INTEGRATION, the VLSI journal 53 (2016) 88–99

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

An efficient tool for the assisted design of SAR ADCs capacitive DACs

S. Brenna [a,*], A. Bonetti [b], A. Bonfanti [a], A.L. Lacaita [a]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy*
[b] *Telecommunications Circuits Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland*

ARTICLE INFO

Article history:
Received 21 April 2015
Received in revised form
20 October 2015
Accepted 15 December 2015
Available online 25 December 2015

Keywords:
Data-converters
Nonlinearity
Assisted design
Charge redistribution
SAR ADC

ABSTRACT

The optimal design of SAR ADCs requires the accurate estimate of nonlinearity and parasitic capacitance effects in the feedback charge redistribution DAC. Since both contributions depend on the specific array topology, complex calculations, custom modeling and heavy simulations in common circuit design environments are often required. This paper presents a MATLAB-based numerical environment to assist the design of the charge redistribution DACs adopted in SAR ADCs. The tool performs both parametric and statistical simulations taking into account capacitive mismatch and parasitic capacitances computing both differential and integral nonlinearity (DNL, INL). An excellent agreement is obtained with the results of circuit simulators (e.g. Cadence Spectre) featuring up to 10^4 shorter simulation time, allowing statistical simulations that would be otherwise impracticable. The switching energy and SNDR degradation due to static nonlinear effects are also estimated. Simulations and measurements on three designed and two fabricated prototypes confirm that the proposed tool can be used as a valid instrument to assist the design of a charge redistribution SAR ADC and to predict its static and dynamic metrics.

1. Introduction

Efficient analog-to-digital converters (ADCs) are essential building blocks of low-power applications, such as wireless sensor nodes, portable biomedical instruments, health monitoring systems, and a wide variety of consumer electronic products that integrate an increasing quantity of sensors. In terms of efficiency, for moderate speeds and resolutions that are typically required by the most of the aforementioned applications, charge redistribution successive approximation register (CR-SAR) converters are the best choice and dominate the ADC market. In the last decades, starting from the Classic Binary Weighted (CBW) SAR ADC [14], other solutions have been proposed to improve the efficiency [16,10] and adopted in various systems [12,9].

Both static and dynamic performance figures of such converters strongly depend on the nonlinearities determined by mismatch and parasitics affecting the capacitive array of the feedback digital-to-analog converter (DAC, see Fig. 1). Impact of mismatch on Differential-Non-Linearity (DNL) and Integral-Non-Linearity (INL) of the most generally adopted array topologies have been studied and formulae are available in the literature [15]. However, no quantitative guideline is available to address nonlinearities arising from parasitic capacitances. This effect is deterministic and strongly depends on the array architecture and on the layout quality. Therefore, the parasitics impact on the converter nonlinearities is addressed and minimized relying on transient simulations performed in Electronic Design Automation (EDA) tool, such as Cadence. Unfortunately, such a procedure is extremely time-consuming and requires heavy data post-processing to estimate the static nonlinearity metrics as well as the Signal-to-Noise and Distortion Ratio (SNDR) and the Equivalent Number of Bits (ENoB). Similar issues arise also in many ultra-low-power designs where sub-10fF unit capacitors are adopted [16,10,11]. In this case, the impact of the DAC parasitics on the converter power consumption becomes not negligible and in a traditional EDA tool environment its estimate always relies on transient analyses, thus being time-consuming. To overcome these limitations, this work proposes a MATLAB-based tool (CSAtool) able to speed-up the simulations needed to estimate the ADC static nonlinearities introduced by the DAC non-idealities, their impact on the converter dynamic performance and also on its power consumption. To the best of authors knowledge, this is the first tool proposed in the literature as a valid instrument to assist the design and the analysis of the SAR ADC capacitive array which deals with both different array topologies and switching algorithms. The tool supports three of the most known and employed SAR converter topologies, namely the Classic Binary Weighted (CBW) [14], the Split Binary Weighted (SBW) [6] and the Binary Weighted with

[*] Corresponding author.
E-mail addresses: stefano.brenna@polimi.it (S. Brenna), andrea.bonetti@epfl.ch (A. Bonetti).

http://dx.doi.org/10.1016/j.vlsi.2015.12.005

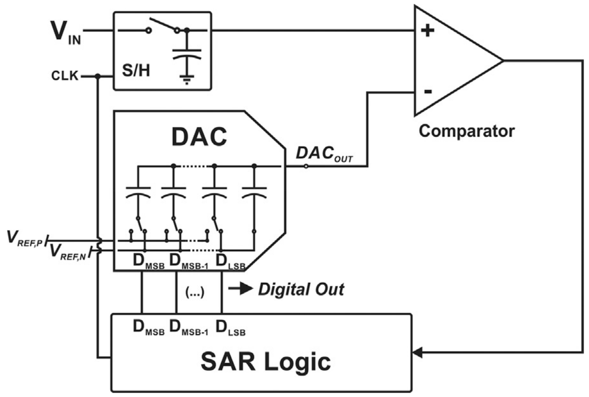


Fig. 1. Generic SAR ADC architecture with a capacitive DAC in the feedback path.

Attenuation Capacitor (BWA) [2] array, both single-ended and fully differential, using either the conventional switching algorithm [14] or the monotonic scheme [13].

The tool reproduces in a MATLAB environment the behavior of the capacitive array, eventually taking into account the effect of mismatch and of the parasitics, and computes the static input–output characteristic, not solving ordinary differential equations, like a Spice-like simulator, but performing arithmetic operations among vectors. In this way, CSAtool allows us to easily estimate:

- the impact of the mismatch and parasitics on the static nonlinearity (DNL and INL) with both single and statistical simulations;
- the impact of the mismatch and the parasitics on the dynamic nonlinearity (SNDR and ENoB);
- the DAC switching energy, including the parasitics contribution, as a function of the output code.

The modeling approach and how the converter linearity performance are estimated have already been presented in [4]. Aim of this work is to explain in a more detailed way the working principle of the proposed tool and to compare its results with the ones achieved by a traditional EDA tool, like Cadence, in terms of speed and accuracy. The switching energy modeling and the related simulation results are also presented and compared to those obtained with Cadence. Moreover, measurement results on two fabricated ADCs out of the three designed are reported to appreciate the validity of the tool.

The paper is organized as follows. Section 2 describes the effects of mismatch and parasitic capacitances on the nonlinearity metrics (DNL and INL) in the implemented converter topologies. Section 3 sketches the tool algorithm based on the evaluation of the A-to-D input–output characteristics by means of simple static operations on vectors. The models of the different converter architectures are described in detail in Section 4, while Section 5 describes the algorithm adopted to compute the switching energy. Section 6 shows the typical design flow of a SAR converter, highlighting the advantages of adopting CSAtool with respect to the traditional EDA tool-based approach. Section 7 compares CSAtool estimates with the Cadence Spectre simulation and measurement results on three designed and two fabricated ADCs. Finally, conclusions are drawn in Section 8.

2. Converter topologies

The capacitive network adopted in SAR ADCs can be described as a composition of one or more binary weighted arrays connected to the output node either in parallel or through an attenuation capacitance. During the conversion cycle, the switches configuration (see Fig. 1) changes to generate the corresponding output voltage. This voltage marks an input transition level between two adjacent digital codes. Therefore, the mismatch and the parasitics of each capacitance affect the conversion accuracy. In the following, the topologies of the converters adopted for the tool validation are briefly described focusing on the impact that both capacitive mismatch and parasitics have on the converters performance.

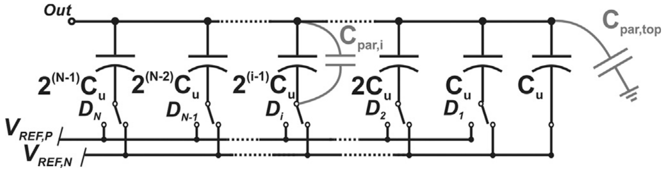


Fig. 2. Schematic of a N-bit CBW array.

2.1. Classic binary weighted array (CBW)

Fig. 2 shows a simple N-bit CBW array where each capacitive block is oriented with the bottom-plate towards the input voltage reference lines to minimize the parasitics impact. From a formal standpoint, the capacitance of each capacitive block is the binary sum of unit capacitors C_u (i.e. $C_i = 2^{i-1}C_u$) plus the contribution $C_{par,i}$ due to the stray capacitances between the top- and the bottom-plate nodes. The parasitic capacitances between the top plates and a reference voltage contribute to $C_{par,top}$ (see Fig. 2), which attenuates the DAC output independently of the code, then causing a gain error without degrading the converter linearity [15]. Also the stray capacitances between the bottom-plates and a reference voltage do not contribute to conversion errors since they are directly driven by the SAR logic drivers.

