A Systematic Design Methodology of Asynchronous SAR ADCs

Chun-Po Huang, Jai-Ming Lin, Ya-Ting Shyu, and Soon-Jyh Chang, *Member, IEEE*

***Abstract*— Successive approximation register (SAR) analog-to-digital converters (ADCs) are widely used in biomedical and portable/wearable electronic systems due to their excellent power efficiency. However, both the design and the optimization of high-performance SAR ADCs are time consuming, even for well-experienced circuit designers. For system designers, it is also hard to quickly evaluate the feasibility of a given specification in a process node. This paper presents a systematic sizing procedure for asynchronous SAR ADCs based on design considerations. A sizing tool based on the proposed design procedure is also implemented, the sizing results of which are highly competitive in comparison with other state-of-the-art manual works. Moreover, the sizing time is relatively short due to the efficient and effective search algorithms employed. In addition to the simulation results, two silicon proofs with different specifications and process nodes are provided to demonstrate the feasibility of this design methodology.**

***Index Terms*— Asynchronous successive approximation register (SAR) analog-to-digital converter (ADC), design automation, design methodology, optimization, sizing tool.**

I. Introduction

ANALOG-TO-DIGITAL converters (ADCs) are the key building blocks between the analog and digital domains. Because the performance of a whole system is often constrained by ADCs, they play an important role in many mixed-signal systems. Due to the prevailing applications of green electronics, consumer electronics, and biomedical electronics, low power consumption has become a critical and essential target for circuit design. Among the various architectures of ADCs, the successive approximation register (SAR) ADC has attracted more attention in recent years due to its excellent power efficiency and suitability for medium-to-high-speed applications. Numerous related papers have been presented in major integrated circuit design conferences [1].

Designing an SAR ADC relies on well-experienced circuit designers, requiring much time and effort even for senior circuit designers. Confounding this issue, it is also difficult for system designers to quickly evaluate the feasibility of given specifications and process nodes. With the increasing complexity of application systems and advancing process technologies, there is great demand for a systematic design procedure and design automation tool for SAR ADCs. Such procedures and tools not only shorten the time-to-market, but also raise the competitiveness of the created products.

Design automation for analog and mixed-signal (AMS) circuits can be primarily divided into three stages, namely, the topology selection, the device sizing, and the layout generation stages. The topology selection stage finds a suitable circuit architecture according to the given specifications and process technology. The device sizing stage determines the proper transistor sizes for better performance and lower chip cost. In addition, issues, such as feasibility of layout and manufacturing, must be considered at this stage. Last, the layout generation stage creates a robust and compact layout, such that the design specifications can be satisfied. This paper focuses on the device sizing stage, since it is usually the most time-consuming portion of the three stages.

The approaches for automatically sizing a device can be classified into two categories. One is the knowledge-based approach and the other is the optimization-based approach [2]. The knowledge-based approach determines a design plan according to circuit designers' experience. In contrast to the optimization-based approach, the knowledge-based approach has the advantage of short execution time, especially for complex AMS circuits. However, to keep pace with technological evolution, the design plan must be maintained. The optimization-based approach can be further divided into equation-based and simulation-based methods. Although the equation-based method has a faster execution speed, precise equations are generally difficult to obtain, and this imprecision may be in a decrease in reliability. Hence, the equation-based method is usually applied to obtain the initial solution. In comparison, the simulation-based method optimizes the results by involving a circuit simulator for precise evaluation. However, despite the device model being reliable, due to a lack of proper constraints and search strategies, the method may not converge to a good result, leading to long execution time. In this paper, we combine the advantages of these methods to simultaneously enhance the speed and the accuracy. The equation-based method is applied for identifying the initial condition; then, the simulation-based method is used to generate accurate sizing results. Finally, the knowledge-based method is employed for enhancing the reliability and reducing the number of simulations.

A. Preliminary

This section first introduces the background of SAR ADCs, including the basic operations of SAR ADCs and the differences between synchronous control schemes and

Manuscript received June 18, 2015; revised September 11, 2015; accepted October 18, 2015. Date of publication November 20, 2015; date of current version April 19, 2016. This work was supported in part by Himax Technologies Inc., and in part by the National Science Council of Taiwan under Grant NSC-102-2221-E-006-263-MY3.

The authors are with the Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan (e-mail: gppo0721@gmail.com; jmlin@ee.ncku.edu.tw; ytshyu@gmail.com; soon@mail.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2015.2494063

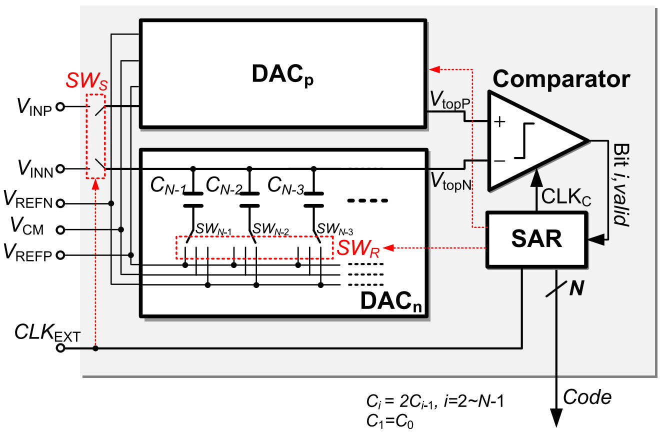


Fig. 1. Typical N-bit asynchronous SAR ADC.

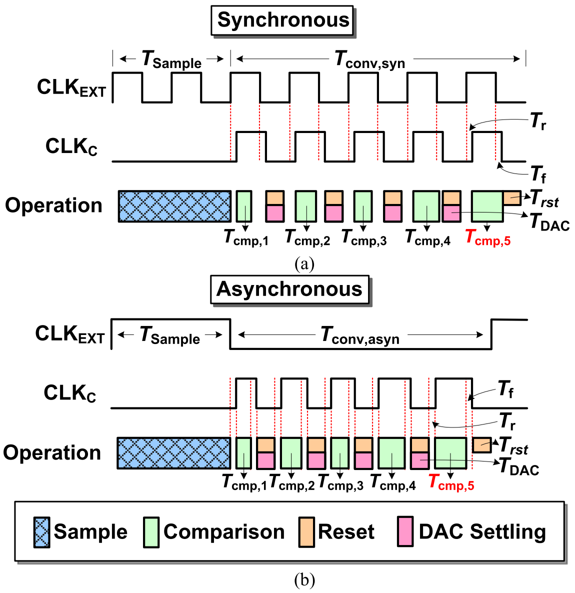


Fig. 2. Timing diagrams of (a) synchronous and (b) asynchronous control schemes.

asynchronous ones [3]. Because our sizing methodology combines the knowledge-based and optimization-based approaches, understanding circuit design is helpful in realizing the methodology.

The typical architecture of an SAR ADC is shown in Fig. 1, which consists of a comparator, a differential digital-to-analog converter (DAC) with an embedded sample-and-hold (S/H) circuit, and an SAR control logic circuit.

1) Basic Operations: As shown in Fig. 2, the analog-to-digital conversion of an N-bit SAR ADC can be separated into a signal sampling phase (blue region) and N bit-conversion phases (green, orange, and pink). In the signal sampling phase, the top plates of the capacitor arrays DAC$_p$ and DAC$_n$ connect to the input voltages (V_{INP} and V_{INN}), while the bottom plates connect to either the positive reference voltage (V_{REFP}), the negative reference voltage (V_{REFN}), or the common-mode voltage (V_{CM}) according to the DAC-switching scheme [4]–[7]. At the end of signal sampling phase, the top-plate switches are turned OFF, and DAC$_p$ and DAC$_n$ hold the input signals on the top plates. In each bit-conversion phase, the comparator compares the voltages on the top plates (V_{topP} and V_{topN}) during the positive period of the comparator clock signal (CLK$_C$), the result of which determines each bit value. During the negative CLK$_C$ period, the comparator resets to the initial condition, and the DAC simultaneously generates a new V_{topP} and V_{topN} for the next comparison. This new voltage is generated according to the comparison result and DAC-switching scheme. The operations of the remaining bits are executed in sequence.

2) Synchronous SAR ADC Versus Asynchronous SAR ADC: There are two types of control schemes for SAR ADCs: 1) synchronous and 2) asynchronous. The differences between these two schemes include the time interval of the internal signal and external clock, timing constraints, and the verification process of the sizing results, as elaborated in the following.

Time Interval of Internal Signal and External Clock: The main difference of these two control schemes is the time interval length of the CLK$_C$ (T_{clkc}). The T_{clkc} value of the synchronous scheme is constant in each bit cycling, while the T_{clkc} value of the asynchronous scheme is unequal. As shown in Fig. 2(a), the synchronous control circuit generates an equal time interval CLK$_C$ according to the external clock signal (CLK$_{\text{EXT}}$). Despite the comparison/reset of the comparator being complete before the end of the positive/negative period, the next operation (i.e., reset/comparison) does not continue immediately, but waits until the transition edge of CLK$_C$. By contrast, CLK$_{\text{EXT}}$ only controls the time interval of the sampling phase (T_{Sample}) in the asynchronous scheme, where T_{Sample} is determined by the duty cycle of CLK$_{\text{EXT}}$, the driving strength of the input signal, and the pulsewidth modulator of the clock generator in the system. The asynchronous control circuit generates CLK$_C$ according to the comparator state rather than CLK$_{\text{EXT}}$ [see Fig. 2(b)]. Once the comparator determines the value of bit i after a certain amount of comparison time, say $T_{\text{cmp},i}$, it sends a valid signal to the SAR control circuit, and then pulls down CLK$_C$ with a delay time T_f. Subsequently, the falling CLK$_C$ resets the comparator in $T_{\text{rst},i}$, sending a trigger signal to the SAR control circuit, which pulls up CLK$_C$ with a delay time T_r. Since each operation executes immediately after the previous one is finished, the conversion speed of an asynchronous SAR ADC is generally faster than a synchronous one [3].

