A Power-Efficient Sizing Methodology of SAR ADCs

Chun-Po Huang*, Soon-Jyh Chang†, Guan-Ying Huang, and Cheng-Wu Lin
Department of Electrical Engineering, National Cheng-Kung University, Tainan, Taiwan
Email*: gppo@sscas.ee.ncku.edu.tw and Email†: soon@mail.ncku.edu.tw

***Abstract*— Analog-to-digital converter (ADC) is a vital component for modern electronic systems, but designing an ADC usually takes much time and effort. Though several synthesis methods have been presented for analog circuits, there exists limited works focusing on ADC design automation. In this paper, we propose a systematic sizing methodology to minimize the power consumption for successive approximation register (SAR) ADCs in transistor level. This method manipulates the characteristics of SAR ADC to develop an efficient searching algorithm for shortening the sizing time. The time complexity of our method is $O(2\log_2|S|)$, where $|S|$ is the number of candidates in the searching space. According to the proposed sizing flow, we develop a sizing tool which is independent of manufacturing process and is able to minimize power consumption for SAR ADCs. By using the developed sizing tool, a proof-of-concept prototype was carried out within only 15 minutes and fabricated in a 1P4M 0.11μm process. The measurement results show the prototype demonstrates a high competitiveness compared to other state-of-the-art works on performance and power efficiency.**

I. Introduction

ADCs as the bridge between analog and digital domains are key components for many mixed-signal systems. Their characteristics generally play an important role in determining the performance of the whole system. In order to extend battery life, power consumptions of mixed-signal circuits and systems in energy-limited applications such as smartphones, portable biosignal acquisition devices, and wireless sensor networks are critical. Thus, power efficiency becomes a primary concern in the design of ADCs used for energy-limited applications. Among many ADC architectures, SAR ADC has received a great interest because of its power efficiency in this kind of applications.

SAR ADC has the advantage of simple structure and low power consumption, and is probably the most widely used ADC structure for medium-resolution, medium-speed applications. Due to the increasing speed of devices in advanced technologies, it has enabled SAR ADCs to penetrate into the medium-to-high speed applications during recent years. A SAR ADC mainly consists of a sample-and-hold (S/H) circuit, a comparator, a digital-to-analog converter (DAC), and digital control logic. SAR ADC is an opamp-free architecture; in other words, SAR ADC does not require high gain and high bandwidth opamps to guarantee its speed and linearity. A high-performance opamp must consume much power, and will suffer from short channel effect and low supply voltage in advanced process technologies. These reasons arouse circuit designers' interest, and it is reflected in the number of related papers that have presented in recent major circuit design conferences [1]. However, there exists limited works studying SAR ADC design automation so far.

In contrast to digital circuits, the number of transistors in mixed-signal circuits is small. But, design automation for mixed-signal circuits is generally regarded as a difficult task and is hard to automate because of the complexity of the interactions that have to be considered. Therefore, designing mixed-signal sub-blocks usually consumes a large part of time and effort in developing a system. How to develop effective synthesis tools for mixed-signal circuits is an attractive topic in electronic design automation field. A typical synthesis flow of analog design automation includes topology selection, device sizing, and layout generation. In this paper, we focus on the device sizing for SAR ADCs, which adjusts component sizes to meet ADC specifications. Many device sizing methods have been presented for analog circuits, and they can be classified into four categories [2]. The first is knowledge-based approach. Several circuit design flows are embedded in the program with well sizing definitions. According to circuit specifications, the program is able to select a correct design flow and calculate the device sizes [3]. The second is simulation-based approach. This approach employs standard circuit simulator to evaluate circuit performance. This method gradually meets the specifications by iterative circuit adjustments and simulations [4]. The third is equation-based approach, which directly solves circuit equations to optimize the circuit performance. Though the sizing process of this method is very fast, the sizing results are usually inaccurate in advanced process technologies due to the over-simplified circuit equations [5]. The fourth is neural-network-based approach. The advantage of this method is fast execution speed, but it requires large amount of design examples and time to train the neural-network in advance [6]. In order to gain accurate sizing results without much execution time, we would like to propose a sizing flow that combines the knowledge-based and simulation-based approaches. Fig. 1 shows the abstract model of our device sizing flow. The establishment of circuit design flow is knowledge-based while the device sizing engine is simulation-based. Thus, both execution speed and accuracy can be enhanced.