Parasitic capacitances are deterministic, depending on wires and array geometry of the converter layout. Thus, once the array is designed, their impact on DNL and INL performance can be assessed computing the converter characteristic through circuit simulations. On the contrary, the capacitor mismatch causes a statistical error. Indeed, analytic expressions are available to estimate the maximum standard deviation of DNL and INL [15,6]. In fact, the capacitive mismatch can be modeled assuming a Gaussian probability distribution of the unit capacitor value with a mean equal to the nominal capacitance, C_u, and a standard deviation of

$$\sigma_C = \frac{k_c C_u}{2\,A} = k_c \cdot \sqrt{\frac{C_{spec} \cdot C_u}{2}}. \tag{1}$$

k_c, A and c_{spec} being the Pelgrom mismatch coefficient, the area and the specific capacitance, respectively. Under this assumption and considering a single-ended CBW array, the maximum DNL standard deviation occurs at the mid-code and is given by [15]

$$\sigma_{DNL,CBW} = 2^{N/2} \cdot \frac{\sigma_C}{C_u}, \tag{2}$$

The corresponding maximum standard deviation value for the INL is

$$\sigma_{INL,CBW} = 2^{(N/2)-1} \cdot \frac{\sigma_C}{C_u}. \tag{3}$$

In a fully-differential configuration, these results have to be divided by a factor of $\sqrt{2}$ [17]. In design practice, the value of the unit capacitor C_u is set to bring the matching-limited DNL and INL values below the requirements and then the linearity degradation due to parasitic capacitances is assessed by circuit simulations.

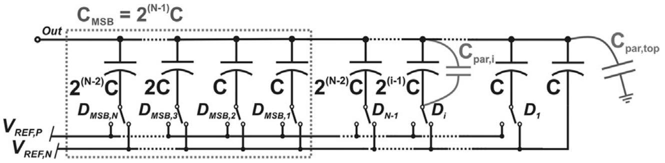


Fig. 3. Schematic of a *N*-bit SBW array.

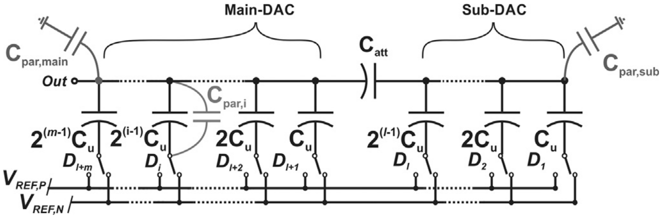


Fig. 4. Schematic of a *N*-bit BWA array.

2.2. *Split binary weighted array (SBW)*

The split DAC topology [7] is shown in Fig. 3. It consists of a binary weighted array where the MSB capacitor is implemented by a binary weighted sub-array that perfectly mirrors the structure of the remaining capacitive banks. This DAC topology features an improved switching efficiency and also a reduced impact of the capacitors mismatch. In fact, the maximum standard deviation of the DNL and INL, which still occurs at the mid-code, is a factor of $\sqrt{2}$ lower than in the CBW array topology [7], being

$$\sigma_{DNL,SBW} = 2^{(N-1)/2} \cdot \frac{\sigma_C}{C_u}, \tag{4}$$

$$\sigma_{INL,SBW} = 2^{(N-2)/2} \cdot \frac{\sigma_C}{C_u}. \tag{5}$$

These relations are referred to a single-ended configuration while a fully-differential topology is a further $\sqrt{2}$ factor less sensitive to mismatch. Moreover, as in the CBW array, only the parasitics $C_{par,i}$ connected between top- and bottom-plate nodes of each array capacitor limit the converter linearity.

2.3. *Binary weighted with attenuation capacitor (BWA)*

In a single-ended BWA array, the capacitive network is divided into two binary weighted arrays separated by an attenuation capacitor, C_{att} (see Fig. 4) [1]. In this work, we will consider the case where both DACs have the same number of bits (i.e. $m = l = N/2$) and $C_{att}{=}C_u$. In fact, this topology leads to the most energy efficient solution [15]. It has been shown that the BWA topology is more sensitive than CBW topology to capacitor mismatch when the same unit capacitance is employed. Closed formulae similar to (2) and (3) are presented in [15] for the single-ended BWA topology. The maximum σ_{DNL} and σ_{INL} set by mismatch are

$$\sigma_{DNL,BWA} = 2^{3N/4} \cdot \frac{\sigma_C}{C_u}, \tag{6}$$

$$\sigma_{INL,BWA} = 2^{3N/4-1} \cdot \frac{\sigma_C}{C_u}. \tag{7}$$

These standard deviations are a factor of $2^{N/4}$ larger than in CBW array. Regarding the impact of the parasitics, in addition to the top-to-bottom plate capacitances $C_{par,i}$, also the stray capacitance connected to the top-plate node of the sub-DAC ($C_{par,sub}$ in Fig. 4) affects the linearity since it makes the DAC output voltage depending on the input code. Instead, the parasitic connected to the top-plate of the main-DAC, $C_{par,main}$, only affects the converter gain [15].

3. Tool working principle

The proposed MATLAB-based tool computes the input–output characteristic of the aforementioned charge redistribution SAR ADCs topologies. The tool does not implement the known equations that estimate the converter nonlinearity (i.e. the maximum standard deviation of DNL and INL) but reproduces the fundamental circuital behavior of each specific array topology, handling both array parasitics and capacitive mismatch. Differently from Spice-like simulators, which solve ordinary differential equations, CSAtool performs arithmetic operations (sums and products) among vectors of capacitances and digital words. In fact, the converter characteristic can be easily derived once all the transition levels of the input voltage are known. These levels can be assessed through a voltage capacitive divider, thus by means of a simple capacitive ratio, eventually taking into consideration the parasitics and the statistical variation of the array capacitances. This approach allows us to drastically lighten the computation of the nonlinearity metrics still achieving an excellent accuracy.

Fig. 5 shows the block diagram of the proposed tool. Once the number of bits, the switching algorithm and the converter topology are fixed, the tool performs the following steps:

1. implementation of the DAC capacitance model (capacitance vector, $\overline{C}$), eventually adding the parasitics and/or mismatch contribution;
2. evaluation of DAC output vector ($\overline{DAC}_{out}$), i.e the DAC output voltages corresponding to all the possible switch configurations, which represents all the transition levels between adjacent codes;
3. assessment of the ADC input-to-output characteristic from the DAC output vector;
4. evaluation of the static metrics (DNL and INL);
5. evaluation of the dynamic metrics (SNDR and ENoB).

The crucial step of the tool algorithm is the implementation of the capacitance vector, $\overline{C}$. From a general standpoint, in any charge redistribution SAR converter, each capacitance of the array can be written as the sum of different contributions

$$C_i = 2^{i-1} C_u + \sum_{j=1}^{2^{i-1}} \delta_j + C_{par,i}, \quad i = 1, \ldots N, \tag{8}$$

where the first term is its nominal value (expressed as the sum of unit elements) and the term δ_j represents the mismatch contribution affecting each of the unit capacitors of C_i. Thus, the effect of mismatch is taken into account considering that the capacitance value of each unit element of the array follows a Gaussian probability function, with a mean equal to its nominal value, C_u, and a standard deviation σ_C as in (1). When statistical simulations are performed, the parameters δ_j for each unit capacitance of all the capacitive banks are varied according to the probability function, like in a Monte Carlo simulation.

The term $C_{par,i}$ in (8) is the parasitic capacitance of the *i*th-capacitive block, obtained by adding the stray capacitances between the top- and the bottom-plate nodes of each unit element of C_i.

Once the vector $\overline{C}$ is known, the next step is to compute the DAC output vector, $\overline{DAC}_{out}$, whose elements are all the voltage transition levels between adjacent codes. In fact, the analog-to-digital conversion is performed by comparing the input signal with subsequent voltage levels generated by the capacitive DAC through a binary search algorithm, as shown in Fig. 6. The $\overline{DAC}_{out}$

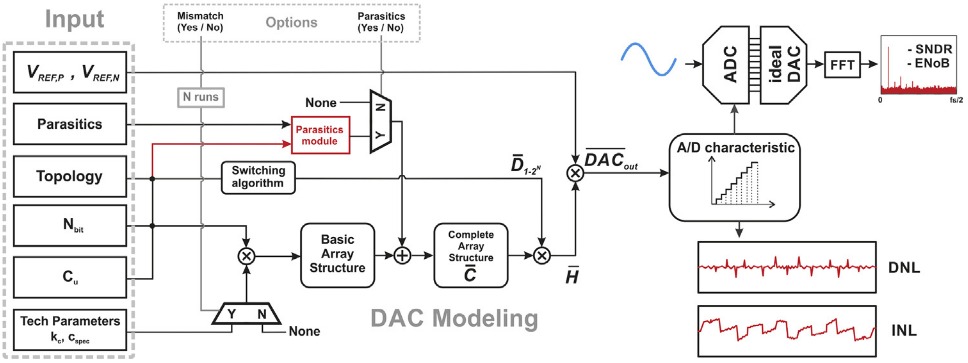


Fig. 5. CSAtool block diagram.

