A hybrid design automation tool for SAR ADCs in IoT

Citation for published version (APA):
Ding, M., Harpe, P., Chen, G., Busze, B., Liu, Y-H., Bachmann, C., Philips, K., & van Roermund, A. (2018). A hybrid design automation tool for SAR ADCs in IoT. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *26*(12), 2853-2862. [8463579]. https://doi.org/10.1109/TVLSI.2018.2865404

DOI:
10.1109/TVLSI.2018.2865404

Document status and date:
Published: 01/12/2018

Document Version:
Accepted manuscript including changes made at the peer-review stage

A Hybrid Design Automation Tool for SAR ADCs in IoT

Ming Ding, *Member, IEEE,* Pieter Harpe, *Senior Member, IEEE,* Guibin Chen, Benjamin Busze, Yao-Hong Liu, *Senior Member, IEEE,* Christian Bachmann, Kathleen Philips, Arthur van Roermund, *Senior Member, IEEE*

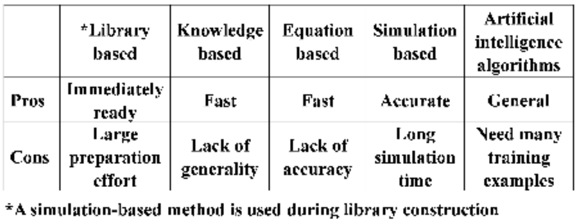
***Abstract*—In this work, a hybrid design automation tool for asynchronous SAR ADCs in IoT applications is presented. The circuit-design-driven tool uses a top-down design approach, and generates circuits from specification to layout automatically. A hybrid approach is introduced for different circuits of a SAR ADC: fully-synthesized control logic; a script-based flow combining equations, library, and template-based design for the DAC; a Look-Up-Table (LUT) approach combined with selective simulation-based fine tuning and template-based layout generation for the S&H; library-based comparator design and script-based layout generation. By balancing the automation and manual effort, the circuit design time is reduced from days down to minutes while still being able to maintain ADC performance. The proposed flow generated two ADC prototypes in 40nm CMOS, an 8b 32MS/s and a 12b 1MS/s SAR ADC, and enabled excellent power-efficiency. The two ADCs consume 187μW and 16.7μW at 1V supply voltage, achieving 30.7fJ/conversion-step and 18.1fJ/conversion-step respectively.**

***Index Terms*—Hybrid approach, Design Automation, SAR ADC, low-power**

I. INTRODUCTION

Today, many System-on-Chips (SoCs) in Internet-of–Things (IoT) applications consist of both digital and analog circuitry for a high integration level and low cost. The highly integrated chips usually consist of millions or billions of devices, and this results in huge circuit design effort. The digital circuit design has already been automated by many mature commercialized tools and flows, from behavior description to layout generation. The analog circuits are often manually designed for both schematic and layout. As a result, although the analog circuit scale in a SoC is usually small compared to that of the digital part, the design time for analog circuits is still long. While more and more functions are moved from analog domain to digital domain, analog-to-digital converters (ADCs) as an interface between analog domain and digital domain are still required. Many Wireless Sensor Nodes (WSN) in IoT/IoE applications (e.g., Bluetooth-Low-Energy (BLE) and IEEE 802.15.4) require low-power Analog-to-Digital Converters (ADCs) with varying resolution (8$\sim$12bit) and speed ($\sim$MS/s). To achieve an optimum performance, each ADC has to be customized, increasing the design cost and time. To overcome this issue, circuit design automation is desired for ADCs to minimize design cost while still maintaining performance.

TABLE I
SUMMARY OF THE AUTOMATION METHODS FOR FRONT-END SYNTHESIS.



	*Library based	Knowledge based	Equation based	Simulation based	Artificial intelligence algorithms
Pros	Immediately ready	Fast	Fast	Accurate	General
Cons	Large preparation effort	Lack of generality	Lack of accuracy	Long simulation time	Need many training examples

*A simulation-based method is used during library construction

The design automation for analog and mixed-signal circuits includes roughly two steps: front-end synthesis and back-end generation. Front-end synthesis can include architecture selection, translation of functional specification to sub-circuit specification, device sizing, and back-end generation can include layout generation of sub-blocks as well as floor planning and routing.

The approaches for front-end synthesis can be roughly divided into five categories as shown in Table I: library-based approach, knowledge-based approach, equation-based approach, simulation-based approach, and artificial-intelligence-based approach. A library-based approach requires a large effort to create a new library for each block in each technology. This may be acceptable for a single simple cell, but not for a complicated system, which includes many blocks. The knowledge-based approach encodes specific heuristic design knowledge from experts into a design plan that is used during the synthesis of the analog circuit [1]. Specification inputs will be translated to the topology selection and the unique solution of the circuit sizing following the design plan. However, a content-independent design plan is difficult to make, which limits the use of this approach. The equation-based approach uses a simplified analytic equation to formulate the performance of the circuit [2], [3]. Constrained optimization algorithms instead of a specific design plan are performed using these equations for optimization of the circuit. Although this approach is more general, the accuracy of the result is a big problem especially in an advanced process because the design equation has to be derived and simplified so that the optimization algorithm can be executed. To obtain higher accuracy, a full-SPICE simulation-based approach is introduced in the optimization loop [4]. The main problem of this approach is the long run time, especially if the initial search space is not

Ming Ding is with IMEC-NL/Holst Centre and Eindhoven University of Technology, Eindhoven, the Netherlands. (Ming.Ding@imec-nl.nl). Guibin Chen was with IMEC-NL/Holst Centre and Eindhoven University of Technology, Eindhoven, the Netherlands, and is now with Tencent Technology, China. Benjamin Busze, Yao-Hong Liu, Christian Bachmann, and Kathleen Philips are with IMEC-NL/Holst Centre, Eindhoven. Pieter Harpe and Arthur van Roermund are with Eindhoven University of Technology, Eindhoven, The Netherlands.

well-defined. In [5], an artificial intelligence based approach uses the idea of evolution to automate the synthesis of analog circuits. However, large amounts of well-designed examples are required for training the neural network.

For the back-end automation, four methods have been used: library-based, script-based, template-based, and standard digital flow. In the library-based method, the layouts of the required components are manually designed beforehand and put together in a library, which can be directly reused later on. The layout can be precisely optimized, but it takes a long design time. The script-based method [6] describes the transistors' location and inter-connection in pure ASCII instead of manual place and route. The template-based method uses a template for a design, and can maintain the layout shape irrespective of different sizes [7]. The standard digital flow is very mature for digital circuit place and route, but is used less often for customized analog layout [8].

