A Tool for the Assisted Design of Charge Redistribution SAR ADCs

S. Brenna[1], A. Bonetti[2], A. Bonfanti [1], A. L. Lacaita[1]
[1] DEIB, Politecnico di Milano, Italy
[2] Telecommunications Circuits Laboratory (TCL), École Politechnique Fédérale De Lausanne (EPFL), Switzerland
email: stefano.brenna@polimi.it , andrea.bonetti@epfl.ch

***Abstract*—The optimal design of SAR ADCs requires the accurate estimate of nonlinearity and parasitic effects in the feedback charge-redistribution DAC. Since the effects of both mismatch and stray capacitances 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 tool (CSAtool) 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 thus computing both differential and integral nonlinearity (DNL, INL). SNDR and ENoB degradation due to static non-linear effects is also estimated. An excellent agreement is obtained with the results of circuit simulators (e.g. Cadence Spectre) featuring up to 10^4 shorter simulation time, allowing a large number statistical simulations which would be otherwise impracticable. Measurements on two fabricated SAR ADCs confirm that the proposed tool can be used as a valid instrument to assist the design of a charge redistribution SAR ADC and predict its static and dynamic metrics.**

***Keywords*—Analog-to-digital conversion, assisted design, numerical tools, charge redistribution successive approximation registers ADC.**

I. Introduction

Efficient analog-to-digital converters (ADCs) are essential building blocks of low-power applications. In terms of energy saving, successive approximation register (SAR) converters are the best choice when moderate resolution and conversion speed are required. Starting from the Classic Binary Weighted (CBW) SAR ADC [1], in the last decade, other solutions have been proposed for better efficiency [2], [3] and adopted in low-power systems [4]. In all cases, both static and dynamic performance figures strongly depend on mismatch and parasitic capacitances affecting the capacitive array of the feedback digital-to-analog converter (DAC, see fig.1).

There is not any quantitative guideline available to address non-linearities arising from parasitic capacitances. This term is deterministic and strongly depends on the array architecture and on the layout quality. These effects are therefore addressed and minimized relying on transient simulations in integrated circuit design tools, such as Cadence. Unfortunately, such a procedure is extremely time-consuming and require heavy data post-processing to estimate the Signal to Noise and Distortion Ratio (SNDR) and the Equivalent Number of Bits (ENoB). To overcome these limitations, the authors proposed a MATLAB-based tool [5] to assist and speed-up the design of the capacitive DAC adopted in SAR ADCs.

The tool handles the models of the most commonly adopted DAC topologies, among which appear the Classic Binary Weighted (CBW) and the Binary Weighted with Attenuation Capacitor (BWA) [6] array using various switching algorithms. The numerical models take into account both mismatch and parasitics effects providing the estimate of differential (DNL) and integral (INL) non-linearities, SNDR and ENoB on the basis of statistical and parametric simulations of the converter. The aim of this work is to evaluate the accuracy and the time saving of the proposed tool with respect to the traditional

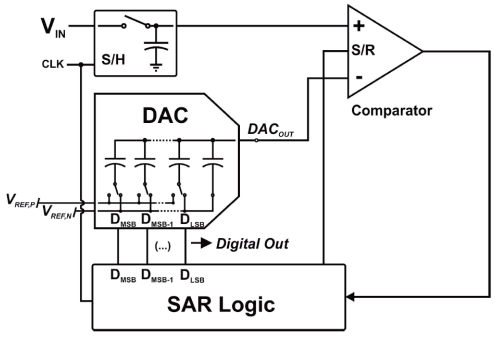

Fig. 1: Generic SAR ADC architecture.

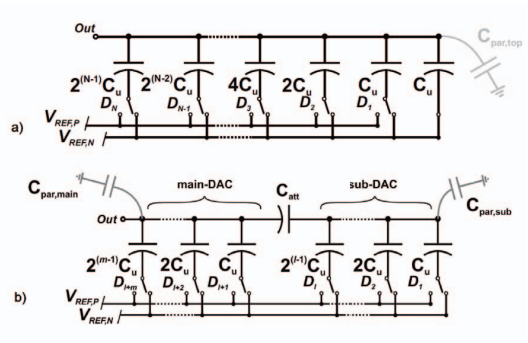

Fig. 2: Schematic of a) a N-bit CBW array and b) a (m + l)-bit BWA array .