Timing Constraints: As abovementioned, after sampling the input signal, N clock cycles are needed to finish the conversion for a synchronous N-bit SAR ADC. In each bit-conversion phase, the comparator operates during the positive period of CLK$_C$. During the negative CLK$_C$ period, the comparator resets to the initial condition, and the DAC generates new V_{topP} and V_{topN} according to the comparison result. Thus, T_{clkc} can be formulated as

$$T_{\text{clkc}} = \frac{T_{\text{conv,syn}}}{N} \tag{1}$$

where $T_{\text{conv,syn}}$ is the period of conversion phases. If the duty cycle of the CLK$_C$ is 50%, the operating speed of the synchronous design is dominated by the longest time among the comparison time, the comparator reset time, and the required DAC settling time. That is, the timing constraint can be formulated as

$$\max(T_{\mathrm{cmp},i}, T_{\mathrm{rst},i}, T_{\mathrm{DAC},i}) \leq \frac{1}{2} T_{\mathrm{clkc}}, \quad i = 1 \sim N \tag{2}$$

where $T_{\mathrm{cmp},i}$ and $T_{\mathrm{rst},i}$ denote the comparison time and the reset time of bit i, respectively, while $T_{\mathrm{DAC},i}$ denotes the required settling time of bit i. $T_{\mathrm{cmp},i}$ is strongly related to the compared voltage (V_{cmp}) of the comparator. A smaller voltage difference for the comparator results in a longer comparison time. The longest $T_{\mathrm{cmp},i}$ is easy to determine via a single simulation, as in [8]. $T_{\mathrm{rst},i}$ almost remains constant and is usually shorter than $T_{\mathrm{cmp},i}$. If the fifth comparison in Fig. 2(a) requires the longest time, then $T_{\mathrm{cmp},5}$ should be shorter than half of T_{clkc}. In contrast, the asynchronous scheme generates CLK_C according to the feedback signal from the comparator. As shown in Fig. 2(b), this results in an unequal CLK_C time interval in each bit conversion. The total conversion time of the asynchronous SAR ADC ($T_{\mathrm{conv,asyn}}$) is calculated by the following equation:

$$T_{\mathrm{conv,asyn}} = \frac{1}{F_S} - T_{\mathrm{Sample}}. \tag{3}$$

The timing constraint is

$$T_{\mathrm{conv,asyn}} \geq \sum_{i=1}^{N} [T_{\mathrm{r}} + T_{\mathrm{cmp},i} + T_{\mathrm{f}} + \max(T_{\mathrm{DAC},i}, T_{\mathrm{rst},i})]. \tag{4}$$

T_r and T_f can be considered constants, while $T_{\mathrm{cmp},i}$ is affected by $V_{\mathrm{cmp},i}$. According to the constraints in (3) and (4), the longest total comparison time ($\Sigma T_{\mathrm{cmp},i}$) should be shorter than a given time budget; otherwise, it may violate the speed specification.

Sizing Solution Verification: A synchronous SAR ADC satisfies the speed requirement if its comparator can indicate the polarity of V_{cmp} within half of T_{clkc}. The condition of the longest comparison time can be obtained by a single simulation under the given small voltage difference and the smallest overdrive voltage of the comparator input pair [8].

Compared with a synchronous SAR ADC that verifies the sizing results by checking the longest single comparison time in a bit cycling, an asynchronous SAR ADC checks the longest total comparison time in the N conversion phases. To satisfy the speed specification of an asynchronous SAR ADC, $\Sigma T_{\mathrm{cmp},i}$ should be shorter than the given time budget for comparison ($\alpha_{\mathrm{cmp}} \times T_{\mathrm{asyn,conv}}$, $0 < \alpha_{\mathrm{cmp}} < 1$). However, the longest $\Sigma T_{\mathrm{cmp},i}$ is difficult to obtain through a single simulation, because it is related to the complex combination of T_{cmp}. An intuitive approach is to record T_{cmp} with respect to all possible V_{cmp} in a lookup table, and calculate the total comparison time from the lookup table based on the voltage differences during the successive conversion processes. However, constructing such a lookup table is rather time consuming and memory intensive. For example, if the minimum voltage step of V_{cmp} is set to $1/k$ of the least significant bit voltage (V_{LSB}) during evaluation of the comparison time T_{cmp}, there are $k \times 2^N$ possible V_{cmp} to be evaluated. Accordingly, $k \times 2^N$ simulations are required to obtain their comparison time. Moreover, when the common-mode voltage is changed during the bit conversion for some DAC-switching schemes [5], [7], more simulations and memory space are needed to construct the lookup table. For example, if the common-mode voltage of V_{cmp} is changed after the first bit, $k \times 2^N$ simulations are required in the first bit, and $k \times 2^{N-1}$ simulations in the remaining bits, since the voltage range of V_{cmp} is scaled down gradually in a typical SAR ADC. If the common-mode voltages of V_{cmp} are all different in bit conversions, it will take $k \times (2^{N+1} - 2)$ simulations (i.e., $k \times 2^N + k \times 2^{N-1} + \cdots + k \times 2$). For a 10-bit SAR ADC, assuming $k = 8$, 16 368 SPICE simulations are necessary, which may take more than 7 h for checking only one sizing result. Clearly, this is not a practical way to size a comparator. Accordingly, this bottleneck must be resolved if a feasible sizing tool is to be implemented.

B. Previous Works

Some previous works about design automation for SAR ADCs have been proposed. Allen and Barton [9] presented the first design automation work related to SAR ADCs. They chose a suitable architecture to realize the ADC according to certain specifications. However, because building blocks and switches are selected from a given cell library, their approach is restricted. The same limitation exists in the technology-independent compiler HiFADiCC, which was presented in [10]. Other works have suggested the use of behavior models in synthesis flow or verification [11]–[14]. Nevertheless, the synthesis results have only shown the basic functions without detailed performance reports, so these works have limited usages and lack accuracy. Moreover, such previous circuit techniques are now rather outdated, and there exist only limited choices in cell libraries. Thus, the power efficiency of these sizing methodologies may be insufficient nowadays.

To improve these drawbacks, a power-efficient sizing methodology was presented in [8], which combined the knowledge-based and optimization-based approaches. This method generates sizing results according to given specifications within a short execution time, the sizing results of which are at the transistor level. However, application of the approach is limited, because the circuit structure was based on an outdated DAC-switching procedure, bottom-plate sampling, and synchronous control scheme [15]. Moreover, the power efficiency, area cost, and operation speed can be improved using the new DAC-switching procedure, top-plate sampling, and asynchronous control scheme [3]–[7].

C. Our Contribution

Asynchronous SAR ADCs have become increasingly employed in recent years; however, most previous works have studied the sizing based on the synchronous structure. In response, this paper proposes a sizing methodology and tool for asynchronous SAR ADCs, named the hybrid-based asynchronous SAR ADC sizing tool (HaSAST). Our proposed tool can automatically generate transistor sizes of an asynchronous SAR ADC according to the given specifications and process data. The properties of our design methodology are listed in the following.

1) HaSAST combines the advantages of both the knowledge-based and optimization-based approaches to shorten the sizing time and enhance accuracy.
2) HaSAST is applicable to various structures of building blocks. Various types of comparator [16]–[19] and DAC-switching schemes [4]–[7] are compatible. For more general description, it is applicable to SAR ADCs employing latch-based dynamic comparators, top-plate sampling methods, and binary-weighted capacitive DACs.
3) HaSAST proposes an iterative flow to optimize the results; consequently, the proposed techniques shorten the optimization execution time. Slightly longer sizing time may yield a better solution through the optimization process, which is an acceptable tradeoff in a sizing program.

Since the arrangement of the asynchronous timing scheme is more flexible than the synchronous one, design complexity greatly increases. To handle this problem, this paper adopts several techniques to reduce complexity, the advantages of which are listed as follows.

1) *Comparator:* This paper adopts the curve-fitting technique, which reduces the number of simulations for estimating the longest total comparison time. We also employ a data-reusing technique to reduce the number of simulations during optimization. In this manner, not only the execution time is reduced by ∼95%, but any unnecessary power dissipation and occupied area are also avoided.
2) *S/H Circuit:* This paper adopts an efficient search algorithm, which combines depth-first-search (DFS) and breadth-first-search (BFS), as in [8], to obtain a good S/H circuit solution in a short time.
3) *Capacitive DAC:* This paper proposes several critical input stimuli (see Section III-C), instead of evaluating all possible input stimuli, to shorten the verification time of the DAC, and also enhances the sizing accuracy by considering the DAC dynamic gain error.
4) *SAR Control Logic:* During the sizing procedures mentioned above, the size of the SAR control logic circuit is also updated for suitable driving strength. Simultaneously adjusting the logic circuit size can save power dissipation and ensure correct function.

Overall, this paper demonstrates the combination of circuit design concept and search algorithms. Accordingly, this paper could aid circuit designers searching for design flow, sizing tool developers for sizing algorithms, and system designers for specification evaluation of building blocks. Furthermore, sizing results could constitute immediate solutions or design quality criteria for junior designers, or the initial solution for experienced designers.

The remainder of this paper is organized as follows. Section II offers an overview of the sizing procedure. Section III describes the function, operations, design considerations, and design methodology of each building block, including the comparator, the S/H circuit, the DAC, and the SAR control logic. Section IV presents the solution space exploration, while Section V reports the experiment results with various processes and specifications, and provides a comparison with some manual state-of-the-art works. We also show the effect of the proposed time-saving techniques. In addition, two silicon proofs are provided to reveal the feasibility of our methodology. Finally, the conclusion is drawn in Section VI.

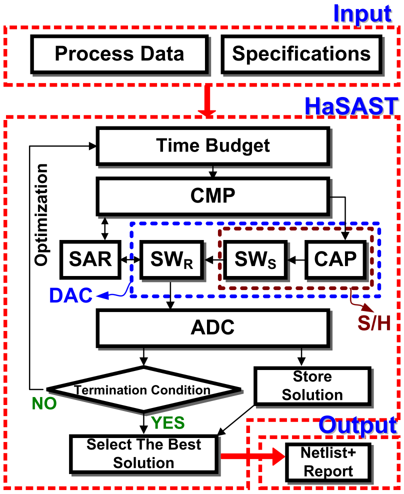


Fig. 3. Execution flow of the proposed sizing methodology.

II. Overview of the Sizing Procedure

This paper presents a systematic sizing methodology for asynchronous SAR ADCs. The execution flow is shown in Fig. 3. First, the process data and the specifications are prepared. Second, the time budget and the specification for each subblock are given. Third, an initial solution of each subblock is obtained. Finally, steps 2 and 3 are repeated to obtain a better solution after a few iterations.

A. Specification and Process Data

The input specifications and process data are listed in Table I, which consists of 14 items. The first seven items are the specifications and conditions of the SAR ADC, while the latter seven items are the parameters related to the process technology and environment.

In a real case, both multivoltage domains and device models, such as low-threshold and high-threshold voltage devices, may be used in an SAR ADC. Moreover, the size limitations of different types of device are different according to process technology. Therefore, these data can be, respectively, defined by a set of parameters.

If designers need to consider the worst case condition, they can run this tool under the given worst case condition. By doing so, the sizing results are guaranteed to satisfy the requirement in the worst case condition.

B. Time Budget of Comparator

In this stage, HaSAST allocates a time budget for the comparator and tries to find a solution. The comparator is the most important building block in an SAR ADC, the speed of which plays a major role in determining the conversion speed. Thus, after receiving user-defined parameters and

TABLE I
INPUT SPECIFICATIONS AND PROCESS DATA

Item	Item	Unit	Comment
Specifications and conditions	F_{S}	MS/s	Sampling rate
	T_{Sample}	ns	Sampling period
	F_{IN}	MHz	Input frequency
	N	bit	Resolution
	V_{DD}	Volt	Supply voltage
	V_{CM}	Volt	Input common-mode voltage
	V_{FS}	Volt	Input full-swing voltage
Process and environment-related parameters	$W_{\min}$	μm	Transistor minimum width
	$L_{\min}$	μm	Transistor minimum length
	$W_{\max}$	μm	Transistor maximum width
	$L_{\max}$	μm	Transistor maximum length
	$C_{\min}$	fF	Unit capacitance in DAC
	T_{emp}	°C	Temperature
	MN	None	Device model name in library

specifications of the SAR ADC, a time budget is allocated for the comparator first to search its solution (CMP).