Prior-art design automation is focusing on either behavior-level, front-end synthesis or back-end synthesis, and elaborates on relatively simple analog cells [1]–[5], [7], [9]–[12].In the context of SAR ADCs, [11] proposed a MATLAB tool that allows statistical simulation of capacitor mismatches and parasitics in the DAC, and a behavioral SAR ADC model is proposed in [12] that allows fast simulation. However, while these tools support with sizing of the components and by reducing simulation time, they do not output a finalized schematic or layout of an entire SAR ADC, which is the purpose of this work. The design automation for complete ADCs is more cumbersome due to its complicated architecture, larger-scale circuitry and more sophisticated signal processing between analog and digital domain. Prior-art tries to synthesize a whole ADC using a single synthesis flow, chosen for a specific architecture. For a stochastic flash ADC [13] the traditional analog circuitry is replaced by pure digital gates so that the whole ADC can be integrated in the digital synthesis flow. However, the resolution and the performance are limited due to the stochastic flash architecture. For a $\Sigma\Delta$ ADC [8] a library-based approach and standard P&R for layout generation have been used. Although all the analog components are still manually designed, the power efficiency is yet behind state-of-the-art. [6] introduces a compile method for the Successive Approximation Register (SAR) ADC, but is limited to layout generation only. [14] introduces a systematic design method on schematic level for a SAR ADC, but large differences between simulation and measurement results are observed (>10×).

Alternatively, instead of synthesizing all the circuits using an identical approach, a design tool based on a hybrid automation approach is introduced in this work, allowing different automation approaches for the individual sub-circuits [15]. Both circuit and layout level are included. With the assistance of this tool, the designer can better balance automation and customization, and reduce the design time due to repetitive labor work while maintaining the ADC performance. This approach is elaborated for a SAR ADC, as the SAR ADC is popular for WSNs thanks to its excellent power efficiency [16]. The design automation approach is verified by generating two prototype ADCs. The measurements show good agreement with the simulation results.

The remainder of the paper is organized as follows: Section II introduces the proposed hybrid automation approach applied to the SAR ADC. The specification translation of the automation approach is shown in section III. The implementation details of the automation tool and the circuits are presented in section IV. The measurement results are shown in section V and the conclusions are drawn in section VI.

II. Hybrid design automation applied to a SAR ADC

A conventional N-bit SAR ADC usually includes 4 main blocks as shown Fig. 1: Sample&Hold (S&H), comparator, control logic and Digital-to-Analog Converter (DAC). In each SAR conversion, the S&H samples the analog input voltage and stores it on the capacitor arrays of the DAC. The DAC output will approximate the sampled input voltage in N comparisons through a Successive-Approximation algorithm. This is performed by the feedback loop which consists of the comparator and the SAR logic. In a synchronous SAR ADC, a high frequency clock (N$\times f_s$) is needed, but this can cause significant overhead in power consumption. Therefore, asynchronous operation is used in this work [16]. Thanks to the dynamic operation of all the ADC building blocks, the power consumption is inherently scalable with the sampling frequency. Further, the topologies of the S&H, comparator and DAC are fixed to a bootstrapped sampling switch [17], a dynamic comparator [18], and a top-sampling fully binary-weighted C-DAC with a monotonic switching scheme [19], [20], as it has been shown that such topologies can already cover a large performance range in terms of speed and resolution. Note that the monotonic switching scheme is not the most energy-efficient one, but thanks to the small capacitors used in the proposed DAC layout, the power consumption of the DAC is nonetheless relatively insignificant compared to that of the ADC (<10%). While the tool currently only implements the above topologies, it could be expanded with other topologies in a similar way.

As mentioned, to ensure the performance of the ADC, manual design is usually applied to all the 4 blocks, which is very time consuming (Table II). In particular, for the SAR logic, both the front-end and back-end design time are in the order of days, thus both need to reduce. For the DAC design, the design time for the back-end is much more than for the front-end, and therefore reducing the back-end design time improves the total design time more. For the S&H, the front-end design and the back-end design consume similar design

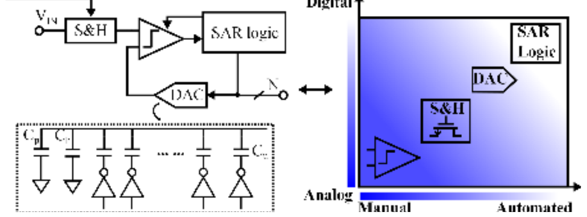
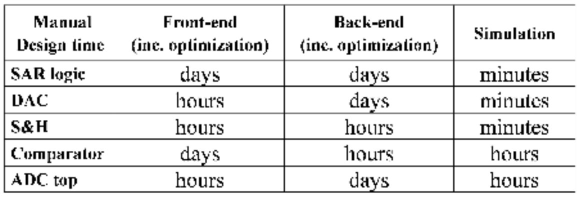


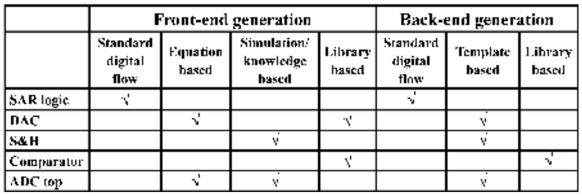
Fig. 1. The block diagram of a SAR ADC and illustration of its sub-blocks.

TABLE II
DESIGN TIME ANALYSIS OF A SAR ADC.



Manual Design time	Front-end (inc. optimization)	Back-end (inc. optimization)	Simulation
SAR logic	days	days	minutes
DAC	hours	days	minutes
S&H	hours	hours	minutes
Comparator	days	hours	hours
ADC top	hours	days	hours

TABLE III
SUMMARY OF THE PROPOSED HYBRID AUTOMATION METHODS.



	Front-end generation				Back-end generation		
	Standard digital flow	Equation based	Simulation/ knowledge based	Library based	Standard digital flow	Template based	Library based
SAR logic	√				√		
DAC		√		√		√	
S&H			√			√	
Comparator				√			√
ADC top		√	√			√	

time. For the comparator, most of the design time is spent in finding the proper sizes for the devices, and reducing the back-end design time helps less. For the ADC top-level, most design time is required for the back-end, as a large number of analog and digital connections has to be made in the layout.

