

Chapter 2

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 book, 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 book.

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 Sect. 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 Sect. 2.7 important DAC design challenges. Finally, a summary of the chapter is provided in Sect. 2.8.

2.2 Functionality and Specifications

Digital-to-Analog Converters (DACs) implement a Digital-to-Analog (D/A) conversion function, see Fig. 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.

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 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 Fig. 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.

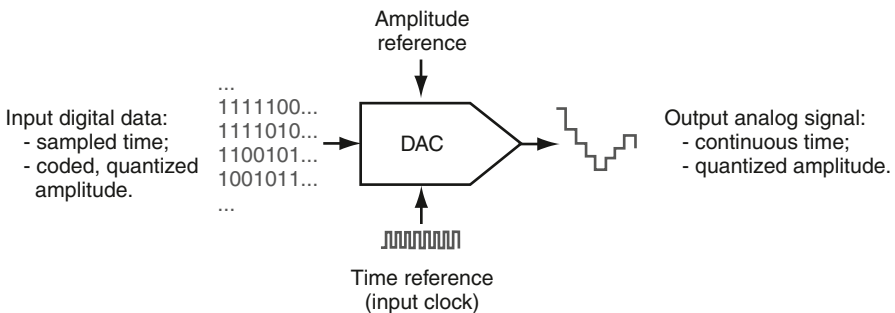


Fig. 2.1 The DAC as black box: input-to-output transformation

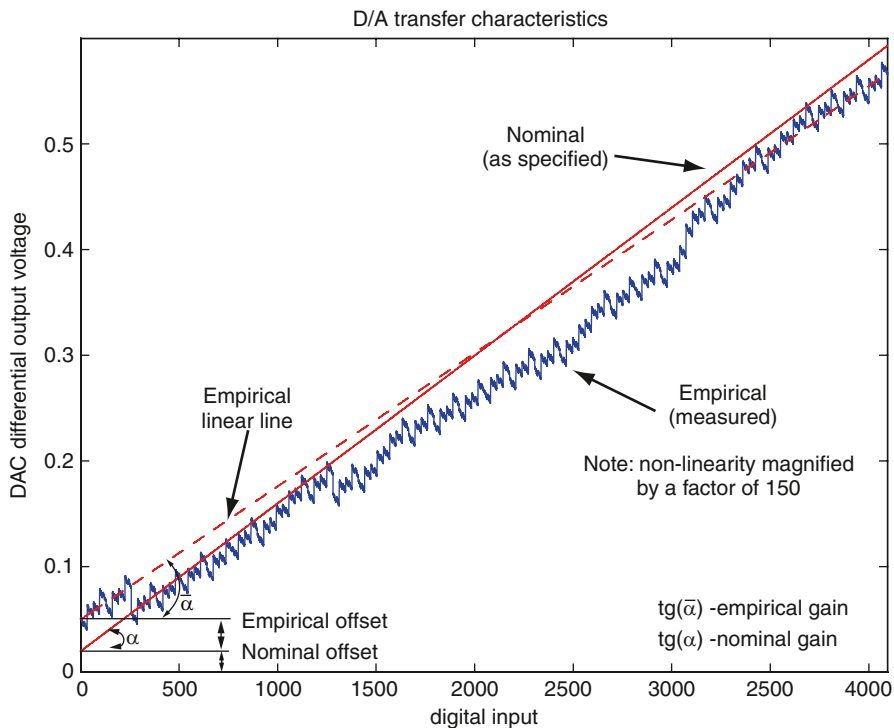


Fig. 2.2 Three D/A functions for a DAC, as designed (*straight line*), empirical ideally linear (*dashed line*), and empirical (*non-linear graph*, derived from measurements with non-linearity magnified by a factor of 150)

The straight line in Fig. 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 Fig. 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 book the line connecting the initial and final points of the actual D/A transfer characteristic is used (dashed line in Fig. 2.2). Based on this line, the empirical, e.g. measured, DAC offset, gain and FS can be defined, as shown in Fig. 2.2. Note that there are other ways to define the empirical ideally linear line, e.g. “best-fit” line, see [4].

