see [64] and section 13.

14.5. THE "MISSING CODE PROBLEM"

Usually, an N bit DAC can be modeled as using 2^N-1 unit elements to convert 2^N digital codes to an analog output. In practice, some of the unit elements are grouped in a binary way. Note that to implement code 0, no unit elements are used. That is why when M such N bit sub-DACs are used, the available equivalent unit elements are $M \times (2^N - 1) = M \times 2^N - M$. Thus, M-1 codes are missing from the complete equivalent $N + \log_2 M$ resolution transfer characteristic. This issue requires extra computation steps for some op-modes of the flexible DAC platform. For example, the op-mode of a single DAC with large full-scale range and high resolution (option 1, decoder-based) given in Table 14.4-1 should recalculate the sub-DAC input $w_i(nT)$ for the active sub-DAC, subtracting the adjustment for the "missing codes" from the DAC input. However, when all sub-DACs have 2^N unit elements, instead of 2^N-1, such recalculation is not necessary and then the LSB portion of the DAC input can be directly assigned to the active sub-DAC. Thus, an extra LSB current cell for each sub-DAC is useful.

The extra LSB current cell can be implemented as a copy of the normal current cell. Such an extra LSB current cell is not exclusively needed only to solve the "missing code" problem. For example, the calibration algorithms described in Equation 8.4-1 and Equation 8.4-2 also use an extra dummy LSB current, anyway.

Note that for some op-modes, the "missing code problem" is not an obstacle. For example, op-modes such as full-scale range and high resolution (option 2, bits shift based), do not require re-computation of the sub-DAC words. Nevertheless, the input digital word should take into account that the DAC system full-scale range is limited to M(2^N-1) codes (M-1 codes less than what it is normally expected for $N + \log_2 M$ resolution DAC).

14.6. CONCLUSIONS

The concept of DAC flexibility is introduced. A flexible architecture for current-steering DAC platforms is proposed. The architecture is based on parallel sub-DAC units, a digital pre-processor, and an analog post-processor. This flexible architecture provides flexibility in DAC design, functionality, and performance. Various flexible hardware configurations are possible, e.g. digital pre-processing based-on mapping, DSP techniques, DEM. For most of these configurations, various flexible programming is further possible. Their choice depends on the available resources, targeted performance, and functional requirements of the application.

The proposed DAC platform can be used as a single or multiple independent DACs. To realize an individual D/A function, various configuration options are available. The single DAC operation is a basic op-mode configuration, for which the gain, the resolution, the power consumption, and the statistic and dynamic linearity are flexible.

The inherent functional redundancy of the parallel sub-DACs enables a number of special operation modes that can further improve DAC performance. These are DAC correction methods, such as all unary and binary currents calibration (introduced in chapter 8.4.3.); high-level mapping (introduced in chapter12), high-level DEM (discussed

in chapter 3.5.), suppression of HD (introduced in chapter 13), etc. The parallel sub-DAC can also be configured in a self-measurement op-mode.

Such DAC correction methods allow relaxing the design requirements to the DAC analog intrinsic core and hence allowing smaller and more area efficient designs. Therefore, the proposed DAC platform architecture is argued to be a suitable candidate for co-integration with other flexible SoC platforms, such as the digital FPGAs.

PART V: DESIGN EXAMPLES

This part includes chapters that present three test-chip DAC implementations – in 250nm, 180nm, and 40nm CMOS processes. Chapter 15 presents a validation of the redundant decoder concept. Chapter 16 presents two new implementations of current self-calibration methods. Chapter 17 presents a validation of the new method for suppression of HD. Chapter 18 presents a flexible DAC platform, based on 4 sub-DAC unit cores, in 180nm CMOS. Chapter 19 presents a flexible DAC platform, based on 16 sub-DAC unit cores, in 40nm CMOS.

15. Chapter

A REDUNDANT BINARY-TO-THERMOMETER DECODER DESIGN

This chapter presents a test-chip implementation of a CML 4-to-15 redundant binary-to-thermometer decoder in a 12 bit 180nm CMOS DAC. The relative increase of the occupied area due to using the new concept is estimated to be as low as 1%, which suggests that the correction methods using the new redundant decoder should be very efficient. Measurement results show that even for such a small increase in the required design resources, a considerable improvement in the DAC performance can be achieved. The measured improvement in INL_{max} is 1.5LSB, while the measured improvement in SFDR is 5.7dB for f_{in}=6.8MHz and F_s=100MS/s. The presented decoder can be used with either mapping DAC correction methods or DEM DAC correction methods.

15.1. INTRODUCTION

Binary-to-thermometer decoders are used in the binary LSB/unary MSB segmented and fully unary DACs. Their main task (function) is to convert the binary coded input digital word into a thermometer (unary) coded digital word. Equation 15.1-1 gives a short example of 4 bit binary to 15 bit thermometer (unary) decoder as a conversion matrix. To represent a number X, the columns of the binary matrix are weighted with the powers of 2, while all the columns of the unary matrix have the same weight. The binary coding features efficiency, while the unary coding features inherent redundancy. Since the digital bits of the unary coding have the same weight, a number X can be represented in different ways (and hence redundancy in coding). For example, the two unary codes 0000001 and 0000010 represent the same number X=1. Therefore, the decoder should implement a "switching sequence", which defines how exactly the numbers are always represented. The particular switching sequence where an increment of X represents switching from 0 to 1 a consecutive unary bit is called in this thesis "*thermometer coding*". It is one case of unary coding.

Equation 15.1-1

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \end{array} \quad \leftarrow weight \rightarrow \quad \begin{array}{ccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$$\begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&0&1&1 \\ 0&1&0&0 \\ 0&1&0&1 \\ 0&1&1&0 \\ 0&1&1&1 \\ 1&0&0&0 \\ 1&0&0&1 \\ 1&0&1&0 \\ 1&0&1&1 \\ 1&1&0&0 \\ 1&1&0&1 \\ 1&1&1&0 \\ 1&1&1&1 \end{bmatrix}_{binary} \xrightarrow[4b \rightarrow 15u]{decoder} \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}_{unary(thermometer)}$$

Chapter 11 presents a new binary-to-unary decoder concept that features two unary switching sequences with low correlation. Applying this concept to a 4 bit binary to 15 bit unary decoder, the two resulting switching sequences are:

Equation 15.1-2

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \end{array} \quad \leftarrow weight \rightarrow \quad \begin{array}{cccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$$\begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&0&1&1 \\ 0&1&0&0 \\ 0&1&0&1 \\ 0&1&1&0 \\ 0&1&1&1 \\ 1&0&0&0 \\ 1&0&0&1 \\ 1&0&1&0 \\ 1&0&1&1 \\ 1&1&0&0 \\ 1&1&0&1 \\ 1&1&1&0 \\ 1&1&1&1 \end{bmatrix}_{binary} \xrightarrow[4b \rightarrow 15u]{decoder} \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}_{unary(\text{mode 1})}$$

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \end{array} \quad \leftarrow weight \rightarrow \quad \begin{array}{cccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$$\begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&0&1&1 \\ 0&1&0&0 \\ 0&1&0&1 \\ 0&1&1&0 \\ 0&1&1&1 \\ 1&0&0&0 \\ 1&0&0&1 \\ 1&0&1&0 \\ 1&0&1&1 \\ 1&1&0&0 \\ 1&1&0&1 \\ 1&1&1&0 \\ 1&1&1&1 \end{bmatrix}_{binary} \xrightarrow[4b \rightarrow 15u]{decoder} \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\ 1&0&0&0&1&0&0&0&1&0&0&1&1&0&0&0 \\ 1&0&0&0&1&0&0&1&1&0&0&1&1&0&0&0 \\ 1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&0 \\ 1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1 \\ 1&0&0&1&1&0&0&1&1&0&1&1&1&0&0&1 \\ 1&0&0&1&1&0&1&1&1&0&1&1&1&0&0&1 \\ 1&0&1&1&1&0&1&1&1&0&1&1&1&0&0&1 \\ 1&0&1&1&1&0&1&1&1&0&1&1&1&0&1&1 \\ 1&0&1&1&1&0&1&1&1&1&1&1&1&0&1&1 \\ 1&0&1&1&1&1&1&1&1&1&1&1&1&0&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1 \end{bmatrix}_{unary(\text{mode 2})}$$

In the absence of mismatch between the weights of the unary coding, mode switching sequence, i.e. mode 1 and 2, deliver the same output result. However, in a DAC IC implementation, the weights are implemented with real analog elements that unavoidably feature some mismatch. Therefore, the two switching sequences provide two different DAC transfer characteristics, i.e. the redundant decoder creates redundancy in the DAC signal transfer function. This chapter shows how this redundancy can be used to improve basic DAC characteristics.

Section 15.2. presents a design example of a 4-to-15 redundant decoder implemented in a 180nm DAC. Section 15.3. presents measurement results. Finally, conclusions are drawn in section 15.4.

15.2. DESIGN EXAMPLE

To validate in practice the concept of the redundant decoder (presented in chapter 11), a 4-to-15 binary-to-thermometer decoder is implemented in a 12 bit current-steering DAC in 180nm. Figure 15.2-1 shows the layout and the chip micrograph of the implementation.

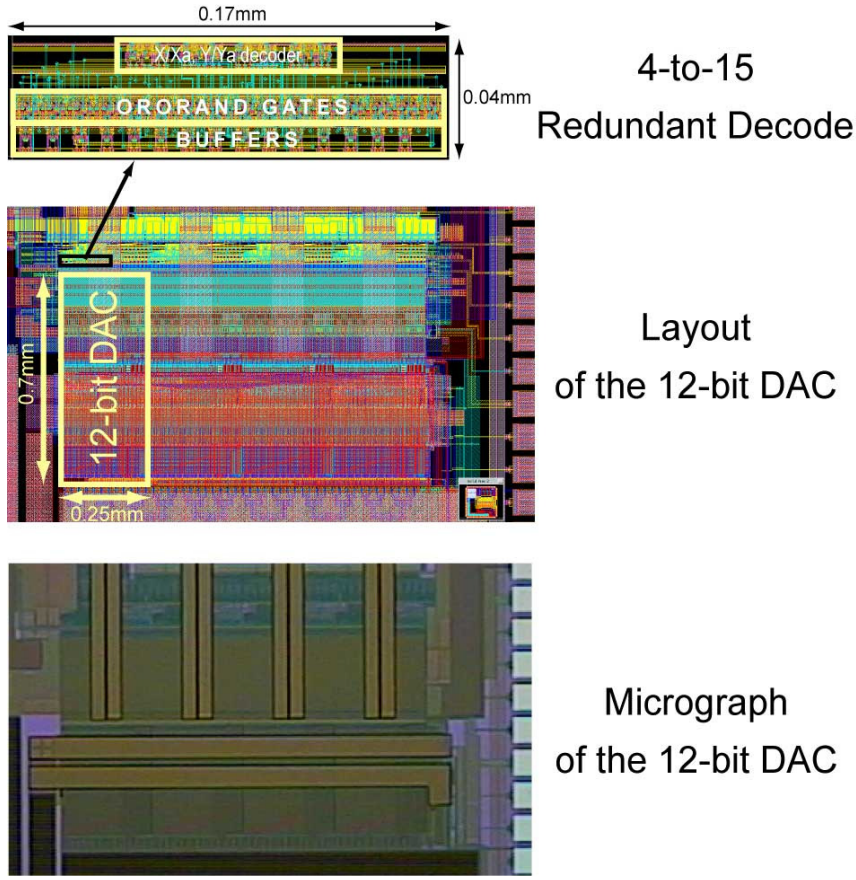
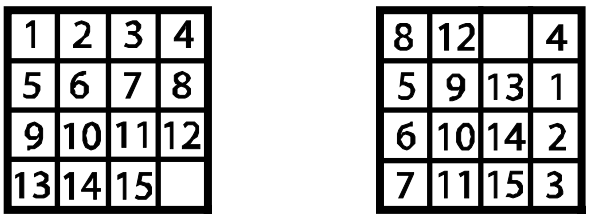


Figure 15.2-1. Layouts of the redundant decoder and the 12 bit DAC with the micrograph of the manufactured test chip in 180nm CMOS IC technology.

The digital logic of the decoder implementation is CML (current-mode-logic). It occupies about 0.0068mm^2, while the 12 bit DAC occupies about 0.175mm^2. Thus, the redundant decoder is about 4% of the DAC area. However, this test chip includes other independent test-circuits within the DAC core, occupying area of about 0.03mm^2. These are self-measurement and self-corrections circuits, which are not relevant for the discussed correction method. If these are excluded, then the portion of the redundant decoder is about 5% of the DAC core, which a realistic figure for a dedicated implementation. Note however that the extra added area is much smaller, since a binary-to-thermometer decoder is needed anyway. Using this argument, the estimation can be made that the extra area of the method itself is less than 1% (see chapter 11 for comparison between a conventional decoder and the new redundant decoder).

The two switching sequences that correspond to the matrices of Equation 15.1-2 are shown in Figure 15.2-2. For a DEM correction method, selecting the switching sequence should be randomized. For a mapping correction method, selecting the switching

sequence should be done based on measured error information. In this test chip, selecting the switching sequence is done manually. That is to say that the self-measurement block and the algorithm block for this prototype implementation of the presented mapping correction method is done manually.



Switching sequence 1 (mode 1) Switching sequence 2 (mode 2)

Figure 15.2-2. Implemented decoder of 4 binary bits to 15 unary bits with two switching sequences.

15.3. MEASUREMENT RESULTS AND DISCUSSION

The test chip implementation is investigated for static and dynamic signals. The measurements follow the theoretical estimations of chapter 11, where statistical results are presented, e.g. yield against current matching estimation. Here performance parameters of an exemplary DAC sample are presented.

An example of a measured INL with the two switching sequences is shown in Figure 15.3-1. The measured INL_{max} for switching sequence 1 (mode 1) is INL_{max}=2.2LSB, while for switching sequence 2 (mode 2) it is INL_{max}=3.7LSB. Both DAC transfer characteristics use the same physical elements – current sources, having the same errors. The two maps (i.e. switching sequences) of the redundant decoder order these errors in different ways and hence provide two different, though correlated, DAC transfer characteristics. However, there is still a considerable improvement, e.g. 1.5LSB improvement of INL_{max} (for statistical simulations, see chapter 11), thanks to the low correlation of both maps (Figure 15.2-2). Note that the low correlation results from the facts that in mode 1 the switching sequence develops horizontally, while in mode 2 the switching sequence develops vertically.

The advantages of having redundant switching sequences for the unary MSB DAC part are also valid for the dynamic DAC performance. The two switching sequences produce two different DAC transfer characteristics with different non-linearity (shapes). That is why there are differences in the DAC output spectral contents for both switching sequences. The measured SFDR for switching sequence 1 (mode 1) for f_{in}=6.8MHz is about 67.2dB (shown in Figure 15.3-2), while the measured SFDR for switching sequence 2 (mode 2) is about 61.5dB (Figure 15.3-3). This is a difference of about 5.7dB for a very small investment in extra area.

Looking at the INL of switching sequence 2 in Figure 15.3-1, an arch shape can be identified that suggest a strong HD_2. This strong HD2 component can be seen in Figure 15.3-3 to limit the DAC SFDR. The DAC INL with switching sequence 1 is more balanced between the positive and the negative INL values, suggesting the dominating non-linearity order should be an odd number. Three major crossings of the x axis can also be outlined, suggesting a fifth order non-linearity. Indeed, Figure 15.3-2 shows that HD_5 limits the DAC SFDR.

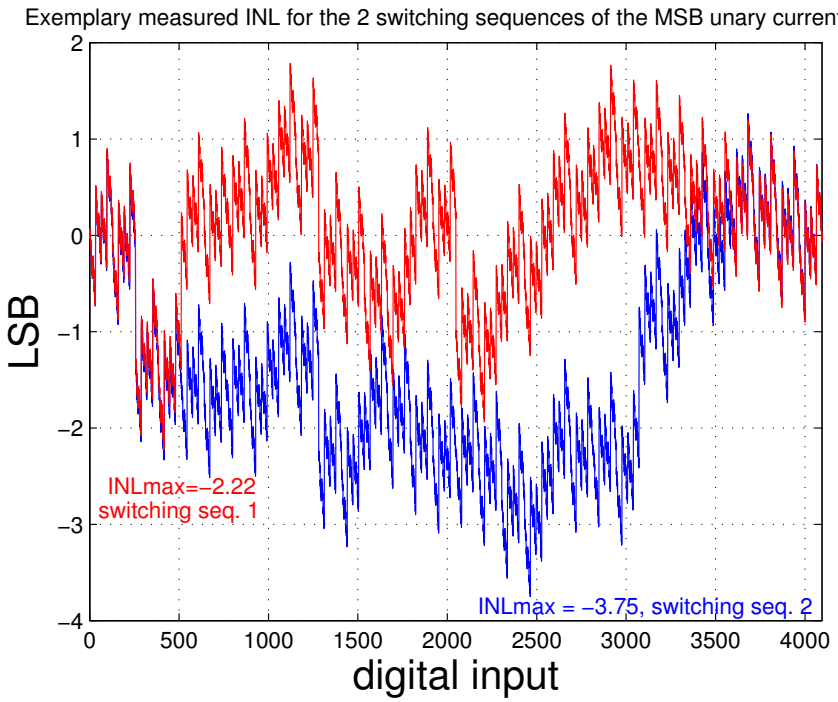


Figure 15.3-1. Exemplary measured INL for both switching sequences of the redundant decoder.

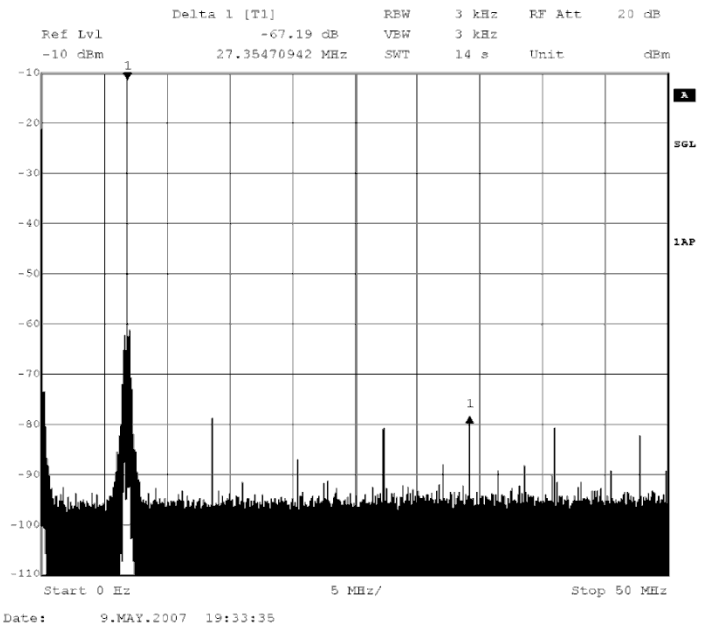


Figure 15.3-2. DAC output spectrum for 6.8MHz@100MS/s, with switching sequence 1, SFDR=67.2dB.

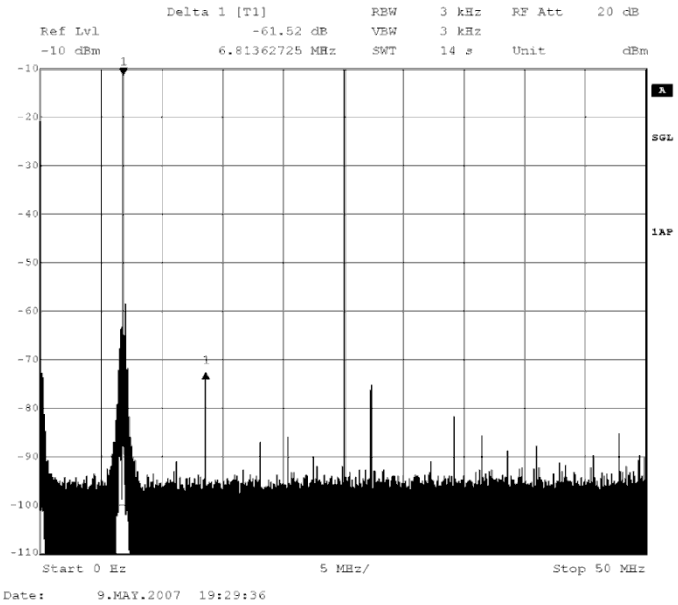


Figure 15.3-3. DAC output spectrum for 6.8MHz@100MS/s, with switching sequence 2, SFDR=61.5dB.

The important advantage of the new decoder architecture is its area and power efficiency with respect to the improvement for the whole DAC. For the high-speed current mode logic, e.g. used in [1], the required change, from an ORAND gate to an ORORAND gate, costs a single transistor and no extra power. Some more resources are required in the first decoding level. However, related to the overall DAC complexity, the required price may be negligible. Note that although this chapter demonstrated the core concept of redundancy through the use of 2 thermometer outputs, the advantage of this technique can be further increased by using even more switching sequences through parallelizing and cascading similar decoders.

15.4. CONCLUSIONS

A new binary-to-thermometer decoder with built-in redundancy for segmented and unary DACs is implemented, according to the concepts of chapter 11. Instead of the conventional single thermometer switching sequence, the proposed decoder can generate two different switching sequences featuring low correlation. This built-in redundancy affects the whole DAC through allowing two different transfer characteristics.

An implementation of the proposed decoder for 4-to-15 binary-to-thermometer decoder is demonstrated in a 12 bit 180nm CMOS DAC. Measurement results of INL show that the two different DAC switching sequences generate two different DAC transfer characteristics featuring different levels of linearity. Measurement results also show that the two different DAC switching sequences also generate two different DAC output spectra, when an input sinewave signal is converted (processed). It is demonstrated that the SFDR difference between the two spectra can be as large as 6dB.

Taking into account the classification of chapter 7, it can be suggested that the proposed decoder can be used either with a mapping method (with self-measurement and static map) or with a DEM method (without self-measurement and dynamically alternating maps). Ultimately, the proposed decoder can lead to improved chip yield and hence to reduced price of segmented DACs of all types: resistor based, switched capacitors, or current-steering type, since it is a generic concept.

16. Chapter

TWO SELF-CALIBRATING DAC DESIGNS

This chapter presents the implementation and measurement results of two DAC test-chip implementations in 250nm and 180nm standard CMOS processes. The chapter considers the test-chips primarily from the calibration point of view. The measurement results show that the practical limit of the presented calibration easily exceeds the 14 bit level. The 250nm and 180nm test chips feature fully-integrated calibration engines. The first test-chip is a 12 bit 250nm current-steering DAC with unary currents calibration. The second test-chip is a flexible 12 bit quad-core 180nm current-steering DAC with both binary and unary currents calibration. This is the first reported DAC implementation that calibrates the errors of all its current sources.

The calibration of all current sources makes the DAC accuracy independent of the tolerances of the manufacturing process. The overall DAC accuracy depends on a single design parameter – the correction step. The three test-chips use simple circuits for the calibration: a current comparator (1b ADC) for self-measurements, small calibrating DACs (CALDACs) connected to the main DAC currents for self-correction and a fully on-chip integrated calibration algorithm. That is why all these test-chips are among the smallest reported examples of high-speed high-performance DACs.

16.1. INTRODUCTION

To practically investigate and prove the presented calibration concepts and design arguments in chapter 10, this work presents three self-calibrated DACs. The circuits of the three self-calibration elements: self-measurement, algorithm, and self-correction, are completely integrated on-chip in two of the test-chip DACs – the 250nm and 180nm CMOS implementations. The third test-chip is a 40nm CMOS implementation featuring on-chip integration of self-correction, self-measurement circuits, and a serial interface for communication with various off-chip algorithms for the sake of research.

The first DAC is designed in a standard 250nm CMOS process. It is a 12 bit DAC with algorithmic segmentation level of 6 MSB unary bits (63 unary cells) and 6 LSB binary bits (cells). All unary current sources are calibrated with an on-chip fully integrated self-calibration method. Based on the measurement results, the calibration of the binary LSB currents is recommended. Note that independent research projects support such a conclusion, too, e.g. [69].

The second DAC is designed in a standard 180nm CMOS process. It is a quad-core 14 bit DAC with unary functional segmentation of four 12 bit sub-DACs, each featuring an algorithmic segmentation of 4 MSB unary bits (15 unary cells) and 8 LSB binary bits (cell). Both unary and binary currents are calibrated with a fully on-chip integrated self-calibration method. This is the first DAC described in the literature to calibrate all current sources both unary and binary. The accuracy of this DAC does not depend on the process parameters A_β and A_{VT}. Its accuracy depends only on the design parameter CALDAC LSB step (the correction step).

This chapter extends further the publications [35] and [47], where the 250nm and 180nm are presented for the first time. Section 16.2. presents measurement results of calibration of unary current, based on the 250nm test chip implementation. Section 16.3. presents measurements results of calibration of both unary and binary currents, based on the 180nm DAC. Section 16.4. compares the achieved results with the published DAC state-of-the art. Finally, conclusions are drawn in section 16.5.

16.2. UNARY CURRENTS SELF-CALIBRATION IN A 250NM DAC

A 12 bit self-calibrated current-steering DAC is implemented in a standard 250nm 1P5M CMOS process with a power supply of 2.5V. The self-measurement probes is based on the principle of current deviation at the second cascode as shown in Figure 8.3-3b. As discussed in chapter 8.3. and shown in Table 8.3-1, this architecture balances between occupied extra area, needed voltage headroom, measurement quality, and charge feed-through. The self-measurement device is based on the example shown in Figure 8.3-8.

As discussed, this measurement device is the most efficient one when only unary currents are calibrated. The reference current for the calibration of the MSB unary currents is defined and constructed as the sum of all binary currents plus one dummy LSB current. Thus, the mismatch error in the transition from the binary currents to each of the unary current is minimized. The calibration algorithm is presented in Figure 8.4-1 and Figure 10.2-5. It is a simple, area efficient 8-state FSM, which controls the correction quantization error, to improve the post-calibration accuracy of the whole DAC (see Figure 10.2-2, Figure 10.2-3, and Figure 10.2-4). The self-correction method is based on 6 bit digitally controlled CALDACs that are attached to the drain of every unary current source transistor.

16.2.1. Design

To reduce the occupied silicon area, the DAC core was designed for 10 bit accuracy. It has 6-6 segmentation and only the 63 MSB unary current sources are calibrated. The 6 LSB binary currents are not calibrated but they are used to construct the reference current. A high-level block diagram of the self-calibrated DAC is shown in Figure 16.2-1, (based on a related publication [35]).

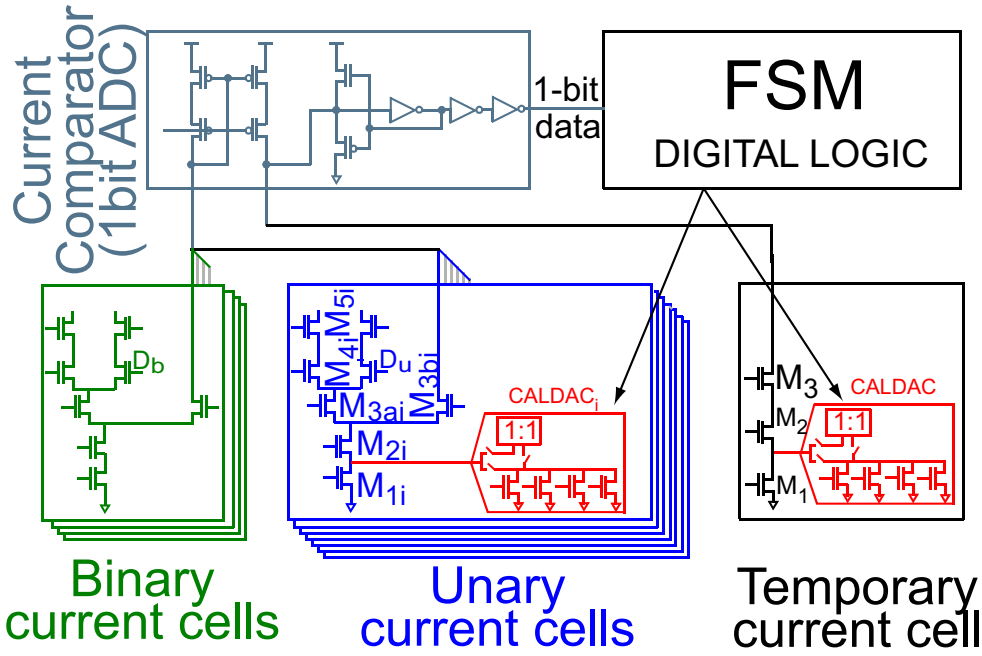


Figure 16.2-1. Self-calibration DAC scheme with unary currents self-calibration.

The 2.5V power supply allows the use of 2 cascodes (M2 and M3) on top of M1. M3 is doubled and used as a switch to redirect the current either to the calibration circuit (via $M3b_i$) during the start-up calibration phase or to the current switch transistors (via $M3a_i$) during the normal operation of the DAC. Thus, the self-calibration apparatus corrects for all random and systematic errors of M1 and M2. The correction of the first cascode M2 is necessary, since M1 is designed for smaller area, i.e. its length is only 4µm (instead of typically around 16µm), and its low output resistance would convert the mismatch in the threshold voltage of M2 into a current error. The correction CALDAC current is injected at the drain of M1. To realize a small correction step, the CALDAC current source transistors M1cbk (see Figure 8.5-4) and the DAC main coarse current source transistors M1i (see Figure 16.2-1) have different bias voltages V_{gs} and different ratios W/L.

The realized 12 bit self-calibrated DAC is shown in Figure 16.2-2. The dimensions of the DAC core are $1.16mm \times 0.98mm$. The array of current source transistors occupies only $0.11mm^2$; the array of CALDACs occupies $0.18mm^2$; the calibration logic occupies $0.11mm^2$.

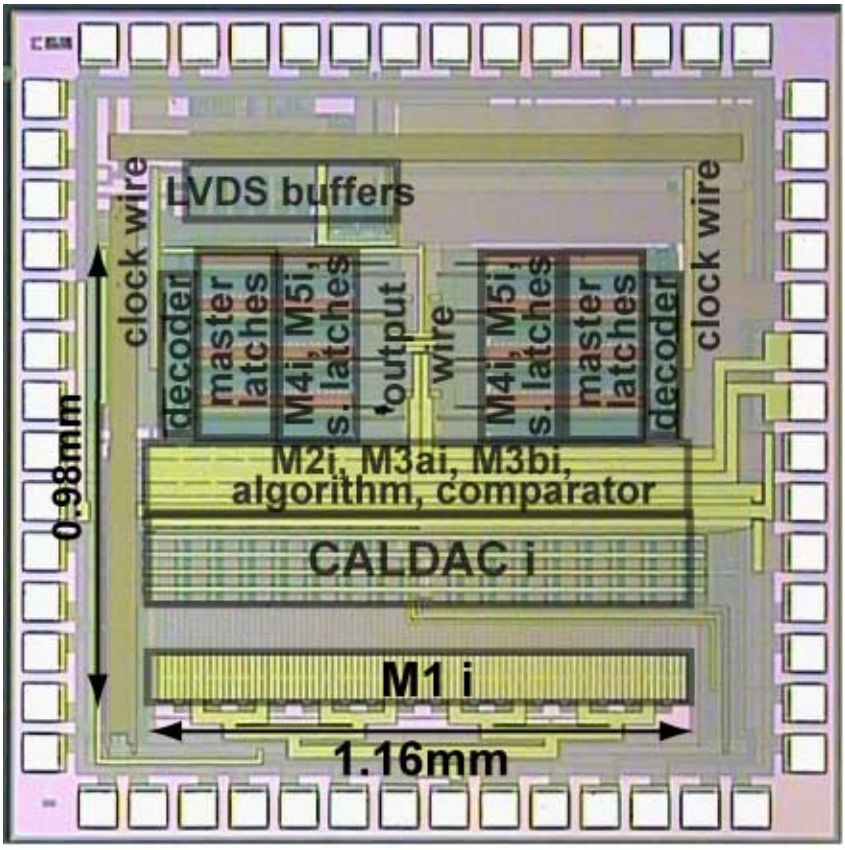


Figure 16.2-2. A micrograph of the 250nm self-calibrated DAC.