In order to meet the design constraints in (3) and (4), the longest total comparison time ($\Sigma T_{\mathrm{cmp},i}$) should be found, where $\Sigma T_{\mathrm{cmp},i}$ is related to the input voltage of the SAR ADC. Moreover, the tool must find a specific input voltage that results in the longest $\Sigma T_{\mathrm{cmp},i}$, yet guarantee that its $\Sigma T_{\mathrm{cmp},i}$ is still shorter than the time budget. If it is not possible to find a solution under the initial time budget, the time budget will be increased gradually. Because the comparison/reset control circuit is simultaneously realized during sizing of the comparator, the remaining parts in (4), i.e., $\Sigma(T_r + T_f + T_{\mathrm{rst}})$, can also be obtained. It can be generally estimated as a constant time period. The summation of this constant period and time budget for comparison must be shorter than $T_{\mathrm{conv,asyn}}$; otherwise, the sizing procedure will be stopped.

C. Sizing of the Remaining Building Blocks

The remaining blocks are the S/H circuit and the capacitive DAC, which have one common device—the capacitor. Hence, the next step is to determine a proper capacitance and build a capacitor array (CAP). Once the capacitance is determined, the sampling switch (SW_S) and the reference switch (SW_R) are sized for ensuring satisfactory accuracy and speed. During these steps in sizing the CMP and SW_R, the size of the SAR control logic gate is modified simultaneously (SAR) to ensure the proper driving strength. Finally, an entire ADC is designed (ADC) upon the completion of the previous steps. Going through the sizing steps, HaSAST either generates an initial solution for an SAR ADC that satisfies the requirements, or it returns a failed result with a report describing the design bottleneck.

D. Optimization

The next step is to find a better solution. The initial solution is obtained under a given time budget for comparison ($\alpha_{\mathrm{cmp}} \times T_{\mathrm{asyn,conv}}$), which may remain a redundant time budget. Thus, a longer time budget (larger α_{cmp}) is set in the next iteration to slow down the comparator speed. In this optimization process, HaSAST selects a narrower range from the record (called data-reusing technique), which approaches the new design target. As such, the new solution in this narrow range is obtained in a shorter time. Details of the optimization are illustrated in Section III-A2. Moreover, all new solutions are recorded in comparison with the previous solutions; finally, the best solution is selected according to the user-defined cost function. If the specifications are loose, the initial solution may be sufficient without optimization; in such cases, users can turn OFF this option to save execution time.

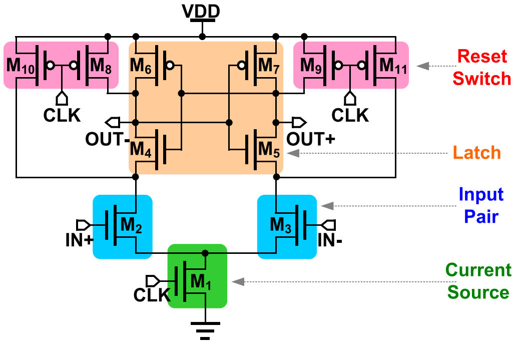


Fig. 4. Typical latch-based dynamic comparator.

E. Tradeoff Between Different Blocks

Because the time budget is divided into two parts (comparator and others), there is a speed tradeoff between the comparator and the other blocks. A faster comparator may result in inferior accuracy and power efficiency [16]. It also induces more kickback noise, which disturbs the DACs, though a larger DAC capacitance may alleviate this problem. During the sizing procedure, the related loadings of each block should be simulated to ensure that separate-block sizing still works for the integrated overall ADC. Hence, the increased capacitive load of the DAC would increase the switch size; accordingly, the control logic circuits would also require enlarging. Finally, because control logic circuits are the comparator loadings, a larger sized comparator would be needed. Again, a larger size of comparator increases the requirement for a larger DAC capacitance. In such cases, the solution may diverge and be difficult to obtain. However, this cycle can be broken by trying to size a slower comparator under a given DAC capacitance. Doing so minimizes the loadings of the DAC, control logic, and even the comparator itself. Moreover, a slower comparator may benefit accuracy and power efficiency. This concept is realized in the optimization procedure, as mentioned in Section II-D.

III. DESIGN CONSIDERATIONS AND SIZING PROCEDURES

This section individually presents the function, operations, design considerations, and design procedure of each building block.

A. Comparator

The comparator converts the analog input voltage into the digital domain in an SAR ADC. There exist two types of comparators: 1) the amplifier-based comparator and 2) the latch-based dynamic comparator. Because the latch-based dynamic comparator consumes less power than the amplifier-based comparator, it has been more widely adopted in recent years. Accordingly, this paper focuses on latch-based dynamic comparators.

Various latch-based dynamic comparator structures have been proposed [16]–[19]. Fig. 4 shows a typical comparator structure, which can be classified into four parts, including the input pair, latch, reset switch, and current source. Its operations can be roughly divided into reset and comparison phases. In the reset phase, it turns OFF the current source; hence, the current paths from the VDD to ground are cutoff. According to the comparator structure, the output nodes are then reset through the reset switches by connecting them to either VDD, ground, or each other. Fig. 4 shows an example of resetting to VDD. In the comparison phase, the current source is turned ON. Then, the input pair amplifies the input voltage, and the latch pushes and pulls the output of the amplifier to the rail-to-rail level. Finally, the polarity of the input signal is obtained.

1) Design Considerations: In order to consider process variation, accuracy, and comparison speed, the transistors of the comparator are sized in sequence as follows.

1) *Input Pair:* The input pair is sized first, because the resolution of the comparator is determined by the voltage offset (V_{OS}), and V_{OS} can be determined after the differential input pair size is determined. V_{OS} is impacted by process variation, which can be formulated as follows:

$$V_{\mathrm{OS}} = \Delta V_{\mathrm{TH}} + \frac{(V_{\mathrm{GS}} - V_{\mathrm{TH}})}{2}\left(\frac{\Delta S}{S} + \frac{\Delta R}{R}\right) \quad (5)$$

where V_{TH} and ΔV_{TH} are the threshold voltage and its mismatch of the input pair, respectively. V_{GS} is the gate-to-source voltage of the input pair, while S and ΔS are the physical dimension and its mismatch between the input pair, respectively. R is the load resistance and ΔR its mismatch [20]. V_{OS} is usually a normal distribution, and its standard deviation is denoted by σ_{offset}. To achieve the ADC resolution, σ_{offset} should be smaller than $V_{\mathrm{LSB}}/6$ [21], as defined in

$$\sigma_{\mathrm{offset}} \leq \alpha_{\mathrm{OS}} \times V_{\mathrm{LSB}}, \quad V_{\mathrm{LSB}} = \frac{V_{\mathrm{FS}}}{2^N} \quad (6)$$

where V_{FS} is the full-scale input swing voltage of the ADC, and N is the ADC resolution. It should be noted that there may exist a multistage structure of comparators. In a multistage comparator, the input pair size of the latter stage can be scaled down according to the estimated gain of the previous stages. These estimations can be verified by simulation in the last step.

2) *Latch:* The latch provides a positive feedback loop. In most cases, small transistor sizes are enough to satisfy the speed requirement of the comparator. However, once the process variation is considered, their size is usually enlarged. To prevent mismatch, the latch size is usually set to the similar order as the input pair.

3) *Reset Switches:* The reset switches must reset the comparator to the initial condition before the end of reset phase. Because the loadings of the reset switches include input pair and latch, the reset switches are sized after sizing the input pair and the latch. In general, the reset switches are all controlled by the clock signal from the digital control circuit; however, a part of the reset switches is controlled by its internal comparator nodes [19]. Moreover, the order of the sizing process is determined by the order of the reset process. A detailed example is presented in the Section III-A2.

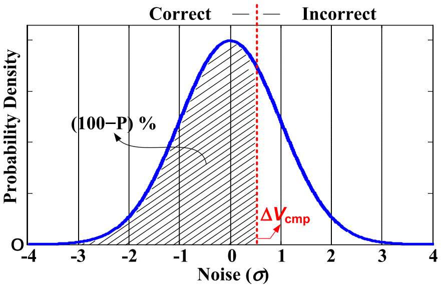


Fig. 5. Probability function of comparator noise.

4) *Current Source:* A comparison of speed and accuracy is the tradeoff in sizing the current source [16]. A larger current-source width usually results in faster comparison speed, worse accuracy, and higher power consumption. After the comparison time budget is given, the sizing result must satisfy the constraints in (3) and (4).

5) *Noise:* Comparator noise is an important issue, especially in medium-to-high-resolution ADCs. Assume comparator noise is normal distribution with mean value μ and standard deviation σ, then the σ of comparator noise can be obtained by a noise transient simulation. It is derived from the inverse cumulative distribution function with the bit error rate (BER) of a given voltage (ΔV_{cmp}) comparison. As shown in Fig. 5, for example, if the BER is $P\%$, then the distribution must satisfy the relationship that the slash area occupies a $(100 - P)\%$. Then, the root mean square (rms) of comparator noise power ($V_{\mathrm{cmp,rms}}^2$) is the variance (σ^2) of this distribution. It could be taken into the noise budget calculation in (7).

2) Design Methodology (CMP): Based on the considerations illustrated in the Section III-A1, this section shows our comparator sizing procedure (see Fig. 6). In order to reduce the running time without compromising accuracy, HaSAST applies the curve-fitting and data-reusing techniques to achieve better power efficiency and lower hardware cost. This procedure can be separated into four steps.

Step 1 (IPL$_I$): The sizes of the input pair and the latch are determined first. Due to the difficulty in determining the initial values, we first give them reasonable initial values [22] and then adjust them in the later steps. The width of the pMOS in the latch is determined by the gain parameter (α_{gain}), where α_{gain} is typically about 4 to 10 from our experience. Then, the nMOS size in the latch is determined by the P–N ratio parameter (α_{PN}), where α_{PN} is about 2 to 3.

