ACSP · Analog Circuits And Signal Processing

Taimur Rabuske
Jorge Fernandes

# Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications

Springer

# Analog Circuits and Signal Processing



More information about this series at http://www.springer.com/series/7381

Taimur Rabuske • Jorge Fernandes

# Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications

Springer

Taimur Rabuske
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
Lisboa, Portugal

Jorge Fernandes
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
Lisboa, Portugal

ISSN 1872-082X ISSN 2197-1854 (electronic)
Analog Circuits and Signal Processing
ISBN 978-3-319-39623-1 ISBN 978-3-319-39624-8 (eBook)
DOI 10.1007/978-3-319-39624-8

Library of Congress Control Number: 2016942569



Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

*To our families*

# Preface

The integrated successive-approximation-register (SAR) analog-to-digital converter (ADC) is known to present a remarkable energy efficiency. Additionally, the SAR ADC is a very scaling-friendly architecture, due to its highly digital and switching-intensive nature and its ability to effortlessly accommodate rail-to-rail signals without resorting to precision amplifiers. These characteristics have been boosting the popularity of SAR ADCs as process scaling reduces the transistor's intrinsic gain and supply voltages. The most basic form of an SAR ADC requires a voltage comparator, a digital controller, a track-and-hold (TH) circuitry, and a digital-to-analog converter (DAC).

In most of the reported designs, the DAC of an SAR ADC is capacitive and relies on the charge-redistribution (CR) principle. Alternatively, an SAR ADC that relies on the charge-sharing (CS) principle has been proposed more recently. The main advantages of the CS-ADC are that it is immune to inaccuracies in the reference buffer during the conversion and requires less demanding buffers for the input signal and the reference voltage. Additionally, the CS-ADC allows the use of nonlinear capacitors. On the other hand, the CS-ADC is less tolerant to noise and to the comparator offset, which limits the competitiveness of the architecture. Finally, the literature lacks a comprehensive analysis of the limiting factors of CS-ADCs.

The aim of this book is to fill the gap in knowledge that exists for CS-ADCs by providing an in-depth analysis of the architecture and quantifying its limiting factors. Finally, the insights gained through this analysis are applied to devise techniques that mitigate the critical drawbacks of the architecture and allow the design of energy-efficient SAR ADCs for low-voltage applications. These techniques are validated through the design of two SAR ADCs that operate at supply voltages down to 0.35 and 0.4 V. Experimental results show that the two reported ADCs present the best energy efficiency among ADCs designed in technologies down to 0.13 μm. Additionally, one of the presented prototypes is the first demonstration of SAR ADC that uses a DAC with very-nonlinear capacitors, benefitting from the improved capacitance density and matching characteristics of metal-oxide-semiconductor (MOS) capacitors.

Much of the material presented in this monograph originates from the work carried out by the first author for his Ph.D., at Universidade de Lisboa, Portugal. This work includes original research results that have been presented at international conferences (ESSCIRC, ISCAS, and ICECS, among others), published in Springer *Analog Integrated Circuits and Signal Processing, IEEE Transactions on Very Large Scale Integration (VLSI) Systems and IEEE Journal of Solid-State Circuits.*

The research reported in this book received contributions from many generous and gifted people. The authors would like to thank all colleagues and friends at INESC-ID Lisboa, Portugal, and UFSM, Brazil.

Finally, this work was financially supported by:

- FCT, Fundação para a Ciência e a Tecnologia (Portugal), under projects UID/CEC/50021/2013 and DISRUPTIVE (EXCL/EEI-ELC/0261/2012)
- CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico (Brazil), under the Ph.D. grant 201887/2011-8

Lisboa, Portugal — Taimur Rabuske
Lisboa, Portugal — Jorge Fernandes
April 2016

# Contents

# Chapter 1
# Introduction

**Contents**



## 1.1 Background

Before the Industrial Revolution, delivering a message to another person depended on rudimentary communication systems, such as horseback post riders and pigeon post. In that era, information was propagated in a speed that hardly exceeded a few kilometers per hour. The outcomes of these lagging methods are perplexing. For instance, the Battle of New Orleans, which was fought in January of 1815 in Louisiana, USA, took place after the Treaty of Ghent was signed between the United Kingdom and the USA 2 weeks before. The treaty was supposed to end the war and hostilities between the parties, but since the message did not arrive from Europe soon enough, more than 400 soldiers lost their life in that battle [1]. Only in 1844 these figures started to change significantly, after the invention of the telegraph and the consequent dissemination of this technology towards different parts of the world. The Mino–Owari earthquake, occurred in Japan in 1891, was reported to London on only 1 day of delay, despite the almost 6000 km separating the countries. Later, the introduction of the vacuum tube, in 1907, the transistor, in 1947, and the integrated circuit, in 1959, enabled further advancements in electronics, that enhanced the telecommunication systems at a similar rate. In the current days, electronics is an omnipresent element of culture, and information crosses the globe in a fraction of a second.



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_1

In the past five decades, scientists and engineers worked towards increasing the level of integration of transistors in integrated circuits and reducing their fabrication costs. In 2009, it was estimated that around one billion transistors were made for every person in the world every year, and that rate was growing fast [2]. Back in 1969, the first integrated circuit with silicon gate from Intel, the 1101 256-bit static random access memory (RAM) chip, comprised approximately 2000 transistors. To put into perspective, a modern graphics processing unit (GPU), such as the Nvidia's GK110, uses 7.1 billion transistors in a 28-nm process. The advancements in integration brought together improvements in the performance and cost of the integrated circuits. A modern central processing unit (CPU) chip fabricated in a 22-nm process runs with a clock over 4000 times faster, and each transistor consumes about 5000 times less energy when compared to the first microprocessor, the Intel 4004, introduced in 1971. Simultaneously, the price per transistor has dropped by a factor of about 50,000 [3].

The unprecedented evolution of speed, power efficiency, and level of integration in modern complementary metal-oxide-semiconductor (CMOS) technologies motivates the replacement of conventional analog-domain signal processing and storage by digital alternatives. On the other hand, the world is still analog, where the signals appear in continuous time and with continuous amplitude. Consequently, there is a need for interfacing circuits to bridge the two signal domains. The role is fulfilled by the digital-to-analog converter (DAC) and the analog-to-digital converter (ADC). The performance of these circuits is measured concerning the triad: power, speed, and resolution. The requirements of resolution and speed for an ADC or DAC are dictated by the system specifications and the characteristics of the signal to be processed. Commonly, power consumption should be simultaneously minimized, specially in battery-powered applications.

The search for improvements in the energy efficiency of an ADC is a persistent engineering problem and is a critical issue for many of the modern technology trends, such as sensor networks, wearable computing, and biomedical electronics. Additionally, many of these applications consume energy that is collected from the environment, e.g. through solar cells or radio-frequency (RF) harvesters. These energy scavengers can provide only a limited amount of power to the circuits and, in most of the cases, a limited voltage as well. A measure commonly employed to quantify the energetic efficiency of an ADC is the figure-of-merit (FOM), which is defined in (1.1), given that $P$ is the total power drained by the ADC, effective number of bits (ENOB) is a quantity that is related to the accuracy of the conversion, and $f_S$ is the sampling frequency.

$$\text{FOM} = \frac{P}{2^{\text{ENOB}} f_S}. \tag{1.1}$$

Many topologies of ADCs exist, covering different applications and requirements. The graph in Fig. 1.1 summarizes the ADCs published from 2000 to 2015 in International Solid-State Circuits Conference (ISSCC) and Symposium on VLSI Circuits (VLSIC), in terms of effective resolution, speed, FOM, and architecture [4].

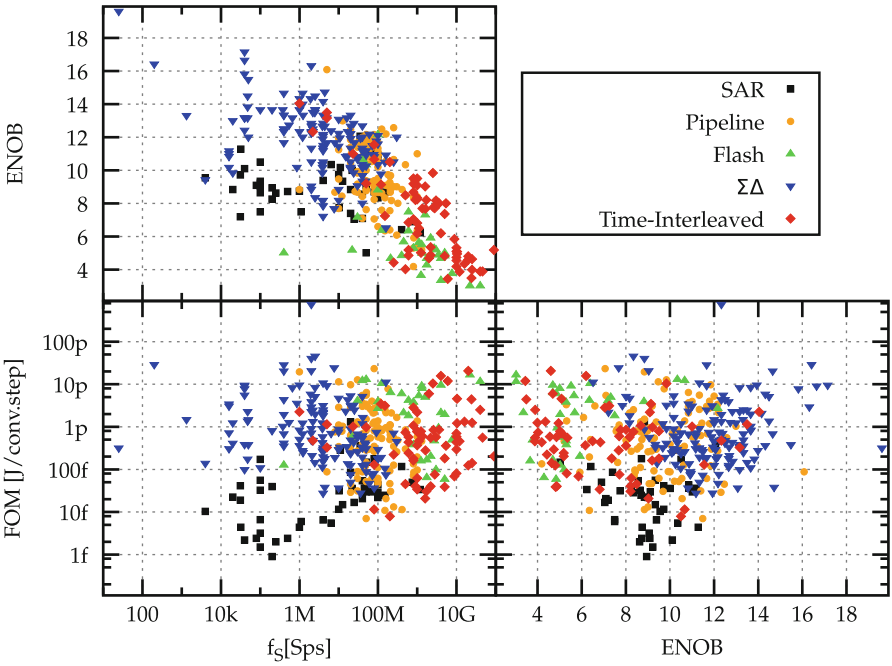



**Fig. 1.1** Summary of ADCs published in ISSCC and VLSIC from 2000 to 2015 [4]

The plot reveals a trade-off involving power, speed, and energetic efficiency. The fastest ADCs have lower resolutions and worse energy-efficiency; the ADCs with higher resolutions are slower and less energy-efficient; finally, the most energy-efficient ADCs are far from the leading-edge of speed or resolution.

The trade-off may be reasoned as follows. As the ADCs approach the leading-edge of conversion speed, the architecture to be employed requires a higher level of parallelism. It is more difficult to increase the resolution of highly parallel architectures, as the circuit complexity grows exponentially with resolution. Moreover, highly parallel architectures are naturally less energy efficient, because a larger number of components are activated simultaneously, draining more power for the same resolution. On the other extreme, the ADCs designed towards the leading-edge regarding resolution present lower sampling rate as they take advantage of oversampling to minimize the quantization noise. High-resolution ADCs are, in most of the cases, noise-limited, requiring large internal capacitances to assist in noise filtering, ultimately increasing the power consumption. Another important issue is that process scaling does not directly improve the noise characteristics of integrated circuits, and therefore, the benefit of scaling in noise-limited topologies is minor. As the supply voltage shrinks, it limits the input swing of the ADCs, consequently reducing the available dynamic range while making the ADCs less noise-tolerant. Hence, the resolution of modern integrated ADCs operating in the

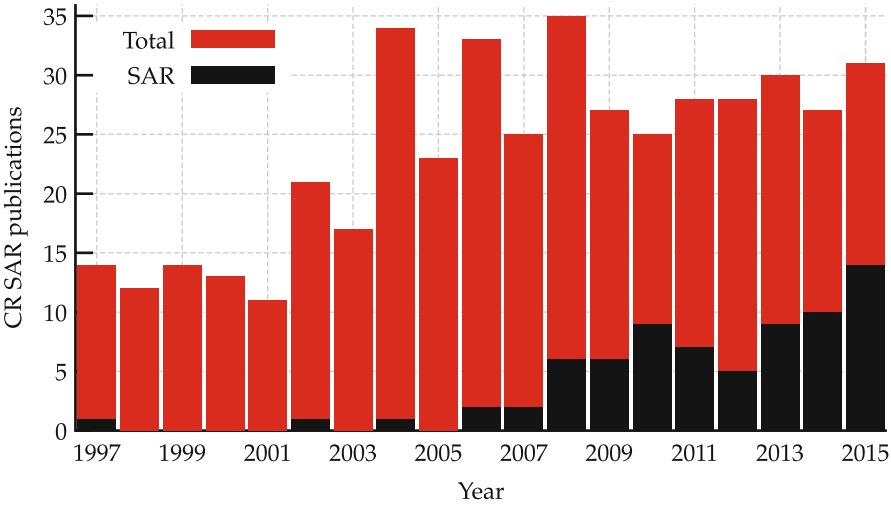



**Fig. 1.2** SAR ADCs (including time-interleaved SAR and pipelined-SAR) versus total of ADCs published in ISSCC and VLSIC per year since 1997 [4]

Nyquist range hardly exceeds 14 bits [5]. Fortuitously, the applications on both high-ends, high-resolution or high-speed, concern performance much more than energy efficiency and may tolerate an increased power consumption.

In the middle of the range in the plot of Fig. 1.1, we find the zone of moderate resolution (8–12 bits) and moderate speed (tens of kSps to tens of MSps) ADCs. According to the figure, these are the ADCs that provide the best energy efficiency. In the past decade, the successive approximation register (SAR) ADC became the dominant topology for this set of specifications while drawing a lot of attention from the scientific community. In Fig. 1.2 it is shown the number of SAR ADCs contrasted to the total number of ADCs published in ISSCC and VLSIC from 1997 to 2015 [4].

Before 2006, SAR ADCs appeared only occasionally in these conferences and most of the specialized publications. Starting in that same year, the number of published SAR ADCs grew consistently. In 2015, one-half of all the ADCs published in ISSCC and VLSIC were SAR or used SAR ADCs as building blocks (for example, in pipelined or time-interleaved architectures). Most of the success of the SAR ADC comes from its scaling-friendly nature, that enables the architecture to present outstanding energy efficiency when implemented in modern technologies. One of the aspects that makes the SAR ADC a befitting topology for deep sub-micron processes is that it does not resort to precision amplifiers, which are known to perform dreadfully in these technologies due to low intrinsic transistor gain. Moreover, since the SAR is based on switched capacitors, it benefits directly from a faster transistor speed, as it leads to also faster-switching speed.

The majority of the SAR ADCs rely on a DAC that uses the charge-redistribution (CR) principle [6], a very mature technique that has been successively employed

for the past four decades. Enhanced switching schemes based on the CR operating principle have been devised in the recent years, leading to further improvements in the energy efficiency of the architecture. A different approach for implementing the DAC of a SAR ADC was proposed in 2007 [7], and is known as the charge-sharing (CS)-ADC. The CS-based topology shares all the merits of SAR ADCs, such as requiring only a comparator as active circuit and operating following a highly digital procedure. Similarly, the CS-ADC also shares most of the advantages of the CR architecture, as it requires very similar circuit blocks. In some aspects, however, the architectures diverge, and the CS-ADC presents advantages and drawbacks when compared to the CR-ADC. For example, the CS-ADC presents an entirely passive operation after the precharge phase, obviating or at least greatly relaxing the reference buffer. The possibility of relaxing the buffer specifications makes the CS-ADC a good choice of topology in applications that operate under low-voltage supplies, where the design of the voltage buffers is burdened by the limited voltage headroom. On the other hand, the CS-ADC is not as tolerant to the comparator offset as the CR-ADC, requiring some sort of offset calibration in the comparator to achieve high linearity.

Nevertheless, the CS-ADC never achieved the same popularity of the CR-based architecture and had been employed in a very limited number of designs. Figure 1.3 shows the total number of CR- and CS-ADC designs presented in ISSCC and VLSIC, from 2007 (the year when the topology was introduced) to 2015 [4]. In these conferences, only four publications present CS-ADCs [7–10]. In addition, to the author's knowledge, there are only two other references of silicon-verified CS-ADCs published in other conferences and journals [11] and [12] excluding the outcomes of this book. Still, [12] reports the same design as [10].

Given the limited literature, and recognizing that all the works except for [12] were published in conferences with limited room for theory, the amount of available tutorial material is scarce. While most of the characteristics of the more mature

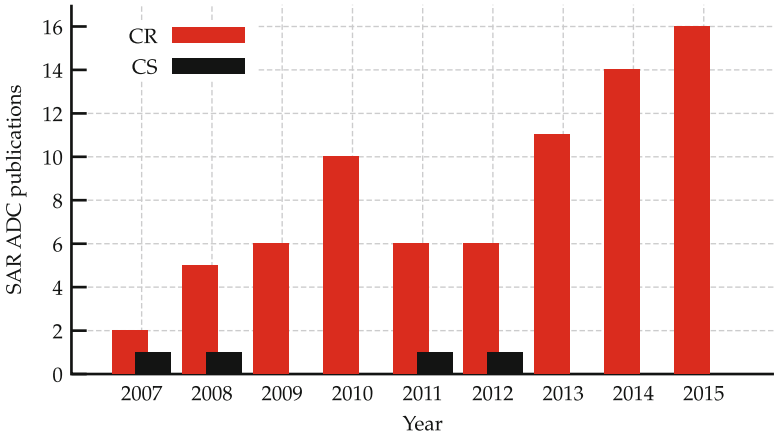



**Fig. 1.3** CS and CR SAR ADC publications per year [4]

CR-ADC have been extensively modeled and verified, including the power consumption [6, 13–19], linearity [20], noise [21], and statistical properties of resolution and yield [22], the same analyses are not available to the CS counterpart. Still, despite the reduced number of publications on CS-ADCs, there is no indication in literature that either of the architectures is universally superior. The CR-ADC is much more often used, probably because of the popularity and maturity of the architecture.

## 1.2 Research Goals

This book aims to fill the gap in knowledge that exists for CS-ADCs, by providing an in-depth analysis of the architecture and modeling of its limiting factors. The achieved understanding is exploited to improve the performance and to mitigate some critical pitfalls of the architecture. While pushing the limits of the topology, the performance of the CS-ADC is brought closer to that of state-of-the-art designs. Finally, the insights gained through the research are applied to the design of energy-efficient SAR ADCs for ultra-low voltage applications.

## 1.3 Book Outline

This book is organized into eight chapters. Following this introduction, Chap. 2 presents the engineering problem and shows the motivation for the design of low-voltage low-power (LVLP) ADCs. Fundamental aspects of data conversion such as sampling and quantization are also revisited, and the three most widely used search methods in Nyquist-rate ADCs are reviewed. This brief review is carried out to support the choice of the SAR topology for LVLP applications. In Chap. 3, the most relevant switching schemes employed in SAR ADCs are reviewed and compared. A summary shows that the CS scheme shows attractive features for LVLP applications. In Chap. 4, the most important sources of error in CS-ADC are analyzed and quantified. Since the performance of the voltage comparator employed in the ADC directly and strongly affects the overall ADC performance, Chap. 5 introduces a noise-aware technique for synthesis and optimization of efficient comparators. Chapter 6 introduces the local voltage boosting and the comparator offset background auto-zeroing techniques. These techniques are experimentally validated with the design of an 8-bit CS-ADC. Chapter 7 introduces a DAC topology based on the use of a metal-oxide-semiconductor capacitor (MOSCAP) as unitary capacitance cell and investigates its advantages in CS-ADCs over the conventional DAC based on linear capacitors. A 9-bit SAR ADC is designed using this DAC topology and the experimental results validate the technique. Finally, Chap. 8 summarizes and concludes this book.

## 1.4 Original Contributions

The main contributions of this book are as follows:

- Chapter 4 carries out an in-depth analysis of the limiting factors to the performance of the CS-ADC. The impact of several nonidealities is analyzed, including mismatch, parasitics, comparator offset, and noise.
- Chapter 5 introduces a noise-aware computational framework for synthesis and optimization of dynamic comparators. The framework employs genetic algorithms and simulation techniques that are traditionally used for RF circuits, such as periodic steady-state and cyclostationary noise analysis [23];
- Chapter 6 proposes two techniques that enable implementations of CS-ADC to present low power consumption and operate with low-voltage supplies: offset auto-zeroing for dynamic comparators and local voltage boosting to decrease the switches resistance. The mentioned techniques are experimentally validated through the implementation and measurements of an 8-bit ADC prototype [24].
- Additionally, a topology of a custom-designed digital successive approximation controller which relies on dynamic logic is proposed and validated through simulations and experimentally. This controller is employed in the fabricated ADC prototypes presented in this book.
- Finally, a topology of DAC for CS-ADCs that uses only MOSCAPs instead of linear capacitors is proposed and validated. MOSCAPs provide the best capacitance density among all the possible implementations of capacitors in integrated circuits. Additionally, it is demonstrated in Chap. 7 that the use of MOSCAPs greatly mitigates the drawbacks of the conventional CS-ADC, namely susceptibility to comparator offset and decreased tolerance to noise. This technique is validated with the design and experimental measurements (including characterization in a temperature chamber) of a 9-bit CS-ADC prototype [25, 26].
- The two presented ADCs are, to the authors' knowledge, the two ADCs with the best FOM for technologies down to 0.13 μm, and the two CS-based ADCs with the best FOM reported so far.

## 1.5 Final Remarks

During the research that led up to this book, a few other techniques were proposed. However, since we were not able to experimentally validate those ideas, they were not included in this book. In [27], a CS-ADC that utilizes an integrated step-down voltage converter to generate the precharge voltage for the DAC is reported. The integrated voltage conversion provides significant energy savings. In [28], a 12-bit CS-ADC that employs background self-calibration of the DAC mismatches is reported. The mismatches are calibrated using dynamic body-biasing. Thus, the bodies of the MOSCAPs employed in the DAC are properly biased to compensate for mismatches. The techniques proposed in these two publications were validated through circuit-level simulations.

## References


1. G. Clark, *A Farewell to Alms: A Brief Economic History of the World*. The Princeton Economic History of the Western World (Princeton University Press, Princeton, 2007). ISBN:9781400827817 (cit. on p. 1)
2. S. Franssila, *Introduction to Microfabrication*, 2nd edn. (Wiley, New York, 2010). ISBN: 9781119991892. doi:10.1002/9781119990413 (cit. on p. 2)
3. Intel Corporation, Fun facts. October 2014 (Online). Available: http://www.intel.com/content/dam/www/public/us/en/documents/corporate-information/history-moores-law-fun-facts-factsheet.pdf (cit. on p. 2)
4. B. Murmann, ADC performance survey 1997–2015. (Online). Available: http://web.stanford.edu/~murmann/adcsurvey.html (visited on 02/2016) (cit. on pp. 2, 3, 4, and 5)
5. W. Liu, Low-power high-performance SAR ADC design with digital calibration techniques. Ph.D. thesis, University of Illinois at Urbana-Champaign, 2010 (cit. on p. 4)
6. J. McCreary, P. Gray, All-MOS charge redistribution analog-to-digital conversion techniques. I. IEEE J. Solid State Circuits **10**(6), 371–379 (1975). doi:10.1109/JSSC.1975.1050629 (cit. on pp. 4 and 6)
7. J. Craninckx, G. van der Plas, A 65fJ/conversion-step 0-to-50MS/s 0- to-0.7mW 9b charge-sharing SAR ADC in 90nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2007), pp. 246–600. doi:10.1109/ISSCC.2007.373386 (cit. on p. 5)
8. V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Van der Plas, J. Craninckx, An 820$\mu$w 9b 40MS/s noise-tolerant dynamic-SAR ADC in 90 nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2008), pp. 238–610. doi:10.1109/ISSCC.2008.4523145
9. J.-H. Tsai, Y.-J. Chen, M.-H. Shen, P.-C. Huang, 1-V, 8b, 40MS/s, 113$\mu$W charge-recycling SAR ADC with a 14$\mu$W asynchronous controller, in *IEEE Symposium on VLSI Circuits* (2011), pp. 264–265
10. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70dB DR 10b 0-to-80MS/s current-integrating SAR ADC with adaptive dynamic range, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2012), pp. 470–472. doi:10.1109/ISSCC.2012.6177095 (cit. on p. 5)
11. Y.-J. Chen, J.-H. Tsai, M.-H. Shen, P.-C. Huang, A 1-V 8-bit 100kS/s- to-4MS/s asynchronous SAR ADC with 46fJ/conv.-step, in *International Symposium on VLSI Design, Automation and Test* (IEEE, New York, 2011), pp. 1–4. doi:10.1109/VDAT.2011.5783590 (cit. on p. 5)
12. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70 dB DR 10b 0-to-80 MS/s current-integrating SAR ADC with adaptive dynamic range. IEEE J. Solid State Circuits **49**(5), 1173–1183 (2014). doi:10.1109/JSSC.2014.2309086 (cit. on p. 5)
13. V. Hariprasath, J. Guerber, S.-H. Lee, U.-K. Moon, Merged capacitor switching based SAR ADC with highest switching energy-efficiency. Electron. Lett. **46**(9), 620 (2010). doi:10.1049/el.2010.0706 (cit. on p. 6)
14. Y. Zhu, C.-H. Chan, U.-F. Chio, S.-W. Sin, S.-P. U, R.P. Martins, F. Maloberti, A 10-bit 100-MS/s reference-free SAR ADC in 90 nm CMOS. IEEE J. Solid State Circuits **45**(6), 1111–1121 (2010). doi:10.1109/JSSC.2010.2048498
15. C.-C. Liu, S.-J. Chang, G.-Y. Huang, Y.-Z. Lin, A 10-bit 50-MS/s SAR ADC with a monotonic capacitor switching procedure. IEEE J. Solid State Circuits **45**(4), 731–740 (2010). doi:10.1109/JSSC.2010.2042254
16. C. Yuan, Y. Lam, Low-energy and area-efficient tri-level switching scheme for SAR ADC. Electron. Lett. **48**(9), 482 (2012). doi:10.1049/el.2011.4001
17. L. Sun, K.-P. Pun, W.-T. Ng, Capacitive digital-to-analogue converters with least significant bit down in differential successive approximation register ADCs. J. Eng. 1–4 (2014). doi:10.1049/joe.2013.0219

18. X. Song, Y. Xiao, Z. Zhu, Vcm-based monotonic capacitor switching scheme for SAR ADC. Electron. Lett. **49**(5), 327–329 (2013). doi:10.1049/el.2012.3332
19. E. Rahimi, M. Yavari, Energy-efficient high-accuracy switching method for SAR ADCs. Electron. Lett. **50**(7), 499–501 (2014). doi:10.1049/el.2013.3451 (cit. on p. 6)
20. M. Saberi, R. Lotfi, K. Mafinezhad, W.A. Serdijn, Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation ADCs. IEEE Trans. Circuits Syst. I Regul. Pap. **58**(8), 1736–1748 (2011). doi:10.1109/TCSI.2011.2107214 (cit. on p. 6)
21. W.P. Zhang, X. Tong, Noise modeling and analysis of SAR ADCs. IEEE Trans. Very Large Scale Integr. Syst. **23**(12), 2922–2930 (2015). doi:10.1109/TVLSI.2014.2379613 (cit. on p. 6)
22. J.A. Fredenburg, M.P. Flynn, Statistical analysis of ENOB and yield in binary weighted ADCs and DACS with random element mismatch. IEEE Trans. Circuits Syst. I Regul. Pap. **59**(7), 1396–1408 (2012). doi:10.1109/TCSI.2011.2177006 (cit. on p. 6)
23. T. Rabuske, J. Fernandes, Noise-aware simulation-based sizing and optimization of clocked comparators. Analog Integr. Circ. Sig. Process **81**(3), 723–728 (2014). doi:10.1007/s10470-014-0428-4 (cit. on p. 7)
24. T. Rabuske, F. Rabuske, J. Fernandes, C. Rodrigues, An 8-bit 0.35- V 5.04-fJ/ conversion-step SAR ADC with background self-calibration of comparator offset. IEEE Trans. Very Large Scale Integr. Syst. (VLSI) **23**(7), 1301–1307 (2015). doi:10.1109/TVLSI.2014.2337236 (cit. on p. 7)
25. T. Rabuske, J. Fernandes, A 9-b 0.4-V charge-mode SAR ADC with 1.6-V input swing and a MOSCAP-only DAC, in *Proceedings of European Solid-State Circuits Conference* (IEEE, New York, 2015), pp. 311–314. ISBN:978-1-4673-7470-5. doi:10.1109/ESSCIRC.2015.7313889 (cit. on p. 7)
26. T. Rabuske, J. Fernandes, A SAR ADC With a MOSCAP-DAC, IEEE J. Solid-State Circuits, **51**(6), 1410–1422 (2016). doi:10.1109/JSSC.2016.2548486 (cit. on p. 7)
27. T. Rabuske, J. Fernandes, F. Rabuske, C. Rodrigues, M.B. dos Santos, A self-calibrated 10-bit 1 MSps SAR ADC with reduced-voltage charge- sharing DAC, in *IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE, New York, 2013), pp. 2452–2455. doi:10.1109/ISCAS.2013.6572375 (cit. on p. 7)
28. T. Rabuske, J. Fernandes, A 12-bit SAR ADC with background self-calibration based on a MOSCAP-DAC with dynamic body-biasing, in *IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE, New York, 2016), pp. 1482–1485 (cit. on p. 7)

# Chapter 2
# ADCs for Low-Voltage Low-Power Applications

**Contents**



## 2.1 Introduction

There is an assortment of emerging applications for which energy consumption is a key metric. Implantable biomedical systems, for instance, are of keen interest in a variety of medical areas, as they can be utilized to treat disorders such as epileptic seizures and Parkinson's disease [1]. These units also led to promising results in rehabilitative prosthetic devices to help disabilities such as deafness and blindness [2]. Another example of highly energy-constrained application is a wireless sensor network (WSN), that relies on the inter-cooperation of many nodes, distributed throughout an area of interest, to sense environmental parameters. This approach to computational networking has been shown to enable a viable solution for health, structural, industrial, military, and habitat monitoring [3–7].

Minimizing the power consumption of all the circuits used in these applications is critical, because it is usually impractical to supply by wires the energy that these devices need to operate. Also, the size of the units is frequently prone to serious limitations, to allow them to be conveniently placed or implanted, consequently restricting the volume of the batteries. As the energy storage in a battery is



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_2

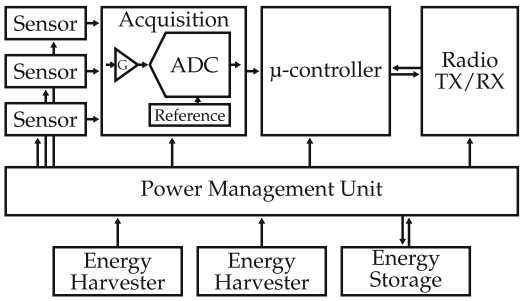



**Fig. 2.1** Block diagram of a sensor node in a WSN

proportional to its volume, the available energy is very limited. Another aggravating factor is that, in many applications, it is inconvenient to replace the battery of the devices: in the case of medical devices, a battery replacement may require a surgical procedure; in the case of WSNs, the nodes may be in harsh environments or be mobile.[1]

For WSNs and many biomedical devices, each unit is a piece of hardware that performs sensing, computation, and communication. They consist of sensors, a data acquisition block, a micro-controller, radio communication circuitry, a power management unit (PMU), and power sources. A block diagram of a WSN node is shown in Fig. 2.1. The sensor converts an environmental parameter such as temperature, humidity, or pH to an electrical signal delivered as voltage or current. Then, the data acquisition block performs preprocessing and amplification on the signal, which is finally converted to the digital form by an ADC. This conditioned signal is processed and stored in the micro-controller. The controller also provides some level of intelligence to the sensor node, such as time scheduling and data compressing. Finally, the RF block allows the node to communicate with the neighboring nodes and the base-station, depending on the network topology. The energy sunk by these modules is provided by the energy sources and is managed by the PMU.

Fortunately, energy harvesting technology has emerged as a promising solution to enable self-sustainable devices. The ultimate goal in this context is that the energy consumption is sufficiently low that can be totally harvested from the environment, presenting a theoretical unlimited lifetime. Still, a rechargeable battery or a super-capacitor is employed to bear the power peaks and to provide a backup source when the environment is unable to suffice the energy required by the unit. Table 2.1 shows power densities of four forms of energy suitable to be harvested in an approximate scale [9, 10]. The outdoor solar energy presents the highest density and RF presents the lowest. While the values provided in Table 2.1 are valid for specific test conditions, the environmental conditions may be subject to large variations and

[1] Take, for example, the ZebraNet [8], which is a habitat monitoring system that requires zebras to wear global positioning system (GPS) collars. The specimens would have to be re-captured for every battery replacement.

**Table 2.1** Power densities for different forms of power harvesting [9, 10]

| Source | Condition | Power density |
|---|---|---|
| Solar | Direct sunlight | 7.5 mW/cm$^2$ |
| | Indoor | 100 μW/cm$^2$ |
| Thermal | $\Delta T = 5\,°C$ | 60 μW/cm$^2$ |
| | $\Delta T = 10\,°C$ | 135 μW/cm$^2$ |
| Vibrational | 1 m/s$^2$ (vibrations in the Hz-range) | 4 μW/cm$^3$ |
| RF | Separated 2 m from the RF transmitter | 14 mW/cm$^2$ |
| | Unless near a RF transmitter | <1 μW/cm$^2$ |

greatly affect the values of available power. For example, the hours of direct sunlight may be severely reduced during the winter, and a rainy day may leave the solar cell with no direct sunlight at all. These factors have to be accounted for while dimensioning the power circuit of the node, and remind us that it is beneficial to minimize the power consumption of the circuit blocks employed in the system. In addition, most of the forms of energy harvesting generate a very limited voltage. A typical single solar cell can output 400–500 mV, depending on lighting conditions, while thermoelectric generators are able to generate 150–250 mV [11]. Therefore, it is necessary that the PMU performs voltage conversion and regulation, to provide a convenient and reliable voltage for the remaining of the circuit blocks in the sensor node.

Some topologies of voltage-converters are most efficient when the values of the input and output voltage are close. In that case, it is of great interest that the voltage supply required by the circuits in the sensor node is also low, to be close to the low output voltage of the energy harvesting sources. Still, even if the voltage-conversion efficiency does not benefit from input/output voltages proximity, it is beneficial from the energy standpoint to supply the circuits with low voltage. The main outcomes of lowering the supply voltage in most of the circuits are reductions in the operation speed and the power consumption. The former is not as critical in LVLP applications as it is in other scenarios. This category of systems generally requires low operation speeds, because the signals to be sensed and processed vary slowly, reaching time constants of seconds or minutes in biomedical and environment monitoring systems [12]. Regarding power consumption, in switching-intensive circuits (as it is the case for CMOS digital gates and switched-capacitor circuits) the energy is proportional to the capacitance being switched and to the square of the supply voltage, as in

$$E_{DD} \propto C_{switching} {V_{DD}}^2. \tag{2.1}$$

It is also pointed out in [13] that the leakage current is strongly correlated and decreases exponentially with the supply voltage. In other words, the usage of ultra-

low voltage supplies helps in reducing both the static and dynamic components of power consumption. Also, it becomes evident that, as long as the timing requirements are fulfilled, the LVLP applications may exploit the usage of very-low voltage supplies in order to increase the battery lifetime or to relax the harvesting specifications. It is also advantageous that all the circuits in the system operate at the same supply voltage, so that there is no need for the power unit to generate multiple voltages, minimizing losses.

This scenario, summed up with the projection of unceasing decrease in the supply voltage of leading edge fabrication technologies [14], has motivated extensive research in LVLP designs and techniques. Many implementations of different classes of circuits have been reported to work with extremely low supply voltages, including digital processors [15] and memories [16], analog amplifiers [17] and filters [18], RF front-ends [19, 20], and ADCs [21–25].

ADCs are mandatory blocks in WSN and in many biomedical implantable/-portable systems and other applications that require LVLP operation. In the case of WSN, the requirements for the ADCs are in most cases low sampling rate (kSps range), moderate resolution (8–12 bits), and minimum power consumption. This book is focused on the design of ADCs in this class. In the remaining of this chapter, basic concepts in data conversion are revisited. The three most commonly used search methods for ADCs operating in the Nyquist frequency range are also reviewed, to determine which one is most befitting for the aimed specifications.

## 2.2 Sampling and Quantization

ADCs convert signals from the analog domain (continuous time and continuous amplitude) into the digital domain (discrete time and discrete amplitude). Thus, the operation of the ADC may be split up into two different processes: sampling (time discretization) and quantization (amplitude discretization).

In the context of ADCs, sampling is a simple process that is frequently implemented by a track-and-hold (TH) circuit. The TH provides two operation modes that are selected according to the polarity of a clock signal $\phi$, as depicted in Fig. 2.2. While in the track mode, the TH output follows (tracks) the input signal; when the circuit is switched to the hold mode, the output is kept steady (held), so that the amplitude can be quantized independently of changes in the input. The period of the clock signal that controls the TH dictates the sampling period, and its inverse is called sampling rate or sampling frequency.

Quantization, on the other hand, is a more complicated task that usually involves more complex circuitry. The quantizer is responsible for searching the output code that better represents the analog sample. However, since the number of possibilities of output codes is finite, the circuit introduces quantization error, that is the difference between the decision level and the analog sample. In an ideal quantizer, the distance between decision levels corresponds to 1 least-significant bit (LSB), and the magnitude of the quantization error is always lower than or equal to half

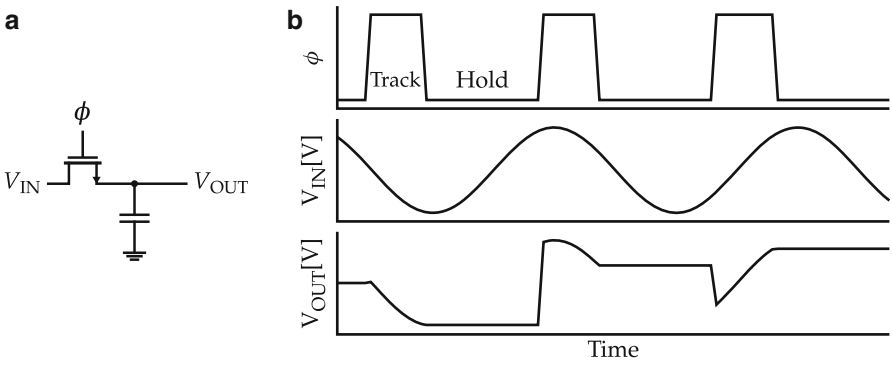



**Fig. 2.2** Simple TH circuit: (**a**) implementation and (**b**) corresponding waveforms

LSB. Most of the practical quantizers provide a binary digital output. Thus, an ideal $B$-bit quantizer divides the full input range into $2^B$ decision levels. The sampling and quantization processes are illustrated in Fig. 2.3, for ideal quantizers of 2, 3, and 4 bits, respectively. Note that the quantization error is always bounded to $-\frac{1}{2}$ LSB and $\frac{1}{2}$ LSB. However, since one LSB corresponds to a smaller range of the input signal as the resolution increases, the quantization error is inversely proportional to the resolution.

In sampling theory, the Nyquist frequency is defined as half of the sampling frequency ($f_S/2$). According to the relationship between the Nyquist frequency and the maximum bandwidth of the input signal that the device can process, the ADCs can be laid down in two major categories. The Nyquist-rate ADCs (frequently abbreviated as Nyquist ADCs) are able to process signals with a bandwidth of half the sampling frequency. The oversampled ADCs, on the other hand, can only process signals with a bandwidth significantly lower than the Nyquist frequency as they sacrifice input bandwidth to improve resolution and reduce noise. As a result, the implementations of quantizers are different for Nyquist and oversampled ADCs. In the next section, the main methods that are used in the quantizers of Nyquist ADCs to search the correct decision level are reviewed. The oversampled ADCs are out of the scope of this book, and will not be further investigated.

## 2.3 Search Methods for Nyquist ADCs

Most of the implementations of Nyquist ADCs rely on one out of the three following methods to search for the correct output level: direct search, binary search, and pipelined binary search.

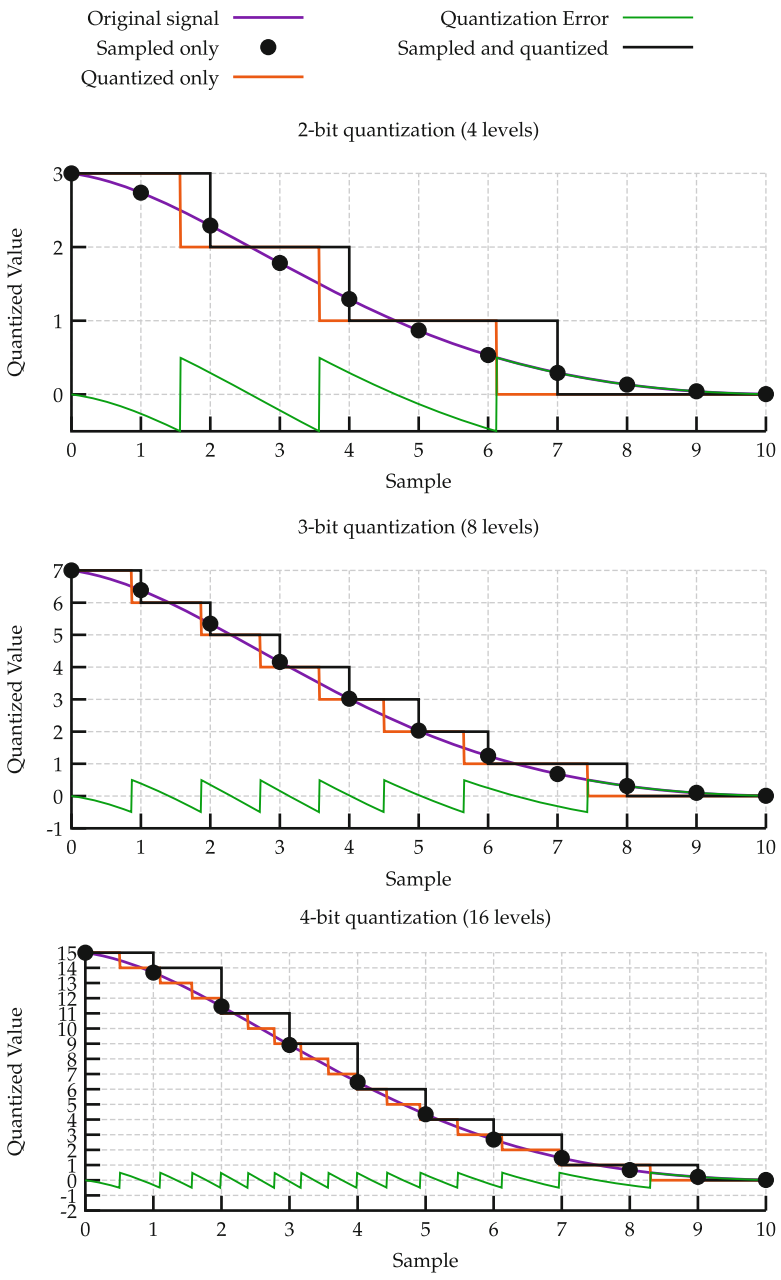



**Fig. 2.3** Sampling and quantization processes for 3, 4, and 5-bit quantizers

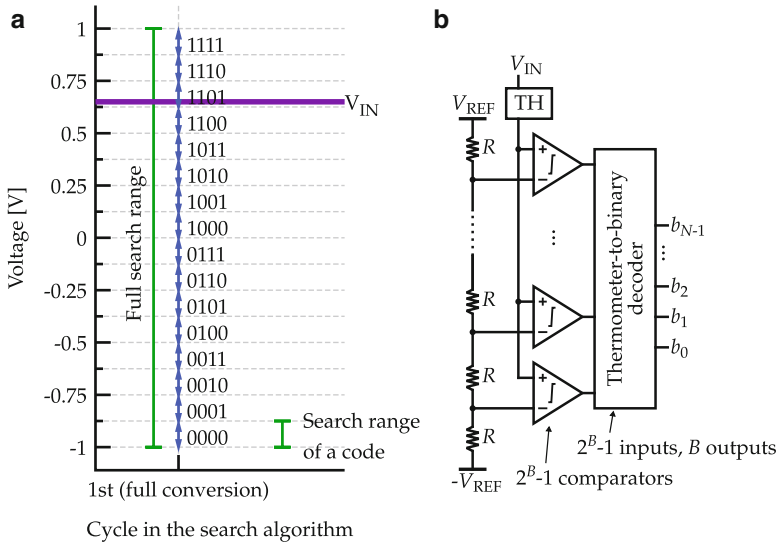



**Fig. 2.4** Direct search method: (**a**) illustrative waveforms and (**b**) example of Flash ADC implementation

### *2.3.1 Direct Search (Flash ADCs)*

In the direct search, the proper output code is found through the simultaneous comparison with all the decision levels covered by the ADC. Then, the comparison results are mapped into a binary representation of the sampled input. If the input falls between the $i$-th and the $(i+1)$-th decision level, the output is $i$ in the digital form. If the input lies below or above all the decision levels, the ADC output returns 00...0 or 11...1, respectively. The operation is exemplified in Fig. 2.4a. In the example, the input voltage range of 2 V ($-1$ to $+1$ V) is mapped into 16 output possibilities, representing a 4-bit quantizer. Since the input voltage of 0.65 V is greater than the decision level of the code 1101 (0.625 V) but smaller than the decision level of the code 1110 (0.75 V), it is converted to the binary code 1101.

The flash ADC falls into the category of direct search ADC. A typical implementation of the flash ADC is shown in Fig. 2.4b. The TH circuit samples the input and holds it steady during the quantization. The resistive ladder generates the decision levels in the voltage domain, which are fed to the comparators and act as reference voltages. Ideally, with a negligible offset voltage, the comparator outputs form a thermometer code that indicates where the input voltage sits. This code is transformed into a binary word by the decoder. In practice, the decoder also includes bubble-removal logic to deal with the comparator offsets.

Since the entire conversion occurs within a single cycle, this search method is very time-efficient and is frequently employed in high-frequency applications. On the other hand, the high level of parallelism requires that many components

are activated simultaneously, and that raises the power consumption significantly. Moreover, the hardware complexity grows exponentially ($\sim 2^B$) with the number of bits $B$, becoming impractical for applications that require higher resolutions.

### 2.3.2 *Binary Search (SAR ADCs)*

The binary search method uses multiple cycles to find the correct output, trading off operation frequency for a reduction in hardware complexity. One bit is resolved in each cycle, and this allows for the search range to be halved for the following cycle. Figure 2.5a depicts the process for a 4-bit quantizer, using an input signal of 0.65 V. The search range for the first cycle is the full ADC input range ($-1$ to $+1$ V), and 0 V is used as a reference to evaluate if the input sits in the upper or lower half. Since 0.65 V is larger than 0 V, the search is bounded to the upper half in the second cycle. Now, 0.5 V is used as a reference to decide if the input sits in the third or fourth quarter of the full range. The method proceeds similarly for the two following cycles, and the ADC outputs the code 1101.

The SAR ADC falls into the category of binary search ADC. A typical implementation of the SAR ADC is shown in Fig. 2.5b. The TH circuit samples the input and holds steady during the quantization. The SAR controller starts feeding the DAC with the digital code 100. . .0, that sets the DAC output to half of the reference voltage. According to the comparison results, the SAR controller adjusts the digital word that feeds the DAC, bringing the difference between $V_{\text{hold}}$ and $V_{\text{DAC}}$ towards zero.

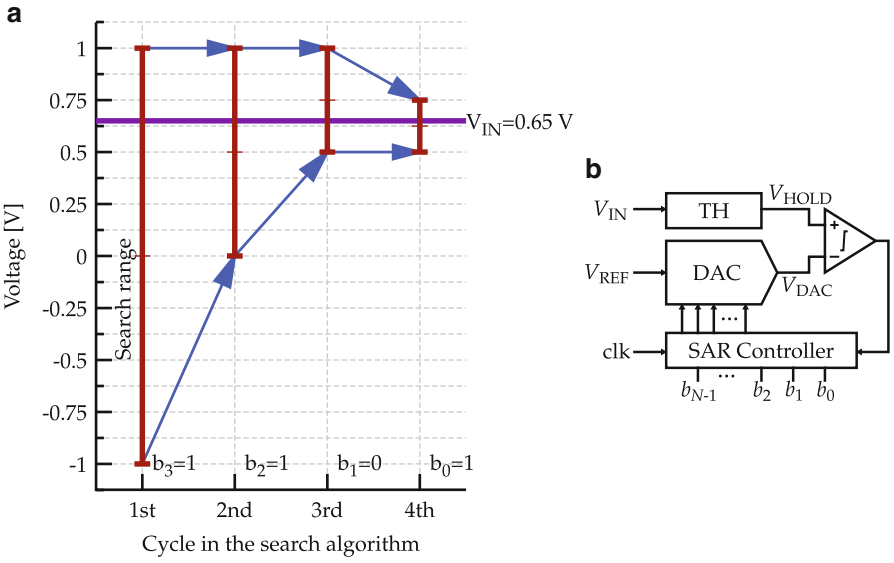



**Fig. 2.5** Binary search method: (**a**) illustrative waveforms and (**b**) example implementation of the SAR ADC

As one cycle is required for each bit of resolution, this search method is less time-efficient than the direct search. On the other hand, the number of comparisons required in the iterative search method is a linear function of the resolution ($B$ comparisons for $B$ bits), instead of exponential, leading to a better energy-efficiency. It is important to note that the performance of the SAR ADCs is very dependent on the performance of the DAC.

### 2.3.3 *Pipelined Binary Search (Pipeline ADCs)*

The binary search may be complemented with pipelining to speed up the conversion process. Different from the previous search method, this type of quantizer does not require that all the bits are resolved before starting a new conversion. Instead, after one bit is resolved, the residue voltage is amplified and sampled by the next stage, at the same time as that stage is resolving the next bit of the previously sampled signal. Then, as this residue is propagated along the chain, the converter uses the front-end stage to sample another input. The outcome is a latency equal to the number of stages of the converter.

Figure 2.6 illustrates the search process in a pipelined architecture. In the first cycle, full input range is used, and the middle decision-code is used as a reference to decide if the input lays in the upper or lower half of the range. Then, the residue signal is produced by subtracting the ideal bit weight from the sampled input. Finally, the residue is multiplied by two, restoring the full signal swing, before

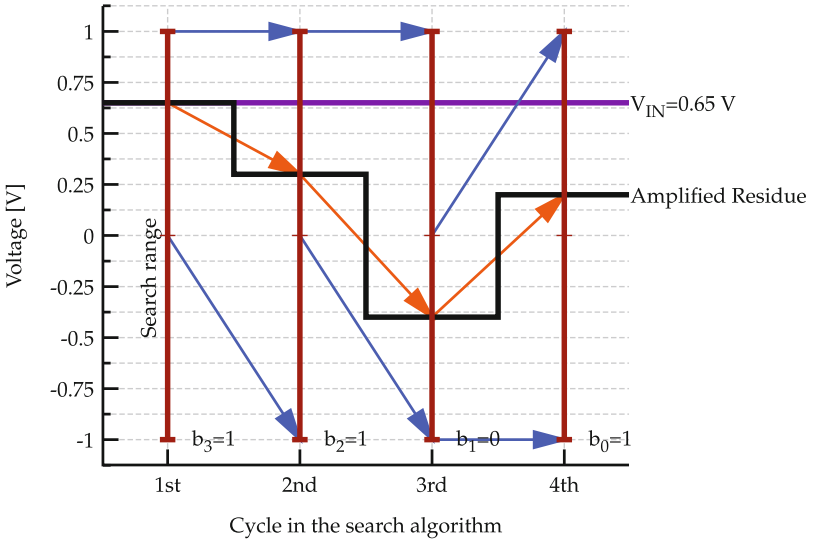



**Fig. 2.6** Illustrative waveforms of the pipelined binary search method

being fed to the next stage of the pipeline (2nd cycle in the figure). The process continues until all the bits are resolved. This way, only the quantization error is amplified and propagated from stage to stage, and the quantized values are stored in the digital domain. Thus, the signal range stays constant for all the stages and the decision levels used as reference are always the same, significantly simplifying their generation. In practice, the stages in the back-end of the chain may also be designed with much-relaxed noise and matching performances. The figure depicts a 1-bit per stage converter, but any stage-resolution may be employed.

In the 4-bit example, the input voltage of 0.65 V is larger than the middle-range (0 V), and the result of the most-significant bit (MSB) is 1. The quantization error is amplified by two and the evaluated voltage on the second cycle is 0.3 V. Again, this value is larger than 0 V, and $b_2$ is also 1. As the process repeats for the other bits, at the end of the conversion, 1101 is available as the digital output.

A traditional pipeline ADC is shown in Fig. 2.7. Each stage comprises a TH, a sub-ADC, a sub-DAC, a subtractor, and an amplifier. The resolution of the stage is arbitrary and affects the resolutions of the ADC and the DAC and the amplifier gain. The stage operates as follows. Initially, the TH samples the input and holds it during the quantization. The sub-ADC converts the signal into the digital domain and the output is fed into the sub-DAC, which brings the signal back to the analog domain. The output of the DAC is subtracted from the sampled signal, and the result is the quantization error. Finally, the quantization error is amplified by $2^K$, where $K$ is the stage's number of bits, and passed to the following stage. At the back-end of the cascaded stages, a simple ADC is employed, as there is no need to propagate the quantization error any further. In practice, DAC, subtractor, amplifier, and TH are brought together in a circuit known as multiplying digital-to-analog converter (MDAC), greatly simplifying the implementation of pipeline ADCs.

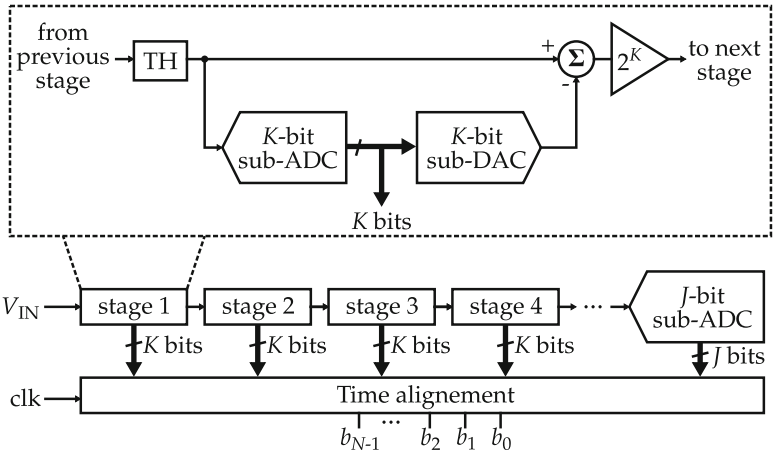



**Fig. 2.7** Example of Pipeline ADC implementation

**Table 2.2** Summary of search methods for Nyquist ADCs and LVLP requirements

| | Direct (Flash ADC) | Binary (SAR ADC) | Pipelined binary (Pipeline ADC) | Common specs for LVLP applications |
|---|---|---|---|---|
| Speed | High | Low | Medium | Low |
| Power | High | Low | Medium | Minimize |
| Resolution (b) | $\leq$8 | $\leq$12 | $\leq$16 | 8–12 |

The pipeline arrangement of this search method enables to improve the sampling frequency when compared to the nonpipelined binary search method. However, it is important to note that the architecture requires precision amplifiers for the residue. As the technology scaling restricts the design of amplifiers, it also hinders the implementation of pipeline ADCs. Similarly, the design of pipeline ADCs is more difficult at reduced supply voltages.