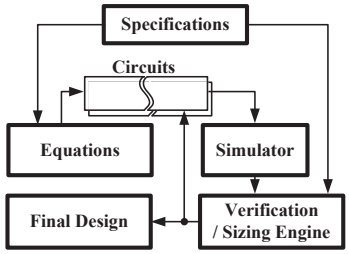

Fig. 1. Abstract model of analog synthesis

In this paper, we present the first SAR ADC sizing methodology considering power efficiency. In comparison to previous works studying analog sizing, we use standard transistor models in all simulations, rather than behavior models and ideal switches [7]. Based on the sizing methodology, we implement a transistor-level synthesis tool for SAR ADCs with conventional architecture. The selected SAR ADC architecture is much applicable since the limitation of input signal swing ($V_{\rm swing}$) and common mode voltage ($V_{\rm CM}$) is eased [8]. For silicon proof, a power-efficient SAR ADC has been synthesized by our tool and laid out through manual design. The measurement results show that the power efficiency are improved significantly by our sizing

978-1-4673-0219-7/12/$31.00 ©2012 IEEE

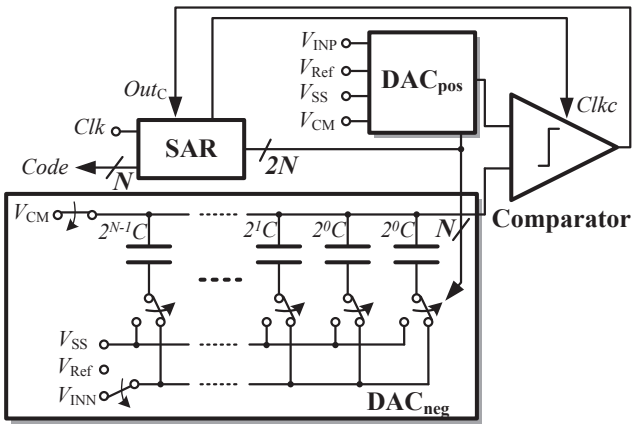

Fig. 2. An *N*-bit SAR ADC in sample mode operation

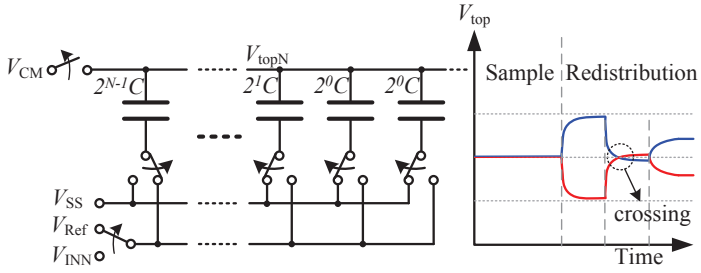

Fig. 3. Redistribution mode operation

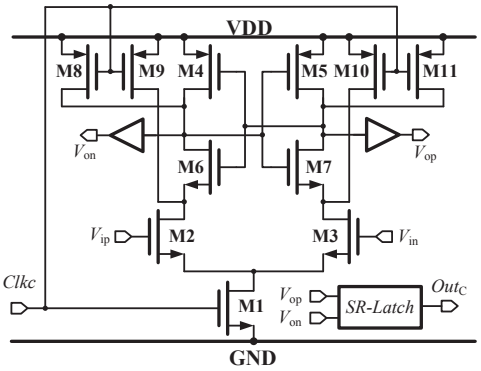

Fig. 4. Comparator

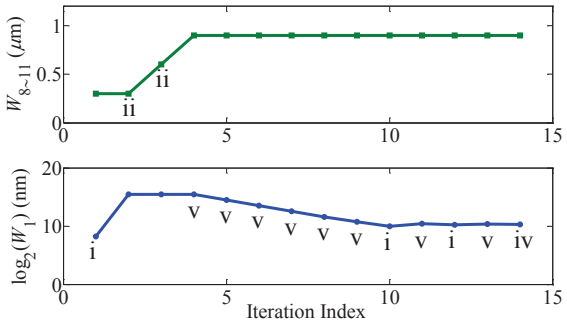

Fig. 5. A sizing example of comparator

methodology. Furthermore, the conversion speed of our measurement result is even faster than the simulation results of previous works. With the developed sizing tool, circuit designers could speed up their work, shorten the time to market, and increase the competitiveness of their products.

The remainder of this paper is organized as follows. Section II describes the basic architecture and operation of SAR ADCs. Section III presents the sizing flow and considerations. Section IV reports the experiment results and the comparison with some state-of-the-art works [11]–[14]. Finally, Section V concludes this paper.

II. Architecture and Operation of SAR ADC

Fig. 2 depicts an *N*-bit fully-differential charge redistribution SAR ADC [9]. The basic building blocks of such a SAR ADC contain a comparator, two binary-weighted DACs with sampling switches, and a digital SAR control logic. A data conversion is accomplished by two operations. First, the sample mode, the capacitor top plates at both positive and negative sides connect to $V_{\rm CM}$ and the bottom plates to the input