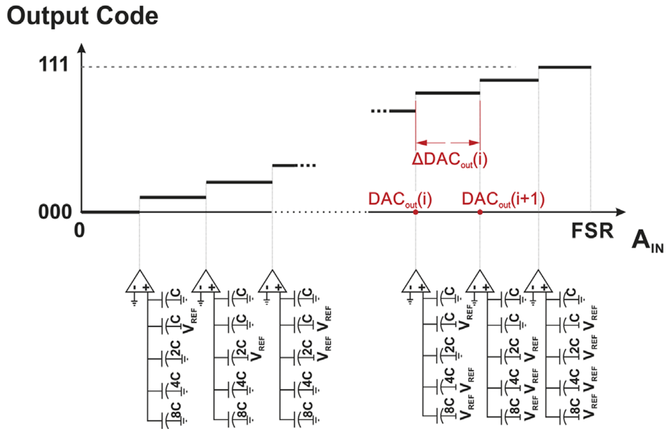


Fig. 6. Conversion characteristic for a 3-bit single-ended AD converter. The analog input transition levels are set by the DAC output.

vector allows us to easily compute the ADC input-to-output characteristic (see Fig. 5). The DNL as a function of the output code is then evaluated by computing the vector $\overline{\Delta DAC}_{out}$ of the differences $\Delta DAC_{out}(i)$ between all the adjacent elements of the DAC output vector

$$\Delta DAC_{out}(i) = DAC_{out}(i+1) - DAC_{out}(i) \tag{9}$$

as

$$DNL(i) = \frac{\Delta DAC_{out}(i) - \mu\left(\overline{\Delta DAC}_{out}\right)}{\mu\left(\overline{\Delta DAC}_{out}\right)}, \quad i = 0, .., 2^N - 1 \tag{10}$$

where $\mu\left(\overline{\Delta DAC}_{out}\right)$ is the average of the $\Delta DAC_{out}(i)$ values. Finally, the INL curve is obtained from the integration of the estimated DNL.

Regarding the dynamic metrics (SNDR and ENoB), the knowledge of the input–output characteristic allows us to compute the response of the converter to an input sinewave. The test-bench is schematically depicted in Fig. 5. A sinewave with amplitude varying from 1 to 100% of the full-scale range and with an arbitrary frequency is converted into a digital format on the basis of the input–output characteristic. The digital words are then converted in decimal format and the spectrum is computed by applying the Fast Fourier Transform (FFT) in order to derive the dynamic metrics, being the ENoB a function of the peak SNDR. This procedure can be repeated considering different values of the mismatch contribution, randomly chosen in accordance to the Gaussian probability function, allowing us to estimate the statistical properties of the considered converter. Indeed, also thermal noise, comparator nonlinearity and aperture time jitter of the sampling clock can limit the dynamic performance of an AD converter. These issues could be taken into account only by complex and time-consuming simulations in EDA tool environments. On the contrary, CSAtool emulates the conversion on the basis of the static input–output characteristic. This, obviously, does not get the impact of all the possible dynamic contributions to the accuracy degradation, but still gives the possibility to estimate the SNDR and ENoB limit imposed by the mismatch and the parasitics of the DAC, which are often a significant error source [8].

4. Capacitive array model

This section is devoted to explain in detail the model implementation and the evaluation of the DAC output vector for the CBW, SBW and BWA converters. For the simplest model, the CBW topology, also some lines of the MATLAB code are reported to illustrate how the computations are performed in the CSAtool.

4.1. CBW model

In a conventional binary weighted topology, the DAC output voltage at each conversion step can be written as

$$DAC_{out} = FSR \cdot H, \tag{11}$$

where *FSR* is the full scale range of the converter and *H*, as shown in Fig. 5, is the scalar product

$$H = \frac{1}{C_{tot} + C_{par,top}} \cdot \overline{C} \times \overline{D}'. \tag{12}$$

In (12), C_{tot} is the total capacitance of the array, $C_{par,top}$ is the parasitic capacitance shown in Fig. 2, $\overline{C}$ is the vector of the array capacitances C_i and $\overline{D}$ is the vector of the digital word updated at each conversion cycle

$$\overline{C} = [C_1 \quad \dots \quad C_N], \tag{13}$$

$$\overline{D} = [D_1 \quad \dots \quad D_N]. \tag{14}$$

The digital word $\overline{D}$, which encodes the DAC output levels at each conversion step, is determined by the adopted switching algorithm. The DAC output vector ($\overline{DAC}_{out}$ in Fig. 5) can be built evaluating (11) and (12) for all the possible vectors $\overline{D}$ depending on the switching algorithm.

This behavioral model, which entails arithmetic operations on vectors, has been implemented in MATLAB. In the following, some lines of the MATLAB code are reported to clarify the working principle of the tool. The first step is to build the capacitive array, $\overline{C}$, whose elements are the binary capacitances of the DAC. Their values are influenced by the statistical mismatch affecting all the unit capacitances, and by the parasitics. The related code is here reported:

```
stdc = Cu*kc/sqrt(2*C/cspec);
for i=1:N
       for j=1 : 2^(i-1)
       CAR( i+1, j ) = Cu + normrnd(0,stdc);
       %temporary variable for the cap. array
       end
end
C=zeros(1,n);
C(1)=CAR(2,1);
for i=2:N
      for j=1:2^(i-1)
      C(i) = C(i) + CAR(i+1,j);
      end
end
clear CAR
C = C + Cpar;
%capacitance array vector
```

Once the array unit capacitance (*Cu* in the MATLAB code) has been chosen, the effect of mismatch can be taken into account considering the statistical property of the adopted capacitance, described by the technological parameters *kc* and *cspec*, i.e. the Pelgrom mismatch coefficient and the specific capacitance. If a statistical analysis is performed, the standard deviation of the unit capacitance is evaluated as in (1) and stored in the variable *stdc*. For each simulation and for each unit element of the binary capacitances, the actual value of *Cu* is sorted using the function *normrnd* that generates a random numbers chosen from the Gaussian distribution with zero mean and a standard deviation *stdc*. Eventually, also the parasitics affecting the array can be supplied through the vector *Cpar*.

Once the array is implemented, the input–output transition levels are evaluated through capacitive ratios, multiplying the capacitance vector with all the possible digital words set by the switching algorithm. In the case of a single-ended CBW converter and for the traditional switching algorithm [14,5], the following MATLAB code computes the transition levels, thus the input/output characteristic:

```
for i=1:(2^N - 1)
 D = fliplr(de2bi(i,n));
 H = C*D/(sum(C) + Cpar_top);
 DACout(i) = FSR*H;
end
levels=DACout;
%vector of input transition levels
```

In these lines of code, *D* is the vector encoding the digital word ($\overline{D}$ in (12)), while *Cpar_top* is the parasitic capacitance from the DAC output node to ground. The vector *levels* includes all the DAC output voltages corresponding to the input transitions. From this vector, the static nonlinearity metrics, DNL and INL, can be easily derived as follows:

```
DNL=diff(levels)/mean(diff(levels)) - 1;
INL=cumsum(DNL);
```

4.2. SBW model

The simple model described in the previous section can be extended to the SBW architecture of Fig. 3. The MSB capacitor is implemented as a sub-array and the switching scheme differs from the conventional algorithm [7]. Thus, the DAC output voltage can be expressed as

$$DAC_{out,SBW} = FSR \cdot (H_{MSB} + H_{1,MSB-1}), \tag{15}$$

H_{MSB} and $H_{1,MSB-1}$ being coefficients related to the MSB and the residual capacitance array, respectively,

$$H_{MSB} = \frac{1}{C_{tot} + C_{par,top}} \times \overline{C_{MSB}} \times \overline{D'_{MSB}} \tag{16}$$

$$H_{1,MSB-1} = \frac{1}{C_{tot} + C_{par,top}} \times \overline{C_{1,MSB-1}} \times \overline{D'_{1,MSB-1}}. \tag{17}$$

Thus, the conversion voltage level is set by two different *N*-bit words, $\overline{D_{MSB}}$ and $\overline{D_{1,MSB-1}}$, and two vectors of capacitances, $\overline{C_{MSB}}$ and $\overline{C_{1,MSB-1}}$, related to the MSB sub-array and to the residual array, respectively.

4.3. BWA model

In the BWA topology, two equal capacitive arrays must be considered: a main-DAC and a sub-DAC, which are related to the MSBs and the LSBs, respectively. Let us indicate as $C_{tot,main}$ and $C_{tot,sub}$ the overall capacitances of the main-DAC and of the sub-DAC, and as $C_{par,main}$ and $C_{par,sub}$ the parasitic capacitances at the top-plate node of the corresponding DAC (see Fig. 4). Due to the presence of the attenuation capacitor, C_{att}, the sub-DAC contribution to the overall DAC output voltage is reduced by an attenuation factor

$$AR = \frac{C_{att}}{C_{tot,main} + C_{par,main} + C_{att}}. \tag{18}$$

Thus, each DAC output in the BWA topology is evaluated as

$$DAC_{out} = FSR \cdot (H_{main} + AR \cdot H_{sub}), \tag{19}$$

where H_{main} and H_{sub} are coefficients related to the main and sub-DAC, respectively,

$$H_{main} = \frac{1}{C_{par,main} + C_{tot,main} + C_{att}} \cdot \overline{C_{main}} \times \overline{D'_{main}}, \tag{20}$$

$$H_{sub} = \frac{1}{C_{tot,sub} + C_{par,sub} + C_{att}} \cdot \overline{C_{sub}} \times \overline{D'_{sub}}. \tag{21}$$

In (20) and (21), $\overline{C_{main}}$, $\overline{C_{sub}}$, $\overline{D_{main}}$ and $\overline{D_{sub}}$ are the capacitance and digital output code vectors related to the main- and the sub-DAC, being

$$\overline{C_{main}} = [C_{N/2+1} \quad \ldots \quad C_N] \tag{22}$$

$$\overline{C_{sub}} = [C_1 \quad \ldots \quad C_{N/2}] \tag{23}$$

$$\overline{D_{main}} = [D_{N/2+1} \quad \ldots \quad D_N] \tag{24}$$

$$\overline{D_{sub}} = [D_1 \quad \ldots \quad D_{N/2}]. \tag{25}$$

5. Switching energy computation

The proposed tool also allows us to compute the DAC switching energy as a function of the output code for all the handled array topologies. To this aim, the same static approach adopted to compute the DAC output as a function of the digital code is employed. Also in this case, mismatch and parasitic contribution can be taken into account.