### *2.3.4 Summary*

The three search methods for Nyquist ADCs are briefly summarized in Table 2.2, together with common specifications for LVLP applications. As the requirements of sampling frequency and resolution in these applications are not too demanding, while power is the main concern, the characteristics of the binary-search method (SAR ADC) fit best in the given requirements. There may be cases where the resolution of a SAR ADC is not sufficient to satisfy the system needs. In those cases, a $\Delta\Sigma$ ADC would provide a good alternative, at the cost of a worse energetic efficiency. In the remaining of this chapter, the implementation details and characteristics of the SAR ADC are presented in greater detail.

## 2.4 The SAR ADC

The first implementations of ADCs based on successive approximation date back from the 1940s [26]. In addition, the first commercial ADC was a SAR implemented with vacuum tubes only, providing 11 bits of resolution and 50 kSps of sampling rate [27]. Denominated DATRAC, it was offered in a 48 cm $\times$ 66 cm $\times$ 38 cm case weighting almost 70 kg, dissipated 500 W and was sold for more than US$ 8000 [28]. Modern SAR ADCs, on the other hand, are notable for their energy efficiency and can fit in a small area footprint inside low-cost CMOS chips. Most of the modern SAR ADCs present only dynamic power consumption, resulting from the use of CMOS logic and the absence of preamplifiers or static latch in the comparator. Therefore, the power consumption is linearly correlated to the sampling frequency if leakage is negligible.

As the process scaling advances, the transistors become faster but present a lower intrinsic gain. As a consequence, the quality of switching improves, and the quality of amplification degrades. The SAR ADCs present a switching-intensive nature, while precluding the use of amplifiers. These characteristics cause the SAR ADC to be the ADC topology that benefits most from technology scaling. Moreover, the absence of amplifiers makes the SAR ADC able to treat rail-to-rail input swings effortlessly, relaxing the noise specifications of the DAC and the comparator. According to Fig. 2.5b, four sub-blocks are necessary to properly implement a SAR ADC: a SAR controller, a comparator, a DAC, and a TH.

The SAR controller often does not pose significant design challenges, as the binary search algorithm can be easily translated into logic gates. A flow-chart of the SAR controller operation for a conventional CR-ADC is shown in Fig. 2.8. The performance metrics of the SAR controller are speed and energy efficiency, and those have a direct impact on the overall speed and power of the ADC. Many recent designs employ full-custom controllers implemented in transistor-level rather than in gate-level to push further the performance of the controller [29].

The comparators employed in most of recent SAR ADC designs are dynamic, meaning that there is no quiescent current. This class of comparators relies on positive feedback to increase the speed and avoid meta-stability. The speed and power of the comparator also directly impact on the performance of the entire ADC.

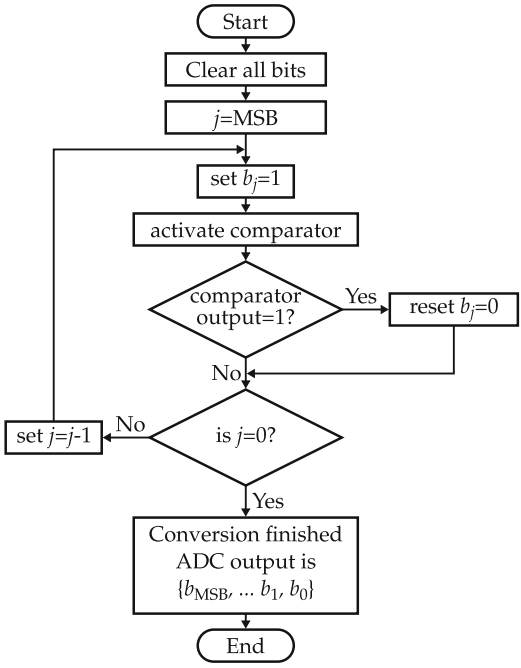



**Fig. 2.8** Fluxogram of the successive approximation algorithm

Additionally, the comparator noise appears summed up to the ADC quantization noise. Hence, the comparator noise is critical to the overall noise performance of the ADC and must be carefully dimensioned.

While all the components of the SAR ADC influence its performance to some degree, the most critical element of the architecture is the DAC. As the DAC is responsible to generate all the reference transition levels, it dictates the linearity of the ADC transfer curve. Moreover, all the CR-based ADCs have the TH functionality merged into the DAC, meaning that the noise on the DAC manifests itself on the ADC output and must be prudently dimensioned. Capacitive SAR ADCs use an array of capacitors and switches as the DAC, and what differentiates the implementations is the way that these capacitors are successively connected during the binary search, generally referred to as the switching scheme. A detailed review of the main switching schemes employed in modern SAR ADCs is presented in the next chapter.

## References


1. W.-H. Chen, W.-F. Loke, B. Jung, A 0.5-V, 440$\mu$W frequency synthesizer for implantable medical devices. IEEE J. Solid State Circuits **47**(8), 1896–1907 (2012). doi:10.1109/JSSC.2012.2196315 (cit. on p. 11)
2. K. Wise, A. Sodagar, Y. Yao, M. Gulari, G. Perlin, K. Najafi, Microelectrodes, microelectronics, and implantable neural microsystems. Proc. IEEE **96**(7), 1184–1202 (2008). doi:10.1109/JPROC.2008.922564 (cit. on p. 11)
3. A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, Wireless sensor networks for habitat monitoring, in *ACM International Workshop on Wireless Sensor Networks and Applications* (ACM, New york, 2002), pp. 88–97. doi:10.1145/570738.570751 (cit. on p. 11)
4. E.S. Biagioni, K. Bridges, The application of remote sensor technology to assist the recovery of rare and endangered species. Int. J. High Perform. Comput. Appl. **16**(3), 315–324 (2002). doi:10.1177/109434200201600310 01
5. L. Schwiebert, S.K. Gupta, J. Weinmann, Research challenges in wireless networks of biomedical sensors, in *International Conference on Mobile Computing and Networking* (ACM, New York, 2001), pp. 151–165. doi:10.1145/381677.381692
6. S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, M. Turon, Health monitoring of civil infrastructures using wireless sensor networks, in *International Symposium on Information Processing in Sensor Networks (IPSN)* (IEEE, New York, 2007), pp. 254–263. doi:10.1109/IPSN.2007.4379685
7. J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008). doi:10.1016/j.comnet.2008.04.002 (cit. on p. 11)
8. P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, in *International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, New York, 2002), pp. 96–107. doi:10.1145/605397.605408 (cit. on p. 12)
9. C.Ó. Mathúna, T. O'Donnell, R.V. Martinez-Catala, J. Rohan, B. O'Flynn, Energy scavenging for long-term deployable wireless sensor networks. Talanta **75**(3), 613–623 (2008). Special Section: Remote Sensing. doi:10.1016/j.talanta.2007.12.021 (cit. on pp. 12 and 13)
10. Y. Tan, *Energy Harvesting Autonomous Sensor Systems: Design, Analysis, and Practical Implementation* (Taylor & Francis, Boca Raton, 2013). isbn:9781439892732 (cit. on pp. 12 and 13)

11. Micropower step up low-voltage booster module enables practical energy capture from low power generators. Advanced Linear Devices, Inc. (Online). Available: http://www.aldinc.com/pdf/MicroPowerStepUpLowVoltageBoosterModule.pdf (visited on 10/2014) (cit. on p. 13)
12. A. Wang, B. Calhoun, A. Chandrakasan, *Sub-Threshold Design for Ultra Low-Power Systems*. Integrated Circuits and Systems. (Springer, New York, 2006). isbn:9780387345017. doi:10.1007/978-0-387-34501-7 (cit. on p. 13)
13. J. Rabaey, *Low Power Design Essentials*. Integrated Circuits and Systems (Springer, New York, 2009). isbn: 9780387717135.doi:10.1007/978-0-387-71713-5 (cit. on p. 13)
14. International technology roadmap for semiconductors, 2013. (Online). Available: http://www.itrs.net/Links/2013ITRS/2013Chapters/2013PIDS.pdf (visited on 10/2014). (cit. on p. 14)
15. M. Konijnenburg, Y. Cho, M. Ashouei, T. Gemmeke, C. Kim, J. Hulzink, J. Stuyt, M. Jung, J. Huisken, S. Ryu, J. Kim, H. de Groot, Reliable and energy-efficient 1MHz 0.4V dynamically reconfigurable SoC for ExG applications in 40 nm LP CMOS in *IEEE International Solid-State Circuits Conference*, Feb 2013, pp. 430–431. doi:10.1109/ISSCC.2013.6487801 (cit. on p. 14)
16. J. Kulkarni, K. Kim, K. Roy, A 160 mV robust Schmitt trigger based subthreshold SRAM. IEEE J. Solid State Circuits **42**(10), 2303–2313 (2007). doi:10.1109/JSSC.2007.897148 (cit. on p. 14)
17. L. Ferreira, S. Sonkusale, A 60-dB gain OTA operating at 0.25-V power supply in 130-nm digital CMOS process. IEEE Trans. Circuits Syst. I Regul. Pap. **61**(6), 1609–1617 (2014). doi:10.1109/TCSI.2013.2289413 (cit. on p. 14)
18. S. Chatterjee, K. Pun, N. Stanic, Y. Tsividis, P. Kinget, *Analog Circuit Design Techniques at 0.5V*. Analog Circuits and Signal Processing. (Springer, New York, 2010). isbn:9780387699547. doi:10.1007/978-0-387-69954-7 (cit. on p. 14)
19. F. Zhang, K. Wang, J. Koo, Y. Miyahara, B. Otis, A 1.6 mW 300 mV- supply 2.4 GHz receiver with −94 dBm sensitivity for energy-harvesting applications, in *IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 56 (IEEE, New York, 2013), pp. 456–457. doi:10.1109/ISSCC.2013.6487813 (cit. on p. 14)
20. J. Cheng, N. Qi, P.Y. Chiang, A. Natarajan, A low-power, low-voltage WBAN-compatible sub-sampling PSK receiver in 65 nm CMOS. IEEE J. Solid-State Circuits **49**(12), 3018–3030 (2014). doi:10.1109/JSSC.2014.2362840 (cit. on p. 14)
21. U. Wismar, D. Wisland, P. Andreani, A 0.2 V 0.44 $\mu$W 20 kHz analog to digital $\Sigma\Delta$ modulator with 57 fJ/conversion FoM, in *IEEE European Solid- State Circuits Conference* (2006), pp. 187–190. doi:10.1109/ESSCIR.2006.307562 (cit. on p. 14)
22. H.-Y. Tai, H.-W. Chen, H.-S. Chen, A 3.2fJ/c.-s. 0.35V 10b 100kS/s SAR ADC in 90 nm CMOS, in *IEEE Symposium on VLSI Circuits* (2012), pp. 92–93. isbn:978-1-4673-0849-6. doi:10.1109/VLSIC.2012.6243805
23. A. Shikata, R. Sekimoto, T. Kuroda, H. Ishikuro, A 0.5 V 1.1 MS/sec 6.3 fJ/conversion-step SAR-ADC with tri-level comparator in 40 nm CMOS. IEEE J. Solid State Circuits **47**(4), 1022–1030 (2012). issn:0018-9200. doi:10.1109/JSSC.2012.2185352
24. D. Senderowicz, G. Nicollini, S. Pernici, A. Nagari, P. Confalonieri, C. Dallavalle, Low-voltage double-sampled $\Sigma\Delta$ converters. IEEE J. Solid State Circuits **32**(12), 1907–1919 (1997). doi:10.1109/4.643649
25. M. Yip, A.P. Chandrakasan, A resolution-reconfigurable 5-to-10b 0.4- to-1 V power scalable SAR ADC, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2011), pp. 190–192, isbn:978-1-61284-303-2. doi:10.1109/ISSCC.2011.5746277 (cit. on p. 14)
26. W.M. Goodall, Telephony by pulse code modulation. Bell Syst. Tech. J. **26**(3), 395–409 (1947). doi:10.1002/j.1538-7305.1947.tb00902.x (cit. on p. 21)
27. B. Gordon, R. Talambiras, Signal conversion apparatus, US patent 3,108,266, 1963 (cit. on p. 21)
28. W. Kester (ed.), *Data Conversion Handbook* (Newnes, Oxford, UK, 2005). isbn:978-0-7506-7841-4. (cit. on p. 21)
29. J.-H. Tsai, Y.-J. Chen, M.-H. Shen, P.-C. Huang, 1-V, 8b, 40MS/s, 113$\mu$W charge-recycling SAR ADC with a 14$\mu$W asynchronous controller, in *IEEE Symposium on VLSI Circuits* (2011), pp. 264–265 (cit. on p. 22)

# Chapter 3
# Review of SAR ADC Switching Schemes

**Contents**



## 3.1 Introduction

A switching scheme, in the context of capacitive SAR ADCs, comprehends the arrangement of switches and capacitors in the DAC, the sequence in which these switches are turned on and off, and the nodes to which these switches connect the capacitors. The switching scheme dictates the energy-efficiency, DAC size, and other critical performance metrics of the ADC.



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_3

## 3.2 Operation Modes of Charge-Based SAR ADCs

The switching schemes employed in modern SAR ADC designs rely on one of the following operation modes: charge redistribution and charge sharing. Before detailing the switching schemes and describing the operation of complete ADCs, we use simplified circuits to introduce and draw the main differences between CR and CS. We limit these circuits to only a pair of capacitors to ease the understanding, but the principles are easily extensible to arrays of any size. For CR-based topologies, the plate of a capacitor which is connected to a comparator input is hereafter referred to as top-plate, independently of its physical construction. The voltage on this node is labeled $V_{\mathrm{TOP}}$ and represents the DAC output. Both architectures require the absence of current paths from/to the top plate of the capacitors during the conversion and, as a consequence, the comparator must present a very high input impedance. Luckily, this is effortlessly achieved when the comparator is designed using the gates of MOS transistors as input nodes.

### *3.2.1 The Charge Redistribution Principle*

In the CR scheme, the DAC output is set by varying the voltage on the bottom plate of the capacitors, while maintaining the total capacitance unchanged. In the example shown in Fig. 3.1, this is achieved by switching $S_1$ between $V_1$ and ground. Since there is no current path from/to $V_{\mathrm{TOP}}$, the charge on the top plate ($Q_{\mathrm{TOP}}$) is constant independently of the changes in $S_1$. The voltage $V_{\mathrm{TOP}}$, on the other hand, varies according to $S_1$. The amount of variation is also dictated by the ratio of $C_1$ to the total capacitance $C_1 + C_2$, as depicted in Fig. 3.1 and mathematically derived as follows.

We define $t_1$ as the instant when the switch $S_1$ is disconnected from ground and connected to $V_1$. Similarly, $t_0$ represents an instant before $t_1$, when $V_1$ is settled and $S_1$ is still connected to ground. Also, $t_2$ is defined as a point in time after the switching when the circuit has already settled with sufficient accuracy. The total charge in the top plate at $t_0$ is given by

$$Q_{\mathrm{TOP}}(t_0) = V_{\mathrm{TOP}}(t_0)\,(C_2 + C_1)\,. \tag{3.1}$$

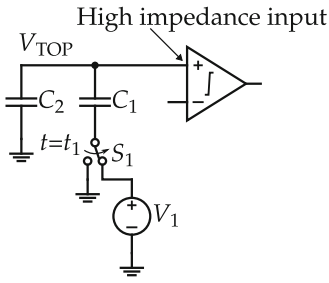



**Fig. 3.1** Simplified circuit demonstrating the CR principle

Similarly, $Q_{TOP}$ can be computed at time $t_2$ using (3.2).

$$Q_{TOP}(t_2) = V_{TOP}(t_2)C_2 + (V_{TOP}(t_2) - V_1)\,C_1. \tag{3.2}$$

Since there is conservation of charge in the top plate,

$$Q_{TOP}(t_2) = Q_{TOP}(t_0). \tag{3.3}$$

Equaling (3.1) to (3.2) enables us to find $V_{TOP}(t_2)$.

$$V_{TOP}(t_2) = V_{TOP}(t_0) + V_1 \frac{C_1}{C_1 + C_2}. \tag{3.4}$$

In practical implementations of CR-based ADCs, the TH functionality is merged into the capacitive array. This is achieved by employing additional switches to assign the input voltage to $V_{TOP}$ at the beginning of the conversion, i.e. $V_{TOP}(t_0) = V_{in}$. The ability to shift the voltage up and down in controlled steps (DAC functionality) is attained by using multiple capacitors with switchable bottom plates, such as $C_1$ in the example. Most commonly, these capacitors have binary weights of capacitance.

### 3.2.2 *The Charge Sharing Principle*

In CS-based ADCs, the DAC output is changed by connecting precharged capacitors to the DAC nodes. Thus, the total capacitance of the DAC increases during the conversion. The initial charge in these capacitors and their capacitances dictates the amount of variation in the DAC output voltage. Since there is no current path from/to $V_{TOP}$, the total charge on the top plate ($Q_{TOP}$) is the sum of the initial charges of all the connected capacitors. The voltage $V_{TOP}$ is given by the total charge divided by the total capacitance. This is depicted in Fig. 3.2 and mathematically derived as follows.

Again, as in the previous example, $t_1$ is the instant when the switch changes its state, $t_0$ is an instant before $t_1$, and $t_2$ represents a moment in time after complete settling. The charges $Q_1$ and $Q_2$ at $t = t_0$ are given by

$$Q_1(t_0) = C_1 V_1. \tag{3.5}$$

$$Q_2(t_0) = C_2 V_{TOP}(t_0). \tag{3.6}$$

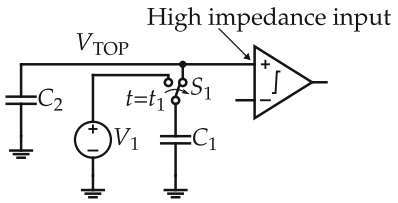



**Fig. 3.2** Simplified circuit demonstrating the CS principle

At time $t_1$, $S_1$ disconnects $C_1$ from $V_1$ and connects to $C_2$ and to the comparator. The total charge on the top plates at $t_2$ is given by

$$Q_{\mathrm{TOP}}(t_2) = V_{\mathrm{TOP}}(t_2)\,(C_2 + C_1)\,. \tag{3.7}$$

Again, due to charge conservation

$$Q_{\mathrm{TOP}}(t_2) = Q_2(t_0) + Q_1(t_0). \tag{3.8}$$

The expressions (3.5), (3.6), (3.7), and (3.8) may be used to find $V_{\mathrm{TOP}}(t_2)$.

$$V_{\mathrm{TOP}}(t_2) = \frac{C_2 V_{\mathrm{TOP}}(t_0) + C_1 V_1}{C_2 + C_1}. \tag{3.9}$$

Differently from the CR-ADC, in the CS-ADC the TH functionality is not performed by the DAC capacitors. In a practical implementation, the input signal would be assigned to $V_{\mathrm{TOP}}$ at the beginning of the conversion, i.e. $V_{\mathrm{TOP}}(t_0) = V_{\mathrm{in}}$. The ability to shift the voltage up and down in controlled steps (DAC functionality) is achieved by connecting binary-weighted capacitors. As these capacitors must remain connected until the end of the conversion, the final voltage suffers some attenuation. Ideally, the comparator detects only the sign of the voltage, and this attenuation has no effect on the linearity of the ADC. The impact of the comparator offset is investigated in Chap. 4.

In the remaining of this chapter, the most relevant switching schemes used in SAR ADCs are reviewed, while the main differences are drawn and a comparison is made concerning their energy efficiency.

## 3.3 Charge Redistribution Switching Schemes

Since the disclosure of the CR-based SAR ADC in [1], many switching schemes appeared in literature, aiming to improve mostly the energy efficiency and circuit area. For this section, some of the most relevant switching schemes are reviewed and summarized: conventional, monotonic, $V_{\mathrm{CM}}$-based, and tri-level. The procedure for calculating the DAC voltage and energy in CR-based architectures is described in Appendix "Voltage and Energy in CR ADCs".

### *3.3.1 Conventional Switching*

The conventional SAR algorithm [1] is depicted in Fig. 3.3 for a 3-bit fully differential implementation. The figure includes the amount of energy that is spent

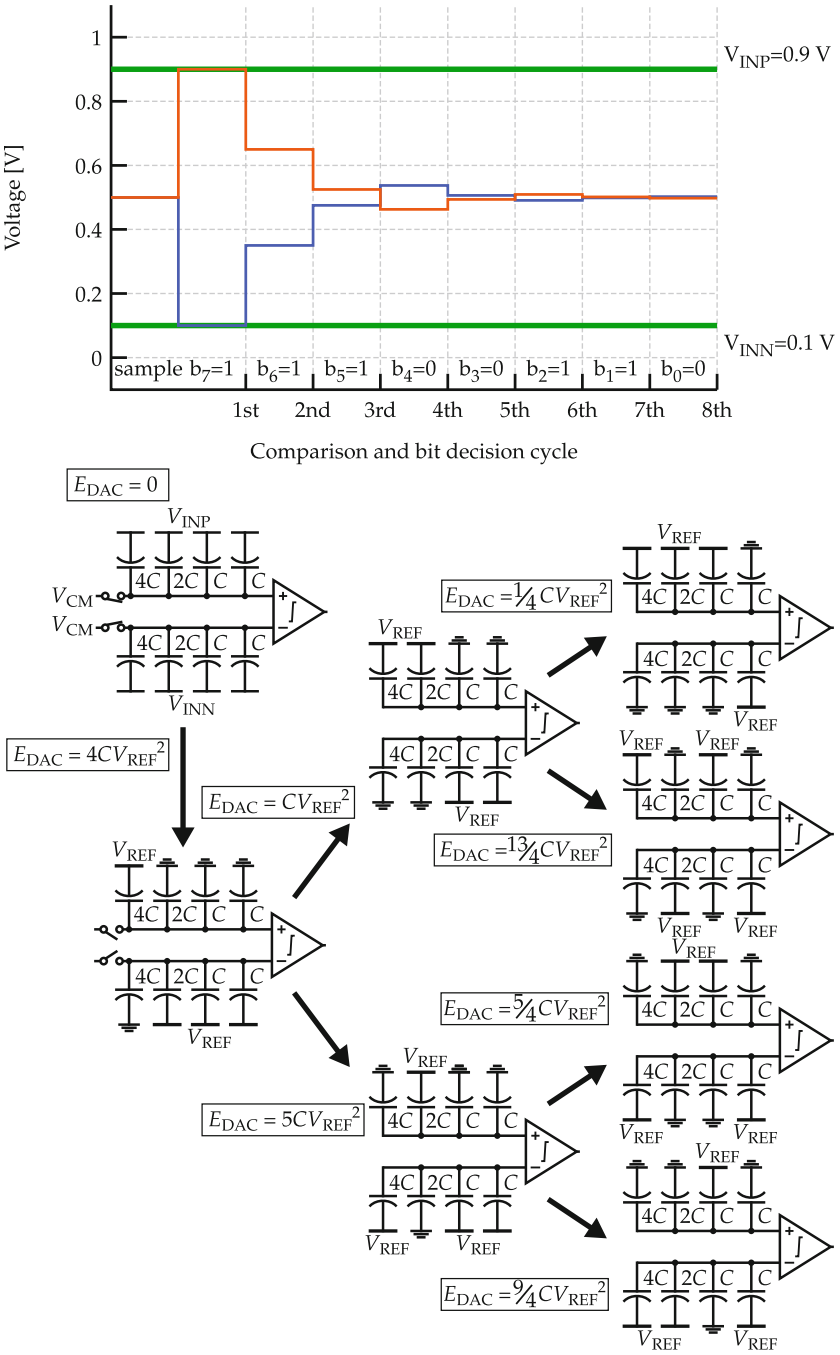



**Fig. 3.3** Conventional CR switching scheme waveform and conversion procedure

to move the charges at each cycle. The voltage at the comparator inputs during a complete 8-bit conversion is also shown in Fig. 3.3, for a 0.8 V differential input with 0.5 V of common-mode and 1 V of reference voltage.

In the first phase of the algorithm, the input is sampled on the bottom plate of the capacitors (those not connected to the comparator). Once the sampling is finished, these capacitors are disconnected from the input. Then, in the top array, the MSB-capacitor is connected to $V_{\mathrm{REF}}$ while the remaining capacitors are connected to ground. On the bottom-array, the opposite is done. For the 3-bit example, this operation results in an energy consumption of $4CV_{\mathrm{REF}}^2$. The comparator is activated, and the result dictates the MSB of the output word. Depending on whether the MSB is "0" or "1," the DAC takes the "up" or "down" transition, as depicted in Fig. 3.3, and consumes $CV_{\mathrm{REF}}^2$ or $5CV_{\mathrm{REF}}^2$, respectively. Following this reasoning, for the 3-bit example, there are four possible trajectories that the ADC may take depending on the input signal.

It should be noted that the charges are not necessarily moved efficiently, and a significant amount of energy is wasted during the operation. There is also a large unbalance between the total energy consumed in the "up" and "down" paths. In the second cycle, the "down" transition uses five times more energy than "up" transition. Because none of the charge is recycled, i.e. some capacitors are discharged to ground while "new" energy is drawn from the supply, the scheme is not very efficient. The average energy of a $B$-bit switching algorithm, assuming all codes are equiprobable, may be derived as

$$\begin{aligned} E_{\mathrm{conventional}} &= \sum_{i=1}^{B} \left(2^{B+1-2i}\left(2^i-1\right)\right) C{V_{\mathrm{REF}}}^2 \\ &= \left(\frac{2^{-B+1}+2^{B+2}}{3}-2\right) C{V_{\mathrm{REF}}}^2. \end{aligned} \tag{3.10}$$

### 3.3.2 *Monotonic or "Set-and-Down" Switching*

Aiming to improve the efficiency of the conventional switching scheme, Liu et al. proposed in [2] a monotonic algorithm, which is depicted in Fig. 3.4. The idea behind the technique is to use only discharging cycles, with no explicit charging operation. Additionally, since the scheme employs top-plate sampling, it requires only $2^{B-1}$ capacitors instead of $2^B$. The simplification is possible because top-plate sampling enables the MSB to be obtained by directly comparing the input signal sampled on the top plates, without switching any capacitor.

In the first phase of the algorithm, the input is sampled on the top plates of the capacitive arrays, while the bottom plates are connected to the reference voltage. The MSB is directly obtained by the comparator. Depending on whether the MSB is "0" or "1," the MSB-capacitor of the bottom array or from the top array is connected

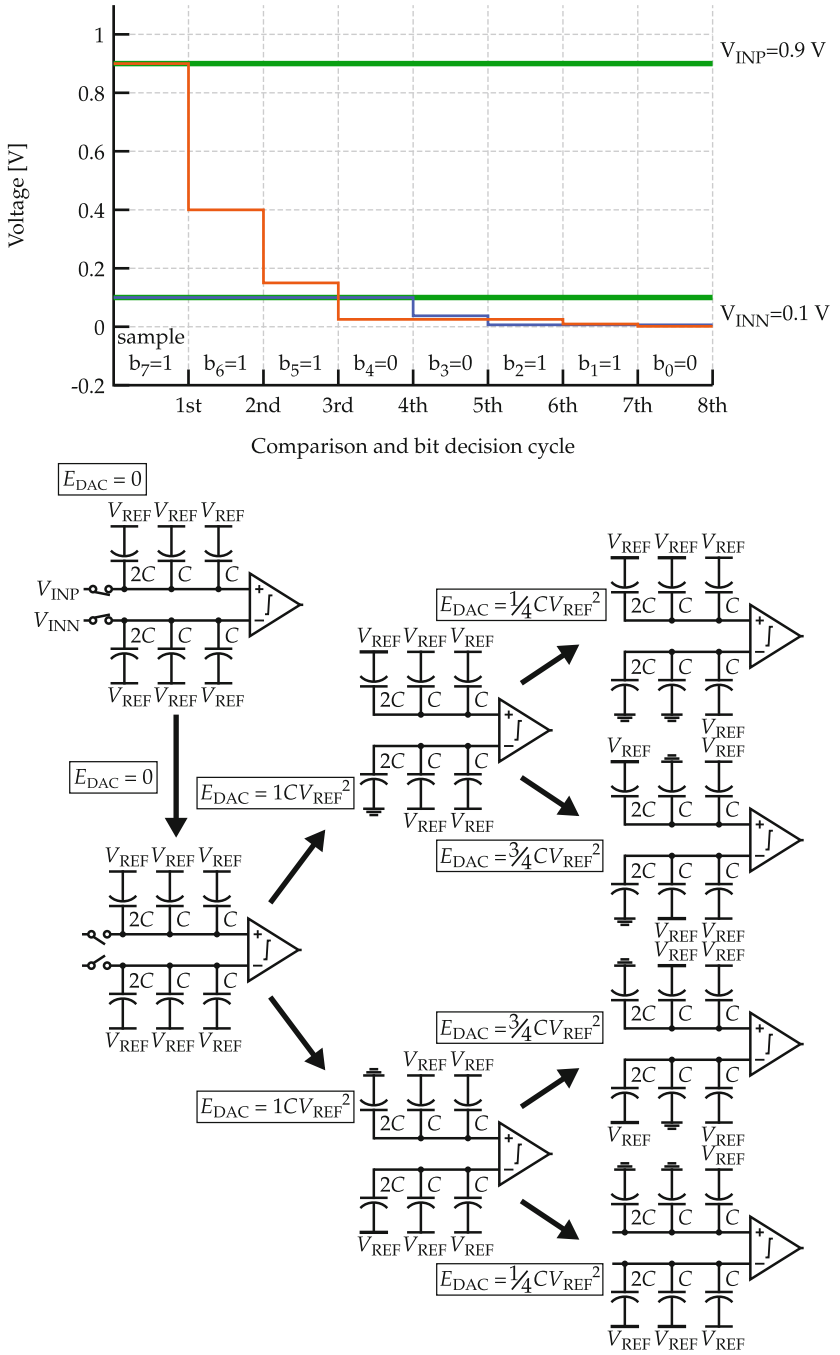



**Fig. 3.4** Monotonic CR switching scheme waveform and conversion procedure

to ground, respectively. An identical procedure is carried out for the following bits, reducing the differential voltage towards zero while the output bits are extracted from the comparator output. As it can be seen in Fig. 3.4, the common-mode voltage of the DAC starts with the same value of the input and decreases towards ground during the conversion. As the comparator performance is sensitive to the common-mode voltage of the capacitive arrays, its performance is expected to vary throughout the conversion. Most critically, the comparator offset is modulated by the common-mode voltage, which may degrade the achievable linearity of the ADC.

Contrasting to the conventional switching scheme, the MSB cycle consumes no energy. The average switching energy of a $B$-bit monotonic switching algorithm, assuming equiprobable output codes, can be derived as

$$\begin{aligned} E_{\text{monotonic}} &= \sum_{i=1}^{B-1} \left(2^{B-2-i}\right) C{V_{\text{REF}}}^2 \\ &= \left(2^{B-2} - \frac{1}{2}\right) C{V_{\text{REF}}}^2. \end{aligned} \tag{3.11}$$

The monotonic switching scheme consumes only 18.74 % of the energy spent on the conventional implementation when comparing 10-bit ADCs.

This switching scheme inspired other works based on a similar operation. The scheme proposed in [3] also operates monotonically but decreases the variation in the common-mode voltage. The signal is sampled in the top-plates, but the bottom plates are initially at $V_{\text{CM}}$. After the MSB comparison, all the bottom plates of the array with lower voltage (top or bottom) are switched to $V_{\text{REF}}$. Afterwards, all the transitions are "down" and with $V_{\text{REF}}/2$ magnitude ($V_{\text{REF}} \rightarrow V_{\text{CM}}$ or $V_{\text{CM}} \rightarrow$ ground). Therefore, the common-mode voltage has a jump of 50 % after the first comparison and converges towards its initial value during the conversion. Also, the scheme performs better regarding energy efficiency at the cost of a more complicated digital controller and additional accuracy requirements on the $V_{\text{CM}}$ source.

### 3.3.3 $V_{\text{CM}}$-Based Capacitor Switching

The $V_{\text{CM}}$-based SAR ADC, alternatively called merged capacitor switching (MCS) [4, 5], is shown in Fig. 3.5. This switching scheme also employs top-plate sampling, so that in the first cycle the inputs are connected to the top plates of the capacitor arrays and the bottom plates are connected to $V_{\text{CM}}$. The MSB is determined from a comparison immediately after the sampling without consuming any energy in the array. Depending on whether the comparator result is a "0" or a "1," the largest capacitor of the top array is discharged to ground or charged to $V_{\text{REF}}$, respectively. The opposite is done for the bottom array. This procedure is adopted for all the remaining bits in the converter.

Similarly to the monotonic switching, the $V_{\text{CM}}$-based switching scheme requires only half of the capacitance for the same resolution, when compared to the

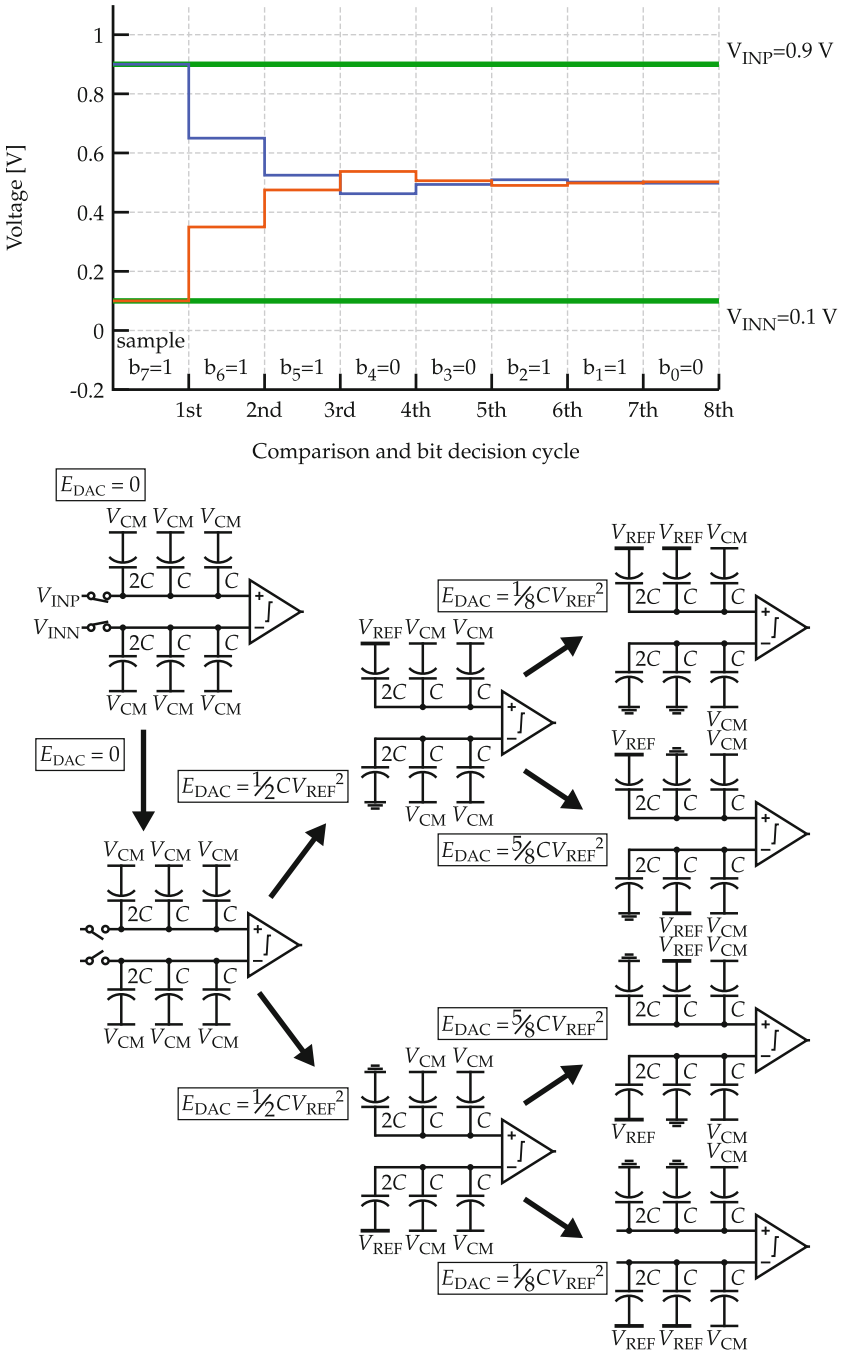



**Fig. 3.5** $V_{CM}$-based (MCS) CR switching scheme waveform and conversion procedure

conventional switching scheme. Further improvement in the energy efficiency is achieved by moving the voltages by only $V_{REF} - V_{CM}$ in the top and bottom arrays, instead of a full $V_{REF}$-step, as happens in the monotonic and the conventional procedure. Since the energy consumption becomes proportional to $C(V_{REF}/2)^2$ in the $V_{CM}$-based scheme, a fourfold energy reduction is achieved for each cycle. Also, the $V_{CM}$-based switching maintains the common-mode voltage of the input during the whole conversion, as can be seen in Fig. 3.5, which does not hinder the design of the comparator as in the monotonic scheme. The average energy of a $B$-bit $V_{CM}$-based switching algorithm is derived as

$$\begin{aligned} E_{MCS} &= \sum_{i=1}^{B-1} \left(2^{B-2-2i}\left(2^i - 1\right)\right) C{V_{REF}}^2 \\ &= \left(\frac{2^{-B} + 2^{B-1}}{3} - \frac{1}{2}\right) C{V_{REF}}^2. \end{aligned} \tag{3.12}$$

It can be demonstrated that the algorithm consumes only 12.48 % of the energy consumed by the conventional SAR in 10-bit ADCs. The given expression is valid only if $V_{CM}$ is precisely half of $V_{REF}$. In that case, the switching array does not consume energy from $V_{CM}$ during the conversion cycles. If this condition is not satisfied, however, the energy consumed from $V_{CM}$ should be accounted for separately. It is worth of note that the linearity of the ADC is not sensitive to the accuracy of $V_{CM}$, as long as it remains constant during the conversion.

An interesting evolution of the $V_{CM}$-based switching scheme is proposed in [6**?** ]. Those works reduce the minimum size of the capacitive array to half when compared to the $V_{CM}$-based scheme, also leading to half the energy consumption. This is achieved by switching only one capacitor among the top and the bottom arrays during the LSB cycle. The drawbacks are an additional requirement that $V_{CM}$ must be accurate and fixed to $V_{REF}/2$, and a little attenuation on the common-mode voltage for the LSB decision.

### *3.3.4 Tri-Level Capacitor Switching*

Yuan et al. proposed the tri-level switching scheme in [7], which is shown in Fig. 3.6. The ADC has an improved efficiency by operating in only one of the arrays of a differential implementation. Initially, the bottom plates of all capacitors are grounded, and the input signal is sampled on their top plates. After the MSB is obtained by the comparator, the SAR logic switches the bottom plates of the whole array which sampled the lower voltage to $V_{CM}$. This operation causes the voltage on the array to grow by $V_{REF}/2$, which enables MSB−1 to be determined. The other capacitor array remains unchanged during the rest of the conversion and does not

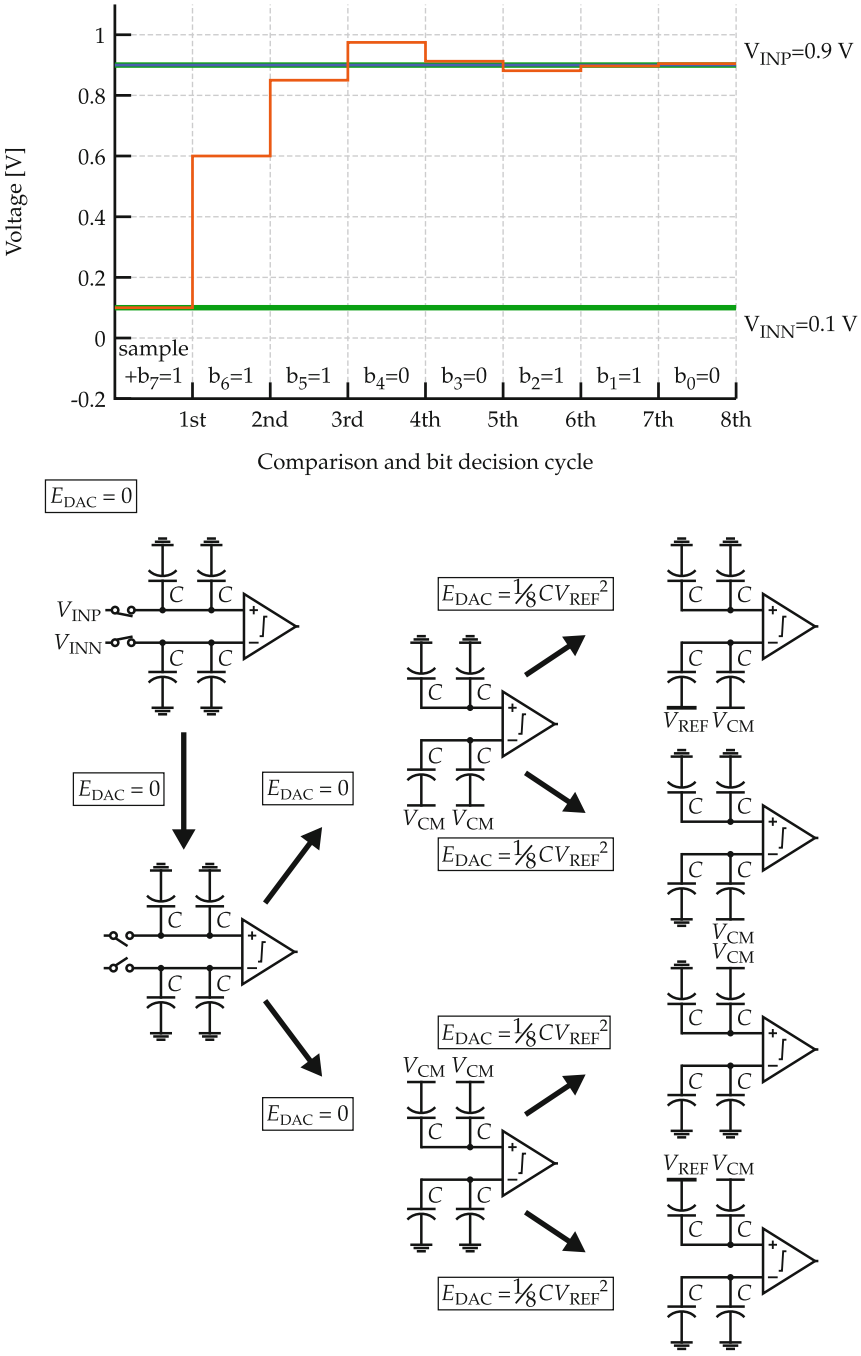



**Fig. 3.6** Tri-level CR switching scheme waveform and conversion procedure

consume energy. According to MSB−1, the bottom plate of the largest capacitor on the active array is connected to either $V_{\mathrm{REF}}$ or ground. The same procedure is used to determine the remaining bits.

The number of capacitors is reduced by half of those used for the $V_{\mathrm{CM}}$-based ADC. The combination of top-plate sampling, single-side operation and the usage of $V_{\mathrm{CM}}$ as an accurate reference in the conversion makes this architecture very efficient in terms of area and energy. The energy consumption is reduced to 3.11 % of that consumed by the conventional scheme, while the area is decreased to one fourth. It is to be noted that the implementation of the SAR controller becomes slightly more complex when using this scheme, which may cause an increase in the power consumption. Also, since the architecture is very sensitive to $V_{\mathrm{CM}}$, in addition to $V_{\mathrm{REF}}$, the former must be treated as an additional reference and its generation may become nontrivial. Finally, the power consumption on the $V_{\mathrm{CM}}$ becomes significant because, differently from the $V_{\mathrm{CM}}$-based scheme, there is energy consumption even when $V_{\mathrm{CM}} = V_{\mathrm{REF}}/2$. The average energy of the DAC of a $B$-bit SAR ADC using the tri-level switching scheme (already accounting the energy of $V_{\mathrm{CM}} = V_{\mathrm{REF}}/2$) is derived as

$$\begin{aligned} E_{\text{tri-level}} &= \sum_{i=1}^{B-1} \left(2^{B-3-2i}\left(2^{i}-2\right)\right) C{V_{\mathrm{REF}}}^{2} \\ &= \left(\frac{2^{-B}+2^{B-3}}{3} - \frac{1}{4}\right) C{V_{\mathrm{REF}}}^{2}. \end{aligned} \tag{3.13}$$

## 3.4 Charge Sharing Switching Scheme

The operation of a fully differential CS-ADC is depicted in Fig. 3.7. The voltage at the comparator inputs during a complete 8-bit conversion is also shown in Fig. 3.7 for a 0.8 V differential input with 0.5 V common-mode and 1 V reference voltage. In the first cycle, the inputs are sampled in the TH, that is implemented explicitly in the CS scheme. Simultaneously, the capacitances in the array are precharged to $V_{\mathrm{PC}}$. In the following cycle, the comparator is activated, evaluating the voltage on the TH to decide the MSB. According to the result, the largest capacitor on the array is connected in parallel or anti-parallel to the TH, adding or subtracting charge, respectively. This procedure is repeated for the other bits, as the differential voltage at the comparator inputs decreases towards zero and the comparison results are stored to form the digital output. The common-mode voltage at the comparator inputs is maintained constant in this scheme and equal to the input common-mode voltage.

The CS decides the MSB right after sampling, without any charge processing, as happens when employing top-plate sampling in CR schemes. This characteristic exempts the implementation of the MSB capacitor, leading to a smaller array capacitance. Moreover, a fully differential implementation of the CS scheme

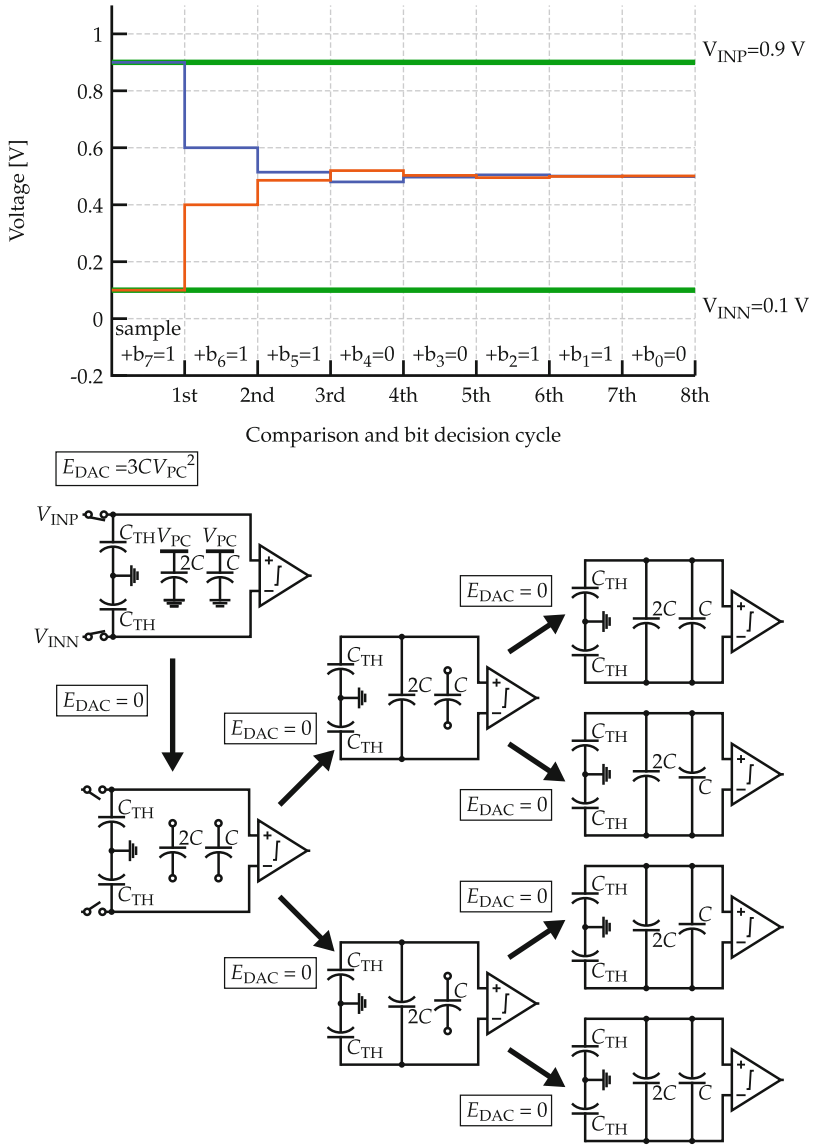



**Fig. 3.7** CS switching scheme waveform and conversion procedure

requires only one capacitor array, unlike the CR schemes that require two. For these reasons, the DAC of a CS-ADC has the smallest area among the reviewed architectures. On the other hand, the CS requires an explicit TH circuit, because sampling is not performed by the DAC capacitors, and the capacitance of the TH has to be accounted for in the total area. In some cases, it is made larger than the

total capacitance of the DAC, which increases the input buffer power consumption and brings a significant impact on the chip area. However, the requirements for the TH capacitors are much less stringent than for the array capacitors concerning matching and does not need to rely on special layout techniques such as common-centroid or reuse of unit-cells. Moreover, the TH is much less sensitive to parasitics, which allows the usage of capacitors with higher densities, reducing the impact on the overall ADC area. Further discussion on the size of the TH is provided at the end of this section and in Chap. 4.

An important aspect of the CS switching scheme is that the total capacitance connected to the comparator is time-varying, i.e. it corresponds to the TH capacitance at the beginning of the conversion and increases as capacitors from the array are connected successively. This characteristic brings two significant drawbacks to the topology. The first drawback is that the voltage seen by the comparator is steadily attenuated. As the voltage is given by the ratio between charge and capacitance ($V = Q/C$), and $C$ increases during the conversion, $V$ decreases accordingly. This voltage attenuation affects the noise tolerance of the topology negatively. In other words, the CS-ADC performs worse in terms of noise than the CR counterpart when both are subject to the same noise sources (identical comparators and thermal noise on the capacitors). The second drawback, brought by the time-varying behavior of the total capacitance, is that the ADC becomes nonlinear in the presence of comparator offset. Both effects are thoroughly investigated and quantified in Chap. 4.

As can be seen in Fig. 3.7, all the energy consumption of the reference source takes place in the precharge cycle. In all the succeeding cycles the DAC behaves passively. This characteristic of the CS switching scheme is contrasted to the CR-based topologies in Fig. 3.8, that shows the shape of the precharge current in the CS-ADC and the reference current in the CR-ADC.

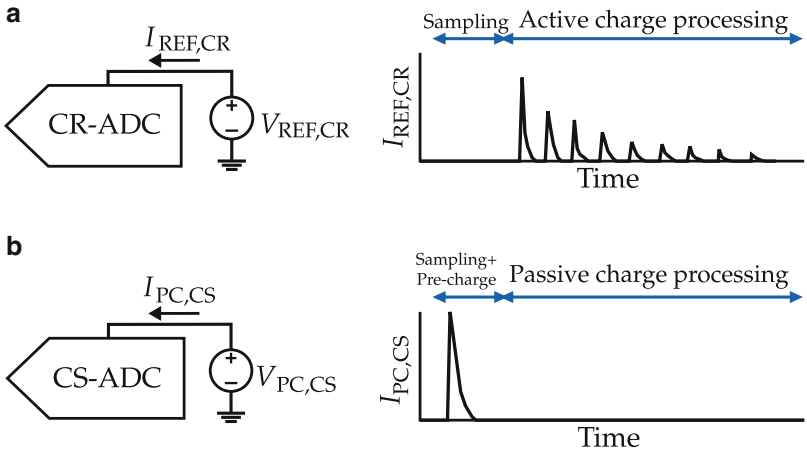



**Fig. 3.8** Waveforms of (**a**) the reference current in a CR-ADC and (**b**) the precharge current in a CS-ADC

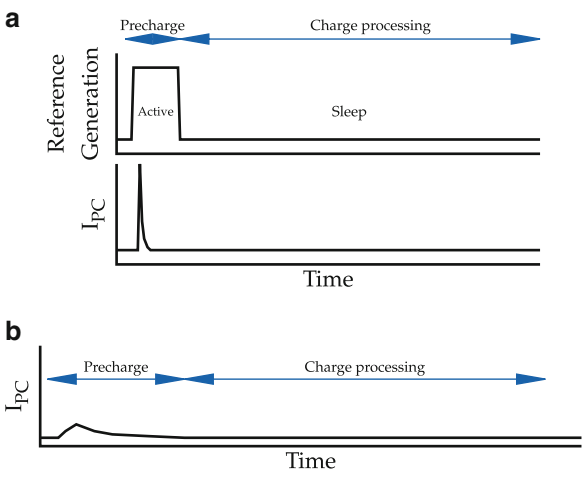



**Fig. 3.9** Approaches for power-saving in the generation of reference voltage for CS-ADCs. (**a**) Duty-cycling of reference generation. (**b**) Relaxing the reference buffer

In the CS-ADC, all the required charge is drained at once, while in the CR, some charge is drained in every steps. Therefore, there are implications at system-level, according to the architecture employed: in the case of the CR-based SAR ADCs, the reference buffer must be active during the whole conversion; for the CS-ADC, on the other hand, the reference buffer must be active only during the precharge. This attribute may be exploited for further energy savings, by employing duty-cycled reference generation or relaxing the reference buffer. Figure 3.9 depicts the two approaches. In Fig. 3.9a, the reference generation circuit (that includes the bandgap voltage reference and voltage buffer) is made active during the precharge cycle and is switched to sleep mode during the charge processing in the ADC. In Fig. 3.9b, the voltage buffer employed for the reference buffer is designed with a higher output impedance and lower power consumption. The outcome is an increase in the time required to precharge the DAC capacitances. The time spent in charge processing, on the other hand, is maintained.

The charge-based operation of the CS-ADC allows for further optimizations regarding power consumption and area. As the topology depends on binary-weighted values of charge to operate, instead of binary values of capacitance, it is possible to reduce the total capacitance of the array by utilizing multiple precharge voltages. Four different approaches to realize the precharge cycle in an 8-bit ADC are shown in Fig. 3.10. In the figure, only the capacitor array and the switches that are responsible for the precharge are shown, while the switches that connect the capacitors to the TH are omitted.