voltage ($V_{\rm INP}$,$V_{\rm INN}$). Second, the redistribution mode, is a differential operation. When a switch connects to $V_{\rm SS}$, the corresponding switch at another side connects to the reference voltage ($V_{\rm Ref}$). We will only describe the operation of one side. As shown in Fig. 3, the top switch that connected to $V_{\rm CM}$ is then opened, and the bottom plates connect to $V_{\rm SS}$ except that the most significant bit (MSB) switch connects to $V_{\rm Ref}$. Then comparator compares the voltage of both top plates ($V_{\rm topP}$,$V_{\rm topN}$). The comparison result determines the MSB. In the next bit phase, SAR logic connects the MSB switch to $V_{\rm SS}$ or $V_{\rm Ref}$ by the given result, and raises the next bit to $V_{\rm Ref}$ again. The operations of remaining bits are executed in sequence.

III. Sizing Flow and Considerations

In Fig. 2, the signal constructs a loop among the building blocks. By assuming that the DAC signal settles well, we could break the loop and choose comparator as the first circuit to design. Then the remaining sub-circuit will be sized one by one.

The structure of comparator is shown in Fig. 4. When $Clkc$ is high (comparison phase), the transistor M1 provides path current (I_1), and the differential pair (M2,M3) and the latch (M4∼M7) distinguish the input signals. When $Clkc$ goes to low (reset phase), M8 to M11 reset the outputs ($V_{\rm op}$,$V_{\rm on}$) to $V_{\rm DD}$. Because the system specifications have determined $V_{\rm CM}$ and the last significant bit voltage ($V_{\rm LSB}$), the uncertain variables are transistor sizes. The sizes of M2 to M7 are dominated by the threshold voltage mismatch ($\sigma_{\rm Vth}$) as equation (1), where $A_{\rm Vth}$ is the area proportionality constant for $V_{\rm th}$.

$$\sigma_{\rm Vth} = \frac{A_{\rm Vth}}{\sqrt{WL}} \tag{1}$$

The transistor size of M1 controls I_1, the comparator with weaker I_1 consumes lower power but takes longer comparison time and may obtain incorrect comparison result. I_1 is increased by enlarging W_1/L_1, and is decreased by oppositive way. The transistors M8 to M11 should be large enough to reset signal before the next comparison phase. Our tool analyses the different simulation results and then adjusts the transistor size.