Before and after calibration, the power consumption is 270mW. It is almost independent of the conversion speed, because the DAC core uses current mode synchronization latches, with only the decoder being realized with differential CMOS logic. The power consumption is mainly due to the synchronization latches while no power is dissipated in the calibration apparatus during the normal operation of the DAC.

16.2.2. Measurements

The presented results are measured for 20mA full-scale current terminated on a 50Ω differential load resistance. Figure 16.2-3 shows the exemplary (and representative) INL static characterization of the DAC before and after self-calibration. The errors of the MSB unary currents dominate and the maximum INL before calibration is about INL_{max}=1.5LSB. After calibration, the errors of the unary currents are corrected and the linearity is improved to INL_{max}<0.4LSB. After calibration, the remaining errors are due to the non-calibrated binary part.

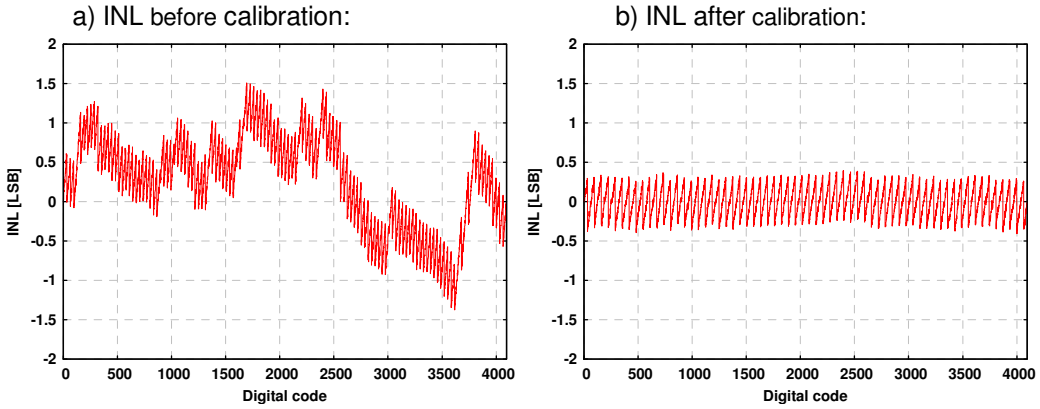


Figure 16.2-3. DAC INL: a) before calibration; b) after calibration;

Figure 16.2-4 shows the DNL static characterization of the DAC before and after self-calibration. The errors of the MSB unary currents and the error of the more-significant binary currents dominate. The maximum DNL before calibration is about DNL_{max}=1LSB. After calibration, the errors of the unary currents are corrected and the linearity is improved to DNL_{max}<0.6LSB, which is limited by the mismatch errors of the non-calibrated more significant bit currents of the DAC binary LSB part.

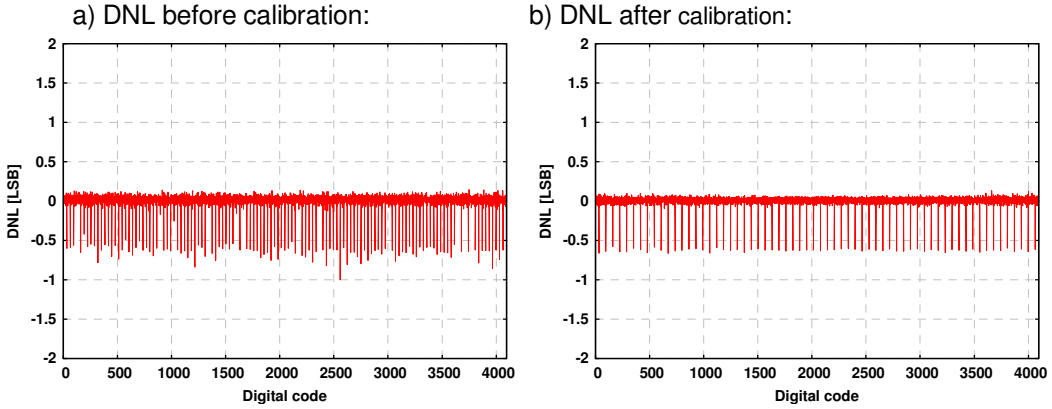


Figure 16.2-4. DAC DNL: a) before calibration; b) after calibration;

To isolate only the calibration improvement from the overall DAC performance and hence to estimate the potential of the calibration method, 1827 unary current sources from different chip samples have been measured before and after calibration. Figure 16.2-5 shows the measurement results as a distribution diagram of the relative LSB accuracy, normalized at 12 bit level. Looking at the distribution of the unary currents before calibration, apparent is that their accuracy is determined by both random and systematic errors. Nevertheless, the self-calibration corrects both errors and hence the accuracy of the unary currents is guaranteed. Note that the accuracy depends only on the size of the correction step, which is a design parameter. Before calibration, the distribution of the unary currents spans over a 3.5LSB range, with standard deviation $\sigma = 1.06LSB$, which accounts for 9 bit intrinsic accuracy. The tolerances of the CMOS process and the particular design determine the pre-calibration distribution. After the calibration, the distribution of the unary currents spans over only about 0.1LSB range, with standard deviation $\sigma = 0.03LSB$, which accounts for more than 14 bit guaranteed accuracy for all of the calibrated currents. The post-calibration distribution of the unary

currents is determined by the correction step size of the CALDACs I_{cor_lsb}. The size of the correction step with respect to the DAC LSB current is about $I_{cor\_lsb} \approx 0.12 \cdot I_{LSB}$.

The empirical measurement results shown in Figure 16.2-5 confirm the theoretical expectations of Figure 10.2-4 for the post-calibration accuracy of the calibrated currents. The post-calibration distribution resembles a triangular shape, despite that the distribution of currents before calibration is different in both figures. Figure 10.2-4 assumes the theoretical uniform distribution of currents before calibration. However, the particular chip implementation introduces significant systematic errors. That is why the distribution of currents before calibration in Figure 16.2-5 features an offset (to the left). Nevertheless, the systematic errors are calibrated, too. The post-calibration distribution of currents depends only on design parameters – the correction step and the algorithm, and hence the post-calibration distribution of current depends on neither the IC fabrication process nor the particular IC implementation. This shows that using smartness in DAC designs is a step towards technology-independent implementations.

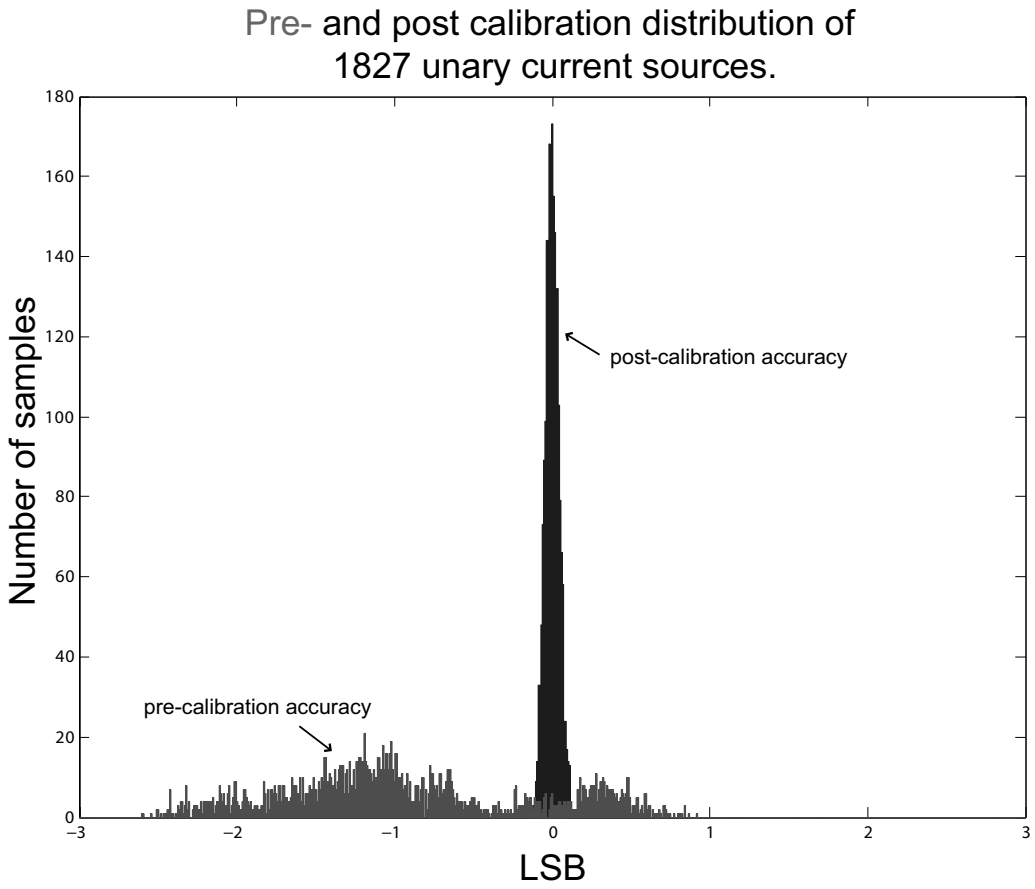


Figure 16.2-5. Distribution of the accuracy (at 12 bit level) of 1827 measured unary current sources, before and after calibration.

The calibration improves the dynamic linearity of the DAC, too. At low input frequencies, the dominant error source for the DAC performance is the mismatch of the DAC signal current sources. Figure 16.2-6 shows the DAC output frequency spectrum for input tone signal $f_{in} \approx 5MHz$ ($f_{in} = 4.97MHz$ to avoid exact ratios with F_s), sampled at $F_s = 50MHz$. The dominant harmonic distortion components are due to the current

mismatch errors. Before calibration, the SFDR is limited to about 68dB. Figure 16.2-7 shows the DAC output frequency spectrum after the calibration. The current mismatch errors are corrected and the DAC performance is improved to SFDR=81dB.

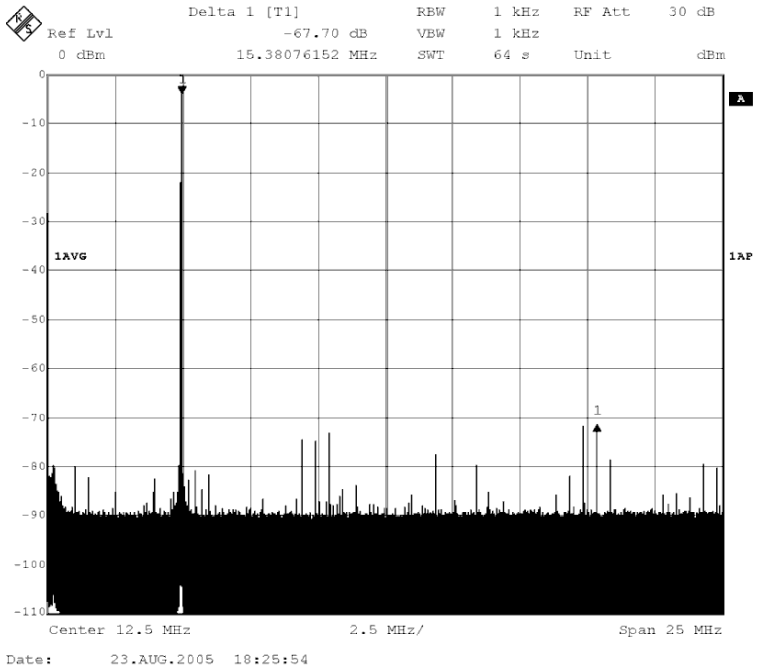


Figure 16.2-6. DAC output spectrum before calibration, SFDR=67.7dB, measured at T=25°C.

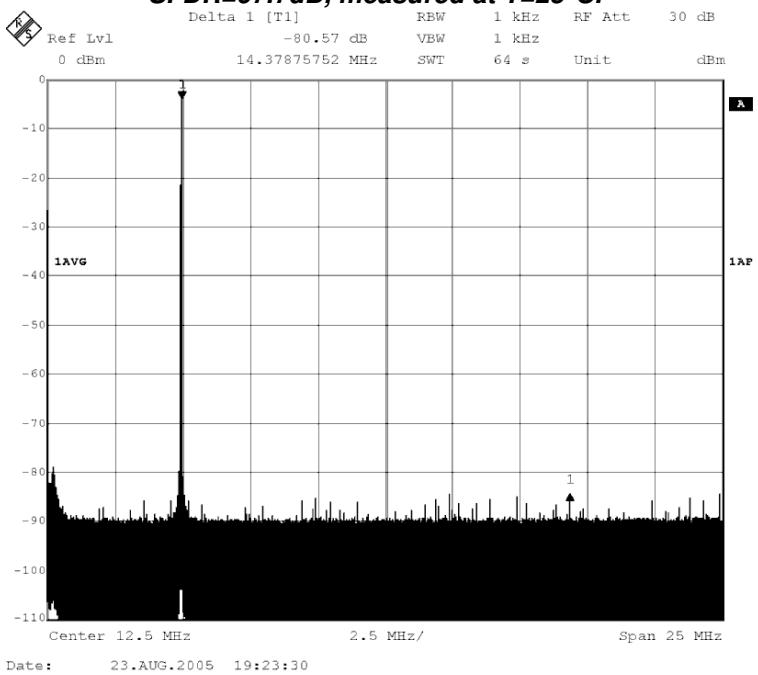


Figure 16.2-7. DAC output spectrum after calibration, SFDR=80.6dB, calibrated and measured at T=25°C.

Note that the main harmonic spurs are well suppressed. The SFDR is limited by high frequency spurs that are mainly related to both the non-calibrated binary currents and the dynamic characteristics of the particular DAC implementation. To isolate only the unary currents calibration improvement from the overall DAC dynamics and hence to estimate the potential of the calibration method, the HD2, HD3, HD4, and HD5 are evaluated against f_{in}.

Figure 16.2-8 shows HD2 thru HD5, before and after calibration. For $f_{in} < 3MHz$, the calibrated HD components are at about a level of -85dB, while the worst HD before calibration is at about a level of -66dB. Thus, the *calibration improvement is almost 20dB*, i.e. more than 3bits, which is also suggested by the static distribution of the calibrated unary currents in Figure 16.2-5. Beyond 6MHz, the dynamic errors dominate the DAC performance and the advantage of the static errors calibration is reduced. However, the overall performance is still generally slightly better after calibration. This phenomenon (seen in other publications too, e.g. [36], [27]) can be explained with the influence of the current amplitude mismatch errors on the timing and digital switching errors, as suggested in chapter 5.3.

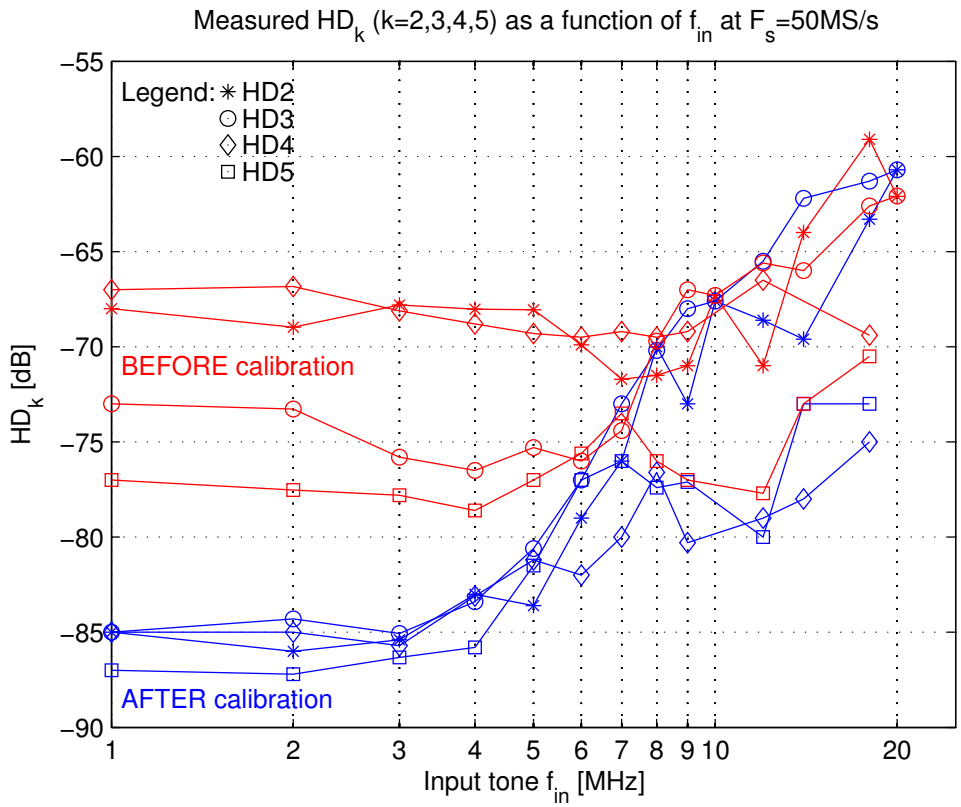


Figure 16.2-8. DAC HD components against input frequency, before and after calibration.

While for $f_{in} < 6MHz$, the DAC SFDR before calibration is limited by the first few harmonic distortion components (see Figure 16.2-6), the DAC SFDR after calibration is limited by some high frequency spurs. Along with the particular DAC realization, the non-calibrated binary currents contribute to the power of these high frequency components. A close look at the DNL and INL characteristics (see Figure 16.2-3 and Figure 16.2-4) reveals that the non-calibrated errors of the binary currents would produce high-frequency

distortion products. Figure 16.2-9 shows the DAC SFDR against the input frequency tones. The linearity of the DAC shows about 80dB for frequencies up to 5MHz.

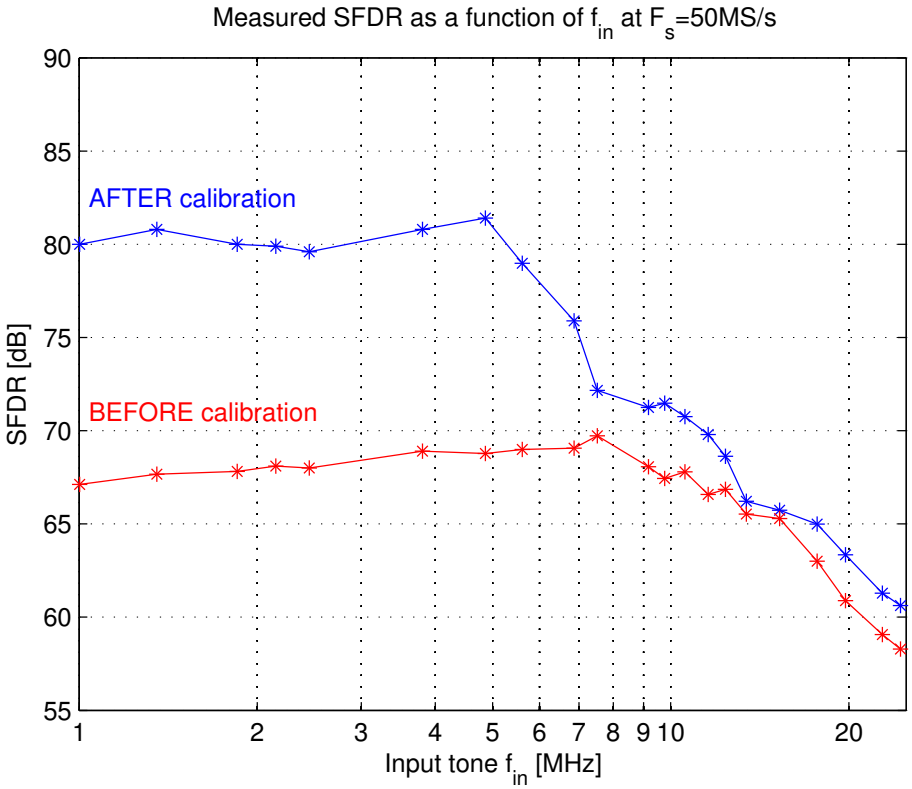


Figure 16.2-9. DAC SFDR against the input tone frequency.

16.2.3. Temperature effects

As discussed in chapter 8.5. , when the correction current is generated by a separate device that is not continuously being updated, the correction is sensitive to temperature changes. In the considered design example of a foreground self-calibration method, the main current is generated by the DAC main current source transistors and their corrections are generated by CALDACs. Both devices feature different temperature coefficients and hence the calibration results are vulnerable to temperature changes.

Generally, the foreground calibration methods execute the calibration activities only once, normally either with chip power up or when the DAC is idle. The calibration results are memorized and the consequent correction is used during the normal operation of the DAC. The change in the chip temperature provokes changes in the current amplitudes of both the DAC errors and their corrections. However, these changes are relatively different because the temperature coefficients of the main current sources and their correction currents are different. Consequently, the applied correction loses some of its effectiveness. The measurements show that the performance deterioration is about 0.12dB/^{o}C. Figure 16.2-6 and Figure 16.2-7 show the DAC output spectrum, respectively before and after calibration at temperature T=25^{o}C. Measured at T=50^{o}C without re-calibrating, the DAC SFDR drops 3dB, see Figure 16.2-10. Measured at 75^{o}C, the DAC SFDR drops another 3dB, i.e. 6dB in total, see Figure 16.2-11. Note that re-calibration is always possible, which adapts the correction to the temperature change, see Figure 16.2-12.

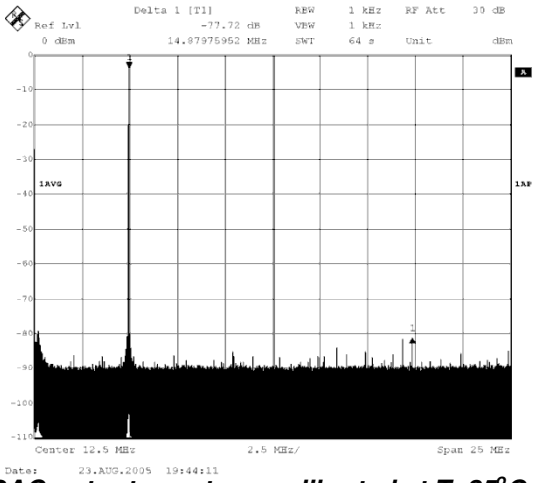


Figure 16.2-10. DAC output spectrum calibrated at T=25°C and measured at T=50°C, SFDR=77.7dB.

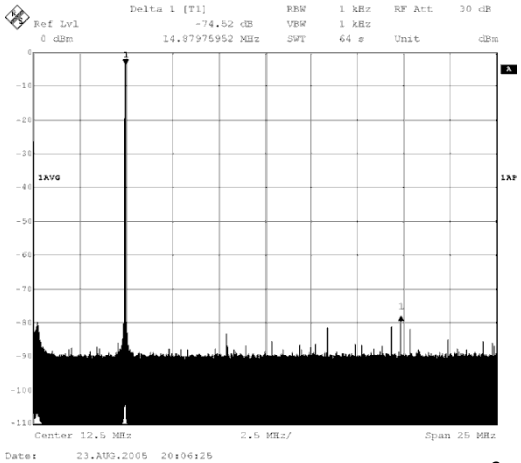


Figure 16.2-11. DAC output spectrum calibrated at T=25°C and measured at T=75°C, SFDR=74.5dB.

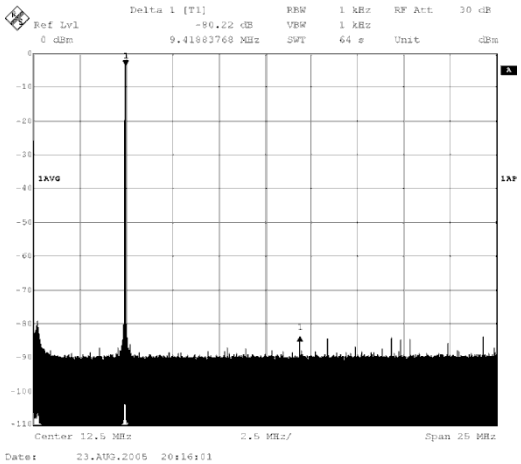


Figure 16.2-12. DAC output spectrum, calibrated and measured at T=75°C, SFDR=80.2dB.

16.3. BOTH UNARY AND BINARY CURRENTS SELF-CALIBRATION IN A 180NM DAC

Calibration of binary currents, i.e. scaled and nominally different, is conceptually described in chapter 10.4. Based on measurements of only the DAC MSB unary currents, section 16.2. suggests that calibration of the LSB binary currents can further improve the DAC performance. Recent independent research projects, e.g. [69] also suggest calibrating the DAC LSB part to further improve performance. Moreover, being able to calibrate all DAC currents decouples the design from the tolerances of the IC fabrication process. This represents a step towards IC fabrication process independent designs and facilitates migration to other IC fabrication processes.

Being able to calibrate binary currents enables the design of both efficient and accurate designs. Binary DACs occupy less area and consume less power than the respective unary implementations. The intrinsic accuracy problems of binary DACs can be greatly relaxed by calibration, as suggested in chapter 10.4.

This chapter presents the first DAC implementation that self-calibrates both unary and binary currents. It is shown that the ultimate accuracy depends on design parameters and hence not on the parameters of the IC fabrication process, since all currents are calibrated. Section 16.3.1. presents the self-calibrated DAC design. Section 16.3.2. presents chip measurements.

16.3.1. Design

The concepts of the binary currents calibration are implemented in a quad-core 12 bit DAC in an industry standard 180nm 1P6M CMOS process. The DAC is based on four parallel 12 bit sub-DAC units implemented with an algorithmic segmented architecture of 8 bit binary LSB and 4 bit unary MSB (15 unary MSB currents). The particular advantages of this flexible architecture are presented in [64]. The implementation includes full on-chip-integrated all-currents calibration, based on the algorithms shown in Figure 8.4-1, Figure 8.4-4, Figure 10.2-5 and Equation 10.4-1. The binary currents calibration method uses the four sets of binary currents of the four sub-DAC cores to implement the binary calibration algorithm. Since the four binary currents sets are available, the binary currents are not split in two sub-parts (hardware segmentation), as in Figure 8.4-3 but a standard architecture is used, as in Figure 8.3-3b.

Figure 16.3-1 shows the high-level block scheme of the implemented calibration engine. The implementation of the 1bit ADC, as a sign detector, is simple – 2 transistors and 3 inverters. The signal current cells comprise 7 NMOS-based transistors and a signed output CALDAC. The coarse signal current source M1 and the CALDAC, generating the fine correcting current, are connected in parallel. M2 is a cascode that shields M1 and the CALDAC. M4 are the mixed-signal current switches and M4b is the current switch for the self-measurement phase. M5 are switch-cascode transistors. A key difference between calibrating only unary currents (Figure 16.2-1) and calibrating both unary and binary current (Figure 16.3-1) is that the latter uses CALDACs for every DAC current.

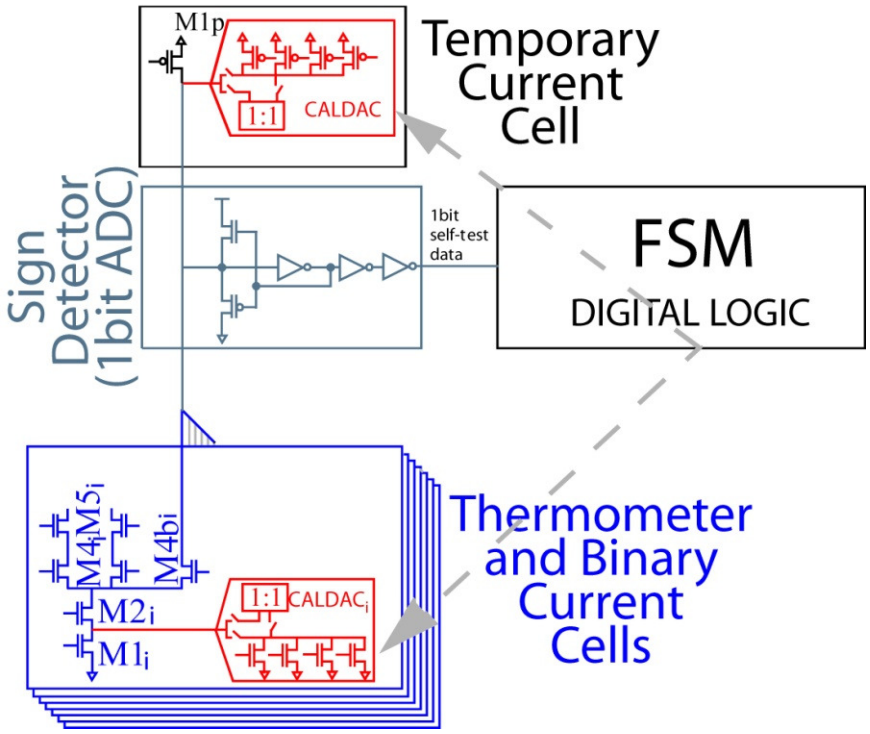


Figure 16.3-1. Self-calibration DAC scheme with unary and binary currents self-calibration.

A micrograph of the test chip is shown in Figure 16.3-2. The four sub-DACs A, B, C, and D are implemented horizontally next to each other, i.e. a unary functional segmentation is applied to the whole DAC system. Part 1 indicates the array of coarse current sources M1i from Figure 16.3-1. To save area, the sizes of M1 transistors are designed for 9 bit accuracy. Part 2 indicates the array of CALDACs. Part 3 indicates the array of cascode transistors M2, the calibration switches M3a and M3b, and the digital calibration logic. Part 4 indicates the synchronization latches implemented with Current-Mode-Logic (CML) like in [89]. Part 4 includes also the CML binary-to-unary decoder for the sub-DAC unary MSB part. Part 5 is the CML pre-processor that allocates the input digital words to the sub-DACs, acting as a de-multiplexor of the chip input. Part 6 is the input data LVDS buffers block and decoupling capacitors. Finally, part 7 is the reference current I_{ref} .

To balance between the area of the CALDACs and the achievable post-calibration accuracy of the DAC, the CALDAC LSB step and the CALDAC resolution are designed to expect worse than 9 bit intrinsic DAC accuracy and to improve it to about 14 bit DAC accuracy. The CALDACs have a resolution of 7bits (6 bits core and a sign bit). To balance between linearity and occupied area, they use a segmentation of 2 binary LSB currents and 4 unary LSB bits (15 unary currents). The CALDAC LSB step of the MSB current is nominally $I_{cor\_lsb} = 0.14 \cdot I_{LSB}$ (given relatively to the DAC LSB current), but for the sake of test it is made externally adjustable. The LSB steps of the rest of the CALDACs scale down by a factor of $\sqrt{2}$ for each bit, as the CALDACs should cover a smaller mismatch range but also should provide a smaller calibration error to compensate for the inherited calibration errors from the previous calibration steps. For the LSB bits from 1 to 4, scaling is not applied because their DAC accuracy requirements are sufficiently low.