For the sake of generality, the approach adopted for the energy estimation is illustrated in the following referring to a 6-bit CBW topology. Fig. 7 shows a single-ended 6-bit CBW array for a

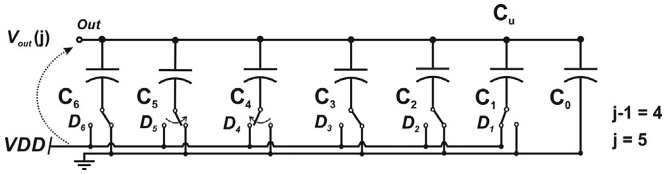


Fig. 7. 4th bit evaluation step of a 6-bit CBW converter. The capacitance C_4 is switched to V_{DD}.

particular configuration of the switches and considering power supply and ground as positive and negative reference voltage, respectively. Each configuration of the switches yields an output voltage, which corresponds to a transition level between two adjacent digital codes, and determines the charge or the discharge of the array capacitances. In particular, Fig. 7 shows the switch configuration when the 4th bit is evaluated. The energy spent by the power supply can be evaluated considering the charge variation of all the capacitances that are connected to V_{DD} at the end of the considered conversion step. For the generic jth-bit evaluation step, the energy absorbed from the power supply is

$$E(j) = [C_j(V_{DD} - V_{out}(j) + V_{out}(j+1))]V_{DD} + \sum_m [C_m(-V_{out}(j) + V_{out}(j+1))]V_{DD}, \tag{26}$$

where $V_{out}(j)$ and $V_{out}(j+1)$ are the output voltages corresponding to the jth- and $(j+1)$th-bit evaluation phase. In (26), the first term refers to the jth capacitance, whose bottom-plate is switched from ground to V_{DD}, while the summation refers to the capacitors whose bottom-plate remains at V_{DD} across the $(j+1)$th- and jth-bit evaluation steps. These capacitances contribute to the energy drawn from power supply because of the variation of the output voltage, which changes from $V_{out}(j+1)$ to $V_{out}(j)$. For the case depicted in Fig. 7, only C_4, which is the switched capacitance, and C_1 contribute to the energy drawn from the power supply and (26) reduces to

$$E(4) = [C_4(V_{DD} - V_{out}(4) + V_{out}(5))]V_{DD} + [C_1(-V_{out}(4) + V_{out}(5))]V_{DD}, \tag{27}$$

The implemented models compute the switching energy at each step on the basis of the DAC output voltage variation, which can be easily evaluated in CSAtool for each of the 2^N possible output codes. The overall switching energy is then obtained as the sum of the energies spent at all the conversion steps.

6. Design flow of a SAR ADC

The typical design flow (see Fig. 8) of a SAR ADC adopting a capacitive DAC requires several steps. The first is the design and the simulation of the overall converter schematic to assure its correct working and its linearity performance. Usually, the topology and the unit capacitance of the feedback DAC are chosen a priori to meet the required linearity specs (e.g. $3\sigma_{DNL} < 0.5$ [11,17]). Once the converter schematic has been established, the layout can be drawn and the stray capacitances extracted with the aid of a parasitic extraction tool. At this point, the same simulations performed on the schematic must be repeated on the post-layout view of the converter to assure that the parasitics do not degrade the linearity performance. To estimate the mismatch effect, also MonteCarlo simulations should be performed at this step. Since rarely the layout is satisfactory at the first attempt and it is hard to analytically predict, and thus minimize, the effect of the parasitics, post layout simulations should be repeated till the linearity requirements are respected.

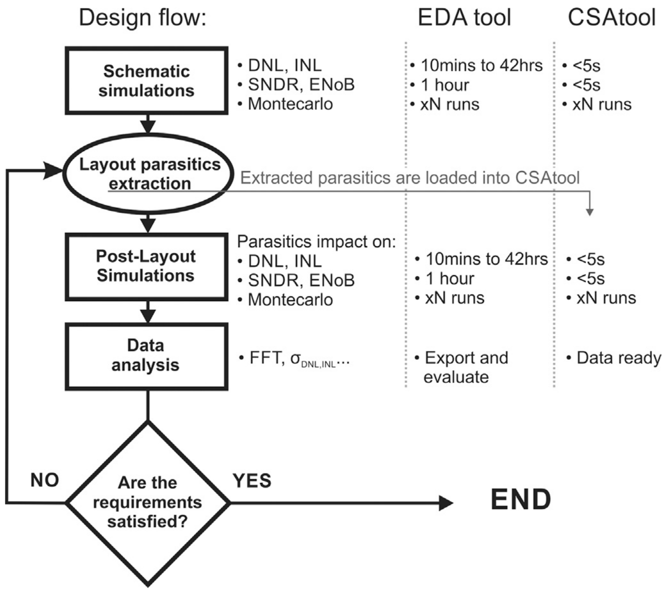


Fig. 8. Typical SAR ADC design flow with a comparison between traditional approach and CSAtool performance.

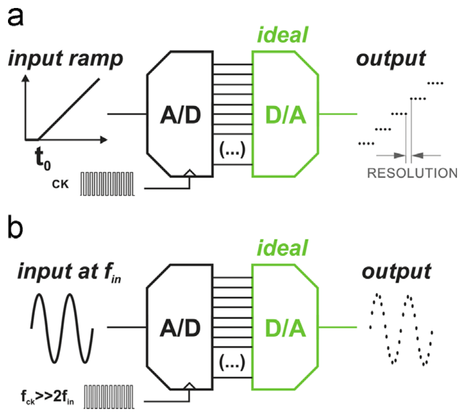


Fig. 9. Schematics of the (a) static and (b) dynamic performance evaluation with a traditional Spice-like simulator.

In particular, to evaluate the static characteristic with the traditional Spice-like simulators, a full-scale ramp is applied to the input of the ADC as shown in Fig. 9(a). To reduce the simulation time, behavioral models (Verilog or VerilogA) of the comparator and of the SAR logic circuit are adopted and only the ADC input and output signals are saved. The strobe and the sampling period are set short enough to guarantee at least 100 points per each conversion level, thus keeping the systematic error on the DNL below 1%. However, since the Spice-like simulator solves ordinary differential equations, computing current and voltage values at each time step, these simulations are very time consuming. On the contrary, in CSAtool the input–output characteristic is directly evaluated on the basis of the DAC output voltage levels by means of static operations among vectors, as shown is Section 3. Once the characteristic is given, the static metrics are easily derived.

Concerning the dynamic metrics estimation, in a traditional test-bench based on transient simulation (see Fig. 9(b)), the converter has to be driven by an analog sinewave according to the Shannon law. Its digital output is evaluated over a desired number of samples, which sets the simulation time. The latter can vary from tens of minutes to few hours. Then, the dynamic metrics, like SNDR and ENoB, can be evaluated exporting the output data into MATLAB in order to perform a FFT. On the contrary, in the CSAtool, the output sinewave is directly obtained according to the estimated input-to-output characteristic, greatly saving effort and time.

From the aforementioned discussion, its evident that the proposed tool is particularly useful in two cases, i.e. (i) to choose the most appropriate converter topology and the unit capacitance value for given area and linearity specs, and (ii) in the post-layout phase, where the designer typically adopts a trial-and-error approach that requires to continuously perform the assessment of the converter linearity metrics. In this case, a fast tool is mandatory to quickly get the linearity estimates, rather than performing heavy and long simulations in Cadence design environment. Clearly, the proposed method could not feature the same accuracy of transient-level simulations, even if the capacitive array, with its mismatch and parasitics, is typically the element limiting the converter linearity. This happens especially if advanced technology processes with reduced power supply are used, due to the leakage current affecting the switches, or when high conversion frequency is employed, since dynamic effects can affect the converter linearity more than the static inaccuracies of the input-output characteristic. Moreover, also the comparator can worse the linearity performance if it is not accurately sized due to signal-dependent input capacitance and kick-back noise. Its effect, however, can be assessed with a stand-alone simulation performed at transistor level, thus avoiding the simulation of the whole converter. In any case, the proposed MATLAB tool allows us to evaluate the ultimate limit of the converter linearity due to the capacitive array whose accuracy is degraded by parasitics and mismatch. Eventually, the tool results can be refined resorting to a more accurate simulation performed in a traditional design tool environment.