In the 1-step precharge scheme (Fig. 3.10a), which is the most straightforward, all the capacitors are charged to the same potential ($V_{DD}$, for example), while their capacitance values follow binary weights. As a result, the total energy consumed to precharge the array is

$$E_{CS} = (2^{B-1} - 1)CV_{PC}{}^2. \tag{3.14}$$

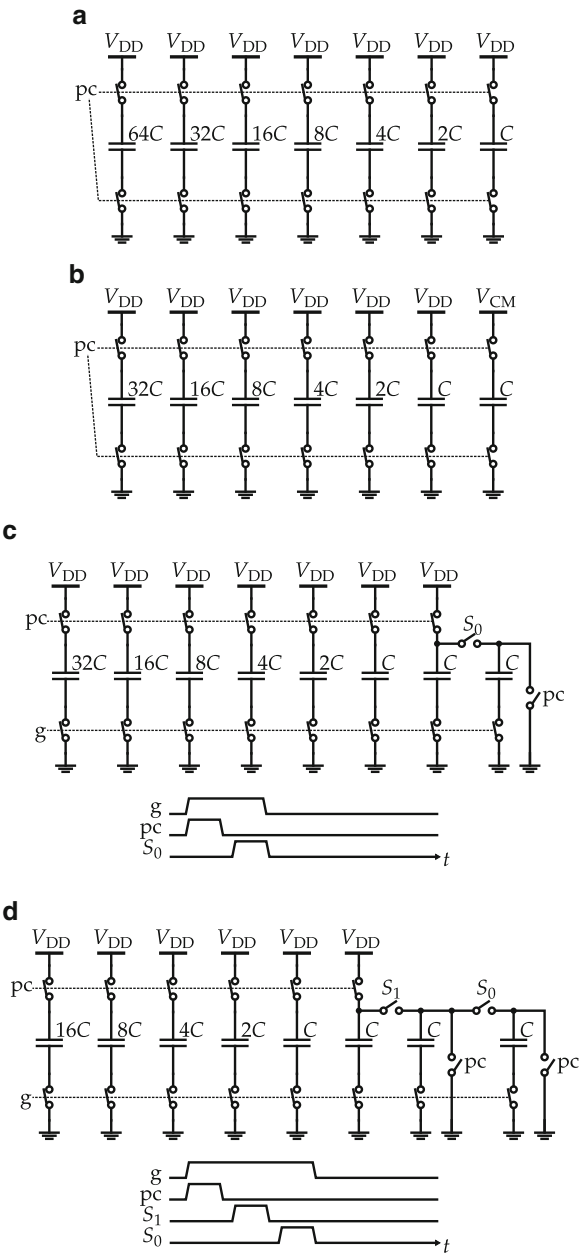



**Fig. 3.10** Approaches for energy saving in the precharge cycle of CS-ADCs: 1-step with $V_{PC} = V_{DD}$ (**a**), 1-step with $V_{PC} = V_{CM}$ only for the LSB (**b**), 2-step with $V_{PC} = V_{DD}$ (**c**), and 3-step with $V_{PC} = V_{DD}$ (**d**)

In similar fashion to what happens for some CR switching schemes, the CS architecture may benefit from using $V_{CM}$ as a complementary reference level. Likewise, this depends on the accuracy of $V_{CM}$ to resolve the LSB and assumes that it is half of $V_{DD}$. As shown in Fig. 3.10b, it is possible to reduce the total capacitance

to half of that employed in the 1-step precharge approach with $V_{DD}$ only. The total energy consumed to precharge the array becomes

$$E_{CS} = (2^{B-2} - 1)CV_{DD}{}^2 + CV_{CM}{}^2. \tag{3.15}$$

Another approach to minimizing the total capacitance and energy was demonstrated in [8], in which the array is implemented using multi-step LSB-precharge. The procedure is depicted in Fig. 3.10c for 2 steps. In the first step, all the MSB capacitors are charged to $V_{DD}$, similarly to the 1-step procedure. Afterward, the MSB switches are opened, and, in the case of the 2-step precharge, the charge of the LSB capacitor is shared with an auxiliary capacitor. Since the LSB and the auxiliary capacitors have the same capacitance value, the charge is split equally between them. The outcome is similar to using $V_{DD}/2$ as an auxiliary supply. The principle is extensible to more steps. Figure 3.10d shows the precharge using three steps. In this case, the voltage on the LSB capacitor is $V_{DD}/4$ at the end of the procedure. The total energy consumed is given by (3.16), where $N_{steps}$ indicates the number of steps employed.

$$E_{CS} = (2^{B-N_{steps}} - 1)CV_{DD}{}^2. \tag{3.16}$$

Furthermore, while pursuing to minimize the power consumption and the area of a CS-ADC implementation, it is possible to employ a combination of the multi-step LSB-precharge and $V_{CM}$-based schemes to create a hybrid alternative.

Another important feature of the CS switching scheme is that the input range of the ADC is not confined to the value of $V_{PC}$. In the CR-based architectures, the range of voltages that the ADC can treat is limited to the values of the voltage sources used to switch the bottom plates of the capacitors, e.g. $V_{REF}$ and ground. Since the operation of the CS-based topologies depends strictly on values of charge, it is possible to trade-off voltage and capacitance in the precharge cycle, i.e. $Q = CV$, while still maintaining control over the input range of the ADC. This approach is depicted in Fig. 3.11.

In the two simplified 3-bit implementations, the differential input range was maximized and lays between $-V_{REF}$ and $V_{REF}$. While using $V_{DD}$ to precharge the array (Fig. 3.11a), the proper size of the TH capacitors is $8C$, and the energy

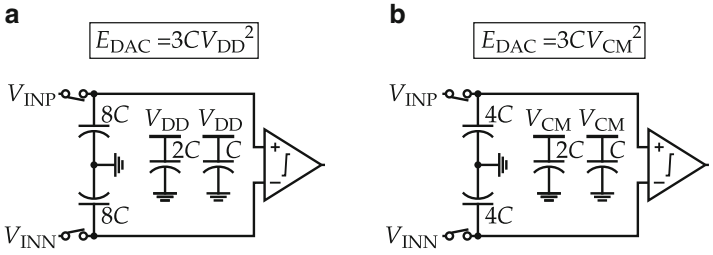



**Fig. 3.11** CS-ADCs with different precharge voltage $V_{PC}$ and same input range: (**a**) $V_{PC} = V_{DD}$ and $C_{TH} = 8C$ and (**b**) $V_{PC} = V_{CM}$ and $C_{TH} = 4C$

consumption becomes $3CV_{DD}{}^2$. In the second implementation (Fig. 3.11b), the array is precharged to $V_{CM} = V_{DD}/2$. As the charge on the array drops to half, the charge on the TH must follow the same proportion to avoid saturation on the ADC, for the same input range. The proper size of the TH capacitors becomes $4C$, and the energy consumption is reduced to

$$E_{CS} = 3C\left(\frac{V_{DD}}{2}\right)^2 = \frac{3CV_{DD}{}^2}{4}. \tag{3.17}$$

The latter implementation brings two advantages over the former: reduction of the TH capacitance to one-half and a reduction by a factor of four in the power consumption. It is also possible to set the input range of the ADC to values larger than the supply voltage, as will be demonstrated in Chap. 7.

## 3.5 Comparison of Reviewed Switching Schemes

To compare the energy efficiency of the reviewed switching schemes, we rely on behavioral simulations. The Python[1]-codes used to simulate the switching schemes are listed in appendix of this chapter. Figure 3.12 shows the simulated energy consumed to convert all the codes in 10-bit implementations of ADCs employing conventional, monotonic, MCS, and tri-level CR switching schemes. The same figure includes the energy for the CS-ADC with four variations of the precharge cycle: 1-step with $V_{PC} = V_{DD}$ (Fig. 3.10a), 1-step with $V_{PC} = V_{CM}$ only for the LSB (Fig. 3.10b), 3-step with $V_{PC} = V_{DD}$ and 3-step with $V_{PC} = V_{CM}$ (Fig. 3.10d). The energy is normalized to the unit capacitance and the reference source ($CV_{REF}{}^2$). It is important to note that this analysis does not consider matching or noise requirements in the ADC. For example, a topology with small DAC capacitance may not fulfill noise requirements while another with larger capacitance does. In practice, to fairly compare the energy between the two implementations, one would need to increase the unit capacitance of the former implementation. For the sake of clarity, these requisites are left out, and it is assumed the same unit capacitance for all the implementations.

It is noticeable that in the CR-based implementations, the energy is strongly dependent on the output code. For all the reviewed switching schemes, the code with the highest energy consumes more than twice than the code with the lowest energy. In the CS-based implementations, the energy is constant for all the output codes, which matches with the mentioned pattern of energy consumption in the topology, where only the precharge cycle drains current from the supply sources.

[1]Python is a widely used high-level, general-purpose, interpreted programming language that emphasizes code readability and allows programmers to express concepts in fewer lines of code. Open-source libraries are available to deal with numeric and symbolic computations.

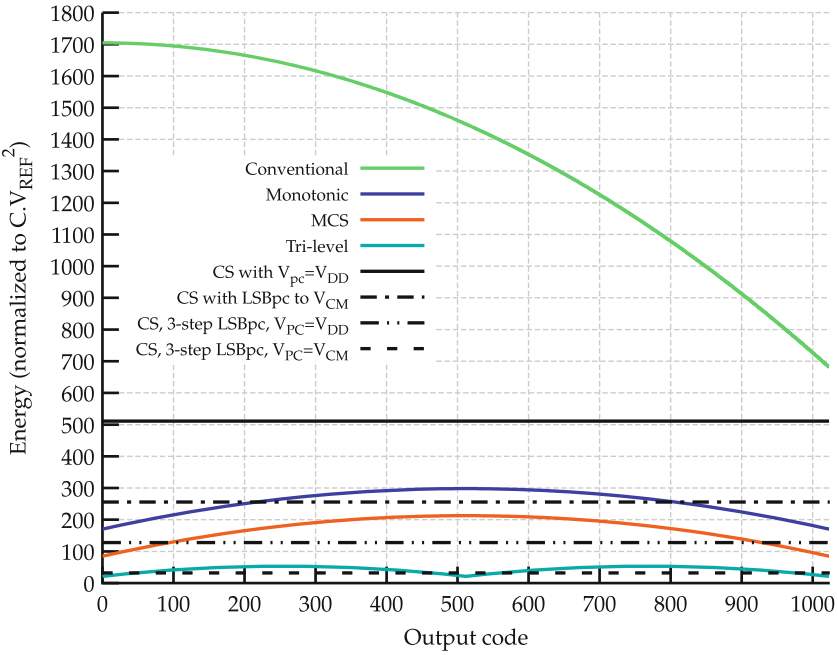



**Fig. 3.12** Comparison of energy consumption as function of the output code of several switching schemes for SAR ADCs

In Table 3.1, the switching schemes reviewed so far are grouped according to the operation principle that they utilize, and their differences are summarized. The table includes the average energy consumption normalized to the conventional switching scheme. The table shows an improvement of more than one order of magnitude in the average energy consumption of the most economic switching schemes when compared to the reference implementation. Also, great enhancements are noticeable for the DAC and input capacitances in the most recent implementations, when compared to the reference CR-ADC. The DAC capacitance may reach 1/16 of the conventional CR-ADC for the smallest CS-ADCs, and 1/4 for the smallest CR-ADCs. The input capacitance, in the case of the CR-ADCs, is the same as the DAC capacitance, since the TH functionality is merged in the DAC. The input capacitance is reduced to 1/4 of the reference implementation in the CR and CS-ADCs that perform best in this matter.

The two most efficient implementations, the tri-level CR and the CS with 3-step precharge to $V_{CM}$, present a large dependency on the common-mode voltage. This is much more critical in the case of the CR, which requires the generation of a second voltage level with the same accuracy and noise requirements of the reference voltage. In the case of the CS-ADC, the common-mode voltage assumes the role of a reference voltage, i.e. a single reference voltage with a lower value has to be generated. The tri-level and monotonic CR-ADCs have another significant

**Table 3.1** Comparison of several switching schemes for 10-bit ADC implementations

| | Energy | Common mode voltage | Sensitivity to comparator offset | Requires explicit TH | Dependency on accuracy of $V_{CM}$ | DAC capacitance | Input capacitance | DAC SNR degradation | Differential input range | DAC buffer requirements |
|---|---|---|---|---|---|---|---|---|---|---|
| *Charge redistribution* | | | | | | | | | | |
| Conventional | Reference (100 %) | Constant | Low | No | – | Reference (1×) | Reference (1×) | No | Limited to $2V_{REF}$ | Hard (DAC drains charge in each cycle of the algorithm) |
| Monotonic | 18.74 % | Varying | High | | – | 0.5× | 0.5× | | | |
| MCS | 12.48 % | Constant | Low | | No | 0.5× | 0.5× | | | |
| Tri-level | 3.11 % | Varying | High | | High (all bits except MSB) | 0.25× | 0.25× | | | |
| *Charge sharing, according to the precharge scheme* | | | | | | | | | | |
| 1-step, $V_{PC} = V_{DD}$ | 37.48 % | Constant | High | Yes | – | 0.25× | 1× | Yes | Adjustable (proportional to $C_{TH}/C_{DAC}$) | Soft (DAC drains charge only in the precharge cycle) |
| 1-step, only LSB precharged to $V_{CM}$ | 18.7 % | Constant | High | | Low (only LSB) | 0.125× | 0.5× | | | |
| 3-step, $V_{PC} = V_{DD}$ | 9.31 % | Constant | High | | – | 0.0625× | 0.25× | | | |
| 3-step, $V_{PC} = V_{CM}$ | 2.33 % | Constant | High | | High[a] (all bits except MSB) | 0.0625× | 0.25× | | | |

[a] Obviates the generation of $V_{REF}$

drawback: the common-mode voltage at the inputs of the comparator changes throughout the conversion, modifying its timing and offset characteristics. Most importantly, the time-varying behavior of the comparator offset affects the linearity of the ADC transfer curve.

The CS-ADCs are also sensitive to the comparator offset. However, the cause is unrelated to the common-mode voltage for the CS-ADC, as it remains constant during the conversion. For CS-ADCs, the DAC capacitance is the parameter that presents a dynamic behavior, because capacitors are successively connected during the conversion. While the value of charge varies linearly with the conversion, the changes in the total capacitance make the voltage on the comparator inputs (i.e., the residue voltage) to present a nonlinear behavior. Therefore, the linearity of the ADC transfer curve is only maintained with zero comparator offset. The increase in the total capacitance during the conversion also brings two major disadvantages to the CS-ADC when contrasted to the CR-ADCs. First, the SNR at the comparator inputs is degraded during the conversion. For a sufficiently high number of bits and with rail-to-rail input voltage, the effective capacitance is approximately doubled at the end of the conversion, which means that the voltage drops by half. The practical implication is that the comparator must be designed to more stringent noise requirements. For example, an 8-bit ADC must use a comparator with 9-bit noise performance. The second drawback of the CS switching scheme is that it requires an explicit TH circuitry. The capacitance of the TH is generally larger than the capacitance of the DAC, which implicates in a significant, although not proportional, increase in area. To illustrate, in the two implementations presented in Chaps. 6 and 7, the THs are smaller than the DACs, even though their capacitances are expressively larger. This is because the THs do not need to be designed using unit cells and are not as much sensitive to the parasitics as the DAC, allowing it to be designed with a much higher density of capacitance.

At the same time, the CS-ADC presents two notable advantages to the CR-ADC, which are particularly valuable in LVLP applications. First, the input range is not limited to the value of the reference voltage, as it is in the CR-ADCs. In the CS, the ratio between the charge in the TH and the charge in the DAC dictates the conversion gain, and consequently the input range. This characteristic allows the CS-ADC to operate with over-rail input signals, or to improve the dynamic range without resorting to active amplifiers, as demonstrated in [9]. Second, the requirements for the reference buffer in the CS-ADC are much more flexible, as the architecture permits duty-cycling of the reference generation circuitry or a buffer with higher impedance and lower power consumption. Finally, as disclosed in Chap. 7, the CS architecture does not require linear capacitors in the DAC, contrary to the CR-based ADC.

The comparison reveals that the CS-ADC presents a very compelling advantage from the system-level perspective, as it allows energy saving in the reference generation and better control over the input range of the ADC. On the other hand, to implement competitive designs of CS-ADCs, the major drawbacks have to be mitigated. Only in this case, the topology can be employed with success in real-world applications. A significant part of this work focus on quantifying the impact

of these pitfalls (Chap. 4), while the remaining of the work suggests and validates techniques to alleviate their influence and ease the design of CS-ADCs in LVLP environments (Chaps. 5–7). Before moving forward, it is important to revisit the state of the art in CS-ADCs. Since only very few CS-based ADCs are found in literature, this survey is brief and is presented in the next section.

## 3.6 State of the Art in CS-ADCs

### *3.6.1 Craninckx and van der Plas [8]*

The pioneer work on CS-ADCs, presented in [8], demonstrated a 9-bit ADC operating at 50 MSps. To reach that sampling speed, the authors utilized an asynchronous logic controller, avoiding a clock signal with a frequency higher than the sampling frequency. The front-end of the design consisted of a pipeline of two passive TH circuits that enabled to track the input for the next conversion at the same time as the rest of the circuit decides the actual conversion result. This procedure is depicted in Fig. 3.13. The drawback is an additional drop of 6 dB in the SNR evaluated by the comparator, due to the charge sharing between the front-end and back-end THs. Moreover, the capacitive array used a 4-step precharge cycle, leading to a DAC with only 35 unit capacitors. The authors mitigated the sensitivity to comparator offset of the ADC by employing a foreground calibration cycle prior to ADC operation. The offset was adjusted by adding capacitances to the faster branch of the comparator.

### *3.6.2 Giannini et al. [10]*

The CS-ADC design presented in [10] has a stronger focus on reducing the comparator noise. For this purpose, the authors make use of two different comparators: one presenting low-power/high-noise while the other presents high-power/low-noise. By using one redundant comparison and switching to the low-noise mode for the two last comparisons, the circuit can restore the errors induced by noise. The approach is also able to correct moderate static nonlinearity, as far as it is smaller

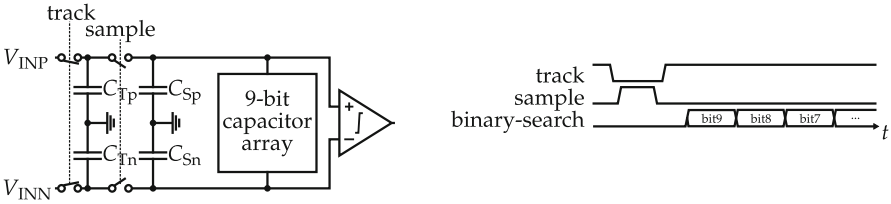



**Fig. 3.13** Pipelined sampling front-end for CS-ADCs

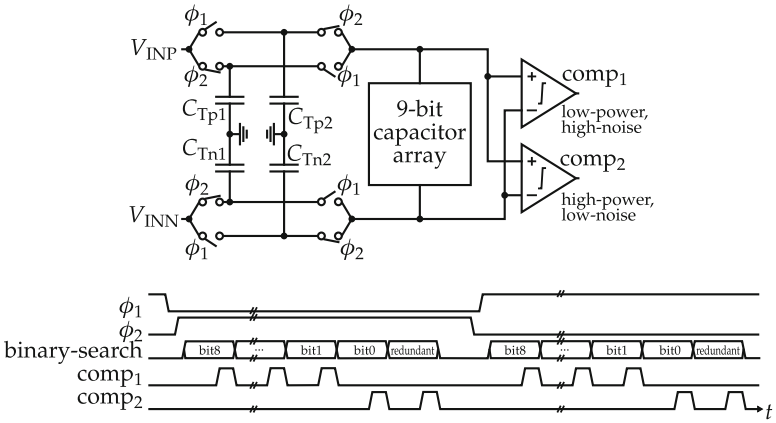



**Fig. 3.14** Time interleaved sampling front-end and noise-reduction technique

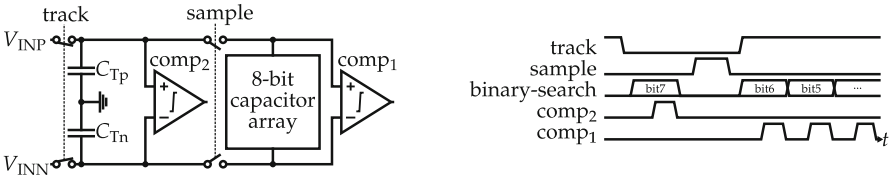



**Fig. 3.15** Pipelined sampling front-end using MSB capacitor as sampling capacitor

than one LSB. The front-end uses two TH circuits that operate in a time-interleaved fashion, mitigating the reduction in SNR that appeared in [8]. The operation of the front-end and the dual-comparator scheme is depicted in Fig. 3.14. Also, the THs include a circuit to lower the common-mode voltage of the signal processed by the comparator, increasing the $V_{GS}$ on the switches and consequently reducing their on-resistance.

### *3.6.3 Tsai et al. [11]*

The design reported in [11] presents a CS-based SAR ADC that uses a full-custom digital controller designed to optimize power consumption. The sampling front-end uses a pipelining approach similar to the one presented in [8], but instead of an additional pair of capacitances, the ADC uses the MSB capacitors of the DAC to sample the charge of the TH. Because of charge sharing, the MSB capacitor must have twice the size it would have in the conventional approach, and an additional comparator is required. On the other hand, the proposed approach reduces the susceptibility to comparator offset, when compared to the original two-step approach [8]. The procedure is depicted in Fig. 3.15. This work uses the

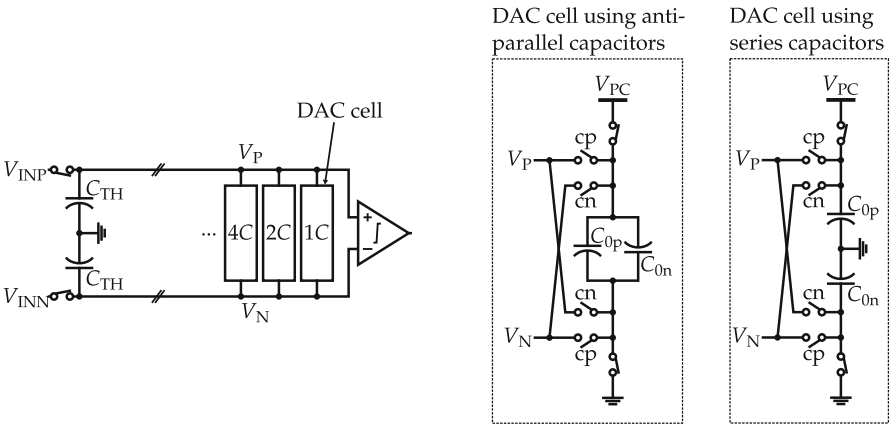



**Fig. 3.16** Different implementations of DAC cells

common-mode voltage as the precharge voltage for the capacitive array. The authors also brought a modification in the DAC cells. In this work, the DAC cells use two capacitors in series, while the previous implementations use only one capacitor or two capacitors in anti-parallel, in order to balance the parasitics. The two arrangements are shown in Fig. 3.16. Since the voltages on the comparator inputs converge to a positive voltage at the end of the conversion, the series connection permits some energy recycling for the $C_{0p}$ capacitor. This is because $C_{0p}$ only needs to be charged from this residual charge to $V_{PC}$, while in the parallel connection no charge is recycled, and both capacitors need to be charged from 0 to $V_{PC}$. While using the series-arranged DAC cell and $V_{CM}$ as the precharge voltage of the DAC, this work was able to recycle approximately 67 % of the energy in the DAC when compared to the designs presented in [8] and [10], at the expense of varying common-mode at the comparator inputs.

### *3.6.4 Malki et al. [9, 12]*

The works [9] and [12] report the same design that uses current integration in the sampling front-end. Instead of feeding the TH capacitors directly from the input signal, which generally requires a power-hungry voltage buffer, the input voltage is converted into a current by a variable-gain transconductor, and this current is integrated as a charge in the sampling capacitors. The principle of the transconductor-based TH and a diagram of the ADC are shown in Fig. 3.17.

By using a programmable TH transconductance, the authors were able to increase the dynamic range of the ADC, which is a very convenient feature for RF applications. Additionally, the authors employed nonlinear MOS capacitances in the TH for passive amplification, relaxing the noise requirements of the comparator.

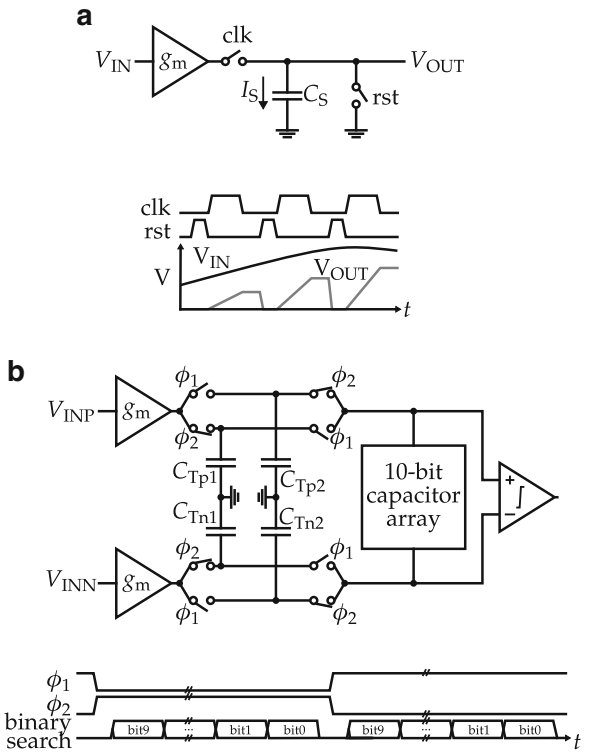



**Fig. 3.17** Sampling front-end using a current-integrating TH: (**a**) operation principle and waveforms; (**b**) ADC implementation

An important characteristic of the transconductor-based TH is that the output amplitude is dependent on the input frequency. The current integration introduces a cardinal sine function behavior, $\sin(x)/x$, often denoted as sinc. The sinc function presents notches at multiples of $1/T_{\text{integ}}$, being $T_{\text{integ}}$ the integration period, and its shape is shown in Fig. 3.18. Thus, in the extreme case of the integration time being an exact multiple of the input signal period, the output of the transconductor-based TH is zero. In order to be used effectively in a Nyquist-ADC, a non-trivial calibration scheme is required to compensate for the TH transfer characteristic. On the other hand, if employed in an RF-chain, the sinc transfer function may be helpful in attenuating interferers and blockers at higher frequencies. A more detailed explanation can be found in [12].

### *3.6.5 Summary*

This section summarizes how the state-of-the-art CS-ADCs deal with the susceptibility of the topology to comparator offset and noise. In [8], foreground calibration of the comparator offset is employed. The offset is compensated by retarding

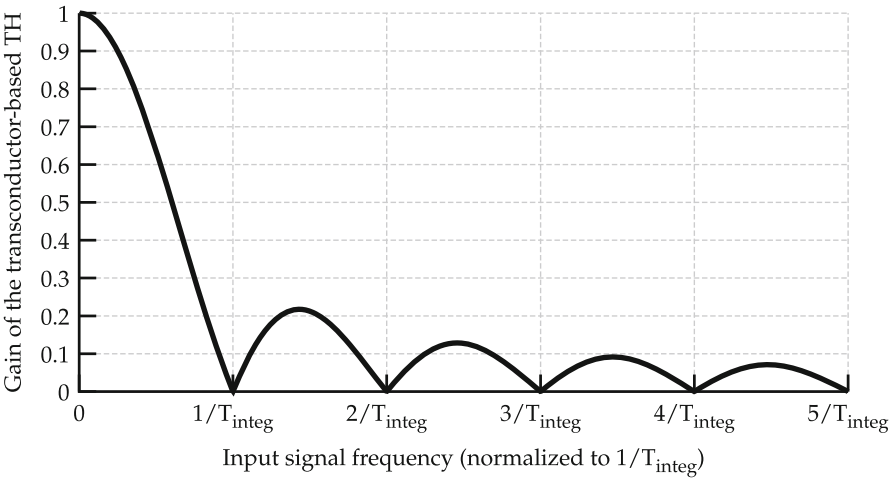



**Fig. 3.18** Input–output transfer characteristic of the transconductor-based TH

the fastest branch of the comparator with digitally configurable capacitances on critical nodes. Thus, the comparator becomes slightly slower after calibration, which is particularly unfavorable in designs for LVLP applications that already struggle with limited speed due to low voltage supplies. Calibration takes place in the foreground and consequently does not track PVT variations. Foreground calibration also requires the ADC to be halted and fed with a known input signal, complicating the design at system-level. The design reported in [10] uses a similar foreground calibration technique to [8], applied to two comparators instead of one. Two comparators are used to mitigate noise, one with low-power/high-noise and other with high-power/low-noise. One redundant conversion step with the latter comparator enables to recover noise-induced errors. The front-end uses time-interleaved THs, suppressing the voltage attenuation brought by the two-step front-end [8]. In [11], a two-step front-end that uses the MSB DAC capacitor instead of an additional TH is used. This also suppresses the attenuation brought by the original two-step front-end of [8]. The publication does not disclose if comparator offset calibration is employed. In [9, 12], foreground calibration of comparator offset is once again utilized. The same design uses a current-integrating TH with passive amplification based on MOS capacitors to improve noise tolerance. However, this approach of front-end is unfeasible for applications that process the full Nyquist band, due to the sinc-shaped frequency response of the front-end.

In conclusion, an alternative to mitigate the impact of comparator offset and noise on CS-ADCs that suffice real-world LVLP applications is still lacking in the literature. The techniques proposed in Chaps. 6 and 7 aim to fulfill this shortfall.

## Appendix: Voltage and Energy in CR ADCs

We developed a simplified and a general model for calculation of voltage and energy in capacitive DAC topologies. These models were used for the calculations presented in this chapter. The Python codes for these models are given at the end of this appendix.

### *Models for Voltage and Energy in CR ADCs*

We consider that the comparator inputs present infinite impedance and, therefore, drain no current from the DAC outputs. Following this assumption, the comparator is removed from the analysis, without compromising the accuracy of the models. Most of the switching schemes use only two voltage levels as reference voltages in the DAC arrangement, namely ground and $V_{\mathrm{REF}}$. For these topologies, we may rely on a simplified model to compute the DAC output voltage and the energy. Some other switching schemes, on the other hand, use another intermediate reference voltage, such as $V_{\mathrm{CM}}$. For the latter, we devise a general model that supports an arbitrary number of references and concurrent switching of multiple capacitors.

#### Simplified Voltage Model for CR DACs

Since the CR DACs are based on switched capacitors, they do not present static power consumption. Therefore, we are only interested in the power consumption that takes place in the transitions between states. The circuit in Fig. 3.19 provides a good representation of a CR DAC transitioning between arbitrary states. In the diagram, $C_{\mathrm{T}}$ represents the sum of capacitances that has the bottom plate connected to $V_{\mathrm{REF}}$ and remain in this condition during the transition. The capacitor $C_{\mathrm{B}}$ represents the sum of capacitances with bottom plates connected to ground that is left unchanged during the transition. The remaining capacitor, $C_{\mathrm{X}}$, represents the capacitance being switched. The switch $S_1$ is initially connected to ground and is switched to $V_{\mathrm{REF}}$ at time $t = 0$. All the voltages and currents on the circuit totally settle at time $t_s$. We can discretize the time domain and define $V_{\mathrm{X}}(0) \equiv V_{\mathrm{X}}[i-1]$ and $V_{\mathrm{X}}(t_s) \equiv V_{\mathrm{X}}[i]$. Since there is no current path for the top plates of the capacitors, the charge at the node $V_{\mathrm{X}}$ is conserved.

$$Q_X[0] = Q_X[1]. \tag{3.18}$$

Equation (3.18) can be rewritten as follows:

$$\begin{aligned} &C_{\mathrm{X}} V_{\mathrm{X}}[i-1] + C_{\mathrm{T}} \left(V_{\mathrm{X}}[i-1] - V_{\mathrm{REF}}\right) + C_{\mathrm{B}} V_{\mathrm{X}}[i-1] \\ &\quad = C_{\mathrm{X}} \left(V_{\mathrm{X}}[i] - V_{\mathrm{REF}}\right) + C_{\mathrm{T}} \left(V_{\mathrm{X}}[i] - V_{\mathrm{REF}}\right) + C_{\mathrm{B}} V_{\mathrm{X}}[i]. \end{aligned} \tag{3.19}$$

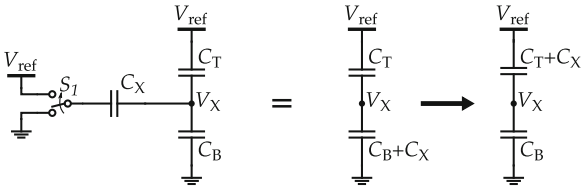



**Fig. 3.19** Circuit representation of a CR DAC transitioning between arbitrary states

Solving (3.19) for $V_X[i]$ leads to

$$V_X[i] = V_X[i-1] + \frac{C_X}{C_X + C_T + C_B} V_{REF}. \tag{3.20}$$

Similarly, if $C_X$ is switched from $V_{REF}$ to ground, (3.20) becomes

$$V_X[i] = V_X[i-1] - \frac{C_X}{C_X + C_T + C_B} V_{REF}. \tag{3.21}$$

To quantify the energy consumed from the reference source when $S_1$ switches ground to $V_{REF}$, let us again consider the simple circuit of Fig. 3.19, assuming that at the instant $t = 0$, $V_X$ is equal to $V_X(0)$ and $S_1$ is disconnected from ground and connected to $V_{REF}$. The energy spent to switch the bottom plate of $C_X$ to $V_{REF}$ is given in (3.22), considering that $V_{REF}$ is a dc voltage source and the current flowing through its terminals is $I_{REF}$.

$$E_{REF} = V_{REF} \int_0^{t_s} I_{REF}\, dt. \tag{3.22}$$

The current $I_{REF}$ can be described in terms of the charge drainded to the capacitors connected to $V_{REF}$. This is seen in (3.23), where $Q_{C_X}$ and $Q_{C_T}$ are the charges stored in $C_X$ and $C_T$, respectively.

$$I_{REF}(t) = -\left(\frac{dQ_{C_X}}{dt} + \frac{dQ_{C_T}}{dt}\right). \tag{3.23}$$

Rewriting (3.22) in terms of (3.23) yields in

$$E_{REF}[i] = -V_{REF} \int_0^{t_s} \left(\frac{dQ_{C_X}}{dt} + \frac{dQ_{C_T}}{dt}\right) dt \tag{3.24}$$

$$= -V_{REF} \left( \int_{Q_{C_X}(0)}^{Q_{C_X}(t_s)} dQ_{C_X} + \int_{Q_{C_T}(0)}^{Q_{C_T}(t_s)} dQ_{C_T} \right). \tag{3.25}$$

Integrating, we arrive at

$$E_{REF}[i] = -V_{REF}\left[(Q_{C_X}[i] - Q_{C_X}[i-1]) + (Q_{C_T}[i] - Q_{C_T}[i-1])\right]. \tag{3.26}$$

$$E_{\mathrm{REF}}[i] = -V_{\mathrm{REF}}C_{\mathrm{X}}\left[(V_{\mathrm{X}}(t_s) - V_{\mathrm{REF}}) - (V_{\mathrm{X}}[i-1])\right] \\ -V_{\mathrm{REF}}C_{\mathrm{T}}\left[(V_{\mathrm{X}}[i] - V_{\mathrm{REF}}) - (V_{\mathrm{X}}[i-1] - V_{\mathrm{REF}})\right]. \tag{3.27}$$

Solving (3.27) leads to

$$E_{\mathrm{REF}}[i] = V_{\mathrm{REF}}C_{\mathrm{X}}\left(V_{\mathrm{REF}} + V_{\mathrm{X}}[i-1] - V_{\mathrm{X}}[i]\right) + V_{\mathrm{REF}}C_{\mathrm{T}}\left(V_{\mathrm{X}}[i-1] - V_{\mathrm{X}}[i]\right). \tag{3.28}$$

Plugging (3.20) into (3.28) yields in

$$E_{\mathrm{REF}}[i] = \frac{C_{\mathrm{B}}C_{\mathrm{X}}}{C_{\mathrm{B}} + C_{\mathrm{T}} + C_{\mathrm{X}}}{V_{\mathrm{REF}}}^2. \tag{3.29}$$

Equations (3.20) and (3.29) may be readily used to calculate the top-plate voltage and the energy in the DAC topologies based on the principle of charge redistribution, respectively, if only $V_{\mathrm{REF}}$ and ground are employed as references. Additionally, the model is only valid if only one capacitor is switched at each transition.

### General Voltage Model for CR DACs

In order to develop a broader model, consider the capacitive array in Fig. 3.20, where the bottom plates of $k$ capacitors are switched at the same time from their initial voltages $V_0[i-1], V_1[i-1] \ldots V_k[i-1]$ to arbitrary voltage levels $V_0[i], V_1[i] \ldots V_k[i]$. Again, the principle of charge conservation holds, and the charge of the $i$-th cycle is the same as the previous cycle:

$$Q_X[i] = Q_X[i-1]. \tag{3.30}$$

By expanding (3.30) similarly as in (3.18) and solving for $V_{\mathrm{X}}[i]$, we arrive at (3.31).

$$V_{\mathrm{X}}[i] = \frac{C_0\left(V_0[i] - V_0[i-1] + V_{\mathrm{X}}[i-1]\right) + \ldots + C_k\left(V_k[i] - V_k[i-1] + V_{\mathrm{X}}[i-1]\right)}{C_0 + C_1 + \ldots + C_k}. \tag{3.31}$$

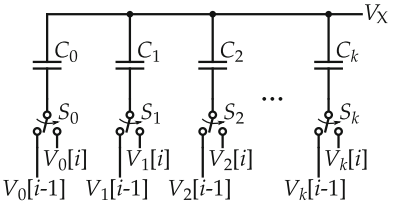



**Fig. 3.20** Circuit used to develop the general voltage model for CR DACs

By collecting the similar terms together, (3.31) may be rewritten as (3.32), wherein $N$ indicates the total number of reference sources employed.

$$V_{\mathrm{X}}[i] = \frac{\sum_{j=0}^{N-1} C_j(V_{\mathrm{X}}[i-1] + V_j[i] - V_j[i-1])}{\sum_{j=0}^{N-1} C_j}. \tag{3.32}$$

In order to compute the energy, we assume that the only voltage levels allowed on the bottom plates of the capacitors are the reference voltages. This simplification does not pose any limitation on the analysis, as this happens naturally in a charge-redistribution DAC. These voltages may comprehend any finite number of voltage sources $V_{\mathrm{REF0}}, V_{\mathrm{REF1}} \ldots V_{\mathrm{REF},k}$, even though practical implementations of charge-redistribution DACs use 2 or 3 (i.e., $V_{\mathrm{DD}}$, ground and the common-mode voltage $V_{\mathrm{CM}}$). In the model for voltage in the DAC, we considered that the capacitances are fixed and that the voltages on their bottom plates vary in time. On the other hand, for the energy model, we will use the assumption of fixed and known voltage levels and compute the variation of the capacitance connected to each one of those voltage sources. This analogy is depicted in Fig. 3.21 and simplifies the calculation of energy when multiple references or capacitors are switched concurrently. Also, this change of standpoint is not harmful to the analysis because we care most about the amount of energy that is spent from $V_{\mathrm{REF}}$, and have limited interest in the distribution of currents among the capacitors.

For every state-transition in the DAC, the capacitance connected to a given voltage source $V_{\mathrm{REF},j}$ can be split up into two components: $C_{j,\mathrm{s}}$, which is the capacitance that was maintained static since the previous cycle; and $C_{j,\mathrm{d}}$, which is the capacitance that was just connected to the reference voltage in the current cycle. Additionally, $C_{j,\mathrm{d}}$ is further subdivided according to the voltage that was previously applied to the bottom plate of these capacitors, so that $C_{j,\mathrm{d1}}$ corresponds to all the capacitors previously connected to $V_{\mathrm{REF1}}$, $C_{j,\mathrm{d2}}$ corresponds to all the capacitors previously connected to $V_{\mathrm{REF2}}$, and so on. Note that if a capacitor is disconnected from the voltage source in the transition of states, it does not contribute to its energy consumption. The assignment of capacitance values is exemplified for $V_{\mathrm{REF0}}$ in Fig. 3.22.

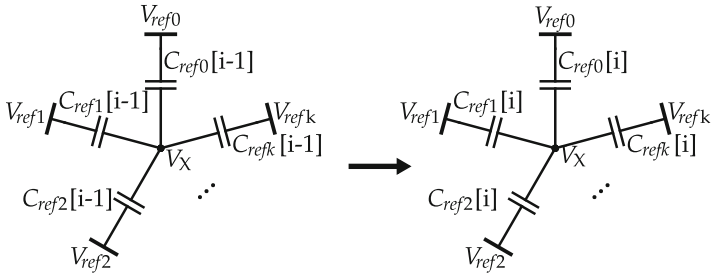



**Fig. 3.21** Circuit used to develop the general energy model for CR DACs

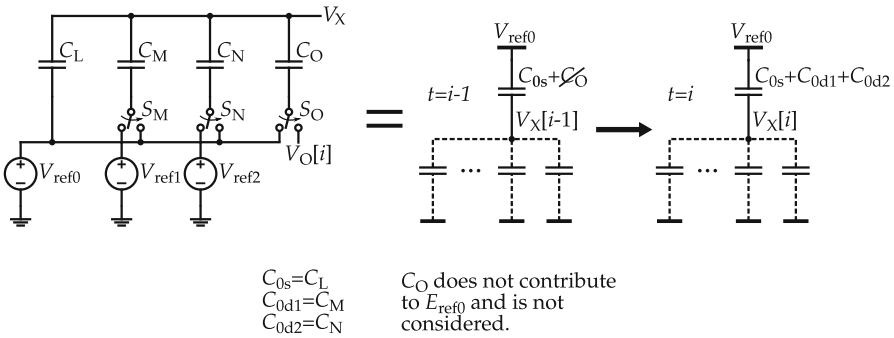



**Fig. 3.22** Example of capacitance values assignment for $V_{REF0}$

At each cycle, the energy drawn from $V_{REF,j}$ is the sum of the energies spent to charge $C_{j,s}$ and $C_{j,d0}, C_{j,d1} \ldots C_{j,d,k}$. Thus, the energy consumed in the $i$-th cycle is given by

$$E_{REF,j}[i] = E_{j,s}[i] + \sum_{k=0}^{N-1} E_{j,d,k}[i]. \tag{3.33}$$

Following a similar reasoning to that presented in Sect. 3.6.5, (3.33) may be rewritten as

$$E_{REF,j}[i] = -V_{REF,j}\left(\int_{Q_{C_{j,s}}[i-1]}^{Q_{C_{j,s}}[i]} \mathrm{d}Q_{C_{j,s}} + \sum_{k=0}^{N-1} \int_{Q_{C_{j,d,k}}[i-1]}^{Q_{C_{j,d,k}}[i]} \mathrm{d}Q_{C_{j,d,k}}\right). \tag{3.34}$$

By solving and simplifying, we arrive at

$$E_{REF,j}[i] = -V_{REF,j}\left(C_{j,s}\left(V_X[i-1] - V_X[i]\right) + \sum_{k=0}^{N-1} C_{j,d,k}\left(V_{j,d,k}[i-1]\right.\right.$$
$$\left.\left. + V_X[i] - V_X[i-1] - V_{REF,j}\right)\right). \tag{3.35}$$

The total energy consumed during a transition is found summing the contribution of all the reference sources.

$$E_{total} = \sum_{j=0}^{N-1} E_{REF,j}. \tag{3.36}$$

Therefore, (3.32), (3.35), and (3.36) can be employed in order to find the total energy consumption and the contribution of all the reference sources in the ADC.

The general model requires solving slightly more complicated equations than the proposed simplified model to compute the energy of switching schemes. On the other hand, this general model is easier to be implemented algorithmically on a computer, which makes it a good alternative to automate the computations. One of the advantages of automating the calculations is that it makes easier to extend the analysis to new switching schemes as they appear. Based on the general model, we wrote a software library that takes care of the low-level computations once the switching scheme is algorithmically described. This library was employed to perform the behavioral simulations and extract the results of Chap. 3. The codes are listed in the next section.

## *Codes*

**Listing 3.1** cap_array.py

```
# -*- coding: utf-8 -*-
"""
Generalized model of energy and voltage in a CR DAC.
@author: Taimur Rabuske
"""

class cap_array:
    def __init__(self):
        self.ct=0
        self.cb=0
        self.c=dict([])
        self.v=dict([])
        #Add ground node
        self.v["gnd"]={"value":0,"energy":0}
        self.prev_vout=0
        self.vout=0

    #Add a capacitor to the array
    def addC(self,name,value,initial="gnd"):
        self.c[name]={"value":value, "node":initial,
            ↪ "initial":initial}
        return 0

    #Add a voltage source (reference) to the array.
    def addV(self,name,value, energy=0):
        self.v[name]={"value":value, "energy":energy}

    #Reset all bottom plates to their initial conditions
    def resetAll(self):
        for cap in self.c:
            self.c[cap]["node"]= self.c[cap]["initial"]

    #Sample a voltage into the top plate
    def sampleTopPlate(self,vin):
        self.vout=vin
```

```
35          return vin
36
37      #Process the charge. The input of this function is a pair
            ↪ of values in the form [C,N], where C is the name of
            ↪ the capacitor to be switched, and N is the node to
            ↪ which it is switched, e.g. Vref or gnd. The method
            ↪ returns the DAC voltage and energy spent in the
            ↪ current cycle.
38      def switchC(self,*args):
39          #Store previous voltage to the prev_node key in the
                ↪ dictionary.
40          for i in self.c:
41              self.c[i]["prev_node"]=self.c[i]["node"]
42          #Connect capacitances to assigned nodes
43          for i in args:
44              cap=i[0]
45              v=i[1]
46              if v not in self.v:
47                  print "Voltage source does not exist"
48              self.c[cap]["node"]=v
49
50  #         Compute the new voltage in the DAC, given by:
51  #           x
52  #          ___
53  #          \
54  #           \    (V_DAC0.C0(i) + C(i).V_ref(i) - C0(i).V_ref0(i))
55  #           /
56  #          /__
57  #
58  #         i = 0
59  # V_DAC= ____________________________________________________________
60  #                                  x
61  #                                 ___
62  #                                 \
63  #                                  \    C(i)
64  #                                  /
65  #                                 /__
66  #
67  #                                i = 0
68          tmp=0
69          #compute the summation in the numerator
70          for i in self.c:
71              tmp=tmp + self.c[i]["value"] *
                    ↪ (self.v[self.c[i]["node"]]["value"] -
                    ↪ self.v[self.c[i]["prev_node"]]["value"] +
                    ↪ self.vout)
72          #compute denominator
73          c_total=sum([self.c[j]["value"] for j in self.c])
74          #store previous dac voltage
75          self.prev_vout=self.vout
76          #set output voltage
77          self.vout=tmp/c_total
78
79          #Calculate the energy spent in charging the DAC
```

```
        E=dict([])
        E_total=0
        #Enumerate all the capacitors connected to each one of
            ↪ the voltage sources, and calculate all the
            ↪ parameters for each one of them.
        for vs in self.v:
            #c_just_connected is an array with all the
                ↪ capacitors that were connected on the
                ↪ actual clock cycle
            c_just_connected=[]
            #c_still is the capacitance that was already
                ↪ connected on previous cycle
            c_still=[]
            #find C_just_connected and c_still
            for cap in self.c:
                if self.c[cap]["node"]==vs:
                    if self.c[cap]["prev_node"]==vs:
                        c_still.append(self.c[cap])
                    else:
                        c_just_connected.append(self.c[cap])
            c_still_sum=sum([k["value"] for k in c_still])
            #E_still is the energy required to move the
                ↪ already connected capacitors from V_dac0 to
                ↪ V_dac
            E_still= self.v[vs]["value"] * c_still_sum *
                ↪ (self.prev_vout-(self.vout))
            #E_just_connected is the energy required to charge
                ↪ the bottom plate of the just connected
                ↪ capacitor from the previous voltage (e.g.
                ↪ ground) to the reference voltage.
            E_just_connected=0
            for j in c_just_connected:
                E_just_connected= E_just_connected -
                    ↪ self.v[vs]["value"]*j["value"] *
                    ↪ (self.vout-self.v[vs]["value"] -
                    ↪ (self.prev_vout-self.v[j["prev_node"]]
                    ↪ ["value"]))
            E[vs]=E_just_connected+E_still
            #E_total is the total energy drained from all the
                ↪ reference sources.
            E_total=E_total+E[vs]
            self.v[vs]["energy"]=self.v[vs]["energy"]+E[vs]
        return self.vout,E_total
```

**Listing 3.2** sar_adc_conventional.py

```
#!/usr/bin/env python2.7
# -*- coding: utf-8 -*-
"""
Model of Conventional CR SAR switching scheme
@author: Taimur Rabuske
"""

```

```
8  from __future__ import division
9  from sympy import symbols, simplify
10 from cap_array import cap_array
11
12 class sar_adc_conventional:
13     def __init__(self,Vref=1.0,bits=3,Cu=1e-15):
14         self.Vref=Vref
15         self.bits=bits
16         self.Cu=Cu
17         self.cap_arrayP=cap_array()
18         self.cap_arrayN=cap_array()
19         self.cap_arrayP.addV("vref",self.Vref)
20         self.cap_arrayN.addV("vref",self.Vref)
21         self.cap_arrayP.addC("dummy",self.Cu,"gnd")
22         self.cap_arrayN.addC("dummy",self.Cu,"vref")
23         for i in xrange(self.bits):
24             self.cap_arrayP.addC(i,2**i*self.Cu,"gnd")
25             self.cap_arrayN.addC(i,2**i*self.Cu,"vref")
26     def convert(self,Vin,return_dac=0):
27         out=0
28         Etot=0
29         dac=[]
30         step=-1
31         self.cap_arrayP.resetAll()
32         self.cap_arrayN.resetAll()
33         VdacP=self.cap_arrayP.sampleTopPlate(-Vin/2-self.Vref/2)
34         VdacN=self.cap_arrayN.sampleTopPlate(Vin/2+self.Vref/2)
35         dac.append([step,"sample",self.Vref/2,self.Vref/2])
36         step=step+1
37         switch_backP=0
38         switch_backN=0
39         for k in range(self.bits-1,-1,-1):
40             if switch_backP:
41                 VdacP,EP=self.cap_arrayP.switchC([k,"vref"],
                       ↪ switch_backP)
42                 VdacN,EN=self.cap_arrayN.switchC([k,"gnd"],
                       ↪ switch_backN)
43             else:
44                VdacP,EP=self.cap_arrayP.switchC([k,"vref"])
45                VdacN,EN=self.cap_arrayN.switchC([k,"gnd"])
46             if VdacP-VdacN<0:
47                 out=out+2**k
48                 switch_backP=0
49                 switch_backN=0
50             else:
51                 switch_backP=[k,"gnd"]
52                 switch_backN=[k,"vref"]
53             Etot=Etot+EP+EN
54             dac.append([step,"b_"+str(k), self.Vref/2+VdacP,
                   ↪ self.Vref/2+VdacN])
55             step=step+1
56         dac.append([step,"", self.Vref/2+VdacP,
               ↪ self.Vref/2+VdacN])
57         if return_dac:
```

```
            return dac
        else:
            return out,Etot
```

**Listing 3.3** sar_adc_mcs.py

```
#!/usr/bin/env python2.7
# -*- coding: utf-8 -*-
"""
Model of MCS CR SAR switching scheme
@author: Taimur Rabuske
"""
from __future__ import division
from sympy import symbols, simplify
from cap_array import cap_array

class sar_adc_mcs:
    def __init__(self,Vref=1.0,bits=3,Cu=1e-15):
        vref,vin=symbols("vref,vin")
        self.Vref=Vref
        vref=self.Vref
        self.bits=bits
        self.Cu=Cu
        self.cap_arrayP=cap_array()
        self.cap_arrayN=cap_array()
        self.cap_arrayP.addV("vref",self.Vref)
        self.cap_arrayN.addV("vref",self.Vref)
        self.cap_arrayP.addV("vcm",self.Vref/2)
        self.cap_arrayN.addV("vcm",self.Vref/2)
        self.cap_arrayP.addC("dummy",self.Cu,"vcm")
        self.cap_arrayN.addC("dummy",self.Cu,"vcm")
        for i in xrange(self.bits-1):
            self.cap_arrayP.addC(i,2**i*self.Cu,"vcm")
            self.cap_arrayN.addC(i,2**i*self.Cu,"vcm")
    def convert(self,Vin,return_dac=0):
        out=0
        Etot=0
        dac=[]
        step=0
        self.cap_arrayP.resetAll()
        self.cap_arrayN.resetAll()
        VdacP=self.cap_arrayP.sampleTopPlate(Vin/2)
        VdacN=self.cap_arrayN.sampleTopPlate(-Vin/2)
        dac.append([step, "+b_9", self.Vref/2+VdacP,
            ↪ self.Vref/2+VdacN])
        step=step+1
        for k in range(self.bits-2,-1,-1):
            if VdacP-VdacN<0:
                VdacP,EP=self.cap_arrayP.switchC([k,"vref"])
                VdacN,EN=self.cap_arrayN.switchC([k,"gnd"])
            else:
                out=out+2**(k+1)
                VdacP,EP=self.cap_arrayP.switchC([k,"gnd"])
```

```
                VdacN,EN=self.cap_arrayN.switchC([k,"vref"])
            Etot=Etot+EP+EN
            dac.append([step, "b_"+str(k), self.Vref/2+VdacP,
                ↪ self.Vref/2+VdacN])
            step=step+1
        dac.append([step,"", self.Vref/2+VdacP,
            ↪ self.Vref/2+VdacN])
        if VdacP-VdacN>0:
            out=out+2**0
        if return_dac:
            return dac
        else:
            return out,Etot
```

**Listing 3.4** sar_adc_monotonic.py

```
#!/usr/bin/env python2.7
# -*- coding: utf-8 -*-
"""
Model of Monotonic CR SAR switching scheme
@author: Taimur Rabuske
"""
from __future__ import division
from sympy import symbols, simplify
from cap_array import cap_array

class sar_adc_monotonic:
    def __init__(self,Vref=1.0,bits=3,Cu=1e-15):
        self.Vref=Vref
        self.bits=bits
        self.Cu=Cu
        self.cap_arrayP=cap_array()
        self.cap_arrayN=cap_array()
        self.cap_arrayP.addV("vref",self.Vref)
        self.cap_arrayN.addV("vref",self.Vref)
        self.cap_arrayP.addC("dummy",self.Cu,"vref")
        self.cap_arrayN.addC("dummy",self.Cu,"vref")
        for i in xrange(self.bits-1):
            self.cap_arrayP.addC(i,2**i*self.Cu,"vref")
            self.cap_arrayN.addC(i,2**i*self.Cu,"vref")
    def convert(self,Vin,return_dac=0):
        out=0
        Etot=0
        dac=[]
        step=0
        self.cap_arrayP.resetAll()
        self.cap_arrayN.resetAll()
        VdacP=self.cap_arrayP.sampleTopPlate(-Vin/2)
        VdacN=self.cap_arrayN.sampleTopPlate(Vin/2)
        dac.append([step,"+b_9", self.Vref/2+VdacP,
            ↪ self.Vref/2+VdacN])
        step=step+1
        for k in range(self.bits-2,-1,-1):
```

```
            if VdacP-VdacN<0:
                out=out+2**(k+1)
                VdacN,EN=self.cap_arrayN.switchC([k,"gnd"])
                EP=0
            else:
                VdacP,EP=self.cap_arrayP.switchC([k,"gnd"])
                EN=0
            Etot=Etot+EP+EN
            dac.append([step,"b_"+str(k), self.Vref/2+VdacP,
                ↪ self.Vref/2+VdacN])
            step=step+1
        dac.append([step,"", self.Vref/2+VdacP,
            ↪ self.Vref/2+VdacN])
        if VdacP-VdacN<0:
            out=out+2**0
        if return_dac:
            return dac
        else:
            return out,Etot
```