To define the best design approach, the 4 basic blocks of a SAR ADC are analyzed first from two aspects: analog/digital-oriented and manual/automated (Fig. 1). Analog-oriented implies that the circuit is physically constrained (by noise, mismatch) and expertise is needed to properly optimize the circuit, which may require manual effort. Digital-oriented implies that the circuit is less physically constrained and can be abstracted and synthesized more easily without losing performance. In this work, different approaches are adopted for the front-end and back-end design, comprising the 4 basic blocks, and the top-level design of a SAR ADC as follows (Table III):

- **SAR logic**: Both the front-end and the back-end of the control logic can be synthesized using a standard synthesis flow. This has the advantage of flexibility in reconfiguring the number of bits, the operation speed, and the timing of the control signals. The design time is much shorter, up to a few hours, compared to that of manual design, and when redesigned, only a few key parameters need to be tuned. The redesign time, including front-end, back-end, and simulation, can be reduced to less than one hour. Besides, it features the possibility to integrate digital calibration logic in the future as will be discussed later. While the power efficiency of manually designed dynamic logic [16] is still better than synthesized logic, a standard synthesis flow is adopted here thanks to its advantages in design time and re-usability.
- **DAC**: The front-end design of the DAC capacitor array can be automated as follows. The DAC capacitance value is computed automatically according to the noise and matching requirements through an equation-based method as shown later. The DAC driver is simply an inverter, and therefore uses a library-based approach. Compared to the front-end, the DAC back-end design usually takes much longer due to the strict requirements, e.g., symmetry and optimization of parasitics at aF level. Fortunately, thanks to the regularity of the DAC layout, its design can be automated through template-based programming as shown later, which reduces the majority of the DAC design time.
- **S&H**: The S&H circuit, together with the ADC input capacitances, influences the ADC bandwidth. This effect can be easily checked by simulating the Spurious-Free-Dynamic-Range (SFDR). Considering the short simulation time, a simulation-based approach is introduced for the front-end design. In addition, to further optimize the S&H circuit while minimizing the manual effort, a knowledge-based approach is used by tuning only the critical devices as shown later. For the back-end, a template-based layout generation through programming is used to reduce the re-layout time during optimization or redesign.
- **Comparator**: The ADC performance is very sensitive to comparator non-idealities (e.g. offset error, noise, layout asymmetry), and therefore the comparator design needs more attention. In the elaborated design, the comparator design is library-based for both the front-end and the back-end.
- **ADC top level**: For the front-end, the ADC top level design uses equations to translate the top-level specifications into sub-circuit requirements at the beginning of the design, and a simulation-based approach to assure that the performance is met. For the back-end, thanks to the regular layout geometry of a SAR ADC, template-based layout generation is introduced, which reduces the layout time from days to minutes.

The discussed design steps are embedded in a design flow to automate a SAR ADC design in 6 steps (Fig. 2). The flow first takes the ADC specifications as input, then translates the input into parameters for the sub-blocks. After that, each sub-block is generated automatically using their corresponding method. Finally, the layout of all the sub-blocks will be assembled automatically and the performance of the generated ADC is verified through simulations using an extracted view. The design process will stop when the simulated ADC performance reaches the design target. Otherwise, further optimization is still possible by fine tuning the parameters of sub-blocks. In this work, the steps 1 to 4 in Fig. 2 are automated, while the simulation and performance check are manual.

III. SPECIFICATION TRANSLATION

The translation of the overall ADC specifications into the sub-block parameters is automated through an equation-based method, in step 2 of Fig. 2. The equations are divided into two categories: noise-related and speed-related, as shown in Fig. 3. The input resolution N determines Signal-to-Quantization-Noise-Ratio (SQNR), which leads to the ADC quantization noise floor $\mathrm{P_{qn}}$ (Fig. 3), based on a provided input signal range $\mathrm{V_{fs}}$. The noise requirements of the sub-blocks (the sampling noise $\mathrm{P_{sh}}$ and the comparator noise $\mathrm{P_{cmp}}$) are computed based on this quantization noise as shown in Fig. 3. As a result, the Signal-to-Noise-Ratio (SNR) degradation due to the extra

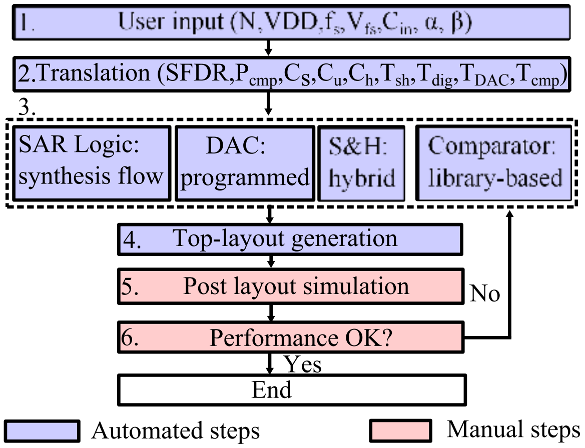

1. User input (N,VDD,f_s,V_{fs},C_{in}, α, β)
2.Translation (SFDR,P_{cmp},C_S,C_u,C_h,T_{sh},T_{dig},T_{DAC},T_{cmp})
3.
SAR Logic: synthesis flow
DAC: programmed
S&H: hybrid
Comparator: library-based
4. Top-layout generation
5. Post layout simulation
No
6. Performance OK?
Yes
End
Automated steps
Manual steps

Fig. 2. Design flow of the proposed method.

circuit noise is $10\log_{10}(1+\alpha+\beta)$ dB. For instance, with $\alpha=\beta=\frac{1}{2}$, 3dB SNR degradation (equivalently 0.5bit ENOB loss) will occur due to the noise from sampling and comparator, but other choices are possible dependent on the user preference.

After that, the derived noise requirements are used to compute the circuit parameters for each sub-block. In particular, the total input capacitance C_s of the ADC can be computed from the sampling noise P_{sh} requirement. At the same time, according to the equations to calculate the input signal range of the ADC ($2VDD{\cdot}C_{DAC}/C_s$), the DAC capacitance C_{DAC} and the attenuation capacitance C_h are computed. From this, assuming a monotonic DAC switching scheme, the minimum unit capacitance C_u of the DAC can be computed based on the noise constraint. The mismatch constraint of the capacitors is not considered as it is expected to be tackled by mismatch calibration as in [21] as discussed later. For the S&H, the required SFDR should be M dB higher than the targeted SNR, where M is typically set to 10dB, but can be overruled by the user. For the comparator, the calculated noise requirement P_{cmp} will decide which comparator is selected from the library.

Meanwhile, the operation speed of the comparator, DAC and logic is related to the overall ADC operation speed f_s (Fig. 3). The total time of the S&H, the comparator, the DAC and the logic should be no more than the period of the sampling clock T_s. By default, half of the period is used for the S&H to track the analog input signal, and the other half is used for the conversion, which consists of N× operation of the comparator, the DAC and the SAR logic. In this work, the logic and DAC are first designed and simulated, and the remaining time is for the comparator.

IV. Block Implementations and Top-Level Layout

As discussed, the requirements of the sub-blocks are derived from the equations in Fig. 3, and will be used for the design of each sub-block respectively and for the top-level layout generation.

A. Fully-synthesized control logic

The digital control logic of the asynchronous SAR ADC performs the binary-search algorithm and also generates an internal clock for the asynchronous ADC. In particular, a monotonic switching scheme [19] is implemented to improve the power efficiency of the ADC. Thanks to the well-developed digital synthesis flow, most of the design effort is now taken over by the digital design tools. For example, when redesigning for a different speed, the designer only has to change the timing constraint parameter of the synthesis flow. The delay of the digital control logic will be optimized automatically by the synthesis flow, to meet the required operation speed. This is more convenient and faster compared to the iterative manual optimization method.