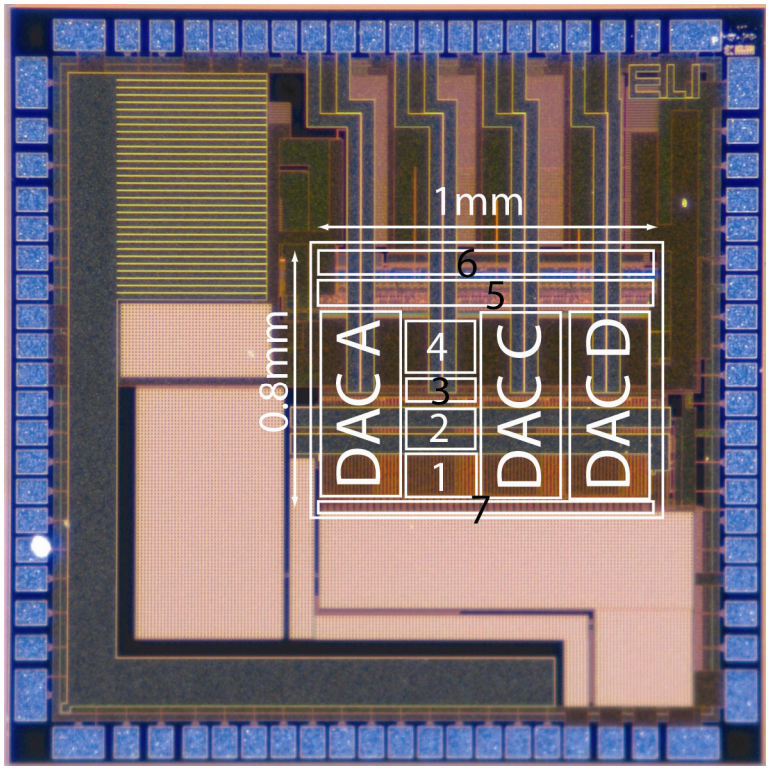


Figure 16.3-2. Micrograph of the flexible self-calibrated quad-core DAC.

The overall DAC area is only 0.8mm^2, hence 0.2 mm^2 per 12 bit sub-DAC unit. Comparatively to the literature, it is one of the smallest published designs. The area of the DAC signal current sources is small, because they are not designed for extreme matching. Their design is relaxed and it takes into account that the DAC calibration engine can improve their accuracy to more than 14 bit. In addition, the DAC implements an algorithmic segmentation with a large portion of binary LSB bits – 8. This reduces the overall area, while the non-linearity drawbacks of the binary bits are answered with the calibration.

16.3.2. Measurements

The chip power consumption consists of three parts: start-up (calibration), digital (data processing), analog (output signal) power consumption. The power consumption for the calibration is practically zero, because the calibration is run once at chip start-up and the results are memorized; during normal DAC operation the CMOS calibration logic is not active. Regarding the DAC analog signal output power, the presented results are measured for 24mA full-scale current, i.e. 6mA per sub-DAC, terminated with a 50Ω differential load resistance. The full digital data processing power consumption of the flexible DAC is 118.8mW at 1.8V supply (66mA current). The distribution of the digital power consumption is 27mW per sub-DAC and 10.8mW for the pre-processor. Thus, the overall DAC power consumption is flexible from 37.8mW (only one sub-DAC is used and the rest are turned off) to 118.8mW (all sub-DACs operate simultaneously). The digital circuits of the DAC core are implemented with CML gate and hence the power consumption does not depend on Fs.

An exemplary (and representative) characterization of the DAC static performance is shown in the INL and DNL plots of Figure 16.3-3 and Figure 16.3-4.

a) INL before calibration: b) INL after calibration:

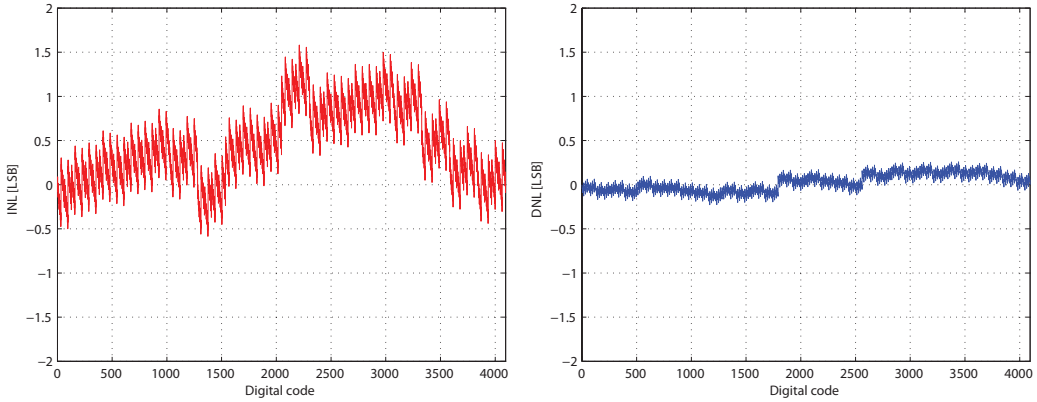


Figure 16.3-3. Measured INL of 4 parallel sub-DACs: a) before calibration; b) after calibration.

Figure 16.3-3 shows the DAC INL before (left, INL_{max}=1.5LSB) and after (right, INL_{max}=0.2LSB) calibration. The improvement in DAC linearity is more than 3 bits. Note the small non-linearity contribution of the LSB binary part. Before calibration, the binary currents contribute about 1 LSB deviation in the DAC INL. After the calibration, the non-linearity of the binary part is reduced almost 10 times. This is the main difference between the presented measurements of the 250nm (only unary currents calibration, section 16.2.) and the 180nm (both unary and binary currents calibration) test-chips. While in Figure 16.2-3b, the static DAC post-calibration linearity is limited by the non-calibrated binary bits, in Figure 16.3-3 they are calibrated and they contribute only little to the DAC post-calibration non-linearity. Note also that the discussed example of Figure 16.2-3b has 6 binary LSB bits, while the example of Figure 16.3-3 has 8 binary LSB bits. Nevertheless, the calibration improves their accuracy to a more than 14 bit level.

The DNL plots further demonstrate the importance of binary currents calibration. Figure 16.3-4 shows the DAC DNL plots before (left, DNL_{max}=0.7LSB) and after (right, DNL_{max}=0.2LSB) calibration. Before calibration the binary LSB bits have about 0.5LSB DNL non-linearity. After calibration, their contribution to INL is about 0.1LSB. The non-linearity after the calibration is due to the post-calibration accuracy of the currents. Note that if only the unary currents are calibrated (Figure 16.2-4), the DNL_{max}=0.6LSB is limited by the non-calibrated binary currents, while in Figure 16.3-4, DNL_{max}=0.2LSB is determined by a calibrated unary current. Furthermore, the DNL_{max} limitation of Figure 16.3-4 is limited by the CMOS process parameters. The 250nm DAC performance (section 16.2.) depends on both design parameters (for the calibrated unary currents) and CMOS process parameters (the intrinsic accuracy of the binary currents). However, the presented 180nm DAC test-chip calibrates all the currents and hence its linearity is limited only by design parameters. For these measurements, the size of the LSB correction step I_{cor_lsb} and the resolution the CALDACs 7 bit (which sets the full-scale range) completely determine the post-correction accuracy.

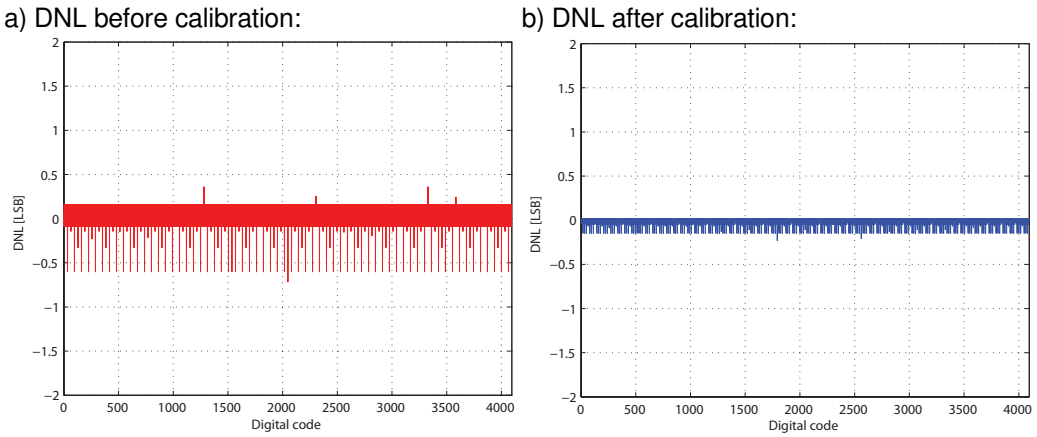


Figure 16.3-4. Measured DNL of four parallel sub-DACs: a) before calibration; b) after calibration.

Figure 16.3-5 shows the measured DNL_{max} and INL_{max} for different sizes of the LSB correction step, given relatively to the LSB current of the DAC. This is the size of the LSB correction step of the unary CALDAC. The correction steps of the binary CALDAC scale down by a factor of $\sqrt{2}$. Both DNL_{max} and INL_{max} depend on the size of I_{cor}. Small I_{cor} guarantees small correction error and hence high post-calibration DAC accuracy. However, if I_{cor} is too small the full-scale range of the CALDAC cannot cover the errors of both the calibrated DAC currents and the measurement device. Thus, some currents cannot be completely calibrated and their post-correction errors deteriorate the DAC accuracy, as shown in Figure 16.3-5 for I_{cor} approaching small values.

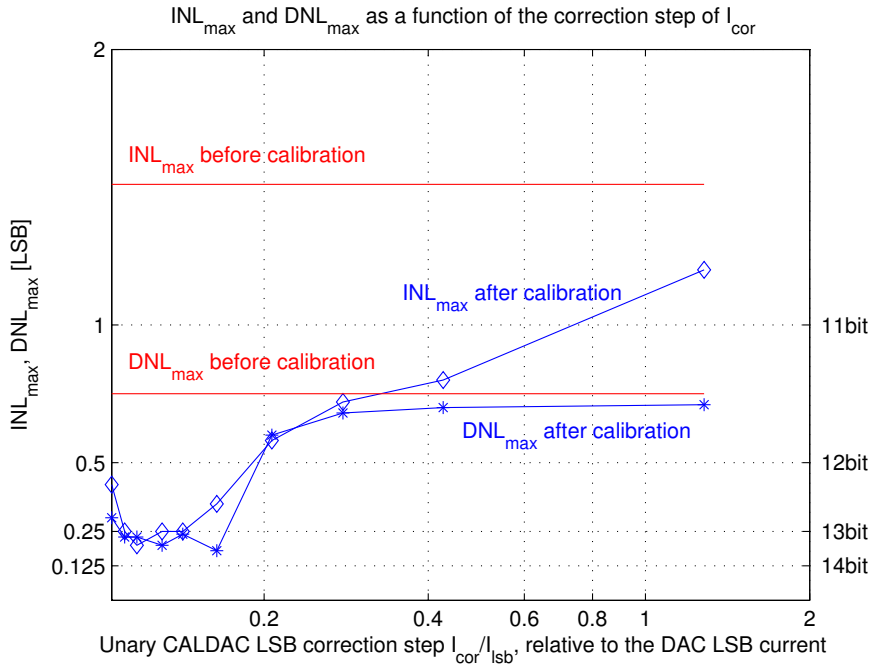


Figure 16.3-5. Measured INL_{max} and DNL_{max} against the relative size of the correction step of the unary CALDAC.

To characterize the dynamic performance of the calibration engine, dynamic measurements are done at a sampling rate of F_s=50MS/s. Figure 16.3-6 shows the DAC SFDR performance for a single tone input as a function of the input signal frequency f_{in}. The measurements show that the calibration of the DAC current sources improves the DAC performance (solid line in Figure 16.3-6). For static and low f_{in} signals, the DAC figures INL and DNL demonstrate the calibration improvements for both unary and binary currents. For frequencies up to 11MHz, the predominant problem is the mismatch of the DAC currents. Calibration improves the DAC SFDR performance to above 80dB. At f_{in}=11MHz, SFDR=80dB is still maintained. Beyond 11MHz the dynamic error mechanisms overshadow the calibration advantages. However, the performance after calibration remains improved in comparison with the DAC performance before calibration. This phenomenon (seen in other publications too, e.g. [36], [27], and in Figure 16.2-9 for the calibration of only the unary currents) can be explained with the influence of the current amplitude errors on the timing errors, as suggested in chapter 5.3. The improvement beyond 11MHz depends on the actual t_{slew} as suggested by Figure 5.3-3. Figure 16.3-7 and Figure 16.3-8 show the DAC output spectrum respectively before and after calibration.

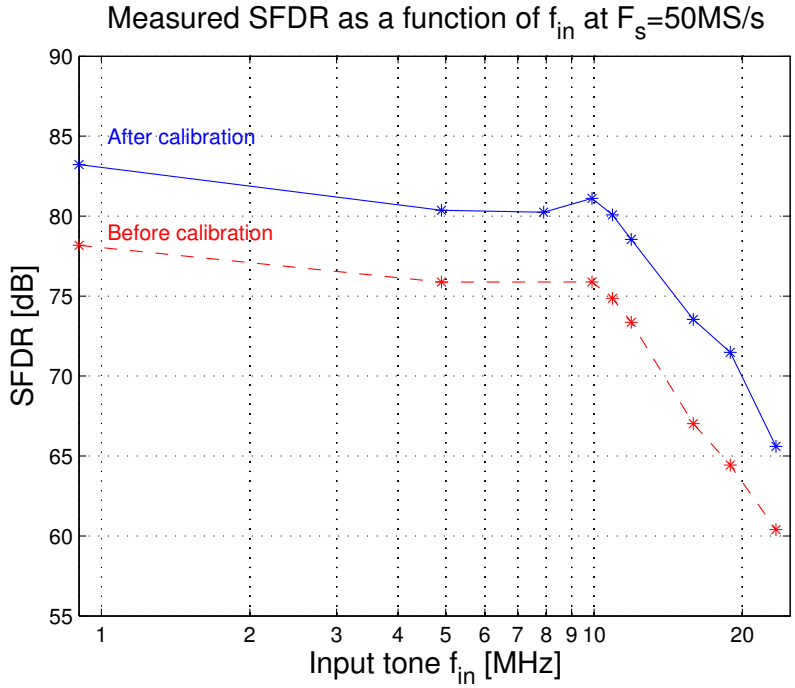


Figure 16.3-6. SFDR performance against f_{in}.

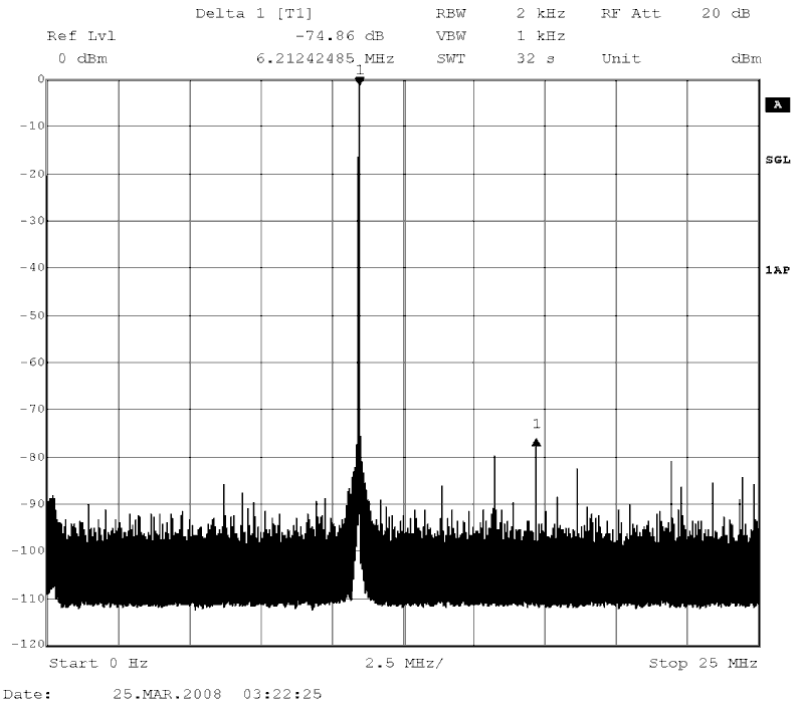


Figure 16.3-7. Output spectrum for f_{in} 11MHz@50MS/s, before calibration. SFDR=74.8dB.

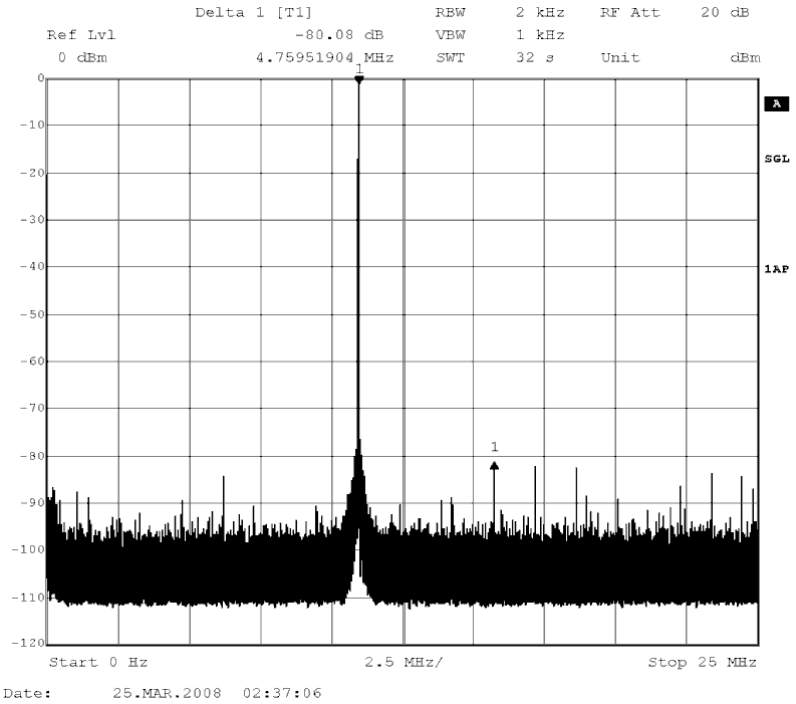


Figure 16.3-8. Output spectrum for f_{in} 11MHz@50MS/s after calibration, SFDR=80dB.

To isolate the advantages of the calibration from the dynamic characteristics of the particular DAC implementation, the harmonic distortion components of the DAC are evaluated for low input frequencies f_{in}=1MHz. For these frequencies, the first few harmonic distortion components in the DAC output spectrum are highly determined by the mismatch between the DAC signal currents. Figure 16.3-9 and Figure 16.3-10 show DAC output spectrums for f_{in}=1MHz. Before calibration, HD2 is about -73dB and HD3 is about -80dB. After calibration, both HD2 and HD3 are less than -90dB and the DAC SFDR is limited by some higher frequency spurs due to causes beyond the scope of the mismatch calibration, e.g. DAC dynamics, various glitches, imperfections of the measurements setup. Figure 16.3-9 and Figure 16.3-10 show the potential of the DAC self-calibration engine. However, to achieve a DAC SFDR of more than 90dB, many factors need to be taken into account. The calibration of the DAC current is one of them. The post-calibration SFDR shown in Figure 16.3-10 differs from the SFDR plot shown in Figure 16.3-6, because these two are measurements of different chip samples.

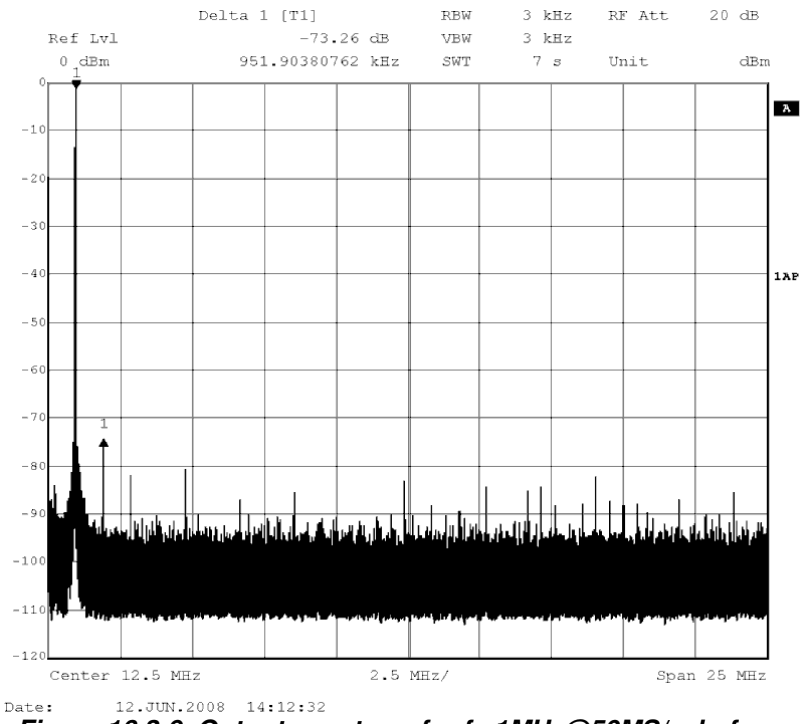


Figure 16.3-9. Output spectrum for f_{in} 1MHz@50MS/s, before calibration. HD2=-73.26dB.

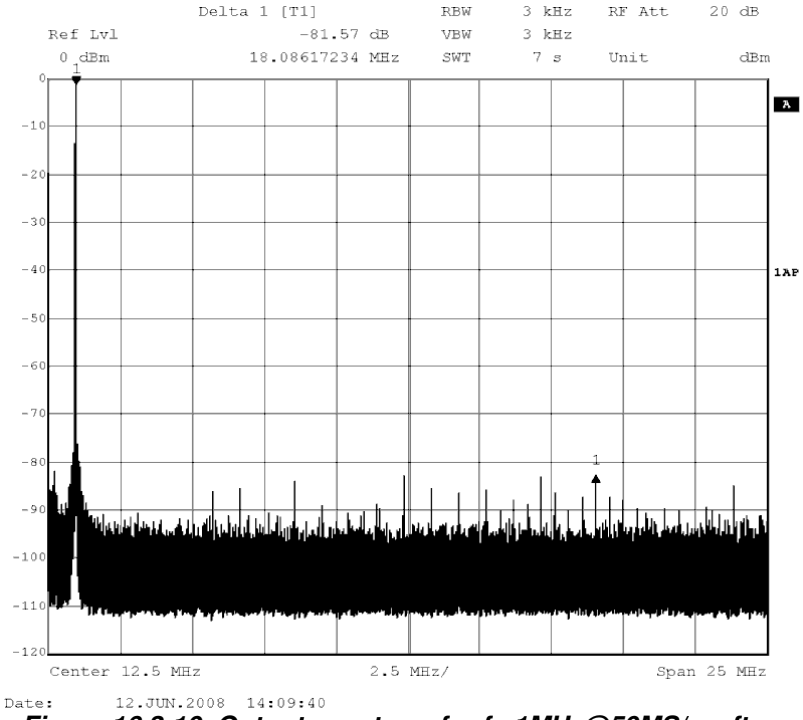


Figure 16.3-10. Output spectrum for f_{in} 1MHz@50MS/s, after calibration. HD2<-90dB.

16.4. COMPARISON WITH STATE-OF-THE-ART DAC PUBLICATIONS

Table 16.4-1 compares this work with a few DAC chips selected from the literature. The first part of the table contains DAC chips that are assisted in performance by various correction methods. The second part of the table contains DAC chips that are not assisted by calibration, i.e. intrinsic DACs.

The general impression is that the calibrated DACs can achieve better resolution and better linearity for slower signals, while the intrinsic DACs achieve better linearity for higher speeds. In addition, the overall occupied area of the calibrated DACs is smaller than the area of the intrinsic DACs. Looking at the calibrated DACs, most of the works achieve SFDR performance above 80dB but hardly maintain it for high f_{in} , while this work maintains the advantages of calibration up to $f_{in} = 11MHz$. Another highlight of this comparison is the small occupied silicon area of the presented DAC – only 0.2mm^2 per sub-DAC core. Note that this is achieved mainly due to the large LSB binary portion – 8 bits combined with the ability to correct the binary currents mismatch errors.

Table 16.4-1. Comparison between selected DAC examples, both calibrated and intrinsic, and the presented implementations of this work.

Calibrated DACs	Back-ground calibration [69]	Background calibration [66]	Calibration via V_{gs} [70]	A parallel DAC calibration [36]	**This work, unary currents calibration, 250nm test-chip**	**This work, unary and binary calibration, 180nm test-chip**
Input bits	13 bit	14 bit	14 bit	14 bit	12 bit	12 bit
Static accuracy	13bit	>14bit	14bit	14 bit	>12bit	13bit
Min[HD2, HD3] at 1MHz	-83.7dB	-84dB	-64dB	-82dB	-85dB	-90dB
Best SFDR point	75dB@ 10MHz	82dB@ 8.5MHz	84dB@ 0.1MHz	82dB@ 0.9MHz	81dB@5MHz	80dB@11MHz
Technology	130nm	350nm	180nm	130nm	250nm	180nm
Area [mm^2]	1.0	11.83	1.0	?? (external ADC and algorithm)	1.14	4x0.2=0.8
Complete calibration?	No	No	No	No	No	**Yes**
Intrinsic DACs	[17]	[18]	[12]	[19]	[89]	[9]
Resolution	8 bit	8 bit	10 bit	14bit	12bit	10bit
Static accuracy	> 9 bit	9 bit	11 bit	12bit	10bit	12bit
Min[HD2, HD3] at 1MHz	-50dB	-55dB	-71dB	?	-78dB	-74dB
Best SFDR point	?	58.9dB@ 0.42MHz	71dB@ 15MHz	67dB@ 270MHz	70dB@ 120MHz	60dB@ 122MHz
Technology	2000nm	1000 nm	350nm	180nm	180nm	180nm
Area [mm^2]	3.4mm^2	**0.5mm^2**	**1mm^2**	6.2mm^2	1.13mm^2	**0.35mm^2**

16.5. CONCLUSIONS

Two 12 bit DAC test-chips with completely integrated calibration engines are realized in 250nm and 180nm standard CMOS processes. The calibration engines use circuits with relaxed design requirements. Small calibrating DACs (CALDACs) are attached in parallel to every current source for calibration. A current comparator (1 bit ADC, a sign detector) is used for self-measurements. The algorithms that control the process of calibration are simple. They are fully on-chip integrated thanks to their efficient implementation as finite-state-machines (FSMs).

The 250nm DAC test-chip calibrates the MSB unary currents to a 14 bit accuracy level. The self-calibration apparatus improves the 9bit intrinsic linearity of the DAC core to more than 12 bit, i.e. INL<0.4LSB and SFDR>81dB at 5MHz signal frequency and 50MS/s sampling rate. The measurement results suggest calibrating the DAC binary currents to further improve the DAC performance.

The 180nm DAC test-chip calibrates all current sources, i.e. both unary and binary currents. This is the first presented DAC realization that calibrates all binary currents, achieving DAC accuracy independence of the IC fabrication CMOS process. The DAC implements a new calibration algorithm that uses the binary currents of the multiple sub-DACs to calibrate all binary and unary currents to a common reference. After calibration, SFDR>80dB is measured for an input tone of up to 11MHz.

The calibration of all currents and the hence the inherent process independence significantly improves the chip yield, optimizes the occupied silicon area, and facilitates the design portability to future technologies.

17. Chapter

A FUNCTIONAL-SEGMENTATION DAC DESIGN USING HARMONIC DISTORTION SUPPRESSION METHOD

This chapter presents a validation in practice for the new DAC correction method for suppression of HD, which is presented in chapter 13. An 180nm CMOS test-chip implementation and a measurements setup are presented. The test-chip design uses functional unary segmentation of four 12 bit sub-DACs. Measurement results are reported for two different implementations of the method. The first implementation uses a single 12 bit conventional DAC to emulate two virtual sub-DACs, converting $\frac{\pi}{3}$ phase-shifted signals. Measurement results show more than 10dB improvement of the DAC linearity. The second implementation of the method uses two real 12 bit sub-DACs, converting $\frac{\pi}{3}$ phase-shifted signals. Measurement results show SFDR=80dB for f_{in}=16.9MHz. For OFDM-like multi-tone measurements, the measured improvement of the DAC linearity is about 8dB.

17.1. INTRODUCTION

A new DAC correction method for suppression of DAC HD is presented in chapter 13. According to the classification of chapter 7, this is a method that: belongs to the STFC (signal-transfer function compensation) group; uses no self-measurement; uses intrinsic redundancy, and is a high-level method. That is to say: it does not require error measurements; it uses DAC core own resources for correction; it features reduced dependence on the DAC algorithmic architecture (i.e. the intrinsic DAC core can be designed independently of the correction method).

In brief, the method uses parallel D/A conversion paths that process phase-shifted replicas of the input signal. Setting correct phase-delays of these replicas and summing the output signals of the parallel branches can result in suppression of the DAC harmonic distortion and hence improve linearity of the overall system.

A recent independent research project [61] validates a conceptually similar method in the field of RF transmitters/receivers. However, in the area of DACs, the implementation significantly differs, since the input signal is digital, the processing is in mixed-signal domain (operating with sampled data) and the output signal is analog.

This chapter further presents a measurement validation of the method for suppression of HD in DACs. Section 17.2. presents the test set-up design. Section 17.3. presents measurement results that validate the method using a single conventional DAC, emulating two virtual interleaved parallel sub-DACs. Effectively, a method is suggested to exchange signal bandwidth for signal linearity using a conventional DAC design. Section 17.4. presents measurement results of two real parallel sub-DACs. Section 17.5. presents measurement results for an exemplary application – an OFDM system. Finally, conclusions are drawn in section 17.6.

17.2. TEST SET-UP DESIGN

To verify the method for suppression of HD in DACs, a test chip is designed in an industry-standard 180nm CMOS process. The test system is based on functional unary segmentation into four parallel instances of a 12 bit current-steering sub-DAC and a flexible measurement board (PCB). In the test-chip, a digital pre-processor de-multiplexes in time domain the input data among the sub-DAC instances. The analog post-processor is implemented on the PCB. The high-level architecture of the parallel DAC system, the test board (printed-circuit board), and the measurement setup for the method of suppression of harmonic distortion (HD) components in DACs is presented in Figure 17.2-1 (see also Figure 13.2-1 for a general conceptual scheme, and [64] for the original publication of the presented implementation). Figure 17.2-2 shows the layout and a micrograph of the manufactured test-chip. The active chip area is about $1mm^2$.