Cadence Spectre simulations on two designed and fabricated SAR ADCs. Also a comparison with measurement results allows to appreciate the validity of the tool. The paper is organized as follows. Section II summarizes the effects of mismatch and parasitic capacitances on both DNL and INL of the most common capacitive array topologies. Section III briefly resumes the approach adopted in the tool to handle the array models and to compute the converter output. Section IV shows the most significant results compared to Cadence Spectre simulations and measurement results on two designed and integrated ADCs. Finally, conclusions are drawn in Section V.

II. Converter Topologies

The capacitive network adopted in SAR ADCs can be described as a composition of one or more binary weighted array connected to the output node either in parallel or through a capacitive attenuation network. Depending on the digital code, 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. Fig. 2a shows a simple N-bit capacitive DAC where, for minimum parasitic impact, each capacitive block is oriented with the bottom plate towards the input voltage reference lines. From a formal standpoint, the

capacitance of the i^{th} capacitive block of the array is the binary sum of unit capacitors C_u (i.e. $C_i = 2^{i-1}Cu$) plus the contribution due to stray capacitances. These terms can be divided into two groups: (i) those due to stray capacitances between the top- and the bottom-plate nodes of each capacitive element and (ii) those from either the top and the bottom plate of each element and fixed reference nodes. All terms referred to parasitics are deterministic, depending on wiring and layout geometry. Thus, once the array is designed, their impact on DNL and INL performance can be addressed by computing the converter characteristic. On the contrary, the capacitor mismatch causes a statistical error. Closed formulae are indeed available to estimate the maximum standard deviation of DNL and INL [7], [8]. In the CBW array the maximum DNL and INL values occur for the mid-code. For a single-ended array, the maximum DNL standard deviation is given by

$$\sigma_{DNL,CBW} = 2^{\frac{N}{2}} \cdot \frac{\sigma_C}{C_u}, \tag{1}$$

where σ_C is the standard deviation of the unit capacitor C_u and depends on the Pelgrom mismatch coefficient k_c and the specific capacitance c_{spec} of the adopted technology, according to the following expression:

$$\sigma_C = k_c \cdot \sqrt{\frac{c_{spec} \cdot C_u}{2}}. \tag{2}$$

The corresponding maximum standard deviation value for the INL is

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

In a fully-differential configuration, the impact of mismatch on DNL (and INL) has to be divided by a factor $\sqrt{2}$ [9] . Based on these considerations, 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 addressed by circuit simulations. In a single-ended BWA, the capacitive array is divided into two binary weighted arrays separated by an attenuation capacitor C_{att} (Fig. 2b) [6].

Despite the modeling approach can be extended to asymmetrical configurations, in this work we will consider the case in which both DACs have the same number of bits (i.e. $m = l = N/2$) and $C_{att} = C_u$. This topology leads in fact to the most energy efficient solution [7]. Closed formulae analogous to eq. (1) (2) are presented in [7] for the single ended BWA topology. Regarding the impact of the parasitics, in addition to the top-bottom plate capacitances also the stray capacitance connected to the top-plate node of the sub-DAC ($C_{par,sub}$ in Fig. 2b) affects the linearity since it makes the DAC output voltage depending on the input code.