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 Fig. 2.3, is defined. The evaluation of the INL usually includes two main properties: the global shape of

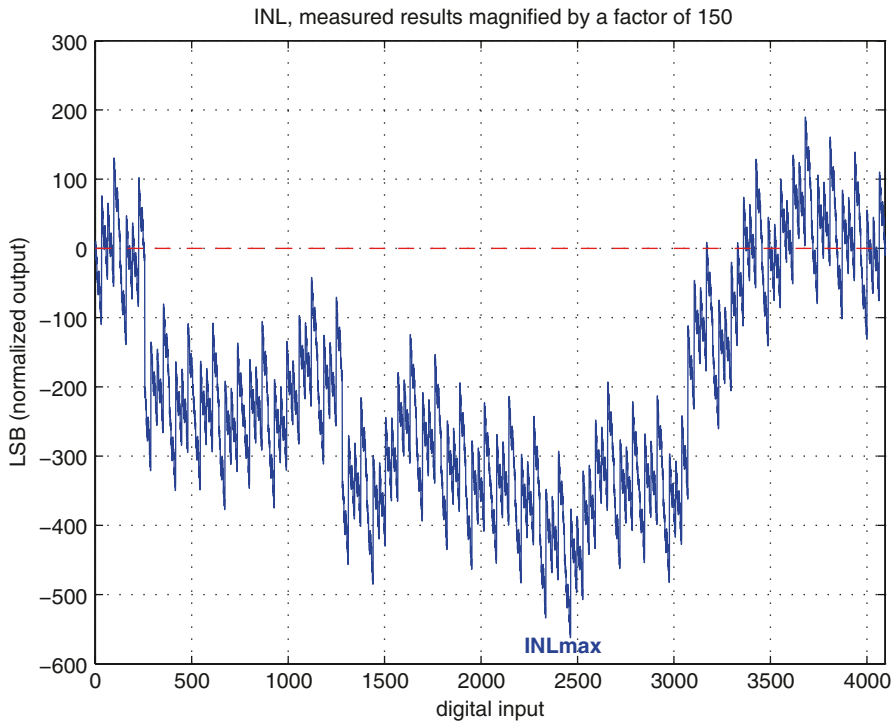


Fig. 2.3 12 bit DAC INL

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 Fig. 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 Fig. 2.3 is about -500 LSB, which suggests a linearity that is 10 bit less than the resolution, i.e. 2 bit DAC linearity. For accurate analytical definitions of INL, refer to Chap. 6.

For many DAC applications, e.g. control and self-calibration as shown further in the book, 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 :

$$DNL_k = INL_{k+1} - INL_k \quad (2.1)$$

Figure 2.4 shows the corresponding DNL characteristic of the D/A transfer characteristic of Fig. 2.2 and the INL characteristic of Fig. 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 .

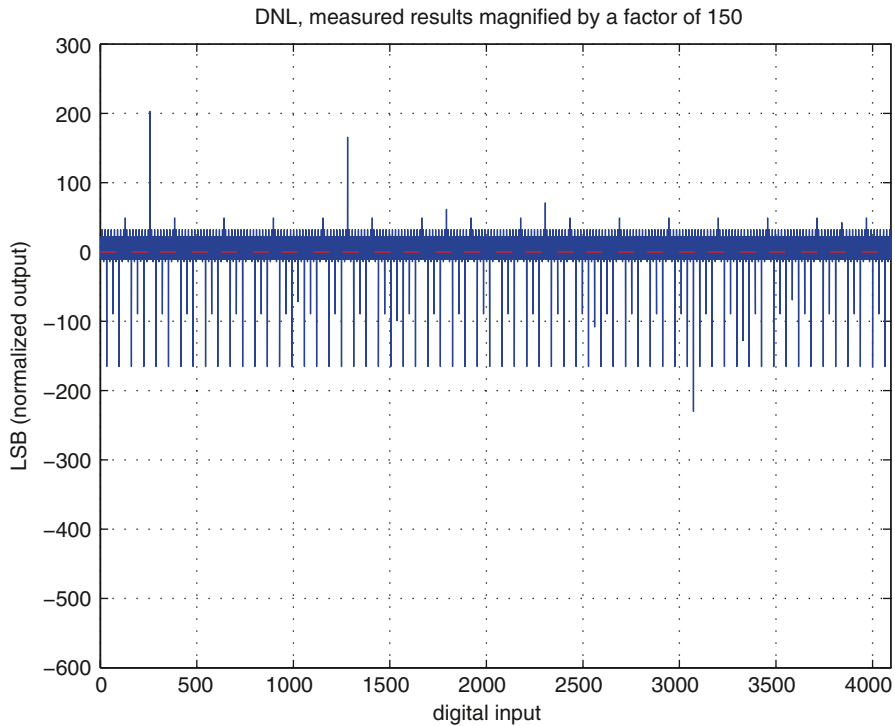


Fig. 2.4 12 bit DAC DNL

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.