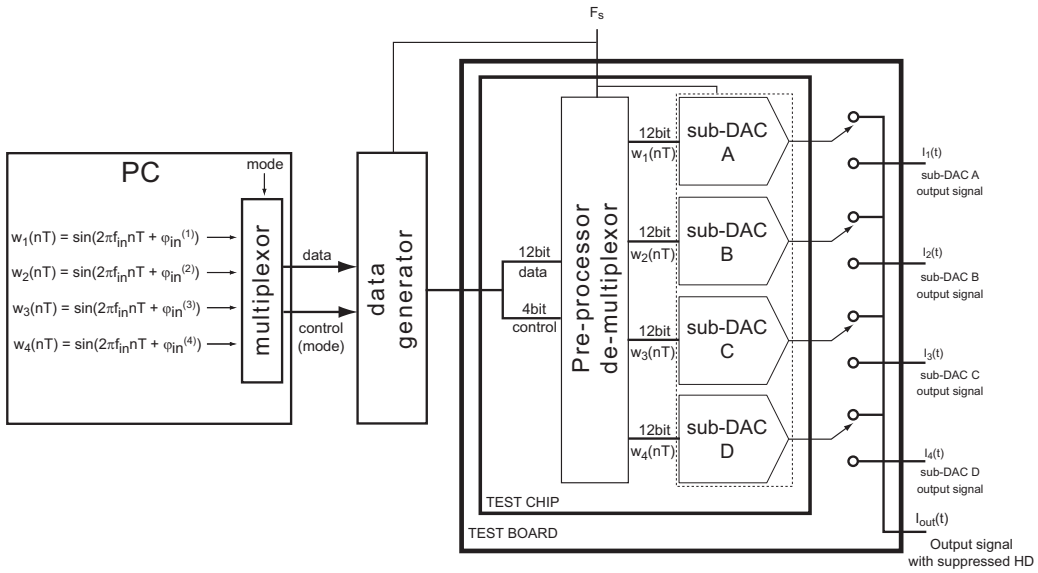


Figure 17.2-1. High level scheme of the test-chip, test board, and measurement setup for validating the method for suppression of harmonic distortion.

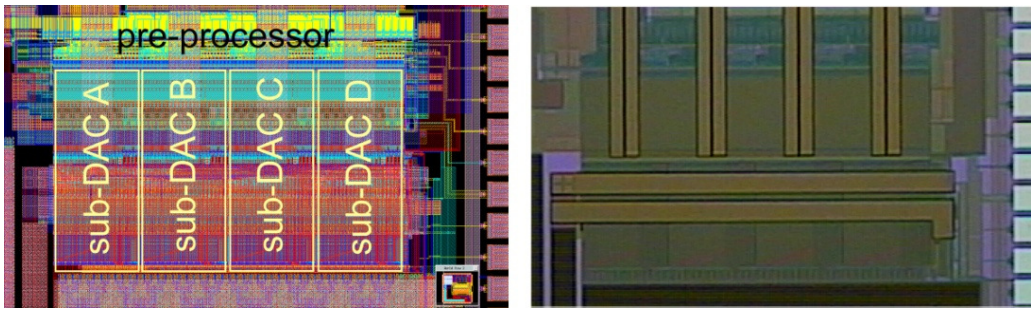


Figure 17.2-2. Layout and micrograph of the 180nm CMOS test chip DAC using four 12 bit sub-DACs.

In the various operation modes, the sub-DACs operate either separately or collectively. When the test-chip is used as a single DAC, two or more parallel virtual sub-DACs can realize the method for suppression of HD components. When the test chip is used as multiple independent sub-DACs, two or more real sub-DACs can realize the method for suppression of HD components.

17.3. PARALLEL VIRTUAL DACS

A single conventional DAC can emulate the operation of multiple parallel virtual DACs by trading some of its operational bandwidth. To achieve this, the input digital code should contain the codes for the virtual sub-DACs in a time interleaved way. For example, to emulate two parallel virtual sub-DACs, the input DAC codes are constructed by interleaving in time the input codes for both sub-DACs, as shown in Figure 17.3-1.

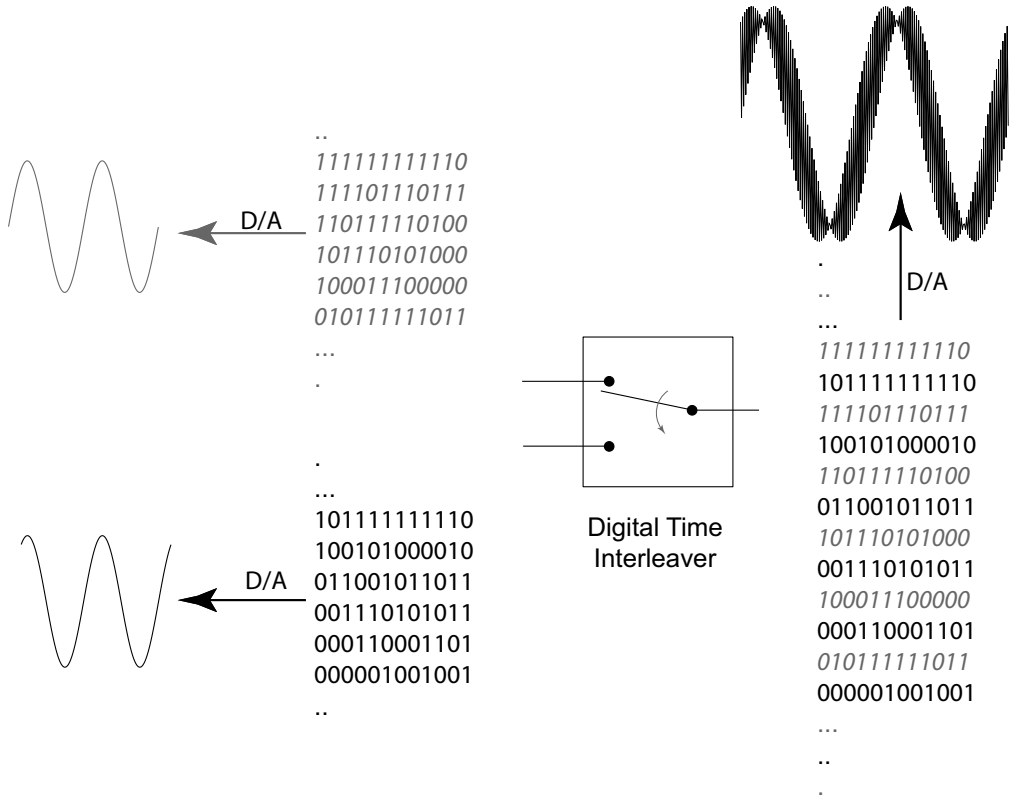


Figure 17.3-1. Digital input code construction to emulate two input signal phase shifted sub-DACs.

At every odd clock period, the digital code for the first sub-DAC is available and respectively at every even clock period the digital code for the second sub-DAC is available. The resulting DAC contains the virtual outputs of the two virtual sub-DACs and its effective sampling frequency $F_{s'}$ is 2 times lower than the digital input sampling frequency F_s, i.e. $F_{s'} = 0.5F_s$. Thus, the two virtual parallel sub-DACs can convert input phase-shifted signals to suppress the DAC HD spurs trading off a reduction of the useful signal band. Figure 17.3-2 plots measurement results of HD3 and SFDR for a normal DAC and a DAC emulating two parallel virtual DACs converting $\pi/3$ input phase-shifted signals. The measurement results confirm the theoretical Equation 13.2-2. For input frequencies up to $F_{s'}/6$, i.e. $f_{in} \le F_s/12$, HD3 is suppressed by more than 10dB and SFDR is no longer limited by HD3. For the frequency band from $F_{s'}/6$ up to $F_{s'}/3$, the HD3 from the first image band is located in the Nyquist band. This harmonic spur is not suppressed due to the difference in the virtual sampling moments, i.e. $\varphi_s^{(1)} = \varphi_s^{(2)} + \pi$. It is important to note here that in this method all odd image bands are in an "anti-phase" with the signal band, because of the phase delayed 0.5T sampling. The effects of this can be illustrated as follows. The Dirac pulses for the second virtual sub-DAC are expressed relatively to the first sub-DAC as:

Equation 17.3-1

$$\delta_{T2}(\omega) = \delta_{T1}(\omega) \times e^{-j\omega\frac{T}{2}} = \frac{2\pi}{T}\sum_{k=-\infty}^{\infty}\delta\left(\omega - \frac{2\pi k}{T}\right) \times e^{-j\omega\frac{T}{2}} = \frac{2\pi}{T}\sum_{k=-\infty}^{\infty}\delta\left(\omega - \frac{2\pi k}{T}\right) \times e^{-j\pi k},$$

where $\delta_{T1}(\omega)$ and $\delta_{T2}(\omega)$ are the Dirac sampling pulses for respectively the first and second virtual sub-DACs, $e^{-j\pi k} = \cos(k\pi) - j\sin(k\pi)$, T is the equivalent sampling period. Thus, for all even k, $e^{-j\pi k} = 1$ and for all odd k, $e^{-j\pi k} = -1$. That is why both virtual sub-DACs are "in-phase" within all even image bands (including the signal band), while they are in "anti-phase" within all odd image bands (where the most important is the first image band k=1). Thus, for the frequency band from $F_{s'}/6$ up to $F_{s'}/3$, the sub-DAC 2 HD3 (i.e. IM3) from the first image band and the sub-DAC 1 HD3 (i.e. IM3) from the first image instead of cancelling each other, fully sum up and yield worse overall HD3 (i.e. IM3) than without the method, as shown in jump of HD3 (plotted as single graph, combining HD3 and IM3, which is frequently referred as the "folded" HD3) in Figure 17.3-2. The same result can be derived from Equation 13.2-2 for the two virtual sub-DACs:

Equation 17.3-2

$$\begin{cases} I_1(j\omega) = \sum_{p=-\infty}^{\infty}\sum_{k=1}^{\infty} A_{k,p} e^{j(k\omega_{in}+p\omega_s)t} \\ I_2(j\omega) = \sum_{p=-\infty}^{\infty}\sum_{k=1}^{\infty} A_{k,p} e^{j((k\omega_{in}+p\omega_s)t+p\pi)} e^{jk\frac{\pi}{3}} \end{cases}$$

For the combined output:

Equation 17.3-3

$$I_{out}(j\omega) = I_1(j\omega) + I_2(j\omega) = \sum_{p=-\infty}^{\infty}\sum_{k=1}^{\infty} A_{k,p} e^{j(k\omega_{in}+p\omega_s)t} + \sum_{p=-\infty}^{\infty}\sum_{k=1}^{\infty} A_{k,p} e^{j(k\omega_{in}+p\omega_s)t} e^{\left(jk\frac{\pi}{3}+p\pi\right)},$$

For all even p, the factor $e^{\left(jk\frac{\pi}{3}+p\pi\right)} = e^{\left(jk\frac{\pi}{3}\right)}$ and for all odd p $e^{\left(jk\frac{\pi}{3}+p\pi\right)} = -e^{\left(jk\frac{\pi}{3}\right)}$.

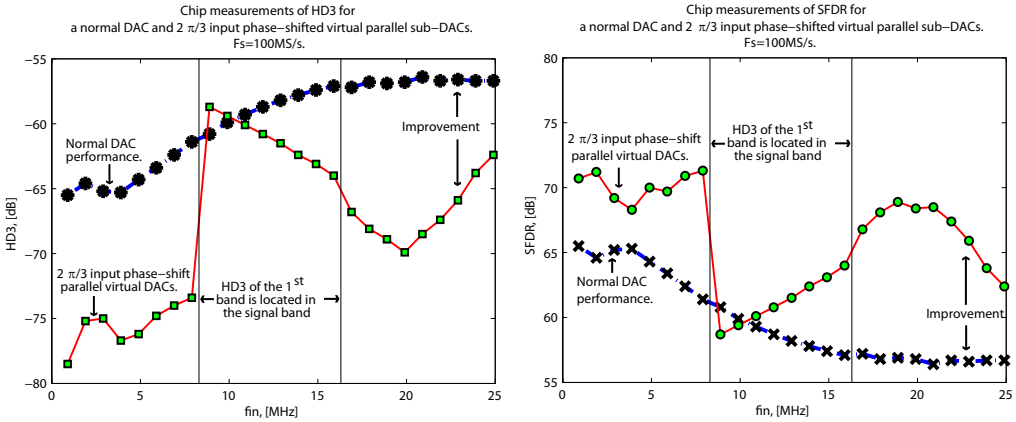


Figure 17.3-2. Measurement results of HD3 (left) and SFDR (right) vs. f_{in} for a normal DAC (blue curve with stars and crosses) and 2 input $\pi/3$ phase shifted virtual DACs (red curve with squares and circles).

Thus, for the frequency band from $F_{s'}/3$ up to $F_{s'}/2$, the HD3 is suppressed too (p=2), though the measured suppression effectiveness is reduced due to extra error

mechanisms, e.g. the increased influence of the random timing errors. The measured DAC output spectrum for $f_{in} = 7.9MHz$ is shown in Figure 17.3-3 and Figure 17.3-4.

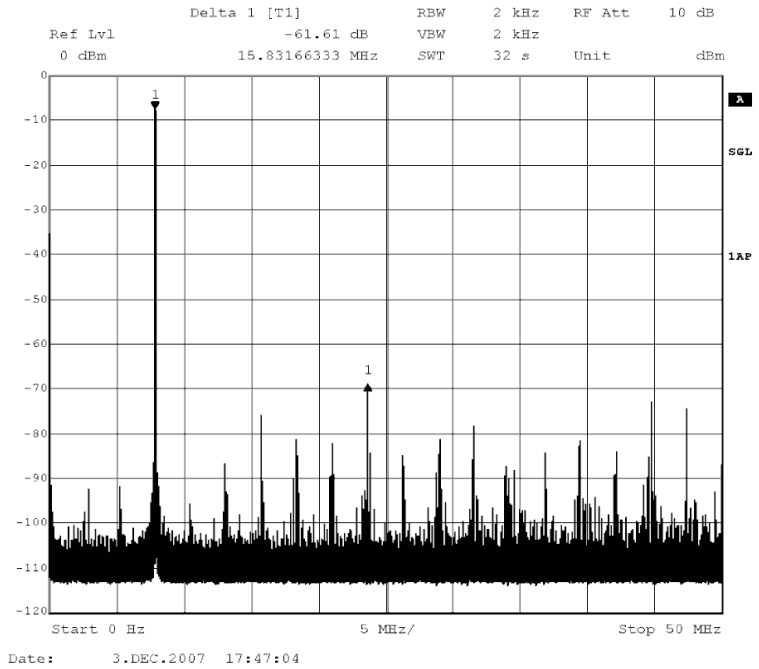


Figure 17.3-3. Measured frequency spectrum of a normal DAC operation for $f_{in} = 7.9MHz$ and $F_s = 100MS/s$.

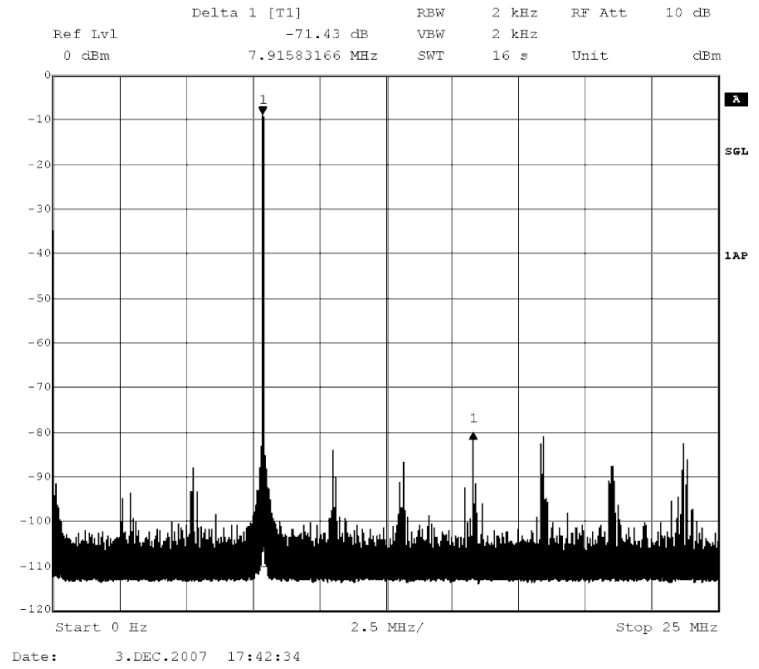


Figure 17.3-4. Measured frequency spectrum of 2 input $\pi/3$ phase shifted virtual DACs for $f_{in} = 7.9MHz$ and $F_s = 100MS/s$.

17.4. PARALLEL REAL SUB-DACS

As shown in section 17.3. , the sub-DACs need to be synchronized, in order to suppress the image harmonic distortion spurs as effectively as those from the Nyquist band. Therefore, DAC systems having actual real parallel sub-DACs can implement the method more efficiently.

To demonstrate the method of suppression of HD, measurements are done for two parallel synchronized sub-DACs that convert $\pi/3$ phase-shifted replicas of the input signal. Figure 17.4-1 plots a single-tone output spectrum of the complete DAC system for $f_{in} = 16.9MHz$. The first several HD components are also indicated. Their powers follow the theoretical prediction of Figure 13.2-2. Since HD6 is intrinsically small, its increase does not limit the SFDR. Figure 17.4-2 shows a SFDR comparison between the intrinsic single DAC performance and two parallel sub-DAC converting $\pi/3$ phase shifted replicas of the input signal frequency f_{in}. The method of suppression of HD delivers SFDR = 80dB for $f_{in} = 16.9MHz$, while the individual sub-DAC performance is around 65dB.

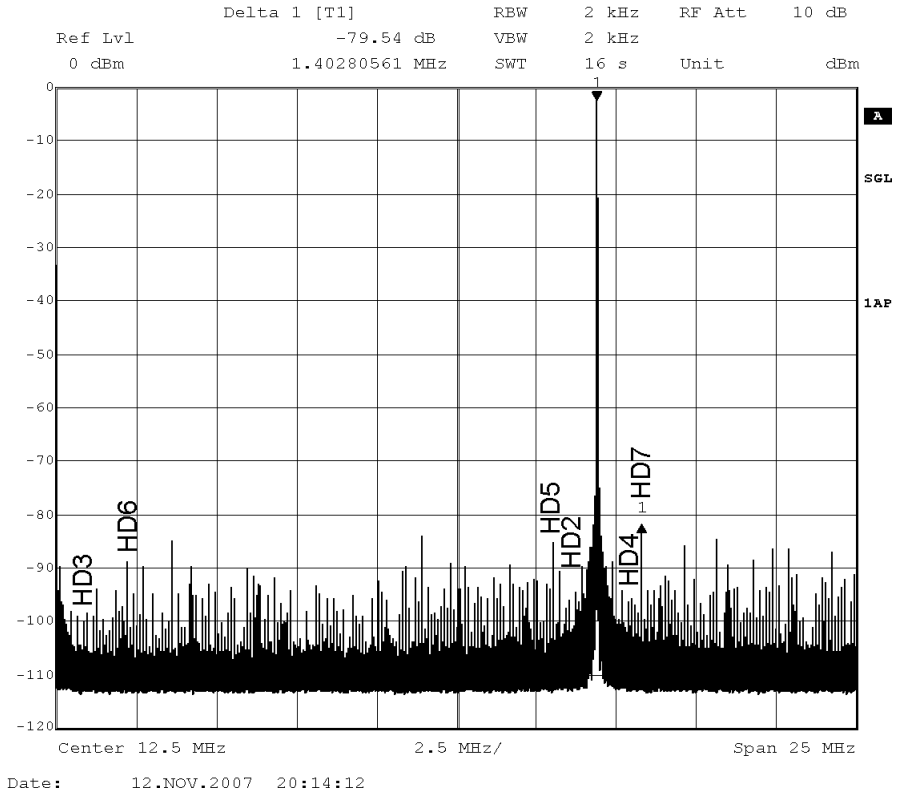


Figure 17.4-1. Frequency spectrum of the combined sub-DAC output for $f_{in} = 16.9MHz$ ***and*** $F_s = 50MS/s$ ***.***

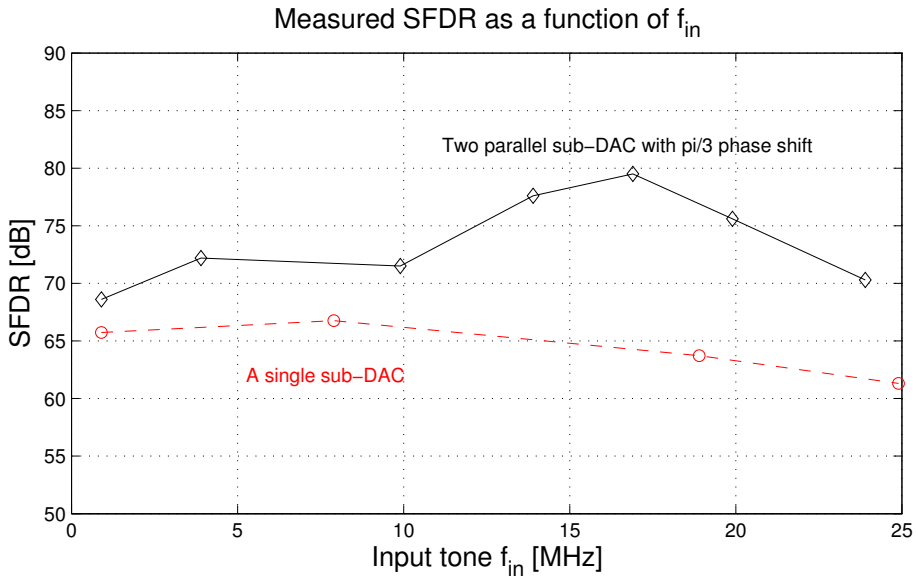


Figure 17.4-2. Comparison of the SFDR DAC performance of the $\pi/3$ parallel phase-shifted input DACs and a single sub-DAC.

17.5. OFDM (MULTI-TONE) SYSTEM APPLICATION

The digital implementation of phase-shifting is simple when the signal is available in the complex domain. Therefore, OFDM systems are excellent candidates for the presented method. Moreover, the OFDM signal measurements, i.e. multi tone input signal measurements, give a (subjectively) more realistic and practical representation of the DAC linearity than the single tone measurements. In the OFDM signal packet a number of frequency bins are present and their harmonic distortion components inter-modulate, “polluting” the spectrum and reducing the useful dynamic range. Figure 17.5-1 (suppression of HD) and Figure 17.5-2 (single sub-DAC) show the D/A conversion of an exemplary OFDM signal packet without and with the presented method. The improvement of the in-band SFDR is more than 8dB.

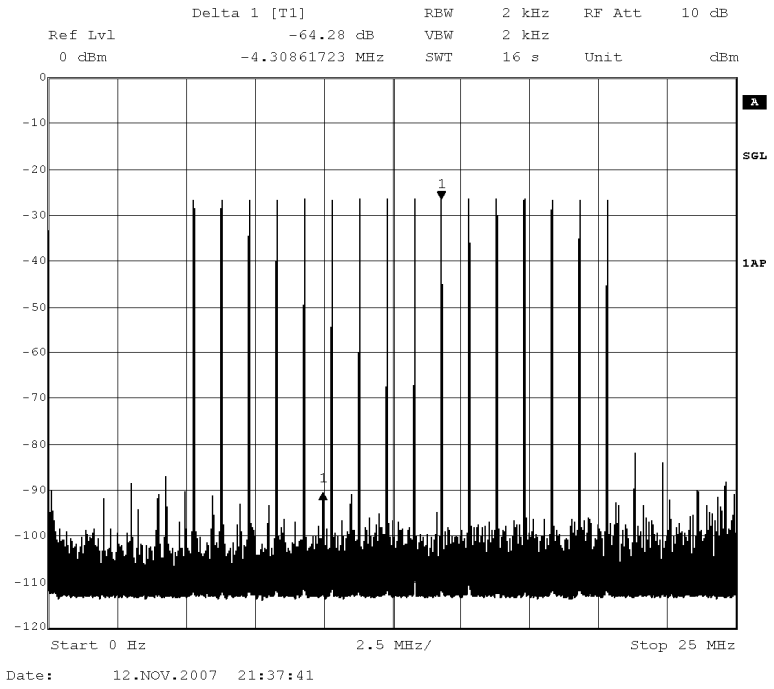


Figure 17.5-1. Multi-tone signal measurement results for 2 input $\pi/3$ phase shifted sub-DACs.

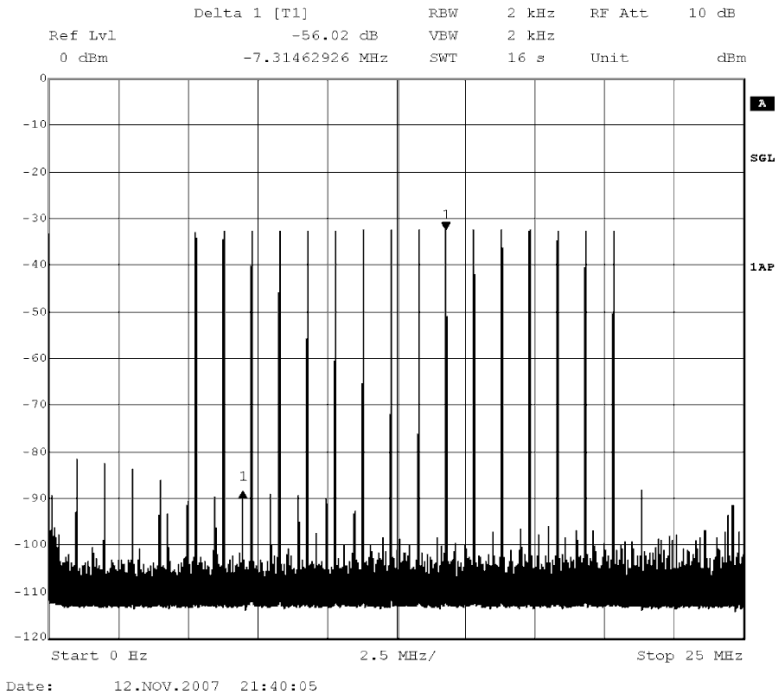


Figure 17.5-2. Multi-tone spectrum measurements for a single sub-DAC.

17.6. CONCLUSIONS

A practical validation through test-circuits implementation and measurements of the new DAC correction method for suppression of HD is presented. A test-chip with four parallel 12 bit sub-DACs in 180m, CMOS process is designed, fabricated and measured. The correction method is demonstrated in two different implementations: using two virtual sub-DACs and using two real sub-DACs.

The method can be applied in conventional DACs by emulating multiple virtual DACs through interleaving in time-multiplexed digital phase-shifted input signals. It is shown that this approach effectively can trade signal bandwidth for signal linearity. When HD3 is targeted for suppression using two virtual parallel sub-DACs and $\frac{\pi}{3}$ phase-shift, an improvement of 10dB is measured for SFDR. It is also shown that this method is not effective when the HD components of the odd image bands are located in the signal band, since due to the time-interleaving of the virtual input signals the odd image bands are inverted (phase delayed by π) with respect to the signal band. That is why using real, instead of virtual sub-DACs, is even more effective.

The method is as well demonstrated with real parallel sub-DACs, which are synchronized and hence the output signal image bands do not feature phase delays. When HD3 is targeted for suppression using two real parallel sub-DACs and $\frac{\pi}{3}$ phase shifts, SFDR=80dB is measured for f_{in}=16.9MHz. It is argued that the method is suitable for OFDM systems, where the input signals area available in the complex domain and the implementation of phase shifts is easy. For an exemplary OFDM system (multi-tone measurements), an improvement in the dynamic range of 8dB is measured.

This method is a solution to improve DAC linearity. It is attractive for very high-speeds, where the known DAC correction methods are not that effective. Considering the implementation of the phase delays, the method is suitable for DDFS and OFDM applications.

18. Chapter

A 14 BIT QUAD CORE FLEXIBLE 180NM DAC PLATFORM

This chapter presents a flexible DAC design that features 4 12 bit sub-DAC cores and it is designed in 180nm CMOS technology. The test chip has a simple pre-processor in the form of an externally controlled de-multiplexer to distribute the digital sub-DAC input words $w_i(nT), i = 1,2,3,4$. The analog post-processing is an off-chip summation of the sub-DAC output currents. For a SoC co-integration, the design of the sub-DAC unit needs to be area efficient, since the production cost of the DAC platform is added to the price of the whole SoC. Therefore, a large binary LSB portion is chosen for the segmentation of the sub-DAC unit. Its architecture features 8 LSB binary bits and 4 MSB unary bits (15 unary currents). To save furthermore silicon area, the sub-DAC unit has relaxed design specifications of about 9 to 10 bits intrinsic linearity. The smart op-modes of the flexible DAC architecture and a built-in self-calibration apparatus can improve the DAC static linearity to state-of-the-art levels, while the occupied silicon area of the flexible DAC platform is one of the smallest reported in the literature. It is 0.2mm^2 per 12 bit sub-DAC unit, i.e. 0.8mm^2 for the whole flexible 14 bit DAC platform.

18.1. INTRODUCTION

Concepts of DAC flexibility presented in chapter 14 are implemented in an 180nm quad-core flexible DAC platform. The platform features 4 parallel 12 bit sub-DACs with a simple pre-processor that is based on multiplexors. It allows using the sub-DACs independently or combined in a flexible op-mode. This test chip demonstrates functional segmentation of the DAC analog resources as presented in chapter 9. It enables four D/A (sub-)functions, which can be used in different flexible op-modes.

The main advantages of this flexible DAC platform are flexible design, functionality and performance. The design of the platform concentrates on the design of the 12 bit sub-DAC, which allows at least four times simpler and hence faster transistor level simulations than those for the complete 14 bit system. The further construction of higher resolution platforms is facilitates through using in parallel the well-characterized 12 bit sub-DAC cores. The functionality of the platform is flexible, since it allows using the 4 DAC sub-function in different ways – either independently or combined. Based on the various op-modes, the platform can deliver flexible performance, which is a defining feature making it different than the conventional DACs which are designed for a point-solution.

The aims set to this test-chip are to validate new DAC correction concepts presented in part IV of the thesis, in particular to validate flexibility concepts presented in chapter 14, and to provide practical insights on the challenges of designing DAC platforms based on functional segmentation. In addition, this test-chip is the precursor of the 12-to-16 bit flexible DAC platform that is presented in chapter 19.

Section 18.2. presents the test-chip design. Section 18.3. presents measurement results. Finally, conclusions are drawn in section 18.4.

18.2. DESIGN

A 12 bit DAC unit is used as a foundation to build a flexible quad-core DAC platform. To reduce area and increase efficiency the DAC unit features large binary algorithmic part segmentation and low intrinsic linearity. The algorithmic segmentation features 8 LSB binary bits and 15 MSB unary bits. The intrinsic linearity is designed to about 9 to 10 bits. An integrated calibration engine improves the DAC linearity to about 14 bit level.

A high-level block diagram of the quad-core flexible DAC platform is shown in Figure 18.2-1. The digital input for the DAC platform includes 12bits of data and 4bits of control signals. The data contains the sub-DAC input words $w_i(nT), i = 1,2,3,4$, distributed in way that is determined by the chosen op-mode.

The control signals contain control information for the data word: for which sub-DAC the data is and at which clock phase the data should become available to the sub-DACs. A simple pre-processor controls the distribution of the input DAC data to the four sub-DACs, depending on the 4 control signals. The “read enable” information for the first multiplexer selects which input data is addressed for the given sub-DAC branch. The “clock phase enable” information for the second multiplexer selects when the received sub-DAC data to become available for the sub-DAC. Flexible op-modes require this option when sub-DACs need to synchronize. The sub-DACs are indicated as sub-DAC 1, 2, 3 and 4 (at some places further, they are referred to as A, B, C, and D).

Figure 18.2-2 shows a micrograph photo of the chip. The sub-DAC units are vertically placed next to each other. The array of signal current source transistors is marked 1. The array of calibration DACs is marked 2, see [79]. The array of current source cascode transistors is 3. The binary-to-unary decoder, the synchronization

latches, and the drivers are shown with 4. The pre-processor is shown with 5. The input LVDS buffers and decoupling capacitors are shown with 6. The current sign detector (1 bit ADC for self measurements) is shown with 7.