The sizing process of transistors M1 and M8 to M11 is determined according to the following conditions: (i) the output voltage difference in comparison phase is too small; (ii) the reset nodes cannot reach V_{DD} in reset phase; (iii) strong I_1 pulls down both output nodes to low voltage ($V_{\rm L}$), this will lead to an incorrect result because the "00" is not a defined state of SR-latch; (iv) late or incorrect result due to weak I_1; and (v) correct result, then updates the solution. In the sizing procedure, our tool will check these conditions from (i) to (v) one by one after each simulation iteration. This sizing procedure is finished when the size reach the maximum width ($W_{\rm max}$) or the size variation is smaller than the limit (ΔW). The whole sizing flow is shown as Algorithm III.1, and an example is shown in Fig. 5. In this case, $W_{8\sim11}$ are monotonic increasing variables, and they converge in a few iterations. In the sizing procedure, our tool will minimize

Algorithm III.1: SIZINGCMP(Specifications)

comment: Initialization

$$\begin{cases} V_{\rm LSB} \leftarrow \frac{V_{\rm swing}}{2^N} \\ L_{1\sim 11} \leftarrow L_{\rm min} \\ W_{8\sim 11} \leftarrow W_{\rm min}, W_1 \leftarrow W_{\rm min} \\ A_{2\sim 3} \leftarrow (\frac{A_{\rm Vth}}{V_{\rm LSB}})^2 \end{cases}$$

comment: Transistor size of M2 to M7

while $\frac{A_{2,3}}{L_{2,3}} > W_{\rm max}$

do $L_{2,3} \leftarrow L_{2,3} + \Delta L$

$$\begin{cases} W_{2,3} \leftarrow \frac{A_{2,3}}{L_{2,3}} \\ W_{4,5} \leftarrow \frac{W_{2,3}}{2} \\ W_{6,7} \leftarrow \frac{W_{2,3}}{4} \end{cases}, \begin{cases} L_{4,5} \leftarrow L_{2,3} \\ L_{6,7} \leftarrow L_{2,3} \end{cases}$$

comment: Transistor size of M1 and M8 to M11

$Finish \leftarrow$ **false**

while $Finish =$ **false**

do
- **CallSimulator**$(V_{\rm DD}, V_{\rm CM}, Clkc, |V_{\rm ip} - V_{\rm in}| < V_{\rm LSB})$
- **if** $|V_{\rm op} - V_{\rm on}| < 0.8V_{\rm DD}$ (i)
  - **then** increase I_1
  - **else if** $|V_{\rm op,n} - V_{\rm DD}| > V_{\rm LSB}$ (ii)
  - **then** $W_{8\sim 11} \leftarrow W_{8\sim 11} + \Delta W$
  - **else if** $V_{\rm op} < V_{\rm L}$ and $V_{\rm on} < V_{\rm L}$ (iii)
  - **then** decrease I_1
  - **else if** late or incorrect comparison result (iv)
  - **then** increase I_1
  - **else** (v)
    - update solution
    - decrease I_1
- **CheckEnd**$(Finish)$

I_1 to minimize power consumption of comparator. Because the search method of I_1 is binary search, the time complexity of this algorithm approximates $O(\log_2 |S|)$ where $|S| = \frac{W_{\rm max} - W_{\rm min}}{\Delta W}$.

After our tool accomplished comparator, SAR logic is also implemented easily according to the digital circuit design principles. Then the last sub-circuit is DAC. We separate it into three parts, capacitor array, S/H circuit and reference switches.