Step 2 (RS): Next, the tool gives the minimum reset-switch size and checks the reset function. If the reset nodes reach the target voltage (e.g., VDD in Fig. 4) within an acceptable error (e.g., <1 V_{LSB}) in the specified reset time, they pass the reset function test. Otherwise, the switch width is increased

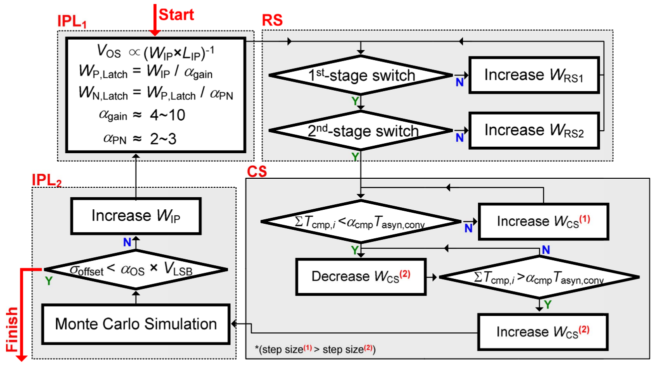

IPL$_1$
Start
RS
$V_{OS} \propto (W_{IP}\times L_{IP})^{-1}$
$W_{P,Latch} = W_{IP} / \alpha_{gain}$
$W_{N,Latch} = W_{P,Latch} / \alpha_{PN}$
$\alpha_{gain} \approx$ 4~10
$\alpha_{PN} \approx$ 2~3
1st-stage switch
Increase W_{RS1}
2nd-stage switch
Increase W_{RS2}
N
Y
IPL$_2$
CS
Increase W_{IP}
$\sigma_{offset} < \alpha_{OS} \times V_{LSB}$
Monte Carlo Simulation
Finish
$\Sigma T_{cmp,i} < \alpha_{cmp} T_{asyn,conv}$
Increase $W_{CS}^{(1)}$
Decrease $W_{CS}^{(2)}$
$\Sigma T_{cmp,i} > \alpha_{cmp} T_{asyn,conv}$
Increase $W_{CS}^{(2)}$
*(step size$^{(1)}$ > step size$^{(2)}$)

Fig. 6. Sizing procedure of the comparator.

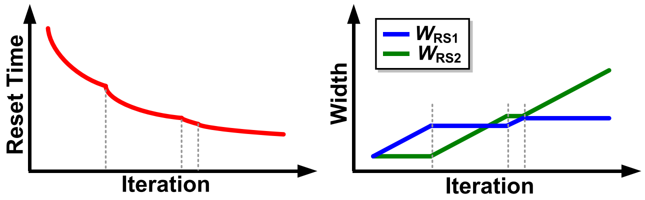
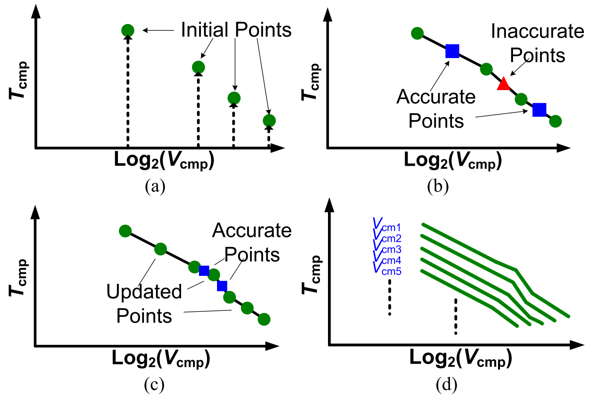


Fig. 7. Sizing example of the reset switches in a two-stage structure. The width of the first stage and the second stage is increased alternately to reduce the overall reset time.

until they pass the test or exceed the size limitation. Moreover, if the later stage is reset by the output of the previous stage in a multistage structure [19], the previous stage should be reset first to ensure the reset function of the later stage. In a two-stage structure, when the switch width of the second stage (W_{RS2}) is increased, its loadings to the first stage are also added. Although the increased W_{RS2} may decrease the reset time of the second stage, the reset time of the first stage may be increased slightly. As such, the switch width of the first stage (W_{RS1}) should be increased again to improve the overall reset time. This procedure will repeat alternately until either both stages pass the test or one of them reaches the size limitation. A sizing example of a two-stage comparator is shown in Fig. 7. As can be seen, the reset time is decreased as the reset-switch width is increased alternately.

Step 3 (CS): The current source is sized in this step. HaSAST first finds the longest $\Sigma T_{cmp,i}$ of minimum size, and gradually adjusts the size to decrease or increase the longest $\Sigma T_{cmp,i}$ until, finally, $\Sigma T_{cmp,i}$ approaches the given time budget. In order to efficiently find the longest $\Sigma T_{cmp,i}$, this paper adopts the curve-fitting and data-reusing techniques.

Because the relationship between the comparison time and the logarithm of input voltage difference is nearly linear [23], the estimated T_{cmp} of V_{cmp} can be obtained by curve fitting, as shown in Fig. 8. In Fig. 8(a), the initial points (green circular points) are obtained by simulation. Because T_{cmp} is linearly proportional to $\log_2(V_{cmp})$, more internal points can be estimated by interpolation, which are then verified by simulations. These estimated points are considered correct if the error between the estimation and the simulation is smaller than a predefined threshold (e.g., 1%). The blue squares in Fig. 8(b) denote the points that meet the required accuracy, while the red triangles indicate that the error exceeds the threshold. Once an error occurs, additional interpolation points are inserted to fit the curve in this region. The blue squares and the red triangles are both updated by the simulation results in this step. After all inserted points fit the accuracy requirements, the entire curve is obtained, as shown in Fig. 8(c). Finally, all curves with various common-mode voltages are obtained in the same way [see Fig. 8(d)]. Moreover, we can obtain the total comparison time of an analog-to-digital conversion from Fig. 8(d). For example, if the input voltage of the ADC is V_{IN} and the input full swing voltage is V_{FS}, then we can determine its $T_{cmp,1}$ at $\log_2(V_{IN})$ in the curve denoted by V_{cm1}; its $T_{cmp,2}$ is obtained at $\log_2(|V_{IN} - 2^{-1}V_{FS}|)$ in the curve denoted by V_{cm2}; its $T_{cmp,3}$ is obtained at $\log_2(||V_{IN} - 2^{-1}V_{FS}| - 2^{-2}V_{FS}|)$ in the curve denoted by V_{cm3}, with the remaining bits obtained in a similar way.



(a)

(b)

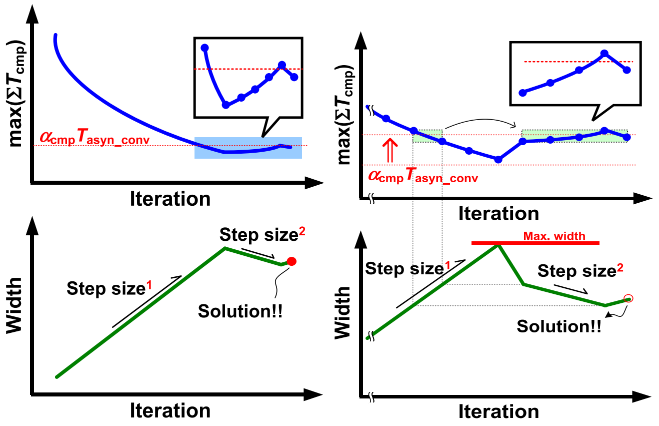
(c)

(d)

Fig. 8. Curve fitting with piecewise linear function. (a) Initial points. (b) Interpolation. (c) Correction and interpolation. (d) Piecewise linear curves.

After the longest $\Sigma T_{cmp,i}$ is obtained from the various values of V_{IN}, the current-source size is adjusted by a bidirectional search method, as shown in Fig. 9(a). If the longest $\Sigma T_{cmp,i}$ is longer than the $\alpha_{cmp} \times T_{asyn,conv}$, the current-source width is increased with a step size (W_{step1}) until $\Sigma T_{cmp,i}$ becomes shorter than $\alpha_{cmp} \times T_{asyn,conv}$. Then, the current-source width is decreased with a smaller step size (W_{step2}) until $\Sigma T_{cmp,i}$ is longer than $\alpha_{cmp} \times T_{asyn,conv}$ again. Finally, the current-source width is increased once with the step size W_{step2}; as such, we can obtain a solution that approaches the target. As shown in Fig. 9(b), if a solution does not exist after searching the full range, α_{cmp} is increased such that $\alpha_{cmp} \times T_{asyn,conv}$ is higher than the existing data points. To avoid searching the whole range again, the new search range is narrowed to the two existing data points that are closest to $\alpha_{cmp} \times T_{asyn,conv}$. Finally, the solution in this narrower range can be obtained in a shorter running time.

Step 4 (IPL$_2$): Finally, the input pair and the latch, which have been roughly sized in IPL$_1$, are verified. We use Monte Carlo simulation to check the validity of the current design's V_{OS}. Note that three standard deviations of V_{OS} (σ_{offset}) should be smaller than half of V_{LSB} or the target defined as σ_{OS}. If it does not pass this test, the width of the input pair (W_{IP}) will be increased, and this tool generates an enlarged input pair and latch in IPL$_1$ to decrease V_{OS}. The sizing loop continues until this comparator passes all aforementioned tests or exceeds the size limitation. In addition, the sizing results of this tool can combine with calibration



(a) (b)

Fig. 9. Two examples of the current-source sizing procedure. (a) Tool searches the full range and obtains a solution. (b) Solution does not exist for the given target, so the tool adjusts α_{cmp} and searches for the solution again in a subrange, which has been roughly searched before (called data-reusing technique).

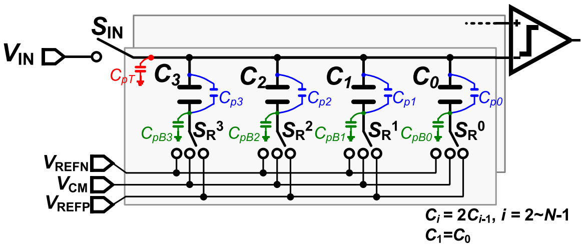


Fig. 10. Typical 3-bit capacitive DAC with embedded S/H circuit and parasitic capacitors.

techniques [19] to enhance the performance. Then, we can remove IPL$_2$ from the sizing procedure, or just reduce the offset constraint to shorten the sizing time.

B. S/H Circuit

The S/H circuit samples the analog input signal in the sampling phase of the SAR ADC, and then holds this signal in the capacitor array for the succeeding bit conversions. The S/H circuit is composed of sampling switches (S_{IN}) and a capacitor array, which is integrated into the DAC. Fig. 10 shows a 3-bit capacitive DAC with an embedded S/H circuit.