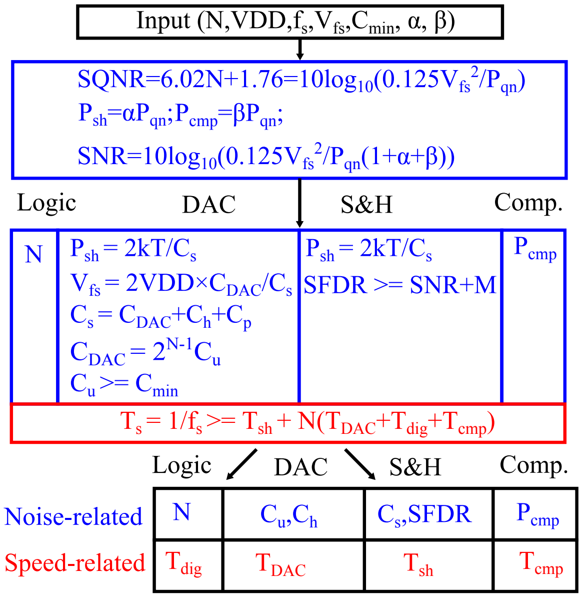

Input (N,VDD,f_s,V_{fs},C_{min}, α, β)
$SQNR=6.02N+1.76=10\log_{10}(0.125V_{fs}^2/P_{qn})$
$P_{sh}=\alpha P_{qn}$;$P_{cmp}=\beta P_{qn}$;
$SNR=10\log_{10}(0.125V_{fs}^2/P_{qn}(1+\alpha+\beta))$
Logic
DAC
S&H
Comp.
N
$P_{sh}=2kT/C_s$
$V_{fs}=2VDD\times C_{DAC}/C_s$
$C_s=C_{DAC}+C_h+C_p$
$C_{DAC}=2^{N-1}C_u$
$C_u>=C_{min}$
$P_{sh}=2kT/C_s$
SFDR >= SNR+M
P_{cmp}
$T_s=1/f_s>=T_{sh}+N(T_{DAC}+T_{dig}+T_{cmp})$
Logic
DAC
S&H
Comp.
Noise-related
N
C_u,C_h
C_s,SFDR
P_{cmp}
Speed-related
T_{dig}
T_{DAC}
T_{sh}
T_{cmp}

Fig. 3. Translating from ADC specifications to block parameters.

B. Hybrid DAC design

The charge-redistribution DAC is often chosen for SAR ADCs thanks to its good power efficiency. It consists of DAC drivers and DAC capacitors, which are designed automatically through a script-based automation flow combining a mix of equation-based, library-based, and template-based design (Fig. 4). An equation-based method is firstly adopted to size the unit capacitor C_u based on the noise constraint as shown in Fig. 3. On the other hand, when the ADC resolution goes beyond 10b, C_u tends to be limited by the capacitor mismatch and thus has to be sized relatively large, degrading power efficiency. Alternatively, to save power, C_u can be sized to just meet the kT/C noise, and calibration can be employed to compensate the capacitor mismatch errors [21]. Calibration is not yet present in the current tool, but can be easily integrated into the digital circuit synthesis. In case mismatch is dominant while calibration is not used, the sizing process of C_u becomes more difficult. In that case, an assistance tool such as [11] could be inserted to determine appropriate sizing.

It can be calculated that for medium and low resolution ADCs with a rail-to-rail input range, the noise-constrained C_u is unpractically small. For example, C_u for an 8bit ADC can be as small as 40aF based on noise requirements, which will

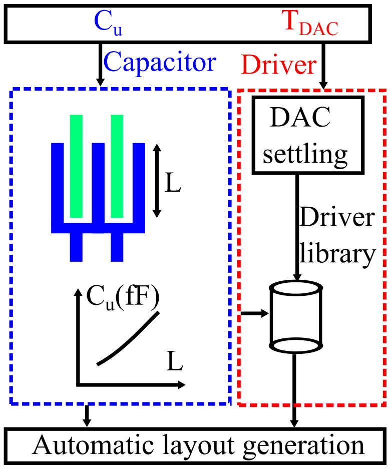


Fig. 4. The DAC design automation flow.

be much smaller than the parasitics. This is undesired since it will reduce the input signal range and make the input signal range more vulnerable to parasitics. To avoid this, the designer can optionally set a lower boundary (C_{min}) for C_u, e.g., 600aF. After selecting C_u, the DAC driver needs to guarantee that the DAC output can settle in time T_{DAC}. The selection is made from a library of drivers with various strengths, matching the selected C_u and the required operation speed f_s. The DAC driver inputs are the load of the digital control logic. When using larger size DAC drivers, the loading of the corresponding digital control bit should be adapted for the synthesis.

The DAC layout is very structured with a repetitive geometry, and therefore can be automated efficiently to save layout design time. In this work, a programmable cell (*Pcell*) is used for the DAC design automation and its layout generation. By programming the key parameters of the DAC, this tool enables users to provide the specification, automate the device sizing and generate the layout accordingly. C_u is a customized finger capacitor as shown in Fig. 4. This allows the employment of a small unit capacitance (sub-fF), and to reduce the DAC area and power consumption at the same time. In addition, the unit capacitance can be approximated as linearly proportional to the overlapping length L between the top plate and bottom plate of the capacitor. Therefore, the unit capacitance can be precisely controlled by simply programming the unit capacitor length L dependent on the required value of C_u (Fig. 4). As can be seen in Fig. 5, the unit capacitor value C_u changes the length L, which is in vertical direction, while the resolution N changes the total number of capacitors placed besides each other in the horizontal direction. As such, L and N can be independently modified without significantly changing the overall DAC floorplan. The DAC layout is automatically generated from the *Pcell*, which is described in *SKILL* language.

C. Hybrid S&H design

For a given DAC capacitance, the on-resistance R_{on} of the sampling switch determines the bandwidth of the S&H. In modern CMOS technologies, the supply voltage drops below 1V, making it difficult to reach the desired bandwidth (∼MHz) for the ADCs in WSNs if directly-controlled switches are used. Therefore, a bootstrapped circuit [17] is used in this work as shown in Fig. 6. As mentioned, a simulation based approach is introduced for the S&H circuit thanks to the short simulation time for SFDR. However, noting that there are multiple devices in the S&H circuit (Fig. 6), the optimization time for the sizing would be too long to be acceptable.

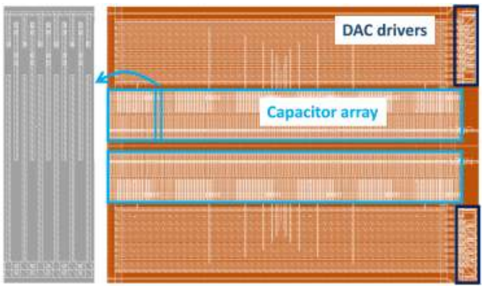


Fig. 5. A generated DAC layout example for an 8 bit ADC.