In moderate resolutions, the capacitance is usually dominant by mismatch effect instead of kT/C noise. It could be determined by the process data and the occupied area budget as an input option. For a S/H circuit, the good linearity performance and wide input bandwidth are highly required. To enhance them, we adopt a bootstrapped switch that is shown in Fig.6(a) [10]. When $Clks$ is high, M1 tracks the input signal. When $Clks$ is low, M1 is turned off by M2 and M3, then the capacitor array holds the input signal. To simplify the sizing procedure and reduce execution time, our program only searches the appropriate device sizes of Cs and M1 to M3. The other devices could be seen as digital circuits. The capacitor Cs could be realized by using large width and length transistor, $25\times W_{\rm min}$ and $25\times L_{\rm min}$ are reasonable values. The wider M1 may have wider input bandwidth and higher effective number of bits (ENOB), but oversize device may induce the serious parasitic capacitors and result in poor performances. The width of M2 and M3 should be larger than the $W_{\rm min}$ to eliminate the sampling time uncertainty. There are N different capacitors in the binary-weighted DAC of an N-bit SAR ADC. For different capacitance, the solution space of M1 has variant features those are shown in. Fig.6(b). Because the information we obtained from the simulation result is only success or failure, and we did not figure out that the reason of failure is induced by large capacitor (small M1) or small capacitor (large M1). Therefore, we have to try both smaller and larger M1 until the first successful result, and then we can start to minimize M1.

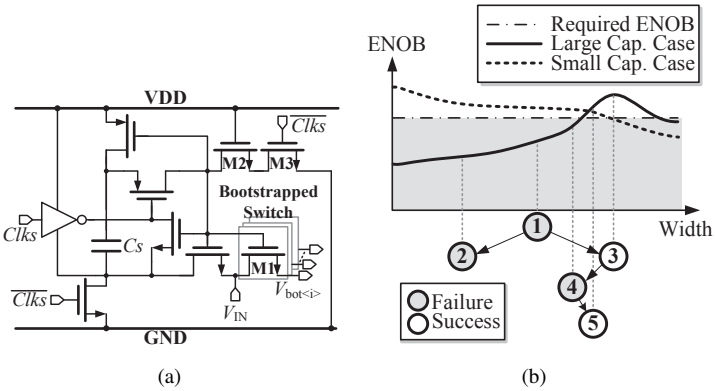

Fig. 6. (a) Bootstrap circuit and switching device. (b) Two cases with different capacitances, and data structure.

We summarize these considerations and propose an algorithm which suits this problem. The full searching space can be represented a binary tree. To reduce the searching space, we define some searching rules. (A) If we have not owned one success node, and node X not meets the requirement. We add left child of node X first to an execution queue, then the right child. (B) If node X meets the requirement, we clear the queue and add the left child of node X to the queue. (C) If we have owned one success node, and node X is a failure node, then we will only add the right child of node X to the queue. These rules are obeyed in Fig.6(b). Node 1 follows the rule (A). Rule (B) could be seen from node 3 and 4. Node 5 is reached since rule (C). This algorithm and data structure, binary tree and queue, are easy to implement. They also contain the operation of breadth-first search (BFS) before the first success node, and the operation of depth-first search (DFS) after the first node. The run time and the range of solution space are negative correlation, this leads to exhausted search when solution space is extremely narrow, and binary search in the other situations. Fortunately, because the capacitance of bit $i+1$ is two times larger than bit i, the solution of bit $i+1$ could be the upper bound of the searching range of the next bit. Then the searching space and the execution time are dramatically reduced. For a reasonable specification, the time complexity approximates $O(\log_2 |S|)$ again. This adaptive feature keeps the balance between execution time and solution quality.

The remaining part of DAC and the whole ADC is the reference switch. It provides $V_{\rm Ref}$ or $V_{\rm SS}$. To reduce occupied area, it is realized by an inverter instead of a bootstrapped switch. The size of inverter can be determined from the DAC signal incomplete settling issue. Our tool increases the size gradually from a reasonable initial value until this issue is eliminated. We give the size of M1 in Fig.6(a) as an initial value by two reasons. First, because the gate-source voltage ($V_{\rm gs}$) of inverter may be smaller than the $V_{\rm gs}$ of M1, the size of inverter should be slightly larger than M1. Second, we have minimized M1 in previous procedure. These reasons guarantee the initial value is as small as possible and still close to the final solution. The power consumption and the execution time are both reduced by this way. The DAC signal incomplete settling may induce missing code when the final value difference is small with signal crossing as shown in Fig.3. There are 2^N output codes for an N-bit ADC. However, we only need to examine two critical input voltages $V_{\rm CT1}$ and $V_{\rm CT2}$ as equation (2).