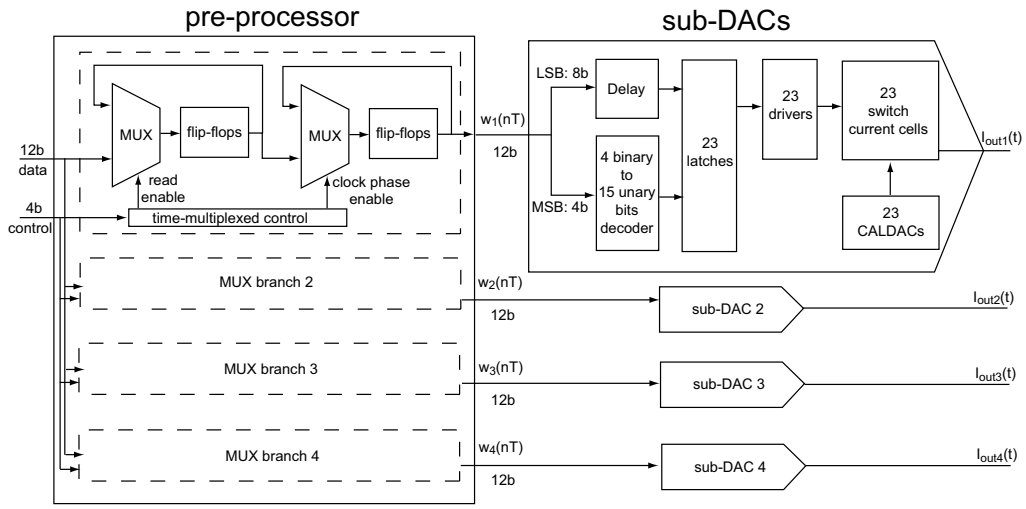


Figure 18.2-1. A high-level block diagram of the implemented 14 bit flexible quad-core DAC platform.

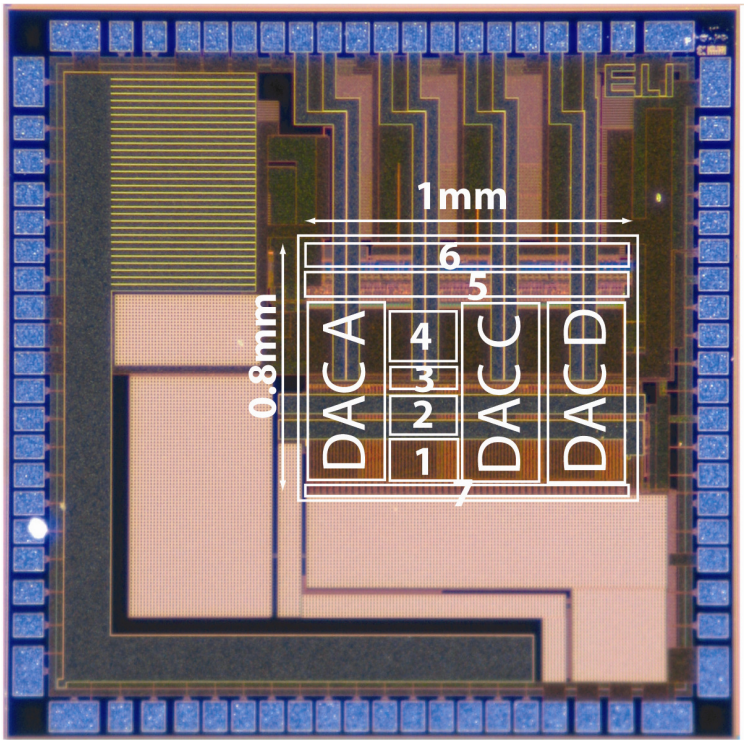


Figure 18.2-2. Chip micrograph.

18.3. MEASUREMENTS

To demonstrate the potential of the proposed flexible DAC platform, the test chip was measured in several op-modes. Since its flexibility decouples the application from silicon design, the test-chip does not provide a “point solution”. However, the flexible DAC delivers different performance for different configurations. That is why the shown performance only reveals the potential of the platform. Better performance can always be delivered with better application design, e.g. better error mapping, tailored power consumption, etc.

The chip power consumption consists of two main parts: digital (data processing) and analog (output signal) power consumption. Concerning the DAC analog signal output power, the presented results are measured for 24mA full-scale current, i.e. 6mA per sub-DAC, terminated with a 50Ω differential load resistance. The full digital data processing power consumption of the flexible DAC is 118.8mW at 1.8V supply (66mA current: the digital circuits have CML implementation). The distribution of the digital power consumption is 27mW per sub-DAC and 10.8mW for the pre-processor. Thus, the overall DAC power consumption is flexible from 37.8mW (only one sub-DAC is used and the rest are turned off) to 118.8mW (all sub-DACs operate simultaneously).

The intrinsic static performance of the DAC is shown in Figure 18.3-1 and Figure 18.3-2. Large gain errors between the sub-DACs were measured which are due to the particular test chip implementation. For these measurements, the gain errors between the sub-DACs are externally compensated and hence not taken into account. Figure 18.3-1 shows the individual static performance of the sub-DACs. Their accuracy is about 10 bit, since the design requirements are relaxed. In 14b op-mode, when the full-scale range of the DAC is increased four times by using the sub-DACs in a “stacked” successive way, the accuracy is again 2 bits less, i.e. 12b; INL and DNL are respectively shown in Figure 18.3-2 left and right, when the sub-DAC gain errors are not taken into account..

The built-in self-calibration apparatus can improve the static performance by at least 3 bits. The 14bit op-mode DAC calibrated static performance is shown in Figure 18.3-3.

Alternatively, the DAC performance can be improved by the mapping technique. Once the errors $e_i(t)$ are known, the greatest challenge is to implement an algorithm to find the best map to distribute $w_i(nT),\ i=1,2,3,4$. Problems arise due to the practically countless number of possible combinations. For the sake of illustration only, the information of the current mismatch errors is processed by a MATLAB script. The script delivers an exemplary map for a limited number of combinations, the results of which are shown in Figure 18.3-4.

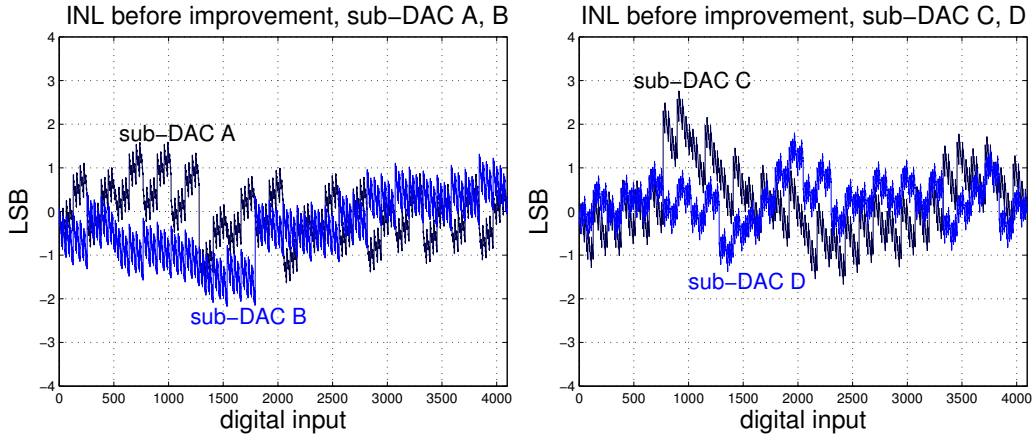


Figure 18.3-1. Intrinsic linearity 12 bit INL of sub-DACs: A, B (left); C, D (right);

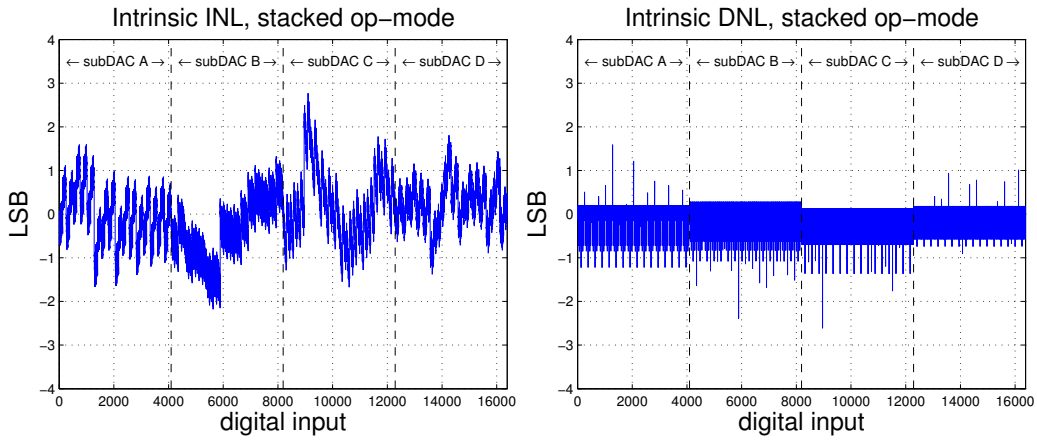


Figure 18.3-2. Intrinsic linearity: 14 bit op-mode INL of "stacked" sub-DACs A,B,C,D (left); 14b op-mode DNL of "stacked" sub-DACs A,B,C,D (right). Gain errors between sub-DACs are not taken into account.

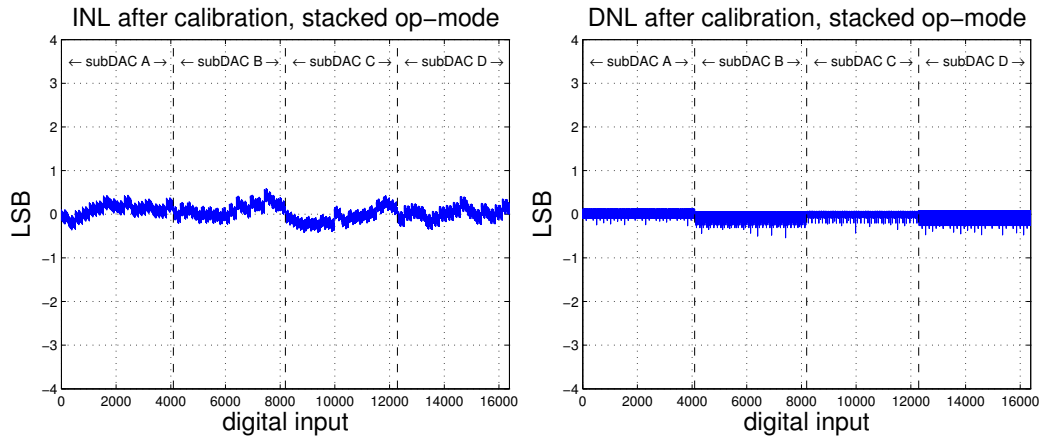


Figure 18.3-3 – 14 bit op-mode DAC INL (left) and DNL (right) after calibration of all binary and unary currents. Gain errors between sub-DACs are not taken into account.

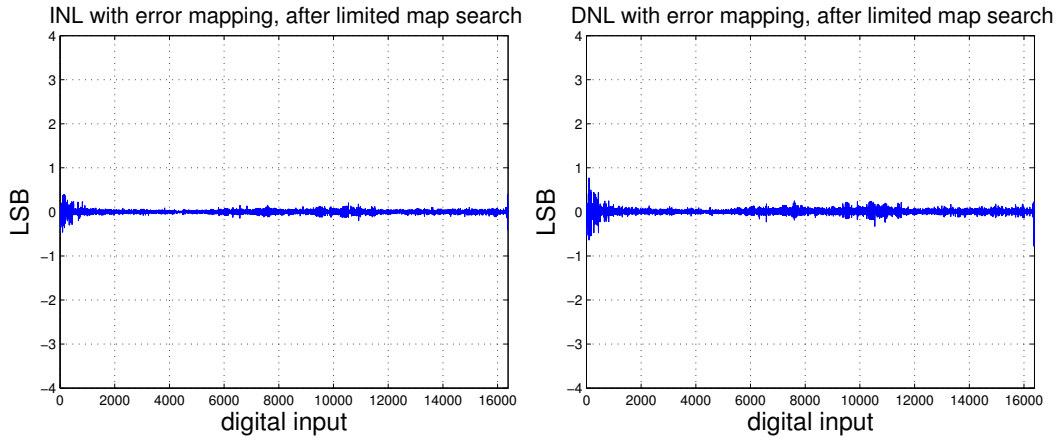


Figure 18.3-4. 14 bit op-mode DAC INL and DNL after errors compensation by means of mapping the distribution of the sub-DAC words. Gain errors between sub-DACs are not taken into account.

For all static measurements, systematic gain errors between the separate sub-DACs are measured. Since these errors are due to the particular test-chip implementation and are beyond the correction scope of the mapping methods, they are mathematically compensated in the presented plots.

Concerning DAC dynamic performance, this chapter presents the experiments of several op-modes. At lower speeds, better linearity in terms of SFDR (Spurious-Free-Dynamic Range) is achieved with op-modes for which more sub-DACs are used. At higher speeds, more than 10dB better intrinsic performance is achieved when only one sub-DAC is used. The main reason for this flexible performance is that different error mechanisms dominate at lower and higher speeds. At low speeds, the predominant error mechanism is the current mismatch error, i.e. the DAC amplitude accuracy. As more sub-DACs operate in parallel the current mismatch errors (or the post-calibration errors), both random and systematic, average out. At higher speeds, the predominant error mechanisms arise due to synchronization errors of the current cells and the dynamic disturbances on the power lines and transistor biasing nets. Figure 18.3-5 demonstrates this trade-off with SFDR measurements for different input signal frequencies f_{in}.

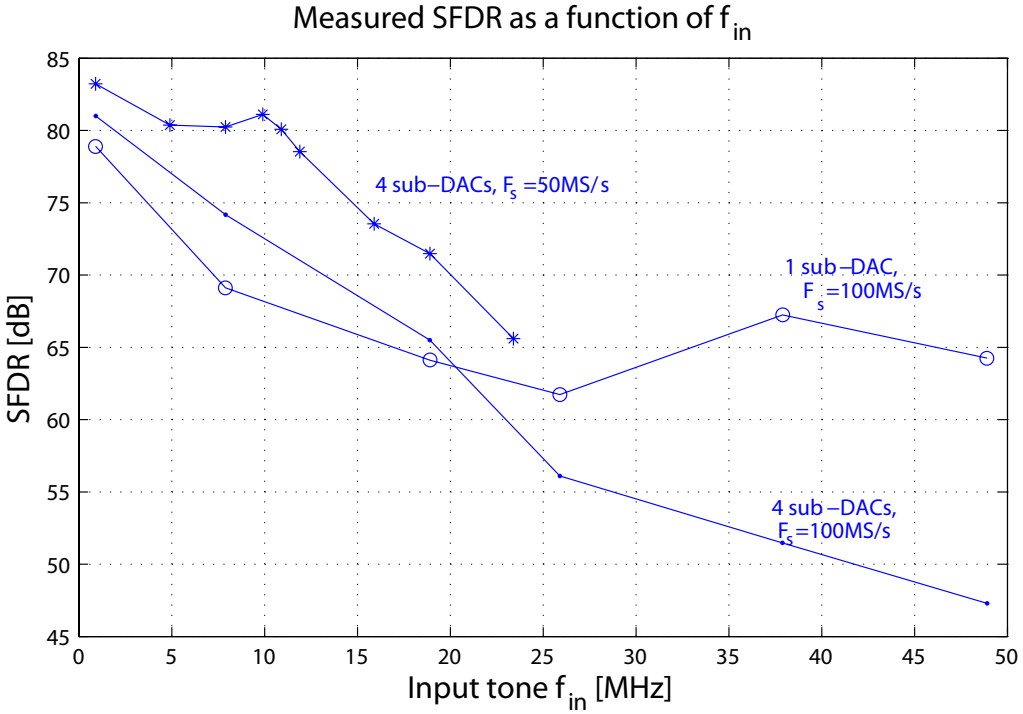


Figure 18.3-5 –SFDR performance against f_{in} for different modes of operation (after calibration).

Figure 18.3-6 and Figure 18.3-7 show the DAC output spectra at high speeds. The sampling rate is $F_s = 100MS/s$ and the input tone frequency is $f_{in} = 49MHz$. Figure 18.3-6 shows the spectrum of the 14b op-mode with SFDR limited to 47.2dB. Figure 18.3-7 shows the spectrum of the 12b op-mode with SFDR limited to 64.2dB. Thus, at high speeds the use of single sub-DACs op-modes is beneficial. At lower speeds, the op-modes using more parallel sub-DACs deliver better performance. For example, the op-mode of four parallel sub-DACs with self-calibration maintains SFDR >80dB up to $f_{in} = 11MHz$. The DAC output spectrum of this op-mode is shown in Figure 18.3-8.

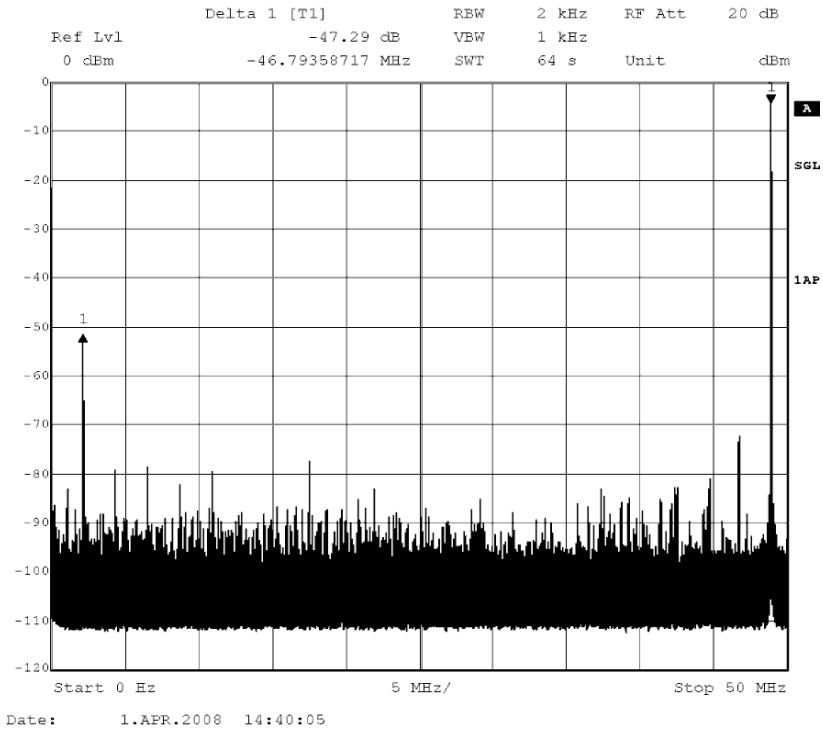


Figure 18.3-6. 14 bit op-mode spectrum for $f_{in} = 49MHz$, SFDR=47dB.

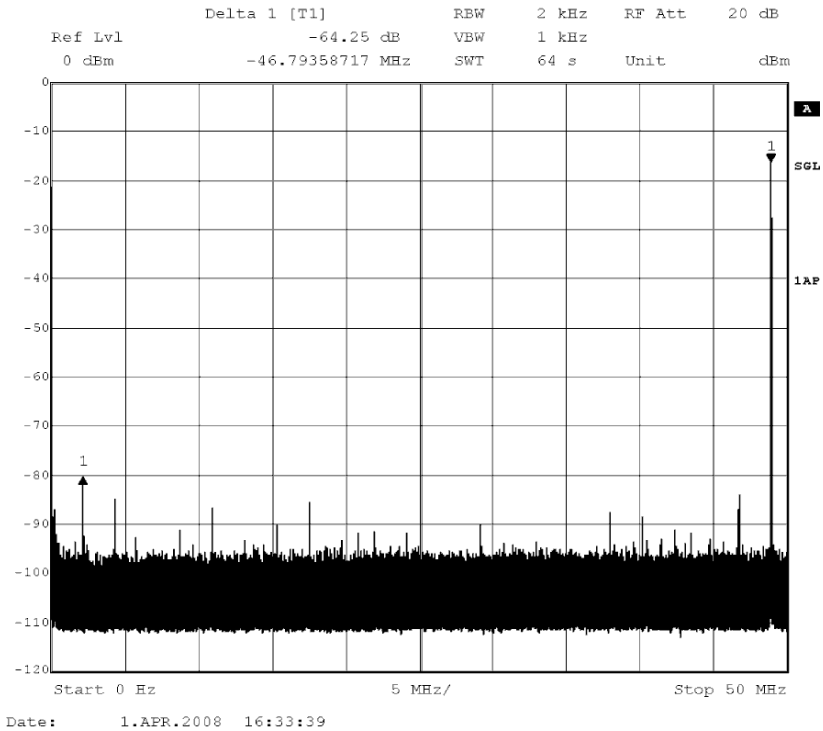


Figure 18.3-7. 12 bit op-mode spectrum for $f_{in} = 49MHz$, SFDR=64dB.

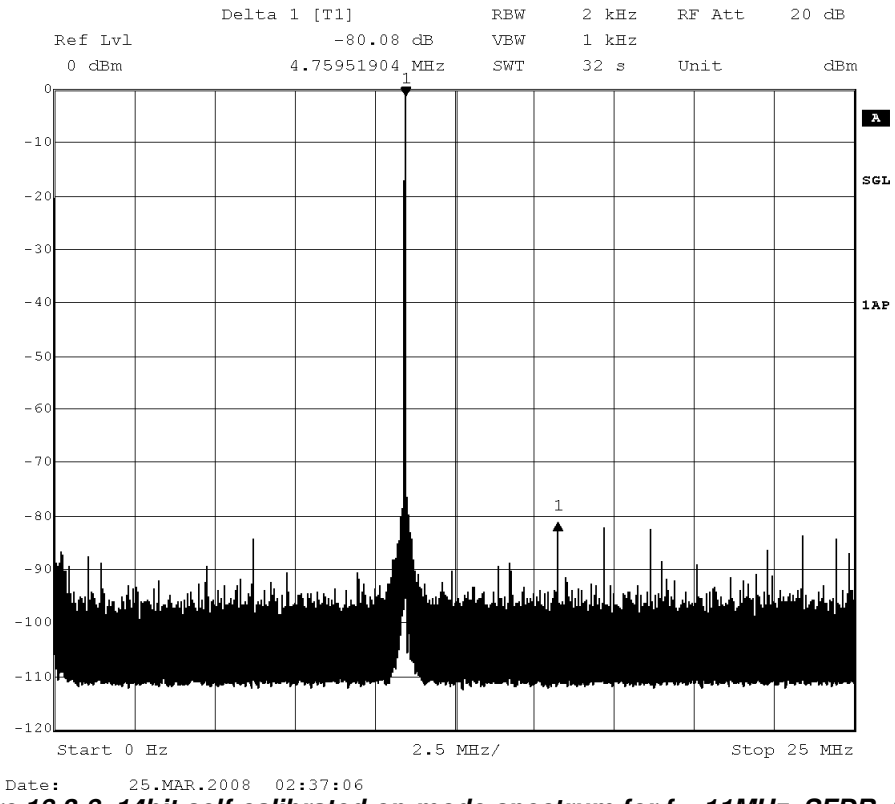


Figure 18.3-8. 14bit self-calibrated op-mode spectrum for f_{in}=11MHz, SFDR=80dB.

18.4. CONCLUSIONS

A concept for DAC flexibility is validated. A new flexible DAC architecture demonstrates this concept. The flexible architecture is based on multiple parallel sub-DACs, a digital pre-processor, and analog post-processing. A test-chip example of a 12 bit quad-core DAC is manufactured in 180nm CMOS. Its main advantages include flexible design, functionality and performance. This test-chip validates in practice the flexibility concepts presented in chapter 14. The inherent redundancy of the parallel sub-DACs enables a number of special operation modes that can further improve DAC performance. This allows relaxing the design requirements to the DAC analog intrinsic core and hence allowing smaller and more area efficient designs. These properties make the proposed DAC platform architecture an excellent candidate for co-integration with other flexible SoC platforms, such as the FPGAs. The insights that this test-chip provides in practice, e.g. sub-DAC gain errors, organization of the pre-processor, are important conclusions for the design of the presented in chapter 19 12-to-16 bit flexible DAC platform.

19. Chapter

A 16 BIT 16-CORE FLEXIBLE 40NM DAC PLATFORM

This chapter presents a flexible DAC design in 40nm CMOS technology. It features large scale functional unary segmentation. 16 parallel sub-DACs construct a flexible DAC platform. The pre-processor is realized in the form of a functional "binary-to-unary" decoder, which generates the sub-DAC input words $w_i(nT), i = 1,2,3,...16$. For the sake of simplified testing, the sub-DACs are internally grouped in packs to form one 15 bit sub-DAC (8 12 bit sub-DACs are connected at their outputs), one 14 bit sub-DAC (4 12 bit sub-DACs), one 13 bit (2 12 bit sub-DACs), and two 12 bit sub-DACs. Each sub-DAC features algorithmic segmentation of 8 LSB binary bits and 4 MSB unary bits (15 unary currents). The transistor design features thin and thick oxide circuits. All digital circuits are implemented with thin oxide transistors to save area and to increase speed. The DAC current cells are implemented with thick oxide transistors to interface the DAC to 2.5V. The occupied silicon area of the flexible DAC platform is the smallest reported in the literature. It is 0.05mm^2 per 12 bit sub-DAC unit, i.e. 0.8mm^2 for the whole flexible 16 bit DAC platform.

19.1. INTRODUCTION

With modern CMOS fabrication technologies, redundancy is becoming increasingly affordable and realizable. The high integration densities lead to a reduced price per transistor and hence to a reduced price per on-chip integrated function. This chapter demonstrates a DAC design in 40nm CMOS featuring massive functional redundancy – 16 identical sub-DACs. The redundancy is used to realize flexibility and correction methods. Section 19.2. presents the DAC design. Section 19.3. reports measurement results. Finally, conclusions are drawn in section 19.4.

19.2. FLEXIBLE DAC PLATFORM BASED ON 16 CORE UNITS

Instances of a 12 bit DAC unit are used to build a 16 bit flexible DAC platform. The resolution of the flexible DAC platform ranges for the various op-modes from 12 bits to 16 bits. A test-chip DAC is designed and fabricated in 40nm CMOS technology. Its high-level block diagram is shown in Figure 19.2-1.

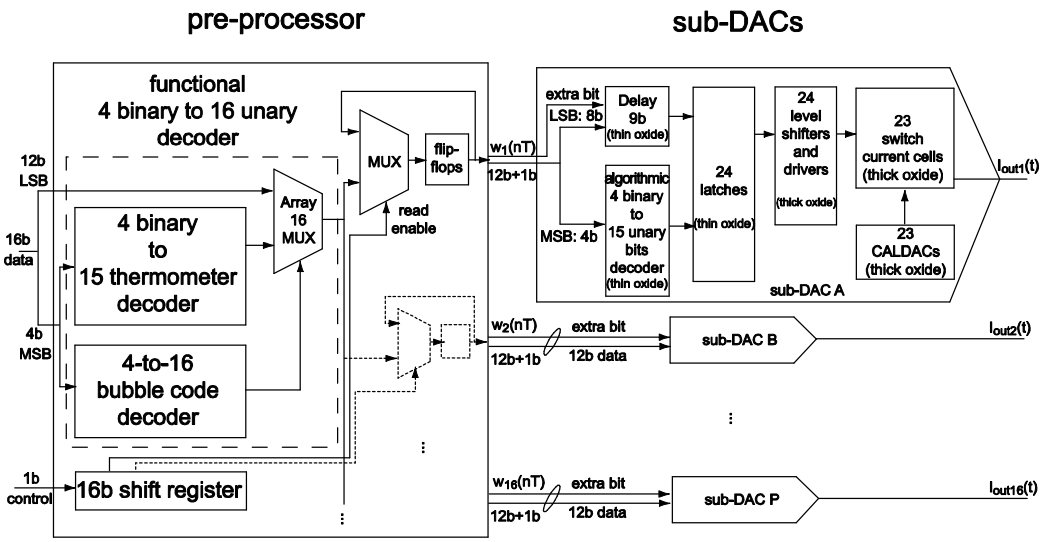


Figure 19.2-1. A high-level block diagram of the implemented 16 bit flexible 16-core DAC platform.

The 16 parallel 12 bit sub-DACs (A, B, C… P) realize a functional unary segmentation as introduced in chapter 9. A functional binary to thermometer (unary) 4-to-16 decoder is implemented in the digital pre-processor. The decoder is based on the single DAC op-mode that is described in Table 14.4-1, row 3, i.e. "*N+log$_2$(m)* bit single DAC with large full-scale range and high resolution (option 1, decoder-based: "stacked sub-DACs" op-mode)". The digital data input for the DAC platform is 16 bits. These are divided into two parts, 4 MSB and 12 LSB. The 4 MSB effectively select which sub-DAC processes the 12LSB bits, which sub-DACs are set to Full-scale (including their extra bits), and which sub-DACs are set to zero. That is to say that the functional decoder divides the whole 16 bit DAC transfer characteristic into 16 segments, each of which is 12 bits. These segments are mapped to the sub-DACs. For a given input digital code, one segment (sub-DAC), pointed by the bubble code, is selected to implement the 12 bit conversion. All segments (sub-DACs) between zero and the chosen segment are set to "1", while all segments (sub-DACs) between the chosen segment and full-scale are set to "0".

Each sub-DAC data channel passes through a “read enable” circuit that allows effectively using sub-DACs independently. These “read enable” circuits are implemented with multiplexers and flip-flops. The multiplexers are controlled by the 1 bit control data, which either allows the current 12 bit data or skips it. In such a way the input data can be made transparent for some sub-DACs and so the sub-DACs can be used independently.

The architecture of a sub-DAC unit features algorithmic segmentation of 4 MSB unary and 8 LSB binary bits. All digital circuits use CMOS logic implemented with thin oxide transistors. At the output of the DAC, the switched current cells are implemented with thick oxide transistors. Level shifters are used to adapt the 1V logic levels of the thin oxide circuits to the 2.5V logic levels of the thick oxide circuits.

The problem of the missing code, as discussed in chapter 14.5. , is solved by implementing an extra code at full-scale for each sub-DAC. Thus, the sub-DAC full-scale range is 2^{12} codes, instead of the usual 2^{12}-1.

Figure 19.2-2 shows the construction of the full transfer characteristic when all sub-DACs (A, B, C… P) are used to construct a 16 bit DAC function. For the sake of simplicity, the differential implementation of the DAC is explained as single-ended. The letters of the sub-DACs are indicated in three different ways depending on their role at a given point of the transfer characteristic. When the letter of a sub-DAC is given in capital bold, it indicates that this sub-DAC is set to full-scale ‘1’ and the extra code at full-scale is also turned on. That is to say that the sub-DAC contributes 2^{12} LSB units to the output. When the letter of a sub-DAC is given in a small italic letter, it indicates that this sub-DAC converts the 12 bit LSB data of the digital input into its analog equivalent at the output. When the letter of a sub-DAC is given in narrow gray capital letter, it indicates that this sub-DAC is set to ‘0’. That is to say that the sub-DAC contributes nothing to the output.