III. Modeling Approach

The proposed MATLAB tool (CSAtool) computes the input-to-output characteristic of the different capacitive DACs with an even number of bits between 6 and 14. For each topology, a behavioral model which handles both parasitics and mismatch is provided. The models do not implement the known equations that estimate the non-linearity (i.e. the maximum standard deviation of DNL and INL), but they rather reproduce the behaviors of the specific circuits architectures which are described by functional capacitive blocks. The simulations can be customized, giving the possibility to evaluate the impact of mismatch, through both single and multiple statistical runs, as well as of the parasitics of each specific array capacitor, thus representing a suitable alternative to Cadence MonteCarlo and post-layout simulations. From a modeling standpoint, in any charge redistribution DACs the actual value of each binary weighted capacitive bank can be written as the sum of different contributions:

$$C_i = 2^{i-1}C_u + \sum_{j=1}^{2^{i-1}} \delta_j + C_{par,i}, i = 1, ...N, \tag{4}$$

where the first term is its nominal value (expressed as the sum of unit elements) and the terms δ_j represent the mismatch affecting each of the unit capacitors that compose the block. The latter is derived at each run by sorting the actual value of each unit element from a Gaussian distribution with average value C_u , and standard deviation σ_C as defined in eq. 2. The term $C_{par,i}$ 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 capacitive element. In general, all parasitics can be computed with the aid of a layout parasitic extractor. In a generic SAR ADC, the analog-to-digital conversion is effectively performed by comparing the input analog voltage signal with subsequent voltage levels generated by the capacitive DAC. The sorted set of all the possible DAC output voltage represent the input transition levels of the conversion characteristic. In the next paragraphs is briefly explained how each of these levels can accurately be evaluated by the implemented models.

A. CBW Model

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

$$DAC_{out} = FSR \cdot H, \tag{5}$$

where FSR is the full scale range of the converter and H is the scalar product

$$H = \frac{1}{C_{tot} + C_{par,top}} \times \bar{C} \times \bar{D}'. \tag{6}$$

In the last equation, C_{tot} is the total capacitance of the array, $C_{par,top}$ is the parasitic capacitance shown in Fig. 2, $\bar{C}$ is the vector of the array capacitances C_i (see eq. (4)) and $\bar{D}$ is the vector of the digital word updated at each conversion cycle:

$$\bar{C} = [\ C_1 \quad \dots \quad C_N\ ] \tag{7}$$

$$\bar{D} = [\ D_1 \quad \dots \quad D_N\ ]. \tag{8}$$

The latter, which encodes the DAC output levels at each conversion step, is determined by the adopted switching algorithm. By means of eq. (6), (7) and (8) it is then possible to compute the ADC conversion characteristic.

B. BWA Model

As far as the BWA topology concerns, two equal capacitive arrays must be considered: a main-DAC and a sub-DAC, which are related to the most- and to the least-significant bits, 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 capacitance at the top-plate node of the corresponding DAC (see Fig. 2b). 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:

$$AR = \frac{C_{att}}{C_{tot,main} + C_{par,main} + C_{att}}, \tag{9}$$

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

$$DAC_{out} = FSR \cdot (H_{main} + H_{sub}) \tag{10}$$

where H_{main} and H_{sub} are coefficients similar to the factor H introduced for the CBW model, but related to the main and sub-DAC, and that take into account the capacitances and digital output code vectors related to the main- and the sub-DAC, respectively.

C. Design Flow

The typical design flow of a SAR ADC adopting a capacitive DAC requires several steps. The first is the schematic simulation, in which the DAC topology is chosen and the circuit is simulated to ensure its correct operation. Once the schematic is assessed, the layout can be designed and the related parasitics extracted with the aid of a parasitic extraction tool. At this point, the same simulation performed in the schematic must be repeated in the post-layout phase to verify that parasitics do not cause the breach of the requirements. To estimate the mismatch effect, also MonteCarlo simulations should be performed in this step. Since rarely the layout is satisfying at the first time, and since it is hard to analytically predict and then minimize the effect of each parasitic, post layout simulations could be repeated until the requirements are respected with a desired margin. To evaluate the static characteristic with a Spectre transient simulation, a full-scale ramp is applied to the input of the ADCs and, to reduce the simulation time, only the converter input and output values were saved. In CSAtool, on the contrary, the input-to-output characteristic is directly evaluated on the basis of the DAC output at each code. This operation can be computed in MATLAB by a product of vectors and taking only few seconds.