$$\begin{cases} 0 < V_{\rm CT1} - \frac{1}{4}V_{\rm swing} < V_{\rm LSB} \\ 0 < \frac{3}{4}V_{\rm swing} - V_{\rm CT2} < V_{\rm LSB} \end{cases} \quad (2)$$

TABLE I

I/O OF THE AUTOMATED SIZING TOOL

Input	Process library
	Simulator Path
	Resolution (N)
	Sampling Rate($f_{\rm s}$)
	Temperature
	Supply Voltage ($V_{\rm DD}$)
	Input Common Voltage($V_{\rm CM}$)
	Input Signal Swing($V_{\rm swing}$)
	Maximum Size($W_{\rm max}/L_{\rm max}$)
	Minimum Size($W_{\rm min}/L_{\rm min}$)
Output	Netlist Files
	Performance Report

TABLE II

COMPARISON WITH THE STATE-OF-THE-ART WORKS

	ISSCC'07 [11]	ISSCC'08 [12]	VLSI'08 [13]	VLSI'09 [14]	This Work
Architecture	SAR	SAR	Pipelined	SAR	SAR
Technology	90nm	90nm	90nm	0.13μm	0.11μm
Supply Voltage(V)	1	1	1.2	1.2	1.2
Sampling Rate(MS/s)	50	40	50	50	20
Resolution(bit)	9	9	9.4	10	10
ENOB(bit)	7.8	8.56	7.91	8.48	8.58
DNL(LSB)	<0.6	0.7/-0.45	+0.37/-0.38	+0.88/-1.00	+0.37/-0.75
INL(LSB)	<0.6	0.56/-0.65	+1.29/-0.88	+2.20/-2.09	+0.95/-0.66
Power(mW)	0.7	0.82	1.44	0.92	0.385
FoM(fJ/Conv.-step)	65	54	119	52	50

If DAC passes these tests, the largest non-linearity induced by DAC signal incomplete settling is almost eliminated. The sizing procedure is simple and effective because the number of test patterns will not grow exponentially with the resolution of ADC.

Based on the aforementioned sizing procedure, the whole SAR ADC has been accomplished to fulfill the required specifications with minimized power consumption. The comparator consumes much power because of the N times operation in each data conversion. Our sizing method reduces the power consumption significantly by minimizing the path current I_1. Additionally, a SAR ADC needs lots of buffers to control the reference switches, these buffers also consume much power. Minimizing the reference switches also relaxes the loading of the buffers and reduces power consumption dramatically. Besides, all the parameters in the sizing flow are calculated from equations or obtained from simulation results, and SAR ADC is also a proper architecture for technology migration. Therefore, this sizing methodology is process independent.

IV. EXPERIMENT RESULTS

The sizing tool has been developed in C++, the input parameters and output data are summarized in TABLE.I. It employs HSPICE as a standard simulator. Because the proposed algorithm reduces the execution time significantly, a proof-of-concept prototype was sized within only 15 minutes. The layout automation tool is not ready yet, so the layout was completed by hand. This prototype is fabricated in a 1P4M 0.11μm process. TABLE.II shows our measurement results and other handmade state-of-the-art works with similar resolution and sampling rate. The DNL is differential nonlinearity and the INL is integral nonlinearity. The DNL and INL results show that there is no missing code in our work. The ENOB is 9.2 bit at 5 MS/s. When the sampling rate increases to 20 MS/s, the ENOB is 8.58 bit. The power efficiency parameter, FOM, is calculated based on the well-known equation

$$FOM = \frac{Power}{2^{ENOB} \times \min(2 \times ERBW, f_{\rm s})} \tag{3}$$

where the $f_{\rm s}$ is sampling rate and the $ERBW$ is the effective resolution bandwidth. From the experiment results, the $ERBW$ is over Nyquist frequency ($0.5f_{\rm s}$). It indicates that the input bandwidth of sampling switch is sufficient. Although the sampling rate of our work is slower due to the less circuit techniques, the FOM is still better than others. These experiment results show that the proposed algorithm for such SAR ADC indeed improves the power efficiency and guarantees the linearity performance at the same time.