1) Design Considerations: A good sampling switch samples the input signal of the SAR ADC without distortion, i.e., high signal-to-noise-and-distortion ratio (SNDR). The key point in designing an S/H circuit is to determine the proper sizes of the sampling switch and capacitance to satisfy the required SNDR. If the input signal is a sine wave and the probability density function of the quantization error is constant, the required SNDR of the ADC output (SNDR$_{\text{ADC}}$) is $6.02 \times N + 1.76$ dB, where N is the resolution of the ADC [24]. SNDR$_{\text{ADC}}$ can also be a user-defined input specification, for which the resolution should be $N = \lceil(\text{SNDR}_{\text{ADC}} - 1.67)/6.02\rceil$. By considering some noise sources, SNDR$_{\text{ADC}}$ can be formulated as follows:

$$\text{SNDR}_{\text{ADC}} = 10\log_{10}\left(\frac{V_{\text{sig,rms}}^2}{V_{q,\text{rms}}^2 + V_{\text{kT}/C,\text{rms}}^2 + V_{n,\text{rms}}^2 + V_{\text{cmp,rms}}^2}\right) \tag{7}$$

where $V_{\text{sig,rms}}$ is the rms of the input signal, $V_{q,\text{rms}}$ is the rms of the ADC quantization noise, $V_{\text{kT}/C,\text{rms}}$ is the rms of the

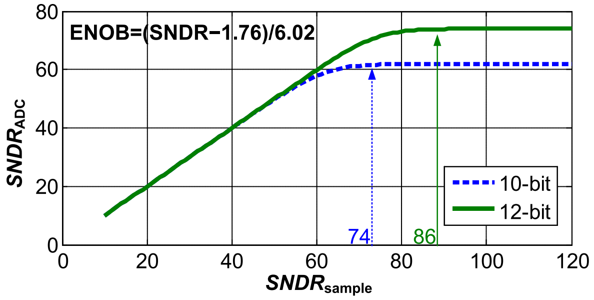


Fig. 11. SNDR of the S/H circuit and ADC.

kT/C noise, $V_{n,\text{rms}}$ is the rms of the sampling noise sourced from the S/H circuit (a parameter sums up the nonidealities of S/H circuit, for example, the nonideality induced by charge injection and clock feedthrough of sampling switch), and $V_{\text{cmp,rms}}$ is the rms of the comparator noise. $V_{q,\text{rms}}$ can be obtained from the input voltage swing and the resolution of the ADC [24], while $V_{\text{kT}/C,\text{rms}}$ can be determined by the capacitance of the sampling capacitor [25], [26].

1) *Sampling Capacitance:* The first step in sizing the S/H circuit is to determine the capacitance value by considering the noise, capacitor physical structure, process mismatch, and parasitic capacitors. In order to reduce the charging/discharging energy during capacitor switching, a small capacitance is preferable. However, a small capacitance exhibits larger kT/C noise and is more sensitive to process variation. The limitation on the minimum value of capacitance and its corresponding variance due to process variation are related to the layout style, including unit capacitor structure, placement, and wire routing [26]–[28]. Moreover, these parameters are usually determined by the physical design simulation tool.

2) *Sampling Switch:* The design target of the sampling switch is to determine the proper size to achieve a low $V_{n,\text{rms}}$ that satisfies the required SNDR$_{\text{ADC}}$ in (7), i.e., high enough SNDR of the S/H circuit (SNDR$_{\text{sample}}$). SNDR$_{\text{sample}}$ is defined as follows:

$$\text{SNDR}_{\text{sample}} = 20\log_{10}\left(\frac{V_{\text{sig,rms}}}{V_{n,\text{rms}}}\right). \tag{8}$$

Thus, SNDR$_{\text{sample}}$ can be determined once SNDR$_{\text{ADC}}$ is given. Fig. 11 offers an example, where the required SNDR$_{\text{sample}}$ should be higher than about $6.02 \times (N + 2) + 1.76$ dB to achieve the sufficient effective number of bits (ENOBs) for the ADC. After the design target of the S/H circuit has been defined, the next step is to find a proper switch size. For a given sampling capacitance and period (T_{sample}), if the transistor width of the sampling switch is too small, it will result in a large resistance and low input bandwidth. In contrast, if the width is too large, the performance is also degraded by clock feedthrough and charge injection. Moreover, because the comparator contributes a part of the parasitic capacitor C_{pT}, as shown in Fig. 10, it is necessary to include the comparator in the simulation while sizing the sampling switch.

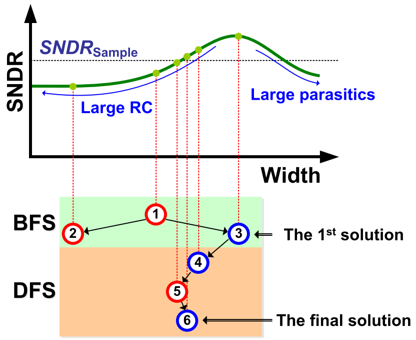


Fig. 12. Sizing example of the sampling switch.

2) Design Methodology (CAP and SW$_S$): The design flow of the S/H circuit is divided into two steps: 1) CAP for the sampling capacitor and 2) SW$_S$ for the sampling switch as follows.

Step 1 (CAP): In this step, HaSAST only confirms whether the kT/C noise consideration is violated. The unit capacitance is related to the physical design and sampling period, while the sampling period T_{Sample} is related to the driving strength of the previous stage. As a result, they are user-defined parameters in this tool.

Step 2 (SW$_S$): A proper transistor width for the sampling switch is obtained in this step. Based on the design concerns [8], we use a complete binary tree to represent the search space, as shown in Fig. 12. Each node in the binary tree denotes a width, and there exists an SNDR value corresponding to each node. Let $W_{\max}$ ($W_{\min}$) represent the maximum (minimum) width of a transistor. The root in the zeroth level of the tree denotes the medium width [i.e., $(W_{\max} + W_{\min})/2$]. There exist 2^i nodes in the ith level of the tree, which are labeled from left to right. Let n^i_j denote the jth node in the ith level. The width of n^i_j is $W_{\min} + (j - 0.5) \times (W_{\max}-W_{\min})/2^i$. In order to search solutions efficiently, our approach combines the BFS and the DFS. To avoid wasting time on an unreasonable target, we set a maximum search depth, which is defined as $I_{\max} = \lfloor \log_2(T_{\text{BS}}/t_{\text{BS}}) \rfloor$, where t_{BS} is the simulation time for checking one sizing result and T_{BS} is the sizing time limitation on sampling switch. The sizing procedure is as follows.

1) *Initialization:* The procedure starts from the root n_{root} in the binary tree (i.e., $i = 0$). If the SNDR of n_{root} is less than SNDR$_{\text{sample}}$, call BFS(i); otherwise, call DFS(n_{root}).
2) *BFS(i):* Go to the next level (i.e., let $i = i + 1$). The procedure stops if i is larger than $I_{\max}$. Search the nodes in the level in the increasing order until the SNDR of node n^i_j is larger than SNDR$_{\text{sample}}$, and call DFS(n^i_j); otherwise, call BFS(i).
3) *DFS(n):* Go to the next level (i.e., let $i = i + 1$). The procedure stops if i is larger than $I_{\max}$. Call DFS(n_{left}) if the SNDR of node n^i_j satisfies the constraint. Otherwise, call DFS(n_{right}). n_{left} (n_{right}) denotes the left (right) child of n^i_j.

The execution time of this search method is related to the difficulty of the target. In the best case, it only requires applying DFS, for which the number of simulations is equal to the depth of the binary tree. A sizing example of the sampling switch is shown in Fig. 12. BFS is initially applied to the first three nodes. After the third node meets the required SNDR, it starts to optimize the width by applying DFS. The procedure stops when the depth is larger than $I_{\max}$.

C. Digital-to-Analog Converter

Due to several advantages, such as lower power consumption and better device matching, capacitive DACs have become more common in recent years. By switching its reference switches, the DAC generates voltage levels to comparator in bit-conversion phases. The 3-bit DAC shown in Fig. 10 consists of a capacitor array (C_i, $i = 0$–3) and reference switches (S^i_R, $i = 0$–3), which select reference voltages among V_{REFP}, V_{REFN}, and V_{CM}.

1) Design Considerations: There exist three main concerns in designing the DAC, namely, the gain error, incomplete settling, and capacitor mismatch. However, capacitor mismatch is strongly related to the physical design, so we focus on the two former issues in this sizing methodology paper.

1) *Gain Error:* Gain error is induced by parasitic capacitors; accordingly, the input swing of the SAR ADC should be decreased to fit the actual input range. Otherwise, the ENOB of the ADC could degrade significantly.
2) *Incomplete Settling:* Too slow settling behavior leads to an imprecise voltage level and thus an incorrect decision at the next comparison phase [29]. Moreover, incomplete settling is generated using an improperly sized reference switch without considering parasitic capacitors. Incomplete settling leads to differential nonlinearity (DNL) and integral nonlinearity (INL) errors.

Before discussing the design considerations, there are three types of parasitic capacitors that should be considered:

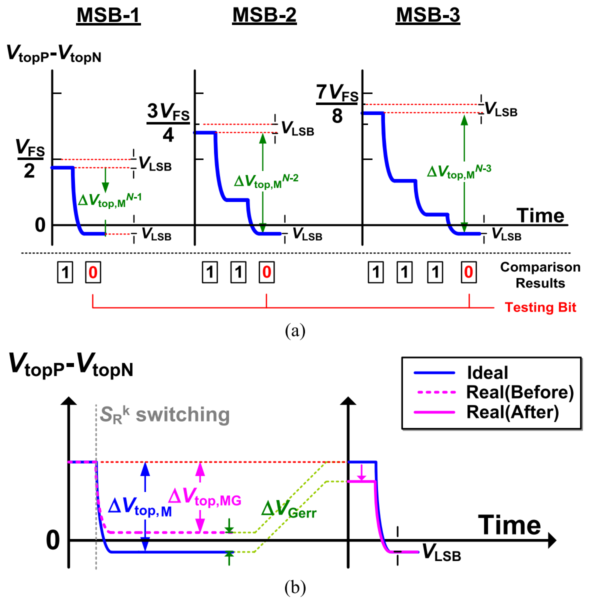
1) C_{pT}: the overall top plate to ground parasitic capacitor, including the input capacitor of the comparator, the source/drain capacitor of the sampling switch, and the top plate of the capacitor array to ground parasitic capacitor;
2) C_{pBi}: bottom plate of C_i to ground parasitic capacitor;
3) C_{pi}: top plate to bottom plate of C_i parasitic capacitor.

Ideal capacitor switching results in perfect voltage change on the top plate ($\Delta V_{\text{top,ideal}}$), which can be computed by

$$\Delta V_{\text{top,ideal}} = \frac{\sum_{i=1}^{N-1} (C_i \times \Delta V_{\text{bot},i})}{\sum_{i=0}^{N-1} C_i} \tag{9}$$

where $\Delta V_{\text{bot},i}$ is the voltage change of the bottom plate of C_i. Due to the capacitor mismatch induced by C_{pi}, nonlinearity is induced if the ratio of C_{pi} is different from the ratio of C_i in

$$\Delta V_{\text{top},M} = \frac{\sum_{i=1}^{N-1} [(C_i + C_{\text{pi}}) \times \Delta V_{\text{bot},i}]}{\sum_{i=0}^{N-1} (C_i + C_{\text{pi}})}. \tag{10}$$



(a)

(b)

Fig. 13. DAC provides different voltage levels on the top plate during bit conversion. (a) Initial input test signals for testing the reference switch without considering the gain error. (b) Example of the adjustment of (MSB−1) the input test signal by considering the gain error.