**Listing 3.5** sar_adc_cs.py

```
#!/usr/bin/env python2.7
# -*- coding: utf-8 -*-
"""
Model of CS SAR switching scheme
@author: Taimur Rabuske
"""
from __future__ import division
from sympy import symbols, simplify

class sar_adc_cs:
    def __init__(self,Vref=1.0,bits=3,Cu=1e-15):
        self.Vref=Vref
        self.bits=bits
        self.Cu=Cu
        self.Cth=2**(bits-1)*Cu
        self.c=[]
        for i in xrange(self.bits-1):
            self.c.append(2**i*self.Cu)
    def convert(self,Vin,return_dac=0):
        out=0
        Etot=0
        dac=[]
        step=0
        Qth=self.Cth*Vin
        Qtotal=Qth
        Ctotal=self.Cth
        Vdac=Vin
        dac.append([step,"+b_9", self.Vref/2+Vin/2,
            ↪ self.Vref/2-Vin/2])
        step=step+1
        Etot=sum(self.c)*(self.Vref)**2
```

```
        for k in range(self.bits-2,-1,-1):
            if Vdac>0:
                out=out+2**(k+1)
                Qtotal=Qtotal-self.c[k]*self.Vref
                Ctotal=Ctotal+self.c[k]
            else:
                Qtotal=Qtotal+self.c[k]*self.Vref
                Ctotal=Ctotal+self.c[k]
            Vdac=Qtotal/Ctotal
            dac.append([step,"b_"+str(k), self.Vref/2+Vdac/2,
                ↪ self.Vref/2-Vdac/2])
            step=step+1
        dac.append([step,"", self.Vref/2+Vdac/2,
            ↪ self.Vref/2-Vdac/2])
        if Vdac>0:
            out=out+2**0
        if return_dac:
            return dac
        else:
            return out,Etot
```

**Listing 3.6** sim_e.py

```
#!/usr/bin/env python2.7
# -*- coding: utf-8 -*-
"""
Script used to simulate the energy of the SAR ADCs as function
    ↪ of the output codes.
@author: Taimur Rabuske
"""

from __future__ import division
import numpy
import matplotlib.pyplot as plt
from sar_adc_conventional import sar_adc_conventional
from sar_adc_monotonic import sar_adc_monotonic
from sar_adc_mcs import sar_adc_mcs
from sar_adc_cs import sar_adc_cs
from multiprocessing import Pool, Lock

#Simulation parameters
samplingrate=1000.0e3
fin=54.6875e3
pts=2048
window="no"

#ADC characteristics
BITS=10
VREF=1.0

sar_adc_conventional=sar_adc_conventional(bits=BITS,
    ↪ Vref=VREF, Cu=1)
sar_adc_monotonic=sar_adc_monotonic(bits=BITS, Vref=VREF, Cu=1)
```

```
sar_adc_mcs=sar_adc_mcs(bits=BITS, Vref=VREF, Cu=1)
sar_adc_cs=sar_adc_cs(bits=BITS, Vref=VREF, Cu=1)

#mp=multiprocessing
mp=1
plot_results=1

PTS_SIM=2**BITS

y=numpy.linspace(-VREF, VREF,  num=PTS_SIM)

print "\n\nSimulating␣ADCs..."
def convert_adc(zipped_args):
    adc_instance,argument= zipped_args
    result,E=adc_instance.convert(argument)
    return result,E

pool=Pool(processes=8)
if mp:
    out_conventional=pool.map(convert_adc,
        ↪ zip(len(y)*[sar_adc_conventional], y))
    out_monotonic=pool.map(convert_adc,
        ↪ zip(len(y)*[sar_adc_monotonic], y))
    out_mcs=pool.map(convert_adc, zip(len(y)*[sar_adc_mcs], y))
    out_cs=pool.map(convert_adc, zip(len(y)*[sar_adc_cs], y))
else:
    out_conventional=map(convert_adc,
        ↪ zip(len(y)*[sar_adc_conventional], y))
    out_monotonic=map(convert_adc,
        ↪ zip(len(y)*[sar_adc_monotonic], y))
    out_mcs=map(convert_adc, zip(len(y)*[sar_adc_mcs], y))
    out_cs=map(convert_adc, zip(len(y)*[sar_adc_cs], y))
out_sar_adc_conventional= numpy.transpose(out_conventional)[0]
out_sar_adc_monotonic= numpy.transpose(out_monotonic)[0]
out_sar_adc_mcs= numpy.transpose(out_mcs)[0]
out_sar_adc_cs= numpy.transpose(out_cs)[0]
E_conventional= numpy.transpose(out_conventional)[1]
E_monotonic= numpy.transpose(out_monotonic)[1]
E_mcs= numpy.transpose(out_mcs)[1]
E_cs= numpy.transpose(out_cs)[1]

numpy.savetxt("E_conventional.csv", E_conventional,
    ↪ delimiter=",")
numpy.savetxt("E_monotonic.csv", E_monotonic, delimiter=",")
numpy.savetxt("E_mcs.csv", E_mcs, delimiter=",")
numpy.savetxt("E_cs.csv", E_cs, delimiter=",")

if plot_results:
    plt.figure(0)
    plt.xlabel("Vin␣(V)")
    plt.ylabel("Dout")
    plt.subplot(211)
    plt.plot(y,out_sar_adc_conventional)
    plt.plot(y,out_sar_adc_monotonic)
```

```
78      plt.plot(y,out_sar_adc_mcs)
79      plt.plot(y,out_sar_adc_cs)
80      plt.grid(True)
81      plt.subplot(212)
82      plt.plot(out_sar_adc_conventional,E_conventional)
83      plt.plot(out_sar_adc_monotonic,E_monotonic)
84      plt.plot(out_sar_adc_mcs,E_mcs)
85      plt.plot(out_sar_adc_conventional,E_cs)
86      plt.grid(True)
87      plt.show()
```

**Listing 3.7** sim_wf.py

```
1  #!/usr/bin/env python2.7
2  # -*- coding: utf-8 -*-
3  """
4  Script used to plot the waveforms for a given input voltage.
5  @author: Taimur Rabuske
6  """
7
8  from __future__ import division
9  import numpy
10 from sar_adc_conventional import sar_adc_conventional
11 from sar_adc_monotonic import sar_adc_monotonic
12 from sar_adc_mcs import sar_adc_mcs
13 from sar_adc_cs import sar_adc_cs
14
15 from multiprocessing import Pool, Lock
16
17 #Simulation parameters
18 samplingrate=1000.0e3
19 fin=54.6875e3
20 pts=2048
21 window="no"
22
23 #ADC characteristics
24 BITS=8
25 VREF=1.0
26
27 sar_adc_conventional=sar_adc_conventional(bits=BITS,
       ↪ Vref=VREF, Cu=1)
28 sar_adc_monotonic=sar_adc_monotonic(bits=BITS, Vref=VREF, Cu=1)
29 sar_adc_mcs=sar_adc_mcs(bits=BITS, Vref=VREF, Cu=1)
30 sar_adc_cs=sar_adc_cs(bits=BITS, Vref=VREF, Cu=1)
31
32 #mp=multiprocessing
33 mp=1
34
35 PTS_SIM=2**BITS
36
37 # ADC input voltage to simulate
38 y=0.8
39
```

```
print "\n\nSimulating ADCs..."

dac_conventional=sar_adc_conventional.convert(y,return_dac=1)
dac_monotonic=sar_adc_monotonic.convert(y,return_dac=1)
dac_mcs=sar_adc_mcs.convert(y,return_dac=1)
dac_cs=sar_adc_cs.convert(y,return_dac=1)

numpy.savetxt("dac_conventional.csv", dac_conventional,
    ↪ delimiter=",", fmt="%s")
numpy.savetxt("dac_monotonic.csv", dac_monotonic,
    ↪ delimiter=",", fmt="%s")
numpy.savetxt("dac_mcs.csv", dac_mcs, delimiter=",", fmt="%s")
numpy.savetxt("dac_cs.csv", dac_cs, delimiter=",", fmt="%s")
```

## References


1. J. McCreary, P. Gray, All-MOS charge redistribution analog-to-digital conversion techniques. I. IEEE J. Solid-State Circuits **10**(6), 371–379 (1975). doi:10.1109/JSSC.1975.1050629 (cit. on p. 28)
2. C.-C. Liu, S.-J. Chang, G.-Y. Huang, Y.-Z. Lin, A 10-bit 50-MS/s SAR ADC with a monotonic capacitor switching procedure. IEEE J. Solid-State Circuits **45**(4), 731–740 (2010). doi:10.1109/JSSC.2010. 2042254 (cit. on p. 30)
3. X. Song, Y. Xiao, Z. Zhu, VCM-based monotonic capacitor switching scheme for SAR ADC. Electron. Lett. **49**(5), 327–329, February 2013. doi:10.1049/el.2012.3332 (cit. on p. 32)
4. V. Hariprasath, J. Guerber S.-H. Lee, U.-K. Moon, Merged capacitor switching based SAR ADC with highest switching energy-efficiency. Electron. Lett. **46**(9), 620 (2010). doi:10.1049/el.2010.0706 (cit. on p. 32)
5. Y. Zhu, C.-H. Chan, U.-F. Chio, S.-W. Sin, U. Seng-Pan , R.P. Martins, F. Maloberti, A 10-bit 100-MS/s reference-free SAR ADC in 90 nm CMOS. IEEE J. Solid-State Circuits **45**(6), 1111–1121 (2010). doi:10.1109/JSSC.2010.2048498 (cit. on p. 32)
6. E. Rahimi, M. Yavari, Energy-efficient high-accuracy switching method for SAR ADCs. Electron. Lett. **50**(7), 499–501 (2014). doi:10.1049/el.2013.3451 (cit. on p. 34)
7. C. Yuan, Y. Lam, Low-energy and area-efficient tri-level switching scheme for SAR ADC. Electron. Lett. **48**(9), 482 (2012). doi:10.1049/el.2011.4001 (cit. on p. 34)
8. J. Craninckx, 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 International Solid-State Circuits Conference (ISSCC) IEEE*, February 2007, pp. 246–600. doi: 10.1109/ISSCC.2007.373386 (cit. on pp. 25, 41, 46, 47, 48, 49, and 50)
9. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70 dB DR 10b 0-to-80 MS/s current-integrating SAR ADC with adaptive dynamic range", in *IEEE Int. Solid-State Circuits Conf (ISSCC)* IEEE, February 2012, pp. 470–472. doi:1.1109/ISSCC.2012.6177095 (cit. on pp. 25, 45, 48, and 50)
10. V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Van der Plas, J. Craninckx, An 82 µw 9b 40 MS/s noise-tolerant dynamic-SAR ADC in 90 nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC) IEEE*, February 2008, pp. 238–610. doi:10.1109/ISSCC.2008.4523145 (cit. on pp. 25, 46, 48, and 50)

11. J.-H. Tsai, Y.-J. Chen, M.-H. Shen, P.-C. Huang, 1-V, 8b, 40 MS/s, 113 μW charge-recycling SAR ADC with a 14 μW asynchronous controller, in *IEEE Symposium VLSI Circuits* 2011, pp. 264–265 (cit. on pp. 25, 47, and 50)
12. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70 dB DR 10b 0-to-80 MS/s current-integrating SAR ADC with adaptive dynamic range. IEEE J. Solid-State Circuits **49**(5), 1173–1183 (2014). doi:10.1109/JSSC.2014.2309086 (cit. on pp. 25, 48, 49, and 50)

# Chapter 4
# Effects of Nonidealities on the Performance of CS-ADCs

**Contents**



## 4.1 Introduction

The performance of the CS-ADC is limited by nonidealities, which include mismatch, parasitics, comparator offset, and noise. As with any other class of circuits, it is crucial to quantify the impact brought by these sources of error at design-time. For this purpose, a detailed analysis of the nonidealities that lead to performance degradation in the CS-ADC is presented. Whenever it is reasonable, a mathematical model is derived to assist the design and, for the cases where this approach is unpractical, the results of extensive simulation are presented. This chapter begins with the derivation of the expressions for the voltage at



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_4

the inputs of the comparator during a given conversion. These expressions are used to model the effect of most of the nonidealities to the performance of CS-ADCs.

## 4.2 Trajectory of the DAC Voltage in a CS-ADC

For the derivation of the expression of the voltage at the comparator inputs, the charge conservation principle is extensively used. It is also assumed that the inputs to the comparator present an infinite resistance, thus no current flows in or out of these terminals.

Consider the diagram in Fig. 4.1, that shows the most important components of a 3-bit CS-ADC, including the TH capacitors ($C_{TH}$), DAC capacitors ($C_0$ and $C_1$), and the DAC parasitics ($C_{p1}$ and $C_{p0}$). The circuit is described in four different cycles. The first cycle encompasses the sampling of the input signal on the TH capacitances and the precharge of the DAC capacitors to $V_{PC}$. In the analysis, it is considered that sampling and precharge were completed beforehand, and the following cycles are represented by the symbol $i$. Cycle $i = 0$ describes the circuit at the state used to decide the MSB when the TH is in "hold" mode and the DAC capacitances are disconnected from the rest of the circuit. Similarly, cycles $i = 1$ and $i = 2$ model

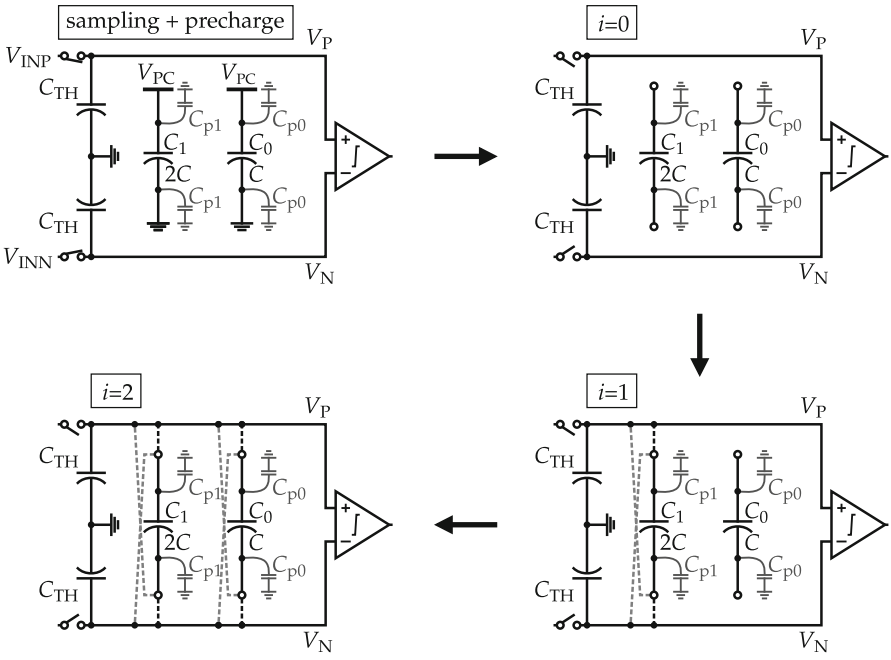



**Fig. 4.1** Circuit schematics of a 3-bit CS-ADC during the different cycles in a conversion

the circuit at the instants when the comparator is triggered to decide the second and third most significant bits, respectively. All the states assume sufficient settling. Note that from cycle 0 onwards, the DAC operation is passive, as none of the nodes are connected to any voltage source. Since there is no current path from/to the exterior at cycles 0, 1, and 2, the net charge of the system is the same as the net charge at the end of the sampling/precharge phase. Also, note that the same reasoning can be applied to ADCs with any resolution.

At cycle $i = 0$, the charges stored in the TH capacitors are dictated by the input voltage at the exact instant when the TH is switched from the "track" to the "hold" mode.

$$Q_{\mathrm{THP}}[0] = C_{\mathrm{TH}} V_{\mathrm{INP}} \tag{4.1}$$

$$Q_{\mathrm{THN}}[0] = C_{\mathrm{TH}} V_{\mathrm{INN}}. \tag{4.2}$$

Similarly, the charges on the positive and negative terminals of the DAC unit cells are given by

$$Q_{1\mathrm{P}}[0] = C_1 (V_{\mathrm{PC}} - 0) + C_{\mathrm{p1}} (V_{\mathrm{PC}} - 0) \tag{4.3}$$

$$Q_{1\mathrm{N}}[0] = C_1 (0 - V_{\mathrm{PC}}) + C_{\mathrm{p1}} (0 - 0) \tag{4.4}$$

$$Q_{0\mathrm{P}}[0] = C_0 (V_{\mathrm{PC}} - 0) + C_{\mathrm{p0}} (V_{\mathrm{PC}} - 0) \tag{4.5}$$

$$Q_{0\mathrm{N}}[0] = C_0 (0 - V_{\mathrm{PC}}) + C_{\mathrm{p0}} (0 - 0). \tag{4.6}$$

The differential voltage $V_{\mathrm{PN}} = V_{\mathrm{P}} - V_{\mathrm{N}}$ is evaluated by the comparator, which outputs the comparison result $k_2$ ("0" or "1"). The bit $k_2$ will be used at the end of conversion to generate the 3-bit ADC output $K = \{k_2, k_1, k_0\}$. Then, the algorithm moves to the next cycle.

At cycle $i = 1$ the capacitor $C_1$ is connected in parallel or anti-parallel to the TH according to the comparison result $k_2$. Therefore, the total charges on the nodes $V_{\mathrm{P}}$ and $V_{\mathrm{N}}$, connected to the comparator inputs, become

$$Q_{\mathrm{P}}[1] = \left(C_{\mathrm{TH}} + C_{\mathrm{p1}}\right) V_{\mathrm{P}}[1] + C_1 (V_{\mathrm{P}}[1] - V_{\mathrm{N}}[1]) \tag{4.7}$$

$$Q_{\mathrm{N}}[1] = \left(C_{\mathrm{TH}} + C_{\mathrm{p1}}\right) V_{\mathrm{N}}[1] + C_1 (V_{\mathrm{N}}[1] - V_{\mathrm{P}}[1]). \tag{4.8}$$

As the charge is conserved, the following also holds.

$$Q_{\mathrm{P}}[1] = \begin{cases} Q_{\mathrm{THP}}[0] + Q_{1\mathrm{P}}[0] & \text{if } k_2 = 0 \\ Q_{\mathrm{THP}}[0] + Q_{1\mathrm{N}}[0] & \text{if } k_2 = 1 \end{cases} \tag{4.9}$$

$$Q_{\mathrm{N}}[1] = \begin{cases} Q_{\mathrm{THN}}[0] + Q_{1\mathrm{N}}[0] & \text{if } k_2 = 0 \\ Q_{\mathrm{THN}}[0] + Q_{1\mathrm{P}}[0] & \text{if } k_2 = 1. \end{cases} \tag{4.10}$$

Combining (4.7) with (4.9) and (4.8) with (4.10) enables solving for $V_{\mathrm{P}}$ and $V_{\mathrm{N}}$, respectively:

$$V_{\mathrm{P}}[1] = \begin{cases} \dfrac{C_1\left(C_1 V_{\mathrm{PC}} - C_{\mathrm{TH}} V_{\mathrm{INN}}\right) - \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)\left(C_1 V_{\mathrm{PC}} + C_{\mathrm{TH}} V_{\mathrm{INP}} + C_{\mathrm{p1}} V_{\mathrm{PC}}\right)}{C_1^2 - \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)^2} & \text{if } k_2 = 0 \\ \dfrac{-C_1\left(C_1 V_{\mathrm{PC}} + C_{\mathrm{TH}} V_{\mathrm{INN}} + C_{\mathrm{p1}} V_{\mathrm{PC}}\right) + \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)\left(C_1 V_{\mathrm{PC}} - C_{\mathrm{TH}} V_{\mathrm{INP}}\right)}{C_1^2 - \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)^2} & \text{if } k_2 = 1 \end{cases} \tag{4.11}$$

$$V_{\mathrm{N}}[1] = \begin{cases} \dfrac{-C_1\left(C_1 V_{\mathrm{PC}} + C_{\mathrm{TH}} V_{\mathrm{INP}} + C_{\mathrm{p1}} V_{\mathrm{PC}}\right) + \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)\left(C_1 V_{\mathrm{PC}} - C_{\mathrm{TH}} V_{\mathrm{INN}}\right)}{C_1^2 - \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)^2} & \text{if } k_2 = 0 \\ \dfrac{C_1\left(C_1 V_{\mathrm{PC}} - C_{\mathrm{TH}} V_{\mathrm{INP}}\right) - \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)\left(C_1 V_{\mathrm{PC}} + C_{\mathrm{TH}} V_{\mathrm{INN}} + C_{\mathrm{p1}} V_{\mathrm{PC}}\right)}{C_1^2 - \left(C_1 + C_{\mathrm{TH}} + C_{\mathrm{p1}}\right)^2} & \text{if } k_2 = 1. \end{cases} \tag{4.12}$$

The voltage $V_{\mathrm{PN}}[1] = V_{\mathrm{P}}[1] - V_{\mathrm{N}}[1]$ is now evaluated by the comparator, which outputs the comparison result $k_1$. Similarly, $Q_{\mathrm{P}}$ and $Q_{\mathrm{N}}$ can be found at $i = 2$, according to the results of the previous comparisons $k_2$ and $k_1$,

$$Q_{\mathrm{P}}[2] = \left(C_{\mathrm{TH}} + C_{\mathrm{p1}} + C_{\mathrm{p0}}\right) V_{\mathrm{P}}[2] + \left(C_1 + C_0\right)\left(V_{\mathrm{P}}[2] - V_{\mathrm{N}}[2]\right) \tag{4.13}$$

$$Q_{\mathrm{N}}[2] = \left(C_{\mathrm{TH}} + C_{\mathrm{p1}} + C_{\mathrm{p0}}\right) V_{\mathrm{N}}[2] + \left(C_1 + C_0\right)\left(V_{\mathrm{N}}[2] - V_{\mathrm{P}}[2]\right) \tag{4.14}$$

$$Q_{\mathrm{P}}[2] = \begin{cases} Q_{\mathrm{THP}}[0] + Q_{1\mathrm{P}}[0] + Q_{0\mathrm{P}}[0] & \text{if } \{k_2, k_1\}=\{0,0\} \\ Q_{\mathrm{THP}}[0] + Q_{1\mathrm{P}}[0] + Q_{0\mathrm{N}}[0] & \text{if } \{k_2, k_1\}=\{0,1\} \\ Q_{\mathrm{THP}}[0] + Q_{1\mathrm{N}}[0] + Q_{0\mathrm{P}}[0] & \text{if } \{k_2, k_1\}=\{1,0\} \\ Q_{\mathrm{THP}}[0] + Q_{1\mathrm{N}}[0] + Q_{0\mathrm{N}}[0] & \text{if } \{k_2, k_1\}=\{1,1\} \end{cases} \tag{4.15}$$

$$Q_{\mathrm{N}}[2] = \begin{cases} Q_{\mathrm{THN}}[0] + Q_{1\mathrm{N}}[0] + Q_{0\mathrm{N}}[0] & \text{if } \{k_2, k_1\}=\{0,0\} \\ Q_{\mathrm{THN}}[0] + Q_{1\mathrm{N}}[0] + Q_{0\mathrm{P}}[0] & \text{if } \{k_2, k_1\}=\{0,1\} \\ Q_{\mathrm{THN}}[0] + Q_{1\mathrm{P}}[0] + Q_{0\mathrm{N}}[0] & \text{if } \{k_2, k_1\}=\{1,0\} \\ Q_{\mathrm{THN}}[0] + Q_{1\mathrm{P}}[0] + Q_{0\mathrm{P}}[0] & \text{if } \{k_2, k_1\}=\{1,1\}. \end{cases} \tag{4.16}$$

Again, combining (4.13) with (4.15) and (4.14) with (4.16), one can find $V_{\mathrm{P}}$ and $V_{\mathrm{N}}$, respectively, at $i = 2$. In fact, the described procedure may be employed to find $V_{\mathrm{P}}$ and $V_{\mathrm{N}}$ at any cycle $i$ of the SAR algorithm for a CS-ADC with any resolution $B$. If simplification is performed on the resulting equations, one arrives at closed-form expressions for the differential voltage $V_{\mathrm{PN}}$.

$$V_{\mathrm{PN}}[i] = V_{\mathrm{P}}[i] - V_{\mathrm{N}}[i] = \frac{\frac{C_{\mathrm{TH}}}{2}\left(V_{\mathrm{INP}} - V_{\mathrm{INN}}\right) - \sum_{j=1}^{i}\left(2k_{B-j} - 1\right)\left(C_{B-j-1} + \frac{C_{\mathrm{p},B-j-1}}{2}\right) V_{\mathrm{PC}}}{\frac{C_{\mathrm{TH}}}{2} + \sum_{j=1}^{i}\left(C_{B-j-1} + \frac{C_{\mathrm{p},B-j-1}}{2}\right)}. \tag{4.17}$$

The expression may be elaborated more intuitively as follows. The numerator represents the amount of charge in the system. The left-hand term of the numerator is the charge initially stored in the TH. The value of $C_{\mathrm{TH}}$ is divided by two because the TH capacitances appear in series for differential signals. The summation on the right side of the numerator represents the charges from the capacitors in the array that are connected in each cycle. The factor $(2k_{B-j}-1)$ converts the comparator results from the conventional binary form $\{0, 1\}$ to $\{-1, +1\}$. The denominator of the expression represents the total capacitance that is connected to the comparator inputs in a given cycle, and again $C_{\mathrm{TH}}$ appears dividing by 2. An identical effect is noticed in the parasitic capacitances, which also appear in series when the differential voltage is evaluated.

The equation is valid for any cycle $i$ including 0, that corresponds to the MSB decision cycle. In the particular case when $i = 0$, the two summations have upper limit equal to 0, which is lower than the lower limit (that is 1). By definition, this is an empty sum, or a sum with no summands, and equals zero. Therefore, for $i = 0$, all the summands disappear remaining (4.18).

$$V_{\mathrm{PN}}[0] = \frac{C_{\mathrm{TH}}\left(V_{\mathrm{INP}} - V_{\mathrm{INN}}\right) - (0)V_{\mathrm{PN}}}{C_{\mathrm{TH}} + 0} = V_{\mathrm{INP}} - V_{\mathrm{INN}}. \tag{4.18}$$

The common-mode voltage at the comparator inputs, at any cycle $i$, is defined by (4.19).

$$V_{\mathrm{CM}}[i] = \frac{V_{\mathrm{P}}[i] + V_{\mathrm{N}}[i]}{2}. \tag{4.19}$$

Solving and simplifying (4.19) yields in

$$V_{\mathrm{CM}}[i] = \frac{V_{\mathrm{INP}} + V_{\mathrm{INN}}}{2} = V_{\mathrm{IN,CM}}. \tag{4.20}$$

One sees in (4.20) that $V_{\mathrm{CM}}$ remains constant for all the cycles, and is equal to the common-mode voltage of the input signal. This observation is valid only when the circuit is balanced, e.g. has identical parasitics and TH capacitances on positive and negative sides of the differential circuit.

## 4.3 Analog-to-Digital Conversion Gain

By omitting parasitics on the model, the differential voltage seen by the comparator is given by (4.21).

$$V_{\mathrm{PN}}[i] = \frac{\frac{C_{\mathrm{TH}}}{2}\left(V_{\mathrm{INP}} - V_{\mathrm{INN}}\right) - \sum_{j=1}^{i}(2k_{B-j} - 1)C_{B-j-1}V_{\mathrm{PC}}}{\frac{C_{\mathrm{TH}}}{2} + \sum_{j=1}^{i} C_{B-j-1}}. \tag{4.21}$$

The remainder of this section considers that the DAC capacitances are matched and obey a binary relationship. If the differential input voltage ($V_{\mathrm{INP}} - V_{\mathrm{INN}}$) of the ADC is equal to the maximum voltage right before the ADC saturates, all the comparison results must be "1"s at the end of the conversion and the residual charge corresponds to 1 LSB, or $Q_0$. Alternatively, one may state that at the end of conversion, the difference between the TH charge and the DAC charge is $Q_0$. If larger, the ADC output is saturated and the quantization error is larger than $\frac{1}{2}$ LSB. If smaller, the input is not at the maximum allowed by the ADC. Thus,

$$Q_0 = Q_{\mathrm{TH,max}} - Q_{\mathrm{DAC,111..1}}. \tag{4.22}$$

Because of the differential and symmetric operation of the ADC, the output is said to be saturated if the input is larger than $V_{\mathrm{IR}}/2$ or smaller than $-V_{\mathrm{IR}}/2$, where $V_{\mathrm{IR}}$ is the differential input range voltage of the ADC. Then, we rewrite (4.22) as (4.23).

$$C_0 V_{\mathrm{PC}} = \frac{C_{\mathrm{TH}}}{2}\frac{V_{\mathrm{IR}}}{2} - \left(2^{B-1} - 1\right) C_0 V_{\mathrm{PC}}. \tag{4.23}$$

Solving for $V_{\mathrm{IR}}$ leads to

$$V_{\mathrm{IR}} = \frac{2^{B+1} C_0}{C_{\mathrm{TH}}} V_{\mathrm{PC}}. \tag{4.24}$$

The ADC gain $G_{\mathrm{ADC}}$ may be defined as the ratio between the input range $V_{\mathrm{IR}}$ and the precharge voltage $V_{\mathrm{PC}}$. Thus,

$$G_{\mathrm{ADC}} = \frac{V_{\mathrm{IR}}}{V_{\mathrm{PC}}} = \frac{2^{B+1} C_0}{C_{\mathrm{TH}}}. \tag{4.25}$$

In most of the reported designs, $V_{\mathrm{PC}}$ equals $V_{\mathrm{DD}}$. Also, one is generally interested in maximizing the input voltage range, in order to increase the SNR of the ADC. Therefore, if the input common-mode voltage $V_{\mathrm{IN,CM}}$ is set to $\frac{V_{\mathrm{DD}}}{2}$, and both the input signals are bounded to $V_{\mathrm{DD}}$ and 0, $V_{\mathrm{IR}}$ is equal to $2V_{\mathrm{DD}}$, because of the differential operation. In this particular case, (4.25) becomes (4.26).

$$G_{\mathrm{ADC}} = 2 = \frac{2^{B+1} C_0}{C_{\mathrm{TH}}}. \tag{4.26}$$

Finally,

$$C_{\mathrm{TH}} = 2^B C_0. \tag{4.27}$$

Therefore, for $V_{\mathrm{IR}} = 2V_{\mathrm{DD}}$ in the 3-bit example, $C_{\mathrm{TH}}$ becomes $8C_0$. It can be easily shown that, for a constant input range, the TH capacitances scale linearly with the precharge voltage. Thus, if $V_{\mathrm{PC}}$ is set to $V_{\mathrm{DD}}/2$, $C_{\mathrm{TH}}$ becomes $4C_0$; if $V_{\mathrm{PC}}$ is set to $V_{\mathrm{DD}}/4$, $C_{\mathrm{TH}}$ becomes $2C_0$, and so forth.

## 4.4 Errors Caused by Mismatch

In this section, the effect of the mismatch between the capacitors inside the ADC on the accuracy of the conversion is quantified. Three cases are considered in the analysis: the mismatch between the two capacitors in the differential TH, the mismatch between these TH capacitors and the total DAC capacitance, and the mismatch between the capacitors inside the DAC.

### *4.4.1 Mismatch Between TH Capacitors*

To analyze the impact of the mismatch between $C_{\mathrm{THP}}$ and $C_{\mathrm{THN}}$, it is assumed that a capacitance mismatch $\Delta_{\mathrm{C}}$ appears in the TH capacitances as follows.

$$C_{\mathrm{THP}} = C_{\mathrm{TH}} + \frac{\Delta_{\mathrm{C}}}{2} \tag{4.28}$$

$$C_{\mathrm{THN}} = C_{\mathrm{TH}} - \frac{\Delta_{\mathrm{C}}}{2}. \tag{4.29}$$

Splitting the mismatch between the two capacitors greatly simplifies the calculations. Now, $C_{\mathrm{TH}}$ represents the average TH capacitance. Then, the expression for $V_{\mathrm{PN}}$ at the cycle $i$ in presence of TH capacitors mismatch is derived and given in (4.30).

$$V_{\mathrm{PN}}[i] = \frac{\left(1 - \left(\frac{\Delta_{\mathrm{C}}}{2C_{\mathrm{TH}}}\right)^2\right) \frac{C_{\mathrm{TH}}}{2} (V_{\mathrm{INP}} - V_{\mathrm{INN}}) - V_{\mathrm{PC}} \sum_{j=1}^{i} (2k_{B-j} - 1) C_{B-j-1}}{\left(1 - \left(\frac{\Delta_{\mathrm{C}}}{2C_{\mathrm{TH}}}\right)^2\right) \frac{C_{\mathrm{TH}}}{2} + \sum_{j=1}^{i} C_{B-j-1}}. \tag{4.30}$$

The factors that are introduced by the mismatch are highlighted. Looking to the expression, it is noticeable that the TH mismatch does not introduce nonlinearity on the ADC transfer curve. However, since the factors only appear for the terms that correspond to the TH charge capacitance, a gain error is expected in the ADC. The phenomenon of gain error in ADCs is depicted in Fig. 4.2 and may be quantified using (4.31), where $V_{111..1}$ and $V_{000..0}$ are the thresholds of the greatest and lowest output codes, respectively.

$$\Delta_{\mathrm{G}}\,(\%) = 100 \left( \frac{V_{111..1} - V_{000..0}}{V_{\mathrm{IR}} - 2V_{\mathrm{LSB}}} - 1 \right). \tag{4.31}$$

A positive gain error $\Delta_{\mathrm{G}}$ means that the ADC saturates before the full input range is driven, and a negative $\Delta_{\mathrm{G}}$ means that a signal larger than the input range is required to activate the lowest and the greatest possible output codes. It is possible

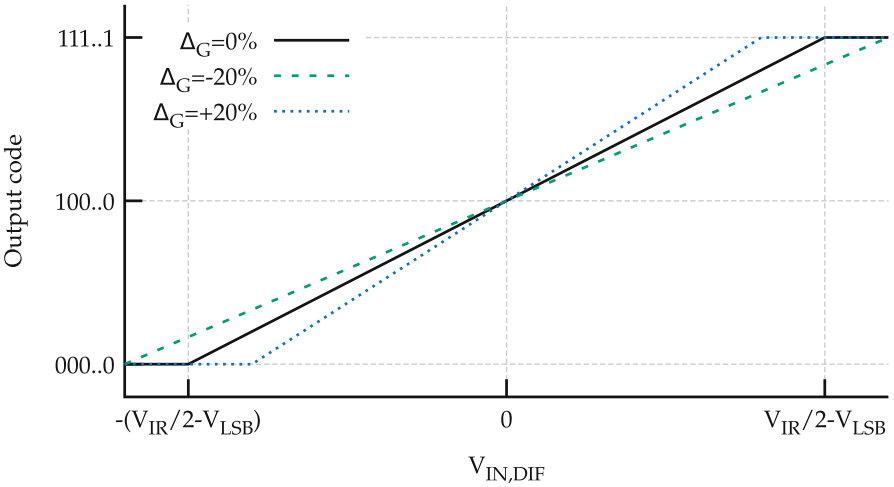



**Fig. 4.2** Illustration of the gain error in ADCs: *x*-axis is the input voltage and *y*-axis is the output code of the ADC

to follow the same procedure employed for (4.25) to find the gain in the presence of mismatched TH capacitors:

$$\frac{V_{\mathrm{IR}}}{V_{\mathrm{PC}}} = \frac{2^{B+1}C_0}{C_{\mathrm{TH}}}\left(\frac{1}{1-\left(\frac{\Delta_{\mathrm{C}}}{2C_{\mathrm{TH}}}\right)^2}\right). \tag{4.32}$$

By dividing the resulting expression by the ideal gain of a CS-ADC, given in (4.26), we obtain the gain error in presence of mismatch on the TH capacitors.

$$\Delta_{\mathrm{G}} = \frac{1}{1-\left(\frac{\Delta_{\mathrm{C}}}{2C_{\mathrm{TH}}}\right)^2}. \tag{4.33}$$

Finally, rewriting (4.32) as a percentage leads to (4.34).

$$\Delta_{\mathrm{G}}(\%) = 100\left(\frac{1}{1-\left(\frac{\Delta_{\mathrm{C}}}{2C_{\mathrm{TH}}}\right)^2} - 1\right). \tag{4.34}$$

To validate the expression, behavioral simulations for an 8-bit ADC with $V_{\mathrm{IR}} = 2V_{\mathrm{DD}}$ and $C_{\mathrm{TH}} = 256\,\mathrm{fF}$ are run for different values of $\Delta_{\mathrm{C}}$. The results obtained from simulations are contrasted with the gain error achieved with (4.34). The comparison is shown in Table 4.1, that indicates great agreement between simulation and analytical calculations. The shape of $\Delta_{\mathrm{G}}$ as function of $\Delta_{\mathrm{C}}$ for a range of $-50$ to 50 fF, corresponding to approximately 20 % of $C_{\mathrm{TH}}$, is shown in Fig. 4.3. Note that

**Table 4.1** Comparison of calculated and simulated gain error for different values of mismatch between TH capacitors, for an 8-bit SAR ADC with $V_{PC} = 1\,V$, $V_{IR} = 2\,V$, $C_{TH} = 256\,fF$, and $C_0 = 1\,fF$

| $\Delta_C$ (fF) | Calculated $\Delta_G$ (%) | Simulated $\Delta_G$ (%) |
|---|---|---|
| 10 | 0.0381 | 0.0379 |
| 20 | 0.1528 | 0.1532 |
| 50 | 0.9628 | 0.9625 |
| 100 | 3.9659 | 3.9660 |
| 150 | 9.3889 | 9.3889 |

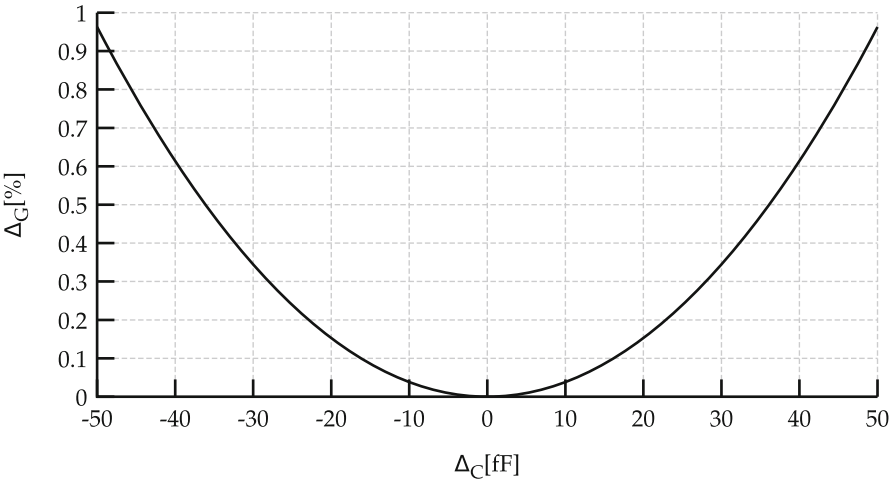



**Fig. 4.3** Analytical gain error as function of the mismatch between TH capacitors, for an 8-bit SAR ADC with $V_{PC} = 1\,V$, $V_{IR} = 2\,V$, $C_{TH} = 256\,fF$, and $C_0 = 1\,fF$

even for a large mismatch of almost 20 % of $C_{TH}$, the gain error is smaller than 1 %. Only if the mismatch is as large as 60 %, or 150 fF, that the gain error approaches 10 %. If we keep in mind that the capacitors in modern CMOS processes provide a very good matching, with a mismatch generally lower than a few percent, the values on Table 4.1 are very unlikely to occur in practical implementations. It is fair to assume that, for real SAR ADC implementations, the gain error introduced by the mismatch between TH capacitors is negligible.

Besides the disturbances on the differential voltage caused by the mismatch in the TH capacitors, the ADC becomes asymmetric and changes in the common-mode voltage at the comparator inputs are expected. The expression in (4.35) can be derived to compute the common-mode voltage at the cycle $i$, where $V_{IN,CM}$ is the common-mode voltage of the input signal.

$$V_{CM}[i] = V_{IN,CM} + \frac{\Delta_C \left( (V_{INP} - V_{INN}) \sum_{j=1}^{i} C_{B-j-1} + V_{PC} \sum_{j=1}^{i} (2k_{B-j} - 1) C_{B-j-1} \right)}{4C_{TH} \sum_{j=1}^{i} C_{B-j-1} + 2{C_{TH}}^2 - \frac{{\Delta_C}^2}{2}}. \tag{4.35}$$

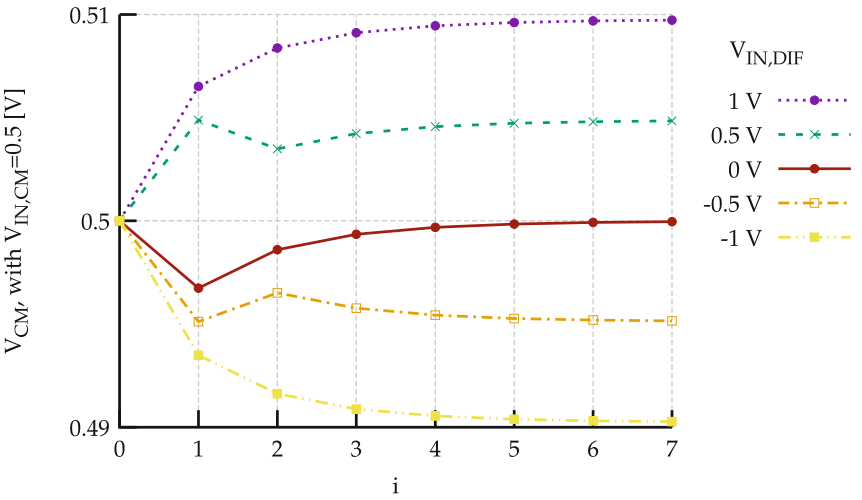



**Fig. 4.4** Simulated trajectory of the common-mode voltage on the comparator inputs for different input voltages on an 8-bit ADC with $V_{PC} = 1\,V$, $V_{IR} = 2\,V$, $C_{TH} = 256\,fF$, $C_0 = 1\,fF$ and $\Delta_C = 10\,fF$

Note that the terms that multiply by $(V_{INP} - V_{INN})$ and $V_{PC}$ in the numerator are both positive, differently than what happens for the differential signal, where they have different signs. Therefore, the maximum and minimum of this function occur for the extremes of the differential input voltage. Figure 4.4 depicts the behavior of $V_{CM}$ for different input voltages for an 8-bit CS-ADC with 2 V of input range (−1 V to +1 V), a TH capacitance of 256 fF and $\Delta_C$ equal to 10 fF. The trajectories in the figure corroborate with the idea that the worst deviation on $V_{CM}$ occur for maximum and minimum input voltages. In the case of maximum positive differential input $V_{IR}/2$, all the comparison results are logic ones, and the second summation of the numerator can be rewritten as $\sum C_{B-j-1}$, instead of $\sum(2k_{B-j} - 1)C_{B-j-1}$. For the minimum differential input of $-V_{IR}/2$, all the comparison results are logic zeros, and the second summation becomes $-\sum C_{B-j-1}$. Therefore, the minimum and maximum values of the common-mode voltage can be found with

$$V_{CM}|_{min,max} = V_{IN,CM} \pm \frac{\Delta_C \left(\frac{V_{IR}}{2} + V_{PC}\right) \sum_{j=1}^{i} C_{B-j-1}}{4C_{TH} \sum_{j=1}^{i} C_{B-j-1} + 2{C_{TH}}^2 - \frac{{\Delta_C}^2}{2}}. \tag{4.36}$$

We can substitute the sum for the value of the capacitance at the end of conversion. Thus,

$$V_{CM}|_{min,max} = V_{IN,CM} \pm \frac{\Delta_C \left(\frac{V_{IR}}{2} + V_{PC}\right) \left(2^{(B-1)} - 1\right) C_0}{4C_{TH} \left(2^{B-1} - 1\right) C_0 + 2{C_{TH}}^2 - \frac{{\Delta_C}^2}{2}}. \tag{4.37}$$

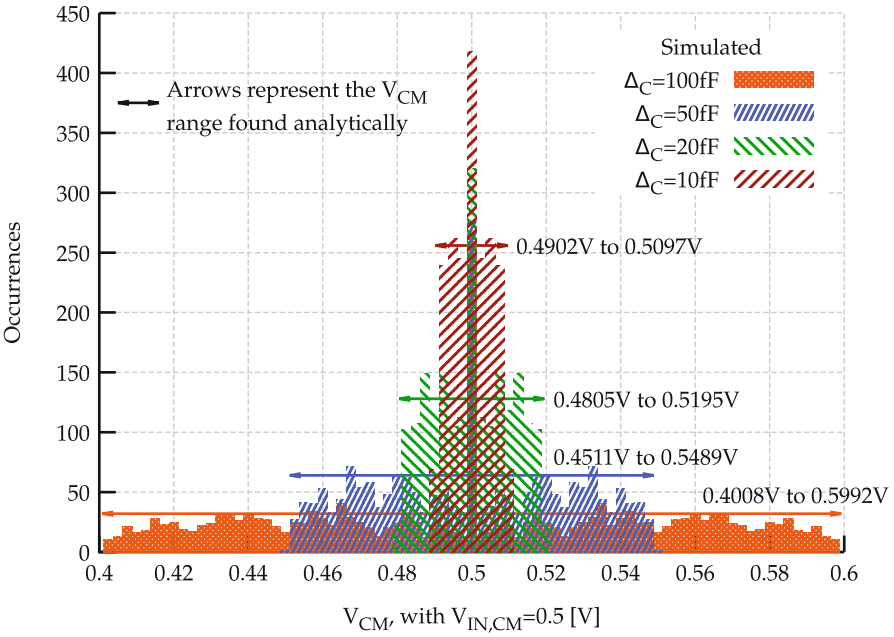



**Fig. 4.5** Histogram showing the simulated spread of the common-mode voltage in the presence of TH capacitor mismatch, for several values of $\Delta_C$, for an 8-bit ADC with $V_{PC} = 1\,V$, $V_{IR} = 2\,V$, $C_{TH} = 256\,fF$, and $C_0 = 1\,fF$

To validate the common-mode voltage model, the same 8-bit ADC is simulated with a linear ramp as the input signal, for different values of $\Delta_C$. A total of 256 points are simulated, ensuring that every output code is activated once. Figure 4.5 shows a histogram with the values of $V_{CM}$ during the test, for all the 8 conversion cycles of the 256 conversions. The plot also includes the maximum and minimum values of $V_{CM}$ found using (4.37). It is possible to verify that the values of common-mode voltage are indeed bounded by the values computed analytically, evidencing the correctness of (4.37). Again, the analysis was extended to exaggerated values of $\Delta_C$, which are not likely to be seen in practical implementations. For reasonable values of $\Delta_C$, the disturbance of the common-mode caused by mismatched TH capacitors is in the order of tens of mV, and this range of values is not expected to cause significant effects on the performance of the comparator. Finally, the range of common-mode voltages that may occur in an 8-bit ADC with $C_{TH} = 256\,fF$, as function of $\Delta_C$, is shown in Fig. 4.6.

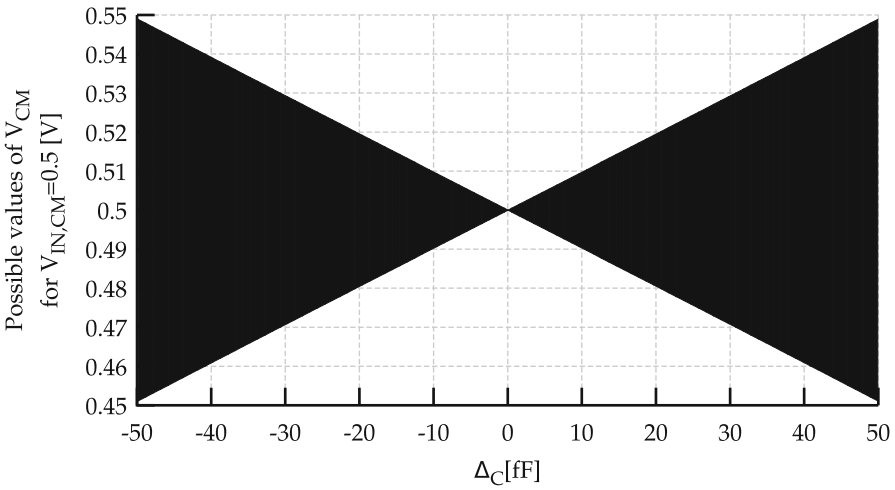



**Fig. 4.6** Plot showing the calculated range of possible values of common-mode voltage during a conversion as function of the mismatch between TH capacitors, for an 8-bit ADC with $V_{PC} = 1$ V, $V_{IR} = 2$ V, $C_{TH} = 256$ fF, and $C_0 = 1$ fF

### *4.4.2 Mismatch Between TH and Total DAC Capacitance*

For this section, we once again assume that the DAC capacitors are matched and obey a binary relationship. This simplification is done to allow focusing on the impact of the mismatch between the values of the TH and the total DAC capacitance. Additionally, for the sake of simplicity, it is considered that the DAC capacitors are fixed, and that the value of the TH capacitance is $C_{TH} + \Delta_C$. The expression (4.21) may be rewritten as

$$V_{PN}[i] = \frac{\frac{C_{TH}+\Delta_C}{2}(V_{INP} - V_{INN}) - \sum_{j=1}^{i}(2k_{B-j} - 1)C_{B-j-1}V_{PC}}{\frac{C_{TH}+\Delta_C}{2} + \sum_{j=1}^{i} C_{B-j-1}}. \tag{4.38}$$

The expression reveals that the conversion remains linear. On the other hand, as the mismatch is added up to the factor that multiplies the differential input signal, a deviation in the gain of the converter is expected. In fact, the gain can be computed by (4.39).

$$G_{ADC} = \frac{V_{IR}}{V_{PC}} = \frac{2^{B+1}C_0}{C_{TH} + \Delta_C}. \tag{4.39}$$

This expression may be rewritten as

$$G_{\mathrm{ADC}} = \frac{2^{B+1}C_0}{C_{\mathrm{TH}}}\left(\frac{1}{1+\frac{\Delta_{\mathrm{C}}}{C_{\mathrm{TH}}}}\right). \tag{4.40}$$

As the left term is the gain of an ideal ADC, the gain error in presence of mismatch between the total DAC capacitance and the TH capacitors becomes (4.41).

$$\Delta_{\mathrm{G}}(\%) = 100\left(\frac{1}{1+\frac{\Delta_{\mathrm{C}}}{C_{\mathrm{TH}}}} - 1\right). \tag{4.41}$$

Simplification leads to (4.42).

$$\Delta_{\mathrm{G}}(\%) = -100\left(\frac{\Delta_{\mathrm{C}}}{C_{\mathrm{TH}}+\Delta_{\mathrm{C}}}\right). \tag{4.42}$$

Behavioral simulations are run for an 8-bit ADC with different values of $\Delta_{\mathrm{C}}$, to validate the expression. The results are contrasted with the gain error obtained with (4.42). These results are shown in Table 4.2. Also, Fig. 4.7 shows the impact of $\Delta_{\mathrm{C}}$ on the gain error for an 8-bit ADC with 256 fF TH capacitance. The results show

**Table 4.2** Comparison of calculated and simulated gain error for different values of mismatch between TH and total DAC capacitance, for an 8-bit SAR ADC with $V_{\mathrm{PC}} = 1\,\mathrm{V}$, $V_{\mathrm{IR}} = 2\,\mathrm{V}$, $C_{\mathrm{TH}} = 256\,\mathrm{fF}$, and $C_0 = 1\,\mathrm{fF}$

| $\Delta_{\mathrm{C}}$ (fF) | Calculated $\Delta_{\mathrm{G}}$ (%) | Simulated $\Delta_{\mathrm{G}}$ (%) |
|---|---|---|
| −20 | 8.4746 | 8.4745 |
| −10 | 4.0650 | 4.0647 |
| 10 | −3.7594 | −3.7591 |
| 20 | −7.2464 | −7.2462 |
| 50 | −16.3398 | −16.3398 |

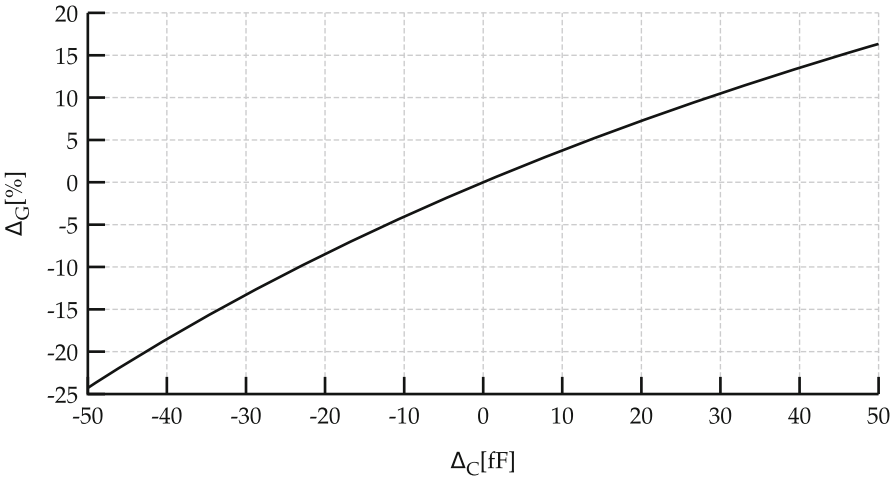



**Fig. 4.7** Analytical gain error as function of the mismatch between TH and DAC capacitors for an 8-bit SAR ADC with $V_{\mathrm{PC}} = 1\,\mathrm{V}$, $V_{\mathrm{IR}} = 2\,\mathrm{V}$, $C_{\mathrm{TH}} = 256\,\mathrm{fF}$, and $C_0 = 1\,\mathrm{fF}$

that the analytical model agrees with the simulation results. Since the mismatch for integrated capacitors in the range of hundreds of femtofarads tends to be very small (smaller than 1 %), the expected gain error in a practical implementation would hardly exceed 1 %.