V. CONCLUSION

This paper presents a power-efficient sizing methodology for SAR ADCs. This method can reduce the power consumption significantly because the path current and the loading of buffers are minimized. Besides, the sizing method is independent of manufacturing process, and thus provides easy technology migration. We have also introduced an efficient searching algorithm and a tree-based data structure for the sizing flow. By this algorithm, the sizing time and searching space are dramatically reduced. A proof-of-concept prototype was sized within only 15 minutes. The measurement results show that the proposed sizing method can achieve high power efficiency and satisfactory performance for SAR ADCs.

ACKNOWLEDGMENT

The authors would like to thank Himax Technologies Inc. for partially sponsoring our research. This work was supported by National Science Council (NSC) of Taiwan under Grants NSC-98-2221-E-006-156-MY3.

REFERENCES

[1] S.-H. Cho, C.-K. Lee, J.-K. Kwon and S.-T. Ryu, "A 550-μW 10-b 40-MS/s SAR ADC With Multistep Addition-Only Digital Error Correction," *IEEE J. Solid-State Circuits*, vol.46, no.8, pp.1881-1892, Aug. 2011.

[2] E. Hjalmarson, "Studies on design automation of analog circuits - the design flow," Ph.D. dissertation, Linkopings University, Linkoping, Sweden, Dec. 2003.

[3] C. A. Makris and C. Toumazou, "Analog IC design automation: part II - automated circuit correction by qualitative reasoning," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 2, pp. 239-254, Feb. 1995.

[4] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums, "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. Computer-Aided Design*, vol. 19, no. 6, pp. 703-717, June 2000.

[5] G. V. der Plas, G. Debyser, F. Leyn, et al., "AMGIE - a synthesis environment for CMOS analog integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 9, pp. 1037-1058, Sept. 2001.

[6] G. Alpaydin, S. Balkir and G. Dundar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 3, pp. 240-252, June 2003.

[7] M. Allam, "Systematic Design Of A Successive Approximation Analog-To-Digital Converter," M.S. thesis, Dept. Electronic and Electrical Communications Cairo University, Giza, Egypt, 2008.

[8] S. Weaver, "Automated Synthesis of Analog to Digital Conversion," Ph.D. dissertation, Oregon State University, Oregon, Sep. 2010.

[9] J. L. McCreary and P. R. Gray, "All-MOS charge distribution analogto-digital conversion techniques-Part I," *IEEE J. Solid-State Circuits*, vol. SSC-10, no. 6, pp. 371-379, Dec. 1975.

[10] A.M. Abo and P.R. Gray, "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, May 1999.

[11] J. Craninckx and G. Van der Plas, "A 65 fJ/conversion-step 0-to-50 MS/s 0-to-0.7 mW 9b charge-sharing SAR ADC in 90 nm digital CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 246-247.

[12] V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Van der Plas, and J. Craninckx, "An 820μW 9b 40 MS/s noise-tolerant dynamic-SAR ADC in 90 nm digital CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2008, pp. 238-239.

[13] J. Hu, N. Dolev, and B. Murmann, "A 9.4-bit, 50-MS/s, 1.44-mW pipelined ADC using dynamic residue amplification," in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2008, pp. 216-217.

[14] C.-C. Liu, S.-J. Chang, G.-Y. Huang, and Y.-Z. Lin, "A 0.92mW 10-bit 50-MS/s SAR ADC in 0.13μm CMOS process," in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2009,pp. 236-237.