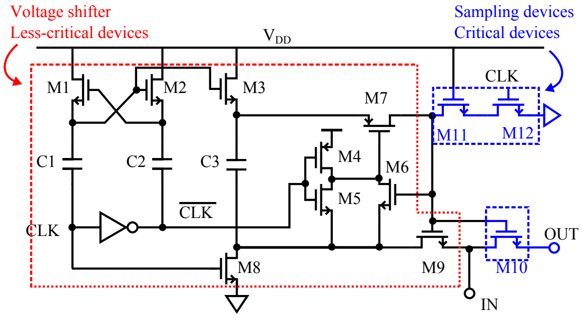


Fig. 6. The S&H circuit.

To reduce the optimization time while still maintaining good performance for the S&H, a knowledge-based approach is added as follows. The circuit can be roughly divided into two parts: the devices for sampling (performance critical) and for the voltage shifter (performance less-critical). A voltage multiplier (M1∼M9) performs the function of boosting the gate voltage of the sampling switch M12. These devices are relatively less critical as long as the function of voltage boosting can be achieved. Therefore, the size of the non-critical devices is fixed for all designs in this automation tool. M10∼M12 are critical for the sample and hold performance, and therefore need to be carefully treated.

Therefore, a hybrid design approach, consisting of a simulation and knowledge based approach, is proposed for the front-end design of the S&H circuit (Fig. 7). One S&H is implemented at first (including layout) and re-used as a template when redesigned for a different specification. When re-designed, the less critical devices are fixed, while the critical devices are tuned for optimization. As a result, the total number of iterations needed to reach the optimum point is significantly reduced, enabling the employment of a simulation-based look-up-table (LUT) approach for the S&H circuit. The LUT can be constructed based on the simulation results for various device sizes as shown in Fig. 8. In this way, in a next design, an approximately optimized device size can be

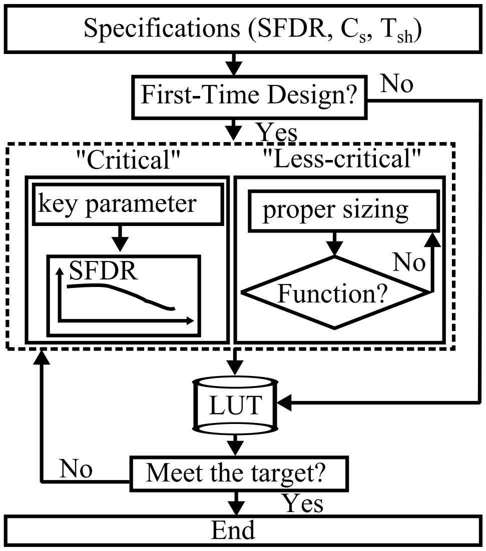
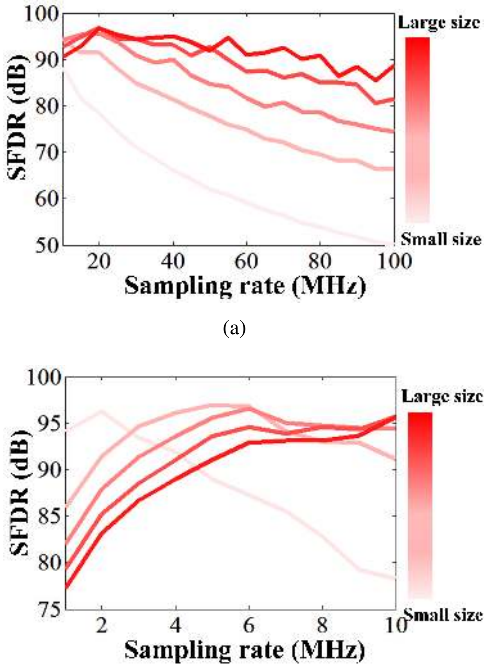


Fig. 7. The S&H design automation method.

directly selected from the LUT. It is crucial to choose the device dimension properly from the LUT according to the required SFDR and the ADC sampling rate. Larger devices do not always result in better SFDR, but can degrade the SFDR for low speed ADCs due to device leakage. Moreover, further optimization is still possible by further fine tuning the critical devices. To save optimization time, one parameter, which is the width of the devices, is used for sizing the three critical devices since they are usually tuned in the same direction. Furthermore, it is worthwhile to mention that the ADC input capacitance C_s influences the SFDR of the S&H circuit. The bandwidth of the S&H is determined by an RC constant, where R is determined by the size of the sampling transistor M10 and C is C_s. This impact of C_s is accommodated indirectly by scaling the R in an opposite direction, to maintain a constant RC value. For the back-end design, considering that most of the devices are fixed and only three transistors are programmed, a template-based layout generation through SKILL is introduced for the S&H, similarly as for the DAC, which further reduces the layout redesign time.

D. Library-based comparator design

In wireless sensor nodes, the comparator performance is crucial for SAR ADCs, as it can either dominate the overall ADC power consumption, or degrade the ADC Effective-Number-Of-Bits (ENOB). Therefore, even for an experienced ADC designer, it usually takes days to optimize the speed, noise and power efficiency of the comparator. However, it is not straightforward to correlate the comparator performance with the device sizing due to its relatively complex circuitry, non-linear operation and sensitivity to layout imperfections. Therefore, a library-based approach is introduced, in which comparators with different performance combinations can be selected by the computer-aided design tool when users input the specifications. The comparators in the library will have a two-dimensional index: input-referred noise and comparison time. The probability of metastability can be controlled by assigning a more conservative comparison time to build in some extra buffer time. According to these two requirements, a proper device in the table will be selected (Table IV). In this work, a dynamic comparator is used thanks to its efficiency and frequency scalability [16], [18].



(a)

(b)

Fig. 8. Simulated SFDR of S&H with various device sizes for relatively high (a) and low (b) sampling frequency respectively.

TABLE IV
ILLUSTRATION OF LIBRARY BASED METHOD FOR COMPARATOR.

P_{cmp} / T_{cmp}	$<=P_0$	$(P_0, P_1]$	$(P_1, P_2]$	$(P_2, P_3]$	... ...
$(0, T_0]$	CMP1.1	CMP1.2	CMP1.3	CMP1.4	... ...
$(T_0, T_1]$	CMP2.1	CMP2.2	CMP2.3	CMP2.4	... ...
$(T_1, T_2]$	CMP3.1	CMP3.2	CMP3.3	CMP3.4	... .
$(T_2, T_3]$	CMP4.1	CMP4.2	CMP4.3	CMP4.4	... .
$(T_3, T_4]$	CMP5.1	CMP5.2	CMP5.3	CMP5.4	... .