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 book mainly considers the DAC dynamic linearity group of figures, since they are very important in the RF communication applications. Figure 2.5 shows an exemplary spectrum of a DAC sine wave output signal. DAC figures that are important for this book are indicated in the spectrum. SFDR (Spurious-Free-Dynamic Range), HD (Harmonic Distortion), and IMD (Inter-Modulation Distortion) are the

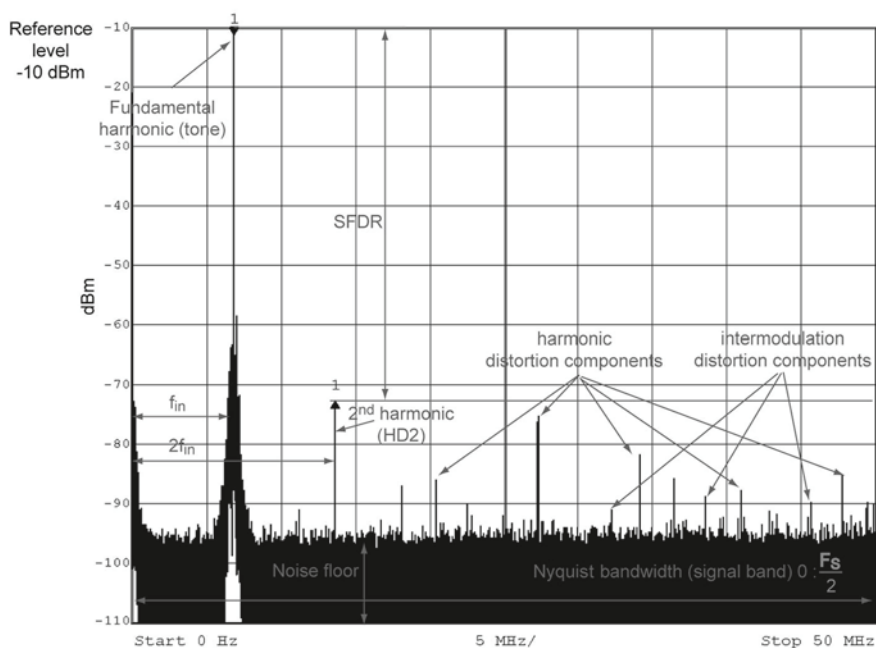


Fig. 2.5 A DAC output spectrum, converting a sine wave signal

most important figures that are further used in the book. In many places of the book, 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).

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 2.6 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*.

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

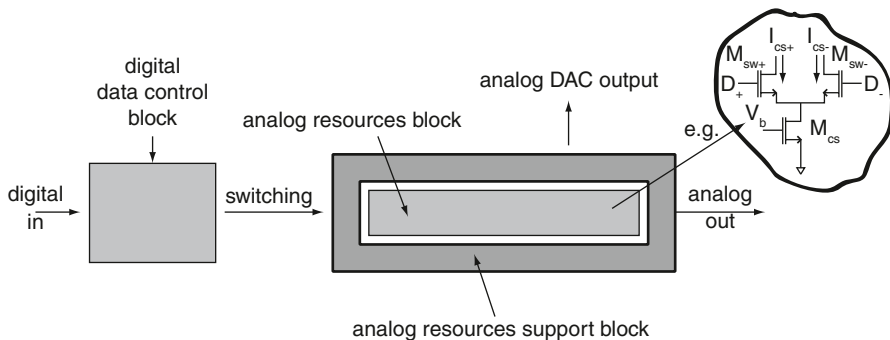


Fig. 2.6 A generalized block diagram of a DAC, with an example of current-steering implementation

(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.

$$\begin{aligned}
 I_{FS} &= \sum_i \frac{1}{2} k \left(\frac{W}{L} \right)_i (V_{gs} - V_{th})^2 (1 + \lambda V_{ds}) \\
 &= \frac{1}{2} k (V_{gs} - V_{th})^2 (1 + \lambda V_{ds}) \sum_i \left(\frac{W}{L} \right)_i,
 \end{aligned} \tag{2.2}$$

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:

$$\left(\frac{\sigma_I}{I} \right)^2 = \left(A_\beta^2 + \frac{4A_{VT}^2}{(V_{gs} - V_{th})^2} \right) \frac{1}{2(WL)}, \tag{2.3}$$

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 Chap. 6.