7. Simulation and measurements results

In this section, CSAtool results for three designed SAR ADCs prototypes are shown and compared to both analytical expressions and Cadence simulations in terms of accuracy and computation time. To isolate the DAC contribution to nonlinearities, all Cadence Virtuoso testbenches were created adopting a VerilogA description for the logic circuit and the comparator. The designed prototypes are:

- a 10-bit fully-differential SBW SAR ADC implemented in a 0.35 −μm CMOS AMS process adopting 23-fF PiP unit capacitors with a specific capacitance of 0.85 f F/μm^2 and a Pelgrom coefficient of 0.45% · μm;
- an 8-bit CBW SAR ADC designed in a 0.35−μm CMOS AMS process employing 80-fF poly-insulator-poly (PiP) unit capacitor with a specific capacitance of 0.85 f F/μm^2 and a Pelgrom coefficient of 0.45% · μm [3];
- a 10-bit BWA SAR ADC featuring a monotonic switching procedure and implemented in a 130-nm CMOS UMC process [5] with 34-fF MiM unit capacitors having a specific capacitance of 1 f F/μm^2 and a Pelgrom coefficient of 1% · μm.

The last two converters were also implemented as prototypes within the framework of different research projects, while the first one was only designed and layouted. The die micro-photographs of the fabricated converters and the layout view of the SBW prototype are shown in Figs. 10 and 11, respectively. For each converter, CSAtool and Cadence post-layout simulations were compared. For the two fabricated converters, measurement results were also compared to simulations. It is worth noting that a comparison between CSAtool and Cadence simulations is in general enough to validate the proposed tool but, on the other hand, a good matching with fabricated circuit performance represents a further evidence of reliability and accuracy of all the adopted simulation methodologies. This accuracy, with the significant reduction of the simulation time, makes CSAtool a suitable alternative to Cadence Spectre simulations for SAR ADC nonlinearity estimations.

7.1. Static metrics

Post-layout simulations for the 10-bit SBW topology performed with Cadence Spectre simulator have been compared to the CSAtool results. In these simulations, mismatch was not considered and only the parasitic capacitances, extracted from the layout

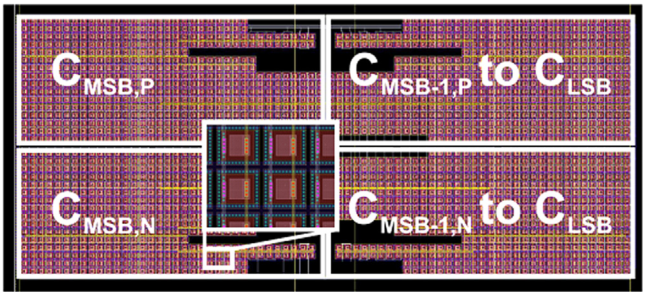


Fig. 11. Layout of the DAC of the prototyped SBW charge redistribution converter with the detail of the connections between the adopted PiP capacitors.

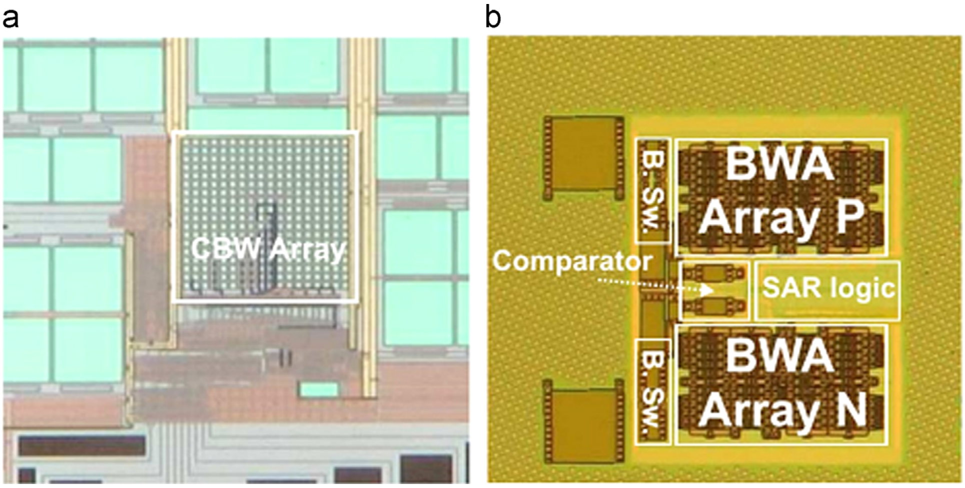


Fig. 10. Die photograph of the two measured prototypes adopting (a) an 8-bit single-ended CBW and (b) a 10-bit fully-differential BWA DAC.

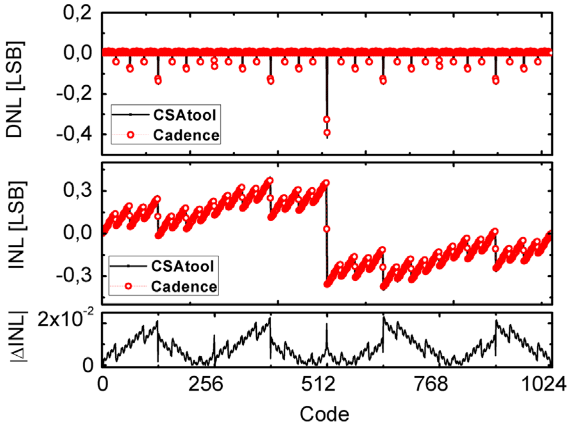


Fig. 12. Comparison between DNL and INL of the 10-bit SBW ADC prototype estimated by Cadence Spectre simulations (red markers) and by CSAtool (black lines). Also the absolute INL error is shown. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

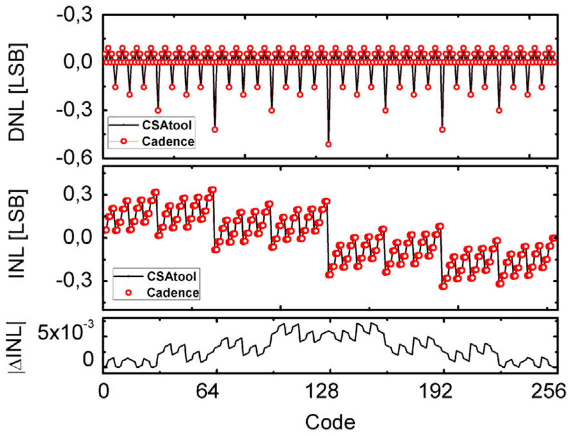


Fig. 13. Comparison between DNL and INL characteristics of the 8-bit CBW ADC prototype estimated by Cadence Spectre simulations (red markers) and by CSAtool (black lines). Also the absolute INL error is shown. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

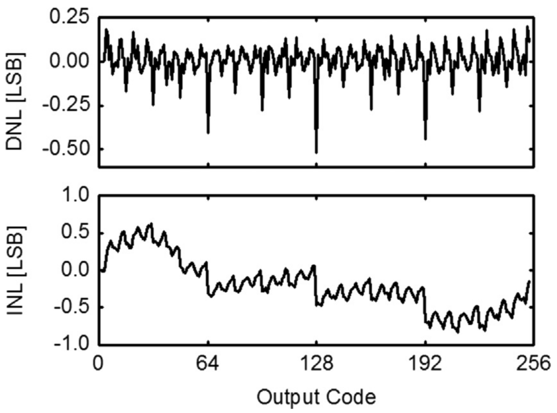


Fig. 14. Measured DNL and INL of the fabricated 8-bit CBW ADC prototype.

view, were taken into account. Fig. 12 shows the comparison of the estimated static metrics for the SBW DAC, highlighting excellent matching, since the difference is always less than 0.005 LSB for both the DNL and the INL curves, which is the resolution limit of the test-bench implemented in Cadence.

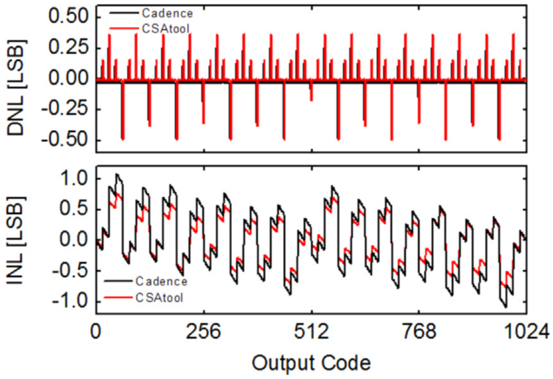


Fig. 15. Comparison between DNL and INL characteristics of the 10-bit BWA ADC prototype estimated by Cadence Spectre simulations (black lines) and by CSAtool (red lines). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

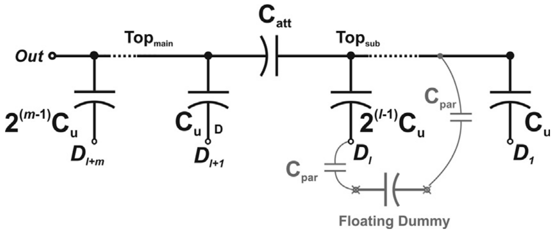


Fig. 16. Effect of floating dummy capacitors on the 10-bit BWA array.