E. Top-layout generation

The traditional Place&Route (P&R) in the standard digital flow minimizes the manual layout effort for digital circuits by automatic P&R [8]. However, this can cause problems for analog circuits, which are sensitive to layout non-idealities (e.g., parasitics, asymmetries, and IR drops), and thus may degrade the ADC performance. Alternatively, template-based layout generation is introduced [22]. This layout generation is used for well-understood designs with sufficient regularity or geometric templates. It only adapts the size for devices that are critical to optimization, while the remaining devices are relatively fixed. The interconnections between the devices are adapted automatically.

The aforementioned tool for the design of the sub-blocks is developed using *SKILL* language and the *Pcell* tool in Ca-

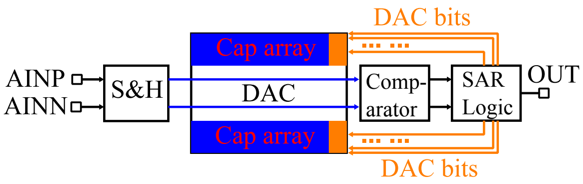


Fig. 9. The template of the ADC top layout.

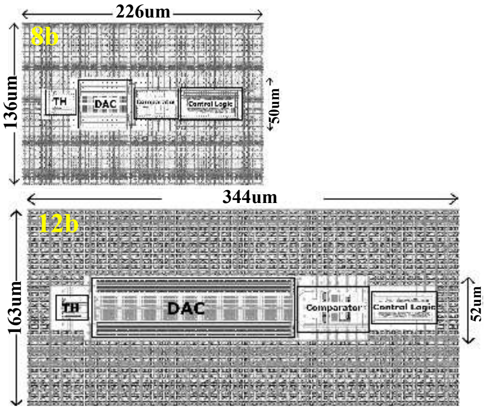


Fig. 10. The generated ADC layouts using the proposed flow.

dence. The layout generations of the ADC top are automated using a geometrical template-based method. This method allocates each sub-block according to a template, and connects the pins between each block through scripting (Fig. 9). The geometrical coordinates of all sub-blocks are parameterized and the pin locations of each block are automatically adapted accordingly. This greatly saves design time for layout, and as a result, it takes only a few seconds for the program to integrate into a DRC and LVS clean compact layout (Fig. 10). After that, simulations based on an extracted view can be executed to verify the performance of the ADC. Note that it is still possible to fine tune each sub-block individually to optimize the overall ADC performance using the implemented design flow.

V. Verification with Prototypes

To verify the proposed design flow, two prototype chips with different specifications for speed and resolution have been implemented in 40nm CMOS: an 8b 32MS/s and a 12b 1MS/s SAR ADC (Fig. 11). The compact layout leads to a small core area of the two ADCs (Fig. 10) of 0.011mm^2 and 0.016mm^2 respectively. Including decoupling capacitors, they occupy 0.031mm^2 and 0.056mm^2.

The 8b ADC consumes 187μW at 32MSps and 1V supply, and achieves 47.4dB SNDR up to Nyquist frequency (Fig. 12). This leads to a 30.7fJ/conversion-step Figure-of-Merit (FoM) as defined in equation 1, which is comparable to the simulated 25.1fJ/conversion-step, considering that simulation results did not include non-idealities (e.g., noise and mismatches).

$$\mathrm{FoM} = \frac{\mathrm{Power}}{2^{\mathrm{ENOB}}\min(\mathrm{f_s}, 2\mathrm{ERBW})} \tag{1}$$

The 12b ADC consumes 61.1μW at 1MSps and 1V, which is close to the simulation results. Considering that the DAC capacitors are sized for kT/C noise instead of mismatch, large INL/DNL errors are measured as shown in Fig. 13, thus limiting the ADC performance. However, this error is assumed to be solved by DAC mismatch calibration [21]. The measured input-referred noise of the 12b ADC is 0.47LSB$_{\mathrm{rms}}$. This indicates that the noise-limited ADC SNDR is above 11b once the DAC mismatch induced error is corrected.

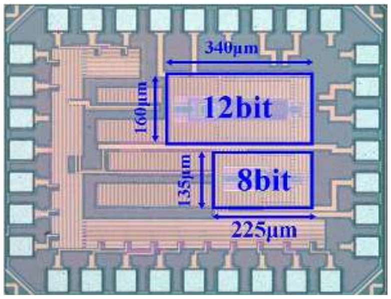


Fig. 11. Die photo of the prototype ADCs in 40nm CMOS.

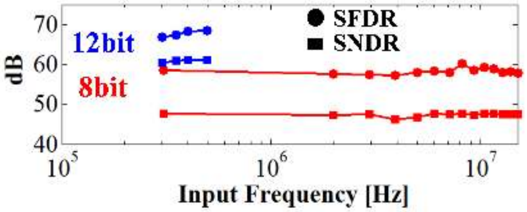


Fig. 12. The dynamic performance of the two ADCs.

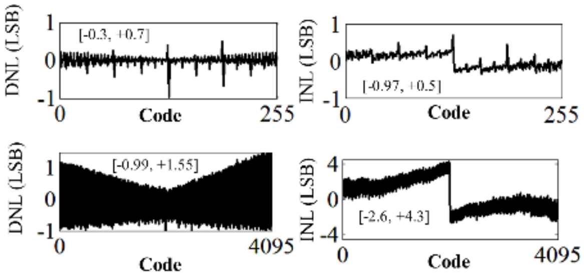


Fig. 13. The measured INL/DNL of the two ADCs.

In addition, both ADCs achieve stable SNDR/SFDR up to the Nyquist frequency (Fig. 12 and 14), indicating a sufficient performance for the S&H circuits. Overall, the measured power consumption and speed match with the simulated numbers. We should note that the simulated numbers are not estimated by the proposed tool, but by post-layout simulation of the layout that is generated by the tool. The measured FoM for the 8b ADC is close to expectation (Table VI). For the 12b ADC, the FoM correspondence is worse compared to the 8b ADC due to the mismatch induced errors. However, the correspondence is better than prior art.

The automated design flow successfully reduces the total ADC design time down to minutes level (Table V), after initial construction of the libraries, templates and code. In particular, both the front-end and back-end design time of the SAR logic are reduced significantly from days to minutes level. The time to prepare the digital design flow is at minutes level. For

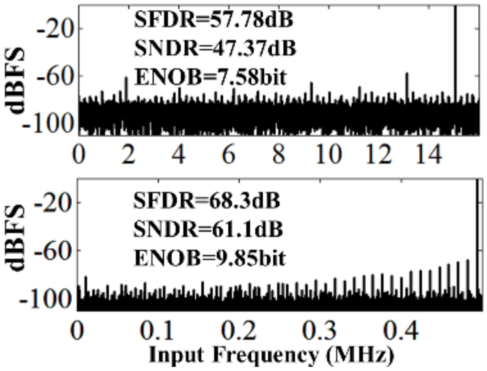


Fig. 14. Measured spectrum of the two ADCs with a near-Nyquist tone.