### *4.4.3 Mismatch Between DAC Capacitors*

A statistical analysis of the expected value and yield of the ENOB in the presence of capacitor mismatch is nontrivial and out of the scope of this book. The reader is referred to the work recently presented in [1] for a complete and in-depth treatment of the phenomenon.

Extensive Monte-Carlo behavioral simulations are run for a range of resolutions and different mismatch spreads, to illustrate and quantify the impact of the DAC capacitors mismatch. The simulated ENOB values are obtained using a 2048-point FFT with a sample of 10,000 randomly mismatched SAR ADCs generated at each resolution and each standard deviation of mismatch. Both topologies are simulated while using the same value of total DAC capacitance, to compare the impact of mismatches in the CS-ADC to a differential CR-ADC. Therefore, since the number of required unit capacitors is four times smaller in the CS than in a differential CR, the unit capacitor of the CS-DAC is made four times larger, so that the total DAC capacitance and area are identical. The obtained results of expected value and standard deviation of the ENOB are shown in Fig. 4.8. One sees almost identical behaviors of the effective resolution for the two SAR ADC topologies, as long as the total DAC capacitances are the same.

## 4.5 Errors Caused by Parasitics

In this section, the impact of the parasitics on the accuracy of the analog-to-digital conversion is quantified. It is noticeable from (4.17) that if the DAC capacitors have balanced parasitics on the two terminals, no linearity errors arise. Therefore, the remaining of this section deals only with unbalanced parasitics.

In order to evaluate the impact of unbalanced parasitics on the DAC terminals, the circuit in Fig. 4.9 is considered. The parasitics in the top and bottom plate of the DAC capacitors are modeled by the combination of $C_{\mathrm{p}}$ and $\Delta_{\mathrm{Cp}}$, that represent the average capacitance and the difference between the value of parasitic capacitances, respectively. An analytical derivation of the behavior leads to a very complicated model, with more than twenty summations ($\Sigma$) terms. Therefore, the effects of unbalanced parasitics were analyzed through numerical simulations. As the ADC gain changes with the parasitics, the following function (found empirically based on simulations, because the impact on ADC gain and INL is not known in advance) is used to choose the size of the TH capacitance.

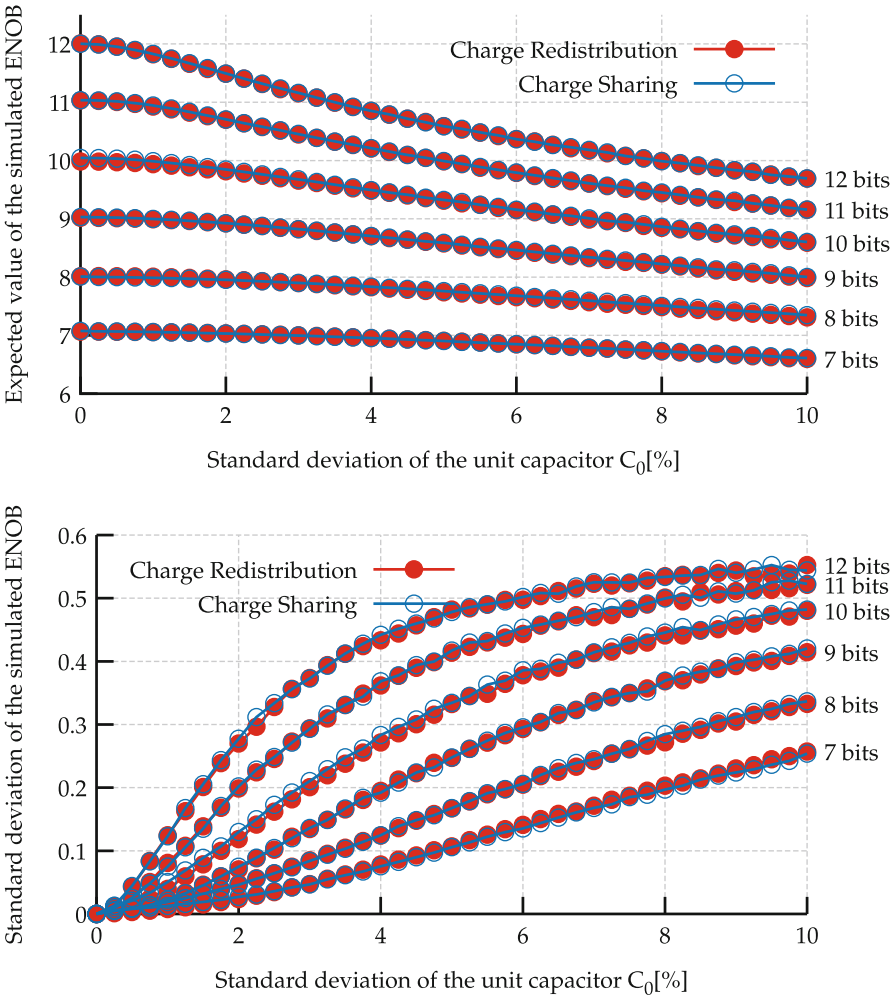



**Fig. 4.8** Mean values and standard deviation of the ENOB as function of the standard deviation of the unit capacitor mismatch for CR and CS-ADCs, extracted with a 2048-point FFT and a sample of 10,000 randomly mismatched SAR ADCs for each resolution and value of mismatch

$$C_{\mathrm{TH}} = 2^{B}(C_0 + 0.5C_{\mathrm{p}} - 0.11C_{\mathrm{p}}\Delta_{\mathrm{Cp}}). \tag{4.43}$$

With a differential input signal of 1.8 $V_{P-P}$, dimensioning $C_{\mathrm{TH}}$ according to (4.43) guarantees that the ADC does not saturate and at least 85 % of the output codes are driven for all the test cases. In other words, the range of codes driven by the input is relatively large and very similar to all the test conditions, to avoid skipping a point of critical INL. Moreover, the ENOB is fitted to the full input range by using (4.44), where $K_{\mathrm{max}}$ and $K_{\mathrm{min}}$ represent the maximum and minimum codes driven by the

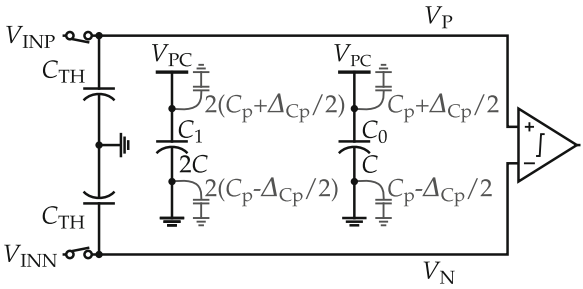



**Fig. 4.9** Circuit used to quantify the impact of unbalanced parasitics on the CS-ADC performance

input signal.

$$\text{ENOB} = \frac{\text{SNDR} - 1.76}{6.02}\left(\frac{B}{\log_2(K_{\max} - K_{\min})}\right). \tag{4.44}$$

Behavioral models were used for simulating the ADC, while using a sinusoidal input and an 8192-point FFT with coherent sampling. For each resolution, from 7 to 12 bits, a total of 2500 different combinations of $\Delta_C$ and $C_p$ ($50 \times 50$) are tested. To increase the readability of the results in the plots, the variables $C_p$ and $\Delta_{Cp}$ are normalized to $C_0$ and $C_p$, respectively. The ratio $\frac{C_p}{C_0}$ shows how large are the parasitics when compared to the DAC capacitances, so that a factor of 1 indicates that the average parasitic has the same size as the DAC capacitors. Additionally, the ratio $\frac{\Delta_{Cp}}{C_p}$ can be understood as the unbalancing factor, with 0 indicating no unbalance between the parasitics on the plates of the DAC capacitors, and 2 indicating total unbalance (top plate parasitic with $2\Delta_C$ and bottom plate with no parasitics). The analysis was extended to the exaggerated value of 5 for $\frac{C_p}{C_0}$, meaning that the average parasitics for a given DAC cell are five times larger than the differential DAC capacitor. This range of values for $\frac{C_p}{C_0}$ and full unbalance in the DAC parasitics ($\frac{\Delta_{Cp}}{C_p} = 2$) are very unlikely to occur in a properly designed DAC. On the other hand, including these ranges in the analysis helps to gain a better understanding of the sensitivity of the technology to these parameters.

The simulated ENOBs of the ADC as function of $\frac{\Delta_{Cp}}{C_p}$ and $\frac{C_p}{C_0}$ are shown in Fig. 4.10. Up to 10 bits of resolution and with $\frac{\Delta_{Cp}}{C_p}$ under 0.5, the drop in ENOB is lower than 0.2 bits for any of the simulated values of $C_p$. Still, with $\frac{\Delta_{Cp}}{C_p} = 0.5$, the ENOB is larger than 10.7 for an 11-bit ADC and larger than 11.4 for a 12-bit ADC. Also, for any simulated resolution, the topology can maintain the drop in ENOB under 0.2 bits for all values of $\frac{\Delta_{Cp}}{C_p}$, as long as the average parasitic does not grow larger than 20 % of the DAC cell capacitance ($\frac{C_p}{C_0} < 0.2$).

It is interesting to see the shapes of DNL and INL in the presence of unbalanced parasitics. Figure 4.11 shows the static characteristics of a 9-bit CS-ADC that has $C_p = \Delta_C = C_0$. Note that DNL and INL are smaller in the second and third quarters

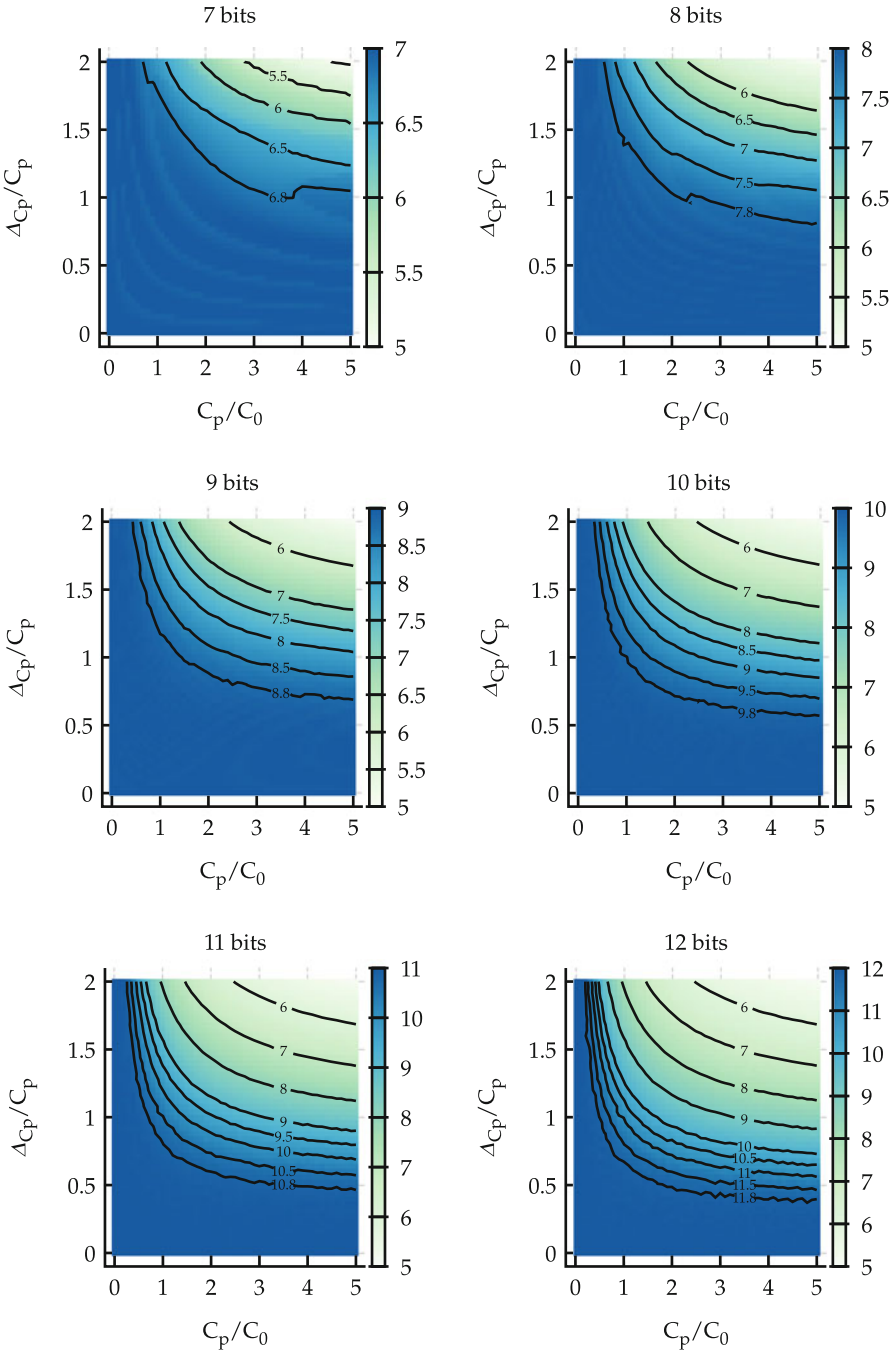



**Fig. 4.10** Simulated ENOB of CS-ADCs as function of $\frac{C_p}{C_0}$ (*x*-axis) and $\frac{\Delta_{Cp}}{C_p}$ (*y*-axis) results, for resolutions between 7 and 12 bits

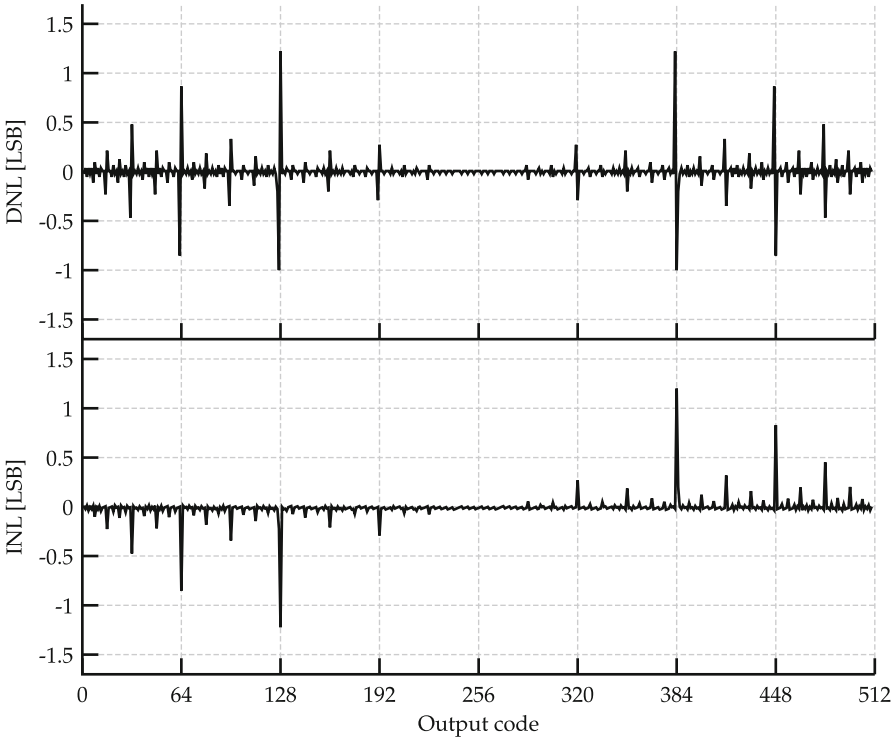



**Fig. 4.11** Simulated DNL/INL shape of a 9-bit CS-ADC with unbalanced parasitics

than in the first and fourth quarters of the codes range, indicating that a compensative effect takes place for the mid-range zone of the transfer curve.

Moreover, unbalance on the parasitics also affects the common-mode voltage of the converter. The simulated worst disturbance of $V_{CM}$ at the comparator inputs as function of $\frac{\Delta_{Cp}}{C_p}$ and $\frac{C_p}{C_0}$ is shown in Fig. 4.12. It is important to note that the results that are shown in Fig. 4.12 are not dependent on the resolution of the ADC. The figure shows that worrisome changes on $V_{CM}$ only take place for exaggerated values of $\frac{\Delta_{Cp}}{C_p}$ and $\frac{C_p}{C_0}$. This observation indicates that for reasonable implementations of CS-ADCs, the effect on common-mode voltage caused by the unbalance in parasitics is negligible.

## 4.6 Errors Caused by Comparator Offset

The case when the comparator has an offset voltage may be modeled by inserting a dc voltage source with the magnitude of the input-referred offset $V_{OS}$ between the DAC and the positive input of the comparator, as shown in Fig. 4.13. If $V_{OS}$ is included in the model, the voltage seen by the comparator at any cycle is given

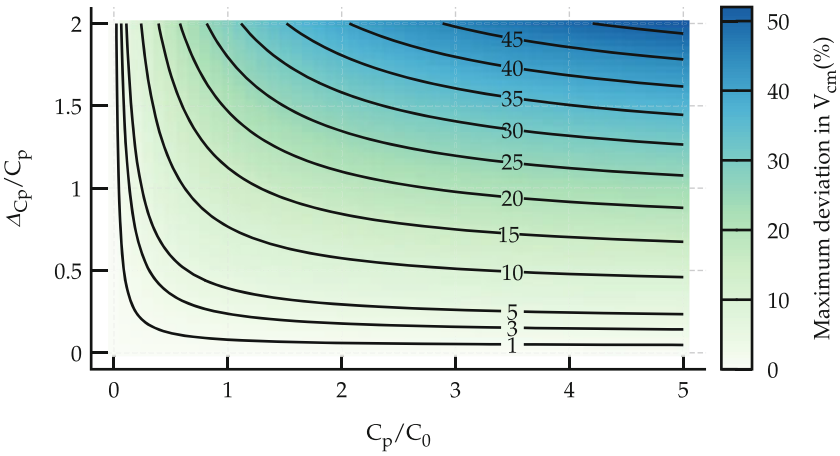
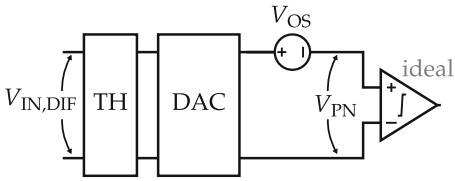



**Fig. 4.12** Simulated maximum disturbance of common-mode voltage on CS-ADCs as function of $\frac{C_p}{C_0}$ ($x$-axis) and $\frac{\Delta_{Cp}}{C_p}$ ($y$-axis) results

**Fig. 4.13** Circuit for analysis of comparator offset




by (4.45).

$$V_{PN}[i] = \frac{\frac{C_{TH}}{2} V_{IN,DIF} - \sum_{j=1}^{i} (2k_{B-j} - 1) C_{B-j-1} V_{PC}}{\frac{C_{TH}}{2} + \sum_{j=1}^{i} C_{B-j-1}} - V_{OS}. \tag{4.45}$$

The thresholds of the transitions for different binary codes $K$, with $K = \{k_{B-1}\, k_{B-2} \ldots k_0\}$, are given in different cycles $i$. For example, the mid-scale transition point from 011..1 to 100..0 occurs at the first cycle $i = 0$, while the transitions from 0011..1 to 0100..0 and from 1011..1 to 1100..0 happen when $i = 1$. Therefore, the cycle $i$ for which $V_{PN} = 0$ for a given code $K$ is a function of $K$, or $i(K)$. In fact, the value of $i(K)$ is easily found with the help of (4.46), where $l$ is the index of $k_l$, and $k_l$ is the least significant "1" in $K$.

$$i(K) = B - 1 - l. \tag{4.46}$$

Differently from what happens with the DAC voltage, the offset voltage is not weighted by the DAC capacitances, causing the nonlinear behavior of the ADC. Equaling $V_{PN}[i]$ to zero and rearranging (4.45) leads to (4.47), that denotes the differential input voltage required on the ADC input in order to make a transition to

the code $K$ on the transfer curve. Also, the factor of $V_{\mathrm{PC}}$ is identified as $\gamma[K]$ and the factor of $V_{\mathrm{OS}}$ as $\eta[K]$.

$$V_{\mathrm{IN,DIF}}[K] = \underbrace{\frac{\sum_{j=1}^{i(K)}(2k_{B-j}-1)C_{B-j-1}}{\frac{C_{\mathrm{TH}}}{2}}}_{\gamma[K]} V_{\mathrm{PC}} + \underbrace{\frac{\frac{C_{\mathrm{TH}}}{2}+\sum_{j=1}^{i(K)} C_{B-j-1}}{\frac{C_{\mathrm{TH}}}{2}}}_{\eta[K]} V_{\mathrm{OS}}. \tag{4.47}$$

Thus, $\gamma[K]$ and $\eta[K]$ are written as (4.48) and (4.49), respectively.

$$\gamma[K] = \frac{2}{C_{\mathrm{TH}}} \sum_{j=1}^{i(K)} \left(2k_{B-j}-1\right) C_{B-j-1} \tag{4.48}$$

$$\eta[K] = 1 + \frac{2}{C_{\mathrm{TH}}} \sum_{j=1}^{i(K)} C_{B-j-1}. \tag{4.49}$$

The expression reveals that $\gamma[k] \cdot V_{\mathrm{PC}}$ describes the ideal behavior of a differential ADC and $\eta[k] \cdot V_{\mathrm{OS}}$ describes the effect of the comparator offset on the transition point. Also, $\eta[K]$ is dictated solely by $i$, and therefore it is determined by $B-1$ unique distinct values, that are arranged in a binary carry sequence [2]. Therefore, in a 3-bit example,

$$\eta = \left\{\eta[1]\ \eta[2]\ \eta[3]\ \eta[4]\ \eta[5]\ \eta[6]\ \eta[7]\right\} \tag{4.50}$$

can be rewritten as

$$\eta = \left\{\eta_2\ \eta_1\ \eta_2\ \eta_0\ \eta_2\ \eta_1\ \eta_2\right\}, \tag{4.51}$$

where $\eta_i$ is the value of $\eta$ for a transition that is determined in the $i$-th cycle. It should be noted that (4.51) is equivalent to (4.50). Moreover, because of the binary-weighted nature of the DAC capacitance, $\eta_i$ can be expressed as in (4.52).

$$\eta_i = 1 + \frac{2^B C_0}{C_{\mathrm{TH}}}\left(1-2^{-i}\right). \tag{4.52}$$

### *4.6.1 Differential Nonlinearity*

The DNL $\delta$ for a given code $K$ in an ADC with differential input is defined as (4.53).

$$\delta[K] = \frac{V_{\mathrm{IN,DIF}}[K+1] - V_{\mathrm{IN,DIF}}[K]}{V_{\mathrm{LSB}}} - 1. \tag{4.53}$$

Since $\gamma[K] \cdot V_{PC}$ describe the behavior of an ideal ADC, the DNL is only function of $\eta[K]$ and $V_{OS}$, and is reduced to (4.54).

$$\delta[K] = \frac{\eta[K+1] - \eta[K]}{V_{LSB}} V_{OS}. \tag{4.54}$$

The largest DNL happens on the MSB transition, since $\eta_0$ is the minimum and $\eta_{B-1}$ is the maximum in $\eta$. Therefore, the worst DNL is given by (4.55).

$$|\delta_{max}| = \frac{2^B C_0}{C_{TH}} \left(1 - 2^{-B+1}\right) \frac{|V_{OS}|}{V_{LSB}}. \tag{4.55}$$

The DNL is inversely proportional to the ratio between the TH capacitance and the DAC capacitance. Therefore, since the total DAC capacitance $C_{DAC}$ is given by (4.57),

$$C_{DAC} = (2^{B-1} - 1)C_0 \tag{4.56}$$

$$= 2^{B-1} C_0 \left(1 - 2^{-B+1}\right), \tag{4.57}$$

the capacitance ratio $\alpha$ is introduced, and can be defined as (4.58).

$$\alpha = \frac{C_{TH}}{2C_{DAC}}. \tag{4.58}$$

Finally, the DNL expression is simplified to (4.59).

$$|\delta_{max}| = \frac{|V_{OS}|}{\alpha V_{LSB}}. \tag{4.59}$$

It is verified that (4.54) presents odd-symmetry. Therefore, if (4.59) returns a value greater than 1 for a given code $K$, (4.52) and (4.54) would result in $\delta[K+1]$ or $\delta[K-1]$ to be lower than $-1$. By definition, a DNL lower than $-1$ in the transfer curve of an ADC represents a nonmonotonic behavior, while a DNL equal to $-1$ represents a missing code. In a SAR ADC, the monotonicity is guaranteed by the architecture, preventing the DNL to present values lower than $-1$. Missing codes, however, are still possible. Therefore, the outcome of a code $K$ with $\delta[K]$ larger than 1 is a range of missing codes on the surroundings of $K$. This is depicted on the simulated DNL/INL curves in Fig. 4.14. The range of codes with DNL equaling $-1$ neighboring a code $K$ with $\delta[K] > 1$ is given by $\lfloor \delta[K] \rfloor$, where $\lfloor \cdot \rfloor$ represents the floor function. The absence of missing codes is guaranteed if (4.60) is verified.

$$|V_{OS}| < \alpha V_{LSB}. \tag{4.60}$$

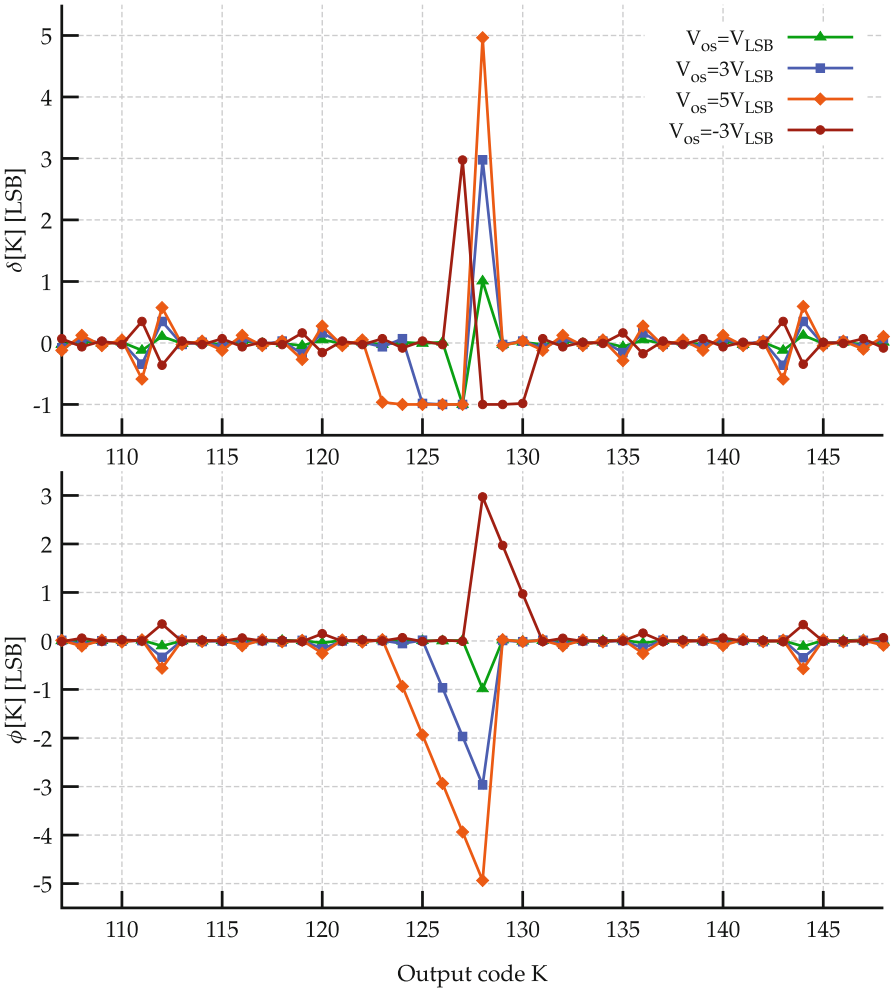



**Fig. 4.14** Simulated DNL and INL curves of an 8-bit CS-ADC with different values of comparator offset, zoomed on the MSB transition

### *4.6.2 Integral Nonlinearity*

The INL of a code $K$, given by $\phi[K]$, is proportional to the difference of the actual transition to the transition of a linear best-fit transfer curve, as given in (4.61).

$$\phi\,[K] = \frac{V_{\mathrm{IN,DIF}}\,[K] - V_{\mathrm{IN,DIF}}^{\mathrm{best-fit}}\,[K]}{V_{\mathrm{LSB}}}. \tag{4.61}$$

The best-fit transfer curve is found by connecting the endpoints of the actual transfer curve. Thus, solving (4.47) for the endpoints shows that the transition points of a best-fit transfer curve are given by (4.62).

$$V_{\text{IN,DIF}}^{\text{best-fit}}[K] = \frac{\sum_{j=1}^{i(K)}(2k_{B-j}-1)C_{B-j-1}}{\frac{C_{\text{TH}}}{2}}V_{\text{PC}} + \left(1+\frac{1}{\alpha}\right)V_{\text{OS}}. \tag{4.62}$$

The second term of (4.62) represents the offset on the actual ADC transfer curve, caused by the comparator offset, which is given by (4.63).

$$\text{ADC}_{\text{offset}} = \left(1+\frac{1}{\alpha}\right)V_{\text{OS}}. \tag{4.63}$$

Substituting (4.47) and (4.62) into (4.61) and solving for different values of $K$ reveals that the $\phi$ also presents a binary carry sequence pattern. Thus, for a 3-bit example, $\phi$ is distributed according to (4.64).

$$\phi = \left\{\phi_2\ \phi_1\ \phi_2\ \phi_0\ \phi_2\ \phi_1\ \phi_2\right\}. \tag{4.64}$$

The value of $\phi_i$, for any resolution, is given by (4.65).

$$\phi_i = \left(2^{-B+1}-2^{-i}\right)\frac{V_{\text{OS}}}{V_{\text{LSB}}}.\frac{2^B C_0}{C_{\text{TH}}}. \tag{4.65}$$

The worst-case INL happens for the MSB transition, or $\phi_0$, as given in (4.66).

$$\phi_0 = \left(2^{-B+1}-1\right)\frac{V_{\text{OS}}}{V_{\text{LSB}}}.\frac{2^B C_0}{C_{\text{TH}}}. \tag{4.66}$$

Finally, (4.66) can be rewritten as a function of $\alpha$, leading to (4.67).

$$\phi_0 = -\frac{V_{\text{OS}}}{\alpha V_{\text{LSB}}}. \tag{4.67}$$

It is to be noted that (4.64) and (4.65) only hold if there are no missing codes in the transfer curve. In order to deal with the enforced monotonicity of the CS-ADC, a different method to calculate $\phi$ in the presence of missing codes must be developed. In this work, this is approached through an alternative definition of $\phi[K]$, which is given in (4.68).

$$\phi[K] = \sum_{j=1}^{K-1}\delta[j]. \tag{4.68}$$

Because of the missing codes surrounding a code with $\delta[K] > 1$, a range of values of INL around a missing code $K$ follows a sequence where the terms are spaced by $-1$ (e.g., {... $\phi[K]$ $\phi[K]-1$ $\phi[K]-2$ ...}), and can be approximated by a triangle shape with legs equal to $|\phi[K]|$, as seen in Fig. 4.14. This characteristic is particularly useful on the derivation of the effective resolution in the presence of comparator offset, that follows.

### 4.6.3 ENOB

The quantization noise of an ideal ADC is defined as in (4.69).

$$V_{\text{NOISE}}{}^2 = \frac{1}{(2^B-1)\,V_{\text{LSB}}} \sum_{j=1}^{2^B-1} \int_{-V_{\text{LSB}}/2}^{V_{\text{LSB}}/2} u^2 \,\mathrm{d}\,u = \frac{V_{\text{LSB}}{}^2}{12}. \tag{4.69}$$

The quantization noise of a non-ideal ADC can be described as function of the INL by changing the integration limits, as (4.70).

$$V_{\text{NOISE}}{}^2 = \frac{1}{(2^B-1)\,V_{\text{LSB}}} \sum_{j=1}^{2^B-1} \int_{(\phi[j]-\frac{1}{2})V_{\text{LSB}}}^{(\phi[j+1]+\frac{1}{2})V_{\text{LSB}}} u^2 \,\mathrm{d}\,u. \tag{4.70}$$

Solving (4.70) yields (4.71).

$$V_{\text{NOISE}}{}^2 = \frac{1}{(2^B-1)\,V_{\text{LSB}}} \sum_{j=1}^{2^B-1} \frac{u^3}{3}\Bigg|_{(\phi[j]-\frac{1}{2})V_{\text{LSB}}}^{(\phi[j+1]+\frac{1}{2})V_{\text{LSB}}}. \tag{4.71}$$

Finally, expanding (4.71) leads to (4.72).

$$V_{\text{NOISE}}{}^2 = \frac{1}{(2^B-1)\,V_{\text{LSB}}} \sum_{j=1}^{2^B-1} \left[ \frac{V_{\text{LSB}}{}^3}{12} \left(-4\phi[j]^3 + 4\phi[j+1]^3 + 6\phi[j]^2 \right.\right.$$
$$\left.\left. +6\phi[j+1]^2 - 3\phi[j] + 3\phi[j+1] + 1\right)\right]. \tag{4.72}$$

Note that most of the terms are canceled when we compute the sum, because they appear twice with different signs. Also, by definition, the INL of the first and the last code transitions ($\phi[1]$ and $\phi[2^B-1]$) are zero. Therefore, (4.72) may be simplified to (4.73).

$$V_{\text{NOISE}}{}^2 = \frac{1}{(2^B-1)\,V_{\text{LSB}}} \sum_{j=1}^{2^B-1} \left[ \frac{V_{\text{LSB}}{}^3}{12} + V_{\text{LSB}}{}^3 \phi[j]^2 \right]. \tag{4.73}$$

Finally, solving the summation leads to (4.74).

$$V_{\text{NOISE}}{}^2 = \frac{V_{\text{LSB}}{}^2}{12} + \frac{V_{\text{LSB}}{}^2}{2^B - 1} \sum_{j=1}^{2^B-1} \phi[j]^2. \tag{4.74}$$

Being $\overline{\phi^2}$ the mean of $\phi^2$, (4.74) can be rewritten as a more intuitive form, as in (4.75).

$$V_{\text{NOISE}}{}^2 = V_{\text{LSB}}{}^2 \left( \frac{1}{12} + \overline{\phi^2} \right). \tag{4.75}$$

The next step assumes that there are missing codes on the transfer curve and that the INL of the code $K$, $\phi[K]$, is a positive integer. Exploiting the enforced monotonicity and the shape of $\phi$, as already anticipated in the previous subsection, the sum of the values of $\phi^2$ caused by the nonlinearity on the transition to the code $K$ are given by (4.76), which represents the sum of the squares of the first $\phi[K]$ natural numbers.

$$\sum_{j=0}^{\phi[K]} \phi[K+j]^2 = \sum_{i=0}^{\phi[K]} i^2. \tag{4.76}$$

The expression is further simplified using series identities.

$$\sum_{i=0}^{\phi[K]} i^2 = \frac{\phi[K]^3}{3} + \frac{\phi[K]^2}{2} + \frac{\phi[K]}{6}. \tag{4.77}$$

So far, it is assumed that $\phi[x]$ is an integer, which is rarely the case. However, it can be shown that (4.77) provides a very good approximation even when $\phi[K]$ is not integer. The absolute approximation error $\epsilon_\phi$ may be computed using (4.78), where $\lceil \cdot \rceil$ represents the ceiling function.

$$\epsilon_\phi = \sum_{j=0}^{\lceil |\phi[K]| \rceil} \phi[K-j]^2 - \left( \frac{\phi[K]^3}{3} + \frac{\phi[K]^2}{2} + \frac{\phi[K]}{6} \right). \tag{4.78}$$

Figure 4.15 shows the resulting value of the sum, together with the error (absolute and relative) caused by the approximation, for $\phi[x]$ ranging from 0 to $10V_{\text{LSB}}$. The relative error converges to zero as the value of $\phi[K]$ grows. While this simplification greatly eases the computations, it is verified that it does not significantly affect the accuracy of the model of ENOB in the presence of comparator offset.

Now, (4.77) may be used to find the noise contribution of the INL at every transition on the transfer curve. Again, exploiting the symmetry of $\phi$ (e.g., the $\phi_1$ code transition pattern happens twice with approximately half the magnitude of $\phi_0$,

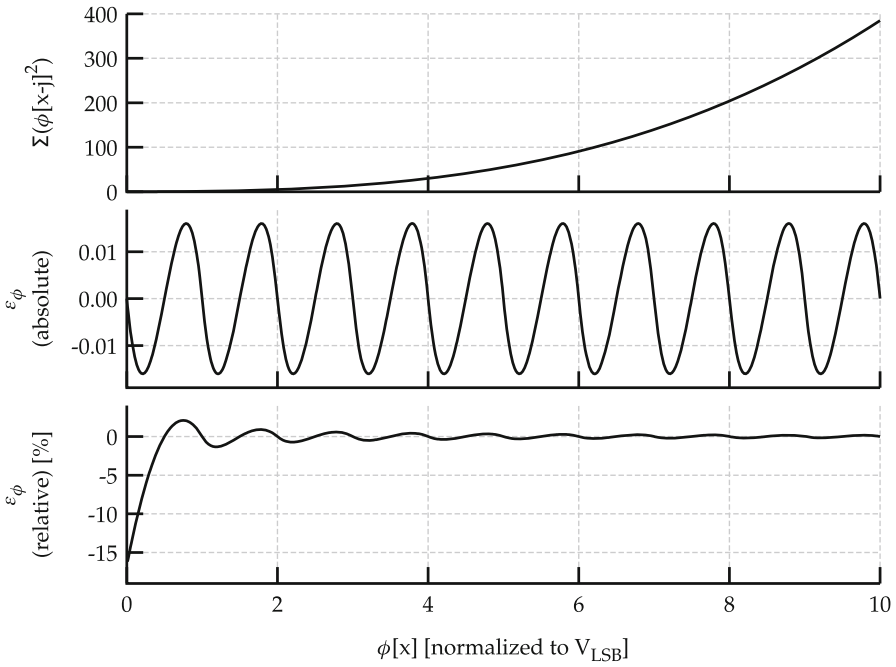



**Fig. 4.15** Absolute and relative error caused by the approximation in (4.77)

$\phi_2$ code transition pattern happens four times with approximately one fourth of the magnitude of $\phi_0$, etc, as given by (4.64) and (4.65)), the following developments are possible:

$$\overline{\phi^2} = \frac{1}{2^B - 1} \sum_{j=1}^{2^B-1} \phi[j]^2 \tag{4.79}$$

$$\approx \frac{1}{2^B - 1} \sum_{j=0}^{B-1} \left( 2^j \sum_{m=0}^{\lceil |\phi[2^{B-1-j}]| \rceil} \phi[2^{B-1-j} + m]^2 \right) \tag{4.80}$$

$$\approx \frac{1}{2^B - 1} \sum_{j=0}^{B-1} \left( 2^j \left[ \frac{1}{3} \left( \frac{\phi_0}{2^j} \right)^3 + \frac{1}{2} \left( \frac{\phi_0}{2^j} \right)^2 + \frac{1}{6} \left( \frac{\phi_0}{2^j} \right) \right] \right). \tag{4.81}$$

Expression (4.81) is further simplified by using geometric series identities, yielding (4.82).

$$\overline{\phi^2} \approx \frac{1}{2^B - 1} \left[ \frac{4}{9} \phi_0^3 \left( 1 - 2^{-2B} \right) + \phi_0^2 \left( 1 - 2^{-B} \right) + \frac{B}{6} \phi_0 \right]. \tag{4.82}$$

For reasonable values of $B$, $\overline{\phi^2}$ can be approximated by (4.83).

$$\overline{\phi^2} \approx \frac{1}{2^B}\left(\frac{4}{9}\phi_0^3 + \phi_0^2 + \frac{B}{6}\phi_0\right). \tag{4.83}$$

We can relate $V_{\text{NOISE}}$ and ENOB assuming a full-range input signal as (4.84) [1].

$${V_{\text{NOISE}}}^2 = \frac{{V_{\text{LSB}}}^2}{12}\left(\frac{2^B}{2^{\text{ENOB}}}\right)^2. \tag{4.84}$$

Equaling (4.75) to (4.84), we find (4.85).

$$\frac{{V_{\text{LSB}}}^2}{12}\left(1 + 12\overline{\phi^2}\right) = \frac{{V_{\text{LSB}}}^2}{12}\left(\frac{2^B}{2^{\text{ENOB}}}\right)^2. \tag{4.85}$$

Simplifying (4.85) leads to (4.86).

$$1 + 12\overline{\phi^2} = \frac{2^{2B}}{2^{2\text{ENOB}}}. \tag{4.86}$$

Isolating ENOB, and finding the base-2 logarithm of the two sides, leads to

$$\log_2\left(2^{2\text{ENOB}}\right) = \log_2\left(\frac{2^{2B}}{1 + 12\overline{\phi^2}}\right). \tag{4.87}$$

The expression can be rewritten as (4.88)

$$\log_2\left(2^{2\text{ENOB}}\right) = \log_2\left(2^{2B}\right) - \log_2\left(1 + 12\overline{\phi^2}\right). \tag{4.88}$$

Applying simplification on the expression leads to (4.89).

$$\text{ENOB} = B - \frac{\log_2\left(1 + 12\overline{\phi^2}\right)}{2}. \tag{4.89}$$

Since $\log_2(4) = 2$, we can rewrite (4.89) as (4.90).

$$\text{ENOB} = B - \frac{\log_2\left(1 + 12\overline{\phi^2}\right)}{\log_2(4)}. \tag{4.90}$$

Also, since

$$\log_4(x) = \frac{\log_2(x)}{\log_2(4)}, \tag{4.91}$$

(4.90) can be rewritten as (4.92).

$$\text{ENOB} = B - \log_4\left(1 + 12\overline{\phi^2}\right). \tag{4.92}$$

The ENOB is found substituting (4.83) into (4.92), which is derived from (4.75), resulting in (4.93).

$$\text{ENOB} \approx B - \log_4\left(1 + 2^{2-B}\left(\frac{4}{3}\phi_0^3 + 3\phi_0^2 + \frac{B}{2}\phi_0\right)\right). \tag{4.93}$$

The expression in (4.93) assumes that the input signal follows a uniform probability distribution. On the other hand, the most frequent approach to characterize the effective resolution of ADCs uses a sinusoidal test signal. Therefore, following the approach proposed in [1], the scalar correction factor $\psi$ is used to reconcile the expression with the preferred sinusoidal testing method. Here, $\psi$ is the ratio between the INL noise contribution from a sinusoidal distribution and a uniform distribution. To estimate the value of $\psi$, the INL of a CS-ADC with comparator offset is simulated. Then, the average noise power contributed by the INL with a uniformly distributed input is calculated by numerically integrating the curve and normalizing to the code range, finally obtaining $\left(\overline{\phi^2}\right)_{\text{uniform}}$. In order to obtain $\left(\overline{\phi^2}\right)_{\text{sinusoid}}$, the squared INL error is weighted by the probability density function (PDF) of a sinusoid, that is given by (4.94), where $K$ represents the output code, and $K_{\min}$ and $K_{\max}$ represent the minimum and maximum output codes that are driven by the input signal.

$$\text{PDF}_{\text{sine}}[K] = \frac{1}{\pi\sqrt{(K - K_{\min})(K_{\max} - K)}}. \tag{4.94}$$

The squared INL weighted by the uniform and sinusoidal distributions are shown in Fig. 4.16, together with the corresponding PDFs. The procedure is mathematically described in (4.95).

$$\psi = \frac{\left(\overline{\phi^2}\right)_{\text{sinusoid}}}{\left(\overline{\phi^2}\right)_{\text{uniform}}} = \frac{\sum_{j=1}^{2^B-1}\left(\phi[j]^2 \cdot \text{PDF}_{\text{sine}}[j]\right)}{\sum_{j=1}^{2^B-1}\left(\phi[j]^2 \cdot \frac{1}{2^B-1}\right)}. \tag{4.95}$$

Then, the ENOB expression is rewritten as (4.96).

$$\text{ENOB} = B - \log_4\left(1 + 12\,\psi\,\overline{\phi^2}\right). \tag{4.96}$$

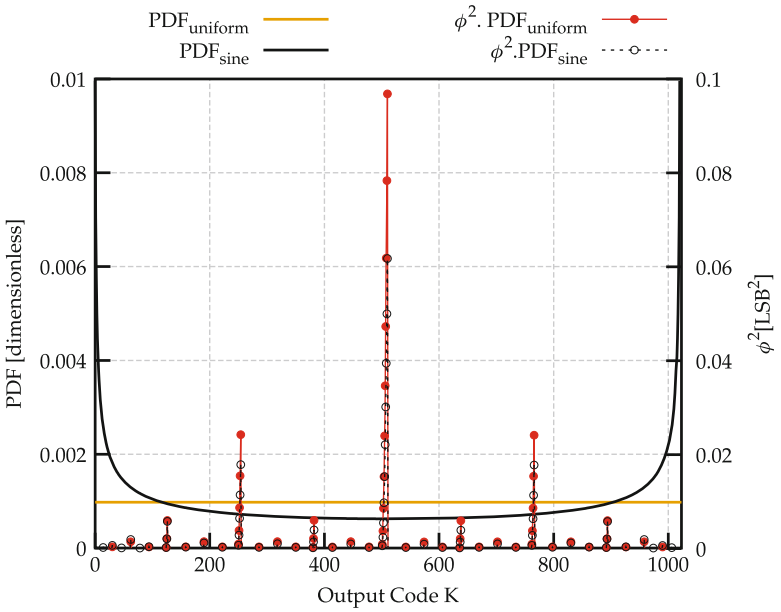



**Fig. 4.16** *Left axis*: PDFs of uniform and sinusoidal full-range inputs; *right axis*: squared INL weighted by these PDFs

Using $V_{OS} = 10V_{LSB}$ and a resolution of 10 bits for the ADC, which are values laid inside a range of practical values, the value of $\psi \approx 0.67$ is obtained. Finally, the ENOB expression for a CS-ADC with comparator offset driven by a sinusoidal input is given by (4.97), where $\phi_0$ is given by (4.67).

$$\text{ENOB} \approx B - \log_4\left(1 + 2^{3-B}\left(\frac{4}{9}\phi_0^3 + \phi_0^2 + \frac{B}{6}\phi_0\right)\right). \tag{4.97}$$

### *4.6.4 Model Verification*

The model of ENOB versus comparator offset is validated against behavioral simulations. This is done for different combinations of resolution (between 7 and 12 bits) and $\alpha$ (0.5, 1, 2, 3, 5). For each combination, the comparator offset voltage is swept from 0 to 5 times the $V_{LSB}$ of the 7-bit ADC ($5 \cdot V_{LSB,8b}$), in a total of 100 linearly spaced steps. For the same input range, $5 \cdot V_{LSB,7b}$ is the same as $10 \cdot V_{LSB,8b}$, $20 \cdot V_{LSB,9b}$, $40 \cdot V_{LSB,10b}$, and so on, and the $x$-axis of all the plots span the same absolute voltage range, which is approximately 80 mV for a differential input range of 2 V. The simulated ADCs are driven by a sinusoid with peak-to-peak amplitude equal to 95 % of the input range and a dc-offset given by (4.63), to avoid saturation.

The ENOB is extracted from a $2^{17}$-point FFT. These results are compared with the results achieved with (4.97) in Fig. 4.17. The relative error between the analytical ENOB and the simulated ENOB is found with (4.98)

$$\epsilon_{\text{ENOB}} = \left( \frac{\text{ENOB}_{\text{analytical}}}{\text{ENOB}_{\text{simulated}}} - 1 \right) \cdot 100\,\%. \tag{4.98}$$

Figure 4.18 plots $\epsilon_{\text{ENOB}}$ for ADCs with resolutions between 6 and 14 bits, $\alpha$ between 0.5 and 5, 2 V of input range and an exaggerated range of comparator offset voltage of 0–100 mV. The plot reveals that the model is less accurate for low values of $\alpha$ and low resolutions, because of the assumptions made while calculating $\psi$. For resolutions above 8-bits with $\alpha$ larger than 1, $|\epsilon_{\text{ENOB}}| < 2\,\%$. Interestingly, the ENOB achieved with (4.97) is slightly pessimistic, despite the approximations used during the derivation.

### *4.6.5 Discussion*

Some important design guidelines can be derived from the analysis: the CS-ADC presents missing codes if (4.60) is not satisfied; the worst-DNL happens at the MSB transition and is given by (4.59); the worst-INL also happens at the MSB transition, and is given by (4.67); the ADC transfer curve presents an offset that is given by (4.63); the maximum achievable effective resolution in a CS-ADC that has comparator offset is approximated by (4.97); finally, the capacitance ratio $\alpha$ can be increased to improve the ADC tolerance to comparator offset.

## 4.7 Errors Caused by Noise

The two operation phases of a CS-ADC, sampling/precharge and binary search, are affected by noise. During the sampling phase, thermal noise is sampled together with the input signal. The noise charge $\sigma_{\text{q,TH}}$ acquired in a CS-ADC during the sampling phase is given by (4.99), where $k$ is the Boltzmann constant, $T$ is the absolute temperature, and $C_{\text{TH}}$ is the size of the TH capacitor. The factor of 2 in the numerator appears because the pair of TH capacitors appear in series for differential signals.

$$\sigma_{\text{q,TH}} = \sqrt{kT\frac{C_{\text{TH}}}{2}} \quad \text{(C)}. \tag{4.99}$$

Similarly, a reset noise charge with the value of (4.100) is left in the DAC when the precharge switches are opened.

$$\sigma_{\text{q,DAC}} = \sqrt{kTC_{\text{DAC}}} \quad \text{(C)}. \tag{4.100}$$

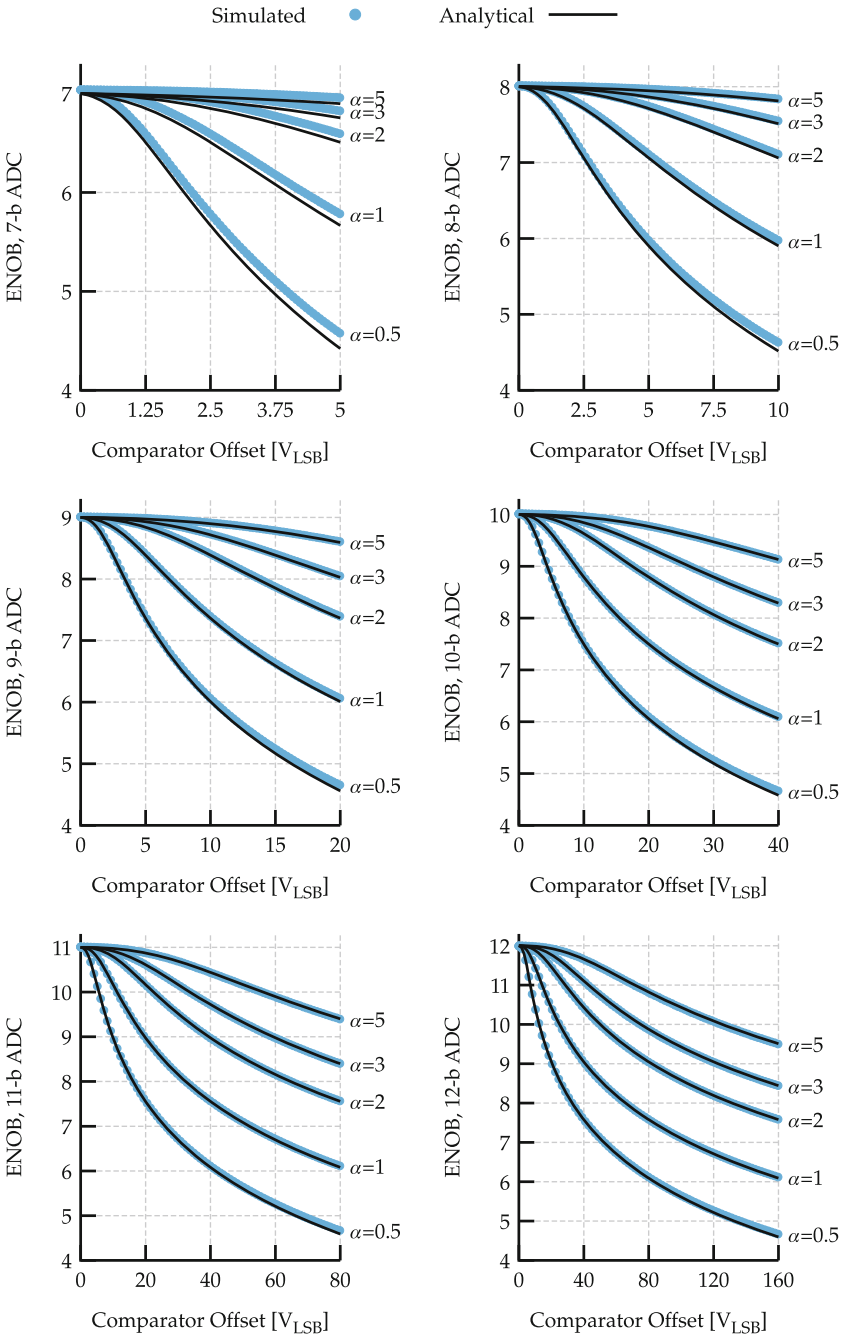



**Fig. 4.17** Comparison between the analytical and simulated ENOB obtained from 500 simulations for each combination of $B$ and $\alpha$, extracted with a $2^{17}$-point FFT with a sinusoidal input near Nyquist frequency with 98 % of the full-range

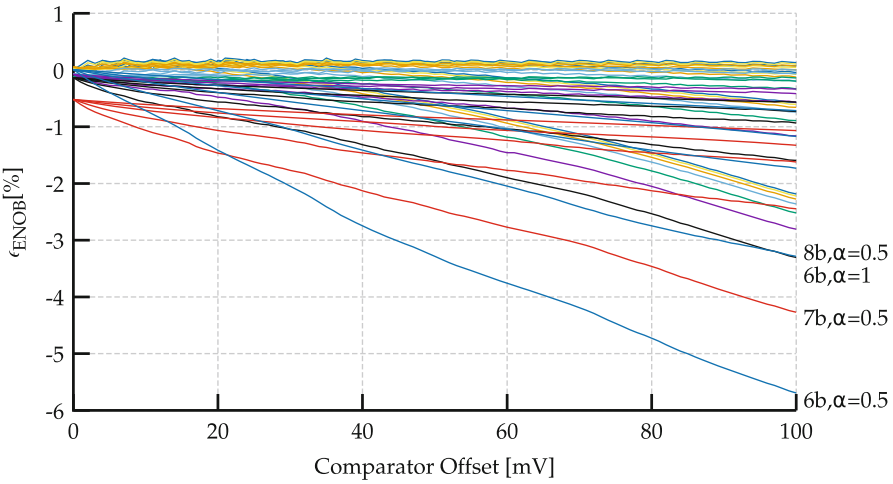



**Fig. 4.18** Plot of $\epsilon_{ENOB}$ for all resolutions between 6 and 14 bits, with $\alpha$ ranging from 0.5 to 5, for ADCs with 2 V of input range

During the binary search phase, the thermal noise on the capacitors connected to the comparator (TH capacitors and the DAC capacitors corresponding to the bits already resolved) is summed up to the comparator intrinsic noise, affecting the conversion.