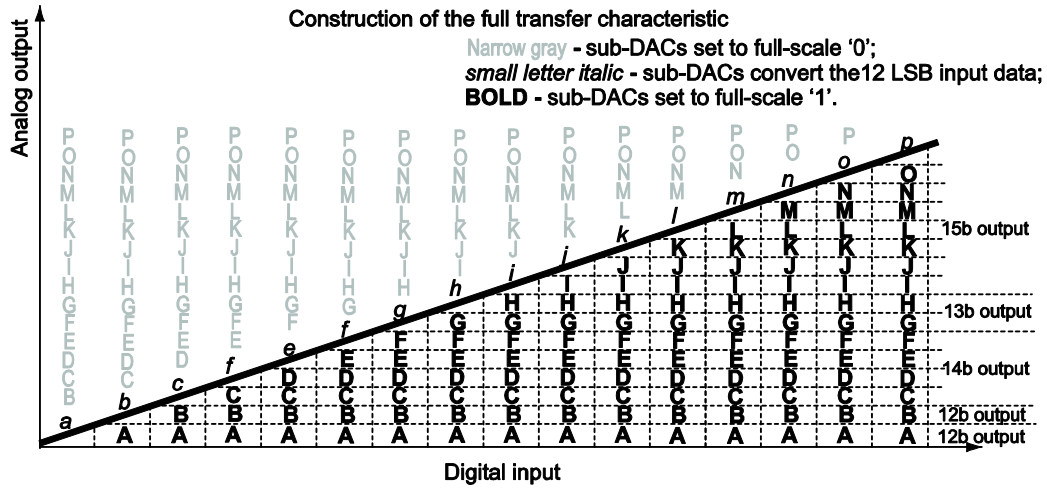


Figure 19.2-2. Construction of the full transfer characteristic by the 16 sub-DAC units.

Each of the 16 sub-DACs can be independently used. However, some of the sub-DAC outputs are connected together due to practical reasons of test chip packaging and measuring. Thus, sub-DAC A is led to a test chip output. Sub-DAC B is led directly to another test chip output. Sub-DACs C, D, E, and F grouped and then led directly to another test chip output, effectively together forming one 14 bit sub-DAC. Sub-DACs G and H are grouped and led to another test chip output, effectively together forming one 13 bit sub-DAC. Sub-DACs I, J, K, L, M, N, O and P are grouped and led to another test-chip output, effectively together forming one 15 bit sub-DAC.

Each sub-DAC includes calibration and self-measurement circuits based on the presented and validated new methods in chapter 16. The test-chip communicates through a serial interface with off-chip calibration algorithms. Using calibration allows an aggressive and efficient silicon design, since the DAC accuracy should be guaranteed by the calibration and not by the fabrication process. Therefore, the DAC intrinsic accuracy is designed for about 8 to 9 bit, but without having any information about the matching of the fabrication process.[1]

Figure 19.2-3 shows the layout and the chip micrograph of the fabricated flexible DAC platform test-chip. The occupied area per 12 bit sub-DAC is about 0.05mm^2. This is the smallest 12 bit DAC reported in the open literature. The overall occupied area of the core of the flexible DAC platform is about 0.75mm^2, which is also the smallest reported size for a 16 bit DAC implementation.

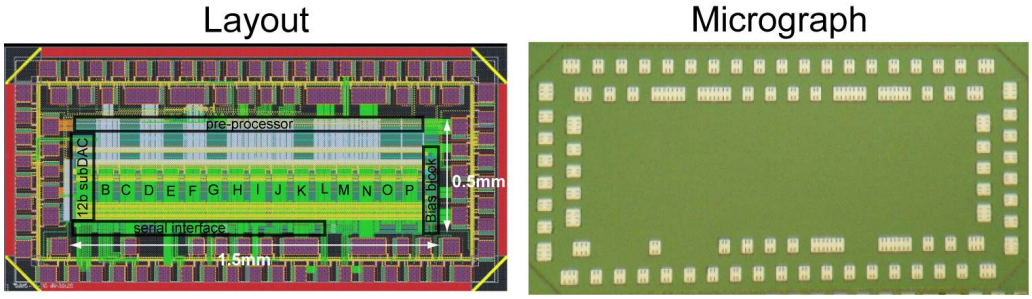


Figure 19.2-3. Layout and micrograph of the flexible self-calibrated 16 bit DAC.

19.3. MEASUREMENTS

The test-chip flexible DAC platform operates with two main power supply voltages – a 1V power supply for the thin oxide based digital circuits of the DAC core and a 2.5V power supply for the DAC periphery, i.e. input buffers, output switched current cells, bias circuits, etc.

The normal power consumption for the thin-oxide digital circuits is about 5mW for the clock network (incl. buffers) and 5mW per used sub-DAC, when the circuit is clocked at about 1GS/s. The digital circuits within a sub-DAC are still functional up to about 1.5GS/s, providing that the power supply for the DAC core is increased to 1.2V. However, the digital circuits of the pre-processor are only functional up to about 100MS/s, which is believed to be due to the large parasitics associated with the elongated layout of the pre-processor (1.5mm width).

The normal power consumption for the thick oxide periphery is about 50mW for the bias circuits, 1mW for the input LVDS buffers, 3mW for the level-shifters and data buffers per sub-DAC, and about 10mW (adjustable) of output signal power per sub-DAC.

When the sub-DACs are considered individually, their performance can be fairly evaluated as in the cases of normal conventional DACs. In general, the sub-DACs demonstrate similar intrinsic accuracy, about 8 to 9 bits. However, there are exceptions that deviate largely from the average. Such an exception is sub-DAC B (only 7 bits accuracy), due to its extremely large mismatch errors between MSB unary currents and the LSB binary currents. Fortunately, the calibration can correct these errors and still guarantee performance.

[1] By the time of the design of the chip, the used 40nm CMOS fabrication process was poorly characterized.

Figure 19.3-1 and Figure 19.3-2 respectively show the INL and DNL characteristics of the 12 bit sub-DAC B, before and after calibration. The calibration has been performed manually. The DAC transfer characteristic was measured and the mismatch errors were evaluated. Then, the appropriated control (static) data for the CALDAC registers was loaded through the serial interface. Next, the DAC transfer characteristic was again measured and the mismatch errors were re-evaluated and new data was reloaded in the CALDAC registers, etc. The whole process had to be controlled and executed manually because of time pressure. This has introduced extra noise errors in the measurement results (these errors are referred to as the influence of the human factor). Thus, the final accuracy is limited by the quantization error of the correction step, the measurement noise and the human factor. (as opposed to the self-calibrating chips in chapter 16, where the self-measurements and algorithm are integrated on-chip, leading to post-calibration accuracy mainly determined by the quantization error of the correction step).

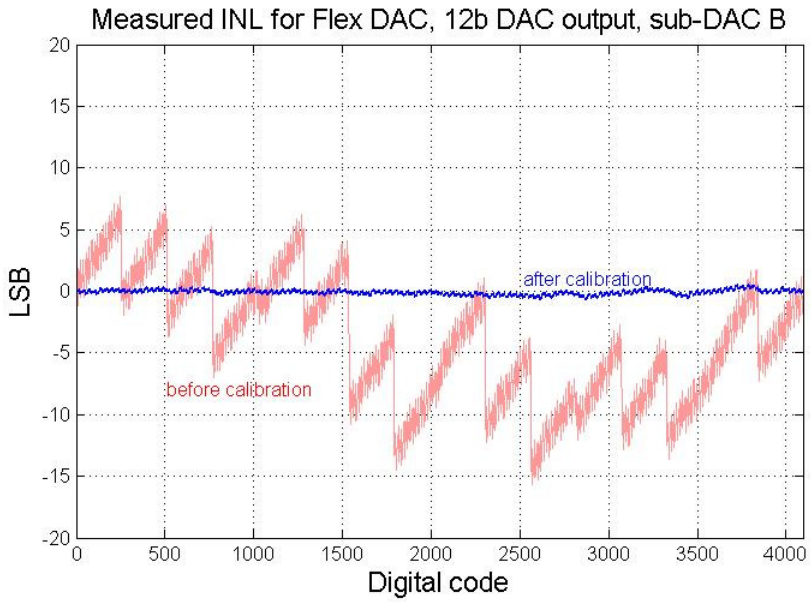


Figure 19.3-1. Measured INL for sub-DAC B before (light red) and after (dark blue) calibration.

INL_{max} before calibration is about 15LSB, indicating 7 bits linearity. INL_{max} after calibration is about 0.6LSB, indicating almost 12 bits linearity. Thus, the improvement is 5 bits. The DNL_{max} before calibration is about 12 LSB and it is improved up to 0.3LSB after calibration.

The static measurement results before calibration of the “14 bit sub-DAC” output, which includes the on-chip internally-connected sub-DACs C, D, E, and F, are shown in Figure 19.3-3, Figure 19.3-4, Figure 19.3-5, and Figure 19.3-6. These measurements show how sub-DACs can be used together (“stacked” op-mode, as defined in Table 14.4-1) to form a higher resolution DAC with the same LSB step but higher full-scale range than in each of the individual units – the sub-DACs.

Figure 19.3-3 shows the measured INL of sub-DACs C, D, E, and F. These show about 8 bit linearity for all sub-DACs before calibration and about 12 bit linearity after calibration. The full-scale current per sub-DAC is set to 5mA for these measurements. These measurements are representative of the expected intrinsic, i.e. before calibration, linearity of the sub-DACs (as opposed to sub-DAC B, shown in Figure 19.3-1, which is an exception).

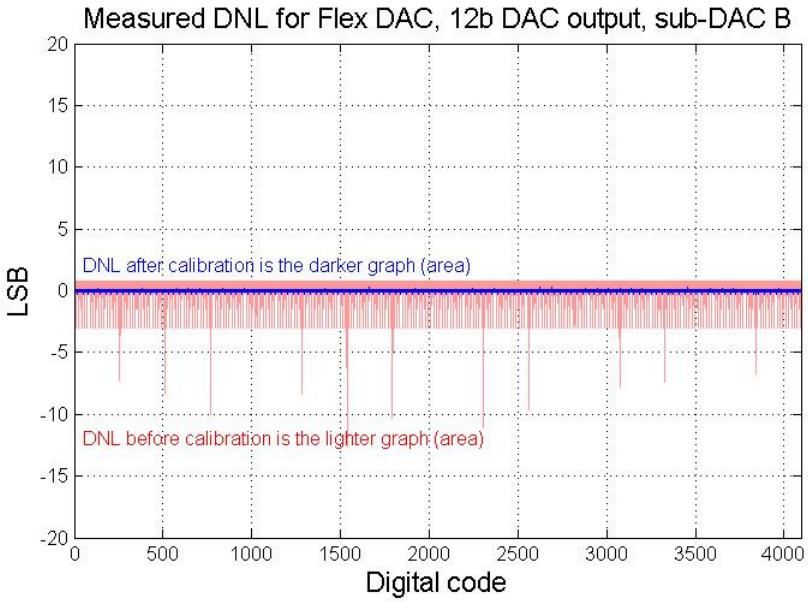


Figure 19.3-2. Measured DNL for sub-DAC B before (light red) and after (dark blue) calibration.

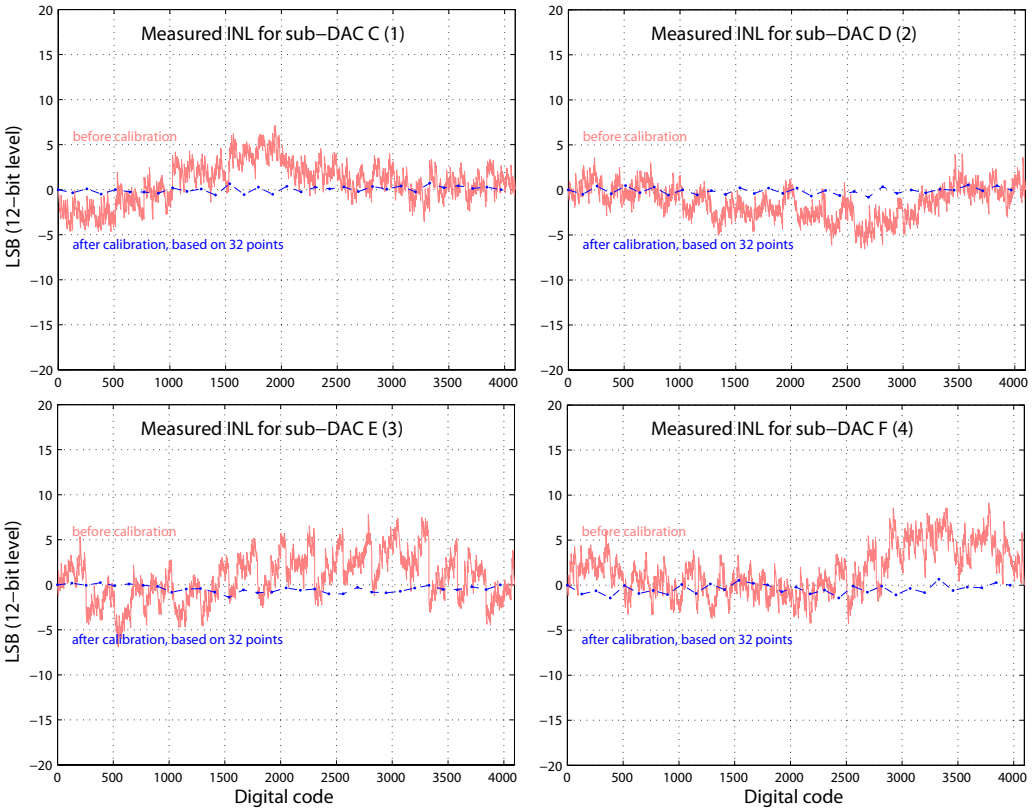


Figure 19.3-3. Measured INL for sub-DACs C (1), D(2), E(3), F(4), before calibration, i.e. 8 bit linearity, while after calibration 12 bit.

Figure 19.3-4 shows the measured DNL of sub-DACs C, D, E, and F. The performance of the individual sub-DACs is statistically the same, featuring about 8 bit linearity.

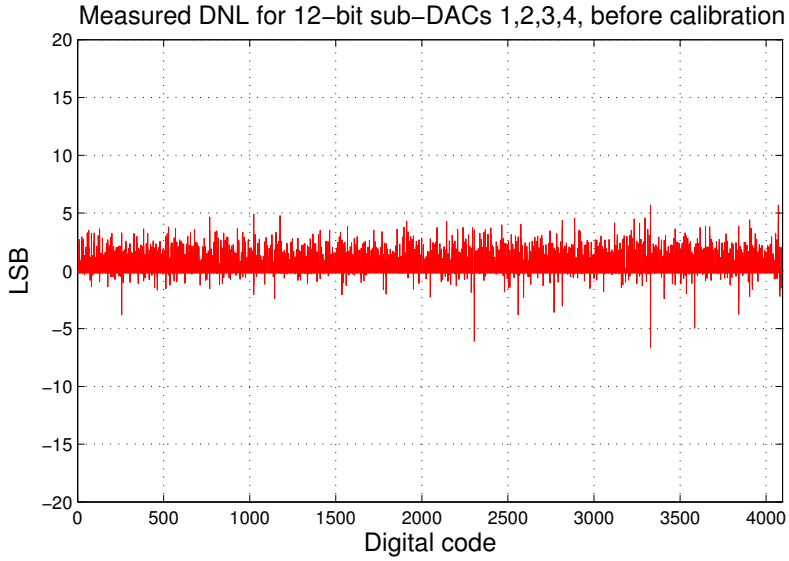


Figure 19.3-4. Measured DNL for sub-DAC 1 (C), 2(D), 3(E), 4(F) before calibration, i.e. 9 bit linearity.

When the 12 bit sub-DACs C, D, E, and F are used in "stacked" mode through the functional decoder shown in Figure 19.2-1and illustrated in Figure 19.2-2, a 14 bit resolution op-mode can be realized. Figure 19.3-5 shows the INL for a 14 bit op-mode based on the "stacked" sub-DACs C, D, E, and F. As discussed in chapters 14.4.1. and 18, the mismatch errors between the gains of the sub-DACs contribute to the overall non-linearity. However, mainly due to the increased resolution and averaging effects, the relative linearity before calibration is 9 bit, which is 1 bit improvement.

Note, however that the gain mismatch errors between the sub-DACs modulate the whole INL characteristic of the "stacked" op-mode into generating low order non-linearity, e.g. 2^{nd}, 3^{rd}, and 4^{th}. Such low order non-linearity of the static DAC transfer characteristic generates HD, e.g. HD2, HD3, and HD4. These HD components would be particularly strong, because the gain mismatch error energy is concentrated into these and not distributed among all HD, like in the code interleaved (listed in Table 14.4-1). Fortunately, the calibration of the current sources can compensate for the gain mismatch among the sub-DACs, too. Figure 19.3-5 shows an INL_{max} of about 2 LSB (14 bit level) after calibration. The measurement of the INL after calibration is based on 128 measured points. The DAC output values for codes between the measured points are interpolated. Such approximation does not compromise the measurement results, since no major non-linearity is expected in the LSB binary part.

Figure 19.3-6 shows the DNL for a 14 bit op-mode based on the "stacked" sub-DACs C, D, E, and F before and after calibration. As discussed in chapters 14.4.1. , the DNL scales with the resolution, because the absolute values of the local errors are unaffected by the stacking of sub-DACs. Therefore, the DNL before calibration for the 14 bit op-mode is at 11 bit level.

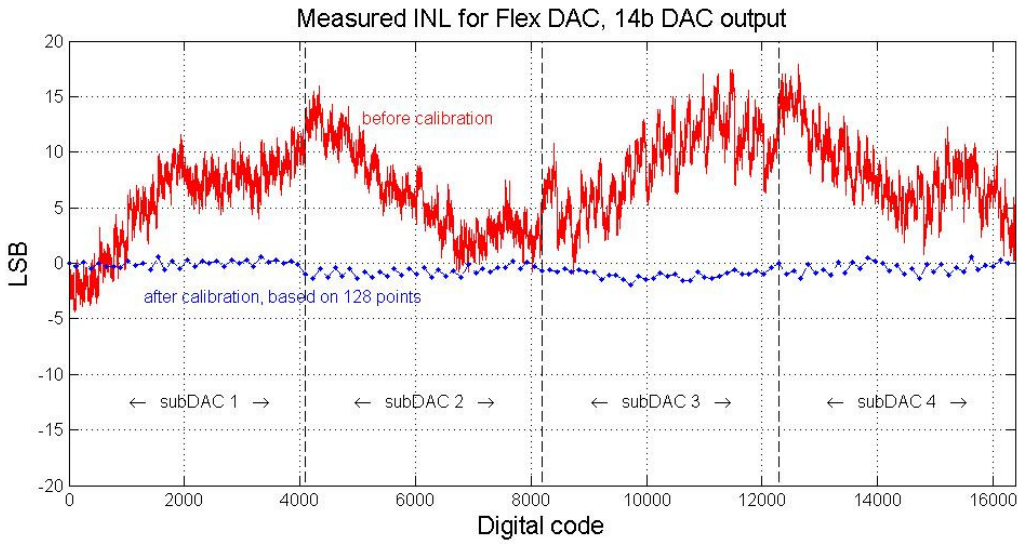


Figure 19.3-5. Measured INL for 14 bit op-mode: 9 bit linearity before calibration (based on 2048 points and interpolation), 12 bit linearity after calibration (based on 128 measured points and interpolation).

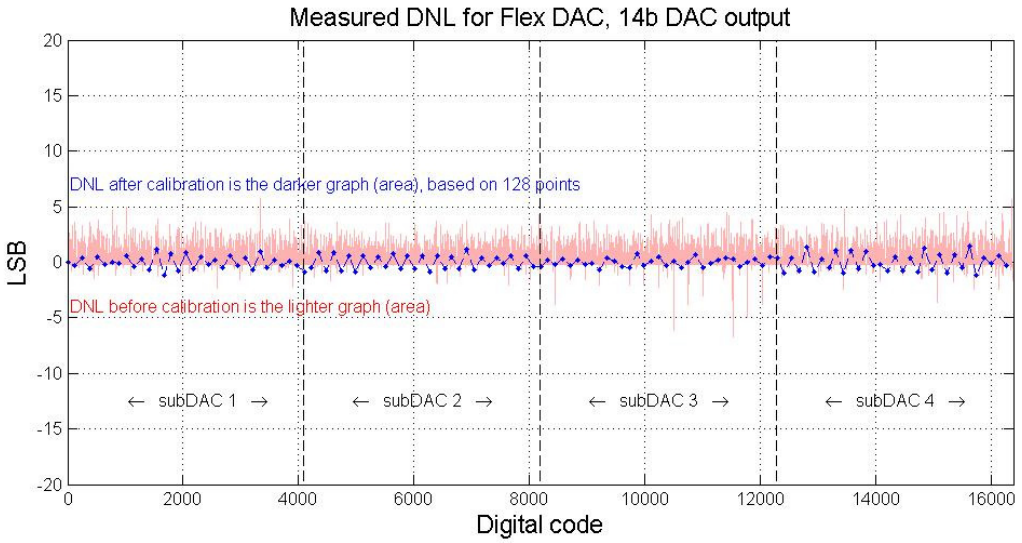


Figure 19.3-6. Measured DNL for 14 bit op-mode: 11 bit linearity before calibration and 12 bit linearity after calibration.

For the two discussed op-modes, 12 bit and 14 bit “stacked” op-modes, the dynamic performance of the flexible DAC test chip, in terms of SFDR versus input single tone signal frequency f_{in}, is shown in Figure 19.3-7 for sampling rate of F_s=100MS/s. A similar performance trade-off is demonstrated in both cases, before and after calibration. The high-resolution op-mode delivers better performance at low f_{in}, while the low resolution op-mode delivers better performance at high f_{in}. These results are consistent with the theory and measurements of chapters 14 and 18.

Note that the calibration of currents improves SFDR performance not only at low f_{in} but also at high f_{in}. These results are consistent with the modeling presented in chapter 5.3. and the measurements presented in chapter 16.

Figure 19.3-8 and Figure 19.3-9 show the 12b op-mode DAC output spectra respectively before and after calibration for f_{in}=3.2MHz. Figure 19.3-10 and Figure 19.3-11 show the 14 bit op-mode DAC output spectra respectively before and after calibration for f_{in}=5MHz.

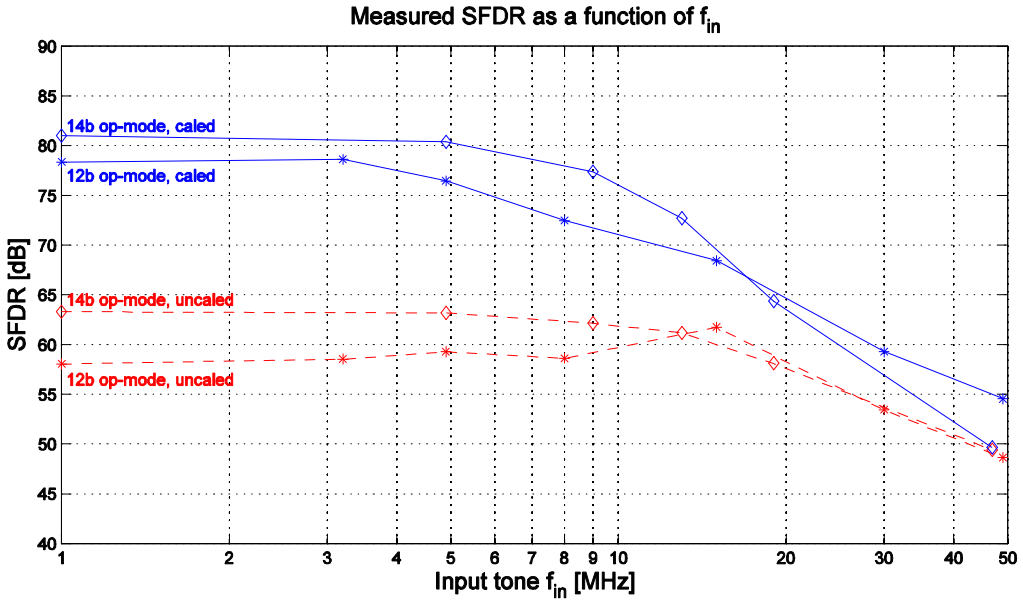


Figure 19.3-7. SFDR performance for 12b (sub-DAC B) and 14b op-mode (sub-DACs C, D, E, and F), before and after calibration, F_s=100MS/s.

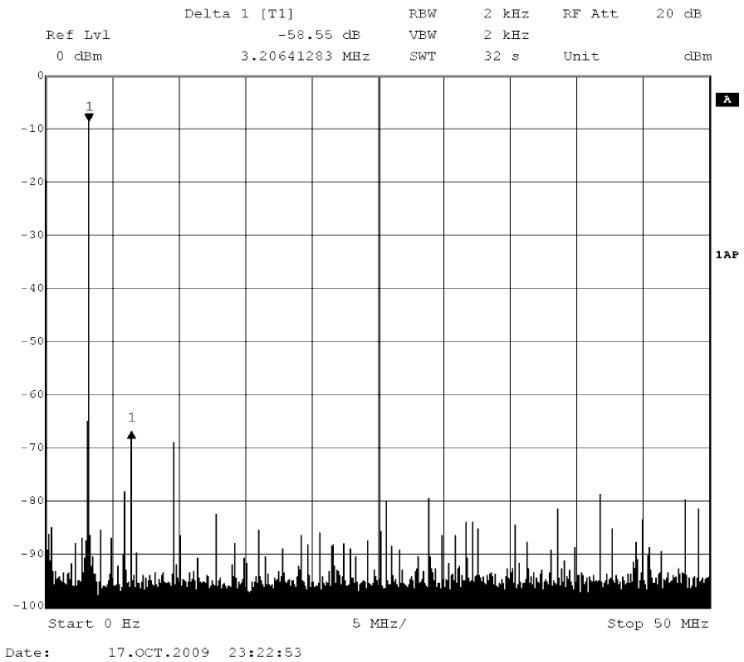


Figure 19.3-8. DAC output spectrum of 12b op-mode for f_{in}=3.2MHz, F_s=100MS/s, before calibration, SFDR=58.5dB.

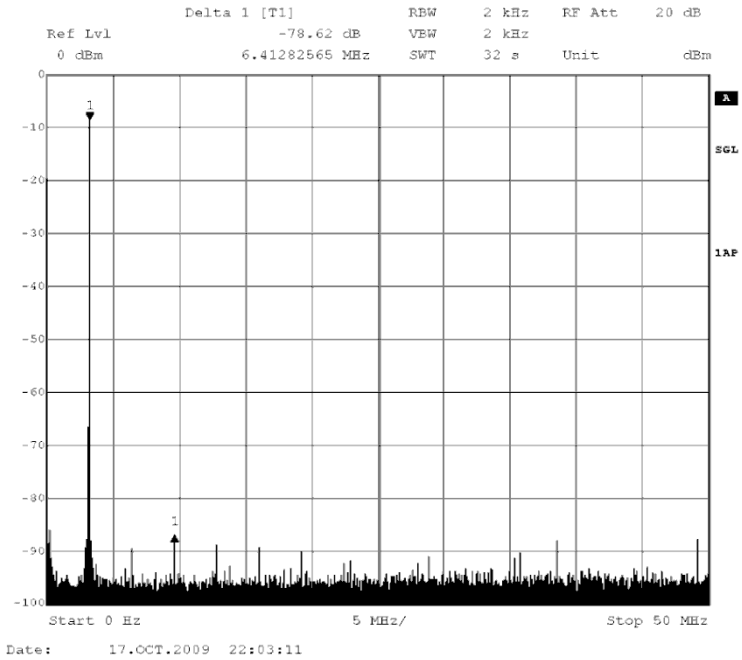


Figure 19.3-9. DAC output spectrum of 12b op-mode for f_{in}=3.2MHz, F_s=100MS/s, after calibration, SFDR=78.6dB.

Due to the large mismatch errors of sub-DAC B, the output signal contains many high order HD components, as shown in Figure 19.3-8. The dominant HD components at levels -60dB are HD2 and HD3, while many other HD components are spread over the whole Nyquist bandwidth at levels of about -70dB. The measured SFDR is about 58.5dB.

The calibration of the DAC currents corrects their mismatch errors and hence improves the dynamic linearity of the DAC, i.e. sub-DAC B in the given example for a 12b op-mode. Figure 19.3-9 shows the respective output spectrum after calibration. The SFDR is improved by about 20dB up to 78.6dB. Note that the calibration is preformed manually. The final accuracy is limited by the quantization error of the correction step, the measurement noise and the human factor.

Figure 19.3-10 shows the DAC output spectrum for the 14 bit “stacked” op-mode. Due to the stacking of the sub-DACs, their gain mismatch errors predominantly contribute to lower order HD component. As seen from the spectrum, the dominant distortion spur in the spectrum is given by HD2=-63.2dB.

The calibration of the DAC current sources removes their mismatch errors and hence corrects for the gain mismatch between the sub-DACs, too. Figure 19.3-11 shows the respective output spectrum after the calibration. The SFDR is improved by 17dB up to 80.4dB. Note that the final accuracy is limited by the quantization error of the correction step, the measurement noise and the human factor.

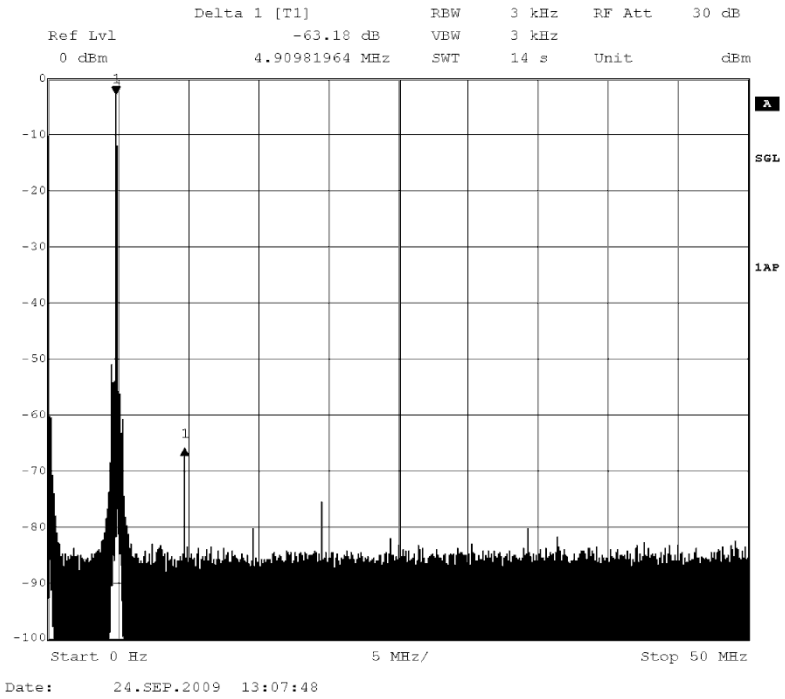


Figure 19.3-10. DAC output spectrum of 14b op-mode for f_{in}=5MHz, F_s=100MS/s, before calibration, SFDR=63.2dB.