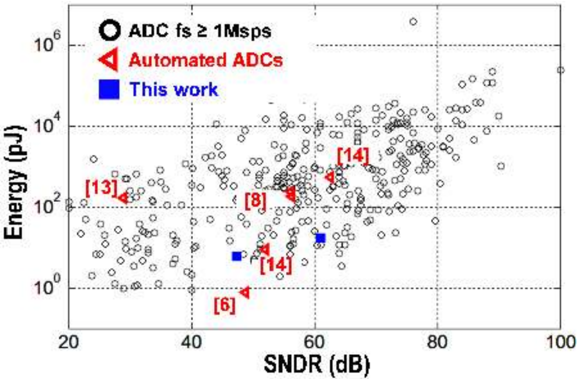


Fig. 15. Benchmark with the state-of-the-art.

TABLE V
ADC DESIGN TIME OF THE PROPOSED FLOW.

Manual Design time	Front-end (inc. optimization)	Back-end (inc. optimization)
SAR logic	days → minutes	days → minutes
DAC	minutes → seconds	days → seconds
S&H	hours → minutes	hours → seconds
Comparator*	days → seconds	hours → seconds
ADC top	minutes → seconds	days → seconds

* Design time for the library is days

TABLE VI
PERFORMANCE SUMMARY OF THE TWO ADCS.

	[13]	[14]				[6]	This work			
Experiment	Meas.	Meas.	Sim.	Meas.	Sim.	Meas.	Meas.	Sim.	Meas.	Sim.
Automation level	Schematic Layout	Schematic				Layout	Schematic Layout			
Architecture	Stochastic	SAR				SAR	SAR			
Technology (nm)	90	180		90		28	40			
Resolution (bit)	-	12		10		9	8		12	
Speed (MSps)	210	0.6		50		20	32	32	1	1
Supply voltage (V)	1.2	3.3		1		0.69	1	1	1	1
INL (LSB)	-	2.79	0.1	1.41	0.06	-	0.7	-	4.3	-
DNL (LSB)	-	5.24	0.22	2.81	0.1	-	0.97	-	1.55	-
Nyquist SNDR (dB)	29	62.7	73.9	52.1	61.9	48.84	47.37	49.3	61.1	71.6
Power (uW)	34800	328	53.5	440	155	15.87	187	192	16.7	16.8
FoM (fJ/c.s.)	7943	500	22	26.7	3.1	3.51	30.7	25.1	18.1	5.4
Meas. vs. Sim. ratio	-	22.7		8.6		-	1.2		3.4	

the DAC, the design time is reduced down to minutes level thank to the automated layout generation. For the S&H circuit, it takes around 3 iterations for the front-end design through the LUT and fine tuning, and the layout can be generated automatically. The layout template preparation time for the DAC and the S&H is a few hours. The comparator library preparation time is still in days since it needs to be manually designed, but this has to be done only once. The ADC top integration is reduced from days to seconds. Overall, thanks to the automated generation of all the sub-circuits, the design time needed for one iteration is greatly reduced. In this way, it speeds up the design time considerably, even for a designer with less expertise. At the same time, the performance of the ADC is maintained (Fig. 15), showing a good balance between design time and ADC performance compared to the state-of-the-art [23]. Compared to other synthesized ADC approaches (Table VI), this work is one of the few to automate both schematic and layout design, while it achieves good power-efficiencies and the best matching between predicted performance and measured results.

At present, the proposed tool aims for design assistance of SAR ADCs at medium speed and resolution. In the future, capacitor mismatch calibration as in [21] can be integrated so that higher resolution ADCs can be generated. For high-speed ADCs, probably more attention is required to obtain a high-speed layout, the logic may need to be custom designed, and other circuit topologies might be required. Beyond SAR ADCs, the hybrid automation method could also be applied to other ADCs architectures, but it is most useful to relatively digital-intensive architectures. The most time consuming part would be to build the layout templates, and the component library where applicable. The more complex analog blocks are required, the less likely it is to benefit from the proposed tool.

VI. CONCLUSIONS

In this work, a circuit-design-driven tool with a hybrid automation approach for SAR ADCs is proposed. Compared to prior-art, instead of simplifying/modifying the circuit architecture to enable synthesis, our hybrid performance-selective approach allows the employment of appropriate circuits for performance, and automation for design-time reduction. In this way, the design time is reduced without compromising performance, meanwhile still keeping the possibility to control performance-critical devices through programming, enabling a balance between automation and customization. The implemented prototypes achieve a performance that is sufficiently competitive with full-manual designs, while strongly reducing design time, and while having accurate control over the performance goal.

REFERENCES

[1] M. G. Degrauwe *et al.*, “IDAC: An interactive design tool for analog CMOS circuits,” *IEEE J. Solid-State Circuits*, vol. 22, no. 6, pp. 1106–1116, 1987.

[2] G. G. Gielen *et al.*, “Analog circuit design optimization based on symbolic simulation and simulated annealing,” *IEEE J. Solid-State Circuits*, vol. 25, no. 3, pp. 707–713, 1990.

[3] H. Y. Koh *et al.*, “OPASYN: A compiler for CMOS operational amplifiers,” *Transactions Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 113–125, 1990.

[4] E. S. Ochotta *et al.*, “ASTRX/OBLX: Tools for Rapid Synthesis of High-performance Analog Circuits,” in *Proceedings of the Design Automation Conference*, 1994, pp. 24–30.

[5] G. Alpaydin *et al.*, “An Evolutionary Approach to Automatic Synthesis of High-performance Analog Integrated Circuits,” *Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 240–252, 2003.

[6] C. Wulff *et al.*, “A Compiled 3.5fJ/conv.step 9b 20MS/s SAR ADC for Wireless Applications in 28nm FDSOI,” in *ESSCIRC*, 2016, pp. 177–180.

[7] R. Martins *et al.*, “LAYGEN II - Automatic Layout Generation of Analog Integrated Circuits,” *Transactions Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 11, pp. 1641–1654, 2013.

[8] A. Waters *et al.*, “A Fully Automated Verilog-to-layout Synthesized ADC Demonstrating 56dB-SNDR with 2MHz-BW,” in *IEEE Solid-State Circuits Conference (A-SSCC)*, 2015, pp. 1 – 4.

[9] H. Graeb *et al.*, “Analog Layout Synthesis - Recent Advances in Topological Approaches,” in *Design, Automation and Test in Europe (DATE)*, 2009, pp. 274 – 279.

[10] E. Yilmaz and G. Dundar, “Analog Layout Generator for CMOS Circuits,” *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 32–45, 2009.

[11] S. Brenna *et al.*, “A Tool for the Assisted Design of Charge Redistribution SAR ADCs,” in *Design, Automation and Test in Europe (DATE)*, 2015, pp. 1265 – 1268.