To illustrate the orders of magnitude of the noise values, the thermal noise voltage on a 1 pF capacitance is approximately 64 μV, while the majority of comparators reported in literature present an intrinsic noise which is nearly ten times larger (0.5 mV–2 mV) [3]. It is important to note that a tenfold factor in voltage equals a hundredfold factor in power. Therefore, for common values of resolution for SAR ADCs (8–12 bits) and common values of DAC and TH capacitors, which are generally lower-bounded because of matching and process limitations for minimum unit-capacitor size, it is fair to assume that the comparator noise dominates during binary search. Additionally, the total noise power resulting from (4.99) and (4.100) is hundreds of times smaller than the charge-domain quantization noise ($\frac{Q_{LSB}^2}{12}$) for a $V_{PC}$ of 0.6 V, for instance. In conclusion, for the following analysis, only the comparator noise is considered.

A rigorous mathematical analysis of noise in SAR ADCs is out of the scope of this book. The reader is referred to the work recently published in [3] for a complete statistical derivation of the noise phenomena in SAR ADCs. In this work, simulations were used to quantify the impact of noise in the CS-ADC. As described previously in Chap. 3, the CS-ADC is less noise-tolerant than the CR counterpart. Therefore, simulations were also run for the CR-ADC, for comparison purposes. It is also noticeable, as in the case of comparator offset, that the tolerance of the ADC to comparator noise is affected by the capacitance ratio $\alpha$, defined in (4.58). Figure 4.19 shows the effective resolution achieved as a function of comparator

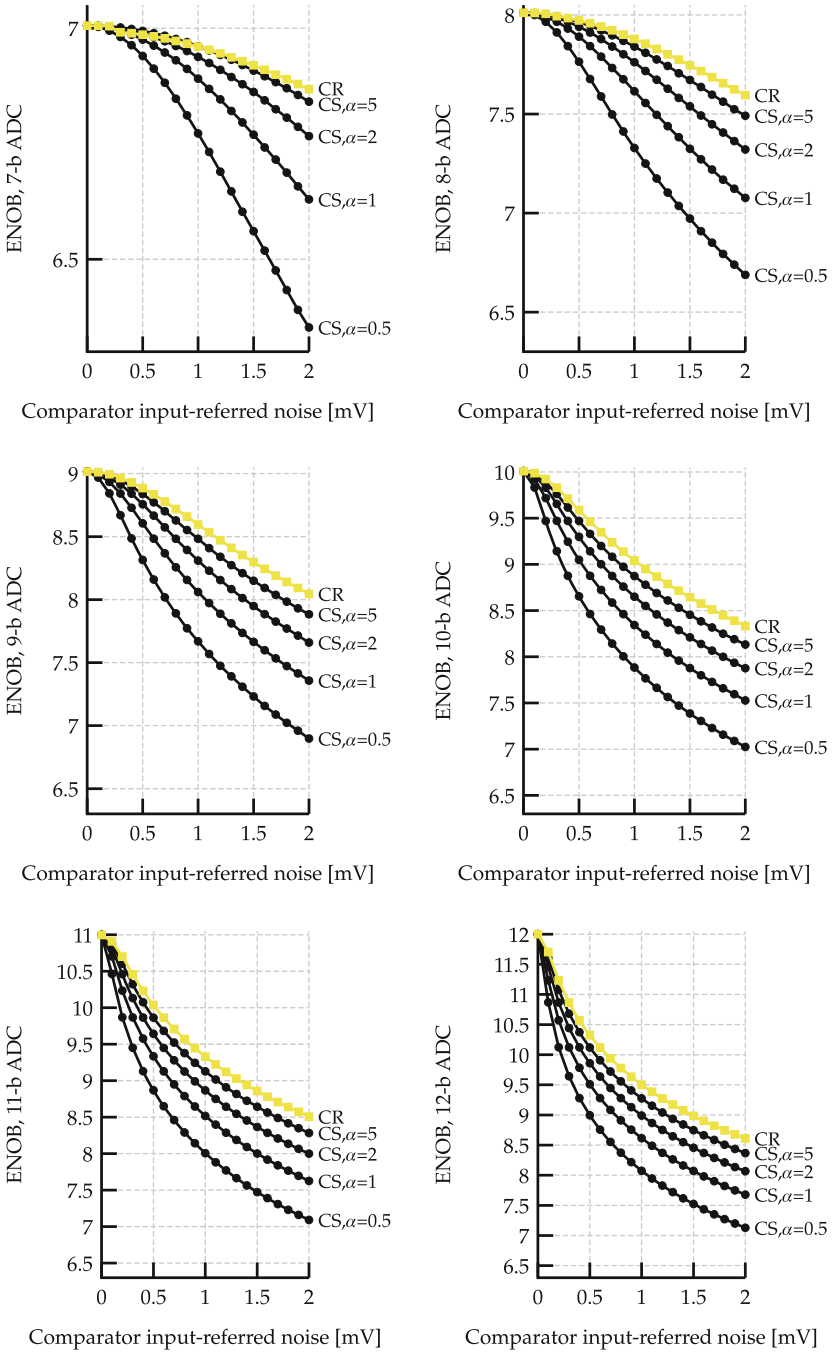



**Fig. 4.19** Comparison between the ENOB obtained from 20 simulations for each combination of $B$ and $\alpha$, extracted with a $2^{20}$-point FFT with a sinusoidal input near Nyquist frequency with 99.9 % of the 2-$V_{P-P}$ differential full-range

noise for various combinations of $\alpha$ and resolution, for ADCs with 2 $V_{P-P}$ of input range. Each point in the plot is computed through a $2^{20}$-point FFT, considering a normal probability density function for the comparator noise, while the sinusoidal input signal drives 99.9 % of the full-range.

As can be seen in the figure, the CS topology is less tolerant to the comparator noise than the CR. This is expected, because of the signal attenuation caused by the DAC capacitors being connected during the conversion cycle. Moreover, as the capacitance ratio $\alpha$ grows, the noise performance of the CS-ADC approaches that of the CR-ADC. This suggests that the factor $\alpha$ may be exploited to improve noise performance, and is discussed in greater detail in Chap. 7.

## References

1. J.A. Fredenburg, M.P. Flynn, Statistical analysis of ENOB and yield in binary weighted ADCs and DACS with random element mismatch. IEEE Trans. Circuits Syst. Regul. Papers **59**(7), 1396–1408 (2012). doi:10.1109/TCSI.2011.2177006 (cit. on pp. 82, 95, and 96)
2. E. Weisstein, *CRC Concise Encyclopedia of Mathematics*, 2nd edn. (Taylor & Francis, London, 2002). isbn: 9781420035223 (cit. on p. 88)
3. W.P. Zhang, X. Tong, Noise modeling and analysis of SAR ADCs. IEEE Trans. Very Large Scale Integr. Syst. **23**(12), 2922–2930 (2015). doi:10.1109/TVLSI.2014.2379613 (cit. on p. 100)

# Chapter 5
# Noise-Aware Synthesis and Optimization of Voltage Comparators

**Contents**



## 5.1 Introduction

A comparator is a circuit able to compare signals and switch its outputs indicating which is larger, and is mandatory in ADCs. Some performance metrics of the comparator directly affect the overall ADC performance, such as power consumption and maximum speed. Still, in the context of ADCs, the stochastic phenomena involved in the comparison process (i.e., offset voltage and noise) may lead to errors on the converters. As discussed before, in some architectures, the comparator offset leads to an offset on the ADC transfer curve, while in others such as the CS-ADC, it may appear as nonlinearity. Fortunately, diverse offset calibration schemes have been devised to bring the comparator offset to levels that satisfy ADC specifications. On the other hand, the noise generated by the comparator circuit during the comparison is more critical, since it appears added to the quantization noise at the ADC output. Some ADC designs on the literature report that a significant drop on the effective resolution was caused by comparator noise [1]. Recently, a technique that employs averaging was proposed in [2] to reduce the impact of comparator noise on the ADC SNR, at the cost of increased energy consumption and reduced speed.



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_5

Nevertheless, comparator noise is an important performance metric, and its estimation is nontrivial, as most of the comparator architectures employed in modern designs rely on a positive feedback loop to speed up the operation and avoid meta-stability. In this context, the operation of such a circuit is highly nonlinear and does not have a constant steady-state operating point. Even though time-domain analysis together with transient noise is possible, it poses a significant computational cost and is extremely time-consuming. In that approach, many comparison cycles must be simulated to achieve sufficient accuracy.

There has been some effort to provide better estimation techniques for the comparators noise. In [3], the authors analyze the noise on comparators relying on the use of stochastic differential equations and provide valuable design guidelines. In [4], the authors provide a LTV analysis framework for the same purpose, which is based on the ISF. Also, the latter approach enables simulation-based verification of comparator noise relying on the use of techniques commonly applied to RF circuits with a PSS operating point [5]. This allows the quick and accurate verification of input referred noise employing an RF circuit simulator bundle, such as SpectreRF or HSPICE-RF.

Still, optimizing the comparator performance poses a very complex task if multiple performance metrics are considered, due to the large amount of trade-offs involved between specifications. If this is summed up to tight requirements, progressive reduction of minimum design features and complex transistor models, it leads to a tough burden even for experienced designers. On the other hand, considering the increasing availability of computational resources, it becomes advantageous the use of computer optimization tools to carry out this task.

In this chapter, we present a methodology to optimize comparator circuits regarding power, delay, and noise. This is done employing a multi-objective evolutionary algorithm together with a simplified method to evaluate the comparator metrics on RF circuit simulators, based on [4]. The accuracy of the method is verified comparing the results with the conventional approach based on transient noise simulation.

This chapter is organized as follows. Section 5.2 provides the theoretical background for the developed estimation method of comparator noise. Following, Sect. 5.3 describes the simulation-based noise measurement procedure, Sect. 5.4 reports the optimization framework based on evolutionary algorithms, and Sect. 5.5 provides a design example for a comparator architectures widely used in literature. Finally, Sect. 5.6 presents and discusses the optimization results, and Sect. 5.7 presents some final remarks.

## 5.2 Review on Comparator Noise Calculation

The ISF $\Gamma(\tau)$ was initially devised for oscillators in [6], and was later generalized for other classes of periodic circuits in [7]. It expresses the time-varying impulse response for impulses arriving at the time $\tau$, evaluated at a predefined observation time instant $t_{\text{obs}}$. Thus, the output voltage of a LTV system observed at $t_{\text{obs}}$ may be

written as in (5.1), demonstrating that the output is a weighted average of the input signal $v_i$ using $\Gamma(\tau)$ as the weighting factor.

$$v_o(t_{\text{obs}}) = \int_{-\infty}^{\infty} v_i(\tau) \cdot \Gamma(\tau)\ \mathrm{d}\tau. \tag{5.1}$$

Similarly, we are able to develop ISFs for all the noise contributors on a given circuit and find the total output noise at $t_{\text{obs}}$. Thus, assuming only white noise sources, the output noise $\sigma_{n,o}(t_{\text{obs}})$ of a system may be expressed as a function of all the $N$ noise sources as in:

$$\sigma_{n,o}(t_{\text{obs}}) = \sqrt{\sum_{i=0}^{N} \sigma_i^2 \int_{-\infty}^{\infty} \Gamma_i^2(\tau)\ \mathrm{d}\tau}. \tag{5.2}$$

Since we are generally interested in the noise power referred to the input of the system, $\sigma_{n,o}$ has to be divided by its near-dc gain $G$:

$$\sigma_{n,i}(t_{\text{obs}}) = \frac{\sigma_{n,o}(t_{\text{obs}})}{G}. \tag{5.3}$$

The near-dc gain of the system is expressed as (5.4) and is equal to the area of the ISF.

$$G = \frac{v_o(t_{\text{obs}})}{v_i(t_{\text{obs}})} = \int_{-\infty}^{\infty} \Gamma(\tau)\ \mathrm{d}\tau. \tag{5.4}$$

In the context of RF circuit simulation, $\sigma_{n,o}$ and $G$ may be found through the use of PNOISE and PAC analyses, respectively, once that $t_{\text{obs}}$ has been properly chosen. This allows us to find $\sigma_{n,i}$ using (5.3). A much more detailed explanation may be found in [4, 7].

## 5.3 Comparator Noise Measurement

We depict the procedure for simulating the comparator noise in Figs. 5.1 and 5.2, where we show the comparator test bench, an example of simulation commands for SpectreRF and the expected simulation waveforms. The requirement to simulate the comparator with PSS/PNOISE analyses is the test setup to be periodic, with the clock signal period equal to the beat period $T_b$ of the simulation. The input differential voltage source must have a small dc amplitude to avoid meta-stability (we have employed 0.5 mV with success). This same voltage source must also have a nonzero PAC magnitude (we use 1 V), allowing us to calculate the PAC gain later. Then, the measurement procedure follows the steps below:

```
pss pss fund=1/Tb outputtype=time
+    errpreset=conservative

pnoise ( outp outn ) pnoise start=1 stop=1/(2*Tb)
+    pnoisemethod=fullspectrum iprobe=Vdif
+    refsideband=0 noisetype=pmjitter
+    crossingdirection=rise thresholdvalue=vdd/2

pac ( outp outn ) pac maxsamples=1
+    crossingdirection=rise thresholdvalue=vdd/2
+    ptvtype=sampled sweeptype=relative
+    relharmnum=0 start=1 maxsideband=0
```

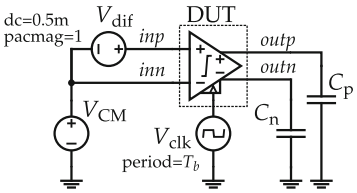



**Fig. 5.1** Test setup for the comparator noise measurement using PSS/PNOISE, with example SpectreRF commands

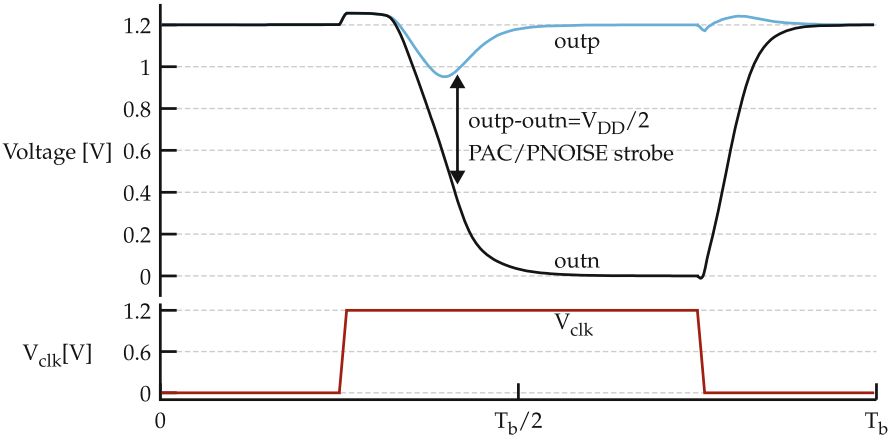



**Fig. 5.2** Illustrative waveforms

1. Initially, the periodic steady-state operating point of the circuit must be found through a PSS analysis;
2. Based on the PSS response, the observation time $t_{\text{obs}}$ has to be chosen (the procedure is described next);
3. Then, we run a PNOISE and a PAC at the specified $t_{\text{obs}}$;
4. The near-dc gain at $t_{\text{obs}}$ is found from the PAC response;
5. The output noise power is found applying the result from the PNOISE analysis in (5.5), where $\text{PSD}_n$ is the noise PSD and $T_b$ is the PSS beat period. We integrate from 0 to $\frac{1}{2T_b}$ because SpectreRF outputs the single-sided PSD.

$$\sigma_{n,o}(t_{\text{obs}}) = \sqrt{\int_0^{1/2T_b} \text{PSD}_n(f, t_{\text{obs}})\ \text{d}f}. \tag{5.5}$$

6. Thus, $\sigma_{n,i}$ is found by plugging (5.5) into (5.3).

An important step of this method to measure comparator input noise is the proper choice of $t_{\text{obs}}$. In [4], the authors develop the choice criteria and demonstrate that this choice is not unique, as a range of time points satisfy the requirements. They also present two methods with this purpose: choose $t_{\text{obs}}$ where the small-signal gain

$G(t_{\text{obs}})$ has the maximum value; and choose the time point where the incremental gain $G^*(t_{\text{obs}})$ computed from two large-signal responses deviates more than 10 % from $G(t)$. However, both the proposed methods rely on post-processing, and need the simulator to evaluate PAC and PNOISE at a range of time points for later choice of $t_{\text{obs}}$, significantly increasing the simulation time. Therefore, these methods are not ideal for an optimization framework, where the computational time spent on each iteration must be reduced, and a one-step approach is preferred.

We have employed a different and very straightforward approach for choosing $t_{\text{obs}}$. SpectreRF allows the evaluation of strobed PAC and PNOISE responses. In other words, we directly calculate the gain and noise when a given trigger signal crosses a specified threshold level. For PNOISE, this is only accessible by the phase modulation jitter (pmjitter) mode that, even though is devised for jitter calculation, also reports the output noise voltage. The trigger signal, in our case, is the differential output signal of the comparator. This is better understood with the example SpectreRF commands shown in Fig. 5.1. We have set the strobing threshold voltage as $V_{\text{DD}}/2$, and we show later in Sect. 5.6 that this choice provides accurate and reliable results.

## 5.4 Multi-Objective Optimization Framework

The optimization framework employed in this work uses a kernel based on the NSGA-II [8] multi-objective GA. The GAs are a class of algorithms based on the principles of population dynamics, and allow the use of black-box models (no gradients necessary) in the evaluation function. This characteristic makes these algorithms good candidates for SPICE-simulation-in-a-loop circuit sizing. Our custom Python-language implementation allows objectives with the forms of "minimize" and "maximize," and constraints with the forms of "smaller than X," "larger than X," and "between X and Y." We have used a polynomial mutation operator with $\eta_m = 20$ and a probability of chromosome mutation of 5 %, and a simulated binary crossover operator with $\eta_c = 20$. A more detailed description of these parameters is found in [8].

## 5.5 Design Example

Using the procedure for noise measurement and the multi-objective optimization framework previously described, we have sized the comparator topology shown in Fig. 5.3. This is a dynamic comparator architecture commonly found in literature, which uses positive feedback to speed up the operation and avoid meta-stability. The mutation probability of the genetic algorithm was set to 10 % and the crossover probability to 90 %. We constrained the sizing to 156 generations of 64 individuals, resulting in roughly 10,000 evaluations. We have employed a 0.13 µm CMOS

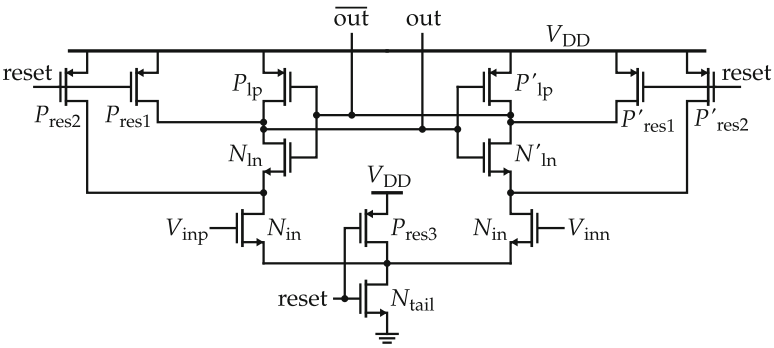



**Fig. 5.3** Topology of dynamic comparator used for optimization

**Table 5.1** Comparator objectives and constraints

| Specification | Objective | Constraint |
|---|---|---|
| Input referred noise ($\sigma_{n,i}$) | Minimize | $\sigma_{n,i} < 300\,\mu\text{V}$ |
| Power ($P$) | Minimize | – |
| Comparison delay ($t_c$) with $V_{DIF} = 0.5\,\text{mV}$ | – | $t_c < 1\,\text{ns}$ |
| Reset delay ($t_r$) | – | $t_r < 1\,\text{ns}$ |

process with 1.2 V supply voltage and constrained the device widths to 0.16–5 µm and lengths to 0.12–1 µm, with discrete steps of 10 nm. The outputs of the comparator under optimization drive 10fF capacitors, which represent realistic loads of some logic gates and routing. The rise and fall times of the clock signal were kept at 100 ps. The optimization takes into account input referred noise, power (given in terms of the spent energy in a complete cycle of comparison and reset), and delays for comparison and reset. The objectives and constraints are summarized in Table 5.1.

## 5.6 Results

We have run the optimization 20 times, with all the parameters set as described in the previous section, in an Intel i7-3770K with 8 GB RAM Linux machine. The average time necessary for a single run is around 27 min (employing eight cores), with a standard deviation of around 12 s. Thus, the average computational time spent for each evaluation is roughly 1.3 s. The Pareto fronts for all the runs are shown in Fig. 5.4, with the $y$-axis demonstrating the energy spent in a comparison (including reset) and the $x$-axis showing the achieved input referred noise. The selection mechanism of the constrained version of the NSGA-II algorithm guarantees that solutions which do not satisfy the constraints do not appear in the Pareto front. Therefore, all the solutions shown in Fig. 5.4 have their comparison and reset delays

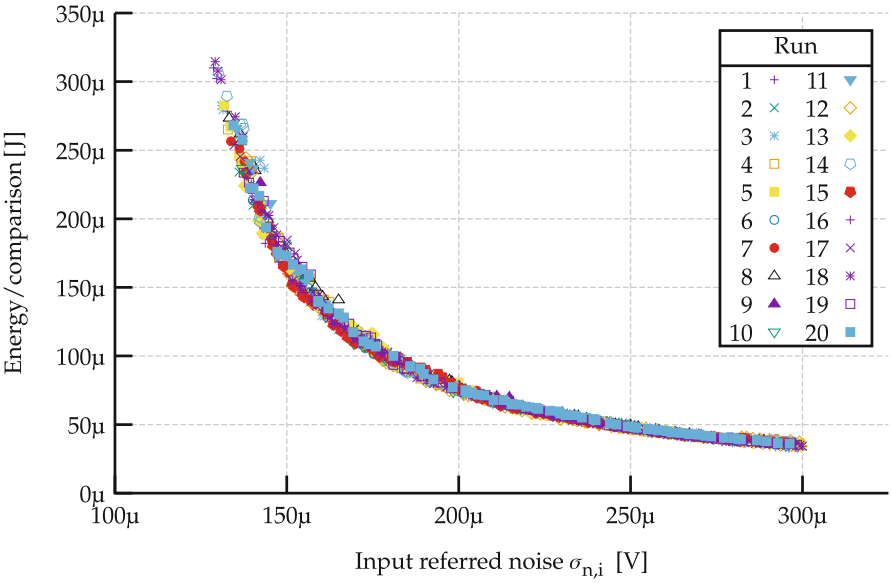



**Fig. 5.4** Pareto fronts achieved in 20 runs of comparator optimization

smaller than 1 ns, and thus we omit these metrics in the plot to improve readability. Moreover, a small set of randomly picked solutions found by the optimizer is shown in Table 5.2.

It is noteworthy that for all the solutions, even those that favor a smaller power consumption at the expense of a larger input referred noise, the transistors of the input pair present relatively large sizes. This observation matches the intuition that the input-pair is the critical part of the circuit when noise is concerned. The input transistors present a large aspect ratio, which consequentially increases the transconductance and decreases thermal noise. The optimizer found solutions with short but nonminimal lengths for the input-pair transistors, which may be related to the dependence of white noise gamma factor $\gamma$ to the channel length [9].

Moreover, the results in Table 5.2 reveal aspect ratios for the tail transistor $W_{\text{tail}}$ that are very small when compared to the aspect ratios of the other transistors, but are still sufficiently large to enable the comparison to be completed during the specified time window. Interestingly, this matches with the design guidelines presented in [3], which show that the input-referred noise is inversely proportional to $\rho$ and $\phi$, given in (5.6) and (5.7), respectively.

$$\rho = \frac{\beta_{\text{in}}}{\beta_{\text{tail}}} = \frac{W_{\text{in}}/L_{\text{in}}}{W_{\text{tail}}/L_{\text{tail}}} \tag{5.6}$$

$$\phi = \frac{\beta_{\text{ln}}}{\beta_{\text{tail}}} = \frac{W_{\text{ln}}/L_{\text{ln}}}{W_{\text{tail}}/L_{\text{tail}}}. \tag{5.7}$$

**Table 5.2** Example comparators sized by the optimizer ($W$'s and $L$'s given in μm)

| Run | $W_{\text{tail}}$ | $L_{\text{tail}}$ | $W_{\text{in}}$ | $L_{\text{in}}$ | $W_{\text{lp}}$ | $L_{\text{lp}}$ | $W_{\text{ln}}$ | $L_{\text{ln}}$ | $W_{\text{rst}}$ | $L_{\text{rst}}$ | $\sigma_{n,i}$(μV) | $E$ (fJ) | $t_c$ (ns) | $t_r$ (ns) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.84 | 0.65 | 4.85 | 0.24 | 4.84 | 0.92 | 3.85 | 0.12 | 4.19 | 0.93 | 134.43 | 268.18 | 0.99 | 0.20 |
| 1 | 1.02 | 0.42 | 4.89 | 0.23 | 3.13 | 0.50 | 3.81 | 0.12 | 3.15 | 0.95 | 149.92 | 172.93 | 0.79 | 0.16 |
| 2 | 0.75 | 0.45 | 4.81 | 0.19 | 0.24 | 0.43 | 2.43 | 0.12 | 4.11 | 0.95 | 162.07 | 134.54 | 0.88 | 0.09 |
| 3 | 0.52 | 0.42 | 4.64 | 0.24 | 0.23 | 0.42 | 2.58 | 0.12 | 2.90 | 0.93 | 175.84 | 106.79 | 0.90 | 0.10 |
| 4 | 0.31 | 0.42 | 4.82 | 0.18 | 0.45 | 0.43 | 3.94 | 0.12 | 1.17 | 0.97 | 192.64 | 82.93 | 0.98 | 0.21 |
| 5 | 0.29 | 0.43 | 4.81 | 0.24 | 0.46 | 0.33 | 0.59 | 0.12 | 1.45 | 0.95 | 213.05 | 67.81 | 0.93 | 0.13 |
| 6 | 0.29 | 0.44 | 3.09 | 0.22 | 0.44 | 0.34 | 1.10 | 0.12 | 1.27 | 0.95 | 229.14 | 59.38 | 0.89 | 0.13 |
| 7 | 0.29 | 0.64 | 4.45 | 0.18 | 0.44 | 0.42 | 0.55 | 0.12 | 0.84 | 0.95 | 239.84 | 53.25 | 0.99 | 0.18 |
| 8 | 0.31 | 0.66 | 3.08 | 0.19 | 0.23 | 0.35 | 1.00 | 0.12 | 0.76 | 0.99 | 256.41 | 46.42 | 0.91 | 0.18 |
| 9 | 0.31 | 0.66 | 3.08 | 0.18 | 0.23 | 0.35 | 0.99 | 0.12 | 0.60 | 0.99 | 269.31 | 42.90 | 0.81 | 0.21 |
| 10 | 0.29 | 0.66 | 3.11 | 0.24 | 0.28 | 0.42 | 0.59 | 0.12 | 0.50 | 0.95 | 286.77 | 38.53 | 0.78 | 0.22 |
| 11 | 0.24 | 0.66 | 3.09 | 0.24 | 0.23 | 0.42 | 0.39 | 0.12 | 0.48 | 0.95 | 296.29 | 36.10 | 0.83 | 0.21 |

To prove that the framework has reached solutions close to optimal, we would need to know the optimal Pareto front. However, if we consider the comparator model as a black-box (and perhaps discontinuous), this is only possible if we employ brute-force evaluation of all the design space. For this problem, this is impractical because the number of parameter combinations is around $1.4 \times 10^{23}$. On the other hand, we can see that the Pareto fronts found in each run are located in the same region in Fig. 5.4, indicating consistency among the runs.

In order to verify the accuracy of the input referred noise simulation method, we compare the results achieved by the PSS/PNOISE method with the noise achieved with transient noise simulations.

For this reference model, we simulate the same comparator (same transistors dimensions) varying the differential voltage at the input, while all the other parameters are preserved unchanged. For each value of input voltage, the outputs of 5000 comparisons within a transient simulation with noise frequency constrained to 50 GHz are stored. Then, the simulated probability of "1"s at the comparator output is plotted as a function of the input voltage. Finally, assuming that the noise is a white Gaussian process, these values may be fitted to the normal CDF, shown in (5.8).

$$\mathrm{CDF}(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]. \tag{5.8}$$

The procedure is depicted in Fig. 5.5. The outcome of the curve fitting is the mean value $\mu$ (deterministic offset) and the standard deviation $\sigma$ (input referred noise) of the comparator. The procedure for each one of the solutions needs approximately 4 h in the same machine, which corresponds to roughly 11,000 times more than the PSS/PNOISE method.

The solutions shown in Table 5.2 were achieved with this approach. The comparison between the results achieved with PSS/PNOISE and the transient noise methods is shown in the bars plot of Fig. 5.6. The maximum difference in the noise measurements is 9.8 %, and the standard deviation of the differences is 3.47 %.

## 5.7 Discussion

In this chapter, we have presented a computational framework for sizing and optimization of clocked voltage comparators. The system minimizes input referred noise and power, with the comparator subject to constraints of maximum delays. Regarding the noise calculation, we have employed a simplified method based on simulation techniques that are commonly employed for RF simulations, namely PSS and PNOISE analyses. The achieved solutions conform with the reference noise model that is based on transient noise simulation. The proposed framework

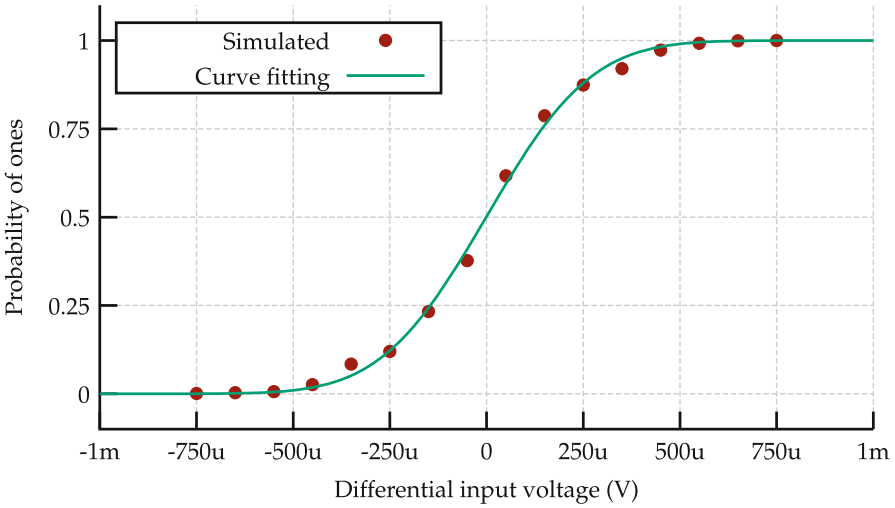



**Fig. 5.5** Simulated probability of "1" with different differential input voltages, and curve fitting to the normal CDF

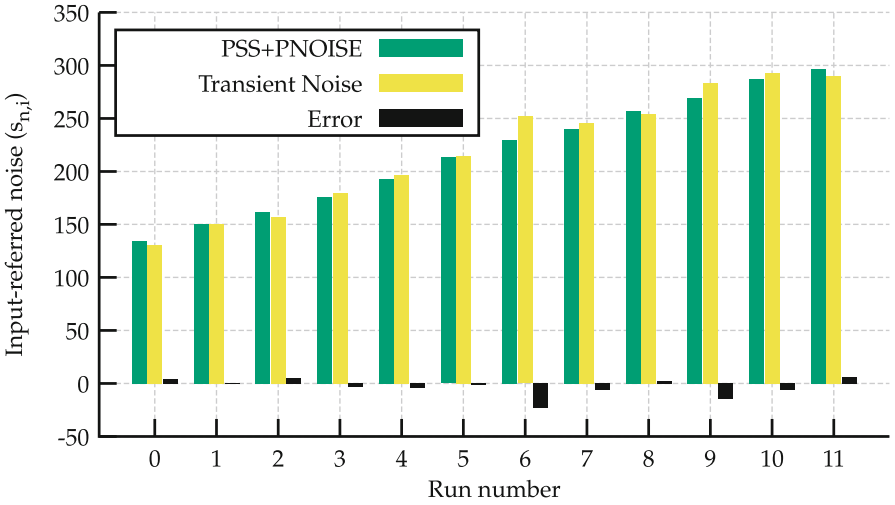



**Fig. 5.6** Results achieved with PSS/PNOISE and transient noise methods

outputs a set of 64 different comparators in the Pareto front that trade-off power and input referred noise, taking about 27 min in a conventional workstation. The characteristics of the proposed optimization framework allow reducing the effort on the design cycle of a comparator circuit drastically.

## References


1. J. Craninckx, G. van der Plas, A 65fJ/conversion-step 0-to-50MS/s 0-to-0.7 mW 9b charge-sharing SAR ADC in 90 nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2007), pp. 246–600. doi:10.1109/ISSCC.2007.373386 (cit. on p. 103)
2. P. Harpe, E. Cantatore, A. van Roermund, A 2.2/2.7fJ/conversion-step 10/12b 40kS/s SAR ADC with data-driven noise reduction, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2013), pp. 270–271. doi:10.1109/ISSCC.2013.6487730 (cit. on p. 103)
3. P. Nuzzo, F. De Bernardinis, P. Terreni, G. Van der Plas, Noise analysis of regenerative comparators for reconfigurable ADC architectures. IEEE Trans. Circuits Syst. Regul. Pap. **55**(6), 1441–1454 (2008). issn: 1549-8328. doi:10.1109/TCSI2008917991 (cit. on pp. 104 and 109)
4. J. Kim, B. Leibowitz, J. Ren, C. Madden, Simulation and analysis of random decision errors in clocked comparators. IEEE Trans. Circuits Syst. Regul. Pap. **56**(8), 1844–1857 (2009). ISSN: 1549-8328. doi:10.1109/TCSI.2009.2028449 (cit. on pp. 104, 105, and 106)
5. K. Kundert, Introduction to RF simulation and its application. IEEE J. Solid State Circuits **34**(9), 1298–1319 (1999). ISSN: 0018-9200. doi:10.1109/4.782091 (cit. on p. 104)
6. A. Hajimiri, T. Lee, A general theory of phase noise in electrical oscillators. IEEE J. Solid State Circuits **33**(2), 179–194 (1998). ISSN: 0018-9200. doi:10.1109/4.658619 (cit. on p. 104)
7. J. Kim, B. Leibowitz, M. Jeeradit, Impulse sensitivity function analysis of periodic circuits, in *IEEE/ACM International Conference on Computer Aided Design (ICCAD)* (2008), pp. 386–391. doi:10.1109/ICCAD.2008.4681602 (cit. on pp. 104 and 105)
8. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). ISSN: 1089-778X. doi:10.1109/4235.996017 (cit. on p. 107)
9. A. Scholten, L. Tiemeijer, R. van Langevelde, R. Havens, A. Zegers-van Duijnhoven, V. Venezia, Noise modeling for RF CMOS circuit simulation. IEEE Trans. Electron Devices **50**(3), 618–632 (2003). ISSN: 0018-9383. doi:10.1109/TED.2003.810480 (cit. on p. 109)

# Chapter 6
# An 8-Bit 0.35-V CS-ADC with Comparator Offset Auto-Zero and Voltage Boosting

**Contents**



## 6.1 Introduction

As discussed in Chap. 4, the comparator offset causes nonlinearity in a CS-ADC. A possible countermeasure is to use larger transistors, improving matching and reducing the offset voltage, but that has the side-effect of increasing the power consumption for the same speed of operation. Foreground calibration of comparator offset is reported as a solution [1], where the faster branch of the differential comparator is slowed down by increasing the capacitance at the critical nodes. However, in WSN and other LVLP applications, besides process variations, the comparator offset is also influenced by changes in temperature, supply voltage and devices aging, and foreground calibration techniques do not cover those. A background calibration technique, on the other hand, accounts for all these variability issues on-line and continuously.

In this chapter, we propose a CS-ADC with a low-power background calibration technique that continuously nullifies the comparator offset voltage and operates



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_6

within the supply-voltage range of the ADC. This method permits the comparator to be implemented with small transistors and burn less power. Furthermore, being a continuous calibration procedure, it automatically compensates changes in PVT. To allow operation with a supply voltage as low as 0.35 V, which is very close to the transistors threshold voltage $V_t$, all the MOS switches in the proposed ADC use local voltage boosting. Thus, a voltage-booster circuit with minimal complexity is proposed, minimizing the impact of this solution to the available area/power budget.

In the next section, the operating principle of the proposed CS-ADC is described, and the background calibration procedure is introduced. Following, in Sect. 6.2.2, we present and explain the self-calibrating comparator circuit. The compact local voltage-boosting circuit is introduced in Sect. 6.3.2. Finally, the measurement results are shown in Sects. 6.4 and 6.5 offers discussions and conclusions.

## 6.2 CS-ADC with Background Comparator Offset Auto-Zeroing

The proposed ADC topology is depicted in Fig. 6.1. The conventional CS-ADC (comprising a TH, a binary-weighted capacitive DAC, the successive approximation logic and a comparator) is complemented with a comparator triggering block, a comparator calibration block, and a pseudo-random binary generator. The calibration circuitry feeds the comparator with the calibration voltage $V_{CAL}$, which modulates its offset voltage. The combination of the calibration circuit and the comparator composes the self-calibrated comparator.

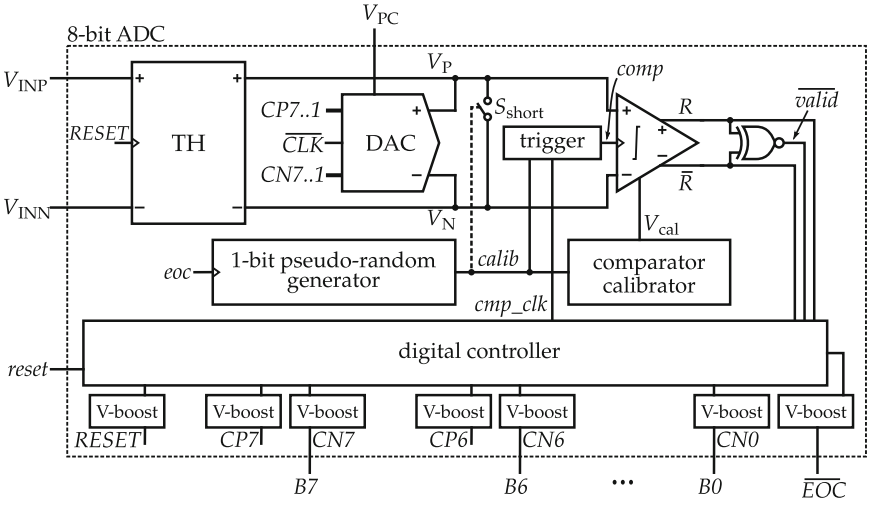



**Fig. 6.1** ADC topology diagram

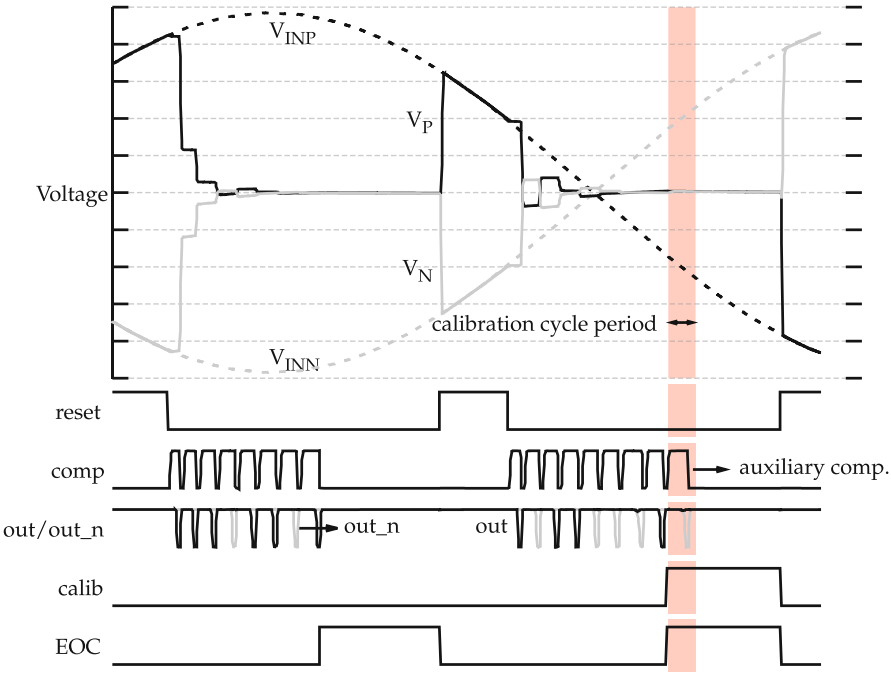



**Fig. 6.2** Illustrative waveforms of the ADC operation

### *6.2.1 Operation Principle of the Proposed CS-ADC*

The operation of the proposed CS-based SAR ADC is exemplified in the waveforms shown in Fig. 6.2. Initially, a conversion is requested when the "reset" signal is pulled to "1." Consequently, the TH tracks the input and the DAC capacitors are precharged to $V_{DD}$. When "reset" returns to "0," the TH is switched to "hold" mode, and the voltage is held on the nodes $V_P$ and $V_N$. Concurrently, the DAC capacitors hold $V_{DD}$. Next, the successive approximation controller initiates the conversion by sending a request to the comparison trigger, which consequently activates the comparator. According to the result of the comparison $k$ ("1" or "0"), the SAR controller adds or subtracts charge in the TH, by closing the corresponding switches in the DAC. This same process is repeated for all the bits and, as $V_{PN}$ reduces towards zero, the controller stores the comparison results. These comparison results will be used to generate the ADC output. When the process is finished, the end-of-conversion is flagged at the "EOC" output pin. Additionally, at the end of a conversion, the voltage $V_{PN}$ naturally converges to the ADC input common-mode level, meaning that the differential voltage is close to zero.

### 6.2.2 Background Comparator Offset Self-Zeroing

As depicted in Fig. 6.2, the proposed method for comparator calibration happens only after the ADC output is already available (EOC = "1"), and takes place only at random cycles, according to the output of the random binary generator. If the random output is a "1," the switch $S_{short}$ shorts $V_P$ and $V_N$, redistributing the residual charge. Then, an auxiliary comparison is requested from the comparator trigger. Since $V_{PN} = 0$, the comparator detects if the offset is positive or negative. The comparator feeds the calibration block with the comparison result, and the latter increases or reduces $V_{CAL}$ by a small step $\Delta V_{CAL}$. The combination of the comparator and calibration block comprises a negative feedback loop, and the offset converges towards zero after a sufficient number of cycles. The process may be stopped after the comparator output begins to alternate between "0" and "1" during the calibration phase, meaning that the offset has been reduced to the minimum allowed by the circuit. On the other hand, if the process is carried out continuously, the system accommodates for PVT variations on-the-fly.

## 6.3 SAR ADC Implementation

An LVLP 8-bit SAR ADC that uses the proposed techniques is designed in a 0.13 μm CMOS process. The prototype employs a digital self-timed SAR controller, a TH, a DAC, a self-calibrated comparator, and a set of voltage boosters. This remaining of this section describes the implementation details of these circuits.

### 6.3.1 Self-Calibrated Comparator

The comparator proposed in this chapter is more suited to low-voltage supplies, as the offset calibration accelerates the slower branch instead of delaying the faster branch, as done in previous works [1–4]. The $V_t$ of the p-type transistors in the 0.13 μm technology employed in this design is very close to $V_{DD}$ at 0.35 V. The input common-mode voltage used for the ADC operating at this supply is 175 mV. However, the comparison is still guaranteed to occur in less than 125 ns (with zero differential input), due to the positive feedback loop present in the topology. All the transistors were drawn with minimum size, with the exception of the input pair which has 5 μm of width and minimum length. This comparator sizing results in an input-referred noise of 0.6 mV, which is below the estimated quantization noise of 0.78 mV ($\frac{V_{LSB}}{\sqrt{12}}$), and limits the maximum achievable ENOB to approximately 7.7 bits.

The offset calibration technique employs a similar scheme to generate the calibration voltage to the one proposed for a two-stage comparator in [5]. However, the proposed dynamic comparator does not use a static preamplifier or current

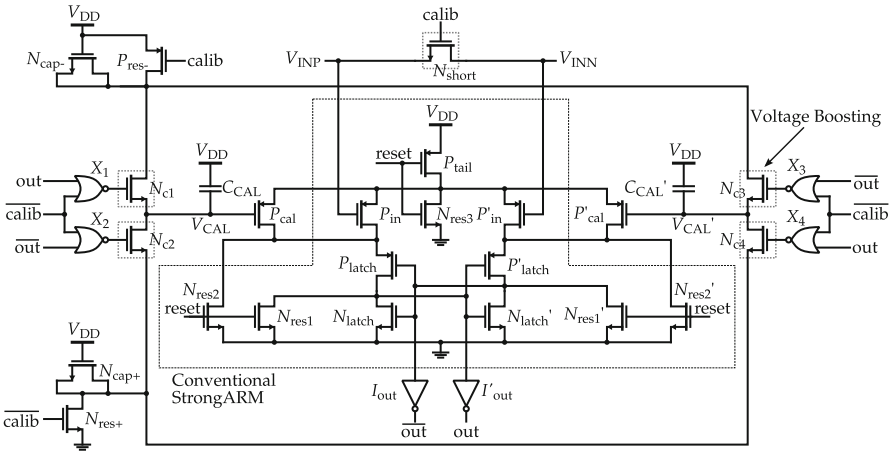



**Fig. 6.3** Comparator with self-calibration circuit

sources, allowing its usage with a lower voltage supply. The circuit is shown in Fig. 6.3, and works as follows. Initially, the calibration capacitors $C_{CAL}$ and $C_{CAL}{}'$ are assumed to be discharged, and $V_{CAL}$ and $V_{CAL}{}'$ are at $V_{DD}$. Therefore, since the transistors $P_{cal}$ and $P_{cal}{}'$ are cut-off, they have a negligible effect on the comparison threshold. Moreover, "calib" is "0," and thus the $p$-MOS $P_{res-}$ and the $n$-MOS $N_{res+}$ are on, discarding the charge accumulated in the MOS-capacitor $N_{cap-}$, and precharging $N_{cap+}$ to $V_{DD}$. When "calib" goes to "1," indicating that a calibration cycle is requested, $N_{short}$ shorts $V_{INP}$ and $V_{INN}$, forcing the differential input of the comparator to zero. After sufficient time for $V_{INP}$ and $V_{INN}$ to settle, a comparison is requested by pulling down the "reset" signal. The circuit behaves as in a conventional latched comparator, and after the comparison is ready, the results "out" and "$\overline{out}$" are propagated to the nor-gates $X_{1-4}$. Since "$\overline{calib}$" is "0," one MOS switch among $N_{c1}$ and $N_{c2}$ and one MOS switch among $N_{c3}$ and $N_{c4}$ are closed, depending on the comparison result. Finally, part of the charge stored in $N_{cap+}$ is redistributed to $C_{CAL}$ or $C_{CAL}{}'$ and part of the charge stored in $C_{CAL}$ or $C_{CAL}{}'$ is lost to $N_{cap-}$. The aftereffect of the charge processing is a reduction on the calibration voltage of the slower branch, $V_{CAL}$ or $V_{CAL}{}'$, leading to an increase in the current of $P_{cal}$ or $P_{cal}{}'$. Since the loop is arranged to provide negative feedback, the comparator offset is reduced after the calibration step. Hence, if enough calibration cycles are provided, the calibration current in $P_{cal}$ or $P'_{cal}$ compensates for the mismatch between the two branches.

Besides, the calibration process is not halted after the comparator reaches zero-threshold. The uninterrupted operation allows the circuit to compensate for the drifts on the calibration voltages (e.g., charge loss caused by the leakage currents of the transistors $N_{c1-4}$). Likewise, it mitigates the impact of PVT variations on the comparator offset.

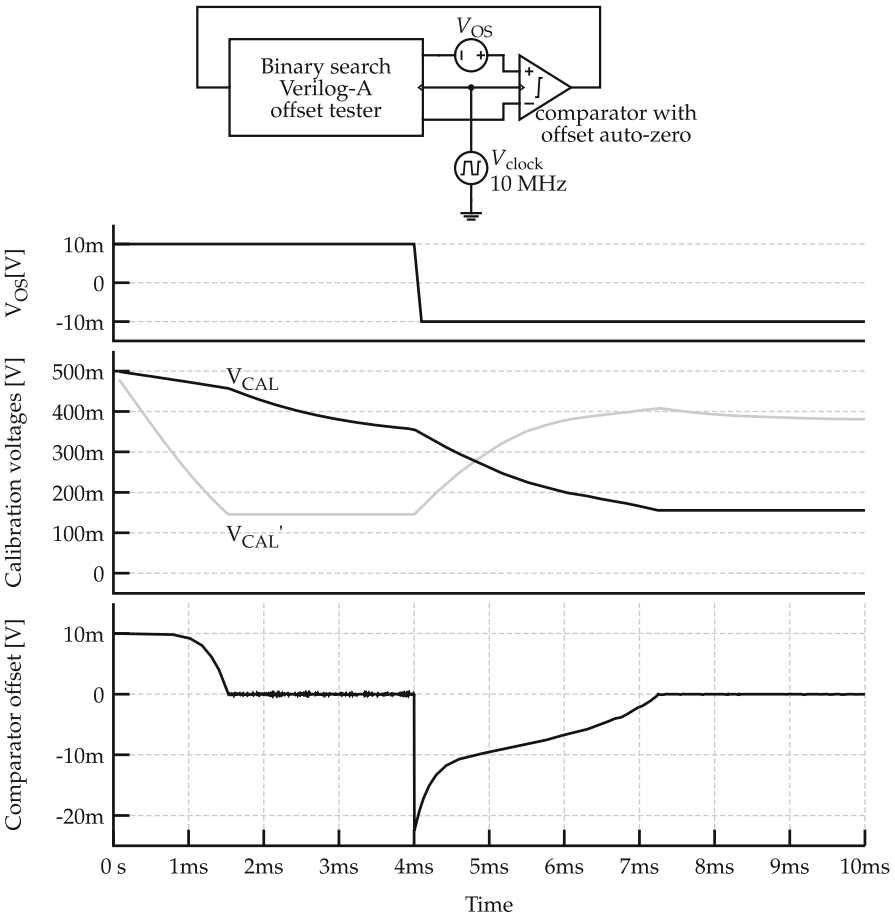



**Fig. 6.4** Test bench for comparator offset simulation and simulated calibration process subject to offset changes

The operation is illustrated by simulation waveforms in Fig. 6.4, that also shows the test bench employed. The measurement of the comparator offset is done by a Verilog-A block that drives the comparator inputs and adjusts them according to the comparator output (the code is given in Listing 6.1). An 18-step binary search is employed to find the actual offset, and then one additional clock cycle is used to compensate the offset. The clock frequency is 10 MHz, which mimics well the comparator operation speed at 0.5 V of supply voltage. The voltage source $V_{OS}$ models the comparator offset voltage and is initially set to 10 mV. At the beginning of the operation, the calibration capacitors are discharged. Thus, $V_{CAL}$ and $V_{CAL}{}'$ are at $V_{DD}$ (0.5 V in the simulation). Since the $V_{GS}$ of the calibration transistors are zero, they do not affect the comparator threshold, which is given only by $V_{OS}$. As the operation continues, the calibration voltages start to compensate for the offset, and

after roughly 1.5 ms (that correspond to less than 800 calibration cycles) the offset is compensated. The offset voltage is inverted to −10 mV at 4 ms, to verify the stability and the ability of the self-calibration method to recover from large offset variations. The comparator is then able to nullify the offset in around 1700 calibration cycles. After complete calibration of the comparator, the offset voltage RMS value, i.e. the calibration noise, is below 50 µV, which is much lower than the value of LSB (∼2.7 mV). Moreover, the absence of overshoot or ringing in the response to the instantaneous offset variation is a good indication that the feedback loop is stable.

**Listing 6.1** Verilog-A code for the offset tester block, offset_tester.va

```
1  `include "constants.vams"
2  `include "disciplines.vams"
3
4  module offset_tester(compinp, compinn, clk, compout);
5     // Inputs/outputs
6     output            compinp, compinn; // Output signals to be
          ↪ connected to the comparator inputs.
7     input       clk, compout; // Input signals: clock and the
          ↪ comparator output.
8     electrical        compinp, compinn, clk, compout;
9     // Parameters
10    parameter real    vcm = 0.6;  // Common-mode voltage
11    parameter real    os_nominal = 0.0; // Deterministic
          ↪ offset. Initial guess of offset voltage.
12    parameter real    clk_threshold   = 0.25; // Threshold
          ↪ voltage for the clock.
13    parameter real    out_threshold   =0.25; // Threshold
          ↪ voltage for the output.
14    parameter real    os_maximum = 0.1; // Predicted maximum
          ↪ offset magnitude. Dictates the search range.
15    parameter integer steps = 15; // Steps in the binary search
          ↪ algorithm.
16    parameter string  filename = "comp_offset.csv"; // Name of
          ↪ the output file with the found offset voltage.
17    parameter string  filemode = "a"; // How to open the output
          ↪ file. "w" for Write, "a" for Append.
18    // Variables
19    real        vdif; ; // Variable to store the output value
          ↪ of differential voltage.
20    integer       i; // Actual cycle on the binary search
          ↪ algorithm.
21    integer       file; // File descriptor.
22
23 analog
24   begin
25      // At the beginning of the simulation.
26      @(initial_step)
27        begin
28     // Open file.
29     file = $fopen(filename, filemode);
30     // Start with cycle 0 in the BS algorithm.
31     i=0;
```

```
32      //Set initial guess of offset voltage.
33      vdif=os_nominal;
34         end
35       // At every falling-edge of the clock.
36       @(cross( (V(clk) - clk_threshold) , -1))
37         // Check if the BS finish.
38         if(i==steps)
39     begin
40        // Write offset value to the output file.
41        $fwrite(file, "%f\n",V(compinp,compinn));
42        // Close file.
43              $fclose(file);
44     end
45         else
46     begin
47        // If comparator output is "1"
48        if(V(compout)>out_threshold)
49          begin
50       // Reduce differential voltage at the comparator inputs.
51       vdif=vdif-os_maximum/pow(2,i);
52          end
53        else
54          begin
55       // Increase differential voltage at the comparator inputs.
56       vdif=vdif+os_maximum/pow(2,i);
57          end
58        // Move to the next cycle in the BS algorithm.
59        i=i+1;
60     end // else: !if(i==steps)
61       // Assign the value of the variables to the output
             ↪ voltages.
62       V(compinp) <+ transition(vcm+vdif/2, 0, 10p, 10p);
63       V(compinn) <+ transition(vcm-vdif/2, 0, 10p, 10p);
64   end
65 endmodule
```

Additionally, we see in Fig. 6.4 that the calibration voltages are still slowly changing after the comparator finishes compensating the offset. The reason is that the time constant relative to the common-mode component of the calibration voltage (i.e., caused by the charge injected into $C_{CAL}$ and $C_{CAL}{}'$ by the parasitic capacitances of $N_{c1-4}$) is much larger than the time constant relative to the differential calibration voltage. Therefore, the common-mode voltage of $V_{CAL}$ and $V_{CAL}{}'$ requires a longer time to settle completely, but this has no impact on the offset. For the same reason, and also because of the leakage current of transistors $N_{c1-4}$, at the beginning of the calibration process $V_{CAL}$ is decreasing, even though all the comparison results are "0"s.