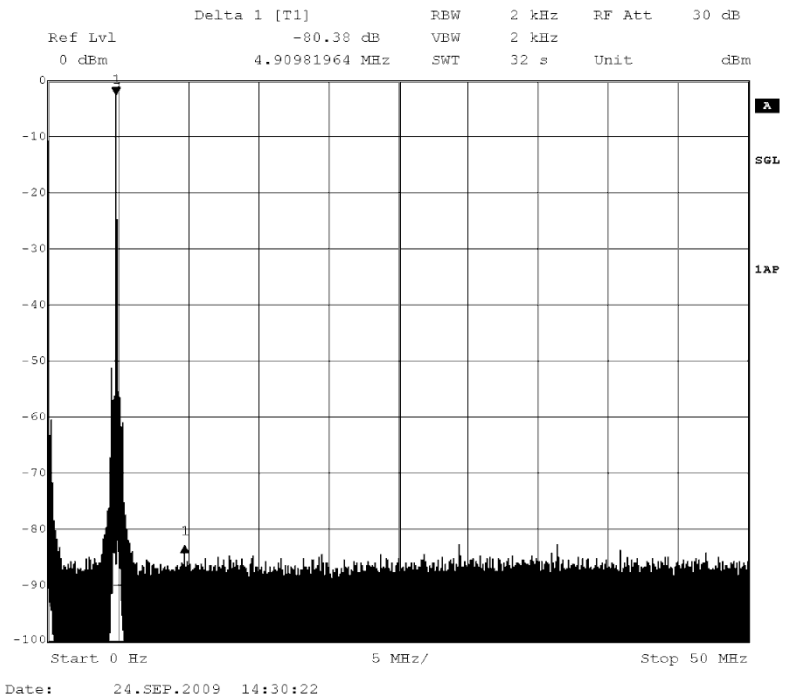


Figure 19.3-11. DAC output spectrum of 14b op-mode for f_{in}=5MHz, F_s=100MS/s, after calibration, SFDR=80.4dB.

To demonstrate and investigate the flexible op-modes, dynamic measurements are done on the 14 bit output of the flexible DAC platform, which includes the outputs of sub-DACs C, D, E, and F. Figure 19.3-12 shows the SFDR performance before calibration for various 12 bit, 13 bit, and 14 bit op-modes, based on “stacking” sub-DACs C, D, E, and F.

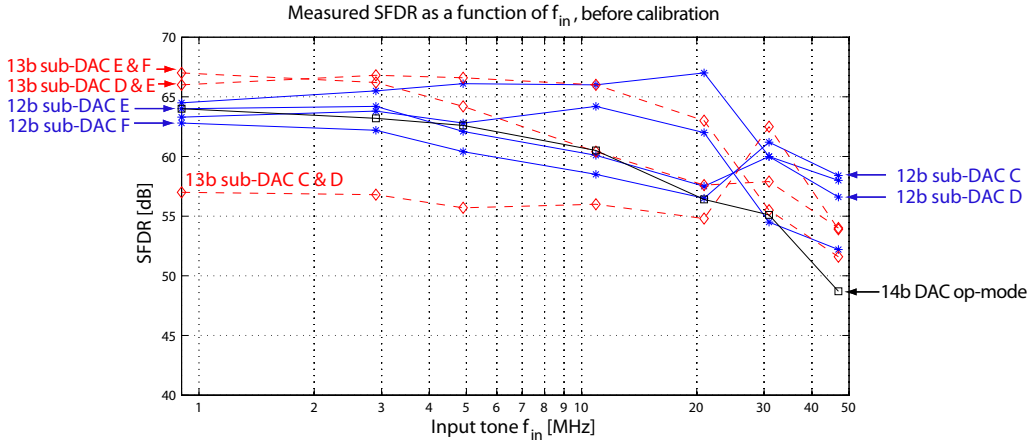


Figure 19.3-12. Measured SFDR performance before calibration for various 12 bit, 13 bit, and 14 bit op-modes based on “stacking” sub-DACs C, D, E, and F against f_{in}, at F_s=100MS/s.

At low f_{in}, the higher resolution op-modes generally deliver better SFDR, mainly due to averaging effects of the mismatch errors. At high f_{in}, the lower resolution op-modes generally deliver better SFDR, mainly due to experiencing less dynamic errors. The observed measurement results follow the general trend of the flexible op-modes that is demonstrated in chapter 18.

Figure 19.3-12 shows how using 4 sub-DACs, i.e. sub-DACs C, D, E, and F, the DAC test chip platform can generate 8 different performance patterns based on “stacking” sub-DACs op-modes. Note that this redundancy in performance can always be used in combination with correction methods to improve overall DAC performance.

For dynamic characterization at high-speeds, the sub-DACs are measured with a single tone input digital signal. Figure 19.3-13 shows a plot of the SFDR performance before calibration of a 12 bit sub-DAC A, clocked at F_s=650MS/s.

Fs=650MS/s is the maximal sampling rate that is available for the proper functionality of the measurement setup. However, other experiments, based on over-sampling, are done, which suggest potential functionality of the DAC test-chip, up to 1.5GS/s. Before calibration, the SFDR of sub-DAC A at low f_{in} is about 62dB.

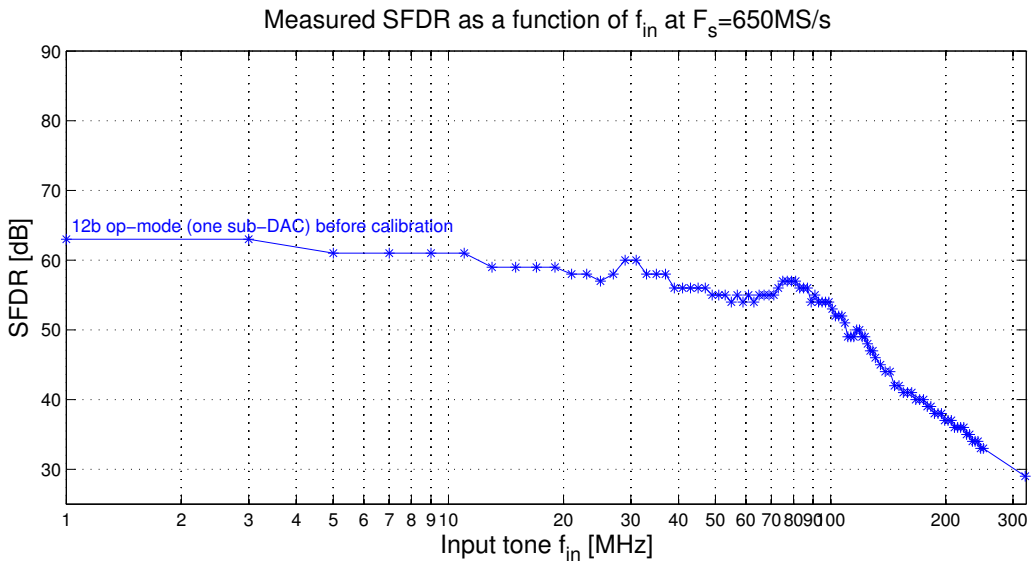


Figure 19.3-13. Measured SFDR performance of a 12 bit sub-DAC against f_{in}, at F_s=650MS/s.

19.4. CONCLUSIONS

The realization of a flexible DAC platform in 40nm CMOS technology is presented. The platform uses 16 sub-DAC units that are controlled by binary-to-unary functional decoder. The decoder allows a direct realization of high resolution op-modes based on "stacking" sub-DACs. The various op-modes of the flexible DAC platform deliver different performances, in terms of input resolution, static and dynamic linearity, power consumption, etc.

Measurement results show that at low input frequencies the high resolution op-modes deliver better SFDR performance than the low resolution op-modes. At high input frequencies, the low resolution op-modes deliver better SFDR performance than the high resolution op-modes.

Measurement results show that for high resolution op-modes via "stacking" of sub-DACs, the gain mismatch of these sub-DACs concentrates error energy predominantly into the low order non-linearity and hence predominantly generate low order HD components. The current calibration can correct both random and systematic mismatch errors. Therefore, it effectively improves the DAC performance for all op-modes.

PART VI: CONCLUSIONS

This part provides summary and general conclusions of the thesis. Future research directions are outlined.

20. Chapter

SUMMARY

This thesis presents research work that introduces and investigates correction methodologies and flexibility concepts for efficient and robust high-performance current-steering DACs.

The thesis considers performance limitations due to dominant DAC errors in chapters 5 and 6. A general model for DAC mismatch errors is proposed in chapter 5, which explains, at first order approximation level, the DAC performance deterioration at high input and sampling speeds. The DAC current amplitude errors are chosen for a further in-depth analysis in chapter 6.

To analytically and hence accurately model the DAC static non-linearity, the application of Brownian Bridge process is proposed in chapter 6. For the first time, it is proven that the INL properties of binary and unary DACs are not the same but rather they are different due to the different switching sequences of their analog elements. This difference is statistically quantified.

A viable approach to counteract the DAC mismatch errors is found in the DAC correction methods. This thesis considers the state-of-the-art of the DAC correction methods in chapters 3 and 4. The thesis provides a methodology for classifying and analyzing the available state-of-the-art knowledge in chapters 7 and 8. The provided classification in chapter 7 orders the knowledge, points to missing DAC correction methods, and provides clues to how to fill-in these gaps. The DAC correction method “calibration of currents” is chosen for a further in-depth analysis in chapter 8. Based on the provided general classification and in-depth analysis, new concepts and correction methods are proposed.

In chapter 9, this thesis proposes a new fundamental concept of segmenting the DAC analog resources – the functional segmentation. The functional segmentation creates high-level redundancy, which enables new DAC correction methods. This thesis also proposes the concept of correction level segmentation, which decouples the intrinsic DAC architecture from the correction method.

Based on these two new segmentation concepts, new self-calibration methods are proposed in chapter 10. Two new methods for calibration of both unary and binary currents are proposed. It is shown that efficiently calibrating all DAC unary and binary currents guarantees that the DAC accuracy depends only on design parameters, e.g. the correction step, and not on parameters of the manufacturing technology. These new contributions improve performance and efficiency, guarantee chip yield, and facilitate design migration to other manufacturing technologies.

In chapter 11, the concept of efficient redundancy is elaborated as an improvement of the available DAC correction methods using low-level mapping of unary currents. A new binary-to-thermometer decoder with redundant and loosely correlated switching sequences is proposed. The different switching sequences cause different distribution of DAC mismatch errors and hence different performance patterns. Since a significant redundancy is created for only little invested extra resources, the efficiency of the new redundant decoder as a correction method is very high.

Based on the new concept of functional segmentation, this thesis proposes new high-level mapping correction method in chapter 12. As the functional segmentation creates multiple DAC sub-functions, these sub-functions are used in a way that they mutually compensate their errors. It is shown that through high-level mapping, very efficient DAC designs become possible, since the algorithmic segmentation for each sub-DAC function can be still chosen to be binary.

In chapter 13, the new functional segmentation is further shown to be the basis for a new DAC correction method that suppresses HD components. The sub-DAC functions are provided with phase-shifted replicas of the input-signal and then the outputs are combined. The HD components compensate each other, while the power of the main signal increases.

In chapter 14, all discussions of efficiency, performance improvement and fabrication process independence, being supported by the concepts of functional and correction level segmentation are shown to converge into a new focal point – DAC flexibility. A new DAC architecture, based on multiple sets of parallel sub-DACs, is proposed. Different operating modes are described and analyzed.

To support and validate the proposed concepts three new current-steering DAC designs are presented in chapters 15, 16, 17, 18, and 19. The three DAC designs are implemented as test chips and measured.

The first test chip is a 12 bit self-calibrated DAC in 250nm CMOS technology. This test-chip features a fully-integrated self-calibration engine featuring novel self-measurement and self-correction circuits for unary currents. The measurement results show that self-calibration of currents can guarantee more than 14 bit accuracy. The size of the core layout block is about $1mm^2$.

The second test chip is a 12-to-14 bit flexible DAC platform, based on 4 12 bit sub-DAC cores, designed in 180nm CMOS technology. This is the first-ever DAC implementation that features a fully-integrated self-calibration engine for correcting all unary and binary currents. The accuracy of the test-chip depends only on the design parameter correction step. This test chip demonstrates in practice the advantages of the redundant binary-to-thermometer decoder. Furthermore, the test chip demonstrates DAC flexibility, being able to operate in different modes. This test-chip demonstrates how the power of its unused parts can be switched off and hence how flexible power consumption can be realized. The size of the core layout block per a 12 bit sub-DAC instance is about $0.2mm^2$.

The third test chip is a 12-to-16 bit flexible DAC platform, based on 16 12 bit sub-DACs, designed in 40nm CMOS technology. It is an example of an implementation of massive functional segmentation. The measured performance is of the order of the reported state-of-the-art DAC implementations for sampling rates of Fs>650MS/s. The test-chip further demonstrates the advantages of DAC flexibility: customer-tailored performance, availability of multiple DAC correction methods, etc. The size of the core layout block of a 12 bit sub-DAC instance is about $0.05mm^2$, which makes it the smallest 12 bit DAC implementation in the open literature at this time.

21. Chapter

CONCLUSIONS

With this thesis, the existing knowledge on DAC correction methods has been refined and advanced. The presented new concept of DAC flexibility enables DAC platforms that deliver flexible application-tailored performance, while featuring higher efficiency and robustness than previously possible. The proposed correction methods and concepts of flexibility have been validated in practice.

A new framework for modeling, analysis and classification of DAC errors and correction methods has led to new concepts and concrete DAC correction methods. It has been shown how the new concepts of functional and correction-level segmentation create new types of redundancy, both improving existing DAC correction methods and enabling conceptually new ones.

The proposed fully on-chip method for self-calibration of unary and binary currents has been validated in practice through three test-chip DAC implementations. These deliver more than 14 bit post-calibration accuracy, while the DAC cores feature compact and efficient design with pre-calibration intrinsic accuracy lower than 9 bit. It has been shown that the combined occupied silicon area of the self-calibration engine, the intrinsic current sources, and the correction circuits is smaller than the area that would be required to implement fabrication process-dependent 11-to-14 bits intrinsically accurate DAC current sources. The current post-calibration accuracy depends on the design parameter correction step and hence not on parameters of the fabrication process. The current post-calibration accuracy of the DAC test-chip implementations has been achieved for all measured test samples, with more than 2000 measured self-calibrated current sources. When all binary and unary currents are calibrated, it has been proven in practice that the DAC accuracy can be made independent of the tolerance parameters of the IC fabrication process (for a first order approximation analysis), just relying on a single design parameter - the correction step. A DAC test-chip implementation is presented that is the first self-calibrated DAC design that corrects all its binary and unary currents such that its post-calibration accuracy is dependent on the chosen correction step size. It has been shown also that the self-calibration of current amplitude errors improves both static and dynamic DAC performance. The robust performance of the self-calibration of currents has

been shown to deliver an SFDR>80dB, maintained up to f_{in}=5MHz, f_{in}=11MHz and f_{in}=5MHz, respectively for the three test-chip implementations in 250nm, 180nm, and 40nm CMOS processes. It has been demonstrated how through calibration and CMOS process scaling an aggressive silicon area reduction can be achieved. The silicon cores of the test-chip implementations, for the three test-chip implementations in 250nm, 180nm, and 40nm CMOS processes, per 12 bit sub-DAC instance, occupy 1mm^2, 0.2mm^2 and 0.05mm^2, respectively.

The new functional segmentation has been shown to enable a number of conceptually new DAC correction methods, e.g. high-level mapping and harmonic distortion suppression. These two methods have been validated in practice with a test-chip DAC implementation. It has been shown that the high-level mapping can deliver more than 14 bit linearity. It has also been shown that the method of harmonic distortion suppression can improve the measured SFDR of an OFDM-like multi-tone signal by more than 8dB, without the need for any error measurements.

An efficient low-level mapping concept has been implemented as a redundant binary-to-thermometer decoder with two switching sequences. This has been validated in practice. It has been shown through simulations that the redundant decoder can lead to 30% relaxation of the design specification and up to 17% improvement in chip yield. Through test chip measurements, an exemplary SFDR improvement of about 6dB has been shown. All this is achieved with very little increase in the occupied silicon area.

This thesis has demonstrated how the introduced concept of DAC flexibility can include multiple operation modes which deliver application-tailored solutions to the end-customers. DAC flexibility, in terms of flexible power consumption, DAC resolution, and dynamic linearity has been validated in practice through test chip measurements. It has also been shown how the DAC correction methods can be combined in a platform DAC architecture to provide flexibility. Two test-chip implementations demonstrate in practice the main flexible concepts – a 12-14 bit flexible 100MS/s DAC and a 12-16 bit flexible >650MS/s DAC. The specific challenges for the design of flexible DACs have been demonstrated. It is shown how co-integration between flexible digital SoC platforms, such as FPGAs, and the proposed flexible DAC platform can lead to mutual benefits.

APPENDIX A.

Published CMOS Digital-to-Analog converters from 1986 until 2009

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
1986	JSSC	4b		Triple DAC; Color Map for video; CMOS 2000nm
1986	JSSC	8b		Symmetrical switching sequence; segment.: 6u-2b; CMOS 2000nm
1987	JSSC	8b		3 high-speed DACs + RAM; 6-2 segmentation; band-gap reference current; matrix unary decoding; CMOS 1200nm
1987	JSSC	16b		monolithic, stereo, over-sampling, noise shaping; 1b SC D/A
1988	JSSC	16bit	79dB @1KHz	Current division small-signal reproduction, a dynamic segmentation technique, geometric averaging technique, averaging
1988	JSSC	15bit	SNDR= =75dB	algorithmic DAC based on interpolation, SC, RZ, CMOS 2500nm
1988	JSSC	10bit (12b)	60db @0.2MHz	Large on-chip integration SoC, CMOS 2500nm
1988	ISSCC	16bit	79dB @1KHz	Current division small-signal reproduction, a dynamic segmentation and geometric averaging technique, CMOS 2000nm
1988	CICC	10bit (11b)		Pipeline SC, CMOS 3000nm
1988	CICC	16bit (14b)	96dB @1kHz	SD DAC, DEM, CMOS 3000nm
1988	ISCAS	15b	63dB @4kHz	Resistor string and SC, CMOS 5000nm
1988	ISCAS	10bit (12b)	60db @0.2MHz	Large on-chip integration SoC, CMOS 2000nm
1989	JSSC	6b (8b)		current sink, glitch reduction, CMOS 3000nm
1989	JSSC	14b		CALIBRATION
1989	JSSC	17b	108dB @1kHz	PWM DAC, multi-stage noise shaping CMOS1500nm
1989	JSSC	15b	63dB @4kHz	Resistor string and SC, CMOS 5000nm
1989	JSSC	16b (15b)		Background calibration, coarse & fine currents, CMOS1600nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
1989	JSSC	10bit (11b)		Pipeline SC, CMOS 3000nm
1989	ISSCC	16b (15b)	SNDR= =91dB	Background calibration, coarse & fine currents, CMOS1600nm
1989	CICC	12b (10b)		turn on sequence, single power supply, CMOS 2000nm
1989	ISCAS	14b		CALIBRATION
1990	JSSC	10b	56dB @7MHz	dual-ladder resistor string, CMOS 1600nm
1990	ISSCC	18b	THD+N = =94dB @1kHz	laser trimming; R-2R; TTL/CMOS transistors; CMOS 2000nm
1990	ISSCC	10b		CMOS 1600nm
1990	CICC			EEPROM configurable DAC; CMOS 2000nm
1990	CICC			Algorithmic DAC for high-speed, CMOS 2500nm
1990	CICC			RAMDAC; RAM-DSP-DAC; CMOS 1200nm
1991	JSSC	8b	41dB @13.5MHz	deglitching circuits; matrix unary decoding; feedback for current stability, CMOS 1000nm
1991	JSSC	10b	59db @13.3MHz	current division; matching; special switching, CMOS 800nm
1991	JSSC		90dB @1kHz	(128) Oversampling, multi bit; calibration background; cal based on Cgs; N- and P- type sub-DACs, CMOS 1600nm
1991	JSSC	10b		hierarchical symmetrical switching; CMOS 1000nm
1991	ISSCC			5b oversampled; CMOS 1600nm
1991	CICC	10b		Glitchless; Fully unary; matix decoding; clocked AND-NOR gate; cascodes on top of the current switches; CMOS 1000nm
1991	CICC		HD=-63dB	Delta-sigma; SC DAC; CMOS 2000nm
1992	CICC			second-order SD; MASH; CMOS 1200nm
1993	JSSC		87dB @4kHz	oversampling (176) DA; semi-digital reconstruction filter; second order noise shaping, CMOS 1200nm
1993	ISSCC		91dB @2kHz	SD DAC; 3-b quantizer; DEM, OSR 128, CMOS 1000nm
1993	ISSCC		100dB @5kHz;	SD DAC; CMOS 1000nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
1993	CICC	9b	40dB @12MHz	Triple DAC; 4u+5b; CMOS 1000nm
1994	JSSC	8b (9b)	52dB @560kHz.	programmable interface; digital delay lines with programmable delays; digitally programmable gain; FIR Filtering; CMOS 1200nm
1994	JSSC	10b (11b)	56dB @3.9MHz	Threshold voltage compensated current sources; 2 stage architecture; symmetrical layout; CMOS 800nm
1994	JSSC	8b (9b)		Triple 8bit DAC; On-chip PLL; Hardware cursor; CMOS 800nm
1994	CICC	10b		Design methodology; Interpolative CS DAC, CMOS 2000nm
1994	CICC	10b	THD= -65dB @10kHz	Active compensation of parasitic capacitances; active output buffer, CMOS 1200nm;
1994	CICC	7b		R-2R array; Amplifier; CMOS 1200nm
1995	JSSC	10b (11b)	58,1dB @0,5kHz	segmented anti-symmetric switching sequence and an innovative asymmetrical switching buffer, CMOS 800nm
1995	JSSC		95dB @1kHz	SC, caps multiplexing; multi bit SD DAC; 2 stage AB opAmp for QVC buffer; CMOS 2000nm
1995	ISSCC		95dB @1kHz	Multi bit SD; Reduced OSR; 5-level SC DAC; CMOS 2000nm
1995	CICC	10b	60dB @5kHz	CS DAC; 5b-5u; on-chip gain error compensation; CMOS 1200nm
1995	CICC	8b		Delayed Driving Scheme; switching adjustment, CMOS 500nm
1996	JSSC		98dB @1kHz	on-chip digital phase-locked loop; A 2^{nd} order modulator with a multi bit quantizer, switched-capacitor (SC) DAC, CMOS 600nm
1996	JSSC		100dB @1kHz	8 OSR; Interpolation; DEM; digital filter; CMOS 600nm
1996	CICC			high yield; fully binary; CMOS 700nm
1997	ISSCC			triple rank architecture; CMOS 1000nm
1997	ESSCIRC	16b		DAC-ADC combo; SAR error correcting, CMOS 800nm
1997	CICC	10b	55dB @1.7MHz	Low distortion; low IM Distortion; 6b-4u, CMOS 600nm
1997	CICC	6b		Multi-input Floating gate MOS, CMOS 1200nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
1997	CICC		90dB @1.3kHz;	2nd order; high-res. SD SC DAC, CMOS 500nm
1998	JSSC			3rd order sigma-delta; immunity to clock; averaging algorithm; DSM; Low-power SC; CMOS 600nm
1998	JSSC		115dB @1kHz	6bit modulator anti sensitivity to clock jitter; noise shape segmentation; scrambling; dual RZ output; CMOS 600nm
1998	JSSC	10b (11b)	73dB @8MHz	DAC with special segmentation binary-unary; CMOS 350nm
1998	JSSC	12b	65dB @8MHz	Double segmentation; Design for high speed; CMOS 500nm
1998	ISSCC	10b	71dB	CMOS 500nm
1998	ISSCC		105dB @1kHz	Segmented Noise-Shaped Scrambling; dual RTZ; CMOS 600nm
1998	ISSCC		95dB @2kHz	Metal-metal charge transfer; SD DAC; CMOS 500nm
1998	ISSCC		90dB dynamic range	Small power consumption; SC DAC, CMOS 600nm
1998	ESSCIRC	8b (12b)	70dB @1MHz	symmetry and synchronization, CMOS 500nm
1998	CICC	12b	70dB @0.3MHz	Low Glitch, CMOS 500nm
1998	CICC	8b	60dB @1MHz	Reduced sample rate SD DAC; CMOS 500nm
1999	JSSC		85dB @15kHz	mismatch shaping; 16 unary currents, CMOS 1200nm
1999	JSSC	14b		pipeline; 6bit modulator; Cgs calibration; OSR 12; Calibrated at 3.91kHz; CMOS 500nm
1999	JSSC	8b	40dB @1kHz	Multi-Input Floating-Gate MOSFET, CMOS 1200nm
1999	JSSC	14b	84dB @0.5MHz	Q2 random walk; CMOS 500nm
1999	JSSC	14b	70dB @26MHz	RZ output stage; avoid switching transient; CMOS 800nm
1999	JSSC		115dB @80Hz	Sampling of the output; SD CS DAC; CMOS 600nm
1999	ISSCC		83dB dynamic range;	massive on-chip integration; SD DAC; CMOS 500nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
1999	ISSCC		120dB dynamic range	Multi bit SD audio DAC; CMOS 500nm
1999	ISSCC	14b	80dB @5.1MHz	RZ output stage; CMOS 800nm
1999	ISSCC	14b	84dB @500kHz	CS DAC; Q2 random walk; CMOS 500nm
1999	ESSCIRC	10b	62dB @35MHz	Segmentation 8b+2b; CMOS 200nm
1999	CICC		89dB	DEM, RDWA Rotated weighted averaging CMOS 2000nm
2000	JSSC	24b	80dB @20kHz	Multi bit 5b SD SC DAC + IIR, partial DEM, CMOS 500nm
2000	JSSC	14b	86dB @1kHz	SD DAC; DEM, OSR 64, CMOS 2000nm
2000	JSSC	14b	82dB @8.5MHz	background self-trimming; double segmentation; track/attenuate output stage; CMOS 350nm
2000	ISSCC		105dB @1kHz	Multi bit SD; DEM; CMOSO 350nm
2000	ISSCC	14b	82dB @8.5MHz	background self-trimming; CMSO 350nm
2000	ESSCIRC	14b	84dB @100KHz	Binary DAC; MSB currents calibrated; CMOS 180nm
2000	CICC	10b		CS DAC; CMOS 250nm
2001	JSSC	10b (11b)	70dB @100MHz	unary DAC;CMOS 350nm
2001	JSSC	14b	84dB @0.1MHz	4 MSB binary currents calibration, power efficiency; CMSO 180nm
2001	ISSCC	12b	62dB @125MHz	Triple centroid switching scheme, CMOS 350nm
2001	ESSCIRC	10b	70dB @24kHz;	Redundant-cell array; calibration; CMOS 250nm
2001	ESSCIRC			SC DAC; Filter; Multi bit SD; CMOS 350nm
2001	CICC	10b	67dB@1M	Small area, CMOS 350nm
2002	JSSC	14b		GMSK co-integration; CMSO 350nm
2002	JSSC	12b	51dB	differential quad switching; CMOS 250nm
2002	JSSC			For direct digital modulation systems; CMOS
2002	ISSCC	14b		co-integration with modulator; CMOS 350nm
2002	ESSCIRC	10b	58dB @1MHz	16tap voltage controlled delay; Linear interpolation; CMOS 350nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
2002	ESSCIRC			Pipeline DAC; Track and Hold; SC DAC; 2-C; CMOS 600nm
2003	JSSC	14b	76dB @10MHz	Trimming using analog charge on floating pFETs, CMOS 180nm
2003	JSSC	12b (13b)	81dB @10MHz	co-integration with digital quadrature modulator; switching technique to reduce switching spurs, CMOS 350nm
2003	JSSC	14b	82dB @0.9MHz	CALIBRATION, parallel CALDAC, CMOS 130nm
2003	JSSC	14b	77dB @20MHz	floating gate trimming; CMOS 250nm, CMOS 180nm
2003	ISSCC	16b	80dB @20MHz	Calibration CALDAC attached to CS; CMOS 250nm
2003	ISSCC			SD DAC; 5^{th} order feed-forward loop; 1b quantizer; CMOS 180nm
2003	ISSCC	14b	81dB @1.4MHz	Calibration/parallel 1 CALDAC; CMOS 130nm
2003	ESSCIRC	6b	40dB	2 parallel DACs clocked in anti-phase to cancel the image band, CMOS 180nm
2003	CICC		75dB	DAC controlled by an oscillating waveform; CMOS 180nm
2004	JSSC		83dB	combo with FIR; SD DAC; inclusive digital mixers to convert baseband I and Q signals to IF, CMOS 250nm
2004	JSSC	12b	70dB @20MHz	7 MSBs in unary coding; design for layout; CMOS 180nm;
2004	JSSC	6b (7b)	40dB	Parallel-Path DACs for Nyquist Signal Generation, CMOS 180nm
2004	JSSC		75dB	The DAC output is controlled by an oscillating waveform; SD DAC; RF DAC; CMOS 180nm
2004	JSSC	13b	80dB	Oversampled DAC with semi-digital reconstruction Filter; SC DAC; OSR 16; CMOS 180nm
2004	ISSCC		100dB @1kHz	PWM DAC; CMOS 350nm
2004	ISSCC	13b		Oversampled DAC with semi-digital reconstruction Filter; SC DAC; OSR 16; CMOS 180nm
2004	ISSCC	14b	72dB @20MHz	Floating current cell; RZ; background calibration; CMOS 180nm
2004	ISSCC	10b	67dB @120MHz	Binary DAC; CMOS 180nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
2004	ISSCC			Real-time DEM; SC CT DAC; CMOS 180nm
2004	ISSCC	14b	67dB @260MHz	segmentation 6u-8b; CMOS 180nm
2004	ESSCIRC	6b	85dB @0.3MHz	OSR; 2nd order noise shaping; Output buffer; N-DAC and P-DAC combo; CMOS 130nm
2004	ESSCIRC	14b	82dB @1MHz	Adjustable INL; CMOS 250nm
2004	CICC	10b		area efficient; R-2R + SC, CMOS 250nm
2005	JSSC		80dB @1kHz;	trade off: area-power-SNR; single op-amp SC DAC, CMOS 130nm
2005	ISSCC		77dB	Low power; 2nd order multi bit SD DAC;OSR10/20/50; CMOS 90nm
2005	ISSCC	12b	70dB @120MHz	CML logic; current switches cascodes; 6b-6u; CMOS 180nm
2005	ESSCIRC	12b	81dB @5MHz	Calibration; CALDAC in parallel to CS; Full integration; start-up; CMOS 250nm
2005	CICC		94dB	SD DAC with reduced activity data weighted averaging; anti-jitter filter. 4th order loop; CMOS 130nm
2006	JSSC	10b (12b)	60db @122MHz	binary DAC; CMOS 180nm
2006	JSSC		82dB?	Oversampled DAC with Multi bit semi digital filtering and boosted subcarrier SNR, CMOS 180nm
2006	ISSCC		78dB	1b 3rd order CT SD DAC with return-to-open;OSR64; CMOS180nm
2006	ISSCC	14b	80dB @35MHz	Two parallel DACs with full data scrambling, CMSO 180nm
2006	ISSCC	6b	50dB	CML circuits; CMOS 130nm
2006	ESSCIRC	8b	68dB @12MHz	Full unary approach; CMOS 130nm
2006	CICC	10b	72dB @42MS/s	CML; CMOS 90nm
2006	CICC	10b	61dB @50MHz	R-2R + CS; CMOS 180nm
2006	CICC	4b		Binary weighted resistor DAC; tunable floating gate CMOS resistors, CMOS 500nm
2006	CICC	20b	90dB@1k	CMOS 130nm
2006	CICC	14b	70dB @50MHz	Switch gate drive; active second cascode; CALDAC to the drain CS; CMOS 130nm

Year	Source, reference	Resolution (Accuracy)	SFDR@f_{in}, linearity	Main features, CMOS technology
2007	JSSC	12b	74dB @1MHz	turn on sequence, single power supply, CMOS 90nm
2007	JSSC	14b	78dB @2MHz	Pre-processing; mapping; CMOS 180nm
2007	ISSCC	13b	75dB @10MHz	background calibration; randomized cal. Cycle; CMOS 130nm
2007	CICC	20b		SD DAC; mixed radix number representation CMOS 130nm
2007	CICC		94dB @2kHz	Noise shaping DAC; 5th order multi bit CMOS 350nm
2008	JSSC	8b	51dB @750MHz	12GS/s, CMOS 90nm
2009	ISSCC	12b	70dB @230MHz	Switch-cascode with bleeding currents, 2.9GS/s, CMOS65nm

REFERENCES

[1] K. Doris, "High-speed D/A Converters: from Analysis and Synthesis Concepts to IC Implementation," in *Eindhoven University of Technology, Faculty of Electrical Engineering*. vol. PhD thesis degree Eindhoven: Eindhoven University of Technology, 2004.