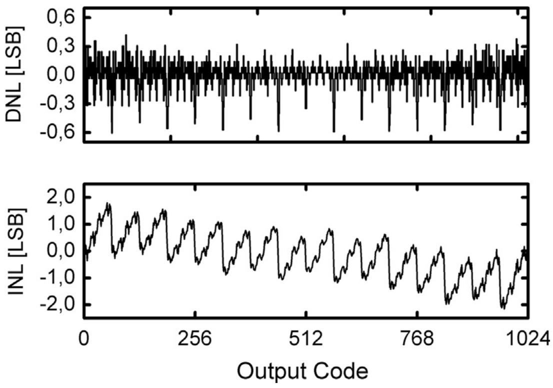


Fig. 17. Measured DNL and INL of the fabricated 10-bit BWA ADC prototype.

Similarly, Fig. 13 shows the comparison between the DNL and INL characteristics obtained by CSAtool and Cadence Spectre simulations for the 8-bit CBW converter. The matching is excellent with a maximum error of 0.07 LSB, confirming the good accuracy of the implemented converter model. Fig. 14 shows the measured static performance of the aforementioned 8-bit converter. The DNL curve shows a good matching with both Cadence and CSAtool estimation, while the INL, despite a similar pattern, drifts from the simulation results for the innermost and the outermost codes. These differences are mainly due to the effect of the comparator nonlinearity.

Fig. 15 shows the comparison between Cadence post-layout simulations and CSAtool results for the 10-bit fully-differential BWA converter. A difference between the estimated static nonlinearities up to 0.1 and 0.25 LSB for the DNL and the INL, respectively, can be observed. This difference has to be ascribed to the floating dummy capacitors that surround the array to improve

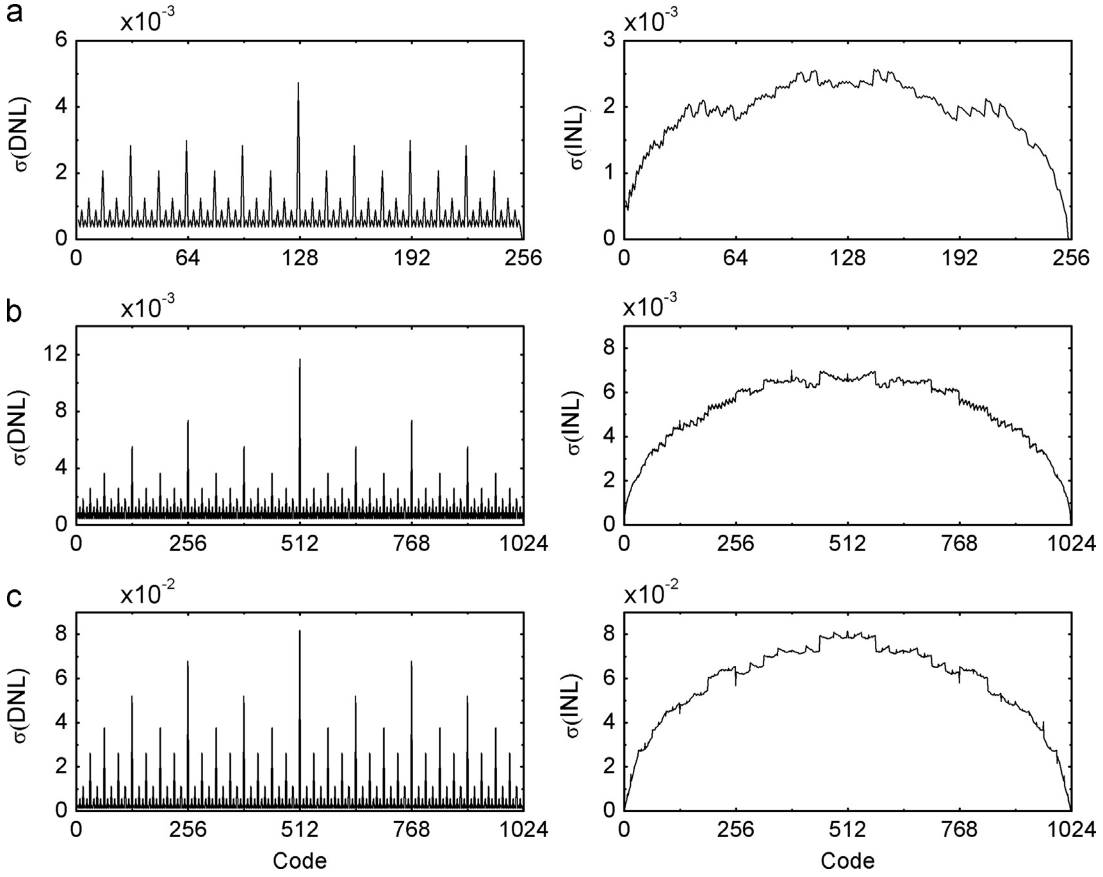


Fig. 18. Standard deviation of DNL and INL as a function of the output code for the (a) 8- and the (b) 10-bit SBW and (c) BWA converters considering the technology capacitive mismatch.

Table 1
Estimates of $\sigma_{DNL,max}$ and $\sigma_{INL,max}$.

Topology	$\sigma_{DNL,max}$		$\sigma_{INL,max}$	
	CSAtool	Equation	CSAtool	Equation
8-bit CBW s.e.	$4.93 \cdot 10^{-3}$	$5.25 \cdot 10^{-3}$	$2.51 \cdot 10^{-3}$	$2.62 \cdot 10^{-3}$
10-bit SBW f.d.	$9.89 \cdot 10^{-3}$	$9.74 \cdot 10^{-3}$	$7.05 \cdot 10^{-3}$	$6.89 \cdot 10^{-3}$
10-bit BWA f.d.	$84 \cdot 10^{-3}$	$81.4 \cdot 10^{-3}$	$84.4 \cdot 10^{-3}$	$81.4 \cdot 10^{-3}$

the matching property. These dummies can create a large number of cross-coupled parasitic capacitances.[1] However, the discrepancy is drastically reduced as soon as the dummy capacitors are connected to ground or to a reference voltage, with an error always lower than 0.05 LSB. Despite this problem, the measured results on the 10-bit converter shown in Fig. 17 show a good matching in terms of DNL with the simulated curves, while the measured INL differs considerably, even if it shows a similar pattern.

Fig. 18 shows the standard deviation of DNL and INL along the output codes for the 8- and 10-bit converters evaluated by CSAtool and considering the technology capacitive mismatch. This analysis is impractical in Cadence environment since it requires at least 100 static characteristic simulations to achieve confident results. Therefore, the results of CSAtool are compared in Table 1 to the analytic expressions of DNL and INL maximum standard deviation available in literature [15,6] and reported in Section 2. The discrepancy is always lower than 0.005 LSB. It is worth pointing out that for the 10-bit BWA prototype, the adopted monotonic switching procedure reduces the maximum DNL standard

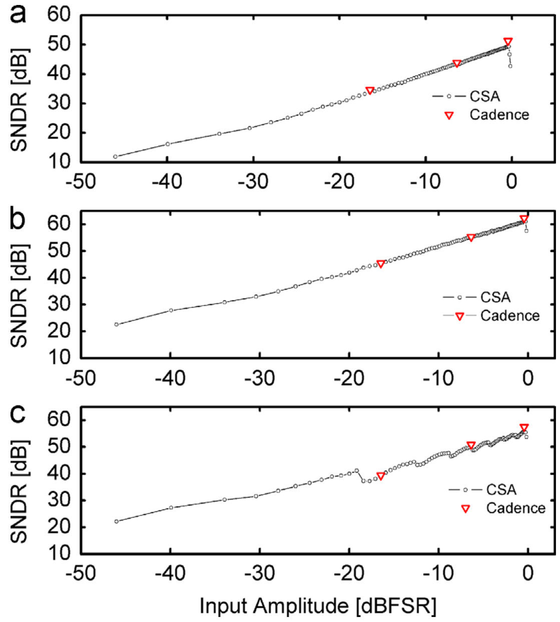


Fig. 19. Simulated SNDR as a function of the input signal amplitude for the (a) 8-bit CBW, (b) the 10-bit SBW and (c) 10-bit BWA designed converters.

[1] Also dummy metal fill in the array region can create an unwanted cross-coupling between the capacitances. In all the presented converters, metal fill above the array was accurately excluded, as shown in Fig. 16, which are difficult to be identified and that can drastically worsen the converter linearity.

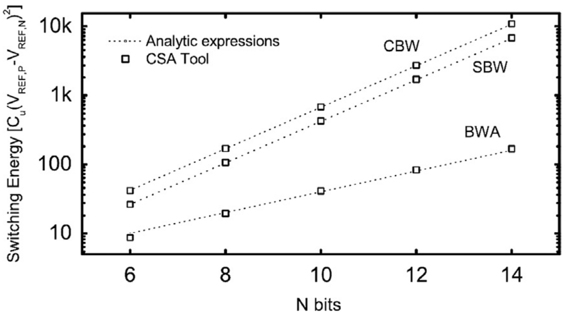


Fig. 20. Average switching energy for the three converter topologies. Dashed lines refer to analytic equations, symbols refer to CSAtool results.