We performed an 1800-run Monte-Carlo simulation, to estimate the comparator offset before and after calibration. This is shown in Fig. 6.5, for calibration limited to 1500 cycles.

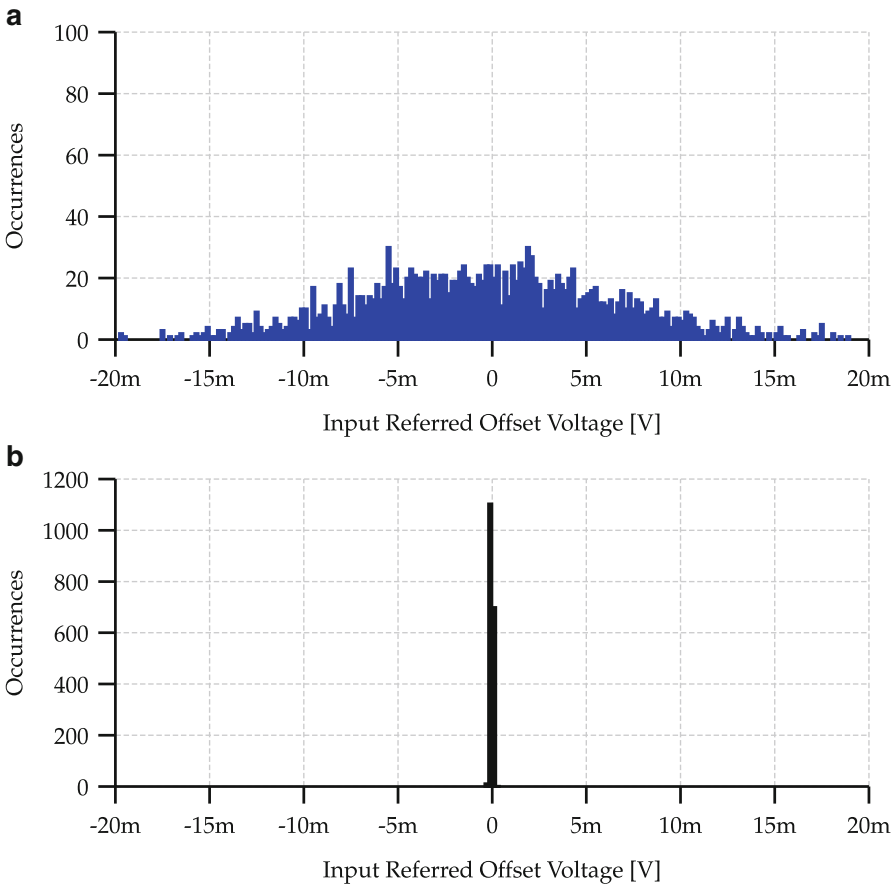



**Fig. 6.5** Histogram of input referred comparator offset voltages: (**a**) before and (**b**) after calibration

If the calibration cycles happen periodically, i.e. they are equidistant on the time axis, the calibration noise appears as a sub-harmonic of the sampling frequency. We mitigate this artifact by calibrating the comparator in random cycles. A simple approach to mimic the operation of a pseudo-random binary generator is to AND-function some ADC outputs, as shown in Fig. 6.6. Assuming that the input signal is random, the probability of "1"s for the implementation with a 3-input AND-gate is 12.5 %, and a calibration cycle is likely to happen once every eight ADC cycles. Noise and jitter guarantee that the calibration noise power is spread across the spectrum. The output codes where the calibration happens are shown in Fig. 6.6.

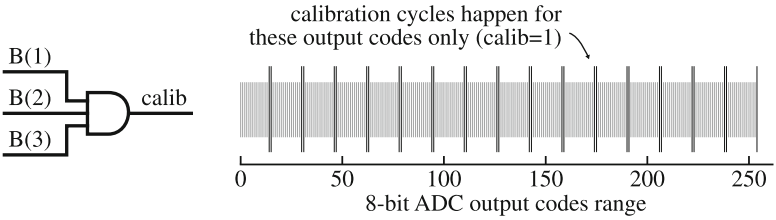



**Fig. 6.6** Pseudo-random calibration cycles with an AND-gate

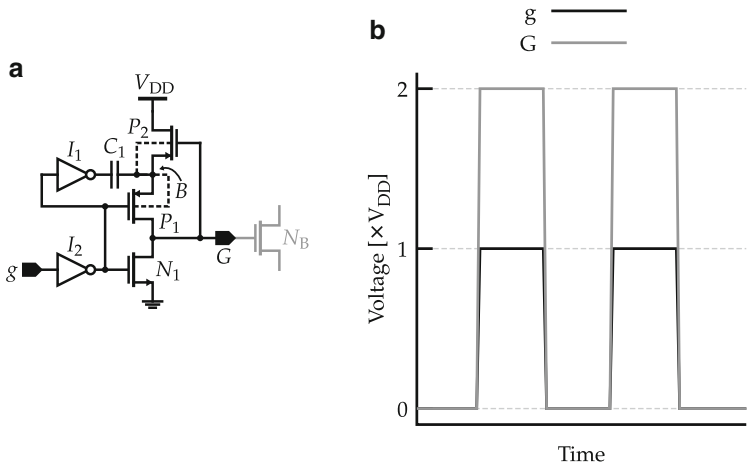



**Fig. 6.7** Boosting switches employed in the ADC: (**a**) schematic and (**b**) illustrative waveforms

### *6.3.2 Local Voltage Boosting*

The ADC operation at very-low voltages is enabled by the use of a VB circuit that doubles the gate voltage for all the MOS switches inside the TH, the DAC, and the comparator calibration circuit. Some of the switches are controlled by the same signal, and, therefore, share a single VB. This is the case for the DAC reset and TH switches, and the two switches in the DAC that connect a given capacitor in parallel and the two that connect in anti-parallel. Therefore, for this particular 8-bit ADC, 24 VBs are required, including those used by the comparator self-calibration scheme. The use of this technique prevents the need for low-$V_t$ transistors, avoiding the increase of leakage current.

The employment of local VB elements is only feasible if the VB circuit fits in a small area and is energy efficient. For this reason, a circuit similar to the one proposed in [6] is employed. The circuit is shown in Fig. 6.7a. As can be seen in Fig. 6.7b, the output signal has twice the magnitude of $V_{DD}$. Compared to other solutions, this circuit presents a reduced devices-count, corresponding to a single capacitor and five small-sized transistors (inverter $I_2$ can be omitted if the input

is already inverted). When the input is low, the output of the inverter $I_1$ connects the bottom plate of $C_1$ to ground, while the node $G$ is also shorted to ground by the transistor $N_1$. This turns $P_2$ on and precharges $C_1$ to $V_{DD}$. When "in" goes to a logic "1," the switch $P_1$ connects the nodes $G$ and $B$, while $I_1$ pulls the bottom plate of $C_1$ to $V_{DD}$, thus forcing node $G$ to $2V_{DD}$. This voltage is maintained since $G$ is a high-impedance node. The circuit has no static power consumption, and the required dynamic power is only to tie the gates of the transistors to the supply rails and to restore part of the charge held in capacitor $C_1$ that is transferred to the gate of $N_B$ while setting it to $2V_{DD}$.

The ADC safely operates with a maximum supply of 0.6 V, which is half the nominal technology supply voltage. Higher $V_{DD}$ may be applied, but reliability would become a concern due to the long-term stress of the MOSFETs gate oxide.

### 6.3.3 Full-Custom Self-Timed SAR Controller

The SAR controller is responsible for carrying out the binary search algorithm and synchronizing the internal circuits. A self-timed controller scheme is proposed, and the only clock signal required is the sampling clock, while the comparator clock is automatically generated on-chip. The binary search controller is shown in Fig. 6.8.

The binary search circuit is designed in a full-custom fashion, to improve power and area efficiency [7]. The 8-bit controller uses 8 instances ($s8, s7 \ldots s0$) of the logic circuit shown in Fig. 6.9, which is based on a TSPC flip-flop. The arrangement of transistors $N_1$–$N_6$ and $P_1$–$P_4$ forms a TSPC latch with modified pull-up $P_1$–$P_2$ and pull-down $N_2$–$N_5$ combinatorial structures. Also, the combination of paralleled transistors $N_2$ and $N_3$ is added in the pull-down network to detect comparison completion, allowing to discharge $\overline{next\_prep}$ exclusively after $cp$ or $cn$ are set. This latch section feeds a differential regenerative stage ($N_7$–$N_{10}$ and $P_5$–$P_{10}$) that processes and keeps the comparison result, generating the $cp$ and $cn$ control signals for every bit. These signals are responsible for connecting the DAC capacitances in parallel ($cp$) or anti-parallel ($cn$). This section of the circuit is triggered by the $\overline{valid}$ falling edge, implying a successful comparison. The $R$ and $\overline{R}$ signals come from the

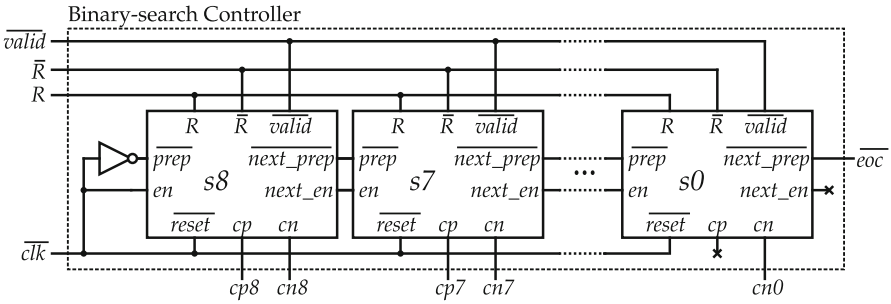



**Fig. 6.8** Schematic of the binary search logic controller

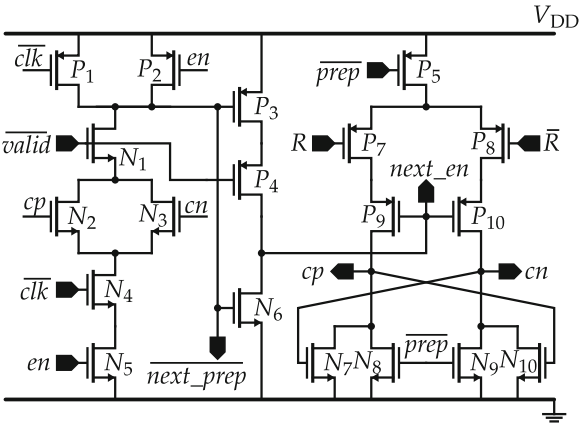



**Fig. 6.9** Schematic of one slice of the binary search logic controller

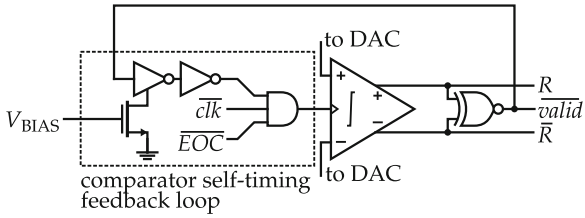



**Fig. 6.10** Schematic of the comparator self-timing feedback loop

comparator, indicating the result of the current comparison. The positive feedback provided in *cp* and *cn* nodes (transistors $N_8$ and $N_9$) avoids fluctuations on these nodes, that would exist because of the high impedance of the nodes in the following comparisons. The feedback arrangement does not drain static current. Every slice also generates the required signals responsible for triggering the subsequent slices.

The comparator self-timing feedback loop is shown in Fig. 6.10 and uses very simple combinational logic and a controllable delay. The $\overline{valid}$ signal is generated from the comparator outputs with an XNOR gate and feeds a pair of inverters that have their propagation delay controlled by $V_{BIAS}$. The 3-input AND-gate prevents the comparator to be triggered after the EOC and triggers the first comparison after the rising edge of the ADC clock ($\overline{clk}$).

### 6.3.4 TH and DAC

In this design, the DAC sizing was mismatch-limited, and is dimensioned to fulfill $\sigma_{DNL} < 0.1$ LSB. The TH capacitors have 1.73 pF each, and charge partitioning in the precharge cycle is employed for the 3 LSB capacitors (see Fig. 3.10d). Therefore, $C_7 \ldots C_4$ have 436, 218, 109, and 54.5 fF, respectively, while $C_3 \ldots C_1$ have 27.25 fF.

## 6.4 Experimental Results

The circuit layout and the chip micrograph of the ADC fabricated in a 0.13 µm CMOS process are shown in Figs. 6.11 and 6.12. The core circuit occupies an area of 0.0377 $mm^2$. The comparator calibration circuit and the voltage boosters correspond to approximately 10 % and 21 % of the overall ADC area, respectively. Figure 6.13 shows the measured DNL and INL for the ADC with comparator self-calibration "on" and "off" for 0.35 V supply. Table 6.1 summarizes the impact of the comparator self-calibration on the ADC performance. The peak values and dispersion ($\sigma$) of the DNL and INL are decreased, and consequently the ENOB is increased from 5.65 to 6.48. The FFT plot of the calibrated ADC is shown in Fig. 6.14 for a near-Nyquist frequency input also at 0.35 V of supply voltage. The achieved SFDR, THD, SNR and SNDR are 45.6 dB, −43.11 dB, 42.81 dB, and 39.94 dB, respectively. Total power dissipation is 84.7 nW at 0.35 V, which leads to a FOM (FOM= $\frac{P}{(f_S 2^{\mathrm{ENOB}})}$) of 5.04 fJ/conversion-step. The comparator self-calibration contributes with 5.5 % of the power consumption and, since it requires another

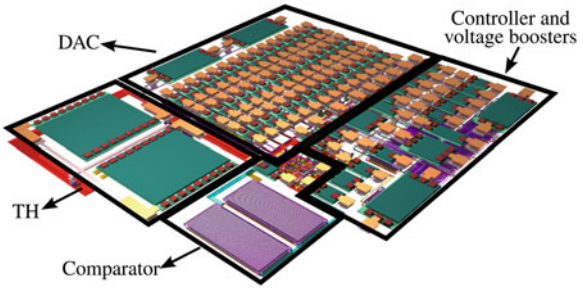



**Fig. 6.11** 8-bit SAR ADC layout

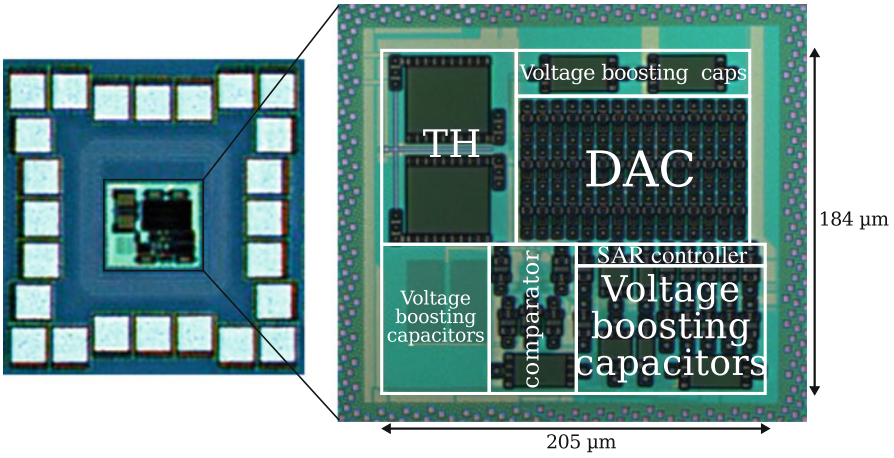



**Fig. 6.12** 8-bit SAR ADC chip micrograph

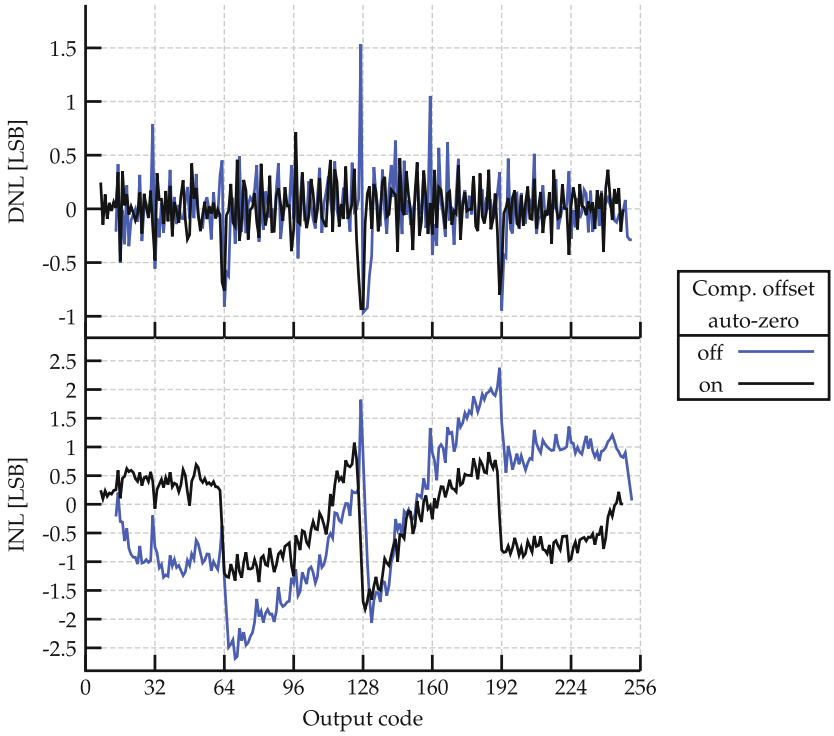



**Fig. 6.13** Measured DNL and INL

**Table 6.1** Improvements on the ADC performance (DNL and INL values in LSB)

| Calib. | Peak DNL | $\sigma_{DNL}$ | Peak INL | $\sigma_{INL}$ | ENOB |
|---|---|---|---|---|---|
| Off | −0.97/1.54 | 0.29 | −2.68/2.37 | 1.23 | 5.65 |
| On | −0.92/0.71 | 0.23 | −1.82/1.07 | 0.65 | 6.48 |

comparison with a small value on the comparator input, the ADC requires up to 11 % more time for a complete conversion.

The test setup for the measurements is shown in Fig. 6.15. The power consumption, ENOB, and corresponding FOM achieved by the ADC are plotted in Fig. 6.16 for different supply voltages and operating frequencies. The plot shows a degradation in the FOM for lower sampling rates. This occurs because the leakage

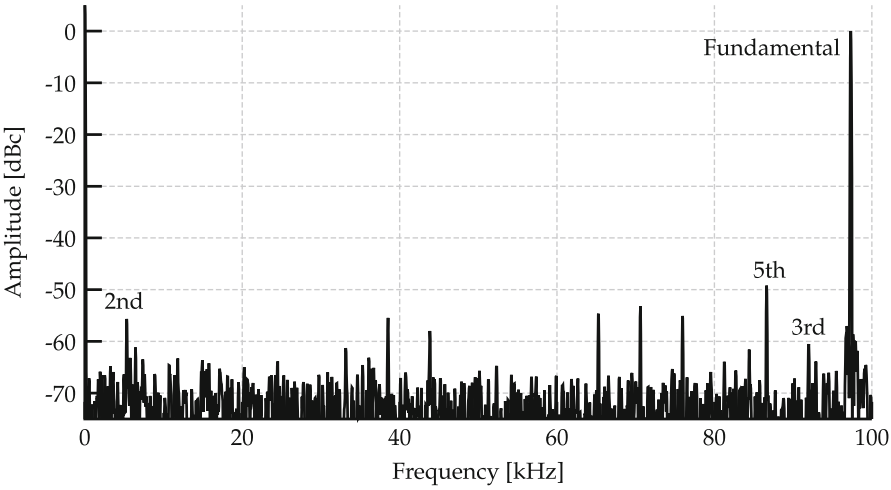



**Fig. 6.14** Measured spectrum (input near Nyquist frequency)

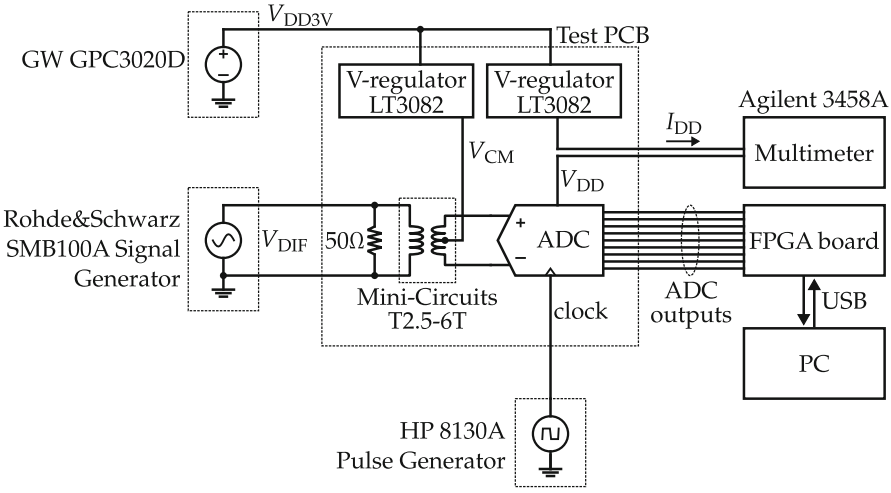



**Fig. 6.15** Test circuit for the measurements

current is approximately constant for a given supply voltage and independent of the sampling rate. In consequence, the leakage dominates the overall ADC power consumption at sufficiently low speeds.

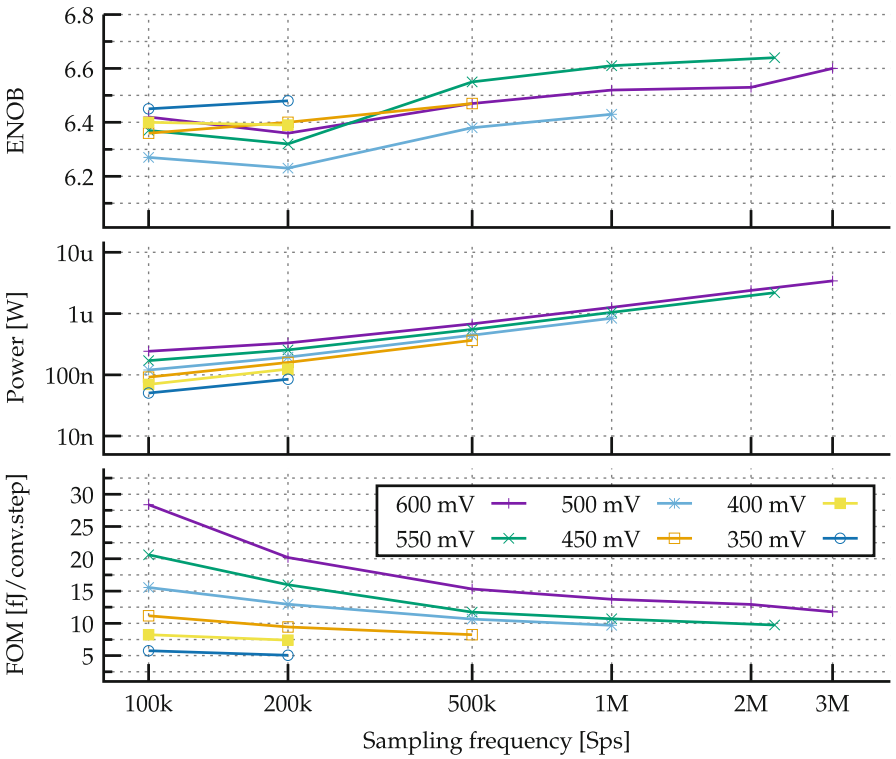



**Fig. 6.16** Power consumption and FOM achieved for different supply voltages

## 6.5 Discussion

This chapter discussed the implementation of a SAR ADC based on a CS-DAC. A novel background self-calibration method was proposed to mitigate the effect of comparator offset on the overall ADC linearity. Differently from previous implementations, this technique can work under very-low supply voltages, not demanding any preamplifier, thus requiring no static power consumption. We have also introduced the use of local voltage boosting for all the switches in the TH, the DAC, and the comparator calibration circuit. Therefore, a compact and energy efficient circuit has also been used. The presented ADC can operate at a voltage supply of 0.35 V while still providing 200 kSps.

The operation was limited to a minimum supply voltage of 0.35 V, mainly because we have employed a custom controller (described in [8]) that is very energy-efficient, but uses the parasitic capacitances to keep the digital values used during the conversion. At lower supply voltages, the controller operation is excessively slow, so the bits stored on the parasitics are lost due to the leakage of the controller

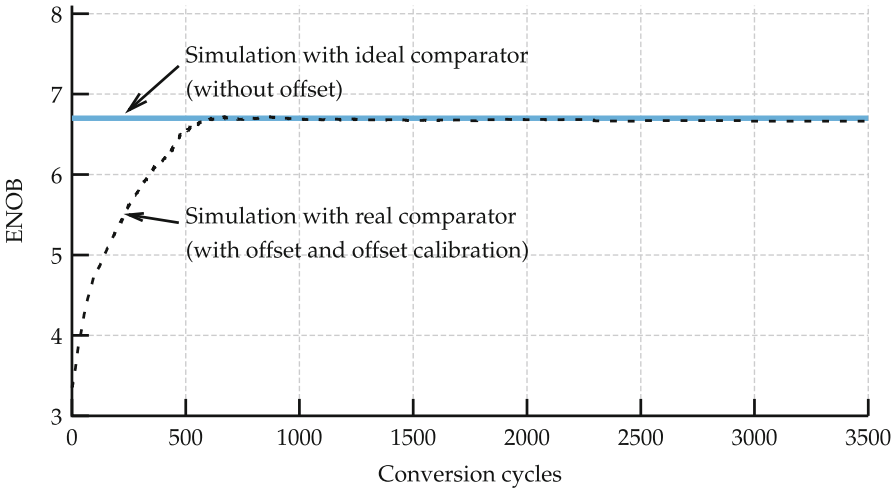



**Fig. 6.17** Impact of offset calibration on ENOB in a post-layout simulation

transistors. Within the $V_{DD}$ range that we were able to test, the leakage currents of the other circuits, such as the DAC and the TH, do not noticeably affect the linearity of the ADC.

To validate the effectiveness of the comparator offset calibration technique, we show in Fig. 6.17 the result of two post-layout simulations: with an ideal zero-offset comparator and with a pessimistic 20 mV comparator offset. We see that the ENOB of the ADC with comparator offset converges to the same ENOB achieved without comparator offset as the offset calibration occurs. It is to be noted a significant drop of 1.5-bit on the ENOB, both in post-layout simulation and measurement, which was verified to have no relationship with the comparator offset self-calibration. A careful debug process showed that this was caused by an unbalance of the parasitic capacitors in the DAC, affecting the linearity of the ADC and causing the missing codes in the static characteristic. Post-layout simulations on a re-arranged layout revealed an ENOB curve with a similar behavior to that shown in Fig. 6.17 but asymptotic to 8 bits, confirming the effectiveness of the offset self-calibration technique.

## References

1. J. Craninckx, G. van der Plas, A 65fJ/conversion-step 0-to-50MS/s 0-to-0.7mW 9b charge-sharing SAR ADC in 90 nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2007), pp. 246–600. doi:10.1109/ISSCC.2007.373386 (cit. on pp. 115 and 118)
2. V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Van der Plas, J. Craninckx, An 82μw 9b 40MS/s noise-tolerant dynamic-SAR ADC in 90 nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2008), pp. 238–610. doi:10.1109/ISSCC.2008.4523145

3. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70dB DR 10b 0-to-80MS/s current-integrating SAR ADC with adaptive dynamic range, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2012), pp. 470–472. doi:1.1109/ISSCC.2012.6177095
4. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70 dB DR 10b 0-to-80 MS/s current-integrating SAR ADC with adaptive dynamic range. IEEE J. Solid State Circuits **49**(5), 1173–1183 (2014). doi:10.1109/JSSC.2014.2309086 (cit. on p. 118)
5. P. Figueiredo, P. Cardoso, A. Lopes, C. Fachada, N. Hamanishi, K. Tanabe, J. Vital, A 90 nm CMOS 1.2V 6b 1GS/s two-step subranging ADC, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2006), pp. 2320–2321. doi:10.1109/ISSCC.2006.1696295 (cit. on p. 118)
6. D. Senderowicz, G. Nicollini, S. Pernici, A. Nagari, P. Confalonieri, C. Dallavalle, Low-voltage double-sampled $\Sigma\Delta$ converters. IEEE J. Solid State Circuits **32**(12), 1907–1919 (1997). doi:10.1109/4.643649 (cit. on p. 124)
7. T. Rabuske, F. Rabuske, C. Rodrigues, A novel energy efficient digital controller for charge sharing successive approximation ADC, in *IEEE Latin American Symposium on Circuits and Systems (LASCAS)* (2011). doi:10.1109/LASCAS.2011.5750295 (cit. on p. 125)
8. T. Rabuske, C. Rodrigues, S. Nooshabadi, An energy-efficient 1 MSps 7 μW 11.9 fJ/conversion-step 7pJ/sample 10-bit SAR ADC in 90 nm, in *IEEE International Symposium of Circuits and Systems (ISCAS)* (IEEE, New York, 2011), pp. 261–264. doi:10.1109/ISCAS.2011.5937551 (cit. on p. 130)

# Chapter 7
# A 9-Bit 0.6-V CS-ADC with a MOSCAP-DAC

**Contents**



## 7.1 Introduction

The DAC in CR-based SAR ADCs requires bias-independent capacitors, i.e. capacitors with constant capacitance for any voltage across its terminals. This characteristic requires that the CR-ADC uses integrated capacitors with very linear capacitance as function of bias voltage ($C(V)$), such as metal-insulator-metal (MIM)[1] or metal-oxide-metal (MOM)[2] capacitors. On the other hand, the CS-based

[1]The MIM capacitors are generally fabricated on top of the routing metal layers and use a thin layer of insulator, which requires additional fabrication steps and increases the fabrication costs. The density of the MIM capacitor is a direct function of the dielectric constant ($\kappa$) of the insulating material. However, for most of the high-$\kappa$ oxides, it is observed that the capacitance increases with the voltage [2, 3], bringing a clear trade-off between linearity and density that limits the performance of MIM capacitors.

[2]The MOM capacitors come for free in digital CMOS processes and are made with stacked wires of routing metals that perform as plates, separated by the BEOL oxide. This type of capacitor clearly benefits from the advances in lithography, that permit metal wires to be drawn closer together.



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_7

ADC permits nonlinear capacitors to be used in its construction [1]. However, as discussed in Chap. 4, the CS-ADC is more sensitive to noise and comparator offset, limiting the competitiveness of the architecture.

In the SAR ADC design presented in this chapter, these two issues are addressed. The tolerances to noise and comparator offset are largely improved by exploiting the very nonlinear $C(V)$ characteristic of MOSCAPs,[3] that are used in this design as the DAC elements. The use of MOSCAPs brings other improvements to the ADC: MOSCAPs come for free in CMOS processes (as do MOMs, but not MIMs), potentially decreasing fabrication costs; MOSCAPs present exceptional matching characteristics; MOSCAPs use the thinnest available insulator in an integrated circuit, yielding in the largest value of capacitance per area among the integrated capacitors, with potential to reduce the DAC area. In the 0.13-μm process employed in the design described in this chapter and a modern 28-nm process, for instance, the MOSCAPs present capacitance densities that are approximately 5× and 4× higher than MOMs, respectively. Furthermore, the charge-mode operation of CS-ADCs, the adoption of a sampling frontend based on a new boost-and-bootstrap switch circuit, and the use of local voltage boosting in the DAC enable the prototype to operate with increased values of peak-to-peak differential IR. The extended IR reaches 1.7 $V_{P-P}$ when the ADC is supplied with 0.6 V, and 1.6 $V_{P-P}$ when the ADC is supplied with 0.4 V.

The 9-bit prototype ADC is fabricated in a 0.13 μm CMOS technology and performs with an ENOB larger than 8.5 bits for temperatures ranging from −40 to 85 °C. At 1 MSps, the ADC burns 2.78 μW, yielding in a FOM of 7.8 fJ/conv.step.

The remainder of this chapter is organized as follows. In Sect. 7.2, the principle of operation of the proposed MOSCAP-based CS-ADC is introduced and analyzed. Section 7.3 describes the implementation of the circuits employed in the prototype ADC. Finally, Sect. 7.4 presents the experimental results and Sect. 7.5 discusses the achievements of this work.

## 7.2 CS-ADC with a MOSCAP-Based DAC

If the DAC capacitors are bias-dependent, (4.21), derived in Chap. 4, can be rewritten as (7.1).

However, in order to minimize the routing parasitics in modern processes, CMOS scaling demands the use of low-$\kappa$ BEOL dielectrics, which ultimately hinders the capacitance density of MOM capacitors.

[3]The MOSCAP is implemented using a MOS transistor with drain and source shorted, and presents the highest capacitance density among the integrated capacitors. It does not require additional fabrication steps. However, due to its highly nonlinear $C(V)$ characteristic, its use is commonly reserved for applications in which the accuracy of capacitance is not critical, including decoupling and frequency compensation.

$$V_{\mathrm{PN}}[i] = \frac{(V_{\mathrm{INP}} - V_{\mathrm{INN}})\,\frac{C_{\mathrm{TH}}}{2} - \sum_{j=1}^{i}\left((2k_{B-j}-1)\int_0^{V_{\mathrm{PC}}} C_{B-j-1}(V)\mathrm{d}V\right)}{\frac{C_{\mathrm{TH}}}{2} + \sum_{j=1}^{i} C_{B-j-1}(V_{\mathrm{PN}}[i])}. \tag{7.1}$$

In (7.1), while $C_{B-j-1}$ in the denominator depends on the varying voltage $V_{\mathrm{PN}}$, $C_{B-j-1}$ in the numerator only depends on a fixed bias voltage $V_{\mathrm{PC}}$. This particularity may be reasoned as follows: the numerator represents the total charge on the system; the DAC capacitors are always precharged to $V_{\mathrm{PC}}$; after precharge, due to the absence of low-impedance path for the charge to flow off, there is conservation of charge, and the total charge is not affected by the capacitance value. Therefore, the linearity of the CS-ADC, according to (7.1), is not affected by the nonlinearity of the DAC-capacitors. Also, according to the same expression, two important conclusions concerning the DAC capacitors can be drawn:

1. It is advantageous to minimize the capacitance $C_x(V_{\mathrm{PN}}[i])$ of the DAC cells during charge processing, as the ADC performance is harmed by the increase in total capacitance during the conversion, as explained in Chap. 4.
2. It is advantageous to maximize the capacitance $C_x(V_{\mathrm{PC}})$ of the DAC cells during precharge, in order to maximize the density of capacitance and reduce the DAC area.

Fortuitously, these conditions are satisfied with a MOSCAP. The simulated quasi-static [4] $C(V)$ curve of the total gate capacitance (which include $C_{\mathrm{GD}}$, $C_{\mathrm{GS}}$, and $C_{\mathrm{GB}}$), as function of $V_{\mathrm{DIF}}$, $C_{\mathrm{GG}}(V_{\mathrm{DIF}})$, is shown in Fig. 7.1a. Please refer to the Appendix for comments on the validity and applicability of the quasi-static $C(V)$ model to this analysis. The simulated $n$-type MOSCAP is wired identically as in the differential DAC of a CS-ADC (Fig. 7.1b). A common-mode voltage $V_{\mathrm{CM}} = 0.4\,\mathrm{V}$ is used to obtain the plot. The MOSCAP presents a relatively large

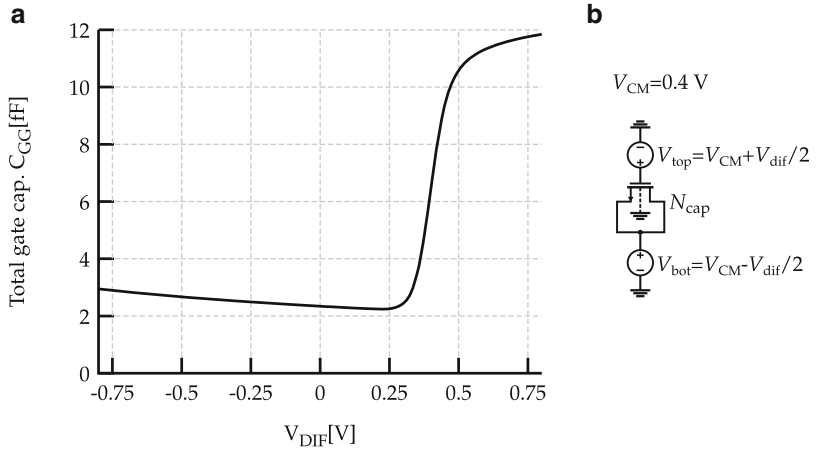



**Fig. 7.1** The MOS capacitor: (**a**) simulated $C(V)$ characteristics and (**b**) circuit used for the simulations

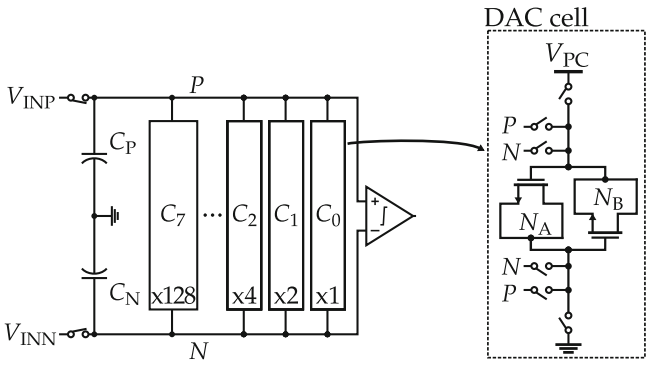



**Fig. 7.2** Proposed MOSCAP-DAC topology

value of capacitance if the differential voltage $V_{DIF}$ is higher than approximately 0.4 V, as the outcome of the inversion layer that establishes in the semiconductor-insulator interface. As $V_{DIF}$ decreases, the MOSCAP enters in the depletion region and the capacitance is decreased considerably. As $V_{DIF}$ is decreased further, the source-to-bulk voltage ($V_{SB}$) increases, causing the threshold voltage $V_t$ of the MOSCAP to raise, ultimately reducing the slope of $C_{GG}$. Thus, the value of $C_{GG}(V_{DIF})$ is nearly constant for $V_{DIF} < 0.3$ V.

The proposed implementation of the DAC capacitance cells based on MOSCAPs is shown in Fig. 7.2. Each unit cell comprehends 2 identical *n*-type MOSCAPs in anti-parallel ($N_A$ and $N_B$), with the purpose of balancing the parasitic capacitances. The bodies of the MOSCAPs are connected to ground. The DAC cells also comprehend switches for precharging the MOSCAPs to $V_{PC}$ and for connection to the TH. Binary weights of charge are achieved by multiple instantiations of these cells. The effect of the varying capacitance of DAC-cells on the ADC can be understood with the help of Fig. 7.3. The figure shows the $C(V)$ curve of the two capacitors employed in a DAC cell, juxtaposing with the dynamic range of $V_{PN}$ according to the cycle in the binary search algorithm. As the residue $V_{PN}$ is forced towards zero during the binary search algorithm, the capacitance of the connected DAC capacitors shrinks accordingly. The voltage range on the comparator begins at full-scale during the first comparison, when no DAC capacitor is yet connected to the TH, and is ideally halved every following comparison. When $V_{PN}$ approaches zero, which is the critical state concerning comparator offset and noise, the MOSCAPs capacitance is at its minimum. This leads to a reduced total capacitance, and minimizes voltage attenuation, improving the ADC tolerance to comparator offset and noise.

Initially, we estimate the linearity and noise of the MOSCAP-based ADC architecture through simulations. First, we designed a complete 9-bit ADC with $\alpha = 1$ ($V_{IR} = 2V_{PC}$) and $V_{PC} = 0.6$ V with a conventional DAC that uses linear capacitors. Afterward, in a copy of this original ADC, we substitute the conventional DAC by a MOSCAP-DAC that has the same total capacitance in the precharge cycle and MOSCAPs with the $C(V)$ curves of Fig. 7.1. Then, we simulate the ADCs in

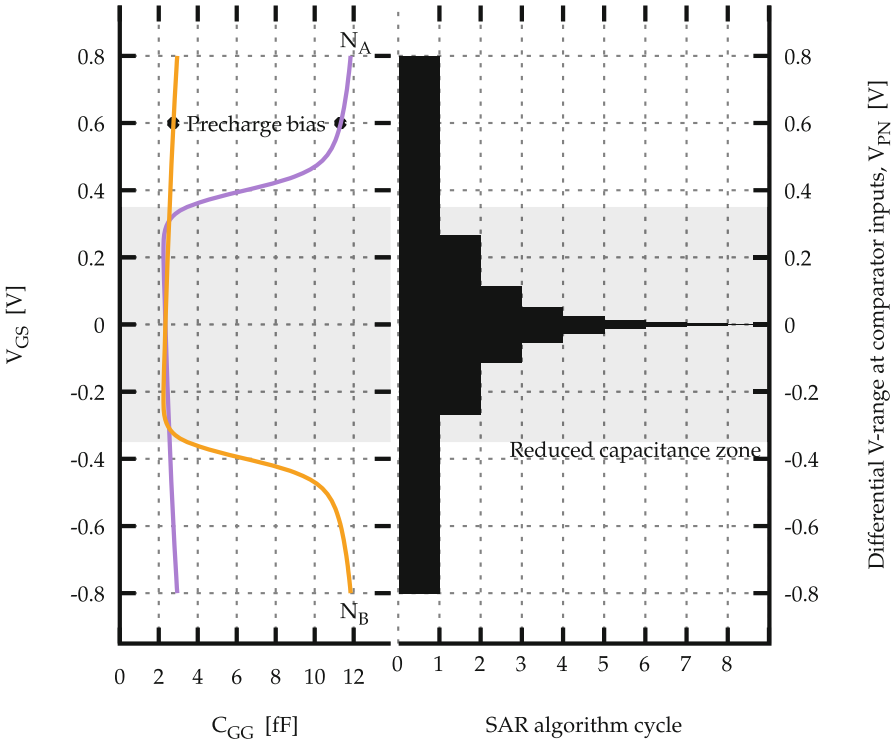



**Fig. 7.3** Trajectory of the MOSCAP capacitance during the binary search algorithm

schematic-level with a full-swing sinusoidal input. Finally, the ENOB is computed for both topologies and several values of comparator noise and comparator offset voltage. The results are compared with the theoretical values achieved with a CR ADC.

In Fig. 7.4, the ENOB as function of the comparator noise, normalized to $V_{\mathrm{LSB}}$, is shown. The simulations reveal that the ENOB of the MOSCAP-based ADC is greatly improved when compared to the conventional CS-ADC with linear capacitors, and only slightly worse than the theoretical ENOB of a CR-ADC. For instance, with an input-referred comparator noise of $0.5V_{\mathrm{LSB}}$, the ENOB of the MOSCAP-based CS-ADC is approximately 7.8 versus 8.0 of the CR, while the conventional CS achieves only 7.3 bits.

Figure 7.5 shows the effective resolution as a function of the comparator offset voltage for the three same topologies. The linearity of the CR ADC is not affected by comparator offset, as this parameter produces only an offset in the ADC transfer curve. Thus, the CR maintains ideal ENOB for the range of simulated offset voltages. Regarding the CS topologies, we notice a remarkable improvement in the tolerance to offset in the MOSCAP-based ADC. While the conventional CS-ADC achieves approximately 6 bits of ENOB with an input-referred comparator offset voltage of $20V_{\mathrm{LSB}}$, the MOSCAP-based ADC performs with more than 8 effective bits. In addition, the MOSCAP-based ADC achieves nearly ideal ENOBs if the

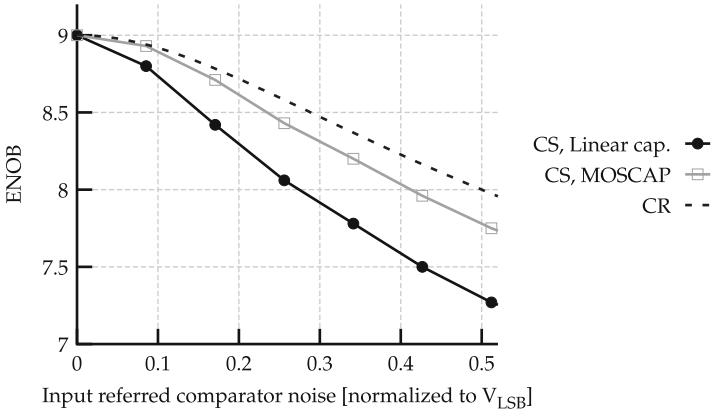



**Fig. 7.4** Comparison of ENOB versus input-referred comparator noise with a CR-ADC and with CS-ADCs with linear capacitors and the proposed MOSCAP-DAC, obtained with full-circuit simulations

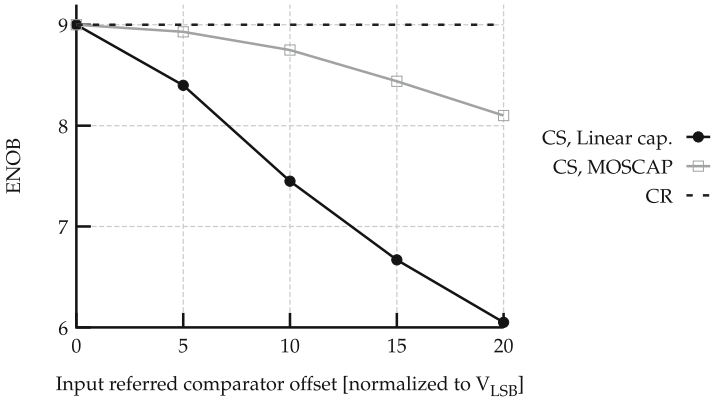



**Fig. 7.5** Comparison of ENOB versus input-referred comparator offset voltage with a CR-ADC and with CS-ADCs with linear capacitors and the proposed MOSCAP-DAC, obtained with full-circuit simulations

comparator offset voltage is equal to or smaller than $5V_{\mathrm{LSB}}$. An interesting and important remark is that the curves of Figs. 7.4 and 7.5 are almost identical if simulated in process corners.

The boost in performance concerning linearity and noise is proportional to the ratio between the capacitances of the DAC cell at precharge and charge processing phases. This ratio, $\beta_C$, can be approximated by (7.2).

$$\beta_C = \frac{C_{\mathrm{GG}}(V_{\mathrm{PC}}) + C_{\mathrm{GG}}(-V_{\mathrm{PC}})}{2C_{\mathrm{GG}}(0)}. \tag{7.2}$$

We can find the values of $C_{GG}$ inspecting Fig. 7.1. For the simulated MOSCAPs and $V_{PC} = 0.6\,V$, $\beta_C$ is approximately 3. In view of these facts, we can draw a vital conclusion regarding the use of MOSCAPs in CS-ADCs: the effect of the nonlinearity of the MOSCAP is similar to the effect of increasing $\alpha$ (the effects of increasing $\alpha$ are discussed in Chap. 4). This is justifiable, as the DAC capacitance is large only during the precharge cycle, and is significantly reduced during binary search, thus reducing the attenuation during conversion. Consequently, we introduce the quantity $\hat{\alpha}$, which represents the apparent capacitance gain in a CS-ADC based on a MOSCAP-DAC. The value of $\hat{\alpha}$ is found by rewriting $\alpha$ to include $\beta_C$. The resulting expression is given in (7.3).

$$\hat{\alpha} = \frac{\beta_C C_{TH}}{2C_{DAC}}. \tag{7.3}$$

Note that with a MOSCAP-DAC, we are not required to use a $V_{PC}$ larger than $V_{IR}/2$, as would be required for linear capacitors according to (4.25), to achieve an apparent capacitance gain larger than unity. A coarse approximation of the ENOB as function of $\beta_C$ for a CS-ADC that employs nonlinear capacitors is obtained by plugging (7.3) in (4.67) and (4.97). Figure 7.6 shows the results obtained analytically and by simulation. The best-fit curve of the analytical model to the simulated points is also shown in the plot, and leads to $\beta_C = 2.75$, very close to the value of 3 estimated with (7.2). The small mismatch is the result of different components of $C_{GG}$ ($C_{GD}$, $C_{GS}$, and $C_{GB}$) not being considered individually. The body, differently to the drain and source, is tied to ground and has a different effect on the magnitude of $V_{PN}$ predicted by (7.1) (which includes only the capacitors between nodes P and N).

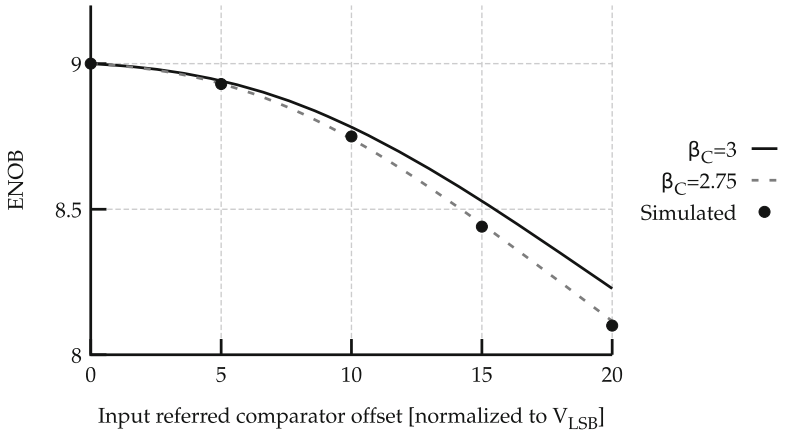



**Fig. 7.6** ENOB achieved with full-circuit simulation and the analytical model of (4.67), (4.97), and (7.3), with $\beta_C = 3$ found graphically and with the best-fit $\beta_C = 2.75$

## 7.3 SAR ADC Implementation

We designed a LVLP 9-bit SAR ADC using the proposed MOSCAP-DAC in a 0.13 µm CMOS process. In Fig. 7.7, we show a block diagram of the designed ADC. Like the design presented in Chap. 6, this prototype uses local voltage boosting [5] to allow operation under low supply voltages. The ADC is designed to operate with a low-voltage supply not larger than half the process nominal voltage of 1.2 V and the control signals can be doubled without reliability issues. Also, the prototype employs the same digital self-timed SAR controller as the one presented in Chap. 6. The other sub-circuits are a configurable TH, a MOSCAP-DAC, a comparator, and a set of VBs. This section describes the implementation details of these blocks.

### *7.3.1 Comparator*

The comparator employed in this design is a StrongARM latch [6, 7], optimized for low-power according to the procedure described in Chap. 5. We designed the comparator to present an input-referred noise close to 0.5 mV. For the ADC operating with an IR of 1.7 V, the value of input-referred comparator noise corresponds to $0.15V_{\mathrm{LSB}}$ and, according to Fig. 7.4, limits the achievable ENOB to approximately 8.8 bits. The standard deviation of the input-referred comparator offset voltage, $\sigma_{\mathrm{offset}}$, is limited in this design to 8 mV, or approximately $2.4V_{\mathrm{LSB}}$. For a maximum acceptable drop in ENOB due to comparator offset of 0.25 bits, this spread of offset represents a 4-$\sigma$ yield, according to Fig. 7.5. To put differently, it means that if the comparator offset voltage follows a normal probability distribution, approximately

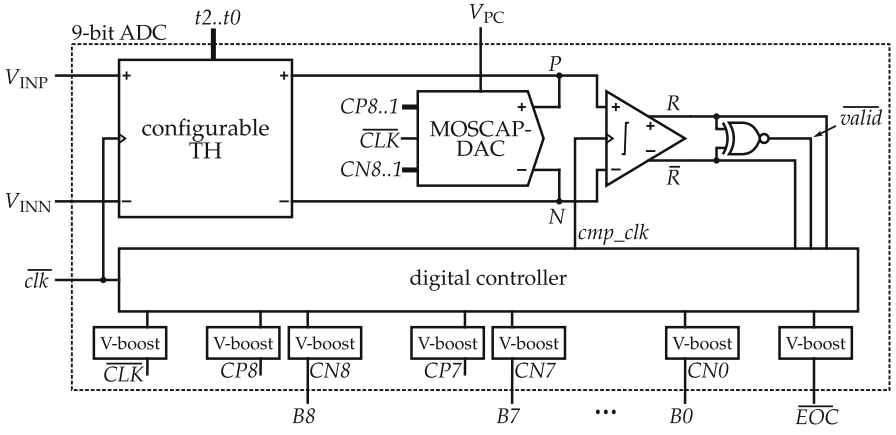



**Fig. 7.7** Blocks diagram of the SAR ADC prototype

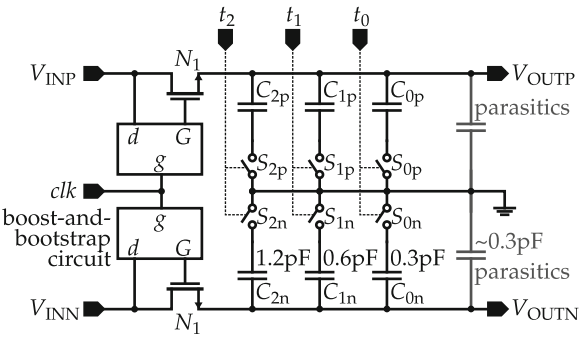



**Fig. 7.8** Configurable TH circuit

99.994 % of the fabricated samples present a drop in ENOB caused by comparator offset lower than 0.25 bits. If the same comparator is used in a conventional CS-ADC with linear capacitors, the yield is only 1.3-$\sigma$.