For what concerns dynamic metrics estimation, in a traditional testbench based on transient simulation, the converter samples a sinusoid according to Shannon's 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. In CSAtool, SNDR and ENoB are computed by driving the ADCs model with a numerical sine-wave. The signal amplitude is varied from the 1% and the 100% of the full-scale range, while the sampling rate is arbitrarily set well over Nyquist. The digital output is then stored and its corresponding spectrum is computed by applying FFT. Being each transient simulation usually time consuming, the only DAC design adopting the traditional flow could take from several hours to days. By using CSAtool the design time can be significantly reduced, allowing also to perform statistical simulations that, otherwise, would be unpractical.

IV. Simulation and Measurement Results

To isolate the DAC contribution to non-linearities, all Cadence Virtuoso testbenches were created adopting a VerilogA description for the logic circuit and the comparator. The designed and fabricated prototypes are the following:

- an 8-bit CBW SAR ADC realized in a 0.35-μm CMOS AMS process adopting 80-fF PiP unit capacitors with a specific capacitance of $0.85fF/\mu m^2$ and a Pelgrom coefficient of 0.45%·μm [10];
- a 10-bit BWA SAR ADC featuring a monotonic switching procedure and implemented in a 130-nm CMOS UMC process [11] adopting a 20-fF MiM unit capacitors with a specific capacitance of $1fF/\mu$m^2 and a Pelgrom coefficient of 1%·μm.

Measurement results on these implemented ADCs are also compared to both CSAtool and Cadence post-layout simulations.

A. Static Metrics

Fig. 3a shows the comparison between the DNL and INL characteristics obtained by CSAtool and Cadence Spectre simulations for the 8-bit CBW converter. In this run mismatch was not included and only the parasitic capacitance values extracted in Assura and loaded into CSAtool were considered. Simulation results between Cadence and CSAtool show excellent matching, with a maximum error below 0.02LSB, confirming

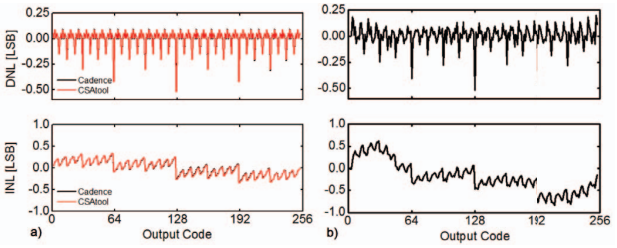

Fig. 3: Comparison between DNL and INL of the 8-bit ADC prototype estimated by a) Cadence simulations (black lines), by CSAtool (red lines) and b) measured.

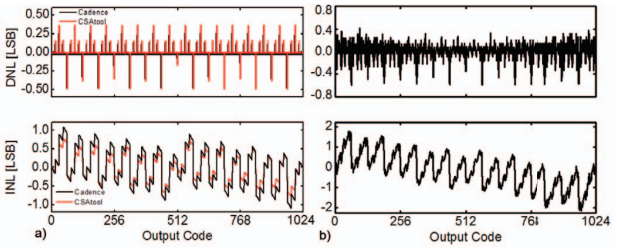

Fig. 4: Comparison between DNL and INL of the 10-bit ADC prototype estimated by a) Cadence simulations (black lines) , by CSAtool (red lines) and b) measured.

the good accuracy of the implemented converter models. As shown in Fig. 3b, also the measured DNL shows a good matching with both Cadence and CSAtool estimation while the INL, despite a similar pattern, drifts from the simulation results at the peripheral output codes. This differences may be due to the technological mismatch of the tested sample and the comparator non-linearity. The comparison between Cadence Post layout simulations and results of CSA tool for the 10-bit fully differential and monotonic BWA designed with MiM unit capacitors of 34fF is illustrated in Fig. 8. The mismatch between CSAtool and Cadence post-layout INL is due to the fact that the array layout includes some floating dummy capacitors that introduce parasitics that were correctly extracted in Assura but that were hard to be completely identified with parasitic probing. For this reason, the equivalent parasitic set loaded into CSAtool was not complete, resulting in a difference between the estimated static non-linearities up to 0.1 and 0.25LSB for the DNL and the INL, respectively. The discrepancy can be recovered by a more accurate design, for example connecting all the dummy capacitors terminals to a reference or to ground. Despite these design inaccuracies the DNL matching between measurements and simulations is evident, while the measured integral non-linearity is worse but shows a similar pattern. Fig. 5 shows the results obtained by taking only into account the mismatch. MonteCarlo simulation performed in the Cadence environment requires at least 100 runs to achieve confident results and the simulation times are absolutely impractical. The results from CSAtool were therefore compared to the analytical expressions from literature [7] also listed in section II. The error obtained after 100 runs in CSAtool is always lower than 0.005LSB. It is worth noting here that for the 10-bit BWA topology, the adopted monotonic switching procedure reduces the standard deviation of the DNL by a factor 2 with respect to the case of the traditional switching algorithm.