[2] A. v. d. Bosch, M. Steyaert, and W. Sansen, "Static and dynamic performance limitations for high speed D/A converters," *Kluwer Academic Publishers,* 2004.

[3] R. v. d. Plassche, *Integrated Analog-to-digital and digital-to-analog converters*: Kluwer Academic Publishers, 1994.

[4] Maxim, "INL/DNL Measurements for High-Speed Analog-to-Digital Converters (ADCs), AN283," 2001.

[5] K. Doris, Roermund, A.H.M. van, Leenaerts, D.M.W. , *Wide-Bandwidth High Dynamic Range D/A Converters*: Springer, 2006.

[6] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *Solid-State Circuits, IEEE Journal of,* vol. 24, pp. 1433-1439, 1989.

[7] B. Razavi, *Design of Analog CMOS Integrated Circuits*: Mc Graw Hill, 2001.

[8] J. Wikner, "Studies on CMOS Digital-to-Analog Converters." vol. PhD: Linkoping University, 2001.

[9] J. Deveugele and M. S. J. Steyaert, "A 10 bit 250-MS/s binary-weighted current-steering DAC," *Solid-State Circuits, IEEE Journal of,* vol. 41, pp. 320-329, 2006.

[10] A. Van den Bosch, M. Steyaert, and W. Sansen, "An accurate statistical yield model for CMOS current-steering D/A converters," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, 2000, pp. 105-108 vol.4.

[11] P. Harpe, J. M. Meulmeester, A. J. Hegt, and A. van Roermund, "Novel Digital Pre-Correction Method For Mismatch in DACs with Built-In-Self Measurement," *Proceedings of IEE ADDA 2005,* 2005.

[12] L. Chi-Hung and K. Bult, "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm²," *Solid-State Circuits, IEEE Journal of,* vol. 33, pp. 1948-1958, 1998.

[13] K. Khanoyan, F. Behbahani, and A. A. Abidi, "A 10 b, 400 MS/s glitch-free CMOS D/A converter," in *VLSI Circuits, 1999. Digest of Technical Papers. 1999 Symposium on*, 1999, pp. 73-76.

[14] P. G. A. Jespers, *Integrated Converters: D to A and A to D Architectures*: Oxford University Press, 2001.

[15] S. Dongwon, A. Weil, and M. Feng, "A 14 bit, 1 GS/s digital-to-analog converter with improved dynamic performances," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, 2000, pp. 541-544 vol.5.

[16] M. Vertregt, "The analog challenge of nanometer CMOS," in *Electron Devices Meeting, 2006. IEDM '06. International*, 2006, pp. 1-8.

[17] A. Abrial, J. M. Fournier, P. Senn, J. Bouvier, and A. Lelah, "A triple low-power voltage output video 8 bit CMOS DAC," in *Circuits and Systems, 1988., IEEE International Symposium on*, 1988, pp. 2825-2828 vol.3.

[18] J. M. Fournier and P. Senn, "A 130-MHz 8-b CMOS video DAC for HDTV applications," *Solid-State Circuits, IEEE Journal of,* vol. 26, pp. 1073-1077, 1991.

[19] B. Schafferer and R. Adams, "A 3V CMOS 400mW 14b 1.4GS/s DAC for multi-carrier applications," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, 2004, pp. 360-532 Vol.1.

[20] C.-H. Lin, F. van der Goes, J. Westra, J. Mulder, Y. Lin, E. Arslan, E. Ayranci, X. Liu, and K. Bult, "A 12b 2.9GS/s DAC with IM3 <-60dBc Beyond 1GHz in 65nm CMOS," in *Solid-State Circuits Conference, 2009. Digest of Technical Papers. ISSCC. 2009 IEEE International*, 2009.

[21] P. J. Quinn, "High-accuracy switched-capacitor techniques applied to filter and ADC design." vol. PhD Eindhoven: Eindhoven University of Technology, 2006.

[22] R. H. Walden, "Performance trends for analog to digital converters," *Communications Magazine, IEEE,* vol. 37, pp. 96-101, 1999.

[23] G. I. Radulov, M. Heydenreich, R. W. van der Hofstad, A. J. Hegt, and H. M. A. van Roermund, "Brownian-Bridge-Based Statistical Analysis of the DAC INL Caused by Current Mismatch," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 54, pp. 146-150, 2007.

[24] B. Murmann, "A/D converter trends: Power dissipation, scaling and digitally assisted architectures," in *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*, 2008, pp. 105-112.

[25] B. Murmann, C. Vogel, and H. Koeppl, "Digitally enhanced analog circuits: System aspects," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, 2008, pp. 560-563.

[26] A. R. Bugeja, B. S. Song, P. L. Rakers, and S. F. Gillig, "A 14-b, 100-MS/s CMOS DAC designed for spectral performance," *Solid-State Circuits, IEEE Journal of,* vol. 34, pp. 1719-1732, 1999.

[27] M. Clara, W. Klatzer, D. Gruber, A. Marak, B. Seger, and W. Pribyl, "A 1.5V 13bit 130-300MS/s self-calibrated DAC with active output stage and 50MHz signal bandwidth in 0.13μm CMOS," in *Solid-State Circuits Conference, 2008. ESSCIRC 2008. 34th European*, 2008, pp. 262-265.

[28] C. Myung-Jun, B. Kwang-Hyun, and M. Teshome, "A 1.6-GS/s 12 bit return-to-zero GaAs RF DAC for multiple Nyquist operation," *Solid-State Circuits, IEEE Journal of,* vol. 40, pp. 2456-2468, 2005.

[29] P. Sungkyung, K. Gyudong, P. Sin-Chong, and K. Wonchan, "A digital-to-analog converter based on differential-quad switching," *Solid-State Circuits, IEEE Journal of,* vol. 37, pp. 1335-1338, 2002.

[30] C. Kok Lim and I. Galton, "A 14b 100MS/s DAC with Fully Segmented Dynamic Element Matching," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, 2006, pp. 2390-2399.

[31] C. Kok Lim, Z. Jianyu, and I. Galton, "A 150MS/s 14 bit Segmented DEM DAC with Greater than 83dB of SFDR Across the Nyquilst band," in *VLSI Circuits, 2007 IEEE Symposium on*, 2007, pp. 200-201.

[32] C. Kok Lim, Z. Jianyu, and I. Galton, "Dynamic Element Matching to Prevent Nonlinear Distortion From Pulse-Shape Mismatches in High-Resolution DACs," *Solid-State Circuits, IEEE Journal of,* vol. 43, pp. 2067-2078, 2008.

[33] C. Kok Lim, N. Rakuljic, and I. Galton, "Segmented Dynamic Element Matching for High-Resolution Digital-to-Analog Conversion," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 55, pp. 3383-3392, 2008.

[34] D. W. J. Groeneveld, H. J. Schouwenaars, H. A. H. Termeer, and C. A. A. Bastiaansen, "A self-calibration technique for monolithic high-resolution D/A converters," *Solid-State Circuits, IEEE Journal of,* vol. 24, pp. 1517-1522, 1989.

[35] G. I. Radulov, P. J. Quinn, H. Hegt, and A. van Roermund, "An on-chip self-calibration method for current mismatch in D/A converters," in *Solid-State Circuits Conference, 2005. ESSCIRC 2005. Proceedings of the 31st European*, 2005, pp. 169-172.

[36] Y. Cong and R. L. Geiger, "A 1.5-V 14 bit 100-MS/s self-calibrated DAC," *Solid-State Circuits, IEEE Journal of,* vol. 38, pp. 2051-2060, 2003.

[37] T. Yongjian, H. Hegt, and A. van Roermund, "DDL-based calibration techniques for timing errors in current-steering DACs," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, p. 4 pp.

[38] L. R. Carley and J. Kenney, "A 16 bit 4'th order noise-shaping D/A converter," in *Custom Integrated Circuits Conference, 1988., Proceedings of the IEEE 1988*, 1988, pp. 21.7/1-21.7/4.

[39] W. G. Bliss, C. E. Seaberg, and R. L. Geiger, "A very small sub-binary radix DAC for static pseudo-analog high-precision memory," in *Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on*, 1992, pp. 425-428 vol.1.

[40] M. Pastre and M. Kayal, "High-precision DAC based on a self-calibrated sub-binary radix converter," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, 2004, pp. I-341-I-344 Vol.1.

[41] P. Harpe, "Smart A/D and D/A converters." vol. PhD Eindhoven: Eindhoven University of Technology, 2010.

[42] G. R. Spalding and R. L. Geiger, "Digital Correction For Improved Spectral Response In Signal Generation Systems," in *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, 1993, pp. 132-135.

[43] K. J. Riley, D. M. Hummels, F. H. Irons, and A. Rundell, "Dynamic compensation of digital to analog converters," in *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE*, 1999, pp. 1310-1315 vol.2.

[44] TI and GC5322, "GC5322 Wideband digital predistortion transmit processor," Texas Instruments datasheet, www.ti.com, 2008.

[45] E. Lopelli, J. D. van der Tang, and A. H. M. van Roermund, "A 1 mA Ultra-Low-Power FHSS TX Front-End Utilizing Direct Modulation With Digital Pre-Distortion," *Solid-State Circuits, IEEE Journal of,* vol. 42, pp. 2212-2223, 2007.

[46] J. Bastos, *Characterization of MOS Transistor Mismatch for Analog Design*: Katholieke Universiteit Leuven, 1998.

[47] G. I. Radulov, P. J. Quinn, H. Hegt, and A. van Roermund, "A flexible 12 bit self-calibrated quad-core current-steering DAC," in *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*, 2008, pp. 25-28.

[48] R. v. d. H. Markus Heydenreich, Georgi Radulov "FUNCTIONALS OF BROWNIAN BRIDGES ARISING IN THE CURRENT MISMATCH IN D/A CONVERTERS," *Probability in the Engineering and Informational Sciences* 2009.

[49] J. Vandenbussche, G. Van der Plas, A. Van den Bosch, W. Daems, G. Gielen, M. Steyaert, and W. Sansen, "A 14 b 150 Msample/s update rate Q² random walk CMOS DAC," in *Solid-State Circuits Conference, 1999. Digest of Technical Papers. ISSCC. 1999 IEEE International*, 1999, pp. 146-147.

[50] B. Greenley, R. Veith, C. Dong-Young, and M. Un-Ku, "A low-Voltage 10 bit CMOS DAC in 0.01-mm² die area," *Circuits and Systems II: Express Briefs, IEEE Transactions on,* vol. 52, pp. 246-250, 2005.

[51] K. R. Lakshmikumar, R. A. Hadaway, and M. A. Copeland, "Characterisation and modeling of mismatch in MOS transistors for precision analog design," *Solid-State Circuits, IEEE Journal of,* vol. 21, pp. 1057-1066, 1986.

[52] C. S. G. Conroy, W. A. Lane, M. A. Moran, K. R. Lakshmikumar, M. A. Copeland, and R. A. Hadaway, "Comments, with reply, on `Characterization and modeling of mismatch in MOS transistors for precision analog design'," *Solid-State Circuits, IEEE Journal of,* vol. 23, pp. 294-296, 1988.

[53] Y. Cong and R. L. Geiger, "Formulation of INL and DNL yield estimation in current-steering D/A converters," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, 2002, pp. III-149-III-152 vol.3.

[54] M. Kosunen, J. Vankka, F. Teikari, and K. Halonen, "DNL and INL yield models for a current-steering D/A converter," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, 2003, pp. I-969-I-972 vol.1.

[55] G. L. Radulov, P. J. Quinn, P. C. W. van Beek, J. A. Hegt, and A. H. M. van Roermund, "A binary-to-thermometer decoder with built-in redundancy for improved DAC yield," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, p. 4 pp.

[56] A. van Roermund, H. Hegt, P. Harpe, G. Radulov, A. Zanikopoulos, K. Doris, and P. Quinn, "Smart AD and DA converters," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, 2005, pp. 4062-4065 Vol. 4.

[57] I. S. 746-1984, "Performance Measurements of A/D and D/A Converters for PCM Television Video Circuits," IEEE, Ed. New York, 1984.

[58] P. Billingsley, *Convergence of Probability Measures*.

[59] M. Heydenreich, R. v. d. Hofstad, and G. Radulov, "Functionals of Brownian bridges arising in the current mismatch in D/A-converters," *Eurandom Report 2006-016,* vol. http://arxiv.org/abs/math.PR/0606584, 2006.

[60] K. L. Chan, N. Rakuljic, and I. Galton, "Segmented Dynamic Element Matching for High-Resolution Digital-to-Analog Conversion," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 55, pp. 3383-3392, 2008.

[61] E. Mensink, E. A. M. Klumperink, and B. Nauta, "Distortion cancellation by polyphase multipath circuits," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 52, pp. 1785-1794, 2005.

[62] S. Ouzounov, E. Roza, J. A. Hegt, G. van der Weide, and A. H. M. van Roermund, "A CMOS V-I converter with 75-dB SFDR and 360-μW power consumption," *Solid-State Circuits, IEEE Journal of,* vol. 40, pp. 1527-1532, 2005.

[63] E. van Tuijl, J. van den Homberg, D. Reefman, C. Bastiaansen, and L. van der Dussen, "A 128f/sub s/ multi bit /spl Sigma//spl Delta/ CMOS audio DAC with real-time DEM and 115dB SFDR," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, 2004, pp. 368-369 Vol.1.

[64] G. L. Radulov, P. J. Quinn, P. Harpe, H. Hegt, and A. van Roermund, "Parallel current-steering D/A Converters for Flexibility and Smartness," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, 2007, pp. 1465-1468.

[65] C. Tao and G. G. E. Gielen, "A 14 bit 200-MHz Current-Steering DAC With Switching-Sequence Post-Adjustment Calibration," *Solid-State Circuits, IEEE Journal of,* vol. 42, pp. 2386-2394, 2007.

[66] A. R. Bugeja and S. Bang-Sup, "A self-trimming 14-b 100-MS/s CMOS DAC," *Solid-State Circuits, IEEE Journal of,* vol. 35, pp. 1841-1852, 2000.

[67] H. Qiuting, P. A. Francese, C. Martelli, and J. Nielsen, "A 200MS/s 14b 97mW DAC in 0.18/spl mu/m CMOS," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, 2004, pp. 364-532 Vol.1.

[68] Y. Tang, "Private discussions," Eindhoven 2008.

[69] M. Clara, W. Klatzer, B. Seger, A. Di Giandomenico, and L. Gori, "A 1.5V 200MS/s 13b 25mW DAC with Randomized Nested Background Calibration in 0.13/spl mu/m CMOS," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, 2007, pp. 250-600.

[70] M. P. Tiilikainen, "A 14 bit 1.8-V 20-mW 1-mm² CMOS DAC," *Solid-State Circuits, IEEE Journal of,* vol. 36, pp. 1144-1147, 2001.

[71] W. Schofield, D. Mercer, and L. S. Onge, "A 16b 400MS/s DAC with <-80dBc IMD to 300MHz and <-160dBm/Hz noise power spectral density," in *Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International*, 2003, pp. 126-482 vol.1.

[72] D. A. Mercer, "Low Power Approaches to High Speed CMOS Current Steering DACs," in *Conference 2006, IEEE Custom Integrated Circuits*, 2006, pp. 153-160.

[73] G. I. Radulov, "Design of a calibrated 12 bit current-steering Digital-to-Analog Converter," *PDEng thesis, Stan Ackermans Institute at Eindhoven University of Technology,* January, 2004 2004.

[74] G. I. Radulov, A. J. Hegt, P. J. Quinn, and A. van Roermund, "A calibration algorithm for current-steering DACs with relaxed design requirements," *Proceedings of the 9th Biennial Baltic Electronics Conference, BEC 2004, Tallin,* October 3-6, 2004 2004.

[75] G. I. Radulov, P. J. Quinn, A. J. Hegt, and A. van Roermund, "A self-calibrating current-steering 12 bit DAC based on a new 1 bit self test scheme," *Proceedings of IEEE IC Test Workshop, Limerick, Ireland,* pp. 49-54, 13-14 September 2004 2004.

[76] G. Radulov, P. Quinn, A. J. Hegt, and A. van Roermund, "Method and apparatus for calibrating a current-based circuit " in *US Patent 7,076,384*: Xilinx, 2006.

[77] G. I. Radulov, P. J. Quinn, J. A. Hegt, and A. H. M. van Roermund, "A start-up calibration method for generic current-steering D/A converters with optimal area solution," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, 2005, pp. 788-791 Vol. 1.

[78] A. T. K. Tang and C. Toumazou, "Novel self-calibrated high-speed D/A converter using trimmable current sources," in *Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on*, 1994, pp. 469-472 vol.5.

[79] G. L. Radulov, P. J. Quinn, P. Harpe, H. Hegt, and A. van Roermund, "A flexible 12 bit self-calibrated quad-core current-steering DAC," in *2008 IEEE Asia Pacific Conference on Circuits and Systems - APCCAS 2008*, Macau, 2008.

[80] G. Radulov, P. Quinn, A. J. Hegt, and A. van Roermund, "Method and apparatus for calibrating a scaled current electronic circuit," in *US Patent 7,466,252*: Xilinx, 2008.

[81] J. Deveugele and M. Steyaert, "A 10b 250MS/s binary-weighted current-steering DAC," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, 2004, pp. 362-532 Vol.1.

[82] D. W. J. Groeneveld and H. J. Schouwenaars, "A dual 3.4V bitstream continuous calibration CMOS D/A converter with 110 dB dynamic range," in *Advanced A-D and D-A Conversion Techniques and their Applications, 1994. Second International Conference on*, 1994, pp. 42-47.

[83] A. Papoulis, *PROBABILITY AND STATISTICS* Prentice Hall, 2002.

[84] R. van de Plassche, "Dynamic element matching for high-accuracy monolithic D/A converters," in *Solid-State Circuits Conference. Digest of Technical Papers. 1976 IEEE International*, 1976, pp. 148-149.

[85] H. T. Jensen and I. Galton, "An analysis of the partial randomization dynamic element matching technique," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on,* vol. 45, pp. 1538-1549, 1998.

[86] T. Chen, P. Geens, G. Van der Plas, W. Dehaene, and G. Gielen, "A 14 bit 130-MHz CMOS current-steering DAC with adjustable INL," in *Solid-State Circuits Conference, 2004. ESSCIRC 2004. Proceeding of the 30th European*, 2004, pp. 167-170.

[87] T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, and Y. Horiba, "An 80-MHz 8 bit CMOS D/A converter," *Solid-State Circuits, IEEE Journal of,* vol. 21, pp. 983-988, 1986.

[88] W. G. J. Schofield, "Thermometer coding circuitry." vol. US6163283, U. S. Patent, Ed. USA: Fujitsu Microelectronics Europe, 2000.

[89] K. Doris, J. Briaire, D. Leenaerts, M. Vertregt, and A. van Roermund, "A 12b 500MS/s DAC with >70dB SFDR up to 120MHz in 0.18um CMOS," in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, 2005.

[90] T. Yongjian, H. Hegt, A. van Roermund, K. Doris, and J. Briaire, "Statistical Analysis of Mapping Technique for Timing Error Correction in Current-Steering DACs," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, 2007, pp. 1225-1228.

[91] R. Shrestha, E. A. M. Klumperink, E. Mensink, G. J. M. Wienk, and B. Nauta, "A Polyphase Multipath Technique for Software-Defined Radio Transmitters," *Solid-State Circuits, IEEE Journal of,* vol. 41, pp. 2681-2692, 2006.

[92] E. Mensink, E. A. M. Klumperink, and B. Nauta, "Distortion cancellation via polyphase multipath circuits," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, 2004, pp. I-1098-101 Vol.1.

[93] R. Hoven, Hegt, van Roermund, "A parallel current-steering DAC architecture for flexible and improved performance," in *ProRisc* Veldhoven, the Netherlands, 2005.

[94] R. A. T. v. d. Hoven, "A pre-correction method for improved static linearity using parallel DACs." vol. MSc. Eindhoven: Eindhoven University of Technology, 2006.

LIST OF PUBLICATIONS

1. “Design of a calibrated 12 bit current-steering Digital-to-Analog Converter”, G.I. Radulov, PDEng thesis, Stan Ackermans Institute, Eindhoven University of Technology, January 2004.
2. “A self-calibrated current-steering 12 bit DAC based on new 1 bit self-test scheme”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, IEEE IC Test Workshop, ICTW, 13th – 14th September 2004, Limerick, Ireland; pp. 49-54, ISBN 0-9548342-0-8
3. “A calibration algorithm for current-steering DACs with relaxed design requirements”, G.I. Radulov, J.A. Hegt, P.J. Quinn, A.H.M. van Roermund, IEE 9th Biennial Baltic Electronics Conference, BEC04, 3rd - 6th October 2004, Tallinn, Estonia, ISBN 9985-59-462-2.
4. “A new current calibration scheme suitable for generic DAC architectures”, G.I. Radulov, J.A. Hegt, P.J. Quinn, A.H.M. van Roermund, ProRisc2004, Veldhoven, the Netherlands;
5. “A start-up calibration method for generic current-steering D/A converters with optimal area solution”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, IEEE ISCAS05, Kobe 23rd – 26th May 2005, Japan.
6. “An on-chip self-calibration method for current mismatch in D/A Converters”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, IEEE ESSCIRC September, 12th – 16th, 2005, Grenoble, France.
7. “A parallel current-steering DAC architecture for flexible and improved performance”, R.A.T. van den Hoven, G.I. Radulov, J.A. Hegt, A.H.M. van Roermund, ProRisc2005, Veldhoven, the Netherlands;
8. “A Binary-to-Thermometer Decoder with Built-in Redundancy for Improved DAC Yield”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, IEEE ISCAS May, 21st – 24th , 2006, Kos, Greece.
9. “Brownian Bridge based statistical analysis of the DAC INL caused by current mismatch”, Georgi I. Radulov, Markus Heydenreich, Remco W. van der Hofstad, Johannes A. Hegt, Arthur H.M. van Roermund, IEEE Transactions on Circuits and Systems II: Express Briefs, Volume 54, Issue 2, Feb. 2007 Page(s):146 – 150.
10. "Functionals of Brownian bridges arising in the current mismatch in D/A-converters", M. Heydenreich, R. van der Hofstad and G. Radulov, Eurandom Report 2006-016, June 2006, available at http://arxiv.org/abs/math.PR/0606584. The paper is submitted to the journal “Probability in the Engineering and Informational Sciences”
11. "Parallel current-steering D/A Converters for Flexibility and Smartness", G.I. Radulov, P.J. Quinn, Pieter Harpe, P.J.A. Hegt, A.H.M. van Roermund, IEEE ISCAS May, 27th – 30th , 2007, New Orleans, USA.
12. “A Binary-To-Thermometer Decoder with redundant switching sequences”, G.I. Radulov, J.A. Hegt, P.J. Quinn, A.H.M. van Roermund, ProRisc2006, Veldhoven, the Netherlands;

13. “Parallel current-steering D/A Converters for Flexibility and Smartness”, G.I. Radulov, P.J. Quinn, P.Harpe, J.A. Hegt, A.H.M. van Roermund, ProRisc2007, Veldhoven, the Netherlands;

14. “Parallel current-steering D/A Converters for Flexibility and Smartness”, G.I. Radulov, P.J. Quinn, P.Harpe, J.A. Hegt, A.H.M. van Roermund, SIREN2007, Delft, the Netherlands;

15. “A flexible 12 bit self-calibrated quad-core current-steering DAC”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, 2008 IEEE ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS - APCCAS 2008, Macau, China;

16. “Functionals of Brownian bridges arising in the current mismatch in D/A-converters”, Markus Heydenreich, Remco van der Hofstad, Georgi Radulov, February 22, 2008, Probability in the Engineering and Informational Sciences, 23(1), 149-172.; http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=2681224 ;

17. “Smart and flexible DACs, classification and design”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, Advances in Analog Circuit Design - AACD 2009, March 31- April 2.

18. “Analog Circuit Design: Smart Data Converters, Filters on Chip, Multimode Transmitters”, chapter 5: “*DAC Smart and Flexible DA Converters, Classification and Design*”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, Springer, November 2009, ISBN-13: 9789048130825, ISBN: 9048130824

19. “Calibration of current-steering D/A Converters”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund, Analogue/Mixed-Signal Innovation Network “Digitally Assisted Analogue”, 22nd October 2009, Xilinx, CityWest Campus, Dublin, Ireland.

20. “Smart Front-Ends, from Vision to Design”, A.H.M. van Roermund, P.Baltus, A. van Bezooijen, J.A. Hegt, E. Lopelli, R. Mahmoudi, G. Radulov, M. Vidojkovic; IEICE transaction on Electronics, vol. e92-c, No.6, June 2009,

LIST OF PATENTS

1. US Patent 7,076,384, Xilinx, “Method and apparatus for calibrating a current-based circuit”; G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund;

2. US Patent 7,466,252, STW, Xilinx, “Method and Apparatus for Calibrating a Scaled Current Electronic Circuit”, G.I. Radulov, P.J. Quinn, J.A. Hegt, A.H.M. van Roermund;

AWARDS

1. Outstanding student paper award, 2008 IEEE Asia Pacific Conference on Circuits and Systems – APCCAS 2008 for paper “A flexible 12 bit self-calibrated quad-core current-steering DAC”;
2. Master of Science degree with honors, Technical University of Sofia, 2001.

ACKNOWLEDGEMENTS

I, the author, would like to acknowledge and express my appreciation to all people who helped me during the work on this thesis.

First of all, I would like to express my sincere gratitude to my promoter prof.dr.ir. A.H.M. van Roermund and to my co-promoters dr. ir. H. Hegt, dr. ir. P.J.Quinn. The interactions with them have always been the main tool shaping my scientific personality.

Arthur, who I admire deeply, was most definitely the person without whom this book would not have been what it is. At the beginning of the project he, together with Patrick, Hans and John Jennings (from Xilinx, Ireland) defined the scope of the project. Throughout the PhD journey he has been a coach and co-author of all my publications. He was always very helpful with his systematic thinking and ability to understanding alternative and controversial ideas and concepts. Toward the end, and especially during the compilation of the thesis, he had invaluable comments regarding the structure and general flow of the argument. For the opportunity to work on this thesis, for the mentoring and the inspiration that you have been to me – thank you!

I am also very grateful to my daily supervisor from TU/e – dr. ir. Hans Hegt. Not only did he read all my written work and provided excellent feedback on it, but he has been someone who always provided me with moral support during the whole PhD journey. During the whole periode of my PhD studies, he was steering my research, he was commenting on all my ideas, he was checking all my formulas and written text. Thank you, Hans!

To my second co-supervisor – Patrick Quinn (form Xilinx, Ireland) – I owe the English language tutoring, and the English proficiency of all my written work. I owe the technical, transistor-level tutoring and discussions. I owe all the technical possibilities and chances that were given to me. Furthermore, he has been invaluable coach and has always helped me with the organization and the planning of the presented work. What is more, throughout this process he has become a great friend, who I appreciate highly.

I am particularly indebted to Arthur, Hans and Patrick for the time spent on reading and suggesting corrections during the final stages of drafting and editing this book.

Next to my direct supervisors there have been many people whose role on various parts of this project I would like to illuminate and underline. prof. dr. R.W. van der Hofstad, Markus, and Olaf – thank you for your help in mathematics, particularly for the chapter “Brownian Bridge based analysis and modeling of DAC linearity, an in-depth view”. The work with them was like a dream that came true, because of my great love for mathematics[1]. It has always been a pleasure to have discussions with you, and to work on common publications.

I am grateful to my fellow PhD colleague Pieter Harpe for our discussions of the Method for suppression of harmonic distortion, and the overall co-research on data converters topics, and beyond. I value highly all conversations and discussions that we had and the support you have always granted to me.

I furthermore thank dr. Kostas Doris for the initial guidance in the research on calibration and D/A converters.

[1] A love that I had to give up for an even greater passion: the passion for engineering.

A PhD project, in our field, is always embedded in several organizations, and one always becomes part of and gets to know various collectivities. In my case, I am deeply thankful to the MsM group for providing such a friendly, supportive and gezellig environment for a PhD to develop his project. The MsM group has become an essential part in my life and what I am now.

Moreover, I am immensely grateful for having the chance to have my project coupled with Xilinx, Ireland. This company provided great technical support, manufactured the test chips described in this book, made my two US patents possible, and provided me with the opportunity to access the most advanced technical and scientific expertise and excellence in the world.

I am also grateful to NXP (ex-Philips research), in particular the group of Leo Warmedam for the general technical discussions we had.

Last but not least, I am very thankful to STW for the financial support which provided the opportunity to accomplish this research.

I also have to acknowledge the students that were supervised by me - Richard worked on correction methods through mapping sub-DACs input codes; Matthijs worked on characterization and measurements of DAC test chips.

No professional achievement is possible without a strong support from family and friends. I firstly, want to thank my parents – Eleonora and Ivan – who raised me, and made me the man I am today – a strong believer in education, science and hard work. They are constantly helping my family, so that “I could concentrate in making science”. I am also grateful to them for not allowing me to deviate away in my youth from the path of engineering and science.

My sister Anna and her new family, emphasizing Viktoria and Ivan, I am grateful for the care and for always providing the precious moments in life for my family. These are the moments spent with the next generation.

Finally, I want to thank my wife Eli for letting me know what love is. The family that we’ve created and our son Ivan are the brightest light, the farthest vision, and greatest love in my life. To you, my dearest, who warm my heart, and who assured my happiness and emotional wellbeing throughout the work on this project, I dedicate this book.

Georgi Radulov was born in Plovdiv, Bulgaria in 1978. He received the M.Sc. engineer degree in electrical engineering in 2001 from the Technical University of Sofia (TU-Sofia), Bulgaria. In 2004, he received the degree Professional Doctorate in Engineering (PDEng) from Stan Ackermans Institute at Eindhoven University of Technology (TU/e).

From 1999 until 2001, he was a student assistant at ECAD Lab of TU-Sofia. Since August 2001, he is member of the Mixed-Signal Microelectronics (MsM) Group, at TU/e. Since 2009, he is an Assistant Professor at the Electrical Engineering faculty of TU/e.

Georgi Radulov holds 2 US patents on current calibration. In 2008, he is awarded the Outstanding Student Paper of the IEEE conference APCCAS 2008, in Macau. Georgi Radulov has more than 20 publications on Digital-to-Analog Converters.