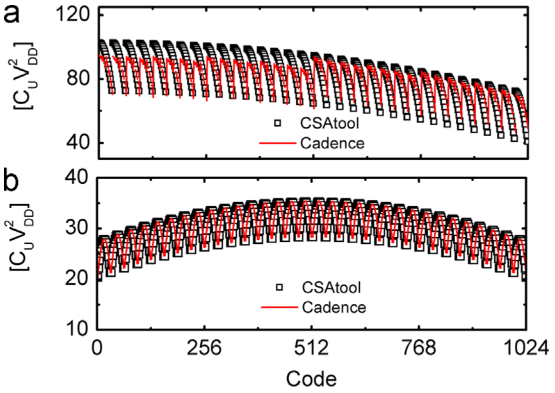


Fig. 21. Switching energy as a function of the output code and normalized to the unit element for a fully-differential BWA ADC employing the (a) traditional and (b) the monotonic switching algorithm. Black markers refer to CSAtool results while the red lines show Cadence simulation results. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 2
Single simulation time.

Metrics	Static		Dynamic	
Resolution	CSA (s)	Cad. (s)	CSA (s)	Cad. (s)
8-bit	0.087	$2.5\cdot10^3$	1.99	$4\cdot10^3$
10-bit	0.272	10^4	2.66	$4\cdot10^3$

deviation by a factor of 2 with respect to the traditional switching algorithm reported in [15].

7.2. Dynamic metrics

Fig. 19 shows the SNDR as a function of the input signal amplitude (referred to the full scale range) for the three implemented converters, evaluated by means of Cadence Spectre and CSAtool simulations. The effect of mismatch was taken into account for a single realization. The comparison between statistical simulations is not feasible due to too large amount of time required in Cadence to perform such simulations. The maximum discrepancy between Cadence and CSAtool results is always lower than 2 dB. However, its worth pointing out that CSAtool allows us to easily compute the SNDR vs. input amplitude curve, while the same analysis in a conventional EDA tool environment is time consuming, thus being possible to compute only few points of the dynamic characteristic. This can result in a not correct evaluation of the peak SNDR and thus of the ENoB.

Also a comparison between the measured and the simulated SNDR has been possible for the two fabricated ADCs. The 8-bit CBW converter achieved a measured SNDR of 46 dB, while the 10-bit BWA converter SNDR is 52.6 dB. The correspondent CSAtool estimations are 49 dB ± 1.7 dB and 56 dB ± 2 dB performed over 100 runs. These differences between the simulated values and the measurements were expected due to the presence of comparator nonlinearity, residual noise and dynamic effects which are not considered either in the tool DAC modeling and in the adopted Cadence test-benches. However, the achieved performance is included in a 2σ-width interval around the mean value.

Table 3
MonteCarlo simulation times for 100 runs.

Resolution	Static metrics (s)	Dynamic metrics (s)
8-bit	6.491	172
10-bit	25.25	183

7.3. Switching energy

Fig. 20 shows the average switching energies as function of the number of bits for the three considered single-ended array topologies evaluated by means of the analytic expressions presented in [15] and here reported:

$$E_{ave,CBW} \cong 0.66\cdot 2^N\left[C_u(V_{REF,P}-V_{REF,N})^2\right] \tag{28}$$

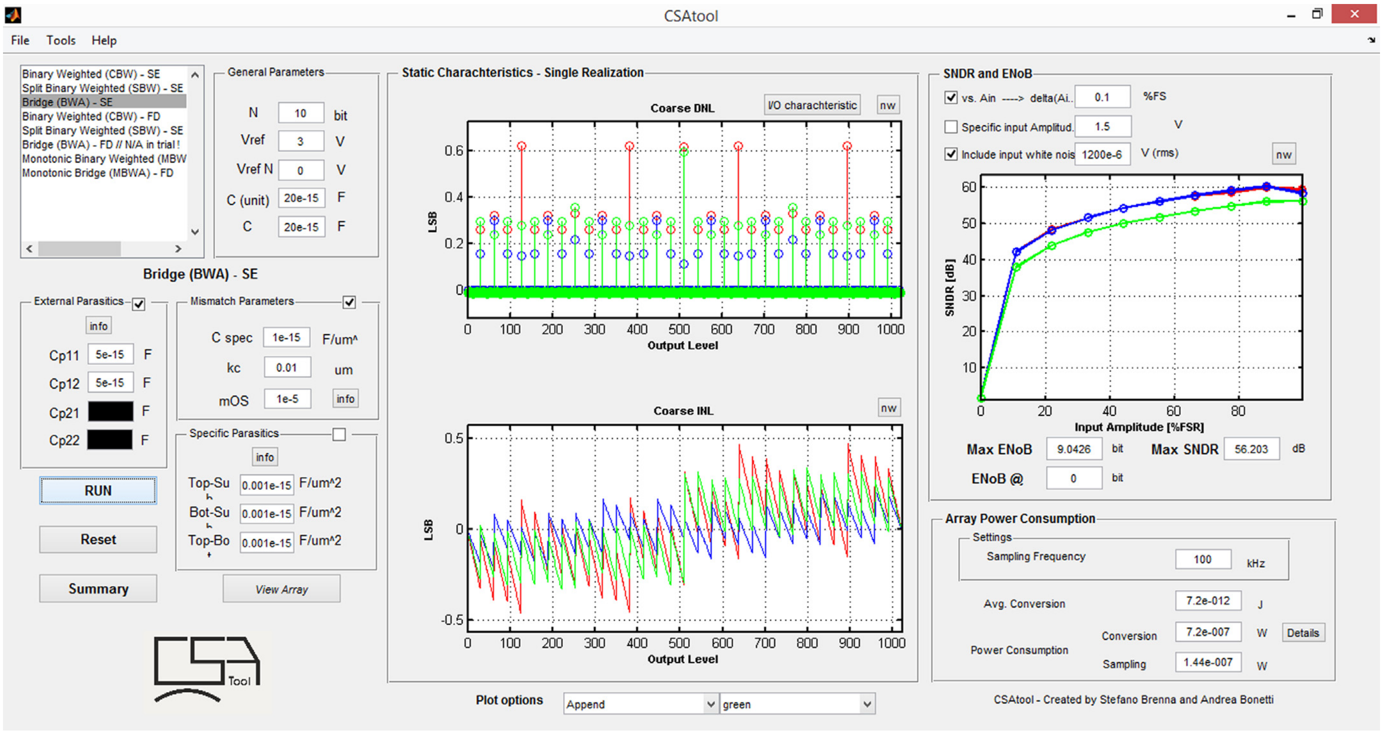
$$E_{ave,SBW} \cong 0.41\cdot 2^N\left[C_u(V_{REF,P}-V_{REF,N})^2\right] \tag{29}$$

$$E_{ave,BWA} \cong 1.25\cdot 2^{\frac{N}{2}}\left[C_u(V_{REF,P}-V_{REF,N})^2\right]. \tag{30}$$

The switching-energy for the fully-differential topologies can be obtained multiplying the above-mentioned equations by a factor of 2. The average energy evaluated with CSAtool is also reported in Fig. 20 and shows a good agreement with the theoretical estimates. As a further evidence of the CSAtool accuracy, the switching-energy as a function of the output code has been evaluated by means of transient simulations in Cadence for the 10-bit fully-differential BWA converter, with both the traditional and the monotonic switching algorithm. The simulated energy values, normalized to the unit capacitance, are shown in Fig. 21 and compared to CSAtool results. The positive and the negative reference voltages have been considered equal to power supply and ground, respectively. The estimated average energy of the BWA topology employing the traditional switching algorithm is 81.3 C_u V_{DD}^2 in CSAtool and 82.74 $C_uV_{DD}^2$ in Cadence, while for the monotonic switching procedure the estimations are 30 $C_uV_{DD}^2$ and 29.8 $C_uV_{DD}^2$, respectively. Similar results can be achieved for all the topologies supported by CSAtool approach, showing that the implemented models are also suitable for the estimation of the DAC switching energy.

7.4. Simulation time

Table 2 shows a comparison between the simulation times needed to compute the static and dynamic metrics with CSAtool and Cadence. The simulation times refer to the 8- and 10-bit CBW converters and to a single simulation run. All the simulations were performed with a 3-GHz Pentium Xeon featuring a 4-Gbyte main memory. For the same accuracy (i.e. DNL lower than 1%), CSAtool features an improvement in terms of simulation time up to 10^4. Table 3 shows the simulation time for a MonteCarlo analysis of 100 runs performed in CSAtool. For example, in a 10-bit converter, the tool allows us to compute the static and dynamic metrics in less than 5 min, while the same analysis in Cadence Virtuoso would require more than a week, being impractical.