B. Dynamic Metrics

Fig. 6 shows the SNDR evaluated by means of Cadence Spectre simulations and CSAtool as function of the input signal amplitude (referred to the full scale range) for the implemented converters. Effects of mismatch were not taken into account. The maximum discrepancy between the Cadence

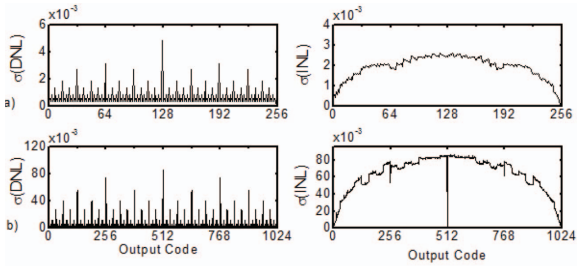

Fig. 5: Standard deviation of DNL and INL as functions of the output code for the a) 8-bit and b) 10-bit converter prototypes considering the capacitive mismatch.

TABLE I:
SINGLE SIMULATION TIME

	Static Metrics		Dynamic Metrics	
Prototype	CSAtool	Cadence	CSAtool	Cadence
8-bit	$0.087s$	$2.5{\cdot}10^3$	1.99s	$4{\cdot}10^3$
10-bit	$0.272s$	10^4	2.66s	$4{\cdot}10^3$

and CSAtool results is always lower than 2dB. However, it is worth pointing out that CSAtool allows to easily compute the SNDR vs. input amplitude curve, while the same simulations in Cadence take a long time allowing to compute only few points of the dynamic characteristic. This can result in a not correct evaluation of the SNDR peak and thus of the ENoB. Also a comparison between the measured and the simulated SNDR has been possible for the two fabricated ADCs previously presented. The 8-bit CBW ADC achieved a measured SNDR of 46dB, while the 10-bit converter SNDR is 52.6dB SNDR. The correspondent tool estimations are of 49dB$\pm$1.7dB and 56dB$\pm$2dB respectively, performed over 100 runs. These differences between the simulated values and the measurements were expected due to the presence of the comparator non-linearity, residual noise and dynamic effects which are not considered either in the tool DAC modeling and in Cadence. However, the achieved performance is still comprised within a two-sigma width interval centered on the mean estimation.

C. Simulation Time

Table I shows a comparison between the simulation times needed to compute a single realization of the static and dynamic metrics with the CSAtool and Cadence environment for the two designed prototypes. All the simulations have been performed with a 3-GHz Pentium Xeon featuring a 4-Gbyte main memory. For the same accuracy (i.e. DNL error lower than 1%, the CSAtool features an improvement in terms of simulation time up to 10^4. Finally, it's worth pointing out that the CSAtool allows to estimate the static nonlinearities and the dynamics metrics in presence of statistical mismatch and parasitic capacitances allowing to perform a MonteCarlo

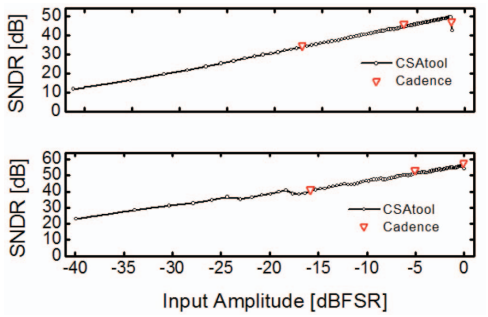

Fig. 6: Simulated SNDR as a function of the input signal amplitude for the two designed converters.

simulation over a large number of runs. For what concerns the simulation time of a 100runs MonteCarlo analysis , for a 10-bit converter i.e., the tool allows to compute the static and dynamic metrics in less than 200s, while the same analysis requires more than 2 hours in Cadence Virtuoso environment. For higher resolutions the time saving allowed by CSAtool is even larger.