In addition to nonlinearity, the gain error G_{err} is induced by considering C_{pT} and can be expressed as in

$$\Delta V_{\text{top,MG}} = \frac{\sum_{i=1}^{N-1}[(C_i + C_{pi}) \times \Delta V_{\text{bot},i}]}{C_{\text{pT}} + \sum_{i=0}^{N-1}(C_i + C_{pi})} = \Delta V_{\text{top},M} \times G_{\text{err}} \tag{11}$$

$$G_{\text{err}} = \frac{\sum_{i=0}^{N-1}(C_i + C_{pi})}{C_{\text{pT}} + \sum_{i=0}^{N-1}(C_i + C_{pi})}. \tag{12}$$

To enhance accuracy, these parasitic capacitors induced by the comparator and sampling switch should be included in the simulations during sizing reference switches. Thus, the DAC should be sized after the comparator and the S/H circuit have been sized.

After the gain error is obtained, the input signal swing is adjusted to size the reference switches. Reference switches select reference voltages, and are controlled by the SAR control logic to perform the search algorithm. DAC incomplete settling happens if the transistor width of the switch is too small or large because of large ON-resistance or large intrinsic parasitic capacitors. Besides, $C_{\text{pB}i}$ slows down the settling speed, since it increases loading on the reference switches. The reference switches are sized under these considerations.

2) Design Methodology (SW_R): The proposed reference switches (SW_R) sizing methodology optimizes the power and area of the DAC. There are also some implementation techniques in this section, which reduce the simulation time and enhance accuracy.

A DAC has to generate the correct voltage level for comparison within the given settling time for each bit k. Hence, our target is to determine the proper size of reference switch S_R^k one by one such that it can completely settle in the given settling time. The voltage change on the top plates of the DAC_p and DAC_n must be measured after S_R^k switching (denoted by ΔV_{top}^k). ΔV_{top}^k is equal to $(V_{\text{topP}}^k - V_{\text{topN}}^k) - (V_{\text{topP,sample}} - V_{\text{topN,sample}})$, where $V_{\text{topP,sample}} - V_{\text{topN,sample}}$ and $V_{\text{topP}}^k - V_{\text{topN}}^k$ are the differential voltages of the DAC at the end of the sampling phase and after S_R^k switching, respectively. To check whether S_R^k completely settles in the given time, our method enters critical input test stimuli [see Fig. 13(a)] to the ADC. At first, the value of $V_{\text{topP}}^k - V_{\text{topN}}^k$ is positive, but becomes negative with a small difference in V_{LSB} (i.e., $-V_{\text{LSB}} < V_{\text{topP}}^k - V_{\text{topN}}^k < 0$) after the test switch S_R^k switching; thereafter, it is called zero crossing. If the DAC completely settles, the testing bit k must be opposite to the previous bit (i.e., 0 to 1 or 1 to 0). The sizing result can be checked by simply observing the testing bit rather than measuring ΔV_{top}^k at uncertain timings, which enhances accuracy.

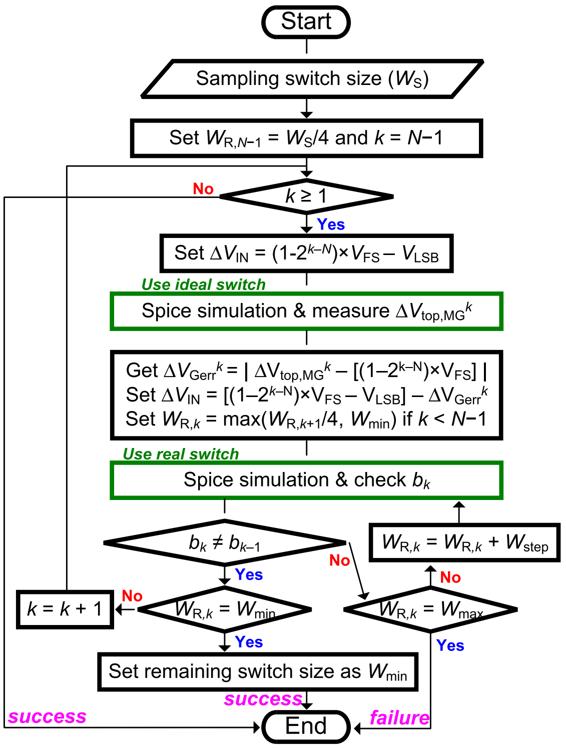
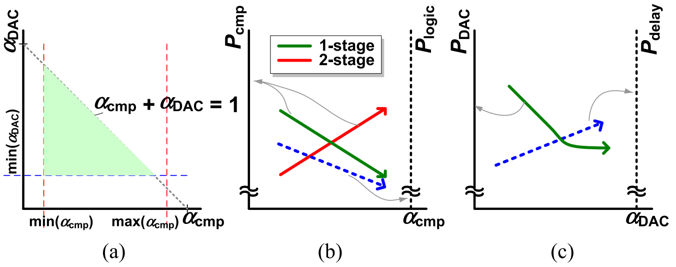

Start
Sampling switch size (W_S)
Set $W_{R,N-1}$ = W_S/4 and k = N–1
$k \geq 1$
No
Yes
Set ΔV_{IN} = (1-2^{k-N})×V_{FS} – V_{LSB}
Use ideal switch
Spice simulation & measure $\Delta V_{top,MG}^k$
Get ΔV_{Gerr}^k = | $\Delta V_{top,MG}^k$ – [(1–2^{k-N})×V_{FS}] |
Set ΔV_{IN} = [(1–2^{k-N})×V_{FS} – V_{LSB}] – ΔV_{Gerr}^k
Set $W_{R,k}$ = max($W_{R,k+1}$/4, W_{min}) if k < N–1
Use real switch
Spice simulation & check b_k
$b_k \neq b_{k-1}$
Yes
No
$W_{R,k}$ = $W_{R,k}$ + W_{step}
k = k + 1
No
$W_{R,k}$ = W_{min}
Yes
No
$W_{R,k}$ = W_{max}
Yes
Set remaining switch size as W_{min}
success
success
failure
End

Fig. 14. Sizing procedure of reference switches in the DAC.

However, as shown in Fig. 13(b), ΔV_{top}^k decreases from $\Delta V_{\text{top},M}^k$ to $\Delta V_{\text{top,MG}}^k$ due to the gain error, where $\Delta V_{\text{top},M}^k = (1 - 2^{k-N}) \times V_{\text{FS}}$ if the binary-weighted capacitor array is used. ΔV_{Gerr}^k is the voltage difference between $\Delta V_{\text{top},M}^k$ and $\Delta V_{\text{top,MG}}^k$. The expected zero crossing would not happen if $|\Delta V_{\text{Gerr}}^k|$ is larger than V_{LSB}. To resolve this problem, the critical input test stimulus must be reduced by the value of ΔV_{Gerr}^k. Based on the above discussion, our sizing procedure is separated into two steps, as shown in Fig. 14.

Step 1: In order to ensure that the settling behavior of a reference switch is correct in Step 2, we have to estimate the gain error of a design. First, $\Delta V_{\text{top,MG}}^k$ is obtained from the simulation result. Then, the difference between $\Delta V_{\text{top},M}^k$ and $\Delta V_{\text{top,MG}}^k$, which is denoted by ΔV_{Gerr}^k, is determined. Then, the input stimulus for the ADC is decreased by ΔV_{Gerr}^k to keep the zero crossing after S_R^k switching. In this



(a) (b) (c)

Fig. 15. Solution exploration. (a) Search boundary. (b) Time budget of the comparator versus power consumption of the comparator and control logic. (c) Time budget of the DAC versus power consumption of the DAC and delay cell.

step, we use ideal switches to estimate the gain error, because the voltage level of the settling target can be obtained in a short simulation time without compromising accuracy.

Step 2: Input the stimulus obtained from Step 1, and check the settling behavior. The width of S_R^k is gradually increased until its settling behavior is acceptable, where k is from $N-1$ to 1 (i.e., MSB−1 to LSB).

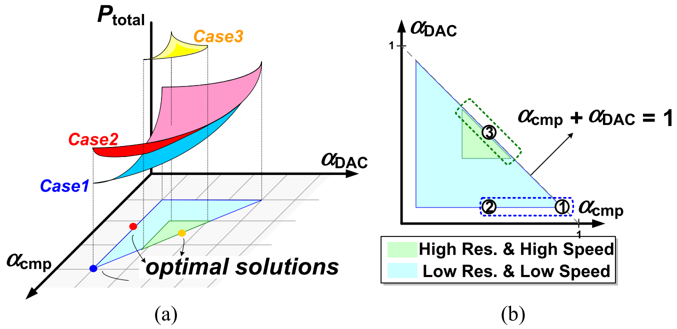
D. SAR Control Logic

In comparison with analog circuits, digital circuits are usually seen as the easier part of sizing. Fixed-size transistors may satisfy the function, but it may also result in poor speed performance or inferior power efficiency. Moreover, the power consumption and the occupied area of the control circuit might become the major part as the analog circuits are optimized. To avoid redundant power consumption and area cost, HaSAST simply adjusts the size of the control circuit according to the general sizing concept, such as parasitic delay, logic effort, and electrical effort [30]. In this tool, SAR modifies the size of the logic gates in each iteration of CMP and SW$_R$ to provide the suitable driving strength for the new loadings.

IV. Exploration of Solution Space

This section shows the exploration of solution space. There is a design tradeoff between the time budget for each block and its power consumption. The solution space is represented in Fig. 15(a), in which the x-axis is α_{cmp} and the y-axis is α_{DAC}, denoting the time budget for the comparator and the DAC, respectively, where α_{DAC} is defined as $\alpha_{\text{DAC}} = \{\Sigma[T_r + T_f + \max(T_{\text{rst}}, T_{\text{DAC}})]\}/T_{\text{asyn,conv}}$. If the required DAC settling time T_{DAC} is longer than the comparator reset time T_{rst}, adding delay cells into the signal path of the asynchronous control logic is needed to satisfy the timing constraint $T_{\text{rst}} + T_{\text{delay}} = T_{\text{DAC}}$, where T_{delay} is the gate delay of the delay cell. The upper and lower bounds of α_{cmp} can be obtained during sizing of the tail current source of the comparator, as described in Section III-A. The lower bound of α_{DAC} is determined by the valid solution with the shortest T_{DAC}. In addition, the summation of α_{DAC} and α_{cmp} should not be larger than 1.

The relationship between α_{cmp} and the power consumption of the comparator (P_{cmp}) and control logic (P_{logic}) is shown in Fig. 15(b). A smaller α_{cmp} (i.e., shorter comparison time) implies a wider tail current source of the comparator (i.e., larger loading to control logic). Hence, α_{cmp} is inversely proportional to P_{logic} (dashed line). The relationship between α_{cmp} and P_{cmp} is dependent on the structure of the comparator (solid lines). For a single-stage comparator, such as in [16], the wider tail current source (i.e., smaller α_{cmp}) consumes higher P_{cmp}, since the current source controls the current path from the supply to ground. For a multistage comparator, such as in [19], the power consumption of the first stage is almost independent of its current-source width. However, the slower first stage (i.e., larger α_{cmp}) implies that the power consumption of the latter stage is higher due to the longer transient time.



(a) (b)

Fig. 16. Three cases of optimal solutions positioned at different locations. (a) Relationship between budget parameters and power consumption. (b) 3-D solution spaces in (a) projected on the budget parameters.