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

$$r_o = \frac{1}{\lambda I_d} = \frac{1}{\Delta L} \frac{V_{ds}}{I_d} L, \quad (2.4)$$

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 C_j , and the sidewall capacitance due to the perimeter of the junction C_{jsw} , 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].

Equations 2.2, 2.3, and 2.4 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 Chap. 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 Chap. 9. Thus, Sect. 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. 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 2.7 shows a block diagram of the binary segmented DAC.

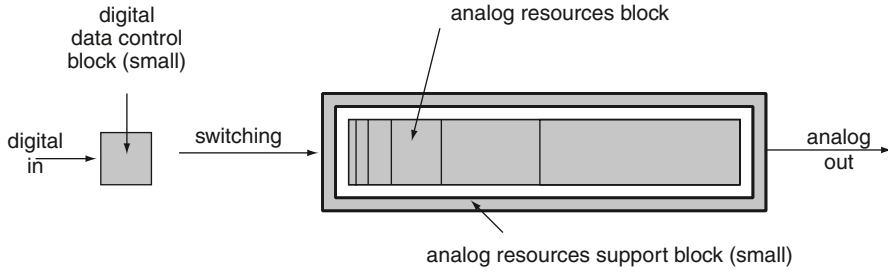


Fig. 2.7 A generalized block diagram of a binary DAC

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.:

$$DNL_{\max} = \sigma(\Delta I) = \sqrt{2^N - 1} \left(\frac{\sigma_u}{I_u} \right), \quad (2.5)$$

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. Figure 2.8 shows a generalized block diagram of a sub-binary radix DAC. 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 linear-

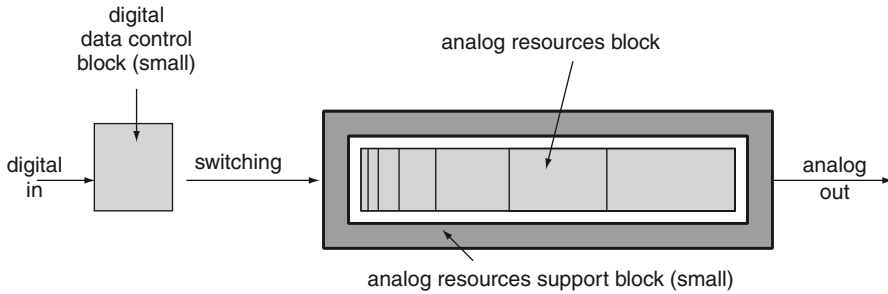


Fig. 2.8 A generalized block diagram of a sub-binary radix DAC.

ity to the target level. The negative DNL errors are removed by a pre-processing DAC correction method, as explained in Sect. 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].

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 Fig. 2.9. 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

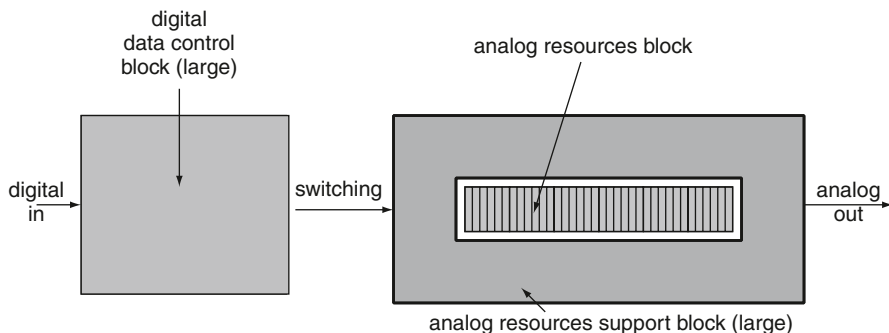


Fig. 2.9 A generalized block diagram of a unary DAC

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 (Sect. 4.3.1) and DEM (Sect. 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.