V. CONCLUSIONS

A fast and accurate simulation tool for the analysis and design of charge redistribution SAR capacitive DACs and ADCs has been presented. A graphic user interface MATLAB environment supports the implemented models that allow to simulate both technology mismatch and parasitic effects on linearity. The tool overcomes the limitations of conventional simulation methods based on transient analyses in terms of computation time and does not require a fine calibration of simulation parameters (points per conversion step, strobe period, etc.). In particular, the tool is helpful every time a fast analysis and an accurate sizing of commonly-used converter architectures is needed. Simulation results show an excellent agreement with conventional post-layout simulations and also a good matching with measurements results of fabricated samples. A trial version of the proposed CSAtool can be requested by contacting the authors.

REFERENCES

[1] J. McCreary and P. Gray, "All-MOS charge redistribution analog-to-digital conversion tecqniques. I," *IEEE J. of Solid State Circuits*, vol. SC-10, no. 6, pp. 371–379, Dec. 1975.

[2] M. van Elzakker *et al.*, "10-bit charge-redistribution ADC consuming 1.9μW at 1 MS/s," *IEEE J. of Solid State Circuits*, vol. 45, no. 5, pp. 1007–1015, May 2010.

[3] P. Harpe, E. Cantatore, G. Haller, and B. Murmann, "A 2.2/2.7fJ/conversion-step 10/12b 40kS/s SAR ADC with data-driven noise reduction," *Dig. Tech. Papers Int. Solid State Circuits Conf.*, pp. 270–271, Feb. 2013.

[4] W. S. Liew, X. Zou, and Y. Lian, "A 0.5V 1.13-μW/channel neural recording interface with digital multiplexing scheme," *Proceedings of ESSCIRC*, pp. 219–222, Sept. 2011.

[5] S. Brenna, A. Bonetti, A. Bonfanti, and A. L. Lacaita, "A simulation and modeling environment for the analysis and design of charge redistribution DACs used in SAR ADCs," *Proceedings of the 37th Int. MIPRO Conf.*, May 2014.

[6] A. Agnes, E. Bonizzoni, and F. Maloberti, "Design of an ultra-low power SA-ADC with medium/high resolution and speed," *Proc. Int. Symp. on Circ. and Syst. (ISCAS)*, pp. 1–4, 2008.

[7] M. Saberi, R. Lotfi, K. Mafinezhad, and W. Serdjin, "Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation ADCs," *IEEE Trans. Circuits Syst. I: Reg. Paper*, vol. 58, no. 7, pp. 1736–1747, Aug. 2011.

[8] A. Chandrakasan and B. Ginsburg, "500-MS/s 5-bit ADC in 65-nm CMOS with split capacitor array DAC," *IEEE J. Solid State Circuits*, vol. 42, no. 4, pp. 739–747, Apr. 2007.

[9] D. Zhang, A. Bhide, and 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*, vol. 47, no. 7, pp. 1585–1593, Jul. 2012.

[10] A. Bonfanti *et al.*, "A multi-channel low-power ic for neural spike recording with data compression and narrowband 400-MHz MC-FSK wireless transmission," *Proceedings of ESSCIRC*, pp. 330–333, Sept. 2010.

[11] S. Brenna, A. Bonfanti, and A. L. Lacaita, "a 6-fJ/conversion-step 200kSps asynchronous SAR ADC with attenuation capacitor in 130-nm CMOS," *Analog Integrated Circuits and Signal Processing*, vol. 81, Aug 2014.