The relationship between α_{DAC} and the power consumption of the DAC (P_{DAC}) and delay cells (P_{delay}) is shown in Fig. 15(c). The larger α_{DAC} (i.e., longer T_{delay}) for the DAC is realized by more delay cells, so α_{DAC} is directly proportional to P_{delay} (dashed line). The smaller α_{DAC} (i.e., faster DAC switching) requires wider transistor as switches and results in larger power consumption (solid line).

As a result, the relationship between the time budget (i.e., α_{cmp} and α_{DAC}) and low power solution is shown in Fig. 16(a). If the specifications of the resolution and sampling frequency are low (Cases 1 and 2), the solution space will be wide and close to the lower bound of α_{DAC} [Fig. 16(b) (blue region)], since the required DAC settling time is short, and the redundant delay cells for DAC settling are unnecessary. The solution with the lowest power consumption may locate at number 1 or 2 [Fig. 16(b)], which depends on the comparator structure. If the resolution and the sampling frequency are high (Case 3), the solution space will be narrow [green region and number 3 in Fig. 16(b)] and close to the line where $\alpha_{\text{DAC}} + \alpha_{\text{cmp}} = 1$. This implies that the time budget is fully utilized. The appropriate search range, which refers to the usage record of the same process technology, may be applied to various specifications and shorten the sizing time.

V. Experimental Results

This sizing methodology was realized in C++ programming language, and ran on a 3.3-GHz Intel Core i7-3960X PC employing HSPICE as the core circuit simulator. During the sizing procedure, the proposed tool loads setup file, generates SPICE and circuit netlist files, calls simulator to verify the result, and adjusts the size according to the simulation result. Finally, the netlist files of the SAR ADC and sizing reports were automatically generated.

TABLE II

SIZING CONDITIONS, SPECIFICATIONS, AND SIMULATION RESULTS

Case	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Technology (nm)	180				90				55				40			
Resolution (bit)	12	12	10	8	12	12	10	8	12	12	10	8	12	12	10	8
Sampling rate (MS/s)	0.2	20	50	80	0.2	50	150	250	0.2	50	150	200	0.2	50	150	200
Supply voltage (V)	0.6	1.8	1.8	1.8	0.4	1	1	1	0.6	1.2	1.2	1.2	0.55	1.1	1.1	1.1
SNDR (dB)	73.9	72.5	61.4	49.9	72.4	73.4	61.1	49.4	73.8	73.4	61.6	49.9	73.8	72.8	61.9	49.9
Peak DNL (LSB)[1]	0.25	0.25	0.25	0.25	0.25	0.50	0.50	0.25	0.25	0.25	0.25	0.25	0.25	0.5	0.25	0.25
Peak INL (LSB)[1]	0.16	0.52	0.20	0.24	0.31	0.50	0.50	0.33	0.13	0.32	0.22	0.17	0.25	0.25	0.25	0.14
Power Consumption (μW)[7]	0.44	681	689	856	0.44	250	395	475	0.21	180	371	286	0.15	155	318	234
FoM (fJ/conversion-step)	0.54	9.60	14.4	42.8	0.66	1.33	2.89	8.03	0.26	0.95	2.50	5.64	0.19	0.88	2.12	4.69
Avg. Sizing Time (min.)[2]	126	45	42	30	25	26	27	26	101	80	183	38	146	71	131	93
$R_{\mathrm{CF,save}}$ (%)	99.7	99.8	99.4	98.4	99.9	99.8	99.5	98.2	99.8	99.8	99.5	98.2	99.8	99.8	99.4	98.0

[1] Magnitude of maximum/minimum DNL or INL. [2] Avg. sizing time does not include verification time.[3] Input frequency: Nyquist rate. [4] The 55 nm and 40 nm processes used in this table are low-power technology processes, so their supply voltages and threshold voltages of transistors are higher than those of 90nm. [5] $R_{\mathrm{CF,save}}$ = Saving time by curve-fitting technique. [6] Case 1, 5, 9, 13 (low supply voltage cases) use the circuit techniques [32][33]. [7] Averaging power breakdown of each sub-circuit: S/H circuit(11.4%), comparator(31.6%), SAR control logic(41.7%), and DAC(15.3%).

A. Comparison With Other Manual State-of-the-Art Works

To verify the feasibility of this methodology, we compared it with some manual state-of-the-art works. In this experiment, we used the monotonic switching method [5] and the comparator structure in [19] as our prototype. Table II lists 16 sizing cases under various conditions, including process technology, resolution, sampling rate, and supply voltage. Table II also shows the sizing results, including SNDR, peak DNL and INL, power consumption, figure of merit (FoM) [31], average sizing time for each solution (excluding the verification time of the ADC), and the saving time ratio of sizing comparator by curve-fitting technique ($R_{\mathrm{CF,save}}$). The sizing results show that the methodology is applicable to various conditions and specifications.

To reveal in the capability of implementing high-performance designs, Fig. 17 shows the sampling rate and SNDR/FoM of the results generated from this sizing methodology (red symbols), a previous work (green symbols) [8], and other manual works (blue symbols), which were published in major integrated circuit design conferences [1]. Note that our results in Fig. 17 are shown at the locations of high speed, high SNDR, and low FoM. This implies that our methodology is able to reach manual design targets in simulation.

Because power consumption will increase after layout, comparing the prelayout simulations and measurement results may introduce bias. However, it is reasonable to assume that the power consumption is double due to the parasitic devices in the layout, and ENOB decreases 1-bit because of nonidealities of manufacture. Under this assumption, the FoM results will be four times worse than those shown in Fig. 17, but our sizing results still have high competitiveness.

B. Execution Time Analysis

This section reveals the efficiency of this sizing methodology. The execution time can be separated into three parts, namely, sizing simulation time, program run time, and verification time. The sizing simulation time is the total run time of the simulator during the building-blocks sizing process, excluding the verification time of the entire ADC. Rather, verification time is the simulation time of verifying the complete ADC. The remaining part is the program run time, which includes the computation, carrying out the proposed algorithms, and the file I/O.

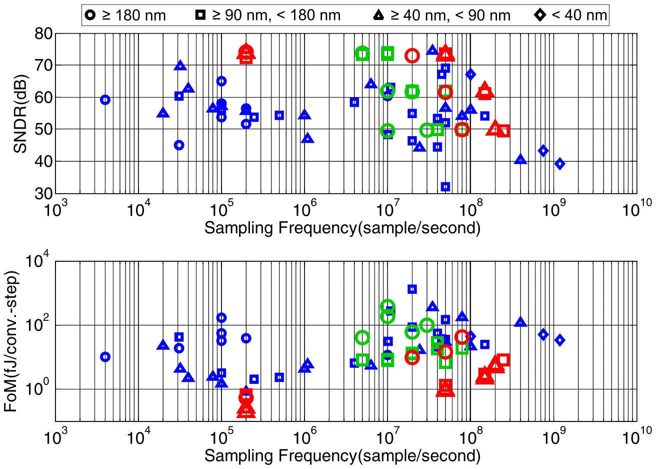


Fig. 17. Sampling frequency and SNDR/FoM of manual works (blue symbols), previous work [8] (green symbols), and this paper (red symbols). These state-of-the-art manual works were published in major integrated circuit design conferences (ISSCC and SoVC) from 1997 to 2015. This figure only shows single-channel SAR ADCs for similar comparisons.

From Table II (last row), the average sizing time for one solution for each case is only about 0.5–2.5 h, and the execution time is very short (hours) in comparison with the one of manual work (days or weeks). According to the experimental results, the average sizing time for the first solution is ~29.5% of the total execution time (T_{exe}), and the average sizing time for one of the rest solutions is about only 6% of T_{exe}. This shows that the data-reusing technique mentioned in Section III-A (Fig. 9) can shorten the sizing time by narrowing the search space during optimization. The total verification time takes about 44.3% of T_{exe}, which implies that T_{exe} would be much shorter as the simulator is improved in the future. The program run time takes about 0.1% of T_{exe}, since the algorithms are simple and efficient.

Moreover, the curve-fitting technique significantly reduces the comparator sizing simulations, which usually occupy the most sizing time, especially in a high-speed SAR ADC. As mentioned in the last paragraph of Section I-A, if the proposed technique is not used, it will take $k \times (2^{N+1} - 2)$

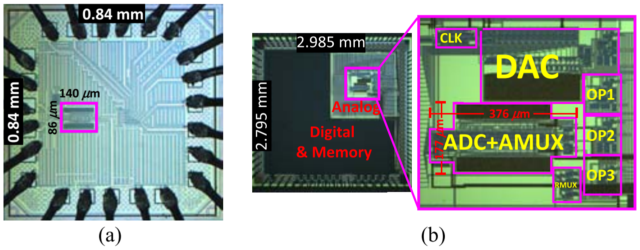


Fig. 18. Die photos of two silicon proofs. (a) 90-nm chip in Table III. (b) 180-nm chip (a biomedical sensor system) in Table III.

TABLE III

MEASUREMENT RESULTS OF SILICON PROOFS

	[8]	This Work					
Technology (nm)	110	180				90	
Experiment	Meas.	Meas.			Sim.[3]	Meas.	Sim.[4]
Resolution (bit)	10	8	10	12		10	
Sampling rate (MS/s)	20	0.6				50	
Supply voltage (V)	1.2	3.3				1	
SNDR (dB)	53.4	49.7	52.1	62.7	73.9	52.1	61.9
Peak DNL (LSB)[1]	0.75	0.29	1.41	2.79	0.1	1.41	0.06
Peak INL (LSB)[1]	0.95	0.24	2.81	5.24	0.22	2.81	0.1
Power consumpt. (μW)	385	328[2]			53.5	440	155
FoM (J/conv.-step)	50f	2.2p	0.7p	0.5p	22f	26.7f	3.1f

[1] Magnitude of maximum/minimum DNL or INL. [2] Estimated value from simulation. [3] Avg. sizing time: 48 (min.) [4] Avg. sizing time: 22 (min.)

simulations for only checking one sizing result of the comparator ($k = 8$ in these experiments). By involving the circuit design concepts rather than exhausting the search, the average number of simulations can be decreased 98% ($R_{\mathrm{CF,save}}$ in Table II). For example, each comparator simulation in Case 1 takes about 1.6 s; accordingly, it will take about 7.28 h to check only one comparator sizing result without our proposed techniques. In this case, it takes only less than 3 min with our proposed techniques.

C. Solution Optimization

The lowest FoM is selected as the best solution in this paper. A good solution with low FoM should consume less power and simultaneously have higher SNDR. Though the first solution may be sufficient, users can active the optimization option and let HaSAST try to decrease the power consumption. From the experimental results, the best FoM would be about 20%–100% (if only one solution is valid) of the first solution.

D. Silicon Proofs

Fig. 18 shows two silicon proofs to reveal the practicality of our sizing results. Both use the switchback switching method [7] and the comparator structure in [19].