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 Fig. 2.10. 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.

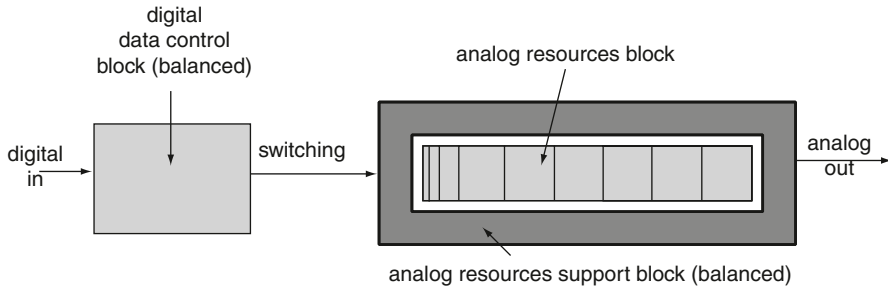


Fig. 2.10 A generalized block diagram of a segmented binary-unary DAC

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 Sect. 2.2 and algorithmic segmentation in Sect. 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 2.11 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 Fig. 2.12a. 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 Fig. 2.12b. 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, Fig. 2.12b, 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.

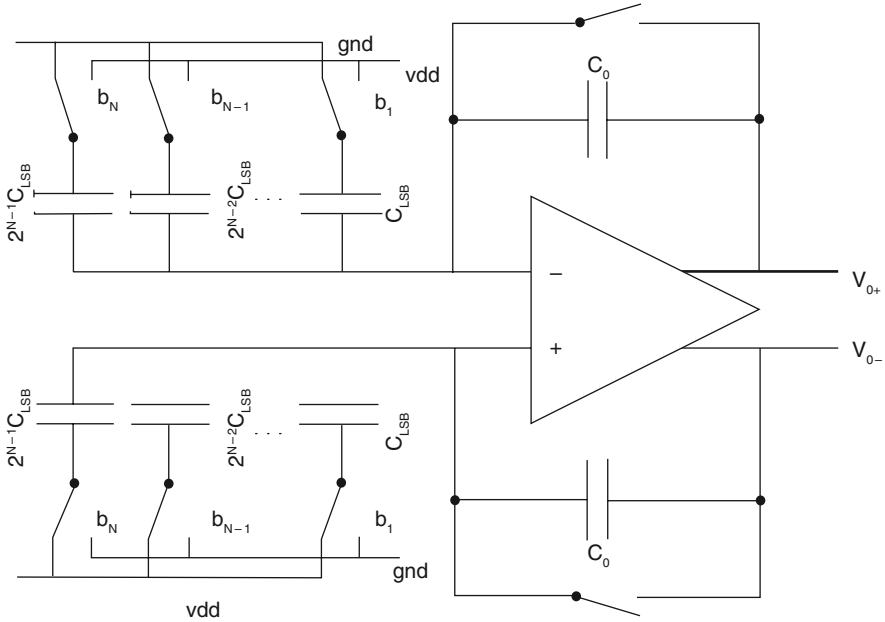


Fig. 2.11 Exemplary N bit charge redistribution DAC, without reset phase

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

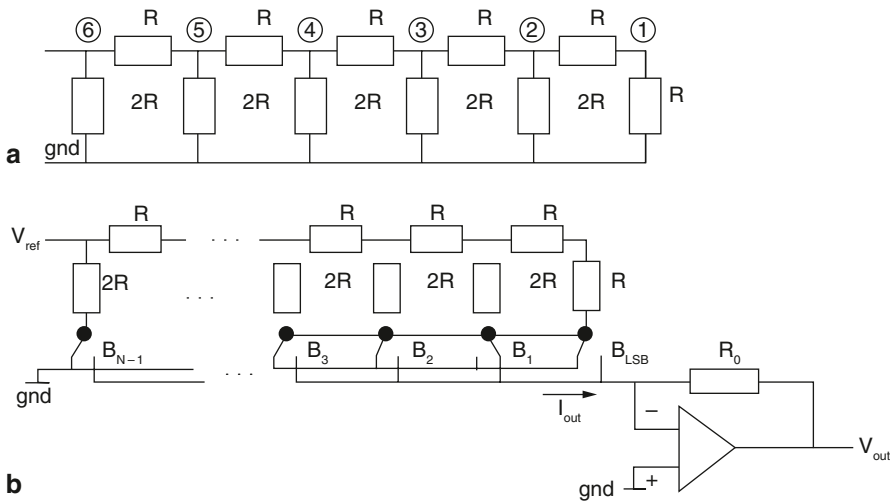


Fig. 2.12 **a** R2R ladder. **b** R2R ladder based DAC

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 2.13 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 $2^{(N-B)} - 1$