Binary Weighted (CBW) - SE
Split Binary Weighted (SBW) - SE
Bridge (BWA) - SE
Binary Weighted (CBW) - FD
Split Binary Weighted (SBW) - SE
Bridge (BWA) - FD // N/A in trial !
Monotonic Binary Weighted (MBW
Monotonic Bridge (MBWA) - FD

General Parameters
N 10 bit
Vref 3 V
Vref N 0 V
C (unit) 20e-15 F
C 20e-15 F

Bridge (BWA) - SE

External Parasitics ☑
Cp11 5e-15 F
Cp12 5e-15 F
Cp21 F
Cp22 F

Mismatch Parameters ☑
C spec 1e-15 F/um^
kc 0.01 um
mOS 1e-5

Specific Parasitics □
Top-Sub 0.001e-15 F/um^2
Bot-Sub 0.001e-15 F/um^2
Top-Bot 0.001e-15 F/um^2

SNDR and ENoB
☑ vs. Ain ----> delta(Ai.. 0.1 %FS
□ Specific input Amplitud. 1.5 V
☑ Include input white nois 1200e-6 V (rms)

Max ENoB 9.0426 bit Max SNDR 56.203 dB
ENoB @ 0 bit

Array Power Consumption
Settings
Sampling Frequency 100 kHz
Avg. Conversion 7.2e-012 J
Power Consumption
Conversion 7.2e-007 W
Sampling 1.44e-007 W

Plot options Append green

CSAtool - Created by Stefano Brenna and Andrea Bonetti

Fig. 22. Screenshot of the CSAtool graphic user interface.

For the sake of clarity, it must be pointed out that the values related to dynamic metrics and reported in Table 3 refer to the evaluation of a single point of the dynamic characteristic. However, since the mismatch can significantly vary the SNDR plot, at least 10 MonteCarlo simulations (of 100 runs each) with different input amplitudes have to be performed to capture the peak SNDR, and thus correctly estimate the ENoB. In any case, there is a clear advantage of CSAtool with respect to Cadence. In fact, an equivalent analysis performed in the traditional EDA-tool environment would require weeks, since the time needed to estimate a single SNDR value takes more than 1 h.

8. Conclusions

A fast and accurate MATLAB-based tool, named CSAtool, for the analysis and design of the capacitive array of SAR ADCs has been presented. It allows us to compute both technology mismatch and parasitics effects on linearity performance and power consumption. The tool relies on static operations among vectors rather than on solving ordinary differential equations, thus greatly reducing the computation time if compared to common integrated circuit design environment. Moreover, it does not require a fine calibration of simulation parameters (time step, strobe period, etc.). CSAtool results show an excellent agreement with the conventional post-layout simulations performed on three designed converters and also a discrete matching with measurement results on two fabricated prototypes. A graphic user interface, shown in Fig. 22, eases the handling of the implemented models allowing to set the technology mismatch parameters, to load the parasitic capacitance pattern and to select the desired analysis. The proposed CSAtool is freely available online and can be directly requested by e-mail to the authors.

Finally, its worth pointing out that the proposed tool can be easily extended to other ratiometric circuits such as resistive or current steering DACs.

Acknowledgments

The authors would like to thank Paolo Maffezzoni for fruitful discussions.

References

[1] A. Agnes, E. Bonizzoni, F. Maloberti, A 9.4-ENOB 1 V 3.8 μW 100 kS/s SAR ADC with time-domain comparator, in: Digest of Technical Papers International Solid-State Circuits Conference, vol. 37, February 2, 2008, pp. 246–610.

[2] A. Agnes, E. Bonizzoni, F. Maloberti, Design of an ultra-low power SA-ADC with medium/high resolution and speed, in: Proceedings of the International Symposium on Circuits and Systems (ISCAS), 2008 pp. 1–4.

[3] A. Bonfanti, M. Ceravolo, G. Zambra, R. Gusmeroli, T. Borghi, A.S. Spinelli, A.L. Lacaita, A multi-channel low-power ic for neural spike recording with data compression and narrowband 400-MHz MC-FSK wireless transmission, in: Proceedings of the ESSCIRC, September 2010, pp. 330–333.

[4] S. Brenna, A. Bonetti, A. Bonfanti, A.L. Lacaita, A tool for the assisted design of charge redistribution SAR ADCs. in: Proceedings of the International Conference of Design, Automation and Test in Europe (DATE), March 2015.

[5] S. Brenna, A. Bonfanti, A.L. Lacaita, A 6-fJ/conversion-step 200 kSps asynchronous SAR ADC with attenuation capacitor in 130-nm CMOS, Analog Integr. Circuits Signal Process. 81 (August) (2014) 181–194.

[6] A. Chandrakasan, B. Ginsburg, 500-MS/s 5-bit ADC in 65-nm CMOS with split capacitor array DAC, IEEE J. Solid State Circuits 42 (April (4)) (2007) 739–747.

[7] A. Chandrakasana, B. Ginsburg, An energy-efficient charge recycling approach for a SAR converter with capacitive DAC, in: Proceedings of the International Symposium on Circuits and Systems (ISCAS), May 2005, pp. 184–187.

[8] J.A. Fredenburg, M.P. Flynn, Statistical analysis of ENOB and yield in binary weighted ADCs and DACs with random element mismatch, IEEE Trans. Circuits Syst. I: Reg. Paper 59 (July (7)) (2012) 1396–1408.

[9] H. Gao, et al., HermesE: a 96-channel full data rate direct neural interface in 0.13-μm CMOS, IEEE J. Solid-State Circuits 47 (April (4)) (2012) 1043–1056.

[10] P. Harpe, E. Cantatore, G. Haller, B. Murmann, A 2.2/2.7 fJ/conversion-step 10/12 b 40 kS/s SAR ADC with data-driven noise reduction, Digest of Technical Papers International Solid-State Circuits Conference, February 2013, pp. 270–271.

[11] P. Harpe, et al., A 26 μW 8 bit 10 MS/s asynchronous SAR ADC for low energy radios, IEEE J. Solid State Circuits 46 (July (7)) (2012) 1585–1595.

[12] W.S. Liew, X. Zou, Y. Lian, A 0.5 V 1.13-μW/channel neural recording interface with digital multiplexing scheme, in: Proceedings of the ESSCIRC, September 2011, pp. 219–222.

[13] C.C. Liu, S.J. Chang, G.Y. Huang, Y.Z. Lin, A 10-bit 50-MS/s SAR ADC with a monotonic capacitor switching procedure, IEEE J. Solid State Circuits 45 (April (4)) (2010) 731–740.

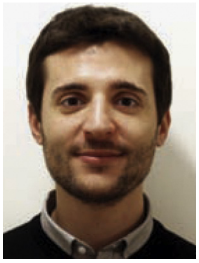
[14] J. McCreary, P. Gray, All-MOS charge redistribution analog-to-digital conversion techniques, I. IEEE J. Solid State Circuits SC-10 (December (6)) (1975) 371–379.

[15] M. Saberi, R. Lotfi, K. Mafinezhad, W. Serdjin, Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation ADCs, in: IEEE Transactions on. Circuits and Systems I: Regular Paper, vol. 58, August (7) 2011, pp. 1736–1747.

[16] M. van Elzakker, et al., 10-bit charge-redistribution ADC consuming 1.9 μW at 1 MS/s, IEEE J. Solid State Circuits 45 (May (5)) (2010) 1007–1015.

[17] D. Zhang, A. Bhide, A. Alvandpour, A 53-nW 9.1-ENOB 1-kS/s SAR ADC in 0.13-μm CMOS for medical implant devices, IEEE J. Solid State Circuits 47 (July (7)) (2012) 1585–1593.

S. Brenna was born in Monza, Italy, in 1986. He received the Bachelor degree in Biomedical Engineering and the Master degree in electrical engineering from Politecnico di Milano in 2009 and 2012, respectively. His Master research was focused on the design of a 130 nm CMOS integrated ultra-low power front-end for neural signal acquisition. He joined the Ph.D. program at the Politecnico di Milano in November 2012 and his current research activity is focused on the design of ultra low power analog mixed-mode integrated front-ends for the signal conditioning in neural probing integrated systems and also for MEMS and AMR magnetic field sensors.



A. Bonetti received the Bachelor's degree in Biomedical Engineering and the Master's degree in Electronic Engineering from Politecnico di Milano, Milano, Italy, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree at the Telecommunications Circuits Laboratory (TCL), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. In 2011, he visited the Bio Engineering Laboratory (BEL) at the Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. From 2012 to 2014, he worked at ams International AG, Rapperswil, Switzerland, developing interfaces for MEMS sensors. His current research activity is focused on timing monitoring and error-tolerance techniques for low-power VLSI signal processing.

A. Bonfanti was born in Besana Brianza (Milan), Italy, in 1972. He received the Laurea Degree in Electrical Engineering and the Ph.D. in Electronics and Communications from the Politecnico di Milano, Italy, in 1999 and 2003, respectively. Since 2009, he has been an Assistant Professor and subsequently Associate Professor of electrical engineering at Politecnico di Milano. His research interests include the study of phase noise in LC-tuned oscillators, and the design of low noise analog front-end and analog-to-digital converters for biomedical applications and MEMS inertial sensors. He is coauthor of approximately 50 papers published in journals or presented to international conferences.

A.L. Lacaita is a professor of Electronics at the Politecnico di Milano, Milano, Italy. He contributed to physics and technology of Single Photon Avalanche Diodes, advancements in semiconductor memory technologies and, in the field of ICs, to design optimization of low-noise VCOs, PLL frequency synthesizers and analog front-ends. He is co-author of more than 350 papers published in international journals or presented to international conferences. He has been visiting scientist/professor at the AT&T Bell Laboratories, Murray Hill, NJ (1989–1990), IBM T.J. Watson Research Center, Yorktown Heights, NY (1999) and Data Storage Institute, Singapore (2011).