The first one is a 10-bit 50-MS/s SAR ADC fabricated in TSMC 90-nm CMOS process [Fig. 18(a)]. The measured SNDR is 52.1 dB, and the FoM is 26.6 fJ/conversion-step. The peak DNL and the INL are shown in Table III, and are dominated by the capacitor mismatch and the bonding wire effect. Though it is not as good as the simulation results, it still has a competitive result in comparison with manual works.

The second one is an 8–12-bit 600-kS/s SAR ADC fabricated in TSMC 180-nm CMOS process. It was integrated into a biomedical sensor system, as shown in Fig. 18(b). Due to the system requirements, it used I/O devices with a 3.3 V supply voltage rather than the typical core devices with 1.8 V. It is also modified to a single-ended input structure due to the single input signal source. After truncating bits for different applications, the measured SNDRs are 62.7 dB (12 bit), 59.5 dB (10 bit), and 49.7 dB (8 bit). Furthermore, it consumes less than 1% power of the entire system-on-chip, and the FoM is about 0.5 pJ/conversion-step. The peak DNL and the INL are shown in Table III, which are truncated bits for 10-bit application. The comparisons of the silicon proofs between previous works and this one are summarized in Table III. The performances of simulation result are close to their design targets, but the power consumption of simulation result is much lower than the measurement result. The reason is the parasitic capacitors are induced after layout. Besides, some circuits are combined with this 180-nm SAR ADC for application, such as clock divider, voltage level shifters, and built-in-self-test circuit. Thus, the power consumption becomes much higher.

The experimental results show that HaSAST can achieve humanlike design targets under various process technologies within a relatively short time and also maintain good competitiveness in power efficiency.

VI. CONCLUSION

This paper presented a systematic design procedure and a sizing tool named HaSAST for asynchronous SAR ADCs. The presented design flow and the search procedure result in fast search speed and good performance. The proposed methodology is not limited to a single-circuit structure; rather, it can be applied to various comparator structures and DAC-switching schemes. The experimental results show that the sizing ability of HaSAST is close to that of human circuit designers. Nevertheless, the main goal of this tool is not to replace circuit designers, but to assist them. Its results can be an initial solution for circuit designers and combined with the postprocesses, such as manual refinement and/or calibration, to enhance outcomes. The evaluation of different specifications under different process technologies will become much easier and faster with HaSAST, which helps system designers to determine the specifications of SAR ADCs in systems.

REFERENCES

[1] B. Murmann. *ADC Performance Survey 1997–2015*. [Online]. Available: http://www.stanford.edu/~murmann/adcsurvey.html, accessed Jun. 2015.

[2] F. A. E. Rocha, R. M. F. Martins, N. C. C. Lourenço, and N. C. G. Horta, "State-of-the-art on automatic analog IC sizing," in *Electronic Design Automation of Analog ICs Combining Gradient Models With Multi-Objective Evolutionary Algorithms*. New York, NY, USA: Springer-Verlag, 2014.

[3] S.-W. M. Chen and R. W. Brodersen, "A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2669–2680, Dec. 2006.

[4] W.-Y. Pang, C.-S. Wang, Y.-K. Chang, N.-K. Chou, and C.-K. Wang, "A 10-bit 500-KS/s low power SAR ADC with splitting comparator for bio-medical applications," in *Proc. IEEE A-SSCC*, Nov. 2009, pp. 149–152.

[5] C.-C. Liu, S.-J. Chang, G.-Y. Huang, and Y.-Z. Lin, "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, Apr. 2010.

[6] V. Hariprasath, J. Guerber, S.-H. Lee, and U.-K. Moon, "Merged capacitor switching based SAR ADC with highest switching energy-efficiency," *Electron. Lett.*, vol. 46, no. 9, pp. 620–621, Apr. 2010.

[7] G.-Y. Huang, S.-J. Chang, C.-C. Liu, and Y.-Z. Lin, "10-bit 30-MS/s SAR ADC using a switchback switching method," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 3, pp. 584–588, Mar. 2013.

[8] C.-P. Huang, S.-J. Chang, G.-Y. Huang, and C.-W. Lin, "A power-efficient sizing methodology of SAR ADCs," in *Proc. IEEE ISCAS*, May 2012, pp. 365–368.

[9] P. E. Allen and P. R. Barton, "A silicon compiler for successive approximation A/D and D/A converters," in *Proc. IEEE CICC*, May 1986, pp. 552–555.

[10] S. G. Sabiro, P. Senn, and M. S. Tawfik, "HiFADiCC: A prototype framework of a highly flexible analog to digital converters silicon compiler," in *Proc. IEEE ISCAS*, May 1990, pp. 1114–1117.

[11] N. C. Horta, J. E. Franca, and C. A. Leme, "Framework for architecture synthesis of data conversion systems employing binary-weighted capacitor arrays," in *Proc. IEEE ISCAS*, Jun. 1991, pp. 1789–1792.

[12] G. Ruan, "A behavioral model of A/D converters using a mixed-mode simulator," *IEEE J. Solid-State Circuits*, vol. 26, no. 3, pp. 283–290, Mar. 1991.

[13] C. Leme *et al.*, "Flexible silicon compilation of charge redistribution data conversion system," in *Proc. 34th Midwest Symp. Circuits Syst. (MWSCAS)*, May 1991, pp. 403–406.

[14] N. C. Horta, J. Vital, and J. E. Franca, "Automatic multi-level macromodel generation for data conversion systems employing binary-weighted capacitor-arrays," in *Proc. IEEE ISCAS*, May 1992, pp. 2561–2564.

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

[16] B. Wicht, T. Nirschl, and D. Schmitt-Landsiedel, "Yield and speed optimization of a latch-type voltage sense amplifier," *IEEE J. Solid-State Circuits*, vol. 39, no. 7, pp. 1148–1158, Jul. 2004.

[17] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, "A double-tail latch-type voltage sense amplifier with 18 ps setup + hold time," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 314–315.

[18] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink, and B. Nauta, "A 1.9 μW 4.4 fJ/conversion-step 10 b 1 MS/s charge-redistribution ADC," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2008, pp. 244–245.

[19] M. Miyahara, Y. Asada, D. Paik, and A. Matsuzawa, "A low-noise self-calibrating dynamic comparator for high-speed ADCs," in *Proc. IEEE A-SSCC*, Nov. 2008, pp. 269–272.

[20] S. Jiang, M. A. Do, K. S. Yeo, and W. M. Lim, "An 8-bit 200-MSample/s pipelined ADC with mixed-mode front-end S/H circuit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1430–1440, Jul. 2008.

[21] T. Ogawa *et al.*, "Non-binary SAR ADC with digital error correction for low power applications," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, Dec. 2010, pp. 196–199.

[22] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.

[23] J. Guerber, M. Gande, H. Venkatram, A. Waters, and U.-K. Moon, "A 10 b ternary SAR ADC with decision time quantization based redundancy," in *Proc. IEEE A-SSCC*, Nov. 2011, pp. 65–68.

[24] D. A. Johns and K. Martin, "Chapter 11 data converter fundamentals," in *Analog Integrated Circuit Design*. New York, NY, USA: Wiley, 1997, p. 451.

[25] J. Mao, F. Jonsson, and L.-R. Zheng, "Mismatch aware power and area optimization of successive-approximation ADCs," in *Proc. 17th IEEE Int. Conf. Electron., Circuits, Syst.*, Dec. 2010, pp. 882–885.

[26] P. J. A. Harpe *et al.*, "A 26 μW 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.

[27] M. P.-H. Lin, Y.-T. He, V. W.-H. Hsiao, R.-G. Chang, and S.-Y. Lee, "Common-centroid capacitor layout generation considering device matching and parasitic minimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 7, pp. 991–1002, Jul. 2013.

[28] C.-W. Lin, J.-M. Lin, Y.-C. Chiu, C.-P. Huang, and S.-J. Chang, "Mismatch-aware common-centroid placement for arbitrary-ratio capacitor arrays considering dummy capacitors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 12, pp. 1789–1802, Dec. 2012.

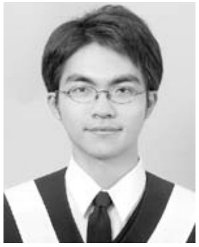
[29] G.-Y. Huang, S.-J. Chang, C.-C. Liu, and Y.-Z. Lin, "A 1-μW 10-bit 200-kS/s SAR ADC with a bypass window for biomedical applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2783–2795, Nov. 2012.

[30] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Reading, MA, USA: Addison-Wesley, 2010.

[31] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[32] D. C. Daly and A. P. Chandrakasan, "A 6-bit, 0.2 V to 0.9 V highly digital flash ADC with comparator redundancy," *IEEE J. Solid-State Circuits*, vol. 44, no. 11, pp. 3030–3038, Nov. 2009.

[33] S. I. Chang, K. Al-Ashmouny, and E. Yoon, "A 0.5 V 20 fJ/conversion-step rail-to-rail SAR ADC with programmable time-delayed control units for low-power biomedical application," in *Proc. IEEE 37th Eur. Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2011, pp. 339–342.



Chun-Po Huang was born in Tainan, Taiwan, in 1986. He received the B.S. degree in electrical engineering from National Cheng Kung University, Tainan, in 2008, where he is currently pursuing the Ph.D. degree.

His current research interests include the design automation for high-speed and low-power analog-to-digital converters.

Jai-Ming Lin received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1996, 1998, and 2002, respectively, all in computer science.

He was an Assistant Project Leader with the CAD Team, Science Park, Realtek Corporation, Hsinchu, from 2002 to 2007. He is currently an Assistant Professor with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His current research interests include floor plan, placement, routing, and clock tree synthesis.

Ya-Ting Shyu received the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2008, where she is currently pursuing the Ph.D. degree.

Her current research interests include integrated circuit design and design automation for analog and mixed-signal circuits.

Soon-Jyh Chang (M'03) was born in Tainan, Taiwan, in 1969. He received the B.S. degree in electrical engineering from National Central University, Zhongli, Taiwan, in 1991, and the M.S. and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1996 and 2002, respectively.

He has been with the Department of Electrical Engineering, National Cheng Kung University, Tainan, since 2003, where he has also been a Professor and the Director of Electrical Laboratories since 2011. He has authored or co-authored around 100 technical papers and holds seven patents. His current research interests include design, testing, and design automation for analog and mixed-signal circuits.

Dr. Chang was a recipient and co-recipient of many technical awards, including the best paper award of the Institute of Electronics, Information and Communication Engineers in 2010, the Best GOLD Member Award from the IEEE Tainan Section in 2010, the ISSCC/DAC Student Design Contest Award in 2011, and the ISIC Chip Design Competition Award in 2011. He has served as the Chair of the IEEE SSCS Tainan Chapter since 2009. He was the Technical Program Co-Chair of the IEEE International Symposium on Next-Generation Electronics in 2010, and the Committee Member of the IEEE Asian Test Symposium in 2009, the Asia and South Pacific Design Automation Conference in 2010, the International Symposium on VLSI Design, Automation and Test in 2009, 2010, and 2012, and the IEEE Asian Solid-State Circuits Conference in 2009 and 2011.