[12] P. Nandi *et al.*, “A Novel Approach to Design SAR-ADC: Design Partitioning Method,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 346–356, 2016.

[13] S. Weaver *et al.*, “Digitally Synthesized Stochastic Flash ADC Using Only Standard Digital Cells,” *Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 1, pp. 84–91, 2014.

[14] C.-P. Huang *et al.*, “A Systematic Design Methodology of Asynchronous SAR ADCs,” *IEEE Transactions on very large scale integration systems*, vol. 24, no. 5, pp. 1835 – 1848, May 2016.

[15] M. Ding *et al.*, “A Circuit-Design-Driven Tool with a Hybrid Automation Approach for SAR ADCs in IoT,” in *Design, Automation and Test in Europe (DATE)*, 2018, pp. 672 – 675.

[16] P. J. A. Harpe *et al.*, “A 26μW 8 bit 10 MS/s Asynchronous SAR ADC for Low Energy Radios,” *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1585 – 1595, Jul. 2011.

[17] A. M. Abo *et al.*, “A 1.5-V, 10-bit, 14.3-MS/s CMOS Pipeline Analog-to-Digital Converter,” *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 599–606, 1999.

[18] M. van Elzakker *et al.*, “A 10-bit Charge-Redistribution ADC Consuming 1.9 μW at 1 MS/s,” *IEEE J. Solid-State Circuits*, vol. 45, no. 5, pp. 1007 – 1015, Apr. 2010.

[19] C.-C. Liu *et al.*, “A 10-bit 50-MS/s SAR ADC with a Monotonic Capacitor Switching Procedure,” *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 731–740, 2010.

[20] P. Harpe *et al.*, “A 7-to-10b 0-to-4MS/s flexible SAR ADC with 6.5-to-16fJ/conversion-step,” in *IEEE ISSCC*, 2012, pp. 472–474.

[21] M. Ding *et al.*, “A 46 μW 13 b 6.4 MS/s SAR ADC With Background Mismatch and Offset Calibration,” *IEEE J. Solid-State Circuits*, vol. 52, no. 2, pp. 423 – 432, Jul. 2017.

[22] J. Kuhn, “Analog module generators for silicon compilation,” *VLSI Systems Design*, vol. 8, no. 5, p. 74, 1987.

[23] B. Murmann, “ADC Performance Survey 1997-2017,” in *[Online]. Available:http://web.standford.edu/ murmann/adcsurvey.html.*

Ming Ding received the B.E. degree from Huazhong University of Science and Technology in 2009, China, and the M.Sc. degrees from the Eindhoven University of Technology in 2011, The Netherlands. In 2011, he started as researcher at Holst Centre / imec, The Netherlands. Since then, he has been working on data converter and ultra-low-power wireless research and design.

Pieter Harpe (SM15) received the M.Sc. and Ph.D. degrees from the Eindhoven University of Technology, The Netherlands, in 2004 and 2010, respectively. In 2008, he started as researcher at Holst Centre/imec, The Netherlands. Since then, he has been working on ultra low-power wireless transceivers, with a main focus on ADC research and design. In April 2011, he joined Eindhoven University of Technology where he is currently an Associate Professor on low-power mixed-signal circuits. Dr. Harpe is co-organizer of the yearly workshop on Advances in Analog Circuit Design (AACD) and analog subcommittee chair for the ESSCIRC conference. He also served as ISSCC ITPC member and IEEE SSCS Distinguished Lecturer and is recipient of the ISSCC 2015 Distinguished Technical Paper Award.

Guibin Chen received the B.S. degree from Central South University (CSU), Changsha, China, and the M.S. degree (Cum Laude.) from Eindhoven University of Technology (TU/e), Eindhoven, the Netherlands, in 2013 and 2015, respectively. From 2014 to 2015, he was with Holst Centre/imec as an internship, working on ADC design automation. In 2016, he was with Nanyang Technological University (NTU) as a researcher. From September 2016, he joined in Tencent AI Lab, Shenzhen, China.

Benjamin Busze is a researcher with experience in digital signal processing and digital circuit design. He received his Dipl.-Ing. degree in Electrical Engineering and Information technology from the RWTH-Aachen University (Germany) in 2008. From 2008 to 2018, he was with Holst Centre/imec, working on digital baseband design for wireless systems. From 2018, he is with Dialog Semiconductor as a senior digital designer.

Yao-Hong Liu received his Ph.D. degree from National Taiwan University, Taiwan, in 2009. Since 2010, he joined imec, the Netherlands. His current position is Principal Membership of Technical Staff, and he is leading the development of the ultra-low power (ULP) RFIC design. His research focuses are energy-efficient wireless transceivers and RF sensing for IoT. He currently serves as a technical program committee of IEEE ISSCC and RFIC symposium.

Christian Bachmann is Program Manager of imec’s ULP Wireless Systems program. He joined imec in 2011, working on ultra-low-power wireless communication systems, digital baseband processing, and hardware/software co-design. In his previous work he has covered various wireless communication solutions for 802.11ah WiFi, Bluetooth LE, 802.15.4 (Zigbee), ultra-wideband impulse radio and others. Prior to joining imec, he has been researching hardware-accelerated power estimation for VLSI systems both with Infineon Technologies and Graz University of Technology in Graz, Austria, receiving his PhD in Electrical Engineering in 2011.

Kathleen Philips is a program director at imec for the IoT. Dr. Philips holds a PhD in Electrical engineering and has 20 years of experience in the domain of low power mixed-signal, RF and integrated system design. She started her career at Philips Research, The Netherlands, working an analog and mixed-signal circuits. She joined the Holst Centre/imec in 2007, where she is currently leading a program on ultra-low power wireless, processing and sensing systems. This is part of an open innovation collaboration, together with local and international industry. The research targets applications in the area of infrastructure and person-centric IoT systems, with a focus on ultra-low power communication and sensing.

Arthur van Roermund was born in Delft, The Netherlands in 1951. He received the M.Sc. degree in EE in 1975 from Delft University of Technology and the Ph.D. degree in Applied Sciences from K.U.Leuven, Belgium, in 1987. From 1975 to 1992 he was with Philips Re-search Laboratories in Eindhoven. From 1992 to 1999 he has been a full professor at the EE department of Delft University of Technology. October 1999 he joined Eindhoven University of Technology, where he was chairman of the Mixed-signal Microelectronics group until Jan. 2017, and director of research of the TU/e-EE Department from 2002 to 2012. He co-organised the yearly workshop on Advanced Analog Circuit Design (AACD) from 2001 to 2012. In 2004 he achieved the Simon Stevin Meester award for his scientific and technological achievements. He was member of three international assessment panels: Politecnico di Milano, Italy, 2007; Aalto University, Finland, 2009; KTH, Stockholm, Sweden, 2012. He published over 700 articles and 30 books, and he is co-editor of another 12 books. Since Sept 1st 2017 he is emeritus professor.