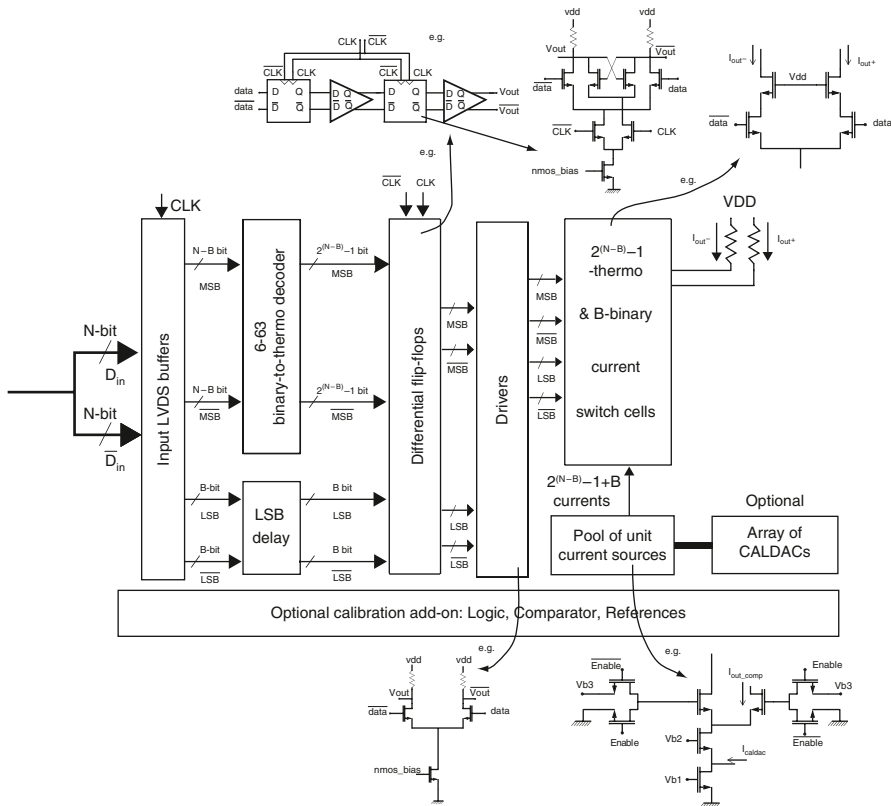


Fig. 2.13 A block diagram of a Current-Steering DAC, with binary LSB—unary MSB algorithmic segmentation, exemplary circuit implementations, and an optional exemplary calibration engine

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.

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

Table 2.1 Development of the Digital-to-Analog converters

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

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 2.1 shows an example of how the performance of some selected state-of-the-art CMOS DACs evolves in the years.

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 five times scaled-down CMOS process, the main DAC improvement is the speed (six times), while the improvements in the resolution (two more bits, i.e. four times increase of the convertible codes) and the occupied area (three 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 2.1. 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, Eq. 2.3 shows the relationship between the transistor area $(WL)_{\min}$ and the relative matching σ_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 Chap. 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} :

$$\mathbb{E}(INL_{\max}^{(Thermo)}) = \left(\frac{\sigma_I \sqrt{2^N - 1}}{\bar{I}} \right) \cdot \frac{1}{2} \sqrt{2\pi} \ln 2 \approx 0.869 \cdot \left(\frac{\sigma_I \sqrt{2^N - 1}}{\bar{I}} \right) \quad (2.6)$$

From Eqs. 2.3 and 2.6, 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 contradictory 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 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 Sect. 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 book 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.

Smart and Flexible Digital-to-Analog Converters

Radulov, G.; Quinn, P.; Hegt, H.; van Roermund, A.H.M.

2011, XIV, 310 p., Hardcover

ISBN: 978-94-007-0346-9