### 7.3.2 *Configurable TH with Boost-and-Bootstrap Switches*

The CS architecture permits control over the ADC IR by changing the total capacitance of the TH. Therefore, we propose the configurable TH of Fig. 7.8. The digital control signals $t_2 \ldots t_0$ drive $n$-type MOS-switches that connect the bottom plates of the TH capacitors to ground. If a switch is on, the corresponding capacitor is summed up to the total TH capacitance. A wide range of programmable input capacitance is achieved by the use of capacitors with binary weights (0.3, 0.6, and 1.2 pF). In addition, the routing parasitics contribute with extra 0.3 pF.

To allow the TH to operate with low values of supply voltage and process over-rails signals, the boost-and-bootstrap circuit represented in Fig. 7.9a is proposed and used in the TH switches. The operation principle is exemplified in Fig. 7.9b, assuming a single-ended input signal of $1.5 \times V_{DD}$ at the drain terminal $d$ of the sampling switch. The circuit drives the gate of $N_{BS}$ with a $V_{GS}$ of 2 $V_{DD}$ during the sampling phase. Since this prototype operates with low supply voltages, which are equal to or lower than $V_{DD}/2$, this poses no reliability issues on the oxide of the MOSFETs.

For a voltage supply of 0.6 V, with $V_{PC} = V_{DD}$, the proposed TH allows an IR of 1.7 $V_{P-P}$, equivalent to a gain of approximately 40 % over a rail-to-rail implementation. An ADC with IR extending over-rails presents interesting advantages over rail-to-rail implementations. First, since SAR ADCs are very switching-intensive, the total energy burned in a conversion is proportional to $V_{DD}{}^2$. Therefore, if the speed requirements allow, great energy savings may be achieved by scaling $V_{DD}$. In implementations that are limited to rail-to-rail inputs, this would also impose a diminished IR, ultimately harming the ADC SNR. In the case of the CS-ADC, an over-rails IR is obtained at the expense of reducing $\alpha$, leading to

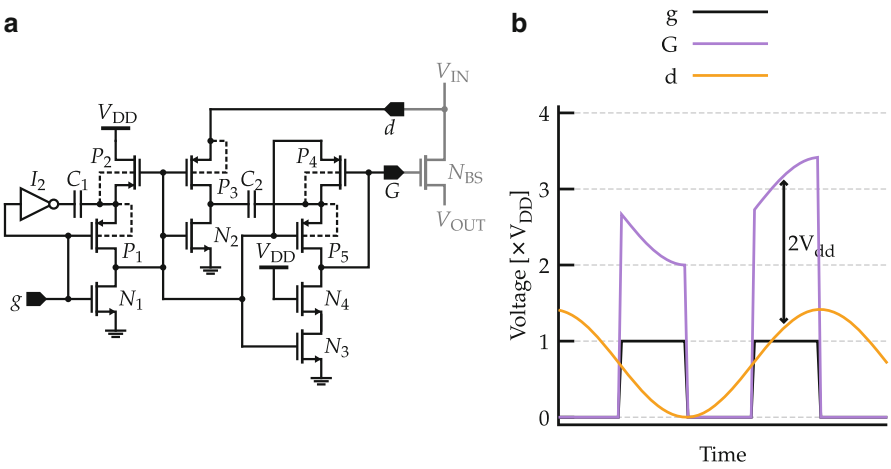



**Fig. 7.9** Boost-and-bootstrap circuit for the TH switches: (**a**) schematic and (**b**) illustrative waveforms

more attenuation. Alternatively, if $V_{PC}$ is maintained fixed, and the TH capacitor is resized to accommodate the increase in IR (differently than the analysis carried out in Chap. 4, where $V_{PC}$ was scaled to accommodate a fixed IR), the voltage range of the comparator inputs is indeed increased. This increase is the result of two opposing effects, that take place while lowering $\alpha$ for a same $V_{PC}$: larger IR, and consequently more dynamic range; and worse attenuation due to dynamic total capacitance. The simulated dynamic range at the comparator inputs as function of the successive approximation algorithm cycle, for a rail-to-rail CS-ADC ($V_{IR} = 2V_{DD} = 1.2\,V$) and for the proposed ADC with increased IR ($V_{IR} = 1.7\,V$, and $V_{DD} = 0.6\,V$) are shown in Fig. 7.10. We also show in the figure the percentage increase in the comparator voltage range of the over-rails ADC when compared to a rail-to-rail implementation. For all the cycles, the comparator is driven by a signal with a voltage range which is at least 17 % larger in the case of over-rails IR.

### *7.3.3 Voltage Boosters*

The presented 9-bit ADC uses 19 VBs: eight pairs for the eight DAC capacitors; one to drive the switches responsible for precharging the DAC capacitances; two additional to level-shift the LSB $B_0$ and the end-of-conversion signal ($\overline{EOC}$) to the same voltage of the remaining digital outputs. As a relatively large number of boosters are required in the circuit, these circuits must be compact and consume little power. The circuit presented previously in Sect. 6.3.2 conforms to these constraints [5] and is employed in the prototype.

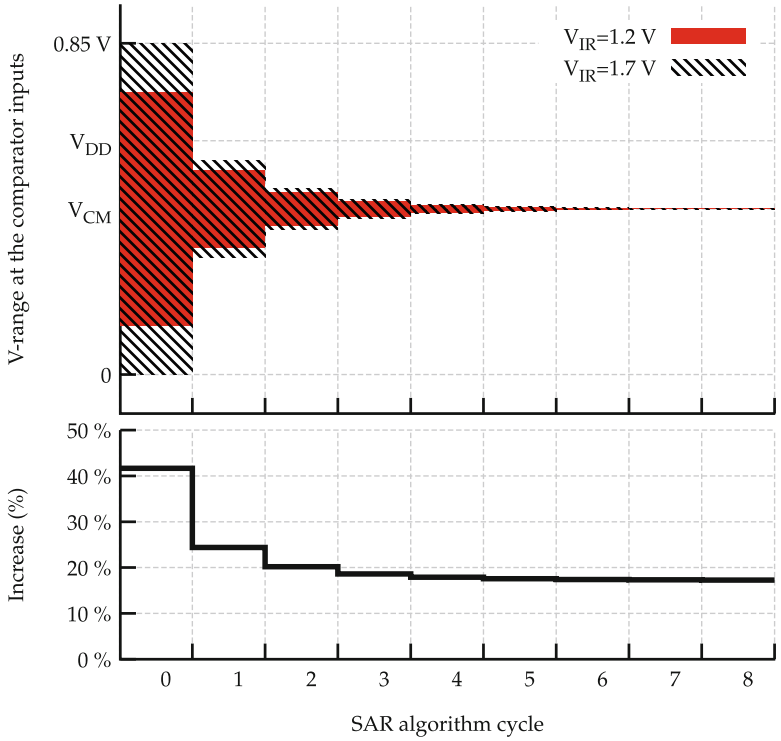



**Fig. 7.10** Simulated voltage range at the comparator inputs, for the proposed over-rails ADC($V_{IR} = 1.7\,V$) and for a rail-to-rail CS ADC($V_{IR} = 1.2\,V$). The percentage increase in comparator voltage range for $V_{IR} = 1.7\,V$ compared to $V_{IR} = 1.2\,V$ is also shown. A reference voltage of 0.6 V is considered

**Table 7.1** Results of a 10-k points Monte-Carlo simulation, showing the spread of total charge stored in 20-fF capacitors, precharged to 0.6 V, at 27 °C

| | Absolute (%) | Relative (%) | Total (%) |
|---|---|---|---|
| MOS | 1.54 | 0.43 | 1.59 |
| MIM | 4.51 | 0.17 | 4.51 |
| MOM | 6.71 | 6.10 | 9.03 |

### *7.3.4 DAC Cells*

We show the DAC cells in Fig. 7.2. All the switches are simple minimum-size *n*-type MOSFETs. The switches conductance is improved by the boosted control signals. The MOSCAPs present very good matching characteristics. This is noticeable in Table 7.1, which compares the simulated absolute variation (mismatch between devices in different dies) and relative variation (mismatch between devices on the

same die) of the total charge in capacitors with the same value of capacitance and three different implementations in the employed 0.13-µm technology: MOS, MIM, and MOM. The values presented in the table are for room temperature and a bias voltage of 0.6 V. The value of 20 fF is the minimum value for MIM capacitors in the technology. Although the relative matching of the MIM capacitor is better than the MOS by a factor of more than two, the area required for the MIM capacitor (including vias) is more than 20× the area of the MOSCAP (also including vias) in this process. Furthermore, the matching of MOSCAPs is one order of magnitude better than MOMs according to the Monte-Carlo models in this technology. Importantly, the models for MIM capacitors provided by the foundry do not include voltage nonlinearity, which may be significant in modern CMOS processes depending on the type of insulator employed [2, 3], and could potentially impact on the $Q(V)$ characteristic of the capacitor.

As the design of this prototype was focused on the validation of the topology, we employed a very conservative capacitance of 11 fF, even though the DNL/INL constraints could be met with much smaller capacitors. This value of capacitance is obtained with a MOSCAP area of approximately 1 µm$^2$, for $V_{PC} = 0.6$ V. It is noteworthy that the aspect ratio ($W/L$) of the transistor has a strong impact on the $C(V)$ curve. We illustrate this in Fig. 7.11, which compares three MOSCAPs with different aspect ratios and similar maximum capacitance. The figure reveals that a large aspect ratio increases the threshold voltage of the MOSCAP and enlarges the zone with reduced capacitance. Simultaneously, $\beta_C$ is reduced due to the larger areas of the drain and source, which lead to larger parasitic capacitances. On the other hand, a small aspect ratio narrows the zone of reduced capacitance and increases the value of $\beta_C$, since the drain and source terminals present less parasitic capacitance. As discussed in this chapter, increasing $\beta_C$ has a positive impact on the tolerance to comparator offset and noise, and we choose a MOSCAP with minimum $W = 0.16$ µm and $L = 6.5$ µm (please see considerations on NQS effects in Appendix).

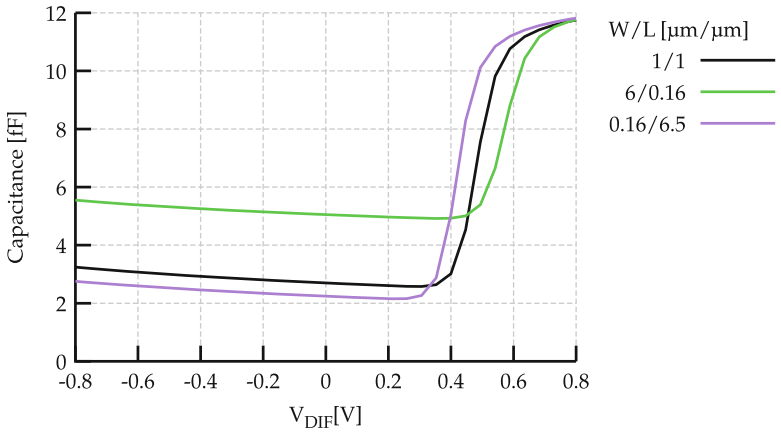



**Fig. 7.11** Simulated $C(V)$ curve of a MOSCAP for three values of aspect ratio $W/L$, with approximately the same MOSCAP area, using the circuit of Fig. 7.1b

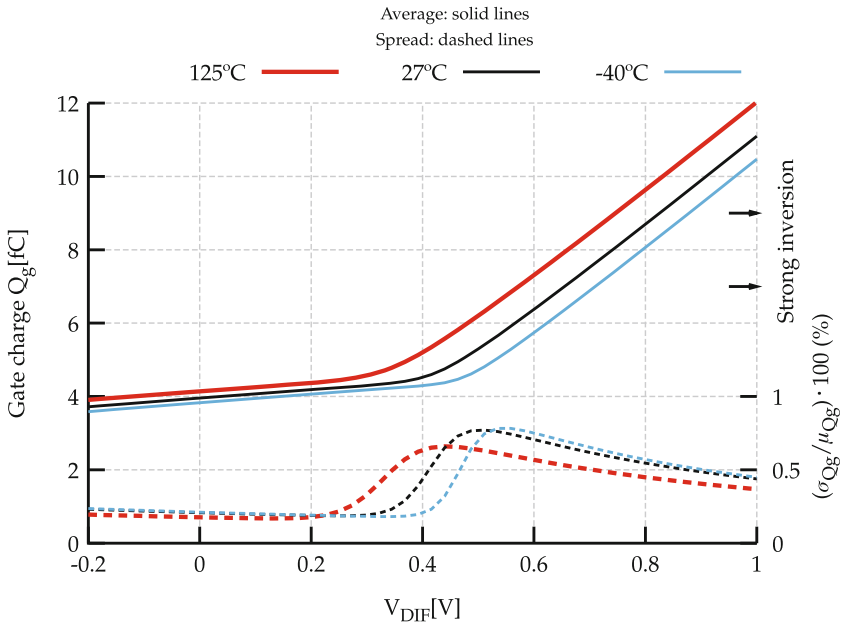



**Fig. 7.12** Average and standard deviation of the C(V) curve of MOSCAPs at different temperatures, using the circuit of Fig. 7.1b, computed from a 300-point Monte-Carlo simulation. *Solid lines* represent mean values of capacitance and *dashed lines* represent the standard deviation normalized to the average, in percent

Another noteworthy aspect of the MOSCAPs is the impact of temperature changes in the $Q(V)$ curve and in the matching characteristics. This can be visualized in Fig. 7.12. At any of the simulated temperatures and $V_{\mathrm{DIF}} \geq 0.6\,\mathrm{V}$, the mismatch is maintained under 0.75 %, having a negligible impact on the ADC performance. The layout of a few MOSCAPs used in the MSB capacitor of the DAC is shown in Fig. 7.13.

## 7.4 Experimental Results

The proposed ADC was fabricated in a 0.13-μm CMOS process. Figure 7.14 shows a microphotograph of the prototype, annotated with the arrangement of the blocks on the final layout. The total active area is $0.046\,\mathrm{mm}^2$, and nearly 40 % is used to accommodate the voltage boosting capacitors. An auxiliary offset auto-zeroing circuit for the comparator is included in the chip. This sub-circuit, however, is disabled during the measurements, as it revealed to have a negligible impact on the performance of the ADC, which is in line with the anticipated increase in robustness of the proposed topology to the comparator offset. Still, the area of this unused

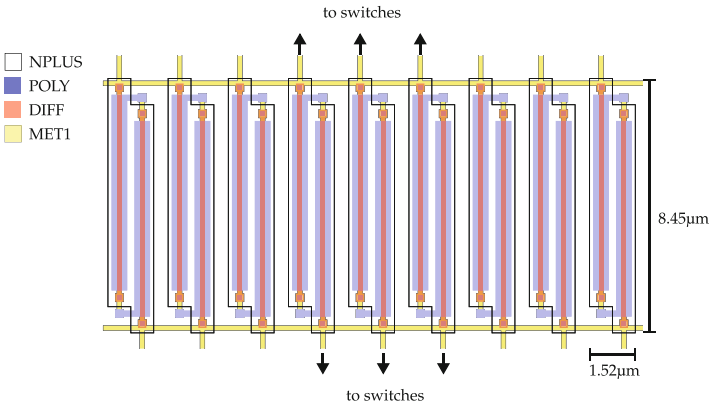



**Fig. 7.13** Partial layout of the DAC MOSCAPs

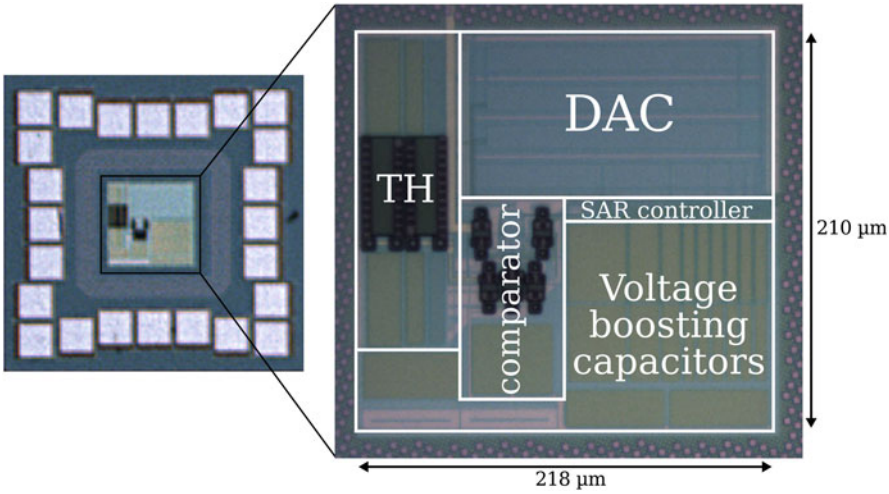



**Fig. 7.14** 9-bit SAR ADC chip micrograph

circuit is accounted for the total die area. A total of 5 samples are measured, and the chip with the lowest ENOB is selected for the measurement results presented in this chapter. The spread of ENOB, $\sigma_{\text{ENOB}}$, is below 0.1 bit.

The test circuit is shown in Fig. 7.15. We use a FPGA board to extract the ADC output, and transfer this data to a computer for post-processing. The front-end relies on a transformer to convert the single-ended input into differential. The test PCB is placed into a temperature chamber for the temperature measurements.

Measured at room temperature, with 0.6 V of supply and reference voltages, full-range sinusoidal input near Nyquist frequency and 1 MSps of sampling frequency, the ADC consumes a total of 2.78 µW. In Fig. 7.16, we show the DNL and INL under these conditions. The DNL/INL curves present larger peaks at the first and fourth

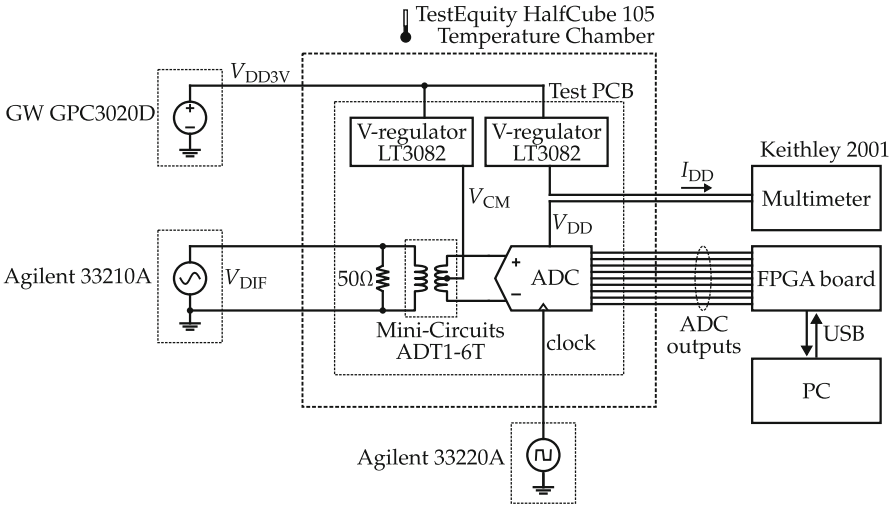



**Fig. 7.15** Test circuit for the measurements

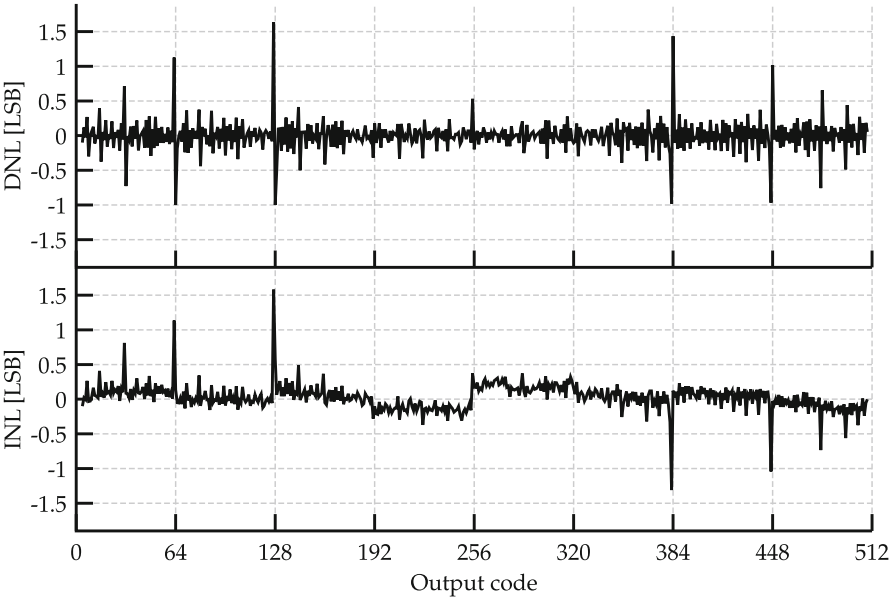



**Fig. 7.16** Measured DNL and INL of the prototype ADC

than in the second and third quarters of the curve. This pattern is consistent among the measured dies, indicating a systematic error. A careful inspection revealed that the cause of the static errors is unbalanced parasitics on the two sides of the DAC unit-cells. This becomes evident by contrasting the DNL/INL curves of the

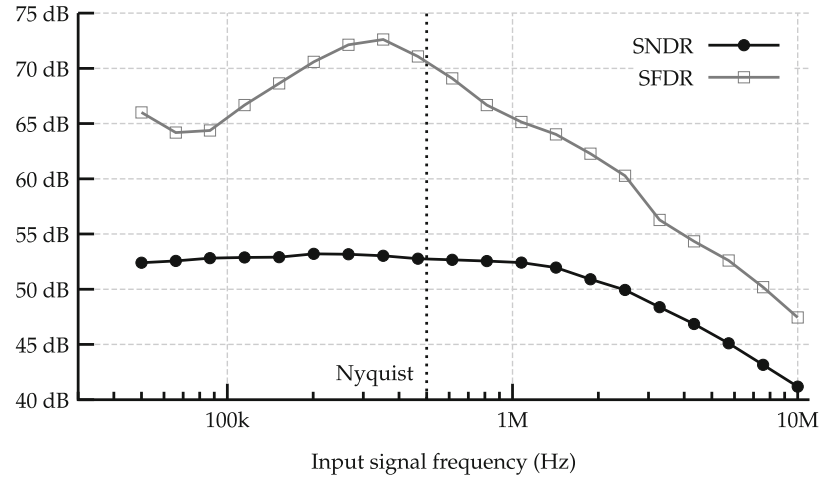



**Fig. 7.17** Measured SNDR and SFDR as function of the input frequency for a full-range signal

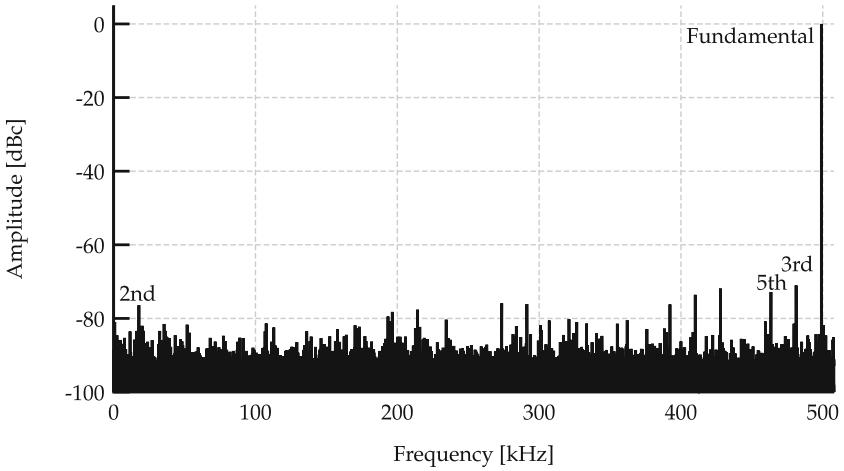



**Fig. 7.18** Measured output spectrum of a near-Nyquist frequency full-range input signal

prototype (Fig. 7.16) with the simulated DNL/INL curves of an ADC with only unbalanced parasitics as a nonideality (Fig. 4.11). The ENOB measured with a full-range input sinusoid at Nyquist frequency is 8.48 bits, while SNR, SNDR, SFDR, and THD are 53 dB, 52.76 dB, 71.01 dB, and −65.55 dB, respectively. In Fig. 7.17, the SNDR and SFDR are shown for various values of input frequency. The output spectrum for a near-Nyquist frequency sinusoidal input is shown in Fig. 7.18. The resulting FOM at Nyquist frequency is 7.8 fJ/conv.step.

For the temperature measurements, we used a sinusoidal input with 85 kHz and 1.4 $V_{P-P}$, to mimic more realistic operating conditions. Reference and supply

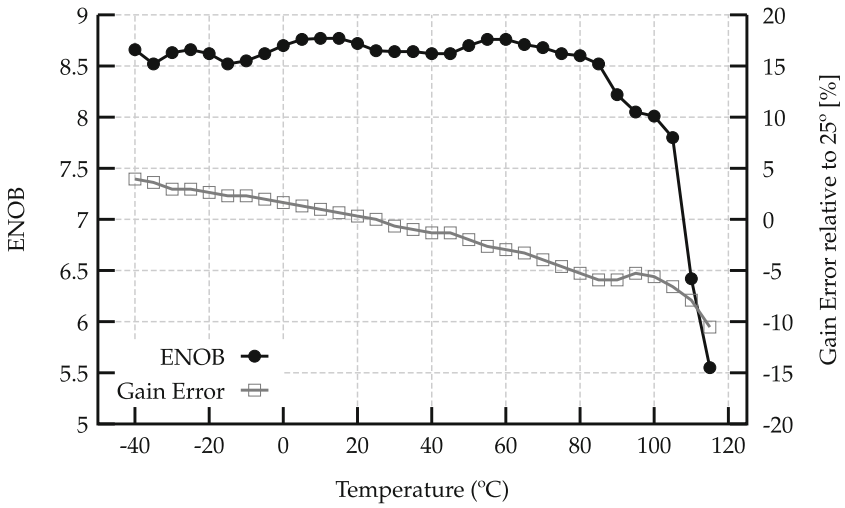



**Fig. 7.19** Measured effective resolution as function of the temperature, at 1 MSps with a 85-kHz 1.4-$V_{P-P}$ input sinusoid

voltages are 0.6 V. The temperature is spanned from −40 to 115 °C, and the ENOB and gain error relative to 25 °C are measured. The results are presented in Fig. 7.19. For the full temperature operating range, the ADC can work with 1 MSps of sampling rate. From −40 to 85 °C, the ENOB remains larger than 8.5 bits. After 100 °C, the ENOB drops below 8 bits, and at 110 °C, the ENOB drops below 6.5 bits. The gain error in the range of temperatures with ENOB>8.5 spans from −5.9 % to 4.3 %, at 85 and −40 °C, respectively. Thus, this implementation is prepared for commercial and industrial temperature ranges (0–85 and -40–100 °C, respectively). We found that the circuit limiting the operation at higher temperatures is the controller self-timing feedback loop, which does not provide sufficient frequency range for higher temperatures. Fortunately, this issue can be solved by increasing the width of the transistor connected to $V_{BIAS}$, and could potentially lead to a wider temperature operating range.

We also measured the prototype at room temperature with 0.4 V of supply and reference voltages with a full-range sinusoidal input near Nyquist frequency [8]. For these measurements, the TH is reprogrammed to the minimum capacitance, in which only parasitic capacitances are used to sample the input signal. At 0.4 V, the ADC operates at 300 kSps and burns 0.354 µW. The ADC performs with an ENOB of 8.01, which yields in a FOM of 4.6 fJ/conv.step. It should be noted, however, that with this supply, the ADC is not tolerant to large temperature variations. The reason is that the precharge takes place at a point of the $C(V)$ curve with a steep derivative, which is largely changed with temperature. To increase the temperature range at this supply voltage, transistors with low-$V_t$ should be employed. A summary of the measurement results with 0.6 and 0.4 V of supply is presented in Table 7.2.

**Table 7.2** Summary of measurements for the prototype with $V_{DD} = V_{PC} = 0.4\,V$ and $V_{DD} = V_{PC} = 0.6\,V$

| $V_{PC} = V_{DD}$ | 0.4 V | 0.6 V |
|---|---|---|
| Input capacitance ($C_{TH}$) | 0.3 pF | 2.4 pF |
| Sampling rate | 300 kSps | 1 MSps |
| Power | 0.354 μW | 2.78 μW |
| Differential input range ($V_{IR}$) | 1.6 V | 1.7 V |
| $V_{LSB}$ | 3.13 mV | 3.32 mV |
| $V_{IR}/V_{PC}$ | 4.0 | 2.83 |
| SNDR[a] | 49.75 dB | 53 dB |
| SFDR[a] | 67.1 dB | 71.01 dB |
| ENOB[a] | 8.01 | 8.48 |
| DNL | 1.12 | 1.58 |
| INL | −1.25 | 1.58 |
| Temperature range[b] (ENOB>8.5) | – | −40 °C–85°C |
| FOM | 4.6 fJ/conv.step | 7.8 fJ/conv.step |

[a]Nyquist, full-range input
[b]85 kHz, 80 % of full-range input

## 7.5 Discussion

This chapter presented a CS-ADC that mitigates the main drawbacks of previous CS-ADC implementations: reduced tolerance to comparator offset and noise. The design is based on a MOSCAP-DAC, that obviates highly linear capacitors in the DAC construction. While the TH of the proposed ADC still requires linear capacitors, the requirements for TH capacitors are much less stringent in terms of matching and density. Nevertheless, a configurable TH circuitry is included in the design, to allow operation under different reference voltages and input ranges. The use of a boost-and-bootstrap switch in the TH circuit increases the input swing of the ADC to 1.7 $V_{P-P}$ when supplied by 0.6 V, and to 1.6 $V_{P-P}$ when supplied by 0.4 V. Good agreement is found between the measured ENOB and the maximum expected ENOB, theoretically limited to approximately 8.8 bits by the comparator noise (Sect. 7.3.1). The proposed topology was validated with a low-voltage ADC, but it is not limited to this category of designs. The use of a higher precharge voltage is encouraged, owing to the fact that the matching properties and the temperature sensitivity of the gate capacitance improve as the MOSCAP enters in the strong inversion region (as seen in Fig. 7.12).

## Appendix: Frequency-Dependency of $C(V)$ in MOSCAPs

One important aspect of MOSCAPs is that its $C(V)$ characteristics are frequency-dependent [4]. To have a better understanding on this phenomenon, we should analyze the circuit used to extract the $C(V)$ curves of a MOSCAP, shown in Fig. 7.20. The voltage source $v_{ac}$ injects an ac signal with small amplitude, and $V_{CM}$ and $V_{DIF}$ are dc. The capacitance at a determined bias is found with (7.4), where $f$ and $v_{ac}$ are the frequency and magnitude of the injected ac signal.

$$C = \frac{\mathrm{Im}(I_{CAP})}{2\pi f v_{ac}}. \tag{7.4}$$

To extract the $C(V)$ curve, $V_{DIF}$ is swept and the values of capacitance found with (7.4) are plotted versus $V_{DIF}$. However, the curve presents different shapes consonant to the choice of frequency of $v_{ac}$, if NQS effects are taken into account [4]. Specifically, for sufficiently high frequencies, the inversion layer is not able to build up fast enough, and the capacitance is drastically reduced in that region. The simulated curve for several values of $v_{ac}$ frequency, ranging from 1 Hz to 500 MHz, is shown in Fig. 7.21. As a remark, although most of the transistor models include NQS effects, they may be disabled by default in a simulation. In the case of the BSIM3v3 model, the model of NQS effects is applicable for both large signal transient and small signal ac analysis. Even though good accuracy is expected from these models [9], especially when an enhanced modeling methodology such as [10] is employed, in this work the simulated value was used only as a coarse estimate to guide the MOSCAP-DAC sizing.

The results in Fig. 7.21 suggest that the NQS effects may affect the pre-charge time of the MOSCAP-DAC. In spite of the fact that the MOSCAPs are pre-charged to a reference voltage $V_{PC}$ that is dc, the settling time allocated for this process is finite. To evaluate the impact of the NQS effects to the pre-charge cycle of the MOSCAP-DAC, the settling time of a single unit-capacitor when subject to a voltage step is evaluated using the circuit of Fig. 7.22, and shown in Fig. 7.23. The gate charge is found integrating $I_{CAP}$. When the NQS effects are not taken into account, the RC constant of the circuit is in the range of hundreds of picoseconds,

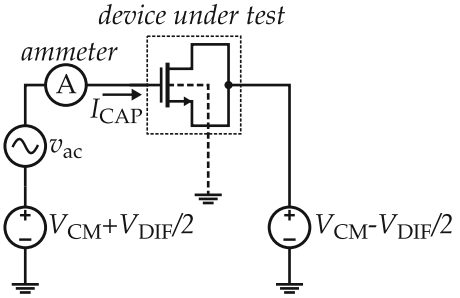



**Fig. 7.20** Circuit used to simulate the $C(V)$ curve of MOSCAPs

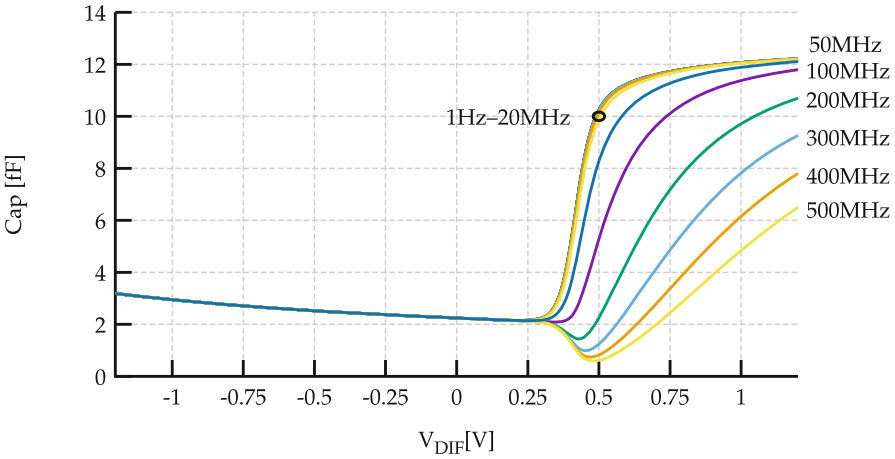
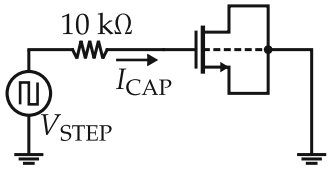



**Fig. 7.21** Simulated $C(V)$ for several values of frequency of the $v_{ac}$ source, ranging from 1 Hz to 500 MHz, simulated with the circuit of Fig. 7.20, with $V_{CM} = 0.3$ V. The size of the MOSCAP is $W = 0.16\,\mu$m and $L = 6.5\,\mu$m

**Fig. 7.22** Circuit used to simulate the impact of NQS behavior in the pre-charge of MOSCAPs. The size of the MOSCAP is $W = 0.16\,\mu$m and $L = 6.5\,\mu$m



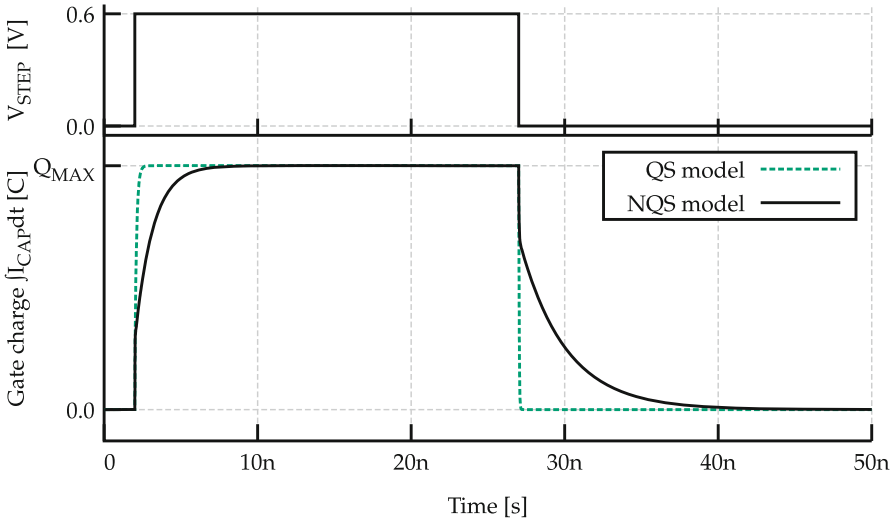




**Fig. 7.23** Simulated pre-charge of a MOSCAP, using quasi-static (QS) and NQS models

dictated by the 10 kΩ resistor and the MOSCAP capacitance. On the other hand, when NQS effects are included in the simulation, the RC constant is increased to several nanoseconds. According to this analysis, a few tens of nanoseconds should be allocated to pre-charge and discharge the MOSCAPs. For this specific design, this is not an issue, but may be a limiting factor in implementations with faster sampling speeds. A possible counter-measure is to use shorter transistors while trading-off $\beta_C$ for a faster response due to limited NQS effects.

## References


1. B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, J. Craninckx, A 70 dB DR 10b 0-to-80 MS/s current-integrating SAR ADC with adaptive dynamic range. IEEE J. Solid State Circuits **49**(5), 1173–1183 (2014). doi:10.1109/JSSC.2014.2309086 (cit. on p. 134)
2. P. Gonon, C. Vallée, Modeling of nonlinearities in the capacitance voltage characteristics of high-k metal-insulator-metal capacitors. Appl. Phys. Lett. **90**(14), 142906-1–142906-3 (2007). doi:10.1063/1.2719618 (cit. on pp. 133 and 144)
3. F. El Kamel, P. Gonon, C. Vallée, Experimental evidence for the role of electrodes and oxygen vacancies in voltage nonlinearities observed in high-k metal-insulator-metal capacitors. Appl. Phys. Lett. **91**(17), 10–13 (2007). doi:10.1063/1.2803221 (cit. on pp. 133 and 144)
4. C. Hu, *Modern Semiconductor Devices for Integrated Circuits*, 1st edn. (Prentice Hall, Englewood Cliffs, 2010) (cit. on pp. 135 and 151)
5. T. Rabuske, F. Rabuske, J. Fernandes, C. Rodrigues, An 8-bit 0.35-V 5.04-fJ/ conversion-step SAR ADC with background self-calibration of comparator offset. IEEE Trans. Very Large Scale Integr. VLSI Syst. **23**(7), 1301–1307 (2015). doi:10.1109/TVLSI.2014.2337236 (cit. on pp. 140 and 142)
6. T. Kobayashi, K. Nogami, T. Shirotori, Y. Fujimoto, O. Watanabe, A current-mode latch sense amplifier and a static power saving input buffer for low-power architecture, in *IEEE Symposium on VLSI Circuits* (1992), pp. 28–29. doi:10.1109/VLSIC.1992.229252 (cit. on p. 140)
7. B. Razavi, The StrongARM latch [a circuit for all seasons]. IEEE Solid State Circuits Mag. **7**(2), 12–17 (2015). doi:10.1109/MSSC.2015.2418155 (cit. on p. 140)
8. T. Rabuske, J. Fernandes, A 9-b 0.4-V charge-mode SAR ADC with 1.6-V input swing and a MOSCAP-only DAC, in *Proceedings of European Solid-State Circuits Conference* (IEEE, New York, 2015), pp. 311–314. isbn: 978-1-4673-7470-5. doi:10.1109/ESSCIRC.2015.7313889 (cit. on p. 149)
9. Y. Cheng, C. Hu, *MOSFET Modeling and BSIM3 User's Guide* (Kluwer Academic Publishers, Norwell, MA, 1999) (cit. on p. 151)
10. SmartSpice BSIM3 v3 Non-quasi Static Model. Silvaco, Inc. (Online). Available: http://www.silvaco.com/tech_lib_EDA/kbase/pages/2900003.html (visited on 01/2016) (cit. on p. 151)

# Chapter 8
# Conclusions and Future Work

**Contents**



## 8.1 Final Remarks

Devices that need to conform to LVLP requirements are already ubiquitous today, and the demand is growing fast. Due to the convenience that the digital domain provides concerning signal processing, data storage, and communication, many LVLP applications rely on an ADC to bring signals from the analog domain into digital. SAR ADCs are the topology of choice for most of these applications, due to its high energetic efficiency. The CS variant of SAR ADCs was recently demonstrated and presents attractive features for LVLP. This book contributed to this topic by providing a detailed analysis of the limitations of the architecture, and by proposing and validating circuit solutions to overcome those.

In Chap. 4, the bottlenecks of the CS-ADC were identified and quantified. Those include parasitics, mismatches, and noise. Based on the results of this analysis, the techniques presented in Chaps. 5–7 were proposed. In Chap. 5, an efficient design/optimization methodology for dynamic comparators that is noise-, power-, and timing-aware is presented and validated. The disclosed methodology enables achieving optimal comparator sizing solutions in less than 30 min in a modern computer, despite the large number of trade-offs involved and the time-variant and nonlinear nature of the clocked comparators. Chapter 6 presented an 8-bit ADC that can work with supplies down to 0.35 V, due to the extensive use of voltage boosting of control signals. Additionally, the ADC makes use of a comparator offset voltage



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8_8

auto-zeroing technique that is consistent with the requirements of LVLP systems. Chapter 7 disclosed an ADC operating with a 0.6-V voltage supply that relies on the strongly nonlinear capacitance of MOSCAPs to mitigate the main shortcomings of CS ADC. The use of MOSCAPs as DAC elements brings additional advantages that are related to the characteristics of this class of integrated capacitors, namely larger density of capacitance and very good matching characteristics.

The performance of the ADCs presented in Chaps. 6 and 7 is contrasted to the state-of-the-art in CS-ADCs and energy-efficient SAR-ADCs in Table 8.1. To the authors' knowledge, the ADC presented in Chap. 7 is the first demonstration of SAR ADC that does not use linear capacitors in the DAC. Additionally, the two ADCs presented in this book are the two charge-sharing ADC with best FOM and the two ADCs (of any topology) with best FOM in technologies down to 0.13 μm.

## 8.2 Future Work

A few possibilities of future investigation appeared during this research. The first emerge from MOSCAPs being 3-terminal devices. While the bulk terminal of all the MOSCAPs is connected to ground in our design presented in Chap. 7, the body terminal can be used to control the capacitance of these devices. This technique could permit analog-domain calibration of the DAC to mitigate process mismatches. While this anticipated body-biasing calibration method would be exclusive to MOSCAP-based DACs, digital-domain calibration algorithms for SAR ADCs, such as [11, 12], can be potentially adapted to the proposed architecture. In that way, the benefits of the MOSCAP-based architecture are maintained while allowing pursuing larger resolutions. Furthermore, some preliminary analysis revealed that MOSCAPs implemented in SOI technologies achieve a larger $\beta_C$ than bulk-CMOS, because the former present lower intrinsic parasitic capacitance due to isolation from the bulk silicon. For instance, a 28-nm SOI MOSCAP presents $\beta_C \approx 5$, while in the 0.13 μm technology employed in this design, $\beta_C \approx 3$, for similar aspect ratios. Given that the improvements in performance of the CS-ADC are strongly correlated to the value of $\beta_C$, we expect a better tolerance to comparator offset and noise in ADCs designed using SOI process.

**Table 8.1** Comparison with state-of-the-art ADCs

| Reference<br>Units | Year<br>– | Tech.<br>– | Area<br>($mm^2$) | Supply<br>(V) | Res.<br>– | S.Rate<br>(MSps) | Mode<br>– | Capacitors<br>– | Temp.<br>(°C) | Diff. IR<br>($V_{P-P}$) | Power<br>(μW) | ENOB<br>– | FOM<br>(fJ/c.-s.) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [1] | 2007 | 90 nm | 0.08 | 1 | 9-b | 20 | CS | MOM | – | – | 290 | 7.4 | 65 |
| [2] | 2008 | 90 nm | 0.09 | 1 | 9-b | 40 | CS | MOM | – | 80 % of R-R | 820 | 8.56 | 54 |
| [3] | 2011 | 90 nm | 0.056 | 1 | 8-b | 40 | CS | MOM | – | – | 113 | 7.75 | 20 |
| [4] | 2014 | 40 nm | 0.08 | 1.1 | 10-b | 80 | CS | TH:MOSCAP DAC:MOM | – | Programmable (max. ∼0.3) | 5450 | 8.71 | 85 |
| [5] | 2011 | 0.18 μm | 0.125 | 0.6 | 10-b | 0.1 | CR | MIM | – | R-to-R | 1.3 | 9.3 | 21 |
| [6] | 2012 | 90 nm | 0.0323 | 0.35 | 10-b | 0.1 | CR | MOM | – | R-to-R | 0.17 | 9.05 | 3.2 |
| [7] | 2012 | 40 nm | 0.0112 | 0.5 | 9-b | 1.1 | CR | MOM | – | R-to-R | 1.2 | 7.5 | 6.3 |
| [8] | 2013 | 90 nm | 0.042 | 0.4 | 10-b | 0.5 | CR | MOM | – | 98 % of R-R | 0.5 | 8.72 | 2.37 |
| [9] | 2013 | 65 nm | 0.076 | 0.6 | 10-b | 0.04 | CR | MOM | – | R-to-R | 0.072 | 9.4 | 2.7 |
| [10] | 2014 | 40 nm | 0.0065 | 0.45 | 10-b | 0.2 | CR | MOM | – | R-to-R | 0.084 | 8.95 | 0.85 |
| 8-bit ADC (Chap. 6) | – | 0.13 μm | 0.0377 | 0.35 | 8-b | 0.3 | CS | MIM | – | R-to-R | 0.0847 | 6.4 | 5.04 |
| 9-bit ADC (Chap. 7) | – | 0.13 μm | 0.046 | 0.6 | 9-b | 1 | CS | TH:MOM DAC:MOS | −40–85 | 1.7 | 2.78 | 8.48 | 7.8 |

## References


1. J. Craninckx, G. van der Plas, A 65fJ/conversion-step 0-to-50MS/s 0-to-0.7mW 9b charge-sharing SAR ADC in 90nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2007), pp. 246–600. doi:10.1109/ISSCC.2007.373386 (cit. on p. 157)
2. V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Van der Plas, J. Craninckx, An 82μw 9b 40MS/s noise-tolerant dynamic-SAR ADC in 90nm digital CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, New York, 2008), pp. 238–610. doi:10.1109/ISSCC.2008.4523145 (cit. on p. 157)
3. J.-H. Tsai, Y.-J. Chen, M.-H. Shen, P.-C. Huang, 1-V, 8b, 40MS/s, 113μW charge-recycling SAR ADC with a 14μW asynchronous controller, in *IEEE Symposium on VLSI Circuits* (2011), pp. 264–265 (cit. on p. 157)
4. J.-H. Tsai, Y.-J. Chen, M.-H. Shen, P.-C. Huang, A 70 dB DR 10b 0-to-80 MS/s current-integrating SAR ADC with adaptive dynamic range. IEEE J. Solid State Circuits **49**(5), 1173–1183 (2014). doi:10.1109/JSSC.2014.2309086 (cit. on p. 157)
5. S.-K. Lee, S.-J. Park, H.-J. Park, J.-Y. Sim, A 21 fJ/conversion-step 100 kS/s 10-bit ADC with a low-noise time-domain comparator for low-power sensor interface. IEEE J. Solid State Circuits **46**(3), 651–659 (2011). ISSN: 0018-9200. doi:10.1109/JSSC2010.2102590 (cit. on p. 157)
6. H.-Y. Tai, H.-W. Chen, H.-S. Chen, A 3.2fJ/c.-s. 0.35V 10b 100kS/s SAR ADC in 90nm CMOS, in *IEEE Symposium on VLSI Circuits* (2012), pp. 92–93. ISBN: 978-1-4673-0849-6. doi:1.1109/VLSIC.2012.6243805 (cit. on p. 157)
7. A. Shikata, R. Sekimoto, T. Kuroda, H. Ishikuro, A 0.5 V 1.1 MS/sec 6.3 fJ/conversion-step SAR-ADC with tri-level comparator in 40 nm CMOS. IEEE J. Solid State Circuits **47**(4), 1022–1030 (2012). ISSN: 0018-9200. doi:10.1109/JSSC.2012.2185352 (cit. on p. 157)
8. C.-Y. Liou, C.-C. Hsieh, A 2.4-to-5.2fJ/conversion-step 10b 0.5-to-4MS/s SAR ADC with charge-average switching DAC in 90nm CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2013), pp. 280–281. doi:10.1109/ISSCC.2013.6487735 (cit. on p. 157)
9. P. Harpe, E. Cantatore, A. van Roermund, A 2.2/2.7fJ/conversion-step 10/12b 40kS/s SAR ADC with data-driven noise reduction, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2013), pp. 270–271. doi:10.1109/ISSCC.2013.6487730 (cit. on p. 157)
10. H.-Y. Tai, Y.-S. Hu, H.-W. Chen, H.-S. Chen, A 0.85fJ/conversion-step 10b 200kS/s sub-ranging SAR ADC in 40nm CMOS, in *IEEE International Solid-State Circuits Conference (ISSCC)* (2014), pp. 196–197. doi:10.1109/ISSCC.2014.6757397 (cit. on p. 157)
11. W. Liu, P. Huang, Y. Chiu, A 12-bit, 45-MS/s, 3-mW redundant successive-approximation-register analog-to-digital converter with digital calibration. IEEE J. Solid State Circuits **46**(11), 2661–2672 (2011). doi:10.1109/JSSC.2011.2163556 (cit. on p. 156)
12. W. Liu, P. Huang, Y. Chiu, A 12-bit 50-MS/s 3.3-mW SAR ADC with background digital calibration, in *Proceedings of IEEE Custom Integrated Circuits Conference (CICC)* (2012). doi:10.1109/CICC.2012.6330694 (cit. on p. 156)

# Acronyms

**ADC** analog-to-digital converter

**BEOL** back-end-of-line

**CDF** cumulative distribution function
**CMOS** complementary metal-oxide-semiconductor
**CPU** central processing unit
**CR** charge-redistribution
**CS** charge-sharing

**DAC** digital-to-analog converter
**DNL** differential nonlinearity

**ENOB** effective number of bits
**EOC** end-of-conversion

**FFT** Fast Fourier Transform
**FOM** figure-of-merit
**FPGA** Field Programmable Gate Array

**GA** genetic algorithm
**GPS** global positioning system
**GPU** graphics processing unit

**INL** integral nonlinearity
**IR** input range
**ISF** input sensitivity function
**ISSCC** International Solid-State Circuits Conference



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8

**LSB** least-significant bit
**LTV** linear time-varying
**LVLP** low-voltage low-power

**MCS** merged capacitor switching
**MDAC** multiplying digital-to-analog converter
**MIM** metal-insulator-metal
**MOM** metal-oxide-metal
**MOS** metal-oxide-semiconductor
**MOSCAP** metal-oxide-semiconductor capacitor
**MOSFET** metal-oxide-semiconductor field-effect transistor
**MSB** most-significant bit

**NQS** non-quasi static
**NSGA-II** Non-dominated Sorting Genetic Algorithm-II

**PAC** periodic-ac
**PMU** power management unit
**PNOISE** periodic noise
**PSD** power spectral density
**PSS** periodic steady-state
**PVT** process, voltage, temperature

**RAM** random access memory
**RF** radio-frequency

**SAR** successive approximation register
**SFDR** spurious-free dynamic ratio
**SNDR** signal-to-noise-and-distortion ratio
**SNR** signal-to-noise ratio
**SOI** silicon-on-insulator

**TH** track-and-hold
**THD** total harmonic distortion
**TSPC** true-single-phase-clocked

**VB** voltage booster
**VLSIC** Symposium on VLSI Circuits

**WSN** wireless sensor network

# Symbols

$\Delta_{Cp}$ Parasitic capacitance mismatch
$\Delta_C$ Capacitance mismatch
$\Delta_G$ ADC gain error
$\alpha$ Capacitance ratio
$\hat{\alpha}$ Effective value of $\alpha$
$B$ Number of bits
$C_0$ Unitary capacitance
$C_{CAL}$ Calibration capacitor
$C_{DAC}$ Total DAC capacitance
$C_{GB}$ Capacitance between the gate and the bulk of a MOSFET
$C_{GD}$ Capacitance between the gate and the drain of a MOSFET
$C_{GG}$ Total gate capacitance of a MOSFET
$C_{GS}$ Capacitance between the gate and the source of a MOSFET
$C_p$ Parasitic capacitance
$C_{TH}$ Track-and-hold capacitor
$C_{THN}$ Track-and-hold capacitor (negative terminal)
$C_{THP}$ Track-and-hold capacitor (positive terminal)
ENOB Effective number of bits
$E_{REF}$ Energy of the reference source
$G_{ADC}$ ADC gain, defined by the ratio between the input range $V_{IR}$ and the reference voltage $V_{REF}$ or the precharge voltage $V_{PC}$
$i$ Cycle in the conversion (0 corresponds to the MSB cycle )
$I_{CAP}$ Capacitor current
$I_{REF}$ Current of the reference source
$K$ Conversion result, $K = k_{B-1}..., k_1, k_0$
$Q_N$ Charge on the node corresponding to the comparator negative input
$Q_P$ Charge on the node corresponding to the comparator positive input
$Q_{THN}$ Charge on the track-and-hold negative terminal
$Q_{THP}$ Charge on the track-and-hold positive terminal



T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8

$Q_{TOP}$ Charge on the top plate of the capacitor
SNDR Signal to noise-and-distortion ratio
$V_{BIAS}$ Bias voltage
$V_{CAL}$ Calibration voltage
$V_{CM}$ Common-mode voltage
$V_{DD}$ Positive supply voltage
$V_{DIF}$ Differential voltage
$V_{GS}$ Voltage between the gate and the source of a MOSFET
$V_{IN,CM}$ Common-mode voltage of the input
$V_{IN,DIF}$ Differential voltage at the ADC inputs
$V_{INN}$ Voltage at the negative ADC input
$V_{INP}$ Voltage at the positive ADC input
$V_{IR}$ Input-range voltage
$V_{LSB}$ Voltage of one LSB
$V_N$ Voltage at the negative comparator input
$V_{NOISE}$ Noise voltage
$V_{OS}$ Input-referred comparator offset voltage
$V_P$ Voltage at the positive comparator input
$V_{PC}$ Precharge voltage source, used to charge the DAC capacitors in the beginning of the conversion
$V_{PN}$ Differential voltage at the comparator inputs
$V_{P-P}$ Volts peak-to-peak
$V_{REF}$ Reference voltage
$V_{SB}$ Voltage between the source and the bulk of a MOSFET
$V_t$ Threshold voltage of a MOSFET
$V_{TOP}$ Voltage on the top plate of the capacitor

# Index





T. Rabuske, J. Fernandes, *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*, Analog Circuits and Signal Processing,
DOI 10.1007/978-3-319-39624-8